Statistics with R - Exercise 2

Philipp Satlawa - h0640348

04/12/2020

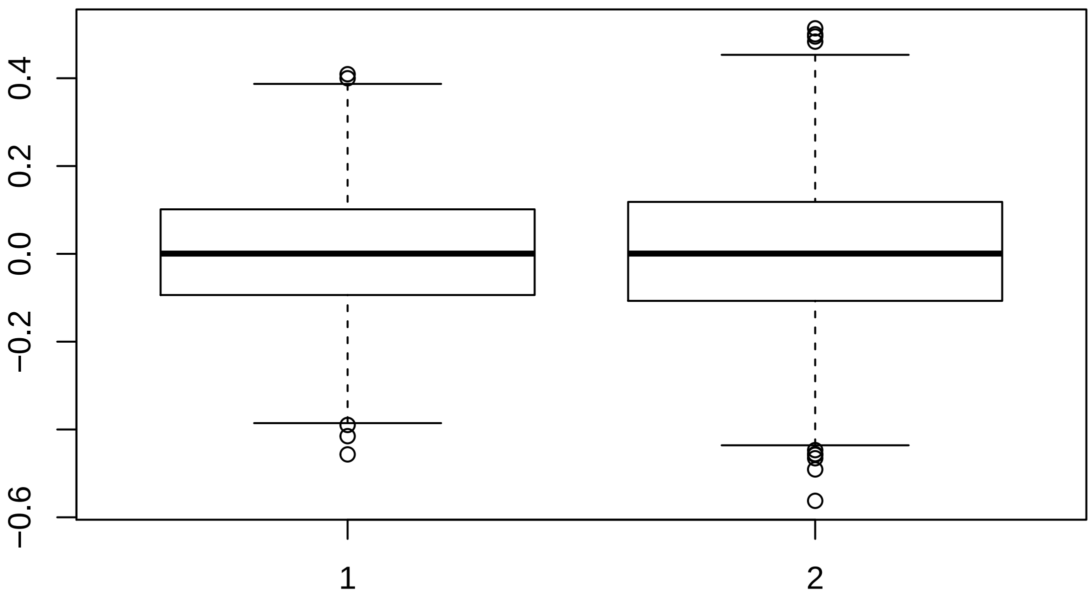
This document contains the answered questions of exercise 2 for the course "Statistics with R".

```r
# import necessary libraries
library("matrixStats")
library("readxl")
library("data.table")
```

Task 1 - Robustness of mean/median

1. Compare mean and median - without outliers

```r
set.seed(640348)
x <- rnorm(n = 50 * 1000, mean = 0, sd = 1)
# matrix with 1000 rows (N) and 50 columns (n)
X <- matrix(x, ncol = 50)

# calculate the mean for every sample
xAvg <- rowMeans(X)

# calculate the median for every sample
xMed <- rowMedians(X)

boxplot(xAvg, xMed)
```



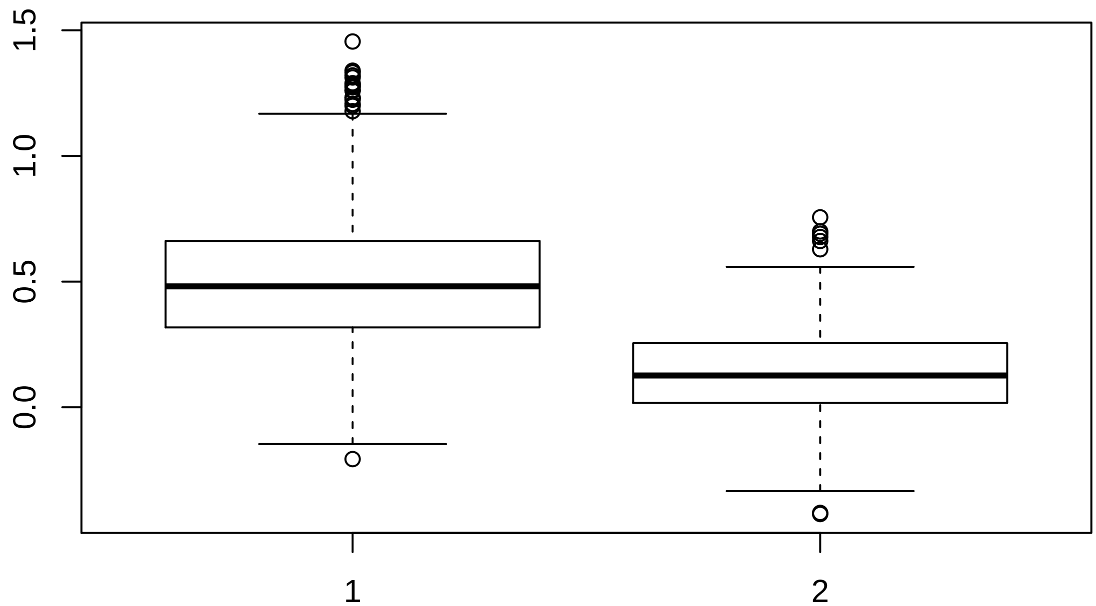

```
#ggplot(mpg, aes(class, hwy)) + geom_boxplot()
```

The mean estimation method provides a better fit to the underlying normally distributed population, since all random values are produced with a normally distributed random generator.

2. Compare mean and median - with outliers

```
set.seed(640348)

# sample from normal distribution (95%) and sample from Exponential Distribution (5%)
x95 <- rnorm(n = 45 * 1000, mean = 0, sd = 1)
x05 <- rexp(n = 5 * 1000, rate = 0.2)

# rearrange vector to a matrix
x95 <- matrix(x95, ncol = 45)
x05 <- matrix(x05, ncol = 5)

# concatenate the two matrices
X <- cbind(x95,x05)

# calculate the mean for every sample
xAvg <- rowMeans(X)

# calculate the median for every sample
xMed <- rowMedians(X)
```

```r
boxplot(xAvg, xMed)
```



The median estimation method provides a better fit to the randomly produced dataset. In this dataset 5% of the random values were generated by the exponential distribution generator. These values generated by the exponential distribution generator added outliers to the normally distributed dataset. Due to the fact that the median is more robust to data with outliers the median estimation method provides a better estimation of the true mean.

Task 2 - Estimation

1. Standard Cauchy Distribution with diffrent sample sizes

```r
cuy100 = rcauchy(100, location = 0, scale = 1)
cuy5000 = rcauchy(5000, location = 0, scale = 1)
cuy100000 = rcauchy(100000, location = 0, scale = 1)

(mean(cuy100))
```

```
## [1] 0.4018709
```

```r
(mean(cuy5000))
```

```
## [1] 1.124204
```

```r
(mean(cuy100000))
```

```
## [1] -3.065202
```

```r
(var(cuy100))
```

```
## [1] 21.01623
```

```r
(var(cuy5000))
```

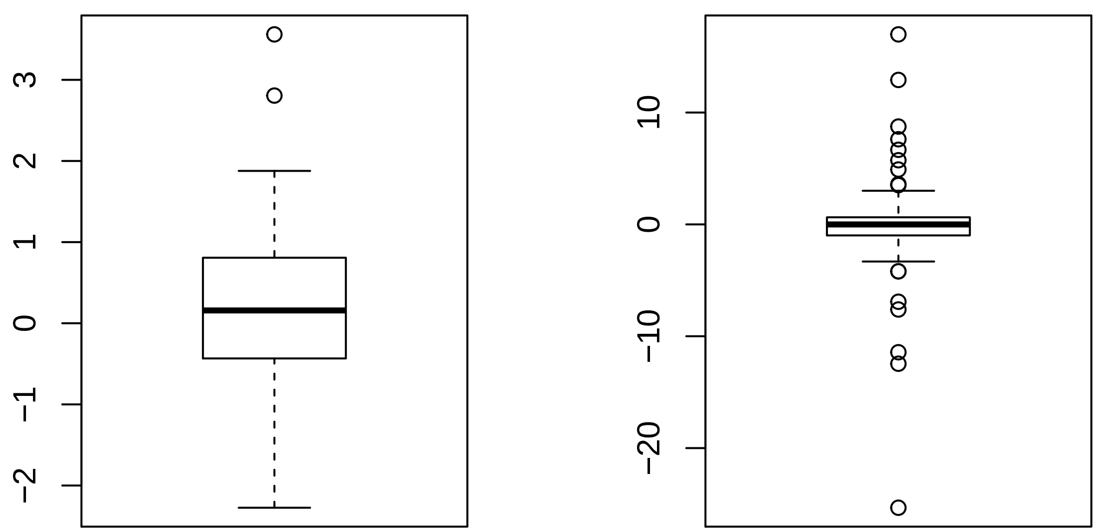
```
## [1] 1905.254
```

```r
(var(cuy100000))
```

```
## [1] 715951.5
```

2.
3. Plotting the standard normal and standard cauchy distribution

```r
# plot the standard normal and standard cauchy distribution as a boxplot
par(mfrow = c(1,2))
boxplot(rnorm(n = 100, mean = 0, sd = 1)) # ... function for normal boxplot
boxplot(rcauchy(n = 100, location = 0, scale = 1)) # ... function for cauchy boxplot
```



```r
par(mfrow = c(1,1))
```

The Standard Cauchy Distribution creates more outliers the bell curve is much wider compared to the normal distribution.

Task 3 - Working with a real data set

```r
#load library data.table
library("data.table")
library("readxl")
library(ggplot2)
```

1. Import data from different sources

import from file .Rdata (R specific file type)

```r
# load data from file
load("~/Documents/boku/statistics_with_R/ex_02/CO2.Rdata")
# convert data.frame to data.table
datCO2 <- setDT(dat)
# check if data was properly loaded
str(dat)
```

```
## Classes 'data.table' and 'data.frame':   1619494 obs. of  6 variables:
##  $ unit     : Factor w/ 2 levels "MIO_T","THS_T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ airpol   : Factor w/ 11 levels "CH4","CH4_CO2E",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ airemsect: Factor w/ 172 levels "CRF1","CRF1-6X4_MEMO",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ geo      : Factor w/ 35 levels "AT","BE","BG",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ time     : num  2017 2017 2017 2017 2017 ...
##  $ values   : num  0.02497 0.04194 0.05682 0.01095 0.00059 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

import from file .txt (tabulator separated values)

```r
# load data from txt-file
data_txt <- read.table(file = "~/Documents/boku/statistics_with_R/ex_02/CO2.txt", header = TRUE, dec =
str(data_txt)
```

```
## 'data.frame':    1619494 obs. of  6 variables:
##  $ unit     : Factor w/ 2 levels "MIO_T","THS_T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ airpol   : Factor w/ 11 levels "CH4","CH4_CO2E",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ airemsect: Factor w/ 172 levels "CRF_INDCO2","CRF1",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ geo      : Factor w/ 35 levels "AT","BE","BG",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ time     : int  2017 2017 2017 2017 2017 2017 2017 2017 2017 2017 ...
##  $ values   : num  0.02497 0.04194 0.05682 0.01095 0.00059 ...
```

import from file .csv (comma separated values)

```r
# load data from csv-file
data_csv <- read.csv(file = "~/Documents/boku/statistics_with_R/ex_02/CO2.csv", header = TRUE)
str(data_csv)
```

```
## 'data.frame':    1619494 obs. of  6 variables:
##  $ unit     : Factor w/ 2 levels "MIO_T","THS_T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ airpol   : Factor w/ 11 levels "CH4","CH4_CO2E",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ airemsect: Factor w/ 172 levels "CRF_INDCO2","CRF1",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ geo      : Factor w/ 35 levels "AT","BE","BG",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ time     : int  2017 2017 2017 2017 2017 2017 2017 2017 2017 2017 ...
##  $ values   : num  0.02497 0.04194 0.05682 0.01095 0.00059 ...
```

import from file .xlsx (MS Excel specific file type)

```r
# list sheets in xlsx-file
excel_sheets("~/Documents/boku/statistics_with_R/ex_02/CO2.xlsx")
```

```
## [1] "Sheet 1"
```

```r
# load data from xlsx-file
data_xlsx <- read_excel("~/Documents/boku/statistics_with_R/ex_02/CO2.xlsx", sheet = "Sheet 1")
# print data structure
str(data_xlsx)
```

```
## tibble [1,619,494 x 6] (S3: tbl_df/tbl/data.frame)
##  $ unit     : chr [1:1619494] "MIO_T" "MIO_T" "MIO_T" "MIO_T" ...
##  $ airpol   : chr [1:1619494] "CH4" "CH4" "CH4" "CH4" ...
##  $ airemsect: chr [1:1619494] "CRF1" "CRF1" "CRF1" "CRF1" ...
##  $ geo      : chr [1:1619494] "AT" "BE" "BG" "CH" ...
##  $ time     : num [1:1619494] 2017 2017 2017 2017 2017 ...
##  $ values   : num [1:1619494] 0.02497 0.04194 0.05682 0.01095 0.00059 ...
```

2. Set seed to the student id

```r
# prepare seed
set.seed(as.numeric(format(Sys.time(), "%H%M%S")))
# set student id
id <- 640348
# set seed with student id
set.seed(id)
```

3. Data exploration

```r
# show the data structure
str(dat)
```

```
## Classes 'data.table' and 'data.frame':   1619494 obs. of  6 variables:
##  $ unit     : Factor w/ 2 levels "MIO_T","THS_T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ airpol   : Factor w/ 11 levels "CH4","CH4_CO2E",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ airemsect: Factor w/ 172 levels "CRF1","CRF1-6X4_MEMO",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ geo      : Factor w/ 35 levels "AT","BE","BG",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ time     : num  2017 2017 2017 2017 2017 ...
##  $ values   : num  0.02497 0.04194 0.05682 0.01095 0.00059 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
# show the data summary
summary(dat)
```

```
##     unit              airpol             airemsect              geo
##  MIO_T:809747   GHG      :249878   CRF1-6XMEMO:  21590   HU      :  57198
##  THS_T:809747   CH4      :247102   CRF2       :  21374   SI      :  52588
##                 CH4_CO2E:247102    CRF2B      :  21298   PL      :  49938
##                 CO2      :232890   CRF2C      :  20040   RO      :  49414
##                 N2O      :229428   CRF2B10    :  19148   ES      :  49118
##                 N2O_CO2E:229428    CRF6       :  18722   EU28    :  48834
##                 (Other) :183666    (Other)    :1497322   (Other):1312404
##       time          values
##  Min.   :1985   Min.   :-458348
##  1st Qu.:1996   1st Qu.:      0
##  Median :2003   Median :      0
##  Mean   :2003   Mean   :   5235
##  3rd Qu.:2010   3rd Qu.:      5
##  Max.   :2017   Max.   :5729428
##                 NA's   :5284
```

The provided dataset contains 6 variables (columns) and 1619494 records (rows). The variables have the following types:

- unit: categorical Factor with 2 levels (unit abriviation)
- airpol: categorical Factor with 11 levels (chemical compound)
- airemsect: categorical Factor with 172 levels (code for the sector)
- geo: categorical Factor with 35 levels (country abrivation)
- time: discrete numerical (year)
- values: continuous numerical (the value)

The variable values has 5284 NA values and is the only variable that contains NA.

4. Select randomly two countries

```r
# create vector with all countries
geo_col <- datCO2[, unique(geo)]
# take 2 random samples
geo_c <- sample(geo_col, 2)
```

5. Filter data

```r
# filter data
datFilter <- datCO2[unit == "THS_T" &
                      airpol == "GHG" &
                      airemsect %in% c("CRF3", "CRF31", "CRF1A3") &
                      geo %in% geo_c]
```

6. Remove columns/variables unit and airpol

```
# remove variables
datFilter <- datFilter[, -c("unit", "airpol")]
```

7. Show records per country

```
#
datFilter[, .N, by = geo]
```

```
##    geo  N
## 1:  IS 84
## 2:  LU 84
```

Both randomly picked countries have 84 observations.

8. Rename variable `airemsect` and its categorical values

```
# rename values in variable "airemsect"
datFilter[airemsect == "CRF1A3", airemsect := "Transport"]
datFilter[airemsect == "CRF3", airemsect := "Agriculture"]
datFilter[airemsect == "CRF31", airemsect := "Livestock"]
# rename variable "airemsect" to "sector"
setnames(datFilter, "airemsect", "sector")
```

9. Calculate the average greenhouse gas (GHG) emission per sector

```
# aggregate by variable "sector" and calculate the mean
datFilter[,.(mean(values)), by = sector, ]
```

```
##         sector        V1
## 1:   Transport 2997.8302
## 2: Agriculture  619.3607
## 3:   Livestock  417.3638
```

The sector `Transport` produced by far the most greenhouse gas emissions.

10. Average GHG per sector and country

```
# aggregate by variable "sector" and "geo" and calculate the mean
datFilter[,.(mean(values)), by = .(sector, geo)]
```

```
##         sector geo        V1
## 1:   Transport  IS  782.3989
## 2:   Transport  LU 5213.2614
## 3: Agriculture  IS  555.7621
## 4: Agriculture  LU  682.9593
## 5:   Livestock  IS  364.4096
## 6:   Livestock  LU  470.3179
```

```r
# calculate the difference in greenhouse gas emissions in the sector "Livestock"
datFilter[sector == "Livestock" & geo == "LU", .(mean(values))] -
  datFilter[sector == "Livestock" & geo == "IS", .(mean(values))]
```

```
##          V1
## 1: 105.9082
```

The country `BE` has higher average greenhouse gas emissions in the sector ”Livestock”.

11. Sum of the “Livestock” sector per country for the period 2000-2017

```r
datFilter[sector == "Livestock" & time %between% c(2000, 2017), .(sum(values)), by = .(geo)]
```

```
##    geo      V1
## 1:  IS 6483.79
## 2:  LU 8388.62
```

The sector “Livestock” produced: * x in the country ‘BE’ * x in the country ‘BE’ for the period 2000-2017.

12. Plotting the GHG emissions of ”Transportation” sector for both countries

```r
ggplot(data=datFilter[sector == "Transport"], aes(x=time, y=values, color=geo)) + geom_line()
```