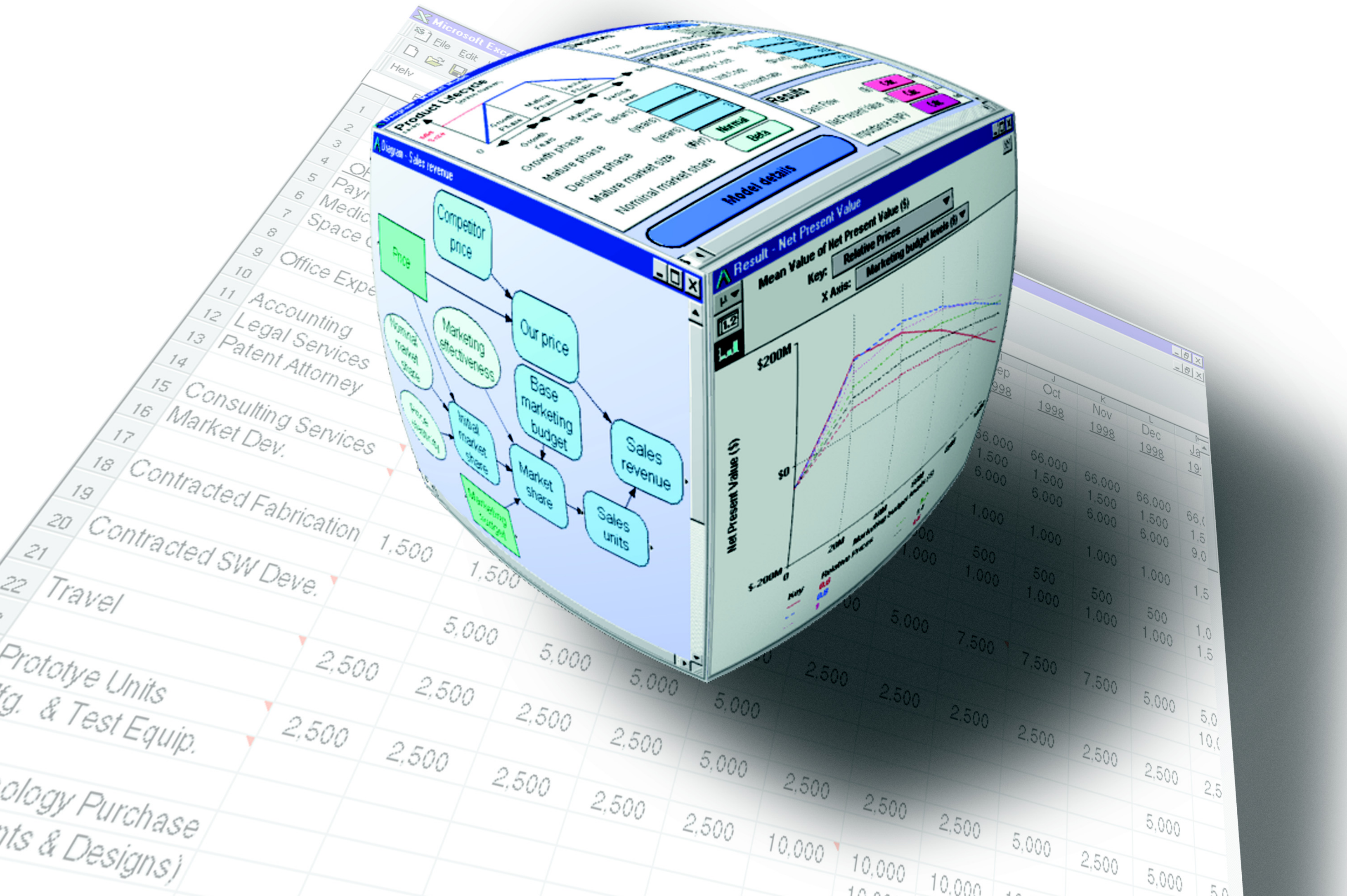
analytica®
Beyond the Spreadsheet

Tutorial

Release 4.1



Lumina Decision Systems, Inc.
26010 Highland Way
Los Gatos, CA 95033
Phone: 650-212-1212
Fax: 650-240-2230
Web Site: www.lumina.com

Copyright notice

Credits

The Analytica Tutorial was written by Brian Arnold and Lynda Korsan with Max Henrion and Randa Mulford (Expert Support, Inc.). New releases have been edited by Lonnie Chrisman and Richard Morgan.

Contents

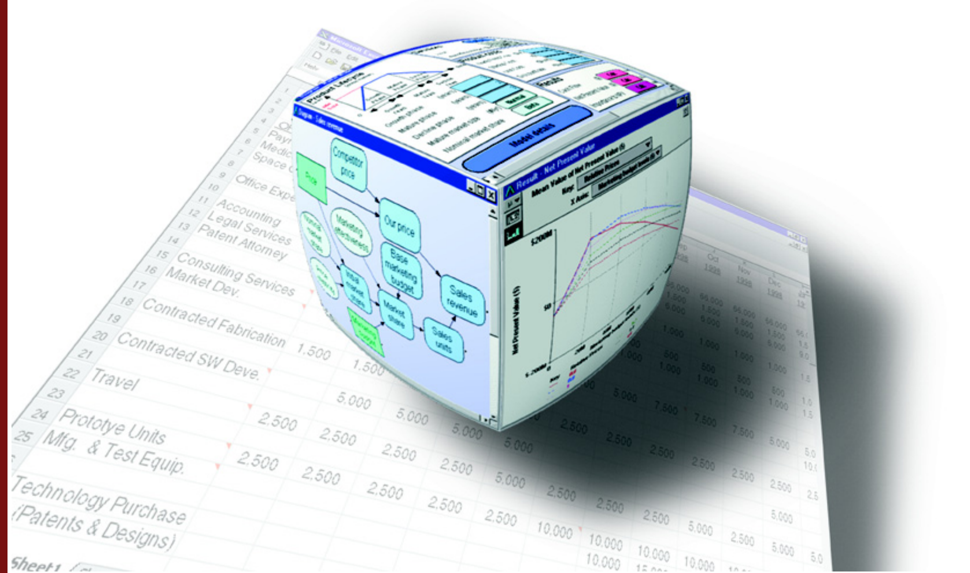
Contents

Contents

Introduction

About Analytica

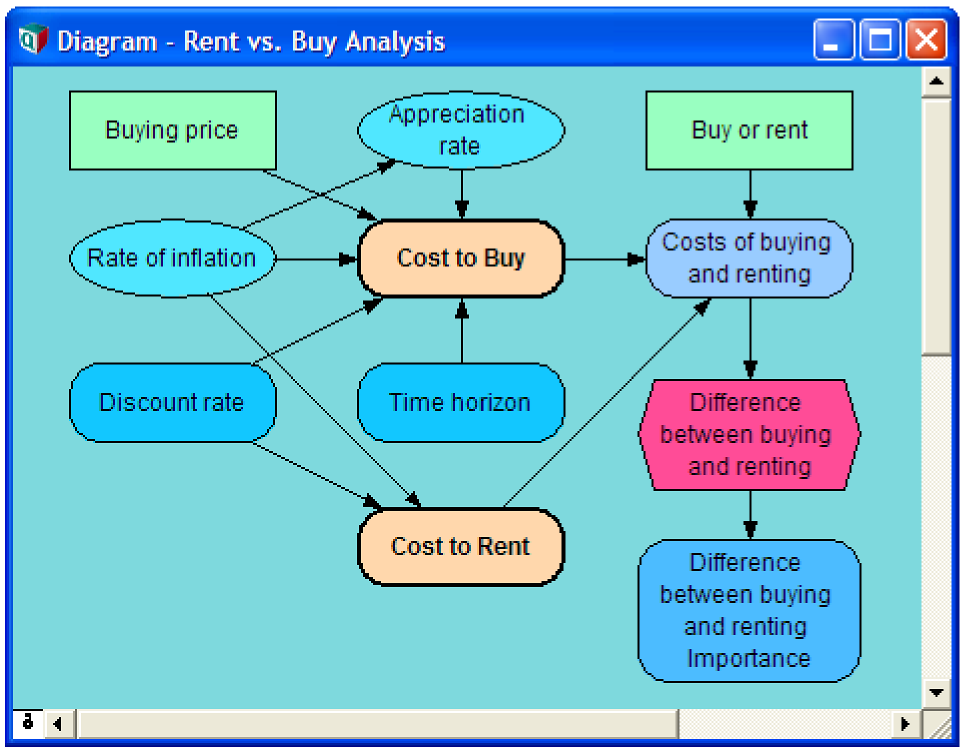
This section introduces Analytica and its uses, explains what is included in this tutorial, and tells you how to use this manual.



Welcome to Analytica

You are about to discover a powerful tool for real-world modeling and analysis. Analytica embodies the idea of using a white board for problem solving. Using a visual, point-and-click approach, you draw nodes and arrows to depict the relationships between model components. This approach allows you to describe the essential *qualitative* nature of the problem without getting lost in the details. As the model develops and your understanding of the problem becomes clear, you can define the exact *quantitative* details of the model.

A key feature of Analytica is its ability to create hierarchies of models. By grouping related components of a problem into separate submodels, you can impose a multi-level organization to your model. This helps you to manage complex relationships and allows other users to more easily grasp important concepts.



Another key feature of Analytica is the use of ***Intelligent Arrays***™. These enable you to add or remove dimensions such as time periods, geographic regions, alternative decisions, etc., with minimal changes to the model structure. Unlike spreadsheets, which require you to repeat formulas with each new dimension, Analytica separates the dimensions from the relationships so that models remain simple. As the dimensions change, Analytica automatically updates, reports, and graphs the results.

Each node, or object, in an Analytica model has a window that displays the node's inputs and outputs, and allows you to enter definitions, descriptions, units of measure, and other documentary information. This self-documenting capability, combined with hierarchical models and Intelligent Arrays, makes it easier to understand and communicate how models work.

Analytica features fully integrated risk and sensitivity analysis for analyzing models with uncertain inputs; powerful facilities for time-dependent, dynamic simulations; powerful graphing capabilities; and over 200 financial, statistical, and scientific functions for calculating just about any type of mathematical expression.

Who can use Analytica

Analytica is for the modeler and problem solver — from the financial analyst modeling business opportunities to the engineer designing new products to the scientist investigating the behavior of physical phenomena.

It is particularly suited to users in the fields where you have to reason with uncertainties, or arrays of data, or both, i.e., management consulting, health and environmental sciences, aerospace, oil and gas, construction, manufacturing, financial services, and investing.

Tutorial overview

This tutorial is a hands-on introduction to using Analytica. Step-by-step instructions show you how to explore and analyze an existing Analytica model and how to create a new Analytica model. Because later tutorial sections build on the material in earlier chapters, you should work through the chapters in order.

We recommend that everyone new to Analytica complete Chapters 1 through 5, which takes two to three hours. If you want to work more quickly, skip the text and only follow the instructions in the boxed steps. Then, if you are unsure about any terms or concepts, look them up in the Glossary or review the text. And before you start your own modeling, you should review Chapter 8, which describes the sample models included with Analytica. This way, you can benefit from examples similar to what you might be modeling.

This tutorial is designed to introduce you to some of Analytica's basic features. When you are familiar with the basics, refer to the *Analytica User Guide* for more detailed information on Analytica's features.

- **Chapter 1: Using the Rent vs. Buy Model**

  This chapter shows how to open and run an Analytica model. Using a simple interface to an example model that analyzes the total costs of buying or renting a house, you will calculate results and change input values to see the effects on the results. You will display uncertain results in a variety of ways.

- **Chapter 2: Exploring the Rent vs. Buy Model**

  This chapter shows you how to browse a model's structure and assumptions by examining its influence diagrams, variables, and definitions.

- **Chapter 3: Analyzing the Rent vs. Buy Analysis Model**

  This chapter shows you how to perform importance analysis and sensitivity analysis to see which uncertain variables most heavily influence the outcome.

- **Chapter 4: Creating Models**

  This chapter shows you how to create a new Analytica model. In the process of building a model that analyzes the costs of owning and operating an automobile, you will create variables, define relationships between variables, add documentary text, and compute results. In addition, you will create modules and add dependencies between modules.

- **Chapter 5: Creating Arrays (Tables)**

  This chapter shows you how to add ***index variables*** and ***edit tables*** (these will be defined later) to a model, and demonstrates how tables work in Analytica, including an introduction to table functions.

- **Chapter 6: Creating the Party Problem Model**

  This chapter walks you through a familiar problem: where to have your next party. This model introduces probability tables and conditional deterministic tables. You should complete this chapter if your models will use discrete or conditional uncertainties.

- **Chapter 7: Creating the Foxes and Hares Model**

  In this chapter you create a dynamic model of population sizes that depend on each other and that change with time. You should complete this chapter if your models will use dynamic simulation or variables that change over time.

- **Chapter 8: On Your Own**

  This chapter briefly describes all the example models provided with Analytica. You should investigate these as you begin to build your own models.

Installing Analytica

Before you start this tutorial, follow these steps to install the Analytica application and associated model files on your computer.

Installing from the Web

1. Go to the Lumina web site: www.lumina.com.
2. Click the **Download Software and Manuals** link.
3. Read the instructions on the web page that comes up.
   Click the **Setupana.exe** link next to the **Analytica 4.1 for Windows** heading.
4. A file download dialog box appears. Click **Run** to download the installation program and start running it to install Analytica. Or you can click **Save** to download the installation program to run later.
5. If you clicked **Run**, the Windows Installer should automatically start up and begin installing Analytica.

The setup program requires some responses from you. For example, you are asked to verify the directory name in which Analytica will be installed. Most users can accept the defaults provided by the setup program. The default installation location for Analytica is **`C:\Program Files\Lumina\Analytica 4.1`**.

Installing from a CD-ROM

1. Start Windows.
2. Insert the Analytica CD in your computer's CD-ROM drive.
3. The Windows Installer should automatically start up and begin installing Analytica.

If the AutoRun function does not work, follow these alternate steps:

4. Click the **Start** button on the Windows taskbar.
5. Select **Run** from the popup menu.
6. In the **Run** dialog box, specify the program **`SETUP.EXE`** on your CD-ROM drive (usually either the D: or E: drive).
7. Click **OK**.

The setup program requires some responses from you. For example, you are asked to verify the directory name in which Analytica will be installed. Most users can accept the defaults provided by the setup program. The default installation location for Analytica is **`C:\Program Files\Lumina\Analytica 4.1`**.

Conventions used in this tutorial

The conventions used in this tutorial are as follows:

- Boxed, numbered instructions along the left side of the page give you the steps to take.

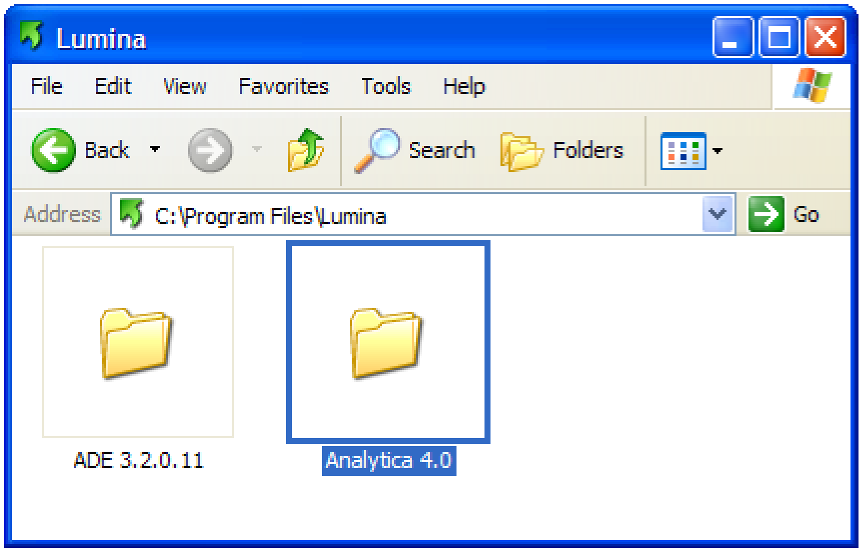
  Explanatory text often follows a step, accompanied by pictures of the Analytica screens similar to the ones you see on your computer.

For example:

1. This is an instruction.

2. In a step, ***this is the text you will enter.***

   This is explanatory text.



- Variable and model titles are displayed in *italic type*.
- The following keys on the keyboard are shown in *italic type*: *Enter*, *Return*, *Tab*, *Delete*, *Shift*, *Alt*, *F1*.
- Special Analytica terms are displayed in ***bold italic type***; they are defined when they are first introduced.

For your reference, a glossary at the end of the tutorial defines the terms used in this tutorial.

Tips alert you to useful or important information. They look like this:

Tip These alert you to useful or important information.

Assumed background

This tutorial assumes that you already have the basic skills needed to run Windows programs, including the following:

Term	Meaning
click	Press and release the mouse button one time.
double-click	Quickly press and release the mouse button two times.
drag	Press and hold down the mouse button while moving the cursor to a new location on the screen, then release the mouse button.
press	Press and hold down the mouse button.
select	Click an interface object, such as a menu command or a cell in a table; selected objects usually appear highlighted.

You also need to know how to use pull-down and popup menus, scroll bars, and windows.

If you are not familiar with these basic operations, look at the reference material that came with your computer.

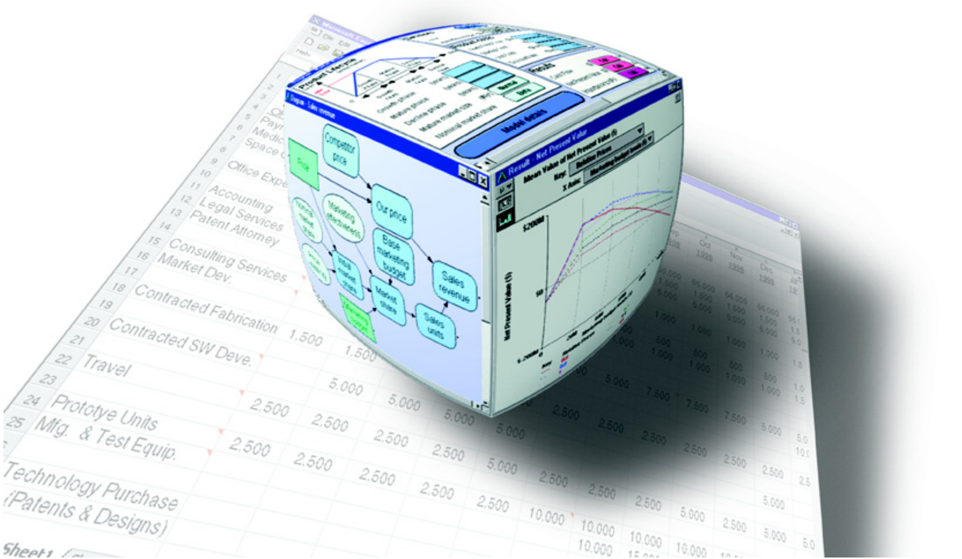
This tutorial also assumes that you have basic skills of financial or quantitative modeling — for example, from previously using a spreadsheet program.

It assumes that you are acquainted with elementary statistics and are comfortable with the concepts of mean, median, and standard deviation. It also assumes that you have some understanding of probability distributions, such as the normal and uniform, and are familiar with the concepts of probability density function and cumulative distribution function. These terms are reviewed briefly in the Glossary at the end of the tutorial.

Chapter 1 *Using the Rent vs. Buy Model*

This chapter shows you how to:

- Open an existing model
- Calculate results
- Change input values to calculate different results

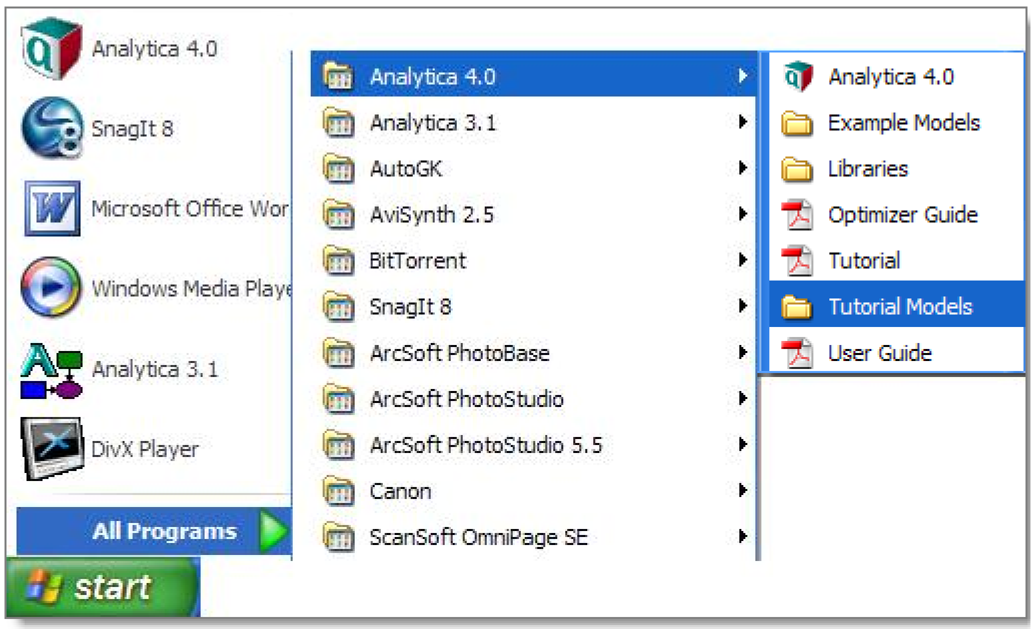


In this chapter, you use the *Rent vs. Buy* model, an Analytica model that compares the cost of renting a house to the cost of buying one. After working through the chapter, you will know how to open an existing model, use it to calculate results, and change input values to calculate different results.

Opening the Rent vs. Buy model

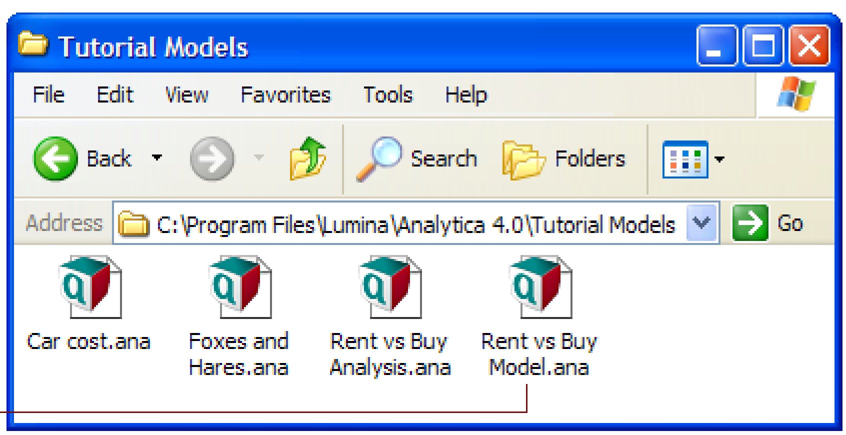
To begin, follow these steps.

1. Click the **Start** button on the Windows taskbar.
2. Click **All Programs** in the **Start** menu.
3. Click **Analytica** in the **Programs** menu.
4. Click **Tutorial Models** in the **Analytica** menu.



5. Open the *Rent vs. Buy* model.

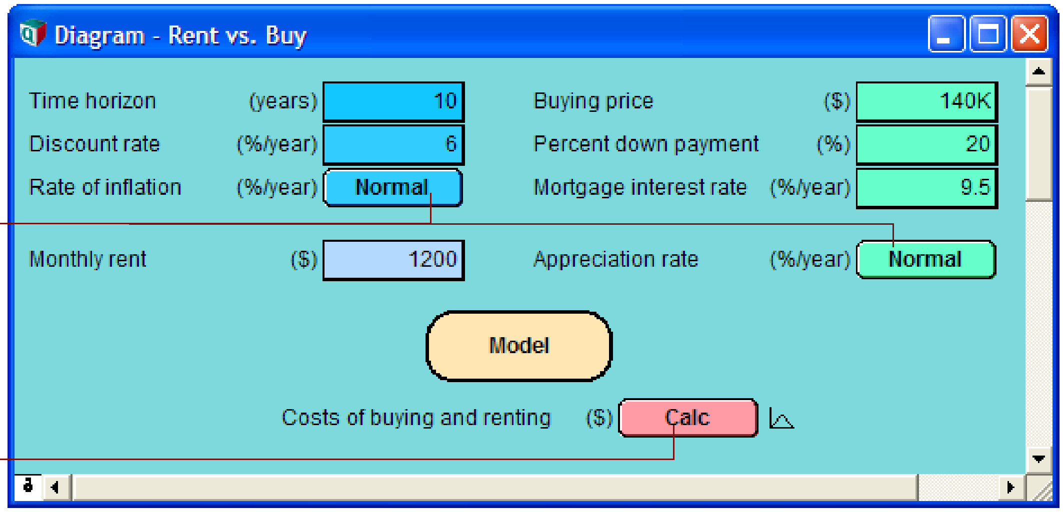
1. Double-click the icon for the *Rent vs. Buy* model to start Analytica.



Analytica reads in the *Rent vs. Buy* model.

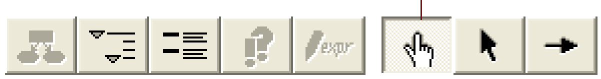
Becoming familiar with the Diagram window

When you open a model, Analytica first displays a top-level **Diagram** window. The *Rent vs. Buy* model diagram shows several input variables that affect the trade-offs between renting and buying, **Normal** buttons, a **Calc** button, and a node labeled *Model*.



This top-level diagram is an end-user interface to the model itself, which is contained in the *Model* node. In this chapter, you use only the interface in this top level diagram; in the following chapters you will explore the model in more depth.

Across the top of the screen is a horizontal palette of buttons. This is called the ***tools palette***.



When you first open the *Rent vs. Buy* model, the ***browse tool*** is highlighted on the palette. With the browse tool selected, the cursor looks like a hand when it is over the diagram. The browse tool allows you to calculate the model, change input values, and examine — but not change — the structure of the model. In this chapter, you only use the browse tool.

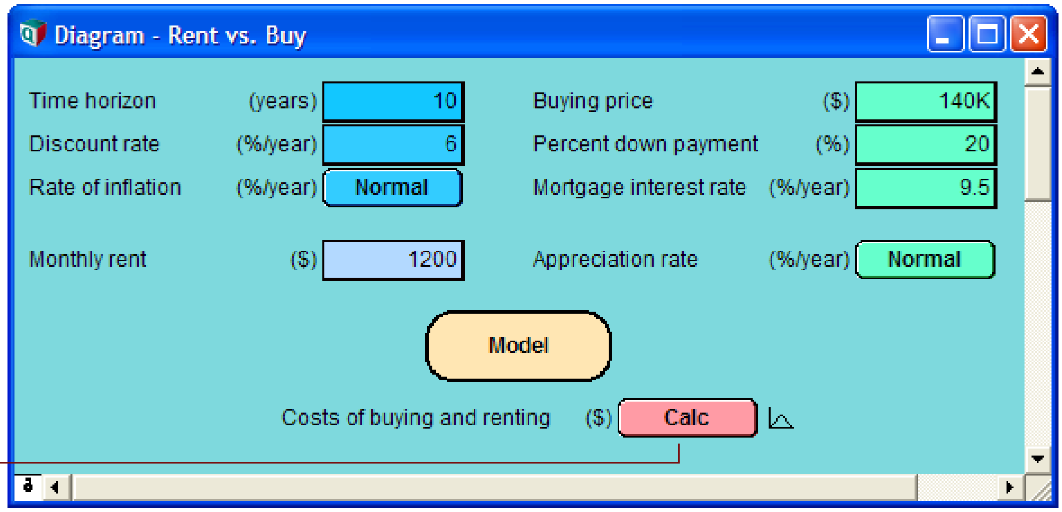
Using Online Help

At any time, you can press the *F1* key on the keyboard or use the **Help** pull-down menu to access Analytica's online help system.

Computing output values

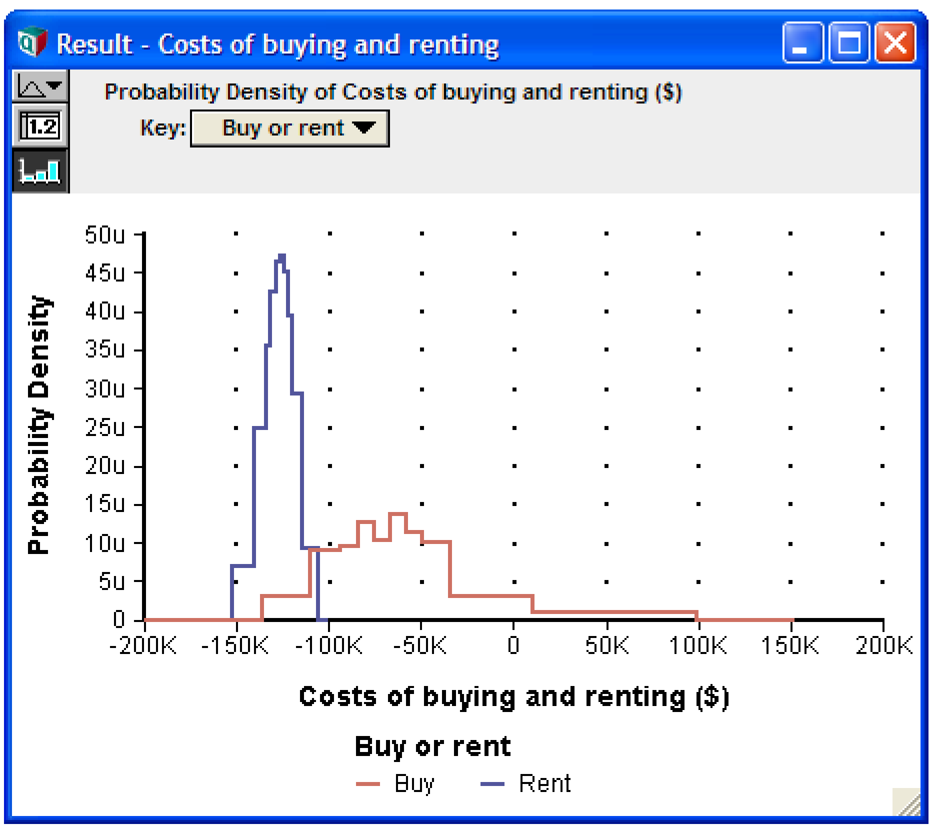
In the *Rent vs. Buy* model, the output value of interest is at the bottom, *Costs of buying and renting*.

1. Click the **Calc** button to compute the comparison of the cost of buying to the cost of renting.



The output value displays in a **Result** window. This **Result** window shows a graph of two ***probability density*** curves, one for buying and one for renting. In a probability density graph, the units of the vertical scale are chosen so that the total area under each curve is 1 (100%). \$25µ corresponds to \$25 x 10^{-6} or \$0.000025.

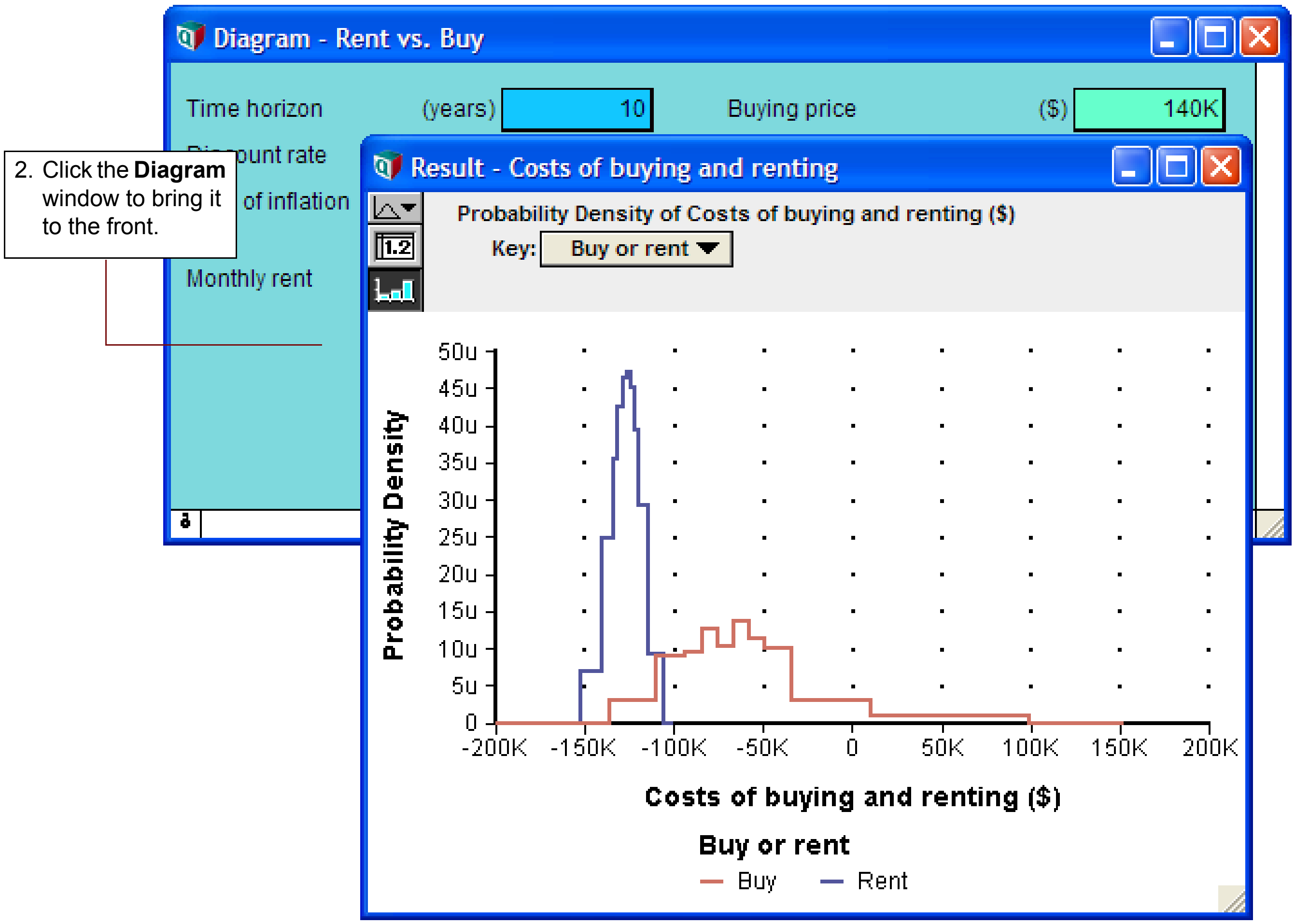
Tip Numerical suffixes like µ and K are used extensively throughout Analytica. A quick reference for these suffixes is given on the back page of this tutorial.



Since the graph is of probability densities, both buying and renting have probabilistic, or uncertain, inputs. The probability density graph for Rent appears to be a bell-shaped curve (***normal*** distribution). The graph for Buy appears “noisy,” with two peaks, indicating it is a more complicated function of uncertain inputs.

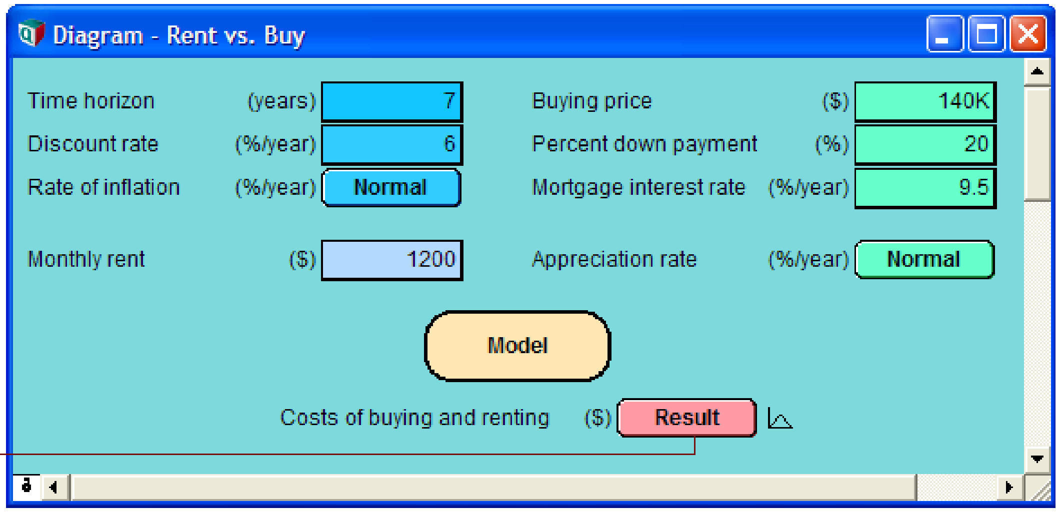
This probability density graph appears jagged because Analytica is using a sample size of 100. A larger sample size would produce a smoother curve, but the curve would take longer to compute. You can also make this graph appear smoother by increasing the number of samples per plot point. See “Uncertainty View Options” in the *Analytica User Guide* for more information.

The graphs show that the cost of renting, given the model’s inputs, are between about $105,000 and $155,000 (the negative numbers mean cost — cash flowing out), while the cost of buying is between $125,000 and a gain of $100,000.



Note: *Your results can vary slightly, since the model is generating random inputs based on a normal distribution for the uncertainty of the rate of inflation and for the appreciation rate.*

Click the model **Diagram** window to bring it to the front. Notice that the button next to *Costs of buying and renting* has changed to **Result**. The **Result** button indicates that the value has been computed; clicking the **Result** button re-displays the computed values.

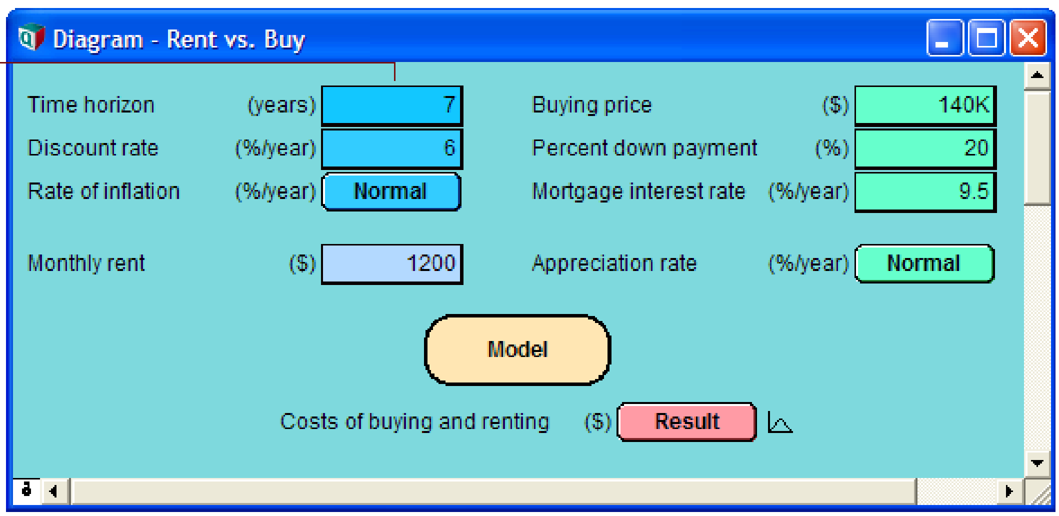


The **Calc** button has changed to **Result**

Changing input values and recomputing

Now you will change some input values to the model and recompute the rent vs. buy comparison. You will change the values of *Time horizon*, *Monthly rent*, and *Buying price*.

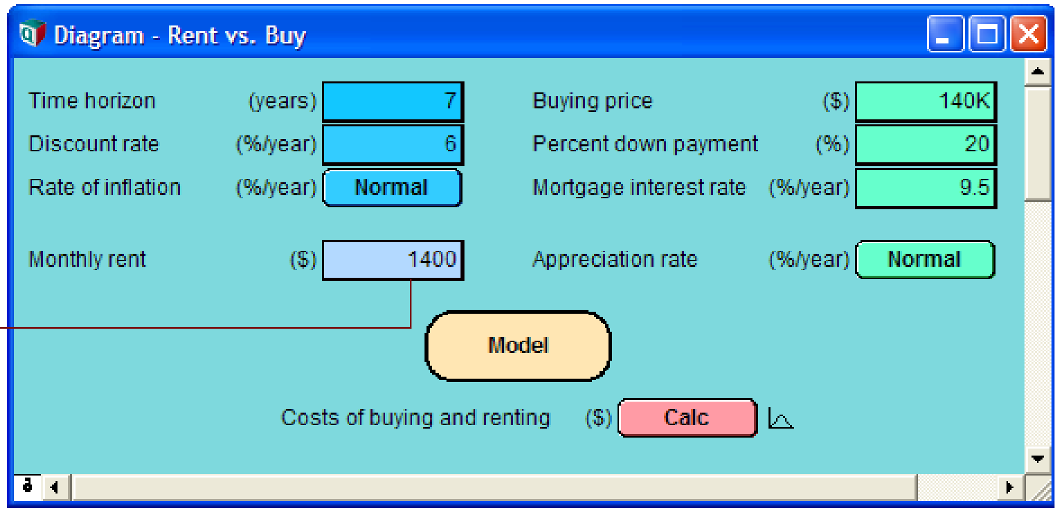
1. Click the box next to *Time horizon*. Change the value to ***7*** and press *Alt+Enter*.



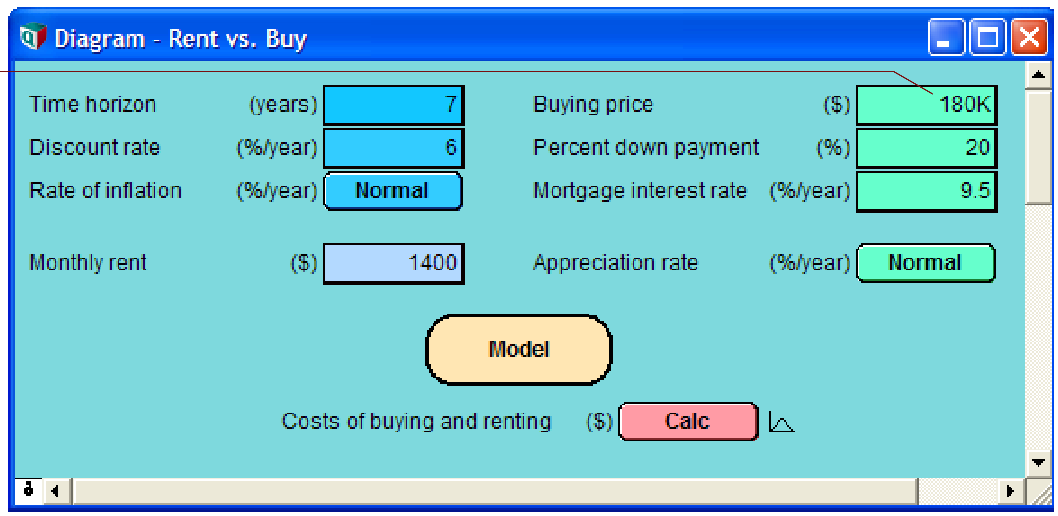
The main *Enter* key and the numeric keypad *Enter* key are not interchangeable. They have different functions in Analytica. *Alt+Enter* is equivalent to the numeric keypad *Enter* key.

As soon as you change an input, the **Result** button changes to a **Calc** button, indicating that *Costs of buying and renting* needs to be recomputed.

2. Click the box next to *Monthly rent*. Change the value to ***1400*** and press *Alt+Enter*.

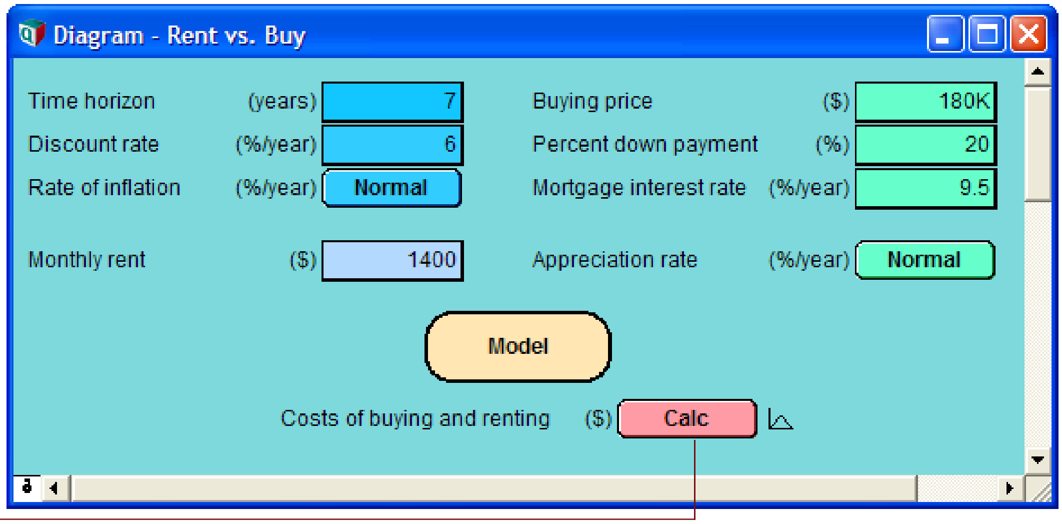


3. Click the box next to *Buying price*. Change the value to ***180K*** (or 180000) and press *Alt+Enter*.



Now you are ready to recompute to see the new results.

4. Click the **Calc** button to compute the comparison of the cost of buying to the cost of renting.



The graphs show that the cost of renting, given these changed inputs, is between $90,000 and $120,000, while the cost of buying is between $135,000 and a gain of $70,000.

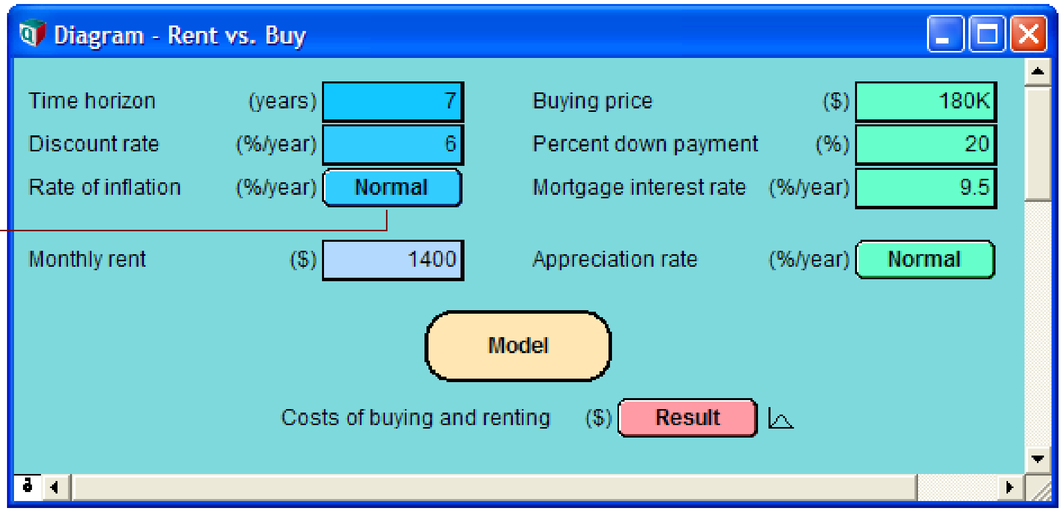
5. Click the **Diagram** window to bring it to the front.

Examining and changing uncertain input

When an input is defined as a probability distribution, a button with the name of the distribution appears next to the input's name. Clicking this button opens the **Object Finder** window, in which you can see details and change the distribution's parameters or type of distribution.

Rate of inflation's button says **Normal**, indicating that it is defined as a normal distribution.

1. Click the **Normal** button next to *Rate of inflation*.

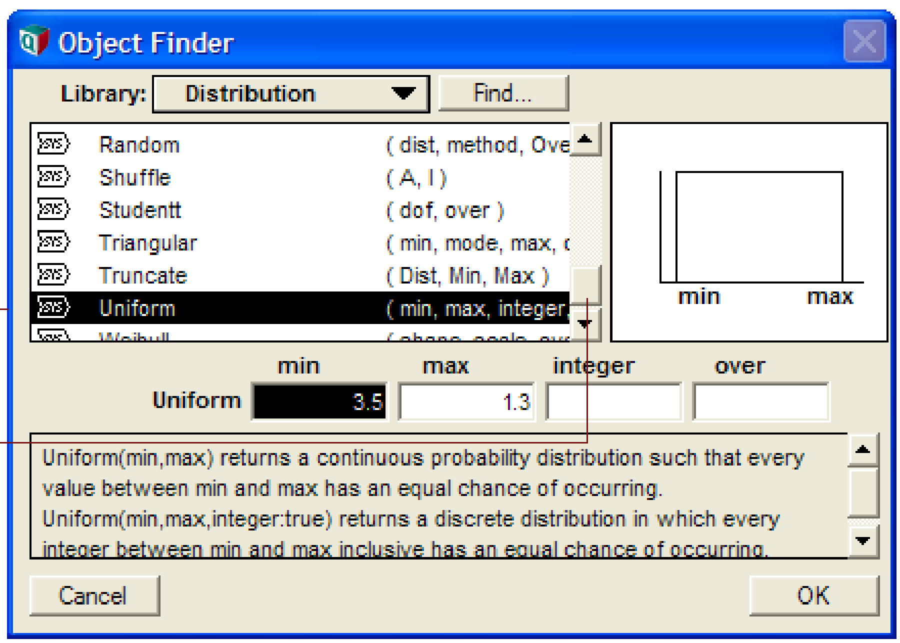


The **Object Finder** window appears. It shows that *Rate of inflation* is defined as a normal distribution with a ***mean*** of 3.5 and a ***standard deviation*** of 1.3.

You will now modify the probability distribution that defines *Rate of inflation*. Rather than using the normal distribution, you will use the uniform distribution, and assume that inflation has an equal probability of being anywhere between 3% and 4% per year.

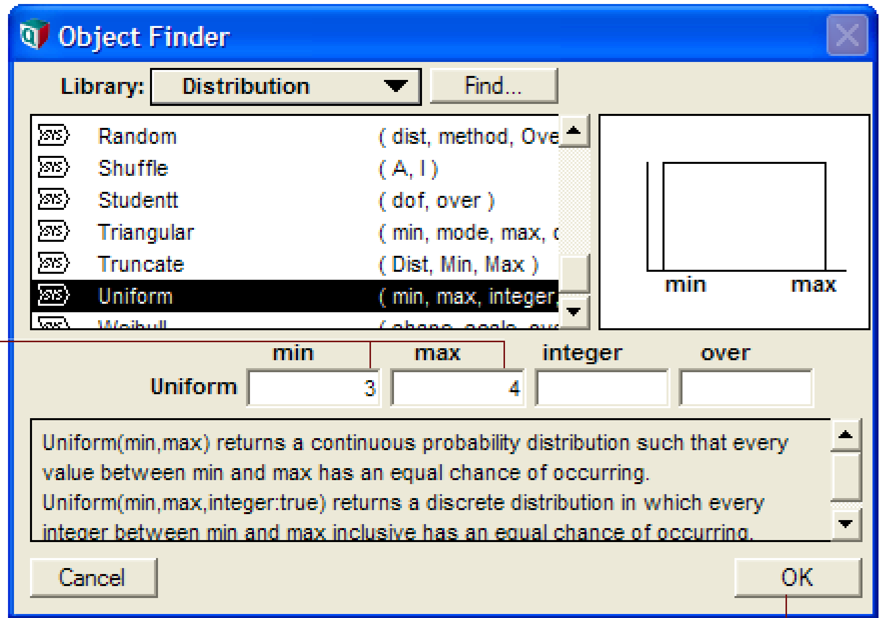
2. Scroll down the list of distributions and select **Uniform**.

Scroll bar

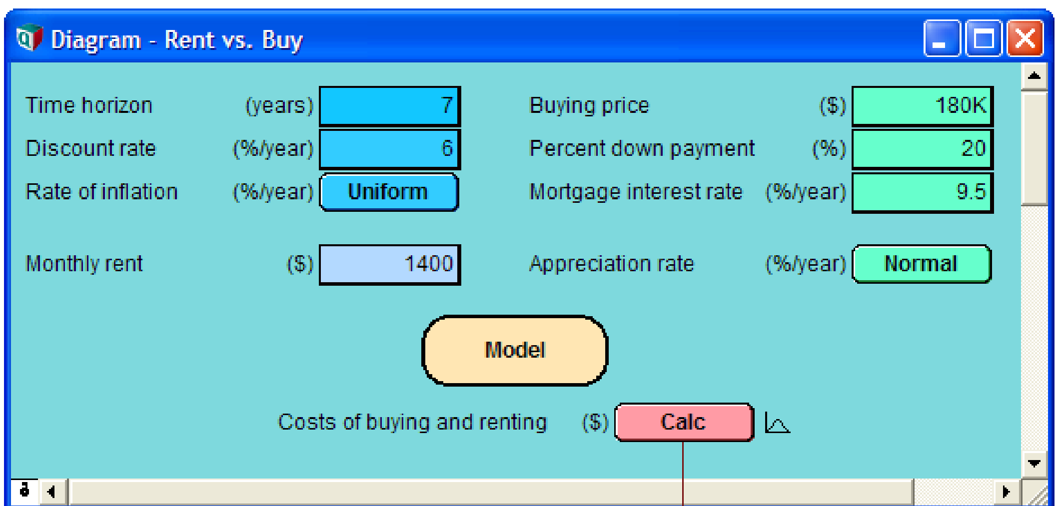


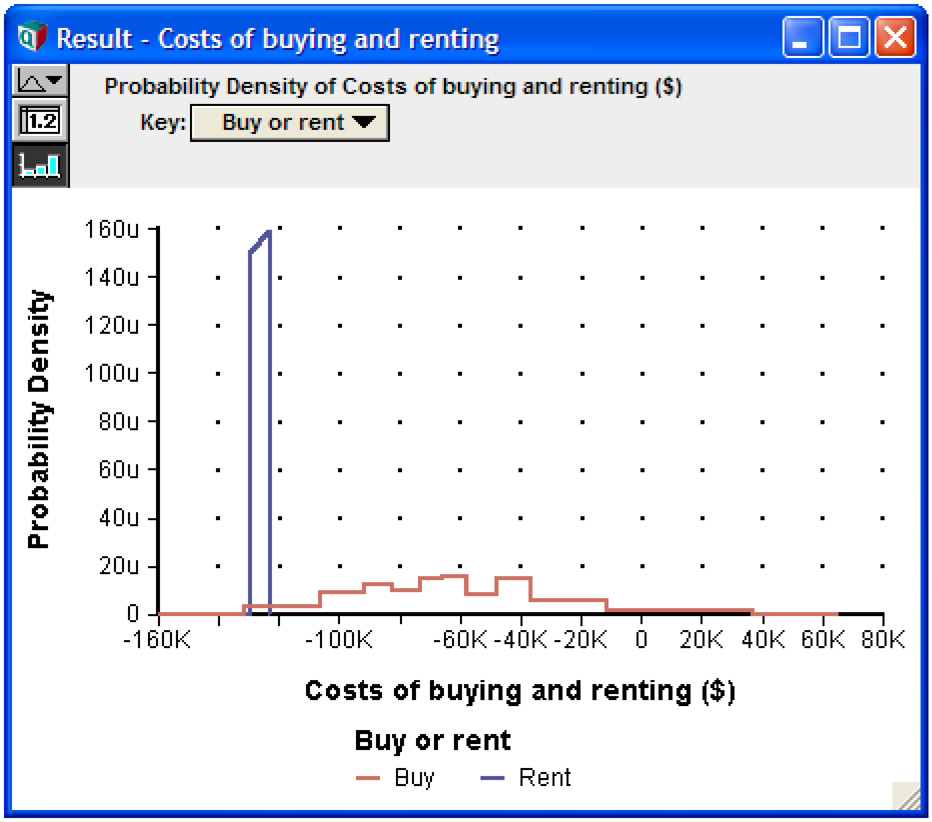
3. Change the minimum to ***3*** and the maximum to ***4***.

4. Click **OK** to accept the change.



5. Click the **Calc** button to compute the new comparison of the cost of buying to the cost of renting.





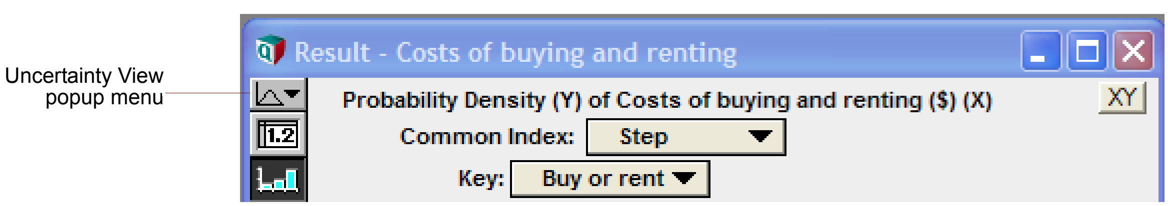
The graphs show that the uncertainty in the cost of renting has narrowed to between about $123,000 and $130,000, while the uncertainty in the cost of buying has flattened to between about $135,000 and a gain of $37,000.

Displaying alternative uncertain views

Analytica offers a variety of views to display uncertain values, including selected statistics, ***probability bands***, the ***probability density*** function, the ***cumulative probability*** distribution function, measures of central tendency, and the table of random numbers from which the uncertain distribution is estimated.

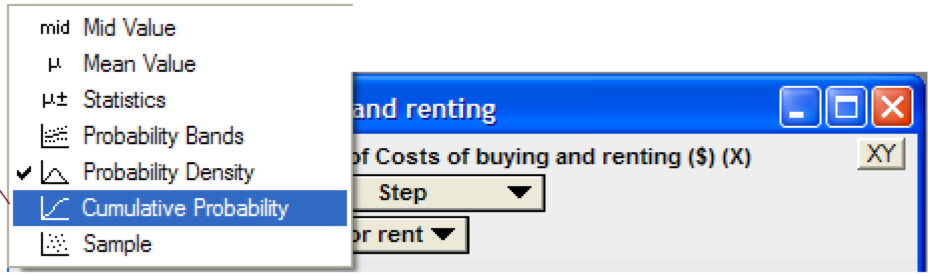
You will now examine several of these views.

In the upper-left corner of the **Result** window is the Uncertainty View popup menu.

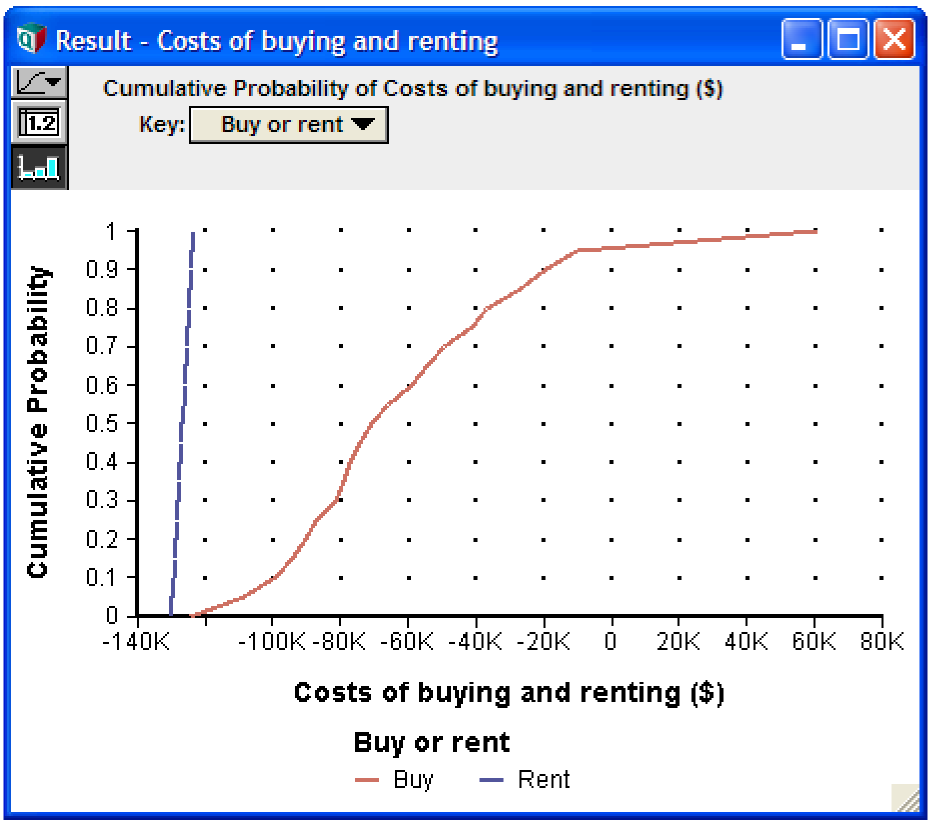


The miniature probability distribution [icon] indicates that ***Probability Density*** is selected.

1. Press on the Uncertainty View popup menu and select **Cumulative Probability** [icon].



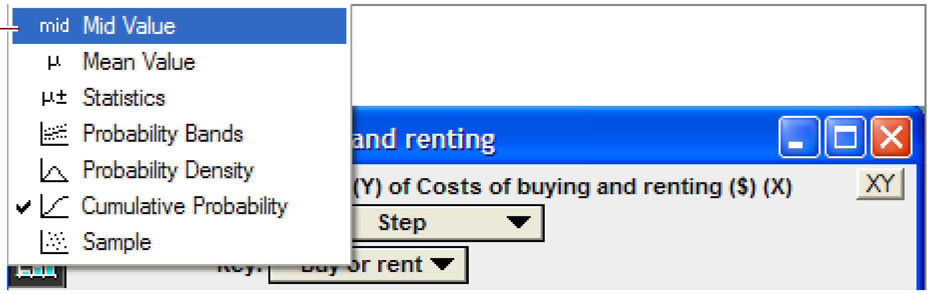
The **Result** window now shows two cumulative probability curves. Along the vertical axis, these curves give the probability that each cost is less than a given value along the horizontal axis.



There appears to be about a 50% probability that the cost to buy is below $70,000, while the cost to rent has a 50% probability of being below about $130,000.

Sometimes you might want to see an uncertain value expressed as a single number — a measure of central tendency. Analytica computes the ***mid value*** (sometimes called the ***deterministic value***) by fixing all input probability distributions at their ***median*** (50% probability) values. The mid value is the only uncertainty view available for nonprobabilistic results.

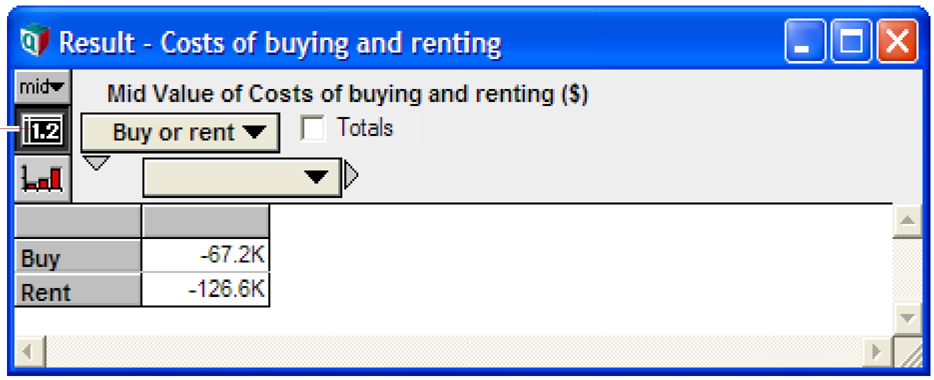
2. Select **Mid Value** mid▾ from the Uncertainty View popup menu.



The **Result** window now displays bar graphs for the two mid values.

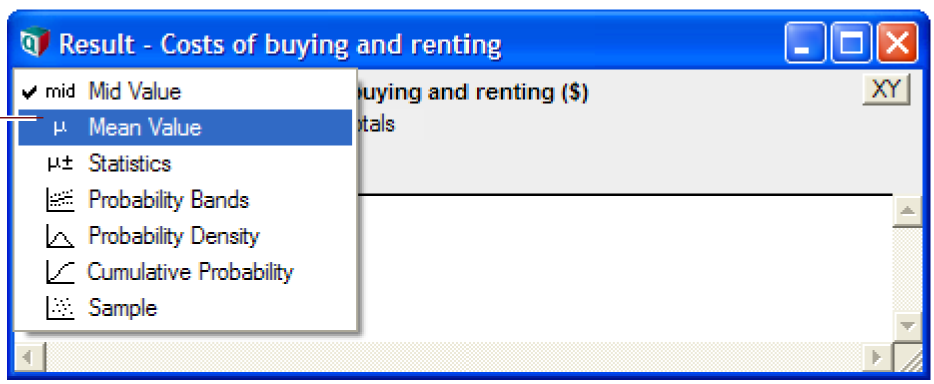
Under the Uncertainty View popup menu are two buttons, [1.2] and [graph]. The [graph] is highlighted, indicating that the **Result** window is displaying a graph view. The **Result** window can also display numeric values in a spreadsheet-like table view.

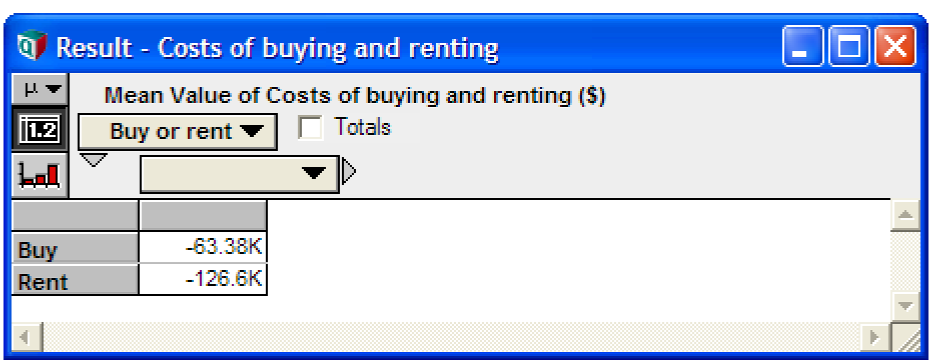
3. Click the **table** view button [1.2] to select the table view.



Analytica also provides the ***mean*** (or average) value.

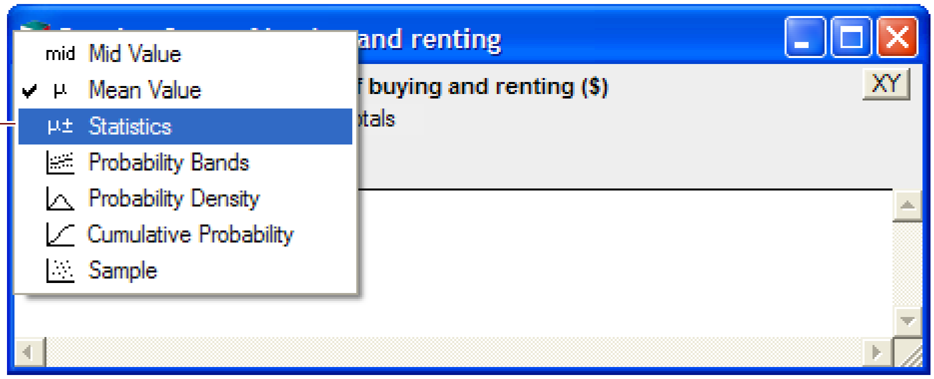
4. Select **Mean Value** μ▾ from the Uncertainty View popup menu.



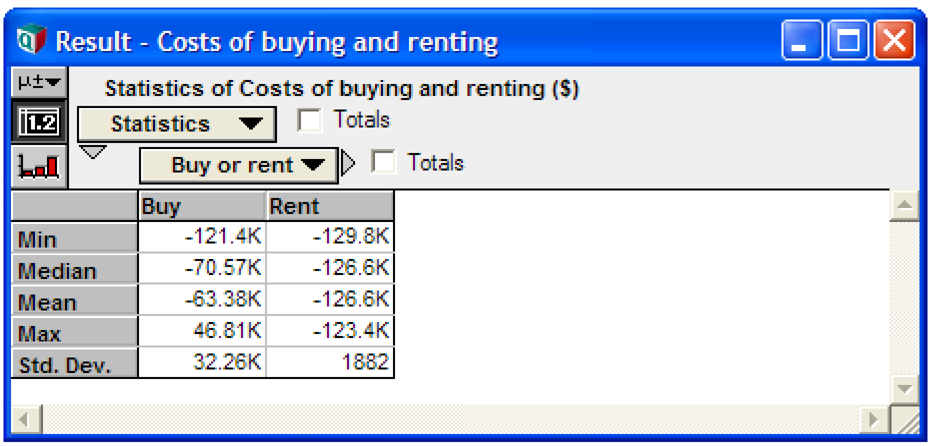


You can also view a set of statistics, including both the median and mean, the ranges (minimum and maximum), and the standard deviation.

5. Select **Statistics** [μ± ▾] from the Uncertainty View popup menu.



The **Result** window now displays the minimum, median[1], mean, maximum, and standard deviation for *Costs of buying and renting*.



The statistics might not be exact, because they are estimated from a sample of values from the distribution.

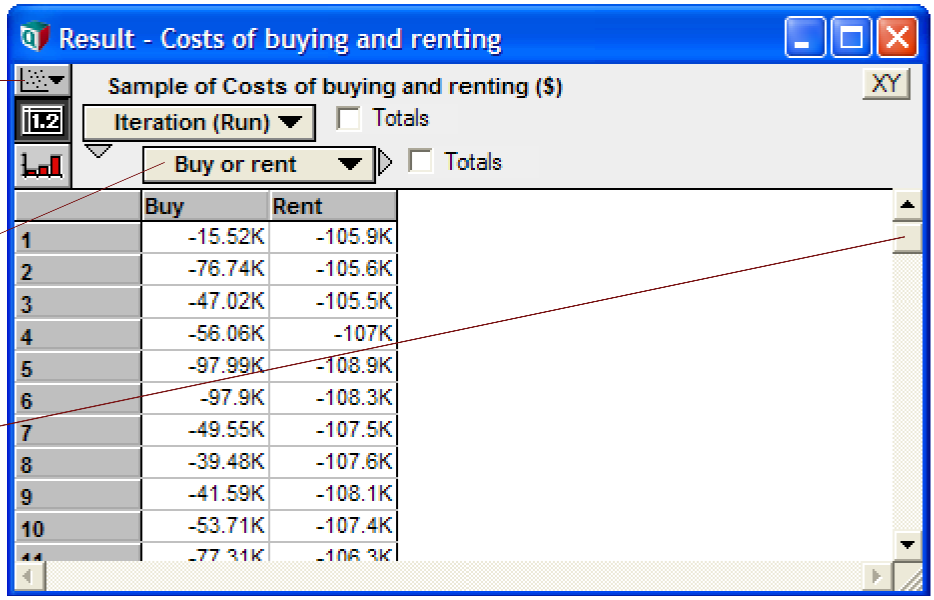
1. Note that the median value is slightly different from the mid value. The mid value is composed of non-probabilistic results generated by using the mean value for each input. The median value is calculated using probabilistic inputs and taking the median of the resulting distribution.

Finally, you see the sample values.

6. Select **Sample** from the Uncertainty View popup menu.

7. Click this pull-down menu to select “Buy or rent” for the columns.

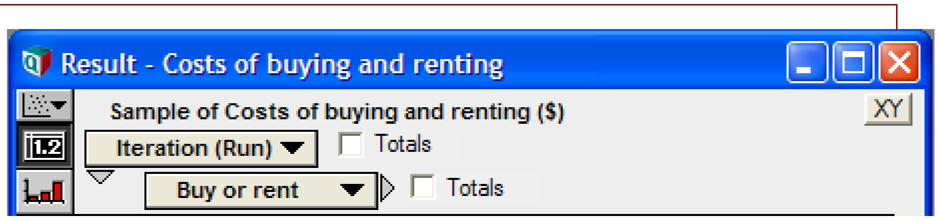
8. Use the scroll bar to examine more sample values in the table.



The table above lists the 100 sample values that Analytica randomly generated from the probability distribution to estimate the statistics.

A sample size of 100 is adequate for most applications; however, if you need more precise estimates, you can increase the sample size. See “Uncertainty Setup dialog box” in Chapter 13 of the *Analytica User Guide*.

9. Click the **Result** window’s close button to return to the **Diagram** window.



Using the Rent vs. Buy model: summary

You have now used the *Rent vs. Buy* model to calculate the results of a model, change input values and probability distributions, and display the uncertain results in a variety of ways. These are the basic techniques for using any quantitative model.

After you create your own models, you might want to give them a top-level input and output diagram like the one used in this chapter. For information about customizing a model for end users, see the *Analytica User Guide*, Chapter 9.

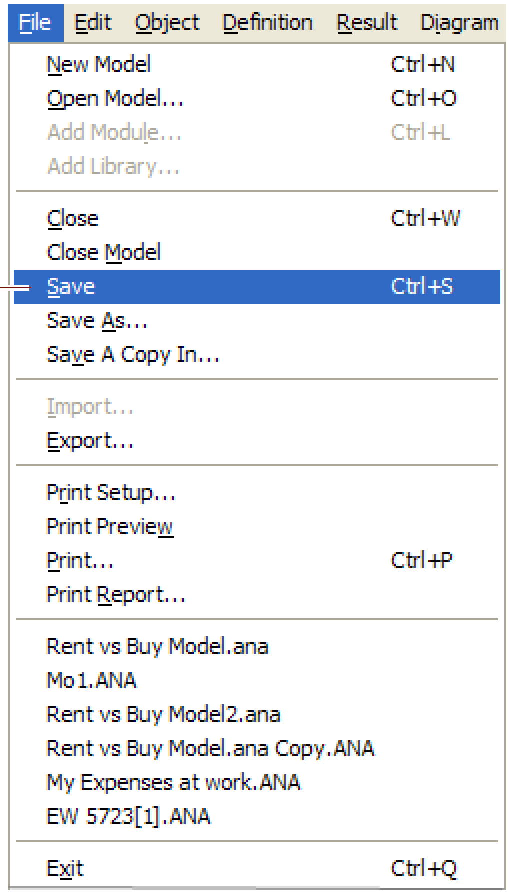
In the next chapter, you will navigate the details of the *Rent vs. Buy* model, exploring its structure and contents.

Saving your model

If you want to save changes to your model, you can do so at this point. (For instructions on quitting without saving, see the next section.)

1. Select **Save** from the **File** menu.

   You can also type the keyboard shortcut, *Control+S*.

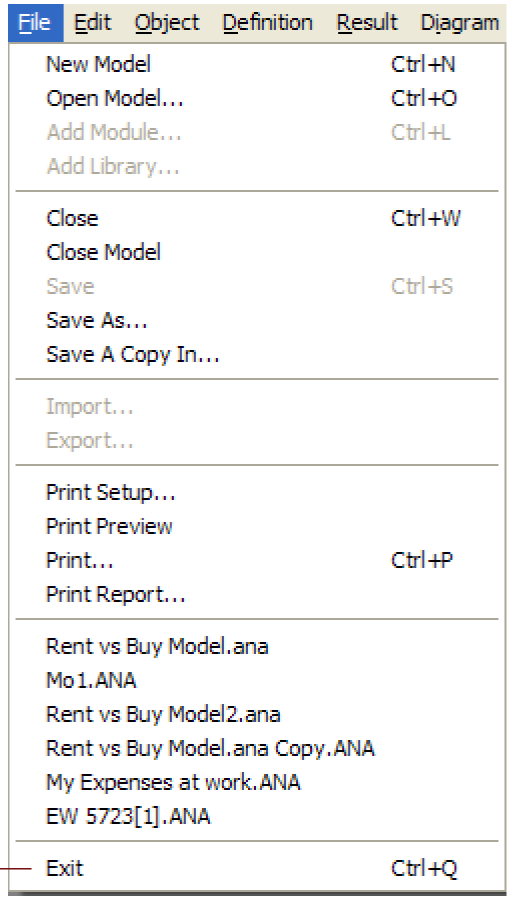


If you wish to save your model as a different file, so that you do not change the original model, select **Save As** from the **File** menu.

Quitting Analytica

When you have finished using a model, you might want to quit Analytica.

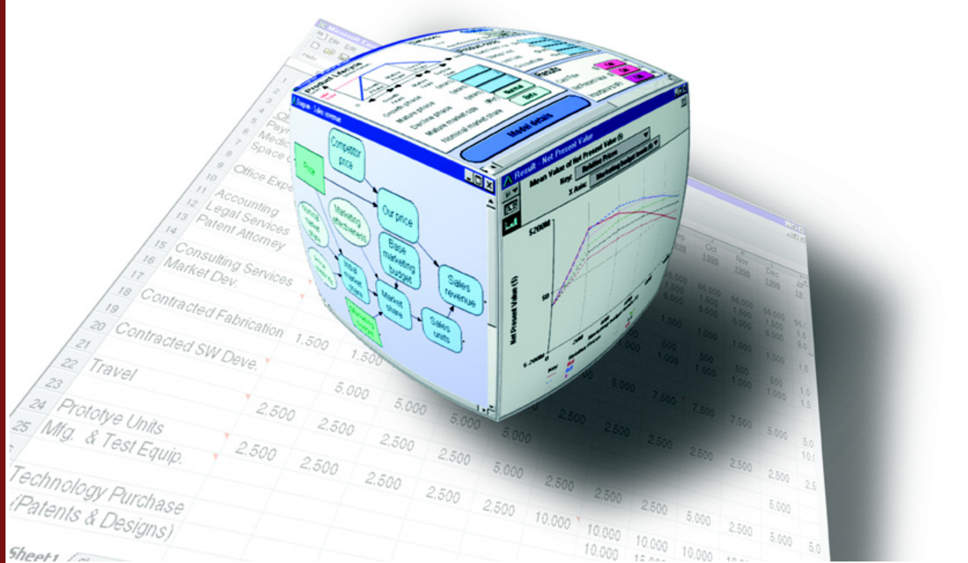
1. Select **Exit** from the **File** menu.



Chapter 2 *Exploring the Rent vs. Buy Model*

This chapter shows you how to explore a model by examining its:

- Influence diagrams
- Variables
- Attributes
- Definitions
- Results



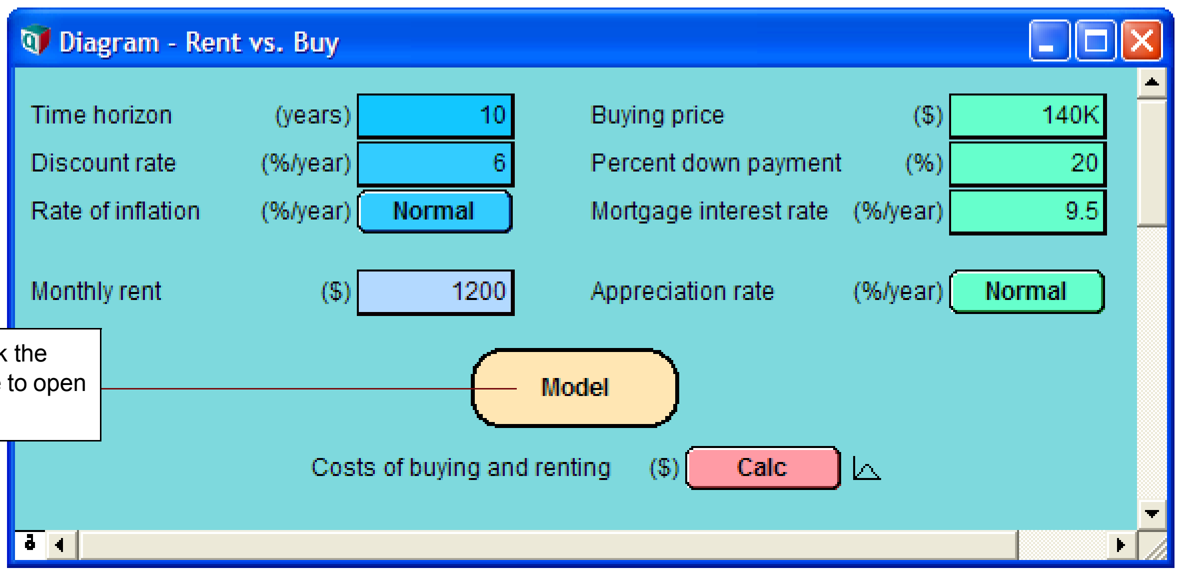
This chapter assumes you have started Analytica and have opened the *Rent vs. Buy* model. If this is not the case, see “Opening the Rent vs. Buy model” on page 8. If you are using the model as modified from Chapter 1, change the value of *Time horizon* back to **10**, the value of *Monthly rent* back to **1200**, and the value of *Buying price* back to **140K**. Also change the *Rate of inflation* back to a normal distribution with a mean of **3.5** and a standard deviation of **1.3**.

In this chapter, you will examine the structure and contents of the *Rent vs. Buy* model.

The *Rent vs. Buy* model uses financial flow conventions: funds flowing in (received) have positive values; funds flowing out (expended) have negative values.

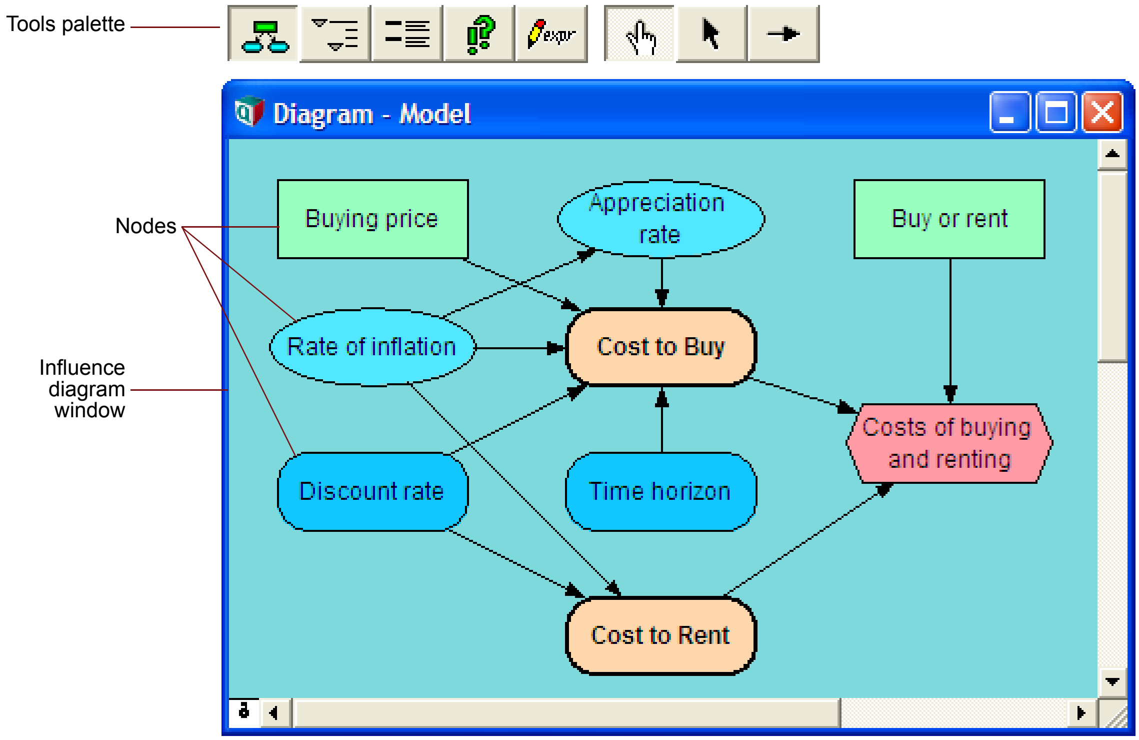
Recognizing influence diagrams

In this chapter, you will delve into some of the details of the *Rent vs. Buy* model. You will not use the top diagram that you used in Chapter 1.



1. Double-click the *Model* node to open it.

The details of an Analytica model display in an ***influence diagram*** window. An influence diagram (shown on the next page) is a graphical representation of a model, showing how different variables in the model interact with each other. A typical influence diagram consists of a number of ***nodes*** connected by ***arrows***.



Nodes represent variables and appear as boxes, ovals, hexagons, and other shapes. Different node shapes represent different types of variables. Analytica uses the term ***variable*** broadly to include anything that has a value or can be evaluated. Note that many of the variables have the same names as the inputs and output at the top diagram that you used in Chapter 1. The top diagram provides an easy way to see and change these nodes' values.

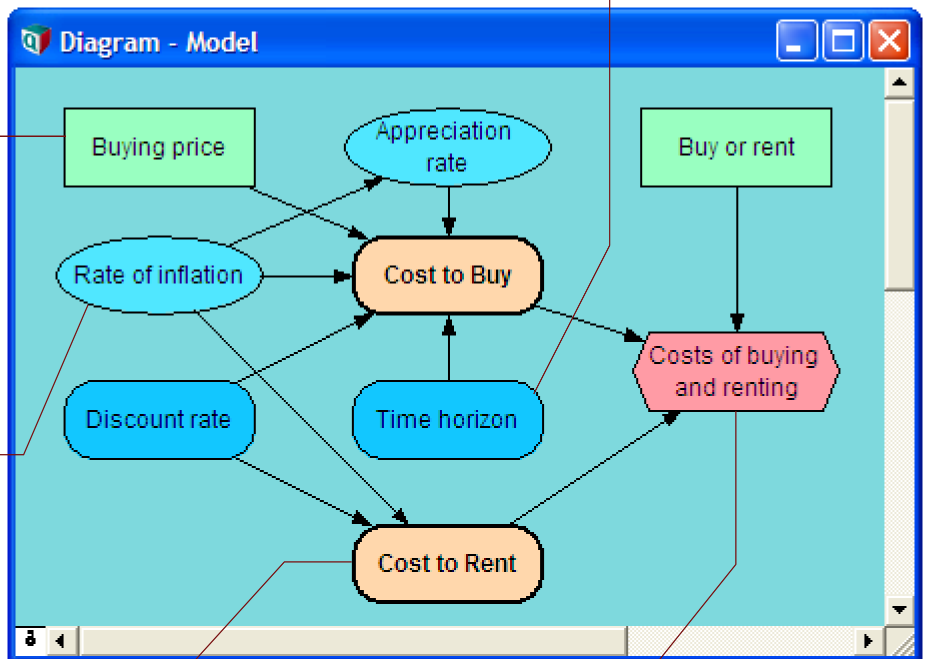
Arrows connecting different variables indicate a relation between the variables. The arrow connecting *Rate of inflation* to *Appreciation rate* indicates that the value of the *Appreciation rate* variable depends on the value of the *Rate of inflation* variable. In the *Rent vs. Buy* model influence diagram, *Cost to Buy* depends on the *Buying price*, *Rate of inflation*, *Appreciation rate*, *Discount rate*, and *Time horizon* variables.

The following figure illustrates different types of nodes.

A ***general variable*** is represented by a rounded rectangle. It can represent any type of variable and is useful when you don't know what the type is. Typically, a general variable is used to represent a deterministic quantity or functional relationship.

A ***decision variable*** is represented by a rectangular node. A decision variable is directly under the control of the decision maker.

A ***chance variable*** is represented by an oval node. A chance variable cannot be controlled directly by the decision maker. It has an uncertain value represented by a probability distribution.



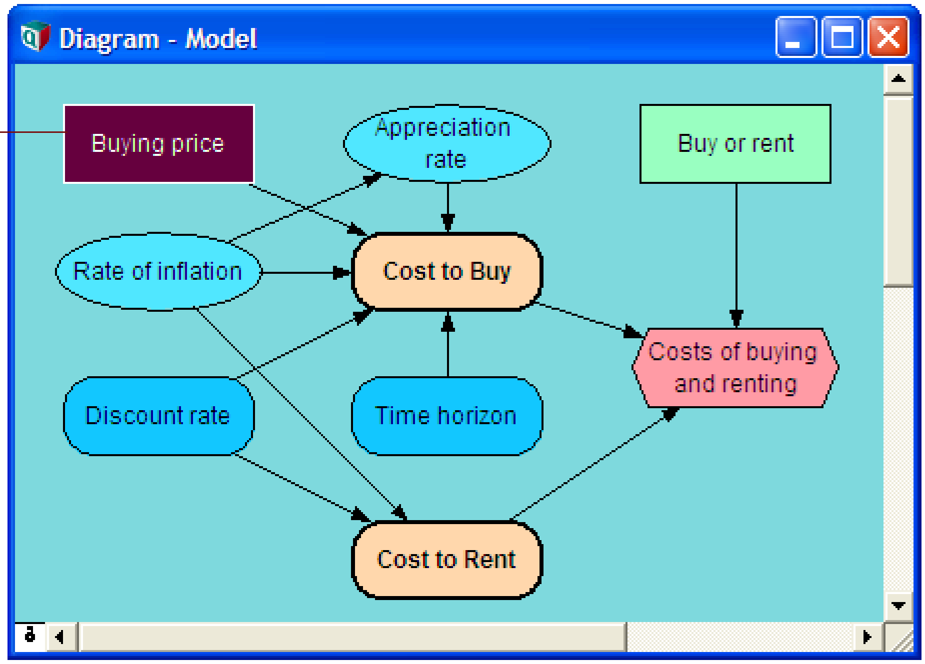
A ***module*** is represented by a thick-lined rounded rectangle. A module contains its own influence diagram, allowing nesting of multiple modules within a model.

An ***objective variable*** is represented by a hexagon. This variable is the model's "goal" and evaluates the overall value or desirability of possible outcomes. In this model, the goal is to evaluate the cost difference between renting and buying. A decision model usually contains a single objective variable.

Opening Object windows

Every object in Analytica has an associated ***Object window*** containing detailed information about it. You can display the **Object** window of any variable by double-clicking its node in the influence diagram.

1. Double-click the *Buying price* node to open the *Buying price* **Object** window.



Information about a variable is provided in a list of attributes. Attributes include the variable's class (for example, decision, chance, or constant), identifier, units, title, description, definition, inputs, and outputs. See the illustration on the next page.

This variable's ***identifier*** is *Price*. The identifier is used to refer to this variable in definitions of other variables. It can contain up to 20 characters and cannot contain spaces.

The ***Units*** attribute indicates the units of measurement for the variable. The *Buying Price* variable is measured in dollars.

This variable's ***class*** is decision variable.

The ***Title*** describes briefly what the variable represents. The title is also shown in the node on the influence diagram.

The ***Description*** provides more complete documentation (unlimited length) about this variable.

The ***Definition*** specifies the variable value, or how to compute the value, sometimes using other variables as inputs. The definition might also be a probability distribution or any other mathematical expression.

Outputs are other variables that depend on this variable.
For each output, its identifier and title are shown.

Tip You can enter numbers with a suffix abbreviation, so *Buying Price* can be defined as either 140K or 140000. A quick reference for these suffixes is given on the back page of this tutorial.

Moving between Object windows

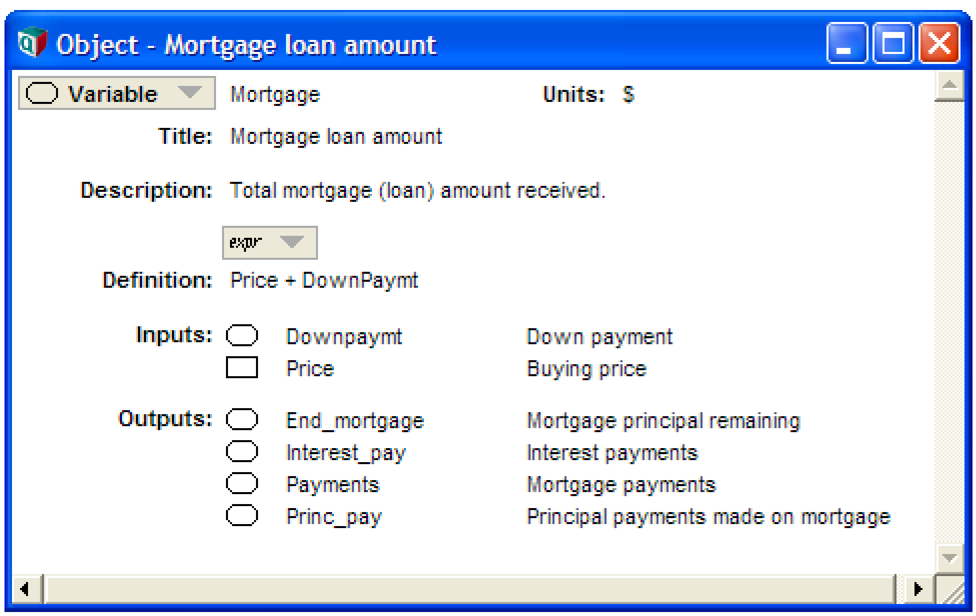
You have opened the **Object** window of a variable (*Buying price*) by double-clicking its node in the influence diagram.

The **Object** window contains a list of the variable's ***inputs*** and ***outputs***, if there are any.

You can open the **Object** window for any input or output variable by double-clicking the one you wish to view.

1. Double-click the output variable titled *Mortgage loan amount*.

Analytica switches to the **Object** window for *Mortgage loan amount*.

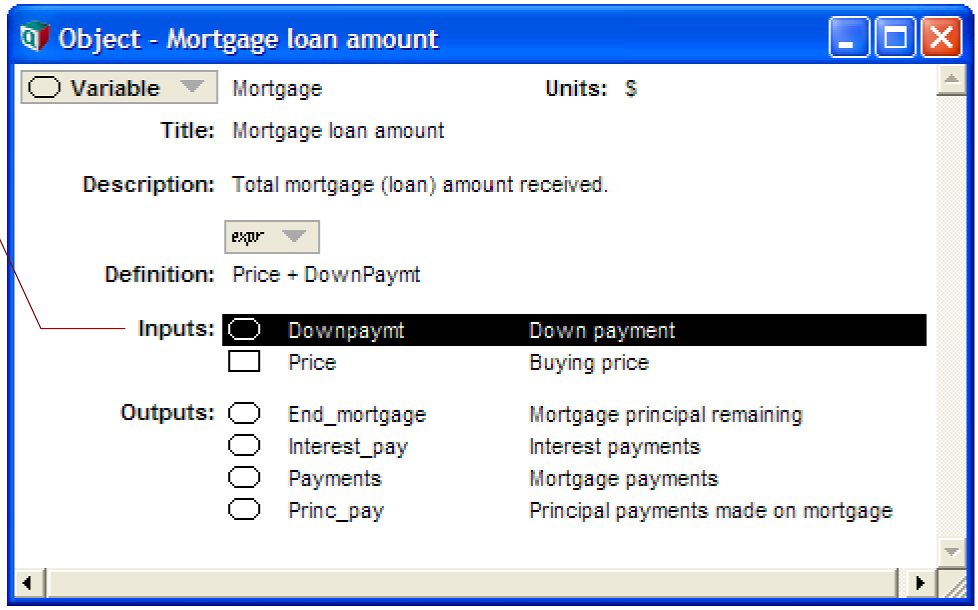


Note in the figure above that the ***Title*** of *Mortgage loan amount* is different from the variable's ***identifier***, *Mortgage*. The title is what the model user normally sees; the identifier is used as a mathematical symbol in the definitions of other variables that depend on this variable.

The ***definition*** of the *Mortgage loan amount* is an ***expression***, the sum of *Buying price* and *Down payment* (which is a negative amount). The definition refers to these variables by their identifiers.

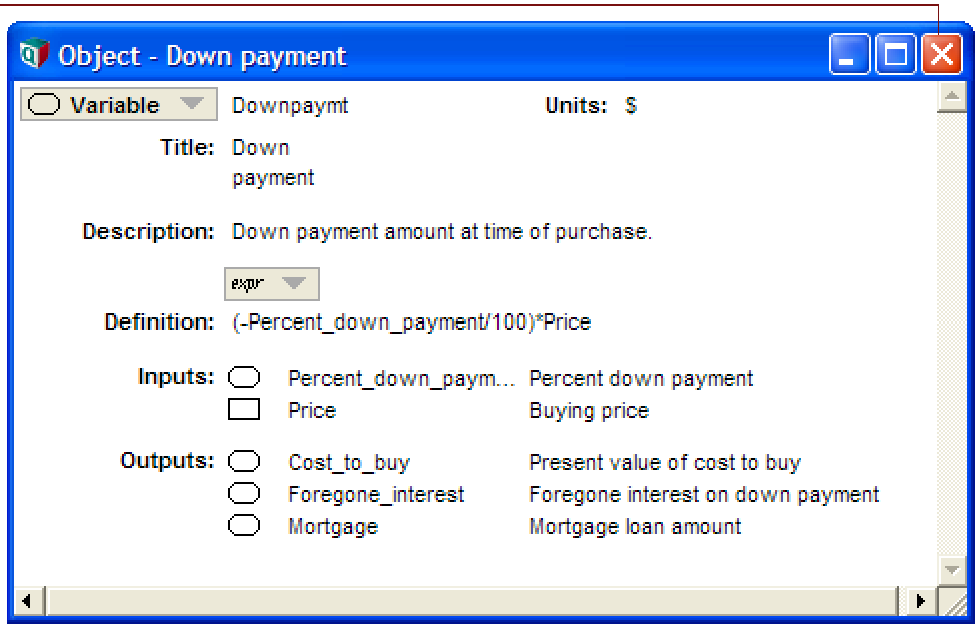
Inputs lists the identifiers and titles of the variables in the definition. *Buying price*, the variable you just examined, is one of the inputs. The other input of *Mortgage loan amount* is *Down payment*.

2. Double-click the *Down payment* input.



The **Object** window now displays the attributes of *Down payment.*

3. Click the close button to close the window and return to the diagram.



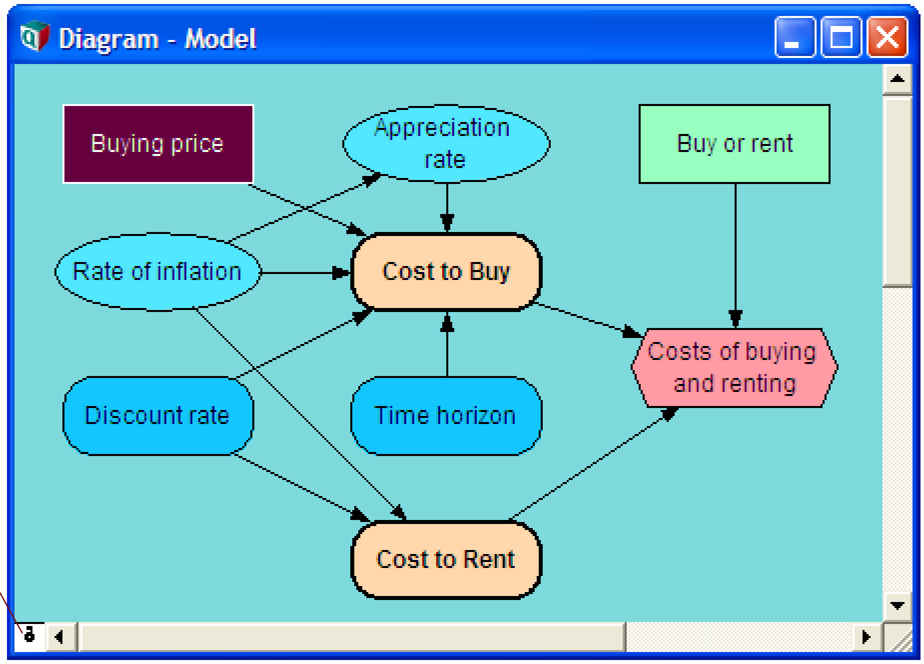
Using the Attribute panel

As an alternative to viewing a variable's attributes in a separate window, you can inspect them in the ***Attribute panel***, which is an auxiliary window pane that you can open below the influence diagram.

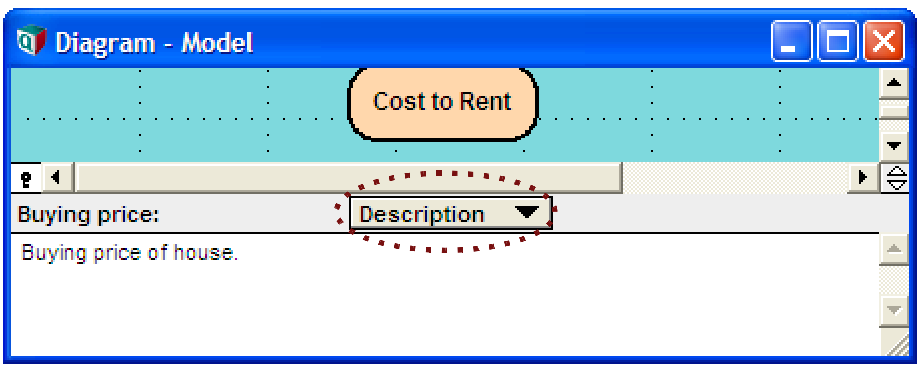
The **Attribute** panel allows you to rapidly examine one attribute at a time of any variable in the model. You select the variable you wish to view and select the attribute to examine from a popup menu.

The variable *Buying price* should be highlighted with a title in white, indicating that it is selected; if it is not, select it by clicking it once.

1. Click the key icon to open the **Attribute** panel.



By default, Analytica displays the ***description*** of the selected node (e.g., *Buying price*) in the **Attribute** panel.

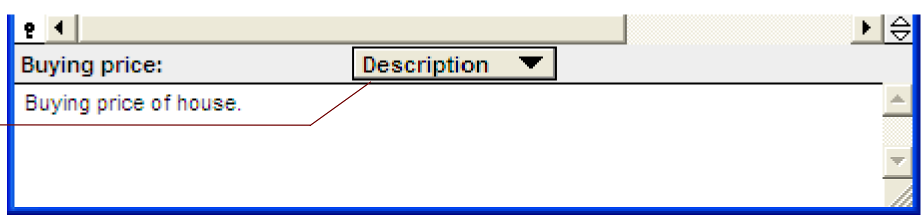


Inspecting definitions in the Attribute panel

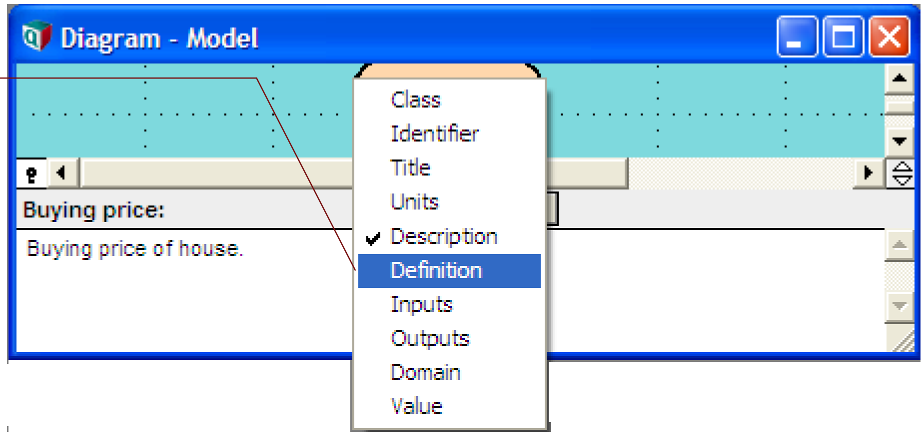
The **Attribute** panel allows you to inspect any attribute of a variable.

In this section, you will see the definition of two variables that you viewed in the top-level diagram in Chapter 1.

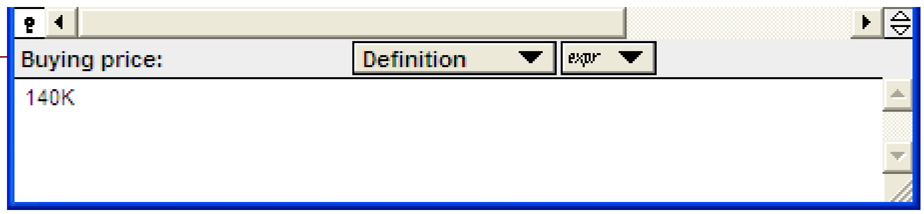
1. Click **Description** to view the Attribute popup menu.



2. Select **Definition** as the new current attribute to display.



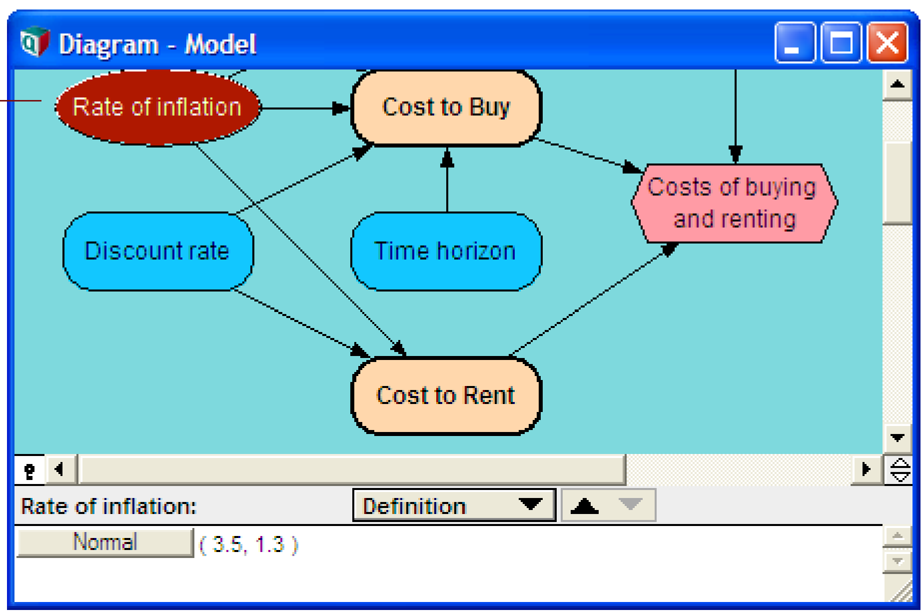
3. The definition of *Buying price* is displayed. It is a single number: 140K (140,000).



When a variable is defined as an ***uncertainty distribution***, a button appears in the **Definition** field.

4. Select the *Rate of inflation* node.

   The definition in this example is a normal distribution; hence, the button is labeled **Normal**.



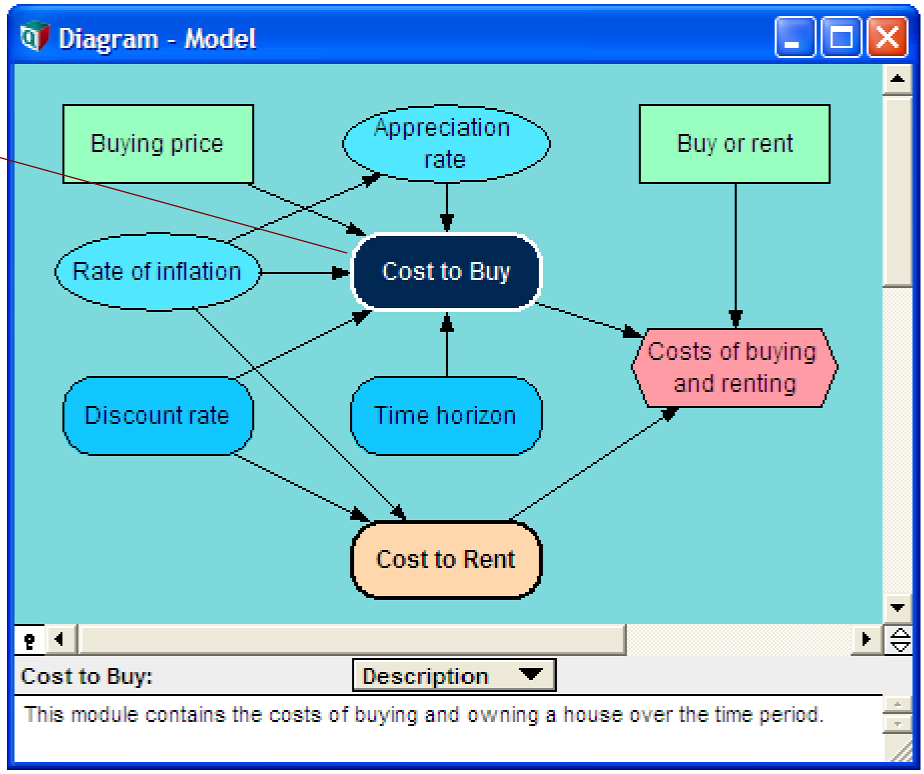
In Chapter 1 you saw that *Rate of inflation* is defined as a normal distribution with a mean of 3.5 and a standard deviation of 1.3. These are the input parameters to the normal distribution, and they display to the right of the **Normal** button.

Opening modules

Analytica models generally contain ***modules***. Each module contains the details of a part of the model, also represented as an influence diagram. In the *Rent vs. Buy* model, *Cost to Buy* and *Cost to Rent* are both modules.

Modules can also contain other modules. In this manner, a large model with hundreds of variables can be organized into a hierarchy of modules, each small enough to be easily understood.

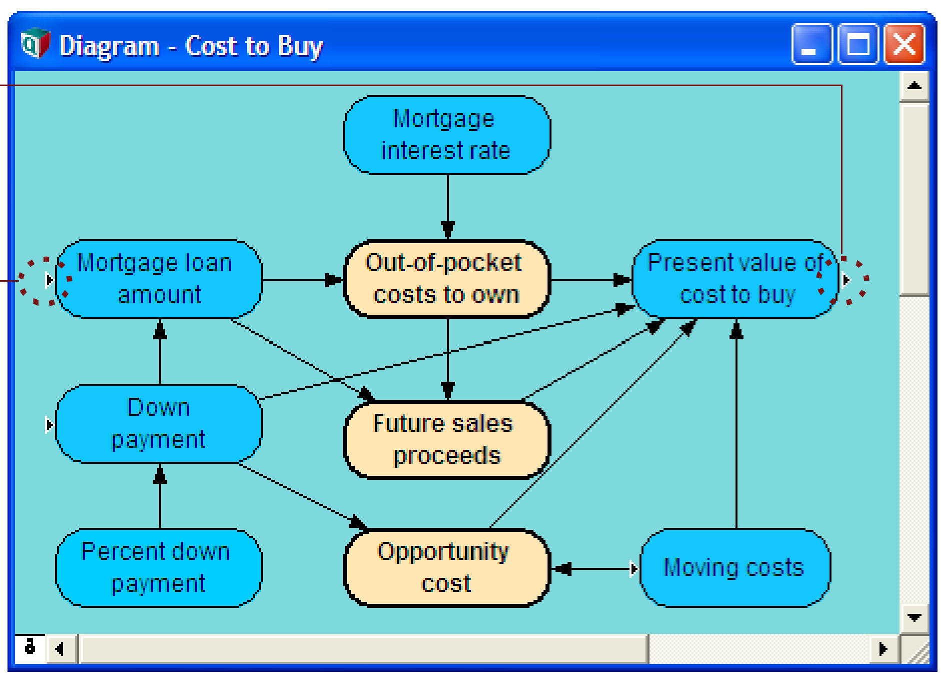
1. Double-click the *Cost to Buy* node to open the module.



Analytica displays the influence diagram of the *Cost to Buy* module. This module contains three additional modules: *Out-of-pocket costs to own*, *Future sales proceeds*, and *Opportunity cost*.

Output arrowhead

Input arrowhead



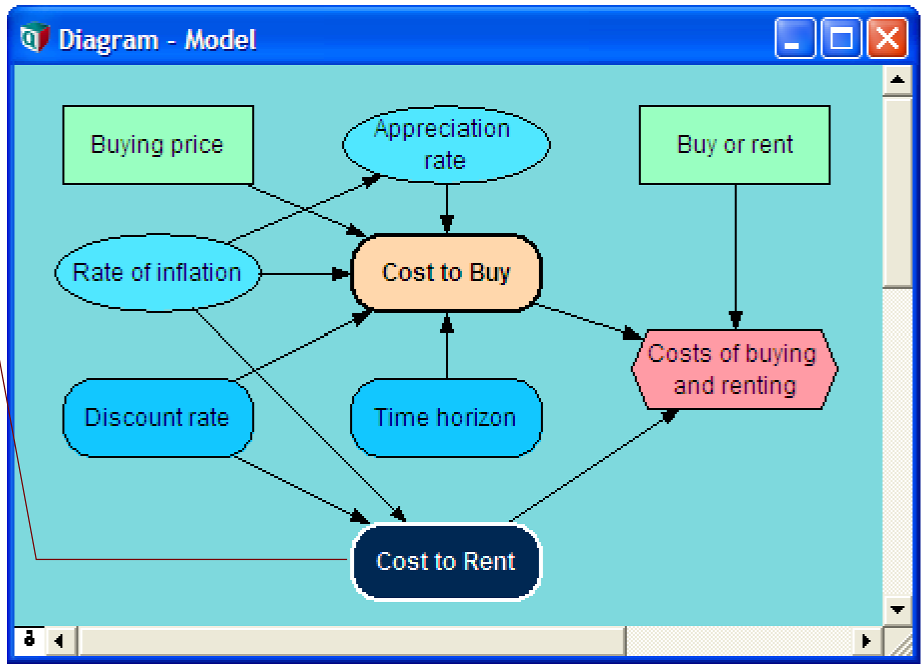
The ***input arrowhead*** (without a trailing line) shows that the node to the right of the arrow has one or more inputs from outside this module.

The ***output arrowhead*** shows that the node to the left of the arrow has one or more outputs outside this module.

2. Click the **Diagram** button  to return to the parent diagram, *Model*.

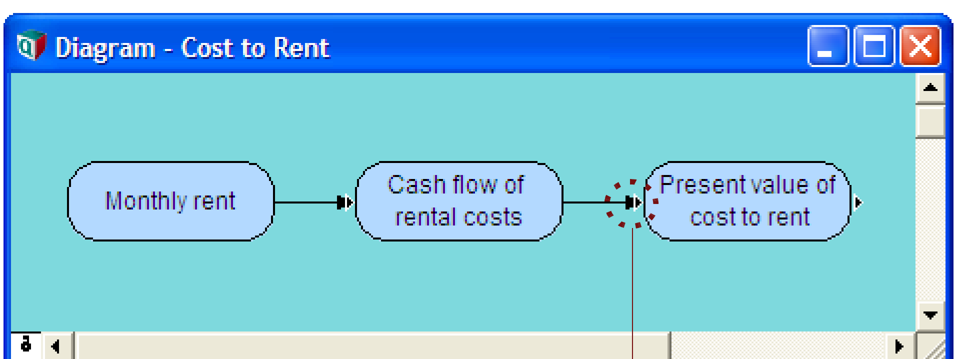
3. Double-click the *Cost to Rent* node to open the module.

The *Cost to Rent* diagram opens (see the figure below).



Analytica limits the number of open windows at each level of the model hierarchy to minimize clutter on your screen. See "Managing Windows" in Chapter 19 of the *Analytica User Guide* for information on how to open more than one module **Diagram** window at a time.

4. Click the **Diagram** button  to return to the parent diagram, *Model*.



Combined arrowhead

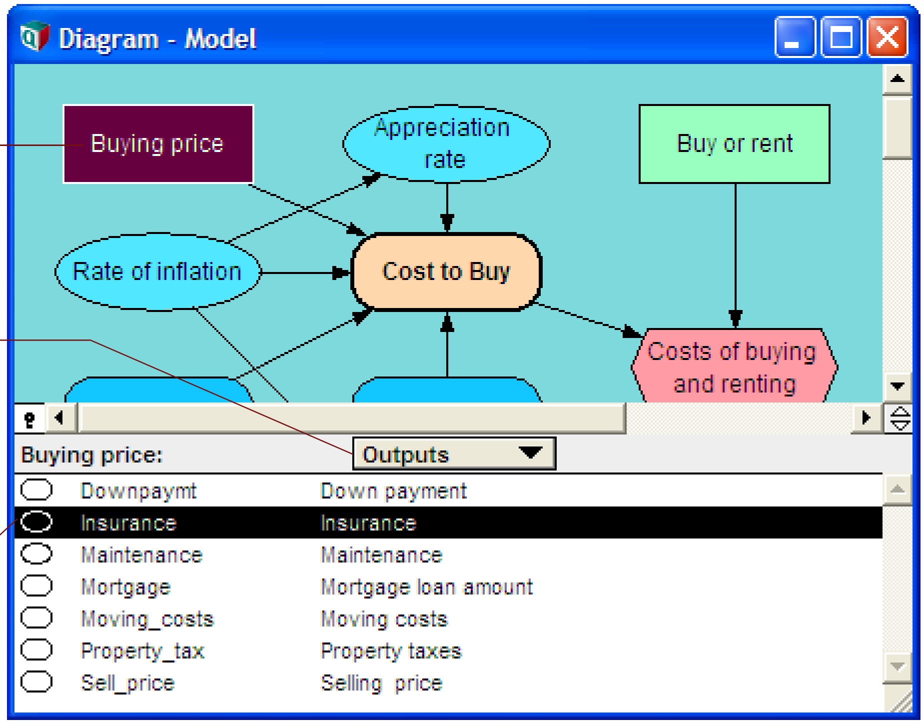
The combined arrowhead, shown above, indicates that the node has one or more inputs from outside this module, plus the input variable in this module.

You can also navigate the model by tracking a variable's inputs or outputs.

5. Select the *Buying price* node.

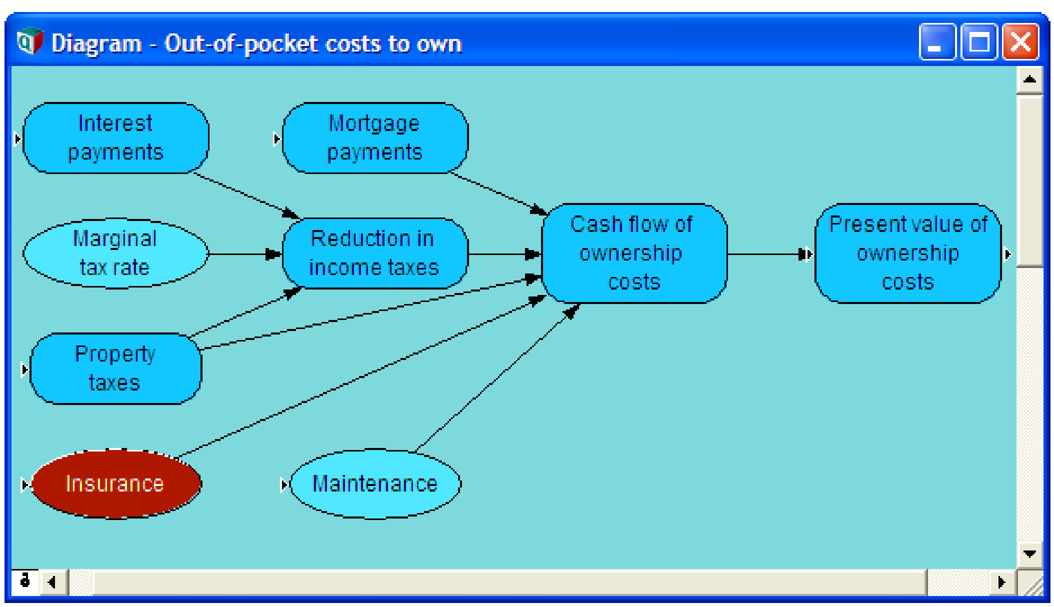
6. Select **Outputs** from the Attribute popup menu to view a list of variables that depend on the *Buying price* variable.

7. Double-click *Insurance*.



The *Out-of-pocket costs to own* module diagram is brought to the front, with the *Insurance* node selected.

8. Click the **Diagram** button to view the *Cost to buy* diagram.



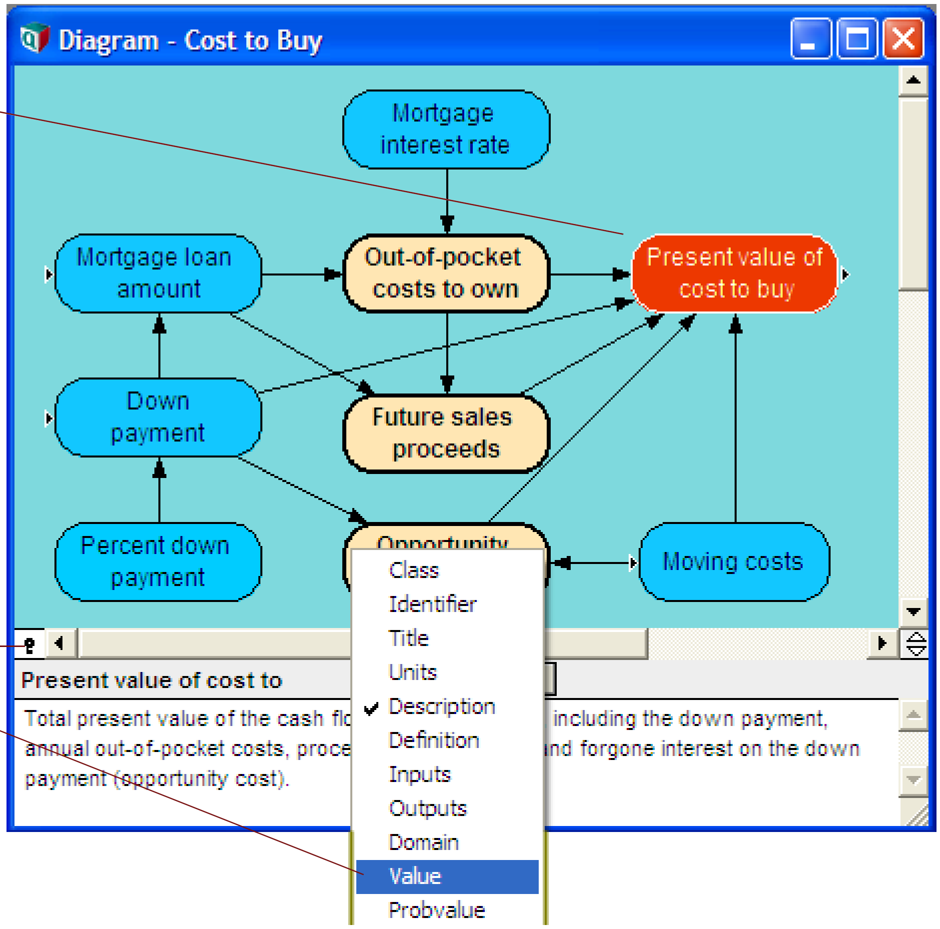
Inspecting values in the Attribute panel

The **Attribute** panel allows you to view certain attributes, such as a variable's ***value***, that are not (initially) displayed in an **Object** window.

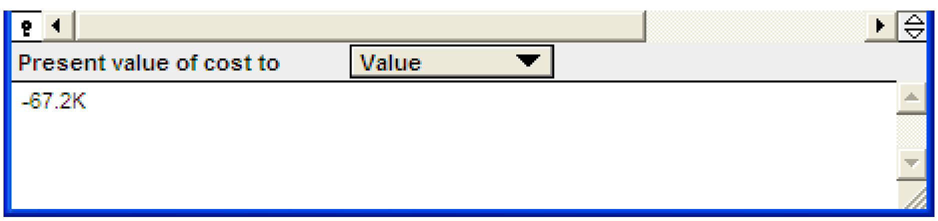
1. Select the *Present value of cost to buy* node.

2. Click the key icon to open the **Attribute** panel.

3. Press the Attribute popup menu and select **Value**.



The deterministic (or mid) value of *Present value of cost to buy* displays, in this case, -67.2K.



If ***Value*** was not previously computed, Analytica computes the variable's value ***deterministically***, assuming that all of the input probability distributions are fixed at their median values. ***Mid value*** is an abbreviation for this deterministically computed value.

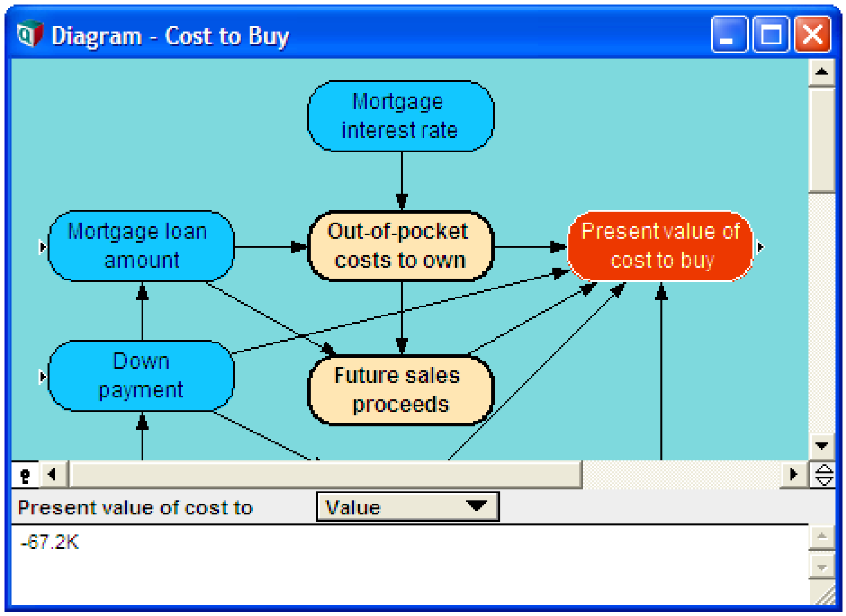
You can use the **Attribute** panel in this manner to examine the mid value of any variable in the model.

It is faster to compute a mid (deterministic) value than an uncertain (probabilistic) value, so it is useful for conducting initial checks of a model before performing any uncertainty analysis.

Displaying results

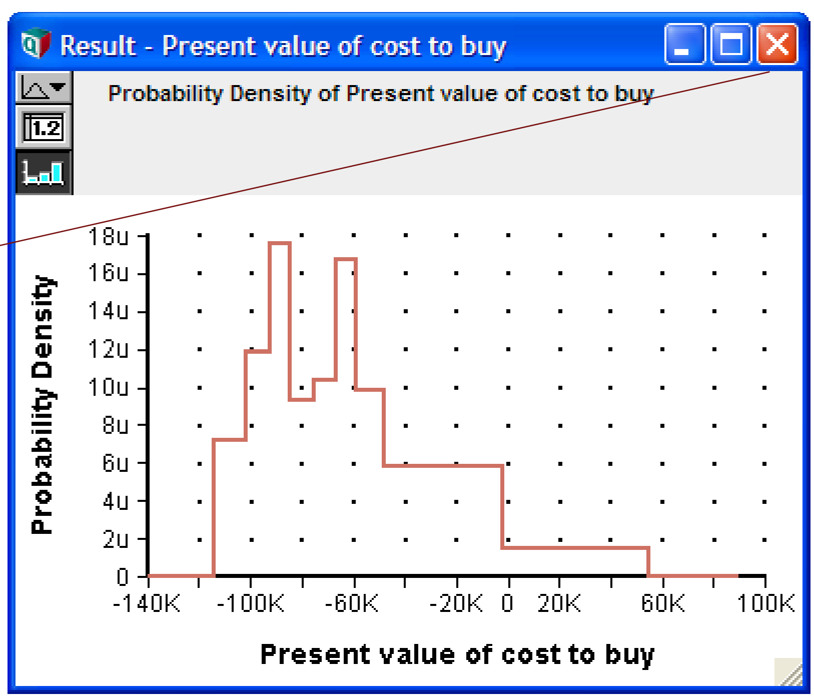
When you are viewing a model's influence diagram, you can evaluate any variable and display its value in a **Result** window.

1. With *Present value of cost to buy* still selected, click the **Result** button  to evaluate it.



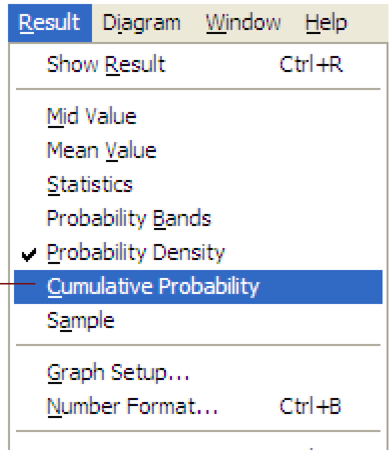
A **Result** window displays the probability density function graph for this variable. Analytica displays the uncertainty view that was most recently selected from the Uncertainty View popup menu, or that was saved with the model.

2. Click the close button to close the **Result** window.



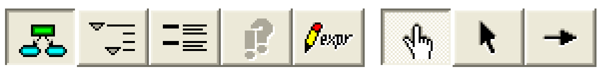
As an alternative to clicking the **Result** button  and then selecting an uncertainty view, you can use the **Result** menu to evaluate a variable and select the uncertainty view of the result.

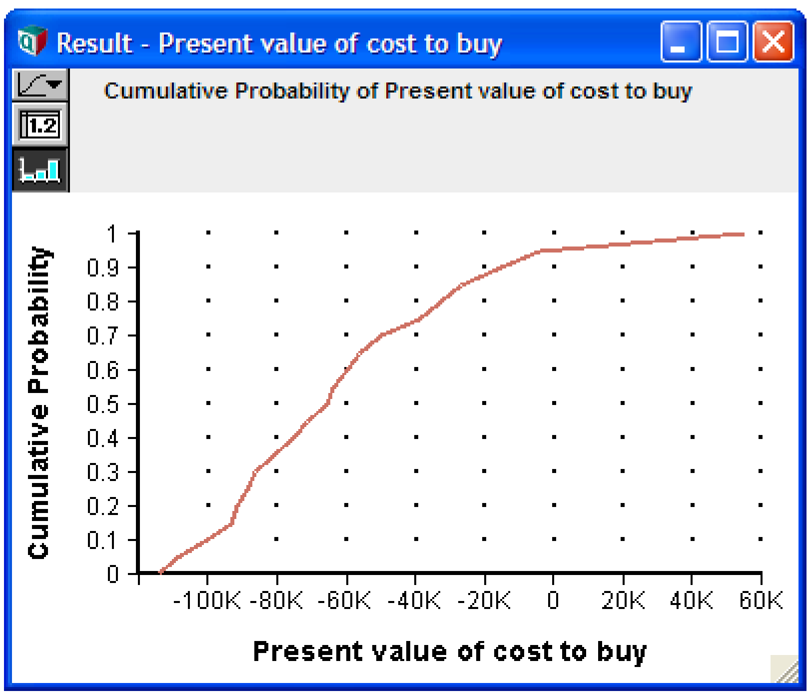
3. With *Present value of cost to buy* still selected, select the **Result** menu. The check mark next to **Probability Density** indicates that the **Probability Density** was last displayed. Select **Cumulative Probability**.



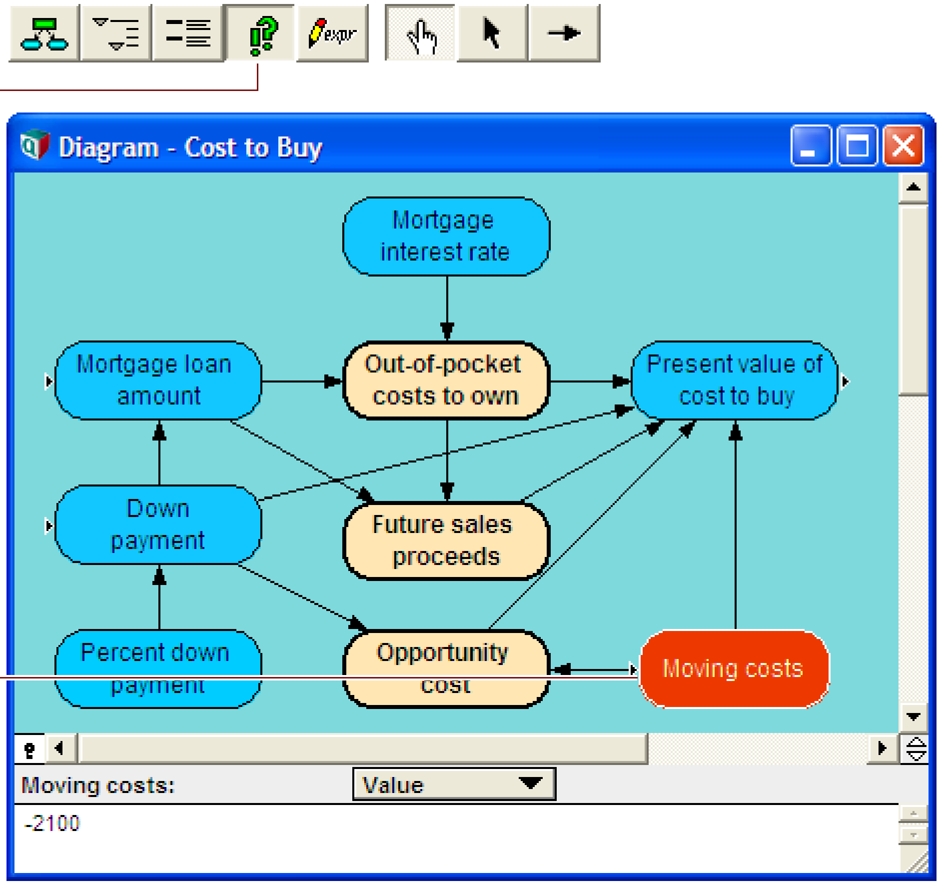
The **Result** window appears displaying the variable's cumulative probability distribution.

4. Click the **Diagram** button  to display the *Cost to buy* diagram.



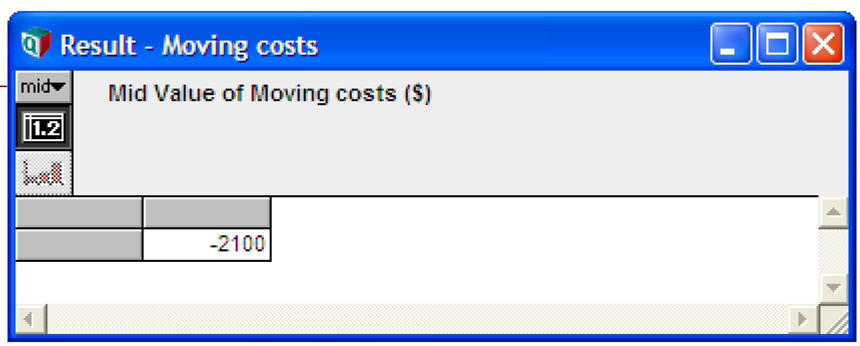


5. Select *Moving costs* and then click the **Result** button to evaluate.



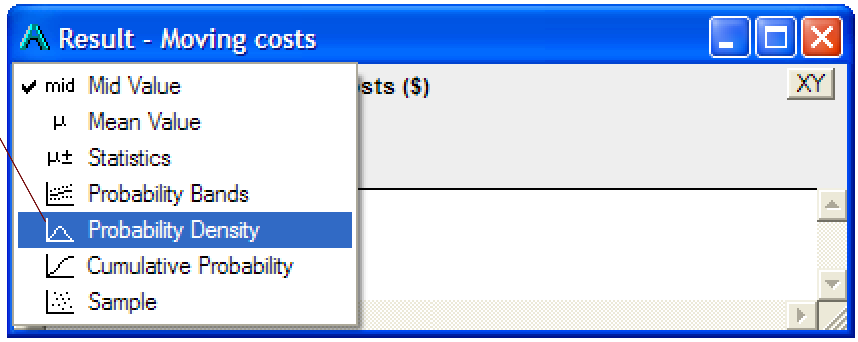
A single mid value appears in table view.

Uncertainty View Popup Menu



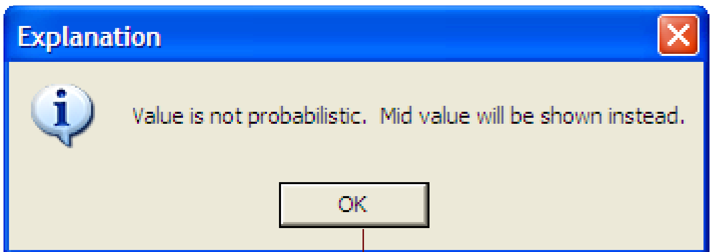
The mid value in table view is the only result view available for a nonprobabilistic variable with a single value.

6. Select **Probability Density** from the Uncertainty View popup menu.



Analytica tells you that this is a nonprobabilistic value.

7. Click the **OK** button.



Exploring the Rent vs. Buy model: summary

You now have browsed the *Rent vs. Buy* model by examining its influence diagrams, variables, attributes, definitions, and results. These are the basic techniques for exploring any Analytica model.

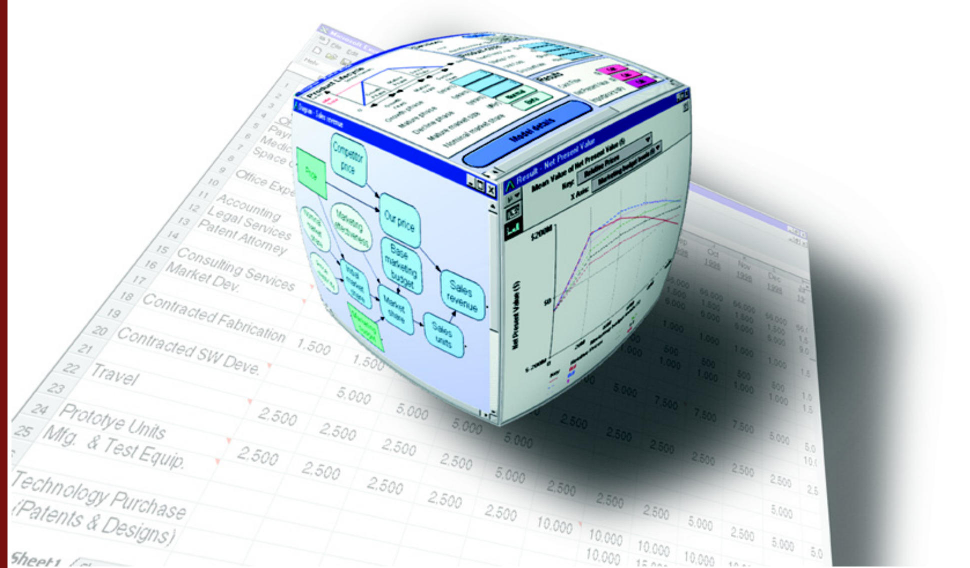
The next chapter shows you how to analyze the *Rent vs. Buy* model.

You can quit Analytica at this point. See “Quitting Analytica” on page 23.

Chapter 3 *Analyzing the Rent vs. Buy Analysis Model*

This chapter shows you how to:

- Perform importance analysis
- Perform parametric analysis
- Set up and compare alternative decisions



In this chapter you will analyze the *Rent vs. Buy Analysis* model, a modified version of the model that you used in Chapter 1, “Using the Rent vs. Buy Model” and Chapter 2, “Exploring the Rent vs. Buy Model.” You will identify its key sources of uncertainty through ***importance analysis, perform parametric analysis***, and ***compare alternative*** decisions.

For instructions on how to open a model, see “Opening the Rent vs. Buy model” on page 8. In this case, however, open the *Rent vs. Buy Analysis* model by double-clicking the file labeled **`Rent vs. Buy Analysis.ana`**.

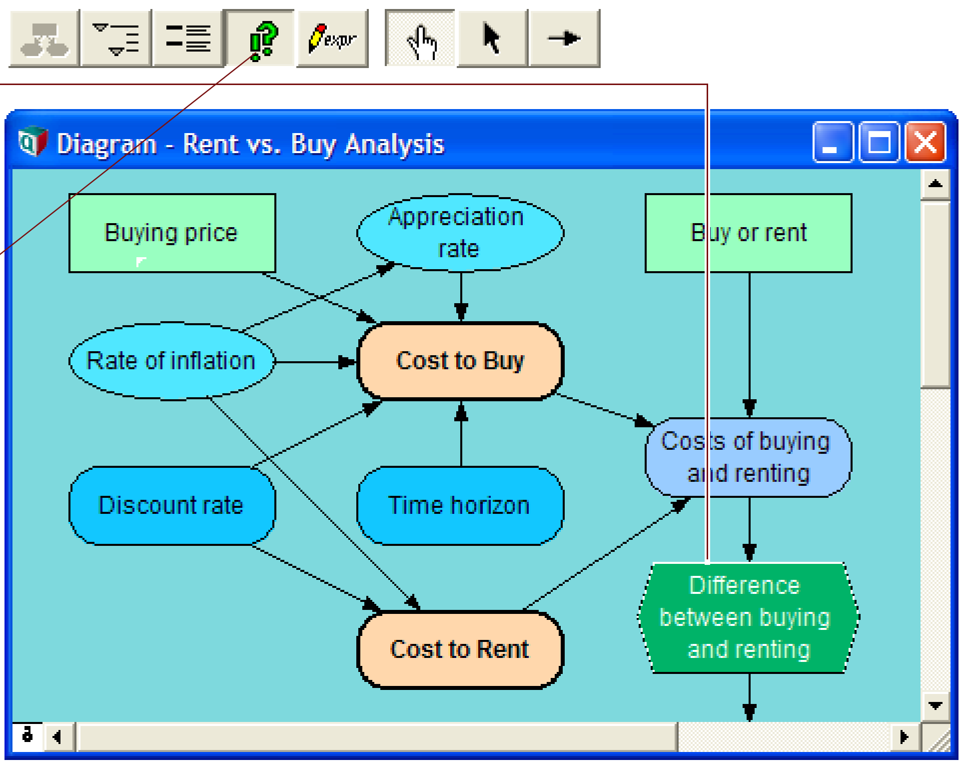
Examining the difference between renting and buying

The *Rent vs. Buy Analysis* model is the module called *Model* that you explored in Chapter 2, “Exploring the Rent vs. Buy Model,” with the addition of nodes to help you understand the importance of the uncertain inputs to the uncertainty in the output.

In Chapter 1, “Using the Rent vs. Buy Model,” you saw that evaluating *Costs of buying and renting* produces a graph of two uncertain values. To understand whether it would be financially advantageous to rent or buy, the *Rent vs. Buy Analysis* model includes the objective node, *Difference between buying and renting*.

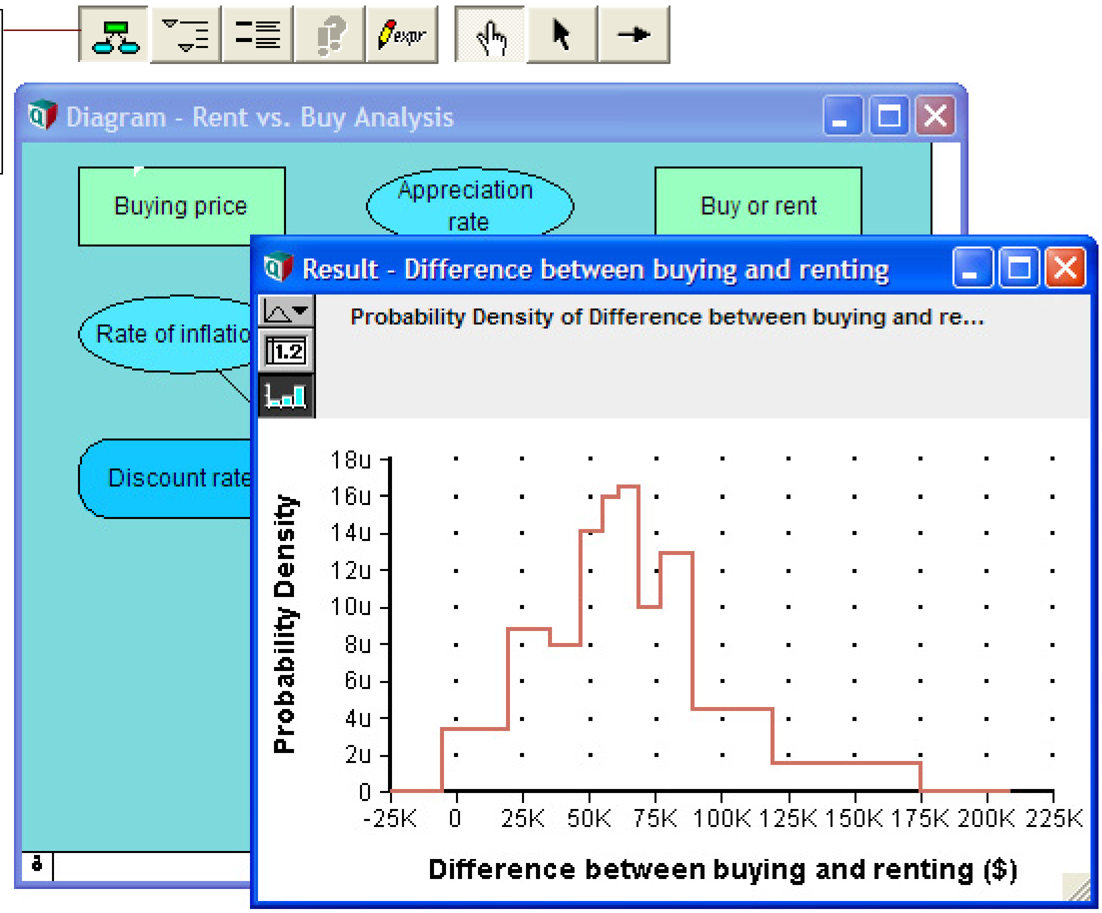
1. Click the *Difference between buying and renting* node to select it.

2. Click the **Result** button to evaluate it.



The difference between the two uncertain values is also uncertain. The difference is positive if buying costs less over the time period, and negative if renting costs less over the time period.

3. Click the **Diagram** button to return to the *Rent vs. Buy Analysis* **Diagram** window.



Importance analysis

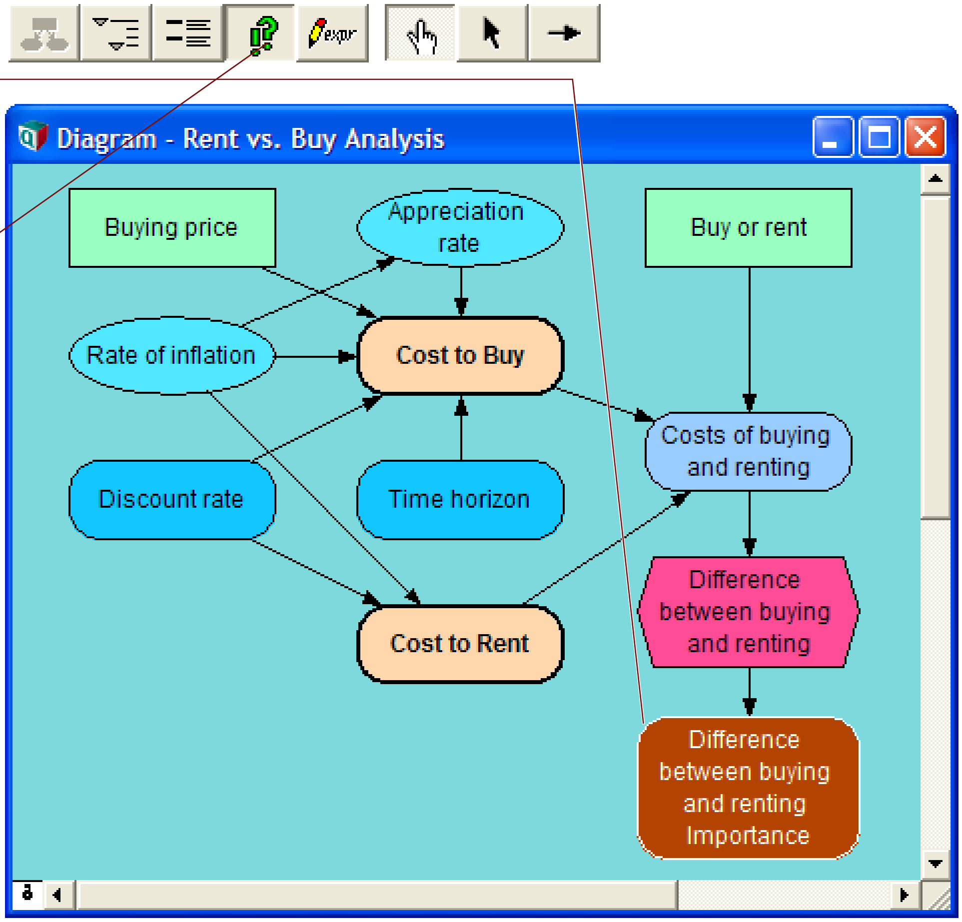
In the *Rent vs. Buy Analysis* model, as in most complex models, several of the input variables are uncertain.

It is often useful to understand how much each uncertain input contributes to the uncertainty in the output. Typically, a few key uncertain inputs are responsible for the lion's share of the uncertainty in the output, while the rest of the inputs have little impact.

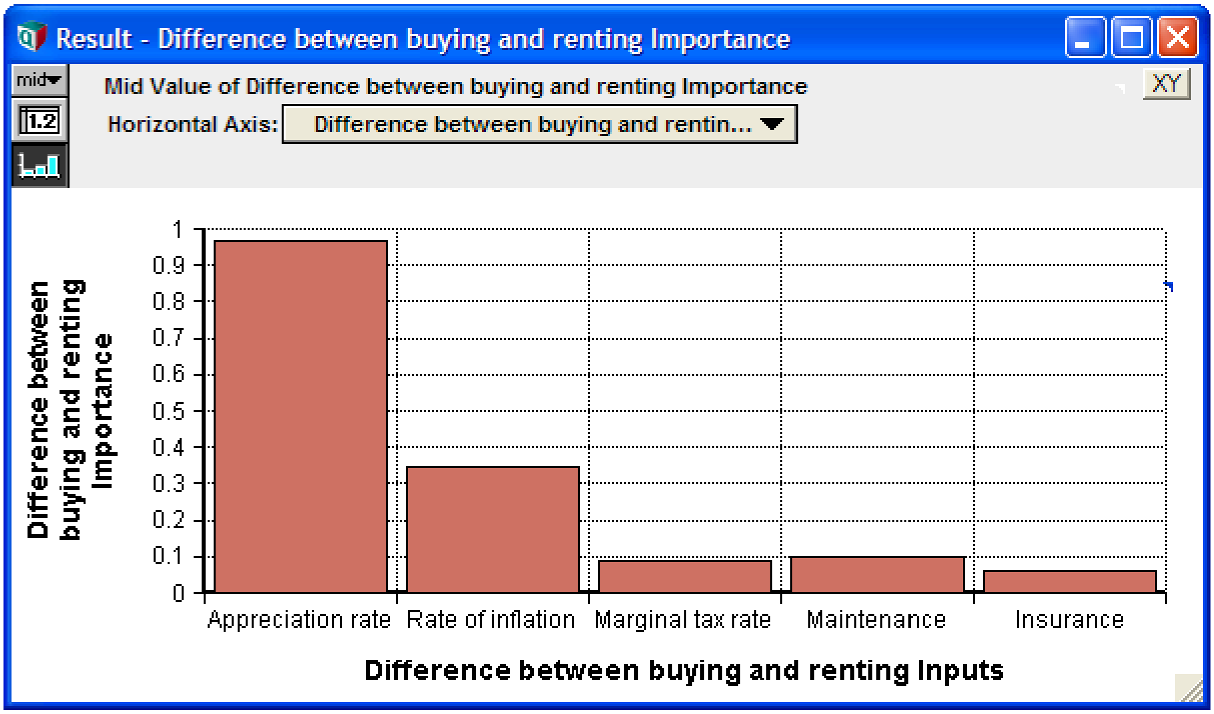
Analytica's ***importance analysis*** features can help you understand which uncertain inputs contribute most to the uncertainty in the output. You can then concentrate on getting more precise estimates or building a more detailed model for the one or two most "important" inputs.

1. Select the *Difference between buying and renting* node.

2. Click the **Result** button to display the importance values.

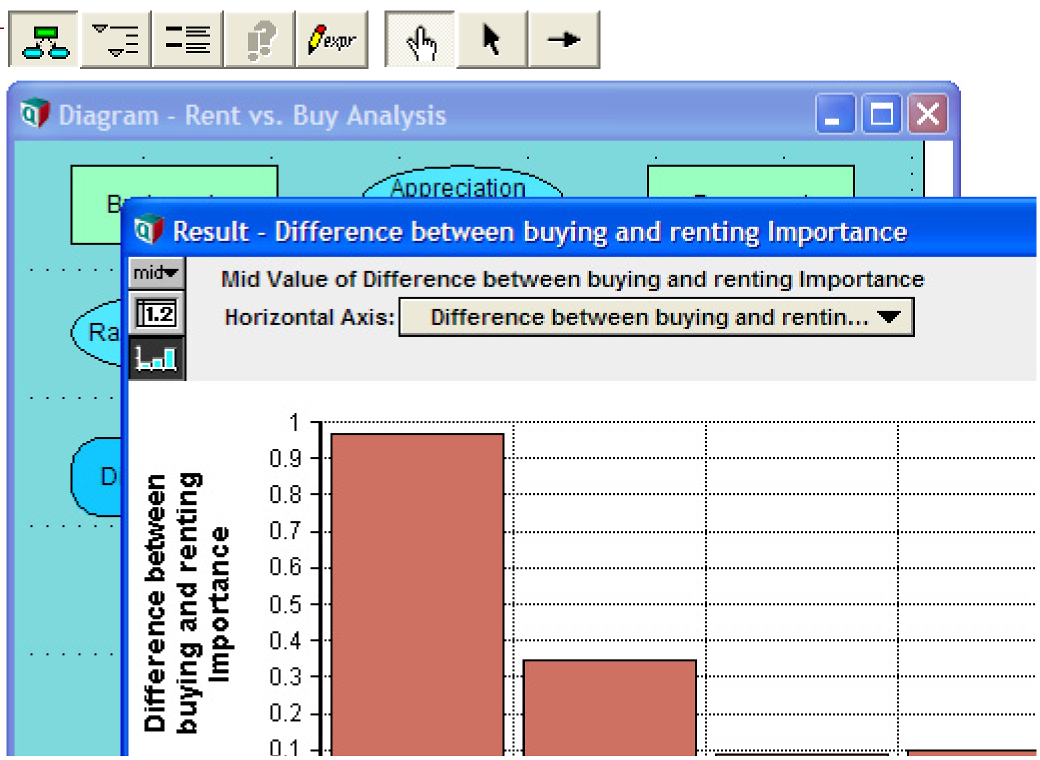


Analytica defines ***importance*** as the rank order correlation between the output value and each uncertain input. Each variable's importance is calculated on a relative scale from 0 to 1. An importance value of 0 indicates that the uncertain input variable has no effect on the uncertainty in the output. A value of 1 implies total correlation, where all of the uncertainty in the output is due to the uncertainty of a single input.



It is clear in the figure above that the input *Appreciation Rate* is contributing most of the uncertainty in the *Difference between buying and renting*.

3. Click the **Diagram** button to return to the *Rent vs. Buy Analysis* **Diagram** window.



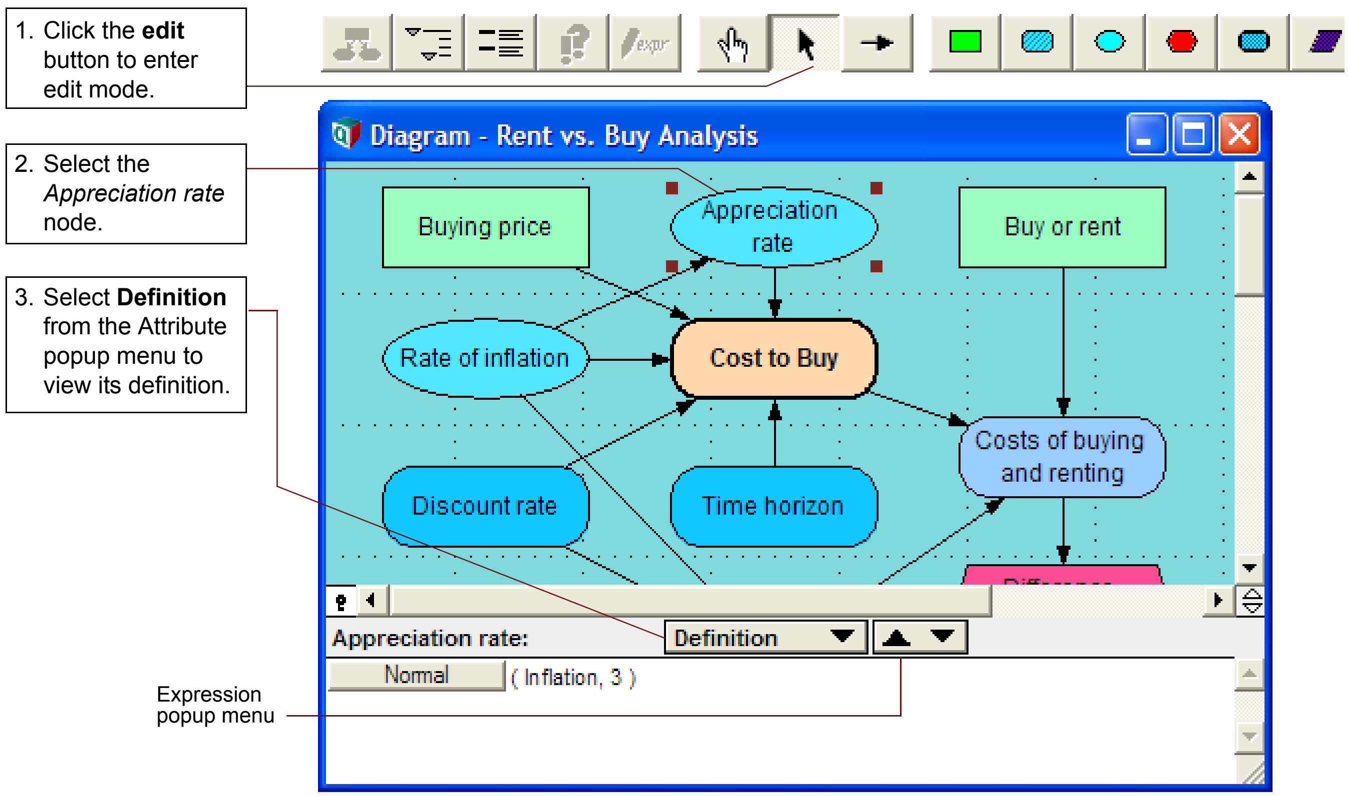
For more information about importance analysis and the steps to create an importance variable in your own model, see “Scatter plots” in the “Sensitivity and Uncertainty Analysis” chapter of the *Analytica User Guide*.

Performing parametric (sensitivity) analysis

Parametric analysis (also called ***sensitivity analysis***) involves varying the value of an input variable to examine its effect on a selected output. Performing sensitivity analysis often provides useful insights into how small changes in input variable values affect the desired outcome.

Because the importance analysis in the section “Importance analysis” revealed that *Appreciation rate* caused most of the uncertainty in *Difference between buying and renting,* you will start the parametric analysis with that input variable. You will change *Appreciation rate*’s definition from a probability distribution to a list of alternative values, and analyze the effect on the *Difference between buying and renting* output.

Before proceeding, click the edit button in the toolbar to switch into edit mode. In edit mode you can modify the model: adding and removing nodes, and modifying existing nodes. Then click the key icon to open the **Attribute** panel, then select the *Appreciation rate* node, and then select **Definition** from the Attribute popup menu to view its definition.



When the **Definition** attribute is displayed, the Expression popup menu appears. The distribution icon in the menu indicates that the definition is a probability distribution.

Before proceeding, click the **edit** tool to switch to edit mode.

The Expression popup menu allows you to change the definition of a variable to one of several different types of expressions.

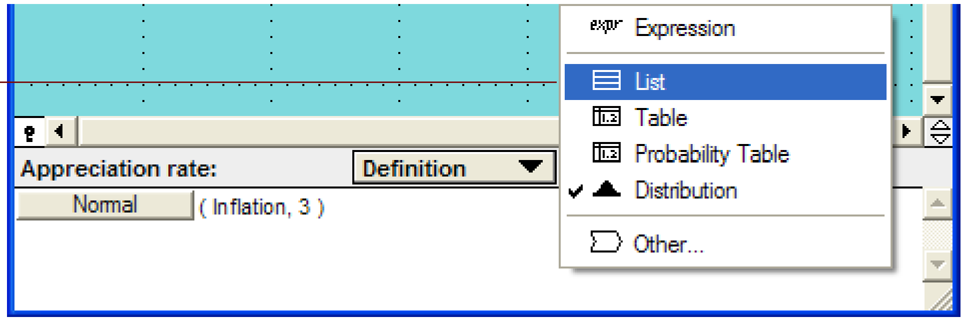
Expression types include:

- Expression, or mathematical formula
- List
- List of Labels
- Table
- Probability table
- Distribution

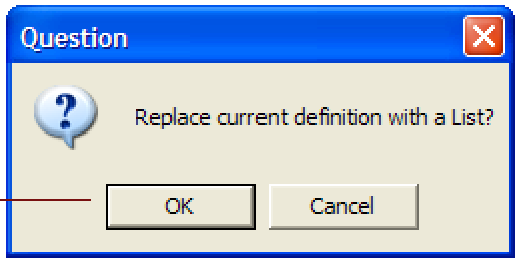
- Choice

You will now use the Expression popup menu to change the definition of *Appreciation rate* from a probability distribution to a list. You will redefine *Appreciation rate* as a list of alternative values from -10% to 10%.

4. While pressing on the Expression popup menu, drag the mouse to **List**, and release the mouse button to select **List**.



5. Click the **OK** button or press the *Enter* key to confirm that you want to change the definition from a distribution to a list.

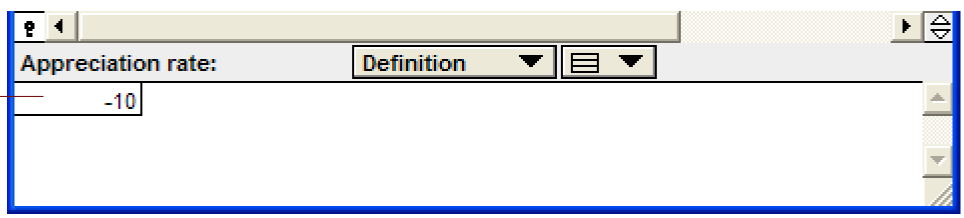


Note that the icon on the Expression popup menu changes to indicate that **List** is selected.

When a definition is first changed to a list, a cell (indicated by a box around it) appears in the definition. The first cell in the list initially contains the expression that was previously in the definition. In this case, you see the expression for a normal distribution (`Normal(Inflation,3)`).

You will replace the entry with a number and add cells to perform parametric analysis.

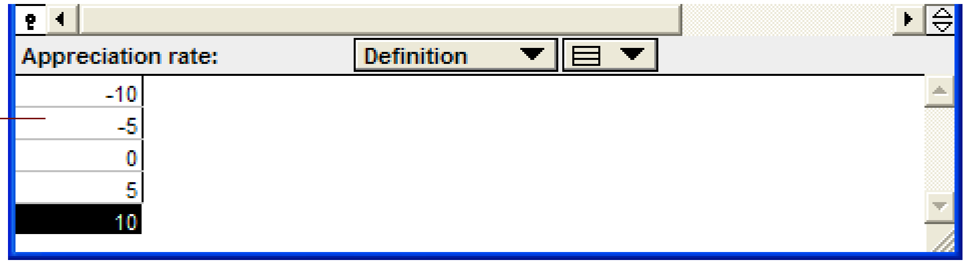
6. Select the cell by clicking in it. Type the value ***-10*** and press the *Enter* key.



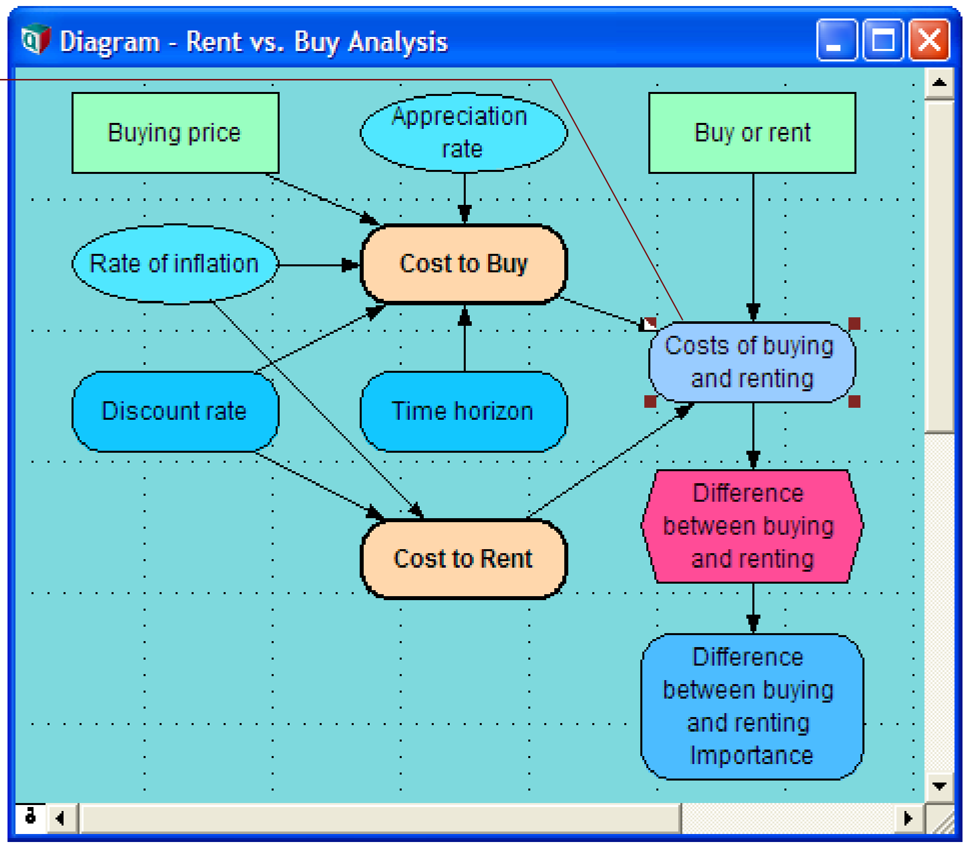
Tip In Analytica, you add cells to a list by pressing the main *Enter* key, not the numeric keypad *Enter* key.

A new cell appears with the value -9. Change its value to -5. After you have entered two values, as you press *Enter* to add a new cell, Analytica automatically fills in the new cell with a value based on the difference between the last two values. You can override the automatic value by typing the desired value.

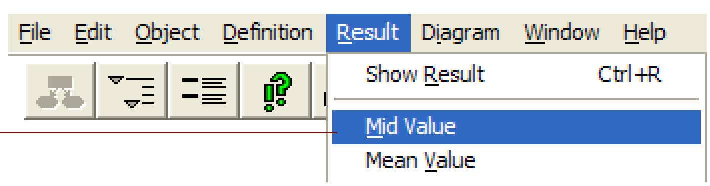
7. Type ***-5***, followed by the *Enter* key. ***0*** automatically appears. Press the *Enter* key two more times; automatically, ***5*** and ***10*** appear.



8. Select the *Costs of buying and renting* node.



9. Select **Mid Value** from the **Result** menu.

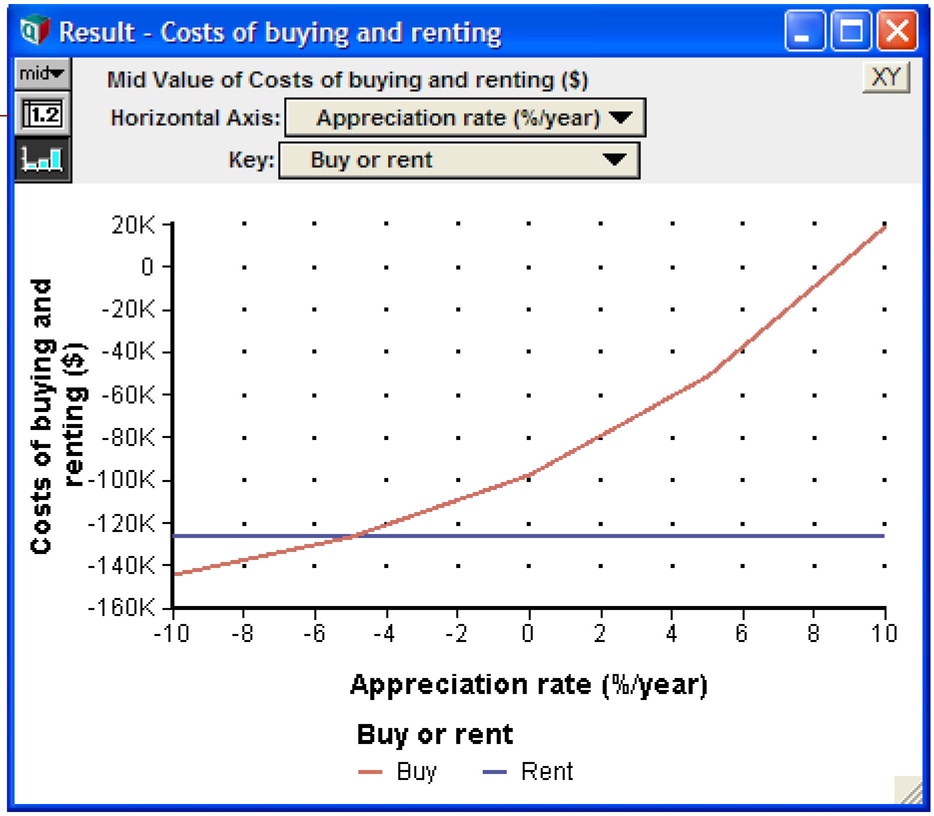


The resulting graph shows the mid value of buying and renting as a function of *Appreciation rate*, which varies from -10% to 10%, as you just entered.

Appreciation rate is informally called an ***index*** because it characterizes a dimension of another variable's value, in this case, *Costs of buying and renting*.

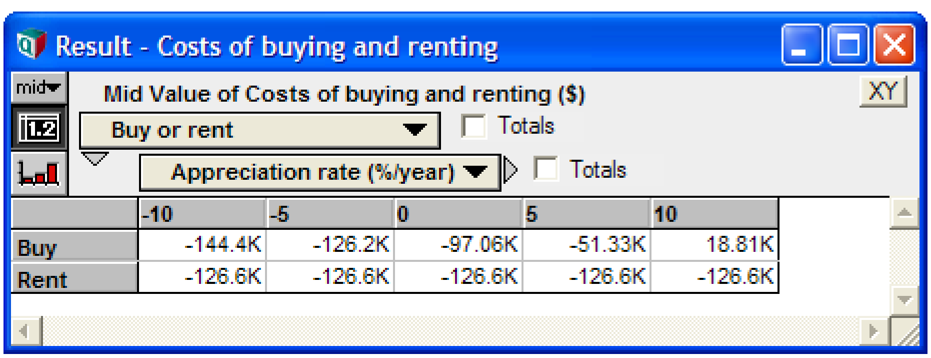
The graph shows that at an *Appreciation rate* of about -5% per year, renting and buying costs the same. If it is less than -5%, it would be better to rent; if it is greater than -5%, it would be better to buy.

10. Click the **Table** button to view the result as a table.



The table shows the values computed for each parameterized value of *Appreciation rate*.

11. Click the **Diagram** button to return to the *Rent vs. Buy Analysis* **Diagram** window.



	-10	-5	0	5	10
Buy	-144.4K	-126.2K	-97.06K	-51.33K	18.81K
Rent	-126.6K	-126.6K	-126.6K	-126.6K	-126.6K

Evaluating alternative decisions

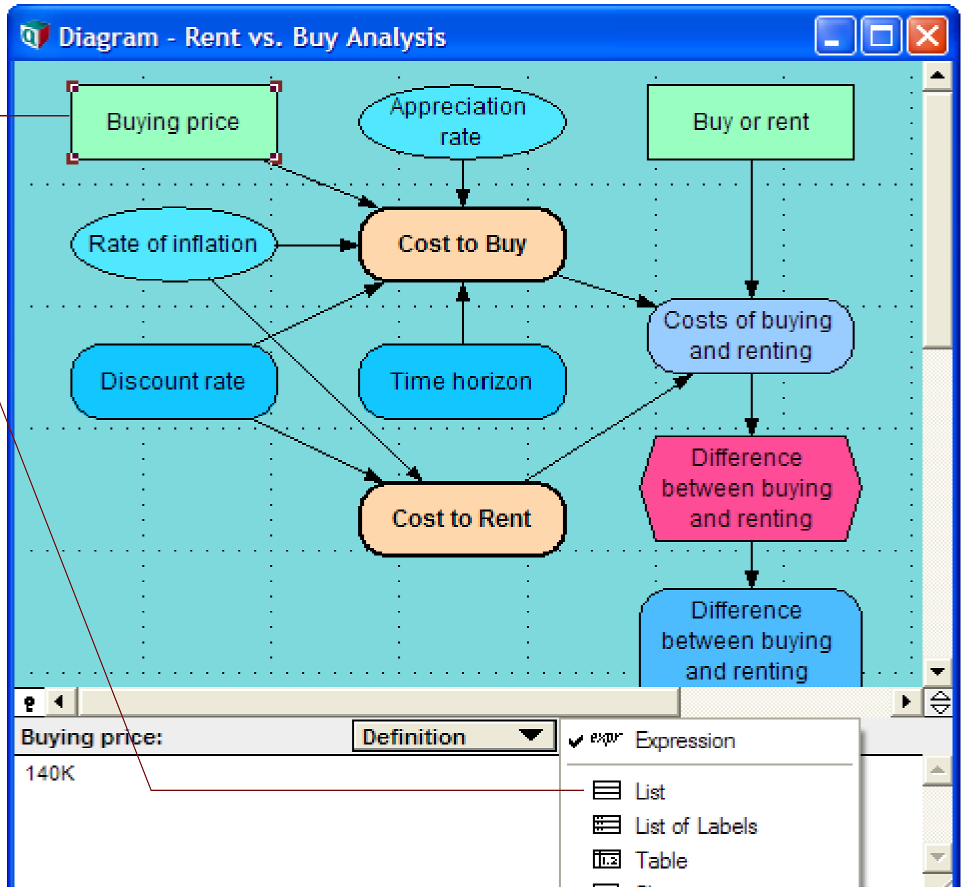
Analytica allows you to perform sensitivity analysis on several variables simultaneously.

In this section, you will change *Buying price* to compare results based on alternative decisions. In doing so, you will perform parametric analysis on both *Buying price* and *Appreciation rate* at the same time.

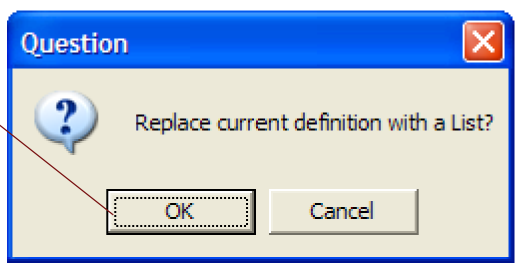
1. Select the *Buying price* node.

2. Select **List** from the Expression popup menu.

   Analytica asks you to confirm that you want to make this selection.



3. Click the **OK** button to proceed.

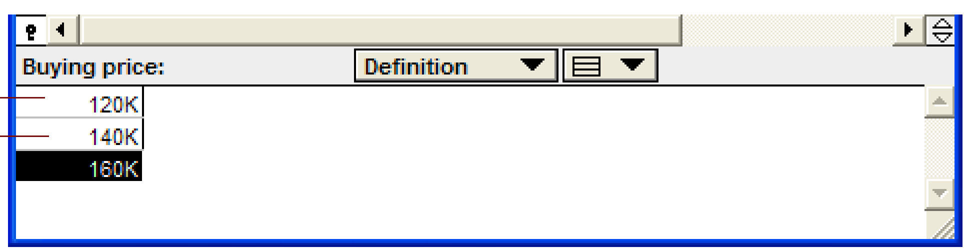


The first cell in this list contains the expression for the previous definition, **140K**. You will change this value, and add additional cells, as you did in Step 6 on page 51 and Step 7 on page 52.

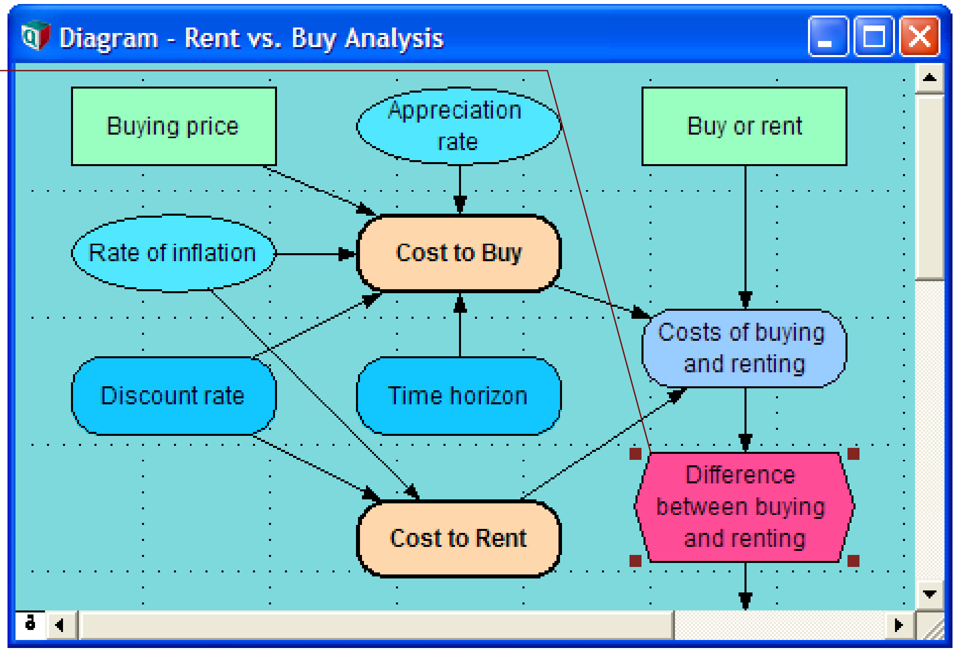
4. Click in the cell to select it. Type ***120K*** and press the *Enter* key.

5. Type ***140K*** and press the *Enter* key.

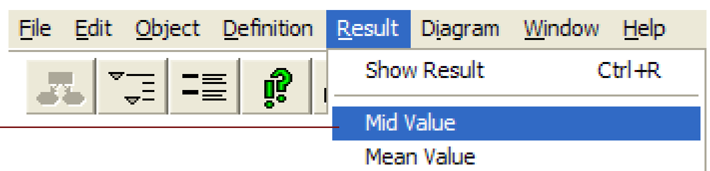
   160K is automatically entered in the next cell.



6. Select the *Difference between buying and renting* node.



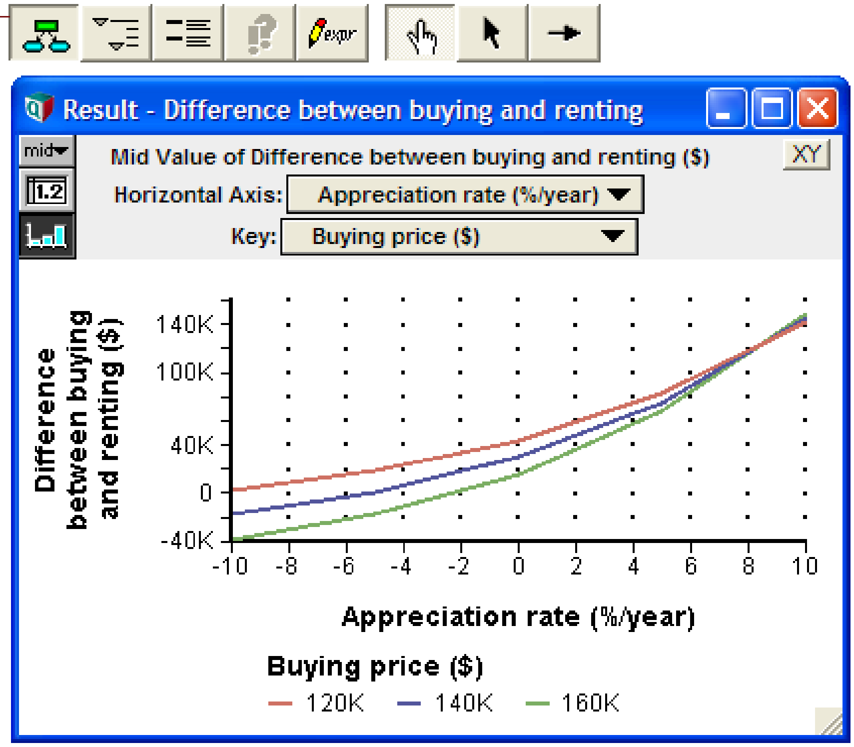
7. Select **Mid Value** from the **Result** menu to recompute and display its value.



The **Result** window appears displaying the variable's mid value. The *Difference between buying and renting* variable is three curves, one for each *Buying price*. Below the graph is a key to identify each curve.

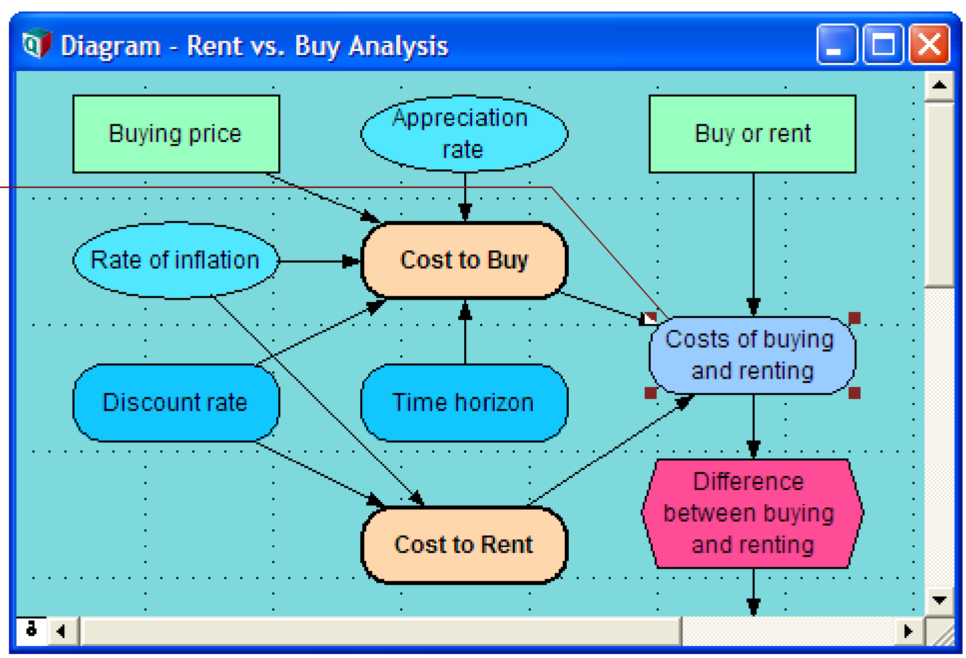
When you examine the mid value results, you can see that only a $160K home, coupled with an appreciation rate of -2%/year or less, or a $140K home, coupled with an appreciation rate of -6%/year or less, results in renting being cheaper than buying. So, what is the best buy, a 120K home or a 160K home? That depends on what you anticipate the appreciation rate will be. For appreciation rates less than 9% per year, the less expensive home is the better investment. For higher appreciation rates above 9%, the more expensive home provides a larger return.

8. Click the **Diagram** button  to return to the *Rent vs. Buy Analysis* **Diagram** window.

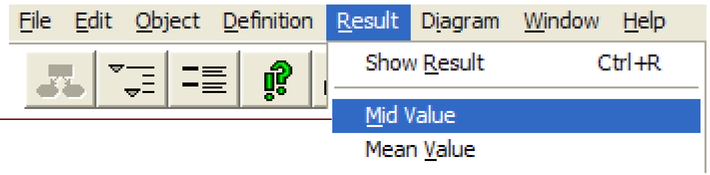


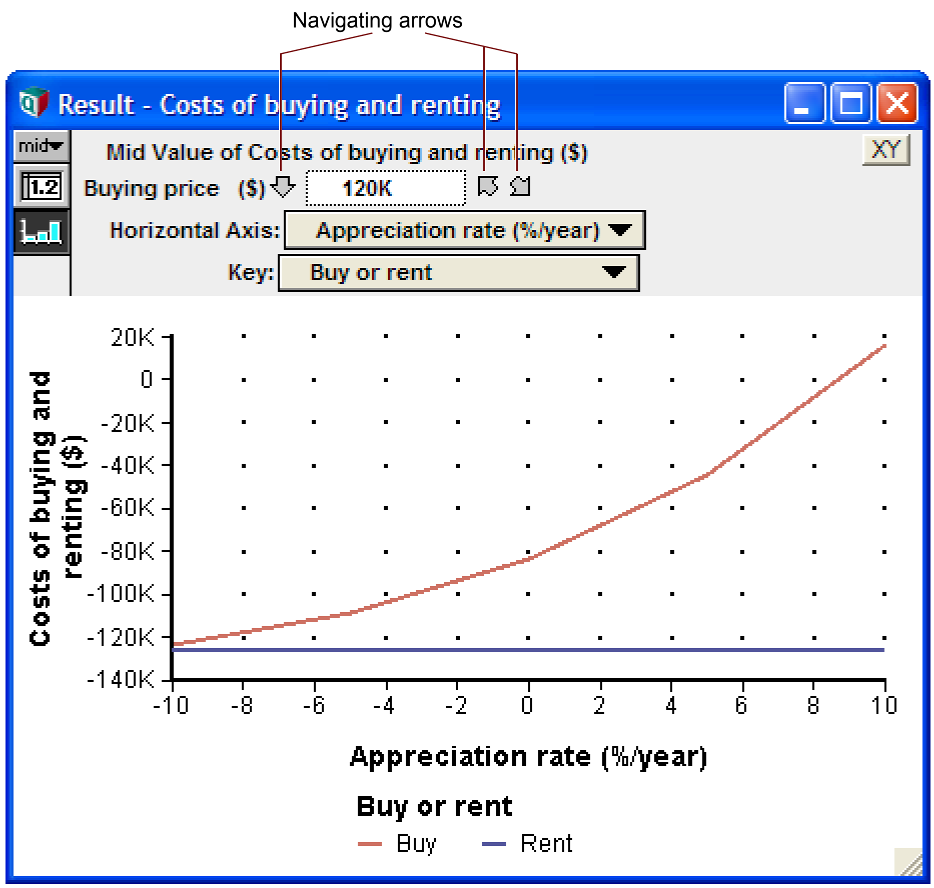
Remember that the cost of renting has been held constant. To further investigate the effect of this, you will examine the *Costs of renting and buying* node.

9. Select the *Costs of buying and renting* node.



10.Select **Mid Value** from the **Result** menu to recompute and display its value.



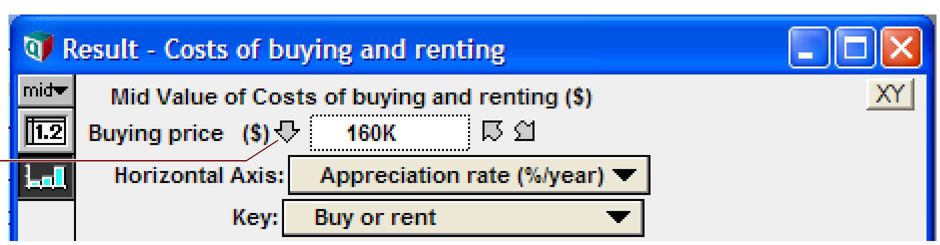


The result has three dimensions, *Buying price*, *Buy or rent*, and *Appreciation rate*, shown in the figure above.

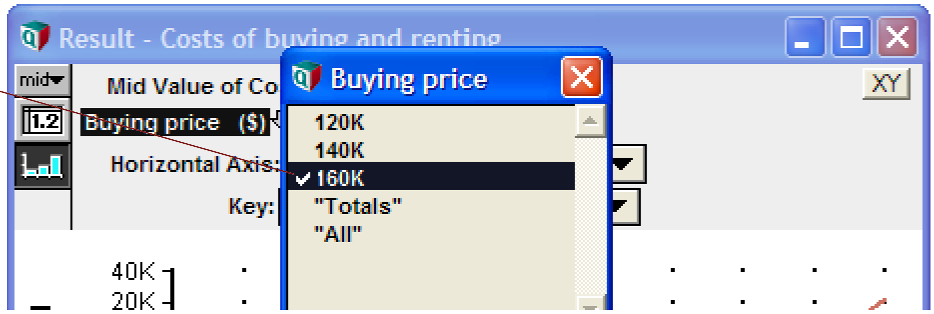
Because only two dimensions can be shown in the graph, Analytica chooses one value of the third dimension to display, in this case, *Buying price* equals **$120K**.

Use the navigating arrows to display different values of the *Buying price* index.

11.Click the down arrow ⇩ to display a scrolling list of alternative values for the variable.

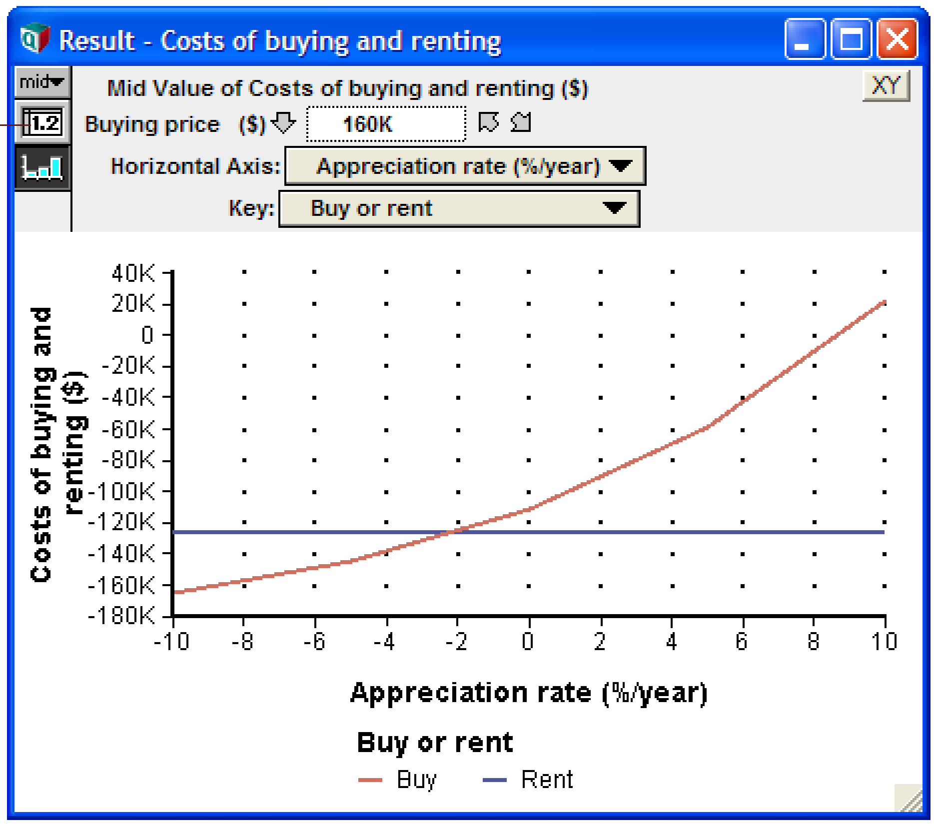


12.Click **160K** to select it.



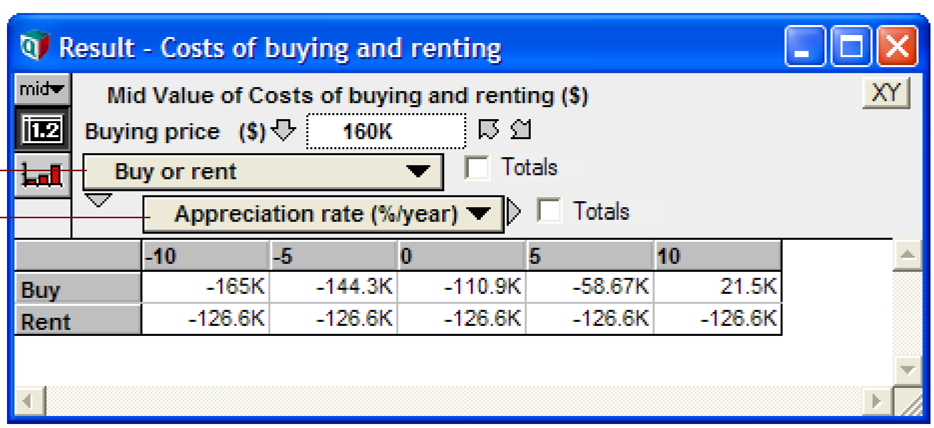
The graph changes to show the mid value of *Costs of buying and renting* given that the *Buying price* equals **$160K**.

13.Click the **Table** button [1.2] to see the table view.



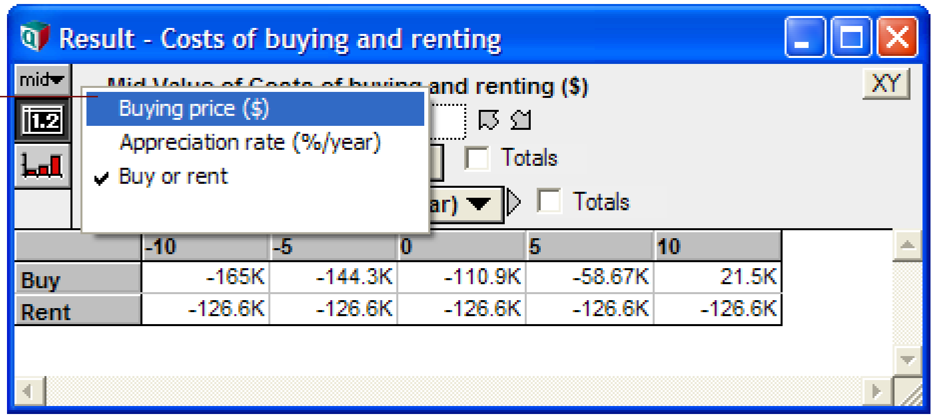
Row index popup menu

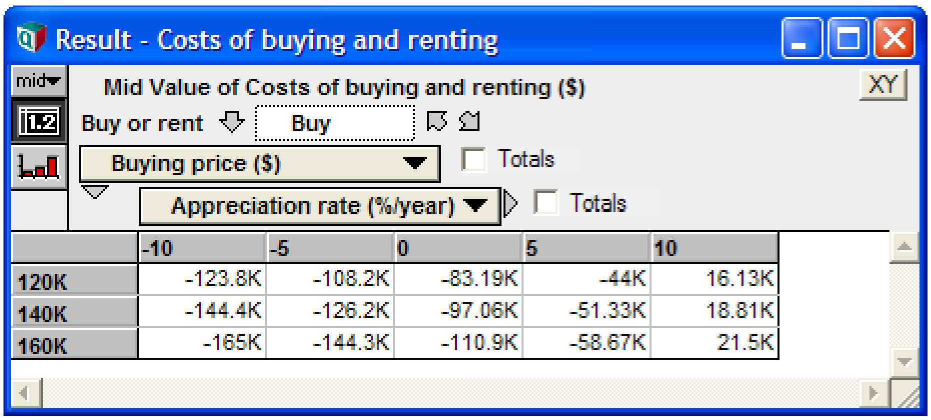
Column index popup menu



14. Select *Buying Price ($)* from the Row index popup menu.

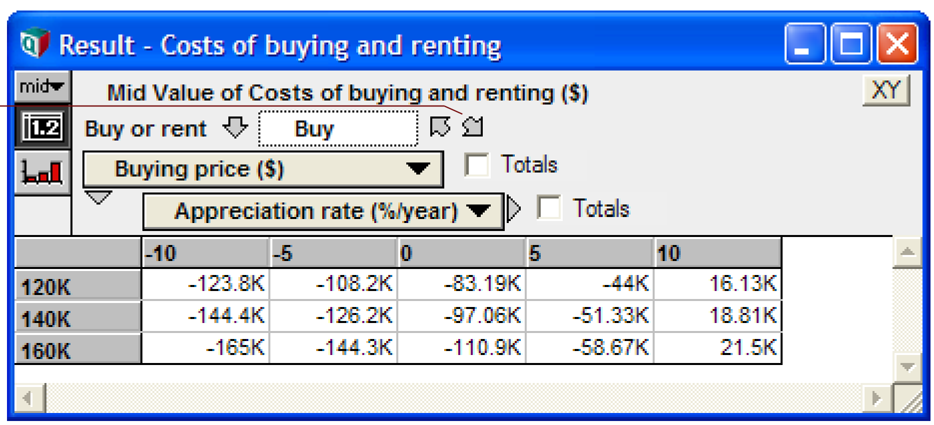
Buy or Rent becomes the third dimension with one value (Buy) displayed.



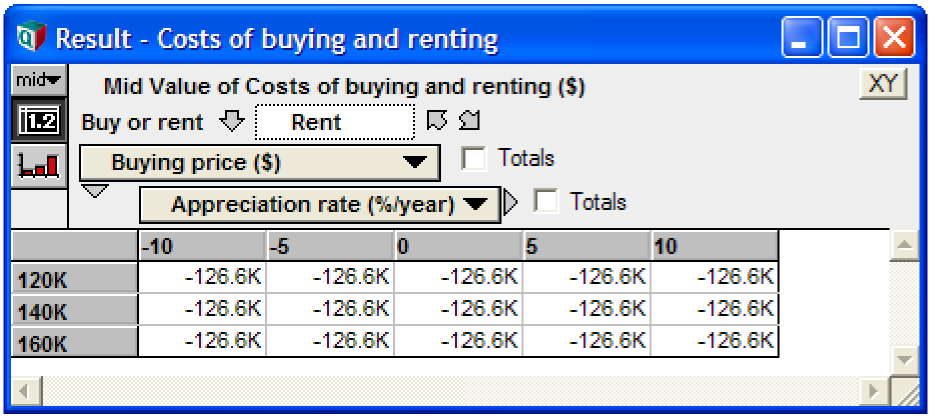


This table shows the mid value cost of buying for the parameterized values of *Buying Price* and *Appreciation Rate*.

15. Click the navigating arrow to show the corresponding table for *Rent*.



This table shows that *Cost to Rent* does not vary with *Buying Price* or *Appreciation rate*.



Analyzing the Rent vs. Buy Analysis model: summary

In this chapter, you have:

- Performed importance analysis.
- Performed parametric analysis.
- Set up and compared alternative decisions.

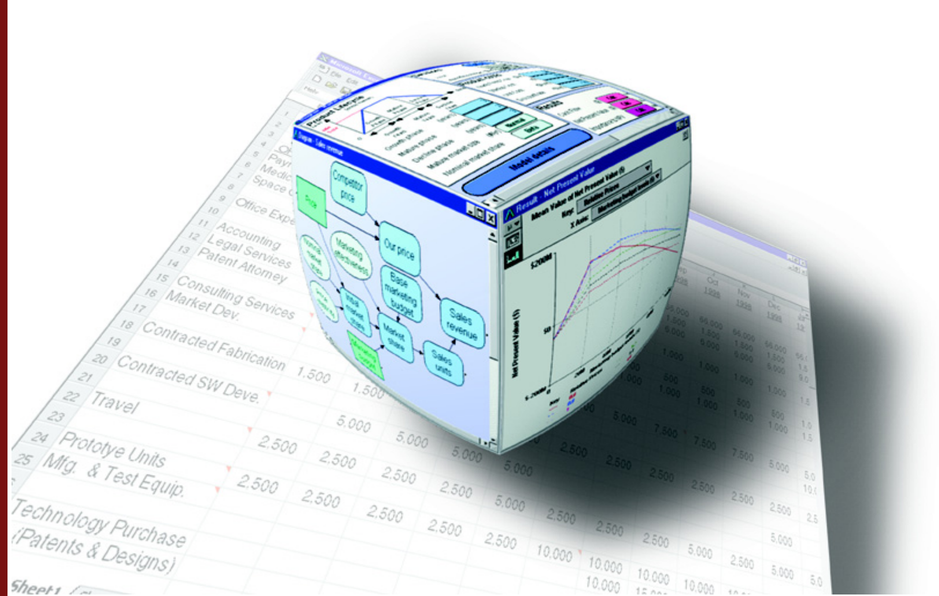
The next chapter introduces you to creating a new Analytica model.

You can quit Analytica at this point. See “Quitting Analytica” on page 23.

Chapter 4 *Creating Models*

This chapter shows you how to:

- Create a model
- Document and define variables
- Draw arrows (define dependencies) between variables
- Create a module within a model



This chapter introduces you to creating a new Analytica model.

In the process of building a model that analyzes the costs of owning and operating an automobile, you will create variables, define dependencies, add documentary text, and compute results. In addition, you will create modules.

Start Analytica by double-clicking its icon as described in “Opening the Rent vs. Buy model” beginning on page 8. Analytica opens with a blank new model.

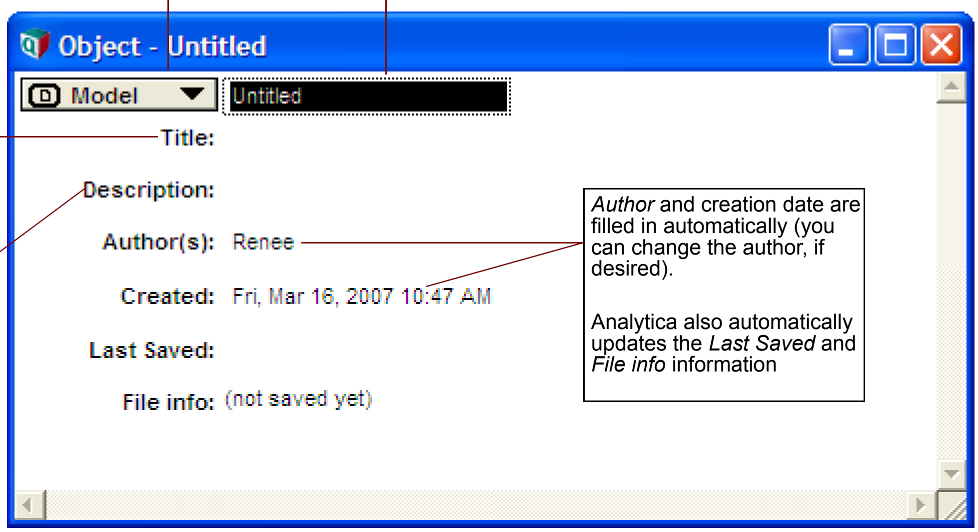
Documenting models

An untitled model window appears when you create a new model. A ***model*** consists of a root module with any number of additional modules.

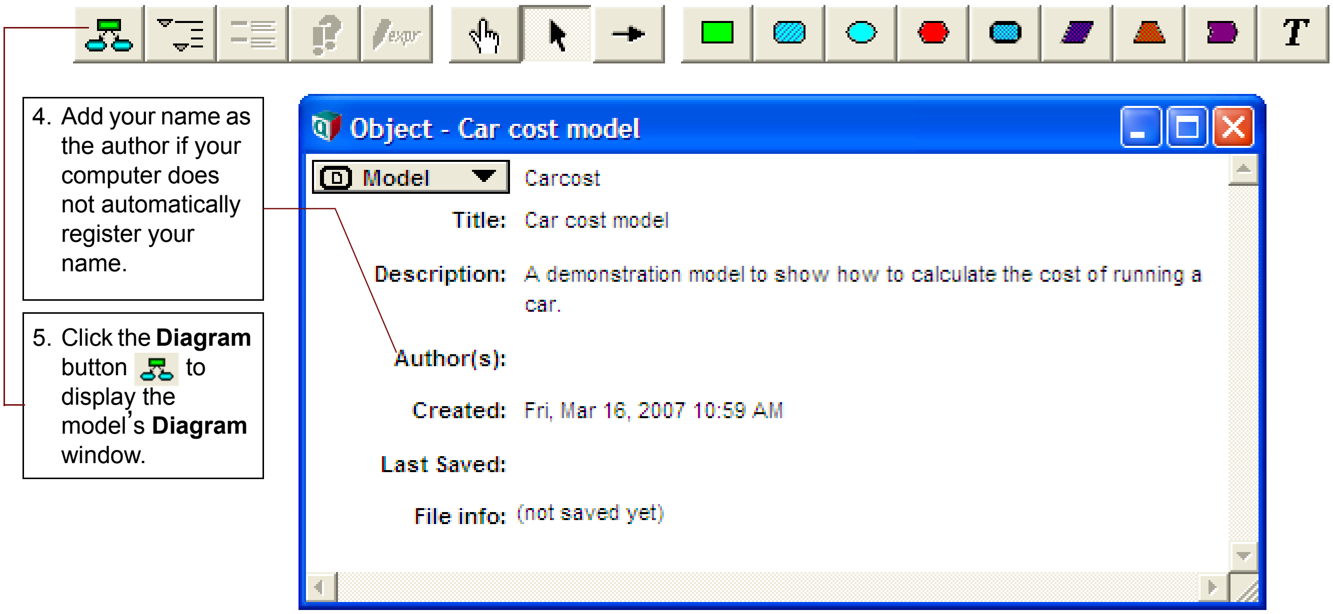
When you first create a new model, you must enter an identifier and other information, also referred to as documentation, describing its title, description, and author.

1. Type ***Carcosts*** in the Identifier field to identify your model.
2. Tab to, or click in, the Title field (to the right of the label **Title**) and type ***Car cost model***.
3. Tab to, or click in, the Description field (to the right of the label **Description**) and type ***A demonstration model to show how to calculate the cost of running a car***. See the following figure for the completed screen.

Identifier (holds up to 20 characters)



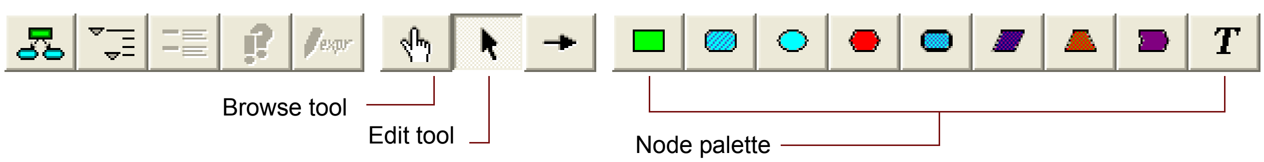
Tip You can either press the *Tab* key or use the mouse to move between fields.



Editing diagrams

In most of the three previous chapters you were in the browse mode, with the ***browse tool*** highlighted in the tool palette. In browse mode you can view an existing model, without changing its structure. When you create a new model, the ***edit tool*** is selected by default. You use the edit tool to create or change a model.

Be sure to note which tool is selected throughout the remainder of this tutorial.



When the edit tool is selected, a menu of icons is displayed in the ***node palette***. These icons represent the different node types and allow you to add nodes to the diagram.

Creating variables

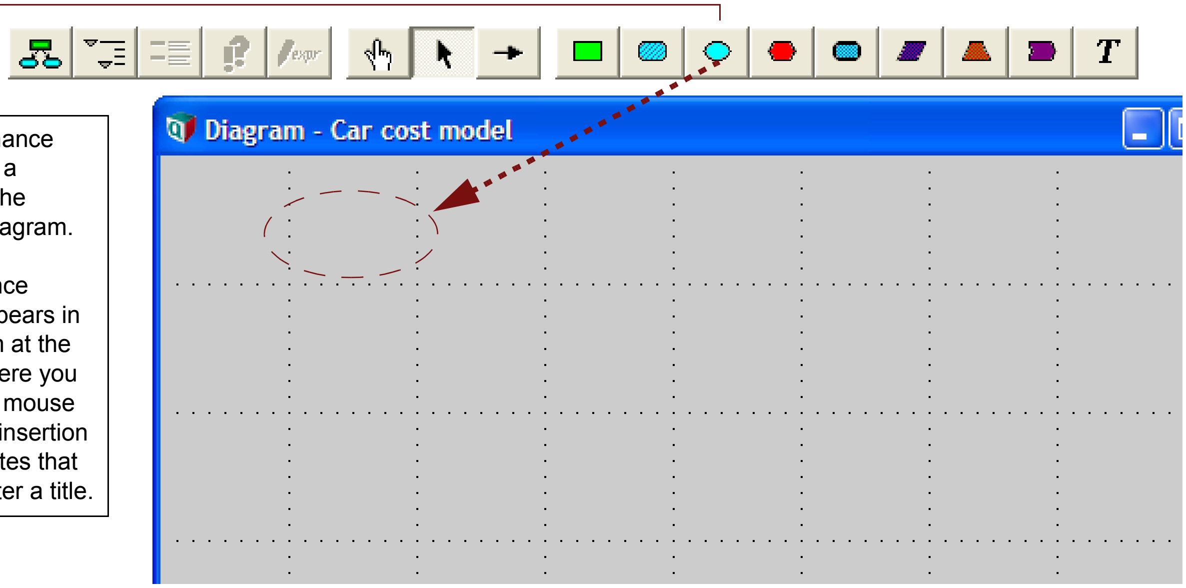
In this section, you will create variables in the *Car Cost* model.

Each variable has a node type; select the node type based on what you know about the variable. If you are not sure what kind of variable it is, or if you know that the variable has a single value, represent it as a general variable. See “Recognizing influence diagrams” on page 26 of this tutorial for a description of the node types.

The first variable you will create is for the cost of fuel. Because the cost of fuel changes, you will use a ***chance variable***.

1. Drag the chance icon to a position in the influence diagram.

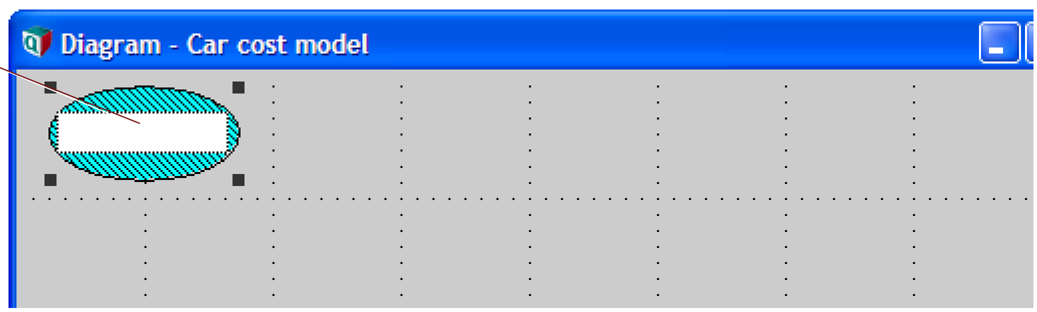
   A new chance variable appears in the diagram at the location where you release the mouse button; the insertion point indicates that you can enter a title.



Tip As you build a model, you should select descriptive titles for your variables. Descriptive variable titles remind you of the model's logic and help inform others about how the model operates.

2. Type ***Fuel cost*** for the variable title.

   Press *Alt+Enter* or click any other location in the diagram to accept the title.

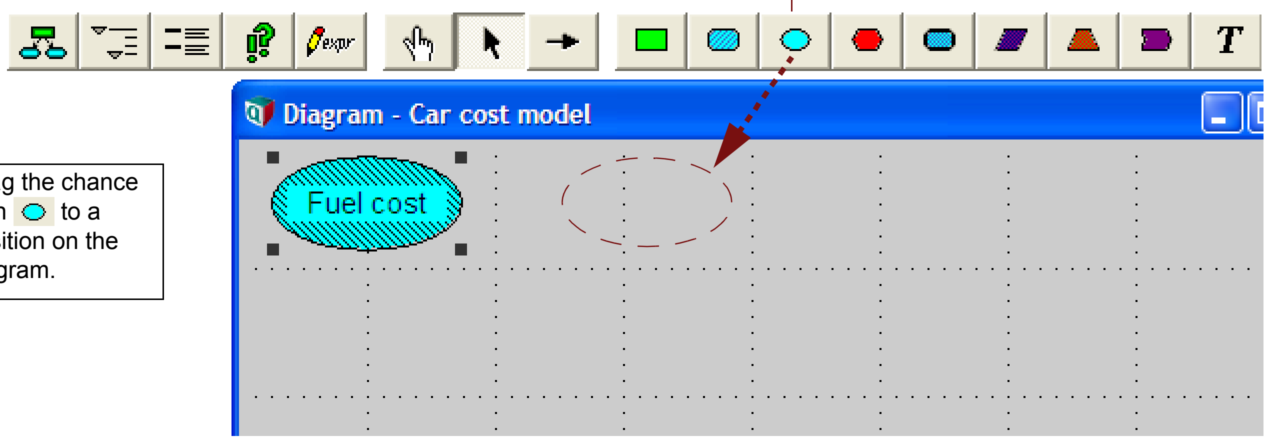


Tip *Fuel cost* is filled with a diagonal line pattern around its text, indicating that it does not have a valid definition.

Repeat Steps 1 and 2 four times to create four more chance variables that affect fuel cost. Title these variables as follows:

- *Fuel price* (price per gallon of gasoline)
- *Annual miles* (number of miles driven each year)
- *Mpg* (miles per gallon of gasoline)
- *Age* (driver's age)

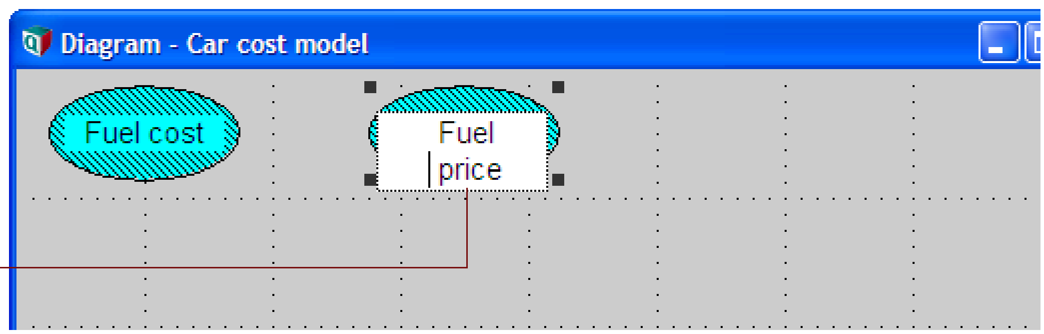
3. Drag the chance icon to a position on the diagram.



When you press *Alt+Enter*, the text in a node title wraps to fit within the node. You can manually create a new line in a title by pressing the *Enter* key at the desired break point(s).

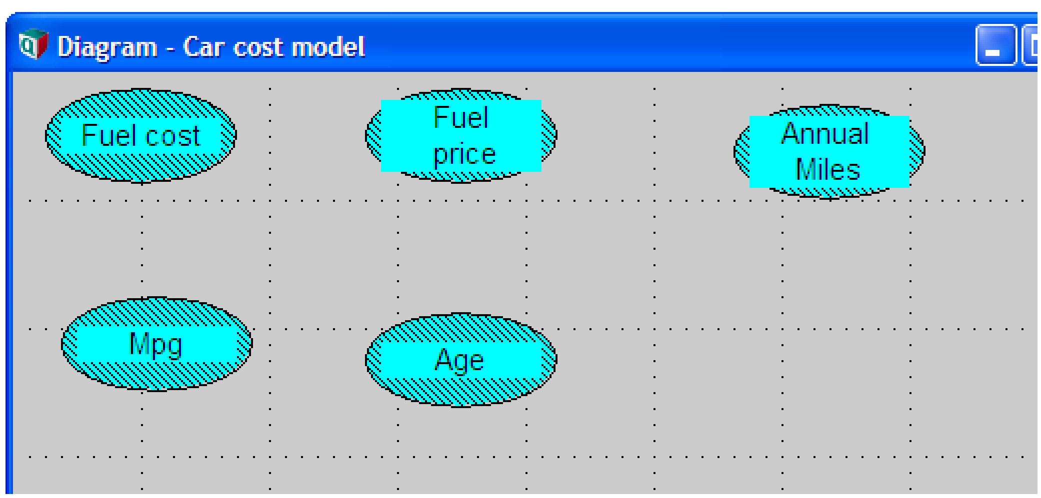
4. Type ***Fuel*** and press the *Enter* key to create a second line.

   Type ***price*** and press *Alt+Enter* to indicate that you are finished.



5. Repeat Steps 3 and 4 to create three more variables, as shown.

   Title the variables ***Annual Miles***, ***Mpg***, and ***Age***.

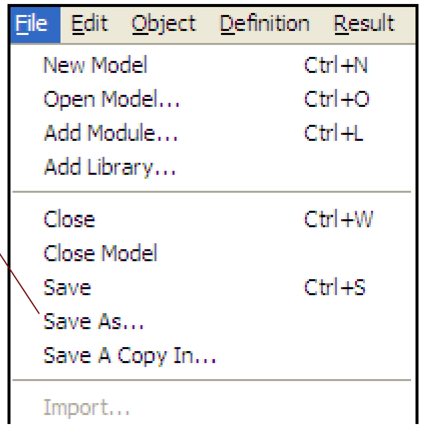


Saving your model

While creating or modifying a model, you should periodically save your changes. You should save frequently so that you do not lose very much information if your system crashes unexpectedly[2]. See “Saving your model” on page 21.

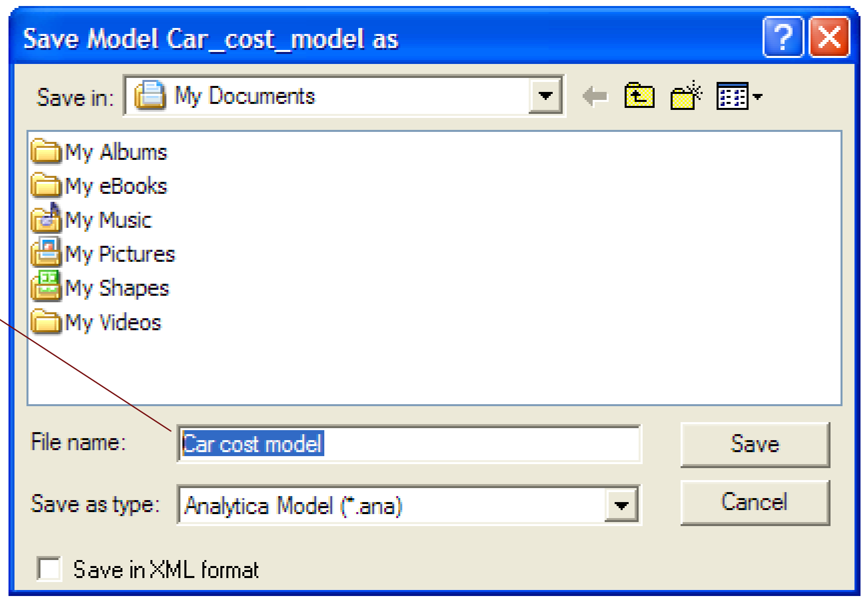
Analytica initially uses the title of your model, *Car cost model*, to name the file that contains the model. To save the file under a different name, select **Save As** from the **File** menu.

1. Select **Save As** from the **File** menu.



Analytica defaults to your *My Documents* folder.

2. Type the new file name here.



Deleting variables

Sometimes you might want to delete a variable that you previously created.

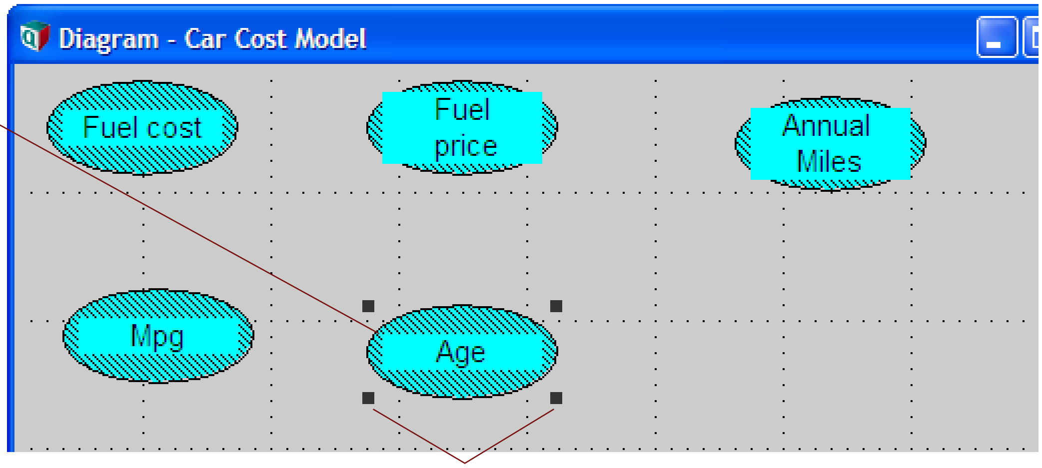
In this example, you realize that the driver's age is not relevant to your understanding of the *Fuel Cost* variable. Therefore, you will delete the *Age* variable.

2. Analytica automatically saves changes you make in your model to a backup file, so you won't lose your changes in the unlikely event of a system crash. When you open the model after an unplanned exit, Analytica asks you if you want to continue using the changes from the backup file or revert to the last previously saved version.

1. If the *Age* node is not selected, select it.

   Handles surround the node to indicate that it is selected.

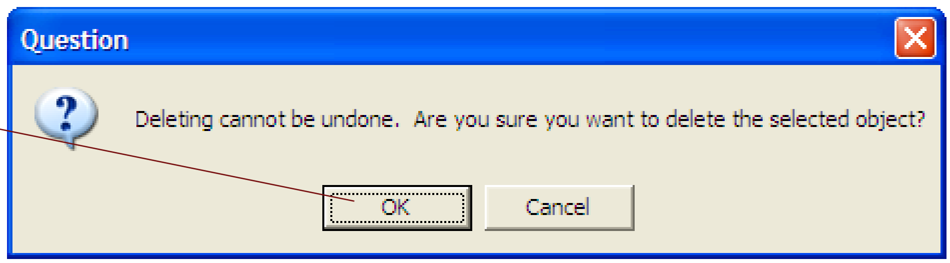
2. Select **Clear** from the **Edit** menu, or press the *Delete* key.



Handles

The **Delete** command cannot be undone; Analytica asks you to confirm that you want to delete.

3. Click **OK** to confirm that you want to delete the selected object.



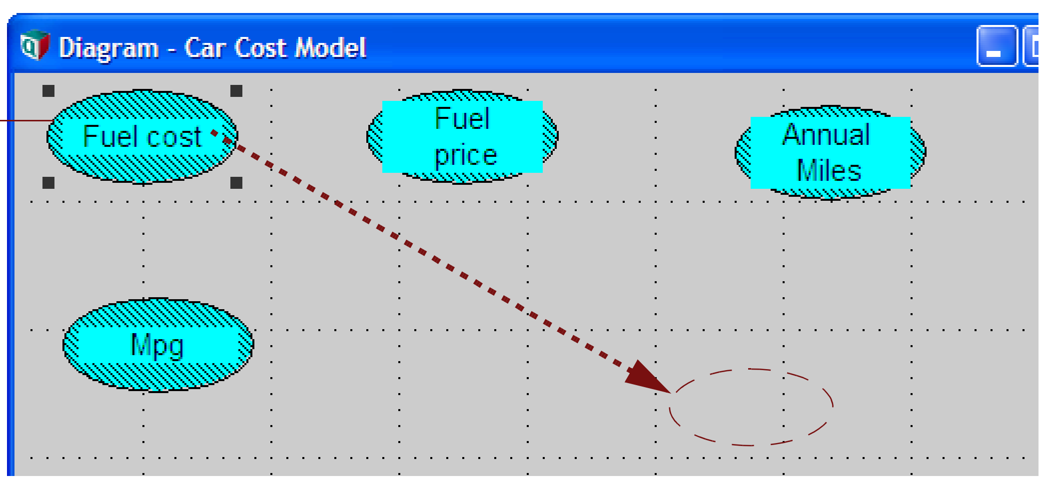
Moving nodes

When you create a model, you should try to structure the model layout to make the model logic easy to understand. As you refine your model, however, you undoubtedly will want to group nodes in different ways. You can move nodes easily to organize your influence diagrams in the best way possible.

In this section, you will move variable nodes so that the three variables contributing to fuel cost surround the *Fuel cost* node. You begin by moving a single variable node in the diagram. You will then select and move multiple nodes at once.

1. Drag the *Fuel cost* node to the position shown by the arrow.

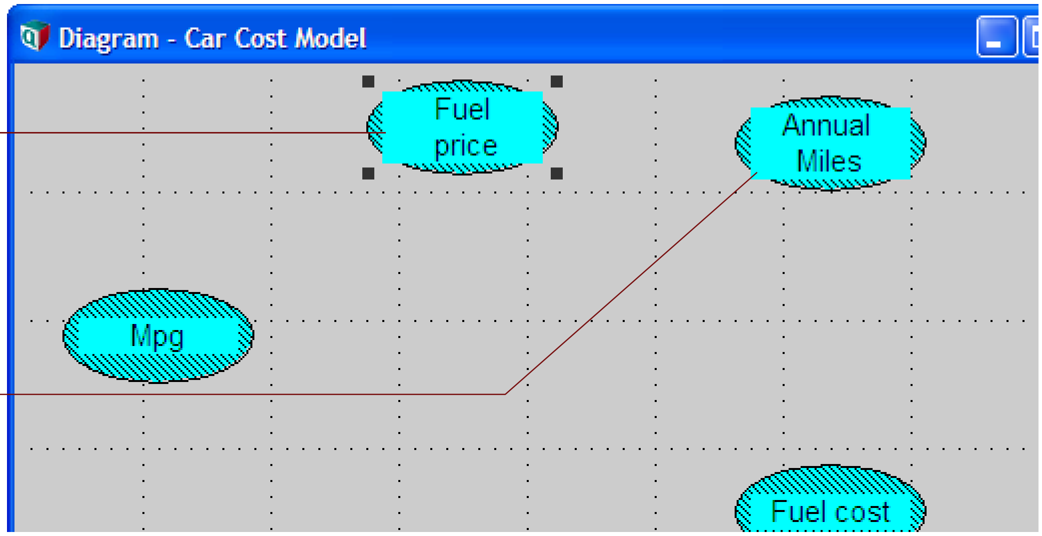
   The node reappears at the location where you release the mouse button (as shown in the following figure).



2. To move the *Fuel price* and *Annual Miles* nodes at the same time, first select the *Fuel price* node.

3. Hold down the *Shift* key and select the *Annual Miles* node.

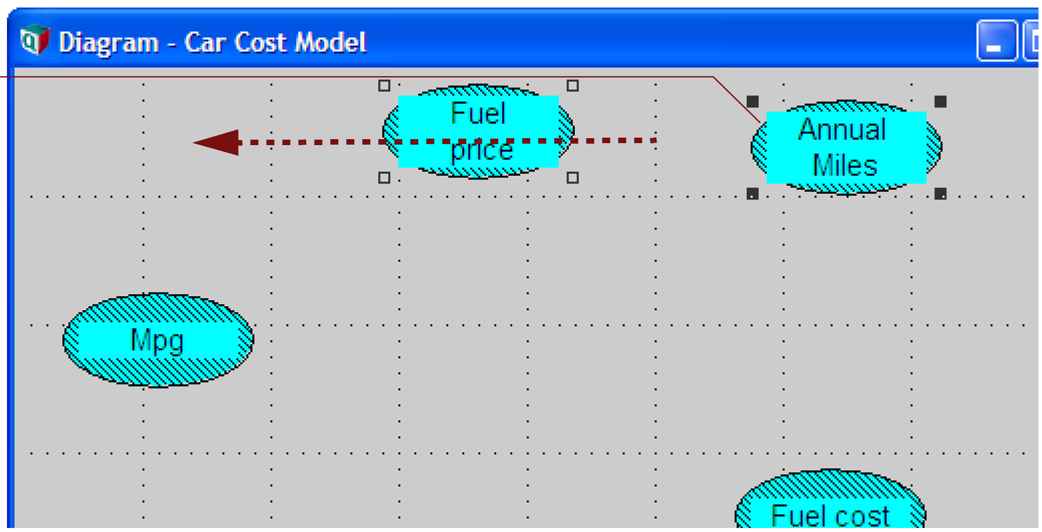
   Now, both nodes are selected (as shown in the following figure).



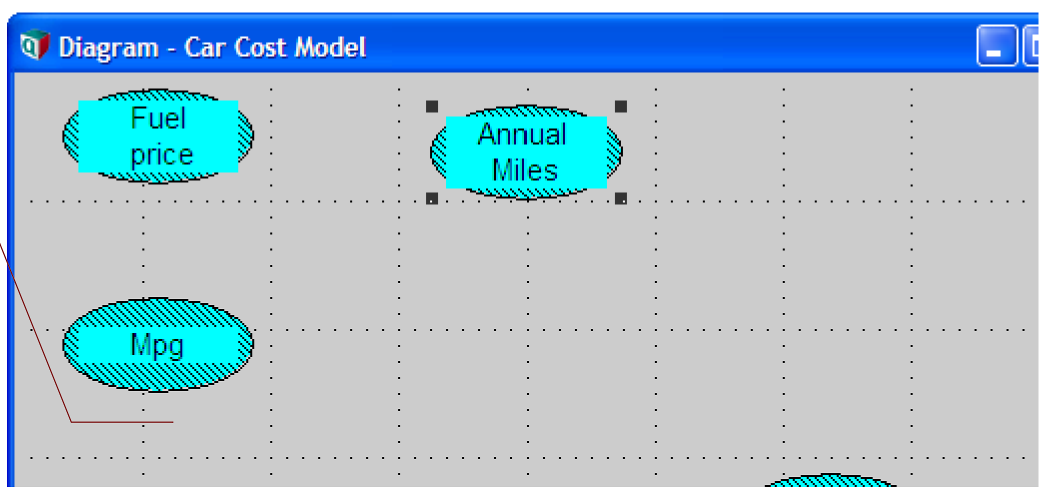
Another way to select multiple nodes is to drag a selection box, using the edit tool, around the nodes you want to select.

4. Drag the *Annual Miles* node to the left, to the position shown by the arrow.

   Both nodes reappear at the location where you release the mouse button (see the following figure).



5. Click anywhere outside the two nodes within the diagram to deselect the nodes.



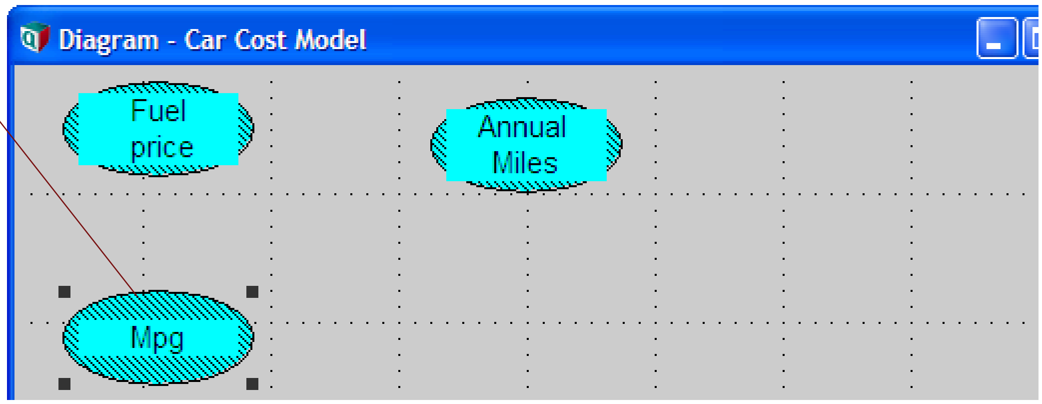
Tip You can undo or redo a drag operation by selecting **Undo/Redo** from the **Edit** menu, or by typing the keyboard shortcut, *Control+Z*.

Editing variable titles

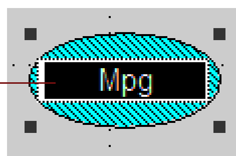
Each variable in the model is represented by a titled node in the model's influence diagram. Occasionally, you will want to change the title of an object to make the model easier to understand.

In this section, you will change the title of the variable *Mpg* to *Miles per gallon*.

1. Select the *Mpg* node.



2. Click again inside the node's title to select its text for editing.

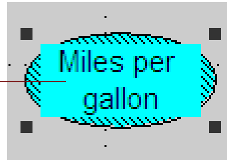


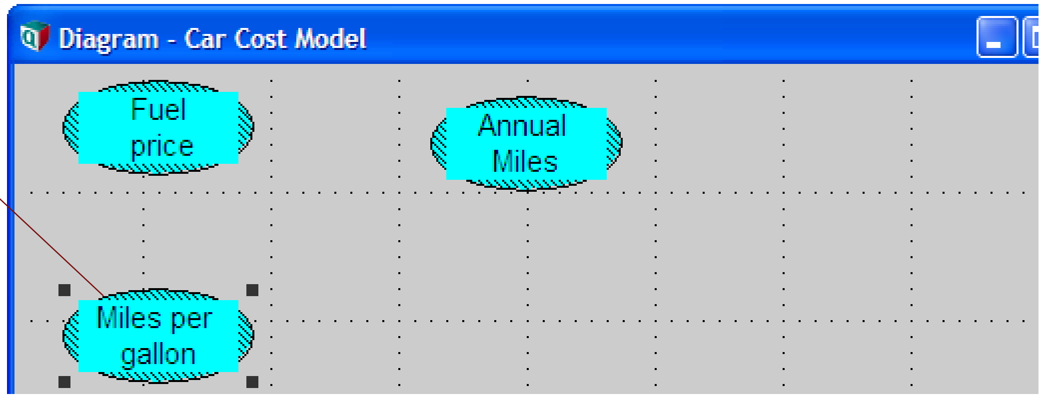
Tip Pause briefly between the click to select the node and the click to select the text within it. If you complete two single-clicks too quickly, Analytica interprets your actions as a double-click and opens an **Object** window.

If you accidentally open the **Object** window, return to the **Diagram** window by clicking the **Diagram** button .

3. Type ***Miles per gallon*** and press *Alt+Enter*.

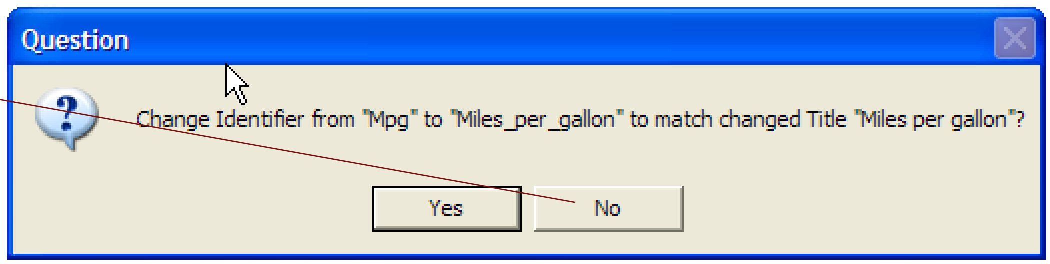
   The new title is displayed.





When you change the title of a node Analytica asks you if you want Analytica to automatically change the identifier of that node so that it will match the new name.

4. Click **No** to close this dialog box and keep the identifier as *Mpg*.

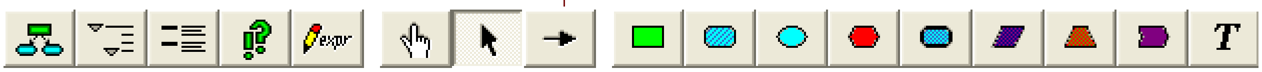


Tip You can change this behavior, to either turn off automatic updating of the identifier or to make it fully automatic, so that you are not asked. See the "Preferences dialog box" section in Chapter 4 of the *Analytica User Guide* for details.

Drawing arrows between nodes

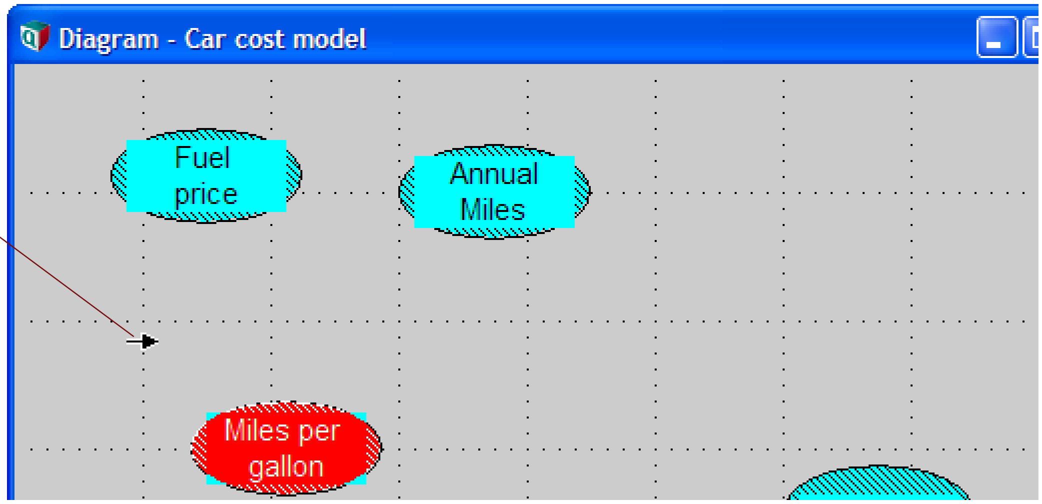
One of Analytica's most powerful features is its ability to show relationships between variables in the influence diagram. ***Influence arrows*** are used to specify the dependencies between variables.

Because the *Miles per gallon* variable influences the *Fuel cost* variable, you will draw an arrow connecting the two nodes.



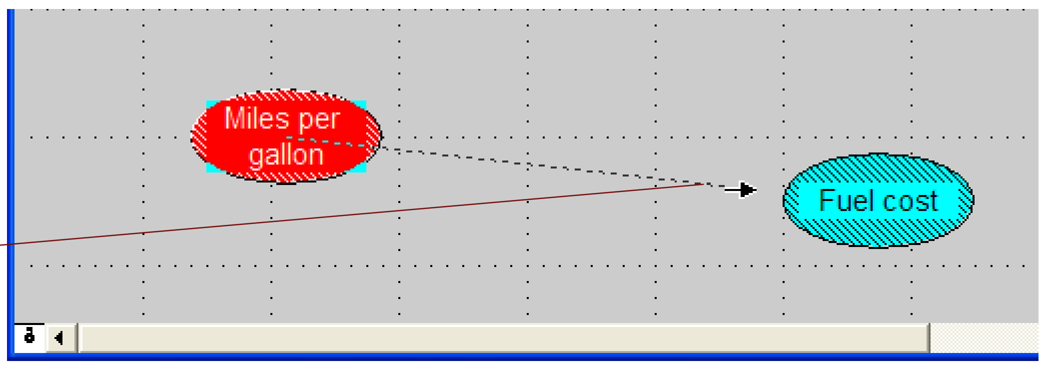
1. Select the **arrow** tool to begin drawing arrows.

   Notice that the cursor turns into an arrow.



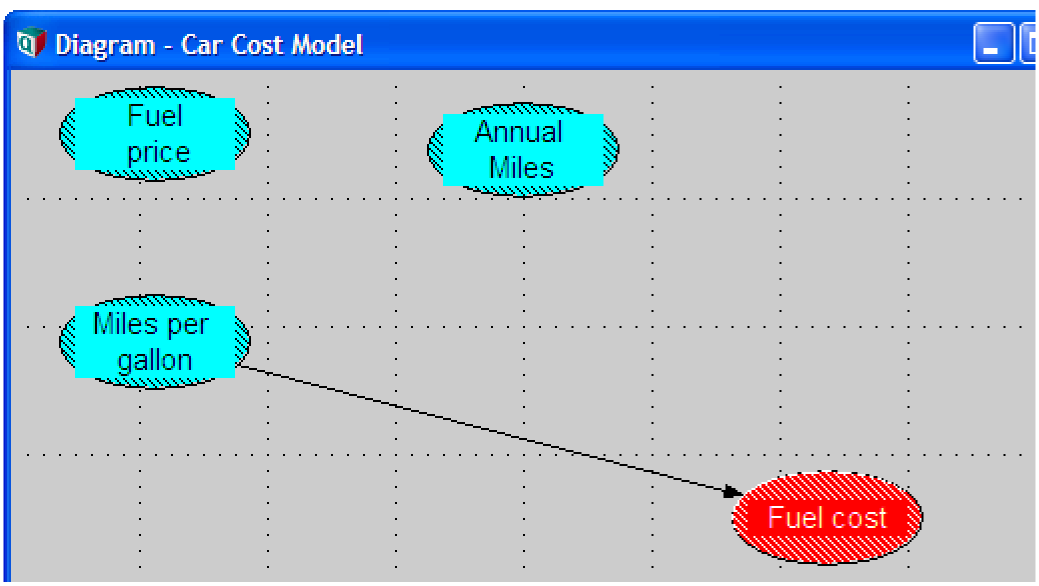
2. Drag from the *Miles per gallon* node to the *Fuel cost* node.

   Fuel cost becomes highlighted when you drag the arrow over any part of it (as shown in the following diagram).



3. Release the mouse button when *Fuel cost* is highlighted.

   The two nodes are now connected by an arrow, indicating that *Miles per gallon* affects *Fuel cost*.



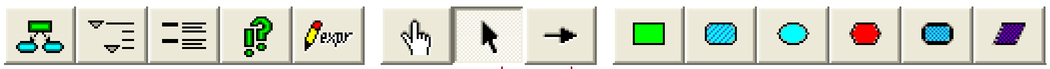
If the nodes are not connected by an arrow, repeat Steps #1 through #3.

Deleting arrows

Occasionally, you might need to delete an arrow because of an earlier mistake or a change in your understanding of the model. This section shows you how to delete the arrow that connects *Miles per gallon* to *Fuel cost*.

You can delete an arrow using either the edit tool or the arrow tool.

First, make sure you have either the edit tool or arrow tool selected.



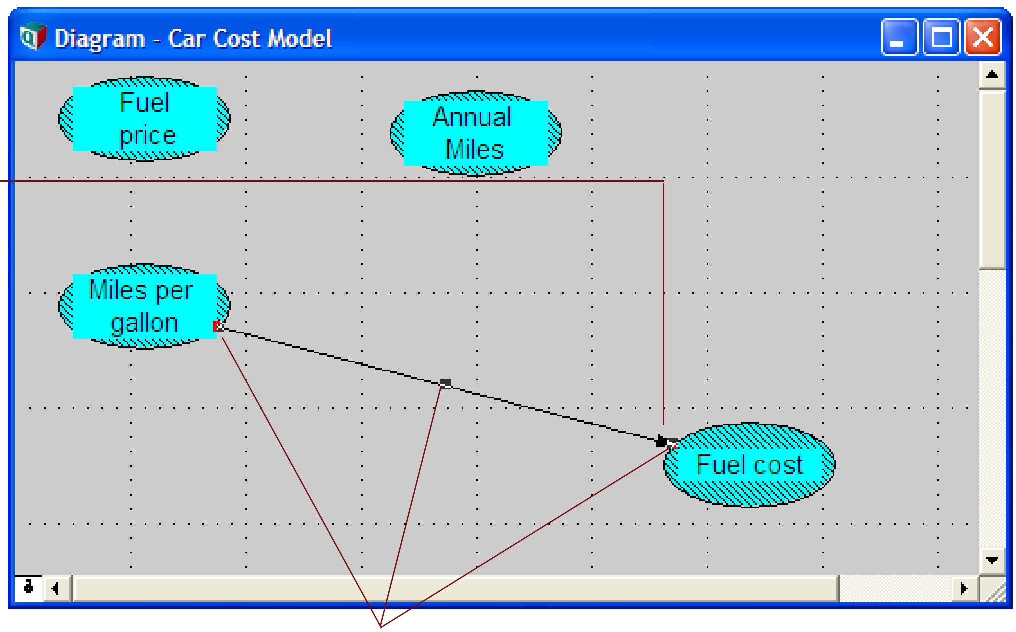
1. Make sure either the arrow tool or the edit tool is selected.

2. Select the arrow.

   It is easiest to select the arrow by clicking the arrow head. Handles appear when the arrow is selected.

3. Press the *Delete* key to delete the arrow.

   The arrow disappears.

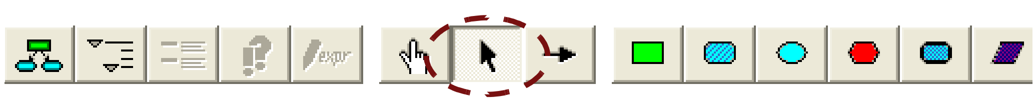


Handles

Connecting multiple arrows

When one variable is influenced by many other variables, you can draw multiple arrows at once. This section shows you how to connect the three variables contributing to the *Fuel cost* variable.

First, make sure the arrow tool is selected.

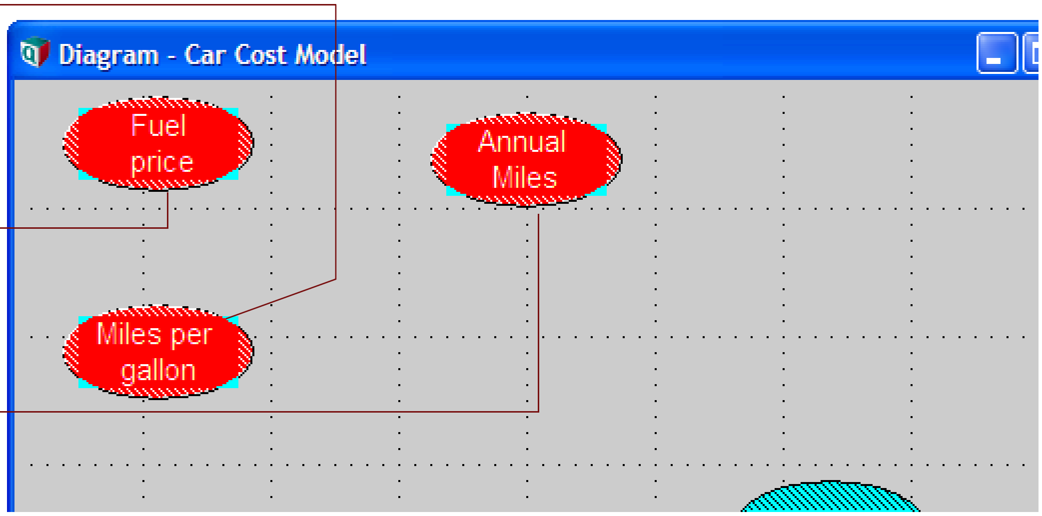


1. Select the *Miles per gallon* node.

2. Hold down the *Shift* key and select the *Fuel price* node.

3. Hold down the *Shift* key and select the *Annual Miles* node.

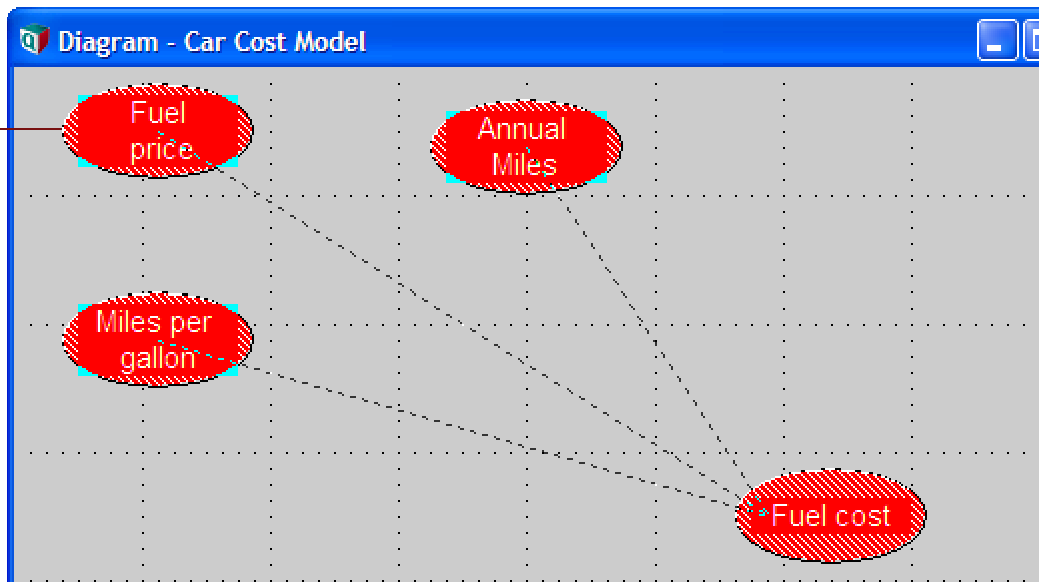
   All three nodes are now selected.



4. Drag from any one of the selected nodes to the *Fuel cost* node.

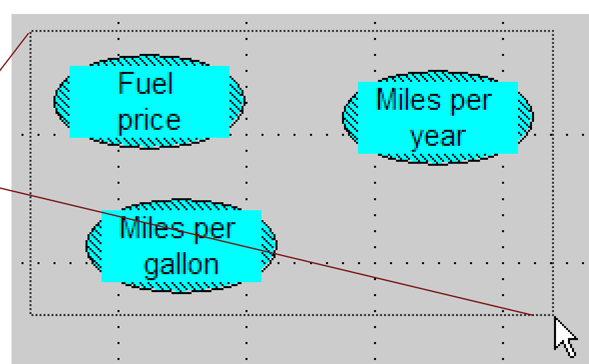
5. Release the mouse button when the *Fuel cost* node is highlighted.

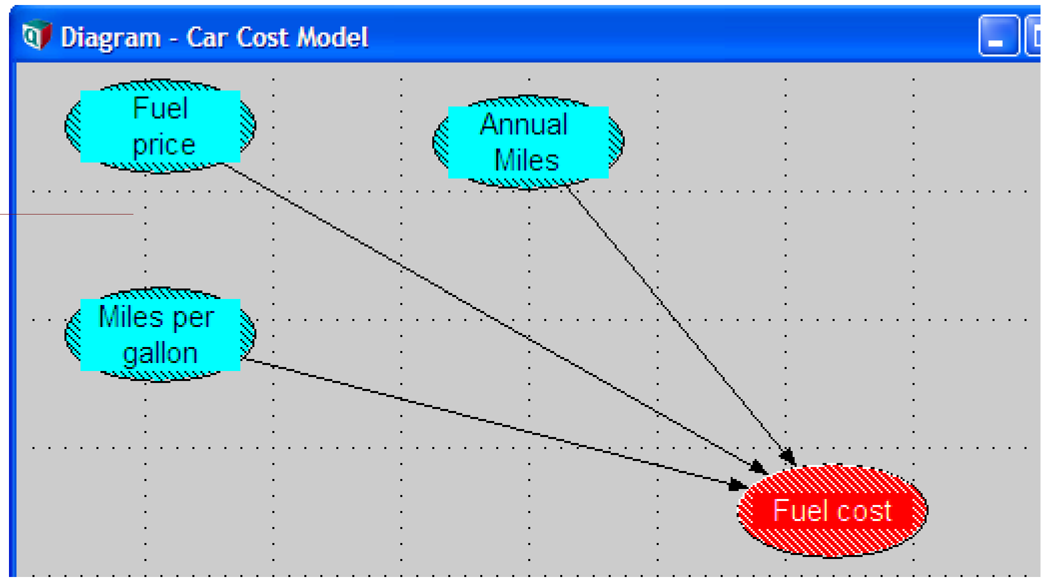
   Three arrows should now point to the *Fuel cost* node (as shown in the following diagram).



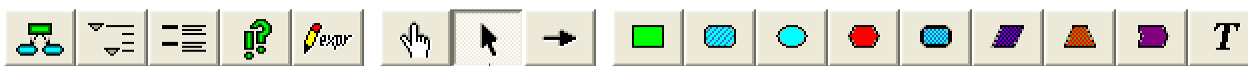
Tip You can also select multiple nodes by “rubber banding.” Press the left mouse button and move the mouse. This is a ***selection box***; all nodes completely enclosed within that region are selected when you release the left mouse button.

Rubber banding region. All three nodes inside this ***selection box*** are selected when the left mouse button is released.

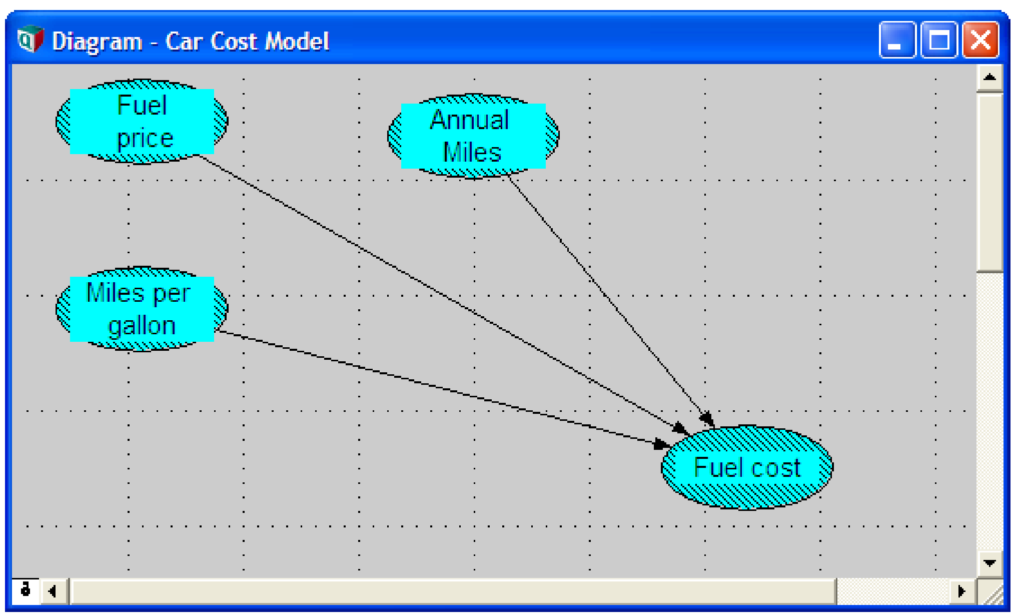




6. Deselect all of the nodes by clicking in any location in the diagram that is not on a node.



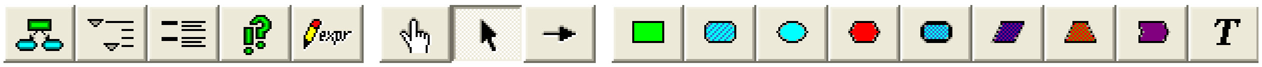
7. Select the **edit** button to turn off arrow drawing.



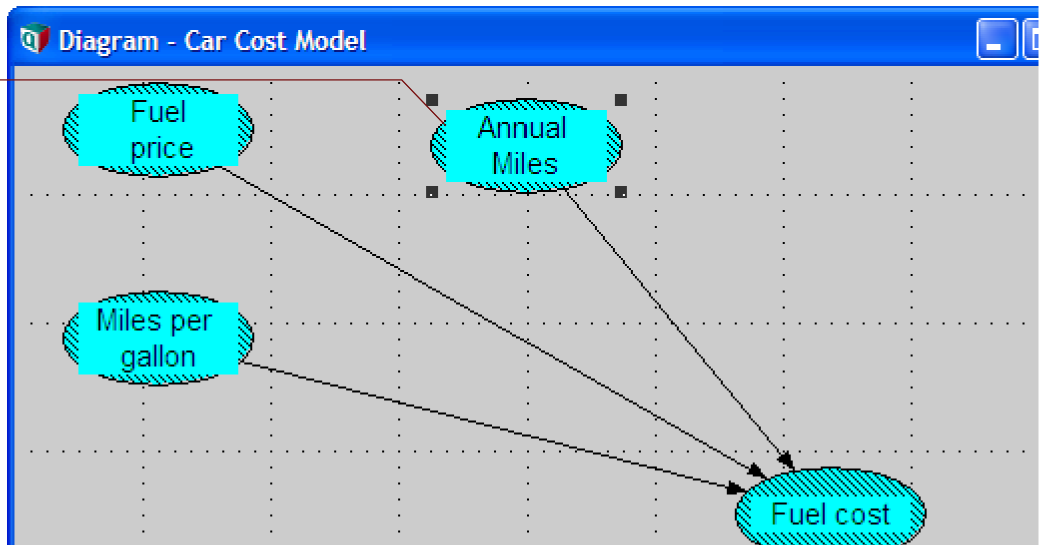
Entering attributes using the Object window

When you create an Analytica model, you can add documentation of the model and of its variables. Analytica supports integrated documentation that can be tied to every object in the model.

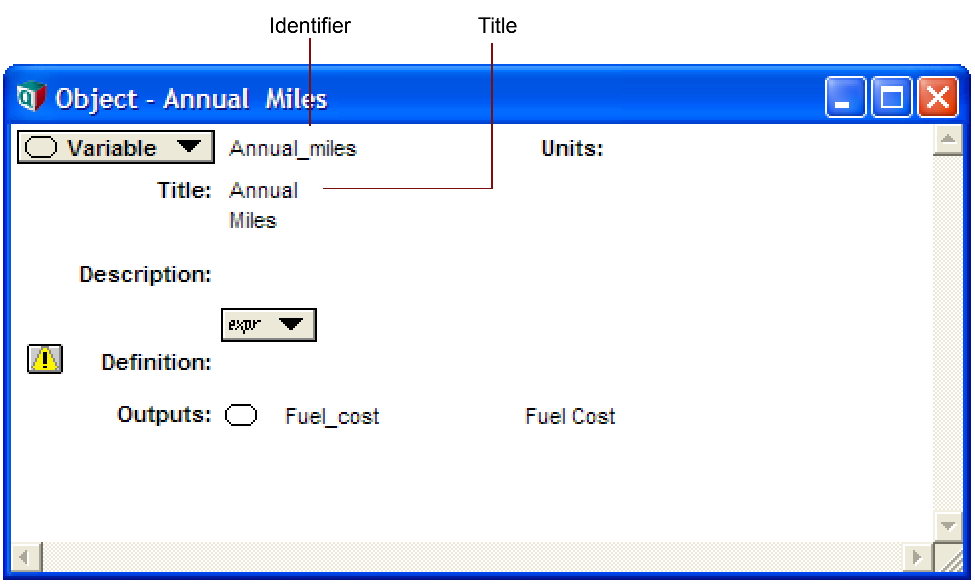
In this section, you will provide documentation for the *Annual Miles* variable in the *Car Cost* model.



1. Double-click the *Annual Miles* node to open its **Object** window.



Tip You can also open a variable's **Object** window by double-clicking the node using the browse or arrow tools. If you are using the browse tool, you will not be able to enter or change documentation.

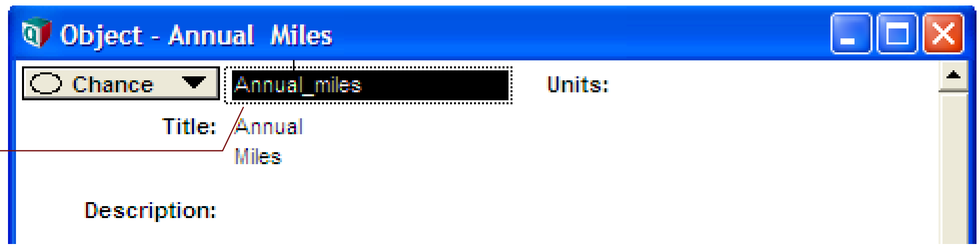


The identifier in the **Object** window shown above is **Annual_miles**. Analytica assigns the identifier when the title is created. It uses the first 20 characters of the title except for spaces or punctuation, which are replaced by underscores (_). Analytica does not differentiate between uppercase and lowercase letters in identifiers.

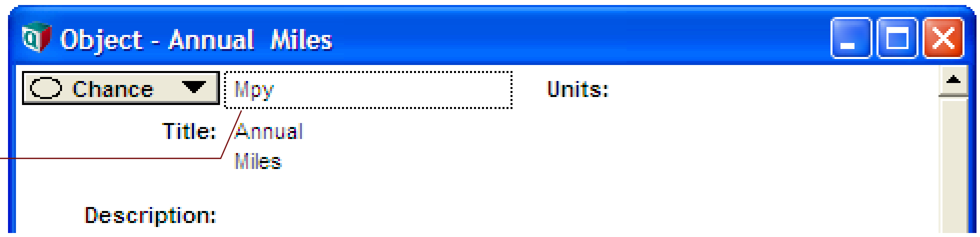
You can directly edit both the identifier and the title.

First, you will change the variable's identifier to a short abbreviation so that it can easily be used later in the definitions of other variables. You will then document the variable more fully.

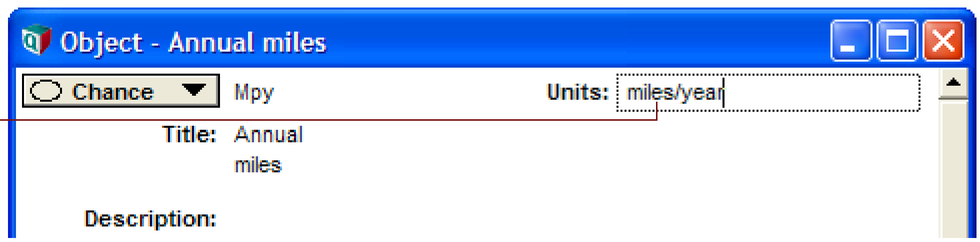
2. Select the contents of the Identifier field.



3. Type ***Mpy*** and press *Alt+Enter*.

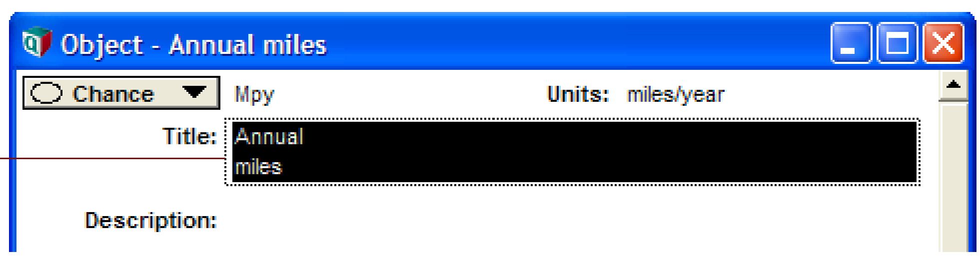


4. Click the Units field. Type ***miles/ year*** and press *Alt+Enter*.

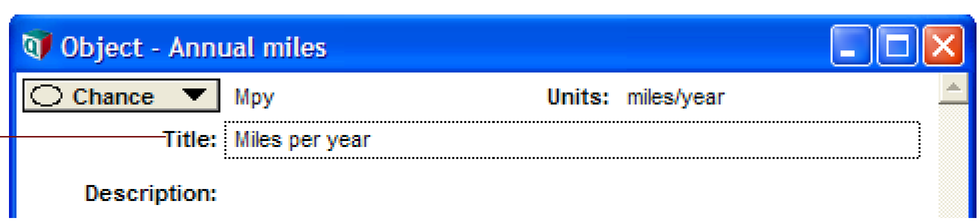


Tip Analytica uses the information from the Units field to label graphs or tables that use the *Miles per year* variable; Analytica does not use it in any mathematical computations.

5. Select the contents of the Title field and press the *Delete* key to erase **Annual miles**.

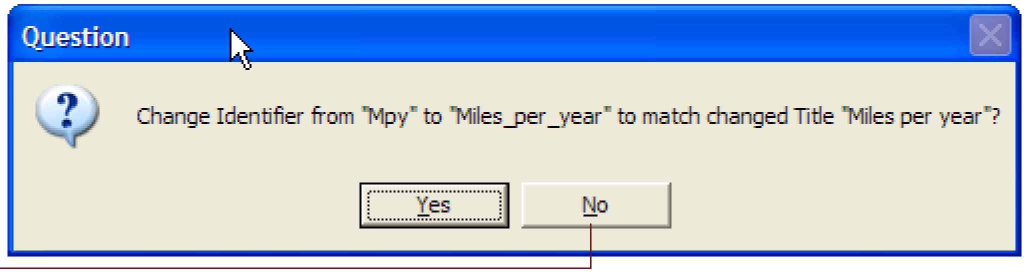


6. Type ***Miles per year*** and press *Alt+Enter*.

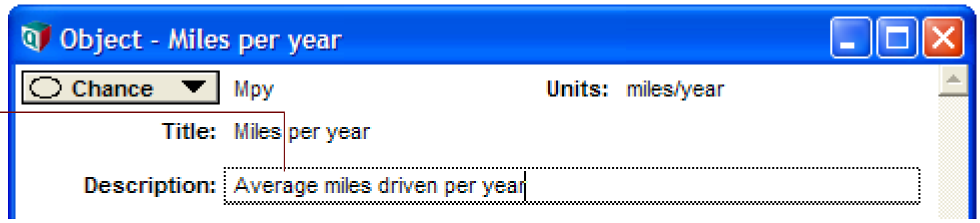


Tip When you change the title of a node, Analytica asks you if you want it to automatically change the identifier to match the new title.

7. Click **No** to keep the identifier as **Mpy**.



8. Click in the Description field, type ***Average miles driven per year***, and press *Alt+Enter*.



Defining variables explicitly

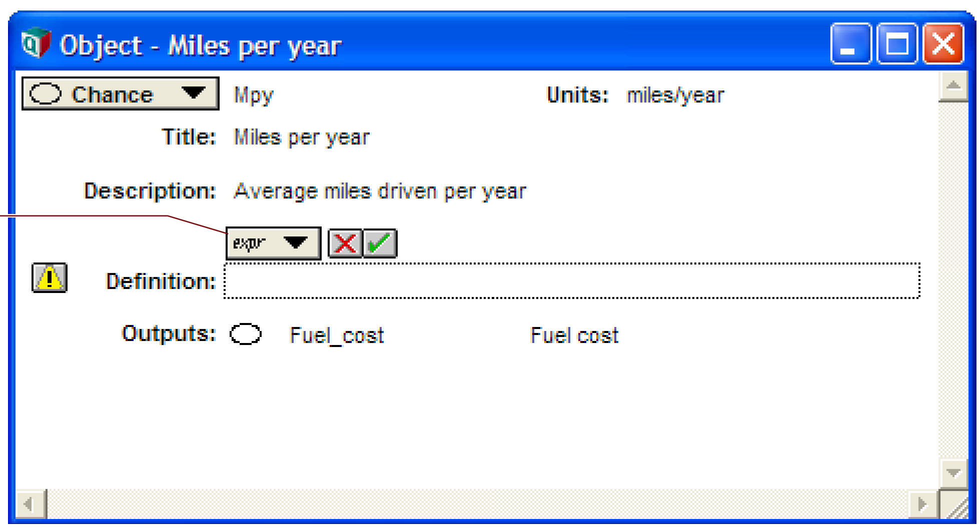
When you create and document a variable, you must also ***define*** the ***mathematical expression*** for computing its value.

In this section, you will provide a mathematical expression for the *Miles per year* variable. You will define *Miles per year* as a single number.

For this example, assume that the average number of miles driven annually is 12,000.

1. Click in the Definition field to enter a mathematical expression for the variable; type ***12K***.

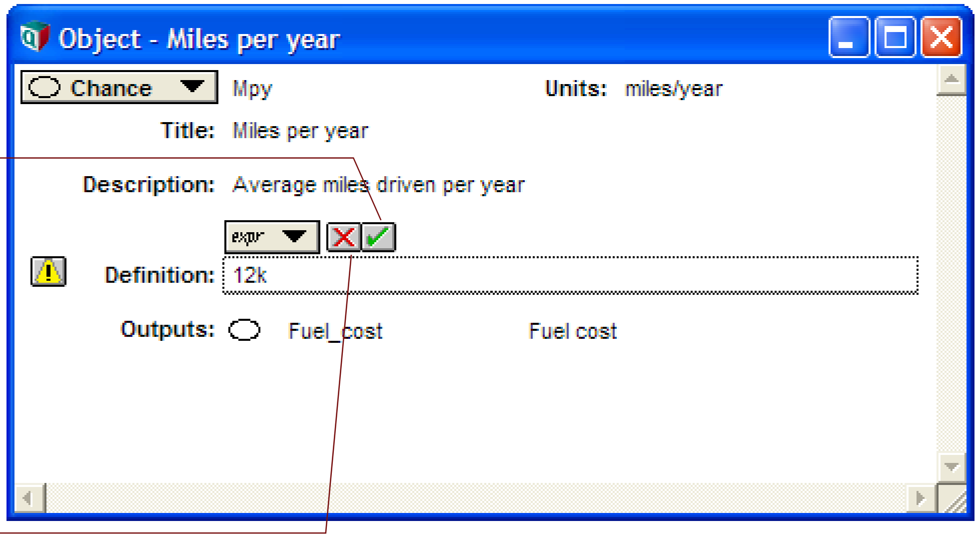
   A warning icon ⚠ indicates that this variable's definition has not yet been accepted.



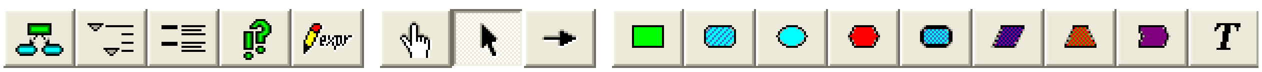
2. Click the check button or press *Alt+Enter* to accept the new definition.

   The warning icon disappears because the variable now has a valid definition.

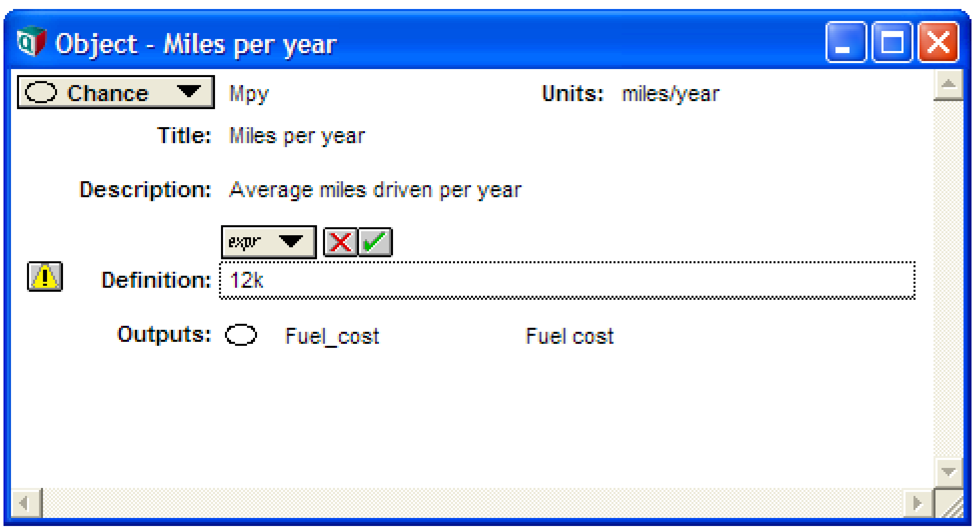
   If you want to cancel what you entered, click the cancel button.



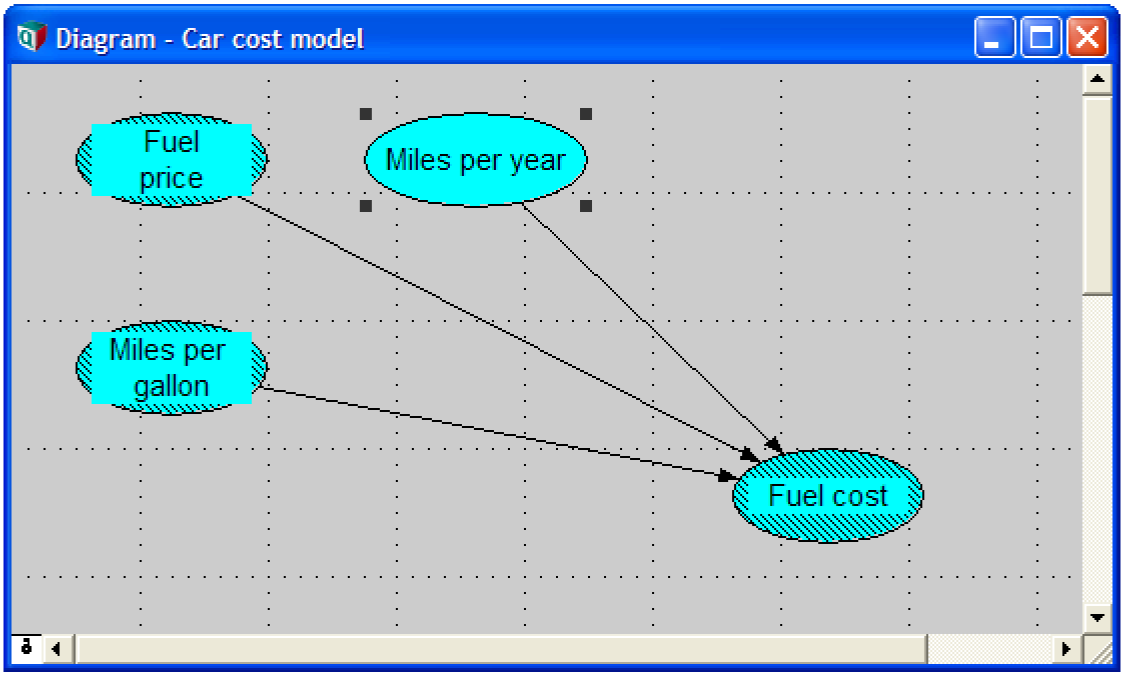
Whenever another variable's definition includes the ***identifier*** (*Mpy*) of *Miles per year*, this defined value, **12K**, is used as its value.



3. Click the **Diagram** button to return to the influence diagram.



Miles per year is no longer filled with a diagonal line pattern around its title, shown in the figure below. The clear node indicates that *Miles per year* now has a valid definition.

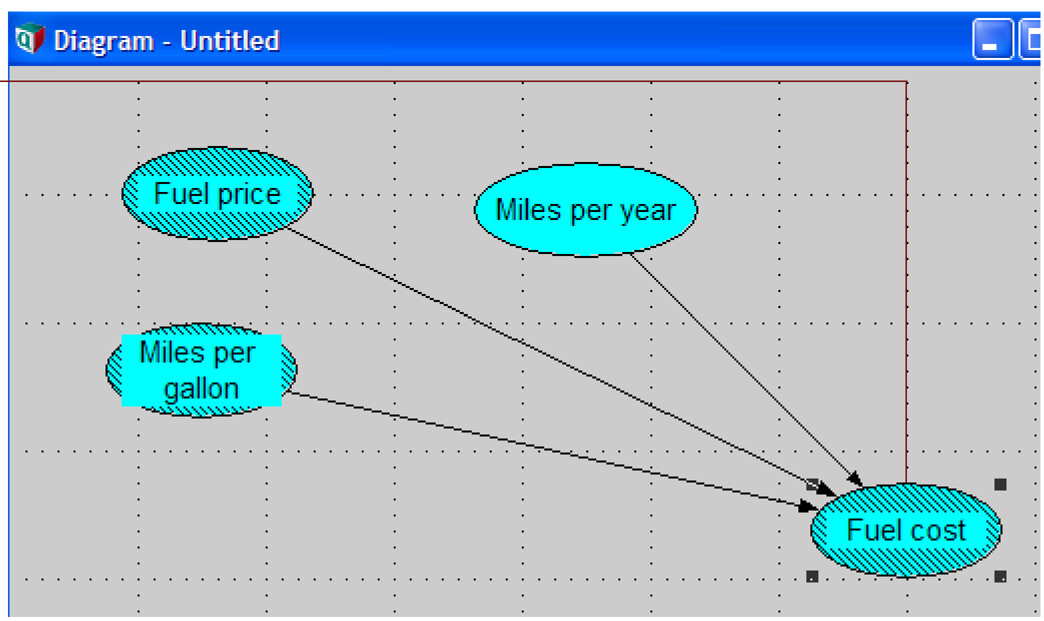


Defining variables that are influenced by other variables

When one variable is influenced by (dependent upon) another variable, you must provide an expression that describes the relationship between the variables. The *Fuel cost* node has arrows entering it from three other variables.

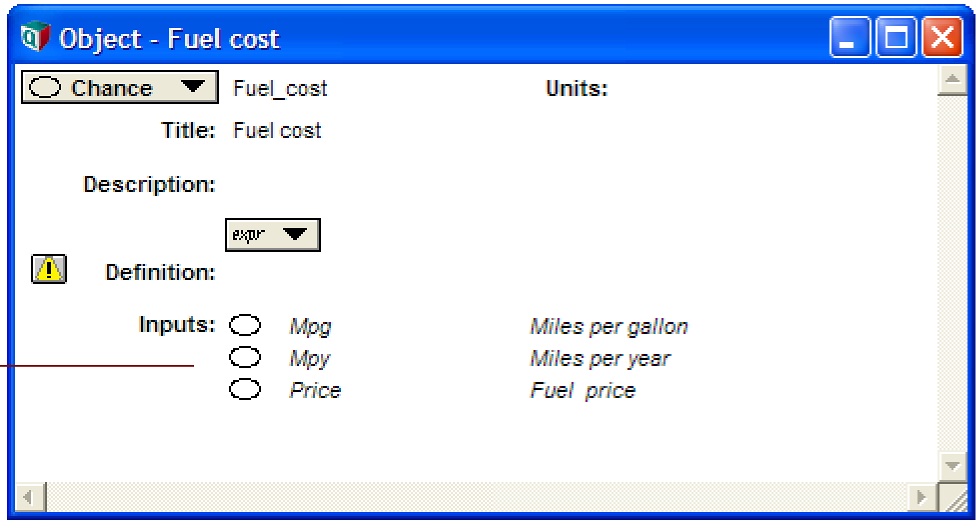
In this section, you will enter a definition for *Fuel cost* that describes how it depends on the *Miles per year*, *Miles per gallon*, and *Fuel price* variables.

1. Double-click the *Fuel cost* node to open its **Object** window.



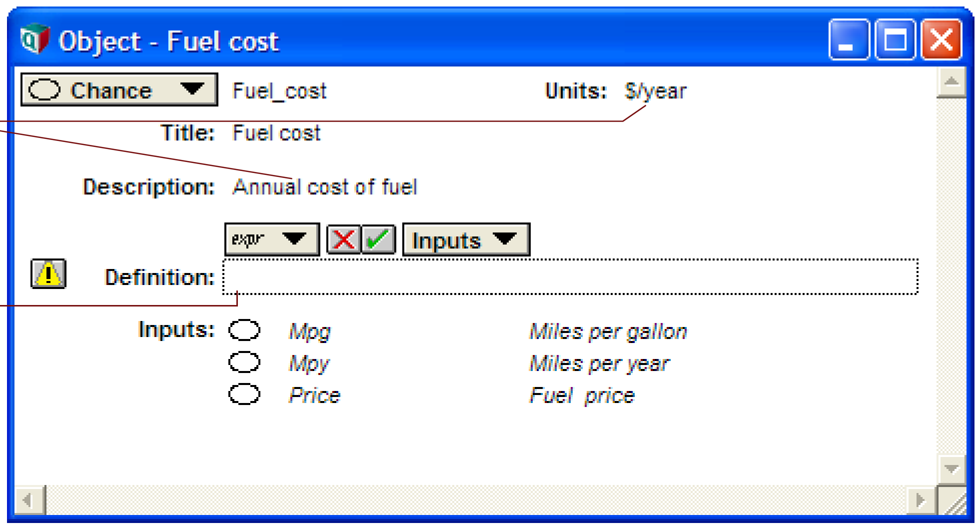
2. The identifiers and titles of the three input variables appear in the Inputs field for the *Fuel cost* variable.

   Because the form of the dependence has not been specified, the Definition field is blank.



3. Enter the variable's units as ***$/year***, and description as ***Annual cost of fuel***.

4. Click in the Definition field to enter a mathematical expression.



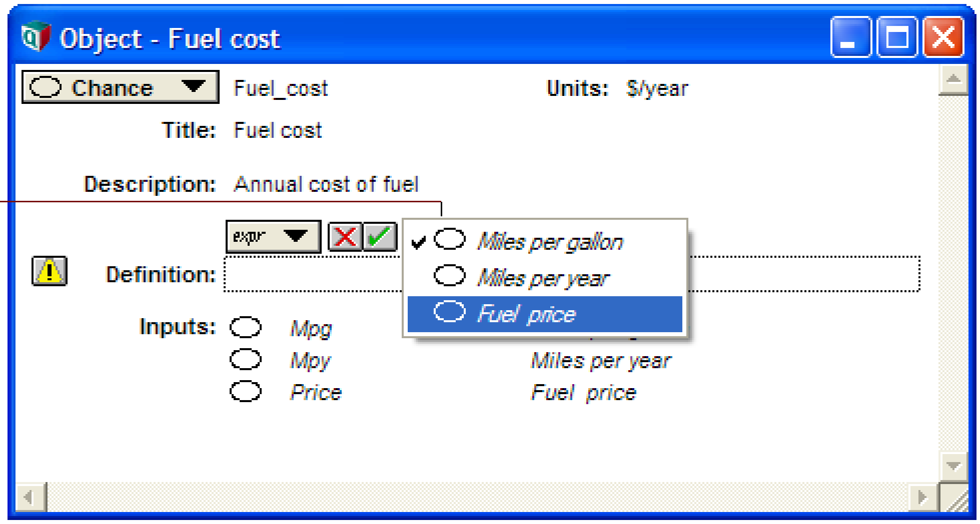
Because Fuel cost is equal to fuel price times miles driven, divided by miles per gallon, you will enter the following expression into the Definition field:

```
Fuel_price * Mpy/Mpg
```

An asterisk (`*`) represents multiplication; a slash (`/`) represents division.

5. From the Inputs popup menu select the name of the variable that you want to add, in this case, *Fuel price*. *Fuel_price* appears in the Definition field.

6. Type an asterisk (*).

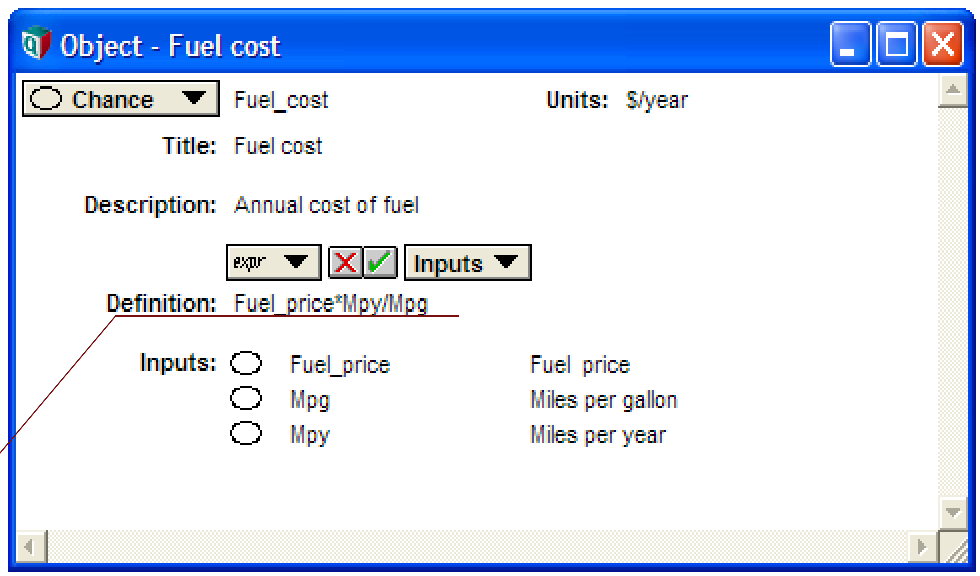


7. Select *Miles per year* from the Inputs popup menu.

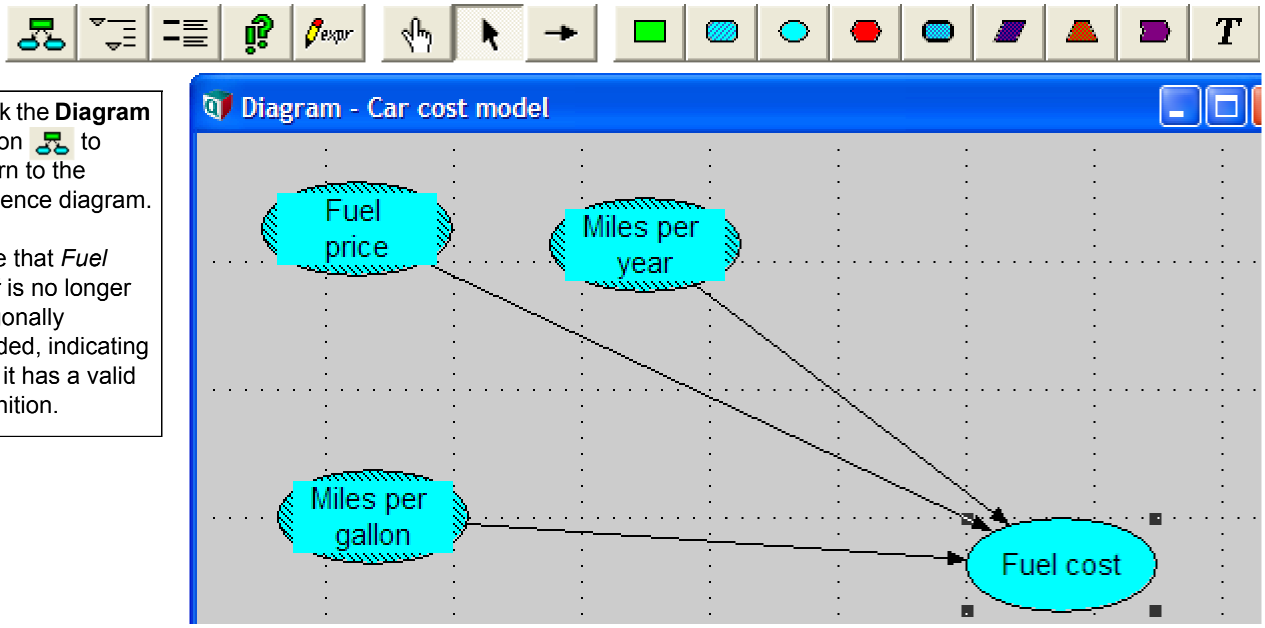
8. Type a slash (/).

9. Select *Miles per gallon* from the Inputs popup menu.

10. Press *Alt+Enter* or click the check button to accept the definition. The Definition field should look like this (whitespace between terms and operators is optional).



Based on the definition you just entered, the value of *Fuel cost* is calculated by multiplying the values of *Fuel price* and *Miles per year*, and then dividing the result by the value of *Miles per gallon*.



11. Click the **Diagram** button to return to the influence diagram.

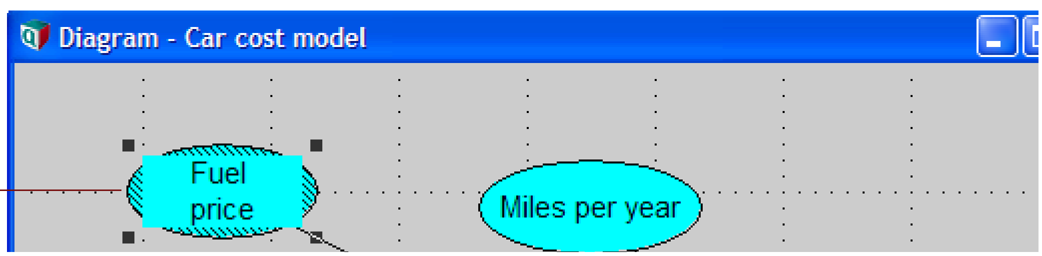
Note that *Fuel cost* is no longer diagonally shaded, indicating that it has a valid definition.

Probabilistic definition

In this section, you will document the variable *Fuel price* and provide it with a ***probabilistic*** definition. You will assume that the exact price of gasoline is not known and that its uncertainty is best expressed as a normal distribution. You will enter this definition and view the distribution.

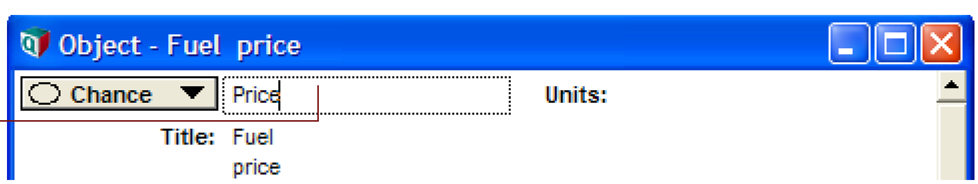
Tip Uncertain values can be expressed using any of a wide range of probability distributions. Consult a probability and statistics text book for information on the characteristics and uses of different distributions.

1. Double-click the *Fuel price* node to open its **Object** window.

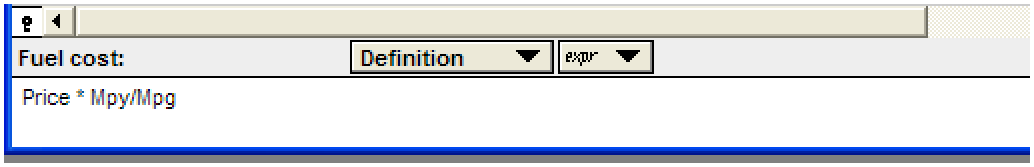


A new variable automatically receives a default identifier based on the first 20 characters of the variable's title. You will now change this identifier.

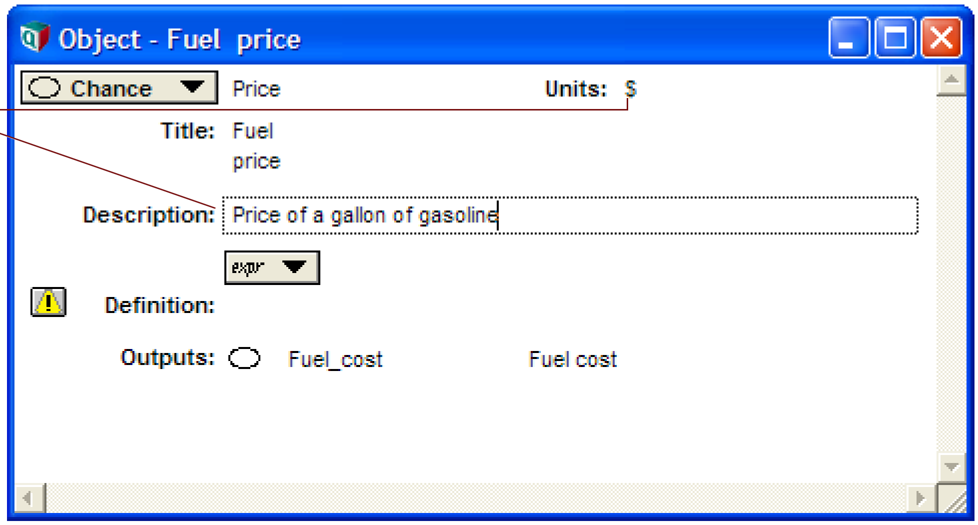
2. Change the identifier to ***Price***.



Tip As shown in the section "Defining variables that are influenced by other variables" beginning on page 79, the definition of the *Fuel cost* variable refers to the *Fuel price* variable. Because you just changed the identifier of *Fuel price*, the definition of *Fuel cost* is automatically updated to refer to the new identifier.



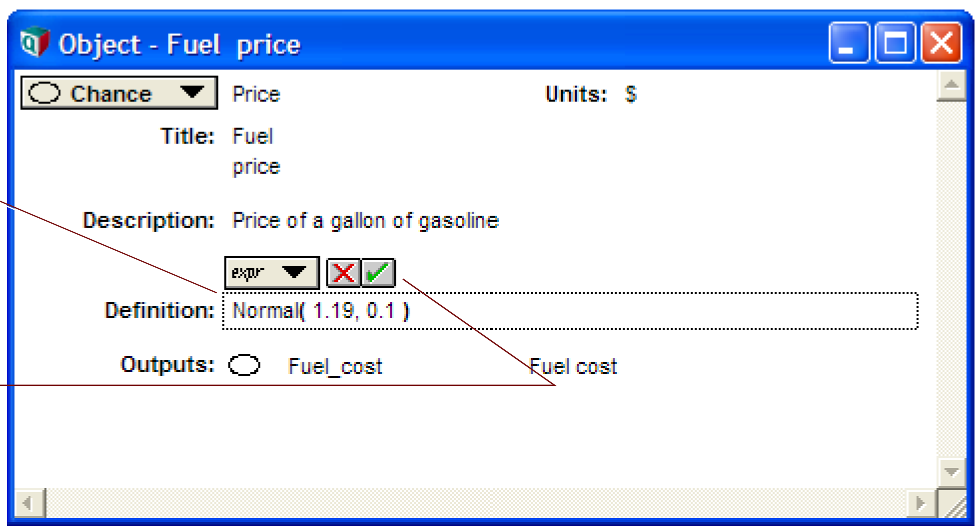
3. Enter the units as ***$***, and the description as ***Price of a gallon of gasoline***, as shown.



Now you will define *Fuel price* as a normal distribution with a mean of $1.19/gallon and a standard deviation of $0.10. You will enter the uncertainty distribution directly as an expression, rather than using the **Object Finder**.

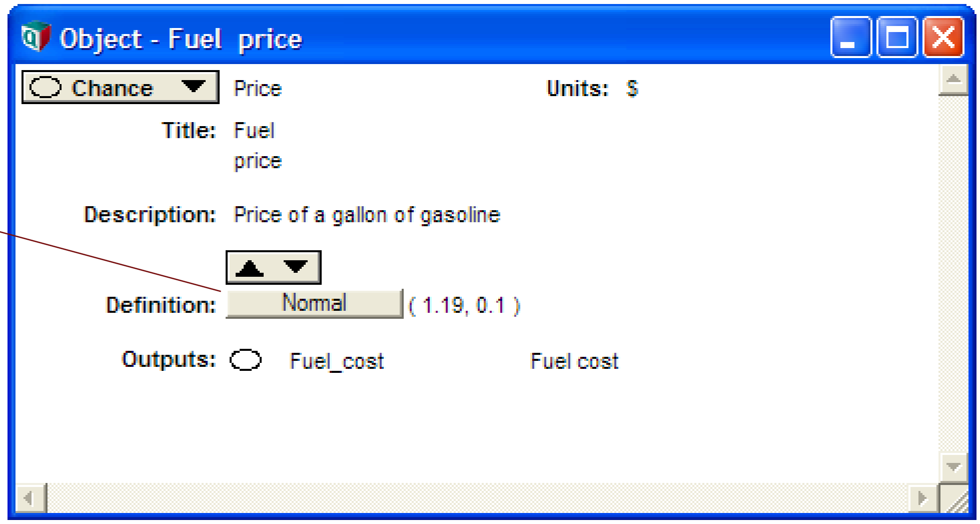
4. Enter the definition for the normal distribution as ***Normal(1.19, 0.1)***.

5. Click the check button or press *Alt+Enter*.



The definition field now contains a button, indicating the type of distribution you just entered, **Normal**.

6. Click the button to see the distribution in the **Object Finder**.



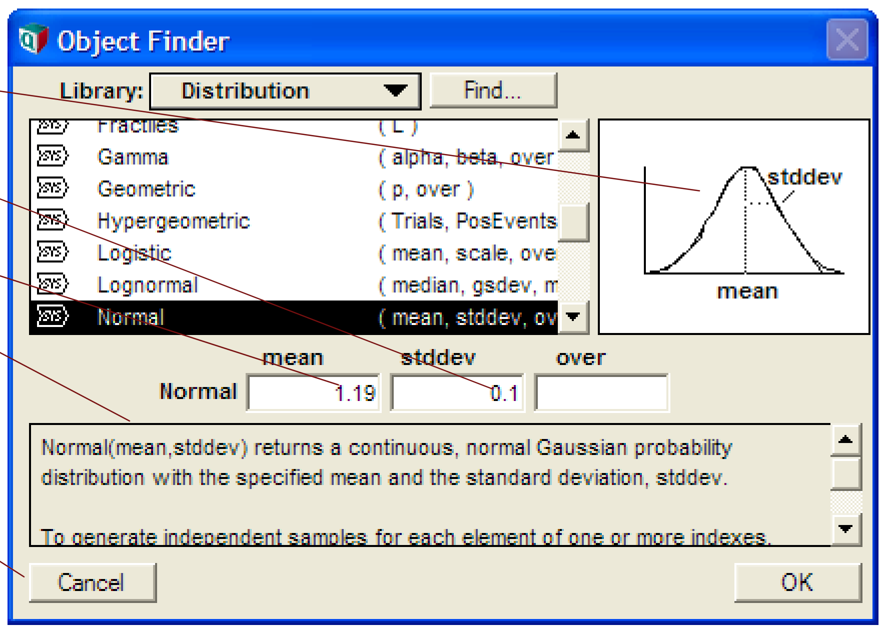
The **Object Finder** dialog box displays the normal distribution along with a graphic, the parameters you specified, and a description of the distribution.

Graphic of distribution

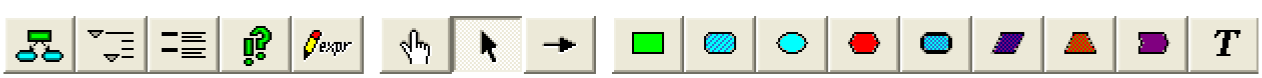
Standard deviation

Mean

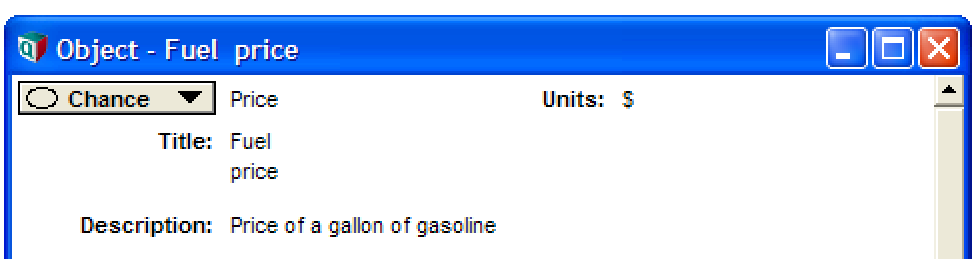
Description



7. Click the **Cancel** button to close this dialog box.



8. Click the **Diagram** button to return to the influence diagram window.
Note that *Fuel price* is no longer diagonally shaded.

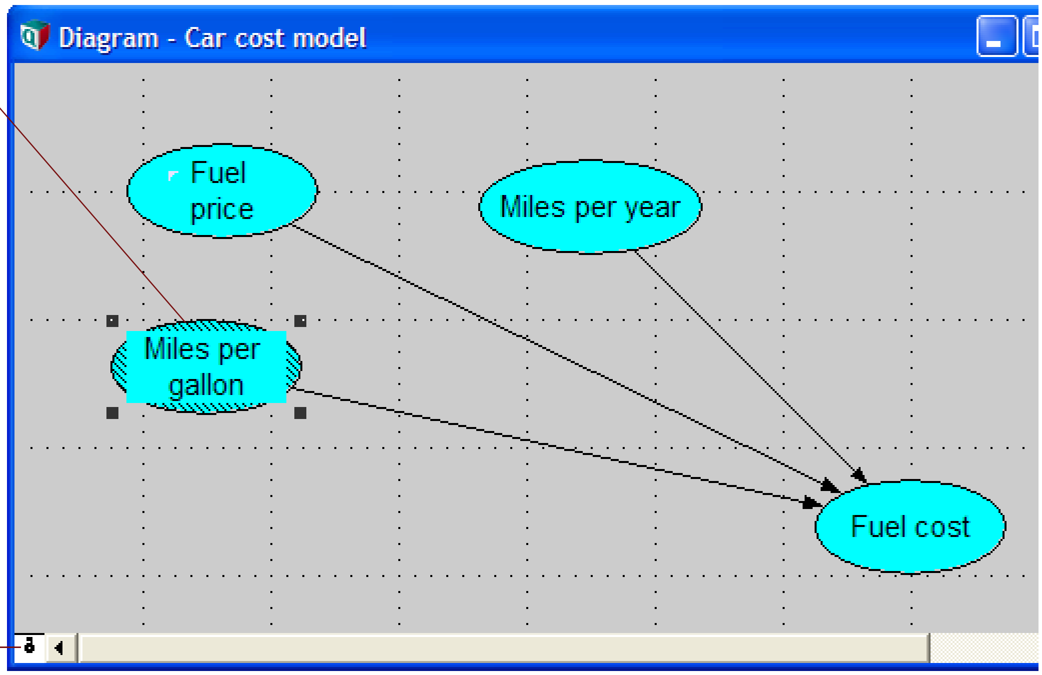


Entering attributes using the Attribute panel

Rather than opening a separate window to alter a variable's attributes, you might prefer to see a variable's attributes in the same window as the model influence diagram. The **Attribute** panel, which appears under the diagram, allows you to edit as well as examine attributes.

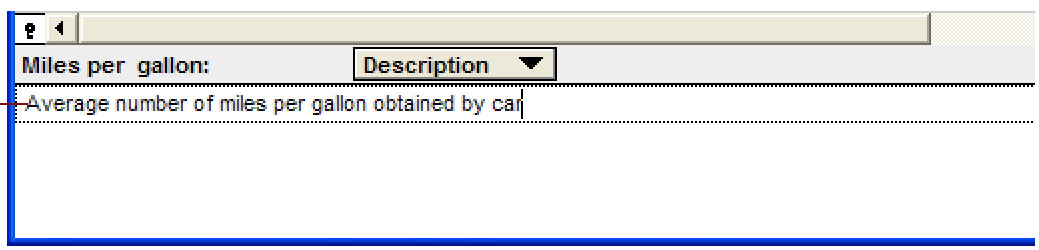
In this section, you will enter data for the *Miles per gallon* variable in the **Attribute** panel.

1. Select the *Miles per gallon* node.
2. Click the key icon to display the **Attribute** panel.

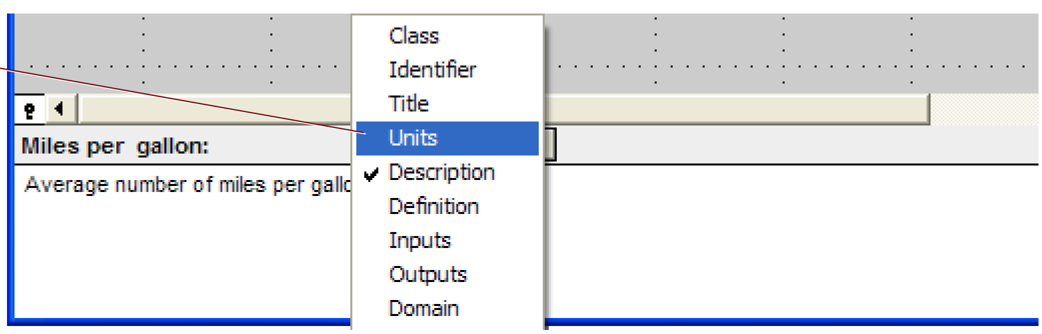


The **Attribute** panel appears below the diagram. You can use this view to enter or edit data for the currently selected variable in the influence diagram.

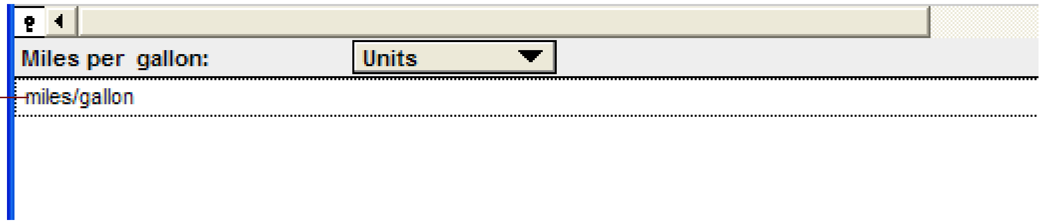
3. Click in the **Attribute** panel to enter the description for *Miles per gallon*. Type ***Average number of miles per gallon obtained by car***, then press *Alt-Enter*.



4. Select **Units** from the Attributes popup menu.



5. Type ***miles/gallon***, then press *Alt+Enter*.



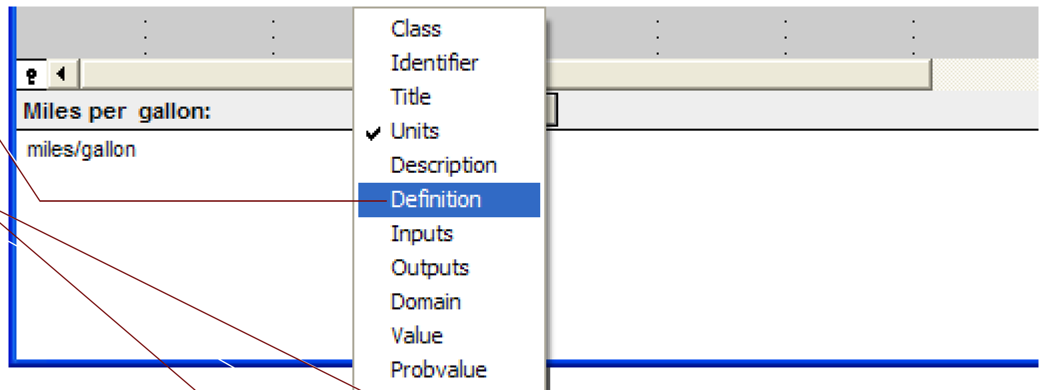
Now that you have entered the documentation for variable *Miles per gallon*, you will enter its definition in the **Attribute** panel.

You will define the value of *Miles per gallon* as an uncertain value that is equally likely to be any value between 20 and 30. The uniform distribution describes an equally likely uncertainty.

You will define the variable as a distribution using the Expression popup menu.

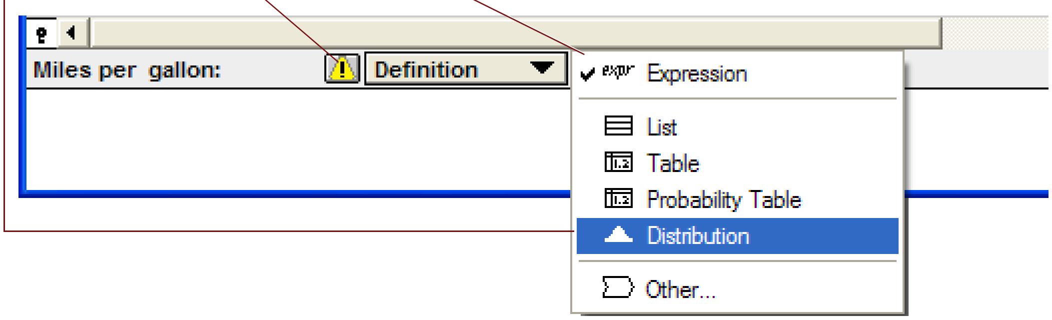
1. Select **Definition** from the Attributes popup menu.

   The warning icon and the Expression popup menu appear above the Definition field.

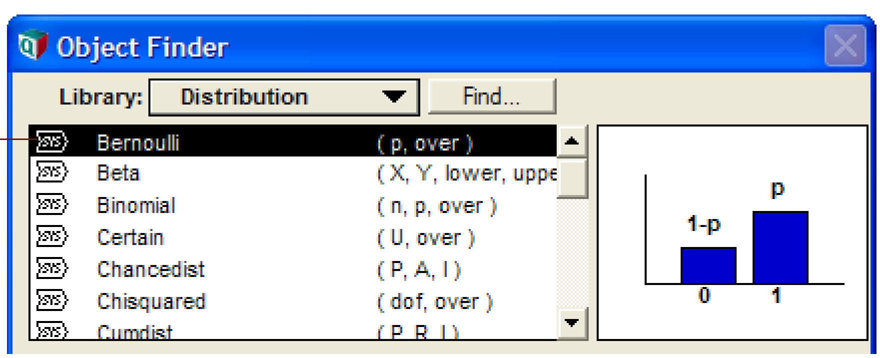


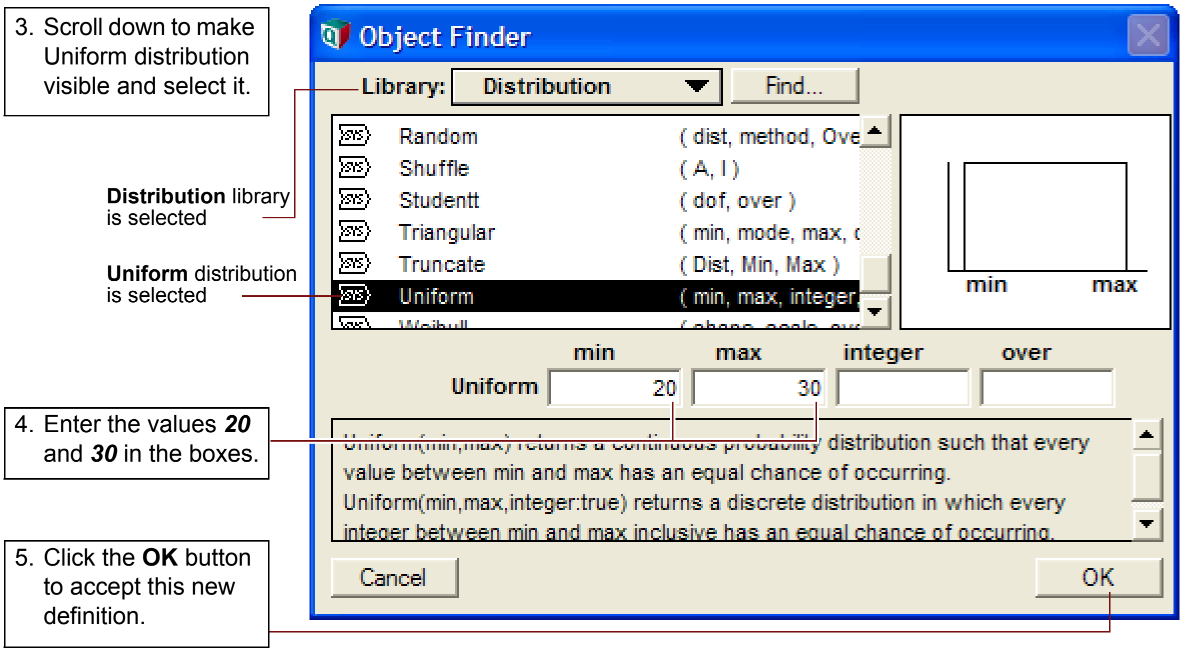
2. Select **Distribution** from the Expression popup menu.

   The **Object Finder** dialog box displays the Distribution library, with the first distribution (Bernoulli) selected. See the figure on the following page.

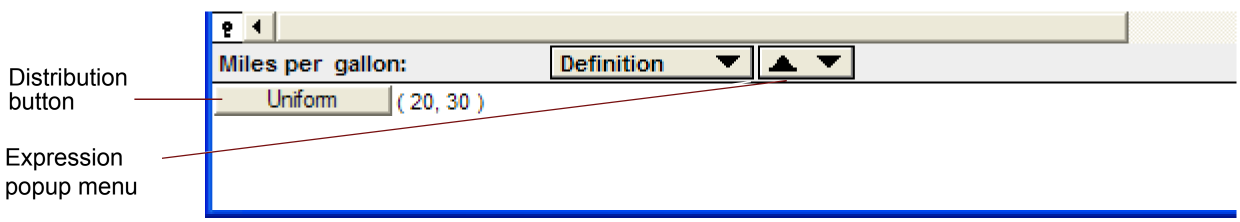


Bernoulli distribution is selected





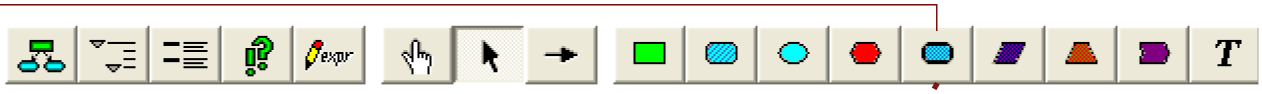
The **Attribute** panel updates as shown in the following figure. The distribution icon  in the Expression popup menu indicates that a probability distribution is selected; the button is labeled with the name of the distribution.



Creating modules

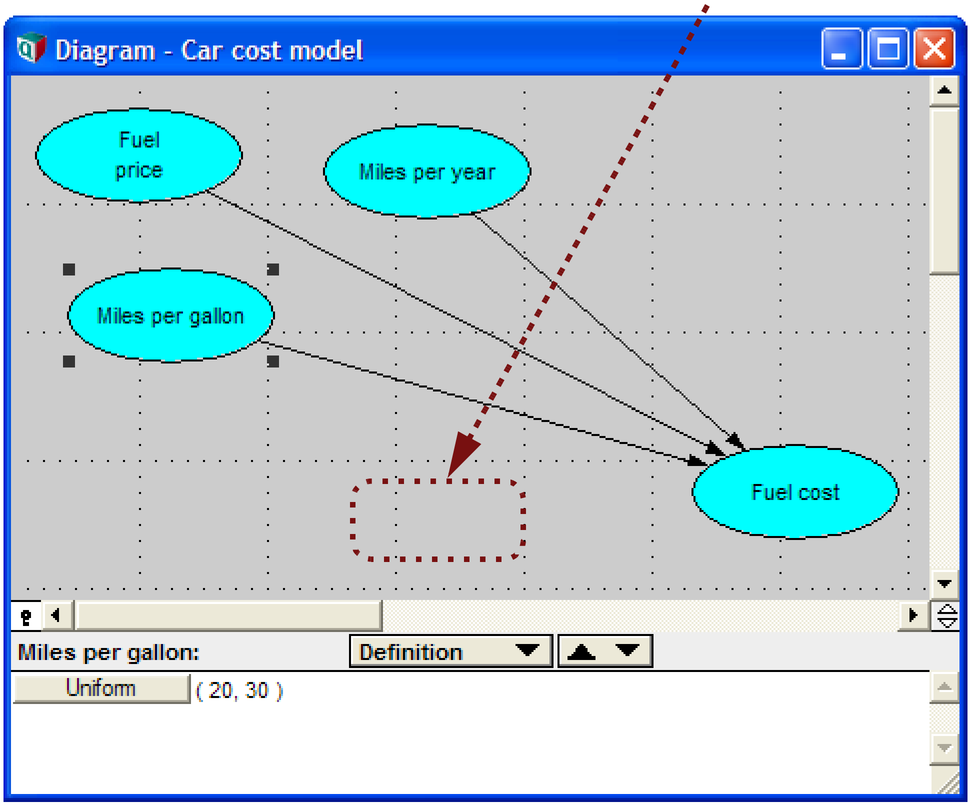
To simplify complicated diagrams, most complex models use submodels, called modules. A ***module*** is an influence diagram containing variables and their relationships to one another.

In this section, you will create a module that contains the fuel cost variables. First, make sure the edit tool  is selected.

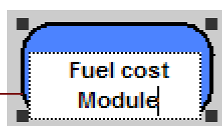


1. Drag the **Module** icon to the position shown by the arrow.

   Notice that a module node has thick outlines.

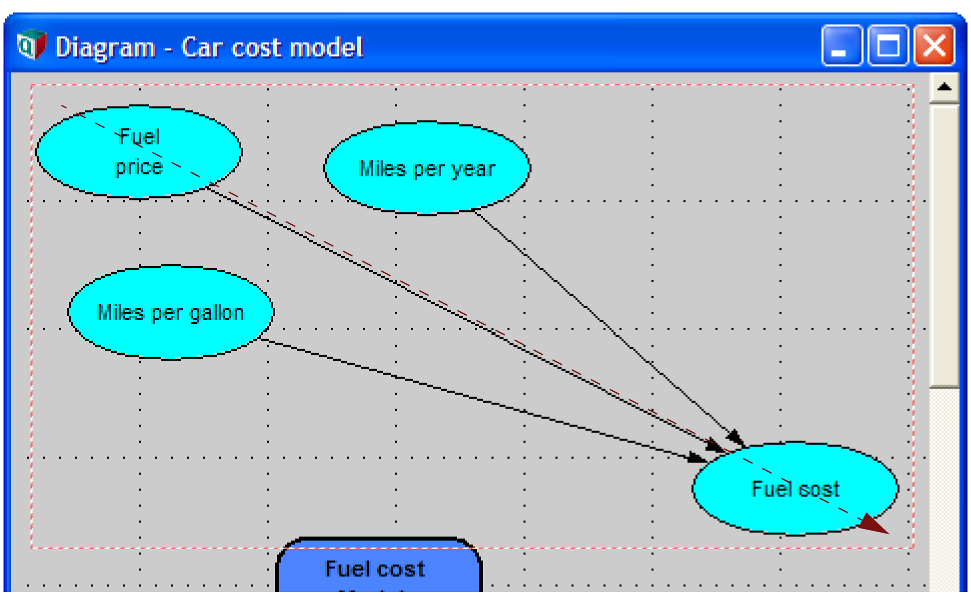


2. Title the node ***Fuel cost Module*** and press *Alt+Enter*.



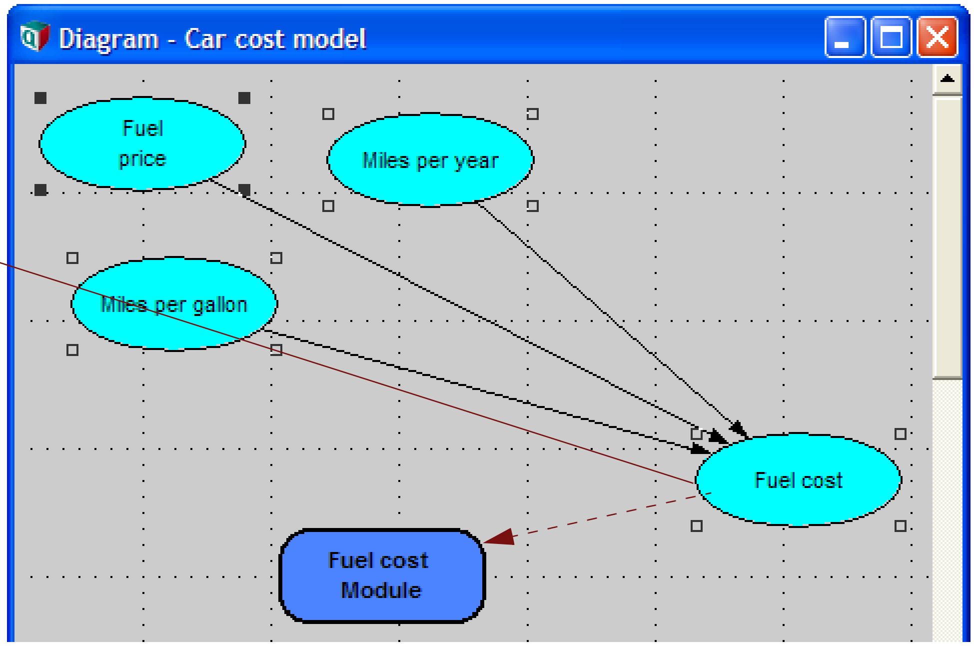
3. Select all of the nodes (*Fuel price, Miles per year, Miles per gallon,* and *Fuel cost*) except for the *Fuel cost Module*.

   To select all of the nodes, use rubber banding to create a selection box surrounding the nodes.

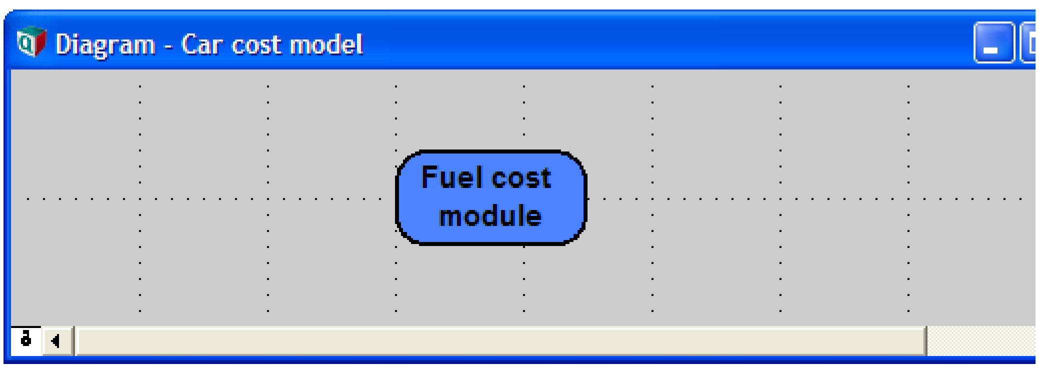


4. Drag from any of the selected variables onto the *Fuel cost Module* node; this node becomes highlighted when the mouse is over it.

5. Release the mouse button.



Your diagram should now look like the following.



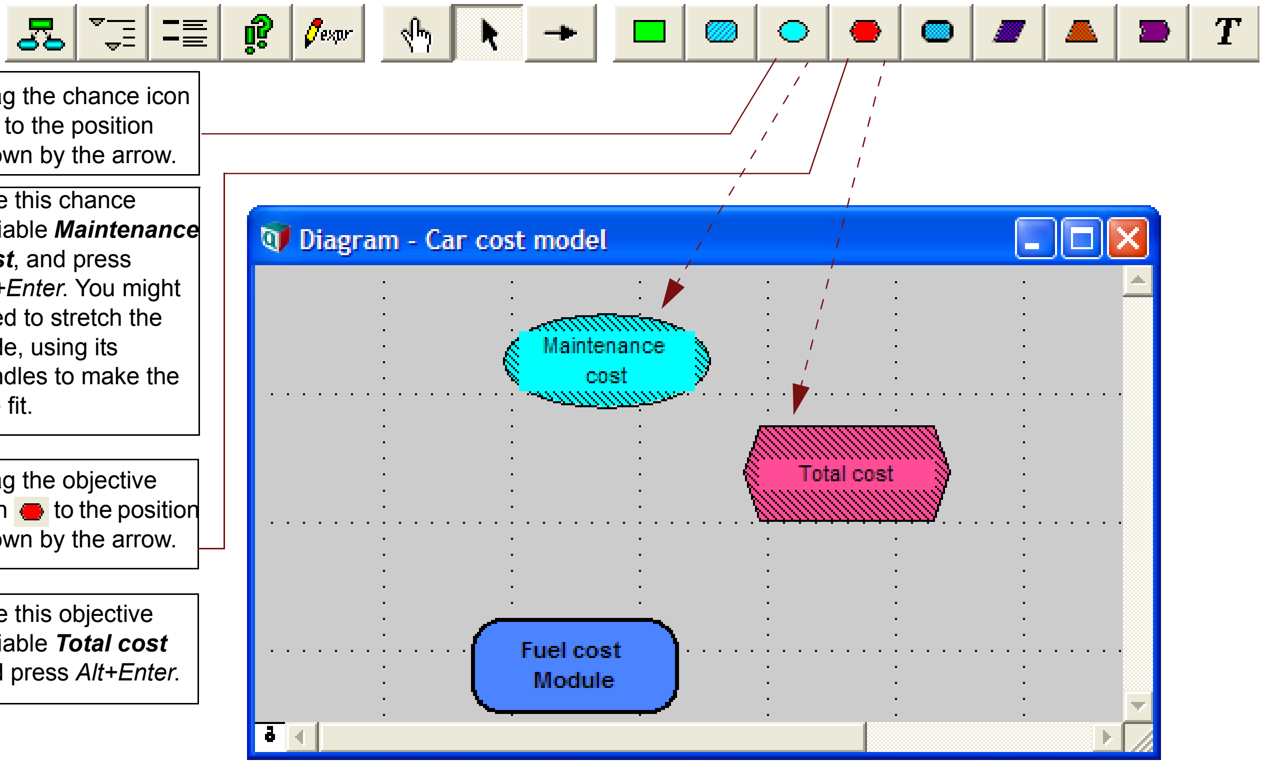
Tip If you left any nodes behind, simply drag them onto the *Fuel cost module* node now.

Drawing arrows between variables in different modules

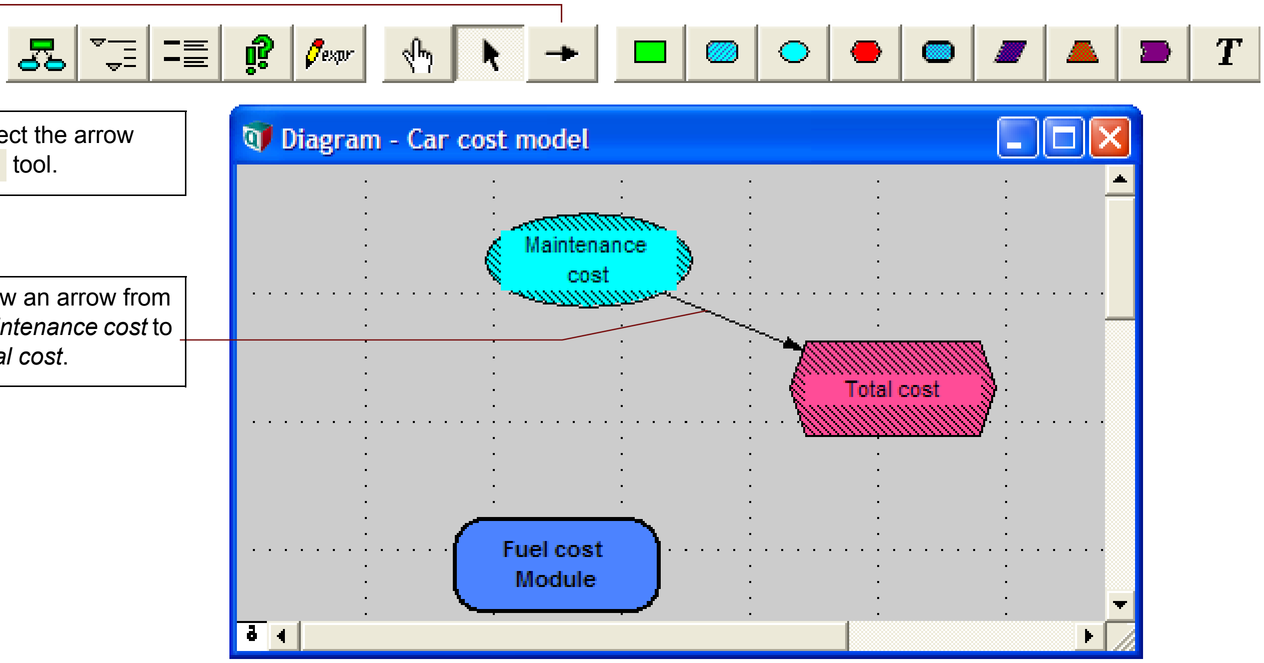
In addition to drawing arrows between variables in a single model or module, you can create dependencies between variables in different modules using the arrow tool.

In this section, you will add more variables to the top-level model and connect them to variables in the module that you just created.

First, you will add a *Maintenance cost* chance variable, and define a *Total cost* objective variable. The *Total cost* variable is used to compute the sum of the *Maintenance cost* and *Fuel cost* variables.



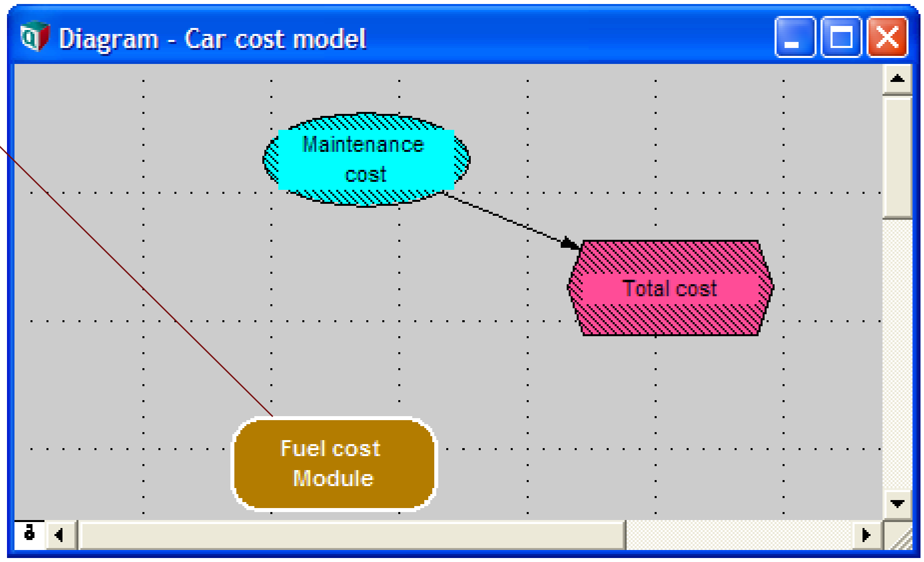
1. Drag the chance icon to the position shown by the arrow.
2. Title this chance variable ***Maintenance cost***, and press *Alt+Enter*. You might need to stretch the node, using its handles to make the title fit.
3. Drag the objective icon to the position shown by the arrow.
4. Title this objective variable ***Total cost*** and press *Alt+Enter*.



5. Select the arrow tool.
6. Draw an arrow from *Maintenance cost* to *Total cost*.

You will now create an arrow between the *Fuel cost* variable (inside the *Fuel cost Module*) and the *Total cost* variable.

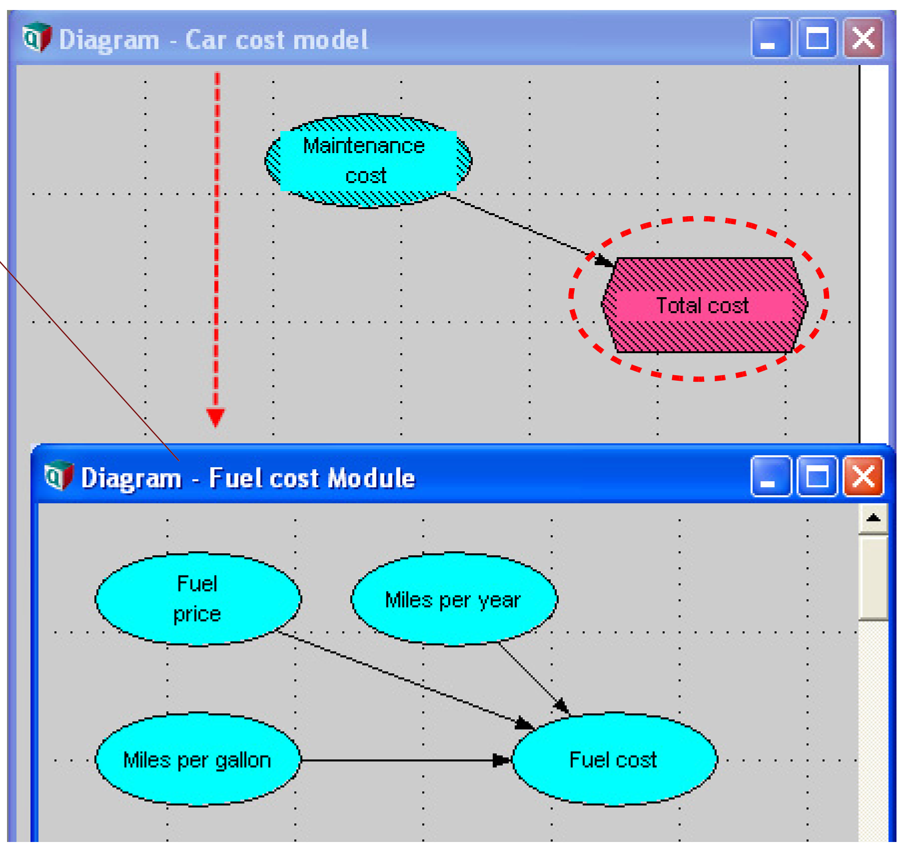
7. Double-click the *Fuel cost Module* (which contains the *Fuel cost* variable) to open it.



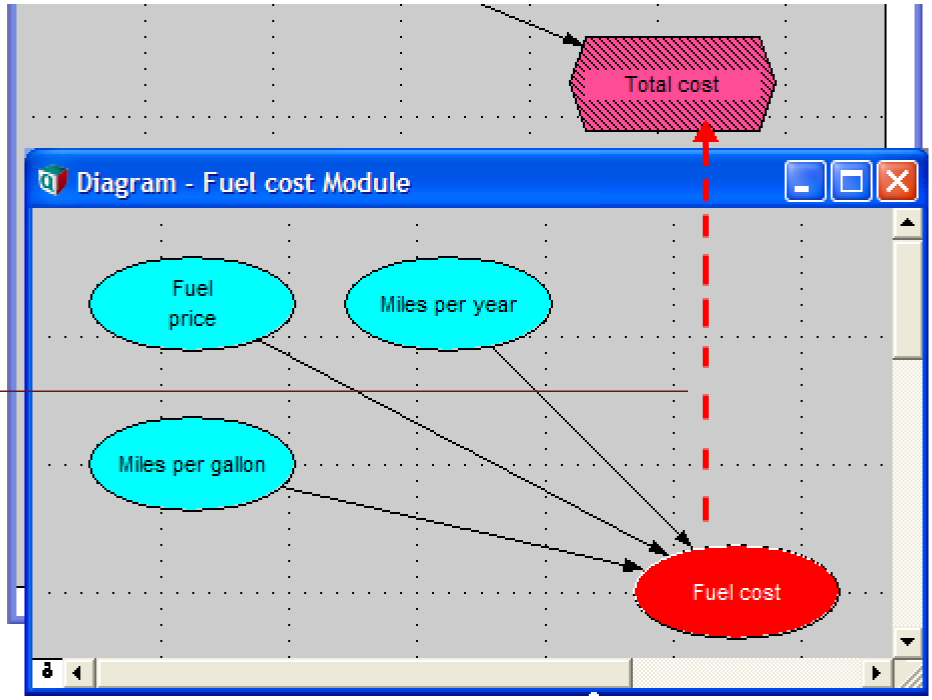
You can double-click a node to open it using any tool. Here, you are using the arrow tool.

The *Fuel cost Module* **Diagram** window is now in the foreground. You will move this window down on the screen to expose the *Total cost* node, which is located behind this window in the *Car cost model* **Diagram** window. Then you will draw an arrow from the *Fuel cost* node in the *Fuel cost Module* window to the *Total cost* node in the *Car cost model* window.

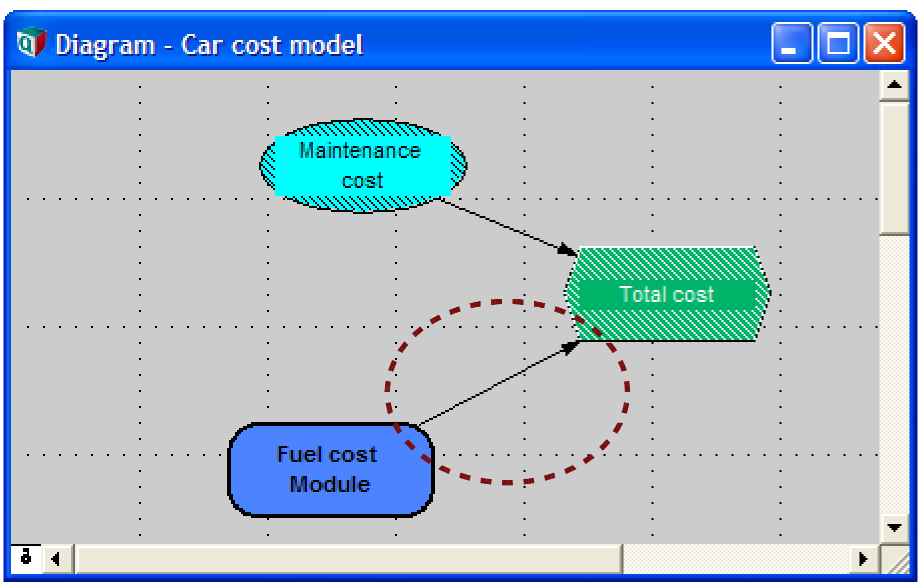
8. Press on the title bar of the *Fuel cost Module* **Diagram** window and move the window down so that you can see the *Total cost* node in the window behind it.



9. Drag from the *Fuel cost* node to *Total cost*.



An arrow is drawn between the *Fuel cost Module* and the *Total cost node*. This arrow does not represent the specific dependency you just created, only that *Total cost* depends on one or more variables in the *Fuel cost Module*.

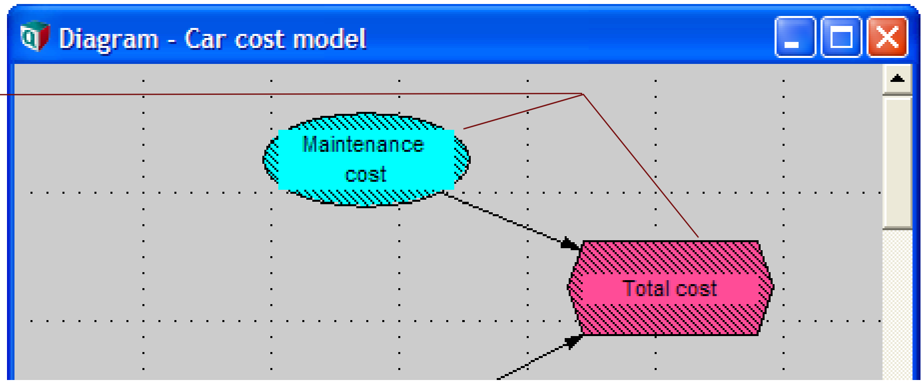


Completing the model

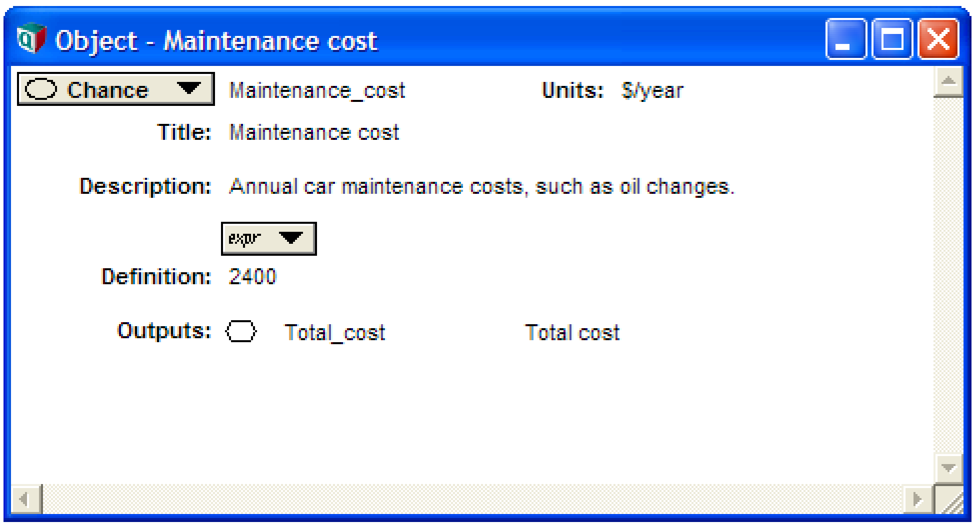
So far, you have used several methods for moving between windows, documenting variables, and specifying their definitions. In this last step of this chapter, it is up to you to complete the model on your own.

Note: *Select the edit tool and either use the* ***Attribute*** *panel at the bottom of the screen or double-click a node to open its* ***Object*** *window.*

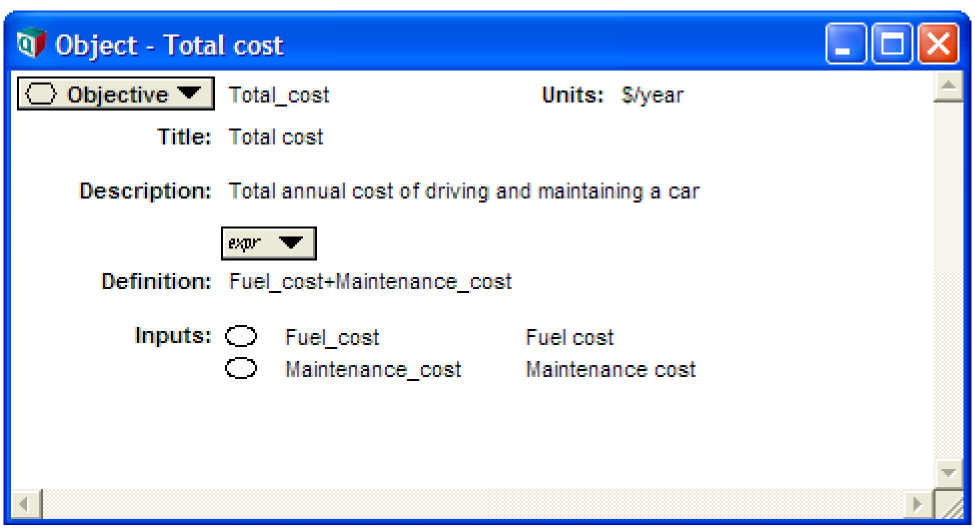
1. Document *Maintenance cost* and *Total cost* and specify their definitions using the information shown in the **Object** windows on the following page.



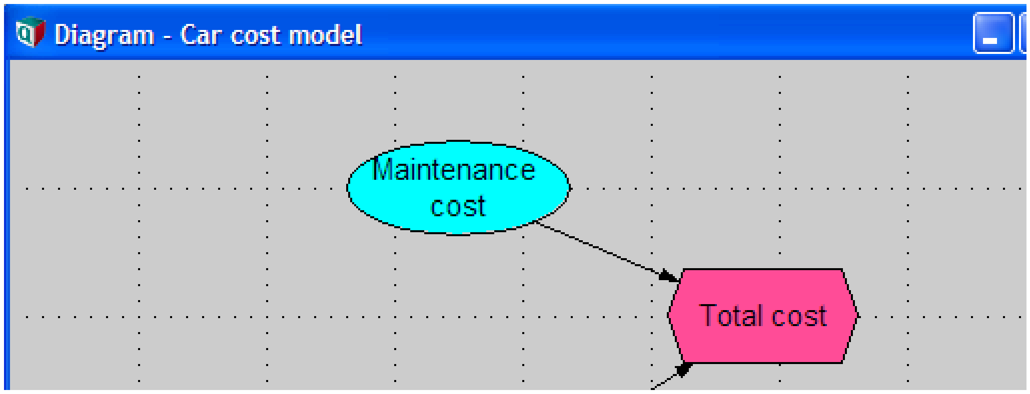
Enter these attributes for *Maintenance cost*:

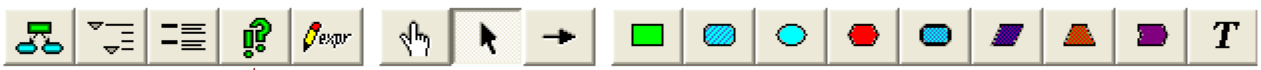


Enter these attributes for *Total cost*:



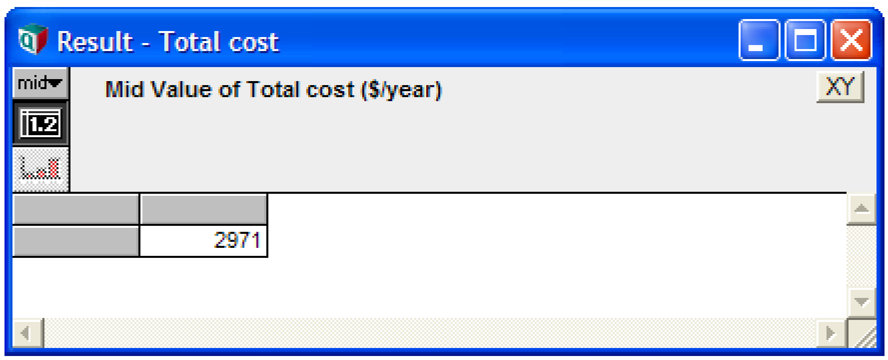
The nodes for *Maintenance cost* and *Total cost* now show that they are defined; they are no longer shaded with a diagonal line pattern.

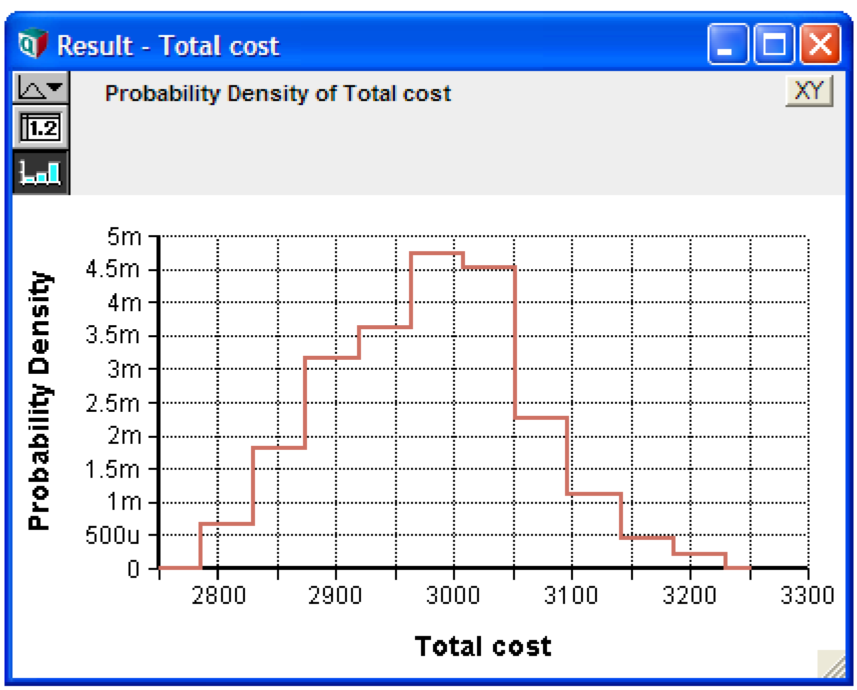




2. Click the **Result** button to evaluate *Total cost*.

Examine the mid value and probability density of *Total cost*.





Creating the Car Cost model: summary

In this chapter, you have:

- Created a model
- Documented and defined variables
- Created a module
- Drawn arrows between variables in the same module and in different modules

In the next chapter, you will extend the *Car Cost* model to include tables of data.

Saving your model and quitting

After you have created part or all of a model, you should save it. Because you previously saved your model, it is saved with the name you gave it.

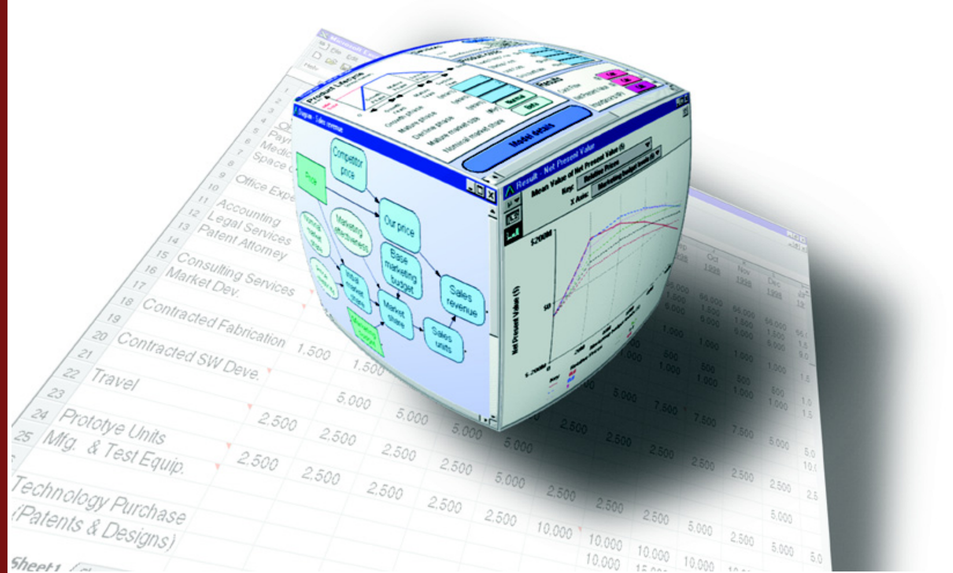
Close the **Attribute** panel by clicking the key icon .

You can quit Analytica at this point. See “Quitting Analytica” on page 23.

Chapter 5 Creating Arrays (Tables)

This chapter shows you how to:

- Define index variables
- Define a table and select its indexes
- Define other variables using the same indexes
- View the results of table calculations
- Combine results from a table using the **Sum()** function



Using the *Car Cost* model created in the previous chapter, you will create tables, edit their size and dimensions, and compute the results.

An ***array*** is a collection of values that can be viewed as one or more spreadsheet-like ***tables***. These terms can be used interchangeably for one- or two-dimensional arrays (tables). If an array has more than two dimensions, you can view it as an ordered collection of two-dimensional tables. With Analytica's Intelligent Arrays, you can define a variable as an array with as many as 15 dimensions. You can then use that variable in calculations as easily as if it had been defined as a single number.

The *Car Cost* model that you saved should be located in the *My Documents* folder on your computer. If you cannot find it, a copy of the *Car Cost* model is provided in the *Tutorial Models* folder, inside the Analytica folder on your computer (normally this is `C:\Program Files\Lumina\Analytica 4.1\Tutorial Models`, but you might have installed Analytica in a different folder).

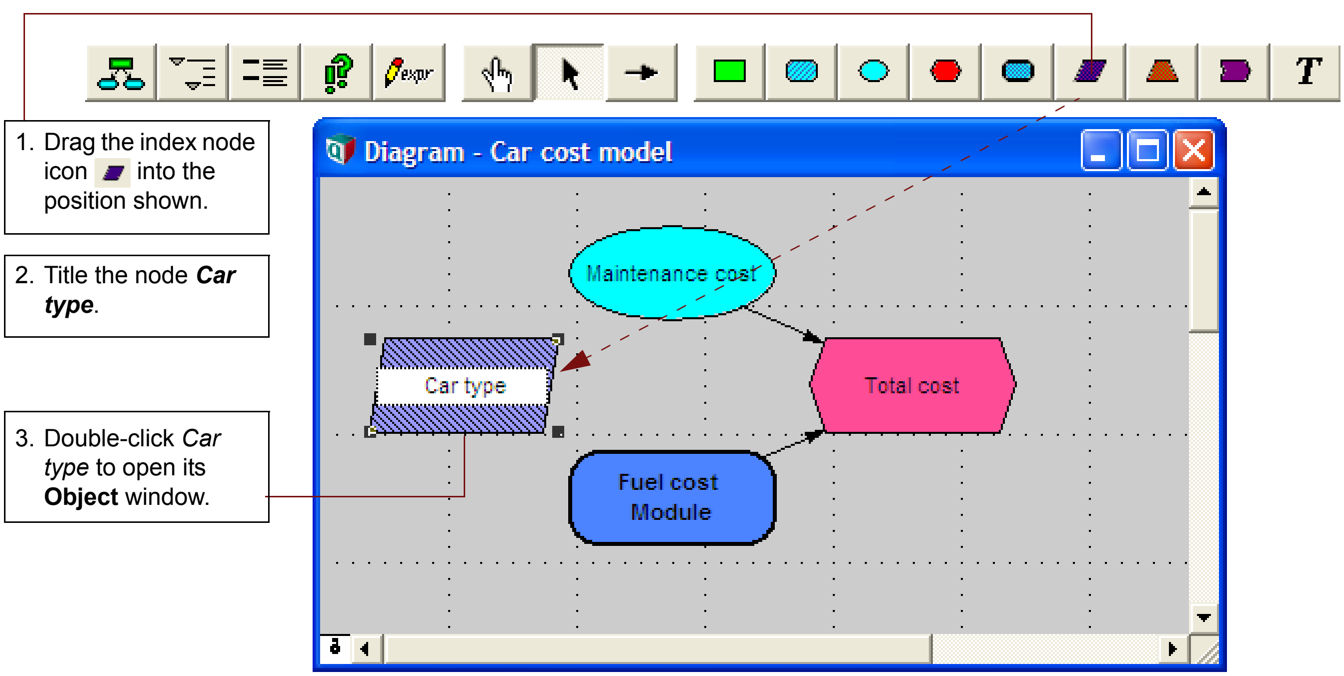
Creating index variables

Suppose you own two cars and wish to evaluate the annual costs of each one at the same time. Each car you own has a different value for *Miles per gallon* and *Maintenance cost*. Using ***tables*** and ***indexes***, you can assign miles per gallon and maintenance costs for each car type and calculate the total costs of each one as simply as calculating the total costs of one car.

In this section you will create and define an ***index variable***, *Car type*, to distinguish the different cars.

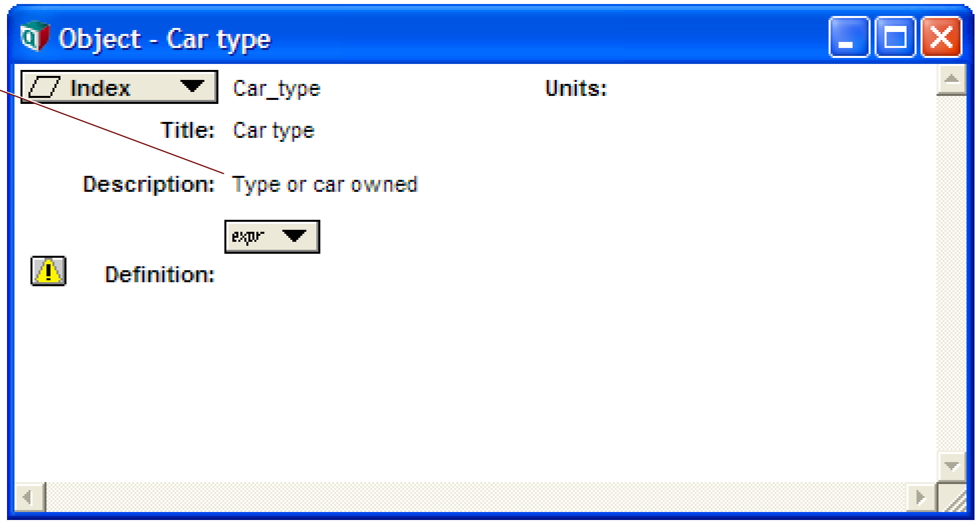
Index variables identify the dimensions of multidimensional variables; it's usually best to plan for and create the indexes before defining the arrays.

First, make sure the edit tool is selected.



Next, you will enter the documentation and definition for this index. *Car type* identifies two different cars by size: small and large.

4. In the Description field, type ***Type of car owned***, and press *Alt+Enter.*

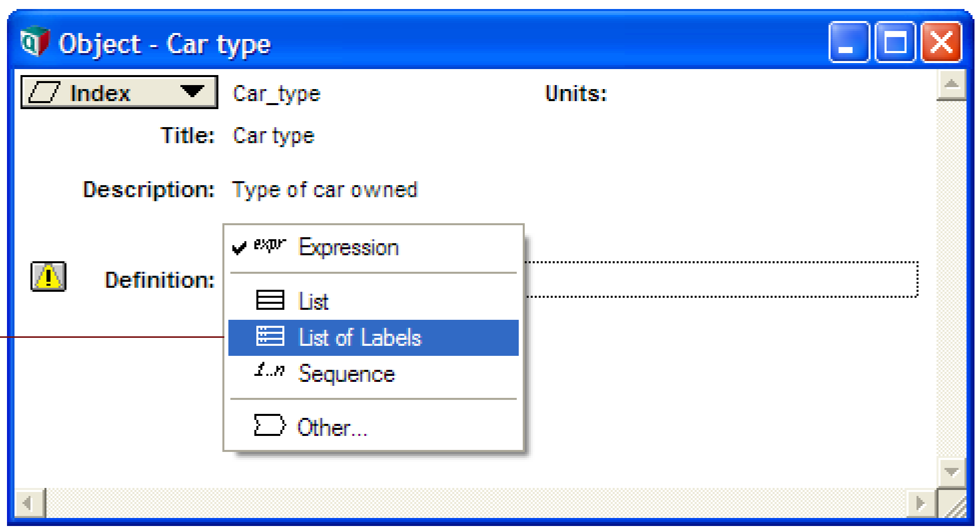


You will define *Car type* as a list of text labels identifying each car type.

Tip

A ***List*** differs from a ***List of Labels*** in that a ***List*** can contain numbers or expressions that compute to numbers, while a ***List of Labels*** only contains text, such as names, which are not evaluated.

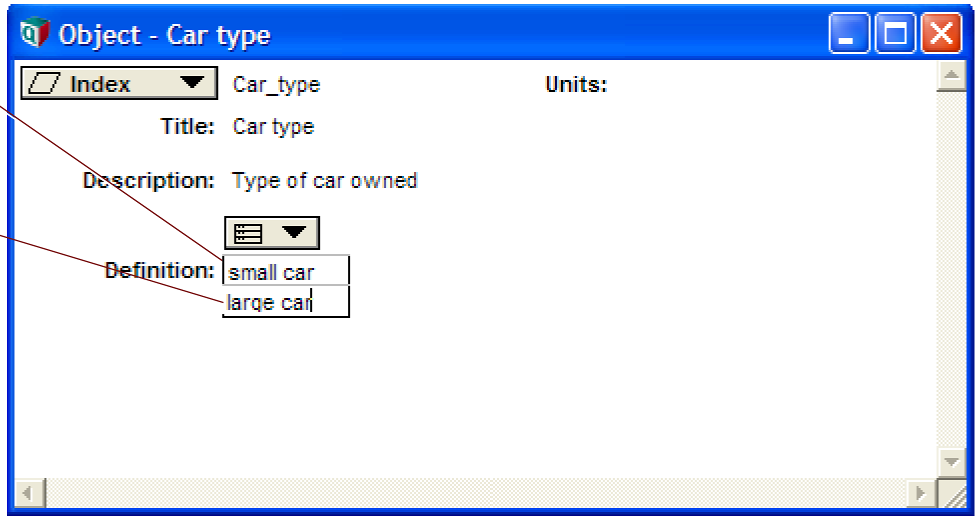
5. Select **List of Labels** from the Expression popup menu.

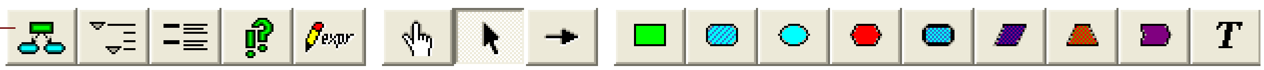


6. Click in the first cell to select it, and type ***small car***; then press the *Enter* key.

7. In the second cell that appears, type ***large car***.

8. Click the **Diagram** button to return to the **Diagram** window.



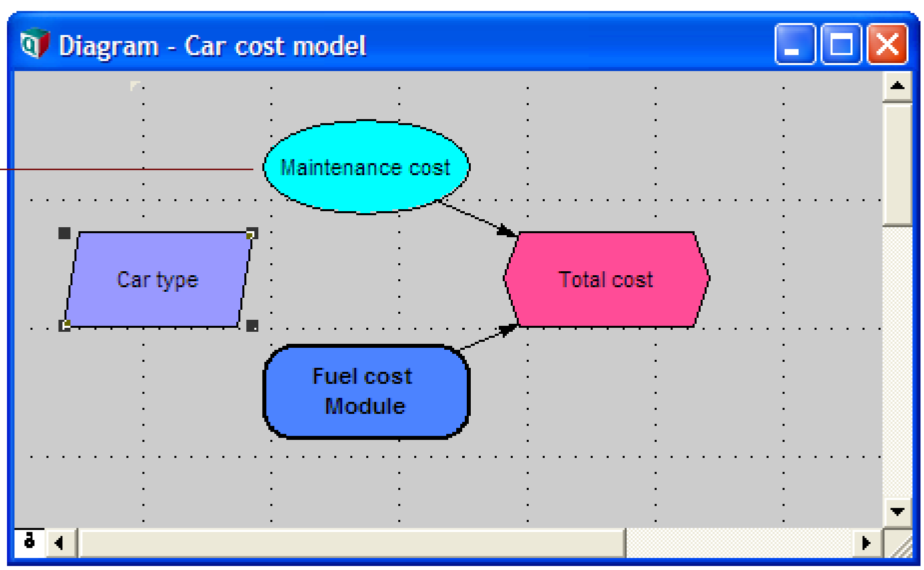


Creating arrays (tables)

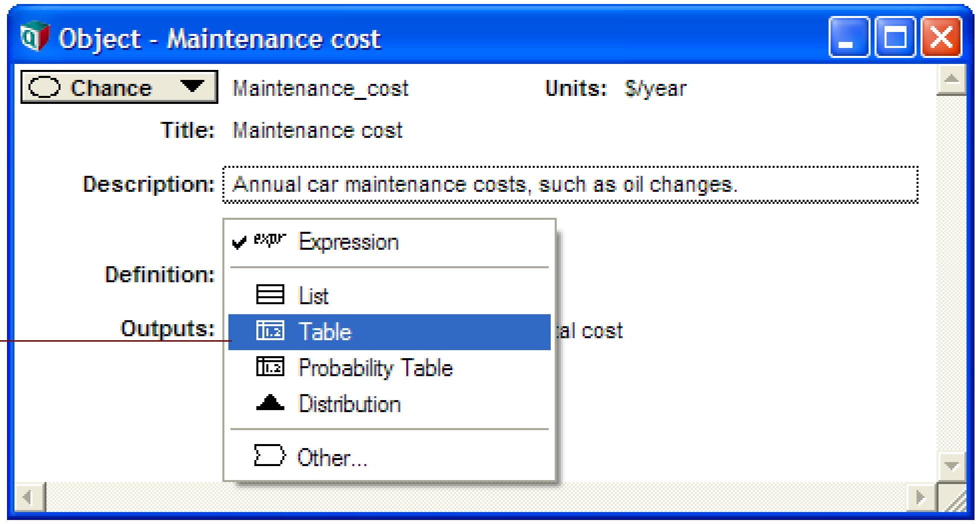
Now that the index, *Car type*, is defined, in this section you will redefine *Maintenance cost* as an array, assigning a different cost for each car type.

You could define *Maintenance cost* as a list of numbers, but Analytica wouldn't know that each number corresponds to a specific car type. By defining *Maintenance cost* as an array (table) with the dimension of *Car type*, Analytica associates the specific maintenance costs to each car type. This makes it easy to use *Maintenance cost* in calculations and obtain total costs by car type.

1. Double-click *Maintenance cost* to open its **Object** window.

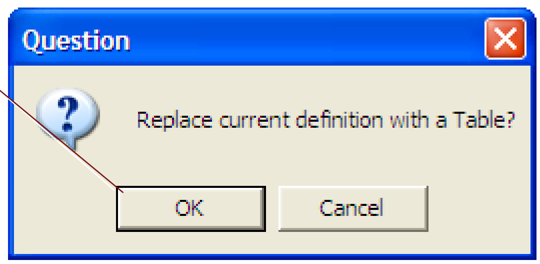


2. Select **Table** from the Expression popup menu.



Because you previously defined *Maintenance cost* as a number, Analytica asks you to confirm that you want to replace the current definition.

3. Click the **OK** button to continue.



Assigning index variables for arrays

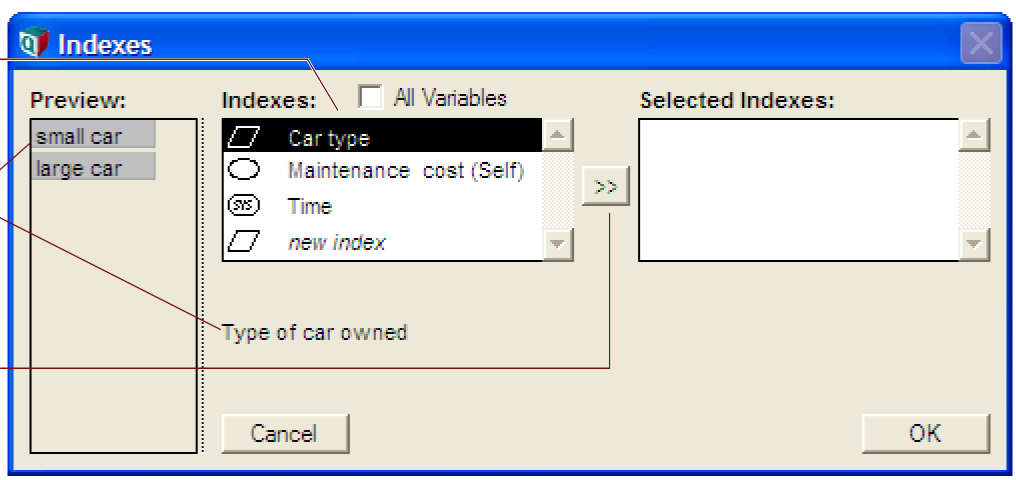
In this section, you will assign the index variable for your array.

When you first create an array (table), the **Indexes** dialog box opens for you to choose the table indexes. Here, *Car type* appears at the top of the Indexes list, on the left.

1. Select *Car type* in the Indexes list.

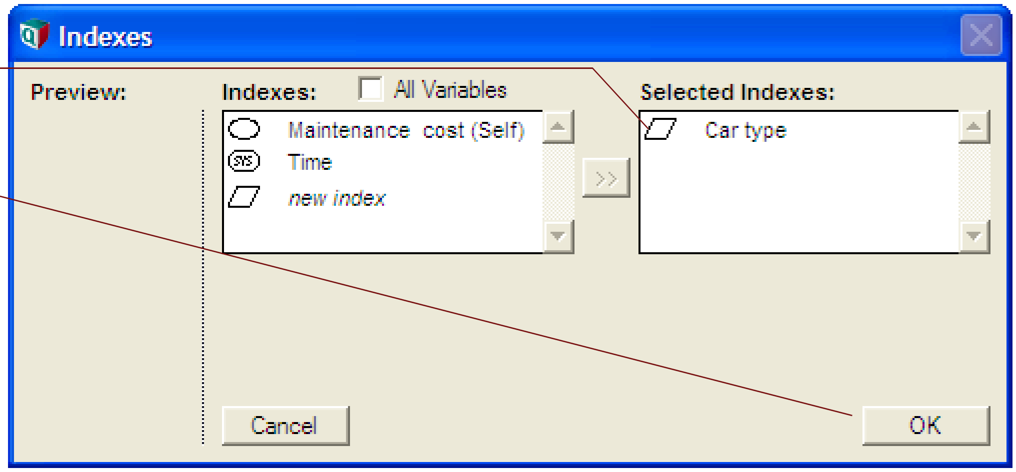
   A preview of the definition and description appear for the selected index.

2. Click the Move >> button to move *Car type* into the Selected Indexes list on the right.



3. *Car type* is now shown in the Selected Indexes list.

Click the **OK** button to accept the index.

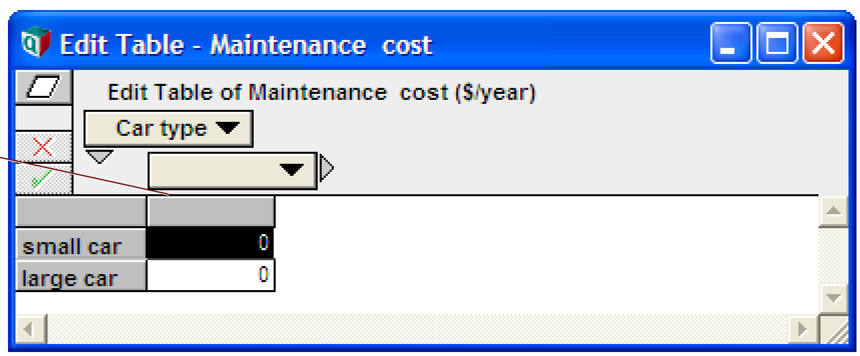


After choosing the index for your table and clicking the **OK** button, an **Edit Table** window opens, indicating that this is an array that you can edit or change.

The **Edit Table** window is like a spreadsheet, with rows and columns labeled according to the indexes selected. As with a spreadsheet, you can navigate between cells using the *Arrow*, *Tab*, *Shift+Tab*, *Return*, and *Shift+Return* keys.

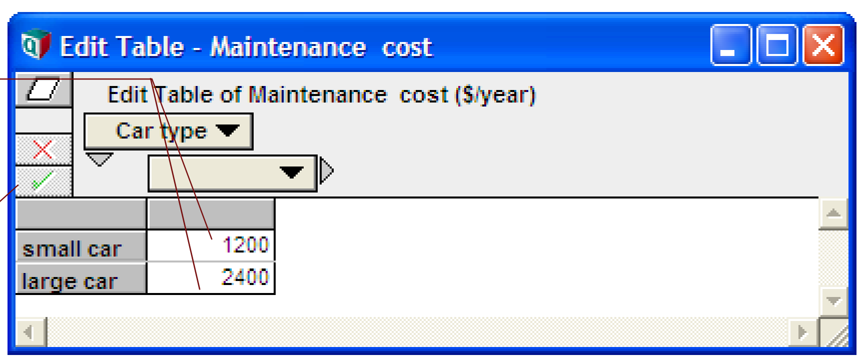
In this example, you have only one index, so *Maintenance cost* is a one-dimensional table. The index, *Car type*, is shown in the first column. The second column holds the corresponding values for maintenance costs.

4. Select the first cell.

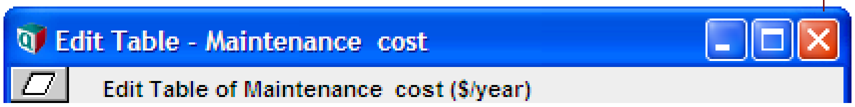


5. Enter ***1200*** for the small car and ***2400*** for the large car.

6. Click the check button ✓ to accept the values.



7. Click the close button to close the window.



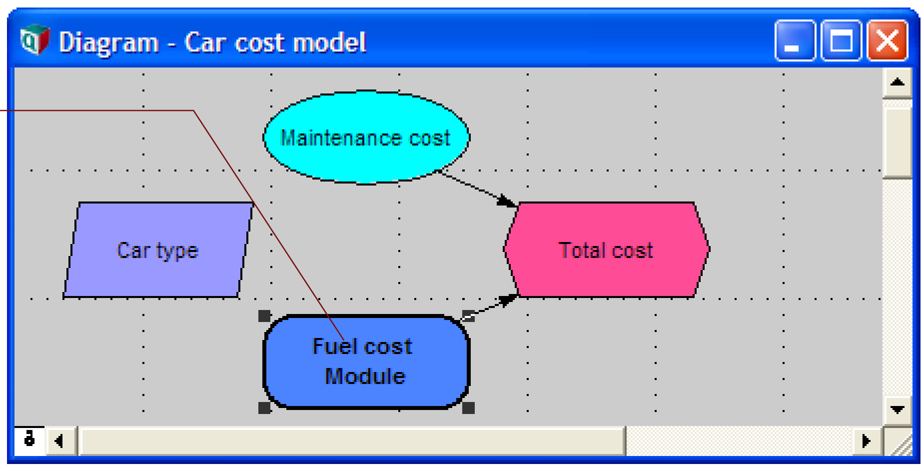
8. Close the *Maintenance cost* **Object** window by clicking its close button.

Creating another array using the same index

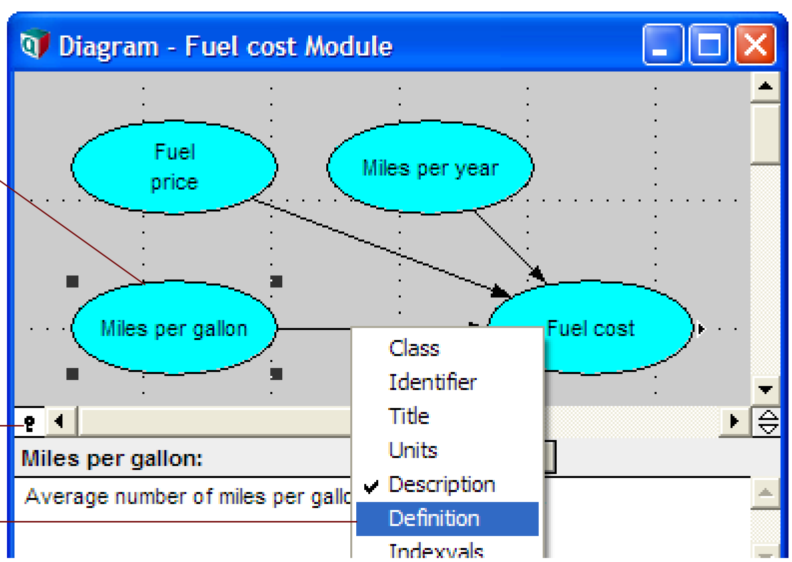
Maintenance cost might not be the only variable that has a different value for each car type. Suppose that *Miles per gallon* is also different for each car type.

In this section you will define *Miles per gallon* as an array indexed by *Car type*.

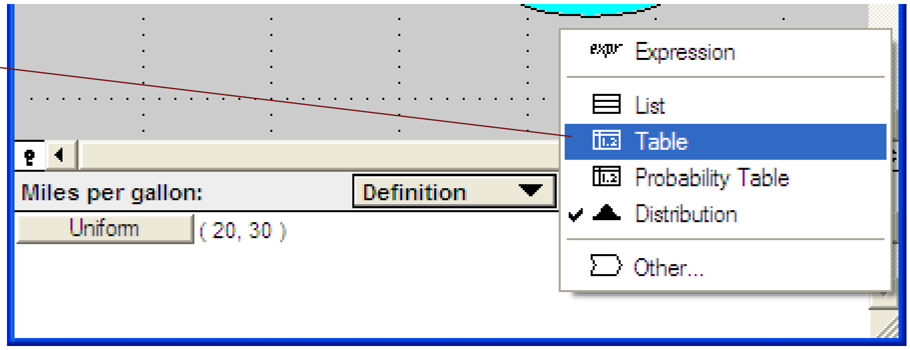
1. Double-click *Fuel cost Module* to open its **Diagram** window.



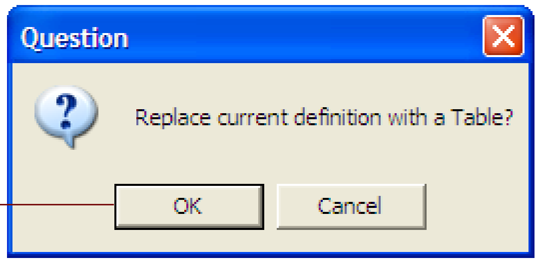
2. Click the key icon to open the **Attribute** panel.
3. Select *Miles per gallon*.
4. Select **Definition** from the Attributes popup menu.



5. Select **Table** from the Expression popup menu.



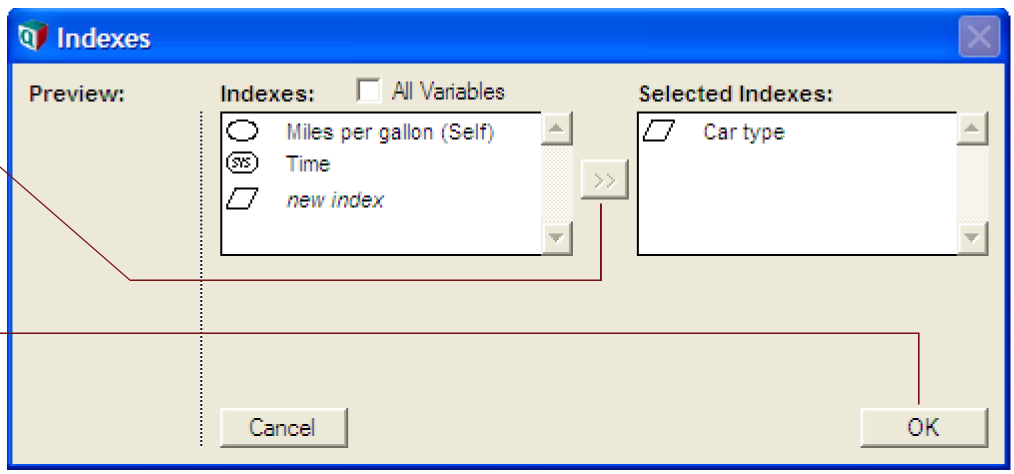
6. Click the **OK** button to change the definition to a table.



7. Select *Car type* in the left-hand list and click the Move >> button.

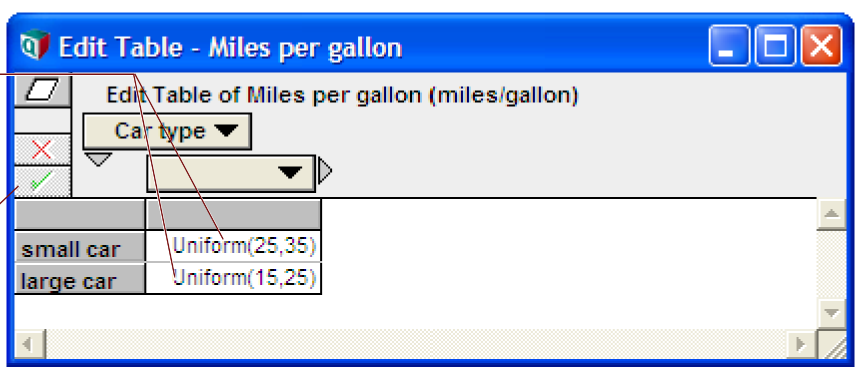
   The result should look as shown here.

8. Click the **OK** button.

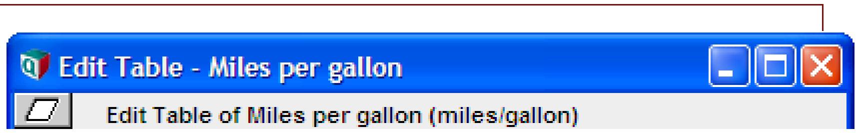


9. Type in a uniform distribution for each car as shown here.

10. Click the check button ✔ to accept the definition of *Miles per gallon*.



11. Close this window by clicking the close button.

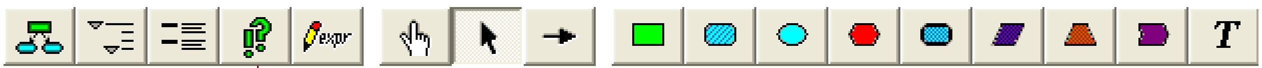


Viewing results of array calculations

So far, you have defined *Miles per gallon* as a one-dimensional table, *Miles per year* as a number, and *Fuel price* as a distribution; *Fuel cost* is defined as a mathematical expression of these variables.

In this section, you will compute the mid value for *Fuel cost* to observe the result of calculating with an array and other types of variables.

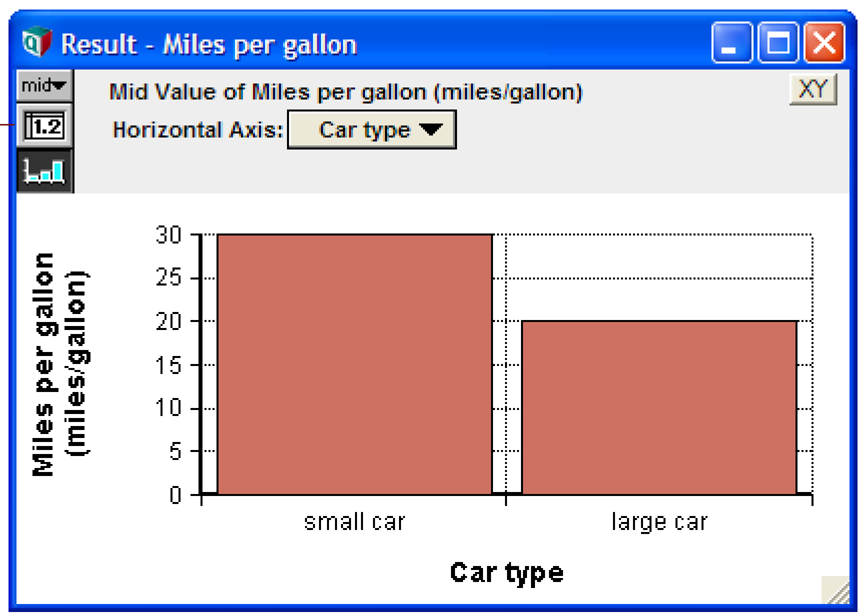
First, you will review the definitions and mid values of the inputs to *Fuel cost*. *Miles per gallon* should still be selected.

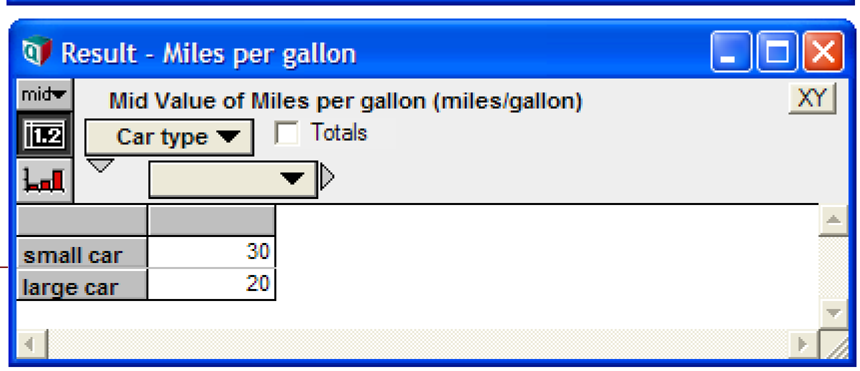


1. Click the **Result** button to evaluate *Miles per gallon*.

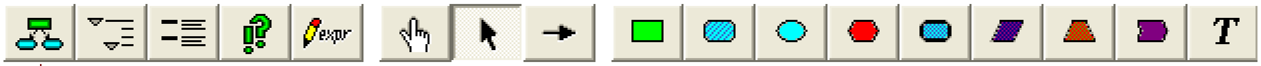
The **Result** window displays a bar graph for the two mid values. (If the **Result** window comes up as a table of values, click the **graph** button to display the graph.)

2. Click the **Table view** button to select the table view.

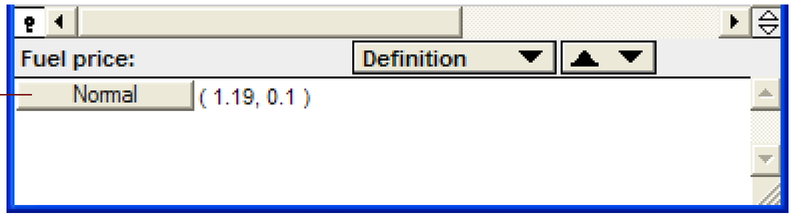




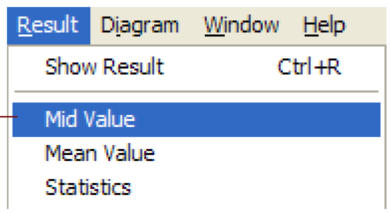
Analytica returns the mid values (20 and 30) for the uniform distributions that you just entered.



3. Click the **Diagram** button to return to the **Diagram** window.

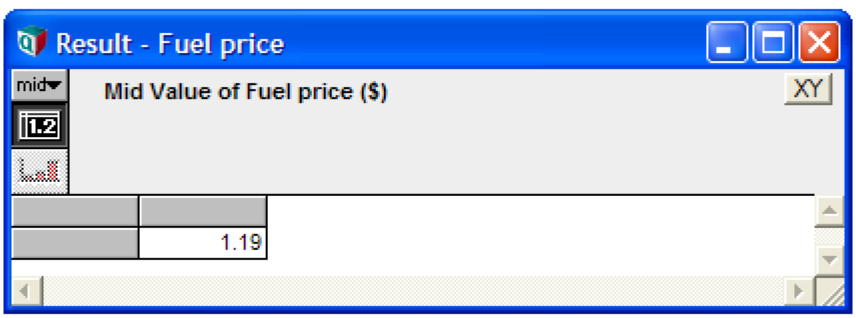


4. Select *Fuel price*. Note that it is defined as a normal distribution.

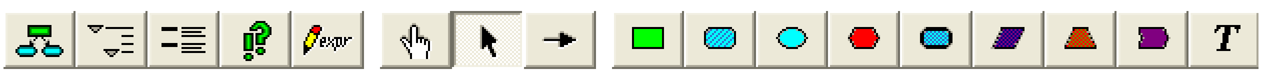


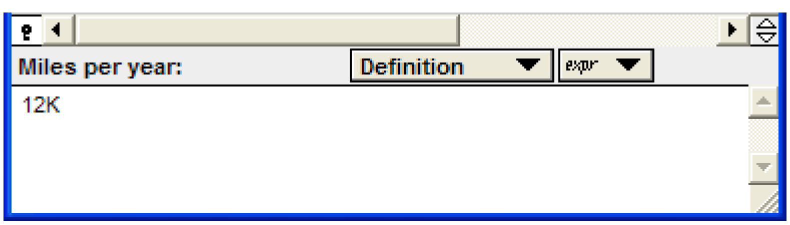
5. Select **Mid Value** from the **Result** menu.

Analytica returns the median of the normal distribution, which is the same as its mean, 1.19.

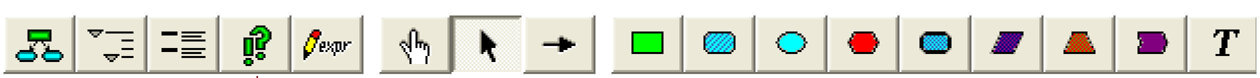


6. Click the **Diagram** button to return to the **Diagram** window.





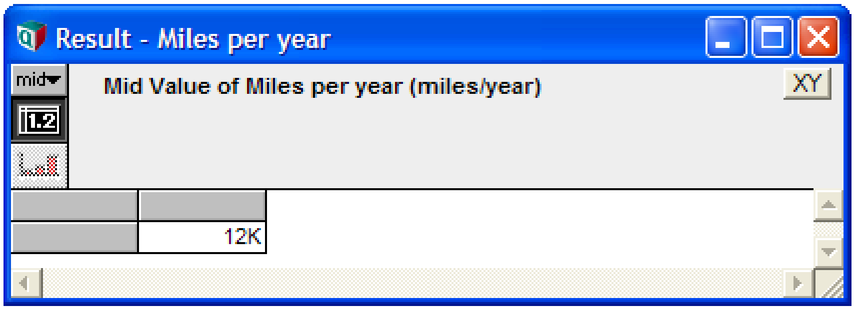
7. Select *Miles per year.* Note that it is defined as a single value.

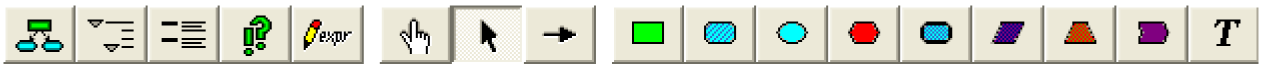


8. Click the **Result** button.

Analytica returns the value 12K.

9. Click the **Diagram** button to return to the **Diagram** window.



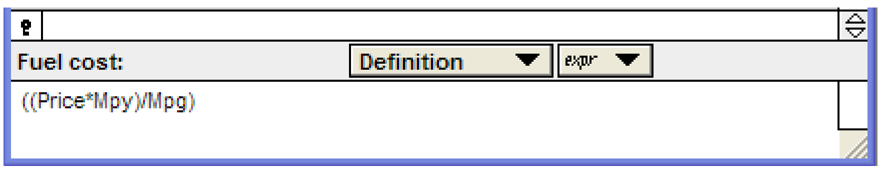


You have seen that the mid values of the inputs to *Fuel cost* evaluate to:

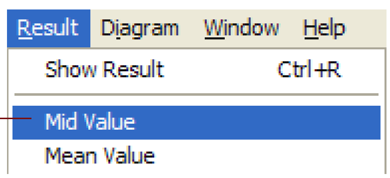
- *Miles per gallon*: small car, 30 and large car, 20 (a one-dimensional table)
- *Fuel price*: a single number, 1.19
- *Miles per year*: a single number, 12K

Now you will see how *Fuel cost* is computed.

10. Select *Fuel cost*. Note that it is defined as an expression.

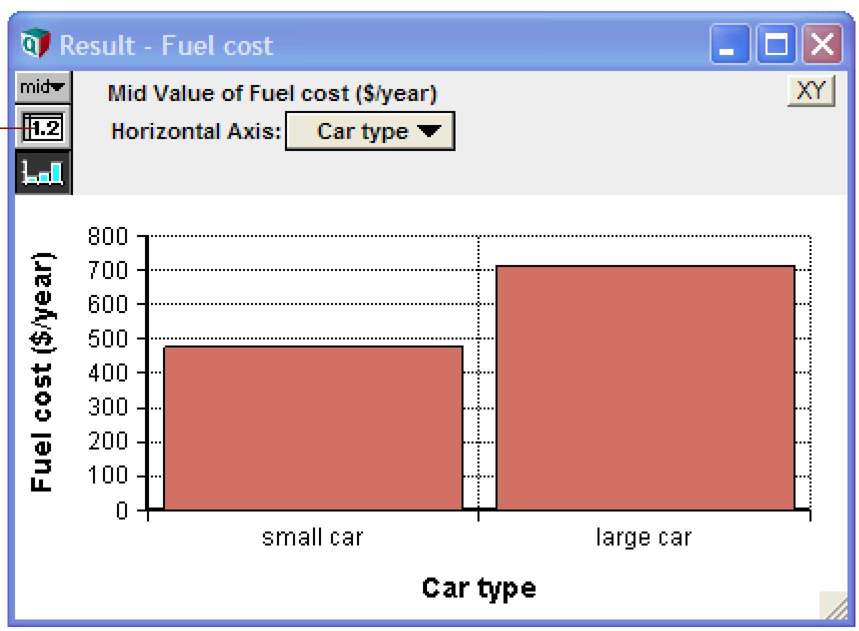


11. Select Mid Value from the **Result** menu.



The **Result** window displays a bar graph with two values. (If the **Result** window comes up as a table of values, click the **graph** button to display the graph.)

12. Click the **Table view** button to select the Table view.

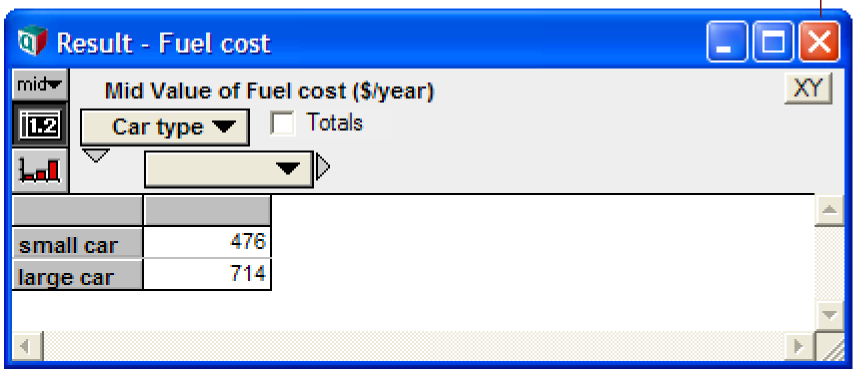


Analytica returns an array, indexed by *Car type*. This is so, even though *Fuel Cost* was not defined as an array, because one of its inputs, *Miles per gallon*, is an array indexed by *Car type*.

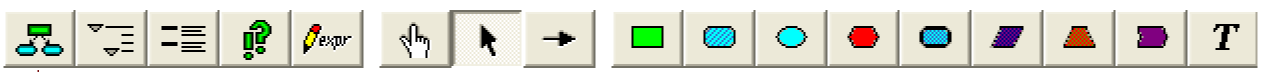
You can see how the values were obtained:

```
Small car:   1.19 * 12K / 30 = 476
Large car:   1.19 * 12K / 20 = 714
```

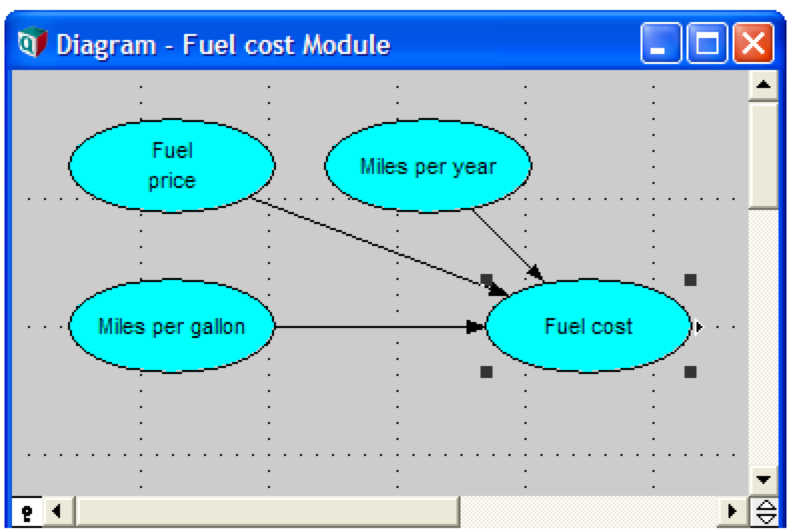
13. Close the **Result** window to go to the *Fuel Cost* **Diagram** window.

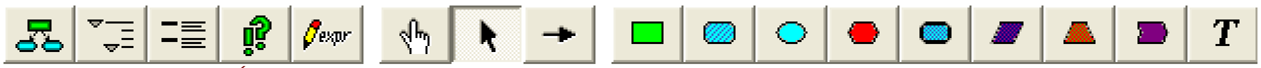


Now you will see how *Total cost* is computed.



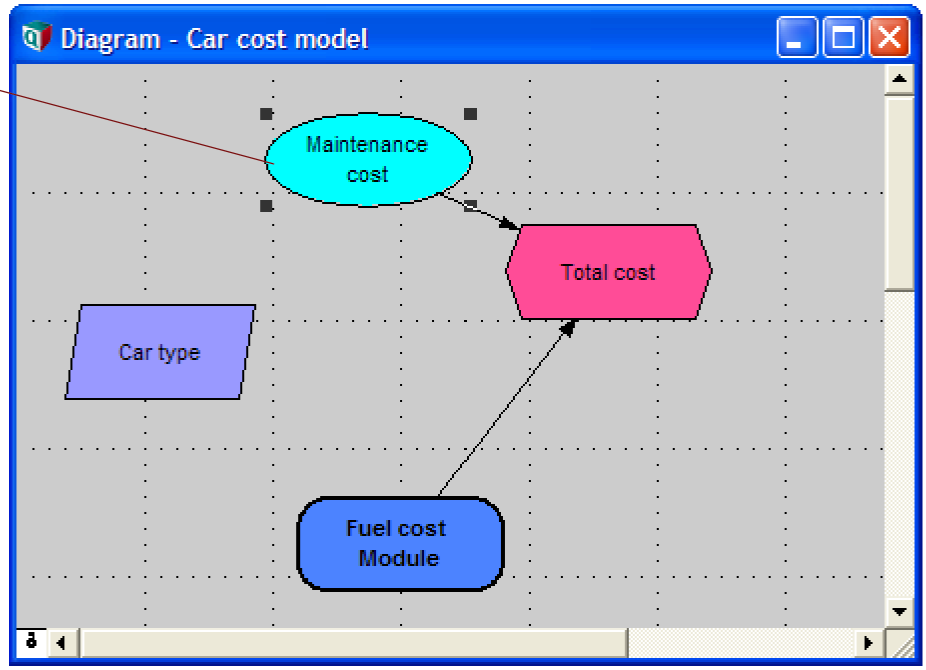
14. Click the **Diagram** button to go to the *Car Cost Model* **Diagram** window.





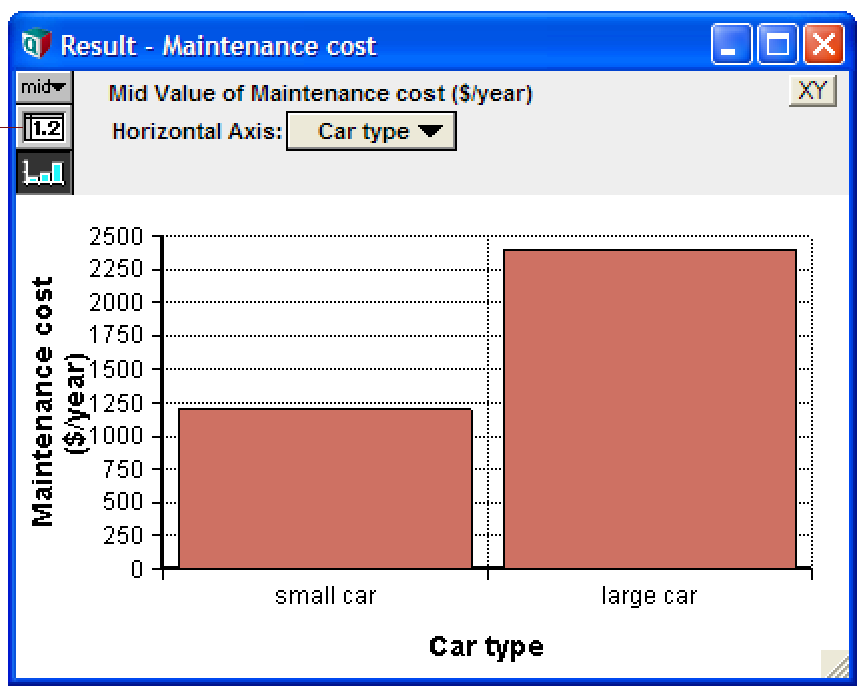
15. Select *Maintenance cost.*

16. Click the **Result** button

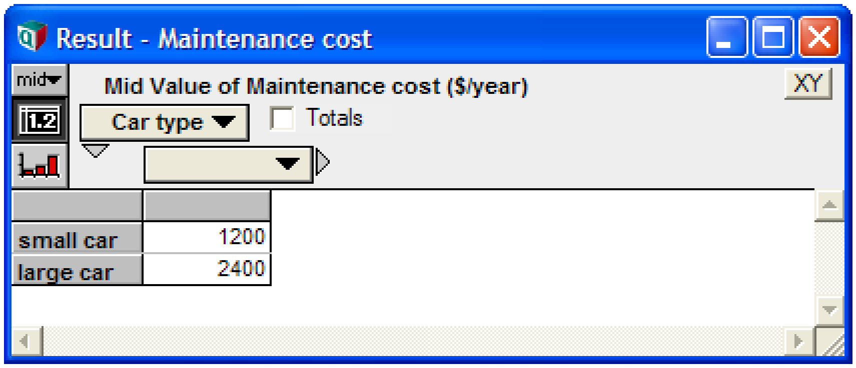


The **Result** window displays a bar graph for the two mid values. (If the **Result** window comes up as a table of values, click the **graph** button to display the graph.)

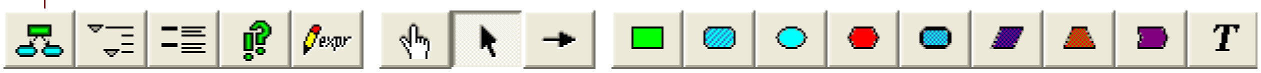
17. Click the **Table view** button to select the Table view.



Analytica returns the array that you entered (1200 for the small car and 2400 for the large car).

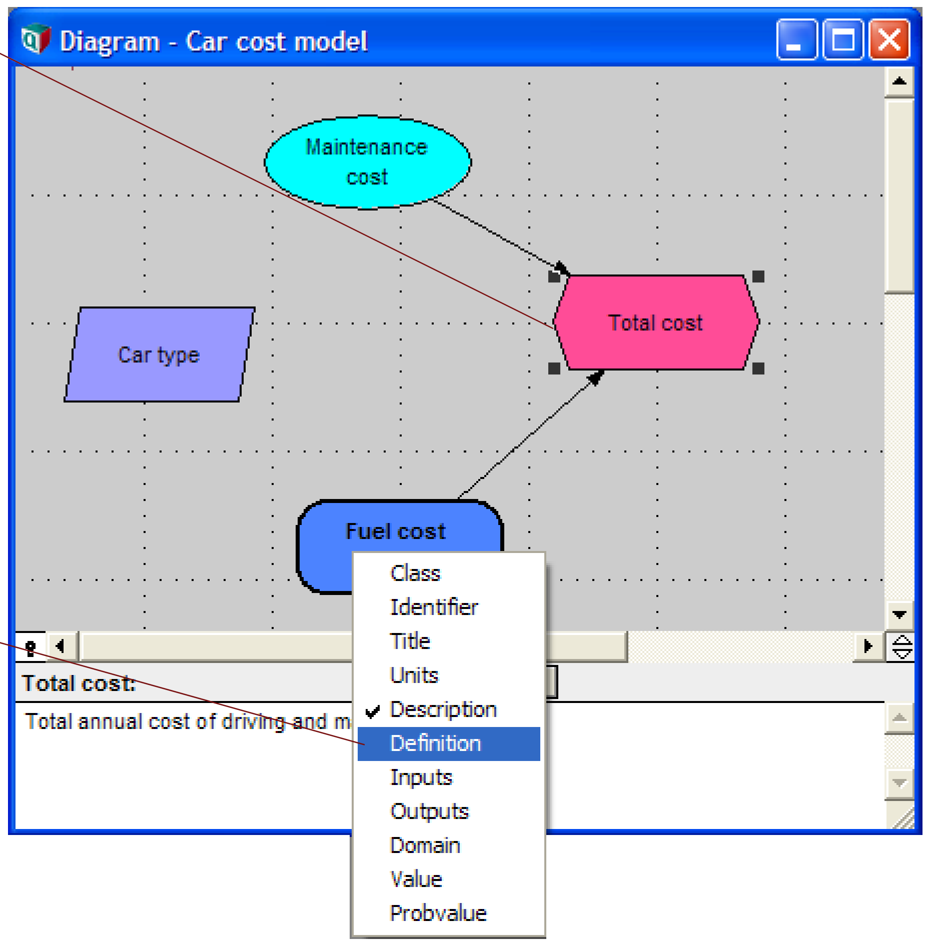


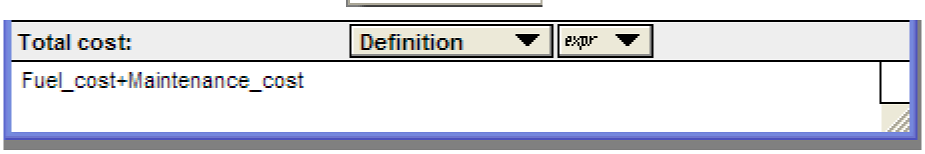
18. Click the **Diagram** button to return to the **Diagram** window.



19. Select *Total cost.*

20. Select **Definition** from the Attribute popup menu.





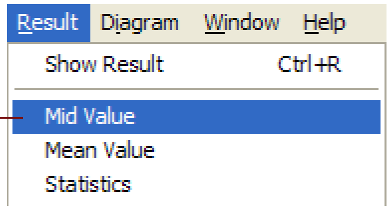
Note that this definition is a simple arithmetic expression.

You know, though, that both inputs are arrays indexed by *Car type*. You can expect that *Total cost* is also an array indexed by *Car type*:

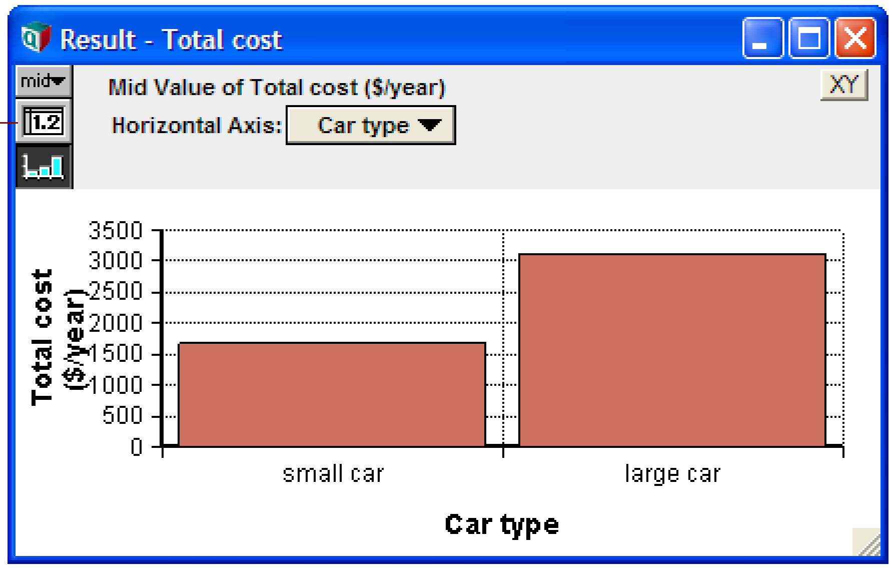
```
Small car:    476 + 1200 = 1676
```

```
Large car:    714 + 2400 = 3114
```

21. Select **Mid Value** from the **Result** menu.

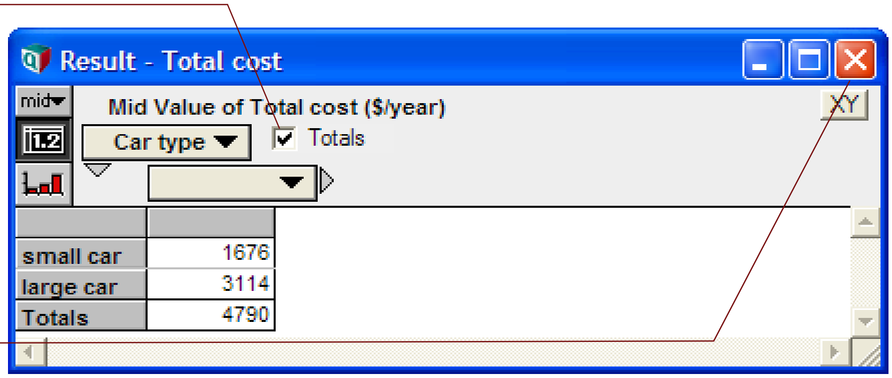


22. Click the **Table** button to view the result as a table.



23. Click the checkbox next to **Totals** to total the values.

24. Close the **Result** window.



The values for *Total Cost* are as expected. Because of Analytica's Intelligent Arrays, *Total Cost* was calculated correctly for each car type.

Tip The sum of the costs for both cars is $4790. Note this value for the next section.

Combining results from tables

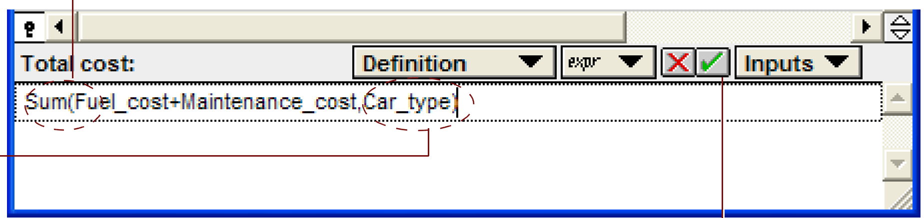
You might wish to view the sum of *Total cost* for all *Car types* — for example, to determine whether you can afford both cars. In this section, you will sum across the dimension *Car type*, giving a single number.

You will sum by using one of Analytica's built-in array functions, the **Sum()** function. For details about the **Sum()** function, see "Array-reducing functions" in Chapter 12 of the *Analytica User Guide*.

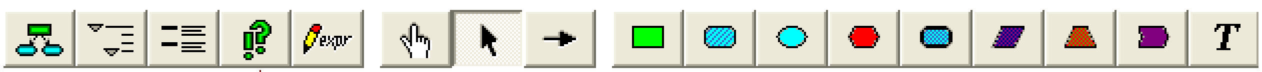
1. Click at the left side of the Definition field and type: ***Sum(***

2. Click at the end of the Definition field and type: ***,Car_type)***

   The Definition field should now look like the expression here.



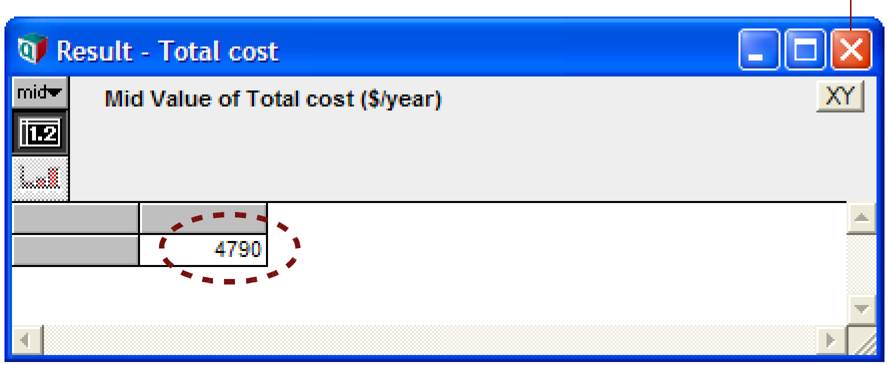
3. Click the check button to accept the new definition.



4. Click the **Result** button to view the result.

The value is now the sum of the *Car type* values, which totals $4790, as was noted in the previous section.

5. Close the **Result** window.



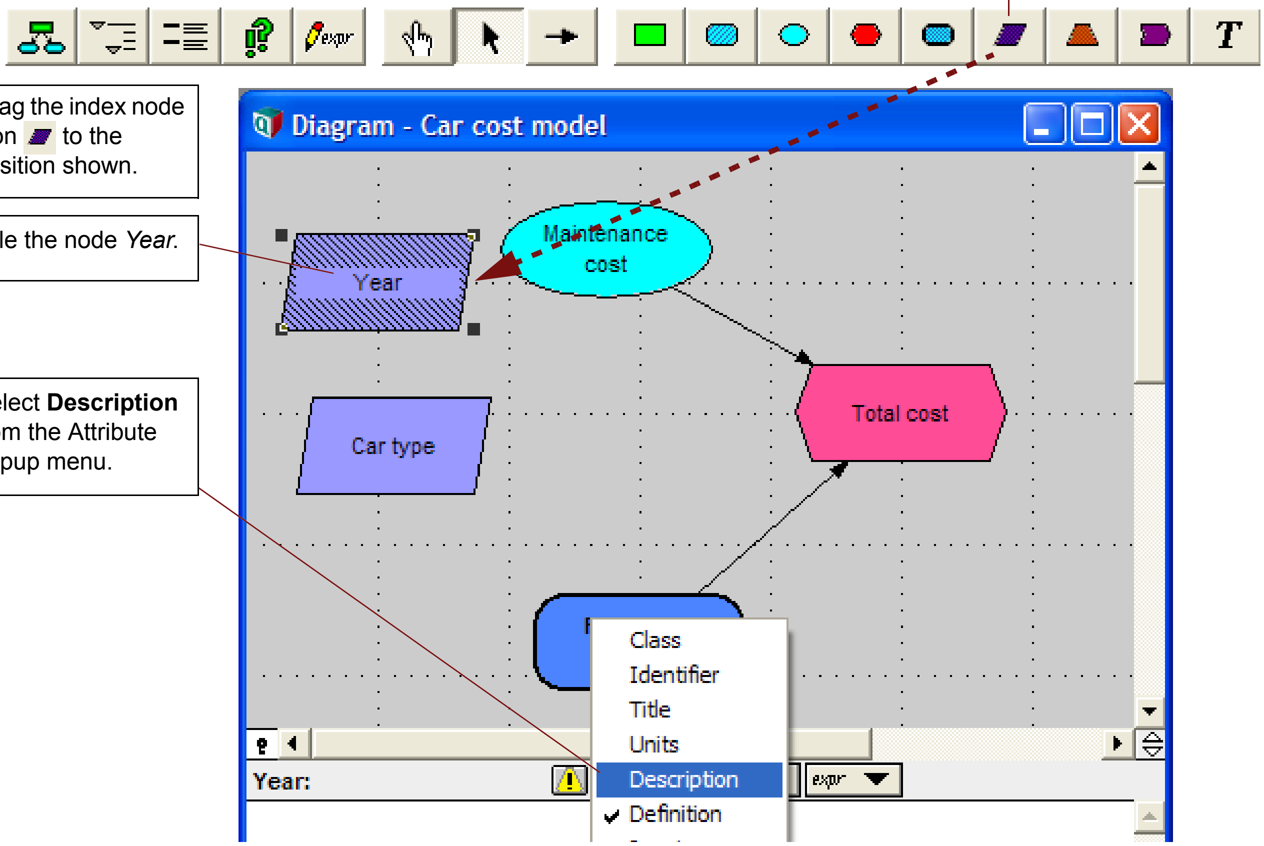
Adding dimensions to variables

In this section you will further extend the *Car Cost* model by adding another dimension to the *Maintenance cost* variable.

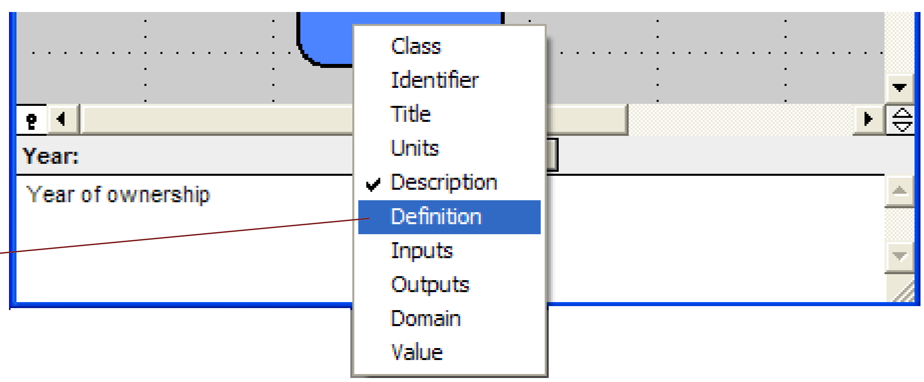
Let's assume that you want to estimate the cost of ownership of your two cars over the next three years. You think that each car's maintenance cost will increase over time. You are also planning a cross-country trip in the small car next year.

Start by creating a second index variable, *Year*.

1. Drag the index node icon to the position shown.
2. Title the node *Year*.
3. Select **Description** from the Attribute popup menu.

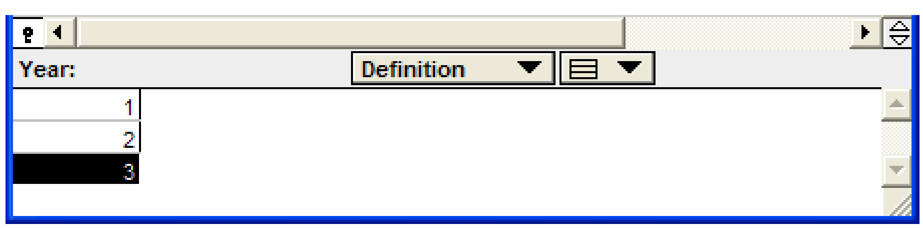


4. In the Description field, type *Year of ownership*, and press *Alt+Enter*.
5. Select **Definition** from the Attribute popup menu.



You will define this index as the numbers 1, 2, and 3, representing the 1st, 2nd, and 3rd years of car ownership.

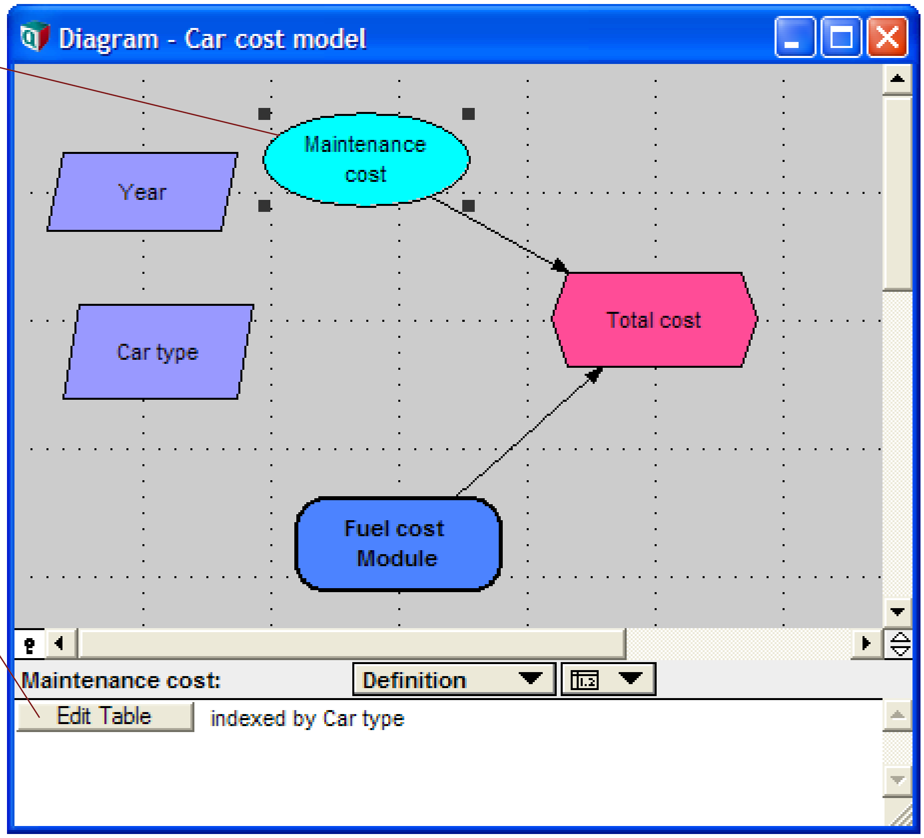
6. Select **List** from the Expression popup menu.
7. 1 is entered by default in the first cell. Select the cell, then press the *Enter* key twice.



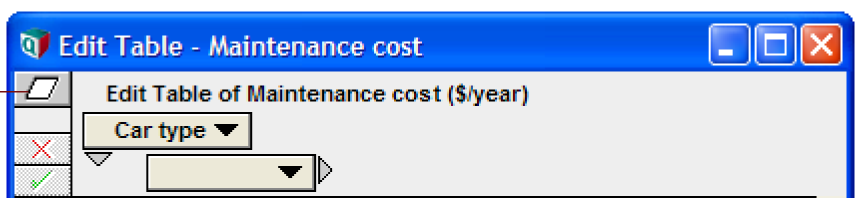
Now you are ready to expand *Maintenance cost*, making it a table indexed by *Year*.

8. Select *Maintenance cost.*

9. Click the **Edit Table** button to open the **Edit Table** window.



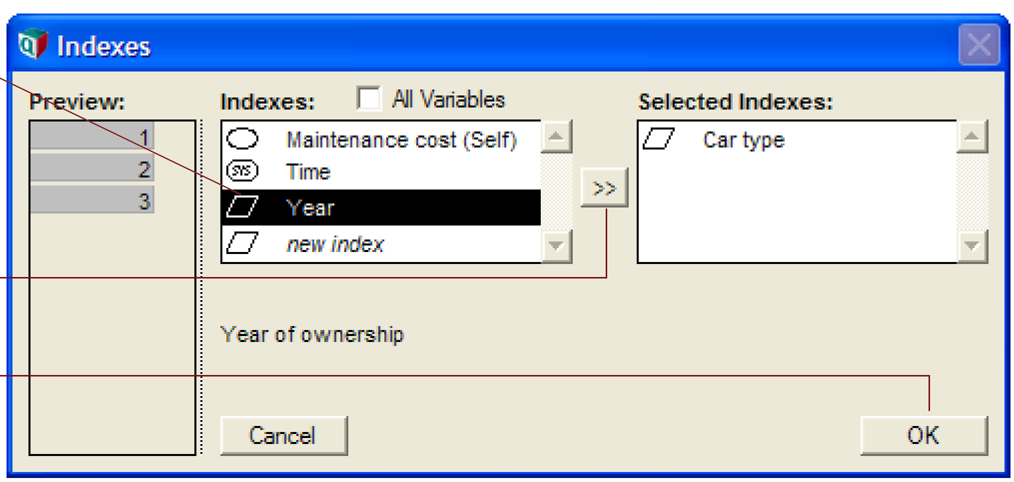
10. Click the indexes button to open the **Indexes** dialog box.



11. Select *Year* in the Indexes list.

12. Click the Move >> button to move *Year* into the Selected Indexes list on the right.

13. Click the **OK** button to accept the change.

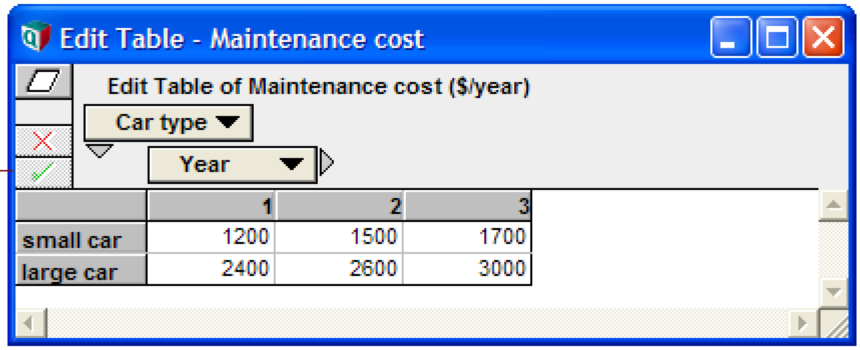


The **Edit Table** window displays again, showing that *Maintenance cost* now has two dimensions. The previous values (1200 and 2400) have been duplicated across all the years.

14. Change the values in the cells, as shown here.

15. Click the check button ✓ to accept the changes.

16. Close the **Edit Table** window by clicking the close button.



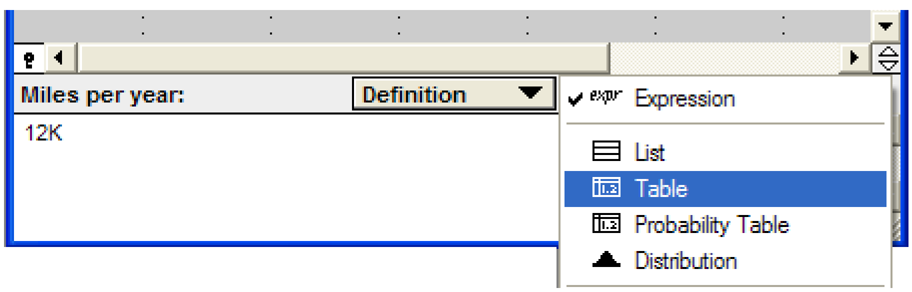
Completing the model

The last step is to expand *Miles per year* to be a table indexed by *Car type* and *Year*, so that you can include the increased miles for next year's cross-country trip and see the effect on fuel cost. Do this step on your own.

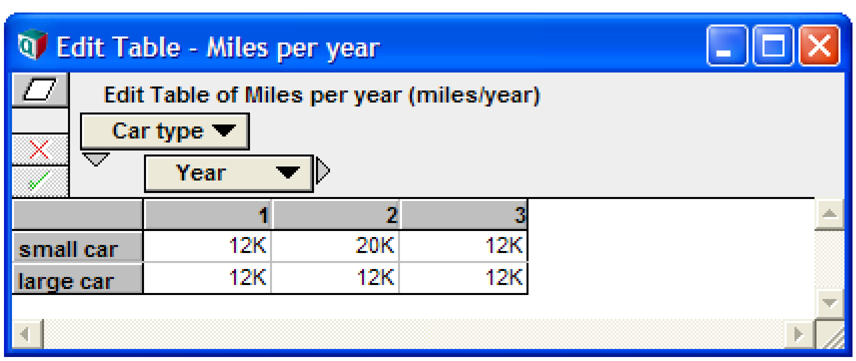
The miles per year variable is in the *Fuel cost* module, so you need to open that module and select *Miles per year* to open its **Object** window.

1. Change *Miles per year* to be defined as a table.

   Select *Car type* and *Year* as indexes for the table.

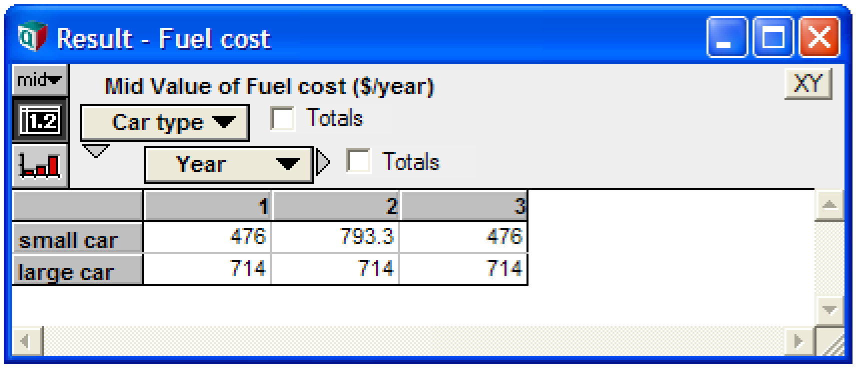


2. Enter the values in the cells, as shown here.



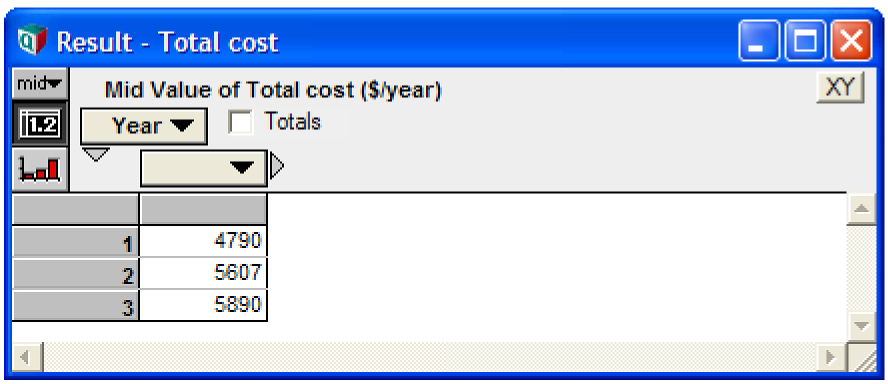
Evaluate *Fuel cost*. You do not need to change its definition for it to calculate correctly.

3. Examine the *Mid Value of Fuel cost.*



Finally, evaluate *Total cost* in the *Car Cost model* **Diagram** window. Again, you will not change its definition. Both of its inputs are two dimensional, indexed by *Year* and *Car type*, and it is computed by summing across the *Car type* dimension. So you might expect *Total cost* to be a one dimensional array, indexed by *Year*.

4. Examine the *Mid Value of Total cost.*



As expected, *Total cost* is a one dimensional array, giving the total cost for each year.

Creating arrays (tables): summary

In this chapter, you have:

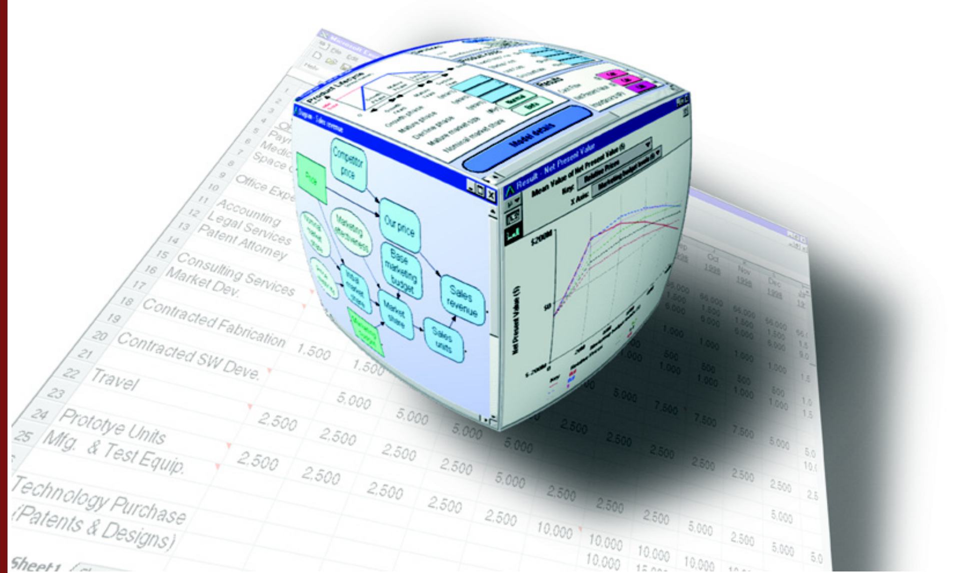
- Defined index variables.
- Defined a table and selected its indexes.
- Defined other variables using the same indexes.
- Viewed the results of table calculations.
- Combined results from a table using the **Sum()** function.

This chapter was a brief introduction to Intelligent Arrays, one of Analytica's most powerful features. For more information, see the *Analytica User Guide*, Chapters 11 and 12.

Chapter 6 *Creating the Party Problem Model*

This chapter shows you how to:

- Define a variable as a probability table
- Define a variable as a deterministic table



In this chapter you will create a new Analytica model called *Party Problem*. (For information about how to create a new model, see the beginning of Chapter 4, “Creating Models”.) The **Party Problem** model illustrates the use of discrete probability distributions.

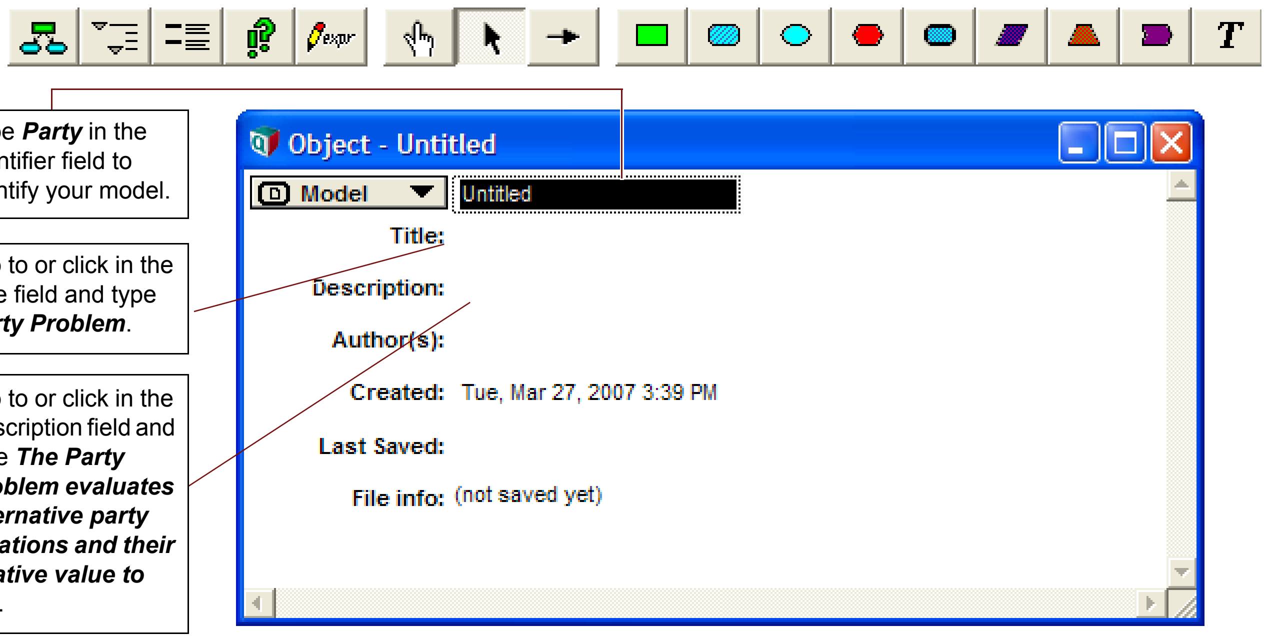
You should study this chapter if your models use discrete or conditional probabilities.

In many kinds of models, your variables can be described using probability distributions based on expert judgment or on empirical data. On those occasions when the outcomes are discrete or qualitative (for example, low, medium, or high), you might need to use ***discrete probability*** distributions.

In the **Party Problem** model, the key uncertain variable is weather: it could be sunny or rainy. The weather has an impact on the decision about the location of a party — indoors, on a porch, or outdoors.

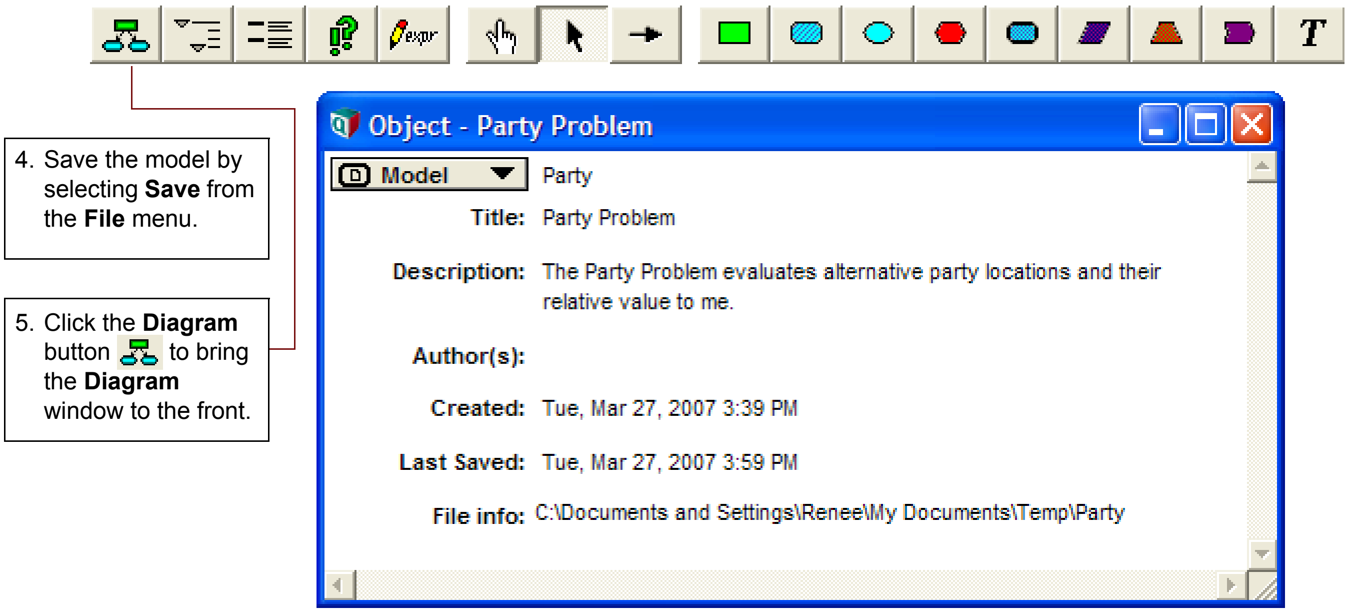
Documenting the model

Specify the information shown in this model’s **Object** window.



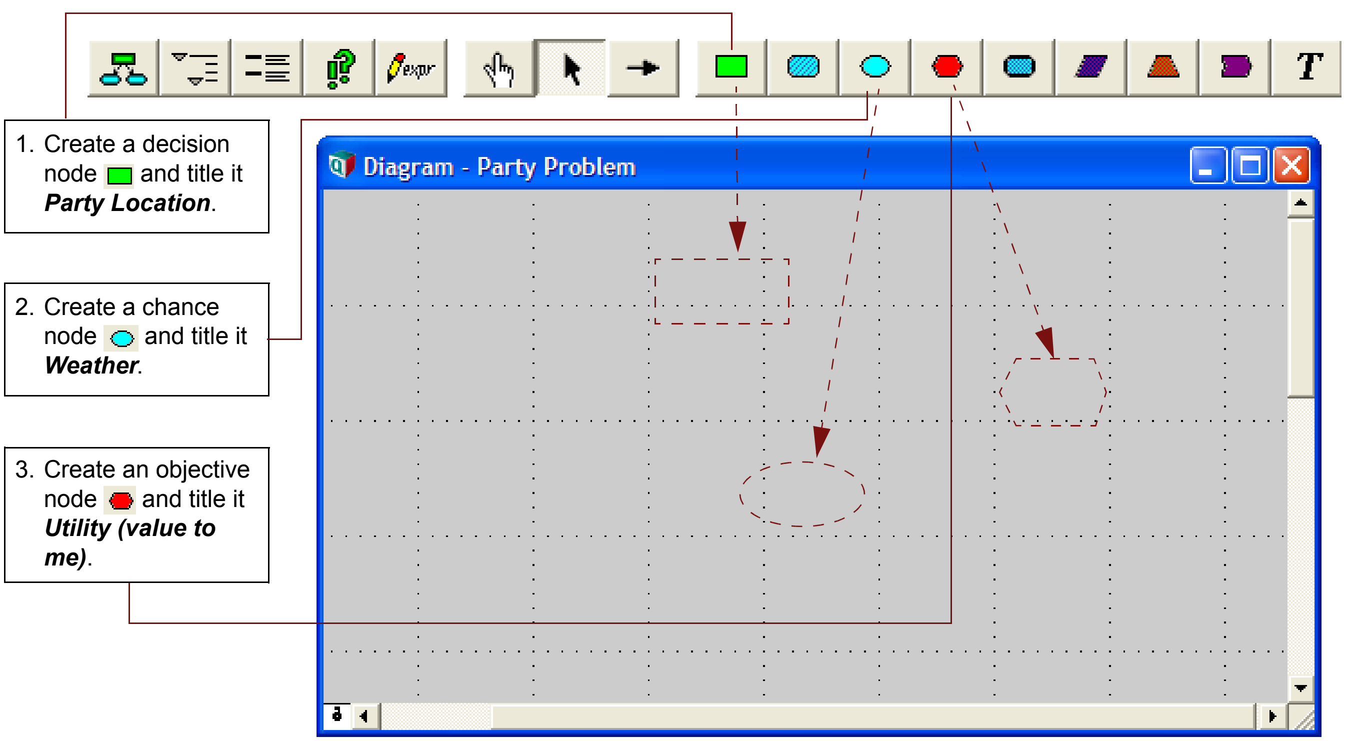
1. Type ***Party*** in the Identifier field to identify your model.
2. Tab to or click in the Title field and type ***Party Problem***.
3. Tab to or click in the Description field and type ***The Party Problem evaluates alternative party locations and their relative value to me***.

The **Object** window should now look similar to this:

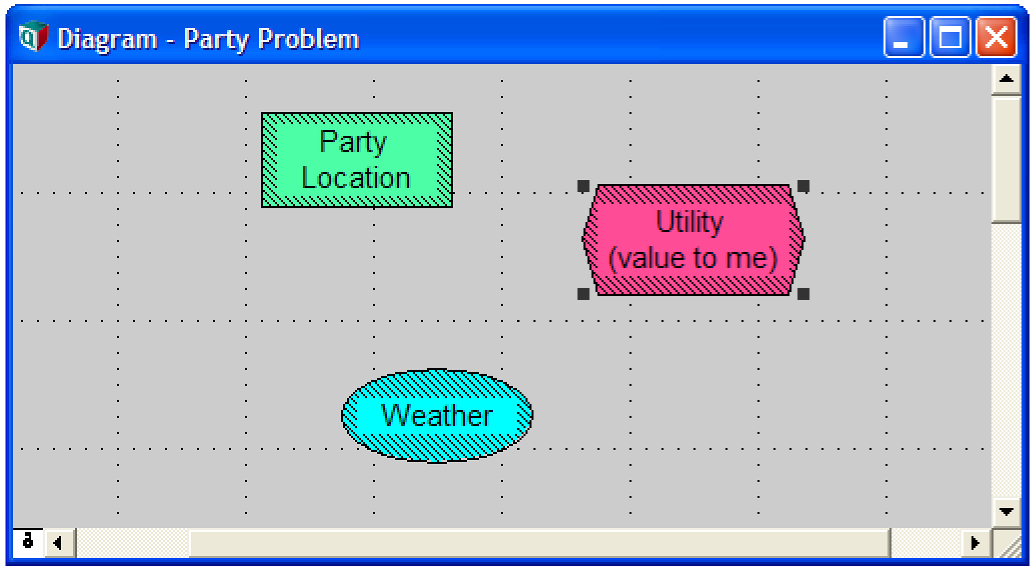


Creating the Party Location, Weather, and Utility variables

Using the techniques introduced in “Creating variables” on page 63, you will create a **Party Location** decision variable, a **Weather** chance variable, and a **Utility** objective variable.

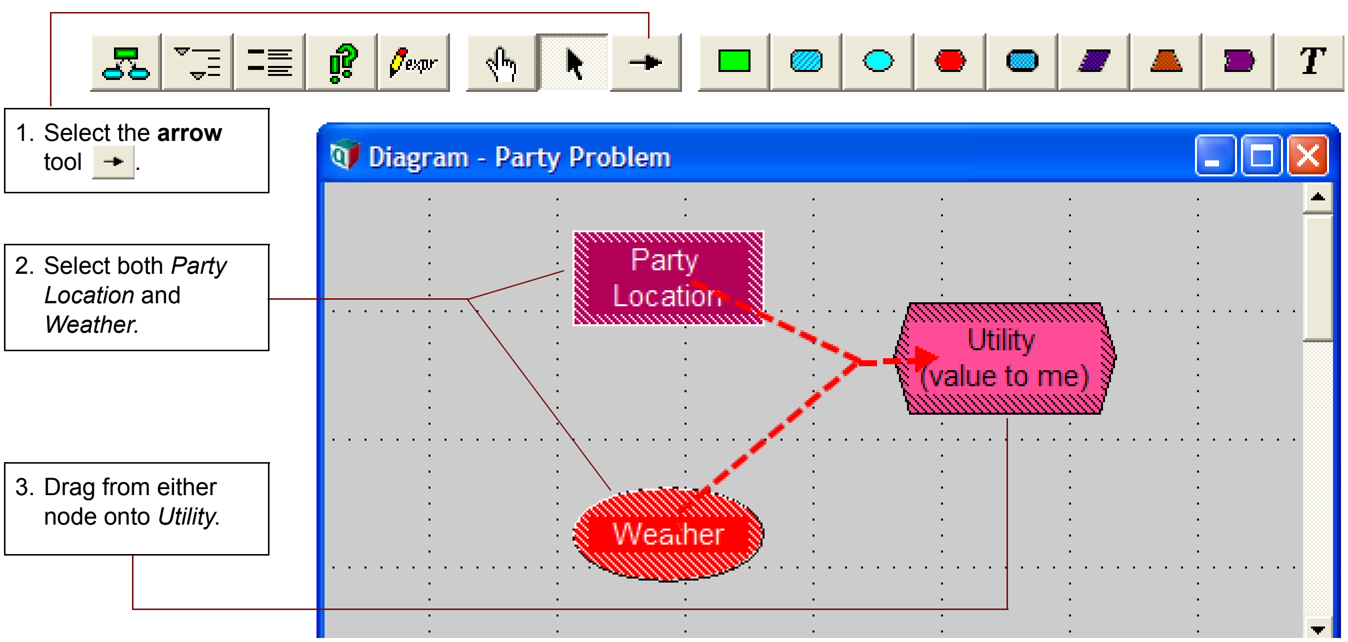


Your diagram should now look like this:

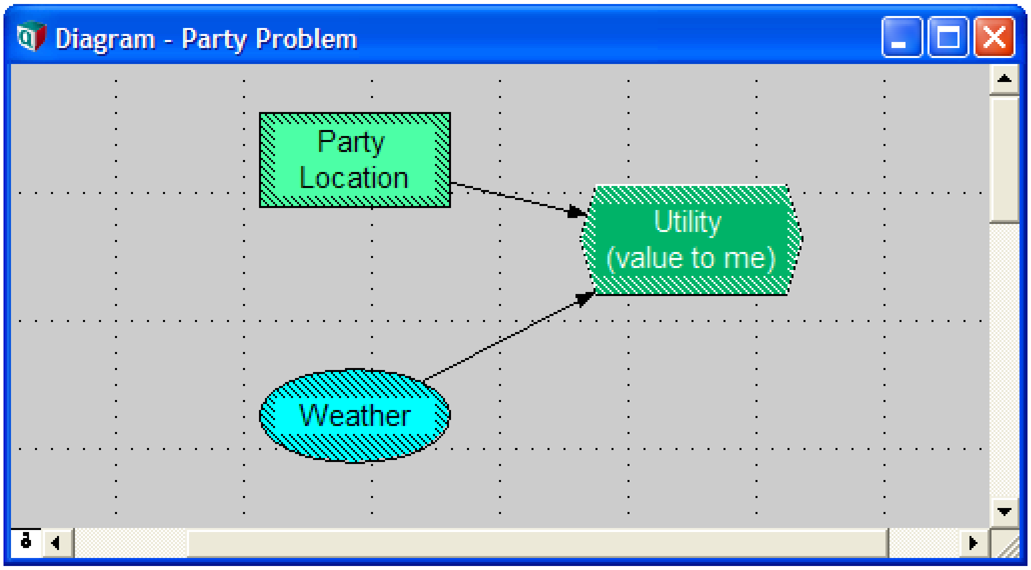


Drawing arrows between variables

In this section, you will draw arrows from *Party Location* and *Weather* to *Utility* using the techniques introduced in “Connecting multiple arrows” on page 72.



Your diagram should now look like this:

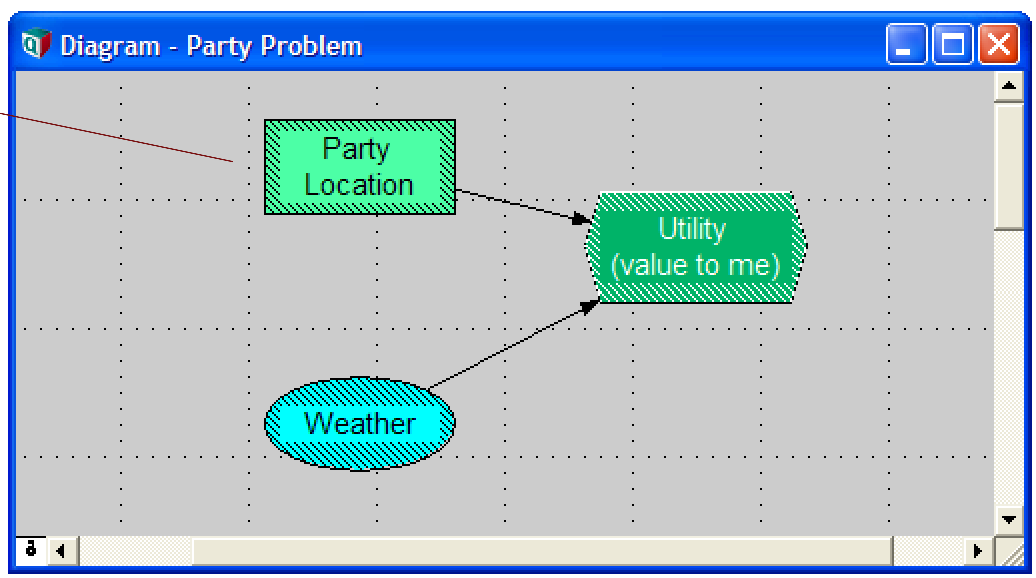


Defining Party Location as a list of labels

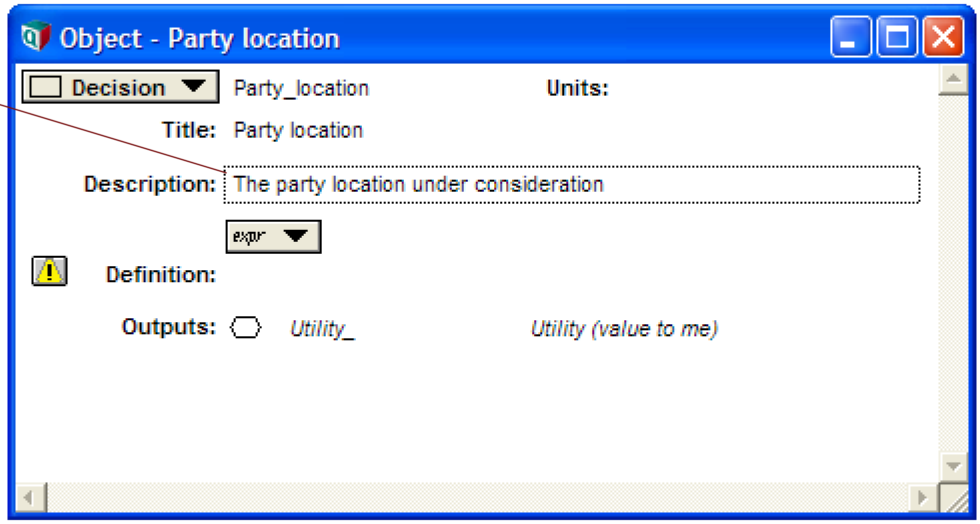
Consider that there are three possible locations where you could hold your party — indoors, on the porch, and outdoors. You will define *Party Location* as a list of labels identifying each location.

Party Location is used to index the *Utility (value to me)* objective node, so it is similar to the *Car Type* index variable created for the *Car Cost* model in the section “Creating index variables” beginning on page 98. (*Party Location* is a decision variable, rather than an index variable, because it is directly under your control.)

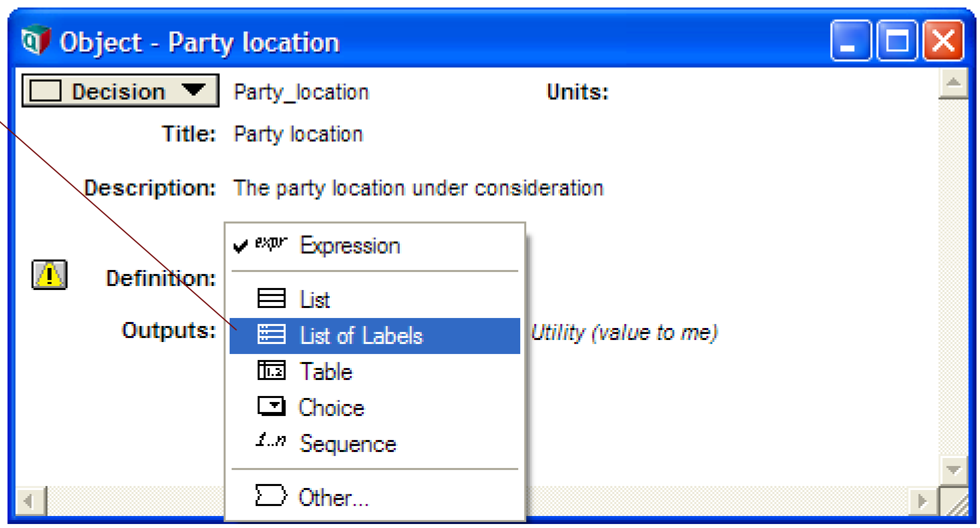
1. Double-click *Party Location* to open its **Object** window.



2. Click in the Description field and type ***The party locations under consideration***.



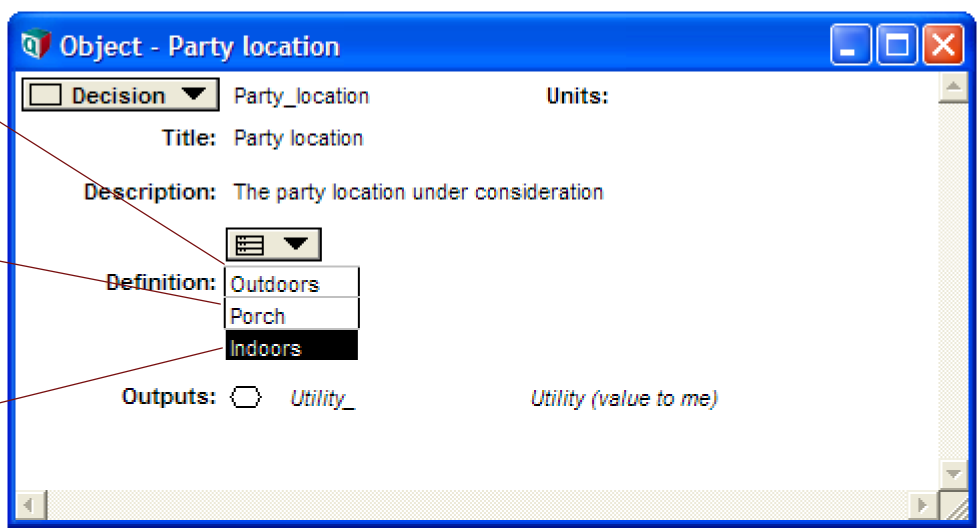
3. Select **List of Labels** from the Expression popup menu.



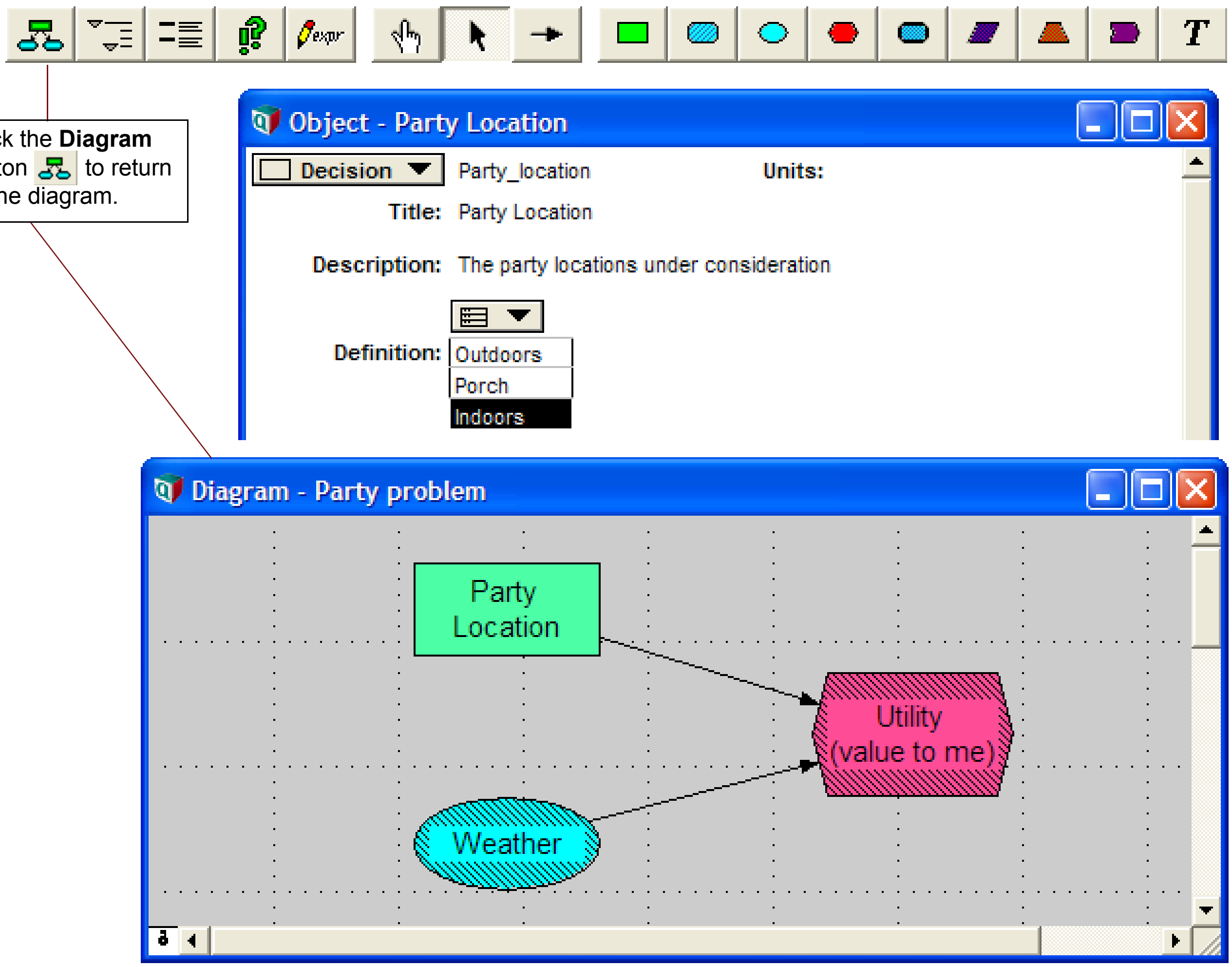
4. Click in the first cell and type ***Outdoors***; then press the *Enter* key.

5. In the next cell, type ***Porch***; then press the *Enter* key.

6. In the next cell, type ***Indoors***; then press *Alt+Enter* to accept the changes.



7. Click the **Diagram** button to return to the diagram.

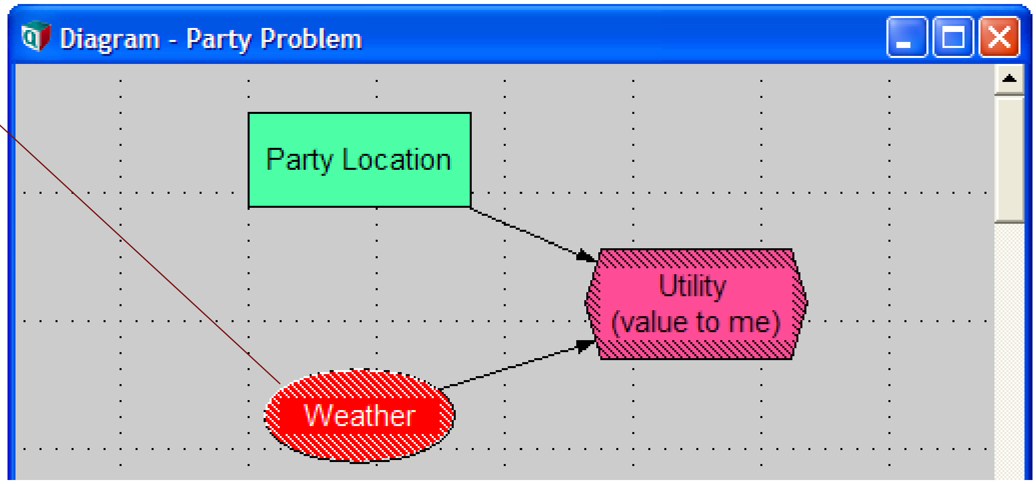


Defining Weather as a probability table

In this model, as in real life, weather is unpredictable.

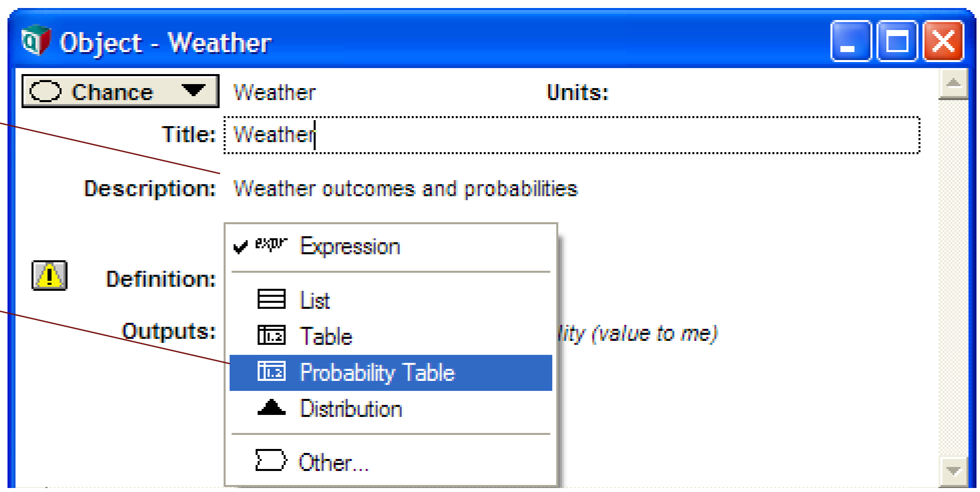
In this section, you will characterize weather as having two possible discrete outcomes, either sunny or rainy. In addition, you will assign probabilities for each possible outcome. This is done by defining weather as a ***probability table***. A probability table is Analytica's function for describing discrete probabilities.

1. Double-click *Weather* to open its **Object** window.



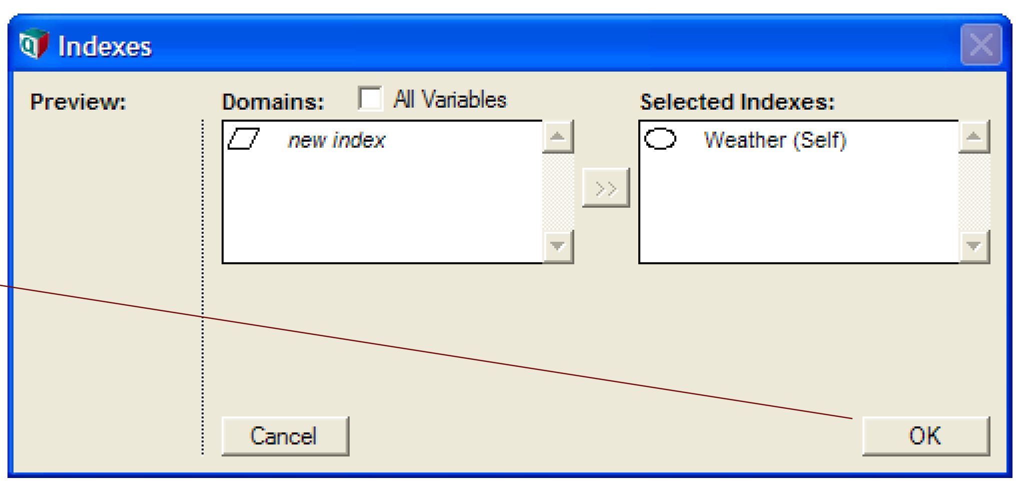
2. Click in the Description field and type ***Weather outcomes and probabilities***.

3. Select **Probability Table** from the Expression popup menu.



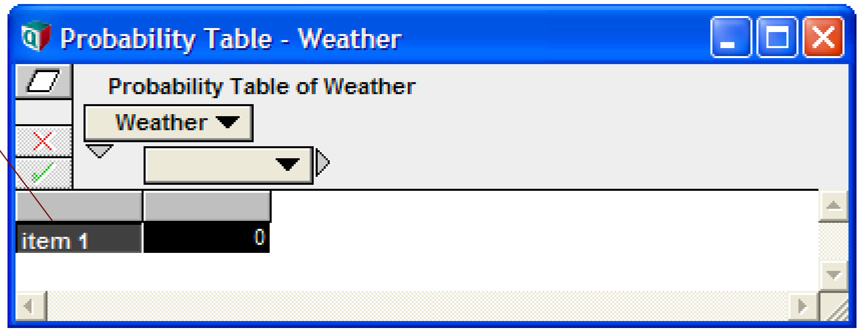
The **Indexes** dialog box opens to confirm your choice of index for the table. *Weather (Self)* appears in the Selected Indexes list. *Self* indicates that the index — the possible outcomes of the discrete distribution — is contained within the probability table. *Self* is required as an index of a probability table.

4. Click the **OK** button.



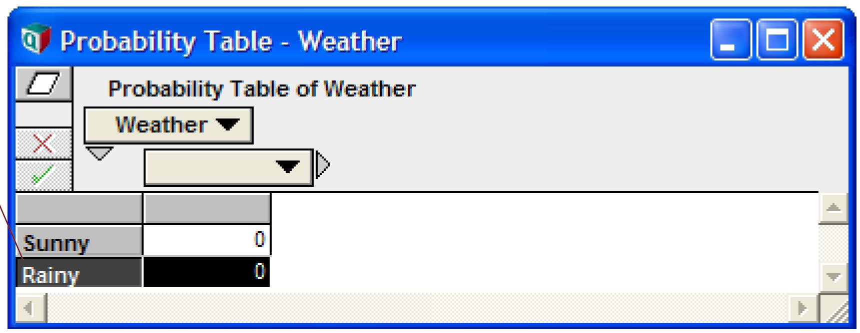
An **Edit Table** window appears. In this table, the first column holds the outcomes and the second column holds their probabilities. You will enter the possible outcomes in the first column.

5. Click in the cell labeled **Item 1** to select the first label for Weather. Type ***Sunny*** into the cell, replacing **item 1**, and press the *Enter* key.



A second row should be added to the table, containing the same label as the first row, *Sunny*. If a second row does not appear, make sure you have the text for *Sunny* selected, and press the *Enter* key again.

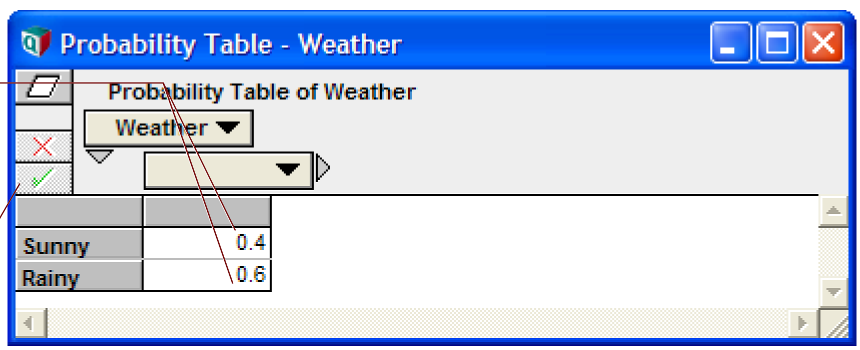
6. A second row is added to the table. Type ***Rainy*** and press *Alt+Enter* to accept the entries.

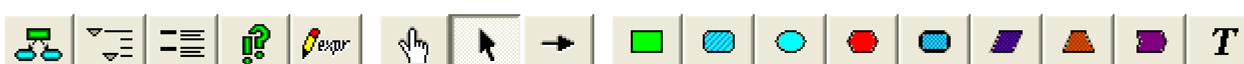


In the second column, you will enter the probabilities of the possible outcomes.

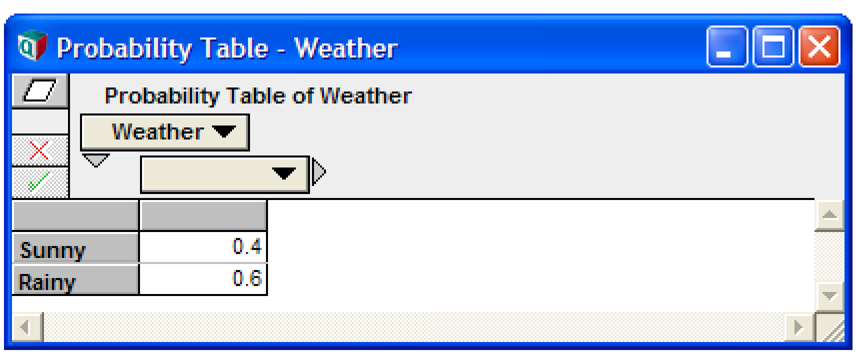
7. Enter the values ***0.4*** and ***0.6*** for the probabilities of sunshine and rain.

8. Click the check button to accept these entries.

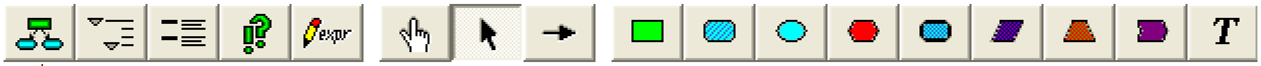




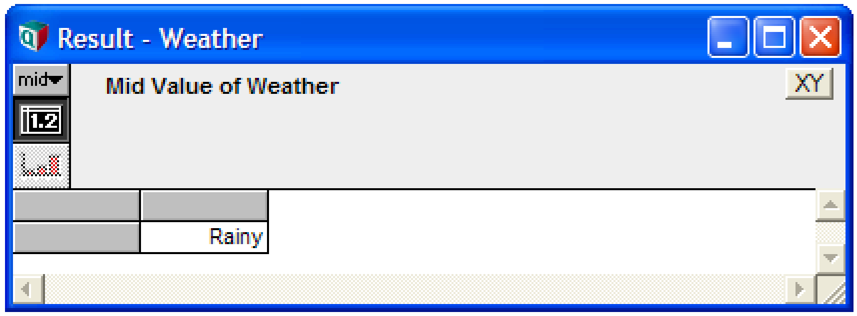
9. Click the **Result** button to evaluate *Weather*.



The mid value is *Rainy*, since more than 50% of the probability was assigned to *Rainy*.



10.Click the **Diagram** button to return to the **Diagram** window.



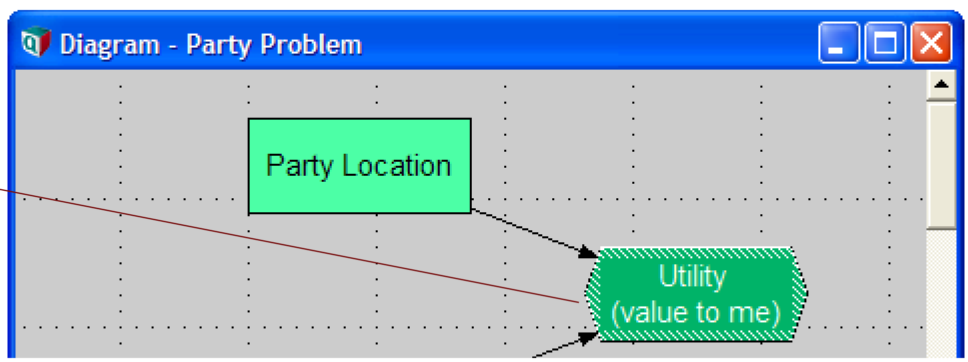
Defining Utility as a deterministic table

The utility of a party location decision depends on the outcome of the weather.

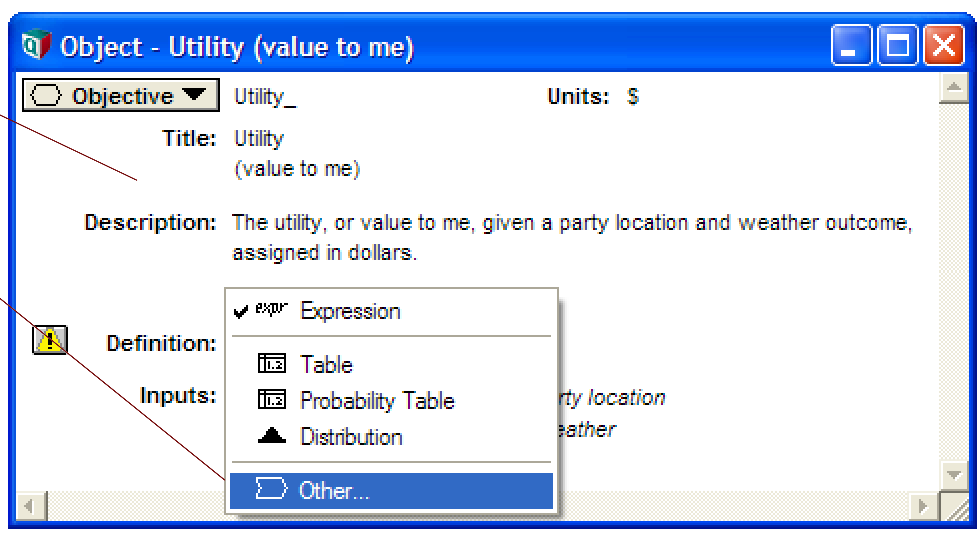
In this section, you will define *Utility* as a ***deterministic table*** (or ***determtable***) using both *Party Location* and *Weather*.

A deterministic table appears similar to an edit table or probability table. At least one index of a determtable must be a discrete probabilistic variable (probability table). The result of evaluating a deterministic table takes into account the probability distribution described by the probability table index.

1. Double-click *Utility* to open its **Object** window.



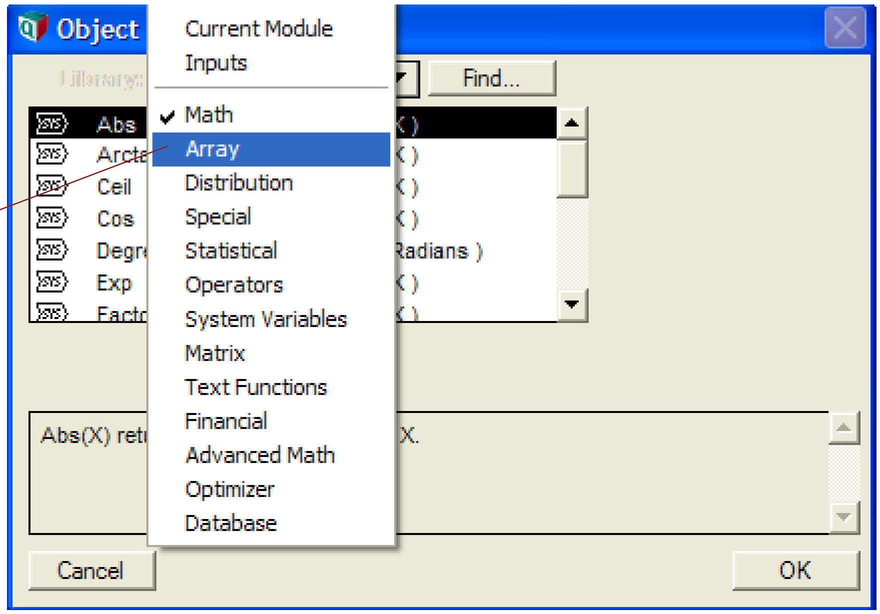
2. Enter the units and description as shown here.
3. Select **Other** from the Expression popup menu.



Because the *Utility* variable hasn't been defined yet, the first function in the first library is displayed. Libraries hold the functions available for you to use. For an overview of Analytica's libraries and their functions, see the section "Definition menu" in Appendix A of the *Analytica User Guide*.

You will select the **Determtable()** function, which is in the **Array** library.

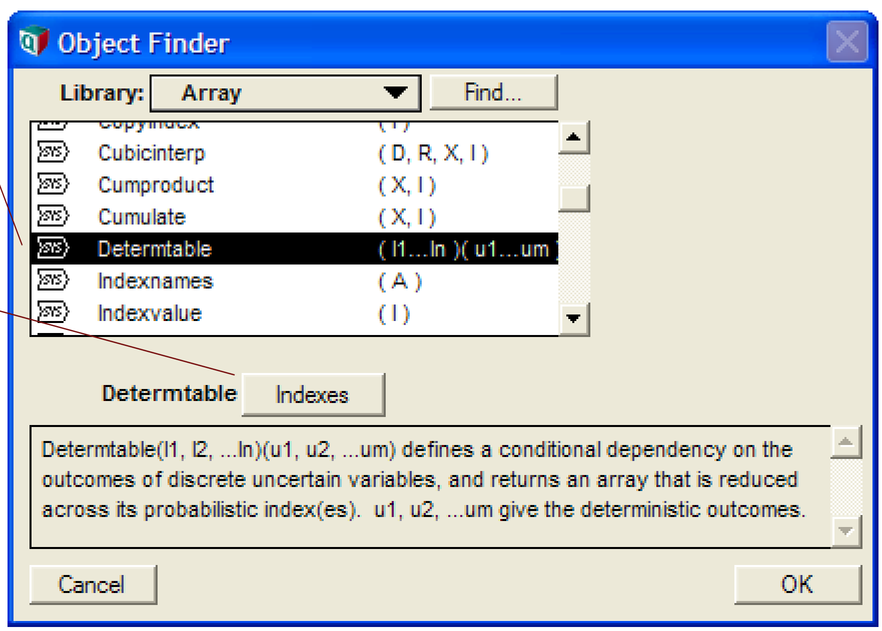
4. Select **Array** from the Library popup menu.



Tip The **Object Finder** window gives a brief description of the selected function.

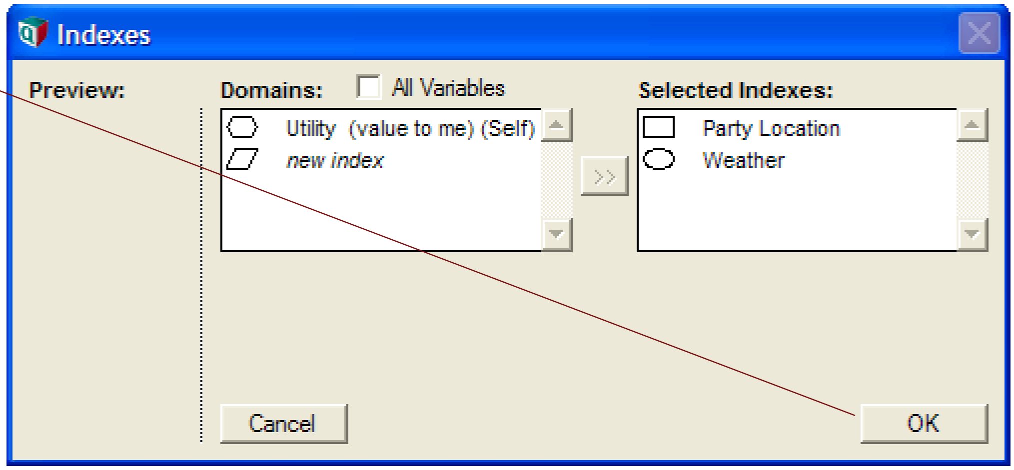
5. Scroll down the list and select **Determtable**.

6. To specify the indexes for the table, click the **Indexes** button.



Party Location and *Weather* are already selected as indexes because you drew arrows from them to *Utility* and they are defined as lists.

7. Click the **OK** button to accept these indexes.



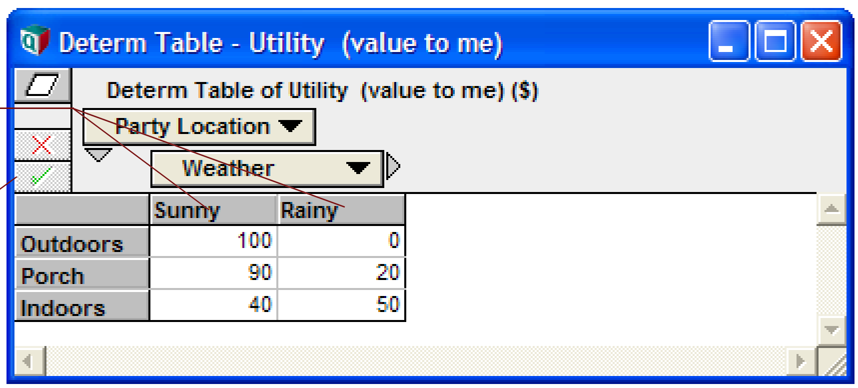
An **Edit Table** window opens for you to specify the dollar values for *Utility*.

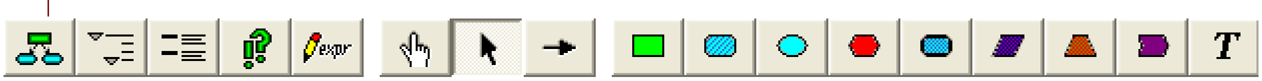
You will give a value for each combination of party location and weather. These values describe the amount of money it is worth to you if you use a given party location and the weather turns out a certain way. Nominally, it is most valuable to have the party outdoors if it is sunny; it is least valuable if it is rainy and you are outdoors. Other values fall between these extremes.

8. Enter the values shown in the cells.

9. Click the check button ✓ to accept these entries.

10. Click the **Diagram** button to return to the **Diagram** window.

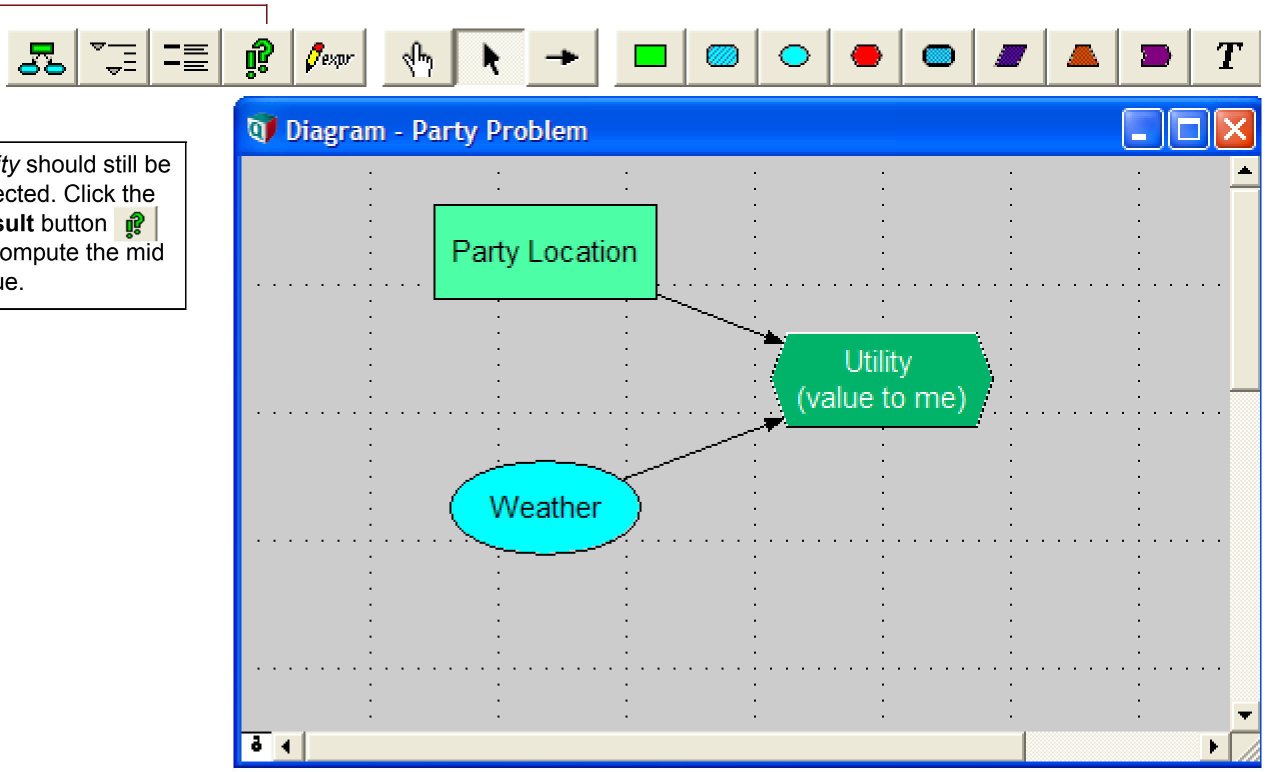




Computing Utility

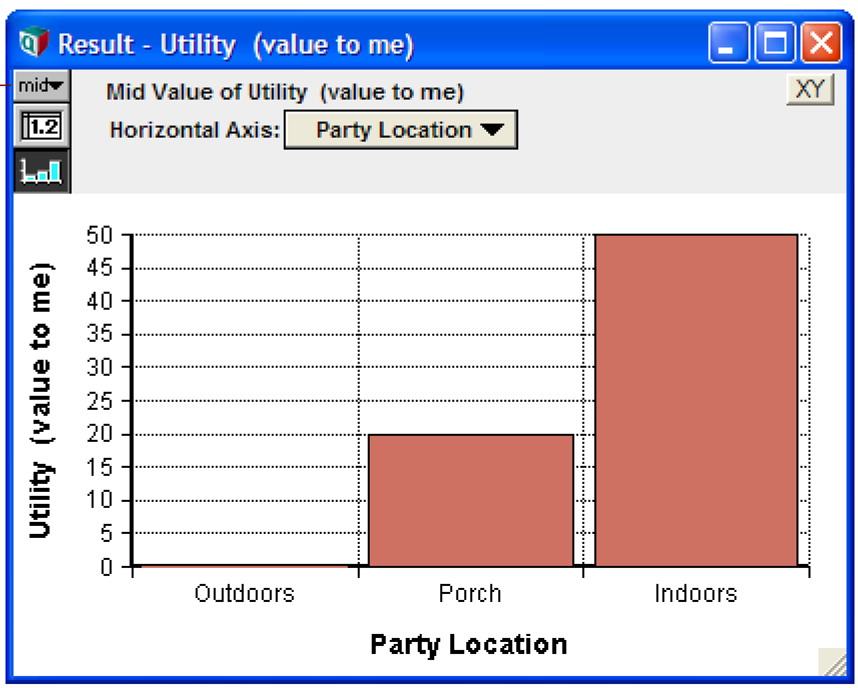
In this section, you will compute the value of *Utility*.

1. *Utility* should still be selected. Click the **Result** button to compute the mid value.

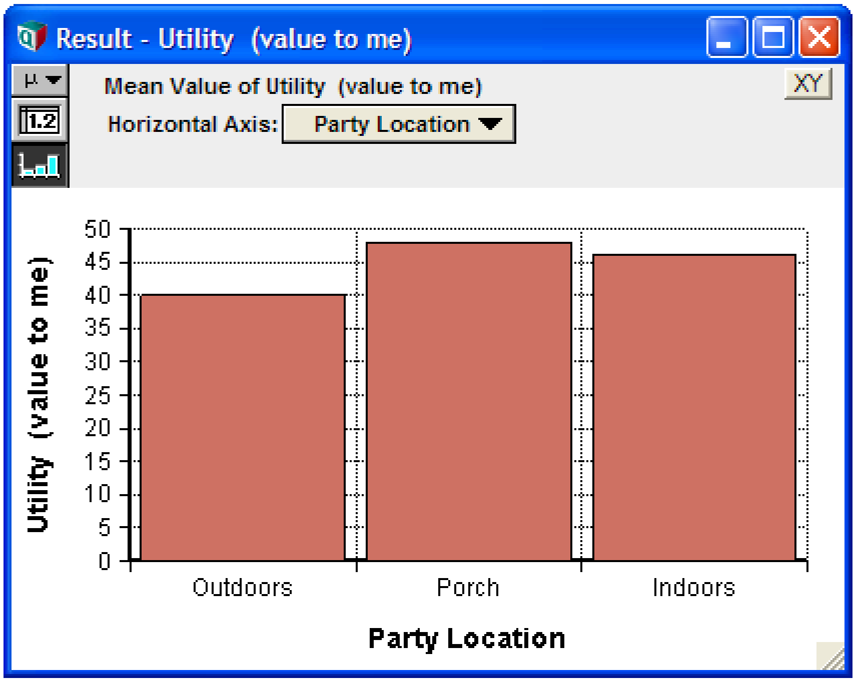


The utility of each party location corresponds to the values you just entered for *Rainy* in the *Utility* deterministic table, because the mid value of *Weather* is *Rainy*. With these outcomes, a deterministic assessment leads you to conclude that your best bet is to hold the party indoors.

2. Select **Mean Value** from the Uncertainty View popup menu.



The mean value gives you a very different assessment: the porch gives you the greatest expected utility. The mean value as estimated by the sample is approximately 40 for outdoors, 48 for the porch, and 46 for indoors.



The exact expected utility for each party location can be calculated by multiplying the probabilities of the outcomes by the values of *Utility*:

```
Outdoors   = (100)*0.4 + (0)*0.6   = 40
Porch      = (90)*0.4 + (20)*0.6   = 48
Indoors    = (40)*0.4 + (50)*0.6   = 46
```

Note to those with a background in discrete modeling

Analytica simulates all probability distributions and calculates the expected (mean) value of a distribution by computing the average of the samples. For a discrete distribution, the computed mean converges, with increasing sample size, toward the value obtained by multiplying the probabilities by the discrete outcome values.

Creating the Party Problem model: summary

In this chapter, you have:

- Defined a variable as a probability table, a discrete distribution.
- Defined a variable as a deterministic table, a function that defines a conditional dependency on the outcome of a discrete uncertain variable.

Exercise

As an exercise, extend the model to examine how the utility of each party location varies as the probability of rain varies from 0 to 100%.

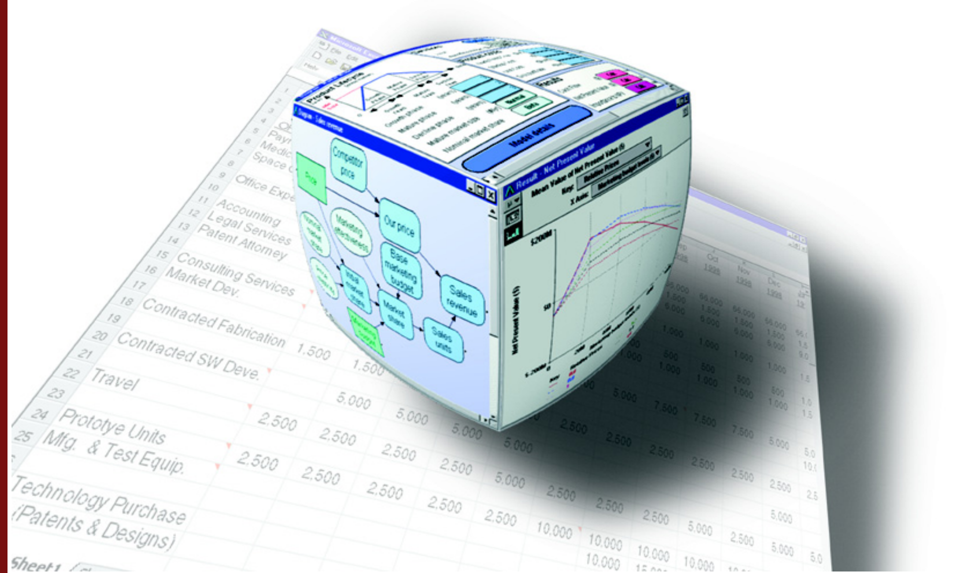
1. Create another chance node, titled `p`, the probability of sunshine. Define it as the range of probabilities from 0 to 100%, as `Sequence(0, 1, 0.5)`.
2. Redefine the probabilities for *Weather* as `p`, for sunny, and `(1-p)`, for rainy.
3. Recalculate the mean value of *Utility*. Display the result as a graph.


Result - Utility (value to me)
Mean Value of Utility (value to me)
Horizontal Axis: P
Key: Party Location
XY
Utility (value to me)
100
80
60
40
20
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
P
Party Location
Outdoors
Porch
Indoors

Chapter 7 *Creating the Foxes and Hares Model*

This chapter shows you how to:

- Use the **Dynamic()** function and the system variable *Time*
- Calculate the results of two nodes simultaneously
- Include variable names in definitions
- Use the **Min()** and **Max()** functions



In this chapter you will create a new Analytica model called *Foxes and Hares*. This model is more complex than those you built in the previous chapters, and introduces you to more advanced model-building techniques.

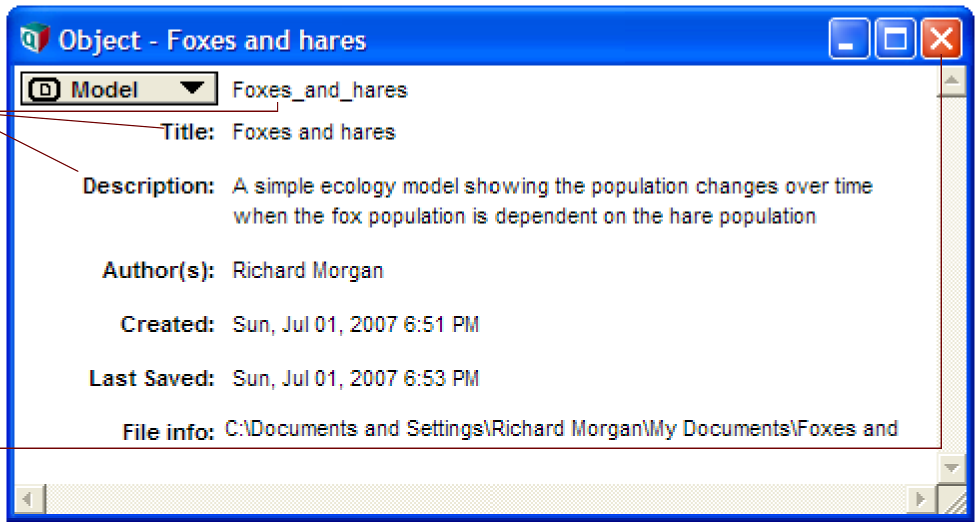
You should complete this chapter if your models use dynamic simulation or variables that change over time.

The Foxes and Hares models a simple ecology where the populations of two species, foxes and hares, are dependent on each other over time.

Documenting the model

Document the model using the information specified in the **Object** window below.

1. Enter the model's Identifier, Title, and Description.
2. Save the model by selecting **Save** from the **File** menu.
3. Click the close button to close the model's **Object** window and bring the **Diagram** window to the front.



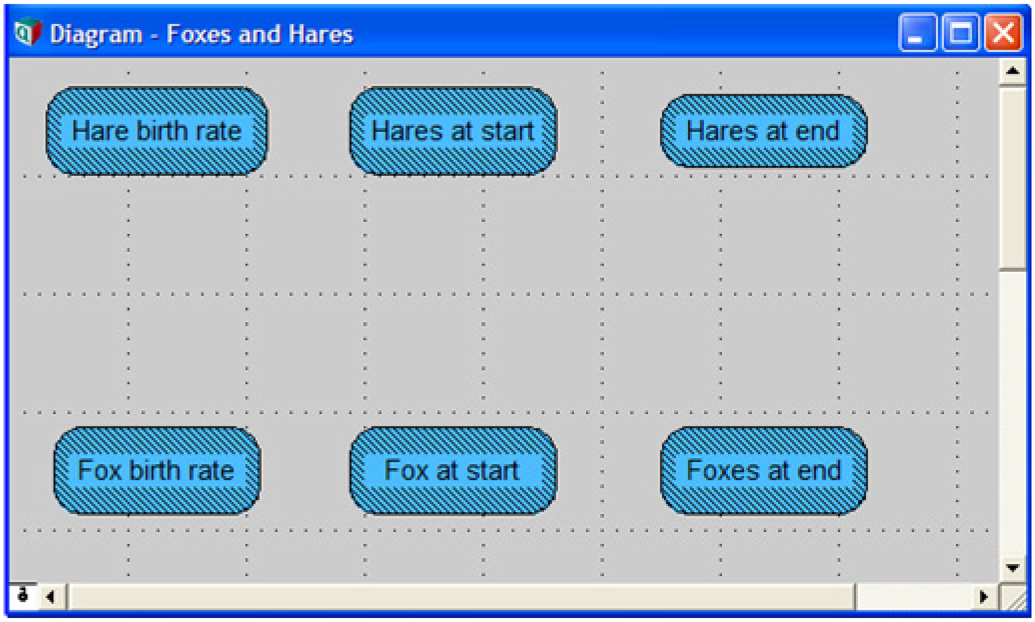
Creating the foxes and hares diagram

To create a model of the fox and hare populations, you will make a number of simplifying assumptions.

Start by assuming a number of time periods. Each time period has the same behavior:

- At the start of the time period, each population grows (from births).
- In the middle of the time period, the foxes prey on the hares.
- At the end of the time period, you tally the populations.

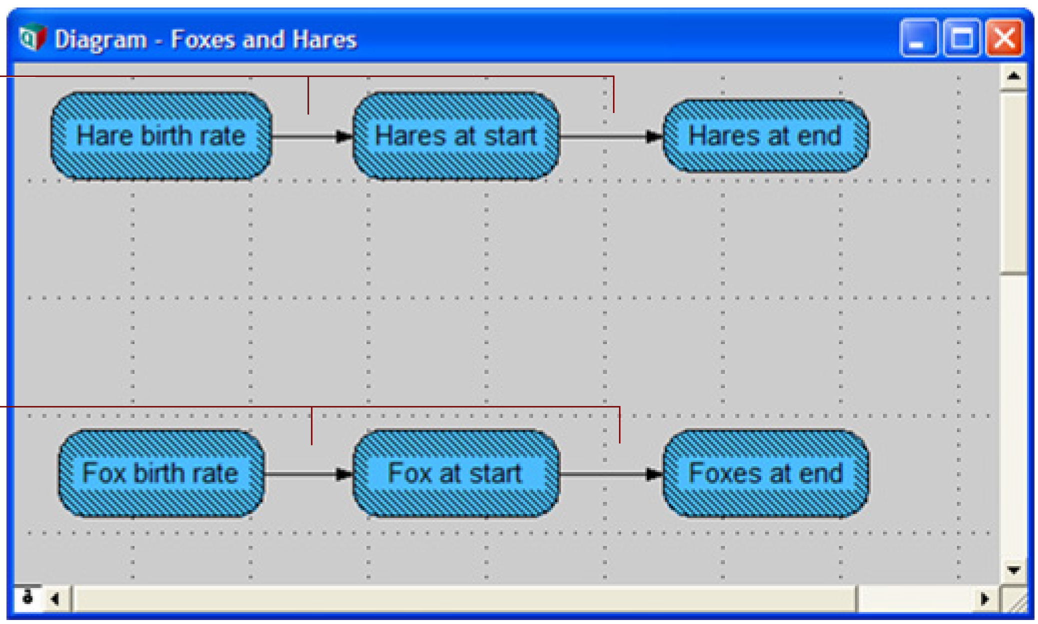
1. Create six general variable nodes and title them ***Hare birth rate***, ***Hares at start***, ***Hares at end***, ***Fox birth rate***, ***Fox at start***, and ***Foxes at end***.



2. Draw arrows from *Hare birth rate* to *Hares at start*, and from *Hares at start* to *Hares at end*.

3. Draw arrows from *Fox birth rate* to *Fox at start*, and from *Fox at start* to *Foxes at end*.

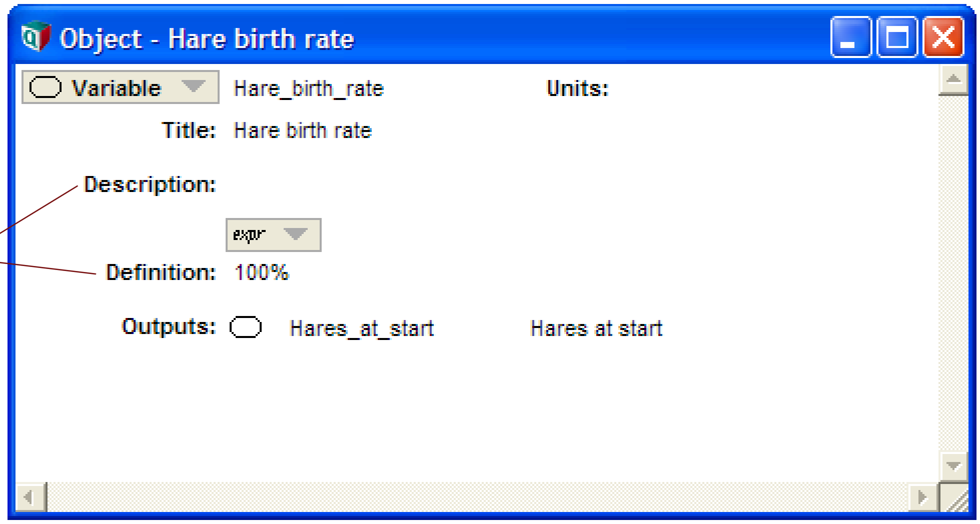
   Your diagram should now appear as shown here.



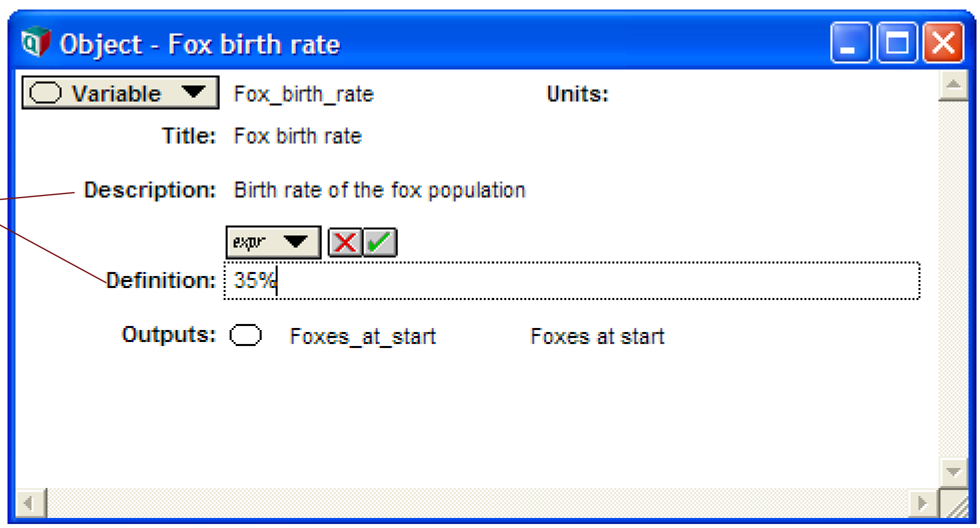
Defining hare birth rate and fox birth rate

Your model assumes that the hare population, left unchecked, doubles in each time period, and the fox population, given an adequate food supply (hares), grows by 35% in each time period.

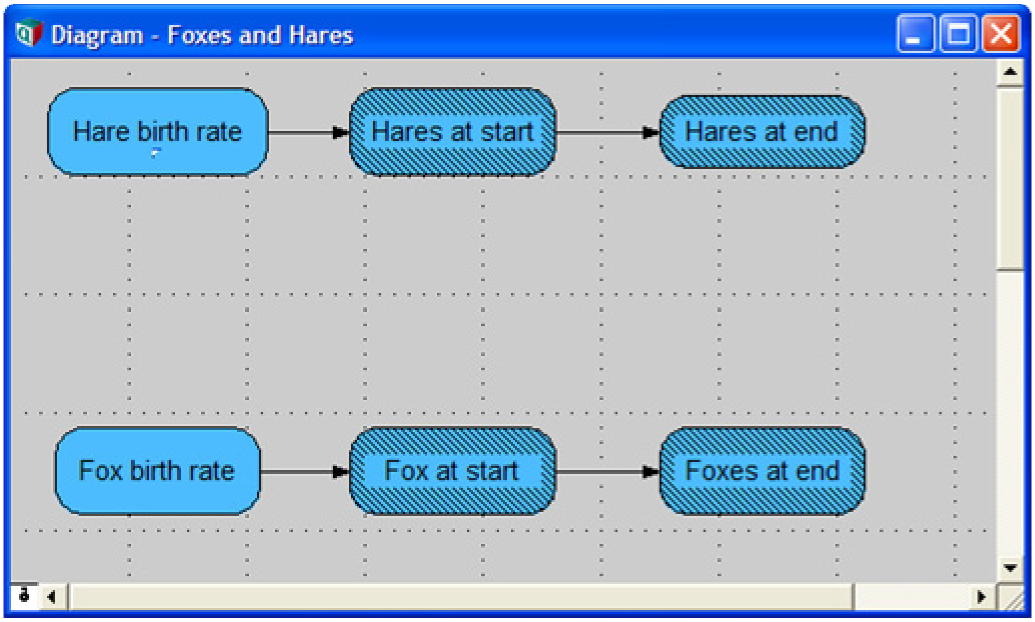
1. Double-click *Hare birth rate* to open its **Object** window.
2. Enter the Description and Definition as shown here.
3. Close the **Object** window to return to the **Diagram** window.



4. Double-click *Fox birth rate* to open its **Object** window.
5. Enter the Description and Definition as shown here.
6. Close the **Object** window to return to the model diagram.



Note that both the *Hare birth rate* and *Fox birth rate* nodes are solid, indicating that their definitions are valid.



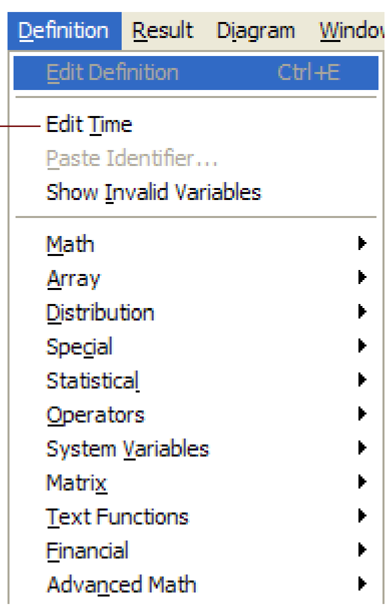
Defining the Time variable

Next, you will investigate how the hare and fox populations grow over time if they are simply allowed to reproduce without any controls. You will start with a time horizon of 10 units.

Analytica has a special built-in variable, `Time`, for use in models such as this one. You will define `Time` as a list: `[1,2,3,4,5,6,7,8,9,10]`. The simplest way to define this list is by using the **Sequence()** function.

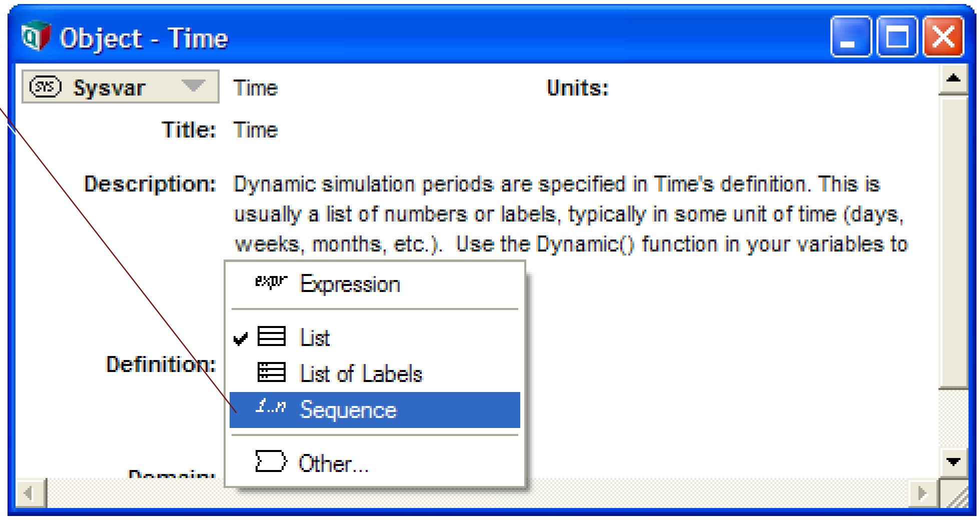
1. Select **Edit Time** from the **Definition** menu.

   The **Object** window for *Time* displays.



2. Select **Sequence** from the Expressions popup menu.

3. Click the **OK** button to replace the existing definition with a sequence.



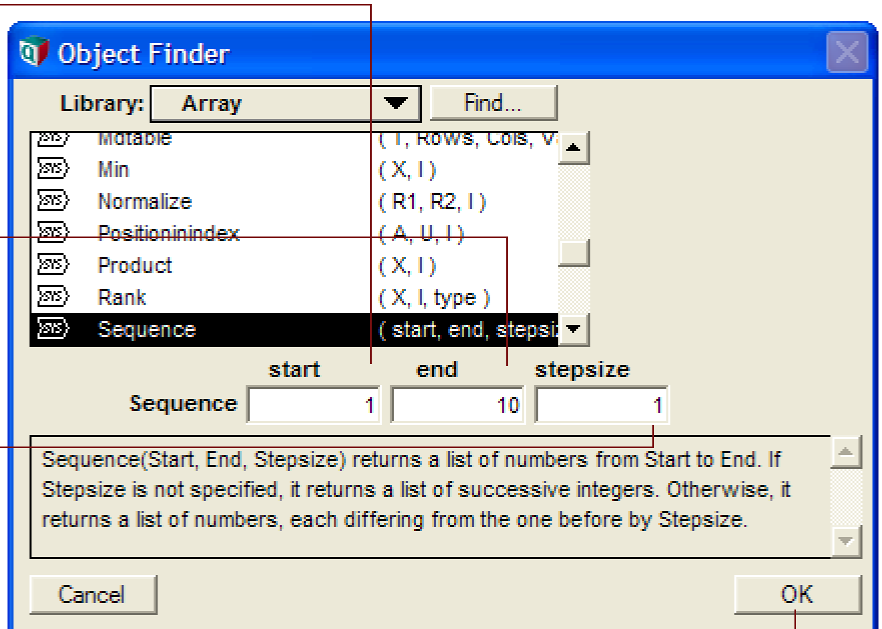
The parameter fields for the **Sequence()** function contain the starting number, the ending number, and the step size.

4. Enter ***1*** (to start at time 1) in the first Sequence parameter field, **start**.

5. Tab to the next field, **end**, and enter ***10*** (to end at time 10).

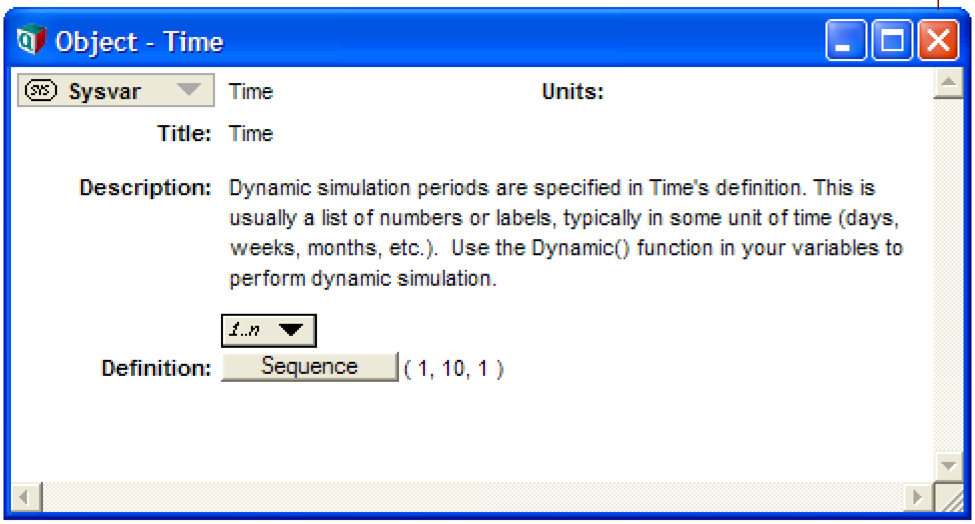
6. Tab to the third field, **stepsize**, and enter **1** (to increment time by 1 unit for each calculation).

7. Click the **OK** button to save your entries.



The **Object** window for **`Time`** shows that the definition is now a sequence from 1 to 10.

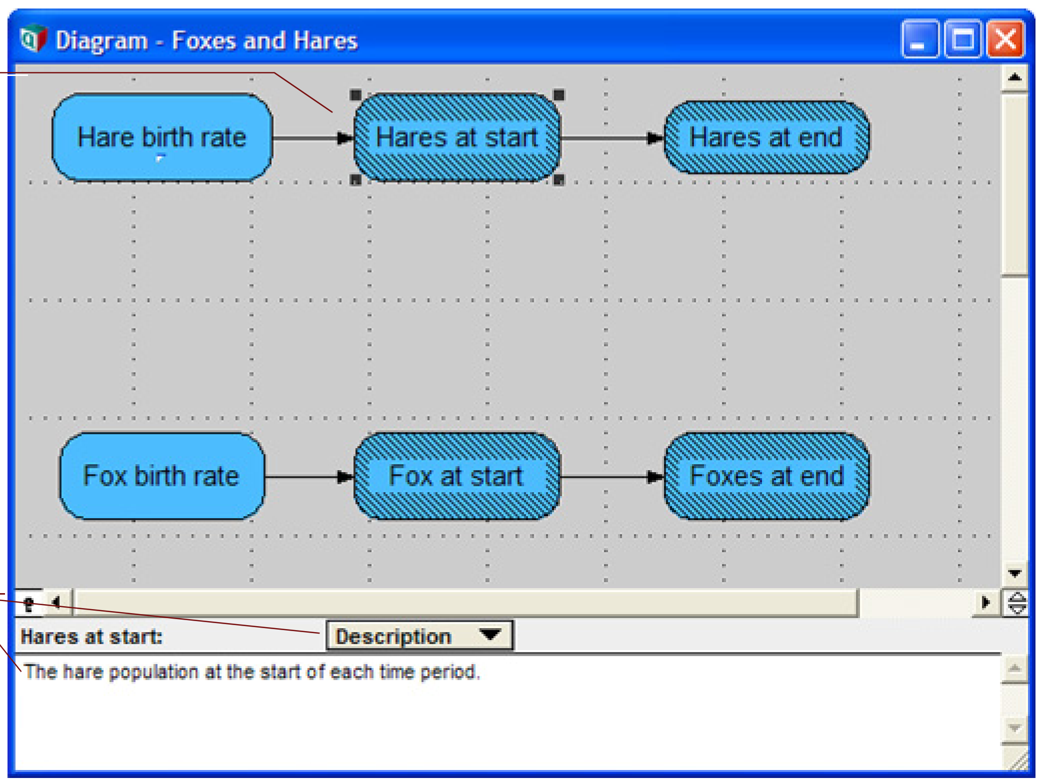
8. Click the close button to close the *Time* **Object** window and return to the **Diagram** window.



Defining hare population as a function of Time

Next, you will define the initial hare population as 100 and the population at the start of each subsequent time period grows based on the value of *Hare birth rate*.

1. Select *Hares at start*.
2. Click the key icon to open the diagram's **Attribute** panel.
3. Click in the Description field and enter the text as shown here.
4. Press *Alt+Enter* to save your entry.



You will define the population at each time period to be a function of its size at the end of the previous time period. A special system function, **Dynamic()**, is used to perform this calculation; it calculates a value for each unit of `Time`.

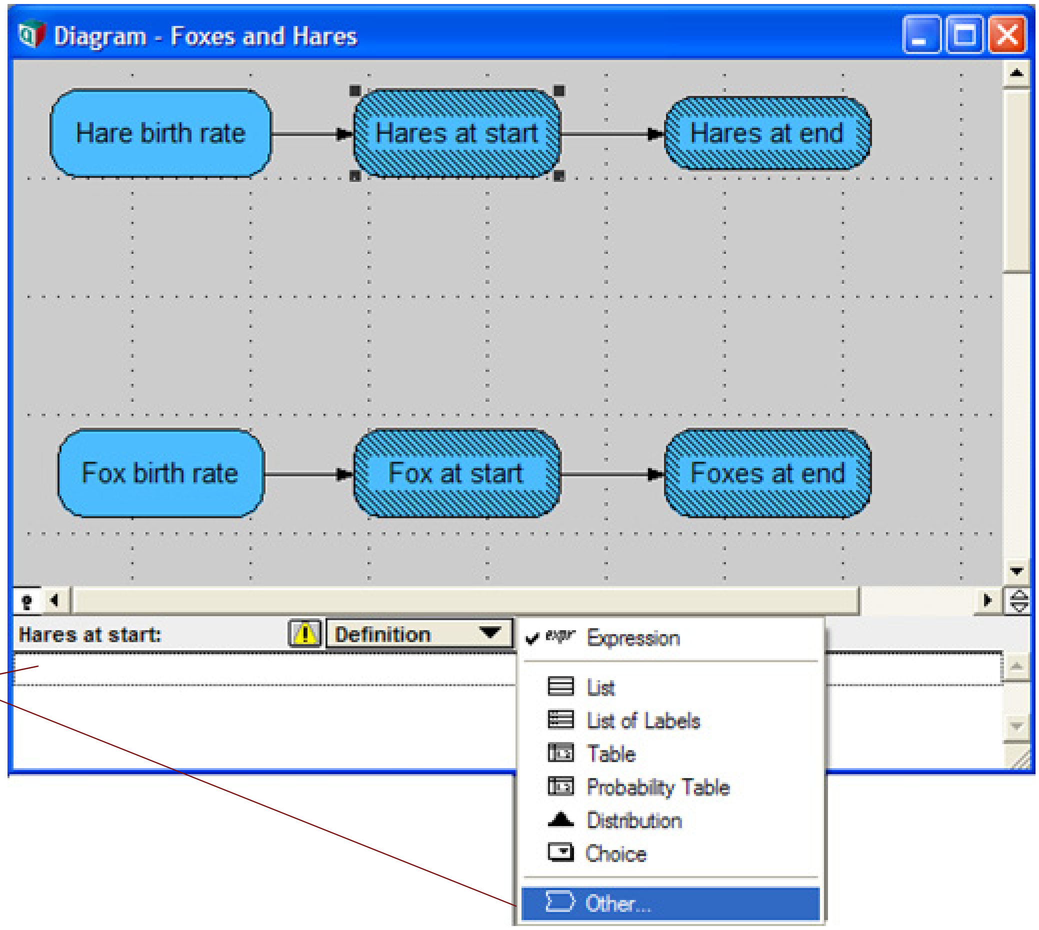
Dynamic() is the only function in Analytica that permits cyclic dependency. In other words, the **Dynamic()** function allows you to refer to the variable that it defines or to other dynamic variables at earlier time periods. The **Dynamic()** function must appear at the topmost level of a definition; it cannot be used inside another expression. The syntax for **Dynamic()** requires an initial value for the variable to which it is being applied, i.e., the value of the variable at the first unit of `Time`.

To define *Hares at start* with **Dynamic()**, you will use the **Object Finder** window.

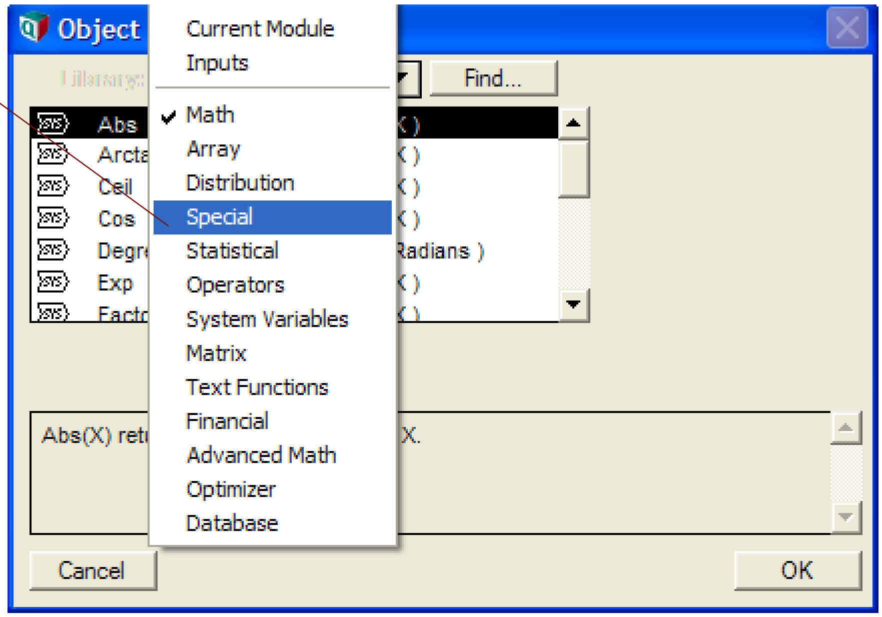
5. Select **Definition** from the Attribute popup menu.

6. Click in the text field, and select **Other** from the Expressions popup menu.

   The **Object Finder** window is displayed.



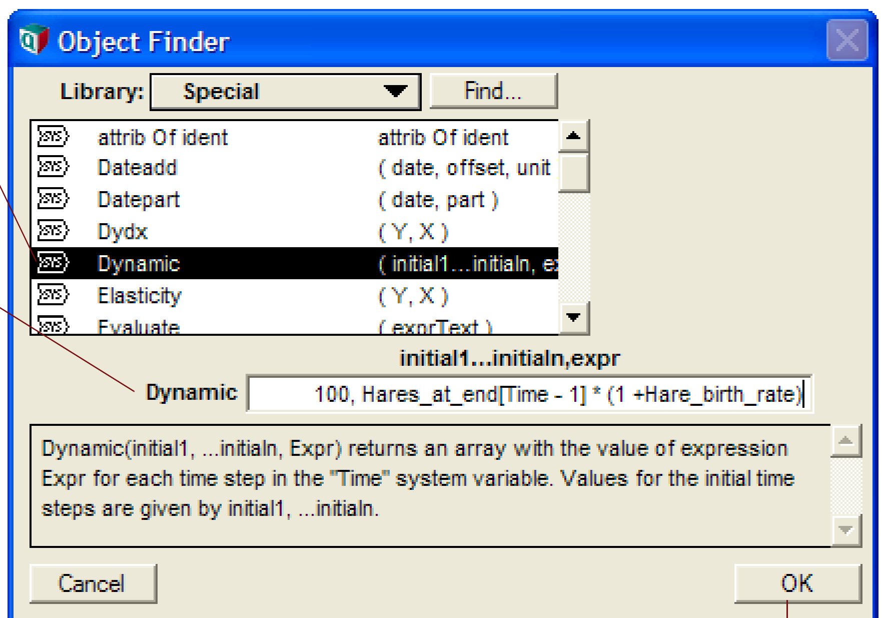
7. Select **Special** from the Library popup menu.



8. Click **Dynamic** to see the description and parameter box for the **Dynamic()** function.

9. Click in the Dynamic field and type ***100, Hares_at_end [Time - 1] * (1 + Hare_birth_rate)***.

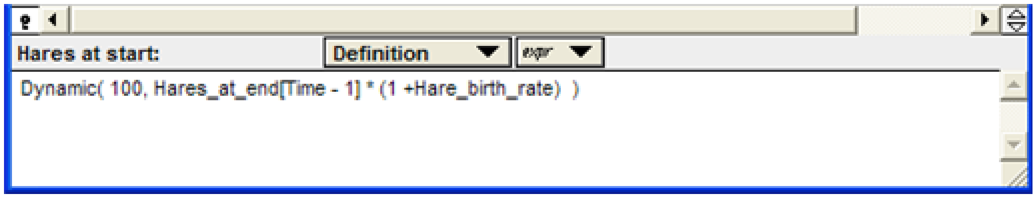
10. Click the **OK** button (or press *Alt+Enter*) to close the window.



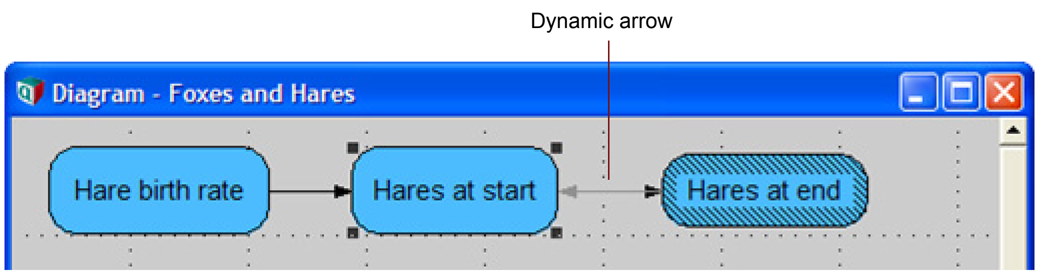
Tip Square brackets are necessary for **`[Time-1]`**. Be sure to use parentheses around other expressions in **Dynamic()**.

The **Attribute** panel shows the **Dynamic()** function in the Definition field. This definition states that the starting population of hares is 100 at the first time period; at the start of all other time periods, the population is equal to the population at the end of the previous time period **`(Time-1)`** plus the number of hares born.

11.Press *Alt+Enter* to accept the definition.



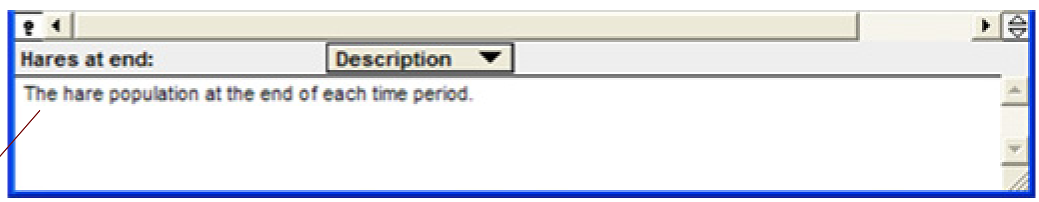
Tip Note that a gray arrowhead points from *Hares at end* to *Hares at start*. The gray arrow is the diagram symbol for a dynamic dependency — that is, the *Hares at end* input is from an earlier time period.



You will now define the population at the end of each period to be equal to the population at the start of each period. (You will later modify this definition to subtract the hares being captured during the period.)

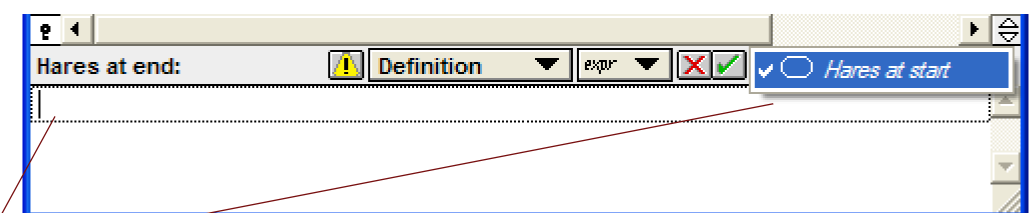
12.Select *Hares at end*.

13.Select **Description** from the Attribute popup menu, click in the text field, and enter the Description as shown here.



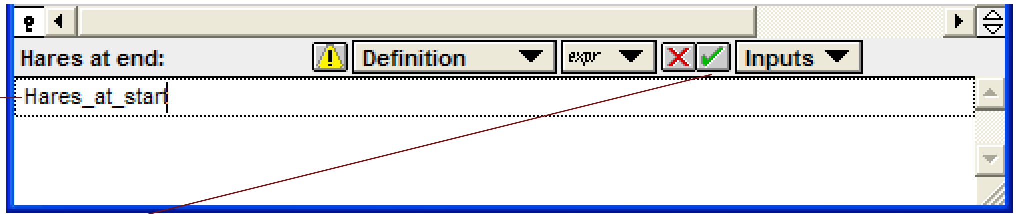
14.Select **Definition** from the Attribute popup menu.

15.Click in the Definition field; then select *Hares at start* from the Inputs popup menu.



16.**Hares_at_start** now appears in the Definition text field.

17.Click the check button (or press *Alt+Enter*) to accept your definition.

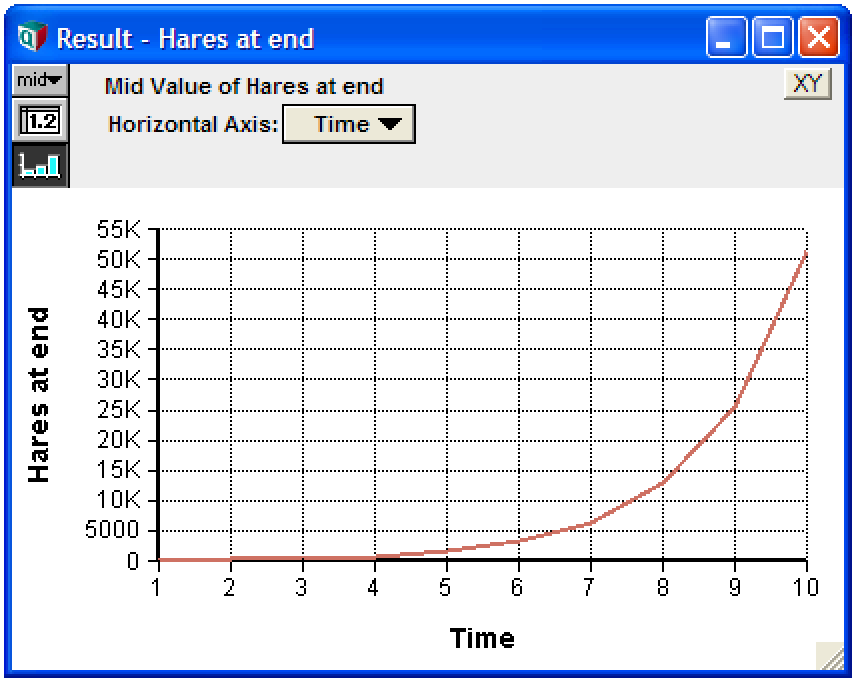


Next, you will look at the result to see the hare population over time.

18.With the *Hares at end* node selected, click the **Result** button .

You can see the population explosion after only 10 time periods!

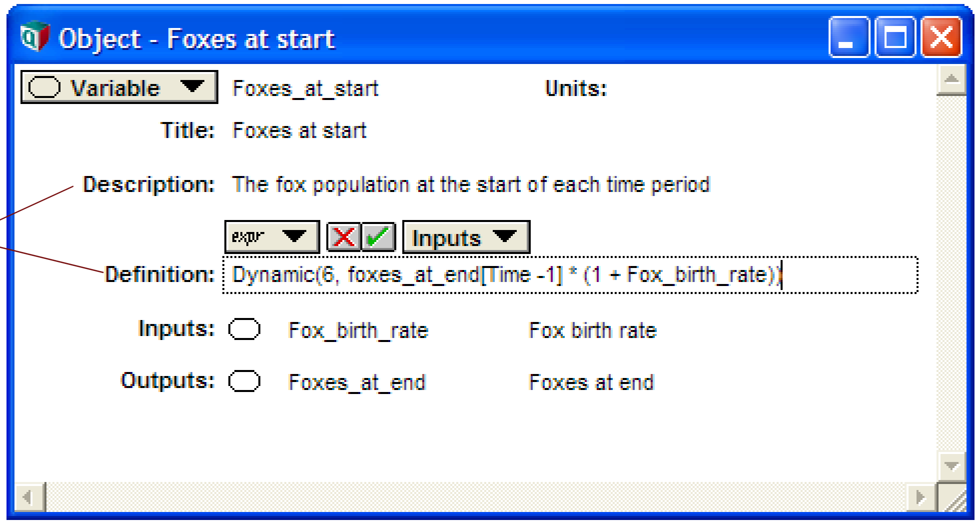
19.Return to the **Diagram** window by closing the **Result** window or clicking the **Diagram** window.



Defining the fox population as a function of Time

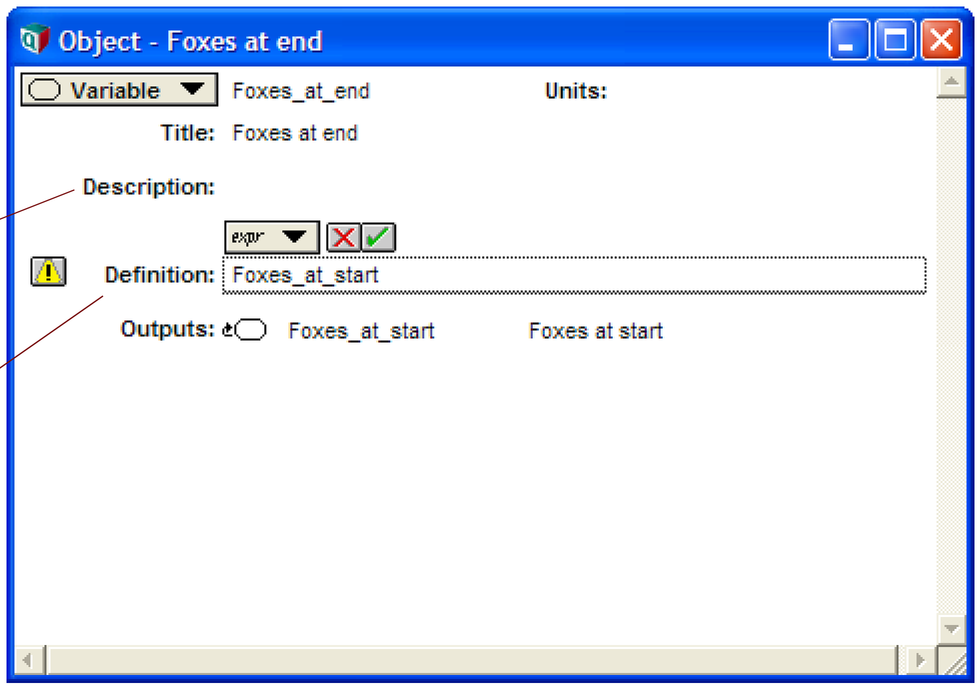
You will follow similar steps to define the fox population. Assume the starting number of foxes is 6. Instead of using the **Object Finder**, you can type directly in the Definition field.

1. Double-click *Foxes at start* to open its **Object** window.
2. Enter the Description and Definition as shown here.
3. Return to the **Diagram** window.



4. Double-click *Foxes at end* to open its **Object** window.
5. Enter the Description as shown here.
6. Enter the Definition using the Inputs popup menu.

   Foxes_at_start now appears in the Definition field.
7. Return to the **Diagram** window.

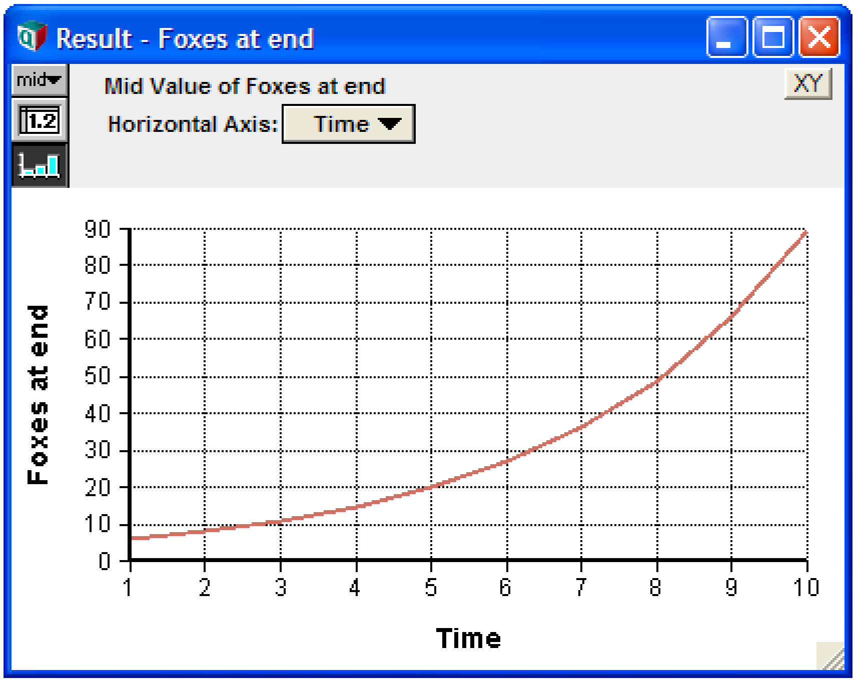


Next, you will look at the result to see the fox population over time.

8. With the *Foxes at end* node selected, click the **Result** button.

   You can see that the fox population also explodes, but grows more slowly than the hare population.

9. Return to the **Diagram** window.

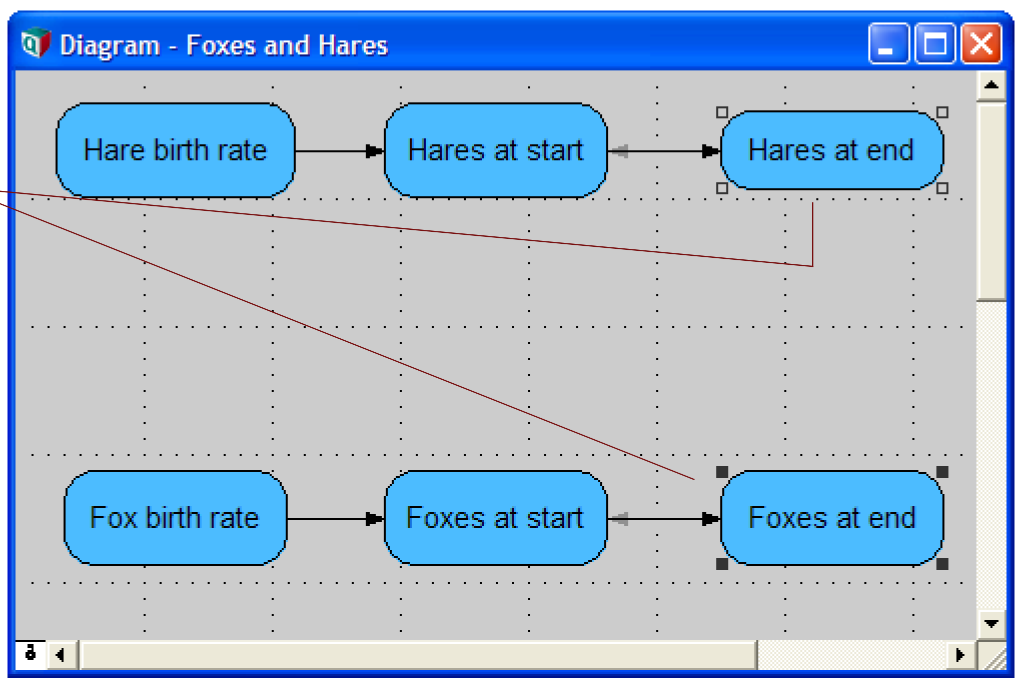


Creating the Populations objective

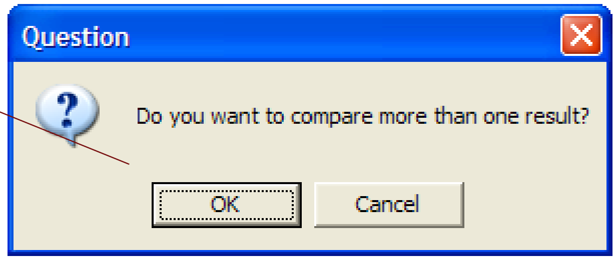
To directly compare the two populations, you will view both populations on one graph. You will create a new variable to show both population results.

1. Select both *Hares at end* and *Foxes at end.*

2. Click the **Result** button .



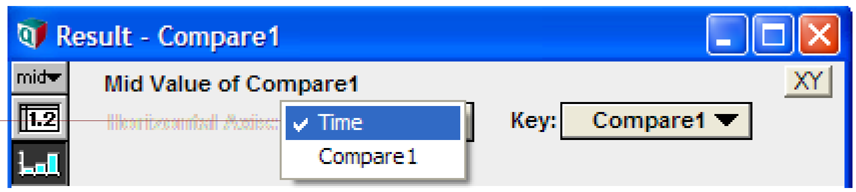
3. Because you want to compare the two results, click the **OK** button (or press *Alt+Enter*).



Analytica creates a new node with a dummy name. You will change the name and move it to a better position on the diagram in just a minute. But first, you will view the resulting graph.

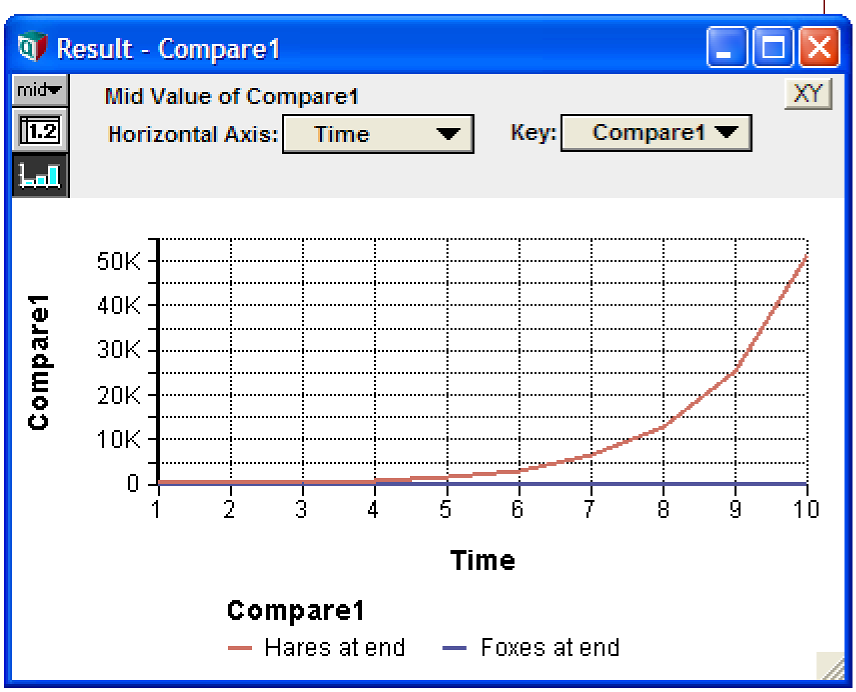
The default graph view is not useful, so you will change the Horizontal axis to `Time` in order to see the two animal populations over time.

4. Select **Time** from the Horizontal axis popup menu.



Now the population explosions of both the hares and the foxes appear on the same graph. Note that the number of foxes is very small compared to the number of hares and is visible on this graph as a line on the horizontal axis.

5. Return to the **Diagram** window.

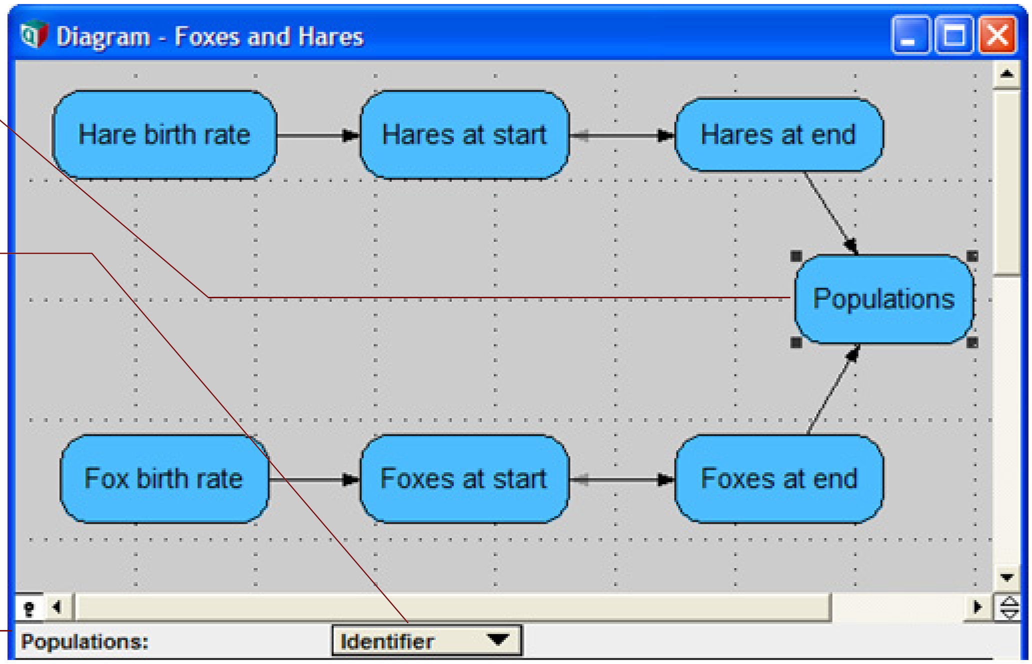


Next, you will adjust the position, name, and class of the new node.

6. Move the new node to the position shown in the figure.

7. In the **Attribute** panel, change the identifier to ***Populations***, replacing the dummy name.

8. Press *Alt+Enter* to save the change.

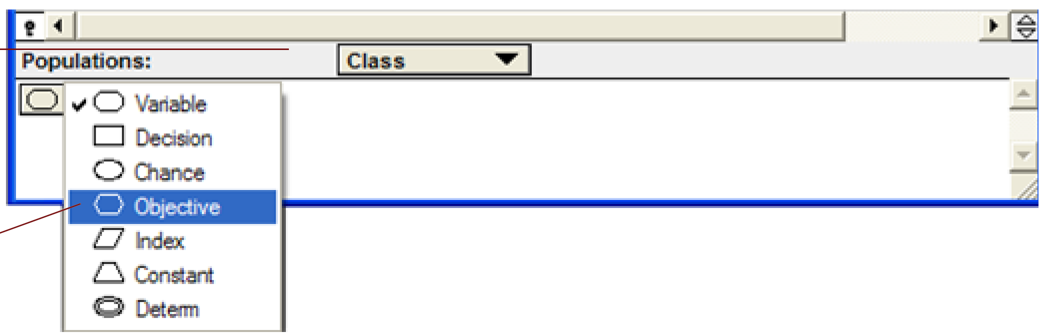


Note that the title also changes.

Since this node is the final output of the model, you will change its class to objective.

9. Select the **Class** attribute from the Attribute popup menu.

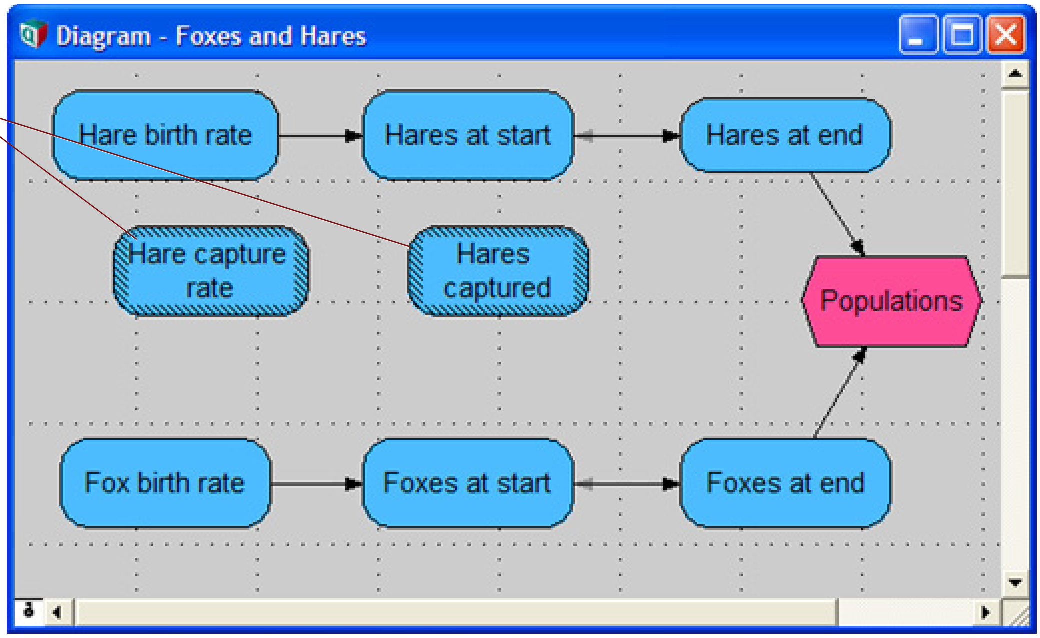
10. From the Class popup menu, select **Objective**.



Defining population control: foxes capture hares

In this section, you will define the population control for the hares, namely that, during each time period, some are captured by foxes. You will assume that the foxes hunt independently from each other and that each fox captures 25% of the hares during a time period.

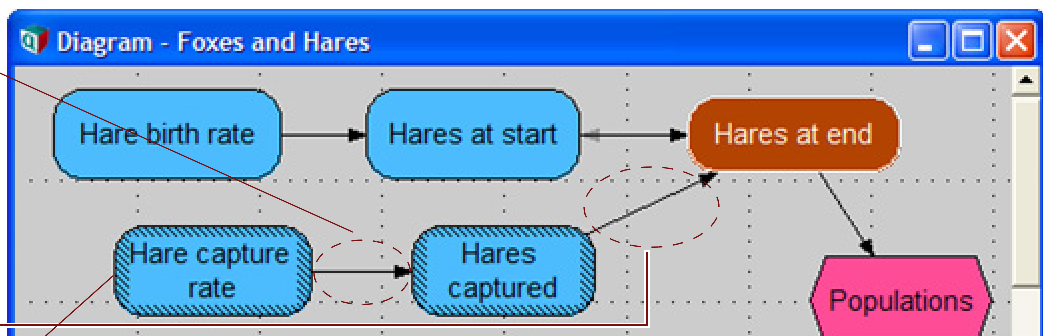
1. Create two general variable nodes and title them ***Hare capture rate*** and ***Hares captured***.



2. Draw an arrow from *Hare capture rate* to *Hares captured*.

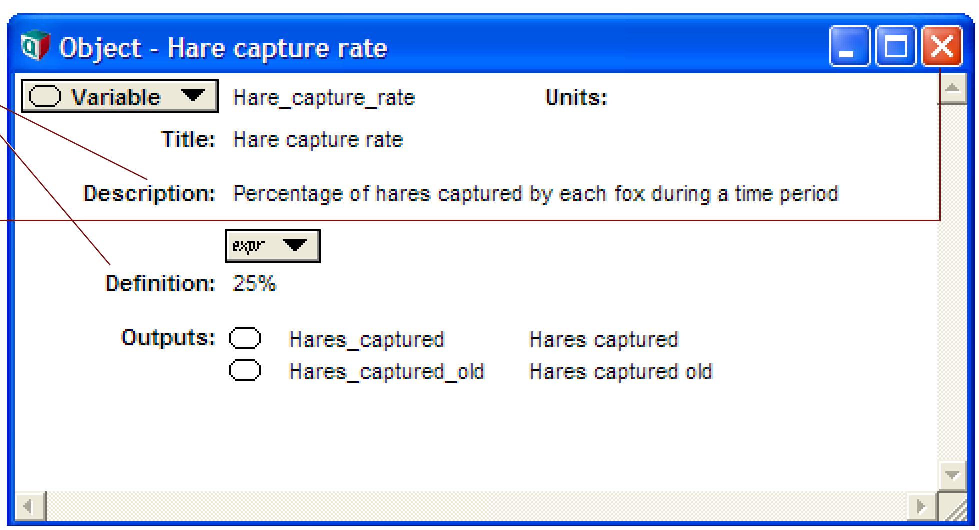
3. Draw an arrow from *Hares captured* to *Hares at end*.

4. Double-click *Hare capture rate* to open its **Object** window.

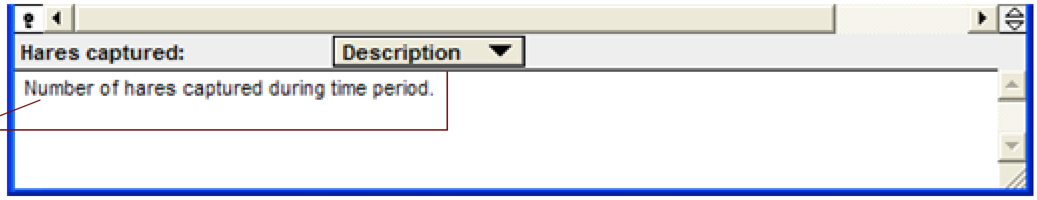


5. Enter the description and definition as shown here.

6. Close the **Object** window.



7. Select *Hares captured* and, with *Description* selected in the Attribute popup menu, enter the Description as shown here.

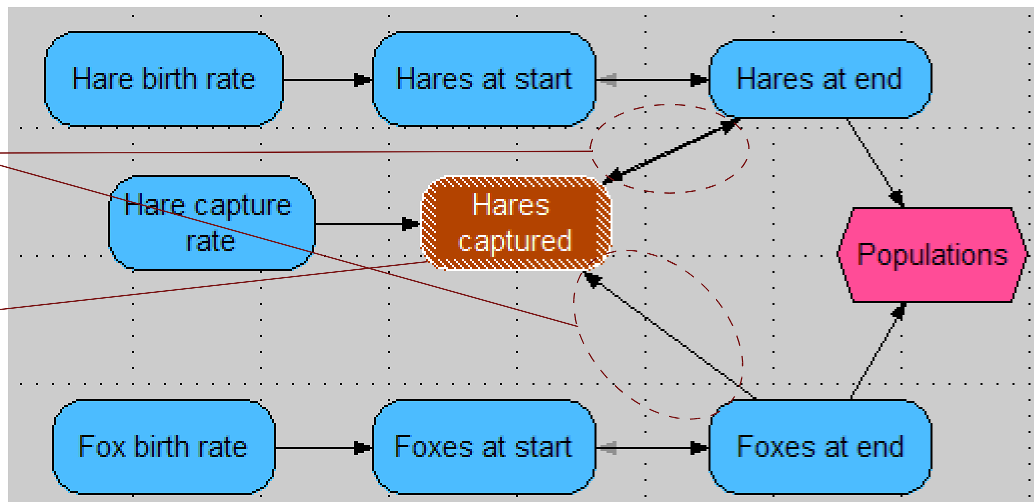


If only adult hares are captured and only adult foxes are captors, then, in a time period, the number of hares captured is the *Foxes at end* of the previous period times *Hare capture rate* times *Hares at end* of the previous period[3].

You will again use the **Dynamic()** function, using an initial value of 75 hares captured. This time you will enter the dynamic expression directly into the Definition field.

8. Draw arrows from both *Hares at end* and *Foxes at end* to *Hares captured.*

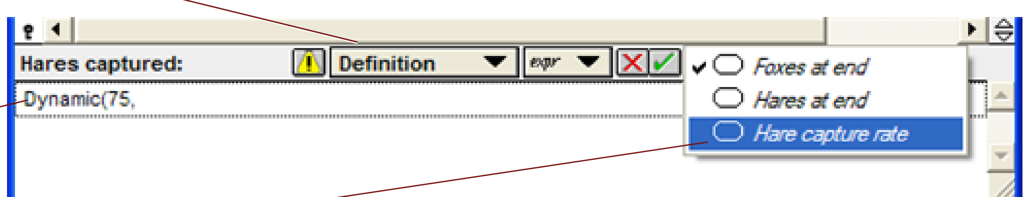
9. Select *Hares captured.*



10. Select **Definition** from the Attribute popup menu and click in the Definition text field.

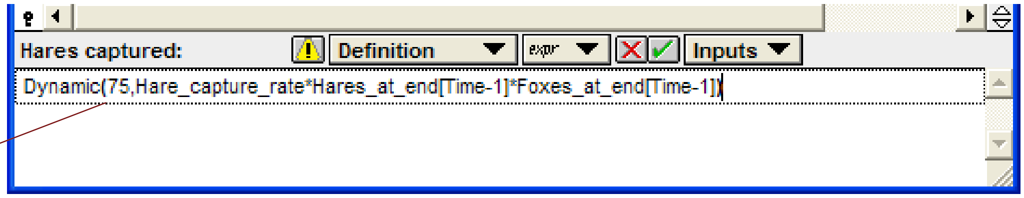
11. Type ***Dynamic(75,***

12. Select *Hare capture rate* from the Input popup menu.



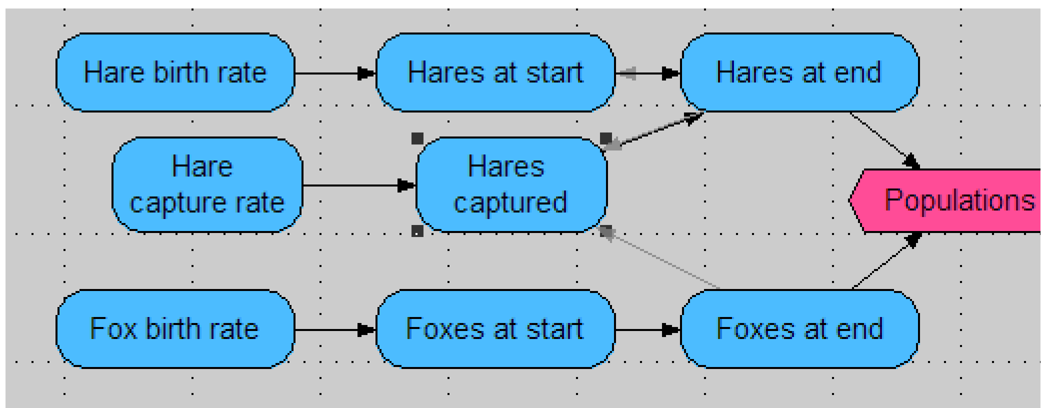
3. We are assuming that both foxes and hares are born at the start of the time period and it takes exactly one time period for them to mature. These simplifications can readily be removed when you are comfortable with how the **Dynamic()** function works.

13. Continue editing the definition of *Hares captured* by typing and by selecting inputs from the Input popup menu until it is as shown here.

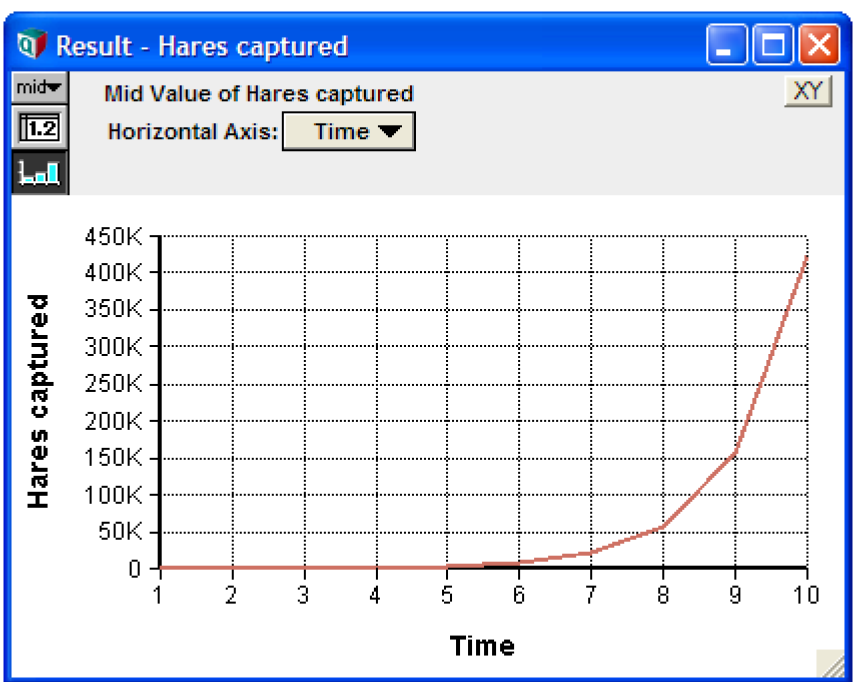


14. Click the check button (or press *Alt+Enter*) to accept your definition.

Note that the arrows you drew in Step 8 above are now gray lines. These lines indicate that the dependency is dynamic — the input variable is from an earlier time period.



15. Click the **Result** button to see the number of hares captured as a function of time.



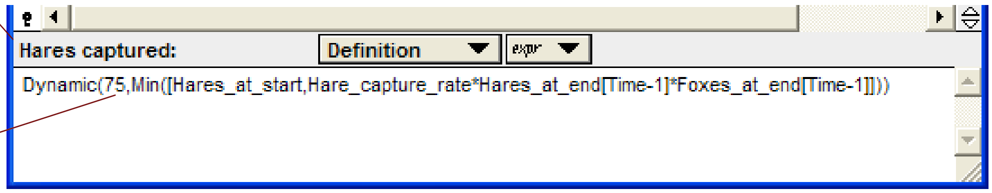
Something is not quite right. These values for *Hares captured* grow to be larger than the number of hares at the start of the period (compare the graph of *Hares at end*, which equaled *Hares at start*, on page 145). This is not reasonable; for example, if there are more than four foxes, it is possible for the calculation to give more than 100% of the hares as captured.

To prevent the number of hares captured from exceeding the number of hares at the start of the period, you will constrain the definition of *Hares captured* by using the **Min()** function, so that no more hares are captured than exist at the start of the time period.

The **Min()** function selects the minimum value from an array of numbers. The syntax for finding the minimum of two numbers **x** and **y** is **Min([x,y])**. (The expression **[x,y]** creates an array from **x** and **y**.) You want the number of hares captured to always be the minimum of the calculated number and the number of hares at the start of the period.

16. Select *Hares captured.*

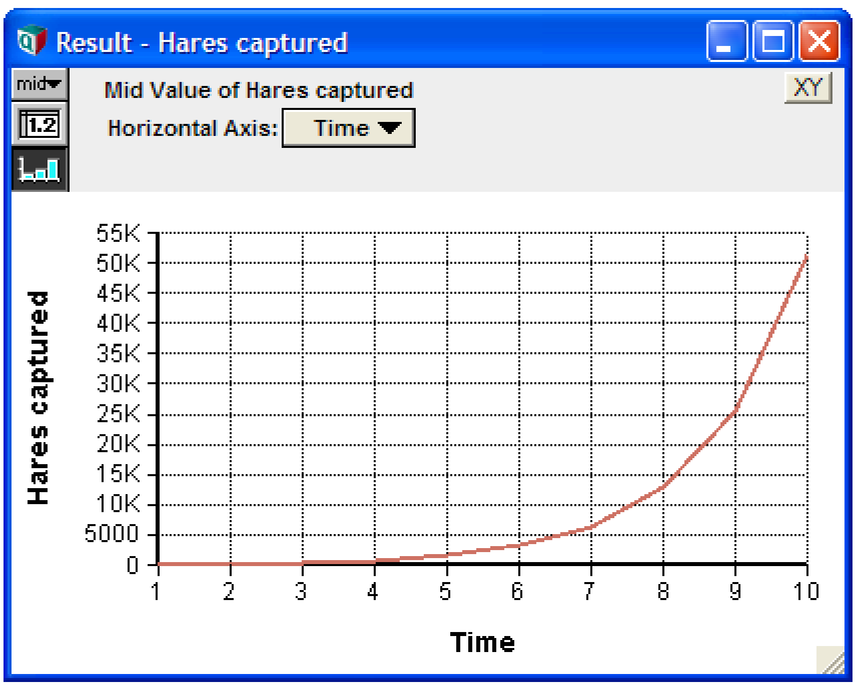
17. In the **Attribute** panel, edit the definition of *Hares captured* until it appears as shown here.



18. Press *Alt+Enter* to accept the revised definition.

19. Click the **Result** button to see the number of hares captured as a function of time.

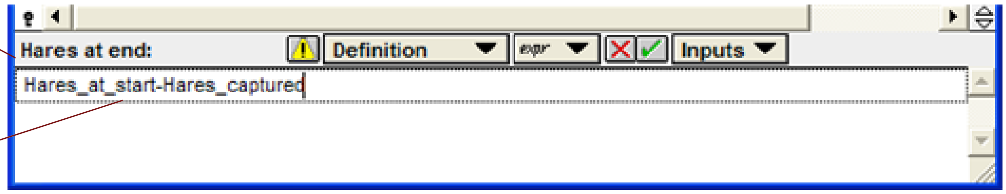
    Now the number captured is no greater than the number at the start of the period.



20. Select *Hares at end* in the model diagram.

Next, you will reduce the number of hares at the end of the time period by the number that are captured.

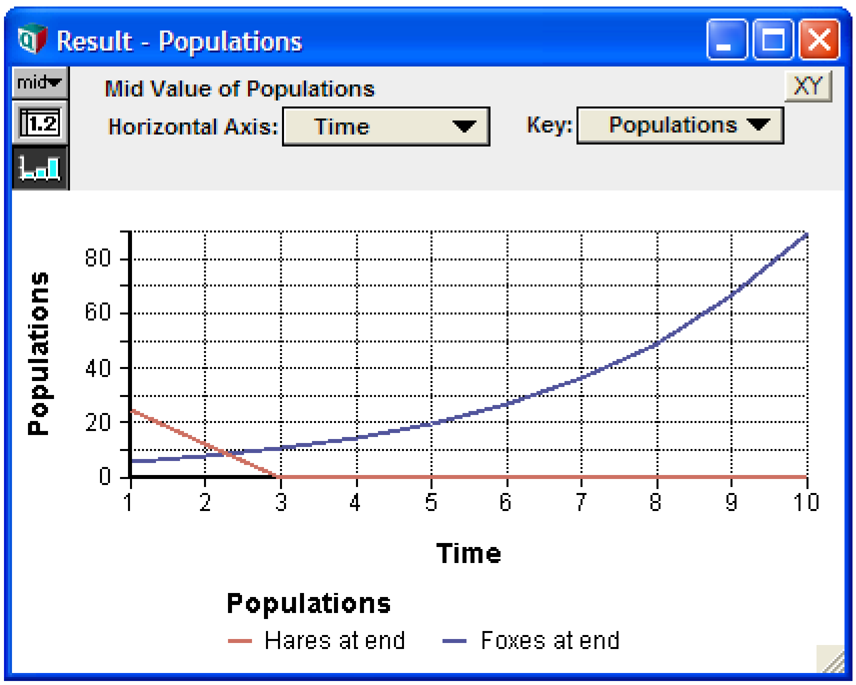
21. In the **Attribute** panel, edit the definition of *Hares at end* until it appears as shown here.



22. Press *Alt+Enter* to accept the revised definition.

23. Select *Populations* in the model diagram.

24. Click the **Result** button to see the resulting populations as a function of time.



Oops! The hare population goes to zero! You don't find extinction very interesting, so you will assume that a small number of hares, say 20, can hide from the foxes and survive. You will further modify the definition of *Hares captured*, to allow 20 to survive.

We can do this by changing the definition of the number of Hares captured. We do not want more than 20 less hares than the number of hares at the start; i.e, **`Hares_at_start - 20`**. In other words, our **Min()** function now should look like this:

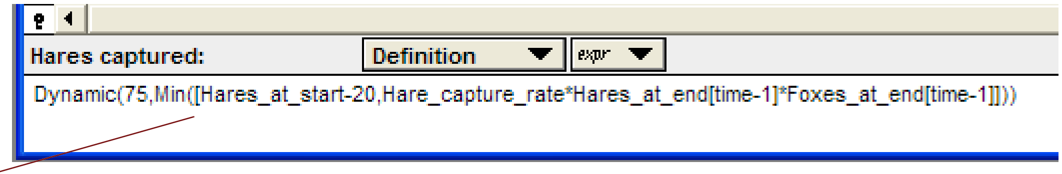
```
Min([Hares_at_start - 20,
Hare_capture_rate*Hares_at_end[time-1]*Foxes_at_end[time-1]])
```

So our full formula for *Hares captured* for each time period is now:

```
Dynamic(75,Min([Hares_at_start - 20,
Hare_capture_rate*Hares_at_end[time-1]*Foxes_at_end[time-1]]))
```

25. Select *Hares captured*.

26. Edit the definition of *Hares captured* until it appears as shown here.

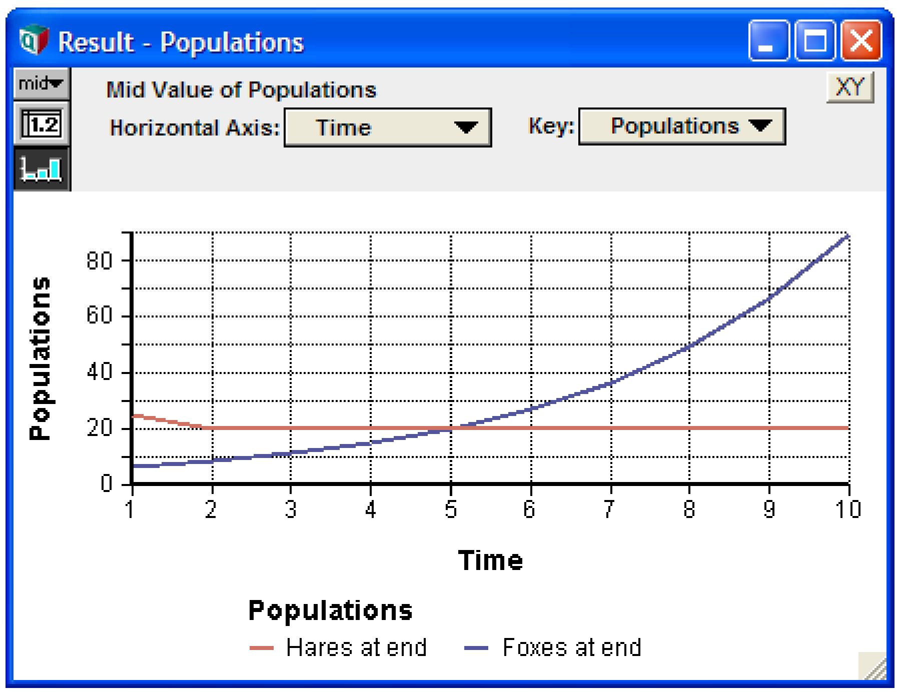


27. Click the check button (or press *Alt-Enter*) to accept the revised definition.

28. Select *Populations*.

29. Click the **Result** button to see the results.

30. Return to the **Diagram** window.

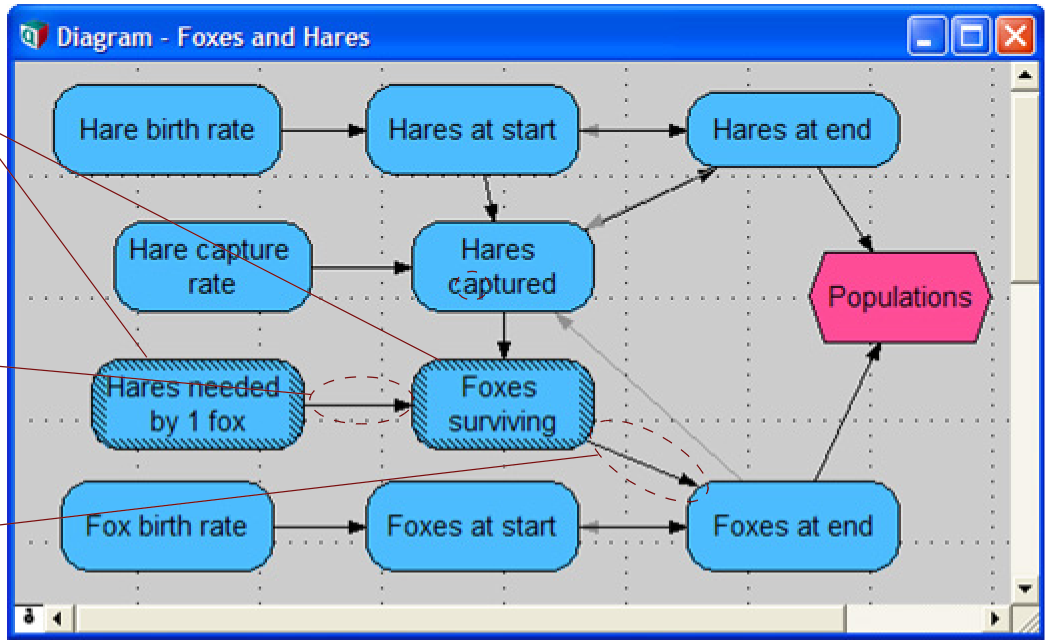


Now the hare population is well under control: it falls to 20 and remains there. However, one piece of the model is still missing: population control for the foxes, who must have enough hares to eat.

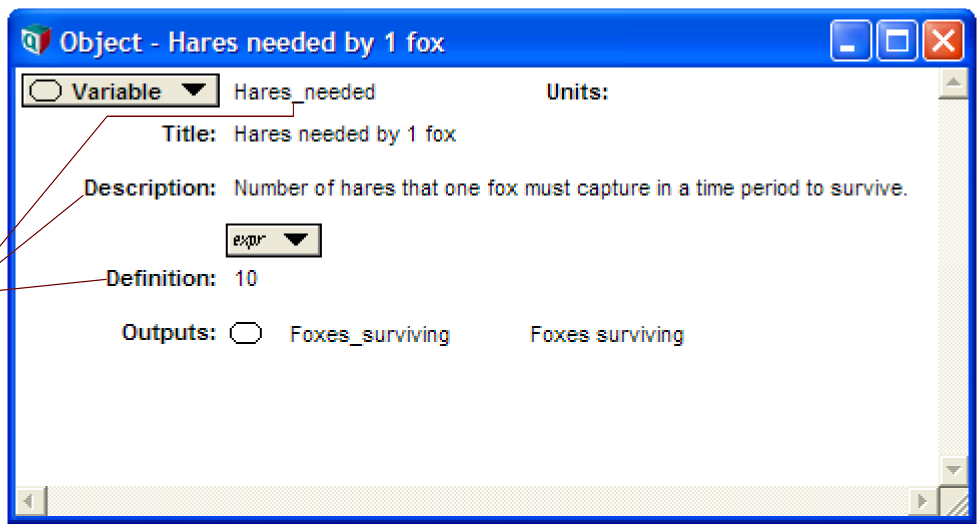
Defining population control: foxes require hares

During any time period, each fox needs to eat at least ten hares in order to survive. Therefore, in any time period, the number of foxes surviving is, at most, the number of hares captured divided by ten. You will now add this constraint to your model.

1. Create two general variable nodes and title them ***Hares needed by 1 fox*** and ***Foxes surviving***.
2. Draw arrows from both *Hares needed by 1 fox* and *Hares captured* to *Foxes surviving.*
3. Draw an arrow from *Foxes surviving* to *Foxes at end*.



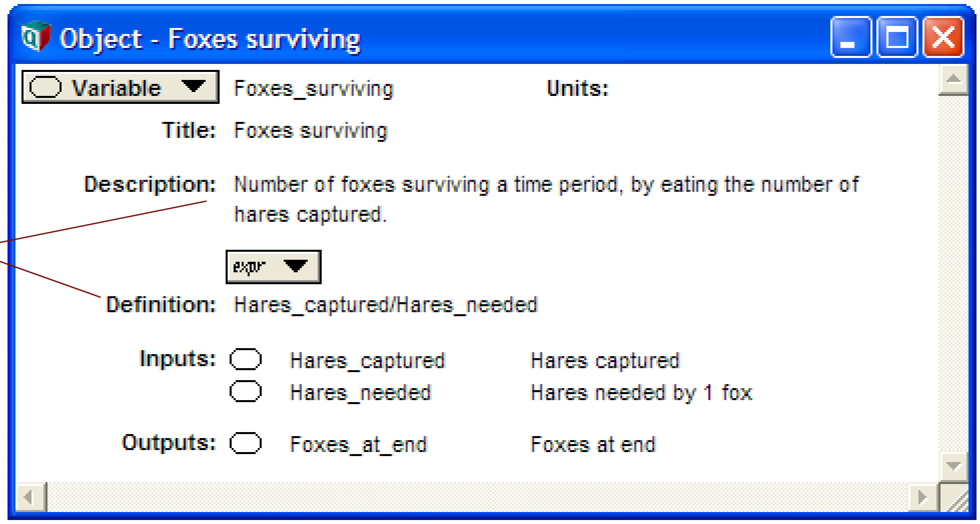
4. Double-click *Hares needed by 1 fox* to open its **Object** window.
5. Edit the Identifier, Description, and Definition as shown here.
6. Close the **Object** window.



7. Double-click *Foxes surviving* to open its **Object** window.

8. Enter the Description and Definition as shown here.

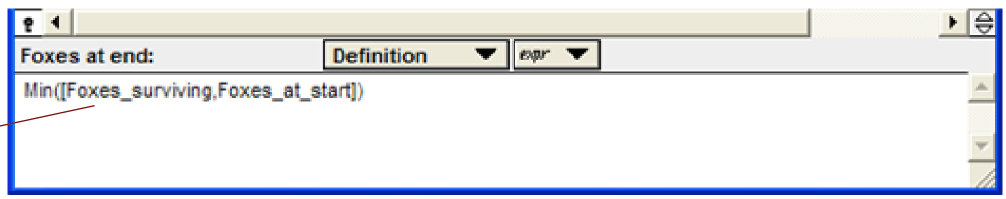
9. Close the **Object** window.



You are ready to adjust the population of foxes at the end of the time period by the number surviving during the period. The number at the end of the period is the smaller of the number of foxes at the start of the period and the number of foxes surviving. Again you will use the **Min()** function.

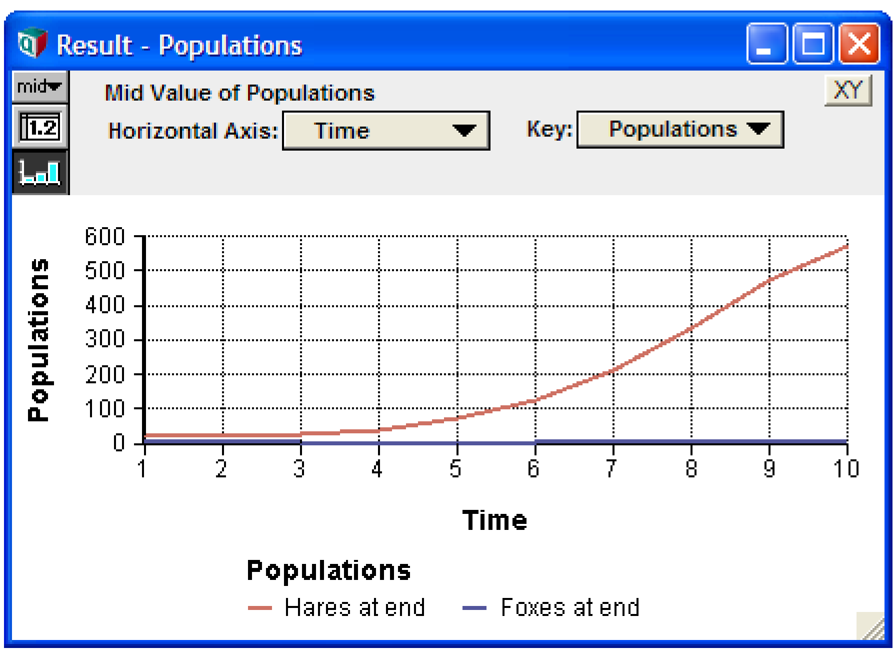
10. Select *Foxes at end*.

11. In the **Attribute** panel, edit its definition to be as shown here.



12. Select *Populations*.

13. Click the **Result** button to see the results.

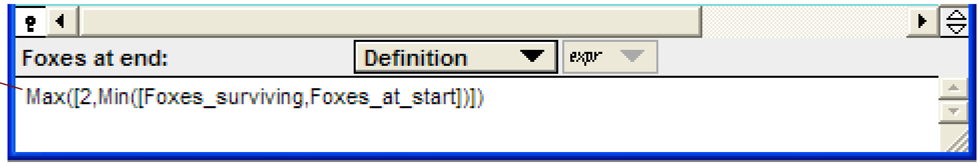


It looks like the population of foxes is declining to nearly 0, allowing the hare population to explode! To verify this, look at the table view by clicking the **table** button 1.2.

You know that in order to continue reproducing, there must be at least two foxes. Therefore, you will further assume that two foxes always manage to survive somehow (and one is male and the other female!). You will modify the definition of *Foxes at end* to constrain it from falling below two.

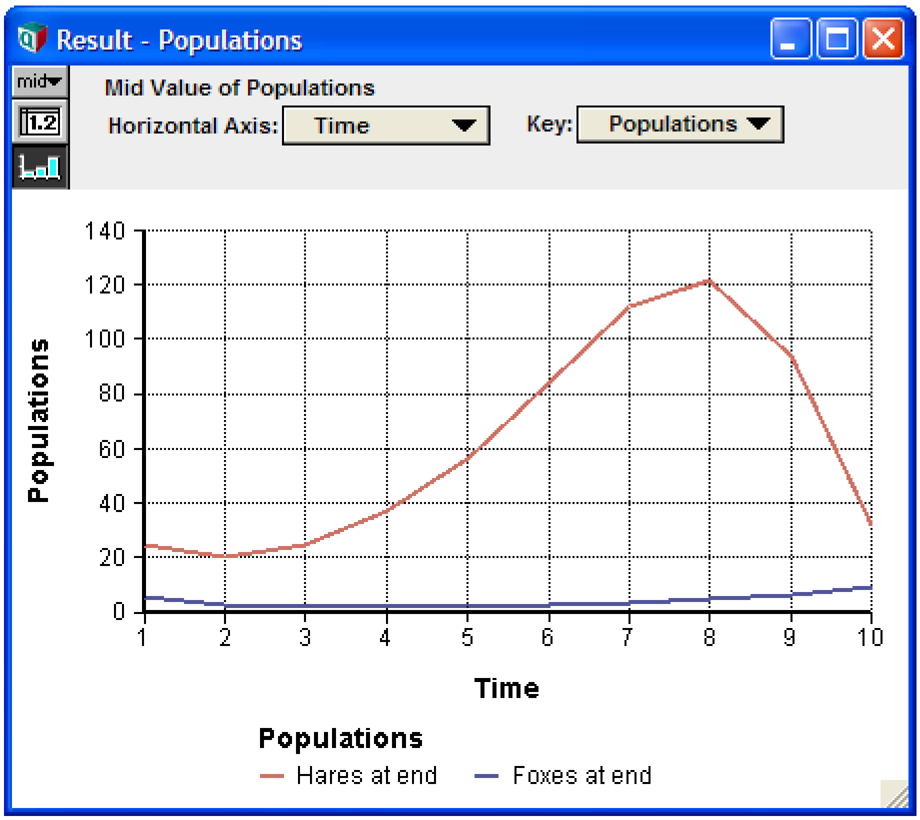
In other words, the population of *Foxes at end* must always be two or the calculated population, whichever is greater. To implement this constraint, you will use the **Max()** function.

14. Edit the definition of *Foxes at end* until it appears as shown here.



15. Select *Populations*.

16. Click the **Result** button to see the results.



Now you can see the fox population falling and rising, with a much larger hare population rising and falling.

Viewing the final results of both populations

Finally, you will examine the fox and hare populations fluctuating over a longer period of time, 100 periods.

1. Select **Edit Time** from the **Definition** menu.

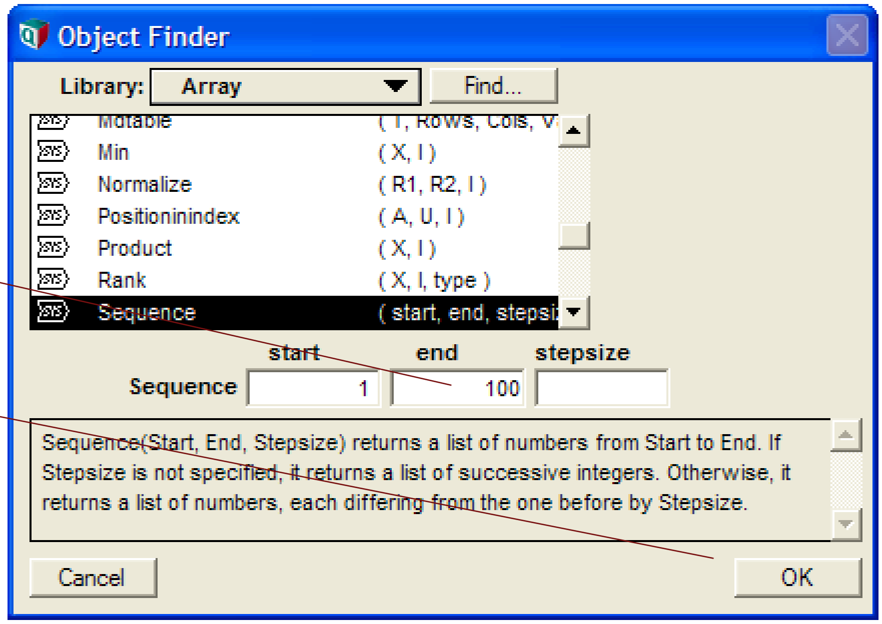
2. Click **Sequence**.

3. Change the **End** parameter from **10** to **100**.

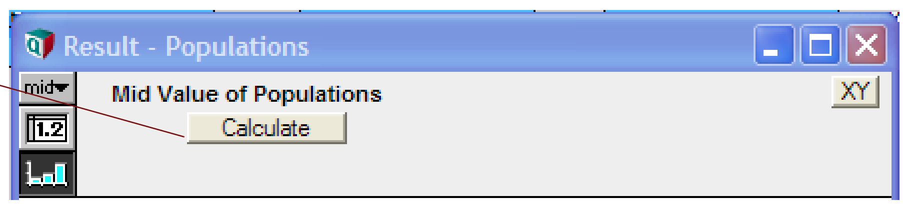
4. Click the **OK** button to save this change.

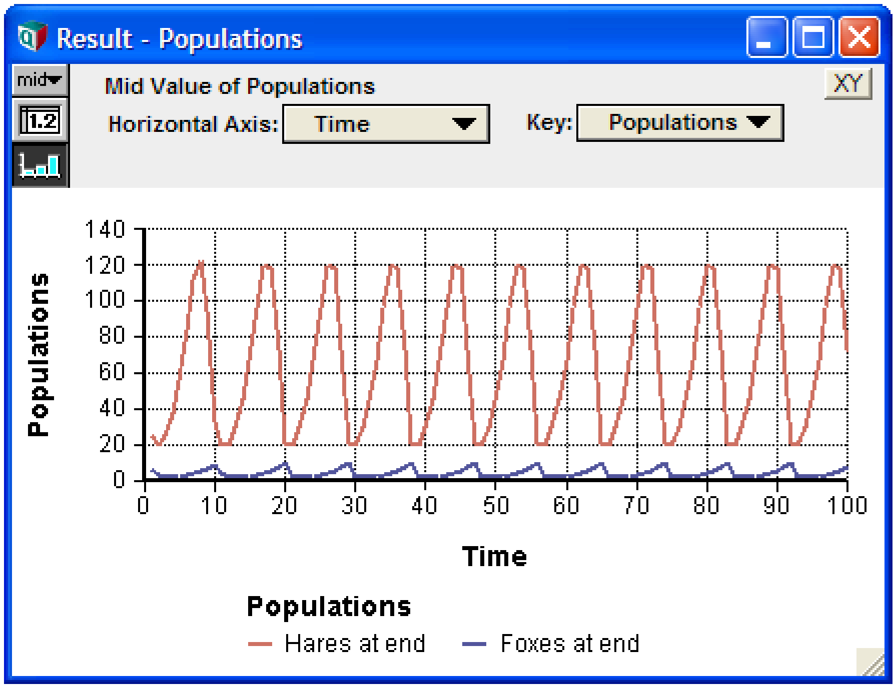
5. Close the **Object** window for *Time*.

   The **Calculate** button appears in the **Result** window.



6. Click the **Calculate** button to see the population cycle swings over the 100 time periods.





Now you can clearly see the rise and fall of the fox population lagging behind the hare population's rise and fall.

Suggestion: Now that the model is complete, go back and revise the input assumptions to see how they impact this result.

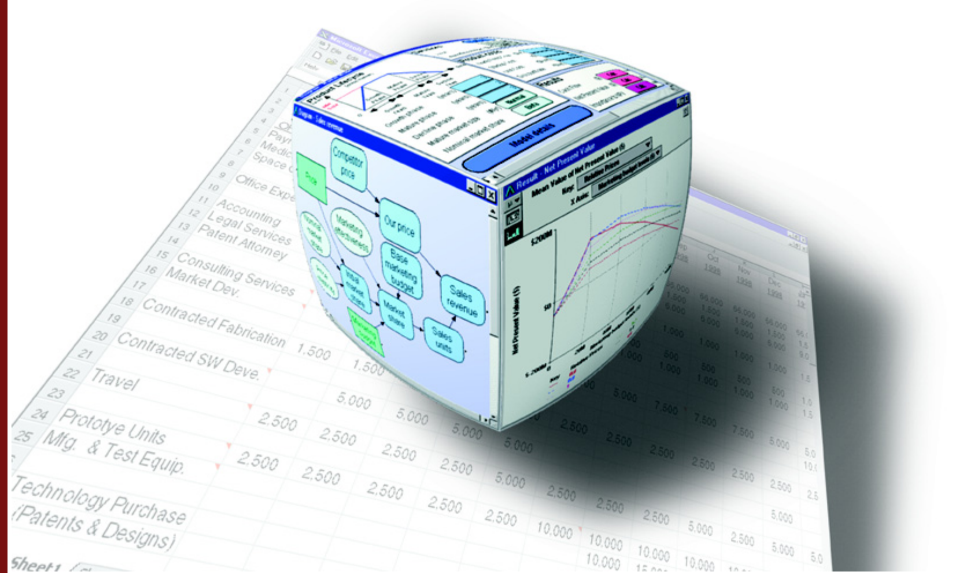
Creating the Foxes and Hares model: summary

In this chapter, you have:

- Defined a variable as a function of another variable at an earlier time period using the **Dynamic()** function and the system variable **`Time`**.
- Had Analytica calculate the results of two nodes simultaneously, and automatically create a new variable to display the results.
- Used the Input popup menu to include variable names in definitions.
- Created more complex expressions to constrain a variable using the **Min()** and **Max()** functions.

Chapter 8 On Your Own

This chapter describes the example models and libraries that are provided with Analytica.



Congratulations on completing the Analytica Tutorial. You are now ready to begin creating your own models.

For more detailed information on Analytica, see the *Analytica User Guide*. It is a reference on all aspects of Analytica, including descriptions of all available functions.

Within the Analytica folder are folders titled **`Example Models`** and **`Libraries`**, which contain a variety of Analytica models, including the examples illustrated in the *Analytica User Guide*. These resources are useful to include when building your own models. Many of the example models were created by users just like you. These models contain a wealth of ideas on using Analytica for practical applications. You should investigate these examples to see some of the different ways in which models can be constructed.

If you create models that you feel would be helpful or interesting to others, please send them to us for inclusion in a future Example Models folder; see the end of this chapter.

The *Example Models* folder is subdivided into these folders:

- Business Examples
- Data Analysis
- Decision Analysis
- Dynamic Models
- Engineering
- Function Examples
- Optimizer Examples
- Risk Analysis
- User Guide Examples

Business Examples

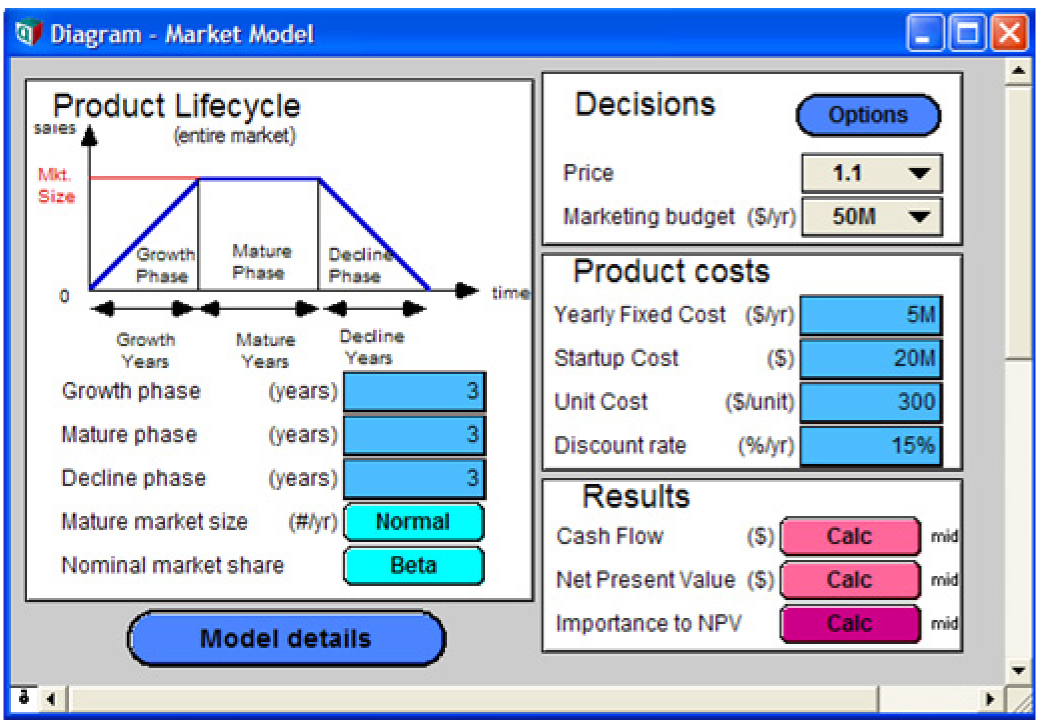
Bond Model This model takes typical bond purchase inputs (purchase price, par value, interest rate, and life to maturity) and calculates bond cash flows, current yield, and yield to maturity.

Breakeven Analysis This model is an example of a breakeven analysis of a set of revenue levels, when the fixed expenses are set at one amount and the variable expenses are a constant fraction of revenue.

Expected R&D Project Value This model evaluates and compares the expected commercialization value of multiple proposed R&D projects.

Financial Statement Templates This model contains a complete set of standard financial statements: a profit and loss statement, balance sheet, and cash flow statement. It provides a step-by-step guide to using these templates to generate your own financial statements. You can enter values into the existing template and modify the variable definitions to reflect your own accounting standards.

Market Model This model explores a market for a new product, and the pricing and advertising budget decisions involved. This example also shows the use of “forms” for receiving input and presenting output for users of the model.



Plan_Schedule_Control This model takes input data for activity paths required to complete a project, and calculates various outputs describing the critical path, timing, and costs for project completion.

Project Portfolio Planner This model evaluates and prioritizes a portfolio of projects based on either the estimated net present value or a multi-attribute score, based on strategy fit, staff development, the generation of public goodwill, and estimated net revenue.

Sales Effectiveness This model evaluates the effects of unit price on salesmen head count and production capacity. The model contains an example of taking user estimates of uncertainty in a standard high-medium-low form, and transforming those inputs into a continuous distribution for propagation through the model.

Derived from *Principles of Systems* by Jay W. Forrester, 1968, ISBN 0-915299-87-9.

Subscriber Pricing This model determines the amount of revenue needed on a monthly basis from each subscriber of a service to just meet the weighted average cost of capital of the firm from the service release date to the end of the study horizon. In other words, it calculates the monthly unit revenue rate required from each subscriber of a service to give a return on investment at the end of the study horizon that is equal to the weighted average cost of capital of the firm.

Data Analysis

Kmeans Clustering

This model shows an example of scatter plots in Analytica. A **K**-means clustering algorithm (where **K** is the number of clusters) is applied to some random data to partition points into groups (clusters) of similar points.

This model also demonstrates the *Iterate* function.

Moving Average Example

This is a simple model that shows how to compute the moving average for a data stream. It defines a **Moving Average** function you can use.

Multidimensional Scaling

This model performs multidimensional scaling. It takes as input **N**, which is the dimensionality of the problem, and **Distances**, which is an **N**x**N** symmetric matrix of distances (or dissimilarities). It calculates and outputs a two-dimensional set of **N** points **XY** (or separately as **Xcoord** and **Ycoord**) that best approximates the spatial layout of points that could generate the input distances.

Reference: *Multivariate Analysis* by K.V. Mardia, J.T. Kent, and J.M. Bibby, Academic Press, London, 1979, Section 14.2.2, page 400. Model supplied by Michael L. Thompson.

Principle Components

Principal components analysis (PCA) is a technique used to reduce multidimensional data sets to lower dimensions for analysis. PCA involves computing the eigenvalue decomposition or singular value decomposition of a data set.

This model shows how to find the principle components in a uses an eigenvalue decomposition to compute the principle components of the covariance matrix of historical stock prices.

Regression Examples

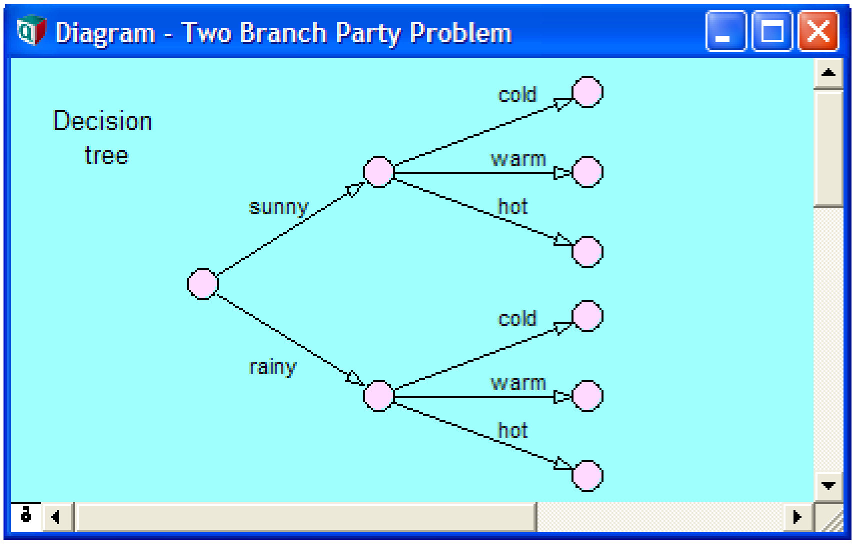
This model demonstrates the use of generalized linear regression by best fit curves of various function forms to a set of (x,y) points. It includes:

- Linear regression
- Quadratic regression
- Polynomial regression
- Discrete Fourier series
- Regression with redundant basis
- Regression using a large arbitrary collection of terms (useful in the situation where you do not have any reason to prefer one functional form over another)
- An auto-regressive series

Decision Analysis

This folder includes models that illustrate the discipline of decision analysis.

Two Branch Party Tree The author of this model wants to throw a party, and can't decide where to throw it. This model shows how to model a two-branch decision tree in Analytica.



Beta Updating This model uses the beta distribution for the Bayesian update of beliefs about the probability that a coin will come up heads.

Biotech R&D Portfolio This multi-project R&D evaluation models a typical R&D decision problem that might be faced by a biogenetic company.

Diversification Illustration This model is an example of a Blitzogram™, which is one way to display the effect of diversifying over a growing set of investments. The example from the model is taken from "Blizograms - Interactive Histograms" in *Informs Transactions on Education*, Vol. 1., No. 2 (Jan. 2001) by Sam Savage. For further information, consult:

- "Beat The Odds: Understand Uncertainty" at http://www.optimizemag.com/issue/002/financial.htm
- Sam Savage's web site: http://drsamsavage.com

Gibbs Sampling in Bayesian Network This model solves a Bayesian network using the Gibbs sampling method, also referred to as Stochastic Simulation. It is an instance of Markov Chain Monte Carlo simulation. This implementation runs multiple simulations simultaneously. You can specify observations for any subset of variables in the model (using the pull-down menus), and compute the posterior probabilities for any of the other variables.

For more on this technique, see S. L. Lauritzen and D. J. Spiegelhalter, "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems," Journal Royal Statistical Society Series B50:2, 1984, p. 157-224.

LEV R&D Strategy This example models R&D decision analysis for investment strategy among several choices of powerplants for a low emissions vehicle (LEV).

Multi-attribute Utility Analysis This model is an example of a multi-attribute utility analysis for cars, showing how to analyze an array of cars across an array of attributes, where different drivers assign differing weights to the importance of each attribute.

Newton-Raphson Method This model implements the Newton-Raphson (or simply Newton's) method, one of the most powerful and well-known numerical methods for finding the root of f(x)=0.

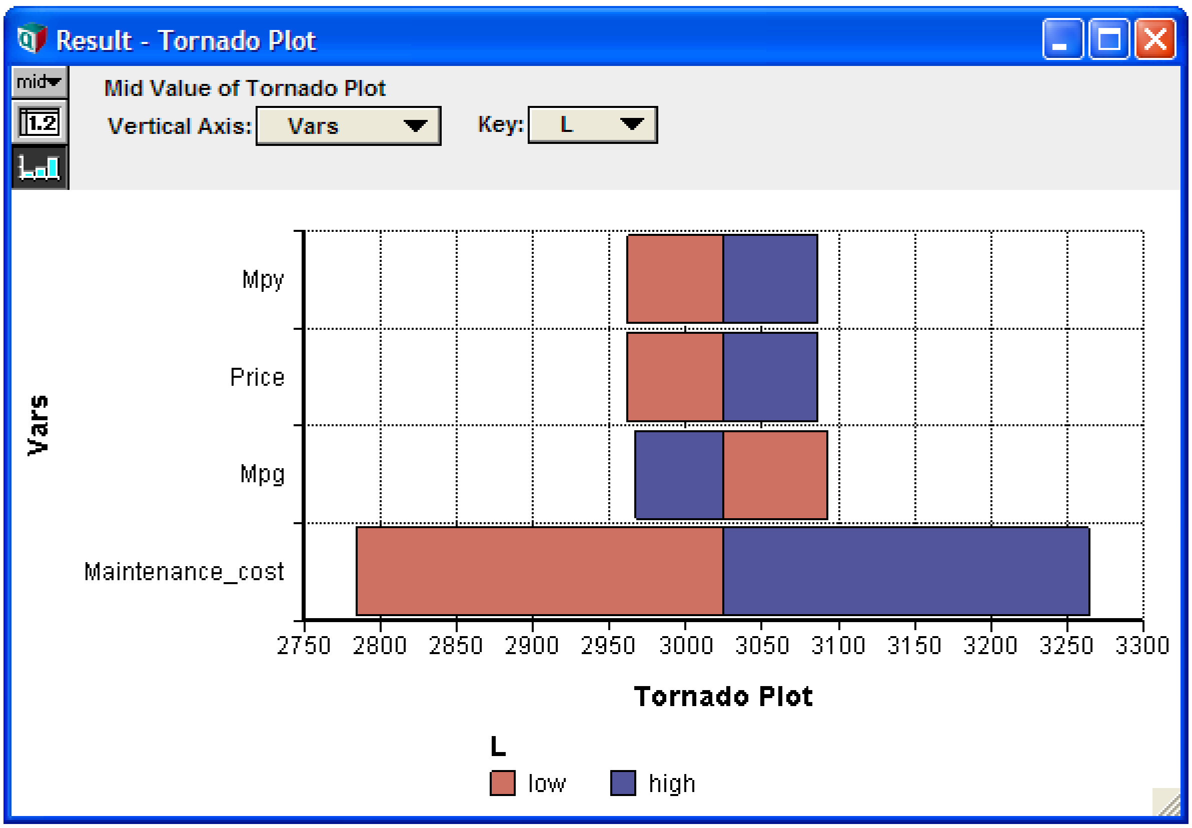
Nonsymmetric Tree This model uses decision tree terminology to provide an example asymmetric decision tree in Analytica.

Party With Forecast This model presents a problem facing a party host. In the face of uncertain weather, what is the best location to hold a party? The value the host assigns to the party is a function of both the location chosen and the weather outcome.

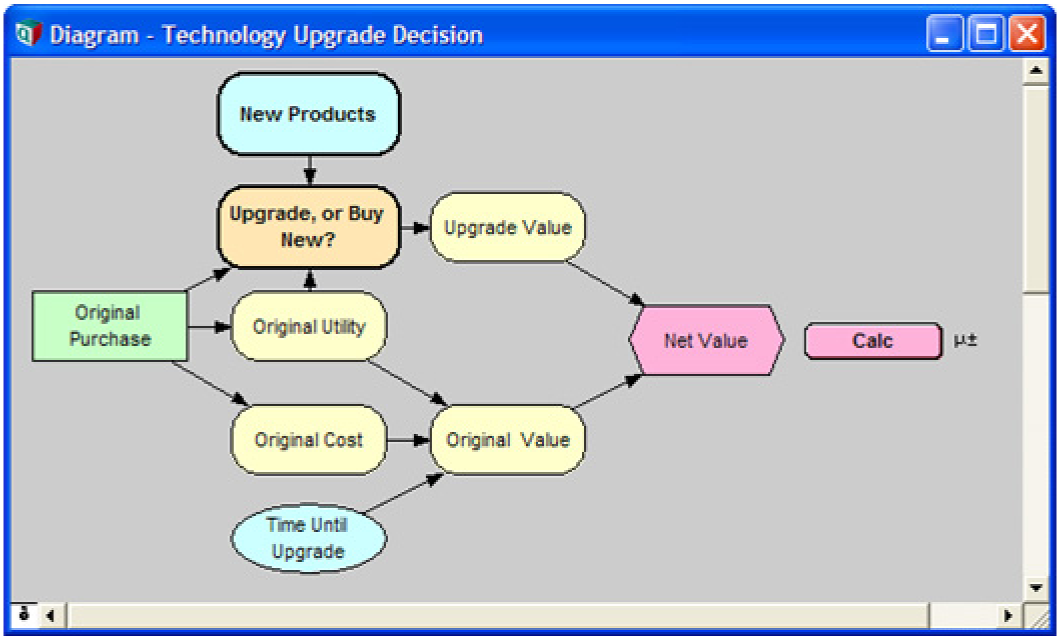
This model augments the basic party model in order to show the value of imperfect information — in this case, a weather forecast — using Bayesian updating.

Supply and Demand This model calculates the required supply level to maximize profit when the profit function is asymmetric around the average demand value.

Tornado Diagrams A tornado diagram is a common tool used to depict the sensitivity of a result to changes in selected variables. The fundamental analysis behind a tornado diagram consists of varying only one input variable at a time, keeping all other variables at their nominal values. Typically, a low and a high value are selected for each input, and the output variable is computed while only one variable varies at a time. This example model shows two methods for selecting high and low values: 1. By varying all inputs by the same relative amount, e.g., low=90% of nominal, high=110% of nominal, or 2. By varying all inputs between two given fractiles. This only makes sense if your inputs are uncertain variables. Example: low=10% fractile, high=90% fractile, nominal=50% fractile.



Upgrade Decision This model represents a decision often faced in today's world: which technology to purchase now, in the face of uncertain future products and prices.



Dynamic Models

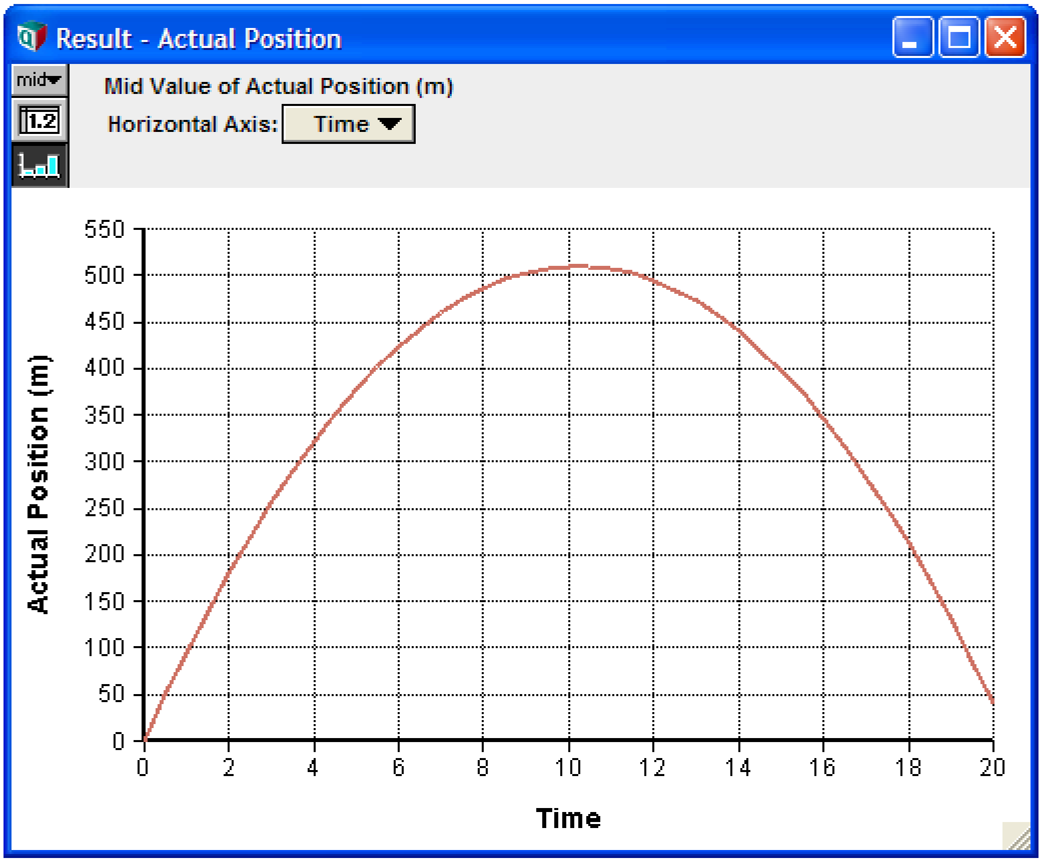
This folder includes models that use Analytica's **Dynamic()** function for performing dynamic simulation (modeling with cyclic dependencies).

Leveling This example levels staff efforts over time according to staff available, computing both the work done over time and idle time.

Markov Chain This model demonstrates how to simulate a Markov process using dynamic time. The example estimates the number of hospital patients over time, modeled as a Markov process.

Mass-Spring-Damper This model simulates a typical free mass-spring-damper system. The term "free system" means that there is no time-dependent driving force or displacement acting on the mass. Ordinarily solutions to such a system are determined from a set of homogeneous second-order differential equations accompanied by the appropriate initial conditions. In this model, the kinematic variables (displacement, velocity, and acceleration) are related to the typical kinematic equations, and the dynamic variables (spring force and damper force) are related to the acceleration and the system mass by Newton's second law. You input the various initial state conditions (spring constant, damper constant, mass, initial displacement, and initial velocity) and the run time of the model. The graphical solutions generated by this dynamic model are comparable to the solutions determined by the corresponding differential equations.

Projectile Motion An example demonstrating how to use the system variable **`Time`** and the **Dynamic()** function to model time-variant behavior — in this case, the motion of a projectile.



Unequal time steps This model gives an example of a dynamic variable that calculates growth over time, where *Time* is defined with unequal time steps. It is an example of exponential or linear growth or decay — that is, a dynamic variable whose values in one time period are exponentially or linearly dependent on the values in the previous time period.

Engineering Examples

Adaptive Filter This model curve fits noisy time-sequence data using an adaptive filter.

Antenna Gain This model calculates the expected gain of an antenna looking at two different satellites.

Failure Analysis This model provides a system simulation demonstrating a failure analysis with both parallel (bulbs) and series (bulbs and switch) components. The model shows the use of a **Determtable** instead of nested **`if...then`** statements to assess the state of the system. Both the switch and the bulbs use an exponential function to assess the probability of failure.

Function Examples

This folder contains examples illustrating a variety of Analytica's functions and modeling techniques.

Assignment from Button This model demonstrates how you can use a button to copy a computed result from one node into the edit table in another node. Analytica Enterprise users can create buttons by dragging a button node from the tool bar onto the diagram.

Autocorrelation This model calculates the auto-correlation coefficients of noisy time sequence data.

Choice and Determtables This model shows that when **Choice** nodes are indexed by "self," you can use **Determtable** functions to propagate the selected choice. This is cleaner than some other methods of using **Choice** outputs.

Correlated Distributions This model reorders a group of probabilistic variables' samples so that they mimic a desired correlation structure as closely as possible.

For more information on this method see R.L. Iman and W.J. Conover, "A distribution free approach to inducing rank correlation among input variables," Commun. Statist.-Simula. Computa. (Marcel Dekker, Inc.), 11(3), 1982, 311-334.

Correlated Normals This model demonstrates a method for creating two normal distributions with a specified correlation between them. The two resulting unit normals can be transformed to have any mean and standard deviation. From E.M. Scheuer and D.S. Stoller, "On the generation of Normal Random Vectors," *Technometrics*, 4:278-281, 1962.

DBWrite Example This model demonstrates how you can write data from an Analytica model to a relational database using ODBC. This model requires Analytica Enterprise; refer to Chapter 22 of the *Analytica User Guide*.

Discrete Sampling This model demonstrates how to generate a distribution from a discrete sample of numbers.

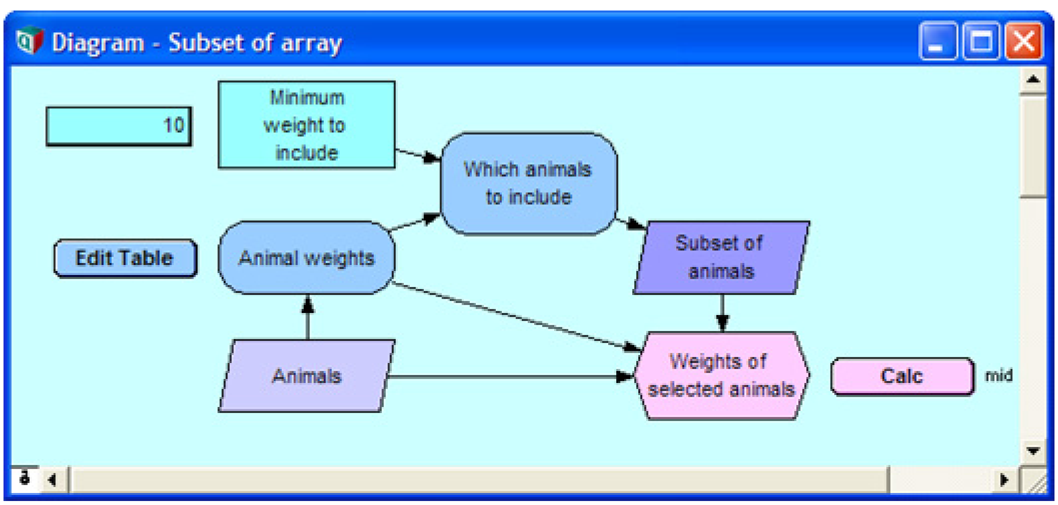
Extracting Diagonal This model demonstrates how to extract a diagonal from a matrix.

Lookup Reindexing This model demonstrates a simple re-indexing operation, essentially how to look up a value from another table. This is shown by the *Salary by person* node, and demonstrates how Excel's **VLOOKUP** function is performed in Analytica expressions.

Sample Size Input Node On occasion, you might want to provide an input node on your form for the sample size system variable, so that a user can adjust the number of samples directly from your form, rather than having to bring up the **Uncertainty Options** dialog. Because you cannot select the **sampleSize** system variable, it is not possible to do this from the Analytica menus. This module provides a way to create this input node — just select **Add Module** and then **Embed**, and you can drag the input node in the module to the form where you would like it to appear. After you do that, you can delete this module, which will then be empty.

Sorting People by Height This example sorts an index (*People*) by a table of values (*Heights*), and then uses the sorted index to created a sorted table of values (*Sorted heights*).

Subset of Array This model creates a subset array out of a larger array based on a decision criterion.



Swapping y and x-index This model swaps a computed or one-dimensional table value with its index, thereby making the computed value an index.

Use of MDTable This model demonstrates the use of the **MDTable** function, which converts records in table form into multi-dimensional arrays. Multi-dimensional arrays are often useful for visualizing large sets of records. By allowing data to be viewed either as a graph or in a pivot table, the geometric relationship between records often becomes immediately evident.

Optimizer Examples

This folder contains examples of how to use Analytica with Optimizer. These models are fully functional only if you have purchased the Optimizer license along with Analytica.

Airline NLP This model gives examples of nonlinear programming optimization and Intelligent Arrays. The examples are:

- Simple airline decision problem to select the number of planes and fares to maximize profit; parametric analysis with respect to demand.
- The same problem with uncertainty, to maximize expected profit.
- The same problem with uncertainty, to maximize profit, given value of uncertain variables
- A dynamic model optimization over multiple years
- A dynamic analysis with optimization in each year

Asset allocation Given many possible investments with varying risk-versus-return trade-offs, one can often reduce risk through diversification. In the best case, investing in two assets with identical expected appreciation, **r**, but which are perfectly anti-correlated in their co-variation yields the expected rate of return with no risk. The more general problem is to select a portfolio that both maximizes return and minimizes risk. There are several possible formulations for this, and this model explores three:

1. Minimize variance (risk) subject to a given expected return.
2. Maximize return subject to a given variance (risk)
3. Balance risk and return by maximizing expected utility, given a person's level of risk aversion.

Automobile Production This model is a linear programming example, taken from "Quick Review of Linear Programming," *Management Science Techniques for Consultants,* by M.A. Trick (1996).

Its objectives show all the varies results from a linear program:

- The optimal solution
- The value of the objective function at the optimum
- The solution status
- The reduced costs (dual values for the variables) at the optimal solution
- The slack or surplus values for the constraints at the optimal solution
- The shadow prices or dual values for the constraints at the optimal solution
- The range over which the objective function coefficient can vary in the linear program without changing the optimal solution
- The range over which a right-hand-side coefficient can vary without changing the dual value (shadow price) of the optimal solution

Big Mac Attack This model addresses the issue of meeting one's daily dietary requirements at McDonalds. The objective can either be to minimize cost, total caloric intake, or total carbohydrates.

The model allows you to solve this problem to result in a *Continuous*, *Integer*, or *Binary* solution.

Selecting *Continuous* results in a computationally easier problem, but less realistic answer, since you cannot order 4.35 Big Macs.

Capital Investment This model is an example of capital budgeting for four possible projects, where the objective is to decide which projects to choose in order to maximize the total return.

NLP with Jacobian This model demonstrates the use of the Jacobian and gradient in a nonlinear optimization with constraints. When a Jacobian is available analytically, it can accelerate the optimization convergence.

The example is a simple geometric problem. Given a set of intersecting circles (where the intersection of all the circles is not empty), which point of those contained in all the circles is closest to a target point?

This model also features a node, *Abstractable NLP*, that is an example of a general mechanism for defining NLPs in an array-abstractable manner.

Optimal can dimensions

This model is a simple example of using the NLP Optimizer in Analytica Optimizer. The model computes the optimal dimensions for a cylindrical can that must hold a given volume. The optimal can has the minimal surface area (and thus, uses the minimum material).

Optimal Production Allocation

This model is a simple example of linear programming in Analytica Optimizer.

An integrated circuit manufacturer produces several different IC products (chips). Each chip is created by a sequence of processes, each carried out by a different machine. Every chip must pass through every process, but the time required for each process depends on the product. Some products require a lot of time at Process1, while other require very little time there but more time at other processes. The company's objective is to determine how much of each product to produce to maximize profit without exceeding the capacity of each process.

Problems with Local Optima

Nonlinear optimization problems often contain local optima as well as global optima. Ideally, we hope that an optimization algorithm would always find the global optima, but no algorithm can guarantee this in the general case. Local minima generally appear to have all the characteristics of a global optima, so that when the Optimizer has found a local minima, it usually terminates the search.

The optional **guess** parameter to **NlpDefine()** provides a way to seed the search to the general area of where you think the global optima might be. When the Optimizer converges to a local optima, it is likely to be in the vicinity of the guess. So by trying a variety of guesses, you might be able to locate a set of local minima. Using the best of those might increase your chances of finding the true global optima.

This model demonstrates this method.

Production Planning LP

This model is an example of production planning linear optimization. A company manufactures four versions of the same product and in the final part of the manufacturing process there are assembly, polishing, and packing operations. For each version, the time required for these operations is different, as is the profit per unit sold. How many of each variant should the company make per year and what is the associated profit?

Solve using NLP

This model demonstrates how an nonlinear programming formulation can be used to solve a nonlinear equation. In this case, the equation is encoded as a constraint (this can be generalized to a system of nonlinear constraints), and the objective function is ignored (constant).

Two Mines Model

This model is another production example. The *Two Mines Company* owns two different mines that produce an ore which, after being crushed, is graded into three classes: high, medium and low grade. The company has contracted to provide a smelting plant with 12 tons of high-grade, 8 tons of medium-grade, and 24 tons of low-grade ore per week. The two mines have different operating characteristics, in terms of cost to operate and production of each type of ore. How many days per week should each mine be operated to fulfill the smelting plant contract?

Risk Analysis

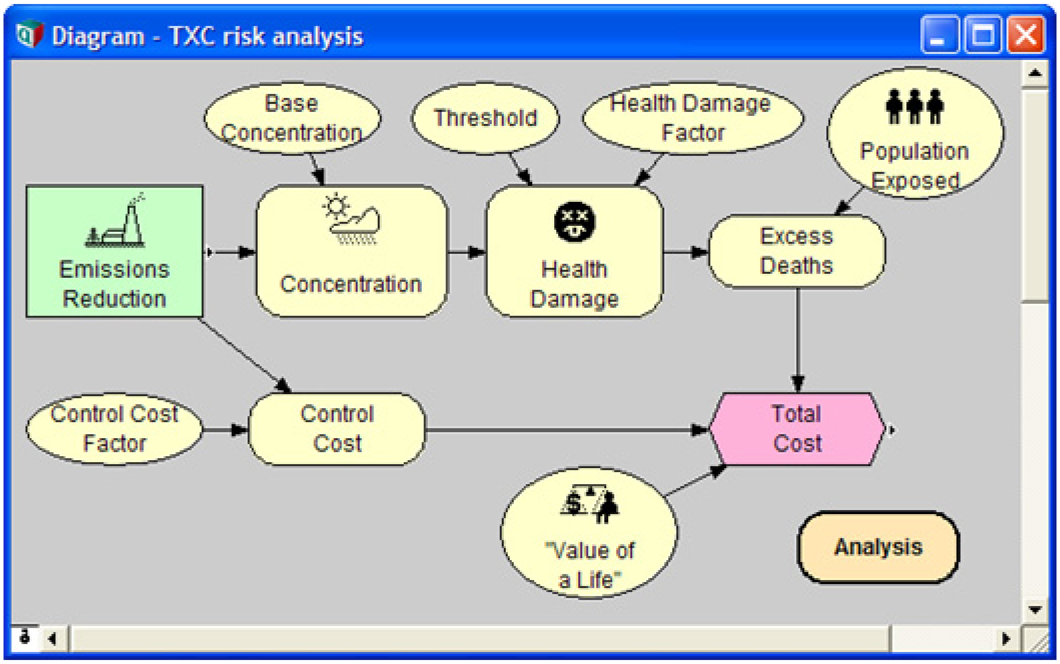
This folder contains applications relating to the field of risk analysis.

Seat belt safety

This model compares the value of various policies for restraints on occupants of automobiles.

Txc

This model demonstrates risk/benefit analysis, in this case regarding the benefits of reducing the emissions of fictitious air pollutant TXC.



User Guide Examples

This folder contains the examples that are given in the *Analytica User Guide*.

Analyzing Unc & Sens The examples in this model demonstrate Analytica's tools for analyzing the uncertainty of variables, relationships between uncertain variables, and sensitivity of outputs to changes in inputs. These include statistical functions and sensitivity analysis functions.

This model is used in the *Analytica User Guide*, Chapter 16, "Statistics, Sensitivity, and Uncertainty Analysis."

Array Examples The examples in this model demonstrate the basics of working with multidimensional arrays.

This model is used in the *Analytica User Guide*, Chapter 11, "Arrays and Indexes."

Array Function Examples The examples in this model demonstrate many more of Analytica's built-in array functions.

This model is used in the *Analytica User Guide*, Chapter 12, "More Array Functions."

Continuous Distributions A continuous distribution is one that is defined for a continuous variable — that is, for a real-valued variable. The examples in this model demonstrate Analytica's built-in functions that create or modify continuous distributions.

This model is used in the *Analytica User Guide*, Chapter 15, "Probability Distributions."

Discrete Distributions A discrete distribution is a probability distribution for a variable that can result only in certain, discrete outcomes. The examples in this model demonstrate Analytica's built-in functions that create or evaluate discrete distributions.

This model is used in the *Analytica User Guide*, Chapter 15, "Probability Distributions."

Expression Examples The examples in this model demonstrate the building blocks for creating and editing variable definitions — expressions, standard operators, and mathematical functions.

This model is used in the *Analytica User Guide*, Chapter 10, "Using Expressions."

Input and Output Nodes This model is used in the *Analytica User Guide*, Chapter 9, "Creating Interfaces for End Users."

Dynamic Models

A dynamic model is a model with one or more dynamic variables — that is, variables that can change over time. These models illustrate various uses of the **Dynamic()** function.

These models are used in the *Analytica User Guide*, Chapter 17, "Dynamic Simulation."

Dynamic & Dependencies This model is a dynamic model that finds the downward velocity and position of a dropped object over a six second time period.

Dynamic & Uncertainty This model shows three ways to use uncertainty with the **Dynamic()** function. In the first case, uncertainty samples are calculated once, at the initial time period. In the other two cases, new uncertainty samples are created for each time period (i.e., the values are re-sampled).

Dynamic Example 1 This model is the most simple dynamic model, with one variable that changes over time. This example finds the gasoline price for each of five years, assuming a 5% growth rate.

Dynamic Example 2 This model is a slight increase in complexity over **Dynamic Example 1**. This model still uses one variable that changes over time. However, instead of assuming a fixed inflation rate, this example, looks at the price with three different inflation rates for comparison.

Libraries

The libraries in this folder contain functions that can be added to your model and used similarly to Analytica's built-in functions. These libraries can be added to your model; see Chapter 19 of the *Analytica User Guide* for information on how to add a library to a model.

Bayes Function This library contains **Posterior()**, a function for calculating posterior probabilities using Bayes' Theorem.

Complex Library This is a library of functions for working with complex numbers. It contains functions for basic arithmetic, polar representations of complex numbers, scalar functions for finding complex roots, logs, exponents, matrix functions, and trigonometric functions. Addition, subtraction, and scalar multiply are performed with the usual operators. Complex multiplication and complex division require the use of explicit functions.

Complex numbers as seen by users of this library should always be in the Euclidean complex form, such as `a + bi`, where `a` is the real part and `b` is the imaginary part.

Concatenation This library contains functions to make concatenation more convenient. Functions **Concat3()** through **Concat10()** are generalizations of the built-in **Concat()** function which concatenate from 3 to 10 arrays in a single call (the built-in **Concat()** function concatenates two arrays). **ConcatRows()** concatenates all the rows of a single array.

Data Statistics Library This library contains functions for calculating statistical quantities for a list of numbers over an explicit index other than **Run**, such as the index used for the statistics results in Analytica: the mean, variance, standard deviation, kurtosis, skewness, fractiles, covariance, correlation, frequency, etc.

Distribution Variations

Expand Index This model contains the function **Change Index**. When this function is given an array indexed by one index, it returns an array indexed by another index.

This model contains an example in which **Change Index** is used to combine cashflows over two different time periods into a single cashflow over a single time period.

Financial Library This model contains a variety of corporate finance functions: Black-Scholes Option Values (**CallOption**, **PutOption**), Capital Asset Pricing Model (**CAPM**), Miles/Ezzell Adjusted Cost of Capital (**CostCapME**), Modigliani/Miller Adjusted Cost of Capital (**CostCapMM**), Present Value of Perpetuity (**PVperp**), Present Value of Growing Perpetuity (**PVgperp**), and Weighted Average Cost of Capital (**WACC**).

Flat File Library This library provides functions for writing data to and from flat files, particularly between two-dimensional tables and comma-separated value (CSV) files.

Garbage Bin Library

This library provides a ***Recycle Bin*** for your model (including a recycle bin icon). To use it, simply drag your discarded objects into the recycle bin module.

Unlike deletion, items in your recycle bin can be retrieved, and the **Undo** command (*Control+Z*) works for items dragged into the bin. If an item you put into the bin has dependents outside the bin, it shows arrows from the bin — a signal that you should either retrieve the item, since it is being used, or also drag its dependent(s) into the bin.

To delete the items in the bin permanently, open the bin, select all its contents (with *Control+A*), and delete them (press *Delete* key).

Linked List Library

This library contains routines for manipulating linked lists.

The simplest linked list is just **NULL** — a linked list with zero elements.

Any other linked list is a reference to a record indexed by **Linked_List** (an index defined in this library) with each cell containing a reference, to the element, and a pointer to the remainder of the list.

Linked lists are created and manipulated using functions in this library:

- Use the function **LL_Push()** to build the list.
- After a list is built, the easiest way to use and view it is to convert it back to an array using **LL_to_RArray()**. This reverses the order of items in the linked list (which has the last item “pushed” into the linked list as the first item in the list) so that the array has the items ordered the same as when they were added to the list.
- The function **LL_to_Array()** returns an array with the items ordered the same way as the linked list (the last entered item is first in the list/array).
- Other functions provide the first item in the list, the N^{th} item in the list, the list length, and allow you to remove (pop) the first item in the list.

Multivariate Distributions Library

This library contains functions for creating several multivariate distributions:

- Gaussian
- Dirichlet
- BiNormal and Multinormal
- Uniform Spherical and MultiUniform
- Sample covariance and Sample Correlation
- Functions for correlating distributions and results

ODBC-Library

This library provides additional functionality (**ValList**, **InsertRecSql**, **WriteTableSql**) for using ODBC access to databases. Note that using ODBC requires Analytica Enterprise; refer to Chapter 22 of the *Analytica User Guide*.

Profiling Library

Use this library to see which variables and functions are taking most of the computation time when running your model. Consult the model’s description for an explanation of how to use the library.

Tip

The ODBC and Profiling libraries require Analytica Enterprise, or ADE. They do not work with other versions of Analytica.

Optimization Functions Library

This library contains functions for optimization and equation solving, using a Newton-Raphson style search.

Summary

If you have created a model that other Analytica users would benefit from, please send it to us for possible inclusion in future versions of the *Example Models* folder. Send your well-constructed, thoroughly documented models to:

support@lumina.com

Also, if you experience any problems with the example models, or if you feel that they need to be changed in some way, please contact us at the above email address.

You can also submit example models to the Analytica wiki, a web application that allows member of the Analytica community to collaborate and share information. Go to:

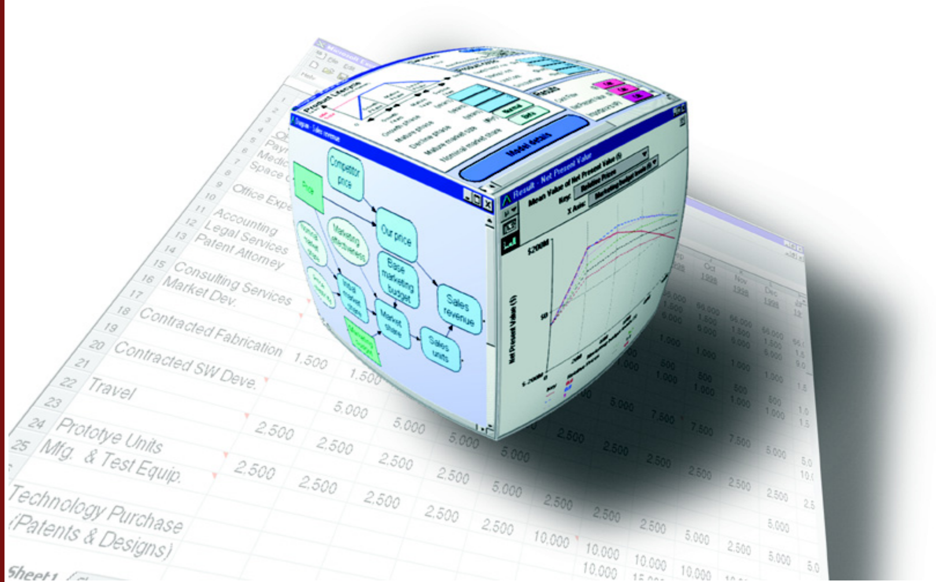
http://lumina.com/wiki/index.php/Example_Models

Here you can view example models submitted by others and to submit your own examples.

Go to the Analytica wiki home page (http://lumina.com/wiki/) to find all sorts of information on Analytica including additional information on Analytica functions, frequently asked questions (FAQs) and their answers, guides to modeling, what's new in Analytica, and much more.

Glossary

This glossary includes a compilation of terms specific to Analytica as well as statistical terms used in this manual.



Glossary

Array A collection of values that can be viewed as one or more tables. An array has one or more dimensions; each dimension is identified by an index.

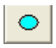
Arrow, influence arrow Influence arrows (or arrows) from one variable node to another indicate that the origin node affects (influences) the destination node. If the nodes depict variables, the origin variable is an input to the destination variable, and the destination variable is an output of the origin variable.

 Arrow tool The tool for drawing arrows between nodes.

Attribute A property of an object, such as its title, description, definition, value, or inputs.

Attribute panel An auxiliary window pane that can open below an influence diagram. Use it to rapidly examine one attribute at a time of any variable, function, or module.

 **Browse tool** A tool for examining the structure and assumptions of a model, with limited ability to make changes to the model.

 **Chance variable** An uncertain variable that cannot be directly controlled by the decision maker. It is usually defined by a probability distribution. A chance variable is represented by an oval node.

Class Analytica objects are organized into the following classes: module, attribute, function, and decision, chance, objective, index, and general variables.

Cumulative probability distribution A graphical representation of a probability distribution. It plots the cumulative probability that the actual value of the uncertain variable **`x`** is less than or equal to each possible value of **`x`**. The cumulative probability distribution is a display option in the Uncertainty View popup menu in a **Result** window.

 **Decision variable** A variable that the decision maker can control directly. A decision variable is represented by a rectangular node.

Definition A specification for computing a variable's value. The definition can be a number, a mathematical expression, a list of values, a table, or a probability distribution.

Description Text explaining what the object represents in the system being modeled. The description is limited to 32,000 characters in length.

Deterministic table A function that gives the value of a variable conditional on the values of its input variables, where the inputs are all discrete variables.

Deterministic value See ***Mid, Mid value***.

Domain The possible outcomes for a variable defined as a probability table.

Edit table A definition that is an array (table) is also called an edit table because it can be edited.

 **Edit tool** A tool for creating or changing a model. Use it to move, resize, and edit nodes, and to expose the arrow tool and node palette.

Expression A formula that can contain any combination of numbers, variables, functions, distributions, and operators, such as **`0.5`**, **`a-b`**, or **`Min(x)`**, combined according to the Analytica language syntax.

Expression type Expression types include expression, list (of expressions or numbers), list of labels (text strings), table, probability table, and distribution. You select an expression type using the Expression popup menu, which appears above the **Definition** field. Note that any definition, regardless of expression type, can be viewed as an expression.

 **General variable** Any type of variable; useful when the variable type is unknown. The general variable typically represents a deterministic or functional dependency.

 **Graph** Format for displaying a multidimensional result. To view a result as a graph, click the **Graph** button. See also ***Table***.

Identifier A short name for an object. A variable's identifier is used to refer to the variable in mathematical expressions in definitions of other variables. An identifier must start with a letter, have no more than 20 characters, and contain only letters, numbers, and the underscore characer (_) (used instead of a space). Compare to ***Title***.

Importance analysis Shows the effect the uncertainty of one or more input variables has on the uncertainty of an output variable. Importance is defined as the rank-order correlation between the sample of output values and the sample for each uncertain input. This is a robust measure of the uncertain contribution because it is insensitive to extreme values and skewed distributions.

Index Identifies a dimension of an array (table). An index is usually a variable defined as a list, list of labels, or sequence.

Indexes Plural of index, indicates a set of index variables that define the dimensions of a table (in an edit table or value).


Index variable A class of variable, defined as a list, list of labels, or sequence, that is used to identify the dimensions of a table, for example, in an edit table. An index variable is represented by a parallelogram node. See also ***Edit table***.

Influence arrow See ***Arrow, influence arrow***.

Influence diagram A graphical representation of a model, consisting of nodes (variables) and arrows (relationships between variables).

Input An input of a variable `x` is a variable that has an arrow drawn to `x`, or appears in the definition of `x`. See also ***Output***.

Input arrowhead Shows that a node has one or more inputs from outside its module. This arrowhead is located on the left side of a node. Press the arrowhead for a popup menu of input variables.

List A type of expression, consisting of an ordered set of numbers or expressions, available in the Expression popup menu. A list is often used to define index and decision variables.

List of labels A type of expression, consisting of an ordered set of text labels, available in the Expression popup menu. A list of labels is often used to define index and decision variables.

Mean The average or expected value.

Median The middle number or value when the data values are ranked in order of size, i.e., the middle data point.

Mid, Mid value A calculation of the variable's value assuming all uncertain inputs are fixed at their median values.

Model A module, or a hierarchy of modules; the main, or root, module at the top of the module hierarchy. Between sessions, a model is stored in an Analytica document file.

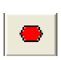

Module A collection of related objects, including variables, functions, and other modules, organized as a separate influence diagram. A module is represented by a rounded rectangular node with a thick outline.

Node A box (rectangular, oval, or any other shape) that represents a variable in an influence diagram. Different node shapes are used to represent different types of variables.

Normal distribution The bell-shaped curve, or Gaussian distribution.

Object Finder A dialog box used to browse and edit the functions and variables available in a model.

Object window List of the attributes for an object (variable, function, or module), including its class, identifier, title, and description.


Objective variable A variable that evaluates the overall value or desirability of possible outcomes. The objective can be measured as cost, value, or utility. A purpose of most models is to find the decision or decisions that optimize the objective — for example, minimizing cost or maximizing expected utility. Most decision models contain a single objective node, although the objective can be composed of several sub-objectives. An objective variable is represented by a hexagonal node.

Output An output of a variable `x` is a variable that has an arrow drawn from `x`, or whose definition refers to `x`. See also ***Input***.

Output arrowhead Shows that a node has one or more outputs outside its module. This arrowhead is located on the right side of the node. Press the arrowhead for a popup menu of the output variables.

Parametric analysis A type of sensitivity analysis in which you specify a set of alternative values for one or more inputs, and examine the effect on selected model output variables. See also ***Sensitivity analysis***.

Probabilistic variable A variable that is uncertain and is defined with a probability distribution.

Probability bands The bands that capture a certain portion of the total probability for a variable. For example, the 5% and 95% probability bands contain 90% of the total probability, while the 50% probability band corresponds to the median value. By default, the 5%, 25%, 50%, 75%, and 95% probability bands are shown. These bands are also referred to as ***confidence intervals*** or ***fractiles***. Probability bands are a display option in the Uncertainty View popup menu on a **Result** window.

Probability density function (PDF) A graphical representation of a probability distribution that plots the probability density against the value of the variable. The probability density at each value of `x` is the relative probability that `x` is at or near that value. The probability density function can be displayed for continuous, but not discrete, variables. It is a display option in the Uncertainty View popup menu on a **Result** window.

Probability distribution Describes the relative likelihood of a variable having different possible values.

Probability table A table for defining a chance variable with a set of outcomes and a discrete probability distribution (numerical probability for each outcome). If the variable depends on (that is, is conditioned by) other discrete variables, each of these conditioning variables gives an additional dimension to the table, so you can specify the probability distribution conditional on the value of each conditioning variable.

Self A keyword referring to the index of a table that is indexed by itself. `Self` refers to the alternative values of the variable defined by the table.

Sensitivity analysis A group of methods to identify and compare the effects of various input variables to a model on a selected output. Example methods of sensitivity analysis are importance analysis and parametric analysis. See also ***Parametric analysis***.

Standard deviation Reflects the amount of spread or dispersion in an uncertainty distribution. It is the square root of the variance.

 **Table** A two-dimensional view of an array-valued definition or result. The array can have more than two dimensions, but only two can be seen at one time. A table is a type of expression available in the Expression popup menu. A definition that is a table is also called an ***edit table***. In the **Result** window, click the **table** button to select the table view of an array-valued result. See also ***Graph***.

Title The full name of an Analytica object, briefly describing what the object represents. The title of a variable, function, or module is displayed in its node, in window titles, and in object lists. It is limited to 255 characters in length. The title can contain any characters, including spaces and punctuation. Compare to ***Identifier***.

Uniform distribution A distribution representing an equal chance of occurrence for any value between the lower and upper values.

Uncertainty View options An uncertain result can be displayed as a ***mid value***, ***mean***, statistics, ***probability bands***, ***probability density***, ***cumulative probability***, or sample. Select the option to display with the popup menu in the top left corner of the **Result** window or in the **Result** menu.

Units The increments of measurement for a variable. Units are used to annotate tables and graphs, but are not used in evaluating a variable.

Value A variable's value attribute is its mid value, computed by assuming that all uncertain inputs are fixed at their median values. It can be a scalar (single) number, a table of numbers, or a probability distribution.

Variable An object that has a value, which can be a text string, a number, or a table. Classes of variable include decision, chance, and objective.

Analytica Windows and Dialogs

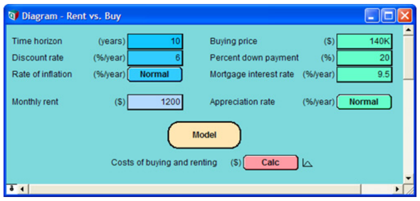


Diagram Window:
Inputs and Outputs

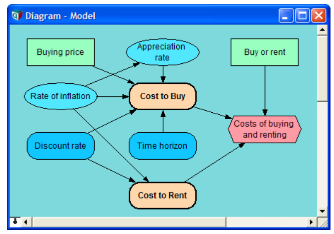
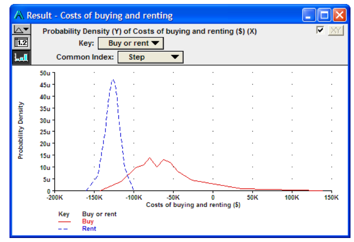
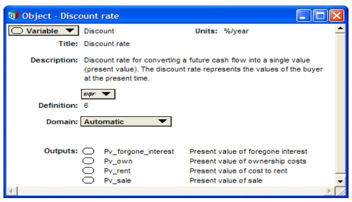


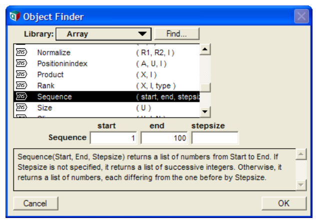
Diagram Window:
Influence Diagram



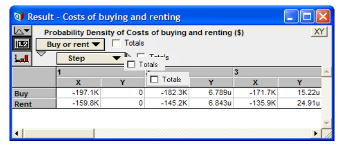
Result Window — Graph View



Object Window



Object Finder



Result Window — Table View

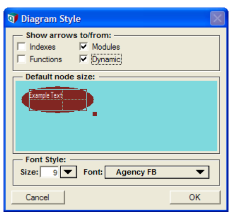
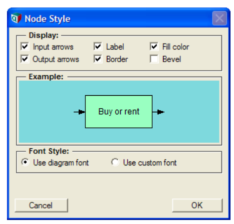
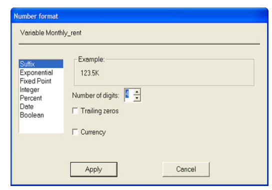


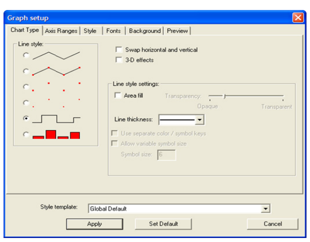
Diagram Style Dialog



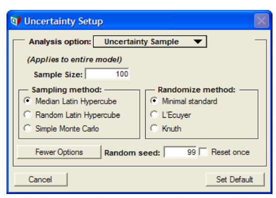
Node Style Dialog



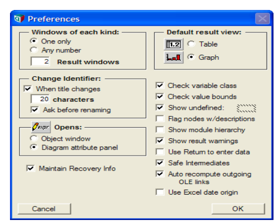
Number Format Dialog



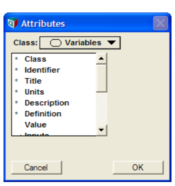
Graph Setup Dialog



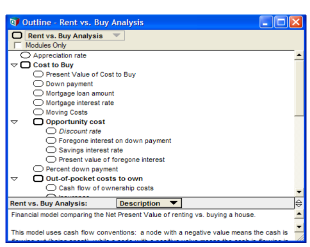
Uncertainty Setup Dialog



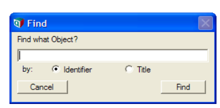
Preferences Dialog



Attributes Dialog



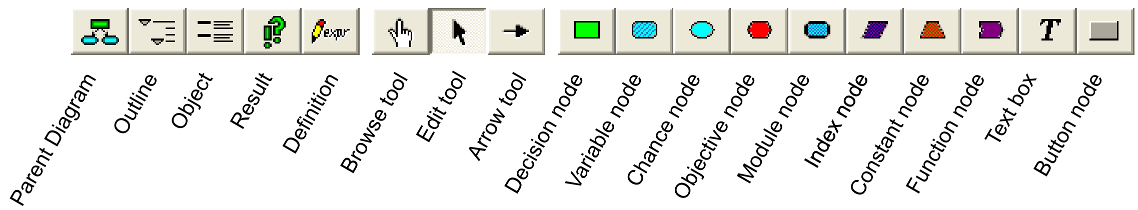
Outline Window



Find Dialog

Quick Reference

The Tool Bar



It displays the node palette when you select the edit tool or arrow tool.

Numerical Formats (Output)

Format	Description	Example
Suffix	letter denotes order of magnitude, such as M for 10^{-6} (see table below)	12.35K
Exponent	scientific exponential	1.235e+004
Fixed Point	fixed decimal point	12345.68
Integer	fixed point with no decimals	12346
Percent	percentage	1234568%
Date	text date	12 Jan 2008
Boolean	true or false	True

Suffix format

Power of 10	Suffix	Prefix
10^3	K	Kilo
10^6	M	Mega or Million
10^9	G	Giga
10^{12}	T	Tera or Trillion
10^{15}	Q	Quad

Power of 10	Suffix	Prefix
10^{-2}	%	percent
10^{-3}	m	milli
10^{-6}	µ	micro (mu)
10^{-9}	n	nano
10^{-12}	p	pico
10^{-15}	f	femto