



Introducción a R

Monika Avila Márquez, Ph.D.

April 21, 2023

Quien soy?

Económetra boliviana con especialización en datos de panel.

- ▶ Post-doctorado en Estadística, Universidad de Ginebra-Suiza.
- ▶ Consultora en modelación econométrica, BIT-Suiza.
- ▶ Doctorado en Econometría, Universidad de Ginebra-Suiza.
- ▶ Utilizo Matlab, Stata, **R**, Python.
- ▶ Comencé a utilizar **R** en 2015.

Como se desarrolla el curso?

1. Explicación de conceptos.
2. Hoja de referencia que recapitulan las funciones explicadas.
3. Lab en los que se practica lo estudiado (Si el tiempo lo permite).

Iterar.

Objetivos Globales

1. Distinguir entre **R** y **RStudio**.
2. Comprender los objetos de **R**.
3. Importar Datos
4. Gestión de datos.
5. Graficar en R.
6. Saber utilizar funciones.
7. Comprender funciones estadísticas básicas.

Agenda

1. Introducción **R**
2. Introducción a **RStudio**
3. Directorio de trabajo
4. Sesión de **R**
5. Objetos en **R**
6. Vectores

Agenda

7. Matrices

8. Variables Categóricas

9. Listas

10. Dataframes

11. Importar Datos

12. Gestión de Datos

Agenda

13. Programando

14. Funciones

15. Funciones estadísticas básicas

16. Gráficos

17. Referencias

Introducción **R**

Objetivos de esta sección

1. Decir de donde viene **R**.
2. Explicar que permite hacer **R**.
3. Decir porque **R** es útil en econometria.

Que es **R**?

R es:

- ▶ un entorno de software libre y lenguaje de programación interpretado para computación estadística y gráfica.
- ▶ un proyecto GNU (un sistema operativo que es software gratuito).
- ▶ compatible con sistemas operativos UNIX, Windows, y MacOS.
- ▶ Mas de 10000 paquetes disponibles en CRAN (Comprehensive R Archive Network).
- ▶ Mas de 2 millones de usuarios.



Historia de **R**

- ▶ Desarrollado por Robert Gentleman y Ross Ihaka del Departamento de Estadística de la Universidad de Auckland en 1993.
- ▶ Desarrollo actual es responsabilidad del R Development Core Team (desde 1997).



Robert Gentleman

Ross Ihaka



Porque usar **R**?

R tiene dos grandes cualidades:

- ▶ "Pizza gratis": software fiable completamente gratuito.
- ▶ "Libertad de expresión": muy pocas restricciones en su uso y en la contribución de paquetes.

Como consecuencia de estas cualidades, **R** es:

- ▶ Transparente, replicable y testado por la comunidad.
- ▶ Accesible, y simple para utilizar.

Uso de **R** en Econometría

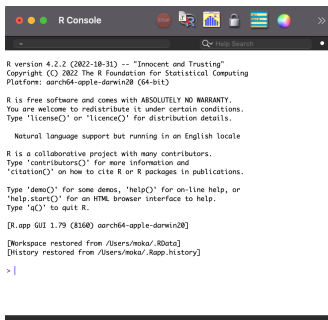
Econometría teórica:

- ▶ Experimento de Monte Carlo para testear métodos propuestos o disponibles.
- ▶ Testear nuevos métodos en bases de datos disponibles.

Econometría aplicada:

- ▶ Análisis de datos.
- ▶ Modelación de datos.
- ▶ Inferencia estadística.

Consola R



```
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.79 (8160) aarch64-apple-darwin20]

[Workspace restored from /Users/maka/.RData]
[History restored from /Users/maka/.Rapp.history]

> |
```

Ayuda y salir de **R**

1. Utilizar la función `help()` en la consola de funciones. El argumento de la función `help()` puede ser el nombre de la función. Ej. `help("sd")`.
2. Utilizar `?` seguido del nombre que se busca en la consola. Ej. `?sd`

`q()`: función para salir

IMPORTANTE: Minusculas, Mayusculas, y nombres de objetos

1. **R** distingue mayúsculas y minúsculas.
2. No utilizar nombres para objetos que ya estén tomados por el sistema (Ejemplo `lm`).

Hoja de Referencia: comentarios y funciones basicas

- ▶ Sabemos que un comando es una función si esta seguida de ().
- ▶ El carácter # sirve para escribir comentarios.
- ▶ Funciones útiles:

```
help() # Interface de ayuda  
help("mean") # Buscar ayuda sobre la función mean  
#q(): # Salir del programa.
```

Pre-Lab **R**: Instalación de **R**

1. Descargar el archivo binario de la [pagina web CRAN](#) para el sistema operativo correspondiente.



CRAN
Mirrors
What's new?
Search
CRAN Team

About R
R Homepage
The R format

Software
R Sources
R Sources
Packages
Task Views
Other

Documentation
Manuals
FAQs
Contributed

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2023-03-15, Shortstop Beagle) [R-4.2.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features](#) and [bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

2. Ejecutar el archivo binario.
3. Abrir **R**, y verificar correcto funcionamiento.

Nota: Se necesitan 500 MB (0.5 GB) de espacio en memoria GPU para la instalación.

Lab **R**: Primeros pasos

- ▶ Abrir R, y calcular $11+8+4+9$.
- ▶ Buscar ayuda para la funcion datasets.

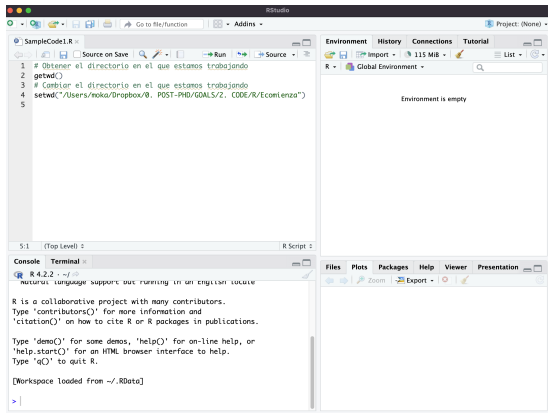
Nota: Guardamos la imagen del workspace (espacio de trabajo) cuando se crearon variables, o se modificaron los datos.

Introducción a **RStudio**

Que es **RStudio**?

RStudio es una aplicación de software para R, es de código abierto.

Existen dos versiones: **RStudio** Desktop, y **RStudio** Server.



Paneles de **RStudio**?

- ▶ Panel de código.
- ▶ Panel de consola.
- ▶ Panel de Ayuda.
- ▶ Panel de Entorno.
- ▶ Panel de historial (también se puede ver el historial de funciones utilizadas con control flecha arriba o alt flecha arriba).
- ▶ Panel de paquetes.
- ▶ Panel de graficos (`plot(women)`).

Y otros que no veremos a detalle en este curso ([Mas detalles](#)).

Hoja de Referencia: accesos directos útiles

- ▶ Utilice el tabulador para completar comandos, bases de datos u objetos en memoria.
- ▶ Utilice Control + arriba o Comando + arriba para obtener el historial de comandos (Solo RStudio).
- ▶ La tecla Escape sirve para salir del proceso iniciado por el comando actual.
- ▶ La tecla flecha arriba sirve para obtener el comando anterior.

Hoja de Referencia: Ayuda y salir de RStudio

Opciones para obtener ayuda:

1. Click en pestaña *help* de RStudio.
2. Utilizar la funcion `help()` en la consola de funciones. El argumento de la funcion `help()` puede ser el nombre de la función.
3. Utilizar `?` seguido del nombre que se busca en la consola. Ej. `?sd`

Salir de RStudio `q()`.

Pre-lab: Instalación de **RStudio**

Primero instalar **R**.

Descargar el archivo binario para instalación dependiendo del tipo de sistema operativo, y ejecutar.

Link para descargar instalador

DOWNLOAD

RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

1: Install R

RStudio requires R 3.5.0+. Choose a version of R that matches your computer's operating system.

DOWNLOAD AND INSTALL R

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR MACOS .Dmg

This version of RStudio is only supported on macOS 11 and higher. For earlier macOS environments, please [download a previous version](#).

Size: 374.55 MB | 100% CPU | 100% RAM | Version: 2023.03.0-1066 | Released: 2023-03-16

Lab **RStudio**: Primeros pasos

- ▶ Abrir **RStudio**, y calcular $11+8+4+9$.
- ▶ Buscar ayuda para el paquete datasets.

Pregunta: Comparar con R, que concluimos?

Nota: Guardamos la imagen del workspace (espacio de trabajo) cuando se crearon variables, o se modificaron los datos.

Directorio de trabajo

Directorio de trabajo

El directorio de trabajo es el lugar en el que se encuentran los archivos con los que trabajamos en **R**.

R va a importar archivos de este directorio, o exportar archivos a este directorio a menos que se indique otra cosa.

Hoja de referencia: Directorio de trabajo

1. `getwd()`: obtener el directorio de trabajo en el que se esta trabajando.
2. `setwd("Directorio en el que se va a trabajar")`
3. `list.files()`: obtener lista de archivos en el directorio de trabajo.

Nota: `~` reemplaza el directorio base en el que se trabaja.

Ejemplo: `"/Users/monika"` puede ser reemplazado por `~`. Entonces, al cambiar el directorio por la carpeta `CursoIntroR` que se encuentra en `"/Users/monika/Dropbox/CursoIntroR"`, podemos simplemente escribir `~/Dropbox/CursoIntroR"`

Si queremos comenzar en un directorio particular (default directorio), se puede ir a las opciones de RStudio.

Lab: Directorio de Trabajo

1. Obtener el directorio de trabajo en el que se esta trabajando.
2. Crear una carpeta en la que se va a guardar el trabajo de este curso.
3. Cambiar el directorio de trabajo a esta nueva carpeta.
4. Obtener la lista de todos los archivos en su directorio.

Sesion de **R**

Sesion de **R**

Una sesion de **R** es una instancia del entorno computacional de **R**. Esta contiene todos los objetos que se crean mientras trabajamos en este entorno y están almacenados en memoria RAM (Veremos los tipos de objetos luego). Por ejemplo:

```
vector1<-c(1,0,1,2,3)
```

```
vector1
```

```
## [1] 1 0 1 2 3
```

Estos objetos se guardan en nuestro espacio de trabajo (workspace). Estos objetos no se guardan automáticamente de sesión a sesión a menos que las guardemos.

Podemos guardar los objetos en nuestro entorno de trabajo.

Hoja de Referencia: Sesión de **R**

Dos opciones, click en el icono guardar o utilizar la función

```
save.image("midirectorio/nombreadarchivo.Rdata")
```

Hoja de Referencia: Sesión de **R**

Podemos listar los objetos creados en una sesión con la función:

```
ls()
```

```
## [1] "vector1"
```

Remover todo lo que se encuentra en el espacio de trabajo utilizando:

```
rm(list=ls())
```

Lab: Sesión de **R**

1. Crear un objeto en la sesión de **R**.
2. Guardar los objetos de esta sesión en un archivo que se encuentra en el directorio de trabajo.
3. Cerrar sesión, y abrir los datos guardados.

Pregunta: Que extension tiene el archivo que acabamos de guardar?

Instalar paquetes

Dos opciones:

- ▶ Utilizar `install.packages("nombre del paquete")`, e.g.
`install.packages("ggplot2")`
- ▶ Utilizar la consola de paquetes, pestaña instalar y buscar el paquete que nos interesa, por ejemplo ggplot2.

Una vez instalado, el paquete se encuentra en memoria GPU. No necesitamos instalar cada vez el paquete a menos que queramos hacer una actualización.

Utilizar funciones de un paquete

Se necesita cargar el paquete en memoria RAM. Para eso se necesita cargarlo como una librería (`library`). Hay dos opciones:

- ▶ Utilizar `library("nombre del paquete")`, e.g. `library("ggplot2")`
- ▶ Utilizar la consola de paquetes, pestaña instalar hacer click en el paquete de interés, ex. `ggplot2`.

Instalar paquetes

Dos opciones:

- ▶ Utilizar `install.packages("nombre del paquete")`, e.g.
`install.packages("ggplot2")`
- ▶ Utilizar la consola de paquetes, pestaña instalar y buscar el paquete que nos interesa, por ejemplo ggplot2.

Una vez instalado, el paquete se encuentra en memoria GPU. No necesitamos instalar cada vez el paquete a menos que queramos hacer una actualización.

Utilizar funciones de un paquete

Se necesita cargar el paquete en memoria RAM. Para eso se necesita cargarlo como una librería (`library`). Hay dos opciones:

- ▶ Utilizar `library("nombre del paquete")`, e.g. `library("ggplot2")`
- ▶ Utilizar la consola de paquetes, pestaña instalar hacer click en el paquete de interés, ex. `ggplot2`.

Hoja de Referencia: Paquetes

Una vez mas: No necesitamos instalar cada vez el paquete a menos que queramos hacer una actualización.

- ▶ Instalar paquete: `install.packages("nombre del paquete")`.
- ▶ Cargar paquete: `library("nombre del paquete")`.
- ▶ Eliminar paquete: `remove.package("nombre del paquete")`.

Lab: Paquetes

- ▶ Ejecutar el código `library(plm)`.
- ▶ Decir si el paquete se encuentra en memoria GPU.
- ▶ Ejecutar la function `data("EmplUK", package="plm")`.

Objetos en **R**

Objetos

Diferentes estructuras de datos son llamados objetos en **R**, podemos encontrar objetos como nombres de nombre de variables, vectores, listas, NULL, funciones, etc.

Una lista de los diferentes objetos en **R** se puede encontrar [aquí](#)

Hoja de referencia: Objetos

Para saber el tipo de objeto se puede utilizar:

1. `typeof()`

Revisamos el significado de los tipos arrojados por esta funcion [aquí](#).

Lab: Objetos

Determinar el tipo de objeto de:

1. mean.
2. a=1.
3. a="hola".
4. help.

Vectores

Vectores

Un vector es una colección de celdas contiguas que contienen datos homogéneos.

Ejemplos:

1. vector de números íntegros.
2. vector de números reales.
3. vector de palabras.
4. vector de booleanos.

Hoja de Referencia: Vectores

Se puede crear un vector de distintas formas:

1. `nombrevector <- c(,)`
2. `nombrevector <- limiteinferor:limitesuperior`
3. Usar generacion aleatoria (lo veremos mas tarde).
4. Para ver si es un vector usar la funcion: `is.vector(vectorint)`

Hoja de referencia: Ejemplos

Create vectors

Creamos un vector sin asignarlo.

```
1:10
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

Creamos un vector asignándolo.

```
vectorint<-1:10
```

```
vectorint
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

Hoja de referencia: Ejemplos

Como saber que tipo de vector creamos?

```
is.vector(vectorint)
```

```
## [1] TRUE
```

```
class(vectorint)
```

```
## [1] "integer"
```

Mas

Lab: Vectores

Generar:

1. Vector de caracteres a,d,r.
2. Vector de palabras bueno,bonito,barato.
3. Vector de números reales 1.21,1.6,1.8.
4. Vector a,1.6,2,casa, y ver si 1.6 y 2 son números.

Matrices

Matrices

Una matriz en **R** tiene dos dimensiones, y solo puede contener datos de un solo tipo.

Hoja de Referencia: Matrices

[Link script R](#)

[Link pdf](#)

Lab: Matrices

Crear una matrix diagonal bloque compuesta de una matriz diagonal de dimensiones (2x2) y una matrix llena de unos de dimension (2x2).

Variables Categóricas

Variables Categóricas

Variables categóricas en **R** deben ser codificadas como variables factor. Es importante codificarlas de esta manera para evitar errores en el análisis estadístico.

Hoja de Referencia: Variables Categoricalas

[Link script R](#)

[Link pdf](#)

Lab: Variables categóricas

Un individuo cuenta con una muestra aleatoria de 150 personas empleadas en Bolivia. 25 personas poseen un trabajo formal y el resto un trabajo informal.

1. Genere la base de datos correspondiente.
2. Obtener estadísticas apropiadas que resuman esta base de datos.

Listas

Listas

Una lista es un objeto en R que contiene datos heterogeneos, puede contener numeros enteros, numeros reales, caracteres, etc.

Hoja de Referencia: Listas

[Link script R](#)

[Link pdf](#)

Lab: Listas

Crear una lista que contiene los siguientes elementos: "cambio climatico", "plastico", 45, "contaminacion"

Dataframes

Dataframes

Dataframes son estructuras de datos que pueden contener datos de diferentes tipos.
Tienen propiedades de listas y matrices.

Hoja de Referencia: Data Frames

[Link script R](#)

[Link pdf](#)

Lab: Dataframes

Utilizando la base datos housing que se encuentra en el paquete MASS, decir:

1. Numero de observaciones.
2. Numero de variables.
3. Obtener la matriz de dispersión.

Importar Datos

Importar Datos

La importación de datos que no se encuentra en paquetes de R se puede realizar utilizando diferentes paquetes y funciones.

Los datos se pueden almacenar en diferentes tipos de archivos, y la función a utilizar para importar datos depende del tipo de archivo.

Hoja de Referencia: Programando

[Link script R](#)

[Link pdf](#)

Lab: Importar Datos

Utilizando un archivo de datos de interés para ustedes:

1. Importar la base de datos utilizando la función correcta .

Gestión de Datos

Gestión Datos

La gestión de datos es el proceso de transformar los datos brutos a datos más apropiados para obtener conclusiones valiosas. Uno de los pasos más importantes en la gestión de datos es la limpieza de los datos. Hoy nos concentramos en:

- ▶ Valores faltantes.
- ▶ valores extremos.

Hoja de Referencia: Programando

[Link script R](#)

[Link pdf](#)

Lab: Gestion de Datos

1. Escoger otra base de datos presente en el paquete mice (Pista: usar la funcion `data()`).
2. Determinar el numero de casos perdidos.
3. Determinar el patron de los casos perdidos.
4. Imputar los datos.

Programando

Programando

En esta seccion veremos:

- ▶ Operadores lógicos.
- ▶ Algebra Booleana
- ▶ Declaraciones Condicionales.
- ▶ Loops o bucle.

Operadores lógicos

Los operadores lógicos son operaciones binarias de comparación, y retornan valores logicos TRUE, y FALSE.

1. Mas grande que >.
2. Mas grande o igual que >=.
3. Mas pequeno <.
4. Mas pequeno o igual que <=.
5. Igual que ==.
6. No es igual a !=.

Algebra Booleana

1. $\&$ significa y.
2. $|$ significa o.

Declaraciones Condicionales: if

La estructura de la sentencia if es la siguiente:

```
if (condición){  
  Hacer algo  
} else {  
  Hacer otra cosa }
```

Los bucles permiten la ejecución repetitiva de bloques de código for una cantidad de veces que esta determinada de antemano.

La estructura:

```
for (i 1:K){  
  Hacer esto  
}
```

Hoja de Referencia: Programando

[Link script R](#)

[Link pdf](#)

Lab: Programando

Utilizando la base de datos USPersonalExpenditure:

1. Obtener el promedio de gastos personales para los años inferiores a 1960.
2. Comparar el promedio de gastos de los años 40s, 50s, 60s.

Funciones

Funciones

Funciones son objetos de R. Estas son bloques de código que convierten las entradas en resultados.

La estructura de función es la siguiente:

```
nombre_funcion(argumento1,argumento2,...){  
  bloque de codigo  
  return(resultado)  
}
```

Se reconoce que es una función si el nombre esta seguido de ().

Tipos de Funciones

Los tipos de funciones son:

1. Funciones internas de R.
2. Funciones de paquetes.
3. Funciones creadas por el usuario.

Uso de Funciones Internas

Escribir el nombre y los argumentos necesarios.

Ejemplo:

`q()`

Uso de Funciones de Paquetes.

Similar a las funciones internas, pero antes se debe instalar el paquete y cargar la librería.

Ej. La funcion `mic()` del paquete `mice`.

Uso de Funciones creadas por usuarios

Dos opciones:

1. Crear la función en el script en el que se trabaja.
2. Crear la función en otro script, y luego llamar la función utilizando `source("nombre de funcion").`

Hoja de Referencia: Funciones

[Link script R](#)

[Link pdf](#)

Lab: Funciones

1. Crear una función que genere la suma de dos números.
2. Añadir una opción que permita realizar el debugging de esta función.
3. Utilizar el argumento ... para que la función admita mas de dos números, y sume mas de dos números.

Funciones estadísticas básicas

Funciones estadísticas básicas

Vamos a utilizar funciones estadísticas básicas para:

- ▶ Generación de datos aleatorios.
- ▶ Obtener un gráfico de la distribución de probabilidad aleatoria.
- ▶ Test de hipótesis.
- ▶ Regresión lineal

Hoja de Referencia: Funciones básicas estadísticas

[Link script R](#)

[Link pdf](#)

Lab: Funciones estadísticas básicas

Utilizando la base de datos Informal que contiene la proporción de empleo informal en el empleo total por sexo y sector.

1. Obtener la distribución de probabilidad empírica para todos los datos.
2. Obtener la distribución de probabilidad empírica por sexo.
3. Testear si la proporción de empleo informal de hombres es igual a la de las mujeres.

Gráficos

Gráficos

Muchas opciones para realizar graficos, y hay cursos completos sobre este tema. Hoy vemos dos basicos:

- ▶ Funcion plot.
- ▶ Paquete ggplot2.

Hoja de Referencia: Gráficos

[Link script R](#)

[Link pdf](#)

Lab: Gráficos

Vamos a generar 10 observaciones que el siguiente proceso $y = x0.1 + \varepsilon$. Con $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, $\varepsilon \sim N(0, 1)$.

1. Realizar un diagrama de dispersión utilizando la función plot.
2. Realizar un gráfico linear utilizando la función plot.
3. Permutar el orden de x, y epsilon y realizar el gráfico de linea. Explicar porque no obtenemos algo correcto (Usar función permute del paquete gtools).
4. Corregir el problema anterior utilizando la funcion ts().

Referencias

Referencias

1. R Programming Fundamentals, StanfordOnline XDFS112.
2. CRAN, R Language Definition
3. Juan Bosco Mendoza Vega, R para principiantes, E-book
4. UCLA, R LEARNING MODULES
5. Utha University, Learn R
6. Hernandez F., Usuga O., Manual de R, e-book
7. Victoria-Feser M., Migliolo C., Blanc G., Model Selection in High Dimensions, e-book