Dynamically Visualizing Data in Excel

Tom Wilson

1 Introduction

In [Wil10], I mentioned some advantages and disadvantages of Microsoft Excel in a discussion about R:

> Excel's advantages are (1) familiarity to others, (2) an interactive nature, and (3) the ability to see a lot of numbers quickly. The interactive nature comes from using forms/controls to select data and to turn features on and off. Excel's disadvantages are (1) slowness in computing large spreadsheets, (2) the difficulty in creating some chart types (e.g., creating a boxplot), and (3) the changes that occur across releases and/or inconsistency across platforms (e.g., PC vs. Macintosh). It is certainly an acceptable analysis tool, especially considering advantage #1.

Some of these advantages and disadvantages may apply to any spreadsheet program. I thought I would expand on advantage #2. While my usage of Excel has been reduced due to my increasing usage of R, I did learn a lot about the tool, and I thought it would be good to share some of that knowledge for those that are frequently using it.

The idea of an interactive nature means that someone (e.g., your boss, co-worker, or customer) can interact with the worksheets and charts in order to see whatever it is that (s)he wants to see (assuming you provided the ability to do so). On many occasions, I have given a "presentation" using a spreadsheet rather than or in addition to slides. Think of this as an interactive demo.

I am assuming that the reader is very proficient at Excel, so I will not be explaining many basic concepts. I have never learned VBA[1], so you will not find any of that here. I do believe that several things that I will talk about can be done in VBA, and such implementations would better facilitate their usage. That is an exercise for the reader. I have not tried any other spreadsheet tool (e.g., OpenOffice.org Calc) to know how much of this information is applicable. It probably is not, but you will have to figure that out for yourself if you use an alternative program. Also, everything in this paper should work in the 2003 and 2007 versions of Excel. There may be minor differences in the details.

A tremendous amount of information concerning Excel charting exists at [Pel10], and I will not cover any of that material here. There are probably other helpful sites like the one referenced. In this paper, I will discuss how formatting cells, drawing objects, controls, multi-chart worksheets, and external worksheets aid in the visualization process.

2 Formatting Cells

The formatting of cells can be a good visual aid for a user when (s)he is interacting with a spreadsheet. I will discuss two techniques that are fairly straightforward: conditional formatting and dynamic measurement units.

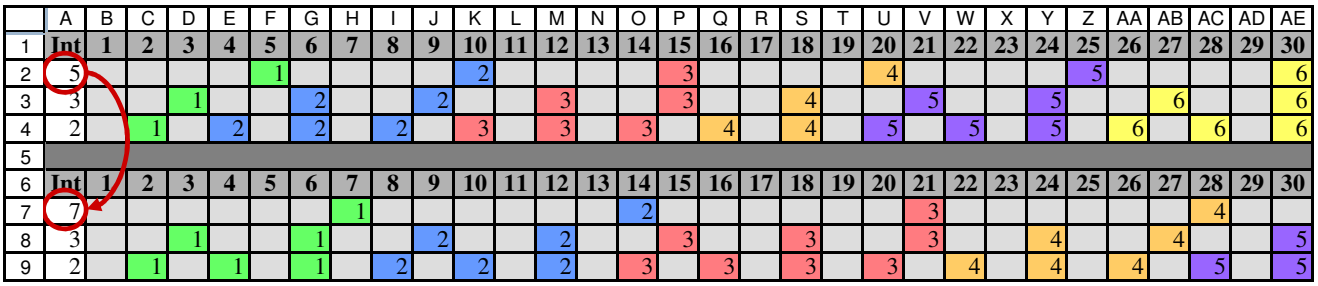
2.1 Conditional Formatting

Conditional formatting allows the formatting for a cell to change based upon several options, one being the cell's contents. Excel 2007 allows a lot more control and possibilities than previous versions.

Figure 1 shows an example of how conditional formatting can provide visualization of the data. In this example, the figure shows the *before* (rows 1-4) and *after* effect (rows 6-9) on the same worksheet. In reality, only rows 1-4 would exist. When cell `A2` is changed from 5 to 7, the other cells update their formatting.

Table 1 shows the formulas in some of the cells. The formulas in `B2:B4` are copied and pasted into `C2:AE4` and `B7:AE9`. The focus is not on these formulas and so I will not devote a discussion to their purpose. Besides the formulas that compute the cells' contents, conditional formatting formulas exist. In this case, there are 6 formulas. They all take the form: "`=MOD(B2,6)=x`" where `x`=1...6. For each formula, a pattern is assigned: 1=green, 2=blue,

[1] VBA = Visual Basic for Applications



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	**Int**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
2	5					1					2					3					4					5					6
3	3			1			2			2			3			3			4			5			5			6			6
4	2		1		2		2		2		3		3		3		4		4		5		5		5		6		6		6
5																															
6	**Int**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
7	7							1							2							3							4		
8	3			1			1			2			2			3			3			3			4			4			5
9	2		1		1		1		2		2		2		3		3		3		3		4		4		4		5		5

Figure 1: This worksheet shows the *before* and *after* effect when conditional formatting is used on worksheet cells.

3=red, 4=orange, 5=purple, and 6=yellow[2]. The formatting for `B2` is copied and pasted to every other cell besides the column headers and the row data entry column (`A`).

Table 1: Cell Formulas

Cell	Formula
B2	`=IF(MOD(B$1,$A2)=0,CEILING(B$1/$A2,1),"")`
B3	`=IF(MOD(B$1,$A3)=0,CEILING(CEILING(B$1/$A3,1)*$A3/$A2,1),"")`
B4	`=IF(MOD(B$1,$A4)=0,CEILING(MAX($A4*CEILING(B$1/$A4,1), $A3*CEILING($A4*CEILING(B$1/$A4,1)/$A3,1))/$A2,1),"")`

When the values in column `A` are changed, the formatting updates appropriately. This example is beyond the limits of Excel 2003, where only 3 formats can be defined. But the concept is the same.

Many visual effects are possible using this facility. It is especially useful in situations where the user has many cells that (s)he must interact with. Otherwise, a chart with controls would suffice (reference Section 4).

2.2 Dynamic Measurement Units

One of the pains in spreadsheet modeling is dealing with measurements with large differences in magnitude (e.g., 1 mile and 1 yard). You can assume a desirable unit, but then be annoyed when you have data that goes against the assumption. You would like dynamically changing units. The forthcoming technique can be used with any kind of measurement (e.g., distances, times, or weights), regardless of the number of measures or the ratio between measures, since it is simply a table lookup.

Figure 2 shows an example worksheet for memory sizes. Each row shows a progressively larger unit (column `B`) and the equivalent number of bytes (column `A`). Cell `A11` shows a raw number and cell `B11` shows the size in a convenient unit (here, MB). Cell `B8` controls how many digits of precision there are in the converted number.

	A	B
1	**Bytes**	**Unit**
2	1	B
3	1,024	kB
4	1,048,576	MB
5	1,073,741,824	GB
6	1,099,511,627,776	TB
7		
8	Digits of Precision	2
9		
10	**Raw Size**	**Size**
11	65,536,000	62.50 MB

Figure 2: This worksheet demonstrates dynamic memory-units.

Table 4 lists the named ranges and formulas defined to make this facility work. The last three names refer to the

[2]This notation is for the paper and is not used in Excel.

size list, units list, and the number of digits parameter. The "left_cell" name refers to a formula. This formula is a relative reference to the cell to the left. That reference is relative to the cell containing this name. The "mem_conv" name refers to a long formula. In this paper, the formula contains line breaks to make it easier to read. This formulas references the other names already described.

Table 2: Named Ranges and Formulas

Name	Refers To
left_cell	=INDIRECT("RC[-1]",FALSE)
mem_conv	=IF(ISNUMBER(left_cell),IF(left_cell>0,TEXT(left_cell/ LOOKUP(left_cell,mem_size_list),"0"&IF(num_mem_digits>0,"."& REPT("0",num_mem_digits),""))&" "& LOOKUP(left_cell,mem_size_list,mem_units_list),0),"")
mem_size_list	=$A$2:$A$6
mem_units_list	=$B$2:$B$6
num_mem_digits	=$B$8

Cell `B11` contains "`=mem_conv`" and references cell `A11` (the cell left of it). When `A11` is changed, `B11` updates. The drawback of this implementation is the need for the adjacent cell. The implementation can be modified to reference other cells (e.g., the one above). This facility could be improved if the format of `A11` could be changed to result in `B11` without requiring the additional cell. Perhaps this is possible using VBA. The same concept can be used for distances, times, or any other measurement where a convenient unit is desired.

Figure 3 shows a worksheet that illustrates dynamic measurement units and conditional formatting. Column `C` is normally hidden and stores the number of bytes. The visible columns to the right contain the "`=mem_conv`" formula. Conditional formatting looks at the units and appropriately formats the cell. Cells with values ending in "GB" are shaded blue; those ending in "TB" are shaded yellow. This makes the consumption of the data in the table much quicker.

	A	B	C	D	F	H	J	L	N	O
1					Summary					
2				Average Size			Maximum Size			
3		Year	Trans DB		Reports DB	Total	Trans DB	Reports DB	Total	
4		2011	2,913,316,098	2.71 GB	23.45 GB	26.16 GB	23.72 GB	110.48 GB	134.21 GB	
5		2012	12,243,968,000	11.40 GB	92.97 GB	104.37 GB	113.40 GB	620.16 GB	733.56 GB	
6		2013	29,662,624,000	27.63 GB	234.34 GB	261.96 GB	288.68 GB	1.54 TB	1.82 TB	
7		2014	84,059,480,000	78.29 GB	583.43 GB	661.72 GB	637.09 GB	3.58 TB	4.20 TB	
8		2015	129,750,336,000	120.84 GB	1.06 TB	1.18 TB	1008.36 GB	6.58 TB	7.56 TB	
9										

Figure 3: This worksheet illustrates dynamic measurement units and conditional formatting.

3 Using Drawing Objects

Excel allows drawing objects to be placed on a worksheet. A drawing object has properties that must be properly set. An object can (1) move or not move and (2) size or not size with the cell that it is associated with. Any particular setting may be desirable.

3.1 Workflow Modeling

Figure 4 shows a worksheet with a workflow model. This worksheet contains several drawing shapes (e.g., rectangles, trapezoids) and lines/connectors. Column widths have been sized to match the smaller drawing shapes' widths, but that was just a desire. Everything else is done via cell formatting.

Figure 5 shows a specific stage of the overall workflow. Here, connecting workflow (e.g., Nat2, WF Bureau) is shown as well as steps in the "National Search" workflow. Next to the drawing shapes are blue and yellow cells. The blue cells compute the number of messages per hour. The yellow cells compute the total sizes of the messages in kilobytes (we could have used dynamic measurement units here). To the right of the workflow, messages are

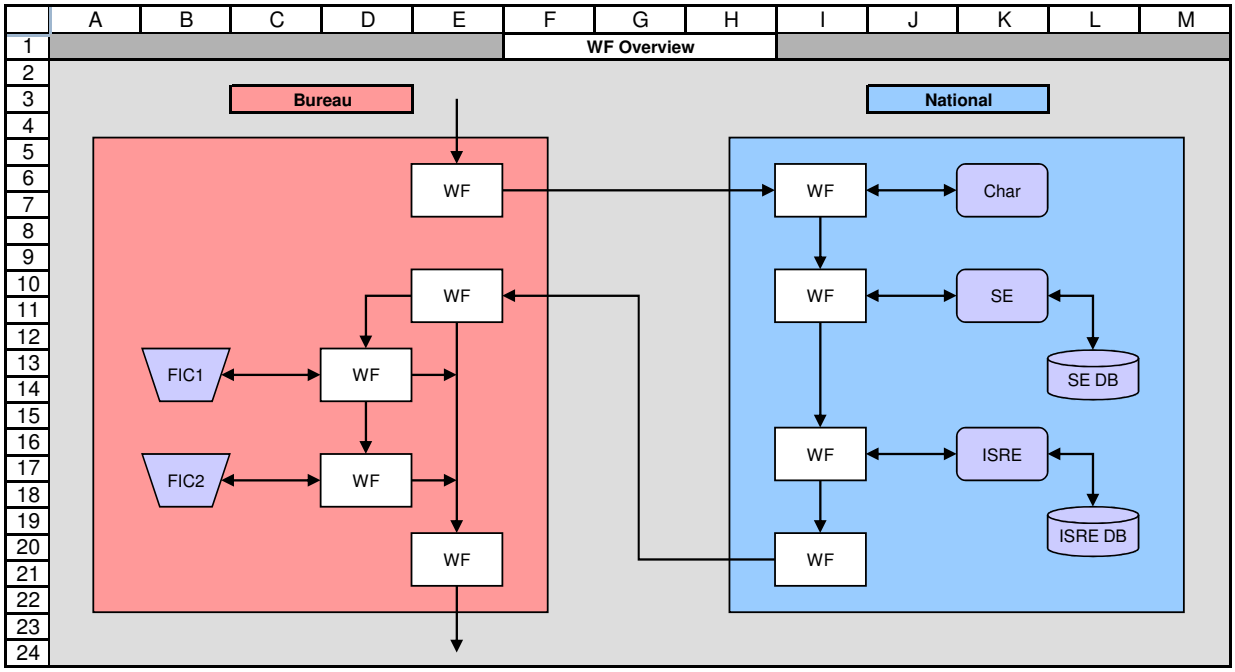


Figure 4: This worksheet demonstrates a simple workflow using drawing objects.

decomposed by type (i.e., by column). Cells with a bold border are data entry cells. All others contain formulas. Each row in this area corresponds to the cell in the same row of the workflow area.

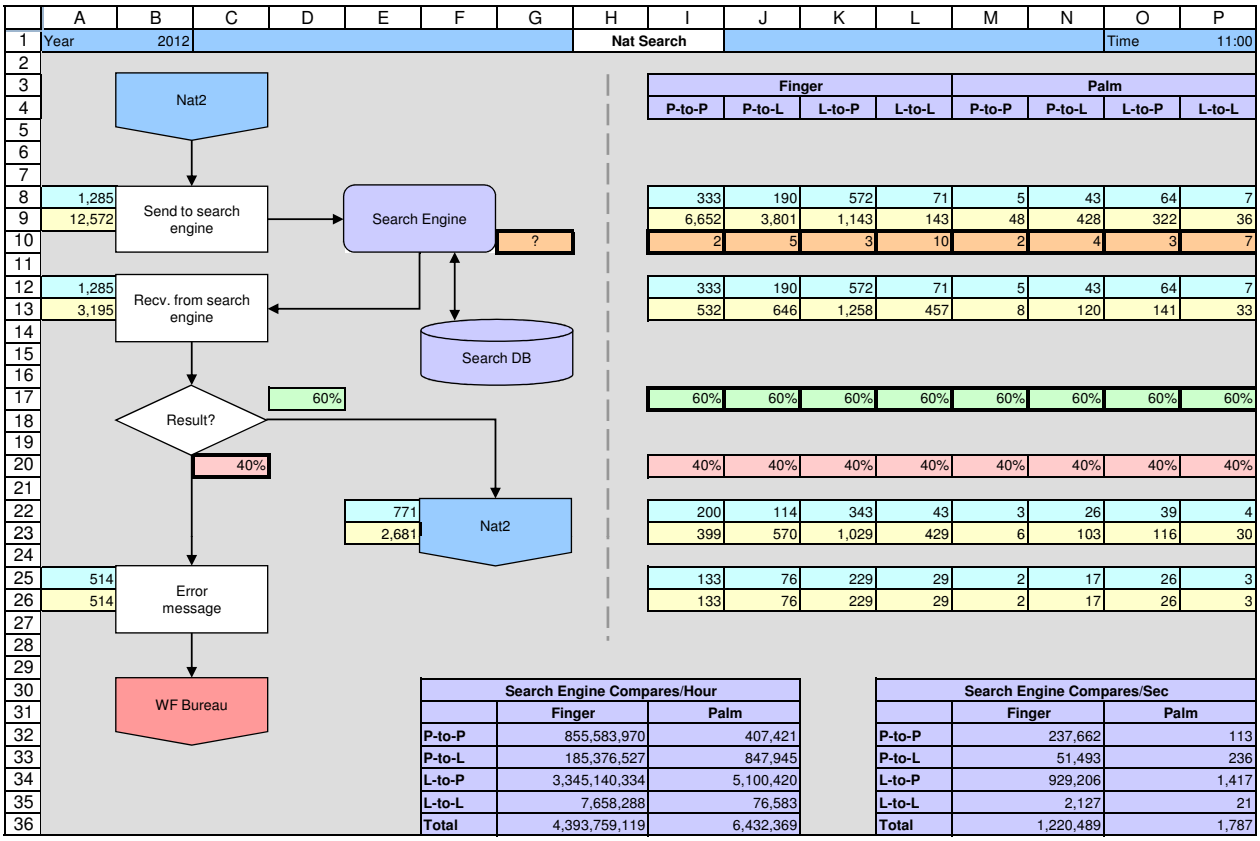


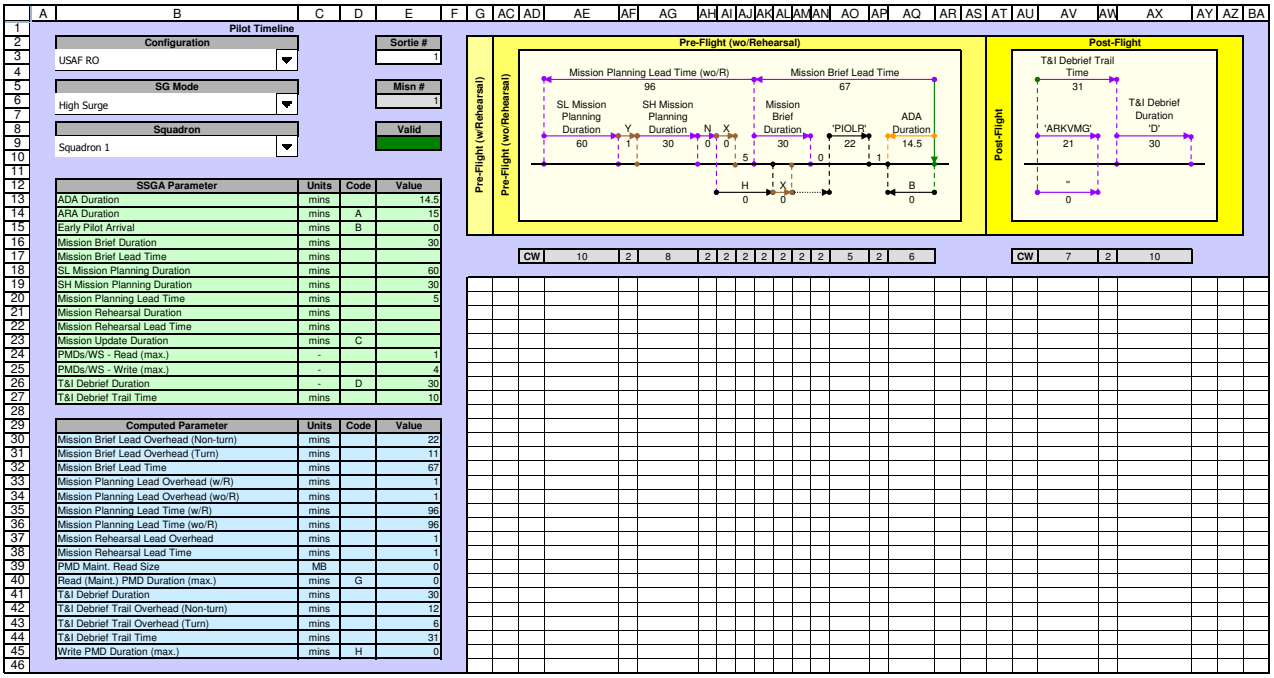
Figure 5: This worksheet demonstrates a complex workflow using drawing objects and common worksheet formulas.

The real model contains more than a dozen worksheets, many of which contain drawings. A drawing shape can contain a reference to a cell, in which case the cell's contents are shown in the shape. As the cell changes, so does the text in the drawing shape. This facility was not used here.

3.2 Simulating Timelines

Microsoft Project has the ability to make timelines (or schedules), but it lacks most of the features available in Excel (although you can embed a spreadsheet in a project in order to leverage some of the spreadsheet features). The timelines dynamically size, but diagrams tend to become very vertical due to the layout of one item per row. Timelines can be created in Excel, but they will not dynamically size themselves when they are drawing objects on a worksheet.

Figure 6 shows a timeline for the workflow of a pilot. The timeline consists of many line drawing objects. Above each horizontal line is a cell with either text or a sequence of letters (e.g., "PIOLR"), each letter referring to a parameter on the worksheet. Below each horizontal line is a cell with a formula computing the duration of the tasks.



SSGA Parameter	Units	Code	Value
ADA Duration	mins		14.5
ARA Duration	mins	A	15
Early Pilot Arrival	mins	B	0
Mission Brief Duration	mins		30
Mission Brief Lead Time	mins		
SL Mission Planning Duration	mins		60
SH Mission Planning Duration	mins		30
Mission Planning Lead Time	mins		5
Mission Rehearsal Duration	mins		
Mission Rehearsal Lead Time	mins		
Mission Update Duration	mins	C	
PMDs/WS - Read (max.)	-		1
PMDs/WS - Write (max.)	-		4
T&I Debrief Duration	-	D	30
T&I Debrief Trail Time	mins		10

Computed Parameter	Units	Code	Value
Mission Brief Lead Overhead (Non-turn)	mins		22
Mission Brief Lead Overhead (Turn)	mins		11
Mission Brief Lead Time	mins		67
Mission Planning Lead Overhead (w/R)	mins		1
Mission Planning Lead Overhead (wo/R)	mins		1
Mission Planning Lead Time (w/R)	mins		96
Mission Planning Lead Time (wo/R)	mins		96
Mission Rehearsal Lead Overhead	mins		1
Mission Rehearsal Lead Time	mins		1
PMD Maint. Read Size	MB		0
Read (Maint.) PMD Duration (max.)	mins	G	0
T&I Debrief Duration	mins		30
T&I Debrief Trail Overhead (Non-turn)	mins		12
T&I Debrief Trail Overhead (Turn)	mins		6
T&I Debrief Trail Time	mins		31
Write PMD Duration (max.)	mins	H	0

Figure 6: This worksheet excerpt demonstrates a timeline using drawing objects and common worksheet formulas. Many rows containing parameters have been omitted.

We would like the horizontal lines to be sized according to the values beneath. Obviously, the whole column needs to be similarly sized. Excel will not do that automatically, but we can do it manually. First, all lines need to have the properties set to "move and size with cell". Next, we need to compute the column width. In the figure, cells in row 17 do that. Note that we can put many lines in the same column, but the column can only have one width. Then, we must manually set the column width. The result is a timeline that "looks" right. VBA might be able to do this work for you. The automatic sizing is possible on a chart (reference the discussion concerning Figure 7) although significant work is necessary to make relationships between the lines.

4 Using Controls

There are two types of controls: (1) a form, which does not require VBA or macros, and (2) ActiveX, which does. Of course, I am going to avoid VBA and discuss forms. The concepts should apply equally to both types. Controls (i.e., forms) allow the user to make choices, which result in cells being changed. These cells are referenced by formulas and cause what is being viewed to change. Controls may be placed on worksheets and charts. A control on a chart must reference a cell on a worksheet. Several controls (e.g., multiple menus) can reference the same cell. When one changes the cell, all related controls update.

Figure 7 shows a chart with numerous forms: 8 pull-down menus, one checkbox, and 11 labels (i.e., text fields). All forms are above the chart; only the chart title is not a form. Each menu or checkbox updates a cell, which in turn impacts what is displayed on the chart. The text field containing "11:30" is linked to a cell that has a formula that combines the hour and minute menu choices. This may seem like overkill, but when the "Yesterday" checkbox is checked, the time becomes "(11:30)" indicating 11:30 yesterday. Excel does not display negative times, but that is a completely different issue; putting the time in parentheses is my solution.

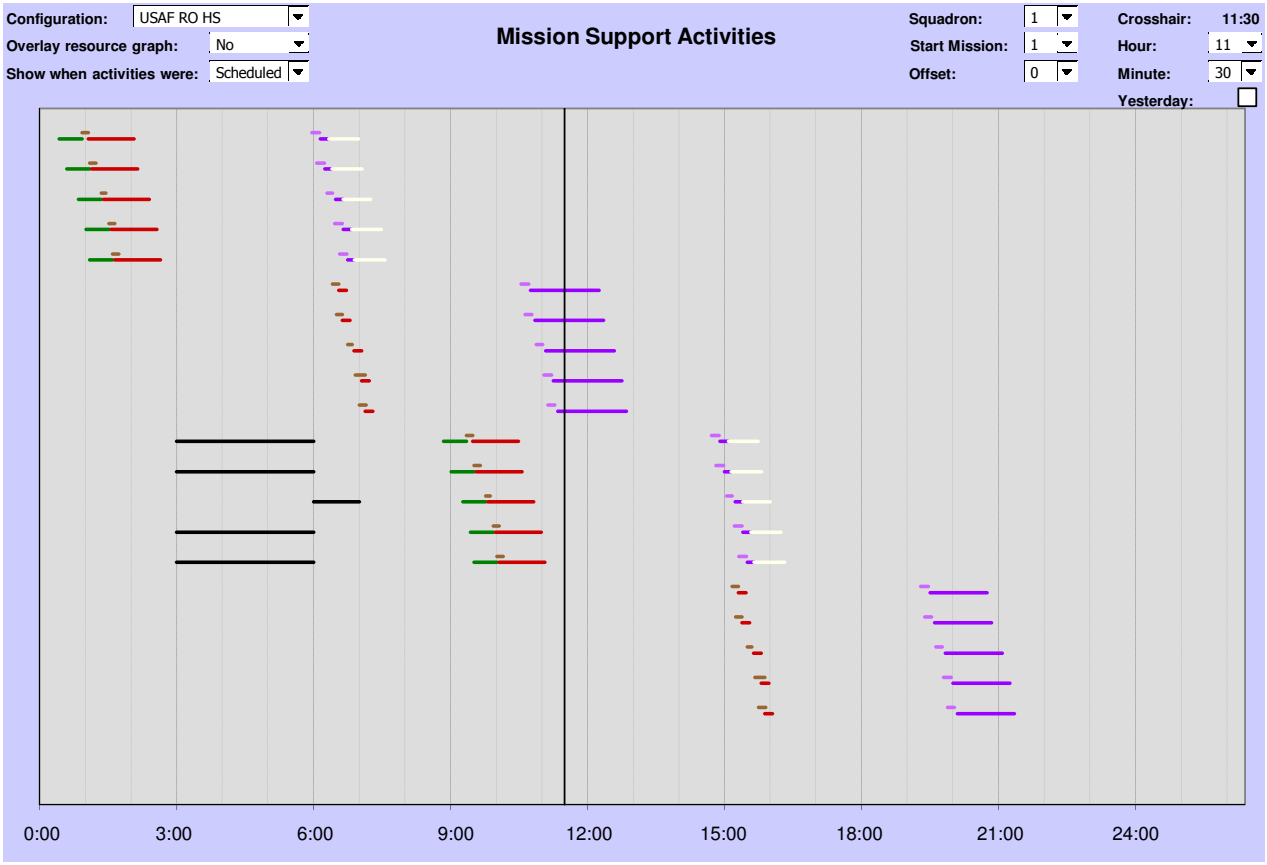


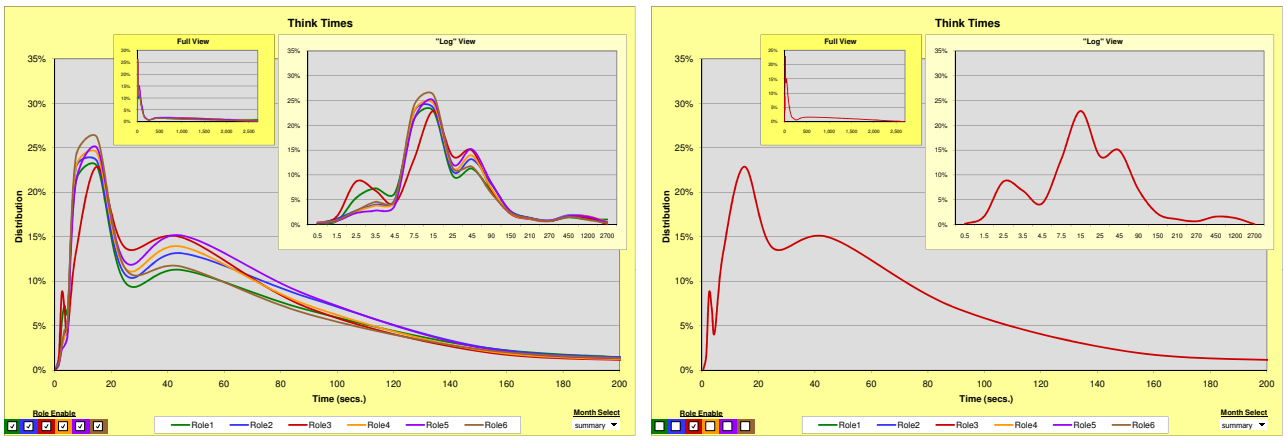
Figure 7: This figure demonstrates the usage of forms on a chart.

As an aside, Figure 7 is a graphical timeline similar to that discussed in Section 3.2. Here, the lines do dynamically size. Excel does not directly support this type of chart; it is an XY scatter chart with a significant effort to create all of the lines (using a couple of hundred series). One interactive aspect cannot be illustrated in the figure, but is helpful to the user. Each series can be named. Putting the cursor above the line will display its name in a "help balloon". In this model, the name specifies what interval the line describes and also contains the start and end times of the interval. Also, lines that are likely to overlap are offset on the y-axis so that the end points can be distinguished.

Figure 8a shows a chart with two embedded/overlaid charts. The two smaller charts show the same data with different x-axes. The checkboxes in the lower left allow data to be included or excluded. Figure 8b shows the result of unchecking several boxes. The menu in the lower right controls which data set is shown.

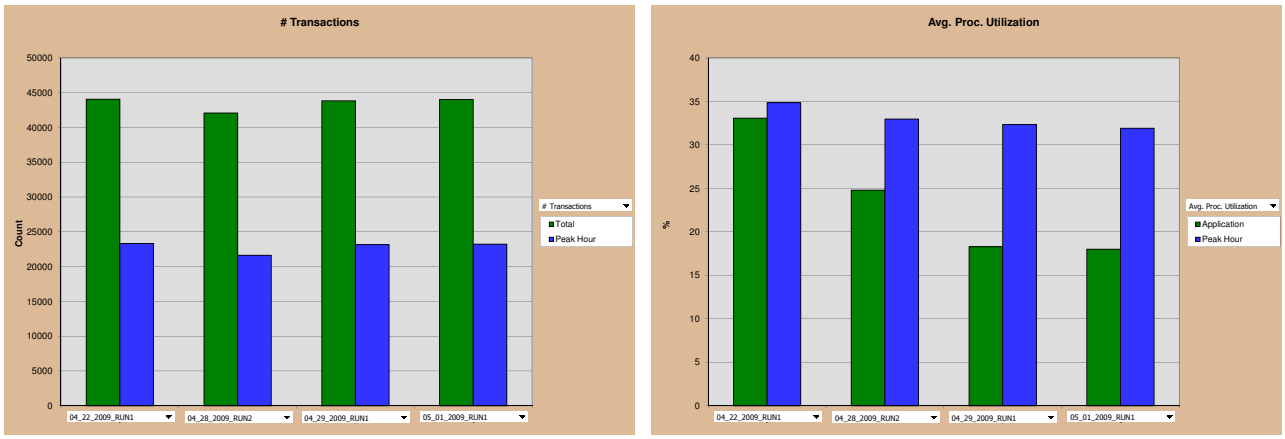
Figure 9a shows a bar chart which allows visual comparison of 4 performance tests. The tests are selected by menus below each pair of bars. The menu above the legend selects what data to display. This figure shows "# Transactions". Figure 9b shows the changes made when the "Avg. Proc. Utilization" choice is made. Notice that the chart title and y-axis title and labels have changed. The titles are linked to cells which update when the menu is changed. The y-axis labels update when the associated range of cells changes. Care must be taken when the y-axis can take on different formats (e.g., integers, reals, percentages).

Controls give the spreadsheet viewer the ability to dynamically "play" with the data and do his own analysis on-the-fly. This facility can be used with the technique shown in Section 6 to choose external worksheets.



(a) **(b)**

Figure 8: This figure demonstrates the usage of multiple charts on a chart. Forms are used to control the contents. Chart (a) has all checkboxes checked, while chart (b) has several checkboxes unchecked.



(a) **(b)**

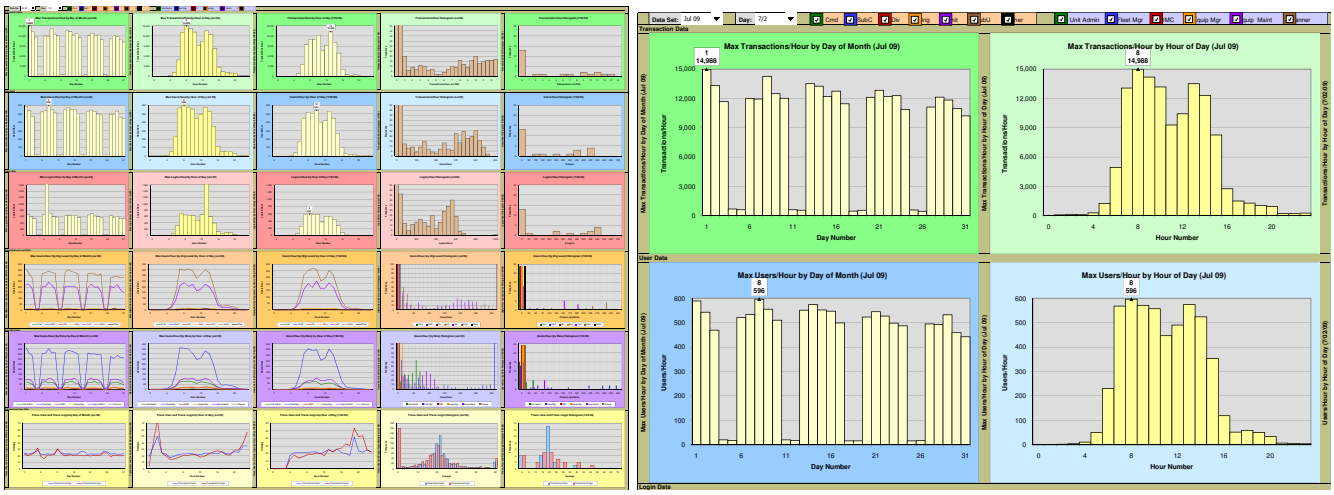
Figure 9: This figure shows a chart where forms are used to control the content. Chart (a) shows one type of data, while chart (b) shows another.

5 Multi-chart Worksheets

Sometimes viewing multiple charts is a good idea. Besides putting multiple charts on a chart (reference Figure 8), multiple charts can be put on a worksheet. Outlining and freezing panes can improve the usability.

Figure 10a shows a worksheet with 30 embedded charts. Normally, the user would see about 4 at a time (like Figure 10b). The worksheet can be scrolled, but this is somewhat tedious. The top row contains controls, so we will want to freeze it, so that it does not move when we scroll down. This does not solve the problem of the controls disappearing when we scroll right. To solve both problems, we will add outlining to the worksheet.

Each chart is the same size and has its properties set to “Move and size with cells”. Note that the controls are set to “Don’t move or size with cells”. Each chart is separated by a row and a column. Outlining is added so that sets of rows or columns can be collapsed, resulting in the charts disappearing. Now, the user can look at many (but not all) combinations of the charts at the same time. If you have a lot of data to show, use this technique and your imagination.



(a) **(b)**

Figure 10: This figure shows a worksheet with 30 embedded charts. Chart (a) is a zoomed-out view. Chart (b) is a typical view. Outlining is used to control which charts are visible.

6 Referencing External Worksheets

You may encounter either of these situations: You have a lot of data, but do not want to share all of it with everyone; or, you have too much data and only want to look at some of it. What you would like to do is have a worksheet for each performance test and then decide which tests you want to include. Moreover, you would like to have some of these worksheets in separate spreadsheets (or workbooks).

Figure 11 illustrates the desired concept. We have a "root" spreadsheet with our chart and possibly some other worksheets (e.g., "example"). We also have some other spreadsheets with one or more worksheets. In this example, we show two files: nov_10.xlsx and oct_10.xls. All of the worksheets containing the data that we want to chart must have the same structure. We must open the root spreadsheet, but we will only open the other spreadsheets when we want to reference them.

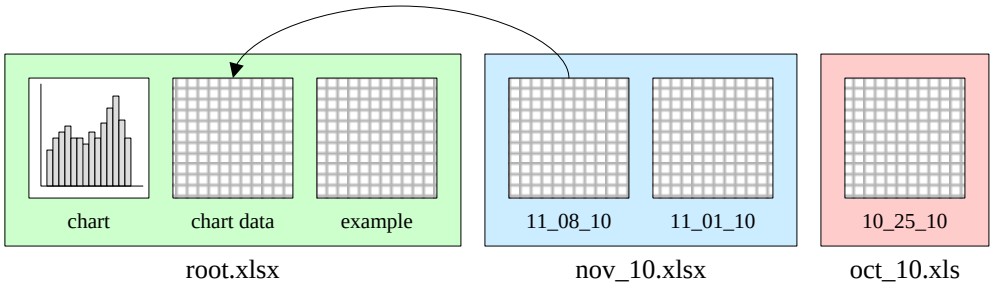


Figure 11: This figure illustrates the concept of referencing external worksheets.

It is best to create another worksheet in the root spreadsheet; we will call this worksheet "work". Figure 12 shows the possible contents of this worksheet. Cells `A4:D8` contain a table with the data supporting Figure 11. To support more worksheets, make the table longer. The "Filename" column lists the filenames. Make an entry blank to reference a worksheet in the root spreadsheet. The "Worksheet" column lists the worksheet names. The "Constructed Address" column contains formulas that create a valid Excel worksheet address. The formula in cell `C5` is in Table 3. The same formula is copied and pasted to each cell in the same column.

The "Menu Name" column contains formulas that copy cells from the referenced worksheets (see Table 3). This formula determines if the constructed address is valid. If it is, it copies the contents of `A1` on that worksheet to here. If it is not, the text "<unavailable>" is printed. In this case, we are assuming that only the "nov_10.xlsx" file is open. Any text can be put in a worksheet's `A1` cell. The intent is that this text will appear in a pull-down menu in

	A	B	C	D
1		**Menu Choice**	**Selected Sheet**	**Valid Choice**
2		1	[nov_10.xlsx]11_08_10	TRUE
3				
4	**Filename**	**Worksheet**	**Constructed Address**	**Menu Name**
5	nov_10.xlsx	11_08_10	[nov_10.xlsx]11_08_10	11/08/10
6	nov_10.xlsx	11_01_10	[nov_10.xlsx]11_01_10	11/01/10
7	oct_10.xls	10_25_10	[oct_10.xls]10_25_10	<unavailable>
8		example	example	Example

Figure 12: This worksheet excerpt illustrates the external worksheet configuration data.

order to choose the worksheet's data. Note that the formula in `D5` assumes Excel 2007.[3]

Table 3: Cell Formulas

Cell	Formula
C2	`=INDEX(sheet_addr_list,B2))`
D2	`=ISREF(INDIRECT(ADDRESS(1,1,1,TRUE,selected_sheet)))`
C5	`=IF(A5="",B5,CONCATENATE("[",A5,"]",B5))`
D5	`=IFERROR(INDIRECT(ADDRESS(1,1,1,TRUE,C5)),"<unavailable>")`

The upper part of the worksheet contains cells that control what is selected and copied for charting. The formulas for cells `C2` and `D2` are in Table 3. These formulas contain named ranges, which are shown in Table 4. If you want more worksheets, the named range "sheet_addr_list" should be updated. A pull-down menu, which is not shown anywhere, controls the contents of cell `B2`. When a choice is made, `C2` and `D2` are updated.

Table 4: Named Ranges and Formulas

Name	Refers To
sheet_addr_list	=$C$5:$C$8
selected_sheet	=$C$2
valid_choice	=$D$2
sheet_copy	=INDIRECT(ADDRESS(ROW(),COLUMN(),1,TRUE,selected_sheet))
sc	=IF(valid_choice,IF(sheet_copy="","",sheet_copy),"")

A worksheet in the root spreadsheet contains formulas to copy the data from the source spreadsheet. In this example, the source sheet is "[nov_10.xlsx]11_08_10". The target worksheet is called "chart data". The structure of this worksheet is the same as the source worksheets. Each cell that should contain data will contain the formula "`=sc`". That "sc" is a named formula in Table 4. If the referenced sheet is valid, the appropriate cells on te source worksheet are copied. If the sheet is not available, empty cells are shown (rather than errors). The "sc" formula checks for empty cells before copying, otherwise they would appear as 0's. The rest of the root spreadsheet references "chart data" and does not care which worksheet is referenced. Depending on your needs, deliver the root spreadsheet and any data spreadsheets as appropriate.

When a referenced spreadsheet is opened, the root spreadsheet will not know it. A calculation must be forced. When a referenced spreadsheet is closed, the root spreadsheet will know it.

7 Conclusion

Several diverse techniques were presented for dynamically visualizing data in Excel. Now that you know the basics, you can apply your other Excel knowledge to enhanced what you have learned. Perhaps you will create new techniques or VBA equivalents of those described here.

[3]For Excel 2003 use: `=IF(ISREF(INDIRECT(ADDRESS(1,1,1,TRUE,C5))),INDIRECT(ADDRESS(1,1,1,TRUE,C5)),"<unavailable>")`.

References

[Pel10] Jon Peltier. "Microsoft Excel Tips and Tricks". `http://peltiertech.com/Excel`, 2010.

[Wil10] Tom Wilson. "Workload Correlation and Visualization". *CMG '10 International Conference*, 2010.