CDA

PRACTICAL 1

Luis Alberto Alvarez Zavaleta

1. **Generate the numbers 1, 2,. . ., 12, and store the result in the vector x.**

```
> x<-1:12
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

2. **Generate four repetitions of the sequence**

```
> repeated_sequence <- rep( c(6, 2, 4), times = 4)
> repeated_sequence
 [1] 6 2 4 6 2 4 6 2 4 6 2 4
```

3. **Generate the sequence consisting of six 9s, then five 2s, and finally four 5s. Store the numbers in a 5 by 3 matrix (populating it columnwise).**

```
matrix_sequence <- matrix(c(rep(9, 6), rep(2, 5), rep(5, 4)), nrow = 5, ncol = 3)
matrix_sequence
     [,1] [,2] [,3]
[1,]    9    9    2
[2,]    9    2    5
[3,]    9    2    5
[4,]    9    2    5
[5,]    9    2    5
```

4. **Generate a vector consisting of 20 numbers generated randomly from a normal distribution. Use the value 100 as seed**

```
set.seed(100)
random_vector <- rnorm(20)
random_vector
 [1] -0.50219235  0.13153117 -0.07891709  0.88678481  0.11697127  0.31863009
 [7] -0.58179068  0.71453271 -0.82525943 -0.35986213  0.08988614  0.09627446
[13] -0.20163395  0.73984050  0.12337950 -0.02931671 -0.38885425  0.51085626
[19] -0.91381419  2.31029682
```

Then, calculate the following statistics about the generated vector: mean, median, variance and the standard deviation.

```
> mean(random_vector)
[1] 0.1078671
> median(random_vector)
[1] 0.0930803
> var(random_vector)
[1] 0.516335
```

Repeat the generation of the vector and the statistics with and without changing the seed and observe what happens.

When you don´t use a seed, the vector created whit rnorm is different in each calling of the function and you have arrays with different numbers, leading to mean, median and variance changes for different arrays

5. From the resources folder at poliformat, download the file “data1.txt” that contains information about students.

(a) Read the data into an R object named students (data is in a space-delimited text file and there is no header row)

```
setwd("C:/Users/alber/OneDrive/Documentos/UPV/CDA")
students <- read.table("data1.txt")
students
   V1 V2     V3      V4
1  181 44   male  kuopio
2  160 38 female  Kuopio
. . .
```

(b) Add the following titles for columns: height, shoesize, gender, population

```
names (students) <- c( "height", "shoesize", "gender", "population")
```

(c) Check that R reads the file correctly

```
students
  height shoesize gender population
1    181       44   male     kuopio
2    160       38 female     kuopio
. . .
```

(d) Print the header names only

```
colnames(students)
[1] "height"     "shoesize"   "gender"     "population"
```

(e) Print the column height.

```
ncol(students)
[1] 5
```

(f) What is the gender distribution (how many observations are in each group) and the distribution of sampling sites (column population) ?

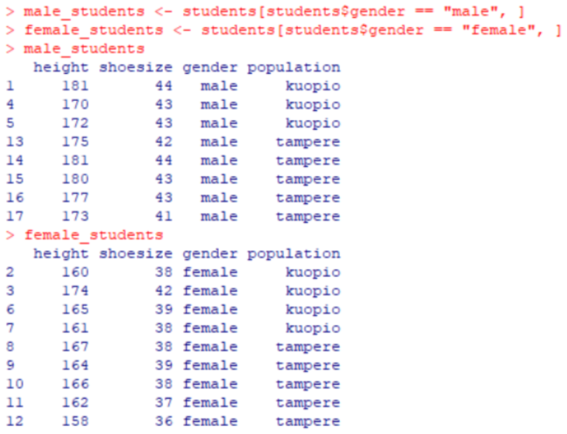
```
table(students$gender)
female   male
    9      8
table(students$population)
 kuopio tampere
     7      10
```

(g)Show the distributions in the above item at the same time by using a contingency table

```
table(students$gender, students$population)
        kuopio tampere
 female      4       5
 male        3       5
```

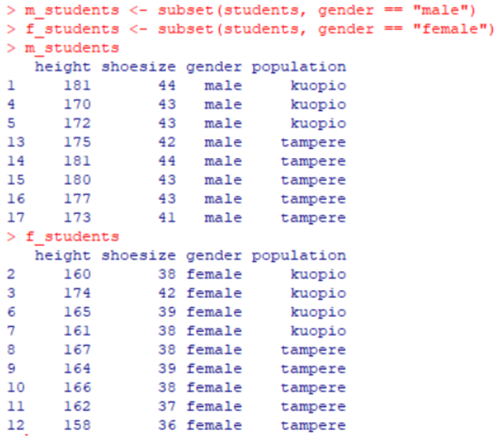
(h)Make two subsets of your dataset by splitting it according to gender. Use data frame operations first and then do the same usingthe function subset.

Using splitting function



```
> male_students <- students[students$gender == "male", ]
> female_students <- students[students$gender == "female", ]
> male_students
   height shoesize gender population
1     181       44   male     kuopio
4     170       43   male     kuopio
5     172       43   male     kuopio
13    175       42   male    tampere
14    181       44   male    tampere
15    180       43   male    tampere
16    177       43   male    tampere
17    173       41   male    tampere
> female_students
   height shoesize gender population
2     160       38 female     kuopio
3     174       42 female     kuopio
6     165       39 female     kuopio
7     161       38 female     kuopio
8     167       38 female    tampere
9     164       39 female    tampere
10    166       38 female    tampere
11    162       37 female    tampere
12    158       36 female    tampere
```

Using the subset function



```
> m_students <- subset(students, gender == "male")
> f_students <- subset(students, gender == "female")
> m_students
   height shoesize gender population
1     181       44   male     kuopio
4     170       43   male     kuopio
5     172       43   male     kuopio
13    175       42   male    tampere
14    181       44   male    tampere
15    180       43   male    tampere
16    177       43   male    tampere
17    173       41   male    tampere
> f_students
   height shoesize gender population
2     160       38 female     kuopio
3     174       42 female     kuopio
6     165       39 female     kuopio
7     161       38 female     kuopio
8     167       38 female    tampere
9     164       39 female    tampere
10    166       38 female    tampere
11    162       37 female    tampere
12    158       36 female    tampere
```

(i)Make two subsets containing individuals below and above the median height. Use data frame operations first and then do the same using the function subset.

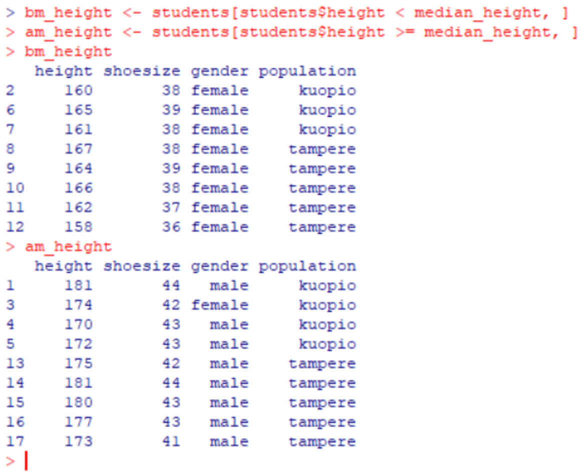
First we get the median of the column height

median_height <- median(students$height)

median_height

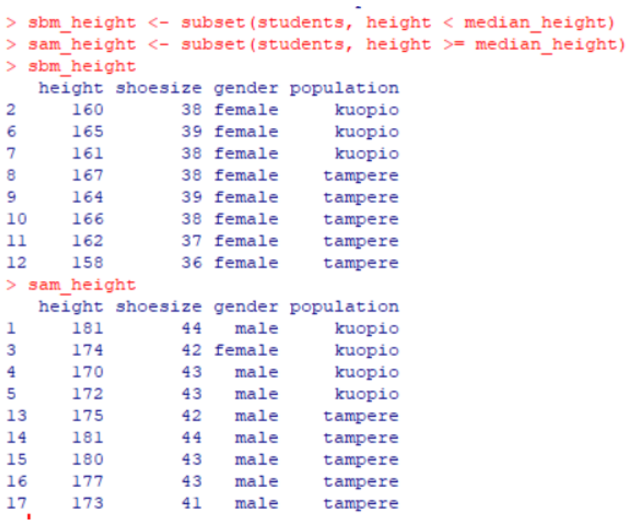
[1] 1.7

Using splitting function



```
> bm_height <- students[students$height < median_height, ]
> am_height <- students[students$height >= median_height, ]
> bm_height
   height shoesize gender population
2     160       38 female     kuopio
6     165       39 female     kuopio
7     161       38 female     kuopio
8     167       38 female    tampere
9     164       39 female    tampere
10    166       38 female    tampere
11    162       37 female    tampere
12    158       36 female    tampere
> am_height
   height shoesize gender population
1     181       44   male     kuopio
3     174       42 female     kuopio
4     170       43   male     kuopio
5     172       43   male     kuopio
13    175       42   male    tampere
14    181       44   male    tampere
15    180       43   male    tampere
16    177       43   male    tampere
17    173       41   male    tampere
> 
```

Using subset function



```
> sbm_height <- subset(students, height < median_height)
> sam_height <- subset(students, height >= median_height)
> sbm_height
   height shoesize gender population
2     160       38 female     kuopio
6     165       39 female     kuopio
7     161       38 female     kuopio
8     167       38 female    tampere
9     164       39 female    tampere
10    166       38 female    tampere
11    162       37 female    tampere
12    158       36 female    tampere
> sam_height
   height shoesize gender population
1     181       44   male     kuopio
3     174       42 female     kuopio
4     170       43   male     kuopio
5     172       43   male     kuopio
13    175       42   male    tampere
14    181       44   male    tampere
15    180       43   male    tampere
16    177       43   male    tampere
17    173       41   male    tampere
```

(j)Change height from centimetres to metres for all rows in the data frame. Do this using in three different ways: with basic primitives, a loop using for and the function apply.

Using Basic Primitives:

```
students$height_cmbp<- students$height / 100
```

Using for Loop

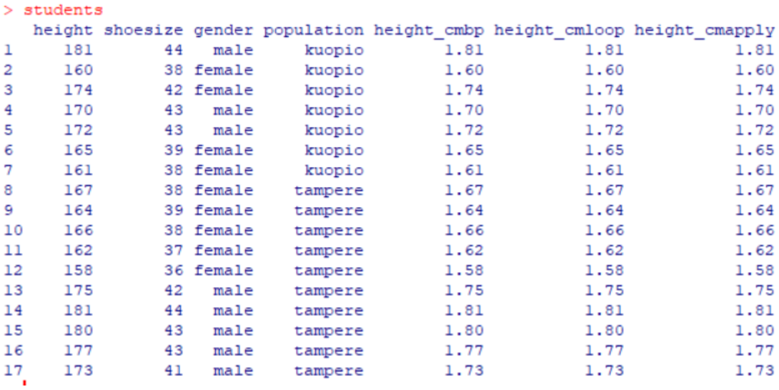
```
students$height_cmloop <-0

for (i in 1:nrow(students)) {
  students$height_cmloop [i] <- students$height [i] / 100
}
```

Using Apply function

```
students$height_cmapply<- apply (students [ "height", drop = FALSE], 1, function(x) x / 100)
```

Result dataframe for comparison

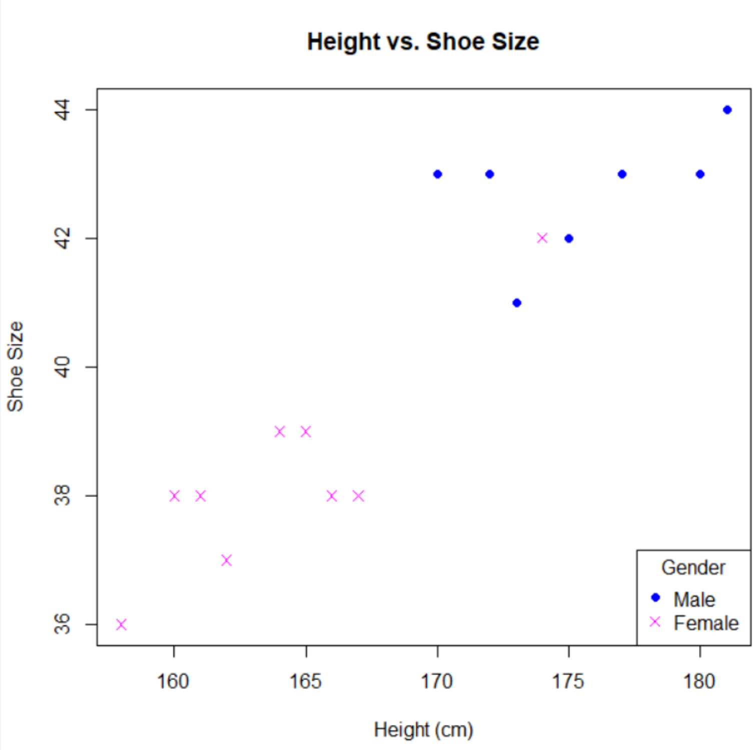


```
> students
   height shoesize gender population height_cmbp height_cmloop height_cmapply
1     181       44   male     kuopio        1.81          1.81           1.81
2     160       38 female     kuopio        1.60          1.60           1.60
3     174       42 female     kuopio        1.74          1.74           1.74
4     170       43   male     kuopio        1.70          1.70           1.70
5     172       43   male     kuopio        1.72          1.72           1.72
6     165       39 female     kuopio        1.65          1.65           1.65
7     161       38 female     kuopio        1.61          1.61           1.61
8     167       38 female    tampere        1.67          1.67           1.67
9     164       39 female    tampere        1.64          1.64           1.64
10    166       38 female    tampere        1.66          1.66           1.66
11    162       37 female    tampere        1.62          1.62           1.62
12    158       36 female    tampere        1.58          1.58           1.58
13    175       42   male    tampere        1.75          1.75           1.75
14    181       44   male    tampere        1.81          1.81           1.81
15    180       43   male    tampere        1.80          1.80           1.80
16    177       43   male    tampere        1.77          1.77           1.77
17    173       41   male    tampere        1.73          1.73           1.73
```

(k) Plot height against shoesize, using blue circles for males and magenta crosses for females. Add a legend.

```
plot(
  students$height,
  students$shoesize,
  xlab = "Height (cm)",
  ylab = "Shoe Size",
  col = ifelse(students$gender == "male", "blue", "magenta"),  # Set color based on gender
  pch = ifelse(students$gender == "male", 19, 4),  # Set circle and crosses based on gender
  main = "Height vs. Shoe Size")
legend(
  "bottomright",
  legend = c("Male", "Female"),
  col = c("blue", "magenta"),
  pch = c(19, 4),
  title = "Gender"
)
```

Result of the plot


Height vs. Shoe Size
Shoe Size
Height (cm)
Gender
● Male
× Female