Macros

Macros in Excel are in short, a recording of keystrokes. Beyond simple recording, you can use macros to automate tasks that you will use over and over. In this exercise, we will find all cells that contain formulas with the understanding that we may use this feature in different spreadsheets or over and over.

*From the previous workshop, bring out the Developer Tab from the File-Options-Ribbon-Developer

- Open the Stocks Spreadsheet
- Click on Cell B1
- In the Developer Tab, click on the Record Macro button.
- Change the Macro Name to “FindFormulas” – no spaces
- Change the Shortcut Key to Ctrl + Shift + T (press Shift + T) to add it as a shortcut.
- For now, keep This Workbook
- Put in a description on what the Macro will do in the “Description” box.
- Press OK
- Go the Home Tab and click on the Find & Select button.
- Add a color to the cells by clicking on the Paint Bucket button in the Home Tab. Add a color.
- Go back to the Developer Tab and click Stop Recording.
- Click anywhere on a blank cell, and all your cells that contain formulas will be highlighted in a color.
- Undo the cell coloring by pressing Ctrl + A and select No Fill from the paint bucket button from the Home Tab.
- Now on the spreadsheet, press Ctrl + Shift + T – this will run your macro!
- Now try the Macro on a Loan spreadsheet. It should highlight your cells with formulas.

Microsoft Visual Basic for Applications Environment

Again, you do not need to know a lot about VBA code, but it is a good idea to at least get familiar with the VBA environment.

- To quickly get to the VBA environment, you just need to press Alt + F11 or you can Click on the Macro button in the Developer Tab.
- On the left side of the screen, you will see the Project–VBAProject menu that will contain your spreadsheets and macros.
- All your macros are saved in the folder titled: Modules
- Since we have one macro, you will see Module 1 when you click on the Modules folder.
- Double click on Module 1 and the VBA code for the macro we created above will appear.
• The code on the right is code to run the macro to find formulas on a spreadsheet and change the cells to a color. Any text in green in the code are just comments and they will not affect anything.

![Image of code]

• The instructor will discuss with the code means, but you can always add code to VBA scripts to make the macro more customized.
• In this example: type `Range("B3").Select` between "End With" and "End Sub"
• What this is going to do is move your cursor to cell B3 after the macro is run.
• Press ALT+F11 and run the macro on the Stocks spreadsheet.
• Your cursor will end up in cell B3

**Macros as Buttons**

There are two ways to create buttons in Excel that will run a macro in the background that will make tasks that much easier to process. In most cases, you will set up buttons when you must send the spreadsheet over to data entry clients so they do not mess up data on your spreadsheet. One method is to create a button from a shape or insert a picture or you can assign a button already made in the Developer Tab.

• Go back to the Table spreadsheet.
• Go to the Developer Tab and click on Record Macro.
• Macro Name: SortTable
• Shortcut Key: Ctrl + Shift + Y
• Click OK.
• Highlight cells A4-A36, and in the Home Tab, click Sort A-Z
• Go back to the Developer Tab and click: Stop Recording.
• Now press the undo button on the top left of the screen to unsort the data.
• Go to the Insert Tab -> Shapes -> Insert a rectangle
• In the rectangle, type Sort by Name
• Right-click in the rectangle and select Assign Macro
So far, we only have one macro assigned to this worksheet, so click on TableSort and change Marcos In to This Workbook. (Since this macro is designed for this spreadsheet, that is why we are choosing this option)

Click Ok.

Click outside the button to deselect the object.

Now when you move your mouse over the button, it should change to a finger ready to press. Press the button, and it should sort your data. (*You can easily insert a picture and turn that into a button what a macro. Most businesses will do this with a logo.)

Excel already has pre-designed buttons that do tasks to help build surveys.

In this example, let’s create a macro to print a spreadsheet and then create a button to press to quickly print.

Go to the Developer Tab -> Record Macro. (Name the macro PrintArea with Cntl+Shift+G)

Print the spreadsheet like you normally do: File -> Print -> Print.

Go back to the Developer Tab -> Stop Recording.

Click on the Insert Button in the Developer Tab and Select the Button command.

You can draw a button on the spreadsheet just like drawing a shape.

When you draw a button, the Assign Macro menu will appear.

Now we should have two macros.

Click on the PrintArea and click Ok.

Now press the button, and it should print the spreadsheet.

You can rename the button to “Print” to add what the button will do.

**Macros with a Conditional IF Statement**

Hopefully you remember conditional formulas from the previous workshops. In a nutshell, we just added a condition (true or false statement) to the logic in a formula to get a specific result based on the condition. We can apply the same logic with macros to set conditions.

- Go back to the Table spreadsheet.
- We need to record another macro. One macro accountants use frequently is to create a toggle macro with accounting underlining.
- Go to the Developer Tab -> Record Macro.
- Highlight cells E3-E35
- Go to the Home Tab -> Underline drop down -> Double Underline
- Go back to the Developer Tab and press Stop Recording.
- Now we need to create a macro removing the double underline.
- Go back to the Developer Tab and press Record Macro.
- Name this macro RemoveUnderline and press Shift + D.
- Highlight cells E3-E35 and press Cntl + U twice to remove the underline.
- Go back to the Developer Tab and press Stop Recording.
- Press Alt + F11 to get to the VBA code.
- The VBA code either went into separate modules or on the same module, depending on if you saved the macro to the worksheet, workbook, or in personal workbook. In this example, it should have been put into Module1. Either way, you can cut and paste the remove underline code into the module that has the insert underline code.
Right now the code should look like this:

```vba
Sub AccountUnderline()
' Applies Underlining
' Keyboard Shortcut: Ctrl+Shift+F
    Range("E3:E35").Select
    Selection.Font.Underline = xlUnderlineStyleDoubleAccounting
End Sub
```

In the Remove Underline, delete the line `Selection.Font.Underline = xlUnderlineStyleSingleAccounting`

Remove the `End Sub` line:

```
Sub RemoveUnderline()
' RemoveUnderline Macro
' Keyboard Shortcut: Ctrl+Shift+D
    Range("E3:E35").Select
    Selection.Font.Underline = xlUnderlineStyleSingleAccounting
    Selection.Font.Underline = xlUnderlineStyleNone
    Range("E2").Select
End Sub
```

If you have any other code, go ahead and delete that out here as well.

In the first `Selection.Font.Underline` line type:

```
IF Selection.Font.Underline = xlUnderlineStyleDoubleAccounting Then
```

Under the last `Range("E2").Select`, type:

```
Else
```

Copy and paste the first two lines without the `If Then` and press Enter:

```
IF Selection.Font.Underline = xlUnderlineStyleDoubleAccounting Then
Range("E3:E35").Select
Selection.Font.Underline = xlUnderlineStyleDoubleAccounting
Else: Range("E3:E35").Select
Seleccion.Font.Underline = xlUnderlineStyleDoubleAccounting
```

Last, type `endif` under the last line before `End Sub` (press the Enter key to apply the new code)
- The final code should look like:

```vba
Sub AccountUnderline()
    ' AccountUnderline Macro
    ' Applies Underlining
    ' Keyboard Shortcut: Ctrl+Shift+F
    Range("E3:E35").Select
    If Selection.Font.Underline = x1UnderlineStyleDoubleAccounting Then
        Range("E3:E35").Select
        Selection.Font.Underline = x1UnderlineStyleNone
        Range("E2").Select
    Else: Range("E3:E35").Select
        Selection.Font.Underline = x1UnderlineStyleDoubleAccounting
    End If
End Sub
```

- Go back to your spreadsheet by pressing Alt+F11.
- Now press Ctrl + Shift + F to apply the macro. Press it again, and it should toggle off.

**Macros with Next and Repeat Actions**

Next and Repeat macro are great when you repeat tasks within a range. In other words, if you see the same sequence of code in the VBA on a macro, you will want to use the Next and Repeat macro to automate the task(s). You will more than likely do this when you have a list of client data down a spreadsheet, and you are attempting to visually change the look of each client.

- Go back to the Table spreadsheet.
- Click in cell B3.
- Press Record Macro. (Macro name: bolit& Shortcut Key: Ctrl + Shift + J)
- Because we are not going to run an Absolute Macro, press “Use Relative References” in the Developer Tab.
- Highlight cells B3-B5 and press Ctrl + B to bold.
- Highlight cells B6-B8 and press Ctrl + I to turn the text italics.
- Return to the Developer Tab and press Stop Recording.
- Press ALT + F11 to open the VBA screen.
- For this exercise, let’s split the screen. Minimize the VBA screen and drag it to the top. Minimize the Excel screen and place it in the bottom of the screen: ->>>
- Splitting the screen will allow you to see the VBA code in action as a macro runs.
Now, to have the Bold – Italics repeat we will have to change the VBA code with a for statement.

Open a space between the ActiveCell.Offset(0, -8) and the Selection.Font.Bold = True

In the open space type: For Counter = 1 to 10. (You can enter any end number, but we are going to pretend we are running this macro ten times before it completes.)

Open a space between the last ActiveCell.Offset and the End Sub.

In the open space type: Next Counter (this will tell Excel to go back to the top and repeat.)

Your code should look like:

```
Sub bolit()
'
' bolit Macro
'
' Keyboard Shortcut: Ctrl+Shift+K
'
    Range("B3").Select
    ActiveCell.Range("A1:A3").Select
    For Counter = 1 To 10
        Selection.Font.Bold = True
        ActiveCell.Offset(3, 0).Range("A1:A3").Select
        Selection.Font.Italic = True
        ActiveCell.Offset(3, 0).Range("A1:A3").Select
    Next Counter
End Sub
```

Click just to left of the Sub bolit() text in the macro VBA script.

Now instead of pressing Alt + F11, in the VBA screen go to Debug -> Step Into

This is going to allow us to see the script run line by line.

Now press F8 and watch the macro in action!

Keep pressing F8 until the macro complete.

**Macro Interactivity and Do Statements**

There is one more popular conditional macro statement that is highly used in Excel, especially for business. What this does is loop a macro and automatically stop. You can also add pop up boxes and other interactivity with macros to help data entry perform a task quickly.

- Go back to the Sub bolit() macro in the VBA screen to get to the code.
- Delete the For counter line and change it to: **Do While ActiveCell <> ""**
- Delete the Next counter and change it to: **Loop**
- Press the enter key to activate the code.
- The code should look like:

```
Sub bolit()
'
' bolit Macro
'
' Keyboard Shortcut: Ctrl+Shift+K
'
    Range("B3").Select
    ActiveCell.Range("A1:A3").Select
    Do While ActiveCell <> ""
        Selection.Font.Bold = True
        ActiveCell.Offset(3, 0).Range("A1:A3").Select
        Selection.Font.Italic = True
        ActiveCell.Offset(3, 0).Range("A1:A3").Select
    Loop
End Sub
```
- Go back to the Excel spreadsheet and press Ctrl + Shift + K to run the Macro. It should automatically run and perform the task.
- Once you move into learning VBA script, there are several other ways script this out. The easier code would be: Do Until IsEmpty(ActiveCell)
- At times you may need to have your data entry clients input data into a spreadsheet after following a prompt; like enter a spreadsheet name to get to an active sheet.
- This really just need a minor adjustment in the VBA code.
- Go back to the VBA screen and go to the Sub bolit() macro
- Under the Sub bolit() line, type the code:
  ```vba
  GoToWorksheet = InputBox(“Enter a Worksheet Name:”, “Worksheet Selector”)
  Worksheets(GoToWorkSheet).Activate
  ```
- The code should look like this:
  ```vba
  Sub bolit()
  WorksheetName = InputBox("Enter a Worksheet Name:", "Worksheet Selector")
  Worksheets(WorksheetName).Activate
  
  ' Keyboard Shortcut: Ctrl+Shift+K
  
  Range("B3").Select
  ActiveCell.Range("A1:A3").Select
  Do While ActiveCell <> ""
  Selection.Font.Bold = True
  ActiveCell.Offset(3, 0).Range("A1:A3").Select
  Selection.Font.Italic = True
  ActiveCell.Offset(3, 0).Range("A1:A3").Select
  Loop
  End Sub
  ```

### Creating Cell Drop Down Menus and Adding a Condition

From time to time you may need to have a list of options for your clients to choose from; this is especially true when creating contracts, invoices, and forms in Excel. For example, you can have a cell activate as a drop down with the options of 1 month, 3 months, 6 months, or 12 months when creating a lease contract. Then you can have the “amount due each month” cell auto-populate with the correct rent amount without having to type the information in yourself.

- Click into cell A41 in the table spreadsheet.
- Type “1 month” in A41, “3 months” in A42, “6 months” in A43, and “12 months” in A44.
- Click in cell B41. (We are going to place the list here.)
- Go the **Data Tab** and click **Data Validation**.
- Click **Data Validation** again.
- The Data Validation menu will appear.
- In the **Allow -> List**.
• Then either type the absolute formula =A41:A44 or select the cells after clicking in cell A41 - A44.
• Make sure In-cell dropdown is selected. This will allow us to apply this drop down anywhere on the spreadsheet.
• Click Ok.
• Now when you click into cell B44, you will see the droplist. You can choose any of the options.

Let’s apply a conditional formula (review the breadcrumb conditional formula from a previous workshop).

• In cell C43 type: =IF(B41="1 month","$400 per month",IF(B41="3 months","$600 per month",IF(B41="6 months","$900 per month",IF(B41="12 months","$1200 per month","Write In"))))
• Now cell C43 will update as the information changes in cell C43! What we are saying with this formula is what lease by month a person signs, the appropriate rent will apply in the cell C43.

Multi-Layered Lists
You can create multi-layered lists with a drop down to auto-input information into a table or data set.

• Go to the Cell List spreadsheet.
• Let’s create list by region and another list of states by region.
• Click the letter A to highlight all of column A.
• Go to the Data tab -> Data Validation -> Data Validation.
• Change Allow to List and Source: Highlight the cells that has the region information (E1 - J1)
• Click Ok.
• Now click into cell A2 and you can see the drop down list.
**This list will apply to the entire column since we highlighted the column when creating the list**
• To add the corresponding states to the region in column B, highlight all columns and rows that contain the data you need — include blank cells in columns if some columns don’t have as long a list. For this example, highlight cells E1 - J9. This should highlight all cells that have data.
• Go to the Formulas tab -> Create from Selection

In the Create Names from Selection box, put a check mark Only in Top Row.
As a sidenote, look at your NameBox on the top left corner when you click in any cell that is highlighted. It should change to the corresponding Region. Hopefully, this is a reminder from the first BasicExcel workshop.

- Click on B to highlight all of column B.
- Go to the Formula tab -> Data Validation -> Data Validation.
- Change Allow to List.
- In the Source type the formula: =indirect(A1) * This is telling Excel to go back to the list in cell A1 to correspond to the data in column B.
- Click OK.
- You more than likely will get an error “The Source currently evaluates to an error. Do you want to continue?” Click Yes.
- Click in cell A2, and choose West.
- Click in cell B2, and the list should only show states in the West.
- Now, to remove the blankspaces in the list of States, highlight all the state data, without the header information. No: West, Southwest, East, etc. (Only cells E2 – J9)
- Go to the Home tab -> Find and Select -> Go to Special
- In the Go To Special select Blanks.
- Click OK.
- Right click in any of the highlighted blank cells -> Delete.
- Select Shift cells up.
- Now test the state drop downs to see if you get any blankspaces at the end of the lists. They should now be removed.
Forecast Spreadsheets
Typically investors and accountants will use the forecast feature when looking at stock or other investment information. In this case, let’s forecast Google Stock by month.

- Go to: https://finance.yahoo.com/quote/GOOG/history?p=GOOG
- Change Frequency to Monthly
- Click Apply
- Now click Download Data. This will export the data into Excel.

- Highlight columns A and E - the Date and Close information.
- Go to the Data tab -> Forecast Sheet
- You should get a preview pop-up window with a chart forecasting Google Stock.
- At the bottom of the Create Forecast Worksheet change Forecast End to 1/1/2018.
- Click Options
  - In the Options menu, you can set data to manipulate the data in the forecast:
    **Confidence Interval:** is important to predict forecast accuracy. Typically, the closer to 0% the more confidence you have in the prediction. That is why it is at 95%; Excel is giving you more of a spread. Try lowering the Confidence Interval and notice the spread shrink.
    **Seasonality:** is the number or periods in which you are looking for data. In our example, we are looking at Google stock by months so the seasonality is 12. You can change that here if you are looking for a different period.
    **Include Forecast Statistics:** this will add more statistical information when you create a worksheet on the data by adding a table using the statistics function FORECAST.ETS.STAT that will include measures and errors when analyzing the data. In other words, forecast by smoothing.
    **Fill Missing Points Using:** Excel will average out any missing points that is missing from the data. That is why Interpolation is selected. If you do not want that information forecasted, select **Zero** in our example, leave it Interpolation.
    **Aggregate Duplicates Using:** Excel will default average out data in the same time-period. You can change that here to a different algorithm here.
- Click Create.
- Now the forecasted data will appear in a new sheet. Typically, you are going to look at the future numbers at the end of the inserted table.
**Working with Data with Controls**

Data validation allows you to control the data that goes into each cell. It also allows you to force your data entry clients to format date the way you want.

- Go to the Macro spreadsheet.
- Click on the letter F to highlight the entire column.
- Go to the Data tab -> Data Validation -> Data Validation.
- Change **Allow** to **Decimal**.
- Change minimum to 100 and maximum to 10000.
- Click on the Input Message tab.
  - This is where we can add a “comment-like” message when we hover over any cell in column F – in Title: **Range Input Message: Only numbers between 1 and 10000**.
- Click on the Error Alert tab. This is where you can customize the error messages your data entry people will see if they type something wrong. In title: **Try Again – Error Message: Only numbers between 1 and 10000**.
- Click Ok.
- Go to cell F30 and type 12000. You should get the error message we just assigned.
- In the same cell, type 5000. It should work now.
- ***Look at all the data options you have in the Data Validation -> Allow options. You can customize and format numbers, decimals, dates, time, etc.

**Data Controls with Formulas**

- Go to the Date spreadsheet.
- Let’s assume we are going to put dates into column F that is at least 7 days from the days in column E. You typically will see this when your business provides shipping and handling.
- Click on the letter F to highlight all of column F.
- Go to the Data tab -> Data Validation -> Data Validation.
- In the Allow -> Custom.
- In the Formula box, type: =AND(YEAR(F1)=2015,F1>E1+7)
  - ***In a previous workshop, we discussed AND/OR and date formulas. We are just combining two.
- Click Ok.*** You may get the error “Anamed range you specified cannot be found” – Click OK.
- Now type 1/16/2015 in cell F2. You should get an error because it is not at least seven days after 1/15/2015.
- Now type 1/25/2015. It should work!
**OneDrive**

Microsoft provides free online storage and document creation directly on the browser with an account. (Outlook, Hotmail, Xbox, MSN, etc.) If you have a Microsoft account – including your student Outlook account – you already have a OneDrive account. Login by going to [www.onedrive.com](http://www.onedrive.com) and use your Microsoft login or Microsoft email username and password. (If you are using a Windows 10 PC, Onedrive is already on your File Explorer.)

With OneDrive, you can also share single documents or entire folders with specific people or with the public.