Logic variables

Data Crunchers

Season 1 / Episode 13

What is this episode about?

When you analyze data you will encounter many types of variables. We have already discussed qualitative and quantitative variables. Logical variables are a specific variation of qualitative variables. They assume logical values true/false.

In this episode you will learn:

* Which variables/ characteristics are referred to as logical variables?
* What basic operations can be performed on logical variables?
* How to summarize/ describe logical variables?

We will use two data sets to present these issues. The first set, koty\_ptaki is quite small, while the other one, auta2012 is a much more extensive data set. Both data sets are available in the package PogromcyDanych.

What does it mean: a logical variable

Logical variables assume only two logical values –true and false. For example: is the height more or less than 150cm? Is it raining or not? Did the patient live 5 years after the operation?

Logical variables are often created in the process of converting quantitative variables into qualitative variables. When we have data on education described by three values we can create an artificial characteristic –has the person got higher education. And when we have data on height in the form of quantitative variables we can construct a logical variable –is the person higher than 150cm.

Logical variables can be treated as qualitative variables as they also assume values from a two-element dictionary. What is more, we can draw a contingency table for both qualitative and logical values.

However, there are some additional logical operations which can be performed only on logical variables.

Three-valued logic

It was stated above that logical values assume values TRUE (logical truth) and FALSE (logical falsity). Yet at the same time any type of variables may also assume in R the unknown value NA. Thus, as a matter of fact logical values operate in three-valued logic.

We often use logical operators such as > (greater than), >= (greater than or equal to), == (equal to), <= (less than or equal to), < (less than) to create logical values. When we compare the unknown value NA using these operators, the result is also the NA value.

Instead of full names TRUE and FALSE you can also use abbreviations T and F when creating vectors of logical values.

Special operators for logical values are | (logical or) and & (logical and). The result of the first operator is the value TRUE if any of its arguments is true. The result of the second operator is TRUE if both arguments are true.

Logical addition and multiplication table

Logical addition table is very useful when you want to present graphically potential results of the operator |. Let us define a vector containing all three possible logical values in order to present that table.

Arguments of the function outer() are two vectors. It applies a function, which becomes its third argument, to each pair of values from the first and the second vector. Another two lines add names of rows and columns so that the result was a nice looking table of logical addition. If the value of any of the arguments is logical truth, then the result of the operator or is also logical truth.

We can create a logical multiplication table in the same way using the operator &.

You should pay special attention to the result of (TRUE | NA). The result is the value TRUE because the logical sum of NA and TRUE is TRUE irrelevant of the real nature of the unknown value. Likewise, the result of the operation FALSE & NA is FALSE.

There are also longer versions of the operators | and & in R, namely || and &&. There are two differences between the short and the long versions. The short version works on vectors and it performs operations on each subsequent elements of vectors, while the long version performs operations on only two first elements of vectors and it return one-element value TRUE or FALSE as a result. The second difference between them concerns more advanced uses. In such cases the long version does not evaluate certain argument if it is not essential.

For example, the first line of the following example (literówka) will be performed correctly because evaluation of the function cat() is essential for determination of the result. The second example will perform the function cat() and then it will report an error as it is impossible to logically add the result of the function cat() to the value TRUE.

Loading data

We will start our examples with koty\_ptaki from the package PogromcyDanych. The episode 2 provides instructions for loading that data.

Let us activate the package and use the function head() to display the first six rows.

Descriptive statistics

We will build the logical variable czy\_to\_kot on the basis of the variable druzyna and a logical variable czy\_jest\_ciezki on the basis of the variable waga.

When we treat the logical variable as a two-element qualitative variable we can apply to it e.g. functions for frequency distribution tables, both one-dimensional and two-dimensional.

When you look at the frequency distribution table including the two variables you will notice that there is only one animal in the whole data set which weights over 10kg and it is not a cat.

Negation operator

Another useful operator is logical negation !. It converts the value of the logical truth into logical falsity and vice versa.

In arithmetic operations logical values TRUE are converted into 1 and logical values FALSE into 0. It allows us to count number of TRUE values by summing up the value of the vector. The following example shows how many cats and not-cats are there in the data set.

Such vector of logical values can be multiplies by a vector of numerical values. The instruction presented below uses just such an operation to calculate a sum of all the cats in the data set. When logical values are multiplied by a vector, certain values are reset.

Loading data

The data set koty\_patki consists of 13 rows. All of them can be displayed on the computer screen. For such small sets we do not need special descriptive statistics to get the gist of their content.

For this reason from now on we will practice working with qualitative variables ? on a much bigger data set containing over 200 thousand values called auta\_2012, which is also available in the package PogromcyDanych.

You can find a description of this data set in the episode https://rawgit.com/pbiecek/MOOC/master/0\_dane/0\_dane.html

Let us load this set and look at its first three rows.

Descriptive statistics

Let us now use logical values and the function table() to check how many cars from our data set is registered in Poland and how many of them were manufactured before 2007. Just like in case of qualitative variables we can use the function prop.table() to determine fractions/percentages.

It may be a little surprising, but less than half of the offers gives Poland as the country of current registration.

Exercises

* Check how many cars are offered for less than PLN 2000.
* Check what percentage of all cars has engines with cylinder capacity of 1500cm3.

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

Tests of types and values

You have to pay special attention to unspecified values when you compare values. NA values does not always behave as we would expect although they are always logical.

For example, the operator != tests whether the values are different (negation of the operator ==). How can we check if the values in the vector are different from the unknown value NA? Let us first see how it shouldn’t be done.

Result of the comparison != NA is a vector of the NA value. It results from the fact that if the right part of the comparison is unknown, then we cannot know if the value 41000 is equal to or different from it.

We can use the function is.na() to check if a given value is an unknown value. This function returns TRUE is the tested value is NA and FALSE if the value is known.

We can also use that function to calculate percentage of values with missing data.

There are more functions for testing data. They all start with is. so if you want to display a list of these functions, you have to write these three characters into the console and press TAB. Some of the functions check whether a given variable is of a certain type (e.g. is.factor() or is.numeric()); others check whether a given variable has certain values (e.g. is.na() or is.nan()).

Logical ‘or’ and logical ‘and’

As operators of logical or and and are frequently used let us practice using them a little more. We will perform operations on the data set concerning cars. Our task is to define two vectors of logical values which will in turn verify whether a given car is currently registered in Poland and whether it is more than 5 years old –and was produced before 2007.

Now we will calculate the logical sum and the product of both vectors. Then we will use the function table to present our results.

Exercises

* Check how many cars registered in Poland are offered for less than PLN 2000.
* Check what percentage of all cars has engines with cylinder capacity of 1500cm3 and runson diesel oil.

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