Lesson 20

Michael T. Gastner

2022-04-05

I need the following packages for this exercise.

```
library(countrycode)
library(readxl)
library(sf)
library(tidyverse)
library(tmap)
```

(1) Code from the exercise about relational data.

```
gdp_per_cap <- read_excel(
  "API_NY.GDP.PCAP.KD_DS2_en_excel_v2_3731742.xls",
  skip = 3
)
life_exp <- read_excel(
  "API_SP.DYN.LE00.IN_DS2_en_excel_v2_3731513.xls",
  skip = 3
)

# Next function can pivot any of the World Bank tibbles
pivot_longer_wb <- function(tb, name) {
  tb |>
    select(-starts_with("Indicator")) |>
    pivot_longer(
      matches("^\\d{4}$"), # String with exactly 4 digits
      names_to = "year",
      values_to = {{ name }},
      names_transform = list(year = as.integer)
    )
}
wb <-
  Hmisc::llist(life_exp, gdp_per_cap) |>
  imap(pivot_longer_wb) |>
  reduce(left_join, by = c("Country Name", "Country Code", "year")) |>
  semi_join(codelist, by = c("Country Code" = "iso3c"))
```

(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()
}

country_borders <- import_and_simplify("wb_countries_admin0_10m.zip")
```

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

```
country_borders <-
country_borders |>
  mutate(ISO_A3 = case_when(
    FORMAL_EN == "Kingdom of Norway" ~ "NOR",
    FORMAL_EN == "French Republic" ~ "FRA",
    FORMAL_EN == "French Polynesia" ~ "FRA",
    FORMAL_EN == "Territory of the French Southern and Antartctic" ~ "FRA",
    TRUE ~ ISO_A3
  ))
```

(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)).

```
countries <-
  country_borders |>
  inner_join(
    wb,
    by = c("ISO_A3" = "Country Code")
  ) |>
  filter(year == 2015) |>
  select(
    name = "Country Name",
    code = "ISO_A3",
    gdp_per_cap,
    life_exp,
    geometry
  ) |>
  print()
```

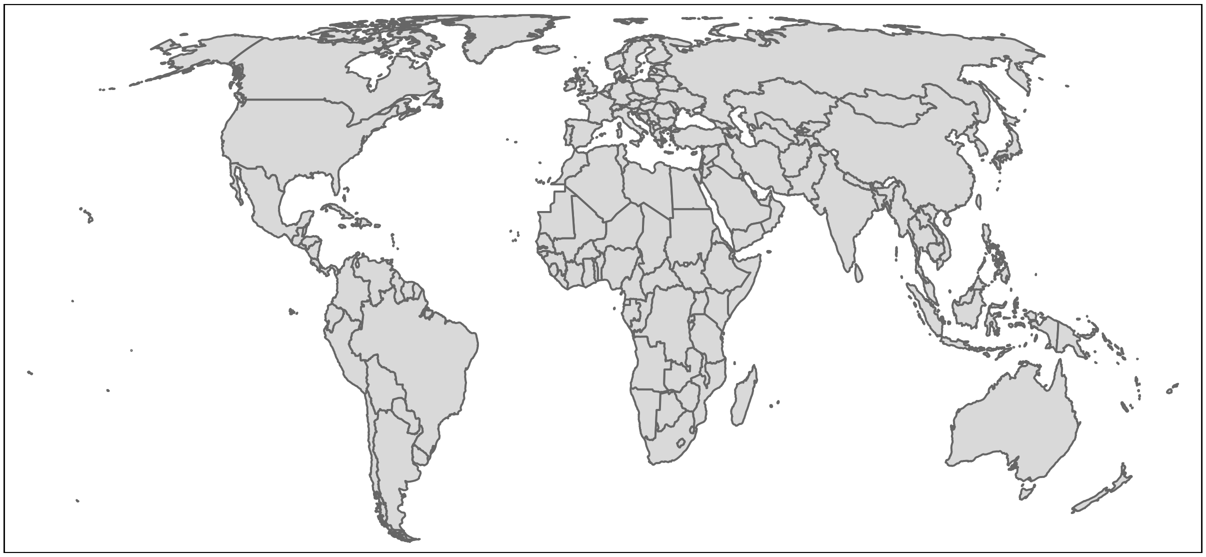
```
## Simple feature collection with 186 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -55.62754 xmax: 180 ymax: 83.6341
## Geodetic CRS:  WGS 84
## # A tibble: 186 x 5
##    name      code  gdp_per_cap life_exp                                 geometry
##    <chr>     <chr>       <dbl>    <dbl>                       <MULTIPOLYGON [°]>
##  1 Indonesia IDN         3332.     70.8 (((117.7036 4.163415, 117.907 4.156683,~
##  2 Malaysia  MYS         9955.     75.5 (((117.7036 4.163415, 117.7854 4.24665,~
##  3 Chile     CHL        13574.     79.6 (((-69.51009 -17.50659, -69.47578 -17.6~
##  4 Bolivia   BOL         3036.     70.3 (((-69.51009 -17.50659, -69.5226 -17.36~
##  5 Peru      PER         6229.     75.8 (((-69.51009 -17.50659, -69.6839 -17.65~
##  6 Argentina ARG        13789.     76.1 (((-67.1939 -22.82222, -67.03769 -22.65~
##  7 Cyprus    CYP        23408.     80.4 (((34.13771 35.41511, 33.97136 35.32128~
##  8 India     IND         1606.     68.6 (((77.80035 35.49541, 78.05563 35.45288~
##  9 China     CHN         8016.     75.9 (((110.6851 20.15331, 110.7883 20.0104,~
## 10 Israel    ISR        35808.     82.1 (((35.60385 33.24009, 35.63817 32.87469~
## # ... with 176 more rows
```

```r
any(st_is_empty(countries))
```

```
## [1] FALSE
```

(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?

```r
tm_shape(countries, projection = "+proj=eck4") +
  tm_polygons()
```



Are any countries or land masses missing? Antarctica, Western Sahara, Ladakh, and Arunachal Pradesh is missing.

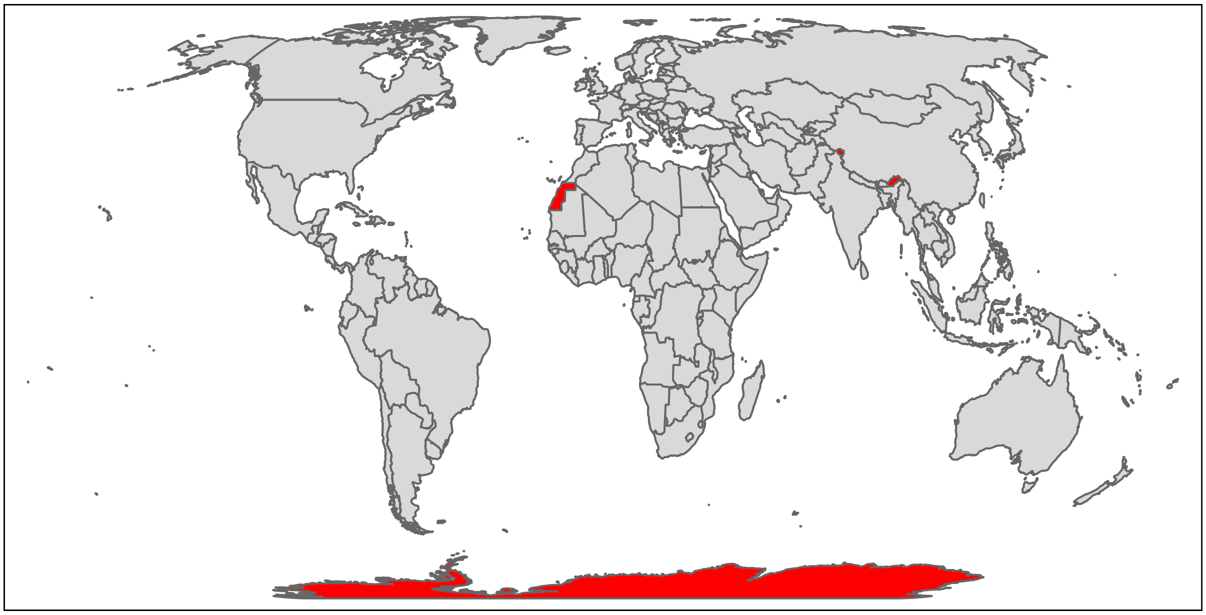
6) Add the missing polygons to the map by

(a) importing and simplifying the World Bank geospatial data for land masses with `import_and_simplify("wb_land_10m.zi`

(b) plotting the land masses in grey as base layer and the country borders in countries as top layer.

```r
land_masses <- import_and_simplify("wb_land_10m.zip")

tm_shape(land_masses, projection = "+proj=eck4") +
  tm_polygons(col = "red") +
  tm_shape(countries, projection = "+proj=eck4") +
  tm_polygons()
```



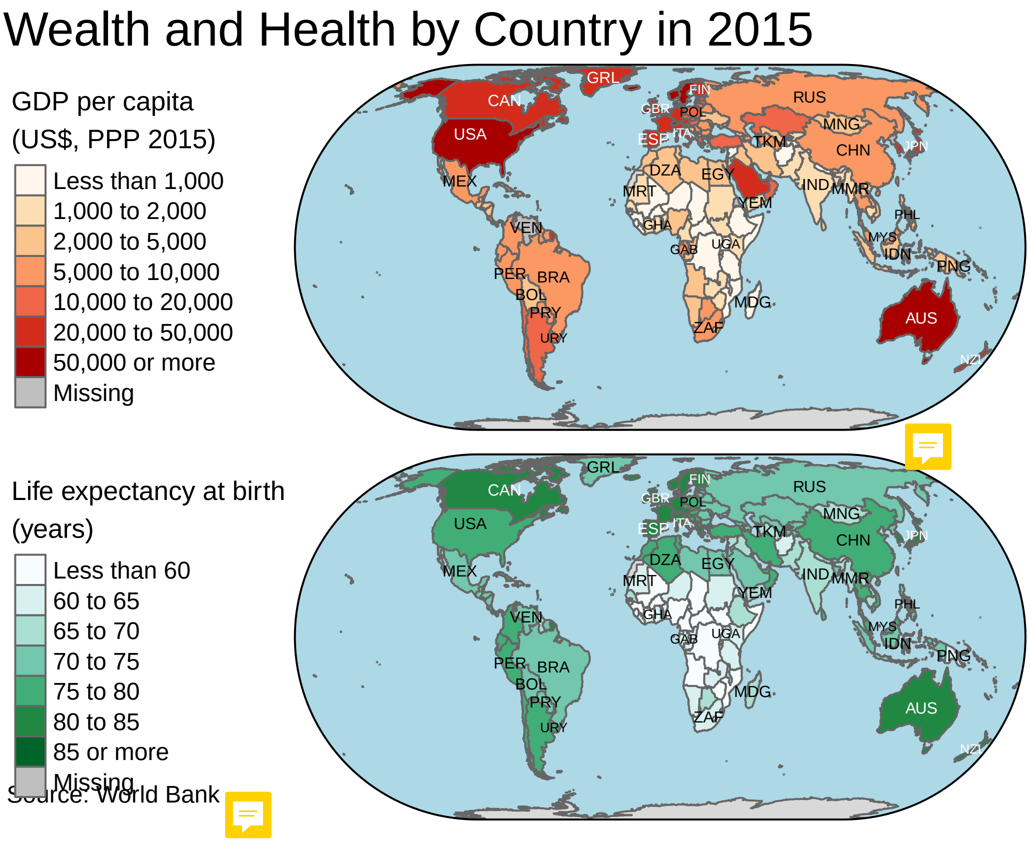
(7) Because we want to make choropleth maps, we should use an equal-area map projection. What is the current projection of the map? Is it an equal-area projection? (Hint: you can find the `proj` string of an `sf` object `x` with `st_crs(x)$proj4string`.)

```
st_crs(land_masses)$proj4string
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

(8)

```
tm_shape(land_masses, projection = "+proj=eck4") +
  tm_polygons() +
  tm_shape(countries, projection = "+proj=eck4") +
  tm_polygons(
    col = c("gdp_per_cap", "life_exp"),
    title = c(
      "GDP per capita\n(US$, PPP 2015)",
      "Life expectancy at birth\n(years)"
    ),
    palette = list("OrRd", "BuGn"),
    breaks = list(
      c(
        -Inf,
        1000,
        2000,
        5000,
        10000,
        20000,
        50000,
        Inf
      ),
      c(
        -Inf,
        60,
        65,
        70,
      75,
      80,
      85,
      Inf
    )
  ),
  labels = list(
    c(
      "Less than 1,000",
      "1,000 to 2,000",
      "2,000 to 5,000",
      "5,000 to 10,000",
      "10,000 to 20,000",
      "20,000 to 50,000",
      "50,000 or more"
    ),
    c(
      "Less than 60",
      "60 to 65",
      "65 to 70",
      "70 to 75",
      "75 to 80",
      "80 to 85",
      "85 or more"
    )
  )
) +
tm_layout(
  main.title = "Wealth and Health by Country in 2015", # https://stackoverflow.com/questions/61355422/
  inner.margins = c(0.01, 0.3, 0.01, 0.01),
  bg.color = "lightblue",
  earth.boundary = TRUE,
  space.color = "white",
  legend.title.size = 1, # https://stackoverflow.com/questions/32890762/how-to-manipulate-tmap-legend
  legend.text.size = 0.8,
  main.title.size = 1.5,
  legend.position = c("left", "top"),
  frame = FALSE # https://stackoverflow.com/questions/58122122/remove-customize-facets-borders-in-tmap
) +
tm_text(
  "code",
  size = "AREA",
  root = 5,
  remove.overlap = TRUE,
  size.lim = c(0.3, 0.5)
) +
tm_credits(
  text = c("", "Source: World Bank"),
  size = 0.8,
  position = c(0.01, 0.02)
)
```


Wealth and Health by Country in 2015

Source: World Bank

(9) What can a reader conclude from the maps? A reader can conclude that the wealth and health of countries in Africa are low compared to those of the rest of the world. Similarly, their life expectancy is also generally lower compared to those in other continents. Given this, readers can conclude that wealth and health and life expectancy have a positive relationship with one another, especially given that countries in North America have both high wealth and health and life expectancy.

(10) What are the advantages and disadvantages of this visualisation in comparison to figure 1? The advantage of this visualization compared to figure 1 is that the points representing each country do not overlap with one another, making it easier to see the data for a single country. However, it becomes harder to compare each country with one another as the shades representing different categories are quite similar. Furthermore, while figure 1 plots these countries on the x and y axes where every number is represented, the countries in the choropleth map are put into categories. Hence, one cannot tell the exact value for GDP and life expectancy that a country has.