In this session you will learn about Tableau as a technology for supporting visual data analysis. You'll use the Brexit and 2011 Census data aggregated to LA-level from the previous sessions.

By the end of this session you should be able to:

- connect Tableau to flat data files
- demonstrate command of the the Tableau interface, particularly:
  - *Dimensions* and *Measures*
  - filters
  - groupings
  - dashboards with linked views
- use Tableau functionality, and your understanding of plot grammars, to generate data-rich information graphics

A print version of this document can be downloaded from [this link].

What is Tableau?

Tableau is a commercial software tool for visual data analysis. It has roots in academia; its research department has contributed several notable papers to the *InfoVis* discipline in recent years. And those of you with an interest in Data Visualization may well have come across Robert Kosara, of Tableau Research, and his highly influential blog. What makes Tableau distinct from other data analysis tools is the very heavy emphasis on the *visual* perspective in data analysis.

Getting Tableau

Tableau is commercial software — so is not free to use and in fact is quite expensive. However, great for you: it is free to students and university staff. The University of Leeds also has an institutional licence — so it is fully installed on the machines in the labs. It also makes sense to have a version of Tableau Desktop running on your own machines. You can do this by following the links below.

- register for a student key
- download Tableau desktop

Why Tableau?

Since the company was founded in 2003, Tableau has seen rapid growth and is widely used in industry (e.g. companies list). Whilst Tableau has its frustrations, underpinning its design and layout are key tenets of visualization design: of data types and their mapping through visual variables. In a similar way to `ggplot2`, then, Tableau forces its users to consider the visual grammar behind their graphics and data analysis. Finally, although not open and free to use, Tableau is sustained by a very large community of users, which is cultivated by Tableau through its Zen Masters programme. Should you wish to develop some expertise in Tableau, you may find their MakeOverMonday useful. Rob Radburn, a UK-based *Zen Master*, has posted some great examples with an implied *geographic* flavour.

Why not Tableau?

Tableau is a software tool rather than programming language. It therefore relies on point-and-click interactions, making reproduction of workflows problematic. As with all software tools, it can be slightly idiosyncratic — you need to understand/convert your thinking into a Tableau way of organising data. You may at first find particularly confusing the means through which data are aggregated and grouped in Tableau. Related to this, there is a layer of abstraction between the user and dataset. Tableau does not offer much support for data cleaning and since aggregation and summarisation tends to be performed by Tableau automatically, a user may not know quite what a plot is showing — you'll discover this for yourselves. Despite these problems, Tableau is a very accessible software tool — some claim that it is *"democratising"* to the extent that it takes business intelligence away from IT departments and into the hands of decision-makers. Crucially, and probably uniquely, it is a genuine, off-the-shelf interactive visual data analysis tool.

How the Tableau display is organised

Data handling

As with R's `data frame` or `tibble`, Tableau works with tabular data — where rows are populated with observations and columns with variables that describe observations. Once loaded into Tableau, data are automatically organised into *Dimensions* and *Measures* (left margin of Figure 1). *Dimensions* are typically categorical variables used for grouping and pivoting data, which might be achieved via faceting to form small multiples or through colour hue, shape or other visual channels. *Measures* are quantitative (numerical) variables and mapped to size, colour and other visual channels. As of Tableau `--version 10.2`, spatial data types are supported. Interestingly, they are handled in the same way as in R using the `SimpleFeatures` package. Geometry information is converted and stored in a variable called `Geometry`, each element of which contains a list of type `MULTIPOLYGON`.

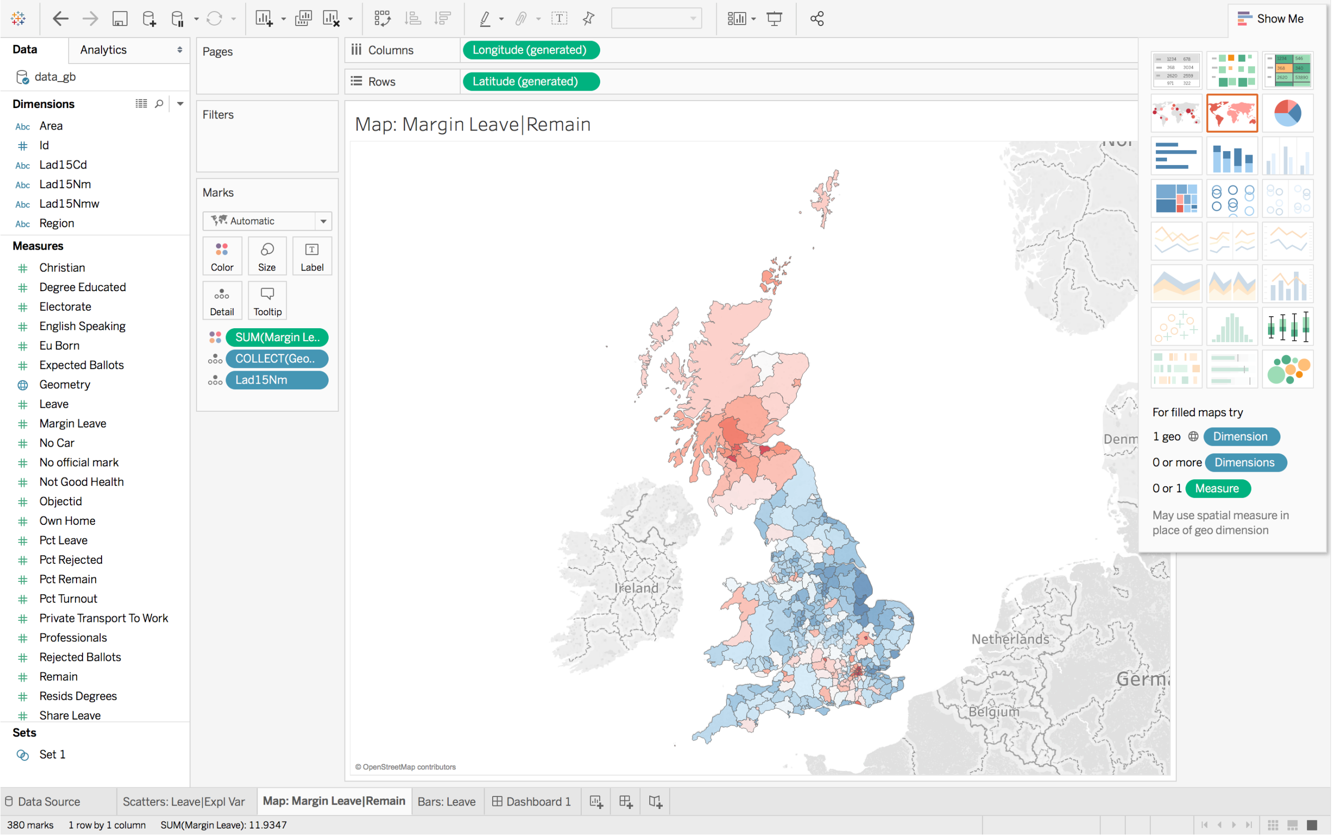
Windows

At the top of Figure 1 are the *Columns* and *Rows* shelves. These can be loosely thought of as the x-position and y-position for your charts in Tableau.

In the second margin of Figure 1 is the *Marks* window. This provides access to the numerous visual channels to which data can be mapped.

You will soon discover that Tableau aggregates data according to the configuration provided to *Rows*, *Columns* and *Marks*. You will often wish to disaggregate, and to do so you will need to drag an attribute to the *Detail* icon (under *Marks*).

Figure 1: The Tableau user interface.



Columns: Longitude (generated)

Rows: Latitude (generated)

Marks: Automatic

SUM(Margin Le..

COLLECT(Geo..

Lad15Nm

380 marks 1 row by 1 column SUM(Margin Leave): 11.9347

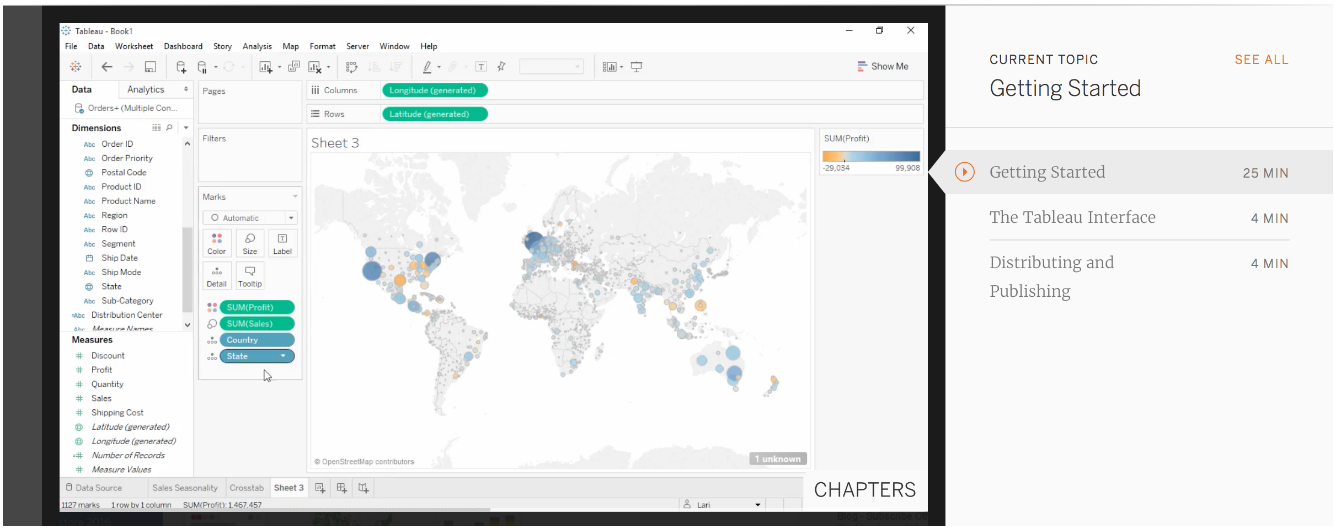
Task 1: Listen and explore

Perhaps the best means of introducing Tableau is through the introductory tutorial provided by Tableau themselves. The tutorial can bebe accessed from this link. Note that you will need to sign in to access the tutorial.

Instructions

Follow the instructions in the c.20-minute video. Use Tableau Desktop to open the data and generate the story as described in the video. Do pause the video to try some of the techniques out. Ignore the Chapters: `Connecting live versus extracting, 03:28`, `Story Points, 22:06`, `Distributing Content, 23:35`.

Figure 2: The Tableau tutorial.



Task 2: Connect to data and familiarise

After completing the R sessions, you might have considered yourself free from writing any code. Whilst that is generally the case, in this session you will be returning to the Brexit dataset and using Tableau to query and explore the data and relationships generated in the previous session as part of the modelling activity.

```
# Convert back to WGS84 (EPSG:4326) for use by Tableau.
data_gb <- st_transform(data_gb, crs=4326)
st_write(data_gb, "data_gb.geojson")
```

Instructions

Open your saved R session from last week's practical. Add the code block to your R script and Run.

You should now have in your R directory a GeoJson file containing the Brexit results, Census variables and model outputs from the previous session. Next, load these data into Tableau by following the instructions below.

Instructions

Click the Tableau icon in the top left margin (`Show Start Page`). You will connect Tableau to a flat file rather than database server, so under `Connect → To a file` click `More…` and navigate to the file you exported from R: `data_gb.geojson`.

You should now see a screen that resembles a spreadsheet and should not be too different from the screen that you see when typing the command `View(<dataframe-name>)` in RStudio. Notice that Tableau has automatically attributed data types to the columns (variables) in this spreadsheet view and that this is also true for spatial data — the `Geometry` column has been annotated with a globe-type icon.

Instructions

Click the `New Worksheet` icon in the bottom margin of your Tableau display screen.

The *Worksheet* view should be familiar to you from the introductory video. You may wish to take a moment to check that you are happy with the way in which Tableau has automatically allocated variables to the *Dimensions* and *Measures* panes. To re-allocate a variable, right click and select `Convert to <Dimension|Measure>`.

Task 3: Reproduce (your Brexit vis)

Tableau is supposed to be an exploratory tool where you as a researcher rapidly explore different variable combinations and visual mappings in order to generate new *views* on a dataset and new *insights*. While R requires some precision and consistency in the syntax used to perform analysis, Tableau does not. Rather than a very prescriptive set of instructions to follow, the rest of the practical will require some creative thinking from you around generating graphics and the mapping of data to visual channels.

Map

Instructions

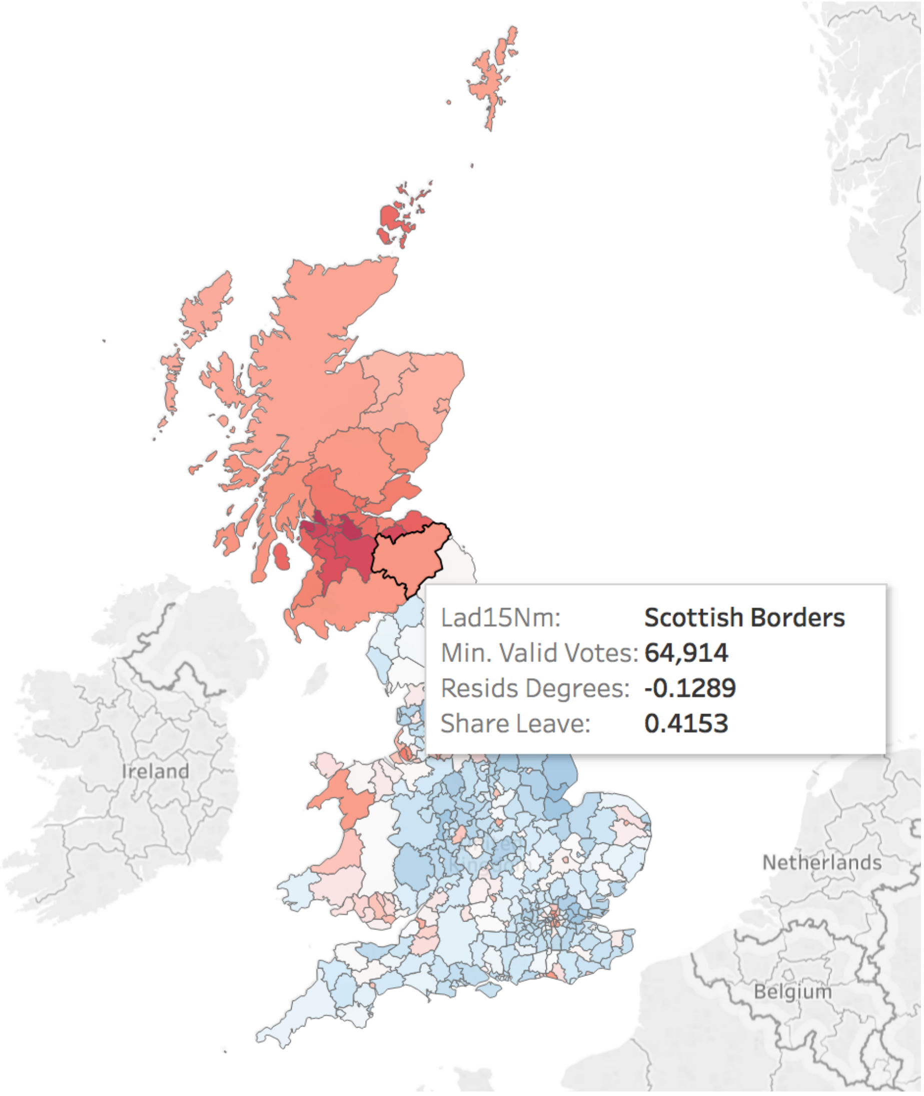
Start a new worksheet and rename it `Map: Margin Leave|Remain`. Drag `Longitude` to the *Columns* shelf and `Latitute` to *Rows*. Drag `Geometry` to *Detail* under the *Marks* pane. Drag `Margin Leave` to *Colour*. Tableau has automatically aggregated this variable over all Local Authorities, using `SUM` in this case. Disaggregate by dragging `Lad15Nm` to *Detail*. You will notice that Tableau has seen that the `Margin Leave` variable crosses zero and automatically applied a diverging colour scale — and a Brewer one at that. Nice! You may wish to alter this colour scale to choose a different scheme and also to change the min/max range to ensure the scheme is symmetrical. These parameters can be set by clicking `Color → Edit Colors`.

Individual task

Generate a similar map, but this time to display the residuals from your univariate regression model. You will have noticed that Tableau automatically allows tooltip-level interaction. Edit this such that the tooltip not only displays the LA-name and residual value, but also the value of the `Share Leave` variable. Your map should resemble that presented in Figure 2.

As discussed earlier, Tableau was developed on the back of academic research in the InfoVis discipline. As a result, it is underpinned by key tenets of data visualization design. You saw this when associating colour to the `Margin Leave` variable. Tableau recognised not only that this was a quantitative variable, requiring a quantitative colour scheme, but that the variable crossed zero and therefore that a diverging scheme was necessary. What do you think would happen if you were to instruct Tableau to colour not by `Margin Leave` but `Region`? Test this by dragging `Region` to *Color*.

Figure 3: Map of residuals.



Scatter

Individual task

Create a scatterplot displaying a selected Census variable against `Share Leave`. Think about how you might map your data to visual channels —the plot grammar—in order to maximise the information-carrying capacity of the graphic.

Bar

Individual task

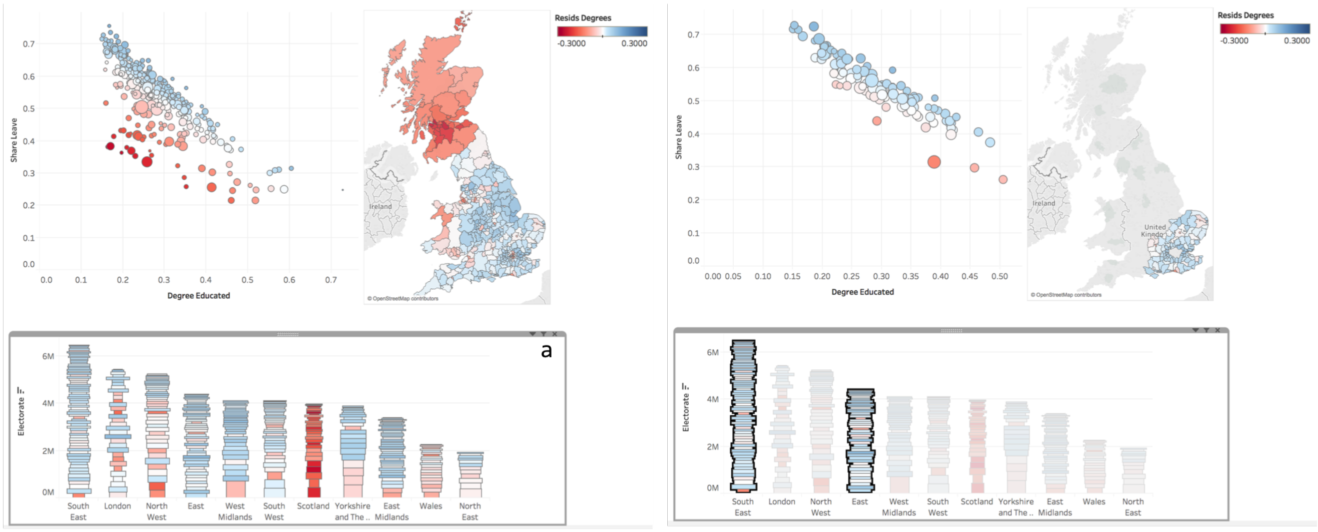
Create a barchart displaying some data aggregated by region. Again, think about how you might map your data to visual channels —the plot grammar — in order to maximise the information-carrying capacity of the graphic.

Task 4: Interact

Individual task

You may have noticed the `New Dashboard` icon in the bottom margin of your Tableau screen. Click on this to create a new dashboard. In the left margin are the list of *Worksheets* you previously generated. Drag these into the centre screen and assemble. You may find it easier to organise chart objects on the dashboard by making those objects *floating*. This can be achieved by right-clicking on the chart and selecting `floating`. An important feature of Tableau's *Dashboard* environment is the support for linking and brushing (Becker & Cleveland 1987) of views. To link views in a dashboard, right click and select `use as filter`.

Figure 4: Example dashboard displaying residuals from a linear model regressing share of Leave on degree-educated.



	data channel	visual channel	encoding type
plot a	region size	box height	continuous – height scaled to electorate
plot a	region size	1d position	ordinal – regions ordered by electorate
plot a	la size	height	continuous – height scaled to electorate
plot a	la share leave	width	continuous – width scaled to share leave
plot a	la residuals direction	colour hue	categorical – negative/positive
Plot a	la residuals size	colour value	continuous – value scaled to resid size

Assessed Task

This is a short, assessed task. It does not assume knowledge or skills above what you have learnt in the session. Ideally, the task should be completed **within** the workshop session — the aim is not to burden you with additional work.

The task is designed to assess your:

- ability to produce outputs in Tableau
- understanding of data types and their encoding through statistical graphics

You can quickly glance at the assessed task below. However, the document into which you'll need to upload your answers can be found on *Minerva*, under this module (*GEOG5022M*), then *Learning Resources*. Click on this week's folder (`Week 8 - Tableau`). You should see a word document called `PD_Workshop_5_Tableau.docx`. Download this document to a local directory — this is the document you will use to paste in your answers. Once you've completed the task, save the document using the filename `PPD_R_<StudentID>`. Upload the completed document to *Turnitin* — again, a link is provided under `Week 8 - Tableau`.

Assessed task 1. *Upload*

> Take a screenshot of your dashboard created as part of task 4 for this week's session. Then paste to `PD_Workshop_5_Tableau.docx`.

Assessed task 2. *Interact*

> Your dashboard should contain some linked views as per this week's session. Hint: you can link views in a dashboard by right-clicking on a chart element and selecting `use as filter`. To provide evidence that you have successfully linked views, perform a filter and create a screenshot of your dashboard whilst that filter is applied (as in the right of Figure 4). Paste the screenshot into `PD_Workshop_5_Tableau.docx`.

Assessed task 3. *Annotate*

> Select **one** chart element and annotate with the visual grammar — the mapping of data to visual channel — used to construct it. Format your annotation in the same way as Figure 4: `<data-channel> <visual-channel> <encoding-type>`.

Data Challenge (post assessed task)

Over the last decade, *data journalism* has emerged as an important discipline in and of itself. Several notable publications have developed some impressive data analysis and visualization competency:

- Check out Alan Smith and Martin Stabe, of the FT and their visual vocabulary;
- Or the incredible data vis at the New York Times.

The Guardian DataBlog has over the last 5-10 years been a great resource for data journalism pieces, but also it has made great efforts to provide access to the datasets that underpin stories. In this *optional* activity — to be completed if you have finished and uploaded your answers to the assessed task — you will use data provided by the Guardian DataBlog.

Individual task

> Download a dataset from the Guardian DataBlog and perform some data analysis and visualization in Tableau. You may choose to reproduce the visualizations already published on the dataset, either from the original Guardian article or published elsewhere online. Or you may wish to create an entirely new visualization, emphasising an underexplored aspect. A slightly ageing, though comprehensive, list of datasets for download can be found here. Typically, the datasets are provided as a Google Spreadsheet. They can be exported by selecting `File`, `Download as Text`.

Further reading

- Munzner, T. (2014) Visualization Analysis & Design, CRC Press. Chapters 2, 5. Available as an e-book via the University of Leeds Library.