

Visually Effective Excel Dashboards

CREATED BY LEILA GHARANI



XelPlus.com

XelPlus.com

COPYRIGHT © 2018 LEILA GHARANI

How to Use the Course Notes

This detailed course notes/script is accompanying documentation for my Visually Effective Excel Dashboards course on XelPlus.com/courses.

The following scripts give you a step-by-step guide on how to create an interactive KPI dashboard from scratch as covered in Modules 8 through 13. We will also create an interactive Slicer dashboard as covered in Modules 14 through 17.

Because the module's results build upon one another (i.e. the results of Module 8.2 serve as the starting point for Module 8.3), if you wish to start at a midpoint within the construction of a dashboard, you can open the Excel file for a module and all the previous work will be completed up to that point in the training.

The course files are arranged by module (*i.e. Module 8*) and then by module objective (*i.e. 8.1, 8.2, 8.2, etc...*). The name of the Excel file you need to work with for each module objective is mentioned at the start of the objective.

DISCLAIMER

We (XelPlus e.U.) has taken every effort to ensure the accuracy of this manual. In case you discover any discrepancies, please send us a quick email to: info@XelPlus.com.

Table of Contents

Modules 8-13 KPI Dashboard.....	1
KPI Dashboard Setup	1
Module 8.1 – KPI Dashboard Overview.....	2
Module 8.2 – Dashboard Wireframe	4
Module 8.3 – Spreadsheet Setup	5
Module 8.4 – Setup of Raw Data Sheet	6
Module 8.5 – Setup of Calculation Sheet.....	8
Scrolling Table and Bars.....	11
Module 9.1 – Setup a Combo Box to Ease Selection	12
Module 9.2 – Complex Lookup with INDEX and MATCH	15
Module 9.3 – Scrolling Data Table.....	18
Module 9.4 – Conditionally Formatted Data Bars.....	21
Module 9.5 – Arrows for Deviation.....	24
Module 9.6 – Formatting of Scrolling Data Table	27
Dynamic Headers.....	29
Module 10.1 – Conditional Comments in Header.....	30
Module 10.2 – Adding Shapes and Textboxes	32
Top/Bottom KPI Based on Selection.....	35
Module 11.1 – Show Top 6 Matches with Unique Identifier	36
Module 11.2 – List Box to Select KPI for Top 6 Matches.....	40
Module 11.3 – Option Button to Toggle Between Top or Bottom 6	42
Module 11.4 – Actual to Budget Comparison – Sorted Bars with Error Bar Technique	45
Module 11.5 – Finalize Top/Bottom Report on Dashboard.....	50
Scrolling Line Chart with Boxes	53
Module 12.1 – Complex Lookup to Prepare Data for Scrolling Chart.....	54
Module 12.2 – Scroll-able Line Chart	57
Module 12.3 – Trick to Avoid Crashing Lines in Line Chart.....	59
Module 12.4 – Check Box to Control Series Visibility (Hide or Show)	62
Module 12.5 – Line Series Tweaks for Better Readability.....	64

Module 12.6 – Finalize Profit Comparison Section on Dashboard	70
<i>Final Touches (KPI Dashboard).....</i>	73
Module 13.1 – Restrict Dashboard Area and Format for Printing	74
Module 13.2 – Dashboard Protection, Hyperlinks and Final Touches	76
<i>Modules 14-17 Regional Dashboard (Pivot Based).....</i>	81
<i>Regional Dashboard Setup.....</i>	81
Module 14.1 – Pivot Slicers as Buttons (also without Pivot Tables)	82
Module 14.2 – Custom Column Chart Controlled by Slicer Buttons.....	91
Module 14.3 – Slicer Custom Sorting & Design.....	97
Module 14.4 – Check to Ensure Single Slicer Button is Selected	101
Module 14.5 – Dynamic Comments with Slicer Selection	102
<i>Advanced Pivot Table Techniques</i>	107
Module 15.1 – Variances in Pivot Table	108
Module 15.2 – Adding Conditional Formatting in Pivot Table.....	113
Module 15.3 – Pivot Table as Part of Dashboard Report.....	117
Module 15.4 – Pivot Charts for Dashboards	119
Module 15.5 – VBA Code to Automatically Refresh Pivot Table.....	122
<i>Impressive Charts Using Shapes.....</i>	127
Module 16.1 – Adding Shapes to (Pivot) Charts	128
Module 16.2 – Overcoming Pivot Chart Limitations	130
<i>Secrets for Impressive Reports</i>	137
Module 17.1 – Camera for Improved Design	138
Module 17.2 – Lookups on Pictures	144
Module 17.3 – Final Touches.....	148
<i>Free Excel Tutorials.....</i>	153
<i>More Resources</i>	154

Modules 8-13

KPI Dashboard

KPI Dashboard Setup

MODULE 8.1 – KPI DASHBOARD OVERVIEW

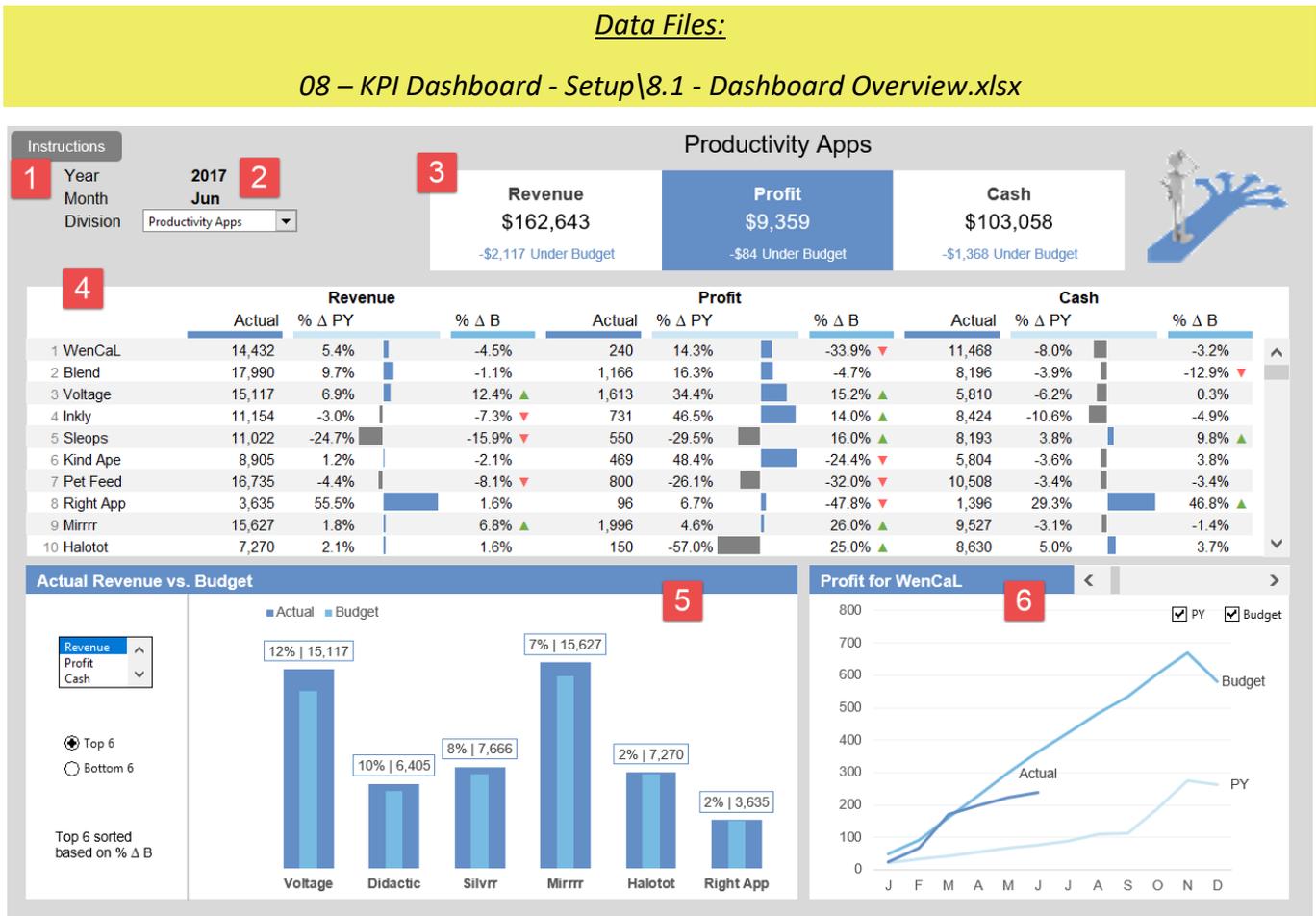


Figure 8.1 - KPI Dashboard (Solution)

Modules 8 through 13 will be devoted to the development of the KPI dashboard displayed above. We will utilize many of the skills visited in the previous modules of our online training as well as learning new skills to solve previously unencountered problems.

The major functional areas of the KPI dashboard are as follows:

1. Instructions for use and version control are provided via an instructions button. This is controlled by a hyperlink to a different sheet.
2. Three input cells to allow the user to define the year, month, and Division for focused analysis. The Division dropdown is provided by a combo box form control. These selections will control the behavior of ALL the other dashboard objects.
3. Three KPIs to display the total revenue, profit, and cash. These KPIs will also display the amount that the KPI is over or under relative to the Budget as well as a text display to over/under status.

4. Scrolling table of apps based on Division selection that displays revenue, profit and cash broken down into Actual, percent change from Previous Year, and percent change from Budget. The Previous Year change is supplemented graphically by an in-cell bar chart. The Budget change is supplemented graphically by symbols. This list is controlled by a scroll bar form control.
5. Actual Revenue vs. Budget is a column chart that overlays Budget atop Actual with differences in widths for easier reading. The labels are custom labels. The chart is customizable by selecting the KPI (controlled by a list box form control) and the Top 6 or Bottom 6 apps (controlled by option button form controls.) A custom message is displayed below the controls to further highlight the user's selections.

Because we don't want the top or bottom based on Actual sales (the same apps always win or lose), we want to see apps that are top or bottom in percentage of Actual versus Budget percentage.

6. Profit for a selected app is presented via a line graph that displays the actuals up to the user selected month, as well as the Budget and Previous Year amounts for the entire year. The chart will display the app selected by the user via a scroll bar form control. The lines for Budget and Previous Year can be hidden or shown by the user via check box form controls. The title for the chart is customized based on the selected app name.

MODULE 8.2 – DASHBOARD WIREFRAME

Data Files:

08 – KPI Dashboard - Setup\8.2 - Dashboard Wireframe.xlsx

08 – KPI Dashboard - Setup\Excel Dashboard Report Vector Elements.xlsx

Start with an empty spreadsheet. Don't be tempted to recycle an existing report to save time. Older reports have cached information that can negatively impact your new report. Some elements that can cause problems when recycled are:

- Named ranges
- Conditional formatting
- Hyperlinks
- Hidden sheets
- Embedded macros
- Old Excel version incompatibility features

First, draw your dashboard as a wireframe on paper or a whiteboard.

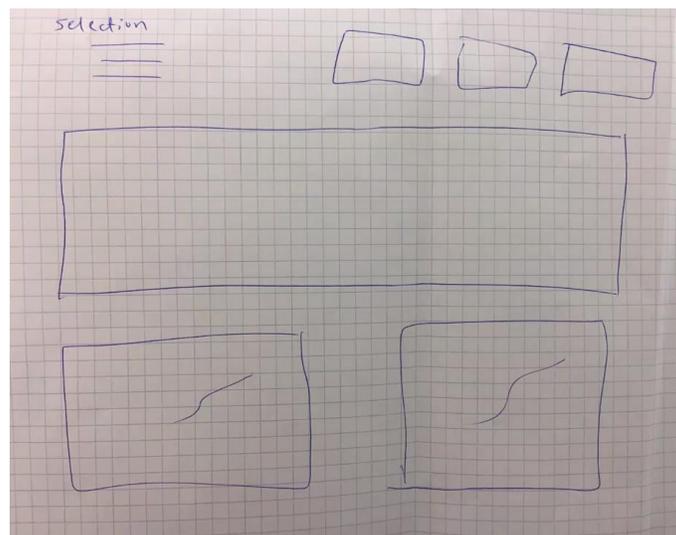


Figure 8.2 - Wireframe Sketch of a Dashboard

This will give you an idea of size, placement, and whether all the information will fit on a single page printout or single computer display.

Second, with the blank spreadsheet, use the vector elements in the supplied Excel file to quickly mock up your design. Sketch the dashboard out in vector elements, shapes, and boxes.

MODULE 8.3 – SPREADSHEET SETUP

Data Files:

08 – KPI Dashboard - Setup\8.3 - Dashboard Setup.xlsx

08 – KPI Dashboard - Setup\Data Visualization.thmx

If you are not beginning from an Excel template, take a few moments to setup the following environment for your spreadsheet:

- Define the color scheme. Use corporate colors if practical.
- Define your global font.
- Save your cosmetic choices as a theme for reuse.
- Add useful features to your Quick Access Toolbar (QAT). Items that are useful during dashboard development are:
 - Format painter
 - Align objects
 - Group objects
 - Insert shapes
 - Shape fill
 - Change outline color

NOTE: If you save your cosmetic changes as a theme, and then use the theme file on a different computer, the theme will be applied to the document, but not added to the Theme library. You must resave the theme in the Theme library folder of the new computer for it to display in the Theme library dropdown in the ribbon.

MODULE 8.4 – SETUP OF RAW DATA SHEET

Data File:

08 – KPI Dashboard - Setup\8.4 - Setup Raw Data.xlsx

In the supplied data set, we are retrieving the following cross-tabular information:

- Actual Revenue, Profit, and Cash for the Current Year
- Actual Revenue, Profit, and Cash for the Previous Year
- Revenue, Profit, and Cash for the Budget
- Actual Profit for each month of the Current Year
- Actual Profit for each month of the Previous Year
- Actual Profit for each month of the Budget

The data is broken down (row by row) as it relates to Division, where each Division is broken down into its respective apps.

NOTE: This is a STATIC data set. In the real world, this data would most likely have been pulled from a database via some connector add-in, like GLWand, Smart View, or Power Query. When the user executes a refresh, the data is pulled based on the year entered by the user. This in turn updates the entire data sheet with new information. In our example, the user will be able to enter a desired year, but our data will not enforce that year selection due to the disconnected nature of this example.

1. Insert a sheet and rename it to **Dashboard**.
2. Reduce the column widths of **A:D** to **15**. This allows for easier additions of new columns if the need arises, or for coloring of borders or placement of index numbers if needed.
3. Place the following text in the listed cells (we will restrict what can be entered in the adjacent cells later):

E5 = "Year" (no quotes)

E6 = "Month"

E7 = "Division"

4. Assign names to the following cells (this will make cell references in upcoming formulas easier to remember and enter):

- F5 = “**cur_year**” (no quotes). For development purposes, enter **2017** in the cell.
- F6 = “**cur_month**” For development purposes, enter “**Jun**” in the cell.

5. On the **Data** sheet, select cell **A2** and enter the following:

```
cur_year
```

6. Select cell **B2** and enter the following:

```
=A2-1
```

7. Select cells **B5:D5** and **H5:V5** and **AI:AT5** and enter the following:

```
=$A$2
```

8. Select cells **E5:G5** and **W5:AH5** and enter the following:

```
=$B$2
```

By linking the years, we create a dynamic list of years that will automatically update when the user selects a different year from the dashboard and refreshes the data.

9. Select cells **B6:J6** and enter the following:

```
=cur_month
```

	A	B	C	D	E	F	G	H	I	J	K	
1	Connected to Database via add-in											
2		2017	2016									
3		Actual			Actual PY			Budget			Actual	
4												
5		2017	2017	2017	2016	2016	2016	2017	2017	2017	2017	2017
6		Jun	Jun	Jun	Jun	Jun	Jun	Jun	Jun	Jun	Jun	Jan
7		Actual	Actual	Actual	Actual	Actual	Actual	Budget	Budget	Budget	Actual	Actual
8		Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue	Profit	Cash	Profit	Profit
9	Productivity Apps	162643	9359	103058	159773	9241	103723	164760	9443	104426	1005	
10	WenCaL	14432	240	11099	13699	210	12058	15113	363	11468	23	

Figure 8.3 - Output of Module 8.4

MODULE 8.5 – SETUP OF CALCULATION SHEET

Data File:

08 – KPI Dashboard - Setup\8.5 - Setup Calculation Sheet.xlsx

The Calculation sheet will hold the following elements:

Calculations needed to feed dashboard elements

Lists to feed data validation dropdown lists

Lists to feed form controls, such as...

- List boxes
- Combo boxes

Holding area for form control selection output

Don't try to predict all the needed sections of the Calculation sheet. Too many things are going to be unpredictable and require moving of existing content. It is better to build items as they are encountered during the development of the other parts of the workbook.

In the real world, you would be responsible for creating the calculation sheet and all the headings. This has been done to simplify the module and expedite module progress.

1. On the **Calculation** sheet, create a list of abbreviated months in cells **B15:B26**. Make sure the abbreviation format matches the abbreviation format of the columns in the **Data** sheet.
2. Enter the Division names (**Productivity Apps**, **Game Apps**, and **Utility Apps**) in cells **C15:C17**.
3. Enter the KPI names (**Revenue**, **Profit**, and **Cost**) in cells **D15:D17**.
4. Add the Division names across the top of the "App to Division Allocation" table in cells **G14:I14**.
5. From the completed dashboard file, copy and paste the app names from the **Calculation** sheet, cells **G15:I29**, and paste them in the same cell locations in the development workbook.
6. On the **Dashboard** sheet, assign a data validation rule to cell **F6** (month) that uses a **List** and references cells **B15:B26** on the **Calculation** sheet.
7. Assign a data validation rule to cell **F5** (year) that uses a **Whole Number Greater Than 2010**.

	B	C	D	E	F	G	H	I	J
13	For Drop-down selections					App to Division Allocation			
14		Productivity Apps	Revenue			Productivity Apps	Game Apps	Utility Apps	
15	Jan					WenCaL	Fightrr	Commuta	
16	Feb	Game Apps	Profit			Blend	Kryptis	Infic	
17	Mar	Utility Apps	Cash			Voltage	Perino	Accord	
18	Apr					Inkly	Five Labs	Misty Wash	
19	May					Sleops	Twistr	Twenty20	
20	Jun					Kind Ape	Hackrr	Tanox	
21	Jul					Pet Feed	Pes	Minor Liar	
22	Aug					Right App	Baden	Mosquit	
23	Sep					Mirrrr	Jellyfish	Atmos	
24	Oct					Halotot	Aviatrr	Scrap	
25	Nov					Flowrrr	deRamblr	Motocyco	
26	Dec					Silvrr	Arcade	Amplefio	
27						Dasring		Strex	
28						Rehire			
29						Didactic			
30									
31									
32									
33									
34									
35									

Figure 8.4 - Output of Module 8.5

Module 9

Scrolling Table and Bars

XelPlus.com

MODULE 9.1 – SETUP A COMBO BOX TO EASE SELECTION

Data File:

09 - Scrolling Table and Bars\9.1 - Setup a Combo Box to Ease Selection.xlsx

In this module, we will create our first form control. The form control will allow the user to select a Division. The Division selection will help to feed the proper data records that each of the major dashboard sections will use to turn into visualizations. The first visualization will be a scrolling table displaying financial information about apps. The apps displayed in the table will be related to the Division selected by the user. This will allow the user to focus on just the apps related to the selected Division, and not have to look at all the apps in the data while trying to figure out which apps are relevant.

The first thing to do is to set up the calculation area to receive the selection made by the combo box form control.

1. On the **Calculation** sheet, create a heading and a landing site cell for the Division selection. (This has been done for you to expedite development. In the real world, you would need to create these items.)
2. On the **Dashboard** sheet, insert a combo box in cell **F7** with the following parameters:

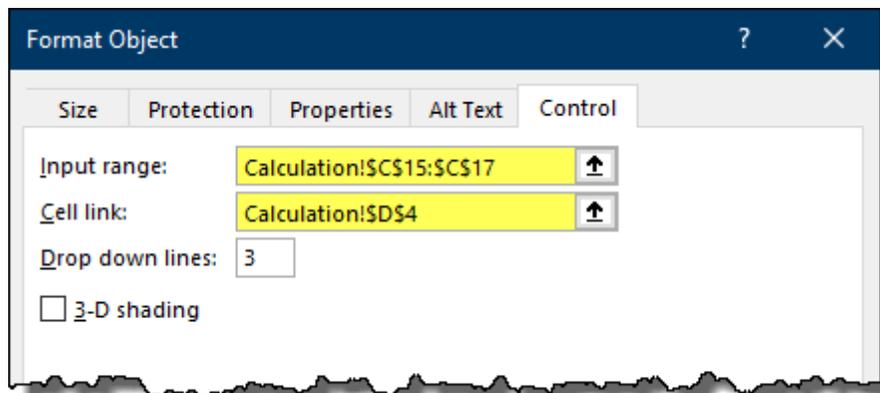


Figure 9.1 - Parameters for Step 2, Module 9.1

[For a review on the purpose and use of the Combo Box form control, see example in Module 6 for details.](#)

3. Test the Combo Box in cell **F7** to ensure you are returning the current number on the **Calculation** sheet.

We need to create a list of index numbers and app names. The index numbers in column “M” of the Calculation sheet will assign unique numbers to the selected Division’s apps. The apps will be pulled from the **App to Division Allocation** list on the Calculation sheet and placed in column “N” on the Calculation sheet under the heading “App”. When the user selects a Division from the combo box, the

related Division apps will be pulled from the table and listed in column “N”. The index numbers in column “M” will serve as selection points for the scrolling table.

4. On the **Calculation** sheet, using the **Format Painter** tool, copy the formatting from cells **B2:C2** and paste into cells **M2:AC2**.
5. In cell **M2**, enter the title “**Calculation for scrolling table**”.
6. In cell **M14**, enter the title “**Nr.**”.
7. In cell **N14**, enter the title “**App**”.

We want to generate a list of apps in cells **N15:N29** based on the user’s selection from the Combo Box on the Dashboard sheet.

8. In cell **N15**, enter the following formula:

```
=INDEX(G15:I15,$D$4)
```

9. Fill the newly created formula through cells **N15:N34**.

NOTE: We are allowing for growth in our app list by filling down beyond the length of our currently longest list.

10. We don’t want to display “0” in the generated app list when there is a corresponding empty row, so we will modify the formula with the following IF statement logic. This will act as a test to replace the original “0” answer on empty rows with empty text (two double-quotes).

```
=IF(INDEX(G15:I15,$D$4)="", "", INDEX(G15:I15,$D$4))
```

11. To create a list of numbers that correspond to the number of returned items in the app list; select cell **M15** and enter the following formula:

```
=IF(N15<> "", ROW(A1), "")
```

12. Fill the newly created formula through cells **M15:M34**.

We want to capture and display the total of the entire selected division. We also want to capture and display the Total Revenue, Total Profit, and Total Cash for the selected division.

13. Select cell **N36** and enter the following formula:

```
=INDEX(G14:I14,D4)
```

14. Color cells **M14:N14** with a **light gray** fill color.

15. For the “Actuals”, enter the titles “Revenue”, “Profit”, and “Cash” in cells **O14:Q14**.
16. Copy and paste cells **O14:Q14** into cells **R14** and **U14**. This will become the heading for the “Previous Year” and “Budget” calculations.
17. In cell **O13**, enter the title “Actual”.
18. In cell **R13**, enter the title “PY”.
19. In cell **U13**, enter the title “Budget”.

To maintain visual consistency with our color scheme, we will use the three shades of blue (listed on the **Wireframe** sheet) to indicate the categories of calculations.

20. Select cells **O14:Q14** and fill the cells with the **darker blue** color (**Wireframe** sheet cell **Q5**).
21. Select cells **R14:T14** and fill the cells with the **medium blue** color (**Wireframe** sheet cell **Q6**).
22. Select cells **U14:W14** and fill the cells with the **light blue** color (**Wireframe** sheet cell **Q7**).
23. Select cells **O13:W13** and fill with a **light gray** color.

Now we will set the titles for the variance calculation of “Actual to Previous Year” and “Actual to Budget”.

24. In cell **X13**, enter the title “Variance Actual to PY”.
25. Copy the headings from cells **R13:W13** to cells **X13:AC13**.
26. In cell **AA13**, enter the title “Variance Actual to Budget”.
27. Fill cells **X13:AC13** with a **light gray** fill color.
28. Update all the titles in cells **X14:AC14** to include “%” symbols at the ends of the titles.

	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
13			Actual			PY			Budget			Variance Actual to PY			Variance Actual to Budget			
14	Nr.	App	Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue%	Profit%	Cash%	Revenue%	Profit%	Cash%	
15		1 WenCaL																
16		2 Blend																
17		3 Voltage																
18		4 Inkly																
19		5 Sleops																
20		6 Kind Ape																
21		7 Pet Feed																
22		8 Right App																
23		9 Mirrrr																
24		10 Haiotot																
25		11 Flowrrr																
26		12 Silvrr																
27		13 Dasring																
28		14 Rehire																
29		15 Didactic																
30																		
31																		

Figure 9.2 - Output of Module 9.1

MODULE 9.2 – COMPLEX LOOKUP WITH INDEX AND MATCH

Data File:

09 - Scrolling Table and Bars\9.2 - Complex Lookup with Index Match.xlsx

In this module, we will move closer to the creation of the scrolling table by pulling data from the Data sheet and placing it in the Calculation sheet.

The app list in column **N** is dynamic based on the number in cell **D4** (returned from the Division combo box on the Dashboard sheet.) This will assist in querying the Data sheet and placing the results in columns **O:AC**.

We will first concentrate on pulling the “Revenue”, “Profit”, and “Cash” for “Actual”, “Previous Year (PY)” and “Budget” (B). If all the columns in the source data were fixed and we could rely on the categories never moving, we could write simple VLOOKUP functions to retrieve the data. Because we want to make this “bullet-proof”, and the real world is seldom static, we will use the INDEX and MATCH functions, in conjunction with custom headings, to query the data source and return the relevant information.

Remember that the INDEX function can move to a row position and a column position in an array. The MATCH function returns a number based on where it finds an item in a list. If we use two MATCH functions, one for up/down lists (rows) and one for left/right lists (columns), we can feed the movement instructions to the INDEX function to return a value from a specific location in a cell array. The formula we will be building will be an INDEX – MATCH – MATCH function.

When we are searching the source data from left to right, we will be searching based on multiple criteria (i.e. year, month, KPI, Actual vs. Budget). We want to keep our formulas as simple as possible, and we want to try not to introduce array functions into the process if it can be avoided. To assist in this process, we will create new headings for the columns to act as “helper columns” that will combine the original headings into a single cell thereby creating a unique key for each column.

1. On the **Data** sheet, select cells **B4:AT4** and enter the following formula:

```
=B5&B6&B7&B8
```

The Calculation sheet needs to have the same header structure as the Data sheet. We will now create a set of “helper cells” to replicate the heading structure of the Data sheet on the Calculation sheet.

2. On the **Calculation** sheet, select cells **O12:W12** and enter the following formula:

```
=O14
```

3. Select cells **O11:T11** and enter the following formula:

```
"Actual" (no quotes)
```

4. Select cells **U11:W11** and enter the following formula:

```
"Budget" (no quotes)
```

5. Select cells **O10:W10** and enter the following formula:

```
=cur_month
```

6. Select cells **O9:Q9** and cells **U9:W9** and enter the following formula:

```
=cur_year
```

7. Select cells **R9:T9** and enter the following formula:

```
=cur_year-1
```

8. Select cells **O8:W8** and enter the following formula:

```
=O9&O10&O11&O12
```

With the “helper headings” in place, create the formula that will query the Data sheet based on a two-criteria lookup; app name (*for row searches*) and “year/month/area/KPI” (*for column searches*).

9. On the **Calculation** sheet, select cell **O15** and enter the following formula:

```
=IF($N15="", "", INDEX(Data!$B$9:$J$51, MATCH($N15, Data!$A$9:$A$51, 0), MATCH(O$8, Data!$B$4:$J$4, 0)))
```

Notice that the INDEX function and the 2nd MATCH function are only selecting data on the **Data** sheet up to column **J**. The reason for this is because starting in column **K** the data is broken down monthly.

10. Fill the formula down to row **34** and fill the formula over to column **W**.

It is now time to create the formulas that will calculate the percentage changes from Actual to Budget for the Previous Year (PY) and the Budget (B).

11. On the **Calculation** sheet, select cells **X15:Z15** and enter the following formula:

```
=IF($N15<> "", (O15-R15)/R15, "")
```

12. Select cells **AA15:AC15** and enter the following formula:

```
=IF($N15<>"", (O15-U15)/U15, "")
```

13. Fill the formula down to row **34**.

The dashboard will also need to see the same “Revenue/Profit/Cash” information for the selected Division.

14. Select cell **N36** and enter the following formula:

```
=INDEX(G14:I14,D4)
```

15. Select cells **O15:AC15** and select **Copy**. **Paste** the copied data into cells **O36:AC36**.

16. Copy the headings from cells **O14:AC14** and **paste** into cells **O35:AC35**.

	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	
8			2017	AugA	2017	AugA	2017	AugA	2016	AugA	2016	AugA	2016	AugA	2017	AugB	2017	AugB	Cash
9			2017	2017	2017	2016	2016	2016	2017	2017	2017								
10			Aug	Aug	Aug	Aug	Aug	Aug	Aug	Aug	Aug								
11			Actual	Actual	Actual	Actual	Actual	Actual	Actual	Budget	Budget	Budget							
12			Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue	Profit	Cash								
13			Actual							Budget									
14	Nr.	App	Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue%	Profit%	Cash%	Revenue%	Profit%	Cash%		
15	1	WenCaL	14432	240	11099	13699	210	12058	15113	363	11468	5%	14%	-8%	-5%	-34%	-3%		
16	2	Blend	17990	1166	7137	16395	1003	7423	18181	1223	8196	10%	16%	-4%	-1%	-5%	-13%		
17	3	Voltage	15117	1613	5828	14138	1200	6212	13455	1400	5810	7%	34%	-6%	12%	15%	0%		
18	4	Inkly	11154	731	8008	11502	499	8953	12031	641	8424	-3%	46%	-11%	-7%	14%	-5%		
19	5	Sleeps	11022	550	9000	14644	780	8670	13112	474	8193	-25%	-29%	4%	-16%	16%	10%		
20	6	Kind Ape	8905	469	6026	8796	316	6251	9096	620	5804	1%	48%	-4%	-2%	-24%	4%		
21	7	Pet Feed	16735	800	10149	17503	1083	10507	18207	1176	10508	-4%	-26%	-3%	-8%	-32%	-3%		
22	8	Right App	3635	96	2050	2337	90	1585	3579	184	1396	56%	7%	29%	2%	-48%	47%		
23	9	Mirrrr	15627	1996	9397	15357	1908	9702	14634	1584	9527	2%	5%	-3%	7%	26%	-1%		
24	10	Halot	7270	150	8948	7119	349	8520	7158	120	8630	2%	-57%	5%	2%	25%	4%		
25	11	Flowrrr	5955	260	5931	5812	233	5128	5977	210	4722	6%	12%	16%	0%	24%	26%		
26	12	Silvrr	7666	274	6624	5060	369	5174	7099	230	6253	52%	-26%	28%	8%	19%	6%		
27	13	Dasring	10857	281	4396	10547	281	4979	10799	333	4779	3%	0%	-12%	1%	-16%	-8%		
28	14	Rehire	9873	370	7076	11069	652	6853	10500	637	8490	-11%	-43%	3%	-6%	-42%	-17%		
29	15	Didactic	6405	363	1389	5995	268	1708	5819	248	2226	7%	35%	-19%	10%	46%	-38%		
30																			
31																			
32																			
33																			
34																			
35			Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue	Profit	Cash	Revenue%	Profit%	Cash%	Revenue%	Profit%	Cash%		
36		Productivity Apps	162643	9359	103058	159773	9241	103723	164760	9443	104426	2%	1%	-1%	-1%	-1%	-1%		
37																			

Figure 9.3 - Output of Module 9.2 - Calculation Sheet

MODULE 9.3 – SCROLLING DATA TABLE

Data File:

09 - Scrolling Table and Bars\9.3 - Scrolling Data Table.xlsx

In this module, we will learn how to create a scrolling table. This table will not only scroll through the apps, it will also react to the Division selected by the user in the combo box on the dashboard. By limiting the number of apps visible at any one time and scrolling through the list, we keep from overwhelming the viewer with what could be thousands of line items in the table and focusing our attention to a small window of displayed apps.

1. The first thing to do is set up the headings of our table. Enter the below listed headings in the locations defined by the screenshot below. Use the Delta Δ symbol from the **Insert** \rightarrow **Symbol** library.

	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
10	Revenue								Profit								Cash					
11	Actual	% Δ PY			% Δ B				Actual	% Δ PY		% Δ B				Actual	% Δ PY		% Δ B			

Figure 9.4 - Parameters for Step 1, Module 9.3

The scrolling table needs to pull 10 records from the Calculation sheet based on a starting line number. In other words, if we select the row 1 in the list, we want to display records 1 thru 10. If we select row 6 in the list, we want to see records 6 thru 15. To determine the starting row number at which to begin returning records we need a form control that will allow us to select a number between 1 and some ending number. A spin box could work, but it may not be as intuitive a control for most users. The form control most users are comfortable with is a scroll bar.

2. On the **Dashboard** sheet, insert a scroll bar with the following parameters:

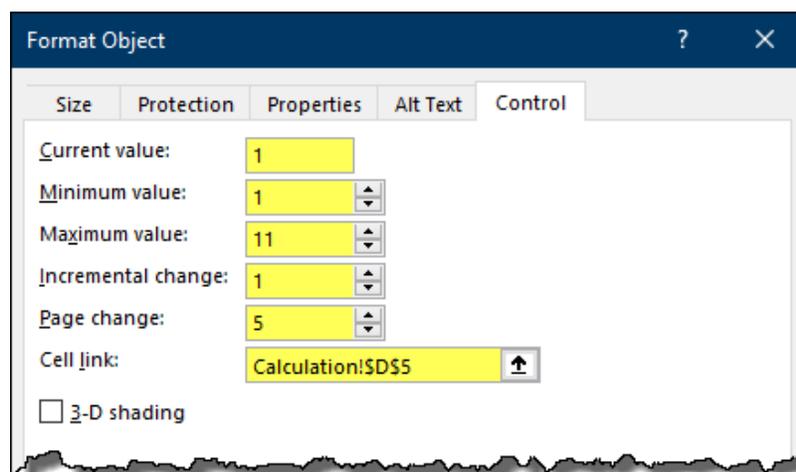


Figure 9.5 - Parameters for Step 2, Module 9.3

[For a review on the purpose and use of the Scroll Bar form control, see example in Module 6 for details.](#)

3. Test the Scroll Bar form control to ensure the number in cell **D5** of the **Calculation** sheet is updating accordingly.
4. Select cell **D13** and enter the following formula:

```
=INDEX(Calculation!M15:M$34,Calculation!$D$5)
```

5. Fill the formula in cell **D13** down to cell **D22**.

If you look closely at the range value in the INDEX function, the bottom of the list is fixed at row 34, but the top of the list is dynamic starting on row 15. This means that as we fill this formula down the rows, the range of possibilities will get smaller, effectively moving the start position of the search.

6. Fill the formula in cell **D13** across to cell **F13** (*do not fill the formatting*), then fill the formulas in cells **E13:F13** down to row **22**.

It is now time to create the same lookup calculations for the percentage change in Previous Year and Budget. Because the logic is almost identical, we will copy the formula from cell **F13** (*from the formula bar*) and paste it in cells **H13** and **K13**.

7. Select cell **F13** and in the formula bar, highlight the formula and select **Copy**. **Paste** the copied formula into cells **H13** and **K13**.
8. Edit the formula in cell **H13** to read as follows:

```
=INDEX(Calculation!X15:X$34,Calculation!$D$5)
```

9. Edit the formula in cell **K13** to read as follows:

```
=INDEX(Calculation!AA15:AA$34,Calculation!$D$5)
```

The remainder of this module is to repeat most of the above steps for the Actual Profit and Actual Cash sections. This can be accomplished by duplicating the formulas in cells **F13:K22** into cells **N13:S22** and cells **V13:AA22**. After the formulas have been duplicated, minor adjustments will be made to reflect column shift issues.

10. Select cells **F13:K22** and click **Copy**.
11. Select cell **N13** and click **Paste**.
12. Select cell **V13** and click **Paste**.

13. Update the following cells with the listed modifications:

Cell	Original Formula	Updated Formula
N13	=INDEX(Calculation!W15:W\$34,Calculation!\$D\$5)	=INDEX(Calculation!P15:P\$34,Calculation!\$D\$5)
P13	=INDEX(Calculation!AF15:AF\$34,Calculation!\$D\$5)	=INDEX(Calculation!Y15:Y\$34,Calculation!\$D\$5)
S13	=INDEX(Calculation!AI15:AE\$34,Calculation!\$D\$5)	=INDEX(Calculation!W15:W\$34,Calculation!\$D\$5)
V13	=INDEX(Calculation!AE15:AE\$34,Calculation!\$D\$5)	=INDEX(Calculation!P15:P\$34,Calculation!\$D\$5)
X13	=INDEX(Calculation!AN15:AN\$34,Calculation!\$D\$5)	=INDEX(Calculation!Z15:Z\$34,Calculation!\$D\$5)
AA1	=INDEX(Calculation!AQ15:AQ\$34,Calculation!\$D\$5)	=INDEX(Calculation!AC15:AC\$34,Calculation!\$D\$5)

Remember to always double and triple check your formulas for accuracy. It is very easy to miss a needed update when copy and pasting formulas, especially when performing many copy/paste operations to reduce repetitive work.

Also, monitor your file’s size. It is possible an incorrect paste operation results in excessive pasting beyond the desired range. (i.e. accidentally pasting a formula to all cells in a column when you only wanted a small range of cells.) Periodically examine the file’s size property in the Info section of the Backstage (after saving.)

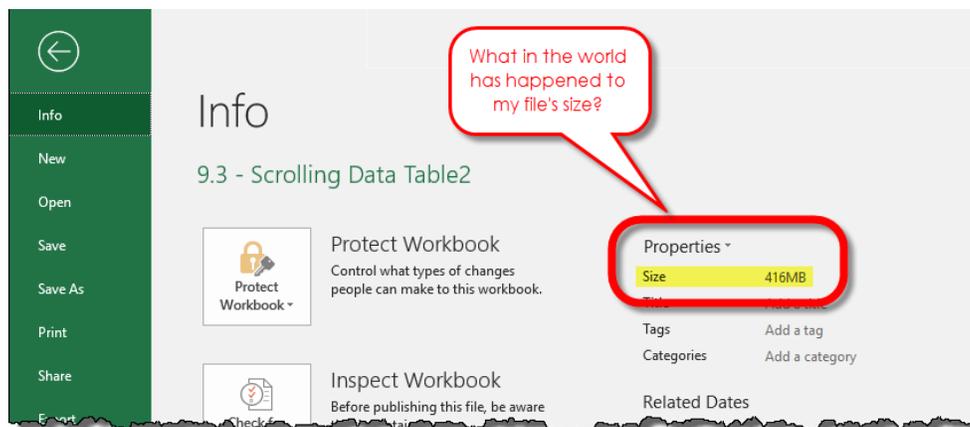


Figure 9.6 - Backstage View - Info Section (file size problem)

	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	A	AD
5		Year	2017																								
6		Month	Aug																								
7		Division	Productivity Apps																								
10			Revenue																								
11			Actual	% Δ PY	% Δ B				Profit	% Δ PY	% Δ B								Cash	% Δ PY	% Δ B						
13	1	WenCaL	14,432	5.4%	-4.5%			240	14.3%	-33.9%			11,468	-8.0%	-3.2%												
14	2	Blend	17,990	9.7%	-1.1%			1,166	16.3%	-4.7%			8,196	-3.9%	-12.9%												
15	3	Voltage	15,117	6.9%	12.4%			1,613	34.4%	15.2%			5,810	-6.2%	0.3%												
16	4	Inkly	11,154	-3.0%	-7.3%			731	46.5%	14.0%			8,424	-10.6%	-4.9%												
17	5	Sleeps	11,022	-24.7%	-15.9%			550	-29.5%	16.0%			8,193	3.8%	9.8%												
18	6	Kind Ape	8,905	1.2%	-2.1%			469	48.4%	-24.4%			5,804	-3.6%	3.8%												
19	7	Pet Feed	16,735	-4.4%	-8.1%			800	-26.1%	-32.0%			10,508	-3.4%	-3.4%												
20	8	Right App	3,635	55.5%	1.6%			96	6.7%	-47.8%			1,396	29.3%	46.8%												
21	9	Mirr	15,627	1.8%	6.8%			1,996	4.6%	26.0%			9,527	-3.1%	-1.4%												
22	10	Halotot	7,270	2.1%	1.6%			150	-57.0%	25.0%			8,630	5.0%	3.7%												

Figure 9.7 - Output of Module 9.3

MODULE 9.4 – CONDITIONALLY FORMATTED DATA BARS

Data File:

09 - Scrolling Table and Bars\9.4 - Conditionally-Formatted Data Bars.xlsx

What we would like to see in our scrolling table are data bars that represent the percentage change from Actual to Previous Year. The “longer” the bar, the further away Actual is from reaching Previous Year numbers. This can be either in a positive or negative direction. We could create a bar chart and position this on top of our table (in fact, this is what we had to do prior to Excel 2010 to get this effect), instead we will use Conditional Formatting’s Data Bars to produce a bar chart effect. This is a much simpler approach.

For Data Bars to work, there must be numbers in the cells being charted. This is a simple matter of repeating all the percentage change numbers into another column.

1. On the **Dashboard** sheet, select cell **I13** and enter the following formula:

=H13

2. Fill cell **I13** down to cell **I22**.
3. With cells **I13:I22** selected, on the **Home** ribbon select **Conditional Formatting** → **Data Bars** and select the dark-blue data bar.

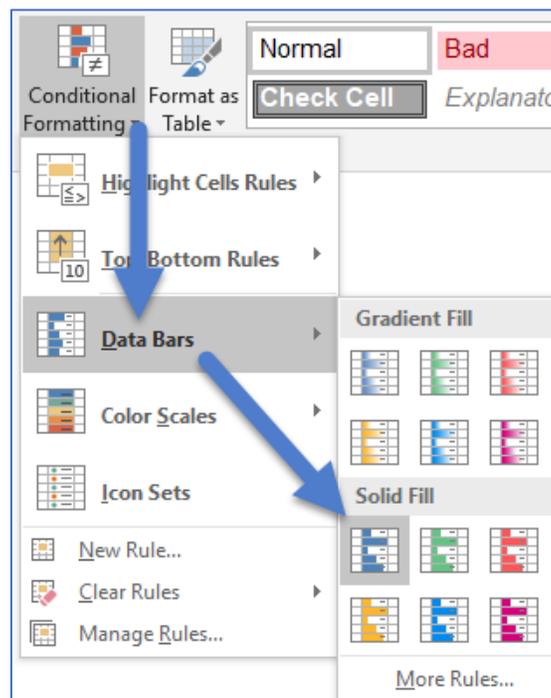


Figure 9.8 - Conditional Formatting - Data Bars feature

There are several adjustments that need to be made to make the data bars look good in our dashboard.

- In the data bar formatting options, hide the numbers by selecting the “**Show Bar Only**” check box.

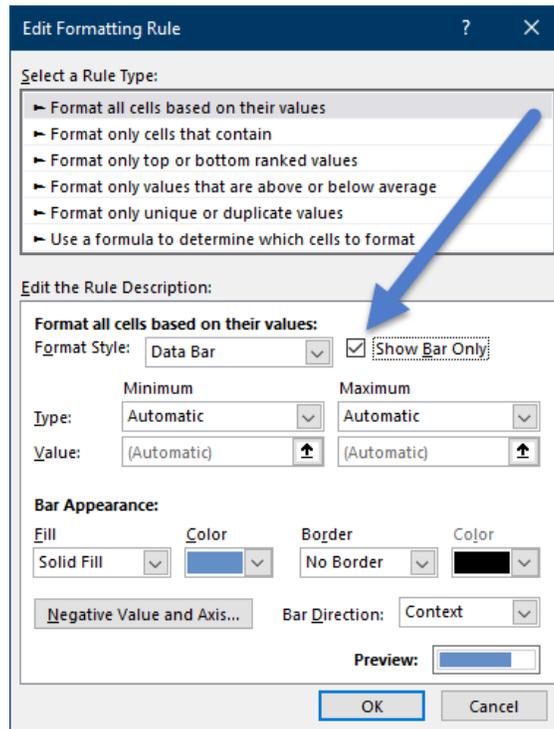


Figure 9.9 - Show Bar Only feature for Data Bars

- Select the **Negative Value and Axis...** button and change the **Negative bar fill color** to **50% gray** and change the **Axis color** to **white**.

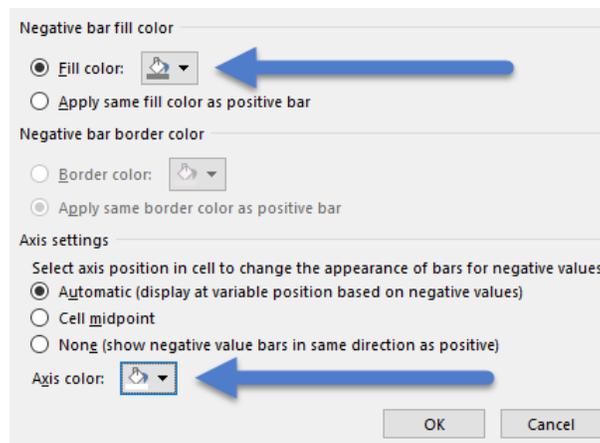


Figure 9.10 - Negative Bar and Axis Color Options

Select the Utility Division and scroll through the list. Notice that the data bar axis shifts from left to right when extreme positive and negative numbers are encountered. We need to fix the axis position, so it does not move based on the value of the visible data. We will need to build some helper calculations to determine the minimum and maximum values within Revenue, Profit, and Cash. Once we have these values, we can direct Conditional Formatting to use those numbers as a display range for the Data Bars.

6. On the **Calculation** sheet, in the cells listed below, enter the following formulas:

	X	Y
39	Min Rev %	=MIN (X15:X34)
40	Max Rev %	=MAX (X15:X34)
41	Min Profit %	=MIN (Y15:Y34)
42	Max Profit %	=MAX (Y15:Y34)
43	Min Cash %	=MIN (Z15:Z34)
44	Max Cash %	=MAX (Z15:Z34)

Figure 9.11 - Parameters for Step 6, Module 9.4

- Return to the **Dashboard** sheet and select cell **I13**. It is not necessary to select all the cells with the conditional format because Excel has this rule defined to apply to the original range, not just the selected cell.
- Return to the formatting controls of the data bars and configure the **Minimum** and **Maximum** – **Type** and **Value** settings as listed in the below graphic.

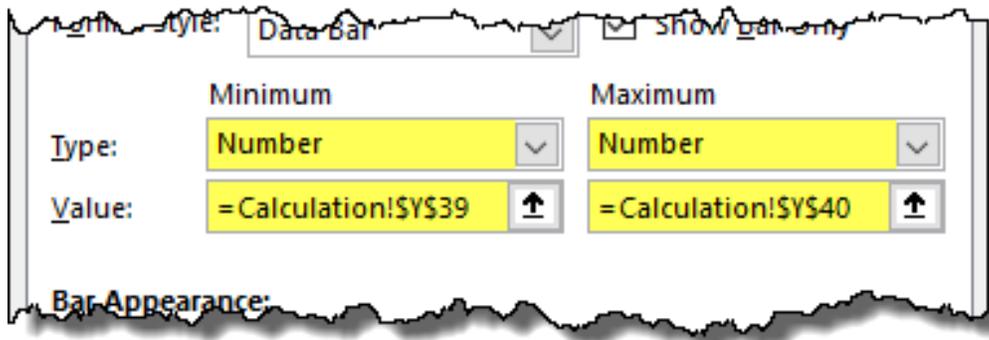


Figure 9.12 - Parameters for Step 8, Module 9.4

- Copy and paste the formulas and conditional formatting from the column **I** cells to the corresponding cells in columns **Q** and **Y**.
- Update the **Minimum** and **Maximum** value settings in the pasted columns to reflect their respective minimum and maximum values defined on the **Calculation** sheet.

	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
10			Revenue							Profit								Cash									
11			Actual		% Δ PY		% Δ B			Actual		% Δ PY		% Δ B				Actual		% Δ PY		% Δ B					
13	1	WenCaL	14,432	5.4%		-4.5%		240	14.3%		-33.9%		11,468	-8.0%		-3.2%											
14	2	Blend	17,990	9.7%		-1.1%		1,166	16.3%		-4.7%		8,196	-3.9%		-12.9%											
15	3	Voltage	15,117	6.9%		12.4%		1,613	34.4%		15.2%		5,810	-6.2%		0.3%											
16	4	Inkly	11,154	-3.0%		-7.3%		731	46.5%		14.0%		8,424	-10.6%		-4.9%											
17	5	Sleeps	11,022	-24.7%		-15.9%		550	-29.5%		16.0%		8,193	3.8%		9.8%											
18	6	Kind Ape	8,905	1.2%		-2.1%		469	48.4%		-24.4%		5,804	-3.6%		3.8%											
19	7	Pet Feed	16,735	-4.4%		-8.1%		800	-26.1%		-32.0%		10,508	-3.4%		-3.4%											
20	8	Right App	3,635	55.5%		1.6%		96	6.7%		-47.8%		1,396	29.3%		46.8%											
21	9	Mirrri	15,627	1.8%		6.8%		1,996	4.6%		26.0%		9,527	-3.1%		-1.4%											
22	10	Halotot	7,270	2.1%		1.6%		150	-57.0%		25.0%		8,630	5.0%		3.7%											

Figure 9.13 - Output of Module 9.4

MODULE 9.5 – ARROWS FOR DEVIATION

Data File:

09 - Scrolling Table and Bars\9.5 - Arrows for Deviations.xlsx

In this module, we will use green up arrows and red down arrows to visually indicate an increase or decrease in the Actual to Budget percentage change amount. Because a small amount of change may not be worth grabbing our attention, we only want an arrow to appear when the percentage change is beyond some defined threshold.

Excel's Conditional Formatting – Icon Sets have some nice icons to choose from, but the problem lies in customization. Icon sets cannot have their color intensities adjusted, nor can you use icons outside of the defined icon library. Using symbols from the symbol library allows for greater flexibility in shape and color, but the tradeoff is a slightly more complicated setup.

It is up to you to decide which path you will take, simplicity with restrictions, or complexity with freedom.

Let's look at the simple approach using built-in conditional formatting icons.

1. On the **Dashboard** sheet, select cell **L13** and enter the formula below:

=K13

2. Fill the formula in cell **L13** down to cell **L22**.
3. Select cells **L13:L22** and apply **Conditional Formatting – Icon Set**    to the cells.
4. Apply the following customizations to the newly added icon sets:
 - a) Show the icons only
 - b) Restrict the values to a defined number
 - c) Set the high-end threshold to 5%
 - d) Set the low-end threshold to -5%
 - e) Do not display an icon for the range between -5% and 5%

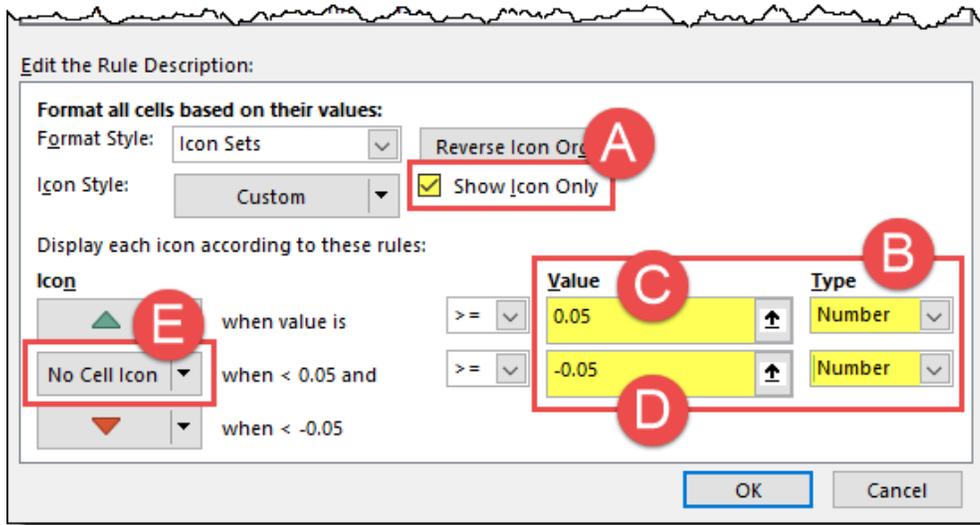


Figure 9.14 - Parameters for Step 4, Module 9.5

Although the icon set approach works well, we may not be keen on the colors or shapes. The following method will use symbols and custom colors to achieve the same result but with a different effect.

1. On the **Dashboard** sheet, select cell **T17** and enter the formula below (the arrow symbols can be copied and pasted from the Calculation sheet in cells I3 & I4):

=IF(S13="", "", IF(S13 >= 0.05, "▲", IF(S13 < -0.05, "▼", "")))

or

=IF(S13="", "", IF(S13 >= 0.05, Calculation!\$I\$3, IF(S13 < -0.05, Calculation!\$I\$4, "")))

NOTE: If you use the second version formula (above) that references cells for the icons, your display cells must use the same font set as the source icons (i.e. wingdings, symbols, webdings, etc...).

2. Fill the formula in cell **T17** down to cell **T22**.
3. Select cells **T17:T22**.
4. Apply a conditional formatting rule that will color the cell's font **green** if the cell's value is equal to the cell holding the reference symbol (**Calculation!I3**).
5. Apply a conditional formatting rule that will color the cell's font **red** if the cell's value is equal to the cell holding the reference symbol (**Calculation!I4**).

You can now change the symbols on the **Calculation** sheet and the corresponding symbols will update on the **Dashboard** sheet.

	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	A	AD
10			Revenue								Profit							Cash									
11			Actual	% Δ PY			% Δ B				Actual	% Δ PY			% Δ B			Actual	% Δ PY			% Δ B					
13	1	WenCaL	14,432	5.4%			-4.5%				240	14.3%			-33.9%	▼		11,468	-8.0%								
14	2	Blend	17,990	9.7%			-1.1%				1,166	16.3%			-4.7%			8,196	-3.9%								
15	3	Voltage	15,117	6.9%			12.4%	▲			1,613	34.4%			15.2%	▲		5,810	-6.2%								
16	4	Inkly	11,154	-3.0%			-7.3%	▼			731	46.5%			14.0%	▲		8,424	-10.6%								
17	5	Sleeps	11,022	-24.7%			-15.9%	▼			550	-29.5%			16.0%	▲		8,193	3.8%								
18	6	Kind Ape	8,905	1.2%			-2.1%				469	48.4%			-24.4%	▼		5,804	-3.6%								
19	7	Pet Feed	16,735	-4.4%			-8.1%	▼			800	-26.1%			-32.0%	▼		10,508	-3.4%								
20	8	Right App	3,635	55.5%			1.6%				96	6.7%			-47.8%	▼		1,396	29.3%								
21	9	Mirrrr	15,627	1.8%			6.8%	▲			1,996	4.6%			26.0%	▲		9,527	-3.1%								
22	10	Halotot	7,270	2.1%			1.6%				150	-57.0%			25.0%	▲		8,630	5.0%								

Figure 9.15 - Output of Module 9.5

MODULE 9.6 – FORMATTING OF SCROLLING DATA TABLE

Data File:

09 - Scrolling Table and Bars\9.6 - Formatting of Scrolling Data Table.xlsx

09 – Scrolling Table and Bars\Corporate Logo.png

Now that the scrolling table is constructed, we can take a moment to start formatting the table to its desired “look and feel” of that when finished. We can also take the opportunity to apply a frame to our dashboard in anticipation of the upcoming dashboard objects.

Perform the following cosmetic operations:

- Center the titles
Highlight **F10:L10**, **N10:T10**, and **V10:AB10** and apply the **Center Across Selection** feature. DO NOT merge the cells.

[For a review on the purpose and use of the Center Across Selection feature, see example in Module 3 for details.](#)

- Add the corporate logo
Bring over the corporate logo from the **Wireframe** sheet or insert the corporate logo from the supplied image file (“Corporate Logo.png”).
- Remove the gridlines
- Bold the Year (**F5**) and Month (**F6**) cells
- Right-align the selection headings (**E5:E7**)
- Reduce rows 1 through 3 to 5 pixels
- Reduce columns A through C to 5 pixels
- Create a “border”
This is done by selecting cells and giving them a fill color. Use the finished version of the dashboard file as an example on which cells to shade.
- Right-align “Actual” titles (columns **F**, **N**, & **V**)
- Adjust the columns to “look good”
- Color band the table rows
Highlight every other row in the table and apply a very light gray fill color

		Revenue			Profit			Cash		
	Actual	% Δ PY	% Δ B	Actual	% Δ PY	% Δ B	Actual	% Δ PY	% Δ B	
1 WenCaL	14,432	5.4%	-4.5%	240	14.3%	-33.9% ▼	11,468	-8.0%	-3.2%	
2 Blend	17,990	9.7%	-1.1%	1,166	16.3%	-4.7%	8,196	-3.9%	-12.9% ▼	
3 Voltage	15,117	6.9%	12.4% ▲	1,613	34.4%	15.2% ▲	5,810	-6.2%	0.3%	
4 Inkly	11,154	-3.0%	-7.3% ▼	731	46.5%	14.0% ▲	8,424	-10.6%	-4.9%	
5 Sleops	11,022	-24.7%	-15.9% ▼	550	-29.5%	16.0% ▲	8,193	3.8%	9.8% ▲	
6 Kind Ape	8,905	1.2%	-2.1%	469	48.4%	-24.4% ▼	5,804	-3.6%	3.8%	
7 Pet Feed	16,735	-4.4%	-8.1% ▼	800	-26.1%	-32.0% ▼	10,508	-3.4%	-3.4%	
8 Right App	3,635	55.5%	1.6%	96	6.7%	-47.8% ▼	1,396	29.3%	46.8% ▲	
9 Mirrrr	15,627	1.8%	6.8% ▲	1,996	4.6%	26.0% ▲	9,527	-3.1%	-1.4%	
10 Halotot	7,270	2.1%	1.6%	150	-57.0%	25.0% ▲	8,630	5.0%	3.7%	

Figure 9.16 - Output of Module 9.6

Module 10

Dynamic Headers

MODULE 10.1 – CONDITIONAL COMMENTS IN HEADER

Data File:

10 - Dynamic Headers\10.1 - Conditional Comments in Header.xlsx

When the viewer opens the dashboard, we want them to immediately notice three specific Key Performance Indicators (KPIs). These are as follows:

- Revenue
- Profit
- Cash

We would like for these KPIs to not only show the Actual amounts of these three areas, but we would also like to see how much each area is above or below the Budget. We will achieve this with several creative elements.

Question: Do we currently possess these calculation answers in our data or in our calculation sheet?

Answer: Yes and no. We have the actuals for Revenue, Profit, and Cash in the Calculation sheet in cells O36:Q66. What we lack are the absolute differences between each KPI and their respective budgets.

We will need to create helper calculations to determine the absolute differences to Budget for Revenue, Profit and Cost.

The following items will be created:

Calculation sheet helper formulas to determine the absolute differences to Budget for Revenue, Profit and Cost.

Helper formulas to generate the “over/under” messaging.

Textboxes to retrieve the calculation results and display them on the Dashboard sheet.

1. On the **Calculation** sheet, select cell **O39** and enter the formula below:

```
=O36-U36
```

2. Select cell **O40** and enter the formula below:

```
=P36-V36
```

3. Select cell **O41** and enter the formula below:

```
=Q36-W36
```

- Select cell **P39** and enter the formula below:

```
=IF(O39>0,"Over Budget",IF(O39<0,"Under Budget"))
```

- Repeat the above formula in cell **P39** down into cells **P40** and **P41**.
- For all the non-percentage numbers on the **Calculation** sheet (columns **O** through **W**), apply a comma style with negative symbols and zero decimal place accuracy.

	N	O	P	Q	R
35		Revenue	Profit	Cash	Revenue
36	Productivity Apps	\$162,643	\$9,359	\$103,058	\$159
37					
38	Calculations for header				
39	Revenue A-B	-\$2,117	Under Budget		
40	Profit A-B	-\$84	Under Budget		
41	Cash A-B	-\$1,368	Under Budget		
42					
43					

Figure 10.17 - Output of Module 10.2

MODULE 10.2 – ADDING SHAPES AND TEXTBOXES

Data File:

10 - Dynamic Headers\10.2 - Adding Shapes and Textboxes.xlsx

In this module, we will use textboxes to bring information from the Calculation sheet and display the information on the Dashboard sheet. The reason for using textboxes instead of linking cells is because textboxes (and shapes) can be placed anywhere on a sheet and are not constrained to cell height and width dimensions. Textboxes literally “float” above the surface of the spreadsheet giving complete freedom to positioning and sizing as needed.

Let’s examine a finished KPI display box.

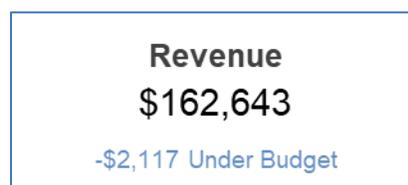


Figure 10.1 - Finished KPI Display Box for Revenue

This display box is made up of four separate elements that have been grouped together:

- A rectangular shape with a white fill and no border to act as the background
- Four textbox objects to display static text or linked data from calculations on the Calculation sheet

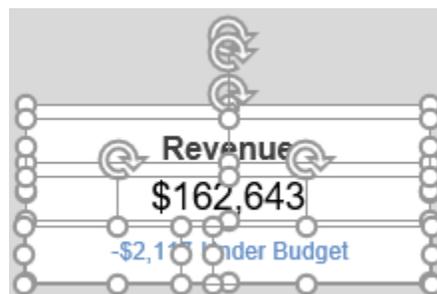


Figure 10.2 - KPI Display Box Elements

It will be easiest to create one finished, grouped textbox object and then duplicate it for the remaining two KPIs. In this way, the only thing that needs to be adjusted are the embedded cell links that display specific information from the Calculation sheet.

For the first KPI, perform the following steps:

1. Insert a textbox approximately **1” tall** and **2” wide**. This will serve as the KPI background.
2. Fill the textbox with a **white** fill.
3. Remove the border from the textbox.

4. Embed the following text in the textbox:

“Revenue” (without the quotes)

5. Set the font to **Bold** and the font size to **12pt** and **center** the text.
6. Insert another textbox.
7. Remove the textbox fill color and border.
8. Embed the following formula in the textbox:

=Calculation!O36

9. Set the font size to **14pt** and center the text.
10. Repeat Steps 6-8 with the following embedded formula in the textbox:

=Calculation!O39

11. Set the font size to **9pt**.
12. Repeat Steps 6-8 with the following embedded formula in the textbox:

=Calculation!P39

13. Set the font size to **9pt**.

Make sure you expand the widths of the textboxes to accommodate large numbers that may be displayed after data updates.

14. Position the items as desired using the below graphic as a guide:



Figure 10.3 - Finished KPI Display Box for Revenue

15. Group the items so they are treated as a single item.
16. Copy the revenue KPI and paste it twice.
17. Space the three identical KPIs next to one another in a side-by-side layout.
18. Update the titles for **Profit** and **Cash**.

19. Adjust the cell reference pointers of the duplicate KPIs to point to their respective calculations.
20. Change the background color of the middle KPI to the darker blue theme color.

The result should appear as follows:



Figure 10.4 - Complete 3 KPIs

21. To put an extra-special touch on the KPI list, add another textbox that will display the selected Division above the KPI block. Link the textbox to the following location:

=Calculation!N36



Figure 10.5 - Completed 3 KPIs with Division Heading

The finished KPI module should appear like the following graphic:

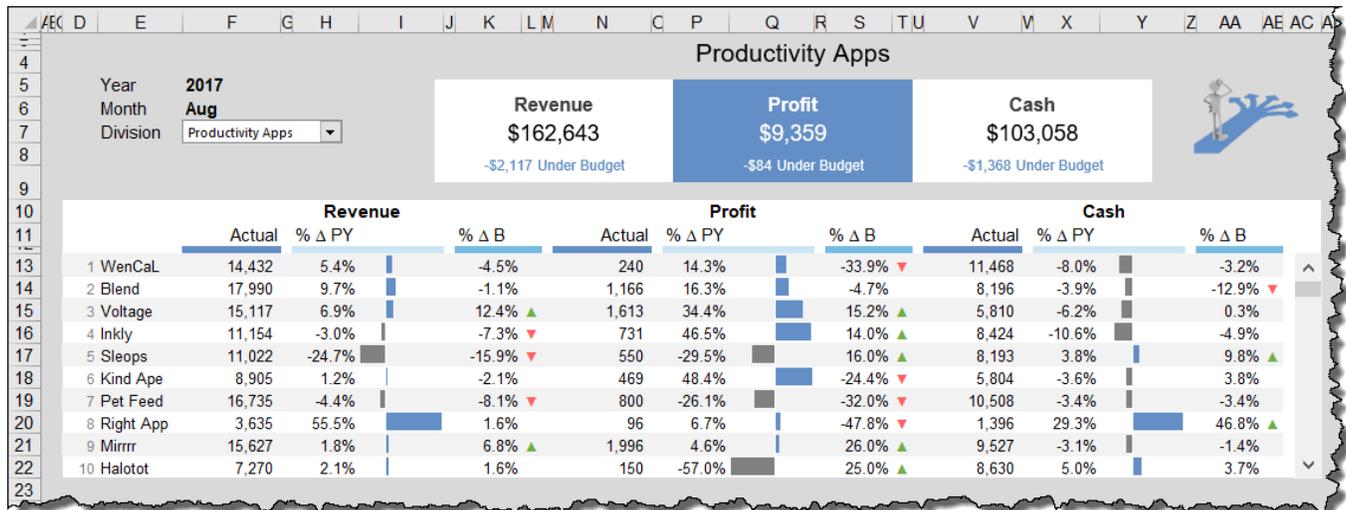


Figure 10.6 - Output of Module 10.2

Module 11

Top/Bottom KPI Based on Selection

MODULE 11.1 – SHOW TOP 6 MATCHES WITH UNIQUE IDENTIFIER

Data File:

11 - Top Bottom KPI Based on Selection\11.1 - Show Top 6 Matches with Unique Identifiers.xlsx

In this module, we will build a column chart that will display the top six or the bottom 6 apps for Actual versus Budget based on the selected Division and the selected KPI. The trick in this chart is to sort the apps in the chart based on the greatest to least percentage difference between Actual and Budget, not necessarily the app that has the greatest Actual value. We want to be able to identify exceptional and unexceptional performing apps using the Budget as a reference.

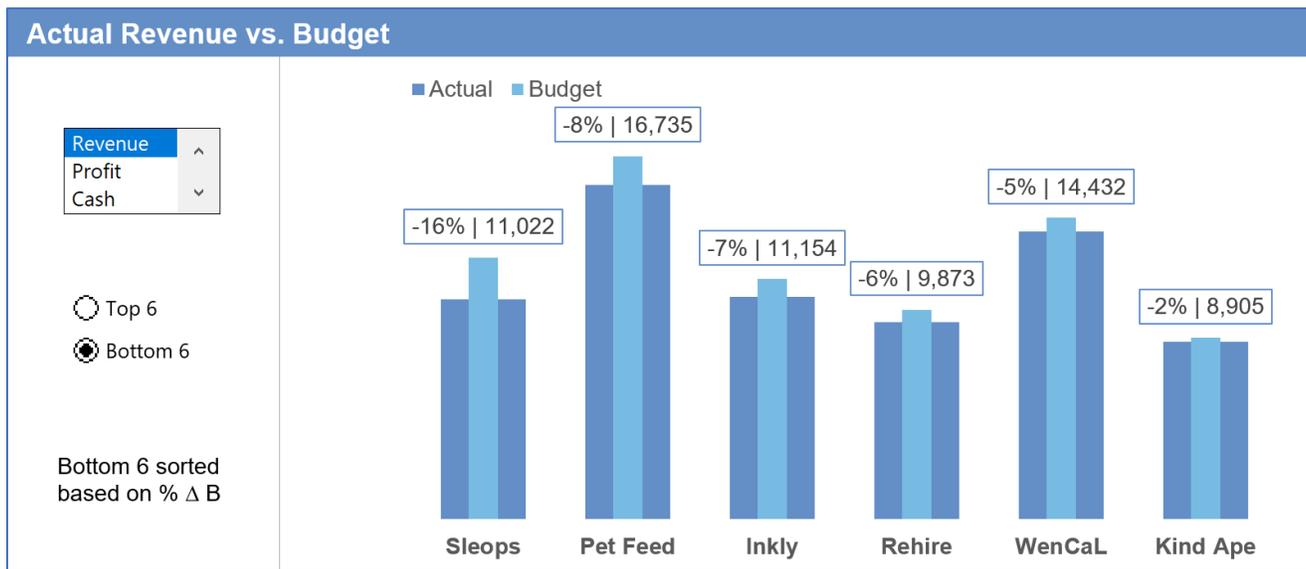


Figure 11.1 - Final Version of Actual Revenue vs. Budget Chart

The option for displaying Top 6 or Bottom 6 will be controlled by the user via form controls. Also, the selected KPI will also be a form-controlled user selection.

If you try to think about constructing all of this at the same time, you will quickly become overwhelmed and frustrated. With two different form controls, a dynamic heading, a dynamic note, and a non-standard chart with custom headings... this can quickly make even the most seasoned Excel user reach for the antacid tablets.

Break the process down into small, manageable pieces, tackle each piece one at a time, and before you know it, you will have reached the finish line.

Another tip is to start static. Don't worry about the form controls to select top/bottom 6 or Divisions. Use temporary, static data to get the chart working. Later, you can bring in the form controls to add interactivity to the chart.

What objects and calculations do we need to include in the chart?

We need the apps listed across the horizontal axis.

We need the Actual numbers and the Budget numbers. We will assume that our chart is only plotting the Revenue KPI and the Top 6 items.

We need the variance (or Delta Δ) for Budget.

1. On the **Calculation** sheet, select cell **AG14** and type the following:

Revenue % Δ B

So that it is easier to see what app is responsible for the percent change in Budget, we will repeat the selected app list from cells **N15:N34** and display them again in cells **AF15:AF34**. In this way, when the user selects a Division, the apps list in column **N** updates causing the app list in column **AF** to update.

2. In cell **AF15**, enter the below formula:

=N15

3. Fill this new formula down to row **34**.

4. In cell **AG15**, enter the below formula:

=AA15

5. Fill this new formula down to row **34**.

We need to derive a list of only the top six items based on the decreasing value of the percentage change to Budget. The LARGE function will be ideal for this operation because we can create 6 LARGE functions, each returning a different rank between 1 and 6. While the LARGE function will be able to locate a number based on a ranking position, we will nest it inside of a MATCH function to locate the position of that ranking number in a list. That position in the list can then be used by the INDEX function to return the app name. The result will be a nested (INDEX(MATCH(LARGE))) formula.

6. Select cell **AI15** and enter the following formula:

=INDEX(\$AF\$15:\$AF\$34,MATCH(LARGE(\$AG\$15:\$AG\$34,\$M15),\$AG\$15:\$AG\$34,0))

7. Fill the newly created formula down to row **20**.

Observe the problem. Because we have duplicate percentages in our list, the MATCH function stops at the first instance of the value being searched. This means that it may stop prematurely and cause the INDEX function to return the incorrect app name.

To overcome this problem, we will add a VERY TINY unique amount to each percentage value. This will result in each number in the list being unique. The number will be so small that it will have no mathematical effect on any calculation results.

8. Select cell **AG15** and modify it with the following information:

```
=AA15+(ROW()/1e10)
```

9. Fill the updated formula down to row **34**.

Because the rows that correspond to unused app positions are returning empty text, the addition of the tiny number to text is triggering a #VALUE! error. We will account for this by testing the cell with an IF function and returning empty text if the app position is empty.

10. Select cell **AG15** and modify it with the following information:

```
=IF(AF15="", "", AA15+(ROW()/10000000000))
```

Keep in mind, the numbers in column AG will not be displayed in the final chart. These numbers are only used to calculate the ranking so the LARGE (and ultimately SMALL) functions can determine ranking among the apps.

Because we will forget why we performed such an exotic operation, we could place a comment in the cell explaining our rationale, but we don't want to draw attention to the cell. This is where the "N" function comes to the rescue. Remember, the N function converts text to zero. If we add text to a function via an N function, it will convert the text to zero, add the zero to the original number, resulting in a net-zero change in the answer.

[For a review on the purpose and use of the N function, see example in Module 5 for details.](#)

11. Select cell **AG15** and modify it with the following information:

```
=IF(AF15="", "", AA15+(ROW()/10000000000)+N("the row function is used here to make sure we have unique values to sort"))
```

12. To calculate the sorted actuals, select cell **AJ15** and enter the following formula:

```
=INDEX($O$15:$O$34, MATCH($AI15, $N$15:$N$34, 0))
```

13. Fill the newly created formula down to row **20**.

To calculate the Budget amounts in column **AK**, we will copy (from the Formula Bar) the formula in cell **AL15** and paste it into cell **AK15**.

14. Edit the formula in cell AK15 to read as follows (*move column O to column U*):

```
=INDEX($U$15:$U$34,MATCH($AI15,$N$15:$N$34,0))
```

	AF	AG	AH	AI	AJ	AK
13	Basis - Unsorted			Sorted		
14	App	Revenue% Δ B		App	Actual	Budget
15	WenCaL	-5%		Voltage	15,117	13,455
16	Blend	-1%		Didactic	6,405	5,819
17	Voltage	12%		Silvrr	7,666	7,099
18	Inkly	-7%		Mirrrr	15,627	14,634
19	Sleops	-16%		Halotot	7,270	7,158
20	Kind Ape	-2%		Right App	3,635	3,579
21	Pet Feed	-8%				
22	Right App	2%				
23	Mirrrr	7%				
24	Halotot	2%				
25	Flowrrr	0%				
26	Silvrr	8%				
27	Dasring	1%				
28	Rehire	-6%				
29	Didactic	10%				
30						
31						
32						
33						
34						
35						
36						

Figure 11.2 - Output of Module 11.1

MODULE 11.2 – LIST BOX TO SELECT KPI FOR TOP 6 MATCHES

Data File:

11 - Top Bottom KPI Based on Selection\11.2 - List Box to Select KPI for Top 6 Matches.xlsx

In the last module, the KPI was fixed as “Revenue”. In this module, we want to create dynamic way of allowing the user to select the KPI and have the Top 6 / Bottom 6 list reflect that KPI selection. The List Box Form Control is an excellent choice for this objective.

We will leave the top/bottom 6 option static for now and focus on the KPI selection. The following module will tackle the problem of a user selectable top/bottom 6 form control.

Even though the list box will be used on the Dashboard sheet, we will build and test the list box control on the Calculation sheet where it is easier to monitor. Once the list box is working properly, it will be moved to the Dashboard sheet.

The first step is to add and configure the list box.

1. On the **Calculation** sheet, add a list box and configure it with the following parameters:

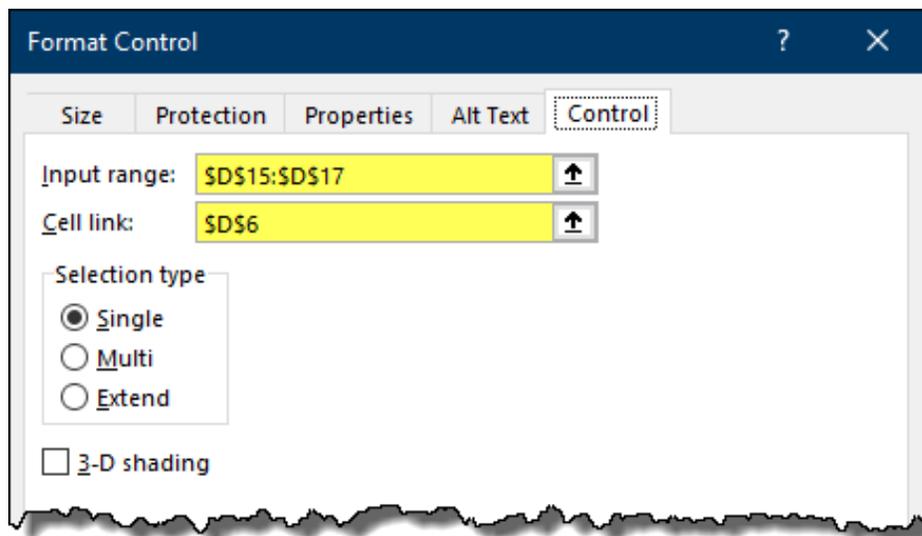


Figure 11.3 - Parameters for Step 1, Module 11.2

[For a review on the purpose and use of the List Box form control, see example in Module 6 for details.](#)

Test the new form control to ensure it is generating the proper numbers based on the user selection (“Revenue” = 1, “Profit” = 2, “Cash” = 3).

2. In cell **AG15**, update the formula by replacing the reference to “**AA15**” with the following INDEX function:

`INDEX(AA15:AC15,D6)`

The updated formula will appear as the following:

```
=IF(AF15="", "", INDEX(AA15:AC15, $D$6) + (ROW() / 10000000000) + N("the row function is used here to make sure we have unique values to sort"))
```

- Fill the new formula down to row **34**.
- In cell **AJ15**, update the formula to reflect the following:

```
=INDEX($O$15:$Q$34, MATCH($AI15, $N$15:$N$34, 0), $D$6)
```

- Fill the new formula down to row **20**.
- In cell **AK15**, update the formula to reflect the following:

```
=INDEX($U$15:$W$34, MATCH($AI15, $N$15:$N$34, 0), $D$6)
```

- Fill the new formula down to row **20**.
- Update the title by selecting cell **AG14** and change the formula to reflect the following:

```
=INDEX(AA14:AC14, D6) & " Δ B"
```

We will now create a formula that will serve as a dynamic heading for the column chart.

- Select cell **AJ13** and enter the following formula:

```
= "Actual " & INDEX(O14:Q14, D6) & " vs. Budget"
```

	AF	AG	AH	AI	AJ	AK
13	Basis - Unsorted			Sorted		
14	App	Revenue% Δ B		App	Actual	Budget
15	WenCaL	-5%		Voltage	15,117	13,455
16	Blend	-1%		Didactic	6,405	5,819
17	Voltage	12%		Silvrr	7,666	7,099
18	Inkly	-7%		Mirrrr	15,627	14,634
19	Sleeps	-16%		Halotot	7,270	7,158
20	Kind Ape	-2%		Right App	3,635	3,579
21	Pet Feed	-8%				
22	Right App	2%				
23	Mirrrr	7%				
24	Halotot	2%				
25	Flowrrr	0%				
26	Silvrr	8%				
27	Dasring	1%				
28	Rehire	-6%				
29	Didactic	10%				
30						
31						
32						
33						
34						
35						
36						

Figure 11.4 - Output of Module 11.2

MODULE 11.3 – OPTION BUTTON TO TOGGLE BETWEEN TOP OR BOTTOM 6

Data File:

11 - Top Bottom KPI Based on Selection\ 1.3 - Option Button to Toggle Between Top and Bottom 6.xlsx

In this module, we will create the form control interface that will allow the user to toggle between the Top 6 apps and the Bottom 6 apps. Since we are only switching between two options, we are creating a binary state of one or the other. We could do this with a single check box form control (checked for TOP and unchecked for BOTTOM), but that can be confusing to the user. A better option (no pun intended) is to use Option Button Form Controls.

[For a review on the purpose and use of the Option Button form control, see example in Module 6 for details.](#)

By creating two option buttons, the buttons will automatically communicate with one another. If one button is selected, the opposing button will clear its selected state. This process will reverse when the opposing button is selected.

Both option buttons will be pointing to the same cell in the spreadsheet to store their status. If one button is selected, the cell will record a 1. If the opposing button is selected, the cell will record a 2.

1. On the **Calculation** sheet, add an option button form control and configure it with the following parameters:

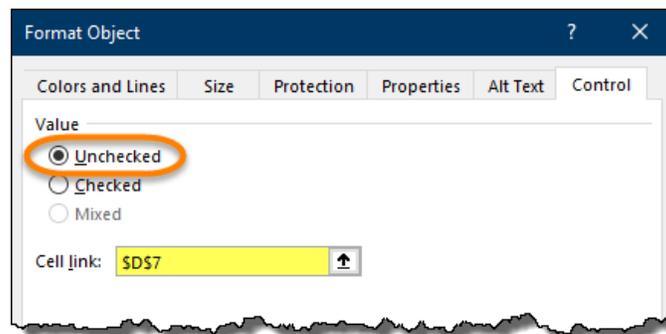


Figure 11.5 - Parameters for Step 1, Module 11.3

2. Rename the displayed text to “**Top 6**”.
3. Add a second option button form control and configure it with the following parameters:

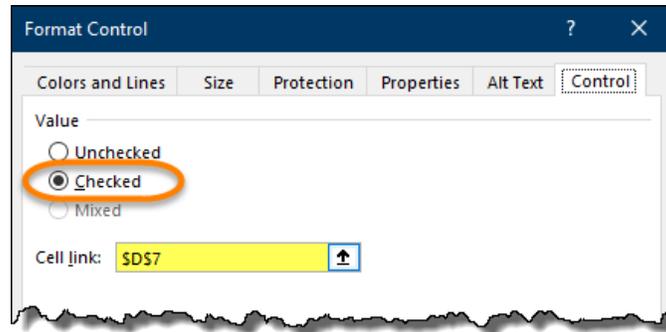


Figure 11.6 - Parameters for Step 2, Module 11.3

4. Rename the displayed text to “**Bottom 6**”.

Test the new form controls to ensure they are generating the proper numbers based on the user selection

(*Top = 1, Bottom = 2*).

To have our top/bottom formulas react to the user’s selection of the option buttons, we need to have two possible scenarios in place; one to search for apps with the LARGE function, and the other to search for apps with the SMALL function.

With the option buttons generating a 0 or a 1, we can assign 0 to the LARGE function and 1 to the SMALL function.

[For a review on the purpose and use of the LARGE and SMALL functions, see Module 5 for details.](#)

The CHOOSE function would be ideal in this scenario because it would allow us to execute a LARGE function if the user selects a button that generates a 0, and a SMALL function if the user selects a button that generates a 1.

[For a review on the purpose and use of the CHOOSE function, see example in Module 5 details.](#)

We need to rewrite the function that is hardcoded to run a LARGE function and make it more dynamic. We want the formula to be able to switch between executing a LARGE function and executing a SMALL function.

5. Select cell **A15** and update the formula with the following changes:

```
=INDEX($AF$15:$AF$34,MATCH(CHOOSE($D$7,LARGE($AG$15:$AG$34,$M15),SMALL($AG$15:$AG$34,$M15)), $AG$15:$AG$34,0))
```

It may look daunting, but if you look closely, we merely replaced the LARGE function...

```
LARGE($AG$15:$AG$34,$M15)
```

...with a CHOOSE function that uses the existing LARGE function but incorporates a second option in the form of a SMALL function.

```
CHOOSE($D$7,LARGE($AG$15:$AG$34,$M15),SMALL($AG$15:$AG$34,$M15))
```

The use of a CHOOSE function is more efficient than writing a two-level nested IF statement that would check for 1 and then check for 2.

We now need to create a formula that will generate a dynamic note that will be placed below the list box and option buttons in the dashboard.

6. Select cell **AJ12** and enter the following formula:

```
=CHOOSE(D7,"Top 6","Bottom 6")&" sorted based on % Δ B"
```

Test the new dynamic note, and the newly updated app list to check for responsiveness and accuracy. As a byproduct, the Actual (column AJ) and the Budget (column AK) should be changing when the top/bottom controls are used.

MODULE 11.4 – ACTUAL TO BUDGET COMPARISON – SORTED BARS WITH ERROR BAR TECHNIQUE

Data File:

11 - Top Bottom KPI Based on Selection\11.4 - Actual to Budget Chart with Sorting.xlsx

Now that we have setup the calculations and form controls to feed and customize the chart, the next step is to generate the Actual column chart.

Remember: The form controls manipulate the data; the updated data will feed the chart and the chart will indirectly react to the form controls.

1. On the **Calculation** sheet, select cells **A114:AK20** and insert a column chart.

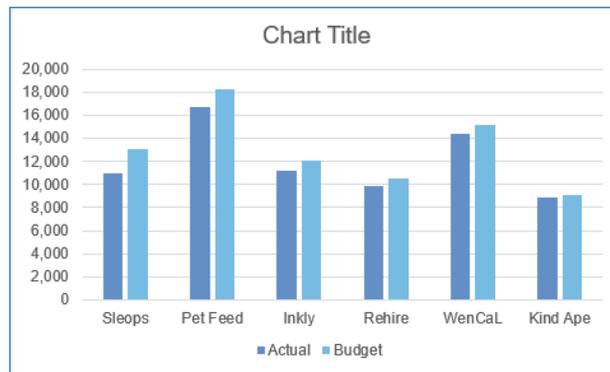


Figure 11.7 - Chart Results for Step 1, Module 11.4

2. To make the chart easier to read, declutter the chart by removing the following chart elements:

- Chart title
- Y axis
- Gridlines
- Chart border

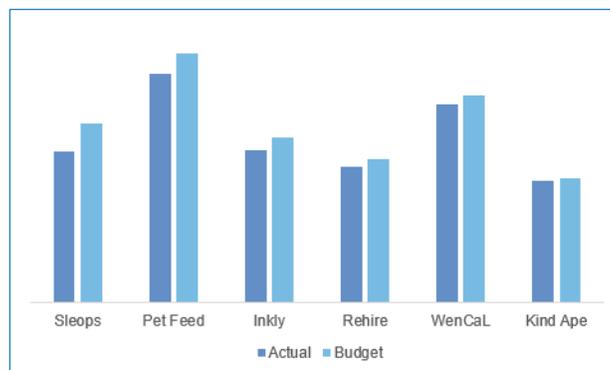


Figure 11.8 - Chart Results for Step 2, Module 11.4

The effect we are going for is to have the bars stacked one behind the other. We want the darker bars for Actual to be in the back and the lighter bars for Budget to be in the front. We also don't want to have the situation where an app comes in under Budget and its dark bar is hidden by the light bar.

To achieve the desired effect, we will assign the Budget series to the Secondary Vertical Axis. Then we will manipulate the Gap Width of the two series to make the Actual bars "fat" and the Budget bars "skinny".

3. In the chart, right-click any light-blue line (*Budget series*) and select **Format Data Series**.
4. In the control panel on the right, set the series to plot on the **Secondary Axis** and set the **Gap Width** to **390**.
5. Select any dark-blue line (*Actual series*) and set the Gap Width to **70**.
6. Select the numbers on the **Secondary Vertical Axis** and remove them from the chart.
7. Perform the following chart modifications
 - Move the legend to the upper-left corner of the chart
 - Make the Horizontal Axis titles bold
 - Remove the **Horizontal Axis** line (set the line option to "no line")
 - Add data labels to the **Actual** series
 - Resize the plot area to use the space at the bottom where the legend used to reside

The results up till now should appear like the following chart.

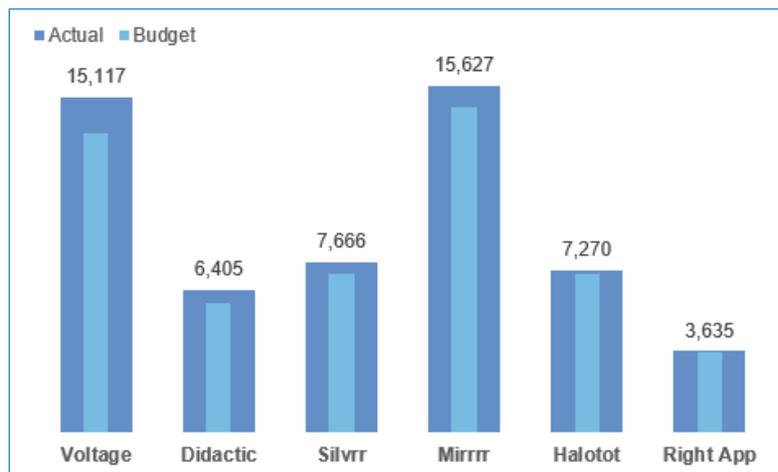


Figure 11.9 - Chart Results for Step 7, Module 11.4

Unfortunately, the chart does not look too appealing when the user selects the option for Bottom 6 apps.

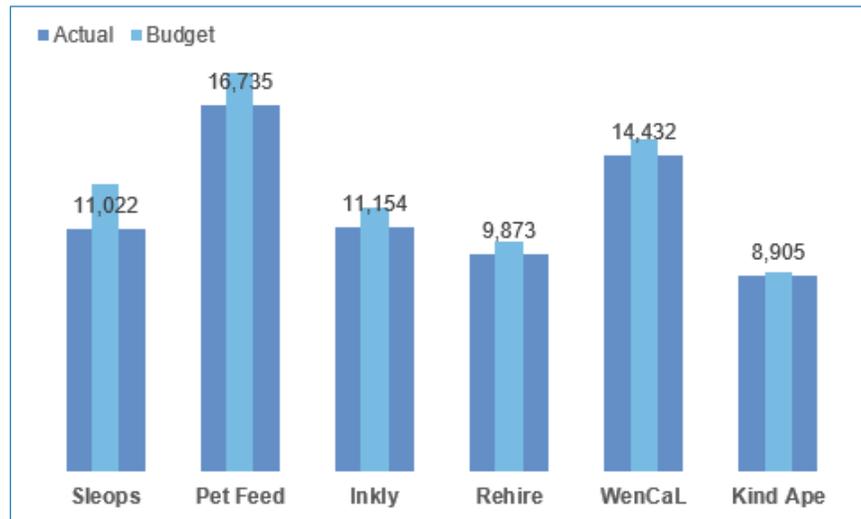


Figure 11.10 - Chart Results when viewing Bottom 6 Apps

We need to create, and bring into the chart, a new set of data labels that we have control over when it comes to placement in the chart.

8. Remove the existing data labels.
9. Select cell **AL15** and enter the following formula:

=MAX(AJ15:AK15)

10. Fill the new formula down to row **20**.

Because we always want the data labels to be above the largest bar for any given app, the MAX function will determine which bar, Actual or Budget, is the tallest bar.

Now we want to create a formula that will build a custom data label that includes the Revenue% from column AA and the Actual value from column AJ. We want to separate these two values with a custom delimiter; in this case, the “pipe” character “|”. Because we are concatenating text and numbers together, we can’t use traditional Number Formatting to get the revenue percentages and actuals to have any meaningful style. This will be accomplished via the TEXT function.

[For a review on the purpose and use of the TEXT function, see example in Module 5 details.](#)

11. Select cell **AM15** and enter the following formula:

=TEXT(INDEX(\$AA\$15:\$AC\$34,MATCH(AI15,\$N\$15:\$N\$34,0),\$D\$6),"0%") & "|" & TEXT(AJ15,"#,##0")

12. Fill the new formula down to row **20**.
13. Add the **Label Position** numbers to the chart by right-clicking on the chart and click **Select Data**. Add the new data series with the following parameters:

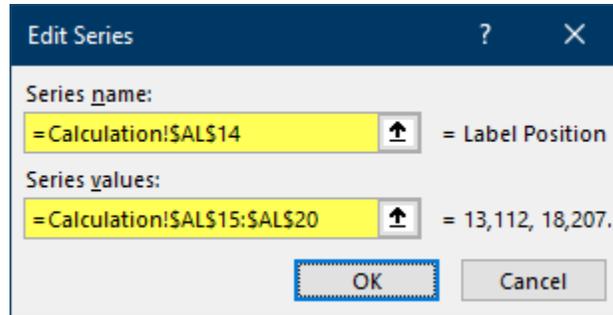


Figure 11.11 - Parameters for Step 13, Module 11.4

14. Select the new **Label Position** series and convert it to a **scatter** chart.
15. Select the **Scatter Chart** dots and add **data labels**.
16. Hide the **Scatter Chart** dots.
17. Select the **data labels** and move them **above** the bars.
18. Resize the chart **plot area** to allow room at the top of the chart for the data labels.
19. Replace the displayed **Y Value** numbers in the data labels with the **Value From Cells** for cells **AM15:AM20**.
20. Add a border to the data labels (*thin, pale-blue line*).
21. Remove “Label Position” from the legend.

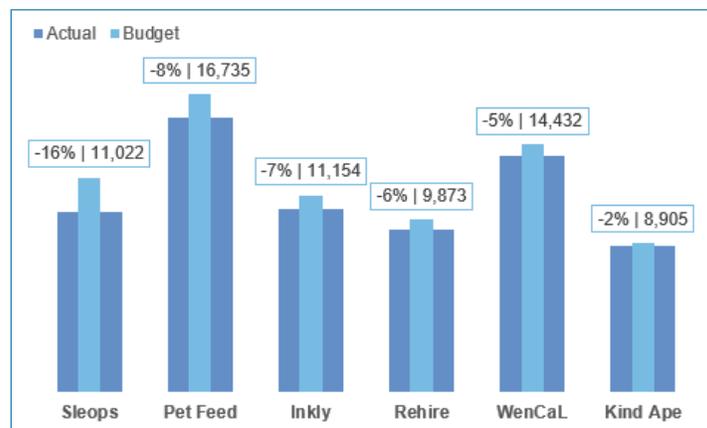


Figure 11.12 - Final Version of Chart for Module 11.4

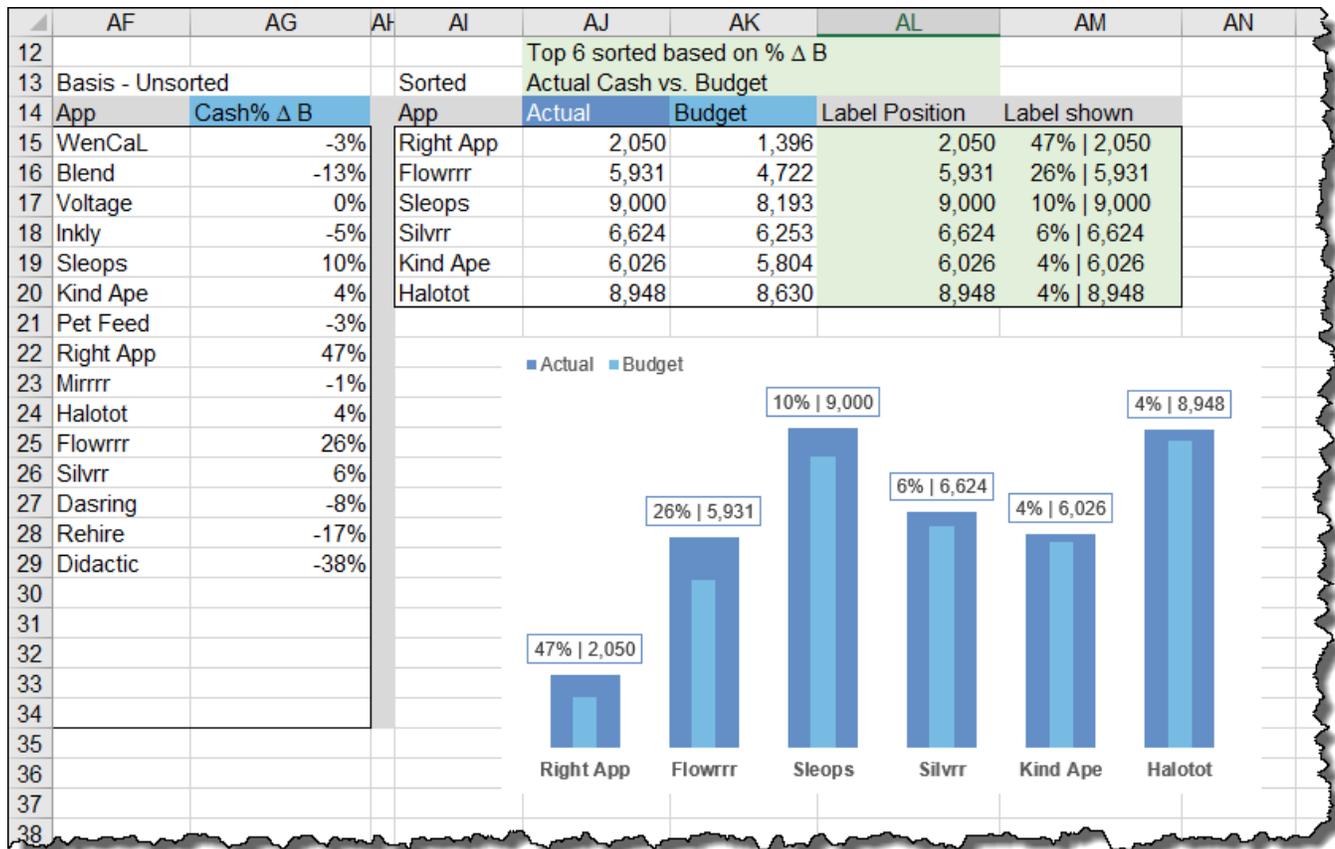


Figure 11.13 - Output of Module 11.4

NOTE: The accompanying video lecture for Module 11.4 demonstrates an alternative method of creating the columns in the above chart using Error Bars. It is worth your time to try this method as it will provide experience using Error Bars and other Excel features in a non-standard way.

Becoming creative with your use of the provided Excel tools, beyond the tool’s intended use, can not only be fun but also provides deeper insight into the way Excel manages data. This will no doubt lead to solutions to problems that were once thought impossible to overcome.

MODULE 11.5 – FINALIZE TOP/BOTTOM REPORT ON DASHBOARD

Data File:

11 - Top Bottom KPI Based on Selection\11.5 - Finalize Top Bottom Report on Dashboard

The final step in creating this piece of the Dashboard is to move all the developed elements to the Dashboard sheet and position them in a visually appealing and easy to use layout. Feel free to experiment with grouping and alignment controls to expedite positioning and ensure consistency in your layout.

1. Copy and paste the following items to the **Dashboard** sheet:
 - **List** box for KPIs
 - Option buttons for Top 6 and Bottom 6
 - Actual vs. Budget **column chart**

NOTE: These items can be deleted from the **Calculation** sheet once we are certain we have everything in place. Leaving a copy in the original location provides us with a “safety net” in case an object becomes corrupt or misconfigured.

One of the downfalls of working from the **Calculation** sheet and then moving the objects to the **Dashboard** sheet when finished is that all the reference pointers in the form control properties will be pointing to the cells on the **Dashboard** sheet.

2. Reset all the cell pointer properties back to the original cell locations on the **Calculation** sheet.
3. Remove the **outline** and **shape fill** of the bar chart.
4. Select cells **D25:Q38** and change the cell fill color to **white**.
5. Select cells **E25:E38** and add a light gray line to the right side of the selected cells.
6. Add a textbox to the chart and link it to cell **AJ12** on the **Calculation** sheet.
7. Set the font size of the newly added textbox to **9 points**.
8. Adjust the size and positioning to reflect the finished chart as best as possible.
9. Add a textbox and place it directly over cells **D24:Q24**.
10. Link the textbox to cell **AJ13** on the **Calculation** sheet.

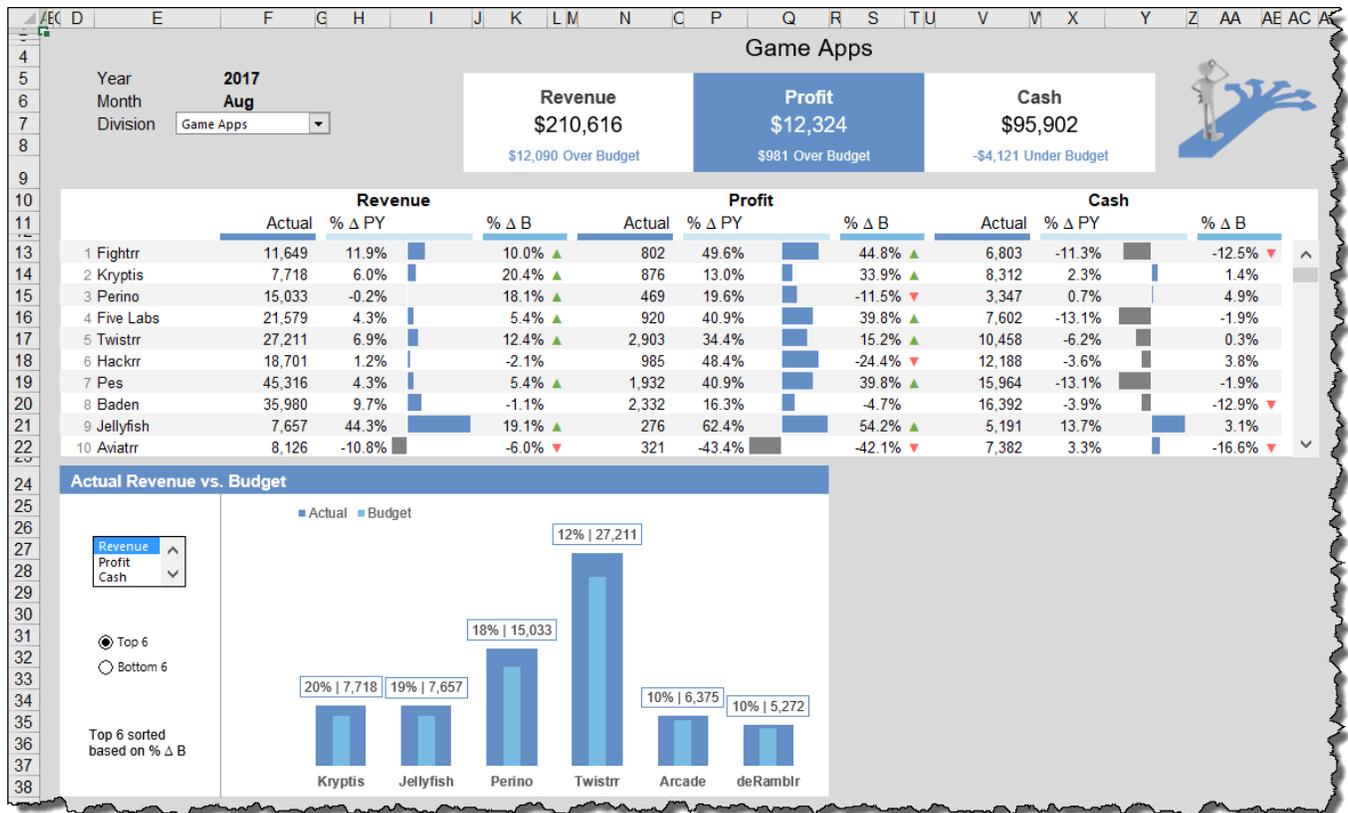


Figure 11.14 - Output of Module 11.5

Module 12

Scrolling Line Chart with Boxes

MODULE 12.1 – COMPLEX LOOKUP TO PREPARE DATA FOR SCROLLING CHART

Data File:

12 - Scrolling Line Chart with Check Boxes\12.1 - Complex Lookup to Prepare Data for Scrolling Chart.xlsx

We are almost finished with creating the dashboard. The final major element is a line graph that will graph the profit for a selected app over the course of a calendar year.

The elements we will be including in our line graph are as follows:

- Calendar months from January to December displayed along the horizontal axis (J through D)
- Lines that chart the actuals for the selected KPI
 - Actual
 - Previous Year (PY)
 - Budget (B)
- A form control (scroll bar) to allow the user to select a specific app
- Two form controls (check boxes) to allow the user to toggle on or off the visibility of the lines for Previous Year and Budget

The completed line graph will appear as follows:



Figure 12.1 - Results of Line Graph for Module 12

NOTE: When adding the months to the horizontal axis, ensure that the month names match the spelling style of the month names in the data (i.e. January, Jan, J, 1, etc.) This ensures that the months will return information when queried.

The data will be coming from the following columns on the Data sheet:

Actuals..... columns K:V

Previous year..... columns W:AH

Budget columns AI:AT

We will not worry about the dynamic nature of the form controls for the first portion of chart development. As with the chart from Module 11, we will assume a fixed value at first, just to ensure the chart is working properly with static information. After that, the form controls will be added to bring the chart to life.

1. On the **Calculation** sheet, select cell **AP11** and enter “**Game Apps**”.
2. Select cell **AQ13** and enter the following formula:

```
=cur_year
```

3. Select cell **AR13** and enter the following formula:

```
=AQ13-1
```

4. Select cell **AS13** and enter the following formula:

```
=cur_year
```

The next step is to write a formula that will retrieve the actuals for the selected year, month, metric and app name. As with the original scrolling table in Module 9, we need to have a unique key that allows precise data retrieval from the data sheet.

If we used the tactic of the scrolling table, we could build a “helper row” that would create a unique key by concatenating the year, month, and metric (Actual, Previous Year, or Budget.) We would have to build this helper data on both the Calculation sheet as well as the Data sheet.

To keep from building too many helper rows, another approach is to build an array formula that performs the unique key construction directly in the formula.

5. Select cell **AQ15** and enter the following formula:

```
=INDEX(Data!$K$9:$AT$60,MATCH($AP$11,Data!$A$9:$A$60,0),MATCH(AQ$13&$AP15&AQ$14,INDEX(Data!$K$5:$AT$5&Data!$K$6:$AT$6&Data!$K$7:$AT$7,0))
```

If we break down the logic of the above formula, the steps are as follows:

- A. Locate the app in the data based on the selection made in **\$AP\$11**. This is performed by the MATCH function (*down rows*). This returns a **“17”** (*“Game Apps” is on the 17th row of the table.*)
- B. Concatenate the year (**AQ13**) to the month (**AP15**) to the metric “Actual” (**AQ14**). This creates a unique key **“2017JanActual”**. The next MATCH function needs to locate that key across the top of the **Data** sheet (*across columns*). Because we don’t have a helper row with such information, we will use an INDEX function to search across columns (**K:AT**) but at the same time search three simultaneous rows (**5:7**). This will return a column position number; in this first case, it returns the number **“1”**.

Because the first MATCH function found the 17th column, and the second MATCH function found the 1st row, the INDEX function can navigate to the coordinates (**row1:col17**) to return a value. (*this is cell K25 on the Data sheet.*)

- 6. Fill the newly created “masterpiece” formula down to row **26** and then across to column **AS**.

The resultant list of values should appear as follows:

Game Apps			
	2017	2016	2017
Month	Actual	Actual	Budget
Jan	1,523	1,228	1,470
Feb	3,030	2,469	2,921
Mar	4,948	4,022	4,767
Apr	7,267	5,341	6,226
May	10,246	6,927	8,273
Jun	12,324	8,527	10,141
Jul	0	10,424	12,032
Aug	0	10,993	12,477
Sep	0	12,312	14,358
Oct	0	13,499	15,805
Nov	0	14,643	17,386
Dec	0	15,392	17,750

Figure 12.2 - Final figures for Step 6, Module 12.1

MODULE 12.2 – SCROLL-ABLE LINE CHART

Data File:

12 - Scrolling Line Chart with Check Boxes\12.2 - Scroll-able Line Chart.xlsx

We would like for the user to be able to select an app via a form control which will feed one of the line graph parameters. To make the chart dynamic, a Scroll Bar Form Control will be an effective way for the user to “walk” through the list of apps for a selected Division.

1. On the **Calculation** sheet, insert a scroll bar and configure it with the following parameters:

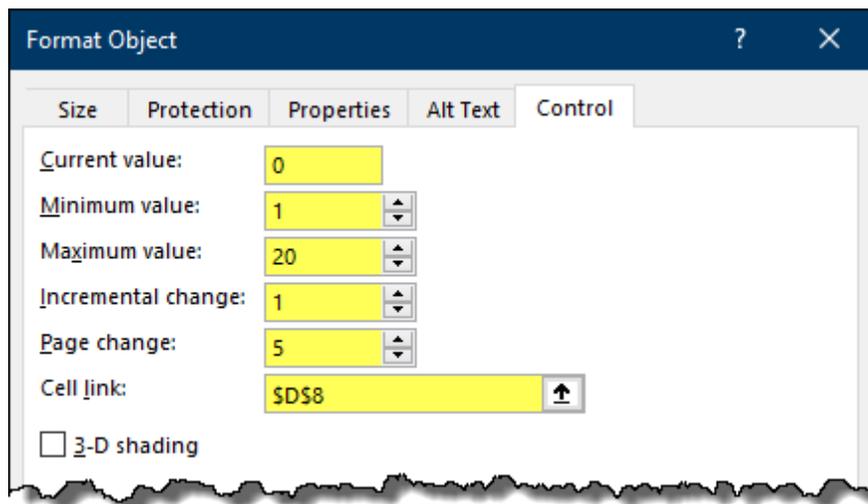


Figure 12.3 - Parameters for Step 1, Module 12.2

The challenge now is to get the scroll bar form control to change the value in cell **AP11** on the **Calculation** sheet. What makes this tricky is that the list of apps we want to search is dependent on the selected **Division** from the form control on the **Data** sheet.

Since we have an existing matrix of apps by Division (cells **G14:I34** on the **Calculation** sheet), we can use the scroll control selection result in cell **D8** to navigate **DOWN** the list to find the app, and we can use the dropdown list selection results in cell **D4** to navigate **ACROSS** the list to find the Division.

2. In cell **AP11**, enter the following formula:

```
=INDEX(G14:I34,D8,D4)
```

To get a graphical version of these numbers, we will now insert a line graph and test the scroll control to see if the line graph changes when the user interacts with the scroll control.

3. Select cells **AP14:AS26** and insert a line graph.

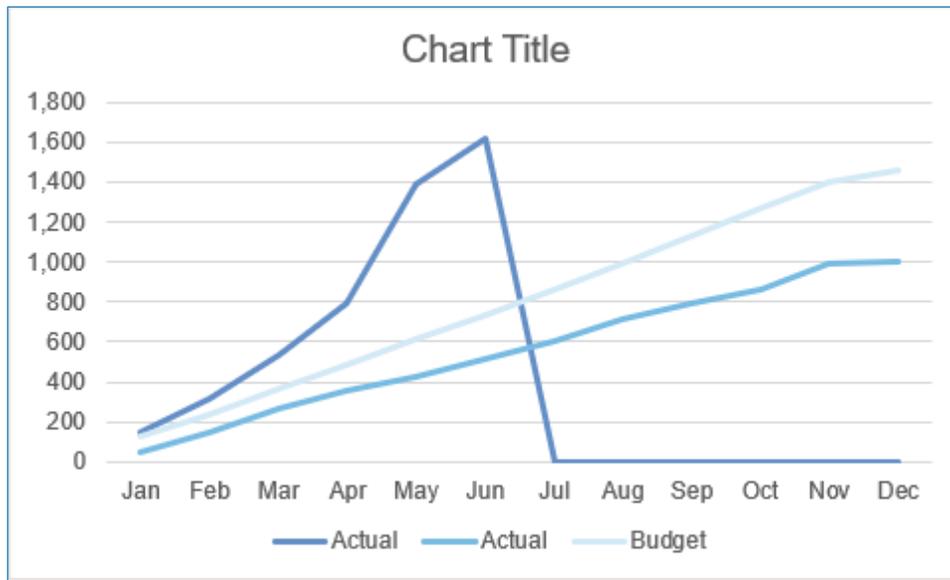


Figure 12.4 - Results of Line Graph for Step 3, Module 12.2

Notice that the line for Actual crashes after the last selected month.

We also want to have the ability to turn on and off the display of the Previous Year and Budget lines.

And finally, we want the series labels to be displayed at the ends of the lines. Normal data labels will not work the way we want (you will see why soon), so this will require the creation of custom label position.

To get these abilities, we will have to create a data preparation table to rework the chart source data before feeding it to the chart.

MODULE 12.3 – TRICK TO AVOID CRASHING LINES IN LINE CHART

Data File:

12 - Scrolling Line Chart with Check Boxes\12.3 - Avoid Line Crash in Line Charts.xlsx

In this module, we will overcome the visual shortcomings of the previous modules chart result.

To recap the problems and objectives:

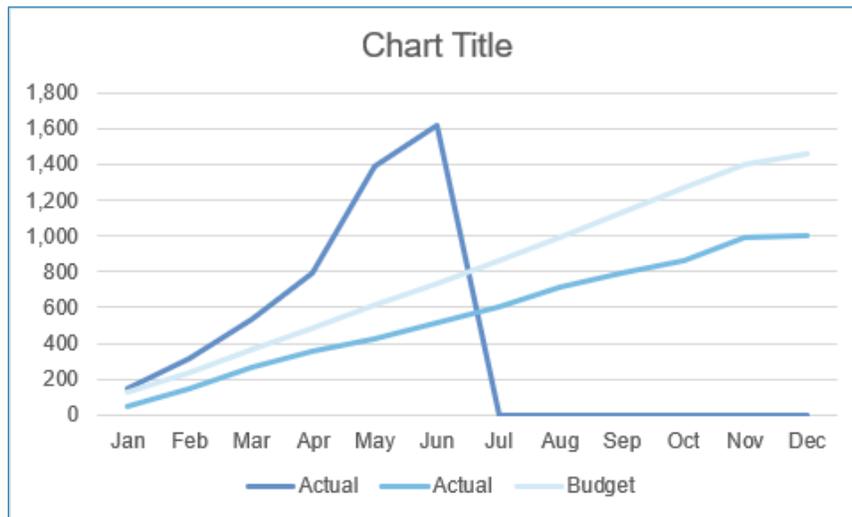


Figure 12.5 - Chart Result from Module 12.2

The line for Actual crashes after the last selected month.

We want to have the ability to turn on and off the display of the Previous Year and Budget lines.

Series labels need to be displayed at the ends of the lines.

To get these abilities, we will have to create a data preparation table to rework the chart source data before feeding it to the chart.

1. On the **Calculation** sheet, select cell **AV14** and enter the following formula:

```
=AP14
```

2. Select cells **AQ13:AS14** and copy / PASTE LINK to cells **AW13:AY14**.

We want the list of months across the horizontal axis to display only the first letter of the month. We will create a new list of axis labels by extracting the first character from each month using the LEFT function.

3. In cell **AV15**, enter the following formula:

```
=LEFT(AP15,1)
```

- Fill the newly created formula down to row **26**.

The next calculations are to solve the following problem:

We need to ensure that the line graph for Actual displays no more months than are selected by the user from the dropdown list on the **Dashboard** sheet in cell **F6** (*cur_month*).

One of the commonly used “tricks” to hiding data points in a chart is to use the **NA()** function. The NA function generates an error that is ignored by charts.

- Select the chart and move the chart’s data range from **AQ15:AS26** to **AW15:AY26**. (TIP: this can be performed graphically by clicking and holding on the blue-box border and dragging it to the new position.)
- Move the chart’s month labels by right-clicking on the chart and click **Select Data**. Unfortunately, this cannot be accomplished graphically like the previous step.

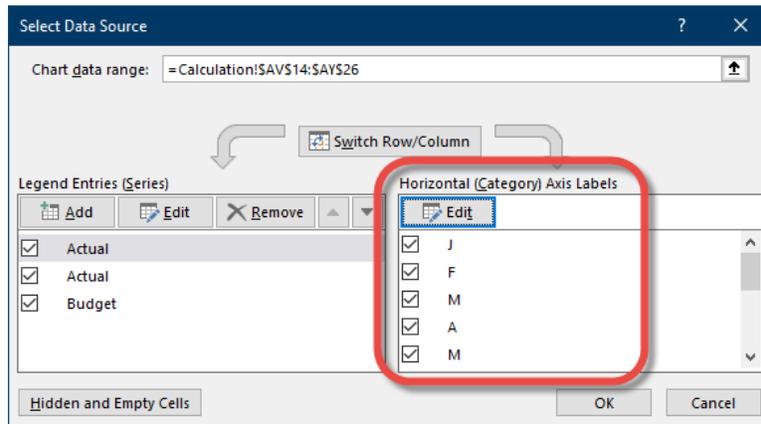


Figure 12.6 - Results for Step 6, Module 12.3

We will now create the formulas that will pull data from the original data preparation table in cells **AP13:AS26**. This formula will check to see if the month selected by the user (*cur_month*) is greater than or equal to the month row in the data preparation table. If the selected month is greater than the table row month, retrieve the data, otherwise hide the data with the **NA()** function.

- Select cell **AW15** and enter the following formula:

```
=IF(MATCH(cur_month,$AP$15:$AP$26,0)>=ROWS($AP$15:AP15),AQ15,NA())
```

To see if the Previous Year and Budget columns are working, enter the following temporary formulas.

- Select cell **AX15** and enter the following formula:

```
=AR15
```

- Fill the newly created formula to row **26**.

10. Select cell **AY15** and enter the following formula:

```
=AS15
```

11. Fill the newly created formula to row **26**.

The chart should now be looking quite a bit cleaner.

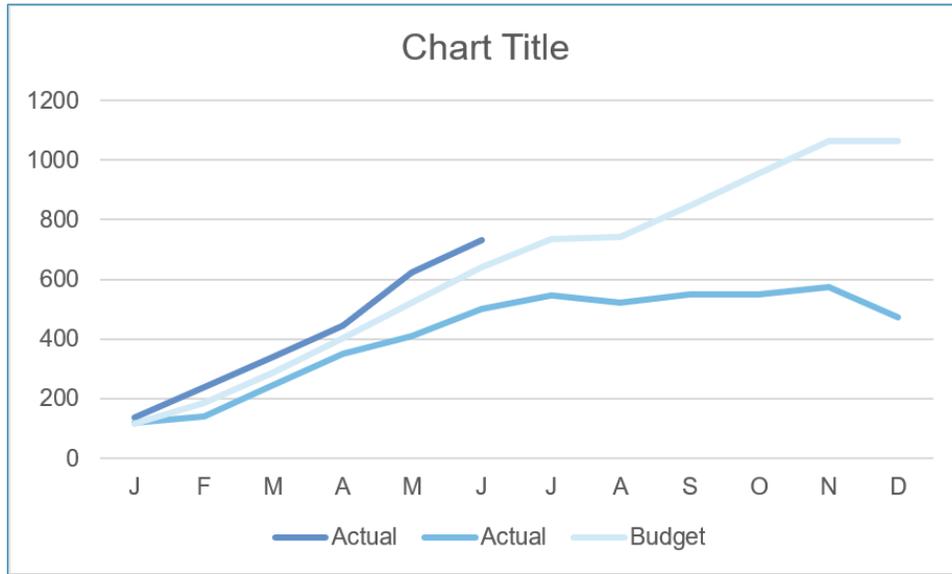


Figure 12.7 - Updated Graph Without Crashing Actual Line

The output at this stage should appear as follows:

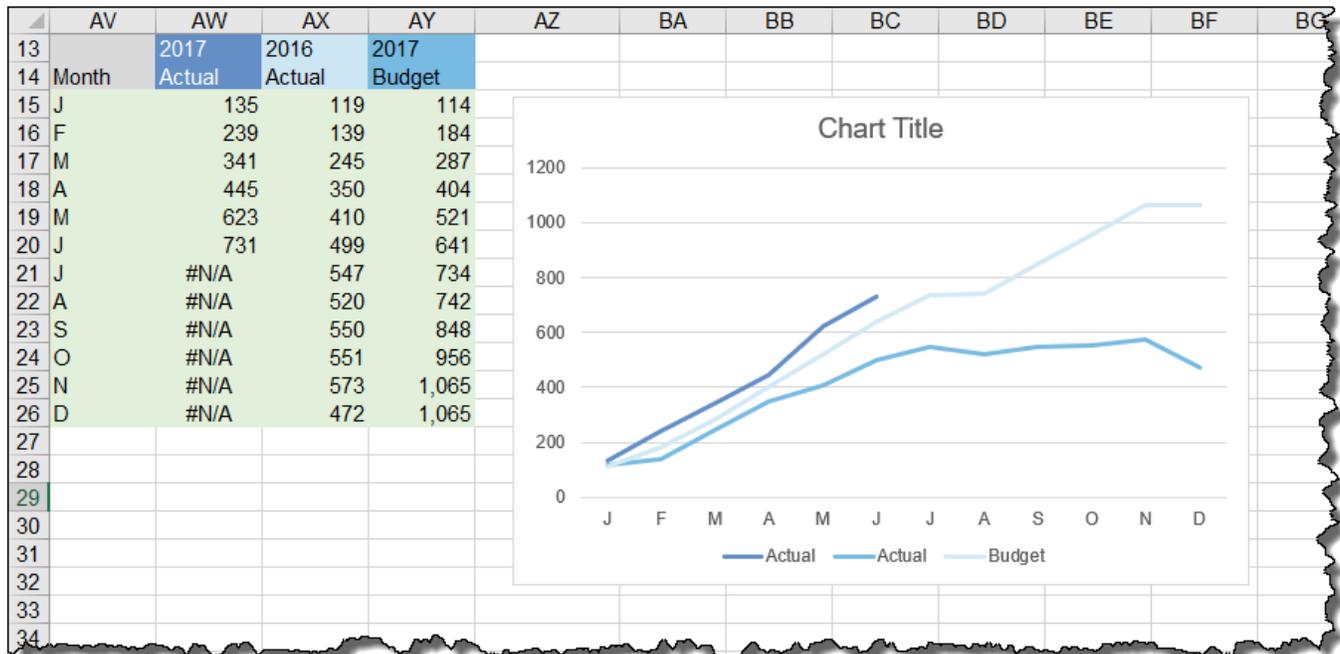


Figure 12.8 - Output of Module 12.3

MODULE 12.4 – CHECK BOX TO CONTROL SERIES VISIBILITY (HIDE OR SHOW)

Data File:

12 - Scrolling Line Chart with Check Boxes\12.4 - Check Boxes to Control Series Visibility.xlsx

In this module, we will add the check boxes to allow the user to toggle the Previous Year and Budget lines on and off from the line graph.

1. Add a **Check Box Form Control** to the chart with the following parameters:

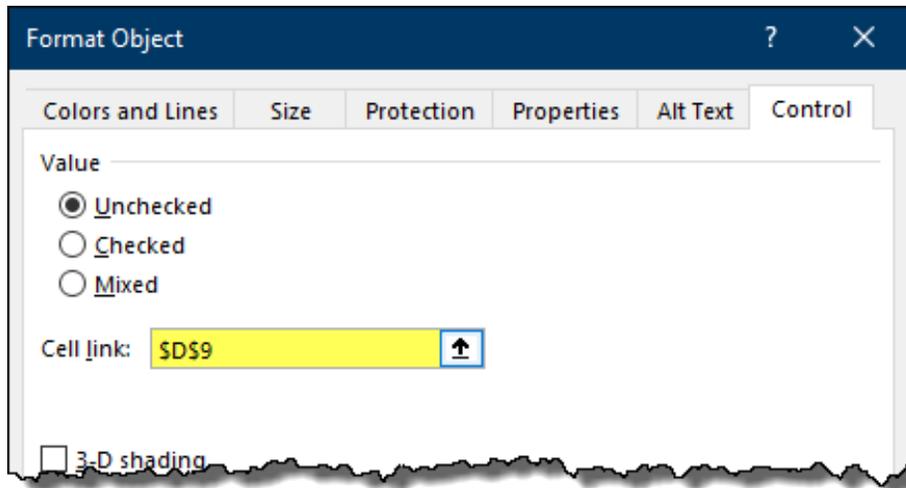


Figure 12.9 - Parameters for Step 1, Module 12.4

2. Rename the check box title to “PY”.
3. Add a second **Check Box Form Control** to the chart with the following parameters:



Figure 12.10 - Parameters for Step 3, Module 12.4

4. Rename the check box title to “Budget”.

5. Select cell **AX15** and update the formula as follows:

```
=IF($D$9,AR15,NA())
```

6. Select cell **AX15** and update the formula as follows:

```
=IF($D$10,AS15,NA())
```

Test the check boxes to ensure they are functioning properly.

Select different months from the Dashboard dropdown list to ensure the NA() function is properly suppressing months that are greater than the selected month.

MODULE 12.5 – LINE SERIES TWEAKS FOR BETTER READABILITY

Data File:

12 - Scrolling Line Chart with Check Boxes\12.5 - Line Series Tweaks for Better Visibility.xlsx

In this module, we will finalize the look of the profit chart to display the labels at the end of the series. This will make it easier for the reader to recognize which lines pertain to which metrics than consulting a legend.

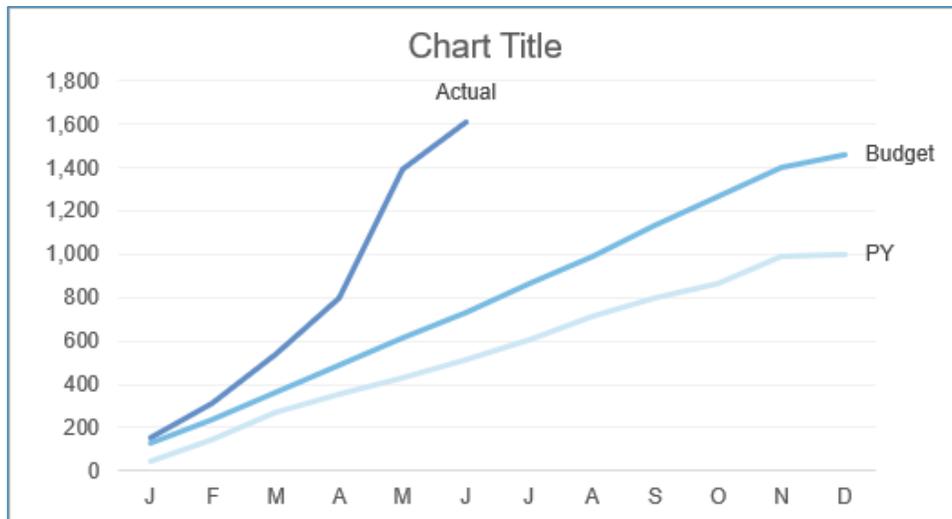


Figure 12.11 - Final Graph Results for Module 12.5

We have a minor issue with the chart. The colors for the Previous Year and the Budget are incorrect; Budget should be a medium-blue while Previous Year should be a light-blue. We could move the data around in the chart to get the theme colors to match, but in this case, it is easier just to click on the lines and set their colors the way they were intended.

1. Click on the light-blue line (Budget) and set its color to the medium-blue theme color.
2. Click on the medium-blue line (Previous Year) and set its color to the light-blue theme color.

Another issue we must address is that our headings for the data read “Actual-Actual-Budget”. We need the headings to read “Actual-PY-Budget”. This will be force-corrected (*no, we’re not talking about some Jedi mind power*) to replace the incorrect heading with a new, correct heading.

Because we only want to add a data label to the LAST data point on the line, we must be VERY careful how we select data points before activating the labels. This will require a little-known trick used when selecting chart elements.

3. Select the pale-blue line for **Previous Year**. This will select **ALL** data points on the **PY** line.

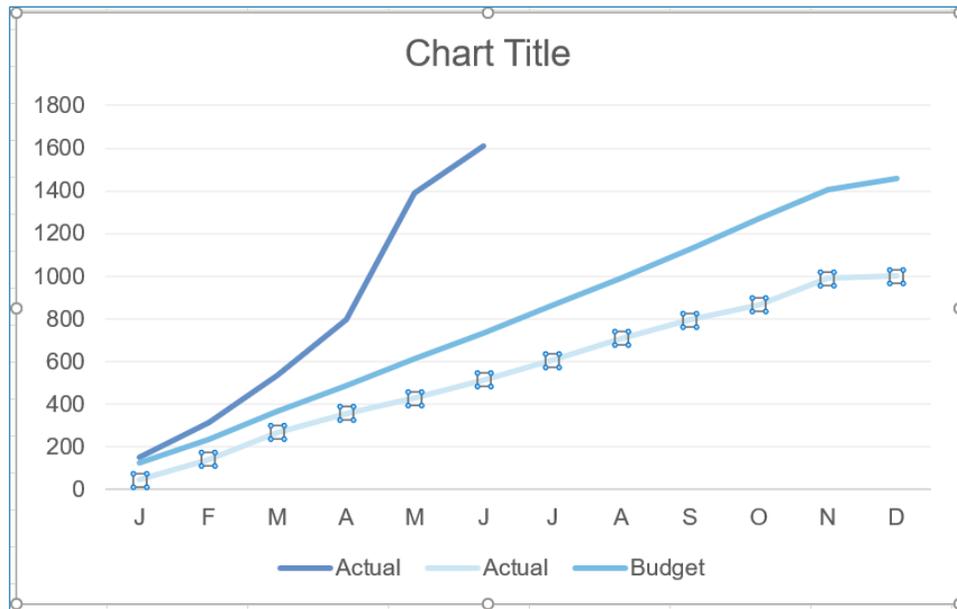


Figure 12.12 - All Data Points for PY Selected

4. Click a **SECOND** time on the **LAST** data point for the **PY** line.

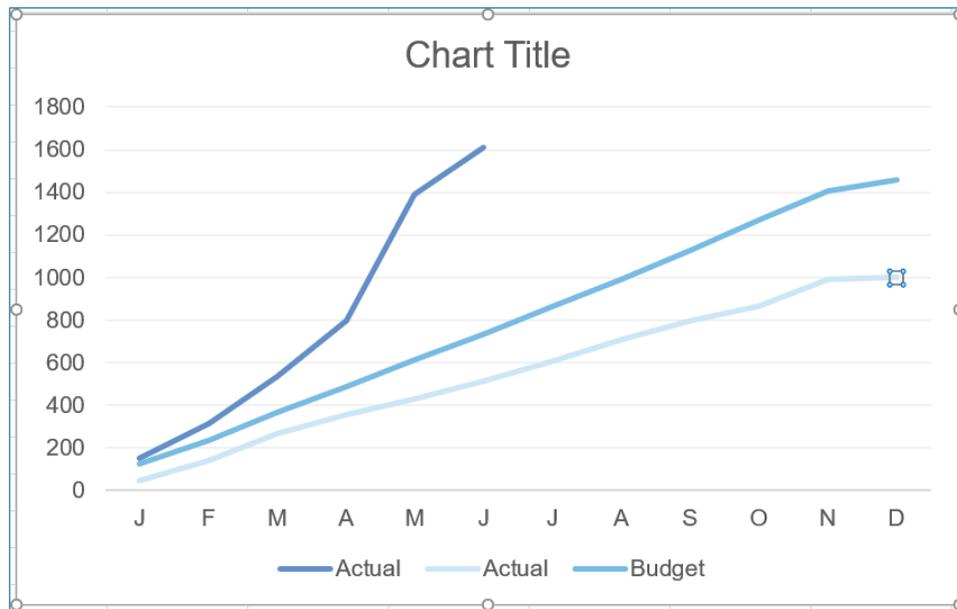


Figure 12.13 - Single Data Point for PY Selected

5. Right-click on the last selected data point and add data labels.
6. Move the right side of the plot area to the left to provide a blank area for the labels to live.

The displayed data label reads “**Actual**”, but we need the chart to display the label “**PY**”.

7. Right-click on the chart and click **Select Data**.

- In the **Select Data Source** dialog box, select the **SECOND** “Actual” and click the **Edit** button.

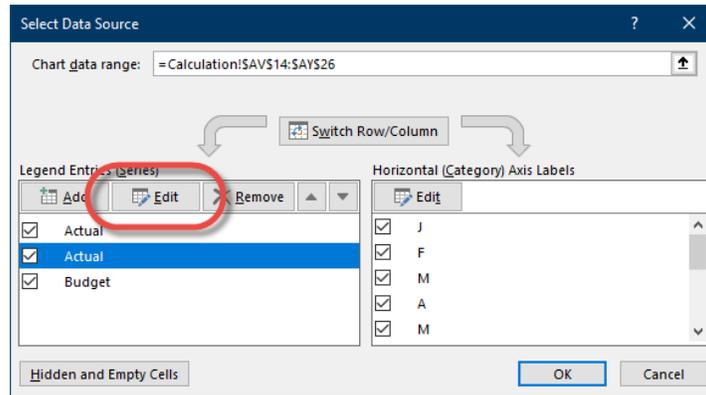


Figure 12.14 - Select Data Source Dialog Box

- In the **Edit Series** dialog box, change the Series Name to read **=“PY”** (don’t forget to include the double quotes.)

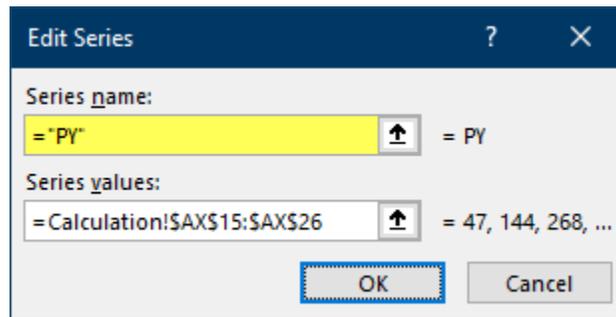


Figure 12.15 - Parameters for Step 9, Module 12.5

- To format the displayed data label, click the data label to display the small blue circles . Click a **SECOND** time to display the larger white circles .

It is VERY IMPORTANT that you select the label TWICE. Otherwise, you will produce very undesirable label results in the next steps.

- Format the data label to display the **Series Name** and hide the **Value**.
- Using the above techniques, add a data label to the last data point for the **medium-blue line** (Budget). Show the **Series Name** but not the **Value**. Don’t forget to perform the “two-times click” trick when selecting the data label prior to customization.
- Add a data label to the last data point for the **dark-blue line** (Actual).
- Lighten the grid lines to a **very light gray**.
- Remove the **legend**.

16. Resize the **plot area** to use the space vacated by the legend in the previous step.

17. Using the scroll form control, scroll to **“Productivity Apps”** (*far left scroll position*).

Notice the problem? The dark-blue line for Actual is behind the PY and Budget lines. This line always needs to be visible. We need to rearrange the series order in the graph.

18. Right-click on the chart and click **Select Data**.

19. Move the **“Actual”** series to the bottom of the series list.

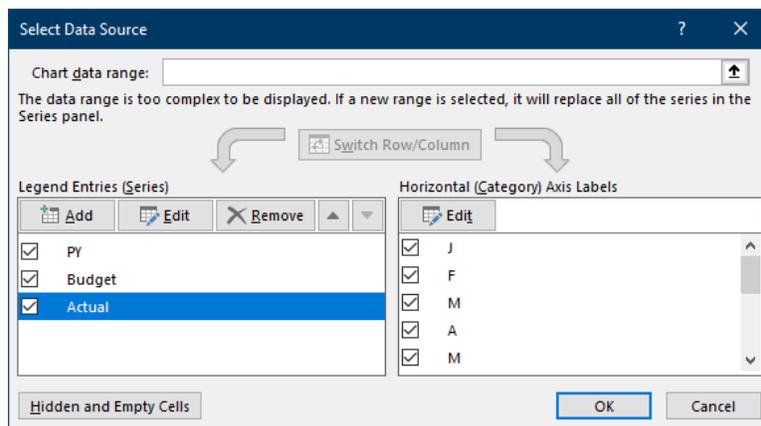


Figure 12.16 – “Actual” Moved to the Bottom of the Series List

It is now time to add the data label to the last data point of the Actual line series. The problem is that if we add the data point to the end of the currently displayed line, when the line extends to include new months, the label will not move with the line.

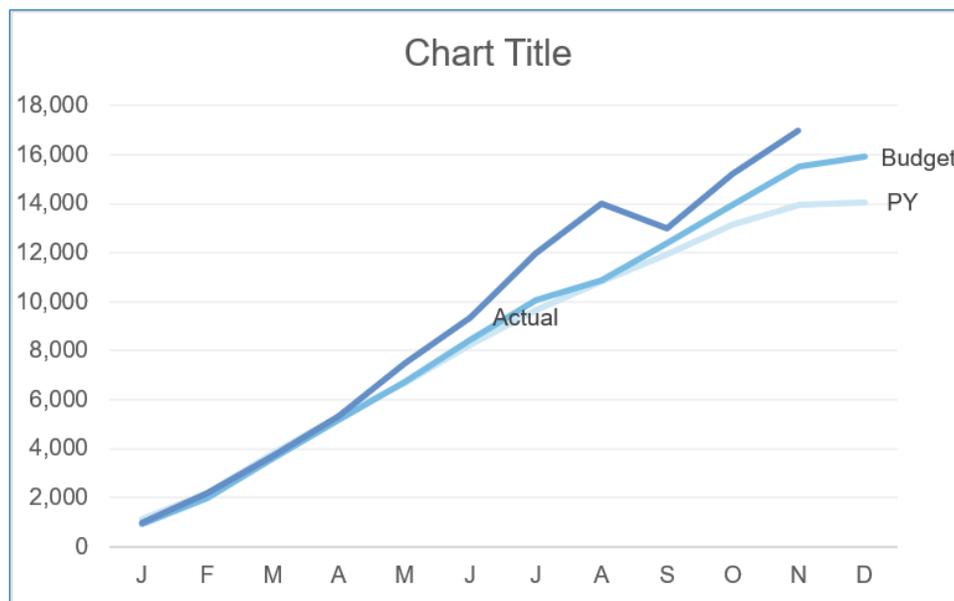


Figure 12.17 - Problem with Non-moving Label

The solution is to create a new data set that displays all the labels for all the data points, but only have the current month’s data point possess a value. All other data points in the series will contain NA() functions which will be ignored by the chart.

20. Select cell **AZ14** and enter the following label:

“Actual Label Position” (no quotes)

21. Select cell **AZ15** and enter the following formula:

=IF(AP15=cur_month,AW15,NA())

22. Add the new data series to the chart by right-clicking on the chart and click **Select Data**.

23. Add the new data series with the following parameters:

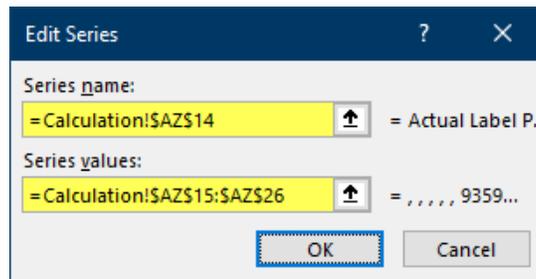


Figure 12.18 - Parameters for Step 21, Module 12.5

24. Select the **Actual Label Position** series. This should produce a selection marker on the end of the dark-blue line.

This time, we need to add the data labels for the **ENTIRE** series. **NO NOT** right click on the displayed data point; this will add the label to only that data point.

25. With the dark-blue line’s last data point selected, click the **Add Chart Element**  button in the upper-right corner of the chart and select **Data Labels**.

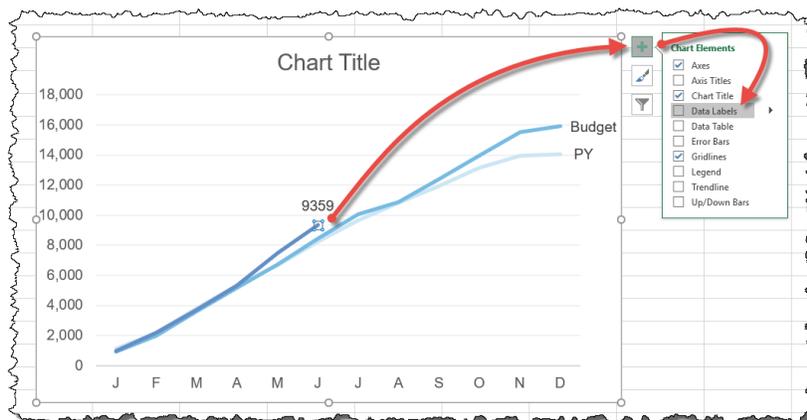


Figure 12.19 – Single data point data label

- 26. Click **ONCE** on the displayed value for Actual and change the **Label Options** to display the **Series Name** and not the **Value**.
- 27. Update the series name in cell **AZ14** to **“Actual”** and add a note in cell **AZ13** to read **“Label Position”**.

We will now create a dynamic title that will be displayed at the top of the chart object in the dashboard.

- 28. Select cell **AV11** and enter the following formula:

=“Profit for ”&AP11

This will be used in the next module when all the pieces are assembled on the **Dashboard** sheet.

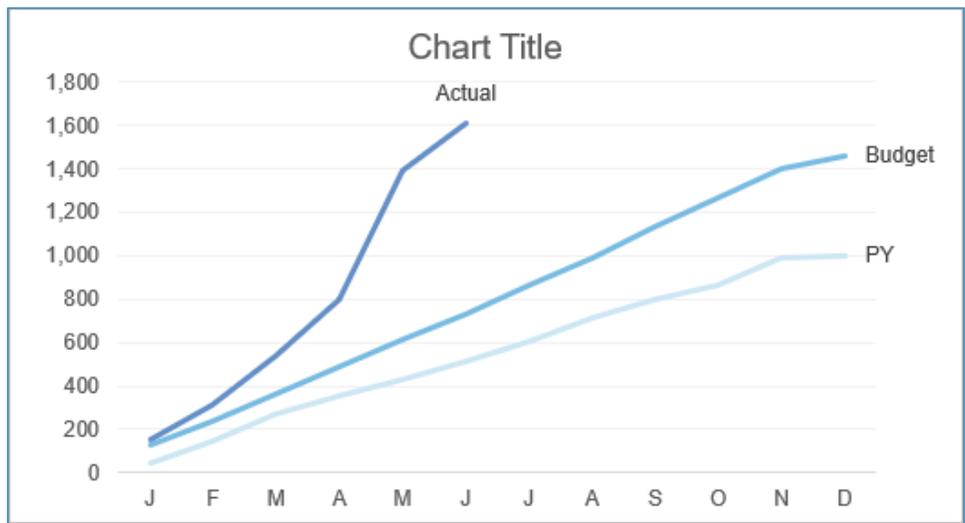


Figure 12.20 - Final Graph Results for Module 12.5

MODULE 12.6 – FINALIZE PROFIT COMPARISON SECTION ON DASHBOARD

Data File:

12 - Scrolling Line Chart with Check Boxes\12.6 - Finalize Profit Comparison Section on Dashboard.xlsx

The final step in creating this piece of the Dashboard is to move all the developed elements to the Dashboard sheet and position them in a visually appealing and easy to use layout. Feel free to experiment with group to and alignment controls to expedite positioning and ensure consistency in your layout.

1. Copy and paste the following items to the **Dashboard** sheet:
 - **Scroll bar** for app selection
 - **Check boxes** for PY and Budget display toggles
 - **Line graph** showing Actual to Budget to PY

NOTE: These items can be deleted from the **Calculation** sheet once we are certain we have everything in place. Leaving a copy in the original location provides us with a “safety net” in case an object becomes corrupt or misconfigured.

One of the downfalls of working from the **Calculation** sheet and then moving the objects to the **Dashboard** sheet when finished is that all the reference pointers in the form control properties will be pointing to the cells on the **Dashboard** sheet.

2. Reset all the cell pointer properties back to the original cell locations on the **Calculation** sheet.
3. Select cells **S25:AC38** and change the cell fill colors to **white**.
4. Remove the **outline** and **shape fill** of the line graph.
5. Remove the **chart title** (*if present*).
6. Add a textbox and place it directly over cells **S24:AC24**.
7. Link the textbox to cell **AV11** on the **Calculation** sheet.
8. Adjust the size and positioning to reflect the finished chart as best as possible.

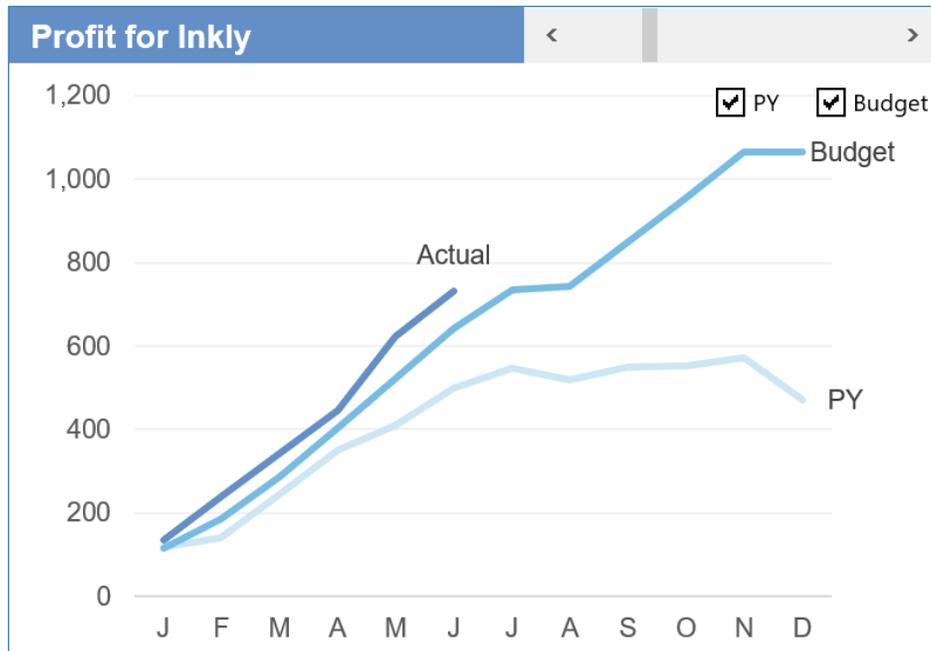


Figure 12.21 - Finished Graph for Module 12

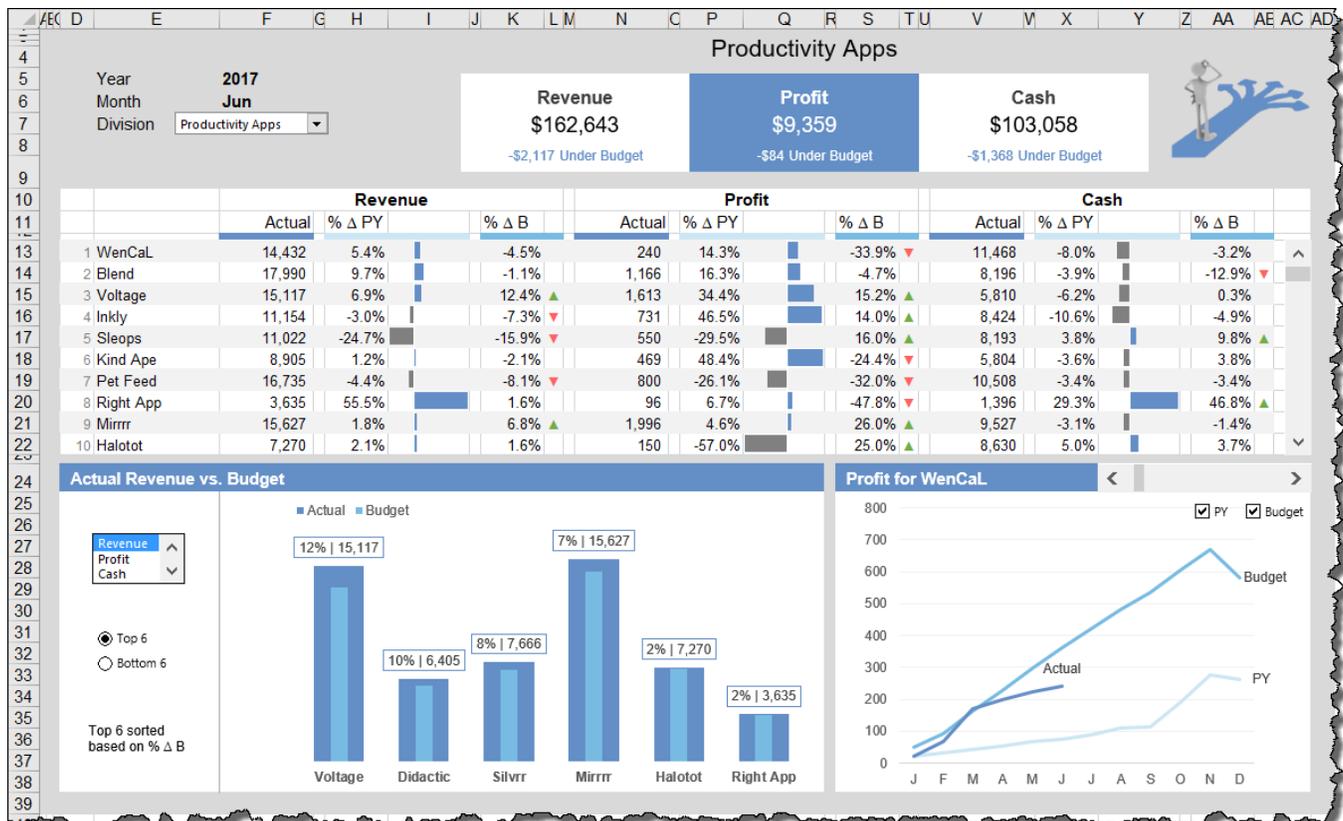


Figure 12.22 - Output for Module 12.6

Module 13

Final Touches (KPI Dashboard)

MODULE 13.1 – RESTRICT DASHBOARD AREA AND FORMAT FOR PRINTING

Data File:

13 - KPI Dashboard - Final Touches\13.1 - Restrict Dashboard Area and Format for Printing.xlsx

Now that the dashboard is complete, it is time to take care of some “housekeeping” issues. We want to ensure that the user cannot navigate to places outside of the dashboard display area. We also want to ensure that the dashboard prints properly and that certain objects on the dashboard do not appear on the printed version of the report.

Perform the following housekeeping operations:

- Highlight all the unused columns and hide them
- Highlight all the unused rows and hide them
- Hide the gridlines
(**View ribbon → Gridlines**)
- Hide the row and column headings
(**View ribbon → Headings**)
- Hide the formula bar
(**View ribbon → Formula Bar**)
- Collapse the ribbon
(**double-click on any tab name**)
- Hide the vertical and horizontal scroll bars
(**File → Options → Advanced → Display options for this workbook**)
- Select all visible data (CTRL-A) and set the print area
(**Page Layout ribbon → Print Area → Set Print Area**)
- Set the page orientation to Landscape
(**Page Layout ribbon → Orientation → Landscape**)
- Set the page to print on a single page
(**Page Layout ribbon → Width → 1 Page**)
(**Page Layout ribbon → Height → 1 Page**)
- Center the dashboard horizontally and vertically on the page
Page Layout ribbon → Page Setup (extended options button) → Margins (tab) → Center on page

- Set the following form controls to not print
(**Properties of control → Print Object (uncheck)**)

 Check boxes on Profit chart (bottom right)
 Scroll bars on Profit chart and scrolling table
- Set the margins to narrow
(**Page Layout ribbon → Margins → Narrow**)
- Consider setting paper size to Legal
(**8½ by 11**)

MODULE 13.2 – DASHBOARD PROTECTION, HYPERLINKS AND FINAL TOUCHES

Data File:

13 - KPI Dashboard - Final Touches\13.2 - Dashboard Protection, Hyperlinks and Final Touches.xlsx

The report is now ready to have version control documentation added and navigational links for easier movement through the file.

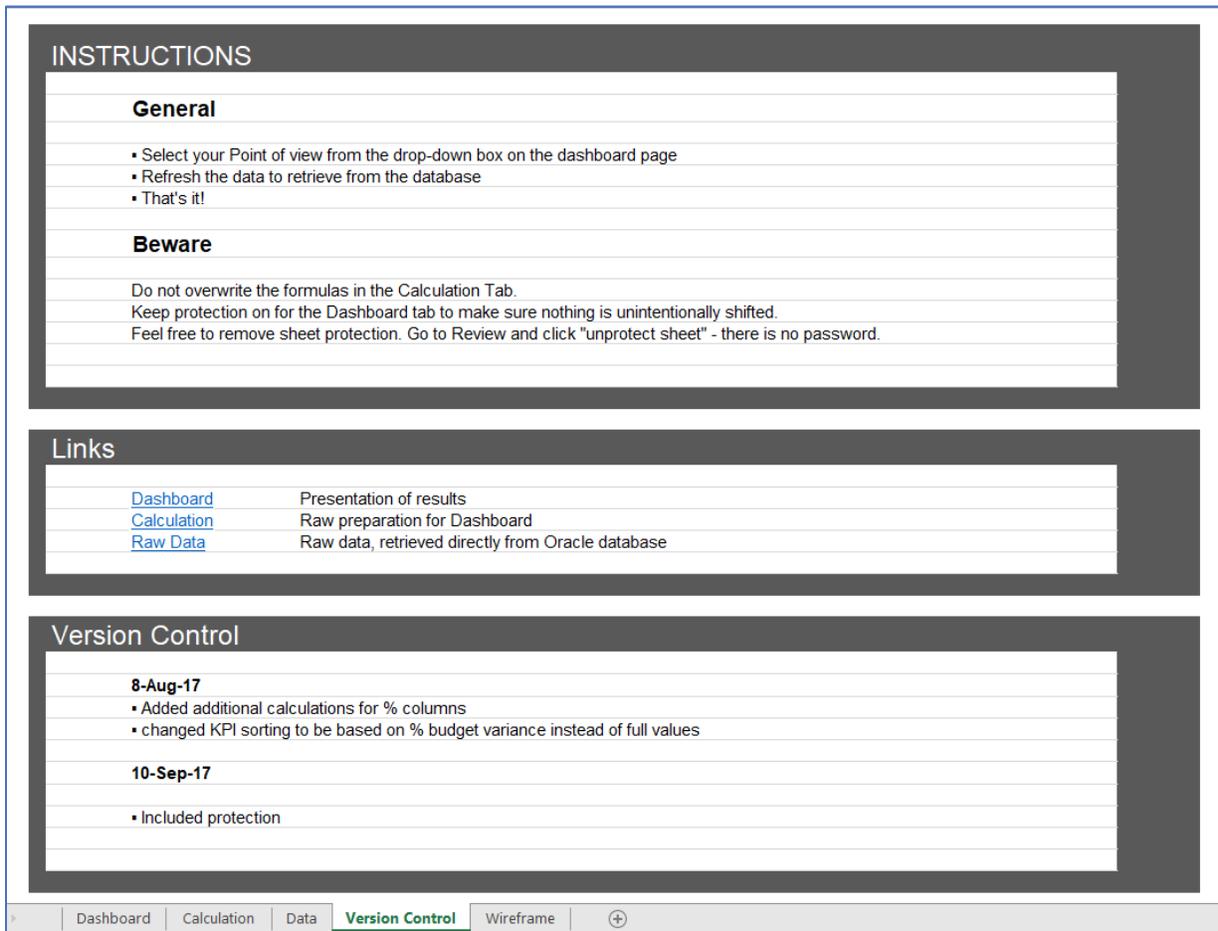


Figure 14.1 - Version Control sheet

We also want to ensure that the user cannot alter any parts of the dashboard except for the input areas and form controls.

In the supplied file, the control sheet has been already created. If you are planning on creating many dashboards, consider creating a control sheet template that you can easily copy and paste into your dashboard file. This will save time and provide consistency in your documentation from report to report.

To add hyperlinks to enable sheet navigation, perform the following steps:

1. On the **Version Control** sheet, right-click on the cell with the word "Dashboard" and select **Link**.

2. Select **Place in This Document** on the left side of the **Insert Hyperlink** dialog box. From there, select **“Dashboard”** from the list of embedded file links.

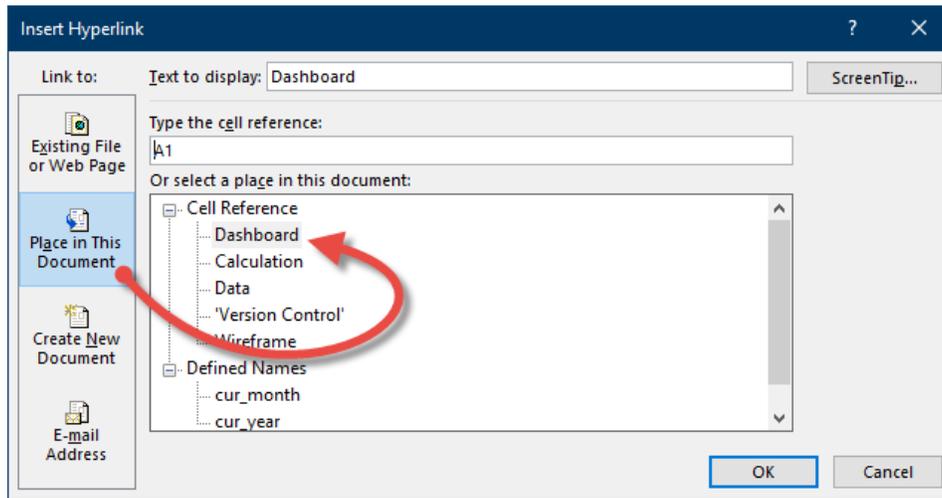


Figure 13.2 - Insert Hyperlink Dialog Box

3. Repeat **Step 2** for the **“Calculation”** and **“Raw Data”** links.
4. On the **Dashboard** sheet, insert a **rectangle shape** and place it in the **upper-left corner** of the dashboard. Size and color the shape in a way you are satisfied.
5. Add text to the shape that says **“Instructions”**.
6. Add a hyperlink to the shape so when clicked will send the user to the **Version Control** sheet.
7. Set the properties of the shape to **not print**.

Because we do not want the user to alter the file in a way that introduces corruption in the logic or presentation of the report, we must take great care to protect our dashboard.

To protect a sheet and prevent any unauthorized modifications, perform the following steps:

8. Select the sheet you want to protect.
9. On the **Review** tab, select **Protect Sheet**.

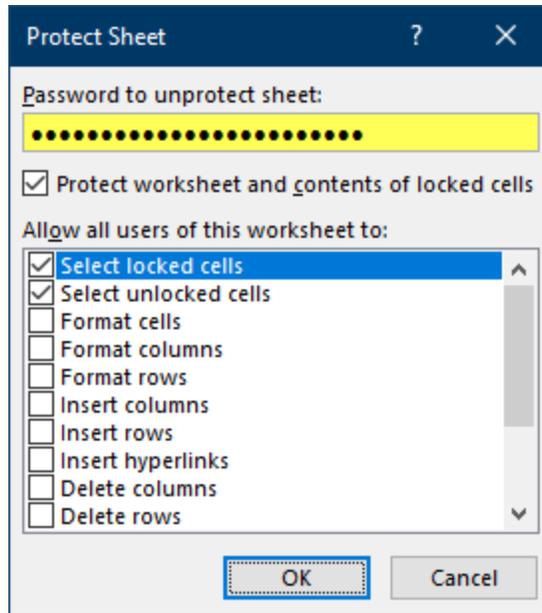


Figure 13.3 - Protect Sheet Dialog Box

10. In the **Protect Sheet** dialog box, enter a password. Don't forget to record this password in a place that authorized personnel can access if the need arises.

Observe that all the form controls are still functioning even when the sheet is protected. If you attempt to type into a cell, the system will deny access to the cell and any modifications will be rejected.

The problem is that the user needs to type in a year and a month in the upper-left corner of the dashboard. To allow this exemption in sheet protection, perform the following steps.

11. On the **Review** tab, unprotect the sheet by clicking the **Unprotect Sheet** button.
12. Highlight cells **G5:G6** (the year and month cells).
13. Right-click the selected cells and click **Format Cells**.

14. On the Protection tab, uncheck **Locked**.

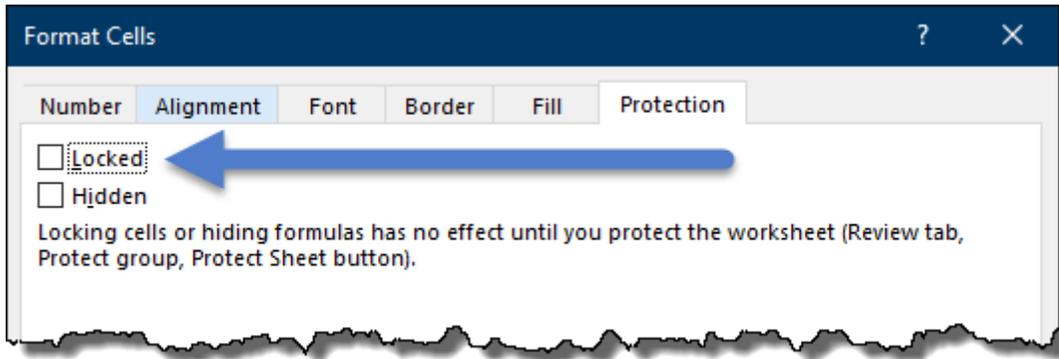


Figure 13.4 - Format Cell Dialog Box

15. Re-protect the sheet with a password.



Figure 12.5 - Finished Graph for Module 12



Figure 13.6 - Final Output of KPI Dashboard

Congratulations!!! You have just created your KPI Dashboard.

Take a moment to reflect on how you felt when this journey began. When you saw an example of the finished dashboard at the beginning of our journey, you likely experienced waves of anxiety and self-doubt. “How will I ever create this?” is a thought that may have crossed your mind.

By taking what appears to be an insurmountable objective and breaking it into smaller and smaller tasks and giving yourself time to think about one task at a time, before you know it, you have reached your goal and probably had a lot of fun and enlightenment along the way.

Now it’s time to take another journey. We will encounter new challenges and obstacles, but with the experience gained from the previous modules our next objective will a bit easier and provide just as much opportunity for growth in our skills and self-confidence.

Modules 14-17

Regional Dashboard (Pivot Based)

Regional Dashboard
Setup

XelPlus.com

MODULE 14.1 – PIVOT SLICERS AS BUTTONS (ALSO WITHOUT PIVOT TABLES)

Data File:

14 – Using Slicer as Buttons\14.1 – Pivot Button as Slicer.xlsx

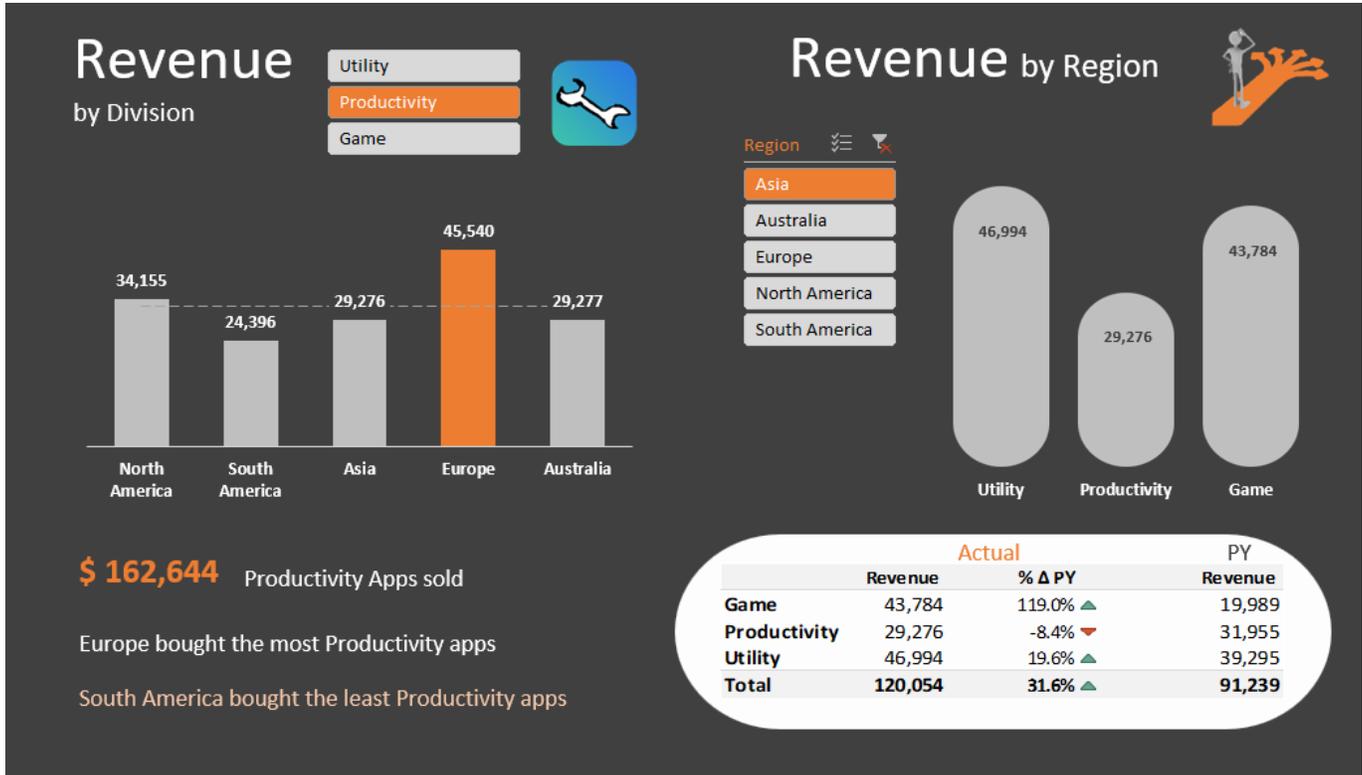


Figure 14.1 - Slicer Dashboard (Solution)

Modules 14 through 17 will be devoted to the development of the Slicer dashboard displayed above. We will utilize many of the skills visited in the previous modules as well as learning new skills to solve previously unencountered problems.

In the next modules, we will see how to add increased functionality in our dashboards by using Slicers.

Slicers were introduced in Excel 2010 to more easily filter Pivot Tables and Pivot Charts. In Excel 2013, they were also extended to Data Tables. The slicers in the above image, are the filter buttons for division and region. Although slicers make it easier to perform filtering, the limitations of the Pivot Tables and Pivot Charts themselves made their widespread use somewhat restricted. Some of the limitations of Pivot Tables and Pivot Charts are:

- Restricted functionality
- Formula complexity is limited in Pivot Tables
- Pivot Charts can't use certain series combinations

With a few tricks though, you can get slicers to control your “standard tables” and “standard charts”. You’ll find out how in the upcoming modules.

To begin building our new dashboard, let’s examine the contents of the starting file.

Dashboard: This sheet serves as an empty canvas.

Data: This sheet contains a Data Table named **RawData**. Although you will most certainly work with larger data sets in the real world, this small data set will serve the needed requirements for this course. This sheet will likely be hidden from the user.

Calculation: Most dashboard calculations will be stored on this sheet to insulate the logic of the dashboard from the presentation aspect of the dashboard. This sheet will most certainly be hidden from the user.

Final Image: This holds a static image of our dashboard objective. If you are unclear as to how the item(s) being constructed should appear, you can consult this sheet for visual advice.

Let’s examine the major driving forces of the first chart (upper-left) on the Dashboard sheet and the needed logic of the Calculation sheet to drive such a chart:

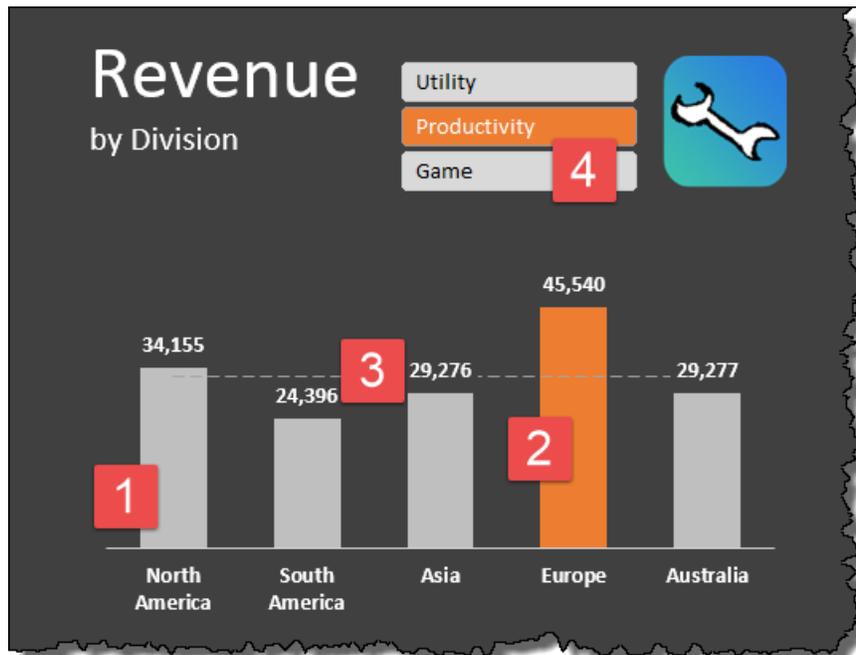


Figure 14.2 - Column Chart Required Elements

1. We need to construct a data set that shows revenue for each region. Each bar represents a different region and the height of the bar is driven by the region’s revenue value.
2. We want to show the region with the greatest revenue as a different color (orange) where the other regions will be colored gray. Each data bar will have its numeric value displayed above their respective bar.
3. A dotted line representing the average revenue will be drawn across the entire data set.
4. A slicer will be used to control the Division selection. The user will select one division at a time, so the chart will display revenue for a single division.
5. If the user selects more than one division from the slicer, a small warning will appear advising the user to select only a single button from the slicer.



Figure 14.3 - Multiple Button Selection on Slicer Message

Building a Slicer

To use a slicer, we first need to build a pivot table. Our first slicer will only display a unique list of Divisions. If we were to use the minimum amount of data from the data table, we could select only the Division column and the pivot table and slicer would function satisfactorily.

	A	B	C	D
1	Year	Division	Region	Revenue
2	Current	Utility	North America	44196
3	Current	Utility	South America	20898
4	Current	Utility	Asia	45540
5	Current	Utility	Europe	43695
6	Current	Utility	Australia	34196
7	Current	Productivity	North America	34155
8	Current	Productivity	South America	24396
9	Current	Productivity	Asia	29276
10	Current	Productivity	Europe	45540

Figure 14.4 - Division Column Selection

In this case, we will select the entire data table when producing our Division pivot table and slicer. The reason for this excessive selection is to limit the spreadsheet to the usage of a single pivot cache.

When you select data of different size data ranges in the production of pivot tables, Excel will create a separate pivot cache for each range. This will not only increase the size of the spreadsheet, but each pivot cache will need to be refreshed when the source data is updated. By limiting the spreadsheet to a single pivot cache, only a single update is required to update all resultant charts and tables and file’s size is kept to a minimum.

To ensure only a single cache is created, we will create our first pivot table using all the data from the data table. For the second and subsequent pivot tables, we will copy and paste the first pivot table and adjust the duplicate pivot table’s settings.

1. On the **Data** sheet, click anywhere in the data range (A1:D31).
2. Select the **Table Tools - Design** tab. In the **Tools** section select **Summarize with Pivot Table**.

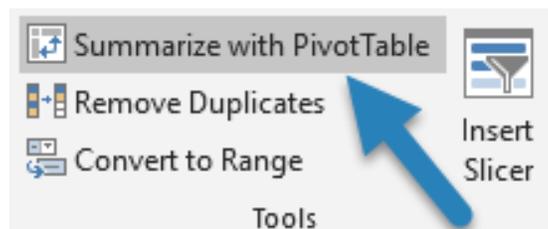


Figure 14.5 - Summarize with Pivot Table Button

3. In the **Create Pivot Table** dialog box, select **Existing Worksheet** and set the **Location** to **Calculation!\$A\$5** and click **OK**.

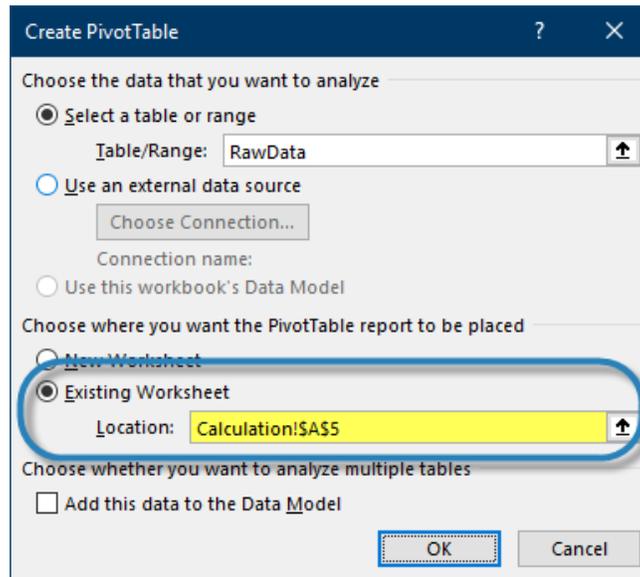


Figure 14.6 - Create Pivot Table Dialog Box

4. Add the **Division** field to the **Rows** section. This will produce a very small pivot table with only the Divisions listed as a unique list.

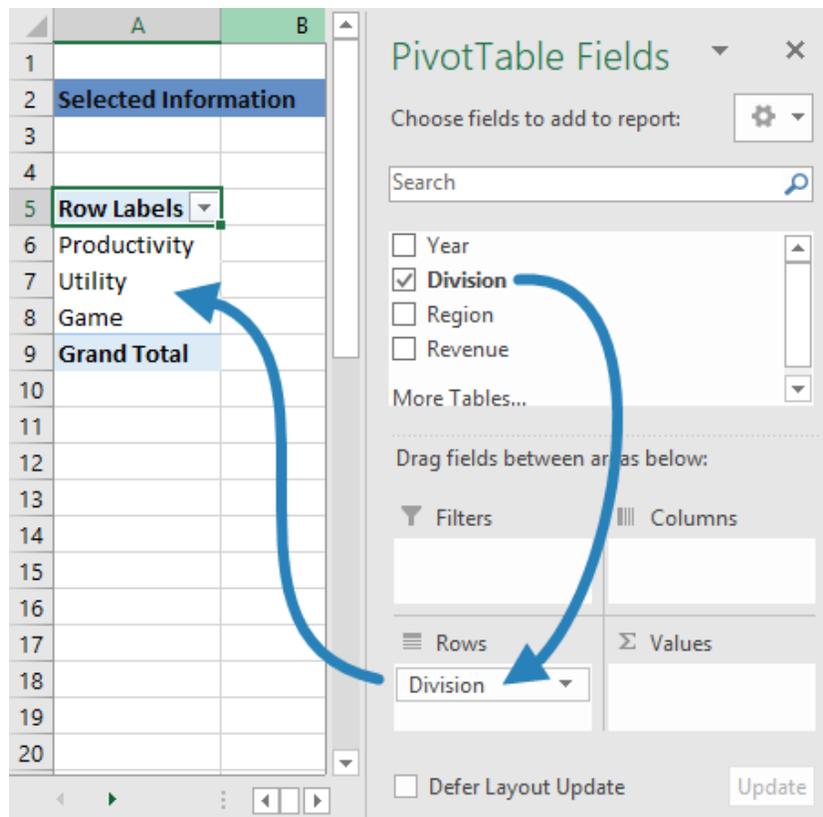


Figure 14.7 - Pivot Table Setup and Results

5. We will now perform some basic visual customization on the pivot table:

- Set the Report Layout to Tabular
- Set the Grand Totals to Off for Rows and Columns

6. Insert a slicer for Division:

- a. With the pivot table selected, click **Pivot Table Tools** → **Analyze** → **Insert Slicer**.
- b. Select **Division** from the Insert Slicers dialog box and click **OK**.

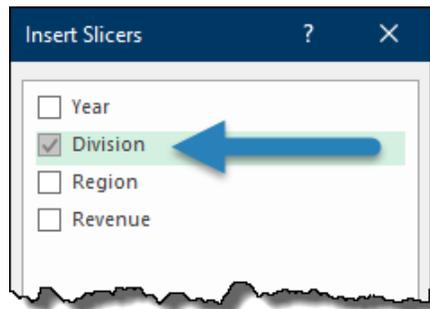


Figure 14.8 - Insert Slicers Dialog Box Selection

Test the functionality of the slicer. Select each Division to see if the pivot table responds by only displaying the selected Division in the slicer.

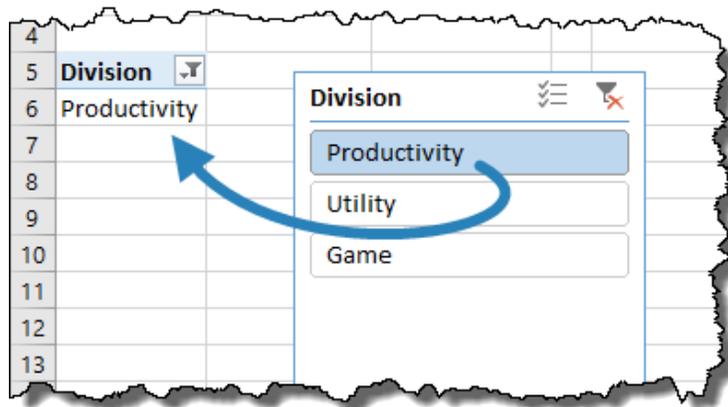


Figure 14.9 - Slicer Test

Observe that the pivot table's width adjusts to the length of the displayed data. To prevent the columns in the pivot table from shifting during use, and to ensure no data extends beyond the confines of the column, perform the following steps:

7. Select the button in the slicer with the most number of characters (i.e. "Productivity").
8. Click anywhere in the pivot table and select **Pivot Table Tools** → **Analyze** → **Pivot Table (group)** → **Options**.

9. In the **Pivot Table Options** dialog box, uncheck the box for “**Autofit column widths on update**”.

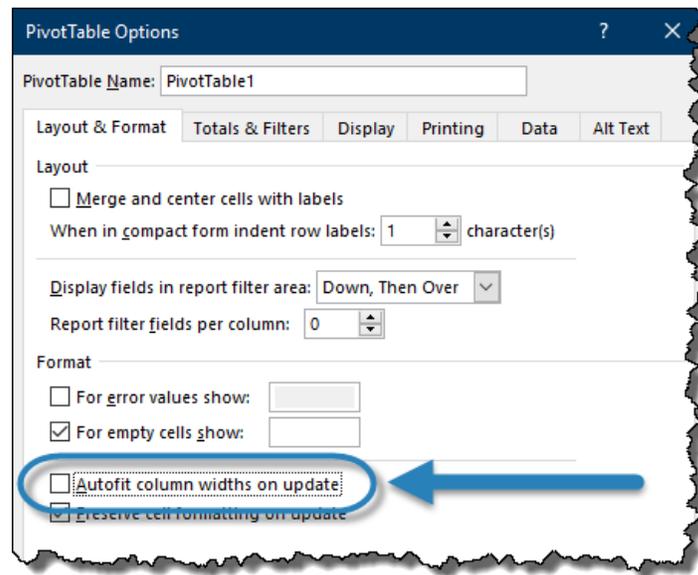


Figure 14.10 - Preventing Column Resizing During Pivot Table Updates

We will now use the result of the pivot table (as controlled by the pivot table slicer) to extract data from the data table to derive a custom chart. Our chart requires the regions to be listed across the bottom (x-axis) and the revenue to govern the height of the columns.

The custom data set will reside on the Calculations sheet. The first thing we need to do is create a list of the regions.

10. On the **Data** sheet, select all the cells containing region names (**C2:C31**) and click **Copy** (or CTRL-C).
11. Select cell **G1** and **Paste** (or CTRL-V) the regions into a new list.
12. With the duplicate set of regions selected, remove the duplicates by selecting **Data** → **Data Tools (group)** → **Remove Duplicates**. In the **Remove Duplicates** dialog box, select **OK**. 25 duplicates should have been removed leaving us with 5 unique region names.
13. Select the 5 unique region names (cells **G1:G5**) and click **Cut** (or CTRL-X).
14. Move to the **Calculation** sheet and select cell **F5** and click **Paste** (CTRL-V).
15. Select cell **G4** and enter the title “**Revenue**”.

We want to produce a list of values in cells G5:G9 to govern the height of the columns in the chart. We have several criteria for the calculation:

- Revenue (**G4**)
- Regions (**F5:F9**)
- Selected Division (**A6**)
- Year – Current or PY (**Data sheet, column A**)

Since we only want to pull from the current year, and we may wish to switch between “Current” and “PY”, we will set this up as a dynamic selection instead of hard coding “Current” directly into the formula.

16. Select cell **B5** and enter the title “**Year**”.

17. Select cell **B6** and enter the label “**Current**”.

We are now ready to write a formula that will extract information from the data table based on the above criteria.

There are many ways to derive this value; in this case, the SUMIFS function will work effectively.

Note: Since we will be writing a formula that includes elements from two different sheets (Data and Calculation), this would be a great time to implement a dual view of the Excel spreadsheet. (see Section 2 – Lecture 10 “4 Time Saving Tips for Excel” – “Open 2 Sessions” for a refresher on this setup.)

18. Select cell **G5** and enter the following formula:

```
=SUMIFS(RawData[Revenue],RawData[Year],$B$6,RawData[Division],$A$6,  
RawData[Region],$F5)
```

19. Fill the formula down through cells **G5:G9**.

Test the formula answers by cross-checking the values in the data table with the selected division from the slicer.

Note: We will not be needing the second session of Excel for some time; go ahead and close the second Excel session.

20. Format the values in cells **G5:G9** with a **Number** style, use a **1000 Separator** and **zero** Decimal places.

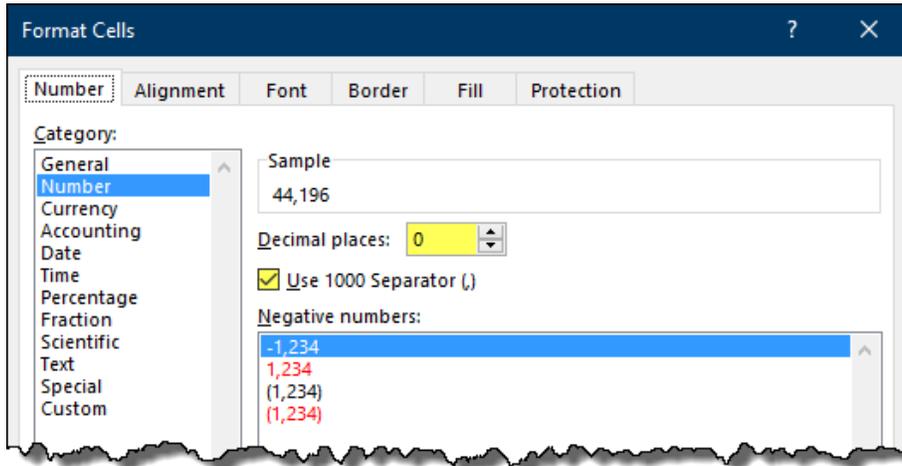


Figure 14.11 - Format Cells Dialog Box

21. Add formatting to the **Revenue** title by selecting cell **G4** and making the cell **Bold**. Add a thin line to the lower border of the cell.

	F	G
4		Revenue
5	North America	44,196
6	South America	20,898
7	Asia	46,994
8	Europe	43,695
9	Australia	34,196
10		

Figure 14.12 - Data Set Result as of the End of Module 14.1

MODULE 14.2 – CUSTOM COLUMN CHART CONTROLLED BY SLICER BUTTONS

Data File:

14 – Using Slicer as Buttons\14.2 – Custom Column Chart Controlled by Slicer Buttons.xlsx

In this module, we will create a chart that is controlled by the data set created in the previous module and indirectly controlled by the slicer.

To recap our chart requirements:

- We need to construct a data set that shows revenue for each region. Each bar represents a different region and the height of the bar is driven by the region’s revenue value.
- We want to show the region with the greatest revenue as a different color (orange) where the other regions will be colored gray. Each data bar will have its numeric value displayed above their respective bar.
- A dotted line representing the average revenue will be drawn across the entire data set.

Because we want to show three different series in the chart, we need three different series in our table. We already have the data for the Revenue; we will now create the data for the Max and the Average.

1. On the Calculation sheet, select cell **H4** and enter the title “**Max**”.
2. Select cell **I4** and enter the title “**Average**”.

We need a series that displays a value if the corresponding Region is the largest value (or MAX) in the Revenue series.

3. Select cell **H5** and enter the following formula:

```
=IF(G5=MAX($G$5:$G$9),G5,"")
```

4. Fill the formula through cells **H5:H9**.

Test the formula answers by changing the slicer selection among the three Divisions. You should see the various values change according to their Region and Division.

We need a series that displays the same average value for the entire Revenue series, but for each Region.

5. Select cell **I5** and enter the following formula:

```
=AVERAGE($G$5:$G$9)
```

6. Fill the formula through cells **I5:I9**.

Test the formula answers by changing the slicer selection among the three Divisions. You should see the average of the selected Division displayed across all the Regions.

- To give the same appearance to the newly created columns in our data set, use the **Format Painter**  to copy the formatting from cells **G4:G9** and apply the formatting to cells **H4:I9**.

Because two of the three series are column chart based (Revenue and Max), we will start by making a standard 2-D Clustered Column chart for all three series. Later, we will convert the chart to a Combo Chart and set the third series (Average) to a line chart style.

- Select cells **F4:I9** and select **Insert** → **Charts (group)** → **2-D Column** → **Clustered Column** (top left selection).

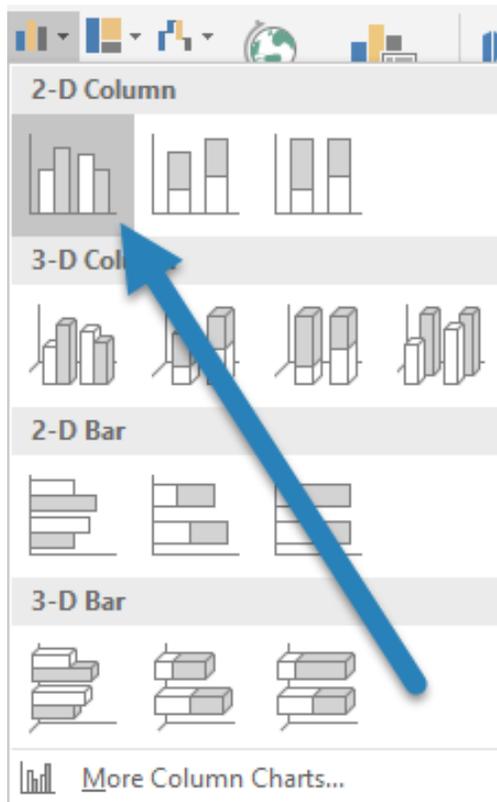


Figure 14.13 - Selection for 2-D Clustered Column Chart

- Select one of the **Average** bars and **right-click** → **Change Series Chart Type** → change the **Average** series to a **Line Chart**.

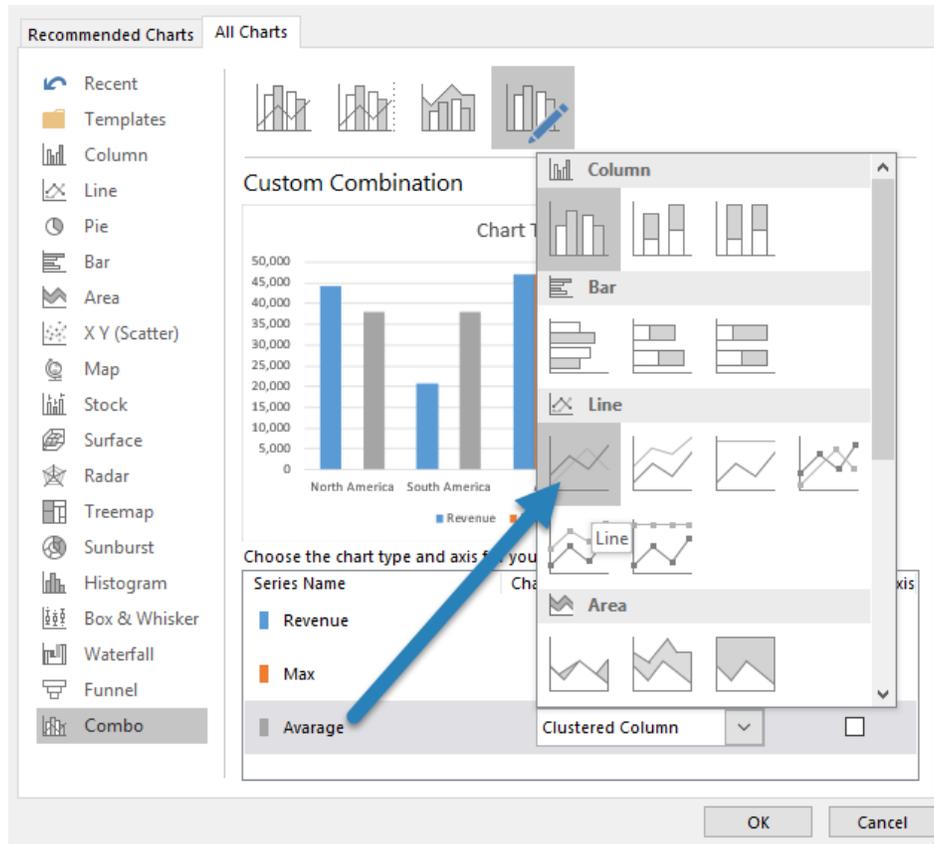


Figure 14.14 - Line Chart Option in Combination Chart

10. Select any blue or orange bar and **right-click** → **Format Data Series** → **Series Overlap** → type **100%**.

11. While we are in the Format Data Series panel, set the **Gap Width** to **100%**.

12. Remove the following items from the chart:

- Legend
- Vertical axis (y-axis)
- Gridlines

13. Add data labels to the blue bars. Right-Click on any blue bar and click **Add Data Labels**.

We will set the color scheme for the 1st, 2nd, and 3rd series. This will be accomplished using a Custom Colors pallet in the Colors tool.

14. Select a **corner of the chart** → select **Page Layout** → **Themes (group)** → **Colors** → **Customize Colors**.

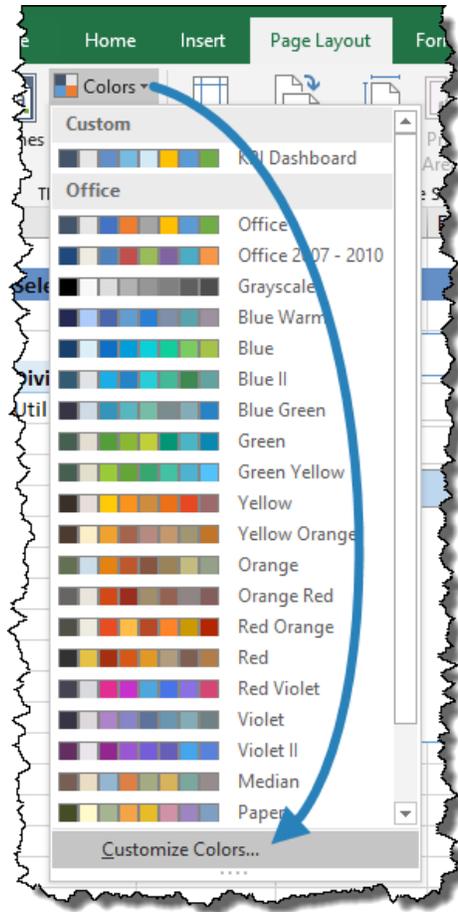


Figure 14.15 - Custom Color Option for Themes

15. The colors for Accent 2 and Accent 3 will work well. Set **Accent 1's** color to **25% Gray**.

16. Name the new color scheme "**Slicer Dashboard**" and click **Save**.

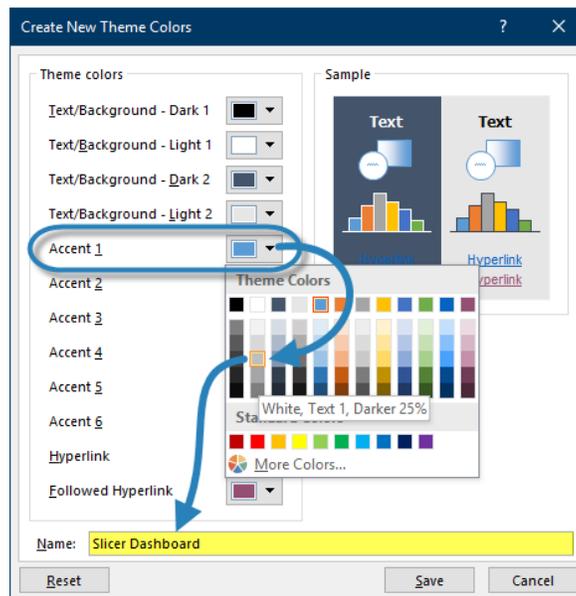


Figure 14.16 - Setting Accent 1's Color and Saving

17. Format the Average line's thickness by **selecting the line** → **Format** → **Shape Styles (group)** → **Shape Outline** → **Weight** → **¾ pt.** Modify the Average line's line style by selecting **Format** → **Shape Styles (group)** → **Shape Outline** → **Dashes** → **Long Dash** (#6 in the list).

It is now time to move the chart and the slicer from the Calculation sheet to the Dashboard sheet.

18. Select the **title area of the slicer** and then hold down **CTRL** and select the **chart**. With the slicer and chart objects selected, click **Cut** (or CTRL-X), select the **Dashboard** sheet, then click **Paste** (or CTRL-V).

19. **Move** and **resize** the slicer and chart objects to appear approximately as the illustration below.

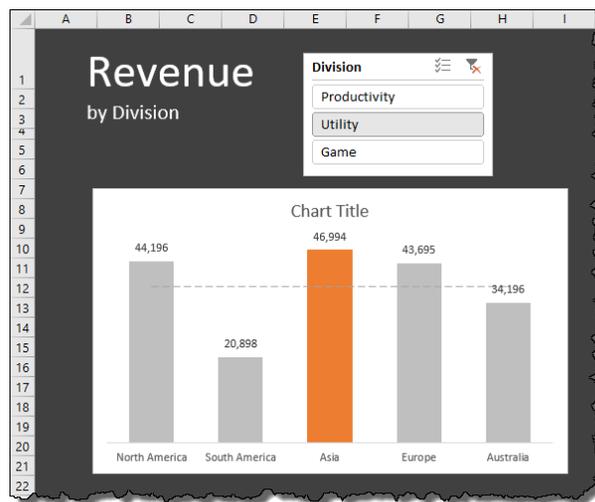


Figure14.17 - Result of Step 19 Content Move to Dashboard

We will now perform some additional formatting to the chart to make it look better with the new darker background of the Dashboard sheet.

20. Select the chart and **remove** the **title**.
21. Select the **horizontal axis** (x-axis) and set the labels to **Bold** and **10.5pt** in size.
22. Use the **Format Painter** to apply the look of the **Region** names (horizontal axis) to the **data labels**.
23. Select a **corner of the chart** and set the **Shape Outline** to **No Outline**.
24. Set the chart's **Shape Fill** to **No Fill**.
25. Change the **font color** of the **Region** names and the data labels to **White**.

A minor visual issue occurs with the Average line when selecting certain Divisions. Ex: Select "Productivity" from the slicer and observe the line interaction with the value displayed above "Asia". In

certain circumstances, the Average line appears to cut through the data label. To prevent this from happening, we will give the data label the same fill color as the chart background.

26. Select any **data label** → **Format** → **Shape Styles (group)** → **Shape Fill** → **25% Black**.

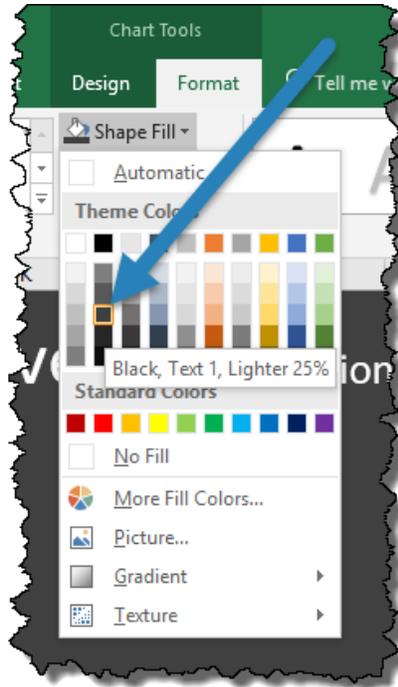


Figure 14.18 - Color Option for Data Label Fill

Test the chart by changing the various Divisions in the Division slicer.

The chart progress should appear as follows:

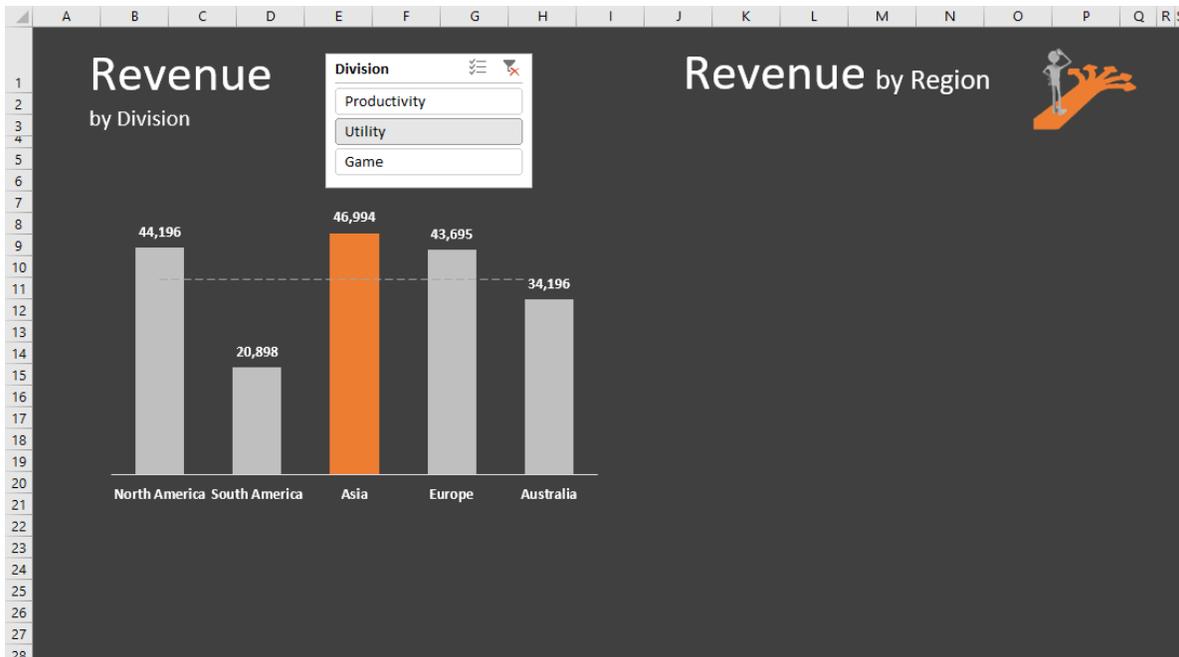


Figure 14.19 - Chart Progress for End of Module 14.2

MODULE 14.3 – SLICER CUSTOM SORTING & DESIGN

Data File:

14 – Using Slicer as Buttons\14.3 – Slicer Custom Sorting and Design.xlsx

In this module, we will change the look of the slicer and its buttons, so it blends in better with the report.

1. Remove the Slicer header by selecting the slicer and then selecting **Slicer Tools – Options** → **Slicer (group)** →  **Slicer Settings**.
2. Uncheck “**Display header**”.

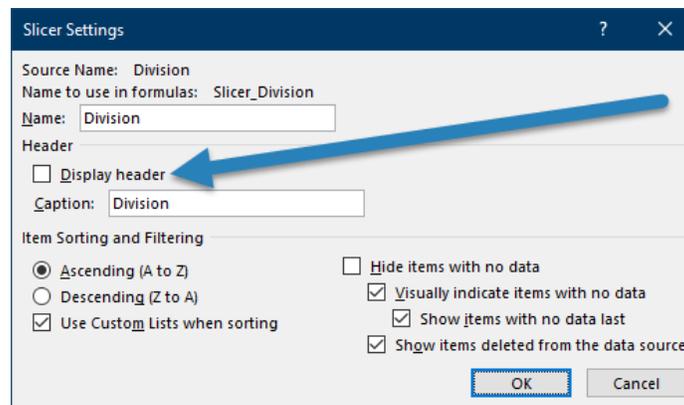


Figure 14.20 - Slicer Settings Dialog Box

Now we will create our own slicer style to use the same colors as the report.

3. With the slicer selected, in the Slicer Styles section of the Slicer Tools – Options ribbon, right-click on the active slicer style and select **Duplicate**.

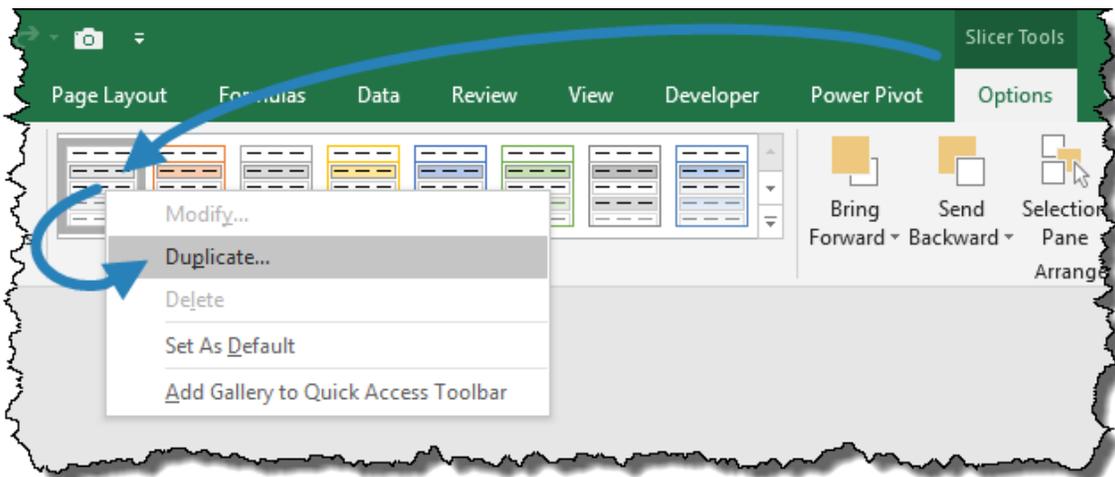


Figure 14.21 - Duplicate Slicer Style

4. Name the slicer **“Slicer Dashboard”**. Select **“Whole Slicer”** and perform the following modifications:
 - Set the Border to None
 - Set the Fill Color to 25% Black (2nd column, 3rd down)

It appears as if the custom slicer did not take effect. By duplicating the slicer and modifying the new version, we have merely added the new slicer style to the Slicer Styles library. We must now apply the new slicer style to our slicer to see the new color scheme in action.

5. With the slicer selected, apply the newly created **“Slicer Dashboard”** slicer color scheme.

The current color scheme of the slicer is using a combination of slightly different gray colors to distinguish between selected and unselected items. This is not only difficult to understand, but it does not fit with our report color scheme. We will now apply the report’s orange color to user selected slicer buttons.

6. Right-click the **“Slicer Dashboard”** slicer style and select **Modify**.
7. In the **Slicer Element** list, click **Selected Item with Data**, click **Format** and perform the following modifications:
 - Format the **Fill Color** to the **Orange, Accent 2** (Top row)
 - Set the **Font Color** to **White**
8. In the **Slicer Element** list, click **Unselected Item with Data**, click **Format** and perform the following modifications:
 - Format the **Fill Color** to the **15% Gray** (1st column, 2nd down)
9. Reduce the slicers overall border size to fit the visible buttons.

The slicer should now have the following appearance:



Figure 14.22 - Results of Steps 6 Through 9

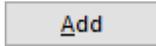
Now it is time to list the Divisions in the slicer in an order that we prefer. Right-clicking on the slicer allows us to sort the items in the slicer in either ascending or descending order. What if the order we want is as follows:

- Utility
- Productivity
- Game

To achieve this user-defined order, we will use the Custom Lists sorting feature in Excel.

10. Select **File** → **Options** → **Advanced**.

11. In the **General** sub-category, click 

12. In the **List Entries** field, type **Productivity, Utility, Game**. Click the  button and then click **OK**.

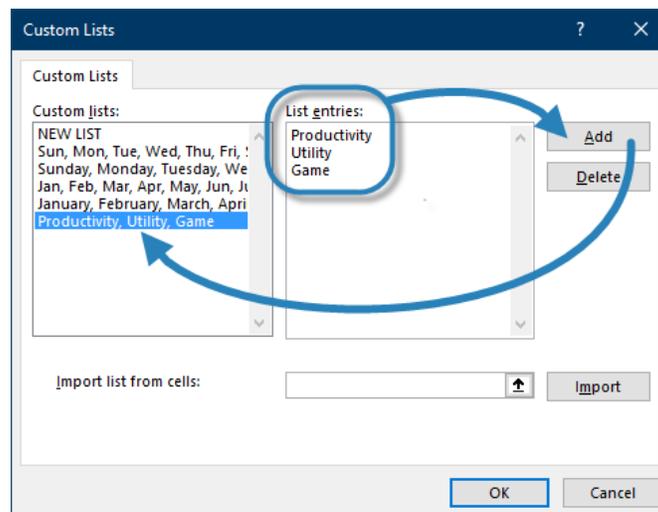


Figure 14.23 - Creating a Custom Sort List

13. Open the **Slicer Settings** dialog box and **clear** the checkbox for “**Use Custom Lists when sorting**” and click **OK**.

14. Open the **Slicer Settings** dialog box and **check** the checkbox for “**Use Custom Lists when sorting**” and click **OK**.

Steps 10 through 14 were a demonstration of how to create a custom sort order. In this report, we would like a descending sort for our slicer.

15. Select the slicer and in the **Slicer Settings**, set the Sort Order to **Descending** without using a custom list.

The updated slicer with its colors and display order should appear as follows:



Figure 14.24 - Result of Descending Sort

MODULE 14.4 – CHECK TO ENSURE SINGLE SLICER BUTTON IS SELECTED

Data File:

14 – Using Slicer as Buttons\14.4 – Check to Ensure Single Slicer Button is Selected.xlsx

The goal of this slicer is to only be able to select one button at a time, thereby limiting analysis to a single Division. We removed the header and the clear button from the slicer, but the user remains able to select multiple slicer buttons using standard Windows CTRL and Shift selection techniques.

If the user selects multiple slicer buttons, the Calculation sheet will show all the selected Divisions, but the data table only monitors the first selected division located in cell A6.

The screenshot shows an Excel dashboard with a slicer on the left and a data table on the right. The slicer has three buttons: 'Game', 'Productivity', and 'Utility'. The 'Game' button is selected. The data table is titled 'Data Table by Region for Chart' and contains the following data:

Region	Revenue	Max
North America	44,675	
South America	42,569	
Asia	43,784	
Europe	46,336	
Australia	49,656	49,656

Figure 14.25 - Multiple Slicer Button Selection Problem

We will add a “check condition” to detect when a user has selected more than one slicer button and display a message to warn against such activity.

We will use a COUNTA function to monitor the cells A6:A8 on the Calculation sheet.

1. On the **Dashboard** sheet, select cell **B6** and enter the following formula:

```
=IF(COUNTA(Calculation!A6:A8)>1,"Please only select 1 Division!","")
```

2. Set the cells font color to **orange** and the font size to **14 points**.

Test the slicer with various single button selections as well as multiple button selects to see if the message is either visible or hidden depending on the appropriate slicer selection.



Figure 14.26 - Multiple Slicer Selection Warning Message

MODULE 14.5 – DYNAMIC COMMENTS WITH SLICER SELECTION

Data File:

14 – Using Slicer as Buttons\14.5 – Dynamic Comments with Slicer Selection.xlsx

In this lecture we will learn how to add dynamic comments to our reports. The message(s) to the user will change based on the results of the data set on the Calculation sheet.

If you examine the static image of our finished report on the Final Image sheet, you can see the dynamic content in the lower left corner of the report.

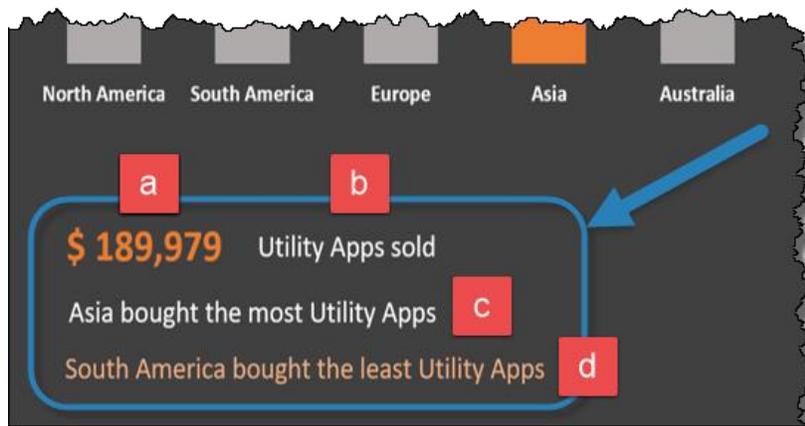


Figure 14.27 - Dynamic Comments (finished)

There are four dynamic messages being displayed:

- a) The total dollar value of the selected Division
- b) The message with the selected Division’s name
- c) The country name with the most sales along with the selected Division name
- d) The country name with the least sales along with the selected Division name

In addition to the messages, we also want to format the text and values with appropriate report colors.

To achieve this, we will use textboxes.

1. On the **Calculation** sheet, insert a textbox at approximately **E12**’s cell position and enter the following formula in the formula bar:

=A\$6

Test the dynamic nature of the new textbox by changing the selection on the Dashboard’s Division slicer. Observe that the textbox displays the selected Division.

A limitation of textboxes is that you cannot place formulas in the textbox itself. The formula must be placed in a cell and then the textbox needs to point to the cell with the formula.

2. Select cell **K5** and enter the title “**Total Revenue for Selection**”
3. Select cell **L5** and enter the following formula:

```
=SUM($G$5:$G$9)
```

The total revenue for all Regions is displayed in cell L5.

4. Modify the formula in cell **L5** to reflect the following:

```
=SUM($G$5:$G$9) & " " & A6 & " Apps sold"
```

Because we combined the value of the SUM function with the text, we lost the number formatting. We can recover the lost number formatting by nesting the above formula in a TEXT function. This will allow us to reformat the text while also concatenating with a text message.

5. Modify the formula in cell **L5** to reflect the following:

```
=TEXT(SUM($G$5:$G$9),"$#,##0") & " " & A6 & " Apps sold"
```

6. Update the original textbox’s embedded formula to read as follows:

```
=$L$5
```

7. Select the textbox and perform the following modifications:
 - Format the **Fill Color** to the **Orange, Accent 2** (Top row)
 - Set the **Font Size** to **14** points
 - Resize the textbox so the entire message can be displayed on a single line

Because we want the revenue number in the textbox to be large and orange, and the accompanying text to be white, we will have to split the information into two textboxes. Textboxes cannot contain information with mixed formatting.

There is an advantage to splitting the information into separate textboxes. If the file were to be opened on a computer with different regional formatting codes, the formatting codes in the formula may not match the regional codes. This will result in misrepresented values in our report.

To give us greater visual flexibility, as well as accommodate differing regional settings, perform the following steps:

8. On the **Calculation** sheet, select cell **L5** and **copy** (or CTRL-C) the formula into cell **M5** (CTRL-V).
9. Select cell M5 and update the formula to read as follows:

`=A6 & " Apps sold"`

10. Edit the formula in cell **L5** to only display the result of the SUM function:

`=SUM(G5:G9)`

11. Update the formatting of cell **L5** with a **Currency** style and **zero** decimal places.

NOTE: If you create a custom number format to display the Step 8’s result (i.e. “\$ #,##0”), you can add a space between the currency symbol and the first number. This will prevent the displayed value from appearing quite so compressed.

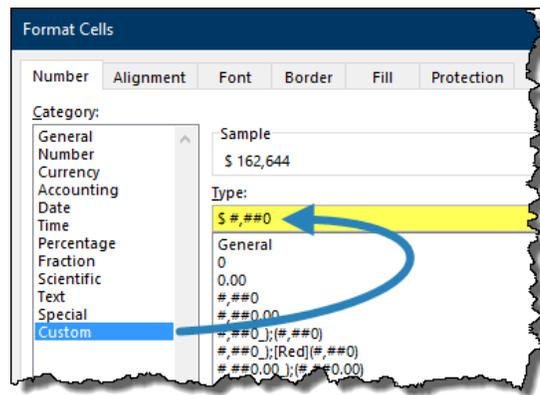


Figure 14.28 - Custom Currency Style Number Format

12. Resize the textbox to make it smaller but remember to keep the textbox wide enough to accommodate larger numbers that may be displayed.
13. Move the textbox from the **Calculation** sheet to the **Dashboard** sheet and place it in the lower-left area of the report.

It appears we have “lost” our sales. This is because the pointer for the textbox does not retain its original sheet reference (i.e. `=Calculation!L5`). The reference remained `=L5`.

14. Undo the previous step (**Step 11**) to place the textbox is back on the Calculation sheet.

To ensure this does not happen again, and to make the report more resilient to change, we will utilize the Name Manager to return the SUM calculation’s result to the textbox.

15. On the **Calculation** sheet, select cell **L5**. Using the **Name Manager** or the **Name Box**, name the cell “**TotalRev**”.
16. Update the textbox from “`=L5`” to “**TotalRev**”.

17. Re-format the textbox with an **Orange** font and a font size of **20** points.
18. Move (cut-and-paste) the textbox from the **Calculation** sheet to the **Dashboard** sheet into the lower-left corner of the report.
19. On the **Dashboard** sheet, add a textbox with the following embedded formula:

```
=Calculation!M5
```

20. Format the textbox's font color to **White** and the font size to **14** points.

Remember to space out the textboxes to allow room for larger values. (Alignment will be taken care of later in the module.)

21. On the **Calculation** sheet, select cell **K6** and enter the title "**Max**".

22. In cell **L6**, enter the following formula:

```
=INDEX(F5:F9,MATCH(MAX(G5:G9),G5:G9,0))
```

23. In cell **K7** and enter the title "**Min**".

24. In cell **L7**, enter the following formula:

```
=INDEX(F5:F9,MATCH(MIN(G5:G9),G5:G9,0))
```

We will now concatenate the Min/Max answers with the explanatory text.

25. Select cell **L6** and add the following after the INDEX/MATCH function:

```
=INDEX(F5:F9,MATCH(MAX(G5:G9),G5:G9,0))&" bought the most " & A6 & " apps"
```

26. Select cell **L7** and add the following after the INDEX/MATCH function:

```
=INDEX(F5:F9,MATCH(MIN(G5:G9),G5:G9,0))&" bought the least " & A6 & " apps"
```

27. On the **Dashboard** sheet, copy and paste the textbox and update the embedded formula to read as follows:

```
=Calculation!L6
```

28. Copy and paste the textbox and update the embedded formula to read as follows:

```
=Calculation!L7
```

Again, we have lost formatting during the previous steps. This can be easily recreated by performing the following steps:

29. Select the “... **apps sold**” textbox and press **CTRL-Shift-C** on the keyboard.
30. Select the “**most**” textbox and press **CTRL-Shift-V** on the keyboard to apply the formatting of the previous textbox.
31. Perform the identical procedure to copy the formatting from the “**most**” textbox to the “**least**” textbox.

NOTE: You could have also used the Format Painter tool to transfer the appearance settings from one textbox to another, but it’s a good idea to know the CTRL-Shift-C/V keyboard shortcut.

32. Select the “**least**” textbox and change the font color to a **light orange 60%** (6th column, 2nd row).

Now we apply alignment settings to the textboxes to present a more professional look to the report.

33. Select the **orange dollar-value** textbox. Hold down **CTRL** and select the “**most**” and the “**least**” textboxes.
34. Align the three textboxes with a **Left Alignment** setting.
35. Select the **orange dollar-value** textbox. Hold down **CTRL** and select the “**apps sold**” textboxes.
36. Align the two textboxes with a **Bottom Alignment** setting.

The report should appear as follows at the end of this module:

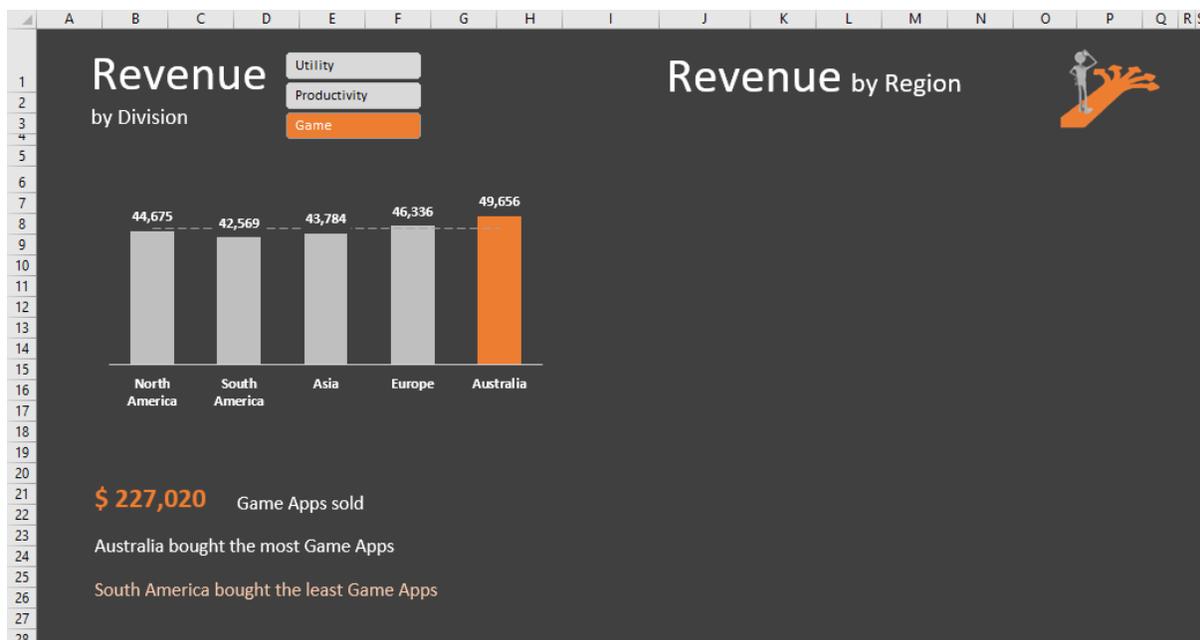


Figure 14.29 - Finished Report at the End of Module 14.5

Module 15

Advanced Pivot Table Techniques

XelPlus.com

MODULE 15.1 – VARIANCES IN PIVOT TABLE

Data File:

15 – Advanced Pivot Table Techniques\15.1 – Variances in Pivot Table.xlsx

In this lecture, we want to create a table that displays revenue by Region. Although we want to group the results by Division, we want the Region to be variable and controlled by the user with a slicer.

The table to be added will be a Pivot Table and contain conditional formatting created in the following module. We will create the table on the Calculation sheet to ease development and testing. Once completed, we will relocate the Pivot Table from the Calculation sheet to the Dashboard sheet.

1. Copy the titles from cells **A2:D2** and paste them into cells **A12:D12**. Update the title to read **“Revenue by Region – Pivot Table”**.

We want to create a new Pivot Table, and we want to ensure we don't create a second pivot cache, thereby adding unnecessary bloat to our file's size. To ensure both pivot tables utilize the same pivot cache, we will copy the Division Pivot Table and then adjust the output of the pasted pivot table.

2. Select cell **A6** (the original Division Pivot Table). Select **Pivot Table Tools → Analyze → Actions (group) → Select → Entire Pivot Table**.
3. Click **Copy** (or CTRL-C).
4. Select cell **A14** and click **Paste** (or CTRL-V).
5. Update the Pivot Table's field settings as follows:
 - Add **“Revenue”** to the **Values** section
 - Add **“Year”** to the **Column** section

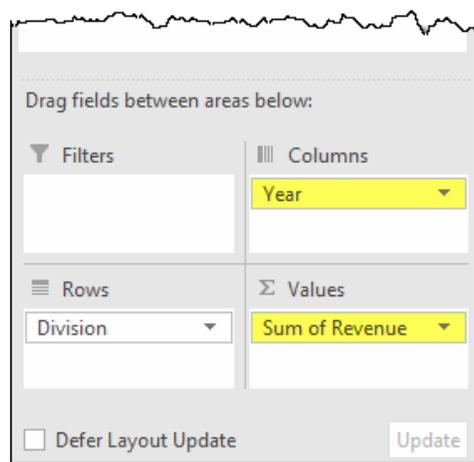


Figure 15.1 - Results of Module 15.1 - Step 5

We want to show all Divisions in the second Pivot Table. Because the two PivotTables are “linked” by the same slicer (when you change the slicer for one Pivot Table, you change both PivotTables), we need to break the dependency between the slicer and the second Pivot Table.

- With the second Pivot Table selected, select **Pivot Table Tools → Analyze → Filter (group) → Filter Connections** and clear the checkbox for the Division slicer.

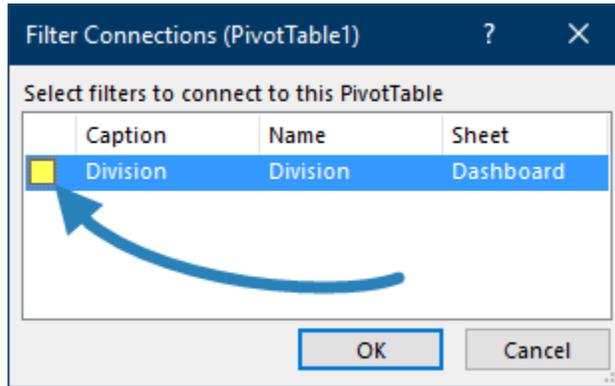


Figure 15.2 - Filter Connections Dialog Box

NOTE: An alternative approach would be to select the slicer and then select **Slicer Tools → Options → Slicer (group) → Report Connections**. The Report Connections dialog box will list all connected pivot tables to the selected slicer. You can select or clear multiple pivot table connections in a single action.

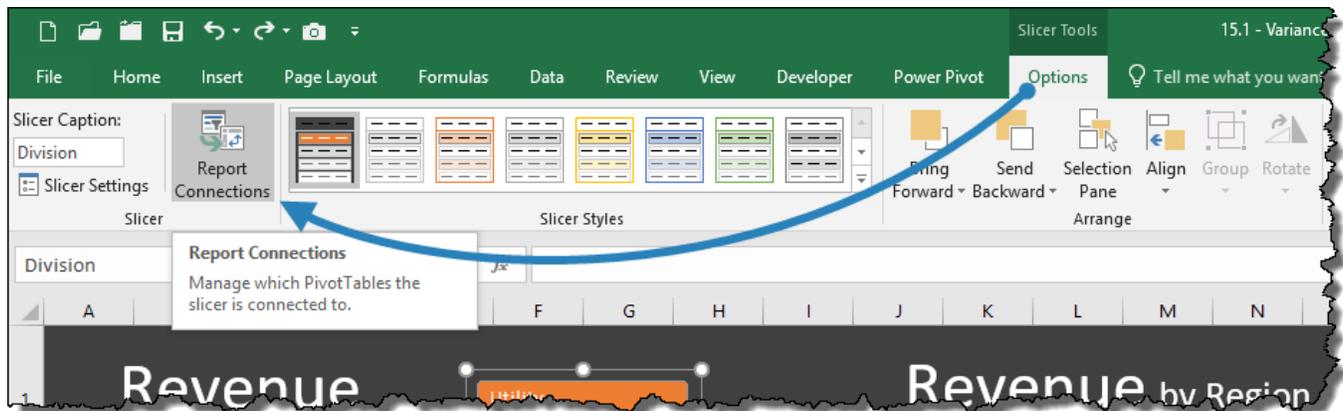


Figure 15.3 - Slicer Tools - Report Connections

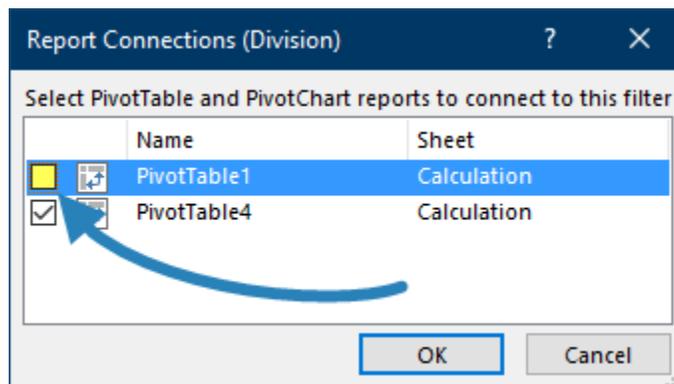


Figure 15.4 - Report Connections Dialog Box

7. Clear the filter from the second Pivot Table.
8. Update the “crowded” look of the second Pivot Table by performing the following visual modifications:
 - Turn off the **Field Headers**
 - Set the number style for “**Current**” and “**PY**” to a **Number** style with a **comma** separator and with **zero** decimal place precision

Now we want to add a calculation that shows the percentage change (% Δ) to the previous year. We want to know if we are doing better or worse in comparison to the previous year’s revenue.

9. With the second Pivot Table, add the **Revenue** field a 2nd time to the “**Values**” section. Expand columns **D** & **E** if necessary.
10. Select cell **C16** and **right-click** → **Show Values As** → “**% Difference From...**”.
11. In the “**Show Values As**” dialog box, set the parameters as follows:

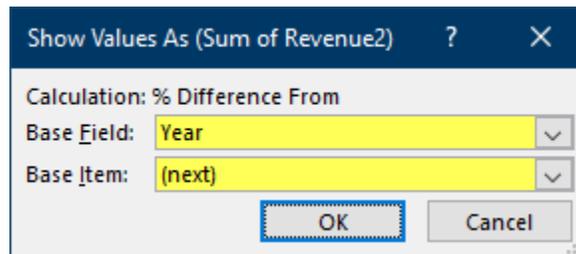


Figure 15.5 - Show Values As Dialog Box

NOTE: This can be confusing because the terms “previous” and “next” do not refer to years, rather to items in the Pivot Table. The values in Column “C” need to be compared against the values in Column “D” (the next column.) If the contents of Column “D” were to move position, this would “break” the logic. If your Pivot Table is subject to structural changes, it would be better to set the “Base Item” option to point to the field used in the Pivot Table (PY) instead of the ordered position of the item in the Pivot Table.

12. Select cell **C16** and **right-click** → **Show Values As** → “**% Difference From...**”.
13. In the “**Show Values As**” dialog box, set the parameters as follows:

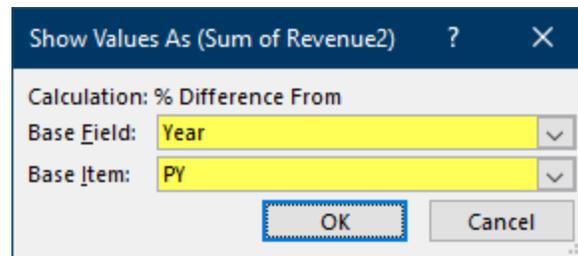


Figure 15.6 – Updated Show Values As Dialog Box

NOTE: Column “E” will remain empty.

14. Update the headings in the Pivot Table with the following titles:

- **B15** “**Revenue**” (Because we have a field named “Revenue”, Excel does not allow for custom names that match existing field names. To overcome this restriction, add a space after the word “Revenue”.)
- **C15**..... “% Δ PY” (The “Δ” is a symbol accessible from the Insert → Symbol feature.)

15. Add a total to the Pivot Table by selecting **Pivot Table Tools → Design → Layout (group) → Grand Totals → “On for Columns Only”**.

16. Rename cell **A19** to “**Total**”.

We will now customize the look of the Pivot Table.

17. Select cell **A16**. On the **PivotTables → Design** ribbon, right-click on the active style (3rd from the left) and select **Duplicate**.

18. Name the new Pivot Table style to “**Slicer Pivot**” and set the **First Column → Format → Font → Bold → OK** (2 times).

It appears as if the custom Pivot Table style did not take effect. By duplicating the style and modifying the new version, we have merely added the new style to the PivotTables Styles library. We must now apply the new style to our Pivot Table to see the new color scheme in action

19. From the **Pivot Table Styles library**, select the newly created “**Slicer Pivot**” style.

Pivot Table font size cannot be customized with a style. To alter the size of fonts in a Pivot Table, the cells containing the text or values must be sized directly with the traditional Font Size control on the Home ribbon.

20. Select cells **A16:D19** and set the font size to **12** points.

21. Select cells **C16:C19** and set the font size to **11** points.

Now we will alter the cell alignment of the Pivot Table headings.

22. Select cells **B14:E15** and set the cell alignment to **Right Aligned**.

23. **Refresh** the Pivot Table.

NOTE: Notice how the alignment of the headings has returned to a left-aligned position. Pivot Table resets can be extremely frustrating. The trick to prevent undesired formatting resets is to alter the Pivot Table options.

24. With the Pivot Table selected, click **Pivot Table Tools → Analyze → Select (group) → Entire Pivot Table**. Set all the selected cells to a **Left Aligned** cell alignment.

25. Select cells **A16:A19** and return the heading alignment to the left

	A	B	C	D	E
12	Revenue by Region - Pivot Table				
13					
14		Current		PY	
15		Revenue	% Δ PY	Revenue	% Δ PY
16	Game	227,020	13.57%	199,894	
17	Productivity	162,644	1.80%	159,774	
18	Utility	189,979	1.53%	187,117	
19	Total	579,643	6.01%	546,785	
20					

Figure 15.7 - Output of Module 15.1

MODULE 15.2 – ADDING CONDITIONAL FORMATTING IN PIVOT TABLE

Data File:

15 – Advanced Pivot Table Techniques\15.2 – Adding Conditional Formatting in Pivot Table.xlsx

In this lecture, we want to add conditional formatting to the Pivot Table. We will add conditional formatting to “% Δ PY” using up (green) and down (red) arrow heads from the Icon Sets library. This will allow us to immediately recognize if the percent change was positive or negative compared to the previous year.

We need to ensure that we add the conditional formatting to the Pivot Table as a feature and not to specific cells within the Pivot Table. This will ensure that the conditional formatting extends to any new Divisions added later.

1. Select cell **C16** (any cell between C16:C18 will work).
2. Select **Home** \rightarrow **Styles (group)** \rightarrow **Conditional Formatting** \rightarrow **New Rule**.

Notice the difference between a standard Conditional Formatting and the one presented here. This Conditional Formatting dialog box contains an additional set of controls at the top of the dialog box.

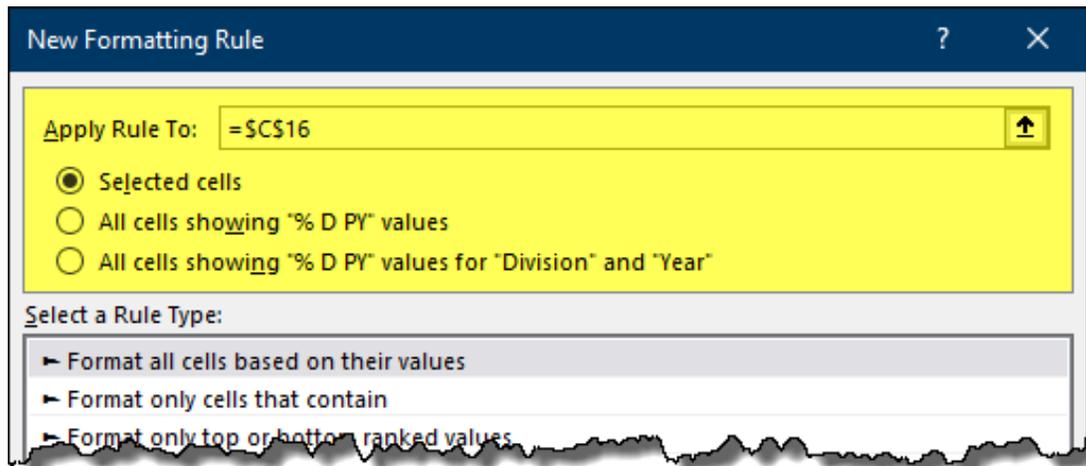
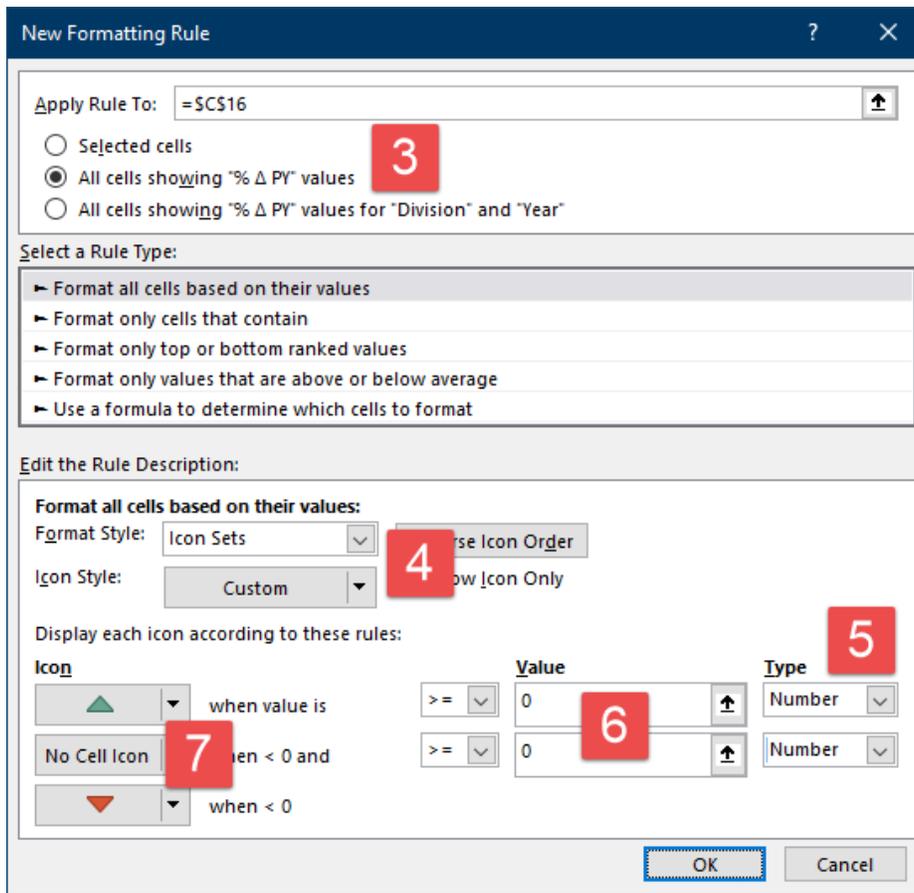


Figure 15.8 - Conditional Formatting Dialog Box for Pivot Tables

The difference between the options are as follows:

- “Selected Cells” – Applies conditional formatting to only the selected cell(s)
- “All cells showing % Δ PY” values – Applies conditional formatting to all cells containing values (includes subtotals and totals)
- “All cells showing % Δ PY” values for “Division” and “Year”” – Applies conditional formatting only to base-level values (excludes subtotals and totals)

Set the following parameters for the Conditional Formatting Pivot Table rule:



3. Select “**All cells showing % Δ PY**” values at the top of the conditional formatting dialog box.
4. From the **Format Style** dropdown list, select **Icon Sets** and select the ▼ ■ ▲ icons from the Icon Style dropdown list.
5. Change both **Type** dropdown settings from **Percent** to **Number**.
6. Set both **Value** options to **0** (zero) (if necessary).
7. Set the **middle icon** to “**No Cell Icon**”.
8. Test the conditional formatting by altering the data. On the **Data** sheet, change the value of cell **D7** to **0** (zero). Return to the **Calculation** sheet and refresh the Pivot Table.

An issue with the conditional formatting is that the arrow icons are too large. If we change the font size of the cells, both the values and the arrow heads are altered. We want to retain the larger font size for the values but have a smaller, less obtrusive size for the icons.

To achieve this effect, we will separate the values from the arrow head icons by placing the arrow head icons in a separate column. With the values separated from the icons, we can make the icons smaller without affecting the values.

9. Clear the rules from the Pivot Table by selecting **Home → Styles (group) → Conditional Formatting → Clear Rules → Clear Rules from This Pivot Table**.
10. Add **“Revenue”** to the **Values** section of the Pivot Table at the **bottom** of the stack.
11. Select cell **D16** and **right-click → Show Values As → “% Difference From...”**.
12. In the **“Show Values As”** dialog box, set the parameters as follows:

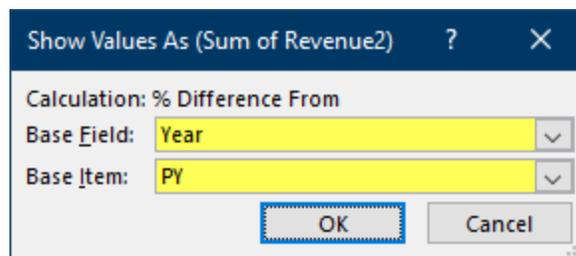


Figure 15.9 – Updated Show Values As Dialog Box

13. Repeat steps **1 through 7** on cell **D16**. Add the additional step of selecting the option for **“Show Icon Only”**. This will prevent the values from being displayed a second time in the Pivot Table.

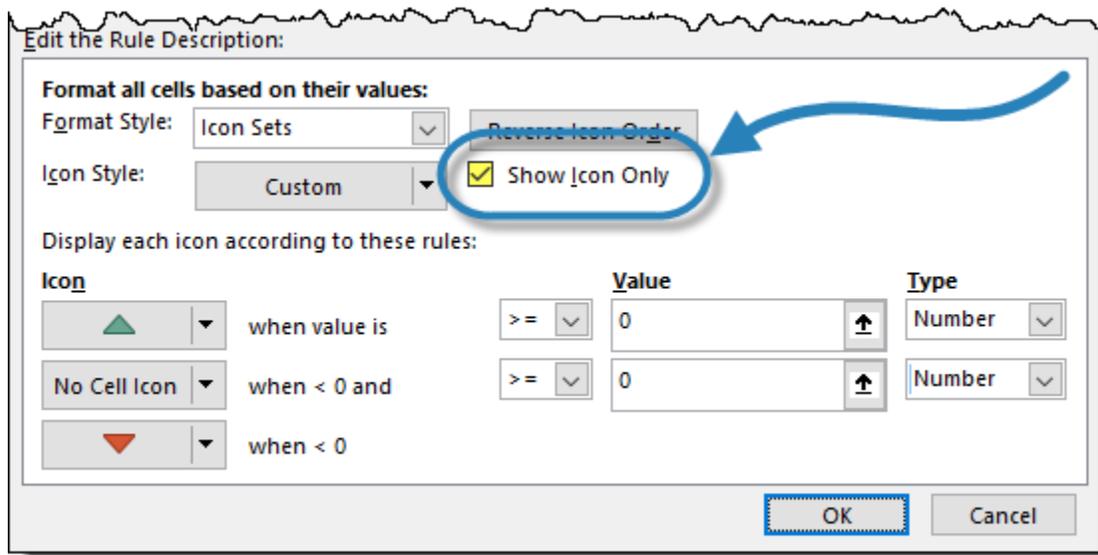


Figure 15.10 - "Show Icon Only" Feature in Conditional Formatting

14. Remove the heading in cell **D15** by entering a **“space”**.
15. Reduce the column width of **Column “D”** and set the cell alignment to **Left Aligned**.

	A	B	C	D	E	F	G
12	Revenue by Region - Pivot Table						
13							
14		Current			PY		
15		Revenue	% Δ PY		Revenue	% Δ PY	
16	Game	227,020	13.57% ▲		199,894		
17	Productivity	128,489	-19.58% ▼		159,774		
18	Utility	189,979	1.53% ▲		187,117		
19	Total	545,488	-0.24% ▼		546,785		
20							

Figure 15.11 - Output of Module 15.2

MODULE 15.3 – PIVOT TABLE AS PART OF DASHBOARD REPORT

Data File:

15 – Advanced Pivot Table Techniques\15.3 – Pivot Table as Part of Dashboard Report.xlsx

It is time to move the Pivot Table to the Dashboard sheet and add a slicer to control the Region(s). Currently, we are looking at ALL the region's revenue. The slicer will allow us to filter the Pivot Table to only display a selected region or regions.

Before we begin, let's make a couple minor cosmetic changes to the Pivot Table:

- Set the “% Δ PY” values to display **1 decimal place** precision
 - Name the Pivot Table to “**Pivot_Div**”
1. On the **Calculation** sheet, select the Pivot Table (cell **A16**).
 2. Select **Pivot Table Tools** → **Analyze** → **Move Pivot Table**. Select “**Existing Worksheet**” and use cell **J20** on the **Dashboard** sheet as the destination.

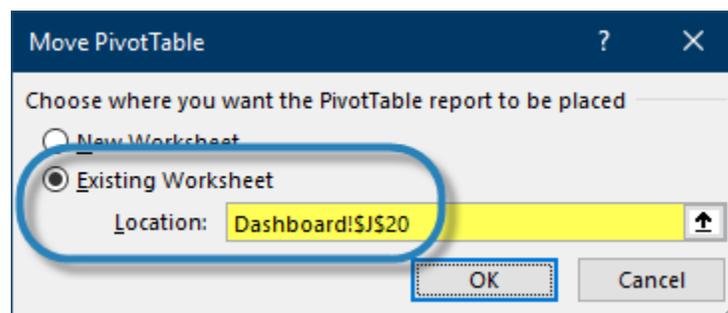


Figure 15.12 - Move Pivot Table Dialog Box

3. On the **Dashboard** sheet, expand **Column “J”** if necessary.
4. We do not need to see Columns “O” & “P”. Hide **Columns “O” & “P”**, then expand **Column “R”** to compensate for the lost background.
5. Add a slicer for the **Region** by selecting **Pivot Table Tools** → **Analyze** → **Insert Slicer** → **Region**.
6. Assign the custom slicer format created in **Module 14.3** to the newly inserted slicer for **Regions**.

Because we were hiding the slicer headings in the Division slicer, we failed to notice a problem with the heading; the heading text is too dark to read. We will adjust the custom slicer style to compensate for darker headings.

7. Right-click the custom slicer style and select **Modify**.
8. Select **Header** → **Format** → **Font** → **Color** → **Orange** (top row).

9. Resize the slicer's height and width to fit the Region names better.

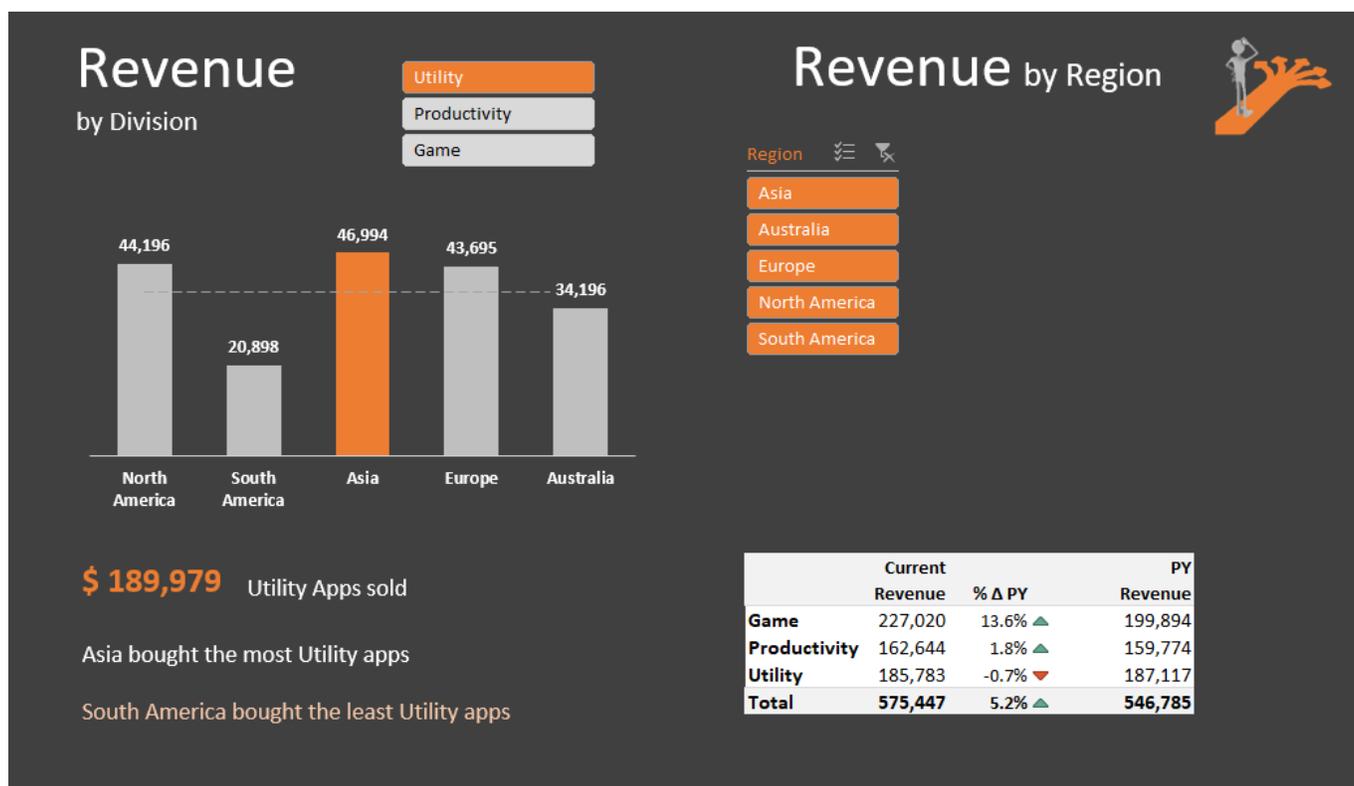


Figure 15.13 - Output of Module 15.3

MODULE 15.4 – PIVOT CHARTS FOR DASHBOARDS

Data File:

15 – Advanced Pivot Table Techniques\15.4 – Pivot Charts for Dashboards.xlsx

We will now add a Pivot Chart that will display Revenue by Region and by Division. The Pivot Chart will also be controlled by the Region slicer.

The Pivot Chart will be a simple 3-column chart where each column represents a Division.

Pivot Charts require a Pivot Table to drive their presentation of the data. Although we have a Pivot Table that contains the necessary information, it contains more information than we want for our chart. To keep from building a whole new Pivot Table, we will place a copy of the Pivot Table on the Calculation sheet and then modify it to suit our needs.

1. On the Dashboard sheet, select any cell in the Revenue Pivot Table and select **Pivot Table Tools** → **Analyze** → **Actions (group)** → **Select** → **Entire Pivot Table**.
2. Click **Copy** (or CTRL-C).
3. Switch to the **Calculation** sheet and select cell **K16** and click **Paste** (or CTRL-V).
4. Select the headings in cells **K2:N2** and **copy/paste** the headings into cell **K12**.
5. Change the newly copied heading's title to "**For Chart – Div. Revenue by Region**".

Notice the note in cell A14 reminding us that we moved the Pivot Table to the Dashboard sheet. In an upcoming module, we will move the Pivot Table back here and explore the benefits of Excel's Camera Tool feature.

Now we will update the chart to remove the categories we do not need.

6. Update the Pivot Table's field settings as follows:
 - Remove " " from the **Values** section (*blank named field at bottom of Values list*)
 - Remove "% Δ PY" from the **Values** section
 - Relocate "Year" from the "Columns" section to the "Filters" section.

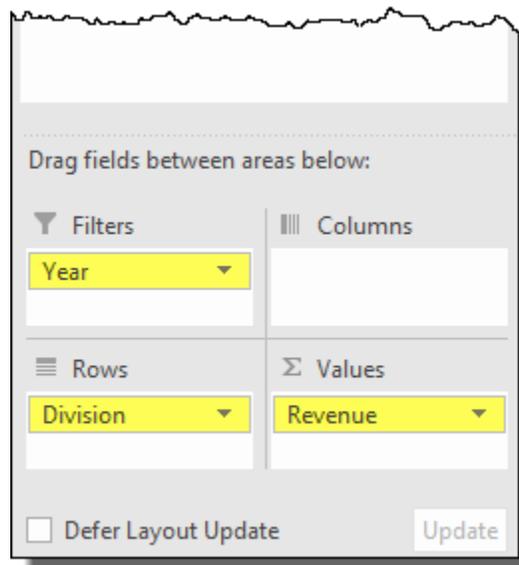


Figure 15.14 - Update Pivot Table Settings

7. Click the “**Year**” filter control (cell **L14**) and set the filter to “**Current Year**”.

We now have a data set that will be perfect for our Pivot Chart.

8. Select any cell in the Pivot Table (i.e. **K18**) and select **Pivot Table Tools** → **Analyze** → **Tools (group)** → **Pivot Chart**.

9. From the **Insert Chart** dialog box, select **Clustered Column** (top-left choice) and click **OK**.

10. Remove the following items from the Pivot Chart:

- Legend
- Gridlines
- Vertical axis (y-axis)
- Chart title
- Field Buttons (*done from Analyze ribbon*)

11. Add data labels to the chart’s columns (*right-click bar* → *Add Data Labels*)

12. Reduce the **Gap Width** (*right-click bar* → *Format Data Series*) to **50%**

13. Move the Pivot Chart to the **Dashboard** sheet.

14. Move and size the Pivot Chart to fit to the right of the Region slicer and above the Pivot Table.

15. Remove the background fill color and border color from the Pivot Chart.
16. Use the Format Painter to copy the formatting of the horizontal axis (x-axis) labels on the “Revenue by Division” chart and past the formatting to the horizontal axis of the “Revenue by Region” Pivot Chart.

NOTE: You can perform the Format Painter operation with a keyboard shortcut. Select the labels of the first chart and press “CTRL-Shift-C”. Next, select the labels of the second chart and press “CTRL-SHIFT-V”.

17. Perform the same paint formatting operation for the **data labels**.
18. Test the slicer to ensure the Pivot Chart is responding to the user’s selection(s).

One of the unfortunate but necessary actions a user must execute on PivotTables is manually refreshing, or updating, the Pivot Cache when the source data changes. We will learn in how to automate this action in the upcoming module.

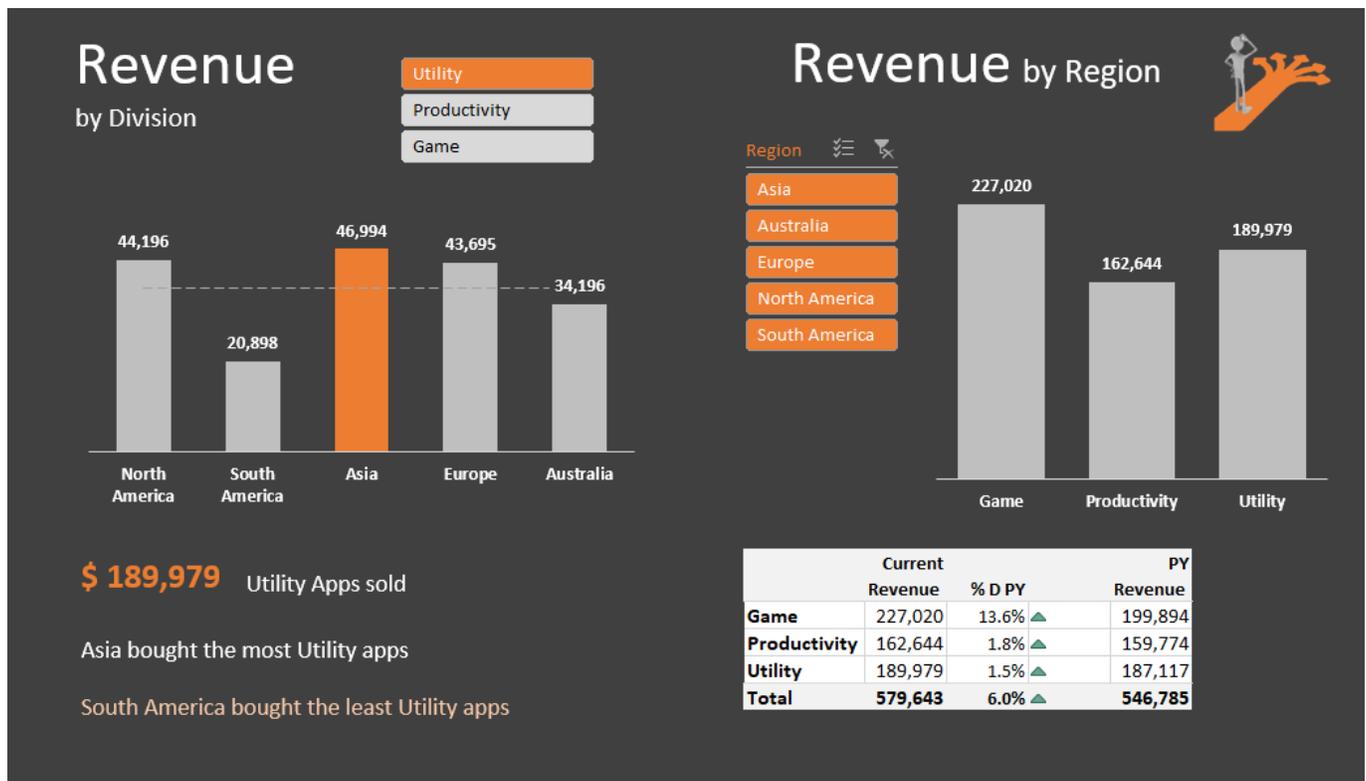


Figure 15.15 - Output of Module 15.4

MODULE 15.5 – VBA CODE TO AUTOMATICALLY REFRESH PIVOT TABLE

Data File:

15 – Advanced Pivot Table Techniques\15.5 – VBA Code to Automatically Refresh Pivot Table.xlsx

The key thing to remember about PivotTables is that they need to have their pivot caches refreshed when the source data changes.

To demonstrate the problem, on the Data sheet, select cell D2 and change the value to 8. Return to the Dashboard sheet and observe that the North American Revenue for Utility has not changed; it remains its original value of 44,196. Right-click on the pivot table and select “Refresh”. The Pivot Table now reflects the updated value or 8.

Unfortunately, there is no Pivot Table option to have the Pivot Table automatically refresh its cache when the underlying data changes. The closest we can get is to go into the Pivot Table’s options (**Pivot Table Tools → Analyze → Pivot Table (GROUP) → Options**) and select the **Data** tab. In the **Pivot Table Data** group of controls, we can select the option labeled “**Refresh data when opening the file**”.

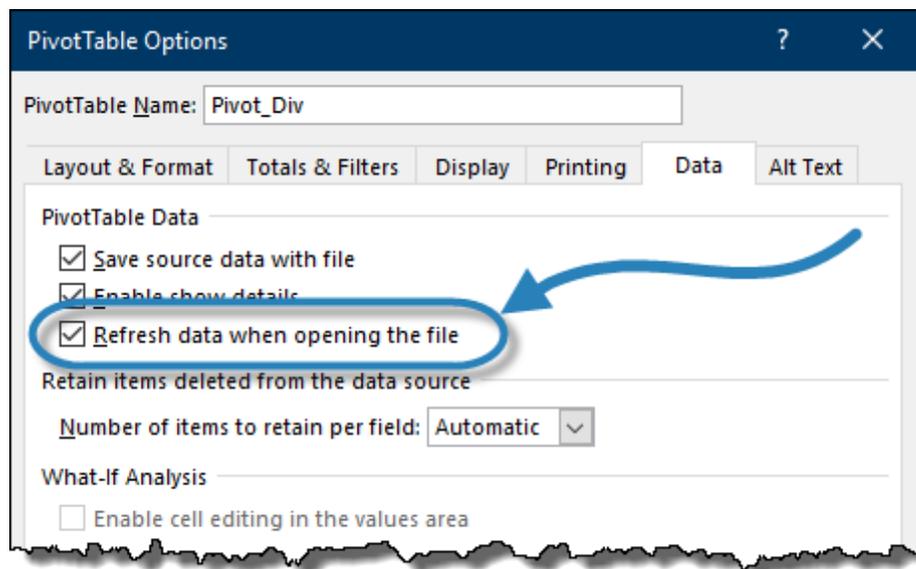


Figure 15.16 - Pivot Table Options Dialog Box

Although this works, it is rarely practical because we don’t want to have to close and reopen the file every time we make changes to the data.

If you forget to refresh the data, you may make incorrect business decisions based on “stale” data.

With one line of “save the day” VBA code, we can have auto-updating PivotTables. With VB, this is easily implemented.

1. Right-click on the **Data** sheet tab and select **View Code**.

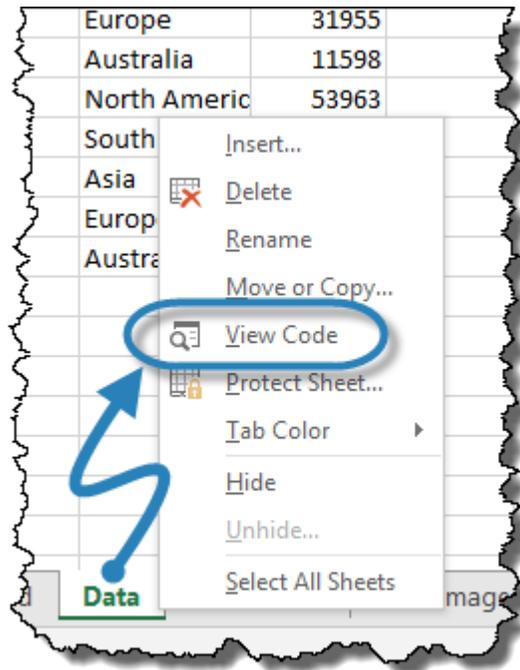


Figure 15.17 - View Code Option to Access VBA Editor

2. Change the Object type dropdown from **General** to **Worksheet**.

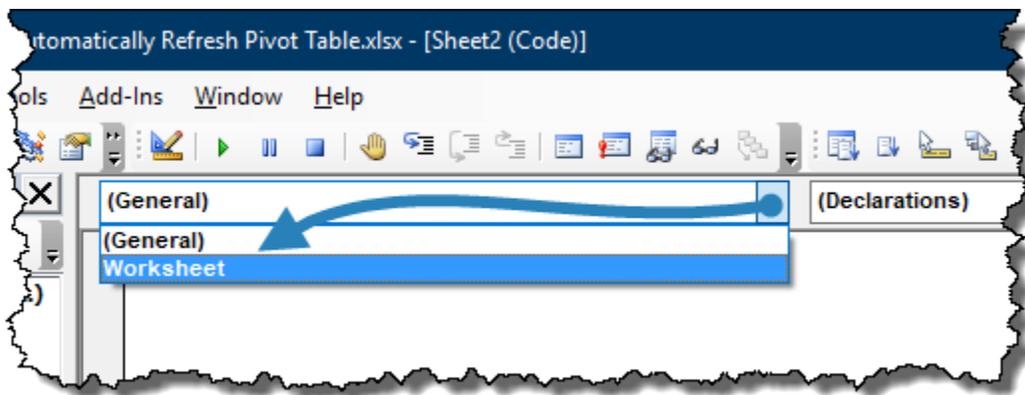


Figure 15.18 - Set Object Type to Worksheet

This automatically creates an “empty shell” of code to initialize a Change Event. A Change Event is a feature of VBA that executes any code placed within the Change Event “shell” any time a cell’s content is changed. This Change Event only monitors the cells of the sheet at which the code is stored. In this case, the Data sheet.

Because all the PivotTables share the same pivot cache, the code for the Change Event will update all the PivotTables in the workbook.

3. Select the **Calculation** sheet and select cell **A6**. Observe the name of the Pivot Table (“**Pivot_Region**”)

4. Return to the VBA Editor (*Tip: the keyboard shortcut ALT-F11 will perform the switch quickly.*)
5. Place your cursor between the text “Private Sub...” and “End Sub” and enter the following line of code:

```
Worksheets("Calculation").PivotTables("Pivot_Region").PivotCache.Refresh
```

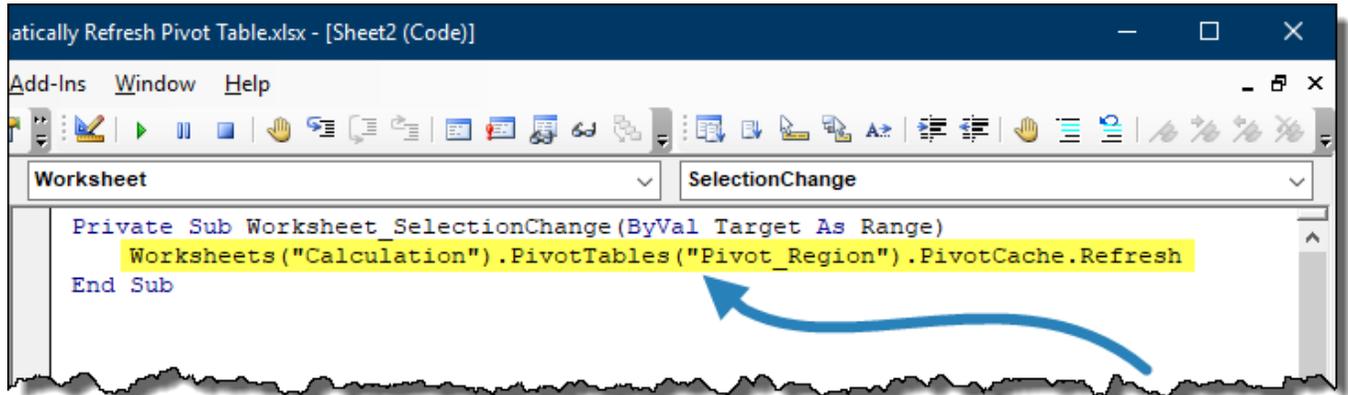


Figure 15.19 - Updated Change Event Code

If you use this for other projects, change the worksheet’s name and the Pivot Table’s name accordingly.

6. Test the code by changing the value for **North America – Utility** on the **Data** sheet. Return to the **Calculation** sheet to see if the value has been updated to reflect the change.

WARNING: because the Pivot Table will refresh when ANYTHING on the Data sheet changes⁴, you should reserve the Data sheet for Pivot Table support data ONLY.

Because this file contains VBA code, we must save the file as a **Macro-Enabled Excel file**. This fill will possess a **.XLSM** extension.

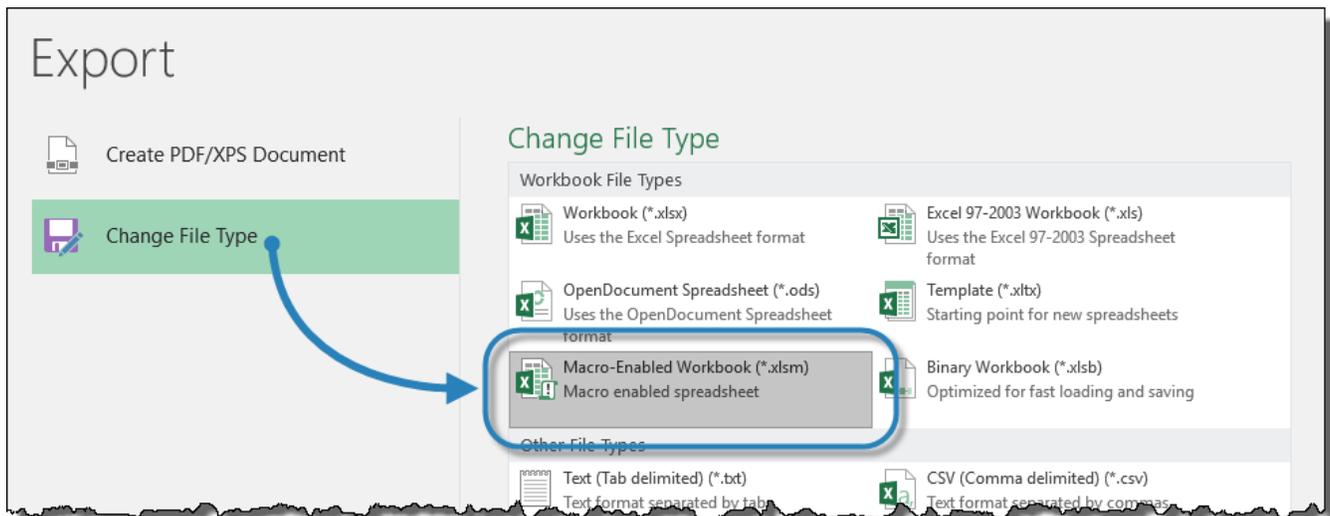


Figure 15.20 - Save As Dialog Box with Macro Enabled Workbook Selection

*TIP: The updates are being detected by a VBA event called a **Change Event**. This means any time a cell's contents been added, changed, or deleted the code will be executed. In our case, we are triggering a Pivot Table refresh every time we alter any of the data. A minor, and perhaps unnoticeable issue is that since we cannot see the dashboard while we are performing updates to the data, executing an update after every cell change seems a bit excessive. On smaller data sets this would not result in any performance issues. On larger data sets where changing a single cell may require several seconds of updates, we may be better served by queueing up all the changes until we are ready to return to the dashboard.*

*By making a small change to the VBA code, we can defer all the updates until we return to the dashboard. The change is to use a **Deactivate** event instead of a **SelectionChange** event. This has the potential to greatly improve performance by executing a single update for all changed data during that modification session.*

To use this tactic, change the first line of the VBA code from:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

to

```
Private Sub Worksheet_Deactivate()
```

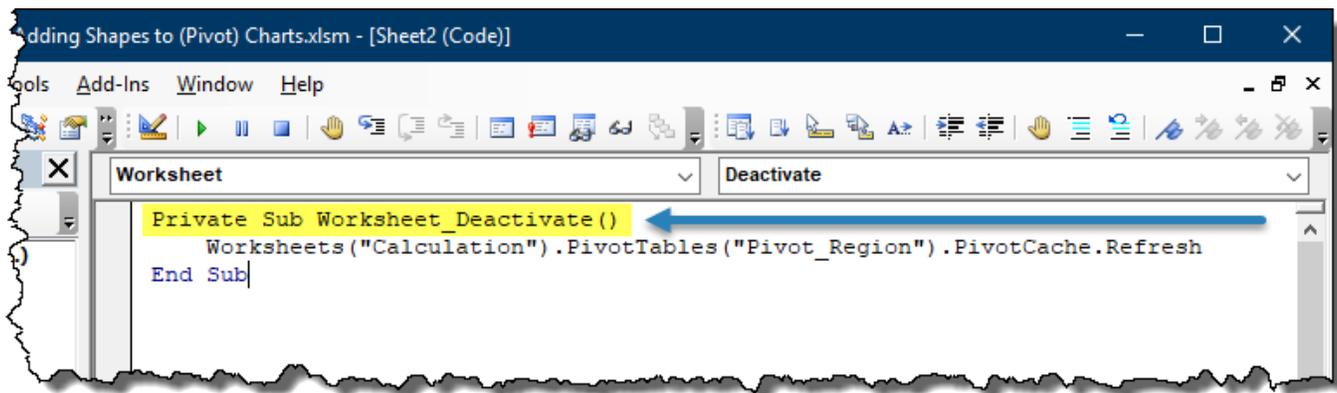


Figure 15.21 - Updated Pivot Table Refresh VBA Code

If you'd like to learn VBA from the ground up, check out my **Unlock Excel VBA & Macros** course:

XelPlus.com/courses

Module 16

Impressive Charts Using Shapes

XelPlus.com

MODULE 16.1 – ADDING SHAPES TO (PIVOT) CHARTS

Data File:

16 – Impressive Charts Using Shapes\16.1 – Info Charts - Adding Shapes to (Pivot) Charts.xlsm

This lecture will focus on transforming the visualizations the right side of the Dashboard sheet to appear as seen on the Final Image sheet.

We will create a non-standard Pivot Chart and reformat the Pivot Table to fit better and be more visually appealing.

These techniques are not limited to dashboards; they can be used in any report.

The good thing about our finished chart is that it is different while at the same time visually effective. Because it is different, people are more likely to examine it and thus remember the numbers.

Issues with the Pivot Chart:

- The first issue with the Pivot Chart is that it automatically adjusts the lower-bounds of the vertical axis (y-axis) based in the displayed data. If you select “Europe” from the Region slicer, the Pivot Chart makes “Utility” look dramatically lower than the other Divisions, when in fact, they are actually very close in value.
- The second issue with the Pivot Chart occurs when comparing the values of one Division against another Division. If you select “Europe” from the Region slicer, “Utility” is valued at \$43K. If you select “North America”, “Utility” is valued at \$40K. the issue is that the \$40K bar is taller than the \$43K bar. This is due to the comparative nature of the opposing Divisions.

To address these issues, perform the following steps:

1. Click anywhere on the chart.
2. Display the **Vertical Axis** (y-axis) by clicking the **Chart Elements** button  in the upper-right corner of the chart. Select **Axis** → **Primary Vertical**.

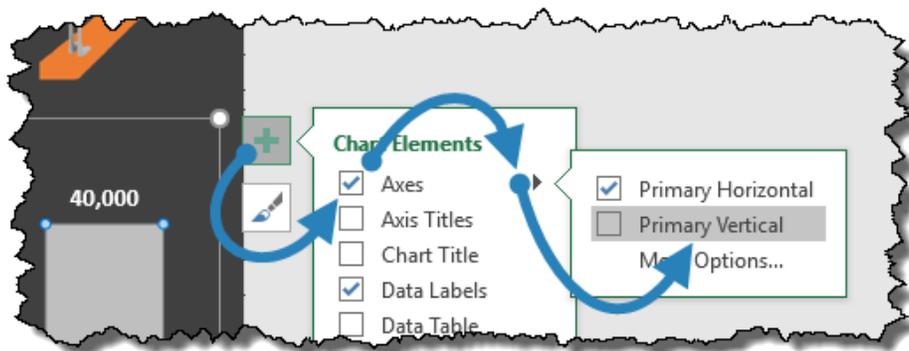


Figure 16.1 - Accessing Chart Elements Controls (Y-Axis)

3. Customize the **Vertical Axis** (y-axis) by right-clicking on the Vertical Axis and selecting **Format Axis...**
4. In the **Axis Options** section, set the Bounds for the **Minimum to 0** (zero).

Even though this is the currently selected value, Excel will adjust this value based on the displayed data. By “hard coding” the value to 0, this will be the lower bounds of the vertical axis regardless of selected Region.

5. Test the results by selecting various **Regions**.
6. Remove the **Vertical Axis** (y-axis) from the Pivot Chart.

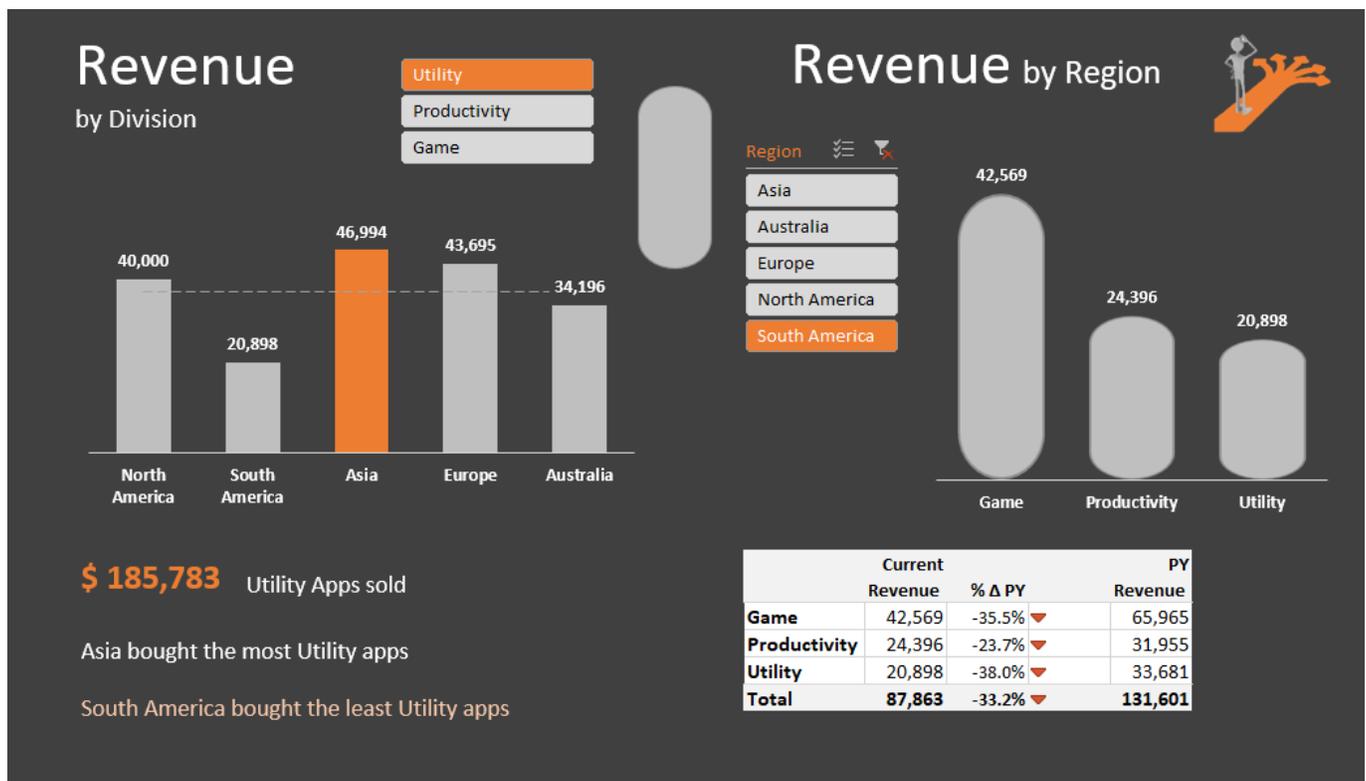


Figure 16.2 - Output of Module 16.1

MODULE 16.2 – OVERCOMING PIVOT CHART LIMITATIONS

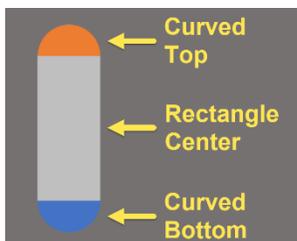
Data File:

16 – Impressive Charts Using Shapes\16.1 – Info Charts – Overcoming Pivot Chart Limitations.xlsm

The objective of this lecture is to normalize the bars in the chart so that we don't get a "squashed" or compressed look to the bars when there are short and tall bars present in the same Region.

If we examine the shape, the shape is made of three parts:

- A curved top
- A rectangular center
- A curved bottom



If we create a stacked column chart with three (3) shapes and set the top and bottom shapes to be the same value across all Divisions, only the center rectangle will stretch, which is not visually noticeable.

Because our chart is a Pivot Chart, and Pivot Charts are limited in their customizability, we will use the `GetPivotData` function to create a new data set based on elements of the original Pivot Table plus elements to control the top and bottom of the new chart bars.

[For a review on the purpose and use of the `GetPivotData` function, see Module 5 for details.](#)

1. Select the **Dashboard** sheet, remove the **Revenue by Region by Division** chart (*upper-right*).
2. Select the **Calculation** sheet and click cell **K14**. Select **Pivot Table Tools** → **Analyze** → **Actions (group)** → **Select** → **Entire Pivot Table**.
3. Press the **Delete** key on the keyboard to delete the Pivot Table.

*NOTE: An alternative way to delete a Pivot Table is to select any empty cell and press **Copy** (or **CTRL-C**), highlight the entire pivot table (this is easy when you have small Pivot Tables, but probably not worth the effort on large Pivot Tables), and press **Paste** (**CTRL-V**). Your Pivot Table is now a thing of the past.*

4. Select cell **K17** and enter the title "**Utility**".
5. Select cell **K18** and enter the title "**Productivity**".
6. Select cell **K19** and enter the title "**Game**".

7. Select cell **L17**, enter an equals sign (=) and then switch to the **Dashboard** sheet and select cell **K22** (*Revenue for Games*). The following formula should appear in the Formula Bar:

```
=GETPIVOTDATA("Revenue ",Dashboard!$J$20,"Year","Current","Division","Game")
```

8. We want the GetPivotData function to be dynamic, so alter the formula so the hard coded **“Game”** parameter points to the dynamic cell reference **K17**. The updated formula should appear as follows:

```
=GETPIVOTDATA("Revenue ",Dashboard!$J$20,"Year","Current","Division",K17)
```

9. Fill the **GetPivotData** function through cells **K17:K19**.

10. Select cell **L16** and enter the title **“Revenue”**.

We will now add the three “stacks”.

11. Select cell **M16** and enter the title **“Stack 1”**.

12. Select cell **N16** and enter the title **“Stack 2”**.

13. Select cell **O16** and enter the title **“Stack 3”**.

In stacks 1 through 3, we need a number that is common to all the Divisions. This will serve as the lower bound of the vertical axis (y-axis).

14. For **“Stack 1”**, select cell **M17** and enter the following formula:

```
=MIN($L$17:$L$19)/2
```

15. Fill the **MIN** function through cells **M17:M19**.

16. For **“Stack 3”**, **copy** the formulas from cells **M17:M19** and **paste** the formulas into cells **O17:O19**.

17. For **“Stack 2”**, select cell **N17** and enter the following formula:

```
=L17-M17-O17
```

18. Fill the formula through cells **N17:N19**.

Now we will create a “regular” chart based on the newly created data set.

19. Select cells **K16:K19**, press and hold the **CTRL** key, select cells **M16:O16**.

20. Select **Insert** → **Charts (group)** → **Stacked Column** (*top row, middle chart*).

21. Invert the grouping logic by selecting **Chart Tools** → **Data (group)** → **Switch Row/Column**.

22. Remove the following chart elements:

- Vertical axis (y-axis)
- Gridlines
- Chart title
- Legend

23. Decrease the **gap width** of the bars to **30%**.

We now see the Orange rectangles for “Stack 2”, the Dark Gray rectangles for “Stack 3”, and the Light Gray rectangles for “Stack 1”.

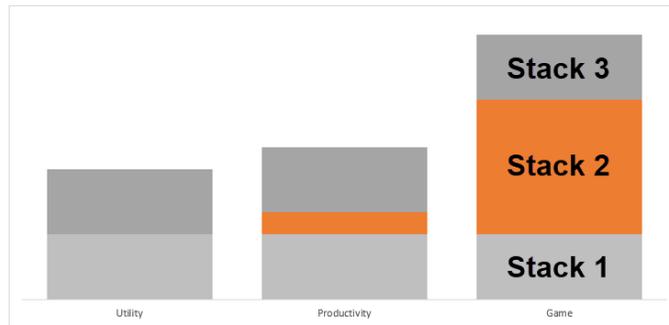


Figure 16.3 - Stacked Column Chart of Custom Data Set

We will now insert our shape  and rotate it 90° counter-clockwise  to achieve the cap of the column. This will be assigned to the top stack (dark gray rectangles).

24. Select **Insert** → **Illustrations (group)** → **Shapes** → **Flowchart: Delay** and draw the shape anywhere on the **Calculation** sheet.

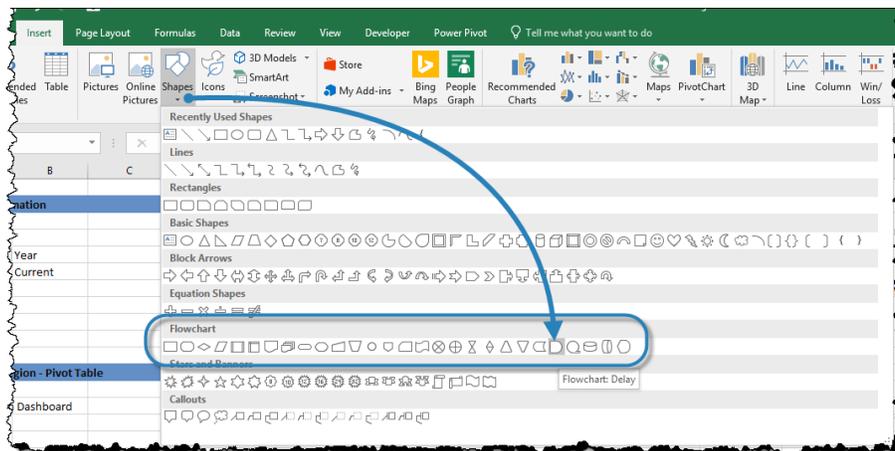


Figure 16.4 - Delay Shape in Flowchart Category of Shapes Library

25. Perform the following modifications to the newly inserted shape:

- Rotate the shape **90°** counter-clockwise
- Remove the outline from the shape
- Set the color of the shape to **25% gray** (*2nd from bottom gray*)

26. To make the bottom curve shape, select the newly created top curve shape and press **Copy** (or CTRL-C). Select an empty cell on the **Calculation** sheet and press **Paste** (or CTRL-V).

*NOTE: A fast way to duplicate any item in a sheet is to select the original object and press **CTRL-D**. This will duplicate the original object.*

27. Rotate the copied shape **180°**.

Assign the newly created shapes to the top and bottom rectangles in the chart by performing the following steps:

28. Select the **second shape** (lower curve) and press **Copy** (or CTRL-C).

29. Select the **bottom stack** (light gray) in the chart and press **Paste** (or CTRL-V).

30. Select the **first shape** (upper curve) and press **Copy** (or CTRL-C).

31. Select the **upper stack** (dark gray) in the chart and press **Paste** (or CTRL-V).

32. Select the **middle stack** (orange) and set the color to **25% gray** (*2nd from bottom gray*).

Now we will add and customize the data labels for the chart.

33. Right-click on the **upper stack** and select **Add Data Labels**.

We don't want to see the values of the upper stack (*10449*), we want to see the values from the Revenue

34. Select one of the data labels. In the **Format Data Labels** panel (right) deselect **Value** and select **Value From Cells**. Set the **Label Range** to the following formula:

=Calculation!\$L\$17:\$L\$19

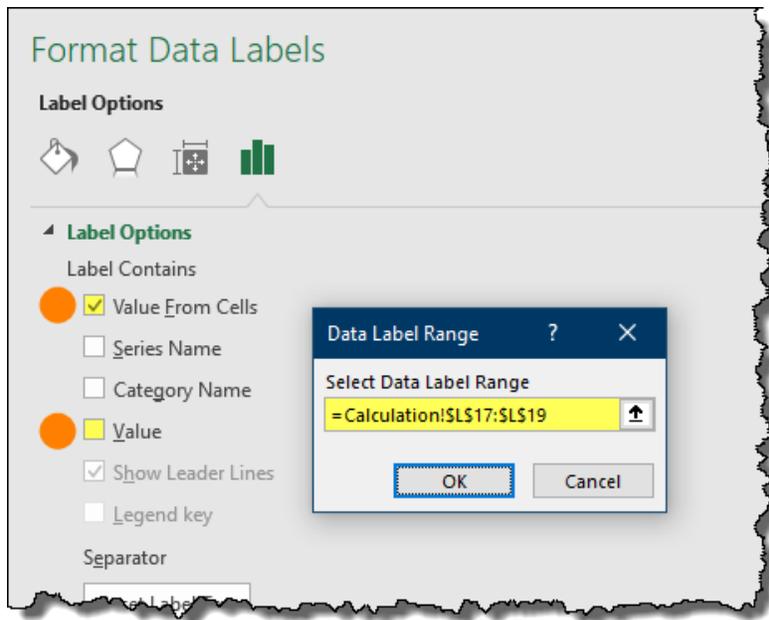


Figure 16.5 - Formatting Data Labels Options

35. Format the values in the labels by selecting cells **L17:L19** and apply a **comma style** with **zero** decimal place precision.
36. Select the data labels and make the font **Bold** and **10** points in size.
37. Move the chart from the **Calculation** sheet to the upper-right area of the **Dashboard** sheet. Size and position the chart as needed within the context of the other dashboard objects.
38. Remove the **fill color** from the **Plot Area** and the **Chart Area** of the chart.
39. Remove the **border** line from the **Chart Area**.

Apply the formatting of the Revenue by Division chart's horizontal axis (x-axis) labels and apply it to the Revenue by Region chart's horizontal axis (x-axis) labels.

40. Select the **x-axis** labels of the **Revenue by Division** (left) chart and press **CTRL-Shift-C**. Select the **x-axis** labels of the **Revenue by Region** (right) chart and press **CTRL-Shift-V**.
41. Select and **delete** the **test shape** (*gray, curve-ended rectangle*).
42. **Remove** the **Shape Outline** from the horizontal axis.

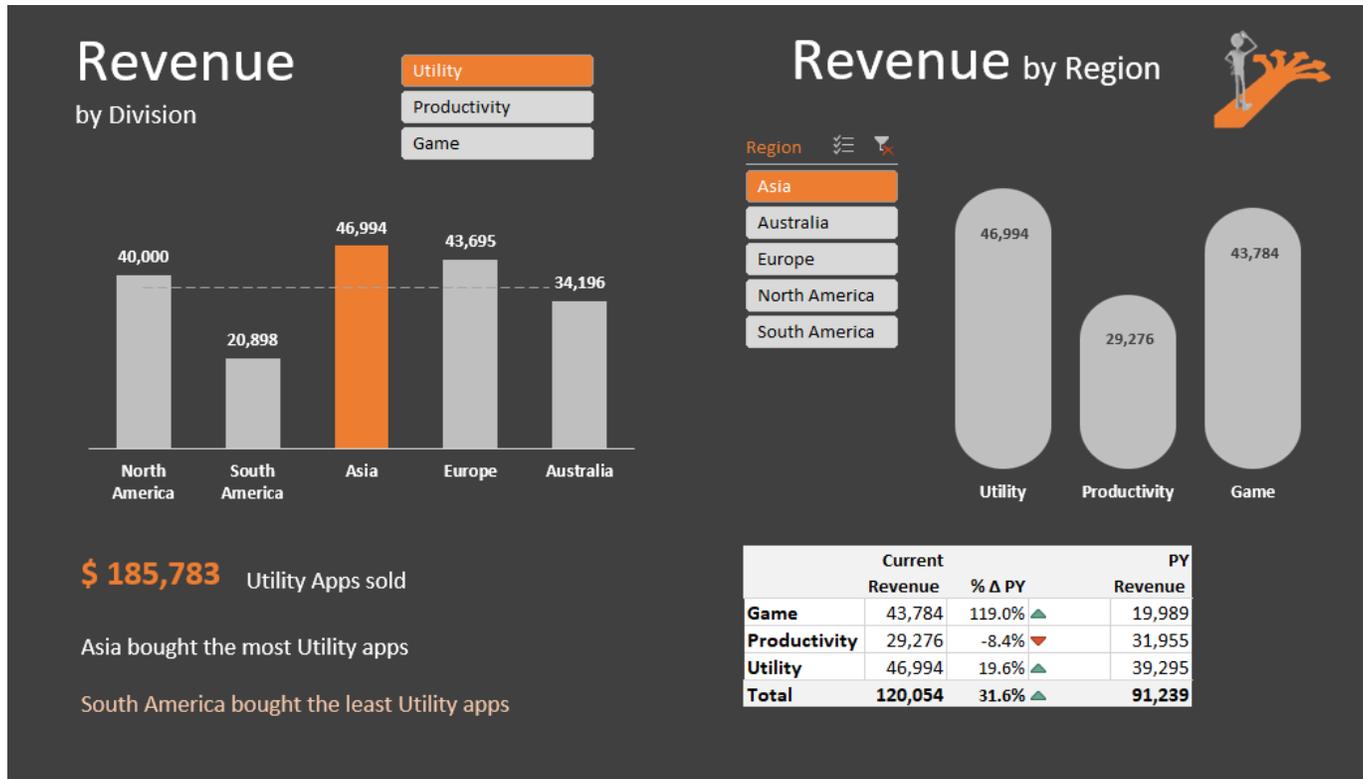


Figure 16.6 - Output of Module 16.2

Module 17

Secrets for Impressive Reports

XelPlus.com

MODULE 17.1 – CAMERA FOR IMPROVED DESIGN

Data File:

17 – Secrets for Impressive Reports\17.1 – Camera for Improved Design.xlsm

In this module, we will look at Excel’s Camera tool to see how we can get discontinuous sections of a spreadsheet to come together without the confines of row and column positioning.

If we had multiple tables, one above the other, both tables would have to adhere to the same column widths. If one table needs wide columns and the other table needs narrow columns, and they are both using the same sheet column, one table will be sized in a less than optimal width. The same can be said for side-by-side tables requiring differing row heights.

The solution is to create an object that sits above the cells and can be moved and sized independently of the natural rows and columns below. This area in Excel is known as the Drawing Layer. All graphics (i.e. shapes, charts, SmartArt, images, etc...) reside on the Drawing Layer. This allows the objects placed on the Drawing Layer to “float” above the row/column grid.

Think of the Drawing Layer as a sheet of glass that sits atop the spreadsheet. There may be content in the cells beneath the glass, but the objects painted on the glass can be moved. And because we are working with “magic glass”, the objects painted on the glass can be resized.

If the underlying row(s) or column(s) are adjusted in height or width, the Drawing Layer objects remain unaffected by the modification.

The Camera tool is not located on any ribbon and must be manually added to either the Quick Access Toolbar (QAT) or placed on a custom ribbon. To add the Camera tool to the QAT, perform the following steps:

1. Select **File** → **Options** to open the **Excel Options** dialog box.
2. In the **Excel Options** dialog box, select the **Quick Access Toolbar** category on the left.
3. From the “**Choose commands from**” dropdown list, select **Camera**.
4. Click the **Add** button in the middle of the Excel Options dialog box to move the Camera tool from the **left panel** to the **right panel**. Click **OK** to close the Excel Options dialog box.

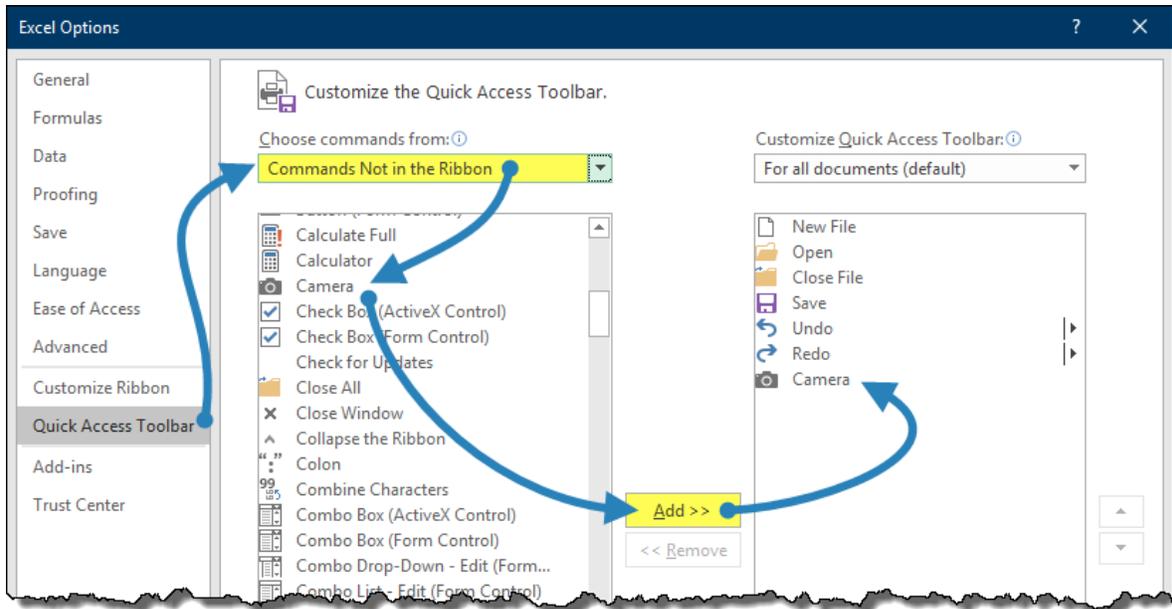


Figure 17.1 - Adding Excel's Camera Tool to the Quick Access Toolbar

To use the Camera tool, and to become acquainted with its operation and features, perform the following test:

5. Select a range of cells (i.e. **Calculation** sheet, **F4:G9**).
6. Click the **Camera** tool on the **QAT**.
7. Click **anywhere** on the **Calculation** sheet.

You can tell that the newly created object is a picture because the Picture Tools ribbon has presented itself for use.

The best characteristic of the picture is that it is a LIVE link to the originally “photographed” cells. The picture behaves more like a security camera than a picture. If you alter the original cells feeding the picture, the picture updates automatically.

We want to use the Camera tool to bring the original Pivot Table data from the Calculation sheet to the Dashboard sheet.

8. **Remove** the test **camera results** from the previous steps.
9. On the **Dashboard** sheet, select anywhere in the pivot table (lower-right) and move the Pivot Table from the **Dashboard** sheet to the **Calculation** sheet in cell **A16**.
10. Remove the temporary message from cell **A14**.
11. Select cells **A16:E21** and click the **Camera** tool on the QAT.
12. Click anywhere in **Column “E”** below the Pivot Table.

13. **Remove** the **border** from the new picture.
14. To apply a light white background color to the Pivot Table, select the original Pivot Table and then select **Pivot Table Tools → Design → Pivot Table Styles →** right-click on the **Custom** design style and select **Modify**.
15. In the Modify Pivot Table Style dialog box, select **“Whole Table”** and click **Format**. Set the **Fill** color to a **custom color format** with the RGB codes as follows:

R: 253, G: 253, B: 253

Notice that the Pivot Table and the live picture have been updated using the new color.

16. **Move** the Pivot Table picture to the **Dashboard** sheet.

Notice that the image is incorrect. This is because the image did not retain the original cell reference as it pertained to the Calculation sheet. The reference behaves like a relative reference, pointing to cells A16:E21 on the Dashboard sheet. We want the reference to behave like an absolute reference and remain pointing to the original cells on the Calculation sheet.

NOTE: If we had used a Named Range to hold the sheet and cell reference, and used the Named Range in the Camera tool, we would not have need to alter the reference when moving the picture from one sheet to another sheet.

17. Update the formula for the picture to the following version:

=Calculation!\$A\$16:\$A\$21

18. Fill in the visual “hole” on the Dashboard sheet by selecting any empty cell (i.e. **J20**) in the report area and selecting **Format Painter**. Select cells **J20:N25** to return the cells to a uniformed look and feel as the rest of the report.

19. Relocate the Camera picture above where the “hole” appeared.

We do not want to show the “Current” and “PY” titles, so we will crop that part of the picture.

20. Select the picture and then select **Picture Tools → Format → Size (group) → Crop**.

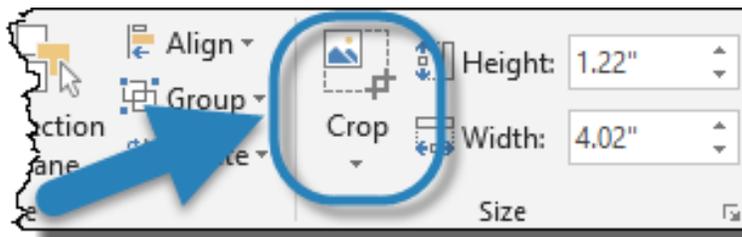


Figure 17.2 - Picture Tools - Crop Tool

- Grab the **12 o'clock crop handle** and pull in a downward direction until the “Current” and “PY” titles are no longer visible.

	Current		PY
	Revenue	% Δ PY	Revenue
Game	43,784	119.0% ▲	19,989
Productivity	29,276	-8.4% ▼	31,955
Utility	46,994	19.6% ▲	39,295
Total	120,054	31.6% ▲	91,239

Figure 17.3 - Crop Tool Handle

- Click the **Crop** button to **deactivate** the Crop tool.

NOTE: Granted, we could have simply omitted the selecting of row 16 on the calculation sheet when first creating the picture with the Camera tool. This step is to demonstrate that Camera pictures can be cropped modified like any other picture.

We want the Pivot Table picture to have the same rounded “app” look of the bar chart above.

- Insert a **Flowchart: Terminator** shape  and place it over the picture of the Pivot Table.

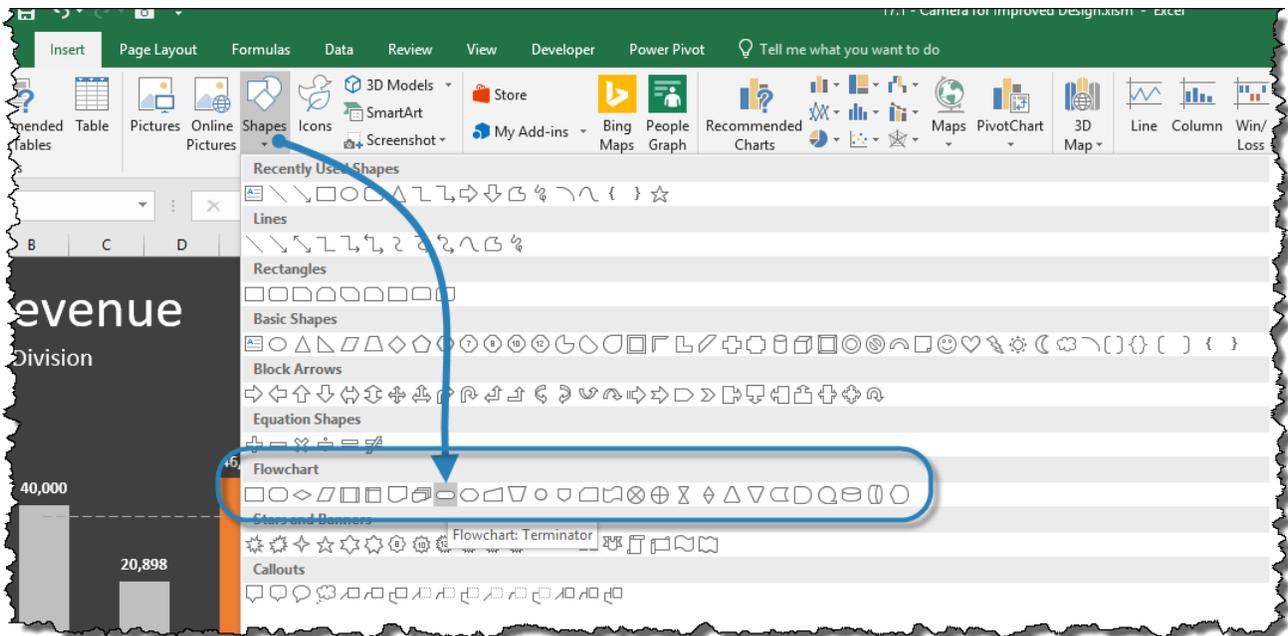


Figure 17.4 - Terminator Shape in Flowchart Category of Shapes Library

- Move the shape behind the Pivot Table picture by right-clicking the Terminator shape and selecting “**Move to Back**”.
- Remove the border** of the **Pivot Table** picture (*if necessary*).

26. Set the **Fill** color of the **Terminator** shape to the same **off-white** as the Pivot Table fill color (RGB: 253, 253, 253).
27. **Remove** the **border** of the **Terminator** shape (*if necessary*).

We will now add some additional titles to the Pivot Table picture.

28. Insert a **textbox** with the word “**Actual**” inside.
29. Set the **font size** to **14** points and set the **font color** to **orange**.
30. Place the textbox above and between the titles “**Revenue**” (*left column*) and “**%ΔPY**”.
31. **Copy** the textbox and paste it above the “**Revenue**” title (*right column*). Change the text in the second textbox from “Actual” to “PY”.
32. Change the **font color** to **25% black** (3rd down, 2nd column).
33. Select all the Pivot Table objects and **group** them together (*right-click → Group*).
34. Test the picture by changing the values on the Data sheet. This should update the Pivot Table which will in-turn update the Camera picture on the Dashboard sheet.

Things to be aware of when using Pictures from the Camera tool:

- Because pictures take up more space than data in cells, the file’s size may become large when using many pictures in a single file.
- Performance may suffer when using 3rd Party add-ins.

IMPORTANT: *Remember to always thoroughly test your solution prior to client distribution.*

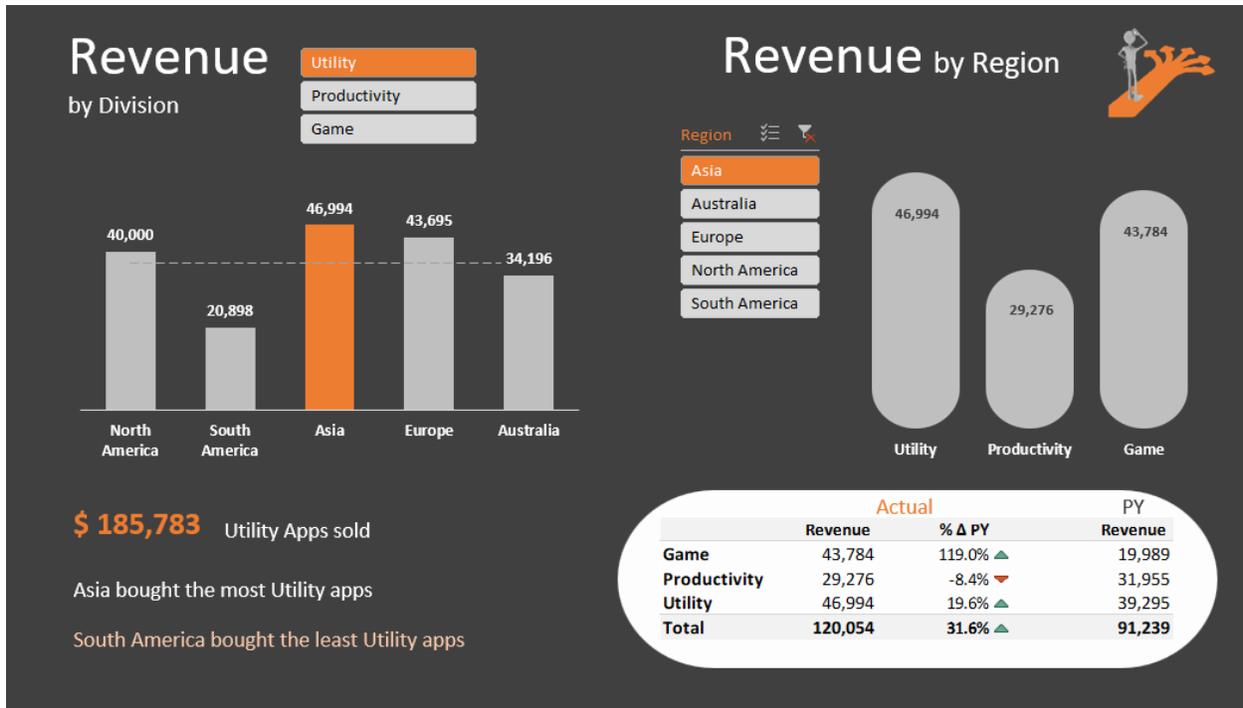


Figure 17.5 - Output of Module 17.1

MODULE 17.2 – LOOKUPS ON PICTURES

Data File:

17 – Secrets for Impressive Reports\17.2 – Advanced Camera – Lookups on Pictures.xlsm

In the previous module, we saw how the Camera tool could be used to display other areas of the spreadsheet. In this module, we will take the Camera tool a step further to perform a conditional lookup to return a picture based on user selection.

Our objective; when the user selects a Division name from the slicer, the logo for the selected Division will appear to the right of the slicer.

We have the three Division’s logos staged on the Calculation sheet.

If we try to select the logo (such as the green calculator) and click the Camera tool, when we attempt to “drop” the picture in place we are presented with a “No link to paste” message.



Figure 17.6 - Message from Camera Tool When Attempting to Photograph an Image Object

The Camera tool cannot photograph object, only cells. We need to highlight the cells directly behind the object and photograph the cells instead. If we place the image in front of the photographed cells, the image will be displayed in the photograph. Think of it as an “Excel Photobomb”.

We will adjust the height and width of the “background” cells to closely resemble the height and width of the image to be photographed.

1. Expand the height of **rows 26 through 28** to **75** pixels. You can resize the rows manually, using the size guide as a reference, or you can select **Home → Cells (group) → Format → Row Height** and define the row height explicitly.
2. Move each image in front of their respective cells (i.e. **B26** – green calculator, **B27** – blue wrench, **B28** – purple controller.)

TIP: If you hold down the ALT key when moving objects, the objects will “snap” into the corners of the cells. This also works when resizing objects. This will allow you to make the images the exact same height and width of the cells they are above.

3. Select cell **B26**. (If your image is the exact same size as the underlying cell, you may need to select an adjacent cell and then use the arrow keys to move the cursor to the cell behind the image.)

4. Click the **Camera** tool on the QAT.
5. Click anywhere on the **Calculation** sheet to create your photograph of cell **B26**.



Figure 17.7 - Results of Photographed Cell

6. **Remove** the **border** from the photographed image.

We want the photograph to have the same dark background as the Dashboard; we don't want the fill of the original cell color acting as a border of the photograph.

7. On the **Dashboard** sheet, select an empty cell (i.e. **J16**) and click **Copy** (or CTRL-C).
8. Return to the **Calculation** sheet and select cells **B26:B28** and click **Paste** (or CTRL-V).

We want the image to change when the user selects a Division button from the Division slicer. The problem is the picture does not allow embedded formulas; it only allows direct cell references or named ranges from the Name Manager.

To overcome this limitation, we will assign names to cells B26 through B28 and use the Name Manager to retrieve the images and send them to the Dashboard sheet.

9. On the **Calculation** sheet, select cells **K17:K19** and click **Copy** (or CTRL-C).
10. Select cell **A26** and click **Paste** (or CTRL-V).
11. Select cells **A26:28** and select **Formulas** → **Define Names (group)** → **Create from Selection**.
12. Check the boxes for **"Top row"** and **"Left column"** and click **OK**.
13. Select the photograph and reprogram the formula in the Formula Bar to read as follows:

=Game

Although this is working, it is still static in its logic. We need to make the logic dynamic. We can achieve this dynamic behavior by using the INDIRECT function in the Name Manager.

[For a review on the purpose and use of the INDIRECT function, see Module 5 for details.](#)

14. Open the **Name Manager**.
15. Click the **New** button.
16. Fill in the **New Name** dialog box with the following parameters:

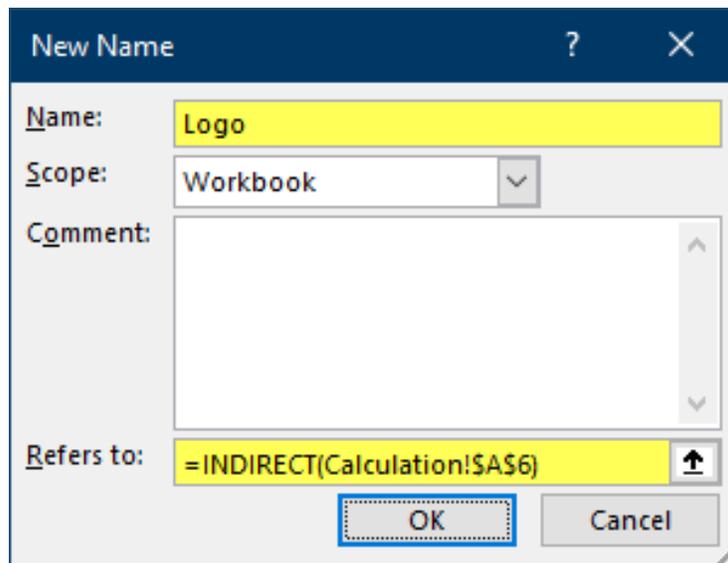


Figure 17.8 - New Name Dialog Box

```
=INDIRECT(Calculation!$A$6)
```

17. Click **OK**.
18. Select the photograph and update the formula in the Formula Bar with the following:

```
=Logo
```

19. Select the **photograph** and click **Cut** (or CTRL-X).
20. Switch to the **Dashboard** sheet and click **Paste** (or CTRL-V).

Although we removed the borders of the picture, we can still see the original cell borders. To solve this problem, crop the edges of the pasted photograph until the original cell borders are no longer visible.

21. Size and position the photograph as necessary.
22. Test the functionality of the photograph by selecting different Divisions from the Division slicer.

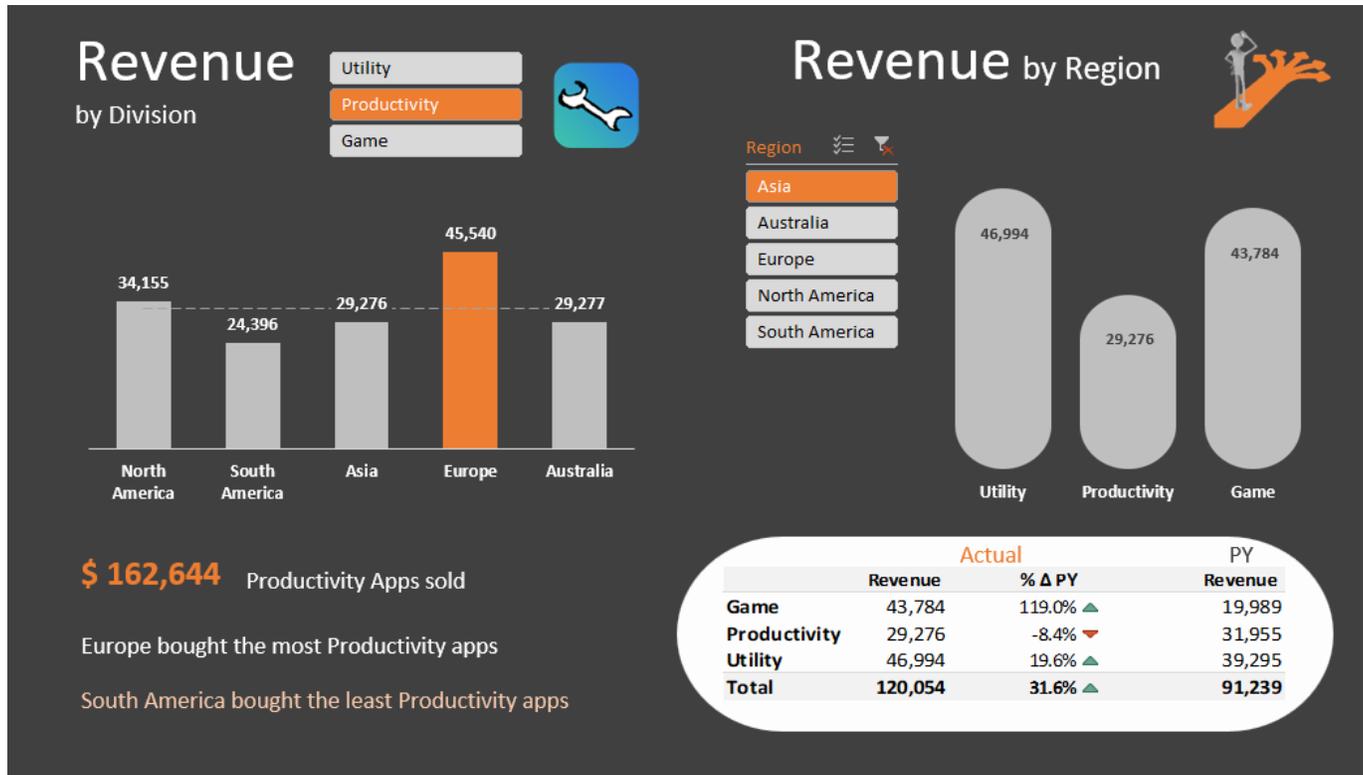


Figure 17.9 - Output of Module 17.2

MODULE 17.3 – FINAL TOUCHES

Data File:

17 – Secrets for Impressive Reports\17.3 – Final Touches.xlsm

This module will focus on adding some finishing touches to the Dashboard sheet.

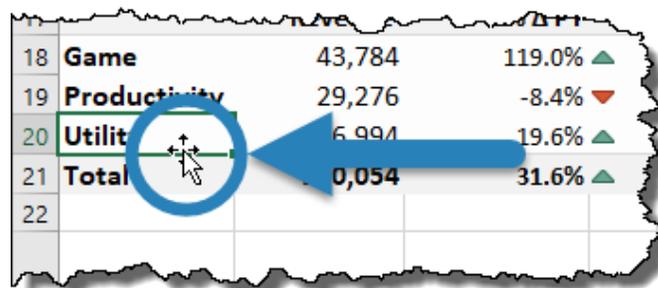
Apart from double-checking the formulas and writing instructions for the Dashboard, we also need to check the “flow” of the Dashboard.

Order of Lists

Example: The order of items in the Division slicer reads “Utility – Productivity – Game”. The items along the horizontal axis (x-axis) of the Revenue by Region chart reads in the same order. However, the Divisions listed along the left side of the Pivot Table (*lower right of Dashboard*) reads “Game – Productivity – Utility”. We need to reorder the list to provide a uniformed order to all Division listings on the Dashboard.

1. On the **Calculation** sheet, select cell **A20** (“Utility”).
2. Place your cursor on the edge of the selection border and **click-and-hold**. Move the “Utility” field to the **top** of the Pivot Table Division list.

(Make sure you see the Move Cell pointer  before you click to ensure a move operation is performed.)



18	Game	43,784	119.0% ▲
19	Productivity	29,276	-8.4% ▼
20	Utility	6,991	19.6% ▲
21	Total	0,054	31.6% ▲
22			

Figure 17.10 - Move Cell Pointer

3. Select cell **A19** (“Game”).
4. Place your cursor on the edge of the selection border and **click-and-hold**. Move the “Game” field to the **bottom** of the Pivot Table Division list.

The Pivot Table’s Divisions order should now be in the same order as the two charts on the Dashboard sheet.

Now we want the items listed in the Region slicer to be in the same order as the horizontal axis (x-axis) of the “Revenue by Division” chart. The slicer is sorted in ascending order. Neither this order, nor descending order, will provide the sought-after results. We will need to create a custom sort order for the slicer to display the Regions in the desired order. Custom sorts are created and managed within the Excel Options portion of the application.

5. Select **File** → **Options** → **Advanced** → **General (category)** → **Edit Custom Lists**.

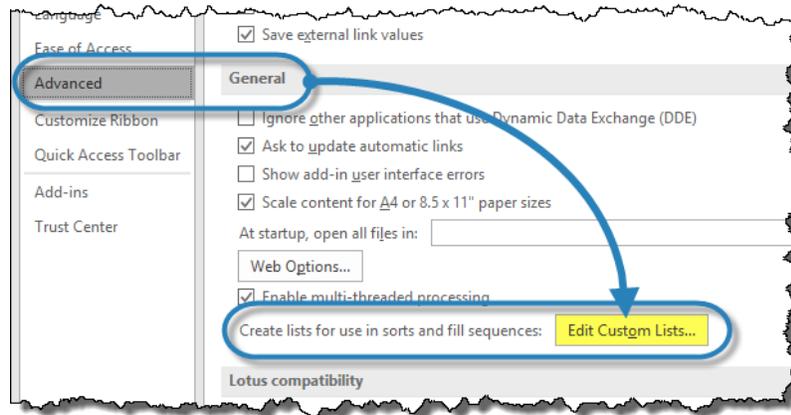


Figure 17.11 - Creating Custom Lists Feature Location

6. Select the “**Import list from cells:**” field and highlight the following location:

=Calculation!\$F\$5:\$F\$9

7. Click **OK** twice to exit the Excel Options dialog box.
8. Select the **Region** slicer and click **Slicer Tools** → **Option** → **Slicer (group)** → **Slicer Settings**.
9. In the Slicer Settings dialog box, check the option for “**Use Custom Lists when sorting**” and click **OK**.

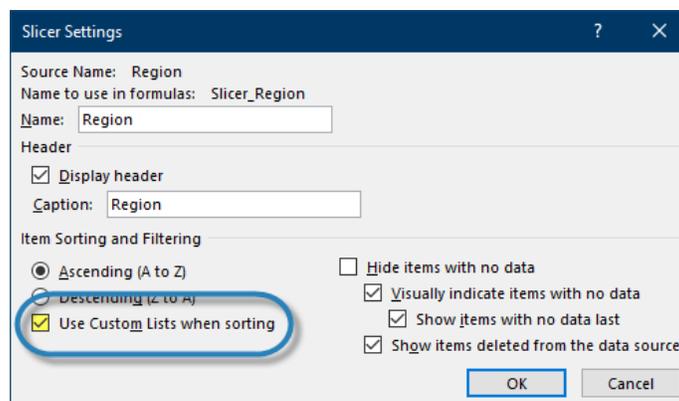


Figure 17.12 - Slicer Settings Dialog Box

NOTE: It may be necessary to **uncheck-OK/recheck-OK** the “Use Custom List when sorting” feature to get the slicer to recognize the newly created custom list.

Lock-down objects to prevent inadvertent (or deliberate) modifications

We need to lock objects like titles, slicers, textboxes, etc... to prevent the user from accidentally (or purposefully) moving, resizing, or deleting the objects.

10. Select **Review** → **Changes (group)** → **Protect Sheet**. Enter your desired password (*twice*) and click **OK**.

The sheet is protected, but the slicers no longer function. We need to exempt the slicers from the protection sheet system.

11. Remove the sheet protection by selecting **Review** → **Changes (group)** → **Unprotect Sheet**.
12. Right-click the **Region** slicer and select “**Size and Properties**”.
13. In the **Format Slicer** panel on the right, expand the **Properties** category and uncheck the “**Locked**” option.

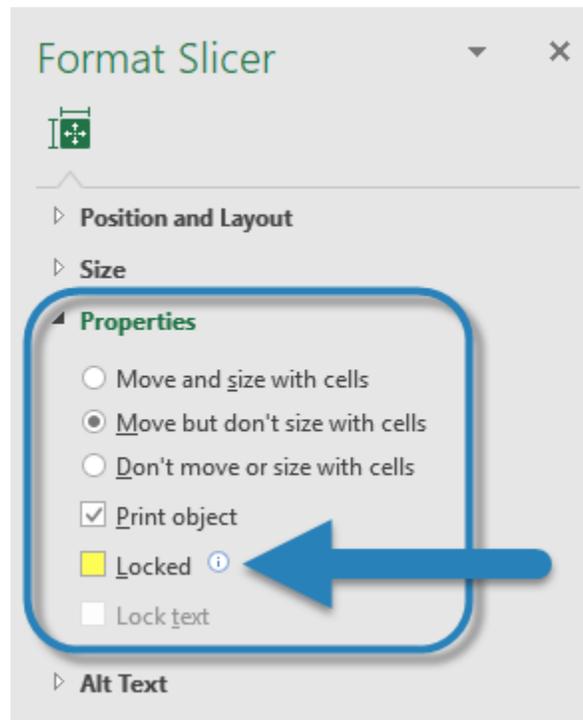


Figure 17.13 - Format Slicer Settings

14. Repeat Steps 12 & 13 for the Division slicer.
15. Protect the sheet (see Step 10) and observe that the slicers are now functional.

NOTE: The slicers can still be moved, resized and even deleted (unfortunately). Until Microsoft builds in an exemption for slicers within the sheet protection system, this is an issue we will have to live with.

Hide sheets not needed by the user

It is a good idea to hide any sheets the user may not require for operation use of the dashboard. You could right-click on the desired sheet's tab and select Hide, but anyone with a mild knowledge of Excel could right-click tab and select Unhide to bring the sheet back into view. Some users likely discover this ability and attempt to exploit it as a means of digging into the deeper workings of the workbook.

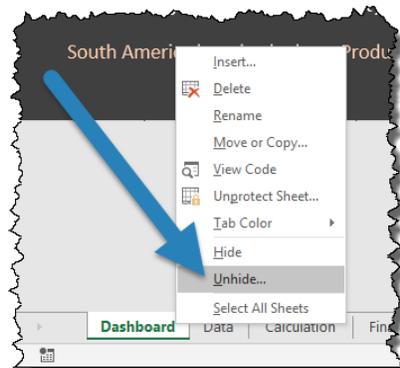


Figure 17.14 - Sheet Tab Right-Click Menu

Our solution, although not foolproof, is much more difficult to discover accidentally and is likely to remain a mystery to most users.

16. Right-click on the **Calculation** sheet and select **View Code**. This will take you into the Visual Basic Editor.
17. In the Visual Basic Editor, select the sheet you wish to hide from the list at the top-left. Now, select the dropdown for the option labeled “Visible” and select “2 – xlSheetVeryHidden”.
18. **Close** the Visual Basic Editor.

Your sheet is now hidden, and if the user right-clicks on a sheet tab, the option for “Unhide” will not work for this sheet.

If you wish to restore the visibility of the “very hidden” sheet, right-click on any sheet tab and select “View Code”. Repeat **Step 17** but change the “Visible” option to “-1 – xlSheetVisible”.

Hide headings and gridlines

Another nice touch is to hide the row and column headings. Since the user will not be building formulas, these are unneeded and take up valuable screen real estate.

19. Select **View** → **Show (group)** → uncheck **Headings**.

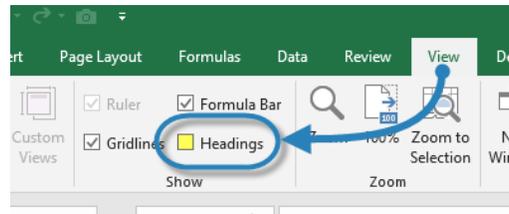


Figure 17.15 - Remove Headings Feature on View Ribbon

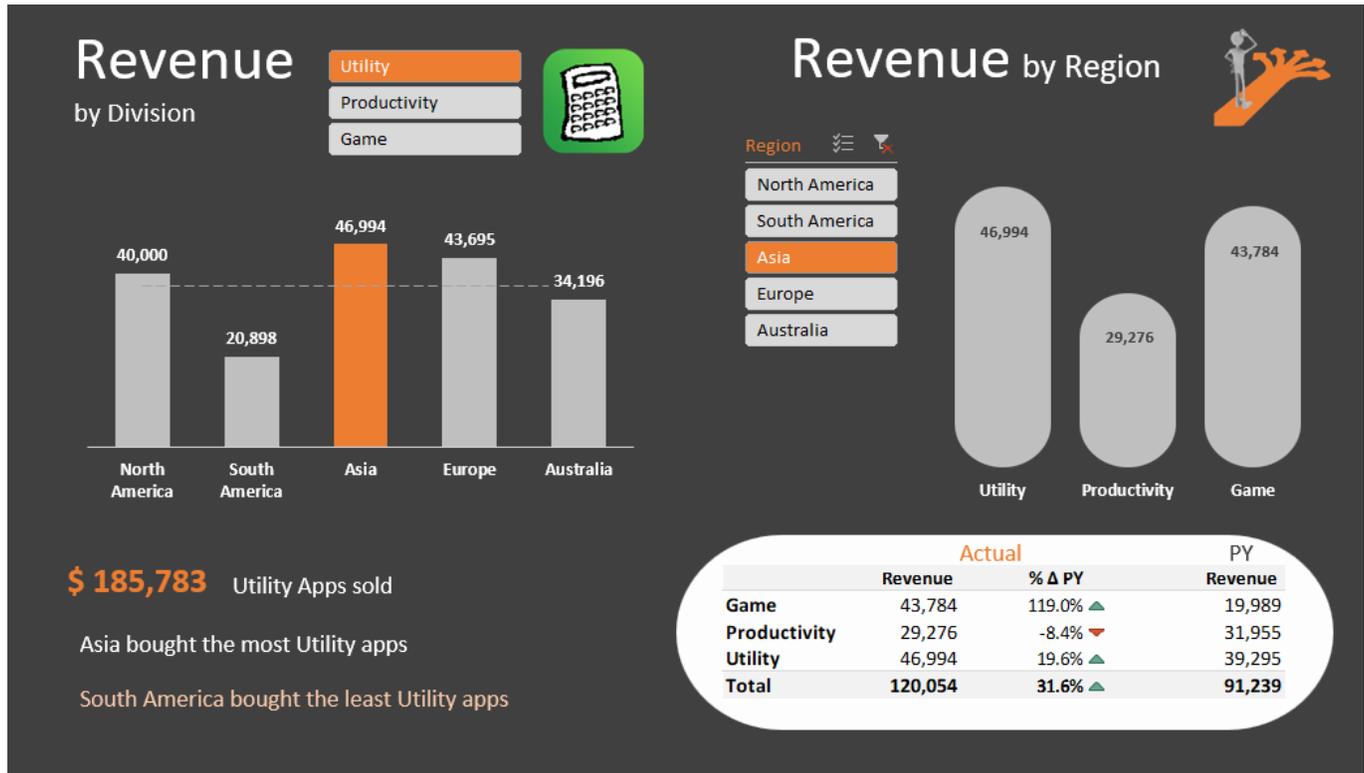


Figure 17.16 - Final Output of Slicer Dashboard Project

Congratulations!!! You have just created your Slicer Dashboard.

As with the KPI dashboard, we have climbed an insurmountable mountain by taking one step at a time.

The skills obtained in the creation of these dashboards will serve you well in all your future Excel projects. You may not create an Excel workbook with every single feature used simultaneously, but odds are strong that every workbook you create will utilize at least one if not many of these techniques and approaches to solving problems.

Free Excel Tutorials

If you'd still like to learn more Excel tips & techniques, check out my Excel Tutorials here:

<https://www.xelplus.com/tutorials>

And if you're someone like me, who likes to have everything in one file, download the workbook at the bottom of this post: <https://www.xelplus.com/free-excel-tutorials/>



You'll find the list as well as the links to all the Excel tutorials I have so far. I'll be updating the workbook on a monthly basis.

More Resources

When I set out to learn something new, I don't just buy one book or one course, I learn from a variety of sources. Each person has their unique view and way of presenting. Each one inspires me in a different way. If you're serious about learning Excel, check out my list of favorite books and courses.

You'll find resources on data visualization, Power Query, Power BI and much more:

<https://xelplus.com/resources>