Exercises: Choropleth maps of country-level statistics

YSC2210 - DAVis with R

Michael T. Gastner

Introduction

In a previous exercise, we made a scatter plot of life expectancy as a function of per-capita GDP (figure 1). Now we want to visualise the data as a thematic map, similar to figure 2.

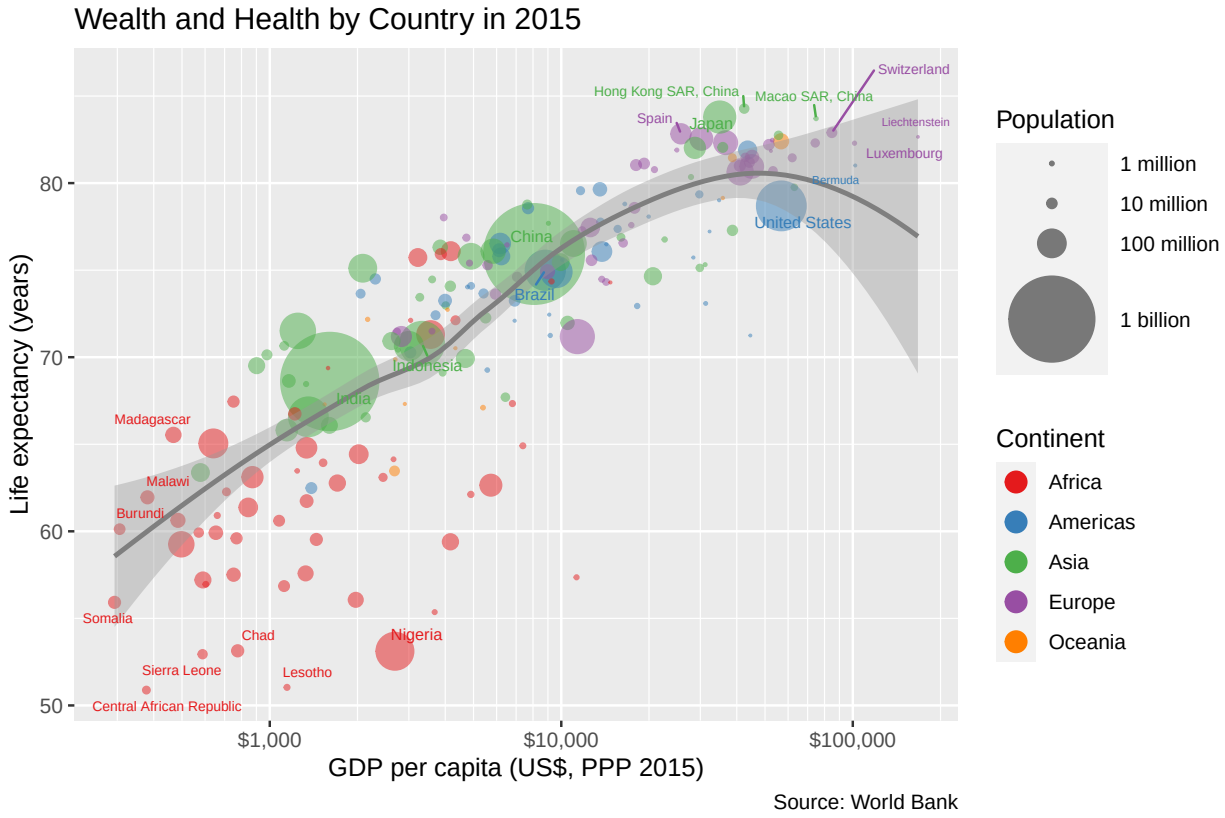

Figure 1: Scatter plot of life expectancy vs. GDP per capita by country.

Data

- GDP per capita (current US$): https://data.worldbank.org/indicator/NY.GDP.PCAP.KD
- Life expectancy at birth, total (years): https://data.worldbank.org/indicator/SP.DYN.LE00.IN
- Geospatial boundaries:
  (1) Go to https://datacatalog.worldbank.org/search/dataset/0038272.
  (2) Click on the download buttons to the right of:

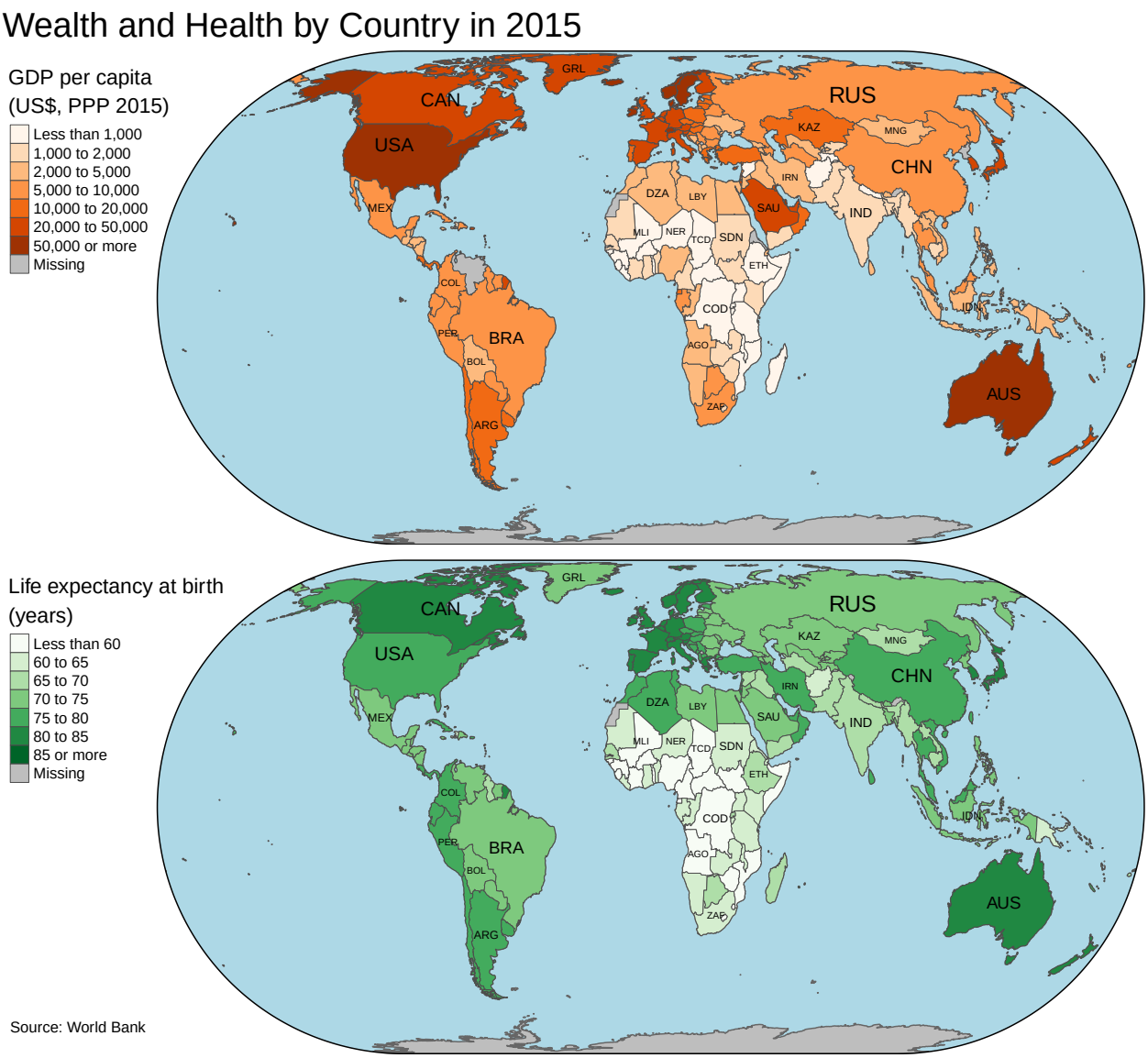

Figure 2: Choropleth maps for per-capita GDP (top) and life expectancy (bottom).

   - 'World Country Polygons - Very High Definition'.
   - 'World Land Area'.

(3) Move the downloaded files `wb_countries_admin0_10m.zip` and `wb_land_10m.zip` to your project folder. These zip files contain so-called shapefiles that can be imported with the **sf** package.[1]

Tasks

(1) Briefly review the R code we wrote in our earlier exercises about the World Bank's GDP and life expectancy data. You can copy parts of that code to get started with today's exercise.

(2) Import the country borders as `sf` objects. Because the boundaries are represented by polygons with many vertices, plotting the data with **tmap** later on would be slow. Instead of importing directly with `read_sf()`, please use the following function, called `import_and_simplify()`, instead. It uses the function `ms_simplify()` from the **rmapshaper** package to reduce the number of vertices.

```
import_and_simplify <- function(zip_file) {
  zip_file |>
    unzip() |>
    str_subset(".shp$") |>
    read_sf() |>
    rmapshaper::ms_simplify()
}
```

(3) Norway and France have invalid country codes in the column `ISO_A3`. Change them to `NOR` and `FRA` respectively.

(4) With the help of the files listed above (in the section called 'Data'), create an `sf` object called `countries` that contains the columns:

- `name`
- `code` (with ISO 3166-1 alpha-3 country code)
- `gdp_per_cap` (only for 2015)
- `life_exp` (only for 2015)
- `geometry` (use the `sf` object you created in the previous task)

Include as many countries as possible. However, exclude any countries for which the geometry is empty (i.e. unknown) because empty geometries would trigger a warning by the **tmap** package later on. (Hint: you can find empty geometries in the `geometry` column by using `st_is_empty(countries)`).

(5) Look at a quick-and-dirty visualisation made with **tmap** that shows the geometries stored in the `countries` object. Are any countries or land masses missing?

(6) Add the missing polygons to the map by

   (a) importing and simplifying the World Bank geospatial data for land masses with `import_and_simplify("wb_land_10m.zip")`.
   (b) plotting the land masses in grey as base layer and the country borders in `countries` as top layer.

(7) Because we want to make choropleth maps, we should use an equal-area map projection. What is the current projection of the map? Is it an equal-area projection? (Hint: you can find the proj string of an `sf` object `x` with `st_crs(x)$proj4string`.)

(8) Make a polished plot with **tmap**. Ensure it has the following features:

- It consists of two maps: one map for GDP per capita, another map for life expectancy.
- Both maps are based on an Eckert IV projection.

[1] The World Bank's Data Catalog also includes an entry 'World Boundaries GeoJSON - Low Resolution'. The topologies in these GeoJSON files is invalid and would lead to problems when trying to plot them with the **tmap** package. Therefore, we import geospatial boundaries from the World Bank's shapefiles instead of the GeoJSON files.

- Use suitable ColorBrewer palettes.
- Choose a consistent shade of grey for missing data.
- Add country codes as labels to the map (but only where the code fits inside a country).
- Choose suitable ranges, scales and breaks for the legends.
- Give a title to the plot and subtitles to both maps.
- Fill the oval that represents the globe with a light blue background colour and everything else with a white background colour.
- Place the legend in a suitable position.
- Credit the World Bank as source of the data.
- Make a sensible choice for the figure dimensions. Labels should be clearly legible without appearing disproportionately large.

Feel free to make more adjustments if you think they improve the quality of the plot.

(9) What can a reader conclude from the maps?

(10) What are the advantages and disadvantages of this visualisation in comparison to figure 1?