

Primeros Pasos en R

Clase 1: Introducción a R

Ana Alvarado, María Constanza Prado y Riva Quiroga

Contenido del curso

Módulo 1: Introducción a R.

- Descripción software R.
- Diferencia entre R y R Studio.
- Interfaz (consola, editor, menú).
- Lenguaje de programación en R.
- Instalación de paquetes.

Módulo 2: Operaciones básicas en R.

- Importar bases de datos.
- Crear, poblar y grabar bases de datos.
- Concatenar / separar archivos (registros y variables) de Estructuras Estadísticas
- Transformar/re codificar variables.
- Creación de objetos.
- Creación de funciones.

Módulo 3: Análisis exploratorio y descriptivo en R.

- Calcular estadísticas de resumen.
- Generar tablas.
- Generar gráficos.



La clase de hoy:



Parte I

- Recomendaciones iniciales.
- Herramientas y software en el mundo de la ciencia de datos.
- Descripción Software R
- R, RStudio y RStudioCloud
 - Instalación
 - Estructura
 - Archivos .R
- Directorios de trabajo y proyectos.

Parte II

- Cálculos en R.
- Lógica de funcionamiento de R.
- Funciones en R

Recomendaciones iniciales

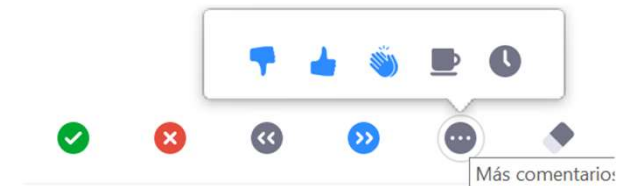


Recomendaciones Iniciales

- Algunas recomendaciones a tener en cuenta durante las clases:
- Usaremos un script de R compartido para que puedan seguir la ejecución del código.
- Botones de ZOOM para proporcionar información.
- Usaremos el Chat de Zoom para consultas y comentarios de contenido que se está viendo en la clase.
- Usaremos un documento compartido para consultas y comentarios de contenidos complementarios a la clase.
- Se espera una participación activa a través de las encuestas de zoom y chat.
- Los ayudantes.
- Mantener sus cámaras y micrófonos apagados durante el curso.

<http://bit.ly/datauc-cursoR-indicaciones>

<https://bit.ly/3skZq2r>

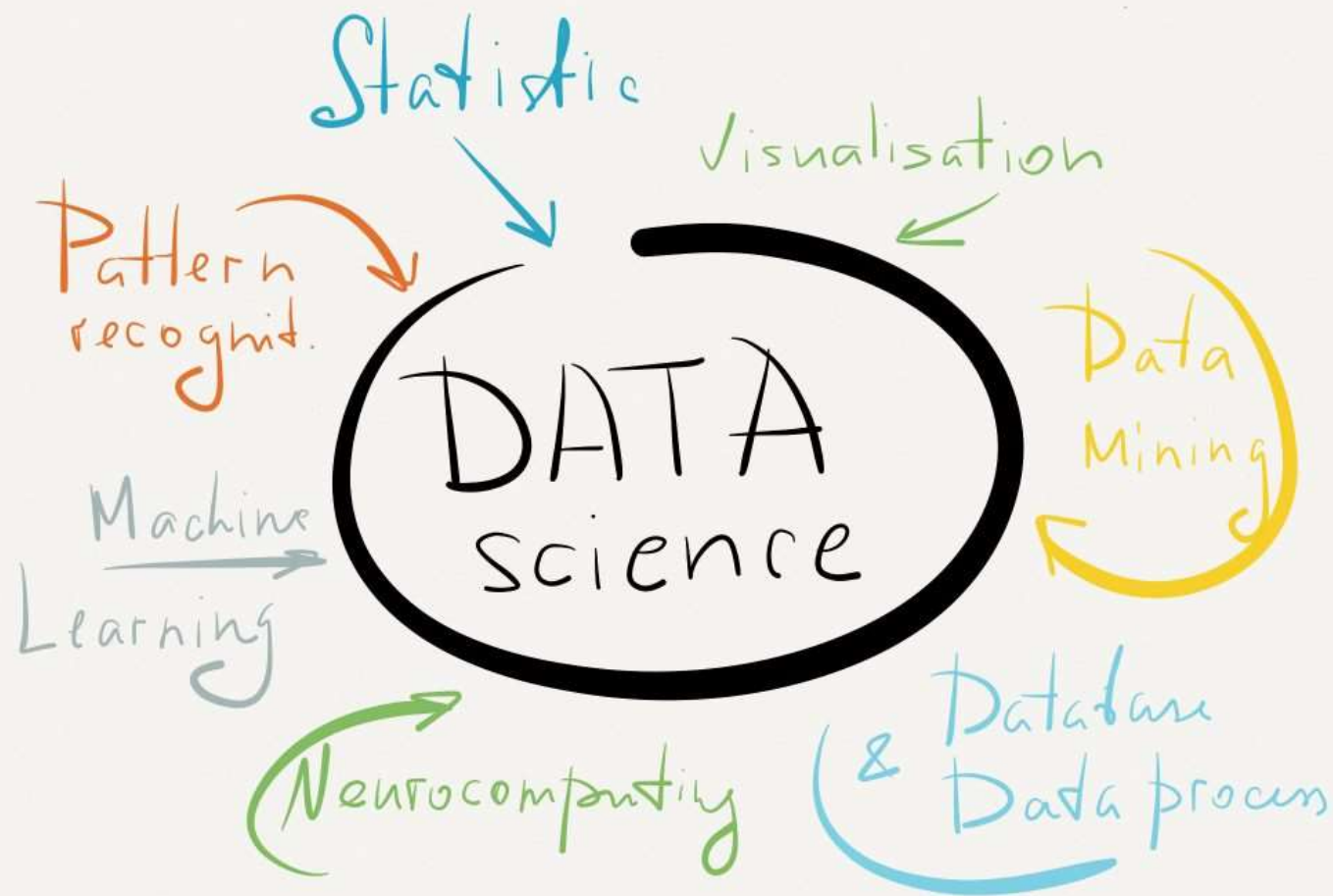


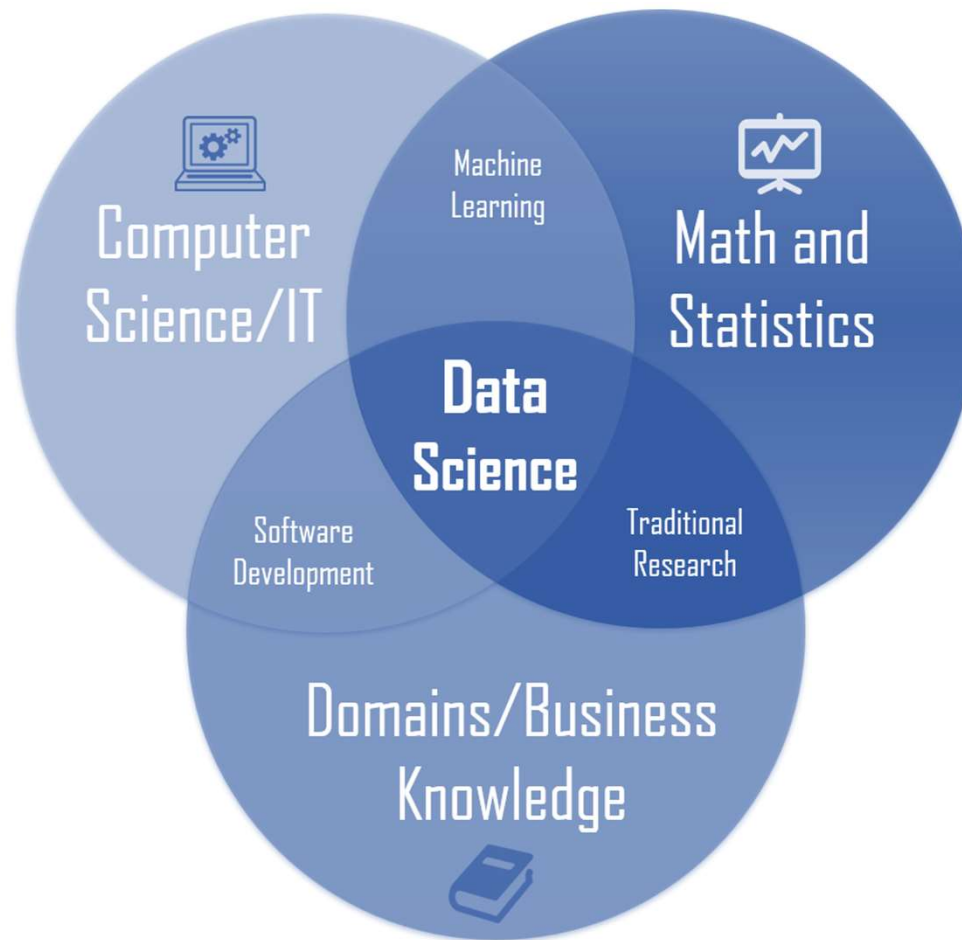
<https://cutt.ly/ljmDiHg>



Herramientas y Software en el mundo de la ciencia de datos







Herramientas y software para DS



Y otras más....

Descripción Software R



¿Qué es R?



El software R es un ambiente y lenguaje de programación libre con un enfoque estadístico para el desarrollo de herramientas, métodos, cálculos y gráficos.

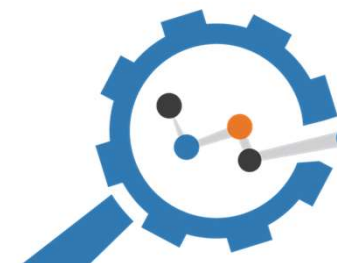
Utiliza un lenguaje específico, anidado principalmente en las palabras comunes de técnicas estadísticas en inglés.

En R, un análisis estadístico se realiza en una [serie de pasos](#), con resultados intermedios que se van almacenando para ser observados o analizados posteriormente. Sin embargo, en otros softwares, el resultado es inmediato, sin tener mucha interacción en la creación o los pasos intermedios.



¿Por qué R?

- Almacenamiento y manipulación efectiva de **datos**.
- Operadores para **cálculo** sobre variables indexadas, en particular, **objetos**.
- Una amplia, coherente e integrada colección de **herramientas** para **análisis** de datos.
- Posibilidades **gráficas** para análisis de datos, que funcionan directamente sobre pantalla o para exportar.
- Lenguaje de **programación** bien desarrollado, simple y efectivo, que incluye condicionales, ciclos, funciones recursivas y posibilidad de entradas y salidas,
- **Interacción** con otros softwares estadísticos,
- Mucho más...



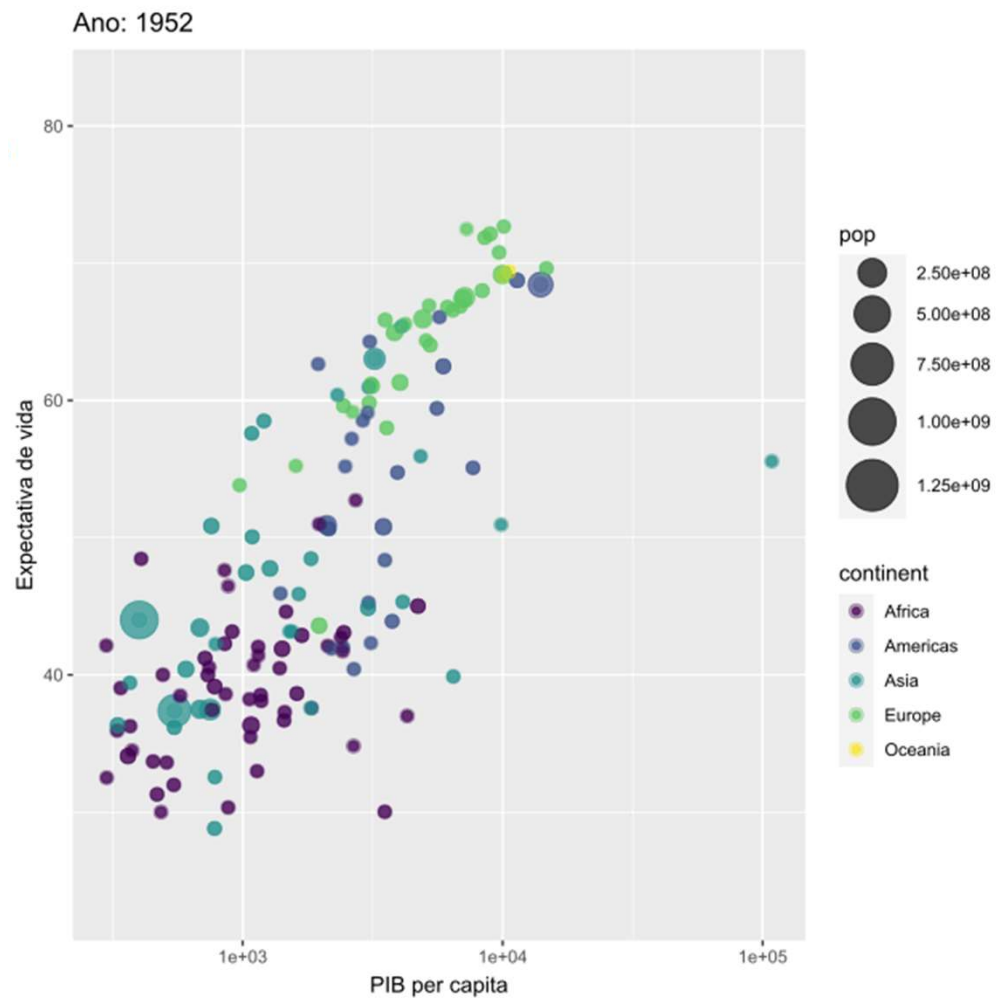
Paquetes de R

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

Son colecciones de funciones, datos y documentación.

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





Data Science - Flujo de trabajo

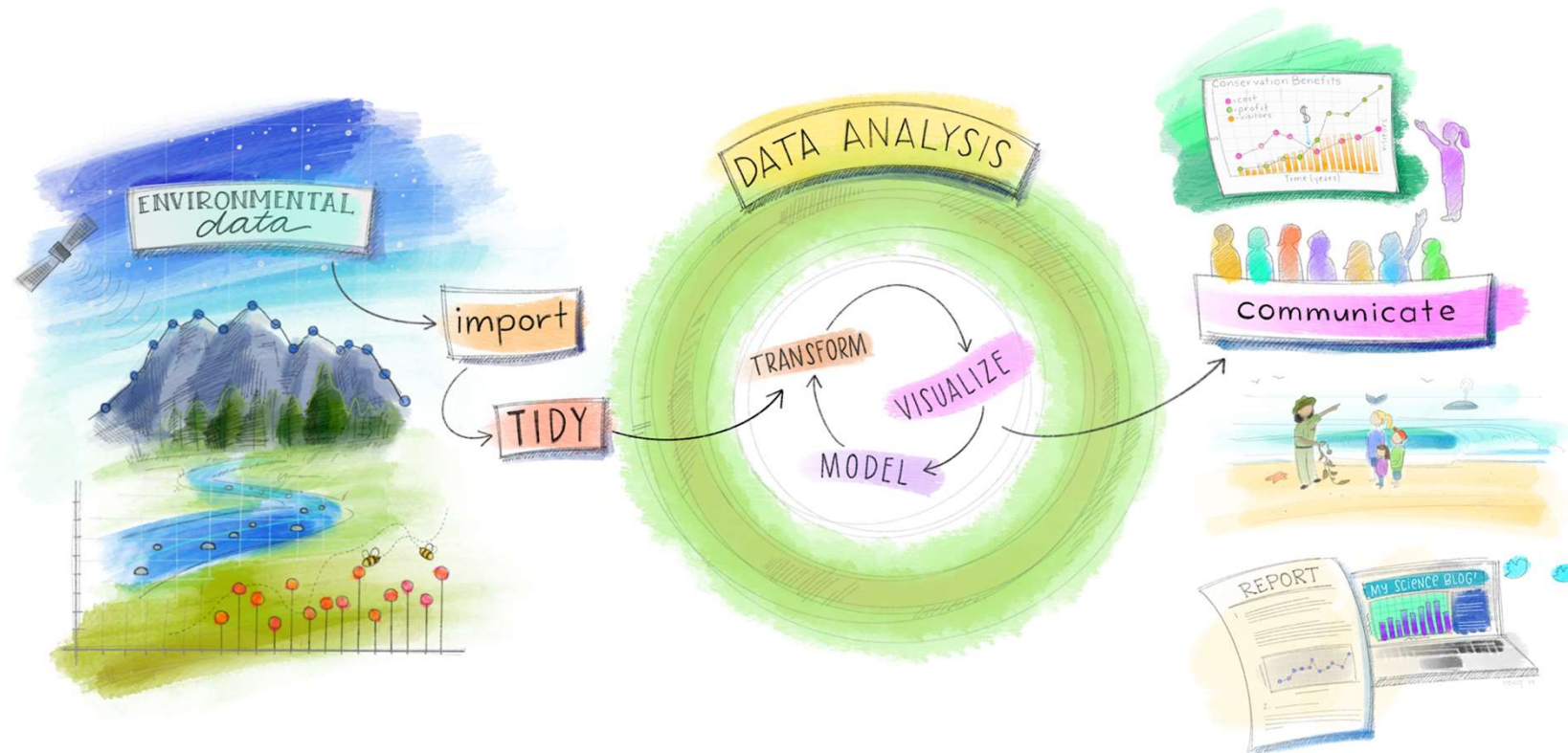


Ilustración por Allison Horst @allison_horst basada en el esquema de flujo del libro R for Data Science de Golemund & Wickham.

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

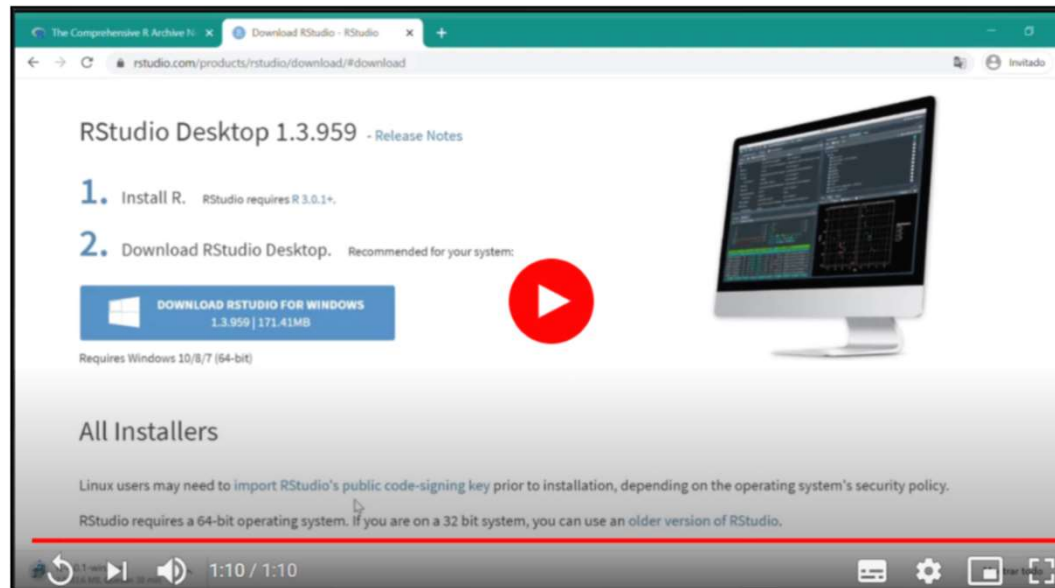


R Studio

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



Sobre la instalación



También pueden revisar el enlace enviado:

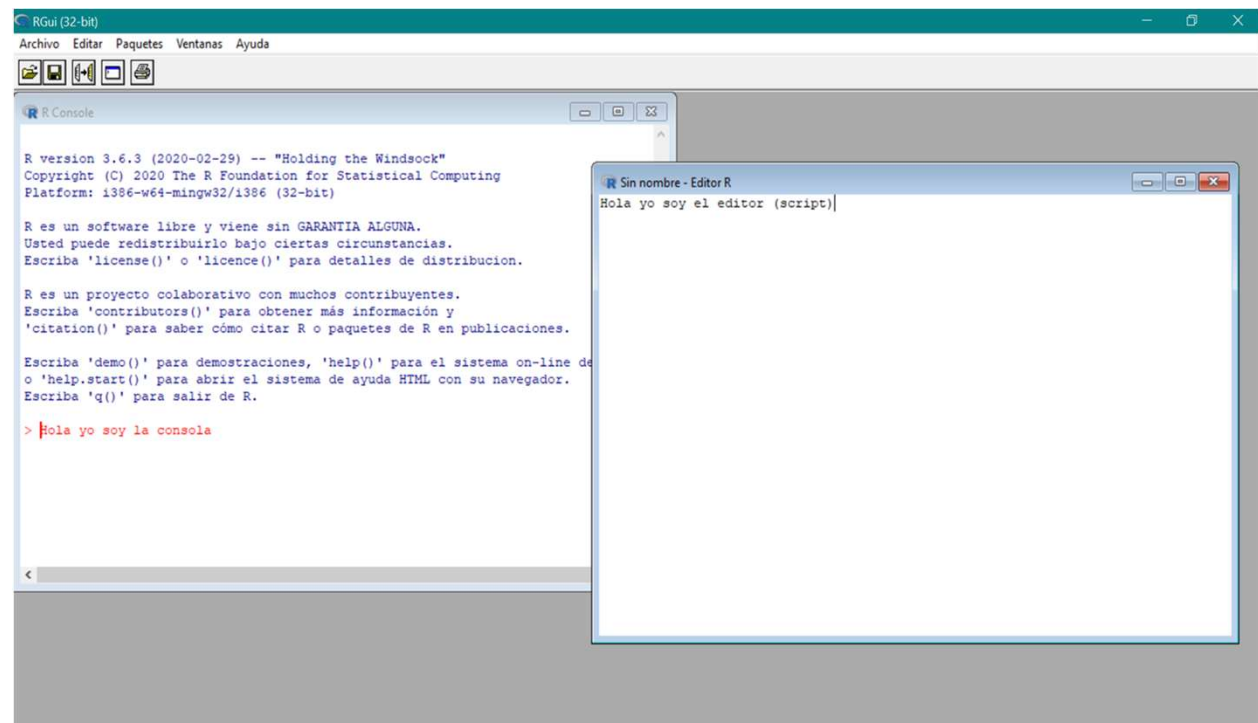
<http://bit.ly/datauc-cursoR-indicaciones>

R



Así es como se ve R:

Tenemos una ventana que corresponde a la [consola](#) y otra que es el editor o [script](#).



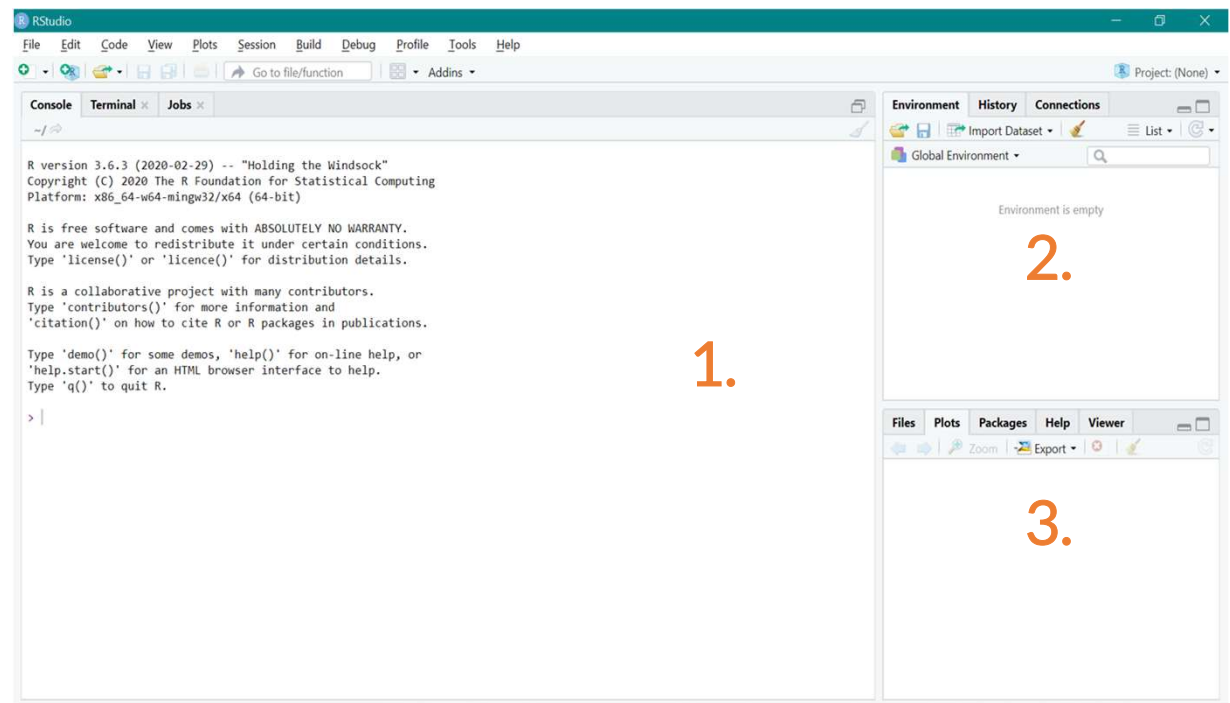
RStudio



Así es como se ve RStudio:

Tenemos varios espacios de trabajo dispuestos en 3 paneles:

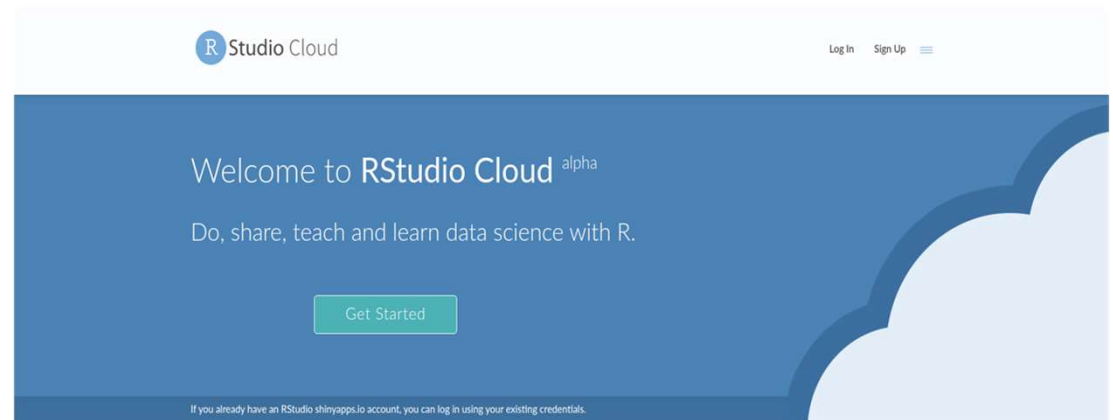
1. Consola
2. Ambiente de trabajo
3. Panel de visualización y ayuda.



RStudioCloud

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

Para empezar a trabajar con RStudioCloud necesitamos crear un [proyecto](#).



THE MISSION

We created RStudio Cloud to make it easy for professionals, hobbyists, trainers, teachers and students to do, share, teach and learn data science using R.



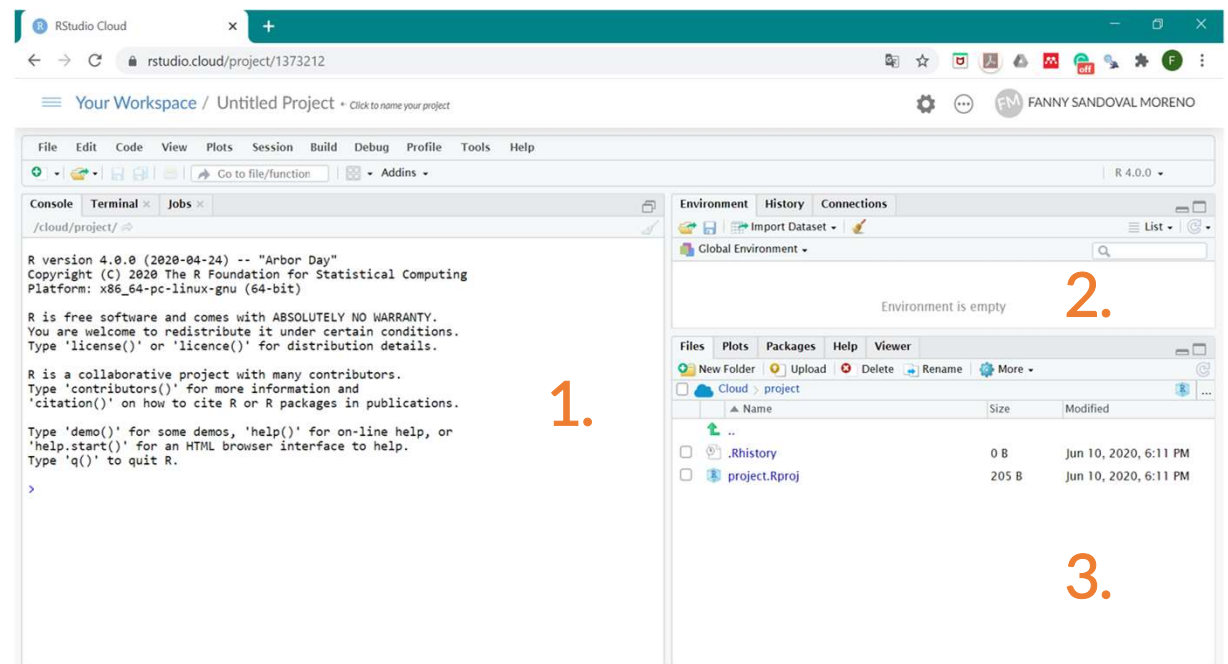
También puedes revisar el enlace con indicaciones que enviamos para más información: <https://bit.ly/2Uw99ng>

RStudioCloud

Así luce RStudioCloud:

También podemos ver:

1. Consola
2. Ambiente de trabajo
3. Panel de visualización y ayuda.



Ahora...

Abre R (instalado en tu equipo o RStudioCloud) escribe en la consola el siguiente código y presiona **enter**:

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

Ahora presiona:

Crtl/Cmd + L

¿Qué pasó?



Script (archivo .R)

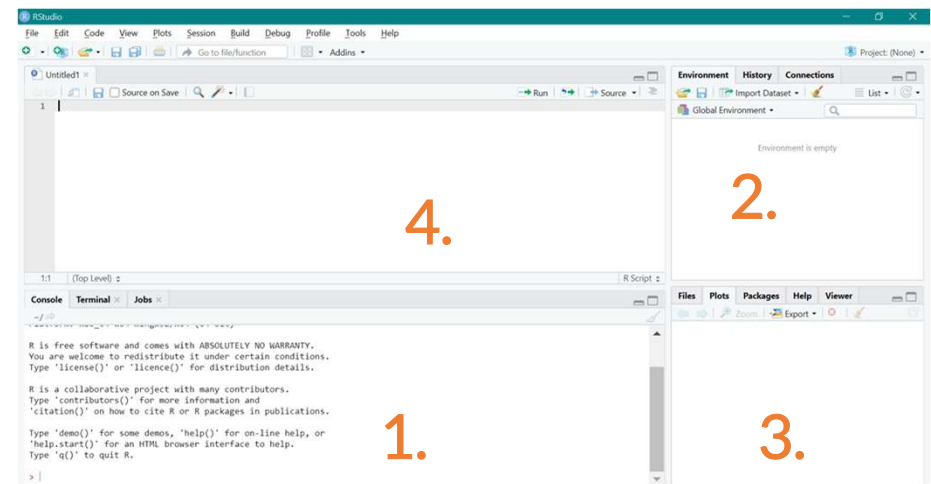
Un script es un espacio en donde escribimos el código que queremos [conservar](#).

Para crear un nuevo script:

- File > New File > R Script
- Crtl/Cmd + Shift + N

Ahora tenemos 4 espacios de trabajo:

1. Consola
2. Ambiente de trabajo
3. Panel de visualización y ayuda.
4. Script



Script (archivo .R)

Para ejecutar ("correr") código:

- Ejecutar solo la expresión seleccionada:

 +  (Mac)

 +  (Windows)

- Para ejecutar todo el script:
Ctrl/Cmd + Shift + Enter

 +  +  (Mac)

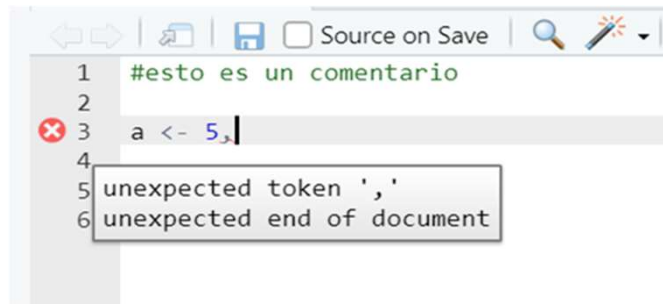
 +  +  (Windows)

Script (archivo .R)

Para comentar usamos el símbolo #

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

Diagnóstico:



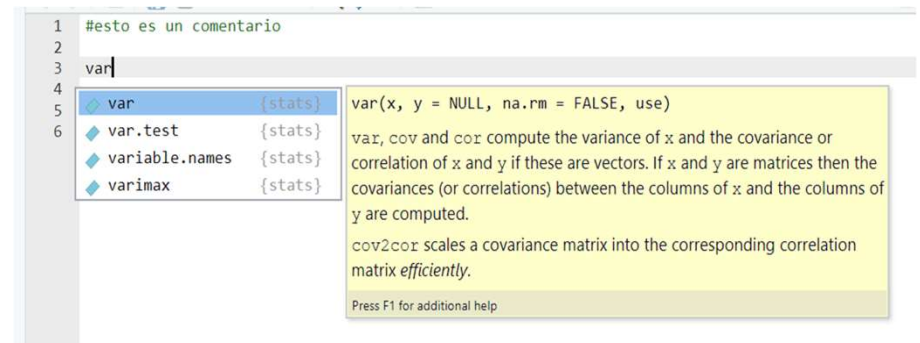
A screenshot of the RStudio editor interface. The top toolbar shows icons for undo, redo, save, and a search icon. Below the toolbar, the source editor shows the following code:

```
1 #esto es un comentario
2
3 a <- 5,
4
5
6
```

A red 'x' icon is next to line 3. A tooltip box is open, displaying the error message:

```
unexpected token ','
unexpected end of document
```

Autocompletado:



A screenshot of the RStudio editor interface showing autocompletion. The top toolbar is the same as in the previous image. The source editor shows the following code:

```
1 #esto es un comentario
2
3
4 var|
5
6
```

A dropdown menu is open, showing the following options:

- var {stats}
- var.test {stats}
- variable.names {stats}
- varimax {stats}

A tooltip box is open, displaying the documentation for the `var` function:

```
var(x, y = NULL, na.rm = FALSE, use)
var, cov and cor compute the variance of x and the covariance or
correlation of x and y if these are vectors. If x and y are matrices then the
covariances (or correlations) between the columns of x and the columns of
y are computed.
cov2cor scales a covariance matrix into the corresponding correlation
matrix efficiently.
Press F1 for additional help
```

Al guardar el script se genera un archivo .R, por ejemplo: *nombre.R*

Ahora...

1. Crea un nuevo script
2. Guárdalo con el nombre "Sesion1"
3. Añade como comentario tu nombre
4. Escribe y ejecuta el siguiente 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")
```

Ahora presiona:
Ctrl/Cmd + Shift + R

¿Para qué sirve?

Directorios de Trabajo y Proyectos



Directorios de trabajo y proyectos

El `directorio de trabajo` es donde R:

- busca los archivos que le pedimos `leer` y
- 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.

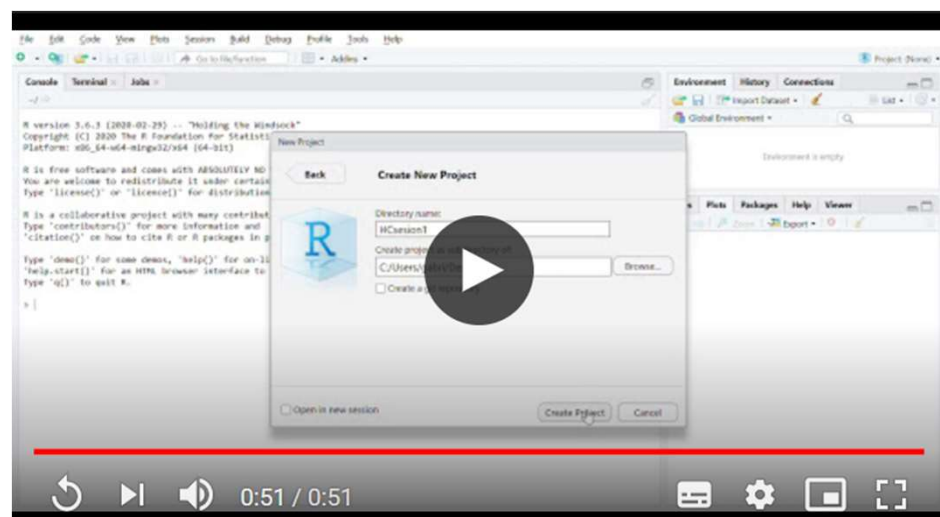
Directorios de trabajo y proyectos

Durante nuestro flujo de trabajo lo ideal es mantener todos los archivos relacionados a un análisis en un mismo lugar.

RStudio nos permite trabajar con [proyectos](#).

Trabajar con proyectos es una buena práctica que nos ahorrará muchos dolores de cabeza.

Para crear un nuevo proyecto ve a la parte superior derecha de la ventana de RStudio,



Ahora...

1. Crea un proyecto para esta clase *HCsesion1*
2. Abre el proyecto y ejecuta el siguiente código: `getwd()`
3. Crea un nuevo script y ejecuta el siguiente código:

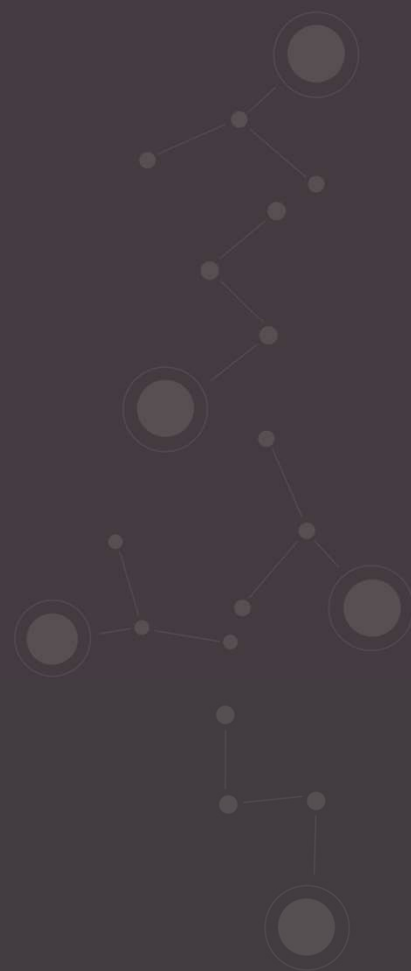
```
poleras <- c(254,203,182,50)
meses <- c("Ene", "Feb", "Mar", "Abr")
min(poleras)
plot(poleras)
barplot(poleras,names.arg = meses,col="orange")
write.csv(cbind(meses, poleras),"poleras.csv")
read.csv("poleras.csv")
```

Ahora...

1. Crea una carpeta dentro de tu proyecto llamada *datos*
2. Ahora ejecuta el siguiente código:

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

Recuerda: los proyectos usan rutas relativas, no absolutas.
Es más fácil reproducir los análisis al compartirlos.



Cálculos en R



Cálculos en R

R se puede usar como una calculadora.

Algunas operaciones que podemos realizar son:

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

Calcule en R:

1. $2 + 4 \times 5 - e^3$

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

Cálculos en R

R se puede usar como una calculadora.

Algunas operaciones que podemos realizar son:

+	suma
-	resta
*	multiplicación
/	división
/%	división entera
^	potencia
sqrt()	raíz cuadrada
log()	logaritmo base e
exp()	exponencial

Calcule en R:

1. $2 + 4 \times 5 - e^3$

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

Solución:

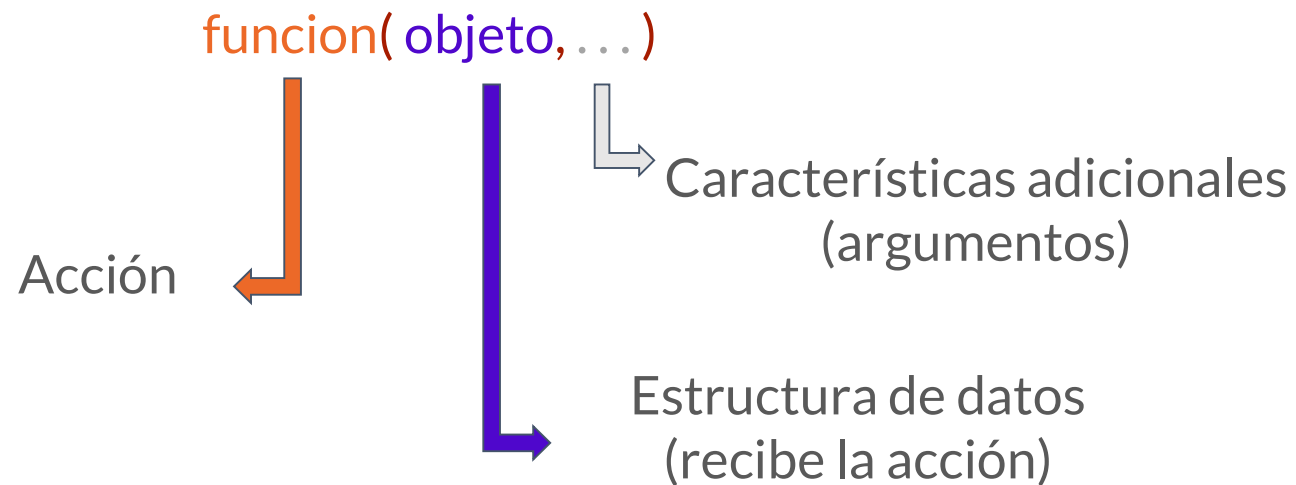
```
> 2 + (4*5) - exp(3)
[1] 1.914463
> log(5) + pi/sqrt(5)
[1] 3.014401
```

Lógica de funcionamiento en R



La lógica de funcionamiento de R

En lo relacionado con [análisis de datos](#), la lógica de R es bastante amigable:



La lógica de funcionamiento de R

¿Qué hacer?
Función

¿A quién?
Objeto

“Traduciendo” nuestras ideas

Datos:

Cantidad de poleras disponibles por mes

Meses	poleras
Enero	254
Febrero	203
Marzo	182
Abril	50

“Traduciendo” nuestras ideas

Datos:

Cantidad de poleras disponibles por mes

Meses	poleras
Enero	254
Febrero	203
Marzo	182
Abril	50



Graficar mis datos



¿Cuál es la cantidad mínima de poleras que tuve disponible?

“Traduciendo” nuestras ideas

Datos:

Cantidad de poleras disponibles por mes

Meses	poleras
Enero	254
Febrero	203
Marzo	182
Abril	50



Graficar mis datos



¿Cuál es la cantidad mínima de poleras que tuve disponible?



“Aprender a programar es como aprender una nueva lengua”

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

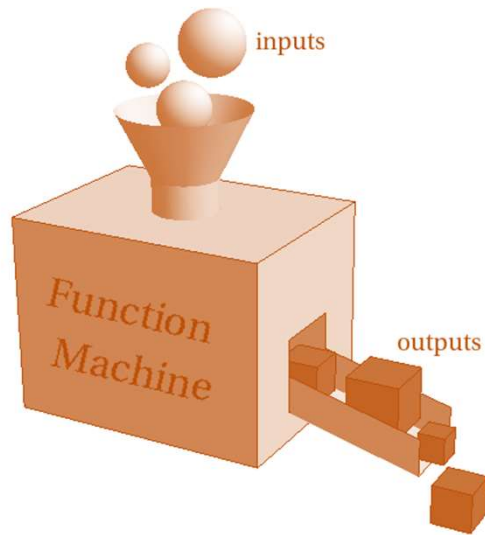
} objetos*

* el detalle de esto lo veremos luego

Funciones en R



Funciones en R



Hacen que “algo suceda”.

Algunas requieren **inputs** y otras no, los inputs (argumentos) van dentro del **paréntesis**

`sqrt(521)`

`getwd()`

`Sys.Date()`

En las próximas clases aprenderemos a crear nuestras propias funciones

Funciones en R

Al escribir y ejecutar solo el nombre de la función podemos consultar el código almacenado en ella.

`sqrt`

Podemos consultar ayuda y detalles de la función al ejecutar:

`?sqrt`

Los inputs son más conocidos como **argumentos** y se separan con una coma.

Para consultar los argumentos de una función usamos:

`args (nombrefuncion)`

Algunos argumentos son opcionales y otros no.

Ahora...

¿Qué argumentos recibe la función `log()`?

`args(log)`

¿Cuál es la diferencia entre `log(100,10)` y `log(10,100)`?

¿Cuál es el error en el siguiente código?

`log(base=10)`



Referencias y material complementario

“R para Ciencia de Datos”: <https://es.r4ds.hadley.nz/> (libro en línea, en español)

RStudio cheatsheets: <https://rstudio.com/resources/cheatsheets/> (“torpedos”; en la parte inferior de la página hay versiones en español disponibles)

RStudio Cloud: <https://rstudio.cloud> (para utilizar RStudio en la nube)

RStudio Primers: <https://rstudio.cloud/learn/primers> (ejercicios interactivos; en inglés)

“Hands-on Programming with R”: <https://rstudio-education.github.io/hopr/> (libro en línea de introducción a la programación en R que no requiere conocimientos previos; en inglés)