Assignment #1 - Data Wrangling in R

François d'Entremont

2022-10-19

First let's load all the packages needed for the assignment

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(tmap)
library(sp)
```

Now let's load the data and take a look at it. The result is a tibble so we can use our libraries on them.

```
# load world data using csv file
world_data <- read_csv("world_data.csv")
```

```
## Rows: 108 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): Country, Landlocked, Religion, Region
## dbl (12): Population, Area, Urban Population (%), Life Expectancy (Female), ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(world_data)
class(world_data)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Task #1 - Subsetting Rows and Columns

We are looking for countries located in Eastern Europe with male life expectancy greater than 68 years and have negative population growth.

```
world_data_subset_piping <- world_data %>% filter(
    Region == "East Europe", `Life Expectancy (Male)` > 68,
    `Population Increase (% per year)` < 0) %>% select(
      Country, Region, `Life Expectancy (Male)`,
      `Population Increase (% per year)`)

View(world_data_subset_piping)
world_data_subset_piping
```

```
## # A tibble: 2 x 4
##   Country  Region      `Life Expectancy (Male)` Population Increase (% per yea~1
##   <chr>    <chr>                          <dbl>                             <dbl>
## 1 Bulgaria East Europe                     69                              -0.2
## 2 Croatia  East Europe                     70                              -0.1
## # ... with abbreviated variable name 1: `Population Increase (% per year)`
```

The chunk of code above pipes the data into the filter function and then pipes that data into a select function to display the relevant columns.

Task #2 - Summary Statistics at the Regional Scale

Calculate the male to female life expectancy ratio and output a grouped summary of the average male to female life expectancy at the regional scale.

```
world_data_m2fer <- world_data %>% group_by(Region) %>% summarise(
  `Male to Female Life Expectancy Ratio` = mean(
    `Life Expectancy (Male)` / `Life Expectancy (Female)`))

View(world_data_m2fer)
world_data_m2fer
```

```
## # A tibble: 6 x 2
##   Region        `Male to Female Life Expectancy Ratio`
##   <chr>                                          <dbl>
## 1 Africa                                         0.937
## 2 East Europe                                    0.891
## 3 Latin America                                  0.923
## 4 Middle East                                    0.941
## 5 OECD                                           0.920
## 6 Pacific/Asia                                   0.946
```

Task #3 - Mapping Spatial Data in R

Loading the world dataset from the shapefile and viewing it

```
world <- st_read("world.shp")
```

```
## Reading layer ‘world’ from data source
##   ‘C:\Users\w0483484\OneDrive - Nova Scotia Community College\GDAA1001_4637\Assignment1\world.shp’
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 108 features and 1 field
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -15679260 ymin: -6224955 xmax: 15351770 ymax: 9252194
## Projected CRS: World_Winkel_II
```

```
View(world)
class(world)
```

```
## [1] "sf"         "data.frame"
```

We see that the column with the country names doesn't match with the column Country in world_data so we'll change it so we can merge them together. We also see that it's a data.frame so we can't use our packages on it.

```
# We must give the columns the same name before we join them
names(world)[1] = "Country"

View(world)
```

Let's check if the entries in the country columns match.

```
sum(world$Country == world_data$Country) == 108
```

```
## [1] TRUE
```

All 108 entries match!

Now let's do the join! We can join a data.frame to a tibble using the inner_join function and the result will be a tibble.

```
world_join <- inner_join(world_data, world)
```

```
## Joining, by = "Country"
```

```
View(world_join)
class(world_join)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Removing all the countries with male to female literacy rates and daily caloric intake values of zero with a single filter function

```
world_join_filter <- filter(world_join, `Literacy Rate for Males (%)` != 0,
                            `Literacy Rate for Females (%)` != 0,
                            `Daily Calorie Intake` != 0)

View(world_join_filter)
```

We see that there are now 59 entries. We removed almost half the countries.

We will now calculate the population density for the remaining countries. We will add a column called Population Density. This column will store values in persons per square kilometre

```
world_join_filter$`Population Density` <-
  world_join_filter$Population / world_join_filter$Area

View(select(world_join_filter, Country, `Population Density`))
```

We will now calculate the male to female literacy ratio for the remaining countries. We will add a column called Literacy Ratio.

```
world_join_filter$`Literacy Ratio` <-
  world_join_filter$`Literacy Rate for Males (%)` /
  world_join_filter$`Literacy Rate for Females (%)`

View(select(world_join_filter, Country, `Literacy Rate for Males (%)`,
            `Literacy Rate for Females (%)`, `Literacy Ratio`))
```

Now we need to find the top 5 countries in terms of population density, in descending order.

```
world_join_top5density <- select(head(arrange(
  world_join_filter, desc(`Population Density`)),5),
  Country, `Population Density`)

world_join_top5density
```

```
## # A tibble: 5 x 2
##   Country     `Population Density`
##   <chr>                      <dbl>
## 1 Singapore                 5368.
## 2 Bangladesh                 872.
## 3 Rwanda                     315.
## 4 India                      290.
## 5 El Salvador                278.
```

Now we need to find the bottom 5 countries in terms of literacy ratio, in descending order.

```
world_join_bot5m2flr <- select(head(arrange(
  world_join_filter, desc(`Literacy Ratio`)),5), Country, `Literacy Ratio`)

world_join_bot5m2flr
```

```
## # A tibble: 5 x 2
```

```
##   Country                   `Literacy Ratio`
##   <chr>                                <dbl>
## 1 Burkina Faso                          3.11
## 2 Somalia                               2.57
## 3 Central African Republic              2.2
## 4 Cambodia                              2.18
## 5 Bangladesh                            2.14
```

Now we will produce two choropleth graphs showing Population Density and Male to Female Literacy

Checking the class when using the st_as_sf function on our data

```
class(st_as_sf(world_join_filter))
```

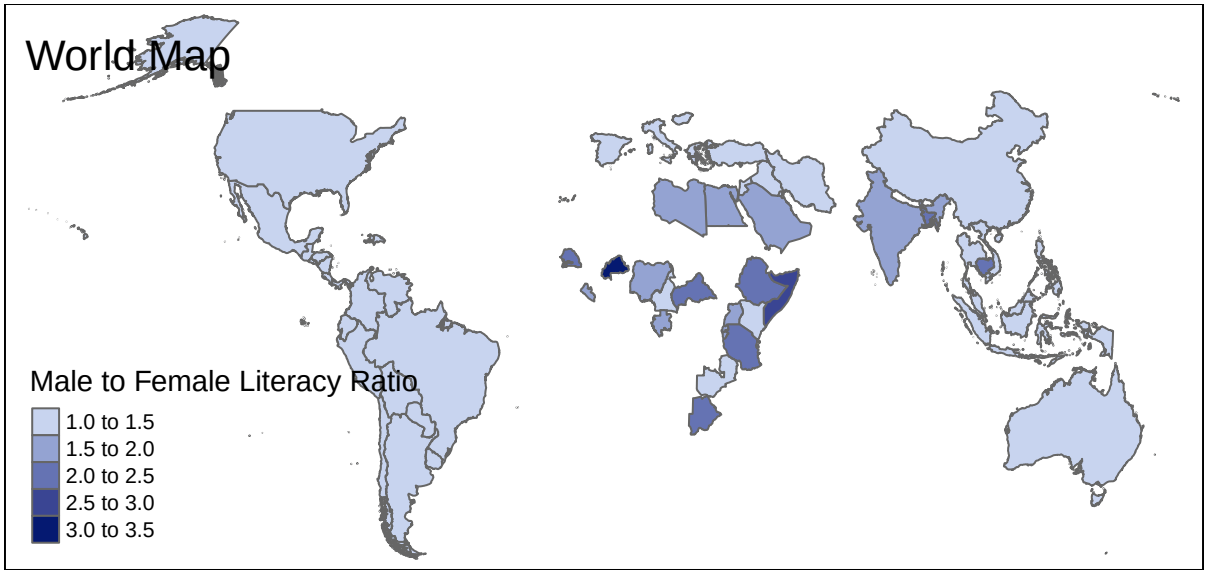
```
## [1] "sf"          "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

We must convert our dataset to an sf object to create the graphs.

Below are two choropleth maps with one showing Male to Female Literacy ratio and the other showing Population Density.

```
M2F_Literacy_Ratio_Graph <- tm_shape(st_as_sf(world_join_filter)) +
  tm_fill("Literacy Ratio", title = "Male to Female Literacy Ratio",
          palette = c("#c8d4ef", "#94a3d2", "#6573b3", "#3a4593", "#051971")) +
  tm_borders() + tm_layout("World Map")

M2F_Literacy_Ratio_Graph
```



```
Population_Density_Graph <- tm_shape(st_as_sf(world_join_filter)) +
  tm_fill("Population Density", title = "Population Density",
          breaks = c(0, 50, 200, 800, 6000),
          palette = c("#d5e8dd", "#9bd4b2", "#45a36b", "#0b4a25")) +
  tm_borders() + tm_layout("World Map")

Population_Density_Graph
```