



Tecnológico  
de Monterrey

# 01. Presentación

**Ciencia de datos para la toma de decisiones I**

Jorge  
Juvenal  
Campos Ferreira

 [juvenal.campos@tec.mx](mailto:juvenal.campos@tec.mx)

# Programa de la clase

- Presentación
- Revisión del temario
- Diagnóstico programación
- Verificación de instalación de R y Python
- Revisión detalles del curso

# Sobre mí



**M.C. JORGE JUVENAL CAMPOS FERREIRA.**

- \* **Analista de datos, México, ¿Cómo vamos? Y Fundación Novagob México**
- \* **Columnista en Atiempo.TV Coahuila**

## Educación Formal:



### Licenciatura:

Ingeniería en Irrigación por la Universidad Autónoma Chapingo. (2009-2014).



### Maestría:

Maestría en Economía por El Colegio de México (2016-2018).

## Contacto:



**GitHub:** JuveCampos



**LinkedIn:** Jorge Juvenal Campos Ferreira



**Twitter:** @JuvenalCamposF



**IG:** juvenalcampos.dataviz



<https://juvenalcampos.com/>

# Experiencia profesional

- **Analista de datos.**
  - CIDE - Laboratorio Nacional de Políticas Públicas.
  - México ¿Cómo vamos?
  - Fundación Novagob México
  - Clave Igualdad
- **Profesor**
  - Periodismo de datos - Maestría en Periodismo del CIDE
  - Tableros en R/Shiny - Datacrunchers
- **Periodista**
  - Columna semanal en Atiempo.TV
  - Escritor y colaborador en Nexos y Animal Político

## Sobre ustedes

- ¿Cómo se llaman?
- ¿De donde son?
- ¿Por que están en esta carrera?
- ¿Que tanto saben de Ciencia de Datos?
- ¿Qué herramientas saben usar?
- ¿Qué esperan de este curso?



# Instalación R y Python

- Vamos a verificar que tengan R y Python instalado.



## Instalar R

<https://cran.r-project.org/>



## Instalar RStudio

[https://posit-co.translate.goog/download/rstudio-desktop/?  
\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://posit-co.translate.goog/download/rstudio-desktop/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

**El curso se va a dar principalmente en R y RStudio, por lo que sí es importante que verifiquen que el programa esté instalado y funcionando.**

# Propósito del curso

## El curso busca que el estudiante:

El objetivo central de esta unidad es que el estudiante sea capaz de **impulsar transformaciones públicas de alto impacto** mediante el uso estratégico de **tecnologías digitales y disruptivas**, evaluando qué herramientas convienen para resolver problemas públicos según sus **costos, beneficios sociales y potencial de cambio**. Además, busca que pueda **diseñar propuestas de política pública resilientes** ante la incertidumbre, sustentadas en análisis **cuantitativos y cualitativos** y en la valoración comparada de alternativas. Finalmente, pretende fortalecer su capacidad para **formular preguntas e hipótesis de investigación** que permitan diagnosticar problemas políticos, económicos y sociales, y para usar la tecnología de forma **consciente y segura**, respetando la **propiedad intelectual** y construyendo una **identidad digital responsable**.

# Propósito del curso

## El curso busca que el estudiante:

El objetivo central de esta unidad es que el estudiante sea capaz de **impulsar transformaciones públicas de alto impacto** mediante el uso estratégico de **tecnologías digitales y disruptivas**, evaluando qué herramientas convienen para resolver problemas públicos según sus **costos, beneficios sociales y potencial de cambio**. Además, busca que pueda **diseñar propuestas de política pública resilientes** ante la incertidumbre, sustentadas en análisis **cuantitativos y cualitativos** y en la valoración comparada de alternativas. Finalmente, pretende fortalecer su capacidad para **formular preguntas e hipótesis de investigación** que permitan diagnosticar problemas políticos, económicos y sociales, y para usar la tecnología de forma **consciente y segura**, respetando la **propiedad intelectual** y construyendo una **identidad digital responsable**.

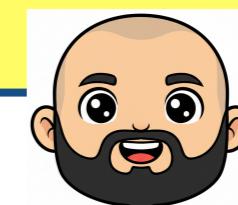
## Lo que yo busco



1. Que **mejoren** sus capacidades de programación en **R**
2. Que **comprendan los fundamentos** básicos de Ciencia de Datos, Estadística y Machine Learning
3. Que **practiquen** la aplicación modelos estadísticos a los problemas que pudieran surgir

# Horarios

## Primer tercio

Lunes	Martes (9:10-10:50)	Miércoles (9:10-10:50)	Jueves	Viernes (9:10-10:50)
				

## Segundo tercio

Lunes	Martes (9:10-10:50)	Miércoles	Jueves	Viernes (9:10-10:50)
				

## Tercer tercio

Lunes	Martes (9:10-10:50)	Miércoles (11:10-12:50)	Jueves	Viernes (9:10-10:50)
				

# Temario y evaluación

Ver Canvas o el Github de la clase:

<https://github.com/JuveCampos/TC2001B.601-Ciencia-de-datos-ene-jun-2026>



# Reto

Por definir



## Reglas de clase

- Mantener el **respeto** hacia compañeros y profesores
- **Participar** de forma ordenada y respetuosa
- Evitar usar **dispositivos electrónicos** para fines ajenos a la clase
- **Traer laptop**, libreta y bolígrafos de colores
- Usar modelos de IA **cuando el profesor lo especifique**

# Examen sorpresa

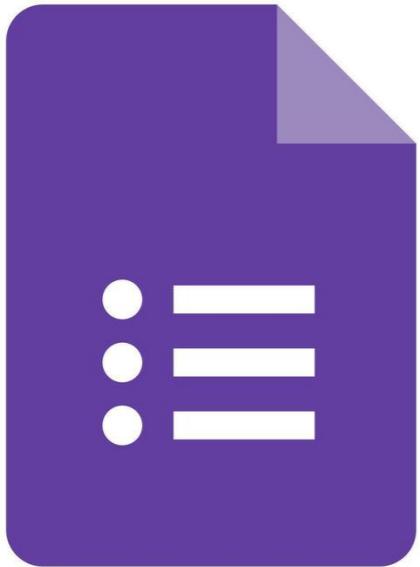
- Mecanismo para evaluar que tanto van aprendiendo.
- Sólo se aplicará en caso de que haya una alta **incidencia de uso de IA** entre los estudiantes.



# **Diagnostico uso herramientas Ciencia de Datos**

# Test diagnóstico

**Ir al siguiente enlace de Google Forms:**



[https://docs.google.com/forms/d/e/  
1FAIpQLSdnw6ZKJPVEwwRYOjp50JRZrFqQauXf3RReiJeZqWOEQYa16A/viewform?  
usp=sharing&oid=103051680334316606510](https://docs.google.com/forms/d/e/1FAIpQLSdnw6ZKJPVEwwRYOjp50JRZrFqQauXf3RReiJeZqWOEQYa16A/viewform?usp=sharing&oid=103051680334316606510)

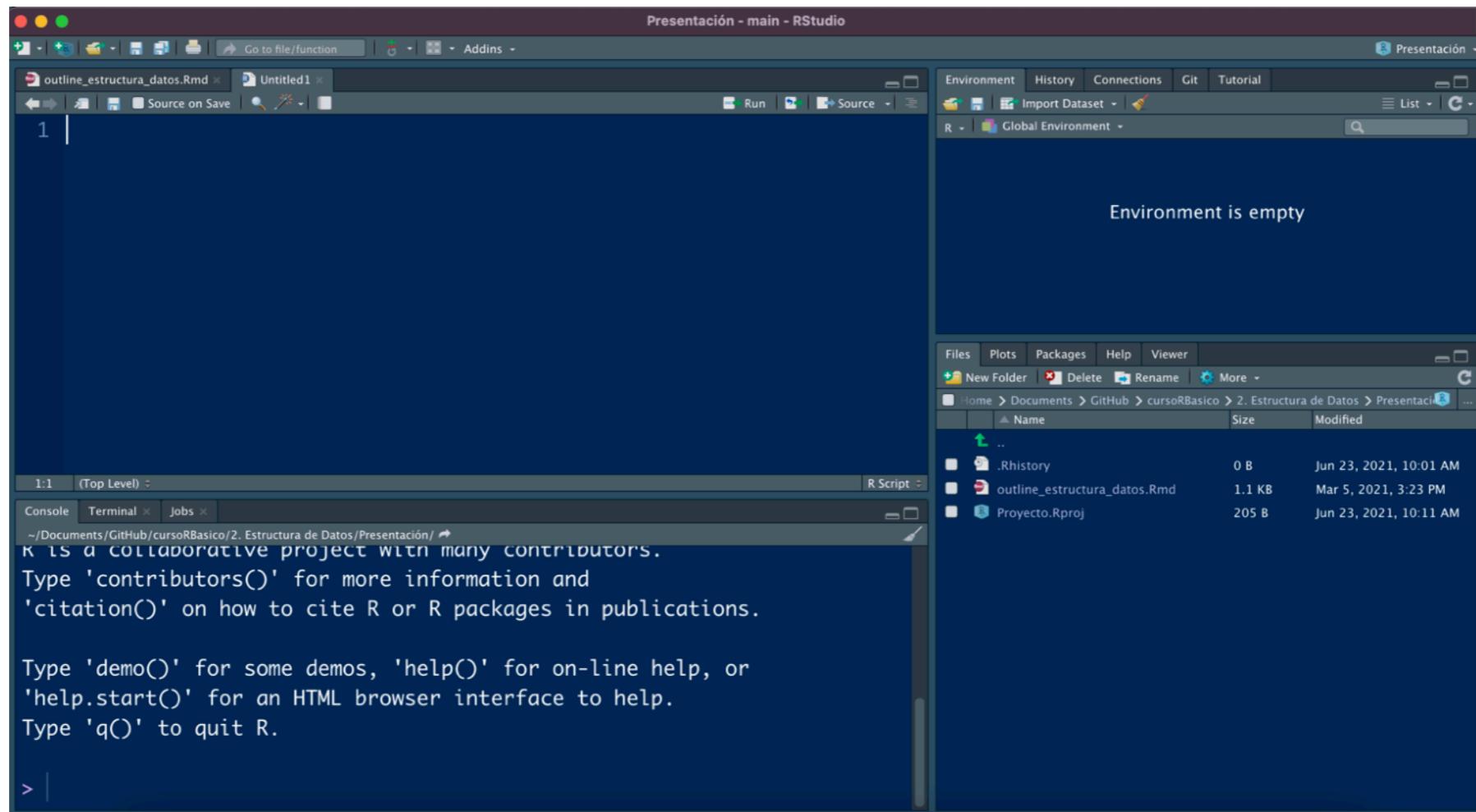
# Instalar R y RStudio

**Instalar R:** <https://www.r-project.org/>

**Instalar RStudio:** <https://posit.co/download/rstudio-desktop/>



RStudio es un programa que provee un entorno de desarrollo (IDE) que nos da las herramientas necesarias para poder programar en R.



# Ventanas



The screenshot shows the R Studio interface with several windows open:

- Editor de texto**: A yellow-highlighted window containing R code for data manipulation. It includes lines 2 through 10 of the script, which loads the tidyverse library and defines variables for names, ages, schools, sex, and knowledge.
- Consola**: A green-highlighted window showing the output of the R code run in the Editor. It displays the definition of the 'sex' variable as a factor with levels M and F, and the creation of the 'sabe\_r' variable.
- Visualizador**: A red-highlighted window showing the "Ambiente" (Environment) tab. It lists variables and their types and values:

Variables	Type	Values
años	num	[1:2] 29...
pers...	chr	[1:2] "J...
sabe...	logi	[1:2] T...
- Presentación - main - RStudio**: The main window title bar.
- Toolbar**: Standard R Studio toolbar with file operations like Open, Save, Print, and Addins.
- Bottom Status Bar**: Shows the current working directory as ~/Documents/GitHub/cursoRBasico/2. Estructura de Datos/Presentación/

# Ventanas



## Editor de texto

Sección del programa en la cual registramos las instrucciones que se van a correr en R. Estas instrucciones se guardan en scripts para volver a ellos más adelante.

Acá se pueden escribir códigos de R, HTML, Python, CSS, Markdown, etc.

The screenshot illustrates the RStudio interface with several colored boxes highlighting different components:

- Editor de texto**: The code editor pane, highlighted in yellow, containing R code for data manipulation and analysis.
- Consola**: The console pane, highlighted in green, showing the output of the R code run in the editor.
- Ambiente**: The environment pane, highlighted in pink, displaying the current global variables and their values.
- Visualizador**: The file browser pane, highlighted in red, showing the project structure and files.

The code in the Editor pane:

```
1 # Librerias ----  
2 library(tidyverse)  
3  
4 # Bases de datos ----  
5 personas <- c("Juvenal", "María")  
6 años <- c(29, 30)  
7 escuelas <- c("Colmex", "UNAM")  
8 sexo <- factor(c("M", "F"),  
9                   levels = c("M", "F"))  
10 sabe_r <- c(TRUE, FALSE)
```

The output in the Consola pane:

```
+ library(tidyverse)  
+ personas  
[1] Juvenal  María  
+ años  
[1] 29 30  
+ escuelas  
[1] Colmex  UNAM  
+ sexo  
[1] M F  
Levels: M F  
+ sabe_r <- c(TRUE, FALSE)
```

# Ventanas



## Consola

Sección en la cual se ejecuta el código que vamos a escribir en el editor de texto.

Igualmente, podemos correr acá código de R que no requerimos guardar para más adelante.

The screenshot shows the RStudio interface with several windows open:

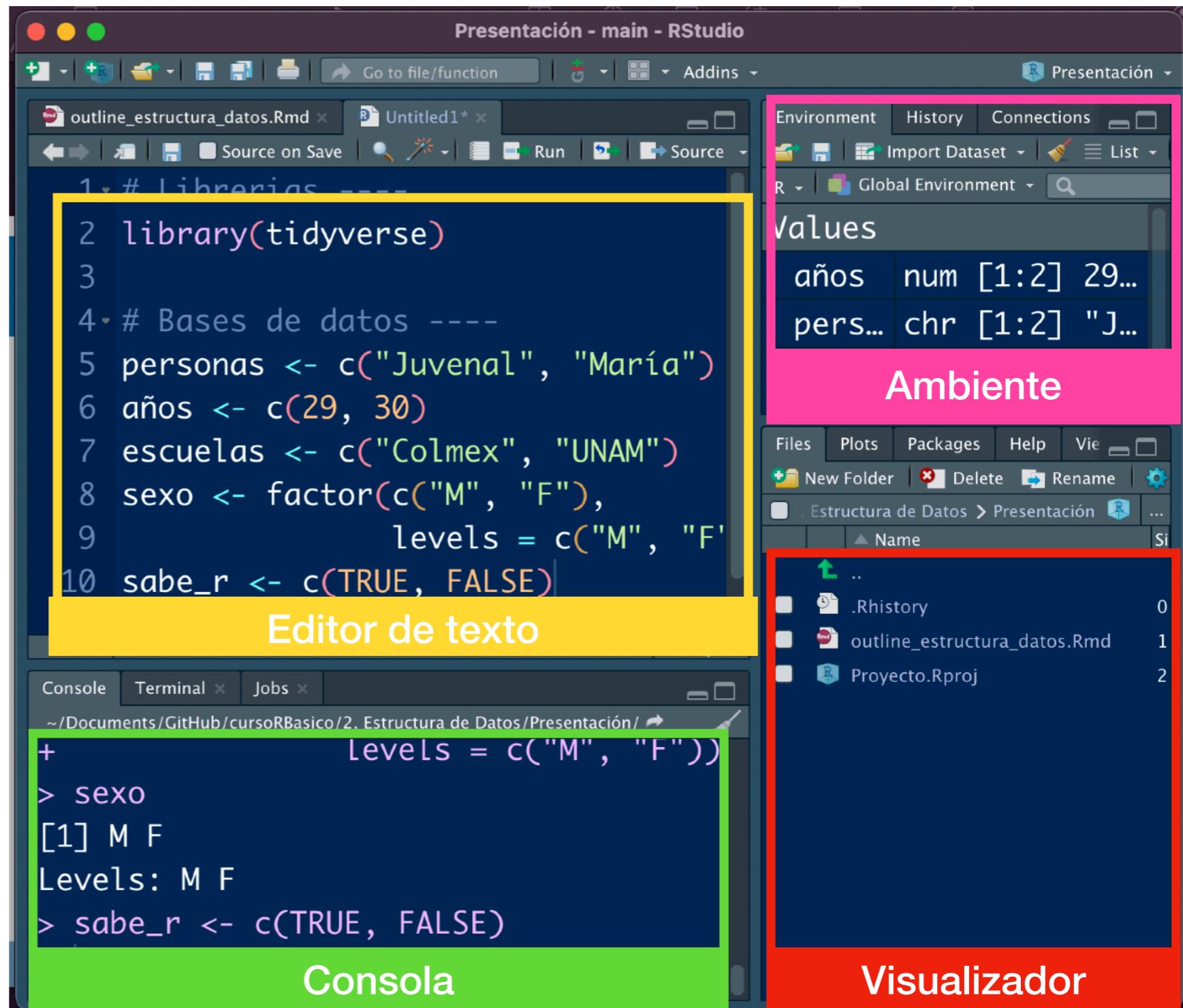
- Editor de texto**: A yellow box highlights the code editor window containing R code for creating variables (personas, años, escuelas, sexo, sabe\_r) and setting levels for the sexo factor.
- Consola**: A green box highlights the console window showing the execution of the R code and its output.
- Environment**: A pink box highlights the environment pane showing the global variables: años, pers..., sabe... (with their respective types and values).
- History**: A pink box highlights the history pane showing the commandLevels = c("M", "F") and its output [1] M F.
- Connections**: A pink box highlights the connections pane.
- Global Environment**: A pink box highlights the global environment pane.
- Ambiente**: A pink box highlights the ambiente pane.
- Visualizador**: A red box highlights the visualizer pane showing the project structure: .., .Rhistory, outline\_estructura\_datos.Rmd, and Proyecto.Rproj.

# Ventanas

## Visualizador

Sección para visualizar cosas:

- 1) Archivos ubicados en nuestro directorio de trabajo
- 2) Gráficas estáticas generadas con ggplot2 o RBase.
- 3) Las librerías instaladas en nuestro RStudio.
- 4) Las visualizaciones web generadas con R.



# Personalización



Para personalizar RStudio  
vamos a  
Tools > Global Options

Podemos configurar:

- como se visualiza el código,
- los colores de las ventanas,
- el tamaño y fuente de las letras,
- el espacio a ocupar del código,
- las cuentas para publicar resultados, etc.

The screenshot shows the 'Global Options' dialog box in RStudio. The left sidebar lists various categories: General, Code, Console, Appearance, Pane Layout, Packages, R Markdown, Sweave, Spelling, Git/SVN, Publishing, Terminal, Accessibility, and Python. The 'General' category is selected and highlighted in blue. The main pane is divided into three tabs at the top: Basic (selected), Graphics, and Advanced. The 'Basic' tab contains several configuration sections:

- R Sessions**: Default working directory (when not in a project): ~/Library/Mobile Documents/com~apple~CloudDocs, with a 'Browse...' button.
- Workspace**:
  - Restore .RData into workspace at startup
  - Restore previously open source documents at startup
- History**:
  - Always save history (even when not saving .RData)
  - Remove duplicate entries in history
- Other**:
  - Wrap around when navigating to previous/next tab
  - Automatically notify me of updates to RStudio
  - Send automated crash reports to RStudio

# R como calculadora

## Operaciones básicas

```
2 + 3  
5 * 4  
10 / 2  
2 ^ 8
```

## Funciones matemáticas

```
sqrt(16)  
abs(-5)  
round(3.7)
```

## Salida en consola:

```
> 2 + 3  
[1] 5  
> 5 * 4  
[1] 20  
> sqrt(16)  
[1] 4
```

# Objetos y funciones



- En R, todo lo que existe es un objeto.
- En R, todo lo que ocurre es una función.



# Objetos y funciones

Los **objetos** son el lugar de la memoria en el cual vamos a guardar información.

Podemos crear nuestros **objetos** o podemos tomar **objetos** hechos por alguna librería.

Los **objetos** son sujetos de ser afectados por las **funciones**.

Las **funciones** son las acciones que le vamos a aplicar a un objeto para obtener un resultado, el cual va a ser un objeto.

Los **objetos** (y las **funciones**) se guardan en el ambiente. El ambiente es el lugar donde se almacenan los **objetos** de la sesión.



# Objetos

Para guardar un **objeto**, utilizamos el operador flechita (`<-`) o el operador igual (`=`).

Si no utilizamos estos operadores, no estamos guardando nada en memoria y, por lo tanto, no lo podremos usar más adelante en nuestro proceso de trabajo.

```
nombres <- c("Joaquín", "María")
sabe.r <- c(TRUE, FALSE)
edad <- c(29, 30)
numero_al_azar <- runif(n = 2, min = 0, max = 10)
datos <- tibble(nombres,
                sabe.r,
                edad,
                numero_al_azar)
```

# Objetos



La sintaxis para guardar un objeto es:

*nombre\_objeto* <- *contenido\_del\_objeto*

El nombre del objeto puede ser cualquiera, tratando de respetar ciertas reglas, como no usar palabras reservadas (como TRUE o FALSE), no empezar con símbolos (. \_ / !, etc.), no empezar con números (1-9) y, de preferencia, no usar símbolos especiales (ñ, 漢字, संस्कृतम्, etc.).

# Objetos



Como vemos en este ejemplo, también se puede guardar como objetos el resultado de funciones; en este caso, estamos guardando el resultado de la función `c()`, de la función `runif()` y de la función `tibble()`.

# Funciones



Las **funciones** son las **acciones** que vamos a realizar sobre los **objetos**. Estas pueden ya estar precargadas de las **librerías base** o pueden provenir de **librerías** descargadas de manera externa.

Las funciones se llaman con la siguiente sintaxis:

```
nombre_funcion(argumento_1 = "argumento_1",
                argumento_2 = "argumento_2",
                ...,
                argumento_n = "argumento_n")
```

Los **argumentos** son como palanquitas a las que hay que moverle para que las funciones funcionen de manera adecuada.



# Funciones

Las **funciones** tienen nombre y apellido. Muchas veces las vamos a encontrar en la literatura de la manera que sigue:

```
dplyr::filter()  
sf::read_sf()  
base::sum()  
leaflet::leaflet()
```

En este caso, lo que va a la izquierda de los ***dos-dos puntos*** es la **librería o paquetería** (apellido) de la cual provienen, mientras que lo que va al lado derecho es el nombre de la función.

Si llamamos la **librería** de origen, meter el apellido ya no va a ser necesario. :3. Si no se le pone apellido, es que proviene de **base**

## Ambiente/entorno global

- El ambiente global en RStudio es el espacio de trabajo donde R guarda todos los objetos que se crean en la sesión
- Aquí viven los objetos que creamos con la `<-`
- En la sesión, vamos a poder acceder a los objetos creados en este ambiente.

# Librerías

## ★ Definición

Las *librerías* son un conjunto de objetos y funciones programados por terceros, que podemos instalar en nuestra sesión de R para potenciar las funciones que podemos realizar.

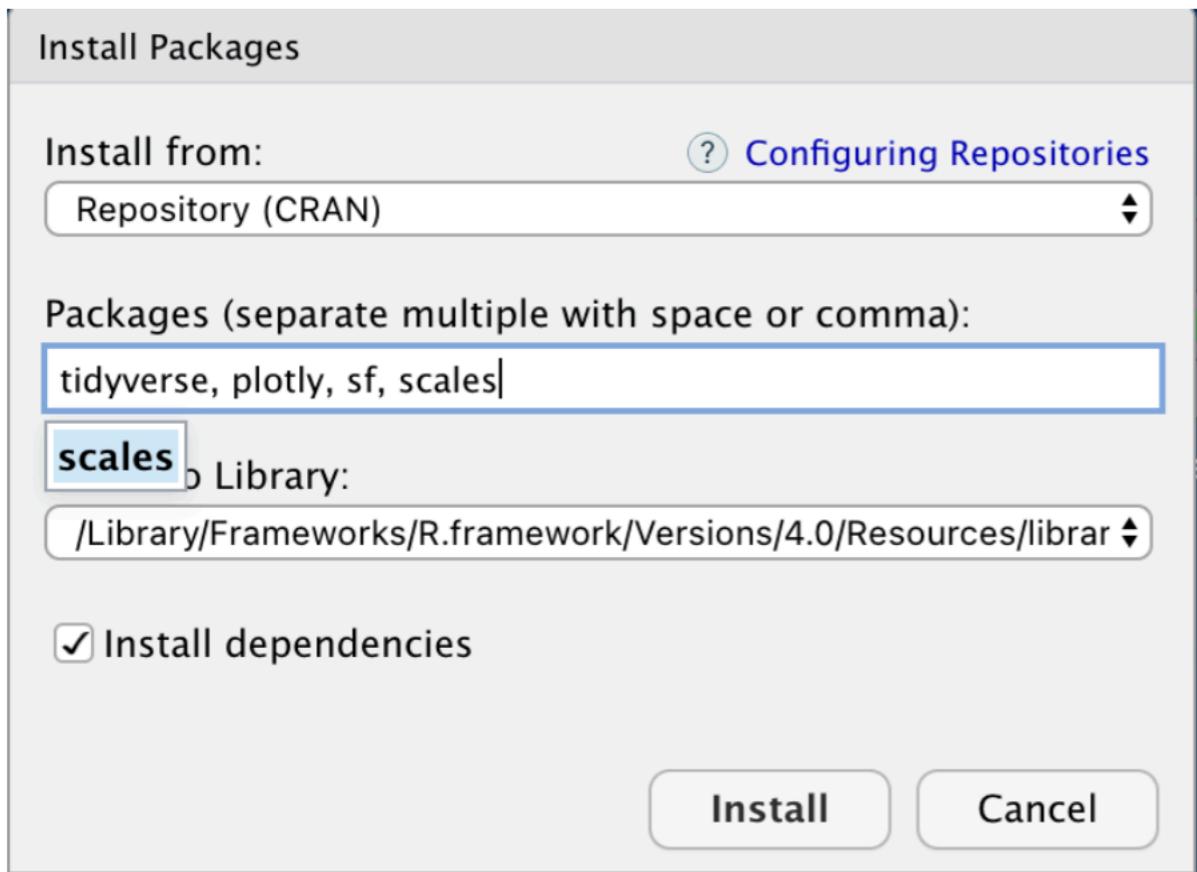
## Instalación

Las instalamos en la sección del **Visualizador**, en la sección de “*Packages*” > *Install* y escribimos la librería que queremos instalar. Otra opción es con la función *install.packages(“librería\_a\_instalar”)*

# Instalar librerías



## Opción 1



## Opción 2

```
> install.packages(c("tidyverse",  
"plotly", "sf", "scales"))
```

# Instalar librerías



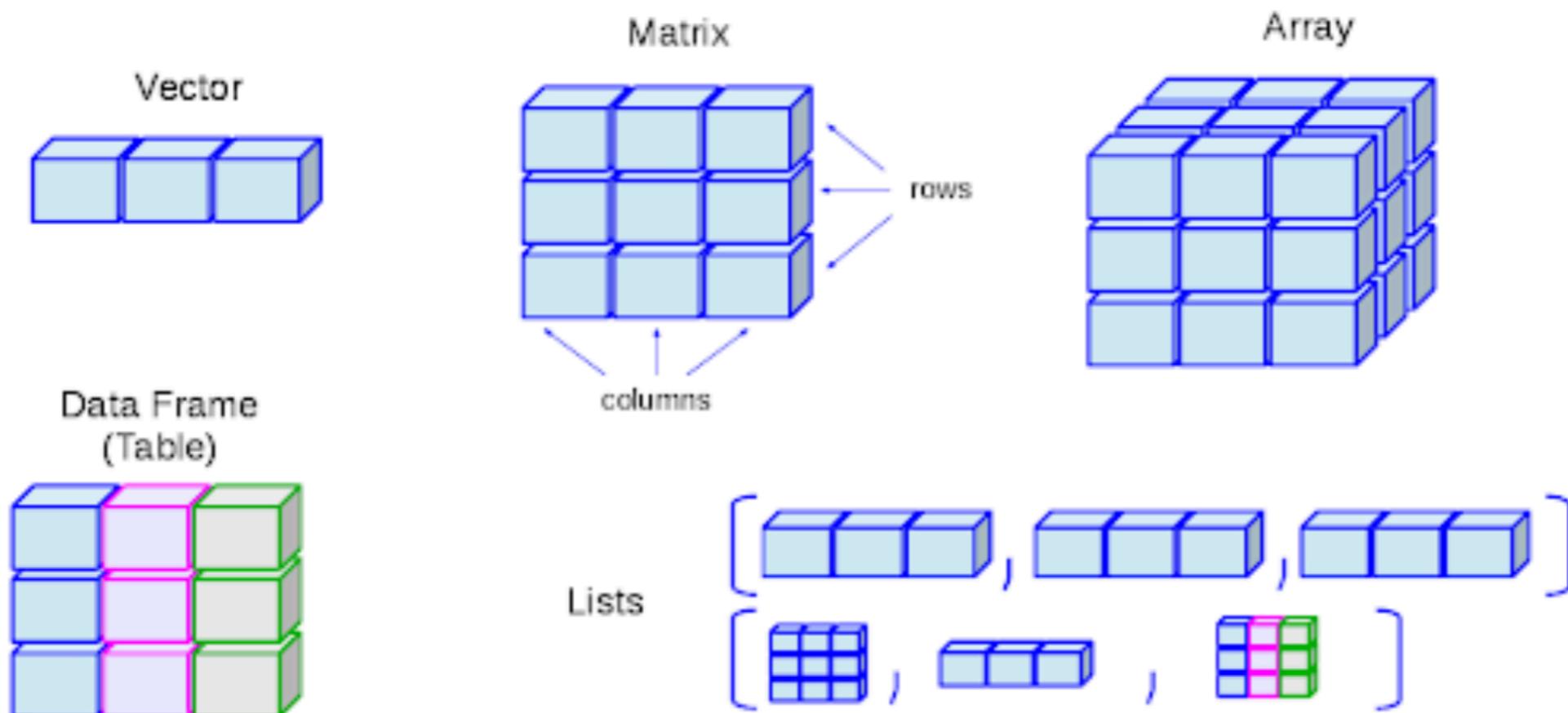
★ Instalar librerías



Llamar librerías



# Estructura de Datos

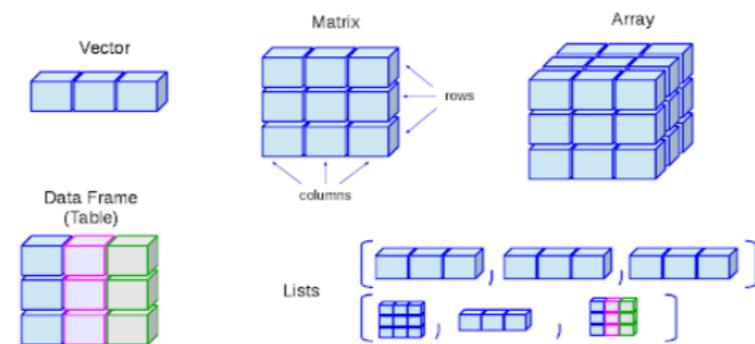


# Estructura de Datos



Los datos suelen agruparse en estructuras básicas de datos:

1. **Escalares**, vectores de 1x1
2. **Vectores**, arreglos de nx1 (n renglones y 1 columna)
3. **Matrices**, arreglos cuadrados de n x n (n renglones y n columnas). Un único tipo de dato.
4. **Arrays**, matrices de matrices multidimensionales n x n x ... x n.
5. **DataFrames**, arreglos cuadrados de n x n (n renglones y n columnas). Múltiples tipos de dato.
6. **Listas**. Contenedores de cualquier cosa.



# Estructura de Datos



Se pueden generar o extraer de muchas formas. Las mas sencillas son las siguientes:

**Vectores:** Usamos la función **combine, `c()`.**

```
vector_1 <- c(1,2,3,4,5)
```

**Matrizes:** Usamos la función **matrix()** y le metemos como ingredientes para esa matriz vectores y argumentos como **ncol** o **nrow** para definir las dimensiones.

```
vector_1 <- c(1,2,3,4,5)
vector_2 <- 10:14
mtx <- matrix(c(vector_1, vector_2), ncol = 2)
```

# Estructura de Datos



Se pueden generar o extraer de muchas formas. Las mas sencillas son las siguientes:

**Dataframes:** Podemos construir dataframes con la función **data.frame()** o con la función **tibble::tibble()**, y le pasamos como ingredientes vectores del mismo tamaño para que sirvan de columnas.

```
df <- data.frame(vector_1,  
                  vector_2,  
                  letras = c("a", "b", "c", "d", "e"))
```

*Este es el tipo de estructura más usado; cuando leemos datos de excel, por ejemplo, nos va a construir automáticamente un DF.*

# Estructura de Datos



Se pueden generar o extraer de muchas formas. Las mas sencillas son las siguientes:

**Listas:** Podemos construir listas con la función **list()**. Como acá le podemos meter dentro la estructura de datos que sea, le metemos como ingredientes lo que sea.

```
lista <- list(df, # Le metemos un df  
             mtx, # le metemos una matriz  
             vector_2, # Le metemos un vector  
             vector_1, # Le metemos otro vector  
             mtcars, # Le metemos otro df pre-construido  
             list(df, vector_2), # Le podemos meter listas  
             sum() # Le podemos meter funciones  
           )
```