Data loading

Alla Tambovtseva

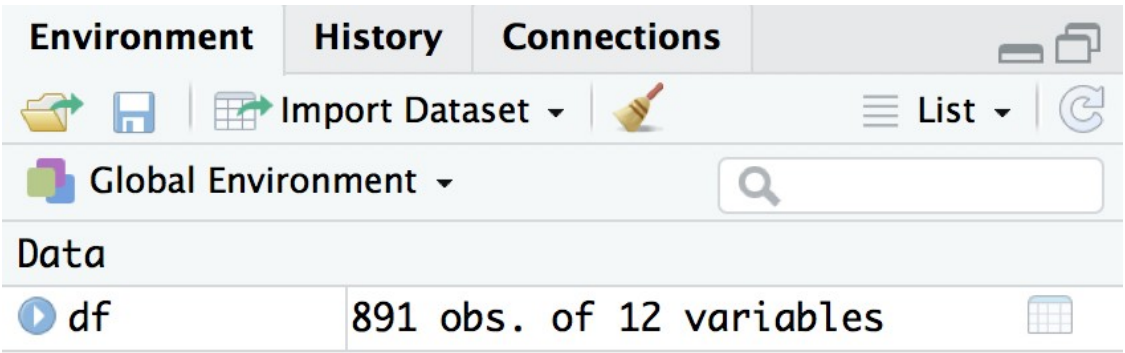
Data loading

Via a link

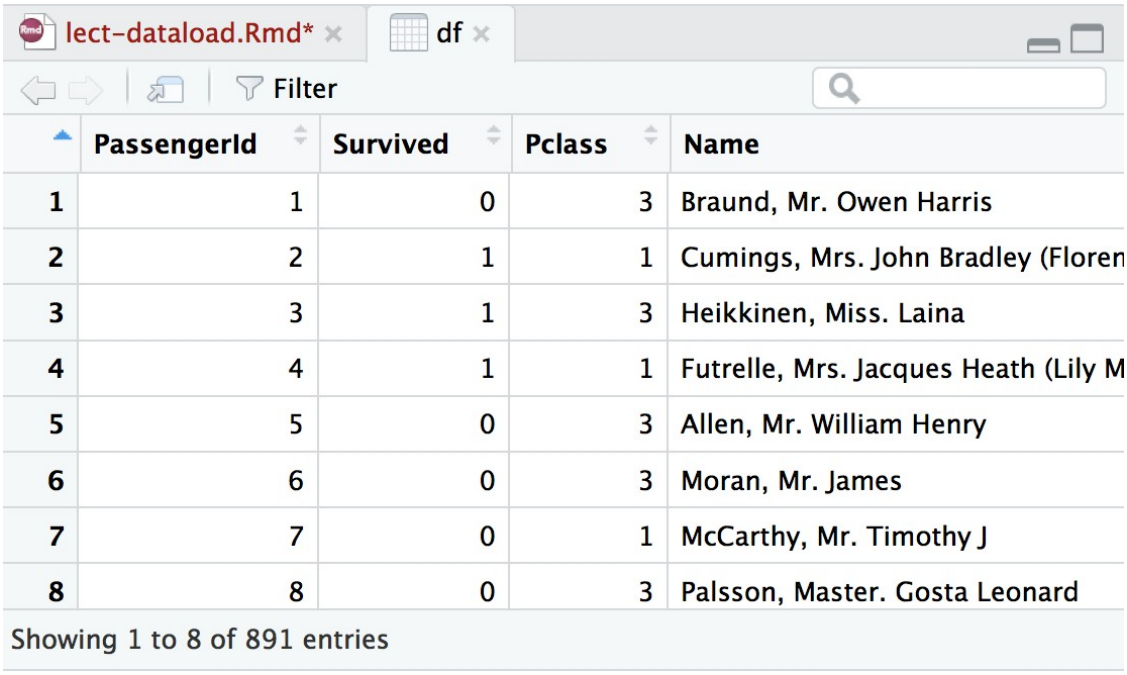
Let's start from a very basic case and load a data set from a csv-file via a link. The format *csv* stands for *comma separated values* as in this file columns are separated with commas by default. The columns separator can be different, though, so we might need to add an argument specifying it. We will discuss these cases later. To load data from a file we use functions starting from `read.` and ending with its extention.

```
df <-
read.csv("http://math-info.hse.ru/f/2018-19/pep/r/Titanic.csv") #
don't forget about quotes "" or ''
```

While loading data via a link, make sure your Internet connection is good. If everything is ok, you will see no error messages in the console and the variable `df` with the dataset will appear in the tab *Environment*.



Here we see general information about this data set, its dimensions: the number of observations and the number of variables. If we click on the `df` (exactly on the name `df`, not on the blue circle on the left), we will see a table in a separate tab (so-called *View mode*):



lect-dataload.Rmd* × df ×

Filter

	PassengerId	Survived	Pclass	Name
1	1	0	3	Braund, Mr. Owen Harris
2	2	1	1	Cumings, Mrs. John Bradley (Floren
3	3	1	3	Heikkinen, Miss. Laina
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily M
5	5	0	3	Allen, Mr. William Henry
6	6	0	3	Moran, Mr. James
7	7	0	1	McCarthy, Mr. Timothy J
8	8	0	3	Palsson, Master. Gosta Leonard

Showing 1 to 8 of 891 entries

In this mode we can view the table in a convenient way. We can scroll it down/right and enlarge clicking on the sign in the top right corner.

Before starting to describe a table we have to learn how to load data from a file saved in a local folder.

From a local file

Firstly, let's look how to know which folder on our computer is a working one. A working directory is the folder from which RStudio launches. By default R sees only files that are stored in this folder. To get the path to the working directory, we need the function `getwd()`:

```
getwd() # wd - from working directory
```

```
## [1] "/Users/allat/Dropbox/вшэ - работа/кафедра высшей
математики/2018-2019/DASS/R"
```

In my case RStudio launches from the folder *Desktop* that is in the folder *allat*. It means that I can place my csv-file to *Desktop*, and I will be able to indicate its name with the extension in `read.csv()` and load it without difficulties:

```
df <- read.csv("Titanic.csv") # file is in Desktop
```

If a file is not stored in the working directory, typing its name like this makes no sense, it will certainly result in the error `cannot open file: No such file or directory`. So as to overcome this problem (if we do not want to move our file to the working directory), we can write the full path to the file:

```
df <- read.csv("/Users/allat/Downloads/Titanic.csv")
```

We can get this path by clicking the right mouse button on this file and choosing *Properties* (Свойства if you work in class at the computer with cyrillics). In properties there is always a line with the file location (Расположение файла on computers with cyrillics). We should copy the location and paste it into the braces of `read.csv()`. Please, mind the slashes. R does not work with back slashes ('') commonly used on Windows, it accepts only direct ones (`/`). Change all the slashes in a path or add them if something went wrong (on Mac paths might be copied without slashes or with dots instead). Make sure your path ends with the file name and its correct extension. The function `read.csv()` opens namely files, not folders!

Changing a working directory

One more way to access files is to change the path to the working directory. Such an approach can be useful when we have a lot of files to work with and we do not plan to copy/move them to the current working directory (and, of course, we are very lazy to write full paths all the time). So as to change the working directory, we need `setwd()` and a path to the directory desired:

```
setwd("/Users/allat/Downloads")
```

Before setting a new path, check whether this new directory exists. If not, R will not create it authomatically and will return an error.

Now let's proceed to more interesting things.

Data description

Now we have the table `df` with the information on "Titanic" passengers that contains data on people's characteristics and the indication whether a passanger survived in this notorious shipwreck.

See a detailed information in the codebook for the data set available by this link. It is a good example of a codebook since there is detailed description of all variables with clear explanations of their values.

Now let's look at the structure of `df`:

```
str(df)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..:
109 191 358 277 16 559 520 629 417 581 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2
2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
```

```
##  $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524
597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1
57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3
4 4 4 2 ...
```

It returns the following information: a number of observations and variables, variable names and their types (numeric, integer, factor, character) as well as several values of each variable. Often character variables are treated as factor ones due to default R settings, so R assign some numeric values (levels) to texts saved in columns.

Now we will look at more substantial things and ask R for descriptive statistics. It can be done with `summary()` command:

```
summary(df)
```

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##                                     Name         Sex
Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.
: 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st
Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1
Median :28.00
##  Abelson, Mr. Samuel                   :  1               Mean
:29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1               3rd
Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1               Max.
:80.00
##  (Other)                               :885               NA's
:177
##      SibSp           Parch             Ticket         Fare

##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :
0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:
7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median :
```

```
14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   :
32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.:
31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6
Max.   :512.33
##                                   (Other) :852

##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

For numeric variables this function returns standard descriptive statistics: minimum (`Min.`) and maximum (`Max.`), lower (`1st Qu.`) and upper (`3rd Qu.`) quartiles, average (`Mean`) and median (`Median`).

For non-numeric (character or factor) variables it returns counts, absolute frequences showing how many times every unique value occurs in a column.

For this table we can, for instance, conclude that on average passengers of "Titanic" were not very old (29.7 is the mean age), the oldest passanger was 80 years old, there were people who paid nothing for the ticket (minimum fare is 0), most people embarked in Southampton (644 vs 168 and 77) and there were more males than females (577 vs 314).

So as to get a description of a particular variable, we can access it using the dollar sign `$`:

```
summary(df$Age) # take Age from df
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

Note that R also counts missing values coded as `NA` (from *Not Applicable*). In programming there are two "empty" types that refer to missing values: `NA` and `NaN` (*Not A Number*). Often they are interchangeable, but `NaN` might stand for non-empty values like *infinity* (∞) that is not a number and not an empty cell.