Practical Monte Carlo Simulation with Excel

Part 1 of 2
Basics and Standard Procedures

marginalbooks.com

Akram Najjar

Practical Monte Carlo Simulation with Excel - Part 1 of 2

(Basics and Standard Procedures)

Akram Najjar

Monte Carlo Simulation with Excel - Part 1
(Basics and Standard Procedures)
By Akram Najjar
POB 113-5623, Beirut, Lebanon

Visit the site for this book at: www.marginalbooks.com
Email the author at: info@marginalbooks.com
For other eBooks by the author visit: www.xinxii.com

Library of Congress Cataloging-in-Publication Data
Registration submitted to the US Copyright Office
Case 1-4306651971 (12 December 2016)

E-Book ISBN: 978-3-96142-310-1
GD Publishing Ltd. & Co KG, Berlin
E-Book Distribution: XinXii
www.xinxii.com

Acknowledgements

1) All line graphics in this book are by the author.

2) All other graphics are imported workouts developed using Microsoft Office.

3) Cover Image by the author

Table of Contents

1.0 Introducing Part 1 of this eBook

Welcome to Part 1 of this very direct eBook that presents practical ways of conducting Monte Carlo Simulations using Excel. Sure, there are other products with packaged and more direct procedures. Let us not start a features war. (Here is a link to a Wikipedia page that shows a table with most of the Monte Carlo Simulation add ins in the market: Click Here). The eBook will only use native features of Excel, no more. (All workouts and supporting material can be downloaded as shown in chapter 5.0).

This is Part 1 of a two part eBook. The reason the eBook is broken down into two parts is that it grew beyond control. Each time I addressed a modeling situation, I found another that warranted simulation. I also felt that much was needed to clarify basic practices that are needed in the modeling.

Part 1 (Basics) starts almost immediately with a detailed example showing how Monte Carlo Simulation is conducted in Excel. It shows you how to prepare Excel for the various workouts. It establishes basic Monte Carlo Simulation practices which will be used in the models in both parts of the eBook. It also includes a variety of appendices that support simulation. These are all supplemented with around 20 fully solved workouts available in the downloadable zipped file. (See the next Chapter).

Part 2 (**Applications and Distributions**) presents different applications of Monte Carlo Simulation. These are mostly grouped by "sector" such as project management, reliability engineering, acceptance sampling and queuing models. It also takes another look at applications by addressing specific statistical distributions and discussing when to use each distribution and how to implement it in Excel. These chapters are all supplemented with around 55 fully solved workouts available in the downloadable zipped file. (See the next Chapter).

2.0 Downloading the Supporting Files

Part1 1 of this eBook is supported by a set of files included in the zipped file that is available for download from www.marginalbooks.com. To download the zipped file, Click Here. Check the last sentence in this book for the code to open the zipped file. On Unzipping the file, the files will go into the following 3 folders: 01. Workouts Part 1 (also contains the solutions)

03. Templates
04. Supporting Documents

1) Workouts: this folder contains the material needed for the workouts in this eBook.

The Naming of Files: when we refer to the names of files in the downloaded zipped file, we will not include the sequence number nor the file extension, just the name. However, all workbooks in the zipped files are numbered as per the workout sequence in the eBook.

2) Workouts for Part 2 of this eBook. These are not included in this download.

3) Templates: one or two templates are included for your use.

4) Examples: various examples referred to in the text are included in this folder. It also contains original Microsoft Visio and Mindjet MindManager documents used in the eBook.

3.0 Alerts, Guidelines, Exclusions and Apologies

The following paragraphs highlight various alerts, guidelines, exclusions and apologies.

1) **Monte Carlo Simulation in the Sciences**: a quick search for "Monte Carlo Simulation" books will result in a large number of books that focus on using Monte Carlo Simulation in physics and some in financial applications (derivative pricing, variance, etc.). These will not be addressed in this book. The main reason is the highly mathematical discussions required for their understanding.

2) **Monte Carlo Simulation Algorithms or Methods**: you will find references in the literature to these two terms. This eBook will not be concerned with such techniques. They are mathematical procedures for very specific and specialized uses (although not too different from ours). We will therefore use the term **Monte Carlo Simulation** to refer to a process and a set of practices. The process will be defined in Chapter 8.0 in this eBook.

3) **VBA or Visual Basic for Applications**: it is better to ask permission than to seek forgiveness. I tried very hard to develop an eBook purely based on Excel Functions. But I hit a wall when using models that have runs where each run has sub-runs. For example, you are simulating a production line with 5 steps. However, each step might follow a different procedure, which you also want to simulate. These become "sub-runs". This is one facility that cannot be implemented without VBA. I have written a standard VBA module that you can simply insert in your model and configure to arrive at simulating sub-runs. No programming competence is needed. Refer to Chapter 13.0 for a detailed discussion of this procedure. The Appendix in Chapter 18.0 covers the setup of the VBA module and its code.

4) **Complete description vs. summarized work**: when planning the development of this eBook I considered two approaches. The first approach was to present a the models and their workouts as summaries with a brief background of the main issues. This would have avoided the level of detail. It would have also left most of the logic to be worked out by the reader on his or her own. This could be useful for persons who are familiar with the main ideas behind Monte Carlo Simulation and Excel. The second approach was to assume a blank slate and start each model or workout from scratch. Later workouts or sections that repeat the use of such material would be more summarized. We can then refer to the more elaborate and earlier formulations for more detail.

The second approach was selected to avoid an annoyance I often face in such books. They present elaborate procedures where the author skips through minor steps that he or she considers need no clarification. The hours spent trying to fathom what a technical author meant by a specific formula or procedure caused me to adopt the second

approach. I figured it would be easier to skip what you do not need than to wonder what and how these steps were arrived at.

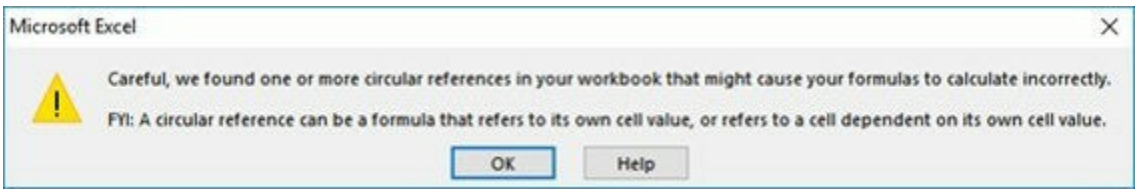
5) **Repetitions**: I assume readers may select to go directly to a specific subject of interest to them. Such a book will most likely never be read sequentially. This meant that chapters had to be almost "stand alone" or “quasi-complete”. The decision was to repeat summarized procedures rather than to cross reference them. It also supported the decision to detail the models as per the above paragraph.

6) **Images and eBooks**: most of you will be reading this eBook in ePub or Mobi format (although there is a PDF version at www.xinxii.com). The PDF version is a copy of the original Microsoft Word version and hence retains all formatting. It also allows images to be as wide as the page. Sadly, ePub and Mobi formats restrict images to be no wider than 600 pixels or 12 cm. The result is that some spreadsheet images had to be shrunk down and may not be visible. To counter that result, I recommend that you refer to the solution of the specific Workout in the Workouts Folder.

7) **Copying from other Workbooks**: many of the models in both parts of the eBook are elaborate to setup and format. Effort was made to define all entries in case you prefer to start each workout afresh. However, you can get to the heart of the matter of each workout by using the solution workouts found in the Workouts Folder: a) Copy labels (in column or row headers)

b) Use the Format Painter to copy the format as the numeric format differs from one cell to the other
c) In the case of complex formulas, to avoid wasting timing entry the exact syntax, copy the formulas from the solutions workouts.

8) **Circular References**: on opening some of the workbooks, you might get a message from Excel indicating that there is a circular reference in the workbook. Whereas this is often a serious error, Excel allows us to override this situation if we wish.



In Chapter 13.0, we discuss a technique whereby for each simulation run there we need to simulate “sub-runs”. This is very useful but it requires circular referencing, under control, of course. In the discussion, we will explain the way circular referencing is used and enforced in Excel.

9) **Recalculation**: several workbooks have a large number of calculations to complete and this happens each time something changes in the workbook. Since the option to automatically recalculate is on, by default, some of the workbooks might take a long time to process or to save.

4.0 The Rationale for Monte Carlo Simulation

Monte Carlo Simulation grew up in the field of physics. Physicists needed to find a way of solving problems for which they did not have a closed form or a mathematical solution. Monte Carlo Simulation allowed them to approach solutions, numerically.

The set of simulation techniques were developed by some of the great minds of the physics world. Early usage was reported in the late 19th century. More serious applications came out in the mid 30s and saw fruition in the Manhattan Project that developed the A Bomb. Ironically, the methods developed when Stanslav Ulam tried to work out the probability of getting a solution (or a closed form) for solitaire. He could not, so he resorted to numerical algorithms that gave him better results. All these efforts were re-enforced by the rise of digital computers (although Enrico Fermi used analog computers for some early Monte Carlo Simulations).

A) The Delphic Oracle and the Delphi Method (Technique)

One of the nightmares anyone in middle management supposed to provide the upper echelons with estimates is that he or she has to provide single point estimates. One value for each parameter. Invariably, such a person will have one and only one chance to provide a point estimate for each parameters. The required estimates are forecasts, no more. What are the chances that such forecasts would be correct? We should remember the wonderful line by Sam Goldwyn (MGM): “forecasting is dangerous, especially about the future”.

We need an alternative. To appreciate that, let me tell you about the Delphi Oracle, both ancient and modern. In ancient Greece, from 800 BC to 300 BC, folks traveled long distances to Delphi to seek predictions about their hopes or fears, whether military, financial, agriculture or personal. The temple was a sanctuary for Apollo whose priestess there spoke in his name using sacrificed animals.

Years later, the term “Delphic Method” would be used for judgmental forecasting, in a more rigorous and robust manner. I had the chance to observe its use in a conference at the Institute of Electrical Engineers (IEE) in London (Now renamed as the Institute of Engineering and Technology, IET). It was in spring 1977, a time of unusual excitement in the electronic world due to the introduction by Intel of microprocessors. Around 400 engineers were attending the talks. At the beginning of the conference, each one was handed an index card that looked like a spreadsheet. The columns indicated quarters starting from summer 1977 and up to 1987. The rows were dedicated to questions such as “when do you think we will have color monitors, as a standard?” or “when do you think microprocessors will replace mini-computer processors?” and so on. Towards 3 pm, the cards were collected, posted into a database and analyzed. The results were

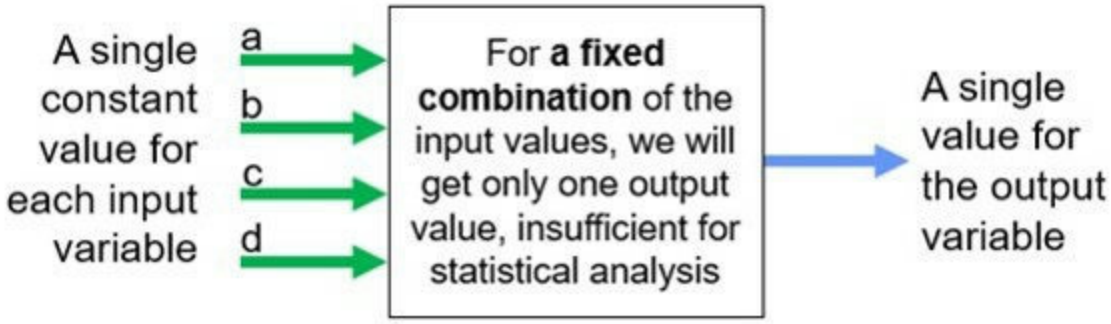
communicated in terms of the average of each response and its standard deviation. Would 400 engineers be wrong? Maybe, but not all of them. Statistical analysis can then provide such information as the average, the standard deviation (the extent of spread of observations around the average) and other significant measures.

The Delphi Method (sometimes called the Delphi Technique) was develop in the early 50s in the Rand Corporation by Olaf Helmer and Norman Dalkey. (Click Here to review the history and mission of the Rand Corporation). These days, it is quite commonly used for judgmental forecasting, i.e., forecasting that is meant to produce exploratory results which are not deterministic or conclusive.

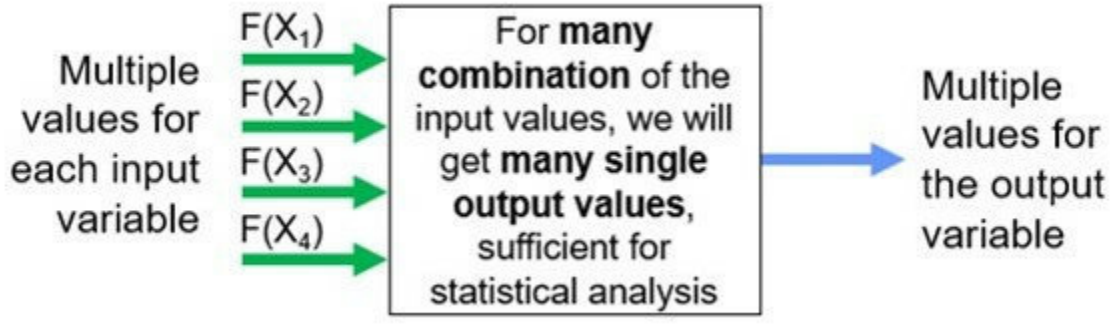
How is Monte Carlo Simulation like the Modern Delphic Method?

Monte Carlo Simulation follows the basic principles of the Delphic Method, numerically. Traditionally, the result of a formulation based on one set of fixed constants (or inputs) will not give a realistic answer. By setting up our formulation (or model) so that it can be run through thousands of times (engineers) using different but realistic values for the input variables will give more significant and realistic results.

We will effectively move from this model:



To this model:



C) The Necessities that Led to Monte Carlo Simulation

What was the necessity that forced scientists to invent Monte Carlo Simulation?

Necessity 1: to remove the bias of single point estimates

If we say "the price is around $120", then when it comes to plugging in the price in a formulation, we must remove the word "around" and only use a single value as an input.

This builds in error and bias. Secondly, whose estimate shall we use? Imagine a brainstorming session where each person insists on using a specific input for the price of the equipment in a proposal.

Necessity 2: **to handle a large number of input variables**

If we have 2 input variables with 3 possible values for each, we would have 3 *3* 3 = 81 combinations. Even such a limited model would be difficult to handle manually. We would need to enter each combination of input variables and then note the resulting output.

When we have a large number of input variables with a wide range of possible values, the number of combinations will grow exponentially. Consider a Net Present Value (NPV) formulation where we have revenues and costs over 12 quarters, a discounting rate and 2 exchange rates. We would then have 2 * 12 + 1 + 2 = 27 variables. Assume that each of these can spread over 20 possible values. How many combinations will we have of the input variables? We will have 20 ^ 27 which is an astronomically large number. Monte Carlo Simulation resolves this issue.

Necessity 3: to handle probabilistic behavior of input variables

Chance disrupts everything. In the above two examples, we are totally in control of specifying our input values. What if these were subjected to random variation?

Suppose you have to estimate the time it takes to test an item such as the strength of a steel rod or the number of defects in a manufactured item. And suppose we know that these measurements are normally distributed (i.e., following the bell shaped curve). Rather than use the average, it is more realistic to create a model that uses thousands of samples for the specific measurement. When plotted, they will result in a bell shaped curve.

There are tens of distributions that can describe various operational phenomena. Each input variable will have a specific behavior that complies with a specific distribution. This is where Monte Carlo Simulation comes in.

Necessity 4: to process models that have no formal solutions

Historically, Monte Carlo Simulation grew out of the need of physicists to solve equations computationally when they could not get a mathematical solution. Luckily for physicists, most of their problems are supported by theoretical or closed formal solutions. We are less lucky in that most of our problems in the operational or financial spheres do not have a mathematical solution. What is the mathematical “closed form” for handling sales forecasts which include returns, end of season discounts and fluctuating cost prices? Again, Monte Carlo Simulation and spreadsheet power come to the rescue.

D) Modeling and Cartography

In Jorge Luis Borges's “Collected Fictions” (Penguin 1999, page 325), there is a

brilliant short story called "On Exactitude in Science". It is no more than 150 words long and is presumably based on another tale called "The Man on the Moon" in a short story by Lewis Carroll called “Sylvie and Bruno Concluded” (Nabu Press, page 156).

In Carroll's story, a fictional map was drawn with a scale of 1 mile to the mile. One of the characters remarks: "we use the country itself, as its own map and I assure you it does nearly as well." Borges takes up the tale and tells the story of cartographer who were not satisfied with traditional maps and developed one that was the same size as the empire, exactly matching it point by point.

Simulation is like cartography. You have to remember what scale you are using. The more realistic the model, the nearer to the truth and the more complex it will be (and harder to fold). Monte Carlo Simulation adds to the significance of our models by trying out different values and providing us with a statistical wrap up of the results. With Carroll and Borges as guides, how can we get lost?

5.0 Guidelines and Good Practices for Modeling with Excel

The following guidelines and good practices apply to the development of spreadsheet models. They also apply to various ways of using the examples in this eBook. We start with common usages and follow those up with some good practices.

We will also introduce the Analysis Toolpack that comes with Excel.

The Appendix in chapter 17.0 has additional Excel facilities such as the use of spinners and the accompanying VBA module for use in sub-runs (see chapter 13.0).

A) Functions and Features you Need to Master in Excel

Although we will usually explain most formulas when we use them, it helps if you are competent in their use before reading this eBook.

Use the Help File in Excel as it has examples that can easily be copied to a temporary workbook for practice. Here is a recommended list to know:

Addressing Functions: INDIRECT(), INDEX(), OFFSET() and ADDRESS()
COLUMN() and ROW()
COUNTIFS() *COUNTIF()* SUMIF() / AVERAGEIF()
Error Checking Functions: IFNA(), IFERROR(), *etc.*
FIND()
Forecasting Functions: SLOPE(), INTERCEPT(), FORECAST(), CORREL()
FORMULATEXT()
FREQUENCY() (Specifically using it as an Array function)
LARGE()
Logical Functions: AND(), OR(), XOR(), IFNA(), IERROR(), *etc.*
MATCH()
PASTE as values, transpose, multiply, add, *etc.*
ROUNDUP() / ROUNDDOWN()
ROW / COLUMN()
String functions (including concatenation using &)
TEXT()
TRUNCATE() / INT() *CEILING()* FLOOR()
VLOOKUP()

We will detail statistical functions during the workouts as they are critical.

There are other features of Excel that you would need to know very well:

Absolute vs relative referencing (for flexible copying of cells)

Autofill (which will be used to generate the Run ID's)
Conditional Formatting
Copying sheets within workbooks and to others
Data Validation
Entry and use of Array Functions (FREQUENCY(), etc.)
Name Manager (defines names for ranges to use in formulas)
Nesting of functions, specifically IF's
Quick Access Toolbar and its customization
Sensitivity Analysis: WHAT IF, Scenario Manager, Goal Seeking

B) The Office Button vs the FILE Menu

We will be using Excel 2013. However, for most of what we will present, Excel 2007 and 2010 are very similar to 2013. The same applies to Excel 2016.

One exception: Excel 2007 does not have a FILE menu (top left) like 2010 and 2013. It has an Office button (sometimes referred to as the Home button). Generally, it behaves in the same way as the FILE menu. We will frequently need to set various options in Excel. In 2007, these are accessible through the **Excel Options** button found at the bottom of the dialog box you get after pressing the Office button.

For versions after 2007, the Excel Options are found under the FILE menu. This is what we will use.

Versions earlier than 2007 have the same computational facilities (except some changes in functions to be noted next). However, the facilities are found under a different menu structure.

C) Functions in 2007 and Later

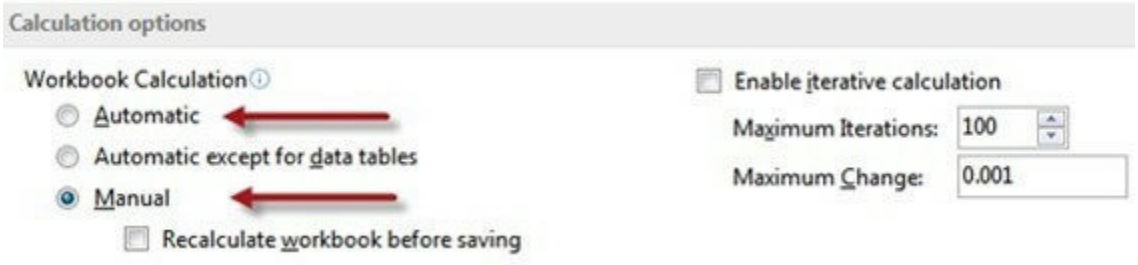
A few of the statistical formulas have been upgraded to be more accurate. The old functions still work but Excel produces more accurate output with the upgraded functions. For example:

NORMINV() becomes NORMS.INV() and
NORMDIST() becomes NORM.DIST()

A quick search on the web will define the differences between the functions of 2003 (and earlier) and those found in 2007 and later. Click Here for a good list from Softwarelosungen.

D) Manual vs Automatic Calculation

We may need to enable or disable the "Calculation options" in the FILE *OPTIONS* FORMULAS tab:



By default, the "automatic" calculation option is enabled. Every time you enter or select anything in the sheet, Excel will recalculate all formulas.

In the early days of slow computers, each time a recalculation was initiated, it took a long while to complete. Spreadsheet developers provided an option to disable calculating unless the analyst wanted to. When this option is disabled, you can press F9 to force a recalculation. Today this is not needed, unless, of course, you have a very large formulation (as some of our models will have). We will assume that the option is enabled for automatic calculation unless we call for the manual option.

E) Add and activate the Analysis Toolpack (ATP)

The Analysis Toolpack (ATP) is an Add In that comes with Excel, free of charge. It contains many statistical functions. It also has a variety of data analysis tools such as random number generation, statistical tests (t, F, ANOVA) and generation of histograms. Although it is powerful, it has two major shortcomings: a) It is visually challenged, reminding those who used PCs in the 80s, of dBase II reports.

b) Most of its output is static. This means, once generated, the output will not change if the input data gets changed.

In all our analyses, we have opted for dynamic analysis. This will allow you, as an analyst, to manipulate your input variables and model parameters and immediately get the output as opposed to have to go through the ATP procedure again.

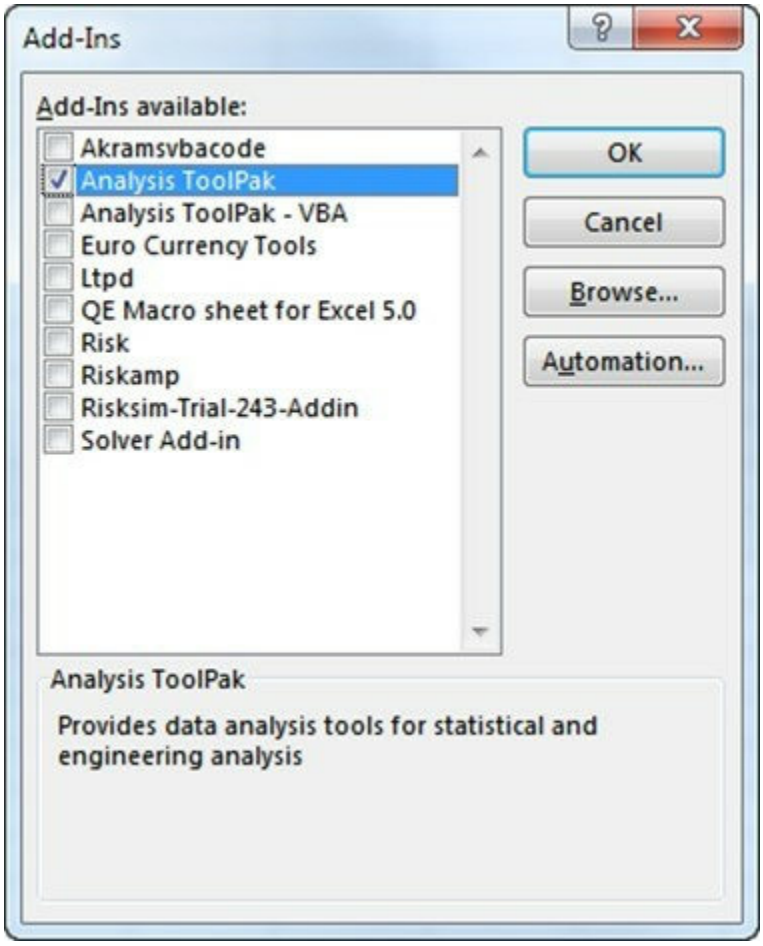
However, there are a few useful features that we will use in this eBook, regardless of the shortcomings above.

The ATP is included during the standard installation of Excel. However, by default, it is disabled as an Add In. You have to enable it when you need it.

To enable or disable the Analysis Toolpack:

a) Select the menu item FILE *OPTIONS* ADD-INS and select the Excel Add-ins and click on the GO button:

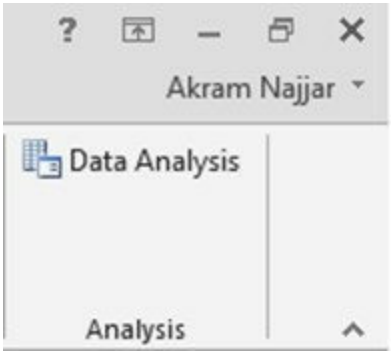
b) Enable the **Analysis Toolpack Add in** by checking its box:



If it is not in the list, then your installation may have excluded it. Try to update the installation and request that the ATP be included.

There is no need to add the **Analysis Toolpack VBA** unless you need to call Toolpack functions from VBA. If you keep it active, it might delay the startup of Excel.

To check that the Toolpack has been installed and enabled, review the DATA menu. It should appear on the extreme right hand side as "DATA ANALYSIS":



If it is not present, then either the ATP is disabled or it is was not installed at all.

F) Excel Conventions used in this eBook

a) **The Naming of Functions in the eBook**: most functions in Excel have one or more arguments. When discussing the functions in detail, the arguments will be explained and specified. However, to improve the legibility of the text, a set of parentheses will be

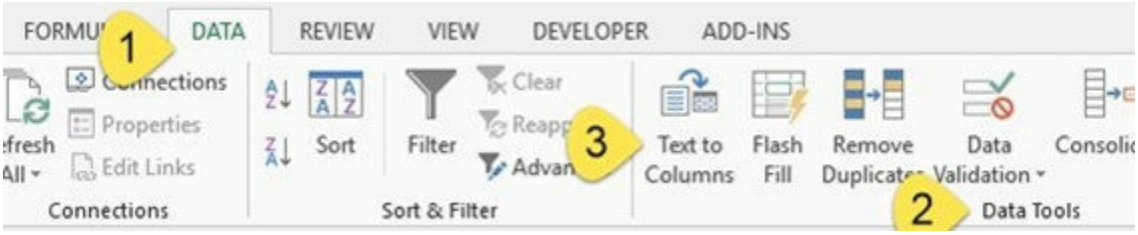
shown without specifying arguments. For example, The full function is: NORM.DIST(probability, mean, standard deviation, cum)

We will refer to it as: NORM.DIST()

In all cases, selecting the menu item FORMULAS / INSERT FUNCTION will lead you to the function wizard. The dialog box for each function has a help link. Each help page has direct examples you can copy into Excel or download the file.

b) **Referring to Menus and Groups in the Ribbon**

Menu and groups will be stated in upper case. For example, to convert text to columns, we will state the instruction as "Select the menu item DATA *DATA TOOLS* TEXT TO COLUMNS". This is found on the ribbon as follows:



c) **Probabilities in % or in Decimal**: it is common to use percentages when discussing probability. However, the range 0 to 100% corresponds to the range 0 to 1 in decimal format.

This eBook will start by using the % format and will slowly move into using the decimal format. The main reason is that using % is a bit confusing in Excel. If you enter a % symbol after the number, Excel assumes your number is a percentage and keeps the value as entered. If you enter a decimal value and then format it as a %, Excel will convert the value by multiplying by 100 and adding the "%" symbol.

It is best to stick to decimals but will keep the "%" in the names of the columns. For example, we will have a cumulative % column with numbers such as 0.15, 0.30, 0.60 and 1.0.

d) **Color Codes**

We will follow the practice to color some cells to give them meaning. We will avoid numerous colors, font sizes and highlights.

Green is used for the input or the changing variables (signifies starting)
Yellow is used for constants (symbolized by the consistency of the Sun)
Blue is the formulation objective (symbolized by the color of the sky)

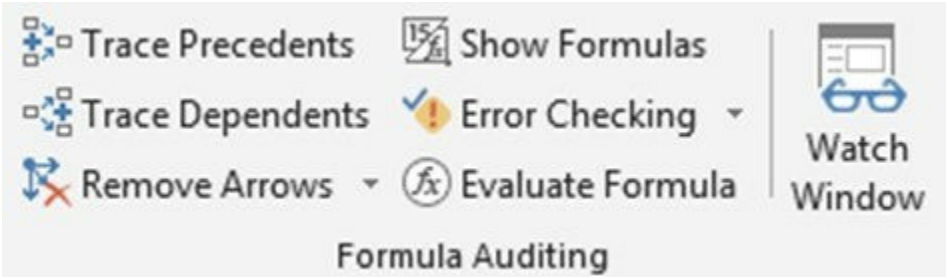
In various columns, we will also use conditional formatting to highlight certain values. For example, all negative values or all cells with "Yes" or with "Balked". These are colored in red with a white font.

G) Good Practices in Excel

Here are some conventions and practices that we will use throughout the eBook. But before that, let me recommend this valuable book. It presents a wide range of guidelines and tips on how to formulate your spreadsheet models: Patrick O'Beirne: "Spreadsheet Check and Control" published by Systems Publishing, Wexford, Ireland, 2005.

1) **Ensure your model is robust**: a robust model is resistant to error. Here are some recommendations:

a) **Protect cells** or ranges to disallow entry. This is particularly useful when the model is to be used by others.
b) **Apply validation rules** to those cells that you use for entry. For example, if a cell contains the name of a location, create a drop down list containing the allowed or valid locations. There are other validation schemes such as defining acceptable ranges for numbers, *etc.*
c) **Cross check formulations**. For example, in all tables that have cumulative totals, the last cumulative entry would be the same as the sum of the individual (non-accumulated) entries. Suppose you have a Row with an expenditure value for each of the 12 months of the project. The Row below it would be a cumulative Row. The total of the 12 months must equal the last cumulative entry, i.e., that of December. Check these two against one another and provide a notification if they mismatch.
d) **Use the formula validation facility.** Under the FORMULAS menu, the FORMULA AUDITING group has a variety of tools to check dependencies between cells, review of formulas and watching of temporary results:



e) **Break a formulation** into modules or even single steps that are easily followed and tested. This allows you to track the logic more easily. This will be found in most of our models. Rather than placing the full formula in one cell, break up the Future Value formula into steps:

	A	B	C
1	Present value	100	
2	Number of periods	12	
3	Periods per year	4	
4	Annual rate	0.12	
5	Rate per period	0.03	=B4/B3
6	Future value	142.58	=B1*(1+B5)^B2

The breakup of a formulation allows you to check the validity of the procedure as

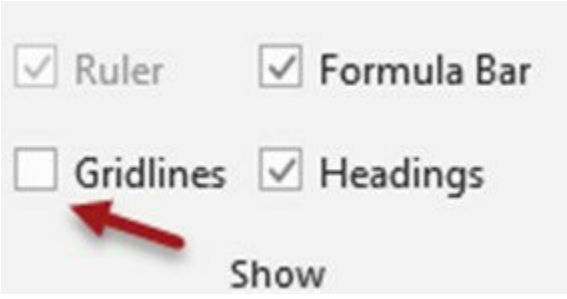
well as to manipulate single values to view their effect on the result.
f) **Use the FORMULATEXT() function** to display the text of a formula in the cell next to it (usually to its right). The entries in B5 and B6 above are supplemented by FORMULATEXT() in C5 and C6.

2) **Develop easy to use models**: this would make it easier for you to modify the model and the users to easily understand it. Here are some recommendations:

a) Provide forms for entry
b) Use spinners and scroll bars to allow for easy manipulation of cell values
c) Use drop down lists (or combos) for preset cell values
d) Customize the **Quick Access Toolbar** to speed up development and use of models
e) When possible, add buttons within the sheet to launch macros
f) Use a standardized color code (as recommended in the previous section)

3) **Format the model for ease of use and documentation**: remember the number of times you went back to a spreadsheet that you had developed only a month ago and could not work out what you were doing then? Here are some recommendations: a) **Format numbers** to the same precision. Excel uses the general format which means it will show as many figures after the decimal as there are. This results in misaligned columns.

b) **Align text and numbers differently**: text to the left and numbers to the right. Avoid centralizing numbers as their different sizes will make them difficult to read.
c) **Avoid over-ornamenting** your spreadsheet as that might inhibit clear analysis. Be stingy with the number of fonts, colors, different highlighting modes (italics, bold and underlines) and sizes.
d) **Remove the grid** after your are half way through the formulation. This will provide you with a clearer view of the formulation. The option is found under the VIEW / SHOW menu:



e) **Use conditional formatting** whenever possible. It would help you spot outliers, wrong values or special values.
f) **Let the formulation flow** from the top left hand corner going down and to the right. This makes the model easier to manipulate, update or reengineer.
g) **Use Excel's Name Manager facility**: this would help you enter formulas more easily.
h) **Move infrequently used data** to other sheets. Create a special sheet that contains

all constants and refer to them by name. You can also create special sheets that contain any tables that you need such as inventories, price lists, *etc.* There is no need to view these in the main sheet of the model.

4) **Avoid bad practices**: a model often gets developed without much planning. This tempts the analyst to take short cuts while developing the proto-type. As and when the model matures, these short cuts and bad practices remain as computational time bombs. It is important to remember the following: a) **Avoid inserting constants in formulas**. If you need constants, place them in a separate sheet. Give them defined names and use these names in the formulas. This would allow you to globally modify the constants. It also reduces clutter in the main Model sheets.

b) Remember to use **absolute references with constants** or use defined names (which are made absolute by default).
c) **Document clever tricks** (if you really have to use them). Without documentation, these become knots in less than 1 week. Documentation can be through comments inserted for particular cells, text boxes or even a whole sheet dedicated to such information.

5) **Watch out for rounding issues**: Excel often formats results showing rounded values but internally, keeps the values under full precision. For example, you have a column made up of 10 values formatted to show #,###.00. What you see will be truncated but Excel will store the actual values in their full precision (within Excel's limits). If an accountant were to total the column as printed, the result would not be the same as what Excel totaled. Here is an example:

	USD Countervalue	
EURO Value	What Excel Stores	What Excel Shows
13,000.00	16,305.90000	16,305.90
14,313.00	17,952.79590	17,952.80
34,125.00	42,802.98750	42,802.99
44,310.00	55,578.03300	55,578.03
32,456.00	40,709.56080	40,709.56
32,445.00	40,695.76350	40,695.76
55,553.00	69,680.12790	69,680.13
	283,725.1686	**283,725.17**

Euro Exchange Rate	1.2543

There is a difference between what Excel stores and what it shows. This difference might arise in more serious places where, for example, you are comparing two values with one another as a decision in the flow of the model. The recommendation is to truncate to two decimal places (or whatever you need in your model) and remain consistently with this precision throughout.

6) **Use Absolute References**: Excel enters formulas with relative addresses.

a) Enter 4 into B4 and 7 into A7
b) Let A1 = B4 * A7 = 28
c) Copy A1 to A2 and the answer will be zero.

Excel uses relative addressing when copying cells from one range to another. If you copy a cell pointing to B8 in the Constants sheet down 2 rows, Excel will change the Row by 2 and point to B10. If you copy the cell contents of B8 four columns to the right, Excel will change the value of B8 to F8. (Click Here for the help page on Microsoft's website).

(Sometimes, you need to fix either rows or columns. For example, expressing the cell address as $B8 fixes the column wherever you copy but not the Row. B$8 fixes the Row but not the column).

The formula in A2 becomes B5 * A8 = 0. To avoid this when there is a need to copy formulas to the right or down, use Absolute Addressing by pressing F4 when highlighting a reference inside a formula. This can freeze both rows and columns or freeze one of them. It needs practice. We will be using it all the time.

7) **Use the help file in Excel** to understand its functions. This is one of the best aspects of this product. Microsoft have provided wonderful examples for each function, all of which can be copied into a worksheet and played with.

8) **Test, test and test** and when you think all is perfect, test again. Testing is a whole discipline on its own but can be summarized as follows: test modules on their own, try different values than what you've entered, check totals by selecting ranges, use the counter check recommended earlier, *etc.* Most importantly: get someone else to test your model for you (**peer review**). Your eyes will get used to what is on the sheet and with time, will hide errors or invalid formulations from you. An outsider will immediately spot these anomalies.

6.0 Our First Full Monte Carlo Simulation

The simulation model we present in this chapter is meant to give you a quick 5 cent tour of the simulation process. It is presented as a step by step implementation of the process that we will be using (and diversifying) in later chapters. It will give you a feel for where we are going and the stations where we will pause. The problem is straightforward and so is the distribution we will use.

Workout 1: Equipment Costing (UNIFORM)

Purpose: your company is preparing a bid for an electro-mechanical engineering project. You are given the responsibility of costing a power generator which is part of the bid. If you price it too high, other lower cost proposals will be favored. Price it too low and you might get the job. However, if you get the job, you will quite likely have to by the generator at a higher price than what you bid. Your task is to arrive at the most realistic costing possible.

The model will use the Uniform Distribution to draw samples of the cost of the equipment and its spares. This means, we will be using prices that range uniformly or are equally likely to arise in the market.

Problem statement: the total cost is made up of the following elements:

The Cost of the Equipment: the power generator has to be purchased. It is known to cost "around" 15,000. However, in the market, the price varies depending on various cost drivers which are not known at the present. The variation of the cost price is uniform: from 12,000 up to 17,000. This means each possible purchase price in that range is equally likely to arise.

The Cost of the Spare Parts for 3 years: these are estimated at around 1200 for each of the 3 years. The variation of the price of the spares is also known to be uniform and varies from 1000 to 1400. The cost of spares in each year is independent from that of the others. Each year should be sampled separately but from the same distribution.

The Maintenance Charges for 3 years: this is charged at an annual rate of 12% of the equipment price (not including the spares). This % is fixed by contract and cannot not be varied. We do not need to sample the maintenance charges as they are dependent on the sampled cost of the equipment. In an extension of the simulation, you may wish to consider the 12% rate as a random input variable, say varying according to a bell shaped curve. (More of this later).

The costing is covered by the following formulas:

Total Cost = Equipment + Spares + Maintenance
Where...
Spares = Spares for Year 1 + Spares for Year 2 + Spares for Year 3
Maintenance = Equipment * Maintenance Rate * Number of Years

We will therefore need:

a) Constants: maintenance rate, number of years
b) Distribution parameters: the upper and lower value of the equipment cost
c) Distribution parameters: the upper and lower value of the spare parts costs

The objective of this simulation is to vary the price of the equipment and spares as per their distributions. We will prepare a work sheet with 1000 rows. Each Row will calculate the total cost as in the formula above. However, the cost of equipment and the 3 cells for the spares will be extracted from a randomly sampled distribution.

By analyzing the 1000 total costs, we can arrive at a budgetary estimate for the total cost which is based on different probabilistic values and the above constants.

Your boss is well versed in statistics and tells you that the company needs to use an estimate of the total price which is safe. What he or she means by safe is not specified now. You need to provide results that allow your boss to view the results according to different levels of "safety". For example, the safety may be expressed as follows: we should make sure that our estimate contains 80% of all possible variations as per our problem above. This means the price selected will be 80% above all possible prices in the replications. This is safe but high. The boss might feel adventurous and say: we need this project, so what is the price that is 30% above the others?

The Procedure:

Here is how we setup and run the Monte Carlo Simulation. Most of these steps will be standard practices which will be reused in later models. In the Workouts Folder there is a fully solved model called: **Equipment Costing - Uniform**.

Step 1: create a new workbook and give it any name you wish. Create the following sheets (you can also use the Model Template with these sheets already created):

Runs will contain the model (or formulation) and the 1000 replicated simulation runs. Color the tab green because it will be changing or it is our start.

Results is initially blank. Color its tab blue because that is the color of the sky, our objective. After you run the simulation in the Runs sheet, you will need to copy the resulting final cost column from the Model into the Results sheet. The Results worksheet will be where we generate histograms, charts and descriptive statistics for analyzing the model.

Constants will contain the constants of the model as well as any parameters and assumptions used. (Color the tab yellow because the sun is constant).

Step 2: The constants we need are the **maintenance rate** and the **number of years** of maintenance. These are entered in the Constants sheet as follows:

	A	B
1	**Maintenance**	
2	Annual Rate	0.12
3	Contract Years	3

Step 3: develop the formulation and test it with constants or fixed values rather than probabilistic estimates. In Step 6 we will replace the input variables by randomly generated samples.

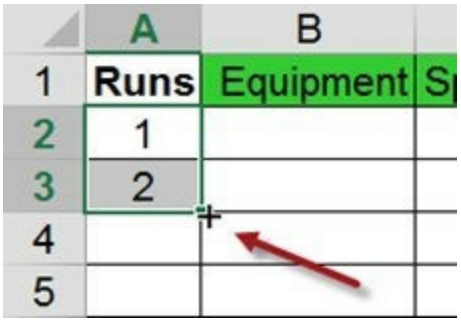
a) In the Runs sheet, enter the following labels:

A1 = Runs
B1 = Equipment
C1 = Spares Y1
D1 = Spares Y2
E1 = Spares Y3
F1 = Maintenance
G1 = Total

b) Create the Run ID in the range A2:A1001. Use Excel's autofill facility to generate the sequence 1, 2... 1000 in the range A2:A1001. To do that:

Enter 1 in A2, 2 in A3 and select A2:A3.

Hover the cursor until Excel shows you a cross:



Drag the cross downwards until you reach Row 1001. Excel will fill a sequence from 1 to 1000 in cells A2:A1001. As you drag the cross downwards, Excel show you the value that it has reached making it easy for you to know where to stop.

Note: the Run ID's have **no computational value**. We use them to refer to rows, if we have to. In some of our more elaborate models, we will use the Run IDs as the basis for looping through the rows. (See chapter 13.0 on the practice of using sub-runs).

c) In the Runs sheet, enter the following fixed values in the range B2:E2. For the time being, these are fixed or single point estimates. They will only be used to validate or

test the calculations in the model. They will later be replaced by probabilistic samples.

B2 = 25,000
C2 = 3200
D2 = 3400
E2 = 3600

d) In F2 calculate the maintenance as the cost (B2) multiplied by 3 years (from B3 in the Constants sheet) multiplied by the Maintenance Rate (from B2 in the Constants sheet):

F2 = B2 * Constants!$B$3 * Constants!$B$2

e) In G2, calculate the total cost which is the sum of the cost of the equipment, the 3 years of spare parts and the maintenance charges.

G2 = SUM(B2:F2)

This gives us the following single point estimate result (notice the color coding of the column headers):

	A	B	C	D	E	F	G
1	Runs	Equipment	Spares Y1	Spares Y2	Spares Y3	Maintenance	Total
2	1	15,000	3,200	3,400	3,600	5,400	30,600

Once you've **checked** that the single point estimates give the right results, you will be able to replace these values (in the range B2:E2) by the corresponding samples from the UNIFORM Distribution as in Step 6.

Step 4: enter the constants in the Constants sheet.

When using uniform distributions, it is recommended to calculate the range of the values and place it in a neighboring cell. The range is the lower limit subtracted from the upper limit. The range cell will reduce the number of elements in our formula: The range of equipment prices is in B8 = B7 - B6

And that of the spare part prices is in B13 = B12 - B11

	A	B	C
1	**Maintenance**		
2	Annual Rate	0.12	
3	Contract Years	3	
4			
5	**Equipment**		
6	Lower Limit	12,000	
7	Upper Limit	17,000	
8	Range	5,000	
9			
10	**Spare Parts**		
11	Lower Limit	1000	
12	Upper Limit	1400	
13	Range	400	
14			
15	**Assumptions**		
16	Maintenance is 12% of equipment cost		
17	Maintenance is for 3 years		

The last 2 lines are pure text. They state the assumptions we will use in the model. They serve to document our work:

a) The maintenance cost = the rate * the equipment cost
b) The maintenance is to be contracted for 3 years

Step 5: replace the constants (in the range B2:E2) with randomly or probabilistically generated values. All 4 input variables are uniformly distributed and can be generated using the **Uniform Distribution**. This generates equally likely prices between the ranges specified in the Constants sheet. For that, we need to use Excel's RAND() function. Later on, we will provide a detailed description of the use of the uniform distribution formula (In Part 2, see chapter that discusses modeling with the Uniform Distribution).

a) RAND() is a function that generates numbers between 0 and 1.

b) Any number generated by RAND() is equally likely to be generated. Another way of saying this is that the probability of getting any number between 0 and 1 is uniform.

c) RAND() does not use any arguments or specific parameters within the parentheses. Simply enter =RAND() in any cell or as part of a formula and Excel will generate a **pseudo-random** value for you.

Random or Pseudo-Random? These numbers are really "pseudo-random". Since they are generated by formulas, the result is "almost" random and they can be tested for that characteristic. We cannot generate **truly random numbers**. The numbers are therefore technically called **pseudo-random numbers**. However, we will use the term "random"

with that knowledge in mind. We will have more to say about random numbers in chapter 10.0.

To get a sample for the cost of the generator, we need a number that is randomly generated from the range 12,000 to 17,000. But RAND() gives us values from 0.00 to 1.00. How do we get such a value?

a) Calculate the **Range** = 17,000 - 12,000 = 5000.
b) **Generate** a number between 0 and 1.
c) **Scale** the numbers by multiplying RAND() by the range: = RAND() * 5000. Our generated numbers will now vary from 0 to 5000.
e) **Shift** the numbers by adding 12,000 or the lower limit to the result.
f) **Result**: our generated numbers will now vary uniformly from 12,000 to 17,000

The equipment cost varies from the lower limit = 12,000 to the upper limit = 17,000. Remember that the range was calculated in B8.

B2 = RAND() * the Range + the Lower Limit
B2 = RAND() * Constants!$B$8 + Constants!$B$6

This last formula reads as follows: multiply the generated random number (from 0 to 1) by the range in B8 (5000) to expand it from 0 to 5000. Then add the result to the lower limit found in B6 (12,000). The result is a randomly generated set of equipment costs varying from 12,000 to 17,000.

Hint: the use of absolute referencing of cells: we need to use absolute references for B8 and B6 (and the formulas of the next step).

To fix the address, use absolute reference by pressing F4 on each cell's name to ensure that all rows will point to the constant cells without changing their address. B8 becomes $B$8. We would thus be asking Excel to freeze or fix the column and the Row when copying.

Step 6: again, get a uniformly distributed value for the cost of the spare parts (for each year separately in cells C2, D2 and E2). The cost of spare parts in each year varies uniformly from 1000 to 1400. Each year might have a different cost. We need to draw 3 different samples, one for each year. Since the variation is the same for any of the 3 years, we will use the same constants for the 3 years in each of the 3 formulas. These are found in the range B11:B13 in the Constants sheet. Here are the 3 formulas for the 3 years in the second Row of the Runs sheet: C2 = RAND() * Constants!$B$13 + Constants!$B$11

D2 = RAND() * Constants!$B$13 + Constants!$B$11
E2 = RAND() * Constants!$B$13 + Constants!$B$11

Each formula above will generate a different uniform value between 1000 and 1400 because each execution will result in a different value of RAND().

There is no need to change F2 and G2 as formulated in Step 3. They are not to be

randomized as they depend on the above cells and will be recalculated whenever the input variables in B2, C2, D2 and E2 change.

Step 7: **to generate the "runs",** replicate Row 2 down to Row 1001. This would be a valid copy since we used absolute references in our formulas. Each of these rows will contain one run. Each Row will have the same formulation tested in Step 3. However, since each Row will have different instances of RAND() expressed in some of the formulas, the results will be different but based on the samples from the Uniform Distributions as specified above. Our model is now complete.

Press F9 a few times (to recalculate the worksheet) and you will see the whole range B2:G1001 changing values. Ask these questions:

a) Are the values within the ranges of the parameters?
b) Is each total in column G still correct?

Check the validity of the formulation.

In Step 3, we calculated the total cost for the equipment in Row 2 and placed it in G2. This is the result of the simulation for Run ID = 1. Since each Row below that had the same formulation, the result of the simulation is a set of 1000 cells in the range G2:G1001. Each cell represents a value generated by 1 run. Each run used values for the cost of the equipment and 3 years of spares sampled from different distributions. It is "as if" we purchased a piece of equipment and 3 years of spares, 1000 times and noted the total cost in the cells of G2:G1001.

Step 8: Copy the results range G1:G1001 from the Runs to the range A1:A1001 in the Results sheet. It is dependent on the formulas in the replicated rows. If we analyze it as it is, then each time we change anything in the sheet, the results in G2:G1001 will change.

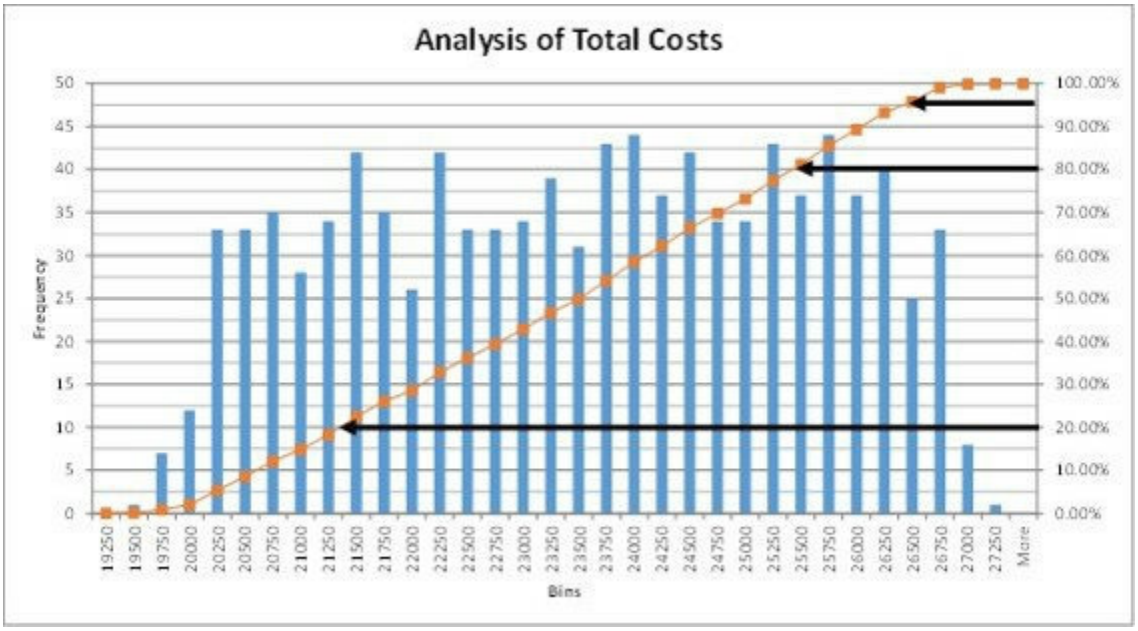
To avoid this problem, we need to copy the range G1:G1001 and paste it into the Results sheet "as values", i.e., without formulas. Here are the general steps:

a) Select and copy the range G1:G1001 from the Runs sheet (this includes the header).

b) Use **Paste Special** to paste **as values** the range starting in A1 in the Runs sheet. We now have A1:A1001 in the Runs sheet as a "static" set of total costs which do not include formulas. They are ready for analysis.

What are we going to do with these results?

Step 15: **analyze the results in the chart**. You now remember the requirement of your boss about pricing too high. The chart gives the answer.



To read the above combo chart, we can ask the following questions:

a) We want to be 80% sure of our value. For a total cost of 21,250, we have 20% of the simulation running below it. This is found by rising vertically from 21,250 until we hit the cumulative curve. We then go right across to the secondary axis (the right axis) and read the corresponding value which is 20%.

b) For a total cost of 25,500, we have somewhere between 78% and 81% as can be checked 80% of the simulation runs below it. This is found in the same manner as for the value 21,250. In order to be 80% safe that our price will be below 80% of all possible prices, we need to bid 25,500. (Of course, we will be risking being out priced with this value, but it is safe. Take your pick).

c) If the boss asks: "if we want 95% of all possibilities to be below our price, what will that price be?". Here, we select 95% on the right axis and draw a horizontal line to the left until it meets the cumulative curve. We then drop a vertical line down to the X-axis to find that value = 26,500.

Going back to the Precision issue or the required number of bins

In the above example, the number of bins was around 30. Given our range, this gives us increments of the total cost of around 250. The results can be shown on the chart where the values jump in increments of 250. The chart can answer our questions by increments of 250. If this is not suitable, i.e., if your boss wants results in increments of 500 or 100, you would need to change the suggested number of bins in C4 accordingly. For example, if we wish to have increments of 100, our range is 7664, so we need around 75 bins. If our required increment is larger, say 500, then we only need 7665/500 = 15. Remember that since the preparation of bins and the Analysis Toolpack is manual, this is not a dynamic entry. If you change the value of B4 from 30 to 15, nothing will change. When adjusting the results for a new increment size, you need to do the following: a) Establish a new “Final Size” in C6

b) Copy the formula in D3 (which uses C6) down to a value that is just above your maximum total cost

c) Repeat the **Histogram** and the **Descriptive Statistics** procedures using the new Bins range you prepared in step b.

Conclusion: the Monte Carlo Simulation process defined in the previous chapter and as per the example in this chapter is a solid basis for developing simulations. There will be more examples that clarify additional distributions, analysis and general tricks to use in simulation. They will all comply with the above process.

Extensions to the Model

Now that we have completed the model, we can vary the formulation. This requires a gaming spirit laced with creative business thinking:

a) Try different distributions that are more realistic than the Uniform Distribution.

b) What if we prepare a model that has a randomized maintenance rate? We can then introduce a distribution for the rate. But the rate is being applied over three ears. So, we can apply three different distributions, one for each rate in each year.

c) Our current model assumes the distribution of spare costs to be the same over the three years. But equipment usually consume more spare parts as they grow older. So, we can either change the ranges in the uniform distributions or apply a growth factor over the years.

d) Why are we restricted to 3 years? Maybe we can extend the model to additional years. But, how do we show our boss a model with optional 3, 4, 5 or 6 years? Here, you can use Excel tricks. In the Constants sheet, enter in a vertical set of cells the texts: “Year 4”, “Year 5” and “Year 6”. In each cell across, enter 0 or 1 where 0 signifies no modeling for that year. In your formulation, calculate the total spares and maintenance for 6 years. Multiply years 4, 5 and 6 with the above multipliers. This will ensure the formulation to include or exclude them.

e) If you decide to model 5 or 6 years, you may wish to include refurbishing costs such as replacement of major components. These will usually not be included under maintenance. Again, you can twist the model with the 0 / 1 trick to include them or not. For example, you can check the value of a specific cell containing RAND(). If it is < 0.5, you include refurbishing. If it is = or > 0.5, you do not.

f) As long as you have long enough rows in Excel, you can add formulations for additional equipment or even other costs such as licenses, transportation, *etc.* The end result will be in a single column, as in the simple initial model.

7.0 Frequency Tables, Relative and Cumulative Frequencies

While developing models for Monte Carlo Simulation several practices require an understanding of probability:

a) The generation of samples for the input variables.
b) The statistical analysis of the output or the results of the simulation.

Both practices require a working but not a theoretical knowledge of frequency tables and distributions. Both depend on knowledge of probability.

In this chapter we will address the second issue: the generation of frequency tables from raw data and the preparation of cumulative % frequency columns to use in the final analysis of results. Distributions will be addressed, mostly, in Part 2 of this book.

If you are happy with these concepts, you can skip to the next chapter.

Monte Carlo Simulation modeling can broadly be summarized as follows (although we will go through a more detailed ten step process in chapter 8.0):

a) The model will produce one column of raw data: our results. It will have thousands of entries.

b) The analyst will summarize the raw data in this column into a frequency table. This will be a more manageable table containing 10 to 30 categories (or bins).

c) From the frequency table, the analyst can then produce charts, generate conclusions and various statistical results.

A) Some Terminology

The Raw Data or Results: these are found in the column of results or raw data as the output of the simulation runs. (Usually found in the Model sheet and copied to the Results sheet). In a real example, these would be measurements or observations. Statisticians have ways to determine how many of these are needed for significant results. Invariably, the larger the number of raw results, the more significant the results. But in practice, this is costly. In Monte Carlo Simulation, it does not cost anything to increase the number of results, sometimes going into thousands. It would be a matter of running the model a few seconds longer in time.

The Raw Data or the results of the model will be numeric, mostly real, i.e., fractional or non-integer numbers. Monte Carlo Simulation models can also result in results that are integers such as the count of clients who prefer some product, defects in manufacture or number of successful interviews. The results will be in a column in the Results sheet.

It is this column that we need to convert into a frequency table.

Examples of models and their raw data:

a) A model that analyzes the purchase value of clients in a showroom.

b) Another that simulates a project and produces a large number of possible durations.

c) A model that simulates a production line where the events or the results are the number of defects in each item produced.

d) A model that simulates a petrol station generate results of the waiting time of each vehicle.

e) A simulation of a door to door salesman operation where the result is made up of the ratio of successes in a large number of days.

A Frequency Table (or Histogram) and its Brackets (or Bins): there is a confusion of terminology here essentially arising from the overlap between terms used for tables and charts.

a) Such tables in Excel are often referred to as Frequency Tables or Histograms
b) The charts generated from such tables are often referred to as Bar Charts or Histograms.

We shall therefore understand by the term "Histogram" either as a table (which can be produced by the Analysis Toolpack in Excel) or as a chart (which can be produced by the INSERT / CHARTS menu in Excel when producing bar charts).

So a frequency table, a histogram and a bar chart can be considered as interchangeable, the context making it clear whether we are talking about a chart or an Excel table.

A **Frequency Table** is a set of paired data (two columns) that summarize the raw data. The main reason for generating a frequency table is to summarize the thousands of raw data results into a manageable yet informative table.

The frequency table groups the results under different categories showing the counts the number of items (frequency) in each category. The first column in the table defines the categories, usually call bins. The second column defines the number of times or the count of items that occur in that category.

Examples of categories are applied to the list of results presented above:

a) Purchase value of clients in a showroom are grouped into brackets of $250 dollars each: 0 to 250, 251 to 500, 501 to 750, 751 to 1000 and so on.

b) A large number of possible project durations can be grouped in brackets of 5 days each: 20 to 25, 26 to 30, 31 to 35, 36 to 40, 41 and above.

c) The number of defects in each item produced are not many. In this case, we know before hand that we have 8 tests and so can have 0, 1, 2,.... 8 defects. These will be the brackets that summarize the large number of items.

d) The waiting time of each vehicle can be bracketed in minutes ranging between 0 to 2, 2 to 4, 4 to 6, 6 to 8, 8 and above. (See note on end values below).

e) The ratio of successes in a large number of days can be grouped between 0 to 20%, 20% to 40%, 40% to 60%, 60% to 80% and 80% to 100%.

As you can see, the number of rows (or brackets or bins) in a frequency table is usually in the order of "tens", i.e., from 10 to 30.

The Problem of End Points of Brackets or Bins: you will have also noticed that some of the examples above include overlapping end points. For example the waiting line example (d) has brackets from 0 to 2 minutes, 2 to 4 minutes and so on. In the model, we will be testing against such overlap. The implied brackets will be defined as: a) Vehicles with waiting times greater than 0 AND equal to or less than 2.

b) Vehicles with waiting times greater than 2 AND equal to or less than 4.
c) Vehicles with waiting times greater than 4 AND equal to or less than 6.
d) And so on.

Note: in this eBook, we will always consider the brackets to start by values greater than the lower limit and less than or equal to the upper limit. The examples above comply with this practice.

Methods for generating Histograms or Frequency Tables:

There are 5 ways you can use Excel to convert raw data into histograms or frequency tables:

1) The COUNTIFS() Function (to be presented in the next workout)
2) The FREQUENCY() Function
3) The Histogram entry in the Analysis Tool Pack
4) The Pivot Table
5) Free or paid for Add-ins that generate histograms.

In this eBook, we will only use the first two methods, COUNTIFS() and FREQUENCY(), for reasons to be discussed below. The remaining methods are usable but each one has its shortcomings, making COUNTIFS() a sure winner, most of the time. However, we will use FREQUENCY() in case our observations are integers.

We shall not present the other techniques as they are well documented in the literature.

B) Using COUNTIFS()

Our function has more than one pair of arguments:

COUNTIFS(range1, criteria1, range2, criteria2, range3, criteria3, …..)

Since we need to check if a specific value in the results column is within the lower and upper limit of each bin or bracket, we only need two pairs:

COUNTIFS(range1, criteria1, range2, criteria2)

Placing this function in a cell, it will count data items in each range (first argument in the pair) that meet the conditions in the corresponding criteria (second argument in the pair). It will then place the frequency count in that cell.

Although most examples assume that the ranges are different, there is no reason why you cannot use the same range, which is our case.

The **range1** and **range2** should always point to our results column, usually in Col A expressed as $A2:$2001 for 2000 data values.

The criteria should express the following checks on the bracket:

> than the lower limit
<= than the upper limit

At the end of "counting", each bin would have the exact number of events that occurred in the whole experiment.

The advantages of using COUNTIFS() over other histogram generating techniques are:

a) It is dynamic. If you change the output data or change the bins, the frequency tables get updated directly. Other methods require you to start all over again.

b) Bins are defined by the analyst and not automatically by Excel.

c) Bins can be irregular, i.e., having different ranges. Some of the other methods generate their own bins.

d) It is much easier to use once standardized, as we shall show in the next workout.

C) Using FREQUENCY()

This function is easy to use. It is dynamic in that it will change as the data changes (as opposed to the static output of the Analysis Toolpack). It is an array function which makes it easier to copy.

However, it has one restriction which favors COUNTIFS() in such cases. FREQUENCY() can only count integers. In an output column with a fractional format, using FREQUENCY() will give misleading results. In the next workout, we will show how to use FREQUENCY in the case of integers.

Note: in some cases, the calculations of the randomizing functions will result in fractional output. However, we can turn these into integers without losing information. For example, assuming you are modeling the arrival of trucks or vehicles to a petrol station. The output will be a time value. However, we do not lose information if we truncate the fractional part and resort to the use of FREQUENCY() function to prepare a frequency table that shows us the distribution of arrival times, in minutes.

Workout 2: Generate Frequency Tables using COUNTIFS()

Purpose: to show how we can use COUNTIFS() with two criteria expressed applied to a single range (our results). second purpose is to provide you with a standard frequency table generating method that will be used throughout the eBook. We will also show how to use FREQUENCY() for the same purpose in the case of output with integer values.

COUNTIFS() has two arguments. This capture is from Excel's formula entry dialog box. It shows how to use COUNTIFS() to find the number of times we have heights in the range A2:A1001 that are > 150 AND <= 160:

You see, even Microsoft makes the mistake of calling the second argument "criteria". This is confusing as we are only allowed one criterion per pair.

This workout shows you how to avoid entry of constants in criteria1 and criteria2.

Step 1: we need data, so open the workbook in the Workouts Folder called **Generate Frequency Tables - DATA**. It has one sheet with data. Save it under any name of your choice. In the Workouts Folder there is a fully solved model called **Generate Frequency Tables with COUNTIFS()**. You can check this workout against it.

Step 2: we need to prepare the bins. To prepare the bins, we need to come up with some estimates. This is the Bin Preparation Area. If you know what your categories or bins are, there is no need for this range.

Enter the labels in Col B as shown below:

	A	B	C
1	Heights	Min	
2	171.9	Max	
3	154.8	Range	
4	165.9	Bins	
5	151.4	Bin Size	
6	160.1	Actual Size	

The labels are yellow to denote constants or parameters.

Step 3: Enter the following formulas and values. The purpose is to find the size of the bins or the brackets:

C1 = MIN(A2:A2001)
C2 = MAX(A2:A2001)
C3 = C2 - C1
C4 = 20
C5 = C3 / C20
C6 = a value that you enter.

Note: that we may be using the range A2:A2001 in several places in the model. It might be easier to define the name of this range (under FORMULAS / DEFINED NAMES / NAME MANAGER). Say you define it as "**results**", you would then be able to use it in the formulas as follows: =MIN(results) and so on.

In our case, the values are:

	A	B	C
1	Heights	Min	138.8
2	171.9	Max	181.9
3	154.8	Range	43.0
4	165.9	Bins	20
5	151.4	Bin Size	2.2
6	160.1	Actual Size	2.0

C1 = 138.8 which comes from the data

C2 = 181.9 which comes from the data

C3 = 43.0 which is the difference between the above two values

C4 = 20, is manually entered and is the number of bins you wish to have. This can be from 10 to 30, depending on the data you are using. The larger the number of bins, the more specific your results will be but they will be more wieldy to handle.

C5 = 2.2 which is 43.0 / 20, this is a calculated but approximate bin size

C6 = 2 which is another value that you enter manually. It is the final size of the bin. It will be used later to "space" the bins in Col E.

Step 4: now that you have decided on the final bin size, we have to create the bin table. This is made up of three columns:

Col E contains the bins, or the range of the brackets
Col F will contain the calculated frequencies using COUNTIFS()
Col G contains the cumulative % frequency to be calculate below

We skipped Col D just to give the worksheet better visibility.

E1 = Bin
F1 = Freq

G1 = Cum % Freq

Step 5: setup the bins in the required range.

Since the calculated value of the bins was 2.2 and we decided it to be 2.0, we will have more than 20 bins. There is no need to crack your head predicting the size of the bin column. This can be done when filling it as follows: a) Enter the starting value in E2 which must be just a bit less than the MIN() in C1. In our case, this can be 137. If you prefer even numbers (as these values will show in the chart, then choose 138). You can also use the ROUNDDOWN() formula as follows: E2 = =ROUNDDOWN(C1,0) which drops off the decimal digits of the MIN and ensures that it is less or equal to it.

b) E3 = E2 + $C$6 where C6 is the final bin size expressed in absolute reference because we want to copy it downwards.

c) Copy E3 downwards. You will not know when to stop as Excel does not show you the values. But you need to stop when the entry in Col E reaches a value just above 181.9, the MAX, i.e., 182. This is dodgy. You may have to drag the + on the highlighted cell downwards a few times till you reach that value.

It turns out that in our case, the last bin would be in E29. Here is an image of the top part of the bins (with still empty frequency columns):

E	F	G
Bin	Freq	Cum % Freq
137.0		
139.0		
141.0		
143.0		
145.0		
147.0		
149.0		
151.0		
153.0		
155.0		

Step 6: we now use COUNTIFS() in Col E to count the number of occurrences of each item in our results within specific brackets. The table would be read as follows:

E2 contains the count of results > 0 and <= 137.0
E3 contains the count of results > 137.0 and <= 139.0
E4 contains the count of results > 139.0 and <= 141.0
And so on

Effectively, E2 should always = 0 since we chose 137 to be lower than the minimum, 138.8 which means there are no results in that range.

Since we have two conditions (as per the above list), we need two pairs of arguments in

COUTIFS():

E2 = COUNTIFS(**$A$2:$A$2001, ">"&E1**, $A$2:$A$2001, "<="&E2)

The two range arguments A2:A2001 are. They are expressed as absolute references since we need to copy E2 down to the bottom of the bin column.

COUNTIFS() allows you to place two criteria on the same range. The first criterion is **">"&E1**. The ampersand allows COUNTIFS() to check if there are any values larger than E1 (which is zero, on this row). The second criterion is **"<="&E2** which checks if there are any values less or equal to the upper limit of the bracket, in this case 137.

Note: the two conditions are a logical AND which means a value has to satisfy both criteria to be counted. In effect, as we copy E2 down, we will be able to get a count of the results in each bin.

Note: the help file in Excel has very good examples of the use of COUNTIFS().

Step 6: copy E2 down to E29 (or whatever is the location of the maximum bin). COUNTIFS() will be automatically calculated.

Here is the top part of the sheet before calculating the Cum % Frequency (as will do in the next workout):

	A	B	C	D	E	F	G
1	Heights	Min	138.8		Bin	Freq	Cum % Freq
2	171.9	Max	181.9		137.0	0	
3	154.8	Range	43.0		139.0	1	
4	165.9	Bins	20		141.0	2	
5	151.4	Bin Size	2.2		143.0	1	
6	160.1	Actual Size	2.0		145.0	12	
7	153.7				147.0	21	
8	154.5				149.0	37	
9	159.6				151.0	66	
10	151.7				153.0	105	
11	161.5				155.0	182	

Step 6: save the workbook as we will need it in the next section.

Workout 3: Generate Frequency Tables using FREQUENCY()

As explained earlier, we need to use integer values. We will use the same values we imported from the **Generate Frequency Tables - DATA** workbook. The data is presently in the range A1:A2001 in the COUNTIFS() sheet in the above workbook.

Step 1: open the workbook that you had prepared for COUNTIFS().

Step 2: right click on the COUNTIFS() sheet and copy as a sheet in a new workbook.

Rename the sheet FREQUENCY(). Give the new workbook the name **Generate Frequency Tables with FREQUENCY().**

Step 3: prepare the data to be in integer format by entering the following formula in A2 in the new sheet and then copying it down to A2001:

A2 = INT('COUNTIFS()'!A2)

Note that you can also use TRUNCATE() and other rounding functions to achieve the same purpose.

Also note that the formatting of various ranges in the earlier sheet was retained. To avoid confusion, reformat the new ranges to show no fractions:

The Heights column or the output data (A2:A2001)
The bins data (C2:C6)
The bins (E2:E29)

The bin data will remain the same since we are using the same data.

Step 3: since FREQUENCY() is an array function, we have to use a new procedure for entering it. We need to define the range it will be in, namely, F2:F29.

a) Select the range F2:F29 and press F2. This will allow you to enter a function in F2.

b) Enter the function as follows:

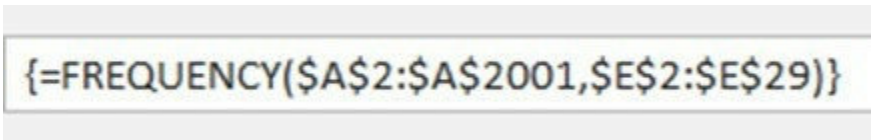
=FREQUENCY($A$2:$A$2001,$E$2:$E$29)

The function has two arguments: A2:A2001 is the data to be analyzed and E2:E29 defines the bins that will be used to prepare the frequency table.

In both cases, we will use the absolute reference since Excel will copy these down throughout the range F2:F29.

c) Here is the difference. With F2:F29 highlighted and the formula entered, simply press CONTROL+SHIFT+ENTER. Excel will spread the formula downwards and show you the frequency counts.

Note: you cannot edit individual cells in the range F2:F29. To recognize an array function, you will see that all such entries will be surrounded by curly braces:



Step 4: we should expect to get the same table as using COUNTIFS(). Let us test that statement. Enter in H2 in the FREQUENCY() table the following test:

H2 = 'COUNTIFS()'!F2 - 'FREQUENCY()'!F2
Copy H2 to I2

This simply subtracts the frequency counts and the cumulative % in the two sheets from one another. Copy H2:I2 down to H29:I2. You will see a large number of discrepancies. The reason is that rounding has changed the 'mapping' of the output results, sending them to different bins in the two sheets depending on the rounding.

Workout 4: Prepare Cumulative % Frequency Tables

Purpose: to produce the cumulative % frequency without having to prepare a frequency % column.

Why do we need the cumulative % frequency? By adding the proportion of "bars" or frequency counts that are found below or above a particular x value, we will be able to state such conclusions as:

60% of all observations are below X
55% of all observations are between X and Y
30% of all observations have values higher than X

More examples will be given at the end of this workout when we produce the Pareto chart.

Although the frequency % will be used in later workouts to define probability, we do not actually need it to prepare the cumulative % frequency. We can compute it directly.

Definition: the cumulative frequency is the sum of the frequency counts that as accumulated as we go from the first bin to a specific but larger bin. It follows that the cumulative % frequency is the same as the cumulative frequency, but it is divided by the total frequency count.

this method of preparing cumulative values will be used throughout our eBook. Most examples in the literature rely on using two formulas: one in the first Row and another in the rest of the rows. The following method allows you to use one formula in all the rows. It exploits the combined use of absolute and relative addresses in the SUM() function.

Step 1: open the workbook that you developed in the previous workout. In the Workouts Folder there is a fully solved model called **Generate Frequency Tables with COUNTIFS()**. You can check this workout against it.

Step 2: G2 = SUM($F$2:F2) / SUM($F$2:$F$29)

This sums the range F2:F2 which is simply 0. The trick is that the start of the range is fixed on $F$2 which is 0. The end of the range is relative. In this case, it is also F2, so the SUM($B$2:B2) = 0.

a) Copy the formula in G2 down to G29. $F$2 remains pointing to F2 while the second F2 changes as the rows increase: F3, F4, F5 *etc.* For each Row, the sum is that of all

frequencies before that Row and the entry in that Row.

E	F	G
Bin	Freq	Cum % Freq
137.0	0	0.00
139.0	1	0.00
141.0	2	0.00
143.0	1	0.00
145.0	12	0.01
147.0	21	0.02
149.0	37	0.04
151.0	66	0.07
153.0	105	0.12

Examine G8. It is = SUM($F$2:F8) / SUM($F$2:$F$29) = 0+1+2+1=12+21+37 = 74. This is the sum of the first 7 frequency counts or bins from G2:Gin D2, D3 and D4. Divide 74 by the total count, which is 2000 and you get 74/2000=0.037 which is rounded by Excel to 0.04. (If you format the cell to show 3 decimal digits, you will get 0.037).

As a check, the last entry in any cumulative % frequency column should be 1 as the total count / total count = 1:

175.0	20	0.99
177.0	13	1.00
179.0	1	1.00
181.0	2	1.00
183.0	1	1.00
185.0	0	1.00

Notice two characteristics of cumulative frequencies:

a) Cumulative values never decrease. (Mathematicians call this a monotonous series).

b) Two values can be the same if the frequency count in the second is zero.

In the next chapter, we will discuss frequency tables as the basis of probability and hence, the practice of sampling from distribution, the heart of Monte Carlo Simulation.

Workout 5: Plot a Pareto Chart: Freq Count and Cum % Freq

Purpose: to prepare the Pareto chart that contains the frequencies and the cumulative % frequencies. Some useful charting tips will be presented. We will also show how to extract analytic information from the chart. Most of the MCS models will be using the

Pareto Chart.

Why is it called **Pareto**? Because Wilfred Pareto was an Italian economist (19th Century) who stipulated that most of the wealth of a nation is in the hands of a few people. Today, we call this, the 80/20 split. Statements such as "80% of our revenues come from 20% of our clients" or "80% of defects come from 20% of our products" are all instances of the Pareto Law. What makes these statements weird is that the two percentages seem to add whereas they come from different populations. 80% is a portion of revenue while 20% is a portion of clients. This is the main logic of the chart, combining two plots that can crisscross at certain points to read off two different percentages or values. Interpretation examples will be extracted from the Heights workout developed previously.

Note: the chart we will be setting up is known in Excel as a combo chart because it contains two data series (on the vertical axis). There are two ways of setting up such a chart. One will be chosen for reasons to be given below.

Step 1: open the workbook that you developed in the previous workout. In the Workouts Folder there is a fully solved model called **Generate Frequency Tables with COUNTIFS()**. You can check this workout against it.

Step 2: highlight the ranges Bins, Frequency and Cum % Freq (including the headers but not including the totals Row).

Step 3: most standard charts in Excel, such as the line and the bar chart, require only one data series on the Y-Axis while the X-Axis is shown as a sequence.

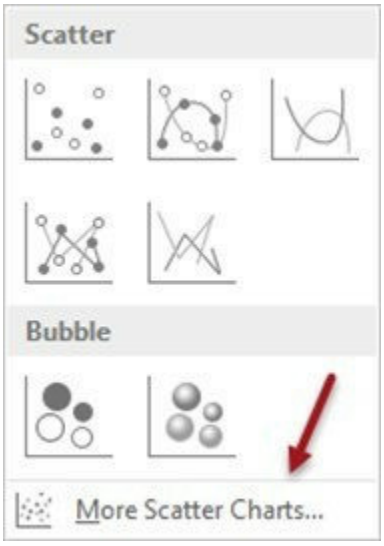
Our charts will mostly be of the type: **Scatter Diagram**. Another name for this would be the x-y plot. Scatter diagrams use a numeric data series from our workbook for the X-axis, or the horizontal axis. In our case, we will be plotting the bins on the X-axis while assigning the Frequency Count and the Cumulative % Frequency to two different vertical axis data series.

Our range is E1:G29. When you plot two data series, Excel calls this chart, a **combo** chart.

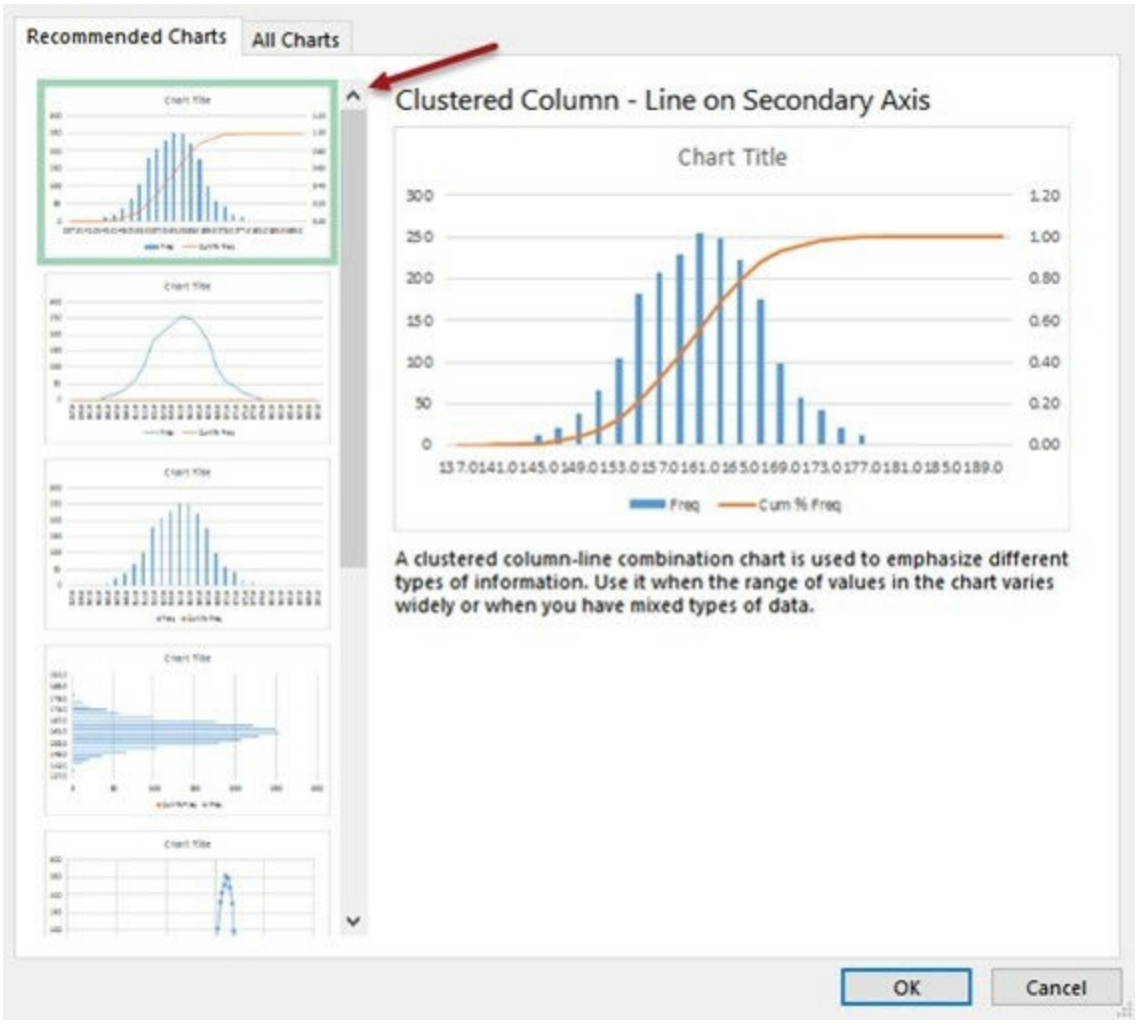
Two weird Excel behaviors:

a) Always select the SCATTER Diagram as shown below:

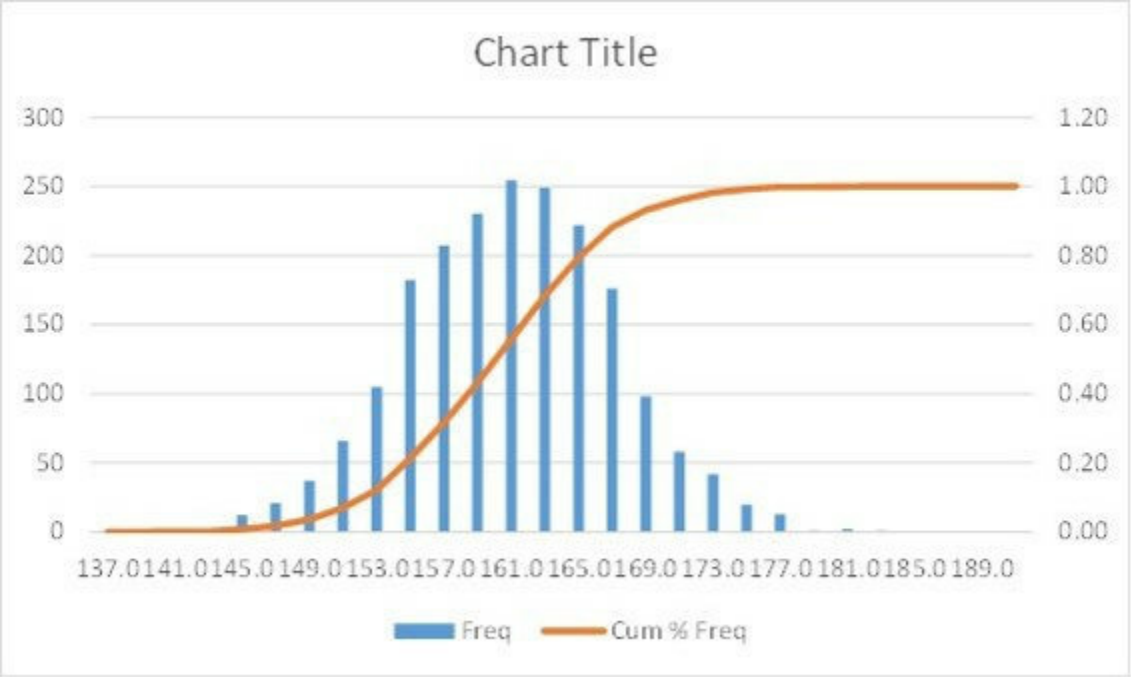
b) insert the SCATTER DIAGRAM as shown, you will get:



c) Once you select MORE SCATTER CHARTS and select the RECOMMENDED CHARTS tab, you will get:



d) Select the CLUSTERED COLUMN-LINE COMBINATION.



This type of chart will automatically place the Freq data series as a bar chart and the cum % frequency data series as a line chart. This will be ideal for "reading off" our conclusions as shown below.

Warning 1: do not use the COMBO option under the ALL CHARTS tab as Excel will understand that you want to plot 3 data series on the vertical axis. That will not be a scatter diagram (x-y plot).

Warning 2: in the above chart, you can see there two wide but blank (non-informative) spaces to the right and the left of the main plots.

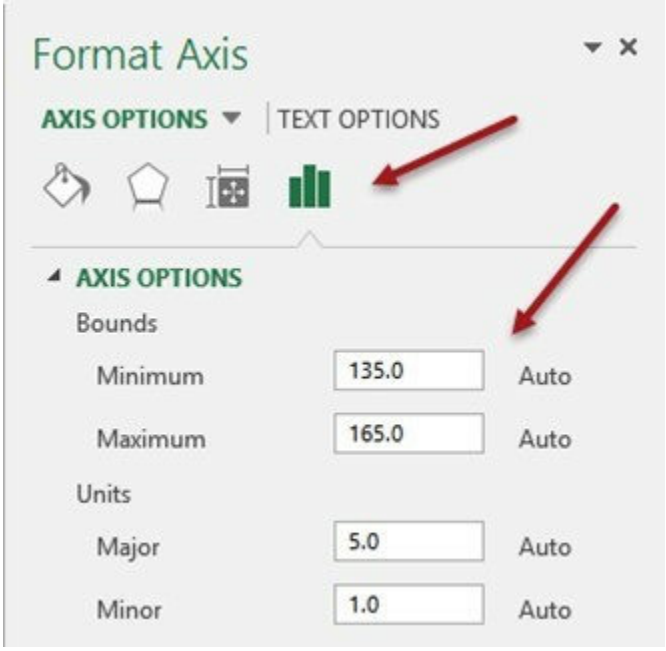
The selections we followed **do not allow you to scale the horizontal axis** (i.e., change the minimum and the maximum data). This is useful when we get blanks on the left and right side of the chart. Reducing the size of the horizontal axis will give you a visually more useful chart.

However, they do in a **regular single data series** scatter diagram. You can edit the horizontal scale to define the minimum and maximum values to be shown.

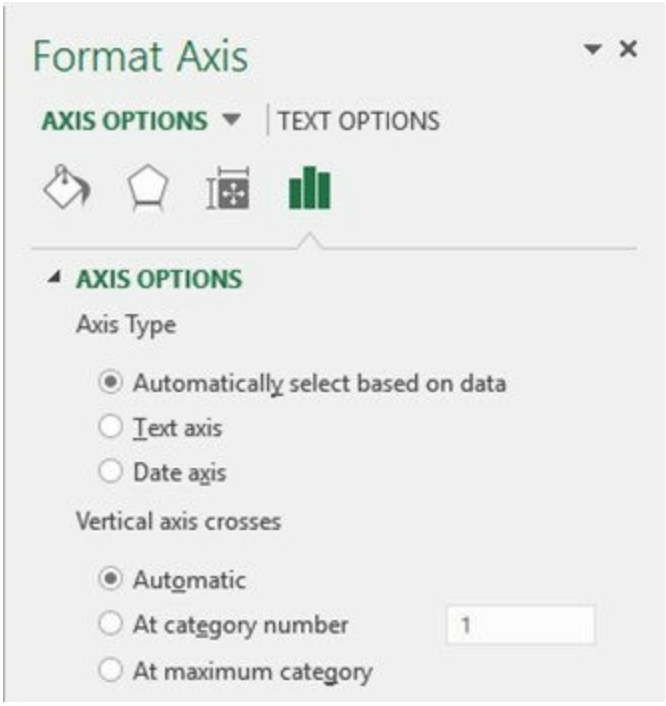
a) Right click on the X-Axis and select FORMAT AXIS

b) Select the AXIS OPTIONS

c) Enter the minimum and maximum values on the X-axis (our bins) that will restrict the chart to the area with significant information:



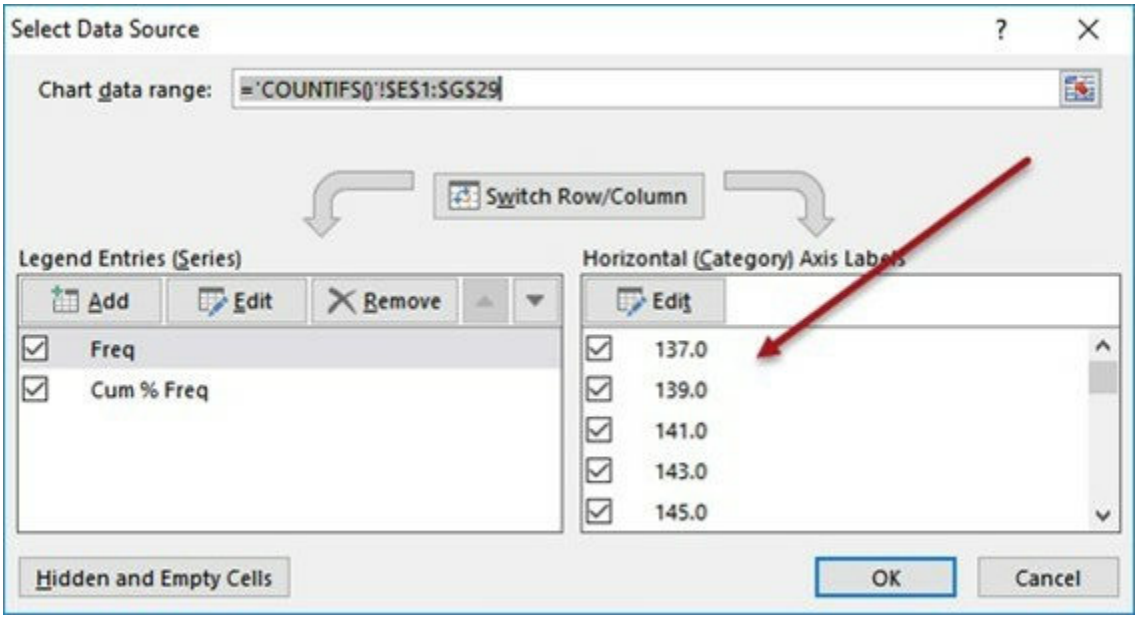
This facility is not available to our combo as we selected it. If you follow steps a) and b) above for our chart, you will get the following:



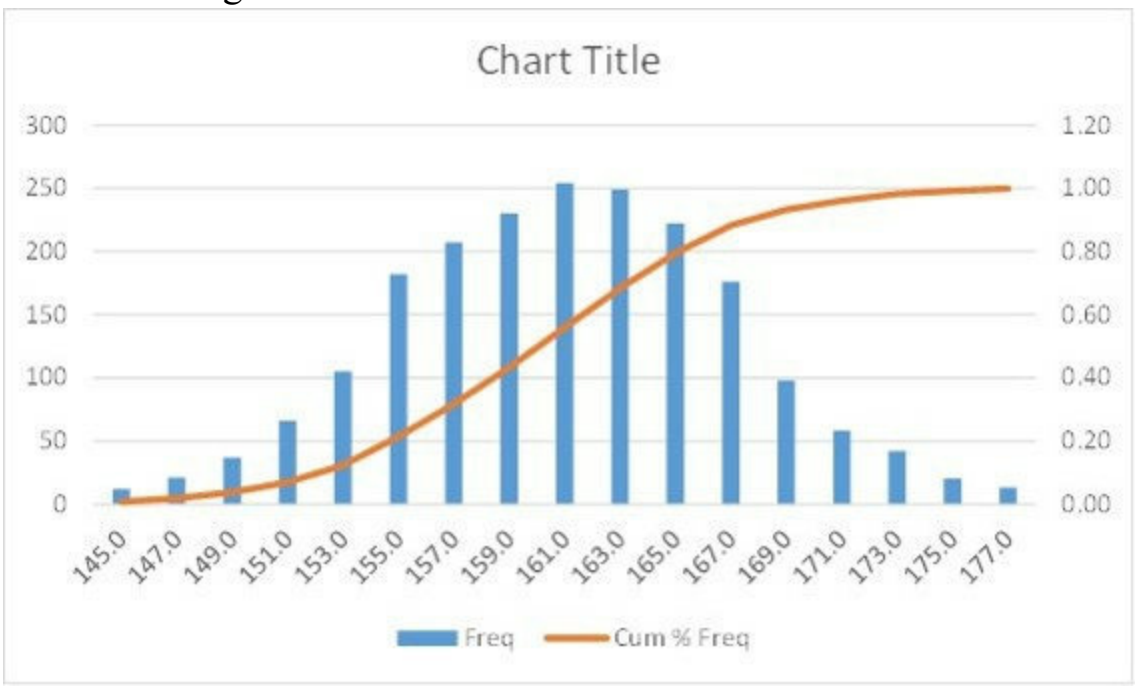
The minimum and maximum fields shown for regular plots are not available anymore.

Workaround to adjust the scale in our Combo Chart:

Right click anywhere in the chart and select the SELECT DATA item OR click anywhere the chart and select the DESIGN / SELECT DATA. In either case, you will get this dialog box:



You can see that the horizontal axis values have been check (included) in full. Remove those on the extreme right and left by unchecking their boxes and press OK. In our case, let us remove 4 points from the bottom values and 7 from the top values of the horizontal axis. We will get this chart:



This is much clearer and more usable.

Step 4: manage the chart title. it is recommended that we do not enter a direct text into the title of the chart. It is better to the title as dynamically linked to a cell value. By placing the title in a cell and linking it to the chart, you will be able to include more information such as numeric values. For example, you can define the title as “Analysis of Acceptance Sampling for a = 5%” where the 5% is dynamic and taken from another cell in your sheet.

For our case,

a) Enter the text “Combined Count and % Cum” in any cell in the worksheet, say in

I1.

b) Click on the chart title.

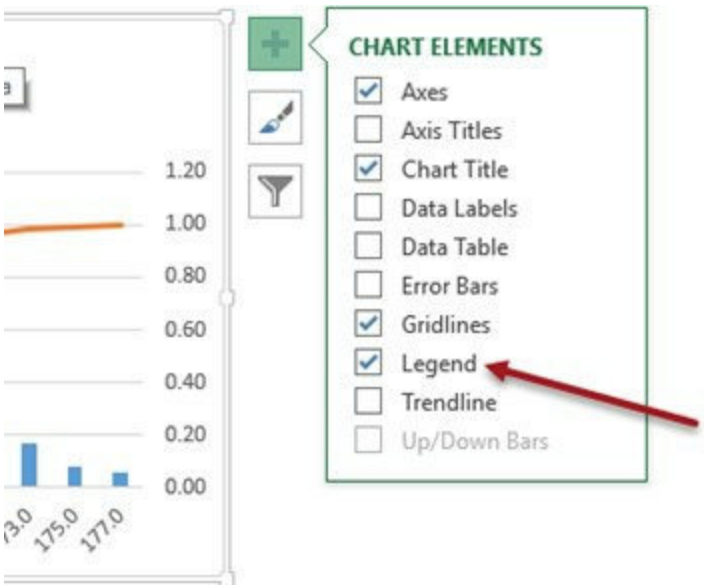
c) Type the equal sign (=) into the Formula bar (not in the title).

d) Select the cell that contains the text you want to place into the chart title.

e) Press Enter

Step 5: you may find it more natural to read the legend just below the title. If you do, follow this procedure:

a) Click anywhere on the chart and you will get a side tab:



b) Click on the Legend and you will get a drop down list from which you can select the "Top" option.

This will now be our chart:

How to read the Pareto Chart:

The bar plot show the frequencies of heights in our company. The orange plot shows the cumulative % frequencies, i.e., the sum of the bars to the right divided by the total population (2000 in this case).

Examples:

a) 80% of employees have a height lower than 165.0 cm. This is read by looking for 0.80 on the right axis, drawing a line to the left until it meets the cumulative plot. Then read the x value of the frequency that crosses the cumulative plot. This is 165.

b) Similarly, you can say that 20% of our employees are taller than 165.

c) What is the proportion of employees whose height is between 157 and 167? The frequency bar for 157 crosses the cumulative plot at around 17%. The frequency bar for 167 crosses the cumulative plot at around 90% (even if it does not cross, you can extend the bar upwards till it meets the orange plot). This means that 90% - 17% = 73% of our employees have heights between 157 and 167.

d) Suppose the 2000 samples were taken from a simulation of a project whose costs were our output. We are proposing this project. We can then say that if we choose 169,000 USD (for the 169 bar), we will be 90% of the time right in our estimate. The higher the value, the better the estimate. Of course, the higher the value, the more likely we will not get the project approved. Here is where a project manager can leverage his or her risks based on intelligent estimates.

Final note: the above information is already found in the table E1:G29. You can easily see case a) numerically:

Bin	Freq	Cum % Freq
137.0	0	0.00
139.0	1	0.00
141.0	2	0.00
143.0	1	0.00
145.0	12	0.01
147.0	21	0.02
149.0	37	0.04
151.0	66	0.07
153.0	105	0.12
155.0	182	0.21
157.0	207	0.32
159.0	230	0.43
161.0	254	0.56
163.0	249	0.68
165.0	222	0.79
167.0	176	0.88
169.0	98	0.93
171.0	58	0.96

You can see that 0.79 (nearly 80%) is the point where the height of 165 crosses. However, in many cases, visually analyzing the chart could provide faster conclusions.

We shall use the techniques in this chapter in most of our workouts. We will therefore not detail them in the coming workouts but assume you that you have gone through the above.

Workout 6: Generate Descriptive Statistics

Purpose: this is not really a workout. We leave that to the fully detailed and statistically intensive Appendix in Chapter 15.0.

Total	
Mean	23,394
Standard Error	62
Median	23,516
Mode	#N/A
Standard Deviation	1,975
Sample Variance	3,902,235
Kurtosis	-1.15
Skewness	-0.08
Range	7,664
Minimum	19,447
Maximum	27,111
Sum	23,393,644
Count	1000
Confidence Level(95.0%)	123

This is just a reminder that the following can be done:

a) Descriptive Statistics is a set of to tools available through a built in Excel Add In. Review Chapter 5.0 where one of the middle Sections shows you how to enable the Add In.

b) Descriptive Statistics are poorly formatted as produced by the Analysis Toolpack. Review the General Model Template in the Templates Folder. It has a formatted range for the Descriptive Statistics. Use the formatting brush to copy such a format if you need to.

8.0 The Monte Carlo Simulation Process

The process presented in this chapter starts after you have conducted a situational analysis and analyzed your problem. When you are ready for modeling the Monte Carlo Simulation, you can follow these practical stage as we did in the previous chapter.

Stage 1: **ready the workbook** which should consist of the following worksheets: Model, Runs, Results and Constants. (A template is included in the downloaded zipped file (see chapter 5.0).

a) **The Model** sheet will most contain the main formulation. In some cases, we might skip this sheet if the model or formulation can be expressed in one Row. The first Row below the header of the next sheet (Runs) can then be used as the formulation. We might sometimes include constants in the Model sheet if we need them to be more accessible when editing formulas. We might even include some output results if we need to monitor them while testing.

b) **The Runs** sheet contains the simulation runs or the duplicated rows. In most cases, we will restrict the runs to 1000 or so. In real life, you may need to go way beyond that, say 10,000. As mentioned above, if the formulation can be expressed in one Row, then Row 2 of the Runs sheet will be the model or the formulation and the lower rows will be the replicated runs. The Runs sheet may also include the Sub-Runs if the model requires them (see chapter 13.0).

c) **The Results** sheet contains the statistical analysis of the runs. It will also include charts.

d) **The Constants** sheet contains all the parameters that you might need when sampling as well as some constants such as rates, number of years, initial conditions, *etc.* It is good practice to include documentation remarks in this sheet as a guide for you or your reviewers.

Stage 2: **develop the formulation in the Model sheet**. The input variables of the formulation (or the first Row in the Runs sheet) should be setup with fixed values. These will be used for testing as in step 5. At a later stage, we would replace them with the randomized or sampled variables.

Stage 3: **identify the input variables and the output cells**: these should be setup in the model in clearly marked cells to make them easy to replicate as in Stage 8. We tend to follow the color code where input cells are shown with green headers while output cells have a blue header.

Stage 4: **define and enter the constants in the model** into the Constants sheet.

Stage 5: **verify the validity of the formulation using the fixed input values**: this step

ensures that your black box is working well. Enter different input values manually and verify that you are getting the correct results (outputs).

Stage 6: **for each input variable, determine the distribution that best describes its behavior**. In Part 2 of this eBook, we will be analyzing various distributions in depth. Once a distribution is determined, define the **parameters** to be used when sampling its using Excel's formulas. These should be setup on the Constants sheet.

Stage 7: **enter the formulas for the distribution functions**: in the input variables (cells), replace the fixed (or test) values entered earlier by formulas that sample specific distributions to generate randomized samples. Make sure that the formulation will still give valid results.

Stage 8: **determine the number of runs and replicate the formulation Row accordingly**: the number of runs is a sample taken from all possible runs or combinations of input variables. There is a benefit in having a large number of runs. Even though a low number (say in the 100s or few thousands) will take less time to setup and run, a higher number will provide more confidence in the results. In chapter 15.0 we will discuss the relationship between our confidence in the results and the size of the sample or the number of runs.

a) Decide on the number of simulation runs for this model and reserve as many rows starting with Row 2 in the Runs sheet (since Row 1 is the header).

b) When Row 2 is setup, tested and plugged with the randomized sampling functions, replicate it downwards as many times as you have simulation runs. (Ensure that you use absolute reference so that the replication will be valid).

c) Press F9 to test that the input variables and the related output cells change as expected.

Each Row will have one or more cells that provide you with the result of the formulation.

Stage 9: **analyze the output**: the analysis will consist of the following general steps:

a) Copy the full output column from the Runs sheet into the Results sheet: either as frozen values or as copies of formulas from the Runs sheet. This will depend on the nature of analysis that you require. Freezing the values allows you to return to your analysis and manipulate the charts and tables. Keeping the values dynamic allows you to view the results as and when you press F9, i.e., for different random samples. Examples will be given of both cases.

b) Prepare a frequency table from the new output column(s) in the Runs sheet. Use the COUNTIFS() function as we presented it in Chapter 6.0.

c) Prepare a chart with two components: a bar chart representing the frequencies in the histogram and a line showing the cumulative values superposed over the bar chart. (Pareto). Again, this was presented in Chapter 6.0.

d) Generate descriptive statistics for the output column. Check the Appendix in Chapter 15.0.

e) Generate specific results dependent on the model in question.

Stage 10: **tune the model**: based on the results of the analysis, you may wish to vary the following:

a) Change the constants
b) Change the parameters of the distributions
c) Use different distributions
d) Change the computational procedures of the formulation

For each of the examples in this eBook we will be following the above 10 stage process (with local variations if needed).

9.0 From Frequency Tables to Probability Distributions

Again, if you are happy with the basics of probability distributions and how to sample them, you can skip this chapter.

Otherwise, I can already hear you groaning, remembering those tedious hours where your professor scribbled endless formulas. And those horrible expressions counting the number of combinations or permutations for seating your friends over dinner. We will not go into any of that. We will restrict our discussion in this chapter to the clarification of two fundamental principles: a) The definition of probability and how it is equivalent to the relative frequency. Why? As an extension of frequency tables presented in chapter 6.0, the concept of probability (relative frequency) is crucial for the understanding of random number generation and randomized sampling.

b) The use of probability distributions, cumulative distributions and inverse distributions in Monte Carlo Simulation. These functions are at the heart of simulation. Without them, we cannot "pretend" that we have 10,000 instances of a sale or a project costing, or a schedule or a production cycle.

In an experiment consisting of several observations (or measurements or events), the calculation of the probability of an event requires us to know two items:

a) The number of times the event we are interested in occurs (its frequency)
b) The number of possible events in the experiment or the total frequency count.

Definition: probability = the number of times a specific event takes place divided by the total number of events in the experiment.

$$\text{Probability} = \frac{\text{Number of Times an Event Takes Place}}{\text{Total Number of Possible Event}}$$

Probability is also called the **expected value** of an event. The logic behind this term is that the probability of an event happening is a measure of how often we expect it to happen in the future (always expressed as a percentage or a ratio). Statisticians like to use fancy names. The two terms are also equivalent to: the **average** of an event happening.

Example 1: **dice**

What is the probability of throwing a 5 using a single die. Using the other terms: what is the expected value of getting a 5 OR what is the average number of times we get 5 after throwing the die a large number of times? Each number on the face of a die is an event, so we have a total of 6 "possible" events. Since 5 only appears in one of the events and

the total number of events is 6, the probability of getting a 5 is 1/6. This is the expected value or the average and it is = 16.66%.

If you toss a die a million times, 166,666 times, you will get a 5.

Example 2: **a deck of cards**

a) The probability of drawing a king = the number of times the king event can take place (or 4) divided by the total number of events (or 52) = 4/52 = 1/13 = 7.69%.

b) The probability of drawing a red card = 26/52 = 50%.

c) The probability of drawing a club = 13/52 = 25%.

Example 3: birthdays

a) The probability of someone's birthday falling in May is 1/12.

b) The probability of someone's birthday falling on a Friday is 1/7.

c) The probability of someone's birthday falling in summer is 3/12 = 1/4.

Probability is a number between 0 and 1. It is often stated as a percentage. In this eBook, we will start by using percentages and slowly move to the decimal form which is a value between 0.00 and 1.00.

Certainty: is when the probability = 100% or when the number of times an event can take place is the same as the total number of events. The probability of your birthday falling in any month = 12/12.

Impossibility: is when the probability = 0% or when there are no events corresponding to what you are measuring the likelihood of. If we toss two dice, there will be 6 * 6 possibilities. The probability of tossing two dice whose sum is 14 = 0 / 36.

Complementing probability: if the probability of something happening is 60% then the probability of it not happening = 100% - 60% = 40%. If the probability of finding a manufactured rod longer than 10.34 cm is 55% then the probability of finding one that is shorter than or equal to 10.34 is 45%.

A) Mathematical vs. Estimated Probability

So far, we have used the mathematical definition of probability where the total population was clearly defined: a deck of cards, the 6 sides of a die, *etc.* We are going to use probability in the operational world where we cannot be so definitive about events and populations.

Let us go back to a day in your teens. You come to the table for dinner and three persons give an opinion of your punctuality:

a) Your father says: "You've never come to dinner on time"
b) Your mother takes your side and says: "That's not fair. Don't listen to him darling, you've always been on time".

c) Your brother who is a pragmatic person and who wishes to contradict both parents, says to you: "I think both of them are wrong. You frequently come on time".

Since there is no deck, die or calendar, we cannot find the two components of probability: the number of times you've come on time and the total number of dinners you've had with your parents. They are implied in the statements. Let us pretend that we know a few numbers and make the implied events explicit: a) Your father says: out of the last 100 dinners, you have not been on time once which signifies a value of 0 for the event over 100.

b) Your mother says: out of the 100 dinners, you've been on time 100 times.
c) Your brother says: I guess out of the 100 dinners, you've been on time around 80 times. (Why not 75?)

No one can remember 100 dinners but we assume some large total population of dinners. Your boss tells you: "Based on our previous work with this client, I feel we are not going to get the contract". Somewhere in his mind there is a total population of times we bid for contracts (total events) and a number of times we did not get contracts (the number of events). Even if there is no such “image”, there is a perception of a large number of proposals. Your boss is computing the probability ratio, figuratively, qualitatively, intuitively but definitely not counting actual occurrences.

There is nothing wrong with this approach as long as the guess or estimate is educated. I would believe you if you wrote me and said: "most authors I've written to do not respond". You do not need to have written to 100 authors but you have intuitively calculated the probability and expressed it in qualitative terms. I would not believe you if you had said: All Frenchmen like French fries.

Why are we doing this? Because in business and industry, most of the time, we resort to measuring the likelihood of events when we neither know the number of times an event took place nor the actual size of the population. We assume a definite "virtual" value. We use "inference" or "educated estimates" rather than mathematical counts. Examples: Next season, our Brand B will have 25% of the market share

The odds of the HR manager refusing this applicant are 1:7
There is a 50/50 chance of winning this contract.

We will try to resolve such issues using Monte Carlo Simulation.

B) Probability and % Frequency

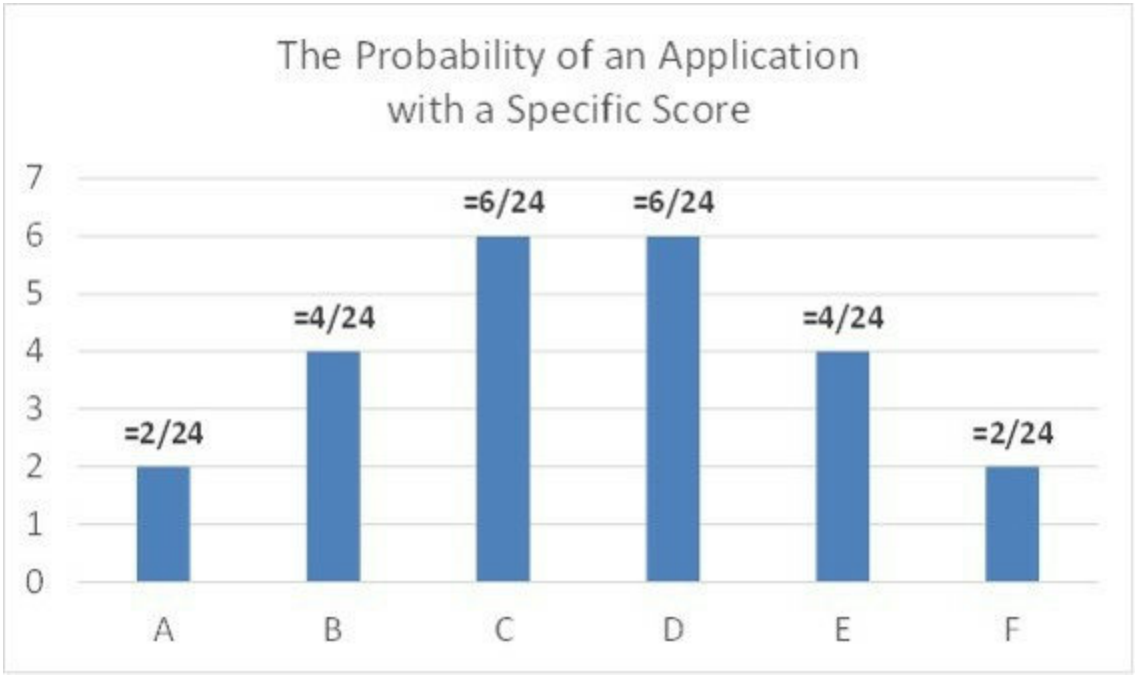
You are a member of the Loans Application Committee in a Bank. Your task is to review each application from a client and score it. The possible scores are from A to F (where A is the highest and F is the lowest score). On your desk are 24 scored applications. You prepare a frequency table showing the number of applications per score level:

Score	Frequency	% Frequency	
A	2	=2/24	8.33%
B	4	=4/24	16.67%
C	6	=6/24	25.00%
D	6	=6/24	25.00%
E	4	=4/24	16.67%
F	2	=2/24	8.33%
Total	**24**		**100.00%**

The 3rd column shows integer ratios which are only shown to highlight the definition of probability and to help convert % frequencies to probabilities. Your boss comes in and asks you: did Mr. XYZ score a B (as he happens to be wife's cousin)? What is the probability of having a score of B? Or: what is the expected value of drawing out an application whose score is B? The answer is the number of B's divided by the total number of applications. This is = 4/24 = 1/6 = 16.66%.

Moreover, if our client population is homogeneous, we can use this to forecast scores. What if we had a new application, what would be the probability of it having a score of B?

Here is the corresponding bar chart (histogram) showing the probabilities instead of the counts:



The above bar chart is the **Probability Distribution** of the event: the score of an application.

C) Cumulative % Frequency and the Addition of Probabilities

Two events are mutually exclusive if they cannot happen at the same time. Example: the set of German cars and the set of Italian cars are mutually exclusive since we cannot have an item belonging to both. The set of German cars and the set of blue cars are not

mutually exclusive since we can have a German car that is blue.

We shall only be concerned with **mutually exclusive events** which are also known as **independent observations**.

Definition: the **probability of two or more mutually exclusive events** taking place at the same time is the sum of their individual probabilities.

Example 1: in your town, there are 10,000 German cars, 5000 Italian cars and 35,000 cars of other origins. The probability of your neighbor buying a German or an Italian car is 10,000/50,000 + 5000/50,000 =20% + 10% = 30%.

Example 2: in our loan applications example, if one of the 24 files drops on the floor, what is the probability that it might have a score of A or D? The probability of A is 2/24 or 8.33% and that of D is 6/24 or 25.00%. The added probability is 8/24 or 33.33%.

So what is the probability if you toss a coin that it will come down a head or a tail? I will also leave you with a story that might help you solve the coin puzzle. A man was waiting for the lift on the 3rd floor. The door opens and there was a statistician inside the lift. The man outside asks: "Is this lift going up or down?" The statistician says: "Yes".

We will use this characteristic of probability to exploit the **Probability Cumulative Distribution.** Rather than ask what the probability of an event is, this distribution can give us the probability that an item or an observation will be found at a value **less than or equal** to a specified level. (Using the complementarity law defined earlier, it can also tell us what the probability is that an observation will be **larger** than a specified level: jus subtract the first value from 1).

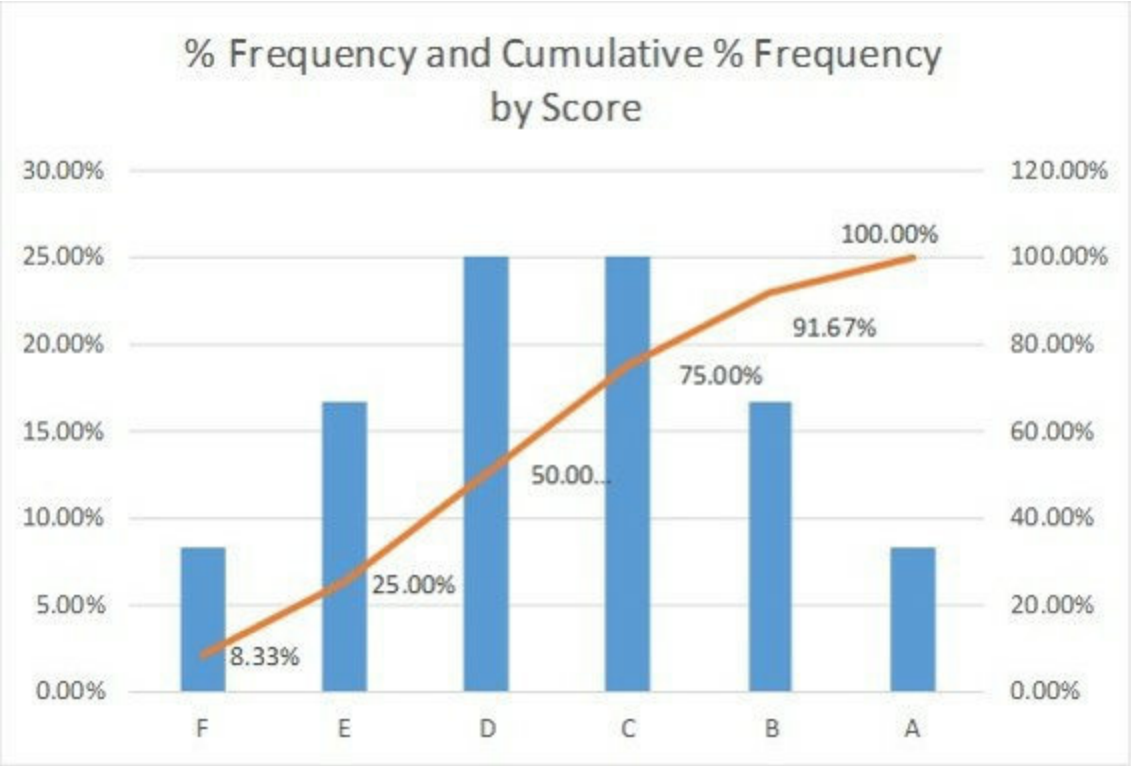
For example:

a) What is the probability of finding a pilot in our airline who is shorter than 180 cm? Answer: we just add the probability of finding a pilot whose height is 130 to that of one whose height is 131 and 132,..... and so on, up to 179.
b) What is the probability of finding a task in our project whose duration is more than 12 days? We add the probability of every task longer than 12 days.
c) As per our loan applications example: what is the probability of finding an application with a score less than B?

Using the same frequency table above but with two cumulative frequency columns, we get the progressive addition that we are after:

Score	Frequency	% Frequency		Cum Frequency	Cum % Frequency
A	2	=2/24	8.33%	2	8.33%
B	4	=4/24	16.67%	6	25.00%
C	6	=6/24	25.00%	12	50.00%
D	6	=6/24	25.00%	18	75.00%
E	4	=4/24	16.67%	22	91.67%
F	2	=2/24	8.33%	24	**100.00%**
Total	**24**		**100.00%**		

The next chart shows a double plot. It uses the above values of the **% Frequency** (blue bars) and the **Cumulative % Frequency** (orange rising line whose values are shown on the right hand vertical axis). To follow the procedure of generating such a chart, refer to chapter 17.0:



Note: the Bar Chart / Cumulative Line combination is usually shown with 2 different scales or vertical axes or Y-axis. The left scale covers the blue bars. The right scale covers the cumulative curve. To get a combo chart to show a secondary axis, simply right click on horizontal axis scale (the X values) and select FORMAT DATA SERIES. Then select the SECONDARY AXIS option.

The questions in this table are answered by adding the probabilities of each applicable score. The **cumulative probability distribution** is equivalent to the **cumulative % frequency**.

Q	Probability of Getting a Score . . .	Mutually Exclusive Events	Answer (Prob)	Answer (%)
1	Worse than A	B+C+D+E+F	(2+4+6+6+4)/24	91.66%
2	Worse than B	C+D+E+F	(2+4+6+6)/24	75.00%
3	Worse than C	D+E+F	(6+4+2)/24	50.00%
4	Worse than D	E+F	(2+4)/24	25.00%
5	Better than C	B+A	(4+2)/24	25.00%
6	Better than D	C+B+A	(6+4+2)/24	50.00%
7	Better than E	D+C+B+A	(6+6+4+2)/24	75.00%
8	Between C and D	C+D	(6+6)/24	50.00%
9	Not Between C and D	F+E+B+A	(2+4+4+2)/24	50.00%

Question 1: the probability of getting a score worse than A is the sum of the probabilities of B+C+D+E+F = 91.66%. (Note that A is not included).

Question 2: the probability of getting a score worse than B is calculated in the same manner. It is the sum of C+D+E+F = 75%. (Again, B is not included).

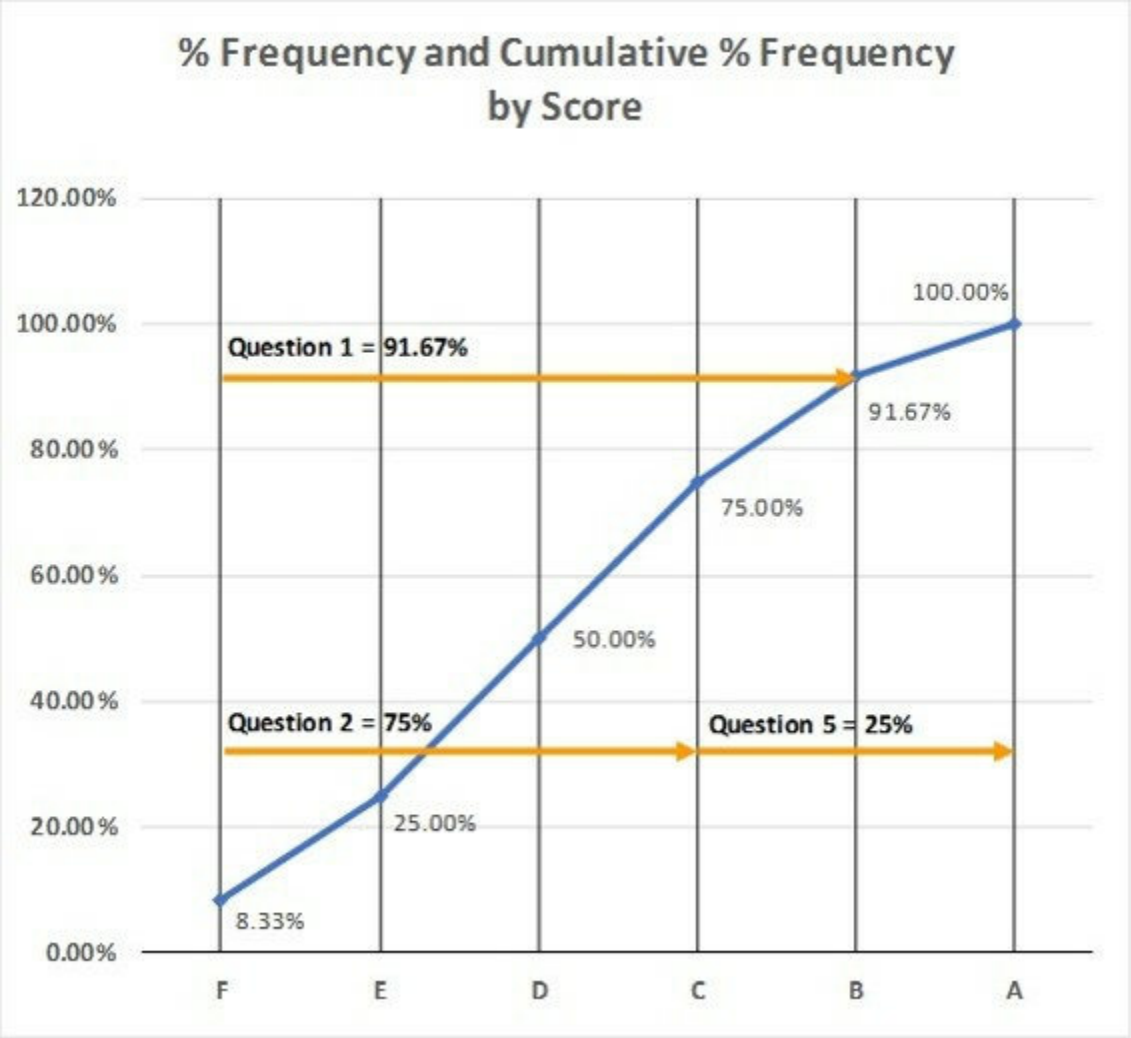
Question 3: the probability of getting a score worse than C is the sum of the probabilities of D+E+F = 50%.

Question 4: the probability of getting a score worse than D is the sum of the probabilities of E+F = 25%.

Question 5: What is the probability of getting a score better than C? This is the sum of B+A = 25%. If we ask what the probability is to get a score less or equal to C, then this is the complement of the first question and is = 100% - the probability of getting a score better than C = 75%. Why? Because the probability of (F+E+D+C) + (B+A) = 100%.

Question 6: what is the probability of getting a score lower than D and higher than C? We are after the probability of F+E+B+A = (2+4+4+2)/24 = 50%. This can also be the sum of the answers of Questions 4 and 5 = 25%+25% = 50%.

Next we show how plotting the cumulative probability distribution of the applications will give us the same result:



Question 1: shows the probability of a score worse than A to be 91.67% as shown earlier.

Question 2: shows the probability of a score worse than B to be 75%.

Question 5: shows the probability of a score better than C to be 100% - 75%.

It might seem to be complex to calculate the cumulative probability visually. In most cases, we will be using distributions that have a closed mathematical form (or one that can be calculated by a specific Excel formula). Using the cumulative distribution becomes a simple task.

D) The Difference between Probability Mass and Density Distributions

So far, we have been using probability distributions where the X-axis shows discrete values: integers, categories, number of responses, *etc.* There are other distributions where the X-axis shows continuous variables such as the Normal or the Weibull Distributions. Examples of continuous variables are: heights, weights, lengths, temperature, percentage failure, *etc.* The measurements are really discrete. The height is measured as 150.01, 150.02, 150.03 and so on. However, since their distributions are Normal or Weibull, they height of the bars can be calculated using a formula. The bar chart will have many "bars" starting from 140.00 and end in 186.00. There are 2600 bars in such a chart.

The logic is the same. All we have to do is to think of the bars as having infinitesimal width and they are all stuck together. Their sum would still be 1. The “envelope” of the bar chart would not be jagged. It would look smooth. However, it is not really smooth because the bars are so close to one another so we cannot see them. The concepts of probability and cumulative probability do not change. However, the terminology does and we wish to avoid any confusion.

The two terms (mass and density) are often used interchangeably although they should not be. In the case of continuous distributions, the probability distribution is called the **Probability Density Function** or **PDF**. In the case of discrete distributions, the probability distribution is called the **Probability Mass Function**. There are mathematical considerations that explain the rationale behind these terms. However, they do not affect us. So, we shall not bother with using two different terms and will make it simpler and just use the term "Probability Distribution" for both continuous and discrete functions. It will be clear from each model whether the variables being sampled are from a discrete or a continuous distribution.

Another problem is that although PDF is a standard term. Yet, the natural and corresponding acronym (PMF) for the Mass function is not in use.

10.0 How to Generate Random Numbers in Excel

We start with a general "proviso" about random numbers. We then define what RAND() does. In the last section, we present an example of the function. We then apply the CHI Squared test to check the validity of the generated numbers: does RAND() really generate numbers that are uniformly distributed between 0.00 and 1.00?

First of all, **we cannot generate real random numbers**. Randomness is often defined as something occurring by chance. This is totally wrong. Everything that happens has a chance of happening. Scientists define some phenomenon as random if it results in some events that do not have a cause or whose probability cannot be known in any way. For example, if we start out with 1 million atoms of Uranium 238, after 1.2 minutes, half of these atoms will decay and become Uranium 234. Quantum theory states that this is a probabilistic phenomenon. There is no way of knowing which atom will decay. This is true randomness. Another example: if we are interviewing airline passengers coming into the checkin bay, we have no reason to define beforehand the probability that the next passenger will be a male. The choice would then be random. If, however, we know that 80% of passengers are males, then the arrival of the next passenger who is a male will not be random.

When it comes to computers generating random numbers, there is no such thing as true randomness. Every series of random numbers generated by a computer program is “caused” by that program. We can repeat the program and it will generate the same number. Some random number generators use initial (seeding) values such as voltages in the motherboard. These may seem random. However, start with the same value of voltage and you will get the same random number series. Hence, the series is not truly random.

The technical term to use is **pseudo-random numbers**. Since it is long and we are not concerned with the issue of pseudo vs real random numbers, we will drop the "pseudo" but we should not forget about it. (We are not compromising too much as this is an industry practice, albeit faulty at heart).

Throughout the years, mathematicians have given a fair amount of attention to two issues related to the generation of pseudo-random numbers:

a) What are the criteria we can use to test a set of numbers that allow us to establish whether they are random or not (or how close they are to randomness)?
b) How to generate random numbers that comply with some or all of the above criteria?

We will not be concerned with this theoretical discussion. Our concern is the use of random numbers in Monte Carlo Simulation. This means that a "good enough" generator

such as RAND() in Excel will be useful.

Other random number generators than RAND() can be found on the web. Moreover, you can develop VBA routines for random number generating algorithms, which are maybe better than RAND(). Some pre-packaged generators can be acquired from the web which can be free or chargeable Add-ins to Excel. Note that the Analysis Toolpack also has a random number generator. However, it uses the same "engine" as used in RAND(). Since this eBook is for learning, we will use RAND(). Which generator to use is you choice.

There are several characteristics to use when testing whether a series of random number is random or not. We will only use the Uniform Distribution test. If you use your search engine to search for "tests for randomness", you will be directed to many papers and websites that lists such tests and provide examples of each. Try this, for example (Click Here**).**

In the next section, as an example, we will use the goodness of fit method (Chi Squared) to check whether the generated numbers by RAND() are significantly random or not.

A) What does RAND() Do?

RAND() does one thing very well: each time something changes in Excel or when you press F9, RAND() will generate a number between 0 and 1. The precision is 15 digits to the right of the decimal point. Here is a sample of 10 numbers with 15 digit precision: 0.1754960621149300

0.6541296990841120
0.0261850185443662
0.1400277640367690
0.2579825078197550
0.0286008248713061
0.4851733035310800
0.8722727636472720
0.3697830486437410
0.4463818849149640

The generated numbers are uniform. This means that any generated number has the same probability of getting generated as any other. If you bet on one and I bet on another, then after a long series of "tossing", my number will come up as many times as yours. No one will win.

B) How to Test Randomness

As mentioned earlier, there are many tests that statisticians use. We can depend on one that is available in Excel: the "goodness of fit" or the Chi-Squared test. It tests if the generated numbers are uniformly distributed or not. Other tests use different criteria that we shall not get into.

What does the Chi-Squared test do? An example. You have a frequency table made up of 10 rows. Each row shows you the number of clients your company has from a specific sector: industry, education, transport, trade, government, *etc.* This is the reference distribution. A new branch was launched recently and it now has a reasonable number of clients. The management wants to know if the distribution of these clients relates to that of the overall company. So now, we have two frequency tables. The Chi-Squared test applies a procedure to these two tables and comes up with a test value. This test value is checked against statistical tables and comes up with one of two conclusions: 1) We can safely say that the clients of the new branch have the same distribution as the mother company.

2) We cannot safely say that the clients of the new branch have the same distribution as the mother company.

The Chi-Squared test establishes whether it is reasonable for us to state that the two results are sufficiently close to each other for us to say they come from the same population.

The Chi-Squared test will be used in some of our models. There is no need to present its theoretical basis. The following Workout is a practical example of its use. Its objective is to compare a frequency table of numbers generated by RAND() made up of two rows with the theoretical or a reference distribution.

Workout 7: Test the Uniformity of RAND with Chi Squared

Purpose: to setup a sheet that shows how the RAND() function generates random numbers uniformly between 0 and 1. The workout will then show how Chi-Squared Distribution can test whether generated numbers are uniformly distributed or not.

Step 1: create a new workbook and save it under any name you wish. In the Workouts Folder there is a fully solved model called **Test the Uniformity of RAND with Chi-Squared**. Rename the first sheet as "Test RAND() with CHI".

Step 2: enter the following labels in Row 1:

A1 = RAND()
B1 = Bins
C1 = Actual Count
D1 = Expected Count
E1 = Sq Dev / Exp

Step 3: let A2 = RAND() and copy it down to A20001. Each time you press F9, Excel will re-generate a new set of 20,000 numbers.

Step 4: let us see how these numbers are distributed. We will let Excel count how many numbers are found in each of 10 brackets: between 0 and 0.10, greater than 0.1 and less

than or equal to 0.2, greater than 0.2 and less than or equal to 0.3 and so on. For that we need 10 bins.

Use Excel's autofill facility to generate the sequence 0.10, 0.20... 1.00 in the range B2:B11. These are the bins or the brackets of the frequency table.

Since the numbers in Col A are supposed to be uniformly distributed, if we segment the range from 0 to 1 into 10 bins, we should **expect** each cell to contain 20,000 / 10 = 2000 entries (frequency).

Step 5: in Col C we will use Excel's COUNTIFS() function. Please follow the procedure defined in chapter 6.0.

Exception: in that chapter, we required the first bin value to be less than the minimum. This does not work here since we are starting with 0.10. There are definitely many values generated by RAND() which are less than 0.10 (around 10%). So, we cannot have the same COUNTIFS() formula we used in that chapter copied from C2 down to C11. The formula in C2 will only test for values less than 0.10 (or what is found in the bin in B2). We will use COUNTIFS() with one argument: C2 = COUNTIF($A$2:$A$20001, "<="&B2)

C3 = COUNTIFS($A$2:$A$20001, ">"&B2,$A$2:$A$20001,"<="&B3)

Copy C3 down to C11 to get:

	A	B	C
1	Rand()	Bins	Actual Count
2	0.5808	0.10	1952
3	0.9566	0.20	1995
4	0.2880	0.30	2064
5	0.8307	0.40	1921
6	0.7651	0.50	2020
7	0.3039	0.60	2084
8	0.9286	0.70	1895
9	0.9280	0.80	2029
10	0.1858	0.90	2039
11	0.3009	1.00	2001

C2:C11 will now contain the frequencies of the 20,000 random numbers in each of the brackets stated in B2:B11. (These will be different in your model).

You can see that the frequency count hovers around the value of 2000. However, we cannot tell, visually, if the numbers are too far from 2000 or not. (Note that only 10 RAND() values are shown in Col A).

Step 6: how is the Chi Square test is prepared? The test is a comparison between the expected values and the actual values. To get a total test value for Chi Square, we need to calculate the following for each category: a) Find the difference between the expected and the actual value. It does not matter which is subtracted from which since we will be

squaring the result. This is often called the “deviation”.

b) Square the deviation.

c) Divide the squared deviation by the expected value, or 2000 in this case.

d) Sum the results of the division. This is our “test value”.

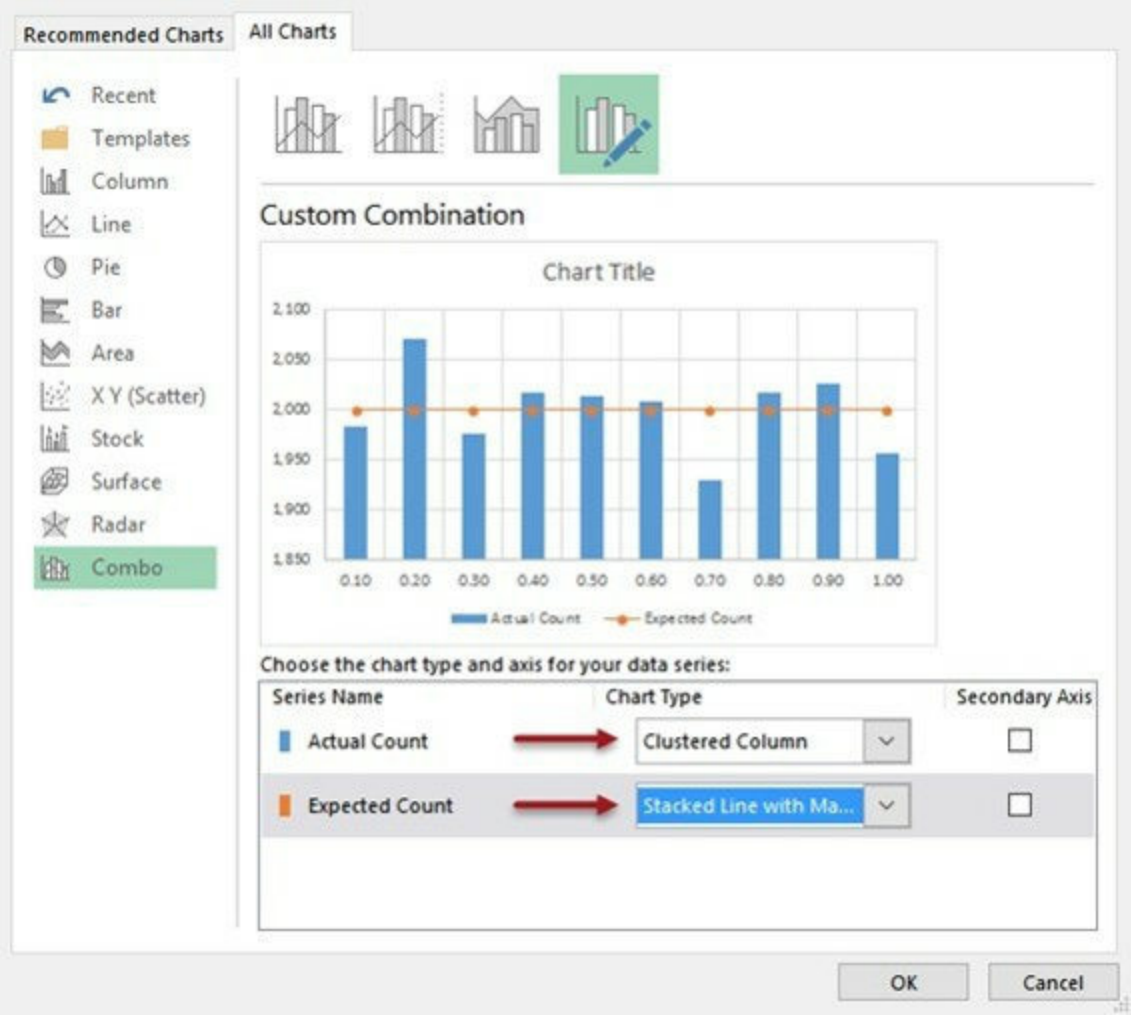
If the numbers are truly uniformly distributed, we should expect 20,000 / 10 = 2000 to be found in each bin. This is the **expected value** of the frequency count. Enter that in the range D2:D11.

Step 7: insert a chart as follows (as this is not a Pareto Chart):

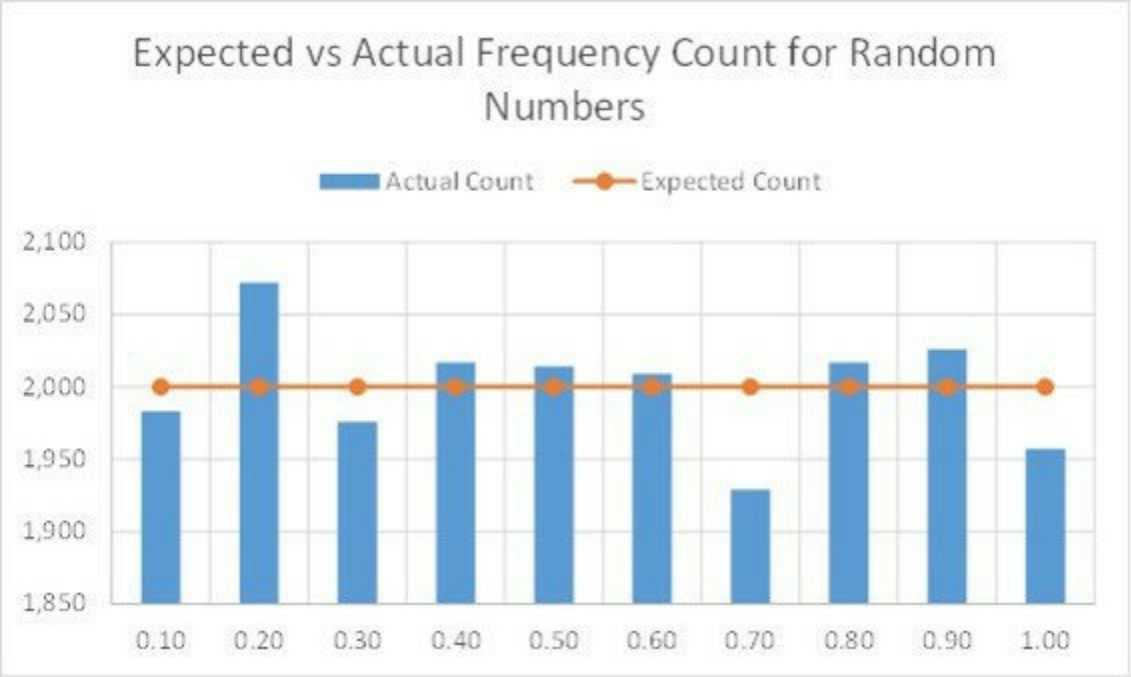
a) Select the menu item INSERT / CHARTS / SCATTER DIAGRAM

b) Select the line with markers chart type

c) Click on the Actual Count plot (the blue line or the C2:C11 range) and select to Change Chart Type. You will get:

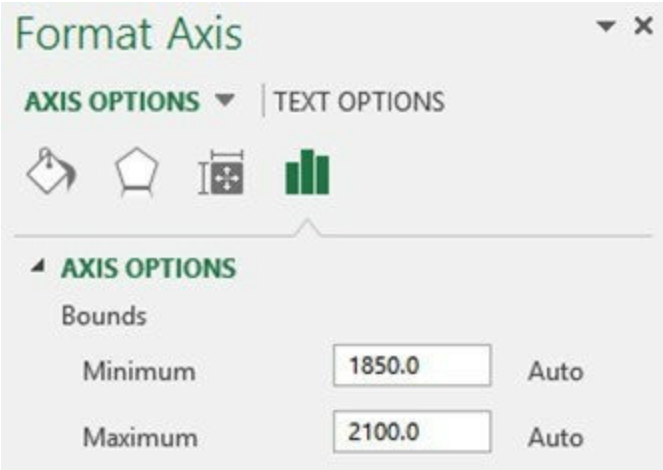


d) Change the types of the two plots to clustered column and stacked line with markers. You will get the following chart.

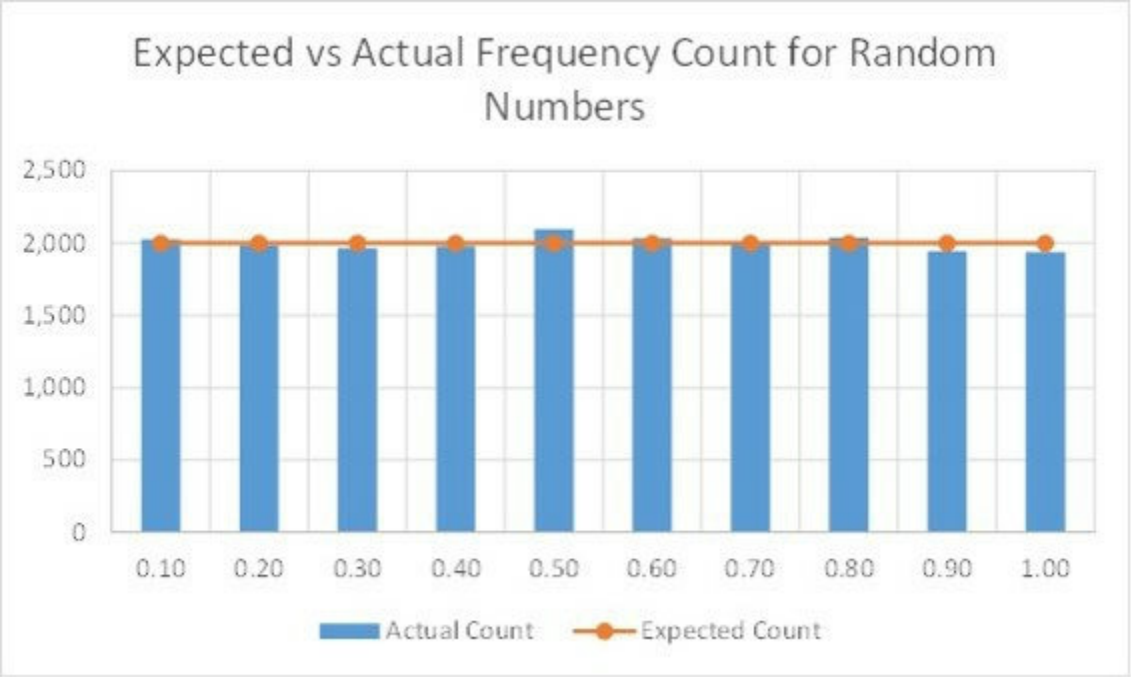


Here, we have a little charting trick. The differences between the actual and the expected count are visually misleading as they are shown as too large. The reason is that Excel chose to "chop off" the bottom of the vertical scale. This rescaling emphasizes the difference. To see the differences in their true light, the vertical axis must be changed from 1850 to 2100 to 0 to 2100.

Select the vertical axis, right click and select FORMAT Axis. You will get this options panel:



Change the minimum to 0 and you will get a more realistic view:



Changing the scale can be applied both ways: either to mask the exaggerated effects or to remove blanks below and above (or to the right and to the left if we are rescaling the horizontal axis).

Step 8: in E2 enter a formula that calculates the squared deviation and divides it by the expected value (from Col D):

E2 = ((D2 - C2) ^ 2) / D2.

Copy this formula down to E11.

Step 9: let E12 = SUM(E2:E11). This is our **test value** or the sum of each squared variances divided by the expected values in that Row.

Step 10: in C18 calculate the "critical value" which always depends on the two cells C16 and C17. These are peculiar to the Chi Square test and we will not discuss them in this eBook:

a) C16 = alpha or the confidence level usually set at 5% (here entered as 0.05)

c) C17 = COUNT(E2:E11) - 1. This formula counts the degrees of freedom of our test which is always 1 less than the number of values in the test. In our case, this is 10 - 1 = 9.

Step 11: let C18 = CHISQ.INV.RT(C16, C17) = 16.61. This is the critical value of the Chi Square distribution based on 5% alpha and 9 degrees of freedom. We need it to test the goodness of fit. For these two values, it is constant.

Step 12: compare the test value (16.61) with the critical value (16.9190):

a) If the test value is less, it means "we do not have any grounds to believe that the actual values differ significantly from the expected values". Another way statisticians say this is: “we do not have any grounds to believe that the actual values come from a

different population than the expected values".

b) If the test value is larger than the critical value, the Chi Square tests confirms that the numbers generated by RAND() are significantly different from their expected value or that they come from a different population.

Step 13: in C19 enter an IF statement that checks the above and places a corresponding text of the result:

C19 = IF(E12<C18,"There is no significant difference", "There is a significant difference")

	A	B	C	D	E
1	Rand()	Bins	Actual Count	Expected Count	Sq Dev / Exp
2	0.1098	0.10	2,029	2,000	0.42
3	0.3837	0.20	1,982	2,000	0.16
4	0.1733	0.30	2,001	2,000	0.00
5	0.9106	0.40	2,013	2,000	0.08
6	0.8932	0.50	1,967	2,000	0.54
7	0.7538	0.60	2,036	2,000	0.65
8	0.5440	0.70	1,979	2,000	0.22
9	0.0269	0.80	1,945	2,000	1.51
10	0.1417	0.90	1,998	2,000	0.00
11	0.5599	1.00	2,050	2,000	1.25
12	0.0667			Test Value	4.85
13	0.4906				
14	0.2986	Total Frequency Count	20,000		
15	0.4974				
16	0.1235	Confidence (Alpha)	0.05		
17	0.6640	Degrees of Freedom	9	=COUNT(B4:B11) - 1	
18	0.2902	Critical Value	16.9190	=CHISQ.INV.RT(C16,C17)	
19	0.9258	Result	There is no significant difference		

To get a feel for this, remember that if our results are exactly the same as the expected values, the deviations will all be zero and our test value will be zero, a lot less than the critical value. The nearer our test value is to zero, the more likely it will be less than the critical value.

In our case, the test value is less than the critical value. This means that the generated frequencies are close enough to the expected frequencies for us consider them as coming from the same population.

Conclusion: press F9 a few times. Each time you will get a different Test Value in E12. In most cases, there will be no significant difference. Only in a very few cases, there will be a difference. This shows that RAND() is good enough for our use.

A) Let us Play....

If you press F9 repeatedly, the value in E12 will change. In most cases, it will be less than the Critical Value in C18 and we can conclude that RAND() is well behaved and is providing us with uniformly distributed values from 0 to 1. But every now and then, you

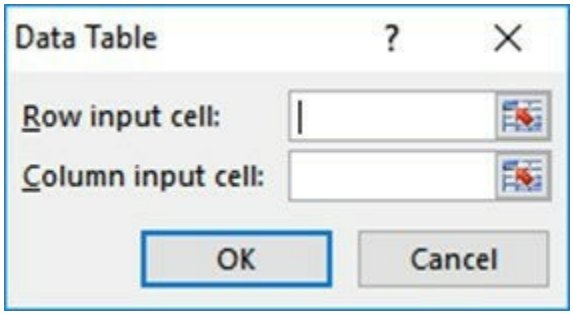
will get a Test Value that is greater than the Critical Value.

How many times? Is this significant? We will **play with our model** by applying sensitivity analysis or using Excel's WHAT IF facility. But the trick we will be using has a "design bug". Normal use of WHAT IF tables allows you to work on two different sheets at the same time. When you do not have input values in the WHAT IF table, that table has to be in the same sheet as the result cell. Here goes: **Step 1**: Use Excel's autofill facility to generate the sequence 1, 2... 100 in the range C41:C140.

Step 2: enter the formula in D40 = E12. You will be literally copying the test value into D40. We need that to generate other test values in the column D41:D140.

Step 3: select the table C40:D141 to include the empty cell C40.

Step 4: select the menu item DATA *DATA TOOLS* WHAT IF ANALYSIS and select DATA TABLE. You will get this dialog box:



This is one of the least intuitive Microsoft dialog boxes. We will be spending more time on WHAT IF analysis in chapter 14.0.

Step 5: click in the Column Input Cell and then click anywhere in the sheet then press OK. Excel will go through every row in the table and recalculate the value of E12 and place it for you in the cells in D41 up to D140. (The regular use of WHAT IF analysis places values in C41:C140 to be used in the calculation. We do not need that so we placed dummy values using the autofill).

Step 6: you will see 100 test values (repeated executions of the Chi-Squared test).

Let E41 = COUNTIF(D41:D140, ">"&C18)

This will count the number of test values out of 100 trials greater than the critical value. You will find there is a 1%-6% chance of that. This is acceptable in any case as we will never use Monte Carlo Simulation results for "contracts" or "specific" values.

	C	D
40		7.20
41	1	1.27
42	2	12.31
43	3	9.12
44	4	7.76
45	5	4.11
46	6	4.78
47	7	5.65
48	8	6.28
49	9	8.79
50	10	18.18

(Only the top 10 test values are shown in the WHAT IF table).

By the way, with this double simulation, i.e., first to get the random numbers then to span the WHAT IF table with test values just about started slowly my PC down.

B) Some Guidelines on Using RAND()

Use results as “input values"

Our results are found in columns that come from calculations using RAND(). As we start manipulating the Results sheet, each time there is a change in Excel, RAND() recalculates and generates a new value. If what you need is a general view, there is no harm in that. If you need to report the results and fix them, then using results that vary each time you change something in Excel is not advisable. You will not be able to refer to specific values in the model.

Solution: copy the results in the Model or the Runs sheet (which depend on RAND()). Paste them into a new column (usually in the Results sheet) where pasting should be "as values". The pasted values will be fixed or pasted as constants and will not vary as you apply calculations in Excel.

Avoid using RAND() Several Times within a Formula

If we are testing the result of RAND() using an IF statement, it would only work if you use it once:

IF (**RAND()**<0.5, "Yes", "No")

Excel generates one value of RAND() and compares it to 0.5. Suppose you generate a random number and wish to check in which of 3 brackets it is found: less than 0.4, between 0.4 and 0.7 or greater than 0.7. This would be the logical formula to use, but we should not: IF(**RAND()**<0.4, "< 0.4", IF(**RAND()**<0.7, "Between 0.4 and 0.7", "> 0.7")

The problem with this formula is that the two values of the RAND() will never be the same because each occurrence of RAND() will result in a newly generated value. So

we are not comparing the same random in the two IF statements. The results will be invalid. The solution is to setup RAND() in a separate cell and use that cell for comparison: B1 = RAND()

B2 = IF(**B1**<0.4, "< 0.4", IF(**B1**<0.7, "Between 0.4 and 0.7", "> 0.7")

By placing the random variates outside the IF statement, in B1, then B1 has the same value in each of its occurrences in the IF statement.

11.0 Models that Sample the Uniform Distribution

We used the uniform distribution in our first example in this eBook, without clarification. It was based on the Excel function RAND() which is a uniform distribution itself. In this chapter, we shall present a Monte Carlo Simulation model that uses the Uniform Distribution.

A) Why do we Need the Uniform Distribution?

In philosophy they have a law called "The Law of Insufficient Reason". It says that in the absence of any reason to think otherwise, any two events will have the same chance of occurrence. This is the reason we expect a coin to come an equal number of heads and tails when we toss it many times (unless we have a reason to expect the coin to land on its edge). The same applies to dice (unless we have a reason to believe the die is fraudulent).

Examples:

a) A price may vary uniformly between $6 and $10.

b) A task can have a duration between 10 and 12 days without us having any reason to favor one value over another in this range.

c) A product is tested and due to insufficient reason, we do not know how many defects to expect. However, we do know that it might have anywhere between 1 and 4 defects.

B) How to Generate Uniformly Distributed Samples?

By definition, RAND() is actually a uniform distribution. No value it generates has a reason to come up more frequently than others. We say that it produces numbers that vary from 0 to 1 uniformly. To generate uniformly varying numbers within different ranges, we have to transform the output of RAND().

Whenever we need such a distribution, we have to define a **lower** and an **upper** limit. In the first example above the lower limit = 6 and the upper limit = 10. Three computational steps (which we will combine into one formula below) result in a Uniform Distribution: a) **Generate** a random number between 0 and 1 using RAND().

b) **Scale** the 0 to 1 range so that it will fit the range of the required values. Our range = 10 - 6 = 4. To scale RAND(), multiply its results by the range 4. This will give us random value ranging from 0 to 4.

c) **Shift** the new range by adding the lower limit 6 to the results of (b). Instead of ranging from 0 to 4, the new range of random values will now range from 6 to 10.

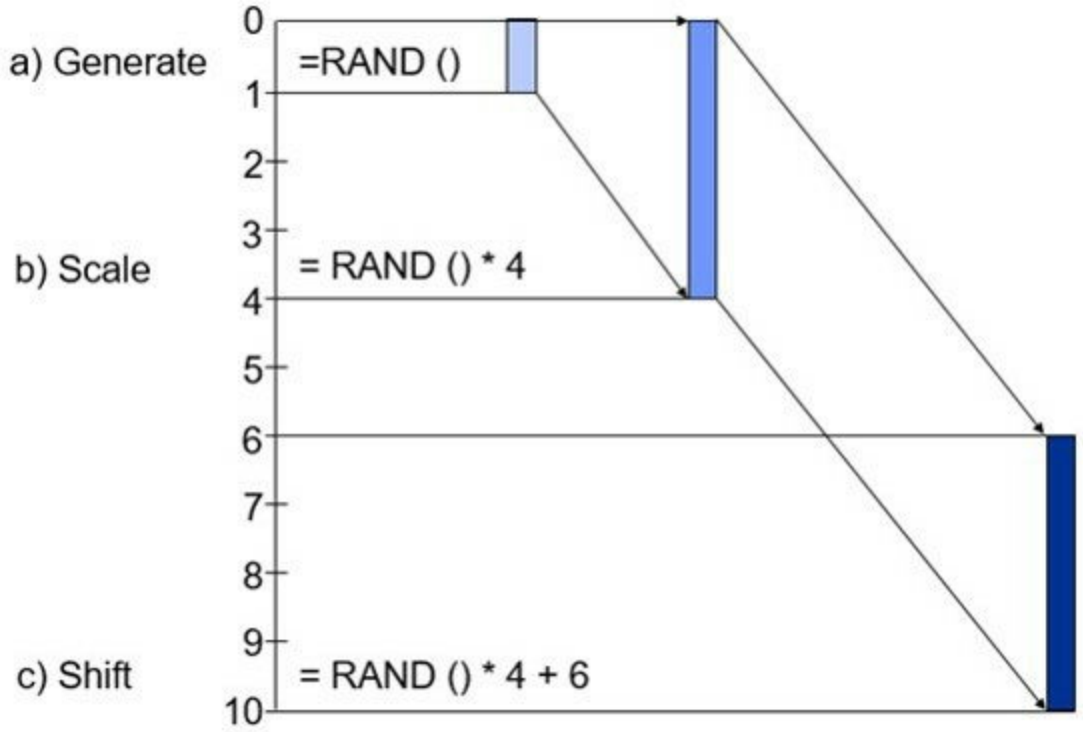
The above procedure can be expressed as a single formula:

RAND() * (Upper limit - Lower limit) + Lower limit
RAND() *(10-6) + 6 = RAND()* 4+6

Hint: as mentioned in our first model, it would be easier to write these formulas if you calculate the range in the Constants sheet. The formulas become:

RAND() * Range + Lower limit
RAND() * 4 + 6

The following diagram is a visual representation of the above 3 steps:



C) Three Different Ways of Generating the Uniform Distribution in Excel

3.1) Use a manually entered formula: this was presented above. This formula gives a set of uniformly distributed numbers between specific upper and lower limits:

= RAND() * (Upper limit - Lower limit) + Lower limit
= RAND() * (B6 - B5) + B5

It is best that you keep the limits and the range in the Constants sheet:

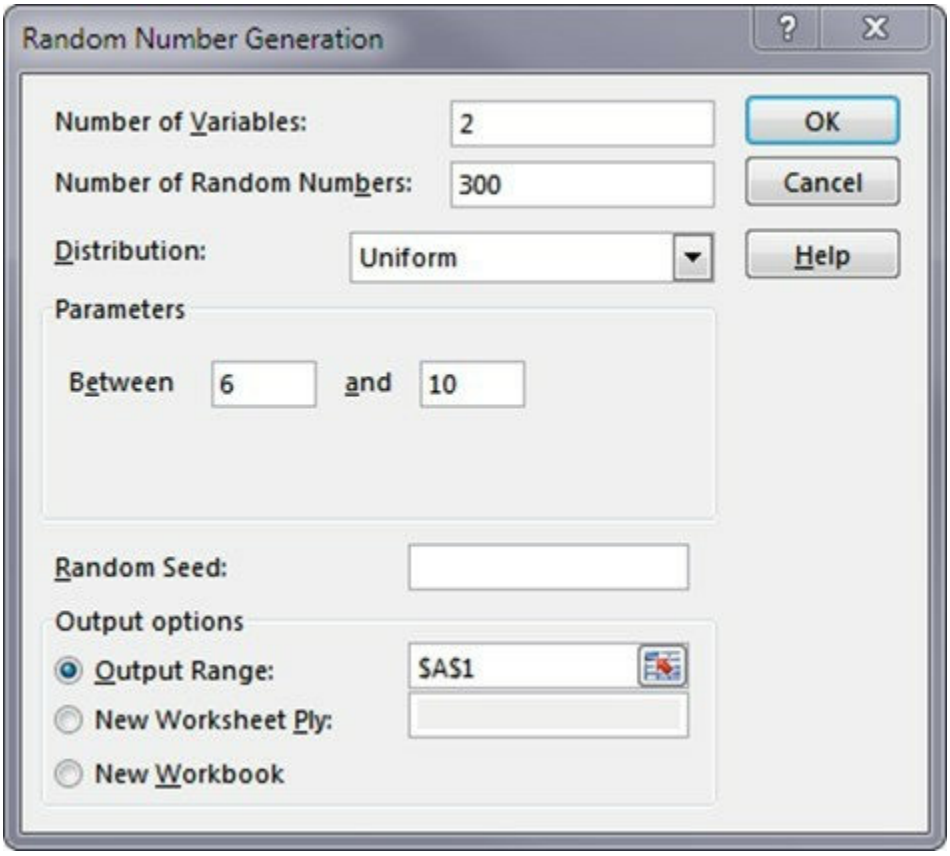
= RAND() * B7 + B5 if B7 = B6 - B5 or the range of variation

Note that it is possible to produce negative numbers using this formula. For example, we might need to simulate an input variable that ranges from -5 to +5.

3.2) Use the function: **RANDBETWEEN(lower limit, upper limit)**. This is a native function in Excel. However, it only produces integers. It also cannot produce negative numbers. In your formulation, you need to check that the upper limit is larger than the lower limit (See Step 5 in the following workout).

3.3) Use the Analysis Toolpack: the Toolpack has a facility called Random Number

Generation. Selecting that you will get a dialog box that generates a wide variety of distributions. If you select the "Uniform" from the drop down list, here are the parameters you need to fill:



Number of variables: this is the number of columns you can generate
Number of random numbers: the length of the columns
Between: the lower and the upper limit (the lower can be greater if you need negative numbers)
Random seed: used to create different sets of values. Not essential.

The facility is useful but it has one drawback: in case you need another set, you have to go back and re-launch the Analysis Toolpack. The first two methods above are dynamic. It means anytime something happens in Excel (or you press F9), you get new values.

We will mostly use the first method.

Workout 8: Animate the UNIFORM Distribution

Purpose: this is not a workout as much as an animation that shows how uniformly generated numbers are "uniformly" plotted. We will use both techniques: the formula defined above (for col B) and the RANDBETWEEN() (for col C).

Step 1: create a new workbook and name it as you wish. In the Workouts Folder there is a fully solved model called **Animate the Uniform Distribution**.

Step 2: generate 2000 random variables by inserting RAND() in A2 and copying it down to A2001.

Step 3: in F2 and F3 enter the lower and upper limits, say 60 and 90 respectively. In F4, calculate the range. Use spinners to raise or lower these values and observe how the chart changes. (The use of spinners is described in the Appendix in chapter 17.0).

Step 4: let B2 = RAND() * $F$4 + $F$3. Note that this formula will be calculated regardless of whether F3 is greater than F2 or not so you need to ensure your lower limit is less than your upper limit by applying validation rules on F2 and F3. Modify the formula with an IF statement to build in a validation test: B2 = IF($F$3<$F$2, RAND()*($F$2 - $F$3) + $F$3,"Error")

You can also use Excel's validation facility if you select the menu item DATA *DATA TOOLS* DATA VALIDATION.

Step 5: in C2 we will use the Excel function = RANDBETWEEN() to generate integers that are uniformly distributed. Since Excel will generate a #NUM error if the upper is not greater than the lower limit, the formula includes a check on ISNUMBER(). It tests if F2 > F3. If true, the formula would be valid and is calculated. If not true, the text "F2 must be > F3" is entered. This will also be an indication that the results in col B are also wrong. Enter this long formula: C2 = IF(ISNUMBER(RANDBETWEEN($F$3,$F$2)), RANDBETWEEN($F$3,$F$2), "F2 must be > F3")

Step 6: plot the values in B2:B1001 as a bar chart, you can see that most values are "represented". This shows that the generation was "uniform".

Secondly, in the range E5:F7, enter the following. Use the AVERAGE(), MAX() and MIN() formulas. They show the correct values since 74.91 is almost the average of the range 60 to 90 and the Max and Min are near enough too.

Average	74.91
Maximum	89.97
Minimum	60.01

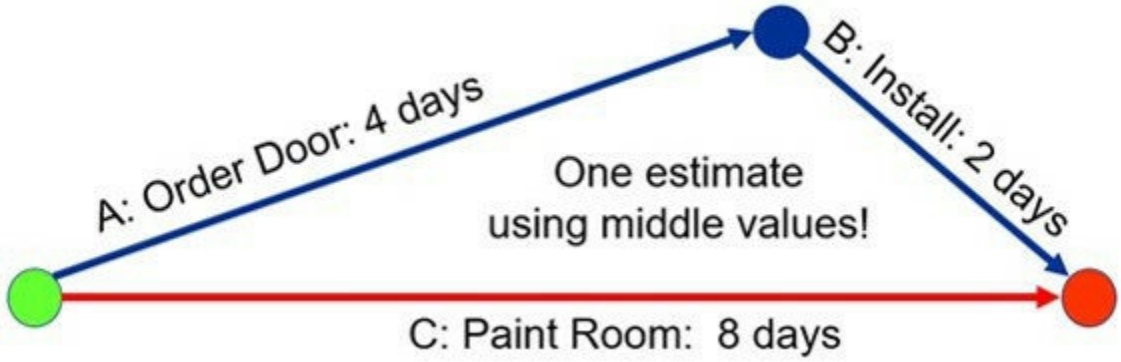
Of course, if you generate 200,000 values instead of just 2000, the above values will get nearer to their real counterparts.

Workout 9: A Project's Critical Path (UNIFORM)

Purpose: to simulate a 3 task project to analyze its total duration. Each of the 3 tasks varies according to a Uniform Distribution. (Later on in Part 2 of this eBook we will present more advanced techniques for simulating project schedules).

Problem statement: we have a small project made up of 3 tasks. Two are executed in

parallel with a third: while awaiting for the door to get delivered and installed, we can paint the room.



Task C for painting the room (8 days) is longer than (Task A for ordering + Task B for installing the door = 4 days + 2 days). Task C is the critical path in the project (because it is the longest). The 2 day slack in the upper path means we can afford a 2 day delay when ordering the door or getting it installed.

We are using the dangerous single point estimates here. But will any room get painted in exactly the number of days the contractor promised? Unlikely. Nor are the other two durations as consistent as that of the critical task. There might be a delay in installation where the total upper path is 4 days for delivery and 5 days for installation = 9 days. The lower path would then be shorter than 9 days which means it is no more on the critical path. The critical path has shifted to the upper path.

In real operations and especially with contractors, the durations are given as a range:

Order and delivery of the door take from 4 to 8 days
Installing it takes from 1 to 3 days
Painting the room takes from 6 to 10 days

Step 1: create a new workbook and name it as you wish. In the Workouts Folder there is a fully solved model called **A Project's Critical Path (UNIFORM)**.

Create the following sheets: "Model", "Results" and "Constants".

Step 2: in the range A1:D3 in the Constants sheet, enter the following parameters. The table contains the upper and lower limits as well as the range (computed in row 4) for the durations of each of the 3 tasks:

	A	B	C	D
1		Order Door	Intall Door	Paint Door
2	Upper Limit in days	8	3	10
3	Lower Limit in days	4	1	6
4	Range	4	2	4

Step 3: in A1 in the Model sheet, enter the label "Run ID".

Use Excel's autofill facility to generate the sequence 1, 2... 500 in the range A2:A501.

Step 4: enter the following header labels:

B1 = Order Door
C1 = Install Door
D1 = Paint Room
E1 = Critical Path

Step 5: enter the formulas that generate uniformly distributed samples for each of our 3 tasks:

B2 = RAND() * Constants!B$4 + Constants!B$3
C2 = RAND() * Constants!C$4 + Constants!C$3
D2 = RAND() * Constants!D$4 + Constants!D$3

Copy the range B2:D2 down to B501:D501 (since these are expressed in absolute references).

Step 6: in E2 find the longest path in the project for the first run. Use it to test whether the path "Order Door + Install Door" path is longer than the "Paint Room" task or not:

E2 = MAX(B2+C2, D2)

Copy E2 down to E501.

The following image shows the first 10 runs from our Model sheet:

	A	B	C	D	E
1	Run ID	Order Door	Intall Door	Paint Room	Critical Path
2	1	5.26	2.05	9.31	9.31
3	2	7.81	1.32	6.80	9.12
4	3	6.13	2.96	8.37	9.09
5	4	5.87	2.24	7.40	8.11
6	5	6.09	2.78	8.41	8.87
7	6	5.68	2.40	8.89	8.89
8	7	5.07	1.90	9.78	9.78
9	8	6.40	1.18	8.92	8.92
10	9	5.13	1.80	7.56	7.56
11	10	7.36	2.18	6.83	9.54

(Why are some cells blue? Further on, we calculate the average of the 500 critical paths and place it in K4. We then use conditional formatting to color blue any cell with a critical path > the average in K4).

So far, Monte Carlo Simulation has given us a column of 500 items of raw data each one representing the critical path for 3 randomized input samples. Even if we do not conduct extensive analysis, this column (found in E) provides beneficial information. The next section shows a modest analysis of this simple model.

Step 7: to analyze the critical paths in E2:E501, copy the range E1:E501 and paste it into A1 in the Results sheet. To avoid the results range being "dynamic", use **Paste as Values** to freeze the results. (The reason we do that is to avoid the results column changing all the time as and when we enter new formulas and data in our model. By pasting "as values", the column in the Results sheet will be static.

Step 8: enter the following block in the Results sheet. We shall use similar blocks in most of our analyses:

	A	B	C	D
1	Critical Path	Min	6.04	Bins
2	6.51	Max	10.81	6.00
3	9.54	Range	4.77	6.10
4	6.38	Bin Count	30.00	6.20
5	8.36	Bin Size	0.16	6.30
6	7.24	Final Size	0.10	6.40
7	7.38			6.50
8	9.28			6.60
9	7.70			6.70
10	6.95			6.80

C1 = MIN(A2:A501)
C2 = MAX(A2:A501)
C3 = C2 - C1
C4 = a suggested bin count to be entered manually, say 30
C5 = C3 / C4 = 0.16 = the calculated bin size
C6 = 0.10 a value that we enter manually guided by the result of C5

Note that the formatting for the above cells has to show fractional values since the critical paths are fractional.

Step 9: prepare the bins as described in Chapter 6.06.0:

a) Enter the label "Bins" in D1

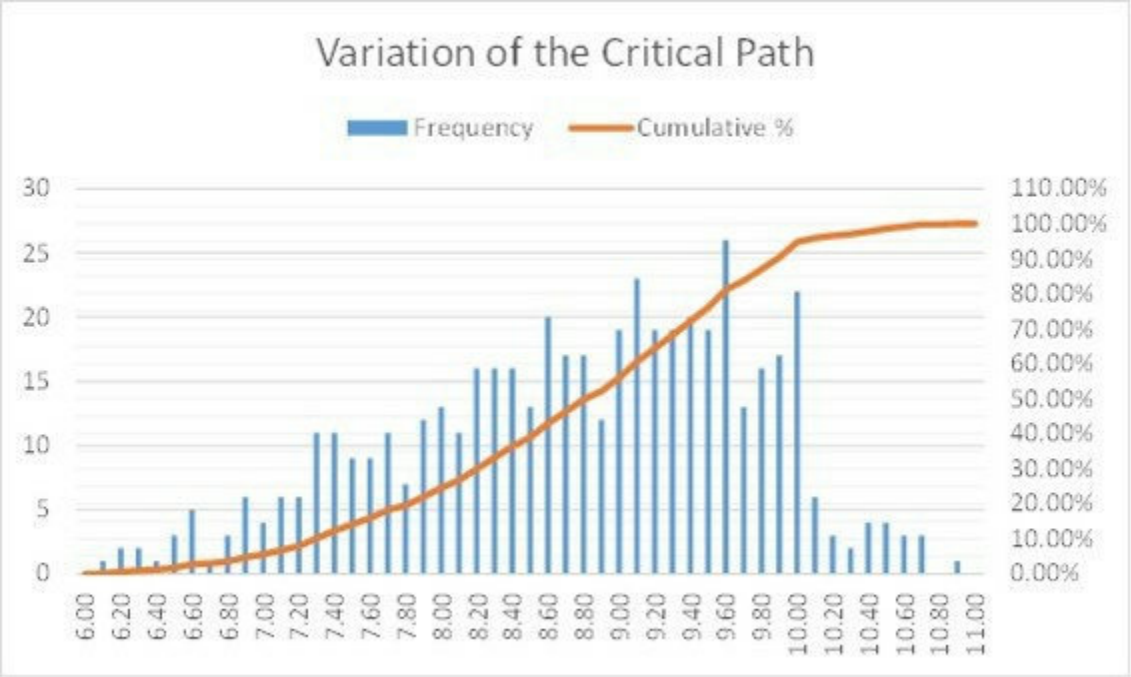
b) To define the first bin, enter a rounded value just below the minimum in C1, say 6.00

c) Let D3 = D2 + $C$6 (the initial value + the final size)

d) Copy D3 down to D52. Using our data, D52 was enough. However, it depends on what your model will have as a maximum critical path (found in C2). In effect, keep on copying downwards till you reach a value one or bin sizes larger than the maximum.

Step 10: use COUNTIFS() as described in Chapter 6.06.0.

Prepare the chart as described in that Chapter too:



Looking at the right axis, locate the 85% value and draw a line to the left. The line will meet our cumulative chart when the bins value = 9.7 or so. This means that if we conduct this project 500 times using different contractors and other conditions, we will have a critical path less than 9.7 days in 85% of the cases. This is better viewed in the bins table, in the rows showing the cumulative % frequency 83.80% to 87.00%:

9.5	19	76.00%
9.6	26	81.20%
9.7	13	83.80%
9.8	16	87.00%
9.9	17	90.40%
10	22	94.80%

Interpolation can give you a more precise result. But it does not really matter. When we say 85% of our critical paths are lower than 9.7, it does not really matter if we are precise or not.

Step 11: prepare the Descriptive Statistics as described in the Appendix in Chapter 15.0. (You can copy the format from the General Model Template found in the Templates folder). Using our model, we get:

Critical Path	
Mean	8.6979
Standard Error	0
Median	9
Mode	#N/A
Standard Deviation	0.9853
Sample Variance	0.97
Kurtosis	-0.49
Skewness	0
Range	5
Minimum	6
Maximum	11
Sum	4348.949321
Count	500.0000
Confidence Level(95.0%)	0.086570294

Another way of "measuring" our results is to use the statistical rule of the thumb which says that around 70% of our results fall within a range determined by this formula:

Lower value = mean - standard deviation = 8.6979 - 0.9853 = 7.71 days
Upper value = mean + standard deviation = 8.6979 + 0.9853 = 9.68 days

This range gives an "engineering" feel for our values since it eliminates outliers or values in the extreme ends of the frequency count. In this manner, we would eliminate the top 15% and the bottom 15% results. (In pure mathematical terms, this is not 70% but 68.34%).

Workout 10: Stock Reordering (UNIFORM) - Importing Data

Purpose: to present a simple inventory model using uniformly distributed demand. The workout shows how a column of actual demand data can be imported into the model rather than resort to sampling a distribution.

Problem statement: a retailer receives orders by phone. If the quantity on hand is enough to meet the demand, the order is delivered. If the demand is higher than the quantity on hand, the retailer will buy the specific balance from a wholesaler but at a higher than the usual contract price. The full order as per the demand can then be delivered.

At the beginning of each month, the retailer checks whether or not the brought forward quantity on hand from the previous month is higher than a specified quantity This is sometimes called the "minimum" or the "reorder level". If it is, no order is placed. If the

expected demand is higher, an order is placed for a specified quantity.

Our first objective is to simulate the above operation over 60 months and analyze the total profit. The demand will be sampled from a uniform distribution. In the second expression of the problem, we will import a column from another Excel sheet so it can replace the uniformly distributed demand. This column represents a sample of actual data imported from another branch or based on some study.

A) The Procedure for Uniformly Distributed Demand Quantity

Step 1: create a new workbook and save it under any name you wish. In the Workouts Folder there is a fully solved model called **Stock Reordering (UNIFORM) - Importing Data**.

Step 2: create the following sheets: Model, Runs, Results and Constants.

Step 3: setup the Constants in the Constants sheet

	A	B
1	**Fixed Demand**	350
2	Lowet Limit	300
3	Upper Limit	450
4	Range	150
5	Opening Quantity on Hand	50
6		
7	Fixed Order Cost (per Order)	450
8	Purchase Price (per Item)	1,500
9	Sales Price	2,200
10	Shortage Cost (per item)	50
11	Holding Cost (per Item)	10
12		
13	**Order Policy**	
14	Reorder Point (Minimum Level)	50
15	Reorder Quantity	350

Step 4: in the Model sheet, setup the single static model or formulation so you can test the computational procedures:

	A	B
1	Opening Quantity on Hand	50
2	Ordered Quantity	350
3	Month's Demand	500
4	Actual Quantity Sold	400
5	Shortage Quantity	100
6	End of Month Quantity	0
7		
8	Fixed Order Cost (order)	450
9	Variable Order Cost (ordered qua	525,000
10	Total Ordering Cost	525,450
11	Shortage Cost	50
12	Holding Cost	0
13	Sales Revenue	880,000
14		
15	Total Profit by End of Month	354,500

The following clarifies the various quantities:

a) **Opening quantity on hand**: this is a constant found in the Constants sheet (hence yellow). When we develop the Runs, this quantity will be filled by the end of month quantity or the quantity on hand of the previous month.

b) **Ordered quantity**: this is tested against the minimum quantity (a constant). If less, then the reorder quantity is placed in the cell, otherwise, we keep it at 0.

Let B2 = IF(B1 <= Constants!$B$14, Constants!$B$15, 0)

c) **Month's demand**: in the Model sheet, this is our input variable. Initially, we enter it as a single point estimate. Later, we will use this cell to insert actual data into it.

d) **Actual quantity sold**: if the retailer is lucky, the quantity on hand + the ordered quantity would be sufficient to cover the demand. The actual quantity sold will be the same as the demand. If the quantity on hand + the ordered quantity is less than the demand, the retailer can only sell what is available. There will be a loss to the retailer expressed as a shortage quantity (see B5 next).

Let B4 = IF(B3 < SUM(B1:B2), B3, SUM(B1:B2))

e) **Shortage quantity**: this is the unsatisfied demand. The retailer assumes that every item not satisfied will cost the company a nominal amount. The rate per item in found in B10 in the Constants sheet.

Let B5 = IF(B3 > B4, B3-B4, 0)

f) **End of month quantity**: this is = opening quantity on hand + ordered quantity - actual quantity sold. (When running the simulation, this quantity will be brought forward to next month's opening quantity and moved to B1).

Let B6 = B1+B2-B4

The following clarifies the various monetary values starting with B8:

a) **Fixed order cost**: if there is an order, this is the fixed cost of the single order (from the Constants sheet).

Let B8 = IF(B2 > 0, Constants!$B$7, 0)

b) **Variable order cost**: if there is an order, this is the total purchase price = ordered quantity * purchase price (from the Constants sheet).

Let B9 = IF(B2 > 0, B2 * Constants!$B$8, 0)

c) **Total order cost**: this is the sum of the fixed + the variable order costs.

Let B10 = B8 + B9

d) **Shortage cost**: this is a variable cost which is included in the month if the demand is higher than the sum of the opening quantity on hand + ordered quantity. It is the shortage quantity * shortage cost (from the Constants sheet).

Let B11 = IF(B5 > 0, Constants!B10)

e) **Holding cost**: this is the cost of storing and handling the items that remain with the retailer at the end of the month. It is equal to a rate (from the Constants sheet) * the end of month quantity.

Let B12 = B6 * Constants!$B$11

f) **Sales revenue**: this is the actual quantity sold * the sales price (from the Constants sheet).

Let B13 = B4 * Constants!B9

g) **Total profit by end of month**: this is the net of revenue reduced by the total ordering cost + the shortage cost + holding cost. This is our output cell (hence the color blue).

Let B15 = - B8 - B9 - B11 - B12 + B13

Step 5: experiment with the Inventory Model by entering various values in the Constants sheet to test your formulation.

Step 6: in the Runs sheet, enter a top Row containing the headers. These are essentially a transposition of the vertical formulation in the Inventory Model. We literally transpose the vertical range B2:b15 in the Model sheet into a horizontal range B2:M2 in the Runs sheet (notice that we drop the blank cells and the total ordering cost in B10 in the model sheet).

Step 7: enter these labels in Row 1 in the Model sheet:

A1 = Run ID
B1 = Open QOH
C1 = Order Qty
D1 = Demand

E1 = Actual Qty Sold
F1 = Shortage Qty
G1 = End of Month Qty
H1 = Fixed Order Cost
I1 = Variable Order Cost
J1 = Shortage Cost
K1 = Holding Cost
L1 = Sales Revenue
M1 = Total Profit (or Loss)

Color the Demand in D1 in green as it is the input variable. Color the Total Profit (Loss) in M1 in blue as it is the output we are interested in.

Step8: use Excel's autofill facility to generate the sequence 1, 2... 60 in the range A2:A61.

Step 9: in Row 2, enter formulas for each of the columns as follows:

a) B2 = Constants!B5 (the opening quantity on hand).

b) C2 = IF(B2<=Constants!$B$14,Constants!$B$15,0).

c) D2 = INT(RAND() * Constants!$B$4+Constants!$B$2)

If the opening QOH is less than the minimum, order the reorder quantity. Otherwise, no orders this month and we enter 0. Since RAND() will produce a fractional quantity, we use the INT() function to truncate the resulting sample.

d) E2 = IF(B2+C2>D2, D2, B2+C2).

The actual quantity sold depends on whether the demand is less than the sum of opening QOH and the ordered quantity or not. If it is, enter the demand from B3 into B4. If the demand is higher than the available quantity, the retailer can only sell the opening QOH for the month and the ordered quantity.

e) F2 = IF(D2>E2, D2-E2, 0)

The shortage quantity is the difference between the demand and the available quantity. It will be used to calculate the shortage cost in K2.

f) G2 = B2+C2-E2

The end of month quantity is simply the sum of the opening quantity + the ordered quantity less the actual quantity sold. The shortage quantity does not come into the calculation.

g) H2 = IF(C2>0,C2 * Constants!$B$7,0)

This is the fixed cost of an order, if we had one this month. We check C2 and if it is > 0, we add the fixed cost from the Constants sheet.

h) I2 =C2 * Constants!$B$8.

This is the variable cost of the order, if we had one this month. As in H2, we check C2 and if it is > 0, we multiply the quantity ordered * the purchase price (from the Constants sheet).

i) J2 = IF(D2 > B2+C2, F2 * Constants!$B$10, 0)

The shortage cost is included if there was a shortage this month. It is checked against cell F2. If that is > 0, we multiply the shortage quantity * the shortage cost (from the Constants sheet).

j) K2 = G2 * Constants!$B$11

The holding cost is the value of the quantity at the end of the month multiplied by the holding cost per item (from the Constants sheet).

k) L2 = E2 * Constants!$B$9

The sales revenue is the actual quantity sold * the sales price.

l) M2 = L2 - K2 - J2 - H2 - I2

The total profit (or loss) is the sum of the sales revenue - the total order cost - the shortage cost - holding cost.

Step 9: we now face a typical modeling practice where the first Row is different from the rest. The reason is because we usually have **constants** in the first Row. In various models, such initial values as the following need to be fixed in the first run and brought forward in the subsequent runs: Opening balances

Start times
Quantity on hand at the beginning of the simulation

The second Row will often contain data that is the result of processing the first Row. We have a brought forward quantity from G2 to B3. We cannot simply copy Row 2 downwards. That will cause B2 (which is a constant) to be copied throughout Col B. We need to replicate the runs in several sub-steps: a) Let B3 = G2

b) Copy B3 downwards and until B61.

c) Replicate the horizontal range C2:M2 downwards and until Row 61.

This will avoid the problem of copying the initial conditions in B2 downwards by linking B3 to G2 in the second Row.

Step 10: setup up the Results sheet. This is a standard procedure that we've done before. So here is a set of "summary" steps.

Step 11: copy the range M1:M61 of the Total Profit column in the Runs sheet. Paste it as value into A1 in the Results sheet. These values should now be frozen and will not change on pressing F9.

Step 12: prepare the bin preparation area and the 3 standard results columns: Bins, Freq

and Cum % Freq. (Use the General Model Template in the Templates Folder to copy the format).

	A	B	C	D	E	F	G
1	Total Profit (or Loss)	Min	-16,370		Bin	Freq	Cum % Freq
2	195,800	Max	195,800		-20,000.0	0	0.00
3	85,450	Range	212,170		-15,000.0	1	0.02
4	45,510	Bins	40		-10,000.0	0	0.02
5	127,850	Bin Size	5,304		-5,000.0	2	0.05
6	83,050	Actual Size	5,000		0.0	0	0.05
7	86,300				5,000.0	0	0.05
8	83,700				10,000.0	1	0.07
9	14,570				15,000.0	5	0.15
10	157,600				20,000.0	2	0.19
11	84,100				25,000.0	0	0.19

Step 13:

In our case, -20,000 is the lowest bin to cater of 16,370. In E3 enter the minimum in D2 + the rounded bin size:

E3 = E2+$C$6

Drag this downwards until you reach a value larger than the maximum of A2:A61. You now have the bin column.

Use the procedure for preparing the frequency count, the cumulative % frequency and the Pareto Chart as per chapter 5.0.

Step 14: save the workbook as we will use it below.

Let's Play:

We can extend the model by manipulating the formulation as follows:

a) Change the UNIFORM Distribution to other more realistic distribution (after you see

how they are used in Part 2 of this eBook).

b) Simulate the variation of the opening balance using Sensitivity Analysis (WHAT IF) as we did with the RAND() workout in chapter 10.0. This problem can be asked in a different way: how does the total profit depend on random variations in the opening balance.

c) Of course, you can repeat b) above for any of the other constants in the formulation such as the ordered quantity, the cost of orders, other costs in the formulation, *etc.*

d) Analyze the markup which was shown as constant in cells B8 (Item Purchase Price) and B9 (Item Sales Price). Since these were fixed, we can analyze the variations by keeping one fixed and randomizing the other or randomizing both.

e) Add new operational items. For example, the retailer might consider items that are overstocked as items that can be sold at a discount. Establish a limit for overstocking and a possible value (usually a percentage) for items sold during end of season discount). The workout on modeling seasonal sales in chapter 1 of Part 2 of this eBook handles such an issue.

f) Stock management is ripe with issues that can be resolved with simulation. For example, the retailer above is purchasing quantities according to a very simply reordering policies: the minimum or the reorder point in B10 (Constants) and the Reorder Quantity (sometimes mistakenly called the maximum). Various purchasing policies can be simulated to address such issues as applying seasonal values for these two parameters, discount breaks, using past data to project demand, *etc.*

B) Repeat the Simulation with Actual Imported Data

This is simpler than it looks. It all depends on where you get your actual data from. Most likely, you will be able to use such data as:

a) The demand of another but similar item
b) The demand of the same item but in another branch
c) The smoothed demand for the last 5 years using moving averages or exponential smoothing.
d) The demand of a competitor's item
e) The forecast or projected demand using various statistical functions. Operational functions can also be used, for example, applying multiplication factors to shape the demand so that it is higher (or lower) in summer than in winter.

Step 1: open the file saved above and save it under another name since we are going to overwrite the sampled data with actual data. In the Workouts Folder, there is a fully solved example called: **Stock Reordering (UNIFORM) - Data**.

Step 2: you can either enter actual data of your own in the range D2:D61 in the Runs sheet or if you do not have data, import it as follows:

a) Copy the range A2:A61 from the workbook **Inventory - The Actual Data to**

Import in the Workouts Folder
b) Paste it onto the range D2:D61in the Runs sheet.

Step 3: press F9 to recalculate the Runs. Nothing changes because we do not have random sampling any more.

Step 4: now repeat the steps 9 to 14 in the previous procedure to analyze the results and complete your analysis.

Workout 11: Business Plan (UNIFORM) - Replicate Rows with WHAT IF

Purpose: to show how Excel's WHAT IF tables can be used to create replications. In order to focus on the method, the model in this workout will be simple.

Problem statement: we have a formulation which cannot be expressed as a single row. We will not be able to replicate the whole formulation downwards. WHAT IF tables can be used to replicate the output cells on their own. The result is a new column containing our output results but not using replicated rows from the formulation.

For example, a financial business plan is presented in a formulation covering the range A1:D28. (See image below).

We are interested in simulating three sub-totals and their grand total. The input variables are the Full Time Employment (FTE's) needed for Design Engineering, Business Analysis and Business Development. Since these are not found in one row, we will not be able to replicate them by direct copying. (Of course, you can replicate them sideways or horizontally, but in general, vertical is easier than horizontal scrolling. We will use the WHAT IF facility to replicate these cells.

Restriction: this method only works if the WHAT IF table is in the same sheet as the source cells or the cells we want to simulate (D9, D18, D26 and D28 below).

Step 1: create a new workbook and save it under any name you wish. In the Workouts Folder there is a fully solved model called **Business Plan (UNIFORM) - Replicate Rows with WHAT IF**.

Rename the first sheet to Model and create another sheet for Constants.

Step 2: in the Model sheet, enter the following formulation which represents a business plan in three blocks.

Block 1 (A1:D9) presents the one time costs of the business plan.
Block 2 (A11:D18) presents Phase 1 covering development and launch.
Block 3 (A20:D28) presents Phase 2 covering the first 6 months after launch

However, to save time entering such data and to get to the heart of the matter, go ahead and copy the range A1:D28 from the solved workbook in the Workouts Folder.

	A	B	C	D
1	Item	Monthly Rate	FTE's	Total
2	One Time Costs			
3	Devices			17,000
4	Software Applications and Licenses			3,000
5	Personal Computers (for design)			12,000
6	Related Units (printers, scanners, etc.)			8,000
7	Domain registration and hosting			1,000
8	Furniture			5,000
9	Sub-Total: One Time Costs			46,000
10				
11	Phase 1 - Development and Launch			
12	Project Leader	8,000	6	48,000
13	Planning Engineer	5,000	6	30,000
14	Design Engineer	6,000	5.01291	30,077
15	Marketing Expert	5,000	6	30,000
16	Business Analysts	4,000	28.2558	113,023
17	Graphic Designer	3,000	3	9,000
18	Sub-Total: Phase 1			260,101
19				
20	Phase 2 - First Six Months after Launch			
21	Product Maintenance Costs	1,000	6	6,000
22	Additional Functions	30,000	1	30,000
23	Web Administration	3,000	6	18,000
24	Customer Support	3,000	6	18,000
25	Business Development Officer	6,000	4.28267	25,696
26	Sub-Total: Phase 2			97,696
27				
28	Total Costs			357,797

All cells have numeric values to be entered directly except the following which require formulas:

a) The monthly cost rates for each of the human resources in Phases 1 and 2 are entered in col B. (These are hard coded and are not entered in the Constants sheet. This is against our recommended good practice, but since we are not after the model but after replication with WHAT IF, we will look the other way.

b) Col C contains the number of FTE's (full time employment index) needed for each profession. Some FTE's are hard coded constants except the 3 we noted above: Design Engineering (C14), Business Analysts (C16) and Business Development Officers (C25). These are colored in green and will be using randomized samples.

c) Col D contains the total cost. It is simply a formula in each row that multiplies the enter in col B by that in col C. (For the one time costs, there is no multiplication). For example, D12 = C12 * B12.

d) Sub-totals are found in D9, D18 and D26 with a grand total in D28. They are colored in blue because they are the outputs we need to analyze.

Step 3: after entering the static model above, we have to randomize the 3 green cells: C14, C16 and C25. To do that, we need parameters for the 3 uniform distributions. These are entered in the Constants sheet as follows:

	A	B	C	D
1	Item	Design Engineer	Business Analyst	Business Development
2	Lower Limit	4	24	3
3	Upper Limit	6	36	7
4	Range	2	12	4

The yellow cells are constant while the range in Row 4 is calculated as the upper limit - the lower limit.

Step 4: randomize the number of FTE's (Full Time Employment) of a design engineer using the Uniform Distribution with parameters from the Constants sheet:

C14 = RAND() * Constants!B4 + Constants!B2
C16 = RAND() * Constants!C4 + Constants!C2
C25 = RAND() * Constants!D4 + Constants!D2

Press F9 a few times to ensure that the cells C14, C16 and C25 are being sampled correctly. We are now ready to analyze the output in cells D9, D19, D26 and D28.

Step 5: enter the following labels in the Model sheet:

G1 = Run ID
H1 = S/T One Time
I1 = S/T Ph 1
J1 = S/T Ph 2
K1 = S/T Total

	G	H	I	J	K
1	Run ID	S/T One Time	S/T Ph 1	S/T Ph 2	Total
2		46,000	277,538	100,082	377,619
3	1	46,000	263,501	106,723	370,224
4	2	46,000	248,276	99,732	348,008

Step 6: to prepare a WHAT IF table, we place the output cells that we are interested in in the range H2:K2 (and color them blue for ease of matching with D9, D19, D26 and D28:

H2 = D9
I2 = D19
J2 = D26
K2 = D28

Step 7: in the Model sheet, use Excel's autofill facility to generate the sequence 1, 2... 1000 in the range G3:G1002. We will need this column as the left most column in Excel's WHAT IF analysis.

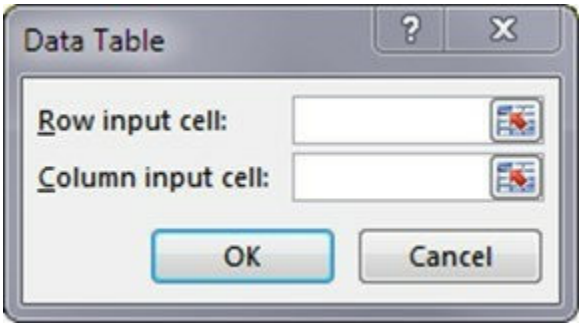
Note: this is not needed. The method we are introducing will live well with any number in the range G3:G1002.

Note: we chose to place the replicated rows in the Model sheet and not in a separate Runs sheet is because WHAT IF tables in Excel require their input cells (or entry fields as per the dialog box below) to be on the same sheet as the output table.

Step 8: select the range G2:K1002 for use with Excel's WHAT IF Table

a) Select the menu item DATA *DATA TOOLS* WHAT IF ANALYSIS / DATA TABLE

b) Click in either entry field (for rows or columns):



c) Now click in the field for COLUMN INPUT CELL and then click on any cell in the sheet and press OK. Excel will use the WHAT IF table procedure to calculate the cells H3:K1002. Each time it recalculates, RAND() will be invoked and that is how we get our simulation runs or replications. This is a table or an Excel Array. The cells cannot be edited.

The outputs in H2, I2, J2 and K2 do not depend on the values in Col G (the Run ID's). Excel copies the range H2:K2 downwards and recalculates their content. Since we have a RAND() in the calculation of these output cells, Each time Excel copies a value, RAND() is recalculated and it results in a different output. Without RAND() in the calculation, we would get a constant column which is of no use.

Here is a sample of the first 10 rows in the table (including the headers which do not form part of the WHAT IF table) and after formatting the numbers:

	G	H	I	J	K
1	Run ID	S/T One Time	S/T Ph 1	S/T Ph 2	Total
2		46,000	277,538	100,082	377,619
3	1	46,000	263,501	106,723	370,224
4	2	46,000	248,276	99,732	348,008
5	3	46,000	283,760	101,772	385,531
6	4	46,000	262,477	101,075	363,552
7	5	46,000	288,542	91,398	379,940
8	6	46,000	243,790	112,964	356,755
9	7	46,000	259,313	102,350	361,663
10	8	46,000	240,411	101,108	341,520
11	9	46,000	248,618	108,927	357,545

Notice that since the one time costs in Col H do not have a randomize input variable,

they will have the same value in all the runs or replications.

In this manner, we can replicate the output cells that we wish to analyze without having to replicate the whole formulation.

12.0 Models that Sample the Discrete Random Variable Distribution

In this chapter, we will be using a second type of sampling. Instead of one input variable varying over a whole range uniformly, our next variables will only be able to take specific or discrete values. (Note that though the term "discrete" is often used to refer to integers, our numbers will be real or fractional. "Discrete" refers to the brackets of the required values such as the table below).

How often a value arises is determined by a specified probability. For example, a price can take on any of these values: P1, P2, P3 or P4. Each of these values will arise with a probability of occurrence as shown in this table:

Price	Probability
P1	20%
P2	30%
P3	15%
P4	35%

Of course, the total must equal 100% to cover all possible prices. This kind of distribution cannot be represented by a closed function. We can only work with it via spreadsheet procedures. It is called a **Discrete Random Variable Distribution**.

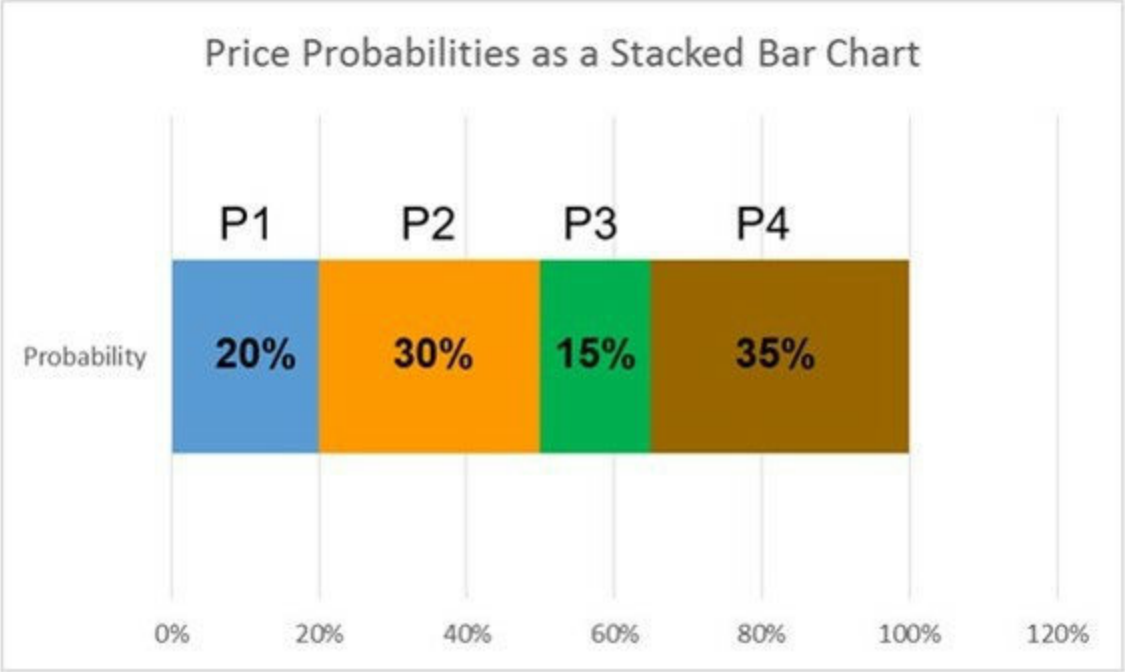
In order to simulate such a case, we need to generate a random number and test it. The results of the test will point to one of the rows and hence allow us to pick up the corresponding price. We will start with a visual approach.

First, in both approaches we use below, we need to compute the cumulative values of the probabilities:

Price	Probability	Cum
P1	20%	20%
P2	30%	50%
P3	15%	65%
P4	35%	100%

The need for cumulative values is best visualized through two graphs: a bar chart showing the probabilities and a horizontally stacked bar showing the cumulative percentages:

And this chart shows the same bars but stacked on their side:



By flipping the individual probabilities on their side, we notice that the end (or right side) of each stacked value is exactly the cumulative value of the probabilities. For example:

P1 goes from 0 to 20%
P2 goes from 20% to 50%
P3 goes from 50% to 65%
P4 goes from 65% to 100%

This is very useful. Since random number generated by RAND() vary from 0 to 1 then all we have to do is toss RAND() and find out under which "color" or bracket the value falls. That will point to the price in question. Here are some examples: a) If RAND() =

0.29, it falls in the orange bracket: use P2 in the model

b) If RAND() = 0.61, it falls in the green bracket: use P3 in the model
c) If RAND() = 0.89, it falls in the brown bracket: use P4 in the model
d) If RAND() = 0.03, it falls in the blue bracket: use P1 in the model

Examples of Discrete Random Variables Distributions

Here are some examples we can simulate corresponding to different events:

a) An item is available 20% of the time, not available at all 45% of the time and will be delivered late 35% of the time.

b) Mondays to Fridays (5 / 7 = 72.43%) our route takes 10 minutes. Saturdays (1 *7 = 14.28%) it takes 8 minutes and Sundays (1* 7 = 14.28%) it takes 5 minutes.

c) The possibility of winning a contract is broken down into 4 different cases: winning as proposed, winning with negotiated discounts, winning with contractual changes or not winning at all. Our marketing team is able to assign a probability to each of these possibilities.

In the next workout, we will introduce two methods to test RAND().

Workout 12: DISCRETE Distribution with IF(), MATCH() and INDEX()

The first Excel function that comes to mind is VLOOKUP(). Once you master it, you will find it has a wide ranging use. In our case, it has at least one shortcoming: there is no way to look up an item that is less than the first entry (first row) in a table. (See below VLOOKUP's other shortcomings).

Since we need to look up values corresponding to cumulative probabilities, we can only use VLOOKUP by adding a row before the first value. This is awkward and non-intuitive. We will therefore not use it in our models. (Refer to this site for a good explanation why VLOOKUP can more easily be replaced by other functions: Click Here).

Here are some other limitations of the VLOOKUP() function:

a) Your data is constrained to be in a table form that follows strict VLOOKUP rules. The lookup column has to be on the right side. The looked up values have to be in parallel vertical ranges and always on the left side.

b) Return values can only be specified by indexing the table. You cannot use other indexing functions to look up values elsewhere in the sheet or the workbook.

c) Most importantly for us is the limitation on the approximation match. As VLOOKUP() searches downwards in the table, if it finds a value in row N larger than the lookup

value, it retrieves the lookup value from row N-1. This is a major problem for cumulative tables such as those we used with the IF method. This requires changing the table to insert a line with 0 cumulative and shifting the looked up values by 1. Too elaborate.

Conclusion: we will resort to two methods:

a) Nest IF()'s which are easy to setup but unwieldy if there are more than 4-5 categories. (In fact, Excel restricts nesting to 7 levels).

b) INDEX() and MATCH() functions which have no limits but which are limitless.

A) Method 1: Use Nested IF()'s

We will always need the cumulative % column as shown below:

Price	Probability	Cum %
$10	0.17	0.17
$12	0.37	0.54
$14	0.27	0.81
$16	0.19	1.00

a) Generate a random number using RAND(). Example, RAND() = 0.6314

b) Find out in which bracket in the Cum % column it falls. Here, you have to read the Cum % as the **upper limit**. In our example, 0.634 falls in the 3rd row because it is greater than 0.54 (row 2) and less than 0.81 (row 3).

c) Read off the price in the same row. In our case, that would be $14

Note that if RAND() = 0.06, the search would assume that the first row of the Cum % ranges from 0 to 0.14. Our price would be $10.

We can use a nested IF statement to find out in step (b) where the RAND() value falls. The method is simple to use but has **two drawbacks**:

1) We can only nest IF's up to 7 levels.
2) Writing a long set of nested IF statements increases the risk of errors in the formulas and makes the formula difficult to review.

Here are the steps to use when we have 7 IF's or less. Use the next method in the case when you have more than 4-5 probabilities at most.

Step 1: create a new workbook and name it as you wish. In the Workouts Folder there is a fully solved model called **DISCRETE DISTRIBUTION with IF(), MATCH() and INDEX()**.

Rename the default sheet as "Disc Rand with IF".

Step 2: enter the following labels:

A1 = Price
B1 = Probability
C1 = Cum %
A9 = Sampled Price
A10 = Random Number
A11 = IF Statement Result

Step 3: Starting with A2, enter the price and probability values in the range A2:B8:

	A	B	C
1	**Price**	**Probability**	**Cum %**
2	$10	0.17	0.17
3	$12	0.37	0.54
4	$14	0.27	0.81
5	$16	0.19	1.00

Step 4: calculate the cumulative value of the probabilities in the range C2:C8. Start by entering one formula in C2:

C2 = SUM($A$2:A2)

Copy C2 down to row C7.

Step 5: since our purpose is to segment the range from 0 to 1.0 as per the cumulative distribution, if a random number is less than 0.17 (in the cumulative % column), we choose $10. If it is greater or equal to 0.17 and less than 0.31, we choose $12 and so on.

Let B8 =RAND()

In B9 enter 4 nested IF's to look up 4 values in 4 rows:

B11 = **IF**(B8<C2,A2, **IF**(B8<C3,A3, **IF**(B8<C4,A4, **IF**(B8<C5,A5))))

For example, if B10 = 0.4171439, it falls in the range >= 0.17 (C2) and < 0.54 (C3) which means, we shall use the second IF or price = $12 in A3.

Step 6: press F9 a few times to generate other random numbers and check that the IF statement is finding the right price.

B) Method 2: Use INDEX() and MATCH()

Problem statement: to use MATCH() to search for a value in the cumulative probability column of tables similar to the above example. On finding the value in a cell, it returns an index (an integer) that points to that cell. We can then use INDEX() to look up the value in the cell containing the corresponding price. INDEX() is similar to OFFSET() but is restricted in some ways. In our context, the restrictions of INDEX() make it easier to use. We will embed MATCH() within INDEX().

Step 1: stay with the current workbook. Right click on the tab of the first sheet "Disc Rand with IF" and copy it to a new sheet. Call it "Disc Rand with MATCH+INDEX".

This saves you formatting the first 3 columns and the first 4 rows. Build it up as follows. All values shown in this image are constant. Formulas will be presented below for columns E, F and G:

	A	B	C	D	E	F	G
1	Price	Probability	Cum %	Values to Look Up	MATCH()	Corrected	INDEX()
2	$10	0.14	0.14	0.05			
3	$12	0.17	0.31	0.25			
4	$14	0.28	0.59	0.55			
5	$16	0.20	0.79	0.63			
6	$18	0.14	0.93	0.85			
7	$20	0.07	1.00	0.95			

In col D, we will place fixed values to use for testing. Each one is less than the corresponding cumulative % in its own row and greater or equal to the cumulative % in the row below. For example, 0.55 in D4 is >= than 0.31 (C3) and < 0.49 (C4).

In col E, we will apply MATCH() to show the correct indices.
In col F, we will correct the #N/A.
In col G, will get the INDEX() function which we use to get the prices using the indices from col F.

Alert 1: #N/A is placed in a cell if when you use MATCH(), the value being checked is less than that in the column being checked. In our case, 0.05 in D2 is less than 0.14 in C2 so we get #N/A in E2.

Alert 2: since MATCH() always points to the upper row, we need to add 1 in the final indexing value as will be shown in col G below.

Step 2: use MATCH() to find the location of the value in the cumulative column (C2:C7) that is just larger than our random number in the range D2:D7.

Let E2 = MATCH(D2, $C$2:$C$7, 1)

This reads as follows:

a) Search for the value found in D2 (which is 0.05) in the range C2:C7.

b) As MATCH() progresses down the column, it stops as soon as it finds a value larger than 0.05 and gives an index which points to the previous row. Since we do not have a previous row when placing MATCH() in D2, this value returns #N/A which we shall deal with below.

c) Copy E2 down to E7.

d) Let us read what happens in E5 = MATCH(D5, $C$2:$C$7, 1). D5 = 0.75 so MATCH() will stop at 0.79 (the fourth cell or index = 4). This is the silly part: MATCH() will now point to the previous row (index = 3). This value is shown in E5.

We need to correct the #N/A error and then proceed to look up the price using INDEX().

Step 3: some formulas give #NA under certain conditions. Excel provides us with a

function called IFNA() which tests if the result of such a formula is #NA. This is an IF statement so behaves like one but without a logical condition. You will get two arguments in IFNA(). One is when there is a #NA in your cell and the next one, when there is not. Simply insert the MATCH() function within the IFNA() function: F2 = IFNA(**MATCH(D2,$C$2**:**$C$7,1)**, 0)

We read this as follows: if the results is #NA, place 0 in the cell. If not, place the result of the MATCH() function.

Copy F2 down to F7. We now have a series from 0 to 5 (based on the values we manually entered in E2:E7). They would be the same as the values in D2:D7 except for D2 which shows the #N/A error.

Step 4: let G2 =INDEX ($A$2:$A$7, F2+1) which reads as follows: pickup the index found in F2 and add 1 to it. The result = 2. Use 2 as the index within the location A2:A7 where A2 has an index = 1. This returns the price = $10.

Copy G2 down to G7. You can see that the combination MATCH/INDEX with the proper adjustments are picking up the right price.

Step 5: enter the following labels:

A9 = Sampled Price
A10 = Random Number
A11 = MATCH/INDEX

Step 6: Let B10 =RAND()

In B11 enter this long function:

B11 = INDEX($A$2:$A$7, **IFNA(MATCH(B10,$C$2**:**$C$7,1),** 0) + 1)

Step 7: press F9 a few times to generate other random numbers and check that the IF statement is finding the right price.

To summarize: when we need to look up a value in a range (usually a column), we use the INDEX() to get the location of the MATCH() but apply the IFNA() to ensure that MATCH() does not return #N/A for the first row.

This procedure will be used whenever we have a distribution that does not have a native inverse function in Excel such as the Poisson Distribution which we will use in Part 2 of this eBook.

Workout 13: The Shortest Route Duration (DISCRETE DISTRIBUTION)

Purpose: to prepare 1000 simulations runs to find out, with confidence, which of 2 routes is shorter. Each route has 3 durations depending on a 3 different probabilities.

The workout aims at using the IF statement and the MATCH() function presented in the previous workout.

Problem statement: you can drive from home to your office via two different routes. If you take Route 1, you will get to the office after 4, 8 or 6 minutes. Statistics have shown that 20% of the time, you will take 4 minutes, 30% of the time, you will take 8 minutes and 50% of the time, it will be 6 minutes. The same logic is applied to Route 2 but with different durations and related probabilities.

The approach is to enter a sampled duration for each route in one row. In the same row, we can then find out which route is shorter. We will simulate 1000 runs and analyze them statistically.

A) Use the IF Statement to Sample Discrete Random Variables

Step 1: create a new workbook and name it as you wish. In the Workouts Folder there is a fully solved model called **Shortest Route Duration (DISCRETE DISTRIBUTION)**.

Create the following sheets: Runs IF and Constants. These will cover the Nested IF()'s version of our model.

Step 2: create in the Constants sheets entries for the two discrete random distributions shown earlier by manually entering the probabilities and the durations:

	A	B	C
1	Route 1		
2	Probability	Cum %	Duration in Min
3	0.20	0.20	4
4	0.30	0.50	8
5	0.50	1.00	6
6			
7	Route 2		
8	Probability	Cum %	Duration in Min
9	0.15	0.15	4
10	0.65	0.80	7
11	0.20	1.00	3

It is not obvious which route to take based on "long run" statistics. We will simulate the two routes and analyze the output. (This problem can be solved directly, using the Expected Monetary Value (EMV) method. However, our purpose is the use of the Discrete Distribution).

In the two ranges B3:B5 and B9:B11, enter the formula for the cumulative probabilities:

B3 = SUM($A$3:A3) then copy B3 down to B5
B9 = SUM($A$9:A9) then copy B9 down to B11

Step 3: in the Runs IF sheet, enter the following labels:

A1 = Run ID

B1 = Rand
C1 = Route 1
D1 = Rand 2
E1 = Route 2
F1 = Best Time?
G1 = Which Route?

	A	B	C	D	E	F	G
1	Run ID	Rand 1	Route 1	Rand 2	Route 2	Best Time?	Which Route?

Step 4: use Excel's autofill facility to generate the sequence 1, 2... 1000 in the range A2:A1001.

Step 5: generate two sets of random numbers using RAND():

B2 = RAND() for the sampling of Route 1
D2 = RAND() for the sampling of Route 2

The reason for this step is to avoid having to use RAND() more than once within the IF() function. If we do, each instance will generate a different value and invalidate the test.

Step 6: in C2 and D2, we will be using samples from the Discrete Distributions placed in the Constants sheet. We will first use the IF statement method. It uses the random number generated in B2 and goes through the cumulative values in the range B3:B5 in the constants sheet. As soon as it finds the value that is just larger than the random number, it will pick up the corresponding duration. For example, if the random number generated is 0.56, then the segment of probabilities that applies is the third one: greater or equal to 0.5 and less than 1.0. The duration is therefore 6 minutes. Here is the IF statement in full: C2 = IF(B2<Constants!$B$3,Constants!$C$3, IF(B2<Constants!$B$4, Constants!$C$4,Constants!$C$5))

For Route 2 in E2 we use the same IF statement but the other random number found in D2:

E2 = IF(D2<Constants!$B$9, Constants!$C$9, IF(B2<Constants!$B$10, Constants!$C$10, Constants!$C$11))

Step 7: find the shorter duration and place it in F2:

F2 = MIN(C2, E2)

Step 8: use an IF statement to place a text result and place the text in G2. The two text statements are found in the header cells C1 and E1:

G2 =IF(C2<E2, $C$1, $E$1)

Apply conditional formatting to highlight Route 1 in G2 in orange.

Step 9: copy the range B2:G2 down to B1001:G1001.

Step 10: to analyze the results, we apply simple tests to the range I1:J7 as follows. Enter the labels:

I1 = Results
I2 = Count Route 1 Shorter
I3 = Count Route 2 Shorter
I4 = Number of Runs
I5 = Count Ratio R2 / (R1+R2)
I6 = Average Route 1 Duration
I7 = Average Route 2 Duration

Step 13: here are the tests:

J2 = COUNTIF(G2:G1001,$C$1)
J3 = COUNTIF(G2:G1001,$E$1)
J4 = COUNT(A:A)
J5 = J3 / (J3+J2)
J6 = AVERAGEIF(G2:G1001, $C$1, F2:F1001)
J7 = AVERAGEIF(G2:G1001, $D$1, F2:F1001)
J8 = J7/J6

Results: these are the results that we need, simple but instructive:

	I	J	K	L
1	**Results**			
2	Count Route 1 Shorter	432	=COUNTIF(G2:G1001,$C$1)	
3	Count Route 2 Shorter	568	=COUNTIF(G2:G1001,$E$1)	
4	Number of Runs	1000	=COUNT(A:A)	
5	Count Ratio R2/(R1+R2)	56.80%	=J3/(J3+J2)	
6	Average Route 1 Duration	5.44	=AVERAGEIF(G2:G1001,$C$1,F2:F1001)	
7	Average Route 2 Duration	4.80	=AVERAGEIF(G2:G1001,$E$1,F2:F1001)	
8	Ratio R1/R2 Duration	0.88	=J7/J6	

This shows that Route 1 is 13% shorter and that over 1000 runs, it came up 56.8% of the time.

B) Use MATCH() and INDEX() to sample Discrete Random Variables

Purpose: to use the MATCH() and INDEX() functions instead of the IF Statement.

Step 1: use the same workbook you used above. In the Workouts Folder there is a fully solved model called **Shortest Route Duration (DISCRETE DISTRIBUTION)**.

Right click on the tab of the Runs IF sheet and copy it to a new sheet. Rename the new sheet: "Runs MATCH".

Step 2: replace the IF statement in C2 with the MATCH/INDEX formula as follows:

C2 =INDEX(Constants!$C$3:$C$5, IFNA(MATCH(B2, Constants!$B$3:$B$5,1),0)+1)

We can read this as follows (starting with MATCH(), the most inward function and reading IFNA() and INDEX() outwards):

a) Match the random number in B2 with the range B3:B5 in the Constants sheet using approximate matching: (MATCH(B2,Constants!$B$3:$B$5,1)

b) Use IFNA() to check whether MATCH() returns a #N/A or not. If it does, replace that by 0: IFNA(MATCH(B2,Constants!$B$3:$B$5,1),0). The Match function would then point to the previous row.

c) Use INDEX to retrieve a value from the range C3:C5 in the Constants sheet offset by the rows found in the MATCH() function above. Add to the rows 1 to ensure they point to the right duration.

This function will return one of the 3 durations based on the generated sample or random number.

Step 3: do the same for Route 2 in E2:

> E2 =INDEX(Constants!$C$9:$C$11, IFNA(MATCH(D2, Constants!$B$9:$B$11,1),0)+1)

The rest of the sheet does not require any change. The results should be the same since we did not change any of our constants, simply the sampling method.

13.0 Models with Primary and Secondary Runs: Hospital Lab Tests

Many models will require you to simulate events at two levels. At the primary level, you will conduct a simulation using the methods we have presented so far. In some models, the row being simulated requires a further simulation. This is the secondary level of simulation.

The Model sheet for such models will have two ranges. The bottom range is the primary range. It contains our typical thousands of runs. The top range is the secondary range. It contains a limited number of rows, usually in the 10s. Each one of the primary rows will get further simulated using the secondary block. The results of the simulation of the secondary block are then transferred to a specific row in the lower, primary block. The section that contains the primary runs is called the Runs section. The section that contains the secondary runs is called the sub-runs section.

The following capture is a visual example of the sub-runs and runs ranges:

Sub Runs ID	A	B	C	D
1	0.52	8.28	5040	90%
2	0.45	24.27	6962	2%
3	0.89	4.10	8182	6%
4	0.90	28.99	5046	98%
5	0.92	22.60	4841	72%
6	0.11	37.78	1055	82%
7	0.80	25.14	4126	18%
8				
9				
10				
Results	**0.66**	**151.16**	**8182**	**52%**

Runs ID	Ratio	Total	Max	Min
Client 1	0.48	169.39	11232	76%
Client 2	0.55	138.65	11222	59%
Client 3	0.66	151.16	8182	52%
Client 4				
Client 5				
Client 6				

a) The primary runs block is shown in the lower part of the sheet. It has M rows (to be defined by us). Each Row summarizes the results of 1 client.

b) The sub-runs block (shown above because in the simulation, it happens first). The number of items in the sub-runs block varies: randomized limit, total value, total weight, end of day, 300 days, *etc.* In the above example, the limit was reached after 7 sub-runs. The totals in the results row are then moved to client 3 in the lower block.

We will have 2 loops. One loop is controlled by a VBA module (short and simple) that goes from 1 to a predefined value M. Another is a loop controlled by formulas that generates up to N sub-runs whose results go into the main runs block.

The method is adapted from a problem posed in Christian Albright and Wayne Winston's wonderful book: “Management Science Modeling” (revised 3rd edition). The problem is presented using @RISK. However, its logic forms the basis of primary and secondary runs that we will use frequently in this eBook. The problem is called “Bidding for a Government Contract” and is in Chapter 12, Example 12.1 on page 653.

Workout 14: Hospital Lab Tests Model - How to Generate Sub-runs

Purpose: this workout introduces a model that uses a two level structure that will also be applied in more complex models later on.

Problem statement: a small sized hospital outsources all of its lab tests to a nearby lab. Recently, they realized it might be cheaper to setup their own testing facility. They usually conduct 10 different types of tests. (This is not a realistic figure, but once the model is validated, it can be increased).

The hospital found out that the viability of its new venture is strongly dependent on the number of lab technicians it needed to recruit. Their aim is to find out whether the cost of conducting their own tests would be cheaper than the cost of outsourcing them. The hospital could not directly estimate the number of lab technicians needed for the in-house facility.

It resorted to Monte Carlo Simulation. For that, it needed to have more detailed information about its tests. Here is what the hospital knew or had to know:

a) The hospital reviewed all the tests it had sent to the external facility over a period of 1000 days. During that period, they had sent out a total of 59,995 tests.

b) A frequency table was developed that mapped the 59,995 tests into each of the 10 types of tests. The count is shown in Col B in the table below.

c) The cumulative % frequencies (Col C) was calculated as per our standard presented in Chapter 5.0.

d) To measure the duration of the tests, the hospital conducted sample internal tests. Each test was conducted by 1 lab technician. For each type of test, 50 tests were conducted and their duration noted. The hospital could then calculate the **average** duration for each type of test (Col D) and the **standard deviation** (Col E) for each type. (If you are not happy with the meaning of average and standard deviation, please refer to the Appendix in Chapter 15.0).

	A	B	C	D	E
1	**Test**	**Freq**	**Cum %**	**Mean**	**St Dev**
2	Test 1	6,184	0.10	15.00	3.00
3	Test 2	8,845	0.25	24.00	3.00
4	Test 3	2,901	0.30	34.00	4.00
5	Test 4	1,781	0.33	14.00	3.50
6	Test 5	9,396	0.49	5.00	1.10
7	Test 6	12,308	0.69	51.00	4.60
8	Test 7	5,793	0.79	70.00	5.80
9	Test 8	3,922	0.85	6.00	0.80
10	Test 9	1,841	0.88	29.00	2.30
11	Test 10	7,024	1.00	8.00	0.95
12	**Total**	**59,995**			

If we assume that the number of working hours per person per day is 8, how can we use this data to find out the number of lab technicians we need to recruit?

The Procedure:

The solution would be to simulate 1000 days. These would be our primary runs. Starting at time = 0 on each day, one technician will conduct one test after the other until the end of the working day is reached, i.e., after 8 hours or 480 minutes. (The model has to check before each test starts whether the previous test went over the 480 minutes available in the day. If it does, the test in the previous row will be finished and included in the simulation. The new test will not be conducted. This means that most days will have a few minutes of work beyond the allocated 480 minutes).

Which test the technician will be conducting will be randomized. By the end of the 1000 days, we will know how many tests have been conducted by 1 technician. We can then divide the 59,995 tests by the result to find out how many technicians would be needed for the 59,995 tests. Working out their cost would tell the hospital whether it is feasible or not to have an in-house testing facility.

The primary block: this is made up of M = 1000 rows, one for each day. It is defined in the range A55:K1055 in the Model sheet. Each Row will have 10 cells that count the number of tests completed during that day for each type. (Effectively, the rows in the primary block are calculated after those of the secondary block of sub-runs).

The secondary or sub-runs block: this is where we simulate consecutive tests in a single day. It is defined in the range A1:G51 in the Model sheet. Our sub-runs block is made up of a maximum of 50 tests to be conducted in one day.

Why 50? This is an "experimental value". We are guessing that in one day, we do not have more than 50 tests. If we do, the model will crash as Excel goes beyond row 51. We cannot know this value in advance. During simulation, we will find out if Excel is going beyond 50 tests in 8 hours or not. Later on, as you monitor the secondary or sub-runs block, you will see that in each day, we would have from 15 to 30 tests. Does this make sense? It does, if you sum up the means of the tests in the range D2:D11. The total is 256 minutes. Since we have 480 minutes in one day, a rough engineering guess would estimate the average number of tests in one day to be 40.

If the simulation requires more than 50 sub-runs, we would then increase N. Some analysts prefer not to wait that long and simply use a large value of N, say 250. Excel is general in rows and in performance. N is arbitrary and it does not affect the results.

A) Setup the constants and the sub-runs block

Step 1: create a new workbook and save it under any name. Ensure that you save it as a **macro enabled** workbook so you can import the **GenerateRuns** VBA procedure discussed later in this chapter. In the Workouts Folder there is a fully solved model called **Hospital Lab Tests Model - How to Generate Sub-runs**.

We need 3 sheets: Model, Results and Constants.

Step 2: setup the Constants sheet by entering the following tables:

	A	B	C	D	E
1	**Test**	**Freq**	**Cum %**	**Mean**	**St Dev**
2	Test 1	6,184	0.10	15.00	3.00
3	Test 2	8,845	0.25	24.00	3.00
4	Test 3	2,901	0.30	34.00	4.00
5	Test 4	1,781	0.33	14.00	3.50
6	Test 5	9,396	0.49	5.00	1.10
7	Test 6	12,308	0.69	51.00	4.60
8	Test 7	5,793	0.79	70.00	5.80
9	Test 8	3,922	0.85	6.00	0.80
10	Test 9	1,841	0.88	29.00	2.30
11	Test 10	7,024	1.00	8.00	0.95
12	**Total**	**59,995**			
13					
14	Hours per Day	8.00			
15	Minutes per Day	480			

Enter 8 in B14 and let B15 = B14 * 60. This will be the cutoff time in the simulation of sub-runs.

Step 3: in the Model sheet, prepare the VBA procedure, the Control Values and the macro button. A fully detailed procedure is presented in the Appendix in Chapter 18.0.

Setup the **Control Values** in the Model sheet for the VBA Module:

a) Enter "Max Number of Runs" in I1 and "Current Run ID" in I2.

b) Enter 1000 in J1 and keep J2 blank.

c) Change L1 to I1 and L2 to I2 in the VBA procedure accordingly.

d) Enable iterative calculations in the FILE *OPTIONS* FORMULAS tab.

Step 4: **prepare the headers of the sub-runs block and the Run IDs**.

The sub-runs block is in A1:G51. The sub-runs block will include a Row for each test conducted in one day (no more than 50 are expected). After each sub-run is simulated, we will know the finish tie of that test. We will test for when the finish time of the next test is > 480 minutes. That is when we stop adding tests or sub-runs and then transfer the results to a single row in the primary runs block.

a) Enter the labels:

A1 = Sub-Run ID
B1 = Start Time

C1 = Test
D1 = Mean
E1 = St Dev
F1 = Duration
G1 = Finish Time

b) Use autofill to generate the sequence 1, 2... 50 in the range A2:A51.

Unlike the Run ID's in the primary block below, these have no computational value.

Step 5: prepare the **start times** in the range B2:B51. We will prepare both B2 and B3 because this model has something peculiar. We see it in most runs/sub-runs models. **The first row in the sub-runs block is different from the rest**. This is because we have to specify a 0 for the start of the day in B2 whereas the other start times in the cells below B2 are governed by specific formulas, namely the brought forward from the G column or the finish time of the previous test. So that B3 = G2 and B4 = G3 and so on. (But there are IF statements for these values that we will explain below).

B2 = 0
B3 = IF(G2>Constants!$B$15, "", G2)

This formula reads as follows: G2 is the finish time of the test in the previous row. If G2 is larger than the number of minutes per day, stop the simulation of the sub-runs (or the daily tests) by placing a null value in B3. (In the steps below, we will be applying a test on B3 to fill the rest of the cells in Row 3 with null values for the same conditions.

If G2 is not larger than 480 minutes, then copy the value of G2 into B3. This will force the finish time of Row 2 to be the start time of Row 3.

We can state this in general terms: say we are in row X. If the finish time of row X-1 is greater than 480, place a null value in the start time. (Subsequent sub-runs will note be generated). This assumes that the technician will stay after office hours to complete the test in row X-1. If the finish time is < or = to 480, then let the start time of row X = the finish time of row X-1.

Copy B3 down to B51.

Step 6: C2 = **the type of test**. Establish which type of test is to be conducted by sampling the **Discrete Distribution** setup in the Constant sheet. Scan the cumulative distribution of the probability in the range D2:D11 in the Constant sheet. Use the technique developed earlier for using MATCH() and INDEX().

C2 = IF(B2="", "", INDEX(Constants!**$A$2**:**$A$11**, IFNA(MATCH(RAND(), Constants!**$D$2**:**$D$11**,1),0)+1))

To reduce the complexity of the formula in B2, its logic is presented from inside out.

a) Use RAND() within MATCH(). It will locate the value within the brackets of the cumulative distribution in Col D in the Constants sheet.

b) Embed MATCH() within IFNA() to ensure that if #N/A is returned, it gets replaced by a zero.

c) Embed the above within INDEX() to locate the name of the test found in Col A corresponding to the bracket of the cumulative distribution found in Col D.

d) Embed the whole lot in an IF statement that checks if B2 = "". If so, this means that the sub-runs have reached a Finish Time that is > 480 minutes and we do not have to look up a test. (This is never the case in the first Row, but we include it to have the first Row use the same formulas as the lower rows, except for B2).

Copy C2 down to C51.

Step 9: **the mean and the standard deviation** are entered in D2 and E2 in the Model sheet. (We could have looked them up in the following step directly from the Constants sheet. However, the VLOOKUP would have needed to be embedded in the sampling formula which would have made it complex to enter and check). This step is therefore, just a transfer of the two parameters from the Constants to the Model sheet.

Use VLOOKUP to locate the mean and the standard deviation of the type of test as entered in C2 in the Model sheet. For example, Test 4 has a mean of 14.00 and a standard deviation of 3.5 minutes. These are picked up from the range E2:E11 and F2:F11 in the Constants worksheet and placed in D2 and E2: Place the LOOKUP formulas within the IF statement that tests if we have an entry in this Row or not (as we did above):

D2 = IF(B2="","", VLOOKUP(C2, Constants!$A$2:$F$11, 5, FALSE))
E2 = IF(B2="","", VLOOKUP(C2, Constants!$A$2:$F$11, 6, FALSE))

Notice how we will always check if B2 (or the cells below it) is a null value signifying the end of testing for that day.

Copy D2 down to D51 and E2 down to E51.

Step 10: in F2, we need a **sample duration** for the test identified in C2. We have not encountered the Normal Distribution yet. We will discuss sampling from Normal Distributions in Part 2 of this eBook. It is enough to say that NORM.INV() will provide us with a random sample drawn from a population whose mean and standard deviation are given in D2 and E2 in the Constants sheet (for the particular test whose type is in C2). If we supply the NORM.INV() formula with a cumulative probability, it will provide us with a sample (more of this logic later).

F2 = IF(B2="","", NORM.INV(RAND(), D2, E2))

Again, we test if B2 = "" and place "" if true.

Copy F2 down to F51.

Step 11: to calculate the **finish time** of the test, enter in G2 the total of the starting time (B2) and the duration (F2).

G2 = IF(B2="","",B2+F2)

Again we test for B2 = "".

Note that G2 is the finish time for Row 2. It will become the start time for Row 3 and so on down the set of tests. We do not test for the end of day in the current row but leave that to the start time of the next row.

Copy G2 down to G51.

Note: without running the VBA module, you will not be able to observe results in the various cells. Some might even be wrong. Therefore, you cannot check the above formulas just yet! The next step will allow you to.

Step 12: Enter 50 in J1 to test the model with a small number of loops. (Normally, we should use values in the thousands). Press the GENERATE RUNS button. You will see the sub-runs being filled and then refilled.

This next image is a sample showing the test sub-runs. During this day, the technician finished 18 tests with the last one finishing at 503.21 minutes. Therefore, the sub-runs had to stop. The formulas in Row 19 and those below it will all give null values since B19 > 480 minutes.

	A	B	C	D	E	F	G
1	Sub-Run ID	Start Time	Test?	Mean	St Dev	Duration	Finish Time
2	1	0.00	Test 4	14.00	3.50	15.33	15.33
3	2	15.33	Test 6	51.00	4.60	48.91	64.24
4	3	64.24	Test 5	5.00	1.10	3.31	67.55
5	4	67.55	Test 1	15.00	3.00	14.62	82.17
6	5	82.17	Test 5	5.00	1.10	5.82	88.00
7	6	88.00	Test 2	24.00	3.00	22.96	110.96
8	7	110.96	Test 3	34.00	4.00	29.90	140.86
9	8	140.86	Test 4	14.00	3.50	16.88	157.74
10	9	157.74	Test 5	5.00	1.10	3.15	160.89
11	10	160.89	Test 2	24.00	3.00	20.47	181.36
12	11	181.36	Test 6	51.00	4.60	40.11	221.47
13	12	221.47	Test 6	51.00	4.60	56.00	277.47
14	13	277.47	Test 7	70.00	5.80	69.71	347.18
15	14	347.18	Test 10	8.00	0.95	9.37	356.55
16	15	356.55	Test 2	24.00	3.00	28.61	385.16
17	16	385.16	Test 10	8.00	0.95	8.81	393.97
18	17	393.97	Test 6	51.00	4.60	54.84	448.80
19	18	448.80	Test 6	51.00	4.60	54.41	503.21

B) Setup the Primary Runs in the range A55:K1055

If we place 1000 in J1, the VBA will pick up that value and use it to loop from 1 to 1000. Each loop represents a day of tests. It will provoke a set of sub-runs in the upper block (as we showed above). From those sub-runs, we collect the number of tests completed that day for each type (this is really a frequency count). The frequency count

(made up of 10 values) will be placed in a Row in the main runs block. Each Row from 56 to 1055 will contain a histogram of the tests conducted on that day.

But which row? Here is where we use the Albright/Winston technique whereby we test if the RUN ID of the primary block is equal to that of the VBA loop.

Step 1: in A55 enter the label "Run ID". In the range B55:K55 use autofill to generate the sequence Test 1, Test 2... Test 10. These represent the test indices which we will use when accumulating the frequency of the test.

Step 2: use autofill to generate the sequence 1, 2... 1000 in the range A56:A1055. These are the Run ID's that will be used to check against the Run ID generated by the VBA procedure in J2.

Step 3: since we need to sum the number of tests, we enter the Sums of the tests simulated in the sub-runs block in Row 53.

a) Enter the label "Sum" in A53

b) Let B53 = SUM(B56:B1055)

c) Copy B53 from C53 to K53

This Row is a histogram of the tests or a frequency count of test types for the whole simulation. The rows 53 and 55 are shown below (to reduce the spread of the image, we are only showing the histogram for the first 5 test types):

	A	B	C	D	E	F
53	Sum	1,873	2,738	907	560	2,833
54						
55	Run ID	Test 1	Test 2	Test 3	Test 4	Test 5

Step 4: we will now present a formula that can be used throughout the range B56:K1055. We will enter it into B56 then copy it to the right then downwards.

For each iteration or a specific value of the Run ID in J2, we will have several tests whose total duration is just > 480 minutes. Our objective here is to count the number of times each test appeared in the sub-runs above or in one day. Each row in the primary block of runs will actually be a daily frequency table for the sub-runs of that day (but horizontally placed). The COUNTIF() formula is the following:
COUNTIF($C$2:$C$51,B$55)

Since B55 contains the text “Test1”, then this gives us the number of times this test appeared in the first day.

As we did in the previous workout, we need to ensure that all cells in the block are filled in such a way that:

b) Cells in the Row corresponding to the Current Run ID are counted by COUNTIF().

b) Cells in the rows after Current Run ID are blank or have null values.

c) Cells before the Current Run ID (found in J2) are left as they are.

This is the formula which embeds the above COUNTIFS() formula:

B56 = IF($A56=$J$2, **COUNTIF($C$2:$C$51,B$55)**, IF($A56>$J$2,"",B56))

The formula contains two IF statements that check the above conditions. It compares the value of the Current Run ID (from J2) with the Run ID found in A56. There are three cases corresponding to the above list:

a) If the Row we are in (Run ID or 56 in our case) = the Current Run ID in J2, we count the number of tests conducted in this sub-run and place them in B56 using the COUNTIF() function. If not, we try the second IF statement.

b) If the Row we are in now is > the Current Run ID, this means this Row has not been calculated and must be blank. We enter a null value "".

a) If the Row we are in now is < the Current Run ID, this means that the Row has already been calculated in a previous loop. We retain the value B56 in B56. (That is why we need to enable iterative calculations).

Step 5: copy the above formula in B56 to the right and up to K56 since all the tests are the same.

Copy the range B56:K56 down to B1055:K1055.

Step 6: press the GENERATE RUNS button to generate the 1000 primary runs.

The model is now complete. The first 10 runs are shown in this capture:

53	Sum	1,886	2,627	882	564	2,846	3,755	1,828	1,189	550	2,126
54											
55	Run ID	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
56	1	2	2	0	1	2	7	0	1	0	6
57	2	7	2	1	0	2	2	2	1	1	3
58	3	1	2	1	0	1	4	2	1	1	1
59	4	0	1	0	0	3	7	1	0	1	2
60	5	3	1	0	0	3	3	3	2	1	2
61	6	1	4	1	0	2	2	3	0	1	1
62	7	0	2	0	0	1	7	2	0	0	2
63	8	3	4	0	0	0	4	2	0	0	2
64	9	1	4	0	2	1	5	1	0	1	3
65	10	2	4	1	0	5	2	2	1	1	3

Note: if your sums in row 53 do not look realistic, there could be due to 2 problems:

a) You should enable "Iterative calculations" on the FORMULAS tab on the OPTIONS page

b) You may have changed around in the sheet without regenerating the runs. Click on the button GENERATE RUNS and get a fresh set of results.

C) Analyzing the Results

In the Model sheet, the counts for each type of test of the 1000 rows are stored in Row 53. The Row 53 is a frequency table of the tests that were conducted during 1000 days

of work.

The assumption was that one lab technician was conducting the tests. We can compare the total tests conducted by 1 technician with the 59,995 conducted in 1000 days by the external facility. That will give us our answer.

However, our simulation resulted in a particular histogram or a frequency table. We need to test our simulation to ensure that our test sample comes from the same population as the set of tests conducted by the external facility (which was entered in the Constants sheet). We need to conduct a goodness of fit test between the two histograms.

Apply the Chi-Squared test to check the fit. Refer to Workout 7: Test the Uniformity of RAND with Chi Squared for the required steps).

Let us layout the two frequencies side by side in the Results sheet and see what they tell us.

Step 1: compare the results visually:

a) Copy the range A1:C12 from the Constants sheet and paste it into the same range in the Results sheet.

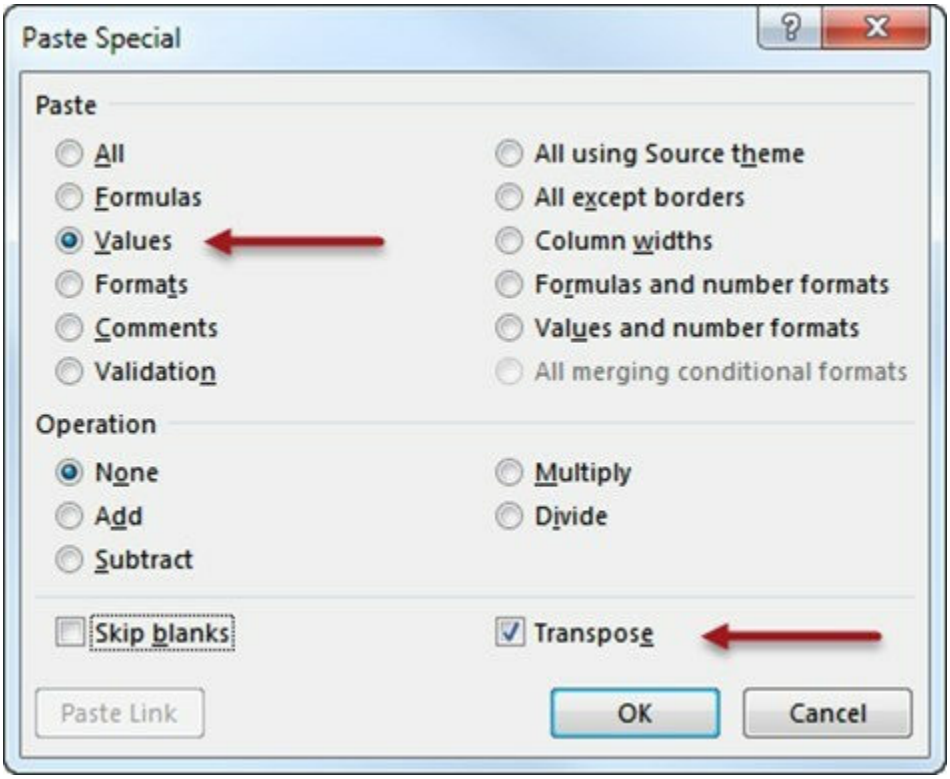
b) Rename B1 to "Outsourced Freq".

c) Enter the labels "Simulated Freq" and "Freq %" in D1 and E1

Step 2: copy the histogram of the simulation from Row 53 in the Model sheet. Note that the histogram in the Model sheet is in the horizontal range B53:K53. It needs to be transposed into the vertical range D2:D11 in the Results sheet which is a vertical range. We need to use the Paste Transpose. Since B53:K53 contains formulas, we also need to Paste as Values.

a) Copy the range B53:K53 from the Model sheet

b) Click on the cell D2 in the Results sheet then select PASTE SPECIAL:



c) Select PASTE SPECIAL and enable both "Values" and “Transpose” options.

In Col D under “Simulated Frequency” we now have the simulation histogram.

d) Let D12 = Sum(D2:D11)

e) Find the Freq % for the range D2:D11 as we've done before using the formula in E2 = D2 / $D$12.

f) Copy E2 down to E11.

	A	B	C	D	E
1	**Test**	Outsource Freq	Freq %	Simulated Freq	Freq %
2	Test 1	6,184	0.10	1,883	0.10
3	Test 2	8,845	0.15	2,626	0.14
4	Test 3	2,901	0.05	881	0.05
5	Test 4	1,781	0.03	565	0.03
6	Test 5	9,396	0.16	2,843	0.16
7	Test 6	12,308	0.21	3,754	0.21
8	Test 7	5,793	0.10	1,830	0.10
9	Test 8	3,922	0.07	1,189	0.07
10	Test 9	1,841	0.03	550	0.03
11	Test 10	7,024	0.12	2,124	0.12
12	**Total**	**59,995**		**18,245**	

Conclusion: the two frequency % columns C and E are almost identical when we use a precision of 2 decimals. If you try the Chi-Squared test, it will show you that there is no significant difference between the two population: the outsourced frequency table and

the simulated run for the same period.

Secondly and more importantly, the hospital found out that in 1000 days it can handle 18,245 tests. (This is the sum of counts in Row 53 in the Model sheet). This means one lab technician can handle 18,245 tests in 1000 days. Yet, the number of tests outsourced was 59,995.

Dividing 59,995 / 18,367 = 3.288. We need 4 lab technicians (testing 8 hours a day) to replace the outsourcing of tests.

Let us Play

We can extend our model to handle the following:

a) Include a larger number of test types.

b) Include lunch and randomized coffee breaks in the simulation.

c) Simulate setup and cleanup times for each type of test.

d) For all tests that have gone over 480 minutes, calculate the time and include it in the costing as overtime.

e) Simulate the breakdown of machines. In Part 2 of this eBook, we have a chapter that presents models in reliability engineering.

f) In our current model, we assumed that the last test in the day will be completed. This means some minutes will be completed after office hours.

	A	B	C	D	E	F	G
1	Sub-Run ID	Start Time	Test?	Mean	St Dev	Duration	Finish Time
2	1	0.00	Test 5	5.00	1.10	5.78	5.78
3	2	5.78	Test 5	5.00	1.10	5.76	11.54
4	3	11.54	Test 3	34.00	4.00	33.64	45.18
5	4	45.18	Test 10	8.00	0.95	5.91	51.09
6	5	51.09	Test 7	70.00	5.80	67.87	118.96
7	6	118.96	Test 6	51.00	4.60	52.43	171.38
8	7	171.38	Test 6	51.00	4.60	51.69	223.07
9	8	223.07	Test 10	8.00	0.95	7.79	230.86
10	9	230.86	Test 6	51.00	4.60	47.32	278.18
11	10	278.18	Test 3	34.00	4.00	38.10	316.29
12	11	316.29	Test 5	5.00	1.10	3.43	319.71
13	12	319.71	Test 7	70.00	5.80	80.58	400.30
14	13	400.30	Test 6	51.00	4.60	54.50	454.79
15	14	454.79	Test 7	70.00	5.80	77.78	532.57

In the above example, on one day, 14 tests were conducted. The last one had a duration of 77.78 but it spilled over the 480 limit by 532.57 - 480 = 52.57. This is almost one hour. We can include a calculation of overtime in the model and add it to the total cost of staff when comparing with the outsourced costs.

One of the useful aspects of Monte Carlo Simulation is the ability to mix models.

14.0 Sensitivity Analysis and Simulation

Sensitivity Analysis is one of the objectives of Monte Carlo Simulation. A generic model for Monte Carlo Simulation is a black box with one or more inputs resulting in one or more outputs. Monte Carlo Simulation stops at the generation of the outputs but it would be of no use if we do not analyze the effect of changes in an input variable on an output variable. Sensitivity analysis asks the question: how does a change in input A affect output G or how does output G respond to changes in input A?

The different techniques for conducting sensitivity analysis that we can use in Monte Carlo Simulation are:

a) **The Results Worksheets**: this is what we have been developing so far. It usually consists of a frequency table with its cumulative counterpart, a chart of both tables and descriptive statistics. These are a kind of sensitivity analysis. They mostly show the effect of randomly selected input variables on one or more output variables. Although these sheets develop the output from the input, they do not provide the analyst with the "behavior" of sensitivity, the “how” of the changes.

b) **The Tornado Chart and Sensitivity**: we can prepare our own manual sensitivity analysis that results in a Tornado Chart showing the effect of each input variable on an output variable we are interested in. These are shown by plotting the minimum and maximum of each input variable. The largest variation would correspond to the input variable that has the largest effect on that output variable. There is no Tornado Chart in Excel. We have to set it up manually. There will be a workout on this below.

c) **WHAT IF Tables**: we can setup an additional simulation run that will contain two values for each input variable against one of the output variables. The values will be the minimum and the maximum. In this manner, we can setup the Tornado Chart.

We propose a technique that is based on the following approach. Suppose we have 4 input variables, A, B, C and D. Fix B, C and D. The fixed inputs are given point estimates which you can plug into the model. You then vary the remaining input variable A. As A varies from one limit to another, the output variable would vary accordingly. The result is made up of 2 values. Proceed by fixing A, C and D and varying B. Collect the 2 outputs resulting from this simulation. Do the same for the other variables. The result is made up of 2 outputs * 4 variables which can later be plotted as a Tornado Chart.

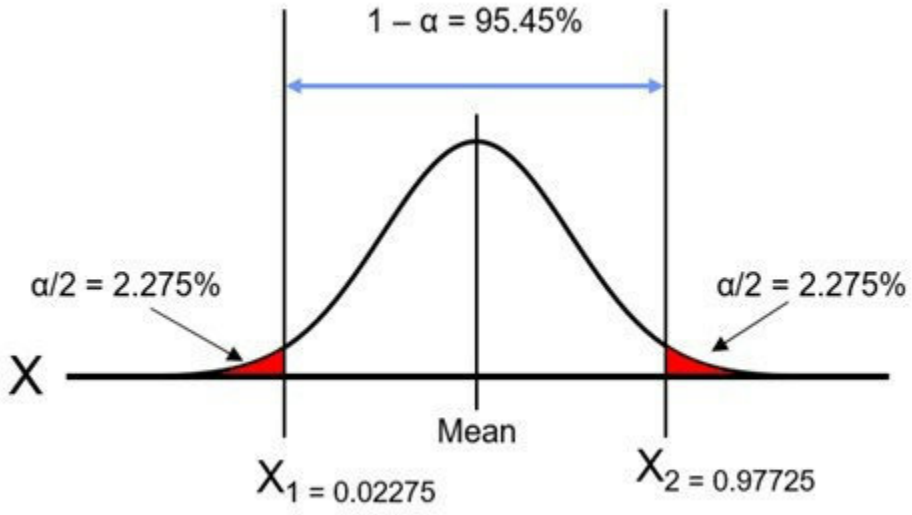
We have two issues to resolve:

First issue: what are the limits of the input variables? Since input variables will most likely be sampled from the distributions we have been discussing (and more to come), we can follow these practices:

a) For **uniform distributions**, we can use the actual lower and upper limits.

b) For **normal distributions**, the lower limit is 2 standard deviations below the mean. The upper limit is 2 standard deviations above the mean. The following formulas give this value for a standard normal distribution that has 0 as the average and 1 as the standard deviation: Upper limit = NORM.DIST(2,0,1,TRUE) = 0.97725

Lower limit = NORM.DIST(-2,0,1,TRUE) = 0.02275
Upper Limit - Lower Limit = 0.97725 - 002275 = 0.9545



In our models, we can calculate the range by using the above with the actual parameters (mean and standard deviation) defined in the Constants worksheet:

Lower limit = NORM.INV(0.02275, mean, standard deviation)
Upper limit = NORM.INV(0.97725, mean, standard deviation)

c) For **discrete random variable distributions**, since there is no closed form for this distribution, we have to calculate our lower and upper limits manually. We have a range with probabilities for each value of the input variable. Simply identify the value of the input variable that resides in the 2.275% and the 97.725% brackets which approximate the confidence interval above. For a rationale behind the choice of these values, review Section E in the Appendix in Chapter 15.0.

d) For **triangular distributions**, again, since we have an inverse function (whether through VBA or Excel formulas), we can get the values through the above probabilities. (Note that inverse functions are crucial in Monte Carlo Simulation. We will be using many of them in Part 2. In the early workouts of Part 2, we will be spending some time on the use of inverse probability functions.

e) For **BetaPERT distributions**, again, we use the inverse function with the above probabilities.

f) For **binomial distribution,** we use the same logic applies as above since we also have an inverse function.

g) For **geometric and negative binomial distributions**, these do not have direct inverse functions in Excel. Later on, we will be using the table of cumulative values to look up and get the value of the input variable. We can do the same for the probabilities 0.02275 and 0.97725.

Other distributions that we might use such as the Poisson, Exponential and Weibull can be treated similarly, using the above confidence interval.

Second issue: what do we use for the single point or fixed values of an input variable when another is being taken to its extremes? This is a simpler issue to resolve than the first. Consider the type of distribution that is being sampled. For example, say we are sampling a normal distribution. It would be logical to use the mean as the single point estimate. If we are using a BetaPERT distribution, the mode would be the logical value to use the mode as the single point estimate. For other symmetrical distributions, the median can be used. The next workout will apply this for the **normal**, **uniform** and the **discrete distributions**.

Workout 15: Budget Projection with Sensitivity Analysis

Purpose: to consider a model with 8 input variables and analyze the sensitivity of the output to changes in each of the input variables. For each input variable, two values will be used, a lower limit (usually at 0.02275 probability) and an upper limit (at 0.97725). We will prepare the model so we can use it as the basis of generic procedures for Sensitivity Analysis: Tornado Chart and different modes of Regression Analysis.

Problem statement: a projection of last year's income statement over 4 years requires us to use 8 growth factors. Without Monte Carlo Simulation, these would have to be single point estimates and will result in one figure: the net profit (or loss) at the end of year 4.

We will first simulate this model using the sampling techniques we have been using so far. In the "sensitivity analysis" phase of the workout, we will use Excel's WHAT IF tables to find the range of values in a single output variable in response to changes in the 8 input variables.

Step 1: to avoid the entry the tedious income statement formulation and its constants, 2 workbooks in the Workouts Folder have been prepared for your use:

a) **Budget Projection with Sensitivity Analysis - BLANK** is the startup workbook that contains the Model and its Constants.

b) **Budget Projection with Sensitivity Analysis** is a fully solved workbook starting with the blank workbook above.

Open the **Budget Projection with Sensitivity Analysis - BLANK** and save it as **Budget Projection with Sensitivity Analysis**.

Later on, we will save the solved Model as the **Budget Projection with Sensitivity Analysis with Tornado Charts** workbook which is also found in the Workouts Folder.

Step 2: let us briefly review the formulation in the Model sheet. It is a toned down income statement:

	A	B	C	D	E	F	G	H
1				Prev Year	Year 1	Year 2	Year 3	Year 4
2	Revenues							
3		Product Sales	0.0800	1,000	1,080	1,166	1,260	1,360
4		Contract Revenues	0.2000	164	196	236	283	339
5		**Total Revenues**		1,164	1,276	1,402	1,543	1,700
6								
7	Cost of Goods Sold							
8		Product Costs	0.3438	359	483	649	872	1,172
9		Subcontractor Costs	0.0402	63	65	68	70	73
10		**Total Cost of Goods Sold**		422	548	717	943	1,245
11								
12	Selling, General, and Administrative							
13		Payroll & Benefits	0.1200	110	123	138	155	173
14		Rent, Telephone, & Insurance	0.1268	119	134	151	170	192
15		Equipment & Software	0.1799	200	236	279	329	388
16		Supplies	0.1211	153	172	192	216	242
17		**Total SG&A**		582	665	760	869	994
18								
19	**Net Profit (Loss) Before Taxes**			160	63	-75	-269	-540
20	**Cumulative Profit (Loss) Before Taxes**			160	223	148	-121	-661
21	**Percent Profit (Loss) on Sales**			0.1596	0.0587	-0.0640	-0.2136	-0.3968

a) Col C contains the 8 growth rates (or multiplication projection factors). These are the input variables that we will randomize. Each will be sampled according to Step 4 below.
b) Col D contains the actual values for the previous year. These are constants and are hence shown in yellow.
c) Cols E to H project the previous year to the next 4 years using the growth rates in Col C.

Step 3: the constants are already setup in the Constants sheet as follows:

	A	B	C	D	E	F	G	H
1	**Component to Simulate**	**Distribution**						
2	**Revenues**							
3	Product Sales	Normal	Mean	0.0800	St Dev	0.0100		
4	Contract Revenues	Normal	Mean	0.1800	St Dev	0.0250		
5								
6	**Cost of Goods Sold**							
7	Product Cost	Uniform	Lower Limit	0.2000	Upper Limit	0.4000	Median	0.3000
8	Subcontractor Cost	Normal	Mean	0.0500	St Dev	0.0150		
9								
10	**Selling, General and Admin**							
11		Discrete						
12		Probabilty	Cumulative	Rate				
13	Payroll & Benefits	0.1500	0.1500	0.1000				
14		0.2800	0.4300	0.1100				
15		0.3500	0.7800	0.1200				
16		0.2200	1.0000	0.1300				
17								
18								
19	Rent, Telephone, & Insurance	Uniform	Lower Limit	0.1000	Upper Limit	0.1300	Median	0.1150
20	Equipment & Software	Normal	Mean	0.2000	St Dev	0.0300		
21	Supplies	Uniform	Lower Limit	0.1200	Upper Limit	0.1800	Median	0.1500

Step 4: in Col C in the Model sheet, enter the sampling formulas for the 8 growth rates. (NORM.INV will be presented in Part 2 of this eBook). The formulas use the constants in the Constants sheet:

C3 = NORM.INV(RAND(),Constants!$D$3,Constants!$F$3)
C4 = NORM.INV(RAND(),Constants!$D$4,Constants!$F$4)
C8 = RAND() * (Constants!$D$7-Constants!$F$7)+Constants!$F$7
C9 = NORM.INV(RAND(),Constants!$D$8,Constants!$F$8)
C13 = INDEX(Constants!$D$13:$D$16, IFNA(MATCH(RAND(), Constants!$C$13:$C$16, 1), 0) + 1)
C14 = RAND() * (Constants!D19-Constants!$F$19)+Constants!$F$19
C15 = NORM.INV(RAND(),Constants!$D$20,Constants!$F$20)
C16 = RAND() * (Constants!$D$21-Constants!$F$21)+Constants!$F$21

(These formulas have been entered into the **Budget Projection with Sensitivity Analysis - BLANK** workbook).

For the COUNTIFS() statement, refer to Chapter 7.0 for a description on how to use it. Alternatively, copy the formula from the General Model Template in the Templates Folder. Remember that the first range D13:D16 is where the values you need in the simulation are found, the growth rate of the Payroll & Benefits. The second Array is the range containing the cumulative % C13:C16. Both ranges are in the Constants sheet.

	A	B	C	D
10	**Selling, General and Admin**			
11		Discrete		
12		Probabilty	Cumulative	Rate
13	Payroll & Benefits	0.1500	0.1500	0.1000
14		0.2800	0.4300	0.1100
15		0.3500	0.7800	0.1200
16		0.2200	1.0000	0.1300

The bottom rows in the Model sheet show the net profit before taxes, the cumulative profit and the percent profit and loss on sales. We are interested in the projection of these 3 financial amounts for the current year + 4 which are in blue (objectives!):

19	Net Profit (Loss) Before Taxes	160	60	-82	-279	-550
20	Cumulative Profit (Loss) Before Taxes	160	219	137	-142	-692
21	Percent Profit (Loss) on Sales	0.1596	0.0550	-0.0694	-0.2176	-0.3946

Step 5: in the Runs sheet, enter the label "Run ID" in A1. In B1, C1 and D1, copy the values in the last year from the Model sheet:

B1 = Model!H19
C1 = Model!H20
D1 = Model!H21

These will be used as the top part of the WHAT IF analysis table.

Step 6: use Excel's autofill facility to generate the sequence 1, 2... 4000 in the range A2:A4001.

Step 7: to apply the sensitivity analysis WHAT IF table, follow the standard procedure:

a) Select the range B2:D4001

b) Select the menu item DATA *DATA TOOLS* WHAT IF ANALYSIS / DATA TABLE.

c) Click inside the "Column input cell" field and then click on any cell in the sheet and press OK.

Excel will use the WHAT IF table procedure to copy cells B2:D2 all the way down to Row 4001. This is a table or an Excel Array. The cells cannot be edited.

Step 8: we can now analyze the 3 cells (H19, H20 and H21) in the Results sheet. We shall only prepare the results for the first cell: Net Profit before Taxes. The rest follow similar procedures.

a) In A1 in the Results sheet, enter the label "Run ID". In the cells B1, C1 and D1 enter the labels "Net Profit", "Cum Profit" and "Gross %". (It is also possible to use the = operator to copy these from the range A19, A20 and A21 in the Model sheet.
b) Copy the range A2:D4001 **as values** from the Model sheet into the same range in the Results sheet.
c) Starting in F1, prepare the bin analysis for the 3 columns B, C and D:

	F	G	H	I
1		Net Profit	Cum	Gross %
2	Min	-966	-1,591	-0.7712
3	Max	12	648	0.0547
4	Range	979	2,239	0.8259
5	Bin Count	30	30	30
6	Bin Size	32.62714	74.6332	0.0275
7	Final Size	25	75	0.0250

d) Enter the following in the range K1:M2

	K	L	M
1	Net Profit before Taxes		
2	Bin	Frequency	Cum

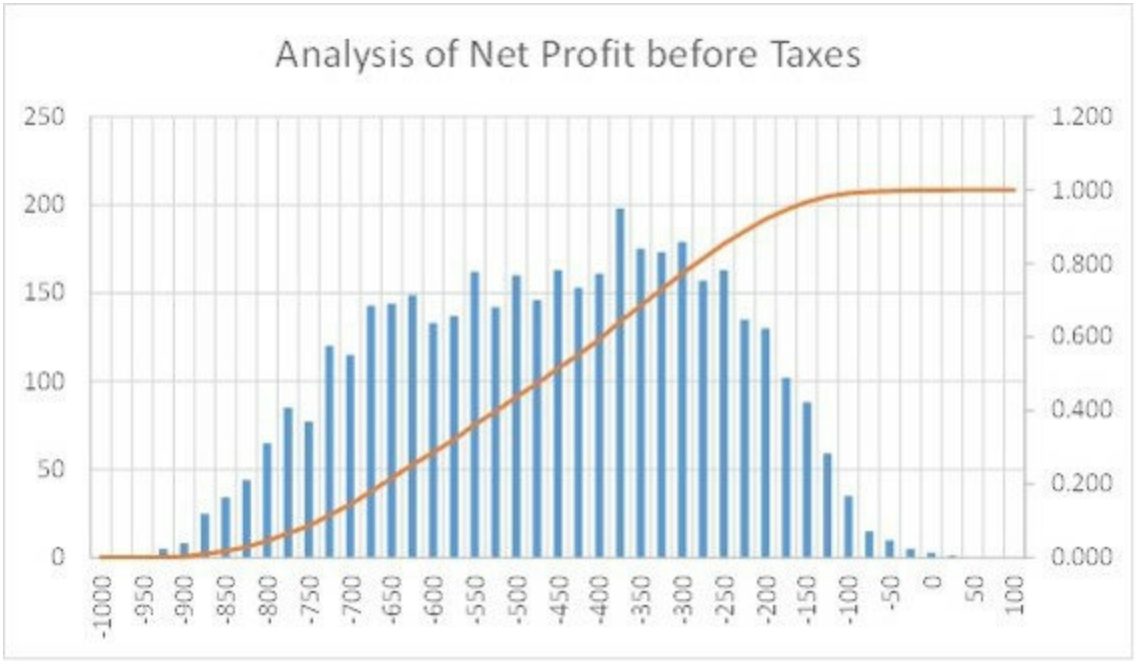
e) Having decided on 25 as the bin size, prepare the bins starting with the value -1000 (which is just below -986) and incrementing the bins by 25 until you reach the value 100.

f) Use the array function COUNTIFS() as presented in Chapter 7.0 to prepare the frequency count of the values in the range L2:L47. This is the formula to enter:

=COUNTIFS($B$2:$B$4001,">"&K2,$B$2:$B$4001,"<="&K3)

g) Prepare the cumulative probability in the range M3:M47. The formula in M3 =SUM($L$3:L3)/SUM($L$3:$L$47)

h) Insert a scatter diagram for the range K2:M47 to get the following chart:



Step 9: save the workbook **Budget Projection with Sensitivity Analysis with Tornado Charts** as we need to use it in the next few models.

Conclusion: almost 90% of the time, the net profit before taxes will be -225. (Of course, if you examine the table, there is no precise value for 90% but for more precise results, you can interpolate the values between 89% and 91.9%.

Even if we resort to the descriptive statistics, we do not have a direct way of identifying the input variable that has the highest effect on the output value (for net profit before taxes or for the others, when plotted). Let us follow the Sensitivity Analysis using this procedure.

Workout 16: Budget Project with Sensitivity Analysis and Tornado Chart

We will now introduce Sensitivity Analysis using the WHAT IF tables with the main objective of showing such analysis using a Tornado Chart. The next workout will use the WHAT IF tables to analyze the input and output on a chart in a completely different manner.

The general steps for such analysis are:

a) **Replace the input variables (green) with single point estimates**. These estimates are found in the Constants sheet and are generally means, modes or medians of the distributions in question. There is no reason why you cannot enter a specific value, manually. It depends on what you consider "typical" for that input variable.

b) **Prepare 8 WHAT IF tables, one for each input variable**. Each table will have two rows, one WHAT IF for an upper and one for a lower limit. The objective is to show the "swing" of the output based on the extreme inputs. (We will use formulas to arrive at these).

c) **Prepare a Tornado chart for the 8 input variables (Y-axis) vs. the output variable (X-axis)**.

Here goes... this workbook is elaborate (read: tedious) but not difficult.

Step 1: open the **Budget Projection with Sensitivity Analysis** workbook which was developed in the previous workout. It is also found in the Workouts Folder. Save it with a new name: **Budget Projection with Sensitivity Analysis with Tornado Charts**.

Ensure that the new workbook also has the ".xlsm" extension as we will be using the standard GENERATE RUNS VBA module with it.

Note: at some time when the model is loaded with WHAT IF tables, it might get overloaded with calculations. You may wish to turn off automatic calculations.

Step 2: right click on the Model sheet and copy it to a new sheet. Rename it "Sensitivity Model".

Step 3: replace the sampling formulas with formulas that provide the most likely single point estimate. At the beginning of this chapter, there are guidelines for specific distributions.

a) For the **normally distributed** variables, use the mean found in the Constants sheet:

C3: = Constants!D3
C4: = Constants!D4
C9: = Constants!D8
C15: = Constants!D20

b) For the **uniformly distributed** variables, use the median calculated as the average of

the two limits added to the lower limit.

C8: = Constants!H7 (which is calculated as = D7 + (F7 - D7)/2
C14: = Constants!H19 (similar calculation as for H7)
C16: = Constants!H21 (similar calculation as for H7)

c) For the **discrete probability distributions** (such as for the Payroll & Benefits), find the average of the probability blocks. The average rate in B18 is the sum of the products of each probability by the corresponding rate: Enter the following in the Constants sheet:

A17 = Average Payroll & Benefits
B17 = SUMPRODUCT(B13:B16,D13:D16)

Enter the following in the Model sheet:

C13 = Constants!B17

Step 4: prepare the 8 WHAT IF tables in the Sensitivity Model sheet. Each one is to be entered in a range of 4 rows x 2 cols. Enter the name of each input variable as labels in J1, J5, J9, J13, J17, J21, J25 and J29.

(To avoid wasting time entering these formulas as shown in Step 5 below, it is simpler to copy this range from the Solution Workout in the Workouts Folder).

	J	K	L	M
1	Product Sales			
2		-462	-522	-0.340
3	0.0600	-560	-751	-0.444
4	0.1000	-359	-283	-0.245
5	Contract Revenues			
6		-462	-522	-0.340
7	0.1300	-513	-632	-0.377
8	0.2300	-405	-401	-0.298
9	Product Costs			
10		-462	-522	-0.340
11	0.2000	-181	54	-0.133
12	0.4000	-816	-1,206	-0.600
13	Subcontractor Costs			
14		-462	-522	-0.340
15	0.0200	-454	-502	-0.334
16	0.0800	-471	-543	-0.346

17	Payroll & Benefits			
18		-462	-522	-0.340
19	0.1000	-452	-499	-0.332
20	0.1300	-471	-541	-0.346
21	Rent, Telephone, & Insurance			
22		-462	-522	-0.340
23	0.1000	-452	-500	-0.333
24	0.1300	-472	-544	-0.347
25	Equipment & Software			
26		-462	-522	-0.340
27	0.1400	-385	-355	-0.283
28	0.2600	-552	-707	-0.405
29	Supplies			
30		-462	-522	-0.340
31	0.1200	-462	-522	-0.340
32	0.1800	-462	-522	-0.340

The formulas for the input variables in the WHAT IF table (in Col J) are shown in Step 5 below.

Each WHAT IF table will have the same 3 output variables (in columns K, L and M). We will then have 8 tables, each of which examines the sensitivity of these outputs to a specific input variable. Later on, we will plot these results in a Tornado Chart.

Place these 3 output variables in each top line of the 8 WHAT IF tables. For example, K2 = $H$19 which comes from the **Net Profit (Loss) Before Taxes** in H19 of the current sheet: Sensitivity Model. The cells L2 and M2 similarly copy the **Cumulative Profit (Loss) before Taxes** from H20 and the **Percent Profit (Loss) on Sales** from H21.

Note: we will only produce a Tornado Chart for one output: **the net profit before tax** in H19. The rest would follow the same procedure.

For the cells shown in blue above, use the = operator to copy the value of the output variable you are analyzing. In our case, we place

= H19 into each of K2, K6, K10, K14, K18, K22, K26 and K30
= H20 into each of L2, L6, L10, L14, L18, L22, L26 and L30 and
= H21 into M2, M6, M10, M14, M18, M22, M26 and M30.

The easiest way to do that is to copy the first 3 cells as follows:

K2 = $H$19
L2 = $H$20
M2 = $H$21

Since the formulas are expressed in absolute references, simply copy the range K2:M2 and paste it into K6:M6, K10:M10, K14:M14, K18:M18, K22:M22, K26:M26 and K30:M30

Step 5: we now have 8 tables whose left side column has to have the input variable values. For example, for "Product Sales", we have to enter single point estimates in J4 and J5. The rationale for these formulas was explained at the beginning of this chapter. It relies on selecting the most suitable lower and upper limits. That depends on the distribution. As a summary: a) For **uniform distributions**, we can use the actual lower and upper limits.

b) For **normal distributions**, the limits are easily found using twice the standard deviation, once above and once below the mean:

Upper limit = NORM.INV(0.97725, mean, standard deviation)

Lower limit = NORM.DIST(0.02275, mean, standard deviation)

c) For **discrete random variable distributions**, this was manually calculated in B18 in the Constants sheet. (It is the SUMPRODUCT() of the probabilities and the corresponding rates.

The formulas for these cells are shown in the above diagram (in Col O in the solution workbook).

J3 = NORM.INV(0.02275,Constants!D3,Constants!F3)
J4 = NORM.INV(0.97725,Constants!D3,Constants!F3)
J7 = NORM.INV(0.02275,Constants!D4,Constants!F4)
J8 = NORM.INV(0.97725,Constants!D4,Constants!F4)
J11 = Constants!D7
J12 = Constants!F7
J15 = NORM.INV(0.02275,Constants!D8,Constants!F8)
J16 = NORM.INV(0.97725,Constants!D8,Constants!F8)
J19 = Constants!D13
J20 = Constants!D16
J23 = Constants!D19
J24 = Constants!F19
J27 = NORM.INV(0.02275,Constants!D20,Constants!F20)
J28 = NORM.INV(0.97725,Constants!D20,Constants!F20)
J31 = Constants!D21
J32 = Constants!F21

Step 6: with the 8 WHAT IF tables prepared, apply the WHAT IF procedure for each table:

a) Highlight the 3 Row x 4 Col range containing the output and the two input variables. For example, for "Product Sales", highlight the range J2:M4. For "Contract Revenues", highlight the range J6:M8, *etc.*

b) Select the menu item DATA *DATA TOOLS* WHAT IF ANALYSIS and choose DATA TABLE.

c) Click in the "Column input cell" field. For each table, click on the input variable being analyzed. For example, for the "Product Sales" table in the range J2:M4, click in the cell C3 where the single point estimate is. Excel will then plug in the two limit values and give you the corresponding output value.

What have we done so far? For each of the input variables (8 all in all), we froze the other 7 using the most likely value: median, mean or sum product. A WHAT IF analysis on that variable will show its effect on the output variable. For example:

Product Sales			
	-462	-522	-0.340
0.0600	-560	-751	-0.444
0.1000	-359	-283	-0.245

As the growth rate of the "Product Sales" varies from 6% to 10%, the Net Profit and Loss in the first column varies from -560 to -359. The Tornado Chart will plot the range that the Product Sales varies through for each of the 8 input variables, giving us the "sensitivity" of this output to each input variable. (See next step).

Step 7: to prepare the Tornado Chart, we need 3 columns: the minimum, the maximum of each output as well as the calculated range or the range (or difference) between the two columns. To make it easier for Excel's charting facilities, we place the range in Col D. Prepare a table starting in B24 as follows:

	B	C	D	E	F	G	H
23					Formulas		
24	**Input Variable**	**Lower**	**Range**	**Upper**	**Lower**	**Range**	**Upper**
25	Product Sales	-560	202	-359	=$K$3	=E25-C25	=$K$4
26	Contract Revenues	-513	108	-405	=$K$7	=E26-C26	=$K$8
27	Product Costs	-181	-636	-816	=K11	=E27-C27	=$K$12
28	Subcontractor Costs	-454	-17	-471	=K15	=E28-C28	=$K$16
29	Payroll & Benefits	-452	-18	-471	=K19	=E29-C29	=$K$20
30	Rent, Telephone, & Insurance	-452	-20	-472	=K23	=E30-C30	=$K$24
31	Equipment & Software	-385	-166	-552	=K27	=E31-C31	=$K$28
32	Supplies	-462	0	-462	=K31	=E32-C32	=$K$32

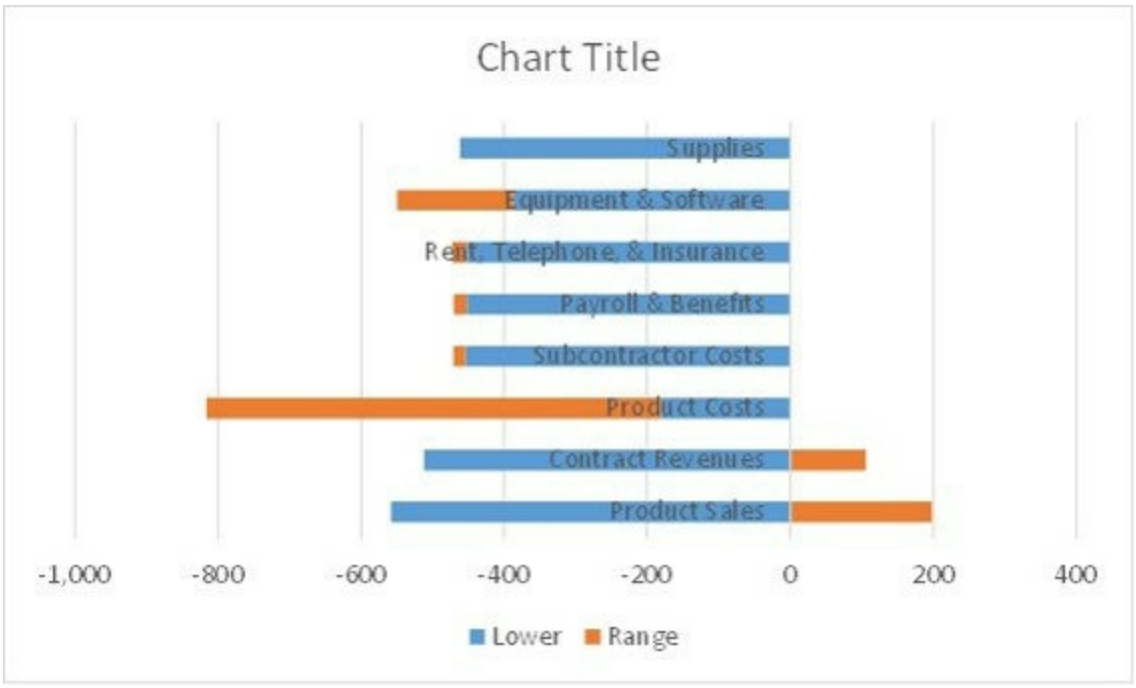
Each one of the cells under the Lower and the Upper headings has to be copied from the corresponding cells in the 8 tables. The formulas for the 3 columns are shown in the columns F, G and H in the Sensitivity Model sheet.

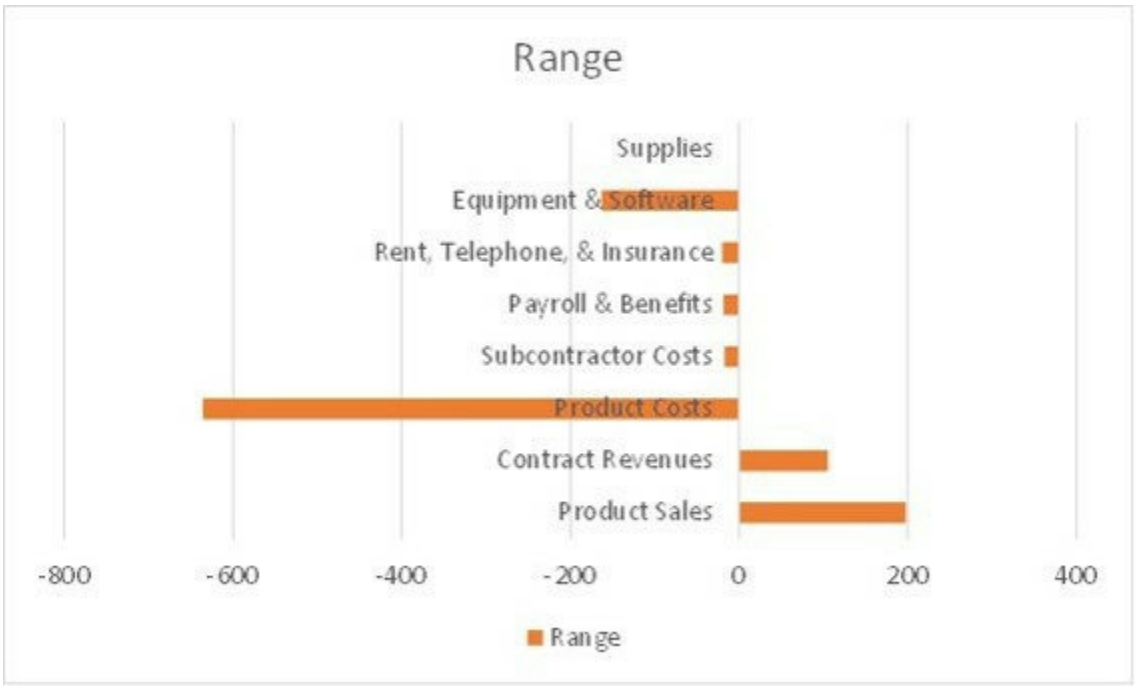
Step 8: since there is no Tornado chart in Excel, we have to prepare it manually:

a) Highlight the range B24:C32 (no need for the upper value)

b) Insert a horizontal stacked bar:

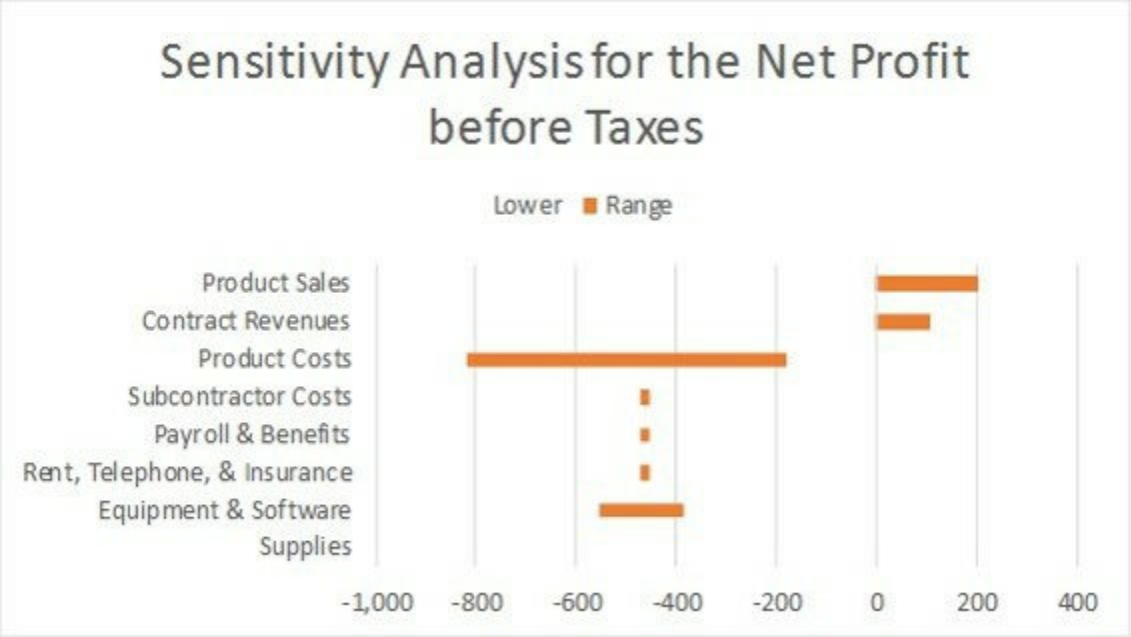
c) Select the plot for the lower range (shown below in blue) and delete the series:





We are now left with the range series or the middle block. This shows the lower and the upper values quite clearly. We need two more edits.

d) Select the labels in the vertical axis and right click to Format Axis. Under the Axis Options, check the "Categories in reverse order" and choose the "Label Position" to be the Low option. You will get the final Tornado Chart:



From the above (and from the table), you can see that the Product Cost is the input variable that most affects the net profit before taxes. (Notice how the legend says "Lower" without showing any color. This is due to our having deleted the first series: the lower values).

Workout 17: Seasonal Sales Model - Basic Model (UNIFORM)

Purpose: this workout has one major and one minor purpose. We will use it to model the sales of one product in a season using the Uniform Distribution. Secondly, in the next workout, this model will be extended to apply sensitivity analysis and regression to analyze the effect of changes in the ordered quantity on the net profit at the end of season.

Our purpose in this workout is simply to present a Seasonal Sales Model and to show that on its own, the results do not yield significant conclusions, therefore, requiring extension using Regression.

Problem statement: fashion wear has a sales demand pattern that depends on the season. If a seller has a high season for certain products in summer, the seller will order a quantity of brand name dresses in spring of that year. This quantity should satisfy the total demand in the months May to September. The seller’s experience tells him that the

demand in each month is uniformly distributed and allows him to specify the upper and lower range of demand, for each month independently.

a) If the actual demand in a specific month (to be sampled) is lower than the available quantity at the beginning of that month, the demand will be totally satisfied.

b) If the actual demand in a specific month is more than the available quantity, only what is available gets sold.

The available quantity at the beginning of May is the quantity on the single order. At the beginning of June, it is the brought forward from May which is the available at the beginning of May less what was sold in May. And so on.

At the end of the season, there might be a quantity of unsold items. A sale will be conducted where these items are offered at a discounted price. Not all remaining items might be sold. Another Uniform Distribution will determine the percentage quantity that will be sold at a discounted price.

The problem is to manipulate the ordered quantity to arrive at the most suitable order size. Normally, this model would have been suitable for implementation using optimization software. However, Monte Carlo Simulation provides a feature not available in optimization software which is the ability to analyze probabilistic behavior.

Step 1: create a new workbook and name it as you wish. In the Workouts Folder there is a fully solved model called **Seasonal Sales Model - Basic Model (UNIFORM)**.

Create the following sheets: Model, Constants and Results.

(For ease of work, you can copy the Results sheet from the **General Model Template**).

Step 2: setup the constants in the Constants sheet. There 5 constant amounts and 6 sets of parameters for sampling.

The 5 constants are setup in the range A1:B5:

	A	B
1	Ordered Quantity	1200
2	Unit Cost Price	$75
3	Unit Sales Price	$110
4	Unit Opportunity Cost	$20
5	Unit Sales Discount Price	$30

The first item in B2 is the amount we are interested in. This is the quantity the seller orders in spring and on which the performance of the summer depends. Order too much and there will be losses from unsold dresses. Order too little and losses will result from lost sales. What is the best level?

The Uniform Distribution parameters cover the demand range for each of the months of May to September.

The last row in the following table defines the range of the % that might be sold during the discount sale after the season is over. This value when multiplied by the remaining quantity at the end of the season (if any) will give the quantity that gets sold during the discount sale. Enter the values in the range A7:D13:

	A	B	C	D
7	Uniform	Lower Limit	Upper Limit	Range
8	Demand May	100	200	100
9	Demand June	150	250	100
10	Demand July	250	400	150
11	Demand August	250	400	150
12	Demand September	150	250	100
13	% Quantity Sold at Discount	0.00	0.70	0.70

Step 3: in col A of the Model sheet, enter the following labels:

	A
1	**Summer Months**
2	Demand Quantity
3	Available Quantity (Beginning of Month)
4	Actual Quantity Sold
5	Lost Sale Quantity
6	Cost of Ordered Quantity
7	Actual Sales Value
8	Lost Sales Value
9	Remainder
10	Quantity Sold at Discount
11	Sales Value of Remainder (Discounted)
12	**Total Profit/Loss**

Step 4: in row 1 set up 8 columns with labels as follows:

B1 to F1: May, Jun, Jul, Aug, Sep
G1: Total (Qty)
H1: Total (Value)

Here is a sample of the range A1:H12 as a guide for entry:

	A	B	C	D	E	F	G	H
1	**Summer Months**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Total (Qty)**	**Total (Value)**
2	Demand Quantity							
3	Available Quantity (Beginning of Month)							
4	Actual Quantity Sold							
5	Lost Sale Quantity							
6	Cost of Ordered Quantity							
7	Actual Sales Value							
8	Lost Sales Value							
9	Remainder							
10	Quantity Sold at Discount							
11	Sales Value of Remainder (Discounted)							
12	**Total Profit/Loss**							

Note: green is used to denote the input variables while blue denotes the output or the result.

Step 5: in each cell of the range B2:F2 in the Model sheet, generate the **Demand Quantity** based on the Uniform Distribution for each of the summer months: May to September.

B2 = RAND() * Constants!$D8+Constants!$B8
C2 = RAND() * Constants!$D9+Constants!$B9
D2 = RAND() * Constants!$D10+Constants!$B10
E2 = RAND() * Constants!$D11+Constants!$B11
F2 = RAND() * Constants!$D12+Constants!$B12

Step 6: in B3 calculate the **Available Quantity** for the month of May. It is equal to the opening balance or whatever the seller will purchase for the whole summer. This is our starting point. It is defined in B1 in the Constants Sheet.

B3 = Constants!$B$1

This amount is the opening balance for the season. It only depends on the purchased quantity. The cells in the range of C3 to F3 signifying the available quantity for months of June to September is dependent on the brought forward. We will enter B3 as shown above. As for C3 to F3, we will enter formulas after calculate all other quantities for May.

Step 7: in C3, the formula for the **Available Quantity** for June is different from than that in B3 for May. In May, we simply place the opening balance or the quantity purchased by the seller for the seller. In June, we need to know how much was left after any sale in May. The brought forward to June in C3 is simply the difference between the available quantity at the beginning of May and the quantity sold in May. If the demand is higher than the available, this would be 0 as we would have sold all the quantity in May.

In May, we sold 100 out of 1200. Therefore, 1100 is brought forward and made available in June. Assume that the demand was 1300, we can only sell 1200 in May. Therefore, the brought forward to June = 0.

C3 = B3 - B4

Copy this formula to the right up until F3.

Let us consider the following table and use it to support the formulas:

	A	B	C	D	E	F
1	**Summer Months**	**May**	**Jun**	**Jul**	**Aug**	**Sep**
2	Demand Quantity	100	225	346	399	209
3	Available Quantity (Beginning of Month)	1,200	1,100	875	529	130
4	Actual Quantity Sold	100	225	346	399	130

a) **May**: the demand was 100 and the available was 1200. Since the available quantity is larger than what was demanded, the actual quantity sold = 100.

b) **June**: the demand = 225. The quantity available at the beginning of June is the brought forward from May = 1200 - 100 = B3 - B4 = 1100. This is found in C3. The demand is < than the available quantity, we can sell as much as was demanded.

c) **July**: the demand = 346. The quantity available at the beginning of June is the brought forward from June = 1100 - 225 = 875. This is found in D3. The demand is < than the available quantity, we can sell as much as was demanded.

d) **August**: follows the same logic as July. The available quantity is 529 and is less than the demand of 399.

e) **September**: the demand 209 which is less than the available 130 or the brought forward from August. So we sell the available quantity on its own or 130.

Step 8: for May, the **Actual Quantity Sold** in B4 is the minimum of the demand (B2) and the available quantity (B3). If the demand is higher than the available quantity, we use the available quantity as the quantity sold. If the demand is lower than the available quantity, then we only sell what was demanded. As in the example below, the demand for May = 100 and the available quantity is 1200. We have enough to sell the 100 and we show that in B4

B4 = MIN(B2, B3)

Since this is the same for all other months, copy this formula to the right up until F4.

Step 9: for May, the **Lost Sale Quantity** is the opportunity loss. If the available quantity to sell (in B3) is less than the demand, we lose the difference between the two. So the formula for the Lost Sale Quantity is: B5 = B4 - B2

For May, the lost sale quantity is B5 = B4 - B2 = 100 - 100 = 0

This result would be zero when there is enough available quantity to meet the demand and B4 = B2. The demand would equal the actual sale. If the demand is higher than the available quantity, the difference B4 - B2 would be negative and would be what the seller would have lost.

Copy this formula to the right up until F5.

Step 10: in row 7 we calculate the **Actual Sales Value**. Each cell is the product of the actual quantity sold (row 4) by the unit sales price in B3 in the Constants sheet.

B7 = Constants!$B$3 * B4

Copy this formula to the right up until F7.

Step 11: in row 8 we calculate the **Lost Sales Value**. Each cell is the product of the lost sale quantity (row 5) by the opportunity unit cost in B4 in the Constants sheet.

B8 = Constants!$B$4 * B5

Copy this formula to the right up until F8.

So far, we have entered the calculations for the individual months. The next few steps will cover the calculation of the quantity and value totals for the whole operation (May to September).

Step 12: first consider col G where some of the quantities are totaled:

G2 = the total demanded quantity = SUM(B2:F2)
G3 is blank, there is no need to know the total available quantity
G4 = the total quantity sold = SUM(B4:F4)
G5 = the total quantity of lost sales = SUM(B5:F5)

	A	B	C	D	E	F	G	H
1	**Summer Months**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Total (Qty)**	**Total (Value)**
2	Demand Quantity	180	178	352	375	186	1,271	
3	Available Quantity (Beginning of Month)	1,200	1,020	843	491	116		
4	Actual Quantity Sold	180	178	352	375	116	1,200	
5	Lost Sale Quantity	0	0	0	0	-71	-71	
6	Cost of Ordered Quantity							($90,000)
7	Actual Sales Value	19,760	19,537	38,700	41,270	12,734		$132,000
8	Lost Sales Value	0	0	0	0	-1,411		($1,411)
9	Remainder						0	
10	Quantity Sold at Discount						0	
11	Sales Value of Remainder (Discounted)							$0
12	**Total Profit/Loss**							**40,589**

In G9 we calculate the remaining quantity or the quantity that could not be sold in summer. This is equal to the available quantity at the beginning (from B1 in the Constants sheet) less the total quantity sold (in G4).

G9 = Constants!$B$1 - G4

In the above example, it is zero since we have sold all the 1200 items.

In G10, we calculate the quantity sold during the discount sale. This is the product of two factors. First, we have the remaining quantity in G9 as calculated earlier. Second, we sample the Uniform Distribution (parameters in row 13 in the Constants sheet). The latter gives a percentage between 0 and 70% as defined in the Constants sheet in row 13. The product of these two factors is the actual quantity sold during the discount sale:
G10 = INT((RAND() * Constants!$D13 + Constants!$B13) * G9)

We apply the INT() function since the sampling formula will result in fractional values. We need whole integer quantities.

Step 13: column H contains a few monetary values.

In H6 calculate the **Cost of the Ordered Quantity**. This is simply the ordered quantity (in B1 in the Constants sheet) multiplied by the unit cost price (in B2 in the Constants sheet):

H6 = - Constants!B1 * Constants!B2

Note that since this is a cost, we have multiplied the value by a minus sign.

Step 14: H7 is the **Total Value of the Quantity Sold**. Since we have already calculated

the sales value for each of the 5 months in B7:F7, we simply sum that range:

H7 = SUM(B7:F7)

Step 15: H8 is the **Value of the Lost Sales**. Since we have already calculated the value of the lost sales for each of the 5 months in B8:F8, we simply sum that range:

H8 = SUM(B8:F8)

We do not add a minus sign here (although this is a loss) since the cells in the range B8:F8 are already negative or 0.

Step 16: H11 is the salvage value or the **Sales Value of the Remainder (Discounted)** stock. The quantity was calculated in G10. In H11, multiply G10 by the corresponding discount rate from the Constants sheet: H11 = G10 * Constants!B5

Step 17: H12 is the **Total Profit/Loss** which we get by adding the cost of the ordered quantity (H6) + actual sales value (H7) + value of lost sales (H8) + value of the quantities sold during the discount (H11): H12 = H6+H7+H8+H11

Note that the cost value (H6) and the value of lost sales (H8) are already calculated as negative amounts.

The Results?

If we analyze the variation of the profit and loss, we get a situation that is common in simulation: lots of work and no significant result. Let us try it before we move on to another mode of analysis:

Step 18: since our formulation is expressed within an elaborate range (A1:H12), it is not simple to generate rows for the simulation runs. We will use the WHAT IF method presented earlier:

a) Let K2 = H2 so we can analyze the total profit and loss in the WHAT IF table.

b) Use Excel's autofill facility to generate the sequence 1, 2, 3... 2000 in the range J2:J2001 (leaving J1 blank).

c) Highlight the range J1:K2001 then select the menu item DATA *DATA TOOLS* WHAT IF ANALYSIS / DATA TABLE

d) Click in the field for COLUMN INPUT CELL and then click on any cell in the sheet and press OK. Excel will use the WHAT IF table procedure to calculate the cells K2:K2001.

Use the results column in the following two steps.

Step 19: apply the technique of using COUNTIFS() to generate the frequencies and the cumulative % frequencies. These were presented earlier in Chapter 6.0.

Step 20: insert a scatter diagram (Pareto) as shown earlier in the same Chapter.

The graph shows that profit is spread over a wide range. That's all. We do not get any significant conclusion from the above model.

Given that we have 5 input variables and a few constants (ordered quantity, unit cost price, unit sales price, etc.), we could be looking at the wrong place for significant results.

In the following workout, we will use the method of sub-runs (presented in Chapter 13.0) to vary one of these constants: the ordered quantity. We will review its effects on the Profit and Loss by applying Simple Linear Regression to analyze the results. Applying the same technique to the other constants would follow the same procedure.

Workout 18: Seasonal Sales Model - Sensitivity Analysis with Regression

Regression can be used for sensitivity analysis though it is not obvious how. It can be used to analyze the sensitivity of an output variable (one at a time) to changes in one or more input variables. (The single variable case is called Simple Linear Regression while the multiple variable case is called Multiple Linear Regression). We will start with Simple Linear Regression. The next workout will show an example of Simple Non-linear Regression.

Refer to the Appendix in Chapter 16.0 for a discussion on regression and a pointer to a workbook with regression examples. Effectively, linear regression considers a table of paired data defined by an X and a Y each. Regression analyzes the behavior of Y as X varies. It is customary to call Y the dependent variable as it depends on the behavior of the independent variable, X. A regression is called "linear" if Y is not expressed using higher powers of X nor is Y expressed with X behaving in a non-linear manner (such as exponential, logarithmic, etc.). The following is an equation that defines a line regression: $Y = a* X + b$

Our aim in regression analysis is to find a and b so we can predict Y for different values

of X.

A) Convert the Basic Seasonal Sales Model into a Sub-Runs Model

Step 1: open the workbook **Seasonal Sales Model - Basic Model (UNIFORM)** developed in the previous workout and also found in the Workouts Folder. Since we need to introduce the VBA Module, save the file **Seasonal Sales Model with Regression** as macro enabled (with the ".xlsm" extension).

Step 2: delete the Runs sheet that was developed in the earlier workbook.

Clear the contents and formatting of the Results sheet as we need to introduce new material into it. The Constants sheet remains the same.

Step 3: the following steps were detailed in chapter 13.0. Let us just summarize them here:

a) Import the VBA module from **GenerateRuns.txt** in the Supporting Documents folder

b) Enable iterative calculations in FILE *OPTIONS* FORMULAS

c) Retain the range L1 and L2 in the VBA module as they are since we will place the Max and Current ID's in the range K1:L2.

	K	L
1	Number of Runs	50
2	Current Run ID	50

d) Create a button called GENERATE RUNS.

Step 4: since the main runs block shall be place in B18:G67, prepare the headers of the runs block in the Model sheet in A17:G17. Enter the following labels:

	A	B	C	D	E	F	G
17	Run ID	Ordered Qty	Profit and Loss	Demand Qty	Sales Value	Lost Sales Value	Discount Value

Step 5: use Excel's autofill facility to generate the sequence 1, 2... 50 in the range A18:A67. These IDs will be used to check with the Current Run ID or the loop counter in L2. If the contents of L2 are equal to a specific ID, that line will get filled. Those earlier lines with a lower ID will be kept as they were entered and those later lines with a higher ID will be kept blank. This will be clarified below.

Step 6: **the ordered quantity**: use the autofill in Excel to generate the sequence 100, 200... 5000 in the range B18:B67. These are the ordered quantities we shall use instead of the earlier quantity (1200) specified in B2 the Constants sheet and entered in B3 in the Model sheet. We will therefore need to modify B3 as follows: B3 = INDEX(B18:B67, L2, 1, 1)

This function uses the current loop ID in L2 to pick up the Ordered Quantity entered in

the column B18:B67 above. For each iteration or loop, a new quantity will be used to generate a new profit or loss value.

Step 7: since all rows from 18 to 67 are the same, we shall describe the entries in the range C18:G18 and copy them down to C67:G67.

C18 is the **profit and loss** picked up from H12. For each run, we need to pick up a new value. We need to match the current loop ID loop ID in L2 with the Run ID in A18:A67.

C18 = IF(A18=$L$2, $H$12, IF(A18>$L$2,"",C18))

This formula is typical of all looped formulas we've had before. It reads as follows:

a) IF the Run ID in our Row (A18) = the Current Run ID pushed by the VBA module into L2, we are on the Row that will need new data or copies of the output from the formulation. Simply let the IF statement insert H12 into C18.

b) IF the Run ID (A18) > the Current Run ID in L2, the formula is in a Row we have not reached yet. The IF statement will place a null value or a "" in that Row in C18. There will be no values for rows greater than the Current Run ID.

c) IF the Run ID (A18) < the Current Run ID in L2, the formula is in a Row we have already processed. Keep the value of C18 in C18. Placing the same value in a cell would not have been allowed without disabling iterative calculations.

Step 8: the remaining cells in the range D18:G18 also pick up values from the upper formulation and check A18 in the same manner:

D18 = IF(A18=$L$2,$G$2, IF(A18>$L$2,"",D18))
E18 = IF(A18=$L$2,$H$7, IF(A18>$L$2,"",E18))
F18 = IF(A18=$L$2,$H$8, IF(A18>$L$2,"",F18))
G18 = IF(A18=$L$2,$H$11, IF(A18>$L$2,"",G18))

These formulas pick up the demand quantity, the sales value, the lost sales value and the discount value from G2, H7, H8 and H11 respectively. We now have a working Row.

Copy C18:G18 down to C67:G67.

Step 9: enter 50 in L1 and click on the GENERATE RUNS button. You will observe the runs block B18:G67 being built up. L2 will cycle from 1 to 50.

This means we have 5 outputs simulated over 50 runs. (We could have used 1000 runs, for better analysis.). The table B17:G67 is properly set for regression analysis. Our independent variables (4 X's) are in B17:B67 while our dependent variable Y is in C17:C67 (including the labels).

B) Use Regression for Sensitivity Analysis

Regression is a mathematical technique used to find the best line of fit between one independent variables (X) and the resulting dependent variable (Y). If we only have one X, we call this: Simple Regression. If we have more than one X, we call it Multiple

Regression.

We usually have no control over the independent variable(s). Its behavior results from causes outside our control such as market conditions, client demands, error count and agriculture produce. The relationship between the independent and the dependent variables is governed by an equation which is expressed as Y(X) = f(X) where f(X) can be linear, exponential, polynomial, logarithmic, *etc.*

Simple linear regression is governed by one equation: Y = a * X + b. If we have a table with several rows of paired entries for X and Y, we can find the value of the two parameters: a is the slope of the curve and b is the intercept (the value of Y when X = 0).

We shall first address **Simple Linear Regression**. We shall then work on some non-linear regression examples. Multiple regression is out of the scope of this eBook due to the complexity of its tests.

Purpose: this extension to the basic Seasonal Sales Model identifies the nature and effect of changes in the **ordered quantity** (which is our independent variable) on the output or our profit or loss (which is our dependent variable). In the basic model, we had assumed the ordered quantity to be a constant.

We will analyze the column C17:C67 in the Model sheet in several ways. The number of ways depends on the "visual" shape of the data. If you feel that the scatter plot resulting from plotting the range B17:C67 is decidedly linear, then only one type of analysis would be required. Otherwise, you would need to try different patterns. We will cover the non-linear behavior after modeling linear regression.

Step 1: copy the range B17:G67 and paste it as values in the Results sheet starting in A1. Here are the first 10 rows from the Results sheet:

	A	B	C	D	E	F
1	Ordered Qty	Profit and Loss	Demand Qty	Sales Value	Lost Sales Value	Discount Value
2	100	-98,573	1,079	11,000	-19,573	0
3	200	-88,913	1,246	22,000	-20,913	0
4	300	-72,680	1,084	33,000	-15,680	0
5	400	-63,367	1,268	44,000	-17,367	0
6	500	-49,532	1,227	55,000	-14,532	0
7	600	-36,661	1,233	66,000	-12,661	0
8	700	-22,073	1,154	77,000	-9,073	0
9	800	-6,079	1,004	88,000	-4,079	0
10	900	112	1,344	99,000	-8,888	0

Step 2: find the correlation between the various columns. This is a statistical test that measures the strength of the relationship between one column of data and another. If there is a strong positive relationship (the more the advertising the higher the sales), the

correlation will be somewhere between 0.7 and 1.0 (and not > 1.0). If there is a strong inversely proportional relationship (the more training, the less the number of errors), the correlation will be somewhere between -0.7 and -1.0 (not < -1.0).

If you only have two columns as in simple regression, you can use the CORREL() function in Excel. CORREL() returns the correlation if you specify the range containing the X's and the range containing the Y's (they must have an equal number of entries).

If you have more than 2, use the Analysis Toolpack as follows:

a) Select the menu item DATA *ANALYSIS* DATA ANALYSIS and click on the item "Correlation". This is the dialog box to fill:

b) Enter the input range which in our case is A1:F51 including the labels.

c) Check the "Labels in first Row" box.

d) Click on the field for the output range and then click in the cell H2 in the Results sheet.

The Analysis Toolpack will give you the following (without proper reformatting):

	Ordered Qty	Demand Qty	Sales Value	Lost Sales Value	Discount Value	Profit and Loss
Ordered Qty	1.00					
Demand Qty	0.19	1.00				
Sales Value	0.68	0.23	1.00			
Lost Sales Value	0.64	-0.04	0.96	1.00		
Discount Value	0.74	0.05	0.38	0.37	1.00	
Profit and Loss	**0.82**	0.17	**0.93**	**0.91**	**0.69**	1.00

Note: the principle we should follow is this: we can only use regression between two variables when there is a high correlation between them. We will consider how the Profit and Loss in the last line correlates with the other variables: a) The above table

only shows one set of values. The cells shown in gray are not used for two reasons: correlating one variable with itself is meaningless. Such cells show a 1 and not useful. Correlating A with B gives the same result as correlating B with A. We therefore exclude the reverse relationships under the gray cells.

b) The items shown in bold in the last Row show that: the Profit and Loss data is highly correlated with Ordered Quantity, Sales Value, Lost Sales Value and Discount Value. It is not highly correlated with the Demand Quantity.

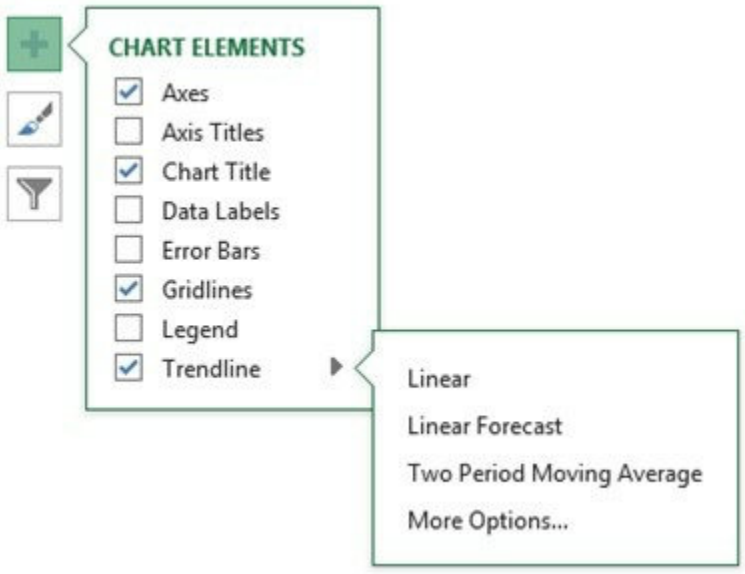
Step 3: let us plot the two columns: Ordered Quantity (X) and Profit and Loss (Y). This will show us the sensitivity of our output variable (profit and loss) to changes in the ordered quantity. The correlation between them was 0.82. This is high enough to question to allow regression analysis.

Highlight the range: A1:B51 and insert a scatter diagram. You will get this chart:

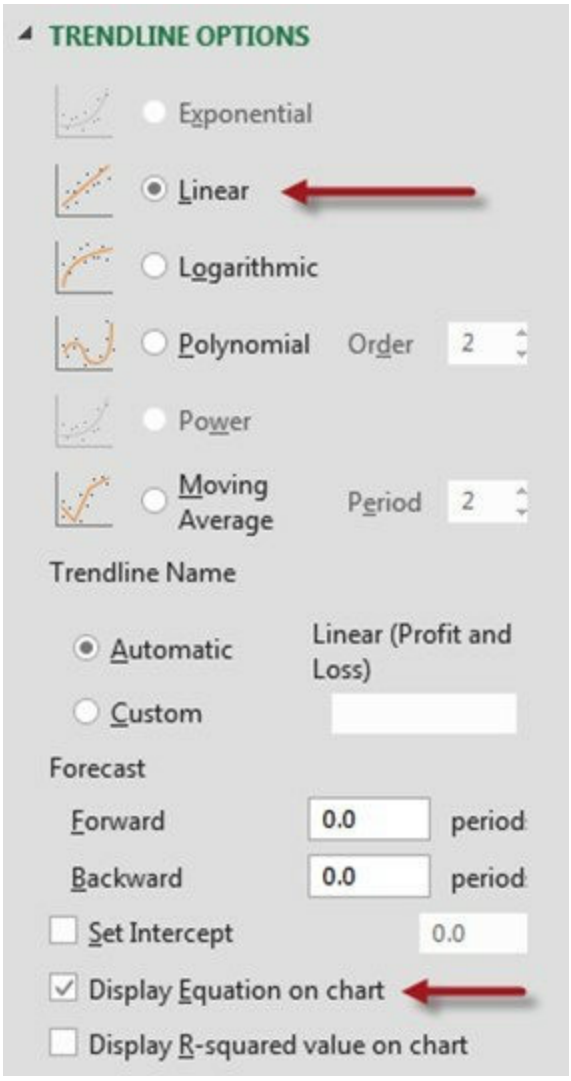
There are two patterns here. The first is a line that progresses from quantity = 0 to around 1400. The second is a line that is less inclined and progresses from 1400 to 5000. Technically, we should apply regression to these two ranges separately. Let us see if there are other ways of getting a better result. Let us start with the dumb assumption: the curve is governed by a linear trend. Let us try to find the parameters a and b for the line Y = a * X + b.

Step 4: there are various trend options available in Excel when drawing a chart. Here is the procedure:

a) Select the chart just created and click on the "+" icon on its right. Select the Trendline option and click on the arrow on its right:



b) When you click on "More Options" you will get a panel on the right side:



c) Select the Linear option and at the bottom, check the box that displays the equation on the chart:

The equation is shown as Y = 26.915 X - 27187. This can be read as follows:

a) If X = 0, i.e., we do not order any quantity, then we have a loss of -27,187. This is the intercept of the best line of fit with the Y axis.

b) If X = 1 or there is a change of 1 in the ordered quantity (X), this will result in a change in Y of 26.915. Each increase of 1 in the quantity will increase our profit by 26.915.

c) If we insert different values of X to predict Y: if X = 6000 (which is not in our table), the equation gives Y = 26.915 * 6000 - 27187 = 134,303.

This is the meaning of "sensitivity" in this method. We can now measure the effect of changes in X = Sales Value, our input variable on Y our net profit and loss.

Note: the main curve is not really a straight line. We must try different "trend" fits to see the best amongst them. This is a matter of trial and error that we can try after the next step. The next step only shows how Excel can visually represent a specific type of trend.

Step 5: we can get to the same results without using the chart. There are several Excel functions that can be used instead. The most direct are: SLOPE() and INTERCEPT(). The first results in the parameter a and the second in the parameter b. The others are FORECAST(), TREND() and LINEST().

a) In H10 and H11 in the Results sheet, enter the labels "Slope" and "Intercept".

b) Let I3 = SLOPE(B2:B51, A2:A51) where B2:B51 points to the column containing the Y values (Profit and Loss) and A2:A51 points to the column containing the X values (Ordered Quantity). We should not include the labels.

c) Let I4 = INTERCEPT(B2:B51,A2:A51). It must cover the same range.

The result is the Slope a = 26.9155 and the Intercept b = -27,187 exactly what the chart showed.

Step 6: try different trend lines to see which one gives the best fit. Right click the

Results sheet and copy it to a new sheet, renaming it "Results Non-Linear".

a) Delete the earlier chart and the other calculations.

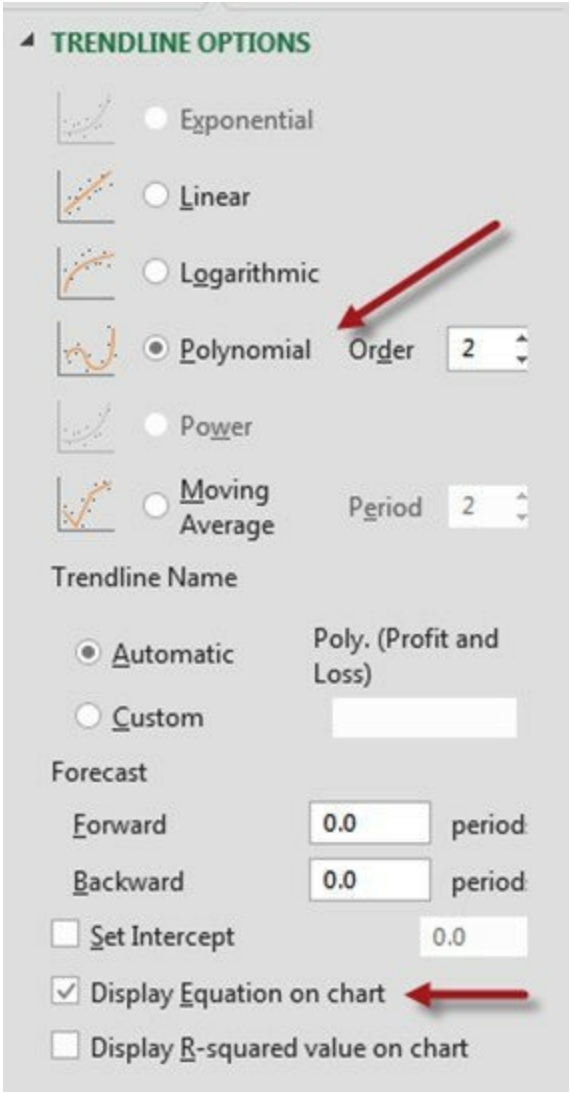
b) Delete the two columns C, D, E and F. You should be left with the 2 data columns that we investigated earlier: the Ordered Quantity and the Total Profit and Loss

Step 7: highlight the range A1:B51 and plot the scatter diagram. You will get this curve which is the same as the earlier curve. Let us see what it is hiding:

Step 8: the curve seems to be a **polynomial** curve, i.e., one that contains high powers of X: squares, cubes, *etc.*

a) Click on the graph and you will see a plus icon to the right. Click on that icon.

b) From the drop down list, check the box that says "trend line" and then select More Options. You will get a panel on the right hand side of your chart:



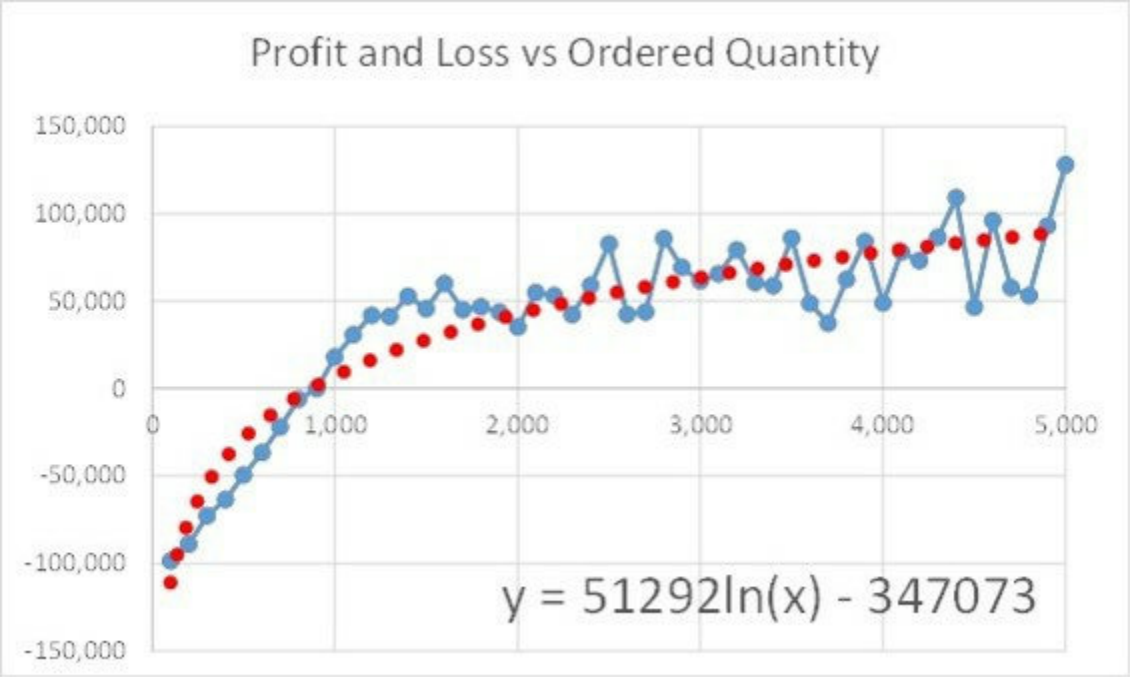
Select the Polynomial with order = 2 (or X squared) and then check the option for displaying the equation on the chart. You will see the equation:

Y = - 0.0107 * X^2 + 81.414 * X - 74419 which shows the effect of the ordered quantity on the net profit and loss. The chart now shows a dotted line showing the polynomial curve which tries its best to fit our data:

c) Try to change the polynomial degree to 3 and you will get a better fit and a different equation:

This graph looks like it has a better fit but looking at the coefficient of X^3, we see that it is a very small number, so there is no contribution from the third power of X. But then simulation is a matter of experimentation and we are now a bit more sure that the second degree polynomial has a better fit.

Step 9: try to present the equation assuming the best line is logarithmic. You will get this plot:



Although the line after X = 1000 seems a better fit, the difference between the two at the lower values could be a bit too high.

C) How do we Test which Line Gives the Best Fit?

There are two statistical tests that can be used for just such a question: MAD (the Mean Absolute Error) and RMSE (the Root Mean Squared Error). However, RMSE is deemed the tighter test. We shall apply it as follows:

Step 1: right click on the tab of the Results Non-Linear sheet and copy it to a new sheet renaming it "RMSE".

Step 2: delete the earlier charts and the other calculations. You should be left with the 2 data columns alone: ordered quantity and the total profit and loss.

Step 3: here are the three equations we arrived at earlier:

Simple linear regression: Y = 26.915 * X - 27187
Second degree polynomial: Y = -0.0107 *X^2 + 82.414* X - 74419
Third degree polynomial: Y = 7E-06 *X^3 - 0.0608* X^2 + 184.59 * X - 120426
Logarithmic: Y = 51292 * ln(X) - 346073

(Note that in the third degree polynomial equation, a is very small. Excel uses the scientific notation: 7E-06 which means 7 / 1,000,000.

The method requires us to use the actual values of the Ordered Quantity (X) in Col A and predict Y (Total Profit and Loss) for each equation. What would have been our Y if we had used these equations? We can then compare the actual Y with the predicted Y and apply some tests that can provide us with a comparison of the predictability of each equation.

C1 = Predicted Y for Simple Linear
D1 = Squared Errors

E1 = Predicted X for Polynom 2
F1 = Squared Errors
G1 = Predicted X for Polynom 3
H1 = Squared Errors
I1 = Predicted X for Logarithmic
J1 = Squared Errors

Since the range is wide, the following are two separate captures:

	A	B	C	D	E	F
1	Ordered Qty	Profit and Loss	Predicted Y for Simple Linear	Squared Errors	Predicted X for Polynom 2	Squared Errors
2	100	-98,573	-24,496	5,487,536,249	-66,285	1,042,567,033
3	200	-88,913	-21,804	4,503,656,943	-58,364	933,246,963
4	300	-72,680	-19,113	2,869,488,145	-50,658	484,981,851
5	400	-63,367	-16,421	2,203,891,121	-43,165	408,089,240
6	500	-49,532	-13,730	1,281,824,001	-35,887	186,187,928

G	H	I	J	K	L
Predicted X for Polynom 3	Squared Errors	Predicted X for Logarithmic	Squared Errors	Broken Linear Prediction	Squared Errors
-102,568	15,956,776	-109,865	127,491,292	-100,441	10,108,492,681
-85,884	9,176,604	-74,312	213,206,296	-87,439	7,680,594,321
-70,332	5,513,590	-53,515	367,316,934	-74,437	5,585,619,169
-55,870	56,199,293	-38,759	605,544,722	-61,435	3,823,567,225
-42,456	50,070,763	-27,313	493,672,778	-48,433	2,394,438,489

Step 4: enter the prediction formulas above:

C2 = 26.915 * A2 - 27187
E2 = -0.0107 * A2^2 + 82.414 * A2 - 74419
G2 = 7E-06 * A2^3 - 0.0608 *A2^2 + 184.59* A2 - 120426
I2 = 51292 * ln(A2) - 346073

Step 5: and now we calculate the deviations:

D2 = (B2-C2)^2

This is the difference between the prediction in C2 and the actual in B2. It is squared (for mathematical reasons we will not get involved in).

F2 = (B2-E2)^2
H2 = (B2-G2)^2
J2 = (B2-I2)^2

Copy the whole range C2:J2 down to Row C51:J51.

Step 7: These will be averaged later and their square root taken to give the Root Mean Square Errors (RMSE).

Step 8: Enter the labels "Mean Squared Deviations" A53 and "Root Mean Squared

Errors (RMSE)" in A54. (Here, error is used to mean deviation).

Step 9: calculate the averages of the ranges D2:D51, F2:F51, H2:H51 and J2:J51:

D53 = AVERAGE(D2:D51)
F53 = AVERAGE(F2:F51)
H53 = AVERAGE(H2:H51)
J53 = AVERAGE(J2:J51)

In the cells just below these, calculate the square root of each average:

D53 = SQRT(D53)
F53 = SQRT(F53)
H53 = SQRT(H53)
J53 = SQRT(J53)

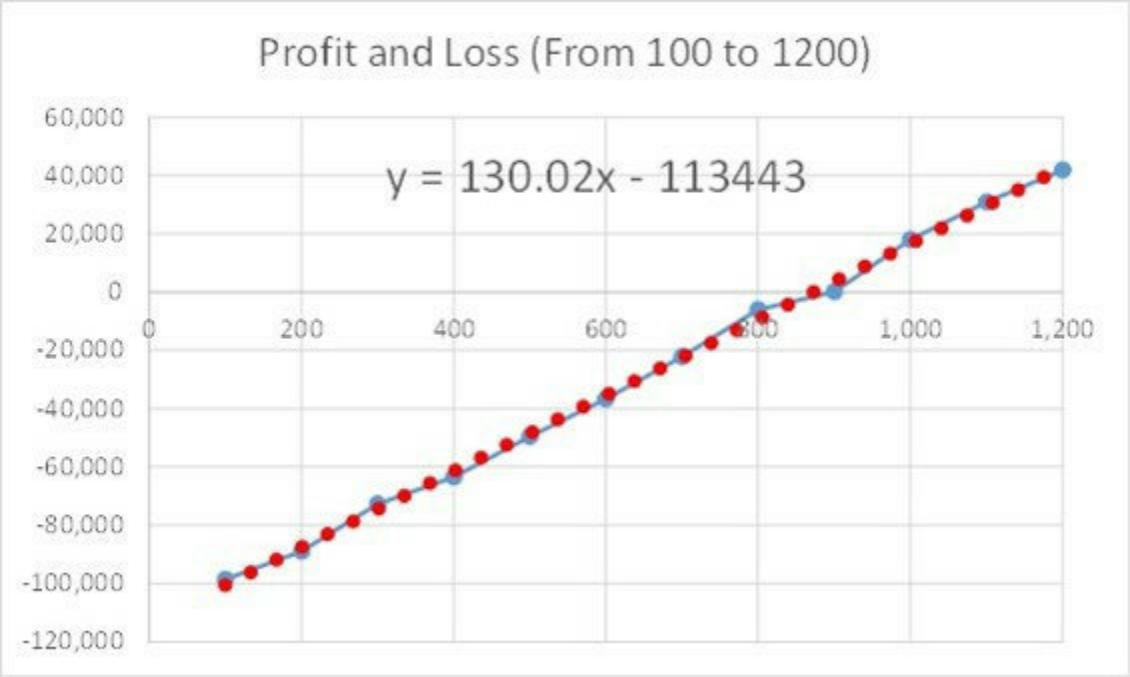
Conclusion: the lower RMSE is for the logarithmic line of fit which looked like the worst but came out on top, statistically:

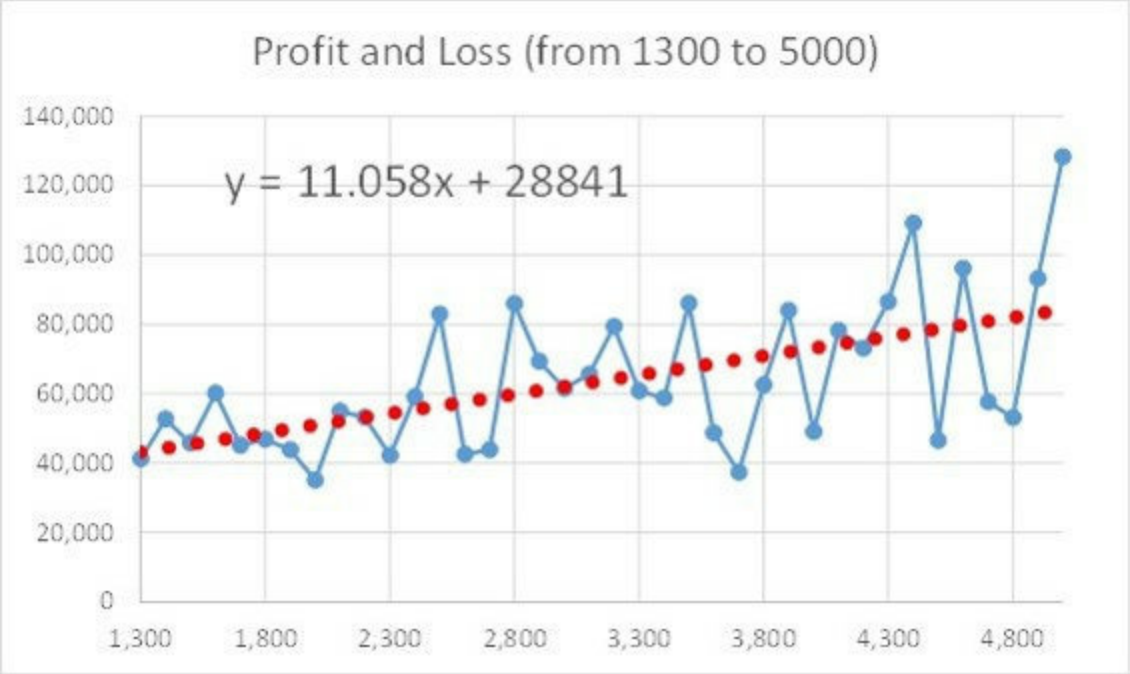
RMSE for Simple Linear Regression = 29,635
RMSE for Polynomial 2 = 22,141
RMSE for Polynomial 3 = 26,826
RMSE for Logarithmic = 18,695

RMSE on its own does not have an explanatory significance. It is the relative value of the 4 results that counts. We can see that the Logarithmic trend gave the least Root Mean Square Error closely followed by the 2nd degree polynomial trend. The logarithmic can then explain the sensitivity of the total profit and loss to changes in the ordered quantity.

An Extension to the Model:

Our total profit and loss can be viewed as two linear trends, we can calculate the simple linear regression for the range of quantity 100 to 1200 and do the same for the remaining range of 1300 to 5000. These result in these two charts:





In the last two columns (K and L), calculate the RMSE keeping in mind that we need to use an IF statement with an internal constant: 1200, the break point above which we use the second of these equations:

Y = 130.02 * X - 113443 (from 100 to 1200)
Y = 11.058 * X + 28841 (from 1300 to 5000)

The RMSE is calculated in the same manner and comes out to be = 59,223.

Ironically, what looked as the best fit, visually, turned out to be the worst, computationally. With the wisdom of hindsight, but backed by solid numbers, we might explain this as follows. Looking at the second set of figure (above 1200), we can see a trend but there is a large variation around the best line of fit. The curve finds the best line but the RMSE tells us: do not believe it.

15.0 Appendix A: Descriptive Statistics and Related Measures

This chapter covers essential measurements and statistical indicators that we can use when analyzing the results of a simulation model. The chart that displays the frequency counts and the related cumulative % curve have been covered earlier in chapter 9.0 and will not be discussed here. Here, we will be concerned with the Descriptive Statistics generated by the Analysis Toolpack and related measures not included in the Toolpack.

Workout 19: Generate Descriptive Statistics with the Analysis Toolpack

Purpose: to clarify the descriptive statistics provided by the Analysis Toolpack and to suggest a few more measurements. The results used in the following workout were taken from one typical run in the workout: **Duration of Batch Production - NORMAL+BETA** presented in Part 2 of this eBook.

Assumptions: not all probability distributions are normally distributed. The Analysis Toolpack assumes they are when the results are calculated. It goes ahead and calculates the Mean, Standard Deviation, *etc.* without limiting the presentation to a specific type of distribution. The results can be meaningful in most situations. However, if the distributions are obviously not Normal or they do not "tend" to be Normal, we should only use those statistics that are not related to the Normal as we will see below.

The results from the above simulation were placed in 1000 cells and were copied into a workbook in the Workouts Folder called: **Generate the Descriptive Statistics - DATA**.

Before starting, ensure that the Analysis Toolpack has been setup and enabled. This is presented in chapter 5.0.

A) Sheets: Descriptive Statistics, Median and Mode

Step 1: create a new workbook and save it under any name you wish. In the Workouts Folder there is a fully solved model called **Generate the Descriptive Statistics with the Analysis Toolpack**.

Create the following sheets:

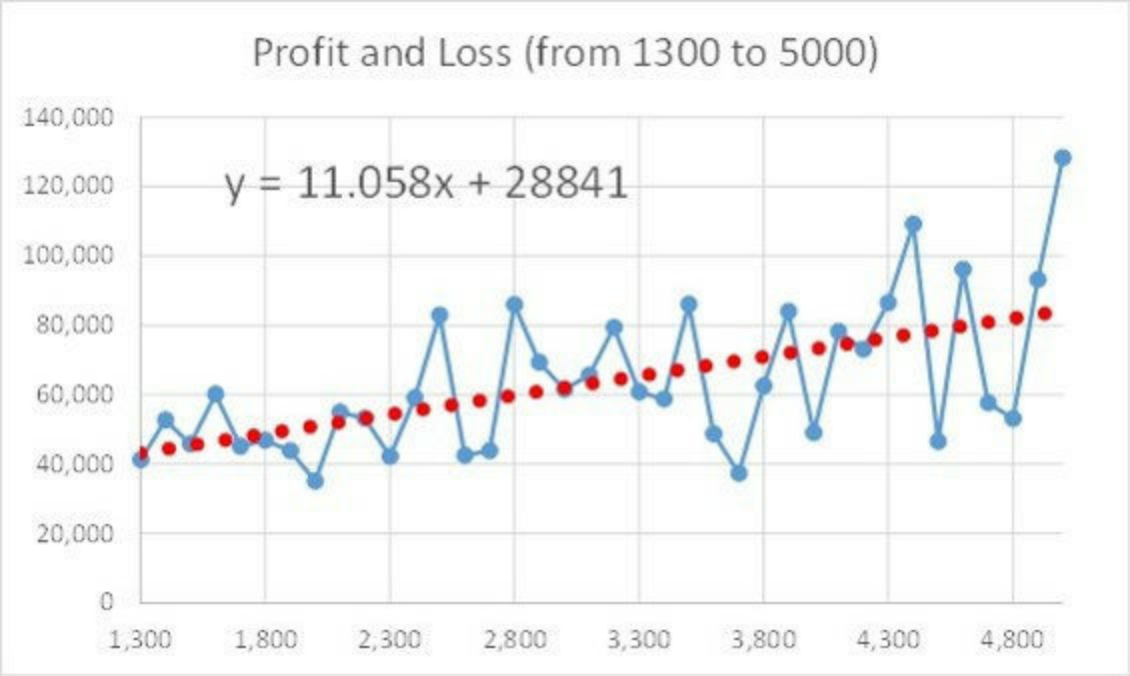
Descriptive Statistics
Median
Mode

St Dev
Normal Distribution
Confidence Interval
Confidence Calculations

Open the workbook associated with this workout: **Descriptive Statistics - DATA** and copy the range A1:A1001 from the Results worksheet. Paste it into A1 in the **Results** sheet of the new workbook.

Note: the values in the workout are fractional. In the new workbook and for ease of discussion, they were truncated.

Step 2: Select the menu item DATA *DATA ANALYSIS* DESCRIPTIVE STATISTICS



Fill up the dialog box as follows:

Input Range: the data column or range you wish to analyze.
Labels in first Row: include this if the above range includes a header Row
Output: can be in the same sheet or in a new sheet or workbook
Summary statistics: generates the table that contains most measurements
Confidence level: we will describe this below
Kth largest or smallest: this is useful for results that are not continuous but discrete such as stock levels, number of clients, number of invoices, *etc*. No need to fill it for our work.

Step 3: the resulting table from the Analysis Toolpack is poorly formatted and shows all numbers with large precision. Revise this to make it more legible by restricting the number of digits to the right of the decimal point, *etc*. The best way is to review the **General Model Template** in the Templates Folder from which you can "borrow" the formatting using the formatting brush.

Here are the Descriptive Statistics as generated by the Analysis Toolpack for the Batch Production workout. Only 17 Inspection Times were shown in Col A out of the copied 1000 results.

	A	B	C	D
1	Inspection Time		Inspection Time	
2	473			
3	508		Mean	459.1190
4	503		Standard Error	2
5	418		Median	455
6	478		Mode	457.0000
7	429		Standard Deviation	49.8381
8	524		Sample Variance	2,483.83
9	490		Kurtosis	0.32
10	452		Skewness	0
11	468		Range	354
12	431		Minimum	316
13	392		Maximum	670
14	462		Sum	459119
15	400		Count	1,000.0000
16	407		Confidence Level(95.0%)	3.0926854
17	500			

1) **The Mean** is the average of all observations (taken over 1000 runs in this case). Excel has a function for the mean: = AVERAGE(Range).

2) **The Standard Error** will be discussed in a later section in this chapter as this is a little more elaborate than the rest of the items in the table above.

3) **The Median**: this is the value that is in the middle of all other values in one experiment or set of results. We are talking "counting" here and not the value of the item. For example, here is a block containing the middle items. The table comes from the sheet **Median** in the above workbook:

429
431
446
453
463
469
474
478

Excel has a function for calculating the median: MEDIAN(Range). Since the number of

items is an odd number, then 463 is exactly in the middle with 10 items below it and 10 items above it. If you change the value of the first two items from 200, 220 to 100, 110, the median will remain in the same place.

If the number of items is an even number, (say we find the Median for the range A1:A20 instead of A1:A21, the median becomes the average of the middle two items = (453+463)/2 = 458.

The median is of use when dealing with asymmetric distributions. However, you have to keep in mind that its value is to separate the items by count and not by value. In other words, in the first case with 21 values, 463 is not the middle point of the range 200 to 590. The middle point in value terms is (590-200) / 2 = 195.

In the **Median** sheet in the solved workbook, you will find examples of the MEDIAN() function.

4) **The Mode**: this is the highest appearing statistic or value in the range. In our case, Excel gives the mode = 457.0000. However, you may face #NA as an answer because we might have more than 1 value that fits the definition of the mode. Excel has a function for finding the mode: MODE(Range).

Refer to the **Mode** sheet and do the following while observing how the Mode changes:

a) The values in Col A have no mode.

b) Duplicate the cell G3 downward to G5 and watch the mode in J1 change.

c) Do the same in Col M and then duplicate Cell M10 down to M12. There are two modes. Excel cannot show them. (They can only be seen visually if a histogram of the column is plotted).

5) **The Standard Deviation and the Variance**: these will be discussed in the next section where we discuss the Standard Error and related measures.

6) **Kurtosis**: this is a measure of whether the shape of the data in a distribution has a peak or is flat. You can also say that it is a measure of whether the items in the region of the tails have a higher probability than those in a Normal Distribution.

Kurtosis >0 shows sharp peaks or light tails
Kurtosis <0 shows flattened peaks with heavy tails

Excel has a function for calculating Kurtosis: KURT(Range). Kurtosis is infrequently used in Monte Carlo Simulation.

7) **Skewness**: this is a measure of "a-centrality". A distribution with zero skewness is centralized or symmetrical. The Normal Distribution has a skewness = 0. As the peak moves to the left, the right hand side tail gets longer. This means that there are many more observations with a lower probability than there are with a higher probability. This is called right skewness and has a positive value. Left skewness is the reverse and is negative. Skewness varies from -1 to +1.

8) **The range, minimum, maximum, sum and count** are self-explanatory. In fact, while setting up histograms, we will often calculate the bin size using manually calculated min, max and range. An alternative way is to generate the Descriptive Statistics first and then use these values from the table. Excel has the following functions to calculate these measures (except for Range which is simply the difference between maximum and minimum): SUM(Range)

COUNT(Range) or COUNTA(Range) for alpha entries
MAXIMUM(Range)
MINIMUM(Range)

9) **The Confidence Interval**: again, this will be discussed in the next few sections with the other terms that we postponed: standard deviation, variance and the standard error.

B) Sheet: St Dev - the Variance and Deviation from the Mean

The following measurements or statistics are related. They gain clarity by being discussed together:

The Variance and the Standard Deviation
The Standardized Normal Distribution
The Cumulative Normal Distribution

They are essential to the understanding of the next two sections on confidence intervals and sample sizes.

The Variance and Standard Deviation measure the **spread** that observations have from the mean of the population. The Variance is simply the square of the Standard Deviation. This manual procedure is not necessary as Excel has all the formulas we need to calculate these two parameters. It is presented because it gives a good idea of what these measures mean: a) Find the mean of a set of N observations.

b) Find the deviation of each observation from the mean = Mean - X

c) Square each deviation = (Mean - X1)^2, (Mean - X2)^2.... (Mean - Xn)^2

d) Find the variance = the average of the squared deviations. Add all the squared deviations and divide their sum by N. The variance is always given in the square of the units of the observations. The variance of a set of observations measuring the length of rods in centimeters is in centimeter squared.

d) The standard deviation = the square root of variance which is expressed in the same units as the observations.

All sets of observations can have their Variance and Standard Deviation calculated. However, the Normal Distribution has particular features that we can use for quick analysis.

We will most often use the Standard Deviation rather than the variance. The reason is that the Standard Deviation has the same unit as the observations (cm's, kg's, minutes,

etc.) whereas the Variance is the square of those units. The Standard Deviation is often found in other distributions too. Excel has several functions to calculate the Standard Deviation which invite confusion.

=STDEVP(Range) - the standard deviation for a full population
=STDEV(Range) - the standard deviation of a sample of observations

These are legacy functions used in versions of Excel before 2007. Since then, Excel has introduced new functions but kept the older functions for compatibility:

=STDEV.P(Range) - Population with numeric values
=STDEV.S(Range) - Sample from a population with numeric values
=STDEVP(Range) - Legacy function replaced by STDEV.P
=STDEV(Range) - Legacy function replaced by STDEV.S
=STDEVPA(RangeTEXT) - Population with numeric, text and logical values
=STDEVA(RangeTEXT) - Sample with numeric, text and logical values

In the workbook, the **St Dev** sheet shows various examples of the above. The reason that STDEV.P and STDEV.S differ is the following.

Assume you have a large population such as "All our clients". Say we have around 40,000 clients. We can calculate the standard deviation of what the volume of their purchases from our company. For this, we use STDEV.P(). Suppose we do not know the population standard deviation. We can take a sample of 300 clients and calculate the standard deviation of their purchases. In such a case and for strictly mathematical reasons, the variance as calculated above where we divided by N should now be divided by N-1. To check this, simply multiply the STDEV.S by SQRT((N-1)/N) and you will get the same as STDEV.P. This is expanded in the **St Dev** sheet in the above workbook.

Just keep in mind that the sample standard deviation is always larger than that of the population standard deviation. This has an intuitive reason. Let us say that S is the standard deviation of your clients' purchases (population). Even though a sample would come from the same population, we would expect it to be more widely spread than the total population.

C) Sheet: Normal Distribution - Standardizing the Normal Distribution

Since every sample would have its parameters, we cannot compare them easily. For example, a student scores 76 out of 100 in Mathematics and 82 out of 100 in Chemistry. It looks like the student has done better in Chemistry. However, if we consider the data from the rest of the class, we can calculate the following: a) Mathematics grades have a mean = 70 and a standard deviation = 5

b) Chemistry grades have a mean = 80 and a standard deviation = 3

In order to be able to compare the two grades in the different classes, we need to standardize them. There are two required steps. First of all, the two means are different,

so we need to shift them so they are the same. Generally, we shift both sets of data so the mean becomes = 0. Secondly, their standard deviations are different. For example, the math score 76 is 7 points above the mean of math grades where as the chemistry score of 82 is only 2 points above its mean. Statisticians decided to work with "units of" standard deviations.

Let us assume we want all sets of observations to be standardized to a single distribution whose mean = 0 and whose standard deviation = 1. Here is how we standardize the above grades. First we shift the observations (the two grades) so their average = 0: a) The mathematics grade = 76 becomes = (76 - 70) = 6. This means it is 6 points above the average

b) The chemistry grade = 82 becomes = (82 - 80) = 2. This means it is 2 points above the average

But since the spread is not the same in the two classes, standardize the above scores by dividing each by its own standard deviation:

c) The standardized score in mathematics = shifted average divided by the standard deviation = 6/5 = 1.2

d) The standardized score in chemistry = 2/3 = 0.67

Our scores are now called **standardized scores**. They are no more in their original units such as grades, centimeters, pounds, dollars. They are now in **units of standard deviation**. We can read the new scores as follows: the mathematics grade is 1.2 which is 1.2 standard deviations above the standardized mean and the chemistry grade is 0.67 which is 0.67 standard deviations above the standardized mean. The mathematics grade is therefore "relatively" better than the chemistry grade even though the chemistry measurement was higher.

Here is the standardization formula:

Z = (Observation - Mean) / Standard Deviation

Excel has a function for the above = STANDARDIZE(X, Mean, St Dev).

In **Normal Distribution** sheet, there are two columns showing the standardized probability and the standardized cumulative probability.

D) Sheet: Normal Distribution - the Cumulative Normal Distribution

As we have seen in most of our results, a probability density distribution (often shortened by dropping the term "density") shows a single bar for each observation or item or event. The observations are discrete, i.e., we have counted the instances of each discrete value to produce a Frequency Table. Most mathematical functions can produce continuous distributions. We shall address these separately when we present the Exponential and other continuous distributions.

The height of the bar gives a count, a value or a probability. In our case, we will

concentrate on the probability distribution. If we start adding the probabilities for all bars from the extreme left and rightwards, we get the **cumulative distribution**. Therefore, whenever we view a cumulative distribution, we can read its value at point X as follows: the cumulative probability at point X is the probability that X is larger than all values to its left. For example, if the cumulative probability of a height 182 cm is 0.75, this means that whoever is 182 cm tall stands a chance of being taller than 75% of the population. We can also read it inversely: that person is shorter than 25% of the population.

Going back to the **Generate the Descriptive Statistics with the Analysis Toolpack** workbook in the Workouts Folder, the **Normal Distribution** sheet gives us the distribution of heights whose mean is 170 cm and standard deviation is 10 cm. Both probability density and cumulative are shown in this table (which only shows the rows for X = 160 to 180 because most of the others have near zero values).

The first column contains our observations or X values. The second column is the probability that an observation with a value = X will occur. The third column shows the cumulative probability or the probability that members of our population are smaller than X.

We use the Excel function NORM.DIST(). For a specific X, mean and standard deviation, it can return either the probability or the cumulative probability.

X = Height	Prob Density	Cumulative
160.00	0.0242	0.1587
161.00	0.0266	0.1841
162.00	0.0290	0.2119
163.00	0.0312	0.2420
164.00	0.0333	0.2743
165.00	0.0352	0.3085
166.00	0.0368	0.3446
167.00	0.0381	0.3821
168.00	0.0391	0.4207
169.00	0.0397	0.4602
170.00	0.0399	0.5000
171.00	0.0397	0.5398
172.00	0.0391	0.5793
173.00	0.0381	0.6179
174.00	0.0368	0.6554
175.00	0.0352	0.6915
176.00	0.0333	0.7257
177.00	0.0312	0.7580
178.00	0.0290	0.7881
179.00	0.0266	0.8159
180.00	0.0242	0.8413

As we shall see below, we would normally use NORM.INV() to find the differences between the various cumulative values. This will be clarified later on. For the purpose of clarifying the use of the cumulative values column, we can view the values on the chart and table more easily. In all cases below, consult the cumulative column (and keep in mind the table does not show the values from 130 - 160 and 180 - 210. Check on the chart too.

Since the mean = 170 and the standard deviation = 10, the values we use below (130,140,150,180,190,200) are all below or above the mean by multiples of 10.

a) The cumulative value **0.50** corresponds to an X = 170 cm. This means that 50% of the population is shorter than 170 cm. Would it upset you if I told you that 0.50 of your compatriots are below average intelligence?

b) The cumulative value **0.1587** corresponds to X = 160 cm. This means that 15.87% of the population is shorter than 160 cm or 84.13% of the population is taller than 160 cm.

c) The chance of finding someone whose **height is between 160 and 180** is therefore 0.8413 - 0.1587 = 0.6873. Since 160 is one standard deviation below the mean and 180 is one above it, it makes sense when you hear statistician say: around 70% of the population is concentrated around the mean and at a distance of +/- one standard deviation.

d) What about a range of 2 standard deviations away from the mean? Or, what is the chance of finding someone whose height is between 150 and 190? Just subtract the two cumulative values at these points: 0.9772 - 0.0228 = 0.9545. This is another case when statisticians say that around 95% of the population is concentrated within 2 standard deviations both sides of the mean.

e) If you wish to try 3 standard deviations, just enter:

=NORM.DIST(200,170,10,TRUE) - NORM.DIST(130,170,10,TRUE)

= - 0.999936658 or 99.9936%. Almost all of our (or any normally distributed) population is within 3 standard deviations of the mean.

E) Sheet: Confidence Intervals and Confidence Calculations

This section presents a few definitions of confidence intervals and errors and discusses their relevance to simulation:

1) The Confidence Interval
2) The Standard Error
3) The Confidence Level (1-alpha)

They are all related to the next discussion that discusses the size of samples (or simulation runs).

The Confidence Interval: this expression is often used when poll results are stated. What does it mean? In what do we have confidence? And how much? This is the context

we are working in:

a) **The Population**: this is usually large. Very often, we have no access to the mean and standard deviation of the population. In fact, if we do have access to the data about the whole population, we do not need to simulate. When we consider the mean and standard deviation of the heights of male in this town or the cholesterol level of all persons between 40 and 60, it is not possible to calculate these two measures because of the lack of access to the whole population. In some cases, the population is known but we do not have such information. For example, we know we have 20,000 employees in this company but we do not know their height or cholesterol levels. Even if we wanted to, it would be too costly to conduct such a survey. The alternative in both cases is to take a sample and assume that its mean and standard deviation are the same as those of the mother population. This is a big leap. It needs to be qualified.

Population mean = u
Population standard deviation = sigma

The Sample: this is a small sub-set of the population. When we run a Monte Carlo Simulation, we generate 1000 or 2000 runs. These are samples of all possible runs. Therefore, in our statistical analysis we should remember that we are dealing with samples and not with population. The reason we have samples is because they hopefully represent the population.

Sample mean = X
Sample standard deviation = s

In the very first Monte Carlo Simulation costing model we introduced this eBook with in chapter 6.0, the descriptive statistics showed that the mean total cost of 1000 runs = 30,563. If we ran another simulation, would we expect the mean to be the same? Quite unlikely. But we are using this result to infer that the mean of the population = 30,563? Or can we say that with the new sample, 80% of the runs would have a total cost less than 32,500? Again, most likely not. We cannot make such statements. We need to qualify them both so that they are statistically tenable.

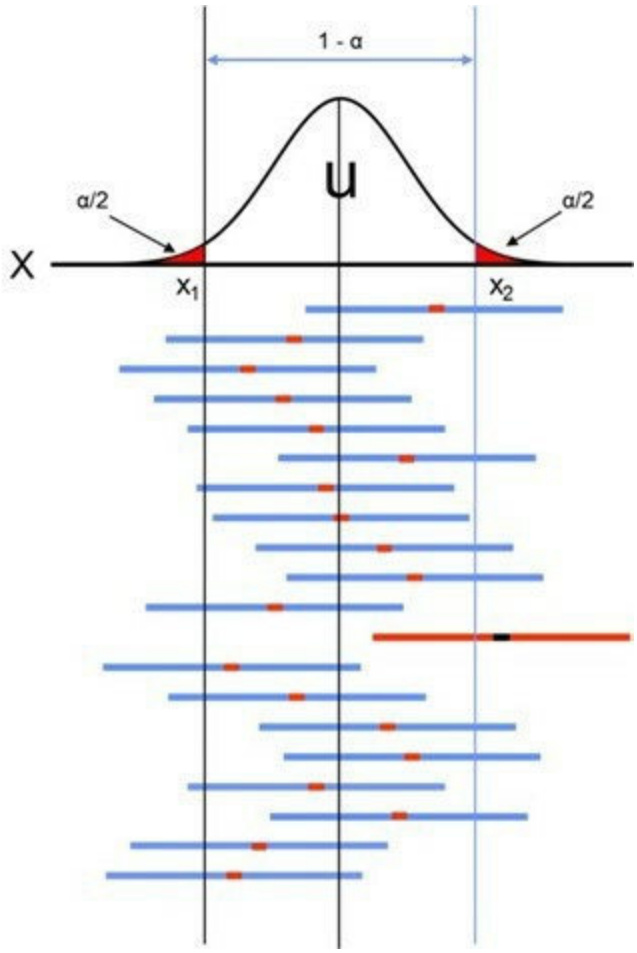
Since the mean we measured in one sample will unlikely be the same as u the population mean, we have some "uncertainty" here. Luckily, statisticians have provided us with easy to use measures that allow us to measure such uncertainties as result from the samples being different. We need to introduce the **Confidence Interval**.

In the following diagram, we show a normal distribution of the variable X (say the total cost or the cholesterol level or the amount of purchases, etc.). Even though we do not know it, let us assume that this distribution is of the total population, hence, its mean = u.

Now we introduce the confidence level. The curve in the diagram is a probability density distribution. (Each bar = the probability that X will take place). Therefore, the areas shown in red under the tails and the block in the middle represent total probabilities or percentages. We can read the lower tail area as follows: the red area

under the left tail is the probability that we will find an observation X less than X1. The same applies to the upper tail: the red area under the right tail is the probability that we will find an observation X larger than x2. Since the normal distribution is symmetrical, these two areas are equal.

Statisticians prefer to define the level by the total uncertainty found in the two tails. So, alpha = 2 * the area under each tail, shown in red in the diagram. If alpha = 5%, then each area = 2.5%. It follows that the area in between the two tails = 1 - alpha = 95%. Keep in mind that these calculations are fictitious since we do not know u. We need them as a "virtual" reference as follows.



Mathematical derivations (which we will not present) allow us to calculate the confidence interval of the true population (to be used below). This is based on using the sample mean X and standard deviation s. If we consider alpha = 5%, then the lower limit (X1) and the upper limit (X2) are calculated as follows: The 95% lower Confidence Limit = X - 1.96 * (s / SQRT (n))

The 95% upper Confidence Limit = X + 1.96 * (s / SQRT (n))

(We will explain where 1.96 comes from soon).

We can now use the values taken from the Descriptive Statistics of the Equipment Costing simulation:

Total Equipment Costs

Mean	30,563
Standard Error	65
Median	30,612
Mode	#N/A
Standard Deviation	2,042
Sample Variance	4,167,784
Kurtosis	-0.97
Skewness	-0.09
Range	9,456
Minimum	25,292
Maximum	34,749
Sum	30,563,253
Count	1000
Confidence Level(95.0%)	127

The 95% lower Confidence Limit = 30,563 - 1.96 * (2042 / SQRT (1000))
The 95% upper Confidence Limit = 30,563 + 1.96 * (2042 / SQRT (1000))
The lower limit = 30,436
The upper limit = 30,690
The range = 253 (the difference between the limits corresponding to 95%)

The upper limits are our Confidence Interval and they allow us to make such a statement: we can be 95% confident that the true mean of the population of total equipment costs falls somewhere between 30,436 and 30,690. But what does "95% confidence" mean?

Say that we have conducted 20 simulations as in the diagram above. This means we have 20 samples of 1000 runs each. If we construct a confidence interval each of the samples and plot them with respect to the real interval (which we do not know), we can say that 1 out of 20 confidence intervals will probably not include u, the population mean. This is shown by the sample with the red bar. Another way of saying it is that there is a 5% chance we might have fallen on a sample whose mean is outside our confidence range.

If we are not happy about 1 in 20, we can increase our confidence but not before we find out where the 1.96 came from.

In the Distributions Folder there is a workbook called **The Normal Distribution**. In the Standardized column D the heights were standardized using the STANDARDIZE() function so the mean = 0 and the standard deviation = 1. Col E shows the standardized cumulative probability using the NORM.S.DIST() function.

We can find the inverse of the normal distribution for z in the same manner as we found

it for X. Remember that the cumulative distribution can answer questions such as:

Given a value X, what is the probability that all observations fall below it?
Given a standardized value Z, what is the probability that all observations fall below it?

The inverse cumulative distribution asks the question in reverse:

Given a probability p, what is X for which all other X's are smaller?
Given a probability p, what is Z for which all other Z's are smaller?

We have to use different Excel function for standardized observations. It answers answer the same question: Z = NORM.S.INV(p).

For a confidence interval of 95%, **alpha = 5%**. The lower tail covers an area = alpha / 2 = 2.5%. This points to the observation X1 in the above diagram.

As for X2, the probability covered below it = the red area of the lower tail (alpha *2 = 2.5%) and the confidence interval (1-alpha) = 95%. and the same goes for the upper tail. This is equal 1 - alpha* 2 = 97.5% = 1 - 0.025.

Z(alpha/2) = NORM.S.INV(0.025) = -1.96
Z(alpha/2) = NORM.S.INV(1.00 - 0.025) = +1.96

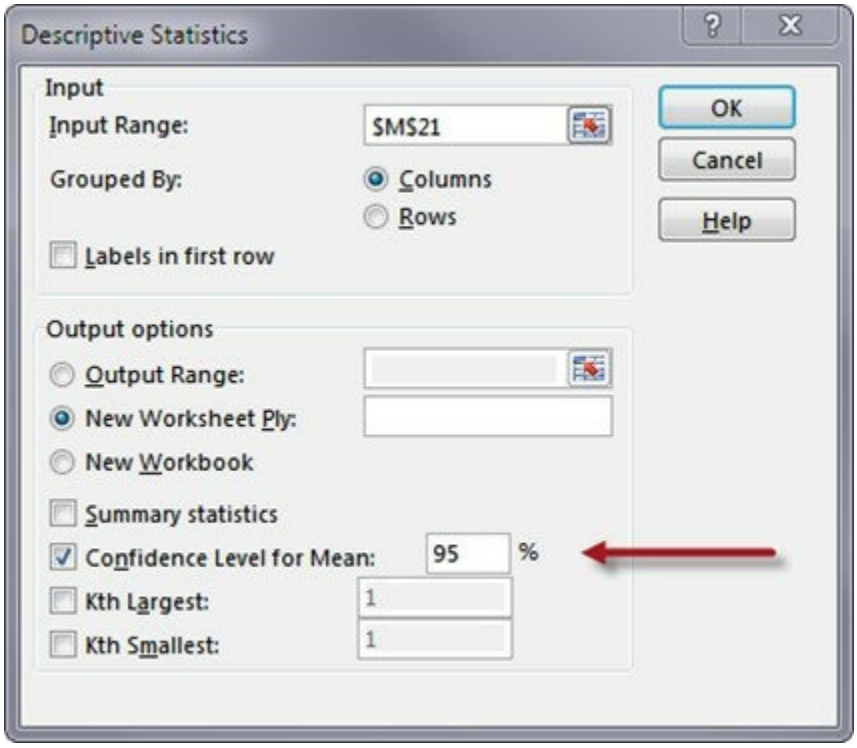
So, if we want a wider **Confidence Interval or 99%**, say with alpha = 1%, we have to find the Z corresponding to that. Alpha divided by 2 = 0.005 and 1 - alpha / 2 = 0.995

Z(alpha/2) = NORM.S.INV(0.005) = -2.5758
Z(alpha/2) = NORM.S.INV(1.00 - 0.005=0.995) = +2.5758

This widens our confidence interval we use **alpha = 1%** as follows:

The 99% lower Confidence Limit = 30,563 - 2.5758 * (2042 / SQRT (1000))
The 99% upper Confidence Limit = 30,563 + 2.5758 * (2042 / SQRT (1000))
The lower limit = 30,397
The upper limit = 30,729
The range = 333 (the difference between the limits corresponding to 99%)

In the Descriptive Statistics dialog box, there is an option to show the **Confidence Interval (1-alpha)** defaulted to 95%:



When the results are shown, for our example (see results above), this is given as 127. Actually, this is a misnomer. The item "Confidence Level (95%)" shown is = 1.96 * (Sample Mean / SQRT(n)). It is real the area under one tail, for alpha = 5% in this case. You can change the percentage as you wish.

This is often referred to by statisticians as the **Maximum Error of Estimate** or the **Margin of Error**. These are better names since they indicate the amount to be subtracted or added to the Mean to give us the Confidence Interval.

Sometimes, statisticians use **alpha = 10%**. Using the above formulas, we get:

The 90% lower Confidence Limit = 30,563 - 1.645 * (2042 / SQRT (1000))
The 90% upper Confidence Limit = 30,563 + 1.645 * (2042 / SQRT (1000))
The lower limit = 30,457
The upper limit = 30,669
The range = 212 (the difference between the limits corresponding to 90%)

Conclusion: the higher the confidence, the wider the interval. We saw above that to increase our confidence level, we have to take provision by considering a wider range where the population mean might fall. We can give an example from real life. Your brother is often late. You are asked to give a time when he will arrive and you say, I have 95% confidence he will arrive between 12:55 and 13:05. Someone says, how did you arrive at that? You say: if we monitor his arrivals, there is a chance of 1 out of 20 he will arrive outside this interval. Now they want you to be more precise, say 1 in 100. In order to be more specific, you state that you have 99% confidence he will arrive between 12:45 and 13:15.

Using our equipment costing results,

For alpha = 10%, lower limit = 30,457, upper limit = 30,669 and range = 212

For alpha = 5%, lower limit = 30,436, upper limit = 30,690 and range = 253
For alpha = 1%, lower limit = 30,397, upper limit = 30,729 and range = 333

You can see how the range increases (or precision decreases) as we increase our confidence.

There is a price to pay for increasing confidence: reducing precision! This has a solution we will consider in item 8 below: increasing the sample size.

Finally, the last sheet in the workbook is called **Confidence Calculations** and shows a variety of values for the above.

F) The Standard Error and the Confidence Level (1 - alpha)

This is another item in the Descriptive Statistics. For our example, it is = 65. It is the factor in the interval calculation without the Z value: Mean / SQRT(n).

In our case = 2042 / SQRT(1000) = 64.57370982 which Excel rounds to 65.

Once we know the Standard Error, we can simply multiply it by the standardized Z values: 1.645 (for alpha = 10%), 1.96 (for 5%) and 2.5758 (for 1%).

The Confidence Level (1 - alpha) is found in Microsoft's Descriptive Statistics and is listed if requested in the dialog box (refer to the first section in this chapter).

G) Determining the Size of Samples

We saw above how we pay for increasing our confidence by widening our range or interval (or reducing our precision). There is another way of increasing our confidence without such a price. We have to remember the formula for the **standard error** = Sample Mean / SQRT(n).

For a sample of 1000 and a sample mean = 2042, the standard error = 2042 / SQRT(1000) = 64.57.

If we increase the sample to 2000, the standard error = 2042/SQRT(2000)=45.66.

This factor is then multiplied by 1.645, 1.96 or 2.5758 depending on the alpha level we chose. We found out that for a sample of 1000, we had the following intervals:

For 90%:

The **90%** lower Confidence Limit = 30,563 - 1.645 * (2042 / SQRT (1000))
The **90%** upper Confidence Limit = 30,563 + 1.645 * (2042 / SQRT (1000))
The lower limit = 30,457
The upper limit = 30,669
The range = 212 (the difference between the limits corresponding to 90%)

For 95%:

The **95%** lower Confidence Limit = 30,563 - 1.96 * (2042 / SQRT (1000))
The **95%** upper Confidence Limit = 30,563 + 1.96 * (2042 / SQRT (1000))

The lower limit = 30,436
The upper limit = 30,690
The range = 253 (the difference between the limits corresponding to 95%)

For 99%:

The **99%** lower Confidence Limit = 30,563 - 2.5758 * (2042 / SQRT (1000))
The **99%** upper Confidence Limit = 30,563 + 2.5758 * (2042 / SQRT (1000))
The lower limit = 30,397
The upper limit = 30,729
The range = 333 (the difference between the limits corresponding to 99%)

Using a sample n = 2000, we get:

For 90%:

The **90%** lower Confidence Limit = 30,563 - 1.645 * (2042 / SQRT (2000))
The **90%** upper Confidence Limit = 30,563 + 1.645 * (2042 / SQRT (2000))
The lower limit = 30,488
The upper limit = 30,638
The range = 150 (the difference between the limits corresponding to 90%)

For 95%:

The **95%** lower Confidence Limit = 30,563 - 1.96 * (2042 / SQRT (2000))
The **95%** upper Confidence Limit = 30,563 + 1.96 * (2042 / SQRT (2000))
The lower limit = 30,474
The upper limit = 30,652
The range = 179 (the difference between the limits corresponding to 95%)

For 99%:

The **99%** lower Confidence Limit = 30,563 - 2.5758 * (2042 / SQRT (2000))
The **99%** upper Confidence Limit = 30,563 + 2.5758 * (2042 / SQRT (2000))
The lower limit = 30,445
The upper limit = 30,681
The range = 235 (the difference between the limits corresponding to 99%)

In the Workouts Folder there is a workbook called **Confidence Interval and Sample Size**. In the **Confidence** sheet there is a table that calculates all the above for 3 different sample sizes and 3 different alpha levels:

Mean	Standard Deviation of Sample	Alpha	Sample Size	Z Multiplier	Standard Error	Confidence Level (1-alpha)	Lower Limit	Upper Limit	Range
23,394	1,974	0.100	1000	1.645	62.44	102.70	23,291	23,496	205
23,394	1,974	0.050	1000	1.960	62.44	122.37	23,271	23,516	245
23,394	1,974	0.001	1000	3.291	62.44	205.45	23,188	23,599	411
23,394	1,974	0.100	2000	1.645	44.15	72.62	23,321	23,466	145
23,394	1,974	0.050	2000	1.960	44.15	86.53	23,307	23,480	173
23,394	1,974	0.001	2000	3.291	44.15	145.27	23,248	23,539	291
23,394	1,974	0.100	4000	1.645	31.22	51.35	23,342	23,445	103
23,394	1,974	0.050	4000	1.960	31.22	61.19	23,332	23,455	122
23,394	1,974	0.001	4000	3.291	31.22	102.72	23,291	23,496	205

You can see that the size of the sample has more impact on the Confidence Range than the alpha. For example, the range = 212 for alpha = 10% and n = 1000. When we quadruple the sample to 4000, we half the range to 106.

Having avoided the price of reduced precision, we face the price of increased sampling. This would have been drastic if the sample size consisted of questionnaires, interviews or visits. Since we are using Monte Carlo Simulation, the price of increasing the sample size is left on the shoulders of Excel.

H) How to Use the Inverse Functions based on Cumulative Distributions?

We have made several statements in our analyses of the form: around 90% of the time, more than 84% of the population, *etc.* There are two ways we can arrive at such statements. In order to appreciate a very critical concept in simulation, we should remember that the "DIST" functions in Excel work as black boxes:

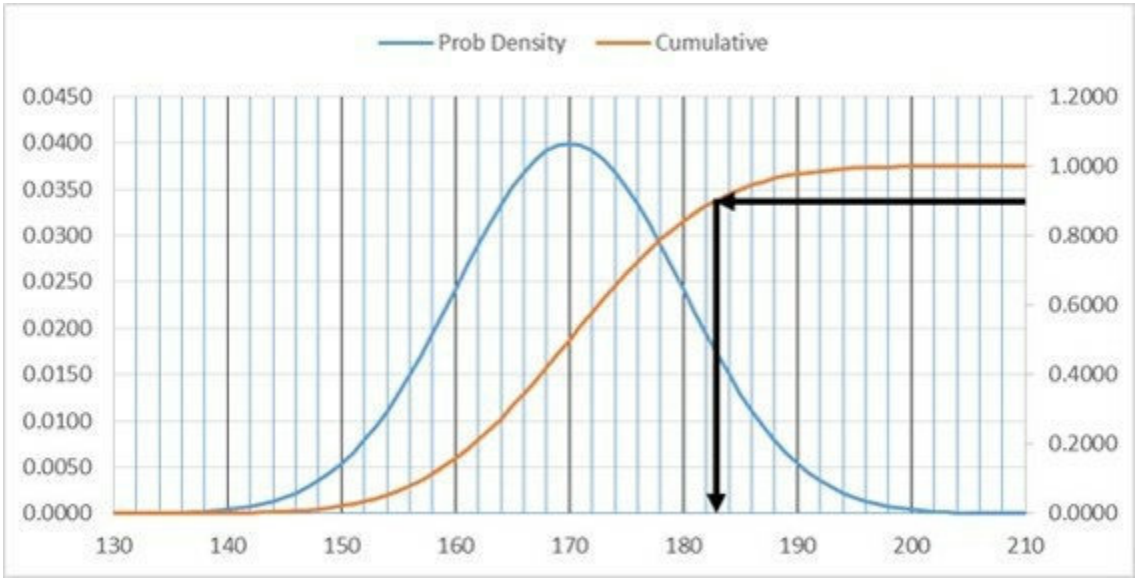
The DIST() functions can also give cumulative probabilities as shown above. This is more important in our case.

We need an **inverse** function INV(). This works as follows:

An INV() function will request a specific probability p. It will return the value of X or the observation for which the proportion p of the population is smaller. For example, if you enter 84.13% or 0.8413 as an argument in NORM.INV(), (assuming that you have also indicated the average and the standard deviation of your population), NORM.INV() will return the value 180 cm. This indicates that 84.13% of the population is shorter than 180 cm.

We can use the inversion concept on the chart and via an Excel function:

a) **The Chart**: draw a line from the right hand axis horizontally from 90% and to the left. Where it hits the cumulative curve, drop a line down and read X:



So 90% of our population is shorter than 183 cm.

b) **The Excel Function**: NORM.INV(Probability, Mean, St Dev). In our case, we enter 0.9 for the probability, 170 for the mean and 10 for the Standard Deviation. The result = 182.8155157 cm. The NORM.INV() function inverts the cumulative distribution. Give it a cumulative probability, it gives you an X.

This is one of the most critical concepts in sampling. In many cases, Excel provides INV() functions such as:

BETA.INV()
CHISQ.INV()
CHISQ.INV.RT()
F.INV()
F.INV.RT()
FISHERINV()
GAMMA.INV()
LOGNORM.INV()
NORM.INV()
NORM.S.INV()
T.INV()
T.INV.2T()

For the other statistical functions that we need such as POISSION(), EXPON.DIST() and HYPERGEOM.DIST(), we do not have Excel INV() functions. We need to revert to workarounds mostly consisting of looking up values in a cumulative table.

16.0 Appendix B: Basics of Simple Linear Regression

This chapter briefly introduces the principles of regression which we have used in several workouts in this eBook.

Workout 20: Regression Examples

Refer to the workbook **Regression - Examples** in the Supporting Documents folder for some examples to be discussed below.

Simple Linear Regression (SLR): this is a statistical method that works as follows. When you have two sets of data (2 columns), you can consider one as the independent variable. This is the one that changes without your control. The dependent variable is the result of the changes in the independent variable. You will be lucky if there is a formula for these variables. For example, Y = aX + b shows X as the independent variable and Y as the dependent variables. If we know a and b, we can predict the value of Y for any given value of X. The relationship is simple because there is only one independent variable. The relationship is linear because X is raised to the power 1 and has no other functions applied to it. (There are cases where Y = a * Log(X) + b which are obviously not linear).

When we have no relationship, but only have data in two columns, Excel can calculate a and b in various way (see the functions SLOPE, INTERCEPT, FORECAST, LINEST, etc.) If you are the hardy type, you can also use statistical formulas to calculate a and b. Finally, Excel allows you to view the formula for paired data on a chart of such data. This is the method we will use.

Multiple Linear Regression (MLR): this is similar in logic but more complex to calculate than SLR. This is the case where you have several independent variables X1, X2, X3 and so on. All of these contribute to the value of the dependent variables. For example, an equation can be Y = aX1 + bX2 + cX3 +dX4 + e.

However, since the coefficients are not enough to give a good look at the equation (we need to incorporate other results in MLR that Excel can give), we will not consider this test in Monte Carlo Simulation.

MLR is only provided in Excel by the Analysis Toolpack whereas the Simple Linear Regression can be analyzed by the Toolpack and through various native functions.

Non-Linear Regression: this is the case where the X does not follow linear values. Non-linear can mean: exponential, logarithmic, polynomial, *etc.* We can only solve for non-linear coefficients in Excel by showing the equation in the chart depicting the paired

data.

17.0 Appendix C: Miscellaneous Excel Facilities

In this chapter we shall introduce some of the procedures used in Excel throughout this book.

A) Use Spinners and Scrollers to Monitor Changes in the Model

Spinners are very useful when you need to increment a value in a cell and observe another. For example, you can change the escalation percentage and observe what happens to the net profit. (In a way, this is a minute simulation, hand crafted).

You can also use spinners when animating a graph. As you increment or decrement a specific cell, the output on the graph changes accordingly.

Workout 21: How to Use Spinners and Scrollers

Purpose: to setup a formulation that uses a spinner to vary a value. The value feeds into a table that is charted. As you increment or decrement the value, the chart changes.

Problem statement: Excel only allows the use of spinners for positive integers (0 included). When you need negative values or fractional values, you need a little work around that we describe below.

Note: out of the box, Spinners and Scrollers do not allow you to increment fractional or negative values. In the coming steps, we resort to scaling to resolve this issue.

Step 1: create a new workbook and save it under any name you wish. In the Workouts Folder there is a fully solved model called **Spinners and Scrollers**.

Rename the first sheet as Spinners.

Step 2: enter the following labels:

A1 = Present Value
B1 = Future Value
E1 = Periods
E2 = Rate

Step 3: use autofill to generate the sequence 100, 200... 1500 in the range A2:A16. The range contains present values (deposits). We wish to find the future value of each using the number of periods (F1) and the rate found (F2).

Step 4: enter 12 as the number of years or periods in F1.

Enter 0.1 as a test value for the rate in F2. This is really 10% but it is better to use the

decimal format.

Step 5: use Excel's PV() function to calculate the future value of the cells in A2:A16.

B2 =FV ($F$2, $F$1,, A2)
F2 is the rate
F1 is the number of periods
The 3rd argument is a payment which is not used in this function
A2 is the present value or the deposit we wish to compound

The result is shown as a negative number because in accounting, when we deposit, we debit (the PV) and when we withdraw, we credit (the FV which is negative).

Copy B2 down to B16.

Step 6: insert a scatter diagram for the range A1:B16. Right click on the chart and fix the ranges so you can see how the line moves when you use the spinner in the next step.

a) Right click on the values in the Y-axis (FV)

b) Select FORMAT AXIS

c) Enter -10,000 in the Minimum and 0 in the Maximum. Now the scale is fixed.

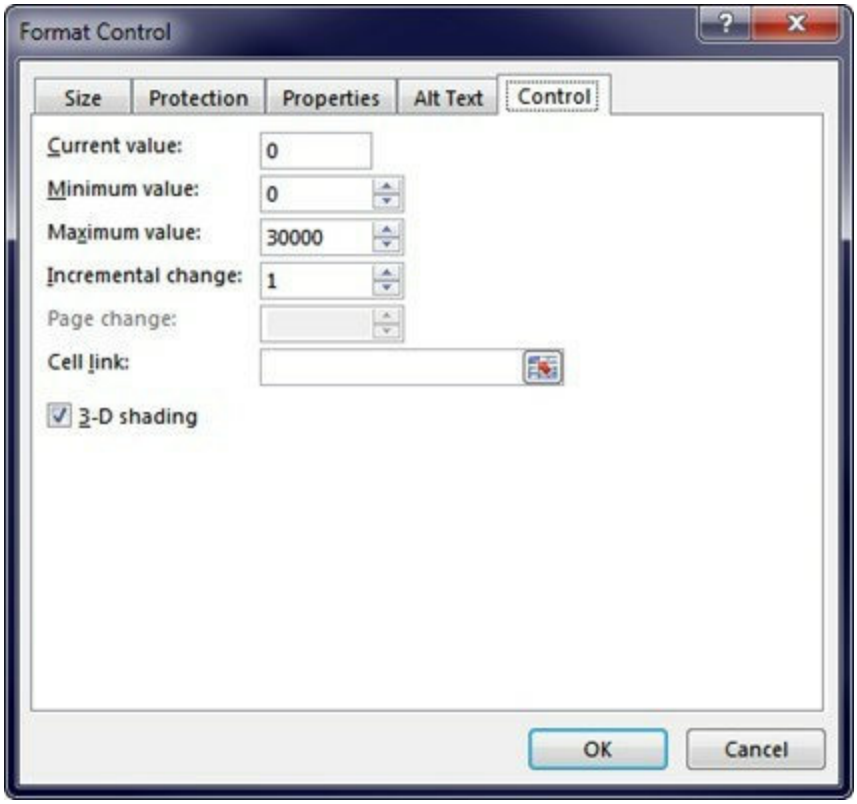
Step 7: we will now use a spinner to vary the number of periods from 6 to 18.

a) Select the menu item DEVELOPER *CONTROLS* INSERT

b) From the available Form Controls, select the spinner:

Draw a spinner using the mouse and make it as high and wide as you wish.

c) Right click on the spinner and select FORMAT CONTROL. The spinner dialog box comes out with default values:



d) Change the minimum value to 6, the maximum value to 18 and keep the incremental change as 1.
e) In the Cell link, point to F1 and press OK.

From now on, as you spin the spinner upwards or downwards, the value of F1 will change according to the parameters entered above.

Step 8: to create a spinner for the rate, we need the work around to force Excel to increment F2 in fractional values. We need an interim cell to which the new spinner will point:

a) Create a new spinner as before
b) Let the minimum be 50 and the maximum bet 150

c) Let the incremental change be 10
d) Point the cell link to G2

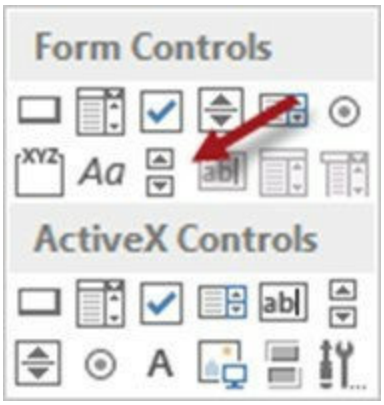
Now if you spin the spinner, G3 will change from 50 to 150 in steps of 10.

Step 9: since the spinner does not allow you to enter fractional increment values, you need to trick it as follows. Enter a formula in F2 = G2/1000. The Spinner should point to G2 while G2 is used in F2 to get the fractional value. This forces our rate to change from 0.05 to 0.15 in steps of 0.01.

View the chart as you spin the spinner.

Step 10: **how to create a scroller?**

A scroller is also found in the set of Form Controls:



It behaves in the same manner as the spinner with two useful additional facilities:

a) You can specify a page count so that instead of spinning by 1, you can also scroll by 10 or 20.
b) You can slide the little bar on the scroller.

Let us create a scroll that acts in the same manner as our first spinner:

a) Create a scroller from DEVELOP/CONTROLS / INSERT
b) Let the minimum be 6 and the maximum be 12
c) Keep the incremental change as 1 and enter the Page change as 3
d) Point the cell link to F1

Now you can use the little arrows on the scroller to spin by 1. You can click in the area between the bar and the arrows to spin by 3. You can also use the little bar as a slider.

Note: if you need to generate negative values, say from -10 to +10, you need to follow a trick similar to that in Step 9 above.

a) Let the spinner point to a specific cell, say G6 and vary from 0 to 20.
b) In F6, the cell you need to use in your formulation, place the formula = G6 - 10. If the spinner value in G6 is 0, F6 will be -10. If the spinner value in G6 is +10, the value in F6 will be +10.

In a way, Spinners and Scrollers can be thought of as “visual” tests of sensitivity.

18.0 Appendix D: Setup the VBA Sub-Runs Module

Each time we need to use the loop method, we have to ensure that the VBA Module is included in the workbook. The code is found in the file **GenerateRuns.txt** found in the Supporting Documents Folder. It is also listed at the end of this chapter.

The module works in all models, as is except for 3 elements that may need configuring:

1) The module reads the value of the **Max Number of Runs** from L1 in the Model sheet. You have to specify L1. It signifies the maximum number of rows, primary runs or replications.

2) As the VBA module loops (using FOR / NEXT), the VBA module pushes the value of the loop counter, K into L2 in the Model sheet. It signifies the **Current Run ID**. This value will be used to specify the primary run row. The row can then be filled with the results of the secondary simulation.

3) If you choose different locations than L1 and L2 or a different name for the sheet than "Model", then you have to modify these elements in the VBA module for your model. This is specified below.

How does the VBA procedure work?

a) The module defines 2 variables: K and NumberOfRuns. Both are integers.

b) The module reads the value of NumberOfRuns from the Model sheet using this VBA formula: **Worksheets("Model").Cells(1, "L")**.

c) The VBA module uses a FOR / NEXT function that uses K as a counter and loops from 1 to NumberOfRuns times in steps of 1.

d) Each time it loops, it increments the value of the index K and pushes that value into L2 in the Model sheet: **Worksheets("Model").Cells(2, "L")**. This value is referred to in our model as the Current Run ID. Once pushed into its cell, this causes Excel to recalculate the sub-runs.

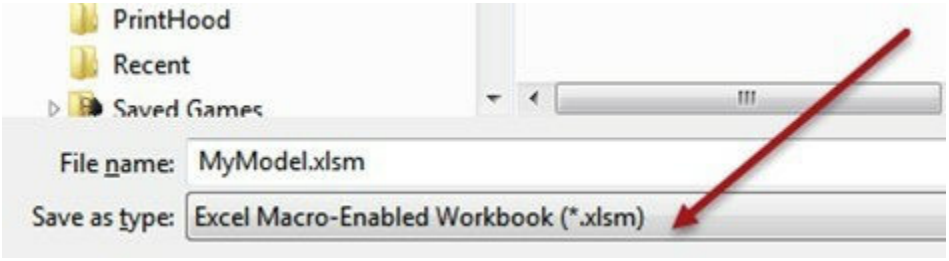
e) As Excel recalculates the sub-runs, we know which data to move from the sub-runs to the row in the runs that is specified by the Current Run ID.

Step 1: in versions earlier than 2007, it was allowed to enter macros in regular Excel workbooks. Since then, Excel only allows the use of macros in such macro enabled workbooks.

Create a new workbook using any name you wish. Ensure that it has the ".xlsm" extension. This is achieved in as follows:

a) Create any Excel workbook

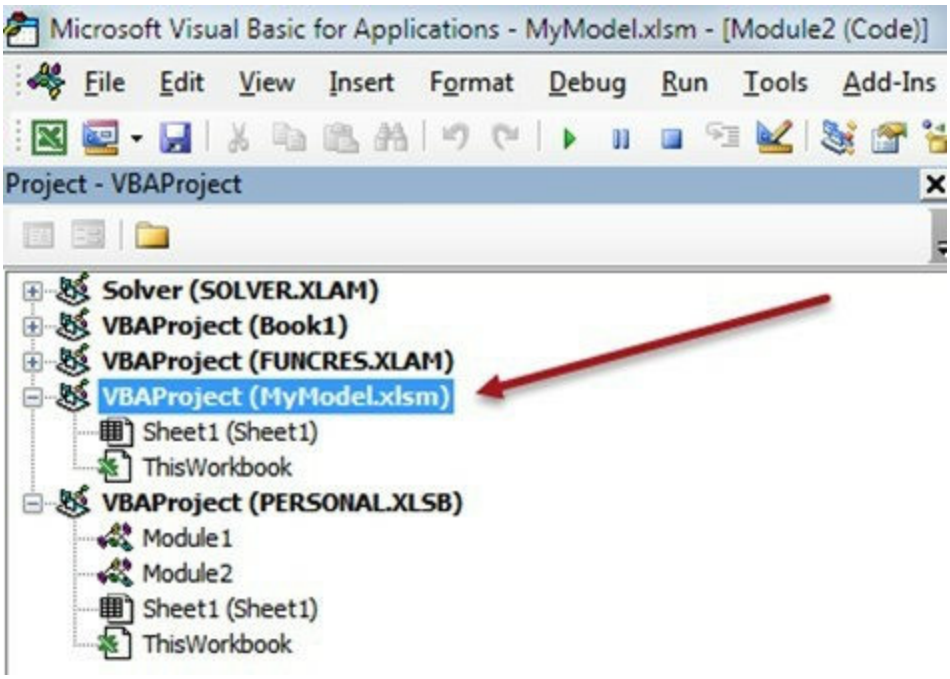
b) Use the SAVE AS option to save it as is but select the “Save as type” to be an Excel Macro-Enabled Workbook *.xlsm”.



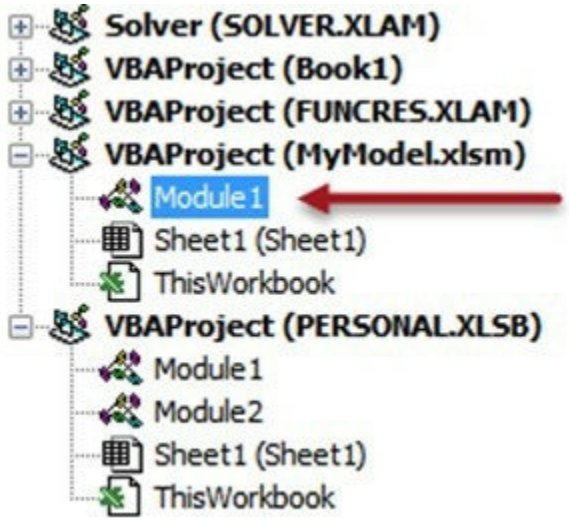
Step 2: import the VBA code from the text file.

a) Press ALT+F11 to open the VBA editor.

b) Click on the VBAProject that relates to your workbook (see the image below).



c) From the Insert menu, select Module. By default it will be named Module1 (or a higher number if you have already created other modules):



d) To rename Module1, press F4 to view its properties and rename it.

e) Open the text file in the Supporting Documents folder: **GenerateRuns.txt**

f) Select all the text (Control A) and paste the copied VBA code into Module1.

Save the file (Control S) but keep the editor open. We need it in the next step.

Step 3: configure the 3 elements discussed earlier (L1, L2 and the sheet name) as per your model. If place **Max Number of Runs** and **Current Run ID** in L1 and L2 in your model, there is no need to configure the VBA module. If your model is in the "Model" sheet, again, there is also no need to configure it.

If not, here are the lines that need to be reconfigured. Assume that only L1 and L2 are different and are in I1 and I2:

a) Edit the line at the bottom of the VBA procedure

NumberOfRuns = Worksheets("Model").Cells(1, "L")

It should be:

NumberOfRuns = Worksheets("Model").Cells(1, "I")

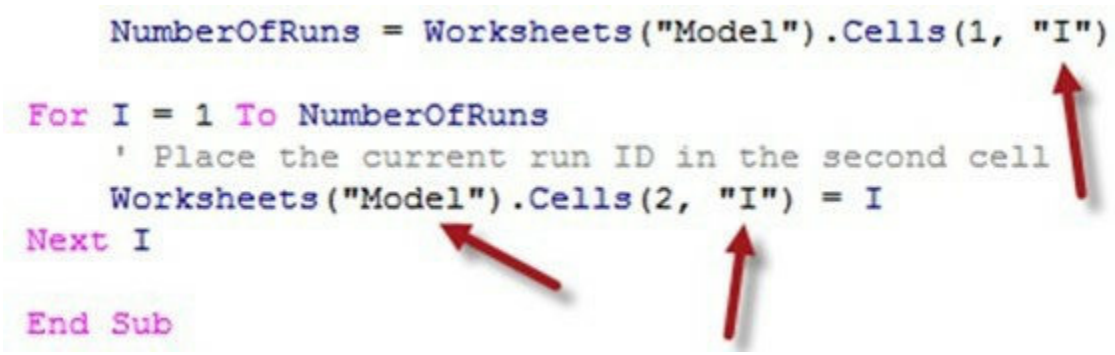
If your Control Value is not in Row 1, change the Row in the Cells(1, "I") to point to your Row. In our case: no change.

b) Do the same for the following line which places the value of the counter in the model sheet:

Worksheets("Model").Cells(2, "L") = K which should become:
Worksheets("Model").Cells(2, "I") = K
The Current Run ID will be in I2 so the above is correct.

c) Ensure that the two lines above point to the correct sheet name. In our case, we are using "Model" so no change is required. If your model is not in a "Model" sheet, change the above two lines.

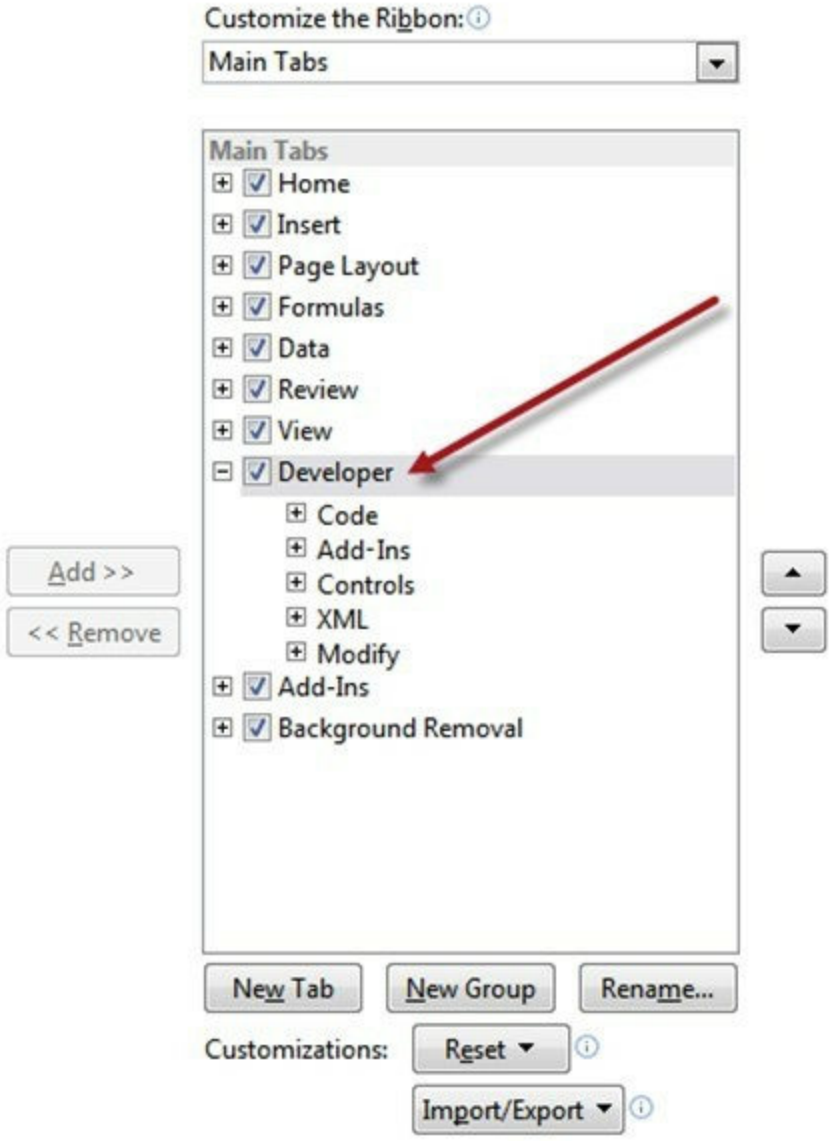
d) Close the VBA editor.



The above capture shows the module after changes to "L" in two places.

Step 4: create a button that can launch the VBA Module:

a) Select the menu item DEVELOPER *CONTROLS* INSERT. Some installations of Excel may have started without activating the DEVELOPER menu. If you cannot see it, follow these steps: FILE *OPTIONS* CUSTOMIZE RIBBON and select Main Tabs in the left panel. Make sure the Developer Menu is checked:



b) When you press INSERT, you will get this Dialog Box:

Click on the Button icon and draw a button somewhere on your sheet.

c) Right click on the button and select ASSIGN MACRO. Then select the VBA Module we just entered: **GenerateRuns**.

d) Change the name of the button to GENERATE RUNS and increase its font

e) Copy this button and paste it into the Constants sheet. We might need it while testing.

We will click this button when the rest of our model is ready.

Step 5: enable iterative calculations

One of the short comings of Excel is that you cannot use a formula in one cell to **push data into another cell or even the same cell**. Formulas always retrieve data from other cells. When in A1, you cannot develop a formula that says: if B1 > 5, place A1 in A1 else place A1+1 in A1. This is called **Circular Referencing**. Moreover, you cannot say in A1, I want H5 to be = 66.7. We need a few tricks here.

At one time or another, I am sure you were faced with a warning from Excel that says: Excel has found a "circular reference" in your formula. This means you have 2 or 3 cells that refer to one another circularly:

	A	B	C
1	=C3	=A1	
2			
3			=B1

A1 contains the value in C3. C3 contains the value in B1. B1 contains the value in A1. Where do we start? We have a vicious circle and quite rightly.

Moreover, Excel does not allow us to refer to a cell from within itself. By default, this formula is not allowed:

A1 = IF (A1<0, "Negative", A1+1)

This formula is trying to check if the value in A1 is negative in which case it enters a text. If it is not negative, the formula in A1 wishes to increment A1 itself! This is a circular reference when encountered, results in an error message from Excel.

For this, we need **circular referencing**. Excel can do it. You need to **enable** such circular referencing. Excel uses the term “circular reference” in the warning and “iterative calculation” in the Options dialog box. They are the same phenomenon.

a) Select the menu item FILE *OPTIONS* FORMULAS

b) Enable the check box "Enable Iterative Calculation".

Alert 1: once you enable this option, it will apply to all workbooks currently open and those to be opened after that. This is dangerous. You really want to avoid circularity at all costs. Ensure that the above option is only enabled when working with a single model. Reverse the option by clearing it when you are through. Otherwise, you might create a circular reference in a workbook that should not allow circularity.

Alert 2: if you enable the option and close the workbook, on re-opening it, Excel starts with the same option disabled. Ensure that whenever you open such workbooks, you enable circularity as needed.

From now on, we will refer to the current chapter when using the looping method without detailing its procedure. The next workout applies the method in a model that loops through a column of actual data.

B) The "GenerateRuns" VBA Module

Here is the VBA module in its entirety. You can copy it from here or from the **VBA - Generate Runs.txt** file in the Supporting Documents Folder.

The VBA procedure is explained in chapter 13.0 in detail where we discuss the various uses of a looping procedure that allowed us to generate sub-runs. The only lines of code are those in bold. The rest are commentary.

```
Option Explicit
Sub GenerateRuns()
Dim K As Integer
Dim NumberOfRuns As Integer

' This sub picks up the required number of runs
' from the cell in our Model sheet called NumberOfRuns
' (This cell is usually L1 and is defined as L1 in this module
' But, in case it changes in your model, you need to adjust
' the particular instance of the VBA module)

' The FOR/NEXT loops as many times as that number
' In each loop it will push the value of K (the current Run ID)

' into the second cell in the worksheet
' (again, usually L2 but can be changed by you)
```

```
' This will force Excel to recalculate
' A new run row is then generated

' We need to define two cells in the subroutine.
' They will be on our worksheet
' Note that since each model will place
' these two cells in different places,
' You have to define them every time you insert
' this module in an Excel workbook.


' In your worksheet:
'
' a) Enter the label "Number of Runs" in cell (Row, Col)
' b) Enter the label "Current Run ID" in cell (Row+1, Col)
' c) Point the code below to cell (Row, Col+1)
' to retrieve the maximum number of runs, to be used in the loop
' d) Point the code below to cell (Row+1, Col+1)
' to display the current Run ID (from within the loop)
' e) Make sure that the worksheet name is correct.
' In most cases, it will be "Model".

' We use the format: Cells(n, "XXX")
' Where the row is n and the col is
' the text value within quotation marks

' Pick up the number of runs from the first cell:
' ------------------------------------------------------------------------


NumberOfRuns = Worksheets("Model").Cells(1, "L")

For K = 1 To NumberOfRuns
' Place the current run ID in the second cell
Worksheets("Model").Cells(2, "L") = K
Next K
End Sub
```

Note: for each implementation of this VBA procedure, you have to define the cells you used in your sheet for the number of runs and the current run number. In this version, they are L1 and L2. Yours may be different.

You also need to ensure that the worksheet name in the last line in the loop is correct. In general, we use "Model" but in some cases, this would be different.

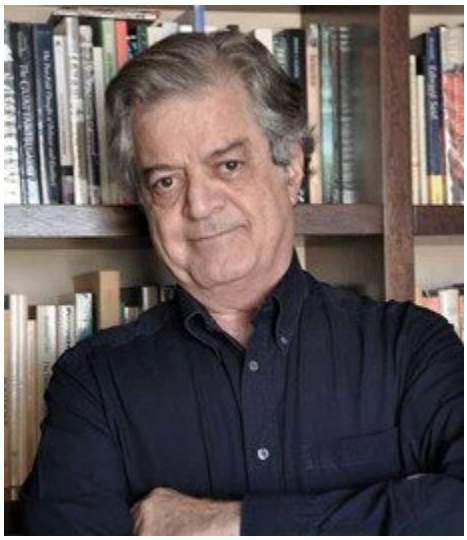
19.0 Appendix E: Acronyms and Abbreviations

Here are some acronyms used in the eBook:

AQL	Acceptance Quality Level
AOQ	Average Outgoing Quality
AOQL	Average Outgoing Quality Limit
ASN	Average Sample Number
ATI	Average Total Inspected
CPM	Critical Path Method
DIST	Distribution (used in Excel functions)
INV	Inverse (used in Excel functions)
LPD	Lot Percent Defective
LTPD	Lot Tolerance Percent Defective
PDF	Probability Density Function
PERT	Project Evaluation and Review Technique
RQL	Rejection Quality Level
RMSE	Root Mean Square Error
SLR	Simple Linear Regression
TTF	Time to Failure
TTR	Time to Repair
VBA	Visual Basic for Applications

20.0 Meet the Author

Akram Najjar completed a B. Sc. in physics and mathematics at the American University of Beirut (1966), Lebanon. While completing his requirements, he took all of his electives in literature and philosophy, subjects that never left him.



He then completed another B. Sc. in electrical and electronic engineering from the Hatfield Polytechnic in England (1969). (Now it is the University of Hertfordshire). As for the Masters, he completed course requirements for a Masters Degree in Systems Engineering at the American University of Beirut (1972).

1967-1968 and between University terms: he spent 6 months as a trainee with Standard Telephones and Cables (ITT) in England and later in 1968, another 6 months with Standard Elektrik Lorenz in Germany in the Telemetry Design Lab. In 1970, he started a 5 year job with Middle East Airlines as a Senior Systems Analyst in the computer center. During the first 3 years he was in charge of implementing large computer projects. During the last 2 years he was responsible for defining and planning MEA's future computerization projects.

From 1975 - 1977 worked on various projects inside and outside Lebanon. In 1978, established Database Sarl, Beirut, a computer consultancy and software company. Carried out a variety of projects: consultancy, software development, management training. Most of the work was in banking and finance.

In 1982, Akram established Infotech, Dubai (United Arab Emirates), to cover a variety of IT projects in the UAE and Gulf countries. Most of the projects were in trade, industry, education, healthcare, banking, construction and media/advertising. Several projects were completed in the public sector. He was also active in delivering various Quantitative Management seminars.

In 1995 to date: after his move back to Beirut, Lebanon in 1995, he left the software development field and concentrated on IT contract work: IT consulting, computer project integration, project management, software application design and management training. He also shifted into the field of business technology concentrating on strategic planning (for both private and public sector organizations), project management framework development, business process reengineering and process mapping.

In 1997, he established InfoConsult Sarl, in Beirut, Lebanon.

In parallel with his consulting work, Akram focused on management training developing and conducting workshops for the above subjects. This and other books are based on the experience acquired in these workshops. They have been published by www.xinxii.com the well known German aggregator. The eBooks are fully detailed on his website: www.marginalbooks.com.

In parallel with his consulting work, Akram focused on management training developing and conducting workshops for the above subjects. This and other books are based on the experience acquired in these workshops. They have been published by www.xinxii.com the well known German aggregator. On that site are also found Akram's puzzle books and literary works.

His eBooks are fully detailed on his website: www.marginalbooks.com. His puzzle eBooks are detailed on their dedicated website: www.thehiddenpaw.com.

Use the code mcxmbplk1 to open the zipped file mentioned in Chapter 2.0.

Table of Contents