Data Handling: Import, Cleaning and Visualisation

Exercise to lecture 4

AUTHOR
Dr. Aurélien Sallin

Working with a data frame

Set your script

Set your `R` Script.

```
########################################################################
# Data Handling Course: Example Script for Data Gathering and Import
#
# Imports data from ...
# Input: import c to data sources (data comes in ... format)
# Output: cleaned data as CSV
#
# A. Sallin, St. Gallen, 2024
########################################################################

# SET UP --------------
# load packages
library(readr)
library(stringi)


# SET PATH ------------------
# financial_data <- read.csv("Path/to/my/file/financial_data.txt")
```

Import data

Have a look at the file `financial_data.txt` using your favorite text editor. What do you notice?

Import the table using the `read.csv()` function in your environment. Make sure you have the right path to access the .txt document. What does this parser do? Explore the data.frame. What is its structure? What are its dimensions?

Summary statistics of your data

Compute the summary statistics for each variable using the `summary()` command. What does this command give you? What do you notice? Make the necessary changes.

```
# Check summary again
summary(financial_data)
```

Variable creation

Create a new variable “costs”, which is the revenue - profit. [There are many ways to create a variable in a data frame. Here, use the `$` index.]

Factor variable

Which variable is (should be) a factor? Recode this variable as a factor. What are the levels? Should we have the variable `Firm` as a factor?

Nests - more difficult question... but still exam relevant 

Split your data using the factor variable into three data frames that are contained in a list. Compute the mean profit for each data frame.

- Hint: use the function `split`.
- Hint: use a `for-loop` over each list element to compute the mean

Functionals - `map`

Do the same as the exercise above using the `map` function. Install the packages `tidyr`, `dplyr`, and `purrr`.