Factors That Affect Optimization Performance .............................................................. 220
Sensitivity Analysis Using a Tornado Chart ................................................................. 225
Maintaining Multiple Optimization Settings for a Model ............................................ 226
Other OptQuest Notes ................................................................................................. 226

Appendix A. Bibliography .................................................................................................. 229
Crystal Ball Tools ........................................................................................................... 229
Bootstrap ......................................................................................................................... 229
Tornado Charts and Sensitivity Analysis ..................................................................... 229
Two-Dimensional Simulation .......................................................................................... 230
Monte Carlo Simulation ............................................................................................... 230
Probability Theory and Statistics .................................................................................. 231
Random Variate Generation Methods .......................................................................... 231
Specific Distributions ......................................................................................................... 232
Extreme Value Distribution ........................................................................................... 232
Lognormal Distribution ................................................................................................... 232
Weibull Distribution ......................................................................................................... 232
Uncertainty Analysis ......................................................................................................... 232
Spreadsheet Design ......................................................................................................... 233
Sequential Sampling with SIPs ....................................................................................... 233
Time-series Forecasting References ................................................................................ 233
Forecasting ....................................................................................................................... 233
ARIMA Forecasting ......................................................................................................... 234
Regression Analysis ......................................................................................................... 234
Optimization References ................................................................................................... 235
OptQuest References and White Papers ......................................................................... 235
Metaheuristics .................................................................................................................. 235
Stochastic (Probabilistic) Optimization Theory ............................................................. 235
Multiobjective Optimization ............................................................................................. 236
Optimization and Simulation in Practice ......................................................................... 236
Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
Introduction

Oracle Crystal Ball is a user-friendly, graphically oriented forecasting and risk analysis program that takes the uncertainty out of decision-making.

Crystal Ball runs on several versions of Microsoft Windows and Microsoft Excel. For a complete list of required hardware and software, see the current Oracle Crystal Ball Installation and Licensing Guide.

About This Guide

The Oracle Crystal Ball Reference and Examples Guide contains examples plus formulas and other reference information. It applies to the following Oracle products:

- Crystal Ball (including Classroom Editions)
- Oracle Crystal Ball Decision Optimizer
- Oracle Crystal Ball Enterprise Performance Management and related products

This guide includes the following additional chapters:

- Chapter 4, “Statistical Definitions”
  This chapter describes basic statistical concepts and explains how they are used in Crystal Ball.
- Chapter 6, “Probability Distribution Examples and Reference”
  This chapter lists the mathematical formulas used in to calculate distributions and descriptive statistics and describes the type of random number generator used in Crystal Ball. This appendix is designed for users with sophisticated knowledge of statistics.
- Chapter 7, “Predictor Examples and Reference”
  This chapter provides formulas and techniques used in Predictor.
Note: Because of round-off differences in various system configurations, you might obtain calculated results that are slightly different from those in the examples.

Technical Support and More

Oracle offers technical support, training, and other services to help you use Crystal Ball. See:

http://www.oracle.com/crystalball
Introduction

This chapter contains information that you can use to improve the overall performance of Crystal Ball. These improvements occur in terms of the accuracy of the model or speed of the results.

Simulation Accuracy

The accuracy of the simulation is primarily governed by two factors:

- The number of trials, or length, of the simulation
- The sampling method

Generally speaking, the more trials you run in a simulation, the greater the accuracy of the statistics and percentiles information. This greater accuracy comes at the expense of lengthier simulation times and higher memory usages (see later sections on simulation speed and memory usage). Also, for a given number of trials, the accuracy of the statistics and percentiles greatly depends on the shape and nature of the forecast distribution.

If you are not sure how many trials to run for a specific level of accuracy, you can use the precision control feature of to automatically determine the appropriate number of trials to run. For a detailed picture of a simulation’s statistical accuracy, you can run the Bootstrap tool to generate a forecast chart for each statistic or percentile of interest.

The sampling method is the other primary factor governing simulation accuracy. During a simulation, Monte Carlo sampling generates natural, “what-if” type scenarios while Latin Hypercube’s sampling is constrained, but more accurate.
Precision Control

The precision control feature sets how precise you want the forecast statistics to be. Crystal Ball runs the simulation until the selected forecast statistics reach the required precision as determined by calculating confidence intervals.

For more information about confidence intervals, see “Confidence Intervals” on page 41.

Generally speaking, as more trials are calculated, the confidence intervals narrow and statistics become more precise. The precision control feature in uses this characteristic of confidence intervals to determine when a specified precision of a statistic has been reached. It compares the specified precision to the confidence interval. When the calculated confidence interval drops to less than the specified precision, the simulation stops.

For each forecast, you can specify precision in either absolute terms in units of the forecast, or in relative terms as percentages. These settings are made on the Precision tab of the expanded Define Forecast dialog or the Forecast Preferences dialog. Each method, absolute or relative, has its own benefits and drawbacks.

Specifying precision in absolute terms offers greater control of the simulation when the shape and scale of the forecast distribution is roughly known. For example, for a Gross Profit forecast (from the Vision Research model) that ranges from $25.5 to $64.0 million dollars, you can require the precision of the mean to be within plus or minus $100,000 or some other convenient measure of accuracy. However, with the same forecast range, an absolute accuracy of $1000 may require an unreasonably large number of trials to reach. So, the drawback of using absolute precision is that it may require experimentation to determine reasonable accuracy values.

Specifying precision in relative terms offers greater control of the simulation when the shape and scale of the forecast distribution are largely unknown and you are interested in the accuracy only as it relates to the overall distribution itself. In the previous Gross Profit example, you may not know or care if the distribution ranges from $25,500 to $64,000 dollars or from $25.5 to $64.0 million dollars. You may require only that the simulation’s estimate of the mean fall within plus or minus 5% of itself.

You may encounter the drawback of using relative precision when the forecast statistic is close to zero. For example, suppose a forecast’s distribution straddles the break-even point of zero. A relative precision of 5% of the mean, or roughly $0.5 million, results in a very small confidence interval (relative to the full range width of $49.1 million) that may take an unexpectedly large number of trials to satisfy.

Finally, combines the individual forecast precision options with the confidence level value found in the Trials tab of the Run Preferences dialog to calculate confidence intervals. Generally, it is a good idea to leave this value at 95% or 90% so that you can have a high degree of confidence that the precision requirements have been met. However, if you have a large number of forecasts defined with precision control set, you can adjust the confidence level up or down to globally change the accuracy of all forecasts together.

Sampling Method

Selecting Monte Carlo or Latin Hypercube sampling affects how random numbers are generated for individual assumptions.
In almost all cases, Latin Hypercube produces more accurate forecast statistics (especially the mean) given the same number of trials as Monte Carlo, because it is a more consistent sampling method. If you are primarily interested in the accuracy of the statistics, you should select Latin Hypercube as the sampling method in the Sampling tab of the Run Preferences dialog.

If you are primarily interested in evaluating how the spreadsheet model behaves under various what-if scenarios, you should select Monte Carlo as the sampling method. Monte Carlo produces assumptions with the most randomness and simulates real-life situations the best.

## Simulation Speed

Monte Carlo simulations can be very time-consuming. You can change a number of factors that affect the speed of simulations. The factors are listed below in order of importance:

1. **Change the Speed preferences.**
   
   In the Speed tab of the Run Preferences dialog are options that can substantially increase the speed of the simulation. Of course, if the Extreme Speed setting is available, you can experience dramatic cuts in run times.

   If the model is incompatible with Extreme speed or the license does not include it, you can, in increasing order of helpfulness:

   - Redraw forecasts less often (available in Extreme speed).
   - Update worksheets only every few seconds or so.
   - Suppress chart windows during simulations (available in Extreme speed).
   - Minimize Microsoft Excel workbooks during simulations.
   - Minimize Microsoft Excel workbooks and suppress chart windows (the fastest combination of settings).

2. **Use the precision control feature.**

   The precision control feature can be used to either increase the precision of the simulations or increase the speed of the simulations. If you set the confidence level to a high number, the simulations will be more precise, but may run significantly longer. However, if you do not need as precise a result, you can set the confidence level to a lower number and the simulation speed will increase.

   Using this feature to speed up the model will require you to experiment with different confidence levels.

3. **Reduce the size of the model by reducing the number of assumptions, forecasts, and correlations.**

   Large models require more time per trial. For example, a model that takes 3 or 4 seconds per recalculation cycle will take up to an hour to simulate 1,000 trials.

   Greater numbers of assumptions and forecasts slow the simulation, especially if the assumptions and forecasts are scattered across many spreadsheets in the model. Start by examining the structure and nature of the model to locate possible efficiencies. You can also use the sensitivity feature or the Tornado Chart tool to determine which assumptions...
Contribute the least amount of uncertainty to the most important forecasts. Freeze or eliminate the least important assumptions from the simulation.

Correlated assumptions can also consume a significant amount of processing time; the time grows geometrically as the number of correlated assumptions increases.

4. Reduce the use of other applications.
   Quitting other applications and closing or minimizing windows can be helpful in reducing overhead and increasing simulation speed.

5. Increase the system’s RAM.
   The amount of RAM in the computer has a large effect on the speed of simulations. Modern operating systems give applications such as spreadsheets the appearance of additional RAM through the use of virtual memory.

   Virtual memory enables you to run a greater number of applications than would otherwise be possible, but slows down overall processing speed because the system is frequently accessing the hard drive. If you hear the hard disk being used during a simulation, there may not be enough RAM to hold all parts of the simulation. Buying more RAM or turning off virtual memory (if possible) are solutions to this problem.

**Sample Size**

The sample size setting is located on the Sampling tab of the Run Preferences dialog. Sample size, which is initially set to 500, affects Latin hypercube sampling. It divides each assumption’s distribution into a number of intervals of equal probability. The sample size governs the number of intervals for each distribution. Crystal Ball generates an assumption value for each interval according to the interval’s probability distribution.

While any sample size greater than 100 should produce sufficiently acceptable results, you can set this number higher to maximize accuracy. There is no absolute limit to sample size, although samples greater than 100,000 work best with at least 1 GB RAM and may take a long time to run. The increased accuracy resulting from the use of larger samples, however, requires additional memory. If memory becomes an issue, reduce the sample size and consider adding more RAM.

**Correlated Assumptions**

There are practical limits to the size of correlation matrixes in a workbook. For matrixes built with pairwise correlations (with or without coefficient cell references), the practical limit is about 1,000 correlations per matrix per worksheet (for example, a 50x50 fully specified matrix or a matrix of 1,000 serially correlated assumptions). For linked matrixes, the practical limit is about 500 assumptions per matrix per worksheet (equivalent to approximately 125,000 pairwise correlations).

To minimize the size of matrixes:
- Consider leaving assumptions with coefficients close to 0.0 as uncorrelated.
• Consider replacing assumptions with coefficients close to 1.0 or -1.0 with a formula in the spreadsheet.

When the simulation starts, Crystal Ball checks each correlation matrix for consistency. (Usually, the larger the correlation matrix, the greater your chance of inadvertently creating inconsistencies.) If one or more matrixes are found to be inconsistent, Crystal Ball determines whether adjustments to the correlation coefficients are possible. This process may take a long time, depending on the number and size of the correlation matrixes.

If adjustments to the correlation coefficients are possible, the simulation stops with a dialog prompting you to select a course of action. Choose from the following:

• Click **OK** to continue the simulation with adjusted coefficients.

• Click **Edit Correlations** to view or edit the adjusted correlations prior to running the simulation. If you make any changes, Crystal Ball rechecks the matrix for consistency.

• To terminate the simulation, click **Cancel**.

For the first two options, the original correlation coefficients in the workbook are replaced permanently with the adjusted ones. Note that pairwise correlations with coefficient cell references are not adjusted.

After a simulation, you can use the **Create Report** feature to list original and adjusted correlation coefficients for each assumption.

### Crystal Ball and Multiple-processor Computers

**Subtopics**

• [Crystal Ball and Multiple Processors](#)

• [Crystal Ball and Multithreading](#)

In general, users should see a linear increase in simulation speed with an increase in processor speed. Using a computer with multiple processors and multiple threads can further enhance performance as described in the following sections.

### Crystal Ball and Multiple Processors

In computers with two or more processors, users should see a slight increase in simulation speed because chart updating occurs on a separate thread. Typically, Extreme Speed is faster than Normal Speed, even with multiple processors. In Normal Speed, when a seed value is set and multiple processors are used with a model that has embedded distribution functions (CB.Normal, and so on) in the Microsoft Excel spreadsheet, results may not be exactly the same between simulations. However the results are valid. This is because Microsoft Excel does not guarantee execution order in this case. This can be corrected by running in Extreme Speed or Normal Speed with a single thread.
Crystal Ball and Multithreading

If you are using a computer with multiple processors—for example, dual cores—you can use a new “multithreaded” option in Microsoft Excel 2007 or later to split up spreadsheet model recalculations into separate tasks. These tasks can then be run independently on each processor to speed up the overall recalculation time. Since time for one calculation is reduced, the time to run an entire simulation is also reduced. For two processors, the increase in simulation speed can be roughly anywhere from 10% to 50%, depending on the model. Crystal Ball uses Microsoft Excel’s multithreading setting by default when performing simulation calculations in Microsoft Excel 2007 or later on multiple-core or multiple-processor computers.

To activate multithreading in Microsoft Excel 2007 or later:

1. Click the Office button.
2. Select Microsoft Excel Options, then Advanced.
3. In the Advanced Options dialog, scroll to the Formulas group and then select Enable multi-threaded calculation.
4. Click OK to accept the setting and close the dialog.

To use multithreading efficiently, you should be working with a spreadsheet model that:

- Is large (that is, it takes more than 0.5 sec for each recalculation).
- Can easily be divided into separate tasks by Microsoft Excel 2007 or later (for example, it may have separate chains or groups of formulas that do not depend on each other).
- Is running at Normal Speed, since Extreme Speed does not increase benefits with this feature.

Depending on model size, it is possible that performance can be improved on multiple-core or multiple-processor computers by manually disabling the use of multithreading. In general, smaller models run more slowly with multithreading and larger models run faster. Changing this setting on single-core or single-processor computers has no impact. To disable multithreading in Microsoft Excel 2007 or later before running models, consult the Microsoft Excel documentation.
Overview

Extreme speed, available in Crystal Ball Decision Optimizer, runs simulations up to 100 times faster than Normal speed. Extreme speed makes it more practical to use Crystal Ball tools that run multisimulations or to seek optimal solutions with OptQuest in a reasonable length of time.

The Extreme Speed feature uses PSI Technology, a high-speed, Microsoft Excel-compatible Polymorphic Spreadsheet Interpreter for running simulations on workbooks. This technology was developed by Frontline Systems, the maker of Microsoft Excel’s Solver add-in. PSI Technology supports nearly all of the 320 standard Microsoft Excel functions, including the financial, statistical, and engineering functions that are part of the Analysis Toolpak.

By default, is set to use Extreme speed when initially loaded. If a model is not compatible with Extreme speed, a dialog offers the opportunity to downshift to Normal speed temporarily for that simulation. below, explains conditions that cause a model to be incompatible with Extreme speed. The simulation speed can be changed using the Speed tab in the Run Preferences dialogs.

This section contains the following main topics:

- “Compatibility Issues” on page 20
- “Calculation Differences in Extreme Speed” on page 26
- “Other Important Differences” on page 27
- “Maximizing the Benefits of Extreme Speed” on page 29

Note: Because the Extreme Speed feature uses the Polymorphic Spreadsheet Interpreter, certain Extreme Speed functions may return slightly different results than the corresponding Microsoft Excel function for extreme values of any input arguments. For example, this can occur with statistical and inverse distribution functions.
Compatibility Issues

Subtopics

- Multiple-Workbook Models
- Circular References
- Crystal Ball Microsoft Excel Functions
- User-Defined Functions
- Running User-Defined Macros
- Special Functions
- Undocumented Behavior of Standard Functions
- Incompatible Range Constructs
- Data Tables

While Extreme speed can greatly decrease simulation run times, not all models are compatible with Extreme speed. When you start a simulation, detects if the spreadsheet is compatible with Extreme speed and warns you if it is incompatible. If you choose, you can run the simulation in Normal speed using standard Microsoft Excel, or you can change the spreadsheet model to correct the incompatibility.

This section lists functions and formula constructs that are not compatible with Extreme speed and suggests workarounds. Besides the issues listed below, some differences may also exist due to undocumented Microsoft Excel behaviors, changes in the newest versions of Microsoft Excel, and so on. Also, small differences in the last few decimal places of certain built-in function values should be expected, due to minor algorithmic differences in the way formulas may be computed.

It is important to notice that incompatibilities in functions and formula constructs only concern the cells involved in the calculation of a forecast cell. If cells that are not part of that calculation path have incompatibilities, these are not detected and the simulation is allowed to run.

This section discusses the following compatibility issues:

- “Multiple-Workbook Models” on page 20
- “Circular References” on page 21
- “Crystal Ball Microsoft Excel Functions” on page 21
- “User-Defined Functions” on page 22
- “Running User-Defined Macros” on page 24
- “Special Functions” on page 24
- “Undocumented Behavior of Standard Functions” on page 24
- “Incompatible Range Constructs” on page 25
- “Data Tables” on page 26

Multiple-Workbook Models

Extreme speed can now run simulations on multiple workbooks. If you are running in Extreme speed and the workbook contains external references to cells in other closed workbooks, obtains
the current value from those workbooks. References to cells in other open workbooks are dynamically updated if those cells depend on one or more assumptions. If the external reference is part of a formula (not a simple external reference), this is not compatible with Extreme speed:

- Example message: "Unable to interpret formula at cell location [Book1.xls]Sheet1!A1. (Code #5524 - Complex external reference)"
- Workaround: If possible, consolidate all of the variables and formulas from a multiple-workbook model containing data into a single workbook.

Circular References

Circular references within a model are supported as long as Iteration is selected on the Tools, then Options, then Calculation tab in Microsoft Excel.

(In Microsoft Excel 2007 or later, click the Office button and select Microsoft Excel Options, then Formulas, and then select Enable Iterative Calculation in the Calculation Options group.)

If detects a circular reference and Iteration is not selected, this error is displayed:

- Workaround: Stop the simulation and select Iteration on the Tools, then Options, then Calculation tab.

In Extreme speed, circular references with short Iteration setting may not match Microsoft Excel’s values because of differences in calculation algorithms. For most consistent results, set Iteration to at least 1,000.

However, if a circular reference is non-converging, its results can differ greatly when run at both Extreme and Normal speed, regardless of the Iteration setting. If a circular reference is non-converging, this error message is displayed:

- Example message: "Unable to run in Extreme speed for the following reason: Circular references do not converge, results cannot be guaranteed to match Normal speed. To bypass this message, turn off "Stop on calculation errors" in the Run Preferences dialog. (Code #5545)"
- Workaround: There is no workaround. Review the formulas in the workbook that have created this circular reference and look for a problem that keeps this circular reference from converging to a single value.

Simulations with circular references run in non-vectorized mode. For this reason, they will probably run more slowly than simulations without circular references.

Crystal Ball Microsoft Excel Functions

The following spreadsheet functions are handled normally:

- CB.IterationsFN
distribution functions (such as CB.Binomial)

These functions are not supported in Extreme speed during a simulation:
- CB.GetForeStatFN
- CB.GetForePercentFN
- CB.GetRunPrefsFN
- CB.GetAssumPercentFN
- CB.GetCertaintyFN

While running in Extreme speed, all values for these functions return #VALUE. At the end of the simulation, performs a final recalculation on the model so these functions are evaluated properly. Normally, this should not present a problem unless one of these functions was defined as a forecast and you are expecting a valid value to be computed during the simulation. If one of these Get functions feeds into a forecast during a simulation, this is not compatible with Extreme speed:

- Example message: "Unsupported Microsoft Excel or Crystal Ball function at cell location [Book1.xls]Sheet1!A1. (Code #5539)"
- Workaround: Defining forecasts on statistical functions that are dependent on other forecasts is generally not a good modeling practice. If you need to have a forecast cell defined on a statistical result from another forecast, use the Auto Extract feature for the dependent forecast instead of using one of the above functions.

User-Defined Functions

Calls to user-defined or third-party functions are allowed. The functions may be written in Visual Basic or they may be in XLL or COM Automation DLL libraries that have been opened in Microsoft Excel.

This section includes the following topics:
- “Pure Functions” on page 22
- “Range Arguments” on page 23
- “Volatile Functions and Array Arguments” on page 23

Pure Functions

To be compatible with Extreme Speed, user-defined functions must be “pure”. A pure function is one that computes its value solely on the basis of values passed to it as arguments. A function that is not pure may reference global data not passed as an argument. For example, it may get the value of a worksheet cell or a defined name and use this as an input to compute the function value. If the global data depends on the assumptions — if, for example, it is a worksheet cell with a formula computed from the assumptions — it will have a distribution of values in Normal Speed, but it will appear deterministic (have a single value) in Extreme Speed. This is because
the worksheet cells change on every trial in Normal Speed, but they do not change in Extreme Speed.

**Range Arguments**

Range arguments in user-defined functions are only compatible with Extreme Speed when they are handled as Variant types. For example, for a function called on the worksheet as =MyFunc(A1:E4, 5, 4):

```vba
Function MyFunc (MyData As Variant, Rows As Long, Cols As Long) As Double
    For I = 1 to Rows
        For J = 1 to Cols
            MsgBox MyData(I, J) 'or otherwise work with the cell range as an array
        Next J
    Next I
End Function
```

**Volatile Functions and Array Arguments**

User-defined functions whose arguments are static (their values do not change during a simulation) are not called by Extreme Speed unless the Volatile property of the function has been set.

When a user-defined function is encountered in Extreme Speed, first checks to see if the function is Volatile. If is denied access to the VBA project and the user-defined function is not passing array or cell range arguments, treats the function as Volatile.

If access to the project is denied and the call is passing array or cell range arguments, the following message is displayed:

> Cannot interpret a user-defined function with array arguments. You must first select the ‘Trust Access to Visual Basic Project’ checkbox in Microsoft Excel’s macro security settings dialog. See Appendix C in the User’s Guide for more information about this error.

**Cause:** The Microsoft Excel Volatile property has not been set.

➤ In that case, in order for Extreme Speed to determine the state of the Volatile property in Microsoft Excel 2003, you must first make the following setting in Microsoft Excel:

1. **Select Tools, and then Options.**
2. **In the Options panel, click Security.**
3. **Click Macro Security.**
4. **Click Trusted Publishers.**
5. **Select Trust Access To Visual Basic Project.**

In Microsoft Excel 2007 or later, you should not encounter this specific problem with array arguments or any others involving the Analysis ToolPak, since the ToolPak is embedded directly into Microsoft Excel 2007 or later.
However, to set the **Trust access...** property in Microsoft Excel 2007 or later to ensure that the Volatile property is set correctly, follow these steps:

1. Click the Office button.
2. Click the following sequence of buttons and links: **Microsoft Excel Options**, then **Trust Center**, then **Trust Center Settings**, and then **Macro Settings**.
3. On the **Macro Settings** page, under **Developer Macro Settings**, select **Trust access to the VBA project object model**.

### Running User-Defined Macros

In Extreme speed, it is not possible to run user-defined macros — such as CBBeforeTrial, CBAfterTrial, and CBAfterRecalc — during a simulation. Before- and after-simulation macros are allowed, such as CBBeforeSimulation and CBAfterSimulation. If the former group of simulation macros are present, they will be flagged as incompatible:

- **Example message:** "One or more user-defined simulation macros were detected (e.g., CBBeforeTrial, CBAfterTrial, etc.). (Code #5701)"
- Workaround: Use Normal speed to run these user-defined macros during a simulation.

### Special Functions

A small group of Microsoft Excel functions are not supported in Extreme speed: CALL, CELL, GETPIVOTDATA, INFO, HYPERLINK, REGISTER.ID, and the CUBE functions added in Microsoft Excel 2007 or later (CUBEMEMBER, CUBEVALUE, CUBESET, CUBESETCOUNT, CUBERANKEDMEMBER, CUBEPROPERTY, CUBEKPIMEMBER). Any forecast formulas that contain one or more of these functions will be flagged as incompatible:

- **Example message:** "Unsupported Microsoft Excel or Crystal Ball function at cell location [Book1.xls]Sheet1!A1. (Code #5539)"
- Workaround: Avoid using these functions if you want to run the model in Extreme speed.

### Undocumented Behavior of Standard Functions

Microsoft Excel allows certain undocumented argument constructs for standard functions. Some of these constructs that were previously unsupported in Extreme speed are now allowed, for example:

\[=\text{SUMPRODUCT}(A1:A10*B1:B10)\]
\[=\text{SUMPRODUCT}(A1:A10/B1:B10)\]

However, some of these argument constructs are still not supported in Extreme speed and will be flagged as incompatible:

- **Example message:** "Unsupported Microsoft Excel or Crystal Ball function at cell location [Book1.xls]Sheet1!A1. (Code #5539)"
- Workaround: For best results, always use standard syntax with fully valid arguments.
Incompatible Range Constructs

The following sections discuss Microsoft Excel range constructs that are not supported in Extreme speed:

- “Dynamic Ranges” on page 25
- “Labels in Formulas That Are Not Defined Names” on page 25
- “Multiple Area References” on page 25
- “3-D References” on page 26

Dynamic Ranges

Extreme speed does not support dynamic ranges, where the OFFSET function is used on one or both sides of the range constructor. For example, =AVERAGE(Cellname1:OFFSET(Cellname2, x, y)).

- Example message: "Unable to interpret formula at cell location [Book1.xls]Sheet1!A1. (Code #5504 - Invalid token)"
- Workaround: Avoid using the OFFSET function to construct dynamic ranges.

Labels in Formulas That Are Not Defined Names

Extreme speed supports defined names and their use in formulas, but not the Microsoft Excel Accept labels in formulas option which allows cell labels to be used in formulas without defining them as names:

- Example message: "Unable to interpret formula at cell location [Book1.xls]Sheet1!A1. (Code #5514 - Undefined identifier)"
- Workaround: For best results, use defined names instead of cell labels in formulas.

Multiple Area References

Extreme speed does not support multiple-area references such as (A1:A5,B1,C1:E1) except as used in standard functions such as SUM that accept a variable-length argument list of cell ranges. The only supported use of the comma is as the separator in the argument list of a standard function, not as a cell-range union operator. A defined name whose value is a multiple-area reference is not accepted:

- Example message: "Multi-area reference not supported at cell location [Book1.xls]Sheet1!A1. (Code #5525)"
- Workaround: For best results, avoid using multiple-area references when defining names or as arguments to any functions except those that accept a variable-length, comma-separated list of cell ranges.
3-D References

Extreme speed does not support 3-D references, where a cell range – used as an argument in a function call, for example – spans multiple workbooks:

- Workaround: For best results, avoid using 3-D cell references.

Data Tables

While previous versions of did not support the Microsoft Excel TABLE function, this data table feature is now supported in Extreme speed. You can use one- and two-variable data tables in models that will run in Extreme speed.

Calculation Differences in Extreme Speed

Subtopics

- Differences in Microsoft Excel Functions by Run Mode
- Calculation Differences and the Compare Run Modes Tool

Ordinarily, there may be slight numeric differences in simulation results between Extreme and Normal speed modes. These differences are primarily due to rounding errors and are usually in the relative range of 1e–12 to 1e–15. In certain circumstances, as for extremely large models containing thousands of formulas, these slight differences can compound and grow in magnitude. Differences can occur in Excel functions and in general formula calculations, as described in the following topics:

- “Differences in Microsoft Excel Functions by Run Mode” on page 26
- “Calculation Differences and the Compare Run Modes Tool” on page 26

Differences in Microsoft Excel Functions by Run Mode

The Extreme Speed feature uses a proprietary spreadsheet interpreter to increase calculation speed. This interpreter calculates Microsoft Excel functions independently of how they are calculated in Normal speed using the native Microsoft Excel calculation code. For this reason, some functions may generate slightly different results in Extreme and Normal speed.

Functions that may have different results are XIRR, XNPV, IRR, PMT, CUMIPMT, and CUMPRINC.

Calculation Differences and the Compare Run Modes Tool

If you are concerned about possible differences in model calculations between Extreme and Normal speed, Crystal Ball provides a special Compare Run Modes tool that you can use to
compare the results between the two run modes. For details, see the Oracle Crystal Ball User’s Guide.

Note: Because of variations in the random number seeds, you may see differences in comparison results if you use Microsoft Excel’s RAND or Crystal Ball’s probability functions (such as CB.Uniform) in the model.

Other Important Differences

This section discusses several other differences between Extreme speed and Normal speed:

- “OptQuest and Other Crystal Ball Tools” on page 27
- “Precision Control and Cell Error Reviewing” on page 27
- “Spreadsheet Updating” on page 28
- “Very Large Models” on page 28
- “Memory Usage” on page 28
- “Spreadsheets with No Data” on page 28

OptQuest and Other Crystal Ball Tools

OptQuest and certain other Crystal Ball tools run multiple simulations of the spreadsheet model back-to-back. Since Extreme speed enhances the performance of each simulation, you will see an overall speed improvement when using one of these tools. All of the compatibility and other differences apply to these tools as well.

Precision Control and Cell Error Reviewing

Extreme speed uses a bursting feature to optimize the balance between speed and system responsiveness. Since bursts are usually several hundreds or thousands of trials at a time, this feature affects a number of features like precision control, confidence testing, and checking for cell errors.

Crystal Ball tests for precision control (and performs confidence testing in OptQuest) after each burst of several hundred or thousands of trials. In Normal speed, these checks are performed every 50 trials by default. As a result, you may see the simulations run more trials than usual when these features are turned on. Generally speaking, these features are not as critical to simulation performance in Extreme speed as they are in Normal speed.

Because of bursting, cannot check cell errors after every trial as it does in Normal speed, only at the end of a burst. If cell errors occur in forecast cells (such as “Divide by zero” or “#VALUE”) and the Run Preferences, then Stop On Calculation Errors option is selected, the simulation stops at the end of the next burst.
Spreadsheet Updating

While running simulations in Extreme speed, cells in the spreadsheet model appear to update every second or so after each burst. This is normal and does not affect simulation results. Single-stepping mode still updates cell values for every step.

Very Large Models

Very large models (with greater than 10,000 formulas, for example), may take many seconds or minutes in Extreme speed to analyze the model and convert it into a form for high-speed simulation. When this happens, the following progress dialog opens.

Figure 1 Extreme Speed Progress Dialog Box

If you choose, you can cancel the simulation by pressing the Ctrl+Break keys or wait for the process to finish.

Memory Usage

Running simulations in Extreme speed is extremely memory-intensive. Depending on the computer, you may encounter issues with memory usage when running long simulations or working with large models.

During a long simulation (greater than 100,000 trials), you may also see that the computer’s virtual memory is being used. For example, you may notice that the hard drive is being accessed continually and the simulation starts to slow down. No message is given when this condition occurs. If such a slowdown occurs and it is absolutely necessary that additional trials be run, you can try to disable the saving of assumption values (see the Run Preferences, then Options tab) and then restart the simulation.

If you run an Extreme Speed simulation multiple times on a large or complex model, you may find that you receive the Sufficient memory may not... warning dialog. This dialog lets you disable the Store Assumption Values run preference and change the number of trials to run. Notice that these changes in the warning dialog reset those preferences in the Run Preferences dialog on the Trials and Options tabs. They are then applied to other models unless you reset them after the large model has run.

Spreadsheets with No Data

Spreadsheets without data always run in Normal speed even if Extreme speed is selected in the Run Preferences dialog.
This could happen if you have spreadsheet functions entered in the model but have not defined assumptions, forecasts, or decision variables.

Maximizing the Benefits of Extreme Speed

To get maximum benefit from running in Extreme speed, you should be aware of several issues in the models that could potentially cause simulations to run more slowly than expected.

When you first start a simulation in Extreme speed, the model is analyzed and converted into a special form for high-speed processing. This form involves the use of a technique called *vectorization* to process batches of formula results together for maximum efficiency. Certain functions and formula constructs in the model can interfere with this vectorization process. Some of the more common cases are discussed in the following sections:

- “String Intermediate Results in Formulas” on page 29
- “Calls to User-Defined Functions” on page 30
- “Dynamic Assumptions” on page 30
- “Microsoft Excel Functions” on page 30

If one or more cases are detected, displays the message “non-vectored” on the Microsoft Excel status bar while the simulation is running. Also, additional information is usually written to the log file, which can be found with the suffix CB Log and extension .txt in:

```
C:\Documents and Settings\username\Application Data\Oracle\Crystal Ball\11.0\Logs\CB Log_x.txt
```

**Note:** If you have a very large number of trials, the virtual memory feature of Microsoft Windows is activated. This can slow down processing. Too, if you have large numbers of forecasts and assumptions, managing these large arrays of values can take extra time. If you think array management could be an issue, try clearing Store Assumption Values For Sensitivity Analysis on the Options tab of the Run Preferences dialog.

String Intermediate Results in Formulas

In Microsoft Excel, it is possible to construct formulas that produce an intermediate character string result, which is used in later formulas to produce a numeric result. For maximum speed gains, you should avoid the use of strings in the formulas. Strings in formulas will not be flagged as incompatible, but will cause simulations to run much slower depending on the number of string formulas in the model.

- **Workaround:** Some simple strings like “Yes” and “No” can be replaced with 1 and 0 values, or True and False Boolean values.
**Calls to User-Defined Functions**

Calls to user-defined or third-party functions cannot be vectorized and will cause simulations to run slower.

- **Workaround:** There is no workaround for this case other than minimizing the use of these functions.

**Dynamic Assumptions**

Assumptions that contain cell references to parameter values that can change during a simulation are called dynamic assumptions. Needs to perform additional validation checks on the parameters for these assumption types during a simulation, causing simulations to run more slowly.

- **Workaround:** If the dynamic assumption feature is not needed, make sure that parameter cell references do not point to other assumption cells in the model.

Cell references in custom distributions are treated as static instead of dynamic when a model is run at Extreme speed.

**Microsoft Excel Functions**

Crystal Ball supports LOOKUP and OFFSET functions in both Normal and Extreme speed. However, if you use more than a thousand in a model, this can slow down the initial parsing time and cause the simulation to run slowly.
Introduction

This chapter provides formulas for the following types of statistics:

- “Measures of Central Tendency” on page 31
- “Measures of Variability” on page 33
- “Other Measures for a Data Set” on page 34
- “Other Statistics” on page 37

It also describes methodology and statistics for:

- “Simulation Sampling Methods” on page 40
- “Confidence Intervals” on page 41
- “Random Number Generation” on page 43
- “Process Capability Metrics Formulas” on page 75

Statistics

This section discusses basic statistics used in Crystal Ball.

Measures of Central Tendency

The measures of central tendency for a data set are mean, median, and mode.
Mean
The mean of a set of values is found by adding the values and dividing their sum by the number of values. The term “average” usually refers to the mean. For example, 5.2 is the mean, or average, of 1, 3, 6, 7, and 9.

Formula:
\[ \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \]

Median
The median is the middle value in a set of sorted values. For example, 6 is the median of 1, 3, 6, 7, and 9 (recall that the mean is 5.2).

If an odd number of values exists, you find the median by ordering the values from smallest to largest and then selecting the middle value.

If an even number of values exists, then the median is the mean of the two middle values.

Mode
The mode is the value that occurs most frequently in a set of values. The greatest degree of clustering occurs at the mode.

The modal wage, for example, is the one received by the greatest number of workers. The modal color for a new product is the one preferred by the greatest number of consumers.

In a perfectly symmetrical distribution, such as the normal distribution (the distribution on the left, below), the mean, median, and mode converge at one point.

In an asymmetrical, or skewed, distribution, such as the lognormal distribution, the mean, median, and mode tend to spread out, as shown in the second distribution (on the right) in the following example (Figure 2).

Figure 2  Symmetrical and Asymmetrical Distributions
Note: When running simulations, forecast data likely will be continuous and no value will occur more than once. In such a case, sets the mode to ‘---’ in the Statistics view to indicate that the mode is undefined.

Measures of Variability

The measures of variability for a data set are variance, standard deviation, and range (or range width).

Variance

Variance is a measure of the dispersion, or spread, of a set of values about the mean. When values are close to the mean, the variance is small. When values are widely scattered about the mean, the variance is larger.

Formula:

\[ \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \]

To calculate the variance of a set of values:

1. Find the mean or average.
2. For each value, calculate the difference between the value and the mean.
3. Square the differences.
4. Divide by \( n - 1 \), where \( n \) is the number of differences.

For example, suppose your values are 1, 3, 6, 7, and 9. The mean is 5.2. The variance, denoted by \( s^2 \), is calculated as follows:

\[
 s^2 = \frac{(1-5.2)^2 + (3-5.2)^2 + (6-5.2)^2 + (7-5.2)^2 + (9-5.2)^2}{5-1} \\
= \frac{40.8}{4} = 10.2
\]

Note: The calculation uses \( n - 1 \) instead of \( n \) to correct for the fact that the mean was calculated from the data sample, thus removing one degree of freedom. This correction makes the sample variances slightly larger than the variance of the entire population.

Standard Deviation

The standard deviation is the square root of the variance for a distribution. Like the variance, it is a measure of dispersion about the mean and is useful for describing the “average” deviation. See the description for the variance in the next section.
For example, you can calculate the standard deviation of the values 1, 3, 6, 7, and 9 by finding the square root of the variance that is calculated in the variance example that follows.

Formula:

\[ s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \] (s)

The standard deviation, denoted as \( s \), is calculated from the variance as follows:

\[ s = \sqrt{10.2} = 3.19 \]

**Coefficient of Variation**

The coefficient of variation provides you with a measurement of how much your forecast values vary relative to the mean value. Because this statistic is independent of the forecast units, you can use it to compare the variability of two or more forecasts, even when the forecast scales differ.

For example, if you are comparing the forecast for a penny stock with the forecast for a stock on the New York Stock Exchange, you would expect the average variation (standard deviation) of the penny stock price to appear smaller than the variation of the NYSE stock. However, if you compare the coefficient of variation statistic for the two forecasts, you will notice that the penny stock shows significantly more variation on an absolute scale.

The coefficient of variation typically ranges from a value greater than 0 to 1. It might exceed 1 in a few cases in which the standard deviation of the forecast is unusually high. This statistic is computed by dividing the standard deviation by the mean.

The coefficient of variation is calculated by dividing the standard deviation by the mean, as follows:

\[ \text{coefficient of variation} = \frac{s}{x} \]

To present this number as a percentage, multiply the result of the coefficient of variation calculation by 100.

**Range (Also Range Width)**

The range minimum is the smallest number in a set of values; the range maximum is the largest number.

The range is the difference between the range minimum and the range maximum.

For example, if the range minimum is 10, and the range maximum is 70, then the range is 60.

**Other Measures for a Data Set**

These statistics also describe the behavior of a data set: skewness, kurtosis, and mean standard error.
**Skewness**

A distribution of values (a frequency distribution) is said to be “skewed” if it is not symmetrical. For example, suppose the curves in the example below represent the distribution of wages within a large company (**Figure 3**).

![Figure 3 Positive and Negative Skewness](image)

Curve A illustrates positive skewness (skewed “to the right”), where most of the wages are near the minimum rate, although some are much higher. Curve B illustrates negative skewness (skewed “to the left”), where most of the wages are near the maximum, although some are much lower.

If you describe the curves statistically, curve A is positively skewed and might have a skewness coefficient of 0.5, and curve B is negatively skewed and might have a -0.5 skewness coefficient.

A skewness value greater than 1 or less than -1 indicates a highly skewed distribution. A value between 0.5 and 1 or -0.5 and -1 is moderately skewed. A value between -0.5 and 0.5 indicates that the distribution is fairly symmetrical.

**Method:**

Skewness is computed by finding the third moment about the mean and dividing by the cube of the standard deviation.

**Formula:**

\[
\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^3 
\]

\[
\frac{s^3}{n}
\]

**Kurtosis**

Kurtosis refers to the peakedness of a distribution. For example, a distribution of values might be perfectly symmetrical but look either very “peaked” or very “flat,” as illustrated below (**Figure 4**).
Suppose the curves in represent the distribution of wages in a large company. Curve A is fairly peaked, because most of the employees receive about the same wage, with few receiving very high or low wages. Curve B is flat-topped, indicating that the wages cover a wider spread.

Describing the curves statistically, curve A is fairly peaked, with a kurtosis of about 4. Curve B, which is fairly flat, might have a kurtosis of 2.

A normal distribution usually is used as the standard of reference and has a kurtosis of 3. Distributions with kurtosis values of less than 3 are described as platykurtic (meaning flat), and distributions with kurtosis values of greater than 3 are leptokurtic (meaning peaked).

Method:

Kurtosis, or peakedness, is calculated by finding the fourth moment about the mean and dividing by the quadruple of the standard deviation.

Formula:

$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^4$$

s

**Mean Standard Error**

The mean standard error statistic enables you to determine the accuracy of your simulation results and how many trials are necessary to ensure an acceptable level of error. This statistic tells you the probability of the estimated mean deviating from the true mean by more than a specified amount. The probability that the true mean of the forecast is the estimated mean (plus or minus the mean standard error) is approximately 68 percent.

**Note:** The mean standard error statistic provides information only on the accuracy of the mean and can be used as a general guide to the accuracy of the simulation. The standard error for other statistics, such as mode and median, probably will differ from the mean standard error.

Formula:

$$\frac{s}{\sqrt{n}}$$
where $s =$ standard deviation and $n =$ number of trials.

The error estimate might be inverted to show that the number of trials needed to yield a desired error:

\[
\hat{n} = \frac{s^2}{\varepsilon^2}
\]

**Other Statistics**

These statistics describe relationships between data sets (correlation coefficient, rank correlation) or other data measurements (certainty, percentile, confidence intervals).

**Correlation Coefficient**

*Note:* Crystal Ball uses rank correlation to determine the correlation coefficient of variables. For more information on rank correlation, see “Rank Correlation” on page 38.

When the values of two variables depend upon one another in whole or in part, the variables are considered correlated. For example, an “energy cost” variable likely will show a positive correlation with an “inflation” variable. When the “inflation” variable is high, the “energy cost” variable is also high; when the “inflation” variable is low, the “energy cost” variable is low.

In contrast, “product price” and “unit sale” variables might show a negative correlation. For example, when prices are low, high sales are expected; when prices are high, low sales are expected.

By correlating pairs of variables that have such a positive or negative relationship, you can increase the accuracy of your simulation forecast results.

The correlation coefficient is a number that describes the relationship between two dependent variables. Coefficient values range between -1 and 0 for a negative correlation and 0 and +1 for a positive correlation. The closer the absolute value of the correlation coefficient is to either +1 or -1, the more strongly the variables are related.

When an increase in one variable is associated with an increase in another, the correlation is called positive (or direct) and is indicated by a coefficient between 0 and 1. When an increase in one variable is associated with a decrease in another variable, the correlation is called negative (or inverse) and is indicated by a coefficient between 0 and -1. A value of 0 indicates that the variables are unrelated to one another. The example below shows three correlation coefficients (Figure 5).
For example, assume that total hotel food sales might be correlated with hotel room rates. Total food sales likely will be higher, for example, at hotels with higher room rates. If food sales and room rates correspond closely for various hotels, the correlation coefficient is close to 1. However, the correlation might not be perfect (correlation coefficient is less than 1). Some people might eat meals outside of the hotel, and others might skip some meals.

When you select a correlation coefficient to describe the relationship between two variables in your simulation, you must consider how closely they are related. You should never need to use an actual correlation coefficient of 1 or -1. Generally, you should represent these types of relationships as formulas on your spreadsheet.

Formula:

\[
\frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \cdot \sqrt{n \sum_{i=1}^{n} y_i^2 - \left( \sum_{i=1}^{n} y_i \right)^2}}
\]

Note: Crystal Ball uses rank correlation, also referred to as Spearman’s rank correlation coefficient, to correlate assumption values. This means that assumption values are replaced by their rankings from lowest to highest value by the integers 1 to \(n\), before computing the correlation coefficient. This method allows distribution types to be ignored when correlating assumptions.

**Rank Correlation**

A correlation coefficient measures the strength of the linear relationship between two variables. However, if the two variables do not have the same probability distributions, they are not likely related linearly. Under such circumstances, the correlation coefficient calculated on their raw values has little meaning.

If you calculate the correlation coefficient using rank values instead of actual values, the correlation coefficient is meaningful even for variables with different distributions.

You determine rank values by arranging the actual values in ascending order and replacing the values with their rankings. For example, the lowest actual value will have a rank of 1; the next-lowest actual value will have a rank of 2; and so on.
Crystal Ball uses rank correlation, also referred to as Spearman’s rank correlation coefficient, to correlate assumptions. The slight loss of information that occurs using rank correlation is offset by two advantages:

- First, the correlated assumptions need not have similar distribution types. In effect, the correlation function is distribution-independent. The rank correlation method works even when a distribution has been truncated at one or both ends of its range.
- Second, the values generated for each assumption are not changed; they are merely rearranged to produce the desired correlation. In this way, the original distributions of the assumptions are preserved.

**Certainty**

The forecast chart shows not only the range of results for each forecast, but also the probability, or certainty, of achieving results within a range. Certainty is the percent chance that a forecast value will fall within a specified range.

By default, the certainty range is from negative infinity to positive infinity. The certainty for this range is always 100 percent. However, you might want to estimate the chance of a forecast result falling in a specific range, say from zero to infinity (which you might want to calculate to ensure that you make a profit).

For example, consider the forecast chart in Figure 6. If your objective is to make a minimum return of $2,000,000, you might choose a range of $2,000,000 to +Infinity. In this case, the certainty is almost 75 percent.

**Figure 6  Certainty of a $2 Million Net Profit**

![Chart showing certainty of a $2 Million Net Profit](image)

**Percentiles**

A percentile is a number on a scale of 0–100 that indicates the percent of a distribution that is equal to or less than a value (by default). Standardized tests usually report results in percentiles. If you are in the 95th percentile, then 95 percent of test takers had either the same score or a lower score. This number does not mean that you answered 95 percent of the questions correctly.
You might have answered only 20 percent correctly, but your score was better than, or as good as, 95 percent of the other test takers’ scores.

Crystal Ball calculates percentiles of forecast values using an interpolation algorithm. This algorithm is used for both discrete and continuous data, resulting in the possibility of having real numbers as percentiles for even discrete data sets. If an exact forecast value corresponds to a calculated percentile, accepts that as the percentile. Otherwise, proportionally interpolates between the two nearest values to calculate the percentile.

Note: When calculating medians, does not use the proportional interpolation algorithm; it uses the classical definition of median, described in “Median” on page 32.

Percentiles for a normal distribution look like the following figure (Figure 7).

![Normal Distribution with Percentiles](image)

**Simulation Sampling Methods**

During each trial of a simulation, selects a random value for each assumption in your model. selects these values based on the Sampling dialog box (displayed when you select Run, then Run Preferences). The sampling methods:

- Monte Carlo: Randomly selects any value from the defined distribution of each assumption.
- Latin Hypercube: Randomly selects values and spreads them evenly over the defined distribution of each assumption.

**Monte Carlo Sampling**

Monte Carlo simulation randomly and repeatedly generates values for uncertain variables to simulate a model. The values for each assumption’s probability distribution are random and totally independent. In other words, the random value selected for one trial have no effect on the next random value generated.

Monte Carlo simulation was named for Monte Carlo, Monaco, whose casinos feature games of chance such as roulette, dice, and slot machines, all of which exhibit random behavior.
Such random behavior is similar to how Monte Carlo simulation selects variable values at random to simulate a model. When you roll a die, you know that a 1, 2, 3, 4, 5, or 6 will come up, but you do not know which for any particular trial. It is the same with the variables that have a known range of values and an uncertain value for any particular time or event (for example, interest rates, staffing needs, stock prices, inventory, phone calls per minute).

Using Monte Carlo sampling to approximate the true shape of the distribution requires more trials than Latin Hypercube.

Use Monte Carlo sampling to simulate “real world” what-if scenarios for your spreadsheet model.

**Latin Hypercube Sampling**

In Latin Hypercube sampling, divides each assumption’s probability distribution into nonoverlapping segments, each having equal probability, as illustrated below (Figure 8).

![Figure 8 Normal Distribution with Latin Hypercube Sampling Segments](image)

While a simulation runs, selects a random assumption value for each segment according to the segment’s probability distribution. This collection of values forms the Latin Hypercube sample. After has sampled each segment exactly once, the process repeats until the simulation stops.

The Sample Size option (displayed when you select Run Preferences, then Sample), controls the number of segments in the sample.

Latin Hypercube sampling is generally more precise when calculating simulation statistics than is conventional Monte Carlo sampling, because the entire range of the distribution is sampled more evenly and consistently. Latin Hypercube sampling requires fewer trials to achieve the same level of statistical accuracy as Monte Carlo sampling. The added expense of this method is the extra memory required to track which segments have been sampled while the simulation runs. (Compared to most simulation results, this extra overhead is minor.)

Use Latin Hypercube sampling when you are concerned primarily with the accuracy of the simulation statistics.

**Confidence Intervals**

Because Monte Carlo simulation uses random sampling to estimate model results, statistics computed on these results, such as mean, standard deviation and percentiles, always contain some kind of error. A confidence interval (CI) is a bound calculated around a statistic that attempts to measure this error with a given level of probability. For example, a 95 percent confidence interval around the mean statistic is defined as a 95 percent chance that the mean will be contained within the specified interval. Conversely, a 5 percent chance exists that the
mean will lie outside the interval. Shown graphically, a confidence interval around the mean looks like Figure 9.

Figure 9  Confidence Interval

\[ CI_{\text{min}} \quad \text{Mean} \quad CI_{\text{max}} \]

For most statistics, the confidence interval is symmetrical around the statistic, so that \( x = (\text{CI}_{\text{max}} - \text{Mean}) = (\text{Mean} - \text{CI}_{\text{min}}) \). This accuracy lets you make statements of confidence such as “the mean will lie within the estimated mean plus or minus \( x \) with 95 percent probability.”

Confidence intervals are important for determining the accuracy of statistics, hence, the accuracy of the simulation. Generally speaking, as more trials are calculated, the confidence interval narrows and the statistics become more accurate. The precision control feature of lets you stop the simulation when the specified precision of the chosen statistics is reached, periodically checks whether the confidence interval is less than the specified precision.

Notice that the Bootstrap tool in enables you to calculate the confidence intervals for any set of statistics using empirically-based methods.

The following sections describe how calculates the confidence interval for each statistic.

**Mean Confidence Interval**

Formula:

\[ z \cdot \frac{s}{\sqrt{n}} \]

where \( s \) is the standard deviation of the forecast, \( n \) is the number of trials, and \( z \) is the \( z \) value based on the specified confidence level (to set the confidence level, from Run Preferences, select Trials).

**Standard Deviation Confidence Interval**

Formula:

\[ z \cdot s \cdot \sqrt{\frac{k - 1}{4 \cdot (n - 1)}} \]

where \( s \) is the standard deviation of the forecast, \( k \) is the kurtosis, \( n \) is the number of trials, and \( z \) is the \( z \) value based on the specified confidence level (from Run Preferences, select Trials).

**Percentiles Confidence Interval**

To calculate the confidence interval for the percentiles, instead of a mathematical formula, uses an analytical bootstrapping method.
Random Number Generation

Crystal Ball uses the random number generator described in the following iteration formula as the basis for all nonuniform generators. For no starting seed value, takes the value of the number of milliseconds elapsed since Windows started.

Method: Multiplicative Congruential Generator

This routine uses the iteration formula:

\[ r \leftarrow (62089911 \cdot r) \mod \left(2^{31} - 1\right) \]

Comment:

The generator has a period of length of \(2^{31} - 2\), or 2,147,483,646. This means that the cycle of random numbers repeats after several billion trials. This formula is discussed in detail in the Simulation Modeling & Analysis and Art of Computer Programming, Vol. II, references in the Oracle Crystal Ball User’s Guide bibliography.
Process Capability Tutorials

Subtopics

- Tutorial 1 — Improving Process Quality
- Tutorial 2 — Packaging Pump Design

**Tutorial 1 — Improving Process Quality**

Six Sigma and other process improvement methodologies strive to save costs and time by minimizing process variation and reducing rework. Six Sigma has five phases for gathering information, analyzing it, and implementing suggested improvements: Define, Measure, Analyze, Improve, Control. The following sections in this tutorial show how can be used in the Analyze and Improve phases to discover inconsistencies and reduce variation in a bank’s loan processing system:

- “Overall Approach” on page 45
- “Using the Loan Processing Model” on page 48

“Tutorial 2 — Packaging Pump Design” on page 59, shows how to use in design quality programs such as Design for Six Sigma (DFSS).

**Overall Approach**

Suppose you are assigned the task of improving loan processing quality at a bank. You want to minimize average processing time, keeping below the bank’s target. At the same time, you want to evaluate and improve processing time consistency using the following techniques:

- Define — You plan to talk with stakeholders in various departments to learn how they are currently working, how they view of the problem, and what outcome they hope to see.
Measure — You will determine current processing times and the amount of variation within them.

Analyze — You plan to use to help you discover process capability, or how closely the processing times map to process requirements.

Improve — You will use information gathered in the previous phases to design and implement improvements in processing time and consistency.

Control — You need to ensure permanent improvements by setting up a system of feedback loops to continuously measure process results and change the output if measures exceed desired control targets and control limits.

In the Define and Measure phases, you obtain and use information to build a model. Then, you use to simulate real-world variations that occur in processing loans. Finally, you can use sensitivity analysis and further optimization to arrive at the best balance between time and consistency.

The following sections give details of each phase:

- “Define Phase” on page 46
- “Measure Phase” on page 46
- “Analyze Phase” on page 47
- “Improve Phase” on page 47
- “Control Phase” on page 47

Define Phase

The bank’s loan processing procedures involve six steps. You interview staff involved with each step and learn that marketing and management groups are hoping for completion of all six steps within 96 working hours. Based on these interviews, you establish both the target and upper specification limit (USL) at 96 hours. There is no lower specification limit (LSL), although consistency (the sigma level) seems to be an issue. You decide to obtain more concrete data.

Measure Phase

You document time and time variation data for each processing step, as follows:

- **Customer Inquiry** involves the creation of an initial rate quote following the first customer contact by phone, in-person visit, or the Internet. Historical data indicates this cycle time is lognormally distributed with a mean of one hour and a standard deviation of 0.25 hours.

- **Loan Application** involves the completion of forms by customers, with or without staff assistance. This step takes from one to five eight-hour days, but usually takes three days.

- **Document Verification** and processing occurs in the bank, where a loan specialist reviews credit data, presents loan alternatives to customers, and independently verifies information. This step usually takes between two and four eight-hour days. However, about two out of ten times, the loan is suspended and takes between four and six days for additional reviews and documentation.
Underwriting, or loan approval, usually takes from one hour to 8 hours. All lengths of time have the same likelihood of occurrence.

Loan Closing includes the preparation of the final documentation, locking in an interest rate, and arranging where to deposit funds. Historical data show that this step has a mean of two days, but you do not yet know how the data are distributed.

Loan Disbursement, moving the funds to the customer’s bank, usually takes 16 hours, with some variation. Two-thirds of disbursement times fall within 12 and 20 hours.

After gathering this information, you decide to load it into Microsoft Excel and use to analyze it.

**Analyze Phase**

During the Analyze phase, you create a Microsoft Excel workbook that includes historical data on loan closing times. Then, you can capture the time and variation information for each step as a assumption. Finally, you can define overall processing time as a forecast. When the simulation runs, you can view capability metrics to gain insight into the degree of process variation, or inconsistency. You can also generate a sensitivity chart to learn more about the main sources of inconsistency.

“Using the Loan Processing Model” on page 48, shows how to use a model in the Analyze and Improve phases to accomplish the quality goals.

**Improve Phase**

You can experiment with the effects of changing the main sources of inconsistency. If you have OptQuest, you can also define decision variables and optimize for the shortest processing time and least time variation.

**Control Phase**

You set up a system of measuring cycle times for each step and analyzing them on a regular basis. You run a simulation on real data once a month with a program of corrective actions to take whenever results exceed desired limits by a specified amount.
Using the Loan Processing Model

Subtopics
- Starting Crystal Ball
- Activating the Process Capability Features
- Setting the Sampling Seed Value
- Opening the Example Model
- Reviewing the Parts of the Model
- Reviewing the Assumptions
- Examining the Custom Distribution
- Fitting a Distribution for Step 5
- Reviewing the Forecast
- Running Simulations
- Analyzing Simulations
- Investigating Improvement Possibilities

This section shows how to construct a model that effectively simulates each step of the loan process with dual goals of minimizing processing time and maximizing time consistency. Then, you run the model and analyze the output using Crystal Ball’s process capability features.

The sections listed above describe how to work with the loan processing model and analyze data from it.

Starting Crystal Ball

If is not already started, start it as usual. If the Welcome screen opens, select Quality and then click Use Crystal Ball.

Activating the Process Capability Features

To confirm that the process capability features have been activated and activate them, if necessary:

1. Select Run, and then Run Preferences.
2. Click the Statistics tab in the Run Preferences dialog.
4. Click the Options button to confirm the settings.
   - In the Capability Options panel, set the options to use short-term metrics with a Z-score shift value of 1.5 (the default Short Vs. Long Term settings) and use the default Calculation Method settings.
5. Click OK in both dialogs to accept the current settings and close them.
Setting the Sampling Seed Value

To more closely reproduce the results in this tutorial, you may want to set the sampling seed value on the Sampling tab of the Run Preferences dialog.

Opening the Example Model

Open Loan Process.xls from the Examples folder. To select this example from a list, select Help, then Crystal Ball, then Examples Guide or select Start, then All Programs, then Oracle Crystal Ball, then Examples in the Windows taskbar.

(In Microsoft Excel 2007 or later, select Resources, then Examples Guide in the Help group.)

The workbook opens, as shown in Figure 10.

Reviewing the Parts of the Model

The loan processing spreadsheet models a process improvement effort with two basic goals: meeting the cycle time target as closely as possible and maximizing the sigma level. The model has these parts:

- The flowchart at the top of the model illustrates the process with approximate measured mean times for each step.
- The blue box beneath the flowchart highlights the cycle time target.
- The green boxes in the Simulated Cycle Time column are assumptions that represent each step of the loan processing effort.
- The assumption parameters beside each assumption show the distribution and the parameters entered for each assumption. They are entered in each assumption as cell references relative to cells in column C.
- The blue box at the bottom of the Simulated Cycle Time column is a forecast that represents the total cycle time.

**Reviewing the Assumptions**

The assumption definitions are based on the measurements obtained in the second Six Sigma phase, described in “Measure Phase” on page 46. For example, Step 1, Customer Inquiry, has a mean of 1 hour and a standard deviation of 0.25 hour as discussed in “Measure Phase” on page 46. You can compare the other assumptions with the descriptions in to see how they match.

Notice that Step 3 is defined with a custom distribution. We will examine it in more detail shortly and then define the distribution in Step 5.

**Examining the Custom Distribution**

Because the measurement for Step 3 involves two separate sets of values, it is defined as a custom distribution.

1. To explore this:

2. Select cell C18, the assumption for Step 3.
3. Select Define, and then Define Assumption to display the Define Assumption dialog for that assumption.

   *Figure 11 shows the expanded Define Assumption dialog for Step 3, Document Verification.*

   The dialog has been expanded by clicking the More button.

   This dialog reflects the information from that 80% of the time, cycle time for this step ranges from two to four days (16 to 32 hours) but 20% of the time, cycle time ranges from four to six days (32 to 48 hours).
Click OK or Cancel to close the dialog.

Fitting a Distribution for Step 5

When you first open Loan Process.xls, no assumption has been defined for Step 5, Loan Closing. The description in states that it takes an average of two days (16 hours) to close loans but the distribution is not known.

➢ To create an assumption for Step 5 and fit a distribution to it:

1. Select cell C22.
2. Select Define, and then Define Assumption to display the Distribution Gallery dialog for that cell.
3. Click the Fit button.

The Fit Distribution dialog opens.

4. Type the range A2:A101 or click the cell selector for the Range edit box. Then, click the Data tab of the workbook and select cells A2 through A101.

When you accept the selection, the Fit Distribution dialog opens with the range entered.

5. Leave the other default settings as shown and click OK.

The Comparison Chart dialog opens (Figure 12. The normal distribution appears to be the best fit.
6 Click **Accept** to accept the normal distribution.

The **Define Assumption** dialog opens.

Figure 13 shows that the new assumption for Step 5 is a normal distribution with a mean of 16 and a standard deviation of 4.

7 Rename the assumption to **Loan Closing** and click **OK**.

Now, all assumptions for the Loan Process model are complete.

**Reviewing the Forecast**

The forecast in cell C26 represents total cycle time. To view its formula in the Microsoft Excel formula bar, select cell C26. The formula is \(\text{=SUM(C14:C24)}\) — the sum of the assumptions for Steps 1 through 6.
To view and edit the forecast definition:

1. **Select Define, and then Define Forecast or click the toolbar button.**

   The **Define Forecast** dialog opens.

   **Figure 14**, following, shows that the forecast has a name, a units entry, and a USL, the upper specification limit of 96 hours. These were entered when the example model was first defined.

   ![Define Forecast dialog](image)

   As established in the Define phase of this project, 96 hours is the loan processing target as well as the USL.

2. **Enter 96 in the Target edit box and click OK.**

**Running Simulations**
At this point, the model’s assumptions and forecast are complete.

To run the simulation and generate a forecast chart, click the **Start [Simulation]** button.

**Analyzing Simulations**
When you run the simulation for this model, the Cycle Time forecast chart opens (**Figure 15**, following). Because the forecast is defined with an upper specification limit and a target, the chart is displayed in Split View with the forecast chart beside the process capability metrics. The message with the metrics indicates that the distribution is non-normal (the normality test failed), so capability metrics are calculated from the forecast values.
Notice that marker lines are displayed for the USL and target. Because the USL is defined, the maximum certainty grabber is automatically set to that value. Because no LSL is defined, the minimum certainty grabber is automatically set to –Infinity. The Certainty text box contains 72.31; about 72% of forecasted values fall below the USL.

Looking at the process capability metrics, the forecast distribution is not normal, so the Z scores are not generally appropriate to use. The p(N/C)-above (defects above the USL) is about .28, which confirms the certainty level shown in the forecast chart; more than one-quarter of the forecasted values fall above the USL. The Cpk-upper and Cpk are about 0.21, where you had hoped to see at least 1.00 (the 3-sigma level). The loan specialists you interviewed were correct — cycle time variation is large.

You generate a sensitivity chart (Figure 16) to determine which of the loan processing steps has the most influence on the Cycle Time forecast variance. To do this, select the forecast chart and select Forecast, then Open Sensitivity Chart.

Figure 16 shows that the loan processing step with the greatest effect on cycle time variance is Document Verification (Step 3). Obviously, this is an ideal target for improvement.
Investigating Improvement Possibilities

You wonder how much you would need to reduce variation in Document Verification time before overall quality would improve significantly. You decide to increase the probability of a cycle time of four days or less.

To do this:

1. Click the Reset button.
2. Type a new probability in cell G18. Change it to 90%.
3. Adjust cell G19 downward to match. Change it to 10%.
4. Select cell C18 and open the Define Assumption dialog.

As shown in Figure 17, the two parts of this assumption are both uniform in height over their range, although the first, lower-value one is nine times taller.
This assumption was previously displayed in Figure 11. Because the probabilities in the table are linked to workbook cells, when you adjusted the workbook, you also adjusted the assumption probabilities.

5 Close the Define Assumption dialog and run the simulation.

6 Review the forecast chart in Figure 18.

Crystal Ball sets the upper certainty level at the USL/Target value (96). If the lower certainty grabber is not located at –Infinity, click the lower grabber and drag it toward the Probability axis as shown in Figure 18. Now, the certainty of cycle time falling at or below the USL is about 78%.

You decide to make a drastic change and edit the assumption parameters for Document Verification as follows (change 32 to 24 in the first row; change 90% to 100% and 10% to 0%):
Now, there is 100% probability that all document verification times will fall between two and three days.

Figure 19 shows that when you run with this new assumption definition, the certainty of falling at or below the USL rises to about 93.5% and PPM-total drops to 62,548. Also, the distribution is now normal; it was not normal for the last simulation. Z-USL and Zst-total are 1.53, the sigma level for this simulation.

You review the sensitivity chart for this simulation (Figure 20) and find that the impact of Loan Application is now much greater than Document Verification.
You believe that Document Verification has been pared down about as much as it can be, so you decide to experiment further with making the Loan Application cycle time shorter and more consistent.

To do this:

1. Reset the simulation and change the Step 2 parameters to 8, 16, and 24.
2. Run the simulation.

Results are much better. shows that the mean cycle time has dropped to 74 hours. The certainty of falling below the USL/Target value is 100% and the sigma level has risen to 3.09. Cpk-upper and Cpk have risen over 1 which, given the shape and location of the distribution, confirms that the sigma level is greater than 3.

Figure 21  Forecast Chart after Loan Application Adjustment

When you review the sensitivity chart (Figure 22), Loan Closing and Loan Disbursement are now the main contributors to cycle time variation. You learn that these processes may be difficult to improve further, so you decide to implement improvements to Document Verification and Loan Application cycle times and then move into the Control phase, discussed in “Control Phase” on page 47.
Tutorial 2 — Packaging Pump Design

Design for Six Sigma (DFSS) and other design improvement methodologies are intended to improve product quality at the start of product development.

DFSS has five phases: Define, Measure, Analyze, Design, Validate (or Verify). The ultimate mission of DFSS is to meet general Six Sigma goals of satisfying customer needs while minimizing variation and to accomplish this as early in the development cycle as possible.

The following sections show how DFSS techniques can be used with to simulate various design options for a fluid pump in a food production setting:

- “Overall Approach” on page 59
- “Using the DFSS Liquid Pump model” on page 62
- “Using OptQuest to Optimize Quality and Cost” on page 69

Overall Approach

Suppose you are a pump manufacturer. You want to design a pump for a liquid packaging system that draws processed fluids from a vat into jars at a consistent rate. You want to meet the specified flow rate target and limits while maximizing performance and minimizing costs.

Using the phases of DFSS, you follow this overall design approach:

- **Define** — Obtain an appropriate flow rate target and limits from the customer and recommend a type of pump based on customer considerations
- **Measure** — Consider the elements of the Known Flow Rate equation to determine the pump design features that affect flow rate and obtain information about tolerances, costs, and so on
- **Analyze** — Use Monte Carlo analysis and sensitivity analysis to study how different design elements affect pump performance and affect production quality
- **Design** — Use optimization techniques with defined tradeoffs between flow rate performance and total cost to determine the design parameters that best meet a defined quality level while reducing cost.

- **Validate** — Based on the optimized design parameters, create physical prototypes and test to confirm that this design configuration yields the best-performing, most cost-effective pump.

In the Define and Measure phases, you obtain and use information to build a model. Then, you use to simulate real-world variations that occur in the actual manufacturing of the product. Finally, if you have Crystal Ball Decision Optimizer, you can use sensitivity analysis and further optimization to arrive at the best balance between quality and cost.

The following sections give details of each phase:

- “Define Phase” on page 60
- “Measure Phase” on page 61
- “Analyze Phase” on page 61
- “Design Phase” on page 61
- “Validate Phase” on page 62

The next main section, “Using the DFSS Liquid Pump model” on page 62, works through these phases using an example model.

**Define Phase**

Marketing and production considerations determine the acceptable range of flow rates based on how many units should be produced daily to meet market demand, what is useful production time per day based on plant setup and productivity, and how many pumps are placed in each plant.

The food processing company translates marketing requirements into these engineering specifications and provides them to you:

- Lower specification limit (LSL) = 47.26 ml/sec
- Upper specification limit (USL) = 53.92 ml/sec
- Target = 50.595 ml/sec

Based on these requirements, you propose a displacement pump with a reciprocating piston. Figure 23, following, shows a basic schematic of the system.
Measure Phase

You notice that the variables in the Known Flow Rate equation are piston radius, piston stroke length, backflow rate, and motor speed. You observe that the in-house machine shop controls the piston radius and stroke length while the backflow valve and motor are purchased components.

The machine shop indicates it is capable of 3-sigma quality levels within a tolerance of ±1 millimeter (mm) for the piston radius and stroke length. The purchasing department asks you to use an inexpensive backflow valve and motor.

Analyze Phase

During the Analyze phase, you create a Microsoft Excel workbook that captures the Known Flow Rate equation, characteristics of the pump components used in the equation (piston radius, piston stroke length, motor speed, and backflow rate), flow rate target and limits, machining cost coefficients, plus nine motor and backflow valve cost options.

Then, you define assumption and forecast cells and simulate the design options.

Design Phase

If you have OptQuest, you can also define decision variables for:

- A range of possible piston radii and associated tolerances (related to standard deviation through capability)
- A range of possible piston stroke lengths and associated tolerances,
- The different possible backflow valve and motor options

After this is done, you can set up quality requirements for flow rate and optimize for the least-cost configuration that meets those quality requirements.
Validate Phase

This phase involves the physical creation of a prototype based on simulation forecasts followed by physical testing to validate the design. Because so much preliminary work was done with spreadsheet modeling instead of building additional prototypes, a final design can be developed quickly and at minimal cost.

Using the DFSS Liquid Pump model

Now, see how to construct a model that effectively simulates the design process with dual goals of maximizing quality and minimizing cost. Then, you run the model and analyze the output using Crystal Ball’s process capability features.

This tutorial contains the following review and lesson sections:

- “Starting Crystal Ball” on page 62
- “Activating the Process Capability Features” on page 62
- “Setting the Sampling Seed Value” on page 62
- “Opening the Example Model” on page 62
- “Reviewing the Parts of the Model” on page 63
- “Running Simulations” on page 66
- “Analyzing Simulations” on page 66
- “Adjusting Models” on page 68

If you have Crystal Ball Decision Optimizer, see for a useful optimization example.

Starting Crystal Ball

If is not already started, start it as usual.

Activating the Process Capability Features

To confirm that the process capability features have been activated using the defaults, follow the steps in “Activating the Process Capability Features” on page 48.

Setting the Sampling Seed Value

Set the sampling seed value to 999 in the Run Preferences dialog, Sampling tab, and use Monte Carlo simulation.

Opening the Example Model

Open DFSS Fluid Pump.xls from the Examples Guide list.

The workbook opens, as shown in Figure 24.
Reviewing the Parts of the Model

The liquid pump spreadsheet models a design effort with two basic goals: meeting the flow rate target as closely and consistently as possible and minimizing cost. The following sections describe six main parts of the model, indicated by numbers in Figure 24:

- “Known Flow Rate Formula” on page 64
- “Flow Rate Variable Assumptions” on page 64
- “Flow Rate and Cost Forecasts” on page 64
- “Flow Rate Target and Limits” on page 65
- “Cost Coefficients” on page 65
- “Motor and Backflow Valve Cost Options” on page 65
Known Flow Rate Formula

The Known Flow Rate formula specifies the relevant variables discussed earlier: piston radius, piston stroke length, backflow rate, and motor speed.

Flow Rate Variable Assumptions

The flow rate variable assumptions are assumptions that allow each of the flow rate variables, listed under the Known Flow Rate formula (1), to be included in the Monte Carlo simulation.

To see how these are defined:

1 Select cell K18 and click the Define Assumption button.

Each assumption is defined as a normal distribution with a mean equal to the Nominal Initial Value and a standard deviation equal to the StDev Initial Value. The Mean and Standard Deviation parameters are not entered directly. Instead, they are entered as cell references to the Initial Value cells.

Note: In this model, whenever you see the word “nominal”, it means approximate, designed, or theoretical, as opposed to the actual measured value.

Flow Rate and Cost Forecasts

The flow rate and cost forecasts are forecasts, the output of the Monte Carlo simulation.

The Flow Rate Forecast is based on the Known Flow Rate formula (1).

To view it:

1 Select cell K23.
   The formula in that cell is the Known Flow Rate formula (a function of the flow rate variables) expressed in Microsoft Excel format.
2 Click the Define Forecast button.
3 Notice that the values in cells E24, E25, and E23 are displayed in the Define Forecast dialog as the process capability lower specification limit (LSL), upper specification limit (USL), and target value, respectively.
They are defined as cell references. To demonstrate this, click in one of the text boxes. The related cell reference is displayed. (Note: If you do not see these text boxes in the Define Forecast dialog, process capability features have not been activated in the Run Preferences dialog.)

The Total Cost Forecast is the sum of the Component Cost Function column values in the upper table. To view it:

4 Select cell K25.

Notice its formula is simply the sum of the component costs in cells J18 through J21. No target or limits have been established for this forecast.

Flow Rate Target and Limits

4

The flow rate target and upper and lower specification limits (4) are values defined by the food processing plant in the Define phase of the DFSS process. The previous section shows how they can be used to set the Target, LSL, and USL for the Flow Rate Forecast in cell K23.

Cost Coefficients

5

The cost coefficients table contains values used to calculate the component cost functions, summed to yield the Total Cost Forecast (3).

► To see how these are used to calculate the component cost functions:

1 Select cell J18.

This formula uses the Base Cost Coefficient of the piston, the initial value of the piston radius, the Tolerance Cost Coefficient of the piston, and the initial value of the standard deviation. The cost is directly proportional to the radius squared due to raw material usage and inversely proportional to the standard deviation squared (a common method to relate tolerance to cost).

2 Optional: View the contents of cells J19, J20, and J21.

These formulas are also similarly derived from cost coefficients or nominal costs relevant to each particular flow rate variable.

Motor and Backflow Valve Cost Options

6

The motor and backflow valve cost options are two sets of nine options each provided by the motor and backflow valve vendors. Different performance specifications have different costs associated with them.
Running Simulations

At this point you have created assumptions and forecasts.

➢ To run the simulation and generate forecast charts, click the Start [Simulation] button.

Analyzing Simulations

When you run the simulation for this model, two forecast charts are displayed.

The Total Cost Forecast (Figure 25, following) includes no assumption cells. It is displayed as a single value instead of a range of predicted values. The forecasted total cost is $26.73 for the pump (shown here with the Mean marker line displayed).

Figure 25 The Total Cost Forecast Chart

Because you entered a target and specification limits, the Flow Rate Forecast chart opens in Split View with the forecast chart first and then the process capability metrics, shown in Figure 26.
Notice that marker lines are displayed for the LSL, Target, and USL values. (You may see different marker lines; the Mean and Standard Deviation marker lines have been removed to simplify the screenshot.) Because the LSL and USL are defined, the certainty grabbers are automatically set to those values. According to the value in the Certainty text box for this simulation, 95.34% of forecasted values fall between 47.2617 and 53.9283, the specification limits. The mean is about one standard deviation lower than the target.

The distribution curve is normal. Looking at the process capability metrics, the Z-LSL shows the LSL is 1.63 sigmas below the mean. You were hoping to see a value of at least 3. The Cp capability index is lower than 1, so the short-term potential sigma level is less than 3.

You generate a sensitivity chart (Figure 27) to determine which of the flow rate variables has the most influence on the Flow Rate Forecast.
Figure 16 shows that the flow rate variable with the greatest effect on the Flow Rate Forecast is the piston radius. Reducing the standard deviation of the piston radius should affect the overall flow rate variance and improve the process capability metrics. You wonder how reducing the piston standard deviation by 50% would affect the flow rate forecast quality. You decide to adjust the model and find out.

Adjusting Models

To see the effect of reducing the standard deviation of the piston radius:

1. Select cell H18.
2. Reduce the value by half, to 0.16667.
3. Enter this same value in the Upper Limit cell, I18.

Now, the table of flow rate variables is displayed as shown in Figure 28.

4. Reset and run the simulation again.

The Flow Rate Forecast chart is displayed, similar to Figure 29.
In Figure 29, quality appears much better than that shown in Figure 15. The mean is less than 1.5 standard deviations lower than the target, and the standard deviation is smaller. USL is almost 6 sigmas away from the mean (Z-USL = 5.68) and LSL is over 2 sigmas away (Z-LSL = 2.49). The certainty of the forecasted flow rate falling within LSL and USL is now over 99%.

The Total Cost Forecast chart has also changed, as shown in Figure 30. The quality improvements result in higher costs. Where the total cost per pump was $26.73, now it is $33.47, a 25% increase. You and the team must determine whether to sacrifice quality to lower the cost or whether to test modifications to the other three flow rate variable components as well.

**Using OptQuest to Optimize Quality and Cost**

If you have with OptQuest, an optimization tool, you can use it to maximize quality and minimize cost. To do this, you must consider the relationship between design parameters and cost. And, you must consider the relationship between motor and backflow valve options and cost.

The Total Cost Forecast is defined as the sum of the four component costs, examined in “Flow Rate Variable Assumptions” on page 64. The formulas used to calculate these values are based on the mean and standard deviation of the piston radius, the mean and standard deviation of the piston stroke length, nine different motor options, and nine different backflow valve options.

To apply OptQuest to this model, follow the steps in these sections:

- “Defining Decision Variables” on page 69
- “Identifying Optimization Goals” on page 70
- “Running OptQuest” on page 71

**Defining Decision Variables**

To optimize these variables with OptQuest, a set of decision variables are defined in cells C20, C21, E18, E19, H18, and H19 (Figure 31).
To view the contents of each decision variable, reset and:

1. Select a decision variable cell.

2. Select Define, then Define Decision.

3. In the Define Decision Variable dialog, there is a name, an upper and lower bound, a type setting, and a step size for discrete variables.

Note: If you changed the Piston Radius standard deviation Initial Value and Upper Limit for the last part of the DFSS Pump tutorial, be sure to change them both back to their original values as shown in Figure 31. Also, be sure that these values are included in the decision variable for the Piston Radius standard deviation (Cell H18). Select it as described in steps 1 and 2.

The upper and lower bound for piston machining variables are the same as those in the Lower Limit and Upper Limit tables of Figure 31. For the motor speed and backflow valve variables, the upper and lower bounds are the lowest and highest options (1 and 9, respectively).

In choosing the piston machining values shown in Figure 31, you could identify raw stock availability to define a range of piston radii and lengths for investigation. You could also identify a feasible range of tolerances for machining the radius and length.

**Identifying Optimization Goals**

If you have OptQuest, you can actually perform the optimization. First, identify optimization goals. For this project, they are:

- Minimize cost.
- Reduce variation of flow rate to at least 3 sigma levels. That is, Z-total, expressed in this example as Zst, should be equal to or greater than 3.
Running OptQuest

To optimize these goals in OptQuest, reset the simulation in and:

1. Open the Run Preferences dialog and set Number Of Trials To Run to 1000. For this example, Sampling was set to Latin Hypercube with an Initial Seed Value of 999 (set on the Sampling tab of the Run Preferences dialog).
2. Select Run, and then OptQuest.
3. When the OptQuest wizard opens, click Objectives in the navigation pane to display the Objectives panel if it is not already onscreen.
4. Confirm that the Objectives panel opens, as shown in Figure 32.

Figure 32 Forecast Optimization Defined in OptQuest

5. Click Next until you reach the Options panel (Figure 33, following).
Select Run For 1000 Simulations, and then click Run.

OptQuest starts looking for the best feasible solution. Ultimately a solution will be found that is similar to that shown in Figure 34.

The optimized values should be similar to these:
<table>
<thead>
<tr>
<th>Option</th>
<th>Optimized Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total cost</td>
<td>$33.66</td>
</tr>
<tr>
<td>Zst (sigma level)</td>
<td>3.14</td>
</tr>
<tr>
<td>Backflow valve option</td>
<td>3</td>
</tr>
<tr>
<td>Motor option</td>
<td>2</td>
</tr>
<tr>
<td>Piston radius (mm)</td>
<td>33.38</td>
</tr>
<tr>
<td>Piston radius standard deviation</td>
<td>0.249</td>
</tr>
<tr>
<td>Stroke length (cm)</td>
<td>28.60</td>
</tr>
<tr>
<td>Stroke length standard deviation</td>
<td>0.214</td>
</tr>
</tbody>
</table>

### Capability Metrics List

The following table lists and defines the statistics shown in Capability Metrics view and indicates whether each statistic is displayed for long-term or short-term data; “y” indicates that it is displayed and “n” indicates that it is not displayed. For a discussion of the equations used to calculate each of these statistics, see “Process Capability Metrics Formulas” on page 75.

Z scores are typically reported only for normal data. Always displays Z scores. It is up to the user to determine if the values are appropriate.

#### Table 1  Capability Statistics Calculated by

<table>
<thead>
<tr>
<th>Metric</th>
<th>Long-term</th>
<th>Short-term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>y</td>
<td>y</td>
<td>Mean of the forecast values</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>y</td>
<td>y</td>
<td>Standard deviation of the forecast values</td>
</tr>
<tr>
<td>Cp</td>
<td>n</td>
<td>y</td>
<td>Short-term capability index indicating what quality level the forecast output is potentially capable of producing. It is defined as the ratio of the specification width to the forecast width. If a Cp is equal to or greater than 1, then a short-term 3-sigma quality level is possible.</td>
</tr>
<tr>
<td>Pp</td>
<td>y</td>
<td>n</td>
<td>Long-term capability index indicating what quality level the forecast output is potentially capable of producing. It is defined as the ratio of the specification width to the forecast width. If a Cp is equal to or greater than 1, then a long-term 3-sigma quality level is possible.</td>
</tr>
<tr>
<td>Cpk-lower</td>
<td>n</td>
<td>y</td>
<td>One-sided short-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and lower specification limit over three times the forecast short-term standard deviation; often used to calculate process capability indices with only a lower specification limit.</td>
</tr>
<tr>
<td>Ppk-lower</td>
<td>y</td>
<td>n</td>
<td>One-sided long-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and lower specification limit over three times the forecast long-term standard deviation; often used to calculate process capability indices with only a lower specification limit.</td>
</tr>
<tr>
<td>Metric</td>
<td>Long-term</td>
<td>Short-term</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cpk-upper</td>
<td>n</td>
<td>y</td>
<td>One-sided short-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and upper specification limit over three times the forecast short-term standard deviation; often used to calculate process capability indices with only an upper specification limit.</td>
</tr>
<tr>
<td>Ppk-upper</td>
<td>y</td>
<td>n</td>
<td>One-sided long-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and upper specification limit over three times the forecast long-term standard deviation; often used to calculate process capability indices with only an upper specification limit.</td>
</tr>
<tr>
<td>Cpk</td>
<td>n</td>
<td>y</td>
<td>Short-term capability index (minimum of calculated Cpk-lower and Cpk-upper) that takes into account the centering of the forecast with respect to the midpoint of the specified limits; a Cpk equal to or greater than 1 indicates a quality level of 3 sigmas or better.</td>
</tr>
<tr>
<td>Ppk</td>
<td>y</td>
<td>n</td>
<td>Long-term capability index (minimum of calculated Ppk-lower and Ppk-upper) that takes into account the centering of the forecast with respect to the midpoint of the specified limits; a Ppk equal to or greater than 1 indicates a quality level of 3 sigmas or better.</td>
</tr>
<tr>
<td>Cpm</td>
<td>n</td>
<td>y</td>
<td>Short-term Taguchi capability index; similar to Cpk but considers a target value, which may not necessarily be centered between the upper and lower specification limits.</td>
</tr>
<tr>
<td>Ppm</td>
<td>y</td>
<td>n</td>
<td>Long-term Taguchi capability index; similar to Ppk but considers a target value, which may not necessarily be centered between the upper and lower specification limits.</td>
</tr>
<tr>
<td>Z-LSL</td>
<td>y</td>
<td>y</td>
<td>The number of standard deviations between the forecast mean and the lower specification limit.</td>
</tr>
<tr>
<td>Z-USL</td>
<td>y</td>
<td>y</td>
<td>The number of standard deviations between the forecast mean and the upper specification limit.</td>
</tr>
<tr>
<td>Zst</td>
<td>y</td>
<td>n</td>
<td>For short-term metrics when only one specification limit is defined, equal to Z-LSL if there is only a lower specification limit or Z-USL if there is only an upper specification limit.</td>
</tr>
<tr>
<td>Zst-total</td>
<td>n</td>
<td>y</td>
<td>For short-term metrics when both specification limits are defined, the number of standard deviations between the short-term forecast mean and the lower boundary of combining all defects onto the upper tail of the normal curve. Also equal to Zst-total plus the Z-score shift value if a long-term index is available.</td>
</tr>
<tr>
<td>Zlt</td>
<td>n</td>
<td>y</td>
<td>For long-term metrics when only one specification limit is defined, equal to Z-LSL if there is only a lower specification limit or Z-USL if there is only an upper specification limit.</td>
</tr>
<tr>
<td>Zlt-total</td>
<td>y</td>
<td>n</td>
<td>For long-term metrics when both specification limits are defined, the number of standard deviations between the long-term forecast mean and the lower boundary of combining all defects onto the upper tail of the normal curve. Also equal to Zst-total minus the Z-score shift value if a short-term index is available.</td>
</tr>
<tr>
<td>p(N/C)-below</td>
<td>y</td>
<td>y</td>
<td>Probability of a defect below the lower specification limit; DPUBELOW</td>
</tr>
<tr>
<td>p(N/C)-above</td>
<td>y</td>
<td>y</td>
<td>Probability of a defect above the upper specification limit; DPUABOVE</td>
</tr>
<tr>
<td>p(N/C)-total</td>
<td>y</td>
<td>y</td>
<td>Probability of a defect outside the lower and upper specification limits; DPUTOTAL</td>
</tr>
<tr>
<td>PPM-below</td>
<td>y</td>
<td>y</td>
<td>Defects below the lower specification limit, per million units</td>
</tr>
<tr>
<td>PPM-above</td>
<td>y</td>
<td>y</td>
<td>Defects above the upper specification limit, per million units</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPM-total</td>
<td>Defects outside both specification limits, per million units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSL</td>
<td>Lower specification limit, the lowest acceptable value of a forecast involved in process capability, or quality, analysis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USL</td>
<td>Upper specification limit, the highest acceptable value of a forecast involved in process capability analysis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>The ideal target value of a forecast involved in process capability analysis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z-score shift</td>
<td>An optional shift value to use when calculating long-term capability metrics. The default, set in the Capability Options panel, is 1.5.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Process Capability Metrics Formulas**

The process capability metrics are provided to support quality improvement methodologies such as Six Sigma, Design for Six Sigma (DFSS), and Lean Principles. They appear in forecast charts when a forecast definition includes a lower specification limit (LSL), upper specification limit (USL), or both. Optionally, a target value can be included in the definition.

The following sections describe capability metrics calculated by Crystal Ball. In general, capability indices beginning with C (such as Cpk) are for short-term data, and long-term equivalents begin with P (such as Ppk).

**Note:** Unless otherwise stated, formulas use the mean and standard deviation of the best-fitted non-normal distribution of the data when normality checks fail for a dataset and capability statistics are needed.

**Cp**

Short-term capability index indicating what quality level the forecast output potentially is capable of producing. It is defined as the ratio of the specification width to the forecast width. If a Cp is equal to or greater than 1, then a short-term 3-sigma quality level is possible.

Formula:

$$C_p = \frac{USL - LSL}{6\sigma}$$

**Pp**

Long-term capability index indicating what quality level the forecast output is potentially capable of producing. It is defined as the ratio of the specification width to the forecast width. If a Pp is equal to or greater than 1, then a short-term 3-sigma quality level is possible.

Formula:
Cpk-lower
One-sided short-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and lower specification limit over three times the forecast short-term standard deviation; often used to calculate process capability indices with only a lower specification limit.

Formula:
\[ C_{pk\text{-LOWER}} = \frac{\mu - LSL}{3\sigma} \]

Ppk-lower
One-sided long-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and lower specification limit over three times the forecast long-term standard deviation; often used to calculate process capability indices with only a lower specification limit.

Formula:
\[ P_{pk\text{-LOWER}} = \frac{\mu - LSL}{3\sigma} \]

Cpk-upper
One-sided short-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and upper specification limit over three times the forecast short-term standard deviation; often used to calculate process capability indices with only an upper specification limit.

Formula:
\[ C_{pk\text{-UPPER}} = \frac{USL - \mu}{3\sigma} \]

Ppk-upper
One-sided long-term capability index; for normally distributed forecasts, the ratio of the difference between the forecast mean and upper specification limit over three times the forecast long-term standard deviation; often used to calculate process capability indices with only an upper specification limit.

Formula:
Cpk

Short-term capability index (minimum of calculated Cpk-lower and Cpk-upper) that takes into account the centering of the forecast with respect to the midpoint of the specified limits; a Cpk equal to or greater than 1 indicates a quality level of 3 sigmas or better.

Formula:

\[ C_{pk} = \min(C_{pk-UPPER}, C_{pk-LOWER}) = C_p(1 - k) \]

where:

\[ k = \frac{\left(\frac{USL + LSL}{2}\right) - \mu}{(USL - LSL)/2} \]

Ppk

Long-term capability index (minimum of calculated Cpk-lower and Cpk-upper) that takes into account the centering of the forecast with respect to the midpoint of the specified limits; a Ppk equal to or greater than 1 indicates a quality level of 3 sigmas or better.

Formula:

\[ P_{pk} = \min(P_{pk-UPPER}, P_{pk-LOWER}) = P_p(1 - k) \]

where:

\[ k = \frac{\left(\frac{USL + LSL}{2}\right) - \mu}{(USL - LSL)/2} \]

Cpm

Short-term Taguchi capability index; similar to Cpk but considers a target value, which may not necessarily be centered between the upper and lower specification limits.

Formula:

\[ C_{pm} = \frac{USL - LSL}{6\sqrt{(\mu - T)^2 + \sigma^2}} \]

where \( T \) is Target value; default is:

\[ \frac{USL + LSL}{2} \]
**Ppm**

Long-term Taguchi capability index; similar to Ppk but considers a target value, which may not necessarily be centered between the upper and lower specification limits.

Formula:

\[
\text{Ppm} = \frac{\text{USL} - \text{LSL}}{6\sqrt{\left(\mu - T\right)^2 + \sigma^2}}
\]

where \( T \) is Target value; default is:

\[
\frac{\text{USL} + \text{LSL}}{2}
\]

**Z-LSL**

The number of standard deviations between the forecast mean and the lower specification limit.

**Note:** \( Z \) scores typically are reported only for normal data.

Formula:

\[
Z_{\text{LSL}} = \frac{\mu - \text{LSL}}{\sigma}
\]

**Z-USL**

The number of standard deviations between the forecast mean and the upper specification limit.

**Note:** \( Z \) scores typically are reported only for normal data.

Formula:

\[
Z_{\text{USL}} = \frac{\text{USL} - \mu}{\sigma}
\]

**Zst**

For short-term data, \( Z_{ST} = Z_{TOTAL} \), expressed as Zst-total,

where

\[
Z_{\text{TOTAL}} = \Phi^{-1}(p(N/C)_{\text{TOTAL}})
\]

and
\( \Phi^{-1}(x) \)

is the inverse normal cumulative distribution function, which assumes a right-sided tail.

In Microsoft Excel:

\[ \Phi^{-1}(x) = \text{NORMSINV}(x) \]

When displaying short-term metrics, \( Z_{ST} \) appears as Zst-total. This metric is equal to \( Z-\text{LSL} \) if there is only a lower specification limit, or \( Z-\text{USL} \) if there is only an upper specification limit.

For long-term data, \( Z_{ST} = Z_{LT} + Z\text{ScoreShift} \). When displaying long-term metrics, \( Z_{ST} \) appears in the capability metrics table as Zst.

**Note:** \( Z \) scores typically are reported only for normal data. The maximum value for \( Z \) scores calculated by from forecast data is 21.18.

**Zst-total**

For short-term metrics when both specification limits are defined, the number of standard deviations between the short-term forecast mean and the lower boundary of combining all defects onto the upper tail of the normal curve. Also equal to \( Z_{LT} \)-total plus the \( Z \)-score shift value if long-term metrics are calculated.

When short-term metrics are calculated, Zst-total is equivalent to \( Z_{ST} \), described in the previous section.

**Note:** \( Z \) scores typically are reported only for normal data.

**Zlt**

For long-term data, \( Z_{LT} = Z_{TOTAL} \), expressed as Zlt-total,

where

\[ Z_{TOTAL} = \Phi^{-1}(p(N/C)_{TOTAL}) \]

and

\[ \Phi^{-1}(x) \]

is the inverse normal cumulative distribution function, which assumes a right-sided tail.

In Microsoft Excel:

\[ \Phi^{-1}(x) = \text{NORMSINV}(x) \]

When displaying long-term metrics, \( Z_{LT} \) appears as Zlt-total. This metric is equal to \( Z-\text{LSL} \) if there is only a lower specification limit or \( Z-\text{USL} \) if there is only an upper specification limit.
For short-term data, \( Z_{LT} = Z_{ST} - Z\text{ScoreShift} \). When displaying short-term metrics, \( Z_{LT} \) appears in the capability metrics table as \( Z_{lt} \).

**Note:** \( Z \) scores typically are reported only for normal data. The maximum value for \( Z \) scores calculated by from forecast data is 21.18.

### \( Z_{lt\text{-total}} \)

For long-term metrics when both specification limits are defined, the number of standard deviations between the long-term forecast mean and the lower boundary of combining all defects onto the upper tail of the normal curve. Also equal to \( Zst\text{-total} \) minus the \( Z \)-score shift value if short-term metrics are calculated.

When long-term metrics are calculated, \( Z_{lt\text{-total}} \) is equivalent to \( Z_{LT} \), described in the previous section.

**Note:** \( Z \) scores typically are reported only for normal data.

### \( p(N/C)\text{-below} \)

Probability of a defect below the lower specification limit; \( DPU_{\text{BELOW}} \).

**Formula:**

\[
p(N/C)_{\text{BELOW}} = \Phi(Z_{LSL})
\]

where \( F \) is the area beneath the normal curve below the LSL, otherwise known as unity minus the normal cumulative distribution function for the LSL (assumes a right-sided tail).

**In Microsoft Excel:**

\[
\Phi(Z) = 1 - \text{NORMSDIST}(Z)
\]

**Note:** When calculated from the best-fitted non-normal distribution, \( p(N/C)\text{-below} = F(\text{LSL}) \), where \( F \) is the cumulative distribution function for the best-fitted non-normal distribution. When calculated from forecast values, \( p(N/C)\text{-below} \) is the percentile of the values in the forecast that fall below the LSL.

### \( p(N/C)\text{-above} \)

Probability of a defect above the upper specification limit; \( DPU_{\text{ABOVE}} \).

**Formula:**

\[
p(N/C)_{\text{ABOVE}} = \Phi(Z_{USL})
\]

where \( F \) is the area beneath the normal curve above the USL, otherwise known as unity minus the normal cumulative distribution function for the USL (assumes a right-sided tail).
In Microsoft Excel:

\[ \Phi(Z) = 1 - \text{NORMSDIST}(Z) \]

**Note:** When calculated from the best-fitted non-normal distribution, \( \Phi(N/C)\text{-above} = 1 - F(\text{USL}) \), where \( F \) is the cumulative distribution function for the best-fitted non-normal distribution. When calculated from forecast values, \( \Phi(N/C)\text{-above} \) is calculated using the percentile of the values in the forecast that fall above the USL.

**\( \Phi(N/C)\text{-total} \)**

Probability of a defect outside the lower and upper specification limits; \( DPU_{\text{TOTAL}} \).

**Formula:**

\[ \Phi(N/C)_{\text{TOTAL}} = \Phi(N/C)_{\text{ABOVE}} + \Phi(N/C)_{\text{BELOW}} \]

**PPM-below**

Defects below the lower specification limit, per million units.

**Formula:**

\[ \text{PPM}_{\text{BELOW}} = \Phi(N/C)_{\text{BELOW}} \cdot 10^6 \]

**PPM-above**

Defects above the upper specification limit, per million units.

**Formula:**

\[ \text{PPM}_{\text{ABOVE}} = \Phi(N/C)_{\text{ABOVE}} \cdot 10^6 \]

**PPM-total**

Defects outside both specification limits, per million units.

**Formula:**

\[ \text{PPM}_{\text{TOTAL}} = \text{PPM}_{\text{ABOVE}} + \text{PPM}_{\text{BELOW}} \]

**LSL**

Lower specification limit; the lowest acceptable value of a forecast involved in process capability, or quality, analysis. User-defined by direct entry or reference when defining a forecast.
**USL**

Upper specification limit; the highest acceptable value of a forecast involved in process capability, or quality, analysis. User-defined by direct entry or reference when defining a forecast.

**Target**

The ideal target value of a forecast involved in process capability analysis. User-defined by direct entry or reference when defining a forecast.

**Z-score Shift**

An optional shift value to use when calculating long-term capability metrics. The default, set in the Capability Options dialog box, is 1.5.
If none of the provided distributions fits the data, you can use the custom distribution to define one. For example, a custom distribution can be especially helpful if different ranges of values have specific probabilities. You can create a distribution of one shape for one range of values and a different distribution for another range. You can describe a series of single values, discrete ranges, or continuous ranges. This section uses real-world examples to describe the custom distribution.

Since it is easier to understand how the custom distribution works with a hands-on example, you may want to start and use it to follow the examples. To follow the custom distribution examples, first create a new Microsoft Excel workbook then select cells as specified.

The listed sections show how to use the custom distribution. Also see Appendix A in the Oracle Crystal Ball User’s Guide.

**Custom Distribution Example 1**

Before beginning example one, open the Custom Distribution dialog as follows:

1. Click cell D11.
2 Select Define, then Define Assumption.
   The Distribution Gallery dialog opens.

3 Click the All category to select it.

4 Scroll to find the custom distribution, then click it.

5 Click OK.
   Crystal Ball displays the Define Assumption dialog.

Figure 35 Define Assumption Dialog for Custom Distributions

Using the custom distribution, a company can describe the probable retail cost of a new product. The company decides the cost could be $5, $8, or $10. In this example, you will use the custom distribution to describe a series of single values.

To enter the parameters of this custom distribution:

1 Type 5 in the Value text box and click Enter.
   Since you do not specify a probability, defaults to a relative probability of 1.00 for the value 5. A single value bar displays the value 5.00.

   Relative probability means that the sum of the probabilities does not have to add up to 1. So the probability for a given value is meaningless by itself; it makes sense only in relation to the total relative probability. For example, if the total relative probability is 3 and the relative probability for a given value is 1, the value has a probability of 0.33.

2 Type 8 in the Value text box.

3 Click Enter.
   Since you did not specify a probability, defaults to a relative probability of 1.00 (displayed on the Probability scale along the first vertical axis) for the value 8. A second value bar represents the value 8.

4 Type 10 in the Value text box.
5 **Click Enter.**

Crystal Ball displays a relative probability of 1.00 for the value 10. A third single value bar represents the value 10.

*Figure 36 on page 85* shows the value bars for the values 5, 8, and 10, each with a relative probability of 1.00.

![Image of Custom Distribution](image)

Now, each value has a probability of 1. However, when you run the simulation, their total relative probability becomes 1.00 and the probability of each value is reset to .3333.

To reset their probabilities before you run the simulation, follow these steps:

1. **Click the bar with a value of 5.00.**
   
   Its value is displayed in the *Value* text box.

2. **Type the probability as the formula =1/3 in the Probability text box and click Enter.**
   
   You could also enter a decimal — for example, 0.333333 — but the formula is more exact.

3. **Follow steps 6 and 7 for the other two bars.**

   Crystal Ball rescales each value to a relative probability of 0.33 on the *Relative Probability* scale along the first vertical axis (*Figure 37 on page 86*).
Custom Distribution Example 2

In this example, you will use the custom distribution to describe a continuous range of values, since the unit cost can take on any value within the specified intervals.

- Before beginning example two, clear the values entered in example one as follows:
  1. Right-click in the chart and select Clear Distribution from the menu.
  2. Select Parameters, and then Continuous Ranges to enter value ranges.
  3. Enter the first range of values:
     - Type 5 in the Minimum text box.
     - Type 15 in the Maximum text box.
     - Type .75 in the Probability text box. This represents the total probability of all values within the range.
  4. Click Enter.

Crystal Ball displays a continuous value bar for the range 5.00 to 15.00, as in Figure 38 on page 87, and returns the cursor to the Minimum text box. Notice that the height of the range is 0.075. This represents the total probability divided by the width (number of units) in the range, 10.
5 Enter the second range of values:

- Type 16 in the Minimum text box.
- Type 21 in the Maximum text box.
- Type .25 in the Probability text box.
- Click Enter.

Crystal Ball displays a continuous value bar for the range 16.00 to 21.00. Its height is .050, equal to .25 divided by 5, the number of units in the range. Both ranges now are displayed in the Custom Distribution dialog (Figure 39 on page 87).

You can change the probability and slope of a continuous range, as described in the following steps:

1 Click anywhere on the value bar for the range 16 to 21.

The value bar changes to a lighter shade.
2 Select Parameters, and then Sloping Ranges.

Additional parameters are displayed in the Custom Distribution dialog (Figure 40 on page 88).

![Figure 40 Sloping Range Parameters, Custom Distribution Dialog](image)

3 Set Height of Min. and Height of Max. equal to what currently is displayed in the chart, 0.05.

This can be an approximate value. Height of Min. is the height of the range Minimum and Height of Max. is the height of the range Maximum.

4 Click Enter.

The range returns to its original color and its height is displayed unchanged.

5 Click in the range again to select it and set Height of Max. to 0.025. Then, click Enter.

The second side of the range drops to half the height of the first (Figure 41 on page 88). The range is selected to show its parameters after the change.

![Figure 41 Sloping Continuous Value Range](image)

6 Optional: Change the range from continuous to discrete values by adding a step value. Type .5 in the Step text box and click Enter.

The sloped range is now discrete. Separate bars are displayed at the beginning and end of the range and every half unit in between (16, 16.5, 17, 17.5 and so on until 21), as shown in Figure 42 on page 89. If the discrete range represented money, it could only include whole dollars and 50-cent increments.

You can enter any positive number in the Step text box. If you entered 1 in this example, the steps would fall on consecutive integers, such as whole dollars. Leave the Step parameter blank for continuous ranges.
Although the bars have spaces between them, their heights and the width of the range they cover are equal to the previous continuous sloped range and the total probability is the same.

While a second continuous range could have extended from 15 to 20, the second range in this example starts at 16 rather than 15 to illustrate a discrete range because, unlike continuous ranges, discrete ranges cannot touch other ranges.

With Crystal Ball, you can enter single values, discrete ranges, or continuous ranges individually. You also can enter any combination of these three types in the same Custom Distribution dialog as long as you follow these guidelines: ranges and single values cannot overlap one another; however, the ending value of one continuous range can be the starting value of another continuous range.

**Custom Distribution Example 3 — Loading Data**

This example describes a special feature in the Custom Distribution dialog: the Load Data button, which pulls numbers from a specified cell range (grouped data) on the worksheet.

In this example, the same company decides that the unit cost of the new product can vary widely. The company feels it has a 20% chance of being any number between $10 and $20, a 10% chance of being any number between $20 and $30, a 30% chance of being any number between $40 and $50, a 30% chance of being a whole dollar amount between $60 and $80, and there is a 5% chance the value will be either $90 or $100. All the values have been entered on the worksheet in this order: range minimum value, range maximum value (for all but Single Value ranges), total probability, and step (for the Discrete Range only) as shown in Figure 43 on page 90.
In this case, discrete ranges have the most parameters. So, you can create an assumption, select Custom Distribution, and then select Parameters, and then Discrete Ranges before loading the data.

If the data also included discrete sloping ranges, you could select Parameters, and then Sloping Ranges before loading the data. The data table would then have five columns and could accommodate all data types.

To complete the data load after the Parameters setting is selected:

1. Click the More button beside the Name text box.

   The Custom Distribution dialog expands to include a data table (Figure 44 on page 90).

A column is displayed for each parameter in the current set (selected using the Parameters menu). Parameters, then Discrete Ranges was selected before viewing the data table, so there is a column in the data table for each discrete range parameter. Because the single value and continuous ranges have subsets of the same group of parameters, their parameters will also fit into the table.

2. Since the values are already on the worksheet, you can click Load Data to enter them into the Custom Distribution dialog.
The **Load Data** dialog opens (Figure 45 on page 91).

The default settings are appropriate for most purposes, but the following other options are available:

- When loading unlinked data, you can choose to replace the current distribution with the new data or append new data to the existing distribution.

- If probabilities are entered cumulatively into the spreadsheet you are loading, select **Probabilities are cumulative**. Then, determines the probabilities for each range by subtracting the previous probability from the one entered for the current range. You can select **View**, and then **Cumulative Probability** to display the data cumulatively in the assumption chart.

3 Enter a location range for the data, in this case A2 : D7. If the range has a name, you can enter the name preceded by an = sign.

4 When all settings are correct, click **OK**.

Crystal Ball enters the values from the specified range into the custom distribution and plots the specified ranges (Figure 46 on page 92).
Entering Tables of Data into Custom Distributions

Subtopics

- Unweighted Values
- Weighted Values
- Mixed Single Values, Continuous Ranges, and Discrete Ranges
- Mixed Ranges, Including Sloping Ranges
- Connected Series of Ranges (Sloping)
- Connected Series of Continuous Uniform Ranges (Cumulative)
- Other Data Load Notes

Follow the rules in this section for loading data. If a data range has been named, you can enter the name (preceded by the = sign) in the range text box of the Load Data dialog.

The listed sections show how to enter tables of data into custom distributions.

Unweighted Values

Single values are values that do not define a range. Each value stands alone. For a series of single values with the same probabilities (unweighted values), use a one-column format or more than five columns (Figure 47 on page 92). The values go in each cell and the relative probabilities are all assumed to be 1.0. Select Parameters, and then Unweighted Values to enter these.

Figure 47  Single Values with the Same Probability
Figure 48 on page 93 shows the custom distribution created by loading the unweighted values illustrated in Figure 47.

**Figure 48  Unweighted Values Loaded in a Custom Distribution**

![Unweighted Values Loaded in a Custom Distribution](image)

**Weighted Values**

For a series of single values all with different probabilities, use a two-column format. The first column contains single values, the second column contains the probability of each value (Figure 49 on page 93).

**Figure 49  Single Values with Different Probabilities (Weighted Values)**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Weight or Probability</td>
</tr>
<tr>
<td>10</td>
<td>2.00</td>
</tr>
<tr>
<td>11</td>
<td>5.00</td>
</tr>
<tr>
<td>12</td>
<td>7.00</td>
</tr>
<tr>
<td>13</td>
<td>8.00</td>
</tr>
<tr>
<td>14</td>
<td>10.00</td>
</tr>
<tr>
<td>15</td>
<td>11.00</td>
</tr>
</tbody>
</table>

**Note:** Blank probabilities are interpreted as a relative probability of 1.0. Values with zero probability should be explicitly entered with a probability of 0.0.

Figure 50 on page 94 shows the custom distribution created by loading the values illustrated in Figure 49.
Mixed Single Values, Continuous Ranges, and Discrete Ranges

For any mixture of single values and continuous ranges, use a three-column format, obtained by selecting Parameters, and then Continuous Ranges. The three-column format is the same as using the first three columns shown in Figure 43 on page 90, Figure 44 on page 90, and Figure 46 on page 92.

If the mix includes uniform (non-sloping) discrete ranges, use a four-column format, as in the first four columns of Figure 51 on page 95 and Figure 52 on page 95. To obtain four columns, select Parameters, and then Discrete Ranges.

Mixed Ranges, Including Sloping Ranges

If sloping ranges are included in a mix of ranges, select Parameters, and then Sloping Ranges to display a five-column data table. The first column contains the range Minimum value, the second column contains the range Maximum value, the third column contains Height of Min. (the relative probability — height — at the Minimum value), the fourth column contains Height of Max. (the relative probability at the Maximum value), and the fifth column contains the Step value for discrete sloping ranges. For continuous sloping ranges the fifth column (Step) is left blank.

Notice that if uniform discrete ranges are included, their first three columns contain the Minimum, Maximum, and Probability as in a four-column format but the fourth column is left blank and Step is entered in the fifth column (Figure 51 on page 95).
Connected Series of Ranges (Sloping)

For a connected series of sloping continuous ranges, select **Parameters**, and then **Sloping Ranges** to use a five-column format. The first column contains the lowest **Minimum** value of the highest-value (right-most) range, the second column contains the **Maximum** value of each connected range, the third column contains the **Height of Min.** (relative probability of the **Minimum** value) if it differs from the previous **Height of Max.** (otherwise it is left empty), and the fourth column contains **Height of Max.** (relative probability of the **Maximum** value) for that range. The fifth column is left blank for continuous ranges but a fifth column is necessary to indicate that these are sloping ranges.
For example, row 20 in Figure 51 on page 95 shows a connected continuous sloping range. The Minimum cell is blank because the Minimum value is equal to 7, the previous Maximum. The Height of Min. is blank because it is equal to 6, the previous Height of Max.

**Connected Series of Continuous Uniform Ranges (Cumulative)**

For a connected series of continuous uniform ranges specified using cumulative probabilities, use a three-column format with the common endpoints of the ranges in the second column and the cumulative probabilities in the third column. The first column is left blank except for the minimum value of the first range, beside the maximum in the second column (Figure 53 on page 96). Be sure to select **Probabilities are cumulative** in the **Load Data** dialog.

![Figure 53 Connected Continuous Uniform Ranges](image)

Figure 54 on page 96 shows the custom distribution created by loading the data illustrated in Figure 53.

![Figure 54 Connected Continuous Uniform Ranges After Loading](image)

**Other Data Load Notes**

You can load each type of range separately or you can specify the range type with the greatest number of parameters and load all types together. Other rules are.

- Cumulative probabilities are supported for all but sloping ranges.
Blank probabilities are interpreted as a relative probability of 1.0. Values with zero probability should be explicitly entered with a probability of 0.0.

For continuous connected ranges, for either endpoint values or probabilities, if the starting cell is blank, the previous end value is used as the start for this range.

When you load a discrete value that exists in the table already, its probability is incremented by 1. For continuous ranges, this is not allowed; an error message about overlapping ranges opens.

Note: In versions of Crystal Ball earlier than 11.1.1.3.00, ranges or values with 0 probabilities were removed. Sloping ranges with Height of Min. and Height of Max. equal to 0 were also removed. The current rules are as follows:

In the simple view for weighted values, if users enter a probability of 0 for a value, the users are prompted about deleting the value. For other types of values, 0 works as before to delete the range or value.

In advanced (table) view for weighted values, if users enter a probability of 0 for a value, it is ignored (or added, which is the same thing). For other types of values, 0 works as before to delete the range or value.

If a custom distribution was created with unlinked data in a version of Crystal Ball earlier than version 11.1.1.3.00, the frequency distribution does not change when the model is loaded in Crystal Ball version 11.1.1.3.00 or later. However, if the input range was linked for a custom distribution, the frequency distribution is updated according to current rules. All new custom distributions, linked and unlinked, are evaluated according to current rules.

Comparing Distribution Types

Many of the distributions discussed in this chapter are related to one another in various ways. For example, the geometric distribution is related to the binomial distribution. The geometric distribution represents the number of trials until the next success while the binomial represents the number of successes in a fixed number of trials. Similarly, the Poisson distribution is related to the exponential distribution. The exponential distribution represents the amount of time until the next occurrence of an event while the Poisson distribution represents the number of times an event occurs within a given period of time.

In some situations, as when the number of trials for the binomial distribution becomes very large, the normal and binomial distributions become similar. For these two distributions, as the number of binomial trials approaches infinity, the probabilities become identical for any given interval. You also can use the Poisson distribution to approximate the binomial distribution when the number of trials is large, but there is little advantage to this since takes a comparable amount of time to compute both distributions.

Likewise, the normal and Student’s t distributions are related. With Degrees of Freedom, then 30, Student’s t closely approximates the normal distribution.
The binomial and hypergeometric distributions are also closely related. As the number of trials and the population size increase, the hypergeometric trials tend to become independent like the binomial trials: the outcome of a single trial has a negligible effect on the probabilities of successive observations. The differences between these two types of distributions become important only when you are analyzing samples from relatively small populations. As with the Poisson and binomial distributions, requires a similar amount of time to compute both the binomial and hypergeometric distributions.

The yes-no distribution is simply the binomial distribution with Trials = 1.

The Weibull distribution is flexible. Actually, it consists of a family of distributions that can assume the properties of several distributions. When the Weibull shape parameter is 1.0, the Weibull distribution is identical to the exponential distribution. The Weibull location parameter parameter you set up an exponential distribution to start at a location other than 0.0. When the shape parameter is less than 1.0, the Weibull distribution becomes a steeply declining curve. A manufacturer may find this effect useful in describing part failures during a burn-in period. When the shape parameter is equal to 2.0, a special form of the Weibull distribution, called the Rayleigh distribution, results. A researcher may find the Rayleigh distribution useful for analyzing noise problems in communication systems or for use in reliability studies. When the shape parameter is set to 3.25, the Weibull distribution approximates the shape of the normal distribution; however, for applications when the normal distribution is appropriate, use it instead of the Weibull distribution.

The gamma distribution is also a flexible family of distributions. When the shape parameter is 1.0, the gamma distribution is identical to the exponential distribution. When the shape parameter is an integer greater than one, a special form of the gamma distribution, called the Erlang distribution, results. The Erlang distribution is especially useful in the areas of inventory control and queueing theory, where events tend to follow Poisson processes. Finally, when the shape parameter is an integer plus one half (e.g., 1.5, 2.5, etc.), the result is a chi-square distribution, useful for modeling the effects between the observed and expected outcomes of a random sampling.

When no other distribution seems to fit the historical data or accurately describes an uncertain variable, you can use the custom distribution to simulate almost any distribution. The Load Data button on the Custom Distribution dialog lets you read a series of data points or ranges from value cells in the worksheet. If you like, you can use the mouse to individually alter the probabilities and shapes of the data points and ranges so that they more accurately reflect the uncertain variable.

**Sequential Sampling with Custom Distributions**

The probability distributions supplied with are useful in a variety of modeling situations. Organizations may still want to prepare their own libraries of distributions based on data specific to their applications and situations. One such system involves libraries of stochastic information packets (SIPs), an approach set forth in the article “Probability Management” (see the 2006 reference by S. Savage et al. in the Crystal Ball bibliography in the Oracle Crystal Ball Reference and Examples Guide).
A SIP is a list of time- or order-sensitive values for a particular variable. These values are sampled as sequential trials during a Monte Carlo simulation. SIPs are used to preserve the correlation structure between SIP variables without having to explicitly compute and define a matrix of correlation coefficients.

SIPs can be represented by custom distributions in and can then be published and shared by organizations using Crystal Ball’s Publish and Subscribe features in the Distribution Gallery.

**Creating Custom SIP Distributions**

The easiest way to load a SIP or similar set of values into is to organize the data in a single column in Microsoft Excel.

1. Then, define an assumption in using a custom distribution:

2. Select a cell and select Define, and then Define Assumption.

3. Looking at the list of All distributions in the Distribution Gallery, scroll to the Custom distribution and double-click it.

4. In the Define Assumption dialog, select Parameters, and then Sample Sequentially. This also switches the parameter set to Unweighted Values.

5. Click the More button, , to expand the dialog.

6. Click Load Data and enter the range of data in the Location of Data text box.

   You can type in the range or a range name, or click the cell selector icon to select the range. Be sure Sample Sequentially (instead of randomly) is still selected and click OK.

The distribution is displayed in the Define Assumption dialog (Figure 55).
Running Simulations with SIPs

To run a simulation with one or more SIPs, define assumptions in the model as usual, pulling in SIPs from a library (Category) in the Distribution Gallery as needed. When the simulation runs, samples the SIP values sequentially instead of randomly. If the simulation has more trials than the SIP has values, sampling wraps around to the first value of the distribution. If you are using several SIPs with different numbers of values, a warning is displayed. You can either continue, knowing that mismatched values will be sampled together after sampling wraps for one SIP but not the other, or you can cancel the simulation and correct the problem. You can also run fewer trials so that sampling doesn’t need to wrap for any SIP.

Correlations are not allowed for assumptions defined as SIPs, since the correlation structures are implicitly defined for them. Warnings are displayed if correlations are defined for them.

Two Developer Kit calls, CB.DefineAssumND and CB.GetAssum, support the new parameter for sequential sampling. The Developer Kit constant is called cbAsmIsSequential, a Boolean type.

Formulas for Probability Distributions

This section contains the formulas used to calculate probability distributions.

Beta Distribution

Parameters: Minimum value (Min), Maximum value (Max), Alpha (α), Beta (β)

Formula:
\[
f(x) = \begin{cases} 
  \frac{z^{(a-1)}(1-z)^{(b-1)}}{\beta(a, b)} \\
  \text{for } Min \leq x \leq Max
\end{cases}
\]

where:
\[
z = \frac{x - Min}{Max - Min}
\]

where:
\[
\beta(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a + b)}
\]

and where \( \Gamma \) is the Gamma function.

Method 1: Gamma Density Combination

Comment: The Beta variate is obtained from:

\[
Min + \left(\frac{u}{u+v}\right) \times (Max - Min)
\]

where \( u = \text{Gamma}(\alpha, 1) \) and \( v = \text{Gamma}(\beta, 1) \).
Method 2: Rational Fraction Approximation method with a Newton Polish step
Comment: This method is used instead of Method 1 when Latin Hypercube sampling is in effect.

**BetaPERT Distribution**

Parameters: Minimum value (Min), Most likely value (Likely), Maximum value (Max)

Formula:

\[ f(x) = \frac{(x - \text{Min})^{\alpha - 1} (\text{Max} - x)^{\beta - 1}}{\text{B}(\alpha, \beta)(\text{Max} - \text{Min})^{\alpha + \beta - 1}} \]

for \( \text{Min} \leq x \leq \text{Max} \)

where:

\[ \alpha = 6 \left( \frac{\mu - \text{Min}}{\text{Max} - \text{Min}} \right) \]

\[ \beta = 6 \left( \frac{\text{Max} - \mu}{\text{Max} - \text{Min}} \right) \]

\[ \mu = \frac{\text{Min} + 4 \times \text{Likely} + \text{Max}}{6} \]

and \( \text{B}(\alpha, \beta) \) is the beta integral.

**Binomial Distribution**

Parameters: Probability of success in each trial \( (p) \), Number of total trials \( (n) \)

Formula:

\[ P\{x = i\} = \binom{n}{i} p^i (1 - p)^{n - i} \]

for \( i = 0,1,2,...,n; \)

\( p > 0; \ 0 < n < 1 \times 9 \)

where:

\[ \binom{n}{i} = \frac{n!}{i!(n - i)!} \]

and \( x = \) number of successful trials

Method: Direct Simulation

Comment: Computation time increases linearly with number of trials.
Discrete Uniform Distribution

Parameters: Minimum value (Min), Maximum value (Max)

Formula:

\[ f(x) = \begin{cases} 
\frac{1}{(Max - Min + 1)} & \text{if } Min \leq x \leq Max, \ Min < Max \\
0 & \text{otherwise}
\end{cases} \]

Comment: This is the discrete equivalent of the uniform distribution, described in “Uniform Distribution” on page 109. In this distribution, a finite number of equally spaced values are equally likely to be observed.

Exponential Distribution

Parameters: Success rate (\(\lambda\))

Formula:

\[ f(x) = \begin{cases} 
\lambda e^{-\lambda x} & \text{if } x \geq 0 \text{ and } \lambda > 0 \\
0 & \text{if } x < 0
\end{cases} \]

Method: Inverse Transformation

Gamma Distribution

This distribution includes the Erlang and Chi-Square distributions as special cases.

Parameters: Location (\(L\)), Scale (\(s\)), Shape (\(\beta\))

Formula:

\[ f(x) = \begin{cases} 
\frac{(x-L)^{\beta-1} e^{\frac{x-L}{s}}}{\Gamma(\beta)s} & \text{if } x > L, \ 0 < \beta < \infty, \ 0 < s < \infty \\
0 & \text{if } x \leq L
\end{cases} \]

where \(\Gamma\) is the gamma function.

Note: Some textbook Gamma formulas use:

\[ s = \frac{1}{\lambda} \]

Method 1:

When \(\beta\) is less than 1, Vaduva’s rejection from a Weibull density.
When $\hat{\beta}$ is greater than 1, Best’s rejection from a $t$ density with 2 degrees of freedom.

When $\hat{\beta} = 1$, inverse transformation.

Method 2: Rational Fraction Approximation method with a Newton Polish step
Comment: This method is used instead of Method 1 when Latin Hypercube sampling is in effect.

**Geometric Distribution**

Parameters: Probability of success in each trial ($p$)

Formula:
\[
P(x = i) = p(1 - p)^i
\]

for:

\[
0 < p < 1 \text{ and } i = 0, 1, 2, ... \infty
\]

where $x =$ number of failures before the first success

Method: Inverse Transformation

**Hypergeometric Distribution**

Parameters:

Number of successful items in the population ($N_x$), sampled trials ($n$), population size ($N$)

Formula:
\[
P(x = i) = \binom{N_x}{i} \binom{N - N_x}{n - i} \binom{N}{n}
\]

where:
\[
\binom{n}{i} = \frac{n!}{i!(n-i)!}
\]

for:

\[
i = \text{Max}(n - (N - N_x), 0) ... \text{Min}(n, N_x)
\]

and $N \leq 1e5$

and $x =$ number of successful trials,

so $N_x =$ number of successful items in the population.

Method: Direct Simulation

Comment: Computation time increases linearly with population size.
Logistic Distribution

Parameters:
Mean ($\mu$), Scale ($s$)

Formula:
$$f(x) = \frac{2}{s(1 + z)^2}$$
for:
$$-\infty < x < \infty,$$
$$-\infty < s < \infty,$$
$$-\infty < \mu < \infty$$

where:
$$z = \frac{\left(\frac{x - \mu}{s}\right)}{e}$$

Method: Inverse Transformation

Lognormal Distribution

Parameters: Location (L), Mean ($\mu_{ar}$), Standard Deviation ($\sigma_{ar}$)

Mean =
$$\mu_{ar} = L + \theta^{\mu_{\log} + \sigma_{\log}^2}$$

Median =
$$L + \theta^{\mu_{log}}$$

Mode =
$$L + \theta^{\mu_{log} - \sigma_{log}^2}$$

Translation from arithmetic to log parameters:
L = L; L is always in arithmetic space.

Log mean =
$$\mu_{\log} = \ln\left(\frac{\mu_{ar} - L}{\left(\sigma_{\log}^2\right)}\right)$$

Log standard deviation =
\[
\log \sigma = \left( \ln \left( \theta \cdot \left( \frac{2 \cdot \ln - \mu}{\sigma^2} - L \right) + 1 \right) \right) ^ {\frac{1}{\theta}}
\]

where \( \ln \) = natural logarithm.

Formula:

\[
f(x, \theta, \mu, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi (x - L)}} \cdot e^{-\frac{1}{2} \left( \frac{\ln (x - L) - \mu_{\log}^2}{\sigma_{\log}^2} \right)}
\]

for:

\( L < x < \infty, \sigma_{\log}^2 > 0 \)

Method: Inverse transformation

Translation from log to geometric parameters: \( L = \mu \)

Geometric mean =

\( e^{\mu_{\log}} \)

Geometric std. dev. =

\( e^{\sigma_{\log}} \)

Translation from log to arithmetic parameters:

\( L = \mu \)

Arithmetic mean =

\( \mu_{ar} = L + e^{\frac{\sigma_{\log}^2}{2}} + \mu_{\log} \)

Arithmetic variance =

\( \sigma_{ar}^2 = e^{2\mu_{\log}} + e^{\sigma_{\log}^2} \left( e^{\sigma_{\log}^2} - 1 \right) \)

**Maximum Extreme Distribution**

The maximum extreme distribution is the positively skewed form of the extreme value distribution.

Parameters: Likeliest \((m)\), Scale \((s)\)

Formula:

\[
f(x) = \frac{1}{s} \cdot z \cdot e^{-z}
\]

for:
Minimum Extreme Distribution

The minimum extreme distribution is the negatively skewed form of the extreme value distribution.

Parameters: Likeliest \( (m) \), Scale \( (s) \)

Formula:
\[
I(x) = \frac{1}{s} \cdot z \cdot e^{z^2}
\]

for:
\[
-\infty < x < \infty,
-\infty < m < \infty \text{ and } \forall s > 0
\]

where:
\[
z = \frac{(x - m)}{s}
\]

Method: Inverse Transformation

Negative Binomial Distribution

Parameters: Probability of success in each trial \( (p) \), Shape \( (\beta) \)

Formula:
\[
P(x = i) = \begin{cases} 
\binom{i - 1}{\beta - 1} p^\beta (1 - p)^{i - \beta} & \text{for } 0 < p < 1 \\
0 & \text{for } 0 < \beta \leq 1 \\
\beta = \beta + 1, \beta + 2, \ldots & \infty
\end{cases}
\]

where:
\[
\binom{i - 1}{\beta - 1} = \frac{(i - 1)!}{(\beta - 1)! (i - \beta)!}
\]

and \( x = \) total number of trials required to achieve Shape number of successes.
Method: Direct Simulation through summation of Geometric variates
Comment: Computation time increases linearly with Shape.

**Normal Distribution**

This distribution is also known as the Gaussian distribution.

Parameters:
Mean (\(\mu\)), Standard Deviation (\(\sigma\))

Formula:
\[
f(x) = \frac{1}{\sqrt{2\pi \sigma}} e^{-(x-\mu)^2 / 2\sigma^2}
\]
for:
\(-\infty < x < \infty\)
\(-\infty < \mu < \infty\)
\(\sigma > 0\)

Method 1: Polar Marsaglia
Comment: This method is somewhat slower than other methods, but its accuracy is essentially perfect.

Method 2: Rational Fraction Approximation
Comment: This method is used instead of the Polar Marsaglia method when Latin HYPERcube sampling is in effect.
This method has a 7–8 digit accuracy over the central range of the distribution and a 5–6 digit accuracy in the tails.

**Pareto Distribution**

Parameters:
Location (\(L\)), Shape (\(\beta\))

Formula:
\[
f(x) = \begin{cases} 
\frac{\beta \cdot L^\beta}{x^{(\beta + 1)}} & \text{if } x \geq L \\
0 & \text{if } x < L 
\end{cases}
\]
for \(\beta > 0\)

Method: Inverse Transformation
Poisson Distribution

Parameters: Rate ($\lambda$)

Formula:

$$P\{X = i\} = e^{-\lambda} \frac{\lambda^i}{i!}$$

for:

$$i = 0, 1, 2, ..., \infty \text{ and } 0 < \lambda \leq 1e9$$

Method: Direct Simulation through Summation of Exponential Variates

Comment: Computation time increases linearly with Rate.

Student's $t$-Distribution

Parameters: Midpoint ($m$), Scale ($s$), Degrees of Freedom ($d$)

Formula:

$$f(z, d) = \frac{\Gamma\left(\frac{d+1}{2}\right)}{\sqrt{d\pi} \Gamma\left(\frac{d}{2}\right) \left(1 + \frac{z^2}{d}\right)^{\frac{d+1}{2}}}$$

where:

$-\infty < x < \infty$, integer

$0 < d \leq 30$

$s > 0$

and where:

$$z = \frac{x - m}{s}$$

and where:

$\Gamma$ = the gamma function

Triangular Distribution

Parameters: Minimum value ($Min$), Most likely value ($Likeliest$), Maximum value ($Max$)

Formula:
where:

\[ h = \frac{2}{\text{Max} - \text{Min}} \]

**Method:** Inverse Transformation

**Uniform Distribution**

Parameters: Minimum value (Min), Maximum value (Max)

Formula:

\[
f(x) = \begin{cases} 
\frac{1}{\text{Max} - \text{Min}} & \text{if } \text{Min} \leq x \leq \text{Max}, \text{Min} < \text{Max} \\
0 & \text{otherwise}
\end{cases}
\]

Method: Multiplicative Congruential Generator

This routine uses the iteration formula:

\[ r \leftarrow (62089911 \cdot r) \mod (2^{31} - 1) \]

Comment:

The generator has a period of length \(2^{31} - 2\), or 2,147,483,646. This means that the cycle of random numbers repeats after several billion trials. This formula is discussed in detail in the Simulation Modeling & Analysis and Art of Computer Programming, Vol. II, references in the Oracle Crystal Ball User’s Guide bibliography.

**Weibull Distribution**

A Weibull distribution with Shape = 2 is also known as the Rayleigh distribution.

Parameters:

Location (L), Scale (s), Shape (\(\beta\))

Formula:

\[
f(x) = \begin{cases} 
\left(\frac{\beta}{s}\right) \left(\frac{x-L}{s}\right)^{\beta-1} e^{-\left(\frac{x-L}{s}\right)^\beta} & \text{if } x \geq L, s > 0, \beta > 0 \\
0 & \text{if } x < L
\end{cases}
\]

where:
\[ \Gamma \] is the Gamma function.

Method: Inverse Transformation

**Yes-No Distribution**

This distribution is equivalent to the binomial distribution with Trials = 1. For details, see “Binomial Distribution” on page 101.

**Custom Distribution**

**Formula:**

The formula consists of a lookup table of single data points, continuous ranges, and discrete ranges. Each item in the table has a distinct probability relative to the other items. In addition, ranges might be positively or negatively sloped, giving values on one side or the other a higher probability of occurring.

**Method:** Sequential search of relative probabilities table.

**Comments:**

A Uniform variate is generated in the range (0, total relative probability). A sequential search of the relative probabilities table is then performed. The Inverse Transformation method is used whenever the uniform variate falls within a continuous or discrete range that is sloped in one direction or the other.

**Additional Comments**

All of the nonuniform generators use the same uniform generator as the basis for their algorithms.

The Inverse Transformation method is based on the property that the cumulative distribution function for any probability distribution increases monotonically from zero to one. Thus, the inverse of this function can be computed using a random uniform variate in the range (0, 1) as input. The resulting values then have the desired distribution.

The Direct Simulation method actually performs a series of experiments on the selected distribution. For example, if a binomial variate is being generated with Prob = .5 and Trials = 20, then 20 uniform variates in the range (0, 1) are generated and compared with Prob. The number of uniform variates found to be less than Prob then becomes the value of the binomial variate.

**Distribution Fitting Methods**

During distribution fitting, Crystal Ball computes Maximum Likelihood Estimators (MLEs) to fit most of the probability distributions to a data set. In effect, this method chooses values for the parameters of the distributions that maximize the probability of producing the actual data.
set. Sometimes, however, the MLEs do not exist for some distributions (for example, gamma, beta). In these cases, Crystal Ball resorts to other natural parameter estimation techniques.

When the MLEs do exist, they exhibit desirable properties:

- They are minimum-variance estimators of the parameters.
- As the data set grows, the biases in the MLEs tend to zero.

For several of the distributions (for example, uniform, exponential), it is possible to remove the biases after computing the MLEs to yield minimum-variance unbiased estimators (MVUEs) of the distribution parameters. These MVUEs are the best possible estimators.

### Distribution Parameter Defaults

**Subtopics**

- Beta
- BetaPERT
- Binomial
- Custom
- Discrete Uniform
- Exponential
- Gamma
- Geometric
- Hypergeometric
- Logistic
- Lognormal
- Maximum Extreme Value
- Minimum Extreme Value
- Negative Binomial
- Normal
- Pareto
- Poisson
- Student's t
- Triangular
- Uniform
- Weibull
- Yes-No

This section lists the initial values provides for the primary parameters in the Define Assumption dialog.

If an alternate parameter set is selected as the default mode, the primary parameters are still calculated as described below before conversion to the alternate parameters.

**Note:** Extreme values on the order of 1e±9 or ±1e16 may yield results somewhat different from those listed here.
Beta

If the cell value is 0:
Minimum is -10.00
Maximum is 10.00
Alpha is 2
Beta is 3

Otherwise:
Minimum is cell value - (absolute cell value divided by 10)
Maximum is cell value + (absolute cell value divided by 10)
Alpha is 2
Beta is 3

For out-of-range values, such as ±1e300:
Minimum is 0
Maximum is 1
Alpha is 2
Beta is 3

BetaPERT

If the cell value is 0:
Likeliest is 0
Minimum is -10.00
Maximum is 10.00

Otherwise:
Likeliest is cell value
Minimum is cell value - (absolute cell value divided by 10)
Maximum is cell value + (absolute cell value divided by 10)

Binomial

If the cell value is between 0 and 1:
Probability is the cell value
Trials is 50

If the cell value is between 1 and 1e9 (the maximum number of binomial trials):
Probability (Prob) is 0.5
Trials is cell value

Otherwise:
Probability (Prob) is 0.5
Trials is 50

**Custom**
Initially empty.

**Discrete Uniform**
If the cell value is 0 or -1e9:
Minimum is 0
Maximum is 10

Otherwise:
Minimum is cell value - INT (absolute cell value divided by 10)
Maximum is cell value + INT (absolute cell value divided by 10)

**Exponential**
If the cell value is 0, rate is 1.0.
Otherwise, rate is 1 divided by the absolute cell value.

**Gamma**
If the cell value is 0:
Location is 0.00
Scale is 1.00
Shape is 2

Otherwise:
Location is cell value
Scale is absolute cell value divided by 10
Shape is 2
**Geometric**

If the cell value is greater than 0 and less than 1, probability is cell value.
Otherwise, probability is 0.2.

**Hypergeometric**

If the cell value is greater than 0 and less than 1:
- Success is 100 times cell value
- Trials is 50
- Population size is 100

If the cell value is between 2 and the maximum number of Hypergeometric trials (1e5):
- Success is cell value divided by 2 (rounded downward)
- Trials is cell value divided by 2 (rounded downward)
- Population size is cell value

Otherwise:
- Success is 50
- Trials is 50
- Population size is 100

**Logistic**

If the cell value is 0:
- Mean is 0
- Scale is 1.0.

Otherwise:
- Mean is cell value
- Scale is absolute cell value divided by 10

**Lognormal**

If the cell value is greater than 0:
- Mean is cell value
- Standard deviation is absolute cell value divided by 10

Otherwise:
- Mean is $e$
Standard deviation is 1.0

**Maximum Extreme Value**
If the cell value is 0:
Likeliest is 0
Scale is 1
Otherwise:
Likeliest is cell value
Scale is absolute cell value divided by 10

**Minimum Extreme Value**
If the cell value is 0:
Likeliest is 0
Scale is 1
Otherwise:
Likeliest is cell value
Scale is absolute cell value divided by 10

**Negative Binomial**
If the cell value is less than or equal to 0:
Probability is 0.2
Shape is 10
If the cell value is greater than 0 and less than 1:
Probability is cell value
Shape is 10
Otherwise, unless the cell value is greater than 100:
Probability is 0.2
Shape is cell value
If the cell value is greater than 100, the shape is 10.

**Normal**
If the cell value is 0:
Mean is 0
Standard deviation is 1.00

Otherwise, unless the cell value is more than 100:
Mean is cell value
Standard deviation is absolute cell value divided by 10.0

**Pareto**

If the cell value is between 1.0 and 1,000:
Location is cell value
Shape is 2
Otherwise:
Location is 1.00
Shape is 2

**Poisson**

If the cell value is less than or equal to 0, the rate is 10.00.
If the cell value is greater than 0 and less than or equal to the maximum rate (1e9), the rate is the cell value.
Otherwise, the rate is 10.00

**Student's t**

If the cell value is 0:
Midpoint is 0
Scale is 1.00
Degrees is 5
Otherwise:
Midpoint is cell value
Scale is absolute cell value divided by 10
Degrees is 5

**Triangular**

If the cell value is 0:
Likeliest is 0
Minimum is -10.00
Maximum is 10.00

Otherwise:
Likeliest is cell value
Minimum is cell value minus absolute cell value divided by 10
Maximum is cell value plus absolute cell value divided by 10

**Uniform**
If the cell value is 0:
Minimum is -10.00
Maximum is 10.00

Otherwise:
Minimum is cell value minus absolute cell value divided by 10.0
Maximum is cell value plus absolute cell value divided by 10.0

**Weibull**
If the cell value is 0:
Location is 0
Scale is 1.00
Shape is 2

Otherwise:
Location is cell value
Scale is absolute cell value divided by 10
Shape is 2

**Yes-No**
If the cell value is greater than 0 and less than 1, the probability of Yes(1) equals the cell value.
Otherwise, the probability of Yes(1) equals 0.5.
Predictor Examples

Subtopics

- About These Examples
- Inventory Control
- Company Finances
- Human Resources

About These Examples

Monica’s Bakery, a hypothetical company used in the Predictor examples, is a rapidly growing bakery in Albuquerque, New Mexico. Since the opening, Monica has kept careful records (in a Microsoft Excel workbook) of the sales of her three main products: French bread, Italian bread, and pizza. With these records, she can better predict her sales, control her inventory, market her products, and make strategic, long-term decisions.

To open Monica’s workbook, select Help, then Crystal Ball, then Examples Guide, and then select Monica's Bakery. (In Microsoft Excel 2007 or later, select Resources in the Help group and then select Examples Guide).

The workbook has worksheets for sales data, operations, and cash flow. The Sales Data worksheet contains all the historical sales data that is available for forecasting. The Operations worksheet calculates the amount of different ingredients required to make different quantities of three breads. The Cash Flow worksheet calculates how much money the bakery has to spend on various capital projects. The Labor Costs worksheet estimates the increase in hourly wages to decide whether to invest in labor-saving equipment.

The following examples track Monica’s decision-making processes as she uses Predictor to work through both short-term and long-term decisions:
For more detailed product tutorials, see the Oracle Crystal Ball Predictor User’s Guide.

**Inventory Control**

The initial reason that Monica needs to forecast is to maintain enough ingredients to keep up with production. Monica’s distributors give her discounts for buying in bulk. However, she must balance this savings with maintaining product quality, which requires using the freshest ingredients possible. Monica wants improved forecasting to help her place orders that give her the best volume pricing while maintaining the quality of her products.

To follow along with this example, open Bakery.xls as described in “About These Examples” on page 119.

The Sales Data worksheet (Figure 56) shows the daily sales data of each of these products from the opening until the end of June 2010.

A summary of data for the three main products by week is displayed at the bottom of the Operations worksheet (Figure 57). Monica can change the table to summarize her results by product, by time period, and more.
Monica wants to order monthly, one month in advance. The bakery has already received this month’s delivery, which she placed last month. This month, she must place the order that will be delivered at the end of this month for the next month, so she must forecast sales for the next two months. Because she is in week 173 of her business, the forecast is for weeks 174 to 181.

To forecast the sales for weeks 174 to 181:

1. In the Bakery.xls workbook, click the Operations tab.
   The Operations worksheet is displayed.

2. Select one cell—for example, C41—in the Historical Demand By Week table at the bottom of the worksheet.

   Predictor automatically selects all the data in that table.

4. Ensure that:
   - The cell range $B$40:$E$213 is selected correctly on the Input Data panel, with headers, dates, and data in columns settings also selected
   - The Data Attributes panel shows time periods are in weeks with Seasonality set to AutoDetect
   - In the Methods panel, Multiple Linear Regression is cleared and all time-series methods are selected
   - Options settings are the defaults, RMSE and Standard forecasting

5. Click Run.

6. In the Predictor Results window, set Periods to forecast to 8, and then click Paste.
In the **Paste Forecasts to Spreadsheet** dialog, use the following settings and click **OK**:

- Select **At end of historical data** to indicate where to paste results.
- Select **Include date series** to list dates in the first column.
- Select **Paste forecasts as Crystal Ball assumptions**.
- Confirm that **AutoFormat** is selected.

The results paste to the end of the Historical Demand table as shown in **Figure 58**.

![Figure 58 Forecasted Bakery Operations Results](image)

The last four weeks of forecast values for each data series are automatically summed and placed into the table at the top of the spreadsheet, in the Sales Forecast column (cells C9:C11). In this table, the monthly sales forecast is converted to the number of items sold and then into the weight of each product.

The second table (below this top table) takes the total weight of each product (in cells C15, C19, C23) and calculates how much of each ingredient is required to produce that much product. The ingredients for each are then summed in the third table (below the second table) into the total amounts to order for the month (cells D31 to D34).

Based on the forecast, Monica should order:

- 12,080 pounds of flour
- 55 pounds of yeast
- 42 pounds of salt
- 125 pounds of tomatoes

**Company Finances**

Monica is always concerned about the bakery’s month-to-month cash flow (on a percent of sales basis). Predictor can help her manage her inventory, and she can use it to predict her revenue.
and understand her cash flow situation better. Understanding the bakery’s cash flow can, in turn, help her better manage major capital expenditures.

Monica is considering two major capital expenditures: a flour silo and a delivery van. She wants to start construction on the silo in July and purchase the delivery van in August. She needs to forecast when the bakery can safely pay for these projects or whether the bakery must finance them.

To follow along with this example, open Bakery.xls as described in “About These Examples” on page 119. If Bakery.xls is already open from the previous example, select Run, Reset to clear any existing results.

**Note:** Any pasted data will remain until you clear it in Microsoft Excel.

The bakery cash flow information is laid out in the Cash Flow worksheet, shown in Figure 59.

**Figure 59  Bakery Cash Flow Worksheet**

This worksheet has a table at the bottom that summarizes the sales data for the bakery’s three main products by month. You can forecast the next three months of revenue to decide when to attempt the capital expenditures.
To forecast the next three months of revenue:

1. In the Bakery.xls workbook, click the Cash Flow tab.

   The Cash Flow worksheet is displayed.

2. Select one cell—C36, for example—in the Historical Revenue By Month table at the bottom of the worksheet.


   Predictor automatically selects all the data in the Historical Revenue table.

4. Confirm the following settings:

   - In the Input Data panel, the cell range $C$36:$AP$36 is selected correctly, with Data in rows selected and no date or header settings selected.
   
   - In Data Attributes, the first setting is Data is in periods with Seasonality set to AutoDetect.
   
   - In Methods, all time-series methods are selected and Multiple Linear Regression is cleared (if available).
   
   - Options settings are the defaults: RMSE and Standard forecasting.

5. Click Run.

6. In the Predictor Results window, enter 3 for Periods to forecast.

7. Click Paste, and then, in the Paste Forecasts to Spreadsheet dialog, confirm that results are set to paste at the end of historical data as Crystal Ball assumptions and click OK.

   The results paste into the table at the bottom of the worksheet, cells AQ36 to AS36, and also are displayed at the top of the worksheet (cells E4 to G4) as shown in Figure 60.
The revenue forecasts for the next three months are used to calculate the percentage expenses in the second table.

The second table calculates the total expenses, and the third table calculates the necessary expenditure for each extraordinary item. Below these tables is the cash flow summary for the next three months, based on the forecasts. The net cash at the end of each month is what Monica is looking for (row 27). Based on forecasted sales, the new net cash values are $19,536.34 for July, $11,833.04 for August, and $35,452.31 for September. Based on the forecast, the bakery should wait until September to buy the van.

**Human Resources**

Monica's Bakery is a labor-intensive operation that pays a competitive wage. However, to maintain her target profitability, Monica must control labor costs. She knows many things are done around the bakery that could be done by expensive machinery, such as kneading, mixing, and forming. By accurately predicting her labor costs, she can decide when to invest in some of this equipment to keep her total expenses within budget.
From her interest in economics, Monica knows that a few key macro-economic figures drive labor costs, such as the Industrial Production Index, local CPI, and local unemployment. All of these figures are available on the Internet on a monthly basis from the Bureau of Labor Statistics and the Department of Commerce.

To follow along with this example, open Bakery.xls as described in “About These Examples” on page 119. If Bakery.xls is already open from the previous example, select Run, and then Reset to clear existing results.

Monica has created her Labor Costs worksheet with an interactive table at the bottom that lists the bakery’s average hourly wage for each month and the monthly numbers for the three economic indicators.

![Bakery Labor Costs Worksheet](image)

The average hourly wage depends on or is affected by the other three variables. Because of the dependency, Monica decides to use regression instead of time-series forecasting. For regression, the dependent variable is Monica’s Average Wage, and the other three are the independent variables.

➢ To forecast the hourly wage using regression:

1. In the Bakery.xls workbook, click the Labor Costs tab.

   The Labor Costs worksheet opens.

2. Select one cell—for example, C14—in the Economic Variables for Regression Analysis table at the top of the worksheet.


4. Ensure that:
   - In Input Data, the cell range $B$13:$F$50 is selected correctly, with data in columns, header, and date settings also selected.
In Data Attributes, the time periods are in months with a Seasonality of AutoDetect.

In Methods, Non-seasonal Methods, ARIMA, and Multiple Linear Regression are selected.

Regression variables are defined as follows: Monica’s Average Wage is a dependent variable, all the others are independent variables.

In Options, RMSE and Standard forecasting are selected.

5 Click Run.

6 In the Predictor Results window, set Periods to forecast to 6 and click Paste.

7 In Paste Forecasts to Spreadsheet, set the controls to paste forecasts as assumptions at the end of historical data with Include date series and AutoFormat selected.

8 Click OK.

The results paste at the bottom of the table (cells B51 to F56) as shown in Figure 62. Notice the results for independent variables are defined as assumptions.

Figure 62  Forecasted Labor Costs for Monica’s Bakery

Predictor first generates a regression equation to define the relationship between the dependent and independent variables. Second, it uses the time-series forecasting methods to forecast the independent variables individually. Third, Predictor uses those forecasted values to calculate the dependent variable forecast values using the regression equation. See Figure 63.

The forecast cells of the independent variables are simple value cells. The forecast cells of the dependent variable are formula cells containing the regression equation and using the forecast values from the independent variables.

The average wage in December is used to calculate the total increase in her payroll. Labor costs actually decrease –2%. With these results, Monica decides that labor costs over the next six months do not justify a major equipment capital purchase.
Exit Bakery.xls without saving the changes.

If you save the changes, you will overwrite the example spreadsheet.

Important Predictor Concepts

Subtopics

- About Forecasting Concepts
- Classic Time-series Forecasting
- Time-series Forecasting Accuracy Measures
- Time-series Forecasting Techniques
- Multiple Linear Regression
- Regression Methods
- Regression Statistics
- Historical Data Statistics
- Events Adjustment
- Data Screening and Adjustment Methods

About Forecasting Concepts

This appendix describes forecasting terminology. It defines the time-series forecasting methods that Predictor uses, as well as other forecasting-related terminology. This appendix also describes the statistics the program generates and the techniques that Predictor uses to do the calculations and select the best-fitting method.

Forecasting refers to the act of predicting the future, usually for planning and managing resources. Many scientific approaches to forecasting exist. You can perform “what-if” forecasting by creating a model and simulating outcomes, as with Crystal Ball, or by collecting data over time and analyzing the trends and patterns. Predictor uses the latter concept, analyzing the patterns of a time series to forecast future data.

The scientific approaches to forecasting usually fall into one of several categories:
- **Time-series** — Performs time-series analysis on past patterns of data to forecast results. This works best for stable situations in which conditions are expected to remain the same.

- **Regression** — Forecasts results using past relationships between a variable of interest and several other variables that may influence it. This works best for situations in which you need to identify the different effects of different variables. This category includes multiple linear regression.

- **Simulation** — Randomly generates many scenarios for a model to forecast the possible outcomes. This method works best where you may not have historical data, but you can build the model of the situation to analyze its behavior.

- **Qualitative** — Uses subjective judgment and expert opinion to forecast results. These methods work best for situations for which no historical data or models are available.

Predictor uses time-series and multiple linear regression for forecasting. Uses simulation. Each technique and method has advantages and disadvantages for particular types of data, so often you may forecast the data using several methods and then select the method that yields the best results.

The following section, “Classic Time-series Forecasting” on page 129, describes forecast methods available in Predictor and their uses.

These sections offer information about other data and forecast methodologies:

- “Multiple Linear Regression” on page 139
- “Historical Data Statistics” on page 143

### Classic Time-series Forecasting

#### Subtopics

- Classic Non-seasonal Forecasting Methods
- Classic Seasonal Forecasting Methods

This section provides more information about classic time-series forecasting methods used in Predictor. Information about ARIMA (also known as Box-Jenkins) time-series forecasting methods are found in “ARIMA Time-series Forecasting Formulas” on page 158.

Time-series forecasting assumes that historical data is a combination of a pattern and some random error. Its goal is to isolate the pattern from the error by understanding the pattern’s level, trend, and seasonality. You can then measure the error using a statistical measurement to describe both how well a pattern reproduces historical data and to estimate how accurately it projects the data into the future. See “Time-series Forecasting Accuracy Measures” on page 135.

By default, Predictor tries all of the classic time-series methods in the Methods Gallery. It then ranks them according to which method has the lowest error, depending on the error measure selected in the Options pane. The method with the lowest error is the best method.

Two primary techniques of classic time-series forecasting are used in Predictor:
“Classic Non-seasonal Forecasting Method Parameters” on page 132 — Estimate a trend by removing extreme data and reducing data randomness

“Classic Seasonal Forecasting Methods” on page 132 — Combine forecasting data with an adjustment for seasonal behavior

For information about regression forecasting methods, see “Multiple Linear Regression” on page 139. See for more information about the formulas Predictor uses for the non-seasonal and seasonal forecasting methods described in the following sections.

**Classic Non-seasonal Forecasting Methods**

Non-seasonal methods attempt to forecast by removing extreme changes in past data where repeating cycles of data values are not present. The following classic non-seasonal forecasting methods are available:

- “Single Moving Average (SMA)” on page 130
- “Double Moving Average (DMA)” on page 131
- “Single Exponential Smoothing (SES)” on page 131
- “Double Exponential Smoothing (DES)” on page 131

For information about associated parameters, see “Classic Non-seasonal Forecasting Method Parameters” on page 132.

**Single Moving Average (SMA)**

Smooths historical data by averaging the last several periods and projecting the last average value forward. Predictor can automatically calculate the optimal number of periods to average, or you can select the number of periods to average.

This method is best for volatile data with no trend or seasonality. It results in a straight, flat-line forecast.

Figure 64 Typical Single Moving Average Data, Fit, and Forecast Line
**Double Moving Average (DMA)**

Applies the moving average technique twice, once to the original data and then to the resulting single moving average data. This method then uses both sets of smoothed data to project forward. Predictor can automatically calculate the optimal number of periods to average, or you can select the number of periods to average.

This method is best for historical data with a trend but no seasonality. It results in a straight, sloped-line forecast.

**Figure 65 Typical Double Moving Average Data, Fit, and Forecast Line**

---

**Single Exponential Smoothing (SES)**

Weights all of the past data with exponentially decreasing weights going into the past. In other words, usually the more recent data has greater weight. Weighting in this way largely overcomes the limitations of moving averages or percentage change methods. Predictor can automatically calculate the optimal smoothing constant, or you can manually define the smoothing constant.

This method, which results in a straight, flat-line forecast is best for volatile data with no trend or seasonality.

**Figure 66 Typical Single Exponential Smoothing Data, Fit, and Forecast Line**

---

**Double Exponential Smoothing (DES)**

Applies SES twice, once to the original data and then to the resulting SES data. Predictor uses Holt’s method for double exponential smoothing, which can use a different parameter for the second application of the SES equation. Predictor can automatically calculate the optimal smoothing constants, or you can manually define the smoothing constants.
This method is best for data with a trend but no seasonality. It results in a straight, sloped-line forecast.

**Figure 67  Typical Double Exponential Smoothing Data, Fit, and Forecast Line**

---

**Classic Non-seasonal Forecasting Method Parameters**

The classic non-seasonal methods use several forecasting parameters. For the moving average methods, the formulas use one parameter, period. When performing a moving average, Predictor averages over a number of periods. For single moving average, the number of periods can be any whole number between 1 and half the number of data points. For double moving average, the number of periods can be any whole number between 2 and one-third the number of data points.

Single exponential smoothing has one parameter: alpha. Alpha (a) is the smoothing constant. The value of alpha can be any number between 0 and 1, not inclusive.

Double exponential smoothing has two parameters: alpha and beta. Alpha is the same smoothing constant as described above for single exponential smoothing. Beta (b) is also a smoothing constant exactly like alpha except that it is used during second smoothing. The value of beta can be any number between 0 and 1, not inclusive.

---

**Classic Seasonal Forecasting Methods**

Seasonal forecasting methods extend the non-seasonal forecasting methods by adding an additional component to capture the seasonal behavior of the data. Predictor offers four classic seasonal forecasting methods:

- “Seasonal Additive” on page 133
- “Seasonal Multiplicative” on page 133
- “Holt-Winters’ Additive” on page 133
- “Holt-Winters’ Multiplicative” on page 134

For associated parameters, see “Classic Seasonal Forecasting Method Parameters” on page 134.
**Seasonal Additive**

Calculates a seasonal index for historical data that does not have a trend. The method produces exponentially smoothed values for the level of the forecast and the seasonal adjustment to the forecast. The seasonal adjustment is added to the forecasted level, producing the seasonal additive forecast.

This method is best for data without trend but with seasonality that does not increase over time. It results in a curved forecast that reproduces the seasonal changes in the data.

![Typical Seasonal Additive Data, Fit, and Forecast Curve without Trend](image)

**Seasonal Multiplicative**

Calculates a seasonal index for historical data that does not have a trend. The method produces exponentially smoothed values for the level of the forecast and the seasonal adjustment to the forecast. The seasonal adjustment is multiplied by the forecasted level, producing the seasonal multiplicative forecast.

This method is best for data without trend but with seasonality that increases or decreases over time. It results in a curved forecast that reproduces the seasonal changes in the data.

![Typical Seasonal Multiplicative Data, Fit, and Forecast Curve without Trend](image)

**Holt-Winters’ Additive**

Is an extension of Holt’s exponential smoothing that captures seasonality. The method produces exponentially smoothed values for the level of the forecast, the trend of the forecast, and the seasonal adjustment to the forecast. This seasonal additive method adds the seasonality factor to the trended forecast, producing the Holt-Winters’ additive forecast.
This method is best for data with trend and seasonality that does not increase over time. It results in a curved forecast that shows the seasonal changes in the data.

**Figure 70 Typical Holt-Winters' Additive Data, Fit, and Forecast Curve**

**Holt-Winters' Multiplicative**

Is similar to the Holt-Winters’ additive method. Holt-Winters’ Multiplicative method also calculates exponentially smoothed values for level, trend, and seasonal adjustment to the forecast. This seasonal multiplicative method multiplies the trended forecast by the seasonality, producing the Holt-Winters’ multiplicative forecast.

This method is best for data with trend and with seasonality that increases over time. It results in a curved forecast that reproduces the seasonal changes in the data.

**Figure 71 Typical Holt-Winters’ Multiplicative Data, Fit, and Forecast Curve**

**Classic Seasonal Forecasting Method Parameters**

The seasonal forecast methods use three smoothing parameters: alpha, beta, and gamma:

- **alpha** ($\alpha$) — Smoothing parameter for the level component of the forecast. The value of alpha can be any number between 0 and 1, not inclusive.
- **beta** ($\beta$) — Smoothing parameter for the trend component of the forecast. The value of beta can be any number between 0 and 1, not inclusive.
- **gamma** ($\gamma$) — Smoothing parameter for the seasonality component of the forecast. The value of gamma can be any number between 0 and 1, not inclusive.
Each seasonal forecasting method uses some or all of these parameters, depending on the forecasting method. For example, the seasonal additive forecasting method does not account for trend, so it does not use the beta parameter.

**Time-series Forecasting Accuracy Measures**

One component of every time-series forecast is the data’s random error that is not explained by the forecast formula or by the trend and seasonal patterns. The error is measured by fitting points for the time periods with historical data and then comparing the fitted points to the historical data.

All the examples are based on the set of data illustrated in the following chart (Figure 72). Most of the formulas refer to the actual points ($Y$) and the fitted points ($Y_t$). In the chart, the horizontal axis illustrates the time periods ($t$) and the vertical axis illustrates the data point values.

![Figure 72 Sample Data](chart.png)

Predictor measures the error using one of the methods described in the following sections:

- “RMSE” on page 135, below
- “MAD” on page 136
- “MAPE” on page 136

Another statistic, “Theil’s U” on page 136, is used as a relative accuracy measure. Also see “Durbin-Watson” on page 137.

**RMSE**

RMSE (root mean squared error) is an absolute error measure that squares the deviations to keep the positive and negative deviations from cancelling out one another. This measure also tends to exaggerate large errors, which can help eliminate methods with large errors.
MAD

MAD (mean absolute deviation) is an absolute error measure that originally became very popular (in the days before hand-held calculators) because it did not require the calculation of squares or square roots. While it is still fairly reliable and widely used, it is most accurate for normally distributed data.

MAPE

MAPE (mean absolute percentage error) is a relative error measure that uses absolute values. MAPE has two advantages. First, the absolute values keep the positive and negative errors from cancelling out each other. Second, because relative errors do not depend on the scale of the dependent variable, this measure lets you compare forecast accuracy between differently scaled time-series data.

Theil’s U

Theil’s $U$ statistic is a relative accuracy measure that compares the forecasted results with the results of forecasting with minimal historical data. It also squares the deviations to give more weight to large errors and to exaggerate errors, which can help eliminate methods with large errors (Table 2).

Table 2  Interpreting Theil’s $U$

<table>
<thead>
<tr>
<th>Theil’s $U$ Statistic</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 1</td>
<td>The forecasting technique is better than guessing.</td>
</tr>
<tr>
<td>1</td>
<td>The forecasting technique is about as good as guessing.</td>
</tr>
<tr>
<td>More than 1</td>
<td>The forecasting technique is worse than guessing.</td>
</tr>
</tbody>
</table>

The formula for calculating Theil’s $U$ statistic:

$$U = \frac{\sum_{t=1}^{n-1} \left( \frac{\hat{Y}_{t+1} - Y_{t+1}}{Y_t} \right)^2}{\sum_{t=1}^{n-1} \left( \frac{Y_{t+1} - Y_t}{Y_t} \right)^2}$$

where $Y_t$ is the actual value of a point for a given time period $t$, $n$ is the number of data points, and

$$\hat{Y}_t$$

is the forecasted value.
**Durbin-Watson**

Detects autocorrelation at lag 1. This means that each time-series value influences the next value. This is the most common type of autocorrelation. For the formula, see “Durbin-Watson Statistic” on page 157.

This statistic can have any value between 0 and 4. Values indicate slow-moving, none, or fast-moving autocorrelation (Table 3).

<table>
<thead>
<tr>
<th>Durbin-Watson Statistic</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 1</td>
<td>The errors are positively correlated. An increase in one period follows an increase in the previous period.</td>
</tr>
<tr>
<td>2</td>
<td>No autocorrelation.</td>
</tr>
<tr>
<td>More than 3</td>
<td>The errors are negatively correlated. An increase in one period follows a decrease in the previous period.</td>
</tr>
</tbody>
</table>

Avoid using independent variables that have errors with a strong positive or negative correlation, because this can lead to an incorrect forecast for the dependent variable.

**Time-series Forecasting Techniques**

**Subtopics**

- Standard Forecasting
- Simple Lead Forecasting
- Weighted Lead Forecasting
- Holdout Forecasting

Predictor uses one of four forecasting techniques to perform time-series forecasting:

- “Standard Forecasting” on page 137
- “Simple Lead Forecasting” on page 138
- “Weighted Lead Forecasting” on page 138
- “Holdout Forecasting” on page 138

Classic time-series forecasting methods can use any of these techniques, but ARIMA methods can use only Standard and Holdout forecasting.

**Standard Forecasting**

Standard forecasting optimizes the forecasting parameters to minimize the error measure between the fit values and the historical data for the same period. For example, consider showing historical data and calculated fit values for periods 1 through 7.
## Table 4  Example of Historical Data and Fit with Standard Forecasting

<table>
<thead>
<tr>
<th>Period</th>
<th>Historical Data Value</th>
<th>Fit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>472</td>
<td>488</td>
</tr>
<tr>
<td>2</td>
<td>599</td>
<td>609</td>
</tr>
<tr>
<td>3</td>
<td>714</td>
<td>702</td>
</tr>
<tr>
<td>4</td>
<td>892</td>
<td>888</td>
</tr>
<tr>
<td>5</td>
<td>874</td>
<td>890</td>
</tr>
<tr>
<td>6</td>
<td>896</td>
<td>909</td>
</tr>
<tr>
<td>7</td>
<td>890</td>
<td>870</td>
</tr>
</tbody>
</table>

Predictor calculates the RMSE using the differences between the historical data and the fit data from the same periods. For example:

\[
(472-488)^2 + (599-609)^2 + (714-702)^2 + (892-888)^2 + \ldots
\]

For standard forecasting, Predictor optimizes the forecasting parameters so that the RMSE calculated in this way is minimized.

### Simple Lead Forecasting

Simple lead forecasting optimizes the forecasting parameters to minimize the error measure between the historical data and the fit values, offset by a specified number of periods (lead). Use this forecasting technique when a forecast for some future time period has the greatest importance, more so than the forecasts for the previous or later periods. For example, single lead forecasting can be used if your company must order extremely expensive manufacturing components two months in advance, making any forecast for two months out the most important.

### Weighted Lead Forecasting

Weighted lead forecasting optimizes the forecasting parameters to minimize the average error measure between the historical data and the fit values, offset by 0, 1, 2, and so on, up to the specified number of periods (weighted lead). It uses the simple lead technique for several lead periods and then averages the forecast over the periods, optimizing this average value. Use this technique when the future forecast for several periods is most important. For example, weighted lead forecasting can be used if your company must order extremely expensive manufacturing components zero, one, and two months in advance, making any forecast for all the time periods up to two months out the most important.

### Holdout Forecasting

Holdout forecasting:
1. Removes the last few data points of the historical data.
2. Calculates the fit and forecast points using the remaining historical data.
3. Compares the error between the forecasted points and their corresponding, excluded, historical data points.
4. Changes the parameters to minimize the error between the forecasted points and the excluded points.

 Predictor determines the optimal forecast parameters using only the non-holdout set of data.

 Notice that if you have a small amount of data and want to use seasonal forecasting methods, using the holdout technique may restrict you to non-seasonal methods.

 For more information on the holdout technique and when to use it effectively, see the Makridakis, Wheelwright, and Hyndman reference in the Bibliography.

**Multiple Linear Regression**

Multiple linear regression is used for data where one data series (the dependent variable) is a function of, or depends on, other data series (the independent variables). For example, the yield of a lettuce crop depends on the amount of water provided, the hours of sunlight each day, and the amount of fertilizer used.

The goal of multiple linear regression is to find an equation that most closely matches the historical data. “Multiple” indicates that you can use more than one independent variable to define the dependent variable in the regression equation. “Linear” indicates that the regression equation is a linear equation.

The linear equation describes how the independent variables \(x_1, x_2, x_3, \ldots\) combine to define the single dependent variable \(y\). Multiple linear regression finds the coefficients for the equation:

\[ y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots + e \]

where \(b_1, b_2, \text{ and } b_3\), are the coefficients of the independent variables, \(b_0\) is the \(y\)-intercept constant, and \(e\) is the error.

Equations with only one independent variable define a straight line. This uses a special case of multiple linear regression called simple linear regression, with the equation:

\[ y = b_0 + b_1 x + e \]

where \(b_0\) is where the regression line crosses the graph’s \(y\) axis, \(x\) is the independent variable, and \(e\) is the error. When the regression equation has only two independent variables, it defines a plane. When the regression equation has more than two independent variables, it defines a hyperplane.

To find the coefficients of these equations, Predictor uses singular value decomposition. For more information on this technique, see “Regression Methods” on page 160.
Regression Methods

Predictor uses one of three methods for calculating multiple linear regression:

- “Standard Regression” on page 140
- “Forward Stepwise Regression” on page 140
- “Iterative Stepwise Regression” on page 141

Standard Regression

Standard regression performs multiple linear regression, generating regression coefficients for each independent variable you specify, no matter how significant.

Forward Stepwise Regression

Forward stepwise regression adds one independent variable at a time to the multiple linear regression equation, starting with the independent variable with the most significant probability of the correlation (partial $F$ statistic). It then recalculates the partial $F$ statistic for the remaining independent variables, taking the existing regression equation into consideration.

The resulting multiple linear regression equation will always have at least one independent variable.

Forward stepwise regression continues to add independent variables until either:

- It runs out of independent variables.

For more information on multiple linear regressions, see:

- “Regression Methods” on page 140
- “Regression Statistics” on page 142
It reaches one of the selected stopping criteria in the Stepwise Options dialog.

The number of included independent variables reaches one-third the number of data points in the series.

Two stopping criteria:

- $R^2$ — Stops the stepwise regression if the difference between a specified statistic (either $R^2$ or adjusted $R^2$) for the previous and new regression solutions is below a threshold value. When this happens, Predictor does not use the last independent variable. For example, the third step of a stepwise regression results in an $R^2$ value of 0.81, and the fourth step adds another independent variable and results in an $R^2$ value is 0.83. The difference between the $R^2$ values is 0.02. If the threshold value is 0.03, Predictor returns to the regression equation for the third step and stops the stepwise regression.

- Partial $F$-test significance — Stops the stepwise regression if the probability of the partial $F$ statistic for a new solution is above a maximum value. For example, if you set the maximum probability to 0.05 and the partial $F$ statistic for the fourth step of a stepwise regression results in a probability of 0.08, Predictor returns to the regression equation for the third step and stops the stepwise regression.

**Iterative Stepwise Regression**

Iterative stepwise regression adds or removes one independent variable at a time to or from the multiple linear regression equation.

To perform iterative stepwise regression, Predictor:

1. Calculates the partial $F$ statistic for each independent variable.
2. Adds the independent variable with the most significant correlation (partial $F$ statistic).
3. Checks the partial $F$ statistic of the independent variables in the regression equation to see if any became insignificant (have a probability below the minimum) with the addition of the latest independent variable.
4. Removes the least significant of any insignificant independent variables one at a time.
5. Repeats step 3 until no insignificant variables remain in the regression equation.
6. Repeats steps 1 through 5 until one of the following occurs:
   - The model runs out of independent variables.
   - The regression reaches one of the stopping criteria (see the for information on how the stopping criteria work).
   - The same independent variable is added and then removed.

The resulting equation always has at least one independent variable.
Regression Statistics

After Predictor finds the regression equation, it calculates several statistics to help you evaluate the regression:

- “R\(^2\)” on page 142
- “Adjusted R\(^2\)” on page 142
- “Sum of Squared Errors (SSE)” on page 142
- “F statistic” on page 142
- “t statistic” on page 143
- “p” on page 143

\(R^2\)

Coefficient of determination. This statistic indicates the percentage of the variability of the dependent variable that the regression equation explains.

For example, an \(R^2\) of 0.36 indicates that the regression equation accounts for 36% of the variability of the dependent variable.

**Adjusted \(R^2\)**

Corrects \(R^2\) to account for the degrees of freedom in the data. In other words, the more data points you have, the more universal the regression equation is. However, if you have only the same number of data points as variables, the \(R^2\) may be deceptively high. This statistic corrects for that.

For example, the \(R^2\) for one equation may be very high, indicating that the equation accounted for almost all the error in the data. However, this value may be inflated if the number of data points was insufficient to calculate a universal regression equation.

**Sum of Squared Errors (SSE)**

The least squares technique for estimating regression coefficients minimizes this statistic, which measures the error not eliminated by the regression line.

For any line drawn through a scatter plot of data, several ways can be used to determine which line fits the data best. One method used to compare the fit of lines is to calculate the SSE (sum of the squared errors, or deviations) for each line. The lower the SSE, the better the fit of the line to the data.

**F statistic**

Tests the significance of the regression equation as measured by \(R^2\). A significant value means that the regression equation accounts for some of the variability of the dependent variable.
**t statistic**

Tests the significance of the relationship between the coefficients of the dependent variable and an individual independent variable, in the presence of the other independent variables. A significant value means that the independent variable contributes to the dependent variable.

**p**

Indicates the probability of the calculated $F$ or $t$ statistic being as large as it is (or larger) by chance. A low $p$ value is good and means that the $F$ statistic is not coincidental and, therefore, is significant. A significant $F$ statistic means that the relationship between the dependent variable and the combination of independent variables is significant.

Generally, $p$ should be less than 0.05.

**Historical Data Statistics**

Predictor automatically calculates the following statistics for historical data series:

- “Mean” on page 143
- “Standard Deviation” on page 143
- “Minimum” on page 144
- “Maximum” on page 144
- “Ljung-Box Statistic” on page 144

If an ARIMA model is displayed in the Method box, the transformation lambda and three selection criteria values are also included in the Statistics group. For more information about these values, see “ARIMA Time-series Forecasting Formulas” on page 158 and the Bibliography.

**Mean**

The mean of a set of values is found by adding the values and dividing their sum by the number of values. “Average” usually refers to the mean. For example, 5.2 is the mean or average of 1, 3, 6, 7, and 9.

**Standard Deviation**

The standard deviation is the square root of the variance for a distribution. Like the variance, it is a measure of dispersion about the mean and is useful for describing the “average” deviation.

For example, you can calculate the standard deviation of the values 1, 3, 6, 7, and 9 by finding the square root of the variance that is calculated in the variance example below.

The standard deviation, denoted as $s$, is calculated from the variance as follows:

$$s = \sqrt{10.2} = 3.19$$
where the variance is a measure of the dispersion, or spread, of a set of values about the mean. When values are close to the mean, the variance is small. When values are widely scattered about the mean, the variance is larger.

**Minimum**

The minimum is the smallest value in the data range.

**Maximum**

The maximum is the largest value in the data range.

**Ljung-Box Statistic**

Measures whether a set of autocorrelations is significantly different from a set of autocorrelations that are all zero. See for the formula.

**Events Adjustment**

The events feature of Predictor enables users to define events, identifiable occurrences that have affected historical data and could affect predicted data. Users can define events for historical and predicted data.

When forecasting from historical data that includes a defined event, the effects of the event are removed from the data and then the forecasting methods are fit to the adjusted data. If there are repeating occurrences of events in the forecasted period, the effects of the events are applied to the forecasts obtained.

Notice that the adjusted data does not appear in the charts. Predictor shows the original data and the fits are adjusted visually to reflect the effects of the events. Any manual adjustments to the forecasts are performed after the effects of the events are applied to them.

For more information about events, see the *Oracle Crystal Ball Predictor User’s Guide*.

**Data Screening and Adjustment Methods**

**Subtopics**

- Outlier Detection Methods
- Outlier and Missing Value Adjustment Methods

Historical data can have missing values and outliers, which are data points that differ significantly from the rest of the data. Settings in the Data Attributes panel of the Predictor wizard enable you to select several ways of handling missing values and identifying and adjusting outliers. Because adjusted outliers are treated as missing values, both of these situations are discussed and handled together.
Outlier Detection Methods

Predictor offers three methods for detecting outliers, or significantly extreme values:

- “Mean and Standard Deviation Method” on page 145
- “Median and Median Absolute Deviation Method (MAD)” on page 145
- “Median and Interquartile Deviation Method (IQD)” on page 146

In each case, the difference is calculated between historical data points and values calculated by the various forecasting methods. These differences are called residuals. They can be positive or negative depending on whether the historical value is greater than or less than the smoothed value. Various statistics are then calculated on the residuals and these are used to identify and screen outliers.

A certain number of values must exist before the data fit can begin. If outliers occur at the beginning of the data, they are not detected.

**Note:** Time-series data is typically treated differently from other data because of its dynamic nature, such as the pattern in the data. A time-series outlier need not be extreme with respect to the total range of the data variation but it is extreme relative to the variation locally.

### Mean and Standard Deviation Method

For this outlier detection method, the mean and standard deviation of the residuals are calculated and compared. If a value is a certain number of standard deviations away from the mean, that data point is identified as an outlier. The specified number of standard deviations is called the threshold. The default value is 3.

This method can fail to detect outliers because the outliers increase the standard deviation. The more extreme the outlier, the more the standard deviation is affected.

### Median and Median Absolute Deviation Method (MAD)

For this outlier detection method, the median of the residuals is calculated. Then, the difference is calculated between each historical value and this median. These differences are expressed as their absolute values, and a new median is calculated and multiplied by an empirically derived constant to yield the median absolute deviation (MAD). If a value is a certain number of MAD away from the median of the residuals, that value is classified as an outlier. The default threshold is 3 MAD.

This method is generally more effective than the mean and standard deviation method for detecting outliers, but it can be too aggressive in classifying values that are not really extremely different. Also, if more than 50% of the data points have the same value, MAD is computed to be 0, so any value different from the residual median is classified as an outlier.
Median and Interquartile Deviation Method (IQD)

For this outlier detection method, the median of the residuals is calculated, along with the 25th percentile and the 75th percentile. The difference between the 25th and 75th percentile is the interquartile deviation (IQD). Then, the difference is calculated between each historical value and the residual median. If the historical value is a certain number of MAD away from the median of the residuals, that value is classified as an outlier. The default threshold is 2.22, which is equivalent to 3 standard deviations or MADs.

This method is somewhat susceptible to influence from extreme outliers, but less so than the mean and standard deviation method. Box plots are based on this approach. The median and interquartile deviation method can be used for both symmetric and asymmetric data.

Outlier and Missing Value Adjustment Methods

Predictor provides two methods for filling in missing values and adjusting outliers:

- “Cubic Spline Interpolation Method” on page 146
- “Neighbor Interpolation Method” on page 146

Missing values at the beginning of a data series are ignored. Missing values at the end of a data series are allowed, but this condition is not ideal. The cubic spline interpolation method is especially sensitive to data missing at the end of the series. If one or two values are missing, cubic spline interpolation can be used. If multiple values are missing, neighbor interpolation provides a better estimate.

**Tip:** An obvious outlier, such as a large data spike, should be replaced by a blank cell in the original data set. Otherwise, neighbor interpolation is probably a better adjustment method, especially if the specified neighbors do not include the spike. Because cubic spline interpolation takes into account the whole data set, that adjustment method will be affected by the outlier.

Cubic Spline Interpolation Method

Cubic spline interpolation is based on a drafting tool used to draw smooth curves through a number of points. The spline tool consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, resulting in a smooth curve.

The cubic spline interpolation is a piecewise continuous curve that passes through each value in the data set. Each interval of the curve has a separate cubic polynomial, each with its own coefficients. These are equivalent to the spline tool's weights. The cubic spline interpolation method considers the entire data set when adjusting outliers and filling in missing values.

Neighbor Interpolation Method

This method is also called the single imputation method. A certain number of neighbors on each side of the missing value are considered when estimating the missing value. This value is called
n, and the total number of neighbors evaluated is $2n$. The missing data is replaced by the mean or median of the $2n$ data points. The default value for $n$ is 1, and the default statistic is Mean.

**Note:** To preserve the local nature of time series data, the number of neighbors on each side of the missing value or values should be small, ideally $n = 1$ or $n = 2$.

### Techniques and Formulas

**Subtopics**

- Time-Series Forecasting Techniques
- Classic Time-series Forecasting Method Formulas
- Error Measure and Statistic Formulas
- ARIMA Time-series Forecasting Formulas
- Regression Methods
- Regression Statistic Formulas

This section contains techniques and formulas used in Predictor.

### Time-Series Forecasting Techniques

This section discusses statistics related to time-series forecasting techniques available in Predictor:

- “Standard Forecasting” on page 148
- “Holdout Forecasting” on page 148
- “Simple Lead Forecasting” on page 148
- “Weighted Lead Forecasting” on page 149

Related terms:

- **Time series**—The original data, expressed as $Y_t$
- **Fit array**—A retrofit of the time series, consisting of one-period-ahead forecasts performed from the data of previous periods; expressed as $F_t$
- **Residual array**—A set of positive or negative residuals, expressed as $r_t$, and defined as $r_t = Y_t - F_t$
- **RMSE**—Root mean square error for forecasting, calculated as described in “RMSE” on page 155, where $n$ is the number of periods for which a fit is available. RMSE depends on the specific forecasting method and technique.
- **Forecasts**—Value projections calculated using the formula for the specific method; they are 1 to $k$ periods ahead, where $k$ is the number of forecasts required
- **Standard error of forecasts**—Used to calculate confidence intervals; see “Confidence Intervals” on page 156
**Standard Forecasting**

In standard forecasting, if the method parameters are already provided by the user, the following are calculated: RMSE and other error measures, forecasts, and standard error. If the parameters are not provided by the user, then the parameters are optimized to minimize the error measure between the fit values and the historical data for the same period.

**Holdout Forecasting**

In holdout forecasting:

- The last few data points are removed from the data series. The remaining historical data series is called in-sample data, and the holdout data is called out-of-sample data. Suppose \( p \) periods have been removed as holdout from a total of \( N \) periods.
- Parameters are optimized by minimizing the fit error measure for in-sample data. If method parameters are provided by the user, those are used in the final forecasting.
- After the parameters are optimized, the forecasts for the holdout periods (\( p \) periods) are calculated.
- The error statistics (RMSE, MAD, MAPE) are out-of-sample statistics, based on only the numbers in the hold-out period. The RMSE for holdout forecasting is often called holdout RMSE. The holdout error measures are the ones reported to the user and are used to sort the forecasting methods.
- Other statistics such as Theil's \( U \), Durbin-Watson, and Ljung-Box are in-sample statistics, based on the non-holdout period.
- Final forecasting is performed on both the in-sample and out-of-sample periods (all \( N \) periods) using the standard technique.
- The standard error for the forecasts is also calculated using all \( N \) periods.

To improve the optimized parameter values obtained for the method, holdout forecasting should be used only when there are at least 100 data points for non-seasonal methods and 5 seasons for seasonal methods. For best results, use no more than 5 percent of the data points as holdout, no matter how large the number of total data points.

**Simple Lead Forecasting**

Simple lead forecasting optimizes the forecasting parameters to minimize the error measure between the historical data and the fit values, both offset by a specified number of periods (lead). Use this forecasting technique when a forecast for some future time period has more importance than forecasts for previous or later periods.

For example, suppose a company must order extremely expensive manufacturing components two months in advance, making the forecast for two months out the most important. In this case, the company could use simple lead forecasting with a lead of 2 periods.

In simple lead forecasting:
The fit for period \( t \) is calculated as the (lead)-period-ahead forecast from period \( t = 0 \). The fit for \( t = 1 \) calculated with simple lead forecasting is the same as the fit for the standard forecast, which is a 1-period-ahead forecast from period \( t = 0 \).

The residual at period \( t \) is calculated as the difference between the historical value at period \( t \) and the lead-period-fit obtained for period \( t \).

The lead RMSE is calculated as the root mean square of the residuals as calculated previously.

The forecasts for future periods and the standard errors for those forecasts are calculated as for standard forecasts.

If the method parameters are already provided by the user, simple lead forecasting is performed as described previously. If the parameters are not provided, then the parameters are optimized to minimize the lead error measure (for example, lead RMSE). After the parameters are optimized, the fit and the forecast are then calculated as for standard forecasting method.

**Weighted Lead Forecasting**

Weighted lead forecasting optimizes the forecasting parameters to minimize the average error measure between the historical data and the fit values, offset by 1, 2, and so on, up to the specified number of periods (lead value). It uses the simple lead technique for several lead periods, averages the error measure over the periods, and then optimizes this average value to obtain the method parameters.

Use weighted lead forecasting when the future forecast for several periods is most important. For example, suppose your company must order extremely expensive manufacturing components one and two months in advance, making forecasts for all the time periods up to two months out the most important.

In weighted lead forecasting:

- Simple lead error measures are calculated for lead values from 1 to the specified lead value.
- The weighted lead RMSE is calculated as the average of the simple lead RMSEs starting from lead value = 1 to the specified lead value period. For a lead value of 3, simple lead error measures for 1, 2, and 3 are obtained and then averaged to get the weighted lead RMSE.
- Method parameters are then obtained by minimizing the weighted lead RMSE.
- After the parameters are obtained, forecasts for future periods and the standard errors for those forecasts are calculated as for standard forecasts.

If method parameters are provided by the user, weighted lead forecasting is performed as described previously. If the parameters are not provided, they are optimized to minimize the weighted lead error measure, such as weighted lead RMSE.

After parameters are optimized, the fit and the forecast are then calculated as for the standard forecasting method. For a lead value = 1, weighted lead forecasting is the same as simple lead forecasting and standard forecasting.
**Classic Time-series Forecasting Method Formulas**

This section provides formulas for the following classic time-series forecasting methods used in Predictor:

- “Classic Nonseasonal Forecasting Method Formulas” on page 150
- “Classic Seasonal Forecasting Method Formulas” on page 151

For ARIMA formulas, see “ARIMA Time-series Forecasting Formulas” on page 158.

**Classic Nonseasonal Forecasting Method Formulas**

Formulas for the classic nonseasonal time-series forecasting methods:

- “Single Moving Average” on page 150
- “Double Moving Average” on page 150
- “Single Exponential Smoothing” on page 151
- “Double Exponential Smoothing” on page 151

**Single Moving Average**

Single moving average formulas:

\[ F_t = \frac{1}{p} \sum_{k=t}^{t-p+1} Y_k \]

(Fit)

\(F_{t+m} = F_t\)

where the parameters are:

p—Order of moving average

**Note:** First fit is available from period \((p + 1)\)

**Double Moving Average**

Predictor uses the following equations for the double moving average method:

(Level) \( L_t = 2 \times M_t - M_t' \)

\[ T_t = \frac{2}{p-1}(M_t - M_t') \]

(Trend)

(Fit) \( F_t = L_{t-1} + T_{t-1} \)

(Forecast for period \( m \)) \( F_{t+m} = L_t + m \times T_t \)

Where the parameters are:

p—Order of moving average
Mt—First order moving average for period t
M'_t—Second order moving average for period t

Note: First fit is available from period (2*p-1).

**Single Exponential Smoothing**
Predictor uses the following formulas for single exponential smoothing:

(Initialization) \( F_1 = 0, F_2 = Y_1 \)

(Fit) \( F_t = \alpha \times Y_{t-1} + (1-\alpha) \times F_{t-1} \)

(Forecast for period m) \( F_{t+m} = F_t \)

Note: First fit is available from period 2.

**Double Exponential Smoothing**
uses Holt’s double exponential smoothing formula as follows:

(Initialization) \( L_1 = Y_1, T_1 = 0 \)

Level: \( L_t = \alpha \times Y_t + (1-\alpha) \times (L_{t-1} + T_{t-1}) \)

Trend: \( T_t = \beta \times (L_t - L_{t-1}) + (1-\beta) \times T_{t-1} \)

Fit: \( F_t = \alpha \times Y_{t-1} + (1-\alpha) \times F_{t-1} \)

Forecast for period m: \( F_{t+m} = L_t + m\times T_t \)

Note: First fit is available from period 2.

**Classic Seasonal Forecasting Method Formulas**
Formulas for the classic seasonal time-series Predictor forecasting methods:

- “Seasonal Additive Smoothing” on page 151
- “Seasonal Multiplicative Smoothing” on page 152
- “Holt-Winters’ Additive Seasonal Smoothing” on page 153
- “Holt-Winters’ Multiplicative Seasonal Smoothing” on page 153

**Seasonal Additive Smoothing**
Crystal Ball uses the following initialization equation for this method:
\[
P = \frac{\sum_{t=1}^{s} Y_t}{s}
\]

Set \( L_t = P, S_t = Y_t - P \) for \( t = 1 \) to \( s \)

Crystal Ball uses the following equations to calculate this method:

- (Level) \( L_t = \alpha \times (Y_t - S_{t,s}) + (1 - \alpha) \times L_{t-1} \)
- (Seasonal) \( S_t = \gamma \times (Y_t - L_t) + (1 - \gamma) \times S_{t,s} \)
- (Forecast for period \( m \)) \( F_{t+m} = L_t + S_{t+m,s} \)

where the parameters are:

\( \alpha \) — Alpha
\( \gamma \) — Gamma

\( m \) — Number of periods ahead to forecast
\( s \) — Length of seasonality
\( L_t \) — Level of the series at time \( t \)
\( S_t \) — Seasonal component at time \( t \)

**Note:** First fit is available from period \((s + 1)\)

### Seasonal Multiplicative Smoothing

Crystal Ball uses the following initialization equation for this method:

\[
P = \frac{\sum_{t=1}^{s} Y_t}{s}
\]

Set \( L_t = P, S_t = Y_t / P \) for \( t = 1 \) to \( s \)

Crystal Ball uses the following equations to calculate this method:

- (Level) \( L_t = \alpha \times (Y_t / S_{t,s}) + (1 - \alpha) \times L_{t-1} \)
- (Seasonal) \( S_t = \gamma \times (Y_t / L_t) + (1 - \gamma) \times S_{t,s} \)
- (Forecast for period \( m \)) \( F_{t+m} = L_t \times S_{t+m,s} \)

where the parameters are:

\( \alpha \) — Alpha
\( \gamma \) — Gamma

\( m \) — Number of periods ahead to forecast
s—Length of seasonality  
$L_t$—Level of the series at time $t$  
$S_t$—Seasonal component at time $t$

**Note:** First fit is available from period ($s + 1$)

**Holt-Winters’ Additive Seasonal Smoothing**

To find the initial values:

Calculate:

$$P = \frac{\sum_{t=1}^{s} Y_t}{s}$$

Set: $L_1 = P$, $b_1 = 0$, $S_t = Y_t - P$, for $t = 1$ to $s$

For the remaining periods, use the following formulas:

- (Level) $L_t = \alpha \cdot (Y_t - S_{t-s}) + (1 - \alpha) \cdot (L_{t-1} + b_{t-1})$
- (Trend) $b_t = \beta \cdot (L_t - L_{t-1}) + (1 - \beta) \cdot b_{t-1}$
- (Seasonal) $S_t = \gamma \cdot (Y_t - L_t) + (1 - \gamma) \cdot S_{t-s}$
- (Forecast for period $m$) $F_{m+m} = L_i + mb_i + S_{i+m-s}$

where the parameters are:

- $\alpha$—Alpha
- $\beta$—Beta
- $\gamma$—Gamma

$m$—Number of periods ahead to forecast  
$s$—Length of the seasonality  
$L_t$—Level of the series at time $t$  
$b_t$—Trend of the series at time $t$  
$S_t$—Seasonal component at time $t$

**Note:** First fit is available from period ($s + 1$)

**Holt-Winters’ Multiplicative Seasonal Smoothing**

To find the initial values:
Calculate: \[ P = \frac{1}{s} \sum_{t=1}^{s} Y_t \]

Set: \( L_t = P, \ b_t = 0, \ S_t = \frac{Y_t}{P} \), for \( t = 1 \) to \( s \)

For the remaining periods, use the following formulas:

(Level) \( L_t = \alpha \ast (Y_t / S_{t-s}) + (1 - \alpha) \ast (L_{t-1} + b_{t-1}) \)

(Trend) \( b_t = \beta \ast (L_t - L_{t-1}) + (1 - \beta) \ast b_{t-1} \)

(Seasonal) \( S_t = \gamma \ast (Y_t / L_t) + (1 - \gamma) \ast S_{t-s} \)

(Forecast for period \( m \)) \( F_{t+m} = (L_t + m^\ast b_t) \ast S_{t+m-s} \)

where the parameters are:

\( \alpha \) — Alpha

\( \beta \) — Beta

\( \gamma \) — Gamma

\( m \) — Number of periods ahead to forecast

\( s \) — Length of the seasonality

\( L_t \) — Level of the series at time \( t \)

\( b_t \) — Trend of the series at time \( t \)

\( S_t \) — Seasonal component at time \( t \)

Note: First fit is available from period \((s + 1)\)

### Error Measure and Statistic Formulas

This section provides formulas for the following types of statistics used in Predictor:

- “Time-Series Forecast Error Measures” on page 154
- “Confidence Intervals” on page 156
- “Autocorrelation Statistics” on page 156

### Time-Series Forecast Error Measures

Crystal Ball calculates three different error measures for the fit of each time-series forecast. Uses one of these error measures to determine which time-series forecasting method is the best:

- “RMSE” on page 155
- “MAD” on page 155
- “MAPE” on page 155
**RMSE**

Root mean squared error is an absolute error measure that squares the deviations to keep the positive and negative deviations from canceling one another out. This measure also tends to exaggerate large errors, which can help when comparing methods.

The formula for calculating RMSE:

$$\sqrt{\frac{1}{n} \sum_{t=1}^{n} (Y_t - \hat{Y}_t)^2}$$

where $Y_t$ is the actual value of a point for a given time period $t$, $n$ is the total number of fitted points, and

$\hat{Y}_t$

is the fitted forecast value for the time period $t$.

**MAD**

Mean absolute deviation is an error statistic that averages the distance between each pair of actual and fitted data points.

The formula for calculating the MAD:

$$\frac{1}{n} \sum_{t=1}^{n} |Y_t - \hat{Y}_t|$$

where $Y_t$ is the actual value of a point for a given time period $t$, $n$ is the total number of fitted points, and

$\hat{Y}_t$

is the forecast value for the time period $t$.

**MAPE**

Mean absolute percentage error is a relative error measure that uses absolute values to keep the positive and negative errors from canceling one another out and uses relative errors to enable you to compare forecast accuracy between time-series models.

The formula for calculating the MAPE:

$$\frac{1}{n} \sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| (100)$$

where $Y_t$ is the actual value of a point for a given time period $t$, $n$ is the total number of fitted points, and
\( \hat{Y}_t \)

is the forecast value for the time period \( t \).

**Note:** If \( Y_t \) equals zero, drops the term:

\[
\frac{(Y_t - \hat{Y}_t)}{Y_t}
\]

**Confidence Intervals**

The confidence interval defines the range within which a forecasted value has some probability of occurring. Predictor uses an empirical method of calculating confidence intervals, using the standard error of forecasts:

- For an \( m \)-period-ahead forecast, the error term \( r_t(m) \) is defined as \( Y_t - F_t(m) \), where \( F_t(m) \) is the \( m \)-period-ahead fit for period \( t \).
- The standard error of prediction for an \( m \)-period-ahead forecast is then expressed as
  \[
  S(m) = \sqrt{\frac{\sum [r_t(m)]^2}{n}}
  \]
  where \( n \) is the number of periods for which \( r_t(m) \) is defined.

Assuming that forecast errors are normally distributed, the formula for predicting the future value of \( Y_{t+m} \) at time \( t \) within a 95 percent confidence interval is

\[
\hat{Y}_{t+m}(t) \pm 1.95996 \ S(m)
\]

The empirical method is reasonably accurate when historical data amount is sufficiently large.

**Autocorrelation Statistics**

Measures of autocorrelation describe the relationship among values of the same data series at different time periods.

The number of autocorrelations calculated is equal to the effective length of the time series divided by 2, where the effective length of a time series is the number of data points in the series without the pre-data gaps. The number of autocorrelations calculated ranges between a minimum of 2 and a maximum of 400.

Autocorrelation formula:
where \( r_k \) is the autocorrelation for lag \( k \).

Related statistics:
- “Autocorrelation Probability” on page 157
- “Durbin-Watson Statistic” on page 157
- “Ljung-Box Statistic” on page 158

**Autocorrelation Probability**

Autocorrelation probability is the probability of obtaining a certain autocorrelation for a particular data series by chance alone, if the data were completely random. To calculate autocorrelation probability:

- Calculate the standard error of autocorrelation:
  
  \[
  SE(r_k) = \frac{1 + 2 \sum_{i=1}^{k-1} r_i^2}{n}
  \]

  where
  
  \( SE(r_k) \) = standard error of autocorrelation at lag \( k \)
  
  \( r_i \) = autocorrelation at lag \( i \)
  
  \( k \) = the time lag
  
  \( n \) = number of observations in the time series


- Calculate the \( t \) statistic:

  \[
  t = \frac{r_k}{SE(r_k)}
  \]

- Calculate the \( p \)-value from the absolute \( t \) statistic; the probability is double the area of \( (1 - CDF(t)) \)

**Durbin-Watson Statistic**

The Durbin-Watson statistic calculates autocorrelation at lag 1.

The formula for calculating the Durbin-Watson statistic:

\[
\sum_{t = k+1}^{n} (y_t - \bar{y})(y_{t-k} - \bar{y})
\]

\[
\frac{k+1}{\sum_{t=1}^{n} (y_t - \bar{y})^2}
\]
where \( e_t \) is the difference between the estimated point \( \hat{Y}_t \)
and the actual point \( Y_t \) and \( n \) is the number of data points.

**Ljung-Box Statistic**

This statistic measures whether a set of autocorrelations is significantly different from a set of autocorrelations that are all zero. The formula for calculating the Ljung-Box statistic:

\[
Q' = \frac{n(n + 2)}{n - k} \sum_{k=1}^{h-1} \frac{r_k^2}{(n-k)}
\]

where:

- \( Q' \) is the Ljung-Box statistic; the probability that the set of autocorrelations is the same as a set of autocorrelations which are all 0.
- \( n \) is the amount of data in the data sample.
- \( h \) is the size of the set of autocorrelations used to calculate the statistic.
- \( r_k \) is the autocorrelation with a lag of \( k \).

The size of the set of autocorrelations is equal to one-third the size of the data sample (or 100, if the sample is greater than 300).

**ARIMA Time-series Forecasting Formulas**

Subtopics

- ARIMA Equations
- Estimation of ARIMA Model Coefficients
- ARIMA Constants
- Stationarity

This topic provides basic formulas for the ARIMA (autoregressive integrated moving average) model implementation used in Predictor. For more information, see the references in the ARIMA section of “Bibliography” included in the *Oracle Crystal Ball Predictor User’s Guide*.

For classic time-series forecasting formulas, see “Classic Time-series Forecasting Method Formulas” on page 150.
**ARIMA Equations**

- Equation for a $p$-th order autoregressive (AR) model — that is, AR($p$) model:
  \[ y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t \]
  Where \( \{y_t\} \) is the data on which the ARMA model is to be applied. That means, the series is already power-transformed and differenced, in that order. The parameters \( \phi_1, \phi_2 \) and so on are AR coefficients.

- Equation for a $q$-th order moving average (MA) model — that is, MA($q$) model:
  \[ y_t = C + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \ldots - \theta_q \epsilon_{t-q} \]
  Where \( \{y_t\} \) is as defined previously and \( \theta_1, \theta_2 \) and so on are MA coefficients.

- Equation for an ARMA($p,q$) model:
  \[ y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \ldots - \theta_q \epsilon_{t-q} \]
  Where \( \{y_t\}, \phi_1, \phi_2, \ldots, \theta_1, \theta_2 \ldots \) are as defined previously.

- Equation for a SARMA($p,q$)($P,Q$) model (seasonal):
  \[ y_t = C + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{P} \Phi_i y_{t-is} + \epsilon_t - \sum_{i=1}^{q} \theta_i \epsilon_{t-i} - \sum_{i=1}^{Q} \Theta_i \epsilon_{t-is} \]
  Where \( \{y_t\}, \{\phi\}, \text{and} \ \{\theta\} \) are as defined previously, and \( \{\Phi\} \) and \( \{\Theta\} \) are the seasonal counterparts.

**Estimation of ARIMA Model Coefficients**


**ARIMA Constants**

The constant term in an ARIMA equation introduces deterministic trend into the model and extends that trend indefinitely into the future. If the model includes a single difference (either non-seasonal or seasonal) and the constant term is present, the trend is linear, where a twice-differenced model has quadratic trend. The *AutoSelect* setting for ARIMA constants in the Predictor ARIMA Options dialog leaves out the constant term in the model if it contains one or more non-seasonal or seasonal differences.

When the constant term is included in the model, the value of the term is calculated by the following equation:

\[ C = \mu \times (1 - \sum \phi_i) (1 - \sum \Phi_i) \]
Where $\phi_i$ are non-seasonal AR coefficients, $\Phi_i$ are seasonal AR coefficients, and $\mu$ is the mean of the series.

**Stationarity**

ARIMA time-series forecasting assumes that the time series mean, variance, and autocorrelation are stationary over time. This characteristic is called stationarity. If a time series statistic has nonstationarity, it must be adjusted:

- **Nonstationarity in the mean**—In this case, the mean is not constant but drifts slowly. This can be true for both seasonal and non-seasonal series and is removed by differencing the series. The automatic ARIMA implementation of Predictor determines the amount of non-seasonal differencing required to make a series stationary by using repeated KPSS (Kwiatkowski-Phillips-Schmidt-Shin) tests with appropriate alpha values. For seasonal series, repeated Canova-Hansen tests with appropriate alpha values are used.

- **Nonstationarity in variance**—In this case, the time series is heteroscedastic; the variance of the data around the mean changes over time. This nonstationarity in variance is removed by applying the Box-Cox transformation, a special type of power transformation:

$$z_t = \frac{y_t^\lambda - 1}{\lambda}, \text{ if } \lambda \neq 0$$

$$= \ln(y_t), \text{ if } \lambda = 0$$

Where the original series is $\{x_t\}$, the transformed series is $\{z_t\}$, and the power transformation constant is lambda ($\lambda$).

Predictor determines a suitable value of lambda with an algorithm that uses the seasonality information to divide the dataset into groups, and then tries to find a lambda value that renders the variance stationary across groups.

For users who want more control over the Box-Cox transformation, Predictor provides commonly used power-transformation options, such as log transformation (lambda = 0) or square-root transformation (lambda = 0.5), and even a custom transformation with a user-selected lambda between $-5$ and $+5$ (inclusive). However, Predictor prevents the use of custom lambda values that would result in transformed values being too large or too small.

**Regression Methods**

Predictor supports two types of multiple linear regression, standard and stepwise (forward and iterative). Some rules:

- Only standard forecasting is used for independent variables.
- Lags can be specified for each independent variable. They must be less than the effective length of the series (not including pre-data gaps).
The number of historical data points must be greater than or equal to the number of independent variables, counting the included constant.

For details:

- “Calculating Standard Regression” on page 161
- “Calculating Stepwise Regression” on page 162

**Calculating Standard Regression**

Standard regression can be calculated with or without a constant:

- “Standard Regression with a Constant” on page 161
- “Standard Regression without a Constant” on page 162

**Standard Regression with a Constant**

The regression equation with constant is

$$y_i = b_0 + b_1 x_{1,i} + b_2 x_{2,i} + b_3 x_{3,i} + \ldots + b_m x_{m,i} + \varepsilon$$

This can be written in matrix format as $Y = bX + \varepsilon$,

where $Y$ and $b$ are column vectors of dimension $n$ by 1 and $X$ is a matrix of the dimension $n$ by $(m+1)$, where $n$ is the number of observations and $m$ is the number of independent variables. The first column of $X$ is 1, to include the regression constant. It is assumed that $n > m$.

Predictor uses singular value decomposition (SVD) to determine the coefficients of a regression equation. The primary difference between the singular value decomposition and the least squares techniques is that the singular value decomposition technique can handle situations where the equations used to determine the coefficients of the regression equation are singular or close to singular, which happens when performing regression on equations that represent parallel lines or surfaces. In these cases, the least squares technique returns no solution for the singular case and extremely large parameters for the close-to-singular case.

Crystal Ball uses the matrix technique for singular value decomposition. Starting with:

$$y = bX$$

Following SVD, $X$ can be rewritten:

$$X = [U][w][V]$$

where $U$, $w$, and $V$ are the factor matrices. The matrix $w$, a square matrix of dimension $(m+1)$ by $(m+1)$, is a diagonal matrix with the singular values (or eigenvalues). $U$ and $V$ are other factor matrices.

The coefficients can then be calculated. For example, the $b$ matrix is $b = [V][w]^{-1}[U^T][y]$

The fit vector ($\hat{Y}$) is then calculated as $\hat{Y} = bX$

For related regression statistics, see “Statistics, Standard Regression with Constant” on page 162.
Standard Regression without a Constant

This case is also known as regression through origin.

The regression equation without constant is

\[ y_i = b_1 x_{1,i} + b_2 x_{2,i} + b_3 x_{3,i} + \ldots + b_m x_{m,i} + \epsilon \]

This can be written in matrix format as \( Y = bX + \epsilon \), where \( Y \) and \( b \) are column vectors of dimension \( n \) by \( 1 \) and \( X \) is a matrix of the dimension \( n \) by \( m \), where \( n \) is the number of observations and \( m \) is the number of independent variables. It is assumed that \( n > m \).

Here, too, Predictor uses singular value decomposition (SVD) to determine \( b \), the coefficients of the regression equation. The only difference between this case and regression with a constant is the dimension of the matrices.

For related regression statistics, see “Statistics, Standard Regression without Constant” on page 165.

Calculating Stepwise Regression

Stepwise regression is described in Appendix C of the Crystal Ball Predictor User’s Guide. Note that the partial F statistic is only used in calculating stepwise regression. For a discussion, see “Partial F Statistic, Stepwise Regression” on page 167.

Regression Statistic Formulas

The statistics used to analyze a regression are different from those used to analyze a time-series forecast. Regression statistics:

- “Statistics, Standard Regression with Constant” on page 162
- “Statistics, Standard Regression without Constant” on page 165
- “Partial F Statistic, Stepwise Regression” on page 167

Statistics, Standard Regression with Constant

These statistics describe a standard regression including the constant:

- “ANOVA, Standard Regression with Constant” on page 163
- “\( R^2 \), Regression with Constant” on page 163
- “Adjusted \( R^2 \), Regression with Constant” on page 163
- “SSE, Regression with Constant” on page 164
- “\( F \), Regression with Constant” on page 164
- “Statistics for Individual Coefficients” on page 164
ANOVA, Standard Regression with Constant

ANOVA (analysis of variance) statistics for standard regression with a constant:

Table 5  ANOVA Statistics, Standard Regression with a Constant

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>Degrees of Freedom</th>
<th>Mean Square</th>
<th>F Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>$SSR = \sum (y_i - \hat{y}_i)^2$</td>
<td>$m$</td>
<td>$MSR = SSR/m$</td>
<td>$F = \frac{MSR}{MSE}$</td>
</tr>
<tr>
<td>Error</td>
<td>$SSE = \sum (y_i - \hat{y}_i)^2$</td>
<td>$n - m - 1$</td>
<td>$MSE = SSE/(n - m - 1)$</td>
<td>$F = \frac{MSR}{MSE}$</td>
</tr>
<tr>
<td>Total</td>
<td>$SST = \sum (y_i - \bar{y})^2$</td>
<td>$n - 1$</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

The $F$ statistic follows an $F$ distribution with $(m, n - m - 1)$ degrees of freedom. This information is used to calculate the $p$-value of the $F$ statistic.

$R^2$, Regression with Constant

$R^2$ is the coefficient of determination. This statistic represents the proportion of error for which the regression accounts.

You can use many methods to calculate $R^2$. Predictor uses the equation:

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

Adjusted $R^2$, Regression with Constant

You can calculate a regression equation by using the same number of data points as you have equation coefficients. However, the regression equation will not be as universal as a regression equation calculated using three times the number of data points as equation coefficients.

To correct the $R^2$ for such situations, an adjusted $R^2$ takes into account the degrees of freedom of an equation. When you suspect that an $R^2$ is higher than it should be, calculate the $R^2$ and adjusted $R^2$. If the $R^2$ and the adjusted $R^2$ are close, then the $R^2$ is probably accurate. If $R^2$ is much higher than the adjusted $R^2$, you probably do not have enough data points to calculate the regression accurately.

The formula for adjusted $R^2$:

$$\text{Adjusted } R^2 = 1 - (1 - R^2) \cdot \frac{n - 1}{n - m - 1}$$

where $n$ is the number of data points and $m$ is the number of independent variables.
**SSE, Regression with Constant**

SSE (standard error of measurement) is a measure of the amount the actual values differ from the fitted values. The formula for SSE:

\[
SSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n - m - 1}
\]

where \( n \) is the number of data points you have and \( m \) is the number of independent variables.

**F, Regression with Constant**

The \( F \) statistic checks the significance of the relationship between the dependent variable and the particular combination of independent variables in the regression equation. The \( F \) statistic is based on the scale of the \( Y \) values, so analyze this statistic in combination with the \( p \)-value (described in the next section). When comparing the \( F \) statistics for similar sets of data with the same scale, the higher \( F \) statistic is better.

The formula for the \( F \) statistic is given in Table 5 on page 163.

**Statistics for Individual Coefficients**

Following are the statistics for the \( p^{th} \) coefficient, including the regression constant:

- “Coefficient” on page 164
- “Standard Error of Coefficient” on page 164
- “t” on page 164
- “p” on page 165

**Coefficient**

The coefficient of interest is expressed as \( b_p \), the \( p^{th} \) component in the \( b \) vector.

**Standard Error of Coefficient**

The standard error of this coefficient is expressed as \( se(b_p) \), or

\[
S\sqrt{c_{pp}}
\]

where \( S \) is the standard error of estimate (SSE) and \( c_{pp} \) is the diagonal element at \((p,p)\) of the matrix \((X^TX)^{-1}\).

**t**

If the \( F \) statistic in ANOVA and the corresponding \( p \) indicate a significant relationship between the dependent and the independent variables as a whole, you then want to see the significance of the relationship of the dependent variable to each independent variable. The \( t \) statistic tests for the significance of the specified independent variable in the presence of the other independent variables.

164  Predictor Examples and Reference
The formula for the $t$ statistic:

$$t = \frac{b_p}{se(b_p)}$$

where $b_p$ is the coefficient to check and $se(b_p)$ is the standard error of the coefficient.

The $t$ statistic ("t" on page 164) follows a $t$ distribution with $(n - m - 1)$ degrees of freedom.

**Statistics, Standard Regression without Constant**

These statistics describe a standard regression including the constant:

- “ANOVA, No Constant” on page 165
- “R^2, No Constant” on page 165
- “Adjusted R^2, No Constant” on page 166
- “SSE, No Constant” on page 166
- “F, No Constant” on page 166
- “Statistics for Individual Coefficients, No Constant” on page 166

**ANOVA, No Constant**

ANOVA (analysis of variance) statistics for standard regression without a constant:

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>Degrees of Freedom</th>
<th>Mean Square</th>
<th>F Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>$SSR = \sum\hat{y}_i^2$</td>
<td>$m$</td>
<td>$MSR = SSR/m$</td>
<td>$F = \frac{MSR}{MSE}$</td>
</tr>
<tr>
<td>Error</td>
<td>$SSE = \sum(y_i - \hat{y}_i)^2$</td>
<td>$n - m$</td>
<td>$MSE = SSE/(n - m - 1)$</td>
<td>$F = \frac{MSR}{MSE}$</td>
</tr>
<tr>
<td>Total</td>
<td>$SST = \sum y_i^2$</td>
<td>$n$</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

The $F$ statistic follows an $F$ distribution with $(m, n - m)$ degrees of freedom. This information is used to calculate the $p$-value of the $F$ statistic.

**R^2, No Constant**

$R^2$ is the coefficient of determination. This statistic represents the proportion of error for which the regression accounts.

You can use many methods to calculate $R^2$. Predictor uses the equation:
\[ R^2 = \frac{SSR}{SST} = \frac{\sum y_i^2}{\sum y_i^2} \]

\( R^2 \) can be extremely large in cases when the regression constant is omitted, even when the correlation between \( Y \) and \( X \) is weak. Because it can be meaningless, many applications do not mention this statistic. Predictor provides this statistic but it is not used for stepwise regression when there is no regression constant.

**Adjusted \( R^2 \), No Constant**

Adjusted \( R^2 \) can be calculated for regression without a constant:

\[
\text{Adjusted } R^2 = \frac{SSR}{SST} = \frac{\sum y_i^2}{\sum y_i^2}
\]

where \( n \) is the number of data points and \( m \) is the number of independent variables.

Like \( R^2 \) for regression without a constant, this is also a very large number without much meaning.

**SSE, No Constant**

SSE (standard error of measurement) is a measure of the amount the actual values differ from the fitted values. The formula for SSE:

\[
SSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n - m}
\]

where \( n \) is the number of data points you have and \( m \) is the number of independent variables.

**F, No Constant**

The \( F \) statistic checks the significance of the relationship between the dependent variable and the particular combination of independent variables in the regression equation. The \( F \) statistic is based on the scale of the \( Y \) values, so analyze this statistic in combination with the \( p \)-value (described in the next section). When comparing the \( F \) statistics for similar sets of data with the same scale, the higher \( F \) statistic is better.

The formula for the \( F \) statistic is given in Table 5 on page 163.

**Statistics for Individual Coefficients, No Constant**

The statistics for the \( p^{th} \) coefficient for regressions without a constant are the same as those for regressions with a constant. See “Statistics for Individual Coefficients” on page 164.
Partial F Statistic, Stepwise Regression

Stepwise regression is discussed in Appendix C of the *Crystal Ball Predictor User’s Guide*. Information about the partial F statistic, not discussed elsewhere, follows:

Predictor uses the $p$-value of the partial F statistic to determine if a stepwise regression needs to be stopped after an iteration. ANOVA (analysis of variance) statistics for standard regression with a constant:

For addition of a variable, the partial F statistic for step $t$, ($PF_t$):

$$ PF_t = \frac{SSE_{t-1} - SSE_t}{MSE_{t-1}} $$

$PF_t$ follows the F distribution with degrees of freedom equal to (1, Error DF at step $t$). Users provide a maximum $p$-value, below which the variable is added to the regression.

For deletion of a variable, the partial F statistic for step $t$, ($PF_t$):

$$ PF_t = \frac{SSE_t - SSE_{t-1}}{MSE_{t-1}} $$

$PF_t$ follows the F distribution with degrees of freedom equal to (1, Error DF at step $t$). Users provide a maximum $p$-value, above which the variable is removed from the regression.
For those with Crystal Ball Decision Optimizer, this section presents a variety of examples using OptQuest. These examples illustrate how to use spreadsheets to model optimization problems, the key features of OptQuest, and the variety of applications for which you can use OptQuest. To open example models, see “Opening Example Models” on page 170. Also see “Recommended Run Preference Settings for Optimizations” on page 171.

**Table 7**, following, summarizes the examples in this chapter and the features illustrated.

### Table 7  OptQuest Examples

<table>
<thead>
<tr>
<th>Application</th>
<th>Decision Variables</th>
<th>Type</th>
<th>Constraints</th>
<th>Requirements</th>
<th>Illustrated Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product mix</td>
<td>5</td>
<td>discrete</td>
<td>3</td>
<td>1</td>
<td>Classic optimization example.</td>
</tr>
</tbody>
</table>
### Illustrative Methods

<table>
<thead>
<tr>
<th>Application</th>
<th>Decision Variables</th>
<th>Type</th>
<th>Constraints</th>
<th>Requirements</th>
<th>Illustrated Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotel design and pricing</td>
<td>3</td>
<td>discrete</td>
<td>0</td>
<td>1</td>
<td>Uses a percentile requirement; shows the risk of using a deterministic solution instead of a probabilistic one.</td>
</tr>
<tr>
<td>Budget-constrained project selection</td>
<td>8</td>
<td>binary (0-1)</td>
<td>1</td>
<td>0</td>
<td>Uses binary decision variables for Yes/No decisions.</td>
</tr>
<tr>
<td>Groundwater cleanup</td>
<td>2</td>
<td>mixed</td>
<td>0</td>
<td>1</td>
<td>Uses a category decision variable to select different sets of assumptions.</td>
</tr>
<tr>
<td>Oil field development</td>
<td>3</td>
<td>mixed</td>
<td>0</td>
<td>0</td>
<td>Uses a percentile objective and a lookup table based on a decision variable.</td>
</tr>
<tr>
<td>Portfolio revisited</td>
<td>4</td>
<td>discrete, step = $100</td>
<td>1</td>
<td>1</td>
<td>Combines several objective functions into one multiobjective using extracted statistics and uses the Arbitrage Pricing Theory for incorporating risk. Example of Efficient Frontier.</td>
</tr>
<tr>
<td>Tolerance analysis</td>
<td>7</td>
<td>continuous</td>
<td>0</td>
<td>2</td>
<td>Uses process capability metrics.</td>
</tr>
<tr>
<td>Inventory system</td>
<td>2</td>
<td>discrete</td>
<td>0</td>
<td>0</td>
<td>Searches a wide solution space with large steps, and then refines the search.</td>
</tr>
<tr>
<td>Drill bit replacement</td>
<td>1</td>
<td>continuous</td>
<td>0</td>
<td>0</td>
<td>Defines time as a decision variable.</td>
</tr>
<tr>
<td>Gasoline supply chain</td>
<td>8</td>
<td>discrete</td>
<td>2</td>
<td>1</td>
<td>Classic optimization example.</td>
</tr>
</tbody>
</table>

**Note:** Most of the examples included here use one of the Advanced Options settings for automatically stopping the optimization when either a solution confidence level or certain number of non-improving solutions is reached. If you follow along with these examples, your results should be similar but may not always be identical.

### Opening Example Models

Each section includes a problem statement, a description and explanation of the spreadsheet model, the OptQuest solution, and optionally additional practice exercises using the model. All Microsoft Excel model files and associated OptQuest files are in the Examples folder under the main Crystal Ball installation folder. You can also display an index to the examples by selecting one of the following command sequences and selecting from the Examples Guide:

- **Help**, then **Crystal Ball**, and then **Examples Guide** (Microsoft Excel 2003 or earlier)
- **Help**, then **Resources**, and then **Examples Guide** (Microsoft Excel 2007 or later)
- **Start**, then **All Programs**, then **Oracle Crystal Ball**, and then **Examples** (Windows Start menu)
Recommended Run Preference Settings for Optimizations

To set Crystal Ball run preferences, select Run and then Run Preferences. For optimization purposes, you should usually use the following Crystal Ball settings:

- **Trials** tab — Maximum number of trials to run set to 1000. Central-tendency statistics such as mean, median, and mode usually stabilize sufficiently at 500 to 1000 trials per simulation. Tail-end percentiles and maximum and minimum range values generally require at least 2000 trials.

- **Sampling** tab — Sampling method set to Latin Hypercube. Latin Hypercube sampling increases the quality of the solutions, especially the accuracy of the mean statistic.

- **Sampling** tab — Random Number Generation set to Use Same Sequence Of Random Numbers with an Initial Seed Value of 999. The initial seed value determines the first number in the sequence of random numbers generated for the assumption cells. This enables you to repeat simulations using the same set of random numbers to accurately compare the simulation results. If you do not set an initial seed value, OptQuest will automatically pick a random seed and use that starting seed for each simulation that is run.

  **Note:** When a Crystal Ball forecast has extreme outliers, run the optimization with several different seed values to test the solution’s stability.

- **Speed** tab — Run in Extreme Speed if possible.

After you define the assumptions, decision variables, and forecasts in Crystal Ball, you can begin the optimization process in OptQuest.

Product Mix

Subtopics

- Product Mix Problem Statement
- Product Mix Spreadsheet Model
- Product Mix OptQuest Solution

The listed sections describe this problem and its OptQuest solution.

Product Mix Problem Statement

Ray's Red Hots, Inc. manufactures five types of sausages. The number of pounds of four ingredients—veal, pork, beef, and casing—used per unit of product and the profit generated per unit are given in the table below.

Table 8  Ray's Red Hots Data Summary

<table>
<thead>
<tr>
<th>Products</th>
<th>Veal</th>
<th>Pork</th>
<th>Beef</th>
<th>Casing</th>
<th>Profit Per Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summer Sausage</td>
<td>0.00</td>
<td>2.50</td>
<td>1.00</td>
<td>1.00</td>
<td>$1.25</td>
</tr>
<tr>
<td>Products</td>
<td>Veal</td>
<td>Pork</td>
<td>Beef</td>
<td>Casing</td>
<td>Profit Per Unit</td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Bratwurst</td>
<td>4.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.50</td>
<td>$1.80</td>
</tr>
<tr>
<td>Italian Sausage</td>
<td>1.00</td>
<td>3.00</td>
<td>1.50</td>
<td>1.00</td>
<td>$1.40</td>
</tr>
<tr>
<td>Pepperoni</td>
<td>0.00</td>
<td>4.00</td>
<td>0.00</td>
<td>2.00</td>
<td>$2.10</td>
</tr>
<tr>
<td>Polish Sausage</td>
<td>0.00</td>
<td>1.00</td>
<td>3.00</td>
<td>1.50</td>
<td>$1.70</td>
</tr>
</tbody>
</table>

Limited amounts of ingredients are available for the next production cycle. Specifically, only 12,520 pounds of veal, 14,100 pounds of pork, 6,480 pounds of beef, and 10,800 pounds of casing are available.

Complicating this situation is:

- The unit profits are only estimates because all customer contracts have not been finalized.
- The amount of casing used per unit may be more than anticipated because of production losses due to tearing or partial rejections during inspection.

The problem is to determine how many pounds of each product to produce in order to maximize gross profit without running out of meat ingredients or casing during the manufacturing run.

**Product Mix Spreadsheet Model**

The Product Mix.xls file, shown in Figure 74, is a spreadsheet model for this problem. The input data and model outputs are straightforward.

![Product Mix Problem Spreadsheet Model](image)
Product Mix OptQuest Solution

**Note:** Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

- To run the optimization:
  1. **With Product Mix.xls open in Crystal Ball Decision Optimizer, set the number of trials in Crystal Ball to 2000, since tail-end percentile requirements need more accuracy.**
  2. **Start OptQuest from the Crystal Ball Run menu and click Next to view each wizard panel:**
     - The objective is to maximize the mean of gross profit.
     - The only requirement ensures that at most a 5% chance exists of exceeding the casing limitation.
     - This problem has five decision variables (one for each product), and three constraints (one each for availability of veal, pork, and beef).
  3. **On the Options panel, click Advanced Options and then select Automatically stop after 500 non-improving solutions.**
  4. **Run the optimization.**

**Figure 75** shows the OptQuest solution. The optimal mean profit is $11,695.62, obtained by producing 2800 pounds of bratwurst, 1310 pounds of Italian sausage, 1280 pounds of pepperoni, 1000 pounds of Polish sausage, and 500 pounds of summer sausage.
Figure 75  Product Mix Model Optimization Results

Figure 76 shows the Casing Remaining forecast chart for these decision variables, verifying that the chance of running out of casing is indeed at most 5%.

Figure 76  Product Mix – Remaining Casing Forecast Chart
Hotel Design and Pricing Problem

Subtopics

- Hotel Design Problem Statement
- Hotel Design Spreadsheet Model
- Hotel Design OptQuest Solution

A downtown hotel is considering a major remodeling effort and needs to determine the best combination of rates and room sizes to maximize revenues.

Hotel Design Problem Statement

Currently the hotel has 450 rooms with the following history:

Table 9  Hotel Example Data Summary

<table>
<thead>
<tr>
<th>Room Type</th>
<th>Rate</th>
<th>Daily Avg. No. Sold</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>$85</td>
<td>250</td>
<td>$21,250</td>
</tr>
<tr>
<td>Gold</td>
<td>$98</td>
<td>100</td>
<td>$9,800</td>
</tr>
<tr>
<td>Platinum</td>
<td>$139</td>
<td>50</td>
<td>$6,950</td>
</tr>
</tbody>
</table>

Each market segment has its own price/demand elasticity. Estimates are:

<table>
<thead>
<tr>
<th>Room Type</th>
<th>Elasticity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>-3</td>
</tr>
<tr>
<td>Gold</td>
<td>-1</td>
</tr>
<tr>
<td>Platinum</td>
<td>-2</td>
</tr>
</tbody>
</table>

This means, for example, that a 1% decrease in the price of a standard room will increase the number of rooms sold by 3%. Similarly, a 1% increase in the price will decrease the number of rooms sold by 3%. For any proposed set of prices, the projected number of rooms of a given type sold can be found using the formula:

\[
\text{rooms sold} = H + \frac{E \cdot H \cdot (N - C)}{C}
\]

where variables are:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Historical average number of rooms sold</td>
</tr>
<tr>
<td>$E$</td>
<td>Elasticity</td>
</tr>
<tr>
<td>$N$</td>
<td>New price</td>
</tr>
</tbody>
</table>
The hotel owners want to keep the price of a standard room between $70 and $90, a gold room between $90 and $110, and a platinum room between $120 and $149. All prices are in whole dollar increments (discrete). Although the rooms may be renovated and reconfigured, there are no plans to expand beyond the current 450-room capacity.

**Hotel Design Spreadsheet Model**

To follow this example, open the Hotel Design example shown in Figure 77.

![Hotel Design spreadsheet model](image)

The decision variables correspond to cells G7 through G9.

**Hotel Design OptQuest Solution**

*Note:* Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

1. With Hotel Design.xls open in Crystal Ball:

   As you click Next to step through the problem, notice:
   - The objective is to maximize the mean of total revenue.
   - To ensure that the probability of demand exceeding capacity does not exceed 20%, the projected number of rooms sold (cell H12) is a forecast in the Crystal Ball model, with a requirement added in the Objectives panel. Specifically, the total room demand is
limited by a requirement using the forecast statistic Percentile (80), with an upper bound of 450.

- This problem has three decision variables and no constraints.

2 On the Options panel, click Advanced Options and select Automatically stop after 500 non-improving solutions.

3 Run the optimization.

The results are displayed in Figure 78. The mean of total revenue is $40,406.61 and room prices are $110 for Gold, $133 for Platinum, and $80.00 for Standard.

Figure 78 Hotel Pricing Model Optimization Results

The Crystal Ball simulation of this solution in verifies that the chance of demand exceeding capacity is just slightly less than 20% (100% – 80.81%).
Budget-constrained Project Selection

Subtopics

- Project Selection Problem Statement
- Project Selection Spreadsheet Model
- Project Selection OptQuest Solution

This example concerns project selection for maximum profitability.

Project Selection Problem Statement

The R&D group of a major public utility has identified eight possible projects. A net present value analysis has computed:

- The expected revenue for each if it is successful
- The estimated probability of success
- The initial investment required for each project

Using these figures, the finance manager has computed the expected return and the expected profit for each project as shown in the following table.

<table>
<thead>
<tr>
<th>Project</th>
<th>Expected Revenue</th>
<th>Success Rate</th>
<th>Expected Return</th>
<th>Initial Investment</th>
<th>Expected Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$750,000</td>
<td>90%</td>
<td>$675,000</td>
<td>$250,000</td>
<td>$425,000</td>
</tr>
<tr>
<td>2</td>
<td>$1,500,000</td>
<td>70%</td>
<td>$1,050,000</td>
<td>$650,000</td>
<td>$400,000</td>
</tr>
<tr>
<td>3</td>
<td>$600,000</td>
<td>60%</td>
<td>$360,000</td>
<td>$250,000</td>
<td>$110,000</td>
</tr>
<tr>
<td>4</td>
<td>$1,800,000</td>
<td>40%</td>
<td>$720,000</td>
<td>$500,000</td>
<td>$220,000</td>
</tr>
<tr>
<td>Project</td>
<td>Expected Revenue</td>
<td>Success Rate</td>
<td>Expected Return</td>
<td>Initial Investment</td>
<td>Expected Profit</td>
</tr>
<tr>
<td>---------</td>
<td>------------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>--------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>5</td>
<td>$1,250,000</td>
<td>80%</td>
<td>$1,000,000</td>
<td>$700,000</td>
<td>$300,000</td>
</tr>
<tr>
<td>6</td>
<td>$150,000</td>
<td>60%</td>
<td>$90,000</td>
<td>$30,000</td>
<td>$60,000</td>
</tr>
<tr>
<td>7</td>
<td>$900,000</td>
<td>70%</td>
<td>$630,000</td>
<td>$350,000</td>
<td>$280,000</td>
</tr>
<tr>
<td>8</td>
<td>$250,000</td>
<td>90%</td>
<td>$225,000</td>
<td>$70,000</td>
<td>$155,000</td>
</tr>
</tbody>
</table>

Unfortunately, the available budget is only $2.0 million, and selecting all projects would require a total initial investment of $2.8 million. Thus, the problem is to determine which projects to select to maximize the total expected profit while staying within the budget limitation. Complicating this decision is the fact that both the expected revenue and success rates are highly uncertain.

**Project Selection Spreadsheet Model**

Figure 80 shows a spreadsheet model for this problem, which you can view by opening the Project Selection.xls file. The decision variables in column H are binary; that is, they can assume only the values zero and one, representing the decisions of either not selecting or selecting each project. The total investment in cell F15 is the required investment in column F multiplied by the respective decision variable in column H.

The expected revenue and success rates are assumption cells in the Crystal Ball model. The expected revenues have various distributions, while the success rates are modeled using a binomial distribution with one trial. During the simulation, the outcomes in column D will be either 0% or 100% (not successful or successful) with the probabilities initially specified. Thus,
for each simulated trial, the expected returns will either equal the expected revenue generated in column C or zero. Consequently, the expected profits can be positive or negative.

Although good solutions may be identified by inspection or by trial and error, basing a decision on expected values can be dangerous because it doesn’t assess the risks. In reality, selecting R&D projects is a one-time decision; each project will be either successful or not. If a project is not successful, the company runs the risk of incurring the loss of the initial investment. Thus, incorporating risk analysis within the context of the optimization is a very useful approach.

**Project Selection OptQuest Solution**

- With *Project Selection.xls* open in Crystal Ball, start OptQuest from the Crystal Ball Run menu. Then:
  1. Start the OptQuest wizard.
  2. On the Objectives panel, set the objective to Maximize the Final Value of Total profit. Notice that there are no requirements.
  3. Click Next to step through the problem and notice the following:
     - There are eight decision variables.
     - One constraint represents the budget limitations. Notice the use of Microsoft Excel’s SUMPRODUCT function in the constraint to create a linear combination of the decision variables and investment amounts.
  4. On the Options panel, click Advanced Options and select Automatically stop after 500 non-improving solutions.
  5. Run the optimization.

Figure 81 shows the results of an OptQuest optimization. The best solution identified selects all the projects except for 2, 3, and 7.
Figure 81 Project Selection Model Optimization Results

Figure 82, the forecast chart for Total Profit, shows that the distribution of profits is highly irregular, and depends on the joint success rate of the chosen projects. A risk of realizing a loss exists. You may want to evaluate the risks associated with some of the other solutions identified during the search.

Figure 82 Project Selection Solution Forecast Chart
Groundwater Cleanup

Subtopics

- Groundwater Cleanup Problem Statement
- Groundwater Cleanup Spreadsheet Model
- Groundwater Cleanup OptQuest Solution

This example concerns choosing a method for cleaning up groundwater contamination.

Groundwater Cleanup Problem Statement

A small community gets its water from wells that tap into an old, large aquifer. Recently, an environmental impact study found toxic contamination in the groundwater due to improperly disposed chemicals from a nearby manufacturing plant. Since this is the community’s only source of potable water and the health risk due to exposure to these chemicals is potentially large, the study recommends that the community reduce the overall risk to below a 1 in 10,000 cancer risk with 95% certainty (95th percentile less than 1E-4).

A task force narrowed down the number of appropriate treatment methods to three. It then requested bids from environmental remediation companies to reduce the level of contamination down to recommended standards, using one of these methods.

Your remediation company wants to bid on the project. The costs for the different cleanup methods vary according to the resources and time required for each (cleanup efficiency). With historical and site-specific data available, you want to find the best process and efficiency level that minimizes cost and still meets the study’s recommended standards with a 95% certainty.

Complicating the decision-making process:

- You have estimates of the contamination levels of the various chemicals. Each contaminant’s concentration in the water is measured in micrograms per liter.
- The cancer potency factor (CPF) for each chemical is uncertain. The CPF is the magnitude of the impact the chemical exhibits on humans; the higher the cancer potency factor, the more harmful the chemical is.
- The population risk assessment must account for the variability of body weights and volume of water consumed by the individuals in the community per day.

All these factors lead to the following equation for population risk:

\[
\text{population risk} = \frac{\text{cancer potencies} \cdot \text{contaminant concentrations} \cdot \text{water consumed per day}}{\text{body weight} \cdot \text{conversion factor}}
\]

Groundwater Cleanup Spreadsheet Model

Open the file Groundwater Cleanup.xls (Figure 83).
This model shows the population risk (cell C25), which is the overall contamination risk to the people in the community as a function of the factors shown in Table 11, following:

**Table 11  Groundwater Cleanup Population Risk Factors**

<table>
<thead>
<tr>
<th>Risk factors</th>
<th>Cells</th>
<th>Description</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer Potency</td>
<td>C18:C20</td>
<td>Cancer potency of each contaminant.</td>
<td>Lognormal</td>
</tr>
<tr>
<td>Concentration Before</td>
<td>D18:D20</td>
<td>Concentration of each contaminant before cleanup.</td>
<td>Normal</td>
</tr>
<tr>
<td>Volume Of Water Per Day</td>
<td>C23</td>
<td>Interindividual variability of volume of water consumed each day.</td>
<td>Normal, with lower bound of 0.</td>
</tr>
<tr>
<td>Body Weight</td>
<td>C22</td>
<td>Interindividual variability of body weights in the community.</td>
<td>Normal, with lower bound of 0.</td>
</tr>
</tbody>
</table>

Remediation costs of the various cleanup methods (cells E8:E10) are a function of factors shown in Table 12, following.

**Table 12  Groundwater Cleanup Remediation Cost Factors**

<table>
<thead>
<tr>
<th>Remediation cost factors</th>
<th>Cells</th>
<th>Description</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Costs</td>
<td>C8:C10</td>
<td>Flat costs for each method to pay for initial setup.</td>
<td>Triangular</td>
</tr>
<tr>
<td>Variable Costs</td>
<td>D8:D10</td>
<td>Costs for each method based on how long the cleanup takes.</td>
<td>Uniform</td>
</tr>
<tr>
<td>Efficiency</td>
<td>D14</td>
<td>Percent of contaminants that the cleanup process removes. Each remediation method has a different cost for different efficiency levels.</td>
<td>None</td>
</tr>
</tbody>
</table>
Groundwater Cleanup OptQuest Solution

Note: Except where indicated, this example uses the recommended Crystal Ball run preferences. See "Recommended Run Preference Settings for Optimizations" on page 171.

To run the optimization:

1. Be sure Groundwater Cleanup.xls is open in Crystal Ball Decision Optimizer.
2. Set the number of trials per simulation to 2000, since tail-end percentile requirements need more accuracy.
   As you click Next to step through the problem, notice the following:
   - The objective is to minimize the mean remediation cost while requiring that the population risk be less than or equal to 1E-4 with 95% certainty.
   - There are two decision variables: Remediation Method (cell D13), and Cleanup Efficiency (cell D14). You can select Show cell locations to confirm decision variable cells. Notice that the Category type was chosen for Remediation Method since it acts as an “index” variable for selecting one of the methods.
   - This problem has no constraints.
4. On the Options panel, click Advanced Options and select Automatically stop after 500 non-improving solutions.
5. Run the optimization.

The results are shown in Figure 84, following. The solution in minimizes costs at $10,909 while keeping the risk level at 9.99E-5, rounded.
The distributions for the total remediation cost and the population risk are shown in and Figure 86.
Oil Field Development

Subtopics

- Oil Field Development Problem Statement
- Oil Field Development Spreadsheet Model
- Oil Field Development OptQuest Solution

This example concerns an oil company analysis of Net Present Value for a new asset.

Oil Field Development Problem Statement

Oil companies need to assess new fields or prospects where very little hard data exists. Based on seismic data, analysts can estimate the probability distribution of the reserve size. With little actual data available, the discovery team wants to quantify and optimize the Net Present Value (NPV) of this asset. You can simplify this analysis by representing the production profile by three phases, shown in Table 13.

Table 13  Oil Production Phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build up</td>
<td>The period when you drill wells to gain enough production to fill the facilities.</td>
</tr>
<tr>
<td>Plateau</td>
<td>After reaching the wanted production rate (plateau), the period when you continue production at that rate as long as the reservoir pressure is constant and until you produce a certain fraction of the reserves. In the early stages of development, you can only estimate this fraction, and production greater than a certain rate influences plateau duration.</td>
</tr>
<tr>
<td>Decline</td>
<td>The period when production rates, $P$, decline by the same proportion in each time step, leading to an exponential function: $P(t) = P(0) \exp(-c*t)$ where $t$ is the time since the plateau phase ended and $c$ is some constant.</td>
</tr>
</tbody>
</table>
With only estimates for the total Stock Tank Oil Initially In Place (STOIIP = reserve size) and percent recovery amounts, the objective is to select a production rate, a facility size, and well numbers to maximize some financial measure. In this example the measure used is the 10th percentile (P90) of the NPV distribution. In other words the oil company wants to optimize an NPV value which they are 90% confident of achieving or exceeding.

As described, the problem is neither trivial nor overly complex. A high plateau rate doesn’t lose any reserves, but it does increase costs with extra wells and larger facilities. However, facility costs per unit decrease with a larger throughput, so selecting the largest allowed rate and selecting a facility and number of wells to match may be appropriate.

**Oil Field Development Spreadsheet Model**

Open the Oil Field Development.xls workbook found in the Crystal Ball Example folder (Figure 87).

**Figure 87 Oil Field Development Problem Spreadsheet Model**

Net present value (cell C30) of this oil field is based on:

- Total discounted reserves (cell C27)
- Oil margin (cell C13), which is equivalent to oil price minus operating costs
- Well costs (cell C28)
- Facilities cost (cell C29), which is determined for various production levels by a look-up table
Facility capacity places a maximum limit on production rate, while the production rate of the wells is defined as a normal distribution (cell C7).

The Production Profile table at the bottom of the model shows that the production phase determines annual production rates. Cumulative oil production is calculated per year and is then discounted at 10% (lognormal distribution in cell B10), resulting in a total discounted reserves value. The model gives an oil (or profit) margin of $2.00 per barrel (bbl) and converts total discounted reserves to present value dollars. Total well and facilities costs are then subtracted for total project NPV.

**Oil Field Development OptQuest Solution**

*Note:* Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

➤ Be sure **Oil Field Development.xls** is open in Crystal Ball Decision Optimizer. Then:

1. **Start the OptQuest wizard.**
   - As you click **Next** to step through the problem, notice:
     - The objective is to maximize the 10th percentile (P90) of the NPV.
     - There are no requirements.
     - There are three decision variables: Wells to drill (cell C8), Facility size (cell C12), and Plateau rate (cell C15).
     - This problem has no constraints.

2. On the **Options** panel, click **Advanced Options** and select **Automatically stop after 500 non-improving solutions**.

3. **Run the optimization.**
   - The results are shown in Figure 88. The 10th percentile of NPV is maximized at 195.50 with a facility size of 100, a plateau rate of 10.4, and 14 wells to drill.
The Crystal Ball simulation of this solution in maximizes the 10th percentile (P90) of the NPV with the same result shown in the OptQuest solution window.
Portfolio Revisited

Subtopics

- Portfolio Revisited Problem Statement
- Portfolio Revisited Method 1: Efficient Frontier Optimization
- Portfolio Revisited Method 2: Multiobjective Optimization
- Method 3 — Arbitrage Pricing Theory

This example concerns analysis of an investment portfolio with respect to risk as well as return. The listed sections describe this problem and several ways to solve it using OptQuest.

Portfolio Revisited Problem Statement

The investor from the Portfolio Allocation example model (Tutorial 2 in the Oracle Decision Optimizer OptQuest User’s Guide) has $100,000 to invest in four assets. Below is a relisting of the investor’s expected annual returns, and the minimum and maximum amounts the investor is comfortable allocating to each investment.

<table>
<thead>
<tr>
<th>Investment</th>
<th>Annual Return</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Money market fund</td>
<td>3%</td>
<td>$0</td>
<td>$50,000</td>
</tr>
<tr>
<td>Income fund</td>
<td>5%</td>
<td>$10,000</td>
<td>$25,000</td>
</tr>
<tr>
<td>Growth and income fund</td>
<td>7%</td>
<td>$0</td>
<td>$80,000</td>
</tr>
<tr>
<td>Aggressive growth fund</td>
<td>11%</td>
<td>$10,000</td>
<td>$100,000</td>
</tr>
</tbody>
</table>

When the investor maximized the portfolio return without regard to risk, OptQuest allocated almost all the money to the investment with the highest return. This strategy did not result in a portfolio that maintained risk at a manageable level. Only limiting the standard deviation of the total expected return generated a more diversified portfolio. For more information about this result, see the Efficient Frontier discussion in the Oracle Decision Optimizer OptQuest User’s Guide.

Portfolio Revisited Method 1: Efficient Frontier Optimization

Subtopics

- Efficient Frontier Spreadsheet Model
- Efficient Frontier OptQuest Solution

OptQuest has a feature that creates an efficient frontier for you automatically. To use the Efficient Frontier function in OptQuest, you need only define a requirement with a variable upper or lower bound. OptQuest then calculates points within the variable requirement range.

The listed sections describe the model for this solution method and its OptQuest solution.
Efficient Frontier Spreadsheet Model

Open the Portfolio Revisited EF.xls workbook found in the Crystal Ball Examples folder. The total expected return forecast, assumptions, and decision variables are the same as in the original model, with the decision variables already defined.

Efficient Frontier OptQuest Solution

Note: Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

Perform these steps:

1. With Portfolio Revisited EF.xls open in Crystal Ball, set the number of trials per simulation to 2000 in the Run Preferences dialog.

2. Start OptQuest from the Crystal Ball Run menu.

As you click Next to step through the problem, notice that the objective, decision variables, and constraints are the same as for the original example (Tutorial 2 in the Oracle Decision Optimizer OptQuest User’s Guide).

Figure 90 shows the Objectives panel with the variable requirement needed for efficient frontier testing.

Figure 90  Objectives Panel with a Variable Requirement

The requirement has a variable upper bound for the standard deviation statistic (less than or equal to $8,000).

The variable requirement bounds are $8,000 for the lower bound and $10,000 for the upper bound in steps of $250.
3 Run the optimization for 2000 simulations (set in the Options panel of the OptQuest wizard).

The results are shown in Figure 91. The mean of total expected return is maximized at $8,657 with fund allocations as follows: Aggressive Growth fund = $50,000; Growth and Income fund = $33,100; Income fund = $16,900; and Money Market fund = $0.

![Figure 91 Portfolio Revisited Efficient Frontier Optimization Results]

When should you use the Efficient Frontier function? This method is useful when it is difficult to determine reasonable lower or upper bounds for requirement statistics.

**Portfolio Revisited Method 2: Multiobjective Optimization**

Another technique for finding efficient portfolios is called multiobjective (or multicriteria) optimization. You can use this technique to optimize multiple, often conflicting objectives, such as maximizing returns and minimizing risks, simultaneously. Other examples of multiobjective optimization include:

- Aircraft design, requiring simultaneous optimization of weight, payload capacity, airframe stiffness, and fuel efficiency
- Public health policies, requiring simultaneous minimization of risks to the population, direct taxpayer costs, and indirect business regulation costs
- Electric power generation, requiring simultaneous optimization of operating costs, reliability, and pollution control
Most forms of multiobjective optimization are solved by minimizing or maximizing a weighted combination of the multiple objectives. In the portfolio example, a weighted combination of the return and risk objectives may be:

\[
\text{mean return} \quad (k \times \text{standard deviation})
\]

where \( k > 0 \) is a risk aversion constant, and the objective is to maximize the function. The relationship between return and risk for the investor is captured entirely by this one function; no additional requirements are necessary.

Geometrically, the optimal solution for a multiobjective function occurs in the saddle point between the optimal endpoints of the individual objectives. In the case of the two-objective function described previously, the optimal solution occurs somewhere on the efficient frontier between the maximum-return portfolio and the minimum-risk portfolio.

For \( k = 0.5 \), the optimal solution occurs at the point where the return minus one-half the standard deviation has the highest value.

The following sections describe the model for this problem and its OptQuest solution:

- “Multiobjective Optimization Spreadsheet Model” on page 193
- “Multiobjective Optimization OptQuest Solution” on page 194

.multiobjective Optimization Spreadsheet Model

Open the Portfolio Revisited.xls workbook found in the Crystal Ball Examples folder. The total expected return forecast, assumptions, and decision variables are the same as in the original model. Scroll down to see the new items added as shown in Figure 92.

Figure 92 Portfolio Revisited Spreadsheet Model

This new function (cell C22) contains the multiobjective relationship described by mean return \( - (k \times \text{standard deviation}) \) with the risk aversion constant (cell C19) broken out into a separate
cell. The mean return and standard deviation variables in this equation are automatically extracted at the end of the simulation from the Total Expected Return forecast (cell C17). See the Oracle Crystal Ball User’s Guide for more information on the Auto Extract feature.

**Multiobjective Optimization OptQuest Solution**

To follow this example:

1. Open **Portfolio Revisited.xls** in Crystal Ball Decision Optimizer.

   **Note:** Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

2. Start the OptQuest wizard.

   As you click **Next** to step through the problem, notice:

   - The objective refers to the new multiobjective function: **Maximize the Final Value of Mean minus stddev**. The statistic to optimize is **Final Value**, to calculate only the statistical values for the total expected return forecast at the end of the simulation. There are no requirements.

   - The decision variables and constraints are the same as previous Portfolio Allocation examples.

3. On the **Options** panel, select **Run for 2000 simulations**, and then click **Advanced Options** and select **Automatically stop after 500 non-improving solutions**.

4. Run the optimization for 2000 simulations.

   The results are displayed in **Figure 93**. Mean minus std dev is maximized at $3052.33 and fund values are as follows:

   - Aggressive Growth = $16,400
   - Growth and Income = $8,600
   - Income = $25,000
   - Money Market = $50,000

   After reviewing the results, close Portfolio Revisited.xls without saving it.
Tip: Multiobjective optimization is especially useful when it is difficult to determine reasonable lower or upper bounds for requirement statistics. This method is also recommended for situations where OptQuest has trouble finding feasible solutions that satisfy many requirements. Using a single objective with requirements is generally easier to implement and understand.

Method 3 — Arbitrage Pricing Theory

Subtopics

- Spreadsheet Model
- OptQuest Solution

A different approach to incorporating risk in a decision model is called Arbitrage Pricing Theory (APT). APT does not ask whether portfolios are efficient. Instead, it assumes that a stock or mutual fund’s return is based partly on macroeconomic influences and partly on events unique to the underlying company or assets (see Brealey and Myers, 1991, listed in the Bibliography). Further, this theory only considers macroeconomic influences, since diversification, as in a portfolio, practically eliminates unique risk. Some macroeconomic influences might include:

- The level of industrial activity
- The spread between short- and long-term interest rates
The spread between low- and high-risk bond yields (see Chen et al. listed in the Bibliography)

A weighted sum of these influences determines the risk factor of an asset. APT provides estimates of the risk factors for particular assets to these types of influences. Higher risk factors indicate greater risk; lower factors indicate less risk. Assume that the risk factors per dollar allocated to each asset are as shown in Table 15, “Sample Asset Risk Factors,” on page 196.

<table>
<thead>
<tr>
<th>Investment</th>
<th>Risk Factor per Dollar Invested</th>
</tr>
</thead>
<tbody>
<tr>
<td>Money market fund</td>
<td>-0.3</td>
</tr>
<tr>
<td>Income fund</td>
<td>-0.5</td>
</tr>
<tr>
<td>Growth and income fund</td>
<td>0.4</td>
</tr>
<tr>
<td>Aggressive growth fund</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Using this method, the investor can specify a target level for the weighted (or aggregate) risk factors, leading to a constraint that limits the overall risk. For example, suppose that the investor can tolerate a weighted risk per dollar invested of at most 1.0. Anything above 1.0 is too risky for the investor. Thus, the weighted risk for a $100,000 total investment must be at or below 100,000. If the investor distributed $100,000 equally among the four available assets, the return would be:

$$0.03($25,000) + 0.05($25,000) + 0.07($25,000) + 0.11($25,000) = 7,000$$

And the total weighted risk would be:

$$-0.3($25,000) - 0.5(25,000) + 0.4(25,000) + 2.1(25,000) = 42,500$$

If this amount were greater than the limit of 100,000, this solution would not be feasible and could not be chosen.

**Spreadsheet Model**

Open the Portfolio Revisited.xls worksheet found in the Crystal Ball Examples folder (Figure 92 on page 193). The total expected return forecast, assumptions, decision variables, and the original constraint limiting the total investment to $100,000 are the same as in the original model. The new item is a constraint limiting the total weighted risk (cell F13), calculated by:

$$\text{total weighted risk} = -0.3 \left( \text{money market investment} \right) - 0.5 \left( \text{income fund investment} \right) + 0.4 \left( \text{growth and income fund investment} \right) + 2.1 \left( \text{aggressive growth fund investment} \right)$$

The total weighted risk is limited to be equal to $100,000.
OptQuest Solution

To follow along with this example:

1 Open Portfolio Revisited.xls in Oracle Crystal Ball.

   Note: This example uses run preferences recommended in “Recommended Run Preference Settings for Optimizations” on page 171.

2 Start OptQuest.

3 When the Objectives panel opens, change the objective so it is the same as in Tutorial 2 in the Oracle Decision Optimizer OptQuest User’s Guide.

   The objective should be set to Maximize the Mean of Total Expected Return.

   There are no requirements. Click Next.

4 You do not need to change the decision variables. Click Next.

5 In the Constraints panel, add a new constraint that references cell F13, the total weighted risk. Note that the formula in cell F13 uses the Excel SUMPRODUCT function to create a linear combination of the risk factors and the decision variables. Limit this risk to $100,000 as shown in Figure 94 on page 197.

   To do this:
   - Click the blank row below the first constraint.
   - Click Insert Reference.
   - Point to F13 and click OK.
   - Type =100000 after the cell reference.

6 To document the cell reference, select the second constraint and click Add Comment. In the Add Comment dialog, type Cell F13 = the sum product of risk factors and decision variables. When you click OK, the comment appears above the second constraint.

7 Click Next. In the Options panel, set the optimization to run for 1000 simulations.

8 Run the optimization.

The results appear in Figure 95 on page 198.
The Crystal Ball simulation of this solution maximizes the mean of the total expected return at $8,436 with the new constraint. Compare this to the original total expected return of $7,752 originally calculated using the different method of limiting risk with the standard deviation.
Tolerance Analysis

Subtopics

- Tolerance Analysis Problem Statement
- Tolerance Analysis Spreadsheet Model
- Tolerance Analysis OptQuest Solution

An engineer at an automobile design center needs to specify components for piston and cylinder assemblies that work well together. To do this, he needs the dimensions of the components to be within certain tolerance limits, while still selecting the most cost-efficient methods. This is called an optimal stack tolerance analysis.

Note: This example involves concepts used only by Six Sigma and similar quality programs. If you are not familiar with Crystal Ball’s process capability features, consider reviewing the process capability appendix in the Oracle Crystal Ball User’s Guide.

Tolerance Analysis Problem Statement

The piston assembly consists of five components, and the cylinder assembly consists of two, each with certain nominal dimensions. These components are then stacked to create the assembly. The difference in length between the two, called the assembly gap, must be between 0.003 and 0.02 inches. This may seem like a simple problem, but since milling processes are not exact and quality control has a direct effect on prices, components have an error associated with each, called tolerance. When stacked, these errors compile or add together to create a cumulative tolerance.

When a batch of components is milled and measured, the components’ actual dimensions form a distribution around the wanted, or nominal, dimension. Standard deviation, or sigma, is a measure of the variation present in a batch of components. The components then have a statistical dimension based on this distribution. The quality of the component and the associated tolerance is described in terms of sigmas, with 1 sigma component having the largest tolerance and a 5 sigma component the smallest. This is called the quality specification.

One simplified solution takes the total tolerance allowed and divides it by the number of components. But, due to individual component complexity and process differences in manufacturing, each component of the assembly has a different cost function associated with the quality specification. This then becomes a juggling act to balance cumulative tolerance and associated cost.

Crystal Ball supports quality programs such as Six Sigma by calculating a set of process capability metrics for forecasts when the process capability features are activated and at least one specification limit (LSL or USL) is entered for the forecasts. OptQuest then includes these metrics in the list of statistics that can be optimized. For more information, see “RMSE” on page 135.

This example assumes that the process capability metrics have been activated in Crystal Ball. Then, the capability metrics are available in the Forecast Statistic list of the Objectives panel.
Tolerance Analysis Spreadsheet Model

Open the Tolerance Analysis.xls file (Figure 96).

Figure 96  Tolerance Analysis Spreadsheet Model

A drawing of the assembly is in the corner. In this example:

- The nominal dimensions are in cells C14:C18 and C23:C24.
- Initial tolerances of each 3-sigma component are in cells D14:D18 and D23:D24.
- The statistical dimension (cells H14:H18 and H23:H24) of each component is defined as an assumption with a normal distribution having a mean equal to the nominal dimension and a standard deviation equal to the component sigma. Notice that the mean and standard deviation are cell references to these cells.

The dimensions of the assemblies are a cumulation of their respective components’ statistical dimensions. The difference in length between the cylinder assembly (cell C5) and the piston assembly (cell C4) is the assembly gap (cell C6).

Component cost (cells F14:F18 and F23:F24) is a nonlinear function of quality specification. The higher the specification, the higher the cost. Also notice that each component has a different cost function associated with it.
In addition to the recommended options, before running OptQuest, in Crystal Ball select Run, and then Run Preferences and set:

- The maximum number of trials run to 2000
- The sampling method to Latin Hypercube
- The sample size to 2000 for Latin Hypercube

On the Statistics tab of the Run Preferences dialog, select Calculate capability metrics.

Since the model is heavily dependent on the tails of the forecast distribution, these settings will provide higher accuracy and will be adequate for this example. In actual practice, to gain better accuracy, the engineer may want to run longer simulations of 5000 or 10,000 trials.

Tolerance Analysis OptQuest Solution

The goal of the following solution is to maximize quality while minimizing cost.

Note: Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

➢ To run the optimization:

1. Be sure Tolerance Analysis.xls is open in Crystal Ball Decision Optimizer and the maximum trials and Latin Hypercube sample sizes have been set to 2000 as described previously.
2. Start OptQuest.
3. On the Objectives panel, set the objective to: Maximize the Zst total of Assembly gap.
4. Set the following requirements:
   - The Minimum of Assembly gap must be greater than or equal to 0.0030 in inches
   - The Maximum of Assembly gap must be less than or equal to 0.0200 in inches.
   - The Final Value of Total assembly cost must be less than or equal to $60.00 dollars.
5. Click Next to step through the problem and notice the following:
   - This problem has seven decision variables, one for the quality specification for each assembly component, with a continuous range between 1 and 5 sigmas.
   - The problem has no constraints.
6. On the Options panel, click Advanced Options and select Automatically stop after 500 non-improving solutions.
7. Run the optimization.

The cost and quality solution values are displayed in Figure 97. The assembly gap Zst-total is 3.57 with a total assembly cost of $59.99. Decision variable values range from 3.5 to 3.9 sigmas.
Inventory System Optimization

Subtopics

- Inventory System Problem Statement
- Inventory System Spreadsheet Model
- Inventory System OptQuest Solution

This example is adapted from James R. Evans and David L. Olson, *Introduction to Simulation and Risk Analysis*. New York: Prentice-Hall, 1998.

Inventory System Problem Statement

The two basic inventory decisions that managers face are:

- How much additional inventory to order or produce
- When to order or produce it
Although it is possible to consider these two decisions separately, they are so closely related that a simultaneous solution is usually necessary. Typically, the objective is to minimize total inventory costs. Total inventory costs typically include holding, ordering, shortage, and purchasing costs.

In a continuous review system, managers continuously monitor the inventory position. Whenever the inventory position falls at or below a level R, called the reorder point, the manager orders Q units, called the order quantity. (Notice that the reorder decision is based on the inventory position including orders and not the inventory level. If managers used the inventory level, they would place orders continuously as the inventory level fell below R until they received the order.) When you receive the order after the lead-time, the inventory level jumps from zero to Q, and the cycle repeats.

In inventory systems, demand is usually uncertain, and the lead-time can also vary. To avoid shortages, managers often maintain a safety stock. In such situations, it is not clear what order quantities and reorder points will minimize expected total inventory cost. Simulation models can address this question.

In this example, demand is uncertain and is Poisson distributed with a mean of 100 units per week. Thus, the expected annual demand is 5,200 units.

Note: For large values of the rate parameter, \( \lambda \), the Poisson distribution is approximately normal. Thus, this assumption is tantamount to saying that the demand is normally distributed with a mean of 100 and standard deviation of \( \sqrt{100} = 10 \). The Poisson distribution is discrete, thus eliminating the need to round off normally distributed random variates.

Additional relationships that hold for the inventory system are:

- Each order costs $50 and the holding cost is $0.20 per unit per week ($10.40 for one year).
- Every unfilled demand is lost and costs the firm $100 in lost profit.
- The time between placing an order and receiving the order is 2 weeks. Therefore, the expected demand during lead-time is 200 units. Orders are placed at the end of the week, and received at the beginning of the week.

The traditional economic order quantity (EOQ) model suggests an order quantity:

\[
Q = \sqrt{\frac{2 \times 5200 \times 50}{10.4}} = 224
\]

For the EOQ policy, the reorder point should equal the lead-time demand; that is, place an order when the inventory position falls to 200 units. If the lead-time demand is exactly 200 units, the order will arrive when the inventory level reaches zero.

However, if demand fluctuates about a mean of 200 units, shortages will occur approximately half the time. Because of the high shortage costs, the manager would use either a larger reorder point, a larger order quantity, or both. In either case, the manager will carry more inventory on average, which will result in a lower total shortage cost but a higher total holding cost. A higher order quantity lets the manager order less frequently, thus incurring lower total ordering costs.
However, the appropriate choice is not clear. Simulation can test various reorder point/order quantity policies.

**Inventory System Spreadsheet Model**

Before examining the spreadsheet simulation model, step through the logic of how this inventory system operates. Assume that no orders are outstanding initially and that the initial inventory level is equal to the order quantity, Q. Therefore, the beginning inventory position will be the same as the inventory level. At the beginning of the week, if any outstanding orders have arrived, the manager adds the order quantity to the current inventory level.

Next, determine the weekly demand and check if sufficient inventory is on hand to meet this demand. If not, then the number of lost sales is the demand minus the current inventory. Subtract the current inventory level from the inventory position, set current inventory to zero, and compute the lost sales cost. If sufficient inventory is available, satisfy all demand from stock and reduce both the inventory level and inventory position by the amount of demand.

The next step is to check if the inventory position is at or below the reorder point. If so, place an order for Q units and compute the order cost. The inventory position is increased by Q, but the inventory level remains the same. Schedule a receipt of Q units to arrive after the lead-time.

Finally, compute the holding cost based on the inventory level at the end of the week (after demand is satisfied) and the total cost.

Open the file Inventory System.xls. This spreadsheet model, shown in Figure 98, implements this logic. The basic problem data are shown several rows down from the title. The decision variables are the order quantity (cell E3) and the reorder point (cell E4). The initial inventory is set equal to the chosen order quantity. This example assumes the specified lead-time is constant.

---

**Figure 98  Inventory System Problem Spreadsheet Model**
In the actual simulation, the beginning inventory position and inventory level for each week equals the ending levels for the previous week, except for the first week, which is specified in the problem data. The demand is in column F as Crystal Ball assumptions.

Since all shortages are lost sales, the inventory level cannot be negative. Thus, the ending inventory each week is:

$$\text{ending inventory} = \max\left\{ \text{beginning inventory level} - \text{demand} + \text{orders received}, 0 \right\}$$

Lost sales are computed by checking if demand exceeds available stock and computing the difference.

The spreadsheet simulates 52 weeks, or one year of operation of the inventory system. Since the objective is to minimize the mean total annual cost, cell O6 is defined as a forecast cell.

Column I determines whether the manager should place an order by checking if the beginning inventory position minus the weekly demand is at or below the reorder point. The ending inventory position is:

$$\frac{\text{ending}}{\text{inventory position}} = \frac{\text{beginning}}{\text{inventory position}} - \text{weekly demand} + \text{lost sales} + \text{weekly orders}$$

This formula may not be obvious. Clearly, if there are no lost sales, the ending inventory position is simply the beginning position minus the demand plus any order that may have been placed. If lost sales occur, computing the ending inventory position this way reduces it by the unfulfilled demand, which is incorrect. Thus, you must add back the number of lost sales to account for this.

In the ordering process, the manager places orders at the end of the week and receives orders at the beginning of the week. Thus, in Figure 98, the order placed at the end of the first week with a lead-time of 2 weeks will arrive at the beginning of the fourth week. Column K determines the week an order is due to arrive, and a MATCH function is used in column D to identify whether an order is scheduled to arrive.

**Inventory System OptQuest Solution**

**Note:** Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

- With Inventory System.xls open in Crystal Ball Decision Optimizer:
  - 1 Start the OptQuest wizard.
    - As you click Next to step through the problem, notice:
      - The objective is to minimize the mean total annual costs.
      - There are no constraints or requirements.
      - This problem has two decision variables.
- The initial search limits are set between 200 and 400 for both variables using a Discrete decision variable type with a step of 5.
- This optimization runs more slowly than some. You may want to run fewer than 1,000 simulations or use the Advanced Options settings to automatically stop the optimization when certain criteria are met. This example assumes that the automatic stop setting is selected.

2 Run the optimization.

Figure 99, following, shows optimization results. OptQuest identified the best solution as having an order quantity of 330 and a reorder point of 320. The Performance Chart shows that OptQuest quickly found a good solution value.

Figure 99  Inventory System Model Optimization Results

Because this optimization used a step size of 5, you can fine-tune the solution by searching more closely around the best solution using a smaller step size while also increasing the number of trials per simulation for better precision. This is a good practice, since selecting too small a step size initially consumes a lot of time or, if time is restricted, OptQuest may not find a good solution. Thus, as the number of decision variables and range of search increases, use larger step sizes and fewer trials initially. Later, refine the search around good candidates.

Figure 100 shows the results of an optimization with Order Quantity and Reorder Point bounded to the range 300 to 360, with a step size of 1, and 1000 trials per simulation. OptQuest identified the best solution as Order Quantity = 332 and Reorder Point = 325. There was very little change from the initial solution.
Figure 100  Inventory System—Second Optimization Results

![OptQuest Results](image)

<table>
<thead>
<tr>
<th>Best Solution:</th>
<th>Simulation #: 85</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objectives</strong></td>
<td></td>
</tr>
<tr>
<td>Minimize the Mean of Total Annual Costs</td>
<td>$2,929</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
<td>Left Side</td>
</tr>
<tr>
<td><strong>Decision Variables</strong></td>
<td>Value</td>
</tr>
<tr>
<td>Order Quantity</td>
<td>2,020</td>
</tr>
<tr>
<td>Reorder Point</td>
<td>325</td>
</tr>
</tbody>
</table>

Figure 101 shows the Crystal Ball forecast chart for the annual total costs. You can see that the distribution of total annual cost is highly concentrated around the mean, but is also skewed far to the high-value end, indicating that very high values of cost are possible, although not very likely. For such highly skewed distributions, run more trials than usual, since statistics like the mean and tail-end percentiles can be susceptible to extreme outliers.

Figure 101  Inventory System Final (Best) Solution Forecast Chart

![Forecast: Total Annual Costs](image)
Drill Bit Replacement Policy

Subtopics

- Drill Bit Replacement Problem Statement
- Drill Bit Replacement Spreadsheet Model
- Drill Bit Replacement OptQuest Solution


Drill Bit Replacement Problem Statement

When drilling wells in certain types of terrain, the performance of a drill bit erodes with time because of wear. After $T$ hours, the drilling rate can be expressed as:

$$\frac{dM}{dT} = \frac{15}{\sqrt{T / 10}} \text{ meters per hour}$$

For example, after 5 hours of consecutive use (starting with a new drill bit), the drill is able to penetrate the terrain at a rate of:

$$\frac{15}{\sqrt{5 / 10}} = 21.21 \text{ meters per hour}$$

While after 50 hours, the penetration rate is only:

$$\frac{15}{\sqrt{50 / 10}} = 6.71 \text{ meters per hour}$$

Eventually, the bit must be replaced as the costs exceed the value of the well being drilled. The problem is to determine the optimum replacement policy; that is, the drilling cycle, $T$ hours, between replacements.

$T$ hours after replacing the bit, the total drilled depth in meters, $M$, is given by the integral of Equation 4.2 from 0 to $T$, or:

$$M = 300 \sqrt{T / 10} \text{ meters}$$

where 300 is a drilling depth coefficient.

The revenue value per meter drilled is calculated to be $60. Drilling expenses are fixed at $425 per hour, and it generally requires $R = 7.5$ hours to install a new drill bit, at a cost of $8,000 + $400R$.

If all drilling parameters were certain, calculating the optimal replacement policy would be straightforward. However, several of the drilling parameters are uncertain, and knowledge about their values must be assumed:

- Because of variations in the drilling process and terrain, the depth coefficient, $C$, is characterized by a normal distribution with a mean of 300 and a standard deviation of 20.
The drill bit replacement time, $R$, varies and is determined by a triangular distribution with parameters 6.5, 7.5, and 9.

The number of 10-hour days available per month, $D$, also varies due to the weather and the number of days in a month, and is assumed to be triangular with parameters 24, 28, and 30.

With these assumptions, the profit/drilling cycle if the bit is replaced after $T$ hours equals the revenue obtained from drilling minus drilling expenses and replacement costs:

$$\text{profit/drilling cycle} = 60M - 425T - (8000 + 400R)$$

Assuming $D$ ten-hour days per month, the average number of cycles per month is $10D/(T + R)$. Therefore, the average profit per month is:

$$\frac{\text{average profit}}{\text{month}} = \frac{10D}{T + R} \left[ 60 \left( \frac{T}{10} \right) - 425T - 8000 - 400R \right]$$

The objective is to find the value of $T$ that maximizes the average profit per month.

**Drill Bit Replacement Spreadsheet Model**

Open the Drill Bit Replacement example, shown in Figure 102, below. This workbook has Crystal Ball assumptions defined for:

<table>
<thead>
<tr>
<th>Cell</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>C6</td>
<td>Replacement time, $R$.</td>
</tr>
<tr>
<td>C8</td>
<td>Drilling depth function coefficient, $C$.</td>
</tr>
<tr>
<td>C10</td>
<td>Number of days available per month, $D$.</td>
</tr>
</tbody>
</table>

One decision variable is defined in cell C12: the cycle time between replacements of the drill bit, $T$. 
The model outputs are computed using the formulas developed in the previous section. The drilling expenses in cell F7 include both the drilling costs and the replacement costs. The forecast cell is F12, profit per month.

**Drill Bit Replacement OptQuest Solution**

**Note:** Except where indicated, this example uses the recommended Crystal Ball run preferences. See “Recommended Run Preference Settings for Optimizations” on page 171.

1. With **Drill Bit Replacement.xls** open in Crystal Ball Decision Optimizer:
   - **Start the OptQuest wizard.**
     - As you click **Next** to step through the problem, notice:
       - The objective is to maximize the mean profit per month.
       - The problem has no requirements or constraints.
       - This problem has one decision variable, with search limits of 1 and 50.
   2. **Run the optimization.**

   Figure 103, following, shows the OptQuest results. The best solution is to replace the drill bit approximately every 19.9 hours.
Figure 103  Drill Bit Replacement Model Optimization Results

Figure 104, following, shows the OptQuest forecast chart and statistics for the simulation of this solution. The profit per month has a relatively large standard deviation compared to the mean (coefficient of variation=0.30); thus, it is likely that the true profit/month is significantly higher or lower than the mean objective value.

Figure 104  Drill Bit Replacement Forecast Chart and Statistics
Gasoline Supply Chain

Subtopics

- Gasoline Supply Chain Statement of Problem
- Gasoline Supply Chain Spreadsheet Model
- Gasoline Supply Chain OptQuest Solution

This example shows how to determine the optimum amount of gasoline to transport between different levels of a gasoline supply chain. The objective is to minimize the total cost, which includes transportation costs and inventory holding costs at various points in the supply chain. It is also important to minimize stockouts at various retail outlets. The complexity of the problem arises from the fact that stochastic (variable) production exists at the refinery level and stochastic demand exists at the retail outlet level.

Gasoline Supply Chain Statement of Problem

The supply chain illustrated here is simplified. It consists of one refinery (SP), two supply depots (SD), and three retail outlets (RO).

A weekly snapshot of this supply chain is as follows:

- The refinery produces a variable amount of gasoline every week, which it transports to SDs for cross-docking.
- SDs supply gasoline to ROs, which realize stochastic demand from end customers.
- All three supply chain levels (Refinery, SD, and RO) face inventory holding costs.
- In addition, the RO’s face the risk of stockouts for not fulfilling customer demands.

The problem is to determine the amount of gasoline to transport between each level of the supply chain to minimize the total operating cost, which is computed as the sum of transportation costs and inventory holding costs. For business reasons, it is helpful to minimize stockouts at the ROs, to a certain extent.

The following is a schematic diagram of the supply chain:

Assumptions about the supply chain are as follows:
The weekly supply of gasoline from the refinery (SP) follows a normal distribution with a mean of 2000 gallons and standard deviation (s.d.) of 450 gallons.

The weekly demands at ROs are distributed lognormally with means and standard deviations of 400 gallons and 50 gallons, 500 gallons and 75 gallons, 650 gallons and 100 gallons respectively at RO1, RO2, RO3.

The inventory holding cost is a dollar for every five gallons.

The transportation costs in dollars per gallon are as follows (notice that these costs include transportation distances):

- SP to SD1 = $15
- SP to SD2 = $12.5
- SD1 to RO1 = $6.5
- SD1 to RO2 = $7.5
- SD1 to RO3 = $9.0
- SD2 to RO1 = $9.0
- SD2 to RO2 = $8.0
- SD2 to RO3 = $7.0

Existing inventories in gallons are:

- Refinery: 200 gallons, SD1: 50 gallons, SD2: 100 gallons, RO1: 120 gallons, RO2: 180 gallons, RO3: 80 gallons.

Other assumptions include:

- No capacity limit exists on transportation links and supply chain points.
- An implicit constraint exists that the SDs do not have any stockouts. This mathematically implies that:
  
\[
\text{Existing Inventory + Supply Received - Demand Fulfilled} \geq 0
\]

**Gasoline Supply Chain Spreadsheet Model**

Open the spreadsheet model for this example, Gasoline Supply Chain.xls, as shown in Figure 105.

This model includes:

- Four Crystal Ball assumptions in cells C5, C33, F33, and I33. These represent stochastic output from the refinery and stochastic demand at the retail outlets.
- Two Crystal Ball forecasts in cells K5 and K7 to represent total costs and the worst-case stockout situation.
- Eight decision variables that represent transportation costs from the refinery to the two supply depots and from each depot to each retail outlet. These are displayed in cells E10, H10, C21, C22, C23, I21, I22, and I23.
For this example, OptQuest can determine how much to supply at each of the SDs and ROs to minimize the total expected cost while maintaining stockouts at ROs at an acceptable level.

**Figure 105  Gasoline Supply Chain Spreadsheet Model**

Gasoline Supply Chain OptQuest Solution

**Note:** Except where indicated, this example uses the recommended Crystal Ball run preferences. See "Recommended Run Preference Settings for Optimizations" on page 171.

With **Gasoline Supply Chain.xls** open in Crystal Ball Decision Optimizer:

1. Start the OptQuest wizard.

As you click **Next** to step through the problem, notice:

- The objective is to minimize the mean of total costs.
- The problem has one requirement: the 95th percentile of the worst-case stockout forecast must be less than 0 gallons.
- This problem has eight discrete decision variables, with bounds of 0 to 2000. These represent transportation costs among the various elements of the supply chain.
- The problem has two constraints (Figure 106) that specify that both links of the supply chain (running through SD1 and SD2) must have sufficient inventories of gasoline.
2 Run the optimization.

Figure 107 shows sample OptQuest results.

Figure 107 shows that if the quantities of gasoline shown for each decision variable are transported between the indicated destinations, the mean total cost will be $35,856.46 and the 95th percentile of the worst-case stockout will be -3.92, less than 0.

Figure 108, following, shows the Total Costs forecast chart and statistics for the simulation of this solution. The Total Costs standard deviation ($95.07) is quite small relative to the mean total cost ($35,856.46), suggesting that this cost forecast is an accurate representation of the true weekly costs.
Optimization Tips and Notes

Subtopics

- Model Types
- Factors That Affect Optimization Performance
- Sensitivity Analysis Using a Tornado Chart
- Maintaining Multiple Optimization Settings for a Model
- Other OptQuest Notes

For those with Crystal Ball Decision Optimizer, this section describes the different factors that affect how OptQuest searches for optimal solutions, including model types. Understanding how these factors affect the optimization helps you control the speed and accuracy of the search.

This section also includes discussion of the Crystal Ball Tornado Chart tool and how you can use it to analyze the sensitivity of the variables in your model and screen out minor decision variables.

These tips and suggestions are followed by some notes to help you avoid unexpected results when using OptQuest. They can also help you troubleshoot any difficulties that may occur.
Model Types

Subtopics

- Optimization Models Without Uncertainty
- Optimization Models With Uncertainty
- Discrete, Continuous, or Mixed Models
- Linear or Nonlinear Models

Selecting the right model for your scenario is essential for obtaining optimal results.

Optimization Models Without Uncertainty

Conceptually, an optimization model may look like Figure 109 on page 217.

Figure 109  Schematic of an Optimization Model Without Uncertainty

The solution to an optimization model provides a set of values for the decision variables that optimizes (maximizes or minimizes) the associated objective. If the world were simple and the future were predictable, all data in an optimization model would be constant, making the model deterministic.

Optimization Models With Uncertainty

In many cases, however, a deterministic optimization model can’t capture all the relevant intricacies of a practical decision environment. When model data are uncertain and can only be described probabilistically, the objective will have some probability distribution for any chosen set of decision variables. You can find this probability distribution by simulating the model using Crystal Ball. This type of model is called stochastic.
A stochastic optimization model has several additional elements:

- **Assumptions** — Capture the uncertainty of model data using probability distributions.
- **Forecasts** — Are frequency distributions of possible results for the model.
- **Forecast statistics** — Are summary values of a forecast distribution, such as the mean, standard deviation, or variance. You control the optimization by maximizing or minimizing forecast statistics, or setting them to a target.
- **Requirements** — Are additional restrictions on forecast statistics. You can set upper and lower limits for any statistic of a forecast distribution.

Stochastic models are much more difficult to optimize because they require simulation to compute the objective. While OptQuest is designed to solve stochastic models using Crystal Ball, it is also capable of solving deterministic models. Figure 111 shows that deterministic results are a single value, while stochastic results are distributed over a curve.

**Discrete, Continuous, or Mixed Models**

Optimization models can be classified as:

- **Discrete** — Contain only discrete decision variables.
- **Continuous** — Contain only continuous decision variables.
Mixed — Contain both discrete and continuous decision variables, or any of the other
decision variable types: binary, category, or custom.

For more information on discrete and continuous decision variables, see the Oracle Crystal Ball
User’s Guide.

Figure 112 on page 219 shows that discrete variable distributions are a series of individual values
while continuous variable distributions are an infinite range of values without distinctive bounds
except the end points.

Figure 112  Comparison of Discrete and Continuous Decision Variables

Linear or Nonlinear Models

An optimization model can be linear or nonlinear, depending on the form of the mathematical
relationships used to model the objective and constraints. illustrates linear and nonlinear
relationships. In a linear relationship, all terms in the formulas only contain a single variable
multiplied by a constant. For example, 3x - 1.2y is a linear relationship since both the first and
second term only involve a constant multiplied by a variable. Terms such as x², xy, 1/x, or 3.1x
make nonlinear relationships. Any models that contain such terms in either the objective or a
constraint are classified as nonlinear.

Figure 113  Comparison of Linear and Nonlinear Relationships

OptQuest can handle both linear and nonlinear objectives and constraints. For information on
defining linear or nonlinear constraints, see the Oracle Decision Optimizer OptQuest User’s
Guide.
Factors That Affect Optimization Performance

Subtopics

- Simulation Accuracy
- Number of Decision Variables
- Base Case Values
- Bounds and Constraints
- Requirements
- Complexity of the Objective
- Simulation Speed
- Precision Control

There are many factors that influence the performance of OptQuest. For example, consider two optimization methods, A and B, applied to an investment problem with the objective of maximizing expected returns. When you evaluate the performance of each method, you must look at which method:

- Finds an investment portfolio with a larger expected return
- Jumps to the range of high-quality solutions more quickly

Below is the performance graph for the two hypothetical methods.

Figure 114  Performance Comparison

![Performance Comparison Graph](image)

Figure 114 on page 220 shows that although both methods find solutions with a similar expected profit after 10 minutes of searching, method A jumps to the range of high-quality solutions faster than B. For the criteria listed previously, method A performs better than method B.

While using OptQuest, you will obtain performance profiles similar to method A. OptQuest’s search methodology (see the references in Appendix B) is very aggressive and attempts to find high-quality solutions immediately, causing large improvements (with respect to the initial solution) early in the search. This is critical when OptQuest can perform only a limited number of simulations within the available time limit.
However, several factors affect OptQuest’s performance, and the importance of these factors varies from one situation to another. The list at the beginning of this topic shows factors that directly affect the search for an optimal solution, discussed in the following linked topics.

**Simulation Accuracy**

For sufficient accuracy, set the number of simulation trials to the minimum number necessary to obtain a reliable estimate of the statistic being optimized. For example, you can reliably estimate the mean with fewer trials than the standard deviation or a percentile.

General guidelines for determining the number of simulation trials necessary to obtain good estimates are:

- 200 to 500 trials is usually sufficient for obtaining accurate estimates for the mean.
- At least 1000 trials are necessary for obtaining reasonable estimates for tail-end percentiles.

Empirical testing with the simulation model using the Crystal Ball Bootstrap tool (see the Oracle Crystal Ball User’s Guide) can help you find the appropriate number of trials for a given situation.

**Number of Decision Variables**

The number of decision variables greatly affects OptQuest’s performance. OptQuest has no physical limit on the number of decision variables you can use in any given problem. As the number of decision variables increases, you need more simulations to find high-quality solutions. General guidelines for the minimum number of simulations required for a given number of decision variables in a problem are:

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Minimum number of simulations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fewer than 10</td>
<td>100</td>
</tr>
<tr>
<td>Between 10 and 20</td>
<td>500</td>
</tr>
<tr>
<td>Between 20 and 50</td>
<td>2000</td>
</tr>
<tr>
<td>Between 50 and 100</td>
<td>5000</td>
</tr>
</tbody>
</table>

For very large numbers of decision variables, you may try running more simulations by lowering the number of trials per simulation, at least initially. After you find an approximate solution, you can rerun the optimization by using the approximate solution as suggested values, further restricting the bounds on the decision variables, and increasing the number of trials to find more accurate results.

**Recommended Number of OptQuest Elements**

For best results, keep the number of OptQuest elements of each type below these limits:

- Decision variables < 4,096
- Constraints < 512
Base Case Values

The base case values are the initial cell values listed in the Base Case column of the Decision Variables panel in the OptQuest wizard. The base case values are important because the closer they are to the optimal value, the faster OptQuest may find the optimal solution. If the values are constraint-infeasible, they will be ignored.

For potentially large models with many decision variables, you may find it helpful to first run a deterministic optimization to search for good base case values. Then, use the results as your base case values and run a stochastic optimization. This technique, however, may not work well if you have objectives or requirements defined with other than central tendency statistics.

Bounds and Constraints

You can significantly improve OptQuest’s performance by selecting meaningful bounds for the decision variables. Suppose, for example, that the bounds for three variables (X, Y, and Z) are:

\[
\begin{align*}
0 & \leq X \leq 100 \\
0 & \leq Y \leq 100 \\
0 & \leq Z \leq 100
\end{align*}
\]

And in addition to the bounds, the following constraint exists:

\[10X + 12Y + 20Z \leq 200\]

Although the optimization model is correct, the variable bounds are not meaningful. A better set of bounds for these variables would be:

\[
\begin{align*}
0 & \leq X \leq 20 \\
0 & \leq Y \leq 16.667 \\
0 & \leq Z \leq 10
\end{align*}
\]

These bounds take into consideration the values of the coefficients and the constraint limit to determine the maximum value for each variable. The new, more restrictive bounds result in a more efficient search for the optimal values of the decision variables.

Since constraints limit the decision variables you are optimizing, OptQuest can eliminate sets of decision variable values that are constraint-infeasible before it spends the time running the simulation. Therefore, limiting the optimization with constraints is very time-effective.

Requirements

While the search process benefits from the use of constraints and tight bounds, performance generally suffers when you include requirements in the optimization model for two reasons:

- Requirements are very time-consuming to evaluate, since OptQuest must run an entire simulation before determining whether the results are requirement-infeasible.
To avoid running requirement-infeasible simulations, OptQuest must identify the characteristics of solutions likely to be requirement-feasible. This makes the search more complex and requires more time.

When you use requirements, you should increase the search time by at least 50% (based on the time used for an equivalent problem without requirements).

**Complexity of the Objective**

A complex objective has a highly nonlinear surface with many local minimum and maximum points.

**Figure 115  Graphs of Complex Objectives**

OptQuest is designed to find global solutions for all types of objectives, especially complex objectives like this one. However, for more complex objectives, you generally need to run more simulations to find high-quality global solutions.

**Simulation Speed**

By increasing the speed of each simulation, you can increase the number of simulations that OptQuest runs in a given time period. Some suggestions to increase speed are:

- Use Extreme Speed when practical.
- Use precision control in Crystal Ball to stop simulations as soon as they reach a satisfactory accuracy
- Reduce the size of your model
- Increase your system’s RAM
- Reduce the number of assumptions and forecasts
- Quit other applications

The *Oracle Crystal Ball User's Guide* discusses these suggestions in more detail.
**Precision Control**

For some models, the accuracy of the statistics is highly dependent on the values of the decision variables. In these cases, you can use Crystal Ball’s precision control feature to run a sufficient number of trials for each simulation to achieve the necessary level of accuracy.

You can use Crystal Ball’s precision control feature for several purposes:

- When you are unsure of how to set the number of trials used for Crystal Ball simulations
- If you believe that the stability of the forecast statistics varies greatly depending on the decision variable values

Precision control periodically calculates the accuracy of the forecast mean, standard deviation, and any indicated percentile during the simulation. When the simulation reaches the target accuracy, it stops, regardless of the number of trials already run.

This feature is especially useful for optimization models such as Portfolio Allocation, where the forecast statistics are highly sensitive to the decision variables. When OptQuest selects conservative investments, the variability of the expected return is low and the statistics are relatively stable. When OptQuest selects aggressive investments, the variability is high and the statistics are relatively less stable. Using precision control increases your forecast statistic accuracy while avoiding running too many trials when a simulation reaches this accuracy quickly.

Notice that finding the appropriate precision control settings may require some trial and error. It can be challenging to decide whether to use absolute or relative precision, what is the best precision value in either case, and which statistics should receive precision control. For more information on setting the precision control feature, see the *Oracle Crystal Ball User’s Guide*.

To see the effects of using precision control with the Portfolio Allocation model:

1. **In Oracle Crystal Ball Decision Optimizer, select Run, and then Run Preferences and change the maximum number of trials from 1000 to 5000.**  
   This maximum limit is always in effect, even when precision control is turned on. Therefore, when using precision control, you must increase the maximum number of trials to let precision control achieve the appropriate accuracy.

2. **Turn on Precision Control.**
   a. Select cell C17.
   b. Select **Define**, and then **Define Forecast**.
   c. Click the **More** button in the **Define Forecast** dialog, then click the **Precision** tab.
   d. Select **Specify The Desired Precision For Forecast Statistics**.
   e. Select **Mean**.
   f. Use an absolute precision of 1000 units.

3. **Select Run, and then Run Preferences, and set the following run preferences:**
   - Maximum number of trials to run set to 1000
   - Sampling method set to **Latin Hypercube**
- Sample Size For Latin Hypercube set to 500
- Random Number Generation set to Use Same Sequence Of Random Numbers with an Initial Seed Value of 999

4 Run another optimization.

Experiment with various other precision control settings to see the difference in the results.

**Sensitivity Analysis Using a Tornado Chart**

One of the easiest ways to increase the effectiveness of your optimization is to remove decision variables that require a lot of effort to evaluate and analyze, but that do not affect the objective very much. If you are unsure how much each of your decision variables affects the objective, you can use the Tornado Chart tool in Crystal Ball (see the Oracle Crystal Ball User’s Guide for more information on the Tornado Chart).

The Tornado Chart tool shows how sensitive the objective is to each decision variable as they change over their allowed ranges. The chart shows all the decision variables in order of their impact on the objective.

**Figure 116 on page 225** shows a Crystal Ball tornado chart. When you view a tornado chart, the most important variables are at the top. This arrangement makes it easier to see the relative importance of all the decision variables. The variables listed at the bottom are the least important in that they affect the objective the least. If their effect is significantly smaller than those at the top, you can probably eliminate them as variables and just let them assume a constant value.

**Figure 116  Crystal Ball Tornado Chart**

Before running the Tornado Chart tool, run an initial optimization so that the base case values of the decision variables are close to the optimal solution for your model. You can use the
Tornado Chart tool to measure the impact of your decision variables. For information, see the Oracle Crystal Ball User’s Guide.

**Maintaining Multiple Optimization Settings for a Model**

In this version of OptQuest, optimization settings are stored in workbooks instead of separate .opt files. Only one group of settings can be stored in each workbook. This is convenient for using and transferring models. However, there are times when you may want to have more than one group of optimization settings for a model. In that case, you can create different blank workbooks with one group of settings stored in each. Then, you can open a “profile” workbook with appropriate settings and use it as the primary workbook in the OptQuest wizard. As long as your main model workbook is also open, OptQuest will use the settings in the blank workbook and your model will still run as you intended.

**Other OptQuest Notes**

Subtopics

- Automatic Resets of Optimizations
- Constraint Formula Limitations
- Minor Limit Violations With Continuous Forecasts
- Solutions Still Ranked Even With No Feasible Solution
- Referenced Assumption and Forecast Cells
- Decision Variables and Ranges With the Same Name
- Linear Constraints Can Be Evaluated As Nonlinear
- Evaluation Tolerances and Constraint Equality Statements

The notes in this section can help you avoid problems while using OptQuest and can also assist in troubleshooting any difficulties that may happen:

**Automatic Resets of Optimizations**

If the first simulation does not run for some reason or generates an error, the entire optimization is reset. Otherwise, if the user stops a running optimization or an error occurs after an optimization starts successfully, the results to that point are kept and the optimization is not reset.

**Constraint Formula Limitations**

Subtopics

- Array Formulas
- Date Formatting

The listed sections describe several limitations in defining constraint formulas.
Array Formulas

Array formulas with brackets are supported by Microsoft Excel but are not allowed in OptQuest. For example, suppose you enter a constraint as follows, referencing a named range:

\[ \text{MyRange} > \{0\} \]

An error about an unrecognized range or variable name is displayed.

Date Formatting

It is possible to reference a decision variable cell formatted as a date (such as 2/19/1900) and enter a constraint as follows in the OptQuest wizard’s Constraints panel:

\[ E2 > 2/19/1900 \]

If you do this, OptQuest interprets it as 2 divided by 19 divided by 1900 and does not display an error message.

This behavior is consistent with that of the Microsoft Excel formula bar. For best results, use the Microsoft Excel \text{DATE()} function.

Minor Limit Violations With Continuous Forecasts

Slight violations of bounds can occur in requirements or constraints when evaluating small, continuous forecast values. If present, these violations should be ignored since the differences are very small compared to the relative magnitude of the forecast values.

Solutions Still Ranked Even With No Feasible Solution

If OptQuest fails to find a feasible solution, the Solution Analysis table still ranks solutions in order from best to worst objective value, even though none are feasible.

Referenced Assumption and Forecast Cells

If a constraint formula references an assumption or forecast cell, that cell is evaluated before each simulation runs. It is therefore not possible to enter a constraint on random values in the assumption cells or on the statistics in forecast cells. In general, you should avoid referencing these in constraint formulas.

In requirements, assumption and forecast cells are evaluated at the end of the simulation.

Decision Variables and Ranges With the Same Name

It is possible to define a decision variable and a cell address or range with the same name. If you do that, only the decision variable is accessible on the Constraints panel of the OptQuest wizard.

For best results, always give decision variables and ranges different names and avoid naming decision variables combinations of letters and numbers that resemble cell addresses.
Linear Constraints Can Be Evaluated As Nonlinear

If you have a cell reference in an OptQuest constraint that is more than seven levels of formulas removed from a decision variable, any constraint based on that cell will be evaluated as nonlinear even though it may be linear.

Evaluation Tolerances and Constraint Equality Statements

Evaluation of OptQuest constraint equalities allows for a tolerance surrounding the right-hand value of constraints instead of forcing a true equality. Satisfied constraints such as the following can result, especially with continuous decision variables: 100000.00002513 = 100000.
## Bibliography

### In This Appendix

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal Ball Tools</td>
<td>229</td>
</tr>
<tr>
<td>Monte Carlo Simulation</td>
<td>230</td>
</tr>
<tr>
<td>Probability Theory and Statistics</td>
<td>231</td>
</tr>
<tr>
<td>Random Variate Generation Methods</td>
<td>231</td>
</tr>
<tr>
<td>Specific Distributions</td>
<td>232</td>
</tr>
<tr>
<td>Uncertainty Analysis</td>
<td>232</td>
</tr>
<tr>
<td>Spreadsheet Design</td>
<td>233</td>
</tr>
<tr>
<td>Sequential Sampling with SIPs</td>
<td>233</td>
</tr>
<tr>
<td>Time-series Forecasting References</td>
<td>233</td>
</tr>
<tr>
<td>Optimization References</td>
<td>235</td>
</tr>
</tbody>
</table>

### Crystal Ball Tools

#### Subtopics
- Bootstrap
- Tornado Charts and Sensitivity Analysis
- Two-Dimensional Simulation

### Bootstrap


### Tornado Charts and Sensitivity Analysis


**Two-Dimensional Simulation**


**Monte Carlo Simulation**


Probability Theory and Statistics


Random Variate Generation Methods


Specific Distributions

Subtopics

- Extreme Value Distribution
- Lognormal Distribution
- Weibull Distribution

Extreme Value Distribution


Lognormal Distribution


Weibull Distribution


Uncertainty Analysis


Spreadsheet Design


Sequential Sampling with SIPs

The SIPs format described in conforms to the Stochastic Library approach set forth in the following articles:


These articles are available for download at:

http://www.probabilitymanagement.org

Time-series Forecasting References

Subtopics

- Forecasting
- ARIMA Forecasting
- Regression Analysis

Forecasting


**ARIMA Forecasting**


**Regression Analysis**


Optimization References

Subtopics
- OptQuest References and White Papers
- Metaheuristics
- Stochastic (Probabilistic) Optimization Theory
- Multiobjective Optimization
- Optimization and Simulation in Practice

OptQuest References and White Papers

These references provide further detail on metaheuristic methods, comparisons of optimization methods, and optimization of complex systems:


A variety of white papers concerning optimization are available on the Web site of OptTek Systems, Inc., the company that developed the OptQuest calculation engine. For a list of papers with abstracts, see [http://www.opttek.com/News/WhitePapers.html](http://www.opttek.com/News/WhitePapers.html)

Metaheuristics


Stochastic (Probabilistic) Optimization Theory


Multiobjective Optimization


Optimization and Simulation in Practice

Subtopics

- Financial Applications
- Quality and Six Sigma Applications
- Petrochemical Engineering Applications
- Inventory System Applications

Financial Applications


Quality and Six Sigma Applications


Petrochemical Engineering Applications


Inventory System Applications