

FALL 2019, OCT. 10-12  
SPOKANE, WA

**WASWUG**  
BE INCREDIBLE

PRESENTED BY



# Advanced Excel

## Automate with Macros

- Record and Edit Macros
- Play Macros with Shortcuts and Buttons
- Write Macros in the Visual Basic Editor
  - Message Boxes, Loops, and Triggers

## Advanced Formatting

- Create Complex Forms
- Advanced Data Validation
- Active X Controls – Checkboxes, Form Fields, Calendars, Buttons

## Perform Advanced Functions – Logical, Lookup, Text, Date and Time, Text, Nested

- Absolute Reference – Naming Cells and Ranges
- Formula Auditing

## Creating Data Models

### Pivot Tables and Pivot Charts

- Group Data
- Calculated Fields
- Advanced Analysis and Reporting

## Conditional Formatting

### Web Queries

### Custom Views

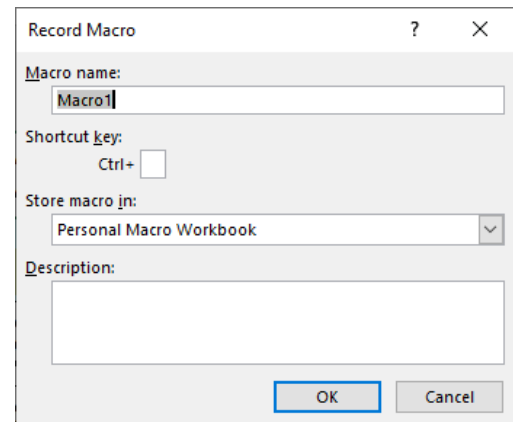
# Macros

A macro is a recording of keystrokes and mouse clicks that can be played back.

Any set of tasks performed exactly the same way every time is a good candidate for a macro

Create a Macro:

1. On the View tab, in the Macros group, click the Macros dropdown arrow and select Record Macro
2. Name the macro
3. Assign a shortcut key (optional)
3. Choose where the macro should be stored (the workbook where the macro is stored must be open for the macro to run)
  - a. Personal Macro Workbook (always available when you are logged in)
  - b. New Workbook
  - c. This Workbook (travels with the workbook)
4. Provide description (optional)
5. Click OK
6. Perform the commands you want to record
7. Go back to the Macros group on the View tab, and choose Stop Recording from the drop-down menu



Run a Macro:

1. Select the View tab
2. Click the Macros button above the drop-down arrow
3. Select the macro you want from the Macro name list
4. Click Run

Edit a Macro:

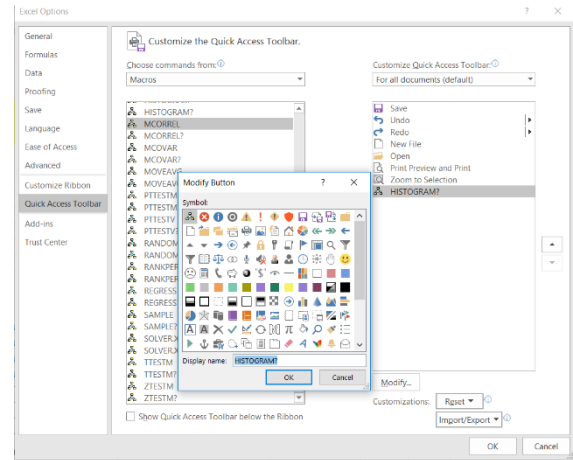
1. Go to the View tab
2. Click the Macros drop-down
3. Select View Macros
4. Select the macro you want to edit (if the Macro is stored in the Personal Macro Workbook, you will need to unhide it from the Window group of the View tab)
5. Click Edit
6. Make text edits to the code
7. Close, and Return to Microsoft Excel

Step Through the Code – The Step Into feature allows you to watch the macro perform a line of code at a time so you can “see” which line is causing you trouble.

1. Open the VBA editor and click in the macro you wish to review. Make sure to also have your Excel window open and visible.
2. Select Debug > Step Into or F8 to initiate.
3. A yellow highlight will indicate which line Excel is executing. Continue hitting F8 until the program is complete.

## Add Macros to the QAT:

1. Click the Office button
2. Click the Excel Options button
3. In the Customize panel, select Macros from the Choose command from dropdown menu
4. In the Customize Quick Access Toolbar, select For all documents
5. In the left pane, select the macro to be added to the QAT
6. Click the Add button
7. Click the Modify button beneath the right pane
8. Type a name for the shortcut in the Display name field
9. Select an image for the shortcut in the Symbol pane
10. Click OK
11. Click OK in the Excel Options window.
12. Review the QAT to see the macro's shortcut with the selected icon button

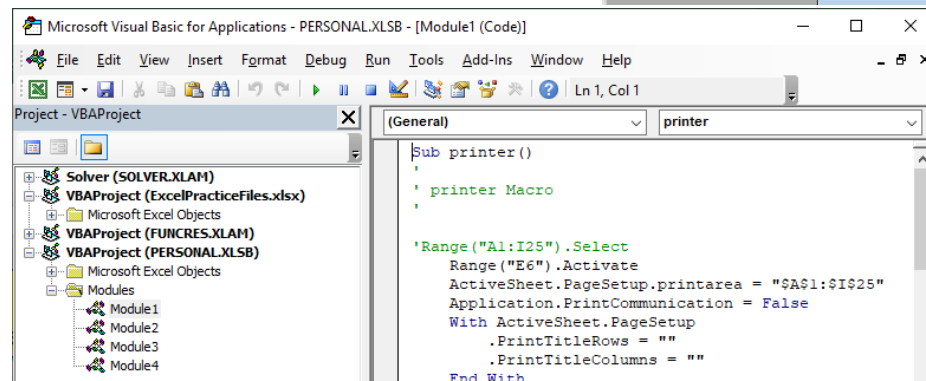
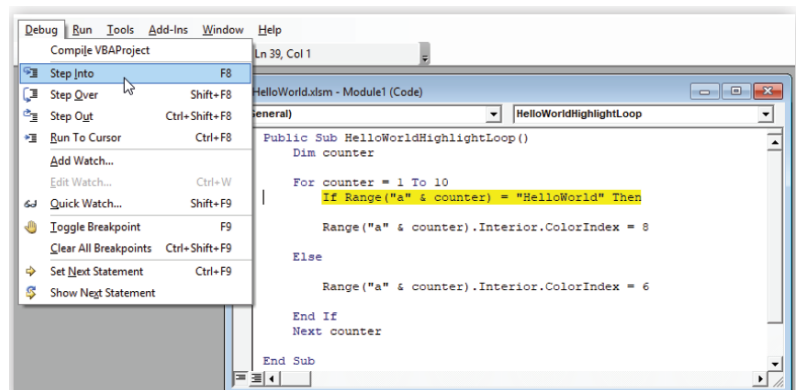


## Add Macro Buttons to a Worksheet:

1. On the Developer Tab in the Controls group, click the Insert drop-down arrow, and choose Button
2. Drag the button on the Developer Tab to the place you want it on the worksheet, and size it
4. Select a macro to assign to the button (this option pops up when you release the mouse button)
5. Click OK
6. Rename the button

## Write or Edit in the VBA Editor

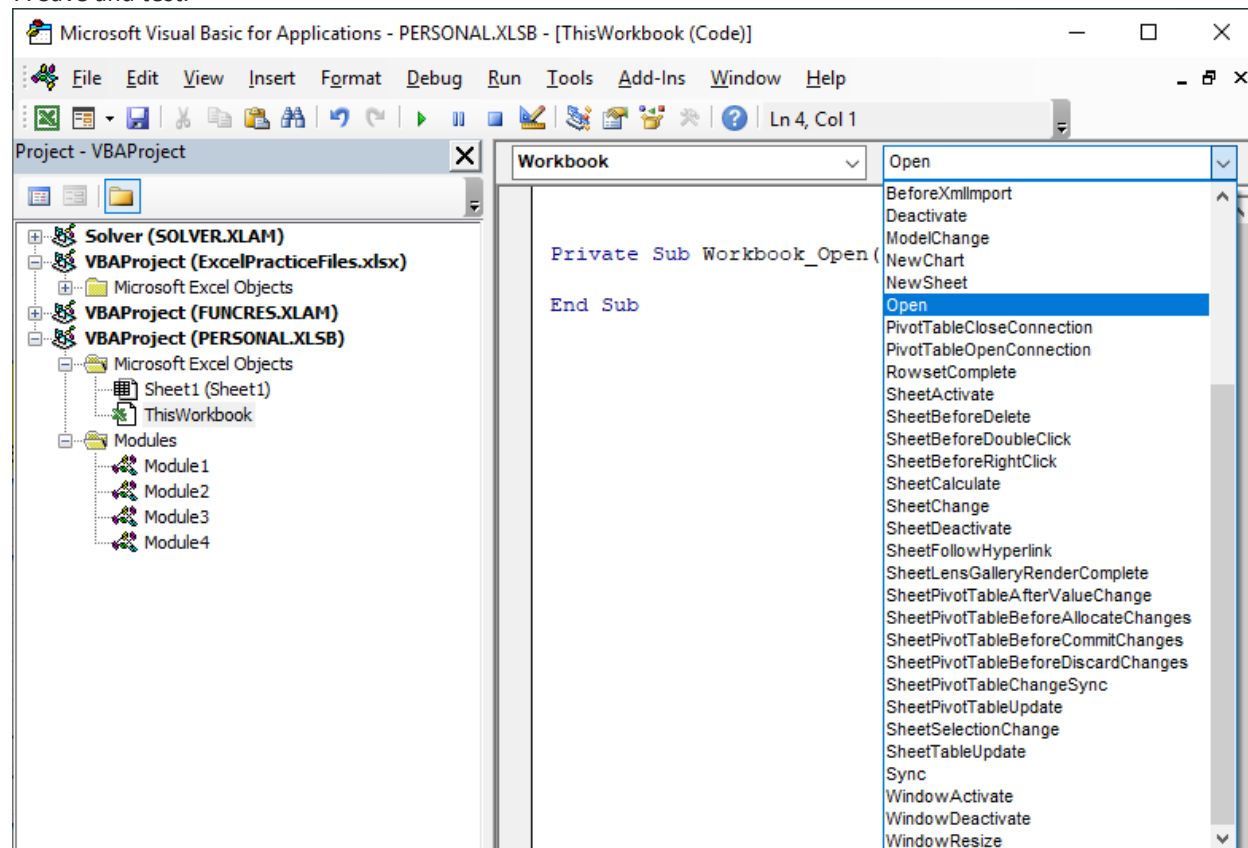
1. Press ALT + F11 to open the VBA editor
2. Select the project in the Project Window that you want (Most macros are stored in modules).
3. Place your cursor in the code module between the Sub and End Sub declarations
4. Type in the macro procedures



## Trigger a Macro from an Event

You may want a macro to run based on some action you have taken. You might want a message box to appear when a user opens a worksheet with information about what is included. Or, you may want to run a macro that sorts your data into your preferred format each time the workbook is saved.

1. Record or plan your Macro.
2. Open VBA Editor.
3. Double-click the module that will store the macro in the Project Window. Event Procedures are not stored in Code Modules like the macros seen so far. Instead, they will be stored in one of two places:
  - a. Worksheet module – Macros will run when events happen in or to a specific worksheet.
  - b. ThisWorkbook – Macros will run when events happen in or to the workbook.
4. Change the (General) dropdown menu to Worksheet or Workbook. [A]
5. Choose the action that will trigger the macro in the worksheet. [B] Excel will create open and close Sub statements. [C] Note that Excel creates a Private Sub procedure for this type of macro. This simply means that the macro will not be seen in Excel's Macro dialog box – it can only be run when the event, or trigger, is detected (or from the VBA editor).
6. Type or copy/paste your macro between the Sub statements as usual. [D]
7. Save and test.

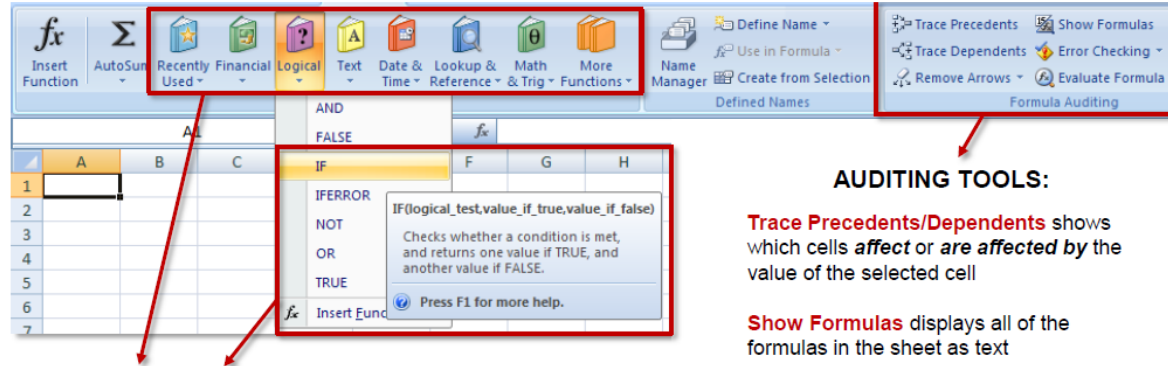


### Add a message box to an existing macro. Syntax: MsgBox ([TEXT], [BUTTONS], [TITLE])

Message boxes are 'pop up' messages that grab the user's attention and require a response before they continue.

```
Sub Msg_exe()  
MsgBox "This is a message!", vbExclamation, "This is the Title"  
End Sub
```

## Formulas and Functions



### FORMULA LIBRARY:

Includes a list of all common formulas, component parts, and brief descriptions of how each formula works

### AUDITING TOOLS:

**Trace Precedents/Dependents** shows which cells *affect* or *are affected by* the value of the selected cell

**Show Formulas** displays all of the formulas in the sheet as text

**Evaluate Formula** allows you to step into a formula and determine the output of each component

## Name Cells and Ranges

1. Select the cell(s) to be included in the named range
2. Click the Name Box
3. Type a name for the range
4. Press Enter

The cell or range can still be referred to by the row and column reference, but it can also be referred to by its name

## Text Functions

**PROPER** Capitalizes the first letter in each text string

**UPPER** Changes text to all caps      **LOWER** Changes text to all lower case

**CONCATENATE** Combines several text strings into one string

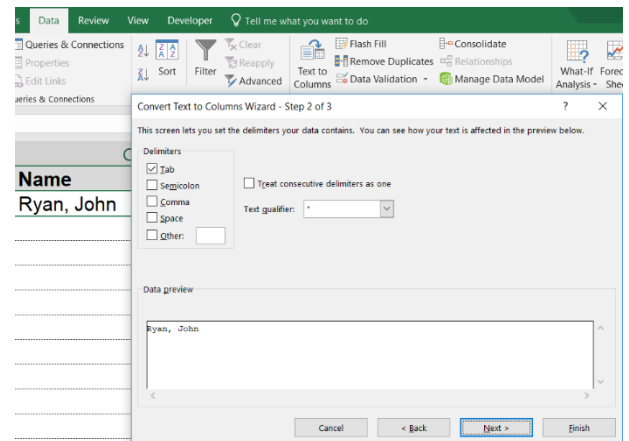
**TRIM** Removes any leading or trailing spaces from a text string

**LEFT RIGHT MID** Return a specific number of characters from a location within a text string

	A	B	C
1	First Name	Last Name	
2	alan	alda	
3	john	cleese	
4	cate	blanchett	
5	ellen	degeneres	
6	beyonce	knowles	
7	will	smith	
8	justin	beiber	
9	dennis	miller	
10	oprah	winfrey	

## Convert Text to Columns:

1. Select the column to be split
2. On the Data tab, in the Data Tools group, click the Text to Columns button
3. Select the Delimited radio button
4. Click the Next button
5. Select the Delimiter(s)
6. Click the Finish button
7. Review your split columns



**RAND() and RANDBETWEEN** act like random number generators in Excel:

	A	B	C	D	E
1	0.5173	0.4091	0.7560	0.9012	0.2167
2	0.0906	0.2317	0.0906	0.5856	0.8646
3	0.1544	0.8240	0.4279	0.8782	0.7795
4	0.0097	0.0872	0.7740	0.9137	0.7815
5	0.2089	0.7028	0.0449	0.8173	0.9983
6	0.0761	0.4388	0.4056	0.5639	0.0668

The **RAND()** function returns a random value between 0 and 1 (to 15 digits)

The **RANDBETWEEN** function returns an integer between two values that you specify

	A	B	C	D	E
1	83	23	64	62	92
2	59	45	40	50	91
3	24	37	70	30	32
4	54	85	69	55	3
5	73	12	36	53	2
6	29	72	68	59	99

=RANDBETWEEN(0,100)

## Absolute Reference

If you do not want the reference to a cell to change when you copy it to a new location, you can lock in the reference by making it absolute. The way to do this is to put a dollar sign in front of the reference you want to lock in. If you are referring to a single cell and do not want the reference to change no matter what direction it moves, lock it in by putting a dollar sign before BOTH the column and the row reference \$C\$4

You can also lock in just the row C\$4 or just the column \$C4

You can type the dollar signs or use F4 to toggle through the 4 types of reference: \$C\$4 C\$4 \$C4 C4

## The IF Function =IF (Logical Test, Value if True, Value if False)

The IF function is the function that allows many other functions to work correctly. How many times do you have everything set, then you realize there is a caveat or exception?



By using **Nested IF Statements**, you can include multiple logical tests within a single formula:

	A	B	C	D	E
1	Location	Temp (F)	Precip (mm)	Freeze	Climate
2	A	75	0	No	Mild
3	B	18	0	Yes	Cold
4	C	86	0	No	Hot
5	D	80	2.3	No	Mild
6	E	28	1.2	Yes	Cold
7	F	68	0.5	No	Mild
8	G	26	0	Yes	Cold

= IF(B2<40,"COLD",IF(B2>80,"HOT","MILD"))

*If temp<40, climate = "Cold", if temp>80, climate = "Hot", otherwise climate = "Mild"*

	A	B	C	D	E	F
1			<b>Class A</b>			
2			<b>Test#1</b>	<b>Test#2</b>	<b>Test#3</b>	<b>Grade</b>
3	<b>Grades</b>	Student 1	90	83	86	
4		Student 2	88	88	85	
5		Student 3	90	85	82	
6		Student 4	77	78	79	
7		Student 5	60	55	55	
8						
9	Average the test scores for Class A and then post the grade in Column F					

Excel's **AND** and **OR** statements allow you to include multiple logical tests at once:

	A	B	C	D	E	F	G
1	Location	Temp (F)	Precip (mm)	Freeze	Climate	Precip Type	Conditions
2	A	75	0	No	Mild	None	Dry
3	B	18	0	Yes	Cold	None	Dry
4	C	86	0	No	Hot	None	Dry
5	D	80	2.3	No	Mild	Rain	Wet
6	E	28	1.2	Yes	Cold	Snow	Wet
7	F	68	0.5	No	Mild	Rain	Wet
8	G	26	0	Yes	Cold	None	Dry

=IF(OR(F2="Rain",F2="Snow"),"Wet","Dry")

*Here we're categorizing conditions as "Wet" if the precipitation type equals "rain" OR "snow", otherwise Conditions = "Dry"*

=IF(AND(D2="Yes",C2>0),"Snow",IF(AND(D2="No",C2>0),"Rain","None"))

*If the temp is below freezing AND the amount of precipitation > 0, then Precip Type = "Snow", if the temp is above freezing AND the amount of precipitation > 0, then Precip Type = "Rain", otherwise Precip Type = "None"*

	A	B	C	D
1	<b>Time Sheet</b>			
2	<b>Name</b>	<b>Time In</b>	<b>Time Out</b>	<b>Hours Worked</b>
3	Ross			
4	Monica			
5	Phoebe			
6	Rachel			
7	Chandler			
8	Joey			

Using **Named Arrays** can simplify a lookup function if you use the same data array in multiple formulas

For example, if you name the array from A1:D6 "Apparel"...

Apparel				
	A	B	C	D
1	Product	Quantity	Product ID	Price
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27838	\$79.99

...you can write your vlookup formula in either of the following ways:

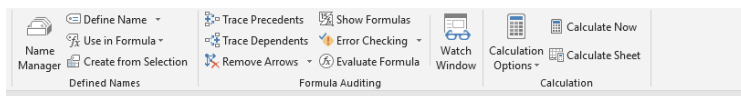
=VLOOKUP(A1,\$A\$1:\$D\$6,2)  
 =VLOOKUP(A1,Apparel,2)

## Formula Auditing

The Formula Auditing Tools on the Formulas tab can help you gain clarity on what is going on with the formulas in your spreadsheet.

Trace Precedents Shows the cells used in calculating the value of a cell. Note: Click the Trace Precedents button more than once to see all cells used.

Trace Dependents Shows all the formulas in which a specific cell is used. Use Remove Arrows to stop showing Precedence or Dependence Arrows.



C	D	E
<b>Tax Rate</b>	25%	
<b>Prices</b>	<b>Shipping</b>	<b>Total</b>
\$12.00	\$2.00	\$14.99
\$15.00	\$3.00	\$19.24
\$18.00	\$4.00	\$23.49
\$25.00	\$5.00	\$32.06



## VLOOKUP Function

The VLOOKUP function searches for a value in the leftmost column of a table and returns a value in the same row from a specified column in the table.

=VLOOKUP (WHAT, WHERE, WHICH)

What = What are you looking up based on?

Where = Where are you looking it up? (Highlight [and name] the table)

Which = Which column number (in your table) do you want to return data from?

An optional fourth argument allows you to input, "FALSE": if an exact match is required.

	A	B	C	D	E	F
1						
2	<b>Commission Rate</b>	10%				
3						
4	<b>Salesperson</b>	<b>Sales</b>	<b>Quota</b>	<b>Commission</b>	<b>Bonus</b>	<b>Total Pay</b>
5	Jim Bruce	\$6,598.00	\$3,000.00			
6	Mark Sanders	\$6,059.00	\$4,000.00			
7	Carmen Foster	\$4,895.00	\$5,000.00			
8						
9						
10	<b>Bonus Table</b>					
11	<b>Quota Level</b>	<b>Bonus</b>				
12	\$2,000.00	\$25.00				
13	\$3,000.00	\$50.00				
14	\$4,000.00	\$75.00				
15	\$5,000.00	\$100.00				
16	\$6,000.00	\$125.00				

Use **HLOOKUP** if your table array is transposed (variables headers listed in rows)

**=HLOOKUP(lookup\_value, table\_array, row\_index\_num, [range\_lookup])**

This is the **value** that you are trying to match in the table array

This is **where** you are looking for the lookup value

Which **column** contains the data you're looking for?

Are you trying to match the **exact** lookup value (0), or something similar (1)?

	A	B	C	D
1	<b>Product</b>	<b>Quantity</b>	<b>Product ID</b>	<b>Price</b>
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27838	\$79.99

**D2=HLOOKUP(A2, \$H\$1:\$L\$2, 2, 0)**

With an HLOOKUP, we search for the product name in F1:J2 and return the value from the 2<sup>nd</sup> row down

	G	H	I	J	K	L
Product		Shorts	T-shirt	Sweater	Spandex Unitard	Socks
Price		\$24.50	\$14.99	\$49.99	\$79.99	\$9.99

The **IFERROR** statement is an excellent tool to eliminate annoying error messages (#N/A, #DIV/0!, #REF!, etc.), which is particularly useful for front-end formatting

**=IFERROR(value, value\_if\_error)**

Formula or value (which may or may not result in an error)

Value returned in the case of an error

*In this case we're replacing an error caused by the A1/B1 formula with "Invalid Formula", and an error caused by a VLOOKUP function with "-"*

**=IFERROR(A1/B1,"Invalid Formula")**  
**=IFERROR(VLOOKUP(A1,D1:E4,2,0),"-")**

### Create a Custom Function

If Excel does not have exactly the function you need, you can create your own using VBA. Review the Macros section for details about opening and editing VBA code.

#### Create a Function Procedure

1. Press ALT+F11 to open the VBA editor
2. Select Insert > Module to insert a new Code Module
3. Select Insert > Procedure
4. Type a name for your function in the Name textbox
5. Select Function radio button in the Type pane
6. Click OK
7. Function and End Function declarations will be added to your code window.

Define the Function's Arguments - Type a short variable name for each argument the function will use to perform its calculations between the parenthesis that follow the function name.

Enter a Formula – Enter the formula that the function will perform using the variables defined in the arguments.

### Custom Views –

#### Print, Format, and Filter Settings

1. Click the View Tab
2. In the Workbook Views group, click Custom Views
3. Click Add
4. Give the view a name
5. Access the view from the same location

Add View ? X

Name:

Include in view

☒ Print settings

☒ Hidden rows, columns and filter settings

OK Cancel

	A	B	C
1	<b>Annual Review</b>		
2	<b>Cynthia Jones</b>		
3	<b>Goals</b>	<b>Yes/No</b>	<b>Yearly Quota Met:</b>
4	Completed Training	No	
5	Customer Satisfaction	Yes	<b>Yearly Goals Met:</b>
6	Displayed Flexibility	Yes	
7	Worked Well with Others	Yes	<b>Bonus:</b>
8	Kept Funnel Full	Yes	(Must meet yearly quota or yearly goals)
9	Attended Sales Meetings	Yes	<b>Achieved Champions Club:</b>
10			(Must meet yearly quota & yearly goals)
11			
12	<b>Quota Required:</b>	<b>\$100,000.00</b>	
13	<b>Sales Achieved:</b>	<b>\$132,452.00</b>	

The **INDEX** function returns the *value* of a specific cell within an array

**=INDEX(array, row\_num, column\_num)**

What range of cells are you looking at?

How many rows down is the value you want?

How many columns over is the value you want?

	A	B	C
1	Tools	Price	Inventory
2	Hammer	\$5.00	55
3	Screw Driver	\$2.50	66
4	Pliers	\$3.34	333
5	Wrench set	\$10.00	234
6	Chain Saw	\$55.48	23
7	Tool Box	\$19.99	5
8	Level	\$2.25	7

**INDEX(\$A\$1:\$C\$5, 5, 3) = 234**

In this case we're telling Excel to find the value of a cell somewhere within the array of A1:C5. Starting from the upper left, we move down to the **5<sup>th</sup> row** and right to the **3<sup>rd</sup> column**, to return the value of **234**

The **MATCH** function returns the *position* of a specific value within a column or row

**=MATCH(lookup\_value, lookup\_array, [match\_type])**

What value are you trying to find the position of?

In which row or column are you looking? (must be a 1-dimensional array)

Are you looking for the exact value (0), or anything close?

1: Find largest value < or = lookup\_value

0: Find exact lookup\_value

-1: Find smallest value > or = lookup\_value

	A	B
1	Tools	Price
2	Hammer	\$5.00
3	Screw Driver	\$2.50
4	Pliers	\$3.34
5	Wrench set	\$10.00

**MATCH("Pliers", \$A\$1:\$A\$5, 0) = 4**

	A	B	C
1	Tools	Price	Inventory
2	Hammer	\$5.00	55
3	Screw Driver	\$2.50	66
4	Pliers	\$3.34	333

**MATCH(66, \$A\$3:\$C\$3, 0) = 3**

Matching the word "Pliers" in column A, we find it in the **4<sup>th</sup> row**. Matching the number 66 in row 3, we find it in the **3<sup>rd</sup> column**



**INDEX** and **MATCH** are commonly used in tandem to act like a LOOKUP function; the only difference is that **INDEX/MATCH** can find values in any column or row in an array

Example: Price Checker

	A	B	C	D
1		Small	Medium	Large
2	Sweater	\$10	\$12	\$15
3	Jacket	\$30	\$35	\$40
4	Pants	\$25	\$30	\$35
5				
6	Product:	Pants		
7				
8	Size:	Medium		
9				
10	PRICE:	?		
11				

In this example, we want to populate the price of a given product and size in cell B10 by returning a particular value within the array B2:D4

**B10=INDEX(B2:D4, MATCH(B6,A2:A4,0), MATCH(B8,B1:D1,0))**

The number of rows down to index depends on what product I'm looking for, so we use a MATCH function and search for the value in cell B6 (in this case "Pants")

The number of columns over to index depends on what size I'm looking for, so we use a MATCH function and search for the value in cell B8 (in this case, "Medium")

Considering the output of each MATCH function, the formula is just a simple INDEX:

**B10 = INDEX(B2:D4, 3, 2) = \$30**

## Building the Dashboard

Create a Dashboard using SUMIFS, COUNTIFS, and AVERAGEIFS

You can create a dashboard using PowerPivot or with the OFFSET function and a scrollbar. This module employs the SUMIFS, COUNTIFS, and AVERAGEIFS functions.

Use SUMIFS, COUNTIFS, and AVERAGEIFS when you want to analyze the sum, count, or average based on multiple criteria. The syntax for COUNTIFS is the criteria range and criteria repeated until all criteria are added.

=COUNTIFS(criteria\_range1,criteria1,criteria\_range2,criteria2,...)

The SUMIFS and AVERAGEIFS syntax adds the sum or average range up front:

=SUMIFS(sum\_range, criteria\_range1,criteria1,criteria\_range2,criteria2,...)

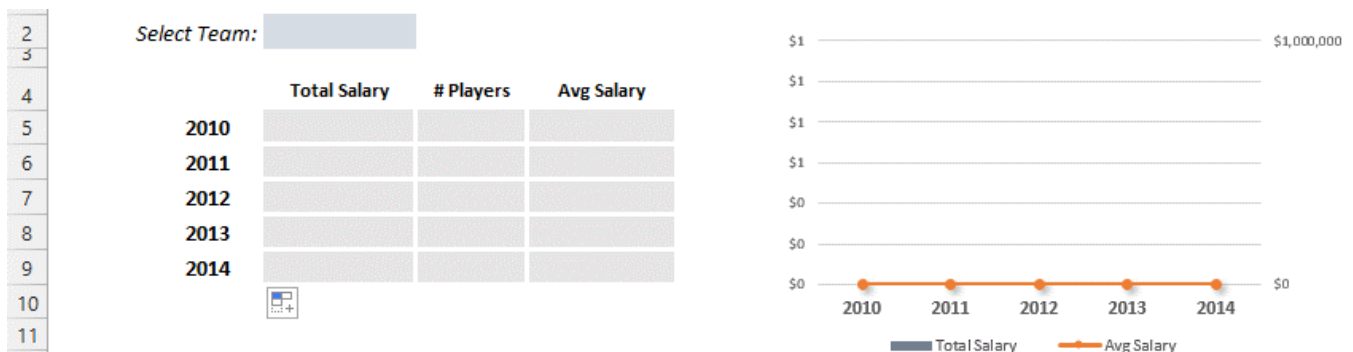
=AVERAGEIFS(average\_range, criteria\_range1,criteria1,criteria\_range2,criteria2,...)

We have gathered salary data on professional baseball players for the years 2010-2014.

Year	Team	League	Player	Salary
2010	NYA	AL	rodrial01	\$33,000,000
2011	NYA	AL	rodrial01	\$32,000,000
2012	NYA	AL	rodrial01	\$30,000,000
2013	NYA	AL	rodrial01	\$29,000,000
2011	LAA	AL	wellsve01	\$26,187,500
2014	LAN	NL	greinza01	\$26,000,000

On a separate sheet, we drew the dashboard and created a corresponding combo chart for you.

1. Use Data Validation to create a drop-down list of team names in cell C2
2. Use the SUMIFS function to pull the salaries into column C
3. Use COUNTIFS to bring the number of players into column D
4. Write the formula to calculate the average in column E



The **OFFSET** function is similar to **INDEX**, but can return either the value of a cell within an array (like INDEX) or a specific *range* of cells

**=OFFSET(reference, rows, columns, [height], [width])**

What's your starting point?      How many rows down should you move?      How many columns over should you move?      If you want to return a multidimensional array, how tall and wide should it be?

An **OFFSET** formula where [height]=1 and [width]=1 will operate exactly like an **INDEX**. A more common use of **OFFSET** is to create dynamic arrays (like the Scroll Chart example in the appendix)



### PRO TIP:

*Don't use OFFSET or INDEX/MATCH when a simple VLOOKUP will do the trick*

## Advanced Data Validation

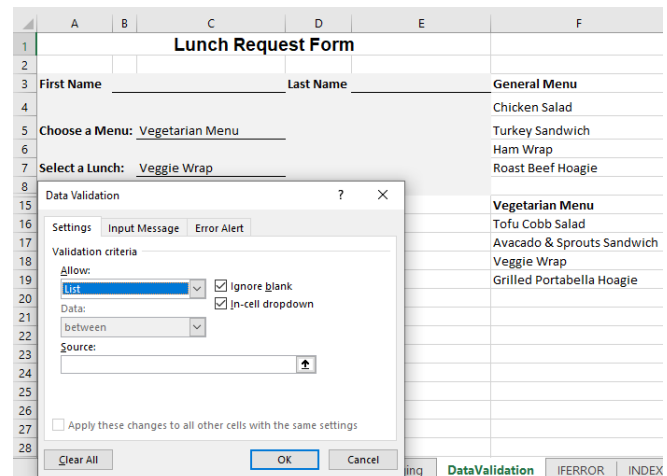
One of best ways to avoid errors in a database is to ensure that the data entered is accurate and consistent. Data Validation allows you to specify and enforce the format of data when it is being entered.

### Custom Criteria

Use a formula to determine validation criteria

1. Select the cell or cells
2. Click Data Validation button in the Data Tools group on the Data tab
3. Choose Custom [B] from the Allow: dropdown options
4. Enter the formula that describes your validation rule

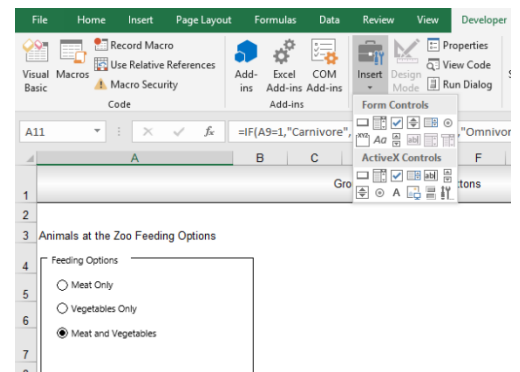
This example counts the number of entries in a cell range and alerts the user if more than 3 suggestions have been entered.



## Form Controls

Form controls can be very useful for adding form-like functionality to a workbook, without having to use VBA. Form controls make use of linked cells and the value in the linked cell is updated in some way, based on the actions taken with the form control at hand. Using formulas in combination with form controls, one can further extend the form control functionality and form-like appearance of the worksheet at hand. To change a control's properties to link it to a background database sheet:

1. Right-Click on the control and select Format Control.
2. Modify control options as needed.
3. To link the control to a background database sheet, input the destination cell address into the Cell Link.



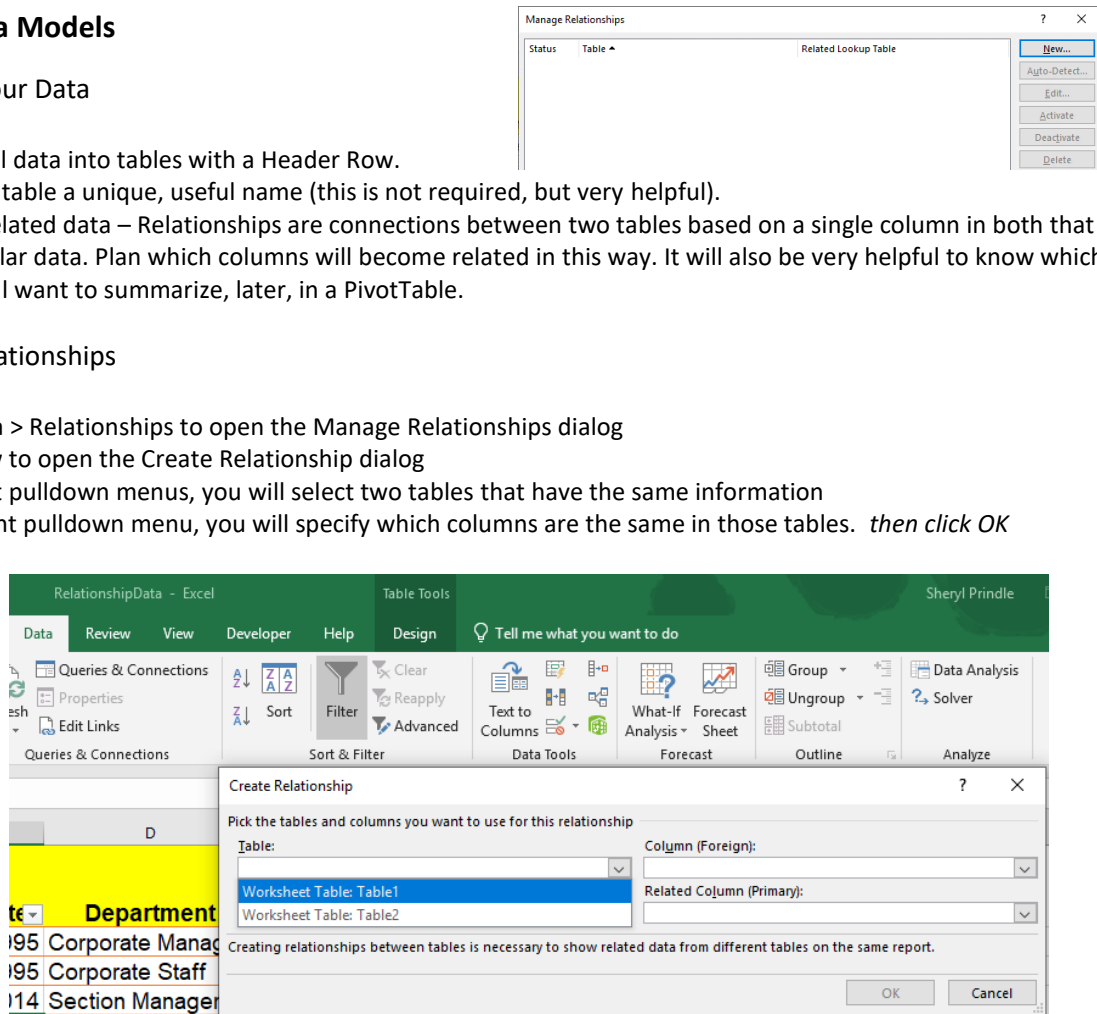
## Excel Data Models

### Prepare Your Data

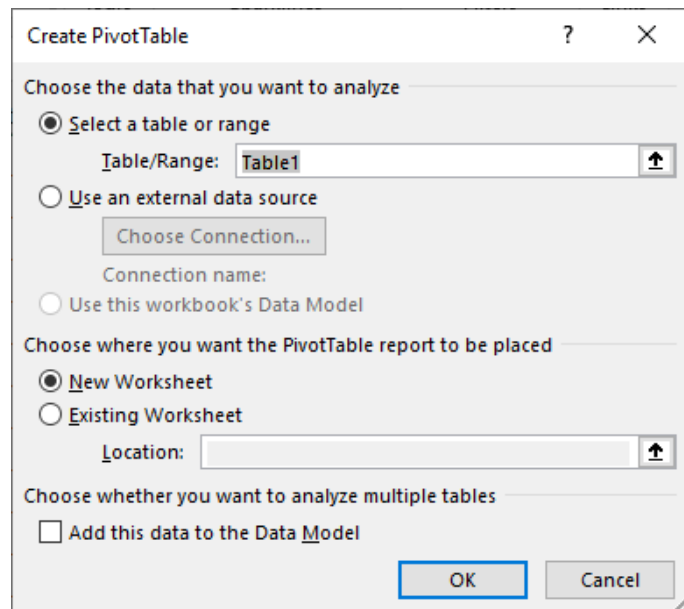
- Convert all data into tables with a Header Row.
- Give each table a unique, useful name (this is not required, but very helpful).
- Identify related data – Relationships are connections between two tables based on a single column in both that share a similar data. Plan which columns will become related in this way. It will also be very helpful to know which field you will want to summarize, later, in a PivotTable.

### Create Relationships

1. Click Data > Relationships to open the Manage Relationships dialog
2. Click New to open the Create Relationship dialog
3. In the left pulldown menus, you will select two tables that have the same information
4. In the right pulldown menu, you will specify which columns are the same in those tables. *then click OK*



You can pull the data from your model into a Pivot Table





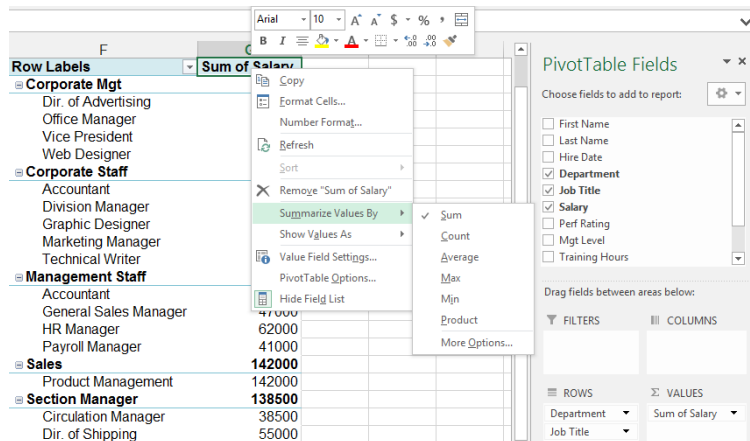
## Pivot Tables

You can take your spreadsheet apart column by column and put it back together in any configuration you want. Pivot Tables allow you to analyze and recombine data into reports and charts.

1. Select one cell within the data
2. Click Insert > Pivot Table > OK
3. Check or drag items in the *Field List* to the *Row*, *Filter*, *Column*, or *Values* areas

Right click *Values* column headers to change the operation

Right click row and column headers to see Group options



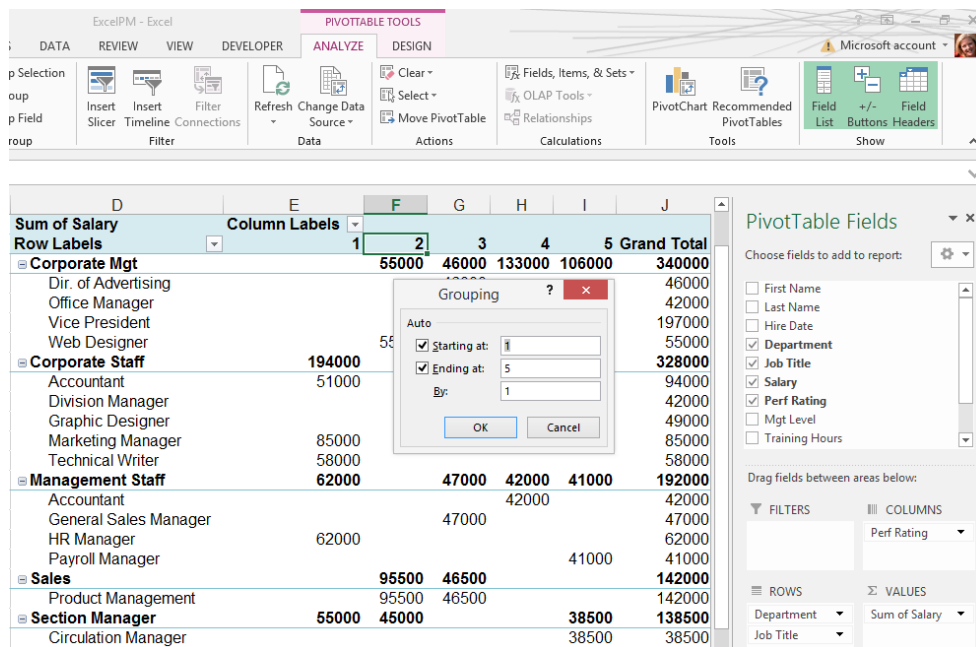
Explore the *Pivot Table Tools Analyze* and *Design* ribbons for options such as:

Create Pivot Chart

Hide Field List

Insert Slicer

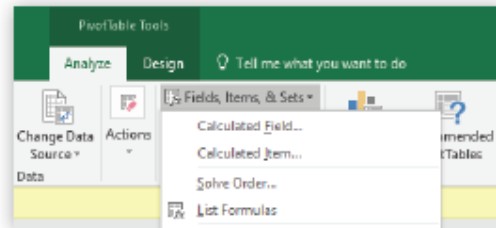
Adjust Layout



Make a Separate Sheet for All Values in a Field

1. Drag the desired field to the Report Filter area
2. On the Analyze Tab, click Options drop-down arrow in the PivotTable group
3. Select Show Report Filter Pages
4. Select the Field you want to use

Once your data is organized within a PivotTable, you will likely wish to run further analysis upon the summarized information. Calculated Fields and Calculated Items allow you to use this aggregated data as a formula source.



#### About Calculated Items

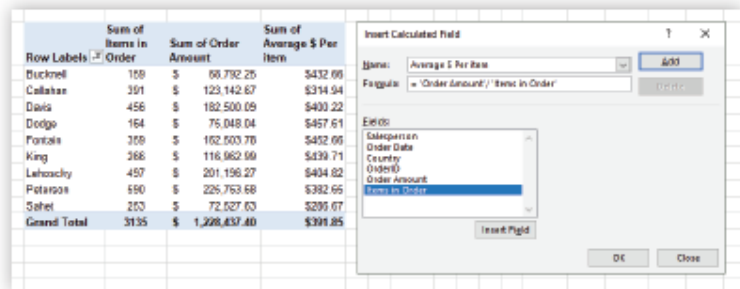
- Calculated items become an additional item in a pivot field.
- It can calculate sums among *other items in the same field only*.
- Calculated items will not appear in the PivotTable Field List.

#### About Calculated Fields

- Calculated fields perform calculations on *other fields in the pivot table*.
- Calculations are performed on the total or summarized amounts.
- Sum is the only function available for a calculated field.
- Calculated fields will appear in the PivotTable Field List.

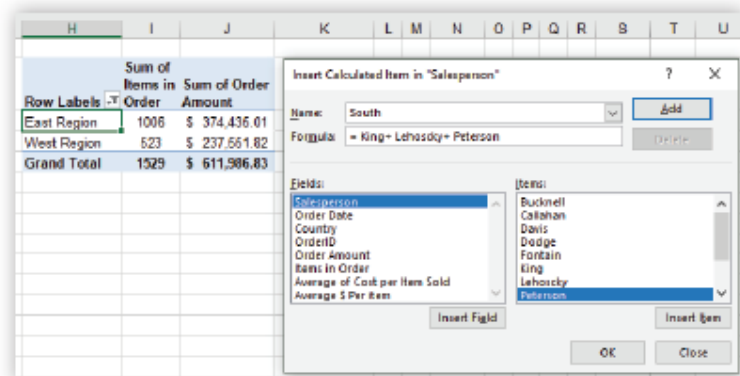
#### Create a Calculated Field

1. Go to the PivotTable Tools Analyze tab.
2. Click the Fields, Items, & Sets button.
3. Select Calculated Field.
4. Type a descriptive name for your new field in the Name textbox.
5. Type your formula using the Fields selection box to insert the fields that will be used in the calculation.
6. Click OK.



#### Create a Calculated Item

1. Go to the PivotTable Tools Analyze tab.
2. Select the field to which you wish to add a new item, then click Fields, items, & Sets.
3. Select Calculated Item.
4. Type a name for the new item in the Name textbox.
5. Select the fields and items that will be used in your calculated item's formula by clicking Insert Field or Insert Items.
6. Click OK.



## Style and Format Your PivotCharts

Styling PivotCharts is similar to working with standard charts with a few twists.

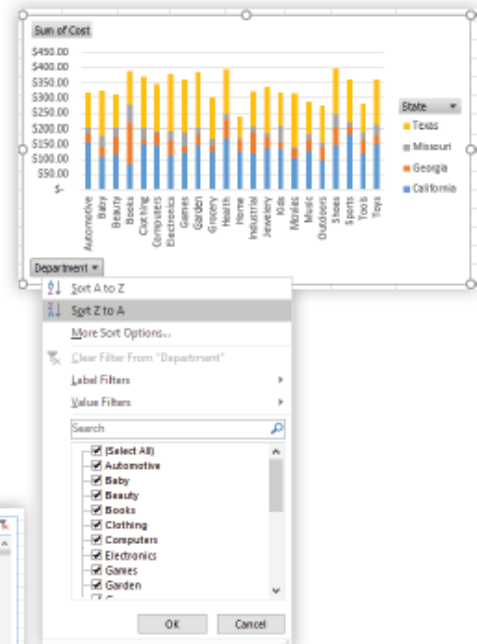
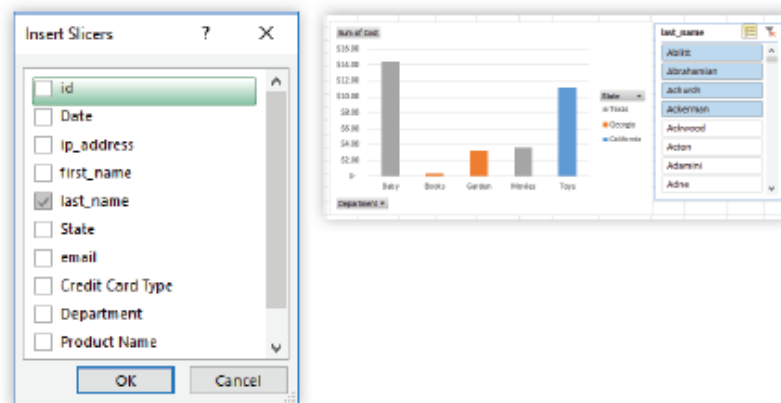
### Filter and Sort Pivot Charts on the Fly

Click the X or Y Axis dropdown button to access sorting and filtering options, right from the chart object.

### Add a Slicer

Slicers add additional ways to filter your chart results beyond the Axis data. To add a slicer to your chart, click **Insert Slicer** button on the **PivotTable Tools Analyze** tab.

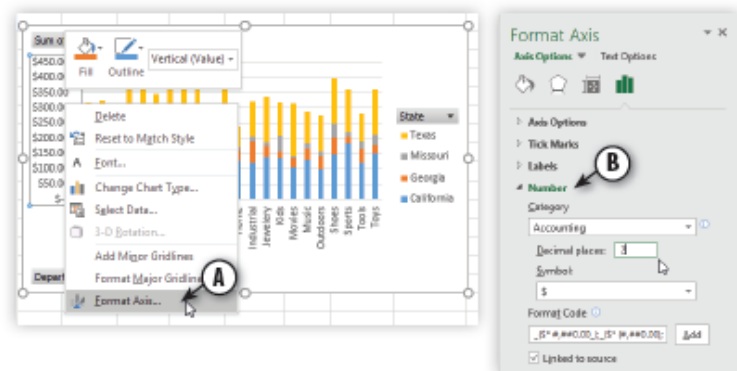
Choose the fields on which you wish to apply filters and click **OK**.



### Choose a Different Number Format for Your Chart

To change the number formats of values in your chart, you can follow the steps above to set the formats in the background PivotTable. However, you may find that there are times when you want your PivotChart to display Axis values in a *different* number format that format than the background PivotTable.

1. Right-click a number in the Axis.
2. Select **Format Axis** [A] to open the **Format Axis** pane.
3. Click the open triangle beside the **Number** option. [B]
4. Make your changes.



## Conditional formatting

- 1 Select the cell or range containing the values to which you want to apply conditional formatting
- 2 In the Styles group, click Conditional Formatting
- 3 Do one of the following:
  - Choose a condition and rule from the menu
  - Choose New Rule. Then specify a condition and format in the New Formatting Rule dialog box

**State:** Arizona

State	Population	Student Pop.	SAT Participation Rate	Mean Verbal Score	Mean Math Score
Alabama	4447100	177884	9%	559	554
Alaska	626932	12539	51%		
Arizona	5130632	102613	34%		
Arkansas	2673400	53468	6%		
California	33871648	1016149	51%		
Colorado	4301261	172050	31%		
Connecticut	3405565	102167	82%		
D.C.	783600	23508	56%		
Delaware	572059	17162	67%		

*In this example we're formatting the cells in columns B through H with a green fill and bold text, but only when the state name is equal to the value in cell \$C\$2*

*Note that the row label is relative (no "\$"), which allows us to apply this formatting to other rows without losing functionality*

**Edit Formatting Rule**

Select a Rule Type:

- Format all cells based on their values
- Format only cells that contain
- Format only top or bottom ranked values
- Format only values that are above or below average
- Format only unique or duplicate values
- Use a formula to determine which cells to format

Edit the Rule Description:

Format values where this formula is true:

=B6:\$H6=\$C\$2

Preview: AaBbCcYyZz

OK Cancel

## Power Query and Power Pivot

File Home Insert Draw Page Layout Formulas Data Review View Developer Inquire Power Pivot

From Access From Web From Other Sources Existing Connections Get External Data

From File From Database From SQL Server Database From Azure From Online Services From Other Sources Combine Queries Data Catalog Search My Data Catalog Queries Data Source Settings... Query Options

From Microsoft Access Database From SQL Server Analysis Services Database From Oracle Database From IBM DB2 Database From MySQL Database From PostgreSQL Database

First Name	Last Name
alan	ald
george	lopez
marie	osmond
bruce	lee
marcia	brady
will	smith

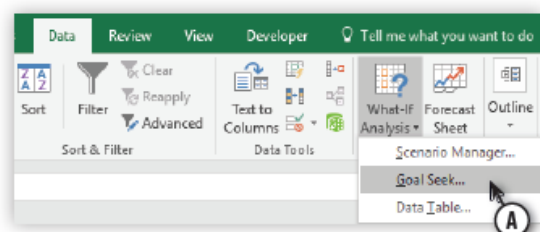
## What-If Analysis

### Goal Seek

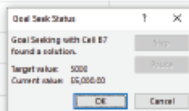
Goal Seek allows you to change the value of various cells to see how the results change. It can help you figure out what numbers need to change in order to get the end result you want.

1. Click the cell you want to change. This is typically the end result which contains a formula.
2. On the Data tab click What-If Analysis.
3. Click Goal Seek. [A]
4. Set Cell: The cell selected will be the set cell. [B]
5. To Value: Set the end value you would like to see in that cell. [C]
6. By Changing Cell: Select the cell you would like to automatically change to reach your goal. [D]
7. The new value will appear in the dialog box.
  - a. To make the changes to the spreadsheet, click OK.
  - b. To keep the original value, click Cancel.

	A	B
1		
2	Starting Investment	\$ 350.00
3	Annual Interest Rate	6%
4	Monthly Investment	\$ 100.00
5	Months	12
6		
7	Future Value	\$1,605.14
8		



	A	B	C	D
1				
2	Starting Investment	\$ 350.00		
3	Annual Interest Rate	6%		
4	Monthly Investment	\$ 375.21		
5	Months	12		
6				
7	Future Value	\$5,000.00		



### Setting Up a Web Query

1. On the File tab, in the Get External Data group, select From Web
2. In the Address field [B], type the Web address for the page that contains the information to be pulled into Excel
3. Click the arrow next to the table(s) to be added to a spreadsheet
4. Click the Import button
5. Select the location for the data and click OK

