Conditional commands

Data tamers

season 1 episode 9

What is this episode about?

Due to multiplicity of available algorithms for data analysis we cannot simply apply any method but we are often forced to look for the best method in our particular case. Each method can be best 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.

**library(PogromcyDanych) head(koty\_ptaki, 3)**

**## gatunek waga dlugosc predkosc habitat zywotnosc druzyna ## 1 Tygrys 300 2.5 60 Azja 25 Kot ## 2 Lew 200 2.0 80 Afryka 29 Kot ## 3 Jaguar 100 1.7 90 Ameryka 15 Kot**

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 on the screen the word Heads 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("Orzeł") } else { cat("Reszka") }**

**## Reszka**

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 false. There are three types of this command in R.

**if (warunek) wyrażenie1 if (warunek) wyrażenie1 else wyrażenie2 ifelse (warunek\_dla\_wektora, wyrażenie1, wyrażenie2)**

If the condition written after the if part is true, the command wyrażenie1 will be performed. Otherwise, the program will carry out the command wyrażenie2 (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 your expression contains more than one command, put it inside the brackets {}.

It is a quite common practice to insert expressions inside the brackets {}even if they contain only one command as it improves readability of the whole code.

**if (warunek) { wyrażenie1 } if (warunek) { wyrażenie1 } else { wyrażenie2 }**

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 t o 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 1. In 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 druzyna ## 9 Strus 150 2.5 70 Afryka 45 Ptak**

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

**if (koty\_ptaki[i,"druzyna"] == "Kot") { cat(koty\_ptaki[i,"gatunek"], "to duży kot.") }**

If we use the command if-else, the word else will be displayed on the screen when the warunek turns out to be false.

**if (koty\_ptaki[i,"druzyna"] == "Kot") { cat(koty\_ptaki[i,"gatunek"], "to duży kot.") } else { cat(koty\_ptaki[i,"gatunek"], "to ptak.") }**

**## Strus to ptak.**

Conditional commands

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

**koty\_ptaki[,"druzyna"] == "Kot"**

**## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE ## [12] FALSE FALSE**

Its first argument may be a vector. For example, the command koty\_ptaki[,”druzyna”] == “Kot” creates a vector of logical values.

We will use that command in the command ifelse() and our result will be a text vector with the values “Kolejny wielki kot” for TRUE value and “Kolejny szybki ptak” for FALSE value.

**ifelse(koty\_ptaki[,"druzyna"] == "Kot", "Kolejny wielki kot", "Kolejny szybki ptak")**

**## [1] "Kolejny wielki kot" "Kolejny wielki kot" "Kolejny wielki kot" ## [4] "Kolejny wielki kot" "Kolejny wielki kot" "Kolejny wielki kot" ## [7] "Kolejny wielki kot" "Kolejny szybki ptak" "Kolejny szybki ptak" ## [10] "Kolejny szybki ptak" "Kolejny szybki ptak" "Kolejny szybki ptak" ## [13] "Kolejny szybki ptak"**

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.

I will display a different formula for cats and a different formula for birds.

***# pętla wykona się dla każdego wiersza* for (i in 1:nrow(koty\_ptaki)) { *# wewnątrz pętli wykonanie zależy od warunku i instrukcji if* if (koty\_ptaki[i,"druzyna"] == "Kot") { cat(koty\_ptaki[i,"gatunek"], "to wielki i szybki kot.\n") } else { cat(koty\_ptaki[i,"gatunek"], "to ptak.\n") } }**

**## Tygrys to wielki i szybki kot. ## Lew to wielki i szybki kot. ## Jaguar to wielki i szybki kot. ## Puma to wielki i szybki kot. ## Leopard to wielki i szybki kot. ## Gepard to wielki i szybki kot. ## Irbis to wielki i szybki kot. ## Jerzyk to ptak. ## Strus to ptak. ## Orzel przedni to ptak. ## Sokol wedrowny to ptak. ## Sokol norweski to ptak. ## Albatros to ptak.**

Conditional commands and loops

Conditional commands are also often used to change the value of the variable for a different value depending on the type of condition.

The example below presents a loop with a conditional command which changes the value of variable liczbaKotów in one case or changes the value of the variable liczbaPtaków in the other case.

***# inicjujemy zmienne* liczbaKotow <- 0 liczbaPtakow <- 0 *# w pętli będziemy zmieniać wartość zmiennych w zależności od tego*  *# czy kolejny wiersz opisuje koty czy ptaki* for (i in 1:nrow(koty\_ptaki)) { if (koty\_ptaki[i,"druzyna"] == "Kot") { liczbaKotow <- liczbaKotow + 1 } else { liczbaPtakow <- liczbaPtakow + 1 } } *# zliczamy ile było kotów i ptaków* liczbaKotow**

**## [1] 7**

**liczbaPtakow**

**## [1] 6**

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.

***# Instrukcja warunkowa, której wykonanie zależy od wylosowanej wartości* liczbaLosowa <- runif(n = 1) if (liczbaLosowa < 0.5) { cat("Orzeł") } else { cat("Reszka") } *# Instrukcja warunkowa z samym blokiem if* i <- 9 if (koty\_ptaki[i,"druzyna"] == "Kot") { cat(koty\_ptaki[i,"gatunek"], "to duży kot.") } *# Instrukcja warunkowa z blokiem if-else* if (koty\_ptaki[i,"druzyna"] == "Kot") { cat(koty\_ptaki[i,"gatunek"], "to duży kot.") } else { cat(koty\_ptaki[i,"gatunek"], "to ptak.") } *# Warunki logiczne można wykonywać na wektorach wartości* koty\_ptaki[,"druzyna"] == "Kot" *# Instrukcja warunkowa ifelse() działa też na wektora* ifelse(koty\_ptaki[,"druzyna"] == "Kot", "Kolejny wielki kot", "Kolejny szybki ptak")**

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.

***# W pętlach, gdy dla różnych wierszy należy wykonać inne operacje wygodnie jest korzystać*  *# z instrukcji warunkowych* for (i in 1:nrow(koty\_ptaki)) { *# wewnątrz pętli wykonanie zależy od warunku i instrukcji if* if (koty\_ptaki[i,"druzyna"] == "Kot") { cat(koty\_ptaki[i,"gatunek"], "to wielki i szybki kot.\n") } else { cat(koty\_ptaki[i,"gatunek"], "to ptak.\n") } } *# W instrukcjach warunkowych nie tylko możemy wyświetlać różne napisy,* *# ale również zmieniać wartości zmiennych.* *# W poniższym przykładzie w zależności od tego jaki wiersz jest odczytywany*  *# inaczej zmieniamy wartości liczników* liczbaKotow <- 0 liczbaPtakow <- 0 for (i in 1:nrow(koty\_ptaki)) { if (koty\_ptaki[i,"druzyna"] == "Kot") { liczbaKotow <- liczbaKotow + 1 } else { liczbaPtakow <- liczbaPtakow + 1 } } *# zliczamy ile było kotów i ptaków* liczbaKotow liczbaPtakow**

Exercises

* Write a conditional command which will display lekkie for animals which weight less than 1kg and ciężkie for animals which weight more than 1kg.
* Write a conditional command which will display lekkie for animals which weight less than 1kg, ciężkie for animals which weight more than 100kg and średnie 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.

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