

Curso R y Rstudio

Módulo 1: Introducción a R y RStudio

09 de June de 2025

1. ¿Qué es R?

Breve Historia y Propósito

- **Origen:** Nació a principios de los 90 en la Universidad de Auckland, Nueva Zelanda.

1. ¿Qué es R?

Breve Historia y Propósito

- **Origen:** Nació a principios de los 90 en la Universidad de Auckland, Nueva Zelanda.
 - Creado por **Ross Ihaka** y **Robert Gentleman**.

1. ¿Qué es R?

Breve Historia y Propósito

- **Origen:** Nació a principios de los 90 en la Universidad de Auckland, Nueva Zelanda.
 - Creado por **Ross Ihaka y Robert Gentleman**.
 - Propósito: Un entorno estadístico **libre (open source)** basado en el lenguaje **S**.

1. ¿Qué es R?

Breve Historia y Propósito

- **Origen:** Nació a principios de los 90 en la Universidad de Auckland, Nueva Zelanda.
 - Creado por **Ross Ihaka y Robert Gentleman**.
 - Propósito: Un entorno estadístico **libre (open source)** basado en el lenguaje **S**.
- **Filosofía:** Combinar la potencia de S con una sintaxis más accesible.

Ventajas de R

- **Especializado en análisis estadístico y científico:**

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**
 - Ejecuta comandos **al vuelo** en la consola.

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**
 - Ejecuta comandos **al vuelo** en la consola.
- No necesita compilación.

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**
 - Ejecuta comandos **al vuelo** en la consola.
- No necesita compilación.
- **Comunidad activa y en crecimiento:**

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**
 - Ejecuta comandos **al vuelo** en la consola.
- No necesita compilación.
- **Comunidad activa y en crecimiento:**
 - Amplia documentación, foros, grupos de usuarios (R-Ladies, RStudio Community, StackOverflow).

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**
 - Ejecuta comandos **al vuelo** en la consola.
- No necesita compilación.
- **Comunidad activa y en crecimiento:**
 - Amplia documentación, foros, grupos de usuarios (R-Ladies, RStudio Community, StackOverflow).
- **Gráficos de alta calidad:**

Ventajas de R

- **Especializado en análisis estadístico y científico:**
 - Miles de **paquetes específicos** en CRAN para casi cualquier tipo de análisis.
- **Lenguaje interpretado y flexible:**
 - Ejecuta comandos **al vuelo** en la consola.
- No necesita compilación.
- **Comunidad activa y en crecimiento:**
 - Amplia documentación, foros, grupos de usuarios (R-Ladies, RStudio Community, StackOverflow).
- **Gráficos de alta calidad:**
 - Con paquetes como `ggplot2`, `lattice`, `plotly`, crea visualizaciones impresionantes.

Instalación de R y RStudio

- **Instalar R:**

Instalación de R y RStudio

- **Instalar R:**

- Descargar desde **CRAN**: <https://cran.r-project.org>

Instalación de R y RStudio

- **Instalar R:**

- Descargar desde **CRAN**: <https://cran.r-project.org>
- Elegir según tu **sistema operativo

Instalación de R y RStudio

- **Instalar R:**

- Descargar desde **CRAN**: <https://cran.r-project.org>
- Elegir según tu **sistema operativo
- Seguir el instalador (las opciones predeterminadas suelen ser adecuadas).

Instalación de R y RStudio

- **Instalar R:**

- Descargar desde **CRAN**: <https://cran.r-project.org>
- Elegir según tu **sistema operativo
- Seguir el instalador (las opciones predeterminadas suelen ser adecuadas).

- **Instalar RStudio (IDE):**

Instalación de R y RStudio

- **Instalar R:**

- Descargar desde **CRAN**: <https://cran.r-project.org>
- Elegir según tu ****sistema operativo**
- Seguir el instalador (las opciones predeterminadas suelen ser adecuadas).

- **Instalar RStudio (IDE):**

- Descargar el **IDE de escritorio** desde:
<https://posit.co/download/rstudio-desktop/>

Instalación de R y RStudio

- **Instalar R:**

- Descargar desde **CRAN**: <https://cran.r-project.org>
- Elegir según tu ****sistema operativo**
- Seguir el instalador (las opciones predeterminadas suelen ser adecuadas).

- **Instalar RStudio (IDE):**

- Descargar el **IDE de escritorio** desde:
<https://posit.co/download/rstudio-desktop/>
- RStudio detectará automáticamente tu instalación de R al abrirlo.******

Sistemas Operativos Soportados

- **Windows:** Ejecutable .exe

Sistemas Operativos Soportados

- **Windows:** Ejecutable .exe
- **macOS:** Paquete .pkg

Sistemas Operativos Soportados

- **Windows:** Ejecutable .exe
- **macOS:** Paquete .pkg
- **Linux:** Paquetes .deb (Debian/Ubuntu) o .rpm (Fedora/Red Hat). También se puede compilar desde el código fuente.

2. Interfaz de RStudio

Paneles Principales

- Consola (Console)

Ejemplo en la Consola

```
> 2 + 2
```

```
> x <- 10
```

```
> x * 3
```

2. Interfaz de RStudio

Paneles Principales

- **Consola (Console)**
 - Donde se ejecutan los comandos directamente.

Ejemplo en la Consola

```
> 2 + 2
```

```
> x <- 10
```

```
> x * 3
```

2. Interfaz de RStudio

Paneles Principales

- **Consola (Console)**
 - Donde se ejecutan los comandos directamente.
 - `>`: indica que R está esperando un comando.

Ejemplo en la Consola

```
> 2 + 2
```

```
> x <- 10
```

```
> x * 3
```

2. Interfaz de RStudio

Paneles Principales

- **Consola (Console)**
 - Donde se ejecutan los comandos directamente.
 - `>`: indica que R está esperando un comando.
 - `+`: indica que R espera la continuación de un comando (ej. en líneas largas).

Ejemplo en la Consola

```
> 2 + 2
```

```
> x <- 10
```

```
> x * 3
```

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).
- **Ventajas:** Permite organizar, documentar y reproducir tu trabajo.

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).
- **Ventajas:** Permite organizar, documentar y reproducir tu trabajo.
- **Atajos para ejecutar:**

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).
- **Ventajas:** Permite organizar, documentar y reproducir tu trabajo.
- **Atajos para ejecutar:**
 - Windows/Linux: Ctrl + Enter (ejecuta línea actual o selección).

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).
- **Ventajas:** Permite organizar, documentar y reproducir tu trabajo.
- **Atajos para ejecutar:**
 - Windows/Linux: Ctrl + Enter (ejecuta línea actual o selección).
 - macOS: Cmd + Enter.

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).
- **Ventajas:** Permite organizar, documentar y reproducir tu trabajo.
- **Atajos para ejecutar:**
 - Windows/Linux: Ctrl + Enter (ejecuta línea actual o selección).
 - macOS: Cmd + Enter.
- **Guardar script:** Ctrl + S (o Cmd + S).

Editor de Scripts (Source)

- **Espacio de trabajo:** Donde escribimos y guardamos nuestro código R (.R files).
- **Ventajas:** Permite organizar, documentar y reproducir tu trabajo.
- **Atajos para ejecutar:**
 - Windows/Linux: Ctrl + Enter (ejecuta línea actual o selección).
 - macOS: Cmd + Enter.
- **Guardar script:** Ctrl + S (o Cmd + S).
 - Sugerencia: Nombrar descriptivamente (ej. analisis_ventas.R).

Entorno (Environment) / Historial (History)

- **Environment:** Muestra todas las **variables y funciones** activas en tu sesión de R.

Entorno (Environment) / Historial (History)

- **Environment:** Muestra todas las **variables y funciones** activas en tu sesión de R.
- `ls()`: Lista todos los objetos en el entorno global.

Entorno (Environment) / Historial (History)

- **Environment:** Muestra todas las **variables y funciones** activas en tu sesión de R.
- `ls()`: Lista todos los objetos en el entorno global.
- `rm(x)`: Elimina el objeto `x`.

Entorno (Environment) / Historial (History)

- **Environment:** Muestra todas las **variables y funciones** activas en tu sesión de R.
- `ls()`: Lista todos los objetos en el entorno global.
- `rm(x)`: Elimina el objeto x.
- `remove(list = ls())`: **Limpia todo** el espacio de trabajo.

Entorno (Environment) / Historial (History)

- **Environment:** Muestra todas las **variables y funciones** activas en tu sesión de R.
- `ls()`: Lista todos los objetos en el entorno global.
- `rm(x)`: Elimina el objeto x.
- `remove(list = ls())`: **Limpia todo** el espacio de trabajo.
- **History:** Almacena un registro de **todos los comandos** ejecutados.

Entorno (Environment) / Historial (History)

- **Environment:** Muestra todas las **variables y funciones** activas en tu sesión de R.
- `ls()`: Lista todos los objetos en el entorno global.
- `rm(x)`: Elimina el objeto x.
- `remove(list = ls())`: **Limpia todo** el espacio de trabajo.
- **History:** Almacena un registro de **todos los comandos** ejecutados.
- Puedes hacer clic para volver a ejecutar o copiar al script.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.
 - Botones para exportar (PNG, PDF) y navegar por el historial de gráficos.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.
 - Botones para exportar (PNG, PDF) y navegar por el historial de gráficos.
- **Packages:** Lista de paquetes **instalados**.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.
 - Botones para exportar (PNG, PDF) y navegar por el historial de gráficos.
- **Packages:** Lista de paquetes **instalados**.
 - Casillas para **cargar/descargar** paquetes.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.
 - Botones para exportar (PNG, PDF) y navegar por el historial de gráficos.
- **Packages:** Lista de paquetes **instalados**.
 - Casillas para **cargar/descargar** paquetes.
 - Botones para **instalar** nuevos paquetes (CRAN, GitHub).

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.
 - Botones para exportar (PNG, PDF) y navegar por el historial de gráficos.
- **Packages:** Lista de paquetes **instalados**.
 - Casillas para **cargar/descargar** paquetes.
 - Botones para **instalar** nuevos paquetes (CRAN, GitHub).
- **Help:** Buscador de **documentación interna** de R y paquetes.

Otros Paneles: Archivos, Gráficos, Paquetes, Ayuda

- **Files:** Navegador de carpetas. Permite abrir scripts, CSV, etc.
- **Plots:** Muestra los **gráficos** generados.
 - Botones para exportar (PNG, PDF) y navegar por el historial de gráficos.
- **Packages:** Lista de paquetes **instalados**.
 - Casillas para **cargar/descargar** paquetes.
 - Botones para **instalar** nuevos paquetes (CRAN, GitHub).
- **Help:** Buscador de **documentación interna** de R y paquetes.
 - Ej: Escribir `mean` y presionar Enter para ver la ayuda de la función `mean()`.

Configuración de Preferencias

- Menú Tools → Global Options...

Configuración de Preferencias

- **Menú Tools** → **Global Options...**
- **Appearance:** Tema (claro/oscuro), tamaño de fuente, tipografía.

Configuración de Preferencias

- **Menú Tools** → **Global Options...**
- **Appearance:** Tema (claro/oscuro), tamaño de fuente, tipografía.
- **Code:** Autocompletado, sangrías automáticas, estilo de código.

Configuración de Preferencias

- **Menú Tools** → **Global Options...**
- **Appearance:** Tema (claro/oscuro), tamaño de fuente, tipografía.
- **Code:** Autocompletado, sangrías automáticas, estilo de código.
- **Keyboard Shortcuts:** Personalizar atajos.

Configuración de Preferencias

- **Menú Tools** → **Global Options...**
- **Appearance:** Tema (claro/oscuro), tamaño de fuente, tipografía.
- **Code:** Autocompletado, sangrías automáticas, estilo de código.
- **Keyboard Shortcuts:** Personalizar atajos.
- **Pane Layout:** Reorganizar la posición de los paneles.

Configuración de Preferencias

- **Menú Tools** → **Global Options...**
- **Appearance:** Tema (claro/oscuro), tamaño de fuente, tipografía.
- **Code:** Autocompletado, sangrías automáticas, estilo de código.
- **Keyboard Shortcuts:** Personalizar atajos.
- **Pane Layout:** Reorganizar la posición de los paneles.
- **Git/SVN:** Configurar control de versiones (si tienes Git instalado).

Directorio de Trabajo

- **Concepto:** La carpeta donde R buscará archivos y guardará resultados por defecto.

Directorio de Trabajo

- **Concepto:** La carpeta donde R buscará archivos y guardará resultados por defecto.
- **Comandos en consola:**

Directorio de Trabajo

- **Concepto:** La carpeta donde R buscará archivos y guardará resultados por defecto.
- **Comandos en consola:**
 - `getwd()`: Muestra el directorio actual.

Directorio de Trabajo

- **Concepto:** La carpeta donde R buscará archivos y guardará resultados por defecto.
- **Comandos en consola:**
 - `getwd()`: Muestra el directorio actual.
 - `setwd("ruta/a/mi/carpeta")`: Cambia el directorio de trabajo.

Directorio de Trabajo

- **Concepto:** La carpeta donde R buscará archivos y guardará resultados por defecto.
- **Comandos en consola:**
 - `getwd()`: Muestra el directorio actual.
 - `setwd("ruta/a/mi/carpeta")`: Cambia el directorio de trabajo.
- **Recomendación:** Usar **Proyectos de RStudio** para manejar automáticamente el directorio de trabajo.

Creación y Manejo de Proyectos

- **Propósito:** Aislar tu trabajo y organizar tus archivos.

Creación y Manejo de Proyectos

- **Propósito:** Aislar tu trabajo y organizar tus archivos.
- **Ventajas:**

Creación y Manejo de Proyectos

- **Propósito:** Aislar tu trabajo y organizar tus archivos.
- **Ventajas:**
 - **Rutas relativas:** Evita problemas con rutas absolutas.

Creación y Manejo de Proyectos

- **Propósito:** Aislar tu trabajo y organizar tus archivos.
- **Ventajas:**
 - **Rutas relativas:** Evita problemas con rutas absolutas.
- **Entorno limpio:** Variables y configuraciones específicas del proyecto.

Creación y Manejo de Proyectos

- **Propósito:** Aislar tu trabajo y organizar tus archivos.
- **Ventajas:**
 - **Rutas relativas:** Evita problemas con rutas absolutas.
- **Entorno limpio:** Variables y configuraciones específicas del proyecto.
- **Integración con Git:** Facilita el control de versiones.

Creación y Manejo de Proyectos

- **Propósito:** Aislar tu trabajo y organizar tus archivos.
- **Ventajas:**
 - **Rutas relativas:** Evita problemas con rutas absolutas.
- **Entorno limpio:** Variables y configuraciones específicas del proyecto.
- **Integración con Git:** Facilita el control de versiones.
- **Cómo crear:** File → New Project... → New Directory → New Project (o usar una carpeta existente).

3. Primeros Pasos en la Consola

Algunas operaciones Aritméticas Básicas

- `Suma > 5 + 3`

3. Primeros Pasos en la Consola

Algunas operaciones Aritméticas Básicas

- **Suma** $> 5 + 3$
- **Resta** $> 10 - 4$

3. Primeros Pasos en la Consola

Algunas operaciones Aritméticas Básicas

- **Suma** $> 5 + 3$
- **Resta** $> 10 - 4$
- **Multiplicacion** $> 7 * 2$

3. Primeros Pasos en la Consola

Algunas operaciones Aritméticas Básicas

- **Suma** $> 5 + 3$
- **Resta** $> 10 - 4$
- **Multiplicacion** $> 7 * 2$
- **Division** $> 9 / 3$

3. Primeros Pasos en la Consola

Algunas operaciones Aritméticas Básicas

- **Suma** > $5 + 3$
- **Resta** > $10 - 4$
- **Multiplicacion** > $7 * 2$
- **Division** > $9 / 3$
- **Potencia** > $2 ^ 3$

3. Primeros Pasos en la Consola

Algunas operaciones Aritméticas Básicas

- **Suma** > $5 + 3$
- **Resta** > $10 - 4$
- **Multiplicacion** > $7 * 2$
- **Division** > $9 / 3$
- **Potencia** > $2 ^ 3$
- **Raiz Cuadrada** > $\text{sqrt}(4)$

Asignación de Variables

- Sintaxis preferida: `nombre <- valor`

Asignación de Variables

- Sintaxis preferida: `nombre <- valor`
- También válido: `nombre = valor` (pero `<-` es más común y claro para asignación).

Asignación de Variables

- Sintaxis preferida: nombre <- valor
- También válido: nombre = valor (pero <- es más común y claro para asignación).
- Asignación con '<-' (preferido)

Asignación de Variables

- Sintaxis preferida: nombre <- valor
- También válido: nombre = valor (pero <- es más común y claro para asignación).
- Asignación con '<-' (preferido)
 - a <- 15

Asignación de Variables

- Sintaxis preferida: nombre <- valor
- También válido: nombre = valor (pero <- es más común y claro para asignación).
- Asignación con '<-' (preferido)
 - a <- 15
- Asignación con '=' (válido)

Asignación de Variables

- Sintaxis preferida: nombre <- valor
- También válido: nombre = valor (pero <- es más común y claro para asignación).
- Asignación con '<-' (preferido)
 - a <- 15
- Asignación con '=' (válido)
 - b = 4

Asignación de Variables

- **Sintaxis preferida: nombre <- valor**
- **También válido: nombre = valor (pero <- es más común y claro para asignación).**
- **Asignación con '<-' (preferido)**
 - `a <- 15`
- **Asignación con '=' (válido)**
 - `b = 4`
- **Operaciones con variables**

Asignación de Variables

- **Sintaxis preferida:** nombre <- valor
- **También válido:** nombre = valor (pero <- es más común y claro para asignación).
- **Asignación con '<-' (preferido)**
 - a <- 15
- **Asignación con '=' (válido)**
 - b = 4
- **Operaciones con variables**
 - c <- a + b

Asignación de Variables

- **Sintaxis preferida:** nombre <- valor
- **También válido:** nombre = valor (pero <- es más común y claro para asignación).
- **Asignación con '<-' (preferido)**
 - a <- 15
- **Asignación con '=' (válido)**
 - b = 4
- **Operaciones con variables**
 - c <- a + b
- **Diferencia:** <- es siempre para asignación. = se usa en funciones para argumentos nombrados (ej. mean(x = c(1,2,3))).

Ver Contenido y Estructura de Variables

- `ls()`: Lista nombres de todos los objetos en el entorno.

Ver Contenido y Estructura de Variables

- `ls()`: Lista nombres de todos los objetos en el entorno.
- `print(x)`: Muestra el valor de `x` explícitamente (escribir `x` en consola también lo muestra).

Ver Contenido y Estructura de Variables

- `ls()`: Lista nombres de todos los objetos en el entorno.
- `print(x)`: Muestra el valor de `x` explícitamente (escribir `x` en consola también lo muestra).
- `str(objeto)`: Muestra la estructura interna (tipo, dimensiones, ejemplo de contenido).

Ver Contenido y Estructura de Variables

- `ls()`: Lista nombres de todos los objetos en el entorno.
- `print(x)`: Muestra el valor de `x` explícitamente (escribir `x` en consola también lo muestra).
- `str(objeto)`: Muestra la estructura interna (tipo, dimensiones, ejemplo de contenido).
- `class(objeto)`: Devuelve la clase del objeto (ej. “numeric”, “data.frame”).

Ver Contenido y Estructura de Variables

- `ls()`: Lista nombres de todos los objetos en el entorno.
- `print(x)`: Muestra el valor de `x` explícitamente (escribir `x` en consola también lo muestra).
- `str(objeto)`: Muestra la estructura interna (tipo, dimensiones, ejemplo de contenido).
- `class(objeto)`: Devuelve la clase del objeto (ej. “numeric”, “data.frame”).
- `length(vector)`: Número de elementos en un vector.

Ver Contenido y Estructura de Variables

- `ls()`: Lista nombres de todos los objetos en el entorno.
- `print(x)`: Muestra el valor de `x` explícitamente (escribir `x` en consola también lo muestra).
- `str(objeto)`: Muestra la estructura interna (tipo, dimensiones, ejemplo de contenido).
- `class(objeto)`: Devuelve la clase del objeto (ej. “numeric”, “data.frame”).
- `length(vector)`: Número de elementos en un vector.
- `dim(matriz_o_df)`: Devuelve filas y columnas de una matriz o data frame.

Comentarios en Código

- Todo lo que sigue al símbolo `#` en una línea es ignorado por R.

Comentarios en Código

- Todo lo que sigue al símbolo `#` en una línea es ignorado por R.
- Crucial para documentar y explicar tu código.

Comentarios en Código

- Todo lo que sigue al símbolo `#` en una línea es ignorado por R.
- Crucial para documentar y explicar tu código.
- `#` Esto es un comentario: el texto aquí no se ejecuta.

Comentarios en Código

- Todo lo que sigue al símbolo `#` en una línea es ignorado por R.
- Crucial para documentar y explicar tu código.
- `#` Esto es un comentario: el texto aquí no se ejecuta.
- `x <- 10 #` Esto es un comentario en la misma línea que el código.

Comentarios en Código

- Todo lo que sigue al símbolo `#` en una línea es ignorado por R.
- Crucial para documentar y explicar tu código.
- `#` Esto es un comentario: el texto aquí no se ejecuta.
- `x <- 10 #` Esto es un comentario en la misma línea que el código.
- `y <- x * 2 #` Calculamos el doble de x

Comentarios en Código

- Todo lo que sigue al símbolo `#` en una línea es ignorado por R.
- Crucial para documentar y explicar tu código.
- `#` Esto es un comentario: el texto aquí no se ejecuta.
- `x <- 10 #` Esto es un comentario en la misma línea que el código.
- `y <- x * 2 #` Calculamos el doble de x
- `print(y) #` Mostramos el resultado

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R
- Este script calcula el producto de dos números

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R
- Este script calcula el producto de dos números
- `numero1 <- 8`

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R
- Este script calcula el producto de dos números
- `numero1 <- 8`
- `numero2 <- 6`

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R
- Este script calcula el producto de dos números
- `numero1 <- 8`
- `numero2 <- 6`
- `resultado_producto <- numero1 * numero2`

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R
- Este script calcula el producto de dos números
- `numero1 <- 8`
- `numero2 <- 6`
- `resultado_producto <- numero1 * numero2`
- `print(paste("El producto es:", resultado_producto))`

4. Scripts y Guardado de Código

Crear un Script

- 1. File → New File → R Script.
- 2. Escribe varias líneas de código.
- Mi primer script de R
- Este script calcula el producto de dos números
- `numero1 <- 8`
- `numero2 <- 6`
- `resultado_producto <- numero1 * numero2`
- `print(paste("El producto es:", resultado_producto))`
- 3. Guarda con Ctrl + S (Windows/Linux) o Cmd + S (macOS), dale un nombre (ej. `script_basico.R`).

Ejecutar Fragmentos y Scripts Completos

- Ejecutar fragmentos:

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**
 - Para comandos rápidos, pruebas, o resultados inmediatos.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**
 - Para comandos rápidos, pruebas, o resultados inmediatos.
 - El código se ejecuta al instante y no se guarda automáticamente.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**
 - Para comandos rápidos, pruebas, o resultados inmediatos.
 - El código se ejecuta al instante y no se guarda automáticamente.
- **Source (Editor de Scripts):**

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**
 - Para comandos rápidos, pruebas, o resultados inmediatos.
 - El código se ejecuta al instante y no se guarda automáticamente.
- **Source (Editor de Scripts):**
 - Para almacenar, organizar y documentar tu código.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**
 - Para comandos rápidos, pruebas, o resultados inmediatos.
 - El código se ejecuta al instante y no se guarda automáticamente.
- **Source (Editor de Scripts):**
 - Para almacenar, organizar y documentar tu código.
 - Permite la reproducibilidad de tu análisis.

Ejecutar Fragmentos y Scripts Completos

- **Ejecutar fragmentos:**
 - Selecciona la línea o bloque de líneas.
 - Presiona Ctrl + Enter (o Cmd + Enter).
 - El resultado aparecerá en la Consola.
- **Ejecutar script completo:**
 - Usa el botón Source (arriba del panel de script).
 - O ejecuta source("ruta/a/tu/script_basico.R") en la Consola.
 - Diferencia entre Consola y Source
- **Consola:**
 - Para comandos rápidos, pruebas, o resultados inmediatos.
 - El código se ejecuta al instante y no se guarda automáticamente.
- **Source (Editor de Scripts):**
 - Para almacenar, organizar y documentar tu código.
 - Permite la reproducibilidad de tu análisis.