

# Laboratorio 1 – Introducción a R y RStudio

## Table of contents

Primeros pasos . . . . .	1
Consola . . . . .	2
Paquetes . . . . .	2
Ejecutar un script . . . . .	2
Comentarios . . . . .	3
Explorar datos . . . . .	3
Carga de datos . . . . .	3
Navegación . . . . .	3
Estadísticas resumidas . . . . .	4
Visualización . . . . .	4
Creación de una nueva variable . . . . .	4
Eliminar observaciones . . . . .	5
Valores perdidos . . . . .	5

El objetivo de este laboratorio presencial es familiarizarse con R y RStudio. El laboratorio se puede realizar en grupo, pero cada estudiante debe entregar su propio trabajo. Para obtener créditos, suba su script .R al lugar correspondiente en Canvas.

## Primeros pasos

1. Abre RStudio en tu ordenador portátil
2. Haz clic en Archivo > Nuevo archivo > Script R.

Verás una sección en blanco en la parte superior izquierda de la ventana de RStudio. Aquí es donde escribirás tu primer script R.

## Consola

En la parte inferior izquierda de la pantalla hay una pestaña llamada «Consola». Básicamente, se trata de una calculadora muy sofisticada.

Prueba la calculadora escribiendo algo como:

```
2+2
```

O incluso algo más sofisticado como:

```
sqrt(pi)
```

## Paquetes

R hace un uso extensivo de paquetes de terceros. No entraremos en detalles ahora mismo, pero para esta clase necesitarás instalar algunos de ellos. Instalar paquetes es bastante fácil. Escribe las dos líneas de código siguientes en la parte superior de tu script:

```
install.packages("tidyverse", repos='http://cran.us.r-project.org')
install.packages("modelsummary", repos='http://cran.us.r-project.org')
install.packages("wooldridge", repos='http://cran.us.r-project.org')
```

Acabas de instalar tres paquetes. Básicamente, los has descargado en tu ordenador. Al igual que con cualquier otro software de tu ordenador, solo tienes que realizar la instalación una vez. Sin embargo, aún debes indicar a R que vas a utilizar los paquetes. Añade las dos líneas de código siguientes a tu script (debajo de las dos primeras líneas que has escrito). Fíjate en que esta vez no hay comillas dentro de los paréntesis.

```
library(tidyverse)
library(modelsummary)
library(wooldridge)
```

## Ejecutar un script

Para ejecutar el script, haz clic en la palabra «Fuente» en la esquina superior derecha del panel superior izquierdo de la ventana. Esto tomará lo que hay en tu script y lo enviará automáticamente a la consola (como si lo hubieras escrito directamente en la consola).

Para guardar el script, haga clic en el icono del disco situado en la parte superior del panel del script (pero no en el icono del disco situado en la parte superior de RStudio). Asigne al script el nombre «ICL1\_XYZ.R», donde «XYZ» son sus iniciales.

## Comentarios

Ahora, coloca un hashtag (#) delante de las dos primeras líneas de código de tu script, así:

```
#install.packages("tidyverse")  
#install.packages("modelsummary")  
#install.packages("wooldridge")
```

El hashtag es la forma de indicarle a R que no ejecute el código de tu script. Esto se conoce como «comentar» tu código.

En la parte superior de tu script, escribe tu nombre precedido de un hashtag.

**A partir de ahora, añade todo el código que veas a tu script.**

## Explorar datos

Ahora que ya conoces algunos conceptos básicos de R, ¡veamos algunos datos!

### Carga de datos

Vamos a cargar un conjunto de datos del paquete `wooldridge`. El conjunto de datos se llama `wage1`.

```
df <- as_tibble(wage1)
```

Lo que hicimos allí fue convertirlo a un «tibble», que es un formato muy útil para conjuntos de datos (véase el capítulo 10 de *@r4ds*). Llamamos al tibble convertido «df», pero puedes llamarlo como quieras: «mydata», «data123», lo que sea.

## Navegación

Si vuelve a ejecutar el script (haciendo clic en «Fuente»... o mejor aún, haciendo clic en la pequeña flecha situada junto a «Fuente», que abre un menú en el que puede hacer clic en «Fuente con eco»), verá algo nuevo en la ventana «Entorno» (arriba a la derecha). Dice «df» bajo el encabezado «Datos».

Haga doble clic en «df» en la ventana Entorno. Esto le mostrará sus datos en un formato similar a una hoja de cálculo de Excel. Puede utilizar esto para examinar fácilmente los datos y asegurarse de que todo parece razonable.

## Estadísticas resumidas

Veamos las estadísticas resumidas de una de nuestras variables. Supongamos que queremos saber: ¿Cuál es el promedio de años de educación en nuestra muestra?

```
datasummary_skim(df)#then look at 'educ' column
```

En tu script, escribe el valor de la media de «educ», precedido por un comentario (el símbolo de almohadilla).

¿Qué fracción de la muestra está compuesta por mujeres?

```
mean(df$female)
# or
datasummary_skim(df) #then look at 'female' column
```

## Visualización

Supongamos que desea visualizar toda la distribución de la educación. Utilizaría el siguiente código:

```
ggplot(df, aes(educ)) + geom_histogram(binwidth=1)+theme_classic()
```

En un comentario, escribe el valor más común de la educación (la moda de la distribución) debajo del código en tu script.

Repita los dos fragmentos de código anteriores, pero esta vez utiliza la variable **wage** en lugar de la variable **educ**.

## Creación de una nueva variable

Supongamos que desea añadir una nueva variable a **df**. Por ejemplo, la variable salario se expresa en dólares de 1976 y desea saber cuál sería el salario en dólares actuales. (Nota: el IPC implica que 1 dólar de 1976 equivale a 4,53 dólares actuales).

```
df <- df %>% mutate(realwage=wage*4.53)
summary(df$realwage)
datasummary_skim(
  df,
  fun_numeric = list(
    `Unique`      = NUnique,
```

```

`Missing Pct.` = PercentMissing,
`Mean`        = Mean,
`SD`          = SD,
`Min`         = Min,
`Median`      = Median,
`Max`         = Max
# NOTA: aquí NO ponemos "Histogram"
)
)
# then look at realwage column

```

Puede verificar que «realwage» se ha añadido a «df» haciendo clic en la vista previa de «df» y desplazándose completamente hacia la derecha.

Para obtener más información sobre «mutate()», consulte la sección 5.5 de @r4ds. También puede eliminar una variable escribiendo «df <- df %>% mutate(realwage=NULL)».

## Eliminar observaciones

Supongamos que queremos eliminar a todos los hombres de nuestros datos. (Por ejemplo, tal vez estemos investigando sobre la participación de las mujeres en la población activa). Para ello, utilizamos la función `filter()`.

Para utilizar `filter()`, debe proporcionar las condiciones para *mantener* una observación específica. Añada lo siguiente a su script:

```

summary(df$female)
df <- df %>% filter(female==1)
summary(df$female)

```

Le dijimos a R que conservara las observaciones en las que «female» era igual a 1. Se puede observar que, antes de la instrucción «filter()», las mujeres representaban el 48 % de los datos. Ahora, representan el 100 %. Por lo tanto, podemos verificar que «filter()» hizo lo que le pedimos.

## Valores perdidos

Una situación habitual en los datos observacionales transversales es la pérdida de valores. Por ejemplo, alguien deja en blanco una pregunta en una encuesta. O el salario de alguien que está desempleado no está definido. En R, los valores perdidos se almacenan como «NA» (que significa «no aplicable»). Para eliminar las observaciones «NA», utilice la función «drop\_na()»:

```
summary(df$wage)
df <- df %>% drop_na(wage)
summary(df$wage)
```