BUSINESS ANALYTICS CLUB

Workshop Series 10.24

Excel Visual Basics for Applications

Janet Ye

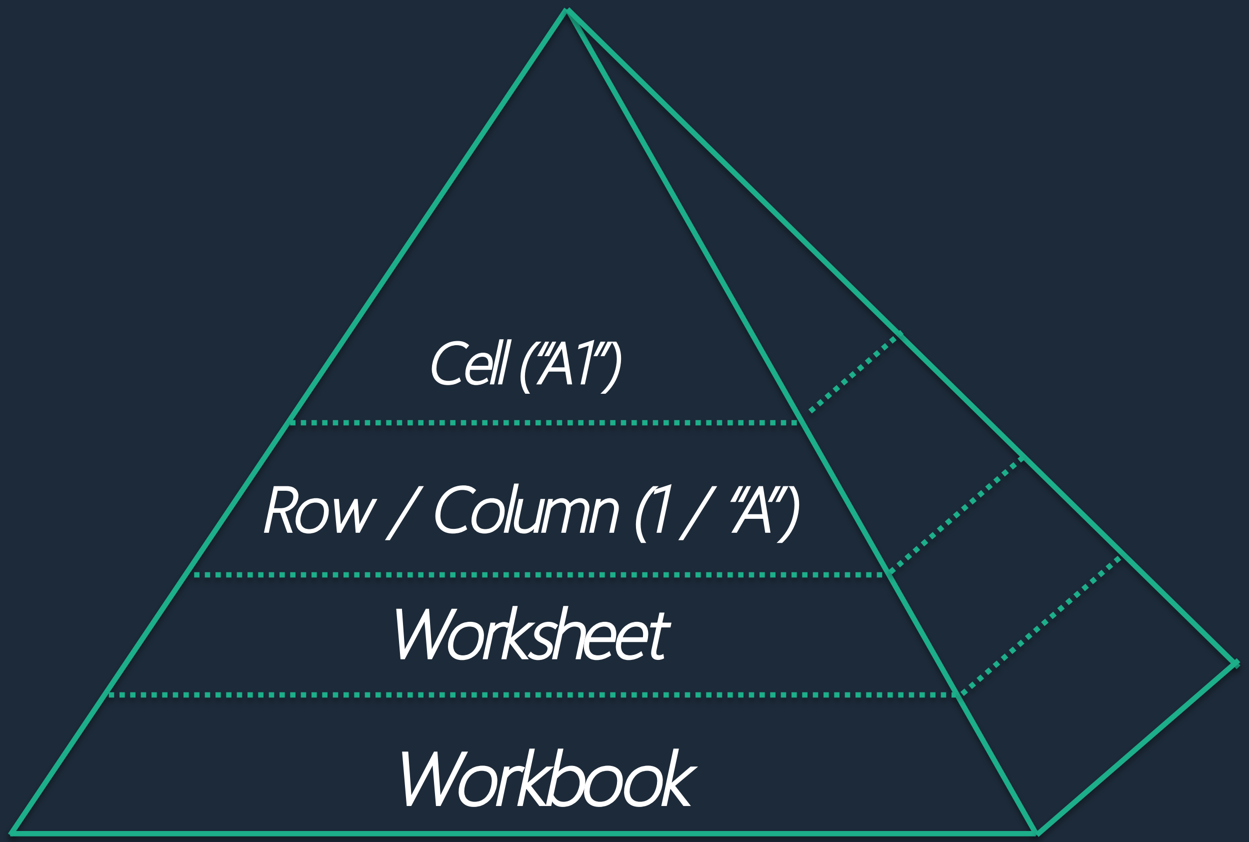
Learning Objective

1. Learn sample tasks in Excel VBA to clean a large dataset for analysis in other programming languages
   a) Data Type
   b) Object Hierarchy
   c) Functions
2. Powerful to to data cleaning
   a) Microsoft Corporate VP, 10/26
   b) Credit Suisse, Big Data, 11/3
   c) Registration / Office hour, 11/5
   d) Digital Reasoning, detecting fraud +/ terrorism, 11/12

Data Type

- Types of Data in Excel
  - Name of city: New York (words)
  - Length of West 4th Street: 2.00 (decimal)
  - Population: 8,500,000 (integer)
  - MetroCard fare: 2.75 (currency)
  - Workshop location: "Tisch Hall" (string)
  - Pass/Fail: false (boolean)
  - Average grade: 95.7 (double)
  - Food Rating: A (character)

Excel Hierarchy



Sample Use Cases

- When user selects certain cells to run: modules
  - User runs the code the tally all numbers
  - User runs the code to form a table from entries
  - 2 Types:
    - Subroutine: code that perform a set of actions / calculations. Does not return values
    - Function: like subroutine, but returns a value
- When action is performed within the entire workbook/worksheet
  - Code is run when the workbook is opened or modified, etc.

Running VBA Codes

- Programs updates automatically
  - No need to “compile” as we would need to in other programming languages
- No undoing once programs are run and make changes
  - You should save the file before running the code
  - OR, sometimes make a duplicate of the file to run the code on

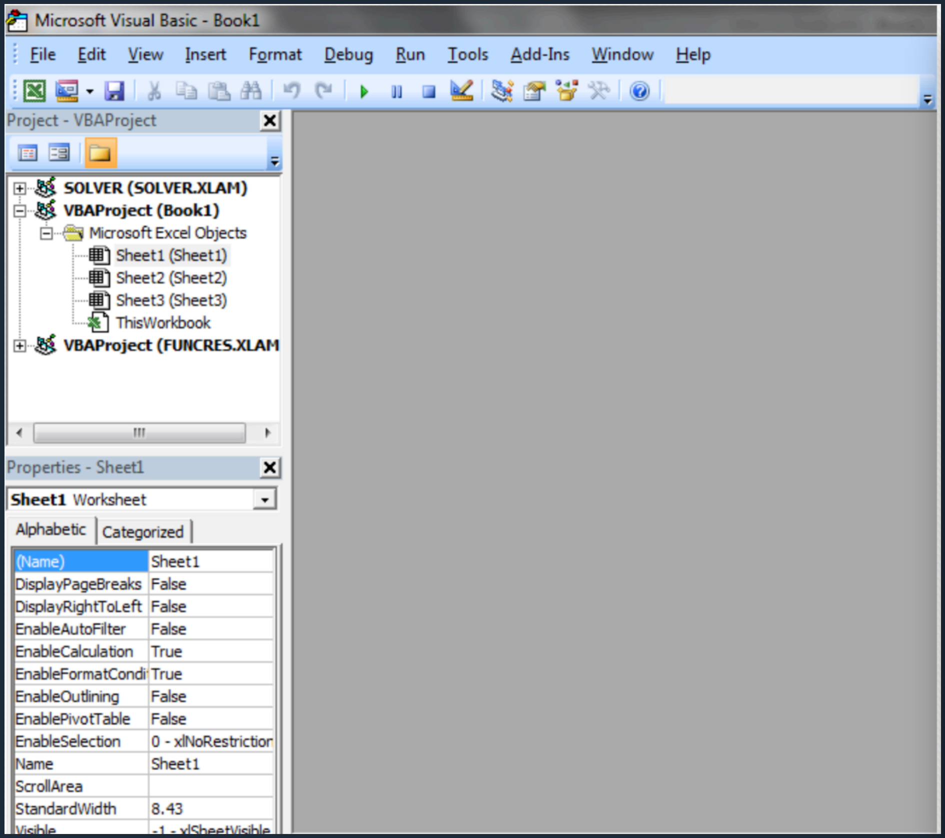
Launching VBA

PC

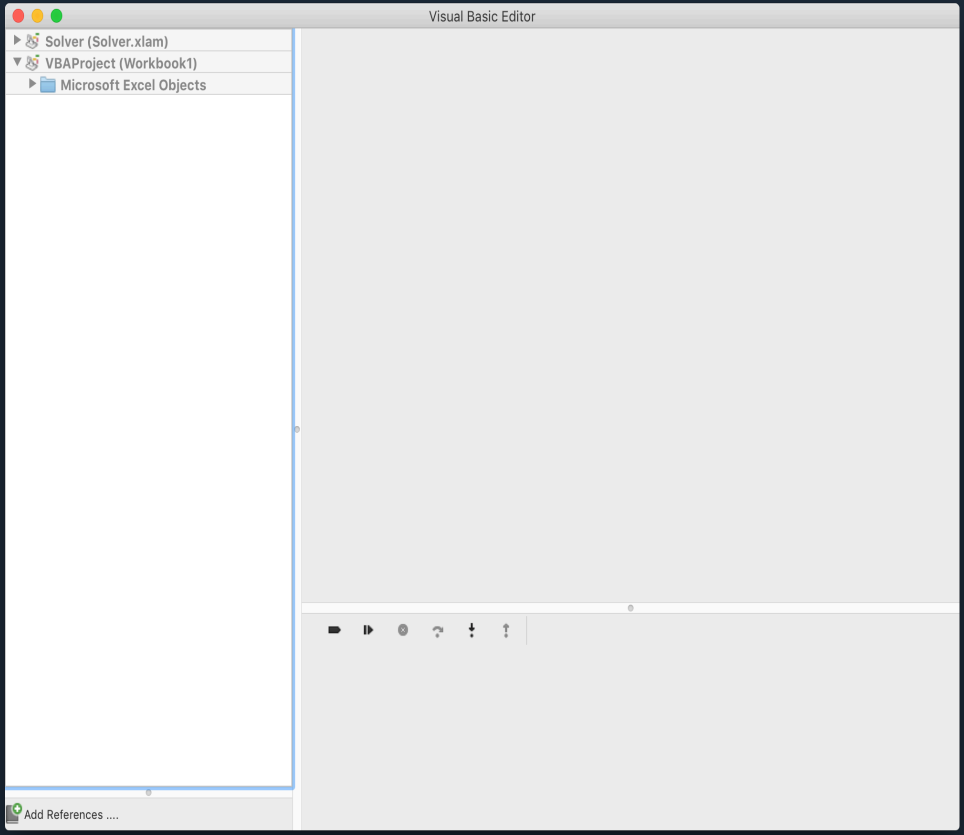
ALT + F11

Mac

Tool -> Macro -> Visual Basic Editor

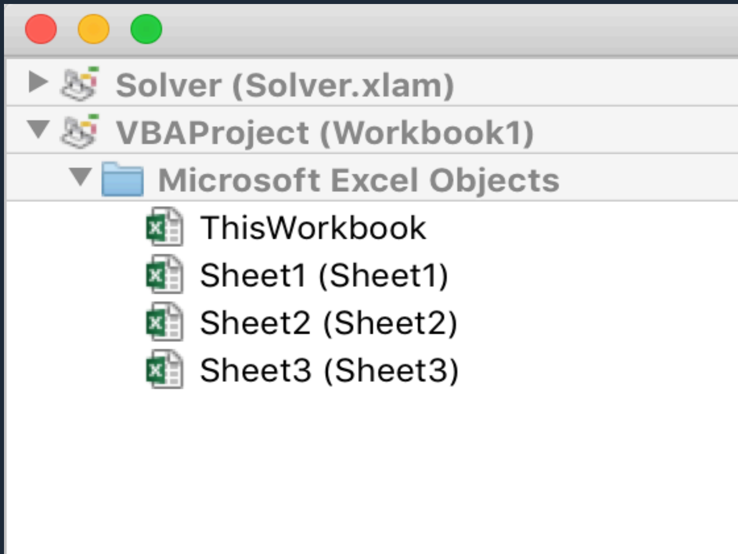


(Name)	Sheet1
DisplayPageBreaks	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatCondi	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestrictior
Name	Sheet1
ScrollArea	
StandardWidth	8.43
Visible	-1 - xlSheetVisible



Microsoft Excel Objects

- Each sheet has been set by default as an object
  - Sheet1 (Sheet1)
  - Sheet2 (Sheet2)
  - Sheet3 (Sheet3)
- The workbook is also an object
- Code should be written in the respective object that you want the actions top be performed



VBA-pardy

100	200	300	400
500	600	700	800

Rules

- The teams will be made up of Left and Right halves of the room. Think of a team name!
- Objective is to score as many points as possible.
- The problem / point value will be projected.
- First team to post a correct answer to the problem on the BAC facebook wall wins: www.facebook.com/SternBAC (specify your team name)
- Team with the winning points will receive a mysterious prize

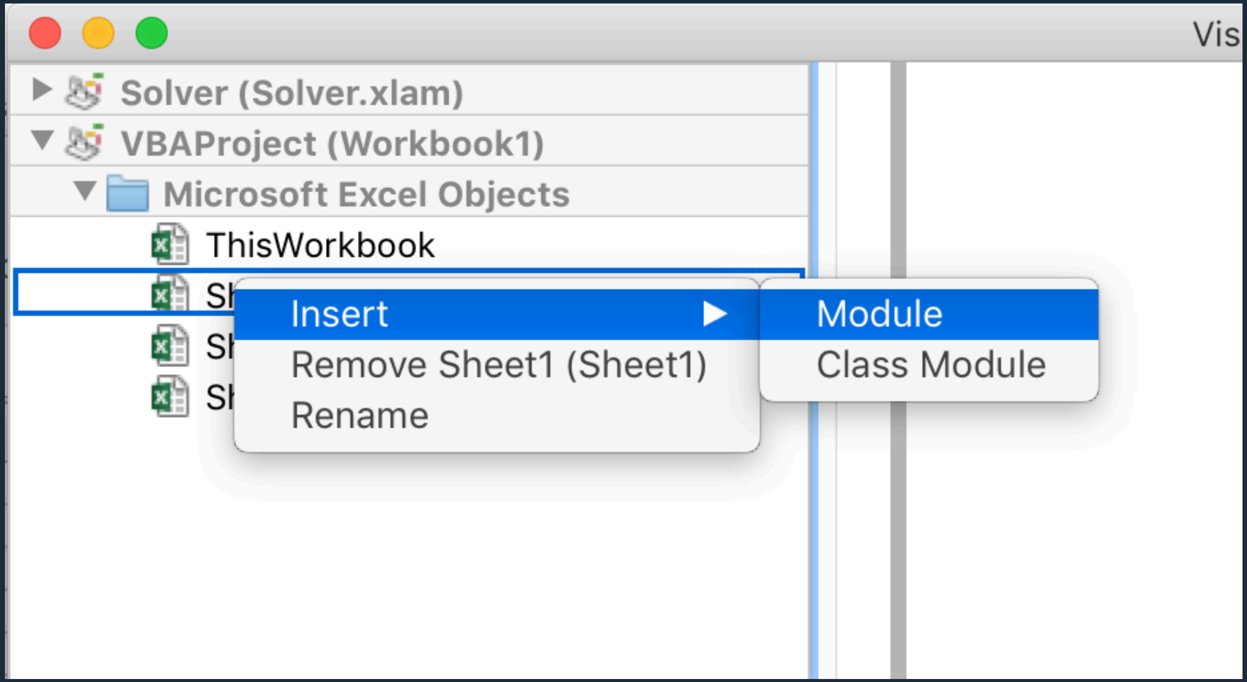
Function Example

Function Example

- Task: given three digits a, b, c, and return the number abc
  - For example: 4, 5, 6 are passed, function returns 456
- Calculation: 100 x a + 10 x b + c
- Call this function concat
- Module OR entire worksheet?

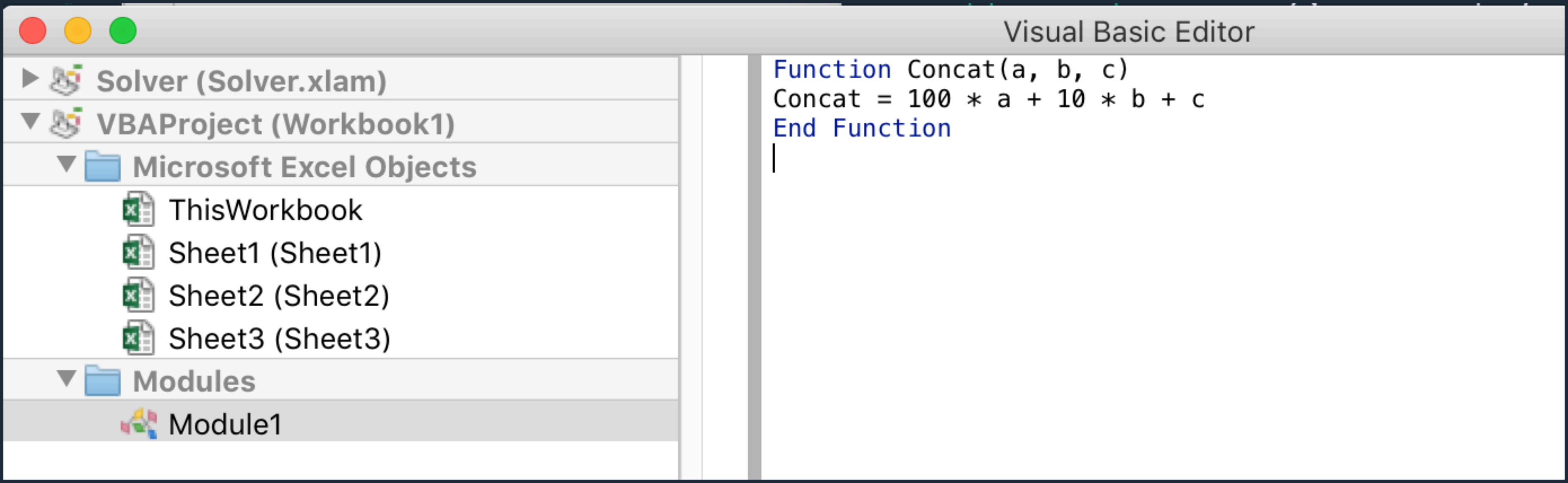
Create a Module

- Right Click Sheet -> Insert -> Module



Create a Module

```
Function Concat(a,b,c)
Concat = 100 * a + 10 * b + c
End Function
```



```
Function Concat(a, b, c)
Concat = 100 * a + 10 * b + c
End Function
```

Boolean Example

Coachella Game

Are these band names Real or Fake?

Gold Panda	*Real*
Petting Zoo	*Fake*
Evil Needle	*Real*
Barilla Penne	*Fake*
Bugseed	*Real*

Boolean Example

Boolean returns whether statement is True or False.

Is cell A1 equal to cell A2?	= A1 = A2
Is A greater than *5 times* B?	= A1 > (A2*5)
Is *10 times* A1 not equal to *one fifth* of A2?	= (A1 * 10) <> (A2 / 5)

Loop Example

Loop Example

Let’s print 1, 2, …, 20 in columna A of the first 20 rows.

A1 / Cell (1,1) = 1
A2 / Cell (2,1) = 2
A3 / Cell (3,1) = 3
…
A20 / Cell (20,1) = 20

For Loop

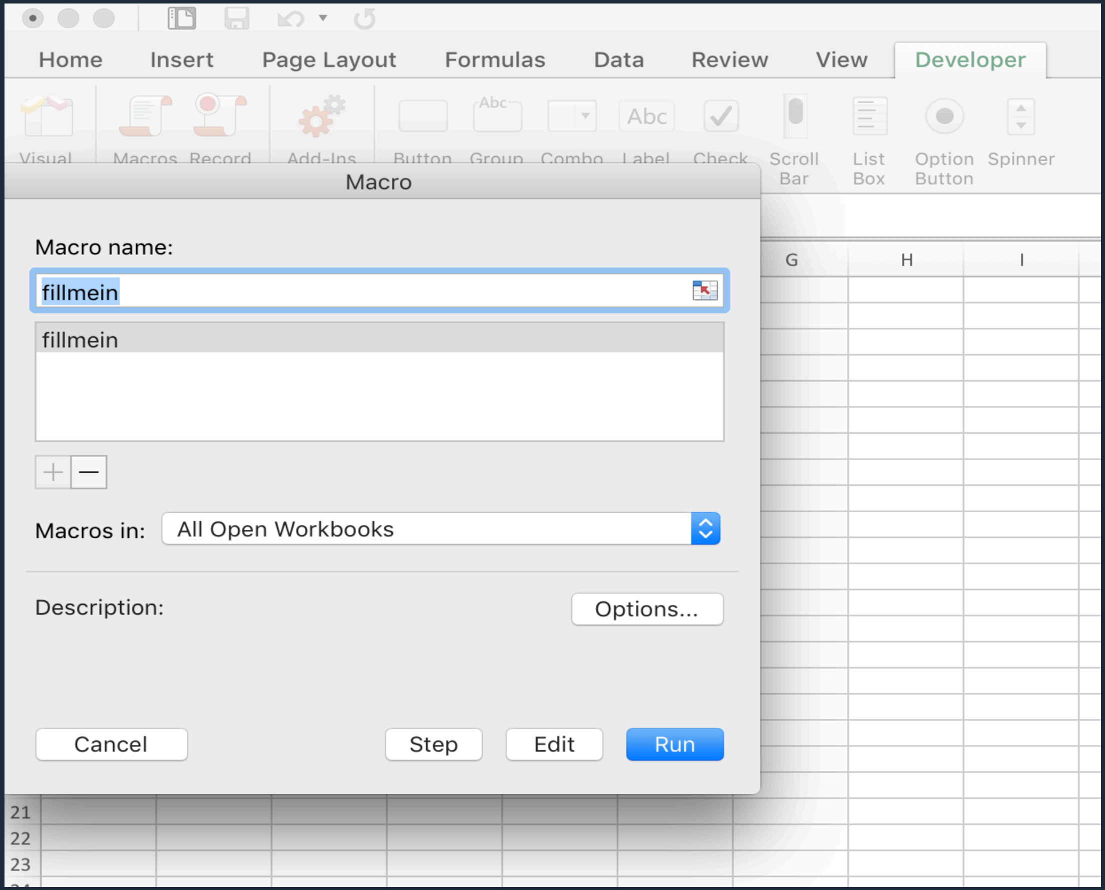
```
Sub fillmein()
For Row = 1 To 20
Cells(Row, 1) = Row
Next Row
End Sub
```

Recall, there is no values to return. We are writing a subroutine, which is just a generic routine – in this case, fill in the first 20 rows with values.

Common Mistakes: `Cells` not `Cell`.
Don't forget `Next Row`, and `End Sub`

For Loop

To run the script, [Developer] tab -> [Macro] -> [fillmein] -> [run]



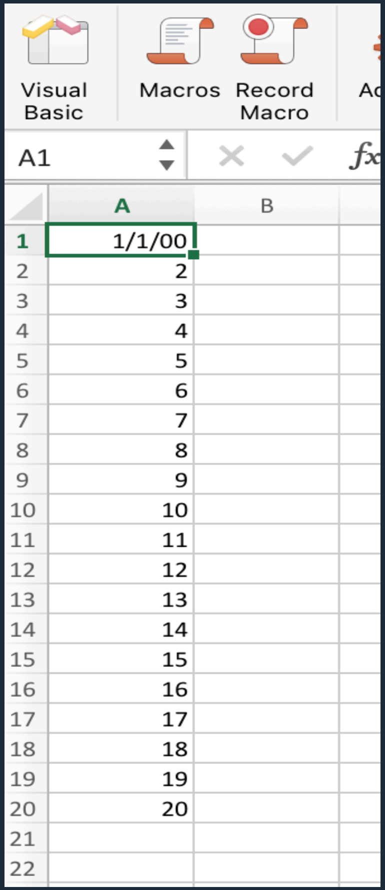
Macro

Macro name: fillmein

fillmein

Macros in: All Open Workbooks

Description:



	A	B
1	1/1/00	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	
11	11	
12	12	
13	13	
14	14	
15	15	
16	16	
17	17	
18	18	
19	19	
20	20	
21		
22		

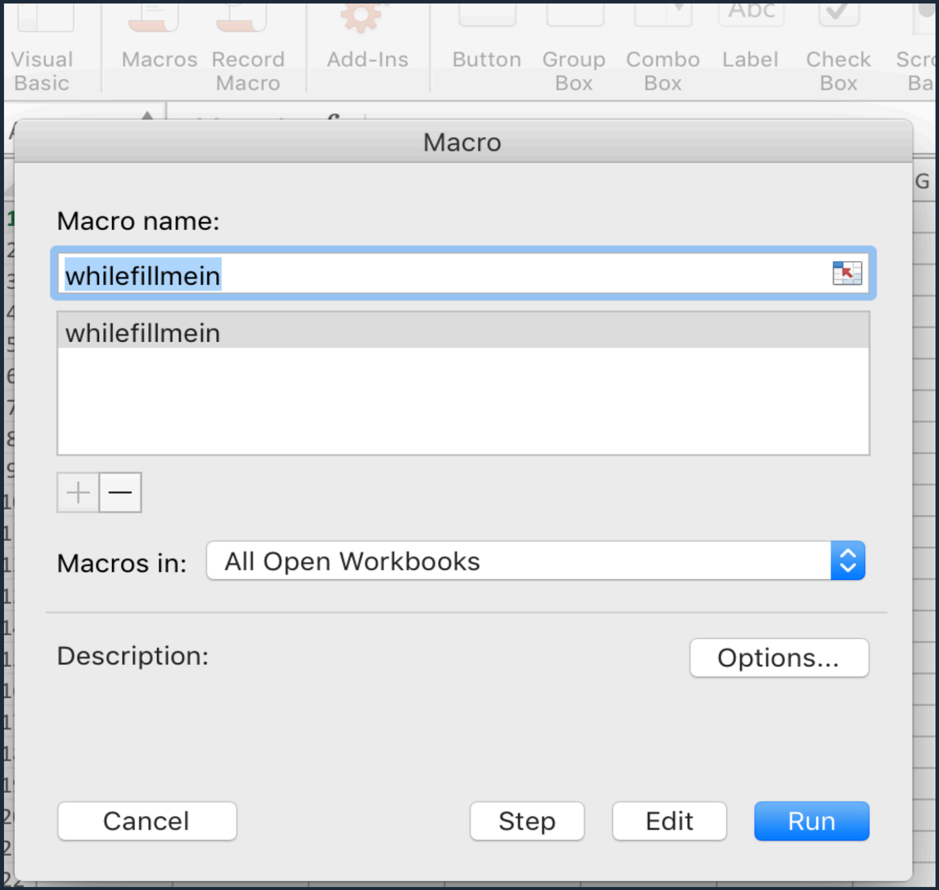
While Loop

```
Sub whilefillmein()
Row = 1
While Row <= 20
Cells(Row, 1) = Row
Row = Row + 1
Wend
End Sub
```

Key Difference: in For loop, we use `Next Row` to increment.
In While loop, we have to manually increment by calling `Row = Row + 1`

While Loop

To run the script, [Developer] tab -> [Macro] -> [fillmein] -> [run]

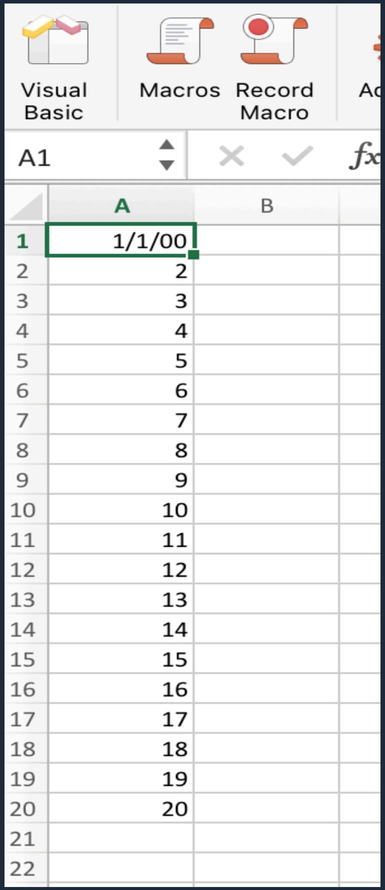


Macro name: whilefillmein

whilefillmein

Macros in: All Open Workbooks

Description:

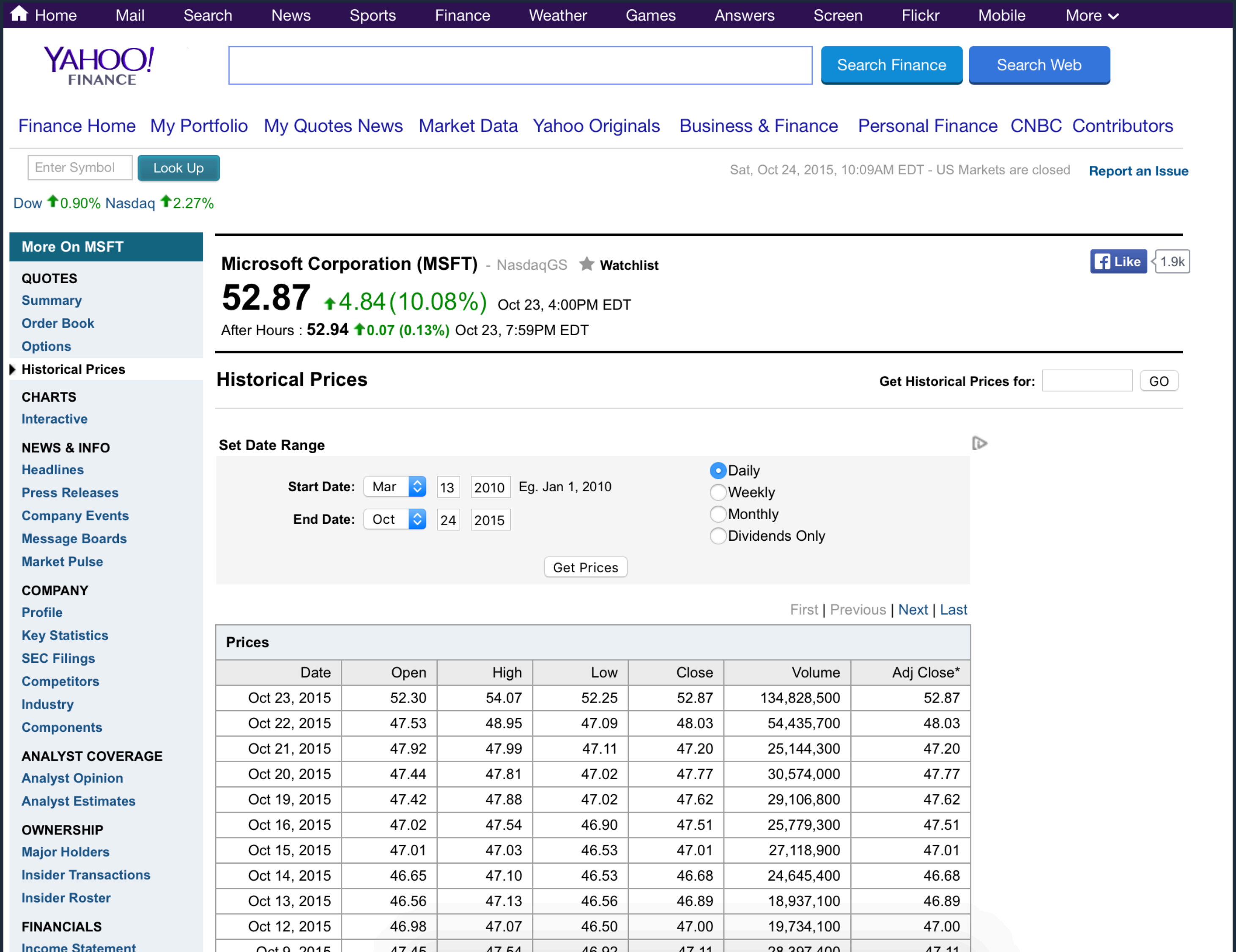


	A	B
1	1/1/00	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	
11	11	
12	12	
13	13	
14	14	
15	15	
16	16	
17	17	
18	18	
19	19	
20	20	
21		
22		

Tying everything together

Simple Trading Buy/Sell Signal

Microsoft Stock Price from Yahoo Finance



Sat, Oct 24, 2015, 10:09AM EDT - US Markets are closed

Dow ⬆0.90% Nasdaq ⬆2.27%

Microsoft Corporation (MSFT) - NasdaqGS

52.87 ⬆4.84(10.08%) Oct 23, 4:00PM EDT

After Hours : **52.94** ⬆0.07 (0.13%) Oct 23, 7:59PM EDT

Historical Prices

Get Historical Prices for:

Set Date Range

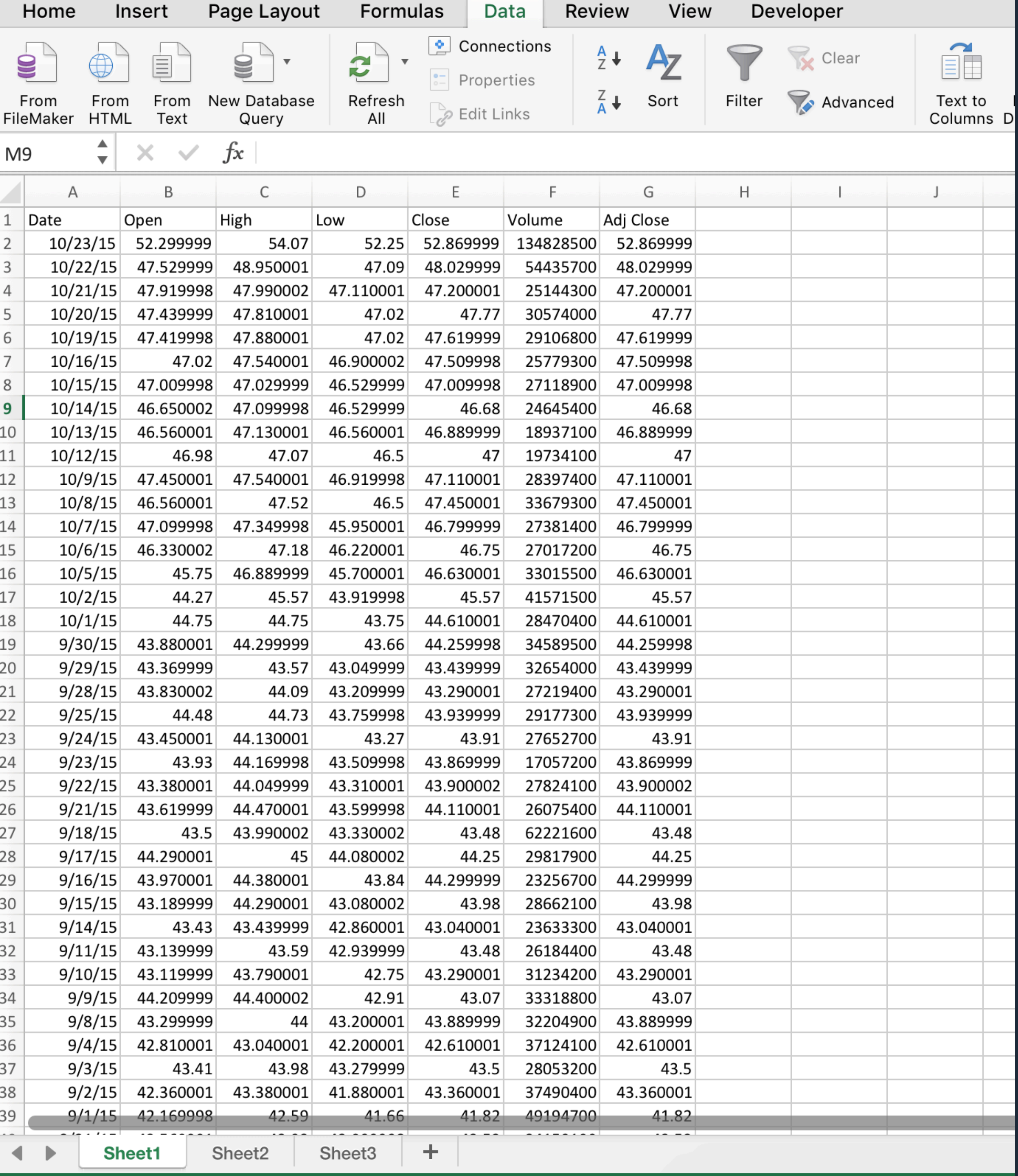
Start Date: Mar 13 2010 Eg. Jan 1, 2010

End Date: Oct 24 2015

● Daily
○ Weekly
○ Monthly
○ Dividends Only

Prices

Date	Open	High	Low	Close	Volume	Adj Close*
Oct 23, 2015	52.30	54.07	52.25	52.87	134,828,500	52.87
Oct 22, 2015	47.53	48.95	47.09	48.03	54,435,700	48.03
Oct 21, 2015	47.92	47.99	47.11	47.20	25,144,300	47.20
Oct 20, 2015	47.44	47.81	47.02	47.77	30,574,000	47.77
Oct 19, 2015	47.42	47.88	47.02	47.62	29,106,800	47.62
Oct 16, 2015	47.02	47.54	46.90	47.51	25,779,300	47.51
Oct 15, 2015	47.01	47.03	46.53	47.01	27,118,900	47.01
Oct 14, 2015	46.65	47.10	46.53	46.68	24,645,400	46.68
Oct 13, 2015	46.56	47.13	46.56	46.89	18,937,100	46.89
Oct 12, 2015	46.98	47.07	46.50	47.00	19,734,100	47.00
Oct 9, 2015	47.45	47.54	46.92	47.11	28,397,400	47.11



	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Volume	Adj Close
2	10/23/15	52.299999	54.07	52.25	52.869999	134828500	52.869999
3	10/22/15	47.529999	48.950001	47.09	48.029999	54435700	48.029999
4	10/21/15	47.919998	47.990002	47.110001	47.200001	25144300	47.200001
5	10/20/15	47.439999	47.810001	47.02	47.77	30574000	47.77
6	10/19/15	47.419998	47.880001	47.02	47.619999	29106800	47.619999
7	10/16/15	47.02	47.540001	46.900002	47.509998	25779300	47.509998
8	10/15/15	47.009998	47.029999	46.529999	47.009998	27118900	47.009998
9	10/14/15	46.650002	47.099998	46.529999	46.68	24645400	46.68
10	10/13/15	46.560001	47.130001	46.560001	46.889999	18937100	46.889999
11	10/12/15	46.98	47.07	46.5	47	19734100	47
12	10/9/15	47.450001	47.540001	46.919998	47.110001	28397400	47.110001
13	10/8/15	46.560001	47.52	46.5	47.450001	33679300	47.450001
14	10/7/15	47.099998	47.349998	45.950001	46.799999	27381400	46.799999
15	10/6/15	46.330002	47.18	46.220001	46.75	27017200	46.75
16	10/5/15	45.75	46.889999	45.700001	46.630001	33015500	46.630001
17	10/2/15	44.27	45.57	43.919998	45.57	41571500	45.57
18	10/1/15	44.75	44.75	43.75	44.610001	28470400	44.610001
19	9/30/15	43.880001	44.299999	43.66	44.259998	34589500	44.259998
20	9/29/15	43.369999	43.57	43.049999	43.439999	32654000	43.439999
21	9/28/15	43.830002	44.09	43.209999	43.290001	27219400	43.290001
22	9/25/15	44.48	44.73	43.759998	43.939999	29177300	43.939999
23	9/24/15	43.450001	44.130001	43.27	43.91	27652700	43.91
24	9/23/15	43.93	44.169998	43.509998	43.869999	17057200	43.869999
25	9/22/15	43.380001	44.049999	43.310001	43.900002	27824100	43.900002
26	9/21/15	43.619999	44.470001	43.599998	44.110001	26075400	44.110001
27	9/18/15	43.5	43.990002	43.330002	43.48	62221600	43.48
28	9/17/15	44.290001	45	44.080002	44.25	29817900	44.25
29	9/16/15	43.970001	44.380001	43.84	44.299999	23256700	44.299999
30	9/15/15	43.189999	44.290001	43.080002	43.98	28662100	43.98
31	9/14/15	43.43	43.439999	42.860001	43.040001	23633300	43.040001
32	9/11/15	43.139999	43.59	42.939999	43.48	26184400	43.48
33	9/10/15	43.119999	43.790001	42.75	43.290001	31234200	43.290001
34	9/9/15	44.209999	44.400002	42.91	43.07	33318800	43.07
35	9/8/15	43.299999	44	43.200001	43.889999	32204900	43.889999
36	9/4/15	42.810001	43.040001	42.200001	42.610001	37124100	42.610001
37	9/3/15	43.41	43.98	43.279999	43.5	28053200	43.5
38	9/2/15	42.360001	43.380001	41.880001	43.360001	37490400	43.360001
39	9/1/15	42.169998	42.59	41.66	41.82	49194700	41.82

Paste into Excel, Text to Column

Use While Loop to Calculate Return

Daily Return = (Close – Open) / Open
Fill in Column H with Return

```
Sub buysellsignal()
return_col = 8
Cells(1,return_col) = “Return”
Row = 2 ‘first row is title
While Cells(Row, 2) <> “”
Cells(Row, return_col) = (Cells(Row, 5)-
Cells(Row, 2))/(Cells(Row,2))
Row = Row + 1
Wend
End Sub
```

Three Day Moving Average

Average of three days' returns. New code highlighted below.

```
Sub buysellsignal()
Cells(1, 8) = "Return"
Cells(1, 9) = "3D MA"
return_col = 8
three_day_ma_col = 9
Row = 2 'first row is title
While Cells(Row, 2) <> ""
Cells(Row, return_col) = (Cells(Row, 5) - Cells(Row, 2)) /
(Cells(Row, 2))
If (Cells(Row + 1, return_col) <> "") And (Cells(Row + 2,
return_col) <> "") Then
Cells(Row, three_day_ma_col) = (Cells(Row, return_col) +
Cells(Row + 1, return_col) + Cells(Row + 2, return_col)) / 3
End If
Row = Row + 1
Wend
End Sub
```

VBA is a Powerful Tool

This is just a toy example.
VBA is capable of doing so much more, such as:

Cleaning data
Updating spreadsheet every time it’s opened or modified