

R para economistas

Alberto R. Durán Pérez

2026-02-03

Contents

0.1 Introducción: ¿Qué es R y por qué es útil en Economía?

R es un lenguaje de programación muy potente diseñado específicamente para el análisis estadístico y la visualización de datos. Es el estándar en muchas áreas científicas y de negocios.

Para profundizar en la introducción de R aplicada a la Economía, es necesario entender que no estamos ante una simple calculadora, sino ante un laboratorio completo de econometría.

R es un lenguaje de código abierto diseñado específicamente para el análisis estadístico y la visualización de datos. En el ámbito económico, su utilidad radica en que se ha convertido en el estándar tanto en áreas científicas como en el mundo de los negocios.

0.1.1 ¿Por qué R es el aliado del economista?

A diferencia de las hojas de cálculo tradicionales, R permite manejar grandes volúmenes de datos y asegurar la reproducibilidad. Si realizas un análisis hoy, cualquier otra persona puede ejecutar tu código y obtener exactamente los mismos resultados, algo vital en la investigación económica.

Haciendo un símil con un coche para entender las herramientas:

- **R (El Motor):** Es la potencia bruta que realiza los cálculos matemáticos y estadísticos. Sin él, no hay movimiento.
- **RStudio (El Volante y Tablero):** Es el entorno de desarrollo (IDE) que hace que escribir código sea una experiencia mucho más amigable, visual y cómoda. Es la interfaz que te permite controlar el motor sin complicaciones innecesarias.

0.1.2 Áreas de aplicación en Economía

1. **Econometría:** Estimación de modelos de regresión lineal y no lineal, series temporales y paneles de datos.
2. **Análisis de Políticas Públicas:** Evaluación de impacto mediante la manipulación de microdatos (como censos o encuestas de hogares).
3. **Visualización de Datos:** Creación de gráficos de alta calidad para reportes financieros o académicos.
4. **Automatización:** Generación de informes automáticos mediante RMarkdown para combinar análisis y resultados en un solo paso.

Esta base sólida es el primer paso del camino, que hemos dividido en tres etapas lógicas: el taller (**instalación**), el lenguaje (**sintaxis**) y la publicación (**RMarkdown**).

0.2 El “taller”: Instalación y configuración

Para trabajar cómodamente, normalmente no usamos **R** “a secas”, sino que lo combinamos con **RStudio**, que es un entorno de desarrollo (IDE) que hace que *escribir código sea mucho más amigable*.

En esta sección, aprenderás a preparar tu equipo de trabajo. Los puntos clave a incluir son:

- **Instalación dual:** Deben instalar primero R y luego RStudio.
- **Interfaz de RStudio:** Es necesario conocer para qué sirve cada ventana del programa para que los usuarios se sientan orientados desde el primer momento.

Veamos las instalación **tanto en Windows como en Mac**. El proceso es muy similar, pero los archivos de instalación del “motor” (R) son distintos.

Vamos a instalar primero el **motor (R)**. Recuerda: sin esto, RStudio no funcionará.

Paso 1: Instalar R (El Motor)

Ve al sitio oficial llamado **CRAN** (Comprehensive R Archive Network). Aquí es donde vive R.

Para Windows:

1. Entra en: cran.r-project.org/bin/windows/base/
2. Haz clic en el enlace grande que dice “**Download R [versión] for Windows**”.
3. Ejecuta el archivo .exe y acepta todo por defecto (Siguiente, Siguiente...).

Para Mac:

1. Entra en: cran.r-project.org/bin/macosx/
2. Aquí hay una pequeña bifurcación importante:
 - Si tu Mac tiene chip **Apple (M1, M2, M3...)**, descarga el archivo que dice **arm64.pkg**.
 - Si es una Mac más antigua con chip **Intel**, descarga el archivo que dice **x86_64.pkg**.
3. Instala el archivo .pkg como cualquier otra aplicación de Mac.

Paso 2: Instalar RStudio (El Tablero)

Una vez que el motor está instalado en ambos equipos, vamos por el tablero de control.

1. Ve al sitio de **Posit** (los creadores de RStudio): posit.co/download/rstudio-desktop/
2. Desplázate hacia abajo hasta encontrar el botón “**Download RStudio Desktop for...**” (la página suele detectar tu sistema automáticamente).
3. Instálalo (en Mac arrastrándolo a Aplicaciones, en Windows con el instalador).

0.2.1 Verificación

Cuando termines, **no abras el icono de “R”**. Busca y abre únicamente el programa **RStudio**.

Deberías ver una ventana dividida en paneles (generalmente 3 o 4). ¿Pudiste abrir RStudio correctamente y ver algo parecido a esto?

0.3 Primeros pasos: Sintaxis

¡Excelente! Si ya tienes **RStudio** abierto, deberías ver una interfaz dividida en cuadrantes. Por ahora, lo más importante es que localices la **Consola** (Console), que suele ser el panel grande a la izquierda o abajo a la izquierda donde aparece un símbolo de mayor que >. Ahí es donde R espera tus órdenes.

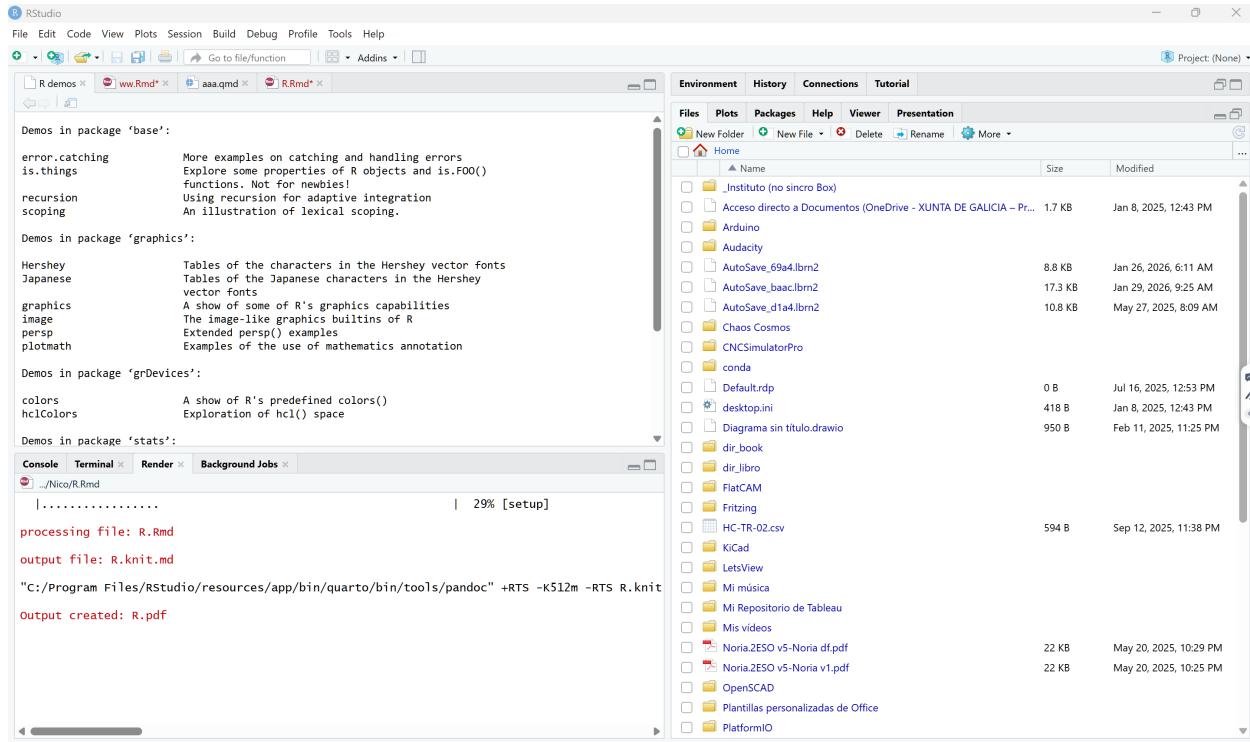


Figure 1: Interfaz de RStudio

Vamos a empezar a “hablar” con R para entender cómo maneja la información. En R, casi todo se basa en **Variables y Funciones**.

0.3.1 Variables y Asignación

Para **guardar un dato y usarlo después**, usamos el operador de asignación `<-` (flecha hacia la izquierda), o bien, el signo igual `=`. Es como una caja con una etiqueta.

```
mi_edad <- 25
# (Guardamos el número 25 en la variable "mi_edad").

nombre <- "Alex"
# (Guardamos texto, que siempre va entre comillas).

cantidad = 127
```

0.3.2 Vectores: Las listas de R

R no solo maneja números sueltos; su especialidad son las **listas de datos** del mismo tipo, llamadas **Vectores**. Se crean con la función `c()` (de combinar).

```
precios <- c(10, 15, 20, 25)

nombres <- c("Ana", "Luis", "Marta")
```

Para ver si el motor está bien aceptado, intentemos un ejercicio rápido en tu **consola**. Imagina que tienes las notas de tres exámenes: 8, 7.75 y 6.50.

¿Cómo escribirías el código para crear un vector (lista) llamado `mis_notas` que contenga esos tres números? (Pista: usa la función `c()`).

¡Muy bien! Vamos a poner a prueba ese motor.

En la **Consola**, escribe la siguiente línea y presiona **Enter**:

```
mis_notas = c(8, 7.75, 6.50)
```

Si lo hiciste bien, notarás que en el panel superior derecho (llamado **Environment**) ahora aparece **mis_notas**. R ya lo tiene en su memoria.

0.3.2.1 Operaciones Vectorizadas Lo mejor de R es que puedes hacer cálculos con toda la lista de datos a la vez. No necesitas sumar uno por uno. Por ejemplo, si el profesor decide regalar **2 puntos** a todos por buena conducta, solo tienes que escribir:

```
mis_notas + 2
```

```
## [1] 10.00 9.75 8.50
```

R le sumará 2 a cada número del vector automáticamente.

0.3.3 Tu turno: El Desafío del Promedio

R tiene funciones integradas para casi todo. La función para calcular el **promedio** (o **media aritmética**) se llama **mean()**.

¿Cómo intentarías calcular el promedio de tu vector mis_notas usando esa función? (Pista: Pon el nombre de tu vector dentro de los paréntesis).

```
mean(mis_notas)
```

```
## [1] 7.416667
```

¡Exacto! Esa es la forma de “llamar” a una función en R. Si ejecutas **mean(mis_notas)**, R te devolverá el promedio exacto de esos tres valores.

Ahora que ya sabes cómo darle órdenes básicas a R, vamos a subir de nivel. Hasta ahora hemos escrito código directamente en la **Consola**, pero eso tiene un problema: cuando cierras el programa, el código desaparece.

Para trabajar como un profesional, usamos **R Markdown**.

0.4 ¿Qué es R Markdown?

R Markdown es un formato de archivo que te permite mezclar **tres elementos** en un solo lugar:

1. **Texto** normal (para explicar lo que haces).
2. **Código** de R (para hacer los cálculos).
3. **Resultados** (gráficas y tablas que se generan automáticamente).

0.4.1 Cómo crear tu primer documento

1. En RStudio, ve a **File -> New File -> R Markdown...**
2. Ponle un título (por ejemplo: “Mi Primer Análisis”) y haz clic en **OK**.
3. Se abrirá un archivo con un texto de ejemplo. Verás unas cajas grises que empiezan con ````{r}`. Esas cajas se llaman **Chunks** (trozos de código).

0.4.1.1 Tu primera misión en R Markdown Busca un botón en la parte superior que tiene un icono de una pequeña madeja de lana y dice **Knit** (tejer) . Al presionarlo, RStudio “tejerá” todo tu código y texto para crear un archivo **HTML** o **PDF** profesional.

¿Qué pasa cuando intentas presionar ese botón de “Knit” por primera vez? (A veces RStudio te pedirá instalar unos paquetes adicionales antes de continuar).

Es muy común que la primera vez que presionas **Knit** , RStudio te muestre un mensaje en la parte superior diciendo que faltan algunos “paquetes” (como `knitr`, `rmarkdown` o `tinytex`). Si te aparece un botón que dice **Install**, dale clic sin miedo; son las herramientas que R necesita para traducir tu código a un documento bonito.

Una vez que el proceso termina, se abrirá una ventana nueva con tu documento terminado. Lo que acabas de hacer es “renderizar” el archivo.

0.4.2 La estructura de un R Markdown

Para que seas un experto en esto, debes distinguir las tres partes de un archivo `.Rmd`:

1. **El Encabezado (YAML):** Está al principio entre líneas ---. Aquí defines el título, el autor y si quieres un HTML, PDF o Word.
 2. **El Texto:** Se escribe fuera de las cajas grises. Puedes usar formato simple:
 - ****negrita****
 - **# Título**
 - **## Subtítulo**
 3. **Los Chunks (Código):** Son las cajas grises que empiezan con “`{r}`”. Todo lo que escribas ahí dentro será ejecutado por el motor de R.
-

0.4.3 Vamos a personalizarlo

Busca en tu archivo de ejemplo una de esas cajas grises (chunks). Borra lo que tiene adentro y escribe el código que usamos antes:

```
mis_notas <- c(85, 90, 78)
mean(mis_notas)
```

```
## [1] 84.33333
```

Ahora, vuelve a presionar **Knit**. Verás que en tu documento final aparece tanto el código como el resultado del promedio justo debajo.

¿Qué te parece el resultado? ¿Te gustaría intentar crear una gráfica sencilla dentro de ese mismo documento para ver cómo se incluye automáticamente?

0.5 Visualización de los datos

¡Excelente! Ya tienes tu primer documento “vivo” donde el código y la explicación conviven.

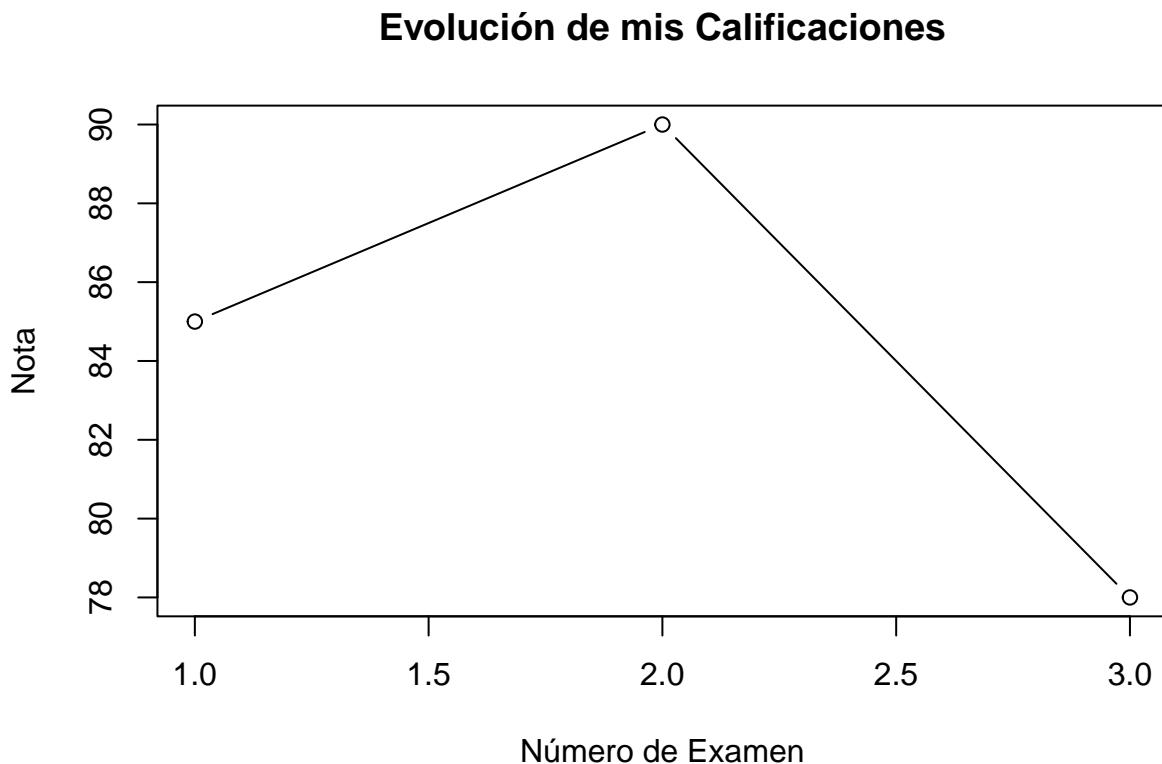
Para que tu reporte pase de ser una lista de números a un análisis real, vamos a añadir una **gráfica**. R es famoso por su capacidad de visualización.

0.5.1 Gráficas instantáneas

En R, existe una función básica llamada `plot()`. Si le pasas un vector de números, R entenderá que quieres ver cómo se distribuyen.

Intenta añadir esto en un nuevo **chunk** (una nueva caja gris de código) dentro de tu archivo R Markdown:

```
plot(mis_notas,
     type = "b",
     main = "Evolución de mis Calificaciones",
     xlab = "Número de Examen",
     ylab = "Nota")
```



0.5.1.1 Entendiendo los argumentos Fíjate que dentro de los paréntesis no solo pusimos los datos, sino también “instrucciones” extra:

- `type = "b"`: Crea una gráfica de puntos y líneas (`both`).
- `main`: El título principal.
- `xlab / ylab`: Etiquetas para los ejes X e Y.

0.5.2 El toque final: Personalización

R Markdown te permite decidir **cómo** se ve ese código en el informe final usando opciones dentro de las llaves del chunk `{r}`.

Por ejemplo, si quieres que la gráfica aparezca en tu PDF/HTML, pero no quieres que se vea el código “feo” que la generó, puedes usar `echo = FALSE`.

¿Te gustaría que probemos a cambiar el color de la gráfica o prefieres que veamos cómo organizar mejor el texto con títulos y subtítulos en el documento?

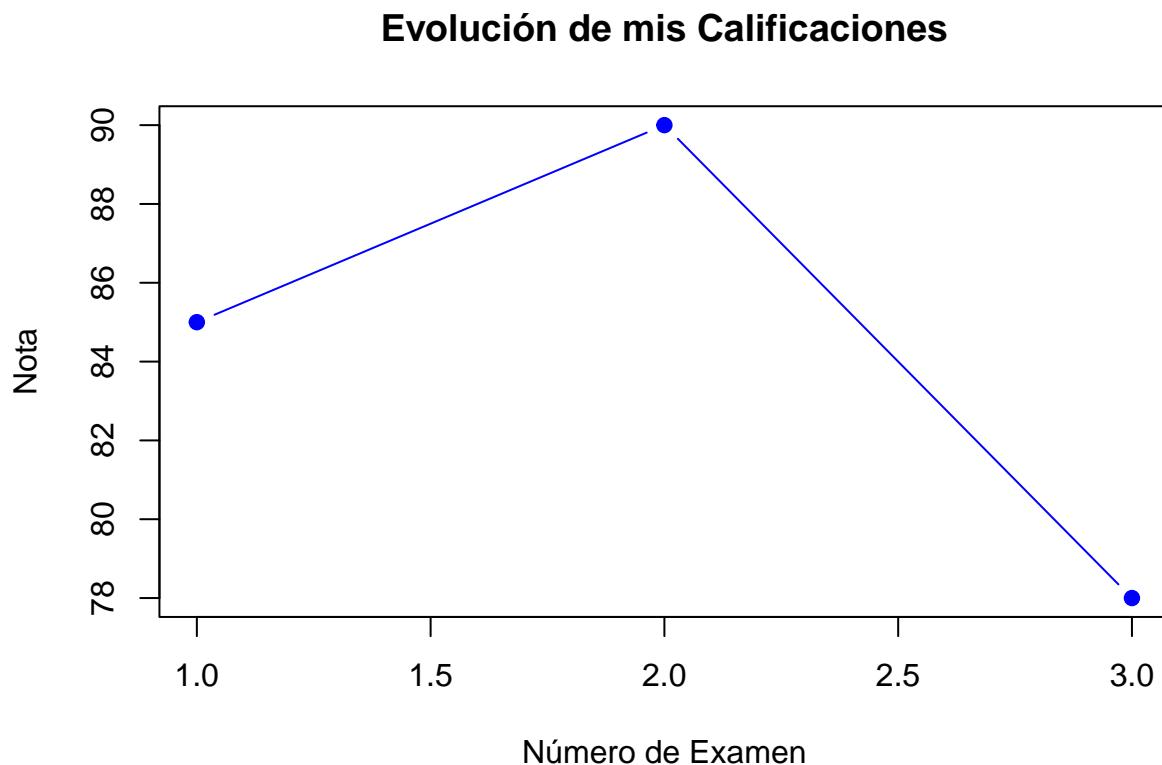
¡Excelente! Vamos a darle un poco de color y estructura a ese documento.

Para cambiar el color de la gráfica, usamos el argumento `col`. R reconoce nombres de colores básicos en inglés (como "blue", "red", "darkgreen", etc.).

0.5.2.1 Personalizando la gráfica

Modifica el código de tu chunk así:

```
plot(mis_notas,
      type = "b",
      col = "blue",           # Cambia el color a azul
      pch = 19,                # Cambia el estilo del punto a uno sólido
      main = "Evolución de mis Calificaciones",
      xlab = "Número de Examen",
      ylab = "Nota")
```



0.5.2.2 Estructurando el texto con Markdown

Fuera de las cajas grises, puedes organizar tu explicación usando niveles de títulos. Esto es muy importante para que el documento final tenga un índice y sea fácil de leer. Pruébalo escribiendo esto en el área de texto:

```
# Reporte del Semestre
Este es mi primer análisis automático usando **R Markdown**.

## Análisis de Exámenes
A continuación, se muestra el promedio y la tendencia de mis notas.
```

Al presionar **Knit**, verás que el texto se convierte en títulos grandes y pequeños, y la gráfica aparecerá con el nuevo color.

0.5.3 ¿Qué sigue en nuestro aprendizaje?

Ahora que ya sabes instalar, escribir código básico y generar un reporte, podemos profundizar en lo que más te interese:

1. **Manipulación de Datos:** Aprender a usar **Data Frames** (tablas como las de Excel) y filtrarlas.
2. **Gráficas Avanzadas:** Introducción a **ggplot2**, la librería estándar para hacer gráficos de nivel profesional.
3. **Configuración de RMarkdown:** Aprender a ocultar el código, cambiar el formato a PDF o añadir tablas automáticas.

¿Cuál de estos caminos te llama más la atención para continuar?
