

jrSpatial - Practical 3

Jumping Rivers

We'll start by loading the necessary packages.

```
library("sf")
library("tmap")
library("jrSpatial")
```

Question 1

Load in the `ukgeom` data using

```
data(ukgeom, package = "jrSpatial")
```

1. Calculate the population density for 2017 and save these in new columns. Population density is defined as the population divided by the area.

Bonus question: Use `tmap()` to create a plot to show the population density in 2017 across the U.K.

2. There is another dataset `ukData` which contains extra statistics about the UK regions such as birth rates and income. Load this data using

```
data(ukdata, package = "jrSpatial")
```

Create a new data frame called `ukgeomfull` which combines `ukgeom` with `ukdata` using `left_join()`.

Does the order of the arguments matter? What happens if you put `ukdata` as the first argument instead of `ukgeom`? (Hint: Look at the class of the output using `class()`.)

3. Use `filter()` to find the regions with the following characteristics.
 - Have a population (in 2017) greater than 1,600,000.
 - Has a disposable income of less than 15000 or an unemployment rate of greater than 5.
4. Use `group_by()` and `summarise()` to answer the following questions. You may wish to drop the geometry column with `st_drop_geometry()` to make the output easier to view.
 - What was the total population in 2017 for each country?
 - What about the average life expectancy?

*Question 2**Hills of UK*

The `ukhills` dataset includes the name, height and geometry of all the Marilyn's in the U.K. A Marilyn is a hill which is relatively high compared to its surroundings. Load it into R using

```
data(ukhills, package = "jrSpatial")
```

1. We'd like to know which region each of the hills are in. In order to do this, we'll need to combine the `ukgeom` data with the `ukhills` data set. There are no common columns, so this time we'll be joining by geometry. Use `st_join` to create a new data frame called `hills_uk` which merges `ukhills` and `ukgeom`. Use `View()` to inspect the output.
2. Use `tm_shape()` and `tm_dots()` to overlay the hills data on the UK regions borders.

Don't worry about the warning "although coordinates are longitude/latitude, `st_intersects` assumes that they are planar" - This is only a problem if points are massive distances away or near the poles.

2. After joining there are NAs in the region column. Why have these occurred? What do they represent? Plotting the data may help you.
3. Use `filter()` and `arrange()` to find the three highest Marilyn's in Cumbria.
4. Use `group_by()`, and `summarise()` to work out which region has the most Marilyn's

Solutions

```
vignette("solutions3", package = "jrSpatial")
```