mguchiQ

Model Building Tutorial 1

A simple Life Insurance Product

Contents

- Overview
- Model Definition
- Spreadsheet Structure
    - Cell and Range Names
    - Model Sheet
    - Products Sheet
    - Instance Variables Sheet
    - Single Variables Sheet
    - Series Variables Sheet
    - Table Variables Sheet
    - Functions Sheet
- Things to look out for
- Appendices
    - Appendix A: Excel Functions

Overview

The objective of this document is to explain how to build a model by taking a working Excel spreadsheet and submitting it to mguchiQ.

It is best to read the document "**020.mguchiQ – Concepts**" first as this gives an explanation of the components and concepts that make up mguchiQ. There is also a Quick Reference Guide at the back of this document for ease of reference.

Building a model is the first step in being able to use the functionality of mguchiQ. Once a model is built it can then be run against a set of data to produce results.

This tutorial describes building a model for a single product only and is the best starting point for understanding mguchiQ.

Later tutorials will introduce more features of mguchiQ including the building of a model with multiple products.

Model Definition

You build a model by specifying the data and calculations that define each product within your model in a predefined format in an Excel spreadsheet. This spreadsheet can then be submitted to mguchiQ which will verify your model and, if correct, build it into an executable piece of code.

If order to demonstrate how to build a model in Excel for submission to mguchiQ, we will use the example of a simple **Life Assurance** product that we will call **Life**. This product will have the following characteristics:

- The **Premium** and S**um Assured** will stay constant over the life of the policy.
- The **Expenses** relating to maintaining the policy will increase according to **Inflation**.

The aim of our model will be to calculate the **Best Estimate Liability (BEL)** of our **Life** product, defined as the present value of all future cash flows. The net cash flow for each future period **t** will be defined as:

NetCashflow(t) **=** **PremiumReceived**(t) – **ExpensesPaid**(t) – **DeathBenefitsPaid**(t)

We will require the following **Instance Variables** for **each policy** - this is information that needs to be supplied for each policy:

Name	Description
AgeAtInception	Policy holder age at inception, in years
DurationInforce	Number of months the policy is in force
Gender	Whether the policy holder is male or female
Premium	The monthly premium payable by the policy holder
SumAssured	How much the policy holder's life is insured for

Next, we require the following **Non-Instance Variables** - this is information that will be **shared by all policies** – there are 3 types of **Non-Instance Variables** that mguchiQ recognises for information shared by all policies:

- **Single Variables** – these are singular (as opposed to array) variables. In our example we will have the following **single variable**:

Name	Description
Expense	The monthly expense incurred to maintain each policy

- **Series Variables** – these are variables made up of a series of values, essentially a one-dimensional array. In our example we will have the following **series variables:**

Name	Description	Index By
Inflation	Forecast inflation rates, for each future time period	Time
Yield	Forecast yield curve, to be used to present value future cash flows	Time
LapseRate	Forecast lapse rates	Policy Age

- **Table Variables** – these are variables made up of a table of values, essentially a two-dimensional array. In our example we will have the following **tables variable:**

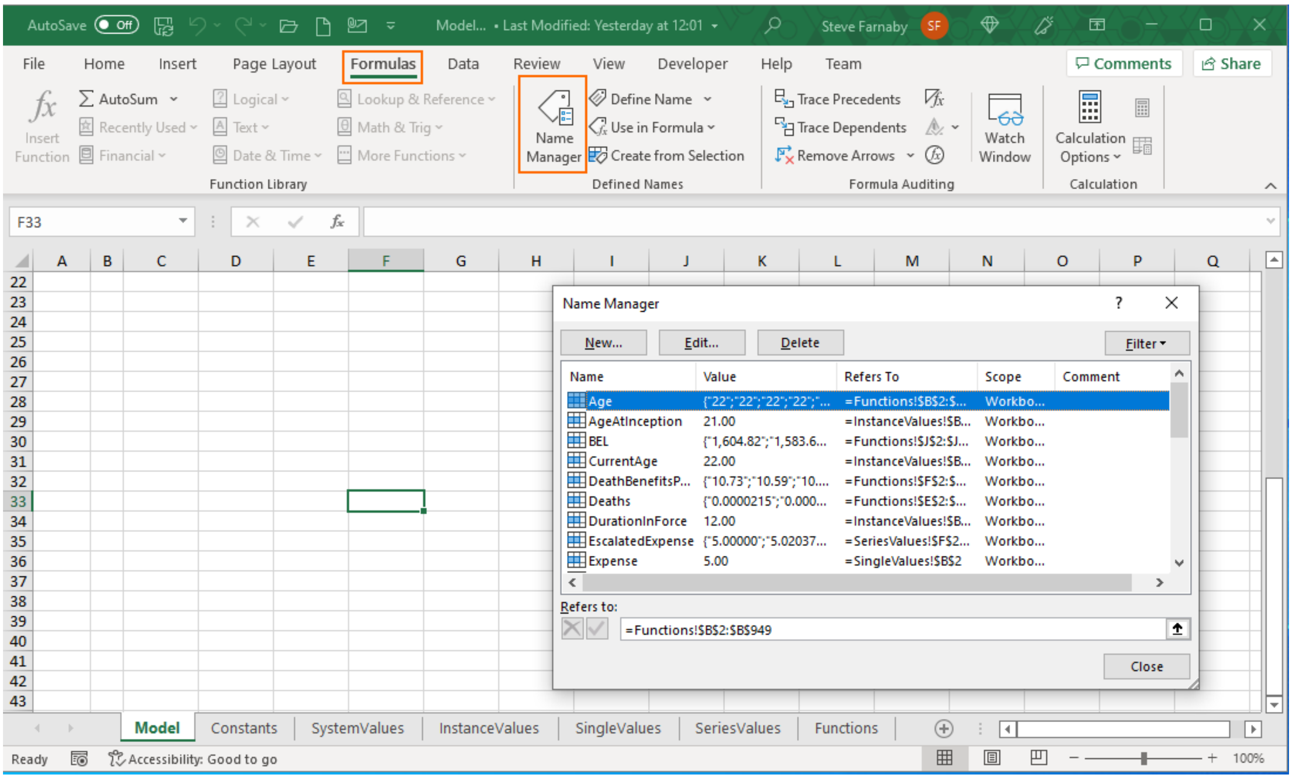
Name	Description	Index By
Qx	Life table, giving the probability of survival for a given policy holder age. We will use a table with 2 columns – the columns representing the policy holders gender.	Policy Holder Age

Spreadsheet Structure

In order to submit a model specification to mguchiQ the spreadsheet needs to conform to a predefined structure.

Cell and Range Names

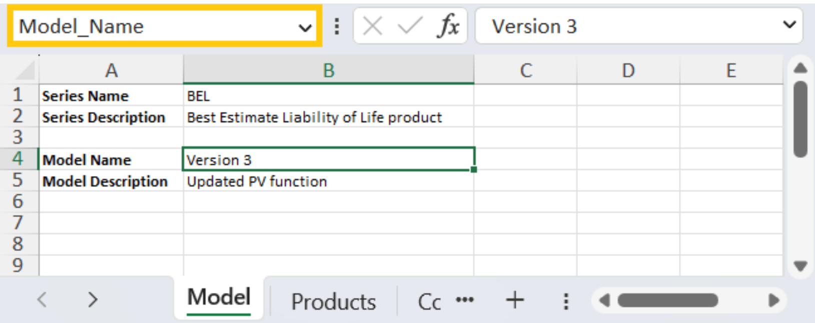
Instead of referencing cells or ranges directly in Excel, a model spreadsheet for mguchiQ makes use of **names**. Please familiarise yourself with naming cells and ranges in Excel before continuing. The **Name Manager** in Excel is a good starting place:



Name	Value	Refers To	Scope	Comment
Age	{"22";"22";"22";"22";"...	=Functions!$B$2:$...	Workbo...	
AgeAtInception	21.00	=InstanceValues!$B...	Workbo...	
BEL	{"1,604.82";"1,583.6...	=Functions!$J$2:$J...	Workbo...	
CurrentAge	22.00	=InstanceValues!$B...	Workbo...	
DeathBenefitsP...	{"10.73";"10.59";"10....	=Functions!$F$2:$...	Workbo...	
Deaths	{"0.0000215";"0.000...	=Functions!$E$2:$...	Workbo...	
DurationInForce	12.00	=InstanceValues!$B...	Workbo...	
EscalatedExpense	{"5.00000";"5.02037...	=SeriesValues!$F$2...	Workbo...	
Expense	5.00	=SingleValues!$B$2	Workbo...	

Refers to: =Functions!$B$2:$B$949

Cell or range names are generally visible in a spreadsheet as follows when the cell or range is selected:



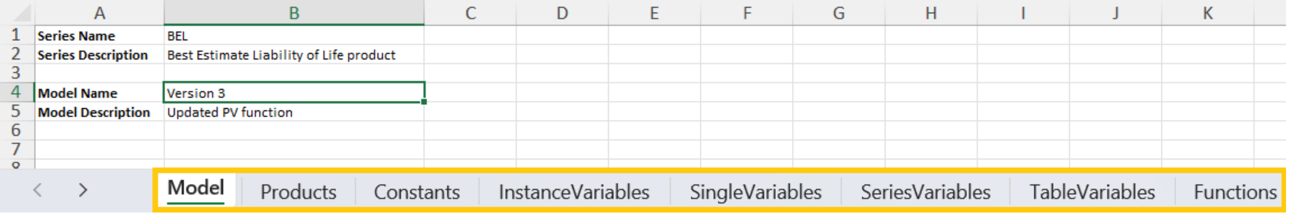
	A	B	C	D	E
1	Series Name	BEL			
2	Series Description	Best Estimate Liability of Life product			
3					
4	Model Name	Version 3			
5	Model Description	Updated PV function			
6					
7					
8					
9					

Note that in the spreadsheet that we use to build a model, the actual numerical values in the spreadsheet are not important. What is important is the name of the ranges you specify and any formulae that calculate results. mguchiQ uses these names and formula to build a model, not the numerical values in any cell.

A basic model spreadsheet generally has the following sheets defined:

Sheet	**Description**
Model	Identifies the Model Series this Model belongs to and describes this Model itself.
Products	Specify the products of the model.
InstanceVariables	Instance Variables are variables associated with each individual Product item.
SingleVariables	Single Variables are variables needed for calculations but are not associated with a Product.
SeriesVariables	Series Variables are one-dimensional array variables (defined as a series from 1 to n) needed for calculations.
TableVariables	Table Variables are two-dimensional array variables (defined as a table of rows and columns) needed for calculations.
Functions	Functions are Formulae that describe the characteristics of Products.

Below is a typical Excel spreadsheet with the required sheets defined:



	A	B
1	**Series Name**	BEL
2	**Series Description**	Best Estimate Liability of Life product
3		
4	**Model Name**	Version 3
5	**Model Description**	Updated PV function
6		
7		

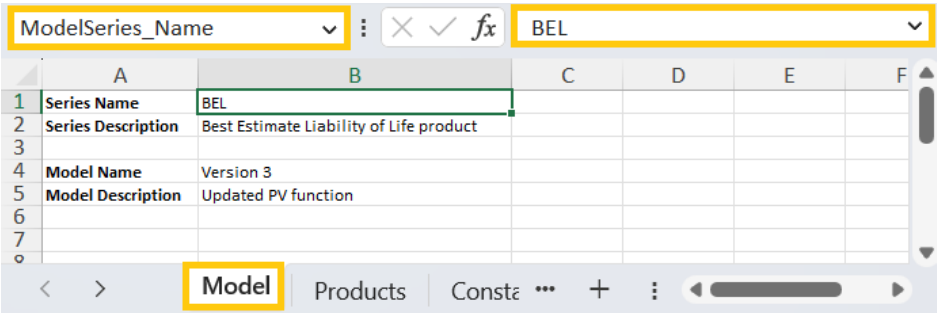
Model Sheet

The **Model** sheet describes which **Model Series** this **Model** belongs as well as the **Model** itself.

Model Series Details

A **Model Series** is typically used to group together a number of **Model** versions.

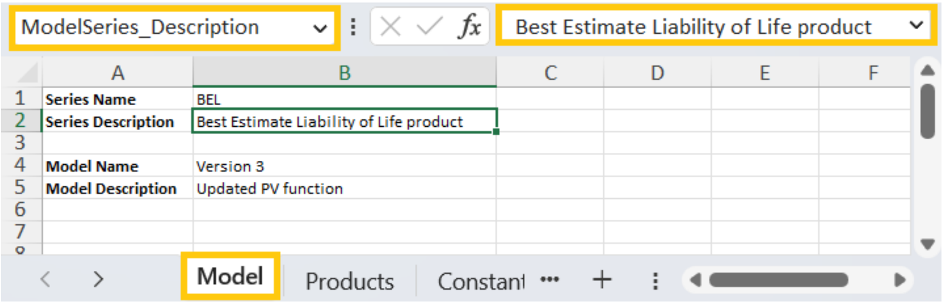
The Model Series **Name** is specified as follows:



	A	B	C	D	E	F
1	**Series Name**	BEL				
2	**Series Description**	Best Estimate Liability of Life product				
3						
4	**Model Name**	Version 3				
5	**Model Description**	Updated PV function				
6						
7						

If the particular Model Series **Name** does not yet exist, it will be created.

The Model Series **Description** is specified as follows:



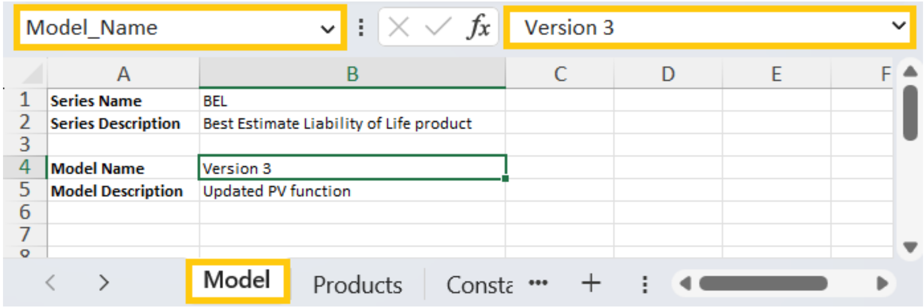
	A	B	C	D	E	F
1	**Series Name**	BEL				
2	**Series Description**	Best Estimate Liability of Life product				
3						
4	**Model Name**	Version 3				
5	**Model Description**	Updated PV function				
6						
7						

Note that if the Model Series already exists, the **Description** can be left blank or the **Description** must be the same as the existing Model Series **Description**. If the Model Series does not yet exist, then a **Description** must be supplied.

Model Details

A Model requires a **Name** and a **Description** in order to identify it.

The Model **Name** is specified as follows:



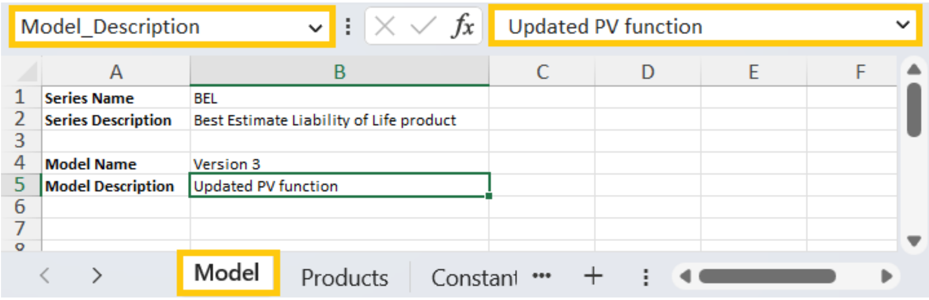
Model_Name

Version 3

	A	B	C	D	E	F
1	**Series Name**	BEL				
2	**Series Description**	Best Estimate Liability of Life product				
3						
4	**Model Name**	Version 3				
5	**Model Description**	Updated PV function				
6						
7						

Note that the actual unique Model Name is generated as a combination of the **Model Series Name** and the **Model Name** separated by a full stop, in this example the unique Model Name will be **BEL.Version 3**

The Model **Description** is specified as follows:



Model_Description

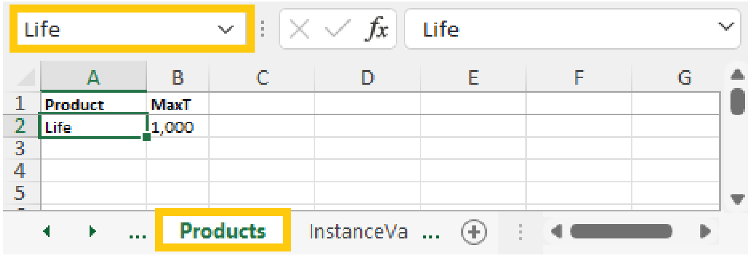
Updated PV function

	A	B	C	D	E	F
1	**Series Name**	BEL				
2	**Series Description**	Best Estimate Liability of Life product				
3						
4	**Model Name**	Version 3				
5	**Model Description**	Updated PV function				
6						
7						

Products Sheet

Declare the **products** of your model here. In this tutorial we will only have one product, called **Life**, representing a life assurance product.

First, declare the product by simply naming a cell to the name of your product:

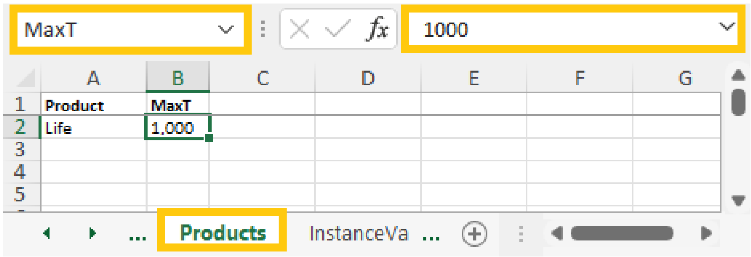


Life

Life

Product	MaxT
Life	1,000

Next, it is necessary to give a specification for the **MaxT** of your product as below. For now we will just use a value of 1,000 which represents 1,000 months into the future.



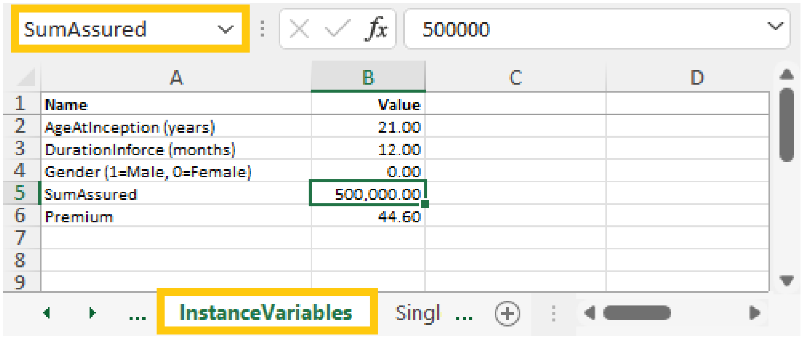
MaxT

1000

Product	MaxT
Life	1,000

Instance Variables Sheet

Instance Variables are entered onto a sheet called **InstanceVariables**, with each **Instance Variable** being named accordingly:



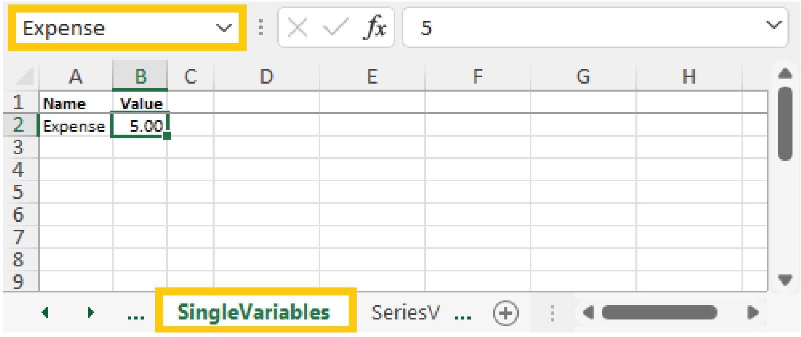
SumAssured

500000

Name	Value
AgeAtInception (years)	21.00
DurationInforce (months)	12.00
Gender (1=Male, 0=Female)	0.00
SumAssured	500,000.00
Premium	44.60

Single Variables Sheet

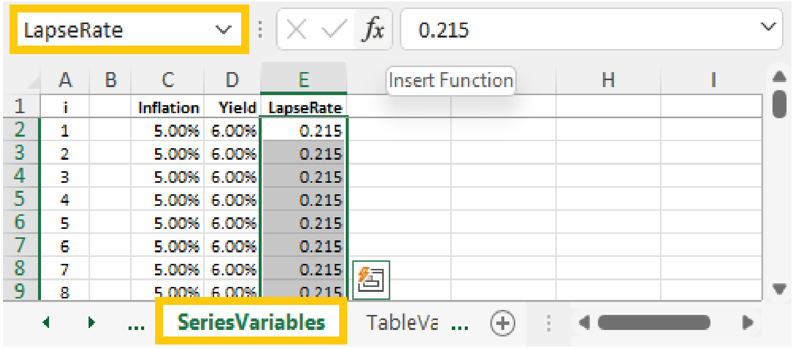
Single Variables are entered onto a sheet called **SingleVariables**, with each **Single Variables** being named accordingly:



Name	Value
Expense	5.00

Series Variables Sheet

Series Variables are entered onto a sheet called **SeriesVariables**, with each **Series Variable** being named accordingly. Note that the entire range of cells that makes up the **Series** must be named. The first column (column A) of the **SeriesVariables** sheet must contain the index for all **Series**:

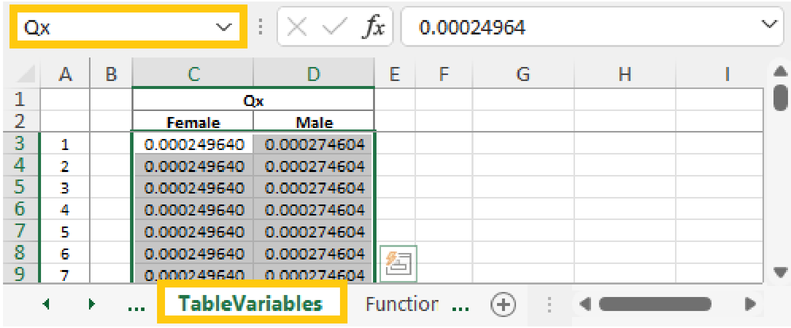


i		Inflation	Yield	LapseRate
1		5.00%	6.00%	0.215
2		5.00%	6.00%	0.215
3		5.00%	6.00%	0.215
4		5.00%	6.00%	0.215
5		5.00%	6.00%	0.215
6		5.00%	6.00%	0.215
7		5.00%	6.00%	0.215
8		5.00%	6.00%	0.215

Note that the number of rows of the a **Series Variable** range must be representative of the maximum number of rows required for all permutations of data input. So, for example, if our **maxT** (explained in a later tutorial) is 1000 and the **Series Variable** is indexed by **t**, then the **Series Variable** range must have at least 1000 rows. mguchiQ uses the number of rows of a **Series Variable** for internal storage sizing purposes.

Table Variables Sheet

Table Variables are entered onto a sheet called **TableVariables**, with each **Table Variable** being named accordingly. Note that the entire range of cells that makes up the **Table** must be named. The first column (column A) of the **TableVariables** sheet must contain the x-index for all **Tables**:



Qx

0.00024964

	A	B	C	D
1			Qx	
2			Female	Male
3	1		0.000249640	0.000274604
4	2		0.000249640	0.000274604
5	3		0.000249640	0.000274604
6	4		0.000249640	0.000274604
7	5		0.000249640	0.000274604
8	6		0.000249640	0.000274604
9	7		0.000249640	0.000274604

Note that the number of rows of the a **Table Variable** range must be representative of the maximum number of rows required for all permutations of data input. So, for example, if a **Table Variable** is indexed by **Age**, then the **Table Variable** range must have at least as many rows as the maximum **Age** we envisage in our calculations, possibly 100. mguchiQ uses the number of rows of a **Table Variable** for internal storage sizing purposes.

Functions Sheet

Now that we have all our data requirements specified, we are going to build a model that calculates **Best Estimate Liability (BEL)**, defined as the present value of all future cash flows.

The logic that performs the calculation of **BEL** for our **Life** product is defined as a set of **functions** as follows:

Function	Formula
Age(t)	FLOOR(AgeAtInception+((DurationInforce+t)/12))
Deaths(t)	PolicyHoldersInforce(t) * (1-(1-Qx[Age(t),Gender+1])^(1/12)))
Lapses(t)	(PolicyHoldersInforce(t) - Deaths(t)) * (1-((1-LapseRate[DurationInforce+t])^(1/12)))
PolicyHoldersInforce(t)	IF(t=1, 1, PolicyHoldersInforce(t-1) - Deaths(t-1) - Lapses(t-1))
PremiumsReceived(t)	PolicyHoldersInforce(t) * Premium
DeathBenefitsPaid(t)	Deaths(t) * SumAssured
ExpensesPaid(t)	PolicyHoldersInforce(t) * (Expense * IF(t=1,1,(1+Inflation(t-1))^((t-1)/12)))
NetCF(t)	PremiumsReceived(t) - ExpensesPaid(t) - DeathBenefitsPaid(t)
BEL(t)	IF(t=MaxT, NetCF(t), NetCF(t) + (BEL(t+1) * (1/((1+Yield(t))^(1/12)))

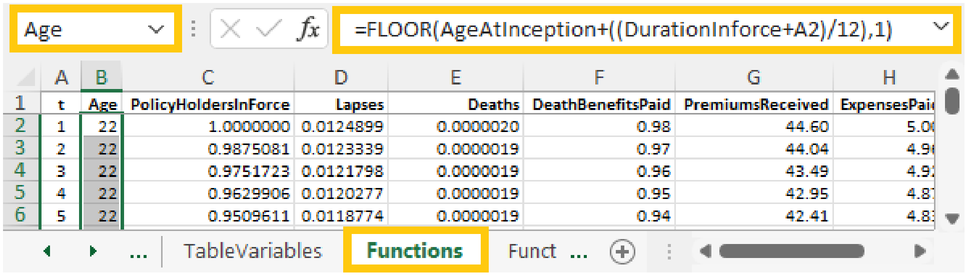
The following will help explain the functions and formulae above:

- We are assuming our time intervals are in months, i.e., **t** represents the number of months into the future from today.
- **t** ranges from **1** to **MaxT**.
- All functions take one parameter, **t**, the time period described above. So, for example, the function **Age(1)** represents the age of the policy holder in month 1, whereas **Age(360)** represents the age of the policy holder in month 360. When referencing other **functions** in our Excel functions we use the **INDEX** function instead of **()**, for example **INDEX(Age,A2)** instead of **Age(t).**
- Each formula can be called for a range of values of **t** ranging from **1** to **MaxT**.
- **Instance Variables** and **Single Variables** are referenced purely by their names, as, for example, **AgeAtInception** and **DurationInforce** are referenced in the formulae above.
- **Series Variables** are referenced by one index in square brackets **[i]**. So, a value of **Inflation[1]** will reference the inflation rate for the first month whereas a value of **Yield[120]** will reference the yield for the 120th month. When referencing **Series Variables** in our Excel functions we use the **INDEX** function instead of **[]**, for example **INDEX(EscalatedExpense,t)** instead of **EscalatedExpense[t]**.
- **Table Variables** are referenced by two indices in square brackets **[i,j]**. A value of **Qx[63,1]** will reference the probability of survival for a female policy holder of age 63. When referencing **Table Variables** in our Excel functions we use the **INDEX** function instead of **[]**, for example **INDEX(QxFactor,Age(t),QxIndex)** instead of **QxFactor[Age(t),QxIndex]**.
- There are 3 system, or built in, functions used in the formulae above: **IF**, **FLOOR**, and **POWER**. A complete list of system functions is provided in **Appendix A**.

Note that **Functions** can reference all **Instance Variables**, **Table Variables**, **Series Variables**, and **Single Variables** in their formulae.

Functions are entered onto a sheet called **Functions**, with each **function** being named accordingly. Note that the entire range of cells that makes up the **function** must be named. The first column (column A) of the **Functions** sheet must contain the time period (t):

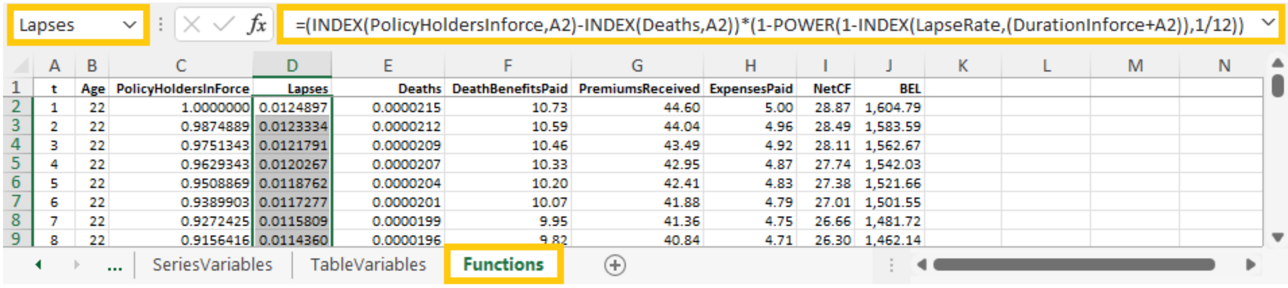
A typical function is defined as follows (function **Age**):



Age | =FLOOR(AgeAtInception+((DurationInforce+A2)/12),1)

	A	B	C	D	E	F	G	H
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPai
2	1	22	1.0000000	0.0124899	0.0000020	0.98	44.60	5.0
3	2	22	0.9875081	0.0123339	0.0000019	0.97	44.04	4.9
4	3	22	0.9751723	0.0121798	0.0000019	0.96	43.49	4.9
5	4	22	0.9629906	0.0120277	0.0000019	0.95	42.95	4.8
6	5	22	0.9509611	0.0118774	0.0000019	0.94	42.41	4.8

A typical **Function** referencing another **Function** and / or a **Series Variable** is defined as follows:

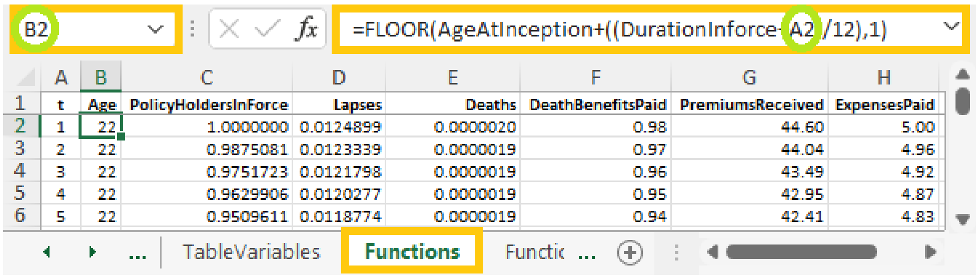


Lapses | =(INDEX(PolicyHoldersInforce,A2)-INDEX(Deaths,A2))*(1-POWER(1-INDEX(LapseRate,(DurationInforce+A2)),1/12))

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPaid	NetCF	BEL				
2	1	22	1.0000000	0.0124897	0.0000215	10.73	44.60	5.00	28.87	1,604.79				
3	2	22	0.9874889	0.0123334	0.0000212	10.59	44.04	4.96	28.49	1,583.59				
4	3	22	0.9751343	0.0121791	0.0000209	10.46	43.49	4.92	28.11	1,562.67				
5	4	22	0.9629343	0.0120267	0.0000207	10.33	42.95	4.87	27.74	1,542.03				
6	5	22	0.9508869	0.0118762	0.0000204	10.20	42.41	4.83	27.38	1,521.66				
7	6	22	0.9389903	0.0117277	0.0000201	10.07	41.88	4.79	27.01	1,501.55				
8	7	22	0.9272425	0.0115809	0.0000199	9.95	41.36	4.75	26.66	1,481.72				
9	8	22	0.9156416	0.0114360	0.0000196	9.82	40.84	4.71	26.30	1,462.14				

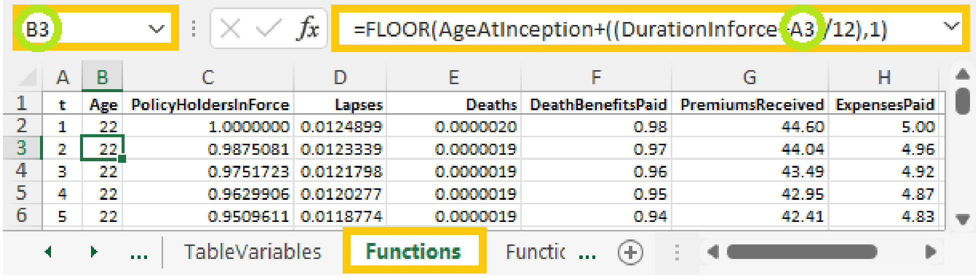
Referencing other **Functions**, **Series Variables**, or **Table Variables** must always be done via the **INDEX** function using the name of the referenced **Function**, **Series Variable**, or **Table Variable** and must not be referenced directly.

All formulae in the **Function** range must have a similar formula, the only thing that must change between formulae on different rows is the reference to column A, as below:



B2 | =FLOOR(AgeAtInception+((DurationInforce+A2)/12),1)

	A	B	C	D	E	F	G	H
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPaid
2	1	22	1.0000000	0.0124899	0.0000020	0.98	44.60	5.00
3	2	22	0.9875081	0.0123339	0.0000019	0.97	44.04	4.96
4	3	22	0.9751723	0.0121798	0.0000019	0.96	43.49	4.92
5	4	22	0.9629906	0.0120277	0.0000019	0.95	42.95	4.87
6	5	22	0.9509611	0.0118774	0.0000019	0.94	42.41	4.83

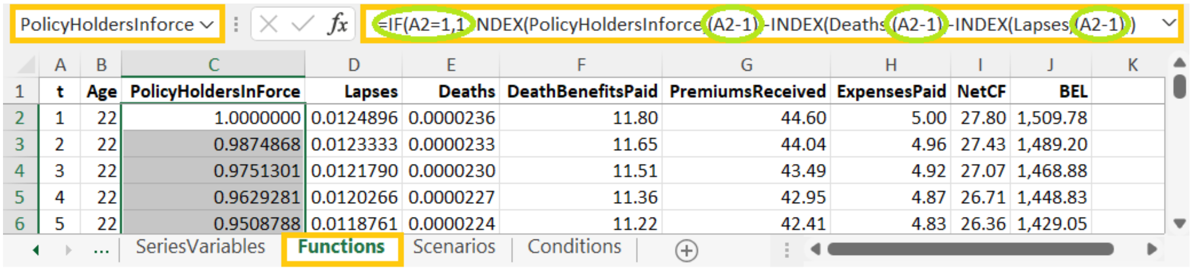


B3 | =FLOOR(AgeAtInception+((DurationInforce+A3)/12),1)

	A	B	C	D	E	F	G	H
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPaid
2	1	22	1.0000000	0.0124899	0.0000020	0.98	44.60	5.00
3	2	22	0.9875081	0.0123339	0.0000019	0.97	44.04	4.96
4	3	22	0.9751723	0.0121798	0.0000019	0.96	43.49	4.92
5	4	22	0.9629906	0.0120277	0.0000019	0.95	42.95	4.87
6	5	22	0.9509611	0.0118774	0.0000019	0.94	42.41	4.83

If a formula references a **Function** indexed by **t-1**, then it must have a conditional component that checks for **t=1**, otherwise it will try and reference a value of **t** equal to **0** at some point which is beyond the range of **1** to **MaxT**. This is evidenced by the function **PolicyHoldersInforce**:

IF(**t=1**, 1, PolicyHoldersInforce(**t-1**) - Deaths(**t-1**) - Lapses(**t-1**))



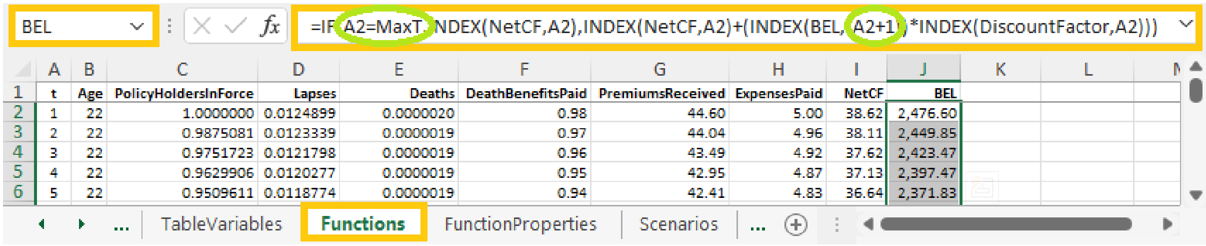
PolicyHoldersInforce

=IF(A2=1,1 NDEX(PolicyHoldersInforce (A2-1) -INDEX(Deaths (A2-1) -INDEX(Lapses (A2-1))

	A	B	C	D	E	F	G	H	I	J	K
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPaid	NetCF	BEL	
2	1	22	1.0000000	0.0124896	0.0000236	11.80	44.60	5.00	27.80	1,509.78	
3	2	22	0.9874868	0.0123333	0.0000233	11.65	44.04	4.96	27.43	1,489.20	
4	3	22	0.9751301	0.0121790	0.0000230	11.51	43.49	4.92	27.07	1,468.88	
5	4	22	0.9629281	0.0120266	0.0000227	11.36	42.95	4.87	26.71	1,448.83	
6	5	22	0.9508788	0.0118761	0.0000224	11.22	42.41	4.83	26.36	1,429.05	

If a formula references a **Function** indexed by **t+1**, then it must have a conditional component that checks for **t=MaxT**, otherwise it will try and reference a value of **t** equal to **MaxT+1** at some point which is beyond the range of **1** to **MaxT**. This is evidenced by the function **BEL**:

IF(**t=MaxT,** NetCF(t), NetCF(t) + (BEL(**t+1**) * DiscountFactor [t])))



BEL

=IF A2=MaxT NDEX(NetCF,A2),INDEX(NetCF,A2)+(INDEX(BEL, A2+1 *INDEX(DiscountFactor,A2)))

	A	B	C	D	E	F	G	H	I	J	K	L
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPaid	NetCF	BEL		
2	1	22	1.0000000	0.0124899	0.0000020	0.98	44.60	5.00	38.62	2,476.60		
3	2	22	0.9875081	0.0123339	0.0000019	0.97	44.04	4.96	38.11	2,449.85		
4	3	22	0.9751723	0.0121798	0.0000019	0.96	43.49	4.92	37.62	2,423.47		
5	4	22	0.9629906	0.0120277	0.0000019	0.95	42.95	4.87	37.13	2,397.47		
6	5	22	0.9509611	0.0118774	0.0000019	0.94	42.41	4.83	36.64	2,371.83		

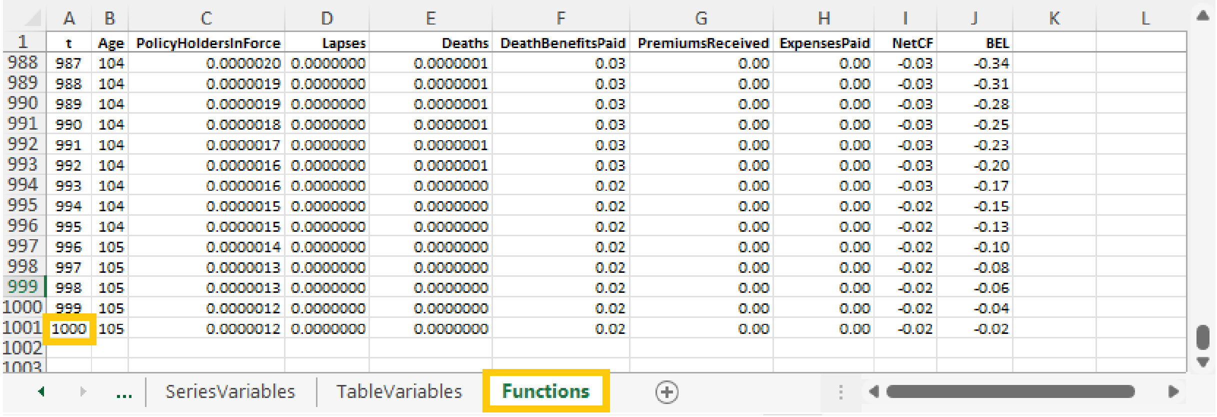
Note: in all the formulae mentioned above the only **direct** reference to a cells is to cells in column **A** – the time period. All other cell references must be by their **name**.

Things to look out for

Before submitting your model spreadsheet to mguchiQ look out for the following:

Cell errors

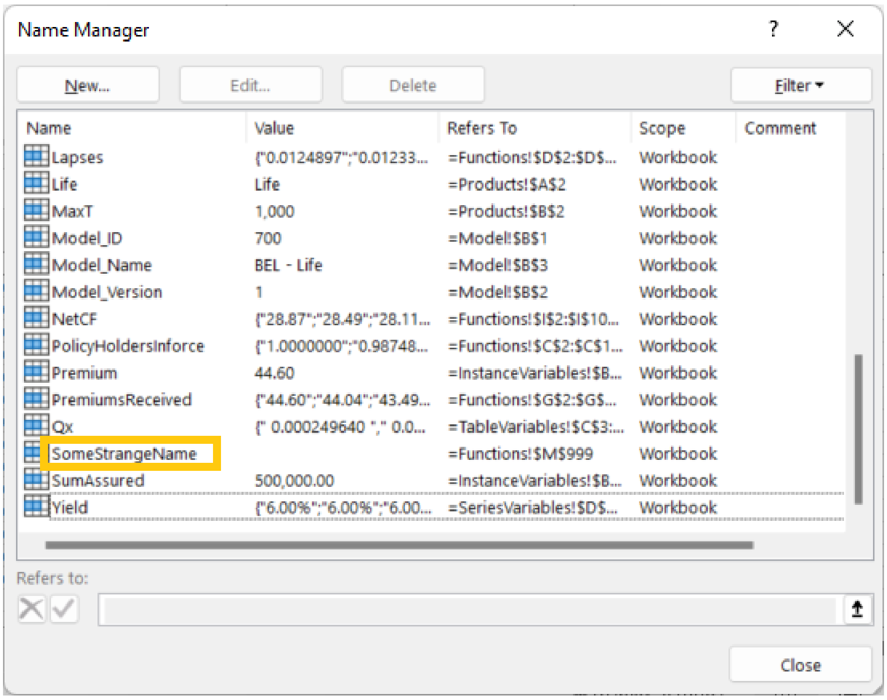
Check that your model spreadsheet has no errors all the way down to MaxT:



	A	B	C	D	E	F	G	H	I	J	K	L
1	t	Age	PolicyHoldersInForce	Lapses	Deaths	DeathBenefitsPaid	PremiumsReceived	ExpensesPaid	NetCF	BEL		
988	987	104	0.0000020	0.0000000	0.0000001	0.03	0.00	0.00	-0.03	-0.34		
989	988	104	0.0000019	0.0000000	0.0000001	0.03	0.00	0.00	-0.03	-0.31		
990	989	104	0.0000019	0.0000000	0.0000001	0.03	0.00	0.00	-0.03	-0.28		
991	990	104	0.0000018	0.0000000	0.0000001	0.03	0.00	0.00	-0.03	-0.25		
992	991	104	0.0000017	0.0000000	0.0000001	0.03	0.00	0.00	-0.03	-0.23		
993	992	104	0.0000016	0.0000000	0.0000001	0.03	0.00	0.00	-0.03	-0.20		
994	993	104	0.0000016	0.0000000	0.0000000	0.02	0.00	0.00	-0.03	-0.17		
995	994	104	0.0000015	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.15		
996	995	104	0.0000015	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.13		
997	996	105	0.0000014	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.10		
998	997	105	0.0000013	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.08		
999	998	105	0.0000013	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.06		
1000	999	105	0.0000012	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.04		
1001	1000	105	0.0000012	0.0000000	0.0000000	0.02	0.00	0.00	-0.02	-0.02		
1002												
1003												

Unnecessary Names

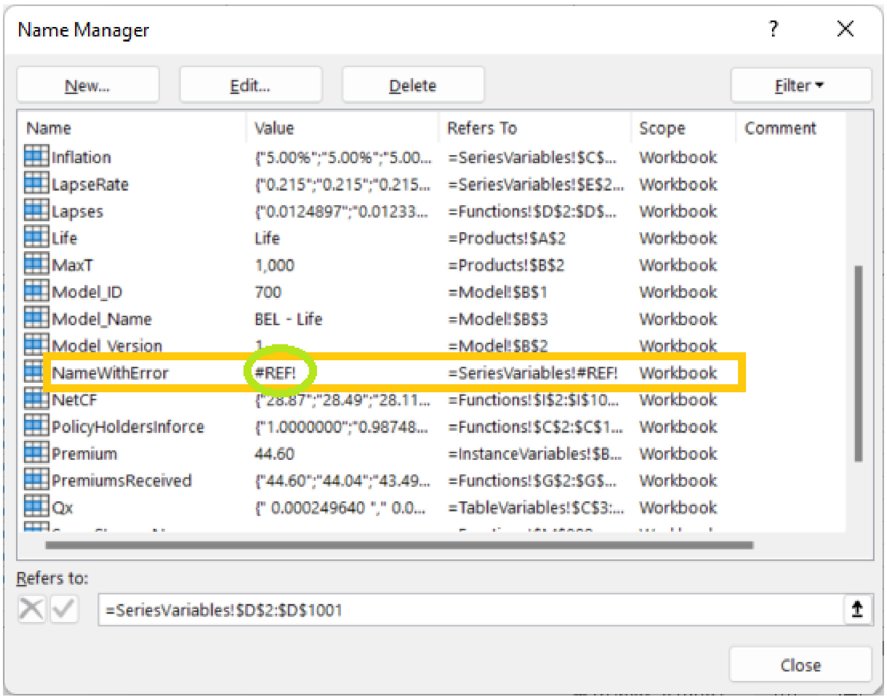
Check that only the variables and functions required by your model are present in the Name Manager. There should be no names present that are not part of your model:



Name	Value	Refers To	Scope	Comment
Lapses	{"0.0124897";"0.01233...	=Functions!$D$2:D...	Workbook	
Life	Life	=Products!$A$2	Workbook	
MaxT	1,000	=Products!$B$2	Workbook	
Model_ID	700	=Model!$B$1	Workbook	
Model_Name	BEL - Life	=Model!$B$3	Workbook	
Model_Version	1	=Model!$B$2	Workbook	
NetCF	{"28.87";"28.49";"28.11...	=Functions!$I$2:$I$10...	Workbook	
PolicyHoldersInforce	{"1.0000000";"0.98748...	=Functions!$C$2:$C$1...	Workbook	
Premium	44.60	=InstanceVariables!$B...	Workbook	
PremiumsReceived	{"44.60";"44.04";"43.49...	=Functions!$G$2:G...	Workbook	
Qx	{" 0.000249640 "," 0.0...	=TableVariables!$C$3:...	Workbook	
SomeStrangeName		=Functions!$M$999	Workbook	
SumAssured	500,000.00	=InstanceVariables!$B...	Workbook	
Yield	{"6.00%";"6.00%";"6.00...	=SeriesVariables!D...	Workbook	

Names with errors

Make sure there are no Names with errors:



Name	Value	Refers To	Scope	Comment
Inflation	{"5.00%";"5.00%";"5.00...	=SeriesVariables!C...	Workbook	
LapseRate	{"0.215";"0.215";"0.215...	=SeriesVariables!$E$2...	Workbook	
Lapses	{"0.0124897";"0.01233...	=Functions!$D$2:D...	Workbook	
Life	Life	=Products!$A$2	Workbook	
MaxT	1,000	=Products!$B$2	Workbook	
Model_ID	700	=Model!$B$1	Workbook	
Model_Name	BEL - Life	=Model!$B$3	Workbook	
Model_Version	1	=Model!$B$2	Workbook	
NameWithError	#REF!	=SeriesVariables!#REF!	Workbook	
NetCF	{"28.87";"28.49";"28.11...	=Functions!$I$2:$I$10...	Workbook	
PolicyHoldersInforce	{"1.0000000";"0.98748...	=Functions!$C$2:$C$1...	Workbook	
Premium	44.60	=InstanceVariables!$B...	Workbook	
PremiumsReceived	{"44.60";"44.04";"43.49...	=Functions!$G$2:G...	Workbook	
Qx	{" 0.000249640 "," 0.0...	=TableVariables!$C$3:...	Workbook	

Refers to:

=SeriesVariables!$D$2:$D$1001

Appendix A: Excel Functions

The following Excel functions can be used in mguchiQ:

Function	
Arithmetic	
	FLOOR
	CEILING
	POWER
	SQRT
	TRUNCATE
	MAX
	MIN
	ABS
Scientific	
	ACOS
	COSH
	COS
	ASIN
	SINH
	SIN
	ATAN
	TANH
	TAN
	LOG10
	LOG
	EXP
Boolean	
	IF
	AND
	OR
Financial	
	PV
	FV
	PMT