**Lab Sheet 5**

**Advanced Data Analytics**

Ian Nabney

This worksheet covers more advanced ways of creating interaction within Tableau using parameters and parameter actions. Scattered through the sheet you will find some **questions in bold** for you to write down an answer. These questions should help you check your understanding of using Tableau and of analysing the results of the visualisation tasks. This sort of question will form the basis of the multiple-choice quizzes that will be part of the assessment of this unit in later labs.

# Parameters

## What are Parameters?

Parameters are dynamic variables that can be used to pass information to actions. A parameter is a workbook variable such as a number, date, or string that can replace a constant value in a calculation, filter, or reference line.

For example, you may create a calculated field that returns True if Sales is greater than $500,000 and otherwise returns False. You can replace the constant value of “500000” in the formula with a parameter. Then, using the parameter control, you can dynamically change the threshold in your calculation.

You can even create a *dynamic* parameter that’s set to automatically refresh its current value (to the result of a single-value, view-independent calculation), list of values (based on a data source column), or range of values. This will happen each time the workbook is opened and Tableau connects to the data source referenced by the parameter, or whenever you select **Refresh** from the data source’s context menu..

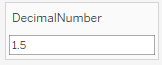
You can make your parameters more dynamic and interactive by using them in parameter actions. Parameter actions let your audience change a parameter value through direct interaction with a viz, such as clicking or selecting a mark.

1. In Tableau, open the Superstore data source and start a new worksheet. Click on the drop-down menu near the top of the Data tab (left-hand side of the window) and select ‘Create Parameter’. Give it the name ‘DecimalNumber’, make its type Float, and define the initial value to be 1.5.
2. You can specify what values of the parameter are allowed. You can select from the following options:

* **All**- The parameter control is a simple text field.
* **List**- The parameter control provides a list of possible values for you to select from.
* **Range**- The parameter control lets you select values within a specified range.

The availability of these options is determined by the data type. For example, a string parameter can only accept all values or a list. It does not support a range. Generally, we use All for most parameters.

1. There are also a large number of display formats available for parameters: for example, you can specify how many decimal places are shown.
2. When you save this parameter, it should appear towards the bottom of the Data tab in the list of Parameters.
3. Now click on the drop-down menu at the right-hand end of the DecimalNumber pill and select ‘Show Parameter’. You should see a small display in the top right of the view with the name ‘DecimalNumber’ and the value 1.5 (see Figure 1). Note that this version of the data already contains two parameters (‘Churn Rate’ and ‘New Business Growth’).

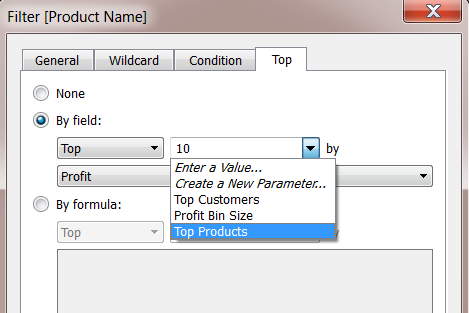


*Figure 1: Parameter display.*

Parameters give you a way to dynamically modify values in a calculation. Rather than manually editing the calculation (and all dependent calculations), you can use a parameter. Then when you want to change the value, you open the parameter control, change the value, and all of the calculations that use that parameter are updated. To use the parameter, you can either name it or drag and drop it onto the calculation editor.

For example, you can use a parameter in a filter to dynamically change its effect. Parameters give you a way to dynamically modify values in a Top N filter. Rather than manually setting the number of values you want to show in the filter, you can use a parameter. Then when you want to change the value, you open the parameter control and the filter updates. For example, when creating a filter to show the Top 10 products based on total profit, you may want to use a parameter instead of the fixed “10” value. That way, you can quickly update the filter to show the top 10, 20, or 30 products.

A list of parameters is available in the drop-down lists on the **Top** tab of the Filter dialog box. Select the parameter you want to use in the filter.

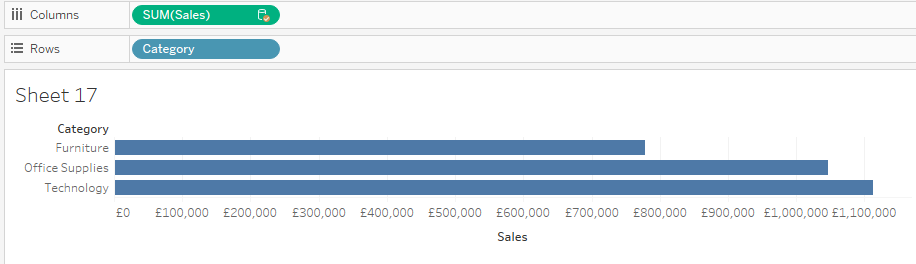


*Figure 2: Defining a filter threshold using a parameter.*

# Parameter actions: Drill-Down

We will first demonstrate the value of parameter actions through a relatively simple example of a context-sensitive drill-down into data.

1. Now click on the drop-down menu near the top of the Data tab (left-hand side of the window) and select ‘Create Parameter’. Give it the name ‘EmptyParameter’, make its type String, and remove the text from the initial value. When you save this, it should appear towards the bottom of the Data tab in the list of Parameters.
2. Drag the Category field into the Rows and Sales (from the Sample Superstore source) into the Columns. This should give you a horizontal bar chart as in Figure 3. To start to link the parameter with user actions, we first need to create a calculated field. Give it the name ‘Cat or SubCat’ and enter a conditional statement into its body:   
    IF [Empty Parameter] = [Category] THEN [Sub-Category]   
    ELSE [Category]   
    END



*Figure 3: Initial bar chart.*

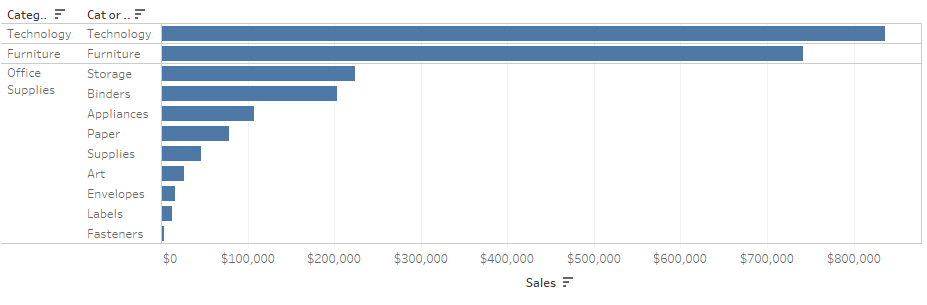
1. Put ‘Cat or Sub-Cat’ to the right of Category in the rows. You should see two columns to the left of the bars: one for Category as before, and a second one headed ‘Cat or Sub-Cat’ but with the same text rows beneath. Select the drop-down menu on the ‘Cat or Sub-Cat’ pill and choose Sort. Change the sort type to ‘Nested’ and ‘Descending’. Make sure that it is sorted on Sales with SUM as the aggregation and in descending order. Do the same thing for the Category. The order of the bars should change to have Technology at the top (because it has the most sales).

Graphical user interface, text, application

Description automatically generated

*Figure 4: Sorted bar chart.*

1. Now it is time to program in the interaction response. Select ‘Action’ from the Worksheet menu at the top of the window. Add a new action and define it to be a ‘Parameter Action’. Call the action ‘Update Empty Parameter’, have it run on Select, make the target parameter be ‘EmptyParameter’ and the Source Field ‘Cat or Sub-Cat’. Now if you click on the second column under Office Supplies, that group of bars should expand to show all the sub-categories while the bars for the other two categories remain the same (see Figure 4). **What happens if you click on the sub-category names?**



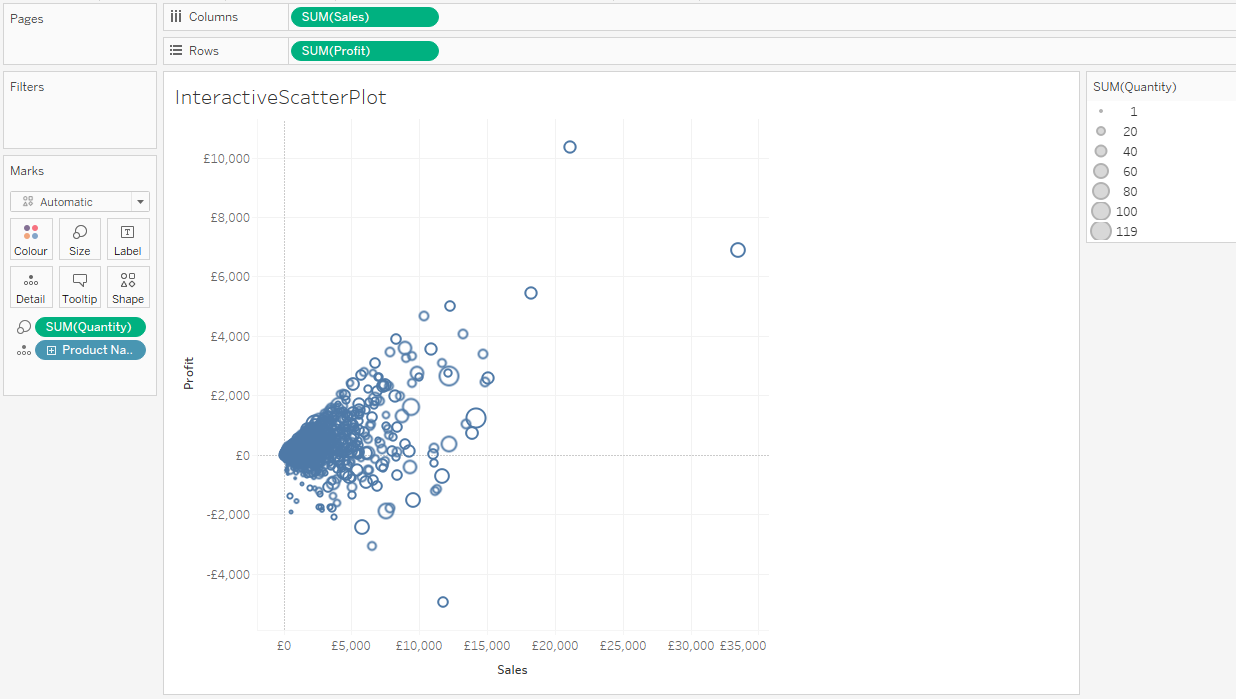
*Figure 5: Basic drill-down showing sub-categories of Office Supplies.*

1. The display is a bit untidy. So we can now modify the display to make it cleaner. Select the drop-down menu on the Category pill and uncheck ‘Show Header’. Right-click on the ‘Cat or SubCat’ header on the view and select ‘Hide Field Labels for Rows’. Click on the Label icon in the Marks palette and select ‘Show mark labels’. If you wish, you can change the font and style.

# Parameter Actions: Interactive Scatter Plot

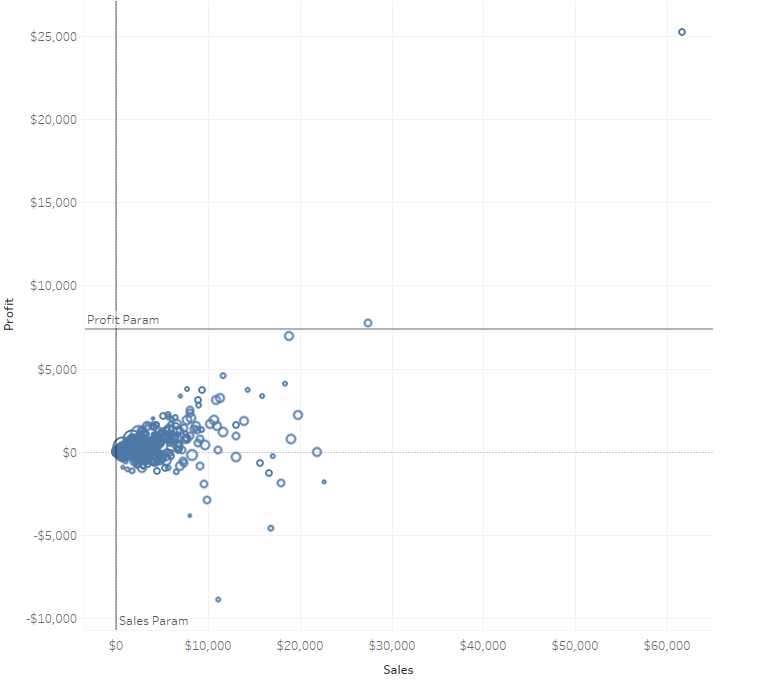
We will now explore a more sophisticated example of a parameter action to create an interactive scatter plot with some summary statistics.

1. Create a new worksheet (still with the Superstore sample data). Drag the Product Name field onto Detail. This generates a waffle chart. **Why do you think that this chart is called a ‘waffle chart’? What happens when you hover the mouse over one of the small squares?**
2. Now create a basic scatterplot by dragging Profit to the Rows shelf and Sales to the Columns shelf. Add Quantity to the Size mark to show how many items are sold for each category. It should like something like Figure 6.



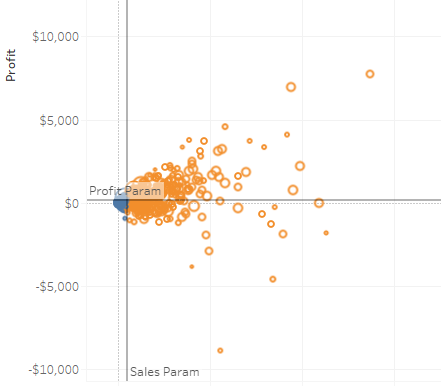
*Figure 6: Basic scatterplot.*

1. Add some more detail to the plot by selecting the context menu for Country/Region and Sub-Category and select ‘Show Filter’ for both of them. This is fine, and allows the user to look at different subsets of the data by selecting different filtered values. But it doesn’t help the user to quantify the differences between the subsets. This is where parameter actions will help.
2. Start by creating two new parameters ‘ProfitParam’ and ‘SalesParam’. Make them of float type and an initial value of 0. Next add a reference line to each axis. Right-click on the region just to the left of the axis line and select ‘Reference Line’. All you need to change is the Value drop-down menu: set it to ‘ProfitParam’. Do the same for the Sales axis and ‘SalesParam’. The reference lines should appear on the chart on the value 0. You can change their format (line width etc.) if you wish. You can prove that they can be controlled by selecting ‘Show parameter’ (as in Section 1) and changing the value in the display: the position of the reference line moves accordingly.
3. However, it rather cumbersome for the user to type in a number into a box and see the line move. What makes this setup more valuable is to move the lines in a way that corresponds to user selections. Create a new worksheet action on parameter change. Name it ‘ProfitParamChange’ and make the Target Parameter ‘Profit Param’, the field ‘SUM(Profit)’ and the Aggregation ‘Average’. Save this. Nothing has happened so far, but now select a small number of points (e.g. by clicking on the scatterplot and dragging the mouse or by clicking on multiple points with the control key held down). Once the points have been selected, the position of the profit reference line should now move to the average of the profits of the selected points. In Figure 6, I have selected two data points and the profit reference line clearly lies between them. When you deselect the points, the line stays where it is – a persistent record of the average of a selected group of points.
4. Now create a similar action for the sales reference line. Experiment with its use as in step 5.



*Figure 6: User-controlled reference line.*

1. You can reset the lines at any time by entering 0 into their display. We are now going to try out a bit of data analysis with this tool. Deselect all the sub-categories, and then select just binders. Select all the data points (e.g. using the control-A shortcut). The profit line should be at about 170 and the sales line at about 860 (hover the mouse over the corresponding lines). If you now bring in all the sub-categories, the non-selected data (all sub-categories apart from binders) are in a much paler gray. If you deselect the points (e.g. by clicking on the chart background), the lines stay in the same place and you can see these summary statistics for the binder data in the context of the whole dataset.
2. We can make this interaction more sophisticated. Create a Boolean calculated field (e.g. named ‘Colouring’) that tests if SUM([Sales])>SalesParam. Drag this field to the Colour mark. Every point which lies to the left of the sales reference line is blue and every point which lies to the right of the line is orange. You can easily change the colours from the drop-down menu for the display of the colouring on the right-hand side of the view.



*Figure 6: Colouring based on location relative to reference line.*

# Further Reading

There are many examples of the use of parameter actions to create dynamic interfaces in Tableau. If you want to know more, here are some training examples:

* <https://www.youtube.com/watch?v=YSGwaTuv-bA> (this lasts one hour).
* <https://help.tableau.com/current/pro/desktop/en-us/actions_parameters.htm#Examples> from Tableau itself.
* <https://www.tableau.com/about/blog/2019/5/parameter-actions-extend-interactivity?_ga=2.210455829.1463765256.1615859984-1788949159.1605716649>
* You can also create actions on dashboards: this blog post describes how <https://playfairdata.com/3-creative-ways-to-use-tableau-parameter-actions/>
* Lindsey Poulter’s Tableau profile <https://public.tableau.com/profile/lindsey.poulter8872#!/> has many inspiring examples of what can be done in Tableau to create dynamic interfaces to data.

# Challenge Exercise 1: Computing Z-scores

In statistics, the Z-score of an observation is the number of standard deviations that it is above or below the population mean. If the attribute has a Gaussian (normal) distribution, then we can calculate the probability of seeing an observation at least as extreme (i.e. at least as far from the mean). For example, 95.45% of observations lie within two standard deviations of the mean.

1. Load the Superstore sample dataset.
2. Create a calculated field ‘AverageSales’ to calculate average sales: WINDOW\_AVG(SUM([Sales])). In a similar way create another calculated field ‘STDEVsales’ to compute the sample standard deviation[[1]](#footnote-1) of the sales (you may need to look up the correct function name).
3. Create a third calculated field (SUM([Sales]) – [AverageSales]) / [STDEVsales] and call it ZScore.
4. Drag ZScore from the Data pane to Columns and Country/Region to Rows. The STDEVsales function is based on the WINDOW\_STDEV function, which is a table calculation function. The ZScore function, in turn, is a table calculation function because it includes STDEVsales in its definition: this can be seen from the small triangle (or capital delta) at the right-hand end of the pill. When you use a calculated field that includes a table calculation function in a view, it's the same as adding a table calculation to a field manually. You can edit the field as a table calculation, and we will do so in the next step.
5. The view pane shows an empty table with country names on the left-hand axis and a small grey pill stating ‘15 nulls’ in the bottom right corner. Fix this by clicking on the ZScore pill on the Columns shelf and selecting ComputeUsing-> Country/Region.
6. Now sort the bars in descending order. (Hint: look in the toolbar).
7. Now drag ZScore from the Data pane onto the colour mark and again onto the Label mark. You should see a graph like Figure 7.
8. The top two country (France) has a Z-score of more than 2: this means that it would be considered and unusual observation if drawn from a Gaussian distribution.
9. **Looking at the graph, what can you tell about the distribution of the data? Do you think it is likely to come from a normal distribution?**

**Chart, bar chart

Description automatically generated**

*Figure 7: Z-scores ranked by country.*

# Challenge Exercise 2: Correlation matrix

We are going to use what we have learned about joins and calculated fields to create a visualisation that is very helpful in data exploration – that is, the initial stage of a data analysis task when you are trying to understand the data better. We will create a visual display for the correlation matrix of a group of variables in a dataset. In terms of what we can learn by analysing the visualisation, this complements the scatterplot matrix that we covered in Exercise 4 from Lab Sheet 4.

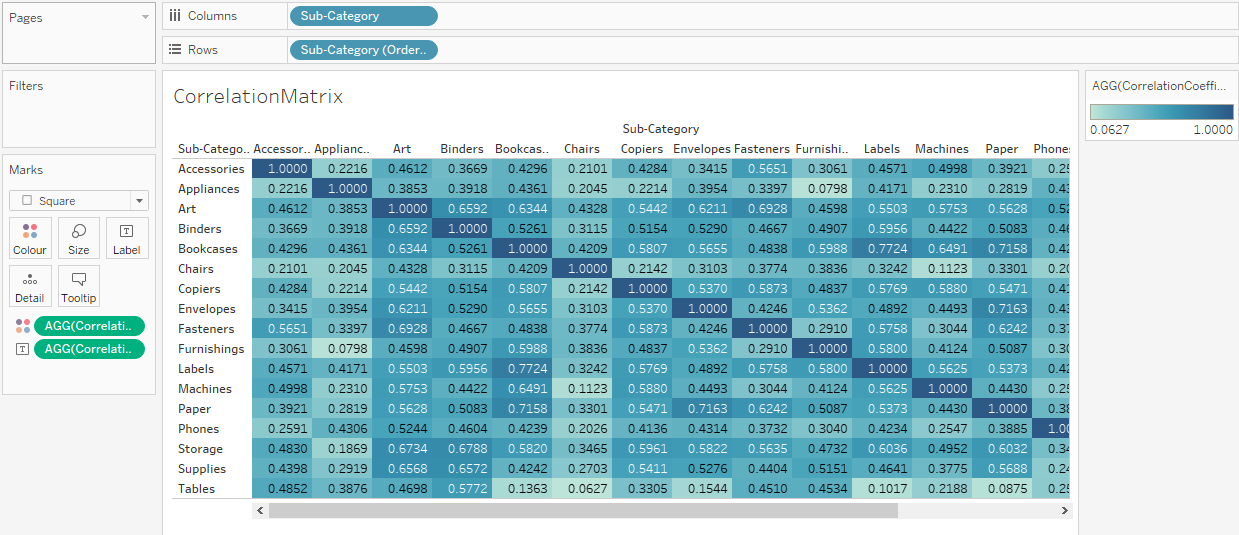
This exercise is based on <https://kb.tableau.com/articles/howto/creating-a-correlation-value-matrix> and there is a workbook attached to that page. If you prefer instruction through a video, then this link <https://www.youtube.com/watch?v=vhrmmvAvOfQ> provides a worked example (on a different dataset).

1. Start by closing the workbook and again loading the Superstore sample dataset in the Data Source tab. To compute the correlation **matrix** we need to compute the correlation of each variable with all the other variables. The best way to do this is through an inner join (to create the Cartesian product of the data with itself). We are going to look at the relationship between the sales of different categories: if a customer buys products from X sub-category, are they more or less likely to buy those from Y sub-category? The join clause should be on the field or fields that define the scope of comparison.
2. Switch the Data Source to the Sample data (I found it easiest to create a new worksheet and then select the sample data in the left-hand data pane before returning to the Data Source tab). Add the Orders table to the canvas. Select Open from the menu (right-click on the Orders table in the canvas) to open the Join canvas. Drag the Orders table to this layer (where it will be called Orders1) and create an inner join on Customer Name = Customer Name(Orders 1). It should look like Figure 8.
3. The second step is to create a field to contain the Pearson correlation coefficient. Create a new worksheet and select Analysis -> Create calculated field. Name the field “CorrelationCoefficient” and enter this formula:   
    CORR( { INCLUDE [Customer Name] : SUM( [Sales (Orders1)])}, { INCLUDE [Customer Name] : SUM( [Sales])})   
   Because we join along the Customer Name dimension, we are aggregating sales for each customer, which is correct for the task we are trying to carry out.
4. Now create a calculated field to filter the value. Select Analysis -> Create calculated field. We will create a simple filter: name it ‘NotSameSubCategoryFilter’.   
    [Sub-Category] != [Sub-Category (Orders1)]
5. Now build the view with the following steps.
6. Drag [Sub-Category] to the Columns shelf (remember that it can be found under Product).
7. Drag [Sub-Category (Orders1)] to the Rows shelf.
8. Drag the [CorrelationCoefficient] to Color on the Marks card.
9. Drag the [CorrelationCoefficient] to Text on the Marks card.
10. In the dropdown on the Marks card, change the Mark type to Square. The matrix should now look as in Figure 9. Note the values of 1.000 down the diagonal. **Why is this?**
11. Drag [‘Not same sub category’ filter] to the Filters shelf. In the Filter dialog, check True and click OK. **What happens to the correlation matrix? Why does this happen?**

Graphical user interface, application

Description automatically generated

*Figure 8. Inner join for the Cartesian product of the Orders table.*



*Figure 9: Initial correlation matrix after step 5(e).*

Of course, you can now improve this visualisation by naming the sheet, selecting a different colourmap etc. A diverging colourmap is a good choice since correlation varies between -1 and +1, and the zero value has an important interpretation. (Unfortunately, in this particular example, the data has changed since last year, and all the values are positive).

**Find the three largest positive correlations in the matrix. How do you interpret these results?**

1. There is a small difference in the denominator between population standard deviation and sample standard deviation but it would take too long to explain the difference here. [↑](#footnote-ref-1)