Exercise #03

Fortgeschrittene Statistische Software für NF - SS 2022/23

C. Ulrike Jooss (Matrikel-Nr. 17000859)

2023-05-27

Exercise 1: Initializing git

For this whole exercise sheet we will be tracking all our changes to it in git.

Erxercise 1a)

Start by initializing a new R project with git support, called `exeRcise-sheet-3`. If you forgot how to do this, you can follow this guide:

My Public Repository: https://github.com/UlrikeJooss/exeRcise-sheet-3.git

Erxercise 1b)

Commit the files generated by Rstudio:

Initial Commit: ExerciseSheet_3_TEMPLATE, named ex03_Jooss.Rmd

Erxercise 1c)

For all of the following tasks in this exercise sheet we ask you to always commit your changes after finishing each subtask e.g. create a commit after task *1d*, *1e* etc.

Erxercise 1d)

Name 2 strengths and 2 weaknesses of git. (Don't forget to create a commit after this answer, see *1c*):

Strengths of Git:

1. Distributed Version Control: Git is a distributed version control system, which means that every user has a complete copy of the repository, including its entire history. This allows for offline work, easy branching and merging, and provides resilience in case of server failures.
2. Fast and Efficient: Git is designed to be fast and efficient, even with large codebases. It uses advanced algorithms for data storage and retrieval, allowing for quick operations such as commits, branching, and merging. Additionally, Git performs most operations locally, reducing the need for network communication.

Weaknesses of Git:

1. Steep Learning Curve: Git has a steep learning curve, especially for users who are new to version control systems. Understanding the concepts of branches, commits, and merges can be challenging initially, and it may take time and practice to become proficient in using Git effectively.
2. Complex Command Line Interface: While Git provides a command line interface, some users may find it overwhelming or less user-friendly compared to graphical user interfaces (GUIs). The command-line nature of Git requires users to remember and execute specific commands accurately, which can be daunting for beginners.

Erxercise 1e)

Knit this exercise sheet. Some new files will automatically be generated when knitting the sheet e.g. the HTML page. Ignore these files, as we only want to track the source files themselves.

Exercise 2: Putting your Repository on GitHub

For this task you will upload your solution to GitHub.

a) Create a new repository on GitHub in your account name `exeRcise-sheet-3`. Make sure you create a **public repository** so we are able to see it for grading. Add the link to the repository below:

My Public Repository: https://github.com/UlrikeJooss/exeRcise-sheet-3.git

b) Push your code to this new repository by copying and executing the snippet on github listed under `...or push an existing repository from the command line`: DONE

c) Regularly push your latest changes to GitHub again and especially do so when you are finished with this sheet: Always Doing

Exercise 3: Baby-Names in Munich

Download the latest open datasets on given names ("Vornamen") from the open data repository of the city of Munich for the years `2022` and `2021`.

Link: https://opendata.muenchen.de/dataset/vornamen-von-neugeborenen

Exercise 3a)

Download the data for both years and track it in git. For small datasets like these adding them to git is not a problem:

Uploaded to Github and Pulled Down into may Project.

Exercise 3b)

Load the data for both years into R. Check the type of the count variable ("Anzahl") and look into the data to determine why it is not numeric? Fix the problem in an appropriate manner, it is OK if some of the counts are inaccurate because of this. Explain your solution and the repercussions.

```r
# Loading Datasets:
library(readr)

vornamen_2021 <- read_csv("vornamen_2021.csv")
vornamen_2022 <- read_csv("open_data_portal_2022.csv")

typeof(vornamen_2021$Anzahl) # character
```

```
## [1] "character"
```

```r
typeof(vornamen_2022$Anzahl) # character
```

```
## [1] "character"
```

```r
library(tidyverse)

vornamen_2021 <- vornamen_2021 %>%
    mutate(Anzahl = as.numeric(Anzahl))
typeof(vornamen_2021$Anzahl) # double
```

```
## [1] "double"
```

```r
vornamen_2022 <- vornamen_2022 %>%
    mutate(Anzahl = as.numeric(Anzahl))
typeof(vornamen_2022$Anzahl) # double
```

```
## [1] "double"
```

In the Excel file, all variables of the observations are stored in one cell with the character data type. When importing with read_csv(), the variables are correctly divided into columns, but they all retain the character data type. The type character is exactly correct for name and sex, but not for number. So I converted it with the as.numeric() function to double.

Exercise 3c)

Calculate the total number of babies born in Munich in 2022 and 2021. Which year had the bigger baby-boom?

```r
# 2021

#is.na(vornamen_2021$Anzahl)

vornamen_2021$Anzahl <- ifelse(
    is.na(vornamen_2021$Anzahl), 0, vornamen_2021$Anzahl)

total_births_2021 <- sum(vornamen_2021$Anzahl)
total_births_2021 # 11524
```

```
## [1] 11524
```

```r
# 2022

vornamen_2022$Anzahl <- ifelse(
    is.na(vornamen_2022$Anzahl), 0, vornamen_2022$Anzahl)

total_births_2022 <- sum(vornamen_2022$Anzahl)
total_births_2022 # 9899
```

```
## [1] 9899
```

```r
# compare:

compare_births <- function(total_births_2021, total_births_2022) {

  if (total_births_2021 > total_births_2022) {
    result <- "2021 had the bigger baby-boom."
  } else if (total_births_2022 > total_births_2021) {
    result <- "2022 had the bigger baby-boom."
  } else {
    result <- "Both years had an equal number of births."
  }
  return(result)
}

compare_births(total_births_2021, total_births_2022)
```

```
## [1] "2021 had the bigger baby-boom."
```

Exercise 3d)

Add a new column `year` to both datasets which holds the correct year for each:

```r
vornamen_2021 <- vornamen_2021 %>%
  mutate(year = 2021)

vornamen_2022 <- vornamen_2022 %>%
  mutate(year = 2022)
```

Exercise 3e)

Combine both datasets into one using `bind_rows()`:

```r
comb_data <- bind_rows(vornamen_2021, vornamen_2022)
```

Exercise 3f)

Combine the counts for same names to determine the most popular names across both years. Print out the top 10 names in a nicely formatted table for both years. Include a table caption.

```
knitr::opts_chunk$set(label = "top10")

name_counts <- comb_data %>%
  group_by(Vorname) %>%
  summarize(count = sum(Anzahl)) %>%
  arrange(desc(count)) %>%
  slice(1:10)

name_counts <- as.tibble(name_counts)
```

```
## Warning: ‘as.tibble()‘ was deprecated in tibble 2.0.0.
## i Please use ‘as_tibble()‘ instead.
## i The signature and semantics have changed, see ‘?as_tibble‘.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
knitr::kable(name_counts,
col.names = c("Name","Number"),
caption = "<b>Top 10 Most Popular Names in 2021 and 2022</b>",
  align = c("l","r"),
  font_size = 12,
  row.names = TRUE)
```

Table 1: Top 10 Most Popular Names in 2021 and 2022

	Name	Number
1	Maximilian	240
2	Emilia	234
3	Felix	220
4	Anton	206
5	Emma	199
6	Leon	195
7	Noah	185
8	Jakob	180
9	Anna	178
10	Lukas	173

Exercise 4: Chat GPT + apply (3 points)

For this task: Specifically use ChatGPT to solve the task and submit your prompts in addition to the solution

a) The code below does not work because the wrong apply function has been used. Find out which apply function would be correct and why it did not work. Correct the code. Also calculate the rowwise means.

###Create a sample data frame

tax_data <- data.frame(Name = c(“Munich GmbH”, “ABC Inc.”, “Backpacks 1980”, “Bavarian Circus”), Tax_2019 = c(5000, 4000, 6000, 3500), Tax_2020 = c(4800, 4200, 5800, 3700), Tax_2021 = c(5200, 3800, 5900, 3400))

###Calculate column-wise means

column_means <- lapply(tax_data

$$, -1$$

, 2, mean)

column_means

b) Using ChatGPT try to understand what the rapply() function does. Create an easy example with mock data where the function is used and explain it in your words.

Final Note

Make sure to push all your commits and changes to GitHub before submittining the exercise sheet.