Excellence with Excel - Quiz Questions

Module 7: Introduction to VBA, Macros, and User-Defined Functions

1. **For which of the following scenarios is it the BEST idea to create a macro in Excel, as opposed to a VBA function, a user-defined function, or a custom keyboard shortcut?**

   a. You have a set of thousands of phone numbers from different countries, all in slightly different formats, and you need to standardize the formatting of each one based on different cases.

   b. You want to quickly calculate Net Income based on a company's Pre-Tax Income and Tax Rate.

   c. You apply the same formatting to all spreadsheets containing financial statements, and you want to active this automatically with a shortcut key.

   d. The keyboard shortcut for hiding a worksheet is too long and you're using it frequently, so you want to create a more concise shortcut.

2. **Should you use absolute or relative references when you record macros in Excel?**

   a. You should almost always use relative references, because you want the macro to work in data sets and spreadsheets of different shapes and sizes.

   b. It doesn't matter as long as you've selected the correct area in your spreadsheet before you activate the macro.

   c. You should use relative references – if you select only a certain cell type (e.g. all constants or all formulas) it won't work correctly in a macro if absolute references are used.

   d. You should use absolute references – with relative references you might accidentally apply your macro to the incorrect rows or columns.

3. **Suppose that you record a macro for formatting your financial model. You've used the correct font colors and background colors, but you've accidentally recorded the action of adding borders around each cell as well. What's the EASIEST way to fix this problem?**

   a. Re-record the entire macro and avoid adding or editing borders this time – you can't just modify a portion of the macro, so you need to re-do it.

   b. Press Alt + F11 (or otherwise enter the VBA Editor), pull up the VBA code for the macro, and delete the portions that refer to the border styles.

   c. Still use the macro as necessary, but simply press Ctrl + Shift + – to remove all the borders immediately after you're done.

   d. Bring up the "View Macros" dialog (Alt + W + M + V) and use the "Step Into" feature to see a visual representation of each step in the macro – and delete the undesired steps.

4. **You are writing VBA code for a complex function that makes calculations and then also formats cells based on the output of those calculations. However, you don't know all the VBA syntax required to access certain cell properties, such as alignment and text wrapping. What is the MOST efficient way to figure out the correct syntax to use?**

   a. Get a book on VBA or sign up for a VBA-specific course and memorize all the syntax and commands taught there.

   b. Use the "Help" function in Excel to look up these properties.

   c. Do random Google searches online until you find the exact syntax and properties you need to use.

   d. Record a macro that performs the formatting you need, bring up the VBA Editor, and copy and paste the syntax and change it around as necessary.

5. **What is the BIGGEST drawback of using macros in your Excel models?**

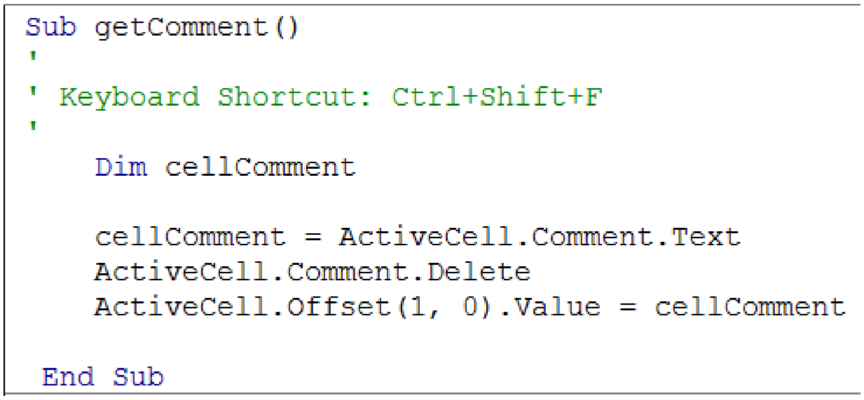
    a. Sometimes, you need to jump into the VBA Editor and modify code if you're writing a more complex function.

    b. You might accidentally apply the macro to the wrong cell or range of cells if you haven't used the correct reference type.

    c. Macros work well for formatting, but not for calculations and formulas.

    d. Undo and Redo do not work for any actions performed via a macro.

6. **Consider the VBA code in the screenshot below. This function is intended to retrieve a comment from the currently selected cell, put the comment text in the cell to the right of it, and then delete the comment. Why does this code not work correctly when we activate this macro with Ctrl + Shift + F, and how can we fix it?**



```
Sub getComment()
'
' Keyboard Shortcut: Ctrl+Shift+F
'
    Dim cellComment

    cellComment = ActiveCell.Comment.Text
    ActiveCell.Comment.Delete
    ActiveCell.Offset(1, 0).Value = cellComment

 End Sub
```

    a. We are deleting the comment before saving it in the other cell – the ActiveCell.Comment.Delete line should be at the bottom of the function instead.

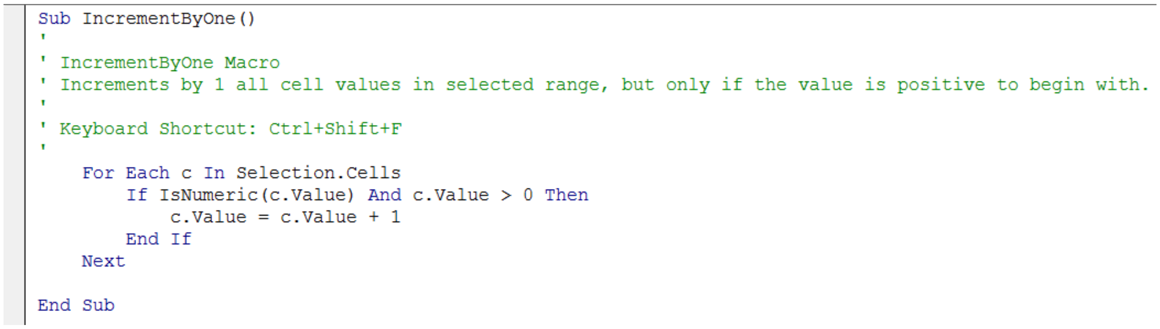
    b. The values within the Offset function are reversed – it should be Offset(0, 1) instead.

    c. We need to specify a range after the Offset function and before the Value function – if you make it ActiveCell.Offset(1, 0).Range("A1").Value then it will work correctly.

    d. To store a comment rather than normal text, you need to use "Set cellComment" rather than "Dim cellComment".

7. **You have to write a "For" loop in VBA that loops through the selected range of cells and increments the value of each cell by 1, but only if the cell actually contains a number and the cell's value is positive. Which of the following screenshots represent the CORRECT code to accomplish this?**

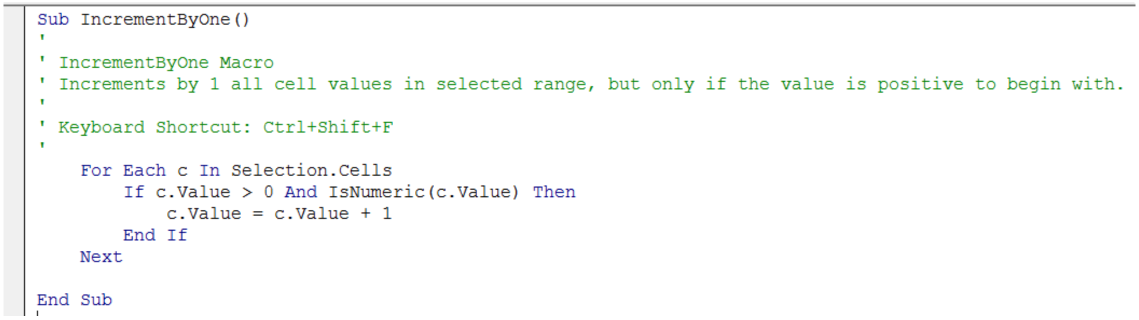
a.



```
Sub IncrementByOne()
'
' IncrementByOne Macro
' Increments by 1 all cell values in selected range, but only if the value is positive to begin with.
'
' Keyboard Shortcut: Ctrl+Shift+F
'
    For Each c In Selection.Cells
        If IsNumeric(c.Value) And c.Value > 0 Then
            c.Value = c.Value + 1
        End If
    Next

End Sub
```

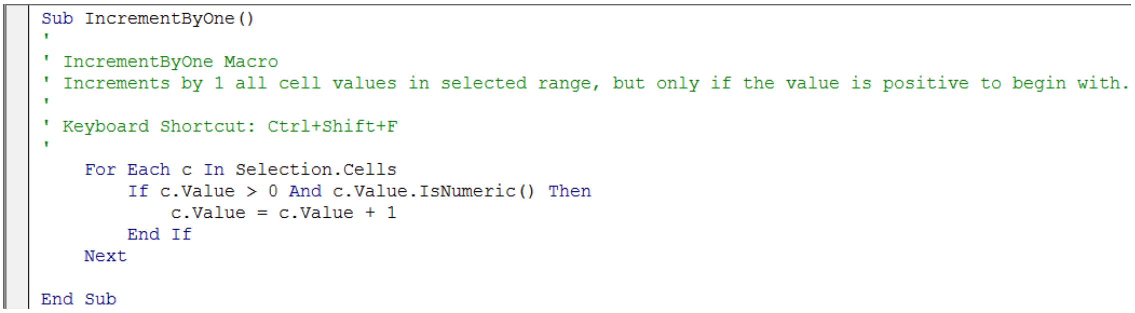
b.



```
Sub IncrementByOne()
'
' IncrementByOne Macro
' Increments by 1 all cell values in selected range, but only if the value is positive to begin with.
'
' Keyboard Shortcut: Ctrl+Shift+F
'
    For Each c In Selection.Cells
        If c.Value > 0 And IsNumeric(c.Value) Then
            c.Value = c.Value + 1
        End If
    Next

End Sub
```

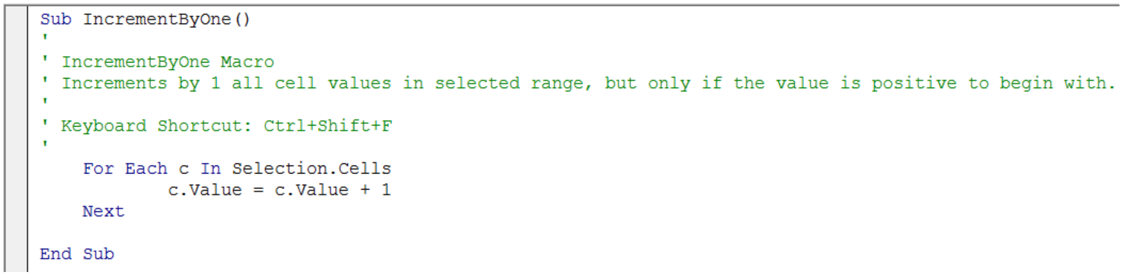
c.



```
Sub IncrementByOne()
'
' IncrementByOne Macro
' Increments by 1 all cell values in selected range, but only if the value is positive to begin with.
'
' Keyboard Shortcut: Ctrl+Shift+F
'
    For Each c In Selection.Cells
        If c.Value > 0 And c.Value.IsNumeric() Then
            c.Value = c.Value + 1
        End If
    Next

End Sub
```

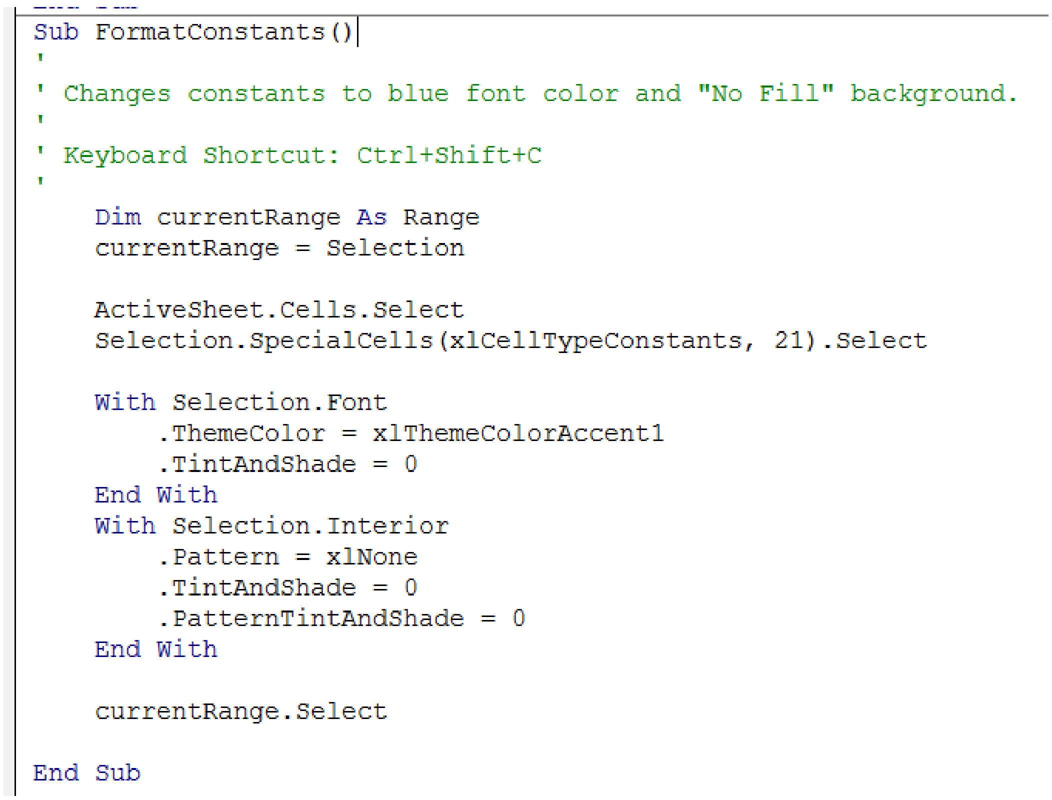
d.



```
Sub IncrementByOne()
'
' IncrementByOne Macro
' Increments by 1 all cell values in selected range, but only if the value is positive to begin with.
'
' Keyboard Shortcut: Ctrl+Shift+F
'
    For Each c In Selection.Cells
            c.Value = c.Value + 1
    Next

End Sub
```

8. **You are writing VBA code to format all constants in the current worksheet in blue font color, with a "No Fill" background. After you activate the macro, the selection in the worksheet should REVERT to whatever you had previously selected before activating the macro. What's the problem with the sample code in the screenshot below, and how can you fix it so that it works correctly?**



```
Sub FormatConstants()
'
' Changes constants to blue font color and "No Fill" background.
'
' Keyboard Shortcut: Ctrl+Shift+C
'
    Dim currentRange As Range
    currentRange = Selection

    ActiveSheet.Cells.Select
    Selection.SpecialCells(xlCellTypeConstants, 21).Select

    With Selection.Font
        .ThemeColor = xlThemeColorAccent1
        .TintAndShade = 0
    End With
    With Selection.Interior
        .Pattern = xlNone
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    currentRange.Select

End Sub
```

a. The problem is the last line of this function – instead of "currentRange.Select", you should use "Selection = currentRange" so that the currently selected range of cells reverts back to what it was before the function was activated.

b. The "ActiveSheet.Cells.Select" line will not work correctly because you need to specify a range of cells to select.

c. The problem is the second line from the top – you should write "Set currentRange = Selection" because you need to use "Set" whenever you're saving ranges to variables (instead of individual values).

d. The "Selection.SpecialCells(x1CellTypeConstants, 21).Select" line won't work correctly because you need to select from among ALL cells in the worksheet – not just from the ones in the currently selected range.

9. **Which of the following answer choices represent DIFFERENCES between Macros and User-Defined Functions (UDFs)?**

   a. You can modify Macros in the VBA Editor, but you cannot do the same with UDFs.

   b. UDFs have more limited functionality than Macros and you can't select or format ranges, as you can with Macros / VBA functions.

   c. Macros are better for actions applied to multiple cells or entire ranges, whereas UDFs are more useful for formulas you need to use that are NOT already built into Excel.

   d. You can record Macros with the built-in recorder in Excel, but you have to write UDFs from scratch by yourself.

10. **Which of the following answer choices represent GOOD use cases for User-Defined Functions (UDFs)?**

   a. Error-checking the Tax or Net Income calculation in a model to handle positives, negatives, 0, and invalid input.

   b. Formatting a wide range of data in Excel based on the value of each cell in that range of data.

   c. Creating a single function to handle a calculation that would normally require multiple steps, and therefore would be difficult to read and understand.

   d. Creating a function that properly fixes poorly formatted phone numbers and which must handle dozens of different cases.

11. You are creating a dynamic chart in Excel to show a company's average order size and total # of orders over the past 20 years. Now, you want to add a Scroll Bar form element to this chart so that it only displays 4 years' worth of data at once. Each time you move the Scroll Bar, it displays the next year in the data set. Which of the following answer choices represent CORRECT steps in the process required to implement this type of Scroll Bar on a dynamic chart?

a. Enable the Developer Toolbar within the Options menu.

b. Insert the Scroll Bar Form Control from the Developer Toolbar.

c. Right click the Scroll Bar, go to Format Control, and link the "Form Input" field to a specific cell in your worksheet that tracks which set of years the chart is on.

d. Right click the Scroll Bar, go to Format Control, and link the "Cell Link" field to a specific cell in your worksheet that tracks which set of years the chart is on.

e. Set the Incremental Change, Page Change, and Minimum Value and Maximum Value to appropriate numbers.

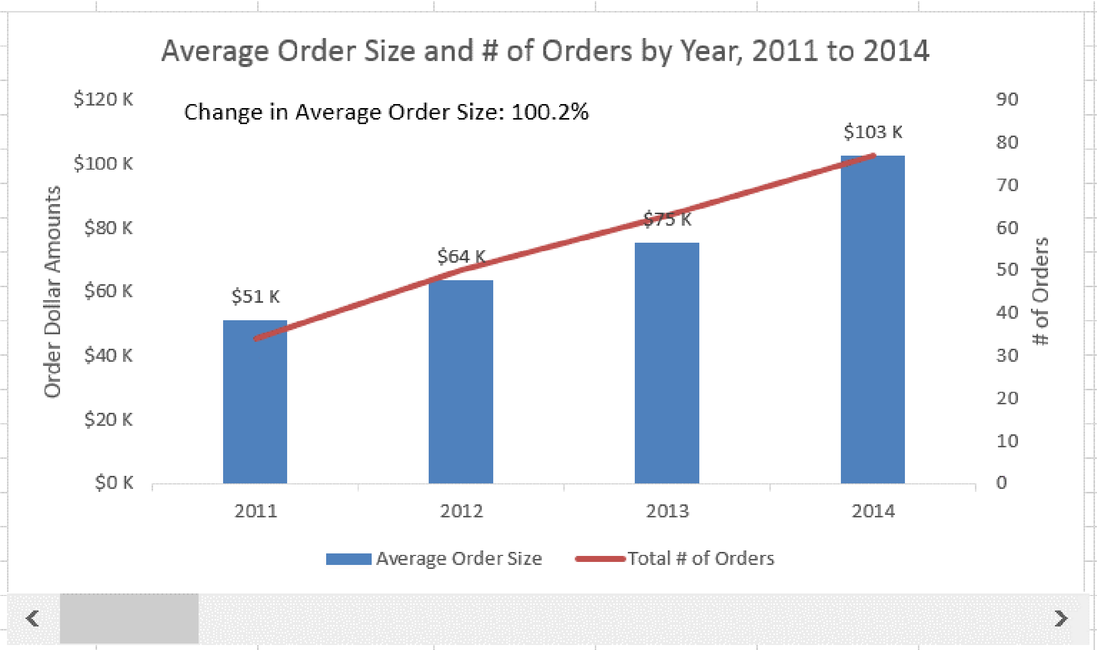
f. Go to the chart itself and select "Source Data" and then link it directly to the data you want to display there.

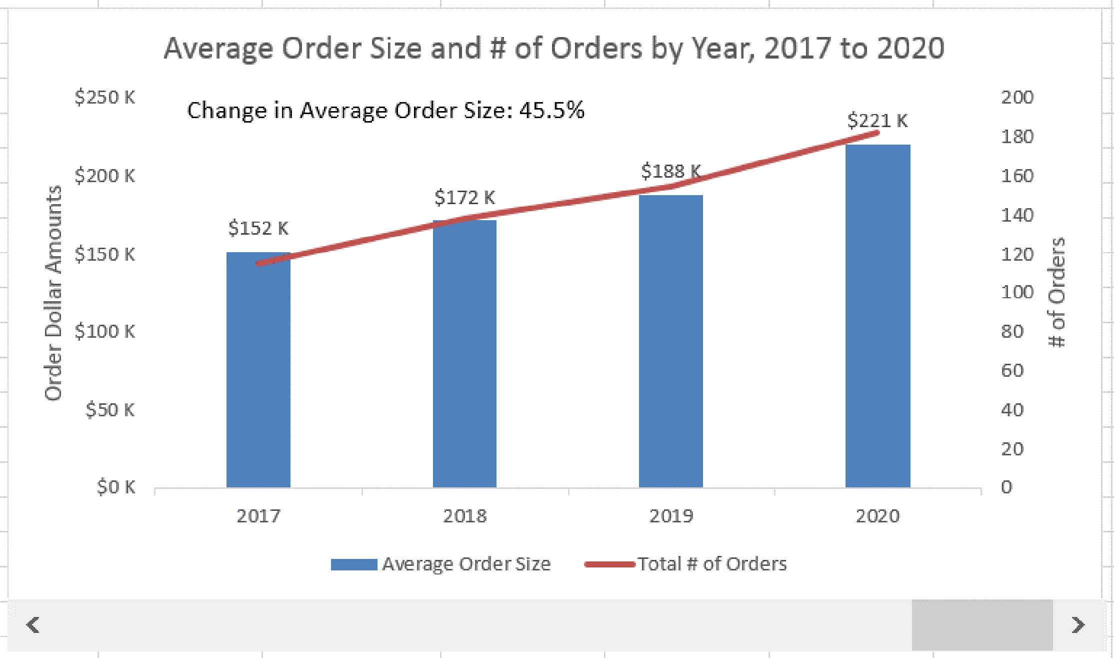
g. Create 4 separate rows for the 4 years of data you want to display at once, and use the INDEX and ROW functions to pull in data from the correct years based on the cell you linked the "Cell Link" field in the Scroll Bar to.

h. Right click the chart, go to "Select Data" and select the new, smaller table of 4 years you just created.

i. Press the Ctrl key and select both the Scroll Bar and the chart and then use the "Attach Objects" feature to ensure that they always stay together.

12. Continuing with the example above, you have now created a dynamic chart in Excel that adjusts the window and displays data for different years as you scroll through it. However, you run into a problem because the vertical axes of the chart change as you scroll through the data, making it difficult to compare the financial data over time. Here's an example of what happens at the minimum and maximum values:





How can you FIX this problem so that the same vertical axes are always shown, regardless of the data that is currently displayed in the chart?

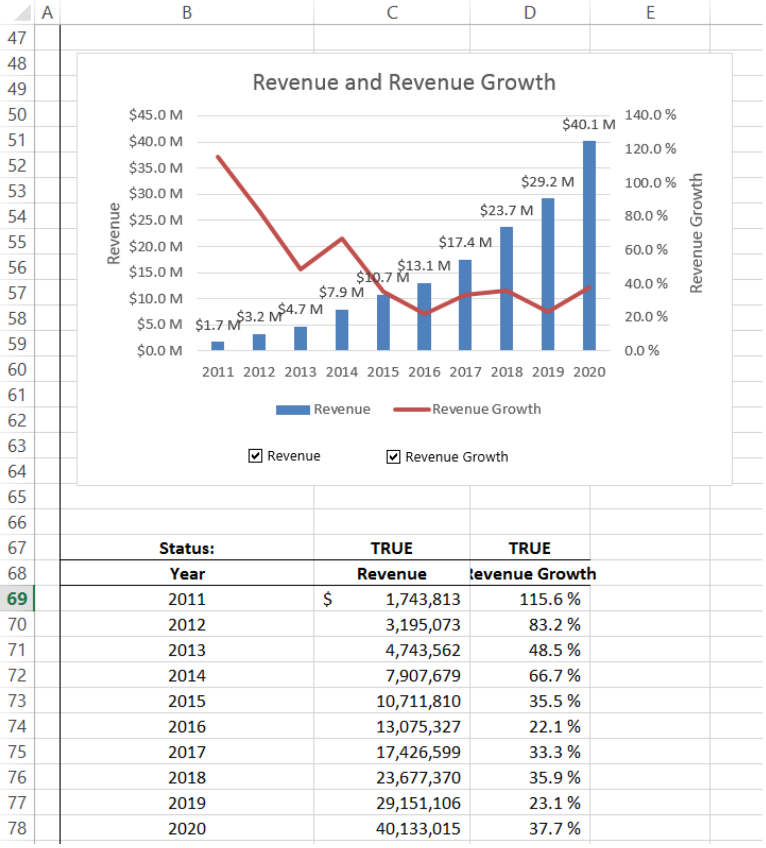
a. Click the chart and go to the Design tab or Layout tab and select the "Use Constant Vertical Axes" option.

b. Right click the Scroll Bar control, go to "Format Control" and then change "Minimum Value" and "Maximum Value" to the appropriate numbers.

c. Scroll to the maximum points in the data series, determine the maximum in the vertical axes, and then go to "Format Axis" for each vertical axis, disable the "Auto" option under the Bounds, and enter your own values under "Maximum".

d. None of the above – it's not possible to fix this problem in the general case, so you have to use a non-dynamic chart to get around it.

13. Consider the following dynamic chart, with two check box options that allow the user to hide or show revenue and revenue growth on the chart:



	B	C	D
67	**Status:**	**TRUE**	**TRUE**
68	**Year**	**Revenue**	**Revenue Growth**
69	2011	$ 1,743,813	115.6 %
70	2012	3,195,073	83.2 %
71	2013	4,743,562	48.5 %
72	2014	7,907,679	66.7 %
73	2015	10,711,810	35.5 %
74	2016	13,075,327	22.1 %
75	2017	17,426,599	33.3 %
76	2018	23,677,370	35.9 %
77	2019	29,151,106	23.1 %
78	2020	40,133,015	37.7 %

In both column C (Revenue) and column D (Revenue Growth), you are checking the cells at the tops of the columns to determine whether or not the corresponding check boxes are checked, and then you are displaying either "#N/A" using the NA() function, or blank values using "".

However, you are using a DIFFERENT formula for this in each column. For Revenue, the formula for the first cell is: =IF($C$67,P27,"") while for Revenue Growth the formula for the first cell is: =IF($D$67,Q27,NA()). P27 refers to the underlying source data for Revenue, and Q27 refers to similar underlying source data for Revenue Growth.

Why are you using "" for Revenue when the check box is not checked, but the NA() function for Revenue Growth when that check box is not checked?

a. It doesn't make a difference – this is strictly optional and it's just a stylistic preference.

b. We need to do that because Revenue is shown in a column chart, while Revenue Growth is shown in a line chart – there would be a big "0" line going across the bottom if we used "" instead of NA().

c. It's related to the fact that we have data labels for the Revenue columns, but not for the Revenue Growth line chart – if we had data labels in place, the "" would work and NA() would be unnecessary.

d. It's not possible to use NA() for the Revenue column charts because of the data labels – using NA() there would incorrectly show "#N/A" text all across the graph even if there are no columns drawn.

14. What is the MOST appropriate way to use custom keyboard shortcuts in Excel?

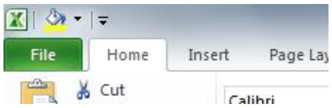
a. If you need to create a new function to error-check a common calculation in your model.

b. If there's a command you use frequently, but which has an inconvenient or non-existent keyboard shortcut.

c. If you need to apply formatting to a wide range of spreadsheets and data ranges, and there are a lot of repetitive actions required.

d. If you need to prompt the user to enter data in each cell and error-check the data as he/she enters it.

15. **As shown below, you've just added "Fill Color" to the Quick Access Toolbar. How can you access this custom keyboard shortcut?**



a. Ctrl + 1
b. Alt + 1
c. Alt + O + F
d. Ctrl + 1, Alt + F