**Determination of New Data Variables [dplyr / mutate]**

pogRomcy danych (Data Masters) Season 1 / Episode 21

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?**

Very often, we create the new data on the basis of the existing ones. With speed in metres per second, we can compute speed in kilometres per hour. With weight and height, we can calculate BMI. With distance in miles, we can compute distance in kilometres, and so on.

The R offers many different ways of adding a new variable to the set of data. The method, which is the most convenient and quick of all, is the use of the function mutate() of the package dplyr.

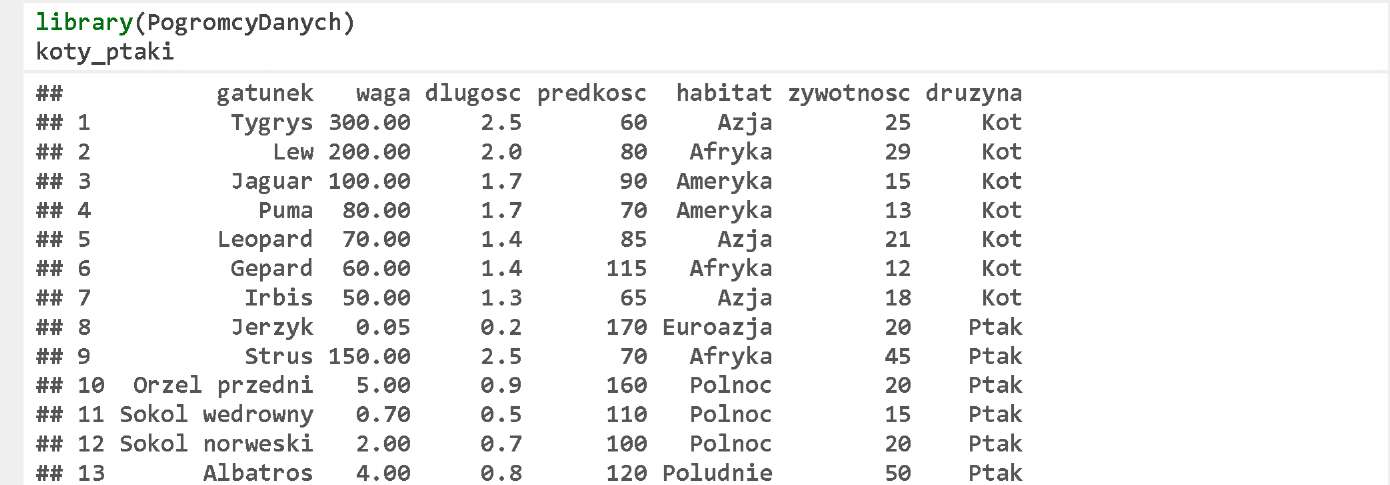
In this Episode, we will learn:

* How to add a new column to a set of data.
* How to overwrite the values in an existing column.

To illustrate these issues, we will use two sets of data. The first small set of data is koty\_ptaki and the second, considerably bigger one is auta2012, both available in the PogromcyDanych package.

**Birds Vs Cats**

Let’s start with an example how to create the new variables using the cats and birds set of data. It is small enough to display the result of the operations on the screen.

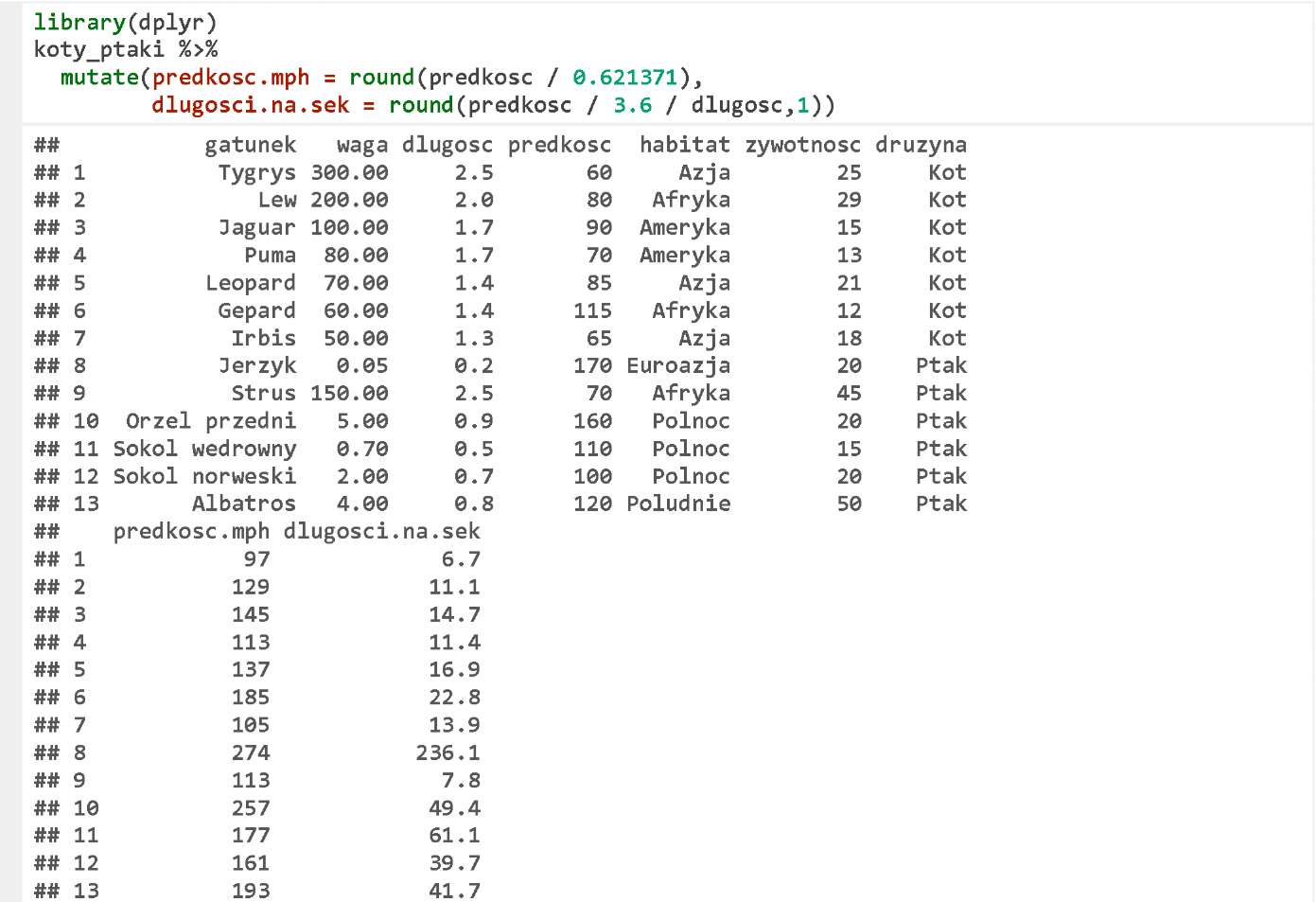


**Computation of New Variables**

In the set of data koty\_ptaki, there is a variable predkosc, which describes velocity in kilometres per hour.

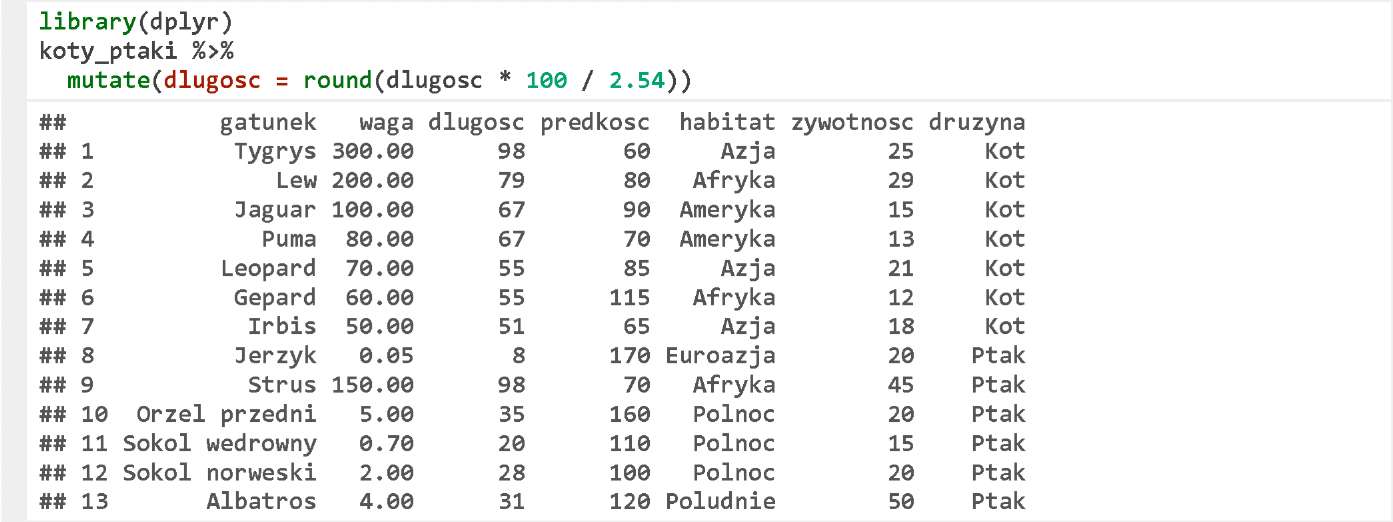
Let’s add two new variables. In one, we are going to convert speed from kilometres per hour to miles per hour, and in the other, we will convert speed into the number of the lengths per second (the animal length is in metres). A new variable can be added using mutate(). The first argument is the data frame, and the other are the new variable statements in a form of nazwa.zmiennej = wzór.na.nową.zmienną.

For processing the data, we will use the operator %>% discussed in the Episode 20.



**Variable Overwriting**

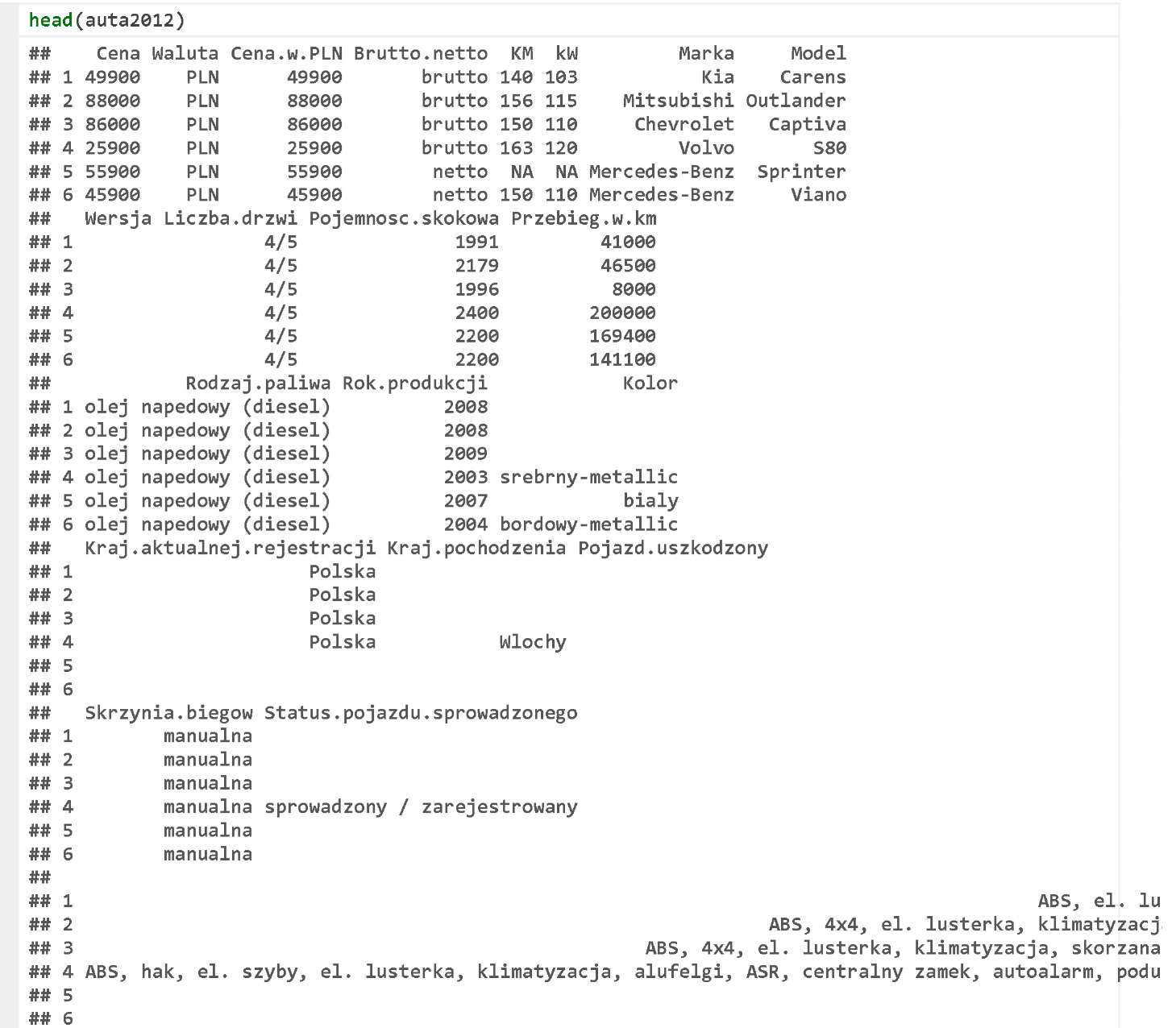
The function mutate () can also be used to change the values of a variable, which is already present in the set of data. All you need is to assign a new value to the name of an existing column. In the example below, we convert the length into inches, and round up or down the result to the total numbers.



**Cars**

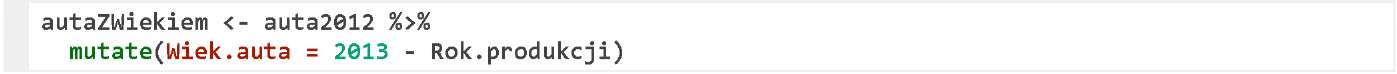
We will also practice the operations of creating the new variables using much larger and more complex set of data. Namely, the set of data on the second-hand car offer prices.

The set of data auta2012 is available after downloading the package PogromcyDanych. Please find below six first rows of this set of data. You can find the detailed description of this set at https://rawgithub.com/pbiecek/MOOC/master/O\_dane/O\_dane.html.

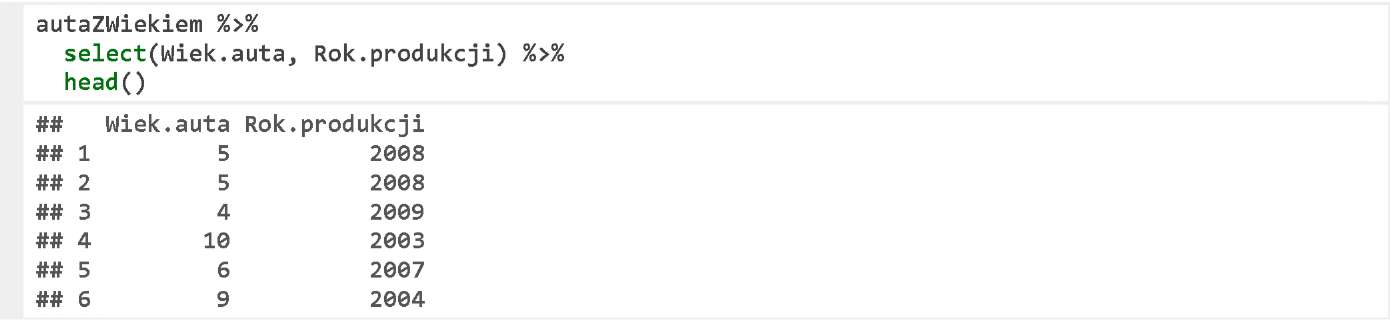


**Computation of One New Variable**

Let’s calculate the age of the cars first. The data had been collected before 2012, so the age of a car is a difference between 2012 plus one and the year of manufacture.



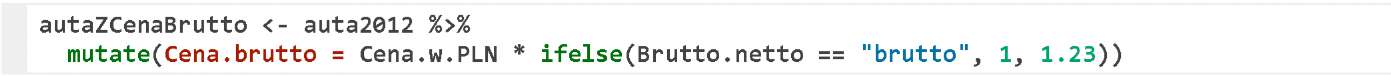
Next, we will display first six rows and two columns (age and year of manufacture).



**Computation of New Variables**

Note that part of the prices is net, and another portion is gross. Let’s convert all prices to the gross prices by adding 23% where the gross price is found.

For this purpose, we will use the function ifelse(), which returns the second or third argument, depending on whether the first argument is true or false.



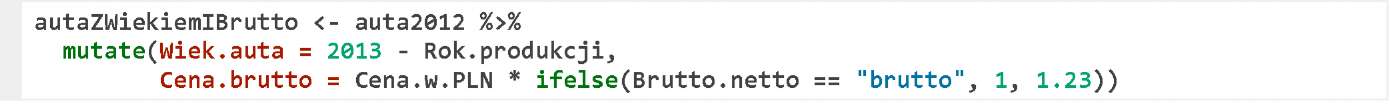
Let’s display the first six rows and the selected three columns.



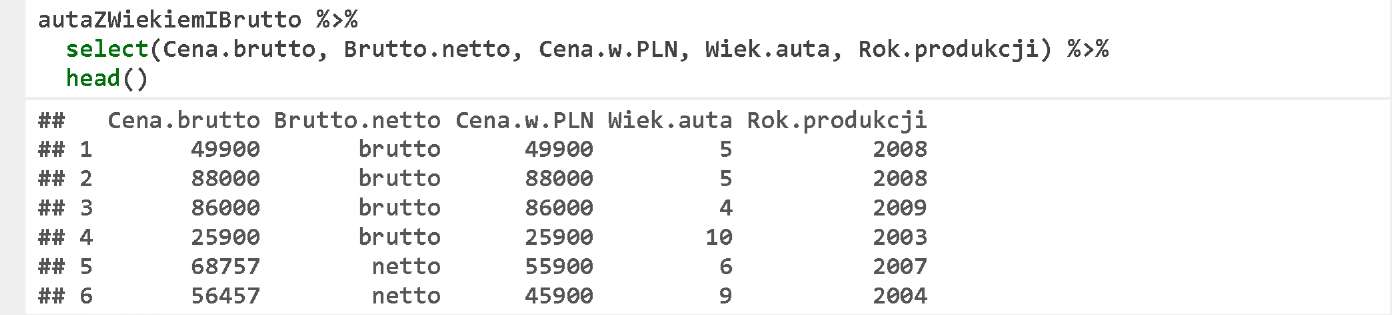
**Computation of New Variables**

In one performance of the function mutate () , several transformations can be specified.

In the following example, we perform both above-presented transformations in one step.



Let’s display the selected columns.



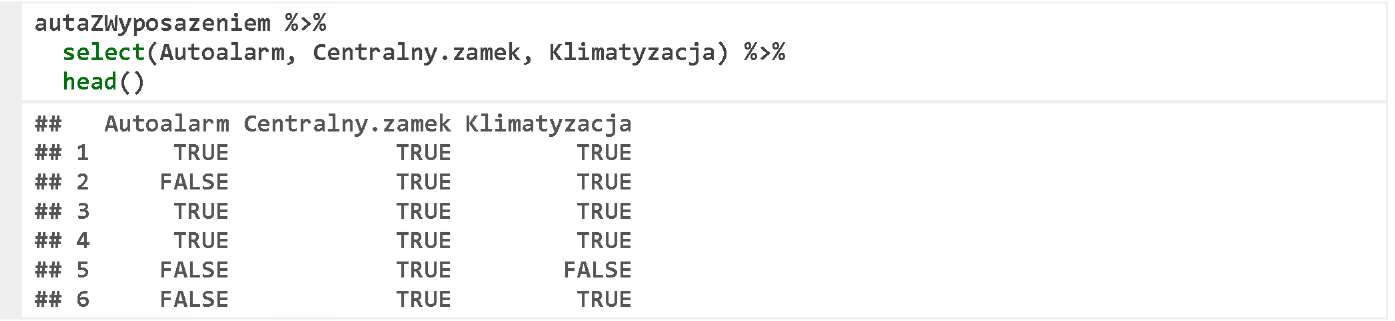
**Air-Conditioning and More**

Let’s use the function mutate () to add the columns that specify whether a specific car features the air-conditioning, central locking, or car alarm.

To check if there is a specific element in the column Wyposażenie.dodatkowe, we will use the function grepl ()



Let’s display the selected columns.



**Exercises**

* Add a new variable to the set of data koty\_ptaki. We know from physics that momentum equals velocity times mass. Compute the maximum momenta for each species and arrange the rows in the descending order.
* Compute the average mileage per year by dividing the mileage by a car age.
* Beside a mechanical horse power, another power unit is the English horse power (HP). One mechanical horse power equals 0.9863 English horse powers. Create new variable, which will displays the car power in English horse powers (BTW: Wikipedia knows a few other interesting power units).

You can find the exemplary answers at https://rawgit.com/pbiecek/MOOC/master/0\_dane/9\_zadania.html.