Exercise 1

Since the project is growing every week and the exercises become more and more inter-related I've put the code for the solution in the actual analysis file in the project folder. You can use that file for the solutions and as a check-point for your own analysis. I will just discus the code snippets here.

(a) Pleae note that I have renamed the data frame from `data_corona` to `data_20200318`. As described on the help page for `aggregate` we can apply the function to several columns by using `cbind`.

```
sum_country <- aggregate(
  cbind(Confirmed, Deaths, Recovered) ~ Country.Region,
  data = data_20200318,
  sum
)
```

Here I use the formula interface for defining the columns that I want to use for grouping and the columns to which the functions shall be applied.

(b) In the very same way we can determine the date of the latest update for each country. Since dates in R are integers we can use the function `max` to determine the latest date.

```
update_country <- aggregate(
  Last.Update ~ Country.Region,
  data = data_20200318,
  max
)
```

(c) We can now easily merge both data frames. The easiest way is to use `merge`. It will automatically match the correct rows by the country column. Since both the respective column has the very same name in both data frames we do not have to specify it explicitly. However, we can do so in order to make the code easier understandable.

```
data_20200318 <- merge(sum_country, update_country)
```

(d) We can answer this question by first filtering out all rows with less than 1000 confirmed cases and then count the number of remaining rows. This works since for now we are only looking at data from a single date.

```
nrow(subset(data_20200318, Confirmed >= 1000))
```

This should yield the number `16`. Please note that `length` does not work here even though we would intuitively define the length of a data frame as the number of rows. But we have to remember here that a data frame is just a list. And the length of the list is the number of elements. Since a data frame holds one vector for each column `length` would return the number of columns.