

Buenas prácticas en R

Natalia Riquelme/ Isidora Castillo

2020/1/16 (updated: 2020-05-27)

¿Cómo afrontar este desafío?

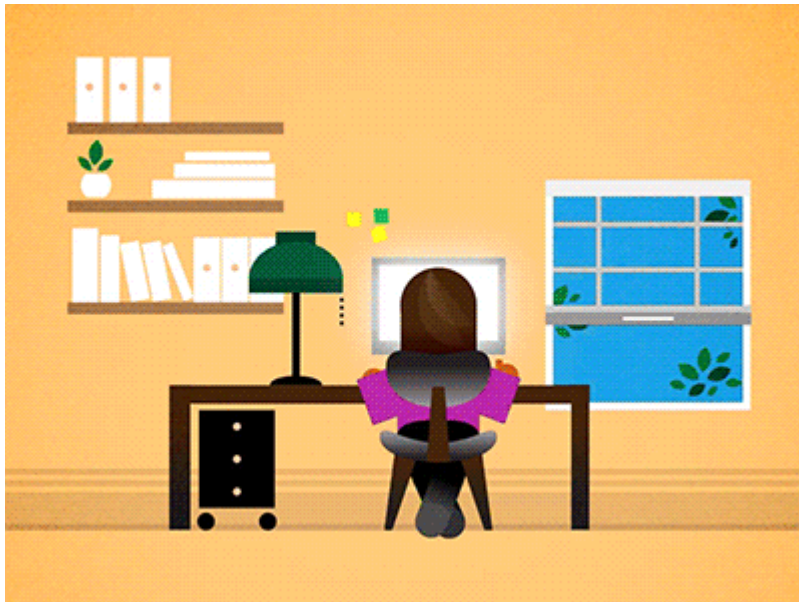
Sé curiosa/o:

La curiosidad es buena: Hazte preguntas sobre los datos que tienes, busca, informate. Esto aumentará tu interés y mejorará tu enfoque.



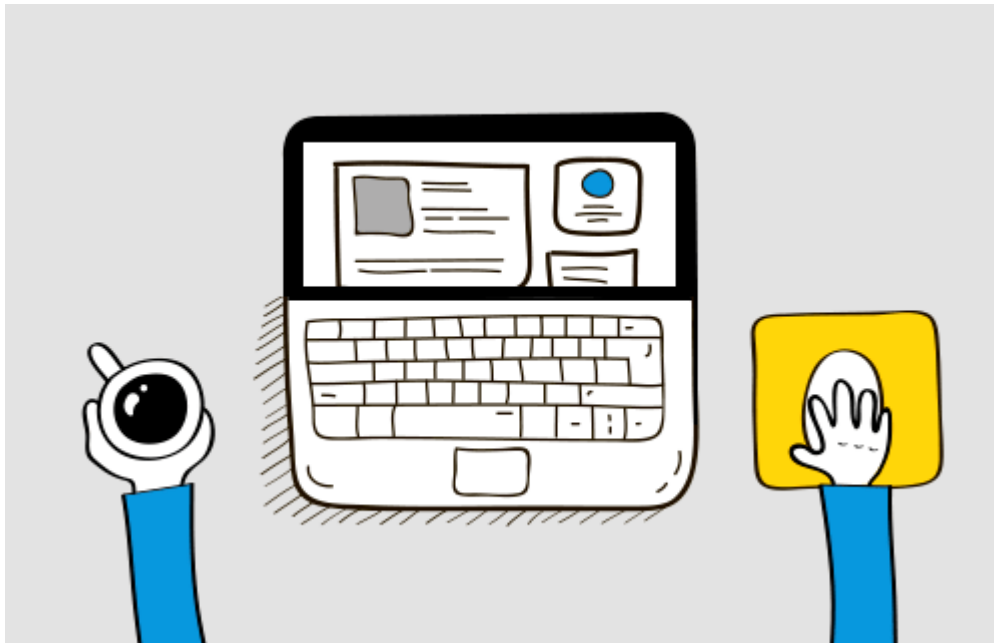
¡No te desanimes!

Los lenguajes de programación no son simples de aprender, pero con trabajo y motivación veras como en poco tiempo lograrás grandes avances.



¡Nunca dejes de aprender!

La web tiene tantos recursos gratuitos, consejos y tutoriales sobre cómo usar R; junto con paquetes de muestra que puedes descargar. ¡Úsalos para tu ventaja!



Descarga e instala R & R Studio IDE (Interactive development environment)

- 5 / 23

The screenshot shows the RStudio interface with a script file named "CC_1_RBasics_Full_Script2.R". The code defines filters for various taxonomic groups and calculates the number of unique species within each group.

```
R
# Taxonomic Group Filters
Beetle <- filter(edidiv, taxonGroup == "Beetle")
Bird <- filter(edidiv, taxonGroup == "Bird")
Butterfly <- filter(edidiv, taxonGroup == "Butterfly")
Dragonfly <- filter(edidiv, taxonGroup == "Dragonfly")
Flowering.Plants <- filter(edidiv, taxonGroup == "Flowering")
Fungus <- filter(edidiv, taxonGroup == "Fungus")
Hymenopteran <- filter(edidiv, taxonGroup == "Hymenopteran")
Lichen <- filter(edidiv, taxonGroup == "Lichen")
Liverwort <- filter(edidiv, taxonGroup == "Liverwort")
Mammal <- filter(edidiv, taxonGroup == "Mammal")
Mollusc <- filter(edidiv, taxonGroup == "Mollusc")

# To find out the number of different species in each taxa,
a <- length(unique(Beetle$taxonName))
b <- length(unique(Bird$taxonName))
c <- length(unique(Butterfly$taxonName))
d <- length(unique(Dragonfly$taxonName))
e <- length(unique(Flowering.Plants$taxonName))
```

The console output shows several errors:

```
> Liverwort <- filter(edidiv taxonGroup == "Liverwort")
Error: unexpected symbol in "Liverwort <- filter(edidiv taxonGroup
> a <- length(unique(Beetle$taxonName))
Error: unexpected ')' in "a <- length(unique(Beetle$taxonName))"
> e <- length(unique(FloweringPlants$taxonName))
Error in unique(FloweringPlants$taxonName) :
  object 'FloweringPlants' not found
> Beetle <- filter(edidiv, taxonGroup == "Beetle")
Error in filter(edidiv, taxonGroup == "Beetle") :
  object 'taxonGroup' not found
>
```

Practica la instalación de paquetes.

Puede usar <http://swirlstats.com/> Es un sitio que te permite aprender R mientras estás usando R

```
## install.packages(swirl)
library(swirl)
```

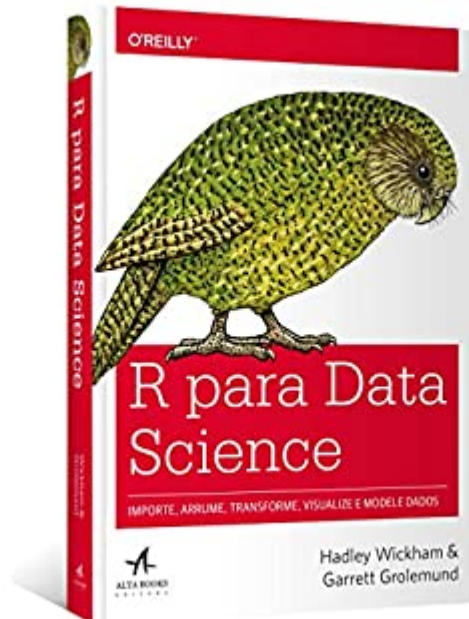
```
##
## | Hi! Type swirl() when you are ready to begin.
```

swirl

Leer la documentación

Utiliza todas las herramientas que que la web puede darte para aprender. Leer manuales y blogs puede ayudar mucho a tu conocimiento. La comunidad de R es bastante activa en twitter, por lo que esta herramienta tambien es de utilidad (tip: busca el hashtag #Rstats).

R para Ciencia de Datos R Bloggers #RStats



Github

Github es una interfaz gráfica web-based que funciona como servicio de alojamiento de repositorios, permitiendo alojar proyectos utilizando el sistema de control de versiones Git.

github



Familiaridad

Al comenzar a aprender sobre el análisis de datos es recomendable usar datos que ya conoces, será más fácil entenderlo todo si está familiarizado con el tema / puntos de datos.

	Avg [stdv]	Avg [stdv]	Median	Median	Min	Min	Max	Max
	BH	OW	BH	OW	BH	OW	BH	OW
pH	7.0 [0.2]	7.3 [0.2]	7.1	7.3	6.3	6.8	7.4	7.9
Cond ($\mu\text{S cm}^{-1}$)	348 [61]	379 [172]	352	365	207	157	437	1091
HCO_3^- (mg L^{-1})	162 [57]	185 [66]	159	185	74	85	277	323
F^- (mg L^{-1})	2.4 [1.2]	1.1 [0.8]	2.5	0.7	0.6	0.3	4.5	3.3
SO_4^{2-} (mg L^{-1})	4.4 [2.0]	7.7 [7.3]	4.1	4.8	1.9	1.2	7.8	38.6
NO_3^- (mg L^{-1})	29.0 [18.2]	21.0 [20.1]	24.6	17.2	0.3	2.2	61.8	97.4
Cl^- (mg L^{-1})	13.3 [8.2]	16.3 [16.2]	12.4	11.5	2.4	3.6	30.1	93.8
Na^+ (mg L^{-1})	24.4 [4.9]	30.3 [17.3]	23.6	26.5	17.1	7.5	35.0	103.0
K^+ (mg L^{-1})	3.3 [1.1]	8.1 [25.0]	2.9	3.2	1.9	1.4	5.4	149.0
Ca^{2+} (mg L^{-1})	30.0 [12.2]	33.5 [12.2]	31.8	33.8	16.2	11.6	43.5	64.2
Mg^{2+} (mg L^{-1})	13.4 [3.4]	10.8 [5.5]	14.1	9.0	6.7	3.0	18.5	30.2

Visualización

Ser creativo, crear cuadros y gráficos y volverlos coloridos permitirá ver lo que estás haciendo y te motivará a seguir adelante.

Comentarios, comentarios y más comentarios

Agrega comentarios a tu código para que al volver a revisarlos puedas recordar qué es lo que querías hacer. Estos comentarios ayudarán a otros a entender tu código y cuando leas el código de alguien más podrás saber qué es lo que quería lograr. Inserta comentarios comenzando la declaración con un signo # **ctrl + shift + r**

Mi primera Sección

95 #----- 2.CREACIÓN DE UNA BASE DE DATOS -----

96

97 #Antes que todo, estableceremos una carpeta de trabajo.

98 #Botones: Session -> Set Working Directory -> Choose directory -> Elegir carpeta

99 #Abreviación de botones: Ctrl + Shift + H

100

101 setwd("C:/Users/Felipe/Desktop/Dropbox/Docencia/Taller R. Estación Lastarria/2. Segunda sesión")

102

103 #

104 1.CREACIÓN Y MANIPULACIÓN DE OBJETOS

105 2.CREACIÓN DE UNA BASE DE DATOS

106 3. BIBLIOTECA, INSTALACIÓN/EJECUCIÓN DE PAQUETES

107 4. IMPORTACIÓN DE BASES DE DATOS

108 5. MANEJO DE DATOS

109 6. RECODIFICACIÓN VARIABLES

110 7. ESTADÍSTICA DESCRIPTIVA SIMPLE

111 8. GRÁFICOS

112 9. RESPUESTA EJERCICIOS

113

114

17:1 # 1.CREACIÓN Y MANIPULACIÓN DE OBJETOS

300000,500000,650000,410000,750000)

uerdo) #resulta como objeto de "datos" en entorno

o (o hacer click en entorno)

Un lenguaje de codificación adecuado

Características de un buen script:

1. Que los humanos puedan leerlo bien
 2. Que sea lo más auto-explicativo posible.
-



Recomendaciones:

La consistencia es clave y, como cualquier otro idioma, permite que otros te entiendan.

- Hacer sangrías
- Usar espacios
- Es recomendado por varios expertos que se utilice '<-', ya que '=' es utilizado por muchas llamadas a funciones y definiciones. **Alt + -**

```
## Código del Taller de calendarización con ggplot realizado por Jav-  
  
# ggplot(dataBI2, aes(x = BI, y = Stage, colour = Name.BI)) +  
#   geom_point(size = 2)+  
#   facet_wrap( ~ Date) +  
#   theme_bw() +  
#   theme(legend.position = "bottom") +  
#   labs(x="Store Visited", y="Stages", title = "") +  
#   theme(plot.title=element_text(family='', face='bold', size=10, hju  
#   scale_y_discrete(limits = c("stage-1","stage-2","stage-3")) +  
#   scale_x_discrete(limits = c(1,2,3,4,5,6,7,8))
```

Crea proyectos y gestionalos

Un proyecto en RStudio es una **colección de trabajos organizados en una carpeta de trabajo**. RStudio proporciona herramientas que te ayudarán a administrar tu trabajo en proyectos. RStudio recuerda qué archivos tenías abiertos y qué pestañas se mostraban cuando cierras un proyecto. Cuando abras el proyecto nuevamente, RStudio abrirá los mismos archivos y mostrará las mismas pestañas. Esto te permitirá retomar rápidamente tu trabajo.

Recomendaciones para el trabajo en proyectos

Si su proyecto es pequeño, es conveniente mantener todos sus archivos en la carpeta de un proyecto creado por RStudio. Para proyectos más grandes con múltiples documentos, guiones, figuras, etc., usar subcarpetas para organizar su trabajo facilitará la vida.

Algunas sugerencias para organizar carpetas dentro de un proyecto.

- Crea una subcarpeta para tus datos originales
- Crea una subcarpeta para los datos limpios
- Crea una subcarpeta para sus scripts
- Crea una carpeta para cada documento
- Crea una carpeta si necesita guardar figuras

Dentro de sus carpeta de scripts puede ayudar la creación de varios ficheros funcionales numerados, por ejemplo:

- **00_load.R**, puedes cargar tus datos en varios formatos
 - **01_clean.R**, para limpiar los datos; típicamente, este fichero suele crecer a lo largo del análisis
 - **02_eda.R**, para en análisis exploratorio y gráfico
 - **03_analysis_cca.R**, por ejemplo
 - **04_analysis_reg.R**, ...
-

Atajos de Teclado

R Studio nos entrega una enorme cantidad de Keyboard Shortcuts que te ayudaran a ahorrar tu recurso más valioso: Tiempo. Algunos de estos atajos son:

- **Ctrl / Cmd + Enter** ejecuta la línea / bloque de código R en la consola R
 - **Ctrl + 1** salta al script
 - **Ctrl + 2** saltar a la consola R
 - **Ctrl + W** cerrar la pestaña abierta (variaciones: Ctrl + Shift + w/ Ctrl + Alt + Shift + w)
 - **Ctrl + Shift + N** nuevo script
 - **Ctrl + Shift + C** comentar la línea
 - **Ctrl + Shift + R** insertar salto de sección
 - **Ctrl + Alt + i** insertar un fragmento Rmd vacío
 - **Alt + -** inserta el <-
 - **Ctrl + Shift + M** inserta el operador de la tubería %>%
 - **Ctrl + D** elimina toda la línea
 - **Alt + ↑/ ↓** Permite mover líneas de código
-

R Studio tiene tantos atajos de teclado, que tiene un atajo para mostrar los atajos: **Alt + Shift + K**

Code Snippets


Los Code Snippets o fragmentos de código son macros de texto que se utilizan para insertar rápidamente fragmentos comunes de código. Nos permiten ahorrar tiempo guardando la gramática de nuestros códigos más usados. Los code snippets pueden ser personalizados llenando a Tools > Code > Edit Snippets o a través de la siguiente línea de código:

```
usethis::edit_rstudio_snippets()
```

```
## * Edit 'C:/Users/natal/Documents/.R/snippets/r.snippets'
```

1
2
3
4
5
6
7
8

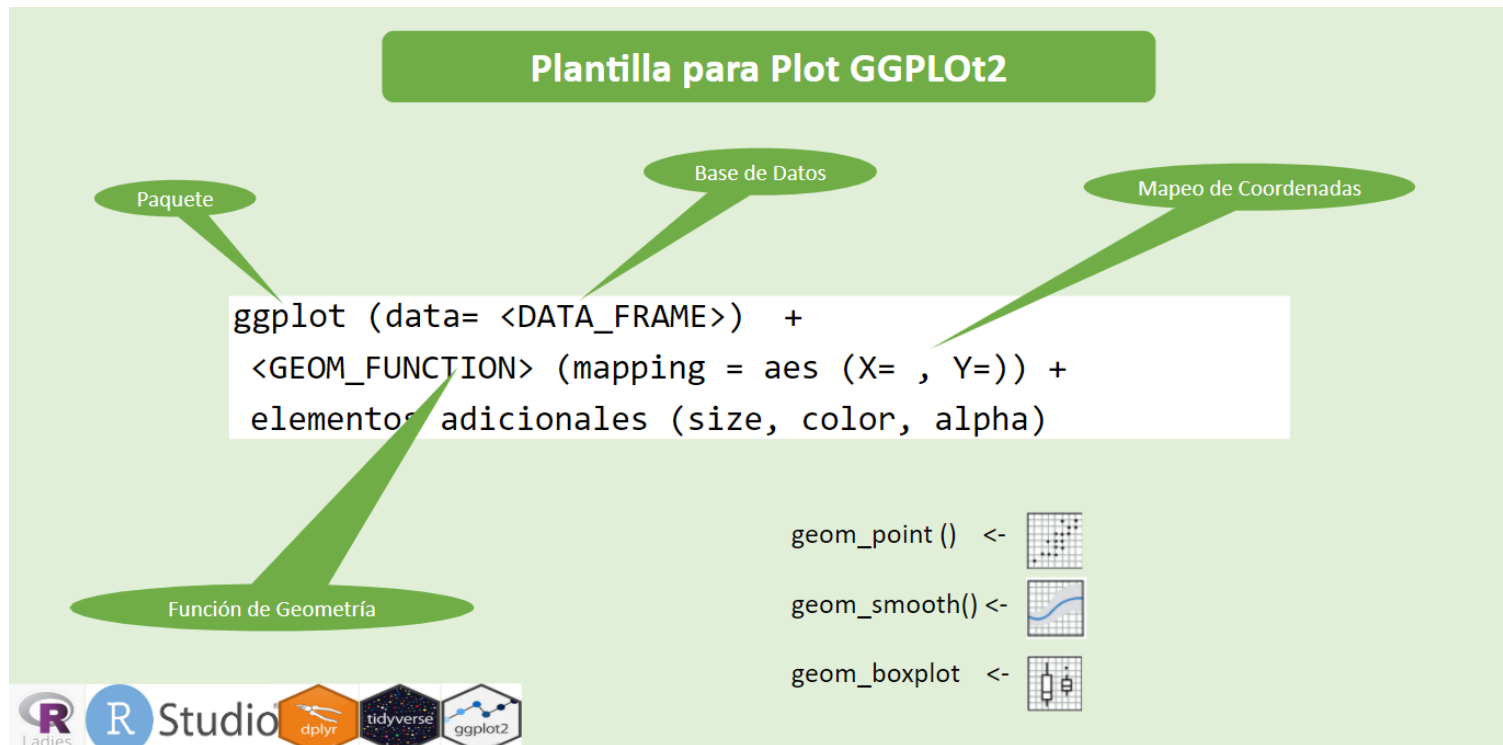
fun

 fun	{snippet}
 function	{base}
 functionBody	{methods}
 functionBody<-	{methods}

```
${1:name} <- function (${2:variables}) {  
  ${3:code}  
}
```

¿Cómo personalizo mis Code Snippets?

Observa la gramática de tu código **#{1:}**



Datos Tidy vs datos no Tidy

A menudo se dice que el 80% del análisis de datos se gasta en la limpieza y preparación de datos. Y no es solo un primer paso, sino que debe repetirse muchas veces a lo largo del análisis a medida que surgen nuevos problemas o se recopilan nuevos datos.

Existen muchas formas de ordenar los mismos datos subyacentes:

```
#tabla1
#> # A tibble: 6 x 4
#>   pais      anio  casos  poblacion
#>   <chr>    <int> <int>      <int>
#> 1 Afganistán 1999     745  19987071
#> 2 Afganistán 2000    2666  20595360
#> 3 Brasil     1999   37737  172006362
#> 4 Brasil     2000   80488  174504898
#> 5 China      1999  212258 1272915272
#> 6 China      2000  213766 1280428583

#tabla2
#> # A tibble: 12 x 4
#>   pais      anio tipo      cuenta
#>   <chr>    <int> <chr>      <int>
#> 1 Afganistán 1999 casos         745
#> 2 Afganistán 1999 población 19987071
#> 3 Afganistán 2000 casos         2666
#> 4 Afganistán 2000 población 20595360
#> 5 Brasil     1999 casos        37737
#> 6 Brasil     1999 población 172006362
#> # ... with 6 more rows
```

```

#tabla3
#> # A tibble: 6 x 3
#>   pais      anio tasa
#>   <chr>    <int> <chr>
#> 1 Afganistán 1999 745/19987071
#> 2 Afganistán 2000 2666/20595360
#> 3 Brasil     1999 37737/172006362
#> 4 Brasil     2000 80488/174504898
#> 5 China      1999 212258/1272915272
#> 6 China      2000 213766/1280428583
# Dividido en dos tibbles
# tabla4a # casos
#> # A tibble: 3 x 3
#>   pais      `1999` `2000`
#>   <chr>    <int> <int>
#> 1 Afganistán    745    2666
#> 2 Brasil      37737   80488
#> 3 China        212258  213766
# tabla4b # poblacion
#> # A tibble: 3 x 3
#>   pais      `1999`      `2000`
#>   <chr>    <int>      <int>
#> 1 Afganistán 19987071  20595360
#> 2 Brasil    172006362  174504898
#> 3 China     1272915272 128042858

```

Datos Tidy

Un conjunto de datos es desordenado o ordenado dependiendo de cómo las filas, columnas y tablas se combinan con observaciones, variables y tipos. En datos ordenados :

1. Cada variable forma una columna.
2. Cada observación forma una fila.
3. Cada valor tiene que tener su propia celda.

pais	anio	casos	poblacion
Afganistán	1999	745	19987071
Afganistán	2000	2666	20595360
Brasil	1999	37737	172006362
Brasil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

variables

pais	anio	casos	poblacion
Afganistán	1999	745	19987071
Afganistán	2000	2666	20595360
Brasil	1999	37737	172006362
Brasil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

observaciones

pais	anio	casos	poblacion
Afganistán	1999	745	19987071
Afganistán	2000	2666	20595360
Brasil	1999	37737	172006362
Brasil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

valores

