This week's session covers the fundamentals of Data Visualization. The lecture introduces some key research from cognitive science, computer science and the geo-sciences that will help you interpret and create information-rich data graphics. In this lab you will apply these principles to create effective data graphics for exploring spatial structure around a current, though somewhat depressing, topic: the 2016 Brexit vote.

During the lab you'll be generating graphics using the `ggplot2` library. `ggplot2` is used widely by data analysts in academia and industry. `ggplot2` has a very strong theoretical underpinning, based on a framework for data visualization known as *The Grammar of Graphics* (Wilkinson 2005) — we covered this a little during the lecture. The general approach is of treating graphical elements separately and building features in a series of layers. Much excellent data visualization is produced by journalists working in R and `ggplot2`: check out this on the BBC's use of `ggplot2` and Jon Burn Murdoch's work at Financial Times.

By the end of this week's learning you will:

- **Appreciate** some core principles of data visualization design
- Be able to **employ** these principles when critiquing data graphics
- **Create** effective data graphics using a key visualization toolkit built on these principles (`ggplot2`)

During the lab itself you will:

- open R using the RStudio Graphical User Interface (GUI)
- install and enable R libraries and query library documentation
- read in text (`.geojson`) files as R `DataFrames`
- perform data wrangling activities on R `DataFrames`
- calculate summary statistics over R `DataFrames`
- create data graphics using the `ggplot2` library
- generate choropleth maps in R using the `tmap` library
- generate small multiples in R using the `ggplot2` library
- make use of more advanced `tidyr` functions, such as `gather()`, for preparing R data frames for charting

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

Further resources for this week

- Lecture slides
- Introductory notes on data vis and `ggplot2` (optional)
- `.Rmd` file with code for this lab

Task 1. Configure your R and RStudio environment

Open .Rmd file and install relevant packages

An R package is a bundle of code, data and documentation, usually hosted centrally on the CRAN (Comprehensive R Archive Network). A particularly important set of packages is the tidyverse: a set of libraries authored mainly by Hadley Wickham, which share a common underlying philosophy, syntax and documentation.

In this session, you'll be using a collection of *packages* that form part of the so-called *tidyverse*. The packages together provide a very intuitive means of interacting with R and support analysis tasks that form most Data Science workflows.

There are two steps to making packages available in your working environment, which you will see on loading the `.Rmd` for this session:

- `install.packages(<package-name>)` downloads the named package from a repository
- `library(<package-name>)` makes the package available in your current session

Instructions

Download this `.Rmd` file, which contains the code needed for this lab and open in your RStudio session. Run the first code block (shown below), which both downloads and makes available the libraries used in this lab.

Instructions

A reminder: you can execute code blocks by selecting and clicking `run` or typing `cmd+r` (Mac) `ctr+r` (Windows); or alternatively you can copy and paste the code directly into the R Console.

```r
# Load required packages. Note that the install() commands are commented out as
packages are already downloaded on lab machines
# install.packages("tidyverse")
library(tidyverse)
# install.packages("sf") # SimpleFeatures package for working with spatial data.
library(sf)
# install.packages("tmap") # tmap library, which uses syntax very similar to ggplot2.
library(tmap)
```

Read in the dataset

In this lab, you'll be working with area-level demographic and voting data covering the UK's referendum vote on membership of the EU. You'll start by exploring the results data published for all 380 Local Authority Districts (LADs) in Great Britain (GB) made available in `.csv` form by The Electoral Commission. Later you'll explore the geography of this vote against demographic data collected from the 2011 Census. These data have been collated for this lab as a single data frame (`data_gb`).

Instructions

Run the code-block in the `.Rmd` file entitled *download-data* (also shown below). You should hopefully see in the *Data* window a new data frame called `data_gb`.

```
# A pre-prepared simple features data frame that loads Local Authority District
outlines
# (as sfc_MULTIPOLYGON) and 2011 Census data
data_gb <-
st_read("http://homepages.see.leeds.ac.uk/~georjb/geocomp/03_datavis/data/data_gb.geoj
son")
```

Notice that `data_gb` appears under the `Data` field of the *Environment* pane. It is stored as a *data frame* — a spreadsheet-like representation where rows correspond to individual observations and columns act as variables. You can inspect a data frame as you would a spreadsheet by typing `View(<dataframe-name>)` or by pointing and clicking on the named data frame in the *Environment* pane. You can also get a quick view on a data frame's contents by typing `glimpse(<dataframe-name>)`.

On inspecting its contents using `glimpse(data_gb)`, you will see that `data_gb` contains 380 observations — each is a LAD in GB. There are 34 variables; the first set of variables — `Electorate:margin_leave` — describe the referendum voting outcomes. `share_leave` describes the number of votes for Leave as a proportion of all vaid Leave/Remain votes; `margin_leave` is a signed value and describes the size of margin in favour of Leave against an expectation of the vote-share being 50:50 in favour of Leave:Remain. Where `margin_leave` for a LAD takes a positive value, there is a Leave majority, where it takes a negative value there is a Remain majority.

`data_gb` is a special class of data frame to the extent that it contains a `list-col` variable (called `geometry`). In the `geometry` variable are coordinates defining the spatial extent of Local Authority boundaries. Storing spatial information in this way is made possible by the *Simple Features* (`sf`) package — an important package for working with spatial data in R.

Task 2. Perform some exploratory data analysis

Do some data wrangling

Data wrangling type operations can be supported by functions that form the *dplyr* package. Again, *dplyr* is within the family of packages that comprise the *tidyverse*. Its functions have been named with verbs that neatly describe their purpose — `filter()`, `select()`, `arrange()`, `group_by()`, `summarise()` and more. The pipe (`%>%`) is a particularly handy operator that allows calls to these functions to be chained together.

Instructions

Run the code-block entitled *data-wrangle* (shown below). Given the result that's printed to your R Console, consider each line in the code block. Hopefully you will appreciate that chaining functions in this way (e.g. using `%>%`) enables you to construct quite expressive code…

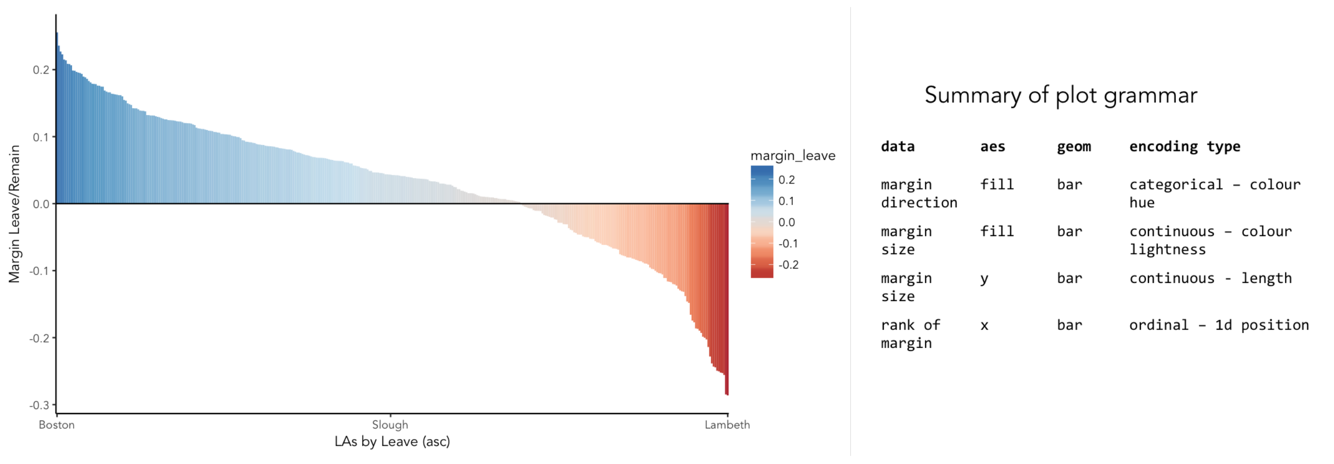
Individual coding task

… Try running the code to generate `region_summary` again, but exclude the final line of code — `arrange(desc(share_leave))` — then run `print(region_summary)`. Notice the difference? We no longer sort, or `arrange()`, regions in descending — `desc()` — order according to the regional `share_leave` vote.

```
# Calculate the LA share of Leave vote by Region.
region_summary <- data_gb %>%
      group_by(Region) %>%
        summarise(share_leave=sum(Leave)/sum(Leave, Remain)) %>%
           arrange(desc(share_leave))
# Clear geometry data.
st_geometry(region_summary) <- NULL
print(region_summary)
```

Create some summary graphics

Figure 3: Size of margin in favour of Leave:Remain by UK Local Authority.



Summary of plot grammar

data	aes	geom	encoding type
margin direction	fill	bar	categorical - colour hue
margin size	fill	bar	continuous - colour lightness
margin size	y	bar	continuous - length
rank of margin	x	bar	ordinal - 1d position

Instructions

Run the code block entitled *summary-graphics* (shown below). You should hopefully see in the *Plots* window a graphic closely resembling Figure 3.

```
# Create bar chart of result ordered by LA.
data_gb %>%
  ggplot(aes(x=reorder(lad15nm,-share_leave), y=margin_leave, fill=margin_leave))+
  geom_bar(stat="identity", width=1)+
  scale_fill_distiller(palette = 5, type="div", direction=1, guide="colourbar",
limits=c(-0.3,0.3))+
  scale_x_discrete(breaks=c("Lambeth","Slough","Boston")) +
  geom_hline(aes(yintercept=0))+
  theme_classic()+
  xlab("LAs by Leave (asc)")+
  ylab("Margin Leave/Remain")
```

Individual coding task

Try experimenting with the `ggplot2` layers for yourself. You might want to start with a simpler plot — for example the code block below, which produces a bar chart of the share of Leave vote by Region. Here vote share is mapped to bar length, but the bars are unordered and colour is not mapped to any data characteristic. Think about the ideas used in the LAD-level bar chart to make a more data-rich graphic.

Individual coding task

Hint: if you were to order your bar chart by the Leave vote, you would want to reorder the x (category axis) using the function `reorder(Region,share_leave)`.

```
# Code for simple bar chart of regional summary data.
region_summary %>%
  ggplot(aes(x=Region, y=share_leave))+
  geom_bar(stat="identity")
```

It is completely understandable if at this point you think somewhat tedious the whole idea of generating code to describe your graphics. However, once you learn how to construct graphics in this way — once you learn the `ggplot2` grammar — it is possible to very quickly generate relatively sophisticated exploratory graphics. That the grammar forces you to think more abstractly about data type and visual mapping is an important point. It introduces you to a key tenet of Information Visualization theory (Semiology of Graphics) by stealth, but also encourages you to think in depth about your data, its structure and (since you're producing graphics) the insights that can be made from visualizing that structure.

Task 3. Perform some exploratory spatial data analysis

Explore spatial variation in the Leave:Remain vote

In this section, you will analyse variation in EU referendum voting behaviour by Local Authority by generating various Choropleth maps. You will do so using the tmap package: a library that adopts a syntax very similar to that of `ggplot2`.

The general form of a *tmap* specification:

- A data frame containing a geometry field `(sfc_MULTIPOLYGON)` must be supplied to `tm_shape()`.
- To `tm_fill()`, we identify the variable values on which polygons should be coloured as well as information such as the colour mapping (sequential, diverging or continuous) and palette to use.
- `tm_layout()` provides control over titles, legends etc.

```
# Generate a choropleth displaying share of leave vote by LAD.
tm_shape(data_gb) +
  tm_fill(col="share_leave", style="cont", size=0.2, id="geo_label", palette="Blues",
title="") +
  tm_borders(col="#bdbdbd", lwd=0.5) +
  tm_layout(
    title="LA share of Leave vote",
    title.snap.to.legend=TRUE,
    title.size=0.8,
    legend.text.size=0.6,
    title.position = c("right", "center"),
    legend.position = c("right","center"),
    frame=FALSE,
    legend.outside=TRUE)
```

Instructions

Run the code block entitled *choropleth-leave* (shown above). You should see a map resembling that in the left of Figure 4.

Individual coding task

Create a new Choropleth that displays the *margin* in favour of Leave:Remain (a spatial equivalent of Figure 4). You will need to affect the column (`col=`) to which `tm_fill()` maps, as well as the `palette=` that is used (it should be `"RdBu"`). Remember, if you need more information about parameter options type `?<method-name>` into the R Console. Your map should look similar to the second map from the left in Figure 4.

Individual coding task

An obvious problem with displaying social phenomena on a conventional map is that visual salience is given to rural areas where relatively few people live, with patterns in urban areas de-emphasised. A partial solution is to split our map on the `Region` variable, resulting is a set of small multiple choropleths. This can be achieved with a call to `tm_facets(<"variable-name">)`. Adapt your code used to generate the choropleth map with this addition (e.g. `+ tm_facets("Region", free.coords=TRUE)`). You should end up with a map resembling the second from the right in Figure 4.

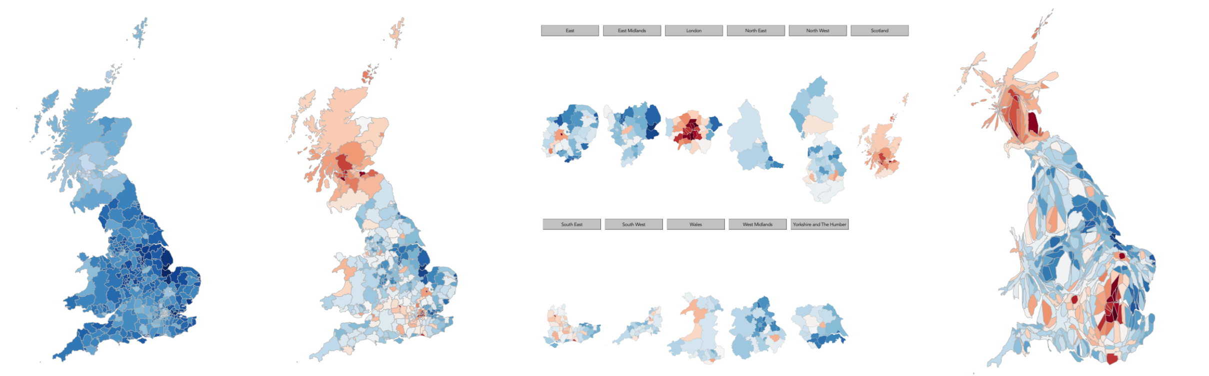
Instructions

An alternative and more direct approach to dealing with the salience bias of rural areas is through a cartogram. Cartograms allow spatial units (Local Authorities in this case) to be sized according to the phenomena of interest. Use the code shown below to create a cartogram similar to that presented in the right of Figure 4.

```
# Install cartogram package.
install.packages("cartogram")
library(cartogram)
# Convert from sf to sp format (supported by cartogram).
sp_gb <- as(data_gb, 'Spatial')
# Construct a new data frame, using the cartogram() function, passing as a
# parameter into the function the variable to which polygons are to be sized:
# Electorate (number of voters in LA). This may take a while.
sp_gb <- cartogram(sp_gb, "Electorate", itermax=10)

# Generate a choropleth using the same specification used in the conventional map,
# but supplying the cartogram SpatialDataFrame to tm_shape:
# e.g. tm_shape(sp_gb) +
#       etc.
```

Figure 4: Choropleths displaying the Leave:Remain vote by UK Local Authority District



data	aes	geom	encoding type
extent leave	fill	polygon	continuous – lightness
la units	x,y	polygon	2d ordinal position

data	aes	geom	encoding type
margin direction	fill	polygon	categorical – colour hue
margin size	fill	polygon	continuous – lightness
la units	x,y	polygon	2d ordinal position

data	aes	geom	encoding type
margin direction	fill	polygon	categorical – colour hue
margin size	fill	polygon	continuous – lightness
la units	x,y	polygon	2d ordinal position
region	facet	polygon	categorical

data	aes	geom	encoding type
margin direction	fill	polygon	categorical – colour hue
margin size	fill	polygon	continuous – lightness
la units	x,y	polygon	2d ordinal position
pop size	x_n,y_n	polygon	continuous – area

A cool feature of the `tmap` library is that it has an interactive mode, providing slippy map type functionality. This can be set with `tmap_mode("view")`. Enter this line into the R *Console* and then re-run the code that generates the Choropleth. To reset back to static mode use `tmap_mode("plot")`.

Data challenge (optional)

Despite seeing various iterations of these maps in the weeks after the referendum, the very obvious contrast between most of England & Wales (Leave) and Scotland and London, as well as certain university cities and towns (Remain), is surprising. Notice the spot of dark red in the East of England representing Cambridge.

Individual coding task

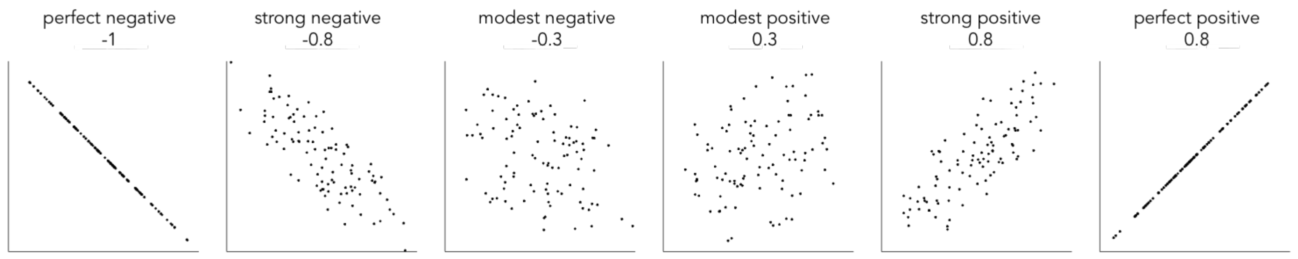
Can you come up with hypotheses as to why these differences in voting behaviour exist? Might the differences relate to differing characteristics of the underlying population? Try plotting choropleth maps of some of the Census variables that are stored in the `data_gb` data frame. Do ask if you need help with wrangling R or the `tmap` library.

Instructions

Now is probably a good time to save the R session you are working in. Select `Session`, `Save Workspace As …`, provide a sensible name and save to a sensible directory so that it can be easily returned to.

Task 4. Explore bivariate associations using correlation

Figure 5: Scatterplots containing varying extents of correlation coefficient.



The maps generated in Task 3 demonstrate a very obvious (and now familiar) pattern of spatial variation in Leave:Remain voting between LADs. Given the popular discourse on the Leave:Remain vote — that Leave represented those people and places structurally left behind by economic change — then it would be interesting to explore whether there is also systematic co-variation with Leave voting and the demographic characteristics of LADs.

This covariation in voting behaviour and demographics can be analysed more directly through correlation analysis. You will have learnt that the *correlation coefficient* can be used to summarise the strength of linear association between two variables. It is a quantity that ranges from perfect negative correlation (-1) to perfect positive correlation (+1) and can be considered a measure of effect size to the extent that it describes how much of something (correlation in this case) exists.

The code below examines the share of Leave vote and demographic characteristics of LADs using this quantity (correlation coefficient) as well as visually through a scatterplot.

```
# Calculate correlation coefficient of share Leave by degree-educated.
data_gb %>%
  summarise(cor(share_leave, degree_educated))

# Generate scatterplot of share Leave by degree-educated.
data_gb %>%
  ggplot(aes(x=share_leave, y=degree_educated)) +
  geom_point(colour="#525252",pch=21, alpha=0.8) +
  theme_bw()
```

Instructions

Run the code block *scatterplot* i your `.Rmd` file.

Individual coding task

Try exploring the relationship between share of Leave and different Census variables stored in the `data_gb` data frame.

Task 5. Generate graphical small multiples

Data analysis relies heavily on comparison. You might ask:

- How do associations between area-level Leave voting and education **compare** with other demographic characteristics?

- When **comparing** between different regions of the country, do these associations vary in any systematic way?

Such comparisons can soon become complex and multifaceted. One visualization solution for supporting such detailed comparison is small multiples — a set of graphics juxtaposed next to one another, ideally in a meaningful order, such that they can be compared. For a deeper overview, check out Nathan Yau's overview of small multiples.

`ggplot2` and related packages usefully support small multiples with functions such as `facet_wrap()` and `tm_facet()`. A requirement of calls to these various `facet` functions is Tidy data — that is, where just one observation appears per row. Rows in the the data frame are then split and a separate chart is created for each tranche of data.

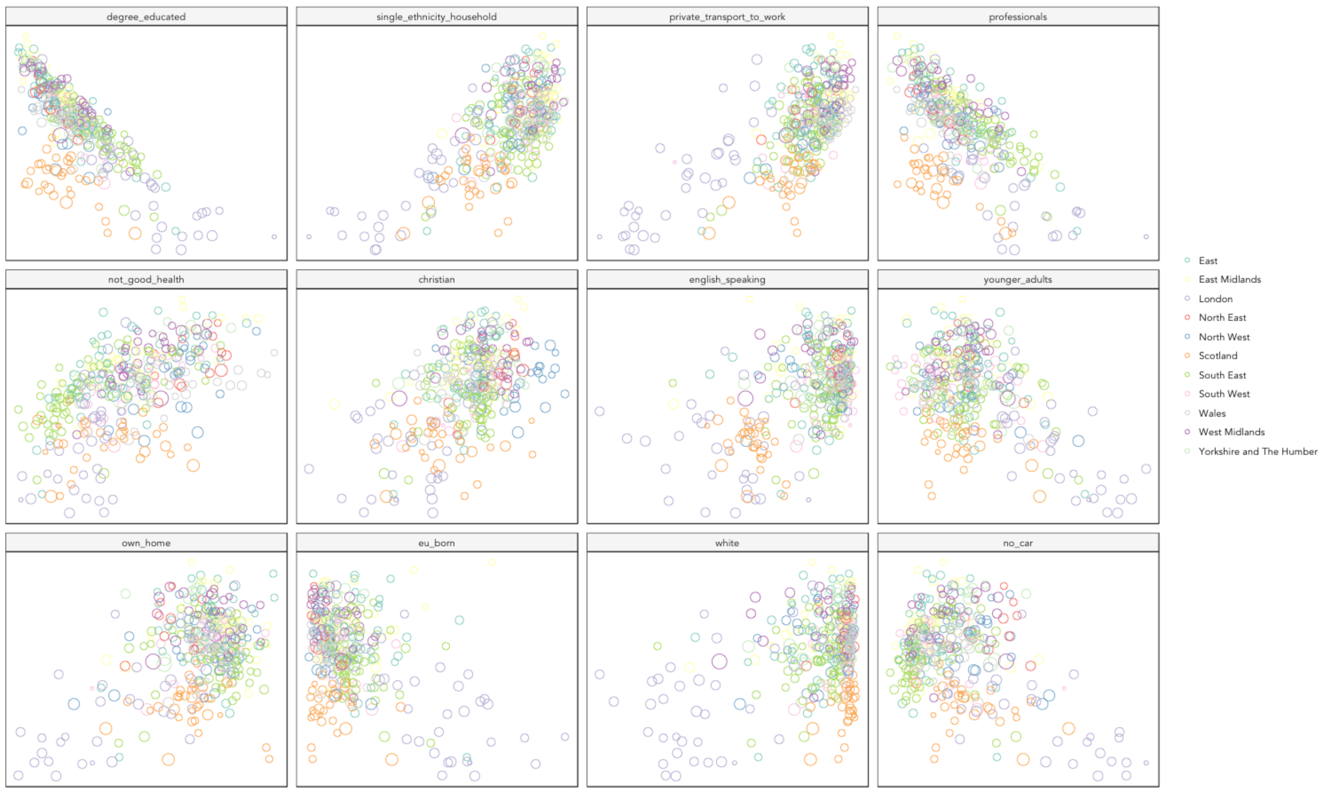
To generate small multiples of the scatterplots in the previous section — `share_leave` against Census variables — we have to collapse our data frame such that a single row is given to each LAD *and* Census variable — basically we need to make our data frame taller and thinner. The *tidyr* package provides methods that allow these types of operations. In the code block below the `gather()` method is used to collapse multiple columns into rows. Remember you can check the documentation for this function by typing into the R Console `?gather()`.

```
data_gb %>%
  gather(c(younger_adults:eu_born), key = "expl_var", value="la_prop") %>%
  ggplot(aes(x=la_prop, y=share_leave))+
  geom_point(colour="#525252",pch=21)+
  facet_wrap(~expl_var, scales="free")+
  theme_bw()
```

Individual coding task

Use the code above to generate small multiple scatterplots and maps similar to those appearing in Figure 6. Notice that the scatterplots are more efficient, data-rich graphics than those in the code provided above. Think about how to affect the *ggplot2* grammar in order to generate similarly data-dense graphics.

Figure 6: Scatterplots of share of Leave against key explanatory variables accompanied with a summary of the chart grammar.



data	aes	geom	encoding type
proportion census var	x	point	continuous – position in x
share of Leave	y	point	continuous – position in y
region	colour	point	categorical – colour hue
correlation rank	facet	point	ordinal – positional left-to-right, top-to-bottom

Data challenge (optional)

There has been much written about the demographics of the Brexit vote and place-based factors — that the Leave vote was associated with places 'left-behind' by globalisation. This seems to be borne out a little in the scatterplots in Figure 6. However, it is conceivable that some of these associations may be stronger for certain Regions of the country than others. Scotland, for example, has a very different history of euroscepticism to that of the home counties of England and so there may be different associations between demographic variables and Leave voting for these areas.

Individual coding task

Explore how associations between the Leave vote and the 12 Census variables held in the `data_gb` data frame vary by Region. You may also wish to investigate this visually by *faceting* on Region, but also by calculating regionally-specific correlation coefficients.

Further reading

If you're interested in extending this lab a little further, you might find interesting this workshop, which covers an exploratory data analysis and comparison of the Brexit and Trump votes.

R for Visualization

- Healey, K. (2018), Data Visualization: A practical introduction with R and ggplot2. This is perhaps the best resource on integrating key Information Visualization theory with practical examples using `ggplot2` and with interesting real social science datasets.

R for data analysis

- Wickham, H. & Grolemund, G. (2017), R for Data Science, O'Reilly. *The* primer for doing data analysis with R. Hadley presents his *thesis* of the data science workflow and illustrates how R and packages that form the *Tidyverse* support this. It is both accessible and coherent and is highly recommended.
- Lovelace, R., Nowosad, J. & Muenchow, J. (2018) Geocomputation with R, This book comprehensively introduces spatial data handling in R. It is a great complement to *R for Data Science* in that it draws on brand new libraries that support *Tidyverse*-style operations on spatial data.
- Kay, M. & Heer, J. (2016) Analysis code for "Beyond Weber's Law: A Second Look at Ranking Visualizations of Correlation". Reading others' data analysis using R is often instructive. Matt Kay, who has research interests in usable statistics and the reproducibility crisis, routinely publishes full data analysis code and documentation for his research papers. He has veleoped impressive research papers and libraraties for incorporating visualization with Bayesian data analysis — e.g. tidybayes — and it is definitely worth checking out his work on uncertainty visualization.

R for analysing Brexit vote

- Beecham, R. , Slingsby, A. and Brunsdon, S. (2018) Locally-varying explanations behind the United Kingdom's vote to leave the European Union, Journal of Spatial Information Science , 16: 117-136.