Seminar 1

Lesnykh Kirill, BSC-153

18 01 2019

```
knitr::opts_chunk$set(echo = TRUE)
```

```
Sys.setlocale(locale = "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/C"
```

Seminar Assignments

```
t <- c("F","D","F","I","A","F","F","D","B","SLO","I","F","GB","B")
10
```

```
## [1] 10
```

```
T <- factor(t)
t
```

```
##  [1] "F"   "D"   "F"   "I"   "A"   "F"   "F"   "D"   "B"   "SLO" "I"
## [12] "F"   "GB"  "B"
```

```
table(t)/length(t)
```

```
## t
##         A         B         D         F        GB         I
## 0.07142857 0.14285714 0.14285714 0.35714286 0.07142857 0.14285714
##       SLO
## 0.07142857
```

```
cumsum(table(t))/length(t)
```

```
##         A         B         D         F        GB         I
## 0.07142857 0.21428571 0.35714286 0.71428571 0.78571429 0.92857143
##       SLO
## 1.00000000
```

```
L <- c("unsatisfactory","poor","average","good","excellent")

L <- c("Не удовлетворен", "Плохо", "Средне", "Хорошо", "Отлично")
L
```

```
## [1] "Не удовлетворен" "Плохо"           "Средне"          "Хорошо"
## [5] "Отлично"
```

```
attributes <- c("Income", "Age", "Number of Kids", "Education")
Alex <- c(2000, 50, 2, 12)
Vlado <- c(1000, 25, 1, 16)
name <- c(Alex, Vlado)
df <- data.frame(Alex, Vlado, row.names = attributes)
df <- t(df)
df
```

```
##       Income Age Number of Kids Education
## Alex    2000  50             2        12
## Vlado   1000  25             1        16
```

```
x <- c(3,2,5)
x
```

```
## [1] 3 2 5
```

The command creates the vector "x"", that contains following numeric elements: 3, 2, 5.

```
A1 <- matrix(c(1,3,5,2,4,6),2,3)
A1
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    4
## [2,]    3    2    6
```

The command creates the matrix "A1" with 2 rows and 3 columns, that contains the numeric elements of the vector c(1, 3, 5, 2, 4, 6). The elements in the matrix are placed firstly for each row, than for each column.

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

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

The command creates the matrix "A" with 2 rows and 3 columns, that contains the numeric elements of the vector c(1, 4, 2, 5, 3, 6). The elements in the matrix are placed firstly for each row, than for each column. So the numbers in matrix are placed raising in amount firstly by column, and than by row.

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

```
##      [,1] [,2] [,3]
## [1,]    4    5    6
## [2,]    1    2    3
```

The command creates the matrix "B" with 2 rows and 3 columns, that contains the numeric elements of the vector c(4, 1, 5, 2, 6, 3). The elements in the matrix are placed firstly for each row, than for each

column.

```
X1<-A%*%t(B)
X1
```

```
##      [,1] [,2]
## [1,]   32   14
## [2,]   77   32
```

The command generates the correct multiplication of the matrixes “A” and “B” that equals to the object “X1”. The command “t()” transponts the matrix “B”. The element “%*%” makes the correct multiplication of the matrixes row by columns.

```
X2<-B%*%t(A)
X2
```

```
##      [,1] [,2]
## [1,]   32   77
## [2,]   14   32
```

The command generates the correct multiplication of the matrixes “A” and “B” that equals to the object “X2”. The command “t()” transponts the matrix “B”. The element “%*%” makes the correct multiplication of the matrixes row by columns.

```
X3<-t(A)%*%B
X3
```

```
##      [,1] [,2] [,3]
## [1,]    8   13   18
## [2,]   13   20   27
## [3,]   18   27   36
```

The command generates the correct multiplication of the matrixes “A” and “B” that equals to the object “X3”. The command “t()” transponts the matrix “A”. The element “%*%” makes the correct multiplication of the matrixes row by columns. Due to the another order of the matrixes in the equation, transpoted matrix “A” stands before matrix “B”, the results would be different from the object “X1”.

```
X4<-t(B)%*%A
X4
```

```
##      [,1] [,2] [,3]
## [1,]    8   13   18
## [2,]   13   20   27
## [3,]   18   27   36
```

The command generates the correct multiplication of the matrixes “A” and “B” that equals to the object “X4”. The command “t()” transponts the matrix “B”. The element “%*%” makes the correct multiplication of the matrixes row by columns. Due to the another order of the matrixes in the equation, transpoted matrix “B” stands before matrix “A”, the results would be different from the object “X2”.

```
d<-diag(A%*%t(B))
d
```

```
## [1] 32 32
```

The command generates the object “b” which contains the results of the summary of the A and transpotted B and after that (due to the dial() function) deletes all other numbers exceot for the ones from the diagonal. So, in the output we have only the numbers from the diagonal of the generated matrix.

```
s<-sum(diag(A%*%t(B)))
s
```

```
## [1] 64
```

The command generates the object “s” which contains the sum of the results of the previouos line: it generates the diagonal numbers of the sum of the matrixes A and transpoted B. So, the generated number in the output is the sum of the diagonal numbers of the sum of the matrixes A and transpoted B.

```
Y<-solve(A%*%t(B))
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

```
t(B)
```

```
##      [,1] [,2]
## [1,]    4    1
## [2,]    5    2
## [3,]    6    3
```

```
Y
```

```
##           [,1]       [,2]
## [1,] -0.5925926  0.2592593
## [2,]  1.4259259 -0.5925926
```

The command generates the object “Y” which generates the matrix. If the matrix “X1”, the result of the correct multipication of the A and transpoted B, would be multiplied by the matrix “Y”, the result would be the identity matrix. That is what the function “solve” does: it gives the matrix, multiplication by which of the given matrix will equal unity matrix.