Лабораторная работа № 1: Загрузка данных в R

### 1. Цель работы

* Приобрести основные навыки работы с данными в R
* Освоить наиболее распространенные способы загрузки данных в рабочую среду R

### 2. Задачи

* Углубить и закрепить знания по основным типам данных и научить загрузить и изучить свойства данных.
* Совершенствовать навыки самостоятельной работы.
* …

### 3. План выполнения работы

### 3.1 Загрузка различных данных

* Классический вариант: загрузка данных форматов .txt или .csv в среду R: для этого воспользуемся командами:

#data <- read.csv("путь к файлу/имя файла.csv", sep = ";")  
#data <- read.table("путь к файлу/имя файла.txt",sep = "\t")

* Загрузка файлов в форматах .xls или .xlsx (форматы программы Excel) : Используя пакет “readxl”:

library(readxl)

M3C <- read\_excel("GitHub/LAB/LAB 1/data/M3C.xls")

View(M3C)

* Загрузка данных из других пакетов R: к данным доступа из определенного пакета используйте параметр пакета, например:

#data(package="FORA")  
#data(FORAYear, package="FORA")

* Загрузка данных из интернета:

#data <- read.csv(“file = "http://somesite.net/YourData.csv")

здесь , file служит для указания пути к загруженному файлу, например file = “C:/Temp/MyData.dat”). В данном примере указана полную URL-ссылка на файл, который предполагается загрузить из Internet.

### 3.2 Свойства данных

Предлагаем, что мы уже загрузили данные в виде data frame c именем data1:

data1 <- read.csv("FORA.csv", sep = ";")

Для получения свойства загруженных данных воспользуемся следующими командами:

* head(x) — посмотреть начальные строки объекта
* tail(x) — посмотреть последние строки объекта
* dput(x) — получить представление объекта в R-синтаксисе
* summary(object) — общая информация об объекте
* str(object) — внутренняя структура объекта
* dim(x) — размерности объекта
* dimnames(x) — имена размерностей объекта
* names(x) — имена объекта
* length(x) — число элементов

Например:

head(data1)

## X.series\_id.category.actual.method.forecast.horizon.timestamp  
## 1 1,Y1,MICRO,5379.75,NAIVE2,4936.99,1,1989  
## 2 2,Y1,MICRO,6158.68,NAIVE2,4936.99,2,1990  
## 3 3,Y1,MICRO,6876.58,NAIVE2,4936.99,3,1991  
## 4 4,Y1,MICRO,7851.91,NAIVE2,4936.99,4,1992  
## 5 5,Y1,MICRO,8407.84,NAIVE2,4936.99,5,1993  
## 6 6,Y1,MICRO,9156.01,NAIVE2,4936.99,6,1994

summary(data1)

## X.series\_id.category.actual.method.forecast.horizon.timestamp  
## 1,Y1,MICRO,5379.75,NAIVE2,4936.99,1,1989 : 1   
## 10,Y1,MICRO,7851.91,SINGLE,4936.99,4,1992 : 1   
## 100,Y1,MICRO,7851.91,ForecastPro,7133.43,4,1992: 1   
## 1000,Y8,MICRO,7294.08,Auto-ANN,8288.37,4,1992 : 1   
## 10000,Y76,MICRO,6850,ForecastPro,5495.22,4,1992: 1   
## 10001,Y76,MICRO,8368,ForecastPro,5840.72,5,1993: 1   
## (Other) :85134

str(data1)

## 'data.frame': 85140 obs. of 1 variable:  
## $ X.series\_id.category.actual.method.forecast.horizon.timestamp: Factor w/ 85140 levels "1,Y1,MICRO,5379.75,NAIVE2,4936.99,1,1989",..: 1 11112 22223 33334 44445 55556 66667 77778 84030 2 ...

names(data1)

## [1] "X.series\_id.category.actual.method.forecast.horizon.timestamp"

dim(data1)

## [1] 85140 1

head(data1)

## X.series\_id.category.actual.method.forecast.horizon.timestamp  
## 1 1,Y1,MICRO,5379.75,NAIVE2,4936.99,1,1989  
## 2 2,Y1,MICRO,6158.68,NAIVE2,4936.99,2,1990  
## 3 3,Y1,MICRO,6876.58,NAIVE2,4936.99,3,1991  
## 4 4,Y1,MICRO,7851.91,NAIVE2,4936.99,4,1992  
## 5 5,Y1,MICRO,8407.84,NAIVE2,4936.99,5,1993  
## 6 6,Y1,MICRO,9156.01,NAIVE2,4936.99,6,1994

### 3.3 Работа с Data Frame

Data frame - это гибридный тип представления, одномерный список из векторов одинаковой длины. Таким образом, каждая data frame - это список колонок, причём внутри одной колонки все данные должны быть одного типа. Проиллюстрируем это на примере:

name <- c("Kolia", "Jenhia", "Sasha", "Kastia", "Vasia", " Jora")  
weight <- c(60, 68, 71, 87, 67, 93)  
height <- c(174, 168, 178, 188, 165, 172)  
size <- c("L", "S", "XL", "XXL", "S", "M")  
sex <- c("male", "female", "male", "male", "female", "male")  
data <- data.frame(name, weight, height, size, sex)  
data

## name weight height size sex  
## 1 Kolia 60 174 L male  
## 2 Jenhia 68 168 S female  
## 3 Sasha 71 178 XL male  
## 4 Kastia 87 188 XXL male  
## 5 Vasia 67 165 S female  
## 6 Jora 93 172 M male

Очень часто бывает нужно отобрать несколько конкретных колонок. Это можно сделать разными способами (исключаем столбец name и отобрать столбец weight):

data[,-1]

## weight height size sex  
## 1 60 174 L male  
## 2 68 168 S female  
## 3 71 178 XL male  
## 4 87 188 XXL male  
## 5 67 165 S female  
## 6 93 172 M male

data[,2:4]

## weight height size  
## 1 60 174 L  
## 2 68 168 S  
## 3 71 178 XL  
## 4 87 188 XXL  
## 5 67 165 S  
## 6 93 172 M

data$weight

## [1] 60 68 71 87 67 93

Отобрать из нашей таблицы только данные, относящиеся к женщинам? Вот один из способов:

data[data$sex == "female",]

## name weight height size sex  
## 2 Jenhia 68 168 S female  
## 5 Vasia 67 165 S female

Или отобрать только нужные данные с помощью команды subset:

data1 <- subset(data, sex == "male")  
data1

## name weight height size sex  
## 1 Kolia 60 174 L male  
## 3 Sasha 71 178 XL male  
## 4 Kastia 87 188 XXL male  
## 6 Jora 93 172 M male

data2 <- subset(data, height >= 168 & sex == "male")  
data2

## name weight height size sex  
## 1 Kolia 60 174 L male  
## 3 Sasha 71 178 XL male  
## 4 Kastia 87 188 XXL male  
## 6 Jora 93 172 M male

Более сложным случаем селекции является сортировка таблиц данных. Для сортировки таблиц данных достаточно применить команду sort(). Например, если мы хотим отсортировать наши данные сначала по полу, а потом по росту, приходится выполнить следующую команду:

data[order(data$sex, data$height), ]

## name weight height size sex  
## 5 Vasia 67 165 S female  
## 2 Jenhia 68 168 S female  
## 6 Jora 93 172 M male  
## 1 Kolia 60 174 L male  
## 3 Sasha 71 178 XL male  
## 4 Kastia 87 188 XXL male