**Data and Analysis Grouping [dplyr / group\_by]**

pogRomcy danych (Data Masters) Season 1 / Episode 23

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?**

We showed in the previous Episode how to compute the statistics / aggregates on the set of data. This option is even more useful if we compute these aggregates in the groups. With this, we can easily compare such groups later. To define the groups, we will use the function group\_by() of the package dplyr.

In this Episode, we will learn:

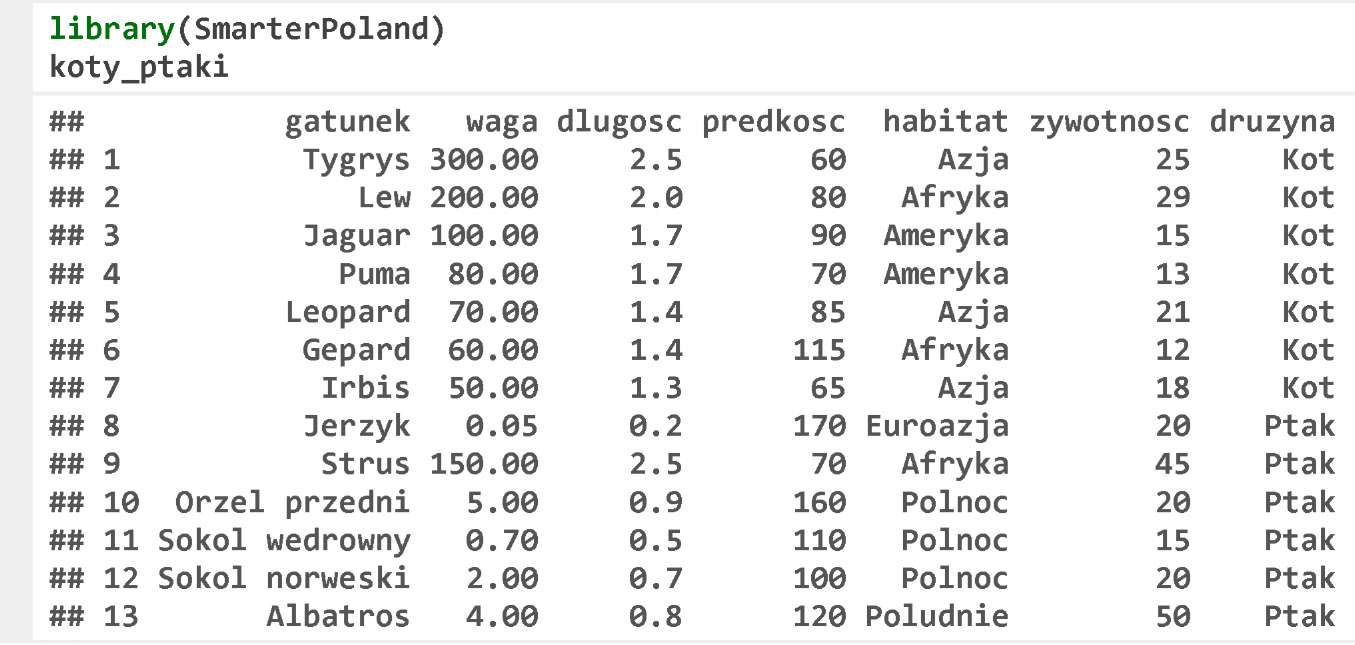
* How to define one or more grouping variables?
* How to calculate the aggregates in the groups?

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 SmarterPoland package.

**Birds Vs Cats**

Let’s start with the example for the small cats and birds set of data. It is small enough to display the result of the operations on the screen.

The variable druzyna is excellently suitable for being a grouping variable.

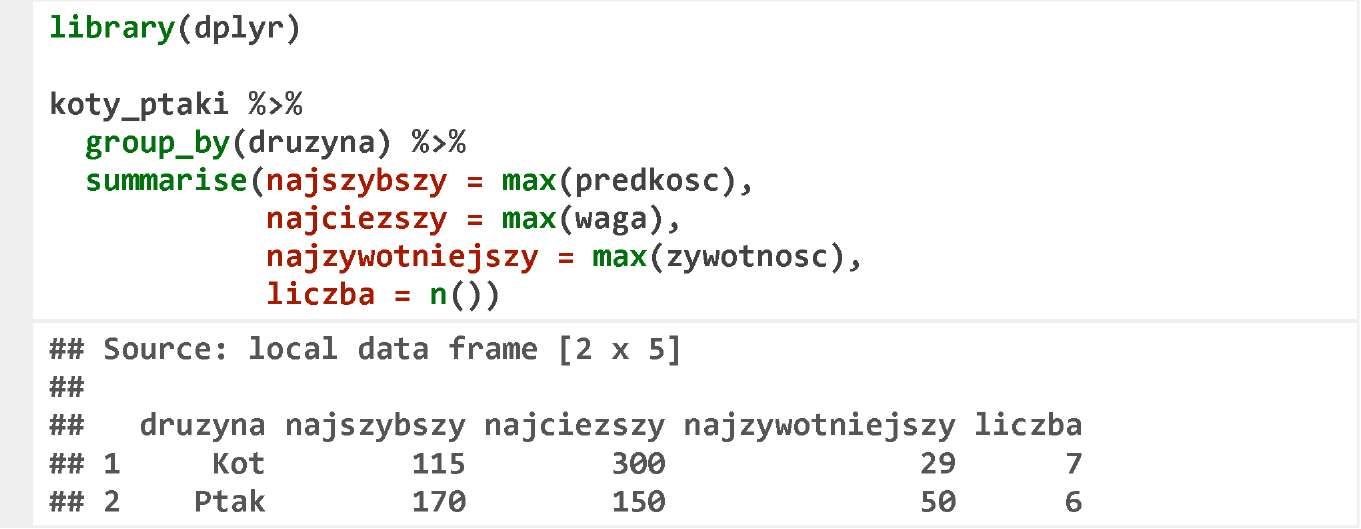


**Birds Vs Cats**

The function group\_by() assumes the dataset to be the first argument, and the grouping variables (one or more) to be the next arguments. The function group\_by() itself does not effect computation of the new values, but only indicates, which variables need to be used for grouping. Thus, the effect of the subsequent functions, such as summarise() will be different.

Let’s see how the use of the function group\_by() to compare the maximum weight, speed, and vitality in the bird and cat group looks like.

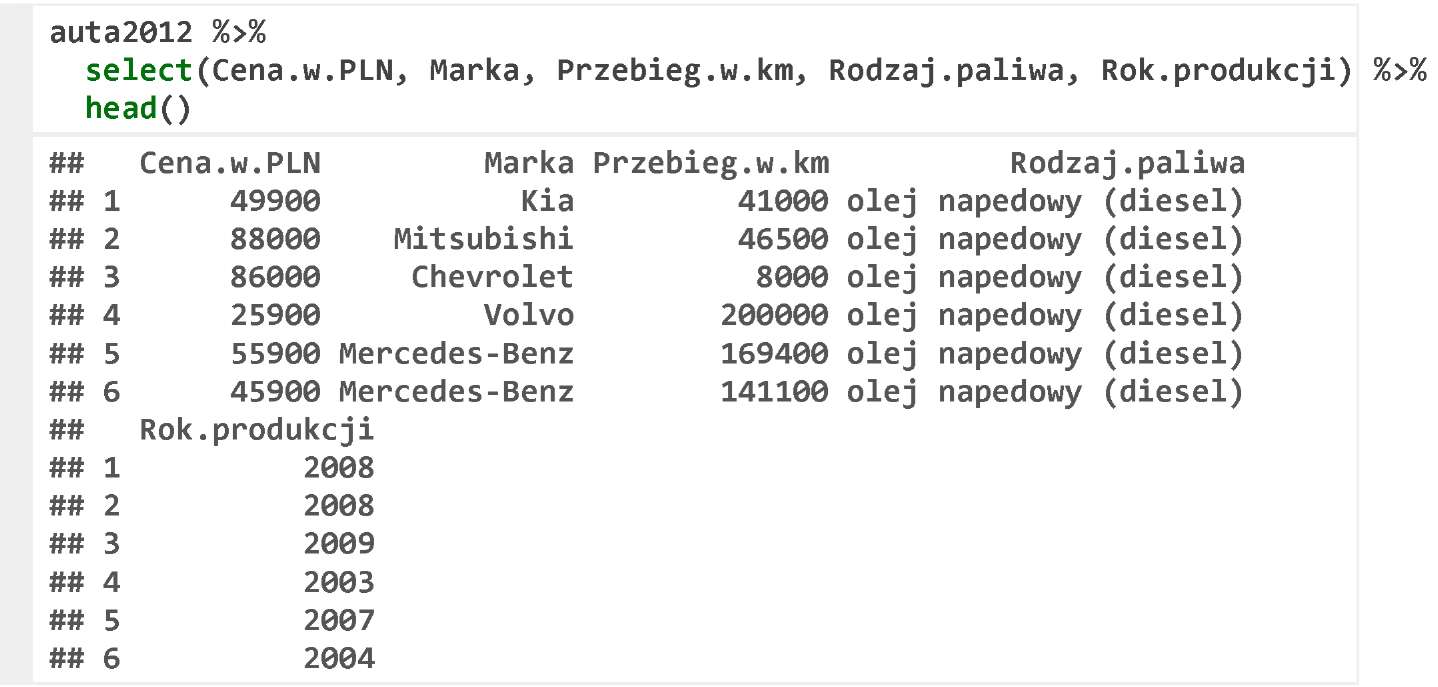
The result is the data frame, the columns of which are first the grouping variables and then the aggregate results in the groups.



**Cars**

In order to present more possibilities of the grouping operation, we will use the set of data auta2012.

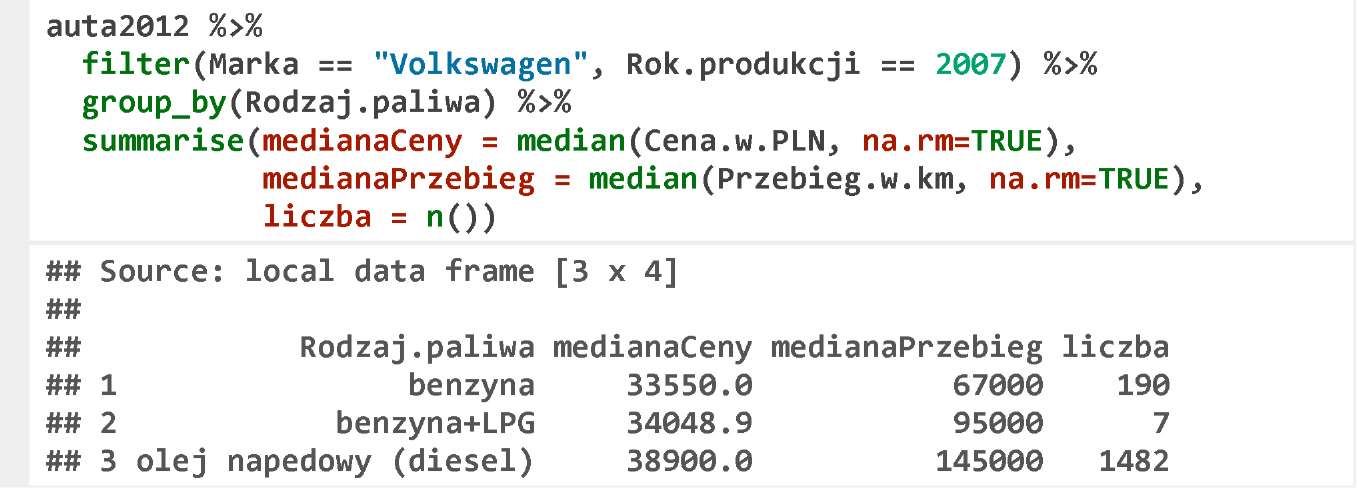
Please find below six first rows and the selected columns of this set of data. You can find the detailed description of this set at https://rawgithub.com/pbiecek/MOOC/master/O\_dane/O\_dane.html.



**Grouping**

We are going to start the examples for the data on the car offer prices with the comparison of the prices and mileages of the cars using different types of fuel.

In order to reasonably compare different types of fuel, we limit ourselves to one car make and one year of manufacture. In the below example, we select only 5-year old Volkswagen, and then for the gasoline-, diesel oil-, or gas-fuelled cars, we compute the median of the price and mileage, and the number of the cars in that group.

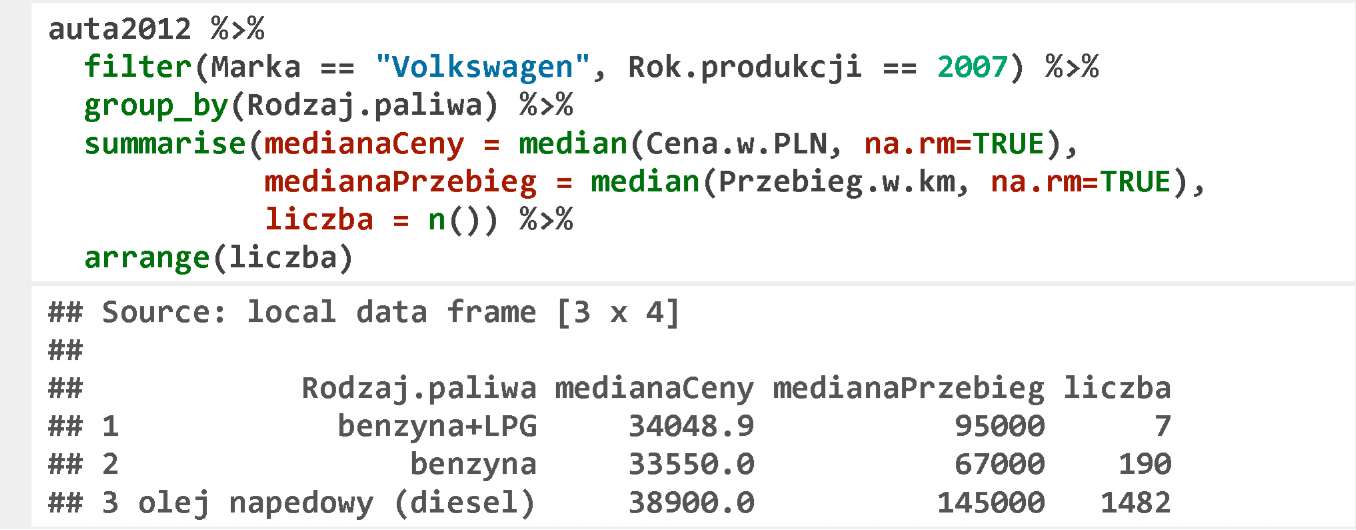


The largest car group includes the diesel oil-fuelled cars. Their mileage is almost twice as high as of the gasoline-fuelled cars (see the medians), and yet they are more expensive by about 15%.

**Sorting**

We can work with the aggregates like with the data frames. We can transfer them further to the functions, which sort, select, or change the variables.

In the below example, we sort the aggregates by the number of observations.

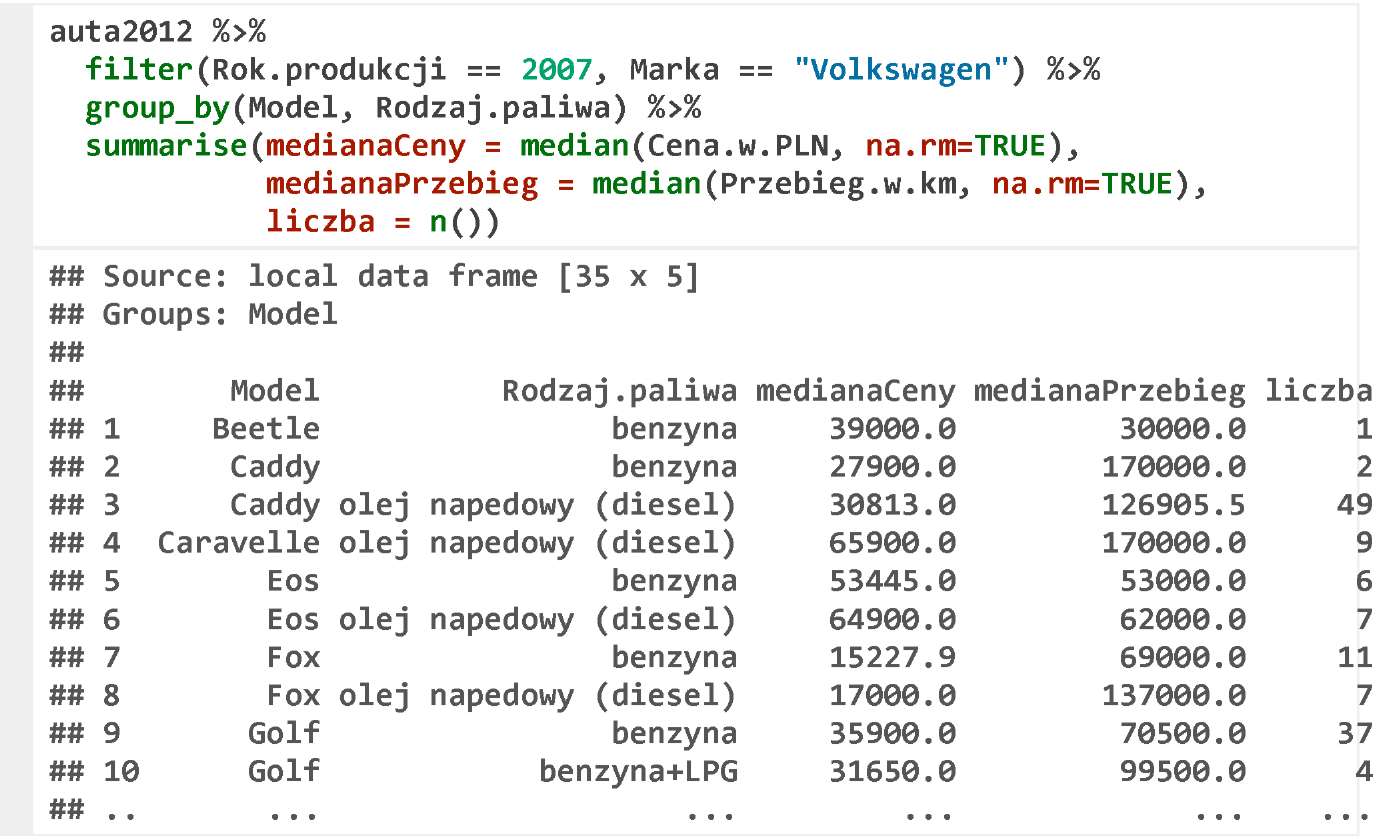


**Grouping by Two Variables**

You can group by several variables, in which case the aggregates are counted in each variable subset.

Let’s see how the mileages and prices would look like, if we additionally divide the cars into the models. In the group of 5-year old Volkswagens, you can find 19 various models.

The below instruction is virtually identical to the previous one, the only difference is that we group by two variables, model and fuel type.

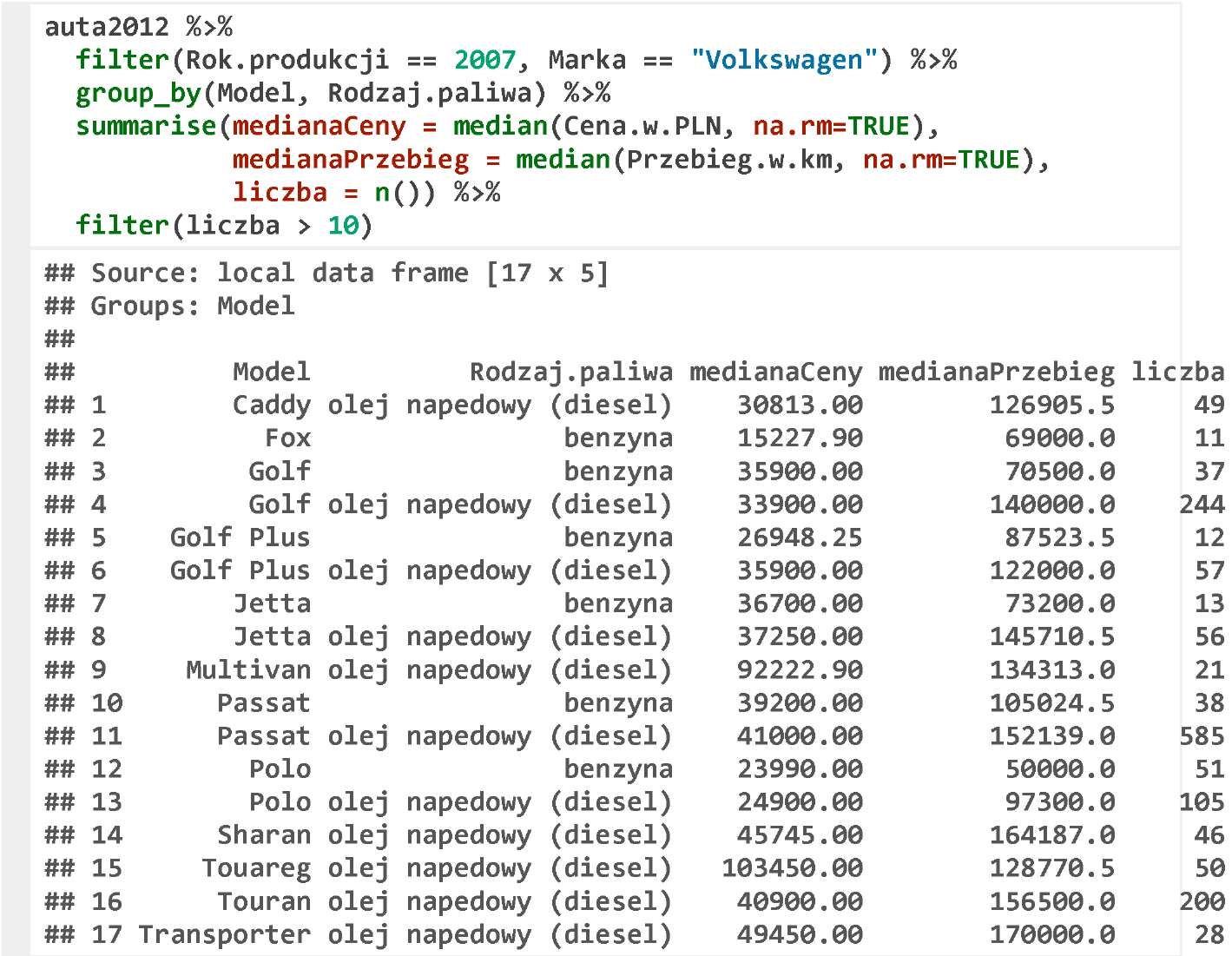


**Grouping and Filtering**

In the previous example, many groups were small, so their results were quite random, and thus difficult to compare.

The application of the function filter() at the end enables to leave only those groups with more than 10 observations.

Please note that in the below example, the function filter() occurs twice, in the first case it filters the rows in the original dataset, and in the second case it filters too small groups.



**Exercises**

* Select 5-year old Volkswagens and check whether the average price depends on a country of the current registration. For this purpose, group by the variable Kraj.aktualnej. rejestracji and compute the mean value in each group.
* Select Peugeots 206 only and calculate the mean price depending on the year of manufacture. Arrange those groups by the year of manufacture

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

**What’s Next?**

The package dplyr has completely changed a manner of data processing in the R. The processing has become easier to describe, 80% of the redundant work can be performed more simply and quicker.

A great supplement to the results of the Episodes on this package will be a graphical 2-page cheat sheet that features the capabilities of this package. This cheat sheet is available at http://www.rstudio.com/wp-content/uploads/2015/01/data-wrangling- cheatsheet1.pdf.

More such cheat sheets are available on the website of RStudio at http://www.rstudio.com/resources/cheatsheets/

A short presentation of the most important functions of the package illustrated with an example of flight data can be found on the workshop website of the dplyr creator at http://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html.