Assessing and Communicating Uncertainty

Technical Documentation for

Monte Carlo Template

Document Version **v1**

Contents

[1 Model Overview 3](#_Toc15651710)

[1.1 Model Contents 3](#_Toc15651711)

[1.2 Model Versions 3](#_Toc15651712)

[1.3 Analytical Quality Assurance 3](#_Toc15651713)

[1.3.1 Developer Testing 3](#_Toc15651714)

[1.3.2 Validation & Verification 3](#_Toc15651715)

[2 Inputs 4](#_Toc15651716)

[2.1 Register 4](#_Toc15651717)

[2.2 Inputs 4](#_Toc15651718)

[3 Assumptions 6](#_Toc15651719)

[4 Calculations 7](#_Toc15651720)

[4.1 Calculations 7](#_Toc15651721)

[4.1.1 Occurrence calculations 7](#_Toc15651722)

[4.1.2 Value calculations 7](#_Toc15651723)

[4.1.3 Output 7](#_Toc15651724)

[4.2 Current Run 7](#_Toc15651725)

[4.3 Outputs 7](#_Toc15651726)

[5 Macros 8](#_Toc15651727)

[5.1 RunAll 8](#_Toc15651728)

[5.2 Setup 8](#_Toc15651729)

[5.3 Run\_MonteCarlo 9](#_Toc15651730)

[6 Model Outputs 10](#_Toc15651731)

Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Version** | **Version Date** | **Model Version** | **Author** | **Changes** |
| 1.0 | 25/7/19 | 1.1 | Andrew Taylor | Initial |

# Model Overview

The model was created to provide an example of how Monte Carlo analysis can be used to estimate the combined uncertainty from a number of distinct sources, taking into account the interactions between the factors.

Depending on the users needs the model may be sufficient, however in many instances the model may need to be adapted to take into account specific interactions or distribution. Information to help with this can be found in the technical guide.

Before using the model, consider whether Monte Carlo is the correct approach to use. It is useful for combining different uncertainties, but does not help understand them individually, and is highly dependent on the quality of assumptions feeding into it. For more information, see the communicating uncertainty material.

## Model Contents

|  |  |
| --- | --- |
| **Worksheet** | **Description** |
| [Controls](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#RANGE!A1) | To record information about the purpose of the analysis |
| [Register](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#Register!A1) | Contains input assumptions about the factors that drive variance in the analysis |
| [Inputs](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#Inputs!A1) | Contains parameters used in running the model |
| [Calculations](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#Calculations!A1) | Runs a single scenario based on the details in the register tab. |
| [Current Run](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#'Current Run'!A1) | Contains the outputs of the run, and any additional overlays that may be applied |
| [Outputs](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#Outputs!A1) | A full list of the results of each run |
| [Model Output](file:///C:\Users\rcq70e\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1F94F51.xlsm#'Model Output'!A1) | A summary aggregating the results into an overall distribution. |

## Model Versions

The version history is located within the model itself.

## Analytical Quality Assurance

### Developer Testing

The model is adapted from an existing process used to estimate the net potential variance due to identified risks and opportunities and so has had extensive practical testing.

### Validation & Verification

Additional QA of this specific version has been carried out by an independent analyst, see the AQA log within the model for more details.

# Inputs

## Register

This worksheet contains the specific entries that are assumed to drive the variance and the detailed of how they should be modelled.

Each record represents a distinct factor that could drive variance in the analysis. This data is manually input and the parameters should either be informed by separate analysis or through engagement with stakeholders to agree sensible assumptions.



A complete list of the default options can be found in the user guide, this section will just cover the categories. The available options are not exhaustive, for details on how to add more options, see the relevant notes in section 4.

For each entry, the details of the entry need to populated. The first few columns are self explanatory, the later ones are as follows:

Confidence Level – sets how likely an entry is to occur. The definitions of these levels can be changed or new ones added on the parameters tab.

Include? – When set to N, the entry is left out of the analysis. This can be useful is assessing the impact of a single entry.

Distribution Type – How the values of this entry should be modelled. The next columns relate to the parameters that are needed for the approach and the relevant ones must be populated

Occurrence dependent on/Type – Specifies which entry alters the chance of this one occurring and how.

Impact dependent on/Type – specifies which entry alters the chance of this one occurring and how.

Dependence through the years – by default each year is considered independently which can be changed here.

## Inputs

This tab sets up some of the parameters used in running the model.

The number of runs box represents the number of iterations the Monte Carlo is run for. The more entries that are in the register, the more runs are needed to converge to an estimate. To check whether the number is sufficient, run several times and check that there is minimal movement in the overall estimated variance.



The table below this provides a definition of the likelihoods. Only the minimum and maximum probability are used and can be changed freely. In addition, new likelihoods can be entered here.

On the right had side are lists of the various options present in the model, used to populate the dropdown lists in the register.

The run button at the bottom calls the Run\_All macro, which starts the main run.

# Assumptions

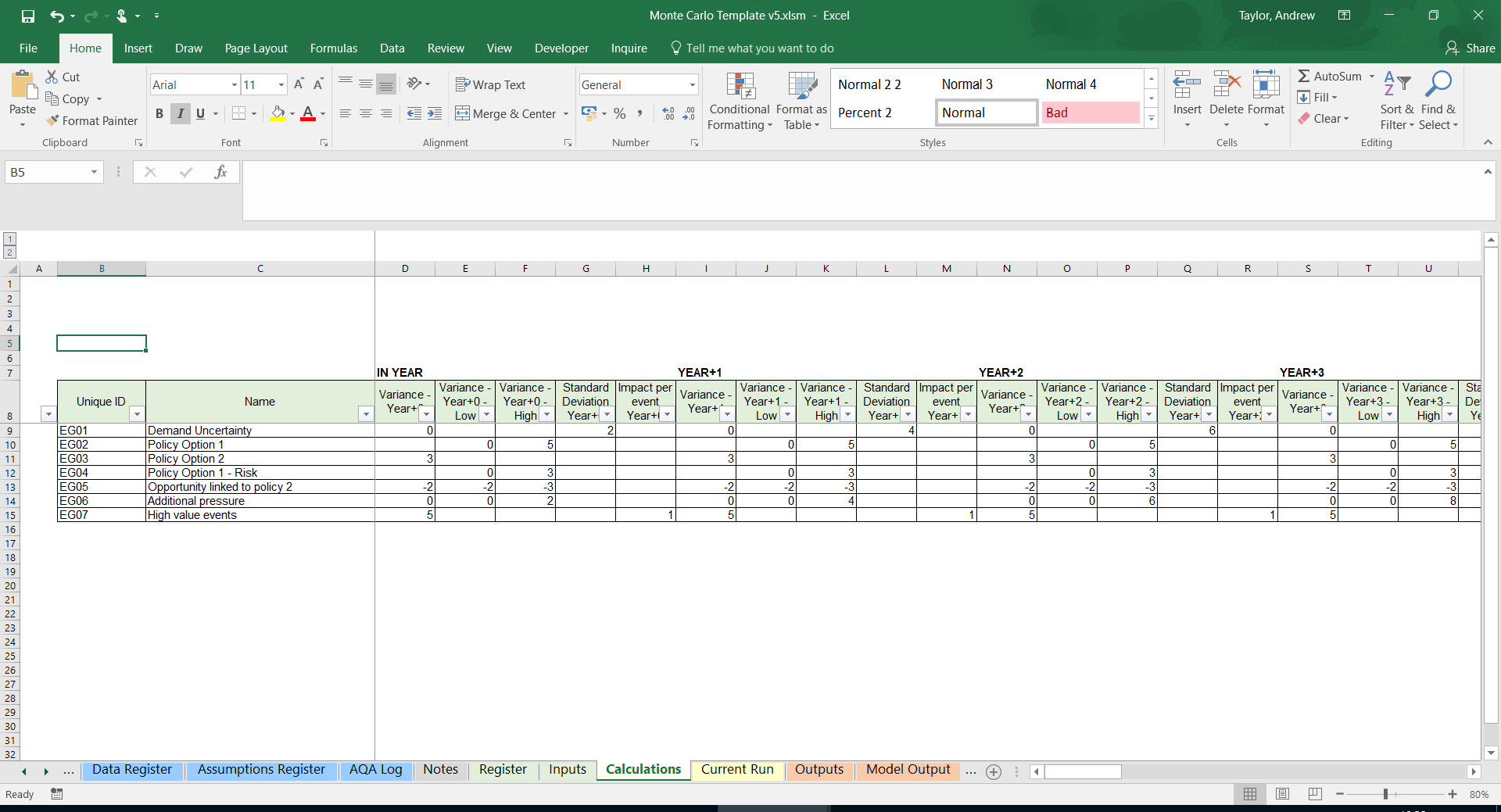
The model is heavily dependent on the accuracy of the entries entered in the register and is only as accurate as these inputs.

# Calculations

The two calculations tabs are used to generate the results of a single iteration of the model, with the macros covered in section 5 relating to the monte carlo loop.

## Calculations

This tab covers the bulk of the calculations for a single run and is where the various model parameters are used. This tab is populate by the SetUp macro after the run button has been pressed. The results in here are based on a randomised number and are change every time the workbook recalculates.



Columns B to AI are taken directly from the register tab with no additional calculations.

### Occurrence calculations

Columns AL to BH perform the calculations to test whether an entry occurs each year or not. Step 1 generates the probability the entry will occur each year by picking a value within it’s specified range. Step 2 then generates a random number of compare to this to test whether or not it occurs, with a 1 representing occurrence.

Step 3 then applies any rules relating to dependence between the years, and this is where any additional rules added later should be coded. Similarly, step 4 relates to dependency between different entries.

### Value calculations

Once the occurrence has been generated, the model calculates the actual impact. Step 5 generates the random number used in the run. Note that this is the step where both the dependence through the years and dependence between entries are taken into account.

Step 6 is broken down into a number of separate calculations. Each of these contains the formula needed to calculate the impact under a specific distribution (as separating these out allows for much simpler formula). Additional distributions can be inserted here. At the very end of step 6, the model looks at the distribution type to see which value should be pulled through.

### Output

The last two steps combine the results of the occurrence and value calulations to assess the actual impact per entry per year and the total net position for the run.

## Current Run

This tab takes the output of the run and applies any additional rules that may be needed. In the example, the numbers are uplifted in future years in line with inflation, but this step may also be useful if there is a need to combine volume and price impacts.

## Outputs

This tab simply contains a complete list of the results of all the runs as well as the mean and standard deviation they suggest.

# Macros

The workbook contains 3 macros used to run the process.

## RunAll

This is a simple macro just used to call the other macros in order to do a complete run.

Example Code

Sub RunAll()

SetUp

Run\_MonteCarlo

End Sub

## Setup

This macro loops through the entries in the register and if they are set to be included moves them into the calculations tab.

Example Code

Sub SetUp()

'Takes entries from the register and moves them to the calculations tab

Dim strUniqueID As String

Dim intMaxRow As Integer

Sheets("Calculations").Select

'Clear old entries

Range("10:6500").Delete

Range("B9").Select

'Loop through register to pick up new ones and copy to calculations sheet

Sheets("Register").Select

Range("A2").Select

Do Until IsEmpty(ActiveCell)

If Range("E" & ActiveCell.Row).Value = "Y" Then

strUniqueID = ActiveCell.Value

Sheets("Calculations").Select

ActiveCell.Value = strUniqueID

ActiveCell.Offset(1, 0).Select

Sheets("Register").Select

End If

ActiveCell.Offset(1, 0).Select

Loop

'Count how many rows are present for copying formula

Sheets("Calculations").Select

intMaxRow = ActiveCell.Row - 1

'End if no entries to avoid overwriting headings, return error and end run

If intMaxRow < 9 Then

MsgBox "Error: No entries." \_

& vbNewLine \_

& vbNewLine \_

& "Ending run."

End

End If

'Copy all formula down

Range("C9:DD9").Copy

Range("C9:DD" & intMaxRow).Select

ActiveSheet.Paste

'Format first column

Range("B9").Copy

Range("B9:B" & intMaxRow).Select

Selection.PasteSpecial Paste:=xlPasteFormats

End sub

## Run\_MonteCarlo

The final macro runs a loop for the specified number of iterations, takings the results from the current run tab and saving them as values in the Outputs tab.

Example Code

Sub Run\_MonteCarlo()

'Takes the results of a run from the outputs repeatedly to build up population

Application.ScreenUpdating = False

Dim intRunNo As Integer

'Clear old results

Sheets("Outputs").Select

Range("B7:G100000").ClearContents

Range("B7").Select

'Take result of model i times

For i = 1 To Range("P\_NoOfRuns").Value

ActiveCell.Value = i

Application.StatusBar = "Run Number: " & i

Sheets("Current Run").Range("C6:G6").Copy

ActiveCell.Offset(0, 1).PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=False

ActiveCell.Offset(1, -1).Select

Next i

End Sub

# Model Outputs

The final tab contains a summary of the potential variance, showing the net impact and the confidence intervals around it. It also contains the option to overlay the results on top of an existing central forecast.