

Spreadsheet User Guide

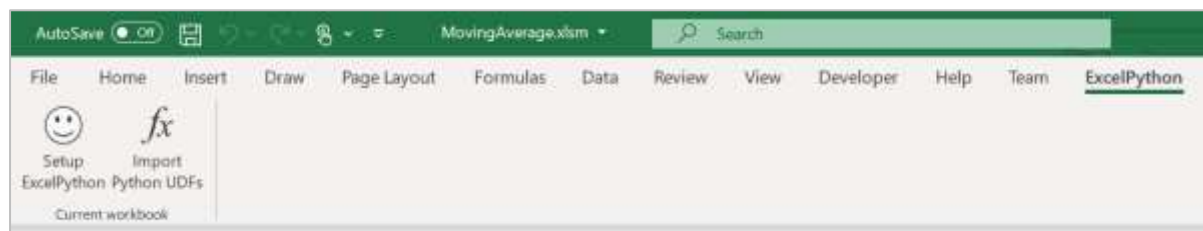
Version 2.1

Table of Contents

<u>SECTION 1: SETTING UP THE SPREADSHEET</u>	3
<u>SECTION 2: README WORKSHEET</u>	3
<u>SECTION 3: INPUT WORKSHEET</u>	4
<u>GENERIC PARAMETERS</u>	4
<u>SPECIFIC PARAMETERS</u>	5
<u>FUNCTION REQUESTS</u>	7
<u>BUTTONS</u>	8
<u>SECTION 4: HISTORICAL DATA</u>	10
<u>SECTION 5 : MA</u>	11
<u>SECTION 6 : CHARTS</u>	12
<u>SECTION 7 : STRATEGY</u>	13
<u>SECTION 8 : TESTING</u>	14
<u>SECTION 9: STRATEGY BACKTESTING</u>	16

SECTION 1: SETTING UP THE SPREADSHEET

To be able to use functionality relying on back-end Python coding, the user needs to set-up xlPython:



You should first click “Setup ExcelPython,” followed by “Import Python UDFs.” For more on the installation of the spreadsheet and its requirement, please consult “User Manual Spreadsheet Installation.docx.” However, it is your responsibility to ensure that any software you need to download or install for the purposes of running the TI Spreadsheet does not conflict with existing software on your device, leading to undesirable issues. Please note that VBA-Web is already installed in the spreadsheet to allow the execution of the VBA code that enables the automatic retrieval of stock prices from Yahoo!Finance.

When setting up (or running the VBA code behind) the TI Spreadsheet, you will be able to track the majority of errors in the error log file “errorLog_MovingAverage.xlsm.csv” that the custom-built VBA code creates automatically in the same folder as the TI Spreadsheet.

SECTION 2: README WORKSHEET

This worksheet briefly describes each subsequent worksheet in the workbook, any notes/warnings as well as the colour coding used in the spreadsheet.

SECTION 3: INPUT WORKSHEET

This is the backbone worksheet for input data; most input data are defined on this worksheet-only the "Testing" worksheet requires defining inputs locally. It also serves as the interface for initiating other functionality such as retrieving historical data from the web, updating the model sheet, launching code testing, running strategy back-testing and generating technical charts (Stock Price and MA).

GENERIC PARAMETERS

	A	B	C	D	E
1		Ticker	AMD		
2					
3					
4		Summary	Date		HolidayDates
5		Current Date	28/02/2022		01/01/2007
6					02/01/2007
7		Historical Data	Date		15/01/2007
8		Start Date	30/04/2018		19/02/2007
9		End Date	29/10/2021		06/04/2007
10					28/05/2007
11		Analysis Data	Date		04/07/2007
12		Start Date	30/04/2019		03/09/2007
13		End Date	26/04/2021		22/11/2007
14					25/12/2007
15		"Python.exe" path name			01/01/2008
16		PyPath	C:\Users\jaa06\A		21/01/2008
17					18/02/2008
18					21/03/2008
19					26/05/2008

This data will be required for any model calculation and any type of analysis.

- Ticker: C1; stock symbol as listed on the stock exchange.
- Dates:
 - ✓ Current Date: C5; this is the current analysis date, usually set to TODAY().
 - ✓ Historical Data Dates: C8, C9; these are the start and end dates (Start Date, End Date) that define the period for the whole data set. The user can then pick and choose which period within those dates will be used to carry out analyses (refer to the below point).
 - ✓ Analysis Data Dates: C12, C13; these are the start and end dates (Start Date, End Date) that define the period for the sample that the user wishes to analyse, or analysis data set. This data set will be extracted from the whole data set, bound by the above historical data dates, using those dates.
 - * These dates should lie in-between the Historical Data dates.
 - ✓ Holiday Dates: E5 and below; the days on which a holiday falls; these are user-defined and could be market specific. Those days will be excluded from any analysis over the period specified by the above analysis data dates. It is the user's responsibility to ensure they are kept up to date.
- PyPath: "Python.exe" path name C16: this is the name of the path where "Python.exe" is found. "Python.exe" is needed for running Python MA computations. If you have followed the installation in the "User Manual Spreadsheet Installation.docx" guide, you will need to type the Anaconda2 path here, e.g., C:\Users\\Anaconda2 or C:\Program Files\Anaconda2.

SPECIFIC PARAMETERS

- MA-Specific Parameters

G		H	
MA-Specific Parameter		Value	
Lookback Period n		7	
sPrice		Close	▼

- Lookback Period n: H6; this is the period upon which MA calculation will be based.
- sPrice: H7; this is the quantity that will be used to calculate MA. You should choose a value from the drop-down box, and it can be: Open, High, Low, Close, Volume, or Adj Close.

- Strategy BackTesting Parameters

J		K	L	M
Strategy BackTesting Input Parameters		Testing Range		
Parameter	Min	Max	increment	
n	7	9	1	
threshold	01/01/19			
Optimization Switch (OptSwitch)	1			
Output to Excel? (outputExcel)	TRUE			
Optimize Returns or PnL? (iOpt)	P&L	▼		

- n: K6:M6; Lookback Period; this is the period upon which MA calculation will be based. The strategy will be optimised based on the range of values that you define here. Min (K6) is the starting value for n, increment (M6) is by how much this value will be incremented at this loop and Max (L6) is the highest value taken into account in the optimisation.
- threshold: K7; this is the date dividing the data set into in-sample and out-of-sample.
- Optimization Switch (OptSwitch): K8; Keep this parameter set at 1 to allow optimisation.
- Output to Excel? (outputExcel): K9; Keep this parameter set to TRUE. It allows results to be written into Excel files.
- Optimize Returns or PnL? (iOpt): K10; this allows you to optimise returns or P&L. Choose Returns for optimising strategy returns and P&L for optimising the P&L
- * Some of these parameters are the same parameters as in the MA-Specific Parameters section but they are requested again to allow the independence of strategy back-testing from MA calculations.

• Strategy Parameters

O	P
Strategy Parameters	
Parameter	Value
Lookback Period n	7
iStrategy	1
Model Choice (modelName)	MovingAverage
Specific Model Choice (iMethod)	SMA
Sample Choice (sampleChoice)	All-Set
Python Speed Improvement (s)	None
Optimization Switch (optSwitch)	OFF
Output to Excel? (outputExcel)	TRUE
Use Model Sheet for Manual Calc? (useModelSheet)	FALSE
Optimize Returns or P&L? (iOpt)	P&L

- Lookback Period n: P6; this is the period upon which MA calculation will be based.
- iStrategy: P7; this is the type of strategy to be built (read-only at 1 at this stage).
- Model Choice (modelName): P8; this is the choice of model; only MovingAverage is fully implemented at this stage.
- Specific Model Choice (iMethod): P9; this is the choice of technique, e.g., SMA or EMA.
- Sample Choice (sampleChoice): P10; this is the type of sample for which the output will be displayed in the Strategy worksheet. You can choose between In-Sample, Out-of-Sample and All-Set (the whole set).
- Python Speed Improvement (s): P11; this allows improving speed in Python function calculation by setting the type an array is defined in Python or using Cython compiling. You can choose None (default order applies), C order, F (Fortran) order and Cython.
- Optimization Switch (optSwitch): P12; Keep this parameter set to OFF.
- Output to Excel? (outputExcel): P13; Keep this parameter set to TRUE. It allows results to be written into Excel files.
- Use Model Sheet for Manual Calc? (useModelSheet): P14; this allows you to use the model sheet, e.g. MA worksheet, to retrieve manual calculation or to redo them again. A TRUE value means the code will use the existing model (e.g. MA) sheet and FALSE means the code will calculate MA first.
- Optimize Returns or P&L? (iOpt): P15; at the moment, “Build Strategy” outputs both P&L and Returns given input parameters and the current strategy implementation, so it does not matter which choice you opt for. The code is anyway hard coded to revert to Returns.

FUNCTION REQUESTS

In this section, the user chooses which functions to run; the subsequent output will be shown in the (MA) model worksheet.

G	H	I	J
MA Calculation		MA Speed Improvement	
<input checked="" type="checkbox"/> SMA via Excel (Manual) <input checked="" type="checkbox"/> EMA via Excel (Manual)		<input checked="" type="checkbox"/> SMA Python Function (C order) <input checked="" type="checkbox"/> SMA Python Function (F order) <input checked="" type="checkbox"/> SMA Python Function (Cython) <input checked="" type="checkbox"/> EMA Python (C order) Function <input checked="" type="checkbox"/> EMA Python (F order) Function <input checked="" type="checkbox"/> EMA Python (Cython) Function	
<input checked="" type="checkbox"/> SMA VBA Function <input checked="" type="checkbox"/> EMA VBA Function <input checked="" type="checkbox"/> SMA Python Function <input checked="" type="checkbox"/> EMA Python Function			

- MA Calculation
 - ✓ MA_SMA via Excel (Manual): this is the step-by-step approach showing MA calculation using the SMA technique on the Excel worksheet. The SMA formula will be displayed in the cell (on the model Excel worksheet). This function should always be selected.
 - ✓ MA_EMA via Excel (Manual): this is the step-by-step approach showing MA calculation using the EMA technique on the Excel worksheet. The EMA formula will be displayed in the cell (on the model Excel worksheet). This function should always be selected.
 - ✓ MA_SMA VBA Function: this is a VBA user-defined function showing MA calculation using the SMA technique.
 - ✓ MA_EMA VBA Function: this is a VBA user-defined function showing MA calculation using the EMA technique.
 - ✓ MA_SMA Python Function: this is a Python user-defined function showing MA calculation using the SMA technique.
 - ✓ MA_EMA Python Function: this is a Python user-defined function showing MA calculation using the EMA technique.
 - ✓ MA_SMA Python (manual) Function: this is a Python user-defined function showing MA calculation using the SMA technique. Manual references the function implementation in the actual Python interface.
 - ✓ MA_EMA Python (manual) Function: this is a Python user-defined function showing MA calculation using the EMA technique. Manual references the function implementation in the actual Python interface.

- MA Speed Improvement: these are user-defined Python functions with built-in performance improvement.
 - ✓ MA_SMA Python Function (C order): the MA calculation is based on the SMA technique; arrays are defined in C order.
 - ✓ MA_SMA Python Function (F order): the MA calculation is based on the SMA technique; arrays are defined in F order.
 - ✓ MA_SMA Python Function (Cython): the MA calculation is based on the SMA technique; Cython compiling.
 - ✓ MA_Smoothing Python (C order) Function: the MA calculation is based on the Smoothing technique; arrays are defined in C order.
 - ✓ MA_Smoothing Python (F order) Function: the MA calculation is based on the Smoothing technique; arrays are defined in F order.
 - ✓ MA_Smoothing Python (Cython) Function: the MA calculation is based on the Smoothing technique; Cython compiling.

BUTTONS

- Generic
 - ✓ <Load Historical Prices into the "Historical Data" Sheet>: this button automatically extracts data from Yahoo! Finance based on the Ticker and the Historical Data dates.

**Load Historical Prices into
the "Historical Data" Sheet**

- Model Specific
 - ✓ <Update "MA" Sheet>: clicking this button sets-up the model page with all function outputs as per your above choices. It is fully automated, but you need to ensure that the input parameters (including Ticker, dates and MA-specific parameters) are set correctly.

Update "MA" Sheet

- ✓ <Generate Charts>: clicking this button creates the Charts sheet and all the technical charts on it. It is fully automated, but you need to ensure that Input, Historical Data and the MA (model) worksheets are set-up properly.

Generate Charts

- Testing
 - ✓ <Run Strategy Back-Testing>: clicking this button back-tests, in Python, the MA (model) strategy giving back optimised returns and parameters to the Strategy BackTesting worksheet. It is fully automated, but you need to ensure that Input, Historical Data and Strategy BackTesting are set-up properly.

Run Strategy Back-Testing

- ✓ <Build Strategy>: clicking this button builds the requested strategy. It creates the "Strategy" sheet and the associated six Excel files (point 6 below).

Build Strategy

SECTION 4: HISTORICAL DATA

This worksheet shows the whole data set for the stock symbol as defined by the user on the Input sheet. the following input data must be defined

- Ticker: C1 in the Input sheet; stock symbol as listed on the stock exchange.
- Historical Data "Start Date": C8 in the Input sheet.
- Historical Data "End Date": C9 in the Input sheet.
- After ensuring the input data is properly set, the user may generate this sheet by clicking the < Load Historical Prices into the "Historical Data" Sheet> button on the Input worksheet.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Volume	Adj Close
2	18/01/2018	12.18	12.68	12.15	12.47	62086200	12.47
3	19/01/2018	12.56	12.88	12.54	12.59	46800700	12.59
4	22/01/2018	12.61	12.74	12.37	12.65	34758600	12.65
5	23/01/2018	12.82	13	12.76	12.94	46417200	12.94
6	24/01/2018	13.02	13.05	12.6	12.71	41956400	12.71
7	25/01/2018	12.78	12.94	12.32	12.41	42966600	12.41
8	26/01/2018	12.7	12.95	12.62	12.95	44239000	12.95
9	29/01/2018	13.13	13.47	13.05	13.32	66645400	13.32
10	30/01/2018	13.12	13.27	12.69	12.87	91236400	12.87
11	31/01/2018	13.24	13.85	13.15	13.74	140977800	13.74
12	01/02/2018	13.62	13.84	13.21	13.25	67543300	13.25
13	02/02/2018	13.05	13.06	12.43	12.45	79432600	12.45
14	05/02/2018	12.05	12.18	10.93	11.57	116568900	11.57
15	06/02/2018	11.04	11.74	11	11.65	87123200	11.65
16	07/02/2018	11.7	11.9	11.53	11.6	55758400	11.6
17	08/02/2018	11.72	11.85	11.21	11.22	56471800	11.22
18	09/02/2018	11.54	11.57	10.63	11.31	85458200	11.31
19	12/02/2018	11.47	11.8	11.43	11.68	63160900	11.68
20	13/02/2018	11.6	11.82	11.53	11.78	33661400	11.78
21	14/02/2018	11.72	12.31	11.63	12.2	51756800	12.2

- The data can be overwritten manually but the column order has to be maintained.

SECTION 5 : MA

This worksheet presents model results as per user’s function requests.

- You need to define the inputs in the “Input” worksheet. In addition to the generic parameters such as dates and ticker, you need to define MA-specific parameters.
 - Lookback Period: H6.
 - Lower Bound: H7; defines the oversold condition.
 - Upper Bound: H8; defines the overbought condition.
 - You also need to select the functions for which you wish to run by checking the checkboxes in the “MA Calculation” and “MA Speed Improvement” sections.
- You should then click the <Update “MA” Sheet> button to generate the MA sheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	18		MA_SM A via Excel (Manual)	MA_EM A via Excel (Manual)	MA_SM A VBA Function	MA_EM A VBA Function	MA_SM A Python Function	MA_EM A Python Function	MA_EM A Python (C order) Function	MA_EM A Python (F order) Function	MA_EM A Python (Cython) Function	MA_SM A Python Function (Cython)	MA_SM A Python Function (F order)	MA_SM A Python Function (C order)
2	Date	Close	MA	MA	MA	MA	MA	MA	MA	MA	MA	MA	MA	MA
3	02/01/2018	10.98												
4	03/01/2018	11.55												
5	04/01/2018	12.12												
6	05/01/2018	11.88												
7	08/01/2018	12.28												
8	09/01/2018	11.82												
9	10/01/2018	11.96												
10	11/01/2018	12.14												
11	12/01/2018	12.02												
12	16/01/2018	11.91												
13	17/01/2018	12.18												
14	18/01/2018	12.47												
15	19/01/2018	12.59												
16	22/01/2018	12.65												
17	23/01/2018	12.94												
18	24/01/2018	12.71												
19	25/01/2018	12.41												
20	26/01/2018	12.95	12.20	12.20	12.20	12.20	12.20	12.20	12.20	12.20	12.20	12.20	12.20	12.20
21	29/01/2018	13.32	12.33	12.32	12.33	12.32	12.33	12.32	12.32	12.32	12.32	12.33	12.33	12.33
22	30/01/2018	12.87	12.40	12.37	12.40	12.37	12.40	12.37	12.37	12.37	12.37	12.40	12.40	12.40
23	31/01/2018	13.74	12.49	12.52	12.49	12.52	12.49	12.52	12.52	12.52	12.52	12.49	12.49	12.49
24	01/02/2018	13.25	12.57	12.60	12.57	12.60	12.57	12.60	12.60	12.60	12.60	12.57	12.57	12.57
25	02/02/2018	12.45	12.58	12.58	12.58	12.58	12.58	12.58	12.58	12.58	12.58	12.58	12.58	12.58
26	05/02/2018	12.55	12.55	12.55	12.55	12.55	12.55	12.55	12.55	12.55	12.55	12.55	12.55	12.55

- The code then generates the MA sheet with all the required results.

SECTION 6 : CHARTS

- This spreadsheet generates a chart of the stock price and MA.
- It is dependent on the MA worksheet.
- It is generated it by clicking <Generate Charts> in the Input worksheet.



SECTION 7 : STRATEGY

- This worksheet tests/verifies the calculation method/code implemented in the back-testing strategy code. The testing is automated.
- It is generated through the Input worksheet by clicking the <Build Strategy> Button.
- It is model (e.g., "MA") dependent so it shares inputs with the MA worksheet. It also requires an extra set of inputs defined by the user on the Input worksheet.
- The process is fully automated; this worksheet is created automatically by the VBA code.

VBA Macro Results			Python Function Results								Manual Excel Back Results								
T	Date	Close	MA	Position	MA_Return	MA_StrategyMA_Return	PnL	MA_Return	MA_ReturnMA_Return	MA_Return	MA_Return	Date	Close	MA	MA_Return	MA_ReturnMA_Return	PnL	MA_Return	MA_ReturnMA_Return
1	19/01/2018	12.54										19/01/2018	12.54						
2	21/01/2018	12.48										21/01/2018	12.65						
3	23/01/2018	12.94										23/01/2018	12.94						
4	24/01/2018	12.71										24/01/2018	12.71						
5	25/01/2018	12.41										25/01/2018	12.41						
6	26/01/2018	12.95										26/01/2018	12.95						
7	29/01/2018	12.32										29/01/2018	12.32						
8	30/01/2018	12.87										30/01/2018	12.87						
9	31/01/2018	13.74										31/01/2018	13.74						
10	01/02/2018	13.23										01/02/2018	13.23						
11	02/02/2018	12.49										02/02/2018	12.49						
12	05/02/2018	12.57										05/02/2018	12.57						
13	06/02/2018	13.49										06/02/2018	13.49						
14	07/02/2018	13.46										07/02/2018	13.46						
15	08/02/2018	13.46										08/02/2018	13.46						
16	09/02/2018	13.46										09/02/2018	13.46						
17	10/02/2018	13.22										10/02/2018	13.22						
18	13/02/2018	12.23										13/02/2018	12.23						
19	14/02/2018	12.54										14/02/2018	12.54						
20	15/02/2018	12.78										15/02/2018	12.78						
21	16/02/2018	12.32										16/02/2018	12.32						
22	19/02/2018	12.19										19/02/2018	12.19						
23	20/02/2018	12.82										20/02/2018	12.82						
24	23/02/2018	12.02										23/02/2018	12.02						
25	21/02/2018	12.72										21/02/2018	12.72						
26	22/02/2018	12.94										22/02/2018	12.94						
27	23/02/2018	12.87										23/02/2018	12.87						
28	26/02/2018	12.42										26/02/2018	12.42						
29	27/02/2018	12.53										27/02/2018	12.53						
30	28/02/2018	13.11										28/02/2018	13.11						
31	01/03/2018	12.90										01/03/2018	12.90						
32	02/03/2018	12.78										02/03/2018	12.78						
33	07/03/2018	12.24										07/03/2018	12.24						

- Output after clicking <Build Strategy> on the "Input" sheet: "Strategy" worksheet (with strategy returns and PnL results based on VBA code, Python code and Excel formulae). Six standalone excel files: "exampleAll.xlsx", "exampleIn.xlsx", "exampleOut.xlsx", "MAOutput_StrategyBackTest_AllReturns", "MAOutput_StrategyBackTest_InReturns" and "MAOutput_StrategyBackTest_OutReturns".

SECTION 8 : TESTING

This worksheet performs code or model testing when specific parameters change. The aim is to test that all functions, which implement the same model, work properly. This is measured by each function's output. All output should be equivalent to the step-by-step approach. It is *different from strategy back-testing*, which looks at maximised returns/PnL and optimised parameters for a specific historical period.

- Input

	A	B	C	D	E	F
1						
2		CODE TESTING				
3						
4		Look-back period				
5		Parameter	Start	End	Increment	
6		Number of periods	10	20	2	
7						
8						
9		Stock Symbol		Testing Dates		
10		Ticker		Test case	Start Date	End Date
11		JPM		1	05/02/2018	04/01/2019
12		BAC		2	16/05/2018	17/01/2020
13		F		3	08/01/2019	25/03/2019

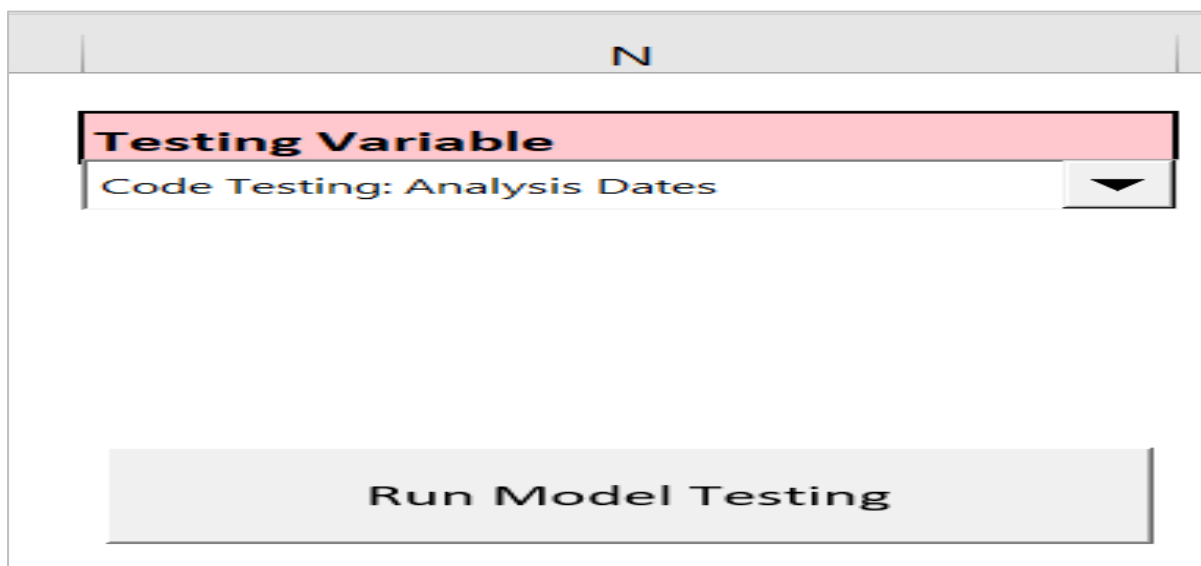
Most input data (for the testing code) is defined on this worksheet. However, you must ensure that the dates on the Input sheet and the Historical Data sheet are properly set-up since these data will be used by the testing code. You must define the range for the following parameters.

- Lookback Period: testing code performance of the requested functions on the Input sheet for different lookback periods; [Start,End] range with the periods in between updated by the increment.
 - Start: Cell C6; the first lookback period.
 - End: Cell D6; the last lookback period.
 - Increment: cell E6; a digit value that will be used to increment the current lookback period in order to get the next one.
 - For each lookback paper, a new worksheet is created.
For example, Start = 11, End = 15 and Increment = 2: lookbackPeriod_1 = 11, lookbackPeriod_2 = 13 and lookbackPeriod_3 = 15.
- Stock Symbol: testing the code for different stock symbols.
 - Input different stock symbols in cell B11.
 - For each stock symbol, a new worksheet will be created. So, in principle, the number of stock symbols to input is limited only by available memory. However, it is advisable to be sensible as to that number to avoid possible Excel crashing or other complication.
- Testing Dates: testing the code for various start and end Analysis Dates.
 - Start Date: starting cell E11.
 - End Date: starting cell F11.
 - As in the case of the other two variables, for each testing date, a new worksheet with the results is created. So the number of test cases that can be created is only limited by available memory. However, it is better to exercise caution in this case in order to avoid spreadsheet crashes or a major slow-down in the code.

- Output

For each variable, the below output files are generated in the Testing folder on the same path as the existing spreadsheet.

- Test_<name of variable>: this is an excel file that shows all the requested function outputs for the testing variable.
 - Test_<name of variable>_Compare: this is an Excel file that compares the results of each function, corresponding to a specific technique, to the benchmark case, normally the step-by-step approach but this depends on the user preferences. If the results in this case are 0 throughout, it means that the code is working well. For any discrepancies, as represented by nonzero values in the function outputs, you must refer to the below file.
 - Test_<name of variable>_Compare: this is a “.csv” file that describes the output results in text. If there are any errors, it will point them out alongside their position in the spreadsheet.
- The testing is fully automated.



- You must choose the variable to be tested in the drop-down box under <Testing Variable> on this sheet.
- Then, you should define the range of values of the corresponding variable in the required cells as defined above.
- You should then click the <Run Model Testing> button on this sheet.

SECTION 9: STRATEGY BACKTESTING

Strategy BackTesting Input Parameters		Testing Range			E		F		G		H		I		J	
Parameter	Min	Max	Increment													
n	7	9	1													
threshold	01/01/19															
Optimization Switch (OptSwitch)	1															
Output to Excel? (OutputExcel)	TRUE															
Optimize Returns or PnL? (iOpt)	2															
sPrice	Close															
MA Optimized Parameters and Returns/PnL				MA_SMA Python Function	MA_EMA Python Function	MA_EMA Python (C order) Function	MA_EMA Python (F order) Function	MA_EMA Python (Cython) Function	MA_SMA Python Function (Cython)	MA_SMA Python Function (F order)	MA_SMA Python Function (C order)					
iMethod	1	2	2	2	2	2	2	2	1	1	1					
Python Speed Improvement Approach (s)	0	0	1	2	3	3	2	1	2	1						
iStrategy	1	1	1	1	1	1	1	1	1	1						
n	7	7	7	7	7	7	7	7	7	7						
Optimization Switch (optSwitch)	1	1	1	1	1	1	1	1	1	1						
Output to Excel also? (OutputExcel)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE						
n	-8.00	-8.00	-8.00	-8.00	-8.00	-8.00	-8.00	-8.00	-8.00	-8.00						
In-Sample Optimized PnL	13.99	16.01	16.01	16.01	16.01	16.01	13.99	13.99	13.99	13.99						
Whole Sample PnL based on Optimized Parameters	-1.50	-6.02	-6.02	-6.02	-6.02	-6.02	-1.50	-1.50	-1.50	-1.50						
Out-Of-Sample PnL based on Optimized Parameters	-13.40	-19.94	-19.94	-19.94	-19.94	-19.94	-13.40	-13.40	-13.40	-13.40						
Description																
ONLY Python functions (excl. Manual) are used for Strategy Back-Testing																
n: Lookback Period				e.g. 13, 50, 150, 200 days												
iMethod: MA Computation Technique				= 2 for SMA technique												
				= 2 for EMA technique												
s: Type of Speed Improvement				= 1 for C order												
				= 2 for F order												
				= 3 for Cython												
				else no speed improvement												
iStrategy: Trading Strategy				= 1 for Strategy 1 (only this strategy is being back-tested now)												
README Input Historical Data Testing Charts MA Strategy BackTesting Strategy																

- This worksheet presents strategy back-testing results: optimised parameters and returns.
- Only Python functions are used for strategy back-testing.
- It is model dependent as it studies trading strategies that are pertinent to a particular model.
- It is generated through the Input sheet by clicking the <Run Strategy Back-Testing> button.
- The process is fully automated. Clicking this button creates a new Strategy Back-Testing worksheet showing input parameters, optimised parameters and optimised returns/PnL for each function you have requested on the Input worksheet.
- Output in two forms is generated: First, the Excel output is shown in the output table (MA Optimized Parameters and Returns) on the worksheet itself. Note that Python functions are populated in the Excel formulae here, rather than being calculated in the VBA code. Second, if “output to Excel?” (outputExcel) is set to TRUE, then three Excel files will be generated (“exampleIn_Opt”, “exampleOpt” and “MAOutput_StrategyBackTest_InReturns_Opt”).
- There is a section at the bottom of the spreadsheet that describes various parameters that feed into Python functions and their respective values.