

Data Visualisation and Insights

WEEK 3 – EXCEL DASHBOARDS – MACROS AND CONTROLS

Introduction

Last week we saw how to create Excel Dashboards using slicers and timelines. It is possible to add extra value to your dashboard by integrating macros and form controls.

There are numerous examples online of macro-enabled dashboards.

We will examine how to integrate macros and form controls into dashboards through a series of examples below.

Forms and Controls

Forms and Controls are a key feature of Excel that allows users to create highly interactive dashboards. We will explore how to customise dashboards using forms and controls through the exercises below. There are numerous controls you can use which are implemented in very similar ways.

You can find out more information at the following link:

<https://wmfexcel.com/2017/06/24/the-basic-of-form-controls/>

To begin you may need to enable the developer tab by selecting *Options* and *Customize Ribbon*. You may also need at this stage to adjust the macros settings by selecting *Trust Centre*, *Trust Centre Settings* and *Enable all macros*.

Introduction to Macros

Excel Macros can save you a ton of time by automating Excel processes that you use often. You can record macros in Excel to replicate tasks you need to perform regularly. But macros are actually quite limited and it is easy to make mistakes with the recording tool. Also, the recording process can be awkward.

Alternatively, using VBA(Visual Basic for Applications) to create macros gives you a great deal more power. You can tell Excel exactly what to do and how to do it. You also get access to a lot more functions and capabilities.

Macros are particularly powerful for controlling navigation and the look and feel of dashboards. We will explore their use through two examples below.

You can find out more about macros through the following link:

<https://powerspreadsheets.com/excel-macro-tutorial-for-beginners/>

To begin you need to enable the developer tab by selecting *Options* and *Customize Ribbon*. You may also need to adjust the macros settings by selecting *Trust Centre*, *Trust Centre Settings* and *Enable all macros*.

To record a macro you select *Record Macro* under the development tab and when you are finished you select *Stop Recording* under the same tab. A macro is also called a sub procedure.

You must save the file as a Macro-Enabled Workbook.

Exercise 1: Forms: Scroll Bar

The sales data from last week includes two categories, “Candy” and “Beverages” which are likely in the near future to be subject to a sugar tax. The company wishes to explore the impact of such a tax rate on the net sales for different sales people.

Using a scroll bar to set the sugar tax rate display a graph to show the sales per person net of any sugar tax. You should also include a timeline to allow the user to explore the impact over different time periods and a slicer to turn off/on certain salespeople. In building this dashboard you will need to include a net sales column in the data based on the output of the scroll bar.

The steps are as follows:

Step 1: Create a new column in the data table labelled NetSalesSugarTax. Assume a standard tax rate of 10% on Candy and Beverages.

Step 2: Create a separate sheet named Dashboard and add a pivot chart showing the net sales per employee.

Step 3: Include a Scrollbar control to adjust a tax rate value between 0 and 25% in steps of 1%. The output is accessed through the link cell.

Step 4: Adjust the formula in the NetSalesSugarTax column to be based on the tax rate from the linked cell.

Step 5: Add a slicer and timeline as described.

Step 6: While you can manually update the pivot table, you need to add a simple piece of code to force the table to update. This can be done through the following macro which is added by assigning a macro named Update to the scroll bar.

```
Sub Update()
```

```
    ThisWorkbook.RefreshAll
```

```
End Sub
```

Verify that the dashboard is working correctly.

Exercise 2: Forms: Data On/Off

It is also possible to control the data that is being shown in graphs and charts. For example, in the CA exercise you may wish to toggle between a dashboard that includes the 9/11 data and one that does not. In the sales data you may wish to examine the sales for transactions that are above or below 0.80. While you could use a slicer for this you could also do this through option controls as follows:

Step 1: Create a new column in the data table labelled SalesCSAT which has a simple formula to link to the Sales Data.

Step 2: Create a separate sheet named Dashboard and add a pivot chart named ChartSalesPerson showing the SalesCSAT per employee.

Step 3: Add a Group Box to the dashboard with two option buttons. Verify that these are working correctly. Label the options as “Satisfied Customers” and “Unsatisfied Customers”

Step 4: Set up a suitable hidden link cell using the format option on one of the option buttons.

Step 5: While you can manually update the pivot table, you need to add a simple piece of code to force the table to update. This can be done through the following macro which is added by assigning a macro named Update to both the Option buttons.

```
Sub Update()
```

```
    ThisWorkbook.RefreshAll
```

```
End Sub
```

Verify your dashboard is working correctly.

Exercise 3: Macro-Driven Menu System

We will build a simple menu system through the following steps:

Step 1: Create a simple spreadsheet with three sheets named “Menu”, “Graphs” and “Tables”.

Step 2: Record a simple macro to switch from the Menu sheet to the Graphs sheet. one sheet to another. Remember to Stop Recording immediately after you select the Graphs sheet. Examine the code to understand the code and logic.

Step 3: Add a simple button control in the Menu sheet and attach the macro developed.

Step 4: By creating multiple macros and amending the code create a simple menu system with two buttons on the Menu sheet and a single button on the Graphs and Tables sheet to return to the Menu sheet.

Exercise 4: Customise Dashboards

It is possible to use Macros to change the look and feel of a dashboard through relatively simple code structures. We do this by hiding and showing various elements on the dashboard.

Follow these steps to create an example:

Step 1: Using the standard data create a new workbook with the data sheet and two graphs which show the total sales by SalesPerson and one to show the total sales by Category. Name these graphs ChartSalesPerson and ChartCategory and place them in a sheet called Graphs.

Step 2: Create a simple macro to hide the SalesPerson graph. We will place the SalesPerson graph on top of the Category graph so there is no need to hide the Category graph. The following code may be helpful

Worksheets("Graphs").ChartObjects("SalesPerson").Visible = False

Step 3: Create a new macro to show the SalesPerson graph.

Step 4: Create two buttons and attach one macro to each button. Name the buttons BtnShowSalesPerson and BtnHideSalesPerson. Label the buttons appropriately as "Show SalesPerson" and "Show Category".

Step 5: Place the SalesPerson graph on top of the Category chart to achieve the correct effect and test.

Step 6: Next you should place the buttons on top of each other and change the code to also show/hide the appropriate button. An example of the VBA code that might be required is as follows:

Worksheets("Graphs").Shapes("BtnHideSalesPerson").Visible = False

Additional Resources

You can use VBA to control almost any aspect of a dashboard. There are some good worked examples available at the following:

<https://analysistabs.com/excel-dashboards/creating-interactive-dashboards-using-excel-vba/>

<https://chandoo.org/wp/excel-dashboard-tutorial-1/>

<https://www.youtube.com/watch?v=pt6-Kp9LrbM>