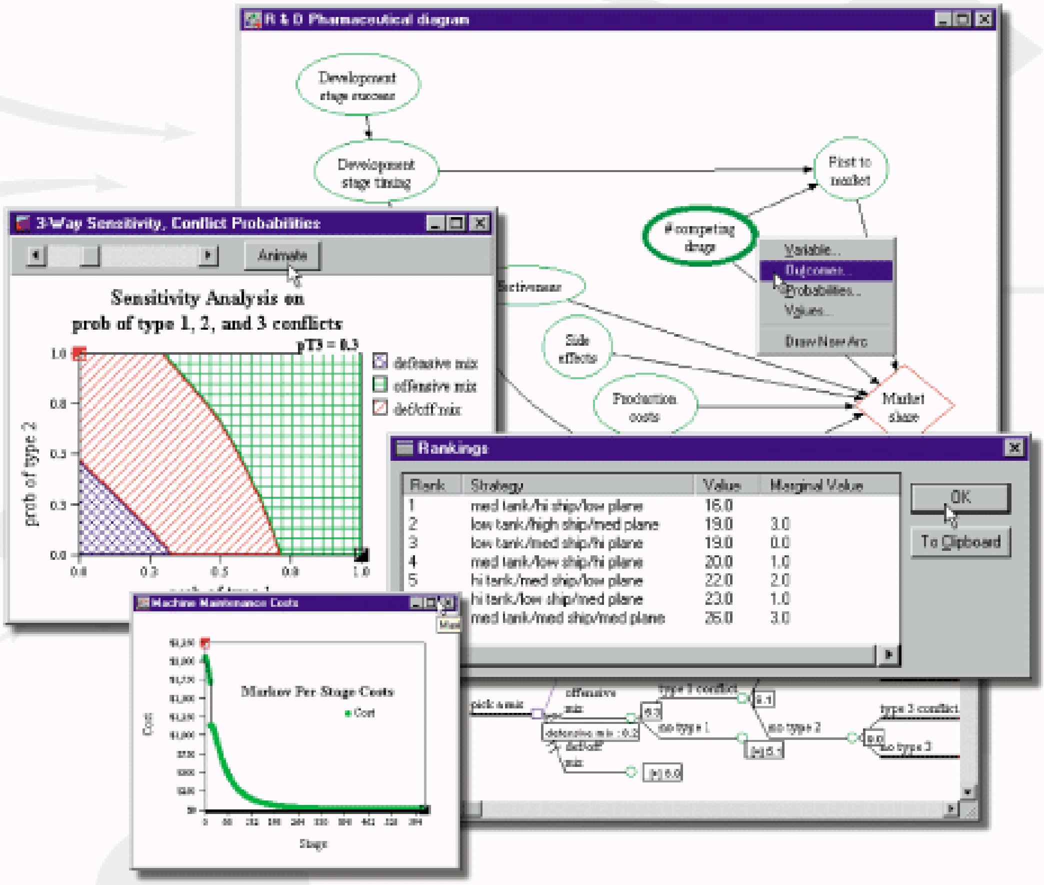
TREEAGE
SOFTWARE, INC.



Rankings

Rank	Strategy	Value	Marginal Value
1	med tank/hi ship/low plane	16.0	
2	low tank/high ship/med plane	19.0	3.0
3	low tank/med ship/hi plane	19.0	0.0
4	med tank/low ship/hi plane	20.0	1.0
5	hi tank/med ship/low plane	22.0	2.0
	hi tank/low ship/med plane	23.0	1.0
	med tank/med ship/med plane	26.0	3.0

DATA™ 3.5 User's Manual

COPYRIGHT

Copyright © 1999 by TreeAge Software, Inc. All rights reserved. No part of this manual may be reproduced in any manner or translated into another language without the express, written permission of TreeAge Software, Inc.

LICENSES AND TRADEMARKS

Decision Analysis by TreeAge, DATA, and DATAScript are trademarks of TreeAge Software, Inc. MS-DOS, Windows, Office and Microsoft are registered trademarks of Microsoft Corporation. Macintosh is a registered trademark of Apple Computer, Inc.

LIMITED WARRANTY

TreeAge Software, Inc. warrants that your original copy of DATA is free from any defects in media. This Limited Warranty is of unlimited duration but applies only to the original purchaser. If a media defect occurs, you may return the disk to TreeAge Software, Inc. for a free replacement, so long as your registration is on file.

DATA is being furnished to you on the basis of a non-exclusive license and limited warranty which are set forth in a separate document.

NEITHER TREEAGE SOFTWARE, INC. NOR ITS LICENSORS, NOR ITS RESELLERS MAKE ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. NEITHER TREEAGE SOFTWARE, INC., NOR ITS LICENSORS, NOR ITS RESELLERS WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR PERFORMANCE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME JURISDICTIONS. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL TREEAGE SOFTWARE, INC., ITS LICENSORS, ITS RESELLERS, OR THEIR RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, RELIANCE, OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF ANY OF THE FOREGOING COMPANIES OR INDIVIDUALS SHALL HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. THE LIABILITY OF TREEAGE SOFTWARE, INC., ITS RESELLERS, OR THEIR RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS TO YOU FOR ACTUAL DAMAGES FROM ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO THE PURCHASE PRICE OF THE SOFTWARE. THE LIABILITY OF TREEAGE SOFTWARE, INC.'S LICENSORS OR THEIR RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS TO YOU FOR ACTUAL DAMAGES FROM ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO $50.

TREEAGE
SOFTWARE, INC.

**1075 Main Street
Williamstown, MA 01267**

**Voice: 413.458-0104
Fax: 413.458-0105
E-mail: info@treeage.com
Web: http://www.treeage.com**

Contents

PART I: INTRODUCTION

CHAPTER 1: GETTING STARTED 13
Organization of this manual 14
Installation and system requirements 15
Getting technical support 15
Differences from DATA 3.0 16
A simple problem: How should I invest $1,000? 18

CHAPTER 2: DECISION ANALYSIS PRIMER 19
Decision trees 21
Influence diagrams 24

PART II: LEARNING TO USE DATA

CHAPTER 3: BUILDING YOUR MODEL AS A DECISION TREE 27
Naming nodes and adding branches 27
Entering payoff values 28
Entering probabilities 29
Setting calculation preferences 29
Calculating the tree 30

CHAPTER 4: BUILDING YOUR MODEL AS AN INFLUENCE DIAGRAM 31
Using influence diagrams 31
Creating and naming nodes 32
Assigning alternatives and outcomes 33
Identifying influences 34
How to draw an arc 35
Assigning values and probabilities 36
Assigning values to “Profit” 37

The basics of asymmetry 38
Entering the asymmetry 39
Viewing the converted tree 40

CHAPTER 5: MAKING CHANGES TO YOUR TREE 43
Using variables 43
Defining a probability as a variable 44
Defining a probability as a variable expression 46
Defining a payoff as a variable or expression 47
Defining a variable using other variables 47
Cut, copy, paste, and clear 50
Inserting, deleting, and reordering branches 52

CHAPTER 6: ANALYZING YOUR MODEL 55
Expected value 55
Roll back 55
Rankings 57
Assessing risk 57
Standard deviation 57
Probability distribution 59
Cumulative and comparative probability distributions 59
Assessing uncertainty 61
Performing a sensitivity analysis 62
Thresholds 64

CHAPTER 7: PRINTING 67
Print preview 67
Printing preferences 69

PART III: IMPROVING YOUR PRODUCTIVITY WITHIN DATA

CHAPTER 8: VARIABLES – CONCEPT AND THEORY 71
Representing model values 71
Using variables and formulas 77
Where payoff variables should be defined 79
Variable definitions and sensitivity analysis 81
Probability variables 84
Defining one variable in terms of other variables 85
Defining variables recursively 87

CHAPTER 9: VARIABLES – TOOLS AND TECHNIQUES 91

HOW TO DEFINE A VARIABLE 91

The Define Variable window 92

Creating multiple definitions 93

USING THE QUICK MENU (WINDOWS) 93

USING FUNCTIONS 93

INSERTING VARIABLE NAMES IN FORMULAS 94

MODIFYING AND DELETING VARIABLES AND DEFINITIONS 95

VARIABLES DISPLAY 96

VARIABLES REPORT 97

THE PROPERTIES DIALOG BOX 98

THE VARIABLES WINDOW 101

THE EVALUATOR 104

SLIDERS 105

CHAPTER 10: CUSTOMIZING DATA'S DISPLAY 107

VALUES DISPLAY 107

Displaying payoff names 107

Terminal Node Columns 108

Terminal node numbers 110

Numeric Formatting 111

Roll back display options 112

Variables display 113

Hiding values 115

TREE STRUCTURE 115

Label nodes 115

Lining things up 115

Tree compression 116

OTHER DISPLAY FEATURES 117

Annotation 118

Changing fonts 120

Zooming 122

CHAPTER 11: SELECTING NODES 125

SELECTING A SUBTREE 125

SELECTING MULTIPLE NODES 126

SELECTING NODES BY CHARACTERISTIC 127

CHAPTER 12: MANAGING LARGE TREES 129

CLONING SUBTREES 130

Nested trees 133

Sensitivity analysis and nested trees 135

Collapse subtree 136

Influence diagrams 136

CHAPTER 13: STORING ANALYSES 137

Using graph templates with stored analyses 138

Using stored analyses with a custom interface tree 139

CHAPTER 14: MISCELLANEOUS PRODUCTIVITY FEATURES 141

Node comments 141

Find/replace 142

Probability wheel 143

Shortcuts 144

Quick menu 144

Undo 144

Numeric entry shortcuts 147

Optimal path 147

Show optimal path 147

Force path 148

Change optimal path 148

PART IV: WORKING WITH OTHER APPLICATIONS

CHAPTER 15: BASIC LINKING 149

Dynamic Data Exchange (Windows) 150

Using bi-directional links (Windows) 155

Publish and subscribe (Macintosh) 155

CHAPTER 16: BI-DIRECTIONAL LINKING 159

Calculating payoffs using bi-directional links 159

Other uses of bi-directional links 160

Setting up bi-directional links 161

Calculations under bi-directional links 163

CHAPTER 17: EXPORTING GRAPHICS AND ANALYSIS DATA 167

Exporting pictures 167

DATA rollback (.TRB) files 169

Exporting graph data 170

CHAPTER 18: BUILDING CUSTOM DATA APPLICATIONS 171
DATA INTERACTIVE™ 171
RUN-TIME DATA 172
Creating a basic custom interface 174
Creating an extended custom interface 177
Protecting your intellectual property 179

PART V: CALCULATION METHODS

CHAPTER 19: SPECIFYING WHAT DATA CALCULATES 181
CHANGING THE CALCULATION METHOD 181
CHANGING THE OPTIMAL PATH CRITERION 182
Cost-effectiveness optimal path parameters 183
Reversing the optimal path 183
CHANGING THE QUANTITY TO BE CALCULATED 183

CHAPTER 20: MULTI-ATTRIBUTE ANALYSIS 187
SETTING UP A MULTI-ATTRIBUTE MODEL 188
ENTERING MORE THAN ONE PAYOFF FORMULA 189

CHAPTER 21: COST-EFFECTIVENESS ANALYSIS 191
HOW DATA CALCULATES COST-EFFECTIVENESS 191
PREPARING A TREE FOR COST-EFFECTIVENESS CALCULATIONS 192
Cost-effectiveness optimal path parameters 192
DECISION MAKING USING COST-EFFECTIVENESS 193
Cost-effectiveness graph 195
Dominance 195
MONTE CARLO SIMULATION 197

PART VI: ADVANCED ANALYSIS AND MODELING FEATURES

CHAPTER 22: ADVANCED SENSITIVITY ANALYSIS 199
VIEWING THE CHANGING VALUE OF A SINGLE SCENARIO 199
INTERPRETING MULTIPLE THRESHOLDS 199
ONE-WAY SENSITIVITY ANALYSIS IN COST-EFFECTIVENESS MODELS 200
SENSITIVITY ANALYSIS OPTIONS 202
Correlated variables 203
Variables with non-numeric definitions 204
Variables with more than one definition 205

Two-way sensitivity analysis 207
Two-way cost-effectiveness sensitivity analysis 209
Isocontours 210
Three-way sensitivity analysis 211
Tornado diagrams 212
Including correlated variables in the tornado diagram 216
Threshold analysis 216

CHAPTER 23: EXPECTED VALUE OF PERFECT INFORMATION 221
Avoiding EVPI's pitfalls 223

CHAPTER 24: BAYES′ REVISION 225
A brief introduction to Bayes′ revision 226
Bayes′ revision in DATA 229
Using variables in the Bayes′ revision dialog 234

CHAPTER 25: BASIC MARKOV MODELING 237
Recursive processes 237
Basic components of a Markov model 238
Building a Markov model in DATA 240
Specifying transitions and absorbing states 241
Assigning rewards 243
Setting the termination condition 245
Using Markov keywords 246
Analyzing the Markov model 248
Markov analysis 248
Monte Carlo simulation 249

CHAPTER 26:TABLES 251
How tables are stored 251
Contents of a table 251
Lookup method 252
Creating tables 252

CHAPTER 27: ADVANCED MARKOV MODELING 257
Cycle-dependent values 257
Initial and final rewards 260
Prior costs 261
Half-cycle correction 262
Markov transition rewards 263

Cost-effectiveness Markov models 265
Cloning Markov subtrees 267
Markov bindings 267
Tunnel states 271
Creating a tunnel state 272
Using the _tunnel keyword 274
Violating Markov strictures 275
Logic nodes and statements 277

CHAPTER 28: DISTRIBUTIONS 279
Monte Carlo simulation 279
Creating a distribution 280
Using the DistSamp() function 282
How DATA calculates distributions 282
Correlating distributions 283
Resampling during Markov processes 284
Custom distributions 285
Distributions and chance nodes 287

CHAPTER 29: MONTE CARLO SIMULATION 291
First-order simulation 292
Second-order simulation 293
Decision nodes 294
Tracker variables 295
Trackers as outputs 296
Using trackers in model logic 296
Making tracker modifications 298
Reproducing identical results 300
Monte Carlo text report 300

CHAPTER 30: RISK PREFERENCE FUNCTIONS 303
Certainty equivalents and risk aversion 304
Two types of risk preference function 305
Risk preference curves 307

CHAPTER 31: WORKING WITH INFLUENCE DIAGRAMS 309
When to use influence diagrams 309
Model size and other considerations 310
Limitations of an influence diagram 310
Time ordering of nodes 311

Asymmetry 313
Variables and values 314
Node variables 314
Deterministic nodes 317
Using the assessment window 318
Using variables in the text editor 319
Probability wheel 320
Linked values (Windows only) 320
Aligning nodes 321
Arc operations 322

CHAPTER 32: ADVANCED INFLUENCE DIAGRAM FEATURES 323
Bayes´ revision 323
Setting up a single forecast 324
Entering probability data 324
Seeing the results 326
Asymmetry inside the Bayesian model 327
Bayes´ revision with sequential tests 327
Expected value of perfect information 328
Clones 329
Sub-models 329

CHAPTER 33: WORKING WITH GRAPH WINDOWS 331
Customizing individual graphs 331
Viewing a graph's underlying numbers 335
Bar graphs 335
Line graphs 336
Region graphs 337
Tornado diagrams 338
Cost-effectiveness graphs 338
Storing preferences for future graphs 338
Creating a graph template 339

CHAPTER 34: MISCELLANEOUS ADVANCED FEATURES 341
Logic nodes 341
Identifying the range of potential payoffs 342

APPENDIX A: MENU AND TOOL BAR REFERENCE 345

Tool bar 364

Navigation Button 364

Status bar 365

APPENDIX B: PREFERENCES DIALOG 367

APPENDIX C: FUNCTIONS AND OPERATORS 385

Operators 385

Operator precedence 386

Built-in functions 387

APPENDIX D: DISTRIBUTIONS 393

APPENDIX E: TECHNICAL NOTES 399

Avoiding sensitivity analysis errors 399

Changing the storage location for tables 403

Details of Markov process calculations 403

Using DATA's command line 405

APPENDIX F: FOR SMLTREE AND DECISION MAKER USERS 409

INDEX 417

CHAPTER 1 GETTING STARTED

Welcome to DATA 3.5!

DECISION ANALYSIS by TREEAGE (DATA™) has been designed to implement the techniques of decision analysis in an intuitive and easy-to-use manner. It transforms decision analysis from a potentially tedious exercise into an easily applied and highly visual means of (1) organizing the decision making process, (2) analyzing the problem at hand, and (3) communicating both the structure of the problem and the basis for the decision reached.

Using decision analysis, a problem is disaggregated into components small enough to be readily understood and analyzed. Next, these components are used to model the problem's essential elements. Under this methodology, the possible events (decisions and uncertainties), together with the relations among them, are expressly identified.

This explicit identification of the sequence and linkage of events is, by itself, of great value in clarifying complex decisions. But decision analysis, as implemented by DATA, does much more. By calculating the value of each chain of events, and by weighting uncertain results by the probability of each possible outcome, the decision maker can evaluate each intermediate point of the model and identify the alternatives that will maximize value, or minimize costs, depending on the objective.

Whatever the objective or problem being modeled, DATA reduces the complexity of decision analysis, both in the initial stage of formulating and structuring the problem and later in calculating and testing the elements of the analysis. DATA's clear, graphical presentation of both model and results enhances communications at each phase of the decision making process.

Those of you experienced in decision analysis will find DATA easy to use following only a cursory review of the software commands, although the richness of the program will become more apparent with further study of the manual. If you have no, or only limited, experience with decision analysis, DATA will make it much easier to learn.

Organization of this manual

This manual is for use with DATA 3.5, both Microsoft® Windows™ and Apple® Macintosh® versions. With limited exceptions, all of the screenshots are taken from DATA for Windows, so if you are using DATA for Macintosh, full screen views and dialog boxes pictured in this manual will not look exactly like the ones shown on your screen. Most of these differences will be cosmetic; substantively, the two versions of DATA are largely identical. In those limited situations where substantive differences exist, they are identified in the text and separate instructions are given.

The instructions assume that you are familiar with the most basic operations of your operating system. These include start-up, use of the mouse to select and drag objects, opening and closing files, and use of the pull-down menus. If unfamiliar with these basics, you should take a few minutes to review them in your Windows or Macintosh user's manual.

Part I of this manual (Chapters 1–2) provides an introduction to the software and to decision analysis in general, including hints for users of prior versions of DATA or other decision analysis software. It also introduces a simple investment decision problem, which will be used in subsequent sections to illustrate the process of decision analysis and its implementation in DATA. If you are unfamiliar with either decision trees or influence diagrams, you should read Chapter 2 ("Decision Analysis Primer") for a general tutorial on decision trees and influence diagrams. Otherwise, you may choose to skip Chapter 2 and proceed to Part II for instruction on using DATA.

Part II (Chapters 3–7) consists of a hands-on tutorial in which you will learn, step by step, how to build the investment model outlined in Chapter 2. First, you will be shown how to set up the model as a decision tree and as an influence diagram, each of which graphically describes the problem. You will then learn how DATA performs calculations designed to assist the decision maker in making the best possible, informed decision. At this stage, you will be ready to draw and calculate basic decision trees and influence diagrams on your own.

Part III (Chapters 8–14) documents many features designed to improve your productivity in using DATA. Following along with the detailed tutorial, you will be able to construct a decision tree of your own design, using variables, a host of shortcuts, and many useful modeling techniques. The features described in Part III are not necessary to the performance of basic decision analysis, but they can enhance your speed and flexibility in working with DATA.

Part IV (Chapters 15–18) covers a number of important topics related to the use of DATA in combination with other software, including spreadsheets, Internet/intranet servers, and Visual Basic applications.

Part V (Chapters 19–21) covers DATA's modeling and analysis of multi-attribute decision trees, including cost-effectiveness analysis.

Part VI (Chapters 22–34) covers DATA's advanced features. Each chapter in Part VI documents a particular feature or group of related features, such as multivariate sensitivity analysis, Monte Carlo simulation, Markov processes, and others. Each chapter contains an overview of the feature, an example of how to use it and, where appropriate, references to other chapters.

Two chapters in Part VI detail DATA's powerful influence diagram interface. These chapters cover everything from the basics of influence diagrams to advanced topics such as asymmetry, Bayes' revision with multiple tests, expected value of perfect information, and conversion to analyzable decision trees.

Following Part VI are several appendices with technical information about individual DATA features, including import facilities, error-checking mechanisms, calculation algorithms, and formula derivations. A comprehensive index for the entire manual follows the appendices.

Installation and system requirements

To install DATA, please follow the separate installation instructions that accompany this manual. These instructions also cover hardware and software (operating system) requirements.

At the conclusion of the installation process, you will be given the opportunity to register your software license. It is important that you do so. Unless we have your registration on file, you will not be able to count on receiving free technical support, notice of software updates, or special prices on software upgrades.

From time to time, we will issue maintenance upgrades to fix bugs and, sometimes, add new features. These will be available for free downloading from our web site. You are urged to check in regularly.

Getting technical support

There are several ways to get help in using the software. It is likely that the answers to most of your questions can be found in this manual.

If, after checking the manual, you still need assistance, please try the following:

- Visit our web site. At http://www.treeage.com you will find a variety of information for users of the software.
- E-mail us at techsup@treeage.com. In addition to a detailed description of the problem, we will need to know which release of DATA 3.5 you are using (see About DATA under DATA's Help Menu) and the serial number. Any technical support questions you e-mail to us will be answered quickly. Be sure to include your telephone number in the message, as we may want to discuss the problem with you by telephone.
- You also may call us at (413) 458-0104 for assistance. Leave a message, including which release of DATA 3.5 you are using (see About DATA under DATA's Help Menu) and the serial number; we will return the call as soon as possible.

Hints for users of earlier versions of DATA

DATA 3.5 can read files created by all prior versions of DATA for Macintosh, DATA for Windows, and DATA for DOS.

If you are familiar with a previous version of DATA, see the following sections for the important differences between earlier versions of DATA and DATA 3.5.

Differences from DATA 3.0

New Parser – DATA's internal calculations are markedly faster in version 3.5. Most calculations will require only 25% to 75% of the time required under DATA 3.0.

Cost-effectiveness – DATA 3.5 easily manages many of the complexities of cost-effectiveness modeling. At decision nodes, the optimal alternative can be identified based on a user-specified threshold marginal cost-effectiveness. One-way sensitivity analysis output can be displayed graphically in a number of ways: cost vs. effectiveness, effectiveness vs. cost, and variable vs. average or marginal values. Two-way region graphs can include isocontours. See Chapters 21–22.

Marginal Lines (Isocontours) – Lines can be added to a two-way sensitivity graph showing marginal values, in addition to the threshold line normally displayed where the marginal value is zero. Isocontours are available for any two-way sensitivity analysis comparing two options. See Chapter 22.

Markov Analysis – DATA 3.5 enables more powerful cost-effectiveness Markov models which calculate in half the time. Also new are Markov state bindings, more flexible transition rewards, and greater reporting detail. See Chapters 25 and 27.

Monte Carlo Simulation – Separate statistics and graphs for cost and effectiveness can be concurrently viewed. The report includes each alternative's cost and effectiveness expected values, when simulations are performed at a decision node. Variables, and even other distributions, can be used as parameters in distributions. See Chapters 28–29.

Correlated Variables – Positive or negative correlations can be established between any variables in your tree. When performing a sensitivity analysis on variables with correlations, you have the option of concurrently varying any or all correlated variables. See Chapter 22.

Recursive Variables – DATA 3.5 supports recursive variable definitions. See Chapter 9.

Built-in Functions – New functions make it possible to move easily between probabilities, odds ratios, and rates. See Appendix C.

Bi-directional Links – Bi-directional links now use ActiveX technology, improving speed and robustness. See Chapter 16.

Endnode display – Columns of information, including node numbers, marginal values, and custom calculations, can be displayed next to the right-most nodes of a rolled-back tree. See Chapter 10**.**

Reports – DATA 3.5 can automatically generate customizable reports on the variables and tables used in your tree. See Chapters 9 and 26.

For a more comprehensive list of changes, please visit the TreeAge web site at http://www.treeage.com.

DATA 2.6 or earlier for Windows, Macintosh, and DOS

DATA 3.0 introduced a number of features, capabilities, and interface enhancements not available in earlier versions of DATA. If you are not already familiar with DATA 3.0, you are encouraged to go through the tutorial in Part II and the productivity features in Part III. Thereafter, you will find it helpful to refer to specific chapters in Part IV–VI and to the appendices to learn about individual features in DATA 3.5.

Differences between Windows and Macintosh versions

There are only limited differences between the Windows and Macintosh versions of DATA 3.5. As you proceed through the manual, you will find these differences noted at appropriate points. For example:

- pressing the ENTER key (Windows) is equivalent to pressing the RETURN key (Macintosh);
- in DATA for Windows, links are maintained with either Dynamic Data Exchange or ActiveX technology, while DATA for Macintosh utilizes Publish and Subscribe;
- in DATA for Macintosh, CONTROL-clicking is usually the equivalent of right-clicking in DATA for Windows;
- certain features in DATA for Windows, such as bi-directional links, are not available in DATA for Macintosh; and
- DATA for Windows can export graphics files as either metafiles or bitmaps, while DATA for Macintosh exports graphics as PICT files.

A simple problem: How should I invest $1,000?

You have $1,000 to invest, and have eliminated from consideration all but two possible investments. One is a potentially volatile equity (stock) investment; the other is a risk-free certificate of deposit (CD). Your decision has a one-year time horizon: you will reconsider your investment decision at the end of one year, but not earlier.

The CD pays simple interest at a rate of 5% annually. For simplicity's sake, let's assume that if you buy the stock, there are only two possibilities: at the end of the year, its market value will have gone up and you make $500, or it will have gone down and you lose $600. You assign a 60% likelihood to the former result.

You are sufficiently wealthy that the possible loss of $600 does not pose a material threat.

For a primer on employing the techniques of decision analysis to model this problem, turn to Chapter 2. If you are already familiar with the fundamentals of decision analysis, you may prefer to go directly to Chapter 3 to learn how to build a decision tree in DATA that models this problem.

Chapter 2 Decision Analysis Primer

Spreadsheet software has made it possible to apply the speed and precision of a personal computer to the basic tasks of business analysis. In using such software to deal with multiplying sources of data, analysts often attempt to calculate the effects of scenario upon scenario, producing reports in such volume that decision makers begin to long for the day of pencil and paper and problems of too little, rather than too much, information.

The fundamental difficulty with this type of analysis is that decision makers, whether they are business managers, government regulators, engineers, economists, attorneys, or physicians, generally face problems which require more than computational ability: what is the probability that a particular R&D project will be successful; what toxic waste remediation technique offers the best balance of cost and effectiveness; what is the settlement value of recently commenced, treble-damage antitrust litigation; should a promising, but risky, treatment for AIDS be cleared for use before clinical testing is complete?

The analysis of these problems is seriously complicated by uncertainty because, invariably, the decision maker lacks control over the consequences of one or more of the scenarios under consideration. Thus, in addition to the types of factors amenable to spreadsheet analysis, the decision maker must make subjective judgments about the likelihood of particular scenarios, and make decisions based not only on costs and benefits, but also on assessments of risk.

The appropriate way to deal with these problems is through decision analysis, a structured methodology that first puts the uncertainties into perspective and then takes them into account in the decision making process.

Decision analysis rests on the concept of expected value (also called *expectation*). This concept is commonly illustrated using a gambling example:

> If someone has a 1 in 4 chance of winning $100, then the expected value of the gamble (the gambler’s *expectation*) is 1/4 of $100, or $25. If given the chance to purchase the “right” to this uncertain payoff prior to the event (e.g. spinning of a wheel, drawing of lots), a reasonable decision would be to pay no more than $25.

Another example, stated a different way:

> If 25% of the money wagered in a lottery is paid out in prizes, then every dollar you spend on a lottery ticket has an expected return of twenty-five cents.

This latter example does not mean that you, as the holder of a $1 ticket, will win (or even have the possibility of winning) exactly twenty-five cents when the lottery results are announced. Rather, most lottery tickets will return nothing, while others will pay out substantially more than the dollar paid in.

The assignment (while the outcome of the lottery remains uncertain) of a twenty-five cent “expected value” to every $1 lottery ticket indicates that (1) the ratio of all money paid as lottery winnings to all money spent on lottery tickets is estimated to be one to four, and (2) consistent with the concept of probabilistic independence, every ticket you or someone else buys offers the same chance of winning.

Determining the expected value of an event is a cornerstone of decision analysis. There are other important considerations when structuring decisions. For example, apples should be compared with other apples, not with anything else. If you need to compare apples with peaches, you must convert both their values to a common unit of measurement, such as dollars (or calories).

Two methodologies are available in DATA 3.5 for modeling a decision analysis problem. They have distinctly different means of visually representing the problem.

The first is through a *decision tree*, a branching structure in which each branch represents an event that may take place in the future. The second is through an *influence diagram*, in which each node represents a different factor that influences the outcome, and arcs between the nodes specify the ways in which one factor influences another.

While DATA's influence diagrams can be extremely helpful for simplifying and presenting complex decisions, they must be converted into decision trees in order to be analyzed. Therefore, the next section of this chapter explains the structure and analysis of decision trees, and the final section describes the design of influence diagrams as an alternative method of generating an analyzable decision tree.

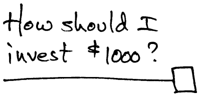
Decision trees

The design of a decision tree is subject to a few guidelines:

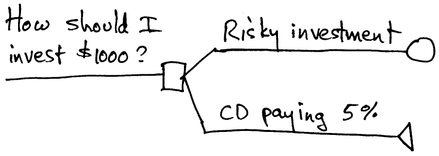
1. *Time flows from left to right.* Decision trees are horizontal structures which proceed from left to right. Each successive branch represents an event or decision as it occurs in time.
2. *All outcomes must be represented.* Each final outcome must be represented as an endpoint on the right side of the tree.
3. *Several types of "nodes" may be used.* In general, a *node* represents a decision, an uncertain event, or an outcome. Each branch of the tree has an associated node located at the right hand end of the branch. A *decision* node (square) is used to indicate a decision facing the decision maker. A *chance* node (circle) is used to represent an event of uncertain outcome. A *terminal* node (triangle) is used to denote a final outcome: the end of a path, often referred to as a *scenario*. All of the nodes at the right edge of the tree must be terminal nodes.
4. *Branches emanating from a decision node represent the options*. All available choices must be represented, and the choices must be designated in a way that none overlap.
5. *Branches emanating from a chance node represent the possible outcomes of the event.* All possible outcomes must be represented, and the outcomes must be designated in a way that none overlap.

In accordance with these guidelines, let's design a decision tree that represents the investment problem posed at the end of Chapter 1.

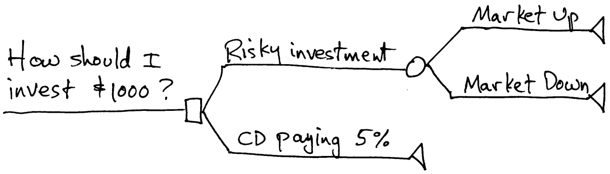
First, you must decide where to invest your money. This decision is represented by a decision node:



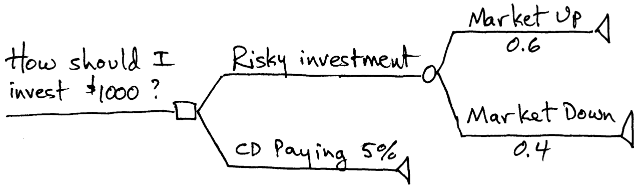
The branches of the decision node must represent all available options. The options are *Risky investment* and *CD paying 5%*. The branch *CD paying 5%* is a final outcome (neither an uncertainty nor a new decision), so it is represented by a triangular terminal node. However, the option *Risky investment* requires consideration of an uncertain event: which way the market will go. So *Risky investment* is followed in your tree by a circular chance node.



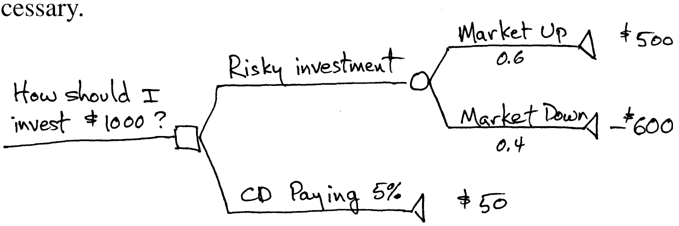
The branches of *Risky Investment* must represent all possible outcomes. In this example, the outcomes are *Market up* and *Market down.* They are both final outcomes, so they are represented by terminal nodes.



Now the structure of the tree is complete. All that remains is to place the values in the tree. There are two types of values: probabilities and payoffs. Probabilities are assigned to the branches emanating from each chance node, and payoffs are assigned at every terminal node. The probabilities at the branches emanating from a chance node must sum to 1.0 (100%), as they are *conditional* probabilities. That is, a probability is assigned to a particular chance outcome under the assumption that the events to its left have already occurred. Probabilities are drawn below the branch line of the event they represent.



Payoffs are drawn to the right of the terminal node. Note that a payoff value is assigned to a terminal outcome under the assumption that the outcome is reached; hence, no consideration of probability values is necessary.



Look at the tree above to be sure you understand it. The decision node *How should I invest $1,000?* is at the root of the tree. You have two choices at that node: *Risky investment* and *CD paying 5%*. The former is a risky event (you can't be sure what will happen to your money), so it is a chance node; the latter involves no uncertainty, so it is represented as a final outcome. The branches of the chance node represent the two possible scenarios as they unfold. Each final outcome has a "payoff" value associated with it.

To calculate a decision tree, one works backward, from right to left. Thus, calculating a tree is often referred to as "folding back" or "rolling back" the tree. The value of each node is determined as follows:

- The value of a decision node is equal to the value of its best option.
- The value of a terminal node is equal to the value of its payoff.
- The value of a chance node is equal to its *expected value*, which is found by weighting the values of each of its branches by their respective probabilities.

Applying these rules to the tree, the value of each of the three terminal nodes is already displayed. Now you work leftward from the rightmost branches, to the node *Risky investment*. You can find its expected value by the following calculation:

expected value of *Risky investment* = (500 * 0.6) + (-600 * 0.4)

where 500 is the value (payoff) of *Market up*, and 0.6 is its probability; similarly, -600 is the value (payoff) of *Market down*, and 0.4 is its probability.

Finishing up the calculation, you have

expected value of *Risky investment* = 300 - 240

expected value of *Risky investment* = 60.

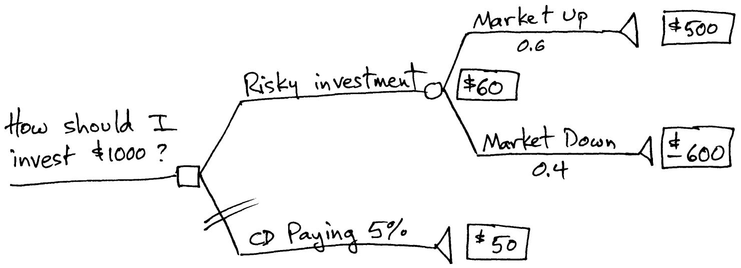
Note that this expression does *not* indicate that if you buy stock you will earn $60, which is clearly untrue, since the model allows for only a gain of $500 or a loss of $600. It simply means that $60 would be your average profit if you were to make the same investment many, many times. This distinction is critical to your understanding of decision analysis.

> *The expected value of an uncertainty is a probabilistic calculation, making it possible to compare one uncertainty with another, or an uncertainty against a certain outcome.*

Let's continue to calculate the tree. All that remains is to calculate the value of the root decision node, and to decide which option to take. As indicated above, the expected value of a decision node is equal to the value of its best option. The value of *CD paying 5%* is $50, and the value of *Risky Investment* is $60. By this calculation, the better option is to buy stock, so the value of the node *How should I invest $1,000?* is $60 also.

Again, be sure you understand the meaning of the $60 expected value calculation. It does not mean that if you follow the recommended strategy, you will earn $60. It means that the *expected value* (which is a mathematical construct, not necessarily a possible outcome) of the investment is $60 if you follow the recommended strategy.

Below is a version of your tree with these values included. When the tree is rolled back, the value (or expected value) of each node is typically drawn in a box to the right of the node.



Influence diagrams

Influence diagrams tend to be simpler on their face than decision trees. While they are less effective at presenting all of the underlying facts at once, they portray more clearly the factors that influence a decision, and how those factors are related.

Even in complex problems, where the decision tree is far too large to fit on a single printed page, the associated influence diagram is almost certain to be small enough for simple reproduction and efficient communication. Moreover, influence diagrams can make it easier to undertake certain calculations, such as Bayes' revision and expected value of perfect information, which depend on considerations of influence.

TIP: See Chapter 31 for more information on the advantages of influence diagrams.

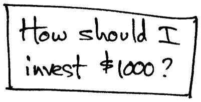
The design of an influence diagram is subject to guidelines:

1. *All factors influencing the decision must be represented.* Each relevant factor or variable influencing the decision should be represented by a node in the diagram.
2. *Several types of nodes may be used.* In general, a *node* represents a decision, a variable, or an objective. A *decision* node (square) is used to indicate a decision facing the decision maker. A *chance* node (circle) is used to represent a variable (or event) whose value (or outcome) is uncertain. A *value* node (diamond) denotes a quantity that measures the desirability of any final outcome.
3. *Nodes are connected by arcs.* An arc drawn from one node to another indicates (a) that the first node influences (conditions) the second node, and/or (b) timing. (See below for more on influence.)
4. *Time flows along the arc lines.* In general, a node which is conditioned on the outcome of another node must occur later in time. There are a few important caveats to this rule which are discussed in Chapter 31. It is important to note that while arcs generally indicate timing, their primary purpose is the representation of influence.

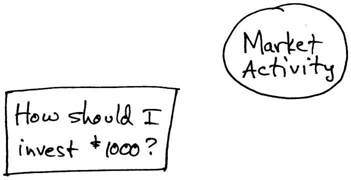
The presence of influence is, as mentioned above, represented by arcs. If, for instance, the probabilities associated with an uncertain event will differ depending on a prior decision or chance outcome, there will be an arc indicating this influence in the diagram. The meaning of arcs and the influences they represent will be discussed more thoroughly later.

In accordance with these guidelines, let's design an influence diagram that, like the decision tree constructed earlier in this chapter, represents the investment problem posed at the end of Chapter 1.

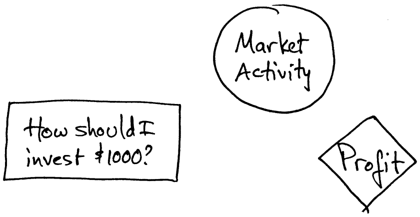
The first step is to draw a node that represents the decision.



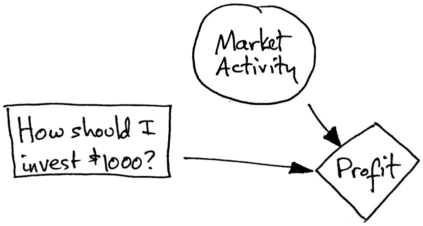
Next, you must consider what uncertainties may influence the final outcome. In this example, there is only one: the activity of the market. Thus, a chance node is added.



The objective by which success is to be measured is the return on the investment, or profit. Thus, a value node, called *Profit*, is added:



Next, we draw in the arcs that denote the influences between nodes. Profit will be affected by your investment decision and by the activity of the market, so two arcs must be drawn.



While the influence diagram is now complete on its face, there is a substantial amount of information missing. For example, the different market states, the probabilities and values associated with those states, and even the specific decision alternatives available to you are not pictured. You will learn to enter these data in the tutorial on influence diagrams in Chapter 4. For the moment, however, it is sufficient to note that the diagram reflects the contours of the decision problem.

CHAPTER 3 BUILDING YOUR MODEL AS A DECISION TREE

Now that you have had a very basic, whirlwind tour of decision analysis, it is time to learn how to use DATA to build decision trees. In this example, you will build the investment tree described in Chapter 2.

TIP: In this manual, the ">" symbol indicates a menu selection. Thus, the text "Options > Add Branches" refers to the Add Branches command in the Options menu. Also, note that words to be typed in a text field are shown in Courier typeface (e.g., "`Node Namë`).

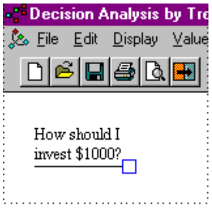
Naming nodes and adding branches

When you start DATA, you are presented with a new tree, which has a single decision node. You can tell that the node is selected because the node symbol (in this case, a square) is filled in. Click elsewhere in the window to deselect the node. To reselect the node, click in its symbol.

► **To enter a node name:**

▷ Select the root node (click once on the square decision node symbol or above the line to its left).

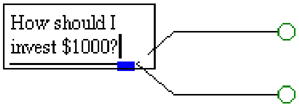
▷ Type `How should I invest $1000?` in the editor box. Optionally, you may want to press RETURN after the word “I” to enter a multi-line node name.



► **To add branches:**

▷ Select the root node.

▷ Choose Options > Add Branches.



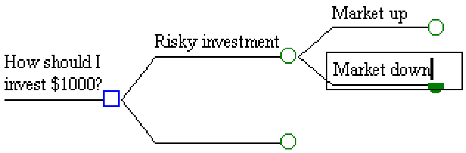
If you wanted to add more than two branches, you could again choose Options > Add Branch (with the root node still selected). The Add Branch command adds only one branch at a time after the initial two.

Another method for adding branches to a node is to double-click on its symbol. When the pointer is over the node's symbol, the cursor will change to a branch cursor to indicate that double-clicking will add branches.

It is also possible to change which node is selected by using the arrow keys to move from one node to another. Thus, to select the top chance node, you can either click on its node symbol (or in the area above the branch line), or you can press the right arrow key while the root node is selected. DATA will deselect the decision node and select the top chance node. Try using the arrow keys to get a feel for this technique.

If the arrow keys do not change the selected node, you may need to depress the Navigation button on the tool bar. See Appendix A for information on changing cursor behavior with the Navigation button.

- ▷ Select the top chance node, and type in its name, `Risky Investment`.
- ▷ Add two branches to this node, and name them `Market up` and `Market down`, so your tree looks like this:
- ▷ Select the node below Risky Investment and name it `CD paying 5%`.

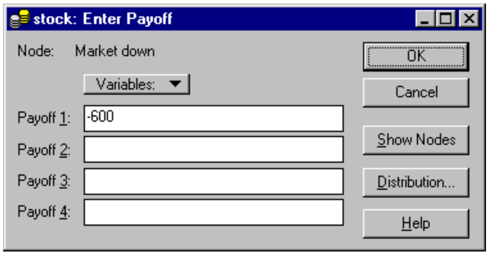


Entering payoff values

Note that each new node you created was a chance node. DATA allows you to change a node to another type (decision, terminal, etc.) by using the Change Node Type command.

► **To enter payoffs:**

- ▷ Select the node Market down.
- ▷ Select Options > Change Node Type.
- ▷ Click on the Terminal button, and press ENTER (Windows) or RETURN (Macintosh).



Node: Market down

Payoff 1: -600

Payoff 2:

Payoff 3:

Payoff 4:

▷ A window for entering the payoff value will open. Type `-600` for payoff 1, and press ENTER or RETURN.

At each terminal node it is possible to enter up to four payoffs (or attributes). The term payoff is used to denote the net value to the decision maker of a specific scenario. Chapters 19–21 cover the use of multiple payoffs in order to calculate the tree using multiple criteria, such as cost and effectiveness.

For now, only payoff 1 will be used. Follow the same procedure to enter payoffs for the nodes Market up (payoff = 500) and CD paying 5% (payoff = 50).

TIP: To change the payoff expression for an existing terminal node, simply select the node and then choose Values > Change Payoff, or double-click on the terminal node symbol.

Entering probabilities

Now, the probabilities must be entered for each potential outcome of Risky Investment. To enter a probability for a node's branches, click below the appropriate branch line. Or, when you are editing a node's name, press TAB to switch to the probability field below the line. Pressing TAB again will switch back to the node's name.

► **To enter probabilities:**

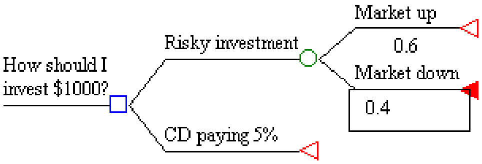
▷ Select the Market up node.

▷ Press TAB to move to the probability field.

▷ Type `0.6`.

▷ Press TAB to return the text editor to the node's name.

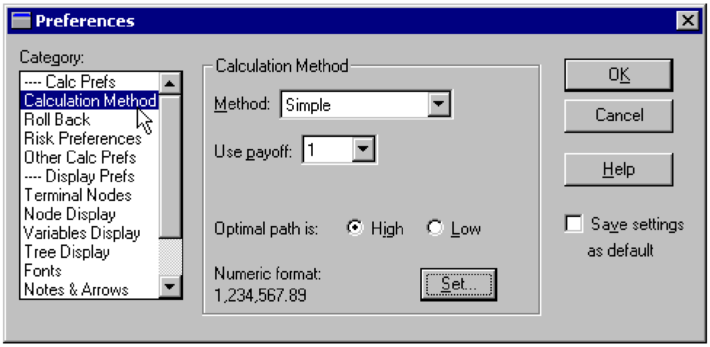
Follow the same procedure to enter a probability of 0.4 for Market down.



Setting calculation preferences

One last detail before you calculate the tree. You must tell DATA that you would like to *maximize* the expected value in this tree. Some trees should be set so that the optimal path is based on minimizing expected value, such as those where cost, rather than revenue, is the value of each node.

- ▷ Choose Edit > Preferences, and select the Calculation Method page. Be sure that the method is Simple, payoff 1 is selected in the list box, and optimal path is set to high.



Preferences

Category:
---- Calc Prefs
Calculation Method
Roll Back
Risk Preferences
Other Calc Prefs
---- Display Prefs
Terminal Nodes
Node Display
Variables Display
Tree Display
Fonts
Notes & Arrows

Calculation Method

Method: Simple

Use payoff: 1

Optimal path is: ● High ○ Low

Numeric format: 1,234,567.89

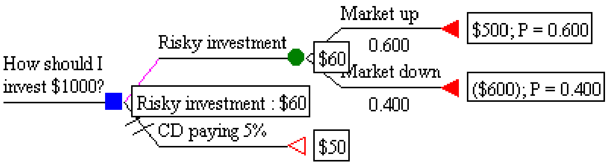
For now, ignore the rest of the Preferences dialog box.

- ▷ Press ENTER or RETURN when you are finished.
- ▷ Save the tree now by choosing File > Save. Give your tree the name Stock Tree.

Calculating the tree

► To calculate expected values for the tree:

- ▷ Choose Roll Back from the Analysis menu. Your rolled-back tree should look like this:



► To turn off roll back display:

- ▷ Pull down the Analysis menu. While the tree is rolled back, a check mark will appear next to the Analysis command. Choosing Analysis > Roll Back again will turn off roll back display, and allow you to make changes to your tree.

Many additional analysis features will be covered in Chapter 6 and in later sections of the manual.

Chapter 4 Building Your Model as an Influence Diagram

DATA's powerful influence diagram interface provides an alternative method of building decision analytic models. While you may find the lessons in this tutorial useful, you do not need to complete the tutorial in this chapter before proceeding to Chapter 5.

Using influence diagrams

An influence diagram is a very compact representation of a decision problem. Each uncertainty and decision is represented by a single node. Arrows (known as *arcs*) are then drawn between certain nodes to indicate potential influence, or conditioning. For example, if the probabilities of the outcomes of event C vary depending on decision D, an arc is drawn from D to C to indicate this influence.

Most influence diagrams fit on a single page, even if a corresponding decision tree has thousands of endpoints, making influence diagrams an excellent communications tool. Moreover, nontechnical decision makers (such as executives, sales people and, possibly, customers) may initially find influence diagrams easier to understand than decision trees.

Another benefit of influence diagrams is their explicit display of the relationships between events. In a decision tree, there is no immediate way to know that the outcome of one event influences the conditional probabilities of another event. This influence is spelled out in an influence diagram.

TIP: For more information on when you should (or should not) input your model as an influence diagram, rather than as a decision tree, see Chapter 31.

DATA extends the capabilities of the standard influence diagram by adding information *inside* the nodes and arcs. This information – including probabilities and payoff values – is used to convert the influence diagram into a fully configured tree, ready for analysis. In addition, it is possible to specify asymmetries in your model so that the resulting decision tree accurately represents the problem. These asymmetries, which exist in almost all decision problems, are difficult or impossible to specify using other software.

If, after completing the tutorials in the manual, you want to learn more about influence diagrams, a number of texts and articles are available. Two very popular texts – covering both influence diagrams and trees in generic decision making – are *Decision Making and Forecasting,* Marshall and Oliver (1995), McGraw-Hill, Inc.; and *Making Hard Decisions,* Clemen (1996), Wadsworth.

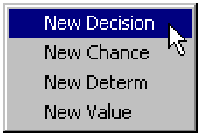
In this chapter, you will employ DATA's influence diagram functionality to model the investment decision described in Part I of the manual, the same problem that was modeled as a decision tree in Chapter 3.

Creating and naming nodes

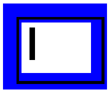
Open a new influence diagram window by choosing File > New..., clicking on Influence Diagram, and pressing ENTER or RETURN. Unlike new trees, which have a decision node and a blinking caret, a new influence diagram window is completely empty.

► **To create a node in an influence diagram:**

▷ In DATA for Windows, click in the influence diagram window with your right mouse button. In DATA for Macintosh, hold down the CONTROL key, and click in the influence diagram window.



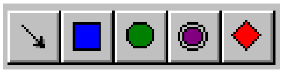
▷ From the pop-up menu that appears, choose New Decision.



You will see a new, selected decision node with a blinking caret. The black box and the caret indicate that you may begin typing to enter the node's name.

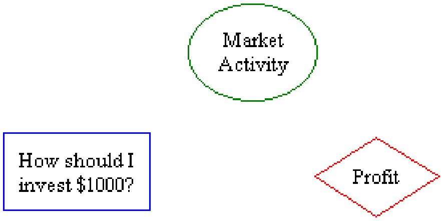
▷ Type `How should I invest $1000?` in the editor. Use the ENTER (RETURN) key to produce multi-line node names.

There is another way to create a new node. First, click on the toolbar button that represents the shape of node you want, then click anywhere in the influence diagram to place the node.



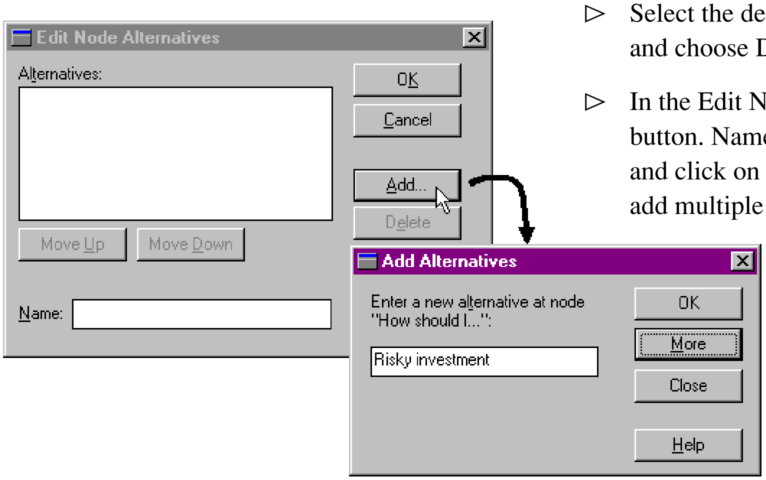
▷ Using either the toolbar buttons or the pop-up menu, create two more nodes: a chance node (circle) named Market Activity and a value node (diamond) named Profit. Do not connect them with arcs just yet.

Here is how your influence diagram should look:



Assigning alternatives and outcomes

All decision nodes must have an enumerated list of possible alternatives, and all chance nodes must have a list of possible outcomes. To enter this list, select the appropriate node and, from the Diagram menu, choose the appropriate menu item (either Alternatives, for decision nodes, or Outcomes, for chance nodes).



Enter a new alternative at node "How should I...":

Risky investment

- ▷ Select the decision node How should I invest $1000, and choose Diagram > Alternatives.
- ▷ In the Edit Node Alternatives dialog, press the Add button. Name the first alternative Risky investment and click on the More button. This allows you to add multiple alternatives without exiting the dialog.
- ▷ Type CD paying 5% to name the second alternative, and press ENTER or RETURN.
- ▷ Now, select the Market Activity node, and choose Diagram > Outcomes. Enter two outcomes, Market up and Market down, and close the dialog.

Value nodes do not have lists of outcomes or alternatives.

TIP: Instead of using the Diagram menu to enter Alternatives and Outcomes, it is possible to generate a pop-up menu from which you can choose the appropriate option. In DATA for Windows, simply right-click to open the pop-up menu. In DATA for Macintosh, you should CONTROL-click.

Identifying influences

There are three types of influence that can be indicated in a DATA influence diagram: probabilistic influence, value influence, and structural influence. An arc may reflect any combination of these influence types.

Three types of influence

A *probabilistic influence* exists between two nodes if the different possible outcomes at the first node require different enumerations of probabilities for the second. In a legal decision, for instance, the probability the defendant will be held liable will differ depending on whether or not the judge allows testimony on a certain issue. This type of influence may occur only at a chance node, although the conditioning node may be either a chance node or a decision node.

Value influence exists when the cost (or profit) of a node differs based on the outcome of another node. For instance, the cost associated with manufacturing a product may depend on the local availability of certain natural resources. Values are used to create a payoff formula in the value node. (This subject will be discussed later.) Value influence may occur at a decision, chance, or value node.

Value nodes do not condition other nodes; while value nodes may have arcs pointing towards them (to indicate value dependence), they do not have arcs pointing away from them. (An exception to this rule is covered in Chapter 31.)

The third type of influence, *structural influence*, is unique to DATA. It enables the creation of asymmetries and other indicators of tree structure within the influence diagram.

This threefold system means that the influence diagram is really three separate influence diagrams layered into one. Each arc may represent one or more types of influence, or none at all. (An arc with no specific influence is used to determine timing, as described in Chapter 31.)

Representing influence in the model

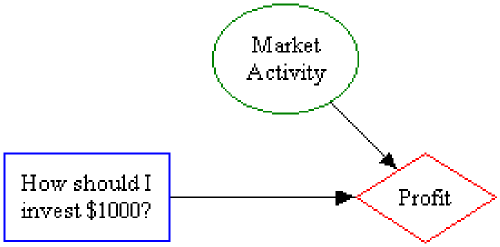
First, consider the existence of probabilistic influences. Since probabilities are entered only at chance nodes, only an arc leading into a chance node may indicate probabilistic influence. In this example, there is only one chance node: the node representing Market Activity.

The probabilities associated with Market Activity are not influenced by the investment decision. (The probability of each outcome remains the same, regardless of your investment decision.) Thus, there will not be any arcs in this diagram which indicate probabilistic influence.

To determine *value* influence, two questions should be asked: “What is the value or cost of the decision?” and “What is the value (or cost) of possible market activity?”

In this problem, both investment options have the same cost, and to keep the model very simple, the value of each outcome is being specified by the model builder, rather than through a formula to be calculated by DATA. Accordingly, under the particular circumstances of this model, there is no value influence to be represented by an arc from the decision node to the chance node. However, in a more sophisticated problem, where a formula is used to calculate the payoffs associated with the various scenarios, the choice of option would be likely to influence values associated with the market activity node.

The value of the final outcome node Profit is influenced, of course, by both your investment decision and the fluctuation of the market. Because of this influence, you should draw the two arcs shown below.



How to draw an arc

There are three ways to draw an arc.

► **To draw an arc using the pop-up menu:**

▷ Right-click (Windows) or CONTROL-click (Macintosh) on the influencing (or *conditioning*) node.

▷ Select Draw New Arc from the pop-up menu.

▷ Move your mouse to the influenced (or *conditioned*) node. You will see a dotted line following your mouse. Click (left-click) on the influenced node to complete the arc.

► **To draw an arc using the CONTROL or OPTION key:**

▷ While holding down the CONTROL key (Windows) or OPTION key (Macintosh), click normally on the conditioning node.

- ▷ Hold down the mouse button, and drag the cursor to the conditioned node. Releasing the mouse button while it is over the conditioned node will complete the arc.

▶ **To draw an arc using the tool bar:**

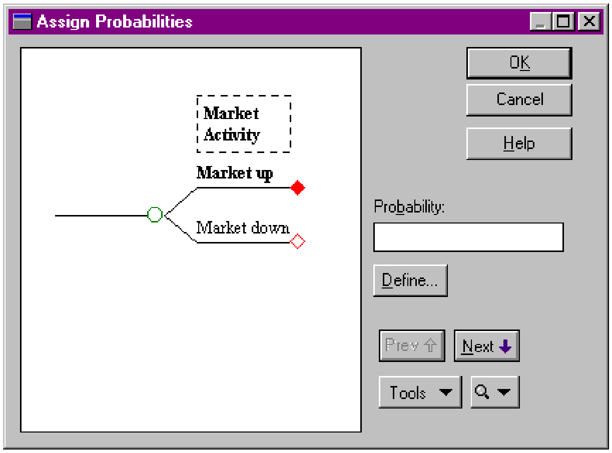
- ▷ Click on the tool bar button representing an arc, just to the left of the influence diagram node-type buttons.
- ▷ Click on the conditioning node and hold down the mouse button.
- ▷ Drag the mouse to the conditioned node, and release the mouse button.

Note that the second and third methods of arc creation require that you click and drag over the two nodes, rather than clicking twice.

If you have not done so already, draw the two arcs indicated in the picture above using any of the above methods.

Assigning values and probabilities

The windows for assigning values (e.g., costs) and probabilities are virtually identical. Here is the window shown when you select Probabilities from the right-click pop-up menu at the Market Activity node:



The pane on the left shows a representation of the *relevant portion* of your model in tree form. Any conditioning events will appear here, although in this example there are none. Each node in the "mini-tree" that requires your attention will have a red diamond to its right. Only one node may be selected at a time; the selected node will have its red diamond filled in and its name drawn in bold. You may select only those nodes that have a red diamond.

To select another (red-diamond) node, you may either click on it directly in the tree pane, or use the Prev and Next buttons. Or, you can use the CONTROL key with the keyboard's up and down arrow keys to move between nodes.

The button with the magnifying glass is a pop-up menu of zoom commands. In influence diagrams with considerable conditionality, zooming out of the tree pane may be very useful to see which conditional values are being assigned. Resizing the dialog box will also make it possible to see more of the tree.

The Tools button is a pop-up menu that enables you to assign probabilities using the probability wheel, or using a distribution. It is also possible to paste linked values from a spreadsheet, as described in Chapter 31.

For this example, we will use only the simplest case: that of numeric probabilities entered directly.

- ▷ When the dialog first appears, type `0.6`in the entry box.
- ▷ Click on the Next button (or click on the Market down node in the tree view).
- ▷ DATA will automatically assume that the probability of this node is #, which indicates the remainder probability. (See Chapter 14.) If this is not already entered in the probability field, type # now.
- ▷ Press ENTER or RETURN to store the probabilities.

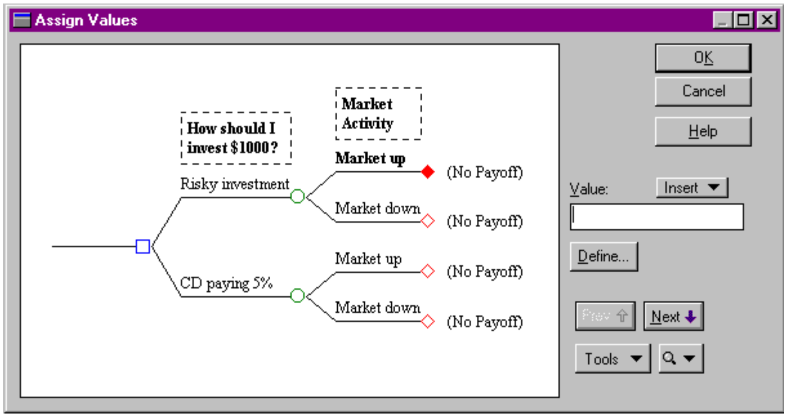
It is also possible to use variables and expressions (such as “1 - p”) for probabilities and values. This topic will be covered in Chapter 31.

Assigning values to Profit

Select the Profit node and choose Values from either the Diagram menu or the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu. The tree view which is shown in the dialog box appears on the next page.

Note that the conditioning events How should I invest and Market Activity both appear in the tree because of the arcs you have drawn.

However, the node CD paying 5% should be a terminal node in the tree, because the market activity has no relevance if you put your money into a CD.



It is now time to go back to create this asymmetry. Press ESC to close the Values dialog.

The basics of asymmetry

Phrased succinctly, the goal of this particular exercise is to reflect the following in the model:

> *If you decide to invest in the CD, the event "Market Activity" will be irrelevant, and its branches should not be drawn in the tree.*

There is clearly a relationship between the influence diagram nodes How should I invest and Market Activity. Of the three types of conditioning discussed above (probabilistic, value, and structural), the relationship is purely structural: it indicates asymmetry.

This type of influence is not recognized in the classical influence diagram. Since an important feature of DATA's influence diagram interface is the ability to convert to a decision tree, it is possible to specify structural information *inside the arc* connecting the two nodes.

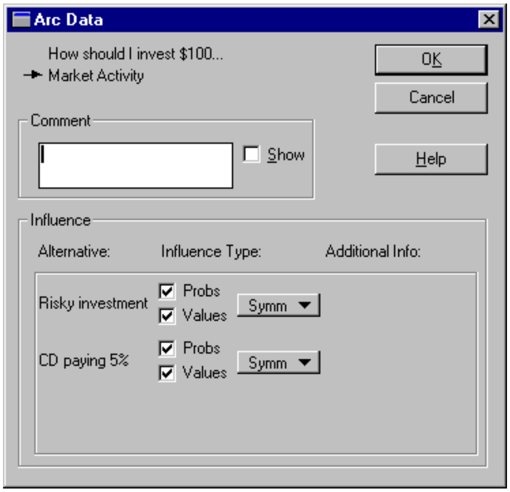
In DATA, you can assign up to three types of influence for each outcome or alternative of the conditioning node. Your decision regarding the investment alternatives under consideration will not influence the Market Activity node, either probabilistically or in terms of value. However, your choice of investment alternatives will affect the structure of the tree. This is handled through the use of an arc to represent structural asymmetry.

Entering the asymmetry

Create an arc pointing from How should I invest $1000 to the Market Activity node. Double-click the arc to see the dialog box shown at left.

At the top of the window you may enter a comment to be shown with this arc. When entering arcs which are only structural, it is often a good idea to include a comment. For now, though, turn your attention to the lower box, called Influence.

Note that each alternative of your decision has a separate group of influence types. Each alternative may have a probabilistic influence (the Probs check box), a value influence (the Values check box), and/or a structural influence (the pop-up menu, currently reading Symm).



How should I invest $100...
→ Market Activity

Comment

□ Show

Influence

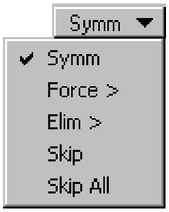
Alternative:	Influence Type:	Additional Info:
Risky investment	☑ Probs ☑ Values	Symm
CD paying 5%	☑ Probs ☑ Values	Symm

► **To enter the asymmetry for this arc:**

▷ Ensure that all four check boxes indicating probabilistic or value influence are cleared. This indicates that your decision has no numeric influence on the market activity.

▷ For the Risky investment alternative, leave the structure pop-up menu reading “Symm.” This indicates that if this alternative is selected, the branches of Market Activity should be drawn.

▷ For the CD paying 5% alternative, click the structure pop-up menu, and select Skip. This indicates that if you decide to invest in the CD, the Market Activity node should be skipped for structural and analytical purposes, since the value of the CD will be unaffected by market activity. Click OK.



If you click on the pop-up menu that reads Symm, the following choices will appear for structural influence: Symm, Force, Elim, Skip, and Skip All. The two structure influence types employed in this model are:

Symm Short for “symmetric,” this indicates that the tree should be as bushy as possible, with all branches drawn. Most of your influences will have this structure.

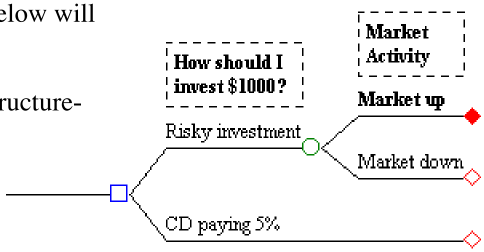
Skip This is the most common type of asymmetry. It indicates that when a particular outcome occurs (or alternative is chosen), branches associated with the conditioned node should not be drawn.

A description of the other structural influence types can be found in Chapter 31.

When an arc has no “real” numeric influence (i.e., probabilistic or value influence) and is used only to indicate asymmetry, it is drawn in dotted gray.

Assigning values, revisited

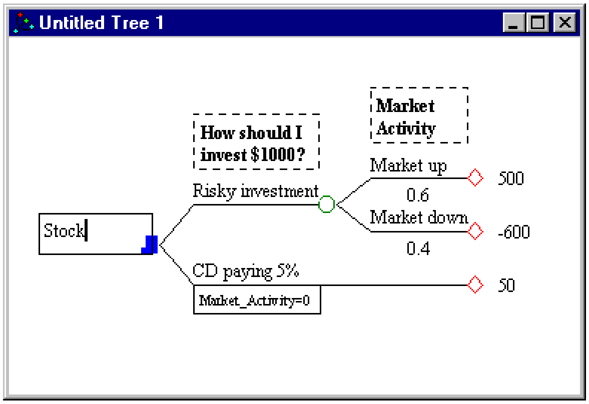
Select the Profit node and choose Values from the Diagram menu. The tree fragment illustrated below will be displayed.



With the addition of the structure-only influence, you have created a tree which has the desired asymmetry. Proper values for profit can now be assigned. Enter 500for Market up, -600for Market Down, and 50for the CD.

Viewing the converted tree

To convert an influence diagram into a tree, choose File > Convert to Tree. If no problems were found in your influence diagram, you will see the final tree.



Your tree may have a few extraneous definitions of variables. Their presence is explained in Chapter 31.

Incomplete influence diagrams will often convert properly. Values and probabilities may be left empty because these can be added later, in the tree window. However, the failure to draw arcs between nodes where influences exist (or the introduction of unnecessary arcs) can cause problems. Of course, you must assign alternatives and outcomes to all decision and chance nodes in your influence diagram before DATA can convert it into a tree containing the appropriate structure.

IMPORTANT! There is no "hot-link" between a converted tree and its related influence diagram. Changes made to the values or structure of the tree will not be automatically included in the influence diagram. Establishing dynamic linkages would require limiting tree modifications to those which can be specified in terms of influence, thereby making unavailable many of DATA's most flexible tree construction features. It is still possible, however, to re-establish a linkage between an influence diagram and a tree. To do so, simply make all structural and value changes in the influence diagram and then convert the modified influence diagram into a new tree.

CHAPTER 5 MAKING CHANGES TO YOUR TREE

If you have worked through the tutorial examples in the previous two chapters, there may be multiple documents open in DATA: the tree you created manually in Chapter 3; the influence diagram you created in Chapter 4; and the tree created by conversion of the influence diagram. Pull down the Window menu to see a list of currently open documents. The tutorial will continue with the Stock Tree, created in Chapter 3. Select it from the Window menu, or open the file if you have closed it.

TIP: DATA's File menu includes a list of recently opened trees, influence diagrams and graphs. Use it to reopen files quickly without having to use the File > Open dialog.

Using variables

DATA can evaluate a decision tree which contains only numeric point values; however, many of DATA's most valuable analytical features can be employed only if uncertain probability and payoff components are defined using variables rather than raw numbers. One of these features is sensitivity analysis, which assesses the extent to which the values and marginal values of your decision alternatives are affected by changes in a particular quantity (e.g. a discount rate or probability). In DATA, all forms of sensitivity analysis require that the quantities being varied are defined as variables.

In some cases, you may wish to design a model using variables from the outset. In others, you may prefer to design the tree using numbers only, and then substitute variables for those numbers after you have completed the model's structure.

Since the Stock Tree has already been structured and contains numeric payoffs, this section will take the latter course, substituting variables for some numeric quantities.

This section of the tutorial contains only a basic introduction to the use of variables in DATA. Chapters 8 and 9 provide a more comprehensive treatment of the subject. It is very important that you go on to read them, not only for help in building more sophisticated models but also to gain a better understanding of how DATA locates and applies the definitions assigned to variables. This background will help improve your modeling productivity and reduce the chance of error.

> **TIP:** The names given to variables must conform to certain rules. Multiple-word names are not allowed. In addition, the name must: (a) begin with a letter or underscore (_) character; (b) contain only letters, numbers, and underscore characters; and (c) be no longer than 32 characters.

Defining a probability as a variable

In many cases, you will be uncertain about the accuracy of probability estimates. Defining probabilities as variables will not only simplify making adjustments, but also will make it possible to perform analyses that assess the significance of this uncertainty.

In the investment tree, you defined the probability of Market up as 0.6. Assume that you wish to assess the impact on your decision of varying the probability between 0.4 and 0.8. Before DATA can perform a sensitivity analysis on this probability, it must be defined as a variable. Following is one method for doing so.

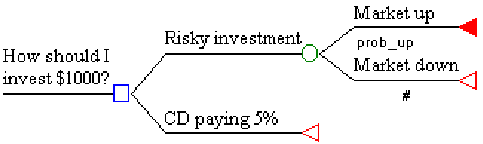
► **To assign a variable to a probability:**

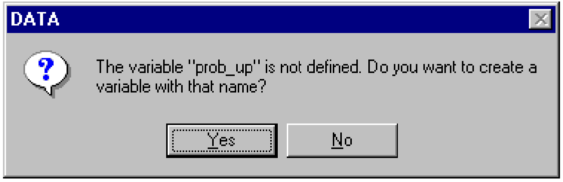
▷ Select the Market up node.

▷ Press the TAB key to access the probability field, and delete the probability expression 0.6 using the BACKSPACE or DELETE key.

▷ Type `prob_up` and press TAB to return to the node name field.

An alert tells you that prob_up was not recognized, because you have never used that name before. DATA needs to know that you want that name to be used as a variable, and that it is not a mistyped variable or function name.



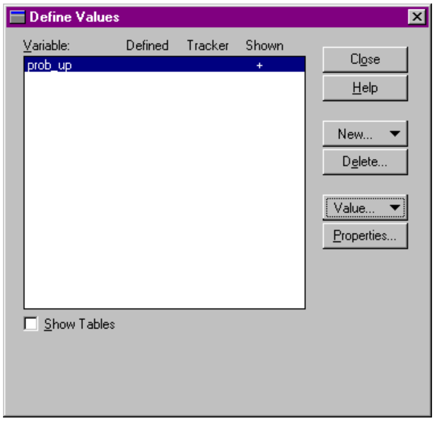

DATA

The variable "prob_up" is not defined. Do you want to create a variable with that name?

▷ To create the new variable, click Yes in the first dialog box.

▷ Click OK in the Properties dialog box that follows. You will learn how to use the Properties dialog in Chapter 9.

You have created a variable entitled prob_up and set the probability of the Market up node to the value of prob_up. Now, you must assign a numeric value to the new variable. In DATA, this step is referred to as *defining* the variable. In this instance, the numeric value to be assigned is 0.6, the probability that the market will rise.



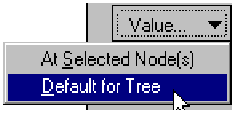
► To define a variable:

▷ Choose Values > Define Values....

The Define Values dialog box appears. This dialog box lists all of the variables created for the tree, as well as all available tables (see Chapter 26). So far, you have created only one variable, so prob_up is the only entry in the list.

▷ Click on the variable prob_up.

▷ Click on the Value... button and hold down the mouse button; a pop-up menu appears.



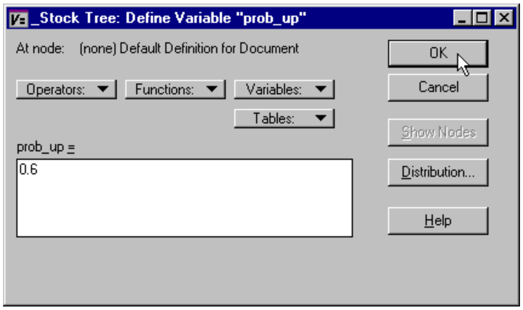
A variable definition can be assigned in either of two ways: as default for the entire tree, or at a particular node. If you define a variable default for the tree, that definition (a value or expression) applies to the entire tree, and its definition is stored at the root node. If you define a variable at a node other than the root node, that definition will be used only in the subtree rooted at the selected node.

Until you fully understand DATA's variable interface, you should make numeric variable definitions *default for the tree*. This will mean creating a separate variable for every uncertain quantity.

As explained in Chapter 8, definitions made *at selected nodes* are appropriate for quantities whose values depend on some decision or event, and are likely to differ at various points in the tree.

There is an important qualification to the above statements: a definition at one node (including the root node) can be *overridden* by a second definition at a node somewhere further to the right. The ramifications of defining a variable at more than one node are explored fully in Chapter 8.

Make the definition of the prob_up variable default for the tree (or at the root node). This value will be applied throughout the tree.



Stock Tree: Define Variable "prob_up"

At node: (none) Default Definition for Document

prob_up =

0.6

▷ From the Value... pop-up menu, choose Default for Tree.

▷ Type 0.6in the Define Variable window, and press ENTER or RETURN.

You have now defined prob_up as a default variable, with value 0.6. When DATA evaluates the probabilities of the branches of Risky Investment, it will use the definition of prob_up, 0.6, as the probability of the Market up node, and it will use the numeric 0.4 as the probability of the Market down node.

While this result is correct given the current definition of prob_up, a problem would arise if the definition of prob_up were changed to, for example, 0.7. In that event, the probabilities of Market up and Market down would sum to 1.1, which is impermissible. To prevent this occurrence, it is advisable to redefine the probability of Market down in terms of the prob_up variable.

Defining a probability as a variable expression

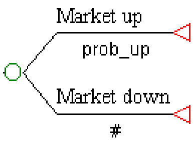
► **To assign a variable expression to a probability:**

▷ Select the probability of the Market down node, 0.4.

▷ Overwrite the probability by typing `1-prob_up` .

Now, the definition process is complete. If the value of prob_up should change to 0.7, the probability of Market down will automatically be calculated as 1 - 0.7, or 0.3, so that the probabilities sum to 1.0.

DATA can further simplify the calculation of this complementary probability expression:



▷ Select the probability expression of the Market down node.

▷ Overwrite the expression by typing #.

During calculations, the # character will automatically cause DATA to calculate 1.0 minus the sum of the probabilities at the other branch or branches. The # character can be used at multiple chance nodes; however, at a given chance node, it can be assigned to only a single branch.

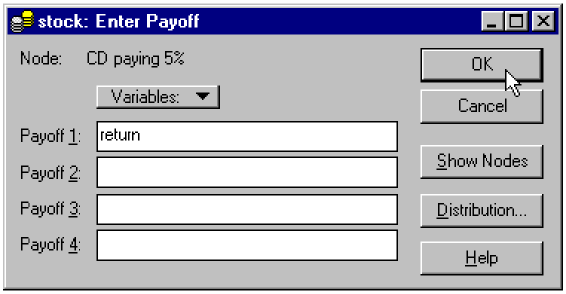
Save the tree by choosing File > Save or by clicking on the diskette icon in the tool bar. You will make use of your new variables when you perform a sensitivity analysis in Chapter 6.

Defining a payoff as a variable or expression

By defining payoffs as variables, you can gain many of the same benefits available from defining probabilities as variables. By using variables, payoffs can be changed more easily and analyzed more fully.

► **To assign a variable to a payoff:**

- ▷ Select the CD paying 5% node and choose Values > Change Payoff….
- ▷ In place of 50, type `return` and press ENTER (Windows) or RETURN (Macintosh).



stock: Enter Payoff

Node: CD paying 5%

Payoff 1: return

Payoff 2:

Payoff 3:

Payoff 4:

- ▷ Accept DATA's suggestion that you create a variable named return, and click OK in the Properties dialog.

You have now assigned the variable return to the payoff of the CD paying 5% node. As with the probability variable above, you must define (assign values to) this payoff variable.

Defining a variable using other variables

Many payoffs consist of, or are derived from, several different quantities. A payoff measuring profit may contain revenue elements, expense elements, a discount rate, and other components. In such cases, it is helpful to define the payoff as an expression that includes multiple quantities. This process renders payoffs less opaque and also facilitates those analyses, mentioned above, that require quantities to be expressed as variables.

In the investment tree, the $50 profit from the CD is calculated by applying a 5% interest rate to a principal amount of $1,000. You could define the return variable as a numeric value or expression: 50, or 1000 * .05; it would also be possible simply to type this formula into each payoff box, and not use the return variable at all. There are advantages, however, to using a variable in a payoff and defining it using a formula.

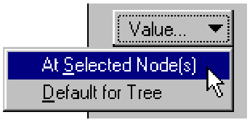
In this case, you could define the return variable using two additional variables, principal and rate. If you decided later to add another element, such as tax, to the formula, it would be much easier to make a single change to the default definition of return, than it would be to repeat the change in every terminal node's payoff formula.

The principal and rate quantities represent the finest level of detail in calculating the payoff of the certificate of deposit; defining these quantities as variables will allow you to easily change their values and perform sensitivity analysis. You will now define the return variable with an expression that includes these quantities.

The formula definition of return will be located at the CD paying 5% node, although it would be equally correct to assign this definition at the root node. Assigning it at the CD paying 5% node will facilitate the use of different versions of the return formula in future iterations of the model which embody other, more complex investment scenarios.

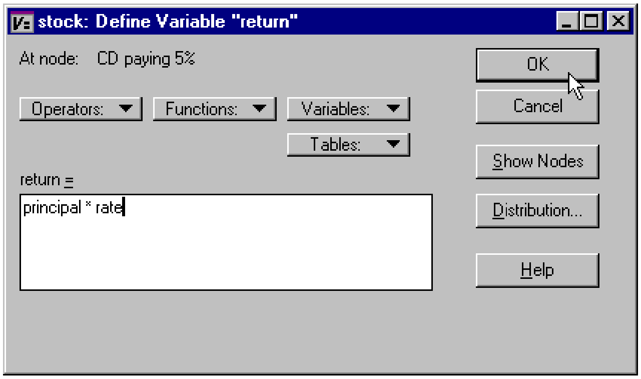
► To define a variable as an expression:

▷ Select the CD paying 5% node, and choose Values > Define Values... or click on the single **V=** icon in the toolbar.



▷ Select the variable named return. Click on the Value... button and choose At Selected Node(s) from the pop-up menu.

▷ In the Define Variable window, type `principal*rate` and press ENTER (Windows) or RETURN (Macintosh).


stock: Define Variable "return"

At node: CD paying 5%

return = principal * rate

▷ For both the principal and rate variables, accept DATA's suggestion that you create variables, and click OK in the Properties dialogs.

The new variables should now appear in the Define Values dialog.

- ▷ Choose Values > Define Values..., or click on the single **V=** icon in the toolbar.
- ▷ Select the variable named principal. Hold down the CONTROL key and select the rate variable as well.

The numeric definitions of principal and rate should be made at the root node.

- ▷ Click on the Value… button and select Default for Tree from the pop-up menu.
- ▷ In the Define Variable window for rate, type `5%` or `.05`and press ENTER or RETURN.
- ▷ In the Define Variable window for principal, type `$1,000` and press ENTER or RETURN.

You have now defined return at the CD paying 5% node equal to principal * rate, and principal and rate at the root node as $1,000 and 5%, respectively. When evaluated, the payoff of the CD paying 5% node will be calculated as 1000 * .05, or 50. In the future, changes to the formula definition of return should be made at the CD paying 5% node, while changes to the principal and rate components should be made default for tree (at the root node).

- ▷ Save the tree before continuing the tutorial.

The same methods demonstrated in the payoff variable exercise can be used to collapse a complex expression used in a probability field. Thus, a single variable can take the place of any valid expression which uses a combination of variables, functions, and operators. Chapter 8 discusses some additional issues you should consider when building these complex expressions.

One definition of a variable can be overridden by a second definition of the same variable at a node to the right of the node where the first definition appears. For purposes of determining the definition in force at a given node, definitions made default for the tree are deemed to reside at the root node.

You might, for example, have a tree where a variable utilized in a payoff expression is the same for all outcomes except one. In such a situation, the component variable can be given a default definition for the tree and then be redefined at the exceptional terminal node. This is an important principle, and there are significant ramifications, which are explored in Chapter 8.

Cut, copy, paste, and clear

Frequently, nodes or subtrees that you create in one part of a tree can be utilized in another part as well. Rather than forcing you to recreate subtrees manually each time you wish to add them, DATA makes it possible to select an entire subtree, copy it to a clipboard, and paste it at one or more nodes. DATA also makes it possible to remove a subtree and, if you wish, reinstate it elsewhere.

> **TIP:** In DATA, a subtree begins with the branches emanating from a node. The root of the subtree and the definitions of variables located at that node are not included as part of the subtree. For instance, it is not possible to select an entire tree, including the root node and any variable definitions stored there, and copy and paste this tree into another tree.

For illustrative purposes, you will create a third investment option, Blue Chip Stock.

▷ Double-click on the root node of the investment tree to create a new branch.

▷ Select the new node and name it `Blue chip stock` .

The activity of the market can be expected to affect blue chip stocks as well as the riskier stock, although probably to a lesser degree. Thus, the subtree that consists of the nodes Market up and Market down is also applicable to the Blue chip stock node.

Copying and pasting subtrees

► **To copy a subtree:**

▷ Click on the Risky investment node, and choose Options > Select Subtree.

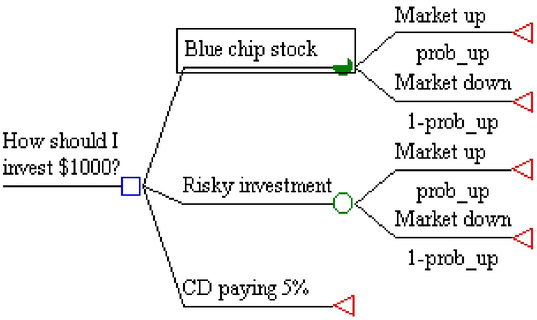
▷ Choose Edit > Copy Subtree.

Note that the Edit menu contains four tree clipboards, of which only one (currently Tree Clipboard 1) is selected. DATA's maintenance of multiple tree clipboards enables you to retain several subtrees at once, each on its own clipboard, to be pasted as needed.

► **To paste a subtree:**

▷ Click on the Blue chip stock node.

▷ Choose Edit > Paste Subtree.



The market activity subtree has been duplicated at the Blue chip stock node.

The duplicated copy is now independent of the original. Changes you make in one will not be automatically reflected in the other. See Chapter 12 for a discussion of clones, which do automatically update.

While the probabilities of Market up and Market down are not likely to be affected by your choice of which stock to buy, the payoffs associated with Market up and Market down will probably be different in the Blue chip stock subtree than in the Risky investment subtree, because Blue chip stock is (presumably) less volatile than Risky investment. Thus, you must select those two nodes and change their payoffs using the Values > Change Payoff… command.

▷ Select the Market up branch of the new Blue chip stock node.

▷ Select Values > Change Payoff…, type `200` and press ENTER or RETURN.

▷ Select the Market down branch of the new Blue chip stock node.

▷ Select Values > Change Payoff…, type `-160` and press ENTER or RETURN.

The payoffs you have entered reflect the lower volatility of the blue chip stock. Use the File > Save As… command to save the expanded tree under the name Two Stock Tree.

Cutting and clearing subtrees

To restore the original state of the tree, you will use the cut and clear functions to eliminate this subtree from the model.

► **To clear a subtree:**

▷ Click on the Blue chip stock node and select Options > Select Subtree.

▷ Choose Edit > Clear Subtree.

The subtree has been deleted without being placed on the clipboard.

The Blue Chip Stock node remains. This is an important property of subtrees: *A subtree begins to the right of its root node*. As mentioned above, when you cut or copy a subtree, the root of the subtree is not included.

Before using the Cut function to eliminate the Blue Chip Stock node, change the current clipboard by choosing Edit > Tree Clipboard 2. When you pull down the Edit menu again, you will see that the check mark has moved to Tree Clipboard 2.

► **To cut a node:**

▷ Click on the Blue chip stock node and select Edit > Cut Node.

The structure of the original investment tree has been restored. Pulling down the Edit menu, you will see that Tree Clipboard 1 contains a subtree (the subtree you copied from the Risky investment node, and Tree Clipboard 2 contains a node (the Blue Chip Stock node). Choosing Edit > Show Tree Clipboard will display the contents of the currently selected tree clipboard.

At any given time, the Cut, Copy, and Clear menu commands will reflect the element selected (a node, a subtree, text, or some combination). Depending on the contents of the active clipboard, the Paste menu command will read Paste Node, Paste Subtree, or Paste… (if both the active Tree Clipboard and the text clipboard are full), as appropriate.

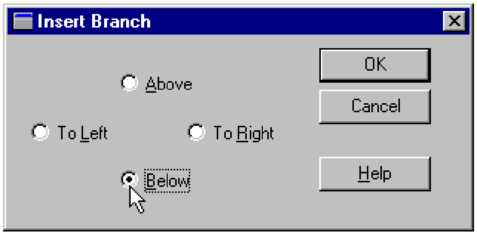
▷ Close the tree by choosing File > Close. Select No in the ensuing dialog box, to ensure that the last changes are not saved; this file should continue to contain the Blue Chip Stock alternative.

Inserting, deleting, and reordering branches

It is often desirable to insert or delete a branch in the middle of a tree, or to change the order of branches emanating from a single node. These results can be achieved through clever use of the Cut, Copy, and Paste commands (see below), but several shortcuts are available.

Inserting Branches

By double-clicking on a node, you can add branches. The Insert Branch command provides more options: you can add a single branch above, below, to the left of, or to the right of, the selected node. This provides a great deal more control over how new branches are added to a model.



Insert Branch

○ Above

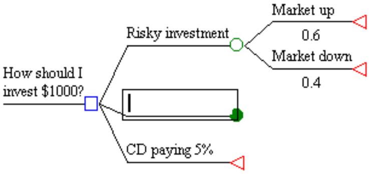
○ To Left ○ To Right

● Below

▷ Open Stock Tree and select the Risky investment node.

▷ Select Options > Insert Branch.

▷ Click on the Below button and press ENTER or RETURN.



A new decision option has been added below the Risky investment node. You could use this branch to model a new investment vehicle.

▷ Reselect the Risky investment node, choose Options > Insert Branch, and select To Right.

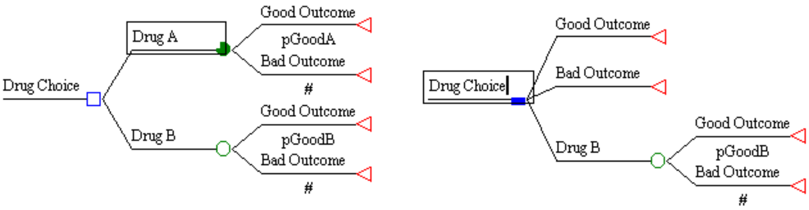
A new node has been added between the Risky investment node and its children. You could use this branch to model an intervening decision or uncertainty that would arise after choosing to make the risky investment.

Close the tree without saving.

Deleting branches

Delete Branch works on any branch, even one in the middle of a tree.

▷ Open a tree, and select a node that has at least two branches.



▷ Select Options > Delete Branch.

The branch ending in the selected node has been deleted. The children of the deleted branch (node) move up a generation and join any siblings of the deleted branch. Be sure to check the impact of this on the design of your model. As demonstrated in the above example, DATA will mechanically execute the Delete Branch command without testing the coherence of the resulting model.

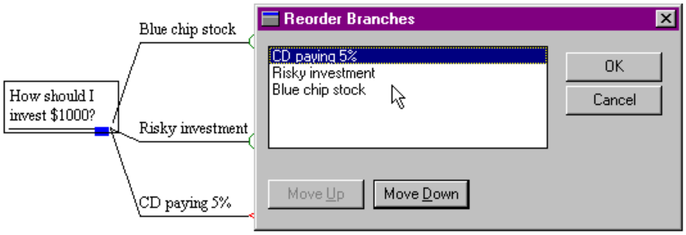
Reordering branches

The Reorder Branches command enables you to change the vertical sequence in which branches of the selected node appear.

▷ Open the Two Stock Tree and select the root node.

▷ Select Options > Reorder Branches.

The ensuing dialog box lists the branches of the selected node in the order, top to bottom, in which they appear.

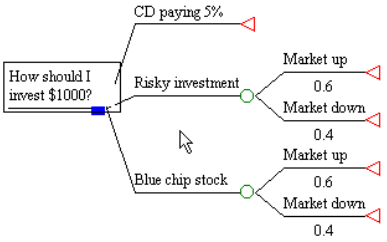


▷ With the Blue chip stock label selected in the list box, click Move Down twice.

▷ Select the CD paying 5% label, and click Move Up once.

▷ Press ENTER or RETURN.

You have reversed the order of the decision options emanating from the root node; CD paying 5% is now the uppermost option, and Blue Chip Stock is the lowermost option.



You need not save these changes in Two Stock Tree.

CHAPTER 6 ANALYZING YOUR MODEL

It is time to use DATA to automate the calculations that were done manually in Chapter 2. If you are unfamiliar with, or need to review, the concept of expected value, refer to pp. 7–8 and 11–12.

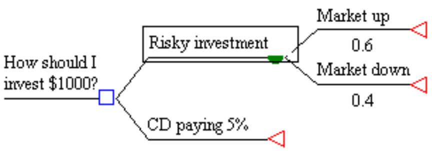
Open the original Stock Tree using the File > Open... command.

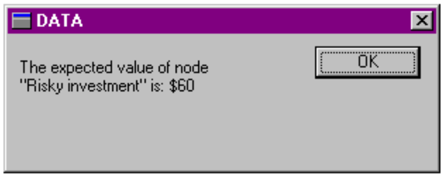
Expected value

► **To calculate the expected value of a node:**

▷ Click on the node Risky investment.

▷ Select Analysis > Expected Value.





The expected value of node "Risky investment" is: $60

A dialog box informs you that the expected value of the node Risky investment is $60. As described in Chapter 2, this value reflects the probabilities and payoffs of both the Market up node and the Market down node.

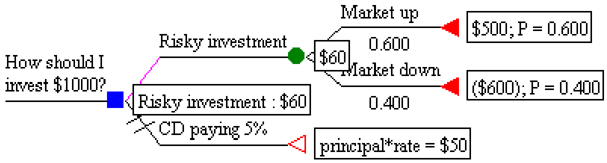
It is possible to perform an expected value calculation at any node. If the expected value calculation is performed at a terminal node, the resulting "expected value" is simply the payoff for the scenario ending at that terminal node; since no uncertainty is involved in this calculation, this is not a probabilistic value. The expected value of a decision node is the expected value of the best option emanating from it. The expected value of a chance node, such as Risky investment, is calculated as described in Chapter 2.

Rather than calculating the expected value of each node in the tree individually, it is possible to calculate and display the expected values and probabilities of all nodes simultaneously. The analysis option that achieves this is called Roll Back.

Roll back

► **To roll back the tree:**

▷ Select Analysis > Roll Back.



To the right of each node is a box that contains its expected value. The box to the right of the root decision node contains the name, Risky investment, and the expected value, $60, of the preferred option (or “optimal path”) emanating from that decision node. The box to the right of the CD paying 5% node contains the payoff, principal * rate, and its value, $50. The box to the right of the Risky investment node contains its expected value, $60. This is the same value that was calculated on the previous page using the Expected Value command.

Finally, the boxes to the right of the Market up and Market down nodes contain their expected values ($500 and -$600, respectively) and their path probabilities (0.600 and 0.400, respectively). The *path probability* of a node is the product of its probability and all of the probabilities to its left. Since there are no probabilities to the left of these nodes, their path probabilities are the same as their individual (or *conditional*) probabilities, 0.600 and 0.400.

By default, numbers representing expected values are displayed with currency symbols and without decimal places. See Chapter 10 for a description of how to change the tree’s numeric formatting settings.

The box to the right of Risky investment may overlap the text of its branches. Whenever an expected value box covers the text of a branch description or probability, the problem can be corrected by moving the expected value box.

► **To move an expected value box:**

▷ While pressing the CONTROL (Windows) or OPTION (Macintosh) key, click in the expected value box, and hold down the mouse button.

▷ Drag the box to a better location; release the mouse button.

While the tree is rolled back, most commands used to analyze or modify the tree are unavailable. The command used to roll back the tree acts as a toggle and will cancel the roll back when selected a second time.

► **To turn off roll back:**

▷ Select Analysis > Roll Back.

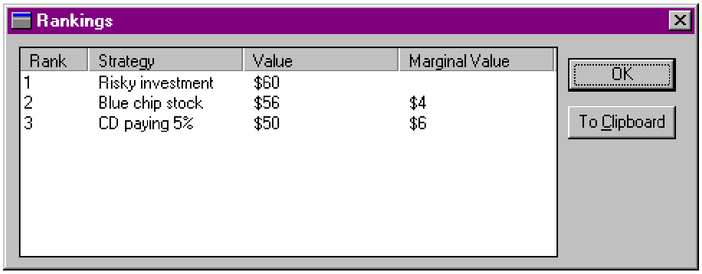
Rankings

The Rankings analysis offers a concise tabular report that identifies and ranks the alternatives available at a specified decision node. The ranking is performed in terms of the currently active calculation preferences. For example:

▷ Open Two Stock Tree.

▷ Select the root node by clicking on it.

▷ Choose Analysis > Rankings.



Rank	Strategy	Value	Marginal Value
1	Risky investment	$60	
2	Blue chip stock	$56	$4
3	CD paying 5%	$50	$6

A dialog box appears which ranks the options, specifies the expected value of each option, and in the case of suboptimal options specifies the marginal value (the amount by which the next most optimal decision dominates). Columns can be resized by clicking and dragging on the dividers between column headings.

Assessing risk

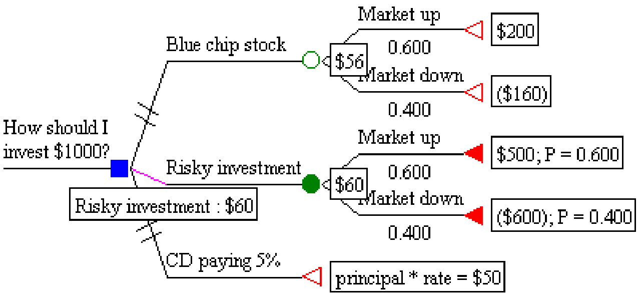
In addition to comparing expected values, DATA offers several options for analyzing risk. The most fundamental options are standard deviation and probability distribution (or risk profile).

To explore these options, you will use the Two Stock Tree created in Chapter 5. Select it from the Window menu or, if it is no longer open, use the File > Open... command.

Standard deviation

In this tree, you have three options: a blue chip stock, a volatile stock, and a CD.

▷ Roll back the tree to see how the expected values of the three options compare.



Both stock alternatives are more valuable than the CD. How do you decide which alternative to choose?

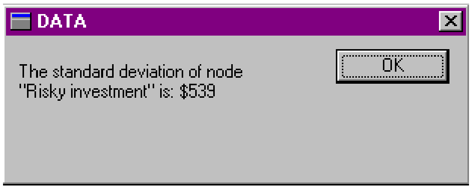
It certainly would be valuable to be able to compare the extent to which the actual payoffs are dispersed around the expected value for each of the alternatives. The standard deviation, a widely used measure in statistics (you can consult any basic statistics textbook for a description), is one such means of comparison.

► **To calculate a standard deviation:**

▷ Select the Risky investment node.

▷ Choose Analysis > Standard Deviation.

The dialog box indicates that the standard deviation of the possible results of buying the risky stock is $539.

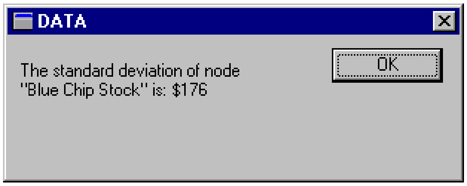


The standard deviation of node
"Risky investment" is: $539

▷ Select the Blue chip stock node.

▷ Choose Analysis > Standard Deviation.

The dialog box indicates that the standard deviation of the possible results of buying the blue chip stock is $176. The narrower standard deviation reflects the lower risk associated with the less volatile stock. Thus, if you were risk averse, you might choose to invest in the blue chip stock even though its expected value, $56, is lower than that of the more volatile stock, $60.



The standard deviation of node
"Blue Chip Stock" is: $176

The Standard Deviation command is available only when you have selected a chance node.

TIP: It is possible to quantify your risk preference and have DATA use it to adjust expected value calculations automatically. To learn how DATA handles risk preference curves, see Chapter 30.

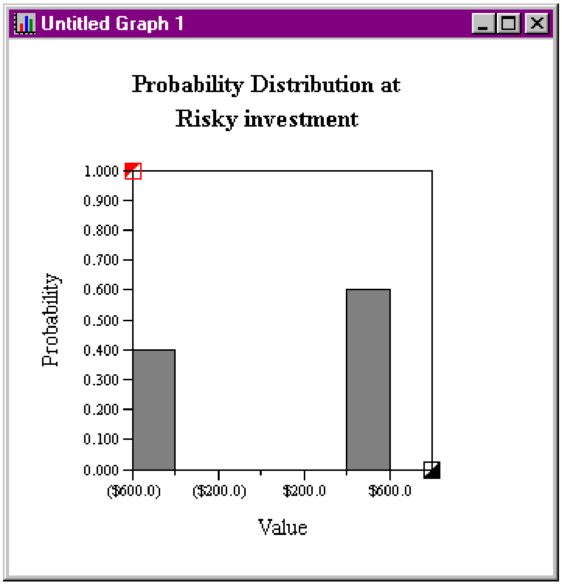
Probability distribution

The risk associated with one of the alternatives under consideration can be shown graphically using a probability distribution graph (or *risk profile*). Such a graph displays, in the form of a histogram, the dispersion of possible outcomes and the probabilities associated with those outcomes.

► **To view a probability distribution:**

▷ Select the Risky Investment node.

▷ Choose Analysis > Probability Distribution.



The resulting graph contains two bars. The first bar, whose height reflects its probability, 0.4, encompasses all outcomes between -$600 and -$400. In the Risky Investment subtree (whose root was selected when you performed the analysis), there is only one such outcome: the Market down node, with a value of -$600. The second bar, spanning $400 to $600, contains the other outcome, Market up, with a value of $500 and a probability of 0.6.

In this simple example, the probability distribution does not provide much additional information. In a tree containing dozens of outcomes, however, the extent to which the bars are clustered or dispersed around the mean would provide a graphical indicator of the risk of that decision.

Chapter 33 explains how to customize a probability distribution graph's display.

Cumulative and comparative probability distributions

A probability distribution bar graph can also be displayed cumulatively. Rather than using discrete bars to indicate the probability of an outcome within a specific range of values, the cumulative graph shows a continuous series of bars. The top of each bar indicates the probability of an outcome at or below a particular value.

To change the probability distribution graph to a cumulative graph, select Graph > Cumulative. To revert back to the noncumulative form, simply select Graph > Cumulative again.

When multiple nodes are selected (though not when a subtree is selected), the Probability Distribution menu item reads Comparative Distributions. Selecting this item will produce probability distributions for each selected node, and display the distributions together (in cumulative format) on a single line-graph.

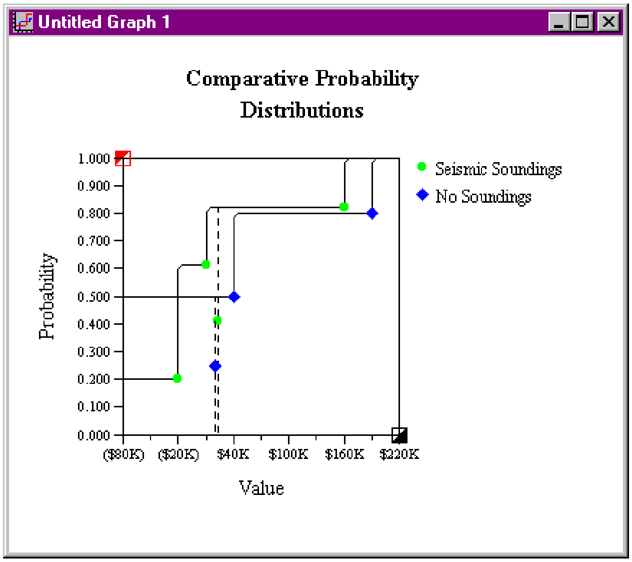
► **To generate a comparative distribution:**

▷ Open the sample file Oil Drilling Problem.

▷ Select the Seismic Soundings node.

▷ Hold down the SHIFT key, and select the No Soundings node.

▷ Pull down the Analysis menu, and select Comparative Distributions.



The resulting graph displays, on a comparative basis, cumulative probability distributions for the competing partial strategies Seismic Soundings and No Soundings. To allow multiple distributions to be displayed in a single graph, the cumulative distributions are displayed using lines, instead of the bars used in a simple comparative distribution. The lines describe the same shape as the cumulative bar graph.

Each strategy has a single vertical, dotted line, labeled with the strategy's legend symbol, indicating its expected value.

Dominance in probability distributions

It is important to understand how comparative probability distributions are interpreted. Comparative distributions can display two types of dominance – deterministic and stochastic. Conditions of dominance can provide more insight into a decision than simple expected value comparison.

Deterministic dominance occurs when one strategy always offers a better outcome than any alternative. Not only does it have a higher expected value, but its worst outcome (if there is any uncertainty) is preferable (or at least equal) to the best outcome of any alternative. In a comparative probability distribution graph, the entire plot of the dominating alternative will (with one possible exception) lie to the right of the lines representing the other strategies. The exception permits the worst outcome of the dominating strategy to be equal to (drawn with the same line as) the best outcome of the second ranked strategy.

Stochastic (or probabilistic) dominance involves more complex rules of interpretation. Stochastic dominance can be inferred if the line describing one alternative's cumulative probability distribution (1) is never located to the left of a competing strategy's line, and (2) is to the right of the competitor in at least one location. It is possible to work backwards from the values where the dominating alternative's line is to the right of the dominated strategy, and thus identify outcomes (both in the dominating and dominated strategies) which are crucial to dominance. The cause of stochastic dominance can often be seen in the dominating strategy as better probabilities, payoffs, or a combination of both. This may identify areas in your model for additional research into probabilities and payoffs.

In situations where their is no clear dominance, the lines of the strategies being compared will cross at one or more points in the graph. This is the case with the comparative risk profile generated for the Oil Drilling model. In comparing risk profiles where no alternative clearly dominates, it may be useful to consider risk preference (in addition to expected value). See Chapter 30 for information on using DATA's risk preference features.

Assessing uncertainty

Quantities incorporated into a tree are often uncertain. Many probabilities are estimates, sometimes representing nothing more than an educated guess. The same is usually true if at least some of the quantities included in payoffs. In the problems considered in decision analysis, future revenues or costs, market activity, competitive environment, mortality rates, discount or interest rates, judicial decisions, jury awards and many other important factors are rarely known in advance with certainty and precision. DATA offers several options for analyzing uncertainty in a model; the most fundamental are the various forms of sensitivity analysis.

Understanding sensitivity analysis

Sensitivity analysis makes it possible to assess how your decision is affected by variation in one or more of the uncertain quantities in the model. Basic one-way sensitivity analysis is covered here. Advanced sensitivity analysis topics are covered in Chapter 22.

To explore sensitivity analysis, you will use the Stock Tree created in Chapter 3. Select it from the Window menu or, if it is no longer open, use the File menu.

In analyzing the investment problem at the outset of the tutorial, you estimated probabilities and payoffs, based on your experience as an investor and the advice of experts.

Neither the probabilities at chance nodes nor the payoffs associated with terminal nodes are certain; they are merely best estimates or guesses. It is important to know how your strategy would be affected by changes, for example, in the probability of a market decline or the interest rate available on CDs.

These and similar questions can arise at any stage of the problem.

Let’s say that you receive new information that raises doubts concerning your earlier estimate that the market would rise with a probability of 0.6. You conclude it would be prudent to perform a sensitivity analysis on the variable prob_up, varying it between 0.35 and 0.7 to see how this affects the initial decision on how to invest.

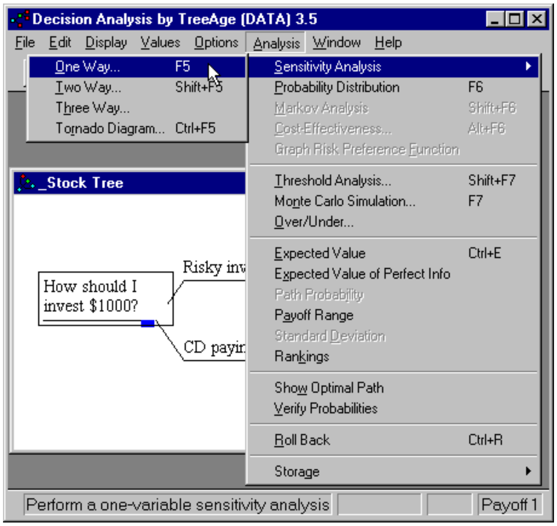
Performing a sensitivity analysis

It is critical to select the appropriate node prior to performing the analysis. In order to test the effect of changes in the probability of the market rising on the decision of where to invest, the sensitivity analysis should be done at the root node. If you had a larger tree with several intermediate decisions, you could view the effect on a particular downstream decision by performing the analysis at that decision node.

The results of the analysis, as you will see, will be represented as two lines, one for each alternative. Deviations of a line from the horizontal indicate changes in the expected value of that alternative as DATA changes the value of prob_up. Points where the lines cross and the optimal alternative changes, known as *thresholds*, provide crucial information about the sensitivity of your model to the varied parameter.

The same analysis process is used whether the variable in question is a probability variable or a component of the payoff formula.

DATA enables you to test the sensitivity of a proposed decision (i) to changes in the value of a single variable (known as a *one-way* sensitivity analysis), (ii) to simultaneous changes in the values of two variables (*two-way* sensitivity analysis), or (iii) to simultaneous changes in the values of three variables (*three-way* sensitivity analysis). This chapter will cover only one-way sensitivity analysis. Multi-way analyses, and advanced options for one-way analyses, are described in Chapter 22.



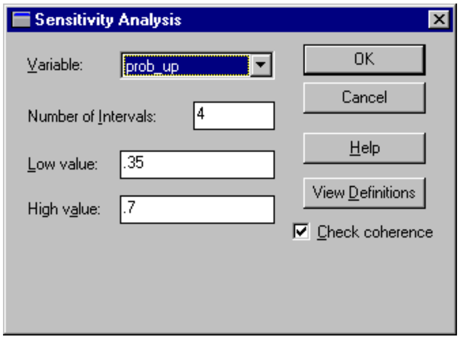
Perform a one-variable sensitivity analysis

► To perform a sensitivity analysis:

- ▷ Select the original Stock Tree from the Window menu or open it using the File menu.
- ▷ Select the root node.
- ▷ Choose Analysis > Sensitivity Analysis > One Way....

You are presented with a dialog box in which you must specify the variable on which the sensitivity analysis is to be performed.

- ▷ Click on the Variable pop-up menu, and select prob_up.



Variable: prob_up

Number of Intervals: 4

Low value: .35

High value: .7

☑ Check coherence

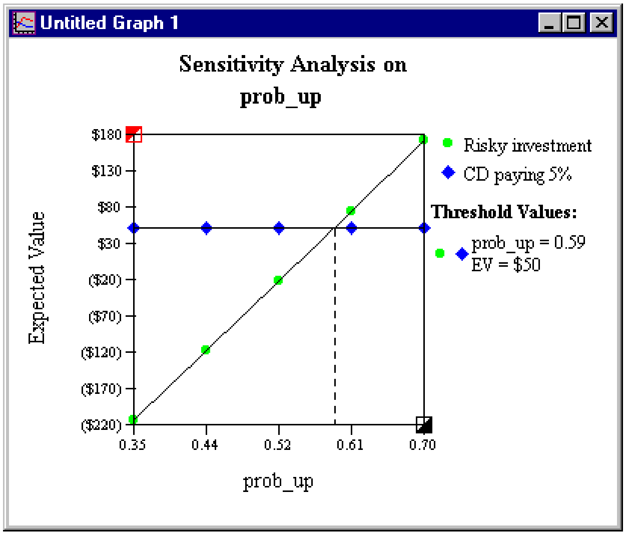
Now, you must specify the range within which to vary the value of prob_up. It is currently defined as 0.6; in this example, you will vary it from 0.35 to 0.7.

- ▷ Type `0.35`in the Low value field.
- ▷ Type `0.7`in the High value field.
- ▷ Press ENTER or RETURN.

Since there is only one (default) definition of prob_up in the tree, DATA presents only the above dialog before performing the analysis. For variables with multiple definitions, you would indicate for each definition whether you wanted to apply the specified range to that definition or leave its value unchanged; this process is described in greater detail in Chapter 22.

The sensitivity analysis graph

As a result of the sensitivity analysis, you are presented with a line graph that contains one line for each branch emanating from the selected decision node. In the graph on your screen, there are two lines, one representing the Risky investment option, the other representing the CD paying 5% option. The lines have markers at each interval, which are explained in the legend to the right of the graph.



The values displayed along the vertical axis are the expected values of the two alternatives. The lines represent the changing expected values of those alternatives as the variable (shown on the horizontal axis) is varied.

If a one-way sensitivity analysis graph contains one or more horizontal (rather than oblique) lines, this means that the alternatives represented by the horizontal lines are unaffected by the change in the chosen variable. In most cases, this indicates that the variable chosen for sensitivity analysis is not involved in computing any expected values for that subtree. It is also possible that the variable *is* used in the subtree, but that the definition of the variable in that subtree has been held constant for purposes of the sensitivity analysis. See Chapter 22 for details.

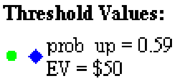
Thresholds

The sensitivity analysis results require some interpretation. At the point at which the lines intersect, the subtrees being compared have the same expected value, and from the standpoint of expected value, the decision maker should be indifferent between the two treatment options. Such points of indifference are also known as *thresholds*. They represent points at which a change occurs in the optimal strategy.

DATA's one-way sensitivity analysis includes the results of a limited threshold analysis. This means that at every crossing point in the optimal path – each point at which a change in the recommended strategy occurs – a dotted line drops down to the horizontal axis.

This threshold information may be hidden by selecting Graph > Display Threshold Values. Turning it off will remove all dotted lines and the entire threshold legend. If no threshold values exist (because the optimal strategy does not change), the menu item is dimmed and no threshold information is displayed.

In sensitivity analysis, DATA identifies the crossing points by linear interpolation, not by successive approximation, so accuracy can be improved by reducing the range of the analysis or increasing the number of intervals.



Threshold Values:

● ◆ prob up = 0.59
EV = $50

A legend specifies the threshold values. There will be a separate legend for each crossing point, displaying the line markers for the lines which cross. The values, at the crossing point, of the variable (e.g., prob_up) and the expected value (or utility) are displayed in the legend to the right of the line markers.

Even though DATA's graphs are otherwise highly customizable, none of the threshold legend text may be directly changed – these are calculated values. The only aspects of the threshold legend which can be modified are the two numeric formats, and only indirectly; if you change the numeric format of either axis, the format of the corresponding item in each legend entry will be adjusted automatically. See Chapter 33 for more information on customizing the display of graphs.

According to the legend, the threshold occurs when the probability that the market will rise is 0.59. To the left of this point, the line representing the CD paying 5% option is always above the Risky Investment line. Therefore, if the probability of the market's rising is less than 59%, you should invest in the CD. Once the probability of the market's rising exceeds 59%, you should bet on the stock market.

A caveat on thresholds

It is important to be aware that the sensitivity analysis determines only discrete points in the graph, which are then joined by lines. The number of discrete points is determined by the number of intervals selected by the decision maker. For example, four intervals corresponds to five discrete points.

The lines charted on the graph are accurate at these discrete points, but not necessarily in between.

Threshold information is also subject to the limitation that it is determined at these discrete points. DATA checks for the optimal strategy at each point and recognizes a threshold if DATA recommends a change in strategy. This also means that DATA will *not* recognize a threshold if the optimal strategy is the same at both ends of the interval, even if the strategy changes back and forth within the interval. You can reduce the likelihood of this error by keeping the size of the intervals reasonably small.

A more sophisticated, and potentially more accurate, threshold analysis is described in Chapter 22.

To test your interpretive skills, try performing the same sensitivity analysis as above on the decision in the Two Stock Tree. The graph generated will contain three lines (one for each investment option) and two thresholds.

Chapter 7 Printing

Printing DATA documents is similar to printing documents from any other Windows or Macintosh application, but some added features allow you to customize the display of the printed pages.

Open Stock Tree using the File > Open... command.

► **To print a document without customization:**

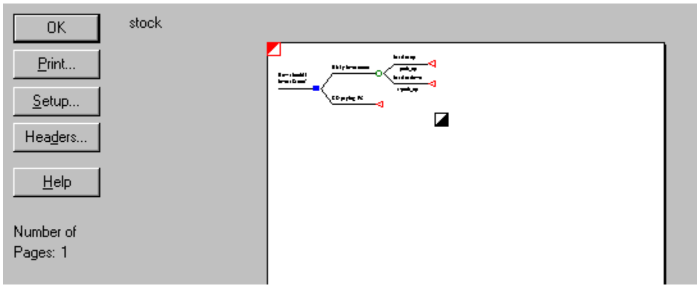
▷ Choose File > Print..., and press ENTER or RETURN.

Your printer should print the investment tree. The tree is not centered in the printout; instead, it is printed in the upper left quadrant of the page.

As with other Windows and Macintosh applications, you may change your printer's settings by selecting File > Page Setup.... There are also two DATA-specific ways to customize printouts. One involves the Print Preview window, and the other involves the Printing page of the Preferences dialog.

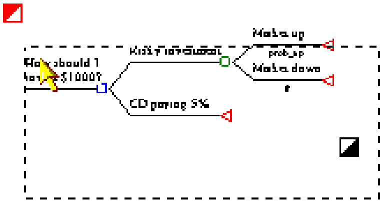
Print preview

You may see a preview of how your printed document will appear on the printed page by choosing File > Print Preview....



Number of Pages: 1

The tree should appear in the upper left of the print area. (If, instead, the tree is centered, go to Printing Preferences, and deselect Center in page.) A half-filled red square is visible at the top left of the tree; a similar black square is visible at the bottom right of the tree. These squares enable you to move and resize the tree within the print area.



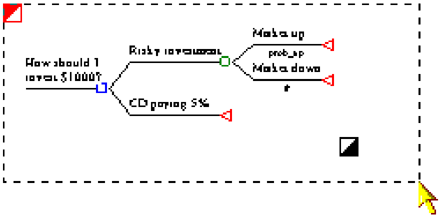
► **To move the tree manually within the printed page:**

▷ Click on the red square.

▷ While holding down the mouse button, move the mouse down and to the right. A rectangle surrounding the tree moves with the mouse.

▷ Release the mouse button. The new position of the tree indicates where it will be printed.

If you wish simply to center the tree on the page, you can do so automatically by following the instructions under Printing Preferences, later in this chapter.



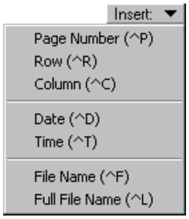
► **To resize the tree within the printed page:**

▷ Click on the black square.

▷ While holding down the mouse button, move the mouse down and to the right. The rectangle surrounding the tree grows and shrinks according to movements of the mouse.

Notice that the growing rectangle retains the proportions of the rectangle that enclosed the original tree. DATA maintains these proportions to prevent distortion of the tree.

▷ Release the mouse button. The tree is displayed in the new size in which it will be printed.



► **To add headers or footers:**

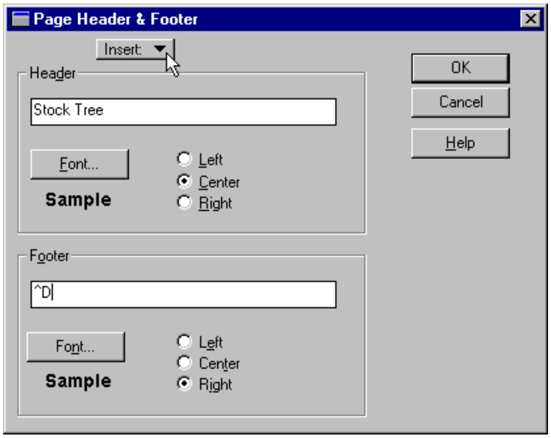
▷ Click on the Headers... button.

▷ Enter the desired text in the Header field and/or the Footer field. You can change the font and alignment of the text by clicking the appropriate buttons, or add special text (page number, date, file name, etc.) by clicking on the Insert pop-up menu.

▷ Press ENTER or RETURN.

Your header and/or footer will appear in the print preview.

Headers and footers are available only when printing a tree. However, it is possible to move and size any printable document.



Header

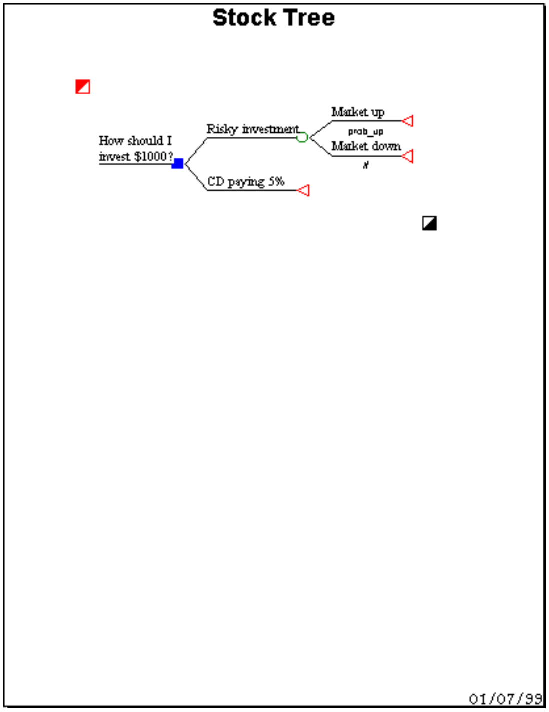
Stock Tree

○ Left
● Center
○ Right

Footer

^D

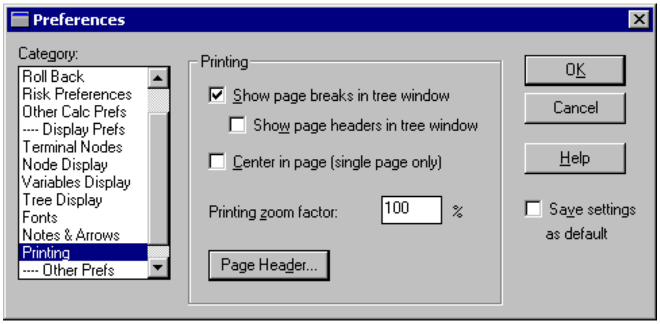
○ Left
○ Center
● Right



Your changes to the printed tree’s placement, size, and headers and footers will be saved with the tree.

Printing preferences

Several additional features for customizing printing are available in the Printing page of the Preferences dialog. To access them, choose Edit > Preferences and select the Printing page (under Display Prefs).



Printing

☑ Show page breaks in tree window

□ Show page headers in tree window

□ Center in page (single page only)

Printing zoom factor: 100 %

□ Save settings as default

For trees that span multiple pages when printed, it can be helpful to see in the Tree Window where the page breaks will fall. You may do so by selecting Show page breaks in tree window. If this option is selected, you may also elect to display headers and footers on each page in the Tree Window by clicking Show page headers in tree window.

It is possible to center a single-page tree on the printed page by clicking Center in page. This setting will be reflected in the Print Preview window.

You may adjust the enlargement or reduction of the tree by changing the Printing zoom factor. The default zoom factor is 100%. The printing zoom factor is also linked to the black square in the Preview window.

Note that you may enter or modify headers and footers by clicking the Page Header... button. The functionality is the same as if you had clicked the Headers... button in the Print Preview window.

Click OK to leave the Preferences dialog.

You have now completed Part II, the basic DATA tutorial. You are urged to continue with Part III of the manual, which covers a number of features designed to make your use of DATA more efficient and productive. Chapters 8 and 9, dealing with variables, are particularly important.

CHAPTER 8 VARIABLES – CONCEPT AND THEORY

This chapter explains the advantages of using variables, rather than specified point values, in computer-aided decision analysis. It also explains the logic that dictates where variables should be defined in a tree and how variables are evaluated. Both standard variable definitions and recursive definitions are covered here. There are special rules for tracker variables, used only in Monte Carlo simulation; these rules are covered in Chapter 29.

Chapter 9 includes detailed examples on using a variety of tools designed to make working with variables in decision trees easier; using variables in influence diagrams is covered in Chapter 31.

Representing values in your model

In building a model, two types of values will require quantification:

- **probabilities** – the likelihoods associated with the outcomes of the uncertain events specified in your model; and
- **payoffs** – the outcome values assigned to each of the scenarios included in your model.

Unless quantitative values are assigned to both probabilities and payoffs, it will not be possible to perform expected value calculations or otherwise subject the model to quantitative analysis.

Virtually all of DATA's calculations are performed on the basis of single, point values. (See Chapters 28–29 for exceptions to this rule.) When values are uncertain, which is invariably the case in the type of problems you will be analyzing, you should specify point values which reflect your best estimate.

It is possible to assign these numeric point values directly to both probabilities and payoffs. An example of this methodology can be found in Chapter 3. This method is simple and straightforward, but it suffers from serious drawbacks when the time comes to analyze the model.

When you roll back a decision tree, DATA will identify the optimal path on the basis of either maximizing or minimizing expected value. Since some of the values used in that calculation are uncertain, it is usually desirable to test whether changing one or more of the uncertain values will cause the optimal path to change. This is accomplished by changing the values of probabilities, payoffs, or their components, and then recalculating the model. This process, called sensitivity analysis, is facilitated in DATA with the use of variables.

Chapter 5 introduced variables, and demonstrated their use in assigning probability and payoff values. This chapter focuses more closely on the definition of variables in decision trees. Chapter 9 illustrates the various interface features and tools available when working with variables in a decision tree.

TIP: Since it is often easier to attach meaning to a word than to a number, variables can be valuable as communication tools. They can serve to clarify for the viewer the meaning of values in your model. For the model builder, variables can both reduce the work required to update values in the model, as well as serve as reminders of the purpose of particular raw numbers. Nonetheless, too many variables may clutter the visual display and increase calculation time. It may be desirable to limit the number of variables that represent constants, but are not required for performing sensitivity analysis.

An oil drilling problem

The following example will demonstrate both the advantages of using variables and some of the basic rules you will have to keep in mind when setting up your model.

Imagine that you are an oil wildcatter and have obtained an option to drill at a particular site. You must decide whether or not to incur the cost of drilling at this site before your option expires.

There are a number of uncertainties which complicate your decision: the cost of drilling, the amount of oil at the site, the cost of raising the oil, and so forth. However, you have analyzed objective records of similar and not-so-similar drillings in this same basin, and you have discussed the peculiar features of this particular site with your geologist, your geophysicist, and your land agent. In this way, you have developed a set of scenarios that describe some likely outcomes (payoffs) for locating and extracting oil at the site.

You could gain further information about the underlying geophysical structure at this site by conducting seismic soundings. The information provided by seismic soundings would be relevant to your inquiry, but would not be a perfect predictor of the presence of oil at the site. Moreover, it would be costly to secure this information. You must decide whether or not it pays to do the seismic soundings before making your final decision: drill or don't drill.

Of course, the problem may encompass other complications that you, the wildcatter, might have to deal with: where to drill, how to drill, whether to share the risk with others, how to raise capital, and so forth. However, you have decided to ignore these matters for now and to concentrate on a single aspect of the problem.

To keep things simple, you have decided to classify into only three categories all possible uncertain states with respect to the amount of oil at a site:

- Dry: There is no oil or just a negligible amount.
- Wet: There is sufficient oil for commercial purposes.
- Soaking: There is an unusually large amount of oil.

Furthermore, after consulting with the experts mentioned above, you have arrived at some estimates with respect to costs and revenues. The cost of drilling, though uncertain, is estimated to be $700,000. The revenues and the costs related to raising the oil will depend on the amount of oil present. Both the price for which the oil can be sold and extraction costs can be expected to vary over time. To keep the model simple, current prices and costs will be used in projecting net revenues.

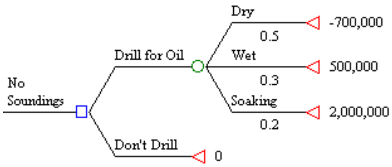
Considering all revenues and costs involved in extraction and sale, excluding the cost of drilling, the potential net revenues are estimated as follows:

•	Dry:	$0
•	Wet:	$1,200,000
•	Soaking:	$2,700,000

Based on your accumulated experience, your review of historical records of successes and failures at different sites, and the opinions of your experts, you have also estimated the probabilities that the uncertain outcomes of drilling described above will, in fact, occur.

Specifically, without any information (from seismic soundings) about the underlying structure, you have arrived at the following probabilities for the amount of oil at the site:

Dry:	0.5 (or 50%)
Wet:	0.3 (or 30%)
Soaking:	0.2 (or 20%)



Thus, in our model of the drilling problem, the expected value of drilling without undertaking prior seismic testing can be arrived at very easily. Simply apply these probabilities to the ultimate payoff of each of the three scenarios, or net revenues less the cost of drilling.

The seismic soundings, if undertaken, introduce additional complexities. First, there is the additional cost of the test, estimated at $100,000. Second, if you undertake the seismic soundings, you must integrate the new information the tests provide into your model.

The possible results of the seismic soundings are manifold but, again for the sake of simplicity, you have decided to summarize them into three possibilities:

- No structure: Test results indicate an absence of underlying geological structure at the site. From the standpoint of finding oil, this is a negative predictor.
- Open structure: Test results indicate the presence of an open geological structure, an ambiguous result.
- Closed structure: Test results indicate the presence of a closed geological structure. This is a positive predictor; the probability of a wet or soaking site is much greater than with the other test results.

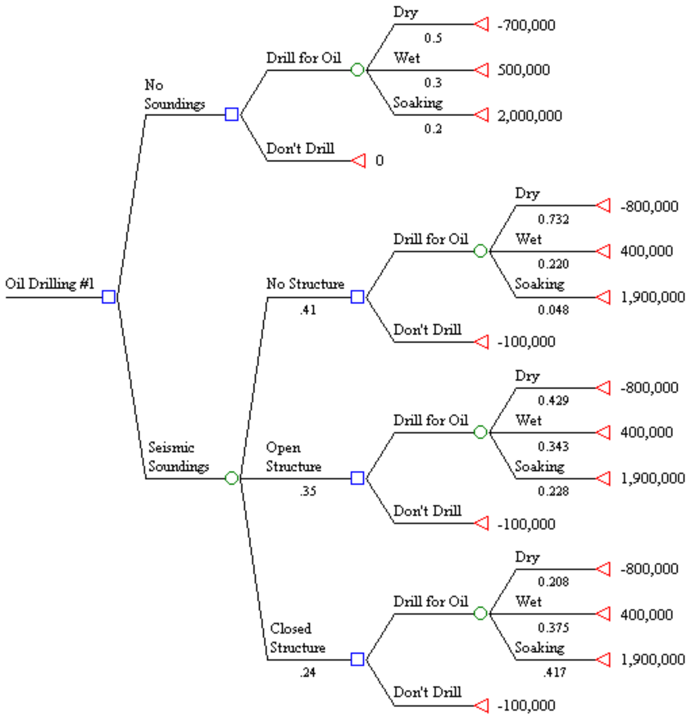
No result of the seismic soundings is a perfect predictor of the absence or presence of oil deposits. A result indicating no geological structure does not entirely rule out the possibility of a very productive well; nor does closed structure ensure that the site will not be dry. Based on accumulated experience, though, you and your experts can assess the probabilities for finding oil, given the outcome of seismic soundings. (How this can be done using DATA is explained in Chapter 24.) The table on the following page sets forth these probability assignments.

	Dry	Wet	Soaking
Given No Structure	0.732	0.220	0.048
Given Open Structure	0.429	0.343	0.228
Given Closed Structure	0.208	0.375	0.417

A tree without variables

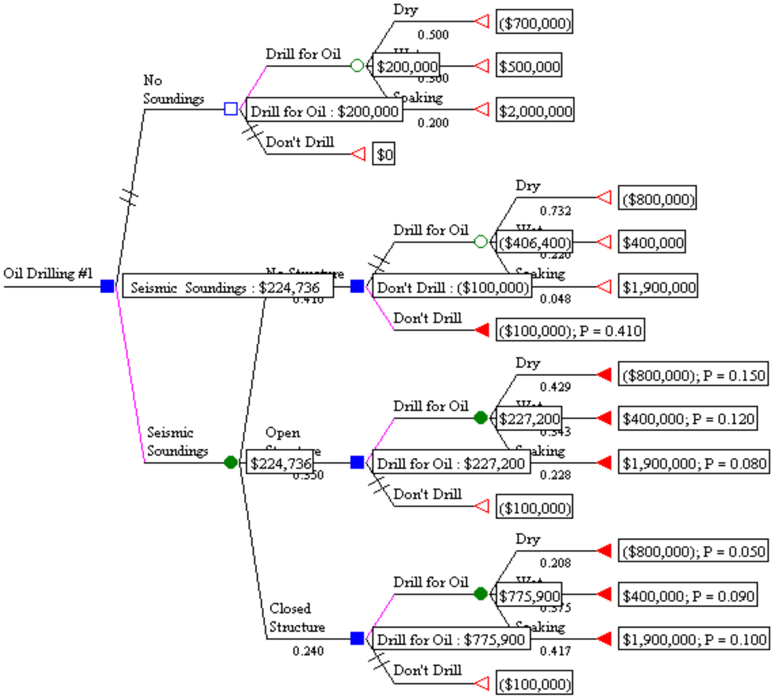
You are ready to use DATA to assist you in making your decision to drill or not to drill. The options and scenarios discussed above are modeled in the file Oil Drilling #1, shown here.

The probabilities come from the tables shown above. The payoff for a given scenario represents the estimated profit for that scenario, calculated by deducting from net revenues the applicable costs of drilling and seismic soundings. Depending on whether or not the scenario included seismic soundings, whether or not you decided to drill, and whether the site turned out to be dry, wet, or soaking, you obtained a different amount of net profit. This amount was assigned as the payoff at the corresponding terminal node.



▷ Open the file Oil Drilling #1.

▷ Select Analysis > Roll Back.



On the basis of expected values, your optimal strategy for the initial decision is to perform seismic soundings. Although seismic testing will add costs at the outset, the new information they provided may substantially reduce risk by improving the probability distribution of outcomes.

For example, if you choose *not* to test and then drill, you will be exposed to a 50% risk of a large loss (-$700,000). If you choose to test, the risk of a comparable loss (-$800,000) can be reduced to 20%.

Any time you choose to drill, even with the benefit of a closed structure finding at the site, the outcome of drilling will be uncertain. Aside from uncertainty regarding the results of drilling, your computation of net profit is based on your projections of drilling costs, test costs, oil extraction costs and sale proceeds, each of which raises issues of uncertainty. Thus, it would be desirable to be able to analyze the tree on the basis of somewhat different estimates of these costs and revenues.

This process, called sensitivity analysis, can be achieved by substituting recalculated values for some or all of the fifteen nonzero payoffs in the tree, recalculating the tree after each set of changes, and analyzing the impact of the changes on the strategy recommendation. While possible, this approach would be tedious and inefficient.

A tree with variables

DATA offers a far more efficient means of testing the sensitivity of optimal path calculations to changes in the underlying assumptions.

▷ Open the file Oil Drilling #2.

This model is identical to Oil Drilling #1 except that it uses variables and formulas to calculate payoffs.

▷ Select Analysis > Roll Back.

As you can see, the calculated values and optimal path recommendations are identical to those produced by rolling back Oil Drilling #1. If you are wondering why the methodology employed in Oil Drilling #2 is preferable, bear in mind that this model is fairly simple. Just imagine the tedium involved in calculating by hand the payoffs for each terminal node in a model having hundreds, or even thousands, of scenarios. Then, consider the added burden involved in a manual recalculation of each payoff every time you want to see the effect of changing one or more of the underlying values.

DATA has been designed to make it easy to set up models using variables. It has also been designed to make the use of variables as efficient as possible, by minimizing the number of steps needed to assign numeric values to them.

In order to define variables correctly and with maximum efficiency, it is important to learn some basic rules. You should start by examining Oil Drilling #2 to see where and how variables are defined.

Oil Drilling #2 should be in the active tree window. If the tree is still rolled back, turn off the display of calculated values by choosing Analysis > Roll Back again.

Using variables in your payoff formula

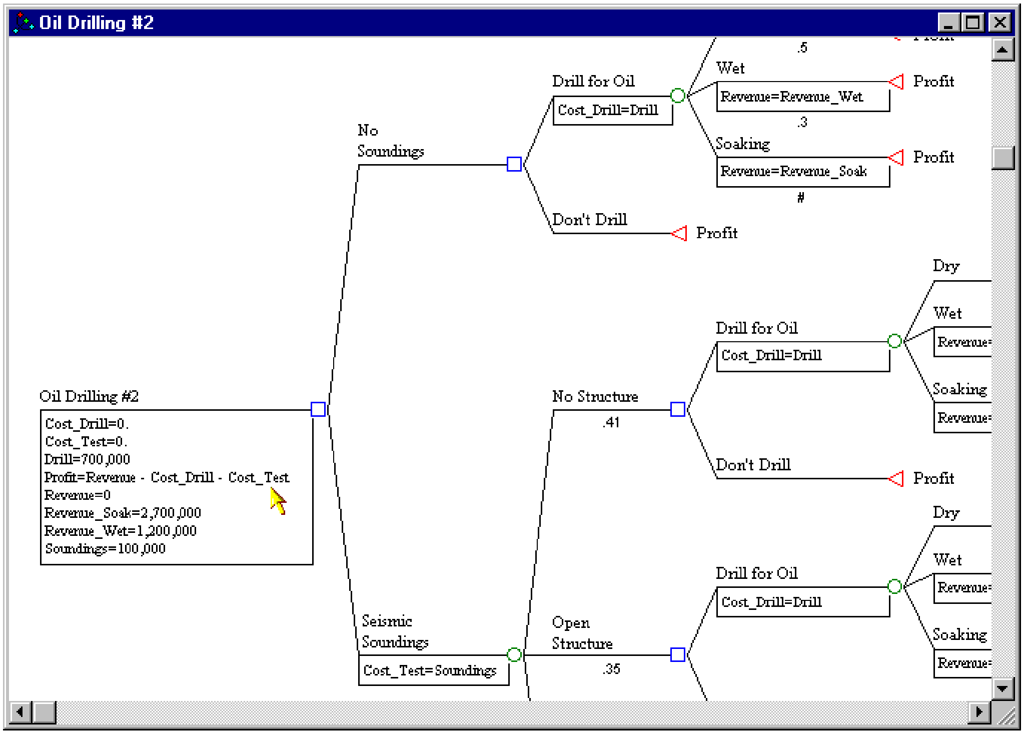
The process of setting up variable expressions for your model's payoff calculations, covered initially in Chapter 5, has three basic steps:

1. create the necessary variable or variables;
2. enter an expression including these variables in the payoff of one or more terminal nodes; and
3. define (assign values to) the created variables.

DATA provides more than one method (and order) by which these steps can be accomplished for your tree. Often, you will find that there is more than one way to set up your payoff variable expressions and definitions. As you make decisions about where to define variables in your model, keep two goals in sight: clarity of relationships (for your audience) and simplicity of sensitivity analysis (for you, the analyst).

In Oil Drilling #2, the payoff formula used throughout the tree is represented by the variable Profit. Each terminal node uses the variable Profit to represent its payoff. Note that Profit does not have a numeric definition anywhere in the tree. Instead, Profit has a default formula definition, shown in the box below the root node:

Profit = Revenue - Cost_Drill - Cost_Test.



Cost_Drill=0.
Cost_Test=0.
Drill=700,000
Profit=Revenue - Cost_Drill - Cost_Test
Revenue=0
Revenue_Soak=2,700,000
Revenue_Wet=1,200,000
Soundings=100,000

The right-hand side of the equation is shorthand for

(Net revenues from sale of oil) less (Cost of drilling) less (Cost of seismic tests).

It is often desirable to use the same payoff formula at every terminal node in your tree, particularly when you are initially learning how to use variables. This will usually mean that components of your payoff formula will be inapplicable to certain scenarios. For example, in the oil drilling problem, the payoff used at every terminal node includes the variable Cost_Test, even though seismic soundings are not performed in any scenario contained in the No Soundings subtree.

As a result, you must be sure to assign a numeric value to Cost_Test, not only in those scenarios where seismic soundings will be performed but also in the other scenarios. In the latter case, the value assigned should be zero. Simply leaving Cost_Test undefined in the No Soundings subtree would result in an error message whenever you perform a calculation, such as roll back, that includes the No Soundings subtree.

Where payoff variables should be defined

It is very important that value assignments be made at the correct nodes. Failure to do so will, at the very least, limit the scope and flexibility of available analyses, and it may result in incorrect calculations and analytical error.

It is not difficult to identify the correct node for each value assignment, but it will require you to become familiar with the processes DATA uses to search for and evaluate variables, particularly those involved in sensitivity analysis. This, along with at least a basic competency with DATA's mathematical operators and built-in functions (see Chapter 9 and Appendix C), is fundamental to good modeling technique.

TIP: When the full display of definitions is activated in your tree, the definitions specified at a particular node will be displayed in a box below the node's branch description (immediately to the left of the node symbol). See Chapter 9 for information on setting your tree's variables display preferences.

DATA's variables search

DATA's expected value analyses begin by calculating the value of every payoff, scenario by scenario. When variables are used in calculating a scenario's payoff, DATA will search each node in the scenario, beginning with the terminal node and traversing leftward towards the root node, looking for definitions for each relevant variable. For a given variable, DATA accepts the *first* definition (value or formula assignment) that it locates on the journey *from terminal node to root node*.

If DATA fails to find a value assignment for any variable specified in the payoff formula, DATA will issue an error message. This is true even in the case of variables which have a zero value for the scenario in question. For example, Revenue has a zero value in every scenario ending with a Don't Drill or a Dry node. Thus, even zero definitions must be expressly assigned. As you will see, this is often accomplished by assigning a zero definition at the root node.

In the case of a variable used to represent a probability, DATA will search for the value assigned to the variable beginning with the node whose probability is being calculated and proceeding leftward to the root node.

If a variable is defined in terms of other variables, the search for definitions of the component variables begins at the original node being calculated. In the Oil Drilling #2 model, for example, the payoff of any terminal node is the variable Profit. To evaluate the entire tree, all 16 terminal nodes' payoffs are evaluated. For each payoff calculation, DATA carries out a right-to-left search for a definition of Profit, getting all the way to the root node before it finds the formula definition.

Then, DATA must evaluate each of the components of the profit calculation: Revenue, Cost_Test, and Cost_Drill. The search for definitions for each of these component variables begins *at the terminal node being calculated*, not at the root node, where the definition of Profit is found.

Similarly, if a variable is used for a probability, and it has a formula definition at, say, the root node, the search for definitions of component variables starts at the node where the probability is used. The same rules apply to formulas used for primary quantities in a Markov process, such as state or transition rewards: the right-to-left search for component variables always begins at the node for which a probability or reward is being calculated. DATA will disregard variable definitions to the right of the node at which the original formula is first referenced.

An important exception to the above search rules is the case of recursive variable definitions (e.g., Costs=Costs+X). This topic will be covered in detail later in this chapter. Please postpone using recursive definitions until you are familiar with the special rules covering them.

TIP: Parsimony is one of the hallmarks of good modeling. This means that each variable should be defined (assigned a numeric value or formula) in as few locations as possible. The goal is to make it as simple as possible to change definitions or to perform sensitivity analysis.

Variable definitions and sensitivity analysis

The oil drilling payoff formula defines Profit in terms of three variables, Revenue, Cost_Test, and Cost_Drill. Respectively, these three variables represent net revenues from the sale of oil, the cost of performing seismic soundings, and the cost of drilling for oil. None of the three variables will have the same value throughout the tree.

Revenues will be zero in those scenarios where you decide not to drill or where you drill and come up dry. Where you drill successfully, revenues will be either $1.2 million or $2.7 million, depending on whether the site is wet or soaking. The cost of seismic soundings will be zero, except in the subtree where the test is performed and this cost will be $100,000. The cost of drilling will be zero, except at each Drill for Oil node, where this cost will be $700,000.

The simplest way to deal with these different values would be to give each of the three variables multiple numeric definitions. For example, Revenue could be given a default definition of zero at the root node and definitions of either $1.2 million or $2.7 million, as appropriate, at each wet and soaking node.

A tree constructed in this fashion will calculate properly. However, your ability to perform sensitivity analysis on the variable Revenue will be severely constrained.

When performing a sensitivity analysis, you must substitute a range of values for the single value at which the variable is nominally defined. If the variable has multiple definitions in the tree, you will be able to substitute the range of values for any or all of the definitions, but the range of values cannot differ from definition to definition. Your only options are to use either the range you have specified or the point value specified in the tree.

Let's see how this would work if you propose to perform a one-way sensitivity analysis on the variable Revenue. Recall that, depending on the scenario, Revenue may have a value of zero, $1.2 million, or $2.7 million.

At the outset of the sensitivity analysis, you will be asked to specify a range of values for Revenue. How will you determine the range? Where you don't drill or don't find oil, there is no uncertainty; the definition of Revenue should be zero.

The purpose of a sensitivity analysis on Revenue must be to address uncertainty regarding net revenues in those situations where you drill and find oil. However, you have already determined that it is necessary to distinguish between those scenarios where the site is soaking and those where it is only wet. Since a single point value could not be used for both, it follows that, when performing sensitivity analysis, different ranges will also be required.

In other words, if there is uncertainty regarding the net revenues to be obtained from the sale of oil from a wet site, it is likely that many of the same factors (price of oil, extraction costs per barrel, etc.) will also apply to a soaking site. A sensitivity analysis on Revenue will need one range of potential net revenues for soaking and another for wet. While this isn't possible directly, there is an expeditious workaround: simply define Revenue in terms of other variables and perform sensitivity analysis on the latter.

Thus, at each Wet node, define Revenue=Revenue_Wet, and at each Soaking node define Revenue=Revenue_Soak. Revenue_Wet is given a default definition of $1,200,000 and Revenue_Soak is given a default definition of $2,700,000. The existing zero definition of Revenue will apply in all other scenarios.

The variables Cost_Drill and Cost_Test are less complicated. While each has two different values in the tree, one of those values is zero – in those scenarios where you don't drill or you don't do the seismic soundings. When performing a sensitivity analysis on either Cost_Drill or Cost_Test, a single range of values will suffice for every scenario where you drill or where you do the seismic soundings. Nevertheless, these is another reason why it may be desirable to use multiple variables to distinguish between those situations where the cost is incurred and those where it is not.

To understand why this is the case, let's see what would happen if you performed a sensitivity analysis on Cost_Drill. Consistent with the rule that each variable should be defined in as few locations as possible, Cost_Drill is given a default value of zero at the root node, and a value of $700,000 at each of the four Drill for Oil nodes.

When performing a sensitivity analysis on Cost_Drill, DATA will identify all five nodes (the root node and four Drill for Oil nodes) at which Cost_Drill has been defined and will give you the opportunity at each to accept the existing point value or to replace it with the range of values you have specified for this analysis.

To avoid having to go through multiple definitions every time a sensitivity analysis is performed on Cost_Drill, consider the following alternative. A default, zero definition is still assigned to Cost_Drill. At each of the four Drill for Oil nodes, define Cost_Drill in terms of a new variable, Drill. At the root node, define Drill=$700,000.

Make sure that you understand why this works – why Cost_Drill will now have a value of zero for every scenario in the tree, except where you drill for oil, where Cost_Drill is assigned the value of Drill, $700,000.

Similarly, assign Cost_Test a default, zero definition. Then, for the Seismic Soundings subtree, define Cost_Test in terms of a second variable, Soundings. Back at the root node, define Soundings=$100,000.

In the case of each component of the payoff formula, Revenue, Cost_Drill, and Cost_Test, *the default, zero definition assigned at the root node will apply in all scenarios except where you have specified an overriding, nonzero definition somewhere to the right of the root node*. Since you have used a separate variable to define each of these uncertain, nonzero values, you can easily perform a sensitivity analysis on any of them.

Thus, when you want to perform a sensitivity analysis on the revenue from a wet or soaking well, use the variables Revenue_Wet or Revenue_Soak, *not* Revenue. Similarly, when you want to perform a sensitivity analysis on the cost of drilling or the cost of seismic soundings, use the variables Drill or Soundings, not Cost_Drill and Cost_Test.

Since each of the new variables will have only a single value in the tree, not only will one-way sensitivity analysis be easier, but it will also be a simple matter to perform multi-way sensitivity analysis or, better, to implement correlations among related variables.

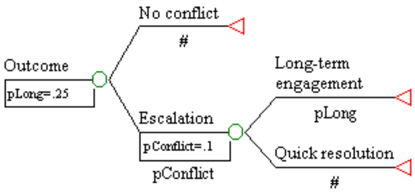
Chapter 22 contains more on the subject of performing sensitivity analysis on variables with more than one definition.

Probability variables

The foregoing discussion related to variables used to calculate the payoff. The rules are largely, but not exactly, the same in the case of variables used to define probabilities.

There are several possible reasons to use variables for probabilities. Most often, variables are used because they are a prerequisite to being able to perform sensitivity analysis on estimated probabilities. Other reasons include:

- to establish linkages between the probabilities associated with different events;
- to establish linkages to other trees (see Chapters 12 and 15);
- to establish linkages to other applications, such as spreadsheets or databases (see Chapter 15); and
- when cloning subtrees (see Chapter 12), in order to have different probabilities for the same event in copies of the cloned subtree.



When DATA encounters a variable in a probability field during calculations, it searches for a value assignment beginning with the node whose probability field contains the variable. From there, it moves leftward towards the root node. In the example shown in the left margin, the probability expression for Escalation, pConflict, will be evaluated using the definition found at the same node. The probability expression at the Long-term engagement node will be evaluated using the default definition of pLong, found at the root node.

As a general rule, it is advisable to avoid using the same variable to define the probability of two or more distinct events. The purpose of this rule is to facilitate easy and accurate sensitivity analysis. Thus, if you have two subtrees representing the same type of uncertainty (for instance, the likelihood of no oil), but the probability values are different, you should use different variables. In the long run, you will save time and effort. See Chapter 9 for more on this topic.

Of course, if the probability value is, and always will be, the same throughout the tree, define a single variable default for the tree.

TIP: Chapter 5 shows how to create a variable by typing its name in a probability field (below the branch line emanating from a chance, logic, or Markov node). It is also possible to place an existing variable in a probability field using the Values > Insert Variable command (see Chapter 14).

Sensitivity analysis and probability variables

When performing a sensitivity analysis, you are first asked to identify the variable(s) which will be the subject of the analysis and to specify a single range of values for each of the chosen variables. It is not possible to employ multiple ranges for one variable during a single sensitivity analysis. Thus, you will not be able to perform a valid sensitivity analysis on a probability variable used at multiple events unless the value assignments in each case are identical.

After you have specified one or more variables and the values range associated with each, DATA checks to see if the variable being analyzed is defined at multiple locations in the tree. If it is, you will be required to specify for each node where the variable is defined whether to: 1) retain and hold constant the value assignment specified at that node; or 2) substitute the range of values specified for the variable at the outset of the sensitivity analysis.

There is an easy workaround to avoid the limitations imposed by this process. It involves defining the variable in question in terms of a second variable, and specifying that second variable as the subject of the sensitivity analysis.

Probability variables in cloned subtrees

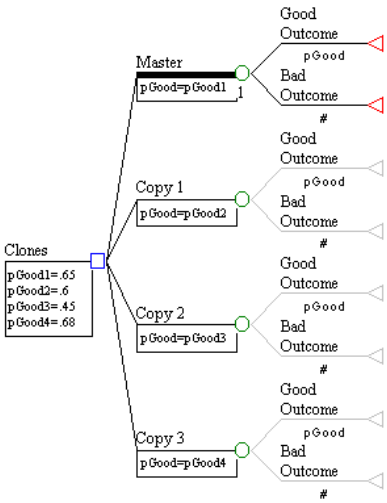
If you are working with cloned subtrees, you cannot avoid using the same probability variable(s) in the clone master and each clone copy, even if the value assignments for each should be different. Although the same variables will appear in every copy of the cloned subtree, they can be defined differently for the master and each copy. Unique definitions (value assignments) are customarily made at the root nodes of the clone master and the clone copies. (If you are unfamiliar with using clones, covered in Chapter 12, you should skip to the next section, which deals with non-clone tree structures.)

Assume that the variable pGood is used to define a probability in a cloned subtree. It appears once in the subtree which acts as the clone master and, since the subtree is replicated three times, in each of the three clone copies.

At the root node of the clone master, pGood is assigned a value of .65. At the root nodes of the clone copies, pGood is assigned values of .45, .6 and .68, respectively.

If you choose to perform a sensitivity analysis on pGood, you must designate a single range of values which can be assigned to pGood. This range can then be assigned at one or more of the four nodes where pGood is assigned a numeric value. At any of these nodes where you choose *not* to assign the range, the single value for pGood already assigned at that node will be retained.

How can you perform a sensitivity analysis assigning different ranges at two or more of the places where pGood is defined?



The answer is to introduce additional variables. There are a number of approaches to consider:

- create four new variables called pGood1, pGood2, pGood3, and pGood4, defined using the default numeric values .65, .45, .6, and .68, and define pGood using each new variable; or
- create a second variable, such as p1, define it numerically at the root node (=.65), and then define pGood using proportional references to p1 (pGood=p1, =.45/.65*p1, =.6/.65*p1, =.68/.65*p1).

The first technique is probably better, at least in this case. Simply define pGood at the root node of the clone master with the formula

pGood = pGood1.

At the root node of the first clone copy define pGood by the formula

pGood = pGood2

and so on.

Since each of the new variables will have only a single value in the tree, it will be a simple matter to perform multi-way sensitivity analysis or, better, to implement correlations among the variables. Thus, different ranges can be applied to the probabilities in the different subtrees during analysis.

A similar process can be used to assign different definitions to variables used to calculate payoffs of cloned subtrees.

Probability variables in non-cloned subtrees

Now consider an essentially identical tree. The only difference is that the subtrees were replicated by the copy/paste method, rather than by cloning. Unlike cloned subtrees, where the name of a variable used in a probability (or payoff) will be identical in the clone master and all clone copies, subtrees created by the copy/paste methodology are not so restricted. You have the flexibility of using a different variable in each of the subtrees, or you can use a single variable and then go through the same two-step definition process just described for use with clone subtrees.

Is one methodology preferable?

If the variable pGood is going to have the same value at each place where it appears in the tree, it is desirable to use a single variable throughout. Someone viewing the tree will recognize that each of the probabilities represented by pGood is related and is identical.

In contrast, if the variable pGood has different values at different points in the tree, it is desirable to use a different variable at each point. Even though it is possible to use a single variable and assign different numeric values at different locations, this may send the wrong message to someone viewing the tree. On the face of the tree, it will appear that each of the probabilities is both related and identical when, in fact, they may be related but are not identical.

Accomplishing both clarity and ease of analysis is easy. Simply use a different variable for each probability.

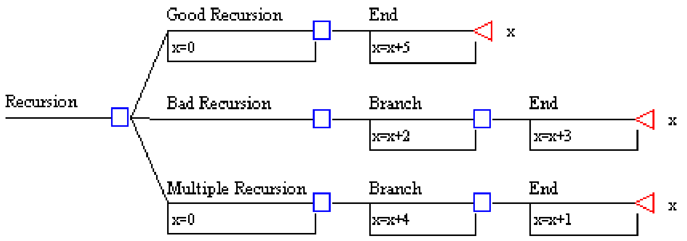
Defining variables recursively

Using recursive (self-referential) variable definitions, a variable can be incremented, decremented, or otherwise modified as events unfold in the tree. This is possible with any type of variable, although it will be most useful in payoff calculations.

The calculation of recursively defined variables will often be intuitive. Value is simply added to (or subtracted from) a variable at each node representing the value-changing event. However, before using recursive variable definitions in a model that includes variables with standard definitions, be sure that you fully understand the differences in how the two types are evaluated.

When DATA encounters a recursive definition (e.g., x=x+1) while performing a normal, right-to-left search during tree calculation, x is flagged as a recursive variable. To evaluate the recursive variable, the search for x is restarted *one node to the left* of the current node.

Open the model called Recursive Variables to see an example of this process.



The top branch, Good Recursion, is a valid use of a recursive variable definition. When calculating the payoff of the terminal node following the Good Recursion node, the variable x must be evaluated. DATA does not yet know that x is a recursively defined variable, though. A normal, right-to-left search for a definition of x finds the self-referential definition x=x+5 at the terminal node. *For the purposes of the current payoff calculation*, x is now identified as a recursively defined variable.

> **TIP:** Like normal, non-recursive variable definitions, recursive definitions are ignored until DATA requires them for a payoff (or probability) calculation. Tracker modifications, covered in Chapter 29, are the only variable definitions that are automatically evaluated in the left-to-right traversal of a tree.

A right-to-left search for a definition of x is now continued one node to the left, at the Good Recursion node. There, the definition x=0 is found and the search is complete. Select the Good Recursion node, and choose Analysis > Expected Value. The calculated value should be 5.

A numeric (non-recursive) value of a recursive variable must eventually be found; definitions can't be infinitely recursive. Look at the second branch, Bad Recursion. In this case, when x is evaluated, DATA locates the recursive definition x=x+3. A recursive search is started one node to the left, at the Branch node, where the definition x=x+2 is found. The search is continued at the Bad Recursion node, but no numeric definition of x is found there or at the root node. If DATA tries to calculate the Bad Recursion subtree, an error message is shown.

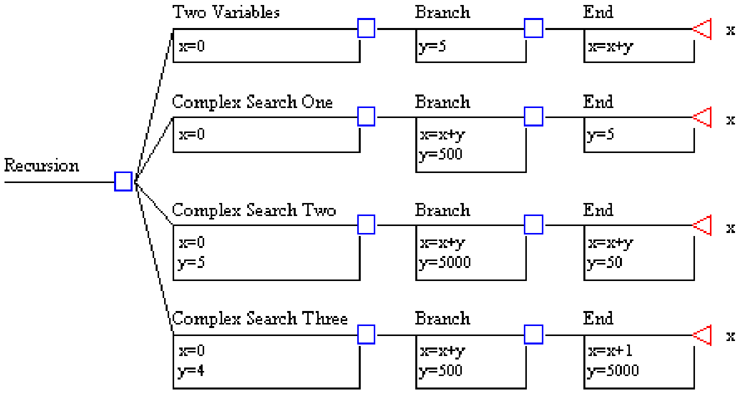
Try this by selecting the Bad Recursion node and calculating its expected value, using the Analysis > Expected Value command.

A multiple-recursion, using a series of recursive definitions of a variable, will work, as shown in the Multiple Recursion subtree. Simply ensure that a numeric definition of the recursive variable will eventually be found. In this case, a recursive search locates the definition x=0 at the Multiple Recursion node. Thus, x=0+4 at the Branch node and 0+4+1 at the End node, resulting in the payoff x=5.

Note that DATA's standard, non-recursive variable definitions will override recursive definitions. Thus, if the first definition DATA locates in a right-to-left variable search is not recursive, DATA will ignore any recursive definitions further to the left.

Complex recursion

Other variables may be used in a recursive definition. Open the Complex Recursion tree to see how this is done.



Calculating the payoff of the Two Variables subtree, the recursive definition x=x+y is found. Before a recursive search for x is started, the variable y is evaluated *as a normal variable*. The search for y is started *at the node being calculated* (here, the terminal node), not necessarily at the node where the original, recursive definition was found.

A normal right-to-left search travels from the terminal node to the node called Branch, where the definition y=5 is found. The recursive search for x is then initiated one node to the left of the initial recursive definition, at Branch. Calculating an expected value for the Two Variables subtree returns the value 5.

When evaluating normal (non-recursive) variable definitions, DATA will always use the definition closest to the node being calculated. However, in a recursive search, different rules apply. Understanding these rules will help avoid making variable definitions at nodes that are skipped during a recursive search.

Examine the Complex Search One subtree. When x is evaluated, the standard search travels to the Branch node, where the initial recursive definition x=x+y is found. The variable y is evaluated as a normal variable, using the definition y=5 found at the terminal node being calculated. The definition y=500 will be ignored.

However, the variable y will not always behave like a normal variable when referenced in a recursive definition – only when it occurs in the initial recursive definition.

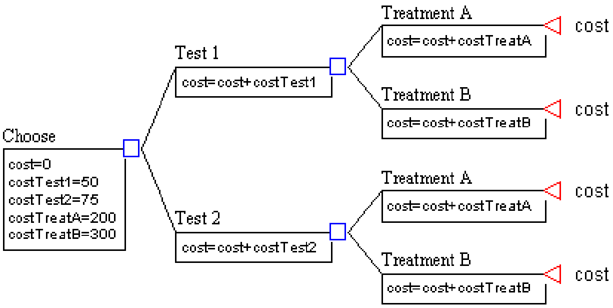
In Complex Search Two, a recursive definition x=x+y is found at the terminal node. A normal search for y finds the definition y=50 at the terminal node. The recursive search for x starts at the Branch node, where the second occurrence of the definition x=x+y is found. Since this definition, which includes the variable y, was found during a recursive search, y is not treated as a normal variable. Recursive searches for x and y start to the left, at the Complex Search Two node, where value definitions are found for both. The expected value calculation for this subtree will return the result 55; make sure you understand why.

In Complex Search Three, y does not occur in the initial recursive definition, so it is never evaluated as a normal variable, and the definitions y=5000 and y=500 will both be skipped in calculations.

Before using complex recursion in your models, it is very important that you become familiar with the logic underlying both standard and recursive variable definitions in DATA.

When to use recursion

Recursive variables are particularly useful for building cost formulas. Rather than using a payoff formula and defining a number of component payoff variables, you can use a single, recursive variable. The variable is modified, as described above, at the appropriate locations in the tree to reflect the accumulating costs.



In the file Cost Recursion Tree, shown in the margin, note that the procedure costs (such as costTest1) are defined at the root node. Recursive definitions reference these variables, incrementing the value of cost by 50 at the Test 1 node, by 75 at the Test 2 node, by 200 at the Treatment A nodes, and by 300 at the Treatment B nodes. Thus, the payoff (the final value of cost) will be different for each scenario.

For another example of recursive variable definitions, open the file Oil Drilling Recursive. The payoff formula has been simplified from the one used in Oil Drilling #2. The variables Cost_Drill and Cost_Test have been replaced by a composite variable, Start_Costs. The variables Drill and Soundings are still used. For each scenario, Start_Costs accumulates these values, as appropriate, using recursive definitions.

CHAPTER 9 VARIABLES – TOOLS AND TECHNIQUES

This chapter covers the tools used to work with variables in decision trees. Chapter 31 covers the use of variables in influence diagrams.

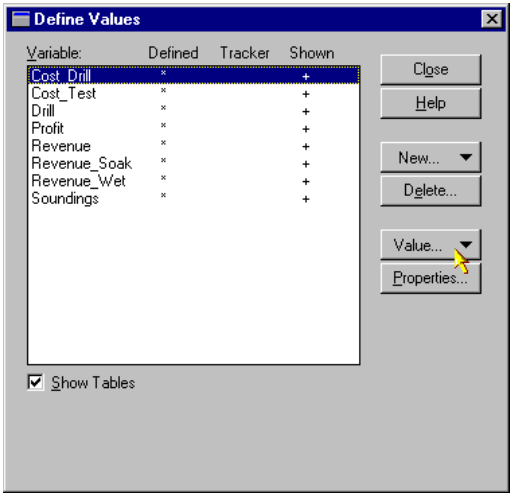
The Define Values dialog

In Chapter 5, variables were created by typing a new variable name in a probability field or a payoff dialog. It is also possible to create (and define) a new variable *before* it is explicitly referenced in a calculated field. Let's see how this is done, using the file Oil Drilling #2.

▷ Select Oil Drilling #2 from the Window menu, if it is currently open. Otherwise, reopen it from the File menu, using either the Open command or the list of recently opened files.

If the tree is open and currently rolled back, roll back display must be turned off before changes can be made to values in the model.

▷ Pull down the Analysis menu; a check mark next to Roll Back indicates that roll back display is active. If necessary, choose Analysis > Roll Back again to turn off roll back display.



Define Values

Variable:	Defined	Tracker	Shown
Cost Drill	×		+
Cost_Test	×		+
Drill	×		+
Profit	×		+
Revenue	×		+
Revenue_Soak	×		+
Revenue_Wet	×		+
Soundings	×		+

☑ Show Tables

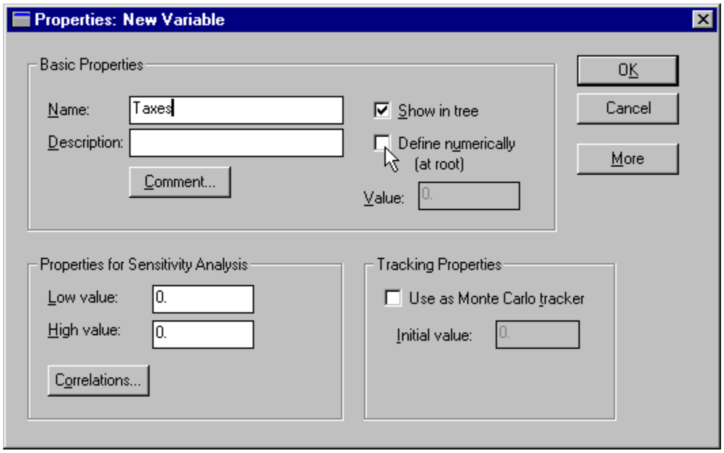
Now, add a new variable to the tree.

► **To create a new variable:**

▷ Select Values > Define Values. A window listing the names of all existing variables will appear.

▷ Click New, and select Variable from the pop-up menu.

▷ In the new variable's Properties dialog, give your variable a one-word name.



Properties: New Variable

Basic Properties

Name: Taxes

Description:

☑ Show in tree

☐ Define numerically (at root)

Value: 0.

Properties for Sensitivity Analysis

Low value: 0.

High value: 0.

Tracking Properties

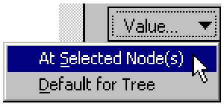
☐ Use as Monte Carlo tracker

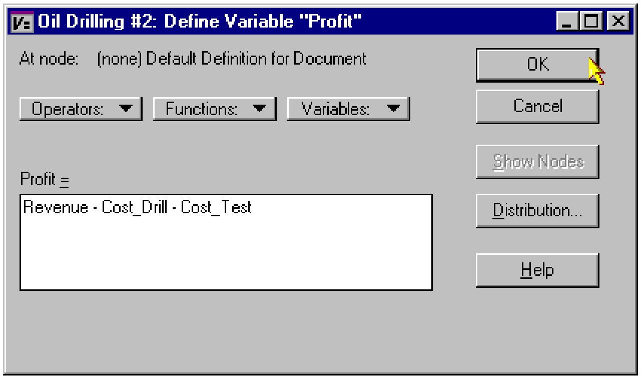
Initial value: 0.

- ▷ Optionally, you may also give it a short description or a longer comment.
- ▷ If you wish to assign a default *numeric* value to this variable, check the box Define numerically, and enter a baseline value. (DATA will automatically define the variable at the root node, and you can skip the steps below for defining a variable.)
- ▷ Press ENTER or RETURN to leave the Properties dialog, and return to the Define Values dialog.

The Define Variable window

Once you have *created* a new variable, you can proceed to *define* it using a Define Variable window. There are a number of ways to open a Define Variable window. One way, illustrated here, is via the Define Values dialog, which you have used previously to create a variable.





Oil Drilling #2: Define Variable "Profit"

At node: (none) Default Definition for Document

Profit =

Revenue - Cost_Drill - Cost_Test

► **To define a variable:**

- ▷ In the Define Values window, select the new variable from the list.
- ▷ Click on the Value… pop-up menu, and choose Default for Tree or, if a node is currently selected in the tree, At Selected Node.
- ▷ In the Define Variable window, enter a number or a formula. Press ENTER or RETURN to store the definition.

While you are in the Define Variable window, you may use DATA's shortcuts to help create a formula. To insert existing variables automatically, simply select the variable from the Variables pop-up menu, and DATA will insert it into the text. Each of the other pop-up menus – Operators, Keywords, Functions, Tables – works similarly. The Functions pop-up menu categorizes DATA's many built-in functions, and includes a Helper to walk you through the steps of properly setting up any function. See Using Functions, below.

TIP: Defining a variable in your tree will not have any effect until you *use* that variable. For example, if you add to your model a new event that has an associated cost, you must both (1) define the cost variable, and (2) include that cost variable in the appropriate payoff formula. DATA does not automatically understand the significance of a variable – you must specifically indicate how that variable is to be used.

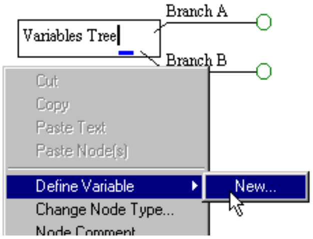
Creating multiple definitions

In the Define Values dialog, which lists all variables in the active tree, you may select more than one variable by holding down the CONTROL key and clicking each variable in turn. This feature makes it possible to define (or delete) multiple variables at the same time.

In addition, you may define one or more variables at more than one node at a time. Simply select the nodes at which you would like to define the variable(s) prior to selecting Values > Define Values.

Using the Quick Menu (Windows)

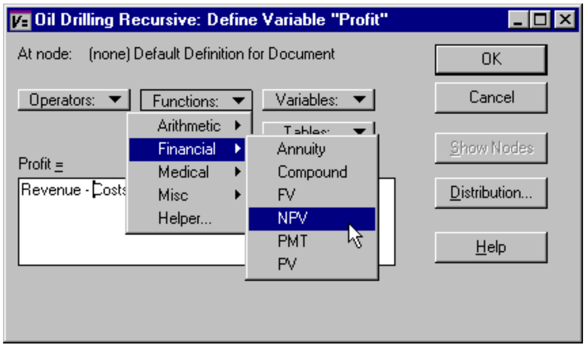
The quickest way to create a variable and define it at a specific node is by right-clicking (CONTROL-clicking) on the node. One of the menu items in the context-sensitive menu is Define Variable. Assign (or modify) an existing variable's definition at the selected node simply by choosing that variable from the sub-menu; you can also define a new variable at that node by selecting New….



Using functions

With DATA's built-in functions, including generic functions like the conditional If() and finance-related functions like NPV() and UtilDiscount(), you can create complex expressions to represent almost any logical or value structure underlying your model. DATA's built-in functions are case-*insensitive* and, in most cases, take arguments. For functions which allow multiple arguments, the arguments must be separated by semicolons (;), rather than commas, because DATA permits the use of commas as place holders in large numbers.

See Appendix C, Functions and Operators, for the full list of functions available in DATA.



Oil Drilling Recursive: Define Variable "Profit"

At node: (none) Default Definition for Document

Profit =

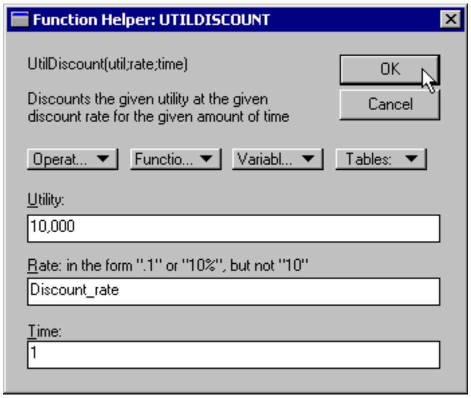
Revenue - Costs

To use a function in a formula, you may simply type it in, or you may select the function from the Functions pop-up menu. If you use the pop-up menu, DATA will automatically add the required parentheses and place the cursor between them, ready for function arguments.

Most functions require only that the proper arguments be assigned. Others, like the Sub() and DistSamp() functions (see Chapters 15 and 28 for details), require additional setup steps before they can be used in calculations.

The Function Helper

Because of the large number of built-in functions with multiple parameters, DATA has a Function Helper dialog (located at the bottom of the Functions pop-up menu) to assist you in setting up complex expressions properly.



Function Helper: UTILDISCOUNT

UtilDiscount(util;rate;time)

Discounts the given utility at the given discount rate for the given amount of time

Utility: 10,000

Rate: in the form ".1" or "10%", but not "10": Discount_rate

Time: 1

Select a function and the Helper will set up the function for you and prompt for the requisite arguments.

Inserting variable names in formulas

In many formula-editing windows, such as the Define Variable window, there are pop-up menus to facilitate the automatic insertion of the listed variables, tables, functions, or keywords.

It is also possible to use the Values > Insert Variable command to insert the name of a variable into a formula.

If you type the first few letters of the variable to be inserted, and select the Insert Variable command, DATA will insert the variable automatically (without the dialog) if there is no ambiguity concerning the variable you want. Otherwise, DATA will limit the list of variables presented to those which begin with the letters you have typed.

TIP: The Insert Variable command can also be used to insert the name of a variable into a probability field (below the line of a branch emanating from a chance, logic, or Markov node).

Modifying and deleting variables and definitions

Once you have defined a variable at a particular node, it is easy to change or remove that particular definition. This functionality also applies to variables defined as default for the tree.

► **To modify an existing definition of a variable:**

- ▷ Select the node at which the variable is currently defined.
- ▷ Choose Values > Define Values.
- ▷ Select the variable whose definition you wish to modify. It should display with a marker in the “Defined” column of the list, indicating that it is defined at the selected node.
- ▷ Click Value..., and choose either At Selected Node or Default for Tree. At Selected Node is available if at least one node is currently selected. Default for Tree is always available when the tree window is active. If the root node is the only node currently selected, At Selected Node and Default for Tree will have the same effect, changing the definition at the root node.

► **To delete a variable from your tree entirely:**

- ▷ Open the Define Values dialog.
- ▷ Select one or more variables in the list, and click Delete.

DATA will warn you before carrying out the deletion. If you proceed with the deletion, all *definitions* of the variable will be removed from the tree. *References* to the deleted variable will not be removed from payoffs, probabilities, and other variable definitions. Unless these formulas are corrected, your model will generate an error message upon calculation. See Chapter 14 for information on using the Find command to search for (and replace) variable references and other text.

You may also delete particular definitions of a variable, while leaving it in the list of variables for future use.

► **To delete a single definition of a variable:**

- ▷ Select the node where the definition exists, and choose Define Values.
- ▷ Select the variable from the list, click on the Values... button, and choose At Selected Node... from the pop-up menu. This will open a Define Variable window for the definition you wish to delete, as if you were planning to edit its formula.
- ▷ Select the entire text of the formula, and press BACKSPACE (Windows) or DELETE (Macintosh). This will delete the text, leaving the editor empty.
- ▷ Press ENTER or RETURN.

DATA will seek confirmation that you wish to remove the definition.

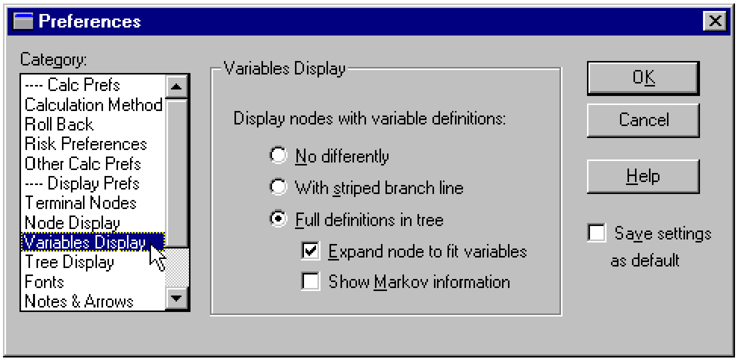
Another alternative to deleting a variable from the tree (and creating a new one in its place), is simply to rename the variable. DATA allows you to change a variable's name and have the change cascade throughout the tree, updating all references in probabilities, payoffs, and variable definitions. This is accomplished via the Properties dialog, discussed later in this chapter.

Variables display

It is possible to modify DATA's display preferences in order to show on the face of the tree the value and location of variable definitions.

► **To display full definitions of all variables in a tree:**

- ▷ Select Edit > Preferences.



Preferences

Category:

---- Calc Prefs
Calculation Method
Roll Back
Risk Preferences
Other Calc Prefs
---- Display Prefs
Terminal Nodes
Node Display
Variables Display
Tree Display
Fonts
Notes & Arrows

Variables Display

Display nodes with variable definitions:

○ No differently
○ With striped branch line
● Full definitions in tree
☑ Expand node to fit variables
□ Show Markov information

□ Save settings as default

▷ Choose Variables Display Preferences.

▷ Click on the option for Full definitions in tree, and, optionally, on the check box for Expand node to fit variables.

▷ Press ENTER or RETURN.

All default definitions are displayed below the root node. Node-specific definitions are below the appropriate nodes. Note that DATA allows you to hide individual variables' definitions from presentation in the tree window. This setting is found in the Properties dialog, discussed below.

Displaying payoff variables and expressions

It is possible to control the display of payoff formulas at visible terminal nodes.

► **To display payoff formulas:**

▷ Select Edit > Preferences.

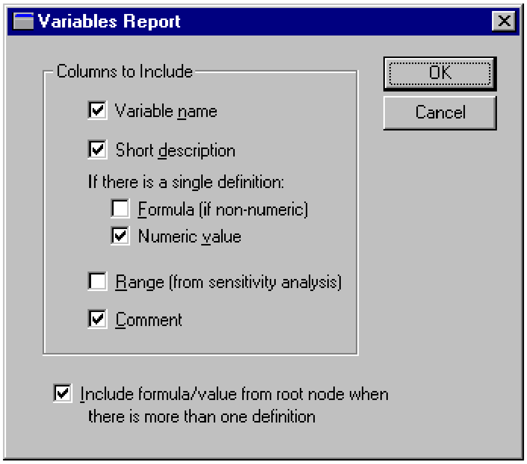
▷ Choose Terminal Node Preferences.

▷ Make sure the box for Display payoff names is checked.

▷ Press ENTER or RETURN.

The Variables Report

Keeping track of the variables in your model can become a challenge as the tree grows larger and the variable expressions more complex. The Variables Report will enumerate all the variables in your tree, along with their descriptions, comments and, in certain cases, formulas and numeric values.



Columns to Include

☑ Variable name

☑ Short description

If there is a single definition:

☐ Formula (if non-numeric)

☑ Numeric value

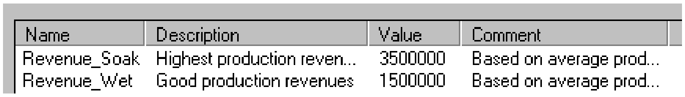
☐ Range (from sensitivity analysis)

☑ Comment

☑ Include formula/value from root node when there is more than one definition

To access the report, choose Values > Report > Variables.... The Variables Report dialog allows you to specify what information is included in the report. For example, you may want to hide the variable's name and use the short description to identify the variable. You also might choose to exclude from the report the definitions of variables, based on whether the definition is by formula or numeric value. If a variable has more than one definition in the tree, no formula or value will be reported unless two conditions are met: 1) one of the definitions is located at the root node; and 2) the check box Include formula/value from root node... is selected in the Variables Report dialog.

The report is displayed initially in a tabular format; it can then be exported to a word processor or spreadsheet document.



Name	Description	Value	Comment
Revenue_Soak	Highest production reven...	3500000	Based on average prod...
Revenue_Wet	Good production revenues	1500000	Based on average prod...

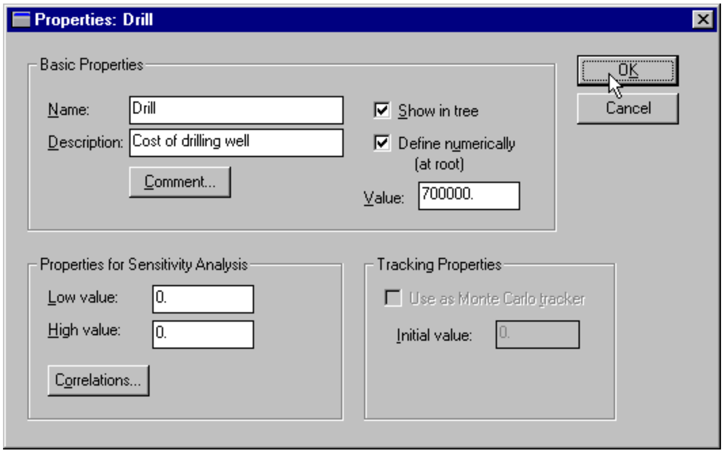
The Properties dialog box

The Properties dialog box appears whenever you create a new variable using the Define Values dialog, or when DATA creates one for you on the fly. You will also see this dialog when you select an existing variable and click the Properties button in the Define Values dialog.

A variable's properties can be classified into three groups: basic properties, properties for sensitivity analysis, and tracking properties.

Basic properties

Each variable has three basic text properties: a name, a short description, and a long comment. It is also possible to specify a default, numeric definition of the variable.



Properties: Drill

Basic Properties

Name: Drill

Description: Cost of drilling well

☑ Show in tree

☑ Define numerically (at root)

Value: 700000.

Properties for Sensitivity Analysis

Low value: 0.

High value: 0.

Tracking Properties

☐ Use as Monte Carlo tracker

Initial value: 0.

The variable’s name is how you will refer to the variable in formulas. Names must conform to the following conventions:

- the name must begin with a letter or an underscore character;
- the name may contain only letters, numbers, and underscore characters; and
- the name may not be longer than 32 characters.

The optional short description is used primarily in graphical analysis output. For example, in the absence of a short description for a variable named p_Up, the title of a graph window would read, “Sensitivity Analysis on p_Up.” If, however, you were to assign the short description Probability Market Will Rise to the variable p_Up, the graph title would read, “Sensitivity Analysis on Probability Market Will Rise.” The short description is also used by the Custom Interface; see Chapter 18.

The long comment is for your own use. It can be used to hold notes that memorialize a variable’s meaning or sources. If you enter a comment, it is shown at the bottom of the window when you are defining that variable, beneath the formula-entry box. It is also displayed at the bottom of the Define Values dialog box when that variable is selected.

The basic properties group also includes a check box called Show in Tree. If you leave this box checked, DATA will display all definitions of this variable when you choose the option in the Preferences dialog to show full variable definitions. If you clear the check box, the variable and all its definitions will be hidden when you choose to show variable definitions in the tree. The variable will, however, appear in other variables’ definitions if those variables are set to display in the tree. Clicking the Show in tree check box will not have any effect unless you choose generally to show variable definitions in the tree.

Finally, you can quickly and easily create a default numeric definition for the variable, to be stored at the root node. Select the Define numerically check box, and enter a number in the Value editor. This is the quickest way to assign a numeric value to a variable. The functionality is exactly the same as selecting a variable in the Define Values dialog, and choosing Values…Default for Tree.

As suggested above, instead of deleting an existing variable, you may want simply to rename the variable throughout the tree. By changing a variable’s Name property, you can do this very quickly.

► **To rename a variable:**

▷ Choose Values > Define Values, and select the variable you wish to modify.

▷ Click Properties..., and simply change the text in the Name field.

▷ Press ENTER or RETURN when you are finished.

The change will be propagated to all formulas in the tree.

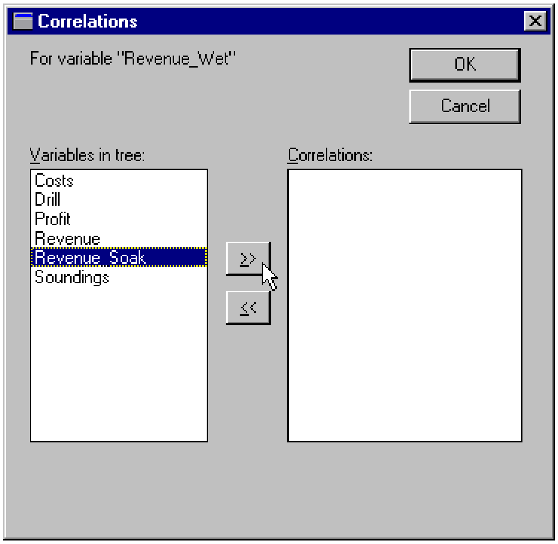
Properties for sensitivity analysis

The low and high values which you may enter in the Properties dialog box are used for suggesting a range for use in sensitivity analyses. The suggested range can be accepted or overridden when specifying the parameters of the sensitivity analysis. It is not necessary to enter values here, although you may find it useful. No additional definitions of the variable are created from these values. The high and low values entered here also define the default range for the variable if it is correlated with a variable selected for sensitivity analysis.

Correlations can be defined among any number of existing variables. When a sensitivity analysis is performed on a variable with correlations, the user has the option of simultaneously varying any or all of the correlated variables over a range. From the sensitivity analysis Correlations dialog, it is possible to specify which correlations should be active during the analysis. It is also possible either to accept the default range or specify a different range of values for each correlated variable. The same number of intervals will be used for each variable in the sensitivity analysis.

For more details on carrying out sensitivity analysis on correlated variables, see Chapter 22.

► **To define variable correlations:**


Correlations

For variable "Revenue_Wet"

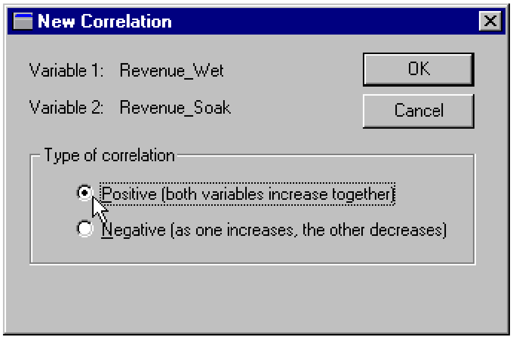
Variables in tree:
- Costs
- Drill
- Profit
- Revenue
- Revenue_Soak
- Soundings

Correlations:

▷ Click on the Correlations... button in the Properties dialog.

▷ In the Correlations dialog, select a variable from the list on the left, and click the >> button. (Note that the variable whose properties you are editing does not appear on the list.)

▷ In the New Correlation dialog, select the type of correlation, positive or negative, and click OK.


New Correlation

Variable 1: Revenue_Wet

Variable 2: Revenue_Soak

Type of correlation

● Positive (both variables increase together)

○ Negative (as one increases, the other decreases)

▷ In the Correlations dialog, the correlated variable now appears in the list on the right, with a plus or minus symbol indicating the type of correlation.

Correlations... +Revenue_Soak

Once created, the identical correlation exists in the properties of both variables. The names of correlated variables, along with their correlation type (+ or -), will be visible in each variable's Properties dialog, next to the Correlations… button. The correlation of any pair of variables can be modified (or removed) from either variable’s Properties dialog. Any number of correlations can be created.

To remove a correlation, return to the Properties dialog for either of the correlated variables and click on the Correlations... button again. Select the correlated variable from the list on the right, and click the << button to remove it from the list. To change a correlation’s type (e.g., from negative to positive), you must remove the existing correlation and recreate it with the proper correlation.

TIP: DATA does not request the degree of correlation (i.e., the r-value). You must manually assign the ranges for all correlated variables during sensitivity analysis.

Tracking properties

A variable which is used as a Monte Carlo tracking variable acts very differently from an ordinary variable. These variables, known as tracker variables, are useful in trees specifically designed for Monte Carlo simulation, especially Markov models. See Chapter 29 for more information.

Unless you are familiar with the operation of tracker variables, you should leave the Monte Carlo tracker box unchecked. Once a variable has been defined as a tracker, it should not be used as a normal variable.

The Variables Window

The Variables Window provides a glimpse into the value structure of a tree, enabling you to see where (at which nodes) and with what value (or in terms of what formula) variables have been defined. It also provides a convenient method for adding and updating definitions.

Opening the Variables Window

You saw how to display the full definitions of variables on the face of a tree. With variables displayed, double-clicking below a branch line in the variables list box will open the Variables Window. Definitions can be easily added, modified, and deleted using this window.

The Variables Window can also be opened by selecting it from the Values menu or by clicking on the toolbar button. This provides an alternative means of viewing and editing variable definitions which avoids cluttering and enlarging the tree window.

▷ Open the file Oil Drilling #2.

▷ Select the Wet node in the No Soundings subtree.

▷ Select Values > Show Variables Window.

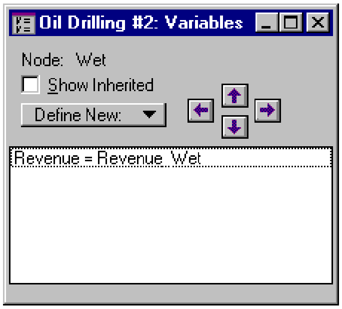
If your tree window is not maximized, a small window will appear near the bottom of the DATA window. If your tree window is maximized, the Variables Window will also fill the entire DATA window.

> **TIP:** It is possible to see some or all open windows on the screen, simultaneously, by resizing windows. Whenever DATA for Windows starts up, document windows are automatically maximized, leaving only the front window is visible. In order to see more than one window on your screen, simply click the small restore button found between the minimize and close buttons, in the right corner of the document window. This feature is useful when editing values (e.g. payoffs, variable definitions), allowing you to see both the tree and the editor windows as you work.

Understanding the Variables Window display

The Variables Window lists the variables defined at the selected node and their values (definitions). At the Wet node in the No Soundings subtree, only the variable Revenue has been defined, equal to Revenue_Wet. Thus, the window contains only the expression

Revenue = Revenue_Wet.



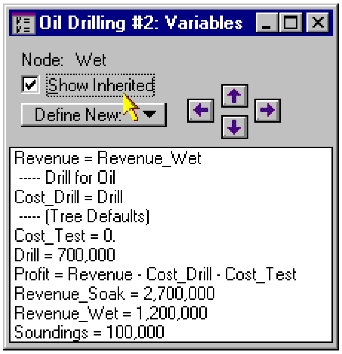
Node: Wet

☐ Show Inherited

Revenue = Revenue_Wet

It is also relevant to know what other definitions of variables may apply at the selected node on the basis of definitions at nodes further to the left. To see these values,

▷ Click on the Show Inherited checkbox.



Node: Wet

☑ Show Inherited

```
Revenue = Revenue_Wet
 ----- Drill for Oil
Cost_Drill = Drill
 ----- (Tree Defaults)
Cost_Test = 0.
Drill = 700,000
Profit = Revenue - Cost_Drill - Cost_Test
Revenue_Soak = 2,700,000
Revenue_Wet = 1,200,000
Soundings = 100,000
```

Beneath the expression Revenue=Revenue_Wet, appears a dotted line. Beneath that line is a listing of the variables defined at nodes to the left

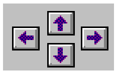
– in this case, the Drill for Oil node and the root node – together with their definitions. The list of definitions at a particular node to the left is shown beneath a title bearing the name of that node.

Keep in mind that definitions at one node can be overridden by definitions of the same variable at a node further to the right (closer to the terminal node). The Variables Window will show only the first definition found in the right-to-left search, starting at the active node.

Changing the node selection

It is possible to select one node after another, watching the Variables Window to confirm that all variables have been correctly defined. Using the mouse to change the selected node will, however, activate the Tree Window, causing it to cover the Variables Window.

There are three ways to avoid this problem. One is to resize and move the Tree Window so that, upon being activated, it will not cover the Variables Window. Second, it is possible to bring the Variables Window to the front by pulling down the Window menu and choosing [name of active file]: Variables.

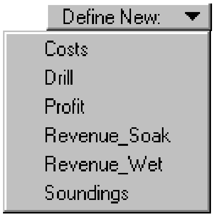


The third, most useful, option involves clicking on the navigation arrows in the corner of the Variables Window. The navigation arrows work in the same way that cursor keys do in the tree, but without the problem of activating the Tree Window and causing it to cover the Variables Window. Each click on one of the navigation arrows will move the selection one node in the direction of the arrow.

If multiple nodes are selected concurrently, the title line in the Variables Window will read “Node: (multiple),” and the remainder of the window will be empty. The Variables Window identifies the variables defined only at a single, selected node, so the feature becomes inoperable when more than one node is selected.

Updating and adding variable definitions

You need not return to the Tree Window to change the definition of a variable. Double-clicking inside the Variables Window on the name or definition of a variable will bring up the standard dialog box for defining that variable at the active node. To create a *new* definition at the selected node of a variable not yet defined at that node, select the variable from the Define New pop-up menu.



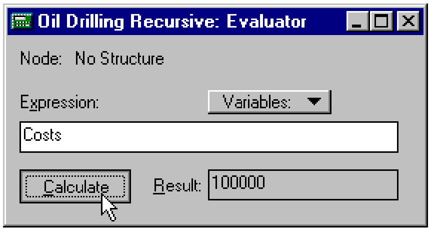
With the Variables Window active, it is possible to create new variables using the Define Values dialog. Simply click on the toolbar button or choose Values > Define Values.

The Evaluator

When building or modifying a tree, it is often helpful to know the value of a variable or expression as calculated at a given node. The Evaluator command is designed expressly for this purpose.

Simply select a node and open the Evaluator from the Values menu. In the Evaluator dialog box, enter an expression whose value you want to know, and DATA will calculate the value of that expression at the selected node. Calculations will be based on the first definitions found in the right-to-left search, starting at the active node.

Assume you are working with the example file Oil Drilling Recursive and want to see the value assigned to the variable Costs at the No Structure node in the Seismic Soundings subtree:



Node: No Structure

Expression: Costs

Result: 100000

▷ Select the No Structure node.

▷ Select Values > Open Evaluator.

▷ Type `Costs` in the editor in the Evaluator window. (Alternatively, select the variable from the Variables pop-up menu.)

▷ Click on the Calculate button.

The Evaluator behaves exactly as the variable parser would during expected value calculations. The result, 100000, reflects both (1) the recursive definition of the variable Costs at the Seismic Soundings node, where it is defined in terms of itself and one other variable,

Costs = Costs + Drill;

and (2) the default numeric values assigned to these variables at the root node,

Costs = 0 and

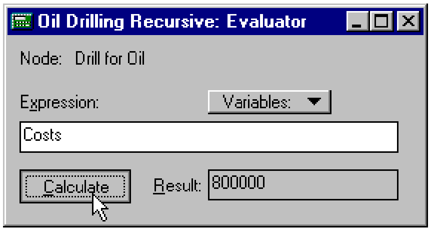
Drill = 100,000.

Like the Variables Window, the Evaluator can remain open while you change the node selection. Since the Evaluator calculates the entered expression at the currently selected node, changing the selection in the tree window may cause a different result to be calculated.

▷ Now, click on the Tree Window, and select the Drill for Oil node to the right of the currently selected No Structure node.

▷ Reselect the Evaluator.

While the node name has been updated, the calculated value has not been automatically updated.


Oil Drilling Recursive: Evaluator

Node: Drill for Oil

Expression: Costs

Result: 800000

▷ Click on the Calculate button.

This time, the calculated result is 800000. This reflects the fact that at the Drill for Oil node, the additional cost of drilling is incurred, and the variable Costs accumulates the value of Drill, assigned a value of 700,000 at the root node.

► **To close the Evaluator:**

▷ Select File > Close, or press CTRL-F4 (Windows) or COMMAND-W (Macintosh).

Sliders

A one-way sensitivity analysis will automate multiple recalculations of a model across a range of values assigned to a single variable. The result is a line graph – which can be converted into a strategy graph – showing the effect on expected value for each of the illustrated scenarios.

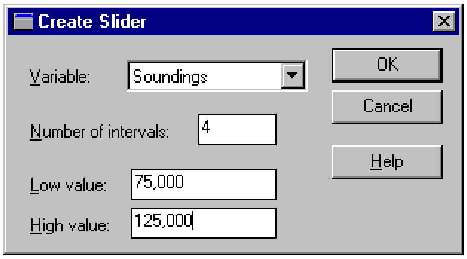
Sometimes it is desirable to view more detailed results at each interval of a sensitivity analysis. Because the output of the line graph is limited to specific numeric quantities (primarily expected values), you might find the slider helpful in more complicated analysis situations:

- You may wish to view how the distribution of values in a Monte Carlo simulation varies with deterministic changes in a (non-distributed) parameter.
- If you want to view a series of one-way sensitivity analyses on one variable while *manually* changing the value of another variable, to see a form of two-way sensitivity analysis.
- To see how changing the value of a variable affects the probability of being in a particular state at the conclusion of a Markov process.

A tool called a slider makes it possible to perform a set of these recalculations without having to go into the tree to redefine the value in question each time. However, just as sensitivity analysis can be performed only if the value in question has been defined as a variable, a slider works only with variables.

► To use a slider:

- ▷ Select the node at which the variable in question has been defined.
- ▷ Select Values > Open Slider.



Create Slider

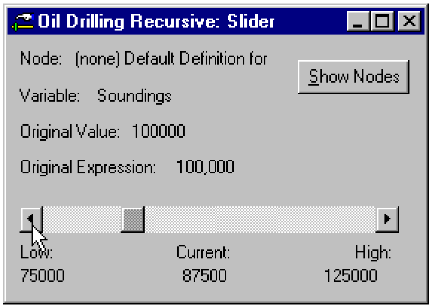
Variable: Soundings

Number of intervals: 4

Low value: 75,000

High value: 125,000

- ▷ In the resulting dialog box, select the appropriate variable and specify the value range (low and high values) and the number of intervals.
- ▷ In the Slider window, move the thumb of the sliding bar to change the value of the variable within the specified range.



Oil Drilling Recursive: Slider

Node: (none) Default Definition for

Variable: Soundings

Original Value: 100000

Original Expression: 100,000

Low: 75000

Current: 87500

High: 125000

With the slider active, you can perform any analysis on the model (including roll back), and the value of the specified variable at the selected node will be as specified in the slider. This temporary value will apply until the slider is closed or a sensitivity analysis is performed on the chosen variable. More than one slider can be opened and activated simultaneously, for different variables and different nodes.

If you click the Show Nodes button, DATA will activate the Tree Window and select the node on which the slider is operating.

Chapter 10 Customizing DATA’s Display

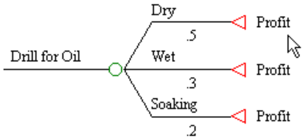
DATA's strengths as a modeling tool are not limited to quantitative analysis; DATA also provides highly visual methods for organizing and communicating decision problems. This chapter covers a number of features useful for customizing the on-screen and printed display of decision trees and influence diagrams.

Values display

Each tree and influence diagram you create with DATA can be given its own distinct set of values display preferences. To change these settings for a particular model (or to change your default settings), you will generally use the Preferences Dialog.

Displaying payoff names

It is often useful to be able to see what payoffs have been assigned. DATA makes it possible to identify the active payoff at each terminal node, even when roll back is turned off. With a tree on the screen,



▷ Select Edit > Preferences, and choose the Terminal Nodes page.

▷ Click Display payoff names.

▷ Press ENTER or RETURN.

The text of the payoff appears to the right of each terminal node where a payoff is assigned. Note that the value is not calculated and displayed until you roll back the tree. If you wish to enclose the payoff names in boxes,

▷ Select Edit > Preferences, and choose the Node Display page.

▷ Click Boxed.

▷ Press ENTER or RETURN.

Terminal Node Columns

Following roll back, it is possible to display information concerning each scenario in your tree in columns which appear to the right of the endnodes. The information contained in these columns can be exported to the clipboard, using the Edit > Copy Special... command, and then pasted into a spreadsheet or text document.

For instance, you can display on the face of the tree:

- individual components of a payoff formula, such as a formula used to calculate the total costs of a process;
- any part of a cost-effectiveness report, including marginal values; or
- the value of any custom formula calculated for each scenario.

► **To set up the columns for a tree:**

▷ Choose Edit > Preferences, then select the Terminal Nodes page.

▷ Click the show columns check box, and click the Set button.

▷ In the Columns dialog, create the desired columns.

Each column has the following information:

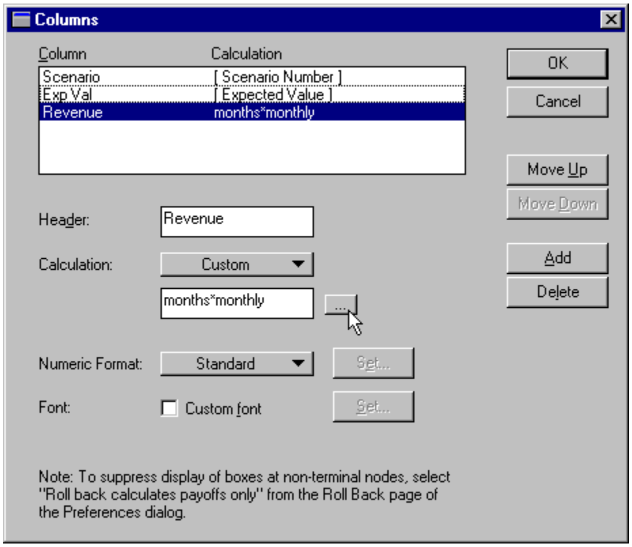
Header – Title displayed above the column in the tree window.

Calculation – You can display (a) the results of any calculations customarily performed by DATA during roll back, including average or marginal expected values, or path probability; (b) the results of a custom calculation formula evaluated at each visual end node; (c) the scenario number (see Terminal Node Numbers, below); and (d) the value of any valid variable at each visual end node.

Numeric Format – Each column can use one of the numeric formats stored in the tree, a custom numeric format, or no formatting.

Font – You can specify a custom font for each column. Otherwise, columns will use the same font that is used in the rollback boxes.

Any column settings you specify will be stored with the tree.



Column	Calculation
Scenario	[Scenario Number]
Exp Val	[Expected Value]
Revenue	months*monthly

Header: Revenue

Calculation: Custom
months*monthly

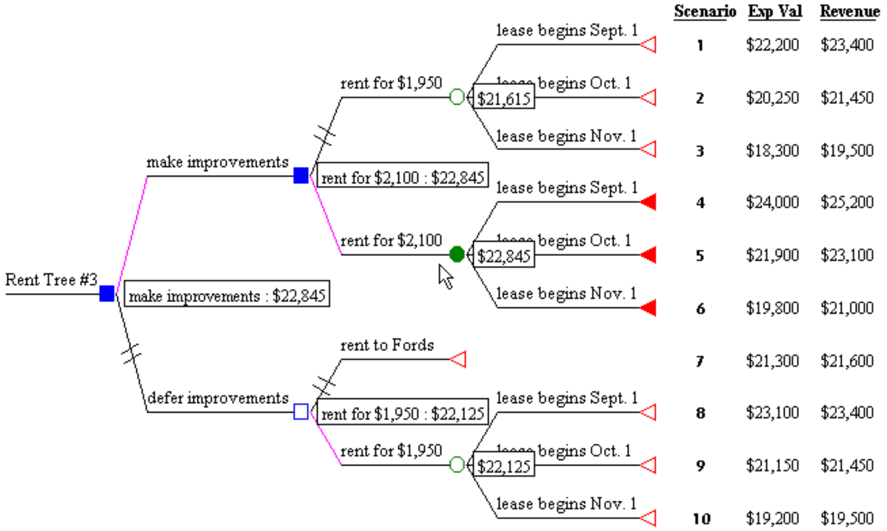
Numeric Format: Standard

Font: □ Custom font

Note: To suppress display of boxes at non-terminal nodes, select "Roll back calculates payoffs only" from the Roll Back page of the Preferences dialog.

- ▷ Close the Columns dialog and Preferences dialog, and choose Analysis > Roll back, to see the tree with columns.

To see an example set of columns, open the example file Rent tree.



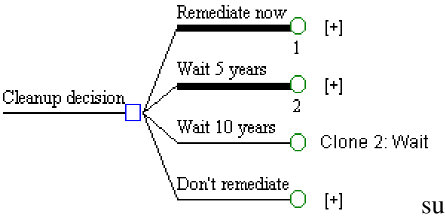
Scenario	Exp Val	Revenue
1	$22,200	$23,400
2	$20,250	$21,450
3	$18,300	$19,500
4	$24,000	$25,200
5	$21,900	$23,100
6	$19,800	$21,000
7	$21,300	$21,600
8	$23,100	$23,400
9	$21,150	$21,450
10	$19,200	$19,500

Column information has already been set up for this tree; however, if you roll the tree back the columns will not display. To turn on columns display for the tree, you must make a change in the tree preferences. Choose Edit > Preferences, and select the Terminal Nodes page. At the bottom of the dialog is the Show columns check box; select the check box to activate the roll back display of columns. Next to the check box is the Set… button; click the button to see the columns information for this tree.

When you are done, close the Terminal Node Columns and Preferences dialogs, and return to the tree. Roll back the tree to see how columns are displayed.

> **TIP:** If you wish to display only terminal node columns in a rolled-back tree, you can suppress the display of roll back boxes and expected value information at non-end nodes. Open the Roll Back page of the Preferences dialog and select Roll back calculates payoffs only.

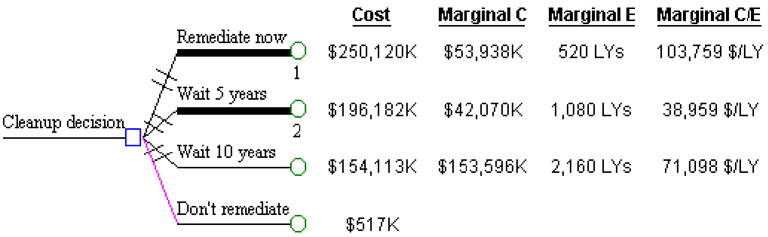
You may display any variable in a custom column, even variables that are not used in calculating the active payoff(s). The variable (or expres-



sion) can be directly entered into the Custom text area; clicking on the ellipsis button to the right will bring up an expression editor dialog.

Columns will display at all *visual* endnodes, even if they are not terminal nodes. This means that you can show columns at collapsed subtrees and hidden clone copies (see Chapter 12). The graphic below shows marginal value columns at the two collapsed subtrees.

Marginal values will be displayed in a column only at the immediate descendants of a decision node. If the endnode is further to the right, this column will be blank. See Chapter 21 on cost-effectiveness.



	Cost	Marginal C	Marginal E	Marginal C/E
Remediate now	$250,120K	$53,938K	520 LYs	103,759 $/LY
Wait 5 years	$196,182K	$42,070K	1,080 LYs	38,959 $/LY
Wait 10 years	$154,113K	$153,596K	2,160 LYs	71,098 $/LY
Don't remediate	$517K			

Use the Collapse Subtree command to hide subtrees and, thus, specify the appropriate endnode.

Columns will be vertically aligned even if the end nodes are not themselves aligned. If your columns do not display properly, check to see if the minimize empty space setting in the Tree Display page of the Preferences dialog is selected. If so, deselect it to cure the problem.

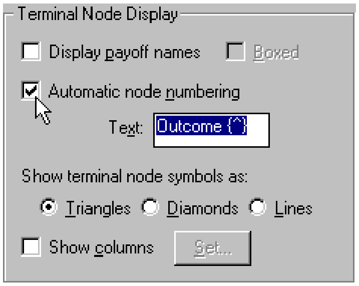
If you specify a custom numeric format for a column, that setting is maintained separately from the general settings utilized during roll back display and for other analyses.

If you wish to show terminal node (scenario) numbers in a column, you must also activate the automatic node numbering setting found on the Terminal Node page of the Preferences dialog.

While roll back is on, you can choose Edit > Copy Special to export the columns to a spreadsheet.

Terminal node numbers

You may choose to have all terminal nodes automatically numbered. To enable this feature, go to the Terminal Nodes page of the Preferences dialog. Select the check box “automatic node numbering” and enter the text to be used for node numbering. You must use the caret (^) in the



Terminal Node Display

☐ Display payoff names ☐ Boxed

☑ Automatic node numbering

Text: Outcome {^}

Show terminal node symbols as:

● Triangles ○ Diamonds ○ Lines

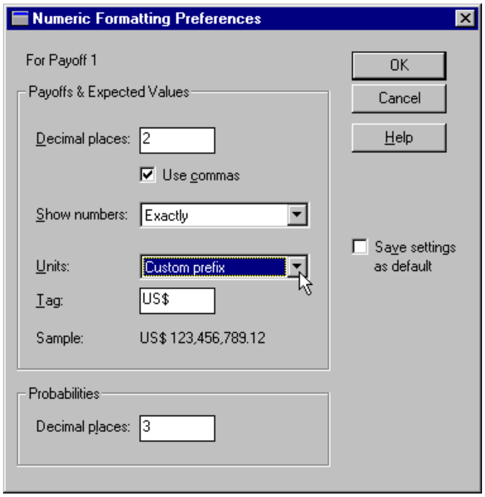
☐ Show columns

text as a placeholder for the node number. The caret can be used alone or with additional text, such as Outcome ^.

If activated, the terminal node numbering text you specify will always be displayed immediately to the right of visible terminal nodes. If you want to display scenario numbers in roll back terminal node columns, instead, you still must select the automatic node numbering check box. To suppress display of node numbering while roll back is off, leave the Text box empty.

Numeric Formatting

DATA's display of calculated quantities is governed by the settings in the Numeric Formatting dialog box. Although internal calculations are always performed at the highest available precision, you may choose to display output values using a limited number of decimal places. Unit symbols (such as “$” or “customers”), can also be set to display next to calculated values. In most cases, display formatting can be changed even after the analysis is performed.



Numeric Formatting Preferences

For Payoff 1

Payoffs & Expected Values

Decimal places: 2

☑ Use commas

Show numbers: Exactly

Units: Custom prefix

Tag: US$

Sample: US$ 123,456,789.12

Probabilities

Decimal places: 3

☐ Save settings as default

The phrase “For Payoff 1,” which appears at the top of the dialog box, indicates that payoff 1 is the currently active payoff and that the currently displayed settings apply to it. For more information about using multiple payoffs for multi-attribute analysis, see Chapter 20.

The large upper box enables you to change the numeric format of calculated payoffs and expected values upon roll back.

For payoffs and expected values, you can specify: the number of decimal places (0-9) that should be displayed after the decimal point; whether to incorporate thousands separators into large numbers; whether to display numbers exactly or in thousands, in millions, or in billions; and what units of measure to use.

Numbers displayed in thousands are followed by K; numbers displayed in millions are followed by M; and numbers displayed in billions are followed by B. For example, 10,400 will display as 10.40K when you opt to display numbers in thousands and with two decimal places.

If you choose currency as the unit of measure, DATA will display expected values using the type of currency set in your operating system’s control panel. For example, in the United States, the number 45 will display as $45, and the number -45 will display as ($45). In

Japan, 45 would display as ¥45. If you choose none, numbers will be displayed as unitless quantities, such as 45.

You may also enter units of your choice, by entering them in the Tag editor which appears when you select the Prefix and Suffix options. These units will be displayed before (Prefix) or after (Suffix) the calculated value. For example, a medical analyst who wishes to show results in quality-adjusted life-years should choose Suffix, and enter QALY as the unit. The resulting display would be 45 QALY.

In the lower box you may change the numeric format of probabilities upon roll back. You can specify the number of decimal places (0-9) that should be displayed after the decimal point for calculated probability values.

Very small or very large numbers may automatically be displayed using scientific notation.

> **TIP:** The international settings specified in your operating system dictate the appropriate character to use as a decimal separator. Be sure to follow this convention when inputting values.

Roll back display options

When a tree is rolled back, there are three additional options available for customizing the display of calculated output: displaying probabilities as numeric equivalents, displaying expected value boxes at a subset of the nodes in the tree, and moving expected value boxes. To test these options, roll back the New Variables tree.

► **To display numerically those probabilities defined as variables:**

▷ Select Edit > Preferences, and choose the Roll Back page.

▷ Click Display probabilities as numeric equivs.

▷ Click OK.

When this option is selected, quantities below the branch line will always display numerically during roll back.

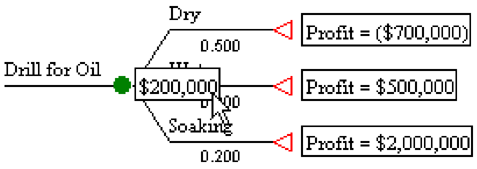
► **To display expected value boxes only at terminal nodes, decision nodes, and branches of decision nodes:**

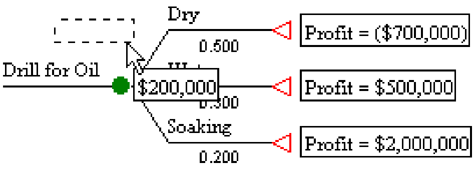
▷ With roll back turned off, select Edit > Preferences, and choose the Roll Back page.

- ▷ Click Display EV at terminal and decision nodes and options only.
- ▷ Click OK, and roll back the tree.

► **To move individual expected value boxes (with the tree rolled back):**

- ▷ Hold down the CONTROL key (Windows) or OPTION key (Macintosh).
- ▷ Click on one of the expected value boxes, and drag it to a better location.
- ▷ Release the mouse and CONTROL key or OPTION key.





The new location of the expected value box will be preserved when the tree is saved.

► **To hide individual expected value boxes:**

- ▷ While roll back is still on, reveal the Quick Menu (described in Chapter 14) by right-clicking (Windows) or CONTROL-clicking (Macintosh) on a node.
- ▷ Select the Hide roll back box option.

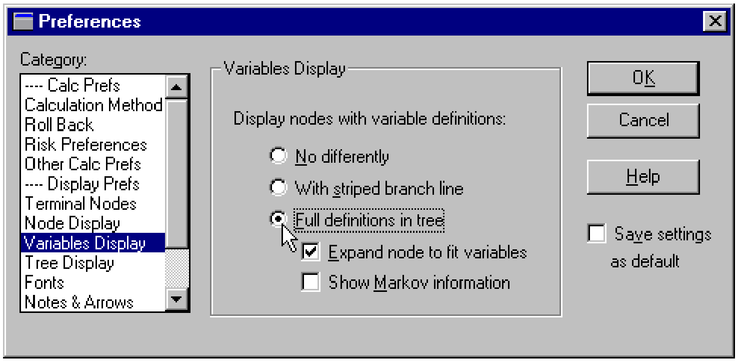
The expected value box associated with the selected node has disappeared. This information is saved with the tree.

Display of the expected value box can be reinstated by right-clicking (or CONTROL-clicking) the node and unchecking Hide roll back box.

Variables display

In the Variables Display page of the Preferences dialog box, you can specify whether (and how) the definitions of variables are to be reflected in the tree window.

Three options are available. Each relates to whether tree display should identify nodes at which variables have been defined and, if so, how.



Category:

---- Calc Prefs
Calculation Method
Roll Back
Risk Preferences
Other Calc Prefs
---- Display Prefs
Terminal Nodes
Node Display
Variables Display
Tree Display
Fonts
Notes & Arrows

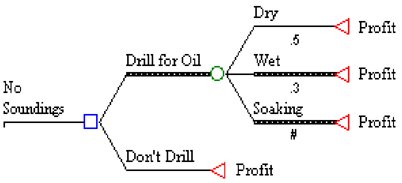
Variables Display

Display nodes with variable definitions:

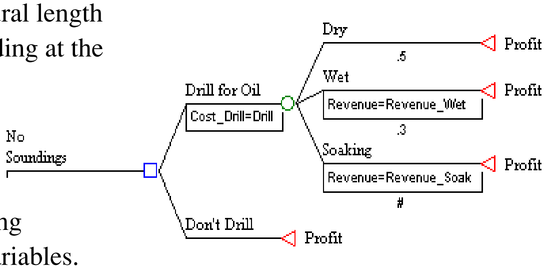
○ No differently
○ With striped branch line
● Full definitions in tree
☑ Expand node to fit variables
□ Show Markov information

□ Save settings as default

When With striped branch is chosen, the branch line preceding each node at which one or more variables have been defined will be drawn in a striped pattern. This option affects only the on-screen display; it does not affect the appearance of the printed or exported tree.



When Full definitions in tree is selected, at each node where one or more variables have been defined each definition will be displayed in a box beneath the node name. To limit interference with tree geometry, take care in applying this option to trees in which many variables are defined at a single node other than the root node. Long definitions will be clipped to the natural length of the branch line ending at the node. You can cause the branch line lengths to expand to fit the definitions by checking Expand node to fit variables.



You can display Markov rewards, tolls, and termination conditions (described in Chapter 27) at the relevant nodes by checking Show Markov information.

Note that it is possible to specify that certain variables are not to be displayed on the face of the tree. See Chapter 9.

You can also set the font used to display the definitions by clicking on the Variables Font... button in the Fonts page of the Preferences dialog box.

If you prefer not to visually identify nodes at which variables are defined, choose No differently.

Hiding values

t is possible to turn off the display of the probability field in your tree, either before or after you enter probabilities. Select Edit > Preferences, and choose the Node Display page. If you wish to turn off the display of both the branch descriptions and the probability field text, select the Hide all node texts option. To turn off the display of probabilities only, select the Hide probabilities only option.

Tree structure

DATA offers a variety of display features that make it possible to clarify the meaning of and relationship between events.

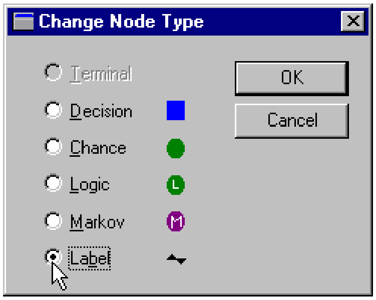
Label nodes

A label node acts like a placeholder. Label nodes have no analytical function, but can be employed to identify more clearly all of the steps in a particular scenario.

► **To create a label node:**

▷ Insert a new branch at the desired location.

▷ Select Options > Change Node Type..., click the Label button, and click OK.


Change Node Type

○ Terminal

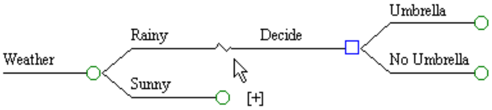
○ Decision

○ Chance

○ Logic

○ Markov

● Label



Only one branch may emanate from a label node. It works exactly like a decision node with one branch, or a chance node with one branch and probability 1.0. A label node's symbol is a simple black zigzag. The "expected" value of the label node is equal to the expected value of the node immediately to its right.

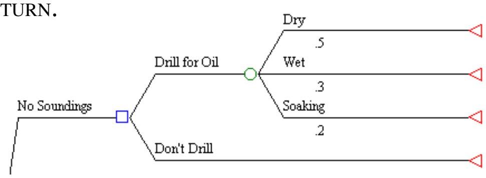
Lining things up

The following options enable you to make changes relating to the horizontal and vertical alignment of portions of a tree. To test these options, open the sample file, Oil Drilling #1. It models an oil wildcatter's decision about whether to drill for oil, and whether it pays to do seismic soundings before making that decision.

► **To align all endnodes at the right edge of the tree:**

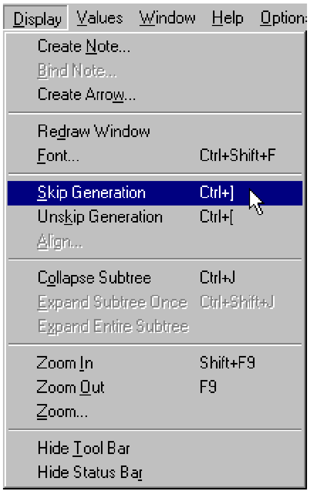
▷ Select Edit > Preferences.

- ▷ Choose Align Endnodes from the Tree Display page.
- ▷ Press ENTER or RETURN.



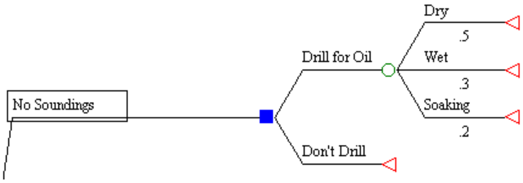
All of the terminal nodes should now be aligned at the far right end of the tree. If you can't see them, scroll horizontally until you can.

In most trees, all the endnodes are not naturally aligned since the number of levels of uncertainty encountered will depend on the path taken through the tree. However, rather than using the Align endnodes feature, it may be more helpful to line up certain interior nodes, such as the point at which a second decision must be made. For the oil wildcatter, this might mean lining up the decision nodes where it must be decided whether or not to drill.



► **To align a selected node with another:**

- ▷ Select an internal node, such as the No Soundings node.
- ▷ Select Display > Skip Generation, to move the subtree one node to the right.



This results in lining up vertically the two internal decision nodes at which you must decide, based on the information then available to you, whether or not to drill for oil.

It is possible to skip as many generations as needed by selecting the menu item again. You may also un-skip generations (on a node which has previously skipped generations) by selecting Display > Unskip Generation.

Tree compression

You can compress a large tree along its vertical axis by checking Minimize empty space in the Tree Display Preferences dialog box. This can yield extremely compact trees. There are four caveats, however.

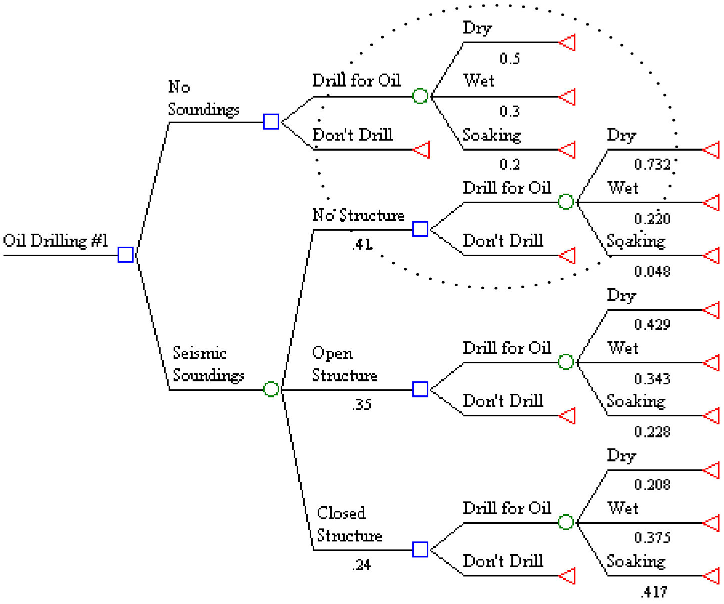
First, Minimize empty space is mutually exclusive with Align endnodes; the two options may not be used together. Second, Minimize empty space can cause problems with the location of expected value boxes when a tree is rolled back. Third, using this option with Branch lines at right angles can result in branch lines which slice through node symbols. Fourth, this option is likely to cause problems with the display of terminal node columns.

► **To compress a tree:**

▷ Select Edit > Preferences.

▷ Choose Minimize Empty Space in the Tree Display page. (You must ensure that Align endnodes is not selected.)

▷ Press ENTER or RETURN.



Other display features

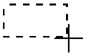
Complex models often benefit from the use of longer annotations than are possible in branch descriptions. Using note boxes and arrows, you can provide a clearer visual display for your audience. DATA also allows complete control over fonts used in notes and other text. For added control of the on-screen display, DATA allows you to zoom in or out to any magnification.

Annotation

Trees and influence diagrams can be annotated using notes and (in the tree window) arrows.

► **To draw a note in a tree:**

▷ Select Display > Create Note.



Your mouse cursor becomes a crosshair.

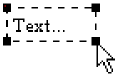
▷ Locate the cursor in a blank area of the tree window.

▷ Hold down the mouse button, and drag down and to the right.

A rectangular box is created. In the upper left corner is a blinking text insertion caret.

▷ Position your mouse cursor near the center of the rectangle.

Note that while it is inside the box, the arrow cursor becomes a text cursor.



▷ Slowly move the cursor to the edge of the box, and stop as soon as it changes from a text cursor into an arrow.

▷ Select the box’s outline with your mouse by clicking on it.

There should be a small handle at each of the four corners. These handles are used to resize the box.

▷ Click on one of the handles and, holding down the mouse button, drag to change both the size and proportions of the box.

▷ Without deselecting the box, release the handle, and click on the box’s dotted outline between handles.

▷ Holding down the mouse button, drag to change the location of the text insertion box.

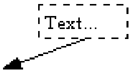
▷ After the box is positioned where you want it, click anywhere in its interior, and type a sentence or two into it.

▷ Change the shape and size of the box, to see how the text conforms to it.

You can also draw arrows in conjunction with annotating a tree. If, for example, you want to eliminate any confusion concerning which node is described by your note, you can draw an arrow from the note to the appropriate node.

► **To draw an arrow from a note to a node:**

▷ Select Display > Create Arrow.



▷ With the mouse, place the pointer beside your note, hold down the mouse button, and move the mouse to the appropriate node. Release the mouse button.

You may change the location of the endpoints of the arc by clicking and dragging them. You may ensure that the arrow is drawn strictly horizontally or vertically by holding the SHIFT key as you draw the arrow, or as you change the location of its endpoints.

Arrows cannot be employed when annotating an influence diagram. This is to avoid confusion with arcs.

Unless you specify otherwise, notes and arrows remain fixed notwithstanding changes in the tree or influence diagram. Thus, if you delete a subtree or add a node, you may find that a previously created note now overlaps with another object. In the tree window, this problem can be avoided by binding notes to nodes.

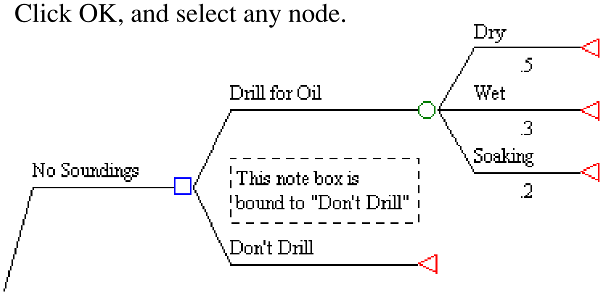
► **To bind a note to a node:**

▷ Select the note by clicking on its frame.

▷ Select Display > Bind Note....

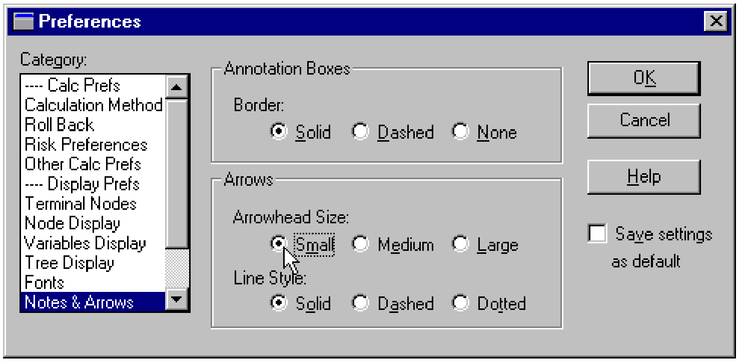
A dialog box tells you to select the node to which you want to bind your note.

▷ Click OK, and select any node.



Your note box has moved directly above the selected node, and it will move with the node through changes in tree geometry. While a note is bound to a node, you may change the contents of the note but not its location.

Notes and arrows can be copied and pasted, using the Edit menu. However, while a note is bound to a node, it is not possible to move, copy, or resize it, but it is possible to copy the text of the note.



Category:
- ---- Calc Prefs
- Calculation Method
- Roll Back
- Risk Preferences
- Other Calc Prefs
- ---- Display Prefs
- Terminal Nodes
- Node Display
- Variables Display
- Tree Display
- Fonts
- Notes & Arrows

Annotation Boxes

Border: ● Solid ○ Dashed ○ None

Arrows

Arrowhead Size: ● Small ○ Medium ○ Large

Line Style: ● Solid ○ Dashed ○ Dotted

□ Save settings as default

It is also possible to customize the appearance of notes and arrows. Annotation boxes can have solid borders, dotted borders or no borders. Arrows can have small, medium, or large heads. Arrow lines can be solid, dashed, or dotted. These characteristics are specified in the Notes & Arrows page of the Preferences dialog. Your changes will apply to all notes and arrows in the active tree. To have the changed characteristics apply in new trees, check Save settings as default.

Changing fonts

You can change the font, size, and style of text used in a tree. These changes can apply to: specified nodes, subtrees, or notes; an entire tree; or all subsequently created trees. To experiment with fonts, open any existing tree file.

► **To change the font of a node:**

▷ Select a node.

▷ Select Display > Font... (or click on the font icon in the toolbar).

▷ Change the font, size, or style, and click OK.

The name of the node should reflect the changes you just made. Similar changes may be made to the probability field of a node by clicking in that field and then selecting Display > Font....

► **To change the font of an entire subtree:**

▷ Select a node that is not an endnode.

▷ Select the subtree emanating from it by selecting Options > Select Subtree.

Alternatively, you could have selected the subtree by holding down the CONTROL key (Windows) or OPTION key (Macintosh) while selecting the node. See Chapter 11 for this and other node-selection techniques.

- ▷ Select Display > Font... (or click on the font icon in the toolbar).

A dialog box appears asking you to confirm that you wish to change the font of the entire subtree.

- ▷ Click OK.
- ▷ Change the font, size, or style, and click OK.

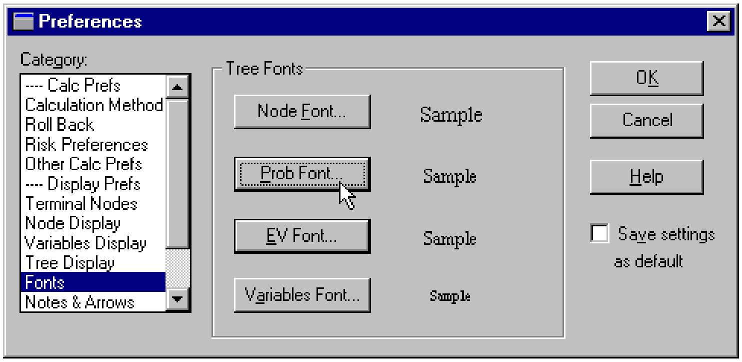
The names and probabilities of all nodes in the subtree should reflect the changes you just made.

► **To change the font of a note:**

- ▷ Click on a note box. If none exists, create one by selecting Display > Create Note…, and enter some text.
- ▷ Select Display > Font… (or click on the font icon in the toolbar).
- ▷ Change the font, size, or style, and click OK.

The text of the note should reflect the changes you just made.

The Fonts page of the Preferences dialog makes it possible to change, for the entire tree, the fonts used for node names, probabilities, expected value boxes, and (if displayed) definitions of variables.



Each button calls up the standard font, size, and style dialog, but changes made in those dialogs apply only in the limited context that their names reflect.

Clicking on the Node Font button allows you to change the font, size, and style for the names of all nodes subsequently created in the active tree. This font is also used for all existing nodes in the active tree, with the exception of nodes and subtrees at which you have individually changed the font.

Clicking on the Prob Font button allows you to change the font, size, and style for the probabilities of all nodes subsequently created in the active tree. This font is also used for the probabilities all existing nodes in the active tree, with the exception of nodes and subtrees at which you have individually changed the probability field font.

Clicking on the EV Font button allows you to change the font, size, and style for expected value boxes displayed upon roll back of the active tree. Changes do not affect any tree other than the active one.

Clicking on the Variables Font button allows you to change the font, size, and style for variables displayed beneath all nodes in the active tree. This option is relevant only if you elected to display full variable definitions in the tree, as described earlier in this chapter. Changes do not affect any tree other than the active one.

Any of the font settings applicable to the entire tree can be saved as defaults for subsequent trees. Simply check the box entitled Save settings as default (on the right-hand side of the Preferences dialog) and click OK. Your preferences will be saved on disk and will govern the font, size, and style of all trees created thereafter. (Note that *all* preferences will be saved, not just those in the Fonts page.)

Fonts in an influence diagram are handled similarly to fonts in a tree. There is a default font used for new nodes, set in the Fonts page of the Preferences dialog, and a default font for arc annotations. Each node and annotation box may use a different font. See the section on changing the font of a node in a tree, above.

Zooming

You may zoom in and out of any window.

► **To zoom out:**

▷ Select Display > Zoom Out (or press F9).

The tree has been reduced in size by a factor of 25%.

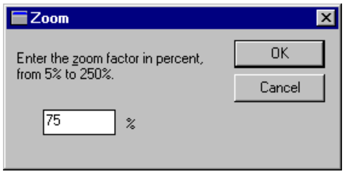
▷ Press F9 again.

The tree has been reduced by another 25%.

If you zoom out far enough, the name of the node under the cursor will appear in the status bar. This feature helps with location once the node names are too small to read in the tree window itself.

► **To zoom in:**

- ▷ Select Display > Zoom In (or press SHIFT - F9).



Zoom

Enter the zoom factor in percent, from 5% to 250%.

75 %

The tree has been enlarged by 25%. Press SHIFT - F9 again to enlarge the tree by another 25% and return the tree to its original size.

You may also enter a zoom factor manually by selecting Display > Zoom.... The factor in the Zoom... dialog box reflects, at all times, the percentage by which the tree's current size differs from its original size.

CHAPTER 11 SELECTING NODES

The model building and analysis features discussed so far require only that you be able to select nodes singly. This is done either by clicking on a node symbol or the branch to its left, or by moving the current selection to an adjacent node with the arrow keys.

If you wish to perform an operation on several nodes (such as defining variables or changing node types), it can be cumbersome to select them one at a time and perform the desired operation on each node successively. When multiple nodes are selected, it is possible to perform certain operations only once, providing the same result as repeating the process one node at a time. There a number of techniques for quickly selecting multiple nodes.

Some operations in DATA (e.g., cloning or replicating tree structures) require that you select a subtree, and not just its multiple nodes. This important operation is discussed first.

> **TIP:** Many menu commands, such as sensitivity analysis, are available only when a *single* node is selected. These menu items will be grayed when multiple nodes (or a subtree) are selected.

Selecting a subtree

► To select a subtree:

- ▷ Open any tree.
- ▷ Select any node that is not an endnode. This is the root node of the subtree that is to be selected.
- ▷ Choose Options > Select Subtree.

It is also possible to select the subtree by holding down the CONTROL key (Windows) or the OPTION key (Macintosh) while initially selecting the subtree's root node.

Selecting a subtree is fundamentally different from other types of multiple-node selections. When selecting a subtree you cause DATA to establish a logical relationship between the subtree and its root node. This relationship is critical for purposes of replicating the subtree, either via the clipboard or by cloning. Thus, to move or duplicate a subtree, you must begin the process by selecting the root node of the subtree and then Options > Select Subtree. No other means of selecting all the nodes in the subtree, such as shift-clicking, may be employed.

In fact, there is really only a single *selected* node when a subtree is selected: the root node of the subtree. The other nodes in the subtree are merely highlighted to identify the scope of the subtree.

Selecting multiple nodes

DATA offers several methods for selecting multiple nodes.

Shift-clicking

► To select several unrelated nodes:

▷ Select any node.

▷ While holding down the SHIFT key, select another node by clicking on it.

▷ Continue adding to the selection using the same shift-clicking operation.

Each selected node will turn solid color, indicating that they are selected. Holding down the SHIFT key while clicking on several nodes successively permits you to enlarge the number of concurrently selected nodes.

Using a selection rectangle

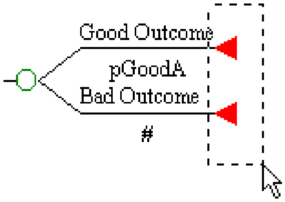
It is possible to select multiple nodes by dragging a selection rectangle around them.

► To select several adjacent nodes:

▷ Position the mouse cursor above and to the left of the topmost node you wish to select. Make sure it is far enough from any tree branches so that the cursor is displayed as an arrow-shaped pointer.

▷ Hold down the mouse button and drag down and to the right, so that the selection rectangle encloses each of the nodes you wish to select.

▷ Release the mouse button.



Adding nodes to the selection

► **To expand the selection:**

▷ Hold down the SHIFT key and click on additional nodes you wish to select, one after the other.

Removing nodes from the selection

► **To remove nodes from the selection:**

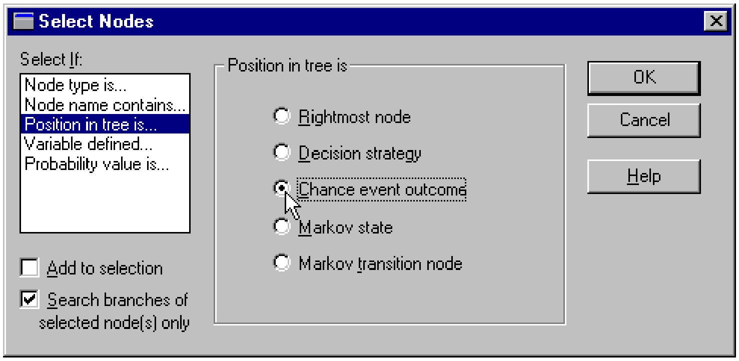
▷ Hold down the SHIFT key and click, one after the other, on each of the selected nodes that you wish to deselect.

You may also use the selection rectangle described above to add or remove nodes from the current selection, by holding the SHIFT key.

Selecting nodes by characteristic

It is also possible to have DATA automatically select specified nodes using the Options > Select If... command, which resembles the standard Find dialog. Using the Select If... command, it is possible to identify and select nodes based on:

- the name of the node;
- the type of the node;
- the location of the node;
- where a specified variable is defined; or
- the value of the associated probability.



Position in tree is

○ Rightmost node
○ Decision strategy
● Chance event outcome
○ Markov state
○ Markov transition node

□ Add to selection
☑ Search branches of selected node(s) only

If one or more nodes are selected at the time you open the Select If dialog, you will be able to add the new selection to the existing selection and/or limit the search to subtrees emanating from the selected node(s).

If you base your selection on node type, every node of the selected type will be selected. For example, if you select Terminal, every terminal node in the active tree will be selected.

Selecting the Node name contains... category will bring up a dialog with a text insertion box. Type in a text string, and DATA will identify every node at which it occurs. The search is not case-sensitive.

The Position in tree is... category allows you to highlight nodes based on their relationships. Choosing Rightmost node will select all nodes which have no branches emanating from them. This option can be used when you want to change all endnodes into terminal nodes. Choosing Decision strategy or Chance event outcome will select all branches emanating from decision nodes or chance nodes, respectively. Choosing Markov state or Markov transition node will select all the appropriate nodes in a Markov subtree.

The Variable is defined... option will locate every node at which a specified variable is defined. Any definition of the subject variable is accepted, regardless of value.

When you want to identify nodes based on their associated probabilities, choose Probability Value Is…, and specify a numeric target value.

Selecting the root node

It is possible to select the root node of the tree at any time by pressing CONTROL-HOME. This shortcut works regardless of the current selection.

Chapter 12 Managing Large Trees

The tutorial involving the investment problem focused on the construction and analysis of a small tree. It is likely that you will soon be producing considerably larger ones.

There are several reasons why it is important to keep trees as compact as possible. These include potential loss of focus and of the tree's usefulness in graphically communicating the problem's composition and your recommended solution.

DATA offers several ways to minimize the problems of working with large trees, although nothing can substitute for a careful, analytical approach to keeping the model as small and focused as possible.

Two features available in DATA which can reduce tree size significantly will be discussed initially. The first involves the use of cloned (rather than copied) subtrees; the second substitutes multiple, linked trees for a single, large tree. There is no restriction on using these features together.

By reducing tree size, cloning and dynamic tree linking provide benefits that go beyond simple speed enhancement. These features can enhance communications clarity even in situations where there are no perceived delays with screen redraw. You will also find it easier to maintain and update complex models using these features.

If, in spite of these features, you continue to experience unacceptable delays when making structural changes to your tree, DATA offers two other features that can speed the process of working with large trees. These features are described later in the chapter.

Remember that you can always get an overview of a large tree through the Zoom feature, which is described in Chapter 10. It is also possible to compress the entire tree structure somewhat, using methods described in Chapter 10.

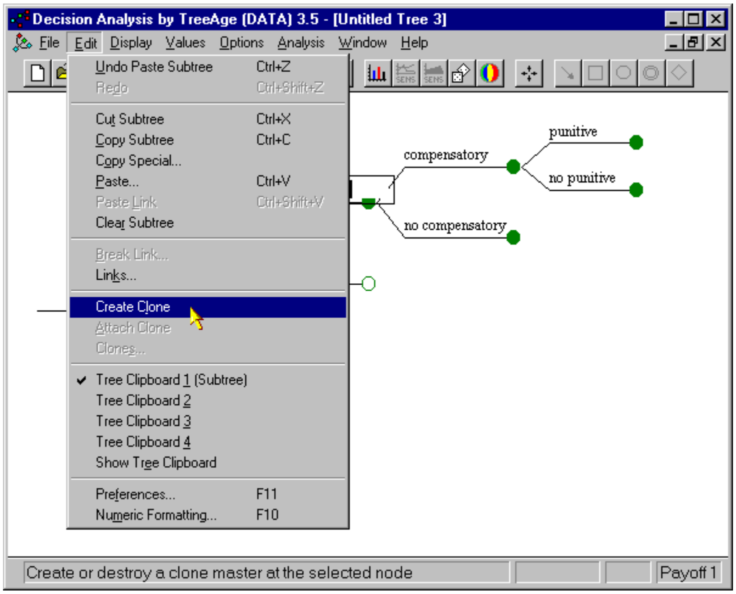
Cloning subtrees

In addition to using the Copy/Paste Subtree commands to replicate subtrees, it is also possible to clone subtrees.

The first step in the cloning process is to create a clone master: a subtree whose attributes (structure and values) are internally *published* for the purpose of being replicated at other nodes.

Clone masters remain fully editable, even after copies have been attached to other branches of the tree. Any changes made to the clone master are automatically and instantly replicated in the copies. In other words, DATA updates the clone copies so that they always remain identical to the clone master. Each subtree which has been designated a clone master can be identified by a heavy bar beneath the branch leading to its root node.

The fundamental difference between clone copies and pasted copies is that clone copies are linked dynamically to the master, and are therefore not directly editable. A clone copy takes its structure and other attributes from the clone master, not only at the time that the copy is created but until some action is taken which breaks the linkage. A pasted subtree, on the other hand, remains an exact duplicate of the original only until changes are made to either subtree. The pasted subtree is editable, unlike the clone copy subtree.



Create or destroy a clone master at the selected node

▶ To create a clone master:

▷ Select a node, select its subtree (see Chapter 11 on selecting subtrees), and choose Edit > Create Clone.

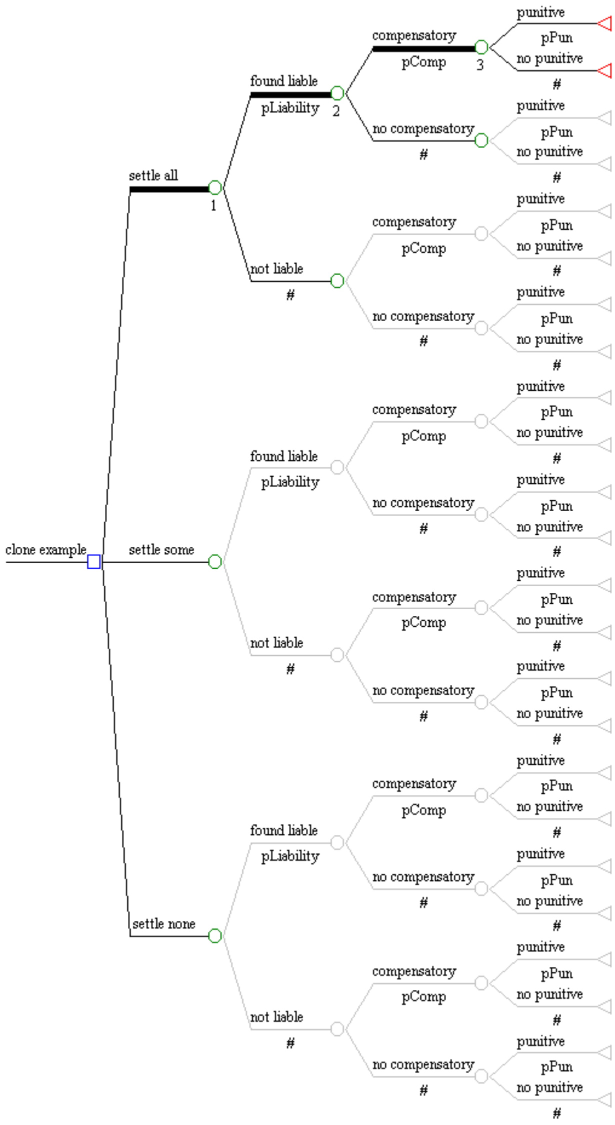
▷ Provide a short name to identify this clone master.

▶ To attach clone copies:

▷ Select an appropriate node (decision, chance, logic, or Markov, with no descendants).

▷ Choose Edit > Attach Clone, select the appropriate master from the list, and click OK.

If you have specified only a single clone master in the tree, it will be attached automatically.



► **To eliminate a clone master:**

▷ Select the clone master subtree.

▷ Choose Edit > Destroy Clone.

This will un-publish the subtree. When you destroy a clone master, the master subtree is not actually removed from the tree – only the copies are removed.

► **To detach clone copies:**

▷ Select the node where the clone copy is attached.

▷ Choose Edit > Detach Clone.

Before clone copies are completely deleted, you will be given the opportunity to leave in place unlinked copies of the original master subtree.

It is possible to change the index number or modify the name of a clone master. Choose Edit > Clones…, select the appropriate clone master from the list, click the Properties… button, and make the changes in the ensuing dialog.

Although it is possible to copy a subtree to the clipboard and paste it into another tree, cloning operates only within a single tree document. Furthermore, the regular tree clipboards do not maintain any clone information. A clone master, when copied as part of a larger subtree, will *not* be pasted as a clone master. The pasted copy of a clone copy will not be connected to the clone master, but the *original* master and copy will retain their connection.

Clone masters may be nested, but they may not be recursive. In other words, a single subtree may have several independent clone masters, with several clone copies attached as well. This is illustrated in the model shown above, in the settle all subtree. However, you may not attach a copy of a master subtree to itself to indicate recursion. You must use a Markov process to implement such cyclical models; see Chapter 25.

How clone copies are calculated

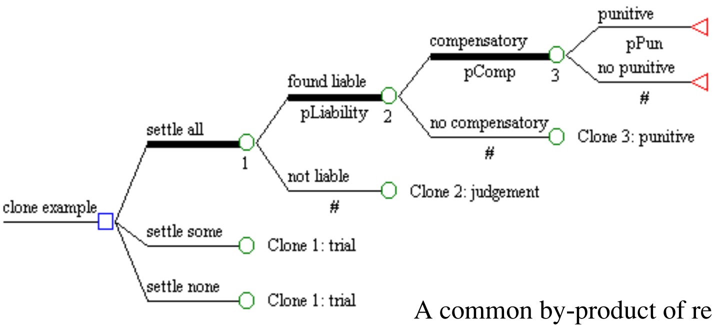
Calculations are performed as if a full copy of the master subtree existed at the location of the clone copy. This is true even if the copy is not displayed on the face of the tree because you elected to hide it (see below).

Since clone copies are identical to the clone master, variables should be used in the clone master when you want its copies to have different probabilities or other values. The probability and payoff expressions in the clone master and copies will all use the same variables, but it is possible for each to use different definitions for these variables. To enable this, specify these variable definitions at the root node of the clone master, not at nodes within the clone master subtree. Later, you will be able to assign different definitions to these variables at the root nodes of the clone copy subtrees.

The file Cloning Example provides an example of using variables with clones. Also see Chapter 8 for a detailed discussion of variables.

Hiding clone copies

In the Tree Display page of the Preferences dialog, you are given the option of hiding the display of clone copies. When this feature is turned on, the entire clone copy subtree is hidden; to the right of the root node of the clone copy is displayed the name of the clone master to which it is linked. Hiding the clone copies will not affect calculations in any way; also, hidden clone copies will continue to be updated when you make changes to the clone master.



By electing to hide the display of clone copies, it is possible to reduce the overall size of the displayed tree, often quite substantially. If you choose to display them, clone copies are displayed in gray. Their structure is not editable directly; a clone copy can be modified only by modifying the master subtree.

A common by-product of reducing tree size by hiding clone copies is enhanced clarity. The essential features of the replicated subtree may be understood by examining the clone master, which is the only instance of the subtree that is displayed. In addition, the linkages within the model which might otherwise be missed are clearly visible, as each clone copy indicates the master to which it is linked.

Nested trees

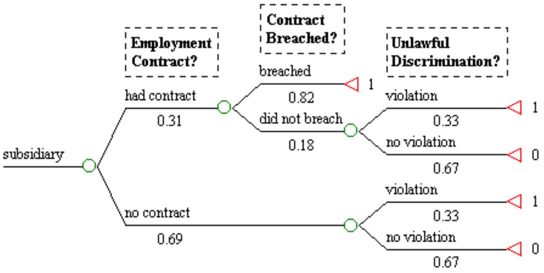
For some complex trees with many distinct areas of uncertainties, it may be possible to divide the model into multiple parts through the use of separate, but dynamically linked, trees. Links can be created to expected values, path probabilities, and standard deviations calculated in a tree; nested trees generally utilize links to expected value. The method by which you create links through Dynamic Data Exchange (Windows) or Publish and Subscribe (Macintosh) is covered in Chapter 15.

By "nesting" trees (designing one "master" tree and one or more subsidiary trees that feed into it), you can segregate some events and keep each tree more manageable than a single large tree that models everything.

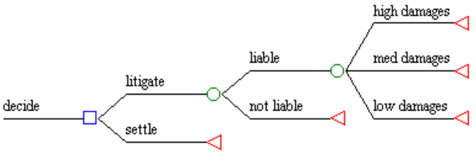
TIP: The use of nested trees has a potential drawback. Nested trees may not be appropriate if you need to perform a sensitivity analysis involving both linked trees, since sensitivity analysis can be performed on only one tree at a time. It may be possible to simulate a sensitivity analysis across trees, though. To enable this workaround, the link from the subsidiary tree's expected value should pasted into a variable in the master tree, rather than directly into a value field. See the detailed discussion, below.

Using a nested tree to calculate a probability

You may wish to employ a subsidiary tree to model the probability of an important outcome in the master tree. In this case, you would create the subsidiary tree (using only chance nodes) to model one or a series of secondary events whose occurrence would influence the probability of the primary event in the master tree.



To calculate a probability for the primary event, assign a value of 1 to each payoff in the subsidiary tree that represents the outcome of interest and 0 to all others. The *expected value* of the subsidiary tree, between 0 and 1, will represent the *probability* that the event will occur.



This technique can be used in trees that model litigation outcomes, for example. The issue of liability depends on a number of uncertainties, which lawyers call issues. These issues can be modeled in a subsidiary tree like the one shown above. Each scenario where liability results is assigned a payoff of one. No-liability scenarios have a payoff of zero. The expected value at the root node of the subsidiary tree becomes the probability of liability in the master tree.

After completing the subsidiary tree, simply create a link to its expected value utilizing DDE (Windows) or P&S (Macintosh). See Chapter 15 for more information. With the link information on the clipboard, the dynamic link between trees can be created.

In DATA for Windows:

- ▷ Select the root node of the subsidiary tree and copy a link to the node's expected value by choosing Edit > Copy Special....
- ▷ In the master tree, paste the link (using Edit > Paste Link) into the probability field of the appropriate node.

In DATA for Macintosh:

- ▷ Select the root node of the subsidiary tree and publish a link to its expected value by choosing Edit > Publishers....
- ▷ Using Edit > Subscribe To..., insert the link into the probability field of the node in the master tree.

Now, when the master tree is calculated, it will evaluate the branch probability using the last available expected value from the subsidiary tree. Changes made to the subsidiary tree will be reflected in the master tree’s calculations through this probability. Both trees must be open, and the subsidiary tree recalculated, for the master tree to update the link value.

Using a nested tree to calculate a payoff

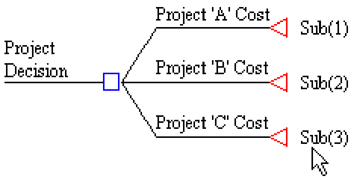
A subsidiary tree can also model the payoff of a terminal node (or nodes) in the master tree.

In DATA for Windows:

- ▷ Select the root node of the subsidiary tree and copy a link to its expected value by choosing Edit > Copy Special....
- ▷ Paste the link (using Edit > Paste Link) into the payoff of the relevant terminal node(s) in the master tree.

In DATA for Macintosh:

- ▷ Select the root node of the subsidiary tree and publish its expected value by choosing Edit > Publishers....
- ▷ Use the Subscribe To... button in the Enter Payoff dialog box to create a linkage at the relevant terminal node(s) in the master tree.



Using a nested tree to model a payoff or payoff component provides similar space conservation and clarity to that achieved by using a nested tree to model a probability. It is most useful in situations where you might otherwise use clones, but wish to exclude the elements of the clone from the master tree entirely by putting them in a different document.

Sensitivity analysis and nested trees

DATA does not *automatically* perform sensitivity analysis on more than one tree at a time. If you decide to create nested trees, there may be a way for you to *simulate* sensitivity analysis across linked trees, though. It requires that you create a variable in the master tree, employ that variable in the appropriate payoff or probability, and define the variable by a link from the subsidiary tree.

Thus, rather than pasting the link directly into a probability or payoff field as directed above, create a variable in the master tree to use the linked expected value from the subsidiary tree:

- ▷ Using the Define Values dialog, for the variable in question open a variable definition window at the root node of the master tree.
- ▷ Select Edit > Paste Link to create the link and close the variable definition window.

The next step is to perform a sensitivity analysis at the root node of the subsidiary tree on a parameter of interest (e.g., pEvent_x). The range of expected values generated by the sensitivity analysis (e.g., 0.3 to 0.8), the variable range specified (e.g., 0 to 1), and the type of correlation between these two ranges, positive or negative, should be noted.

Finally, a sensitivity analysis can be performed in the master tree on the variable initially defined by the link (e.g., pLiable). During this sensitivity analysis, instead of the value of the link, the variable (pLiable) will use the *the range of expected values* generated by the first sensitivity analysis of the subsidiary tree (on pEvent_x).

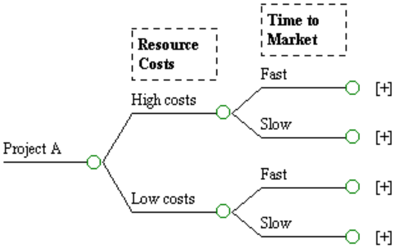
This is not a perfect substitute for an actual sensitivity analysis, as it assumes a linear relationship between the variable and the subsidiary tree's expected value. It may provide a useful workaround, though.

If the parameter varied in the sensitivity analysis of the subsidiary tree (pEvent_x) also exists in the master tree, it should be correlated to the linked variable (pLiable) during the second sensitivity analysis. The value range used for the correlated variable should be the same as the range used in the original sensitivity analysis in the subsidiary tree. The correlation type, positive or negative, can probably be inferred from the original sensitivity analysis, using the correlation between the value of the variable in the subsidiary tree and the expected value of that tree. In other words, if increasing the value of the variable (pEvent_x) causes the expected value of the subsidiary tree to increase, the correlation would be positive; otherwise, it would be negative.

Collapse subtree

When sharing your analysis with an audience, it can be hard to focus everyone's attention where you want it. This can be especially difficult if the tree is large or your audience is unfamiliar with decision analysis.

The Collapse Subtree command is designed to help with these problems.



▷ Select any interior node in your tree.

▷ Select Display > Collapse Subtree.

The entire subtree which emanates from the selected node has collapsed and is no longer visible. If, for example, the root node had been selected, the entire tree, except for the root node, would have been hidden.

A plus sign (+) is displayed to the right of the node, indicating that more of the tree exists. The collapsed subtree can be expanded all at once:

▷ Select Expand Entire Subtree;

or one generation at a time:

▷ Select Expand Subtree Once.

Collapsing and expanding subtrees in this way affects only the display of the tree. Calculations are not affected.

Influence diagrams

Influence diagrams offer a very compact visual representation of a model. In DATA, large models can be built almost entirely using the influence diagram interface, and then converted into a decision tree when calculations are required. When working with complex models, influence diagrams can both simplify the model building process and clarify the elements of a model to your audience.

CHAPTER 13 STORING ANALYSES

After performing a sensitivity analysis, you may wish to make some structural or numeric changes to your tree and then rerun the same analysis. To avoid having to specify the parameters of the analysis a second time, it is possible to save your analysis parameters (variables, ranges, etc.) for easy recall later.

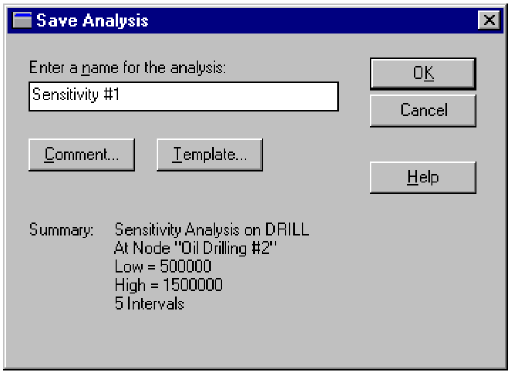
Most of DATA's analyses may be stored for future use.

Storing an analysis

► **To store the parameters of an analysis:**

▷ Open the Oil Drilling #2 tree.

▷ Perform a one-way sensitivity analysis at the root node on the variable Drill. Use five intervals, varying Drill from 500,000 to 1,500,000.



Enter a name for the analysis: Sensitivity #1

Summary: Sensitivity Analysis on DRILL
At Node "Oil Drilling #2"
Low = 500000
High = 1500000
5 Intervals

▷ As soon as the graph window appears, switch back to the tree window.

▷ Choose Analysis > Storage > Save Last. You must choose this command storage for only one set of analysis parameters.

▷ Give a short, descriptive name to your analysis. This name is only for your own reference. You may optionally enter a longer description of the analysis by clicking the Comment button. For now, ignore the Template button.

▷ Press ENTER or RETURN.

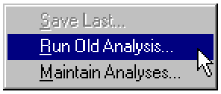
You can use this method to store the parameters for all of the analyses available in the Analysis menu which require the user to input parameters and/or select a particular node. This includes all items in the Sensitivity… submenu. Analysis parameters are saved with the tree file.

Running and updating stored analyses

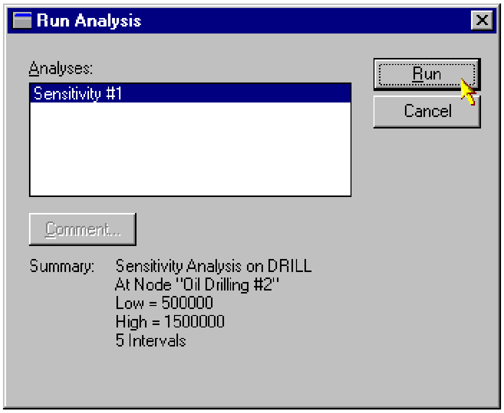
► **To run a previously stored analysis:**

▷ Choose Analysis > Storage > Run Old Analysis.

▷ Select the analysis from the list of stored analyses. For each, you will see a summary which DATA generated to describe the nature of the analysis. If you have entered a comment for the analysis, you may view it by pressing the Comment button.

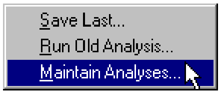


▷ Click the Run button.



Summary: Sensitivity Analysis on DRILL
At Node "Oil Drilling #2"
Low = 500000
High = 1500000
5 Intervals

If, after storing an analysis, the tree is structurally modified or changes are made in the location of variable definitions directly involved in the analysis (e.g., sensitivity analysis variables and their correlates), DATA may be unable to reconcile the changes. The analyses may be aborted, but it may also result in incorrect calculations.



You may delete or rename analyses by using the Analysis > Storage > Maintain Analyses command. You may also edit or assign a longer comment or a template to each analysis. Templates are described in Chapter 33, Graph Windows, and in the next section.

Using graph templates with stored analyses

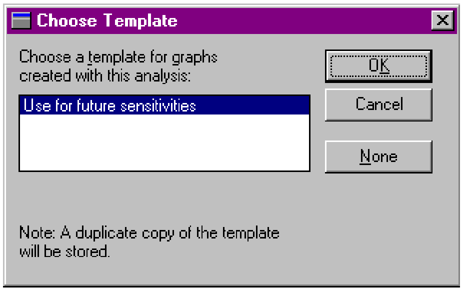
A graph template is a description of the non-numeric content of a graph, including font information, specific text items, and numeric formatting. Normally, one applies a graph template after the graph is created. It is also possible to store a template with any stored analysis that generates a graph. The stored template will be applied automatically when the analysis is re-run and the graph is created.

See Chapter 33 for general information on graph templates. This section assumes you are familiar with templates, and only describes the use of templates in the context of stored analyses.

► **To store a template with an analysis:**

- ▷ Create a template from a graph window, as described in Chapter 33.
- ▷ In either the Save Analysis dialog box or the Maintain Analyses dialog box, click the Template button.
- ▷ Select a template from the list, and press OK.
- ▷ Press OK again to store the template.

DATA stores a full duplicate copy of the template with the analysis. If you change the original template or delete it altogether (from the Graph > Maintain Templates command), the duplicate copy stored with the analysis will not be affected. You may change which template is stored with the analysis by using the Template button in the Maintain Analysis dialog. You may also detach the template by clicking the None button in the template selection dialog box.



Choose a template for graphs created with this analysis:

Use for future sensitivities

Note: A duplicate copy of the template will be stored.

It is not possible to modify directly the template copy stored with an analysis. Instead, replace the template copy with a different template.

Using stored analyses with a custom interface tree

This section assumes that you are familiar with the Custom Interface feature, which is covered in Chapter 18.

The Maintain Analyses dialog box has some features which may be particularly useful when designing a tree with a Custom Interface. The user of your tree will be presented with the same list of analyses that appears in the Run Analysis dialog box, although a slightly different interface is used.

Be sure to take care in assigning a meaningful description and long comment. The long comment is always displayed in the main Run Analysis dialog for run-time users, rather than in a secondary Comment dialog. Since the summary which DATA generates (describing the analysis parameters) is not shown to run-time users, you may wish to include some of that information in your comment.

You can use the Move Up and Move Down buttons in the Maintain Analyses dialog to change the displayed order of the stored analyses.

To see how the run-time user's Run Analysis dialog box will look, pull down the Options menu, and select Show Custom Interface.

Building custom decision analysis applications

DATA Interactive™, a software package for designing custom decision analysis applications, makes it possible to use a Visual Basic script to build interactive analyses for use with your DATA decision trees. Analyses and trees can be distributed either from an Internet or intranet server, or on a CD-ROM. See Chapter 18 for more information.

CHAPTER 14 MISCELLANEOUS PRODUCTIVITY FEATURES

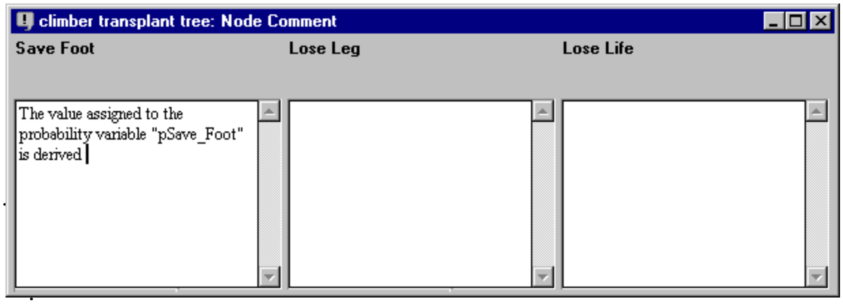
Node comments

The use of note boxes to annotate models was covered in Chapter 10; assigning variable comments was covered in Chapter 9. It is also possible to add long comments which are associated with the branches of a node. Node comments are saved with the tree but, unlike note boxes, are not displayed on the face of the tree.

Extensive annotation can be stored at each node. Node comments are particularly useful for recording the basis on which probability assignments were made at a particular node.

▷ Open a tree, and select a node that has at least two branches.

▷ Choose Options > Node Comment....



climber transplant tree: Node Comment

Save Foot

Lose Leg

Lose Life

The value assigned to the probability variable "pSave_Foot" is derived

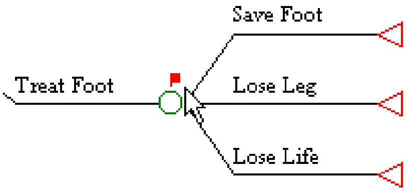
The window that appears will have one pane for each branch emanating from the selected node. You can move forward from pane to pane by pressing the TAB key; SHIFT-TAB will move the insertion point in the reverse direction.

▷ Type some comments into the first pane, and close the window.

▷ Select Edit > Preferences... and choose the Node Display page.

▷ In the resulting dialog box, select Mark nodes with comments, and press ENTER or RETURN.

In the tree window, the node at which you entered a node comment is identified by a small flag. This flag will not appear in a printout or when the tree is imported into another application.



Node comments will not be printed along with the tree. You may, however, print a node comment as a separate document. You may also preview the printed document.

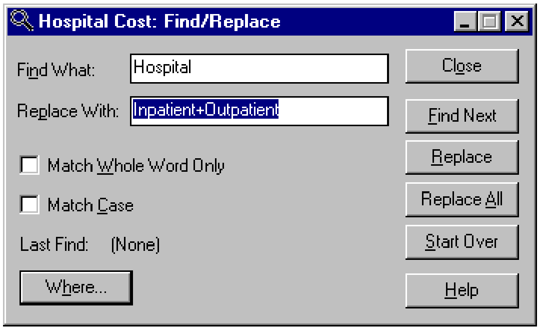
You may change the font associated with the node comment by choosing Display > Font when the Node Comment window is open. This font will be used for both display and printing.

Find/replace

DATA can help you to quickly search for and modify text elements of your tree. DATA's Find feature is the best way to locate, for example, every place where you have used a variable in a formula. It can reliably find all occurrences of a variable's name in formulas throughout a tree, and replace them with another variable name. It is not recommended as a way to change existing variable names; name changes should be made in the Properties dialog box.

To open the Find dialog, choose Options > Find.

The Find Next button searches for the next instance of the specified text after an instance has been located. Replace will replace the last found instance of the search string; Replace All will repeat the Find Next and Replace operations until the tree is fully traversed. Start Over recommences the search from the beginning of the tree.



Find What: Hospital

Replace With: Inpatient+Outpatient

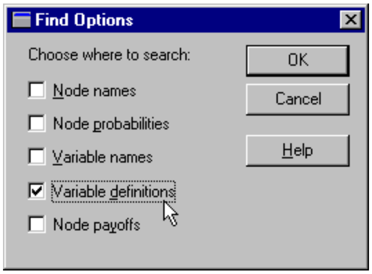
□ Match Whole Word Only

□ Match Case

Last Find: (None)

If the Match Whole Word Only box is checked, DATA will find the specified text only if it is a whole word; if the Match Case box is checked, DATA will find the specified text only if it has the same combination of upper and lower case as that specified by the user.

Finally, the locations for the search may be specified using the Where button in the ensuing dialog box. You may, for example, wish to limit your search to node names, or to variable definitions.



Choose where to search:

- ☐ Node names
- ☐ Node probabilities
- ☐ Variable names
- ☑ Variable definitions
- ☐ Node payoffs

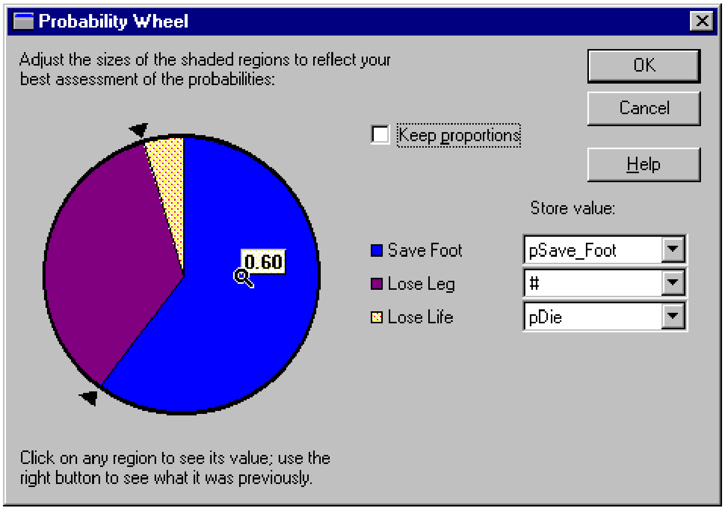
Changes made using the Find/Replace method cannot be undone. After replacements have been made, you must revert to an earlier saved version of the tree to reinstate the modified text.

If you wish to search for logical or structural element of the tree, instead of a text phrase, use the Options > Select If… command covered in Chapter 11.

Probability wheel

A problem inherent in decision analysis is the subjective assignment of probabilities. Many experts strongly endorse the use of a graphical aid called a probability wheel in making these assignments, on the premise that it is preferable to assign subjective probabilities visually, rather than numerically.

DATA features a probability wheel which can be accessed by choosing Values > Probability Wheel, or by clicking its icon in the tool bar. The wheel is available when you have selected a single chance node. It works whether or not you have already assigned probabilities to its branches. If you have assigned probabilities, DATA will use them as initial values for the wheel. The chance node can have up to seven branches; each branch will be assigned its own slice of the wheel.



Adjust the sizes of the shaded regions to reflect your best assessment of the probabilities:

☐ Keep proportions

	Store value:
■ Save Foot	pSave_Foot
■ Lose Leg	#
▨ Lose Life	pDie

Click on any region to see its value; use the right button to see what it was previously.

To use the probability wheel, drag the pointers around the edge of the wheel until the sizes of the pie wedges match your best assessment of the probabilities being assigned. Each pointer corresponds to a dividing line between two probabilities. If the selected node has three or more branches, you will see a check box named Keep proportions. If selected, DATA will ensure that ratios are maintained on each side of the pointer you move.

Moving the mouse cursor over one of the pie wedges will change the pointer into a magnifying glass. Holding down the mouse button will display the numeric value (probability) of the wedge over which the magnifying glass is positioned. Right-clicking (Windows) or CONTROL-clicking (Macintosh) will display that wedge's numeric value at the time you opened the wheel.

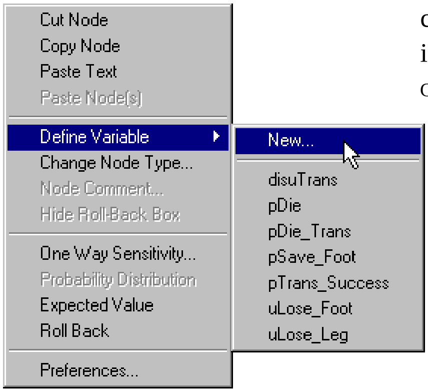
DATA enables you to store the values from the wheel numerically or as variables. For each wedge, you can choose to store the value in one of three ways: numerically, as “#”, or as the value of a variable. If you choose numerically, the value of the pie wedge is inserted directly into the probability field of its associated node. If you use # (and you may do so for only one slice), the node associated with that slice will have # as its probability (signifying the remainder after all other probabilities are calculated). You may also choose a variable which will store the value of the probability. The variable will be defined at the selected chance node.

The probability wheel is also available in the Define Variable window. When you are defining a variable, you can call up the wheel to edit the value of the variable. The numeric value will be inserted as the variable’s value when you finish with the wheel.

Shortcuts

Quick menu

In DATA for Windows, clicking in the tree window with the right mouse button (“right-clicking”) will provide access to many common commands via a quick menu. In DATA for Macintosh, the quick menu is invoked by clicking in the tree window while holding down the CONTROL key (“CONTROL-clicking”).



The quick menu provides separate commands for pasting text and pasting nodes, thereby enabling you to avoid the Paste... dialog when both types of items are available on the clipboard. It also allows you to define variables, change node types, add node comments, and hide individual expected value boxes. Finally, it contains some common analysis options and offers access to the Preferences dialog. A number of the commands will only be available when a node, or a particular kind of node, is selected.

Undo

There are likely to be occasions when you will want to undo the last action that you took, but prefer not to employ the Revert to Saved command because it will wipe out all of your changes since the file was last saved. On these occasions, DATA’s Undo command, located in the Edit Menu in both the tree and influence diagram windows, can be a real time saver.

DATA maintains in memory details of the last ten actions. This makes it possible, in most situations, to undo each of the last ten actions, beginning with the most recent action and working back one at a time.

The limitation is that certain actions cause the entire list of undo items to be lost. This may occur as a result of closing the tree or influence diagram, distributing children, rolling back the tree, replacing text from within the Find dialog, using the probability wheel, or changing the font of an entire subtree. In addition, certain actions may not be undone at all, such as saving the tree.

DATA has separate Undo and Redo commands. Both are found in the Edit menu.

▷ Open the Climber Transplant tree.

▷ Select the Treat Foot node, and choose Options > Select Subtree.

▷ Select Edit > Clear Subtree.

▷ With the Treat Foot node still selected, choose Edit > Clear Node.

<Body/The Treat Foot subtree and the Treat Foot branch have now been deleted. Since the Clear command was used, no copy of either the subtree or the branch exist on the clipboard. If you had meant to delete the Foot Transplant subtree, but had mistakenly deleted the Treat Foot subtree, you would want to undo the most recent changes.

▷ Pull down the Edit menu.

Since the last action taken was to clear a node, the topmost command under the Edit menu reads Undo Clear Node.

▷ Select Edit > Undo Clear Node.

The Treat Foot branch reappears.

▷ Pull down the Edit menu again.

Now, the topmost option reads Undo Clear Subtree, because the action you performed prior to clearing the Treat Foot node was clearing its subtree. Note that if you decided you did want to remove the Treat Foot node, you would choose the second option, Redo Clear Node.

▷ Select Edit > Undo Clear Subtree.

The Treat Foot subtree reappears.

Calculating complementary probabilities for all nodes

By selecting the Calculate complementary probabilities automatically option located in the Other Calc Prefs page of the Preferences dialog, DATA will automatically identify the single branch whose probability has not been entered. Then, the remainder probability will be evaluated and displayed as its calculated, numeric value.

► **To cause the automatic calculation of complementary numeric probabilities:**

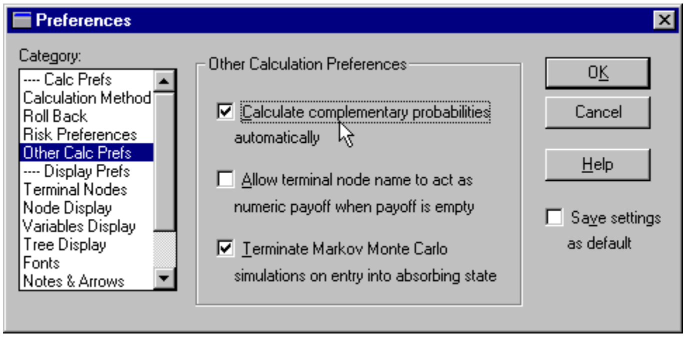
▷ Open the sample file Rock Climber Tree.

▷ Select Edit > Preferences and choose the Other Calc Prefs page.

▷ Click Calculate complementary probabilities automatically, and press ENTER or RETURN.

▷ In the Treat Foot subtree, delete the probabilities associated with the Save Foot, Lose Leg, and Lose life outcomes.

▷ Enter 0.6 and 0.05 as the probabilities of the Save Foot and Lose Life branches, and click outside the node.



Other Calculation Preferences

☑ Calculate complementary probabilities automatically

☐ Allow terminal node name to act as numeric payoff when payoff is empty

☑ Terminate Markov Monte Carlo simulations on entry into absorbing state

☐ Save settings as default

The probability of the Lose Leg subtree, 0.35, is automatically filled in for you. The numeric format of an automatically entered probability will be determined by the default numeric formatting.

This technique would not be available at a subtree using one or more a probability variable, because DATA will assign a complementary probability only when all branches use numeric probabilities. In other words, after *numeric* probabilities have been entered for all but one of the branches emanating from a chance node, DATA will determine the probability at the remaining branch.

The calculated value is inserted as editable text. If you later change one of the probabilities on the branches, the complementary probability is not automatically updated. It is a one-time calculation.

This option and the # symbol (see Chapter 5) can both be used within a single tree. If Calculate complementary probabilities automatically is turned on, it will apply to numeric probabilities throughout the tree, except to any set of branches where you have inserted # in one of the probability fields.

Numeric entry shortcuts

Most numeric entry boxes, such as those for sensitivity analysis or graph axis modification, accept special numeric characters: “k” to indicate thousands, “m” to indicate millions, “b” to indicate billions, and “e” to indicate scientific notation, as in 3.4e7.

Terminal node names as payoffs

If certain conditions are met, it is possible to have DATA treat the branch description at a terminal node as that node’s numeric payoff value. If no payoff has been assigned at this terminal node, and you have selected this option in the Other Calc Prefs page of the Preferences dialog, DATA will attempt to use the name of the terminal node (its branch description) as the payoff.

For this to be successful, the name must be wholly numeric, with no arithmetic operators, although it may contain a currency sign and thousands separators. Multiple payoffs can be entered if they are separated with a backslash, as in $100\50. If DATA is unable to interpret the node name, you will be informed that the payoff is empty.

Optimal path

DATA identifies the optimal path at decision nodes when a tree is rolled back by highlighting the appropriate subtree. There are a number of related features covered here.

Show optimal path

Once all the values and probabilities have been assigned in a tree, you can quickly determine the optimal choice at any decision node.

- ▷ Open the Climber Transplant tree.
- ▷ Select the decision node.
- ▷ Select Analysis > Show Optimal Path.

The selection moves to the node Treat Foot. This indicates that when faced with this decision, the benefits to the patient are likely to be greater if the Treat Foot action is taken

Note that this information is also shown when you select Analysis > Expected Value at a decision node.

Force path

With the Force Path command, you can indicate the certain occurrence (or inevitability) of a particular event at a chance node, or of a commitment to a specific alternative at a decision node. Invoking Force Path at the decision node:

- sets the probability of the specific event or alternative to 1 and the probability of the other branch(es) to 0; and
- converts the selected node's parent to a logic node (see Chapter 34).

When you have a complex tree and wish to indicate that a previously uncertain event has taken place, select the node that represents that event and choose Options > Force Path.

The selection of either high or low optimal path will control DATA's analysis at each decision node within a given tree. See Chapter 19, Changing What DATA Calculates, for more information on setting up a tree to maximize or minimize.

Occasionally, however, you may want DATA to apply the opposite criterion at a specific decision node.

► **To change the optimal path for a given decision node:**

▷ Select the node.

▷ Select Options > Change Optimal Path.

▷ Click Yes in the confirming dialog.

The decision node will reappear with an arrow inside it. The arrow will point upward if that decision node has been reset for maximization, or downward if the node has been reset for minimization.

If you reverse the optimal path for an entire tree in which the optimal path of one or more nodes has been individually changed, the arrows in all of the individually changed nodes will reverse, signifying that they remain different from the rest of the tree.

The reversal of a node's optimal path can be undone by selecting the node and then performing the operation described above a second time.

Chapter 15 Basic Linking

This chapter covers the use of Dynamic Data Exchange (Windows) and Publish and Subscribe (Macintosh) for exchanging calculated values between applications. Additional means of exchanging information between DATA and other applications are described in Chapters 16 through 18.

Using Dynamic Data Exchange ("DDE") is very similar to using the standard clipboard for a onetime transfer of information. With DDE, however, new information is updated automatically, eliminating the need to repeat the copy/paste sequence whenever the information changes. Publish and Subscribe ("P&S") offers most of the functionality provided by DDE, but the methodology and terminology are somewhat different.

In building and analyzing DATA models, there are several ways to exchange values via DDE (Windows) or P&S (Macintosh):

- Trees can capture individual values from another application for use in formulas. These values will be updated as determined by the external application. If the value is stored in a spreadsheet cell, for instance, a new value will be sent to your tree whenever the value of that cell changes. The data linking process is managed by DATA's Sub() function. (*Sub* is shorthand for Subscribe; DATA is acting as a client that *subscribes* to information published by a server application.)
- Individual nodes in a tree can export calculated values for use in external applications. You may export the expected value, path probability, and/or standard deviation as calculated at a specified node. A new value will be sent when the tree is rolled back or when that node is selected and the corresponding command in the Analysis menu is chosen. This application of DDE (Windows) is implemented by the Copy Special and Paste Link menu items in the Edit menu. Implementation under P&S (Macintosh) utilizes the Publish and Subscribe To commands in the Edit menu.

- Individual nodes in a tree can export calculated values for use in another DATA tree. Trees linked in this way are said to be *nested*. Nested trees are discussed in Chapter 12. The basic process for setting up links between trees is similar to that described above for linking between DATA and another application.

In addition to DDE linking, DATA for Windows (but not DATA for Macintosh) includes more powerful, bi-directional linking. With bi-directional links, information can be passed back and forth between a decision tree and a Microsoft® Excel™ (97 or higher) spreadsheet, with the spreadsheet utilized as a data repository and calculation engine. Excel can be used, for example, to calculate complex payoff formulas. For each scenario in the tree, definitions of specified variables will be sent to your spreadsheet, the spreadsheet will be recalculated, and the result of the recalculation will be imported by DATA for that scenario. There are many uses of bi-directional linking; see Chapter 16 for detailed discussion of this feature.

Dynamic Data Exchange (Windows)

There are a few terms you should be familiar with before proceeding. In DDE, a document or application which makes information available is called a *server*, and a document or application which receives information over a DDE link is called a *client*.

DATA for Windows can act as either client or server, or both.

Using DATA as a DDE client

► **To receive externally stored values via DDE:**

▷ Create a new document in your external application, for instance an Excel spreadsheet, and save the file.

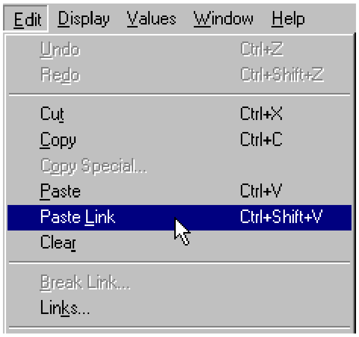
▷ Select the item in your external application (for instance, a cell in the spreadsheet). Choose Copy from the other program's Edit menu. (Some applications may have a Copy Link or Copy Special menu item that is more appropriate.)

▷ Switch to DATA and activate the tree.

Linked data can be pasted into DATA models in two ways.

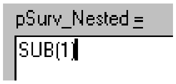
► **To insert a new link directly into a formula:**

▷ Place the cursor in the appropriate variable definition, payoff formula, or probability field.



▷ Choose Edit > Paste Link.

The Paste Link command is available only when there is a link on the clipboard and the text cursor is in a formula editor. This command will create the link, assign an index, and insert the text "Sub(n)" at the insertion point. Using the Paste Link command is the quickest way to create new links.



pSurv_Nested =

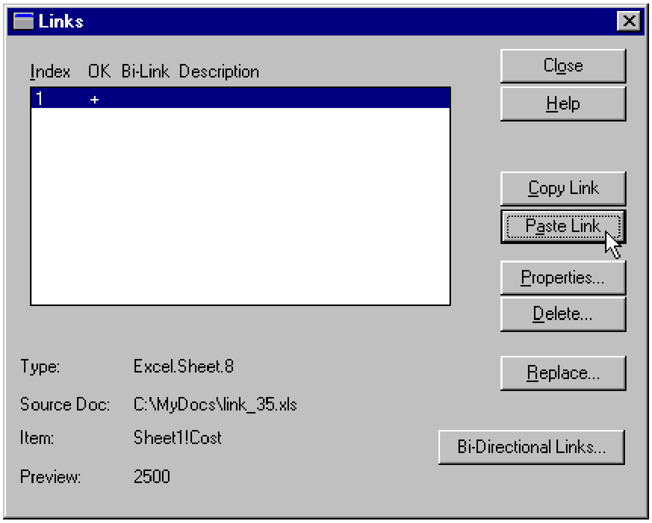
SUB(1

The Links dialog, on the other hand, provides more control over the link creation process, as well as immediate confirmation of the value currently available through the link.

► **To create a link using the Links dialog:**

▷ Select Edit > Links.

▷ In the Links dialog, press the Paste Link button.



Index	OK	Bi-Link	Description
1	+		

Type: Excel.Sheet.8

Source Doc: C:\MyDocs\link_35.xls

Item: Sheet1!Cost

Preview: 2500

The Links dialog displays, for each link you create, the following information:

Index The index, which is initially assigned by DATA but may be changed by you, is used in the Sub() function. When the Sub() function is encountered in a formula, DATA uses the most recent value for the link with the specified index. You may use the value of a single link as often as you wish by using Sub() with the same index inside several different formulas. In place of an integer, any expression which evaluates to a valid index can be used in a Sub() function. For example, in a Markov process you might define a probability or payoff using a reference like Sub(_stage+1).

OK A “+” will appear in this column in the dialog if the link is currently open. This will occur when the server document is open and DATA is able to establish a DDE connection with it. DATA will not automatically open DDE link source documents.

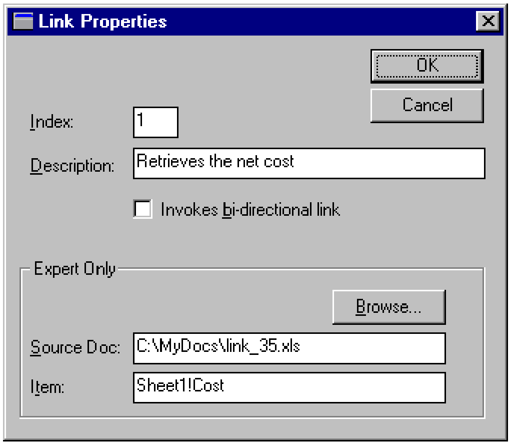
Bi-Link A “Y” will appear in this column if the link invokes a bi-directional link (see below).

Description For your own annotation purposes. This field is optional.

More about DATA as a DDE client

If you are linking to a named cell in a spreadsheet, DATA will link to the name rather than to the cell position. It is advisable to used named cells so that the link will be maintained even if modifications to the spreadsheet cause the cell location to change. See your spreadsheet application’s documentation for details on naming cells.

To update the value of a DDE link, both the client document (your tree) and the server document must be open, and a tree calculation using the link must be performed. In addition, if there are multiple worksheets in a source spreadsheet, the worksheet with the linked cells should be on top. Linked-to cells should not use currency, or other text-based, formats. DATA will not prohibit links to cells with improper formatting; links will simply not update, potentially halting calculations.



Link Properties

Index: 1

Description: Retrieves the net cost

□ Invokes bi-directional link

Expert Only

Source Doc: C:\MyDocs\link_35.xls

Item: Sheet1!Cost

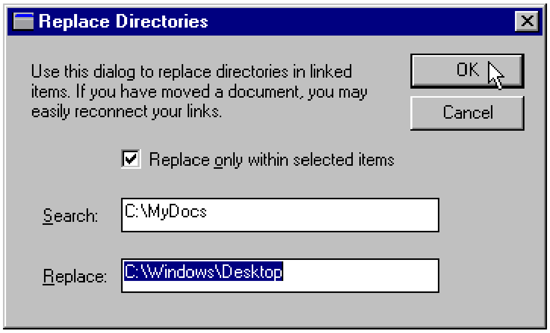
You can change the index or description of an individual link item by selecting it in the Links dialog and clicking the Properties… button.

Note that changing the location of the client tree will not affect the link, as long as the server document is still accessible to DATA. If the location or name of the server document, worksheet, or cell reference has changed, you may need to update the Link Properties dialog; if you have to update many links at once, try using the Replace… button in the main Links dialog, instead.

The Replace… button enables you to quickly update the Source Doc property of many links at once. This feature is useful if you create multiple links to a spreadsheet, and then move the spreadsheet to a different directory, drive, or computer.

► **To replace path strings in many links at once:**

- ▷ In the Links dialog, select all the links with a common path element that must be updated, and then click on the Replace… button.
- ▷ Enter a path string to search for, such as "C:\MyDocs", and a replacement string, such as "C:\Windows\Desktop".



Replace Directories

Use this dialog to replace directories in linked items. If you have moved a document, you may easily reconnect your links.

☑ Replace only within selected items

Search: C:\MyDocs

Replace: C:\Windows\Desktop

- ▷ If you wish to search only in the selected links, check the Replace only within selected items option. If this option is not checked, all link items will be searched and updated.
- ▷ Press ENTER or RETURN to update the links.

DATA will find all occurrences of the search text in the selected source documents' paths, substituting the new path string you provide.

The Copy Link button in the Links dialog box enables you to duplicate links from one tree to another. Select the links that you would like to pass to another tree and choose Copy Link(s). Open the Links dialog box in the other tree and choose Paste Link(s). Note that links copied in this way may not be pasted into other applications as an active link; the feature is only for use within DATA.

The Bi-Directional Links… button opens the Bi-Directional Links dialog. Bi-directional linking is covered below, and in Chapter 16.

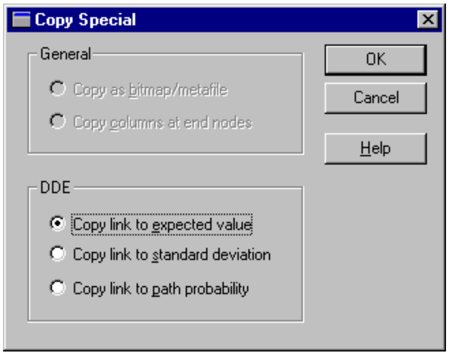
The Delete button allows you to remove unused links from the tree. To delete multiple links, select one link item, then hold down the CONTROL key as you click on additional links. Pressing the Delete key will remove all selected links from the tree.

Using DATA as a DDE server (Windows)

Using DDE, DATA for Windows can send calculated values from a tree to any DDE client document, including other trees (see Chapter 12 on nested trees) or spreadsheets. Each node can export via DDE its expected value, path probability, and/or standard deviation.

> **TIP:** DATA can be used as a DDE server by a client spreadsheet, as illustrated below; more reliable and robust links are possible, though, using DATA's ActiveX™-based, *bi-directional* links. If you plan to use a node's calculated value in an Excel (97 or later) spreadsheet, the additional work required initially to set up bi-directional, rather than DDE, links is a worthwhile investment. For more on bi-directional links, see below and Chapter 16.

► **To use a node's calculated value in another document:**



Copy Special

General

○ Copy as bitmap/metafile

○ Copy columns at end nodes

DDE

● Copy link to expected value

○ Copy link to standard deviation

○ Copy link to path probability

▷ Select a single node whose value you would like to use.

▷ Choose Edit > Copy Special.

▷ In the Copy Special dialog box, choose the appropriate radio button for the value you wish to export. Then, close the Copy Special dialog.

▷ The DDE link information is now on the clipboard. In the client document (which may or may not be a DATA document) choose Paste, Paste Link, or Paste Special.

Each application handles DDE links differently; consult its documentation for details. See the previous section for how to paste links into DATA.

If the client document is open, all links to calculated values will be updated whenever you roll back the tree. In addition, particular calculated values are updated and sent to the client when the corresponding Analysis menu item and node are selected. For instance, if you have copied a link to the path probability of a node, the link will be updated both upon roll back and when you choose Analysis > Path Probability with that node selected.

If the client document is closed when you roll back the tree, it will not receive these newly calculated values. When using DDE, both documents must be open. See Chapter 16 and the tip, above, for some advantages of using ActiveX links, instead of DDE.

Using bi-directional links (Windows)

If you have developed a complex formula in a Microsoft Excel spreadsheet, for instance to calculate the value of a scenario (what DATA calls the payoff), you may not have to duplicate it in your decision tree. DATA for Windows can harness the spreadsheet as a calculation engine. During each calculation of a scenario in your tree, the values of specified variables (the "inputs" of a spreadsheet formula) can be passed to Excel, the spreadsheet recalculated, and specified cell formula values returned to the tree.

This two-way exchange of calculated values can be used, in conjunction with Excel's ODBC-connectivity, to utilize the results of database queries in tree calculations. Using bi-directional links, the spreadsheet's external link to a database query can be refreshed, and the query results returned to the spreadsheet and, ultimately, the tree.

Bi-directional links can also be used (instead of DDE links) when a spreadsheet requires DATA to export calculated values from one or more nodes (or trees). This is true even if you don't need to use Excel as a calculation engine for your tree. Bi-directional links are flexible, easy to maintain and update, and substantially more stable than DDE links.

Bi-directional linking, implemented with ActiveX technology, is covered in Chapter 16.

Publish and subscribe (Macintosh)

In DDE, a document or application that makes information available is called a *server*. In P&S (Macintosh), the corresponding term is *publisher.* In DDE, a document or application that receives information via a link is called a *client*. In P&S, the corresponding term is *subscriber.*

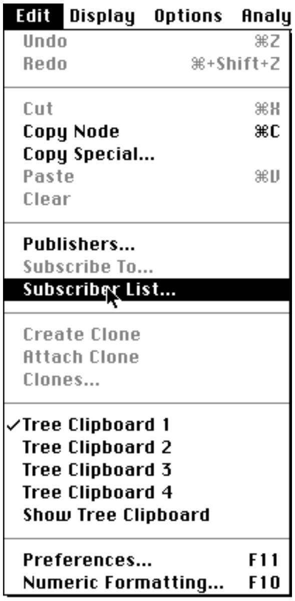
Unlike DDE, where both applications must be open at the same time and the clipboard is used to transfer information between them, P&S utilizes an intermediate file, known as an *edition* file. The edition is a separate file, which is linked to the publisher and contains a copy of the most recent information published by the publisher. The edition can be set to update automatically. One or more subscribers can be linked to the edition and can be set to update automatically as well.

Using DATA as a P&S subscriber (Macintosh)

Utilizing P&S, any numeric quantity used in DATA can be defined by a dynamic linkage to another application or to another tree. In this situation, DATA is the subscriber and the other application (or the other tree) is the publisher.

► To capture externally stored values via P&S:

- ▷ Select the item in the external application (for example, a cell in a spreadsheet). Choose Edit > Create Publisher, or the equivalent menu item in the external application. (See below for how to create a publisher in DATA.)
- ▷ In the resulting Save File dialog, give the new edition file a descriptive name. Make sure that this file is saved in an appropriate folder.
- ▷ Switch to DATA, and select Edit > Subscriber List. Press the New Subscriber button, and use the Open File dialog to select the edition file you just saved.

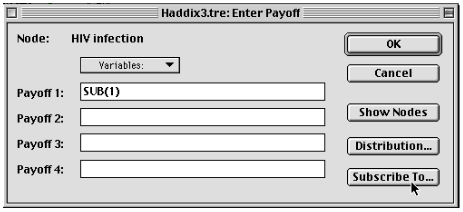


You have just created a subscriber – a link – to the new edition file. Each subscriber is automatically assigned an index number. The index number can be changed by you, as described below.

In DATA, the index number is used in conjunction with the Sub() function to specify a subscription to a particular edition file; the index number is inserted between the parentheses. The Sub() function can be used in DATA wherever a numeric value would be appropriate. There is no limitation on the number of times that the same edition file can be invoked in a DATA model; just be sure to use the Sub() function with the same index each time.

In addition to the index, it is also possible to assign a description to each subscriber, for annotation purposes. To enter a description or to change the index or an existing description, click the Properties button in the Subscriber List dialog.

The Subscriber List dialog has several other buttons. Pressing the Cancel Subscriber button will break the link to the edition file. Pressing the Open Publisher button will close the dialog and open the document to which the subscriber is linked. If you have linked to a spreadsheet, the cell will then be selected; if you have linked to another tree, the publishing node will be selected.



Node: HIV infection

Payoff 1: SUB(1)

Payoff 2:

Payoff 3:

Payoff 4:

DATA also offers an alternative method for creating a subscriber. If your text cursor is inside a formula editor – such as a variable definition, probability field, or payoff entry box – you can select the Edit > Subscribe To command. This will present the Open File dialog for selecting an edition file. Once you select the edition file, DATA will create a new subscriber, assign it an index, and insert the text “Sub(n)” in the editor.

There is also a Subscribe To button in the Define Variables window. This button is an alternative to the Edit > Subscribe To command for creating a new subscriber.

Using DATA as a P&S publisher (Macintosh)

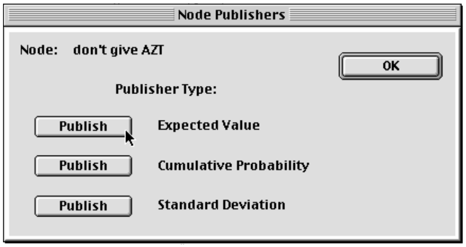
DATA for Macintosh can publish calculated values for use in a P&S client document, including spreadsheets, databases, word processors, and other trees. Each node can export via P&S its expected value, path probability, and/or standard deviation.

► **To use a node's calculated value in another document:**

▷ Select the node whose value you would like to use.

▷ Choose Edit > Publishers.

▷ In the Publishers dialog box, press the appropriate Publisher button to create an edition file for the value you wish to export.



Node: don't give AZT

▷ In the Save File dialog, give the edition file a descriptive name, and save the file where it will be easy to locate.

▷ In the client document (which may or may not be a DATA document), choose Subscribe To (or the equivalent command, depending on the application), and select the appropriate edition file from the ensuing Open File dialog.

DATA will update the calculated values whenever you roll back the tree. The edition file will also be updated when you choose the corresponding calculation from the Analysis menu with the publishing node selected. For example, if you have published the path probability at a designated node, the edition file will be updated when you select that node and choose Analysis > Path Probability.

Chapter 16 Bi-directional Linking

Chapter 15 covered the one-way transmission of calculated values between DATA and another application (or between trees) using DDE (Windows) or Publish & Subscribe (Macintosh). This chapter covers a more powerful and flexible method for exchanging values in Windows called bi-directional linking.

Bi-directional linking in DATA 3.5 for Windows uses ActiveX technology, and works only with Excel 97 or higher spreadsheets. The ActiveX implementation represents a performance improvement over the DDE-based bi-directional linking available in DATA 3.0.

Bi-directional linking is not available in DATA for Macintosh.

Bi-directional links work during any type of analysis (e.g. roll back, sensitivity analysis, Markov analysis, and Monte Carlo simulation). While standard DDE links capture only a single external value, a bi-directional link can provide a different result Any tree variable can be specified as an input to a spreadsheet during bi-directional linking. The resulting, calculated outputs to the tree are limited only by the calculations possible in a spreadsheet: everything from simple mathematical functions to database queries and random number generation.

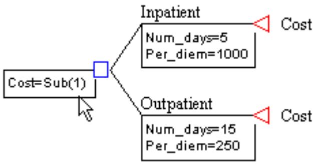
> **TIP:** This chapter documents using bi-directional links in the 32-bit version of DATA for Windows. If you are using the 16-bit version, please see the on-line help file.

Calculating payoffs using bi-directional links

While you may find bi-directional links useful in many situations, the principle use of this feature is to integrate spreadsheet calculations with DATA's payoff calculations.

If you have developed a complex payoff formula in a spreadsheet, you may not have to duplicate it in your tree. Rather than recreate the entire formula in DATA, you can use the spreadsheet as a calculation engine. For each scenario in your tree, DATA can pass node-specific and default variable values (the “inputs” of the formula) to the spreadsheet, recalculate the spreadsheet formulas, and return the final output values to a payoff expression.

Consider the following simple example, based on the sample file Bi-Link tree. Instead of creating the payoff formula Cost = Num_days * Per_diem in the tree, the same formula will be stored in a spreadsheet. For the multiple scenarios in the tree, different definitions of the Num_days and Per_diem variables will be passed to the spreadsheet and used to recalculate the payoff formula.



Note that none of the terminal nodes in the Bi-Link tree have "Sub(1)" as their payoff. Instead, the variable Cost is used for all payoffs, and is defined at the root node using the Sub(1) reference. Based on the description of one-way links in Chapter 15, one would expect that all terminal nodes will have the same payoff value, since all subscribe to the same spreadsheet cell to calculate the payoff. However, with bi-directional links, it is possible to retrieve many different values from the same spreadsheet cell during tree evaluation.

Other uses of bi-directional links

Although designed for complex payoff calculations, DATA's ActiveX-based, bi-directional links also provide a reliable means of exporting calculated node values to Excel. DDE tree-to-spreadsheet links can be used for this purpose, but DDE is a dated, imperfect technology (see Chapter 15 on creating DDE server-side links). Bi-directional links, because they utilize current technology, are more flexible, easier to maintain and update, and substantially more stable than DDE links.

In using bi-directional links to serve expected values from a tree to a spreadsheet, a second, intermediary tree is created. This secondary tree, requiring only a single terminal node, includes a variable defined by a link to the node value of interest in the primary tree, as well as a bi-directional link to export that node calculation to a spreadsheet cell. This setup requires that the primary tree and then the intermediary tree be calculated before the expected value is updated in the spreadsheet.

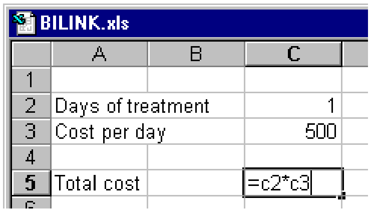
Other potential applications of DATA's bi-directional linkages to Excel spreadsheets include:

- sampling external probability distributions during Monte Carlo simulation;
- dynamically querying databases; and
- recording, cycle-by-cycle, detailed Markov analysis parameters not available in a standard trace report.

Setting up bi-directional links

To see a complete example using Bi-Link tree, you will use a small spreadsheet file called BILINK.XLS. To open this file (and to use bi-directional links) you must have Excel 97 or higher.

All spreadsheet cells which will receive values from, or return values to, DATA must be "named." To see how this is done, select a blank spreadsheet cell. Type a short descriptive name in the name box (to the left of the formula bar) and press ENTER; alternatively, choose Insert > Name > Create.... Use names that will help associate each cell with the appropriate variable created in DATA. In this example, the names "Num_days" and "Per_diem" were assigned to the appropriate input cells in the spreadsheet, and the cost calculation cell was called "Cost." Click on a cell to see its name displayed in Excel's name box.



BILINK.xls

	A	B	C
1			
2	Days of treatment		1
3	Cost per day		500
4			
5	Total cost		=c2*c3

► **To create the initial client-side link:**

▷ In BILINK.XLS, click once on the calculated Cost cell (B5, in the example), to select it.

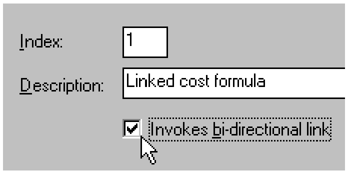
While in the cell, you can look at the simple formula used to arrive at a payoff value for the decision tree.

▷ Choose Edit > Copy to place a link to the calculation cell on the clipboard.

▷ Switch to the Bi-Link tree in DATA, and choose Edit > Links.

▷ In the Links dialog, press the Paste Link button to create a normal, client-side link. Ensure that there is a "+" in the OK column, indicating a confirmed connection between the two documents, as explained in Chapter 15.

▷ Select the link from the list and click on the Properties... button. Ensure that the Invokes Bi-directional Link box is checked.



Index: 1

Description: Linked cost formula

☑ Invokes bi-directional link

With a normal link having been created to the spreadsheet output, the next step is to set up the bi-directional linkage.

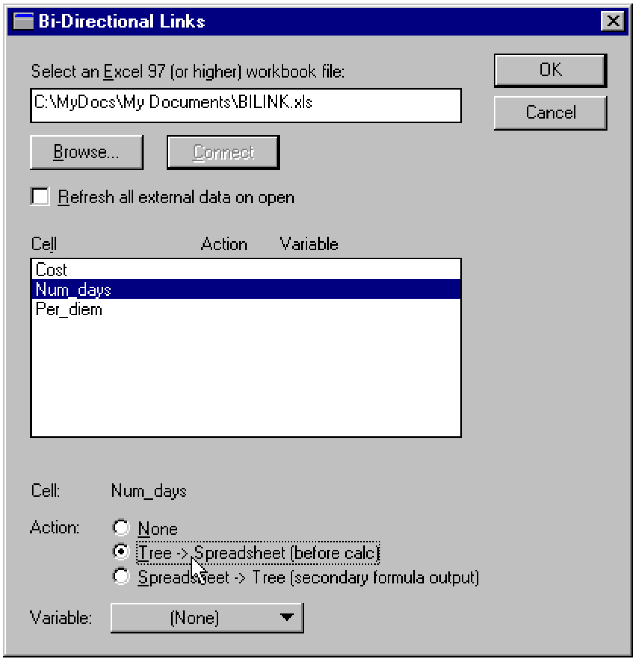
► **To prepare the connection between DATA and the spreadsheet:**

▷ Choose Edit > Links, and press the Bi-directional Links... button.

- ▷ Locate the spreadsheet with which to link. Enter the path for the BILINK spreadsheet, or click on the Browse... button to locate the file. It should be in the ..\DATA\Examples directory (e.g., C:\Program Files\DATA\Examples\).
- ▷ After entering the appropriate path, click on the Connect button. If the spreadsheet has been closed, it will now be reopened.
- ▷ A list of all named cells found in the spreadsheet will appear. *In order for cells to send or receive values during the bi-directional link, they must be named.*

For the bi-directional linkage to work properly, at least one tree variable must provide a calculation input to a target cell.

► To set up the tree-to-spreadsheet output:



Select an Excel 97 (or higher) workbook file: C:\MyDocs\My Documents\BILINK.xls

☐ Refresh all external data on open

Cell	Action	Variable
Cost		
Num_days		
Per_diem		

Cell: Num_days

Action: ○ None ● Tree -> Spreadsheet (before calc) ○ Spreadsheet -> Tree (secondary formula output)

Variable: (None)

- ▷ From the list of named cells, select the cell called Num_days, one of the Cost formula inputs.
- ▷ Three radio buttons indicate the appropriate action for the selected spreadsheet cell: (1) no action, the default; (2) take a variable value and put it in the spreadsheet cell; or (3) optionally pass the spreadsheet cell's value as secondary output into a variable definition. Click on the tree-to-spreadsheet button to specify that the input cell Num_days will receive values from a variable.
- ▷ Finally, from the Variables pop-up list, select the variable Num_days.
- ▷ Repeat the same steps for the other formula input cell, Per_diem, associating it with the Per_diem variable.

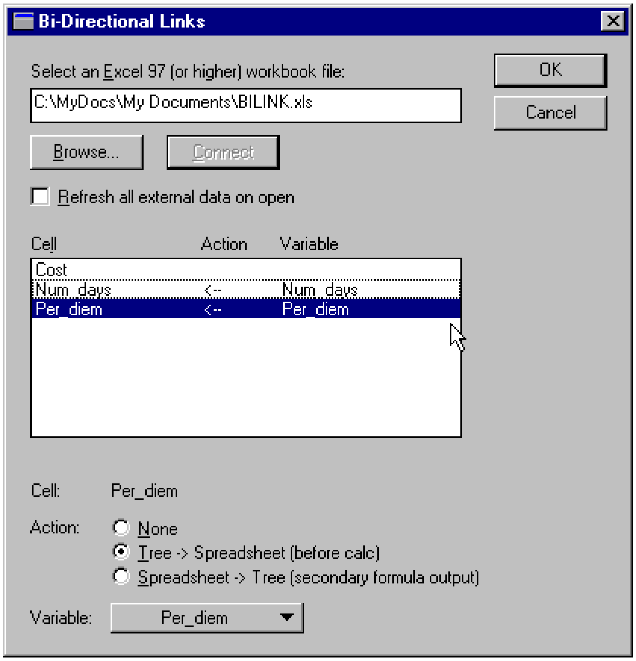
In this example, no secondary spreadsheet-to-tree links are required. These could be set up very simply, though, by associating a cell with a variable, and indicating that this is a spreadsheet-to-tree link. Each such link will define the appropriate variable *at the node being calculated*, using the cell's calculated value.

TIP: Note that you don't use spreadsheet-to-tree links to get the master output of your formula, since you've already created a normal link to that cell. In some cases, you may want to associate additional calculation cells with tree variables to return parameters to the tree, but these spreadsheet-to-tree links are optional.

► **To set up additional spreadsheet outputs:**

▷ Select the associated cell and variable, and choose spreadsheet-to-tree.

▷ Close the Links dialogs, and return to the tree.



Bi-Directional Links

Select an Excel 97 (or higher) workbook file:
C:\MyDocs\My Documents\BILINK.xls

OK | Cancel | Browse... | Connect

☐ Refresh all external data on open

Cell	Action	Variable
Cost		
Num_days	<--	Num_days
Per_diem	<--	Per_diem

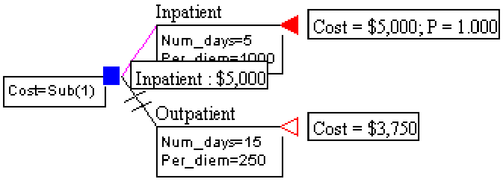
Cell: Per_diem

Action:
○ None
● Tree -> Spreadsheet (before calc)
○ Spreadsheet -> Tree (secondary formula output)

Variable: Per_diem

Calculations under bi-directional links

Now, rolling back the tree will demonstrate that the payoff values have been recalculated by your spreadsheet for each scenario.



Although the payoff at each terminal node is Cost, and Cost is defined default for the tree equal to Sub(1), the calculated values of each scenario differ. Once the bi-directional link is created, all calculations – not just roll back – will utilize the linkage.

If the spreadsheet is closed when you try to calculate the tree, Excel will be started, if necessary, and the spreadsheet opened. If the location of a linked cell is changed on the worksheet, the bi-directional links will not be adversely affected, so as long as the cell names are not changed or lost.

If, on the other hand, the workbook or worksheet is renamed, moved to a new location on your hard drive, or transferred to another computer, the bi-directional link will be broken. You must update the location in the trigger link's properties, as well as in the Bi-Directional Links dialog, and reconnect. See Chapter 15.

Troubleshooting bi-directional links

A tree set up for bi-directional links should have the following structural features:

- A standard client link exists from an Excel cell to the tree. The cell will usually contain the payoff formula which provides calculated output varying from one scenario to another. Unlike standard, DDE links, this link can export many different calculated values from the spreadsheet to the tree during a single tree calculation. The method for setting up this link, with DATA as the client, is described in Chapter 15. After this primary link is created, you must specify in its properties that a Sub() function reference to this link will invoke the exchange between DATA and Excel – this is the bi-directional link "trigger."
- When bi-directional linking is being used not to calculate a payoff formula, but to export node calculations to the spreadsheet, a trigger link is still required to initiate the exchange of values.
- Each unique trigger link is set to invoke the bi-directional link. If, for instance, not all scenarios use the same spreadsheet formula, you will need to copy a link from each calculation cell. Then, you must set each trigger link to invoke the bi-directional calculation.

- A Sub() reference using the trigger link's index appears in the appropriate tree calculation. The link reference is usually made in the payoff expression of terminal nodes requiring Excel's calculation and output (but can be in a Markov reward or probability calculation). The Sub() reference can be entered directly in each appropriate formula; alternatively, it can appear in the definition of a variable used in calculating the scenario.
- At least one variable for export to Excel is associated with spreadsheet cells, and then defined for every scenario in the tree that will use a bi-directional spreadsheet calculation. These variables usually represent various components of the spreadsheet's calculation of a payoff formula. Definitions can be numeric values or more complex expressions; expressions will be evaluated before being passed to the appropriate cell. See Chapter 8 for a discussion of selecting appropriate node locations for definitions.

If you have properly set up the bi-directional link, but it still will not work, the following checklist may help identify the cause:

- You must be using Excel version 7.0 (for Office 97) or later.
- Ensure that, in the Links dialog, a “Y” appears in the Bi-links column for the subscriber to the output of the bi-directional spreadsheet calculation. You must manually specify in the properties of a client link that it initiates the bi-directional link. If you do not select the Invokes bi-directional links checkbox in the appropriate subscriber's properties, DATA will use only the normal client-side linking, causing the same link value to be used for all scenarios.
- Confirm that the bi-directional link-invoking Sub(*n*) reference used in tree calculations has a valid parameter n; check the properties of the link with index n, to be sure it is valid.
- If your spreadsheet file has moved on disk or has been transferred to another computer, or the worksheet name has been changed, the new path must be specified in DATA.

Chapter 17 Exporting Graphics and Analysis Data

This chapter deals with exporting graphics and certain underlying information from DATA to another application. First, different methods are discussed for transferring a picture of your document – whether a tree, influence diagram, line graph, or area graph – into presentation or word-processing software. Next, exporting the numeric information underlying any one of DATA’s line, area, or bar graphs is covered.

Exporting pictures

If you want to include a picture of a DATA model or graph in a report created in another program, you must export a picture of the DATA document. Using the Snapshot and Copy commands, pictures of trees, influence diagrams, and graphs can be saved as graphics files or copied onto the clipboard.

In DATA for Windows, two formats are available for exported pictures: metafiles and bitmaps. Metafiles are the preferred format, as they provide high quality images both on screen and when printed.

A *metafile* stores an image in vector format. It saves the actual drawing commands as objects, which are reproduced in your word processor or presentation program, and on your printer. A metafile generally requires less memory than a bitmap, and it will always print at the highest resolution available, but fewer applications recognize metafiles than recognize bitmaps. Depending on the program used to open the metafile, you may be able to edit the picture using standard drawing and text formatting tools. In addition, programs like Microsoft® Word™ and PowerPoint™ will allow you to stretch or shrink a metafile without losing any information.

A *bitmap*, on the other hand, stores an image in raster format. It saves the actual pixels used to create the picture of your document. To edit a bitmap you must work at the pixel level; this requires the use of painting tools. Bitmaps print only at screen resolution even if your printer is capable of a much higher resolution, so both text and graphics are likely to have jagged edges. Stretching or shrinking a bitmap will rarely produce a pleasing result. The principal advantage of bitmaps is that they are compatible with more applications than are metafiles. Your use of bitmaps should be limited to situations where it is not practical to use a metafile.

In DATA for Macintosh, all pictures of trees, influence diagrams, and graphs are exported in PICT format. Like a metafile, PICT is a vector format.

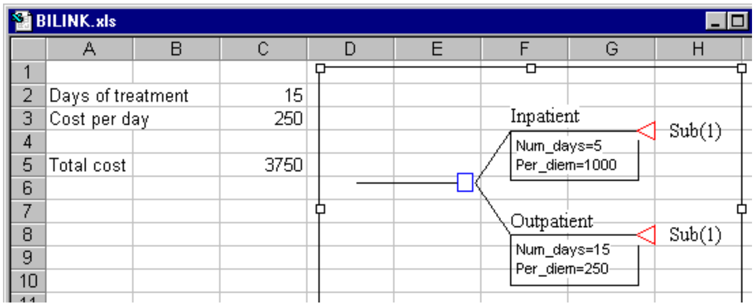
In both DATA for Windows and DATA for Macintosh, you have the option of transferring pictures either via the clipboard or by saving a snapshot of your document to a graphics file. Note the following differences between the results of the two methods:

- If you want to export only a single subtree, instead of an entire tree, you must use the clipboard method.
- Exporting a picture of a tree or subtree over the clipboard will not capture unbound annotation boxes or arrows.
- Exporting a picture of a tree or subtree over the clipboard will show all nodes as being selected. Use the snapshot method to avoid this result.

Exporting a picture using the clipboard

Before a picture of the tree or subtree can be placed on the clipboard, you must select it in the tree window. Click on either the tree's root node or an internal node, and choose Options > Select Subtree. With an influence diagram or graph, it is not possible to copy only a part of the document – a picture of the entire document will be created regardless of what is selected.

The command located immediately below Copy in the Edit menu is used for clipboard export. It may read Copy as Bitmap/Metafile (Windows) or Copy as PICT (Macintosh), in which case you should simply select it. In other situations, it may read Copy Special..., in which case you should choose it. In the ensuing dialog, select the radio button Copy as Bitmap/Metafile (Windows) or Copy as PICT (Macintosh).



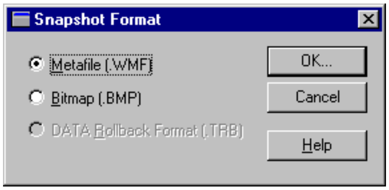
BILINK.xls

	A	B	C
1			
2	Days of treatment		15
3	Cost per day		250
4			
5	Total cost		3750
6			
7			
8			
9			
10			

After copying the picture of your document, switch to your presentation program and choose Edit > Paste (or Edit > Paste Special). Consult that program's documentation for more information; some programs use the word "picture" to refer to metafile or PICT formats.

Exporting a picture to a file: Snapshots

Choose File > Snapshot to export a picture of the active document to a file. Next, pick the format of the file: metafile or bitmap (Windows); PICT (Macintosh); or TRB (see below). Then, using the standard Save As dialog box, choose a location to store the snapshot file.



● Metafile (.WMF)
○ Bitmap (.BMP)
○ DATA Rollback Format (.TRB)

Word processing and presentation programs which can read .WMF (Windows metafile), .BMP (bitmap - Windows) or PICT (Macintosh) graphic files often include a command called Insert > Picture > From File. Consult the other application's relevant documentation for more information on placing graphic files.

A snapshot of a tree contains all of the tree's elements, including all note boxes and arrows. To export a picture of only a portion of a tree, use the Edit > Copy Special… command. Refer to Chapter 10 for information on customizing the tree window display before exporting a picture.

With the exception of .TRB files (see below), DATA cannot open any of the graphic files it exports.

DATA rollback (.TRB) files

A .TRB file is an uneditable copy of your rolled-back tree. You may use this type of file to distribute a viewable representation of your calculated tree. The file type is proprietary; TRB files can be opened only by a version of DATA 3.5 (including the trial/demo software).

When you open a TRB file, you will see the tree already rolled back. The values and structure of the tree may not be changed, but you may change the display settings. For instance, you may collapse and expand subtrees, or alter the numeric format of the calculated values.

TRB files are useful for distribution of results to other DATA users. Users immediately see the results of the roll back analysis, and can change the display formatting, if necessary. They will not be able to make changes to the tree's structure or values, though. See Chapter 18 for information on other methods of distributing DATA decision trees.

► **To create a TRB file:**

▷ Roll back your tree.

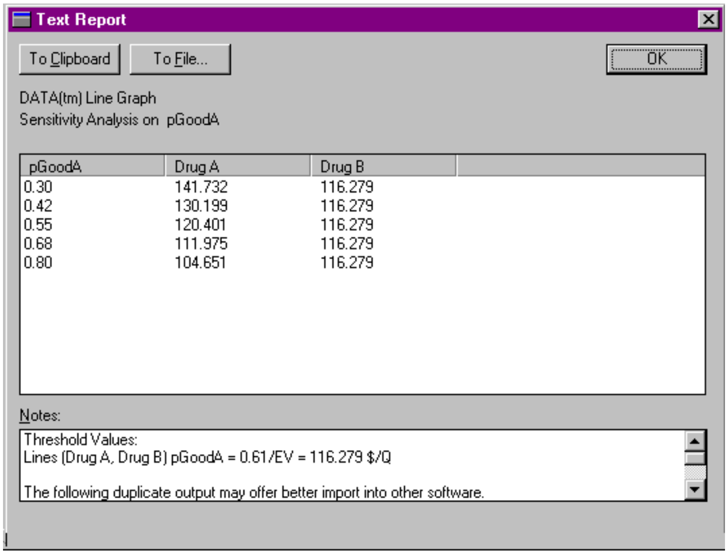
▷ While the tree is still rolled back, choose File > Snapshot.

▷ In the Snapshot Format dialog, select the button named DATA Rollback Format (.TRB).

▷ Save your TRB file using the standard Save As dialog box which appears. In DATA for Windows, the .TRB extension will be appended automatically.

Exporting graph data

You can export the numeric values that underlie any graph created with DATA, with the exception of animated, three-way sensitivity analysis graphs. With the graph in the active window, choose Graph > Text Report to view these values. From the Text Report window, you can copy the information to the clipboard using the To Clipboard button or command, sometimes found under the Export pop-up menu. You can also save the information to a text file using the To File button or command.



DATA(tm) Line Graph
Sensitivity Analysis on pGoodA

pGoodA	Drug A	Drug B
0.30	141.732	116.279
0.42	130.199	116.279
0.55	120.401	116.279
0.68	111.975	116.279
0.80	104.651	116.279

Notes:

Threshold Values:
Lines (Drug A, Drug B) pGoodA = 0.61/EV = 116.279 $/Q

The following duplicate output may offer better import into other software.

The identical information can be placed on the clipboard by choosing Edit > Copy Special. In the ensuing dialog box, choose the radio button Copy as spreadsheet-accessible text.

The numeric data are exported in tab-delimited format, in an arrangement that is easily imported into other programs for graphing or analysis.

CHAPTER 18 BUILDING CUSTOM DATA APPLICATIONS

Public and private sector organizations are expanding their use of decision analysis. In particular, they are looking for new ways to make decision analysis accessible, as well as more powerful.

DATA offers a number of tools for building sophisticated, user-friendly applications based on DATA decision trees. This chapter describes two alternatives: DATA Interactive™ and Run-time DATA. Both tools are designed to ensure that decision analysis applications can reach a variety of audiences in appropriate formats.

DATA Interactive is a separate product available from TreeAge Software, Inc. This chapter will provide a short overview of its capabilities. Since the functionality of building interfaces for use with Run-time DATA is included in DATA 3.5, it is covered here in detail.

DATA Interactive™

DATA Interactive is a calculation engine for analyzing models created in DATA. You design a user-friendly interface to the DATA model using a programming technology of your choice (HTML and Visual Basic being the most popular), and link it to the DATA Interactive calculation engine using a simple set of programming objects, methods, and properties.

DATA Interactive makes it feasible for all the decision makers in your organization – even those without expertise in decision theory or model building – to perform sophisticated decision analysis remotely, over an intranet or the Internet. Alternatively, your decision analysis application can be distributed on CD-ROM (or other media) to a targeted audience.

For example, DATA Interactive makes it feasible to:

- continually monitor and analyze R&D projects and their associated uncertainties, including costs, timing, market characteristics, and return on investment.

- incorporate expert knowledge and client requirements into models evaluating insurance risks and premiums; and
- apply cost-effectiveness models developed during drug research and trials to subsequent marketing efforts, giving a pharmaceutical sales force a dynamic tool able to take into account customer-specific costs and alternative treatments.

Additional details and a demo of the DATA Interactive software are available at the TreeAge Software web site, or by contacting TreeAge Software directly.

Technical Specifications

DATA Interactive is an ActiveX control designed to run on a Windows NT server. It complies with industry standard protocols for exchanging data between applications and can be accessed on an intranet or web server. Since all processing can be done on the server side, there is no restriction on the choice of browser used to access your models.

DATA Interactive can also function as a stand-alone control in any 32-bit Windows environment. This makes it possible, for example, to build an application around a decision tree for distribution on a CD-ROM, using Visual Basic as the host application.

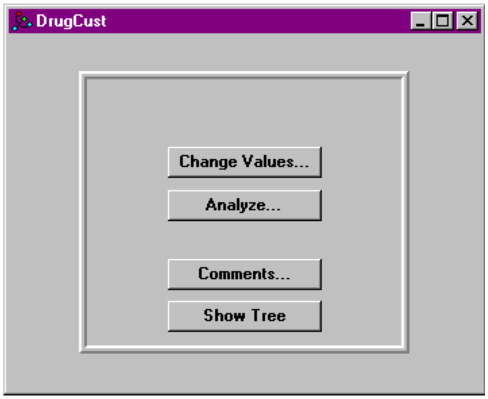
Run-time DATA

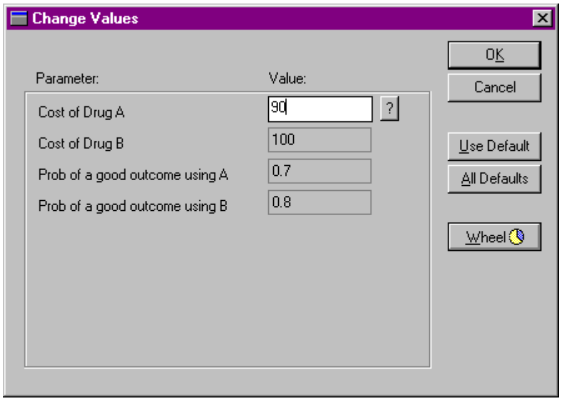
Run-time DATA offers another means of distributing decision analyses to clients (or others in your organization). DATA's built-in Custom Interface feature lets the modeler customize a simple screen in a matter of minutes; all customization is done in DATA, and stored with the decision tree. Creating a custom, run-time interface does not require familiarity with any programming languages and tools; it is very useful for applications which do not require DATA Interactive's powerful Internet/intranet capabilities or its complete interface customization features.

Run-time DATA makes it easy to distribute your model in a format that anyone can use. Licensed copies of the run-time software can be distributed on floppy disks or a CD-ROM, along with all appropriate models (and any tables and linked spreadsheets). Simplified model windows allow for changing selected parameters and performing preset analyses.

The first step, of course, is to build the necessary model or models. This may include creating links to spreadsheets. Once complete, backup copies of the DATA decision trees should be stored before adding Custom Interface features; the Custom Interface will become part of the tree file.

The model builder chooses between two modes, Basic and Extended, when designing a tree's Custom Interface. This choice should be based on the sophistication and needs of the ultimate user (the “client”).





Parameter:	Value:
Cost of Drug A	90
Cost of Drug B	100
Prob of a good outcome using A	0.7
Prob of a good outcome using B	0.8

The Basic mode is primarily for clients who do not need (or should not be allowed) to interact with the tree directly. When a Basic mode file is opened with the run-time version of DATA, a gray window (the “Basic Client Window” or “BCW”) with several buttons will appear. The two most important are the Change Values and Analyze buttons. By clicking on the Change Values button, a client can change specified parameters in your model by entering the new values in a simple list, without having to select the appropriate node and open a Define Variable window. The Analyze button will bring up a list of analyses prepared (i.e., run and stored) by the model builder for use by the client. The run-time user may be permitted to view the tree structure, but you can specify what, if any, Analysis menu items will be enabled.

The Extended mode is for clients who are more savvy about decision trees. When an Extended mode file is opened with Run-time DATA, the tree is in full view. In contrast to the Basic mode, where a single Change Values dialog applies to the entire tree, each node can have its own Change Values dialog box for selected definitions located at the node.

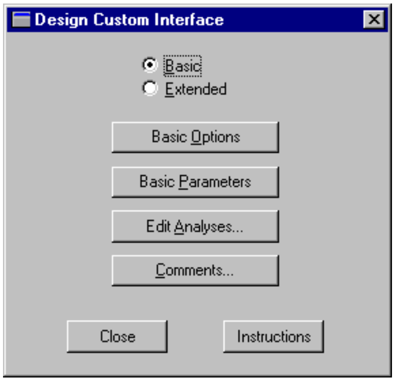
In either mode, analyses stored by the model builder may be run by the client. In the Extended mode, it is also possible to allow clients to run certain of their own analyses; the model builder can decide which Analysis menu commands are to be disabled and which are to remain active.

Designing a Custom Interface for one of your trees is quite simple:

- Set the display and analytical privileges you want your client(s) to have.
- Decide which variables in your model can be modified by clients. For each such variable, you may need to enter additional information for use in the Custom Interface.
- If you are designing an Extended mode interface, turn on the display of variables in the tree under Edit > Preferences…. Clients double-click on variables display boxes to modify model parameters.
- Prepare the analyses that you want clients to perform. Analyses stored (in the Analysis > Storage submenu) will be available for recall by clients.
- Save your file; the custom interface will be saved with the tree.

Creating a basic custom interface

The Basic mode has a single Change Values dialog. It applies only to numeric values stored as defaults for the entire tree. Thus, any parameters that will be modified by the client must be stored by the model builder as numeric default values at the root node. In designing your model, remember that the client using the basic interface will only be able to directly modify *numeric definitions at the root node*; no non-numeric definitions. Thus, if you are modifying an existing tree for use as a run-time tree, additional variables, as well as modifications to existing variable definitions, may be required.

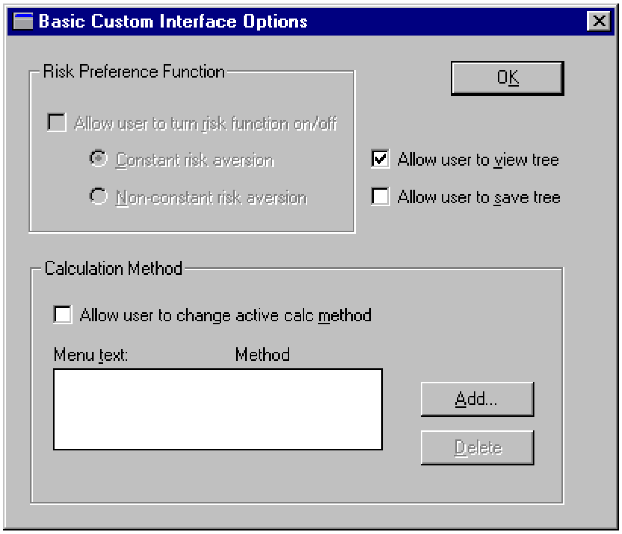

Design Custom Interface

● Basic
○ Extended

TIP: Remember to maintain backup copies of your original tree, without the Custom Interface features. The Custom Interface becomes a permanent part of a tree file.

To begin, choose Options > Design Custom Interface. Ensure that the Basic radio button is selected and click the Basic Options button. The Basic Custom Interface Options dialog box shown on the next page provides options for allowing run-time users to:

- view the tree;
- save changes to the tree;
- turn a risk preference function on or off (if entered by the model builder); and
- change the calculation method (e.g. from simple to multi-attribute) and active payoff.



Basic Custom Interface Options

Risk Preference Function

□ Allow user to turn risk function on/off

● Constant risk aversion

○ Non-constant risk aversion

☑ Allow user to view tree

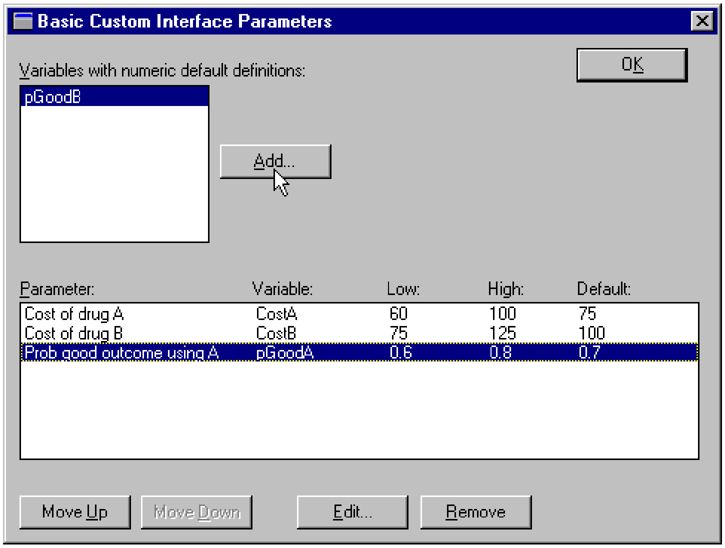
□ Allow user to save tree

Calculation Method

□ Allow user to change active calc method

Menu text: Method

If you opt to allow clients to change the active calculation method, they will see a pop-up menu in the BCW that will allow them to select from the items you enter in this dialog. The client's current selection for calculation method does not change the list of available analyses, so you must indicate for each stored analysis any dependence on a specific calculation method. In some cases, separate trees may be required to handle multiple attributes.



Basic Custom Interface Parameters

Variables with numeric default definitions:

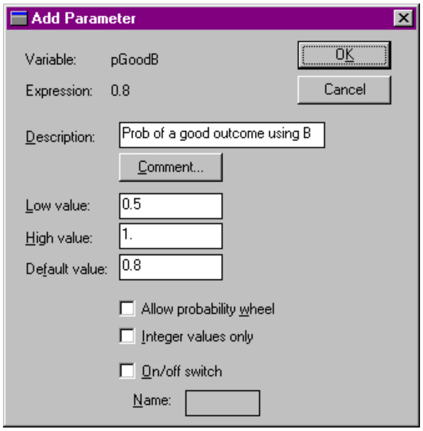
pGoodB

Parameter:	Variable:	Low:	High:	Default:
Cost of drug A	CostA	60	100	75
Cost of drug B	CostB	75	125	100
Prob good outcome using A	pGoodA	0.6	0.8	0.7

To select default values that should be accessible to the client, return to the main Design Custom Interface window and click the Basic Parameters button. The list at the top of the window shows all variables with default definitions. From this list, select the variables that may be modified by the client and click the Add… button. After you complete the information in the Add Parameter dialog, the variable will be added to the table at the bottom of the main Parameters dialog.

You may change the order of presentation by selecting an item in the bottom list and clicking either the Move Up or Move Down button. If you click Remove, the selected parameter will no longer be included in the list of modifiable values, and will return to the top list. Clicking Remove will not remove that variable from the underlying tree.

The description, comment, low value, high value, and default (baseline) value are taken from the Properties dialog for variables (see Chapter 9 for details), but are maintained separately. This information can be changed when a parameter is first added to the parameter list or, later, by selecting the parameter and clicking the Edit… button.



Variable: pGoodB

Expression: 0.8

Description: Prob of a good outcome using B

Low value: 0.5

High value: 1.

Default value: 0.8

□ Allow probability wheel

□ Integer values only

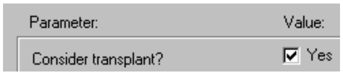
□ On/off switch

Name:

- **Description**: The text you enter here will be seen by the user in the Change Values dialog, instead of the variable's name. If you have previously entered a short description for the variable (in the Properties dialog), that description will be used by default.
- **Comment**: If you have entered a longer comment for the variable (in the Variable Properties dialog), the user will see a button labeled with a "?" to indicate that more information is available for a particular parameter.
- **Low/High values**: These specify the range of values that you will allow clients to enter.
- **Default value**: The client always has the option of resetting a given parameter to its default value. The value you enter here will be the value to which a parameter will revert upon the client's request. Note that the default value is not used for initialization. When a client first opens the model, the initial values are those that are saved in the tree. If you allow clients to save the tree, whenever the tree is reopened, the initial values used are those that were saved, not the default values.
- **Allow probability wheel**: If the low and high values are 0 and 1, you may allow clients to enter this parameter by using the wheel. Unlike the wheel when it is called directly from the tree window, this option will work only on one value at a time. Thus, if you have a chance node with more than two branches whose probabilities you would like to allow users to change, you have little control over the coherence of the probabilities.
- **Integer values only**: If this box is selected, only integers will be allowed. Otherwise, any real number in the specified range is accepted.
- **On/off switch**: If you select this option, clients will not see a numeric editor in the Change Values dialog. Instead, a check box will be displayed with the name you enter. For instance, if

you enter the information shown at left in the Add Parameter dialog, clients will see the following:

The client’s selection will be stored as 1 if the check box is selected, and 0 if it is cleared. The default value must be 0 or 1 to indicate the initial state of the check box.



Parameter: Value:

Consider transplant? ☑ Yes

Completing the basic custom interface

As explained above, the analyses that will appear in the run-time interface list are those that you have saved using the Analysis > Storage submenu. See Chapter 13, Analysis Storage, for detailed information on this topic. Note that your basic custom interface will not become active until you store at least one analysis in the tree; instead, the user will see the tree and menus, as in an extended interface.

Every analysis available in the Analysis menu may be stored for use from the Basic Custom Interface except Graph Risk Preference Function, Show Optimal Path, Verify Probabilities, and Roll Back. If you set the Basic Options so that the user can see the tree, they will have the option of running any of these analyses from the Analysis menu. In order to deactivate the Analysis menu when the tree is displayed, the designer must temporarily switch the tree to extended interface mode and disable the Analysis menu items there.

To view the Basic Custom Interface, simply select Options > Show Custom Interface. This will activate the BCW. As a designer, you will always have access to the Show Tree button, even if you have not enabled this option for your clients. To fully test the interface, open the tree using a run-time version of DATA.

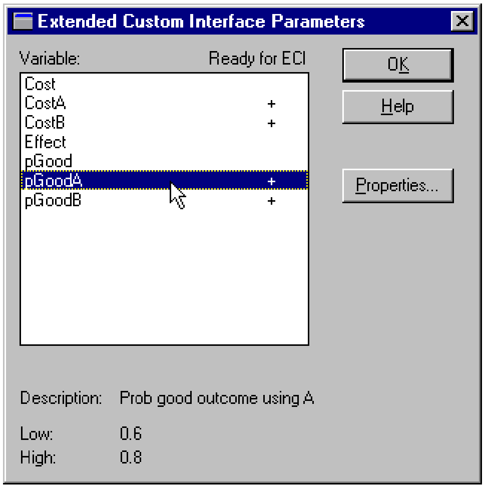
Creating an extended custom interface

When you select Extended Options from the Design Custom Interface dialog, you will be able to select which analyses clients will be able to perform manually. Typically, you should prepare analyses using the Analysis > Storage menu, but you may wish to allow clients to experiment with performing some simpler analyses on their own. In this event, you should consider disabling the sensitivity analyses, threshold analysis, and tornado diagrams, as these may require considerable knowledge of your model and DATA’s interface. This will avoid the risk that your client will get a result that has the appearance of being correct, but is not.

From the Design Custom Interface dialog, select the Extended radio button to begin designing the interface. Press the Extended Parameters button, to see a list of all variables in your tree. A variable will be indicated as ready for use in the Extended Custom Interface if all of the following apply:

- It is shown in the tree.
- It has an assigned range of values. The low and high values will be used to check the validity of the user's input.
- It has a short description.
- It is *not* a Monte Carlo tracking variable.

You may verify these characteristics (or set them) in the Properties dialog box, which you can access by pressing the Properties button either here or in the Define Values dialog box. If all of the above criteria are met, a "+" symbol will appear in the Ready for ECI column in the list box of the Extended Custom Interface Parameters dialog. To remove a variable indicated as ready for ECI from the list, simply change its properties to specify that it not be shown in the tree.



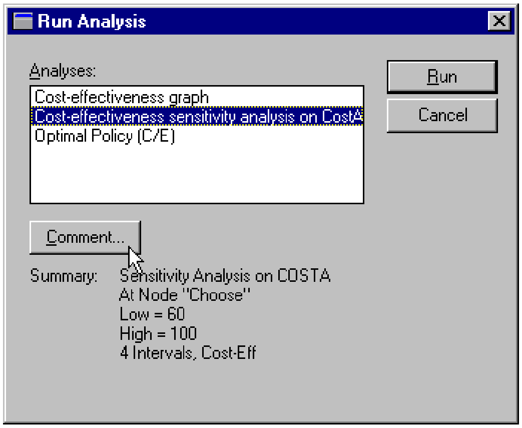
Variable:	Ready for ECI
Cost	
CostA	+
CostB	+
Effect	
pGood	
pGoodA	+
pGoodB	+

Description: Prob good outcome using A

Low: 0.6
High: 0.8

The only definitions of variables which may be modified by clients are those which are numeric. If, for example, the variable TestCost is designated Ready for ECI, clients will be able to use the Change Values dialog box to modify any numeric definition of TestCost. Any instances where TestCost is defined in terms of another variable, or where the definition consists of an expression, will not be modifiable.

These limitations are not necessarily imposed when the user runs analyses directly from the Analysis menu, if you so permit in the Options dialog, discussed above. For example, if you permit the client to select Analysis > Sensitivity Analysis..., the client will be able to run a sensitivity analysis on any variable in the tree, even those with non-numeric definitions.



Analyses:
Cost-effectiveness graph
Cost-effectiveness sensitivity analysis on CostA
Optimal Policy (C/E)

Summary: Sensitivity Analysis on COSTA
At Node "Choose"
Low = 60
High = 100
4 Intervals, Cost-Eff

When an Extended mode tree is opened by a client using the Run-time DATA, the full tree appears, not the minimal BCW interface associated with the Basic mode. To run analyses previously prepared and stored by the designer, clients will need to choose Analysis > Storage > Run. The resulting dialog will be the same as that shown above, for the Basic Custom Interface. In addition, clients may be allowed to perform their own analyses, as described above.

Changing values in the extended custom interface

To open the Change Values dialog box at a particular node, the client must double-click inside the variables box at that node. The variables box is the list of variables that is displayed below the node name when the model builder has elected to show variable definitions in the tree. (To use the Extended Custom Interface, this flag *must* be set; see the Variables Display page of the Preferences dialog.) Not all variables that appear in the variables box will be editable in the Change Values dialog box, nor will all nodes that have a variables box necessarily display a Change Values dialog box. See the section about parameters, above.

Clients will not be able to modify probabilities directly. If you wish to allow clients to change probabilities, you must either set up the appropriate probability variable for use in the Change Values dialog, or enable the probability wheel (discussed below). All probabilities will be displayed as their numeric equivalents, even when the tree is not rolled back.

The probability wheel can be very useful in the Extended Custom Interface. In order to enable its use at a particular chance node, you must ensure that at least one branch has a probability stated as a variable that has been prepared for the Extended Custom Interface. This variable's definitions need not be shown in the tree. Clients using Run-time DATA will not be able to change the storage location for probability values. See Chapter 14 for detailed instructions on using the probability wheel.

T esting the extended custom interface

When building a model’s Extended Custom Interface, you may want to test it from time to time to see how it will look when opened with Run-time DATA. This is possible from within the full version of DATA. Simply pull down the Options menu, and choose Mimic Run-Time. If you set this option, DATA will enable and disable menu items exactly as it does for users of Run-time DATA. To turn off run-time emulation, simply repeat the same menu selection.

Protecting your intellectual property

If your model is going to be distributed to users outside of your organization, you may want to take a few additional steps to help protect the model’s intellectual property content.

It is possible to prevent anyone from using the full version of DATA to open your model. To do this, you specify that a particular model can be opened only with Run-time DATA. You may also specify a start-up message (e.g., a license agreement) to be displayed when the model is opened.

Any attempt to open a run-time-only tree with the full version of DATA will generate a message that the file can be opened only with Run-time DATA. As described above, you can use the Custom Interface to limit what aspects of the model users of Run-time DATA can view or change.

This level of protection may be desirable even if you are not concerned with protecting intellectual property, but only with ensuring that users not misuse or make undesirable changes to the model.

Users of Run-time DATA will be presented with your license agreement or other start-up message. If the terms are not accepted, the model will not open.

For information on how to implement these features, please visit our web site at http://www.treeage.com.

Chapter 19 Specifying what DATA Calculates

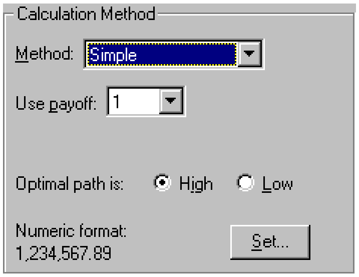
A tree may be calculated and evaluated in a variety of ways, depending on your objectives. For a given tree, you can specify the calculation method (using a single attribute, or payoff, at a time, or one of the various forms of multi-attribute calculations), the optimal path criterion, and the quantity to be calculated during roll back (expected value, path probability, or *maximin*). These options are explained below.

Changing the calculation method

The calculation method represents the criteria used to select or combine payoff sets (or Markov rewards) in calculating your tree. There are four types of calculation method, which are grouped into two classes:

- Single-attribute calculations are handled by the Simple calculation method.
- Multi-attribute calculations are handled by the remaining three calculation methods: Cost-Effectiveness, Benefit-Cost, and (Generalized) Multi-Attribute.

DATA allows you to assign up to four attributes for the scenarios in a tree. Normally, your initial work will be based on one attribute at a time (e.g., profit) using a single payoff or attribute. For some models, these simple calculations may be all that you require. For other models, you will eventually need to combine two or more of the attributes during a single calculation, such as in cost-effectiveness analysis.



Calculation Method

Method: Simple

Use payoff: 1

Optimal path is: ● High ○ Low

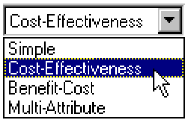
Numeric format: 1,234,567.89

► **To view the current calculation method:**

▷ Choose Edit > Preferences. By default, the Calculation Method page of the Preferences dialog will be visible.

It is also possible to see the currently selected calculation method without opening the Preferences dialog. The status bar, displayed at the bottom of the DATA window, provides basic information about the active tree's setting. If the tree is currently using the Simple calculation method, then "Payoff n" will appear (where n is the currently active payoff number). The multi-attribute calculation methods also display important contextual information in the status bar, which is described further in Chapter 20.

► **To change the calculation method:**



- ▷ With the Preferences dialog open, select the appropriate calculation method from the Method pop-up menu.
- ▷ If you have chosen the Simple method, specify which payoff should be used. If you have chosen one of the multi-attribute methods, other information is required. See Chapter 20, Multi-Attribute Analysis, for details.

After closing the Preferences dialog, with the tree selected, the new calculation method will now be displayed in the status bar.

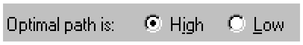
Changing the optimal path criterion

To select the optimal path at each decision node, DATA must know what the decision maker’s goal is (for example, to maximize value or minimize cost). This is accomplished by setting the tree's optimal path criterion.

For a tree that maximizes, such as one whose payoff formula is in terms of profit, DATA selects as its optimal strategy the alternative with the *highest* numeric value. For a tree that minimizes, such as one whose payoff formula is in terms of costs, DATA selects the alternative with the *lowest* numeric value. A special set of parameters is used to determine an optimal path in a cost-effectiveness tree.

► **To set the optimal path criterion:**

- ▷ Choose Edit > Preferences. By default, the Calculation Method page of the Preferences dialog will be visible.
- ▷ Ensure that your calculation method is set correctly, as described above.
- ▷ Select the appropriate radio button next to the text labeled Optimal path is. Select the High button for trees that maximize, or the Low button for trees that minimize.



Optimal path is: ● High ○ Low

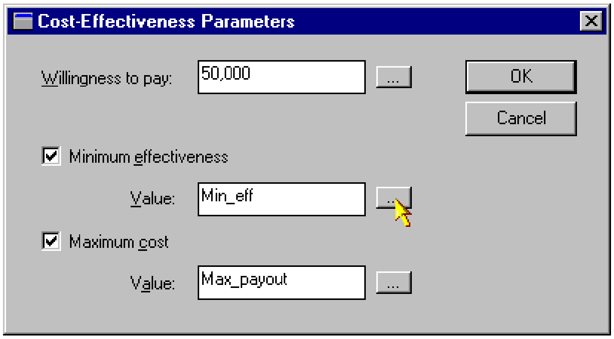
- ▷ Press OK.

Under the Simple calculation method, each of the four payoffs may have its own optimal path criterion. For example, if your tree models four payoffs, each may be separately set to maximize or minimize.

If your tree uses the generalized multi-attribute calculation method, you must specify the optimal path criterion. The benefit-cost calculation method uses a fixed optimal path criteria – set to maximize.

Cost-effectiveness optimal path parameters

Evaluating an optimal path under the cost-effectiveness calculation method is more complex than simply selecting the high or low path. By specifying a set of cost-effectiveness parameters, a variety of objectives can be included in the evaluation.


Cost-Effectiveness Parameters

Willingness to pay: 50,000

☑ Minimum effectiveness

Value: Min_eff

☑ Maximum cost

Value: Max_payout

► **To set the optimal path parameters for cost-effectiveness calculations:**

- ▷ Choose Edit > Preferences.
- ▷ Ensure that your calculation method is set to cost-effectiveness.
- ▷ Click on the CE Params… button and enter the required information in the dialog.
- ▷ Press OK.

The cost-effectiveness optimal path parameters are covered in more detail in Chapter 21.

Reversing the optimal path

If you need to force a single decision node to evaluate based on the opposite criterion from other decision nodes in the tree, see the discussion of changing the optimal path, found in Chapter 14.

Changing the quantity to be calculated

DATA can display several different quantities in the boxes to the right of each node during roll back. Typically, DATA will display the expected value of each node and, for those terminal nodes in the optimal path, the path probability.

► **To specify the quantity to be calculated during rollback:**

- ▷ Choose Edit > Preferences.
- ▷ Select the Roll Back page from the list of categories on the left.

▷ Select the appropriate button from the radio group labeled Roll back calculates.

The other quantities that can be displayed during roll back (other than expected value) are described below.

Custom columns: Columns of information of your choosing can be displayed at each endnode. For example, you may display not only the cost-effectiveness ratio, but also the marginal values, individual cost variables, etc. See Chapter 10.

Payoffs only: Only the values of terminal nodes will be displayed. Optimal paths will be calculated and indicated using the usual hash marks and colored lines, but no expected values will be displayed.

Path probabilities: This option suppresses both the calculation and display of expected values. The calculation of path probabilities does *not* take into account the optimal path. Thus, the sum of the path probabilities of all terminal nodes in the tree will be greater than 1.0 (if your tree has at least one decision node).

Maximin: This option will consider the most pessimistic possibility at each uncertainty, regardless of probabilities. At each decision point, the “best of the worst” is selected.

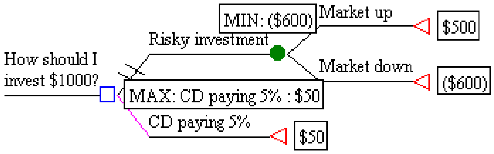
Maximin roll back

Here, specifically, is how a tree is rolled back under Maximin:

- The value assigned to every chance node is equal to the worst (least optimal) value of any of its potential outcomes. Probabilities are ignored, and may be omitted.
- The value assigned to every decision node is equal to the best possible selection of alternatives, as usual.

Maximin should be treated as an additional perspective and certainly not as a substitute for expected value calculations. Maximin calculations assume a pessimistic view of events. They are based on the principle that one should deal with risk by identifying the worst possible outcome for each scenario and then selecting that scenario which yields the best of these bad outcomes.

On the following page is a picture of Stock Tree when rolled back using Maximin calculations. The value of the uncertainty, Risky investment, is taken from the lowest possible value, which is -600. The decision then maximizes (as usual) between the values of its branches.



The approach taken in Maximin calculations may entail certain drawbacks. It is likely that opportunities which might be identified by expected value calculations will be rejected by Maximin calculations, due to the latter's focus on the most adverse outcomes.

However, the Maximin approach can sometimes be useful during early stages of model development, because it is possible to apply Maximin calculations to a tree before probabilities have been assigned.

When a tree is rolled back under Maximin, the value in the box next to each chance node will be prefaced with MIN: and the value next to each decision node will be prefaced with MAX:, in order to indicate the operation being performed. These prefixes are switched if the optimal path is set to low (minimization).

Chapter 20 Multi-Attribute Analysis

DATA offers three different varieties of multi-attribute calculations. Two options are likely to be of particular interest to public sector and healthcare analysts: cost-effectiveness analysis and benefit-cost analysis. Cost-effectiveness analysis is particularly robust in DATA. A third option, generalized multi-attribute calculations, enables you to specify the relative weights for a maximum of four attributes (see the tip below for information on setting up more than four attributes).

Trees evaluated on the basis of a single attribute are referred to as being calculated using the Simple calculation method. When DATA is shipped, the default method of calculation for each tree is Simple (using Payoff 1). You can easily change this default setting, but it is probably better to change the calculation method for individual trees, as needed. See Chapter 19 for information on changing the calculation method.

Each of the three types of multi-attribute calculations has a separately maintained numeric format, independent of the four simple payoffs' numeric formats. Thus, when you change the calculation method in the Preferences dialog, you should also change the numeric format to be used for the combined calculated results.

If you use Markov processes, be sure to refer to the section on assigning Markov rewards in Chapter 25 for specific information on how to combine multi-attribute modeling with Markov analysis.

TIP: Cost-effectiveness is the only multi-attribute calculation method which strictly requires that you use the methods described in Chapters 19–21. Both benefit-cost and generalized multi-attribute models can be set up, for example, using only a single, complex payoff expression (e.g., set payoff 1 equal to total_Benefit - total_Cost). DATA simply provides functionality which makes multi-attribute modeling easier.

Setting up a multi-attribute model

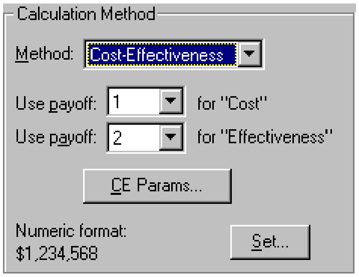
The first step in preparing a multi-attribute model is setting the calculation preferences.

► **To set up a tree for multi-attribute modeling:**

▷ Choose Edit > Preferences. By default, the Calculation Method page will be active. The rest of this procedure details the proper settings in this page of the Preferences dialog.

▷ From the Calculation Method pop-up menu, choose either Cost-Effectiveness, Benefit-Cost, or (Generalized) Multi-Attribute. This tells DATA which specific form of multi-attribute analysis you will be using.

▷ If you are using Cost-Effectiveness or Benefit-Cost, select two payoff numbers to represent, respectively, the two attributes in your tree. The set numbers you select are not important, but you must be consistent. In other words, if you choose payoff 1 to represent costs, it is critical to ensure that, at every terminal node, the first entry (the entry for payoff 1) contains costs.



Calculation Method

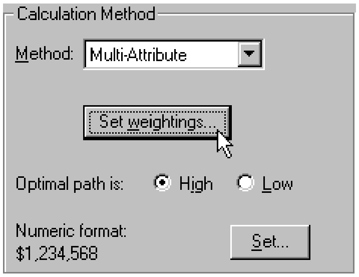
Method: Cost-Effectiveness

Use payoff: 1 for "Cost"

Use payoff: 2 for "Effectiveness"

Numeric format: $1,234,568

▷ If you are using Cost-Effectiveness, set the optimal path criteria by clicking the CE Params… button and entering the required parameters. See Chapter 19 on setting the optimal path criterion and Chapter 21 on cost-effectiveness modeling and analysis.



Calculation Method

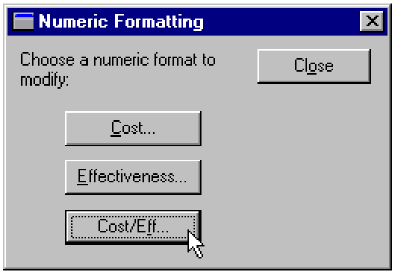
Method: Multi-Attribute

Optimal path is: ● High ○ Low

Numeric format: $1,234,568

▷ If you are using Generalized Multi-Attribute, you will have to enter an appropriate formula for combining the individual attributes into a single quantity. DATA requires the formula to be a linear combination of attributes; see the section on Generalized Multi-Attribute models at the end of this chapter.

▷ If you are using Generalized Multi-Attribute, set the optimal path to low or high, as you would for a single-attribute model. This will specify whether DATA should choose lowest or highest values at each decision node. If you are using Benefit-Cost, DATA will automatically set the appropriate optimal path criterion.



▷ Click the Set button next to the Numeric Formatting text item. This numeric format will be used for quantities that have been combined using the relevant multi-attribute formula. (Cost-effectiveness users should see Chapter 21, Cost-Effectiveness, for specific details.) See Chapter 10 for information on modifying the numeric formatting.

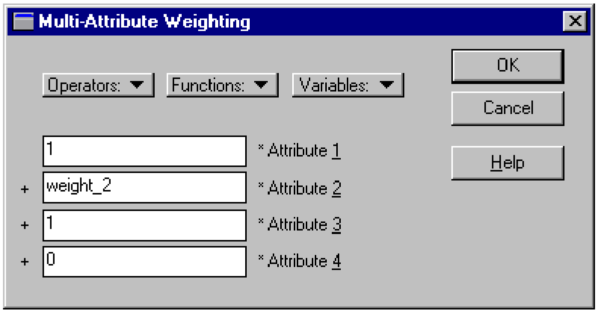
Even after creating a multi-attribute tree in this fashion, it is still possible to view calculated results based on only one of the attributes (cost, for example). To accomplish this, return to the Calculation Method page of the Preferences dialog, and change the calculation method to Simple. You can then select which payoff number should be used for single-attribute calculations.

Changing the preferences in this way will not affect the content of your tree. In other words, no formulas are lost by switching to the Simple calculation method. Only calculations will be affected, and these only temporarily; the changes to the calculation preferences can be reversed at any time.

The currently selected calculation method is displayed in the status bar. For example, if you have selected Simple and chosen payoff 3 as the active payoff, “Payoff 3” will appear in the status bar. If you have selected Cost-Effectiveness and chosen payoffs 1 and 2 for cost and effectiveness, respectively, the status item will read “C/E, 1/2.” If you have selected Benefit-Cost and chosen payoffs 4 and 2 for benefit and cost, the status item will read “B-C, 4-2.” If you have selected (Generalized) Multi-Attribute, the status item will read simply “MultiAttr.”

Entering more than one payoff formula

The process for entering a payoff for a terminal node under multi-attribute calculation methods is identical to that used for the Simple (single-attribute) calculation method. With multi-attribute models, however, you must enter at least two payoffs at each endnode.



1 * Attribute 1
+ weight_2 * Attribute 2
+ 1 * Attribute 3
+ 0 * Attribute 4

The Enter Payoff window is shown when one or more terminal nodes are selected and you choose Values > Change Payoff (or change the node type to terminal). If a multi-attribute calculation method is being used, the payoff titles in the window will indicate which payoffs are to be used. For cost-effectiveness, the two selected payoffs will be labeled “Cost” and “Effect.” For benefit-cost, they will be labeled “Benefit” and “Cost.” For generalized multi-attribute models, they will be labeled “Attr 1” through “Attr 4.”

The process for defining the variables that are components of the payoff formulas is the same as under single-attribute models. It is more complex because, customarily, each attribute will have a different payoff formula with its own set of variables (such as CostA + CostB or IncomeA + IncomeB). You will have to specify definitions for each set of variables at appropriate locations in the tree. For example, components of the cost formula and components of the effectiveness formula may be defined at the same or at different nodes, depending on the structure of the tree.

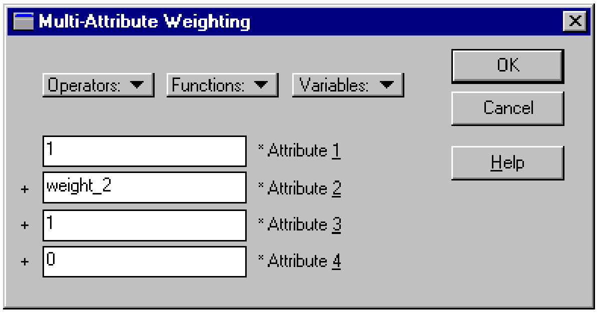
How multi-attribute models are calculated

Cost-effectiveness models: This calculation method separately calculates the expected values of the cost numerator and effectiveness denominator for each node, allowing incremental cost-effectiveness to be calculated and conditions of dominance evaluated. See Chapter 21 for a complete discussion.

Benefit-cost models: This calculation will subtract the cost of a scenario from its benefit. In contrast to cost-effectiveness analysis, where different units of measurement can be employed (e.g., measuring costs in dollars and effectiveness in quality-adjusted life years), both attributes in a benefit-cost analysis must be measured in the same units.

Generalized multi-attribute models: It is possible to enter a linear combination of up to four attributes to serve as the basis for calculating the tree. This is done via either the Set Weightings button in the Calculation Method Preferences dialog box or the Multi-Attribute Weights command under the Values menu.

For example, if you assigned a weight of 0.6 to attribute 1 and a weight of 0.4 to attribute 2, each node would be evaluated based on the expression *0.6 * attribute 1 + 0.4 * attribute 2*. It is possible to use variables in the weighting functions. This is particularly useful when there is uncertainty concerning how much one factor should be weighted versus another. Thus, if you use a weight for attribute 1 such as weight_1, you will be able to perform a sensitivity analysis on weight_1. For models with more than four attributes, you have the option of using a more complex expression in each attribute (e.g., Factor4 * Weight4 + Factor5 * Weight5).


Multi-Attribute Weighting

1 * Attribute 1
+ weight_2 * Attribute 2
+ 1 * Attribute 3
+ 0 * Attribute 4

Chapter 21 Cost-Effectiveness Analysis

Cost-effectiveness (C/E) analysis is a method of evaluating decisions based on two independent criteria using different outcome scales. It is of particular interest in situations where scarce resources require balancing the desire for high effectiveness and the need to contain costs.

This chapter builds on topics covered in Chapters 19 and 20.

How DATA calculates cost-effectiveness

The C/E calculation method determines expected values and selects optimal paths in a manner peculiar to cost-effectiveness calculations. At each chance node, rather than using the average C/E ratios of the outcomes, DATA separately calculates expected (or average) cost and effectiveness. For the purposes of reporting a single value for each node (e.g., in the rolled-back tree), DATA will calculate an average C/E ratio. The separate cost and effectiveness values of a node, though, not its ratio, are used during calculations by the node to the left. This process is repeated for every scenario, starting at the terminal node and working to the left.

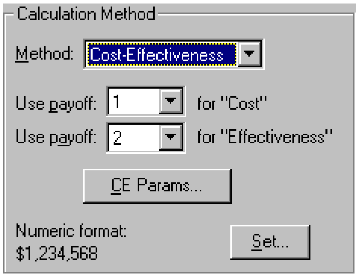
To evaluate decision nodes, DATA examines the expected and incremental cost and effectiveness of each option, as well as its expected and incremental C/E ratios. Based on this information and a set of model-specific C/E parameters, DATA examines conditions of *dominance* and selects an optimal alternative. The value assigned to a decision node is the value of its preferred alternative. (These crucial topics – the calculation of incremental values, DATA's C/E parameters, and dominance – are all discussed in detail in this chapter.)

Notwithstanding that effectiveness constitutes the denominator in C/E ratios, it *is* permissible to have zero effectiveness payoffs. Since cost and effectiveness are calculated separately at each node, DATA will simply report an *undefined* ratio at each node where effectiveness is zero. This zero value is then used in calculating expected effectiveness at chance nodes to the left.

Preparing a tree for cost-effectiveness calculations

In setting up a tree for cost-effectiveness analysis, you must specify: (i) that the tree will use the C/E calculation method, rather than the Simple (single-attribute) calculation method; and (ii) the numeric formatting for each attribute and the C/E ratio.

► **To set up a tree for cost-effectiveness calculations:**



Calculation Method

Method: Cost-Effectiveness

Use payoff: 1 for "Cost"

Use payoff: 2 for "Effectiveness"

Numeric format: $1,234,568

▷ Choose Edit > Preferences. By default, the Calculation Method preference page will be active.

▷ From the Calculation Method pop-up menu, choose Cost-Effectiveness. This tells DATA that you will be using this specific form of multi-attribute calculations.

▷ Select payoff numbers to represent cost and effectiveness values, respectively, in your tree.

There is no constraint on which payoff to assign to cost and which to effectiveness, but you must be consistent. If, for example, you choose payoff 1 to represent cost values, then you must ensure that, at every terminal node, the first payoff field contains costs.

> **TIP:** The payoff selections in the calculation method preferences represent which value sets are currently used for calculations. It is possible, for example, to enter at each terminal node one set of values representing costs and up to three sets of values representing utility, each on a different scale. To change the effectiveness scale being used to calculate the C/E ratio, simply select a different effectiveness payoff.

Cost-effectiveness numeric formatting

► **To specify numeric formatting for C/E calculations:**

▷ Press the Set button next to the Numeric Formatting text item. You must assign three separate numeric formats: one for cost alone (perhaps in units of dollars), one for effectiveness alone (perhaps in units of quality-adjusted life-years), and one for the ratio (perhaps in dollars per QALY).

See Chapter 9 for information on modifying a numeric format. The CE Template tree provides an example of how to set up the numeric formats for a cost-effectiveness model.

Cost-effectiveness optimal path parameters

Evaluating an optimal path under the cost-effectiveness calculation method is more complex than simply selecting the high or low path. By specifying a set of cost-effectiveness parameters, a variety of objectives can be included in the evaluation. While it is possible to set these parameters so that DATA simply minimizes the average C/E ratio at each decision node, it is also possible to have DATA do any or all of the following:

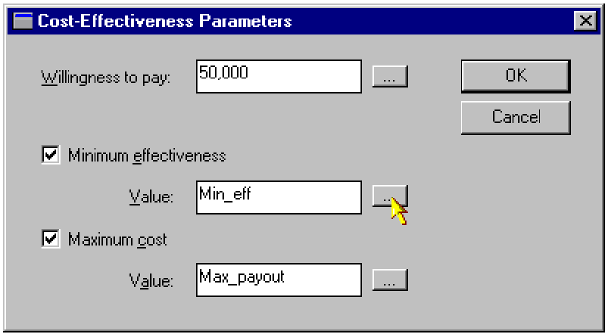
- select the most effective option within an incremental cost-effectiveness threshold (specified under Willingness to pay);
- eliminate options falling below a minimum expected effectiveness; and/or
- eliminate options above a certain expected cost.

The next section discusses exactly how these parameters will be applied in your tree.

► To set the optimal path parameters for C/E calculations:

▷ Press the CE Params… button. In the Cost-Effectiveness Parameters dialog, select the desired options and enter appropriate parameter values.

▷ Press OK to accept the entered parameters.



Cost-Effectiveness Parameters

Willingness to pay: 50,000

☑ Minimum effectiveness

Value: Min_eff

☑ Maximum cost

Value: Max_payout

Clicking on the ellipsis buttons next to the parameter text boxes will open an expression editor dialog, where complex formulas using variables, distributions, functions, and table references can be entered. Note that expressions entered for the cost-effectiveness parameters will be evaluated once, at the root node of the tree, regardless of the location of the decision node being evaluated.

Existing default C/E parameters should *not* be automatically accepted for any model. You should design a set of criteria that is appropriate to your tree; see the following section on decision making.

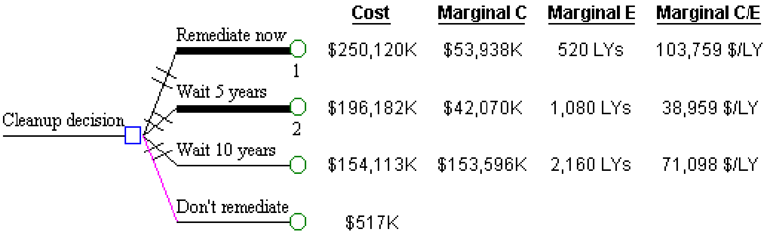
Decision making using cost-effectiveness

DATA's rollback display will recommend an optimal strategy based on the parameters you entered in the C/E Parameters dialog (primarily your willingness-to-pay value or expression). First, strategies will be ordered by increasing cost. Second, options excluded by your minimum effectiveness and/or maximum cost constraints are eliminated. Third, dominated options are excluded. Fourth, options which fail your willingness-to-pay criterion are excluded. Finally, if two or more options remain, they are scanned in order of cost (and, by implication, effectiveness), from lowest to highest.

The selected optimal alternative is the most costly (and most effective) option *with a marginal C/E less than your willingness-to-pay* value. If you specify a willingness to pay of zero and turn off the minimum effectiveness and maximum cost options, the alternative with the lowest expected cost-effectiveness ratio will be selected. It is also possible to have no optimal alternative, if your minimum effectiveness and maximum cost expressions eliminate all options from consideration.

Displaying marginal values in terminal node columns

The current values of your cost-effectiveness parameters are *not* automatically displayed on the face of the tree, and neither are the marginal values used to select an optimal path. It is possible to make this information explicit in your rolled-back tree, though.



	Cost	Marginal C	Marginal E	Marginal C/E
Remediate now	$250,120K	$53,938K	520 LYs	103,759 $/LY
Wait 5 years	$196,182K	$42,070K	1,080 LYs	38,959 $/LY
Wait 10 years	$154,113K	$153,596K	2,160 LYs	71,098 $/LY
Don't remediate	$517K			

To get a visual display of marginal values in the rolled-back tree, you can create terminal node columns. This feature, covered in Chapter 10, makes it possible to display custom columns of calculated values, including incremental (marginal) values, to the right of visual end nodes during roll back. It is also possible to define the C/E parameters (e.g., willingness to pay) using variables, and then display the value of these variables in a custom column.

Other ways of viewing incremental values, discussed below, include cost-effectiveness analysis and one- and two-way sensitivity analysis.

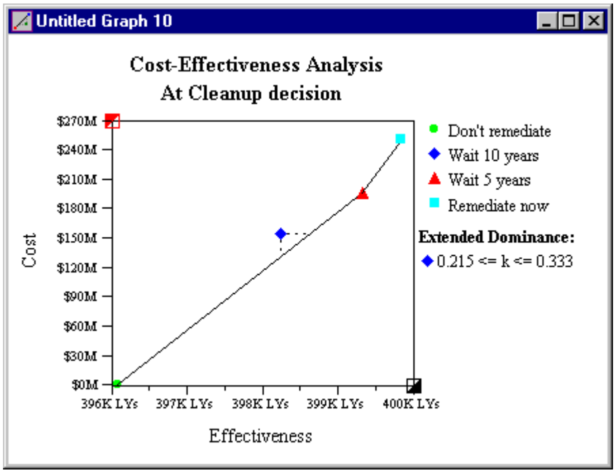
Cost-effectiveness sensitivity analysis

DATA includes two powerful forms of sensitivity analysis adapted for use in cost-effectiveness decision making. Both are based on the calculations used in cost-effectiveness analysis, discussed above and in the following section.

Performing a one-way sensitivity analysis at a decision node in a cost-effectiveness tree provides eight graph options; detailed text reports are also available. In DATA's two-way sensitivity analysis, it is possible to identify regions of equal marginal cost-effectiveness using isocontours. Sensitivity analysis on cost-effectiveness models is discussed in Chapter 22.

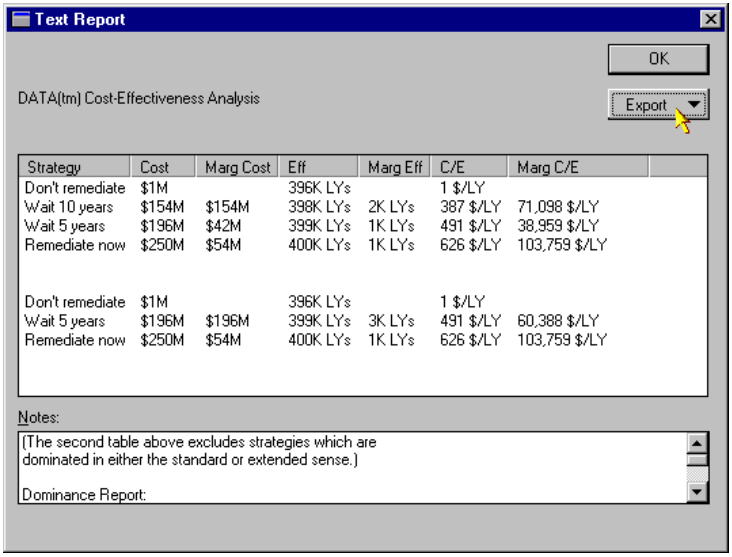
Cost-effectiveness graph

With a cost-effectiveness tree active, select a decision node and choose Cost-Effectiveness from the Analysis menu. You are presented with two graph options: to display cost on the X-axis (effectiveness on the Y-axis), or vice versa. This will generate a strategy dominance graph, in which each option is represented by a point measured on axes representing cost and effectiveness.



In the graph, options which are not eliminated from consideration by absolute dominance or are not subject to considerations of extended dominance (see below) will be connected to form a set of potentially optimal alternatives. Any instances of extended dominance will be clearly identified, together with the range of possible coefficients of inequity.

The text report of the graph contains all of the relevant numeric data for the set of alternatives. DATA will list for each option the expected cost, marginal cost, expected effectiveness, marginal effectiveness, expected C/E, and marginal C/E.



DATA(tm) Cost-Effectiveness Analysis

Strategy	Cost	Marg Cost	Eff	Marg Eff	C/E	Marg C/E
Don't remediate	$1M		396K LYs		1 $/LY	
Wait 10 years	$154M	$154M	398K LYs	2K LYs	387 $/LY	71,098 $/LY
Wait 5 years	$196M	$42M	399K LYs	1K LYs	491 $/LY	38,959 $/LY
Remediate now	$250M	$54M	400K LYs	1K LYs	626 $/LY	103,759 $/LY
Don't remediate	$1M		396K LYs		1 $/LY	
Wait 5 years	$196M	$196M	399K LYs	3K LYs	491 $/LY	60,388 $/LY
Remediate now	$250M	$54M	400K LYs	1K LYs	626 $/LY	103,759 $/LY

Notes:

(The second table above excludes strategies which are dominated in either the standard or extended sense.)

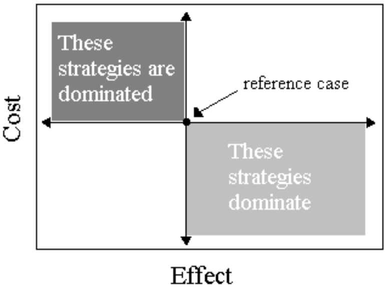
Dominance Report:

In addition, the report will include a textual description of each instance of either absolute or extended dominance. If any alternatives were dominated (in either sense), a second listing of marginal values will be shown, with the dominated options excluded.

Dominance

In the context of a cost-effectiveness analysis, one alternative is said to be *dominated* by another if the first both costs more and is less effective. In some problems, when this is the case, the dominated alternative may be removed from consideration.

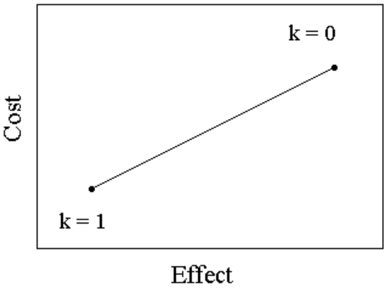
The use of relative position to infer dominance is illustrated in the graph at right. Effectiveness increases from left to right; cost increases from bottom to top. The crossing point of the axes represents one alternative. Its comparators can then be placed on the graph, more costly alternatives, above, and more effective alternatives to the right.



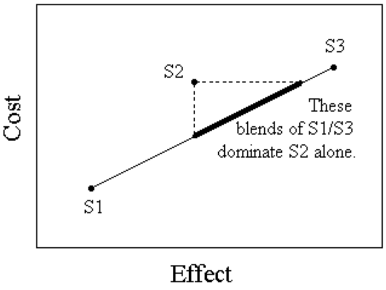
Thus, an alternative is dominated if it lies both above and to the left of another alternative. (Note that the axes are sometimes reversed, giving different rules for inferring dominance visually.)

Extended dominance

When making certain population-wide policy decisions, two decision alternatives may sometimes be "blended" to create any number of intermediate alternatives. For example, one alternative may be applied to 20% of the population, and another to 80%. All possible blends between two alternatives are represented by the line connecting them, as shown at left.



The constant k represents the proportion of the population receiving the less effective treatment; it is called the coefficient of inequity. The net cost of any given blend is k * (cost of less expensive treatment) + (1-k) * (cost of more expensive treatment). The net effectiveness is calculated similarly.



Graphically, an alternative is said to be *dominated in the extended sense* when it lies above (or falls below, depending on how the axes are displayed) the line which connects two other alternatives. The heavy bar along the blend line shown at left represents all possible blends which dominate S2. That is, a blend of S1 and S3 which falls on the heavy bar will have better overall effectiveness for less overall cost than S2 alone. This is the basis of extended dominance.

Mathematically, alternative S2 can be eliminated by extended dominance when the slope between S1 and S2 is greater than the slope between S2 and S3. This means that the marginal cost-effectiveness of S2 relative to S1 is greater than the marginal cost-effectiveness of S3 relative to S2.

If, for example, alternative S2 were to be rejected on the basis of extended dominance, this would mean that a certain proportion (k) of the population will receive an alternative which is less effective than S2. Thus, in contrast to absolute dominance, an alternative should not be automatically eliminated from consideration simply because it is dominated in the extended sense.

In those situations where financial constraints or limited availability may preclude treating the entire population with S3 (the most effective option), it is important to recognize that S1 and S2 are not the only affordable options; rather, any two options may be blended to arrive at a new balance of cost and effectiveness. However, if a blend of options S1 and S3 is chosen over S2 alone, a divided system results, in which some receive more effective treatment and some receive less effective treatment. How is it to be decided who receives S1, and on what basis?

More information on this subject can be found in “Cost-Effectiveness Analysis, Extended Dominance, and Ethics,” Scott B. Cantor, *Medical Decision Making* 14:259 (1994).

Monte Carlo simulation

Performing a Monte Carlo simulation at a decision node in a cost-effectiveness tree does not automatically result in the reporting of marginal values. If you specify that DATA should reevaluate the optimal path during the simulation, though, the 2nd-order simulation will take into consideration the C/E parameters discussed at the beginning of this chapter (e.g., willingness to pay) when evaluating decision nodes.

To see the incremental values used in this process, it is possible to export the text report of a simulation to a spreadsheet. There, additional columns can be added and used to calculate incremental C/E ratios.

Chapter 29 covers Monte Carlo simulation.

CHAPTER 22 ADVANCED SENSITIVITY ANALYSIS

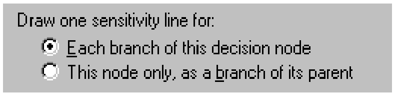
The advanced analyses covered in this chapter are grounded in one-way sensitivity analysis. This subject is covered in Chapter 5; the reader is assumed to be proficient with it. Other important topics related to the use of variables in sensitivity analysis are covered in Chapter 8.

Viewing the changing value of a single scenario

Normally, when performing a one-way sensitivity analysis, a decision node is selected and DATA displays one line for each of the alternative scenarios rooted at the selected node. It is possible to focus a one-way sensitivity analysis on a single scenario, rather than on all of the scenarios emanating from a decision node.

If the node you select prior to performing the sensitivity analysis is not a decision node, DATA will assume that the results should be presented as a single line. This will represent the changing expected value of the scenario rooted at the selected node. (Note that this option is not available for cost-effectiveness sensitivity analyses, which must be performed at a decision node.)

If, however, you select a decision node which is an immediate descendant of another decision node, DATA will give you the option of drawing one line for the selected node (as a branch of its parent), or multiple lines for the branches emanating from the selected decision node.



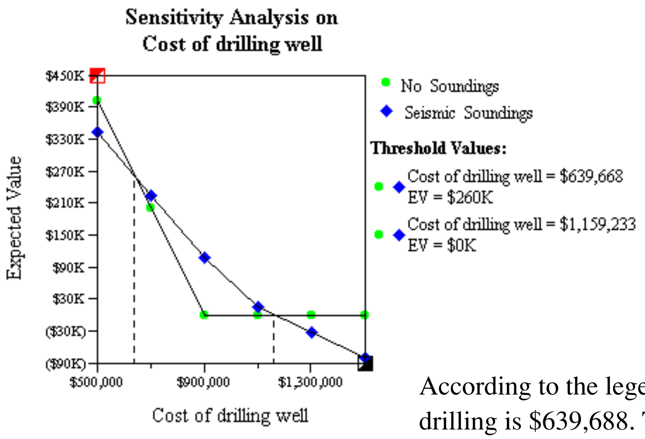
Draw one sensitivity line for:

● Each branch of this decision node

○ This node only, as a branch of its parent

Interpreting multiple thresholds

In Chapter 6, it was noted that the results of the threshold analysis performed during a one-way sensitivity analysis require interpretation. This is particularly true in models representing sequential decisions, like the oil drilling problem introduced in Chapter 8.



Open the file Oil Drilling #2 and perform a sensitivity analysis at the root node on the variable Drill from $500,000 to $1,500,000. In the resulting graph, the lines representing the two subtrees intersect at two different points. At each of these points, both subtrees have the same expected value; the decision maker basing his choice on expected value should be indifferent between the two alternatives: undertaking seismic soundings or not undertaking them.

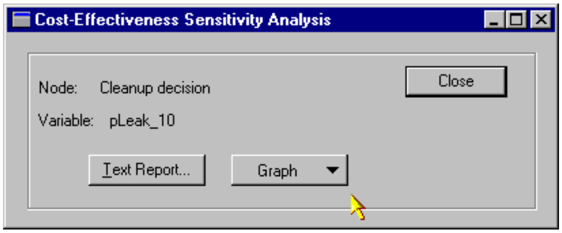
According to the legend, the first threshold occurs when the cost of drilling is $639,688. To the left of this point, the line representing the No Soundings option is always above the Seismic Soundings line; if drilling costs are lower than $639,668, you should choose the No Soundings option. Between $639,668 and the second threshold, at $1,159,233, you should perform the tests. Once drilling costs exceed $1,159,233, you should not spend money on seismic soundings.

This means that when drilling costs fall below about $639,000 or exceed about $1,160,000, the actual costs of seismic soundings outweigh their potential benefits. Increasing the range of the above sensitivity analysis, performing analyses at the various drilling decision nodes, as well as rolling back the tree for key values of Drill, may help you understand why this is true. For example, at very low values of Drill, all test results will suggest Drill for Oil; clearly it doesn't pay to spend *any* money on such advice. As Drill exceeds about $400,000, a No Structure finding now suggests Don't Drill. The benefit of the test increases with Drill until the testing decision's first threshold is reached. When Drill exceeds $900,000, the No Soundings subtree now suggests not drilling (EV=0), and the benefit of testing begins to fall. Eventually the test has no value, and the second threshold is reached

One-way sensitivity analysis in cost-effectiveness models

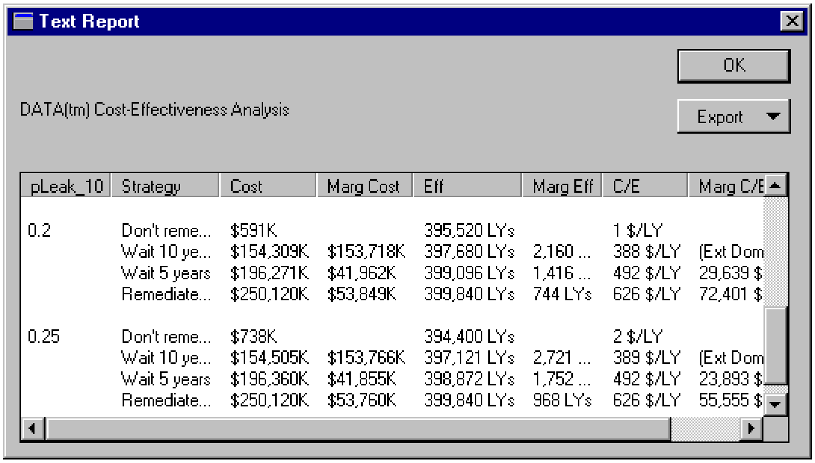
To perform a one-way sensitivity analysis on a cost-effectiveness tree, simply select the appropriate decision node and choose Analysis > Sensitivity Analysis... > One-Way.... See Chapters 19–21 for details on setting up a cost-effectiveness model.

DATA displays an intermediate output window after the analysis is complete. This window does not show analysis results directly. However, from this window, you can elect to show the complete text report, or to view any of several graphs.



Node: Cleanup decision
Variable: pLeak_10

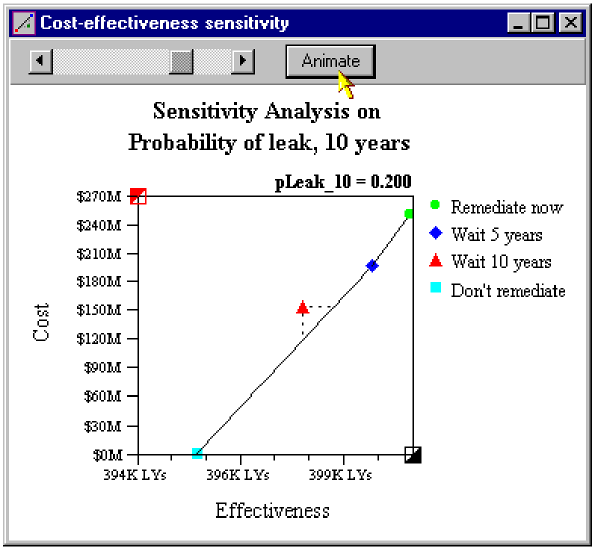
Here is the text report from a sensitivity analysis on a probability variable. As you can see, at every interval, all of the average and marginal values are shown for each alternative:



DATA(tm) Cost-Effectiveness Analysis

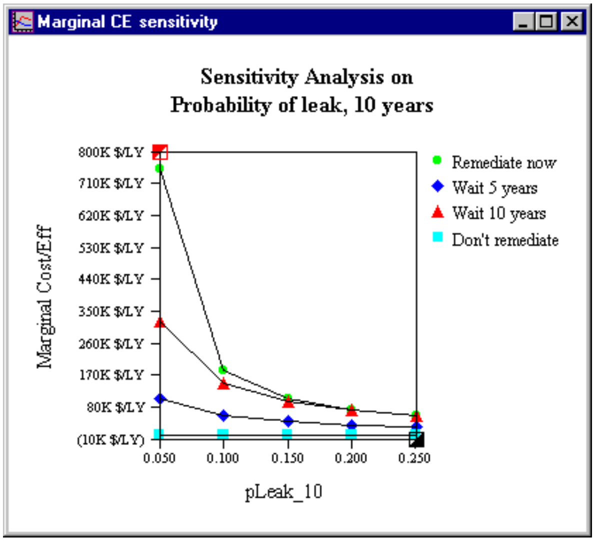
pLeak_10	Strategy	Cost	Marg Cost	Eff	Marg Eff	C/E	Marg C/E
0.2	Don't reme...	$591K		395,520 LYs		1 $/LY	
	Wait 10 ye...	$154,309K	$153,718K	397,680 LYs	2,160 ...	388 $/LY	(Ext Dom
	Wait 5 years	$196,271K	$41,962K	399,096 LYs	1,416 ...	492 $/LY	29,639 $
	Remediate...	$250,120K	$53,849K	399,840 LYs	744 LYs	626 $/LY	72,401 $
0.25	Don't reme...	$738K		394,400 LYs		2 $/LY	
	Wait 10 ye...	$154,505K	$153,766K	397,121 LYs	2,721 ...	389 $/LY	(Ext Dom
	Wait 5 years	$196,360K	$41,855K	398,872 LYs	1,752 ...	492 $/LY	23,893 $
	Remediate...	$250,120K	$53,760K	399,840 LYs	968 LYs	626 $/LY	55,555 $

There are many ways to view the sensitivity analysis output graphically, via the Graph pop-up menu in the intermediate output window:



- **Cost vs. Effectiveness** – The results of the sensitivity analysis are presented as an animated cost-effectiveness analysis, with cost on the x-axis; see Chapter 21 on interpreting a C/E graph. This dynamic graph window displays a series of frames, each showing the comparative cost-effectiveness of competing alternatives. Pressing the Animate button causes DATA to step through each interval of the analysis; the current value of the variable being analyzed is displayed above the graph's upper right corner.
- **Effectiveness vs. Cost** – Same as above, with the axes inverted; effectiveness is on the x axis and cost on the y-axis.

There are six other graph types available from the intermediate analysis window. Each graph resembles a standard, one-way sensitivity analysis line graph, showing how the chosen value (e.g., marginal cost-effectiveness) varies with changes in the input variable:



- variable vs. marginal cost-effectiveness;
- variable vs. marginal cost;
- variable vs. marginal effectiveness;
- variable vs. average cost-effectiveness;
- variable vs. average cost; and
- variable vs. average effectiveness.

In each of these six line graphs, the sensitive variable is shown on the x-axis. The values on the y-axis are displayed in terms of your selected basis for analysis, such as marginal cost-effectiveness. Unlike a standard sensitivity analysis, which displays only expected values, cost-effectiveness sensitivity analysis line graphs do *not* show any threshold information. To see threshold information for average cost and average effectiveness graphs, change the tree's calculation method to Simple and select the appropriate payoff set. Performing a sensitivity analysis will then provide the standard display of expected value thresholds.

There are two situations in which there is no meaningful value in the sensitivity analysis output of marginals. The first occurs when an alternative is the least expensive (in looking at marginal cost or marginal cost-effectiveness) or least effective (in looking at marginal effectiveness). The second occurs when, in the cost-effectiveness graph, an alternative is dominated by another – it has a positive marginal cost and a negative marginal effectiveness (i.e., it is more costly and less effective than the dominating option). In this situation, the marginal cost-effectiveness ratio is meaningless.

When either of these situations occurs, DATA will display a marginal value of zero in the sensitivity analysis graph. This value is not meant to be mathematically accurate; it indicates that a meaningful marginal value is not available at that interval, for that alternative.

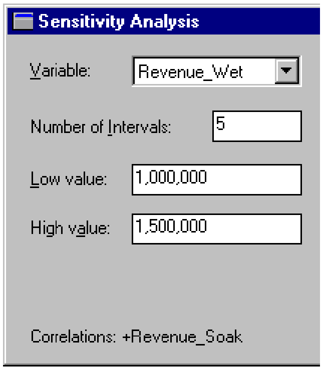
Sensitivity analysis options

There are a variety of options available when performing any of the sensitivity analyses discussed in this chapter, not just during one-way sensitivity analysis.

Correlated variables

Correlations among any number of variables can be specified in your tree for use during sensitivity analysis. Correlations are set up from the Properties dialog box for either of the correlated variables, by clicking on the Correlations... button. See Chapter 9 for detailed instructions on defining the sensitivity analysis properties for a variable.

If, for example, you set up a correlation for a pair of variables, the correlation is identified in the properties of *both* variables. Thus, when you choose to perform a sensitivity analysis on any member of a correlated group of variables, DATA will remind you that one of the parameters to be varied has one or more correlates that can also be varied during the analysis. Once the analysis parameters have been entered, you will have the option of simultaneously varying any or all correlated variables over their own value ranges.



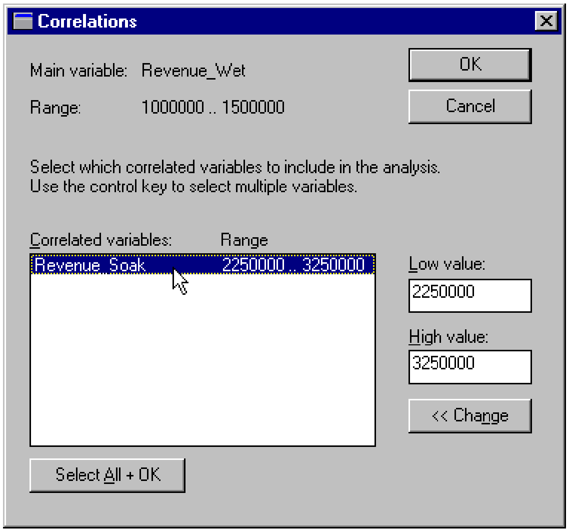
Variable: Revenue_Wet

Number of Intervals: 5

Low value: 1,000,000

High value: 1,500,000

Correlations: +Revenue_Soak



Main variable: Revenue_Wet

Range: 1000000 .. 1500000

Select which correlated variables to include in the analysis.
Use the control key to select multiple variables.

Correlated variables:	Range
Revenue_Soak	2250000 .. 3250000

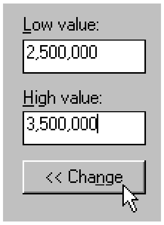
Low value: 2250000

High value: 3250000

After choosing to perform a sensitivity analysis on a variable with correlations, the Correlations dialog will appear. From this dialog, it is possible to specify which correlations should be active during the analysis.

To include all displayed correlations, simply click the Select All + OK button. To include only particular correlated variables, highlight each variable name for inclusion and then click OK. Finally, to exclude all correlates, click the OK button immediately on entering the Correlations dialog, leaving all correlate variable names unselected.

It is also possible to change the range of values applied to a correlated variable. Clicking on a correlate's name in the list activates two text boxes allowing you to change the default high and low values. To update the range using the changes you make, click on the Change button below the two text entry boxes.



Low value: 2,500,000

High value: 3,500,000

During the analysis, the same number of intervals specified for the original variable will be used in dividing the range applied to each correlated variable.

You should always pay close attention to the ranges for correlated variables. Because DATA does not parameterize the correlation, you must ensure that if you use a narrow range for one variable (its 75% confidence intervals, for example), you use an appropriately narrow range for the correlated variables, as well.

In all resulting sensitivity analysis graphs (except for three-way sensitivity analysis), a text item will display the name and the range of each correlated variable which was varied during the analysis. On the graph, this text will be placed adjacent to the name of the original variable at the appropriate axis.

It is not possible to change the type of correlation (positive or negative) from within the sensitivity analysis dialog; these changes must be made in the variable's Properties dialog. Detailed instructions on modifying sensitivity analysis properties of variables can be found in Chapter 9.

Correlations are available during all sensitivity analyses. If you choose to perform a two- or three-way sensitivity analysis on variables that are already correlated, DATA will provide a warning prior to prompting for the correlations to be included in the analysis. DATA will not prevent you from performing an analysis which both varies and correlates the variables, but the results will probably be invalid

Variables with non-numeric definitions

A sensitivity analysis can be performed on any variable in your tree, whether it has a numeric value definition (e.g., X=1 or X=Exp(2)) or a variable expression (e.g., X=Rate*Principal). When performing a sensitivity analysis on a variable defined using other variables, you have multiple options. You can perform a sensitivity analysis on the component variables (e.g., Rate and Principal) using variable correlations or a multi-way sensitivity analysis. Alternatively, you can perform a one-way sensitivity analysis on the original variable (e.g., X) based on an estimated numeric value range.

If you treat X as the independent variable, however, the formula will be ignored during the course of the analysis. Definitions of Rate and Principal in your tree will not be used during this analysis.

We recommend that you focus your sensitivity analysis on the finest-grain parameters. In the example above, X is no longer finest-grain once you define it in terms of its two component variables. In general, you should design your models so that the sensitive variables have only one numeric definition.

Variables with more than one definition

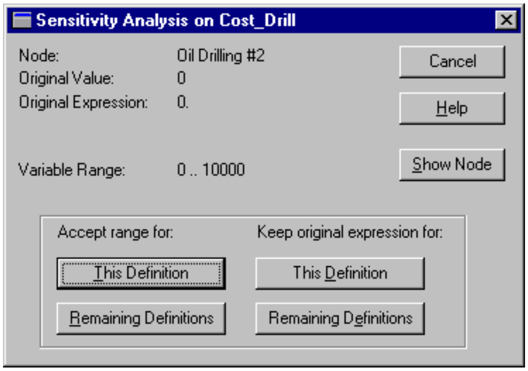
In some instances, once you have constructed your model, you may find more than one *numeric* definition assigned to a single variable. For example, you may have defined Costs at the root node with a default value of 100, and then redefined Costs at an internal node with a value of 400. This situation, having multiple value definitions of the same variable, can cause complications when performing sensitivity analysis. It is generally preferable, instead, to assign only a single numeric definition to a given variable.

Only a single range can be applied to a variable selected for sensitivity analysis. This is not necessarily a problem if there are only two numeric definitions of the variable, as long as one of the definitions will always be zero (or another constant). However, this is the only exception to the rule against using multiple numeric definitions of the same variable. *In those situations where a variable must take on different values at different locations in the model, that variable should be defined in terms of other variables – not numerically.*

Thus, in the above example, Costs can be defined at the root node as being equal to LowCosts and at the internal node as being equal to HighCosts; at the root node define LowCosts=100 and HighCosts=400.

Following this methodology, variables with a numeric definition are defined only once. Since the variable Costs does not have any numeric definitions, it may be redefined as many times as needed, each time in terms of another, unique variable. Under this arrangement, sensitivity analysis will always be performed on an appropriate variable which has a single numeric definition. In the above example, the sensitivity analysis would be performed either on LowCosts or HighCosts (or both), and never on Costs itself.

If this rule is not followed, complications will arise should you attempt to perform a sensitivity analysis on a variable with multiple numeric definitions. In particular, you must specify which definitions should be varied across the specified range, and which should be held fixed. In the example above (with two numeric definitions of Costs), you might indicate that Costs=100 should not be varied, while Costs=400 should take on values in the specified range, perhaps between 300 and 500. Alternatively, you could provide a range to substitute for the definition Costs=100 and not vary Costs=400 at all. What you cannot do is provide a different range for each definition of Costs. If this is required, you must define multiple variables, using the methodology illustrated earlier.


Sensitivity Analysis on Cost_Drill

Node: Oil Drilling #2
Original Value: 0
Original Expression: 0.

Variable Range: 0 .. 10000

DATA uses the special dialog box displayed in the margin to prompt for information on which definitions to vary during the analysis.

For each definition of the variable, DATA will identify the node at which it is defined, and its value (either assigned directly or calculated). If the particular definition of the variable should take on values from the specified range, under Accept range for, click the button reading This Definition. Otherwise, the original definition should be retained; under Keep original expression for, click the button labeled This Definition.

When the original expression is retained, DATA will calculate the value of the variable by using whatever formula (or numeric value) is assigned. If you choose to accept the range for any given definition, DATA will replace the actual definition with a numeric value taken from the range.

TIP: If all of the remaining definitions of the chosen variable are to be varied across the range specified in the initial dialog box, you can simply click on the button labeled Accept range for Remaining Definitions. This action has the same effect as selecting Accept range for This Definition at each of the remaining definitions of the chosen variable. A similar and corresponding action is produced by selecting Keep original expression for Remaining Definitions.

Probability coherence

Most forms of sensitivity analysis offer an option labeled Check coherence. When this option is selected, DATA will ensure that, at each interval, (i) all probabilities sum to 1.0 and (ii) no probabilities are negative. The analysis will be halted if at any time either rule is violated.

If the subject variable is used to define a probability, you are encouraged to leave this option selected. This will ensure the validity of your model over the range of the analysis. This is particularly important in the initial stages of testing your model's validity. The downside is that calculation time is increased. If no probability variables are being varied in the sensitivity analysis, you may safely opt to turn off coherence checking.

Normalizing probabilities

Often, it is possible to structure the chance events represented in your tree so that each chance node has only two branches. If chance nodes are limited to two outcomes, one branch can be given a probability expression while the other is automatically assigned the remainder (e.g., using the # sign). This approach greatly simplifies the process of performing sensitivity analysis on probability variables; so long as the specified probability expression evaluates to a number from 0 to 1, the probabilities of the chance event will remain coherent.

Another option is to *normalize* the appropriate probability expressions. This is a useful technique if you cannot resolve each chance node to two outcomes. For example, let's say you have three outcomes, A, B, and C. Rather than assigning expressions to two probabilities, and using the # sign to calculate the remainder for the last, you could assign three expressions that would always sum to 1: a/(a+b+c), b/(a+b+c), and c/(a+b+c). No matter what value is assigned to the three parameters a, b, and c (so long as none are negative), probability coherence will be maintained. This technique could be extended to any number of outcomes.

Two-way sensitivity analysis

Two-way sensitivity analysis enables you to test the sensitivity of a proposed decision to simultaneous changes in the values of two independent variables. Results are presented in a region graph, in which regions of different colors are used to indicate the optimal choice at any given values of the two variables.

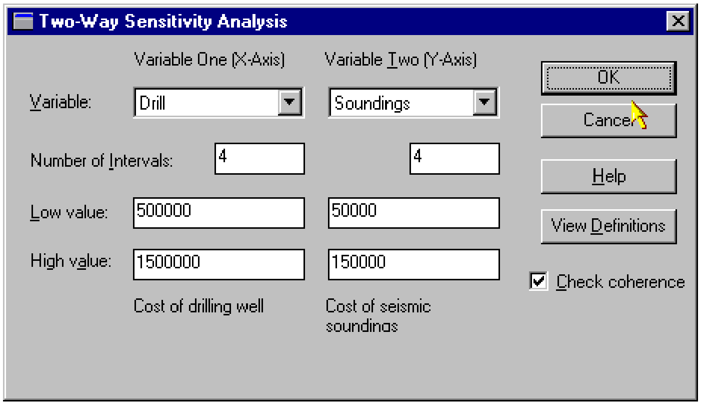
> **TIP:** You should use two-way analysis *only* when the two chosen variables are independent. See above for a description of using correlated variables.

The two-way sensitivity dialog resembles the one for a one-way sensitivity analysis, except that you must specify two variables and a range of values for each. Note that a decision node must be selected in order to perform a two-way sensitivity analysis.

► **To perform a two-way sensitivity analysis:**

▷ Open the sample file Oil Drilling #2.

▷ Select the root node, and choose Analysis > Sensitivity > Two-Way.



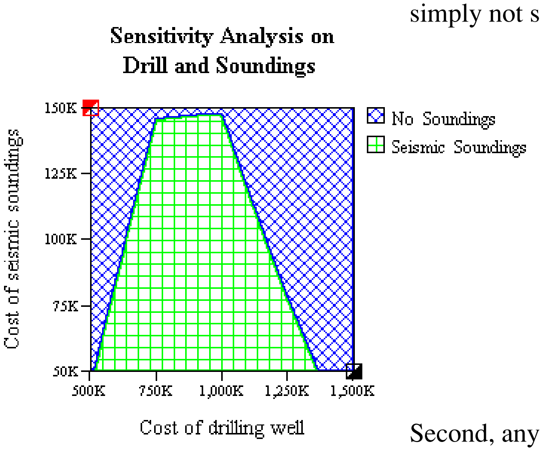
Two-Way Sensitivity Analysis

	Variable One (X-Axis)	Variable Two (Y-Axis)
Variable:	Drill	Soundings
Number of Intervals:	4	4
Low value:	500000	50000
High value:	1500000	150000
	Cost of drilling well	Cost of seismic soundings

☑ Check coherence

- ▷ For the first variable, select Drill, and use four intervals and a range from `500,000` to `1,500,000`.
- ▷ For the second variable, select Soundings, and use four intervals and a range from `50,000` to `150,000`.

DATA will then calculate the tree 25 (5x5) times. As you request more intervals, processing time will increase exponentially. You may also find that two-way analyses often require more intervals per variable to attain a reasonable level of accuracy than do one-way analyses. This is because a two-way analysis graphically represents only the threshold values – the optimal path crossings. In contrast, a one-way analysis shows all expected values at all points on the graph. The one-way analysis may show significant details which are simply not shown in the two-way analysis.



The graphical representation of two-way sensitivity analysis results (and, by extension, of three-way analysis results) has some unavoidable limitations. First, the accuracy of threshold lines may be compromised around the edges of the graph. The unavoidable result of using approximation techniques to identify thresholds is the appearance of distortion when two edges of a region of optimality draw closer together than one-half the width of an analysis interval. Accuracy can be enhanced by running the analysis using more intervals.

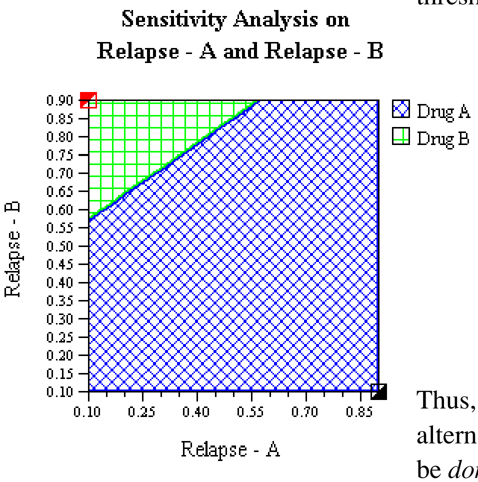
Second, any regions of indifference are not shown. Areas of the graph where indifference exists are, instead, assigned to an option. You should use the text report (accessed via the Graph > Text Report command) to identify any areas of indifference. This report will specify the exact calculated values at each interval.

A legend displaying the name and the range of each correlated variable involved in the two-way sensitivity analysis will be placed adjacent to the appropriate axis.

TIP: The text report of two-way and three-way sensitivity graphs is not readily imported into graphing programs. The numeric data given represent the expected values of all strategies at all values of the two variables. While this information may prove valuable in many cases, threshold values are not included in the text report. Thresholds are evaluated graphically, not analytically. To view more accurate threshold values, you should perform either a one-way sensitivity analysis or a threshold analysis while fixing one of your two variables at a particular value.

Two-way cost-effectiveness sensitivity analysis

In a tree set to calculate a single attribute (e.g., cost or utility), a two-way sensitivity analysis identifies the optimal decision alternative for each combination of values of the two variables. This is represented in a region graph, with each alternative's region of optimality defined by threshold lines.



The interpretation of a graph for a *cost-effectiveness* model is more complex. One piece of information is relatively unambiguous: any given region is assigned to the alternative with the *lowest cost* for those combinations of variable values.

What is not immediately obvious, in any given part of a region assigned to a particular alternative, is whether that alternative has a higher or lower effectiveness – in other words, whether the alternative is dominant (see Chapter 21) or represents a sacrifice of effectiveness for cost savings.

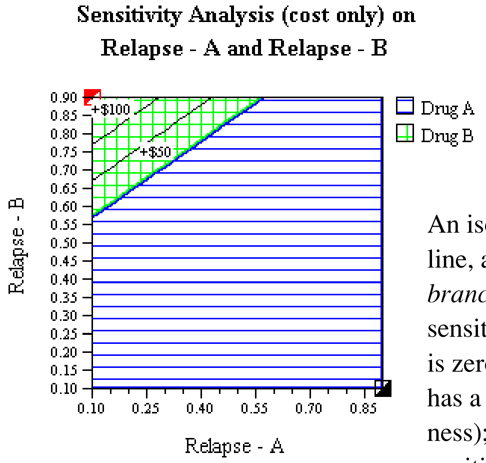
Thus, if a cost-effectiveness region graph is assigned entirely to one alternative (i.e., all one color/pattern), the represented alternative may be *dominant for all* combinations of variables, be less effective as well as *less costly over all* combinations, or a mix of these two conditions (i.e., dominant for some combinations and less effective for the rest).

If both alternatives are assigned a partial region, interpretation is even more complex. Any of the three interpretations might apply to each alternative's *region of optimality*.

In a graph where more than one alternative is represented as optimal, the threshold line separating two regions indicates a transition point, where costs are equal. A different kind of transition, that of an optimal alternative changing from simply being less costly to being *dominant*, is hidden.

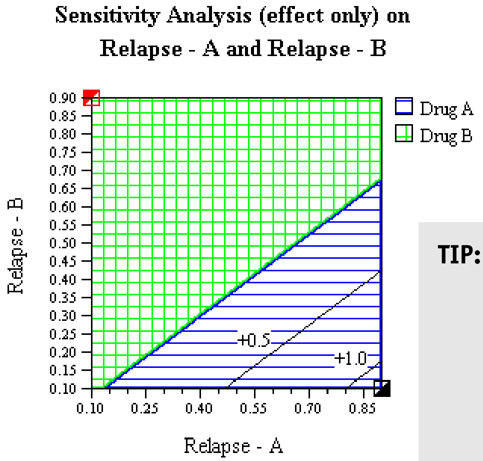
Isocontours

Clearly, interpreting a two-way sensitivity analysis in a cost-effectiveness context can be complex. If the decision being evaluated has only two alternatives, DATA aids in the interpretation of the graph by allowing you to add *isocontours*. Isocontours are available in *any* two-alternative, two-way sensitivity analysis – not just in cost-effectiveness calculations.



The two graphs shown here represent the same sensitivity analysis (see the example file CE Isos Tree) shown on the previous page, but calculated for cost and effectiveness separately. These graphs show how isocontours can improve the region graph's display.

An isocontour represents, for the combinations of variables along the line, a constant marginal value (cost or effectiveness, here) of the *upper branch of the decision node*. Thus, a threshold line in a two-way sensitivity analysis is already an isocontour, where the marginal value is zero. Additional, positive isocontours show where the *upper branch* has a positive marginal value (i.e., cost, effectiveness, or cost/effectiveness); negative isocontour values show where the lower branch has a positive marginal value. See below for an important discussion of interpreting marginal cost-effectiveness isocontours



To create custom isocontours, select Graph > Isocontours while a two-way sensitivity graph (comparing two alternatives) is active. Enter the marginal values you would like shown.

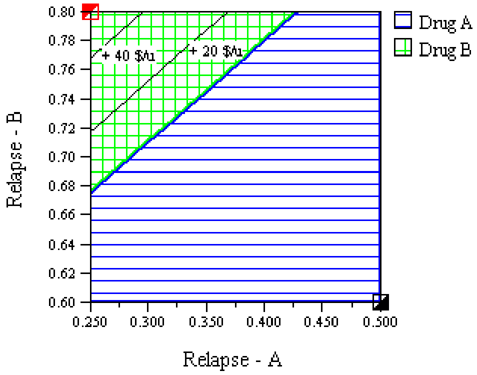
DATA does not automatically create labels for isocontours; use the Graph > New Label… command, described in Chapter 33.

TIP: Adding only a 1000 isocontour will not show lines representing both +1000 and -1000 marginal values; you need to add both 1000 and -1000 values to the list of isocontours in the graph to see both the +/- 1000 isocontours in the graph. Negative value isocontours, in this context, merely indicate that the comparator has changed from the upper option to the lower branch; see the discussion, below.

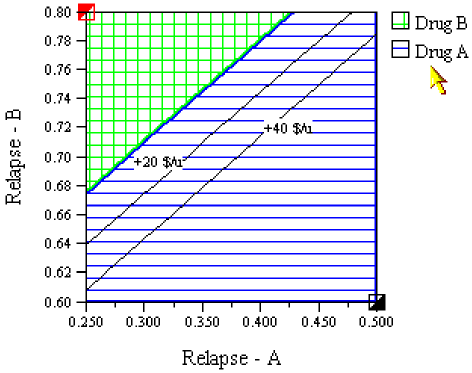
Custom isocontours may be of particular interest when evaluating cost-effectiveness models.

If both alternatives are assigned a partial region in the two-way sensitivity analysis graph, the threshold line dividing the regions represents a line of *zero marginal cost* (equal cost), and thus a marginal cost-effectiveness of zero. By adding isocontours, other marginal cost-

Sensitivity Analysis on
Relapse - A and Relapse - B



Sensitivity Analysis on
Relapse - A and Relapse - B



effectiveness values can be represented, and a particular alternative's region of optimality can be better defined.

A comparison of the graphs displayed in the left margin may help demonstrate the importance of understanding what isocontours mean in a cost-effectiveness context.

It is crucial to remember that DATA automatically bases the calculation of isocontours on the visual order of alternatives at the decision node (i.e., top minus bottom). Although it is not apparent in the top graph at left, a point on one of the positive isocontours actually locates a combination of variable values for which Drug A's marginal cost and marginal effectiveness are both *negative* (resulting in a positive ratio).

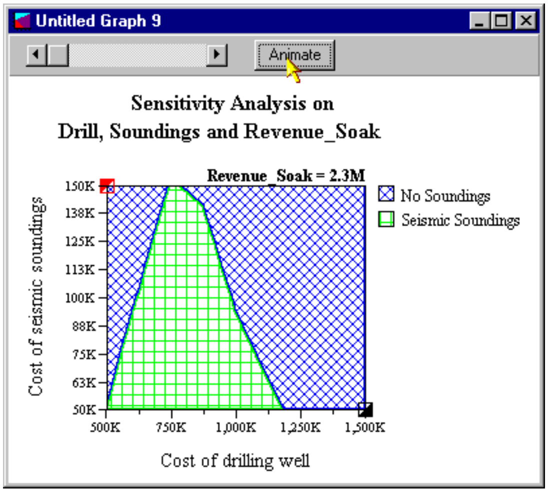
The graphs on the previous page, generated for cost and effectiveness separately, help illustrate the condition of dominance in the area of the graph (cost-effectiveness) assigned to Drug B, as the less costly alternative. There, Drug B is not only less costly, but is also more effective than Drug A – Drug B dominates Drug A.

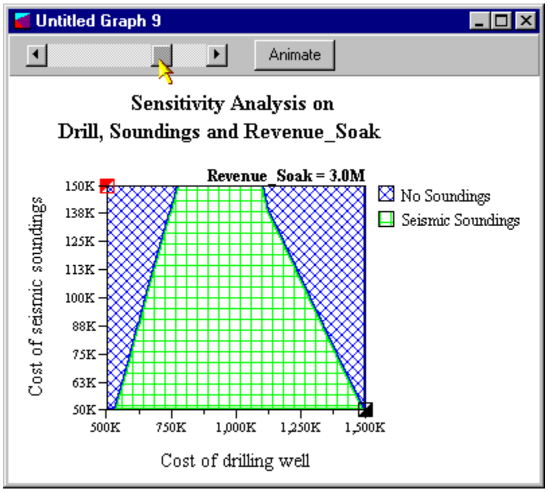
It would be more useful to use isocontours to look at an area where Drug B is still more effective, but is also more costly. Switching the order of the two alternatives at the decision node, so that Drug B becomes the topmost branch and the comparator, and rerunning the analysis, yields the second graph in the left margin.

Three-way sensitivity analysis

The results of a three-way sensitivity analysis are presented as an animated region graph, shown on the following page. At first glance it looks like a two-way graph, with the first variable assigned to the X axis and the second variable displayed along the Y axis. You may be wondering what happened to the third variable.

In a three-way analysis, the third variable changes value within the designated range; the number of changes is based on the number of intervals chosen by the user. Clicking on the Animate button (at the top of the graph window) will cause the third variable to cycle through its range, interval by interval. Alternatively, you can use the scroll bar to the left of the Animate button to move manually from frame to frame.





At each frame you will see a snapshot of the three-way analysis, showing you how the two-way analysis of the first two variables is affected by varying the value of the third variable. At each frame, the value of the third variable is displayed in a special label near the upper right corner of the graph.

TIP: Some users may experience difficulty printing region graphs from DATA. In a 16-bit operating system, each region can be assigned a maximum of 16K bytes of data. Printing at high resolution may cause a region to exceed this limit, leading to unpredictable printing errors. Possible solutions are to reduce the printer's resolution or the size of the graph region, change or eliminate region hash marks, or to export your graph as a snapshot or a screenshot, and print from another application. The best solution is to print from Windows NT.

Three-way sensitivity analysis is not available for cost-effectiveness calculations.

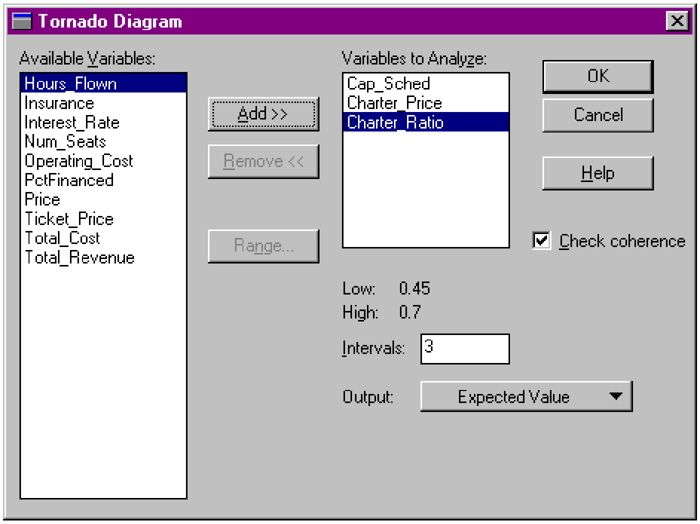
Unlike other sensitivity analysis graphs, the three-way sensitivity analysis does not include a legend displaying the name and the range of each variable’s correlates.

Tornado diagrams

A tornado diagram is a set of one-way sensitivity analyses brought together in a single graph. It can include all or a subset of the variables defined in your tree. You specify which variables are to be included in the analysis and assign a range of values to each of them.

In the resulting graph, a horizontal bar is generated for each variable being analyzed. Expected value is displayed on the horizontal axis, so each bar represents the selected node's range of expected values generated by varying the related variable. A wide bar indicates that the associated variable has a large potential effect on the expected value of your model. The graph is called a tornado diagram because the bars are arranged in order, with the widest bar (the most critical uncertainty) at the top and the narrowest one at the bottom, resulting in a funnel-like appearance.

The example file Airline Problem is ready for a tornado diagram. The model is a simple cost function, each of whose inputs may be varied to see how each may affect the expected value. The results may be seen in the file Tornado Graph, or you may create the tornado diagram yourself.



Available Variables:

Hours_Flown
Insurance
Interest_Rate
Num_Seats
Operating_Cost
PctFinanced
Price
Ticket_Price
Total_Cost
Total_Revenue

Variables to Analyze:

Cap_Sched
Charter_Price
Charter_Ratio

☑ Check coherence

Low: 0.45
High: 0.7
Intervals: 3
Output: Expected Value

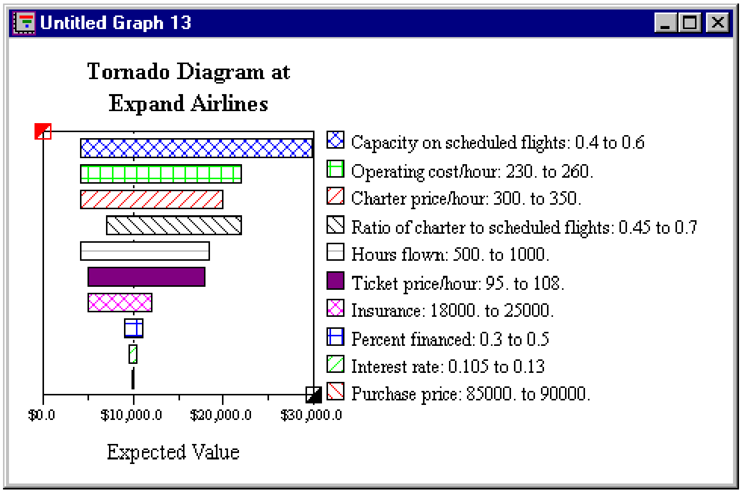
► To create a tornado diagram:

- ▷ Select a decision or chance node and choose Analysis > Sensitivity > Tornado Diagram.
- ▷ On the left side of the dialog box is a list of every variable in your tree. For each variable you wish to analyze, select it in the left-hand list, and click the Add button. This will move the variable to the list on the right, which contains all variables scheduled for analysis.
- ▷ DATA will ask you for a range for the variable.
- ▷ If the variable is correlated to other variables, you will also specify the correlations and their ranges at this point; see the section on correlated variables, below.
- ▷ If your variable has multiple definitions, you will be required to specify which definitions should be varied, and which definitions should be ignored; this process is described earlier in this chapter.

In light of the number of inputs required to set up the tornado diagram analysis, these parameters may be stored for future use; see Chapter 13 for details. Note that, as with any stored analysis, the parameters of the tornado diagram may not be modified from their stored values. In other words, if you repeat a stored tornado diagram analysis, all variables and ranges will be the same as when you first created the tornado diagram. If you need to change a range, or add or subtract a variable from the list to analyze, you will have to set up the entire analysis from scratch.

Understanding the results

If you click any bar once and hold down the mouse button, you will see the input and output range for that parameter. The *input* range is the range over which you varied the associated variable. The *output* range is the range of expected values that may occur when the variable is varied.



Each bar represents a one-way sensitivity analysis performed at the selected node. If you double-click on a bar, you will see the full line graph as it was generated from the sensitivity analysis. All relevant threshold information will be included.

The tornado diagram includes a vertical dotted line indicating the expected value at the selected node. You can use this as a visual “fulcrum” to view the impact of each variable relative to the original (baseline) expected value.

Tornado diagrams are available at either chance nodes or decision nodes. If you are performing the analysis at a decision node, thresholds within the specified range of values for an individual variable will be identified in resulting bars. A heavy vertical line will be drawn at each point where the optimal strategy changes.

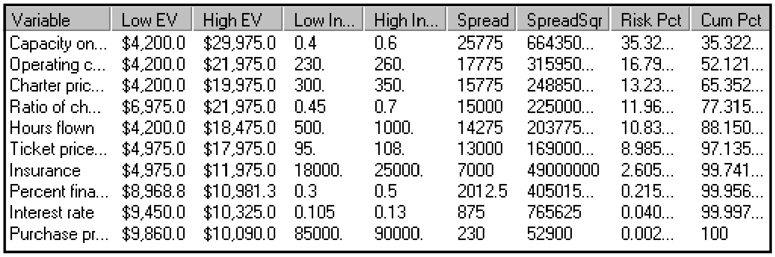
Note that these threshold indicators are drawn at the expected value at which the optimal path changes, not at the value of the variable. You should therefore use the heavy bars only as an indication that an optimal path change occurs; double-click the bar to view the full line graph with threshold information. Thresholds will often be obscured in the tornado diagram because they occur at either the left or right end of the bar. This situation will occur, for example, when one alternative's expected value is unchanged over the portion of the variable range for which it represents the optimal path. This is the case with almost every bar in the Airline Problem tornado diagram, where the value of the money market investment, $4200, is unaffected by changes in any variable.

Individual bars may be hidden and not displayed in the graph. Select Graph > Show/Hide (when the graph window is in front) to indicate which bars should be displayed.

A tornado diagram generated in a cost-effectiveness tree will show average values for cost-effectiveness, not marginal values.

Additional calculations in the text report

Selecting Graph > Text Report while the tornado diagram is active will display, in addition to the input and output ranges for each parameter, a number of other useful calculated values.



Variable	Low EV	High EV	Low In...	High In...	Spread	SpreadSqr	Risk Pct	Cum Pct
Capacity on...	$4,200.0	$29,975.0	0.4	0.6	25775	664350...	35.32...	35.322...
Operating c...	$4,200.0	$21,975.0	230.	260.	17775	315950...	16.79...	52.121...
Charter pric...	$4,200.0	$19,975.0	300.	350.	15775	248850...	13.23...	65.352...
Ratio of ch...	$6,975.0	$21,975.0	0.45	0.7	15000	225000...	11.96...	77.315...
Hours flown	$4,200.0	$18,475.0	500.	1000.	14275	203775...	10.83...	88.150...
Ticket price...	$4,975.0	$17,975.0	95.	108.	13000	169000...	8.985...	97.135...
Insurance	$4,975.0	$11,975.0	18000.	25000.	7000	49000000	2.605...	99.741...
Percent fina...	$8,968.8	$10,981.3	0.3	0.5	2012.5	405015...	0.215...	99.956...
Interest rate	$9,450.0	$10,325.0	0.105	0.13	875	765625	0.040...	99.997...
Purchase pr...	$9,860.0	$10,090.0	85000.	90000.	230	52900	0.002...	100

- **Spread** – This is the width of the bar (i.e., High EV - Low EV).
- **SpreadSqr** – The spread value, squared.

DATA also calculates a summary value called NetRisk, by adding the SpreadSqr values, in order to calculate two additional measures for each analysis parameter.

- **Risk Pct** – This indicates how much of the risk this bar represents (i.e., SpreadSqr / NetRisk). The RiskPct values sum to 1.0.
- **Cum Pct** – The cumulative version of Risk Pct. It makes it easy to scan the bars and say “to address 90% of the risk, I must consider the uncertainty represented by the following variables….”

Including correlated variables in the tornado diagram

As in other sensitivity analyses, if you select a variable for the tornado diagram which has correlations, you have the option of including the correlated variables in the analysis, as well. Correlates which are varied together in the analysis will appear as a single bar, for which you provide a description (e.g., "Market Prices").

The names of the correlated variables in a given group will not be displayed in the tornado diagram itself. If you single-click the bar representing those parameters, the names and input ranges of all correlated variables in the group will be shown. If you double-click the bar to open up the line graph, the variables will be displayed as in a normal one-way sensitivity analysis, with the *main* variable labelling the x axis, and its correlates and their ranges displayed beside the axis.

Threshold analysis

Threshold analysis is a specialized form of sensitivity analysis. It offers the ability to search thoroughly and accurately for threshold information. The result of this analysis is a detailed, textual description of how the optimal strategy is affected by changing the value of a single variable across a designated range.

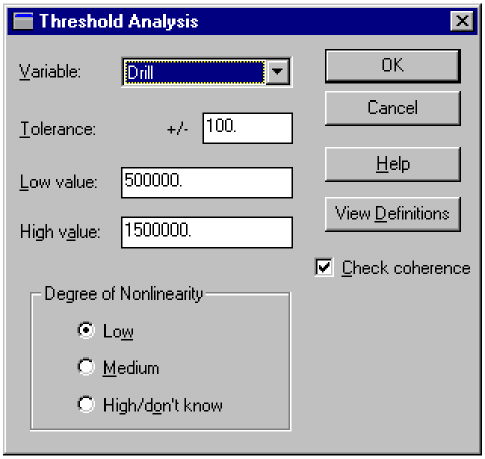
In a standard one-way sensitivity analysis, the user designates the number of intervals into which the range is to be divided; actual calculations occur only at these intervals. As a result, the accuracy of the associated threshold analysis is limited to values determined by linear interpolation.

In contrast, the Threshold Analysis menu option has been designed to maximize accuracy of the analysis in situations where accuracy is more critical than speed. The specified range is iteratively searched until a specified minimum tolerance is reached.

How threshold analysis works

After you select Threshold Analysis, a dialog box will appear asking four questions (these will be explained further, below):

- the name of the variable on which the sensitivity analysis is to be performed;
- the value range over which the designated variable is to be varied;
- a value for the “threshold tolerance”; and
- information concerning the non-linearity of the analyzed function.



Variable: Drill

Tolerance: +/- 100.

Low value: 500000.

High value: 1500000.

☑ Check coherence

Degree of Nonlinearity

● Low

○ Medium

○ High/don't know

If the variable in question has multiple definitions, you will be asked to specify which definitions should be varied, and which held constant. This part of the process is the same as for a normal, one-way sensitivity analysis. See the detailed discussion, above.

To get a sense of how the process works, you should perform a Threshold Analysis on the variable Drill at the root node of the Oil Drilling #2 tree. Use a tolerance of 100 and a non-linearity of "low," and assign a range of values for Drill from 500,000 to 1,500,000.

Tolerance

The tolerance is stated in the same units of value as the variable in question; it is not a percentage. The tolerance is related to the value of the variable, not to expected value. Thus, entering a tolerance of 100 means that the actual location of any threshold will be within plus or minus 100 units of the specified value. For example, if DATA indicates finding a threshold at Drill=639,024.4, this means that the threshold definitely occurs somewhere between 638,924.4 and 639,124.4. Because DATA applies linear interpolation after it meets your tolerance, you can expect the actual reported value to be even more accurate than the tolerance.

The designated tolerance has a second function. DATA uses this value as a basis for determining the number of decimal places (not significant digits) to specify in the result of its Threshold Analysis. The number of decimal places displayed will be one greater than the number of decimal places specified in the tolerance. Assume, for example, that DATA finds a threshold for a particular variable at 500.463333333. If the tolerance had been set at 0.01, the threshold value would be reported as 500.463. If, instead, a tolerance of 10 were used, a threshold of 500.5 would be displayed.

Nonlinearity

Performing a one-way sensitivity analysis on Drill from 500,000 to 1,500,000 will produce two threshold values. At both ends of the graph the optimal policy is not to perform the seismic soundings, while between about $640,000 and $1,160,000 the optimal policy is to do the seismic soundings.

However, this series of changes in policy will be identified correctly by DATA only if the thresholds appear in different intervals in the first iterative pass. Since linear interpolation is used to find thresholds in a sensitivity analysis, *only one threshold can be found per analysis interval*.

There is no way to avoid this problem entirely. DATA can even subdivide a range into 100 intervals and still miss policy changes within an interval if the same optimal policy is selected at both ends. Even if different strategies are optimal at either end of an interval, and DATA identifies a threshold in that interval, it is still possible that one or more additional thresholds in that same interval will have been missed.

For example, three alternatives, A, B, and C, might be compared using a sensitivity analysis; A is optimal at the beginning of an interval, B in the middle, and C at the end. Although you know that two thresholds (A to B, then B to C) actually occur, DATA will find just one (a nonexistent one, A to C) from looking at the optimal alternative at the ends of the interval.

The non-linearity hint is an attempt to minimize the likelihood this will occur. The more nonlinear you describe the graph's shape to be, the smaller the interval used by DATA, so as to insure catching any double thresholds.

Performing a one-way sensitivity analysis on the variable in question before performing a threshold analysis will indicate the appropriate non-linearity setting. If the sensitivity analysis graph is very nonlinear (i.e., it has multiple thresholds), use a higher setting for the measure of non-linearity. This will cause DATA to increase the number of intervals searched at each step. Increasing the non-linearity setting also increases the time needed to perform the analysis. For this reason, it is not recommended that you automatically use the Medium or High settings.

Initially, DATA will subdivide the given range into a number of intervals. The number of intervals searched relates to the non-linearity radio buttons as follows:

- Low: 4 intervals
- Medium: 8 intervals
- High/Don’t Know: 12 intervals.

At each interval where a change in optimal strategy is identified, DATA will either

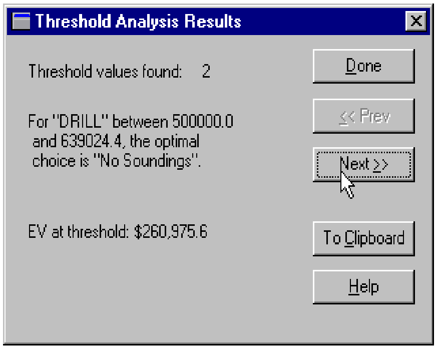
- calculate a threshold value, if the width of the range is less than twice the given tolerance; or
- redivide the interval into 4, 8, or 12 subintervals, as indicated, and search those, as above.

As the calculation proceeds, the progress bar shows how far over the given range DATA has searched. If a threshold value is found, the bar will slow down considerably, but it will move more quickly over intervals in which no threshold value is found.

Understanding the results

A dialog box will appear with the results of the Threshold Analysis. It will state the total number of threshold values identified by the analysis, specify the optimal policy between the low end of the range and the first threshold (change in policy), and identify the expected value at the threshold.

For example, if a Threshold Analysis is performed at the root node of the Oil Drilling #2 tree by varying Drill in the range from 500,000 to 1,500,000, the first dialog box would provide the information illustrated here.



Threshold values found: 2

For "DRILL" between 500000.0 and 639024.4, the optimal choice is "No Soundings".

EV at threshold: $260,975.6

The dialog box specifies a single interval throughout which the optimal policy is consistent. The term “EV at threshold” refers to the expected value when the variable in question is given the value at the top of the interval being described. The “Prev” and “Next” buttons can be used to view each of the other policy intervals.

The To Clipboard button is used to transfer all of the threshold information to the clipboard in text format.

Other sensitivity analysis tools

Two additional tools may prove useful in analyzing the critical variables in your tree. Sliders, which are discussed in Chapter 9, allow you to implement a semi-automated sensitivity analysis. You can create a slider to adjust graphically the numeric value of a variable. After you select a value, you can perform whatever analysis or calculation you like, and note the results. Adjust the slider again, and perform the analysis using a different value. This strategy can be useful to view the results of some analyses – a probabilistic distribution of outcomes, or the final probability of a state in a Markov process, for instance – which DATA cannot produce directly from a sensitivity analysis.

Monte Carlo simulation provides a different way to analyze the effects of uncertainty. It is based on the random sampling of values (probabilities and payoffs) from discrete and continuous distributions during individual trials. Observing the statistical properties of many trials can provide additional insight into a model's behavior in a more realistic setting. See Chapter 29 for more information.

Chapter 23 Expected Value of Perfect Information

When faced with making a decision under uncertainty, the question often arises whether it would pay to expend the time and resources needed to eliminate or reduce the uncertainty before making the decision.

Assume that you could acquire information that perfectly predicted the outcome of a particular uncertainty, so that you would know exactly which decision to make. What would this information be worth to you?

Keep in mind that perfect information does not mean that you can control the outcome; it means that you have acquired a perfect *predictor* of the outcome. In a decision tree, one normally models the acquisition of perfect information by taking the chance node representing the issue in question and moving it to the left of the decision node. The probabilities associated with the uncertainty are now the probabilities associated with the outcomes to be perfectly predicted.

If you roll back the revised tree, it is likely to have an improved expected value relative to the original tree – higher if you are maximizing benefits, lower if you are minimizing costs. The absolute value of this change in expected value is known as the expected value of perfect information (EVPI).

While predictive information is rarely perfect, the usefulness of EVPI is in calculating a maximum value of information. If perfect information in a particular situation has a base value of *x*, one should certainly not pay more than *x* for *imperfect* information.

The remainder of this chapter discusses the implementation of EVPI in the tree window. For more information on using EVPI in the influence diagram window, please see Chapter 32, Advanced Influence Diagram Features.

A simple example

Before examining DATA’s automation of EVPI, it is worth spending a few moments to see what the analysis portrays.

▷ Open the file Stock Tree.

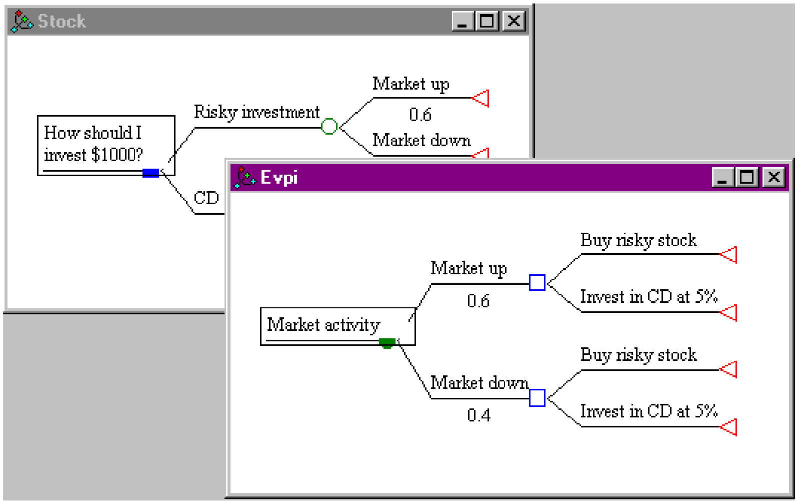
▷ Roll back the tree. Its expected value is $60.

▷ Without closing the Stock tree, open the file EVPI Tree.

This tree contains an inverted representation of the investment problem. You will never need to create such a tree, but it is included to illustrate the mechanism behind EVPI.

▷ Roll back EVPI Tree. Its expected value is $320.

▷ Subtract the expected value of Stock Tree from the expected value of EVPI Tree. The difference is $260.



This is the expected value of having perfect information about the market activity modeled in Stock Tree. Note that the chance node precedes the decision, meaning that the market activity is known when you decide how to invest your money.

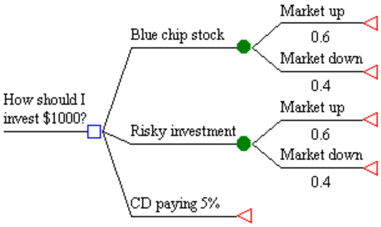
This is the maximum amount you should be willing to pay to obtain perfect information about this market activity. It also affords some basis for appraising the value to you of a less than perfect predictor of market activity.

► **To calculate EVPI automatically:**

- ▷ Go back to the Stock tree; make sure that Roll Back is turned off.
- ▷ Select the chance node Risky investment.
- ▷ From the Analysis menu, choose Expected Value of Perfect Info.

DATA will report a value of $260, the same value you just calculated.

A more complex example



The file Two Stock Tree shown at left illustrates a slightly more complicated situation. As you will recall from Chapter 5, each of the two stock investments under consideration is followed by the same uncertainty — whether the market will go up or go down during the period being considered.

In order to have DATA calculate EVPI on the state of the market, both "stock" nodes must be selected. (See Chapter 11 for how to select multiple nodes.) To generalize, before selecting the EVPI menu item, *it is necessary to select all nodes in the tree which represent the same event*. In the example shown, both the Risky investment and Blue chip stock chance nodes represent the uncertain market activity.

In addition, all selected chance nodes must be siblings in a generational sense, and they must be descendants of the same decision node(s). Moreover, all selected nodes must have identical branches emanating from them. To be identical, each set of branches must have the same probabilities. As long as these immediate branches are identical, it does not matter if there are differences in the subtrees further to the right.

Avoiding EVPI's pitfalls

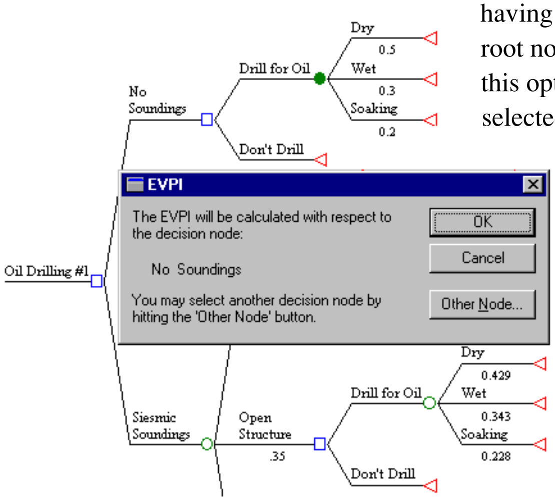
EVPI is a shorthand for placing the chance node to the left of the decision node. If there is more than one decision node in the ancestry of the selected chance node(s), DATA will give you the opportunity to identify the decision node at which you want to calculate EVPI; if there is only one decision node in the ancestry, DATA will use that node as the basis for its calculations.

It is important for you to be alert to the danger that it is possible to force invalid calculations by having DATA calculate EVPI under circumstances where it would make no sense to perform the analysis manually.

For example,

- ▷ Open Oil Drilling #3.
- ▷ Select the Drill for Oil node in the No Soundings subtree (the topmost Drill for Oil node).
- ▷ Pull down the Analysis menu, and select Expected Value of Perfect Info.

In the resulting dialog boxes, you are presented with the option of having the analysis performed at the Oil Drilling #3 node (the root node of the tree) or at the No Soundings node. You are given this option because there are two decision nodes to the left of the selected uncertainty.



EVPI

The EVPI will be calculated with respect to the decision node:

No Soundings

You may select another decision node by hitting the 'Other Node' button.

Performing the calculation at the No Soundings node is similar to the analysis undertaken above in connection with the EVPI tree. It certainly makes sense to calculate the value of knowing the state of oil before deciding whether or not to drill.

If, instead, you were to tell DATA to perform the EVPI calculation at the Oil Drilling #3 node, you would find that DATA will return a value of $437,500, or $87,500 higher than at the No Soundings node. Is this meaningful information?

The structure of the oil drilling problem already includes the option of securing imperfect information in the form of a seismic test. The decision whether or not to perform this test occurs at the root decision node. Placing the chance node which represents the uncertain amount of oil to the left of this decision is meaningless. Having already received *perfect* information, the decision whether to obtain additional *imperfect* information regarding the same subject would have no value or relevance.

This inconsistency is exposed by understanding the process behind EVPI. Before you calculate and interpret the expected value of perfect information, remember that you are inverting the time-ordering of a single chance event and a single decision. In the tree, this means moving the chance node to the left of the decision node. In this spurious case, the decision being considered – whether or not to obtain imperfect information – does not lend itself to EVPI.

Chapter 24 Bayes' Revision

If your model includes imperfect tests or forecasts, you may wish to utilize Bayes' revision to ensure that correct probabilities are used. DATA can automatically perform the calculations that implement probability revision using Bayes' theorem. The process occurs once, during the initial construction of the model; based on your answers to three questions, DATA will generate a set of variable definitions that represent the calculated, *inverted* probabilities. The probability expressions will be recalculated every time the model is evaluated, and results can change as your estimates of prior and likelihood probabilities (see below) change.

Bayes' revision is implemented in both the tree and influence diagram windows. Bayes' revision in the tree window is able to revise probabilities automatically based upon a single test. To revise probabilities associated with sequential tests, you should initially build your model as an influence diagram. In the influence diagram window, DATA can handle correlations among any number of tests. The complex set of inverted probability variable definitions will be created upon converting the influence diagram to a tree.

Influence diagrams are also useful, although not strictly necessary, if you need to create a tree modeling two or more *independent* tests. These might, for instance, be tests of independent events or conditions (e.g., the exposure of a person to a particular substance and that substance's toxicity). The influence diagram's probability revision process makes it easier to set up this kind of model correctly.

This chapter describes the use of Bayes' revision for models constructed in the tree window. For information on applying Bayes' revision to influence diagrams, refer to Chapter 32, Advanced Influence Diagram Features.

A brief introduction to Bayes' revision

Bayes' revision, also known as probability revision, allows decision makers to calculate *decision* probabilities from *likelihood* probabilities. Likelihood probabilities, or forecast likelihoods, are answers to questions in the form, "If this test is performed on a part *known* to be faulty, what is the probability of a positive result, indicating a problem?" This type of probabilistic information is often available, but is not immediately useful in making decisions. What is needed are the decision probabilities, which are calculated using Bayes' revision and address questions such as, "If a particular part tests positive, what is the probability that this part is faulty?"

The distinction between likelihood probabilities and decision probabilities is subtle, but vitally important. All probabilities express a relationship between some *evidence* and a *hypothesis*. The difference is that a forecast likelihood describes an inference *from a hypothesis to some evidence*, while a decision probability describes the standard inference, from evidence to hypothesis.

A likelihood probability represents the probability of obtaining evidence (i.e., the result of a test) that correctly or incorrectly matches the hypothesis (i.e., the presence or absence of a specific underlying condition) *when the hypothesis is already known to be true or false*; it is a retrospective measure of the accuracy of the forecast.

A decision probability represents the probability of the hypothesis being true or false given a certain piece of evidence; it is not a direct measure of forecast accuracy. The decision probabilities are so named because in the real world, they are the probabilities upon which decisions are based. These are also sometimes called posterior (or *a posteriori*) probabilities.

The basic formula for inverting probabilities is relatively simple, although its intuitive application can be quite difficult:

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)}$$

where H is the hypothesis and E is the evidence. The formula is applied once for each hypothesis-evidence combination. Each P(E|H) (the probability of E given H) represents a likelihood probability (such as the probability of a true positive), and each P(H) represents a prior (or *a priori*) probability. The marginal values P(E) are calculated as part of the revision.

A manufacturing illustration

The following example is designed to offer a sense of the meaning and usefulness of Bayes' revision. If you are already familiar with the type of applications that require Bayes' revision, you may want to skip this section.

Consider an automated test for defect X in a semiconductor. The defect is present in 1% of the items under scrutiny; this is the *a priori* probability. It has been demonstrated that the available test will detect 98% of the faulty materials, meaning that 2% of those pieces with defect X will not be picked up by the test. Also, the test is known to incorrectly identify as faulty 3% of those pieces that are without defect.

You have under consideration installing a machine to perform this test in your facility. What is the likelihood that a part which tests positive actually has defect X? How certain can you be parts that tested negative don't have a defect?

Information about the accuracy of the testing equipment provided by its manufacturer does not provide the probabilities needed to answer these crucial questions. Bayes' revision must be applied to this information you have in order to turn the test into a useful diagnostic tool.

Once you know (or be able to estimate) the prevalence of condition X in the items or population being tested, then the test likelihoods can be applied. For our purposes, let's say we have a batch of 10,000 items to be tested. If the estimated prior probability of defect is 1% (from previous experience), we would expect 100 to have defect X. Of these 100, about 98 should test positive. Of the 9,900 pieces without the defect, we would expect approximately 297 to test positive. Thus, a total of 395 (297 + 98) test subjects would test positive.

The first decision probability is the ratio 98/395, or approximately 25%; this represents the probability that a positive test result actually indicates the presence of the defect. In this case, 75% of the positive results are in error. The other decision probabilities can be similarly calculated. With this information in hand, the decision maker can evaluate the likely performance of the new test, and easily compare it with existing methods and competing technologies.

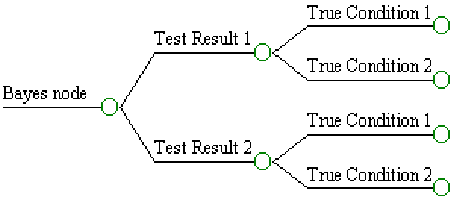
Before you begin

To use DATA's automatic probability revision, you should first obtain numeric values for the likelihood probabilities associated with the test and *the a priori* probabilities for the hypotheses.

Then, your tree must be set up with a specific structure:

- The root of the Bayes subtree (the *Bayes node*) must be a chance node; it does not have to be the root node of your tree.
- The immediate descendants of the Bayes node must be chance nodes. They represent the possible outcomes of the test or forecast (the evidence; e.g., Test Abnormal or Test Normal). If you are modeling a two-outcome (*dichotomous*) test for a binary hypothesis and you have test sensitivity and specificity information, see the following tip on the healthcare version of Bayes' revision.
- The test subtree must be symmetrical, i.e. each test outcome (e.g., T+ and T-) must have an identical subtree whose branches represent the possible underlying conditions (the hypothesis; e.g., C+ and C-). You may not use clones to set up these duplicate subtrees.

Branches should be named descriptively. DATA will walk you through the process of inputting the appropriate values or variables, and will need the branch descriptions in order to be able to phrase questions about the known and calculated probabilities.



Note that the structural limitations specified above apply only at the time of using DATA's automatic Bayes' revision command. After DATA calculates and inserts the decision probabilities, you may refine the structure of your model to include asymmetries or intermediate events, including decisions interposed between the evidence and hypothesis nodes. The structural requirement allows DATA to decipher the natural structure of your problem during the probability revision calculations.

The structural limitations specified above do not apply when Bayes' revision is used in the influence diagram window. See Chapter 32.

Bayes' revision in DATA

Once you have properly constructed the test subtree, you can choose its root node and perform Bayes' revision. DATA will then ask a series of questions. If the Bayes subtree represents a dichotomous test for a binary hypothesis (two possible results, two possible conditions), DATA will ask if you have sensitivity and specificity information. If this is not a medical test, you will generally say no.

Then, for each hypothesis, DATA will ask you to enter the *a priori* probability that the condition is present (in the population). DATA will also ask you to enter the test likelihoods associated with the hypothesis.

If you provide variable names in the dialog expression fields, the a priori and likelihood probabilities that you enter in the value fields will be stored in these variables. This will allow you, for example, to perform sensitivity analysis on these probabilities. It is also possible to enter only numeric probabilities, using just the expression fields.

After you have completed the entry of probabilities, DATA will create an additional set of variables representing the decision probabilities to insert into your tree, and each will automatically be defined in terms of the appropriate calculated value.

TIP: A medical version of Bayes' revision is available if your test subtree includes exactly two test results for two conditions. DATA will ask you initially if you have sensitivity and specificity information for the test in question. If you say yes, you will also need to indicate which evidence node represents a positive test result and which hypothesis node represents the presence of the condition for which you are testing.

► **To perform Bayes' revision:**

▷ Open the sample file Oil Drilling Bayes.

The oil drilling problem was introduced in Chapter 8, using the files Oil Drilling #1 and #2. Both of these examples used numeric probabilities, rather than variables. In the No Soundings subtree, simple probability estimates were used for the outcomes of Drilling:

• Dry:	0.5 (or 50%)
• Wet:	0.3 (or 30%)
• Soaking:	0.2 (or 20%)

The probabilities used in the Seismic Soundings node in Oil Drilling #1 and #2 were actually calculated manually with the Bayes' revision formula, using the prior probabilities shown above and a table of likelihood probabilities. These likelihood probabilities reflect the result of seismic soundings performed at sites where the results of drilling (the state of the site) is known. For example, it might be demonstrated that seismic soundings performed at a site know to be dry will indicate no structure 60% of the time, open structure 30% of the time, and closed structure 10% of the time. The following table lists the likelihood probabilities for each of the known states (dry, wet, and soaking):

	No Structure	Open Structure	Closed Structure
Given Dry	0.6	0.3	0.1
Given Wet	0.3	0.4	0.3
Given Soaking	0.1	0.4	0.5

Thus, if you obtain knowledge about the underlying geological structure through seismic soundings, you should revise your initial probability distribution (Dry=0.5, Wet=0.3, Soaking=0.2) of the extractable oil deposits at the site.

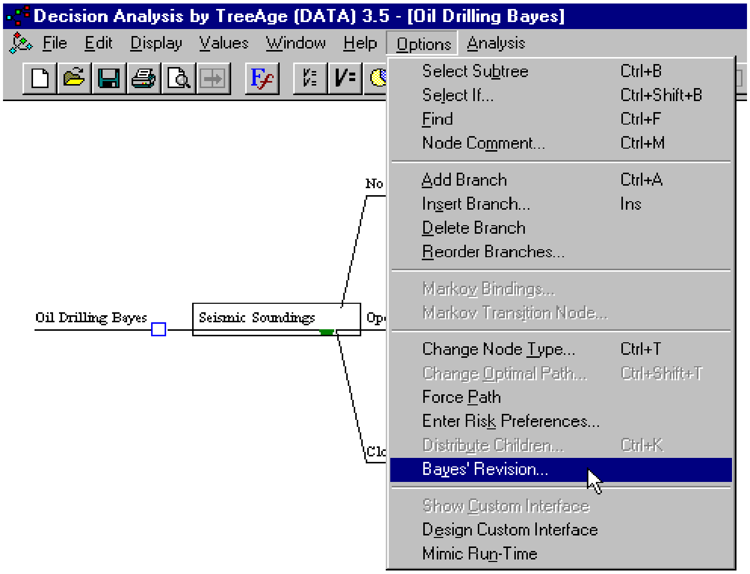
Although the probability revision calculations can be done by hand, DATA is able to manage this much more efficiently. Moreover, if the tree is set up using DATA's Bayes' revision dialog, it will be possible to carry out sensitivity analysis out on prior and likelihood probabilities.

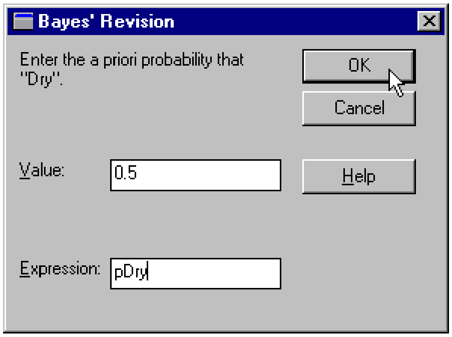
The Oil Drilling Bayes model, a simplified representation of the Seismic Soundings subtree in the other Oil Drilling trees, will be used to demonstrate how to perform Bayes' revision.

Before beginning, you should take a moment to examine the structure of the Oil Drilling Bayes tree. The reason for including an extra decision node at the root will be explained later. The chance node closest to the root node has three branches. These represent the possible results of the test on the underlying structure of a potential oil field. The ultimate condition of interest to the decision maker, though, is the amount of oil that can be extracted, not the geology of the location. This uncertainty is represented by the three subtrees with the branches Dry, Wet, and Soaking.

Let's get started with Bayes' revision.

▷ Select the Seismic Soundings node, and choose Options > Bayes Revision.


Decision Analysis by TreeAge (DATA) 3.5 - [Oil Drilling Bayes]
File Edit Display Values Window Help Options Analysis
Select Subtree Ctrl+B
Select If... Ctrl+Shift+B
Find Ctrl+F
Node Comment... Ctrl+M
Add Branch Ctrl+A
Insert Branch... Ins
Delete Branch
Reorder Branches...
Markov Bindings...
Markov Transition Node...
Change Node Type... Ctrl+T
Change Optimal Path... Ctrl+Shift+T
Force Path
Enter Risk Preferences...
Distribute Children... Ctrl+K
Bayes' Revision...
Show Custom Interface
Design Custom Interface
Mimic Run-Time
No
Oil Drilling Bayes
Seismic Soundings
Op
Clo



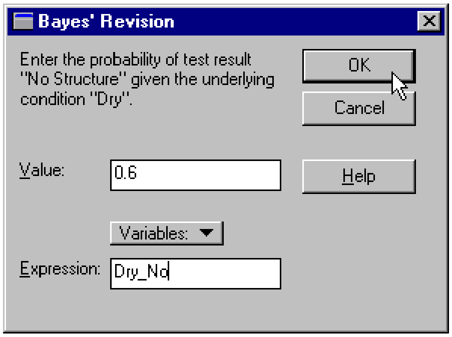
Enter the a priori probability that "Dry".

Value: 0.5

Expression: pDry

▷ For the *a priori* probability of Dry (your estimate prior to receiving the results of seismic sounding) enter `0.5` in the Value field and `pDry` in the Expression field. Click OK. DATA will not recognize the name pDry, so you should accept the suggestion to create a new variable. Click OK to accept the defaults in the Properties dialog.

Alternatively, you can enter only a variable name in the Expression field, and leave the Value field blank. Then, the variable can be defined independently, either in its Properties dialog or using the other variable definition methods discussed in Chapter 9. See the discussion of variables, below, for more detail.



Enter the probability of test result "No Structure" given the underlying condition "Dry".

Value: 0.6

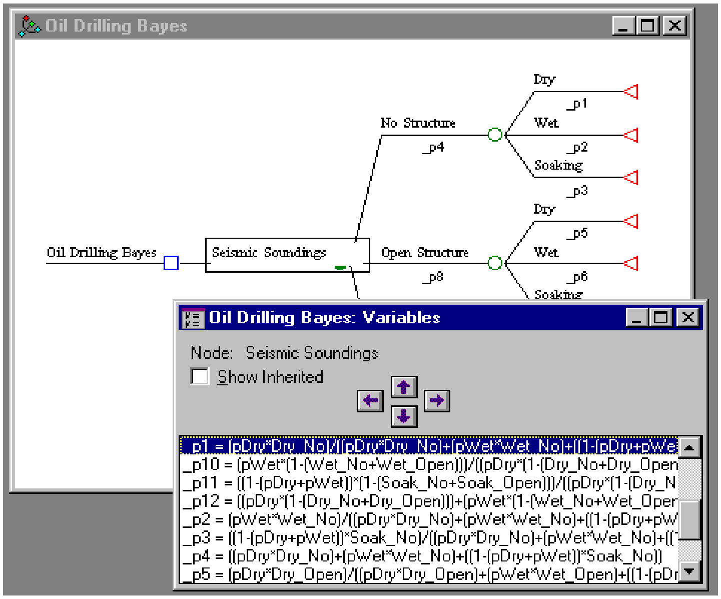
Expression: Dry_No

▷ For the likelihood probability that no structure will be detected in a field known to be dry, enter 0.6 in the Value field and pDry_No in the Expression field. Click OK. DATA will not recognize the name pDry_No, so you should create a variable as you did with pDry, above.

▷ For the likelihood probability that open structure will be detected in a field known to be dry, enter 0.3 in the Value field and pDry_Open in the Expression field. Click OK. DATA will not recognize the name pDry_Open, so you should create the variable as before.

- ▷ For the likelihood probability that closed structure will be detected in an area that is dry, accept the default expression 1-(pDry_No+pDry_Open).
- ▷ For the *a priori* probability of Wet use the value `0.3` and the new variable pWet. For the likelihood probability that no structure will be found in a wet area, use the value 0.3 and the new variable pWet_No. For the likelihood probability that open structure will be found in a wet area, use the value 0.4 and the new variable pWet_Open. For the likelihood probability that closed structure will be found in a wet area, accept the default expression 1-(pWet_No+pWet_Open).
- ▷ For the *a priori* probability of Soaking accept the default expression 1-(pDry+pWet); this will calculate to 0.2. For the likelihood probability that no structure will be found in a soaking area, use the value 0.1 and the new variable pSoak_No. For the likelihood probability that open structure will be found in a soaking area, use the value 0.4 and the new variable pSoak_Open. For the likelihood probability that closed structure will be found in a soaking area, accept the default expression 1-(pSoak_No+pSoak_Open).

After you enter the last value and expression pair, DATA will calculate the decision probabilities and define new variables at the Bayes node.



Node: Seismic Soundings

□ Show Inherited

```
_p1 = (pDry*Dry_No)/((pDry*Dry_No)+(pWet*Wet_No)+((1-(pDry+pWe
_p10 = (pWet*(1-(Wet_No+Wet_Open)))/((pDry*(1-(Dry_No+Dry_Open
_p11 = ((1-(pDry+pWet))*(1-(Soak_No+Soak_Open)))/((pDry*(1-(Dry_N
_p12 = ((pDry*(1-(Dry_No+Dry_Open)))+(pWet*(1-(Wet_No+Wet_Oper
_p2 = (pWet*Wet_No)/((pDry*Dry_No)+(pWet*Wet_No)+((1-(pDry+pW
_p3 = ((1-(pDry+pWet))*Soak_No)/((pDry*Dry_No)+(pWet*Wet_No)+((
_p4 = ((pDry*Dry_No)+(pWet*Wet_No)+((1-(pDry+pWet))*Soak_No))
_p5 = (pDry*Dry_Open)/((pDry*Dry_Open)+(pWet*Wet_Open)+((1-(pDr
```

The complex expressions used to define these variables (with names like "_pn") will not immediately be displayed in the tree, even if you have variable display turned on. DATA automatically turns off the Show in tree property of these variables. See Chapter 9 for information on changing variable properties and related tree preferences. To view the definitions, you can select the Bayes node, and choose Values > Variables Window.

You may later perform sensitivity analysis on any of the variables you created to represent underlying quantities, both prior probabilities and likelihoods. You should not perform sensitivity analysis directly on the decision probabilities used below the chance node branches; analyzing the component prior and likelihood probabilities, instead, will correctly cause the values of the decision probabilities to change.

Changing structure in the Bayes subtree

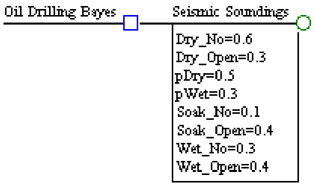
Once you have completed the Bayes' revision process, it will probably be necessary to modify the structure of the tree in a number of ways. The file Oil Drilling #3 illustrates a tree initially set up using Bayes revision, and then modified to reflect the various decisions represented in Oil Drilling #1 and #2. To get from the Oil Drilling Bayes tree to the completed Oil Drilling #3 tree, you must add the initial decision whether or not to undertake seismic tests, and the four decisions whether to drill.

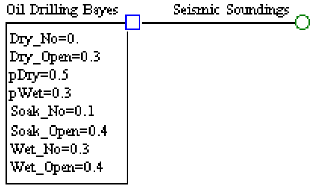
Chapter 5 covered the basic tasks of inserting, deleting and reordering branches, as well as copying and pasting subtrees. These methods may be used to update the structure of your model after Bayes' revision.

In the Oil Drilling Bayes tree, you can insert three of the required decision nodes by selecting No Structure, Open Structure, and Closed Structure, one at a time, and choosing Options > Insert Branch. Choosing to insert a branch to the right of each will create new chance nodes. Their existing branches will represent the outcomes of drilling; thus, these new nodes should be named Drill for Oil.

Since Drill for Oil represents a decision alternative, its parent should be changed to a decision node. The second alternative, Don't Drill, must be added to each of the three new decision nodes, and the Don't Drill nodes must be changed into terminal nodes.

Adding the initial decision, whether to undertake the seismic testing, requires close attention to detail. In this case, an extra decision node was included at the outset to simplify matters somewhat. Nonetheless, it





is critical to take into account the location of the variable definitions created during Bayes' revision.

In the case of the Oil Drilling problem, the same definitions of the prior probability variables pDry and pWet should apply in both the Seismic Soundings and No Soundings subtrees. However, as explained later in this chapter, the prior and likelihood probabilities may have been defined at the Seismic Soundings node, but not at the root node. To move the definitions to the root node, where they can be made accessible to both subtrees, you can edit the definitions at the Seismic Soundings node, deleting the existing expressions. Next, you would recreate the same definitions at the root node, at least for the variables pDry and pWet.

It is possible to avoid this problem; see the section on variables, below.

Using variables in the Bayes' revision dialog

The top, Value field in each Bayes' revision dialog is optional; the Expression field is not. It is not permissible to enter only a number in the Value field; you must enter an appropriate expression (i.e., variable, formula, or number) to be used in calculating decision probability variables, as described on the next page. Typically, you will enter a single variable name in the Expression field, and a number in the Value field. DATA will then define the variable using the numeric value at the Bayes node, and utilize the variable in setting up formulas to calculate the decision probabilities. If you want to perform Bayes' revision on raw numbers only, and do not wish to create variables, you may enter the numeric values in the expression fields. However, this will foreclose any opportunity to perform sensitivity analysis on the prior and likelihood probabilities, and will otherwise limit the ability to update the decision probabilities' formulas.

Two types of entries are valid in the Expression field. You may enter the name of a variable, as in the above examples, or you may enter an expression, such as "pCan1+pCan2." Expressions are scanned for new variables. You may *not* use the hash mark (#) in the Expression field.

Any variables already created in the tree may be selected from the pop-up menu and inserted into the expression.

If the expression is a simple variable and not a formula, the variable will be defined at the Bayes node with any numeric value you assign in the Value field of the dialog. Variables defined in this manner will be good candidates for sensitivity analysis.

If you leave the Value field blank, no definition of the variable will be added to the tree. This is useful if, for instance, you defined prior and likelihood variables before performing Bayes revision, or if you are defining the variables using their default numeric value property.

Bayes node location

As noted above, it may be important to consider the final location of the Bayes node in your tree – perhaps at the root of the tree, but perhaps more likely at an internal location.

If you use a new variable name during the Bayes' revision process, DATA will present the familiar variable Properties dialog. Setting a default numeric value using the variable Properties dialog always creates a definition at the root node of the tree. Thus, if the Bayes node is also the root node of the tree, setting this value in the Properties dialog will be the same as entering that value directly in the Bayes dialog. If the two definitions are different, the value entered in the Bayes dialog will be used, overwriting the default value property.

If, on the other hand, the Bayes node is an internal node, entering a numeric value in both the Bayes dialog and the variable Properties dialog will create definitions in two places: at the Bayes node and at the root node of the tree. The root node definition will, of course, be overridden in the Bayes subtree by the Bayes node definition. If the values are the same, there will be no immediate harm. The risk is that the value will later be changed at the root node definition, and not the Bayes node definition, resulting in incorrect calculations.

CHAPTER 25 BASIC MARKOV MODELING

This chapter covers the basics of implementing Markov processes in DATA. It assumes that you have a reasonable understanding of Markov (recursive) processes. In addition to illustrating software features, some conceptual background is provided.

Although the examples used in this chapter are healthcare-related, Markov models have been used in other fields of decision analysis, including engineering and finance. In these fields, the processes being looked at are often evaluated using measures like time to failure, rather than in terms of life expectancy or quality-adjusted life expectancy. The same methods and tools illustrated using medical models can be used in modeling manufacturing, insurance, and financial processes.

Chapter 27 will discuss more advanced Markov topics.

Recursiveprocesses

Most decision trees include a simple notion of time (i.e., events to the right of the tree occur after those to the left). There are no strict rules about representing time in a standard tree structure, though. A Markov model, on the other hand, is designed to represent a *recursive* process – a series of events which unfold over time in fixed intervals – and Markov structures and values must be clearly related in time.

Markov models can be used to simulate short-term processes (growth of a tumor) or long-term processes (life cycle of a power generating plant). A wide variety of outcomes (e.g., expected utility, long-term costs of maintenance, survival rate, or number of recurrences) can be calculated. Markov models have both retrospective and predictive applications. Finally, although this chapter refers to *individuals* moving through a process, Markov models can be applied to any life cycle – that of a machine, disease or organization, as well as a person.

The basic Markov model requires that you define a finite set of *states* in which an individual can be found. The states must be enumerated in such a way that, in any given interval, the individual will be in one state only – no more and no less.

The progress of a Markov model is viewed and evaluated at discrete time intervals. The length of this interval, the model's *cycle length*, is determined by the model-builder. Any useful interval can be used – an hour, a day, or a year – but it must remain fixed for the duration of the calculation. (Monte Carlo simulation can get around this limitation; see Chapter 29.)

Between cycles, an individual may move from one state to another, or remain in the same state. Which transitions are possible at the end of the interval will depend on the state the individual has been in during the current interval. While many transition paths may be available to an individual, only one may be taken at the conclusion of a given interval. These *transitions* are probabilistically defined.

The standard way to analyze a Markov model is using a *cohort simulation*. A very large group of individuals – the cohort – is run through the model, and viewed probabilistically. Thus, if the probability of making a transition from state A to state Z is 0.2, then 20% of the cohort membership in state A at cycle n will be in state Z at cycle n+1. At the end of each interval, after all transitions have occurred, the results of all transitions are summed to provide the percentage of the cohort in each state. Continuing with the example, if 50% of the cohort began cycle n in state A, then 0.1 (0.5 * 0.2 = 0.1) is added to the percentage of the cohort in state Z at cycle n+1.

Markov models can also be analyzed using Monte Carlo simulation; this subject is covered in Chapter 29.

Basic components of a Markov model

When designing a Markov model, it is helpful to view it as having four components:

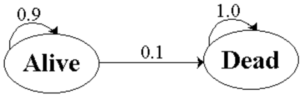
- **Structure** – A model's structure consists of a list of states, together with the transitions specified for each state.
- **Probabilities** – Each transition must be assigned a probability; the set of transition probabilities for each state must sum to 1.0. A separate set of probabilities must describe the initial distribution of the Markov cohort among the states immediately before the process begins.
- **Rewards** – In a decision tree, the term *payoffs* refers to the values of the scenarios being modeled (e.g., costs, patient utilities, number of lives saved, etc.). In a Markov model, the term *rewards* is roughly comparable.

- **Termination Condition** – There must be some way of stopping calculations, to prevent an infinite recursion. The *termination condition*, or stopping rule, is a test performed at the end of each cycle to determine if the process should continue calculating.

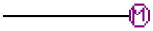
Representation

The standard (or canonical) graphical representation of a Markov model uses circles to represent the states, and arrows to represent the allowed transitions. If an individual may remain in a given state from one cycle to the next, this is indicated by an arrow that points back to the state from which it begins.

The simplest Markov model includes just two states:



The numbers along the arrows indicate the probabilities of making the given transition during each cycle. The probabilities on the arrows emanating from any state must sum to 1.0.

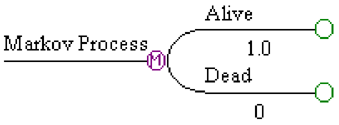


DATA does not provide this representation of a Markov process, but uses instead what is called a *cycle tree*. To build a cycle-tree representation of a Markov model, a special node type, called a Markov node, must be used. A Markov node, and its associated cycle tree, can be attached to a standard DATA decision tree anywhere you might place a terminal node.

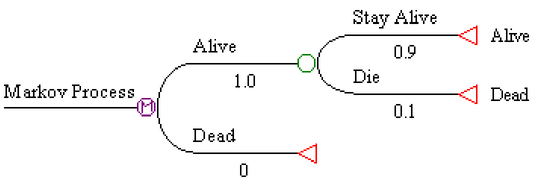
TIP: In a decision tree which includes a Markov subtree, the Markov node acts like a terminal node when the tree is evaluated. The Markov subtree functions like a calculator, with its value representing an outcome's payoff.

The Markov node marks the entrance into the Markov portion of a decision tree. Markov nodes can be placed anywhere in the tree structure – they can be the root node of a tree or they can follow a series of decisions and chance events. Any number of Markov nodes can be included in a decision tree. Note that you cannot represent decisions or additional Markov processes within a Markov subtree (i.e., to the right of a Markov node).

The branches of the Markov node enumerate the states of the model. Here, the nodes Alive and Dead represent the states.



The values below the branch lines indicate the probabilities of beginning the process in each state; these initial probabilities must sum to 1. In this simple example everyone begins the process alive, so the initial probability of the Dead state is 0.



The subtree emanating from a state indicates the allowed transitions from that state. Thus, the Alive node is represented as a chance node; its children, Stay Alive and Die, represent the allowed transitions for an individual who is in the Alive state.

To the right of each transition node is the name of the state to enter at the beginning of the next cycle. Below the branch line is the probability of making the transition (e.g., from Alive to Dead) at the end of any interval.

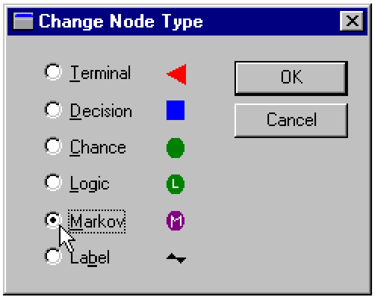
Terminal nodes in a Markov model do not necessarily indicate a final outcome, as they do in a standard decision tree structure. In a Markov model, terminal nodes indicate the last event of a particular interval and show where individuals should go for the following cycle.

The Dead state is an *absorbing state*: there are no transitions from Dead to other states. An absorbing state, like a transition node, is represented using a terminal node symbol. Models are not required to have an absorbing state.

DATA's use of arcs rather than straight lines to draw branches in Markov subtrees is simply to make it easier to distinguish a Markov model from a decision tree.

Building a Markov model in DATA

Now you will build the simple Markov model illustrated above. To begin building a Markov subtree, the Markov node type must be used. The root node of an empty tree can be changed to a Markov node; alternatively, a Markov subtree can be appended to an existing model.



○ Terminal
○ Decision
○ Chance
○ Logic
● Markov
○ Label

▶ **To create a Markov node:**

▷ Create a new tree, if necessary.

▷ Select a node without any descendants; it can be the root node of an empty tree, or any right-most node in a tree.

▷ Choose Options > Change Node Type..., click on Markov, and press ENTER or RETURN.

▷ Type `Markov Node` for a text description of the new Markov node.

Now, the standard tree-building tools covered in Chapters 3 and 5 (such as Options > Change Node Type, Add Branches, and Insert Branch) can be used to create the Markov subtree.

► **To construct a Markov subtree:**

▷ Choose Options > Add Branches to attach two Markov states to the Markov node.

▷ Type `Alive` and `Dead` above the two new branches; below the branches, enter the initial probabilities, `1` and `0`, respectively.

DATA will use the names entered above the branch lines to reference the Markov states when transitions are created; see below. The initial probabilities entered below the branch lines will be used only once, when initiating the Markov process. In the Markov subtree, as in standard tree structures, variable expressions may be used instead of numeric values to define probabilities.

> **TIP:** Initial probabilities are used only once during the evaluation of a Markov model, to specify the initial distribution of the cohort at cycle zero. All subsequent calculations utilize transition probabilities. See Chapter 27 for a discussion of initial rewards, prior costs, and the half-cycle correction.

Specifying transitions and absorbing states

▷ Select the Alive node, and choose Options > Add Branches to create the simple transition subtree for this Markov state.

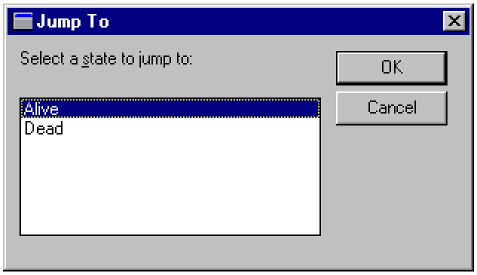
▷ Type `Stay Alive` and `Die` above the Alive node's two branches; below the branches, enter the transition probabilities, `0.9` and # (to calculate the remainder probability), respectively.

To create a transition node, select a right-most node in the Markov subtree and change its node type to terminal. Instead of prompting for a payoff value, as in non-Markov tree structures, DATA will ask you to indicate which state the transition points to.

▷ Select the Stay Alive node.

▷ Choose Options > Change Node Type.

▷ Click on the Terminal button, and press ENTER or RETURN.

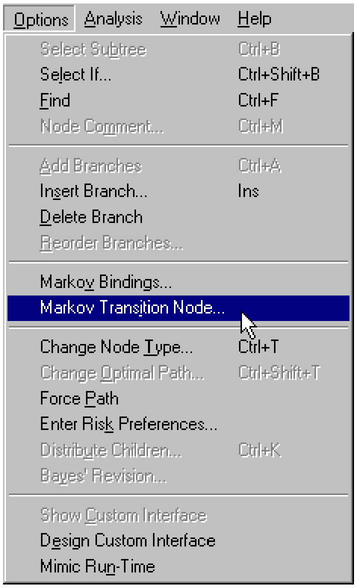


▷ In the Jump To dialog, select Alive from the list of existing states, and press ENTER or RETURN.

▷ Select the Die node, choose Options > Change Node Type, click on the Terminal button, and press ENTER or RETURN.

▷ In the Jump To dialog, select Dead from the list of existing states, and press ENTER or RETURN.

This completes the transition subtree for the Alive state.



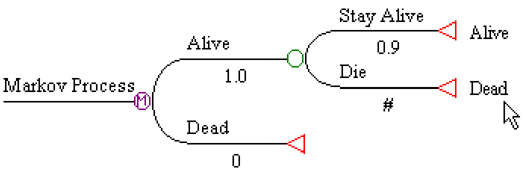
Once a particular jump-to state has been assigned to a transition node, it is possible to change the transition by selecting the transition node and choosing Options > Markov Transition Node. Try this with the Alive node. Also, if the name of a Markov state is changed, DATA will automatically update existing transition to use the new state name.

> **TIP:** In DATA 3.0, the name of the transition node itself was required to match the name of the state being pointed to. DATA 3.5 allows you to name the transition node as you like, with the name of the jump-to state stored (and entered) separately.

The Dead node will represent an absorbing state, with transitions only back to Dead.

▷ Select the Dead node, choose Options > Change Node Type, click on the Terminal button, and press ENTER or RETURN.

The Jump To dialog does not appear when you create an absorbing state. If you select a Markov state (an immediate descendant of the Markov node) and change its node type to Terminal, DATA automatically sets the transition for that state back to itself.



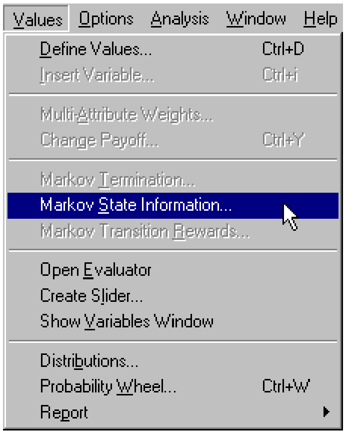
Normal transition nodes will display the assigned jump-to state's name in the area to the right of the node; absorbing states will not. It is not possible to turn off the display of the jump-to state names.

▷ Save the partially complete tree as 2-State Markov.

Assigning rewards

An assignment of value in a Markov model is called a reward, whether you are trying to minimize cost or maximize benefit. Rewards can be accumulated in both states and transitions. Rewards are typically assigned as an incremental quantity received by the membership of a given state during any cycle. While there are other types of rewards you will eventually need, the incremental state reward is the most important. Other reward types (such as a cost or benefit associated with a particular transition, or a onetime reward assigned for beginning or ending the Markov model in a particular state) will be discussed in detail in Chapter 27.

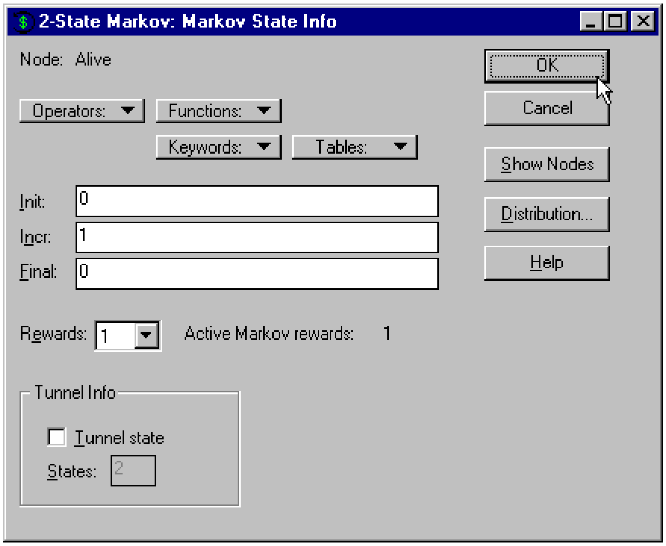
In the 2-State Markov tree, assume that yearly transition probabilities (e.g., the probability of dying) are being used, so the model's cycle length is one year. The simplest analysis to perform using this Markov model is to calculate average life expectancy (for the cohort); the calculation should be in terms of life years (not days). The transition probabilities for this model have defined a cycle as one year, so an incremental reward of 1 (not 365) should be assigned to the membership in the alive state during each cycle.



► **To define incremental state rewards:**

▷ Select the Alive node and choose Values > Markov State Information....

▷ Ensure that the Rewards pop-up menu is set to 1 (equivalent to Payoff #1 in standard terminal nodes), and enter `0`, `1`, and `0` in the three text boxes.


2-State Markov: Markov State Info

Node: Alive

Init: 0

Incr: 1

Final: 0

Rewards: 1 Active Markov rewards: 1

Tunnel Info

□ Tunnel state

States: 2

Values must be entered for all three types of state rewards for every Markov state, even if all rewards are zero. DATA will issue an error message upon calculation of the Markov node if any state reward is left blank.

▷ Press ENTER or RETURN to accept the entered set of state rewards.

Assigning an incremental reward of 1.0 means that, for every cycle that an individual is alive, one unit of reward (one year of life) is accumulated. Once complete, evaluating the 2-State Markov model will yield the *average* life expectancy for a member of the cohort. At each successive cycle, the probabilistic redistribution among the Markov states is calculated; in a given state, a percentage of the incremental reward is accumulated based on the percentage of the cohort then in a given state.

Absorbing states require the same specification of the three state reward components. Note that members of the cohort that enter an absorbing state continue to accumulate incremental rewards until the Markov process terminates.

To complete the 2-State Markov model, assign an incremental reward of 0 to the Dead state.

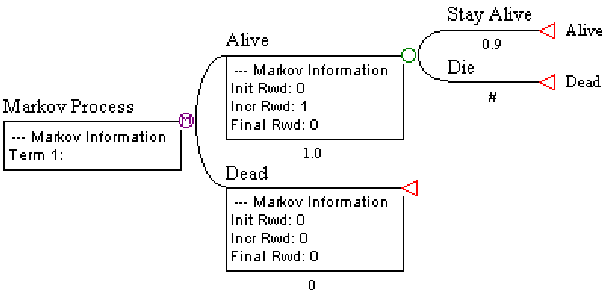
▷ Select the Dead node and choose Values > Markov State Information....

▷ Ensure that the Rewards pop-up menu is set to 1, and enter `0`, `0`, and `0` in the three text boxes.

▷ Press ENTER or RETURN.

Displaying Markov information

You may display all Markov rewards (along with other Markov information) directly in the tree window, as you would variable definitions. In the Preferences dialog, select the Variables Display category. Select Full definitions in tree, and also check Show Markov information. You may also wish to select Expand node to fit variables. It is not possible to selectively display Markov information.



Adjusting for cycle length

Markov cycle length is not specified explicitly in the tree. Instead, it must be considered by the model builder when defining other values: rewards, probabilities, and termination condition.

Multiple factors may influence the ultimate choice of cycle length: the maximum duration of the entire process being modeled (e.g., a lifetime versus a two-month drug trial); the duration of the shortest useful interval; and the time frame of available probability data (e.g., monthly versus yearly survival), as well as cost and other reward information.

Incremental state rewards are also known as *time-dependent* rewards because the value used depends on the cycle length. If a particular state has a certain incremental cost per year, that reward value must be adjusted if a cycle length other than one year is used.

The unit of a single cycle does not have to be equivalent to the unit used in calculating rewards; in the 2-State Markov example, it is possible to calculate life expectancy in months, while still using yearly transition probabilities. An incremental reward of 12 (months), in place of 1 (year), would be accumulated for each interval spent in Alive. What if, on the other hand, monthly transition probabilities were used? The Markov Process tree still could calculate life expectancy in years, despite the 1 month cycle length. Simply assign an incremental reward of 1/12 (of a year) to the Alive state.

TIP: Generally, the cycle length remains fixed for all states and all intervals in the model. It may be possible to create a Markov process with a variable cycle length, using tunnel states or Monte Carlo tracker variables. Advanced Markov topics, such as these, are covered in Chapter 27.

Setting the termination condition

It may appear that the 2-State Markov model will terminate when all members of the cohort are dead. During cohort simulation, though, a Markov process is evaluated probabilistically, as though the cohort had an infinite number of members. In this model, the number of living subjects declines exponentially. In other words, more and more members of the cohort will die over time, but there is *never* a time when *all* members are dead. The alive portion of the cohort dwindles asymptotically toward, but never reaching, zero.

In the 2-State Markov model, it is necessary to specify a mathematical approximation of the time at which all cohort members are dead. For instance, the process can be stopped when 99.9% of the cohort is dead, and the remainder treated as an approximation error.

The model can also be set to run for a specified number of cycles, independent of the distribution of the cohort. This would be appropriate, for example, if the process being modeled where a treatment of fixed duration (e.g., a drug therapy of 10 doses).

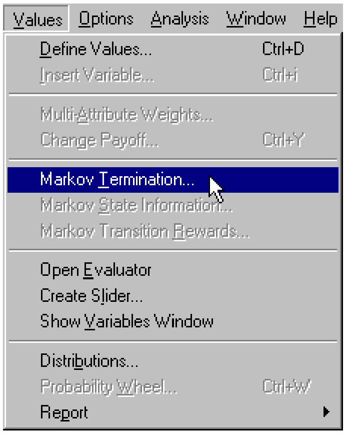
At the end of each cycle, the termination condition you specify is evaluated. If it evaluates to *true*, the Markov process is completed, and the net reward can be calculated.

Using Markov keywords

DATA provides several built-in variables, called *keywords*, which are available only in a Markov node or its subtree:

- **_stage** – contains the number of the current cycle being evaluated.
- **_stage_reward** – contains the reward received by the cohort in the previous cycle.
- **_total_reward** – contains the cumulative value of all previous stage rewards; at the end of calculations, the value of this keyword is used as the overall expected value of the process itself.

The keywords can be used to define an appropriate termination condition; they can also be used in probability and reward expressions elsewhere in the Markov model. These Markov keywords have the same value throughout the Markov subtree during a particular cycle. There are several other keywords, used in tunnel states and in cost-effectiveness Markov models; these are discussed in Chapter 27.

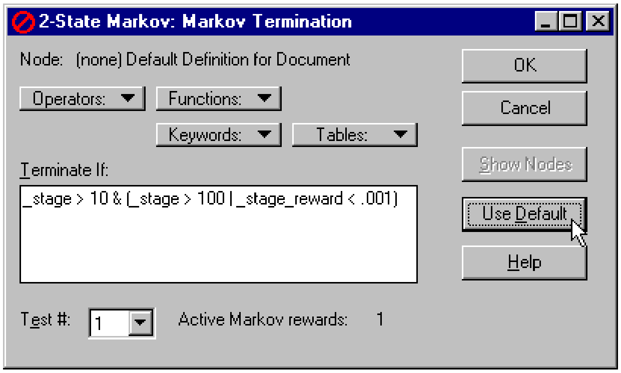


It is possible to combine multiple logical and mathematical statements in the termination condition. An example is the default termination condition, which you can view by selecting a new Markov node and choosing Values > Markov Termination. The default expression, which is not meant to be used unedited, is:

_stage > 10 & (_stage > 100 | _stage_reward < .001)

The vertical bar (|) means OR, and the ampersand (&) means AND. See Appendix C for more information on logical and relational operators.

The default condition specifies that the process should terminate when a minimum number of stages has elapsed (11), and either a maximum number of stages has elapsed (101) or the net reward accrued during a single stage drops below a threshold (0.001). A threshold value is often used for a termination condition when there is a zero reward for being in an absorbing state. As more people from the cohort enter the absorbing state, the net accrued reward per stage approaches, but may never reach, zero. Using an appropriately small threshold value allows the process to be terminated when the cohort has been sufficiently absorbed.



Node: (none) Default Definition for Document

Terminate If:

_stage > 10 & (_stage > 100 | _stage_reward < .001)

Test #: 1 Active Markov rewards: 1

The expressions used for the minimum and maximum number of cycles, and for the threshold stage reward, are always model-dependent. DATA's default termination condition should not be accepted without consideration of these values.

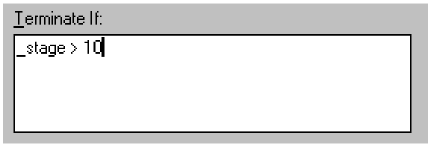
For the purposes of the 2-State Markov model, it is desirable to simplify the stopping rule. The following steps will cause the model to be run for 11 cycles, regardless of the value of _stage_reward.

► **To enter the termination condition:**

▷ Select the Markov node of the 2-State Markov tree.

▷ Choose Values > Markov Termination.

▷ In the data-entry box, enter `_stage > 10` as the termination condition for this model.



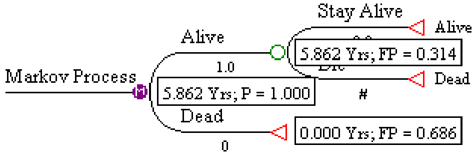
Terminate If:

_stage > 10

▷ Save the now complete Markov tree.

Analyzingthemodel

Once you have entered the rewards and the termination condition, you may analyze your model. To see how DATA calculates an expected value for the Markov process, roll back the tree, changing the numeric formatting if necessary (use 3 decimal places, and a custom unit suffix of “Yrs”); see Chapter 10 for details.



Interpreting the roll-back display

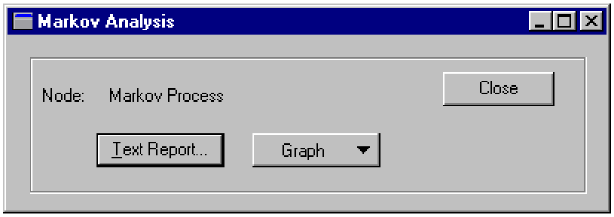
Next to the Markov node is the value 5.862 Yrs. This indicates that a member of the cohort can expect to receive this much reward during the process (within the 11-cycle limitation imposed by our stopping rule). The one-unit-per-cycle reward translates into a life expectancy value of 5.862 years.

The box next to the Alive state also contains the value 5.862. A member of the cohort can expect to receive this much reward while in the Alive state. In single attribute models, the sum of the values next to all states in the model will sum to the value of the Markov model itself. In this case, all of the reward was received by individuals while they were alive.

The FP value next to each state represents its *final probability*, or the portion of the cohort in that state at the end of the process.

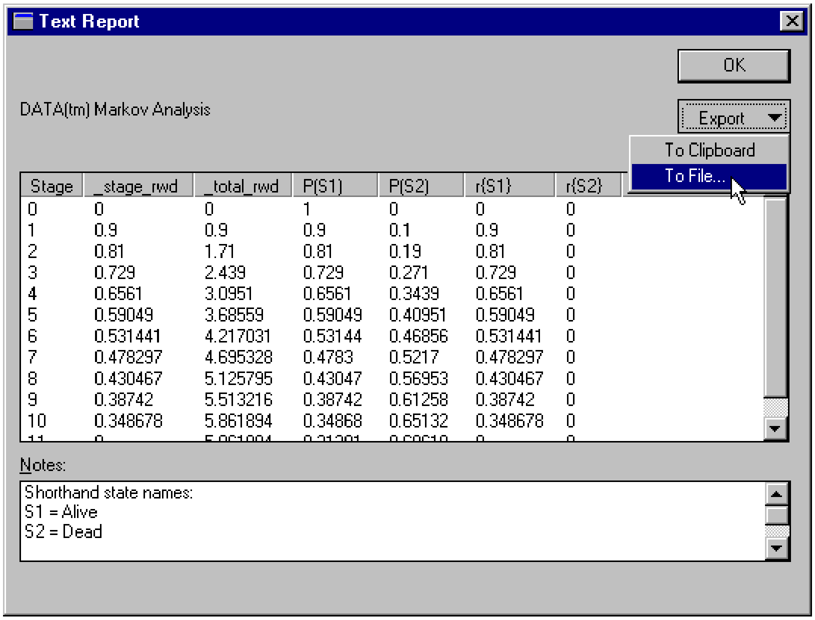
Markov analysis

Performing a Markov analysis provides a higher level of detail of the calculations which are used to provide the basic roll back analysis output. Select the Markov node and choose Analysis > Markov analysis to perform the cohort simulation at that node. When the analysis finishes (i.e., the process is terminated), you will see an interim window that offers several output options.


Node: Markov Process

Here is a summary of the different output options:

- **Text Report** – The full trace of the Markov calculations is shown in the text report. All of the graphical options (discussed next) are contained numerically within this report.

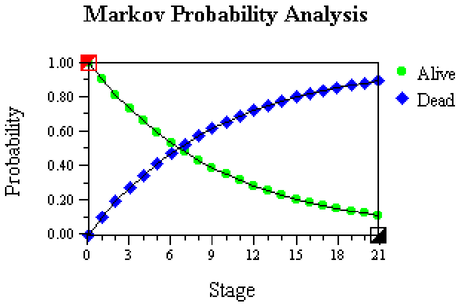


DATA(tm) Markov Analysis

Stage	_stage_rwd	_total_rwd	P(S1)	P(S2)	r{S1}	r{S2}
0	0	0	1	0	0	0
1	0.9	0.9	0.9	0.1	0.9	0
2	0.81	1.71	0.81	0.19	0.81	0
3	0.729	2.439	0.729	0.271	0.729	0
4	0.6561	3.0951	0.6561	0.3439	0.6561	0
5	0.59049	3.68559	0.59049	0.40951	0.59049	0
6	0.531441	4.217031	0.53144	0.46856	0.531441	0
7	0.478297	4.695328	0.4783	0.5217	0.478297	0
8	0.430467	5.125795	0.43047	0.56953	0.430467	0
9	0.38742	5.513216	0.38742	0.61258	0.38742	0
10	0.348678	5.861894	0.34868	0.65132	0.348678	0
11	0	5.861894	0.31381	0.68619	0	0

Notes:

Shorthand state names:
S1 = Alive
S2 = Dead



- **State probabilities graph** – This graph shows how the cohort is distributed at each cycle. If you view the graph for this model, you should recognize two important features that have already been mentioned: the transition from Alive to Dead is exponential; and a significant portion of the cohort is still alive at the end of the eleventh cycle.
- **State rewards graph** – This graph shows, for each state, what reward was received at each stage. (For cost-effectiveness models, this graph type is disaggregated into state costs and state utilities graphs.)
- **_stage_reward and _total_reward graphs** – One line is shown for each graph, representing the value of the keyword (discussed above) at each cycle.

Monte Carlo simulation

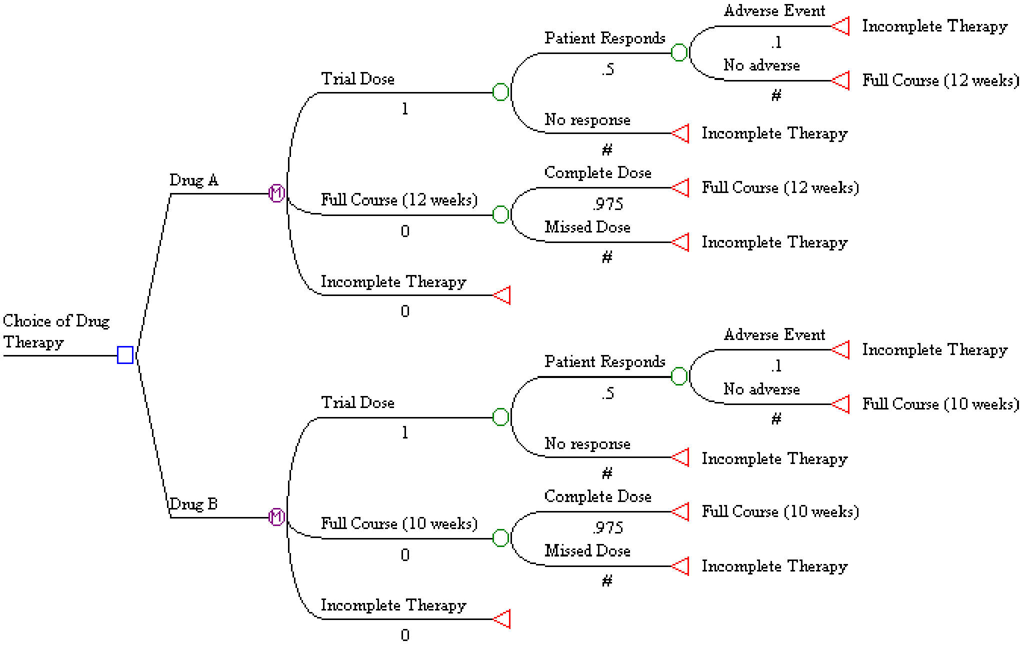
The roll back and Markov analyses options evaluate Markov models using cohort simulation methods. Monte Carlo simulation offers another way to analyze your Markov model, using individual trials. For some problems, both Markov cohort simulation and Monte Carlo simulation may be relevant; for others, only one or the other method may be appropriate. See Chapter 27, on advanced Markov topics, for more details on Monte Carlo simulation of Markov subtrees.

The general use of Monte Carlo analysis with decision trees is discussed in more detail in Chapter 29.

A more complex Markov model

The sample file Complex Markov compares two hypothetical drug therapies on the basis of cost-effectiveness. Utility is measured as the percentage of the cohort starting a therapy who successfully complete the regimen (effectively, the number of cures).

Drug A and Drug B are presumed to have the same response and side effect rates. Drug B requires one less dose, though, reducing the risk of failure due to a missed dose. Drug B is also more expensive per dose. In each Markov process, a cycle represents a bi-weekly dose of medication.



Open the file and perform a roll back, a cost-effectiveness analysis, or Markov analysis. The calculation method can also be changed from Cost-Effectiveness to Simple, in order to look at cost or utility, alone.

Chapter 26 Tables

In DATA, a table is a type of variable that contains a series of different values. For example, tables can be used to store the mortality rates of patients at different ages or stages of disease, the changing values of a transition probability or state reward in a Markov process, or a distribution of values to sample during a Monte Carlo simulation.

How tables are stored

In DATA for Windows, each table is stored as a separate file (with a .TBL extension) in the Tables subdirectory of the program directory (e.g., C:\PROGRAM FILES\DATA\TABLES). In DATA for Macintosh, tables are stored in a folder entitled Tables which must be located in the folder that holds the DATA 3.5 application. See Appendix E, Technical Notes, for information on changing the drive or directory in which table files are stored.

Tables are global; no table data is stored in tree documents. This means that all tables are accessible to all trees. However, it also means that if you want to transfer to another computer a tree that uses particular tables, you must also transfer all table files referenced by that tree to the TABLES folder on the new computer.

DATA's tables utilize a proprietary format. You cannot simply place a spreadsheet file or database table into the tables directory and expect DATA to read it or write to it. There are easy ways to exchange table information with other programs, as discussed below.

Contents of a table

Tables have two columns. The first column contains the Index; the second column contains the associated Value. Indexes need not be consecutive or integral. For example, the following (named myTable) is a valid table:

Index	Value
0	1.0
2.4	12.0
18	6.4

A table has three properties:

- its internal name (i.e., the name by which you refer to the table in expressions);
- its file name; and
- its lookup method.

To use a table in a formula, refer to it by its internal name, with the index in square brackets. For example, myTable[18] would return the value 6.4.

In order to share a tree that includes table references with another user, you will also need to provide the appropriate table files, found in DATA's Tables directory. If you use the same name for the table internally and for its file, it will be easier to manage the contents of your Tables folder.

Lookup method

To accommodate different uses, DATA can utilize one of three lookup methods with each table you create. It is very important that you understand how each method works.

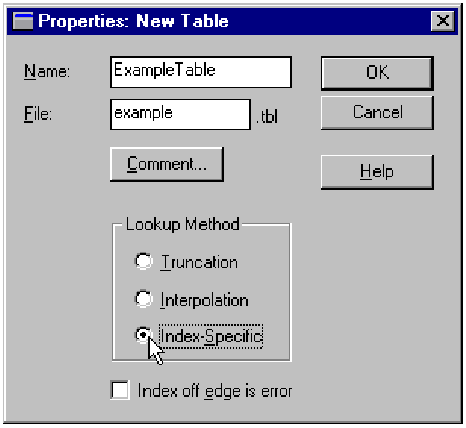
- **Truncation** – returns the value associated with the largest index that is less than or equal to the requested index. So myTable[x] would return 1 for all values of x greater than or equal to 0 and less than 2.4.
- **Interpolation** – will return a value which is found by linear interpolation between successive indexes.
- **Index-Specific** – will only produce valid return values when the exact requested index is included in the table; all other values will cause DATA to report a table-lookup error.

In addition, each table has an option called “Index off edge is error.” If this option is checked, DATA will generate a table-lookup error any time you attempt to access the table with an index less than the lowest defined index or greater than the highest defined index. If you leave the option unchecked, values beyond the boundaries of the available table indexes will return the value associated with the lowest or highest available index, respectively.

Creating tables

► **To create a table, specifying its properties:**

▷ From the Values menu, choose Define Values. Ensure that the Show Tables button at the bottom of the window is checked.



Properties: New Table

Name: ExampleTable

File: example .tbl

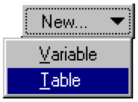
Lookup Method

○ Truncation

○ Interpolation

● Index-Specific

□ Index off edge is error



▷ Press the New button, and select Table from the pop-up menu.

▷ Give your table both an internal name and a file name. (If you are using the 16-bit version of DATA for Windows, the file name can be at most eight characters, and the ".tbl" extension is automatically added for you.)

▷ Select a lookup method (interpolation is often a good choice), and check or uncheck the Index off edge is error checkbox. You may optionally add a longer comment to the table.

▷ Press OK to create the empty table file.

> **TIP:** New tables can also be created on-the-fly by entering a table reference in an expression – *pTable[_stage]* for a probability, for instance.

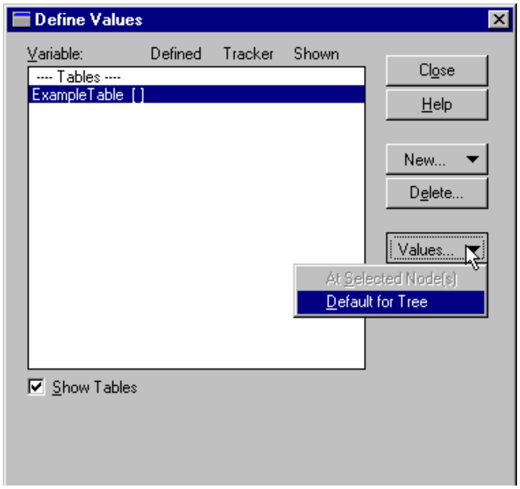
► **To edit the properties of a table:**

▷ From the Values menu, choose Define Values. Ensure that the Show Tables button at the bottom of the window is checked.

▷ Select the table from the list, and click Properties. Make any changes you desire, and press OK.

Entering values in a table

A table can be populated either by direct data entry or via an imported, two-column set of values.



Define Values

Variable: Defined Tracker Shown

---- Tables ----

ExampleTable [ ]

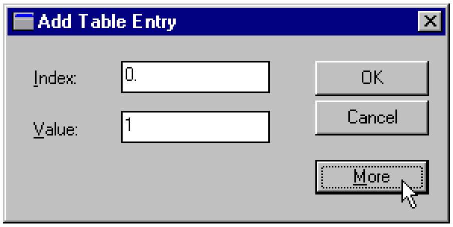
☑ Show Tables

► **To enter data manually into a table:**

▷ From the Values menu, choose Define Values. Ensure that the Show Tables button at the bottom of the window is checked.

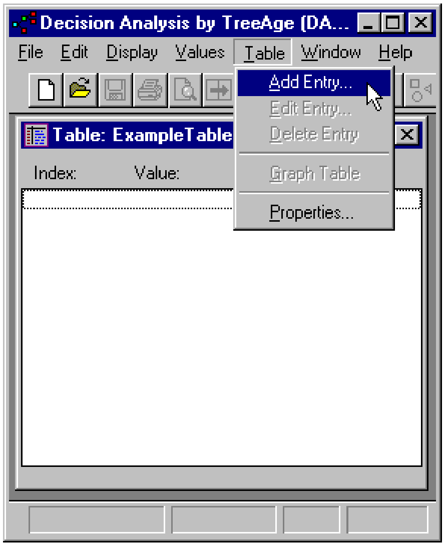
▷ Select the table from the list. Press the Values button, and choose Default for Tree from the pop-up menu. (Of course, the values in the table are default for *all* your trees.)

If the selected table is new, or otherwise empty, the Add Table Entry dialog will automatically be opened over the Table Window.



Index: 0.

Value: 1



▷ In the Add Table Entry dialog, assign a new index/value pair.

▷ You may enter multiple values in quick succession by using the More button in the Add Entry dialog.

▷ Press OK when you are done.

Once you have closed the initial Add Table Entry dialog, a table window will open showing the current contents of the table, and the Table menu will appear in the menu bar. This menu is available only when a table-editing window is in front.

► **To create additional table entries:**

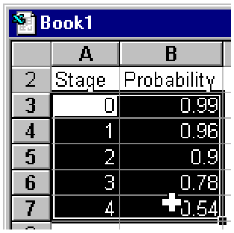
▷ From the Table menu, choose Add Entry.

▷ Assign a new index/value pair. You may enter multiple values in quick succession by using the More button in the Add Entry dialog. Press OK when you are done.

With the table-editing window in front, other editing functions are also available. The Table menu contains menu items for adding, deleting, or modifying entries. In addition, it contains an option to convert your table into a line graph. You may also edit the properties for the table by choosing Table > Properties.

Importing tables

The preferred method for building tables is to first create the list of indexes and values in a spreadsheet, database, or word processor, and then copy and paste it into a DATA table. If you choose to use a word processor, be sure that you use tabs between columns, and that nothing appears to the left of the Index column. When using a spreadsheet, the relevant portion must format-clean (free of currency symbols, commas, and other text.). Also, the columns for indexes and values must be adjacent.



	A	B
2	Stage	Probability
3	0	0.99
4	1	0.96
5	2	0.9
6	3	0.78
7	4	0.54

► **To import a table from another program:**

▷ Select the contents of your table in the spreadsheet or word processor. If your table has a row of titles (such as *Index* and *Value*), do *not* include the titles in your selection.

▷ Switch to DATA. Open a table-editing window as described above.

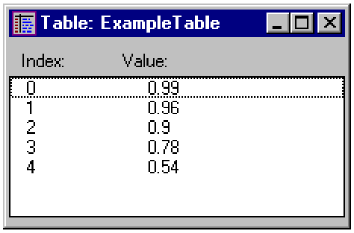


Table: ExampleTable

Index:	Value:
0	0.99
1	0.96
2	0.9
3	0.78
4	0.54

▷ Choose Edit > Paste Table. If the Paste Table command is not available, check the spreadsheet to ensure that the copied cells are format-clean (no currency or other text formatting).

This operation will overwrite the current contents of the table, if any.

There is another way to create a table within DATA: by importing values from a line graph generated by DATA, such as a one-way sensitivity analysis graph or a risk preference function graph.

► **To create a table from a line graph:**

▷ With a line graph in front, choose Graph > Line to Table. If your graph contains more than one line, select which line should be converted into a table.

▷ Enter the properties for the new table, as discussed above.

▷ A table-editing window will open with the contents of your table. You may edit it or close it.

Some analysis output, such as a Monte Carlo simulation text report, must be exported to a spreadsheet and cleaned up before it can be pasted into a DATA table.

See Chapter 33 for more information on graphs.

Exporting tables

You may export a table by choosing Edit > Copy Table when a table-editing window is in front. The table is placed on the clipboard in tab-delimited format. This text may then be placed into any word processor or spreadsheet.

Chapter 27 Advanced Markov Modeling

This chapter covers advanced concepts and features associated with Markov processes and their implementation in DATA. It assumes that you have read Chapters 25 and 26, and can build basic Markov tree structures and set up tables in DATA.

Cycle-dependent values

The 2-State Markov model that is the subject of the tutorial in Chapter 25 is an example of a basic form of Markov process, known as a *Markov chain*, in which all parameters remain constant throughout the analysis. Markov chains, which mathematicians represent using a simple table of transition probabilities (a *p-matrix*), do not require an explicit, cycle tree representation such as that used in DATA.

$$\begin{Bmatrix} 0.9 & 0.1 \\ 0 & 1.0 \end{Bmatrix}$$

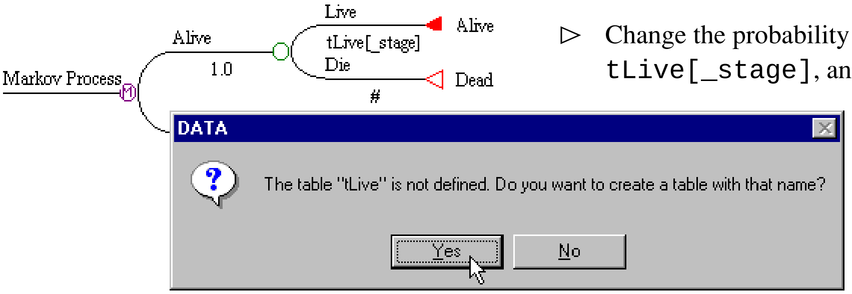
However, in order to model many real-world problems, transition probabilities (as well as incremental rewards) have to vary with time. By combining Markov tree structures with tables (see Chapters 25 and 26), probabilities and rewards can easily be varied from one cycle to the next.

Probability tables and expressions

► **To use a table lookup in a transition probability:**

▷ Open the 2-State Markov tree you created in Chapter 25, or select it from the Window menu.

▷ Change the probability for Stay Alive to the formula `tLive[_stage]`, and press ENTER or RETURN.



The table "tLive" is not defined. Do you want to create a table with that name?

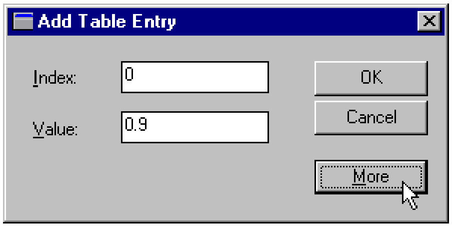
▷ DATA will ask if you want to create a table with the name tLive; click Yes.

A blank table called tLive (both file name and internal name) has been created. DATA will look up probability values in this table using the expression in the table reference brackets, _stage. The use of the _stage keyword in the reference means that probability values can change over time, as appropriate.

Before you can evaluate the Markov model using the new probability expression, you must enter pairs of indexes and values into the referenced table.

Index	Value
0	0.9
2	0.6
7	0.5
20	0.6

► **To enter values in a table:**



Add Table Entry

Index: 0

Value: 0.9

▷ Click on the Values pop-up button in the Define Values dialog, and select Default for Tree.

▷ Choose Table > Add Entry... to create an entry for index `0`. Using the More button, repeat this step for each entry.

▷ Choose File > Close to return to the tree.

Now, in place of the fixed 0.9 transition probability on the Stay alive branch, DATA will evaluate the formula tLive[_stage] at each new cycle. The first transitions will use the 0.9 table value; subsequent transitions will use lower values.

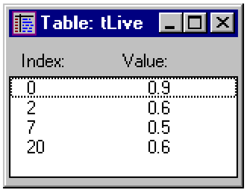
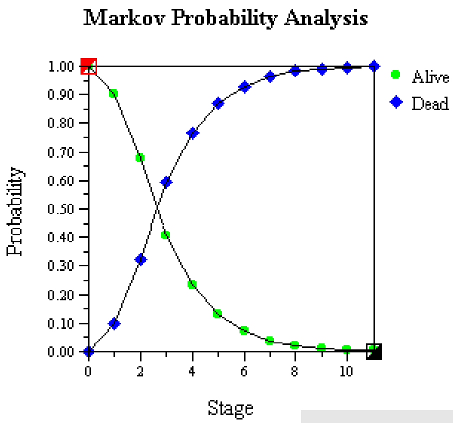


Table: tLive

Index:	Value:
0	0.9
2	0.6
7	0.5
20	0.6

The lookup method for new tables is, by default, interpolation. When you created the tLive table, its lookup method property was not changed (although this can be done at any time – see Chapter 26). Thus, the missing value for cycle 1 will be calculated using linear interpolation between the probability values for indexes 0 and 1. In cycle 1, for instance, the calculated probability used for Stay Alive will be 0.6 plus half of the difference between 0.9 and 0.6, or 0.75. Other missing indexes will be similarly calculated.



Run a Markov analysis, and compare the new state probabilities graph with a graph generated using the constant transition probabilities. You will see that the graph is no longer exponential due to the transition probabilities being varied over time.

Tables are the most flexible way of implementing time-varying probabilities or rewards. You may also use a formula which includes the _stage keyword, such as 0.9^(_stage+1). In addition to _stage, other keywords and variables can be used in transition probability expressions.

TIP: DATA does not restrict you to using the _stage keyword for a table lookup value. Any valid expression can be used (e.g. tLive[5] or tLive[StartAge + (_stage/12)]). While the use of tables is not restricted to defining rewards and probabilities in Markov models, this is their primary function.

Incrementing the value of _stage

The initialization pass occurs at cycle number 0. The order of operations during initialization is as follows:

- the cohort is distributed according to the initial probabilities entered for the Markov states;
- initial rewards are accumulated;
- the first transitions occur; and
- the termination condition is checked.

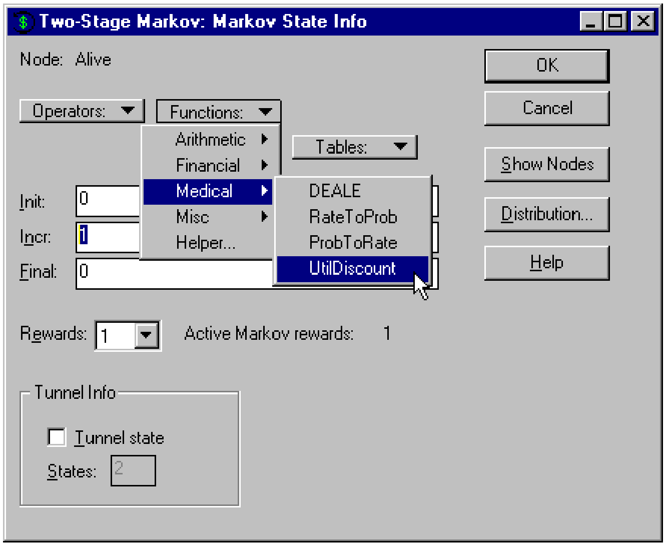
Only then (after the first set of transitions) is the value of _stage incremented to 1.

You should ensure that tables of stage-dependent values will be correctly referenced at cycle 0. If a table of transition probabilities does not include an entry with index 0, your table reference must take this into account. For example, your probability table may be an age-specific mortality table with an initial index of 40, corresponding to the starting age of the cohort. If your model has a one-year cycle length, you would use the expression tDie[40+_stage] as your table lookup.

Discounting rewards

Tables are also valuable for describing cycle-dependent rewards, such as costs or patient utilities. However, if all you need is to be able to discount the costs or utilities, this can be accomplished with a simple exponential formula rather than building a table. An expression – for example, Reward / (1+rate) ^ _stage – can be used to discount the value at each stage. You must ensure that the rate matches the cycle length. For example, if you have a yearly rate variable but are looking at monthly stages, you must use _stage/12 to adjust the formula for monthly discounting.

Rather than typing in the formula, you can use DATA's built-in discounting function, UtilDiscount(). This function takes three parameters: utility, rate, and time. It will perform the same calculation described above. Despite its name, it may be used for discounting values other than utilities, such as costs.



Two-Stage Markov: Markov State Info

Node: Alive

Init: 0

Incr: 1

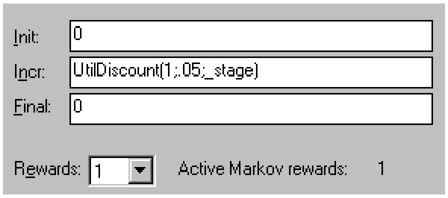
Final: 0

Rewards: 1 Active Markov rewards: 1

☐ Tunnel state

States: 2

See Appendix C for information on using DATA's built-in functions, including UtilDiscount; DATA's function Helper is covered in Chapter 9.



Init: 0

Incr: UtilDiscount(1;.05;_stage)

Final: 0

Rewards: 1 Active Markov rewards: 1

Initial and final rewards

Incremental state rewards are accumulated starting at cycle one and at each subsequent cycle. There are a number of situations which may require, instead, a onetime state reward.

Each state in the model can be assigned an initial and final reward. The initial reward value will be given to an individual or a portion of the cohort in that state at cycle zero, after the initial distribution and prior to the first transitions. The final reward is accumulated at the last cycle, after the termination condition is found to be true. To implement onetime costs or utilities in other cycles, transition rewards, discussed below, can be used.

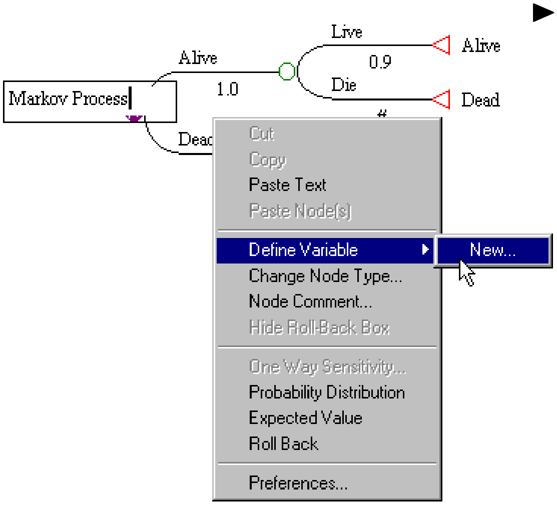
TIP: Initial state rewards are received only at cycle zero; from cycle one onward, DATA ignores all initial reward expressions.

Prior costs

Sometimes it is necessary to account for costs (or other values) accumulated prior to the Markov process. Consider, for example, a decision tree which represents two treatment alternatives. Both options include multiple uncertainties prior to reaching the Markov subtree, and various costs are dependent on the outcomes of these uncertainties. If this were represented using standard tree structure, costs would be incorporated into a payoff formula and incurred at terminal nodes. In a Markov model, costs must, instead, be incorporated into Markov state rewards.

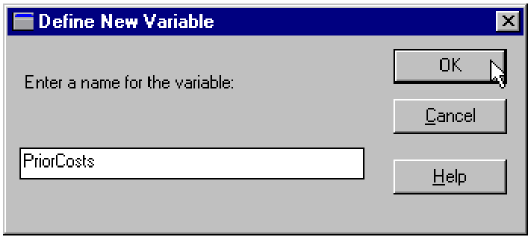
You must enter prior value expressions in the initial reward formulas of all states with a nonzero initial probability. Remember to include them in the appropriate reward set, if your model uses more than one attribute. (Multi-attribute Markov models are discussed later in this chapter.)

The following is one example of how you might go about including prior values in a Markov model.



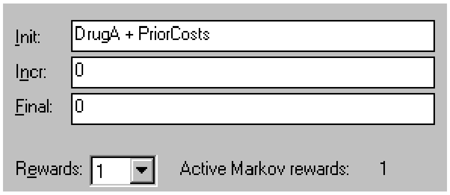
► **To include prior costs in a Markov model:**

▷ Create a variable called `PriorCosts`, and define it at the Markov node; here, the Quick Menu (see Chapter 14) is used with the Markov node selected.



Enter a name for the variable: PriorCosts

▷ Include in the definition of PriorCosts a summation of all costs accumulated before the Markov process.



Init: DrugA + PriorCosts
Incr: 0
Final: 0
Rewards: 1 Active Markov rewards: 1

It is important to bear in mind that DATA will not automatically include in its calculations any variables which you create; you must explicitly enter them in an appropriate formula, such as a payoff formula or state reward.

▷ For each state with a nonzero initial probability, update the initial reward expression (in the appropriate reward set) to use PriorCosts. If the initial reward is zero, it should be changed to `PriorCosts`; otherwise, append the expression `+ PriorCosts` to the existing initial reward.

Half-cycle correction

Markov models, by definition, occur as a sequence of snapshots. In contrast, real-world problems occur in continuous time. Transitions in a Markov simulation occur at the end of a cycle of fixed length; in reality, changes of state occur throughout the course of a process or problem. These approximations can lead to calculation errors which, depending on the particular situation, may be significant. Costs and utilities may be underestimated in the model, especially when quantities are discounted over time.

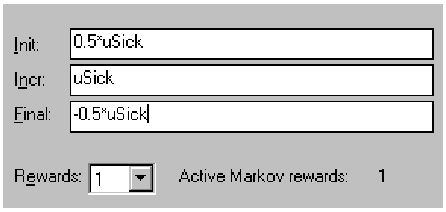
The *half-cycle correction* is an straightforward way to improve modeling approximations. Instead of simulating transitions at the *end* of the cycle, the half-cycle correction simulates transitions at the *midpoint* of the cycle.

To implement the half-cycle correction, you should include an initial reward that is half of the incremental reward accumulated during cycle one. This applies to all states with nonzero initial probabilities and incremental rewards. To complete the half-cycle shift to the left, you should also subtract from all states' final rewards one-half of the incremental reward accumulated during the last cycle. If your model runs until a high percentage of the cohort has entered an absorbing state, and no costs or other rewards are accumulated in this state, a final reward will have minimal impact.

Here is an example of how to include the half-cycle correction in your models. You must perform these actions for each state, and for each reward set (attribute) used at each state. You need not make any changes to transition rewards.

► **To account for the half-cycle correction:**

- ▷ Open the Markov State Information dialog for each state which will accumulate an incremental reward at cycle one.
- ▷ Select the text entered in the incremental reward editor, and choose Edit > Copy.
- ▷ Paste the incremental reward value or expression into the initial reward, and multiply by `0.5`. Use parentheses to separate the new expression from any existing formula.
- ▷ Paste the incremental reward value or expression into the final reward, and multiply by `-0.5`. Use parentheses to separate the new expression from any existing formula.



Init: 0.5*uSick
Incr: uSick
Final: -0.5*uSick
Rewards: 1
Active Markov rewards: 1

If the incremental reward expression at that state is stage-dependent, you may need to make slight changes to the copied formulas. If the _stage variable is used, change the expression to use 1 (or _stage+1) in the initial reward, and _stage-1 in the final reward

The table below shows the values for one sample Markov state before and after the half-cycle correction is applied.

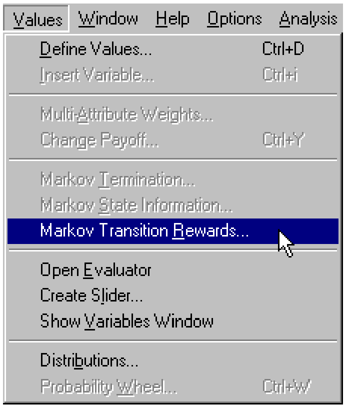
	Before	After
Init:	0	0.5 * uSick
Incr:	uSick	uSick
Final:	0	-0.5 * uSick

Basing your correction on the value of a half-cycle may not always provide the best approximation of reality, as it assumes that transitions occur at precisely the midpoint of a cycle. If, instead, you are modeling a situation where most of the transitions occur early in the cycle, it would be appropriate to modify the correction factor.

In the absence of such special considerations, the half-cycle correction should probably be employed in all Markov models using incremental state rewards.

Markov transition rewards

In some models, you may need to account for a reward which does not naturally fit into a state reward. *Transition rewards* make its possible to implement a onetime reward associated with an event which is not modeled as a state. (Events are nodes to the right of the states, modeling transitions.)



For instance, a onetime cost of admission may be associated with a transition from an outpatient state to an inpatient state; this cost is not incremental, and should not be accumulated in each interval spent as an outpatient or as an inpatient. Similarly, a cost might be incurred due to an event that is modeled in a transition subtree, but that does not lead to a different state. For example, a cost-effectiveness model, whose states represent only varying degrees of health, might include the possibility of malpractice and associated settlement costs. The uncertain litigation events could be represented using chance nodes within a particular state subtree, while not having an impact on future state transitions.

Another way to view transition rewards is as *time-independent* rewards; because the associated event is not part of the enumeration of states, it does not take any time in the model (though it may take time in reality). State rewards are *time-dependent*, because the exact value depends on the cycle length. If the cycle length changes from one year to one month, then the cost per cycle must go down. On the other hand, the cost for an *event* does not depend on the length of the cycle.

Transition rewards *cannot* be assigned at the Markov node or its branches (Markov states), but can be associated with any other nodes to the right of the Markov state nodes. As their name suggests, transition rewards are typically associated with transition nodes.

> **TIP:** What DATA calls transition rewards are called "tolls" in some other software packages. However, transition rewards are *added* to the net reward, whereas tolls are *subtracted*. Transition rewards should be entered using the same sign (positive or negative) as comparable state rewards: if you are tracking costs as positive numbers, your transition costs should also be positive.

► **To assign a transition reward,**

▷ Select the node at which the reward applies, and select Values > Markov Transition Rewards.

▷ Select the appropriate reward set and enter an expression for the transition cost. Leave unused transition rewards empty, rather than entering 0. Press ENTER or RETURN.

When the Markov subtree is evaluated, transition rewards are assigned after incremental rewards and before a new cycle begins. Thus, transition rewards that occur following the initial distribution of the Markov cohort, during the first set of transitions, are associated with cycle 0 and the starting state. In all subsequent cycles, accumulated transition rewards are associated with the *from* cycle, but the *to* state. In Markov analysis results, transition rewards are not accounted for separately from the associated state reward.

> **TIP:** In DATA 3.0, transition rewards were allowed only at transition nodes. You may now assign transition rewards at any non-state node in a Markov process.

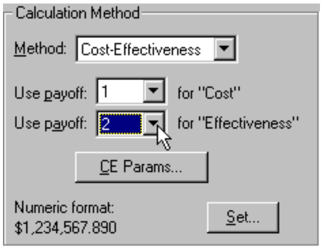
Cost-effectiveness Markov models

Building a cost-effectiveness Markov model is quite similar to working with a cost-effectiveness decision tree (see Chapters 19–21). First you must set the appropriate preference settings, including calculation method and numeric formatting. Then, separate sets of values (rewards) must be entered for the cost and effectiveness attributes.

Additionally, a cost-effectiveness Markov model requires that you enter a distinct termination condition for cost-effectiveness calculations. In a given tree, DATA maintains separate termination conditions for *each* single attribute, and *each* type of multi-attribute calculation, including cost-effectiveness.

► **To set up a cost-effectiveness Markov model:**

▷ Choose Edit > Preferences, and set the tree preferences for cost-effectiveness calculations, noting which payoff value is used for cost and which for effectiveness.

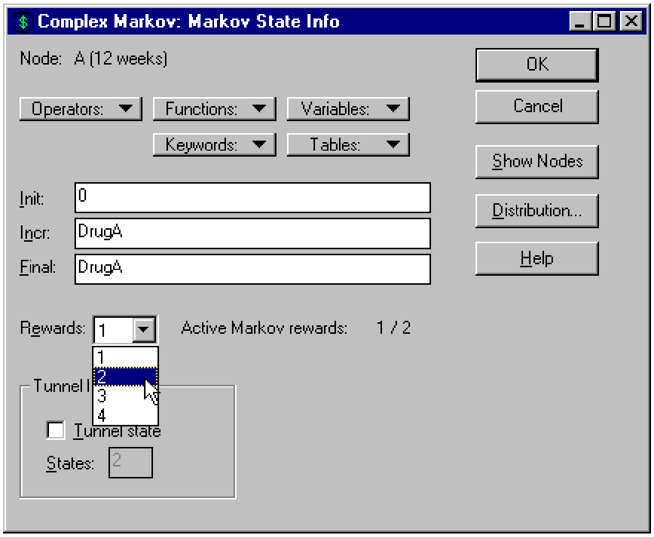


Method: Cost-Effectiveness
Use payoff: 1 for "Cost"
Use payoff: 2 for "Effectiveness"
Numeric format: $1,234,567.890

It is helpful to view a Markov node (and its subtree) as an evolved terminal node. In the Markov model, an attribute is referred to as a *reward set*, instead of a payoff; and, just as each terminal node in a decision tree can have up to four payoff expressions, each state in a Markov model can have up to four reward sets.

A reward set is comprised of the three reward types – initial, incremental and final – for a single attribute. The three reward values or expressions must be entered for all Markov states, and all active reward sets – for cost-effectiveness calculations, normally sets one and two.

▷ Select each state, one at a time, and choose Values > Markov State Information to open the Markov State Information dialog.


Complex Markov: Markov State Info

Node: A (12 weeks)

Init: 0

Incr: DrugA

Final: DrugA

Rewards: 1

1
2
3
4

Active Markov rewards: 1 / 2

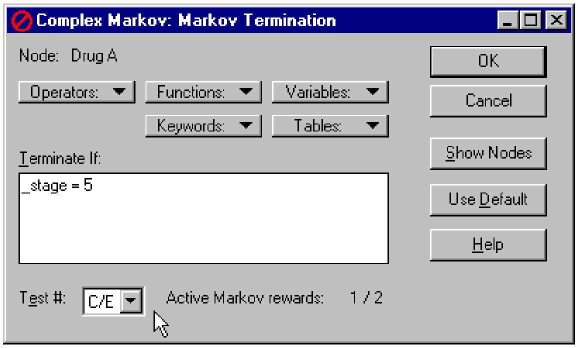
Tunnel I

☐ Tunnel state

States: 2

Below the three state reward text editors are two crucial pieces of information. The pop-up menu indicates which reward set is currently being edited in the text boxes. To switch from one set to another, simply select the set number from the pop-up menu. (This does not change the tree calculation method, which can only be changed from the Preferences dialog.) Next to the pop-up menu is a text indication of which rewards are active in tree calculations – “1/2” indicates that reward set 1 will be used for calculating the cost numerator, and reward set 2 for the effectiveness denominator.

▷ Assign the state reward expressions for both the cost and effectiveness sets, and press ENTER or RETURN.


Complex Markov: Markov Termination

Node: Drug A

Terminate If:

_stage = 5

Test #: C/E

Active Markov rewards: 1 / 2

After assigning the Markov rewards, you must enter the unique termination condition for use during cost-effectiveness calculations.

▷ Select the Markov node, and choose Values > Markov Termination.

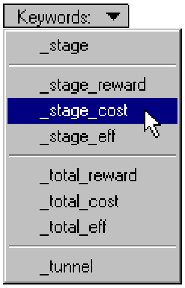
▷ Enter the termination condition in the dialog box, and click OK to close the dialog.

If DATA finds termination conditions for the individual attributes, and they are exactly duplicated, DATA will enter this as the default cost-effectiveness termination condition.

TIP: In DATA 3.0, cost-effectiveness Markov models were calculated in two passes; one for cost, and one for effectiveness. DATA 3.5 eliminates this redundancy. Now, instead of using two separate termination conditions, only one is necessary. When DATA 3.5 opens a cost-effectiveness Markov model created in DATA 3.0, it will attempt to build a unified termination condition. If the 3.0 termination conditions for cost and effectiveness were identical, the same condition will be used as the new, cost-effectiveness termination condition. If they are not identical, no cost-effectiveness termination condition will automatically be created for your tree. The Markov process will not calculate under cost-effectiveness until you enter the new condition.

Cost-effectiveness keywords

There are several keywords which are available only in a cost-effectiveness Markov model, and may be useful in a cost-effectiveness termination condition. The keywords ***_stage_cost***, ***_stage_eff***, ***_total_cost***, and ***_total_eff*** calculate the single-attribute values; _stage_reward and _total_reward are less meaningful, in this case, as they refer to the ratio values.



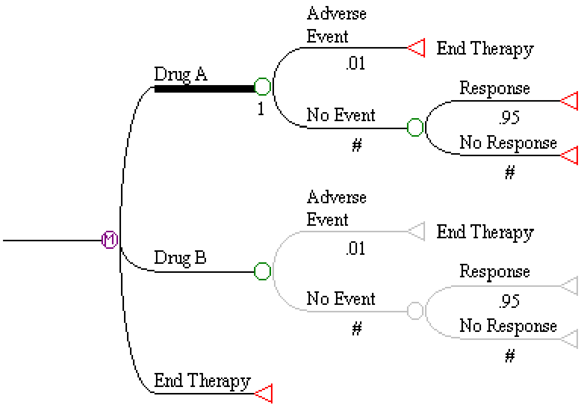
Cloning Markov subtrees

A major advantage of using clones in any tree is the ability to maintain a single, uniform structure across multiple subtrees, while having probability and other value expressions that can calculate differently in each subtree (see Chapter 12). In Markov models, a feature called Markov bindings allows transition states, as well as values, to vary among clone copies.

Markov bindings

Markov bindings function much like a standard variable in a clone; rather than substituting for numeric values, state bindings stand in for Markov state names at transition nodes. Markov bindings can be redefined at the root of cloned subtrees, either a Markov node or a node to its right, allowing unique transitions to occur in clone copies.

Look at the Markov Bindings tree. Currently the tree is in an unfinished state; transitions must be assigned to the Response and No Response

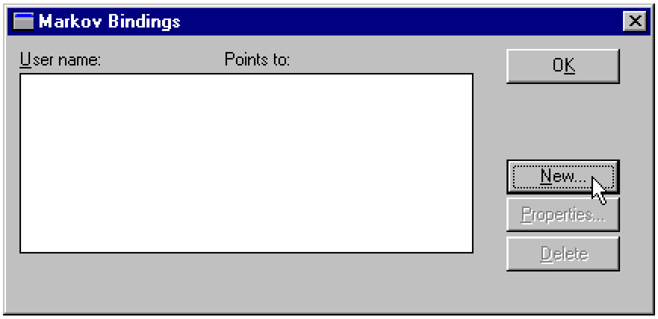


nodes in the clone master. Without Markov bindings, clones could not be used with this tree. Using Markov bindings, though, it will be possible to have the Response transition node in Drug A's subtree jump back to Drug A, while the same node in Drug B's clone copy subtree jumps, instead, to Drug B. Similarly, Drug A's No Response node could be pointed to Drug B, while Drug B's No Response node could point to End Therapy. To do this, you will create Markov bindings.

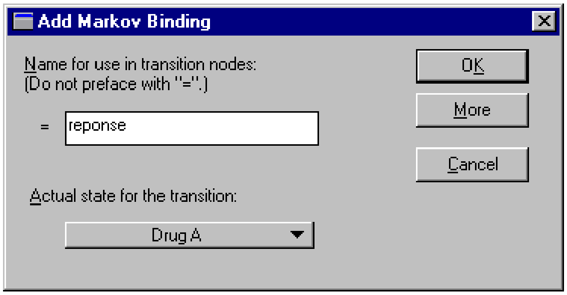
▶ To create a Markov binding:

▷ Open or activate the Markov State Bindings tree.

▷ Select the Drug A node, choose Options > Markov Bindings, and press New to create a new binding.



▷ Type the name `response` in the text box, and select Drug A from the pop-up menu list of states. Press More, and create a second binding; call it `no response` and point it to Drug B.



Name for use in transition nodes:
(Do not preface with "=".)

= reponse

Actual state for the transition:

Drug A

- ▷ At the Drug B node, also create two bindings called `response` and `no response`; point these to Drug B and End Therapy, respectively.

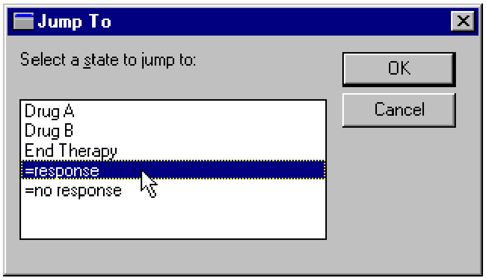
If Markov information is not currently visible directly in the tree, it can be turned on in the Preferences dialog, under the Variables Display page. Markov state bindings will then be displayed below other Markov information, in the form *Response >> Drug A*.

Bindings will have no effect until used at a Markov transition node, in this case, in the transition nodes of the clone master. Selecting the Markov binding names in the Jump To dialog, rather than the name of a specific state, will cause the bindings created above to be interpreted differently in the clone master and clone copy subtrees.

► **To use a Markov binding:**

- ▷ Select the Response branch of Drug A (the clone master), and choose Options > Markov Transition Node.

The Jump To dialog appears, showing a list of states and state bindings. The binding names displayed in the Jump To list are prefixed with the equal sign (=) to distinguish them from actual states. When naming states, you should avoid using a leading =, although this is not strictly forbidden.

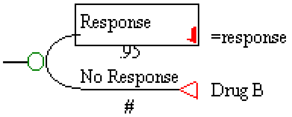


Jump To

Select a state to jump to:

Drug A
Drug B
End Therapy
=response
=no response

- ▷ Select =response as the transition for Response, the currently selected branch, and Press OK.



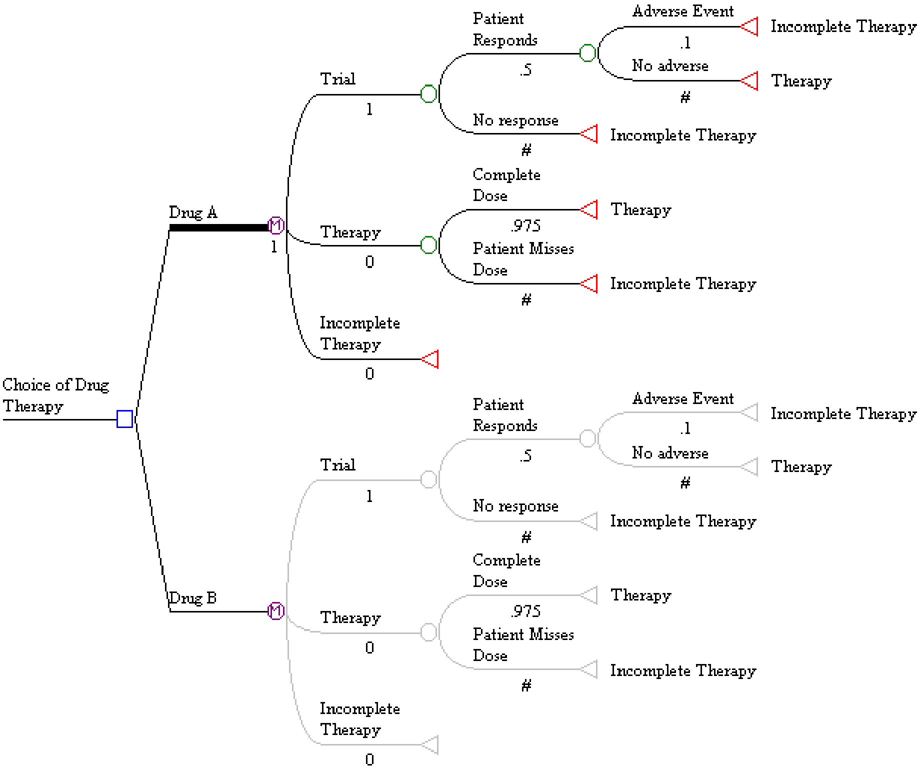
- ▷ Now, select the No Response branch of Drug A, choose Options > Markov Transitions… to open the Jump To dialog again, and select the binding =no response. Press OK to close the dialog.

Markov bindings do not have to be created at Markov states; they can be created on any branch to the right of the Markov node, or at the

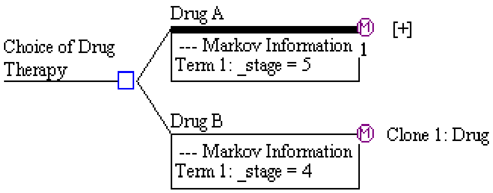
Markov node itself. When a Markov binding is needed at a transition node, the search for the binding proceeds in right-to-left fashion, as with variables.

Cloning an entire Markov process

Consider the Complex Markov model, shown at the end of Chapter 25. A similar model, built using clones, is shown below:

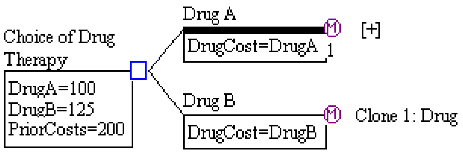


In this tree, Drug B's transition subtree is a clone copy of Drug A's subtree. On the surface, these subtrees appear to be identical; in fact, the strategies have different termination conditions (assigned at the Markov node, *outside* the clone master).



--- Markov Information
Term 1: _stage = 5

--- Markov Information
Term 1: _stage = 4



Also, incremental costs will calculate differently. In state rewards, a variable, DrugCost, is used. This variable is defined twice (outside of the clone, at each Markov node) providing a different incremental cost in each Markov process.

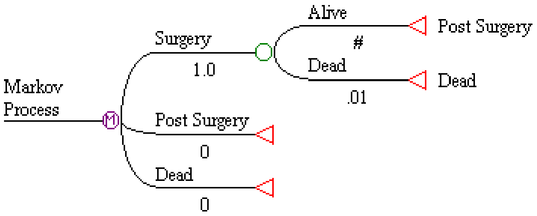
Similarly, different values could be used for each subtree's probabilities simply by converting the numeric probabilities to variables in the clone master, and then uniquely defining the variables at both the Drug A and Drug B Markov nodes.

Tunnel states

A *tunnel* is a concept describing an event which requires more than one cycle to unfold. This concept may also present itself in your Markov model as a series of similar events or states. DATA offers a useful, compact notation for creating tunnels. Before addressing the shorthand, it is important to first explain the concept of a temporary state.

Temporary states

Consider the basic Markov model shown here:

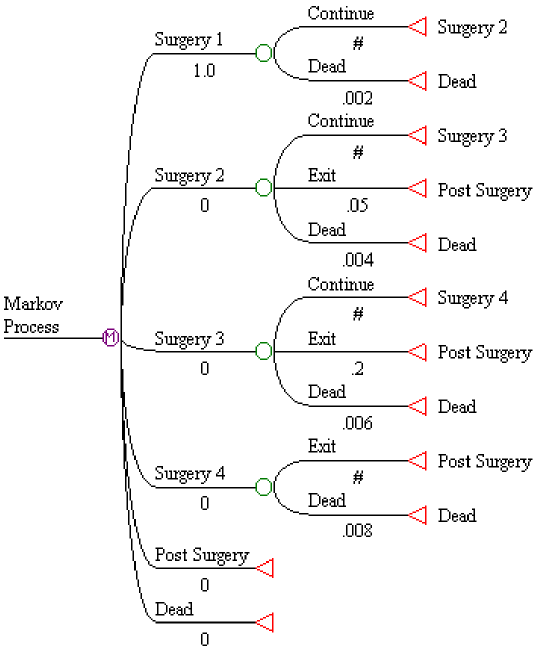


Because there is no path from Surgery back to itself, the patient must spend exactly one cycle in that state, and then exit to either Post Surgery or Dead. Because Surgery is limited to a single cycle, it is referred to as a *temporary state*.

Now, to include more detail, a *chain* of temporary states can be created: Surgery 1, Surgery 2, Surgery 3 and Surgery 4. While the patient might leave surgery prior to Surgery 4, the patient must *begin* at Surgery 1. This chain of temporary states is called a *tunnel*.

One might view the group of Surgery states as a single event which takes up to four cycles. At each cycle, there is a chance of continuing through the tunnel to the next Surgery node, of exiting the tunnel to Alive, or of exiting the tunnel to Dead.

Here is a graphic of what the new Surgery model, including temporary states, might look like (without the benefit of DATA's tunnel states):



This kind of chain should be distinguished from a state which simply transitions back onto itself for three cycles. In a cohort simulation, a *recycling* state cannot distinguish among individuals. It treats all its members exactly alike, whether they have just entered the state or have been in it for many cycles.

The temporary state concept is useful; in the Surgery state example, it is crucial that one cycle of surgery be distinguished from another. For instance, the transition probabilities for patients in Surgery 1 are different than in Surgery 2, and so on. Likewise, the incremental and transition rewards (costs and utilities) will likely be different for each of the four Surgery states. Moreover, it is very convenient to be able to model a single multiple-cycle event as a tunnel.

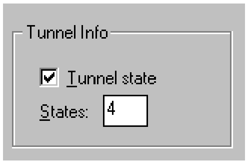
Creating a tunnel state

Up to this point, the discussion has been purely conceptual. Nothing about temporary states requires any special software feature; they are merely a helpful way of conceptualizing a common situation. In real-world models, though, building a set of temporary states creates a model which grows exponentially in size, because of the duplicated transition subtrees. DATA's implementation of *tunnel states* offers a shorthand notation which visually represents the group of temporary states as a single event, or state.

▷ Open the Markov Tunnel tree.

The state called Surgery appears as a single state; we would like it to represent a series of temporary states. The first step is to indicate that the Surgery state is actually a tunnel which visually represents a specific number of internal, temporary states.

▷ Select the Surgery state and choose Values > Markov State Information.



☑ Tunnel state

States: 4

▷ Click Tunnel state at the bottom of the window. DATA will then know to expand Surgery internally to several temporary states during calculations.

▷ Enter 4 for the number of states.

TIP: If, for some reason, the exact number of temporary states to be needed is difficult to calculate (or is dynamic), it is better to err on the side of excess, initially. For instance, setting the number of tunnels in the Markov Tunnel tree to 10 will have no adverse effects, other than to create unnecessary, empty temporary states which will require space in graphs and reports. Later, after analyzing the model, you can reduce the number to a more reasonable value and modify transition probabilities to ensure that all individuals exit in the last temporary state.

The tunnel transition subtree

Prior to building a tunnel subtree, the *superset* of all possible transitions out of the temporary states must be enumerated. Similarly, all excluded transitions should be noted. In the Markov Tunnel tree, there is a possibility of exiting the Surgery tunnel to Post-Surgery at temporary states 2, 3, and 4, while this transition should not be allowed at temporary state 1. That path, from Surgery to Post-Surgery, must appear in your list of possible transitions.

A unified transition subtree including all paths possible at any temporary state, will be attached to a single state node in this example, Surgery. It will represent the transition subtrees for all of the temporary Surgery states *which are not individually represented*. Transitions that are not possible at particular temporary states in the tunnel must be given a zero probability using methods discussed below.

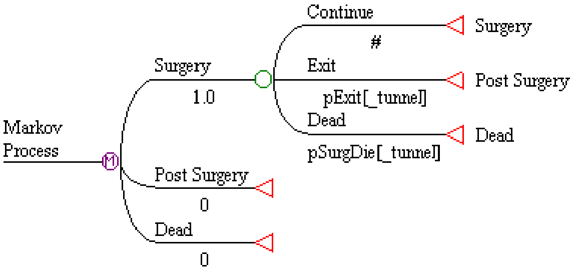
Using the _tunnel keyword

We need a way to indicate on the face of the unified subtree that values (in this case, probabilities) differ for each temporary state. In other words, the probability of surgical death is different at Surgery 1 than at Surgery 2, and so on. DATA includes the keyword _tunnel to facilitate this.

During calculations, DATA maintains an internal representation of each temporary state. (Looking at a Markov analysis text report for a tree with a tunnel state, you will see separate columns of state probabilities and rewards for each temporary state, in addition to the other Markov states.) For a particular instance, DATA sets the _tunnel keyword to the number of the temporary state being evaluated. For its internal Surgery 1 subtree, for example, DATA sets _tunnel to 1.

In the same way that a table can be defined and the _stage keyword (or an expression including it) used to retrieve a particular value from the table, it is possible to use the _tunnel as a table lookup value.

Here is the transition subtree after creating and populating two tables, pExit and pSurgDie:



All transitions are possible at any instance of the Surgery temporary state, except for two specific cases. The value of pExit[1] must be set to zero, to disallow a transition from Surgery 1 to Post-Surgery. Also, the value of pExit[4] + pSurgDie[4] must sum to one; otherwise, at Surgery 4, there would be some probability of a disallowed continuation through the tunnel, which ends at _tunnel=4. DATA will not provide an error message if this occurs; the portion of the cohort that continues into the undefined temporary state will be lost from calculations. You must ensure that your probability expressions and tables do not allow such errors.

Violating Markov strictures

One of the important assumptions about the cohort in a Markov model in its purest form is that it has no memory. Since an infinitely large cohort is used in probabilistic calculations, there are no actual cohort members to carry *memories* with them through the process.

This lack of absolute memory means that a given state's transition probabilities, rewards, and other values cannot depend on prior events (unless they can be directly inferred from being in that state at that time). All members of a given state at a given time are treated as full equals, with no regard given to the different paths they may have followed to get to that point. Information such as how many times a particular state has been visited, or what particular transition costs have been incurred, is not accessible during cohort simulation (roll back, Markov analysis, and other expected value calculations).

For example, the likelihood that an individual will experience a particular event in cycle n is (theoretically) independent of what happened in cycle n-1 or earlier. In the Markov cohort simulation, the only way to remember where a particular portion of the cohort has been is to create additional states and transitions (or use clones and tunnels) to keep separate those cohort members who experience different events. Even if this can be accomplished, it results in unwieldy models.

Monte Carlo simulation

Many real-world models will depend on some kind of memory. For example, consider a Markov model whose states represent multiple treatments. If one drug treatment has failed, that knowledge should be carried along with the patient (as in a patient record), to ensure that the same treatment will not be revisited. Another example involves the case of a tumor which grows unpredictably during the course of a model. The tumor's size is not itself a health state, but rather a piece of information which should follow the patient through the model. This information should be used to vary transition probabilities, or to determine treatment paths inside the Markov model.

Once individuals require memory, it is necessary to stop thinking in terms of the probabilistic cohort simulation. In the cohort simulation, individuals are abstract – part of an undifferentiated group that flows from state to state and is absorbed completely at each interval. There is no way to separate the current membership of a state into one group that just entered and another group that has been there since the outset. Similarly, there is no way to track a single individual through the process.

To give a Markov process memory, it must be analyzed via Monte Carlo simulation. In Monte Carlo simulation of a Markov model, one individual at a time is randomly stepped through the process; this can be repeated for a very large number of trials. Special tracker variables can then be used to track each individual's particular steps through the process.

For details on tracker variables, see Chapter 29.

It may be possible, although not recommended, to create Markov models with *variable cycle length*, for evaluation using Monte Carlo simulation. A model of this kind might use one cycle length to define probabilities and rewards for most states, while a different cycle length could be implemented for certain tunnel states. The possibility of error would be greatly increased, though, and the work involved may be prohibitive: for example, the _stage keyword would no longer be meaningful and separate tables of time- and cycle-dependent values would be required for each set of states.

Making decisions in a Markov process

Although the tree structure DATA uses to visually represent a Markov process looks similar to a standard decision tree, there are important differences. Viewed in the context of the decision tree in which it is embedded, a Markov node is simply an evolved terminal node; the Markov subtree acts like a payoff calculator. And, although Markov cycle trees are powerful and extremely flexible, not all aspects of a standard decision tree can be implemented in a Markov process.

In standard tree structures, a decision node makes its choice, or recommendation, by looking at the expected value of each alternative and selecting the optimal path. This is a very straightforward process: the subtrees rooted at the decision alternatives are each rolled (or folded) *back*. The work of a decision node is relatively simple, because the tree is evaluated by working backwards – calculating first the payoffs, and then the expected values at each prior uncertainty, working leftward until a decision point is reached. The labor of calculations is complete when the decision node must make its choice.

In contrast, a decision node cannot feasibly be embedded in a Markov model, because Markov models are calculated forwards, not backwards. In order to make a decision that looks to an uncertain future, the entire remaining process for the portion of the cohort would have to be evaluated multiple times, once for each alternative.

The recursive nature of Markov processes would introduce additional complications. The decision point might be encountered in subsequent cycles by new individuals. Recursive decisions might occur: before making a decision at cycle n, the same decision at cycle n+1 may have to be resolved, and so on.

Logic nodes and statements

DATA does make it possible to look to the *past* (or present) to make a decision in a Markov model, by evaluating logic statements. Tracker variables can be used to give the subjects of a model memory beyond a single cycle during a Monte Carlo simulation (see Chapter 29); the value of a tracker, representing past events, can then be used to force a particular choice among alternative paths. In cohort simulations, tracker variables are not available, but the value of a Markov keyword (e.g., _stage or _total_reward) might serve to force a path; during a single cycle, within a given state and its transition subtree, standard variables can also be used to force logical decisions.

It is also possible for the Markov process to look *outside* itself, to another decision tree or a spreadsheet, in order to select among alternative paths. Using DATA's linking features (see Chapters 15–16), the expected value of a node or the dynamically calculated value of a spreadsheet cell could be used to choose among available alternatives. It is possible to duplicate the Markov model entirely and use the linked value of the duplicate Markov node to help make a decision in the original.

A *logic node* is generally used in these situations. It acts like a decision node, in that it selects one path from its branches; rather than looking at expected values, it chooses a path by evaluating logical expressions. Each of its branches has an expression associated with it; starting at the top branch, the first node with an expression that evaluates to *true* is selected.

TIP: DATA also includes a conditional If() function, which can be used to evaluate a logical expression and, based on the result, return one of two expressions. See Appendix B for information on this and other DATA functions.

A simple logic node might have two branches, X and Y, with the expression "_stage > 4" below branch X and "_stage <=4" below branch Y. When the logic node is encountered, either branch X or branch Y is followed, based on the current value of _stage.

More likely, logical expressions will use a tracker variable. For example, a tracker variable may serve to remember, during a simulation trial, the number of detectable strokes an individual has experienced. A logic node could decide the transitions of the individual based on the current number of strokes.

See Chapter 34 for more on logic nodes.

CHAPTER 28 DISTRIBUTIONS

DATA includes numerous continuous and discrete distributions that can be randomly sampled during Monte Carlo simulation of your decision tree. In addition to their use in analysis, DATA's built-in distributions also can be used in model-building; if the branches of a chance node are a representation of values which are drawn from a distribution, DATA can calculate the probabilities and values for the branches.

This chapter provides general background on creating and referencing distributions in DATA. Appendix D includes details about each built-in distribution's required parameters, as well as its underlying formula and, where appropriate, domain. Chapter 26 provides important information on building tables for custom distributions.

TIP: The Dist() function used in DATA 2.6 is no longer supported. If you have models which use the Dist() function, you must convert them to use the newer DistSamp() function, in order for DATA 3.5 to calculate these models.

Monte Carlo simulation

Many of your estimates for model parameters will be point-values, but there are few true constants in the real world. Sensitivity analysis can often deal with the uncertainty inherent in estimating model parameters. The addition of variable correlations can help your sensitivity analyses take into account relationships between uncertain values. Finally, Markov models using cycle-dependent tables of values provide another means of making your model variables behave more dynamically during an analysis.

All of these techniques have limitations, though: even a three-way sensitivity analysis may not be a sufficient test of a very complex model's assumptions; variable correlations assume *linear* relationships between parameters; and in a Markov process, change occurs only in an orderly, predictable way.

To facilitate more realistic analysis of all models, large and small, DATA offers Monte Carlo simulation, making it possible to randomly sample model parameters from continuous and discrete distributions. The resulting benefit is that individual simulation trials, and even individual Markov stages, can use different values for specified model parameters. Any number of distributions can be defined in a model; it is possible to run a simulation with every parameter being randomly sampled from a distribution. And, similar to a standard sensitivity analysis, Monte Carlo simulation allows distributions to be correlated.

During all analyses other than Monte Carlo simulation, the value taken from a distribution will always be its mean (or median, where appropriate). It is only in Monte Carlo simulations that a distribution will be sampled. See Chapter 29 for more information on Monte Carlo simulation.

> **TIP:** It is important to avoid confusing the analytic, parameterized distributions covered in this chapter with the multi-branch probability distributions attached to each uncertainty in a tree. The distinction between the two types of distributions should become increasingly clear as you proceed through Chapter 28.

Creating a distribution

Distributions are employed in formulas (for defining variables, payoffs, or probabilities) by using the DistSamp() function. Based on the properties of the value being modeled, you create each specific distribution, including its parameters and a comment. These are stored in a list in the tree and assigned an index. This index is used as the single argument to the DistSamp() function, which can be referenced in formulas.

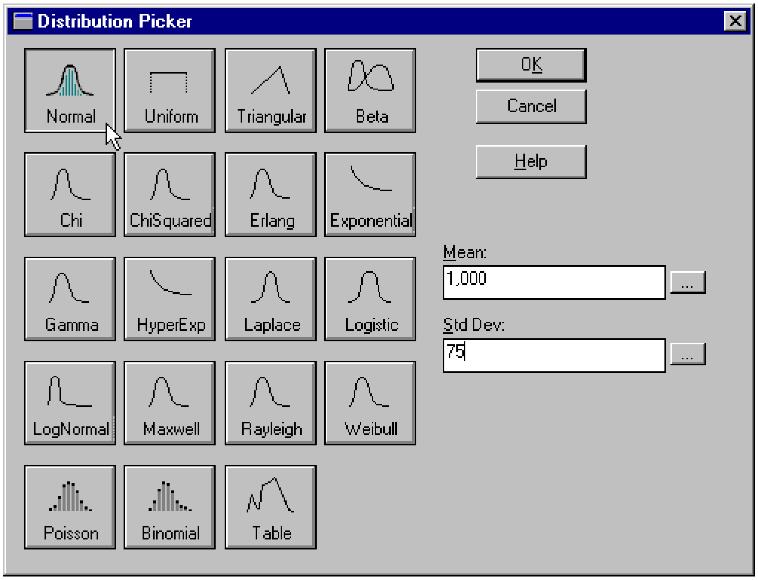
► **To create a normal distribution (mean = 1000, standard deviation = 75):**

▷ With a tree window open, choose Values > Distributions.

▷ In the Distribution dialog, press New… to create a new distribution.

▷ From the palette of distributions, click on the Normal button (if it is not already selected).

▷ For the parameters, enter `1,000`for the mean and `75`for the standard deviation.



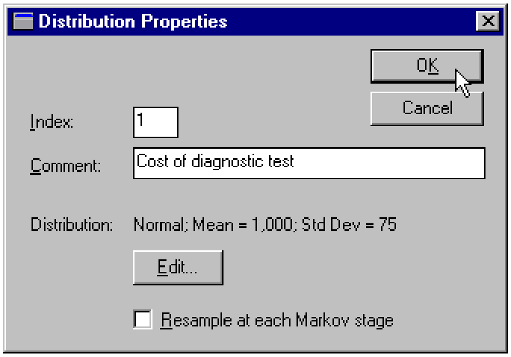
Mean: 1,000

Std Dev: 75

Clicking on the ellipsis button to the right of a parameter's text entry box will open an expression editor dialog. This makes it possible to use complex expressions, including variables and even other distributions, as distribution parameters. See Correlating Distributions, below, for important information on using non-numeric parameters.

▷ Press ENTER or RETURN to store the new distribution.

One last window appears, allowing you to change the index assigned by DATA or to enter a comment for your distribution. The index is a required piece of information; you will use it to reference the appropriate distribution in your model. The comment is optional, although it can become critical if you have many distributions to distinguish between.



Index: 1

Comment: Cost of diagnostic test

Distribution: Normal; Mean = 1,000; Std Dev = 75

☐ Resample at each Markov stage

Finally, there is a check box entitled Resample at each Markov stage. When checked, DATA will force the distribution to take a new value at each stage during Monte Carlo simulations of Markov processes, rather than taking a single value for the duration of a single trial. See below for an illustration of using distributions in a Markov process. For now, do not select the check box.

▷ Type `Cost of diagnostic test` for the comment.

▷ Press ENTER or RETURN to accept your changes, and press the Close button to close the Distributions dialog box.

Making changes to distributions

► **To make changes to an existing distribution:**

- ▷ In the tree window, choose Values > Distributions.
- ▷ In the Distribution dialog, make changes to the index, comment, and Markov resampling properties.
- ▷ To make changes to the distribution's parameter values and expressions, click on the Edit… button. Press ENTER or RETURN to accept your changes.
- ▷ Press ENTER or RETURN to close the Distribution dialog and accept your changes.

Using the DistSamp() function

DATA will not automatically include a distribution that you define in tree calculations; you must make explicit reference to the appropriate distribution in a formula. To do this, you will use the DistSamp() function. The DistSamp() function allows you to create a particular distribution once (see above), and then reference it any number of times in your model; the function takes a single parameter, the integer index of a valid distribution.

If, for example, you want to assign the distribution created above to a variable called TestCost:

- ▷ Create the TestCost variable, and enter it in a probability or payoff expression.
- ▷ Define the variable TestCost = DistSamp(*index*), where *index* must match the index assigned to your distribution.

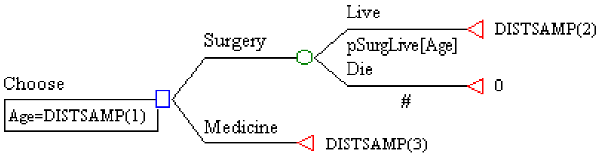
TIP: Rather than typing distribution references, you can quickly insert them into variable definitions, payoff formulas, and the parameters of other distributions by using the Distribution… button found in each window. Select the correct distribution from the list, or click New… to add another distribution. Then, in the Distribution dialog, click Use. The text "DistSamp(*n*)," with n being the index of the selected distribution, will automatically be inserted in your expression.

How DATA calculates distributions

Now, when you perform Monte Carlo simulations on your tree, you can choose to sample any or all of the distributions that you have created. All sampled distributions, without exception, will be evaluated at the root node of the tree. One sample is taken per trial, with the exception of distributions that you specify to resample during Markov simulations; in any given trial, these distributions will be sampled once per cycle.

Note that DATA *will* sample distributions that are not referenced directly or indirectly in the calculation of payoffs and probabilities. It is usually wise to delete unnecessary distributions, in order to speed up calculations, although it may be useful to see the sampled values of a correlated distribution in the simulation text report, even if it is not used in simulating the model.

During any expected value calculations (i.e., all analyses other than the second-order step of each Monte Carlo simulation trial), distributions will not be sampled. The mean value of the distribution will always be used in these cases. (For custom distributions, the median value is used; see below.)



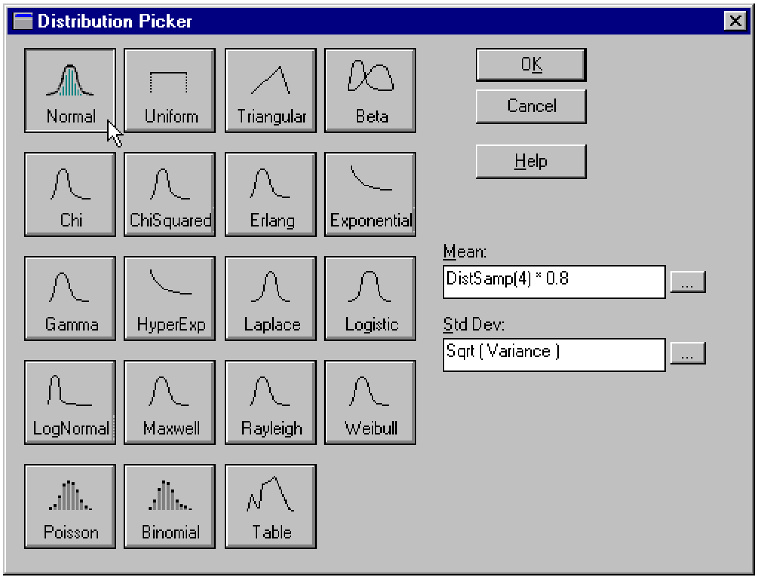
The model Distributions Tree, shown here, utilizes three distributions. The probability that a patient will survive surgery is a function of age. The first distribution, DistSamp(1), employs a random value generated during Monte Carlo simulation to specify the patient's age (during the current trial); this value is then used as the lookup value for the table entitled pSurgLive. The second and third distributions are used to represent the quality-adjusted life expectancy of patients who survive surgery versus those who are treated with medicine, respectively.

Open the Distributions Tree and try both rolling back the tree and performing a Monte Carlo simulation (at various nodes). See Chapter 29 for an explanation of the Monte Carlo simulation output.

There is no limitation on the number of distributions, each with different parameters, that can be used in a model. It is also possible to reference the same distribution at multiple locations in the same tree; this will ensure that the same sample value is used by DATA throughout each trial. To use distinct values during a single trial, parameters must reference different distributions.

Correlating distributions

A distribution can be used just like a variable. For example, a distribution can take the place of a variable in a probability or payoff expression, in a logical statement, or even in the parameters of another distribution.



Mean: DistSamp(4) * 0.8

Std Dev: Sqrt (Variance)

If you plan to use a distribution (or other non-numeric expression) as a parameter of another distribution, it will be important to take into account how distributions are evaluated during a simulation.

All distribution parameters will be evaluated at the root node. If there is no default definition for a variable used as a distribution parameter, an error will occur. The order in which distributions are sampled is based on index number, beginning with the distribution having the lowest index. To be able to use the output of one distribution as the input to another, the dependent distribution must have a higher index than the input distribution. Otherwise, the input's sample value from the previous trial (or the mean value, if there are no previous trials) will be used. A distribution may not reference itself as a parameter.

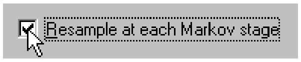
Resampling during Markov processes

Distributions can be used to specify any value used in dynamically evaluating a Markov process (with the exception of the number of temporary states associated with a tunnel). For instance, distributions can be sampled to provide values for:

- a termination condition for all trials, or specific to the current trial (e.g., a time period equal to the sampled life expectancy of an individual);
- a Markov reward that accumulates a sampled value (e.g., yearly treatment costs); or

- an initial or transition probability.

For some Markov models, it may not be sufficient to simply use a single sample value for each trial. Instead, it may be necessary to generate a new sample value at each successive stage during a single trial. With DATA, it is possible to use a new distribution sample at every stage of each trial.



☑ Resample at each Markov stage

Selecting the Resample at each Markov stage check box in the Distribution Properties dialog will cause a new sample to be taken at each incremental stage of a Markov process. Use this option if, for instance, a tumor grows stochastically during a Markov model. The tumor size can be modeled more accurately if the incremental growth is sampled from a distribution at each stage, rather than using a fixed increment throughout an individual's life.

Custom distributions

There are two ways to design custom distributions in DATA; both methods use a DistSamp() function and a table. See Chapter 26 for information on using value tables in DATA.

Creating a user-defined DistSamp() function

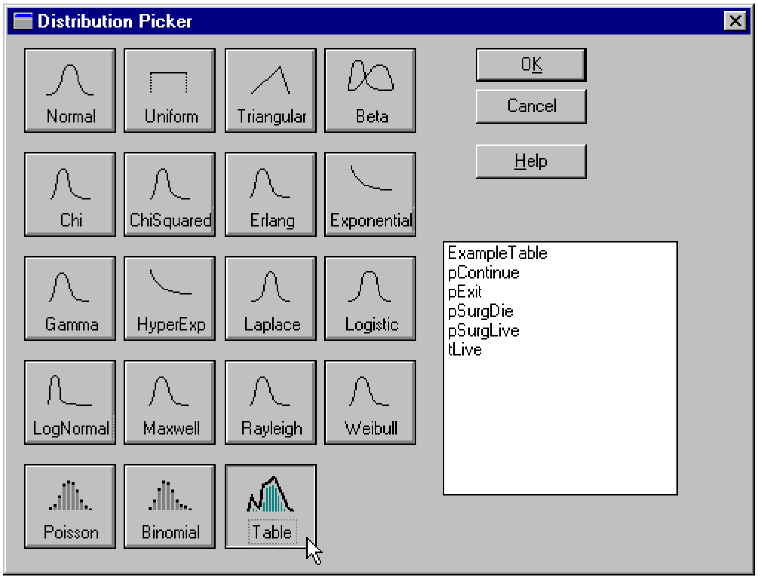
If none of DATA's built-in distributions describe the particular function you need, there is an easy way to sample values from a distribution of your own design: simply create a new table describing the distribution's probability function, and then create a new DistSamp() referencing the table.

Index	Value
800	0.2
1000	0.5
1200	0.3

If your distribution is continuous, you must discretize it into individual intervals for the table. Each interval has a height, or probability (entered as the value of a table entry), and an x-axis location (entered as the index of a table entry). For instance, you might create the table shown here for use as a distribution representing the cost of surgery.

As you can see, the surgery is most likely to cost $1000, with fixed outer limits of $800 and $1200; the probabilities of the three intervals total one. (This particular distribution might be represented, instead, using a three-branch chance node. With a large number of intervals, though, a custom distribution offers a clear advantage over chance node representation. For more discussion, see Distributions and Chance Nodes, below.)

Then, open the Distributions dialog, and click on the Table button. The only parameter required for this type of distribution is the name of the DATA table to generate samples from. Select the correct table, and close the Distributions dialog.



How DATA samples from a table

You bear responsibility for correctly designing the table. Be sure to enter the x-axis value of the custom distribution as the *index* of the table, and the associated probability as the *value* of the table. DATA will take samples from a table even if, for example, the probabilities specified in the table sum to more or less than 1.0.

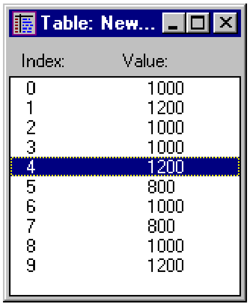
Sample values will only be drawn from exact table entry indexes, regardless of which lookup method you specify. To represent a continuous distribution more accurately, simply add more intervals to the table.

The median value of the custom distribution will always be used in non-Monte Carlo calculations. Note that neither the index-specific lookup method nor the index off edge is error option will generate an error during sampling.

Referencing a table using a uniform distribution

There may be situations where your distribution data will not fit the strictures of a user-defined, distribution, or where the data set is too large or dynamic to be represented in a DATA table. It may still be possible to set up a custom distribution for use during Monte Carlo calculations.

Perhaps you have a table of *observed* costs (similar to the cost distribution described above) for a particular procedure which you would like to sample.



Index:	Value:
0	1000
1	1200
2	1000
3	1000
4	1200
5	800
6	1000
7	800
8	1000
9	1200

To sample these values during a Monte Carlo simulation, first set up a new table as it appears at left. Change the table's properties so that the lookup method is Truncation, and ensure that the indexes increment regularly. Note that the values (costs) are now located in the right-hand, Value column; also, probability information is not included on each line, but is reflected in the frequency of a value's appearance in the entire table.

You will reference the values in the new table using a uniform distribution representing the range of table indices. Create a new, uniform distribution with the parameters low=0 and high=10. This is a continuous distribution – during a Monte Carlo simulation, DATA will randomly sample all numbers from 0 to 10, inclusive. Since the table lookup method is truncation, samples less than 1 will retrieve the first, zero-index value (1000), samples from 9 to 10 will retrieve the last value (1200), and so on.

The actual reference in a payoff expression should look like this:

MyTable[DistSamp(1)],

where MyTable is the custom distribution table and DistSamp(1) is the linear distribution of indexes.

Linking to Excel tables

If you have a complex table in an Excel spreadsheet which cannot be represented using a DATA table, you may be able to use DATA's bi-directional linking feature to look up values from the spreadsheet. See Chapter 16 for details on setting up bi-directional links.

Distributions and chance nodes

The analytical distributions described so far in this chapter are, of course, different from a probability distribution represented in a tree by the branches emanating from a chance node. However, if the uncertainty at that chance node is itself described in terms of a distribution, DATA's Distribute Children command can be employed to assign discrete branches, probability estimates, and value estimates at the chance node.

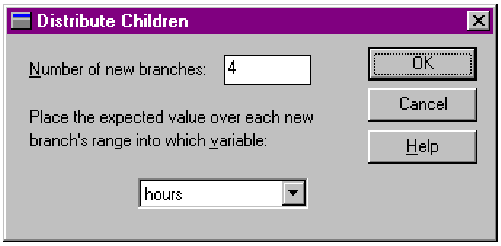
The Distribute Children feature is useful when you are modeling an uncertainty with potential values drawn from a continuous analytical distribution. For instance, if you are modeling the amount of time spent on a project as a normal distribution with mean 40 hours and standard deviation 6 hours, DATA will generate the branches, probabilities, and values associated with the specified distribution.

To use Distribute Children, you must have a chance node from which no branches emanate, and a variable which will hold the values of the distribution (in this case, the number of hours spent on the project).

► **To use Distribute Children to assign branches to a chance node:**

▷ Select a childless chance node, and select Options > Distribute Children.

▷ In the first dialog box, enter the number of branches you would like created, and select the variable which will hold the values of the distribution. Press OK to continue.



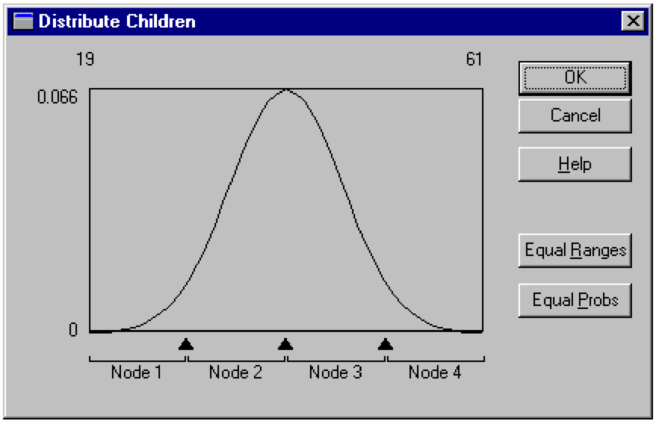
Distribute Children

Number of new branches: 4

Place the expected value over each new branch's range into which variable: hours

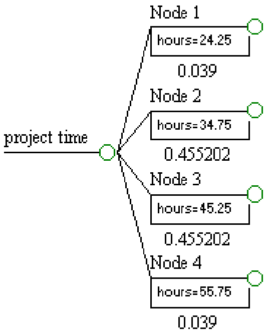
▷ Next, select a distribution and enter its parameters in the Distribution Picker dialog box. Press ENTER or RETURN.

▷ In the final dialog box, indicate how the branches should be distributed.



In most cases, you will probably want to assign each node an equal segment from the full range of the distribution. If you select Equal Ranges, or leave the default ranges, DATA will calculate the probability to be used for each branch. However, you may instead assign each node an equal probability. If you do so, DATA will calculate the value of the distribution for each node, as described below. You may also drag the handles which separate the nodes to create a custom allocation of the distribution.

▷ When you press ENTER to store your distribution, DATA will create the branches for you. At each branch, the probability and the value of the distribution will be stored.



DATA assesses the probability for each branch by finding the area beneath the curve over the node’s range. The value of the distribution variable at a given branch is the midpoint of that node’s range. In some circumstances, the probabilities of the distributed children may not add to 1.0 because of rounding errors, so a slight manual correction may be required to avoid an error message when you try to calculate the tree.

Custom distributions are not available with the Distribute Children command.

Chapter 29 Monte Carlo Simulation

When you roll back a tree, DATA identifies the optimal strategy on the basis of expected values (probabilistically weighted, long run averages). While these calculations will likely be a very important part of your analysis, in most cases your inquiry should not end there. Analysts often find that examining the probability distribution (or *risk profile*) of a model, and performing a variety of sensitivity analyses, gives them a better understanding of the set of real potential outcomes.

By incorporating Monte Carlo simulation, DATA significantly improves on the standard expected value analyses by integrating probabilistic distributions assigned at each chance node with value-oriented distributions assigned as a payoff (or probability) component. For example, here are some of the questions answered by a Monte Carlo simulation of a model:

- Based on expected values calculated for a model of an offshore drilling decision, the recommended strategy in your model is to attempt to extract from a gas field. For varying numbers of trials, what percentage of the time can I expect extraction to provide a *worse* outcome, based on profit?
- The probability distribution graph displays an uneven spread when cost values are estimated as point values. How does the spread differ when cost values are sampled from a distribution?

Many useful statistical analyses can be performed on the results of a Monte Carlo simulation. While DATA is not a statistics program, it is easy to export the results of a simulation into any popular statistics program for further analysis. Simple spreadsheet analysis of DATA's output can answer additional questions:

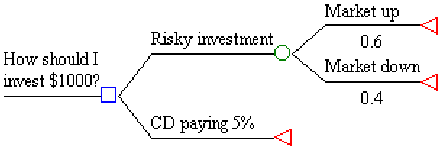
- For what percentage of patients will a particular surgical procedure offer at least six months additional survival over treatment with drugs?
- What is a likely distribution of the number of breakdowns that component X will experience before total failure?

To use Monte Carlo simulations effectively, you should be familiar with Chapter 28, Distributions.

First-order simulation

At its simplest level, a single simulation trial will randomly select a path at each uncertainty (chance node) based on the probability distribution of the outcomes. In a simulation of a tree or subtree composed only of chance nodes, each trial will select one end node by randomly following a single path through the tree. If the simulation includes one or more decision nodes, each choice will be made on the basis of expected value.

Consider the Stock Tree discussed in earlier chapters:



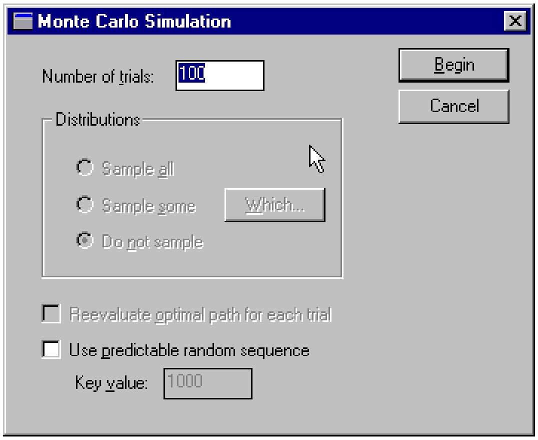
The analytical value of Monte Carlo simulation on this tree is somewhat limited, as it includes no distributions. However, it is useful to illustrate the process of first-order simulation. During a Monte Carlo simulation of the investment decision in Stock Tree, the following series of events will take place for each trial run:

- The decision at the root node is made on the basis of expected value. If the tree samples from distributions during the simulation, different sample values could be used in the expected value calculations required to make the initial decision for each trial. This is referred to as second-order uncertainty. No DistSamp() functions are used in the investment model; hence, the decision will be made once, at the first trial, using the point values in the tree. Every trial will choose the optimal Risky investment alternative.
- From the Risky investment chance node, each trial will pick a branch at random. Averaged over many trials, about 60% will select the Market up outcome, and the remaining 40% will select the Market down branch. Over just a few trials, the observed distribution may diverge from these expected outcomes.
- Whichever path is chosen, its terminal node's value is used as the final outcome value for that trial. In this case, about 60% of trials will result in a gain of $500, and about 40% will result

in a $600 loss. The payoff of the CD paying 5% node, $50, will never be an outcome in this simulation, because it is eliminated based on the initial expected value calculation at the root decision node.

► To perform a simple Monte Carlo simulation:

- ▷ Open the sample file Stock Tree.
- ▷ Select the root decision node.
- ▷ From the Analysis menu, choose Monte Carlo simulation, or click on the dice icon in the toolbar.
- ▷ Enter 100 for the number of trials. Ensure that all of the check boxes are cleared.
- ▷ Press ENTER or RETURN to begin the simulation.



Number of trials: 100

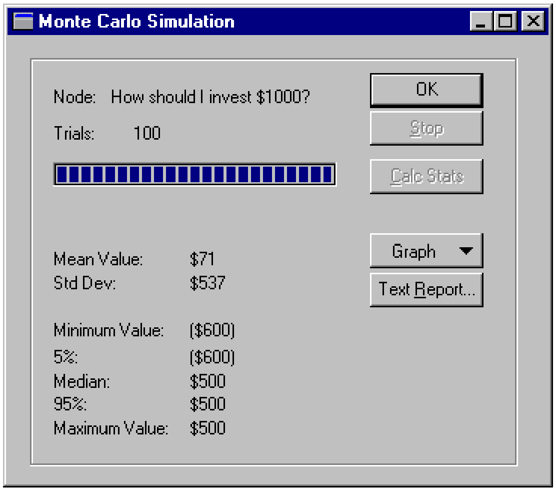
Distributions

○ Sample all
○ Sample some
● Do not sample

□ Reevaluate optimal path for each trial
□ Use predictable random sequence

Key value: 1000

A dialog box appears that displays the progress of the simulation and, after calculations are complete, a summary of the simulation's output. This basic statistical data can also be displayed during the progress of the simulation, if the model is very complex or you are running a very large number of trials. Simply click on the Calc Stats button, as often as you like, for a snapshot summary of the current statistics.



Node: How should I invest $1000?

Trials: 100

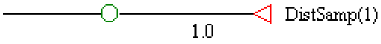
Mean Value:	$71
Std Dev:	$537
Minimum Value:	($600)
5%:	($600)
Median:	$500
95%:	$500
Maximum Value:	$500

If you click on the Graph button, DATA will present a histogram of all trial outcomes. The Text Report button will create a text listing of all outcomes sequentially, which can be copied to the clipboard or exported to a text file. Most spreadsheet, graphing, and statistics programs will be able to read the output from the text report. The text report is described later in this chapter.

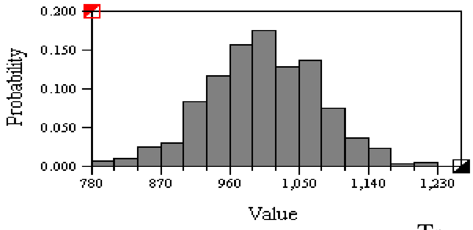
Second-order simulation

Monte Carlo simulations on decision tree models which do not use distributions (i.e., do not use the DistSamp() function) will rarely offer an advantage over the probability distribution graph. As you add more trials, the simulation will approximate more closely the distribution of outcomes generated by an expected value calculation.

The real value of simulating your model is in seeing how parameters which may vary from trial to trial affect the results. When you enter parameters as distributions, rather than as fixed point-estimates, you can force a new sample value to be taken for each trial. In this respect, simulation is similar to sensitivity analysis, which also generates information concerning the impact of changing certain values.



At left is a tree which demonstrates the results of sampling from distributions. What would otherwise be a trivial one-node tree has been expanded to two nodes because it is not possible to perform a Monte Carlo simulation at a terminal node.



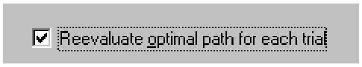
Open up the sample file Monte Carlo #1, which contains this two-node tree. Perform a Monte Carlo simulation at the root node, being sure to sample all distributions. The distribution graph illustrates the effects of sampling the distributions. In the graph shown at left, DistSamp(1) is a normal distribution with mean 1000 and standard deviation 100, using 500 trials.

Try using Monte Carlo #2 to see a more complex simulation of a similar situation. This tree has more than one distribution; you might try temporarily disabling the sampling of certain distributions, to see the effect on the output.

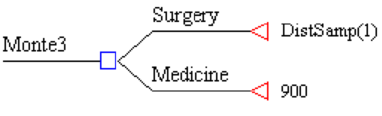
Decision nodes

Unless you specify otherwise, all trials will choose strategies which are recommended by expected value calculations. By default, these decisions will be made before distributions are sampled, and all trials will use the same decision strategy (or strategies, if a model includes more than one decision point).

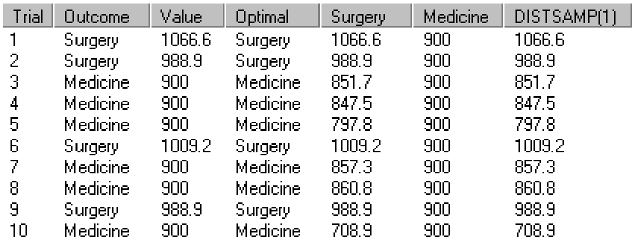
You can choose, however, to have DATA recalculate the optimal path separately for each trial, using new distribution samples to evaluate strategies. This may more closely simulate the ability of individual decision makers to account for their own situation before making strategy or policy choices, although it will take longer to calculate.


☑ Reevaluate optimal path for each trial

The check box in the Monte Carlo simulation dialog that controls the behavior of decision nodes is called Reevaluate optimal path for each trial.



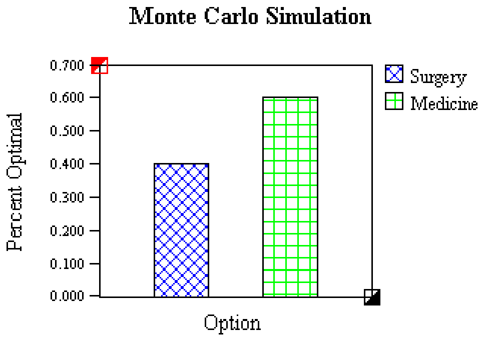
To see the effect of this option, open the sample file Monte Carlo #3 and perform a Monte Carlo simulation at the root node. If you leave the Reevaluate optimal path check box checked, the text report will look something like that on the following page.



Trial	Outcome	Value	Optimal	Surgery	Medicine	DISTSAMP(1)
1	Surgery	1066.6	Surgery	1066.6	900	1066.6
2	Surgery	988.9	Surgery	988.9	900	988.9
3	Medicine	900	Medicine	851.7	900	851.7
4	Medicine	900	Medicine	847.5	900	847.5
5	Medicine	900	Medicine	797.8	900	797.8
6	Surgery	1009.2	Surgery	1009.2	900	1009.2
7	Medicine	900	Medicine	857.3	900	857.3
8	Medicine	900	Medicine	860.8	900	860.8
9	Surgery	988.9	Surgery	988.9	900	988.9
10	Medicine	900	Medicine	708.9	900	708.9

The output of the text report is more fully described later in this chapter. In summary, the Optimal column specifies the strategy chosen during the specified trial (second-order simulation); the next two columns show the expected values for each strategy used to make the decision. The Outcome and Value columns display the name and payoff of the end node arrived at in the first-order simulation. In this example, the Outcome and Optimal columns show identical results because each scenario ends with the branches emanating from the decision node. The last column shows the sample value of the distribution.

Returning to the main Monte Carlo Simulation dialog, press the Graph button and select the Strategy Selection Frequency graph to display how often each option was selected.



Note that if the optimal path were not reevaluated for each trial, Surgery would always be optimal, based on the mean values of the DistSamp() functions.

Tracker variables

A tracker variable can have two basic functions in first-order Monte Carlo simulations, both as an output, or attribute, and as a structural element of a tree. The former is useful in evaluating simulations of almost any complex model; the latter is primarily a tool for simulations of Markov models.

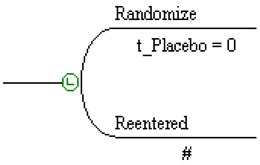
Trackers as outputs

As their name suggests, tracker variables provide a detailed memory of each trial of a simulation, beyond the standard Monte Carlo simulation output described above. The standard Monte Carlo report tells you essentially three things about a particular trial: its final value, where it started (the alternative chosen), and where it ended (the terminal node which represents the final outcome). By using tracker variables, you can create additional outputs, such as values indicating whether or not (or how many times) a particular event occurred.

It is possible, for example, to employ one set of tracker variables to indicate how often a patient has undergone a particular treatment during a Monte Carlo simulation, while another set of tracker variables is used to keep track of the current size, type, and location of a tumor. By using trackers in this fashion, it is possible to avoid having to create multiple health states in order to indicate, for example, the current size of the tumor.

There is no limit on the number of tracker variables that can be used in your tree; the Monte Carlo text report will include the value of each for every trial in the simulation.

Using trackers in model logic



The final values of all tracker variables in the tree will be part of the simulation output, but some trackers might also be used to determine transitions and rewards intelligently during a trial. For example, a logic node (see Chapter 33) could compare the value of the tumor size tracker variable to some threshold, and appropriately transition the individual to a new state. Or, the tumor location tracker variable could be used as a lookup value in retrieving an appropriate Markov reward from a table of surgical costs.

Structurally, tracker variables are useful primarily in simulating Markov models, where they can be used to implement recursive variable definitions. Tracker variables are not required for this purpose in simulating non-Markov tree structures, where regular variables can be given a recursive definition (e.g., x=x+1, each time a given procedure is done). However, recursive definitions of regular variables will not work in Markov subtrees, because their value is forgotten at the outset of each successive stage. In contrast, tracker variables, which have a single (global) definition throughout a model, can be utilized in simulating a Markov model. In a Markov Monte Carlo simulation, a tracker's value will be carried over from one stage to the next.

A caveat on trackers

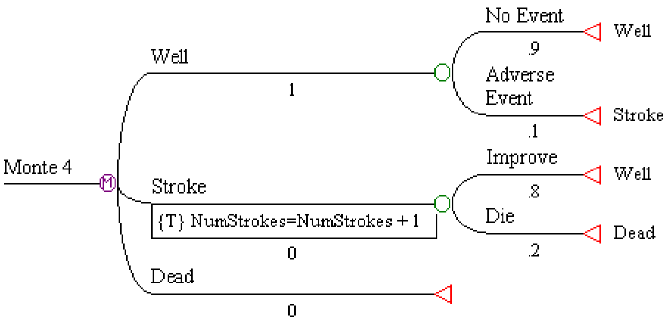
During expected value calculations, all tracker variables will have a value of zero. If your model uses tracker variables only as outputs, it should be safe to perform expected value calculations and related analyses on the tree. If, on the other hand, tracker variables have structural functions in your model, perhaps in logic statements or value assignments, you should not plan on using expected value-based calculations.

This caveat not only relates to analyses such as roll back, Markov, or sensitivity analysis, but also means that Monte Carlo simulations including decision nodes, which utilize expected value calculations, may not work correctly; see below for more information.

TIP: A tracker is the only type of variable that is automatically evaluated when encountered during *left-to-right* traversal of the decision tree. Normal variables referenced in a payoff or probability expression are evaluated only when needed for calculations, and then in a *right-to-left* traversal. Trackers have a zero value during all calculations other than the 1st-order portion of a Monte Carlo simulation.

A simple example with tracker variables

Open the file Monte Carlo #4. It contains a simple Markov process which illustrates the use of trackers. The tracker variable NumStrokes keeps count of the number of times a trial patient enters the Stroke state.



At the start of each trial, NumStrokes is set to zero (as specified in the variable Properties dialog box). When the trial enters the Stroke state, the tracker modification NumStrokes = NumStrokes + 1 is executed. (DATA does not wait for a calculation to reference NumStrokes to evaluate this expression, as with normal variables.) This changes the global value of the tracker.

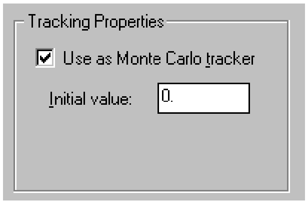
The text report output of a Monte Carlo simulation at the Markov node will look something like this:

```
Trial   Outcome     Value   NumStrokes

1       Monte 4     40.4    2
2       Monte 4     7.2     1
3       Monte 4     56.4    3
```

DATA will specify the final value of each tracker variable used in your model. The Value column refers to the overall value for the trial, in this case the accumulated Markov reward.

Setting up a tracker


Tracking Properties

☑ Use as Monte Carlo tracker

Initial value: 0.

► **To create a new Monte Carlo tracker variable:**

▷ From the Define Values dialog, choose New > Variable.

▷ In the Variable Properties dialog box, name the variable.

▷ Check the Use as Monte Carlo tracker box.

▷ Assign an initial value, which will be used at the start of each individual Monte Carlo trial.

When a tracker variable is created, it must be assigned an initialization value. At the start of each trial of a Monte Carlo simulation, tracker variables are initialized with this starting numeric value. For instance, you might start the size of a tumor at zero.

▷ Press ENTER or RETURN to close the Define Values dialog.

Making tracker modifications

When a tracker variable is displayed, it is prefixed with {T} to indicate that it is not a normal variable. You should not include this prefix when referring to the tracker in formulas.

Tracker variables do not have definitions in the normal sense. However, it is possible to assign a new value to a tracker by creating a recursive definition, such as Tracker_A = Tracker_A + 1. Since the expression modifies the global value of the tracker, definitions of tracker variables are called *tracker modifications*. Tracker modifications are almost always recursive.

TIP: Non-tracker variables may also be recursive. However, despite this apparent similarity, the two types of variables are quite different. See Chapter 8 for more on recursive definitions of non-tracker variables.

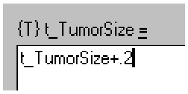
► **To create a tracker modification:**

▷ Select the node where the modification should occur.

▷ Choose Values > Define Values, and select the tracker variable from the list of variables in the Define Values dialog.

▷ Click the Value… button, and choose At Selected Node.

- ▷ In the Define Variable window, enter a recursive definition (such as the one shown at right), and click OK.



{T} t_TumorSize =

t_TumorSize+.2

For each trial, a hypothetical patient will traverse the tree (and/or Markov process), moving from one node to the next. As a node is entered, it is checked for tracker modifications. If a tracker modification is found, it is evaluated, and the new global value of the tracker is stored. In other words, if the size of the tumor should change when a certain event occurs, a tracker modification like TumorSize = TumorSize + DistSamp(1) could be assigned at that node.

When a tracker variable is referenced in the formula of another variable, the tracker's global value is used. No right-to-left search needs to be performed to find the value of the tracker variable for that scenario, as would be required with normal variables, since a tracker variable has only a single value for the entire model. It is possible to use a tracker variable in the argument for one of DATA's functions. For instance, the reward for a Markov state can be determined based on a function, such as If (TumorSize > 4; 0.3; 0.8). This will result in varying the utility for that state based on the current value of the tracker.

TIP: If you have more than one tracker modification at a single node, they will be applied in reverse alphabetical order. To avoid any confusion, if one tracker modification is dependent on another tracker modification, it is usually preferable to place them at successive nodes rather than at the same node. This can be accomplished by inserting a label node to the right of the node where the "input" tracker modification occurs, with the label node holding the dependent tracker modification.

Using expected value calculations with trackers

While expected value calculations are not recommended for trees which use tracker variables for purposes other than output, they are permitted. If you intend to perform expected value calculations on your tree, including Monte Carlo simulation of decision nodes, you should read the information in this section carefully.

Since tracker modifications are not meaningful when during expected value calculations, DATA will simply ignore. DATA will also ignore the initialization value set in the tracker's variable properties.

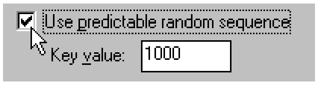
Anywhere the tracker variable is referenced in the tree, its value for purposes of expected value calculations will be zero. Therefore, you are likely to run into problems using tracker variables outside Monte Carlo simulation. For example, in the formula described above, If (TumorSize > 4; 0.3; 0.8), the value used for the tracker will always be zero.

In light of the potential for error, you are strongly advised to limit your use of trees with tracker variables to Monte Carlo simulation at non-decision nodes. For other calculations, a modified copy of the tree – this one without tracker variables – should probably be employed. Alternatively, it may be possible to modify a tree, using repetition of subtrees and logical statements, so that it works properly for both simulation trials and expected value calculations.

Note that trees that use trackers only as outputs are not subject to these problems.

Reproducing identical results

In certain situations you may want to force the same set of trials for several different runs of the simulation. You can cause DATA to use the same set of random probabilities and sample values in each run by the following two-step process:



☑ Use predictable random sequence

Key value: 1000

- ▷ In the set up dialog for a Monte Carlo simulation, check the Use predictable random sequence box.
- ▷ Specify a key value, which must be a positive integer up to 20,000.

Each key value will produce a different set of trials. However, if you use the same key value from one run to the next, DATA will choose the same random values each time. Changing your model (the structure, probabilities, or distributions) will not affect the random number generator, but the resulting values and calculations may be different.

The Monte Carlo text report

Here is a description of the columns you can expect to see in the text report output of your simulation.

Trial Lists each trial's order within the simulation.

Outcome Indicates the terminal node which was selected for this trial. (This will not be meaningful for Markov processes. DATA treats the Markov node as the endnode, and separately handles the internal details of the Markov process.)

Value The final value (not expected value) for this trial. If your tree is set to calculate cost-effectiveness, this column will display the cost-effectiveness ratio, and there will be two additional columns indicating the separate cost and effectiveness values.

Cost, Effect See Value.

Optimal If your simulation was performed at a decision node, this column will indicate which alternative was selected as optimal. This column will only be available if you have chosen to recalculate the optimal path at each trial. The optimal path is calculated using normal expected value methodology, but using the trial-by-trial sample values of your distributions.

You will also see additional value columns for each decision alternative. These columns will report, trial-by-trial, the expected values of each alternative at the selected decision node. Separate cost and effectiveness columns will be shown, if appropriate.

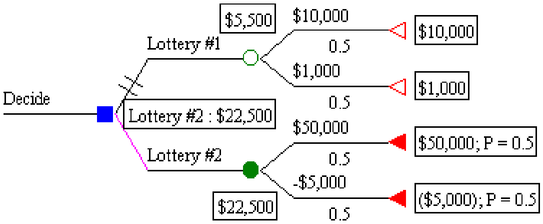
Trackers If you used Monte Carlo tracker variables, their values at the conclusion of the trial will appear, one column for each variable.

Distributions The value of each distribution you elected to sample will appear in its own column. Note that distributions which are sampled at each Markov stage will list only the trial’s final sample value.

Chapter 30 Risk Preference Functions

Assume that a rich uncle offers you an opportunity to win some money. He proposes to flip a coin giving you the opportunity to receive either $10,000 or $1,000, depending on whether you correctly predict the outcome. If you call the flip correctly, you will receive $10,000, and if you are wrong you will receive $1,000.

To make this game more interesting, assume that your uncle complicates matters by offering an alternative opportunity. The alternative is also a coin flip. Under this one, you will receive $50,000 if you are correct, but you will have to pay him $5,000 in the event you lose on the coin flip. There will be only a single coin flip; it is up to you to choose between the two. As you will see, it may not be wise to base your decision solely on traditional expected value calculations.



The tree at the left models your uncle’s offer.

As the tree demonstrates, there are two lotteries. Both provide the same (50 - 50) odds of winning, but they have different outcomes. You must choose one of them. On the basis of expected value, you should choose lottery # 2. Its expected value ($22,500) is more than four times that of lottery # 1 ($5,500).

However, what about the risk posed in lottery # 2 that you could actually end up losing $5,000? At least in lottery # 1 there is no risk of being out-of-pocket – you are guaranteed to win something. How one responds to the downside risk posed by lottery #2 involves a subjective analysis of the decision maker’s *aversion to risk*.

DATA enables you to enter your risk profile, which can be used to account for personal preferences. The end result will be that the comparative value of lottery # 2 will be mathematically decreased, based on the extent of your aversion to the risk of losing $5,000.

TIP: Some statisticians refer to a risk preference function as a *utility* function.

DATA can use risk preference functions only if the calculation method is set to Simple.

Certainty equivalents and risk aversion

Consider lottery # 1 described above. The expected value is $5,500. Would you sell the opportunity to play this lottery for $4,000? In other words, if you were offered $4,000 by a third party who wanted to buy into the lottery, would you sell? Would you sell for $3,000? What is the minimum value for which you would sell the lottery? This value is your *certainty equivalent* for this lottery. The certainty equivalent of a lottery can be perceived as the expected value of that lottery, adjusted for risk preference (the *risk-adjusted* expected value).

A certainty equivalent is similar to an expected value, in that it is a single numeric quantity which represents the value of an uncertain event. The expected value of an uncertainty is calculated mathematically as the probabilistic average of the possible outcomes. The certainty equivalent, on the other hand, is a purely subjective quantity. It is the answer to a question of the form, "What is the minimum (or maximum) value for which I would trade this uncertainty?"

Now consider a situation which is undesirable from the start. Lottery # 3 is a coin flip in which you will either owe your uncle $2,000 or you will owe him $12,000. In this situation, we are interested in finding the *maximum* amount that you are willing to pay to a third party to assume your obligation under the lottery. Would you pay $4,000? Or $5,000? Your answer to this question is your certainty equivalent for that lottery.

As you can see, the certainty equivalent for a lottery is usually in the same numeric range as the expected value. The gap between the certainty equivalent and the expected value is a measure of *risk aversion*.

Most decision makers are risk averse to some degree. They are willing to pay a premium, small or large, to avoid risk. Their certainty equivalent for any lottery will be lower than the lottery's expected value. In contrast, a risk-seeking decision maker is one whose certainty equivalent for a lottery is higher than the lottery's expected value. The risk taker is willing to pay a premium in order to participate in the lottery.

Creating a risk preference function

DATA is able to record your risk function as a mathematical curve, and apply this curve to the expected value of an uncertainty. Recommendations are then made based on your derived certainty equivalents, rather than on expected values.

There are two types of curves, or risk functions, which DATA can use. The first is called the *constant risk-aversion* function. It is calculated by using the formula

$$U(x) = 1 - e^{-x/R}$$

where U is an arbitrary utility scale, and R is a risk preference coefficient, described below. The utility scale is used only for internal calculations; the formula's inverse is later applied to find certainty equivalents.

If Constant Risk Aversion is selected in the Enter Risk Preference dialog, you will be asked to supply a single value. Specifically, you will be shown a simple lottery in which you have a .5 probability of winning X and a .5 probability of losing one-half X; and you will be asked to specify the largest value of X for which you would be willing to take part in the lottery. This value is used as the risk preference coefficient in the above formula.

You can think of the lottery as representing an investment in a biotech company which is about to get a judicial ruling on the validity of an important patent. If the ruling is favorable (.5 probability), the investment will double in value; if unfavorable (.5 probability), the investment will fall in value by 50%.

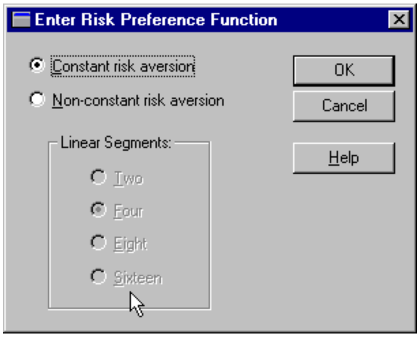
What is the most you would invest under these circumstances? This amount is referred to as your risk preference coefficient.

How do you assess a corporation's risk preference coefficient? The best way is to interview the CEO, but it is also possible to approximate this figure on the basis of net income or market value. For example, you might estimate that a company which is a moderate risk-taker has a risk preference coefficient about equal to annual net income, or to about one fifth of its market value.

Although the constant risk aversion function sufficiently describes many utility functions, in some cases the *non-constant risk-aversion* function may be more appropriate, as it is tailored to fit the issues and values specifically in question. On the other hand, it requires more patience and effort to set up.

DATA will ask you a series of questions about your certainty equivalents for the model you are working on. It will then create a linear approximation of your true risk function. Because this risk function is assessed on the basis of the range of payoffs in your model, it is not possible to set up this risk function until after your model is complete.

► **To assign a constant risk-aversion preference function:**



Enter Risk Preference Function

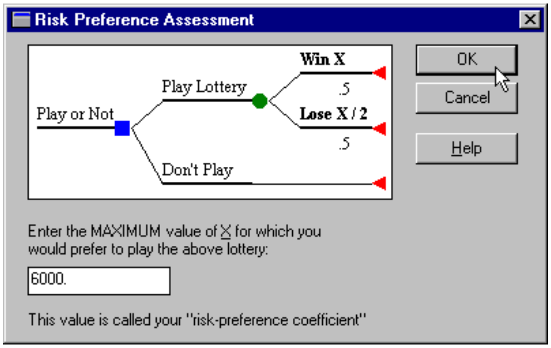
● Constant risk aversion

○ Non-constant risk aversion

Linear Segments:

○ Two
● Four
○ Eight
○ Sixteen

- ▷ From the Options menu, choose Enter Risk Preferences.
- ▷ Ensure that the Constant risk aversion button is selected, and choose OK.
- ▷ Answer the single lottery question shown. This is the risk preference coefficient.



Risk Preference Assessment

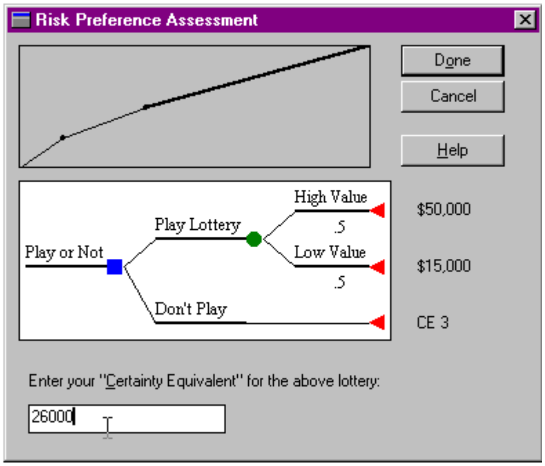
Enter the MAXIMUM value of X for which you would prefer to play the above lottery:

6000.

This value is called your "risk-preference coefficient"

► **To assign a non-constant risk-aversion preference function:**

- ▷ Ensure that your model calculates correctly without using any risk preference function. To use this risk preference function, DATA must be able to determine the range of potential payoff values.
- ▷ From the Options menu, choose Enter Risk Preferences.
- ▷ Ensure that the Non-constant risk aversion button is selected. Select the number of linear segments used to approximate your preference function. More segments will result in a more accurate picture of how your risk preferences vary over a range of potential outcomes, but you must take the time to answer more questions about your certainty equivalents.



Enter your "Certainty Equivalent" for the above lottery:

26000

- ▷ DATA will ask you for certainty equivalents for a series of lotteries based on your tree. For each lottery, you must assign the minimum (or maximum) certain value for which you would trade the uncertainty displayed.
- ▷ Click the More button after entering each certainty equivalent. For the last certainty equivalent, click Done.
- ▷ As you supply the requested information, DATA will generate your risk preference curve, shown in the top part of the window.

TIP: If you change any of the values in your model after having developed a non-constant risk preference function, the latter will no longer be valid if any of the new payoff values fall outside the range of payoffs considered in the risk preference function. DATA will warn of an attempt to apply an invalid non-constant risk preference function.

Risk preference curves

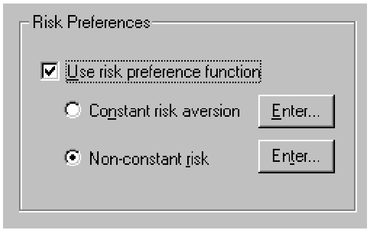
A straight-line risk-preference curve represents a decision maker who is *risk-neutral*. This type of decision maker bases decisions on expected values rather than certainty equivalents.

A risk-averse decision maker will have a curve with a decreasing slope, meaning that certainty equivalent is less than expected value. The curve will typically be steeper in the low value range, where aversion to risk is weak, and will grow progressively flatter as the values get larger (both positive and negative), where aversion to risk becomes stronger. The more risk-averse you are, the more your curve will deviate from the 45° straight line representing risk neutrality.

If you encode a curve that includes some unexpected bumps, this means that some of your responses were inconsistent. You should repeat the process. Don't be discouraged; developing a meaningful non-constant risk utility curve takes hard thinking and careful consideration.

Other features

You may set your risk preferences in the Preferences dialog box. From there you may turn the risk preference function on or off, and you may enter the risk preference functions directly.



Risk Preferences

☑ Use risk preference function

○ Constant risk aversion

● Non-constant risk

Even if a risk preference function has been developed, it will not be applied in calculating the tree unless the Use Risk Preference Function option (in the Preferences dialog) is selected. When this option is turned on, all values are mapped to certainty equivalents.

When the risk preference function has been turned on, an item in the status bar will read RISK. In addition, the boxes that appear to the right of each node following Roll Back will be drawn with rounded corners.

Your risk preference function can be graphed by selecting Graph Risk Preference Function under the Analysis menu.

If you are using the constant risk-aversion preference function, you may perform a one-way sensitivity analysis on the risk preference coefficient. At the bottom of the list of variables specified in the sensitivity analysis dialog box will appear Risk Preference Coeff. If you select this “variable,” your analysis will graph the effect of varying the coefficient’s value.

CHAPTER 31 WORKING WITH INFLUENCE DIAGRAMS

This chapter builds on the basic information covered in Chapter 4 on using influence diagrams. Knowledge of the topics covered in this chapter – including how to create asymmetries, how influence diagrams are converted to trees, and how to use variables and value nodes – will apply to every influence diagram you create in DATA.

Chapter 32 also deals with influence diagrams. It covers a number of advanced topics which will apply only in selected situations.

When to use influence diagrams

DATA's influence diagram window has both benefits and drawbacks compared to the tree window, which may affect your decision about whether to begin a particular model as an influence diagram. First, the benefits:

Bayes' revision

DATA's implementation of influence diagrams includes some features that simply are not available in the tree window. Possibly most significant is the ability to have DATA automatically calculate posterior probabilities using Bayes' revision with sequential tests. The implementation of Bayes' revision in the tree window can handle only a single test. Moreover, even if your model involves only a single test, you may find that DATA's implementation of Bayes' revision is handled more intuitively in the influence diagram window. For example, in the tree window, you must perform Bayes' revision prior to modeling any intervening decisions. Since this restriction does not apply to influence diagrams, several steps in the model-building process can be avoided.

EVPI

Calculation of EVPI is also handled more elegantly in the influence diagram window. There are some situations that the tree window's implementation of EVPI cannot handle, especially in larger trees. Calculating EVPI in the influence diagram window is straightforward and simple, and it handles all cases. See Chapter 32 for detailed instructions on implementing Bayes' revision and EVPI in the influence diagram window.

Model size and other considerations

Even in situations where these special computational features are not important, it is often beneficial to implement your model as an influence diagram. Many experienced model builders report that the ability to build the model as an influence diagram that can be converted automatically into a fully-functioning, asymmetrical tree offers the best of both worlds.

With its compact size, it is often far more practical to print and distribute copies of an influence diagram than the associated tree. Moreover, building an influence diagram forces you to consider issues of influence that may be overlooked when building a tree, and the resulting information displayed on the face of the influence diagram can enhance communication about the problem being modeled.

For many people learning to build influence diagrams, the greatest frustration flows from uncertainty about whether their influence diagram correctly models their problem. In DATA, the remedy is to convert the influence diagram into a tree. If its structure is correct, then so is the influence diagram's. If the tree is not the one you expected to see, return to the influence diagram, correct it, and test its accuracy by converting it into a tree. Since DATA produces fully-configured trees with appropriate asymmetry, you should have no trouble telling whether the model is accurately structured.

Limitations of an influence diagram

With all these good reasons for starting your model as an influence diagram, you may be wondering why it might ever be preferable to skip the influence diagram and begin your model as a tree. In some cases, it is a matter of balancing benefits against effort and time consumed.

For example, certain features – particularly multi-attribute (such as cost-effectiveness) models and Markov processes – are not available in influence diagrams. This doesn't mean that a model begun as an influence diagram can never contain these features, but simply that they will have to be added after the influence diagram has been converted into a tree.

As the model builder, you will have to decide, on a case-by-case basis, whether it pays to begin a model as an influence diagram when it cannot be converted into the completely functional tree you require. In some cases, such as where the model involves the need to apply Bayes' revision to sequential tests, the benefits of beginning the model as an influence diagram will continue to tip the balance in its favor.

Similar questions will have to be asked if your model will employ a large number of analytical distributions for Monte Carlo simulation, because distributions are handled more thoroughly in the tree window. Of course, you may find that doing the preliminary structural modeling in the influence diagram suits your needs; once you convert your model to tree form, you can utilize Markov processes, multi-attribute analysis, or distributions as if you had started in the tree window.

Time ordering of nodes

In DATA, it is not possible to perform calculations directly on an influence diagram. To calculate the model, the influence diagram must be converted into a tree; you may need to do this a number of times while building the influence diagram. Thus, to correctly structure an influence diagram you must know the rules by which DATA will convert that influence diagram into a tree. Fortunately, most of these rules conform to the standard rules for constructing influence diagrams.

When an influence diagram is converted into a tree, DATA uses a fixed set of rules to determine the order in which influence diagram nodes should be converted into tree nodes, beginning at the root node of the tree.

> **TIP:** The *no-forgetting* principle of arcs simply states that if a node precedes a decision, it must also precede all subsequent decisions. Thus, the information is remembered at all subsequent decision points. In DATA, no-forgetting arcs are not required, so long as you draw an arc from each decision or chance node to its immediate successor decision node. However, in order to avoid confusion when sharing your model with others, you may want to include arcs to all subsequent decisions.

Here is the complete algorithm used by DATA to determine node ordering. The algorithm conforms to standard practice.

- **Decision nodes are ordered.** DATA determines in what order decisions occur by looking at the arcs between them. You should draw arcs from a decision node to every other decision node which occurs later in time. This convention not only resolves any ambiguity regarding decision ordering, but also conforms to the standard use of *no-forgetting* arcs. (Any remaining ambiguity in ordering the decisions is resolved by graphical node position, as described below.)

- **Chance nodes are grouped.** For each chance node, DATA determines which decision nodes it precedes and which it follows. This determination is made solely on the basis of direct arc-connections. If there is an arc from the chance node to a decision node, the chance node must precede the decision, and all subsequent decisions. If there is no arc, or if the arc points from the decision node to the chance node, the decision node will be converted first.

  You should draw arcs from a chance node to every decision it precedes. If the outcome of an uncertain event is known when the decision is made, the uncertainty precedes the decision. This also conforms to the *no-forgetting* principle of arcs.

  At this point, chance nodes will be grouped in positions between decisions, although no ordering has taken place within each group. Some chance nodes may, of course, precede or follow all decisions.

- **Chance groups are individually sorted.** For each group of chance nodes, the order is determined by considering arc flow, with a node at the base of an arc converted before the node at the tip. Ambiguities are resolved by considering graphical position, as described below. Each group is ordered independently; for this purpose, arcs from nodes in one group to nodes in another group are ignored.

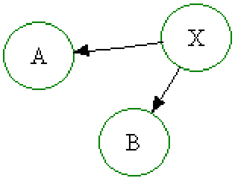
How the standard conversion algorithm makes it possible to automate Bayes' revision and EVPI is covered in Chapter 32.

In DATA, it is possible to create timing-only arcs to indicate the ordering of nodes, consistent with the *no-forgetting* principle. To draw a timing-only arc, simply uncheck all of the Probs and Values check boxes in the arc editing window, and ensure that all the structural influences are set to "Symm" (symmetric).

Using graphical position to resolve time order

Some timing issues may not be resolved by considering arcs alone. In these cases, DATA will convert nodes on the left of the influence diagram before nodes on the right. (There is a preference item which will force this left-to-right ordering to be performed top-to-bottom.)

For instance, consider the following influence diagram fragment.

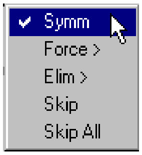


The arcs indicate timing, so X will be converted before both A and B. Since the arcs provide no way to tell between A and B which should appear first in the tree, their respective positions in the influence diagram are used to make this determination. Since A is to the left of B, the nodes will be converted X, A, B.

The left-to-right (or top-to-bottom) ordering of nodes uses the *center* of each node as the point by which sorting occurs.

Asymmetry

This topic was covered initially in Chapter 4. If you have not worked through the tutorial in that chapter, it may be helpful to do so now.



Here is a description of the various structural influence types and how you might use them:

- **Symm -** Short for symmetric, Symm this indicates that, for the influenced event, the tree should be as bushy as possible, with all branches drawn.
- **Force -** Use this influence type to indicate that when one *conditioning* event occurs (or one alternative is chosen), the result (or choice) associated with the *conditioned* node is known or determined. You will need to pick which outcome is forced via a pop-up menu in the Additional Info field.

  Consider, for example, a situation in which you must decide whether to (i) replace a home-heating furnace thought to be faulty, (ii) perform maintenance on it, or (iii) leave it alone. If you leave it alone, the outcome is uncertain (with possible outcomes “Catastrophic failure,” “Minor failure,” and “No problem”). However, if you choose to replace the furnace, then the outcome will always be “No problem.” In some cases this can be handled with the “Skip” command, but in many situations you will want to include (force) the branch “No Problem” for clarity.
- **Elim -** Similar to “Force,” this command eliminates one possible outcome or alternative from the conditioned node. You will need to pick which outcome is eliminated via a pop-up menu in the “Additional Info” field.

In the furnace example above, we might use the “Elim” command to indicate that if we choose to perform maintenance, then “Catastrophic failure” may be eliminated. The situation is still uncertain, since the part may experience minor failure, but one possibility has been eliminated.

- **Skip -** This is the most common type of structural influence. It indicates that when one outcome occurs (or one alternative is chosen), all branches associated with the *conditioned* node should be omitted; events that follow the skipped node will still be included in the model.
- **Skip All -** Use this to indicate that *all* subsequent events are to be eliminated. This is a shortcut for creating many Skip arcs to other nodes in the influence diagram. Whenever the particular outcome or alternative is reached, it will become a terminal node in the tree.

When an arc has no probabilistic or value influence, and is used only to indicate asymmetry, it is drawn in dotted gray. You may choose not to print these nonstandard arcs; see the section describing preferences below.

Variables and values

This section describes how to use node variables, value nodes, and deterministic nodes to create payoff formulas for your models.

Node variables

Each node in an influence diagram represents a parameter in your model, and DATA will create a variable for purposes of including that parameter in its calculations. Although you may name the node anything you wish, the variable which represents that parameter must meet the naming guidelines set forth in Chapter 8. DATA will automatically generate a conforming variable name from the node name when you first create it. You may then modify the variable name manually.

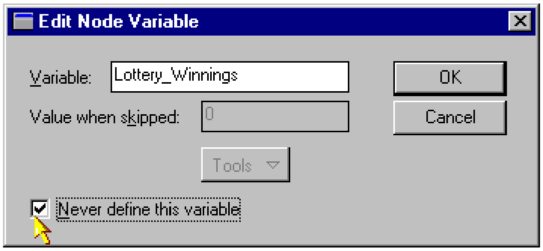
A variable associated with an influence diagram node may be used only to calculate a payoff formula. Node variables may *not* be used to assign probabilities; see the discussion below on using the assessment window.

Not every parameter in your model will be a part of a payoff formula; indeed, not every model will have a payoff formula. Accordingly, DATA makes it possible to suppress the use of variables for each node individually.

► **To indicate that a node’s variable will not be used:**

▷ Select the node, and choose Variable from the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu (or from the Diagram menu in the main menu bar).

▷ Check the box labeled Never define this variable.



Edit Node Variable

Variable: Lottery_Winnings

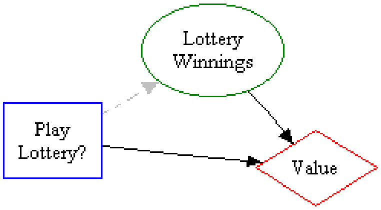
Value when skipped: 0

☑ Never define this variable

Setting this flag will prevent the node’s variable from being defined in the tree. As a result, the node’s variable will not be available for use in the payoff formula, and the node will not be available for value conditioning by another node (no arc pointing to the node may indicate a values influence type).

The use of node variables in formulas is discussed below in the context of value nodes.

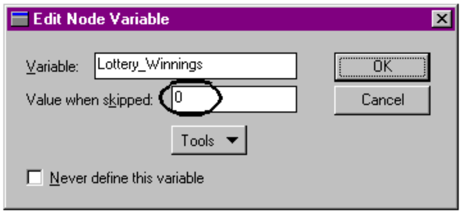
Node variables and asymmetry



The model in the left margin illustrates asymmetry. The value of the model (i.e., the payoff formula) is the difference between the value of the lottery and the cost of playing. If, however, you decide not to play the lottery, and the uncertainty is skipped, how will DATA determine the value of the Lottery variable for purposes of calculating the payoff?

If a node whose variable is used as part of the payoff formula is skipped by asymmetry, DATA offers you the chance to give it a default value. For instance, if you decide not to play the lottery, you would give the lottery a value of 0 whenever asymmetry indicates that the node should be skipped.

► **To assign a default value to a variable:**



Edit Node Variable

Variable: Lottery_Winnings

Value when skipped: 0

□ Never define this variable

▷ Select the node, and choose Variable from the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu (or from the Diagram menu in the main menu bar).

▷ Enter the default value in the data entry box named Value when skipped.

Value nodes

There are two different uses for value nodes. The primary purpose of value nodes is to act as a placeholder for the model's final outcome (payoff). This payoff node has arcs leading in from the nodes whose outcomes affect the payoff, but has no arcs leading out. A payoff node either may be enumerated on a scenario-by-scenario basis, or may be described by a formula, as discussed below.

The secondary purpose of value nodes is to create an intermediate formula. This is useful for combining formula components in the main payoff. For example, if you want your main payoff formula to be calculated as Income-Costs you might create an intermediate value node called Income, using a formula UtilDiscount(Sales * Price; Rate; 0.5). The various components of this formula would be represented by other nodes with arcs pointing into the intermediate value node, which would then have an arc pointing to the final value node. The same could be done for Costs.

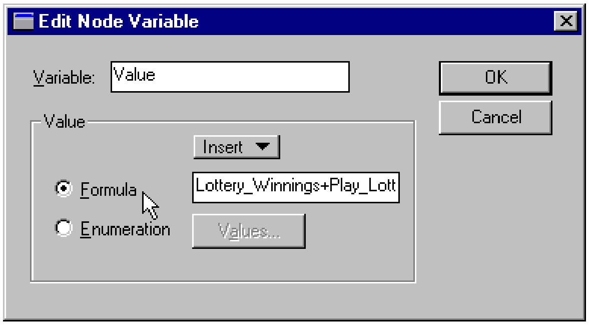
Every influence diagram must contain a *final* value node for assigning final outcomes. The use of intermediate value nodes is optional.

Final value nodes may be described either by formula or by full enumeration. If you opt to assign a formula to a final value node, that formula will be defined at the root node of the converted tree, and used as the payoff value for all terminal nodes. If you choose to enumerate the values for the final value node, you will be able to assign a different value (or possibly a formula) to each terminal node in the tree.

In contrast, intermediate value nodes may be described only by formula.

► **To assign a formula (intermediate or final) to a value node:**

▷ Select the value node, and choose Variable from the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu (or from the Diagram menu in the main menu bar).



Edit Node Variable

Variable: Value

Value

● Formula Lottery_Winnings+Play_Lott

○ Enumeration

▷ Select the Formula radio button.

▷ Enter the formula in the editor. You may use the Insert pop-up menu to select variables from nodes which directly influence the value node.

► **To enumerate payoff values for a (final) value node:**

▷ Select the value node, and choose Variable from the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu (or from the Diagram menu in the main menu bar).

▷ Select the Enumeration radio button.

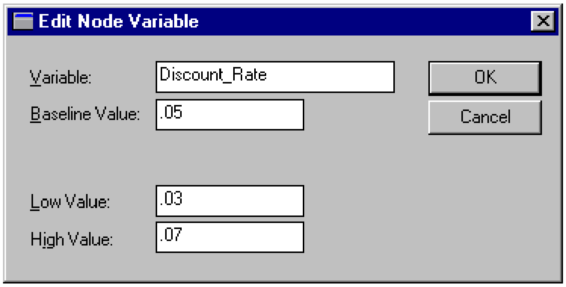
▷ Select the Values button to bring up the tree-style value-assessment dialog.

As an alternative to using the Values button in the Variable dialog, you may choose the Values menu item in the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu. In this case, you must first ensure that you have selected Enumeration in the Variable dialog. (Enumeration is the default for final value nodes.)

Deterministic nodes

A deterministic node is useful for including a parameter which has a single, fixed value in your model, even if your estimate is uncertain. You have the option of assigning not only a numeric value to a deterministic node, but also a range of possible values for future sensitivity analysis.

Discount
Rate



Variable: Discount_Rate
Baseline Value: .05
Low Value: .03
High Value: .07

In other words, deterministic nodes in an influence diagram are used in much the same way that variables are used in a tree. Rather than entering a fixed numeric value, use a deterministic node to allow for more complete analysis. The variable associated with the deterministic node may then be used in the probability or value associated with whatever node(s) it influences. See the section Using the assessment window for more details.

All variables created from deterministic nodes will be defined at the root node of the converted tree.

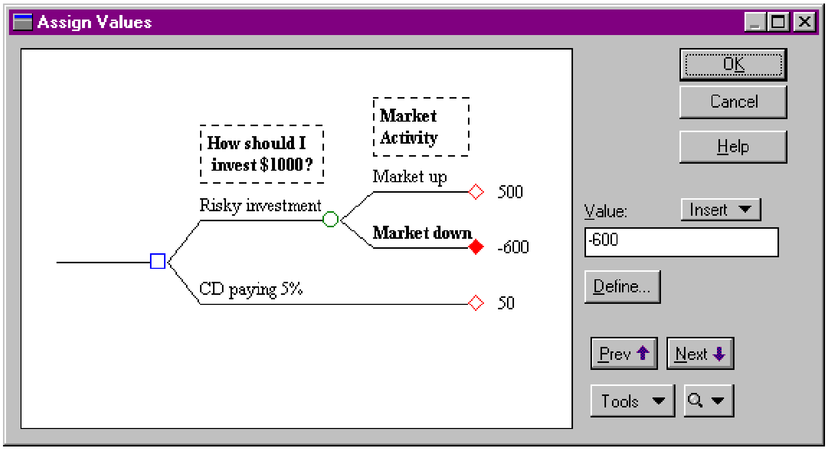
Using the assessment window

This window is used to assign both probabilities and variable (or payoff) values. Its usage is virtually identical in both.

Maneuvering around the mini-tree

Each node which requires your attention will be displayed with a red diamond. Note that while these nodes look like terminal nodes in this *mini-tree*, they are not necessarily final outcomes.

The first red diamond node, at the top of the mini-tree, will be selected initially. A selected node will have its diamond filled in and its name drawn in bold. Only red-diamond nodes may be selected.



When a node in the mini-tree is selected, the text editor in the right part of the dialog becomes active. The information you enter in the text editor will be used for the selected node.

To select another node in the tree, you may use any of the following methods:

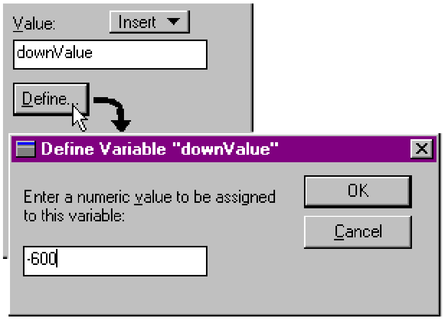
- click directly on the node in the mini-tree;
- use the Prev/Next buttons to select the next node in the indicated direction; or
- hold the CONTROL key and press either UP or DOWN ARROW.

To view more or less of the mini-tree, use the Zoom pop-up menu (displayed with a magnifying glass icon) or resize the window itself.

Entering values in the text editor

The text editor accepts any valid number or expression. You may enter numeric values directly, such as "0.4," or you may enter expressions, such as *1-pDown*, *Cost/(1+rate)^time*, or *#*.

You may also define a new variable. This will have an effect similar to using a deterministic node, without the visual clutter of an extra node in your diagram. Unlike deterministic nodes, in which the value is always defined at the root of the converted tree, variables defined in the mini-tree are defined at the selected node (or its parent) in the converted tree. This may result in multiple (identical) definitions in the tree.



downValue

Enter a numeric value to be assigned to this variable:

-600

► **To define a new variable in the assessment window:**

▷ Type the name of the variable in the editor box.

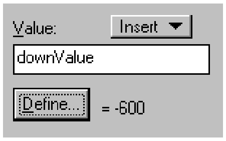
▷ Click the Define button.

▷ Enter a numeric value in the ensuing dialog box, and press ENTER or RETURN.

The numeric value is displayed next to the Define button, as well as in the mini-tree.

▷ To change the definition of the variable, click Define again.

▷ To eliminate the definition, simply type over the variable's name in the editor. DATA will use the new expression and eliminate the old variable.



downValue

= -600

Using variables in the text editor

The Insert pop-up menu enables you to use variables from other, influencing nodes. For example, consider a decision node, Project, whose values represent the cost of the project, and a chance node, Sales, whose values represent the possible sales revenues. If these two nodes influence a value node, you may use the Insert pop-up menu to create the expression Sales - Project.

You are urged to use variables in the editor box rather than assigning numeric values directly. These variables may be defined either in the editor box (as just described) or at a separate deterministic node. Entering variable names directly in the editor (regardless which method you use to define the variables) will encourage you to assign different variable names to different outcomes, such as “CostLow” and “CostHigh,” making it very simple to perform sensitivity analysis on your parameters. (See Chapter 8 for more on this subject.)

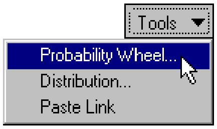
If, instead, you enter numeric values directly, you run the risk of having multiple numeric definitions of the same node variable. In other words, the node’s main variable will be defined at each branch with each of the different numeric values in turn; this will cause analysis problems later. (See Chapter 22 for a discussion of the complexities of performing sensitivity analysis on variables having multiple numeric definitions.)

Probability wheel

The probability wheel is available when editing probabilities for a chance node with at least two but not more than seven outcomes. The probability wheel and its use are described in Chapter 11.

► **To use the probability wheel:**

- ▷ Select one of the branches of a chance node. The wheel operates on the *set* of branches, so it does not matter which of the branches you select.
- ▷ From the Tools pop-up menu, choose Probability Wheel.



Distributions

While it is possible to assign analytical distributions in the influence diagram assessment window, it is recommended that you wait until after you have converted the diagram into a tree. There are several reasons for this.

Distributions may be thought of as having both a probabilistic element and a value element; they typically represent the probability of attaining a value in a particular range. However, in the influence diagram assessment window, you will need to assign the distribution separately in the probabilities assessment and values assessment windows.

Furthermore, in the influence diagram window, the distribution will be created in such a way that it will not be sampled during Monte Carlo simulation. For information on using distributions in the tree window, see Chapters 28 and 29.

Linked values (Windows only)

In DATA for Windows, it is possible to link to externally stored values while in the influence diagram. The links you create will be included in the converted tree as a DDE linkage.

► To link to a value stored in another program (Windows):

- ▷ Switch to the other application. Select the value to which you will link, and choose Edit > Copy.
- ▷ Switch to DATA. Select a node in the influence diagram and choose Diagram > Probabilities or Diagram > Values.
- ▷ In the mini-tree window, select the node at which the link should be stored.
- ▷ From the Tools pop-up menu, select Paste Link.

The link will be stored in the influence diagram window. You may edit the link using the Links dialog in the Edit menu. When you convert the influence diagram into a tree, the links will be copied and included as DDE links. When the value changes in the external application, DATA will update it in both the influence diagram and the converted tree.

See Chapter 15 for more information on using DDE links.

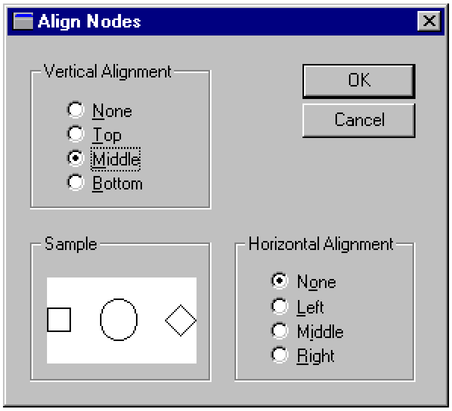
Miscellaneous

Node description

You may annotate nodes in the influence diagram. Select the node you wish to annotate, and choose Diagram > Description. In the Node Description dialog, you may change both the node name (as it displays in the window) and the hidden annotation. This annotation is not carried over to the converted tree.

Aligning nodes

When your influence diagram is ready for presentation, you may wish to align nodes so that arcs appear perfectly straight. To accomplish this, select those nodes you wish to align, and choose Display > Align.


Align Nodes

Vertical Alignment
○ None
○ Top
● Middle
○ Bottom

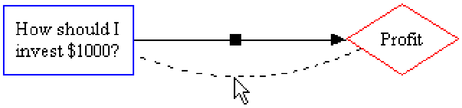
Horizontal Alignment
● None
○ Left
○ Middle
○ Right

If you vertically align the centers of your nodes, then arcs between them will be drawn perfectly horizontally. If you horizontally align the centers of your nodes, then arcs between them will be drawn perfectly vertically.

Remember that ambiguous time-ordering is resolved on the basis of node position; in particular, the position of the *centers* of the nodes is used to sort them. If ordering is performed left-to-right, and two nodes have been horizontally aligned at their middles, so their horizontal position is identical, then the resolution of time-ordering is unpredictable. In situations such as this, you should use a visible, even exaggerated, horizontal displacement to indicate time-ordering.

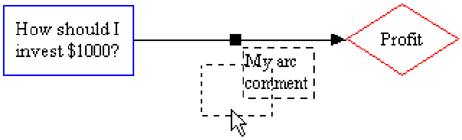
Arc operations

You may curve a selected arc by dragging its square handle . To straighten a curved arc, select Diagram > Straighten Arc.



To flip a selected arc’s direction, choose Diagram > Flip Arc. Any asymmetry specified in the arc will be lost, as the direction of influence has been reversed. This operation is primarily graphical; it does not perform intelligent reassignment of probability or value information. However, careful arc-flipping can help you calculate EVPI in the converted trees. See Chapter 23 for more on EVPI.

Arcs may be annotated. Double-click the arc, and enter a comment in the dialog. If the comment displays in an undesirable location, it can be moved by clicking anywhere inside the comment and dragging.



Arcs may not be cut and pasted like other objects, because the information stored in an arc is specific to the nodes that the arc connects. If you copy two nodes linked by an arc, then the arc will be copied (and pasted) with the nodes. To eliminate an arc, choose Edit > Clear Arc.

CHAPTER 32 ADVANCED INFLUENCE DIAGRAM FEATURES

Bayes' revision

Bayes' revision constitutes an exception to the rule that all of DATA's calculations are performed in the tree window. Since Bayes' revision involves not only calculations but also elements of model design, DATA performs all calculations associated with Bayes' revision during the process of converting your influence diagram into a tree.

Accordingly, you must enter the likelihood probabilities (such as the probability of a positive test result on a patient known to have a certain disease) within the influence diagram window. Based on this information, DATA will calculate the decision probabilities (such as the probability that a patient has the disease if she has tested positive) at the time that the influence diagram is converted into a tree.

See Chapter 24 for further information on Bayes' revision.

If you set up your Bayesian tests properly in the influence diagram window, you will have no need for DATA's Bayes' revision tool in the tree window, as all of the revised probabilities will have been calculated for you.

The following tutorial will generate the model which is included as sample file Bayes ID. You may want to check your results against that file after working through the tutorial.

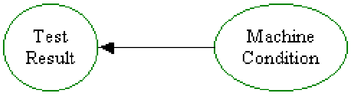
The problem being modeled involves deciding whether or not to replace the product of a manufacturing process, referred to in the model as a machine. If the machine is faulty, it should be replaced. If it is not faulty, the cost of replacement would not be warranted. A reasonably reliable, but imperfect, test is employed to help ascertain the machine's condition.

This chapter assumes you are familiar with how DATA converts influence diagrams into trees, as discussed in Chapters 4 and 31.

Setting up a single forecast

You must first create the nodes which represent the true condition (whether the machine is faulty) and the forecast of that condition (the test).

- Create a chance node called Machine Condition, with two outcomes Faulty and Not Faulty.
- Create a second chance node called Test Result with two outcomes: Test Positive and Test Negative.
- Draw an arc from Machine Condition to Test Result.

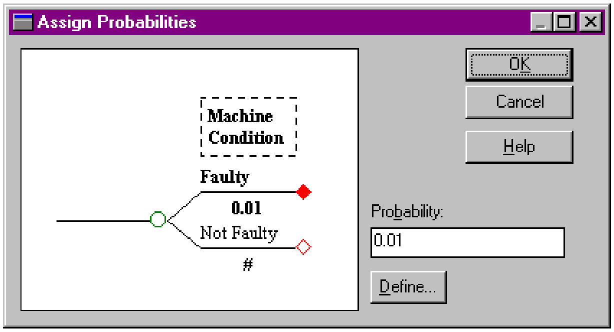


The direction of the arc indicates the direction of influence; probabilities of testing positive or negative are conditioned on the actual status of the machine. This conditioning reflects the data to which one typically has access, namely, the quality of the test.

The order in which the events unfold is opposite to the direction of influence. Even though Machine Condition influences Test Result, the outcomes of the test are known before the true machine condition is determined. Under normal circumstances, a node at the base of an arc is converted before the node at the tip; in other words, arcs represent timing as well as influence. In this situation, the timing of the nodes should be opposite from the direction of the arc. Only by adding an intervening decision, as you will do momentarily, can the software determine that the conversion order is opposite to the arc flow.

Entering probability data

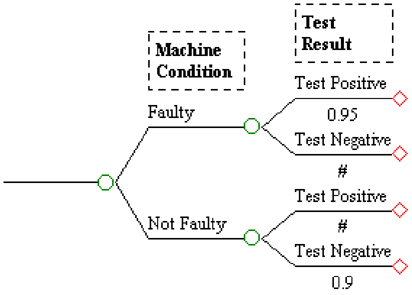
- Select the Machine Condition node, and choose Probabilities from the right-click (Windows) or CONTROL-click (Macintosh) pop-up menu (or from the Diagram menu in the main menu bar).
- Assign `0.01` to the Faulty outcome, and # to the Not Faulty outcome.



Probability: 0.01

These values are called the *a priori* probabilities; they represent the prevalence of a certain fault in machines of this type.

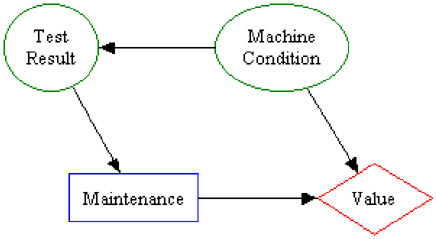
- ▷ Select the Test Result node, and choose Diagram > Probabilities.
- ▷ For the test outcomes conditional on Faulty, enter 0.95 for Test Positive, and # for Test Negative.
- ▷ For the test outcomes conditional on Not Faulty, enter # for Test Positive, and a value of `0.9` for Test Negative.



TIP: In the included sample file Bayes ID, prior and likelihood probabilities are assigned using variables, such as pTruePos. If you use variables, as is generally recommended, you will be able to perform sensitivity analysis on these values after converting the influence diagram to a tree. See Chapter 31 for more detail on using variables in the Assign Probabilities window.

Adding an intervening decision and a value node

Next, it is necessary to add a decision based on the test results, and a final value node. Without an intervening decision, Bayes' revision will not be applied, and the chance nodes will not be ordered correctly. The value node is required to create a properly structured influence diagram.

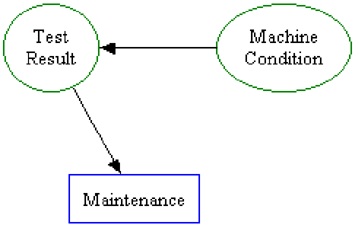


- ▷ Create a decision node called Maintenance, with two alternatives: Replace and Don't Replace.
- ▷ Create an arc from Test Result to Maintenance to indicate that the test results are known prior to making the decision.
- ▷ Create a value node called Value.
- ▷ Create two arcs, one from Maintenance to Value and the other from Machine Condition to Value.

Since the purpose of this example is to work through the steps needed for Bayes' revision, you can skip the process of assigning values at the value node. If, however, you wish to view the calculated probabilities during roll back, you might indicate that the value node has a formula (rather than a full enumeration), and assign a formula of zero. While this is not meaningful, it will at least allow DATA to roll back the tree.

Understanding the implications

For the moment, ignore the value node, as it has no bearing on the conversion order. Focus your attention on the relationships among the other three nodes.



The only arcs are from Machine Condition to Test Result, and from Test Result to Maintenance.

The absence of an arc from Machine Condition to Maintenance indicates that the uncertainty will not be resolved until *after* the decision is made. In contrast, the arc from Test Result to Maintenance specifies that the test result is known *before* the decision is made.

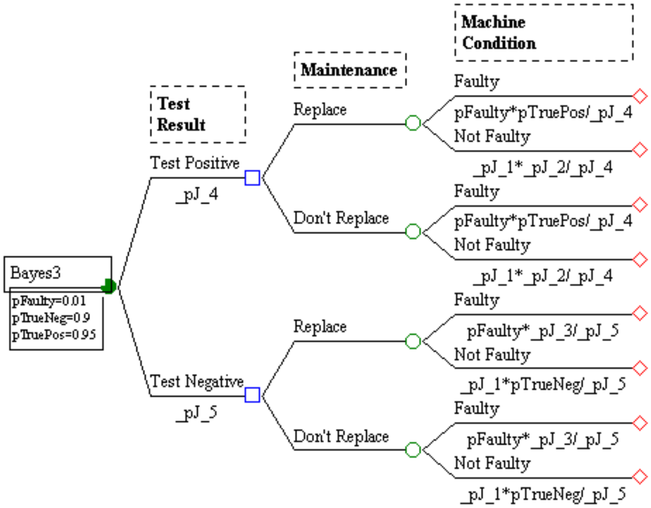
When DATA converts your influence diagram into a tree, the nodes will be converted in the following order: Test Result, Maintenance, Machine Condition. This makes sense – the test result is first learned, then a decision is made, and then the true condition is learned.

You may be wondering about the arc between Machine Condition and Test Result, which indicates probabilistic dependence (described above). Why will these nodes be converted in the *opposite* order from the direction of arc flow? You may recall that node ordering is performed by first ordering the decisions, then grouping the chance nodes according to which decisions they precede and which they follow, and finally ordering the nodes within specific chance groups. Because Test Result and Machine Condition fall into different chance groups (one before and the other after the decision), the arc between them is not used to determine their relative ordering.

In cases such as this one – where the order of conversion will be opposite to the arc flow between two chance nodes – DATA will automatically apply Bayes' revision when the influence diagram is converted into a tree.

Seeing the results

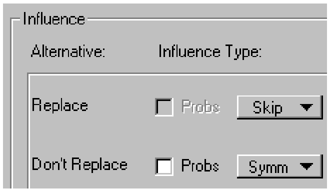
Convert the influence diagram into a tree to see how DATA performs Bayes' revision.



It is possible to view the calculated probabilities by rolling back the tree. To do so, however, you should temporarily set all payoffs to zero, and set the option to display probabilities as numeric equivalents (in the Roll Back page of the Preferences dialog). When you roll back the tree, the inverted probabilities will be displayed in your tree.

Asymmetry inside the Bayesian model

One of the decision alternatives is costly, but risk free. If the potentially faulty machine is replaced, the new machine will run, regardless of whether the machine replaced was actually faulty.



Influence

Alternative:	Influence Type:	
Replace	□ Probs	Skip ▼
Don't Replace	□ Probs	Symm ▼

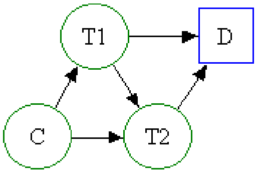
In this situation, you may add an arc from Maintenance to Machine Condition. You should ensure that the arc contains no probabilistic influence, as the condition of the original machine is not dependent on the decision. This arc may only contain structural influence. Now, you can indicate that the Machine Condition node should be skipped when the alternative Replace is chosen.

Convert the influence diagram into a tree to view the resulting asymmetry.

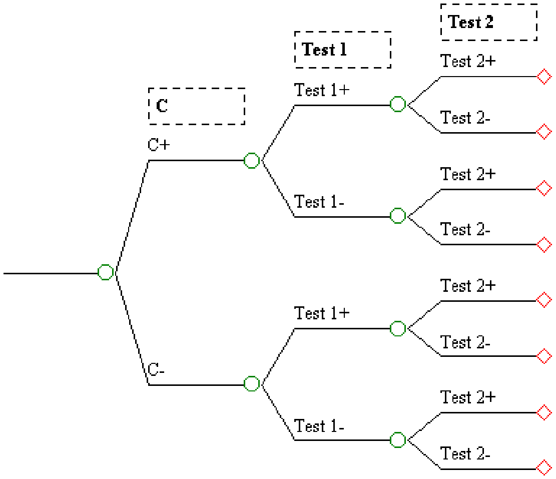
Bayes' revision with sequential tests

DATA knows how to apply Bayes' revision when more than one test (or predictor) is used on a single event (or hypothesis).

Setting up the required influence diagram structure, shown at right, is quite simple.



The hard part comes next. In order for DATA to be able to perform its probability revision calculations, you must provide all the information identified in the tree fragment.



For example, you must know the probability of testing positive on test 2, given that the underlying hypothesis C is positive and that test 1 returned a negative result. Obtaining these conditional probabilities is critical and may be difficult. There is no simple rule; the problem is always situation-dependent.

It is possible to specify asymmetry in the arc from Test1 to Test2 if, for instance, Test2 is not taken if Test1 is negative.

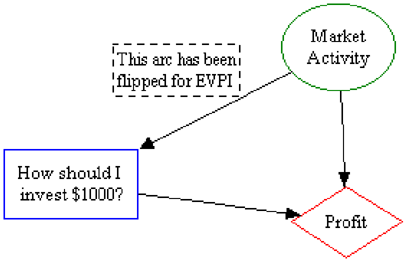
Expected value of perfect information

The expected value of perfect information (EVPI), as described in Chapter 23, is a measure of the maximum amount one should be willing to pay for a predictor of uncertainty. See the discussion in that chapter for general information on the concept of EVPI.

EVPI is calculated by inverting the time order of a chance node and a decision node. Under normal circumstances, the outcome of the chance node is not known before the decision is made. For EVPI calculations, we assume that the outcome is known before the decision.

To calculate EVPI, you should flip the arc pointing from the decision node to the chance node. Or, if there is no arc, add a new one from the chance node to the decision node. This will ensure that the uncertainty is resolved before the decision. The expected value of perfect information is equal to the value of this converted tree minus the value of the original tree.

A meaningful value of EVPI requires that there be arcs from both the subject decision node and the subject chance node to the final value node.

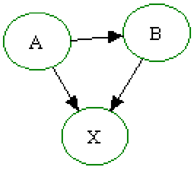

This arc has been flipped for EVPI
Market Activity
How should I invest $1000?
Profit

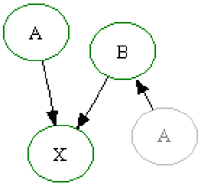
Revising the influence diagram in this way may result in making some nodes unnecessary. For instance, if there is an imperfect predictor of the chance node in question, it will become irrelevant in the presence of a perfect predictor. DATA will not automatically remove these nodes for you, but they will not affect calculations in the converted tree.

Other points to bear in mind:

- the application of Bayes' revision may be affected by these changes to the model; and
- any asymmetry specified in the original arc (from the decision node to the chance node) will be lost when the arc is flipped.

Clones





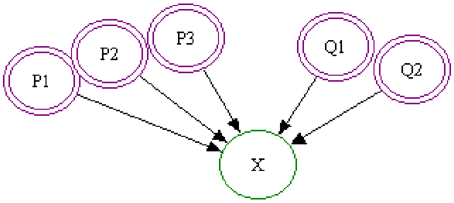
A clone of an influence diagram node is drawn in gray. Any changes you make to the clone will actually be made in the original ("master") node. Any arcs to or from the clone node act as if they pointed to or from the master node. Thus, the two influence diagrams in the left margin are functionally identical.

To create a clone, select a node and choose Create Clone from the Edit menu. You may also destroy all clones of a node by selecting Edit > Destroy Clones when the clone master is selected.

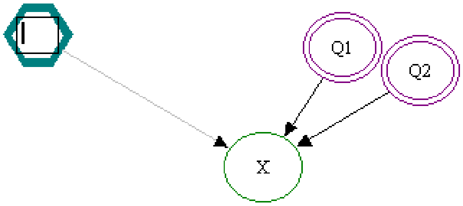
Note that the position of a clone is not relevant during conversion to tree. If graphical positioning is needed to resolve ambiguities in time-ordering, only the position of the master node will be considered.

Sub-models

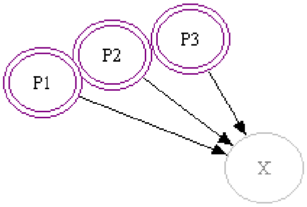
If you have a number of nodes which belong to the same logical group, you may find it helpful to put them into a sub-model. This can greatly simplify the printed and screen display of your model.



In the diagram above, the deterministic nodes are logically grouped. If you select the three “P” nodes and choose Display > Collapse to Sub-Model, the selected nodes will be hidden, and a single, blank hexagonal node created in its place.



The new node represents the *sub-model*; it is possible to name the sub-model as you wish. Even though a sub-model is displayed as a node in the diagram, it has no functionality of its own, except as an organizational tool. Double-clicking on the sub-model node will open a new document window displaying the collapsed nodes.



Sub-models and clones

DATA will automatically create a clone of any node influencing, or influenced by, the sub-model nodes (in this case, X). This clone will be placed in the sub-model. You may also place new clones created in the main influence diagram window inside the sub-model. If there are sub-models in an influence diagram window when you create a clone, DATA will offer to place the clone inside one of the sub-models instead of in the main window.

The position of a sub-model node is not relevant during conversion to tree. If graphical positioning is needed to resolve ambiguities in time-ordering, only the positions of the nodes it contains will be considered.

A sub-model may contain further sub-models. This recursive nesting can be dangerous while building a model, as it is easy to lose track of node locations. Even though the functionality is available, it may be unwise to place sub-models inside sub-models.

The sub-model is not a stand-alone document; it cannot be saved independently of the main model. On the other hand, it is possible to print a sub-model separately. The preference settings for the main influence diagram document apply to all of its sub-models.

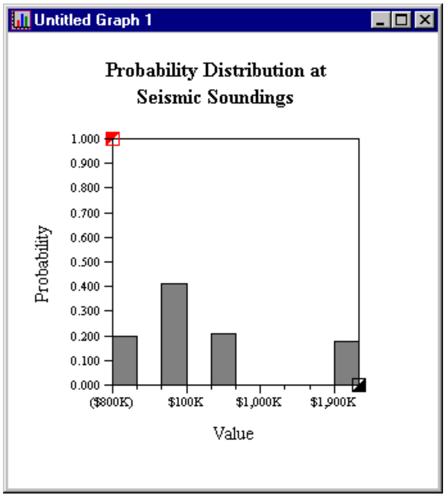
CHAPTER 33 WORKING WITH GRAPH WINDOWS

Each graph created in DATA can be customized. It is also possible to store preferences that govern the display of subsequently created graphs. Both of these features are covered in this chapter, together with a discussion of the custom options that are available in certain types of graphs.

Customizing individual graphs

The information in this section applies to all types of graphs.

There are three kinds of changes that can be made to an individual graph: modifying the text, location, and style of any label; modifying the graph scale, numeric format, and making other numeric changes; and moving or resizing the entire graph. To explore them, a graph is needed:



- ▷ Open the file Oil Drilling #2.
- ▷ Select the Seismic Soundings node.
- ▷ Select Analysis > Probability Distribution.

Making textual changes

DATA has suggested a heading for the graph on your screen: Probability Distribution at Seismic Soundings. DATA can create only one-line graph labels. To create multi-line labels, as here, text is divided among multiple labels. It is possible to edit the contents of text labels, as well as change their fonts.

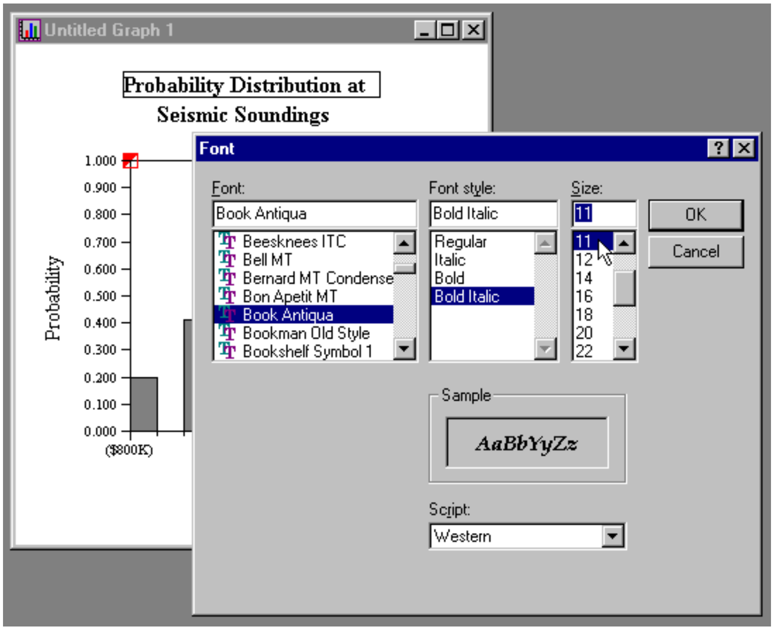
► **To modify a graph label's text and font:**

▷ Click on the first line of the graph heading. A rectangle should appear around the text to indicate that it is selected.

▷ Select Display > Font, or press the Font button on the tool bar.

▷ Choose a different font. You might also want to experiment with changing font size and style.

▷ When you are done, click OK.



Font: Book Antiqua

Font style: Bold Italic

Size: 11

Script: Western

The heading should reflect the changes you just made in the Font dialog.

▷ Click within the rectangle, and drag to change its location.

▷ When you are satisfied with the location and appearance of the heading, click outside of the rectangle to deselect it.

▷ Double-click anywhere within the heading.

▷ In the dialog box, replace the words "Probability Distribution" with `Risk Profile`, and press ENTER or RETURN.

The heading should now read "Risk Profile at Seismic Soundings."

The same technique is used to make changes to any of the labels in the window.

▶ To add a custom label to the graph window:

▷ With a graph window open, choose Graph > New Label.

▷ Enter the text of your new label, and press ENTER or RETURN.

▷ Drag the new label to the desired location.

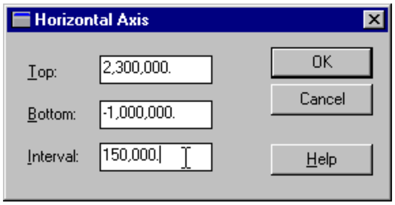
Making numeric changes

▶ To modify the graph's scale:

▷ Double-click on the numbers (values) appearing beneath the horizontal axis. A dialog box will appear in which you can change the scale of the axis.

▷ Type `2,300,000` in the box marked Top.

▷ Press TAB.


Horizontal Axis

Top: 2,300,000.

Bottom: -1,000,000.

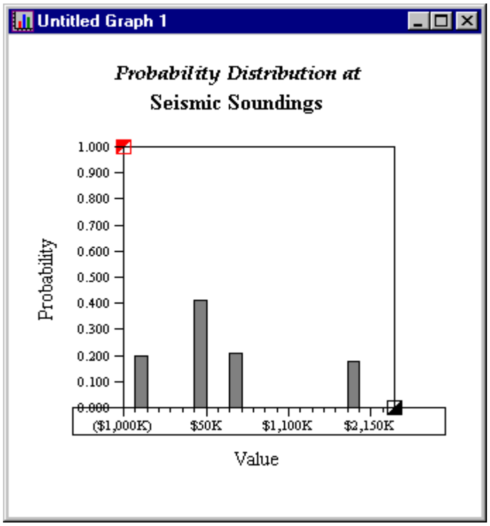
Interval: 150,000

▷ Type `-1,000,000` in the box marked Bottom. You have now changed the upper and lower bounds of the scale.

▷ Press TAB.

▷ Type `150,000` in the box marked Interval.

▷ Press ENTER or RETURN.

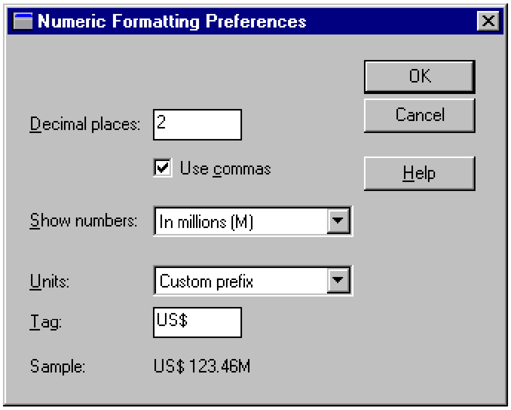


The graph window reappears, but with the changes in scale you just made. There is a limit on the extent to which the scale interval can be decreased. If you attempt to exceed it for any graph, an alert box will let you know.

The size of the interval may affect the number of final outcomes included in any single graph bar. If the interval is relatively large and there are several outcomes whose payoffs differ by less than the amount of the interval, one bar may comprise several outcomes.

Of course, if several payoffs are identical they will always appear together on the graph.

▶ To change the numeric format of a graph’s axis:


Numeric Formatting Preferences

Decimal places: 2

☑ Use commas

Show numbers: In millions (M)

Units: Custom prefix

Tag: US$

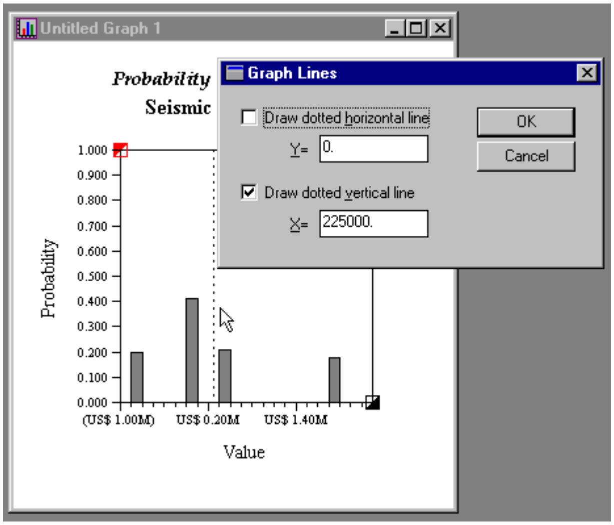
Sample: US$ 123.46M

- ▷ Click on the values associated with the horizontal axis to make the selection rectangle reappear.
- ▷ Select Edit > Numeric Formatting. This will open a dialog box which enables you to change the numeric format of the numbers appearing next to the axis.
- ▷ Type `2`, in order to change the number of digits after the decimal point from 3 to 2.
- ▷ Change Show numbers to In millions (M).
- ▷ From the Units drop-down list select Custom prefix, and type US$ for the tag text.
- ▷ Press ENTER or RETURN.

See Chapter 10 for a complete discussion of changing the numeric formatting of displayed numbers.

It is also possible to change the format of the vertical axis. In the case of probability distribution graphs, however, you may change only the number of decimal places displayed.

For three-way sensitivity analysis graphs, it is possible to change the numeric format of the third axis. The current value of this parameter is displayed in a special label at the bottom right corner of the graph window. After selecting this label, choose Edit > Numeric Formatting to change its format.


Graph Lines

☐ Draw dotted horizontal line

Y= 0.

☑ Draw dotted vertical line

X= 225000.

It can often be helpful to add a dotted horizontal or vertical line as a reference. In this example, a dotted vertical line will be added at 225,000 to represent the mean value of the distribution (which is also the expected value of the Seismic Soundings node).

▶ To display a dotted line on the graph:

- ▷ Choose Graph > Set Lines.
- ▷ Check the button named Draw dotted vertical line.
- ▷ In the box marked X=, type `225,000`, and press ENTER or RETURN.

Moving and sizing the graph

An existing graph can be moved or resized using the handles (small squares) at the corners of the graph.

► **To move or resize an entire graph:**

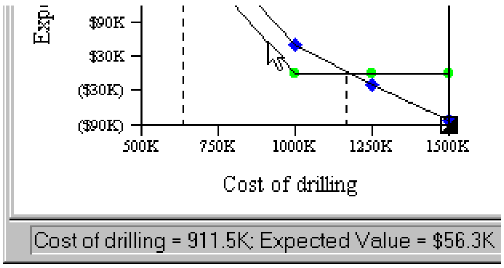
- ▷ Click on the handle on the top left of the graph area and drag to move the graph.
- ▷ Click on the handle on the bottom right of the graph and drag to resize the graph.

Viewing a graph's underlying numbers

Every graph window offers a view into its numeric genesis. Choose Graph > Text Report to view the specific numbers associated with a graph.

See Chapter 17 for details on using the Text Report window. The specific contents of each graph's text report are discussed below.

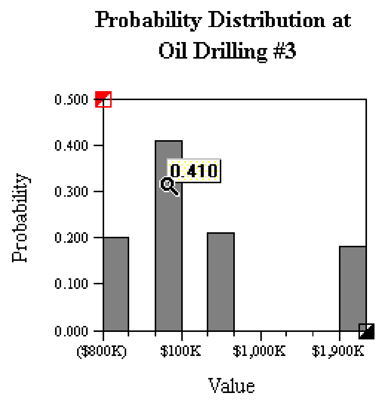
There is another way to view numbers underlying the graph. If you hold down the CONTROL key, the current location of the cursor will be displayed in the status bar.



Cost of drilling = 911.5K; Expected Value = $56.3K

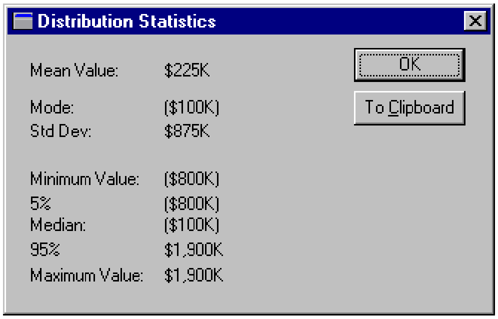
Bar graphs

Probability distributions are displayed as a bar graph. Initially, the distribution will be noncumulative; to convert the graph into a cumulative distribution, select Graph > Cumulative. Reselecting this command will cancel the cumulative display and return the graph to noncumulative format.



In either the cumulative or noncumulative display, it is possible to have DATA report the height of a bar. With your mouse, move the cursor over the bar. The cursor will change to a magnifying glass; hold down the mouse button to display the information.

If a bar is too high for the range set on the vertical axis, a small white arrow will be displayed at the top of the bar.


Distribution Statistics

Mean Value: $225K

Mode: ($100K)
Std Dev: $875K

Minimum Value: ($800K)
5% ($800K)
Median: ($100K)
95% $1,900K
Maximum Value: $1,900K

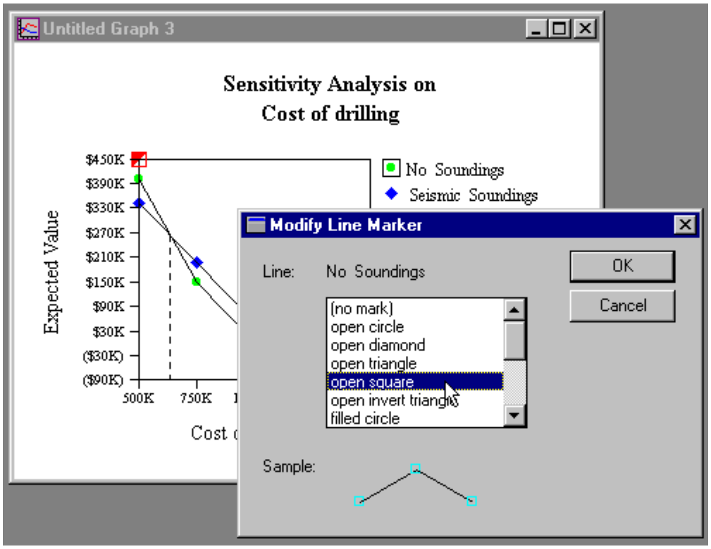
You may view the basic statistics for any bar graph by selecting Graph > Distribution Statistics. The numeric format used for presentation of these statistics is taken from the numeric format of the horizontal axis.

The text report of a bar graph will specify the value and probability of every outcome. In addition, the Notes section of the text report will list the range and height of all bars as currently displayed. The distribution of bars may be affected by the scale of the horizontal axis.

Line graphs

▶ **To change the marker used at each point of a line:**

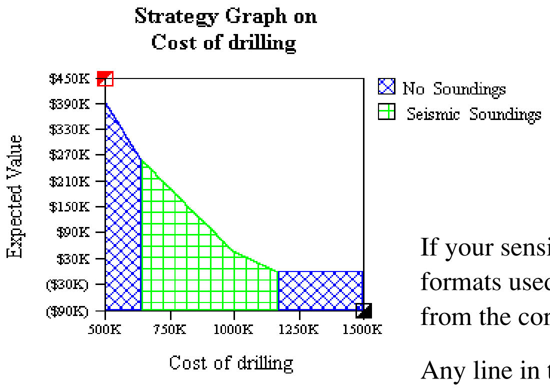
▷ Double-click on the marker as it is displayed in the legend to the right of the graph. Clicking on a marker inside the graph will not work.



Modify Line Marker

Line: No Soundings

▷ Select a marker from the list displayed in the resulting dialog box. Click OK.



If your line graph was generated from a one-way sensitivity analysis, you may convert it to an optimal policy chart, also known as a strategy graph, by using the Graph > Strategy Graph menu item. This graph type displays only the information necessary to show the optimal policy over the range of the analysis.

If your sensitivity analysis displays threshold information, the numeric formats used to indicate the location of optimal path changes are taken from the corresponding axes.

Any line in the graph may be converted into a table for subsequent use in formulas. Select the Graph > Line To Table command, and enter the table properties. See Chapter 26 for more information on tables.

You may hide individual lines in the graph. Select Graph > Show/Hide to select which lines should be displayed.

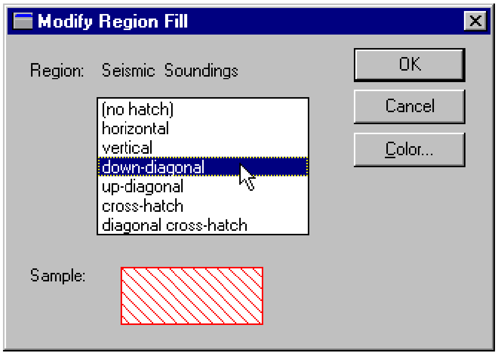
The text report for a line graph will include the numeric data twice. In the upper part of the text report, the information will be presented vertically to allow for easy viewing of long-run analyses. In the bottom part of the text report, the information will be presented horizontally to facilitate proper export into graphing programs.

Region graphs

Region graphs are generated from two- and three-way sensitivity analyses. A strategy graph, generated from a one-way sensitivity analysis, is also a type of region graph.

► **To change the pattern used to fill each region:**

- ▷ Double-click on the marker as it is displayed in the legend to the right of the graph. Clicking on a region inside the graph will not work.



Region: Seismic Soundings

(no hatch)
horizontal
vertical
down-diagonal
up-diagonal
cross-hatch
diagonal cross-hatch

Sample:

- ▷ Select the desired hatch pattern from the list displayed in the dialog box, or select “no hatch” for a solid region.
- ▷ To change the color used for drawing regions, press the Color button and choose from the available palette of colors.
- ▷ Press ENTER or RETURN.

A strategy graph has no text report; that information is available from the text report for the line graph from which the strategy graph was generated. See Chapter 22, Advanced Sensitivity Analysis, for important information regarding two-way and three-way sensitivity analysis graphs.

> **TIP:** DATA uses a proprietary format to display and print graphics, including region graphs. You may experience difficulty printing region graphs from DATA on some computers. The solution may be to convert the graph to a standard graphics format, such as a bitmap or Metafile, using DATA's File > Snapshot command. You can also try the following: reducing the printer's resolution (say, from 600 to 300 dots/inch); reducing the size of the graph or the range of analysis; or changing or removing the color and hash marks.

Tornado diagrams

In a tornado diagram, the process for changing the pattern used to fill each bar is identical to that described above for region graphs.

When you initially generate a tornado diagram, a vertical dotted line is drawn to indicate the expected value at the selected node.

To hide the display of individual bars in the graph, use the Graph > Show/Hide command.

Cost-effectiveness graphs

The process for changing the marker used for each strategy in a cost-effectiveness graph is identical to that described above for line graphs. The contents of the text report for a cost-effectiveness graph are described in Chapter 21, Cost-Effectiveness Analysis.

Storing preferences for future graphs

You may store information about the appearance of a graph using a graph template. By applying a template to a new graph you can easily maintain consistency of fonts, numeric formatting, and even text.

Each graph template contains the size of the graph area. In addition, you may optionally store the following information:

What a graph template contains

- **Fonts** – This will store the fonts used for each item in the graph.
- **Numeric Formats** – The numeric formats of both axes may be stored.
- **Title Texts** – Many graphs have two title lines. You may choose to store one or both titles. By default, the top line of a bar graph will read “Probability Distribution at.” If, for example, you want it always to read “Risk Profile at,” store the first title line but not the second. In this case, the second title will be created on-the-fly, as usual.
- **X / Y Axis Labels** – You may save either or both text labels in a template.
- **Custom Labels** – Any labels which you have added with the New Label command in the Graph menu may be stored. Their positions and fonts will also be stored. If you have not created any new labels, this option will not be available.
- **X / Y Lines** – If you have elected to draw a custom line in your graph (in the Set Lines dialog), you may elect to have the line(s) appear in new graphs. This option is not available if no custom lines have been drawn.

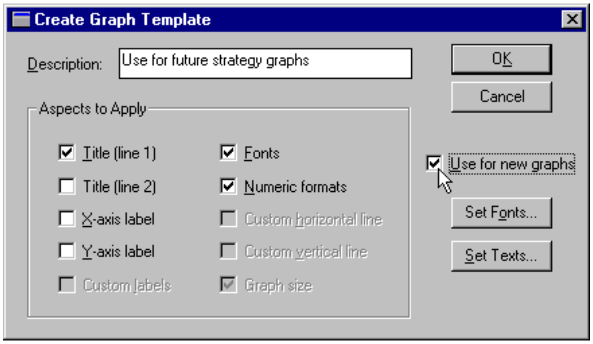
Creating a graph template

► **To create a new graph template:**

▷ Open an existing graph, or create a new one.

▷ Modify the graph to create the appearance you wish to duplicate.

▷ Choose Graph > Create Template.



Create Graph Template

Description: Use for future strategy graphs

Aspects to Apply

☑ Title (line 1)
☐ Title (line 2)
☐ X-axis label
☐ Y-axis label
☐ Custom labels
☑ Fonts
☑ Numeric formats
☐ Custom horizontal line
☐ Custom vertical line
☑ Graph size
☑ Use for new graphs

▷ Give the template a description to be used only for your own references.

▷ Check the appropriate boxes in the list to indicate those items you wish to save with the template. (In actuality, all items are stored with the template. This list stores those items which will be *applied* when the template is used.) You may later return to change these flags.

▷ Click OK.

Creating a default template

The *default* template is applied to every new graph. Unlike normal custom templates, the default template will only store font and graph size information. To set a template as the default template, select the Use for New Graphs button.

Applying a graph template

To apply a template to an existing graph, choose Apply Template from the Graph menu when the graph window is in front. Select the template from the list of stored templates.

The attributes which were saved with the template will immediately be applied to the open graph.

Modifying an existing graph template

With a graph window in front, Choose Graph > Maintain Templates. Select the template from the list of stored templates, and press the Properties button. You may then change which flags are stored with the graph, and the specific fonts and texts used. You may also indicate that a template should be used for new graphs (the default template).

You may not directly modify the default template. To change the fonts and graph size used for new graphs, you must create a new default template, as described above.

Using graph templates with stored analyses

One of the most useful features of templates is that they can automatically be applied when a graph is created from a stored analysis. See Chapter 13 for a full description of this feature.

CHAPTER 34 MISCELLANEOUS ADVANCED FEATURES

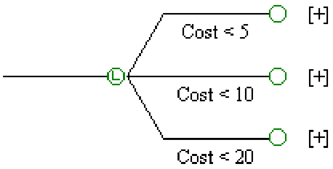
Logic nodes

A logic node acts like a decision node, in the sense that an optimal path is selected and the remaining branches emanating from the logic node are ignored. The optimal path is chosen by DATA based on logical criteria that you specify. These criteria, in the form of logic expressions, are specified below the branch line (where a probability would be entered if the logic node were a chance node).

A logic node selects the optimal path as follows:

- The logic expressions of each branch are evaluated in turn, from top to bottom.
- A true logic expression is assigned a value of 1 and a false expression is assigned a value of 0.
- The first branch whose logic expression evaluates to nonzero (and non-negative) is chosen.
- If all expressions evaluate to zero or negative, the bottom branch is chosen.

TIP: Don't be confused by numeric values below the branch lines emanating from a logic node. A logic node acts more like a decision node than a chance node, in that there is no requirement that values below the branches add to one. As in a chance node, though, the hash mark (#) can be used below a single branch. Its value will be one minus the other branches' values.



More than one branch of a logic node may evaluate to nonzero. When this occurs, only the first (starting at the top) is chosen. In the subtree to the left, if Cost is less than 5, the probability expression of each branch will evaluate to nonzero, so branch one (the topmost) will be chosen. If Cost is greater than or equal to 5 and less than 10, branch two will be chosen. If Cost is greater than or equal to 10, branch three will be chosen.

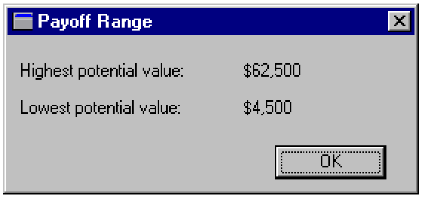
In this example, Cost can be defined at the logic node or at a node to its left. Sensitivity analysis can be carried out on Cost. Logic nodes have no recursive functionality; for recursive operations, a Markov node must be used. However, Cost may be a Monte Carlo tracking variable; see Chapter 29 for details.

Identifying the range of potential payoffs

Several important analysis options make it possible to view measures of the spread of potential payoffs. For example, the Standard Deviation and Probability Distribution commands provide information that nicely supplements the expected value information provided by roll back. DATA also includes tools for looking at other measures related to expected values.

Payoff range

You may view the minimum and maximum payoffs by choosing Analysis > Payoff Range. This analysis will tell you the highest and lowest potential value which may occur from the selected node in your tree.


Payoff Range

Highest potential value: $62,500

Lowest potential value: $4,500

In computing these values, DATA assumes that the decision maker will follow its advice, by choosing alternatives at every decision node in accordance with maximizing (or minimizing) expected value. Therefore, payoffs at terminal nodes that would never be reached by such a decision maker are ignored in computing the payoff range.

Over/under

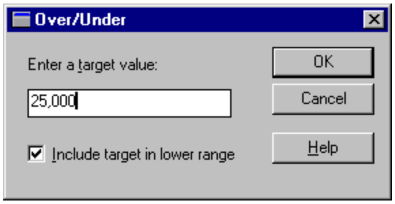
This analysis calculates the probability of an outcome having a value over the target, and the probability of an outcome under the target.

► **To calculate the over/under probabilities:**

▷ Select a node in your tree whose potential value interests you.

▷ Choose Analysis > Over/Under.

▷ Enter a target value. Potential payoffs above the target will be separated probabilistically from potential payoffs at or below the target.

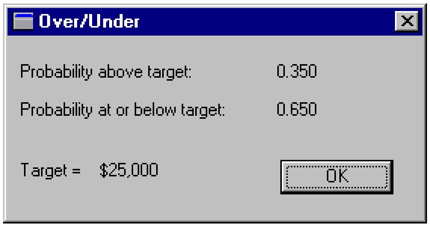

Over/Under

Enter a target value: 25,000

☑ Include target in lower range

▷ Indicate whether payoffs which exactly match the target value should be included in the “under” (lower) range. If you do not select the check box, those payoffs will be included in the “over” (upper) range.

▷ Press ENTER or RETURN.



Probability above target: 0.350

Probability at or below target: 0.650

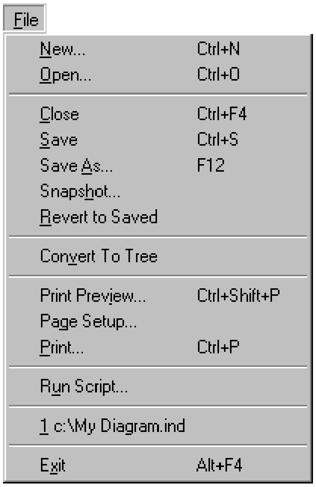
Target = $25,000

As with the Payoff Range analysis described above, DATA includes only those outcomes that will be reached if the decision maker follows the recommended strategy at each decision point.

Appendix A Menu and Tool Bar Reference

A number of menu items will display different text depending on context; thus, the contents of a menu may vary from the pictures shown in this Appendix. Also, some commands may be enabled or disabled differently, depending on context. Menus in DATA 3.5 for Macintosh will also differ in certain commands and keyboard equivalents.

File menu



New…

Creates a new document — either a tree or influence diagram. This option allows you to build a new model from scratch.

Open…

Presents the Open file… dialog box for you to open an existing model or graph.

Close

Closes the active window. If changes have been made to the document since the last time it was saved, choosing Close will be followed by a dialog giving you the option of saving, not saving, or canceling the close request.

Save

Saves the document in the currently active window. If a file name has been specified, the document is saved under that name; otherwise you must enter the file name under which it should be saved.

Save As…

Same function as Save, but you are prompted for a file name, whether or not one has already been specified.

Snapshot…

Allows you to export your document as a picture (metafile, bitmap, or PICT) for use in a word-processor or presentation program. You may also use the Snapshot command to create a TRB file, which is a permanently rolled-back picture of your tree, viewable only in DATA.

Revert to Saved

Reverts to the most recently saved version of the document in the active window. This option should be used if you want to eliminate all changes made to a document subsequent to the last time the document was saved.

Convert

If an influence diagram is in the active window, this command converts it into a tree.

Print Preview…

Displays the document in the active window, indicating the location of page breaks and the number of pages required to print it (in accordance with the options selected under Page Setup). Buttons available in the preview window permit direct access to Page Setup and Print.

Page Setup…

Displays the standard Page Setup dialog box for the particular printer you are using.

Print…

Prints the document in the active window.

Run Script…(Windows only)

Allows you to run a text file containing commands in the DATAScript language. This is only for compatibility with scripts written for DATA 3.0.

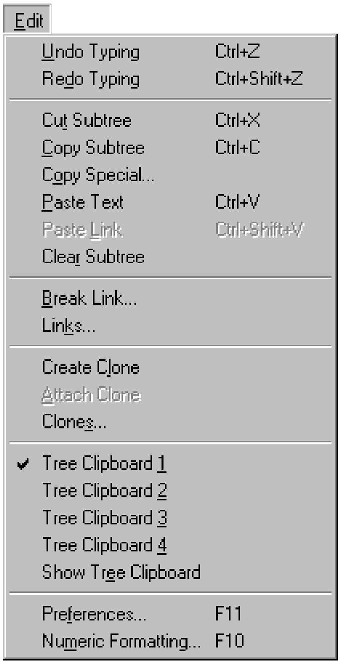
(File Names)

DATA displays a list of your most recently opened files. Selecting one from the list will reopen that document, so long as its name and location have not changed since the last time it was opened.

Exit (Windows) / Quit (Macintosh)

Exits (quits) the application, after offering the option to save any changes made to documents open when this command is given.

Edit menu



Undo

This command permits you to reverse (undo) your most recent action if, for example, you issued the wrong command or made a mistake in typing or tree construction. In the case of typing or text formatting, this command will reverse multiple, consecutive changes. See Chapter 14 for more information.

Redo

Re-executes the most recently undone action.

Cut

Cuts the selected portion of the active window onto the clipboard. This can result in one of several different actions, depending on the context. For example, if a tree window is active, the Cut command will read, depending on the context, "Cut Node" (when no branches emanate from the selected node), "Cut Subtree," "Cut Note," "Cut Text." When it is not clear whether text or a portion of the tree is to be affected, the command will read "Cut..." and will be followed by an appropriate dialog box. A node (together with the branch leading to it) or a subtree is cut to the active tree clipboard (see below).

If an influence diagram window is active, you may cut one or more nodes and the arcs connecting them. Note that clone nodes may not be placed on the clipboard.

If a graph window is active, the selected text is cut to the clipboard. There are separate clipboards for nodes and text. Therefore, cutting or copying an item of one type does not always remove an item of other types from the clipboard.

Copy...

Copies the selected portion of the active window onto the appropriate clipboard. With limited exceptions, it has the same functionality as the Cut command, described above. The principal difference is that the Copy command does not delete the selected material from the active window, but simply places a copy of it on the clipboard.

Copy Special... / Copy as PICT

Copies the selected portion of the active window onto the clipboard in a format other than the standard DATA format. All of DATA's main documents (trees, influence diagrams, and graphs) will allow you to copy the document in bitmap and metafile formats (Windows) or in PICT format (Macintosh). See Chapter 17.

When a tree window is active, you may copy a value calculated at the selected node as a DDE link (Windows only). When a graph window is active, you may copy the graph data as spreadsheet-accessible text. This functionality is also present in the Graph > Text Report dialog. See Chapter 15.

When a tree is active and rolled back, and custom columns are displayed at end nodes, this menu command will copy the columns to the clipboard. The text may then be pasted into a text document or spreadsheet. See Chapter 10.

Paste

The inverse of "Copy," this command copies the contents of the clipboard to the selected point in the active window. For example, if a tree window is active and there is a subtree on the active tree clipboard, the subtree is pasted onto the selected node(s). Tree clipboards are maintained separately from all other types of clipboard contents. When the selection is unambiguous, DATA will automatically paste the item of the appropriate type into the active window. When it is unclear whether text or a tree component is to be pasted, a dialog box will appear asking the user to make the selection.

Paste Link (Windows)

Allows you to paste an item, which is dynamically linked to a spreadsheet, database or other tree, into the active tree. Paste Link is only available when the cursor is in a value field (such as a variable definition or node probability) in a DATA Tree Window. (Choosing this menu item eliminates the step of opening the Links dialog before pasting the link; see below.) See Chapter 15 for more information.

Clear

Deletes the selected portion of the active window altogether. It has the same functionality as "Cut," but the material being cleared is deleted without being transferred to the clipboard, and any material on the clipboard is unaffected.

Links… (Windows)

Manages the sharing of data between trees or from another application to a tree, using bi-directional ActiveX links or Dynamic Data Exchange (DDE). The Links dialog is an alternative to the Paste Link method, described above, of creating client links in a tree. Incoming items are listed in this dialog by their index numbers. Selecting an index number allows you to paste or cancel the item associated with it and edit information about the source and contents of that item. The Bi-directional Links dialog is accessed here; see Chapter 16.

Break Link… (Windows)

Destroys the DDE server link(s) to calculated values at the selected node.

Publishers… (Macintosh)

Creates an edition file containing information which can be shared (subscribed to) by another tree file or influence diagram, or by another application. When a tree is in the active window with a node selected, choosing this command will allow you to publish the expected value, path probability, or standard deviation of the subtree rooted at the selected node. The edition file will be updated whenever the subject tree is rolled back.

Subscribe To… (Macintosh)

Subscribes to an edition file containing information published by a tree file or by another application. The subscription can be utilized in connection with defining a probability or payoff. Updating the edition file will cause any definitions subscribing to it to be updated.

Subscriber List… (Macintosh)

Displays a list of the indices of all of the subscribers to an edition file being utilized by the active tree or influence diagram. Selecting an index results in displaying the name and value of the related subscriber and allows you to cancel the subscription. See Chapter 15.

Create Clone…

Designates the selected subtree as a clone master, which may be replicated at different points throughout the tree. Changes subsequently made to the clone master will “ripple through” the tree, affecting all copies of that clone. When a clone master is selected, this command will be entitled “Destroy Clone.” See Chapter 12 for more information on clones.

This command is also used to create (and destroy) clones in an influence diagram window. See Chapter 32 for more information.

Attach Clone…

Attaches to the selected node a dynamically-linked copy of a clone master, which is designated by the user from a list of all clone masters previously created in the active tree. When a clone copy is selected, this command will be entitled “Detach Clone.” See Chapter 12.

Clones…

Allows you to manage a list of clones in a tree. You may rename or renumber your clones. You may also destroy clones from this dialog, rather than from the Destroy Clone menu item.

Tree Clipboards 1…4

For convenience in working with large trees, there are four distinct “tree clipboards” which can be used in cutting, copying, and pasting subtrees. Tree clipboards are maintained independently of other types of clipboard contents, such as text, annotation boxes, or influence diagram nodes.

The clipboard with the checkmark (“✓”) next to it is designated the active clipboard; when a tree clipboard is not empty, the type of content is designated in parenthesis. For example, when a tree is in the active window and a subtree is selected, choosing “Cut,” “Copy,” or “Paste” invokes the active clipboard. Thus, cutting a subtree with Tree Clipboard #1 active puts the subtree on Tree Clipboard #1, and pasting with Tree Clipboard #2 active pastes the contents of Tree Clipboard #2 into the tree. By switching the active clipboard (accomplished by selecting the menu item), you can keep up to four commonly appearing subtrees in clipboards from which they can be pasted at will.

Switching the active clipboard does not make any immediate or automatic changes to either your model or the contents of the four clipboards.

Show Tree Clipboard

Displays the contents of the currently active tree clipboard. No editing may be performed in the Tree Clipboard window.

Preferences…

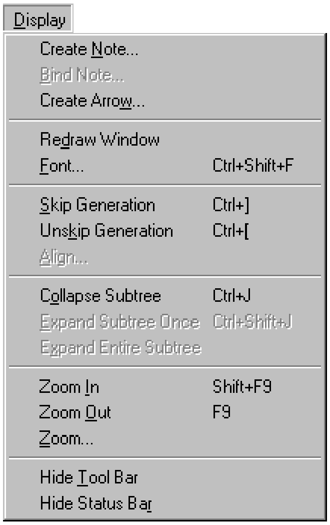
Displays the main Preferences dialog box. See Appendix B for more information.

Numeric Formatting…

Allows you to change the presentation of numeric values in trees and graph windows. See Chapter 10 for more information.

Display menu

Create Note…



Allows you to annotate a tree or an influence diagram. After choosing this command, a rectangular box can be drawn in the window and the annotation typed within it. The box containing the annotation can be moved or resized. Notes can be cut, copied and pasted (see the Edit menu, above). See Chapter 10.

Bind Note…

Allows you to permanently link an annotation box to a particular node in your tree. Notes which are not bound remain floating above the tree, and are not automatically moved when the tree is resized.

Create Arrow…

Allows you to draw an arrow in your tree, as an aid to annotation.

Redraw Window

Redraws the active window to eliminate blank areas and other problems with screen display.

Font

Allows you to choose the font, size, and style in which any selected text will be displayed and printed. Note that you cannot have different fonts, sizes or styles within the same text item. In other words, a node label must be all in one font, size, and style; likewise, an outline must be all in one font, size, and style.

Skip Generation

Forces the selected nodes to display as if they spanned an extra generation. The text will remain at the left edge, while the node's symbol will move to the right so that it lines up with the symbols of the next generation. The primary use of this feature is in asymmetrical trees where it is desirable to have all events within a given stage or time frame displayed in vertical alignment. A node may be made to skip as many generations as desired. See Chapter 10.

Unskip Generation

Negates the effect of a single Skip Generation command when performed on one or more *skipped* nodes.

Align…

Allows you to align two or more nodes in an influence diagram. See Chapter 31.

Collapse Subtree…

Hides the subtree emanating from the selected node. This is useful when working with large trees, when focusing an audience on a particular location in your tree, or when viewing custom columns at endnodes during roll back. See Chapter 12.

Expand Subtree Once

Expands the collapsed subtree at the selected node. Only the first generation will be displayed; the subtrees emanating from the nodes in the first generation will remain collapsed.

Expand Entire Subtree

Reverses the effect of Collapse Subtree.

Zoom In

Displays your document at a larger magnification.

Zoom Out

Displays your document at a smaller magnification.

Zoom…

Allows you to set the magnification factor for the active window. See Chapter 10.

Show Tool Bar

Shows the tool bar at the top of the screen. Icons on the tool bar allow you to issue various commands with a single mouse click. These include opening, saving, printing, rolling back, changing node type, and changing the function of cursor keys. When the tool bar is visible, this command reads “Hide Tool Bar.”

A full tool bar reference is included at the end of this appendix.

Show Status Bar

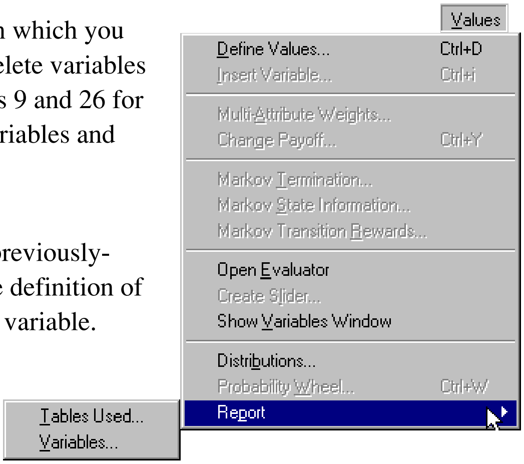
Displays, at the bottom of the screen, a one-line explanation of the selected menu or tool bar item. In addition, the status bar displays some useful information about the active tree. When the status bar is visible, this command reads “Hide Status Bar.”

A full status bar reference is included at the end of this appendix.

Values menu

Define Values...

Displays a dialog box in which you can create, modify or delete variables and tables. See Chapters 9 and 26 for more information on variables and tables, respectively.



Insert Variable

Allows you to insert a previously-created variable into the definition of a probability or another variable.

Multi-Attribute Weights…

Allows you to set the formula used for generalized multi-attribute calculations. See Chapter 20.

Change Payoff…

Allows you to change the payoff (variable, expression, or value) associated with the selected terminal node(s).

Markov Termination…

Allows you to specify the condition(s) upon which a given Markov process will cease to cycle.

Markov State Information…

Accepts values for the initial, incremental, and final rewards associated with a given state of a Markov process. You may also indicate that the selected state should be an automatic tunnel state.

Markov Transition Rewards…

Assigns a reward to a non-state node in a Markov process.

Open Evaluator

Displays the Evaluator dialog box, which allows you to enter an expression whose value you wish to calculate at the selected node. Calculations are performed based on the values of the variables in effect at the selected node.

Create Slider…

Allows you to edit manually the value of a variable at a given node. After choosing a variable and specifying a range, drag the sliding “thumb” to change temporarily the value of the variable. The changed value will remain in effect until the slider is closed, or until a sensitivity analysis is run on the chosen variable.

Show Variables Window

Displays the list of variables which have been defined at a selected node. Node selection can be changed by using the arrow buttons in the Variables Window. Double-clicking in the Variables Window on a variable you wish to redefine brings up the Define Variable window for that variable at the then selected node. When the Variables Window is displayed (whether or not it is the active window) this command will read “Hide Variables Window.”

Distributions

Allows you to assign parameters to an analytic distribution for use in a payoff formula or probability. Distributions are used primarily for sampling during Monte Carlo simulation.

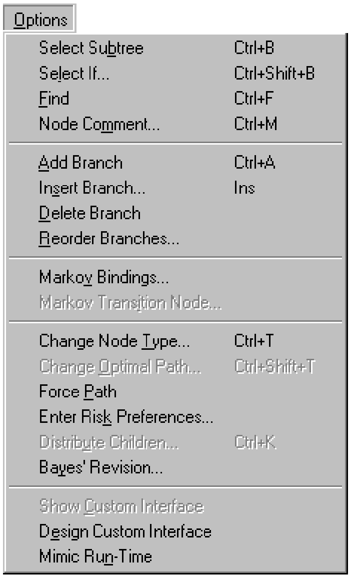
Probability Wheel

Displays the probability wheel at the selected chance node. The wheel is used to assign subjective probabilities to an uncertainty.

Options menu

Select Subtree

If the currently selected node has branches, this option causes the entire subtree rooted at this node to be selected. This is useful in cutting or copying a subtree, which requires selection of the entire subtree. It can also be used to select the tree (i.e., the subtree emanating from the root node), prior to copying the tree to the clipboard, either in DATA format or for export in bitmap/metafile format (Windows) or PICT format (Macintosh).



Select If…

Allows you to select nodes using a rule. For instance, you may select all terminal nodes, or all nodes at which a particular variable is defined. See Chapter 11.

Find…

Searches for (and replaces, if desired) specified text in the tree window. See Chapter 14.

Node Comment…

Allows you to assign an annotation to the branches of the selected node. This feature is often used to record the reasons underlying probability assignments. See Chapter 14.

Add Branch(es)

Adds two branches to the currently selected node. (You may change the default number of branches in the Preferences dialog.) If the selected node already has branches, this command will be entitled “Add Branch” and will add one additional branch to the selected node. A chance node is automatically added at the end of each new branch; you may later change the node type.

Insert Branch…

Inserts a new node to the left, to the right, above, or below the selected node. See Chapter 5 for more information.

Delete Branch…

Deletes the selected branch from the tree, and attaches any branches emanating from the deleted branch to its parent. See Chapter 5.

Reorder Branches…

Allows you to change the top-to-bottom ordering of the branches emanating from a node. See Chapter 5.

Markov Bindings…

At the selected node *within a Markov subtree* (including the Markov node), allows you to associate a state name variable with a Markov state. These *Markov bindings* can be used in place of a Markov state name when assigning the transition for a Markov transition node (see below). Markov state bindings are useful when cloning within Markov subtrees. See Chapter 27.

Markov Transition Node…

For the selected Markov transition node, allows you to choose a from a list of existing Markov states (within the Markov process) and Markov state bindings (to the left of the transition node). See Chapter 25.

Change Node Type…

Presents a dialog box which allows you to change the type of the selected node(s).

In the tree window, there are six types of nodes: decision nodes (squares), which indicate a decision to be made; chance nodes (circles), which indicate an event over which the decision maker lacks complete control; terminal nodes (triangles), which indicate a final outcome or endpoint; Markov nodes (circle with an “M” inside), which indicate the root of a Markov subtree used to simulate a recursive process; logic nodes (circle with an "L" inside), which act as a special type of decision node; and label nodes (zigzag line), used only as a placeholder. If "terminal" is chosen, a dialog box appears for entering one or more payoffs to be associated with the scenario determined by the path between the root node and the selected node.

In the influence diagram window, the Change Node Type command is found under the Diagram menu; see below.

Change Optimal Path...

When one or more decision nodes are selected, allows you to change the optimal path for the selected decision node(s) to be the opposite of the default for the rest of the tree. See Chapter 14 for more information.

Force Path

This option allows you to indicate the occurrence (or inevitability) of a particular event at a chance node, or of a commitment to a specific alternative at a decision node. DATA will change the selected node to a logic node and set the logical expression of the specific event or alternative to 1 and the logical expressions of the other branch(es) to 0.

Enter Risk Preferences...

Accepts information about the risk aversion of the decision maker for application of a risk preference curve to the decision at hand. See Chapter 30.

Distribute Children...

Approximates a continuous distribution of values by creating a specified number of branches and assigning a particular value to a variable at each of those branches, according to a distribution which you chose and parameterize. See Chapter 28.

Bayes' Revision...

Uses Bayes' theorem to revise probabilities in a subtree to reflect knowledge gained from an imperfect predictor of an uncertainty. The *a priori* and likelihood probabilities are automatically converted to the probabilities needed to make a decision based on the outcome of the imperfect test. All values can be preserved as variables for purposes of sensitivity and threshold analysis. See Chapter 24.

Show Custom Interface...

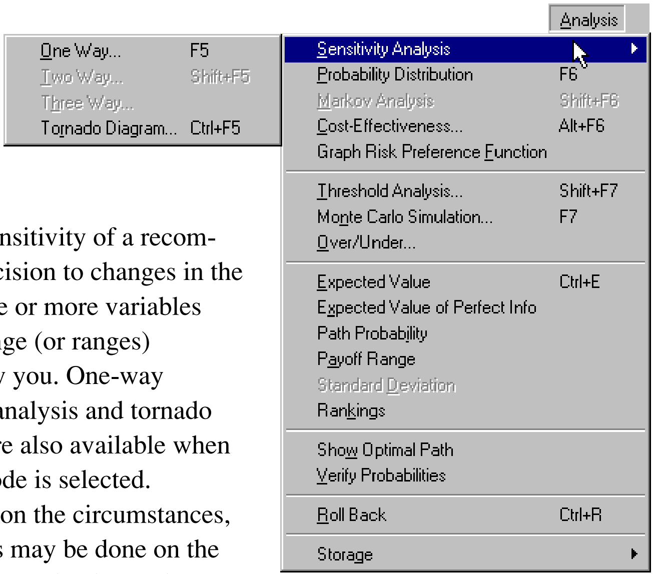
Displays the window associated with the Basic Custom Interface, as described in Chapter 13. When the Custom Interface is shown, this command will read "Hide Custom Interface."

Design Custom Interface…

Allows you to design a basic or extended Custom Interface. The Custom Interface is useful for sharing models with less sophisticated users. See Chapter 13.

Mimic Run-Time

Causes the full version of DATA to emulate the run-time version in certain respects. This will affect menu items and preference settings, many of which are not available to run-time users. Use this command to test an Extended Custom Interface tree.



Analysis menu

Sensitivity Analysis…

Tests the sensitivity of a recommended decision to changes in the value of one or more variables across a range (or ranges) specified by you. One-way sensitivity analysis and tornado diagrams are also available when a chance node is selected. Depending on the circumstances, calculations may be done on the basis of expected value and marginal value.

See Chapters 6 and 22 for more information.

Probability Distribution / Comparative Distributions

Draws a bar graph that displays the distribution of payoffs in terms of their probability of occurrence. The graph depicts how closely clustered or widely distributed are the outcomes, as well as the probability that payoffs within a certain interval will occur.

If multiple nodes are selected, this option reads Comparative Distributions. Selecting this item produces probability distributions for each selected node, and displays the distributions together (in cumulative format) on a single line-graph. The expected value at each of the selected nodes is also displayed, using a dashed line extending from the horizontal axis upward until it meets the graph line.

See Chapters 6 for more information on these analyses.

Markov Analysis…

Allows you to graph, for a given Markov process, many of the quantities which vary over the course of the process. You may also view a full “trace” of the process, with all values displayed in a table. See Chapters 25 and 27.

Cost-Effectiveness…

For a selected decision node in a cost-effectiveness tree, displays each alternative in a graph with increasing cost to the right and increasing effectiveness toward the top of the graph. This analysis will also show dominance and extended dominance, and all marginal values are available via the Graph > Text Report command. See Chapter 21.

Graph Risk Preference Function

Graphs the currently active risk preference function as a line graph; see Chapter 30.

Threshold Analysis…

Searches more thoroughly and accurately for threshold information in connection with a single variable than does a one-way sensitivity analysis. The result of this analysis is a detailed, textual description of how the optimal strategy is affected by changing the value of a single variable across a designated range. See Chapter 22.

Monte Carlo Simulation…

Performs a Monte Carlo-style simulation at the subtree rooted at the selected node. See Chapter 29.

Over/Under…

Allows you to specify a target value at a selected node, and then calculates the probability of an outcome having a value over the target, and the probability of an outcome at or under the target. See Chapter 34.

Expected Value

Displays the expected value of the subtree rooted at the selected node. If this node is a chance node, the calculated value is the probabilistic expected value (the average value weighted by the probabilities) of the values of the branches departing from the selected node. If the selected node is a decision node, the result will be the maximum (if high optimal path is selected in Preferences) or the minimum (if low optimal path is selected) of the values in the branches departing from the selected node. If the selected node is a terminal node, the result will be the value of the payoff assigned to that terminal node.

If multiple nodes are selected, DATA will calculate and display the sum of the expected values at these nodes.

For exporting the expected value to another tree or a different application, see Chapter 15 for a information on using DDE (Windows) or P&S (Macintosh).

Expected Value of Perfect Information

Calculates the maximum value of a perfect test to determine the outcome of an uncertainty. See Chapter 23.

Path Probability

Displays the probability of reaching the selected node, that is, the probability that the scenario represented by the path between the selected node and the root node will occur. This command can be used to determine the probability of the path to any node in the tree, with the exception of the root node (since there would be no path).

If multiple nodes are selected, DATA will calculate and display the sum of the path probabilities.

Note that this command does not assume you will follow DATA's recommended optimal path, unless you select it while roll-back is turned on.

For information on exporting the path probability, see Chapter 15.

Payoff Range

Determines the absolute, unweighted highest and lowest potential payoffs in the subtree rooted at the selected node. See Chapter 34.

Standard Deviation

Calculates the standard deviation of the potential outcomes at a selected chance node. The standard deviation is weighted by the probabilities of the branches. It indicates the extent of dispersal, around the expected value at the selected node, of the values in the branches departing from that node. This gives an indication of the risk involved in the subtree rooted at the selected chance node.

For information on exporting the standard deviation, see Chapter 15.

Rankings…

Displays the alternatives associated with a decision node, ranked in order of optimality. See Chapter 34.

Show Optimal Path

Specifies the branch emanating from the selected decision node which represents the best choice that can be made. This information is also displayed when you select Analysis > Expected Value at a decision node.

Verify Probabilities

This command is used to check every chance node in the active tree to determine whether all sets of probabilities sum to 1.0. An appropriate message specifying the location of any error, or indicating a successful verification, appears at the end of the verification process.

Roll Back

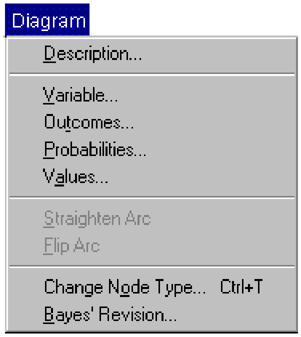
This option does all the basic calculations on the active tree and causes the results to be displayed in the tree window. See Chapters 6 and 19 for more information on roll back and changing what the tree calculates, respectively.

Choosing the Roll Back option again returns the tree to its normal display, and the information described above disappears.

Storage…

Enables you to store the parameters of an analysis that you just ran, recall and rerun a stored analysis, or delete or rename a stored analysis. See Chapter 13.

Diagram menu (influence diagram window)



Description…

Allows you to annotate a node in an influence diagram.

Variable…

Makes it possible to rename the variable which holds the value for a node in an influence diagram. The variable may then be used in a payoff formula at a value node. See Chapter 31.

Outcomes / Alternatives…

This command is used to enter a set of outcomes (for a chance node) or alternatives (for a decision node) in an influence diagram. Its name will change depending on the type of node selected.

Probabilities…

Enables you to assign the conditional probability distribution(s) associated with the outcomes of a chance node in an influence diagram.

Values…

Makes it possible to assign the conditional value distribution(s) to a chance or decision node in an influence diagramfor use in a payoff formula. This menu item can also be used to enumerate payoff values for a value node.

Straighten Arc

If an arc in an influence diagramhas been curved (by dragging the black selection handle), this command can be used to straighten it.

Flip Arc

Changes the direction of the selected arc in an influence diagram. This operation is particularly useful for performing EVPI calculations, as described in Chapter 32.

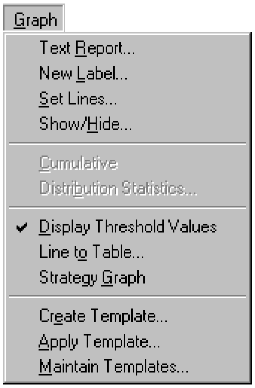
Change Node Type…

Enables you to change the type of a node in an influence diagram. There are four types of representative nodes: decision nodes (squares), which indicate a decision to be made; chance nodes (circles), which indicate an event over which the decision maker lacks complete control; value nodes (diamonds), which must be used for the model’s final outcome and may be used to create an intermediate formula; and deterministic nodes (circle with double outline), which are used to specify a parameter having a fixed value and, optionally, a value range for purposes of sensitivity analysis.

Bayes’ Revision…

Identifies, in a text report, any nodes at which Bayes’ revision will be performed when the influence diagram is converted into a tree.

Graph menu



Text Report…

Displays the numerical data which underlie the graph in the active window. These data may then be exported for further analysis or graphing in a spreadsheet, statistics, or database program. See Chapter 33.

New Label…

Adds a new, custom label to the active graph.

Set Lines…

Allows you to indicate a dotted horizontal or vertical line in the active graph. This is useful for emphasizing a particular threshold value, such as a significant expected value or cost constraint.

Show/Hide…

Allows you to hide certain pieces of information from the selected graph. You may hide lines in a line graph or bars in a tornado diagram.

Cumulative

Causes the probability distribution in the active window to be displayed cumulatively, so that each bar represents the probability that the amount to be spent or received will be that value *or less*. Unchecking this option restores a noncumulative display.

Distribution Statistics…

Displays the basic statistical values associated with a distribution graph.

Isocontours…

Allows you to set lines of equal marginal value in a two-way sensitivity analysis region graph representing two alternatives. See Chapter 22.

Display Thresholds

When a one-way sensitivity analysis graph is in the active graph window, displays information about thresholds (break even points) next to the graph. When threshold information is visible, this item reads “Hide Thresholds.”

Display Extended Dominance

When a cost-effectiveness graph is active, displays (or hides) indications of extended dominance and the associated coefficients of inequity.

Line to Table…

Converts a line in an active line graph into a table.

Strategy Graph

Graphs the optimal frontier of a one-way sensitivity analysis graph.

Create Template…

Stores the layout of the active graph window for later use in other graphs. See Chapter 33.

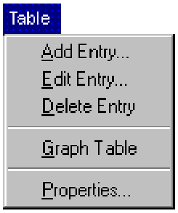
Apply Template…

Applies an existing graph template to the active graph window.

Maintain Templates…

Allows you to edit the information stored in specific graph templates.

Table menu



Add Entry…

Adds a new entry to the open table.

Edit Entry…

Edits the index/value associated with the selected entry in the open table.

Delete Entry

Deletes the selected entry in the open table.

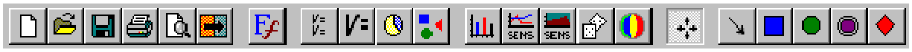
Graph Table

Displays the contents of the open table as a line graph.

Properties

Edits the properties (name, file name, lookup method) for the open table.

Tool bar



File > New

File > Open

File > Save

File > Print

File > Print Preview

File > Convert

Display > Font

Tree Window only

Values > Show Variables Window

Values > Define Values

Values > Probability Wheel

Options > Change Node Type

Analysis > Probability Distribution

Analysis > Sensitivity Analysis > One-Way

Analysis > Sensitivity Analysis > Two-Way

Analysis > Monte Carlo Simulation

Analysis > Roll Back

Navigation button (see below)

Influence Diagram only

create new arc

create new decision node

create new chance node

create new deterministic node

create new value node

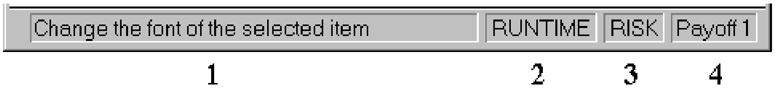
Navigation Button

The navigation button acts as a toggle between two modes of using the arrow keys in the tree window. When navigate mode is active (i.e., the button is down), the arrow keys will change which node is selected. For instance, if a single node is selected, and you press the left arrow key, DATA will select the parent of the selected node.

When navigate mode is not active (i.e., the button is up), the arrow keys operate on the text insertion cursor. It is possible to use the arrow keys to maneuver the node selection even when the navigation button is up. This requires holding down the CONTROL key when using the arrow keys.

In all windows other than the tree window, arrow keys operate on the text insertion cursor.

Status bar



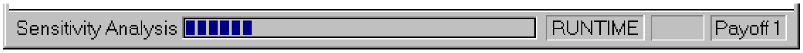
1. This area displays information about the currently-selected menu item or tool bar button. On occasion, other context-specific information is displayed in this area.

2. When the software being used is the run-time version, this area will display “RUNTIME.” The text also displays when the full version is being used with the “Mimic run-time” option, as set in the Options menu.

3. This area displays the text “RISK” when the active tree is set to calculate using a risk preference function, rather than using expected values.

4. The current calculation method for the active tree is displayed in this area. For example, if the tree is set to calculate cost-effectiveness, with payoff 2 used for cost values and payoff 1 used for effectiveness values, this area will display “C/E, 2/1.”

The status bar will also display a progress indicator when a long analysis is being run. Because this is such a useful indicator, it is desirable to leave the status bar visible at all times.



Appendix B Preferences Dialog

The Preferences dialog, which controls many settings and options in both the tree and influence diagram windows, can be found under the Edit menu (or by pressing the F11 key). It is a dynamic dialog: choose a category from the list at the left, and the appropriate options appear on the right.

In the tree window, the list of available categories is divided into three parts: Calculation Preferences, Display Preferences, and Other Preferences. In the influence diagram window, a single heading – Influence Diagram – is shown in both Macintosh and Windows; Other Preferences is available only in Windows.

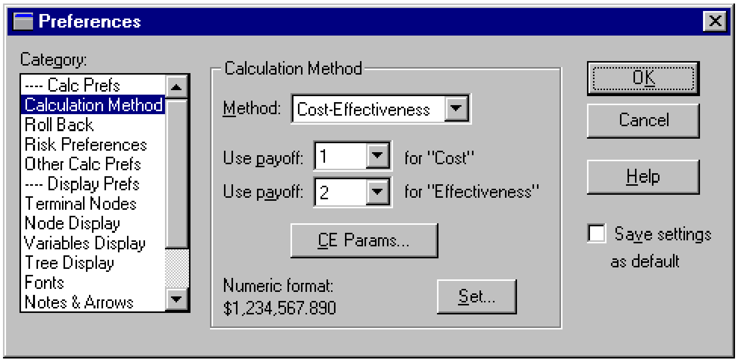
The category headings cannot themselves be selected; you must select an individual category under one of the headings, either using the mouse or typing the first letter in the category name.

Changes to the settings apply automatically to *the then active tree* (or influence diagram). You also have the option of having the current preferences, including any changes, saved as the default settings. This will automatically apply to all new tree or influence diagram documents, but *not to previously created documents*. If you want to establish the new settings as a default, click the check box at the lower right of the dialog box.

Note that the Save settings as default checkbox relates to *all* preferences, not merely those in view in the then open Preferences dialog box.

In this appendix, pages in the Preferences dialog box are presented in their order of appearance in the categories list. Following the Global Preferences page are all preferences pages associated with an influence diagram.

Calculation method preferences



Method: Cost-Effectiveness

Use payoff: 1 for "Cost"

Use payoff: 2 for "Effectiveness"

Numeric format: $1,234,567.890

□ Save settings as default

Option Description

Method

This drop-down list box specifies the calculation methods available in DATA: Simple, Cost-Effectiveness, Benefit-Cost, and Multi-Attribute.

If you select Simple, calculations performed on the active tree will be based on a single payoff. The other three calculation methods involve multi-attribute modeling, in which multiple criteria will be involved in the calculations.

See Chapters 19 through 21 for more information on multi-attribute modeling.

Use payoff

DATA allows you to assign up to four payoffs at each terminal node. This is where you specify which payoff or payoffs are to be used in calculating the then active tree. The options available will depend on the calculation method chosen above.

If the Simple calculation method is active, this option will permit you to select a single payoff. If Cost-Effectiveness is active, you must specify two payoffs to be employed in DATA's calculations, one for cost and the other for effectiveness. If Benefit-Cost is active, you must specify two active payoffs, one for benefit and the other for cost.

CE params…

If the Cost-Effectiveness calculation method is active, you should provide a willingness-to-pay value – a threshold marginal cost-effectiveness – to use when determining an optimal path at a decision node. You also have the option of setting minimum effectiveness and maximum cost thresholds. The CE params… button opens the dialog where these parameters can be entered. See Chapters 19–21.

Set weightings…

If the Multi-Attribute calculation method is active, you will have to set weightings for each of the attributes, up to four, that you want DATA to use in its calculations. As a result, the Use payoff drop-down list box will be replaced by a Set weightings… button.

Optimal path is

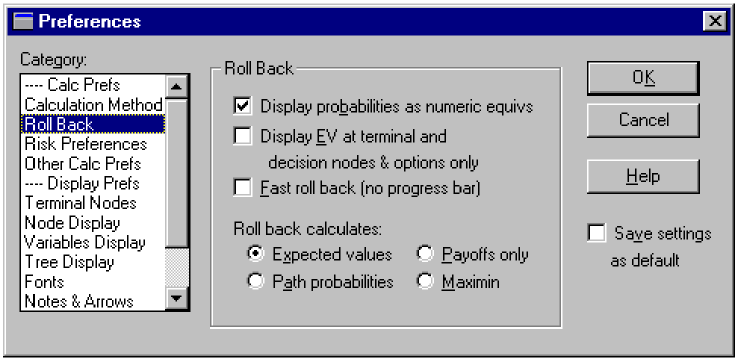
There are two option buttons: High and Low. You should select High when your tree is to be calculated on the basis of profits, income, cash flow, or other criterion that should be maximized. Select Low where less is better, such as when the payoff is based on costs, or any other quantity which should be minimized.

Keep in mind that separate settings of this flag are maintained for the multi-attribute calculation method and, for the Simple calculation method, each of four possible payoff/attribute sets and each of the three multi-attribute methods, has its own setting for this flag.

Numeric format

The Set… button is used to establish the format in which DATA displays numeric values. As with the Optimal Path setting, there are individual numeric formatting options for each Simple calculation method and each multi-attribute method. See Chapter 10 for more information on numeric formatting.

Roll back preferences



Roll Back

☑ Display probabilities as numeric equivs

□ Display EV at terminal and decision nodes & options only

□ Fast roll back (no progress bar)

Roll back calculates:

● Expected values ○ Payoffs only

○ Path probabilities ○ Maximin

□ Save settings as default

Option — Description

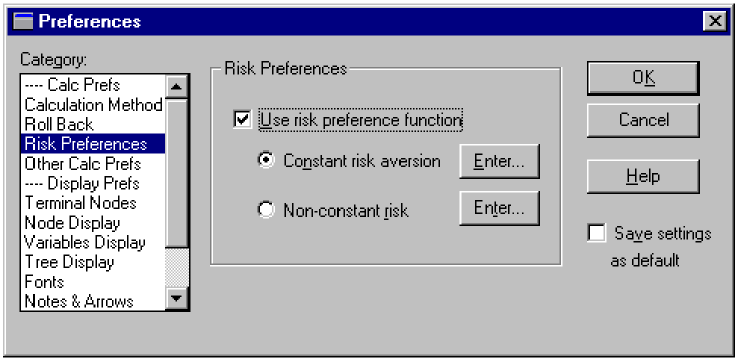
Display probabilities as numeric equivs
This option relates to probabilities which have been entered as variables. If the option is turned on, the numeric value of each of the probabilities will be displayed following roll back. Once roll back is turned off, the original expression will be restored to view.

Display EV at terminal and decision nodes & options only
When this option is on, a node will display an expected-value box during roll back only if one or more of the following conditions are met: the node is a terminal node; the node is a decision node; or the node is a decision option (i.e., its parent is a decision node).

Fast roll back
Normally, a progress indicator will display in the status bar while a tree is rolling back. Selection of the Fast roll back option will suppress this display, resulting in speed increases of up to 100%, depending on the size and complexity of the tree. With either option, you may cancel calculations by pressing ESC.

Roll back calculates
The four roll back calculation options are described in Chapter 19.

Risk preferences



☑ Use risk preference function

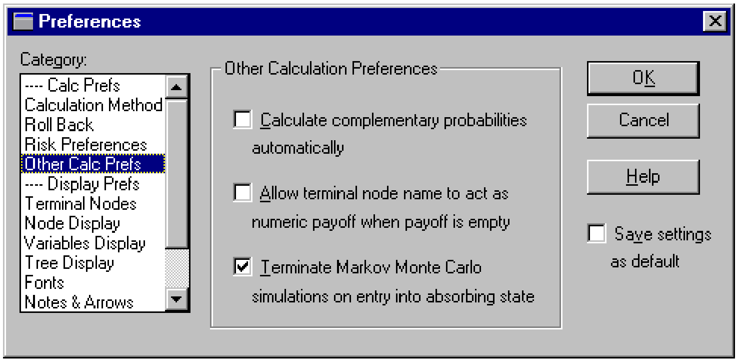
● Constant risk aversion

○ Non-constant risk

□ Save settings as default

Option	Description
Use risk preference function	When this box is checked, DATA's calculations will be based on a risk preference function rather than expected value. If this option is dimmed, a risk preference function has not previously been entered for the active tree. You may enter a risk preference function either by clicking on one of the Enter buttons in this dialog box, or by selecting Options > Enter Risk Preferences.
Constant risk aversion	When this option is selected, calculations will be based on a constant risk aversion function, rather than expected values.
Non-constant risk	When this option is selected, calculations will be based on a non-constant risk aversion function, rather than expected values.
Enter	There are two Enter buttons, one for a constant risk aversion function and the other for entering a non-constant risk aversion function. These two functions and their differences are described in Chapter 30.

Other calculation preferences



Preferences

Category:

- ---- Calc Prefs
- Calculation Method
- Roll Back
- Risk Preferences
- Other Calc Prefs
- ---- Display Prefs
- Terminal Nodes
- Node Display
- Variables Display
- Tree Display
- Fonts
- Notes & Arrows

Other Calculation Preferences

□ Calculate complementary probabilities automatically

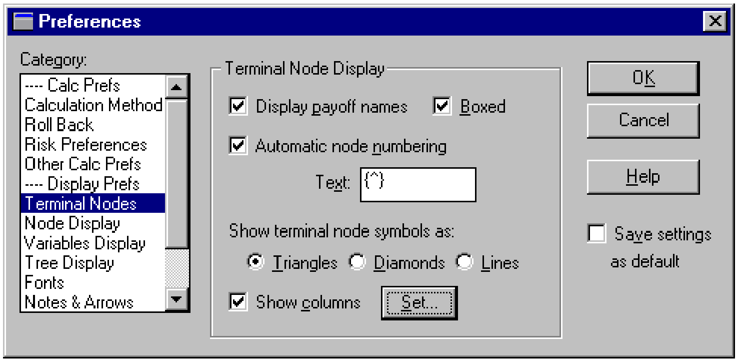
□ Allow terminal node name to act as numeric payoff when payoff is empty

☑ Terminate Markov Monte Carlo simulations on entry into absorbing state

□ Save settings as default

Option	Description
Calculate complementary probabilities automatically	When this option is checked, DATA will fill in the last probability in a set of branches emanating from a chance node, so long as all the other probabilities on branches emanating from that node are wholly numeric (i.e., no variables are used). See Chapter 14.
Allow terminal node name to act as numeric payoff	It is possible to have DATA treat the branch description at a terminal node as that node's numeric payoff value. See Chapter 14 for a list of prerequisites to using this feature.
Terminate Markov Monte Carlo simulations on entry into absorbing state	By setting this option, you indicate that the termination conditions should be ignored during Monte Carlo simulations of a Markov process. See Chapters 27 and 29 for more information.

Terminal node display preferences



Terminal Node Display

☑ Display payoff names ☑ Boxed

☑ Automatic node numbering

Text: {^}

Show terminal node symbols as:

● Triangles ○ Diamonds ○ Lines

☑ Show columns

☐ Save settings as default

Option **Description**

Always display payoff names

When this option is on, each terminal node will display the name of its then active payoff. This can be helpful for identifying terminal nodes where a payoff has not been assigned. In the case of trees having multiple payoffs, this feature makes it possible to see at a glance which payoff is active.

Boxed

If you have chosen to always display payoff names, this option lets you choose whether the payoff names should be enclosed in a box. This option relates only to tree display prior to roll back; during roll back, calculated values are always boxed.

Automatic node numbering

If this option is on, terminal nodes will display the custom text entered in the field. Use the ^ (caret) symbol in the text to represent the scenario number.

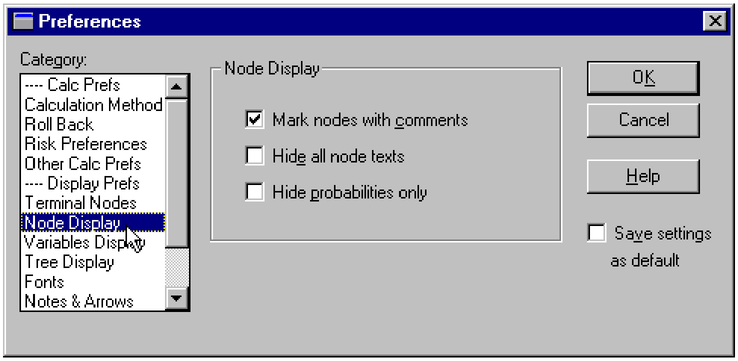
Show terminal nodes as

DATA can show terminal nodes using any of the three methods shown. Triangles are the default (and standard) method of displaying endnodes. Diamonds are used to indicate the parallelism between terminal nodes in a tree and value nodes in an influence diagram. Lines are for those applications when you do not want any symbol displayed to the right of a final outcome.

Show columns

Selecting this option will allow you to display custom columns of values to the right of the rolled-back tree. Clicking the Set… button opens the Terminal Node Columns dialog, where you can choose the calculated values (including expected and marginal values, as well as custom calculations) and formats you want. See Chapter 10.

Node display preferences



☑ Mark nodes with comments

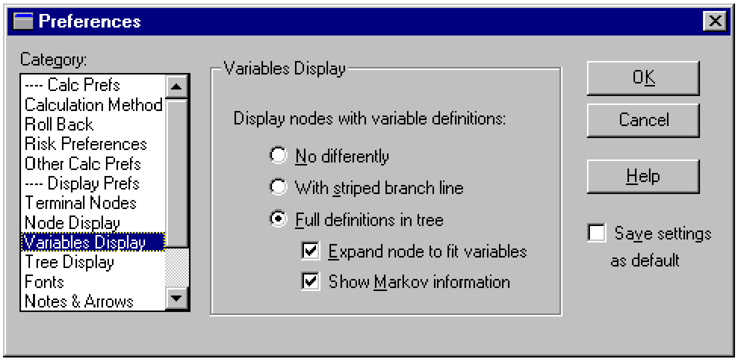
□ Hide all node texts

□ Hide probabilities only

□ Save settings as default

Option	Description
Mark nodes with comments	If this option is on, nodes at which you have entered a Node Comment (see Chapter 14) will be displayed with a small flag above the symbol. This flag does not print or export.
Hide node texts	When this option is on, the display of all textual information in the tree window is suppressed. Use this flag to get a picture of the structure of your tree.
Hide probabilities only	When the Hide node texts option is *off*, this option is available. The display of probabilities is suppressed in the tree window, while all other textual information is visible. Use this flag to temporarily simplify the display of complex trees with many uncertainties.

Variables display preferences



Preferences

Category:
---- Calc Prefs
Calculation Method
Roll Back
Risk Preferences
Other Calc Prefs
---- Display Prefs
Terminal Nodes
Node Display
Variables Display
Tree Display
Fonts
Notes & Arrows

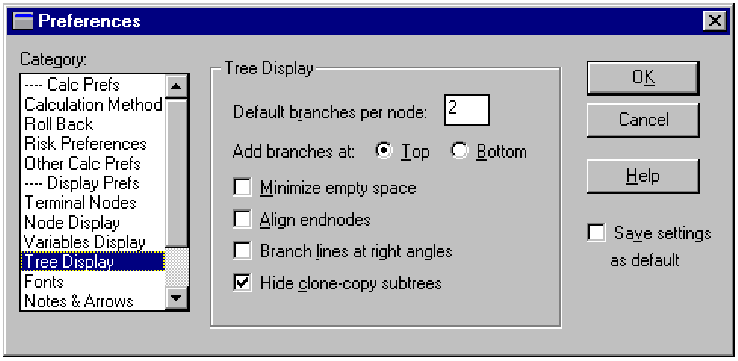
Variables Display

Display nodes with variable definitions:

○ No differently
○ With striped branch line
● Full definitions in tree
☑ Expand node to fit variables
☑ Show Markov information

Option	Description
No differently **With striped branch line** **Full definitions in tree**	These three options are described in Chapter 9.
Expand node to fit variables	When this option is not selected, long definitions will be clipped to the natural length of the node. Select this option to force node lengths to expand to fit the definitions.
Show Markov information	If you opt to show full definitions in the tree, you may also opt to show all Markov quantities — termination conditions, rewards, and Markov bindings — in the tree as well. This information is shown in the variables box with any variable definitions.

Tree display preferences



Preferences

Category:
- ---- Calc Prefs
- Calculation Method
- Roll Back
- Risk Preferences
- Other Calc Prefs
- ---- Display Prefs
- Terminal Nodes
- Node Display
- Variables Display
- Tree Display
- Fonts
- Notes & Arrows

Tree Display

Default branches per node: 2

Add branches at: ● Top ○ Bottom

□ Minimize empty space

□ Align endnodes

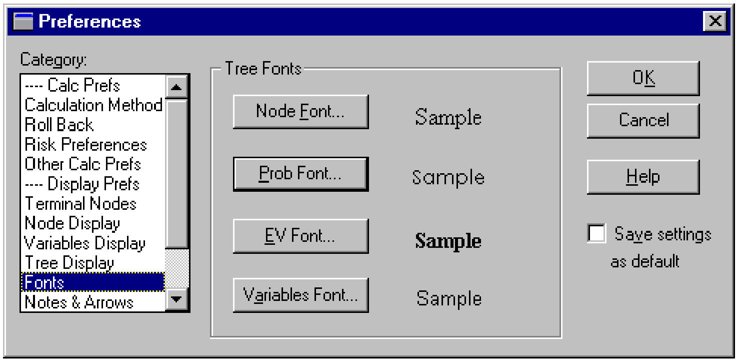
□ Branch lines at right angles

☑ Hide clone-copy subtrees

□ Save settings as default

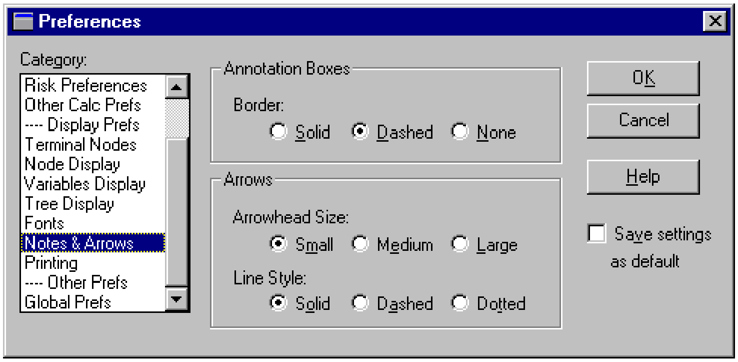
Option	Description
Default branches per node	This option sets the number of branches which are added to a node when you select Add Branches from the Options menu. The default number applies only the first time that branches are added at a given node. Once a node has branches, additional branches are added one at a time.
Add branches at…	This option enables you to control whether additional branches are to be added above or below existing branches. The Insert Branch command under the Options menu provides additional flexibility in this area. See Chapter 14 for more information.
Minimize empty space	Use of this option produces a “compressed” version of your tree. No vertical space is wasted. Because each node no longer has its own horizontal “slice” of the tree display, this option may not be used with Align endnodes.
Align endnodes	Forces all terminal nodes to line up at the rightmost edge of the tree.
Branch lines at right angles	Normally, branch lines are drawn at whatever angle is needed to provide the most direct connection from one node to the next. When this option is on, all branch lines are drawn vertically, then horizontally, rather than obliquely.
Hide clone-copy subtrees	Suppresses the display of clone-copy subtrees; display of clone masters is not affected. When this option is selected, only the name of the clone is displayed to the right of the copy node. Use of this option can dramatically reduce the physical size of your tree.

Tree font preferences



Option	Description
Node Font	This option enables you to select the default font used for naming nodes (branch descriptions). The font selected in this manner will apply to any new nodes created in the active tree and to any existing nodes, except for any nodes where the font has been set individually.
Prob Font	This option enables you to select the font used in the probability fields of the active tree, in both the rolled-back and unrolled-back state. This allows you to clearly distinguish between probability variable names and adjacent node descriptions.
EV Font	This option enables you to select the font used in the expected value boxes generated during roll back. It also applies to other information which is not user-editable and is displayed next to a node, such as clone names when clones are hidden, or payoff names when Always show payoff names is selected.
Variables Font	This option enables you to select the font used to display the definition of variables when that option is selected in the Variables Display page of the Preferences dialog.

Notes & arrows preferences



Preferences

Category:
- Risk Preferences
- Other Calc Prefs
- ---- Display Prefs
- Terminal Nodes
- Node Display
- Variables Display
- Tree Display
- Fonts
- Notes & Arrows
- Printing
- ---- Other Prefs
- Global Prefs

Annotation Boxes

Border: ○ Solid ● Dashed ○ None

Arrows

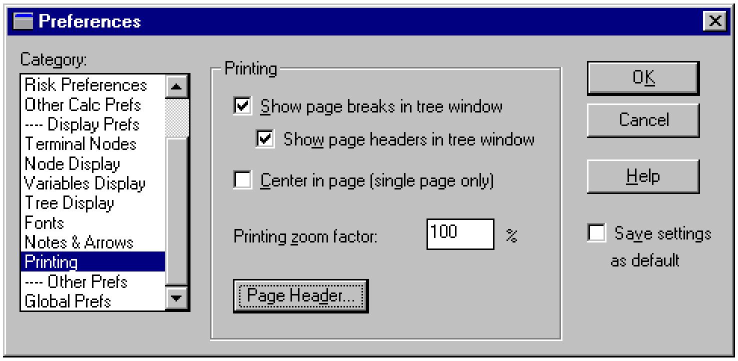
Arrowhead Size: ● Small ○ Medium ○ Large

Line Style: ● Solid ○ Dashed ○ Dotted

□ Save settings as default

Option	Description
Annotation Box Borders	This option enables you to specify the type of border which surrounds annotation boxes in the active tree. The options are a border drawn with either a solid line or a dashed line or, alternatively, no border. Your selection will apply to every annotation box in the active tree, bound or unbound.
Arrowhead Size	When your tree contains one or more arrows, this option enables you to choose whether they should be drawn with large, medium or small arrowheads. Your selection will apply to every arrowhead in the active tree.
Arrow Line Style	When your tree contains one or more arrows, this option enables you to choose whether they should be drawn with solid, dashed or dotted lines. Your selection will apply to every arrow in the active tree.

Printing preferences



☑ Show page breaks in tree window

☑ Show page headers in tree window

□ Center in page (single page only)

Printing zoom factor: 100 %

□ Save settings as default

Option	Description

Show page breaks in tree window

If this option is on, the screen display of the tree will include dotted lines to indicate where a new printed page will begin. This is likely to be more accurate than page break information shown in the print preview window.

Show page headers in tree window

If this option is on, the screen display of the tree will include any page header or footer that will be included in a printout of the tree.

Center in page

This is one of two ways in which it is possible to determine the location of documents in printouts. This option will apply only to documents which are sized to fit on a single page. When selected, the printout of the one-page document will be centered on the page. If more than one page is required for the printout, selecting this option will have no effect. The alternative method of positioning the tree or graph in a printout is described in Chapter 7.

Printing zoom factor

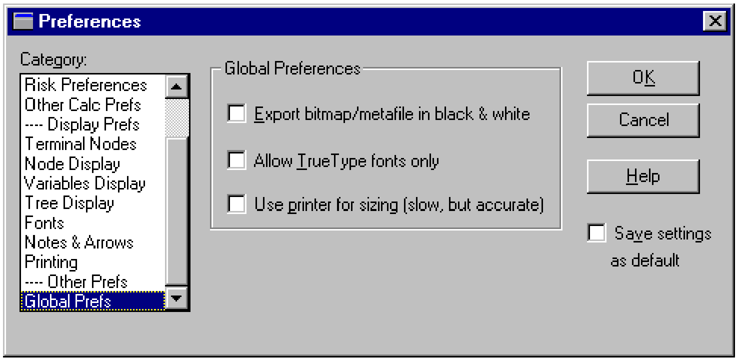
This option enables you to store a percent reduction/enlargement factor with each document. This scaling factor is document-specific. Note also that it is independent of any scaling specified under Page Setup. Thus, if your printer driver allows scaling via the Page Setup command, you run the risk of applying one percentage against another.

The printing zoom factor is also independent of the screen-display zoom factor, set in the Display > Zoom commands.

Page Header

After clicking this button, you will be able to set the page header and footer information for the active tree.

Global preferences (Windows only)



□ Export bitmap/metafile in black & white

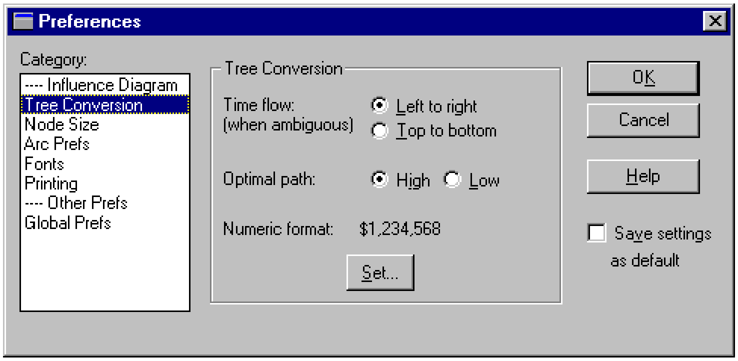
□ Allow TrueType fonts only

□ Use printer for sizing (slow, but accurate)

□ Save settings as default

Option	Description
Export bitmap/metafile in black & white	Use this option if your exported documents will eventually be printed on a black and white printer.
Allow TrueType fonts only	If this option is on, font-selection dialogs will allow you to select only TrueType fonts, which print and display identically. This selection will *not* affect fonts you have already selected.
Use printer for sizing	Whenever the tree must be resized (such as after you add nodes or change a branch description), DATA must calculate the width and height of each body of text. However, the screen and printer do not always agree on the exact amount of space needed. Selecting this option may improve the quality of the printed output, eliminating problems such as a branch description overlapping the node symbol. However, depending on the speed of your printer driver, you may find that selecting this option causes an appreciable slowdown in screen redraw. If this happens, you may want to turn on this option only when ready to print. You may find that selecting this option, while curing some problems in the printout, causes similar problems in the screen display.

Tree conversion preferences



Time flow: (when ambiguous) ● Left to right ○ Top to bottom

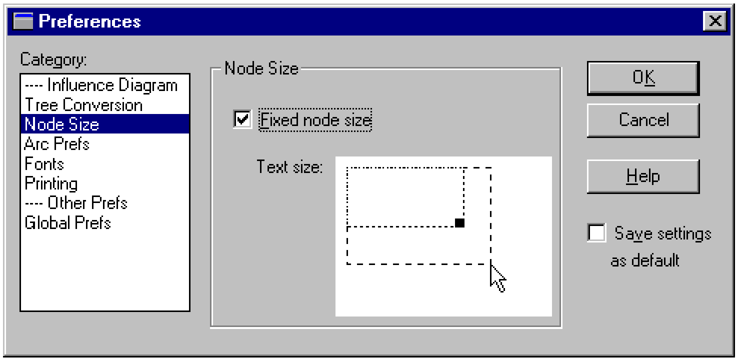
Optimal path: ● High ○ Low

Numeric format: $1,234,568

□ Save settings as default

Option	Description
Time flow	The complete algorithm used for converting influence diagrams into trees is discussed in Chapter 31. If, at any point during conversion, the ordering of two nodes is ambiguous, their locations in the window are used as a final determination. Select Left to right, if you would like nodes on the left of the influence diagram converted before nodes on the right; or select Top to bottom if you would like nodes toward the top of the influence diagram converted before nodes toward the bottom. These options apply only after all other rules have failed to determine the proper node ordering.
Optimal path	While no calculations actually occur in the influence diagram window, you may set this flag to avoid having to reset it in the tree each time you convert the influence diagram. See Calculation Method preferences, above, for details.
Numeric format	This is also carried over to the converted tree. You may enter your numeric formatting preferences to avoid reentering it each time you convert to a tree.

Node size preferences

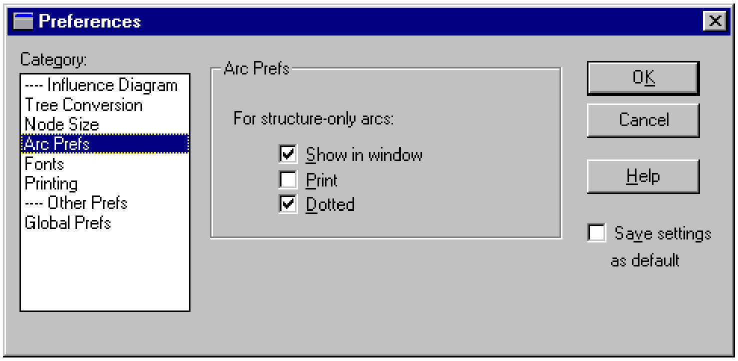


☑ Fixed node size

□ Save settings as default

Option	Description
Fixed node size	Some influence diagram users prefer to have each node sized identically, rather than having DATA determine the node's size individually. If you do not select this option, DATA will size each node based on the amount of text entered to name the node. If you opt for a fixed node size, you may indicate the size of the text area in the lower area of the dialog. Drag the corner handle to the appropriate size allocated to the text of a node; DATA will add extra space for the node's border.

Arc preferences



Arc Prefs

For structure-only arcs:

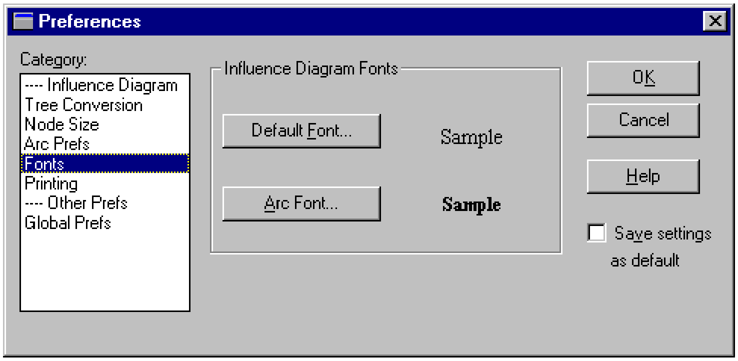
☑ Show in window
□ Print
☑ Dotted

□ Save settings as default

All three options in this dialog apply to structure-only arcs. A structure-only arc contains no probabilistic or value influence, and is used only to indicate asymmetry or timing.

Option	Description
Show in window	Deselect this option to suppress screen display of structure-only arcs altogether. You will not be able to select these arcs for editing until you reselect this option.
Print	If this option is selected, structure-only arcs will print with the rest of the influence diagram.
Dotted	Select this option to force structure-only arcs to display or print as dotted lines, rather than as straight lines.

Influence diagram font preferences



Option	Description
Default font	If you change the default font, this will affect all new nodes in the influence diagram, as well as all other nodes at which you did not set the font individually. See Chapter 10 for more information.
Arc font	This font is used for the comments of all arcs in the influence diagram. You may not set the font for a single arc comment individually.

Appendix C Functions and Operators

Variables and probabilities may be defined in terms of expressions or formulas. DATA comes with a broad selection of financial, arithmetic and other functions already defined so that you need only type the name of the function and the requisite arguments.

Mathematical expressions created in DATA necessarily contain operators (such as addition or multiplication signs) and may also contain functions (such as logarithm or net present value). Described below are the operators and functions available in creating expressions.

Operators

Arithmetic operators

These operators perform arithmetic on the values that surround them. DATA uses the traditional syntax for expressions, known as *infix* notation. For example, an expression that adds three and seven would be written 3 + 7, rather than 3, 7 +.

Symbol	Example	Explanation
+	x+y	**Addition.** Returns the sum of x and y.
-	x-y	**Subtraction.** Returns the difference between x and y. (Also used for negation, i.e., to denote negative numbers.)
*	x*y	**Multiplication.** Returns the product of x and y.
/	x/y	**Division.** Returns the quotient of x and y.
^	x^y	**Exponentiation.** Returns x to the yth power.
( )	x*(y+z)	**Grouping.** Returns the product of x and the sum of y and z. In any expression, there must be an equal number of left and right parentheses.

Relational operators

These operators return a *true* or *false* value, depending on the veracity of the expression in which they appear. A true value is represented by a numeric 1, a false value receives a numeric value of 0.

Relational operators are useful in many settings: in If() and Choose() functions, in expressions evaluated at logic nodes, and in the Markov termination condition for a Markov process.

Symbol	Example	Explanation
<	x<y	**Less than.** Returns true if x is less than y, and false if x is greater than or equal to y.
<=	x<=y	**Less than or equal to.** Returns true if x is less than or equal to y, and false if x is greater than y.
>	x>y	**Greater than.** Returns true if x is greater than y, and false if x is less than or equal to y.
>=	x>=y	**Greater than or equal to.** Returns true if x is greater than or equal to y, and false if x is less than y.
=	x=y	**Equals.** Returns true if x equals y, and false if x is not equal to y.
<>	x<>y	**Not equal to.** Returns true if x is not equal to y, and false if x equals y.

Logical operators

Three logical operators are also available: logical AND, logical (inclusive) OR, and logical NOT. AND is represented by the ampersand (&), OR by the vertical bar (|), and NOT by the exclamation mark (!).

Like the relational operators (which return 1 if a comparison is true and 0 if not), these logical operators are zero-centric. That is, any operand that is *non-zero* is treated as true, and *only* a zero operand is treated as false. The returned value is 1 if the evaluation is true, and 0 if false.

Operator precedence

In most situations, you will not need to know the details of which operators bind most tightly. However, when formulas do not appear to calculate correctly, you should check this section to see if precedence is a factor.

Operator precedence is how DATA decides where you intended to put parentheses. Consider the following example:

A + B * C + D

A quick check of the precedence list below indicates that multiplication has higher precedence (binds tighter) than addition. DATA will therefore interpret your expression as:

A + (B * C) + D

This process is continued until all uncertain bindings are resolved.

The table below lists the operators available in DATA in order of precedence. Operators with higher precedence will bind more tightly. Adjacent operators having the same precedence value will be applied from left to right.

Operator	Character	Precedence Value
Unary minus (e.g., "-30,000")	-	8
Logical "Not"	!	8
Exponentiation	^	7
Multiplication	*	6
Division	/	6
Addition	+	5
Subtraction	-	5
Comparators	<,<=,>=,>,=,<>	4
Logical "And"	&	3
Logical "Or"	|	2
Parentheses	()	1
Brackets	[]	1
Function argument separator	;	1

Notice that parentheses are at the bottom of the list. This simply means that the operators inside the parentheses will bind tightly to stay within the parentheses. They are your most useful tool for indicating your particular precedence requirements.

Built-in functions

DATA's built-in functions are case-insensitive and, in most cases, take arguments. Many functions have multiple arguments, which must be separated by semicolons (";"). Semicolons rather than commas must be used because DATA permits the use of commas as place holders in large numbers (e.g., 10,000).

Any valid expression can be used as an function argument

Functions which here indicate an argument named "LIST" actually take a flexible number of arguments. For example, "Average" returns the arithmetic mean of all of the arguments that follow it. "Average(1;4;8;13)" would return 6.5.

You may use the Function Helper, available at the bottom of the Functions pop-up menu in all formula editors, to assist you in assigning the correct parameters to each of the functions below.

Arithmetic Functions

In the first two arithmetic functions listed below, the symbol “e” represents the base of the natural logarithm, or approximately 2.718.

Function	Explanation
Exp(x)	“e” to the xth power
Ln(x)	natural (base “e”) logarithm of x
Log(x)	base 10 logarithm of x
Sqrt(x)	square root of x
Root(x;y)	yth root of x
Min(LIST)	minimum, or lowest value, of a list
Max(LIST)	maximum, or highest value, of a list
Sum(LIST)	sum of a list
Prod(LIST)	product of a list
Abs(x)	absolute value of x
Int(x)	integer component of x
Average(LIST)	arithmetic mean of a list
Round(x)	x rounded to the nearest integer
Floor(x)	greatest integer smaller than x
Ceiling(x)	smallest integer larger than x
Modulo(x;y)	remainder of x divided by y

Financial functions

For financial functions, the argument “rate” can be entered as either a percent or a decimal. For example, eight percent can be represented as either “8%” or “.08,” but not simply as “8.”

Function	Explanation
Annuity(rate;#periods)	To calculate the present value of a series of equal future payments, multiply this function times the amount of a single payment. *Formula:* $(1-(1+rate)^{-periods})/rate$
Compound(rate; #periods)	Returns the compound interest rate (effective yield) at a fixed rate over a fixed number of periods. If used as a multiplier, this function can be used to calculate future value; if used as a

divisor, it can be used to calculate discounted, present value.

Formula: $(1+\text{rate})^{\text{periods}}$

FV(pmt; pv; rate; #payments)

Returns the future value of a series of equal, periodic payments. The “pv” parameter represents an initial payment.

Formula: $[\text{pmt}*((1+\text{rate})^{\text{payments}}-1)]/\text{rate} + \text{pv}*(1+\text{rate})^{\text{payments}}$

NPV(rate; LIST of flows)

Returns the net present value of periodic cash flows, discounted.

Formula: $\sum(f_i / (1+\text{rate})^i)$

PMT(principal; rate; term)

Returns the size of equal, periodic payments required to pay off a loan, given the principal, interest rate, and term of the loan.

Formula: $\text{principal}*(\text{rate}/(1-(1+\text{rate})^{-\text{term}}))$

PV(pmt; fv; rate; #payments)

Returns the present value of equal, periodic payments at a fixed interest rate. The “fv” parameter represents a final payment.

Formula: $[\text{pmt}*(1-(1+\text{rate})^{-\text{payments}})]/\text{rate} + \text{fv}/((1+\text{rate})^{\text{payments}})$

Miscellaneous functions

Function **Explanation**

If(condition; trueval; falseval)

Evaluates a condition, using relational operators, and then returns “trueval” if the condition is true or “falseval” if the condition is false. For example, “IF(x<0;50;75)” would return 50 if x were negative and 75 if x were non-negative.

Choose(index;LIST of values)

Returns a value based on its location in the LIST, as specified by the index. For example, “Choose(2;100;200;300)” returns 200, because the index is 2, and 200 is the second value in the list. The index must be a positive integer; an error is reported for fractional or out-of-range values. See below for comments.

Dist(LIST)

Obsolete. See DistSamp(), below.

DistSamp(index)

Returns the value associated with an analytic distribution in your tree. During expected value calculations, this function returns the distribution’s mean or median. During Monte Carlo simulations, this function returns a randomly sampled value taken from the distribution. See Chapter 24.

Sub(index)	Returns the value associated with a bi-directional or DDE link. See Chapter 12.
Stdev(LIST)	Returns the standard deviation of a list of numbers.
Pi()	Returns π, or approximately 3.1416. Although no arguments are accepted, you must type the left and right parentheses, as this is a function, not a variable.
Inf()	Returns infinity (∞). Although no arguments are accepted, you must type the left and right parentheses.

The Choose() function can be used to good effect in a number of situations:

- If you have a model which must serve several different populations, you should consider using the Choose() function. For instance, in a model which uses different values for people of age 20, 30, and 40, you might assign a variable Age a value of 1, 2, or 3 for each age bracket. Then a Choose() function could determine which values to use; for example, "Choose(Age; prob20; prob30; prob40)." Without the Choose() function, you might need three independent trees.

- You can use the Choose() function for a form of "nonlinear" sensitivity analysis. If you perform the sensitivity analysis on the index parameter, you can see the impact of using the different values specified in the list.

- The Choose() function can be used to include more than four attributes in your model. You might assign Attr1 through Attr3 as normal; then, for the fourth payoff formula, enter "Choose(finalAttr;Attr4;Attr5;Attr6)." The value of the variable finalAttr could be set to indicate which of the last three utility scales should be used as the fourth attribute.

Medical Functions

The following four functions employ the arguments *rate* (or prob) and *time*. In each case, it is essential that the values for these parameters be based on the same scale. Thus, if rate or probability is in terms of years (such as deaths per year), time must also be in years.

Function	**Explanation**
UtilDiscount(util; rate; time)	Discounts a specified utility at the specified discount rate over the specified period. *Formula:* util / ((1 + rate)^{time})

RateToProb(rate; time)

Converts a rate, such as deaths per year, into a probability, such as the probability of any single patient dying in any given year.

Formula: $1 - e^{-rate*time}$

ProbToRate(prob; time)

Converts a probability, such as the probability of any single patient dying in any given year, into a rate, such as deaths per year.

Formula: ln(1 - prob) / time

DEALE(rate; time)

DEALE is an acronym for "declining exponential approximation of life expectancy."

Formula: $(1 - e^{-rate*time}) / rate$

To use UtilDiscount(), you will likely use the keyword _stage in the time parameter to indicate that discounting is cumulative. For instance, consider a Markov process with year-long stages. You have estimated that the discount rate for quality-adjusted life expectancy should 5% per year. You might use *UtilDiscount(1; 0.05; _stage)* as the incremental reward in the Well state, 1, representing one quality-adjusted year of life. If the cycle length was changed to one month, you might use the expression *UtilDiscount(1/12; 0.05; _stage/12)*. If you were calculating months of life, the first parameter would simply be changed to 1.

RateToProb() is also suited to Markov processes. Consider the same Markov process, using a one-year cycle length. In a given population, the disease under observation has a mortality rate of 5 deaths per 100 per year. As a transition probability from the Well state to the Dead state, you might use *RateToProb(5/100; 1)*. Note that the keyword _stage is conspicuously absent. This is because the time parameter here is *not* cumulative; in this case, one is interested in the probability of dying during a *single* interval – not the probability of dying during this, or any previous, interval. Using a *cumulative* time parameter, in a different situation, might allow you to simulate memory during a Markov cohort simulation – for example, calculating the probability of an event having occurred at all during any cycle, based on a single-cycle rate.

ProbToRate() provides the reverse function to RateToProb(), ultimately enabling a probability to be converted to a rate, proportionally increased or decreased using the additive property of rates, and then converted back to a probability.

In both RateToProb() and ProbToRate(), the second, time parameter will allow you to convert between a rate for an interval of one length and a probability for a interval of a different length.

The DEALE() function is cumulative, so the time parameter means "over the course of this amount of time."

These functions are particularly sensitive to user errors. You are urged to exercise great care when using them in your models. It is recommended that you use the Evaluator feature, located under DATA's values menu, to test expressions using these functions; see Chapter 9.

Here are three functions related to calculating probabilities:

OddsToProb(odds)	Converts an odds ratio into a probability. *Formula:* odds / (1 + odds)
ProbToOdds(prob)	Converts a probability into a odds ratio. *Formula:* prob/ (1 - prob)
ProbFactor(prob; fac)	First converts the probability to an odds ratio, then multiplies it by the given factor, then converts it back to a probability. See the two functions, above.

Appendix D Distributions

Derivations of and from these formulae may be found in most texts on probability theory. See, for example, Christensen, Ronald; *Data Distributions: A Statistical Handbook* (2nd Ed.); Lincoln, Massachusetts: Entropy Limited, 1989.

See Chapter 28 for the usage of distributions in DATA.

Normal distribution

Formula: $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

Domain: $-\infty < x < \infty$

Parameters: Mean $-\infty < \mu < \infty$, standard deviation $\sigma > 0$

Uniform distribution

Formula:

$$f(x) = \begin{cases} 0 & x < LowX \text{ or } x > HighX \\ \frac{1}{HighX - LowX} & LowX \le x \le HighX \end{cases}$$

Domain: LowX < x < HighX

Parameters: Low X, High X

Triangular distribution

Formula:

$$f(x) = \begin{cases} 0 & x < LowX \text{ or } x > HighX \\ 2 * (HighX - LowX) * \dfrac{x - LowX}{Mode - LowX} & LowX \le x \le Mode \\ 2 * (HighX - LowX) * \left(1 - \dfrac{x - Mode}{HighX - Mode}\right) & Mode < x \le HighX \end{cases}$$

Domain: LowX ≤ x ≤ HighX

Parameters: Low X, High X, Mode

Beta distribution

Formula: $$f(x) = \frac{(n-1)!}{(r-1)!(n-r-1)!} x^{r-1}(1-x)^{n-r-1}$$

Domain: $0 < x < 1$

Parameters: $r > 0$, $n > r$

Chi distribution

Formula: $$f(x) = \frac{2\left(\frac{n}{2}\right)^{\frac{n}{2}}}{\theta\, \Gamma\left(\frac{n}{2}\right)} x^{n-1} e^{\left(\frac{-nx^2}{2\theta}\right)}$$

Domain: $x > 0$

Parameters: $n > 0$, $\theta > 0$

Chi-Squared distribution

Formula: f (x) = $\dfrac{x^{\left(\frac{n}{2}-1\right)} e^{-\frac{x}{2\theta}}}{(2\theta)^{\frac{n}{2}}\Gamma\left(\frac{n}{2}\right)}$

Domain: x > 0

Parameters: $n > 0$, $\theta > 0$

Erlang distribution

Formula: f (x) = $\dfrac{(k\lambda)^k x^{k-1}}{(k-1)!} e^{-k\lambda x}$

Domain: x > 0

Parameters: $k > 0$, $\lambda > 0$

Exponential distribution

Formula: f (x) = $\lambda e^{-\lambda x}$

Domain: x > 0

Parameter: $\lambda > 0$

Gamma distribution

Formula: f (x) = $\dfrac{\beta^{\alpha} x^{(\alpha-1)}}{\Gamma(\alpha)} e^{-\beta x}$

Domain: x > 0

Parameters α > 0, β > 0

Hyper-exponential distribution

Formula: f (x) = $2\,\lambda p^{2} e^{-2\lambda p x} + 2\,\lambda\left(1-p\right)^{2} e^{-2\,\lambda(1-p)x}$

Domain: x > 0

Parameters: λ > 0; 0 < p < 1

Laplace distribution

Formula: f (x) = $\dfrac{1}{2b} e^{-\frac{|x-a|}{b}}$

Domain: - ∞ < x < ∞

Parameters: *b* > 0, - ∞ < a < ∞

Logistic distribution

Formula: f (x) = $\dfrac{ae^{-(ax+b)}}{\left[1+e^{-(ax+b)}\right]^2}$

Domain: - ∞ < x < ∞

Parameters: $a > 0$, - ∞ < b < ∞

Lognormal distribution

Formula: f (x) = $\dfrac{1}{x\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{\ln(x)-\mu}{\sigma}\right)^2}$

Domain: x > 0

Parameters: σ > 0, - ∞ < μ < ∞

Maxwell distribution

Formula: f (x) = $\dfrac{4x^2}{\alpha^3\sqrt{\pi}} e^{-\frac{x^2}{\alpha^2}}$

Domain: - ∞ < x < ∞

Parameters: α > 0

Rayleigh distribution

Formula: f (x) = $\frac{x}{\alpha^2} e^{\left(\frac{-x^2}{2\alpha^2}\right)}$

Domain: x > 0

Parameters: $\alpha > 0$

Weibull distribution

Formula: f (x) = $\lambda k x^{k-1} e^{-\lambda x^k}$

Domain: x > 0

Parameters: $\lambda > 0$, k > 0

Poisson distribution

Formula: P (x=i) = $e^{-\lambda} \frac{\lambda^i}{i!}$

Domain: i = 0, 1 , 2, …

Parameters: $\lambda > 0$

Binomial distribution

Formula: P (x) = $\binom{n}{x} p^x (1-p)^{n-x}$

Domain: x = 0, 1, 2, …, n

Parameters: 0 < p < 1, n = 1, 2, 3, ...

Appendix E Technical Notes

Avoiding sensitivity analysis errors

Sensitivity analysis is one of DATA's most powerful features. Because the results of a sensitivity analysis can have a major impact on decision making, it is vital that the analysis be carried out correctly and accurately. This makes it is necessary for you to understand how the linking of variables and the location of definitions of variables can affect the accuracy of DATA's output.

The following examples assume you understand how variables are used in DATA, including where they are defined and how the location of the definition can affect calculated results. Before proceeding to the examples, you may want to reread the section in Chapter 8 which explains how and where to define variables. A clear understanding of this methodology is an absolute prerequisite to obtaining accurate analyses.

Terminology: A *variable* refers to any name assigned to a quantity in the tree, such as pSave_Foot. A *variable definition* is any single value (or expression) assigned to that variable, such as (in the Rock Climber problem) the definition of pSave_Foot=0.6 at the root node.

How sensitivity analysis works

Before addressing the particular problems that can arise, first consider the mechanism which DATA uses to implement sensitivity analysis.

After you select a variable and a numeric range, DATA scans the relevant portion of your tree for definitions of the chosen variable. (Nodes are scanned both in the subtree of the selected node, and in the path history of the selected node.) If no definitions are found, an error is reported; if exactly one is found, the range is applied and the analysis continues. If more than one definition of the selected variable is found, DATA will ask you to specify which definitions should be varied, and which should retain their original expressions. This process is described in Chapter 22.

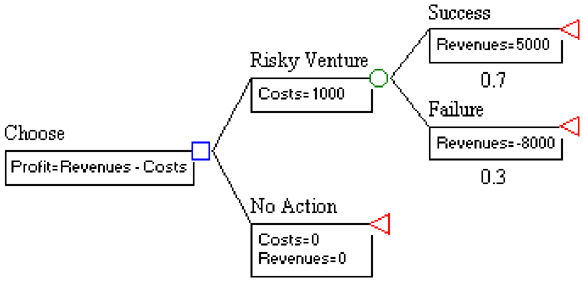
During the course of the analysis, a variable definition which is to be varied will be replaced by a different numeric value at each interval. The original contents of the variable definition are *overwritten* with the numeric value. When the analysis is complete, the original expression is replaced.

If you opt to retain the original expression for any definitions of the selected variable(s), those definitions will not be affected by the sensitivity analysis process, and will retain their original expression throughout the analysis.

There are two potential problems which may arise from an incomplete understanding of this process.

Problem one: analyzing the correct variables

Consider this tree:



How should you do a sensitivity analysis on Profit? This seems like a perfectly reasonable thing to do.

There is only one definition of Profit in the tree. However, it is used at every terminal node. If you analyze Profit over a range from -$10,000 to +$10,000, a numeric value from this range will be placed in the definition of Profit at each interval. The original definition, “Profit = Revenues - Costs,” will be replaced by a numeric definition, “Profit = -10,000,” for instance. The link between the variables Revenues and Costs is lost! Every terminal node which uses Profit as its payoff will then have a value of -10,000, and the tree will not calculate correctly.

One possible solution is to include additional definitions of “Profit = Revenues - Costs.” You could choose to retain the original expressions for these definitions, and so retain the link to the values of the intermediate variables. This solution, however, is suboptimal. It adds unneeded redundancy, which can quickly become a nightmare to maintain.

The best solution is to avoid performing sensitivity analysis on Profit. The assigned formula indicates that you are modeling monetary values at a finer grain: you are modeling Revenues and Costs independently. Hence you should analyze these variables, Revenues and Costs, because they are of the finest grain used in your model.

The general rule can be stated as follows:

> *Sensitivity analysis should only be performed on variables all of whose definitions are exclusively numeric.*

Any variable which is defined as a formula should not be considered a candidate for sensitivity analysis. In fact, any variable which is defined as a formula should have exactly one definition; any other definitions (be they another formula or a numeric value) probably represent a separate quantity, and should use a different variable name.

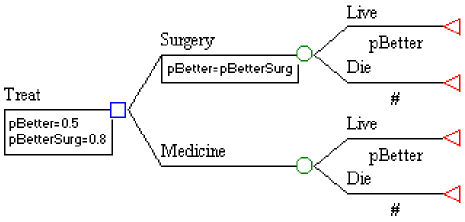
An even better rule combines the above generality with the topics discussed in Chapter 22:

> *Sensitivity analysis should be performed only on variables which have exactly one definition, which is numeric.*

Problem two: broken bindings

Consider the tree fragment below, from a medical decision model.

If you understood the material discussed in Problem one, you will recognize that the variable definitions used in this tree are improper. But for purposes of explanation, and because users often create trees which structure variables in this manner, let's go on to consider the potential problems.



In this example, the definition of pBetter is linked to the value of pBetterSurg at the node Surgery. If you perform a sensitivity analysis on pBetter, you will be asked for two ranges: at the node Treat, and at the node Surgery. If, instead, you were to do a sensitivity analysis on pBetterSurg, you would be asked for one range, at the node Treat.

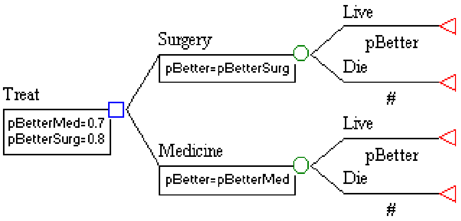
So far, so good. Now, assume you would like to do a two-way sensitivity analysis on pBetter and pBetterSurg. This certainly sounds innocent and useful enough, but it raises a potential problem.

As explained above, the formula pBetter=pBetterSurg will be replaced during analysis with the numeric values given by the user for that variable definition. That is, the linking of pBetter to pBetterSurg will be lost for the duration of the analysis, in which case any range given for pBetterSurg at the root node will be disregarded.

This consequence will occur only when a range of values is specified for the variable definition in question. In other words, if you accept the specified range of values for the definition of pBetter at the Surgery node, DATA will break the linkage between pBetter and pBetterSurg. If you, instead, choose to retain the original definition at the Surgery node, DATA will leave the linkage intact. This will mean, in the latter case, that the range of values specified for pBetterSurg will be correctly reflected in the variable definition of pBetter at node Surgery.

Below is shown a version of the tree which is one step closer to being a properly defined model.

You can see that the variables which may be properly analyzed are pBetterSurg and pBetterMed. The formulaic variable pBetter should *never* be analyzed in a sensitivity analysis.



The best tree would eliminate pBetter altogether, and use pBetterMed and pBetterSurg directly under the branch lines of the appropriate nodes. This solution is not always possible, especially when you are using cloned subtrees, so the above tree should be considered a good example of how to structure your variable definitions.

Changing the storage location for tables

It is possible to change the directory (Windows) or folder (Macintosh) in which tables are stored. This is particularly useful for network installations where the software is stored on a network drive on which users do not have write-permission. To change the directory (Windows) or folder (Macintosh), you must create a "pointer" file which specifies the name of the directory or folder to be used for table storage.

In DATA for Windows, create a text file called "TABLE.DIR" in the root application directory (e.g., "C:\Program Files\DATA"). Inside the file, type the name of the directory which will be used for table storage. For example, your TABLE.DIR file might contain the text "C:\Med\Info\Tables" to cause DATA to look in that directory for tables. This would give each computer on the network a private tables directory.

In DATA for Macintosh, create a text file called "Table Directory" in the same folder as the application. Inside the file, type the name of the folder which will be used for table storage. For example, your Table Directory file might contain the text "Hard Disk:Users:Tables:" to indicate the folder to be used for tables. Note that the entry in the Table Directory file must end with a colon.

DATA will not create the custom directory or folder for you; you must ensure that it exists for each computer on the network before creating the pointer file. Be sure to copy whatever tables you are currently using into that directory/folder.

It is important to keep in mind that DATA reads the contents of the table directory or folder only when you initially start the program. Any changes you make manually either to its contents or to the optional pointer file will not be recognized until the next time you start the program.

Details of Markov process calculations

Below are listed the steps which DATA takes, in the order they occur, during calculations. You may find this section useful when looking for errors in your model.

For expected value calculations (cohort simulation)

Initialization phase

- The values of _stage, _stage_reward, and _total_reward are set to zero.
- The cohort is distributed among the states according to the initial probabilities, and initial rewards are accumulated.

- Initial rewards are tallied and added to _stage_reward and _total_reward.
- The cohort is redistributed according to the transition probabilities.
- During the redistribution, transition rewards are accumulated.
- Transition rewards are tallied and added to _stage_reward and _total_reward.

__Processing phase__

- The value of _stage is incremented.
- The termination condition is checked. If it is true, the Processing phase is exited.
- Incremental rewards are accumulated.
- The cohort is redistributed according to the transition probabilities, and transition rewards are accumulated.
- The value of _stage_reward is reset to zero
- Tallied incremental and transition rewards are added to _stage_reward and _total_reward.

__Final phase__

- The value of _stage_reward is reset to zero.
- Final rewards are accumulated, tallied and placed in _stage_reward and added to _total_reward.
- The value of _total_reward becomes the value of the Markov process.

For Monte Carlo simulations

__Monte Carlo initialization__

- The values of _stage, _stage_reward, and _total_reward are set to zero.
- The trial chooses a state in which to begin the process according to the distribution of initial probabilities, and receives any initial reward.
- Any initial reward is placed in _stage_reward and _total_reward.
- The trial chooses a new state according to the distribution of transition probabilities; transition rewards may be received depending on the transition path followed.

- Transition rewards are added to _stage_reward and _total_reward.

Monte Carlo processing phase

- The value of _stage is incremented and distributions are resampled, if appropriate.
- The termination condition is checked. If the option *Terminate Markov Monte Carlo simulations on entry into absorbing state* is turned on (in the Other Calc Prefs page of the Preferences dialog), that particular condition is checked first. If either is true, the Processing phase is exited.
- An incremental reward is received.
- The trial chooses a new state according to the distribution of transition probabilities, and transition rewards may be received, depending on the transition path followed.
- The value of _stage_reward is reset to zero
- Any incremental and transition rewards are added to _stage_reward and _total_reward.

Monte Carlo final phase

- The value of _stage_reward is reset to zero
- Any final reward is received and added to _stage_reward and _total_reward.
- The value of _total_reward becomes the value of the Markov process for this trial.

Using DATA's command line

A number of parameters can be used on the command line used to start up DATA.

Using custom start-up flags

You can employ one or more start-up flags to alter certain functionality:

-n Starts DATA in “novice” mode. When DATA is in novice mode, its menus are slightly shorter.

-i Tells DATA that you usually prefer to input models as influence diagrams rather than trees. At start-up you will see a new influence diagram rather than a new tree; the default listing under the File > Open dialog will be influence diagrams rather than trees; and the File > New dialog will assume you want a new influence diagram.

-t Suppresses the display of a new, empty tree.

-u Overrides international (non-U.S.) setting in your operating system's control panel. If your country setting utilizes a comma between decimals and integer and a period to separate thousands (e.g. “1.000,00”), this command will force DATA to use the U.S. setting, which reverses these usages, so that you can conveniently share models with others who customarily use the U.S. setting.

In DATA for Windows, these flags are invoked on the command line of the executable. To set the command line in the Explorer (Windows 95/98), create a shortcut to DATA – on the Windows Desktop, for example. Then, open the shortcut’s Properties dialog, edit the Target field (this shortcut can be renamed to reflect a tree being opened). In the Program Manager (Windows 3.1), create a new icon for DATA, and press ALT + ENTER when it is selected.

In DATA for Macintosh, these flags are placed in a text file called “Startup Flags” which is stored in the same folder as the application.

In DATA for Windows, it is possible to designate the names of files to be opened at startup. The names of these files are specified on the command line, in which case the “-t” flag is automatically applied. If the file name, or any directory in its path, contains spaces or punctuation, you must enclose the file name in quotation marks. This is a good idea for any file you open with the 32-bit version of DATA.

You may automatically define variables in the specified tree upon startup. For example, the target field of a shortcut icon in Windows might read:

```
data32 c:\MyTree.tre?v1=3?v2=v1^2;v3=x
```

DATA will open MyTree and define three variables at the root node:

```
v1 = 3
v2 = v1 ^ 2
v3 = x
```

There must not be any whitespace in the formulas. The question mark is the delimiter.

DATA for Macintosh does not accept file names in its Startup Flags file. To open a particular file at startup, use standard Macintosh techniques, such as dropping it onto the application's icon.

If you are invoking DATA for Windows from another program, such as Visual Basic, you may pass the SW_MINIMIZE flag to the WinExec function. DATA will operate as a minimized icon, and will not display its splash screen.

Here is an example of a command line in DATA for Windows with flags set to open DATA in novice mode, with startup files My Tree.tre and MyInf.ind:

```
DATA32.EXE -n “c:\MyTree.tre” c:\files\MyInf.ind
```

APPENDIX F FOR SMLTREE AND DECISION MAKER USERS

This appendix is designed to help users who are familiar with SMLTREE or Decision Maker make the transition to DATA. In light of the strong similarity between SMLTREE and Decision Maker, any reference in this Appendix to either applies equally to both.

Reading SMLTREE and Decision Maker files (Windows)

DATA for Windows can read most SMLTREE and Decision Maker files directly. Simply use the File > Open command to open a SMLTREE file as though it were a DATA file. DATA will create a new, untitled decision tree containing the imported file; you should save the imported tree with a name or location different than the original file.

Due to conceptual differences in tree format, the import facility should be regarded as an initial step in a multi-step process. It is possible that some manual modifications to the tree will be required.

File import options

When you open a tree created by SMLTREE or Decision Maker, you will see the File Import Options dialog. In it, you can specify three preferences:

- **Use clones at duplicated subtrees** – Check this box duplicate subtrees using the clone feature. If you leave it unchecked, separate, editable subtrees will be used. See Chapter 12.
- **Use hash mark explicitly in last branch of each chance node** – Decision Maker effectively ignores the probability on the lowest (lower) branch, so it is possible that your model contains sets of probabilities that do not sum to one. Because DATA requires probabilities to sum to one, you may find that some trees will not calculate immediately. If you select this check box, the bottom branch of each chance node will be changed to use # for the probability. (In DATA, the pound sign,

#, can be used as the probability of any single chance node branch to automatically calculate the remainder probability.)

- **Truncate identifiers to 8 characters** – SMLTREE and Decision Maker require identifiers to be 8 characters or less. However, in a formula, you are allowed to use longer names; they will be internally truncated, though. If, for example, you were to enter a formula *total = testcostSurg + testcostMed*, SMLTREE or Decision Maker would internally truncate the formula to *total = testcost + testcost*. If you select this check box, DATA will convert the former formula to the latter, thereby preventing calculation errors.

If there are multiple unattached nodes in the file, you will be asked to specify which should be used as the root node of the new DATA tree. This will enable to you repeat the import process a number of times, in order to convert separate trees with each unique root node.

Diagnostic messages

The following diagnostic messages may be displayed after the import is complete:

- **Invalid bindings** – SMLTREE and Decision Maker allow bindings that have no meaning in DATA. For example, you may change individual values in a table by binding *pLive[0] := 0.4* at a node. You may also set more than one variable in a single binding, as in *pDie := pDieMed := 0.5*. Bindings such as these are ignored when the file is read.
- **Existing tables** – If an imported tree file contains a table with the same name as a table already existing on your system, the new table is ignored.
- **Recursive clones** – Recursive subtrees are not allowed in DATA.
- **Variable name changes** – In SMLTREE and Decision Maker, you may use the period in a variable name, such as *u.Med*. DATA will convert the period to an underscore, to create the new variable *u_Med*.

Conversion limitations

DATA is very sensitive to the format of SMLTREE and Decision Maker files. It may not be able to import files which you have edited manually in a text editor.

In addition, some concepts are not implemented in the file import. Trees that use these features may not calculate correctly (or at all) in DATA, without significant changes.

- **Recursive trees** – Recursive tree structures are not allowed in DATA.
- **Markov access functions** – Decision Maker includes functions such as InitProb() and CyclProb() to give you direct access to Markov calculations as they occur. These are not implemented in DATA, although it may be possible to achieve some of the same results by reworking a model to use Markov keywords. See Chapters 25 and 27 and Appendix D for information on how DATA calculates Markov models.
- **Multiple trees** – DATA can only import a single tree at a time. If your file has more than one node which may be a root node, DATA will ask you to select a single root node and discard the nodes not attached to its subtree. You may reimport the file multiple times to recreate any other subtrees.
- **Distributions** – Trees which contain distributions may not import or calculate correctly, in which case you will have to recreate distributions in the new tree.

Common import problems

A number of key differences between the implementation of decision trees and Markov processes in DATA and that in Decision Maker and SMLTREE.

Bindings: See below for a discussion of the differences between variables in DATA and in SMLTREE.

Functions: Functions fall into three categories: (1) functions that DATA can use directly, (2) functions that the importer converts into a form usable by DATA, and (3) functions that have no counterpart in DATA, and cannot be used.

1. Most imported functions can be used directly by DATA: LOG, LN, EXP, ABS, SQRT, INT, MAX, and MIN.
2. The DEALE2 function is called DEALE in DATA. The MULTIWAY function is identical to DATA's Choose function. These simple changes are performed automatically for you when you import a tree.

The DUAL function represents multiple payoffs; the importer will use the two arguments as payoffs one and two at the terminal node. See Chapters 20 and 21 for more information on multi-attribute analysis.

Also in this category is the PASSBIND function. As mentioned above, all variables in DATA are PASSBIND variables, so the function has no effect on its formula argument. The importer will remove the text "PASSBIND" from your formulas.

3. The functions NODE, RND, TRUE and FALSE are not available in DATA. These functions will remain intact through the import process, but will prevent your tree from calculating.

Tables: In SMLTREE, tables can store non-numeric values for evaluation during tree calculations. DATA tables only accept numeric values for entries. DATA tables also cannot be modified during a model analysis.

Variable definitions

With the exception of recursive variable definitions (e.g., x=x+1), all variables in DATA are *passbind* definitions. That is, the variable binding statement a := Passbind(b) in SMLTREE or Decision Maker is equivalent to a variable definition a = b in DATA. The only way to simulate non-passbound variable definitions in DATA is to use recursive definitions (e.g., a=a+b). This difference will not necessarily cause any changes in calculations in your converted models, but its implications must be considered.

Please read Chapters 8 and 9 for a complete description of how DATA uses variables.

Utilities

Decision Maker and SMLTREE use the Dual and Multiway functions to allow multiple utility scales. In DATA, each terminal node may have up to four different payoffs. (*Payoffs* in DATA are akin to *utilities* in Decision Maker and SMLTREE.) You may choose which payoff is active for calculations in the Preferences dialog.

See Chapter 19 for more information on changing what the tree calculates, and Chapters 20 and 21 for information on multi-attribute analysis, including cost-effectiveness.

Markov processes

You will find that while DATA's implementation of Markov processes is somewhat similar to that in Decision Maker and SMLTREE, there are some important differences, including differences in terminology. For instance, a *cycle* in SMLTREE is called a *stage* in DATA. *Utilities* (initial, incremental, and final, for each Markov state) are referred to as *rewards*. *Tolls* are called *transition rewards*.

In DATA, Markov states (i.e., the direct descendants of a Markov node) may have any name; they are not restricted to variable-type names. To indicate a transition from one state to another in SMLTREE, you used the name of the state being entered as the utility value of the jump node. In DATA, you will assign the name of the state being entered to the jump node using a special dialog box.

The names of states (and transitions) are not restricted as they are in SMLTREE. You may assign a name of arbitrary length, and include punctuation and spaces.

Each of SMLTREE's reserved variables used in Markov processes has an analogue in DATA:

In SMLTREE	In DATA
m.CYCLE	_stage
m.CYCSUM	_stage_reward
m.CUMSUM	_total_reward
m.TUNNEL	_tunnel (*see below*)

Decision Maker and SMLTREE use m.TUNNEL for two purposes: first, the user binds a value of m.TUNNEL to the Markov state to indicate the number of substates to be created; second, the program reassigns it to each internal substate to indicate which substate is being evaluated, so you can use different transition probabilities (and rewards) for each substate.

In DATA, the _tunnel keyword is used only for the latter functionality. You may *not* assign its value using a binding; to indicate the number of tunnel substates to be created, use the Markov State Information dialog, as described in Chapter 27.

To assign utilities (called "rewards" in DATA), you do not bind variables. Instead, you assign rewards in the Markov State Information dialog at each state. There are no variables similar to m.uINIT, m.uINCR, or m.uFINAL, although the same functionality is provided.

Similarly, the m.TOLL and m.TOLL2 reserved variables have no parallel in DATA. Instead, transition rewards are assigned at non-state nodes in the Markov subtree. To enter these values, use the Markov Transition Rewards command in the Values menu. Note that transition rewards in DATA are *added* to the net reward, rather than *subtracted* from it, as in Decision Maker and SMLTREE.

Please read Chapters 25 and 27 thoroughly to understand DATA's implementation of Markov processes. You might also read the relevant technical note included in Appendix E.

Subtrees

Each node in Decision Maker and SMLTREE is the root of a subtree; the same is true in DATA.

To attach the same subtree to two different nodes in Decision Maker or SMLTREE, you need only create the subtree once, then attach it as a branch wherever you wish. DATA offers comparable functionality and lets you choose between two methods for replicating subtrees: (1) use the tree clipboard to copy and paste a subtree, or (2) use clones to attach uneditable duplicates of a single subtree. Clones may seem more familiar to you, but both ways will achieve the same end result for calculation purposes.

- To copy a subtree to another location, select the root of the subtree, then choose Options > Select Subtree. Then choose Edit > Copy. Select the node (or nodes) at which you would like to replicate the subtree, then choose Edit > Paste. Pasted subtrees are fully independent and editable copies of the original.
- To use a cloned subtree, select the root of the subtree which will remain editable, then choose Options > Select Subtree. Then choose Edit > Create Clone, and name your clone master. Select the node (or nodes) at which you would like to duplicate the subtree, and choose Edit > Attach Clone. Clone copy subtrees are not editable directly; only the original clone master may be edited. However, changes made in the original are automatically propagated to all copies.

See Chapter 12 for more information on cloning subtrees.

Recursive subtrees may only be achieved by using a Markov node. You may not attach a clone copy to a branch emanating from the master to indicate recursion. You may, however, paste a subtree to itself using the clipboard, although this method has two drawbacks: (1) the level of recursion will only be as deep as the number of times you paste the duplicated subtree; and (2) if you make structural or value changes to the first subtree, these changes are not automatically propagated, as they are with clones.

Node types

The *Boolean* node is called a *Logic* node in DATA. See Chapter 34.

SMLTREE's *Nadir* node is implemented in DATA as a decision node with its optimal path reversed. See Chapter 14 for a discussion of this feature, and Chapter 19 for more on the optimal path setting in general.

Tables

In Decision Maker and SMLTREE, tables are stored as a part of the tree file; in DATA, tables are stored globally, separate from your trees. This makes it very easy to reuse tables. When you import a SMLTREE file which contains tables, DATA automatically create table files. You should edit the new tables' properties to determine the appropriate setup for each converted table. See Chapter 26 for more information on tables.

Tree display for medical publications

You may display (and print) trees which conform to the standards (as of 1998) of Medical Decision Making magazine. Of particular interest are Branches at right angles, which will draw branch lines vertically, and Always show payoffs / Boxed. You will find these options in the Tree Display and Node Display sections of the Preferences dialog.

See Chapter 10 for more information on changing the display of your tree, and Appendix B for the Preferences dialog.

INDEX

Symbols

! 374
# 25, 132
& 374
+ 124, 140
-directional links 149
; 82
? 164
_tunnel 262
| 374

A

ActiveX 142 *See also* bi-directional links
align
  endnodes 364
  influence diagram nodes 309
  internal nodes 104
alternatives
  assigning 21
analysis data
  creating tables from 279
  exporting 158
analysis storage 125
  analysis description 125
  deleting 126
  renaming 126
  run previous 126
  save last 125
  tornado diagram 202
AND 374
annotation boxes
  bound 107, 339
  creating 339
  drawing 106
  editing 106
  fonts 109
  moving 106
  preferences 108, 366
  resizing 106
arcs 19
  annotating 310
  asymmetry 27
  curving 310
  drawing 23
  editing 27, 310
  flipping 310, 316
  fonts 372
  gray 28, 302, 371
  influence diagram 13
  on clipboard 310
  structure-only 28, 371
  timing 300, 312, 314
arrow keys 16, 352
arrows 106, 366
asymmetry 301–302 *See also* arcs: asymmetry
  basics 26–29
  default variable values 303
  elim 301
  entering 27–29
  force 301
  skip 27, 302
  skip all 302
  symm 27, 301
attribute *See* calculation method

B

Bayes' revision 213–223
  binary hypothesis 216
  dichotomous test 216
  example 214, 216
  influence diagram
    test correlations 213
  influence diagrams 311–316
  introduction 214
  locating the subtree 223
  multiple tests 297, 315–316
  performing 217
  preparing the subtree 215
  sequential tests 213 *See also* influence diagrams: Bayes' revision
  structural requirements 216
  tree window 213
  using the dialog 217
Bayes’ revision 344
  influence diagrams *See* influence diagrams: Bayes’ revision
benefit-cost 178
bi-directional links
    142, 143, 147, 147–153, 337
  as node calculation server 142
  connection to spreadsheet 149
  described 147
  ODBC 143
  preparing your tree 152
  server 148
  trigger 152
  troubleshooting 152
  vs. DDE server links *See* bi-directional links: server
bitmap 155, 157, 334, 368
branches
  adding 15, 343
    default number 364
  at right angles 364
  decision tree 9
  deleting 41
  distributing 275
  inserting 40
  reordering 42
break link 337
buttons *See* tool bar: reference

C

calculation method 169, 356
  attribute 169
  benefit-cost 169
  cost-effectiveness 169, 175 *See also* cost-effectiveness
    maximum cost 171, 181
    minimum effectiveness 171, 181
    optimal path criteria 171, 181
    willingness-to-pay 171, 181
  expected value 169
  maximin 169
  multi-attribute 169, 175–178
  multi-attribute, generalized 169, 175
  multi-attribute, weighted 169
  optimal path 169, 170
  path probability 169
  simple 175

status bar indicator 169, 170
CE params 356
certainty equivalent 292
change node type 343
choose function 378
clipboard 38, 335, 336
copy pictures using 156
clones
influence diagram 317, 335, 337
tree window
attaching 119, 338
creating 337
described 118
destroying 118, 337
detaching 119, 338
hiding 364
maintaining 338
recursive 119
using variables 72, 73
coherence 194
cohort *See* Markov processes
command line 393, 395
comments
node 129, 343, 362
variable 87
comparative distributions 47–48
convert *See* outlines
influence diagram to tree 28–29, 369
algorithm 299–310
copy link 141
copy special 142, 156
correlated distributions 271
correlated variables 191
correlations *See* distributions: correlating
cost-benefit 178 *See also* calculation method: benefit-cost
cost-effectiveness 171, 179–185, 181, 189 *See also* calculation method: cost-effectiveness
analysis 183
calculation preferences 356
coefficient of inequity 184
decision-making 181
described 179
dominance 179, 183
extended dominance 184
graph 183, 326, 346
influence diagrams 298
numeric formatting 180
parameters 356
preparing your tree 180
sensitivity analysis 197
one-way 188
text report 183
tornado diagrams 203
create publisher 144
Cum Pct 203
cumulative distribution 47–48, 323
custom interface *See* DATA Interactive
basic mode 161, 162
calculation method 162
risk preferences 162
testing 165
change values 161, 167
DATA Interactive 159–160
extended mode 161, 165
ready for ECI 166
run analyses manually 165
users' view 167
html 159–160
Internet 159–160
stored analyses 127, 161, 165
Visual Basic 159–160
cycle length *See* Markov processes: cycle length

D

DATA 3.0 4–5
data entry 135
DATA Interactive 159–160
databases
querying 143
DEALE 379
decision analysis 1
Decision Maker 397–403
bindings 400
functions 399
importing files 397–400
Markov processes 401–402
access functions 399
tables 403
variables 400
decision node
reversing optimal path 171
decision probabilities 214
decision tree
basic calculations 11
decision trees
design 9
DecisionMaker *See* SMLTREE
Define Values (dialog box) 33, 36, 37, 81, 240, 341
deterministic nodes 305–310, 307
discounting 247, 378, 379
display
decision trees 95–111
hiding text 103
hiding values 103
influence diagrams 95–111
numeric formatting 99
scenario numbers *See* terminal node numbers
terminal node columns *See* roll back: terminal node columns
variables 101
distribute children 275
distribution statistics 324
distributions 267, 271 *See also* DistSamp function
correlating 271–272
custom 273
examples 271
formulas
beta 382
chi 382
chi-squared 383
Erlang 383
exponential 383
gamma 384
hyper-exponential 384
Laplace 384
linear (uniform) 381
log-normal 385
logistic 385
Maxwell 385
normal 381
Rayleigh 386
triangular 382
in influence diagram 308–310
limitations 271
resample at each Markov stage 269
sampling observed values 274
DistSamp function 268, 270
comment 269
index 269
Markov processes 269

resampling 269
dominance 179, 183
comparative probability distributions 48
extended 184
in sensitivity graph 190
dynamic data exchange (DDE) 122, 138, 336, 337
client 138, 140, 143
cell formatting 140
link properties 139
updating links 140
in influence diagram 308–310
server 138, 143, 148
drawbacks 142

E

Evaluator 92–93
evaluator 341
examples
drug trial Markov 238, 258
faulty machine 311
home furnace 301
investment 6, 211, 280
nested litigation trees 121
rich uncle 291
surgery Markov 261
Excel
bi-directional links 138
database connectivity 143
linking 275
ODBC 143
sampling from tables 275
using as calculation engine 143
expected value 11, 43, 346 *See also* calculation method: expected value
expected value of perfect information (EVPI) 209–212, 297, 310
defined 209
in influence diagrams 316–317
export 334, 368
exporting
analysis data 158
analysis graphic 155
graphics 155
metafile 155
tree graphic 155
extended dominance 184

F

fonts 108–110, 320, 339
annotation box 109
arcs 372
default for nodes 109, 365
default for probabilities 110
expected values 110
graph labels 320
graph template 327
influence diagram 110, 372
node 108
node comment 130
payoff formulas 110
preferences 109
subtree 108
TrueType only 368
footers 56
force path 344
functions 373–380
arithmetic 376
Abs 376
Average 376
Ceiling 376
Exp 376
Floor 376
Int 376
Ln 376
Log 376
Max 376
Min 376
Modulo 376
Prod 376
Root 376
Round 376
Sqrt 376
Sum 376
financial 376
Annuity 376
Compound 376
FV 377
NPV 377
PMT 377
PV 377
helper 82, 376
medical 378–380
DEALE 379
OddsToProb 380
ProbToRate 379
RateToProb 379
UtilDiscount 378, 379
miscellaneous 377
Choose 377
Dist *See* functions: miscellaneous: DistSamp
dist 377
DistSamp 377
If 377
Inf 378
Pi 378
Stdev 378
Sub 139, 378
syntax 375
using in formulas 81

G

graphics
bitmap 155
exporting 155
metafile 155
raster 155
vector 155
graphs 319–328
axis modification 321
custom labels 321, 350
graph template 327
custom lines 322, 350
graph template 327
data
exporting 158
exporting 158
labels
editing text 320
fonts 320
moving 320
moving 323
numeric formatting 322
resizing 323
templates 326–328
analysis storage 126, 328
applying 328
contents 327
creating 327
default 328
modifying 328
text report 197, 323
bar graph 324
cost-effectiveness 183
line graph 325

H

half-cycle correction *See* Markov processes: half-cycle correction; Markov processes: initial rewards
headers 56
health states *See* Markov processes
help 3
hide node texts 362
hide probabilities 362

I

imperfect test *See* Bayes' revision
influence diagram
  changing screen size *See* zoom
  fonts 110
influence diagrams 12–14, 19–29, 297–310, 311–318
  Bayes' revision 311–316
  Bayes' revision
    asymmetry 315
    decision node required 313
    how it works 314
    multiple tests 315
  clones 317
    create 317
    destroy 317
  collapsing 317
  conversion algorithm 299–310
  EVPI 316–317
  fixed node size 370
  graphical position of nodes 309, 369
  hiding nodes 317
  limitations
    cost-effectiveness 298
    Markov processes 298
    multi-attribute 298
  Markov processes 298
  mini-tree *306*
  Monte Carlo simulation 299
  multi-attribute 298
  node description 309
  payoff formula 304
  sub-models 317
  when to use 297–299
intellectual property 167
international settings 394
isocontours 198–202
  cost-effectiveness 198
  labels 198
  limitations 198

L

label nodes 103
life cycles *See* Markov processes
likelihood probabilities 214, 215
line graphs 324
links 137–145
  bi-directional 143, 337 *See also* bi-directional links
  copying 141
  properties 139
    assign index 139
    path 140
    replace path 140
logic nodes 136, 329

M

marginal values 45
  sensitivity analysis, *See* isocontours
  zero marginal 190
Markov analysis 236
Markov cycle tree 227
Markov node 227, 228
Markov processes 102, 119, 225–238
  _stage 247
  _tunnel 262
  absorbing state 228, 230
  basic elements 226
  basic models 225–238
  clones 255–259
    entire subtree 258–259
    Markov bindings 255
  cohort simulation 226
  cost-effectiveness 253
  counters 234
  cycle 234
  cycle length 226, 233, 233–234
    reward units 233
  decisions 264–265
    limitations 264
    logic nodes 264
    using logic 264
  discounting 247–249
  displaying values in tree 363
  distributions
    resampling 272
  final rewards 248–251
  graphing 346
  half-cycle correction 250–255
  health states 225
  incremental rewards 231
  initial distribution 229
  initial probabilities 226
  initial rewards 248–251
    after cycle zero 251
  jump-to dialog 230
  keywords 234
  life expectancy 232
  logic nodes 265
  Markov analysis 236
    graphs 236
    text report 236
  Monte Carlo
    distributions 269
  Monte Carlo simulation 226, 237–238, 263, 263–264
    resampling distributions 272–273
    technical details 392, 392–393
    termination options 360
    tracker modifications 265
    tracker variables 284
  payoff calculator 227
  prior costs 249
  probabilities 226
  recursive model 225
  rewards 226, 231–232
  root node 228
  stage 234, 247
  stage reward 234
  state rewards 226, 231
  state transitions 226
  states 225
    adding 229
    names 229
  technical details 391–393
  temporary states 259–260
  termination condition 227, 233–234, 247
  time-dependent rewards 233
  tolls 252
  total reward 234
  trace 346
  tracker modifications 263

tracker variables 284
transition nodes
hiding state name 230
transition probabilities 228
transition rewards 251–253
transition subtree 228, 261
transitions 226, 229
creating 229
terminal nodes 228
transitions and cycle length 231
tunnel states 259–262
UtilDiscount 248
violating Markov strictures 263–266
maximin 172
maximize 170 *See also* calculation method: optimal path
maximum value of information *See* expected value of perfect information (EVPI)
menus 333–353
Analysis 345–348
Diagram 348–349
Display 339–340
Edit 335–338
File 333
Graph 350–351
Options 342–345
Table 351
Values 341–342
metafile 155, 157, 334, 368
mimic run-time 345
mini-tree 24, *306*
minimize 170 *See also* attribute; calculation method: optimal path
minimize empty space 104, 364
Monte Carlo
Markov processes *See* Markov processes: Monte Carlo
Monte Carlo simulation 279–289
chance nodes 280
cost-effectiveness
marginal values 185
decision nodes 282
decisions 280
details 280
dialog 282
first-order 280–281
graphs 283
Markov processes *See* Markov processes: Monte Carlo simulation
probabilistic distributions 279
reevaluate optimal path 282
sampling distributions 282
second-order 281–282
sensitivity analysis 208
statistical analysis 279
summary statistics 281
text report 283
tracker variables 283–288
as outputs 284
Markov processes 284
trackers 89
using distributions 282
variables
trackers 89
mouse
right button 23, 101
multi-attribute *See* calculation method: multi-attribute, generalized
numeric formatting 175
setting up 176
using Choose function 378
viewing singly 177

N

navigation button 352
nested trees 120
nks
-directional 149
node comment *See* comments: node
nodes
decision tree
chance 9
decision 9
terminal 9
influence diagram
chance 13
decision 13
value 13
label 103
Markov 227
terminal
aligning 103
normalizing probabilities 195
NOT 374
note boxes 106 *See also* annotation boxes
novice mode 393
numeric formatting 44, 53, *99*
cost-effectiveness 180
currency 99
decimal places 99
graph axis 322
graph template 327
influence diagrams 369
multi-attribute 175, 177
probabilities 100
setting 357
units 99

O

objectives *See* calculation method
ODBC 143
operators 373–380
arithmetic 373
logical 374
precedence 374, 374–375
relational 373, 373–374
optimal path 173 *See also* calculation method: optimal path
changing 136, 344
high or low 17, 136, 357
sensitivity 60
showing 135, 348
OR 374
outcomes
assigning 21
over/under 330, 346

P

Paste link *See* dynamic data exchange (DDE): client
paste link 336
automatic 139
Links dialog 139
path probability *See* calculation method: path probability
payoff
boxes *See* roll back: boxes
payoff range 347
payoffs *See* calculation method *See also* calculation method
displaying 95, 361
multi-attribute 17, 177

multiple 177
node name as 360
using variables 59
perfect predictor 209
PICT 156, 157, 334, 336
pictures
editing 155
exporting 155
preferences 105, 134, 355–372
align endnodes 103
calculation method 169, 356–357
cost-effectiveness 180
fonts 109
global (Windows) 368
influence diagram 369–372
arcs 371
fonts 372
node size 370
notes & arrows 366
other calculation 360
printing 57, 367, 368
risk 296, 359
roll back 358
terminal node display 361
tree 355–372
tree display 364
tree font 365
variables display 363
print preview 55, 334
printing 55
center in page 58, 367
headers & footers 56, 367
offsetting the document 56
page breaks 57, 367
problems 200
show headers 58, 367
use printer for sizing 368
zoom 56, 58, 367
probabilities 214 *See also* #
complementary 134, 360
conditional 44
display 100, 358
estimating 59
fonts 110
hiding 103, 362
normalizing 195
numeric formatting 100
path 44, 347
using variables 59
probability distribution 47, 323, 345
comparative 47–48
probability revision *See* Bayes' revision
probability wheel 164, 167
in influence diagram 308
properties dialog 86
publish & subscribe 143
edition 337
publisher 138, 143, 337
subscriber 137, 138, 143, 337
publish & subscribe (P&S) 122

Q

QALY 100, 180, 271
queries *See* ODBC & Excel

R

rankings 45
cost-effectiveness *See* cost-effectiveness
recursive processes 225 *See also* Markov processes
recursive trees 119
recursive variables *See* variables: recursive
redo 133, 335
region of optimality 197 *See also* sensitivity analysis: two-way
reports
variables 85
revert 132, 334
Risk Pct 203
risk preferences 291–296
certainty equivalents 292–296
coefficient 293
constant risk-aversion 293
assigning 294
non-constant risk-aversion 293
assigning 294
turning on 359
ROC curve *See* Bayes' revision: test cutoff points
roll back 43, 45, 100, 348
boxes 173, 296
contents 44–54, 171
fonts 365
moving 44, 101
selective display 100, 101, 358
calculations
changing displayed value 171
expected value 171
maximin 171
path probability 171
defined 11
display 96–98
options 100
display preferences 95
preferences 171
terminal node columns 96–98
copying to clipboard 96–111
cost-effectiveness 98
marginal values 98
setting up 96
run-time version 160, 167, 345

S

sampling
tables 274
scientific notation 135
scientific notations 100
scripting 334
select if 115, 342
sensitivity & specificity *See* Bayes' revision: tree window
sensitivity analysis 50–51, 345
advanced one-way 188
at chance nodes 187
avoiding errors 387–390
cost-effectiveness 188
described *50*
deterministic nodes 305
graphs, interpreting 52
how it works 387
influence diagram 305
legend 52
line graph 52, 324
marginal values 182 *See also* isocontours
on probability variables 73
on risk preference coefficient 296
options 190
preparing 88
sequential decisions 187
sliders 93
strategy graph 325, 326
three-way 199, 322
threshold analysis *See* threshold

analysis
thresholds 52–53
caveats 53
tips 73, 193, 387–390
tornado diagram 200, 326–328
two-way 195
indifference, regions of 196
isocontours 198
requirements 195
using Choose function 378
show inherited 90
simple calculation method 175
simulation *See* Markov processes: Monte Carlo simulation
skip generation 104, 339
sliders *93*, *93–94*, 208, 341
SMLTREE 397–403
bindings 400
functions 399
importing files 397–400
Markov processes 401–402
access functions 399
tables 403
variables 400
snapshots 334
Spread 203
spreadsheet 143, 147, 239, 242
exporting data to 158
links 337
moving *See* links: properties: replace path
named cells 140
SpreadSqr 203
standard deviation 45, 347
start-up 393, 395
state bindings *See* Markov processes: clones: Markov bindings
status bar 169, 170, 296, 340, 353
calculation method 177
graph windows 323
progress indicator 358
reference 353
strategy graph *See* sensitivity analysis: strategy graph
Sub *See* functions: miscellaneous: Sub
Sub function 137
sub-models 317
subscriber list 144
subtrees 40
fonts 108
selecting 38, 113, 342
using the clipboard 119

T

tab key 32
tables 239
contents 239
creating 240
custom distributions 273, 274
editing manually 241, 351
files 239, 391
format 239
from line graph 243, 325
importing 242
index off edge is error 240
properties 240, 241
referencing in formulas 240
storage location 239, 391
technical notes 387–395
technical support 3
templates *See* graphs: templates
terminal node columns *See* roll back: terminal node columns
terminal node display 361
terminal node numbers 98
terminal nodes
aligning 103
tests 214
text report *See* graphs: text report
threshold analysis 325, 346 *See also* sensitivity analysis
non-linearity 205
tolerance 205
thresholds
sensitivity analysis
two-way 198
tool bar 20, 24, 131, 340
reference 352
tornado diagram 200–204, 345
summary calculations 203
tornado diagrams
correlated variables 204
tracker variables 89 *See also* Monte Carlo simulation: tracker variables
TRB files 334
tree
calculation method 169
changing screen size *See* zoom
tree clipboard 38, 335, 338
tunnels *See* Markov processes: tunnel states

U

uncertainty 49–54
undo 132, 335
utility functions 292 *See also* risk preferences

V

value nodes
assigning a formula 304
enumerating 305
value of information *See* expected value of perfect information (EVPI)
values
estimating 59
hiding 103
variables 31, 59–78, 79
box 85
custom interface 167
font 110, 365
concepts 59–78
correlations 88, 191–192
debugging 92–93 *See also* evaluator
default definition 87
default for tree 33, 80, 85
defining 33, 79–80, 87, 91
displaying in tree 84, 85, 87, 363
Evaluator 92–93
global *See* Monte Carlo: trackers
in influence diagram
defining in mini-tree 307
deterministic nodes 305
hiding 303
nodes 302–303, 303
in payoff formulas 178
in probabilities 72
inserting 81, 83
multi-attribute 178
overriding 33, 37
preferences 84, 85
properties 86–89
comment 87
correlations 88
hiding definitions 87

name 86
sensitivity analysis 88–89
short description 87
value range 88
recursive 75–78
complex recursion 77
search algorithm 67, 285
sensitivity analysis, in 51
short description 80
theory 59–78
tips 72, 73, 81, 178, 193
trackers 89 *See also* Monte Carlo: trackers
using in formulas 81
where to define 33, 81
window 89–92
variables window 89–92, 342
navigation arrows 91
show inherited 90
verify probabilities 348

W

weighting function 178
willingness to pay *See* cost-effectiveness: calculation preferences

Z

zoom 25, 110
print 58