mguchiQ

Specifying Data

Contents

- **Overview**
- **Data**
  - **Instance Variables**
  - **Single Variables**
  - **Series Variables**
  - **Table Variables**
- **Supplying Series Variables**
- **Supplying Table Variables**
- **Supplying Stochastic Variables**

Overview

The objective of this document is to explain how to specify data that will be used as input to a **Run** in mguchiQ. This document makes use of examples in the tutorials available on the mguchiQ website.

Data

In order to run our model, we will require data for all the variables we have defined in our model. Let's now have a look at how this data would need to be supplied.

Instance Variables

Instance Variables represent the actual (non-abstract) products that will be fed into the model for calculation. In our example we have two actual product types, **Life** and **Annuity**. Below is an example of 3 instances of both **Life** products and **Annuity** products. Note that the first 2 columns, **Product Number** and **Product Type** are required for all **Instance Variable** definitions so that mguchiQ can uniquely identify each product instance:

Product Number	Product Type	Age At Inception	Duration In Force	Gender	Smoker	Sum Assured	Premium
1	Life	21	240	0	1	500,000.00	125.00
2	Life	43	12	1	0	400,000.00	120.00
3	Life	61	87	0	0	200,000.00	75.00

Product Number	Product Type	Age At Inception	Duration In Force	Gender	Smoker	Annuity	Annuity Escalation
4	Annuity	55	12	0	1	3,456.00	5%
5	Annuity	57	34	0	1	8,500.00	6%
6	Annuity	67	90	0	0	6,750.00	0%

Single Variables

We have 1 **Single Variable** that is defined at the **Global** product level, namely **Expense**. We have two options in supplying this **Expense** data depending on whether it is the same value regardless of whether it relates to the **Life** product or the **Annuity** product. If this value is the same for the **Life** product and the **Annuity** product (i.e., it costs us the same amount to maintain a **Life** product as it does to maintain an **Annuity** product) then we can supply **Expense** at the **Global** product level as follows:

Product	Expense
Global	4.5

If, however, it costs us a different amount each month to maintain a **Life** product as opposed to an **Annuity** product then we would need to supply the **Expense** data separately as follows:

Product	Expense
Life	4.5
Annuity	5.3

Series Variables

For **Series Variables** it is necessary to supply a range of values ranging from **1** to **the maximum we require for the series.**

For series that are **time** related it is necessary to supply a range of values from 1 to the maximum **time period** our model may use, let us assume this is 1200 (months) for our variables that are **time** based (**Inflation**, **Yield**, and **LapseRate**)

The following **Series Variables** need to be supplied:

Product	t	Inflation
Global	1	6.1%
Global	2	6.2%
…	…	…
Global	1199	8.2%
Global	1200	8.0%

Product	t	Yield
Global	1	7.0%
Global	2	7.1%
…	…	…
Global	1199	9.2%
Global	1200	9.3%

Product	t	LapseRate
Life	1	20%
Life	2	20%
…	…	…
Life	1199	10%
Life	1200	10%

Table Variables

For **Table Variables** it is necessary to supply a range of values with the **x-index** ranging from **1** to **the maximum x value required** and a **y-index** ranging from **1** to the **maximum number of columns in our table.**

For **Table Variables** that are **age** related it is necessary to supply a range of x-values ranging from **1** to the maximum **age** our model may use, let us assume this is 101 for our **Table Variable** that is **age** based (**Qx**). We need to supply 4 columns worth of data as our **Qx** Table Variable is declared to have 4 columns - representing a combination of gender (male / female) and smoker status (smoker / non-smoker).

The following **Table Variable** needs to be supplied:

Product	Age	Qx_1	Qx_2	Qx_3	Qx_4
Global	1	0.00027	0.00025	0.00028	0.00026
Global	2	0.00028	0.00026	0.00029	0.00027
…	…	…	…	…	…
Global	100	0.31854	0.28958	0.31954	0.29058
Global	101	0.35096	0.31905	0.36096	0.32005

Supplying Series Variables

mguchiQ provides a short-hand method for filling in **Series Variables** by way of two simplistic calculation mechanisms:

Method	Description
Flat	Keeps the previous value constant until a new value is encountered
Interpolate	Linear interpolates between previous and next values

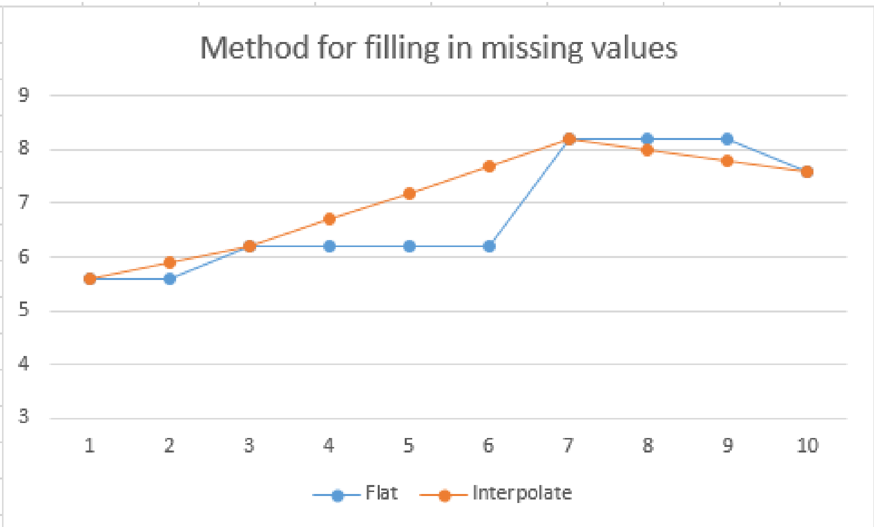
These two methods can best be described by an example. Let's assume we are given the following incomplete **series** data (where **series size** = 10):

i	Value
1	5.6%
3	6.2%
7	8.2%
10	7.6%

The result will be as follows once missing values have been calculated and inserted:

i	Flat	Interpolate
1	**5.6%**	**5.6%**
2	5.6%	5.9%
3	**6.2%**	**6.2%**
4	6.2%	6.7%
5	6.2%	7.2%
6	6.2%	7.7%
7	**8.2%**	**8.2%**
8	8.2%	8.0%
9	8.2%	7.8%
10	**7.6%**	**7.6%**

Graphically this looks as follows:



Please note the following:

- If the first supplied **series value** is for **i** > **1**, then the first supplied **series value** is used for all values of **i** from **1** until the first supplied **i** value, regardless of the calculation mechanism chosen.
- If the last supplied **series value** is for **i** < **series size**, then the last supplied **series value** is used for all values of **i** from the last supplied **i** value until **series size**, regardless of the calculation mechanism chosen.

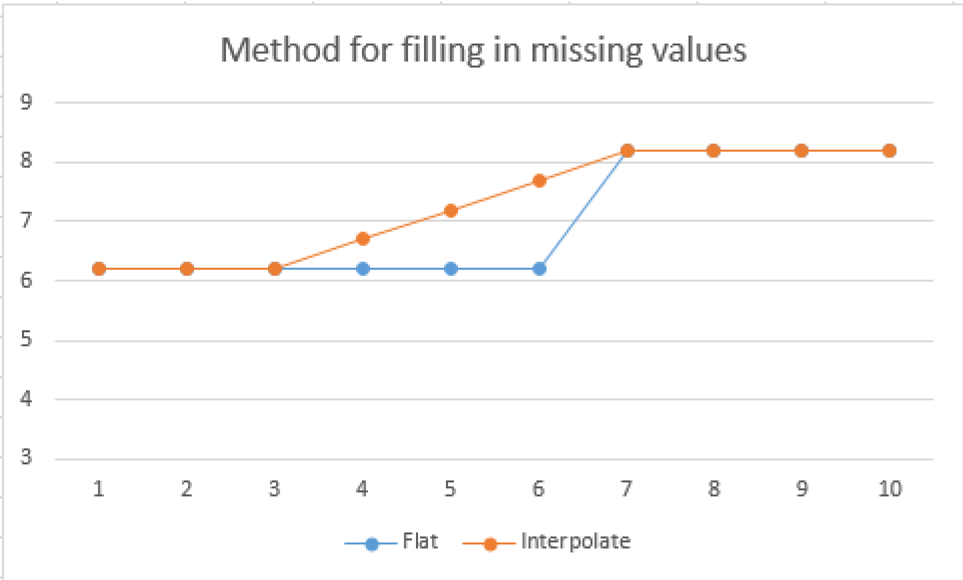
This can best be illustrated by way of an example. Let's assume we are given the following **series** data (where **series size** = 10):

i	Value
3	6.2%
7	8.2%

The result will be as follows once missing values have been calculated:

i	Flat	Interpolate
1	6.2%	6.2%
2	6.2%	6.2%
3	**6.2%**	**6.2%**
4	6.2%	6.7%
5	6.2%	7.2%
6	6.2%	7.7%
7	**8.2%**	**8.2%**
8	8.2%	8.2%
9	8.2%	8.2%
10	8.2%	8.2%

Graphically this looks as follows:



Supplying Table Variables

The same shorthand mechanism for supplying **Series Variables** can also be used to supply **Table Variables**, with each column of the table acting as an independent series.

Supplying Stochastic Variables

In general, stochastic variables are usually generated by a system that takes many, possibly interrelated, factors into account to arrive at a stochastic distribution that could be input into mguchiQ for stochastic model calculation purposes. In is not the intention for mguchiQ, at this stage, to be a complex stochastic distribution calculation engine. Thus, stochastic inputs to mguchiQ would be specified as a series of values, the size of which depends on the number of stochastic permutations required. As an example, we will use a **return curve**, representing the expected annual return on an investment for the next 5 years as follows:

t	Return
1	5%
2	6%
3	7%
4	6%
5	5%

A stochastic representation of this curve, representing 10,000 permutations, or **scenarios**, may look something like this:

t	Scenario 1	Scenario 2	Scenario 3	…	Scenario 9,999	Scenario 10,000
1	5%	4.67%	7.09%	…	5.23%	6.08%
2	6%	6.32%	7.55%	…	6.03%	6.35%
3	7%	5.48%	6.77%	…	6.37%	6.54%
4	6%	7.05%	3.78%	…	5.58%	6.56%
5	5%	4.84%	4.55%	…	4.42%	3.99%

Although mguchiQ is not currently focussed on being a complex stochastic distribution calculation engine, in the simplified case where the variables that are to be represented stochastically are not interrelated, and their stochastic distributions can be modelled by a **normal distribution**, mguchiQ will calculate a set of stochastic values given a **mean** and **standard deviation** for each variable that is to be modelled stochastically, as below:

t	Return_Mean	Return_StandardDeviation
1	5%	1%
2	6%	1%
3	7%	1%
4	6%	1%
5	5%	1%

A **seed** can also be provided to make sure the same set of random values are generated in subsequent runs.