STAT1003

Introduction to Data Science

Lab 3

Contents

Gapminder data from Workshop 2

In the first part of this Lab, we'll continue analyzing the Gapminder data from Lab 2 to produce some plots. Please use the help file for the different functions we're going to be using.

```
# Load the data from Lab 2
```

1. Construct a plot of the average life expectancy over the last four years in each country against its average GDP per capita over that same time period, and label it appropriately.
2. What do you notice about the scale of GDP? Transform it appropriately and re-plot.
3. Use different symbols for countries in different continents. Hint: you'll need the argument `pch`, so see the help file for `plot`. First create a vector that gives you the continent that each country is in.
4. Add a legend so that we know which symbol corresponds to which continent.
5. Construct a plot similar to Q4, but this time use the same symbol but different colours for each continent.

Where do students live?

For a particular set of first-year units this semester, `S1_2020_STAT1003_Workshop3.RData` contains some information on the number of students who live in different suburbs in WA. There is also information on the latitude and longitudes of the centroids of each postcode.

1. Load the data file. What objects are loaded, and what's in them? For the time being, ignore the object `WA`. Which suburb has the highest number of students?
2. **Merge** the two datasets together to create a new data frame that contains only information on the students for whom we have information. Call this merged data frame `GeoTable`.
3. Install the libraries `sp`, `maps` and `mapdata` onto your computer, and then load them in to your *R* session.
4. Using the following code, look at the appropriate help files to plot on a map of the Perth area the frequency of students from each suburb. Use a larger symbol for larger frequencies. Fill in the appropriate values of `XX`, `YY`, `ZZ`, etc. Change the `FALSE` to `TRUE` in the code chunk header to evaluate the code.

```
map("worldHires", ylim = c(YY, YY), xlim = c(XX, XX), fill = TRUE, col = "lightgrey",
    mar = c(4.5, 4, 1, 1))
map.axes()
map.scale(x = 116.1, ratio = FALSE)
points(GeoTable$long, GeoTable$lat, cex = GeoTable$Frequency/ZZ, pch = 21, col = "black",
    bg = rgb(1, 0, 0, 0.5))
map.cities()

# Add a blue circle for the co-ordinates of Curtin University.
points(115.89405, -32.00469, pch = 16, col = "blue")
```

5. As appealing as the figure above might be, we can produce an even more useful map known as a *choropleth*. A choropleth is simply a map in which geographic regions are shaded according to a variable of interest. In this case, the geographic regions are suburbs, and of course, the variable of interest is the number of students from that suburb taking these units.

   The object `WA` is a *shapefile*, which contains, among other things, co-ordinates of polygons representing the boundaries of suburbs in WA, their names, areas, and other information.

   As above, **merge** the object `GeoTable` above with the shapefile `WA`, and call the resulting object `Freq_by_Table`. Then run the code below.

   Which plot is more useful?

```
spplot(Freq_by_Suburb, z = "Frequency", xlim = c(XX, XX), ylim = c(YY, YY), col.regions = colorRampPale
    "orange", "red"))(24), scales = list(draw = TRUE))
```