Урок 1. Чтение и запись файлов

Урок 1. Чтение и запись файлов

Всем привет! В прошлых модулях мы познакомились с основами программирования в R, а в этом модуле мы перейдем непосредственно к данным. И начнем с загрузки файлов с данными в R.

Рассмотрим два популярных формата: файл в формате `.csv` (от *comma separated values*, значения, разделенные запятыми) и файл Excel.

С форматом csv все просто — для его загрузки не требуется никаких вспомогательных библиотек. Воспользуемся функцией `read.csv()` (вообще многие функции для чтения файлов в R так выглядят, *read* + название формата) и функцией `file.choose()`, которая позволит выбрать файл с компьютера в отдельном окне:

```
dat <- read.csv(file.choose())
```

Выберем файл `firms.csv`, с которым мы будем активно работать в следующем уроке. Все файлы из этого занятия можно найти в материалах к уроку.

Посмотрим, как выглядит загруженный датафрейм:

```
View(dat) # смотрим на dat
```

Вспомним, как загружать файл, прописывая путь к нему. Зайдем в свойства файла и скопируем его расположение:

```
dat <- read.csv("/Users/allat/Desktop/firms.csv")
```

Если файл лежит в рабочей папке, то достаточно указать просто его название с расширением. А узнать, какая папка является рабочей, можно так:

```
getwd()  # get current working directory
```

```
## [1] "/Users/allat/Desktop"
```

По умолчанию в качестве разделителя столбцов используется запятая. Как быть, если разделитель другой? Например, точка с запятой? Можно воспользоваться опцией sep (от *separator* — разделитель) и выставить нужный символ вручную. Сравним два случая.

Случай 1 (неверный разделитель)

```
# все склеилось в один столбец
dat1 <- read.csv("/Users/allat/Desktop/example1.csv")
```

Случай 2 (верный разделитель)

```
# все хорошо
dat2 <- read.csv("/Users/allat/Desktop/example1.csv", sep=";")
```

Также при загрузке файлов в R можно столкнуться с еще одной проблемой — выбором десятичного разделителя. Как мы уже убедились, в R дробная часть числа отделяется точкой, но в файлах можно встретить и запятую. Чтобы избежать замены запятой на точку в каждом столбце таблицы и последующего приведения типов, можно указать нужный разделитель в опции `dec` (от *decimal*):

Опять сравним два случая.

Случай 1 (неверный разделитель)

```
# запятые - столбцы с такими числами
# воспринимаются как текстовые
dat3 <- read.csv("/Users/allat/Desktop/example2.csv", sep = ";")
```

Случай 2 (верный разделитель)

```
# все хорошо
dat4 <- read.csv("/Users/allat/Desktop/example2.csv", sep = ";", dec=",")
```

Теперь перейдем к файлу Excel. В R есть много библиотек, позволяющих читать файлы `.xls` и `.xlsx`, однако многие из них задействуют в работе библиотеку `rJava`, которая требует наличия актуальной версии Java (хотя и с новой версией бывают проблемы). Поэтому воспользуемся библиотекой `readxl`, попроще.

Сначала установим эту библиотеку:

```
install.packages("readxl")
```

Библиотеки в R устанавливаются один раз, при каждом запуске RStudio ставить их заново не нужно.

Теперь обратимся к этой библиотеке — загрузим ее для работы (вот это уже нужно делать при каждом запуске RStudio), чтобы R понимал, откуда брать функции.

```
library(readxl)
```

Загрузим файл Excel и сохраним его содержимое в датафрейм:

```
ex.dat <- read_excel("/Users/allat/Desktop/example.xlsx")
```

Теперь самое время приступить к практическому заданию и потренироваться загружать файлы в R.