# Conditional commands

Data tameRs

season 1 / episode 9 All rights reserved. Using without permission is prohibited Press `A` to change slides into text Press `T` to display table of contents.

# What is this episode about?

Due to multiplicity of available data analysis algorithms, we cannot simply apply any chosen method, but we are often forced to look for the best method in our particular case. Each method can be the best one in a given context. In many cases the best solution is to conditionally choose this or that method depending on the type of data that we deal with.

We use conditional commands in order to perform conditionally different groups of commands.

In this episode you will learn:

- How to use conditional commands?
- How to use conditional commands in a loop?

As an illustration we will use the data set `koty_ptaki` available in the package `PogromcyDanych`.

```r
library(PogromcyDanych)
head(koty_ptaki, 3)
```

# Conditional commands

When you are designing a program, you often find yourself in a situation in which you see that your further actions depend on some unknown or potentially changeable value.

For example, when you randomly choose one number from the range 0-1 and you want to write `Heads` or `Tails` depending on the result, you do not know what to write while you are writing the code as you do not know which value you will draw.

Thanks to conditional commands you can write alternative scenarios to be acted out according to the given condition.

The following code will display the word `Heads` on the screen if a number smaller than 0,5 is drawn and the word `Tails` if a number bigger or equal to 0,5 is drawn.

```
liczbaLosowa <- runif(n = 1)

if (liczbaLosowa < 0.5) {
  cat("Heads")
} else {
```

# Conditional commands

When we are preparing a data processing algorithm, we often need to take into account various possible situations. We can use conditional commands to create alternative processing schemes carried out depending on the state of certain variables.

The most commonly used conditional command is `if else`. There are three types of this command in R.

If the `condition` written after the if part is true, the command `expression1` will be performed. Otherwise, the program will carry out the command `expression2` (if it is defined).

The third type specified below is the command `ifelse()` which operates not only on one logical value TRUE/FALSE but on a whole vector of logical values.

```
if (condition)
  expression1

if (condition)
  expression1
```

# Conditional commands

Now I will use the first row from the data frame koty_ptaki to demonstrate various types of the conditional command. We first need to transform the column gatunek into the column of characters. Otherwise names of the species would not be properly displayed. We will work only on the first row but we will save the index of that row in the variable i. This way it will be easier for us to use that code in a loop later on.

```
koty_ptaki$gatunek <- as.character(koty_ptaki$gatunek)
i <- 9
koty_ptaki[i,]
```

```
##   gatunek waga dlugosc predkosc habitat zywotnosc druzyn
## 9   Strus  150     2.5       70  Afryka        45    Pta
```

If we use only the if part, nothing will be displayed on the screen when the condition is false.

```
if (koty_ptaki[i,"druzyna"] == "Kot") {
```

# Conditional commands

If you work on vectors, the easiest solution is to use the vector command `ifelse()`.

Its first argument may be a vector. For example, the command `koty_ptaki[,?druzyna?] == ?Kot?` creates a vector of logical values.

```
koty_ptaki[,"druzyna"] == "Kot"
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FAL
## [12] FALSE FALSE
```

We will use that command in the command `ifelse()` and our result will be a vector of text values with `?Another big cat?` for TRUE value and `?Another fast bird?` for FALSE value.

```
ifelse(koty_ptaki[,"druzyna"] == "Kot",
  "Another big cat",
  "Another fast bird")
```

# Conditional commands and loops

Conditional commands are frequently used inside the loops when a certain operation needs to be repeated several times but its course may have several potential scenarios.

Let me illustrate such situation using an example.

This code will display a different tetx for cats and a different text for birds.

```
# loop will be executed for each row
for (i in 1:nrow(koty_ptaki)) {
  # inside the loop, the execution depends on the conditio
  if (koty_ptaki[i,"druzyna"] == "Kot") {
    cat(koty_ptaki[i,"gatunek"], "is a big and fast cat.\n"
  } else {
    cat(koty_ptaki[i,"gatunek"], "is a bird.\n")
  }
}
```

```
## Tygrys is a big and fast cat.
```

# Conditional commands and loops

Conditional commands are often used to change the value of the variable depending on the condition.

The example below presents a loop with a conditional command which changes the value of variable `liczbaKot?w` in one case or changes the value of the variable `liczbaPtak?w` in the other case.

```
# we create the variables in the first place
liczbaKotow  <- 0
liczbaPtakow <- 0
# inside the loop, we will change the value of those varia
# whether a row describes a cat or a bird
for (i in 1:nrow(koty_ptaki)) {
  if (koty_ptaki[i,"druzyna"] == "Kot") {
    liczbaKotow  <- liczbaKotow + 1
  } else {
    liczbaPtakow <- liczbaPtakow + 1
  }
}
```

# Summary of R instructions

In this episode we have discussed conditional commands.

Below you can see a list of all the commands used in this episode.

```r
# Conditional command which will be executed differently
#depending on the value we draw
randomNumber <- runif(n = 1)
if (randomNumber < 0.5) {
  cat("Heads")
} else {
  cat("Tails")
}

# Conditional command with the if block alone
i <- 9
if (koty_ptaki[i,"druzyna"] == "Kot") {
  cat(koty_ptaki[i,"gatunek"], "is a big cat.")
}
```

# Summary of R instructions

In this episode we have discussed conditional commands.

Below you can see a list of all the commands used in this episode.

```
#Inside loops, when you wish to perform different operatio
#it is easy to use conditional commands

for (i in 1:nrow(koty_ptaki)) {
  # inside the loop, the execution depends on the conditio
  if (koty_ptaki[i,"druzyna"] == "Kot") {
    cat(koty_ptaki[i,"gatunek"], "is a big and fast cat.\n'
  } else {
    cat(koty_ptaki[i,"gatunek"], "is a bird.\n")
  }
}

# With conditional commands we may not only print text,
#but also change the values of the variable
#In the example below, the values of counting variables
```

# Exercises

- Write a conditional command which will display `light` for animals which weight less than 1kg and `heavy` for animals which weight more than 1kg.
- Write a conditional command which will display `light` for animals which weight less than 1kg, `heavy` for animals which weight more than 100kg and `medium` for animals which weight between 1-100kg. You can achieve the same effects using two commands `if()` or the function `switch()`.
- Write a loop and a conditional command summing up weights of all the cats and all the birds separately

You may find sample answers at `https://rawgit.com/pbiecek/MOOC/master/0_dane/9_zadania.html`