DATA VISUALISATION

LAB WORKSHEET 7

Creator: Dr Mark Taylor

What's Happening in this Document?

More maps!

Specifically, more maps where we're including data that we've downloaded from the web.

It's unusual that we're working with data that's already on our machines. So, today, what we're going to do is download some data from the web as it comes, in a format that's a tiny bit messy.

Getting set up

Let's start, as we always do, by loading some packages. We're going to continue drawing maps using the sf package, with which a handful of people ran into issues last week.

```
library(tidyverse)
library(sf)
```

Q-Step A step-change in quantitative social science skills Funded by the Nuffield Foundation, ESRC and HEFCE
Nuffield Foundation
Sheffield Methods Institute.
The University Of Sheffield.

As with our last worksheet, we're loading data from our R Working Directory. We are using some of the same file and it should already be in your Working Directory. If this has worked, you'll be able to run the following line:

```
westminster <- read_sf(dsn = ".", layer = "westminster_const_region")
```

Once we have done this, a shapefile of parliamentary constituencies should load (It's the same file as our last worksheet).

Switch to the web

Let's download some data! Specifically, let's download some data that relates to England and Wales from the 2011 Census. (The data for Scotland, and for Northern Ireland, are a bit different, for tedious reasons, which is why we're sticking with England and Wales for now.)

Please navigate as follows: - navigate to https://www.nomisweb.co.uk/census/2011 (https://www.nomisweb.co.uk/census/2011); - from the wide range of options, click on Key Statistics; - from the wide range of options, click on KS102EW, described as "Age Structure"; - the page you end up with gives you some information about the variables associated with age structure for different areas. Under Download (.csv), on the bottom left, when you're presented with a range of different types of areas, scroll down to parliamentary constituencies 2010 - hit Download.

This should download you a file called bulk.csv

Please now move your new file to your Working Directory, and rename it to age_structure.csv.

Loading the new data

Let's load the new data! If you've managed to move it successfully, the following should work:

```
age_structure <- read.csv("age_structure.csv")
```

and we can have a look at what it looks like.

```
names(age_structure)
head(age_structure)
```

OK, this is pretty usable (compared with some other data that we can download from the web, this is astonishingly well put-together and accessible). However, when we load data that I've cleaned up, we normally have column names that are brief, and easy to understand. These are clear, but quite lengthy, which can make it a bit trickier to interpret what's going on.

So, what we can do is just go through and rename the variables. What we're not doing here is adding new variables with shorter names: by renaming them, the old variable names are disappearing. We can do this with the **rename** command, like so:

```
age_structure_renamed<- age_structure %>%
rename(date = date,
          geography =geography,
          CODE =geography.code,
          rural =Rural.Urban,
          population=Age..All.usual.residents..measures..Value,
          between0and4 = Age..Age.0.to.4..measures..Value,
          between5and7 = Age..Age.5.to.7..measures..Value,
          between8and9 = Age..Age.8.to.9..measures..Value,
          between10and14 = Age..Age.10.to.14..measures..Value,
          between15 = Age..Age.15..measures..Value,
          between16and17 = Age..Age.16.to.17..measures..Value,
          between18and19 = Age..Age.18.to.19..measures..Value,
          between20and24 = Age..Age.20.to.24..measures..Value,
          between25and29 = Age..Age.25.to.29..measures..Value,
          between30and44 = Age..Age.30.to.44..measures..Value,
          between45and59 = Age..Age.45.to.59..measures..Value,
          between60and64 = Age..Age.60.to.64..measures..Value,
          between65and74 = Age..Age.65.to.74..measures..Value,
          between75and84 = Age..Age.75.to.84..measures..Value,
          between85and89 =Age..Age.85.to.89..measures..Value,
          between90andmore=Age..Age.90.and.over..measures..Value,
          mean_age = Age..Mean.Age..measures..Value,
          median_age = Age..Median.Age..measures..Value)
```

That was a lot. (You might wonder whether we needed all the variable names that I've started with between: in practice, it might have been overkill to rename every single variable, rather than just the variables I was going to use. But we've come this far; we might as well carry on and get the practice!)

Right, let's merge the data.

Merging the data

We did some merging in our last worksheet. Let's now look at it in a bit more detail.

```
age_with_map <-
  merge(westminster, age_structure_renamed, id = "CODE")
```

What have we got here? Let's go through the steps one-by-one.

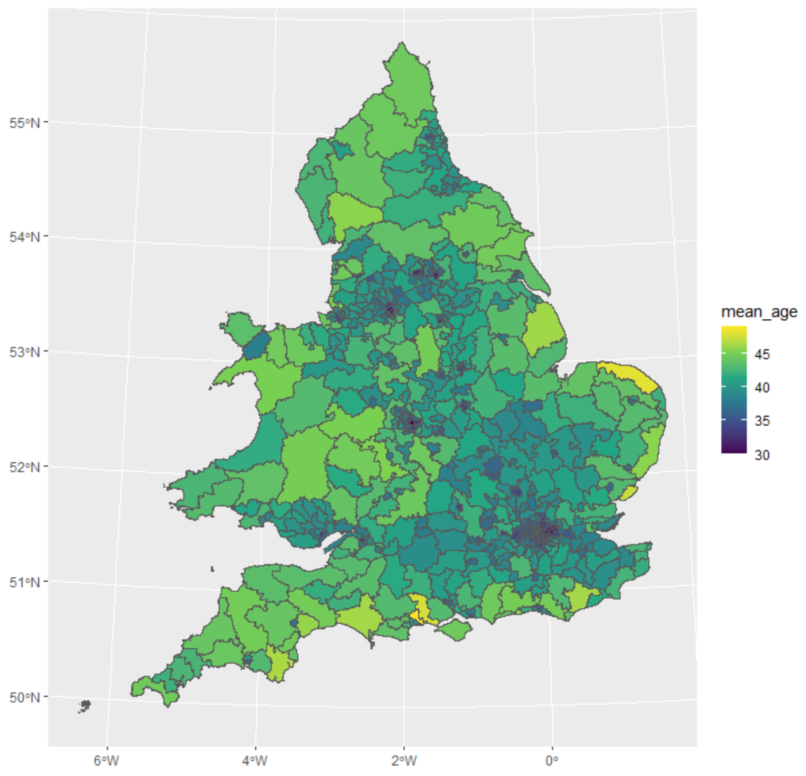
- age_with_map is the name of our new object;
- the arrow means it's going to consist of the thing on the right;
- the merge command means we're combining two datasets.
- following the merge command, we're specifying the two datasets we want to combine;
- finally, we're specifying the id variable that both datasets have. R can use this to match up observations. Crucially, this means that a dataset with a common id variable, but a flaky label variable (eg a difference between Cheshire West and Chester and Cheshire West & Chester) is a real problem if you're using the name field to join, but not a big deal if you're using the id field;
- you'll note that when we renamed our variables earlier we renamed geography_code to CODE; this was so that the common variable was called the same thing in both data frames.

Let’s draw some maps

Has it worked? Let’s find out.

1. Which parliamentary constituency has the highest mean age?

```
ggplot(data = age_with_map) +
  aes(fill = mean_age) +
  geom_sf() +
  scale_fill_viridis_c()
```



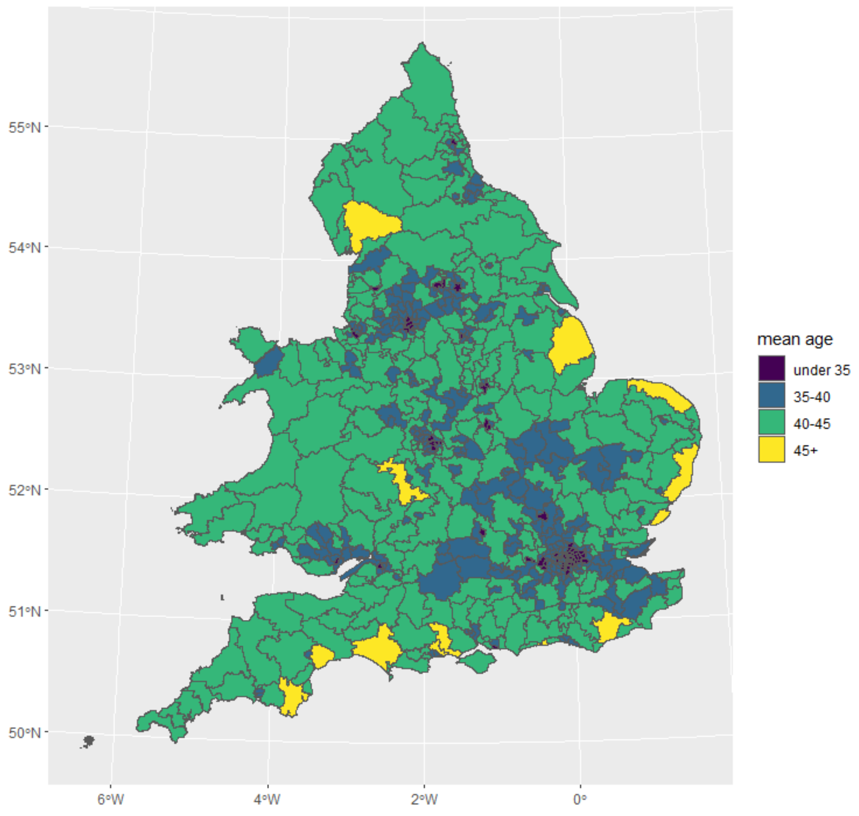
Let’s try some different bins

OK, this looks interesting. Let’s do some binning to understand the categories in more detail.

2. You’ll remember last time we tried two different approaches to binning: one where there were equal intervals, one where there were equal numbers. Let’s try specifying the bins ourselves: maybe we’re interested in particular ranges, such as areas that have mean ages of under 35.

```
ggplot(data = age_with_map) +
  aes(fill = cut(mean_age,
                 breaks = c(0, 35, 40, 45, 100),
                 labels = c("under 35",
                            "35-40",
                            "40-45",
                            "45+"))) +
  geom_sf() +
  scale_fill_viridis_d() +
  labs(fill = "mean age")
```

This looks OK. What does this show?



Bonus Exercise

Draw a map of the fraction of people in each parliamentary constituency who are 65 or older.