Loops

season 1 /espisode 8

What is this episode about?

Real strength of modern computers is their ability to repeat certain operations at a very high speed. Our computers are not very intelligent (as yet) but they are very fast and, what is more, they are great at repeating, repeating, repeating...

When we analyze data we very frequent want to repeat certain operations. We may for example want to repeat it for each variable, each day or each data set undergoing analysis.

A tool which allows us to repeat again and again certain operations is ... a loop.

In this episode you will learn:

* How to create loops?
* What are loops used for?
* How to use the loop for()?
* How to create a loop in the loop?
* What are loops in the loops used for?

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**

Loops

You can use loops to repeat certain operations a given number of times.

Let us start with an example in which we will display a certain text on the screen several times. R language contains a number of types of loops which can be used for that purpose but the most popular loop is the loop for(). I would like to present first this particular loop.

We would achieve the same desired effect, if we entered the command cat() (which displays its arguments on the screen) five times.

But what if we wanted to repeat that function 10 000 times? I am sure that none of us would like to keep entering these commands for several days. Especially so, if the same effect can be achieved with a simple loop.

**odliczajDo <- 5 for (i in 1:odliczajDo) { cat("Wartość zmiennej i: ", i, "\n") }**

**## Wartość zmiennej i: 1 ## Wartość zmiennej i: 2 ## Wartość zmiennej i: 3 ## Wartość zmiennej i: 4 ## Wartość zmiennej i: 5**

**cat("Wartość zmiennej i: 1\n") cat("Wartość zmiennej i: 2\n") cat("Wartość zmiennej i: 3\n") cat("Wartość zmiennej i: 4\n") cat("Wartość zmiennej i: 5\n")**

Loops

Instructions for use (the so-called loop syntax) of the for loop are as follows:

The command C will be performed for every value from the vector B as a result of that loop. Moreover, the current loop interval is determined by the variable A and this value can be used inside the C command.

In the example presented on the previous slide the command cat("Wartość zmiennej i: ", i, "\n") is performed for each element of the vector 1:odliczajDo. The variable i assumed the subsequent value from the vector 1:odliczajDo at every phase of the loop.

**for (i in 1:odliczajDo) { cat("Wartość zmiennej i: ", i, "\n") }**

Loops

It was easy to predict the result of the whole loop from the previous example. It displays five subsequent numbers from 1 to 5.

Now I would like to present to you a more complex example which simulates the so-called damage process. Let us assume that we have a fleet of cars. This fleet may take part in X number of car accidents per year (we will select the number of X at random). Repair of damages caused by each bump will cost from 0 to PLN 10 000.

In this example the function runif() is used to draw n numbers from the range starting from min and ending at max. Firstly, we use the function to draw the number of collisions within the range from1 to 10. The function round() is used to round down the results to whole numbers.

At each stage of the loop subsequent values are randomly chosen and displayed on the screen. Although this loop appears to be simple and quite innocent, a very similar manner of drawing the size of damage is really used in management of uncertainty concerning amount of damage resulting from car accidents or other accidental events which may concern real companies.

***# losujemy liczbę stłuczek* liczbaStluczek <- round(runif(n = 1, min = 1, max = 10)) *# dla każdej losujemy wartość szkody* for (i in 1:liczbaStluczek) { cat("Stłuczka ", i,", a jej koszt naprawy to ", runif(n = 1, min=0, max=10000), " PLN\n", sep = '') }**

**## Stłuczka 1, a jej koszt naprawy to 4138.41 PLN ## Stłuczka 2, a jej koszt naprawy to 1559.58 PLN ## Stłuczka 3, a jej koszt naprawy to 9179.269 PLN ## Stłuczka 4, a jej koszt naprawy to 4175.649 PLN ## Stłuczka 5, a jej koszt naprawy to 1366.501 PLN ## Stłuczka 6, a jej koszt naprawy to 7.79279 PLN ## Stłuczka 7, a jej koszt naprawy to 9354.869 PLN ## Stłuczka 8, a jej koszt naprawy to 930.7848 PLN ## Stłuczka 9, a jej koszt naprawy to 2526.503 PLN**

Loops

As you have seen in the previous case, the number of repetitions performed by the loop does not need to be predetermined. It may depend on e.g. number of rows or columns in a data frame.

In the example presented below a list of various values from each column of the data frame koty\_ptaki is displayed. The next slide will explain to you in detail what is going on in the code that you can see below.

***# wektor z nazwami kolumn* kolumny <- colnames(koty\_ptaki) kolumny**

**## [1] "gatunek" "waga" "dlugosc" "predkosc" "habitat" "zywotnosc" ## [7] "druzyna"**

***# pętla, która dla każdej kolumny wyświetla różne wartości w niej występujące* for (i in kolumny) { wartosciWKolumnie <- unique(as.character(koty\_ptaki[,i])) cat("Kolumna", i, "\n") cat(" ", length(wartosciWKolumnie), "różnych wartości:", wartosciWKolumnie, "\n") }**

**## Kolumna gatunek ## 13 różnych wartości: Tygrys Lew Jaguar Puma Leopard Gepard Irbis Jerzyk Strus Orzel przedni Sokol wedrowny Sokol norweski Albatros ## Kolumna waga ## 13 różnych wartości: 300 200 100 80 70 60 50 0.05 150 5 0.7 2 4 ## Kolumna dlugosc ## 10 różnych wartości: 2.5 2 1.7 1.4 1.3 0.2 0.9 0.5 0.7 0.8 ## Kolumna predkosc ## 12 różnych wartości: 60 80 90 70 85 115 65 170 160 110 100 120 ## Kolumna habitat ## 6 różnych wartości: Azja Afryka Ameryka Euroazja Polnoc Poludnie ## Kolumna zywotnosc ## 10 różnych wartości: 25 29 15 13 21 12 18 20 45 50 ## Kolumna druzyna ## 2 różnych wartości: Kot Ptak**

Loops

Let us discuss line by line the loop from the previous slide.

The first command uses the function colnames() to determine names of the columns. These names will be attributed to the vector kolumny, which will be in turn displayed on the screen.

The command for() begins the loop. This time the variable i will cover names of columns. Loop body for name of each column will be executed.

A vector of various values is determined inside the loop. Although several commands are necessary to create such vector, they are included in one line. The starting point is the vector koty\_ptaki[,i]. The function as.character() transforms all these values (for some columns values are numbers and for others text) into text. Next, the function unique() removes repeated elements.

The next two commands cat() display on the screen the name of the column in the first line and the number of various values in the second line (length of the vector wartosciWKolumnie is specified by the function lenght()) as well as all the various values.

***# wektor z nazwami kolumn* kolumny <- colnames(koty\_ptaki) kolumny**

***# pętla, która dla każdej kolumny wyświetla różne wartości występujące w tej kolumnie* for (i in kolumny) {**

**wartosciWKolumnie <- unique(as.character(koty\_ptaki[,i]))**

**cat("Kolumna", i, "\n") cat(" ", length(wartosciWKolumnie), "różnych wartości:", wartosciWKolumnie, "\n") }**

Loops

The typical use of loop is repetition of a certain operation with reference to each row or each column of the data set.

I would like to show you now an example of use of loop for each row of the data set. Our first example is a loop for a row with index i=1.

I want to display information on the maximum weight of the representatives of a given species.

***# Zamiana zmiennej factor na zmienną napisową, po to by była dobrze wyświetlana* koty\_ptaki$gatunek <- as.character(koty\_ptaki$gatunek) *# wykonamy instrukcję dla pierwszego wiersza, indeks wiersza zapiszemy w zmiennej i* i <- 1 cat(koty\_ptaki[i, "gatunek"], "może ważyć do", koty\_ptaki[i, "waga"], "kg\n")**

**## Tygrys może ważyć do 300 kg**

Loops

We will repeat that operation for each row. You can check the number of rows in the data frame with a command nrow() (from number of rows).

The sequence 1:nrow(koty\_ptaki) creates a vector of numbers from 1 to13 (as there are 13 rows). For each row we need to activate the command from the previous slide, which displays weight of subsequent species.

**for (i in 1:nrow(koty\_ptaki)) { cat(koty\_ptaki[i,"gatunek"], "może ważyć do", koty\_ptaki[i,"waga"], "kg\n") }**

**## Tygrys może ważyć do 300 kg ## Lew może ważyć do 200 kg ## Jaguar może ważyć do 100 kg ## Puma może ważyć do 80 kg ## Leopard może ważyć do 70 kg ## Gepard może ważyć do 60 kg ## Irbis może ważyć do 50 kg ## Jerzyk może ważyć do 0.05 kg ## Strus może ważyć do 150 kg ## Orzel przedni może ważyć do 5 kg ## Sokol wedrowny może ważyć do 0.7 kg ## Sokol norweski może ważyć do 2 kg ## Albatros może ważyć do 4 kg**

Vectorization

Loops can be very often avoided as R works on vectors quite naturally.

For example, you can achieve the same effect as the one presented in the previous slide directly using column vectors. Both koty\_ptaki[,”gatunek”] and koty\_ptaki[,”waga”] are vectors. The function paste() will combine them and make a text vector out of them, while the function cat() will display all the elements of that newly formed vector.

In R, as well as in any well developed programming language, the same effect can be achieved in many different ways.

**cat(paste(koty\_ptaki[,"gatunek"], "może ważyć do", koty\_ptaki[,"waga"], "kg\n"))**

**## Tygrys może ważyć do 300 kg ## Lew może ważyć do 200 kg ## Jaguar może ważyć do 100 kg ## Puma może ważyć do 80 kg ## Leopard może ważyć do 70 kg ## Gepard może ważyć do 60 kg ## Irbis może ważyć do 50 kg ## Jerzyk może ważyć do 0.05 kg ## Strus może ważyć do 150 kg ## Orzel przedni może ważyć do 5 kg ## Sokol wedrowny może ważyć do 0.7 kg ## Sokol norweski może ważyć do 2 kg ## Albatros może ważyć do 4 k**

A loop in a loop

Loops can be placed inside other loops. At first glance it might seem strange but it is very common practice.

Let us assume that we want to calculate something, e.g. a correlation, similarity or difference between each pair of columns. How to do it? The easiest way is to use two loops –one loop inside the other one.

**for (i in colnames(koty\_ptaki)) { for (j in colnames(koty\_ptaki)) { *# poniższa instrukcja wykona się dla każdej kombinacji nazw kolumn i i j* cat("Kolumna '", i, "' i kolumna '", j, "' mają w sumie", nchar(i)+nchar(j), "znaków.\n") } }**

A loop in a loop

We will use the possibility of placing one loop in another loop to visualize data on speed of animals.

Let us start from one loop which will present speed with dots. The more dots, the higher the speed. Each dot on the screen will stand for 5 km/h.

The first row in the data set koty\_ptaki describes speed of the Gyrfalcon. We will read it (with the command koty\_ptaki[i,”predkosc”]; then we will calculate the number of dots which we need to use to present speed (by dividing by 5). At the end we will draw the dots using the command cat().

***# używamy pętli by rysować koty\_ptaki[i,"predkosc"] / 5 kropek* i <- 1 n\_kropek <- koty\_ptaki[i,"predkosc"] / 5 for (i in 1:n\_kropek) { cat(".") }**

**## ............**

A loop in a loop

Let us now repeat the loop for each row containing data in the data frame koty\_ptaki, that is, let us repeat the loop for each different species.

Now we will repeat the loop presenting speed of the first species, but this time the loop will apply to each row separately. Speed is presented by a series of dots followed by name of the given species.

***# Dla każdego wiersza w tabeli `koty\_ptaki`* for (i in 1:nrow(koty\_ptaki)) { n\_kropek <- koty\_ptaki[i,"predkosc"] / 5 *# rysowanie kropek* for (j in 1:n\_kropek) { cat(".") } *# nazwa gatunku* cat(" ", koty\_ptaki[i,"gatunek"], "\n") }**

**## ............ Tygrys ## ................ Lew ## .................. Jaguar ## .............. Puma ## ................. Leopard ## ....................... Gepard ## ............. Irbis ## .................................. Jerzyk ## .............. Strus ## ................................ Orzel przedni ## ...................... Sokol wedrowny ## .................... Sokol norweski ## ........................ Albatros**

Summary of R commands

We have discussed the loop for in this episode. It is one of the most popular loops. When you master functions in the episode 10, you will be taught how to use also other loops which have similar properties as for() but are much shorter/easier to write.

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

***# pętla odliczająca od 1 do 5* odliczajDo <- 5 for (i in 1:odliczajDo) { cat("Wartość zmiennej i: ", i, "\n") } *# pętla losująca wartości stłuczek dla losowej liczby stłuczek* liczbaStluczek <- round(runif(n = 1, min = 1, max = 10)) *# dla każdej losujemy wartość szkody* for (i in 1:liczbaStluczek) { cat("Stłuczka ", i," a jej koszt naprawy to", runif(n = 1, min=0, max=10000), "PLN\n") } *# pętla wypisująca liczbę różnych wartości i listująca te wartości dla każdej kolumny* kolumny <- colnames(koty\_ptaki) for (i in kolumny) { wartosciWKolumnie <- unique(as.character(koty\_ptaki[,i])) cat("Kolumna", i, "\n") cat(" ", length(wartosciWKolumnie), "różnych wartości:", wartosciWKolumnie, "\n") }**

Summary of R commands

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

***# pętla wypisująca dla każdego wiersza ciężar zwierzaka* for (i in 1:nrow(koty\_ptaki)) { cat(koty\_ptaki[i,"gatunek"], "może ważyć do", koty\_ptaki[i,"waga"], "kg\n") } *# używając operacji na wektorach czasem można uniknąć pisania pętli* cat(paste(koty\_ptaki[,"gatunek"], "może ważyć do", koty\_ptaki[,"waga"], "kg\n")) *# pętla w pętli pozwala na wykonanie pewnej instrukcji dla każdej pary kolumn* for (i in colnames(koty\_ptaki)) { for (j in colnames(koty\_ptaki)) { cat("Kolumna '", i, "' i kolumna '", j, "' mają w sumie", nchar(i)+nchar(j), "znaków.\n") } } *# używając pętli w pętli możemy też stworzyć tekstową wizualizację danych* for (i in 1:nrow(koty\_ptaki)) { n\_kropek <- koty\_ptaki[i,"predkosc"] / 5 *# rysowanie kropek* for (j in 1:n\_kropek) { cat(".") } *# nazwa gatunku* cat(" ", koty\_ptaki[i,"gatunek"], "\n") }**

Exercises

* Write a loop which will display the lifespan of given species from each row of the table koty\_ptaki.
* Write a loop which will display lifespan as a diagram in which one year is presented as one #.
* Write a loop which will draw a diagram presenting lifespans and place names of species on the left-hand side. Moreover, before each name of the species insert spaces so that the names were aligned right. You can read the length of the name expressed in the number of characters with the function nchar().

Sample answers are available at https://rawgit.com/pbiecek/MOOC/master/0\_dane/9\_zadania.html