

# Curso Primeros Pasos en R

## Clase 1: Introducción a R

**Profesora: Ana María Alvarado**

Pontificia Universidad Católica de Chile

Noviembre 2021

# Clase 1: Introducción a R

---

- Recomendaciones iniciales
- Introducción
- R, RStudio y RStudioCloud
- Directorios de trabajo y proyectos
- Paquetes
- Cálculos
- Actividades prácticas
- Referencias y material complementario

# Recomendaciones Iniciales

# Recomendaciones Iniciales

- Usaremos un script de R compartido para que puedas seguir la ejecución del código. Los scripts se compartirán en la carpeta que está en el siguiente [link](#).
- Usaremos el Chat de Zoom para consultas y comentarios del contenido que se está viendo en la clase.
- Usaremos el foro de la plataforma del curso para consultas y comentarios de contenidos complementarios a la clase
- Participa en el curso a través de las encuestas de zoom, botones de participación y chat.
- Trata de tener más de una pantalla, puede ser tu celular, una tablet o incluso un televisor. Esto te permitirá ver al mismo tiempo las clases y ejecutar el código.

# Introducción

# ¿Qué es Data Science?



# Flujo de Trabajo

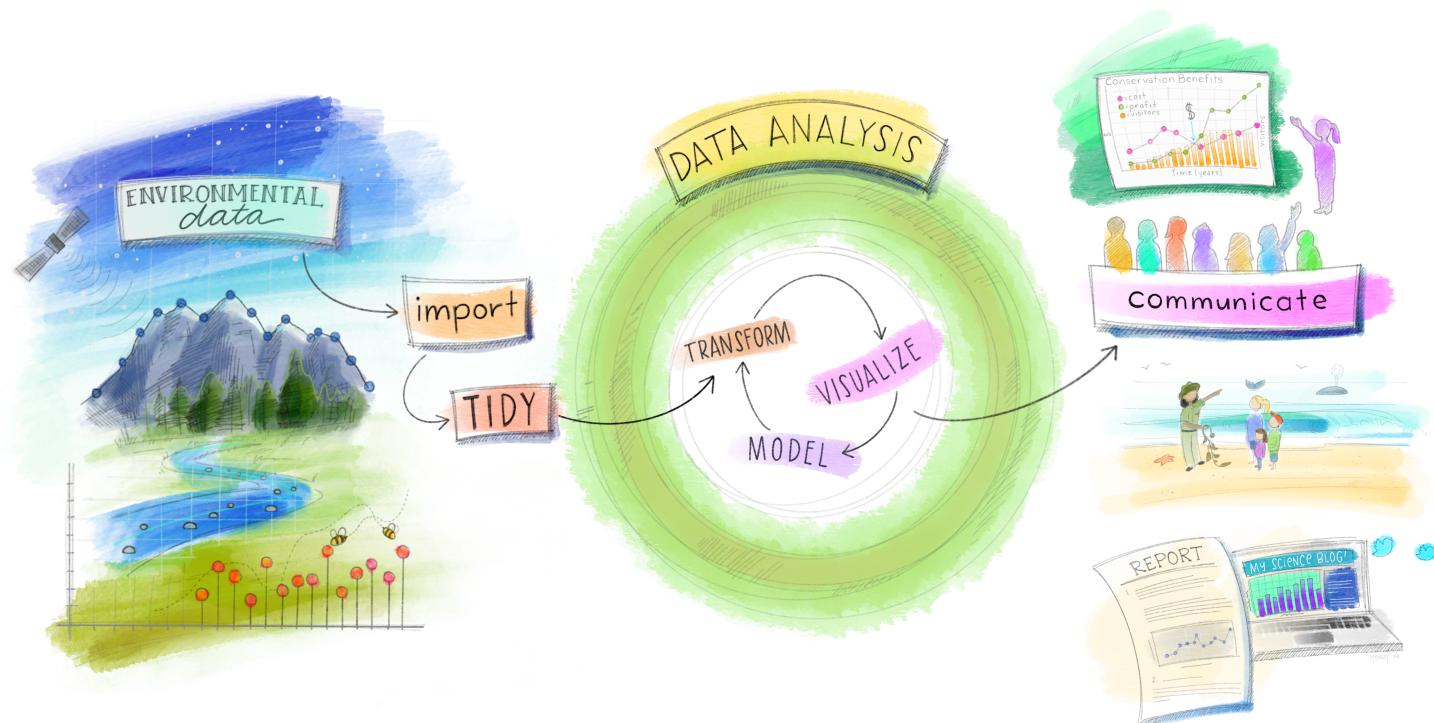
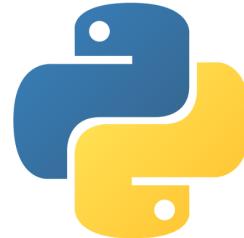
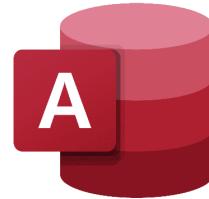


Ilustración por [Allison Horst](#) basada en el esquema de flujo del libro R for Data Science de Grolemund & Wickham.

# Herramientas y software para Data Science



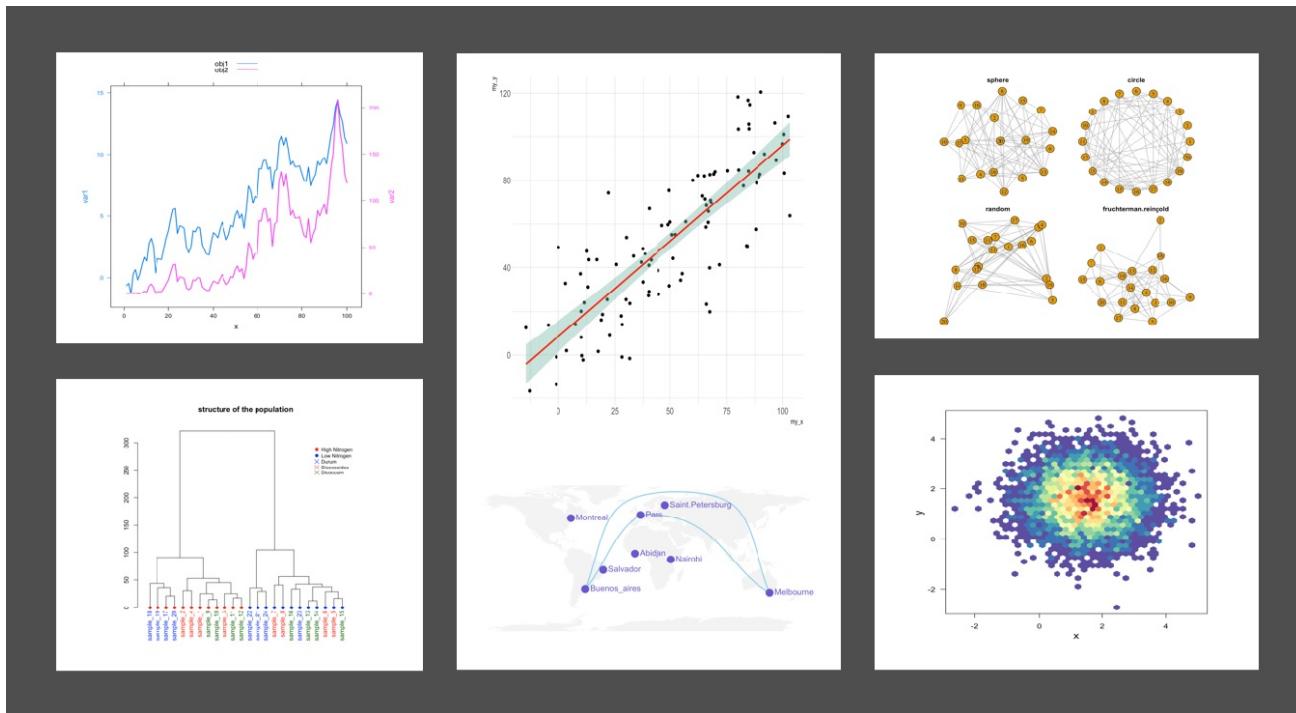
¡Entre otros!

# ¿Por qué R?

- **Gratis y de código abierto.**
- **Herramienta estadística poderosa, hecha por y para estadísticos.**
- **Eficiente y fácil de usar.**
- **Integración con otros sistemas.**
- **Sólida y amplia comunidad mundial.**
- **Muchas más...**

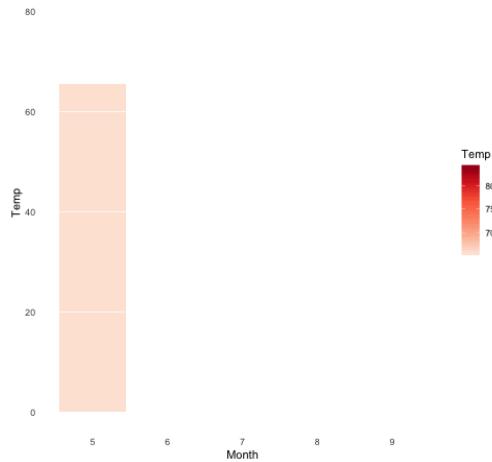
# Ejemplos...

## Modelamiento y análisis estadístico

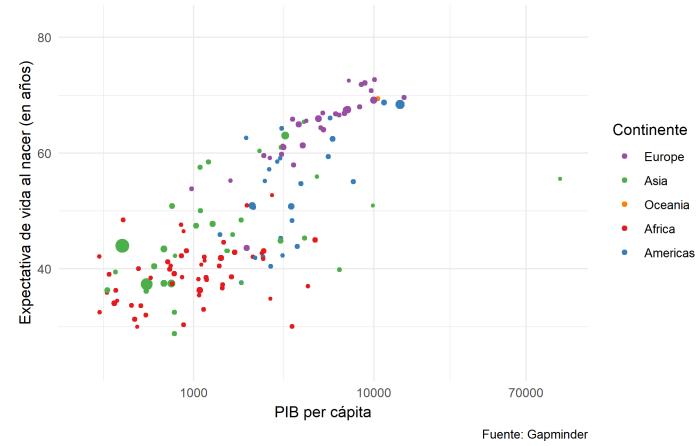


# Ejemplos...

## Gráficos animados

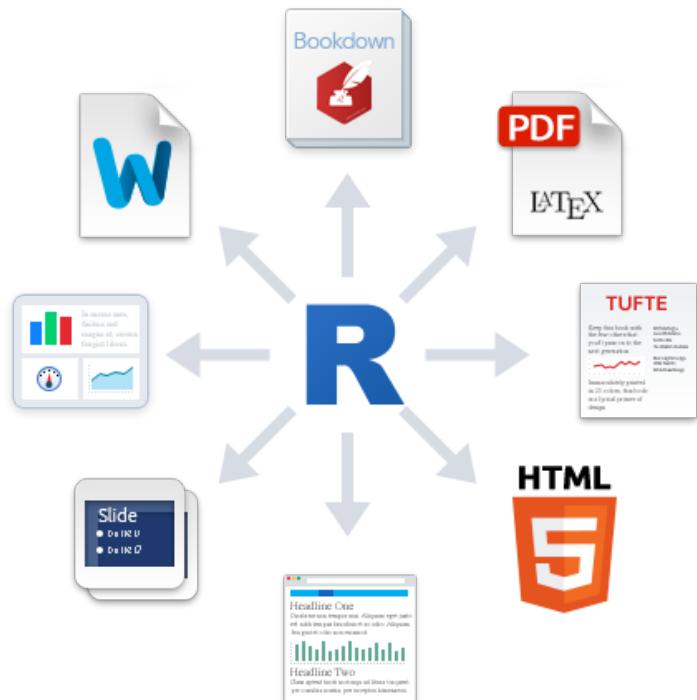


Año: 1952  
Expectativa de vida al nacer según nivel de ingreso



# Ejemplos...

Creación de documentos, presentaciones e informes automatizados.



Más información en [RMarkdown](#).

# Ejemplos...

## Libros y libros electrónicos a partir de documentos de R Markdown.

The screenshot shows a bookdown document interface. On the left, a sidebar displays the table of contents:

- Creación de libros con R Markdown
- Prefacio
  - Por que leer este libro
  - Estructura del libro
  - Información y convenciones del s...
  - Expresiones de gratitud
- Sobre el Autor
- 1 introducción
  - 1.1 Motivación
  - 1.2 Empiece
  - 1.3 Uso
  - 1.4 Dos enfoques de renderizado
  - 1.5 Algunos consejos
- 2 componentes
  - 2.1 Sintaxis de Markdown
    - 2.1.1 Formateo en línea

The main content area displays the first chapter's introduction:

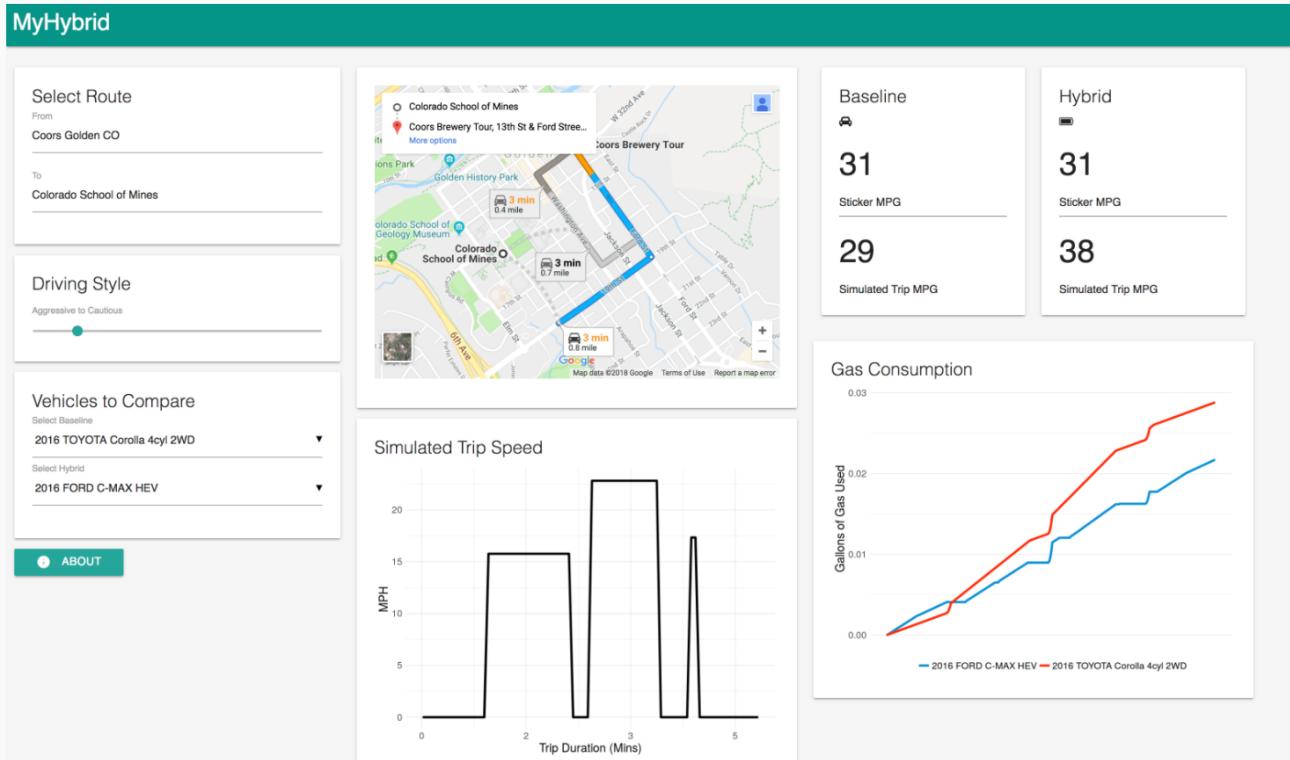
Este breve libro presenta un paquete R, **bookdown**, para cambiar su flujo de trabajo al escribir libros. Debe ser técnicamente fácil escribir un libro, visualmente agradable ver el libro, divertido para interactuar con el libro, conveniente para navegar a través del libro, sencillo para que los lectores contribuyan o dejen comentarios a los autores del libro y, lo que es más importante, los autores no siempre deben distraerse escribiendo detalles.

El paquete **bookdown** se basa en R Markdown (<http://rmarkdown.rstudio.com>) y hereda la simplicidad de la sintaxis de Markdown (puede aprender los conceptos básicos en cinco minutos; consulte la Sección 2.1), así como la posibilidad de múltiples tipos de formatos de salida (PDF / HTML / Word / ...). También ha agregado características como salida HTML de varias páginas, numeración y referencias cruzadas de figuras / tablas / secciones / ecuaciones, inserción de partes / apéndices, e importó el estilo GitBook (

Más información en [bookdown](#).

# Ejemplos...

## Creación de aplicaciones web interactivas



Más ejemplos en el siguiente [Link](#)

# R, RStudio y RStudioCloud

# R y RStudio



R

Es un lenguaje y entorno para computación estadística y gráficos. En este entorno se implementan técnicas estadísticas y se extiende muy fácilmente a través de paquetes. Está disponible como software libre.

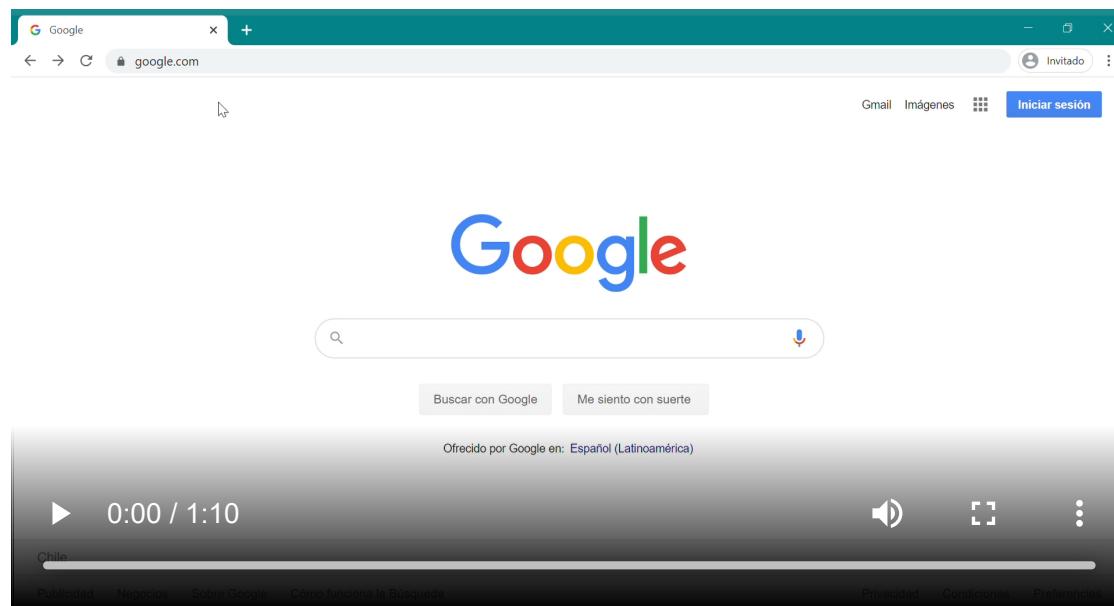


RStudio

Es un entorno de desarrollo integrado (IDE) para R. Incluye varios elementos que facilitan las tareas durante el flujo de trabajo.

# Instalación (Video)

En el siguiente video se puede ver el paso a paso de donde descargar e instalar R y RStudio:

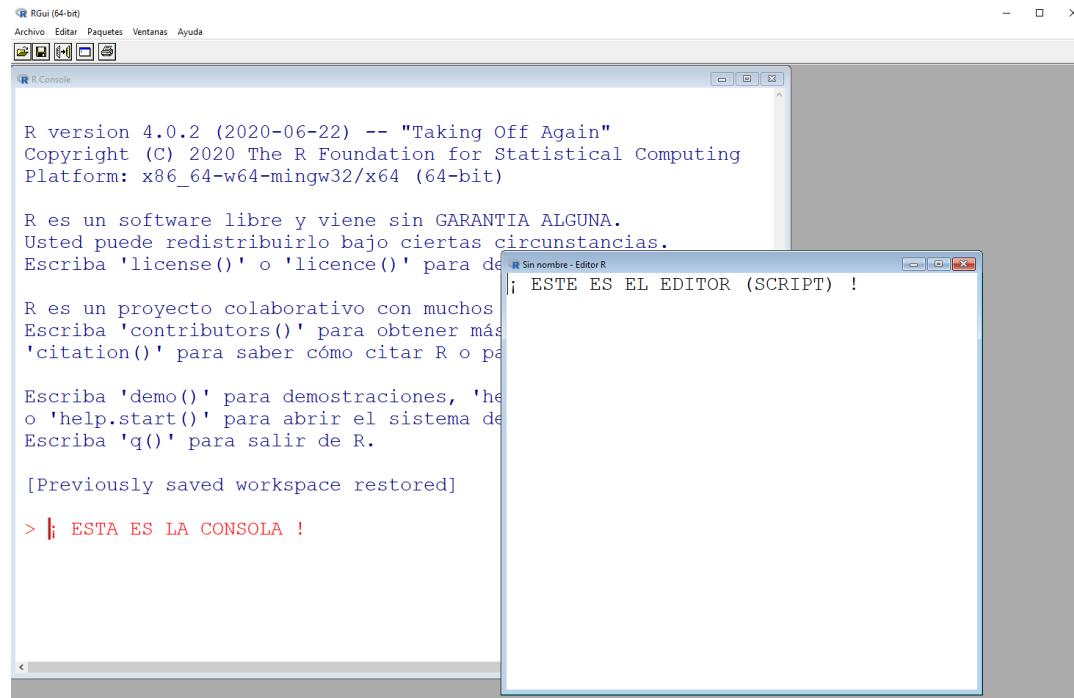


Los links de descargas se pueden encontrar a continuación:

- Descargar R
- Decargar RStudio

# R

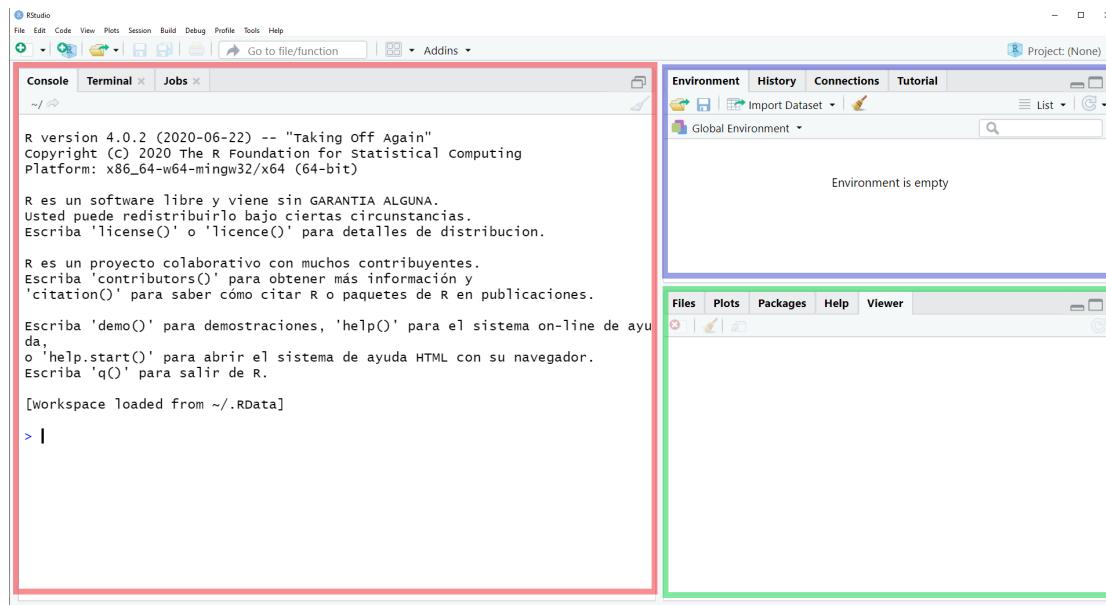
Así se ve R:



Tenemos una ventana que corresponde a la **consola** y otra que es el **editor o script**.

# RStudio

Así se ve RStudio:



Tenemos varios espacios de trabajo dispuestos en 3 paneles:

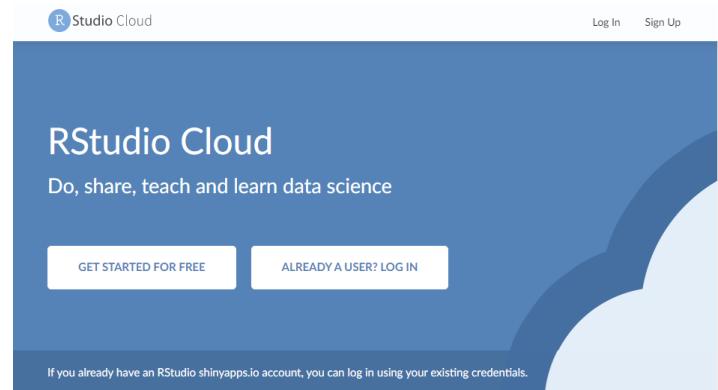
- Consola.
- Environment (ambiente de trabajo).
- Panel de visualización y ayuda.

# RStudioCloud

Si no queremos/podemos instalar R en nuestros equipos, tenemos la opción de usar su versión en la nube, llamada **RStudioCloud**

A diferencia de RStudio, para empezar a trabajar con RStudioCloud necesitamos crear un **proyecto**.

El plan gratuito de RStudioCloud es suficiente para el trabajo que realizaremos en este curso, puedes crear una cuenta accediendo al siguiente **link**



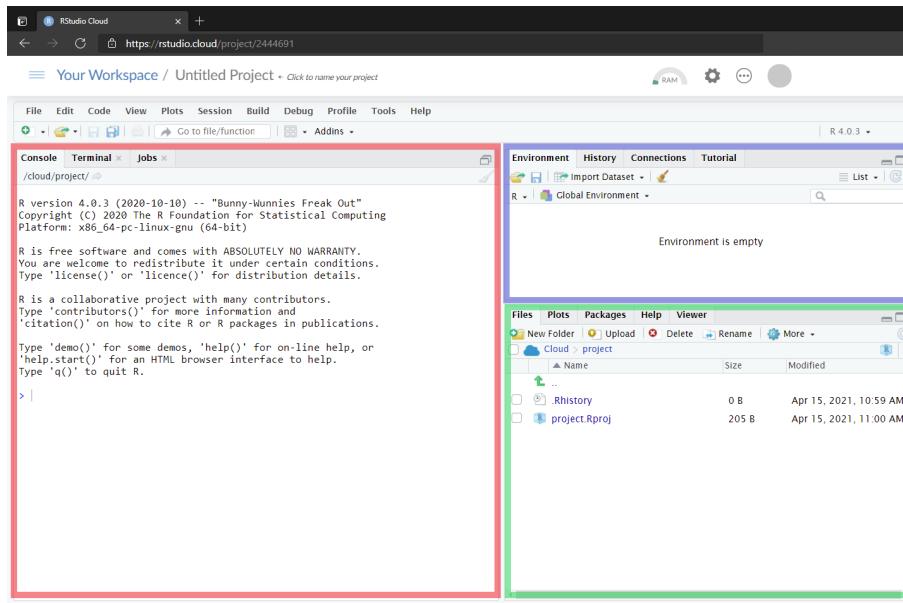
Data science without the hardware hassles

RStudio Cloud is a lightweight, cloud-based solution that allows anyone to do, share, teach and learn data science online.

- Analyze your data using the RStudio IDE, directly from your browser.
- Share projects with your team, class, workshop or the world

# RStudioCloud

Así se ve RStudioCloud, es prácticamente igual a RStudio:

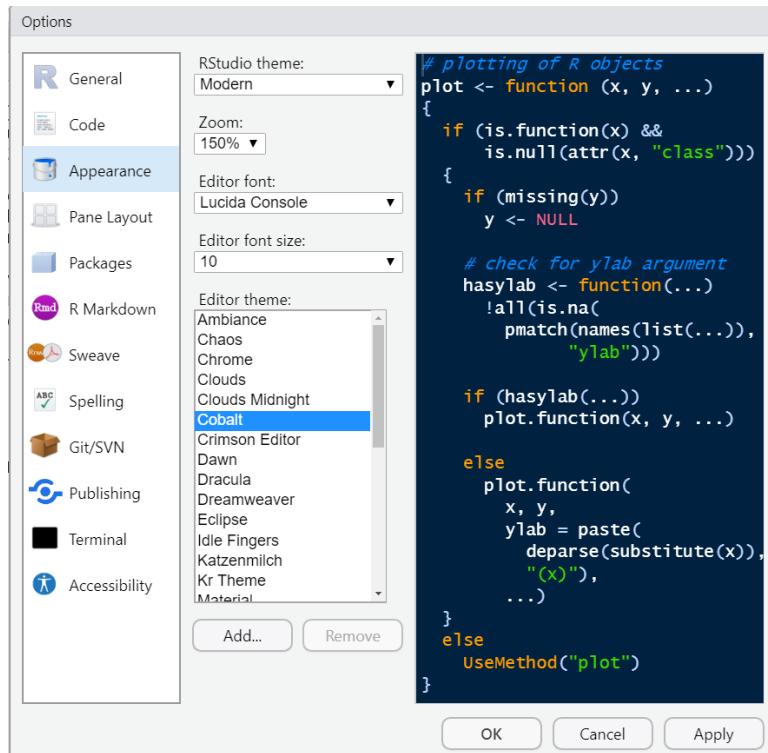


Tenemos varios espacios de trabajo dispuestos en 3 paneles:

- **Consola.**
- **Environment** (ambiente de trabajo).
- **Panel de visualización y ayuda.**

# Personalización

Un aspecto importante de RStudio y RStudioCloud es que podemos personalizar la apariencia de nuestro entorno. Como estaremos expuestos a muchas horas de programación es recomendable cambiar el tema predeterminado a uno oscuro, esto se hace en:



# Actividad práctica

1. Abre RStudio o RStudioCloud.
2. Escribe (Copia y Pega) en la consola el siguiente código y presiona **Enter**

```
print("Hola R")
paste("Hola", "R", sep=" ")
```

Al hacerlo, obtendrás en la consola lo siguiente despues de cada ejecución:

```
## [1] "Hola R"
```

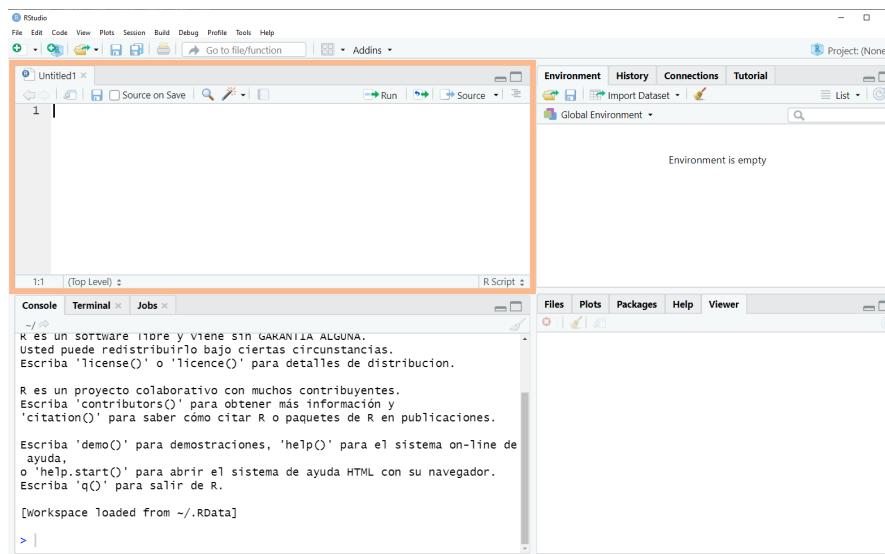
Ahora teclea la secuencia **Ctrl+L / Cmd+L** ¿Qué ocurrió?

# Script (archivo .R)

Un script es un editor de texto en donde escribimos el código que queremos conservar.

Para crear un nuevo script podemos hacer cualquiera de estas opciones:

- Click en **File > New File > R Script**
- Teclear **Ctrl/Cmd + Shift + N**



Ahora tenemos cuatro paneles, el **script** aparece en el bloque de la esquina superior izquierda.

# Script (archivo .R)

Para ejecutar (“correr”) código en el script de R, tenemos distintas opciones:

- **Ctrl/Cmd + Enter** → Ejecutar línea de código/expresión seleccionada
- **Ctrl/Cmd + Shift + N** → Ejecutar todo el código (source)

Para guardar nuestro script podemos teclear **Ctrl + S** o hacer click en el ícono de guardar que está sobre el script, esto guardará un archivo .R con el nombre que le asignemos.

En RStudio tenemos **diagnóstico de errores** en la sintaxis de código y **autocompletado**:

The image shows two panels of the RStudio interface. On the left, the 'Diagnóstico de errores' (Error Diagnostics) panel displays a script with syntax errors. Line 3 contains the assignment 'a <- 5,' which is highlighted with a red error icon. A tooltip box is overlaid on this line, showing the error message 'unexpected token ','' and 'unexpected end of document'. On the right, the 'Autocompletado' (Completion) panel shows a list of available functions starting with 'sum', such as 'sum', 'summary', 'summary.aov', etc., each with its package name in brackets (e.g., {base}, {stats}).

```
1 #esto es un comentario
2
3 a <- 5,
4
5 unexpected token ','
6 unexpected end of document
```

| Completion Suggestion | Package |
|-----------------------|---------|
| sum                   | {base}  |
| summary               | {base}  |
| summary.aov           | {stats} |
| summary.connection    | {base}  |
| summary.data.frame    | {base}  |
| summary.Date          | {base}  |
| summary.default       | {base}  |
| summary.factor        | {base}  |
| summary.glm           | {stats} |

# Comentarios y Secciones

En R podemos comentar líneas de código anteponiendo el símbolo `#`, esto nos permite ordenar nuestro código, hacer descripciones de que estamos ejecutando o guardar código que no queramos ejecutar:

```
### Este es un comentario de R, puedes usarlo para poner títulos!
5+5 ## También para comentar después de operaciones/código!
# print("Diplomado en Data Science UC")
```

Tambien uno puede comentar una o múltiples líneas seleccionando el código y tecleando **Ctrl/Cmd + Shift + C**.

Finalmente, la secuencia **Ctrl/Cmd + R** nos permite seccionar fácilmente nuestro código. Al teclearlo aparecerá una ventana emergente donde podemos ingresar el nombre de la sección.

```
# Sección 1: -----
```

# Actividad práctica

1. Crea un nuevo script
2. Guárdalo con el nombre "Sesion1"
3. Añade una sección con tu nombre (**Ctrl/Cmd + R**)
4. Copia, pega y ejecuta este código en el script:

```
poleras <- c(254,203,182,50)
meses <- c("Ene", "Feb", "Mar", "Abr")
min(poleras)
plot(poleras)
barplot(poleras, names.arg = meses, col="orange")
```

# Directorios de trabajo y proyectos

# Directorios de trabajo

El directorio de trabajo es donde R:

- Busca los archivos que le pedimos leer.
- Coloca los archivos que le pedimos guardar.

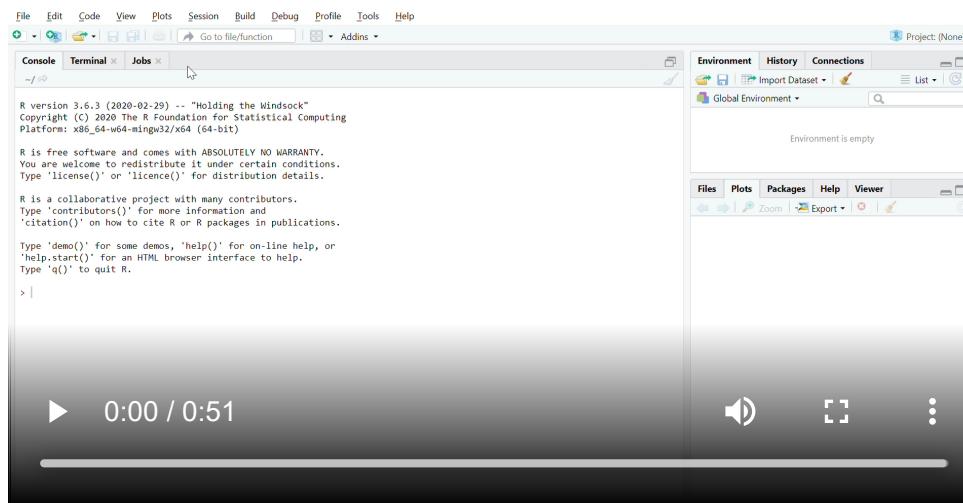
Las rutas y directorios pueden causar complicaciones principalmente porque se escriben de manera diferente según el sistema operativo y porque es muy difícil que alguien más tenga la misma configuración de directorios que nosotros.

Ejecuta el código: `getwd()` para conocer tu directorio de trabajo actual.

# Proyectos de RStudio

Durante nuestro flujo de trabajo lo ideal es mantener todos los archivos relacionados a un análisis en un mismo lugar, es por esto que trabajaremos con los Proyectos de RStudio. Trabajar con proyectos es una buena práctica que nos ahorrará muchos dolores de cabeza.

En el siguiente video se muestra como crear un proyecto:



# Actividad práctica

1. Crea un nuevo proyecto para esta clase, llámalo ProyectoClase1
2. Abre el proyecto y ejecuta el siguiente código `getwd()`
3. Crea un nuevo script y ejecuta el siguiente código:

```
poleras <- c(254, 203, 182, 50)
meses <- c("Ene", "Feb", "Mar", "Abr")
min(poleras)
plot(poleras)
barplot(poleras, names.arg = meses, col="orange")
```

Ahora crea una carpeta dentro de tu proyecto llamada datos y ejecuta el siguiente código.

```
write.csv(cbind(meses, poleras), "datos/poleras.csv")
read.csv("datos/poleras.csv")
```

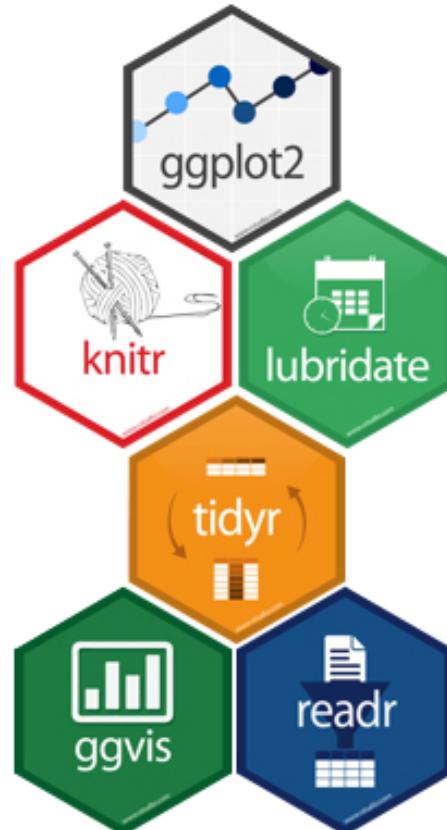
# Paquetes de R

# Paquetes de R

Para hacer efectiva la actualización del software, y basado en el foco académico, R utiliza librerías como conjunto de funciones, datos y documentación que denominaremos paquetes o packages.

Cuando iniciamos R se carga solo un conjunto de funciones, datos y documentación que se conoce como **R Base**.

Para extender las posibilidades de **R Base** tenemos que instalar otros paquetes.



# Paquetes de R

A la fecha de hoy, el total de packages disponibles en el **CRAN** de R es:

**18.365**

# Paquetes de R

Los packages se pueden instalar directamente desde el software o descargarlos de forma manual, esto último se puede hacer desde dos fuentes:

## CRAN

### Comprehensive R Archive Network

Es el repositorio oficial de paquetes de R. Para estar acá los paquetes deben ser aprobados.

```
install.packages("nombrepaquete")
```

Los paquetes se instalan solo una vez en nuestro equipo (pero se deben actualizar cada cierto tiempo).

## GitHub

Plataforma para desarrollo de software. Muchos paquetes solo se comparten por acá. En Github también podemos acceder a la versión en desarrollo de paquetes que están en CRAN.

```
install.packages("remotes")
remotes::install_github("usuario/paquete")
```

# Paquetes de R

Para usar las funciones y datos contenidos en un paquete tenemos que “activarlo” en nuestra sesión de R:

```
library(nombrepaqute)
```

Luego de esto, aparecerán disponibles en nuestra sesión las funciones de ese paquete.

Hay tantos paquetes en R que a veces pueden existir múltiples funciones con el mismo nombre, si tenemos ambos paquetes cargados esto puede causar problemas en nuestro código. Podemos especificar de qué paquete viene una función usando `::` de la siguiente forma:

```
nombrepaqute::nombrefuncion
```

# Actividad Práctica

Instalemos el paquete **ggplot2**

```
install.packages("ggplot2")
```

Para activar el paquete usamos:

```
library(ggplot2)
```

# Cálculos en R

# Cálculos en R

R se puede usar como una calculadora.

Algunas operaciones que podemos realizar son:

| Operación                           |
|-------------------------------------|
| + suma                              |
| - resta                             |
| * multiplicación                    |
| / división                          |
| %% división entera                  |
| ^ potencia                          |
| <code>sqrt()</code> raíz cuadrada   |
| <code>log()</code> logaritmo base e |
| <code>exp()</code> exponencial      |
| <code>abs()</code> Valor Absoluto   |

# Actividad práctica

Calcule en R

$$a. - 2 + 4 \times 5 - e^3$$

$$b. - \log(5) + \frac{\pi}{\sqrt{5}}$$

Solución

```
2 + (4*5) - exp(3)
```

```
## [1] 1.914463
```

```
log(5) + pi/sqrt(5)
```

```
## [1] 3.014401
```

```
]
```

# Referencias y Material Complementario

1. **R para Ciencia de Datos** : Libro en línea, en español.
2. **RStudio cheatsheets** : “Torpedos”; en la parte inferior de la página hay versiones en español disponibles.
3. **RStudio Primers**: Ejercicios interactivos; en inglés.
4. **Hands-on Programming with R** : Libro en línea de introducción a la programación en R que no requiere conocimientos previos; en inglés.

# ¡Gracias!

Ana María Alvarado Celis  
[amalvara@uc.cl](mailto:amalvara@uc.cl)

Esteban Rucan  
[errucan@uc.cl](mailto:errucan@uc.cl)