Importación de datos

Para aplicar las herramientas de R a nuestro trabajo, es necesario poder importar nuestros datos a R. R tiene conectores ya implementados para casi cualquier tipo y formato de datos. Entre los más comunes están[1]:

- Archivos de texto plano (como csvs) con `readr`.
- Datos desde una API con `httr`.
- Binarios (como excel o sas) con `haven` y `readxl`.

Lectura

Para leer un archivo, recordemos el comando `getwd()` para encontrar la carpeta a la cual R esta dirigido en este momento. Una buena practica es considerar el directorio de trabajo como el lugar en donde esta guardado el archivo o `sript` en el que se trabaja y "moverse" desde ahi hasta el archivo que se quiere leer.

R tiene conexion con basicamente todos los tipos de archivo. Veremos algunos de los mas relevantes.

csv (archivo separado por comas)

```
misdatos <- read.table("c:/misdatos.csv", header=TRUE,
   sep=",", row.names="id")
misdatos <- read.csv("c:/misdatos.csv")
```

Archivo de texto plano (txt, tsv, psv, etc)

`read.table` es mucho mas amplio que `read.csv` pues nos permite especificar casi cualquier particularidad en un archivo de texto plano.

```
misdatos <-
  read.table("C:/misdatos", header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

Excel

Dentro del paquete `readxl` se encuentra la funcion `read_excel` que es muy util en este caso.

```
misdatos <- read_excel("C:/misdatos.xlsx", sheet = 1, col_names = TRUE,
col_types = NULL, na = "", skip = 0)
```

[1] Los materiales para readr, httr, haven y readxl están basados en el capítulo de importación de datos de Grolemund y Wickham (2016).

SPSS

SPSS guarda los datos bastante bien: si uno les pone etiquetas entonces tiene el valor y las etiquetas para factores, etc. Este tipo de cosas, si ya fueron realizados por alguien mas, es una pena perderlos al convertirlo en un csv o un excel.

El paquete `foreign` permite leer archivos desde spss. (extension .sav)

```
misdatos <- read.spss("C:/misdatos.sav", use.value.labels = TRUE,
          max.value.labels = Inf, trim.factor.names = FALSE,
          trim_values = TRUE, reencode = NA)
```

Stata

No desperdiciar esos *do files*.

```
misdatos <- read.dta("C:/misdatos.dta", convert.dates = TRUE, convert.factors = TRUE,
         missing.type = FALSE,
         convert.underscore = FALSE, warn.missing.labels = TRUE)
```

Escritura

El mas comun es `write.table`

```
write.table(misdatos, file = "C:/misdatos", append = FALSE,
            quote = TRUE, sep = " ",
            eol = "\n", na = "NA", dec = ".", row.names = TRUE,
            col.names = TRUE, qmethod = c("escape", "double"),
            fileEncoding = "")
write.csv(misdatos, file = "C:/misdatos.csv")
```

Propios de R

Tambien pueden guardar objetos especificos del ambiente dentro de un formato especial con extension `rdata` o `RData`. Esto es muy util cuando no han acabado o quieren seguir trabajando con algo.

```
save(..., list = character(),
     file = stop("'file' must be specified"),
     ascii = FALSE, version = NULL, envir = parent.frame(),
     compress = isTRUE(!ascii), compression_level,
     eval.promises = TRUE, precheck = TRUE)
```

O pueden guardar un solo dataframe dentro de un archivo comprimido que R lee facilmente y mantiene toda la limpieza que han realizado sobre un dataframe

```
saveRDS(misdatos, file = "C:/misdatos.rds", ascii = FALSE, version = NULL,
        compress = TRUE, refhook = NULL)
```

Para leerlos

```
misdatos <- readRDS("C:/misdatos.rds")
```