

Introducción a R y RStudio

EST-11102 · Inferencia Estadística

C. Vladimir Rodríguez-Caballero

Departamento Académico de Estadística, ITAM

17 de agosto de 2026

Objetivo de esta sesión

- Familiarizarnos con **R** y **RStudio**, las herramientas de cómputo del curso.

Objetivo de esta sesión

- Familiarizarnos con **R** y **RStudio**, las herramientas de cómputo del curso.
- **No** es un curso de programación: es una introducción práctica para que puedan trabajar sin fricción.

Objetivo de esta sesión

- Familiarizarnos con **R** y **RStudio**, las herramientas de cómputo del curso.
- **No** es un curso de programación: es una introducción práctica para que puedan trabajar sin fricción.
- Todo el código del curso (ejemplos, tareas) estará disponible en R.

¿Por qué R?

- **Software libre** y de código abierto, mantenido por la comunidad estadística.

¿Por qué R?

- **Software libre** y de código abierto, mantenido por la comunidad estadística.
- Es el *lenguaje estándar* en estadística académica y muy usado en econometría aplicada.

¿Por qué R?

- **Software libre** y de código abierto, mantenido por la comunidad estadística.
- Es el *lenguaje estándar* en estadística académica y muy usado en econometría aplicada.
- Enorme ecosistema de paquetes: series de tiempo, econometría, finanzas, macro.

¿Por qué R?

- **Software libre** y de código abierto, mantenido por la comunidad estadística.
- Es el *lenguaje estándar* en estadística académica y muy usado en econometría aplicada.
- Enorme ecosistema de paquetes: series de tiempo, econometría, finanzas, macro.
- Reproducibilidad: el mismo script genera siempre el mismo resultado.

¿Por qué R?

- **Software libre** y de código abierto, mantenido por la comunidad estadística.
- Es el *lenguaje estándar* en estadística académica y muy usado en econometría aplicada.
- Enorme ecosistema de paquetes: series de tiempo, econometría, finanzas, macro.
- Reproducibilidad: el mismo script genera siempre el mismo resultado.

Contexto económico

Bancos centrales, organismos como el Banco Mundial y el FMI, y buena parte de la industria financiera usan R para análisis de datos y modelación.

R vs. RStudio: ¿cuál es la diferencia?

R

- El **motor**: el lenguaje y el intérprete.
- Se descarga de CRAN.
- Se puede usar solo, pero es incómodo.

RStudio

- El **IDE**: entorno de desarrollo.
- Se descarga de posit.co.
- Requiere tener R ya instalado.

R vs. RStudio: ¿cuál es la diferencia?

R

- El **motor**: el lenguaje y el intérprete.
- Se descarga de CRAN.
- Se puede usar solo, pero es incómodo.

RStudio

- El **IDE**: entorno de desarrollo.
- Se descarga de posit.co.
- Requiere tener R ya instalado.

Analogía: R es el motor del coche; RStudio es el tablero, el volante y los asientos.

Instalación (orden importa)

- 1 Instalar **R** desde <https://cran.r-project.org>

Instalación (orden importa)

- 1 Instalar **R** desde <https://cran.r-project.org>
- 2 Instalar **RStudio Desktop** (versión gratuita) desde <https://posit.co/download/rstudio-desktop/>

Nota

Si actualizan R en el futuro, RStudio detecta la nueva versión automáticamente; no es necesario reinstalar RStudio.

Instalación (orden importa)

- 1 Instalar **R** desde <https://cran.r-project.org>
- 2 Instalar **RStudio Desktop** (versión gratuita) desde <https://posit.co/download/rstudio-desktop/>
- 3 Abrir RStudio (no R directamente) para todo el trabajo del curso

Nota

Si actualizan R en el futuro, RStudio detecta la nueva versión automáticamente; no es necesario reinstalar RStudio.

La interfaz de RStudio

Script (.R) donde se escribe y guarda el código	Environment / History objetos creados en la sesión
Consola donde el código se ejecuta	Files / Plots / Packages / Help gráficas, paquetes y documentación

Consola vs. Script

Consola

- Ejecuta código *inmediatamente*.
- No se guarda automáticamente.
- Útil para probar cosas rápido.

Script (.R)

- Se guarda como archivo.
- Se ejecuta línea por línea con `Ctrl+Enter` (`Cmd+Enter` en Mac).
- Es lo que usaremos siempre en el curso.

Consola vs. Script

Consola

- Ejecuta código *inmediatamente*.
- No se guarda automáticamente.
- Útil para probar cosas rápido.

Script (.R)

- Se guarda como archivo.
- Se ejecuta línea por línea con Ctrl+Enter (Cmd+Enter en Mac).
- Es lo que usaremos siempre en el curso.

```
# Esto es un comentario
x <- 5           # asignacion
y <- 3
x + y           # se ejecuta y muestra 8
```

Conceptos básicos I: asignación y tipos

```
# Asignacion (preferir <- sobre =)
pib <- 1200000
tasa_interes <- 0.075
pais <- "Mexico"
es_miembro_ocde <- TRUE

class(pib)           # "numeric"
class(pais)          # "character"
class(es_miembro_ocde) # "logical"
```

Conceptos básicos I: asignación y tipos

```
# Asignacion (preferir <- sobre =)
pib <- 1200000
tasa_interes <- 0.075
pais <- "Mexico"
es_miembro_ocde <- TRUE

class(pib)           # "numeric"
class(pais)          # "character"
class(es_miembro_ocde) # "logical"
```

- R distingue mayúsculas/minúsculas: PIB $\neq$ pib.
- Nombres de objetos: sin espacios, no empezar con número.

Conceptos básicos II: vectores y data frames

```
# Vector: la estructura mas basica
inflacion <- c(4.2, 4.5, 3.9, 4.1)
mean(inflacion)
sd(inflacion)

# Data frame: tabla de datos (como Excel)
datos <- data.frame(
  pais = c("Mexico", "Chile", "Colombia"),
  inflacion_2024 = c(4.2, 4.0, 6.1)
)
head(datos)
```

Paquetes: extendiendo R

- R base ya trae mucho, pero los **paquetes** añaden funciones especializadas.

Paquetes: extendiendo R

- R base ya trae mucho, pero los **paquetes** añaden funciones especializadas.
- Se instalan *una* vez, se cargan *cada sesión*.

```
# Instalar (solo una vez por computadora)
install.packages("tidyverse")

# Cargar (cada vez que se abre RStudio)
library(tidyverse)
```

Paquetes: extendiendo R

- R base ya trae mucho, pero los **paquetes** añaden funciones especializadas.
- Se instalan *una* vez, se cargan *cada sesión*.

```
# Instalar (solo una vez por computadora)
install.packages("tidyverse")

# Cargar (cada vez que se abre RStudio)
library(tidyverse)
```

Paquetes que usaremos en el curso

tidyverse, tseries, nortest, entre otros para pruebas de hipótesis y validación de normalidad.

Organización: RStudio Projects

- Un **proyecto** (`.Rproj`) fija la carpeta de trabajo automáticamente.

Organización: RStudio Projects

- Un **proyecto** (`.Rproj`) fija la carpeta de trabajo automáticamente.
- Evita el uso de `setwd()` con rutas absolutas (que no funcionan en la computadora de alguien más).

Buena práctica

Todo script del curso debe poder ejecutarse de principio a fin sin errores en una sesión nueva de R (“reproducibilidad”).

Organización: RStudio Projects

- Un **proyecto** (`.Rproj`) fija la carpeta de trabajo automáticamente.
- Evita el uso de `setwd()` con rutas absolutas (que no funcionan en la computadora de alguien más).
- **Recomendación:** un proyecto por curso o por tarea, con subcarpetas `datos/`, `scripts/`, `figuras/`.

Buena práctica

Todo script del curso debe poder ejecutarse de principio a fin sin errores en una sesión nueva de R (“reproducibilidad”).

Ayuda y recursos

- `?funcion` o `help(funcion)` dentro de R.
- Paquete `swirl`: tutoriales interactivos dentro de la consola.
- Libro gratuito *R for Data Science* (Wickham & Grolemund).
- CRAN Task View: Econometrics — paquetes especializados.
- Los scripts del curso estarán en el sitio del curso (GitHub Pages).

Lo que viene

- De aquí en adelante, cada tema del curso vendrá acompañado de código en R.
- El objetivo **no** es programar de forma elegante, sino **entender la estadística** usando la computadora como herramienta.
- Practiquen: instalar R y RStudio antes de la próxima clase.

¡Bienvenidos a R!