5. Recopilación de datos

Índice

1. Operaciones con ficheros y carpetas

Algunas funciones útiles para manejo de carpetas y ficheros en R:

- Obtener la carpeta actual de trabajo:

```
getwd()
```

- Cambiar la carpeta actual de trabajo:

```
setwd("./data")
```

- Listar ficheros:

```
list.files(path = ".")
list.files(full.name = TRUE)
list.files(pattern = ".csv", recursive = TRUE)
```

- Comprobar si existe un fichero:

```
file.exists("houses_in.txt")
```

- Crear una carpeta:

```
if (!dir.exists("backups")) {
  dir.create("backups")
} else {
  "Folder exists already"
}
```

- Crear un fichero vacío:

```
file.create("newdata.csv")
```

- Copiar un fichero:

```
file.copy("houses_in.txt", "./backups/houses_in.txt")
```

- Obtener información de un fichero:

```
file.info("houses_in.txt")
```

- Borrar un fichero o una carpeta:

```
unlink("newdata.csv")
file.remove("newdata.csv")

unlink("backups", recursive = TRUE)
```

- Obtener el nombre base o la ubicación de un fichero:

```
basename("houses_in.txt")
dirname("houses_in.txt")
```

- Obtener la extensión de un fichero:

```
library(tools)
file_ext("houses_in.txt")
```

2. Lectura y escritura de datos

R ofrece algunas funciones integradas para trabajar con ficheros de texto:

- Lectura a un vector:

```
housesVector <- scan("houses_in.txt", what = "character")
housesVector <- scan("houses_in.csv", what = "character")
```

- Lectura a una lista:

```
housesList1 <- scan("houses_in.txt", what = list("", "", "", "", "",
""))
housesList2 <- scan("houses_in.csv", what = list(0.0, 0.0, 0, 0, 0.0,
""), skip = 1)
```

- Lectura a una matriz:

```
housesMatrix <- matrix(scan("houses_in.txt", "", skip = 1), nrow=5,
ncol=6, byrow=TRUE)
```

- Lectura a un data frame:

```
housesDf <- read.table("houses_in.txt", header=TRUE)
housesDf <- read.csv("houses_in.csv", header=TRUE)
```

- Escritura a un fichero de texto

```
write.table(housesDf, file = "houses_out.txt")
write.table(housesDf, file = "houses_out.csv", sep=',')
```

Otra forma de cargar datos es introducirlos directamente por consola:

```
userInputData <- scan("")
```

O cargar los datasets con datos de prueba que ofrece R para aprender a trabajar con el lenguaje:

```
data()
```

3. Carga de datos con paquetes externos

Si bien R trae de base algunas funciones ya mencionadas para cargar datos de ficheros externos de texto plano, existen paquetes especializados en esta tarea que ofrecen más opciones de carga o que contemplan otros tipos de ficheros.

Texto plano

La alternativa más popular a las funciones integradas de R probablemente sea `readr`, que está incluido a su vez en el paquete `tidyverse`, una colección de subpaquetes útiles para trabajar con datos en todas las fases de su análisis.

Para instalarlo y usarlo, hay dos opciones:

- Colección completa:

```
install.packages("tidyverse")
library(tidyverse)
```

- Paquete individual:

```
install.packages("readr")
library(readr)
```

Dependiendo del tipo de ficheros, `readr` ofrece algunas funciones específicas:

- read_csv(): Ficheros de valores separados por coma (CSV)
- read_csv2(): Ficheros de valores separados por punto y coma con , como delimitador numérico decimal (CSV2)
- read_tsv(): Ficheros de valores separados por tabuladores (TSV)
- read_delim(): Ficheros delimitados (como los casos partculares de CSV y TSV)
- read_fwf(): Ficheros de ancho fijo
- read_table(): Ficheros separados por espacios
- read_log(): Ficheros de web log

Por ejemplo, para el caso de CSV:

```
dataFromCsv <- read_csv('hotel_bookings_clean.csv', show_col_types = FALSE)
```

Excel

Para la carga de datos desde ficheros de Excel, se puede hacer mediante el paquete `readxl`:

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

dataFromExcel <- read_excel("tesla_deaths.xlsx", sheet = 1)
```

Nota: `readxl` pertenece a `tidyverse`, de manera que no necesita instalarse si ya se instaló la colección completa.

JSON

Para la carga de datos desde ficheros JSON, se puede hacer mediante el paquete `rjson`:

```
install.packages("rjson")
library(rjson)

dataFromJson <- fromJSON(file = "drake_data.json")
```

Que puede ser convertido en un data frame de esta forma:

```
dataFrameFromJson = as.data.frame(dataFromJson[1])
```

XML

Para la carga de datos desde ficheros XML, se puede hacer mediante el paquete `xml2`:

```
install.packages("xml2")
library(xml2)

dataFromXml <- read_xml("plant_catalog.xml")
dataXmlParse <- xmlParse(dataFromXml)
```

Que puede ser convertido en un data frame de esta forma:

```
dataNodes = getNodeSet(dataXmlParse, "//PLANT")
dataFrameXml <- xmlToDataFrame(nodes=dataNodes)
```

PDF

Para la lectura de texto desde ficheros PDF, se puede hacer mediante el paquete `pdftools`:

```
install.packages("pdftools")
library(pdftools)

textFromPdf <- pdf_text("1403.2805.pdf")
cat(txt[18])
```

Y para obtener datos de tablas en el PDF, se puede usar el paquete `tabulizer`:

```
dataFromPdf <- tabulizer::extract_tables(file = "1403.2805.pdf", pages =
18:19)
```

Referencias

Working with files and folders in R

Reading data from files
R scan examples
scan documentation
read.table documentation

How to import data into R
readr documentation
readxl documentation
rjson documentation
xml2 documentation

pdftools documentation
tabulizer documentation
pdftools & tabulizer example (I)
pdftools & tabulizer example (II)