Exercise 1

(a) The files are in standard csv format. This means that we can use `read.csv` to read in the data for March 18. We have to set `stringsAsFactors = FALSE`.

```
data_corona <- read.csv(
    "./data/raw/03-18-2020.csv",
    stringsAsFactors = FALSE
)
```

Please note that we have used a relative file path here. We always assume that we are in the root directory of our project.

(b) As usual you can use `str` and `head` to get an impression about the data.

(c) With (b) we know that the format of the `Last.Update` column looks like `2020-03-18T12:13:09`. When we look on the help page for `strptime` we see that the year can be encoded with `%Y`, a month consisting of two figures is `%m` and the day of a month with two figures is `%d`. So we can convert the column to dates as follows:

```
data_corona$Last.Update <- as.Date(data_corona$Last.Update, "%Y-%m-%d")
```

We use the subsetting operator `$` to select and overwrite the respective column with the result of the right-hand function call. Please note that we do not have to further specify anything after the day, i.e., `T12:13:09` is automatically ignored.

(d) We can export the data frame as a CSV file with `write.csv`. For exporting to Excel files we have to use a third-party package like `open.xlsx`.

```
library("open.xlsx")

write.csv(data_corona, "./data/processed/processed_data.csv", row.names = FALSE)
write.xlsx(data_corona, "./data/processed/processed_data.xlsx")
```

So your `analysis.R` file should look like below.

```
library("open.xlsx")

data_corona <- read.csv(
    "./data/raw/03-18-2020.csv",
    stringsAsFactors = FALSE
)

data_corona$Last.Update <- as.Date(data_corona$Last.Update, "%Y-%m-%d")

write.csv(data_corona, "./data/processed/processed_data.csv", row.names = FALSE)
write.xlsx(data_corona, "./data/processed/processed_data.xlsx")
```