

## *jrSpatial - Practical 2*

### *Jumping Rivers*

Start by loading the correct libraries

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

#### *Question 1 - first steps in tmap*

We're going to be using the same UK shape data from the first practical but don't worry, we'll not get you to read it in again. We've got it stored inside the course package

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

- a) Using **tmap**, plot the 41 regions within `uk_shape`
- b) Colour the polygons by the population in 2017
- c) The legend isn't a very good position, change it to either the top left or outside of the graph. Hint: If you don't like the position of the legend outside the graph, using `legend.outside.position` to change it.
- d) By default the legend title is the variable name, we can change this by using the `title` argument within `tm_fill()`. Have a go!
- e) Have a go at changing the colour palette. See <http://colorbrewer2.org> for some options!
- f) Colour the polygons by both the population of 2015 and 2017. Can you see any major differences?

#### *Question 2 - Overcrowding*

Let's say we're interested in evaluating overcrowding in the UK. Absolute population isn't a very good measure because each region has a different area in total. A better measure would be area per person. We can work this out as we have an `area` column in the shape data. Below is the code to create the column then plot the data

```
library("dplyr")
ukgeom = mutate(ukgeom, areapps = area/pop_2017 * 1000)
# * 1000 to change units from km^2 to m^2
tm_shape(ukgeom) + tm_layout(legend.position = c("left", "top")) + tm_fill("areapps",
  title = "Area per person") + tm_borders()
```

Because the area at the top of Scotland is so vast an unpopulated compared to the rest of England, we're getting a very skewed colour scale. Change the breaks within `tm_fill()` to something more appropriate

```
tm_shape(ukgeom) + tm_layout(legend.position = c("left", "top")) + tm_fill("areapps",
  title = "Area per person", breaks = c(seq(0, 20, 4), 100)) + tm_borders()
```

### *Question 3 - Hills!*

As well as our regional polygons data for the UK, we also have a data set of points for the highest hill (or mountain) in each region. You can load the data like so

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

We can add several layers of different shapes onto a **tmap**, just by adding another `tm_shape()` like so

```
tm_shape(ukgeom) + tm_fill() + tm_borders() + tm_shape(ukhills_big) + tm_bubbles()
```

- a) Try changing the colour of the mountain points depending on it's height, and try changing the size of the point using the `size` argument.
- b) Run the following line of code and then re-run the code for your graph in question 3a). You should get an interactive version of the map!

```
tmap_mode("view")
```

- c) Currently the text that appear when you hover over a region is the region code, instead of the region name. For mountains this text is the region, instead of the mountain name. Try adding the arguments `id = "region"` to `tm_fill()` and `id = "Mountain"` to `tm_bubbles()`. Try clicking on the shapes!

### *Solutions*

```
vignette("solutions2", package = "jrSpatial")
```