Finding, Merging, and Displaying Data

Andy Eggers, Oxford Q-Step Centre

Autumn 2017

1 Finding data

Before we can do interesting things with data, we have to locate the data. We have listed some useful data sources on another handout. Here I just wanted to mention a few others:

- Google: it's amazing what you can find with some keywords and the word "data"!
- For politics (my specialty), this list of datasets is amazing: https://github.com/erikgahner/PolData
- Replication datasets: Increasingly, academics make available the datasets they use in their research. Sometimes this is because academic journals require this. If you are interested in the data used in an academic study, you might find it on the author's website, on the journal's website, or by contacting the author directly.

2 Merging data

If we're lucky, we can find a dataset that contains all the variables we want to analyze. For example, we might find a dataset that has information about each country's level of female political representation, political corruption, and GDP per capita.

Often, we are not so lucky: we can find all the data we want, but it's not in the same dataset. This means we have to merge multiple datasets.

In very simple cases, where the datasets we want to merge have exactly the same units (e.g. the same 190 countries, labeled and listed alphabetically), we could make sure the rows are in the same order in each dataset and just paste the columns together in Google Sheets or Excel.

Quite often, there are some units in one dataset that are not in the other and it is much more convenient to use a merging function, which knits the datasets together by identifying rows that match on specified columns (e.g. country code).

2.1 When can two datasets be merged?

Obviously the datasets must describe the same basic unit or thing: merging won't work if each row in one dataset describes countries and each row in the other dataset describes cities.

The other key requirement is that there must be a column (or columns) that appears in both datasets and that can be used to determine which row goes with which row. For example, in the dataset of Brexit voting results there is a column called "Area code" that identifies the local authority; happily, a column called "Area code" also appears in census datasets describing the demographics of the local authorities. We can use this column to merge the two datasets.

2.2 Merging data in Google Sheets

To merge two datasets in Google Sheets, you first need to upload the two datasets to Google Sheets. With Google Sheets open, click on "File", then "Import. . . ", then "Upload" and find the file on your computer. To make the data appear in the same spreadsheet (i.e. same file, same browser tab), choose "Insert new sheet(s)", which will create additional sheets in the current file/browser tab.

You also need to install an Add-on named "Power Tools": click on "Add-ons", then "Get add-ons…", and then find and install "Power Tools".

Next, click on "Add-ons" again, hover over "Power Tools", and click "Start", then "Data", then "Merge sheets". Choose the first sheet you want to merge (which will be called the "Main table"); you can specify the range (rows and columns) here. It is a good idea to include the column headers. Then choose the second sheet you want to merge, which will be called the "Lookup table". Next you choose the column or columns[^1] that should be matched in the merging.

2.3 Merging Data in `R`

To merge two datasets in `R`, you first need to load the two datasets into memory in `R`. `R` can load files saved in many formats, including Excel files (`.xls, .xlsx`). I like to use CSV files; you can use Google Sheets or Excel to convert `.xls` or `.xlsx` files into CSV files.

Having saved the necessary files as CSV files on my Desktop, I load them into `R`:

```
voting.results = read.csv("~/Desktop/EU-referendum-result-data.csv")
country.of.birth = read.csv("~/Desktop/country_of_birth.csv") # available only for England and Wales
```

Then I merge them:

```
D = merge(x = voting.results, y = country.of.birth, by.x = "Area_Code", by.y = "Area.code")
```

See these resources for more on merging data in R:

- Oscar Torres-Reyna: "Merge/Append using R"
- DataCamp: "15 Easy Solutions To Your Data Frame Problems In R"

3 Displaying data

3.1 Displaying data in Google Sheets

Many chart types are possible in Google Sheets (similar to Excel).

To make a scatterplot of two variables:

- Select the data you want to plot, then click "Insert", then "Chart"
- Chart type should be "Scatterplot"
- Make sure the range of the data is correctly specificed in "X-AXIS" and "SERIES"
- Click "Customize" to make adjustments:
  - "Chart & axis titles"
  - "Series": adjust point size and point shape; add a trendline (linear, k-order polynomial, etc)
  - Adjust or remove legend, etc
- Using menu at top-right of chart, copy, export, publish the chart

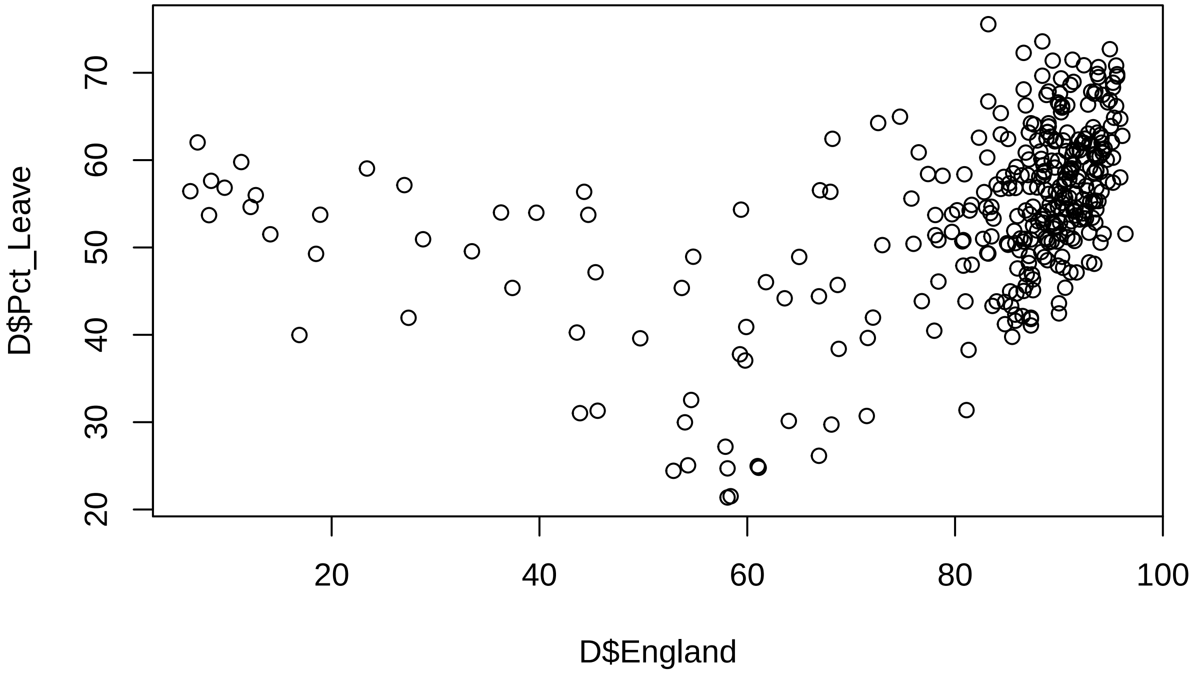
A scatterplot made in Google Sheets using the Brexit data appears at the end of this worksheet.

3.2 Displaying data in R

You can make a staggering variety of figures and plots in R.

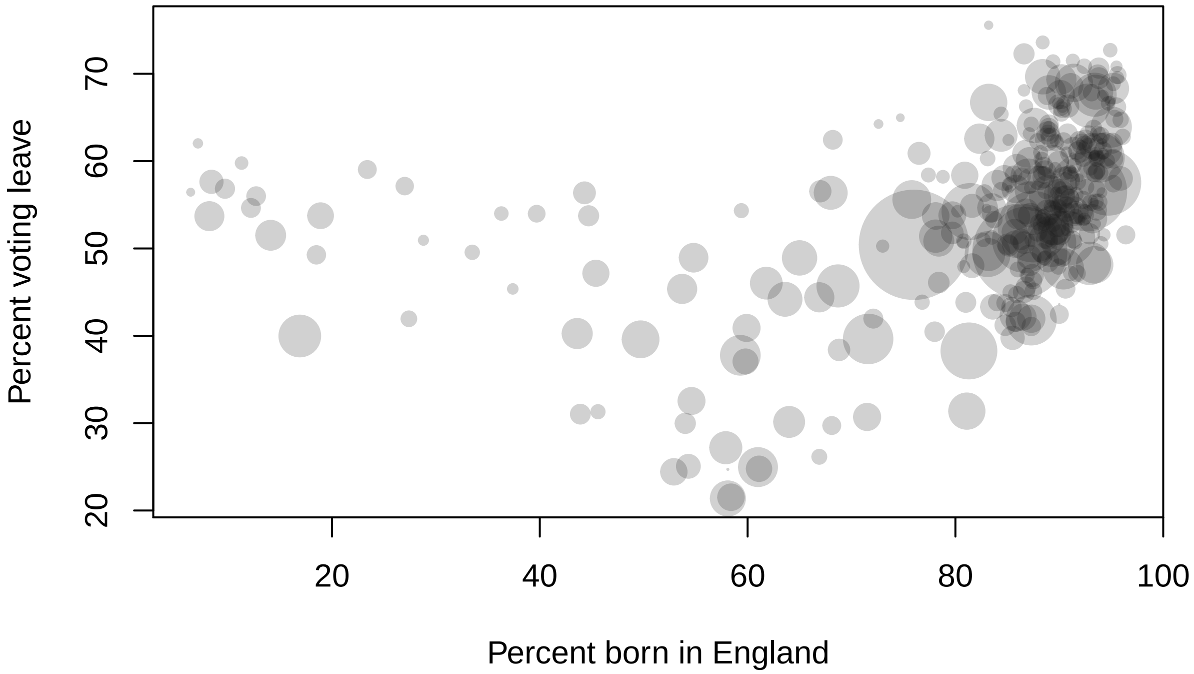
Here is a simple scatterplot that uses the Brexit data we merged above:

```
plot(D$England, D$Pct_Leave)
```



We can make this a little prettier by changing the axis labels, using a different plotting symbol, and adjusting the size of the plotting symbols to be proportional to the size of the electorate in the local authority:

```
# "comments" after #-sign are not part of code
plot(D$England, D$Pct_Leave,
     pch = 19,    # makes the plotting symbols be filled in circles
     cex = D$Valid_Votes/60000, # size proportion to valid votes
     col = rgb(.1, .1, .1, alpha = .2), # color is gray partially transparent
     xlab = "Percent born in England", # x-axis label
     ylab = "Percent voting leave")    # y-axis label
```



Any ideas why the relationship between the percentage of local authority residents born in England appears to have a U-shaped relationship with percentage voting leave?

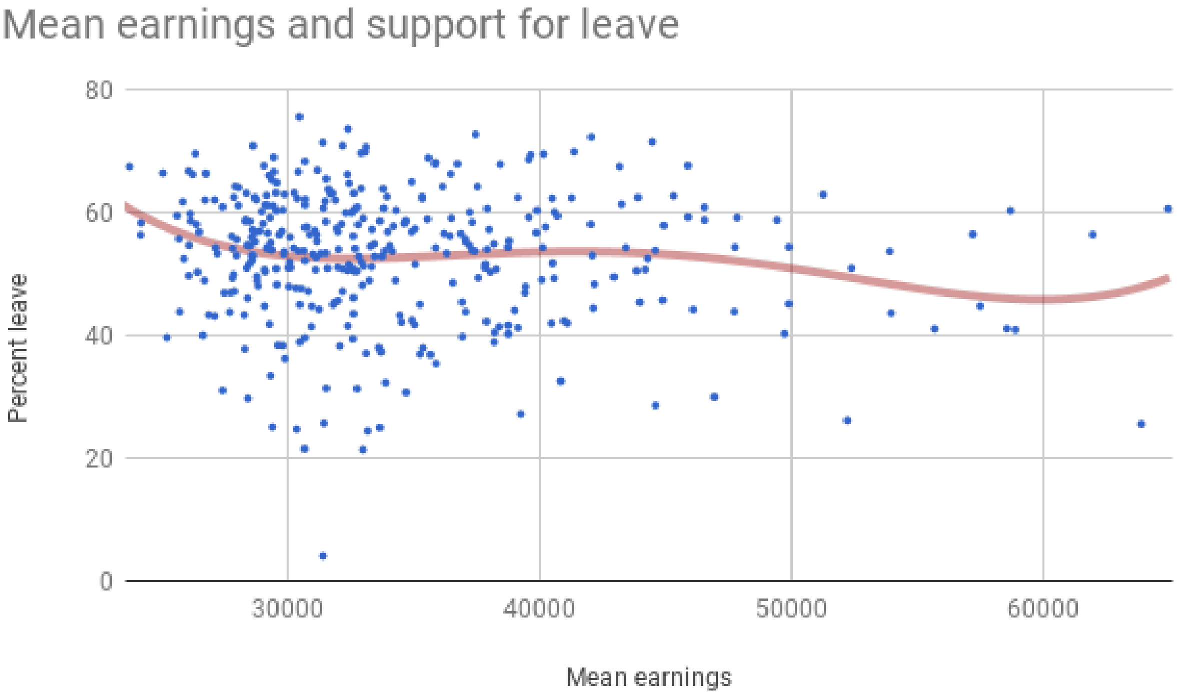


Figure 1: A scatterplot made in Google Sheets from the Brexit data