Prof. Dr. M. Kateri,
Dr. W. Herff,
L. Kaufmann, M.Sc.

RWTH Aachen, SS 2022
Release date: April 4th, 2022
Solution: Week starting from April 11th, 2022

Applied Data Analysis

R-Laboratory 1

Vector & Matrix Calculation – Data Types & Structures – Functions

Useful functions:

- `cut()`
- `read.table()`
- `prod()`
- `seq_along()`
- `which()`
- `sd()`
- `dnorm()`
- `pnorm()`
- `qnorm()`
- `rnorm()`

Task 1

(a) Let $a, b \in \mathbb{R}^3$, $A, B \in \mathbb{R}^{3\times 3}$ be vectors and matrices, respectively, with values

$$a = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix},\ b = \begin{pmatrix} 3 \\ -5 \\ 0 \end{pmatrix},\ A = \begin{pmatrix} 3 & 5 & -1 \\ 1.5 & -\pi & \mathrm{e}^{2.5} \\ 1/8 & -6 & 9 \end{pmatrix},\ B = \begin{pmatrix} 1 & -2 & 3 \\ -2 & 4 & -6 \\ 3 & -6 & 9 \end{pmatrix}.$$

(i) Create vectors `a, b` and matrices `A, B` in R, which have the same values as a, b, A, B and print them on screen.

(ii) Calculate $A \cdot a$, $B \cdot b$, A^2, $A \cdot B$, $a^\intercal \cdot b$ and print them on screen.

(iii) Suppose $\widehat{+}$, $\widehat{-}$, $\widehat{\cdot}$, $\widehat{:}$ denote the respective component-wise operations to $+, -, \cdot, :$. How can you realize $b \mathbin{\widehat{\diamond}} a$ and $B \mathbin{\widehat{\diamond}} A$ in R for $\widehat{\diamond} \in \{\widehat{+}, \widehat{-}, \widehat{\cdot}, \widehat{:}\}$?

(iv) Create an additional matrix $B' \in \mathbb{R}^{2\times 2}$, whose elements are obtained from B by summing up the first two rows and columns.

(b) Consider the vector

```
x <- c(5, 2, 6, 4, 1, 2, 2, 5, 4, 4, 6, 4, 2, 5, 5, 3, 6, 1, 4, 5)
```

of the integer values 1 up to 6. Create an additional vector `y` applying the following mapping on each component of `x`

$$f\colon \{1,\dots,6\} \longrightarrow \{1,2,3\}, \quad v \mapsto f(v) := \begin{cases} 1, & v \in \{1,2\}, \\ 2, & v \in \{3,4\}, \\ 3, & v \in \{5,6\}. \end{cases}$$

Hint: Use the R function `cut`.

Task 2

Consider the following vectors

```
v1 <- c(TRUE, TRUE, FALSE, TRUE, FALSE); v2 <- 1:6; v3 <- 5:10
```

(a) Read the data into R (by copy-pasting the above code) and print the data on screen.

(b) What happens, when we write `sum(v1)` and `prod(v1)` and why? Convert `v1` to a numeric vector and save it as `v4`.

(c) Denote by $(a_1, \ldots, a_6)'$ the vector representing `v2` and by $(b_1, \ldots, b_6)'$ the vector representing `v3`. Then, compute the value of $\sum_{i=1}^{6}(a_i \cdot b_i)^i$ on two different ways:
(i) using a loop and (ii) without using a loop

(d) Search the first index, where `v2` has a larger element than `v4`. Do this with and without using a loop.

(e) Write a function `example.function(vec1,vec2)`, where `vec1` and `vec2` are numeric vectors representing $a \in \mathbb{R}^{d_1}$ and $b \in \mathbb{R}^{d_2}$, which computes the value of the function $f\colon \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \longrightarrow \mathbb{R}$,

$$(a, b) \mapsto f(a, b) := \begin{cases} \sum_{i=1}^{d_1}(a_i \cdot b_i)^i, & d_1 = d_2, \\ \min\{i \in \{1, \ldots, \min\{d_1, d_2\}\} \,:\, a_i > b_i\}, & d_1 \neq d_2, \end{cases}$$

where $\min(\emptyset) := \infty$.

Task 3

(a) (i) Download the file *credits.wsv* from the RWTHmoodle space of the course Applied Data Analysis in the section "R-Lab Datasets" and import the data as a `data.frame` object into the R workspace.

(ii) Print the variable `amount` in the `data.frame` object in the console. Of which data type is `amount`? Print on screen the second column/row, every column/row apart from the first eight columns/rows and the first/last six rows of *credits.wsv*.

(iii) Calculate the arithmetic mean and median of the variable `amount`. Print the `summary` of `amount` to the console.

(b) (i) Write a function `my.sd(data, corrected)` returning the sample standard deviation (SD) of `data`, where

- `data` is a numeric vector,
- `corrected` is a logical value, indicating whether the corrected or uncorrected sample standard deviation shall be used. The default should be the corrected sample standard deviation. The corrected sample SD s_c and the uncorrected sample SD s_{uc} for a sample $x_1, \ldots, x_n$, $n \in \mathbb{N}$, are defined by

$$s_c = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \quad \text{and} \quad s_{uc} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2},$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ denotes the arithmetic mean.

(ii) Use the internal R function `sd` to calculate the sample SD for the variable `amount` in the cars–data set from part (a) as well as your own function (with `corrected=TRUE` and `corrected=FALSE`). Does `sd` compute the corrected or the uncorrected variant of SD by default?

Task 4

(a) Draw random samples of size $n = 10, 50, 100$ from a $\mathcal{N}(\mu, \sigma^2)$ distribution with $\mu = 5$ and $\sigma^2 = 4$.

(b) For each sample size n compute the proportion of values that lie in the intervals

(i) $I_1 = [3, 7]$

(ii) $I_2 = [5, 9]$.

Compare the proportions with the true probabilities that a normal distributed random variable with $\mu = 5$ and $\sigma^2 = 4$ lies in those two intervals.

(c) For each sample size n compute the α-quantiles of the generated values for

(i) $\alpha = 0.3$

(ii) $\alpha = 0.5$

(iii) $\alpha = 0.75$.

Compare these quantiles with the quantiles of a $\mathcal{N}(\mu, \sigma^2)$ distribution.

```
######################
########TASK1########
######################
# a) i define this
a = c(1,2,3)
b = c(3, -5, 0)
A = matrix(c(3, 1.5, 1/8, 5, -pi, -6, -1, exp(2.5), 9),
           nrow = 3,
           ncol = 3)

B = matrix(c(1, -2, 3, -2, 4, -6, 3, -6, 9),
           nrow = 3,
           ncol = 3)
#ii)
# A %*% a
# t(a) %*% b

#iii
#component-wise operation
# + ,-, *, /

#iv
# construct the new matrix by
# 1. add the first two columns
# 2. add the first two rows
# > B
# [,1] [,2] [,3]
# [1,]    1   -2    3
# [2,]   -2    4   -6
# [3,]    3   -6    9
# > rbind(B[1,] + B[2,], B[3,])
# [,1] [,2] [,3]
# [1,]   -1    2   -3
# [2,]    3   -6    9
# > cbind(rbind(B[1,] + B[2,], B[3,])[,1] + rbind(B[1,] + B[2,], B[3,])[,2] ,rbind(B[1,] + B[2,], B[3,])[,3])
# [,1] [,2]
# [1,]    1   -3
# [2,]   -3    9
#b) Use cut and label data
x = c(5,2,6,4,1,2,2,5,4,4,6,4,2,5,5,3,6,1,4,5)

# set x -> 1, if x = 1, 2,
# set x -> 2, if x = 3, 4
# set x -> 3, if x = 5, 6

cuts = cut(x/2, breaks = c(0,1,2,3), include.lowest = TRUE, label = c(1,2,3))
```

```
#########TASK2########
#####################
#a)
v1 = c(TRUE, TRUE, FALSE, TRUE, FALSE)
v2 = 1:6
v3 = 5:10

#b)

v4 = v1*1

#c)
k = 0
for(i in 1:length(v2)) {

  k = k +  (v2[i]*v3[i])^i

}
print(k)

# to get the power of values corresponding their index
# i.e. a^1 b^2 c^3 d^4 e^5
#
#sum((v2*v3)^c(1:length(v2)))

#d)

# by using loop
short_len =  if (length(v2) < length(v4)) length(v2) else length(v4)

for(i in 1:short_len) {

  if(v2[i]> v4[i]){

    print(v2[i])
    break

  }

}

# without loop
# which(v2[1:short_len] - v4 > 0)[1]

mywhich <- function(a1, a2){

  samelen = length(a1) == length(a2)

  if(samelen){

    return( sum((a1*a2)^c(1:length(a1))))

  } else {

    short_len =  if (length(v2) < length(v4)) length(v2) else length(v4)

    return(which(a1[1:short_len] - a2[1:short_len] > 0)[1])

  }

}
```

```
#i)
credits.df <- read.table("R-Lab-Datasets/credits.wsv", header = TRUE, sep = " ")
#ii)
#credits.df[2,]

# exclude from the 1st and 8th column
# credits.df[, -(1:8)]
#iii)
mean(credits.df$amount)
median(credits.df$amount)
summary(credits.df$amount)


my.sd <- function(data, corrected=FALSE){

  len = if(corrected) length(data) else length(data)-1

  return( sqrt(sum((data - mean(data))^2)/len))

}
```

```
######################
#########TASK4########
######################

#a)

mu = 5
sd = 2

rnorm10 = rnorm(10, mean = mu, sd = sd)

rnorm50 = rnorm(50, mean = mu, sd = sd)

rnorm100 = rnorm(100, mean = mu, sd = sd)

#b)
ss
# case rnorm10

pnorm(7, mean(rnorm10), sd(rnorm10)) - pnorm(3, mean(rnorm10), sd(rnorm10))

pnorm(9, mean(rnorm10), sd(rnorm10)) - pnorm(5, mean(rnorm10), sd(rnorm10))


pnorm(7, mean(rnorm50), sd(rnorm50)) - pnorm(3, mean(rnorm50), sd(rnorm50))

pnorm(9, mean(rnorm50), sd(rnorm50)) - pnorm(5, mean(rnorm50), sd(rnorm50))

pnorm(7, mean(rnorm100), sd(rnorm100)) - pnorm(3, mean(rnorm100), sd(rnorm100))

pnorm(9, mean(rnorm100), sd(rnorm100)) - pnorm(5, mean(rnorm100), sd(rnorm100))

#c) quantiles

qnorm(0.3, mean(rnorm10), sd(rnorm10))
qnorm(0.3, mean(rnorm50), sd(rnorm50))
qnorm(0.3, mean(rnorm100), sd(rnorm100))
qnorm(0.3, mu, sd)

qnorm(0.5, mean(rnorm10), sd(rnorm10))
qnorm(0.5, mean(rnorm50), sd(rnorm50))
qnorm(0.5, mean(rnorm100), sd(rnorm100))
qnorm(0.5, mu, sd)

qnorm(0.75, mean(rnorm10), sd(rnorm10))
qnorm(0.75, mean(rnorm50), sd(rnorm50))
qnorm(0.75, mean(rnorm100), sd(rnorm100))
qnorm(0.75, mu, sd)
```