

# Introducción a R

## Sesión 1: Panorama general de R

Jorge de la Vega Góngora

Instituto Tecnológico Autónomo de México

Sábado 21 de enero de 2023

## 1 Introducción

- ¿Qué es R? Un poco de historia
- Instalación de R y RStudio
  - Instalación de R
  - Instalación de Rstudio
  - Instalación de paquetes
- Primeros pasos
  - Generar un documento reproducible

## 2 Práctica con R

- Cómo se usa R
- Ejemplos de aplicaciones de R

# Introducción

- El lenguaje `s` se desarrolló en Laboratorios AT & T-Bell, principalmente por [John M. Chambers](#), en 1976. Participan también Rick Becker y Allan Wilks.
- Su uso se expandió rápidamente después de la publicación del libro de John Tukey: “Exploratory Data Analysis” (EDA).
- `S-Plus` fue una versión comercial de `s` que inició en 1987. Su popularidad incrementó dramáticamente después de 1990, y se mantuvo hasta la versión 8 en 2007.
- `S-Plus` fue atractivo porque contaba con una interfaz de usuario gráfica (GUI), y soportaba muchos formatos para importación y exportación de datos y gráficas.

## Stanford | Statistics SCHOOL OF HUMANITIES & SCIENCES

[News & Events](#)[People](#)[Academic Programs](#)[Admissions](#)[Industrial Affiliates](#)[Resources](#)[About](#)

### People

[Faculty](#)[Stein Fellows](#)[Research Scientists & Lecturers](#)[Postdocs](#)[Students & Alumni](#)[Staff](#)

### John Chambers

*Adjunct Professor of Statistics*

#### Mailing Address:

Department of Statistics  
Sequoia Hall  
390 Serra Mall  
Stanford University  
Stanford, CA 94305

Email: [jmc@stat.stanford.edu](mailto:jmc@stat.stanford.edu)

- R comenzó su desarrollo en los 90's, realizado por estadísticos como una alternativa de código abierto a S-Plus, en parte porque entonces no había una versión de S-Plus para Linux.
- **Robert Gentleman** y **Ross Ihaka**, de Nueva Zelanda, son los creadores pioneros de R.



- R se basa en el mismo lenguaje S utilizado en la versión 2000 de S-Plus, con algunas excepciones menores. Sin embargo, su arquitectura es un poco diferente.
- Ventajas:
  - bien documentado
  - fácil de obtener e instalar
  - fácil de actualizar sus bibliotecas de funciones
  - Comunidad muy amplia de usuarios

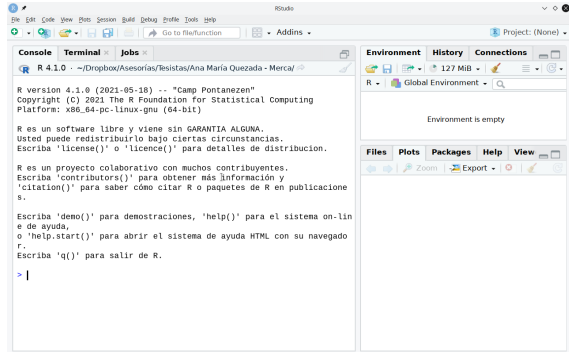
- Algunas desventajas de aquella época:
  - GUI muy básica (hasta la aparición de RStudio)
  - manejo de memoria RAM ineficiente
  - menos capacidades para importar y exportar datos de otros formatos
  - no mucha interacción con Office

Estas desventajas se han ido resolviendo con el tiempo.

RStudio es una herramienta que facilita el trabajo con R, sobre todo en los siguientes puntos:

- Trabaja con R y sus gráficas de manera interactiva
- Permite organizar sus programas y tener organizados varios proyectos
- Permite realizar *investigación reproducible*, que esta tomando relevancia en el mundo científico
- Da mantenimiento a los paquetes instalados en el sistema
- Permite crear y compartir reportes
- Permite el intercambio de programas y la colaboración con otros usuarios
- Se puede llevar un control de proyectos en la nube conectando con github

RStudio es tanto una interfaz gráfica (GUI) como un ambiente de desarrollo integrado (IDE). Hay versiones para Windows, Linux y Mac OS X. También permite correr R desde un servidor remoto.



## ¿Porqué se promueve el uso de R? I

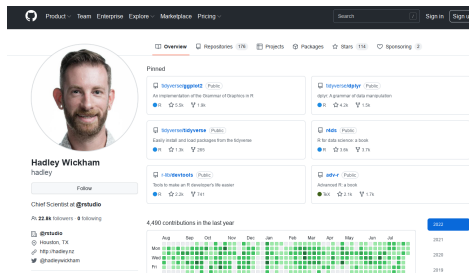
- R es gratuito con un lenguaje poderoso orientado a objetos.
- Permite análisis de datos y gráficas en forma interactiva y reproducible
- Es fácil implementar nuevos métodos y distribuir a otros usuarios
- Su programación y código son abiertos: siempre puedes saber qué estas haciendo
- Investigadores de punta en estadística son programadores de R y cuenta con una muy amplia y extensa comunidad a nivel global
- Millones de recursos (libros, tutoriales, videos, papers, etc.) disponibles de manera gratuita y de fácil acceso. Un Congreso anual de usuarios [UseR!](#).
- R fue planeado para ser extensible
  - Usuarios escriben nuevas funciones, al igual que los desarrolladores.
  - La documentación para agregar funciones es excelente.
  - Las funciones creadas por los usuarios se invocan igual que las internas.
  - Usuarios pueden crear sus propios tipos de datos y agregar atributos, p.ej. comentarios a cada pieza de datos de R.
- R permite trabajar tanto con datos estructurados como no estructurados
- Es un ambiente para análisis estadístico y también un lenguaje de alto nivel: unos cuantos comandos hacen mucho trabajo.



- R permite crear las mejores gráficas científicas.
- En R ya hay varios paquetes, contribuidos por diferentes personas, que permiten aplicar eficiente y correctamente muchos métodos estadísticos.
- Permite concentrarse más en la interpretación de los resultados que en la implementación.

# Desarrollo de R en los últimos años

- La comunidad de usuarios de R ha crecido significativamente en los últimos años, en parte por la popularidad de la ciencia de datos, y en parte por la difusión que se le ha dado por su fácil acceso.
- Hadley Wickham** ha contribuido de manera importante a su difusión, con la elaboración de paquetes (ggplot2, tidyverse, dplyr, lubridate, ...) que facilitan el manejo de datos.



The screenshot shows the GitHub profile of Hadley Wickham. At the top, there's a navigation bar with links for Product, Team, Enterprise, Explore, Marketplace, and Pricing. Below this, the profile header includes a circular profile picture of Hadley Wickham, his name 'Hadley Wickham', and his handle 'hadley'. To the right of the profile picture, there are tabs for Overview, Repositories (176), Projects, Packages, Stars (156), and Sponsoring (3). The 'Overview' tab is selected. Below the header, there's a 'Pinned' section with four repositories: 'tidyverse/ggplot2', 'tidyverse/dplyr', 'tidyverse/tidyverse', and 'tidyverse/nlars'. Each repository card shows the repository name, a brief description, and statistics for R packages (stars, forks, and downloads). Below the pinned repositories, there's a section titled '4,490 contributions in the last year' with a calendar heatmap showing activity from August 2019 to August 2021. The heatmap uses green squares to indicate contributions, with a density of contributions visible across the months.

- También se han creado comunidades que facilitan, que promueven ambientes seguros para los y las programadoras, como **r-ladies** y que organizan seminarios frecuentemente.

R tiene muchos fans, casi fanáticos



# ¿Porqué NO R?

No todo es maravilloso...

- Para utilizar R a su máximo potencial, la curva de aprendizaje es algo elevada.
- Se requiere, eventualmente, saber programar.
- El uso de paquetes está condicionado al menos parcialmente a los caprichos de su creador: puede decidir cuando cambiar cosas, dejar de dar mantenimiento, etc.
- Es posible que en el software/paquetes haya errores aunque usualmente se detectan, pero **no siempre**.
- Muchas empresas no permiten el uso de *Open Source* para sus procesos. Prefieren pagar por “soporte”, (aunque en la práctica no sirva de nada).
- Es lento para algunas aplicaciones por su estructura, aunque hay maneras de hacerlo más eficiente.

Actualmente, un lenguaje prometedor complementario/sustituto es **Julia**



- ➊ Primero hay que instalar R.
- ➋ Instalar RStudio.
- ➌ Instalar los paquetes complementarios que necesitemos para nuestro trabajo desde RStudio. Esto se puede hacer conforme se requiera.
- ➍ Creamos un proyecto de trabajo para cada uno de los temas que queramos trabajar en R. Estos se guardarán en sus respectivos directorios.

- Para Windows Seguir las instrucciones indicadas en <https://cran.r-project.org/bin/windows/>.
- Para Mac: Seguir las instrucciones indicadas en <https://cran.r-project.org/bin/macosx/>.
- En Linux, cada versión puede tener instrucciones diferentes. Para Ubuntu y sus variedades, se puede consultar [esta liga](#).
- Puede haber varias versiones de R instaladas en paralelo (cada una tiene su directorio).

- La versión para cualquiera de los sistemas operativos, se obtiene el ejecutable de:  
<https://www.rstudio.com/products/rstudio/download/>
- Una vez obtenido el archivo, lo ejecutamos y seguimos las instrucciones que se indiquen.

## Ejercicio 1

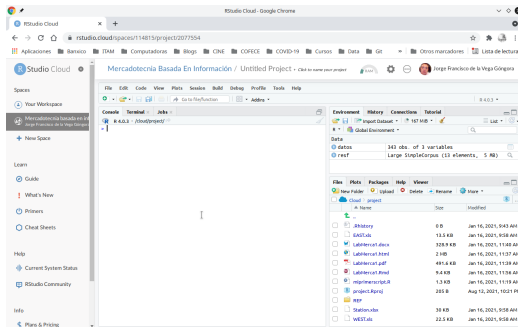
Instalar R y RStudio en sus computadoras personales.

- Hay una gran cantidad de paquetes (*libraries*) que son colecciones de funciones empaquetadas que tienen un propósito definido. Los paquetes se publican en un repositorio conocido como **CRAN** (Comprehensive R Archive Network).
- Algunos paquetes en realidad son colecciones de otros paquetes, o bien algunos paquetes dependen de otros.
- Ejemplos de paquetes útiles que usaremos:
  - `knitr`
  - `rmarkdown`
  - `tidyverse`
  - `dplyr`
  - `forecast`
- Los paquetes se pueden instalar fácilmente desde RStudio a través de menús, o bien directamente desde la consola de R:

```
install.packages(c('knitr', 'rmarkdown', 'tidyverse', 'dplyr', 'forecast'))
```



- Es posible usar una versión de RStudio en línea, a través de <https://rstudio.cloud/>.



- Se puede usar una versión limitada en memoria y espacio de manera gratuita. Para iniciar el aprendizaje puede ser suficiente, pero en proyectos grandes se requiere más espacio.
- La ventaja es que se puede usar sin problemas de instalación y/o configuración. Todo fluye sin problemas.

## Ejercicio 2: Prueba de instalación y primer documento

- Abrir RStudio
- File → New File → R Markdown...
- En la ventana que aparece, Poner título
- Esto crea una plantilla en markdown<sup>a</sup>. Guardar con Save as... y poner un nombre con la extensión .Rmd

---

<sup>a</sup>Markdown es un lenguaje parecido a html pero muchísimo más simple que permite dar formatos marcando el texto y luego usando un intérprete para cambiar el formato

## Práctica con R

R se puede utilizar de varias maneras, dependiendo de la experiencia, conocimiento y necesidad de sistematización:

**Consola** De manera directa, a través de la consola de comandos

**Gui** A través de una interfaz de usuarios gráfica (GUI). En este tema, hay varias opciones, entre las más importantes están RStudio, Jupyter, RCommander, Rattle, Emacs.

**Batch** En modo *Batch*, a través de un ícono.

- Comandos pueden introducirse en más de una línea. el prompt que indica la continuación de una línea es el signo más: +
- Comandos múltiples pueden ser introducidos en una misma línea separados por punto y coma (;)
- Los comentarios inician con #.
- Espacios y tabuladores son ignorados excepto cuando están entre comillas.
- R hace distinción entre mayúsculas y minúsculas.
- Se pueden usar las teclas  $\uparrow$  o  $\downarrow$  para navegar entre los comandos que han sido tecleados previamente.
- Se puede obtener ayuda de una función, por ejemplo `sin`, usando la ayuda HTML o bien tecleando `?sin` o `help(sin)`.
- La combinación de teclas `Ctrl + L` limpia la consola.

Los comandos más elementales consisten de expresiones o asignaciones.

```
2 + 3
```

```
[1] 5
```

```
sqrt(3/4)/(1/3-2*pi^2)
```

```
[1] -0.04462697
```

```
# genera datos aleatorios con distribución normal.
```

```
x <- rnorm(100,sd=4); y <- runif(100,-2,5)
```

```
# Los siguientes comandos devuelven por default los primeros y últimos 6 datos respectivamente
```

```
head(x); tail(x)
```

```
[1] 4.100420 -3.156826 -3.365561 -8.393549 1.085769 7.123904
```

```
[1] -5.6332487 -1.9944051 -3.6594870 -5.8655660 1.8938763 -0.2941443
```

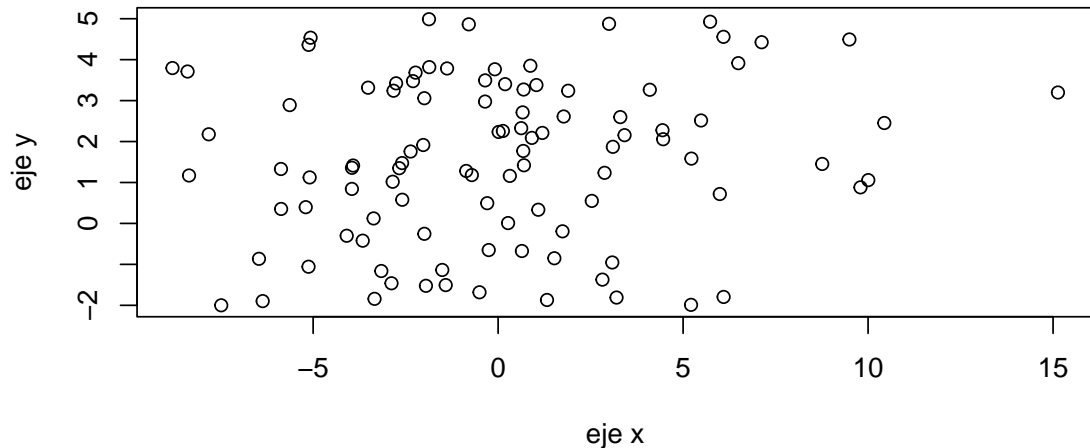
```
head(x,10) # Se puede cambiar el número de datos que queremos ver.
```

```
[1] 4.1004195 -3.1568256 -3.3655609 -8.3935491 1.0857691 7.1239042
```

```
[7] -0.8621447 3.1018217 0.3167458 -5.0936772
```

```
plot(x,y, main="Mi primera gráfica\n ya soy expert!", xlab="eje x", ylab="eje y")
```

**Mi primera gráfica  
ya soy expert!**



# Ejemplos I

```
z <- cbind(x,y) # 'pega' dos vectores por columnas
head(z)
```

```
      x      y
[1,] 4.100420 3.2647454
[2,] -3.156826 -1.1622194
[3,] -3.365561 0.1221379
[4,] -8.393549 3.7107744
[5,] 1.085769 0.3341662
[6,] 7.123904 4.4255068
```

```
t(z) %*% z # producto de matrices
```

```
      x      y
x 2053.9249 130.2945
y 130.2945 625.0443
```

```
crossprod(z)
```

```
      x      y
x 2053.9249 130.2945
y 130.2945 625.0443
```

```
h <- chull(x,y) # cubierta conveza del conjunto de puntos
plot(x, y, pch = 16, col = "red", main = "Cubierta conveza del conjunto de puntos")
polygon(x[h], y[h], dens = 15, angle = 30)
```



## Cubierta convexa del conjunto de puntos

