Exploratory Data Analysis and Visualization - MSc AIDA UoM

Professor George Evaggelidis

Academic Year: 2021-2022

The subject of the assignment is data processing/transformation (data wrangling) with the help of the dplyr and tidyr packages from tidyverse. Fill in your answers in the available space immediately after each question.

Starting with dplyr and using the iris dataset:

1. Display entries 6 to 10.
2. Display all versicolor flowers with Sepal.length > 6.
3. Display all flowers that have a Petal.length greater than the average Petal.length.
4. Display all columns except Species.
5. Display columns that start with 'S'.
6. Display variables that contain the string "Len".
7. Rename the Species column to Type.
8. Sort the file by Petal.Length (descending) and then by Petal.Width (ascending).
9. Add two columns with the sums of the lengths and the widths.
10. Calculate the sum of each column.
11. For each species, calculate the min, max, mean, median of Petal.Length.

Continuing with tidyr:

12. Variables: from columns to pairs of type (key, value). Often a variable occupies many columns. For example, in the dataset iris, we may wish to consider Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width as variations of the unique variable measure. Transform iris into iris2 with the following structure:

```
patient    measure  value
setosa     Sepal.Length   5.1
setosa     Sepal.Length   4.9
...
```

13. Variables: From pairs of type (key, value) to columns. Suppose we want to revert iris2 to the original iris. Unfortunately, this is no longer possible! We have lost the information of the combinations of Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width that correspond to the same flower... To avoid this, we should have followed one of the following two approaches:
    (a) First give a unique ID to each flower and then proceed to what was requested in question 12. Therefore, add a column with a unique ID (1,2,3...,150) to each entry of iris before you create iris2.
    (b) Alternatively, number the 50 instances of each type of flower from 1 to 50. In the first scenario, the key is the ID field we added, while in the second scenario, the key is the combination (ID, species). (hint: see what the function rep does).

Alternatively, you can add to iris2 the appropriate numbering to make each original entry unique. Note that in iris2 we have 50 entries with Sepal.Length for setosas, followed by 50 entries with Sepal.Length for versicolors, etc. Since the combination of one ID (1,2,3,...,50) for each type of flower with the type of the flower is unique for each entry, it is enough to add a new column in iris2 with content 12 times the vector 1:50.

In general, for spread or pivot.wider to work, each set of "spread" values must correspond to a unique combination of values of the other columns.

14. Variables: both in pairs of type (key, value) and in columns (completely messy!) Often variables are stored both in rows and in columns... See the following example where the data is completely messy! The final goal is to have entries that are uniquely determined by the patient's name and the visit (i.e., each visit will have a series of measurements).

```
patient <- c("Giorgos", "Giorgos", "Natasha", "Natasha", "Ismini", "Ismini")
measure <- c("high_pressure", "low_pressure", "high_pressure", "low_pressure", "high_pressu
visit1 <- c(170, 100, 135, 85, 120, 80)
visit2 <- c(150, 90, 110, 70, 140, 90)
df <- data.frame(patient, measure, visit1, visit2)
df
```

(a) First, place all the visit values in one column:

```
   patient   measure        visit  value
1  Giorgos   high_pressure visit1  170
2  Giorgos   low_pressure  visit1  100
3  Natasha   high_pressure visit1  135
4  Natasha   low_pressure  visit1  85
5  Ismini    high_pressure visit1  120
6  Ismini    low_pressure  visit1  80
7  Giorgos   high_pressure visit2  150
8  Giorgos   low_pressure  visit2  90
9  Natasha   high_pressure visit2  110
10 Natasha   low_pressure  visit2  70
11 Ismini    high_pressure visit2  140
12 Ismini    low_pressure  visit2  90
```

(b) Now, create separate columns for high and low pressure.

```
  patient  visit  high_pressure  low_pressure
1 Giorgos visit1  170            100
2 Giorgos visit2  150            90
3 Ismini  visit1  120            80
4 Ismini  visit2  140            90
5 Natasha visit1  135            85
6 Natasha visit2  110            70
```