# Practical 1

Introduction to R

Joan Navarro Bellido

## EXERCICE 1

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

x <- 1:12
print(x)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

## EXERCICE 2

```
# Generate four repetitions of the sequence of numbers (6, 2, 4).

y <- rep(c(6, 2, 4), 4)
print(y)
```

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

## EXERCICE 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).

sequence <- c(rep(9, 6), rep(2, 5), rep(5, 4))
matrix_sequence <- matrix(sequence, nrow = 5, ncol = 3, byrow = FALSE)
print(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
```

## EXERCICE 4

```
# Generate a vector consisting of 20 numbers generated randomly from
# a normal distribution. Use the value 100 as seed (in order to be able
```

```r
# to replicate the experiments).

set.seed(100)
random_vector <- rnorm(20)
print(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
```

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

mean_value <- mean(random_vector)
median_value <- median(random_vector)
variance_value <- var(random_vector)
standard_deviation_value <- sd(random_vector)

print(paste("Mean: ", mean_value))
```

```
## [1] "Mean:  0.107867147646684"
```

```r
print(paste("Median: ", median_value))
```

```
## [1] "Median:  0.0930803020313303"
```

```r
print(paste("Variance: ", variance_value))
```

```
## [1] "Variance:  0.516335002736962"
```

```r
print(paste("Standard deviation: ", standard_deviation_value))
```

```
## [1] "Standard deviation:  0.718564543194946"
```

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

set.seed(100)
random_vector <- rnorm(20)
print(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
```

```r
mean_value <- mean(random_vector)
median_value <- median(random_vector)
variance_value <- var(random_vector)
standard_deviation_value <- sd(random_vector)

print(paste("Mean: ", mean_value))
```

```
## [1] "Mean:  0.107867147646684"
```

```r
print(paste("Median: ", median_value))
```

```
## [1] "Median:  0.0930803020313303"
```

```r
print(paste("Variance: ", variance_value))
```

```
## [1] "Variance:  0.516335002736962"
```

```r
print(paste("Standard deviation: ", standard_deviation_value))
```

```
## [1] "Standard deviation:  0.718564543194946"
```

```r
# Random vector with seed 200
set.seed(200)
random_vector <- rnorm(20)
print(random_vector)
```

```
##  [1]  0.08475635  0.22646034  0.43255650  0.55806524  0.05975527 -0.11464087
##  [7] -1.02057835 -0.29705130  0.16815003  1.41987233 -0.09952507 -0.81829697
## [13] -0.46930224  0.57504497 -1.87174513 -0.63183110 -0.04243820  1.44210693
## [19] -0.92089342 -0.01560860
```

```r
mean_value <- mean(random_vector)
median_value <- median(random_vector)
variance_value <- var(random_vector)
standard_deviation_value <- sd(random_vector)

print(paste("Mean: ", mean_value))
```

```
## [1] "Mean:  -0.0667571647297675"
```

```r
print(paste("Median: ", median_value))
```

```
## [1] "Median:  -0.0290233979238126"
```

```r
print(paste("Variance: ", variance_value))
```

```
## [1] "Variance:  0.616924355239516"
```

```r
print(paste("Standard deviation: ", standard_deviation_value))
```

```
## [1] "Standard deviation:  0.785445322883468"
```

```r
# Random vector with seed 300
set.seed(300)
random_vector <- rnorm(20)
print(random_vector)
```

```
##  [1]  1.37379088  0.86210687  0.47348910  0.70126281 -0.08505527  1.56870212
##  [7]  0.81739197  0.39476860  1.21269855  0.35508066  2.21627421 -0.09054039
## [13] -1.31652811  0.06653479  0.51217262  0.05002974  1.46739718 -1.28952052
## [19] -0.25769151  0.29305131
```

```r
mean_value <- mean(random_vector)
median_value <- median(random_vector)
variance_value <- var(random_vector)
standard_deviation_value <- sd(random_vector)

print(paste("Mean: ", mean_value))
```

```
## [1] "Mean:  0.466270781584136"
```

```r
print(paste("Median: ", median_value))
```

```
## [1] "Median:  0.434128853552544"
```

```r
print(paste("Variance: ", variance_value))
```

```
## [1] "Variance:  0.77779170713689"
```

```r
print(paste("Standard deviation: ", standard_deviation_value))
```

```
## [1] "Standard deviation:  0.881925000857153"
```

# EXERCICE 5

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

students <- read.table("./dataset/data1.txt", header = FALSE)

# Add the following titles for columns:
# height, shoesize, gender, population

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

# Check that R reads the file correctly.

print(students)
```

```
##    height shoesize gender population
## 1     181       44   male     kuopio
## 2     160       38 female     kuopio
## 3     174       42 female     kuopio
## 4     170       43   male     kuopio
## 5     172       43   male     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
## 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
```

```r
# Print the header names only.

print(colnames(students))
```

```
## [1] "height"     "shoesize"   "gender"     "population"
```

```r
# Print the column height.

print(students$height)
```

```
##  [1] 181 160 174 170 172 165 161 167 164 166 162 158 175 181 180 177 173
```

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

gender_distribution <- table(students$gender)
population_distribution <- table(students$population)
print(gender_distribution)
```

```
##
## female   male
##      9      8
```

```r
print(population_distribution)
```

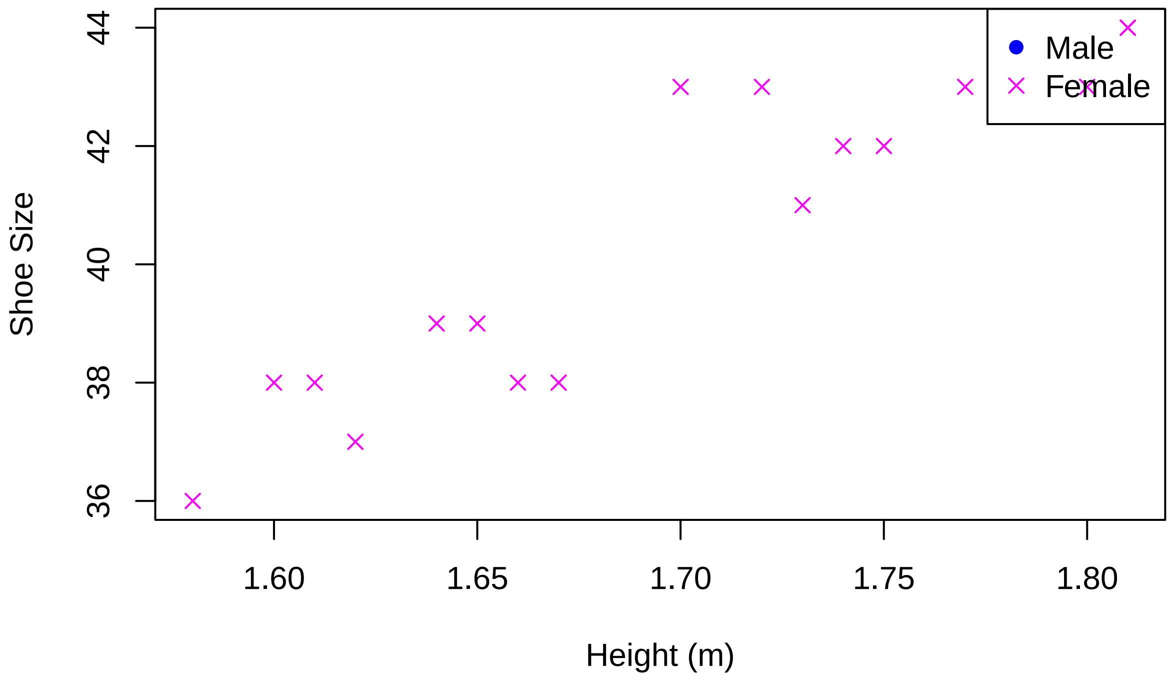
```
##
##  kuopio tampere
##       7      10
```

```r
# Show the distributions in the above item at the same time by
# using a contingency table.

contingency_table <- table(students$gender, students$population)
print(contingency_table)
```

```
##
##          kuopio tampere
##   female      4       5
##   male        3       5
```

```r
# Make two subsets of your dataset by splitting it according to gender.
# Use data frame operations first and then do the same using
# the function subset.

male_students <- students[students$gender == "Male", ]
female_students <- students[students$gender == "Female", ]

# Subsets by gender using subset
male_students_subset <- subset(students, gender == "Male")
female_students_subset <- subset(students, gender == "Female")

# 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.

median_height <- median(students$height)
below_median_height <- students[students$height < median_height, ]
above_median_height <- students[students$height > median_height, ]

# Subsets by height using subset
below_median_subset <- subset(students, height < median_height)
above_median_subset <- subset(students, height > median_height)

# 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
```

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

# Using a loop
for (i in seq_len(nrow(students))) {
  students$height[i] <- students$height[i] / 100
}

# Using apply
students$height <- sapply(students$height, function(x) x / 100)

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

# Correct the conversion of height to meters
students$height <- students$height * 10000

plot(students$height, students$shoesize,
     col = ifelse(students$gender == "Male", "blue", "magenta"),
     pch = ifelse(students$gender == "Male", 16, 4),
     xlab = "Height (m)", ylab = "Shoe Size")
legend("topright", legend = c("Male", "Female"),
       col = c("blue", "magenta"),
       pch = c(16, 4))
```


Shoe Size
Height (m)
● Male
× Female