Portfolio 3

Sigurd Sørensen

2022-08-31

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.5   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

Instructions: For this assignment, you need to answer a couple questions with code and then take a screenshot of your working environment. Submit the solutions including the URL to the screenshot in a doc/pdf to Brightspace.

1. Use R to figure out how many elements in the vector below are greater than 2 and then tell me what their sum (of the larger than 2 elements) is. rooms <- c(1, 2, 4, 5, 1, 3, 1, NA, 3, 1, 3, 2, 1, NA, 1, 8, 3, 1, 4, NA, 1, 3, 1, 2, 1, 7, 1, 9, 3, NA)

rooms <- c(1, 2, 4, 5, 1, 3, 1, NA, 3, 1, 3, 2, 1, NA, 1, 8, 3, 1, 4, NA, 1, 3, 1, 2, 1, 7, 1, 9, 3, NA)  
  
c(rooms > 2, complete.cases(rooms))

## [1] FALSE FALSE TRUE TRUE FALSE TRUE FALSE NA TRUE FALSE TRUE FALSE  
## [13] FALSE NA FALSE TRUE TRUE FALSE TRUE NA FALSE TRUE FALSE FALSE  
## [25] FALSE TRUE FALSE TRUE TRUE NA TRUE TRUE TRUE TRUE TRUE TRUE  
## [37] TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE  
## [49] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE

rooms[rooms >2 & !is.na(rooms)] %>%   
 length()

## [1] 12

rooms[rooms >2 & !is.na(rooms)] %>%   
 sum()

## [1] 55

**Answer:** 12 rooms > 2 and their sum is 55.

1. What type of data is in the ‘rooms’ vector?

class(rooms)

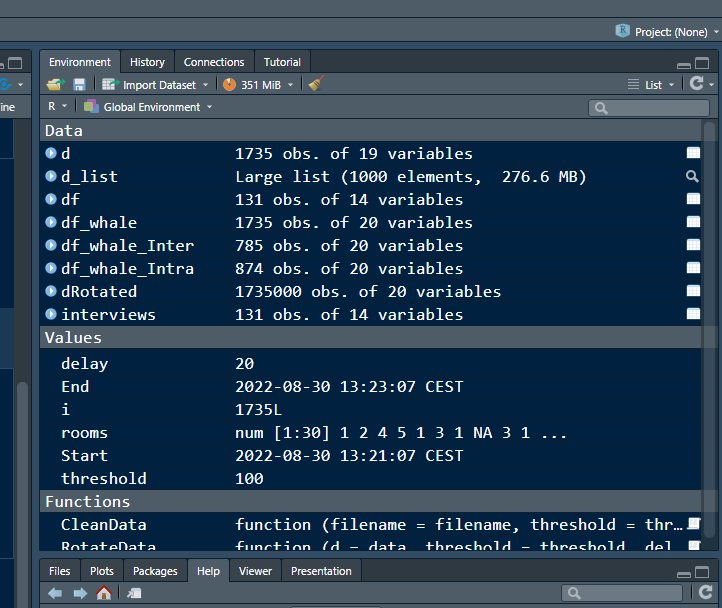
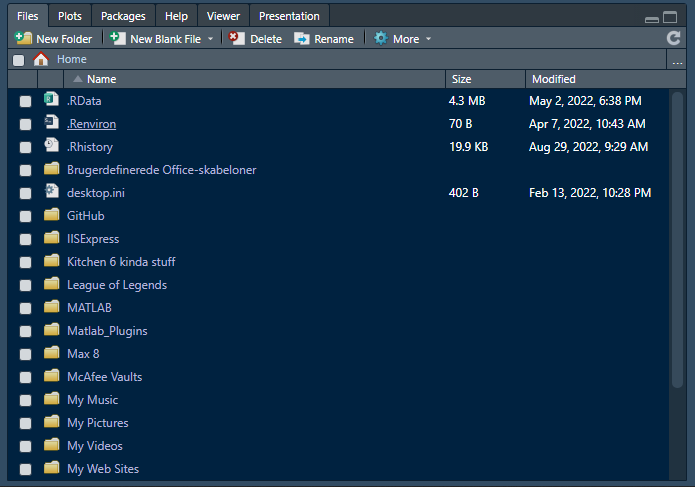
## [1] "numeric"

**Answer** Numeric but more specifically integers.

1. Submit the following image to Github: Inside your R Project (.Rproj), install the ‘tidyverse’ package and use the download.file() and read\_csv() function to read the SAFI\_clean.csv dataset into your R project as ‘interviews’ digital object (see instructions in <https://datacarpentry.org/r-socialsci/setup.html> and ‘Starting with Data’ section). Take a screenshot of your RStudio interface showing
2. the line of code you used to create the object,

interviews <- read\_csv("SAFI\_clean.csv")

## Rows: 131 Columns: 14  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (7): village, respondent\_wall\_type, memb\_assoc, affect\_conflicts, items...  
## dbl (6): key\_ID, no\_membrs, years\_liv, rooms, liv\_count, no\_meals  
## dttm (1): interview\_date  
##   
## 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.

1. the ‘interviews’ object in the Environment, and 
2. the file structure of your R project in the bottom right “Files” pane. Save the screenshot as an image and put it in your AUID\_lastname\_firstname repository inside our Github organisation (github.com/Digital-Methods-HASS) or equivalent. Place here the URL leading to the screenshot in your repository. 
3. Challenge: If you managed to create your own Danish king dataset, use it. If not, you the one attached to this assignment (it might need to be cleaned up a bit). Load the dataset into R as a tibble. Calculate the mean() and median() duration of rule over time and find the three monarchs ruling the longest. How many days did they rule (accounting for transition year?)

The data uses , delimeters so an easy solution would be to use read\_csv2() which be default uses , delimters.

## ℹ Using "','" as decimal and "'.'" as grouping mark. Use `read\_delim()` for more control.

## Rows: 47 Columns: 4  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ";"  
## chr (2): Kings, Yearasruler  
## dbl (2): Start\_date, End\_date  
##   
## 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.

But a way more fun and hacky way is this way too complicated work around. :)

## Rows: 47 Columns: 1  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): Kings;Start\_date;End\_date;Yearasruler  
##   
## 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.

## # A tibble: 6 × 1  
## `Kings;Start\_date;End\_date;Yearasruler`  
## <chr>   
## 1 "Gorm den Gamle;NA;NA;Unknown"   
## 2 "Harald 1. Bl\xe5tand ;NA;NA;Unknown"   
## 3 "Svend 1. Tvesk\xe6g ;NA;NA;Unknown"   
## 4 "Harald 2.;1014;1018;4"   
## 5 "Knud 1. den Store;1018;1035;17"   
## 6 "Hardeknud;1035;1042;7"

This is not readable or useful at the moment. We will have to clean it up.

newnames <- strsplit(colnames(df\_kings), split = ";")[[1]]  
  
df\_kings <- df\_kings %>%   
 separate(colnames(df\_kings)[1], newnames, sep = ";") %>%  
 filter(Yearasruler != "Unknown") %>%   
 mutate\_at(c("Start\_date", "End\_date", "Yearasruler"), as.numeric)  
  
head(df\_kings)

## # A tibble: 6 × 4  
## Kings Start\_date End\_date Yearasruler  
## <chr> <dbl> <dbl> <dbl>  
## 1 "Harald 2." 1014 1018 4  
## 2 "Knud 1. den Store" 1018 1035 17  
## 3 "Hardeknud" 1035 1042 7  
## 4 "Magnus den Gode " 1042 1047 5  
## 5 "Svend 2. Estridsen" 1047 1074 27  
## 6 "Harald 3. Hen " 1074 1080 6

Voila a few magic tricks and we get the data in tidyformat. We can now run our stats on it.

summary(df\_kings$Yearasruler)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 7.75 14.00 18.68 29.25 60.00

df\_kings %>%   
 arrange(desc(Yearasruler))

## # A tibble: 44 × 4  
## Kings Start\_date End\_date Yearasruler  
## <chr> <dbl> <dbl> <dbl>  
## 1 "Christian 4. " 1588 1648 60  
## 2 "Erik 7. af Pommern" 1396 1439 43  
## 3 "Christian 7. " 1766 1808 42  
## 4 "Valdemar 2. Sejr " 1202 1241 39  
## 5 "Erik 6. Menved" 1286 1319 35  
## 6 "Valdemar 4. Atterdag " 1340 1375 35  
## 7 "Chrstian 1." 1448 1481 33  
## 8 "Hans " 1482 1513 31  
## 9 "Frederik 4. " 1699 1730 31  
## 10 "Frederik 6. " 1808 1839 31  
## # … with 34 more rows

sort(df\_kings$Yearasruler, TRUE)[1:3]

## [1] 60 43 42