Diplomado de Análisis Estadístico usando R

Módulo 2:
R Markdown para la elaboración de documentos y presentaciones

Profesor: *Víctor Macías E.*

1. Paquete *"tidyverse"*

El paquete *tidyverse* consiste de un conjunto de paquetes. Para una descripción detallada se recomienda revisar la siguiente página: https://www.tidyverse.org/.

Entre los paquetes que son parte de *tidyverse* y que permiten la importación de datos en varios formatos, se encuentran:

- *readr* (https://readr.tidyverse.org/)
- *readxl* (https://readxl.tidyverse.org/)
- *haven* (https://haven.tidyverse.org/)

1.1. readr

Comando	Formato
read_csv	valores separados por coma (csv)
read_csv2	valores separados por punto y coma (csv)
read_tsv	valores delimitados por tab (tsv)
read_delim	formato de archivo con cualquier delimitador

A continuación se presentan varios ejemplos relacionados al uso de *readr* para importar archivos con extensión *csv* que es uno de los más comunes para el almacenamiento de datos.

```
library(readr)
```

- La primera línea será usada como nombre de las variables

```
read_csv("Zona, Hombres, Mujeres
          A,700,200
          B,350,400")
```

```
## # A tibble: 2 x 3
##   Zona  Hombres Mujeres
##   <chr>   <dbl>   <dbl>
## 1 A         700     200
## 2 B         350     400
```

- Si el archivo contiene líneas de texto (por ejemplo, un título) puede omitirse usando `skip`

```
read_csv("Distribución de la muestra
          Zona, Hombres, Mujeres
          A,700,200
          B,350,400",
         skip = 1)
```

```
## # A tibble: 2 x 3
##   Zona  Hombres Mujeres
##   <chr>   <dbl>   <dbl>
## 1 A         700     200
## 2 B         350     400
```

- Si los datos no tienen nombres en sus columnas, se puede especificar `col_names = FALSE` para evitar que la primera fila de datos sea usada como nombre de las variables.

```
read_csv("A,700,200\nB,350,400", col_names = FALSE)
```

```
## # A tibble: 2 x 3
##   X1       X2    X3
##   <chr> <dbl> <dbl>
## 1 A       700   200
## 2 B       350   400
```

Se quiere evitar lo siguiente:

```
read_csv("A,700,200\nB,350,400")
```

```
## # A tibble: 1 x 3
##   A     `700` `200`
##   <chr> <dbl> <dbl>
## 1 B       350   400
```

- Se puede también especificar los nombres de las columnas usando `col_names`

```
read_csv("A,700,200\nB,350,400", col_names = c("Zona", "Hombres", "Mujeres"))
```

```
## # A tibble: 2 x 3
##   Zona  Hombres Mujeres
##   <chr>   <dbl>   <dbl>
## 1 A         700     200
## 2 B         350     400
```

- Si en los datos originales los *missing values* están representados por ., se pueden reemplazar por *NA* incluyendo `na = "."`

```
read_csv("Zona, Hombres, Mujeres
          A,.,200
          B,350,.",
         na = ".")
```

```
## # A tibble: 2 x 3
##   Zona  Hombres Mujeres
##   <chr>   <dbl>   <dbl>
## 1 A          NA     200
## 2 B         350      NA
```

- El paquete *readr* adivina automáticamente el tipo de cada columna.

```
read_csv("Fecha, var1, var2, var3
          2020-10-20, Mañana,.,200
          2020-10-31, Tarde, 350,.",
         na = ".")
```

```
## # A tibble: 2 x 4
##   Fecha      var1    var2  var3
##   <date>     <chr>  <dbl> <dbl>
## 1 2020-10-20 Mañana    NA   200
## 2 2020-10-31 Tarde    350    NA
```

- Si el tipo de variable no ha sido importado correctamente, se puede especificar el tipo de cada una usando `col_types`

```
read_csv("Fecha, var1, var2, var3
          2020-10-20, Mañana,.,200
          2020-10-31, Tarde, 350,.",
         na = ".", col_types =
           cols(
             Fecha = col_date(),
             var1 = col_character(),
             var2 = col_double(),
             var3 = col_double()
           ))
```

```
## # A tibble: 2 x 4
##   Fecha      var1    var2  var3
##   <date>     <chr>  <dbl> <dbl>
## 1 2020-10-20 Mañana    NA   200
## 2 2020-10-31 Tarde    350    NA
```

1.2. readxl

Comando	Formato
read_excel	autodetecta el formato (xls, xlsx)
read_xls	formato antiguo (xls)
read_xlsx	formato nuevo (xlsx)

Notas:

1. Si un archivo excel contiene más de una hoja puede usarse la función *excel_sheets* para identificar la hoja que nos interesa y luego agregar como argumento el nombre de la hoja a importar.
2. Para chequear cómo están separados los valores y si el archivo contiene un *header* se puede usar *read_lines*.

1.3. haven

Comando en R	Formato
read_dta	Stata
read_sav	SPSS
read_sas	SAS

2. Paquete *"data.table"*:

Este paquete se utiliza para manipulación de datos, pero también para importar datos, especialmente archivos muy grandes. Uno de sus atractivos más importantes es su gran rapidez. Para una descripción detallada, se recomienda ir al siguiente link https://cran.r-project.org/web/packages/data.table/data.table.pdf

3. Paquete *"WDI"*:

"*World Development Indicators*" (WDI) constituye el principal conjunto de indicadores de desarrollo del Banco Mundial que se extiende desde 1960 a la actualidad. Los datos se recolectan de diferentes fuentes internacionales reconocidas oficialmente y se pueden obtener de la siguiente página:

https://data.worldbank.org/products/wdi

Cada indicador tiene su propio código. Por ejemplo, las primeras 10 variables que incluyen en su nombre "gdp per cápita", se pueden obtener con el siguiente comando:

WDIsearch('gdp per capita')[1:10,]

Estos datos pueden ser importados a R, usando varios paquetes, según se detalla a continuación:

> "The WDI module and wbstats module offer excellent options for reading World Bank data directly into R, and both packages integrate with ggplot2 for graphing. Other option is to use the Quandl package, which also provides access to data sources from many other organizations.
>
> The *rWBclimate* package provides access to the climate data api".

Para bajar estos datos usaremos el paquete WDI.