Importing Data Into R

Hamid Abdulsalam

Packages

Download the following Packages

- library(tidyverse)
- library(readxl)
- library(DBI)
- library(RMySQL)
- library(haven)

Data

We have a series of data files that we will be working with.

- auto: data on automobiles
- personality: data on Big Five personality traits for 434 persons
- birth1.sas7bdat: data on birth weight
- fertz: data on fertilizer
- who_suicide_statistics: data on suicide
- potatoes: impact of storage and cooking on potatoes' flavor
- Employees: data on employees of certain company

Importing data using Base R functions

read.table()

Generally used to load data in different format into R

- **file:** Path to the file containing the data file.
- **sep:** field separator character. ‘’ is used for tab-delimited file.
- **header:** logical value. If TRUE, read.table() assumes that your file has a header row, so row 1 is the name of each column. If that’s not the case, you can add the argument header = FALSE.
- **dec:** the character used in the file for decimal points

read.table()

Let's load CSV file using read.table()

```
data_1 <- read.table("data/who_1.csv",
                     sep = ",", header=TRUE)
data_1[1:4,1:4]
```

```
##   country year    sex         age
## 1 Albania 1985 female 15-24 years
## 2 Albania 1985 female 25-34 years
## 3 Albania 1985 female 35-54 years
## 4 Albania 1985 female  5-14 years
```

read.table()

Let's Load a Comma Seperated TXT file using read.table()

```
auto_1 <- read.table("data/auto.txt",
                     sep = ",", header=F)
auto_1[1:4, 1:6]
```

```
##   V1  V2           V3  V4  V5   V6
## 1  3   ? alfa-romero gas std  two
## 2  3   ? alfa-romero gas std  two
## 3  1   ? alfa-romero gas std  two
## 4  2 164         audi gas std four
```

read.table()

Let's load a TAB delimited files (txt) using read.table()

```
pot_1 <- read.table("data/potatoes.txt",
                    sep = "\t", header=F)
pot_1[1:4,]
```

```
##   V1 V2 V3 V4 V5  V6  V7  V8
## 1  1  1  1  1  1 2.9 3.2 3.0
## 2  1  1  1  1  2 2.3 2.5 2.6
## 3  1  1  1  1  3 2.5 2.8 2.8
## 4  1  1  1  1  4 2.1 2.9 2.4
```

read.table()

Let's load a Semicolon Separated files using read.table()

```
who_2 <- read.table("data/who_2_semi.csv",
                    sep = ";", header=TRUE)
who_2[1:4,1:4]
```

```
##   country year    sex         age
## 1 Albania 1985 female 15-24 years
## 2 Albania 1985 female 25-34 years
## 3 Albania 1985 female 35-54 years
## 4 Albania 1985 female  5-14 years
```

read.csv

We can also use read.csv function to import Comma Separated CSV files into R.

```
who_3 <- read.csv("data/who_1.csv", header=TRUE)
who_3[1:4,1:4]
```

```
##   country year    sex         age
## 1 Albania 1985 female 15-24 years
## 2 Albania 1985 female 25-34 years
## 3 Albania 1985 female 35-54 years
## 4 Albania 1985 female  5-14 years
```

read.csv2

We can specifically use **read.csv2** to import Semicolon Separated Files into R.

```
who_4 <- read.csv2("data/who_2_semi.csv", header=TRUE)
who_4[1:4, 1:4]
```

```
##   country year    sex          age
## 1 Albania 1985 female 15-24 years
## 2 Albania 1985 female 25-34 years
## 3 Albania 1985 female 35-54 years
## 4 Albania 1985 female  5-14 years
```

read.delim()

For reading Tab Delimited Files in R

```
pot_n<-read.delim("data/potatoes.txt", header=F)
pot_n[1:4, ]
```

```
##   V1 V2 V3 V4 V5  V6  V7  V8
## 1  1  1  1  1  1 2.9 3.2 3.0
## 2  1  1  1  1  2 2.3 2.5 2.6
## 3  1  1  1  1  3 2.5 2.8 2.8
## 4  1  1  1  1  4 2.1 2.9 2.4
```

readr Package

- In terms of speed, readr is ~10x faster than base R functions (read.table,read.csv,read.csv2).
- By default , strings are untouched and common date/time formats are automatically passed.

readr Package

- read_csv(): comma delimited files
- read_csv2(): semicolon separated files
- read_tsv(): tab delimited files
- read_delim(): files with any delimiter

Reading Excel Files

We can always use the readxl package to get data out of Excel and into R. The readxl package supports both .xls format and the modern xml-based .xlsx format.

To import excel sheet into R, we use the function read_excel() and specify the sheet number in the arguments.

read_excel()

```
library(readxl)
sht1 <- read_excel("data/Employees.xlsx", sheet = 1)
sht2 <- read_excel("data/Employees.xlsx", sheet = 2)
sht1[1:4, 1:4]
```

```
## # A tibble: 4 x 4
##   EmployeeID Last_Name First_Name Gender
##        <dbl> <chr>     <chr>      <chr>
## 1        120 Collins   Barnabas   M
## 2        121 Kenobi    Obi-wan    <NA>
## 3        123 Bouchard  Angelique  F
## 4        124 White     Cassandra  F
```

Importing Data From Databases

Reading From Database

To Load data from a database you first have to create a connection to such database. The DBI package is used to connect to the SQL Server while the RMySQL package will be used to perform SQL queries within R.

The function **dbConnect()** creates the connection between R session and the SQL database.For hosted SQL DB , we need to specify the following arguments in dbConnect(): dbname, host, port, user and password.

Establish a connection

To extract data from database in a remote server, we need to first establish the connection to the server in R.

```
library(DBI)
library(RMySQL)
host <- "courses.csrrinzqubik.us-east-1.rds.amazonaws.com"
connect <- dbConnect(RMySQL::MySQL(), dbname = "tweater",
host = host, port = 3306, user = "student", password =
"datacamp")
```

List the database tables

We can use **dbListTables()** to see what tables in the database

```
tables <- dbListTables(connect)
tables
```

```
## [1] "comments" "tweats"   "users"
```

Import data from tables

We can use the **dbReadTable()** function to read data from the database tables.

```
users <- dbReadTable(connect, "users")
users
```

```
##   id      name     login
## 1  1 elisabeth  elismith
## 2  2      mike     mikey
## 3  3      thea   teatime
## 4  4    thomas tomatotom
## 5  5    oliver olivander
## 6  6      kate  katebenn
## 7  7    anjali    lianja
```

Importing data from the web

We can use **read.csv()** to directly import csv files from the web.

```
house <-read.csv("https://factual.ng/course/house.csv",
header = T)
house[1:4,1:4]
```

```
##      MLS.       Location  Price Bedrooms
## 1 132842 Arroyo Grande 795000        3
## 2 134364   Paso Robles 399000        4
## 3 135141   Paso Robles 545000        4
## 4 135712     Morro Bay 909000        4
```

Importing data from other statistical software

To import data from other statistical software such as Stata, SPSS, Sas. We use the package called haven.

- SAS: read_sas()
- STATA: read_dta()
- SPSS: read_sav()

SAS Data File

The **read_sas()** function in haven package can be used to read SAS Data files into R.

```
library(haven)
birth <- read_sas("data/birth1.sas7bdat")
birth[1:4,1:4]
```

```
## # A tibble: 4 x 4
##   Weight Black Married   Boy
##    <dbl> <dbl>   <dbl> <dbl>
## 1   4111     0       1     1
## 2   3997     0       1     0
## 3   3572     0       1     1
## 4   1956     0       1     1
```

STATA Data File

The **read_stata()** function in haven package can be used to read STATA Data files into R.

```
alcohol <- read_dta("data/alcohol.dta")
alcohol[1:4,]
```

```
## # A tibble: 4 x 4
##   adults  kids income consume
##    <dbl> <dbl>  <dbl>   <dbl>
## 1      2     2    758       1
## 2      2     3   1785       1
## 3      3     0   1200       1
## 4      1     0    545       1
```

read_sav()

The **read_sav()** function in haven package can be used to read SPSS Data files into R.

```
pers <- read_sav("data/personality.sav")
pers[1:4,]
```

```
## # A tibble: 4 x 4
##   Neurotic Extroversion Agreeableness Conscientiousness
##      <dbl>        <dbl>          <dbl>             <dbl>
## 1       39           38             31                12
## 2        6           38             27                12
## 3       17           39             32                13
## 4       28           35             39                13
```

The End