**Data Cleaning and Introduction to Data Processing**

pogRomcy danych (Data Masters) Season 1 / Episode 16

All rights reserved.  
The use shall be subject to prior consent of the author.

Press A to see the plain text instead of the slides.  
Press T to display the table of contents.

# What Is This Episode About?

Working with the data mostly includes cleaning and repeated change of the representations. Why the data need to be cleaned? Both the data collected by humans and the automatic sensors tend to contain gaps, outliers, or common mistakes. In order to meaningfully analyse the data, we need to convert them into the forms, in which the results of our analysis are reliable.

Cleaning is a very important phase. Because no matter how sophisticated and well-chosen our data analysis tool is, if the input data are garbage, the output data will be garbage, too. Garbage in garbage out

In this episode, we will learn:

* How to edit the sets of data to make them coherent.
* How to replace one value by another.
* How to identify the outliers / redundant data.

As soon as the data are cleaned, we can move on to another stages, i.e. pre-processing, analysis, and visualisation.

# Data Creation Using ‘data.frame()’ Function

For the purpose of data cleaning exercise, we will create an artificial data frame with a few errors intentionally added.

We create the data frame with the function data.frame(), where the subsequent arguments are the vectors, which will be converted into the subsequent columns in the data frame. The last argument stringsAsFactors = FALSE defines that the strings of characters are NOT to be replaced with the qualitative variables (replacement is default).



We will want to correct the following errors in the resulting frame:

* replace the name of the column oczo by oczy
* replace the variable wiek by quantitative variable; for this purpose we need to replace , with .first.
* replace the name Zosia by Zofia
* replace the eye colours with two groups niebieskie and ciemne.

# Column Name Replacement

To view the column name, you can use the function colnames(). The result is a vector of strings; you can use the indices for the elements of this vector, as in the following example. Similarly, the function rownames() allows you to work with the names of rows.

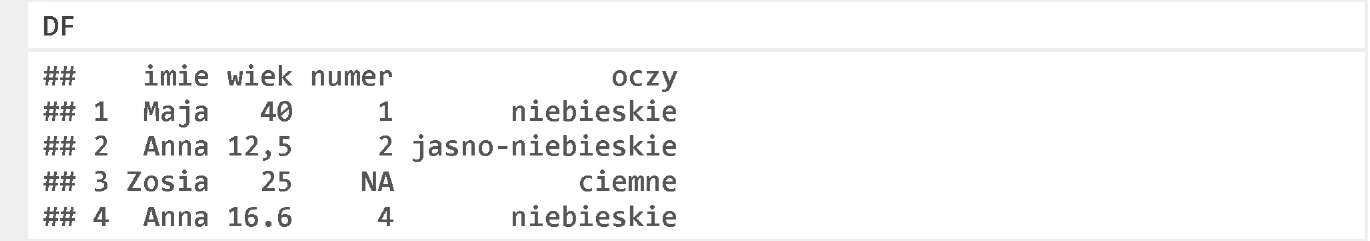


The function colnames() can also be used to change the column names. The following two instructions will run the function colnames<-(), which will change the column names. You can also change all or only selected vector elements of the column names.

Both of the below instructions will correct the name of the last column.

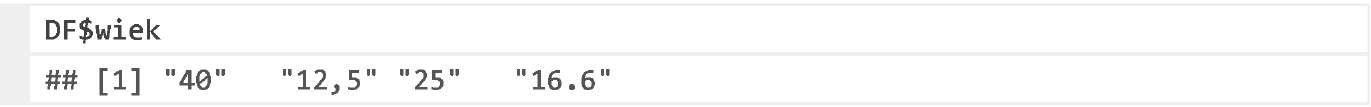


It will yield a set of data.



# Conversion of Comma into Dot

If we are working on the data entered by a human, it is often the case that she or he alternates between . and ,. This problem escalates if the data are entered by several different persons.

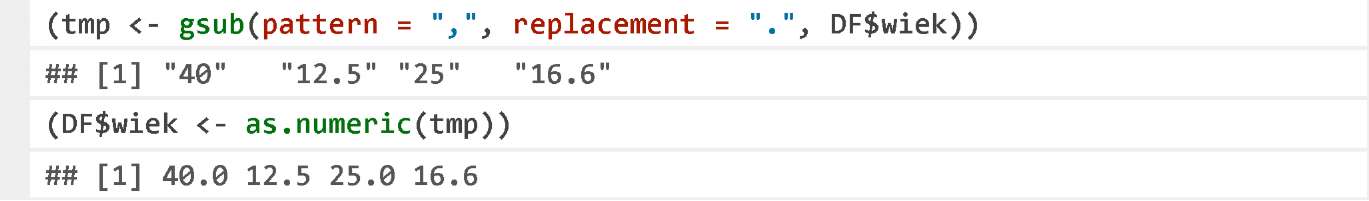


We will replace the comma with the dot, and convert the strings of characters into the numbers.

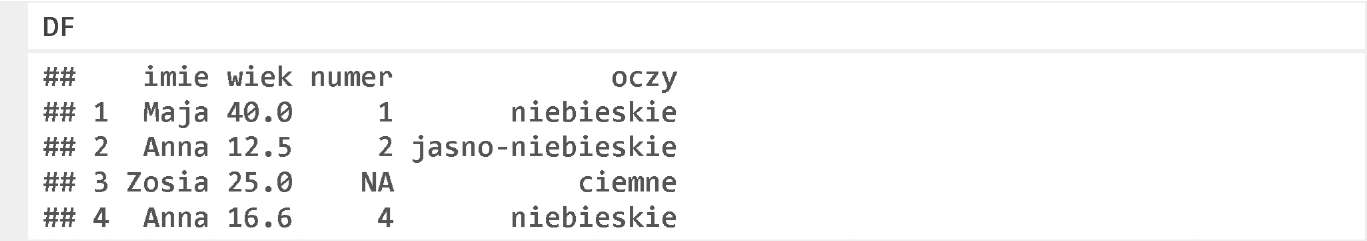
To replace the characters, we will use the function gsub(), (also described in the Episode 14 of this Season). The argument names pattern and replacement can be omitted as their order is default, but we enter them here for clarity.

Then, we use the function as. numeric () to convert the strings of characters into the numbers. Finally, we assign the result to the column wiek.

We apply the parentheses () to two lines to display the assignment result on the screen.



It will yield an amended set of data.



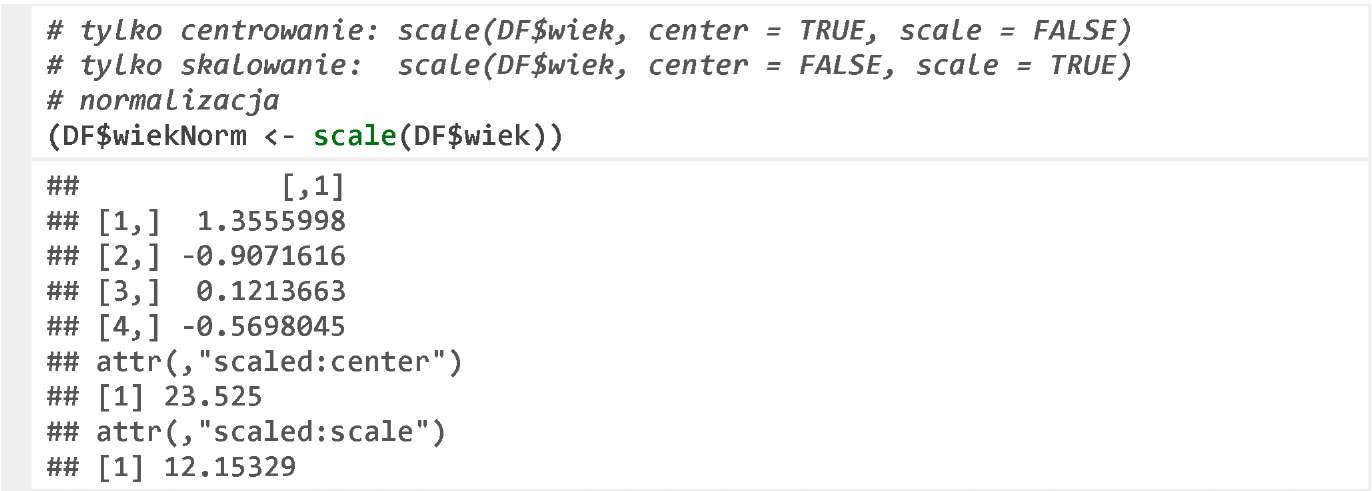
# Column or Row Normalisation

In analysing the data, we often need to normalise the data in a column.

To normalise or centralise, i.e. to remove a mean value from them (so as the mean value equals 0), and re-scale, i.e. divide so as the standard deviation equals 1.

For normalisation, the function scale() can be used. By default, it centralises and scales the data, but by setting additional argument, we can only centralise or re-scale the data.

The result of the function scale() can be assigned to the column of the set of data.



As a result we will get a new, extended set of data.



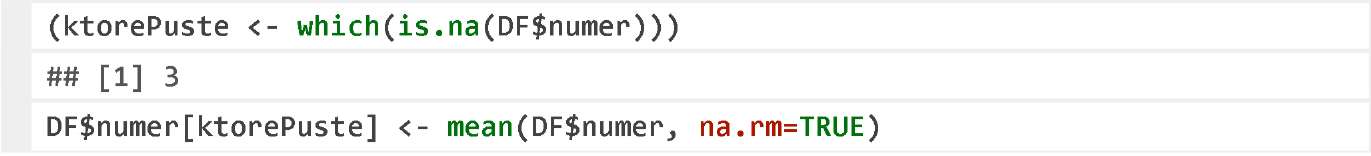
# Removal or Replacement of Missing Data

In the column numer, there are missing values NA.

These values can be removed (i.e. the whole row needs to be removed) using the function na.omit().



Alternatively, replace with, for example, a mean value in the column. In the following example, the function is.na() identifies the missing values, the function which() determines the indices of the missing values. Next, these indexes will be used to overwrite the missing values by the mean value determined by the function mean().

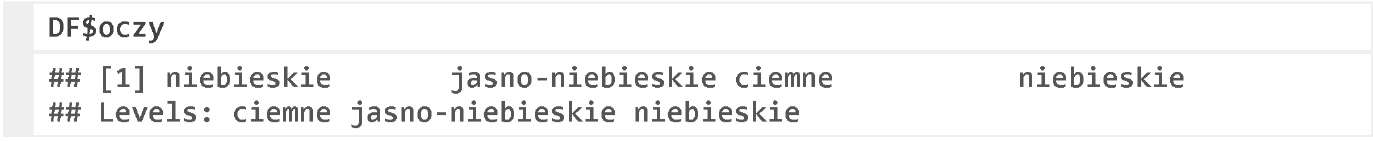


As a result we will get a new, complete set of data.



# Names of Qualitative Variable Factors

If we process a qualitative variable, we often change the level names, order, or paste several levels into one.



To view or change the level names, you can use the function levels(). Similarly as for colnames (), we can use this function to view or change the names vector.

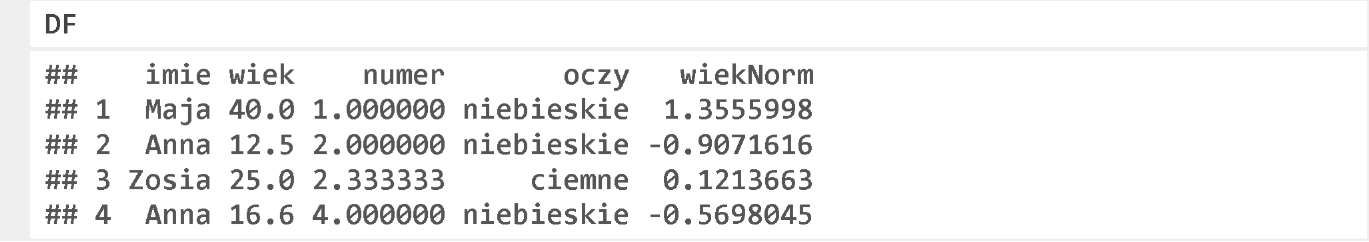
If we assign the values with the repeated strings of characters to such vector, the defined levels will merge into one.



If we want to change the level order in a qualitative variable, the easiest way to do it is to apply the function factor(). We can specify any order of the levels in the argument levels. The values are the same, but the order of levels is different. The level order is important in creating the charts (it corresponds to the order of the levels of the chart) and in the statistical modelling.



As a result we will get a new, amended set of data.



# Replacement of Any Value

It is often the case that we need to replace all values by others in a column. Perhaps someone entered 100.00 instead of 10000 by mistake, and you need to replace one value by the other. Perhaps someone wrong typed the name of a therapy, and we want to change these names.

In the following example, we will replace the name Anna with Joanna. The comparison result returns the vector of the logic values.



This vector can be used to index a row. If we assign one value to a several-element vector, as illustrated below, then all elements of such vector will be replaced by this value.

C:\Users\Kasia\AppData\Local\Temp\FineReader10\media\image18.jpeg

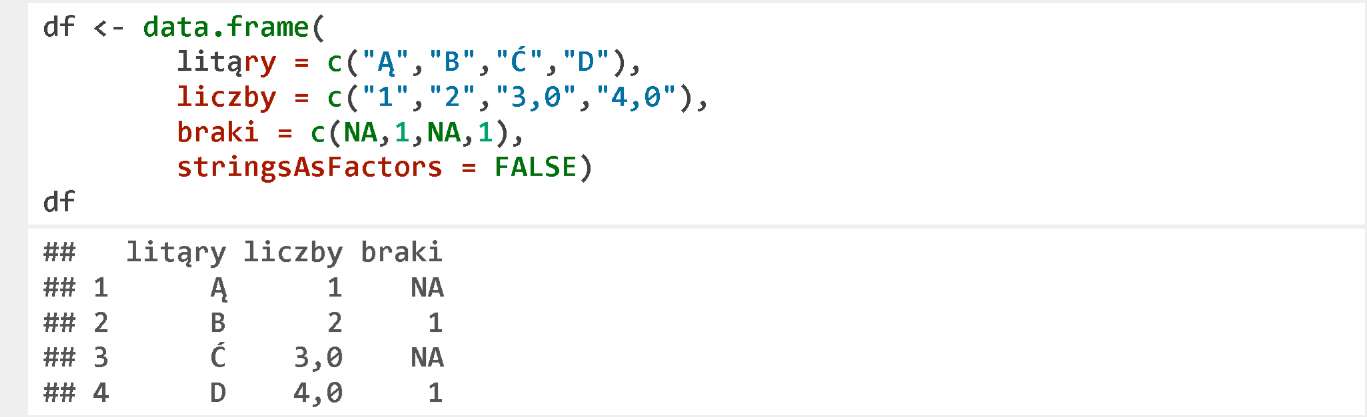
As a result of this instruction, we will get a new, amended set of data.



# Exercises

After loading the below data frame, correct the following errors:

* correct the column name litąry,
* replace the value Ą and Ć to remove a ‘tail’,
* replace the column liczby by the number column,
* replace the missing data in the column braki.



# What’s Next?

We have only outlined the topic of data cleaning on these several slides. For more information on improving the data sets, please refer to the following publications:

* Chapter 3.3 “Pre-Processing of Data” in the book “R Package Guide” available at http://biecek.p1/R/
* The English book titled “An introduction to Data Cleaning with R” at http://cran.r- project.org/doc/contrib/de\_Jonge+van\_der\_Loo-Introduction\_to\_data\_c1eaning\_with\_R.pdf
* The English course on the Courser’s website “Getting and Cleaning Data” at https://www.coursera.org/course/getdata