Multivariate Statistical Analysis - Exercise Session 1

13.01.2022

Problem 1: Introduction to R and RStudio

a) Setting working directory and getting help

There are at least three ways to set the working directory:

1. from the user interface of RStudio: `Session` → `Set Working Directory` → `Choose directory`,
2. by pressing `Ctrl + Shift + H` or
3. Use `setwd` command (example below).

```
setwd("~/teaching/multivariate21/01ex")
```

Getting help in R:

```
?matrix
help(matrix)
```

b) Calculate affine transformation $\mathbf{y} = \mathbf{x}\mathbf{A}^{-1} + \mathbf{b}$

First let us create all the matrices and vectors. Notice that $\mathbf{x}$ and $\mathbf{b}$ are row vectors. We create the vector $\mathbf{x}$ in two ways. Firstly by using `c` command and secondly using `matrix` command.

```
a <- matrix(c(2, 1, 5, -2, 7, 0, 5, -8, -1), ncol = 3, byrow = TRUE)
x1 <- c(8, -4, 2) # vector
x2 <- matrix(c(8, -4, 2), nrow = 1, byrow = FALSE) # matrix
b <- c(3, 10, -19)

a
```

```
##      [,1] [,2] [,3]
## [1,]    2    1    5
## [2,]   -2    7    0
## [3,]    5   -8   -1
```

```
x1
```

```
## [1]  8 -4  2
```

```
x2
```

```
##      [,1] [,2] [,3]
## [1,]    8   -4    2
```

```
b
```

```
## [1]   3  10 -19
```

Next let us calculate $\mathbf{y}$. With `solve` command we can calculate inverse matrix and operator `%*%` performs the matrix multiplication.

```r
y1 <- x1 %*% solve(a) + b
y2 <- x2 %*% solve(a) + b
y1
```

```
##          [,1]     [,2]      [,3]
## [1,] 3.774775 11.45946 -17.12613
```

```r
all(y1 == y2)
```

```
## [1] TRUE
```

So we can perform calculations by using either `x1` or `x2`. Let us play a little bit more with vectors and matrices.

```r
a %*% x1 # no error
```

```
##      [,1]
## [1,]   22
## [2,]  -44
## [3,]   70
```

```r
x1 %*% a # no error
```

```
##      [,1] [,2] [,3]
## [1,]   34  -36   38
```

```r
x2 %*% a # no error
```

```
##      [,1] [,2] [,3]
## [1,]   34  -36   38
```

```r
# A %*% x2 # error
```

One can find more information about the behavior of matrix multiplication operator in help pages.

```r
?"%*%"
```

Especially, in the help pages the following is said:

> "If one argument is a vector, it will be promoted to either a row or column matrix to make the two arguments conformable."

c) Installing packages and sampling from multivariate normal distribution

Create variables for expected value μ, covariance matrix Σ and sample size n.

```r
mu <- c(3, 1)
sigma <- matrix(c(4, 1, 1, 2), byrow = TRUE, ncol = 2)
n <- 100
```

Packages are installed with the command `install.packages` where package name is given as a string.

```r
install.packages("mvtnorm")
```

One can use functions from the package straight away by specifying the *namespace*.

```r
mvtnorm::rmvnorm(n, mu, sigma)
```

Alternatively, one can import the package and then there is no need to specify the namespace.

```r
library(mvtnorm)
set.seed(123)
```

```r
x <- rmvnorm(n, mu, sigma) # Matrix with 2 columns and 100 rows.
dim(x)
```

```
## [1] 100   2
```

```r
head(x)
```

```
##          [,1]       [,2]
## [1,] 1.823028  0.5149745
## [2,] 6.103696  1.5613434
## [3,] 3.766090  3.4096342
## [4,] 3.535096 -0.6118415
## [5,] 1.508960  0.1794479
## [6,] 5.527988  1.8617391
```

Figure 1 should look similar if you set seed to 123 before creating the sample.

```r
plot(x, pch = 20, xlab = expression("X"[1]), ylab = expression("X"[2]),
     main = expression(paste("Sample from ", "N(", mu, ",", Sigma, ")")))
```

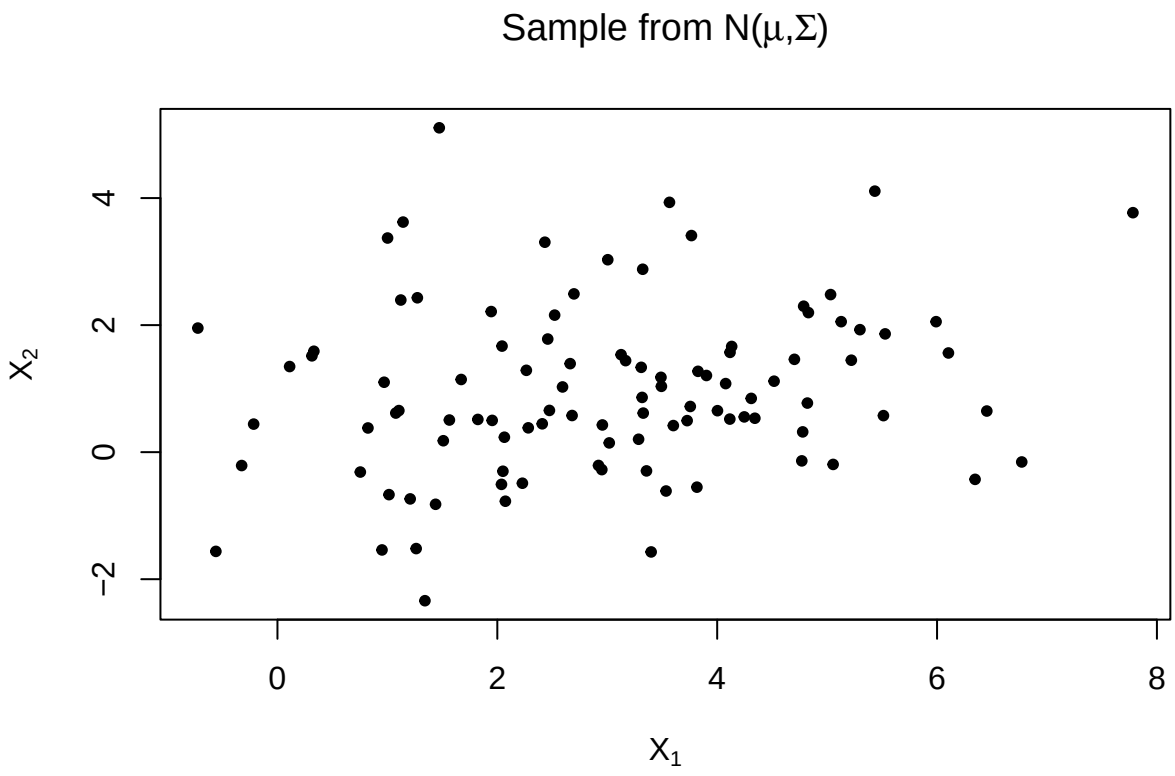

Figure 1: Scatter plot of sample from a bivariate normal distribution.

d) Sample mean, sample covariance and eigenvalues

There are a couple of ways to calculate the sample mean vector.

```
mx <- apply(x, 2, mean) # apply function "mean" to every column
mx2 <- colMeans(x)
all(mx == mx2)
```

```
## [1] TRUE
```

```
mx
```

```
## [1] 3.004655 0.970002
```

One can use function `cov` in order to calculate the sample covariance.

```
sx <- cov(x)
sx
```

```
##           [,1]      [,2]
## [1,] 3.0744739 0.4545397
## [2,] 0.4545397 1.8184535
```

Next let us calculate eigenvalues and eigenvectors of sample covariance matrix $\mathbf{S}_x$. Columns of variable `eigvec` contain the eigenvectors.

```
eig <- eigen(sx)
eigval <- eig$values
eigvec <- eig$vectors
eigval
```

```
## [1] 3.221708 1.671220
```

```
eigvec
```

```
##            [,1]       [,2]
## [1,] -0.9513361  0.3081552
## [2,] -0.3081552 -0.9513361
```

Lastly, let us verify that

$$\text{Tr}(\mathbf{S}_x) = \lambda_1 + \lambda_2$$

and

$$\text{Det}(\mathbf{S}_x) = \lambda_1 \lambda_2.$$

```
sum(diag(sx)) - sum(eigval)
```

```
## [1] 0
```

```
det(sx) - prod(eigval)
```

```
## [1] -1.776357e-15
```

e) Affine transformation $\mathbf{Ax_i + b}$ $\forall$ $i \in \{1, 2, \dots, 100\}$

We can avoid having any for loops by doing matrix operations and using function `sweep`. Function `sweep` is quite similar to `apply`. For example, below are two ways to sum vector $\mathbf{b}$ to each row of $\mathbf{A}$.

```
b <- c(3, 1)
a <- matrix(c(1, 2, 3, 1), byrow = T, ncol = 2)

test1 <- sweep(a, 2, b, "+")
test2 <- t(apply(a, 1, "+", b))
all(test1 == test2)
```

```
## [1] TRUE
```

```
test1
```

```
##      [,1] [,2]
## [1,]    4    3
## [2,]    6    2
```

Now that we have introduced function sweep, let us perform the affine transformations. Notice that

$$\mathbf{X}\mathbf{A}^T + \mathbf{1}_{100}\mathbf{b}^T = \begin{pmatrix} (\mathbf{A}\mathbf{x}_1 + \mathbf{b})^T \\ (\mathbf{A}\mathbf{x}_2 + \mathbf{b})^T \\ \vdots \\ (\mathbf{A}\mathbf{x}_{100} + \mathbf{b})^T \end{pmatrix}, \tag{1}$$

where $\mathbf{1}_{100} \in \mathbb{R}^{100}$ is a column vector of ones.

```
# First way
y <- sweep(x %*% t(a), 2, b, "+")

# Second way
ones <- rep(1, n)
y2 <- x %*% t(a) + ones %*% t(b)

all(y == y2)
```

```
## [1] TRUE
```

We can use any matrix norm $\|\cdot\|$ to check that two matrices $X, Y \in \mathbb{R}^{n\times m}$ are equal. This works since

$$X = Y \iff \|X - Y\| = 0.$$

We can choose to use, e.g. Frobenius norm.

```
norm(colMeans(y) - (a %*% colMeans(x) + b), type = "F")
```

```
## [1] 8.881784e-16
```

```
norm(cov(y) - (a %*% cov(x) %*% t(a)), type = "F")
```

```
## [1] 3.552714e-15
```

f) Loading data and functions

Be sure that you know the path to the data with respect to the working directory. For example, it is convenient to simply save the data in the working directory.

```
getwd()
```

```
## [1] "/home/perej/teaching/multivariate21/01ex"
```

```
data <- read.table("Data1.txt", sep = "\t", header = FALSE)
class(data)
```

```
## [1] "data.frame"
```

```
data <- as.matrix(data)
```

Here we create the function that plots the pairwise scatter plots and centers the data. Additionally, Figure 2 shows the resulting plot.

```
center <- function(x) {
  pairs(x, pch = 19, col = "midnightblue", gap = 0, upper.panel = NULL,
        cex.labels = 1)
  ave <- colMeans(x)
  cent <- sweep(x, 2, ave, "-")
  return(cent)
}

data_center <- center(data)
```

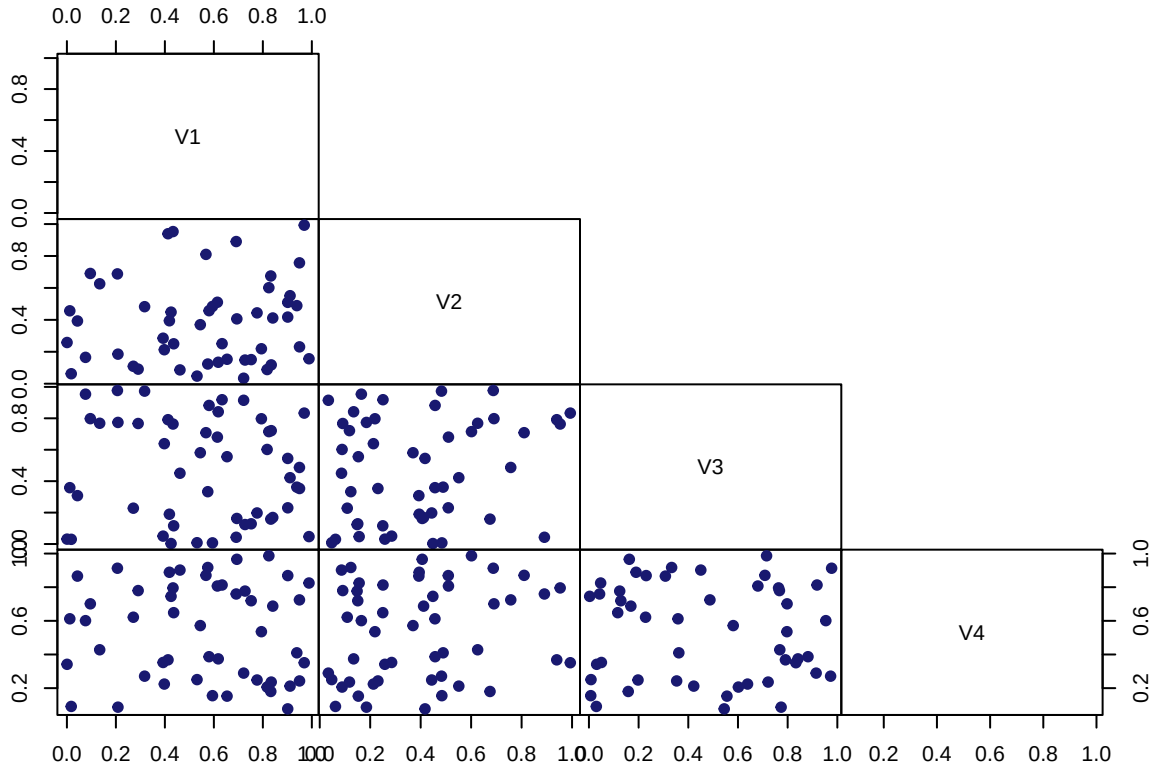

Figure 2: Scatter plot of variables.

Lastly, we calculate sample covariance, sample correlation and the corresponding eigenvalues and eigenvectors.

```
cov_center <- cov(data_center)
cor_center <- cor(data_center)
eigen(cov_center)$values
```

```
## [1] 0.11163469 0.08903608 0.08697829 0.05218629
```

```
eigen(cov_center)$vectors
```

```
##             [,1]        [,2]        [,3]       [,4]
## [1,]  0.24232756  0.51237112 -0.76290091  0.3110231
## [2,] -0.30797566  0.55314104 -0.04145492 -0.7729602
## [3,] -0.91739205 -0.09949153 -0.23313028  0.3068282
```

```
## [4,] -0.06942745  0.64931676  0.60159285  0.4600583
```

```
eigen(cor_center)$values
```

```
## [1] 1.2367328 1.0762947 1.0211537 0.6658188
```

```
eigen(cor_center)$vectors
```

```
##             [,1]       [,2]       [,3]       [,4]
## [1,] -0.02338207  0.7717167 -0.5016233  0.3902315
## [2,]  0.72821948  0.2244426 -0.1739016 -0.6237629
## [3,]  0.47737240 -0.5465891 -0.4787554  0.4941145
## [4,]  0.49118760  0.2352002  0.6992321  0.4631307
```

Problem 2: Eigenvalues of a symmetric matrix

Definition 1 (Eigenvalues and eigenvectors)**.** A scalar λ_i is called an eigenvalue of $p \times p$ matrix A if there is a nontrivial solution v_i to

$$Av_i = \lambda_i v_i,$$

where v_i is called an eigenvector corresponding to the eigenvalue λ_i.

Proposition 1. *Eigenvalues of real symmetric matrix are always real valued.*

Proof of Proposition 1. Let $A \in \mathbb{R}^{p\times p}$ be a real symmetric matrix. Let λ_i be an eigenvalue of A and v_i the corresponding eigenvector. We need to prove that $\lambda_i = \bar{\lambda}_i$. From the properties of matrix A and from the definition of eigenvalues and eigenvectors we get

$$\begin{aligned}
&Av_i = \lambda_i v_i \\
&\Rightarrow \overline{Av_i} = \overline{\lambda_i v_i} \\
&\Rightarrow \bar{A}\bar{v_i} = \bar{\lambda}_i \bar{v_i} \\
&\overset{A \text{ real}}{\Rightarrow} A\bar{v_i} = \bar{\lambda}_i \bar{v_i} \\
&\Rightarrow v_i^T A\bar{v_i} = v_i^T \bar{\lambda}_i \bar{v_i} \\
&\overset{A=A^T}{\Rightarrow} v_i^T A^T \bar{v_i} = v_i^T \bar{\lambda}_i \bar{v_i} \\
&\Rightarrow (Av_i)^T \bar{v_i} = \bar{\lambda}_i v_i^T \bar{v_i} \\
&\Rightarrow (\lambda_i v_i)^T \bar{v_i} = \bar{\lambda}_i v_i^T \bar{v_i} \\
&\Rightarrow \lambda_i v_i^T \bar{v_i} - \bar{\lambda}_i v_i^T \bar{v_i} = 0 \\
&\Rightarrow (\lambda_i - \bar{\lambda}_i) v_i^T \bar{v_i} = 0
\end{aligned}$$

Thus we must have that $\lambda_i - \bar{\lambda}_i = 0$ or $v_i^T \bar{v_i} = 0$. However, note that $v_i^T \bar{v_i} = \langle v_i, v_i \rangle$ is the canonical hermitian inner product. By properties of inner product we have

- $\langle v_i, v_i \rangle \geq 0$ and
- $\langle v_i, v_i \rangle = 0$ if and only if $v_i = 0$.

Remember that by definition of eigenvectors we assume that $v_i \neq 0$. Thus the option $v_i^T \bar{v_i} = 0$ is not possible. That is, we must have $\lambda_i = \bar{\lambda}_i$. □

Remark. Real *nonsymmetric* matrices can have complex eigenvalues. For example, consider matrix

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

```r
a <- matrix(c(1, -1, 1, 1), ncol = 2, byrow = TRUE)
eigen(a)$values
```

```
## [1] 1+1i 1-1i
```

Thus the assumption of symmetricity cannot be dropped from Proposition 1.

Hints for homework 1

Alone from wikipedia one can find useful hints. For example, check the following wikipedia pages.

- Sample mean and covariance
- Square root of a matrix
- Correlation
- Covariance matrix