

2010

U.S.A.F

Michael T. Cook

michael.cook@usafa.edu

Phone: 719-333-9730

[ANALYST GUIDE TO EXCEL]

Intro

Excel is a powerful tool for analysis. Not only is it powerful, but it is very common for many deployed locations use it to track information. Although it's not the best for tracking information nor for analysis, its widespread use mandates attention from analysts.

Setting up an Excel Database

Whether you start from scratch or if you already have a tracking product, creating/reformatting an excel database into an organized format should help you set up your product for quick analysis and reports. The following is an organizational system to get you started.

Point of Contact (POC) Tab

The POC tab should be a place where users can indicate what they use on your spreadsheet. When working with users who contribute to your data or use your products, this will be beneficial. Users should list what they use on the product or input into the product. It gives you a list of constraints for redesigning/reformatting your database or who to contact if something changes. Just have a simple first tab, that has a name, point of contact, and which sheets/cells are used.

Information and Instructions Tab

The information and instructions should contain relevant information about your data. You should include information on your data sources that answer the following questions:

- Where do you get your information from?
- What data do your columns hold? What are the units?
- What assumptions do you make, if any?

Data Entry Tab

The data entry tab should be setup as the fastest way to get from your data source to your data sheet. Data entry for users does not come in convenient row by row format. If it is not in row form or if you get data from multiple sources, it might be beneficial to have a tab where you can paste data from your sources and have that tab reformat the data to a single row that you can paste into your data sheet (note: this could be a simple macro for more advanced users).

Data Tab

The data tab should only have data. Don't put reports or summaries in this tab. Cluttering this tab will only reduce your efficiency.

Reports Tab

The reports tab should contain all the data reports that you need. You can create smaller data tables that pull data from the main data tab to create moving averages or moving predictions.

General Tools

The tools and methods here may make more sense as you read the other sections. You can use the tools here as a reference and come back as needed.

Color Use

Color coding columns/rows/cells can speed up your processes. Assign different colors to areas that contain calculations, user variables, and results. Be consistent with your color scheme and document it on the information and instructions tab.

Caution: Be careful what you are trying to color. If you want to count it later on, then you should not color it, but assign an additional column so you can put in something that can be counted.

Pivot Tables

Pivot tables are a quick way to sort, organize, and sum up the data in data tab. While pivot tables are great tools, they are not the best solution for every job. They are great for quick turn analysis, but repeat products will probably be easier with other tools. Pivot tables make it difficult to configure graphs based on pivot tables because you can't change the view without changing pivot table. And even then some parts of the pivot table graph cannot be changed. But a graph built from regular cells, however, is fully configurable. Pivot tables will be discussed more later on.

Full columns vs specific ranges

Since your data tab has only data in it and one column of data is always the same, you can take advantage of the simplicity and use column names only in your formulas. In doing so, you don't have to worry about the last row of data.

Example: =Sum(A:A) vs =Sum (A1:A100)

Excel will take either form, but =Sum(A:A) will automatically sum all of the data as the database gets updated. The other formula will stop at row 100.

(Note: You can also make dynamic named ranges, but they seem to be a little buggy)

Combining Text

Sometimes you have to combine text fields in excel. To combine text and/or numbers you'll use the ampersand (&) above the 7 key. You can combine text and formulas to make statements.

Examples in a Cell

= "The dog has " & randbetween(0,5) & " flees."

= "Our clients bought \$ "&=round(A15,2)&" worth of product."

= "We dropped "&=A15&" bundles of food."

You can also combine text inside of a formula

Example in a Formula

=sumifs(D:D, A:A, ">="A2, A:A, "<="&A3)

Here the ampersand is used to combine >= and A2 into a single formula that is interpreted as >=A2

*Averageifs/Sumifs/Countifs (*ifs statements)*

The averageifs/sumifs/countifs all have the same basic structure. These functions are used when you have one or more criteria for a sum, average, or count. For example, if you want to count the number of times an emergency airdrop occurred, but you only need to know the ones that happened between certain days and also further wanted restrict that to a certain parachute type, then you can quickly make a *countifs* statement to count those emergency airdrops based on the two criteria mentioned (specific days and parachute type).

The general structure for the *ifs statements is as follows:

- =Averageifs(Range to be Averaged, CriteriaRange1, Criteria1, CriteriaRange2, Criteria2, ...)
- =Sumifs(Range to be Summed, CriteriaRange1, Criteria1, CriteriaRange2, Criteria2, ...)
- =Countifs(Range to be Counted, CriteriaRange1, Criteria1, CriteriaRange2, Criteria2, ...)

Sample Shoe Store Data

	A	B	C	D
1	Date	Type	Store	Cost
2	17-Mar-10	GX1	East	77.89
3	22-Mar-10	AR72	West	75.98
4	25-Mar-10	GX1	West	74.43
5	26-Mar-10	GX1	East	73.85
6	28-Mar-10	AR72	North	73.92
7	31-Mar-10	GX1	East	75.13
8	1-Apr-10	GX1	West	77.29
9	5-Apr-10	AR72	East	73.73

Problem1: Find the average cost of the GX1 at the west store between 21 Mar 10 and 2 Apr 10.

- Solution1:
 - You could solve the problem specifically by using the following command:
 - =averageifs(D:D, A:A, ">="& date(2010,3,21), A:A, ">="&date(2010,4,1), B:B, "=GX1")
- Solution2:
 - =averageifs(D:D, A:A, ">="ReportTab!A2", A:A, "<="ReportTab!A3", B:B, "=ReportTab!A4")
 - Cells A2, A3, and A4 in the report tab should then have the start date, end date, and type.

Sumproduct

The Sumproduct command computes the sum of a product. In the table below the total cost can be found by the command =sumproduct(B:B,C:C)

	A	B	C
1	Date	Quantity	Cost
2	17-Mar-10	15	77.89
3	22-Mar-10	2	75.98
4	25-Mar-10	6	74.43
5	26-Mar-10	8	73.85
6	28-Mar-10	9	73.92
7	31-Mar-10	10	75.13
8	1-Apr-10	5	77.29
9	5-Apr-10	1	73.73

Shortcuts

Copy (Control-c)

Cut (Control-x)- Similar to copy except the selection

Paste(Control-v)

Paste Special – Values Only (Alt-e-s-u)

Select to Last filled Cell – (Control-Shift-Arrow)

Move to Last Filled Cell – (Control-Shift-Arrow)

Select All – (Control-a)

Move to Start – (Control-Home)

Move to End – (Control-End)

Dragging Formulas and Locking Cells

If you have a formula you want to copy to other cells, you need to make sure to 'lock' the appropriate cells or columns. The dollar sign before a row or column will lock the row or column so that dragging the formula does not affect the row/column in the formula.

For example, let's make the multiplication table in Excel.

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1									
3	2									
4	3									
5	4									
6	5									
7	6									
8	7									
9	8									
10	9									

In cell B2 enter the formula =A2*B1. Drag the formula to J10 by selecting cell B2, move your cursor to the bottom right of cell B2. When your cursor changes to a cross shape you can left click and drag the formula to the right, then with those cells still selected do the same thing and drag the row of formulas down to the end of the 10th cell.

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1	2	6	24	120	720	5040	40320	362880
3	2	2	4	24	576	69120	5E+07	3E+11	1E+16	4E+21
4	3	6	24	576	331776	2E+10	1E+18	3E+29	3E+45	1E+67
5	4	24	576	331776	1E+11	3E+21	3E+39	8E+68	2E+114	3E+181
6	5	120	69120	2E+10	3E+21	6E+42	2E+82	2E+151	4E+265	#NUM!
7	6	720	5E+07	1E+18	3E+39	2E+82	3E+164	#NUM!	#NUM!	#NUM!
8	7	5040	3E+11	3E+29	8E+68	2E+151	#NUM!	#NUM!	#NUM!	#NUM!
9	8	40320	1E+16	3E+45	2E+114	4E+265	#NUM!	#NUM!	#NUM!	#NUM!
10	9	362880	4E+21	1E+67	3E+181	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!

Looks funny doesn't it? Not the multiplication you learned in grade school.

Cell C6 ends up with the formula =B6*C5 when instead we want it to be =A6*C1. This is because without locking the row or column the formula will increase the row number when you drag the formula down one row and likewise for the columns. We want the first term in the product to always come from Column A and the second term in the product to come from Row 1.

Change the formula to =\$A2*B\$1. This 'locks' on the right column and row as mentioned above. Drag the formulas in the same way mentioned above.

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1	2	3	4	5	6	7	8	9
3	2	2	4	6	8	10	12	14	16	18
4	3	3	6	9	12	15	18	21	24	27
5	4	4	8	12	16	20	24	28	32	36
6	5	5	10	15	20	25	30	35	40	45
7	6	6	12	18	24	30	36	42	48	54
8	7	7	14	21	28	35	42	49	56	63
9	8	8	16	24	32	40	48	56	64	72
10	9	9	18	27	36	45	54	63	72	81

You should now have a familiar looking, correct multiplication table.

You can use the same process with any number of formulas. Any time you reference a cell/row/column in Excel you can use the dollar sign to lock onto that cell/row/column.

Linking to PowerPoint

If you've set up a graph or table in Excel, you can quickly link to that table in PowerPoint. Select the table or graph from Excel that you want in your presentation. In PowerPoint go to the paste special menu. On the left side you should see an option to paste a link. Use the Microsoft Excel Worksheet Option and press ok. Done!

You can break the links (press windows-e-k) if you wish to send the presentation out, but be careful to save the file as a different name after you break the links or you will have to relink everything.

Cautions:

- When you open your presentation it will now ask you if you want to update links. If you select, this upon opening, it will update all links in the presentation.
- If you have your PowerPoint on a different server than your Excel file, you may have some difficulties.

Moving Averages/Totals

Moving summaries can dynamically updating report tabs with the latest and greatest information.

Moving averages are good to use to determine what happened and if objectives were met. Don't get into using recent history to assess the near future. The best way to make predictions is to find out what causes the outcome is related to and then use predictive assessment tools.

The most difficult part of the moving summaries is finding the last row of your data. Once you find the last row of your data, everything else flows pretty smoothly. The way your data is recorded determines how you will do this.

Daily Entries

If you have single line per period of interest (e.g. one line per day) you can use a relatively simple process to find your last line of data.

```
=counta(A:A)
```

This will count all the rows that don't have a blank entry. It requires every row to have at least one column filled (so put a name in the header row). You can do this naturally with a date column or create an artificial column that has a value for every row (e.g. all 1s, or counting by 1s).

Once you know the total number of rows you can use that in an index function. The index function returns a cell value from a specific range in a specific row/column. The general format for the index function is below.

```
=index(array, row_num, column_num)
```

To find data in the last row in column C, you could use the formula:

```
=index(A:Z, counta(A:A), 3)
```

To find data in the next to last row, replace counta(A:A) with counta(A:A)-1

The vlookup formula will produce similar results but operates differently. Since the vlookup function can search for a string of text vertically, it can be used to find a date and report a number in that row. You will still need to use the index function to find the last row of data

Multiple Entries per Day

Having many entries per day removes the ability to know an exact row where your information resides. The task doesn't get much more difficult. In fact, you can use this method for the daily tally as well.

Set the sumifs command sum range to whichever column you are interested in, and then use the date for the criteria.

You can still make a moving average by using a combination of the counta and index functions. But since one day might not be entirely complete, you can also just use a user set cell. All the rest of the cells can be based off the first cell.

Pivot Tables

The Pivot Table is a powerful counting/sorting tool. I use it mostly for quick turn analysis. Standard products should be set up using the **ifs* commands.

The core of the pivot table is an advanced sorting machine. Say you took 2 decks of cards and shuffled them together then discarded a third. You could then sort them by suit one by one in the following categories:

Clubs	Diamonds	Hearts	Spades
14	19	16	20

You could further sort them into suit and face cards vs non-face cards as well as suit.

	Clubs	Diamonds	Hearts	Spades
Non-Face Cards	11	16	14	14
Face Cards	3	3	2	6

The pivot table quickly performs this type of counting operation and more. The pivot table treats each row of data as having the properties described in the column. If you had rows of data for the cards, they would look like this:

	A	B	C	D
1	Card #	Number on Card	Suit	Face Card
2	11	J	Clubs	Y
3	3	3	Clubs	N
4	10	10	Hearts	N
5	5	5	Diamonds	N
6	14	A	Diamonds	N
7	13	K	Clubs	Y
8	5	5	Hearts	N
9	12	Q	Spaids	Y
10	7	7	Diamonds	N
11	13	K	Hearts	Y
12	9	9	Clubs	N
13	11	J	Spaids	Y
14	10	10	Spaids	N
15	3	3	Spaids	N

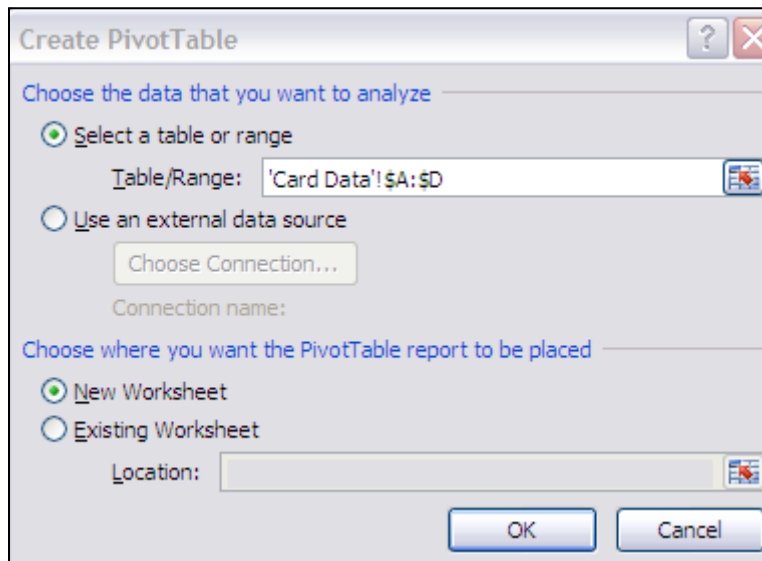
The pivot table will sort through each row of data and add one to the count of cells when it meets the specified criteria.

HOW TO:

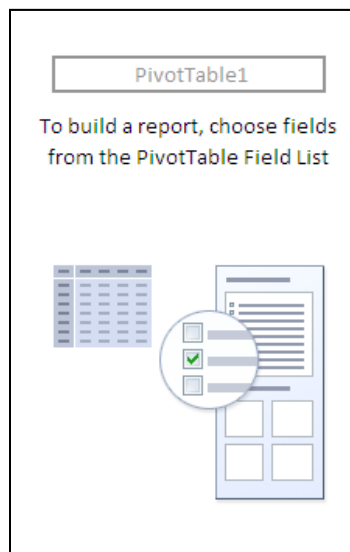
- 1) Setup Your Data:
 - a. Make sure your data tab has just data. It's not mandatory, but a best practice.
 - b. Each column MUST have a column header. Name it something intuitive.
- 2) Select Your Data:

To actually create a pivot table, select all of your data. You can select just the column letters unless you need to group by date quickly. To group by date you'll need one of the following options:

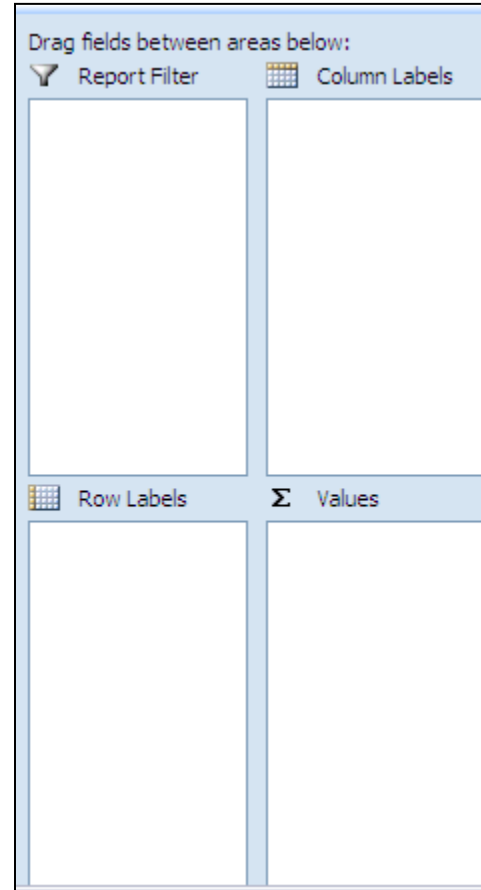
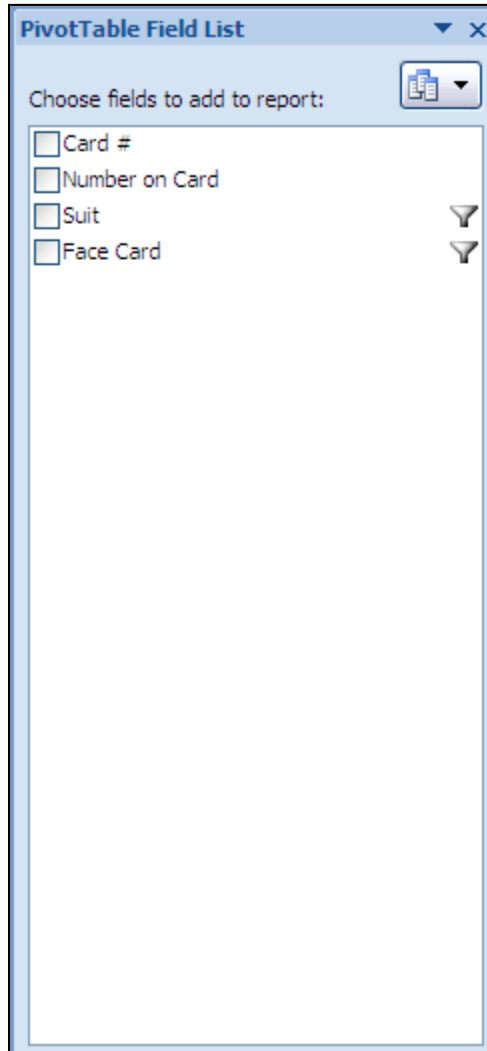
- a) Fastest – Select only the rows and columns with data.
 - b) Sustainable – Create additional columns in your data that compute the time period of interest. For example =Year(Date_Cell) or =Month(Date_Cell) or =Weeknum(Date_Cell+1)
 - c) Slowest - Manually group dates on the pivot table.
 - d) Most Complex – Create a dynamic named reference that uses the COUNTA function to determine the number of active rows and columns.
- 3) Go to the insert tab and press the insert pivot table in the upper left corner. If prompted for a location, don't put it in your data tab. Remember to keep your data tab clean.



- 4) Once you've inserted the pivot table, your screen should show a Pivot Table on the left and Field list on the right. You can drag the items in the Field List to the boxes below the Field List.
- a. Pivot Table – The pivot table is where all the computations show up.



- b. Pivot Table Field List – You can manipulate what is calculated by dragging items from the Field List to the Report Filter, Column Labels, Row Labels, and/or Values.



- 5) Drag the items you want to sort by into the row or column labels. When you do this the row/column labels should become the complete list of values in your data under that data column heading.

Example: If you drag Number on Card to the Row Labels, the rows for the pivot table will become all of the numbers and letters used on your cards in your data table.


- 6) Next drag an item from the field list into the values. If the item from the field list contains non-numeric data in the data table, the values will default to counting the number of occurrences. If the item contains numeric values in the data table, you can change the values to show the average, sum, count, or many other functions.

Other Pivot Table Actions:

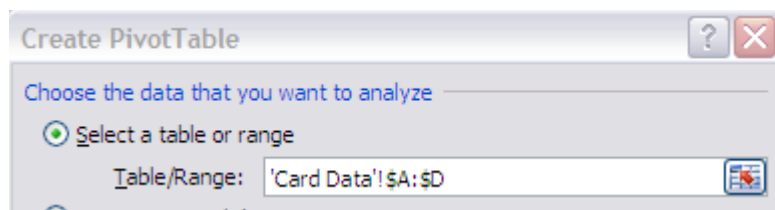
Grouping Row/Column Headers - You can right click on the row headers in the table or the column headers and auto-group the data (if it is numeric). If it is not numeric you can still group items, but you'll have to select the group of items and manually group each set.

% of Row/Column – Right click on the data in the pivot table. Click value field settings. Use the 'Show values as' tab. Use the drop down menu to indicate % of row or % of column.

Number Format – Right click on the data in the pivot table and select the change number format. You'll be able to change the format just like you could any other cell.

Filtering – You can filter the data or filter the headers. To filter a header select the header you want to filter. Look for and left click the filter symbol . You'll have the option to filter values or filter labels.

Refreshing the pivot table – If you've changed the data table that the pivot table is referencing (whether it's a data value or adding a whole new column), you'll be able to refresh the pivot table so that it reflects the new data. This is only true if you are changing (inserting/adding/changing) cells that were originally referenced in your pivot table range setup.



In the above pivot table setup, if you insert a column between column A and D it will be in the pivot table when you hit refresh. If you just start typing in Column E, it will be outside the bounds of the pivot table setup and it will not be included when you refresh the pivot table. To implement such changes you'll need to change the data source.

Visual Basic Code

Why Code?

In my opinion, we have to write code as analysts because data collection wasn't set up properly. I've seen a lot of data that was kept a daily tally in a spreadsheet day after day. It's not that horrible bad people do this, they just don't know any better. Analysts or anyone else can't quickly summarize information using such formats.

To get the data it's necessary to write code. Visual Basic Code for Applications (VBA) is relatively easy to catch onto. With some basic coding knowledge and the record macro feature, your requirement to be a good programmer is dramatically reduced. The record macro feature will record the actions you take in excel, word, and PowerPoint and translate those actions into

You'll likely have to write code to do something very repetitive. So the key components will be making a loop and rearranging data. Sometimes to rearrange data you will have to open a series of files or go from application to application (eg Excel to Word). Another critical component to getting code to assemble the right information is adding strings. Adding strings is not hard, but creative use can allow you to make file names and paths, define cell locations, and combine data.

When you start a macro you'll have the first two lines given to you.

```
Sub Macro_Name
```

```
End Sub
```

Everything needs to fit between those two lines for now. As you get more advanced, you can make additional modules to prevent repetitive coding. But we'll start simple for now. Make a loop inside your code. While there are many different types of loops, usually one type will suffice. In this case, we'll use the do/while loop.

```
Sub Macro_Name
```

```
Dim Counter as Integer
```

```
Counter=0
```

```
Do while Counter < 15
```

```
Counter = Counter + 1
```

```
Loop
```

```
End Sub
```

Notes:

- I'll add a dimension statement to define my variables anytime I add one. This is not always necessary, but it's a good practice.
- I've indented the code inside the loop. This helps you and others follow your work.

If you run this program now, you'll have counter = 0 outside the loop. You start the loop with the 'DO' statement, then counter becomes equal to itself (which is zero) and then add 1. The 'LOOP' statement returns to the 'DO' line as long as the criteria in the 'DO' line is not met. In this case, 1 is still less than 15, so the code repeats in the 'DO LOOP' until Counter is 15.

Now that you can loop, we'll do something in excel with the loop. We'll start by creating random numbers and putting those values in cells. To put the random numbers in we'll either need to assemble a text string or use the down arrow command.

```
Sub Macro_Name
```

```
'COMMENT: Purpose-Create 14 Random Numbers between 200 and 300
```

```
Dim Counter as Integer, RndNumber_Cell as String
```

```
'COMMENT **** Reset Counter
```

```
Counter=0
```

```
'COMMENT **** Start Loop, Goes 15 cycles
```

```

Do while Counter < 15
    Counter = Counter + 1
    'COMMENT **** RndNumber_Cell variable is created from the 'A' Column and
    the row specified by the Counter variable
    RndNumber_Cell = "A" & Counter
    'COMMENT **** Put a random number in the cell
    Range(RndNumber_Cell)=Int((300-200+1)*Rnd+200)
Loop
End Sub

```

Notes

- It's difficult at first to tell what the variables from the functions. I will try to keep this difference apparent by defining variables I create in the 'DIM' line.
- I've added comments to the code to start tracking what's going on
- The Range function is important, it allows you to specify which cell range you are dealing with. You can also use commands like Range(A1).Select, Range(A1).Copy, and Range(A1).Paste. Where the A1 refers to the cell.

Now we'll use the record macro feature and copy the resulting random numbers to a different location in the worksheet 100 times. We'll run into a problem if we try to use the counter to add 1 to the columns because the columns are letters. We could use some command to convert the letter to a number and then back to a number. Since that is complicated, we'll use the transpose function in excel to turn the column of data into a row of data.

Go to the developer tab click 'record macro' to get a dialog box for recording macros. Enter a name for your macro and hit 'OK'. The macro is now being recorded. Do the following:

- 1) Select the random numbers in Column A.
- 2) Hit cntrl-c for copy
- 3) Select cell A20
- 4) In the Home Tab go to the arrow under the paste button.
- 5) Select Paste Special
- 6) Click Values and Number formats
- 7) Check Transpose
- 8) Hit OK

Go back to your developer tab and hit stop recording macro. Select the macros button to the left of the stop recording macro button. Open the macro you just recorded by pressing the 'edit' button. You'll have some code ready for you. Excel just created VBA code for the steps you took. Your objective now is to find the parts that you need to use and incorporate them into your program. Fortunately the VBA code is fairly intuitive and reads as much like a sentence as you can expect of code. The following four lines will be valuable:

```
Range("A1:A15").Select
```

```

Selection.Copy
Range("A20").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,
Transpose:=True

```

Incorporate them into your previous code so that we can repeat the process 100 times, we'll need another loop.

```

Sub RandNumb()
    Dim Counter As Integer, RndNumber_Cell As String, PasteRow As Integer
    PasteRow = 20
    Do While PasteRow < 121
        PasteRow = PasteRow + 1
        PasteCell$ = "A" & PasteRow 'COMMENT**** assembles paste cell
        Counter = 0 'COMMENT **** Reset Counter
        Do While Counter < 15 'COMMENT **** Start Loop, Goes 15 cycles
            Counter = Counter + 1
            'COMMENT **** RndNumber_Cell variable is created from the 'A' Column and
            the row specified by the Counter variable
            RndNumber_Cell = "A" & Counter
            'COMMENT **** Put a random number in the cell
            Range(RndNumber_Cell) = Int((300 - 200 + 1) * Rnd + 200)

            Loop
            Range("A1:A15").Select 'COMMENT *** This is in the same place every time
            Selection.Copy
            Range(PasteCell$).Select
            Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,
            Transpose:=True
        Loop
    End Sub

```

NOTES:

- The outer loop became 'Do while PasteRow<121'. PasteRow did not have to be the criteria for the loop. We could have used another variable name and used 2 counters (one for PasteRow and one for the loop variable).
- The last 4 lines after loop were inserted. The Range().Select was modified to move one row down every iteration.

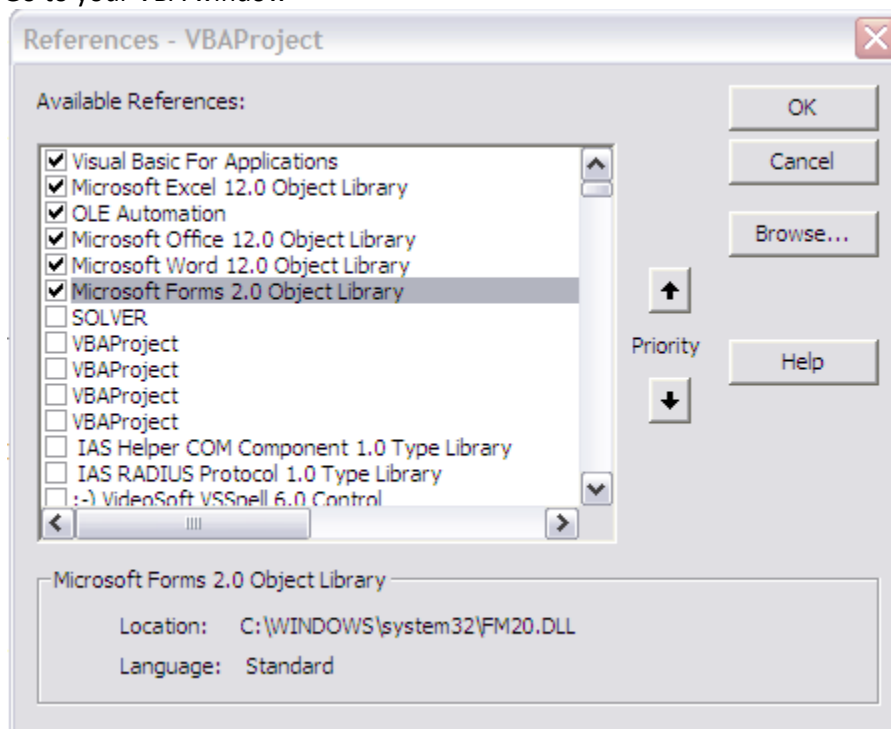
Open Excel Files from a Specified Directory

```
Sub Open_Excel_Files()  
    strFilePath = "C:\Documents and Settings\michael.cook\Desktop\Combat Analyst Training\  
    strCurFileName = Dir(strFilePath)  
    Do While strCurFileName <> ""  
        If strCurFileName Like "*.xls*" Then  
            Workbooks.Open strFilePath & strCurFileName  
            'COMMENT **** Code processing here.  
            ActiveWorkbook.Close 0  
        End If  
        strCurFileName = Dir  
    Loop  
End Sub
```

Open Document Files from a Specified Directory

Before you can run an excel macro that interfaces with word you need to add the right library to your excel vba window.

Go to your VBA window



Go to Tools>References

Make sure the correct Microsoft Word Object Library is included.

The following code should cycle through opening and closing word documents in the directory defined as *strFilePath*.


```

Sub Open_Word_Documents()
    Dim objWordApp As Object
    strFilePath = "C:\Documents and Settings\michael.cook\Desktop\Combat Analyst
    Training\Misreps\"
    strCurFileName = Dir(strFilePath)
    Set objWordApp = CreateObject("word.application")
    objWordApp.Visible = True

    Do While strCurFileName <> ""
        objWordApp.Documents.Open strFilePath & strCurFileName
        'COMMENT **** Code processing here. Use objWordApp to refer to word documents
        objWordApp.ActiveDocument.Close 0
    Loop

    objWordApp.Quit
    Set objWordApp = Nothing
End Sub

```

<http://stackoverflow.com/questions/831458/excel-vba-to-open-multiple-word-files-in-a-loop>

Notes:

- Excel VBA is using Word VBA. To get code for word, use the word VBA record macro feature.
- To tell the Word VBA to do anything, you'll need to refer to it's defined object name (e.g. objWordApp)

Debugging Tips

Use the F8 command to cycle through the lines of code.

As you use F8, mouse over variable names to find out what they are.

Rather than mouse over, you can go to the Debug menu and 'Add Watch'. Simply type in a variable name in the expression box.

Add a forced break by putting your cursor on a line of code and select Toggle Breakpoint (F9). This is a stop sign for your program.

Press F5 to run your program start to finish.

Experiment on copies of files, you never know what you can break.

Other Excel/VBA Functions

- LEN(Cell) - determines the length of a cell or string
- RIGHT(Cell,# Characters) – takes the specified number of characters from the right end of a text string (or cell)
- LEFT(Cell,# Characters) – takes the specified number of characters from the left end of a text string (or cell)
- MID(Cell, start position, # Characters) – returns the number of characters starting at the specified start position in the string (or cell).
- IFERROR(Cell with Value, Value if Error) – If the cell referred to contains an error, the IFERROR function can fill in any value you want.