

Practica 1 - Importacion de ficheros en R

Carlos Santiago Martinez Torres

2025-05-23

Contents

Configuracion inicial	2
Ejemplo ‘DFP1_11122015 124700.csv’	2
Importacion de fichero	2
Verificacion de los datos importados	2
Guardar el fichero con save	4
Ejemplo ‘BreadBasket_DMS.csv’	5
Importacion de fichero	5
Guardar el fichero con save	5
Ejemplo ‘PsychBike.sav’	5
Importacion de fichero	5
Guardar el fichero con save	5
Ejemplo ‘FileCodificado.json’	5
Importacion de fichero	5
Guardar el fichero con save	6
Exportar datos en formato JSON	6
Ejemplo ‘ERCA.xlsx’	6
Importacion de fichero	6
Guardar el fichero con save	6
Ejemplo ‘subjectInfo.xlsx’	7
Importacion del fichero	7
Guardar el fichero con save	8

Configuracion inicial

```
# Limpiar el entorno
rm(list = ls())

# Instalacion de librerias con pacman

# Asegurarse de que el paquete 'pacman' está instalado
if (!require("pacman")) install.packages("pacman")

# Poner las librerias necesarias
pacman::p_load(tidyverse, readxl, lubridate, knitr, tinytex, haven, jsonlite)

# Con tidyverse: ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr, forcats
# readxl, libreria que requiere una carga aparte explícita
# lubridate, se utiliza con frecuencia con paquetes de tidyverse, pero requiere una carga aparte
# tinytex, para LaTeX
# haven, para read files with `read_sav`
# jsonlite, trabajar con ficheros JSON
```

Ejemplo 'DFP1_11122015 124700.csv'

Importacion de fichero

```
# Si estoy en la carpeta que contiene la carpeta data,
# solo usar 'data/'. No es necesario el '/' al inicio.
carpeta <- 'data/'

fichero <- 'DFP1_11122015 124700.csv'

# Paste0 es una concatenacion pegada
# Paste personaliza el separador
f1 <- paste0(carpeta, fichero)

# ?read.csv
df1 <- read.csv(file = f1,
                header = TRUE,
                stringsAsFactors = FALSE,
                sep = '\\t')
```

Verificacion de los datos importados

```
str(df1)

## 'data.frame':   8887 obs. of  5 variables:
## $ FECHA          : chr  "11/12/2015" "11/12/2015" "11/12/2015" "11/12/2015" ...
## $ HORA           : chr  "12:47:02" "12:47:02" "12:47:02" "12:47:03" ...
```

```
## $ Altura          : num  138 138 138 138 138 ...
## $ Combinacion_Canya: num   2  2  2  2  2  2  2  2  2 ...
## $ Temperatura     : num  14.6 14.6 14.6 14.6 14.6 14.6 14.6 14.6 14.6 14.6 ...
```

Actualizacion de tipo de dato para fecha y hora

```
# Hacemos uso de la tuberia
df1 <- df1 %>%
  mutate(Fecha = as.Date(x = FECHA, format = '%d/%m/%Y'),
         Hora = hms(x = HORA)) %>% # hms::as_hms
  select(Fecha, Hora, Altura, Combinacion_Canya, Temperatura) # Select (-FECHA, -HORA). Select es para

# Verificamos los datos
str(df1)
```

```
## 'data.frame': 8887 obs. of 5 variables:
## $ Fecha          : Date, format: "2015-12-11" "2015-12-11" ...
## $ Hora           : Formal class 'Period' [package "lubridate"] with 6 slots
## .. ..@ .Data : num  2  2  2  3  3  3  3  3  4  4 ...
## .. ..@ year  : num  0  0  0  0  0  0  0  0  0  0 ...
## .. ..@ month : num  0  0  0  0  0  0  0  0  0  0 ...
## .. ..@ day   : num  0  0  0  0  0  0  0  0  0  0 ...
## .. ..@ hour  : num  12 12 12 12 12 12 12 12 12 12 ...
## .. ..@ minute: num  47 47 47 47 47 47 47 47 47 47 ...
## $ Altura        : num  138 138 138 138 138 ...
## $ Combinacion_Canya: num   2  2  2  2  2  2  2  2  2 ...
## $ Temperatura    : num  14.6 14.6 14.6 14.6 14.6 14.6 14.6 14.6 14.6 14.6 ...
```

Anotacion importante para fechas - TEMA 4

```
# Fechas - as.Date()

# Numero de días/meses/años
# 1970-01-01

# Fechas + Horas - as.POSIXct()

# Numero de horas/minutos/segundos
# 1970-01-01 01:00:00

# Fechas + Horas + TimeZone - as.POSIXlt()

# Valores de fecha/hora como una lista de componentes
```

Formato para las fechas

```

# DIAS
# %d - día como número, de 01 a 31
# %a - nombre del día abreviado, Mon
# %A - nombre completo del día, Monday

# MESES
# %m - mes como número, 00 a 12
# %b - nombre del mes abreviado, Jan
# %B - nombre completo del mes, January

# AÑO
# %y - dos dígitos del año, 07
# %Y - año completo, 2007

# EJEMPLOS
fecha <- '08-01-1988' # La fecha está como char

formatoDate <- as.Date(x = fecha, format = '%d-%m-%Y') # Debe ir '-', que es el formato en que está; no
formatoDate

```

```
## [1] "1988-01-08"
```

```

# Una vez que está como Date
format(x = formatoDate, format = 'She was born on %B %d del %y')

```

```
## [1] "She was born on enero 08 del 88"
```

```
format(x = formatoDate, format = 'Día de nac %A')
```

```
## [1] "Día de nac viernes"
```

```

# Otros ejemplos
fecha <- '8 enero 88'
dmy(fecha)

```

```
## [1] "1988-01-08"
```

```
format(dmy(fecha), format = '%d/%m/%Y')
```

```
## [1] "08/01/1988"
```

Guardar el fichero con save

```

# save(obj1, obj2, objn, file = 'nombrefichero.RData')
f1_salida <- 'dfp1.RData'
f1_salida <- paste0(carpeta, f1_salida)
save(df1, file = f1_salida)

```

Ejemplo ‘BreadBasket_DMS.csv’

Importacion de fichero

```
fichero2 <- 'BreadBasket_DMS.csv'
f2 <- paste0(carpeta, fichero2)

df2 <- read.csv(file = f2,
               header = TRUE,
               stringsAsFactors = FALSE)
```

Guardar el fichero con save

```
f2_salida <- 'bread.RData'
f2_salida <- paste0(carpeta, f2_salida)
save(df1, file = f2_salida)
```

Ejemplo ‘PsychBike.sav’

Importacion de fichero

```
fichero3 <- 'PsychBike.sav'
f3 <- paste0(carpeta, fichero3)

df3 <- read_sav(file = f3)
```

Guardar el fichero con save

```
f3_salida <- 'bikes.RData'
f3_salida <- paste0(carpeta, f3_salida)
save(df1, file = f3_salida)
```

Ejemplo ‘FileCodificado.json’

Importacion de fichero

```
ficheroJSON <- 'FileCodificado.json'
fjson <- paste0(carpeta, ficheroJSON)

# Si no simplifyVector, importa lista de listas por cada registro
# fjson <- read_json(path = fjson)
dfjson <- read_json(path = fjson, simplifyVector = TRUE)
```

Guardar el fichero con save

```
fjson_salida <- 'FileCodificado.RData'
fjson_salida <- paste0(carpeta, fjson_salida)
save(dfjson, file = fjson_salida)
```

Exportar datos en formato JSON

```
# Prueba con el conjunto Iris de R
# El conjunto iris ya esta cargado en R

irisJSON <- paste0(carpeta, 'iris.json')
write_json(x = iris, path = irisJSON)
```

Ejemplo 'ERCA.xlsx'

Importacion de fichero

```
fichero4 <- 'ERCA.xlsx'
f4 <- paste0(carpeta, fichero4)

df4 <- read_excel(path = f4, sheet = 2) # El formato del fichero no es correcto

# Si usamos el import del entorno, tenemos esto
ERCA <- read_excel("data/ERCA.xlsx", sheet = "Hojal - Tabla 1",
  col_types = c("numeric", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text",
    "text"))
```

Guardar el fichero con save

```
f4_salida <- 'ERCA.RData'
f4_salida <- paste0(carpeta, f4_salida)
save(ERCA, file = f4_salida)
```

Ejemplo 'subjectInfo.xlsx'

Importacion del fichero

```
fichero5 <- 'subjectInfo.xlsx'
f5 <- paste0(carpeta, fichero5)

# Lectura de cada hoja hasta la cuarta
hoja1 <- read_excel(path = f5, sheet = 1)
hoja2 <- read_excel(path = f5, sheet = 2)
hoja3 <- read_excel(path = f5, sheet = 3)
hoja4 <- read_excel(path = f5, sheet = 4)

# Todas las hojas automaticamente

# Obtener el nombre de las hojas del fichero excel
sheets <- excel_sheets(path = f5)

# Guardar los nombres en una lista
df_excel <- list()

for (n in 1:length(sheets)){
  df_excel[[n]] <- sheets[n]
}

df_excel

## [[1]]
## [1] "measurments"
##
## [[2]]
## [1] "neuropathy"
##
## [[3]]
## [1] "tuning fork NSS"
##
## [[4]]
## [1] "mobility"
##
## [[5]]
## [1] "Durometer"

# Hacer lo mismo con lapply
# lapply, aplica una funcion a todos los elementos de un vector o una lista, y devuelve una lista con l

# Funcion que lee una hoja de un fichero, pasando como argumento el nombre de la hoja
read_sheet <- function(sheet_name) {
  read_excel(path = f5, sheet = sheet_name)
}

# A cada hoja que hay en sheets (nombres de las hojas de f5), aplica la funcion anterior
x <- lapply(sheets, read_sheet) # Lista de dataframes
```

```
names(x) <- sheets # Cada dataframe tiene el nombre correspondiente a la hoja
```

Guardar el fichero con save

```
f5_salida <- 'FicheroExcel.RData'  
f5_salida <- paste0(carpeteta, f5_salida)  
save(x, file = f5_salida)
```