# Political Spectrum of Data Survey – Power BI Dashboard Step by Step

## 1.Power Query

### a. Raw Survey Data (Responses)

1. Get Data > New Source > Text/CSV > “ethics\_quiz\_responses\_LBAG\_demo\_pre-hashed.csv”
2. Filter **IPConv** column – does not equal *189da6f0dffb66d0368018cb603d9b41* (my hashed IP address used for form testing).
3. Transform>Run Python Script (for demo only). Add this script in after the first line provided (# ‘dataset)

import pandas as pd

import hashlib

def md5hash(x):

return hashlib.md5(x.encode('utf')).hexdigest()

dataset['IPConv'] = dataset.IP.apply(lambda x: md5hash(x))

1. Privacy Box – Ignore/Continue
2. Now you get a single-row Name/Value table. The Value row is your table – click on it (should be a yellow link), and it will expand out your table with the new Python-transformed column.

NB – for steps 3-5, you need Python installed on your computer (it’s not native within Power BI), as well as the Python libraries hashlib and pandas. Power BI treats the table you’re running the script on as a pandas Dataframe, with the variable name ‘dataset’.

1. Remove Column **IP**

### b.Question Lookup Data (QuestionLookup)

1. New Source > Text/CSV > “*QuestionLookup.csv*”
2. With **Question\_ID** column selected, Add Column From Examples>From Selection. Train the output column to be the number after “Q” at the start.
3. Rename this column “QuestionOrder”, and change type to Whole Number.

### c. Category Outcome Thresholds (OutcomeThresholds)

1. Home > Enter Data
2. Add 4 Values (in exact order stated) to Column 1: *Privacy Protector, Data Regulator, Savvy Sharer, Big Data Liberator*. Rename this column ***OutcomeGroup***
3. Add a new column and rename it to ***MinScore***, values: 0,24,48,71
4. Add a new column and rename it to ***MaxScore***, values: 23,47,70,93
5. Add a new column and rename it to ***PerAnswerMax***, values: 0.74,1.52,2.26,3.00
6. Add a new column and rename it to ***GroupOrder***, values: 1,2,3,4
7. Finally, Rename table “OutcomeThresholds”

Close and Apply – ready for the DAX-mobile…

## 2. DAX and modelling

Objectives:

* Respondents table that aggregates the data by IP addresses, calculating their average score, categorising them into ‘Protector/Regulator/Sharer/Liberator’ with a calculated column that looks up the **OutcomeThresholds** table
* Build a ‘cartesian product’ of the question lookup and 5 ‘Custom groups’, for user custom grouping and analysis.

1. Go to the Relationships Pane and check that Power BI has detected and created a single-direction, one-to-many relationship between **QuestionLookup**.QuestionID and **Responses**.Question
2. Go to Data pane, and create a new Calculated table with the following DAX:

Respondents = SUMMARIZE(Responses, Responses [IPConv],"Sum of Responses",SUM(Responses [Answer]),

"Count Attempts",DIVIDE(COUNTROWS(Responses),31))

This is the respondents table that will class each person into their categories based on the average sum of their ‘Strongly Disagree’/’Disagree’/’Agree’/’Strongly Agree’ responses – encoded as 0/1/2/3 respectively.

This calculated table will also filter the responses raw table to give question-by-question analysis from that.

1. Some respondents did the survey more than once – so before categorising, we need to weigh their total answer points by how many times they attempted. To do this, add a calculated column to this calculated ‘Respondents’ table as follows:

Weighted Sum of Responses = ROUND(DIVIDE(Respondents[Sum of Responses],Respondents[Count Attempts]),0)

1. Now we can use **Weighted Sum of Responses** to link to the **OutcomeThresholds** table with DAX calculated columns.
   1. Add a calculated column to **Respondents** as follows:

Outcome Category = CALCULATE(VALUES(OutcomeThresholds[OutcomeGroup]),FILTER(OutcomeThresholds,Respondents[Weighted Sum of Responses] >= OutcomeThresholds[MinScore] && Respondents[Weighted Sum of Responses] <= OutcomeThresholds[MaxScore]))

For each row in **Respondents**, this gets you a list of the unique OutcomeGroup column values in **OutcomeThresholds** table where the row’s Weighted Sum of Responses is between the MaxScore and MinScore for that OutcomeGroup row in **OutcomeThresholds**. Because the MaxScore/MinScore combinations are exclusive and non-duplicate, the list of unique values returned will always be just one value – this is what makes this technique work (it’s very useful for ‘binning’ data into ranges). If there’s a possibility that more than one unique value will be return, your calculated column will error, so bear this in mind.

1. Do the same as step 4 to get *GroupOrder* as well as *OutcomeGroup* into **Respondents**.

Outcome Category Order = CALCULATE(VALUES(OutcomeThresholds[GroupOrder]),FILTER(OutcomeThresholds,Respondents[Weighted Sum of Responses] >= OutcomeThresholds[MinScore] && Respondents[Weighted Sum of Responses] <= OutcomeThresholds[MaxScore]))

1. Let’s add a simple measure to the **Responses** table that averages the ‘Answer’ column:

Average Response = AVERAGE(Responses[Answer])

1. We have the answers as 0,1,2,3, but we also need them as text in the form of ‘Strongly Disagree’/’Disagree’/’Agree’/’Strongly Agree’ for legend purposes on the visuals. To do this, add a calculated column as follows to **Responses**:

AnswerText = SWITCH(Responses[Answer],0,"Strongly Disagree",1,"Disagree",2,"Agree",3,"Strongly Agree")

1. Column Orders: For the new **Responses.**AnswerText column in step 7, set it’s “Sort by column” property to be *Answer*. Also set this property for **OutcomeThreshold.**OutcomeGroup to be **OutcomeThreshold.**GroupOrder, and this property for **Respondents**.Outcome Category to be **Respondents**.Outcome Order.
2. Go back to the Relationships pane, and add a single-direction, one-to-many relationship between **Respondents**.IPConv and **Responses**.IPConv, so that the Respondents table can filter the **Responses** table of raw answers data, meaning we can analyse question patterns by response group.

That will do now for DAX and relationships – but we’ll be adding more soon as and when we need it to progress the dashboard.

## 3.Visuals

### Questions Analysis page

1. Make a page called ‘Questions Analysis’ with a black background. Add a 100% Stacked Bar chart visual:
   1. Axis=**QuestionLookup**.*QuestionOrder*
   2. Legend=**Responses**.*AnswerText*
   3. Values=Count of **Responses**.*Question* (Implicit Measure)
   4. Tooltips= **QuestionLookup**.*QuestionText,* **QuestionLookup**.*Category*
2. Remove background, set text colours to white and data colours to Red, Pink, Light Blue Dark Blue for Strongly Disagree/Disagree/Agree/Strongly Agree
3. Take off X-Axis, change Title to “Response Breakdown by Question (1-31)”
4. Add a table visual with two columns, **QuestionLookup**.Category column and the measure *Average Response*. Change the Total Label from “Total” to “Average” and add “By Question Category” as a Title. No background, white header text.
5. Add a clustered bar chart visual using the same columns: **QuestionLookup**.Category on the axis and *Average Response* measure as the values. White header fonts, background = black.
6. Fix the X axis scale to be 0.0 as min and 3.0 as max
7. Set Data Labels on with white font, light grey background. Set the Data colours as black, with same transaparency as background. This should hide them. Play around until the colours of page background, visual background and bars match. Set gridlines off (this is in the x-axis property)
8. Since we have 0 -3 as the x axis scale, we can over lay the 4 category thresholds as custom gridlines using Constant Lines. Go to Analytics (spyglass) tab, and add 4 Constant lines as follows:
   1. Dark Red at 0.74 – name this ‘Protector’(needs a triple click)
   2. Dark Red at 1.52 – name this ‘Regulator’
   3. Dark Red at 2.26 – name this ‘Sharer’
   4. Dark Red at 3.0 – name this ‘Liberator’

Use the same colour schema as the Disagree/Disagree/Agree/Strongly Agree Legend in Step 2. Notice that the constant line values selected are the ones from the *PerAnswerMax* column in the **OutcomeThresholds** table. Unfortunately no way to automate this, but suggestions welcome!

1. Final step for this page is a slicer for the Respondents’ Response categories, so that we can filter the average responses by responses among Liberators vs Regulators and see how that impacts the subject categories.
   1. Add **Respondents**.Outcome Category as slicer.
   2. Formatting>General>Orientation: Switch from Vertical to Horizontal. Arrange so that it’s 1 row 4 columns.
   3. Import ‘Data Spectrum’ image and line up with slicer

### Questions Analysis – User Defined Groups page

1. Great! That’s the question analysis done, duplicate this page, and rename it to be “Questions Analysis – User Defined Groups”. This will be the same, but we’ll add a slicer pane and do some work on the model and relationship to allow users to set their own groups of questions, independent of the provided categories of Health, Crime, Transport, Business etc.
2. This is the bit that’s going to allow users to (as far as possible) replicate custom grouping in Excel pivot tables. Firstly, remove the 100% stacked bar chart from this page.
3. We need to work on the model to set up the custom grouping option. Go to the Data pane and add a new table called **QuestionsCategorisable.** Build it with the following DAX query:

QuestionsCategorisable = CROSSJOIN(QuestionLookup,DATATABLE("CustomGroupName", STRING, {{"CustomGroup1"},{"CustomGroup2"},{"CustomGroup3"},{"CustomGroup4"},{"CustomGroup5"}}))

This is cross-joining the question LookupTable with an “on the fly” DAX-generated table of 1 column and 5 rows text values of CustomGroup1-5. This will basically repeat the rows in **QuestionLookup** 5 times, one for each unique value in the on-the-fly table (of which there are 5). This is known in SQL as a Cartesian Product.

1. Now go back to the relationships pane and make a many-to-many relationship between this new table and the **Responses** raw survey fact table. Ignore the warning. This should be single direction so that the new table can filter the **Responses** table, not the other way round.
2. Now back to the Visual pane to the User Defined Groups page. For both the table and the bar chart visuals, replace **QuestionLookup**.Category with **QuestionsCategorisable.**CustomGroupName. Ignore the numbers.
3. The numbers are 1.09 for all, because all5 custom Groups show the average of all 31 answers. This is because in the new table, each category value has 1 row for each of the 31 questions, as per the cartesian product. The next step is to use slicers and bookmarking to give the report view the chance to ‘custom categorise’ by taking questions out of each Custom Category.
4. Add a text box in the left middle with “**This page allows you to group questions into your own categories using the button above and compare their average response scores. You can make up to 5 custom groups.**”
5. Add a bookmark button top left. When clicked, this will get the user to open a slicer pane with which they can customise the groups, so give it some text to describe this.
6. Add a new column to the table in the top right – Count (Distinct) of **QuestionsCategorisable.**Question\_ID (implicit measure). Rename the header “# Questions Selected”
7. View > Bookmarks. Add 2 Bookmarks: “WithSlicer” and “NoSlicer”. Remove “Data” from the ticked options on both. This makes sure that the data selected by the users (by their slicer choices) in one bookmark state stays the same in the other bookmark after switching, which is what we want so that the slicer pane works.
8. Go to the button top left, and set its Action to be Type=Bookmark, Bookmark=”WithSlicer”
9. Add a Rectangle shape almost as wide and high as the page.
10. Add a Back button over this shape, top right, and set its Action to be Type=Bookmark, Bookmark=”NoSlicer”
11. Add a slicer on top of the rectangle shape, and add 2 fields to it: **QuestionsCategorisable.**CustomGroupName, then **QuestionsCategorisable.**QuestionText**.** This creates a hierarchy effect.
12. Now, the user can theoretically assign questions to ‘custom groups’ himself (using as many or as few custom groups as they like), and use it to alter the average scores. But, they have no way to check they haven’t added the same question in 2 groups…

So, add a matrix visual over the rectangle with **QuestionsCategorisable.**Question\_ID as row value, **QuestionsCategorisable.**CustomGroupName as column value, and an implicit measure to provide a count.

Take off the row total, size so it fits in with all 5 custom groups and the column total, and add conditional formatting highlighting column total greater than 1. (Do this in the cond formatting for your implicit measure using Rules setting on Totals only). These represented duplicate selections.

1. Now we need to configure the bookmarking so that the rectangle, back button, slicer hierarchy and audit matrix only show in the ‘WithSlicer’ bookmark.
   1. Bring the selection pane into view
   2. Mark the rectangle, back button, slicer hierarchy and audit matrix in the Selection pane as Hide. This should put a line through the eye icon next to these 4.
   3. Immediately after doing this, click the 3 dots to the right of “NoSlicer” in the Bookmarks pane and select update. Now toggle between The 2 bookmarks on this page to check that the slicer items disappear and reappear. Test the 2 buttons with Ctrl + click
2. Great – your users can now compare scores for their own groups!
3. As a bonus, how about a tooltip on the bar chart showing a list of the questions selected for each group?
   1. Add a new measure to the **QuestionsCategorisable** table as follows:

QuestionList = CONCATENATEX(DISTINCT(QuestionsCategorisable[QuestionText]),QuestionsCategorisable[QuestionText], UNICHAR(10) & "-----" & UNICHAR(10))

This concatenates all the unique questions in the table into a string, with each unique value separated by line breaks and a dashed line. The filter context will evaluate the concatenation for each custom group when used on a chart or table.

* 1. Add this measure to the tooltips for the bar chart on this page.

### Main Page/WaffleChart

1. Duplicate the “Questions Analysis” page and remove all visuals except for the “DataSpectrum” image. Rename the page “Main Page”
2. Import the **waffleChart.2.2.0.0** custom visual into the model from the Visualisations pane. (Import custom visuals from file option)
3. Add this new visual to the page, selecting **Respondents**.Outcome Category as the Category Data and Count (Distinct) of **Respondents**.IPConv as an implicit measure for the Values.
4. This looks ok, but the % breakdown figures are wrong – this is an inherent problem with the waffle chart defaults. To correct this, 2 steps:
   1. Add a new measure to the **Respondents** table as follows:

MaxWaffle = CALCULATE(COUNTROWS(Respondents),ALL(Respondents))

This measure will use the ALL function to return a count of all respondents, ignoring any filters selected by the user or created by the category splits on a visual, which is what we want here.

1. Add this new **MaxWaffle** measure to the Max Values option on the waffle chart. Make sure the background is off and the visual is sorted ascending by Outcome Category.

BONUS

This looks good, but you can further customise a Waffle chart in Power BI using its **Path** option – this allows you to replace the circles in each 10x 10 category grid with something more befitting the breakdowns you are visualising…

How about padlocks for the Protectors, unlocked padlocks for the Regulators, Handshakes for the Sharers, and Keys for the Liberators?

To do this, we need to get SVG (Scalable Vector Graphic) paths from an icon library.

<https://iconify.design/icon-sets/?query=lock>

For each, the bit we need is the ‘d’ property in the embeddable SVG code for the icons we want. This is a long combination of letters, numbers and spaces that represents the line drawing. We’ll build the d codes for each category into a SWITCH statement, and then add that as a calculated column to the **Respondents** table as follows:

SurveySVG = SWITCH(Respondents[Outcome Category],"Privacy Protector","M704 384h-32V262q0-111-72.5-186.5T415 0q-74 0-133 35.5t-90.5 95T160 262v122h-32q-53 0-90.5 37.5T0 512v384q0 53 37.5 90.5T128 1024h576q53 0 90.5-37.5T832 896V512q0-53-37.5-90.5T704 384zM224 262q0-84 53-141t138-57t139 56.5T608 262v122H224V262zm544 634q0 17-8.5 32T736 951.5t-32 8.5H128q-26 0-45-19t-19-45V512q0-26 19-45t45-19h576q26 0 45 19t19 45v384zM416 576q-27 0-45.5 18.5T352 640q0 36 32 55v105q0 13 9.5 22.5T416 832t22.5-9.5T448 800V695q32-19 32-55q0-27-18.5-45.5T416 576z",

"Data Regulator","M19 7h-2a4 4 0 0 0-8 0v3h9a3 3 0 0 1 3 3v6a3 3 0 0 1-3 3H6a3 3 0 0 1-3-3v-6a3 3 0 0 1 3-3h1V7a6 6 0 1 1 12 0zm-1 5H6a1 1 0 0 0-1 1v6a1 1 0 0 0 1 1h12a1 1 0 0 0 1-1v-6a1 1 0 0 0-1-1z",

"Savvy Sharer","M21.71 8.71c1.25-1.25.68-2.71 0-3.42l-3-3c-1.26-1.25-2.71-.68-3.42 0L13.59 4H11C9.1 4 8 5 7.44 6.15L3 10.59v4l-.71.7c-1.25 1.26-.68 2.71 0 3.42l3 3c.54.54 1.12.74 1.67.74c.71 0 1.36-.35 1.75-.74l2.7-2.71H15c1.7 0 2.56-1.06 2.87-2.1c1.13-.3 1.75-1.16 2-2C21.42 14.5 22 13.03 22 12V9h-.59l.3-.29M20 12c0 .45-.19 1-1 1h-1v1c0 .45-.19 1-1 1h-1v1c0 .45-.19 1-1 1h-4.41l-3.28 3.28c-.31.29-.49.12-.6.01l-2.99-2.98c-.29-.31-.12-.49-.01-.6L5 15.41v-4l2-2V11c0 1.21.8 3 3 3s3-1.79 3-3h7v1m.29-4.71L18.59 9H11v2c0 .45-.19 1-1 1s-1-.55-1-1V8c0-.46.17-2 2-2h3.41l2.28-2.28c.31-.29.49-.12.6-.01l2.99 2.98c.29.31.12.49.01.6z"

,"Big Data Liberator","M57.6 6.4a15 15 0 0 0-23.8 17.7L13.6 44.4l-5.9-6L2 44.1 8 50l-4.7 4.7a4.3 4.3 0 1 0 6 6l30.6-30.5A15 15 0 0 0 57.6 6.4zM51.9 22a7 7 0 1 1 0-9.9 7 7 0 0 1 0 9.9z")

Then, add this new calculated column into the Path option for the Waffle chart, and the circles will switch to your selected icons.