Exercises for Lecture 13

Statistical Computing with R, 2022-23

Exercise 1

Load the `heights` dataset and convert it to a data frame:

```
library(brolgar)
df_long = as.data.frame(heights)
```

1. Select observations for European countries obtained between 1870 and 1930
2. Draw a spaghetti plot of `height_cm` versus `year`. Given the large number of countries, use the `cex.leg` argument to reduce the size of font in the legend. Moreover, use the `legend.inset` argument to move the position of the legend in such a way that it appears to the right of the chart.

Exercise 2

1. Convert the `df_long` created in exercise 1 from long to wide format.
2. Check the new variable names. If some of those new names are numbers (e.g., `1870`), edit them by adding `height_` before the year (e.g., `height_1870`).
3. Draw a scatter plot with height in 1870 on the x axis, and height in 1930 on the y axis. Add to the plot the line $y = x$ as a dashed, red line of width (`lwd`) 2.
4. Do you notice any pattern? Did mean height increase during that period? Are there any exceptions?

Exercise 3

In exercise 6 of lecture 3 you imported data on population by country from the World Bank website, and saved the imported data into two data frames called `population` and `metadata`. In this exercise, we will merge the information contained in these two data frames.

1. Retrieve the `.RData` file that you created in lecture 3, and which contains `population` and `metadata`.

2. What type of merge do you need to perform to join the two data frames (horizontal, or vertical)?
3. Identify the variable / variables that can be used as key(s) for the merge.
4. Merge the two data frames.
5. Remove from the merged data frame those rows for which `Region` is missing. Note that here missing values might have been denoted by an empty string of length 0, rather than by properly formatted `NA`s.
6. For each region, find a) the largest population value in 2020 and b) the country it corresponds to.

Exercise 4

Consider the following chunck of code:

```
data(iris)
df.list = split(iris, iris$Species)
lapply(df.list, summary)
```

1. Rewrite the code using the pipe operator `|>`.
2. Rewrite the code using the pipe operator `%>%`.

Exercise 5

1. Load and clean the `irish_polls` data that was used in the exercises of lecture 5.
2. Subset the data such that it includes all polls that ended on 27-05-2021 up until and including 09-09-2021. Include all parties that do not have any missing values.
3. Transform the data to the long format. Additionally, transform the data type of `Fieldwork.End` to `Date`.
4. Make a spaghetti plot of the percentage of votes for each of the parties over time. Make sure that the plot is nicely formatted, think of titles, legends, colors, etc.
5. Calculate the mean percentage for each party individually using piping.
6. Which party had the highest mean vote percentage? Look at the spaghetti plot you made, is this reflected there as well?