ITA-2017 R-Tutorial

Hands-On Exercise 2 - data.frame

1. Download the file islands.txt from http://www.smiffy.de/ita-2017/islands.txt and store it on your disk.
2. Import the file into a data frame with name 'islands', using the following command (you have to adapt the path of the file):
```
path <- "d:/somewhere-on-your-disk/islands.txt"
islands <- read.table(
      path,
      header=TRUE,
      stringsAsFactors=FALSE,
      sep="\t")
```
3. Check the content with the commands
```
head(islands)
str(islands)
```
4. Substitute the „NULL"-values for the column Longitude by the NA value.
```
islands$Longitude[islands$Longitude=="NULL"]<-NA
```
5. Transform the datatype of the Longitude column to numeric, using the following command:
```
islands<-transform(islands, Longitude=as.numeric(Longitude))
```
6. Perform the last two steps also on the columns Latitude and Area
```
...
```
7. Add the island „Kauai", which belongs to the „Hawaii Islands" to the data.frame `islands`.
```
islands<-rbind(islands, c("Kauai","Hawaii Islands", NA, NA, NA))
```
8. List all islands which belong to the Hawaiian Islands
```
islands[grep('.*Hawaii.*', islands$Islands),]
```
9. Formulate the following queries:
   - How many islands are in the data frame?
   - Return the islands which reside between the 15th and 25th latitude
10. Count, how many islands, belong to the different island-groups (column Islands)
```
islands_per_group<-aggregate(islands$Name,by=list(islands$Islands),
                             FUN=length)
```

11. Print the island-groups together with the number of islands, which belong to them, ordered by the number of islands (decreasing)

```
islands_per_group[order(islands_per_group$x,decreasing=T),]
```

12. Accessing a database:
    Download the mysql-jdbc-driver from https://dev.mysql.com/downloads/file/?id=472395 and store it locally. Extract the mysql-connector-x.y.z-bin.jar file and store it somewhere on your disk.
13. Run the following code to install the RJDBC-library:

```
if (!require("RJDBC"))
    install.packages("RJDBC", dependencies=T)
library(RJDBC)
```

14. Access the mysql-database 'mondial' and return all cities from the United States, ordered by decreasing population (adapt the path to the jdbc-jar file from step 12.)

```
drv<-JDBC("com.mysql.jdbc.Driver",
  "c:/temp/mysql-connector-java-5.1.43-bin.jar")
conn<-dbConnect(drv, "jdbc:mysql://10.33.3.13/mondial",
                      "mondial_readonly","mondial_readonly")
data<-dbGetQuery(conn, "select name, population
                         from city
                        where country='USA'
                        order by population desc");
str(data)
data
```