Introducción al lenguaje R

Tutorial de Congreso Argentino de Agroinformática 2019

Mg. Yanina Bellini Saibene – INTA Anguil

Dra. María Florencia D'Andrea – IRB - CNIA

48 JAIIO
JORNADAS ARGENTINAS DE INFORMÁTICA

Configuración

Modo: on

El material del curso estará disponible en https://flor14.github.io/cai_2019/

Obteniendo datos

Campo

Obtención de datos a campo – Planillas

Ej: Planilla con valores de presencia – ausencia de especies

Laboratorio

Procesamiento de muestras de experimentos – Mediciones

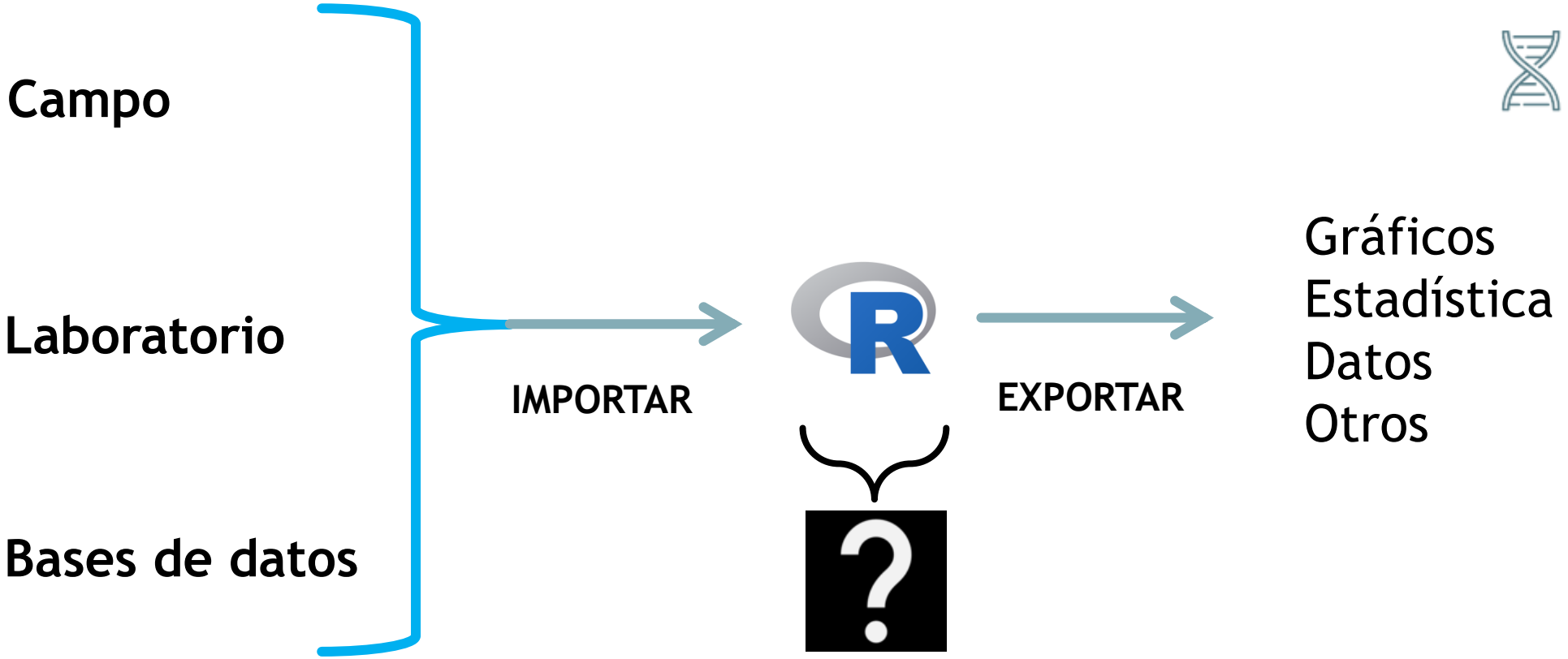
Ej: Planilla con valores de mediciones enzimáticas.

Bases de datos

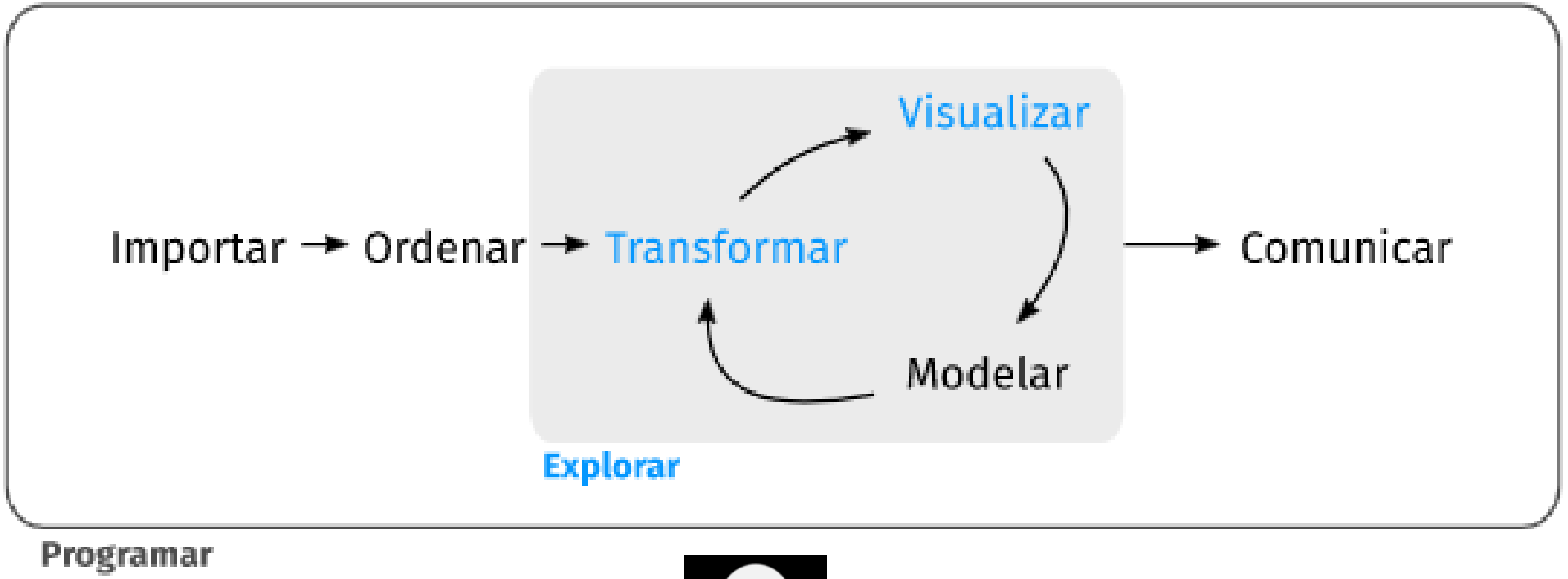
Trabajo sobre datos ya obtenidos

Ej: Datos de estaciones meteorológicas.

Procesamiento de datos



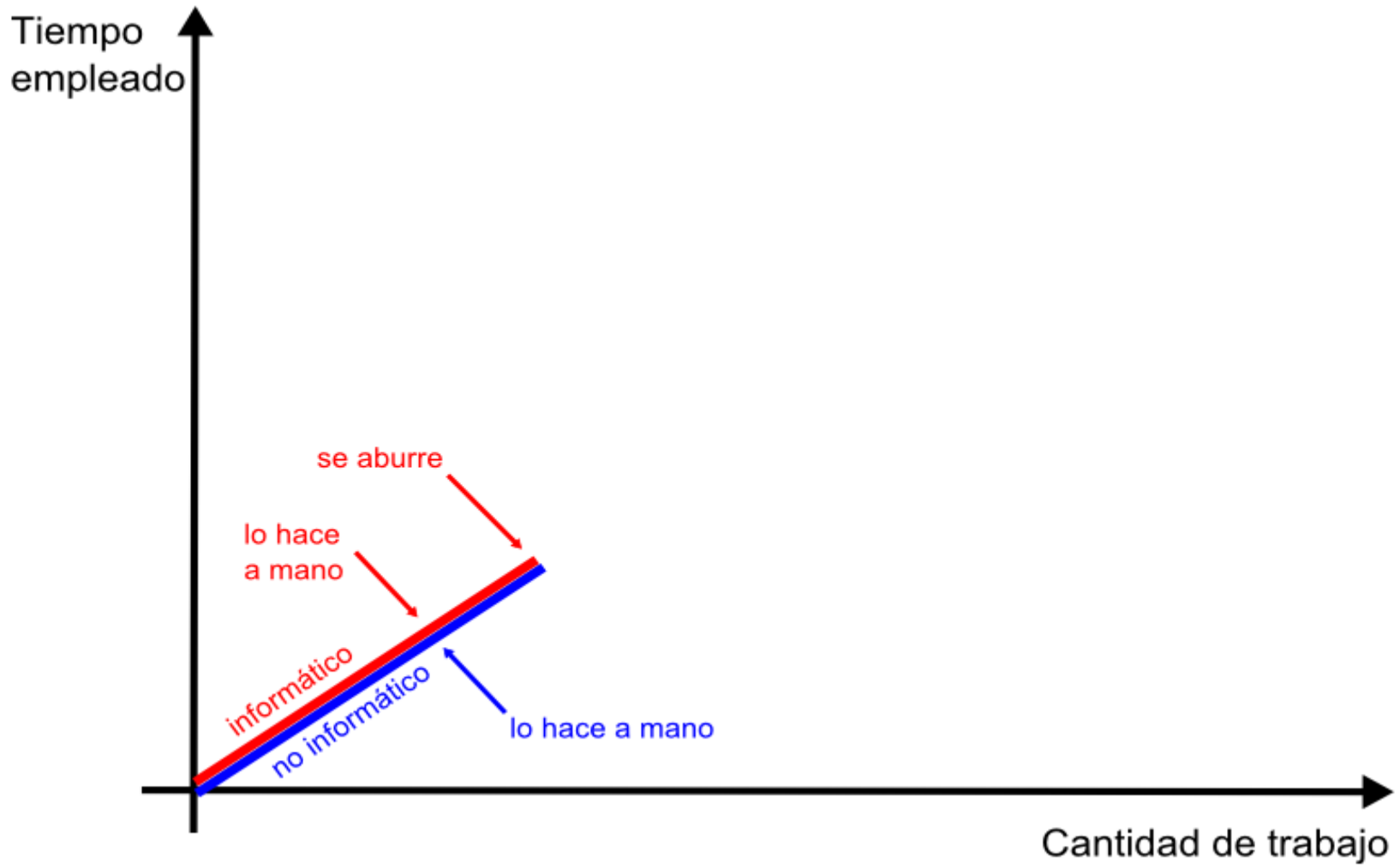
Ciencia de datos



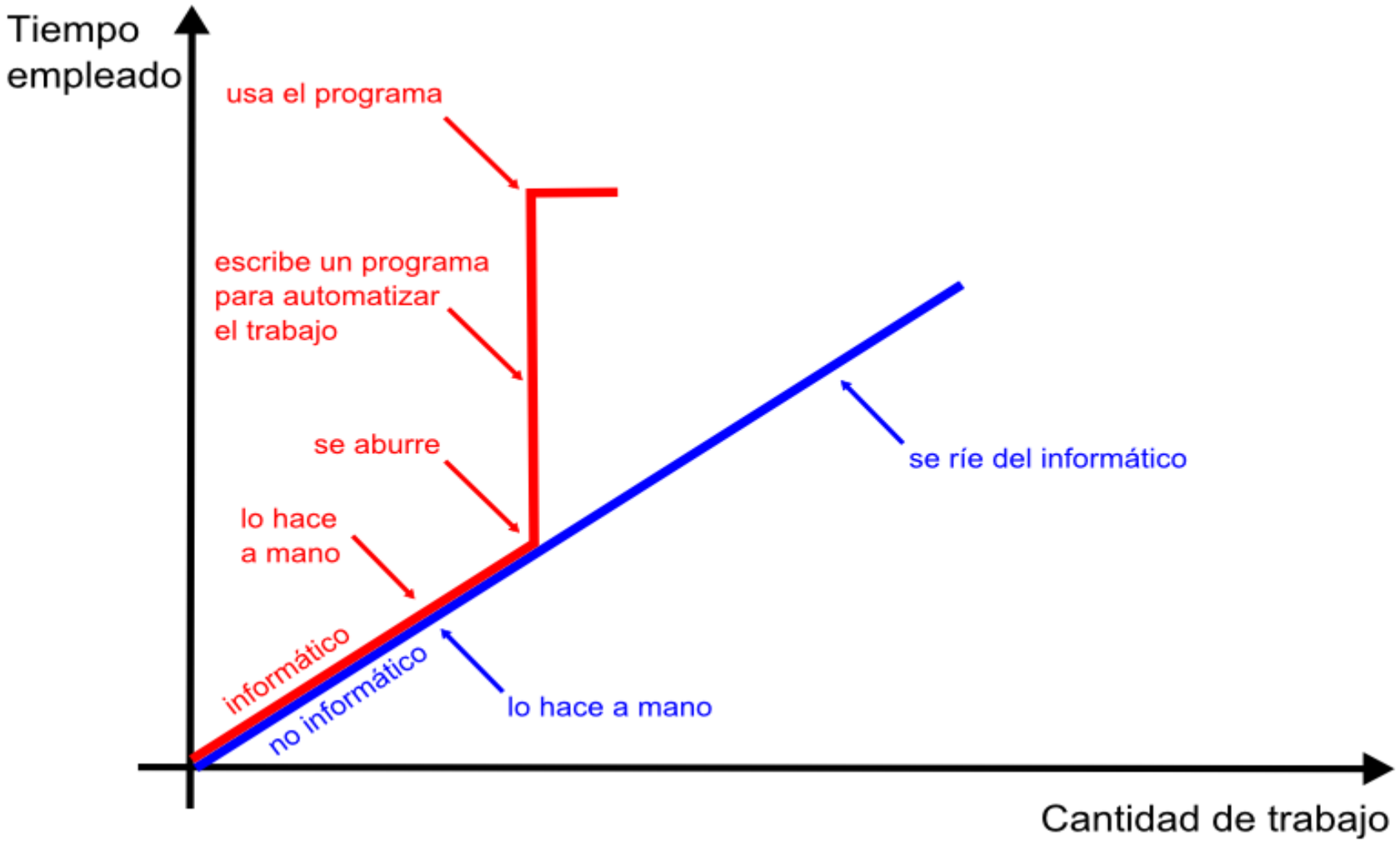
https://es.r4ds.hadley.nz/explora-introduccion.html

¿Por qué programar?

Cuando un trabajo se hace a manualmente, el tiempo empleado para realizar un trabajo suele ser directamente proporcional a la cantidad de trabajo

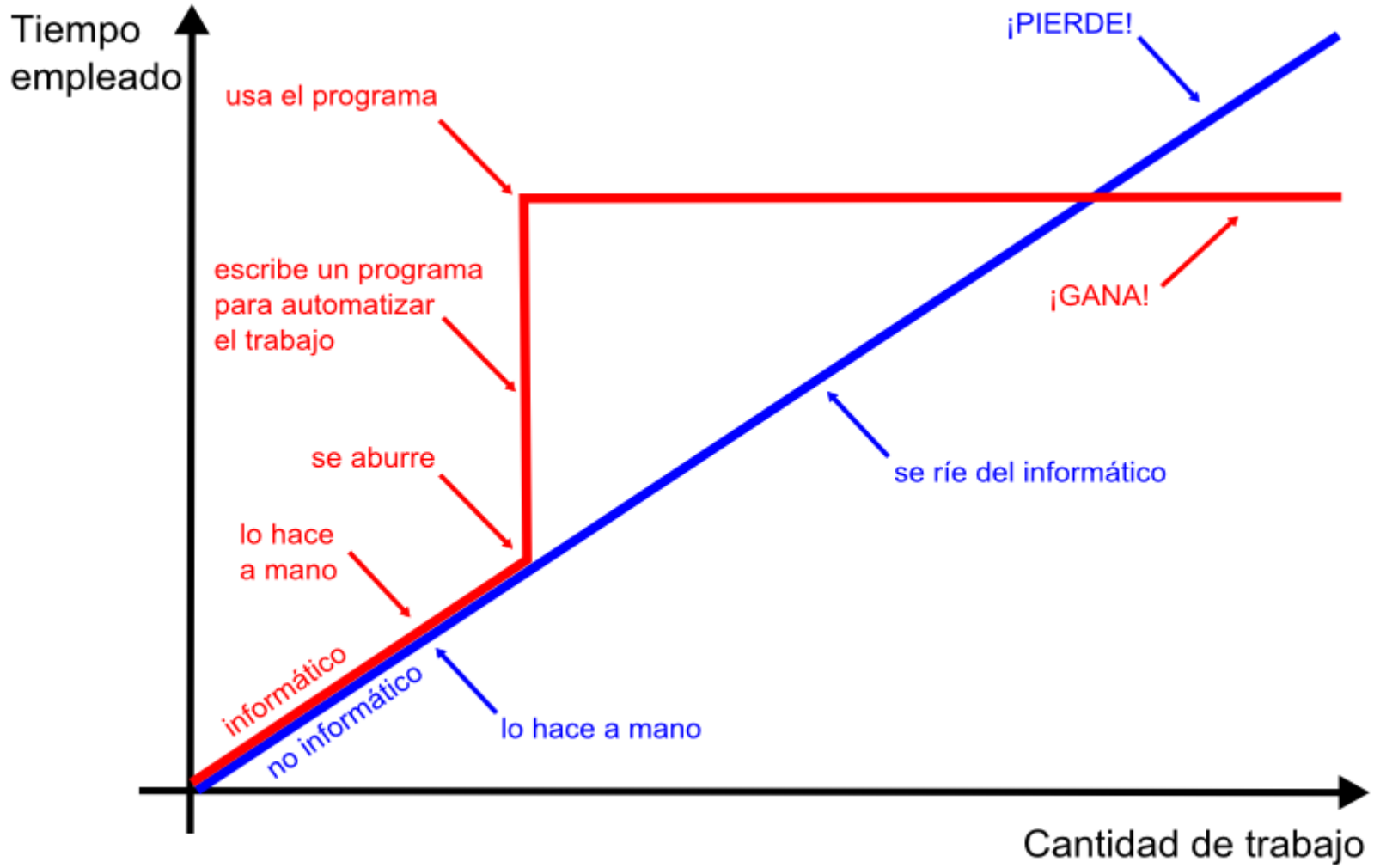


http://mclibre.org/consultar/python/otros/lenguajes-programacion.html



http://mclibre.org/consultar/python/otros/lenguajes-programacion.html

Al final, el informático puede hacer más trabajo sin necesidad de invertir más tiempo, mientras que el no informático descubre que no puede competir.



http://mclibre.org/consultar/python/otros/lenguajes-programacion.html

La programación nos permite automatizar tareas

Nos permite manejar bases de datos más grandes

Los paquetes permiten explorar varias funcionalidades de forma relativamente sencilla

Múltiples formas de comunicar resultados:

con R podes crear reportes en distintos formatos, paginas web, mapas, gráficos, libros, posters, presentaciones, etc..

Es software libre

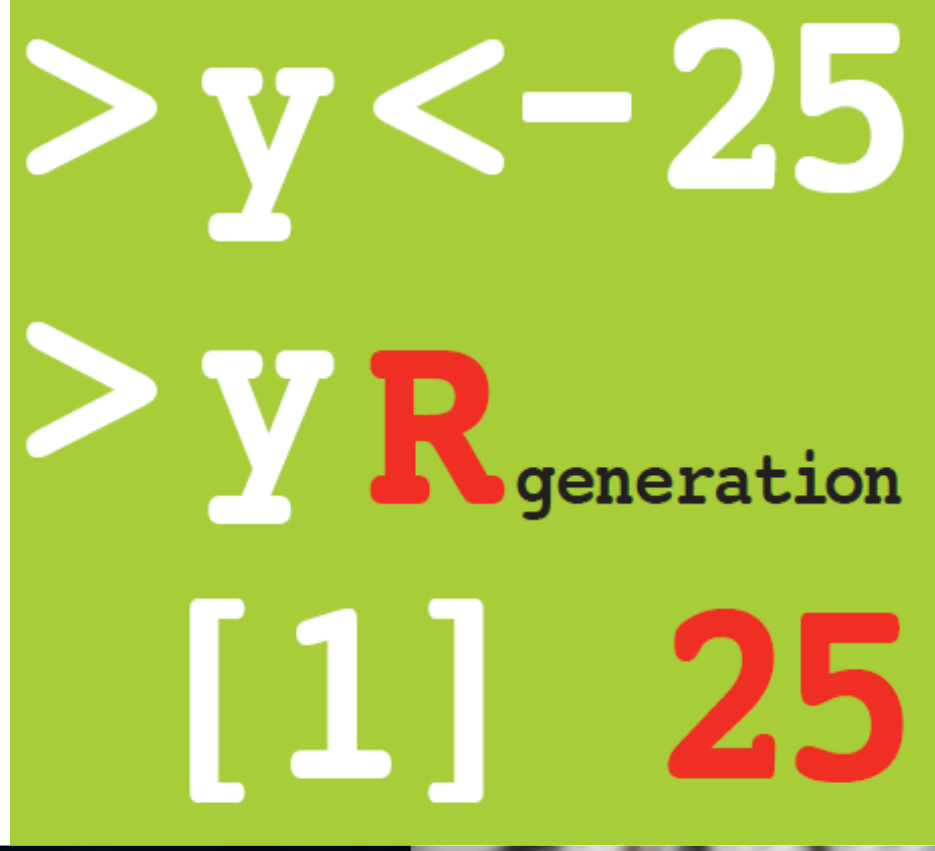
R **lingua franca** de la estadística

R es un lenguaje y entorno para computación estadística y gráficos.

https://www.r-project.org/about.html

R cumplió 25 años

1993 <- Nace
2000 <- Se libera la versión 1.0.0,
la primera de uso público



Thieme, N. (2018). R generation. Significance, 15(4), 14–19.

Nace una nueva generación de usuarios menos preocupados en la mecánica de R y más en las posibilidades que lo que R les permite hacer.

Serie de paquetes conocidos como «tidyverse» que vuelve «más simple», la limpieza y visualización de los datos.

Thieme, N. (2018). R generation. Significance, 15(4), 14–19.

R y RStudio

R y RStudio

https://www.r-project.org/about.html

R: motor

RStudio: tablero

Ismay C, Kim A. (2019) Chapter 2: Getting Started with Data in R. En: Moderndive. Statistical Inference via Data Science. A moderndive into R and the tidyverse

R es un lenguaje de programación que ejecuta cómputos.

Software gratuito y de código abierto

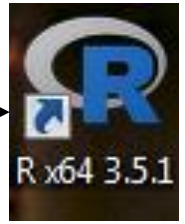
RStudio es un entorno de desarrollo integrado (IDE) que proporciona una interfaz al agregar muchas funciones y herramientas convenientes

Vamos a usar R desde RStudio

Para instalar R y R-Studio pueden usar el link que esta en la página del curso.

Cuando lo instalamos en nuestra máquina, se presentan dos íconos

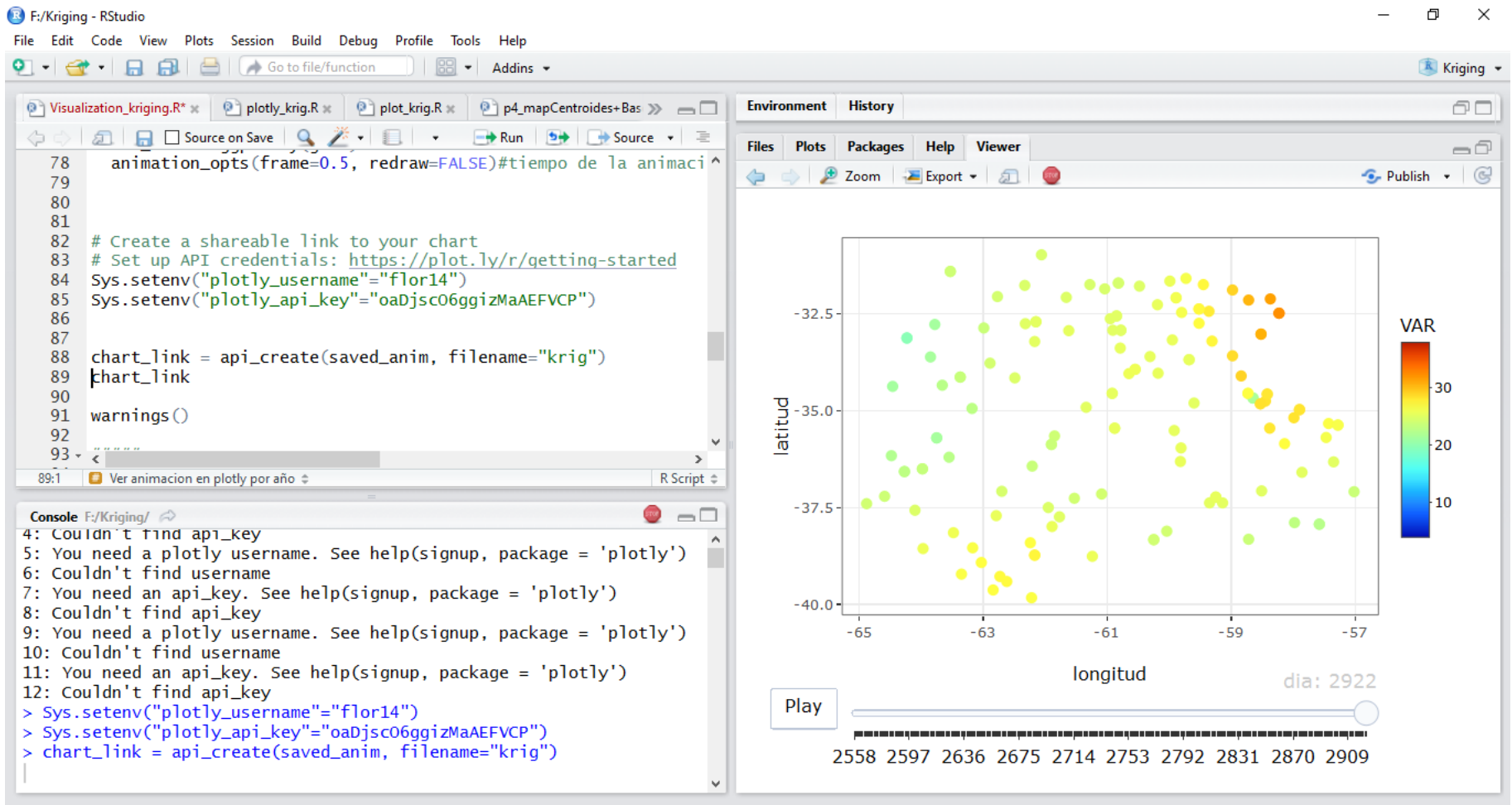
No abrir este ícono

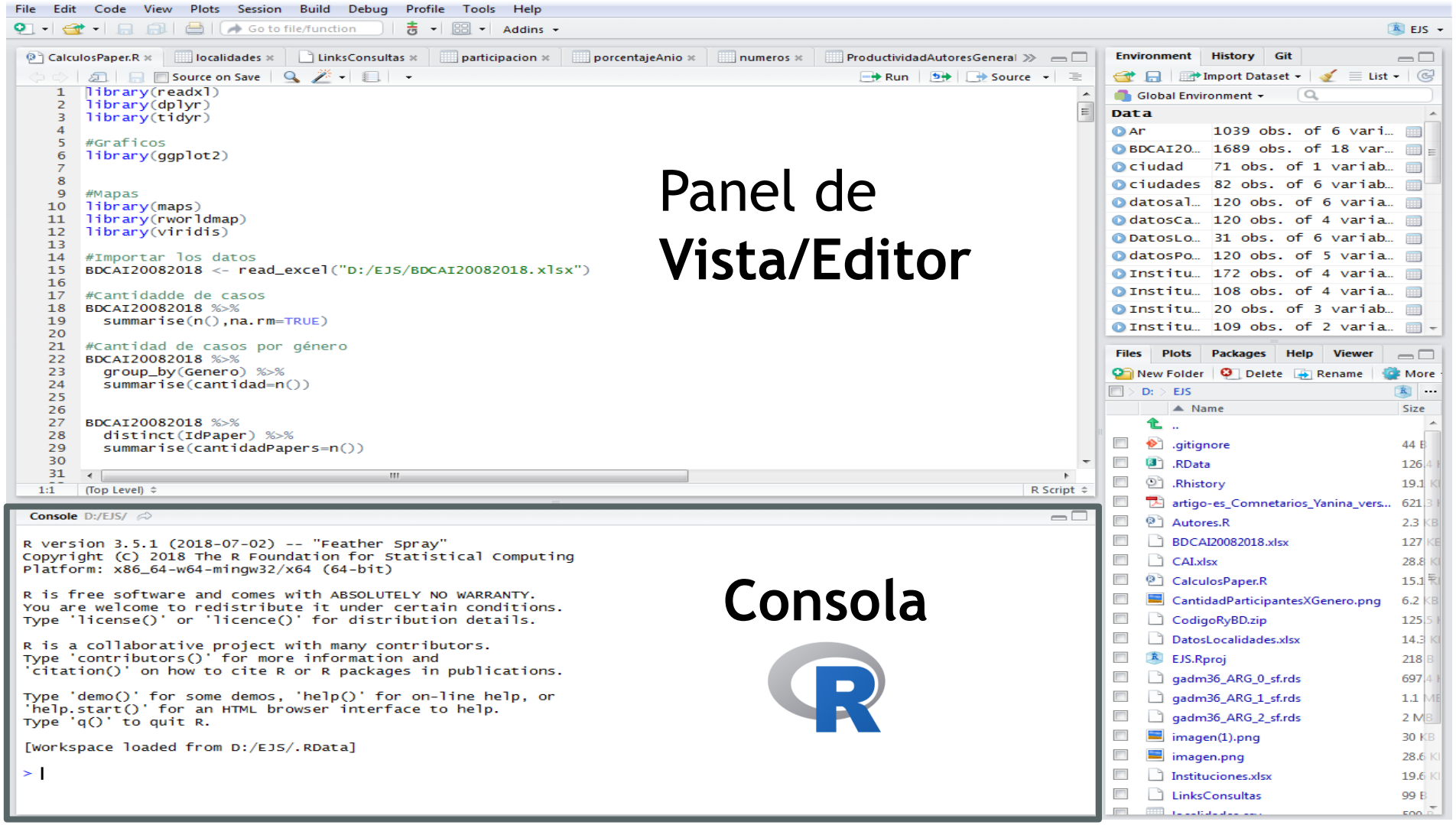


Abrir este ícono

RStudio (IDE)

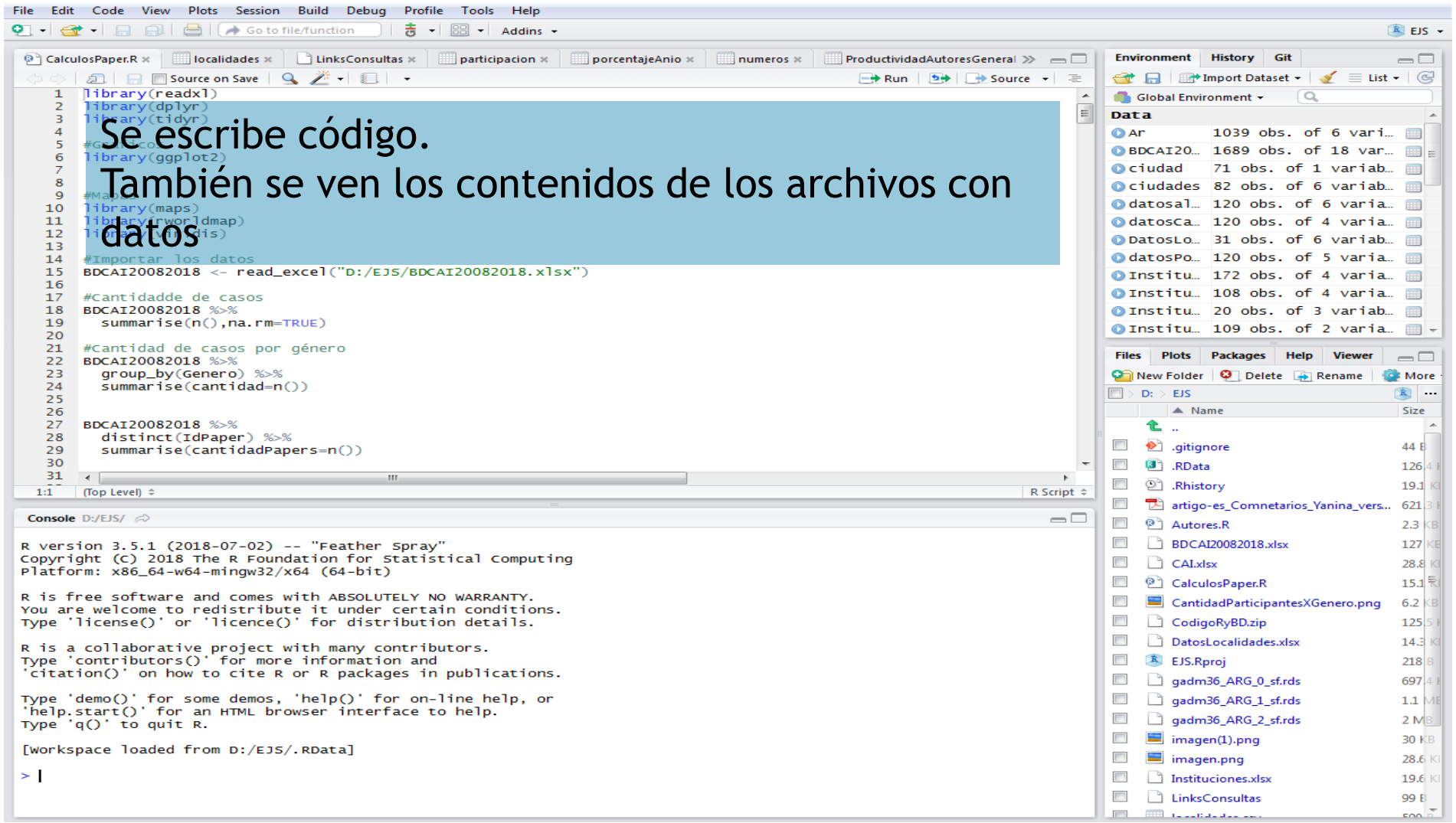
entorno de desarrollo integrado para R.





Panel
Ambiente
Historial

Panel
Archivos
Gráficos
Paquetes
Ayuda
Visor



Se escribe código.
También se ven los contenidos de los archivos con datos

Objetos en memoria (tablas, variables)

Aquí se ven archivos, gráficos, ayuda, paquetes instalados

Resultados y salidas del código

Paquetes

Paquetes

Los paquetes son las unidades fundamentales de código reproducible en R

- Típicamente, un paquete incluirá código (¡puede no ser solo de R!)
- Documentación del paquete y de las funciones internas,
- Algunas pruebas o tests para verificar que todo funcione como debería
- Conjuntos de datos

http://r-pkgs.had.co.nz/package.html

Funciones

Grupo de **sentencias** bajo un **mismo nombre** que realizan **una tarea específica**

doblar(izquierda, 90 grados)

Nombre

Argumentos

¿Cuál es la diferencia entre un paquete y una librería?

Una biblioteca o ***library*** es simplemente un directorio que contiene paquetes instalados.

Puedes tener múltiples bibliotecas en tu computadora.

http://r-pkgs.had.co.nz/package.html

paquete

Usas la función **library()** para llamar un paquete de la librería

Analogía en tweet de H. Wickham

Desarrollo de paquetes

Rol

Autores

Mantenimiento

Los paquetes que no se mantienen pueden quedar obsoletos o huérfanos

Colaboradores

http://r-pkgs.had.co.nz/package.html

ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics

A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Version:	3.2.0
Depends:	R (≥ 3.2)
Imports:	digest, grDevices, grid, gtable (≥ 0.1.1), lazyeval, MASS, mgcv, reshape2, rlang (≥ 0.3.0), scales (≥ 0.5.0), stats, tibble, viridisLite, withr (≥ 2.0.0)
Suggests:	covr, dplyr, ggplot2movies, hexbin, Hmisc, knitr, lattice, mapproj, maps, maptools, multcomp, munsell, nlme, profvis, quantreg, rgeos, rmarkdown, rpart, sf (≥ 0.7-3), svglite (≥ 1.2.0.9001), testthat (≥ 0.11.0), vdiffr (≥ 0.3.0)
Enhances:	sp
Published:	2019-06-16
Author:	Hadley Wickham [aut, cre], Winston Chang [aut], Lionel Henry [aut], Thomas Lin Pedersen [aut], Kohske Takahashi [aut], Claus Wilke [aut], Kara Woo [aut], Hiroaki Yutani [aut], RStudio [cph]
Maintainer:	Hadley Wickham <hadley at rstudio.com>
BugReports:	https://github.com/tidyverse/ggplot2/issues
License:	GPL-2 | file LICENSE
URL:	http://ggplot2.tidyverse.org, https://github.com/tidyverse/ggplot2
NeedsCompilation:	no
Citation:	ggplot2 citation info
Materials:	README NEWS
In views:	Graphics, Phylogenetics, TeachingStatistics
CRAN checks:	ggplot2 results

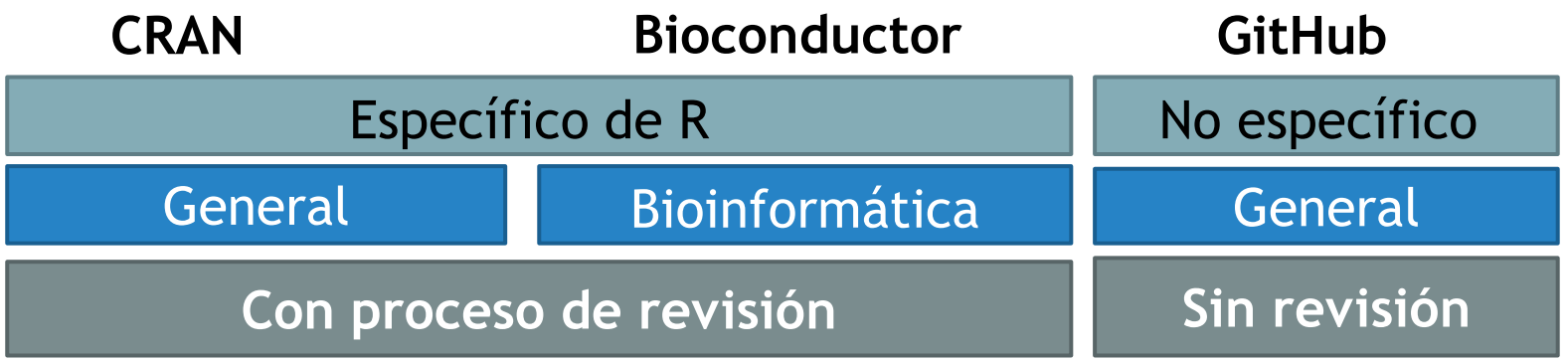
Downloads:

Reference manual:	ggplot2.pdf
Vignettes:	Extending ggplot2 Aesthetic specifications Profiling Performance

¿Qué es un repositorio?

Un repositorio es un lugar donde se encuentran los paquetes para que puedas instalarlos desde allí.

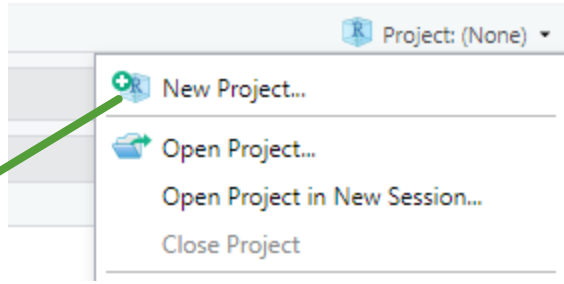
Están en línea y son accesibles para todos.

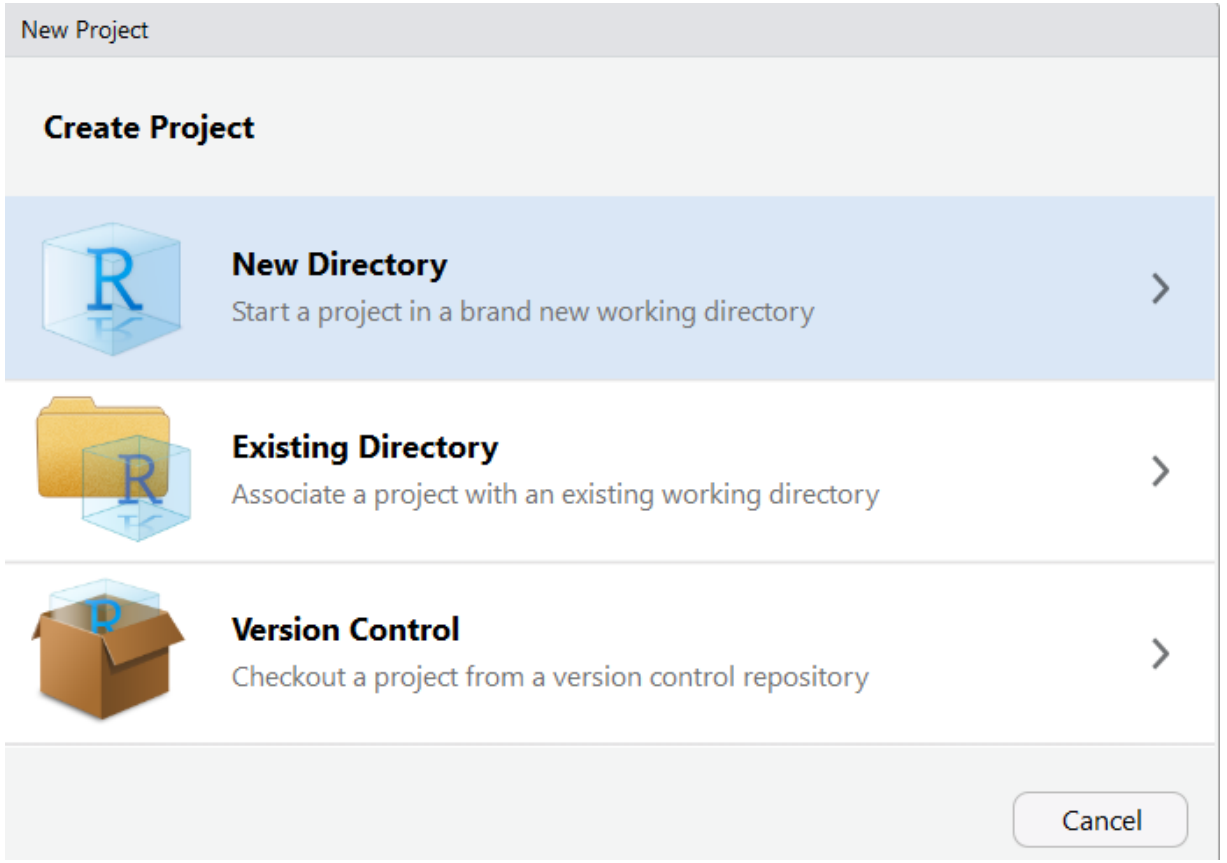


Desarrollo de paquetes por la comunidad de usuarios

> **Paquetes muy específicos**

Uso de proyectos



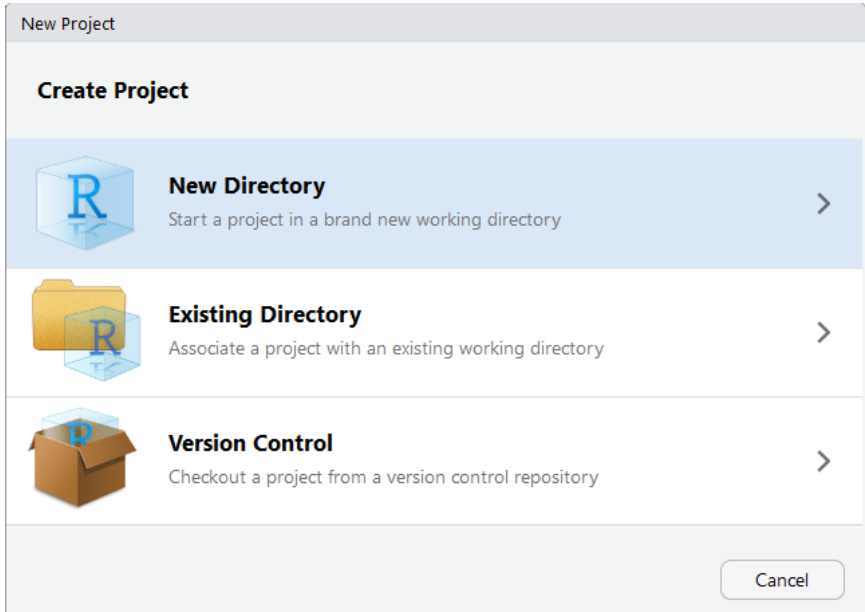


https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects

Trabajar con proyectos

- Hace que sea más sencillo compartir código con otra persona
- Permiten cargar fácilmente código con el envío de un manuscrito
- Hacen que sea más fácil retomar el proyecto después de un descanso.

Trabajar empleando proyectos no sólo vuelve a tu ciencia reproducible, sino que también hace tu vida fácil



https://swcarpentry.github.io/r-novice-gapminder/02-project-intro/

¡Manos a la obra!

- Diferencia entre R y RStudio
- Abrir un proyecto de trabajo
- Instalar y llamar un paquete en R

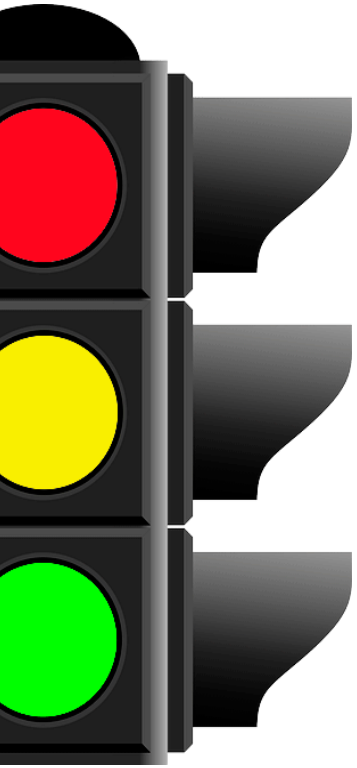
https://flor14.github.io/cai_2019 -> **r_rstudio_proyectos.R**

```
Error in xy.coords(x, y, xlabel, ylabel, log) : object '
```

¿Abandono R?

¡NO!

Mensajes de R



Errores:
Aparecerá "Error in…" y tratará de explicar qué fue lo que salió mal.
Generalmente cuando hay un error, **el código no se ejecutará.**

Advertencias:
Aparecerá "Warning" y tratará de explicar por qué hay una advertencia.
En general, **el código seguirá funcionando, pero hay algo a revisar o estar atentos.**

Mensajes:
Cuando el texto en rojo **no comienza con "Error in" o «Warning** mensaje amistoso.
Son mensajes de diagnóstico útiles y no impiden que el código funcione.

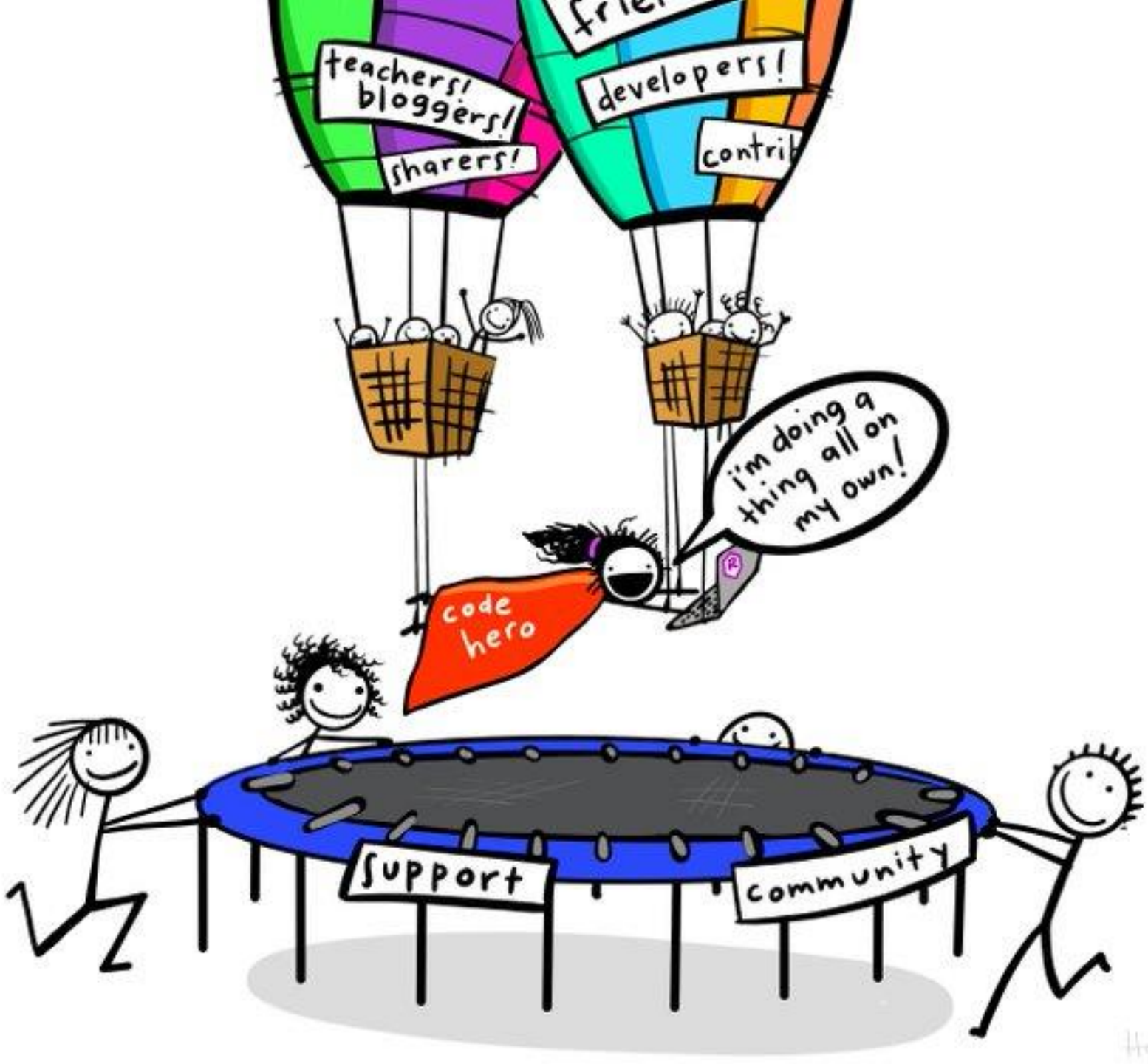
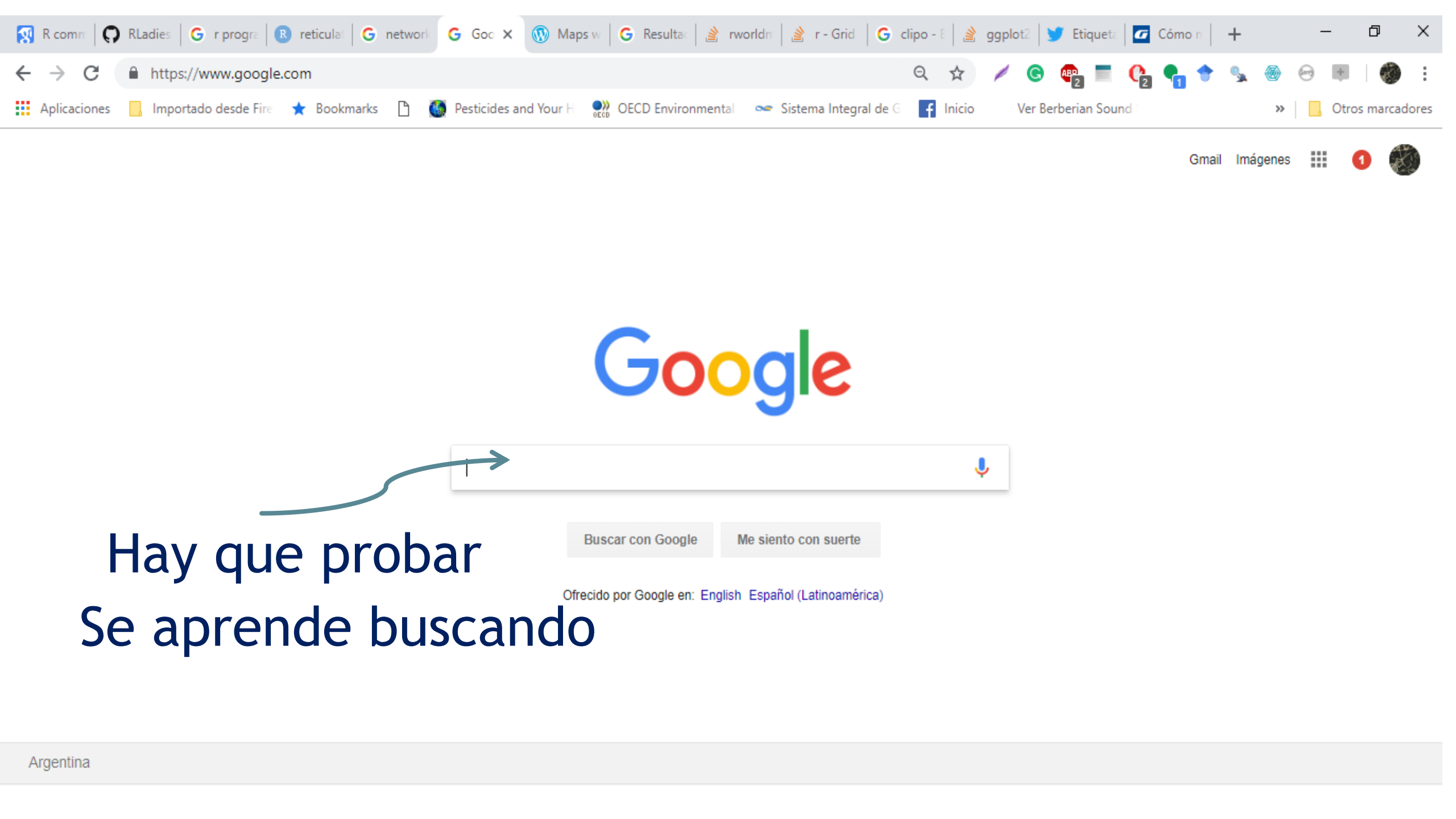


Imagen por **Allison Horst**
twitter.com/allison_horst



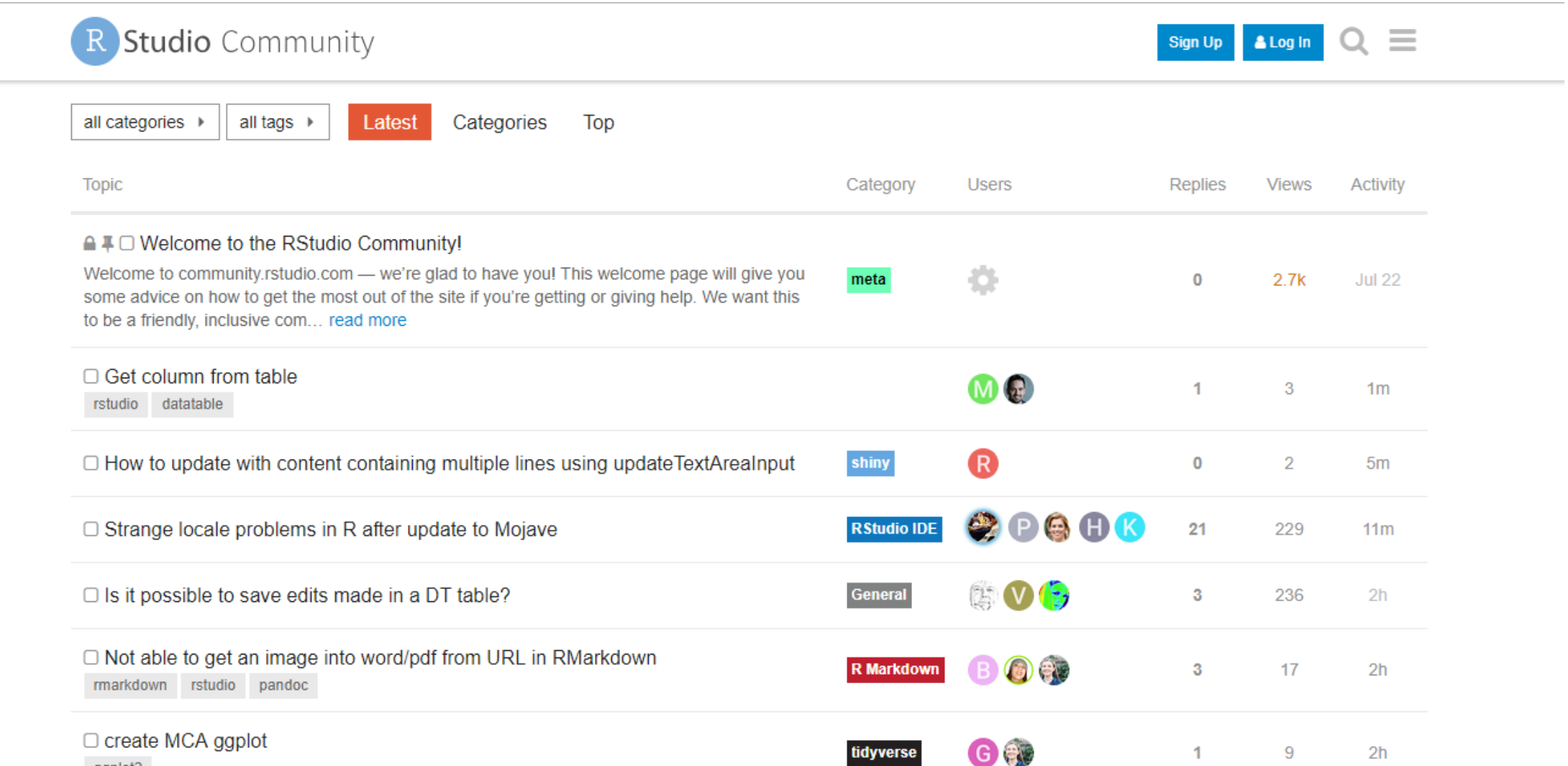
Hay que probar

Se aprende buscando

RStudio Community

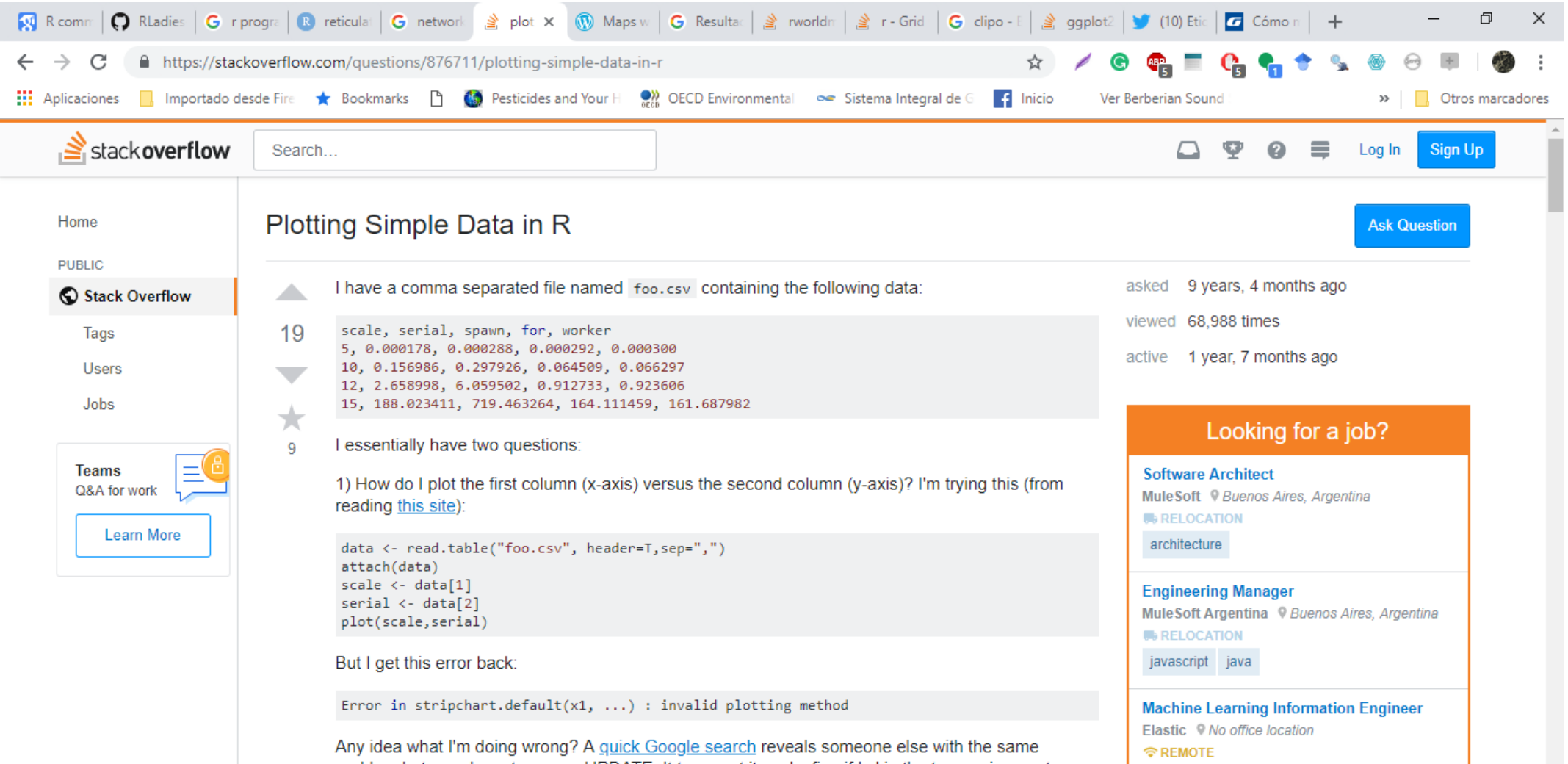
FORO

https://community.rstudio.com/



StackOverflow (no es solo de R)

FORO
https://stackoverflow.com/



Twitter

Seguir cuentas de Twitter sobre #rstatsES o #rstats

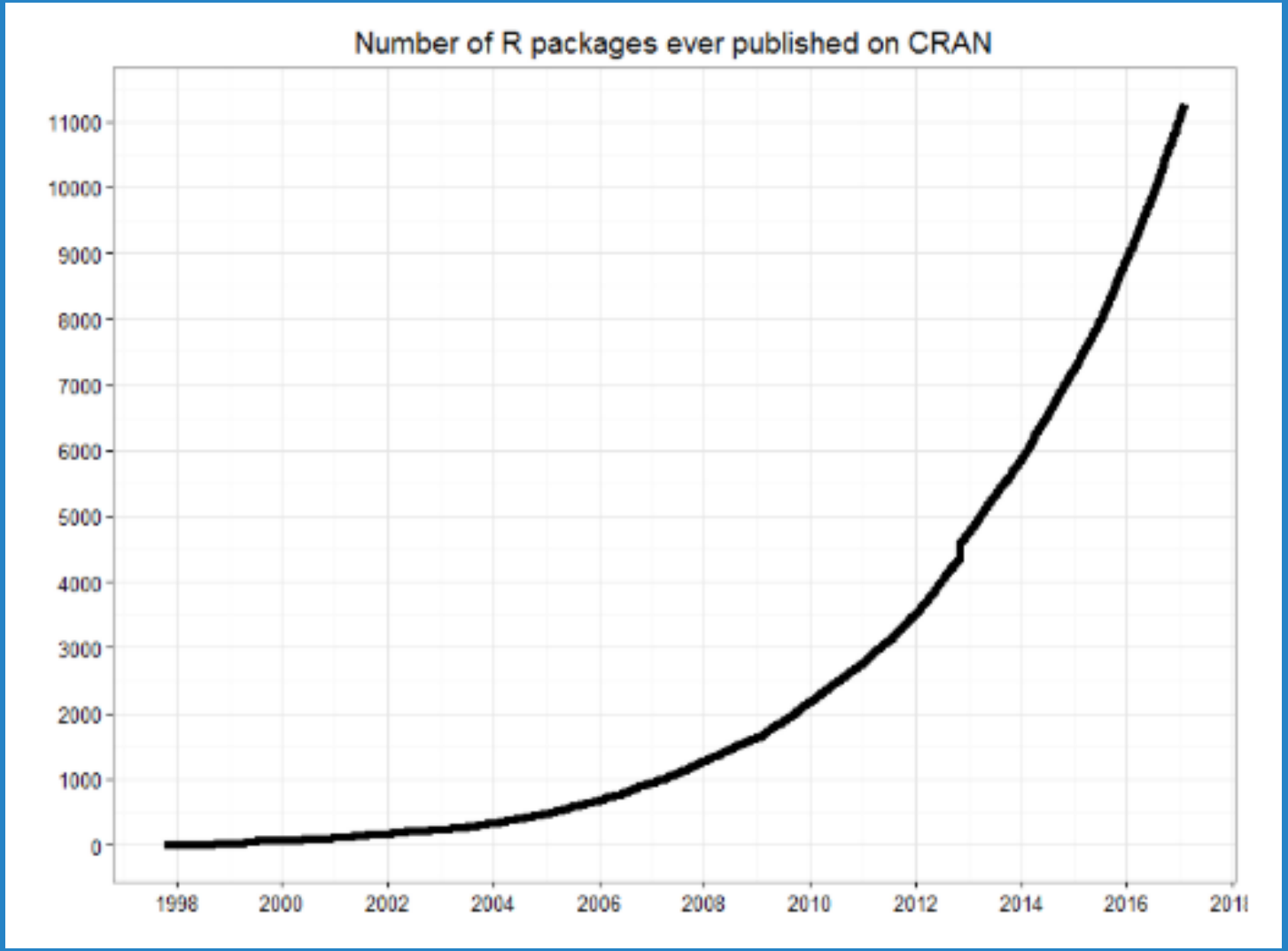
- Nuevos paquetes
- Actualizaciones
- Funciones útiles
- Conferencias (UseRs y RStudioConf)

https://twitter.com/rweekly_org

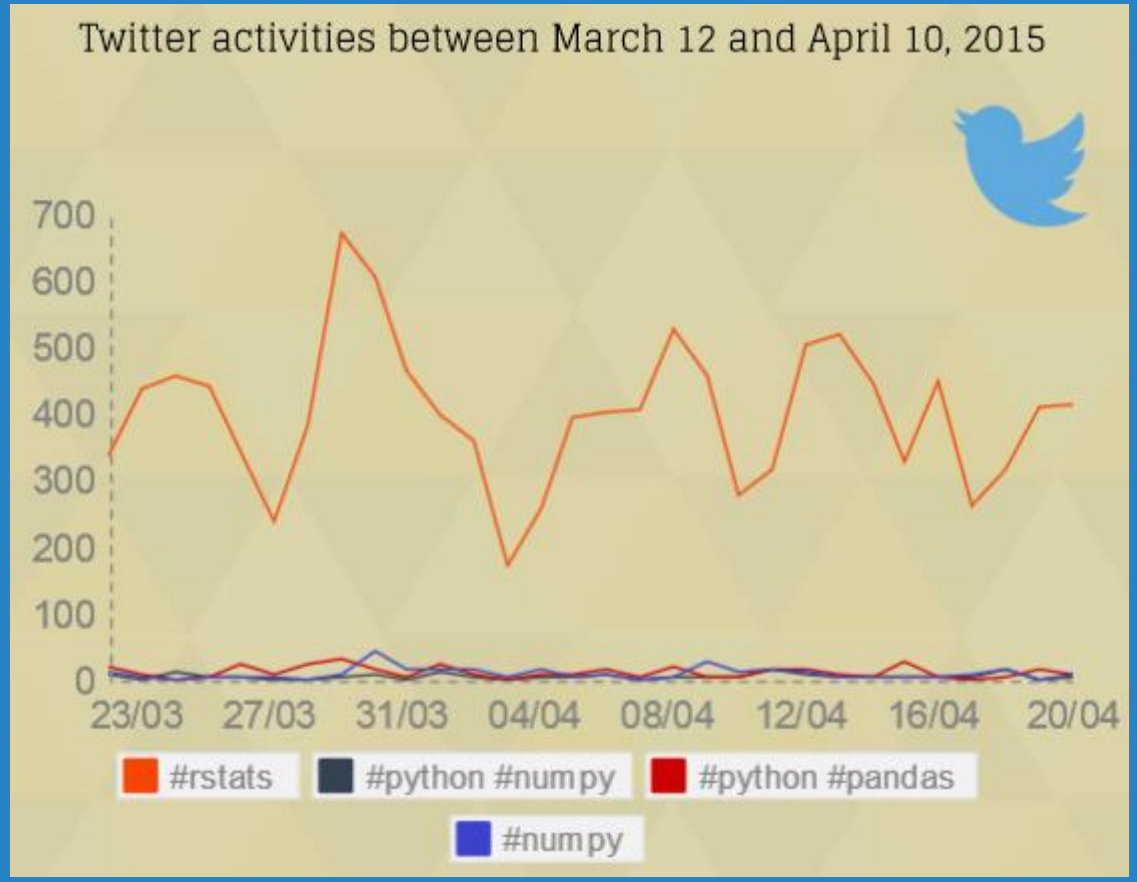
https://twitter.com/dataandme

Información

Avanza MUY rápido



La comunidad de R es twittera



Tidyverse

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Proje

Untitled1* | diamantes | diamonds

Source on Save | Run | Source

```
53
54
55 # transparencia ----------------------------------------
56
57 plot(diamonds$carat, diamonds$price)
58
59
60
61
62
63
64
65
```

64:1 | transparencia | R Script

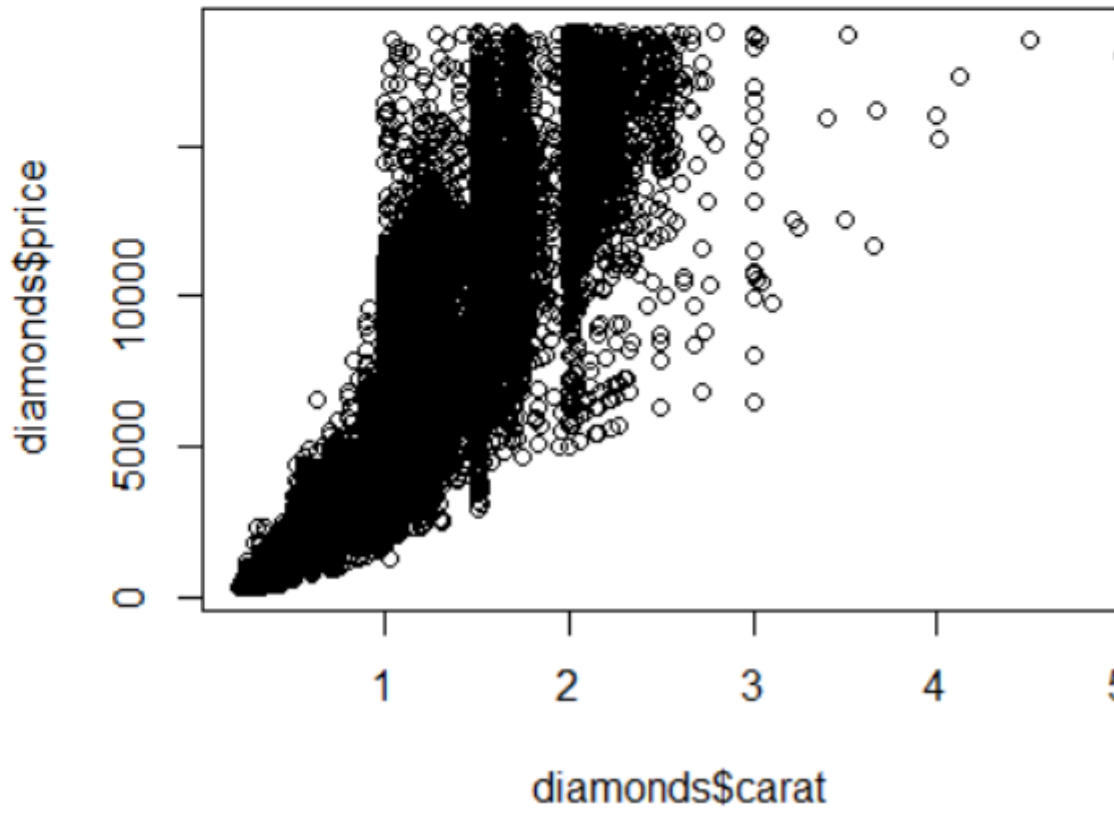
Console | Terminal | Markers

```
4 0.290 Premi~ I     VS2     62.4    58    334   4.2   4.23  2.63
5 0.31  Good   J     SI2     63.3    58    335   4.34  4.35  2.75
6 0.24  Very ~ J     VVS2    62.8    57    336   3.94  3.96  2.48
> write.table(head(diamonds), "D:/Desktop/diamantes.txt", sep = " "
)
> diamantes <- read.csv("D:/Desktop/diamantes.txt", sep="")
>   View(diamantes)
> plot(diamonds$carat, diamonds$price)
>
```

Environment | History | Connections

Files | Plots | Packages | Help | Viewer

Zoom | Export



stackoverflow Search...

Home

PUBLIC

Stack Overflow

Tags

Users

Any way to make plot points in scatterplot more transparent in R?

44

I have a 3 column matrix; plots are made by points based on column 1 and column 2 values, but colored based on column 2 (6 different groups). I can successfully plot all points, however, the last plot group (group 6) which was assigned the color purple, masks the plots of the other groups. Is there a way to make the plot points more transparent?

```
s <- read.table("/.../parse-output.txt", sep="\t")
dim(s)
[1] 67124     3
x <- s[,1]
y <- s[,2]
z <- s[,3]
cols <- cut(z, 6, labels = c("pink", "red", "yellow", "blue", "green", "purple"))
plot(x, y, main= "Fragment recruitment plot - FR-HIT", ylab = "Percent identity", xlab = "Bas
```

r plot ggplot2 alpha

share improve this question

edited Oct 22 '12 at 0:59
mnel
88.6k ● 14 ● 212 ● 219

asked Oct 21 '12 at 6:58
Steve
387 ● 1 ● 5 ● 12

add a comment

Esta pregunta se acerca a lo que quiero resolver

https://stackoverflow.com/questions/12995683/any-way-to-make-plot-points-in-scatterplot-more-transparent-in-r

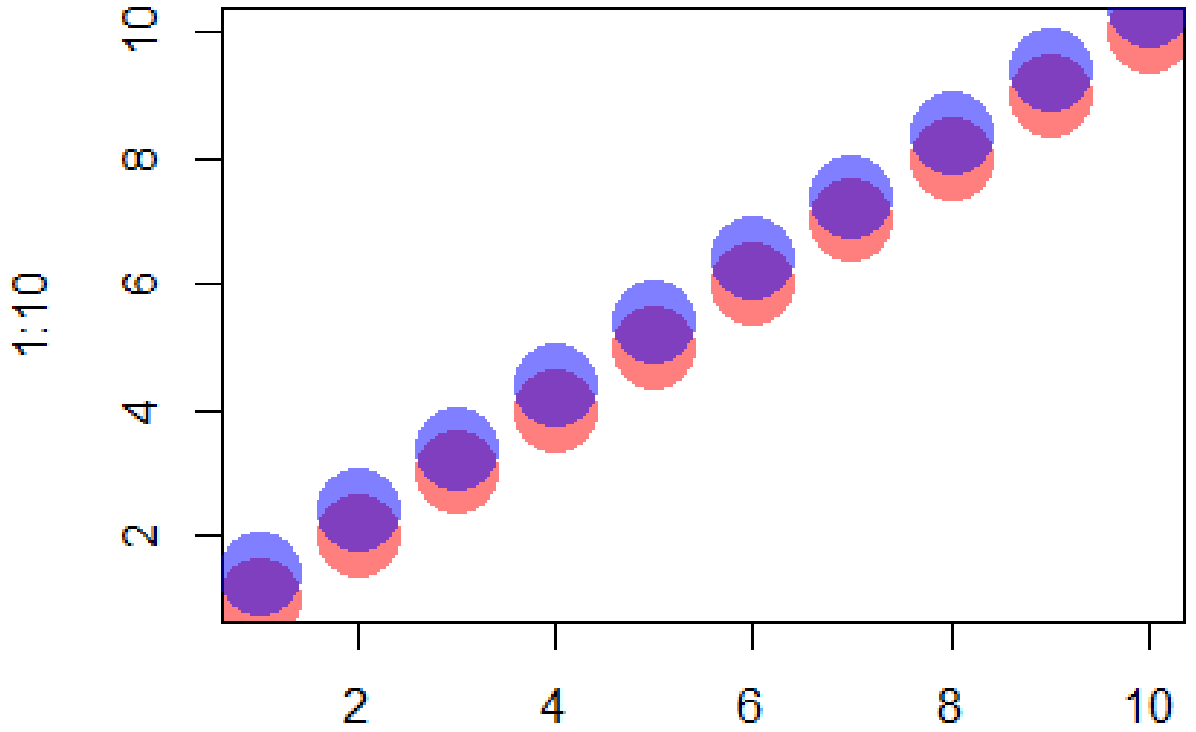
Creo que encontré una posible solución

Respuesta 1

32

When creating the colors, you may use `rgb` and set its `alpha` argument:

```
plot(1:10, col = rgb(red = 1, green = 0, blue = 0, alpha = 0.5),
     pch = 16, cex = 4)
points((1:10) + 0.4, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.5),
       pch = 16, cex = 4)
```



https://stackoverflow.com/questions/12995683/any-way-to-make-plot-points-in-scatterplot-more-transparent-in-r

Respuesta 2

10

If you decide to use `ggplot2`, you can set transparency of overlapping points using the `alpha` argument.

e.g.

```
library(ggplot2)
ggplot(diamonds, aes(carat, price)) + geom_point(alpha = 1/40)
```

share improve this answer

answered Oct 21 '12 at 7:44

Maiasaura

16.9k ● 16 ● 83 ● 99

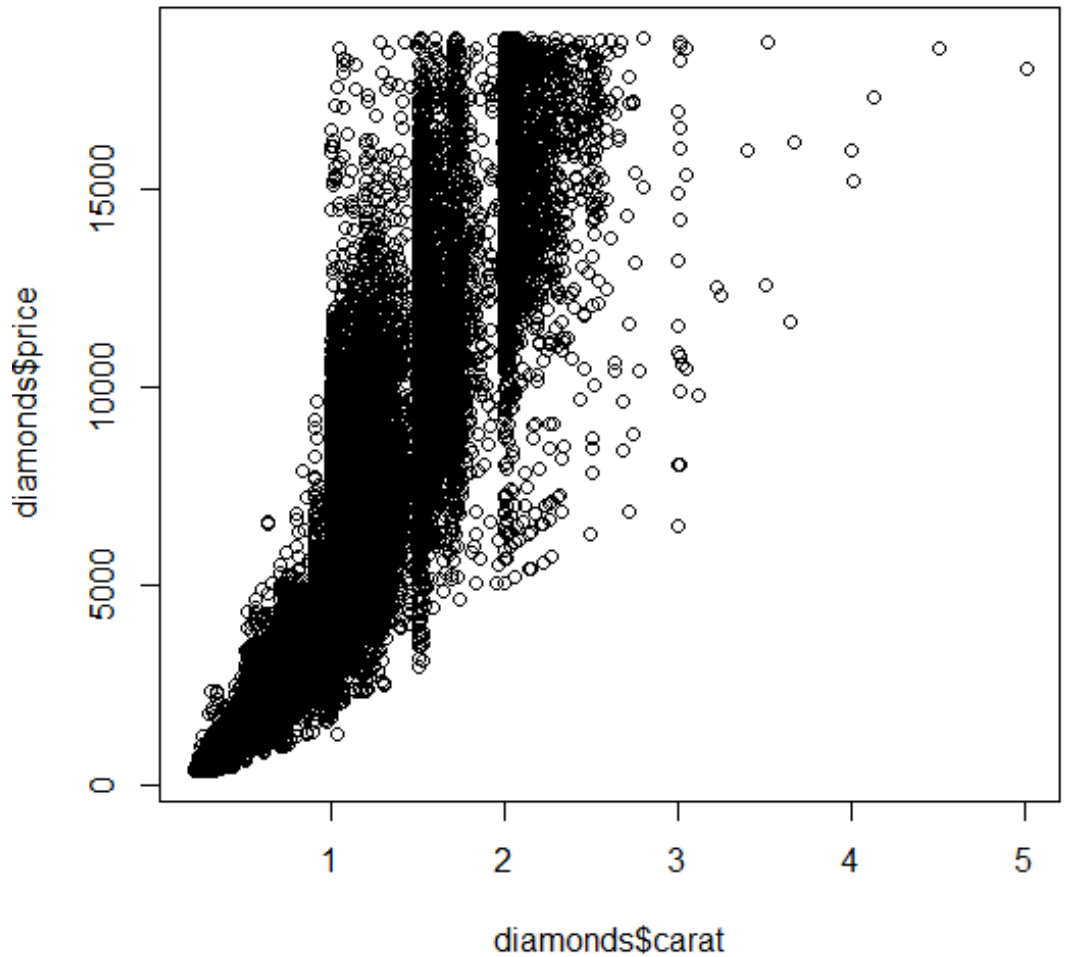
https://stackoverflow.com/questions/12995683/any-way-to-make-plot-points-in-scatterplot-more-transparent-in-r

plot()

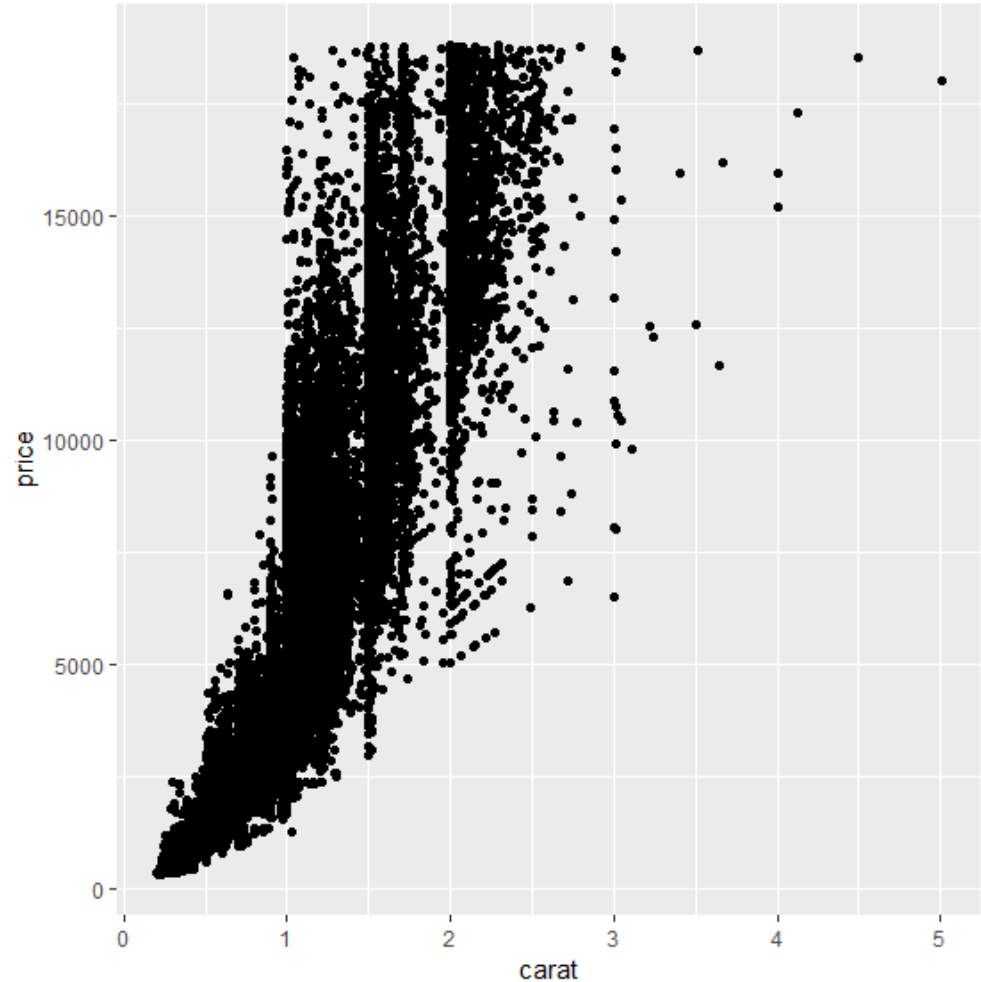
ggplot()

R base

tidyverse



R base



tidyverse

Mismo gráfico... escrito diferente

R base

```
plot(x = diamonds$carat, y = diamonds$price)
```

tidyverse

```
library(ggplot2)
ggplot(diamonds, aes(x = carat, y = price)) +
geom_point()
```

Mismo gráfico... escrito diferente

R base

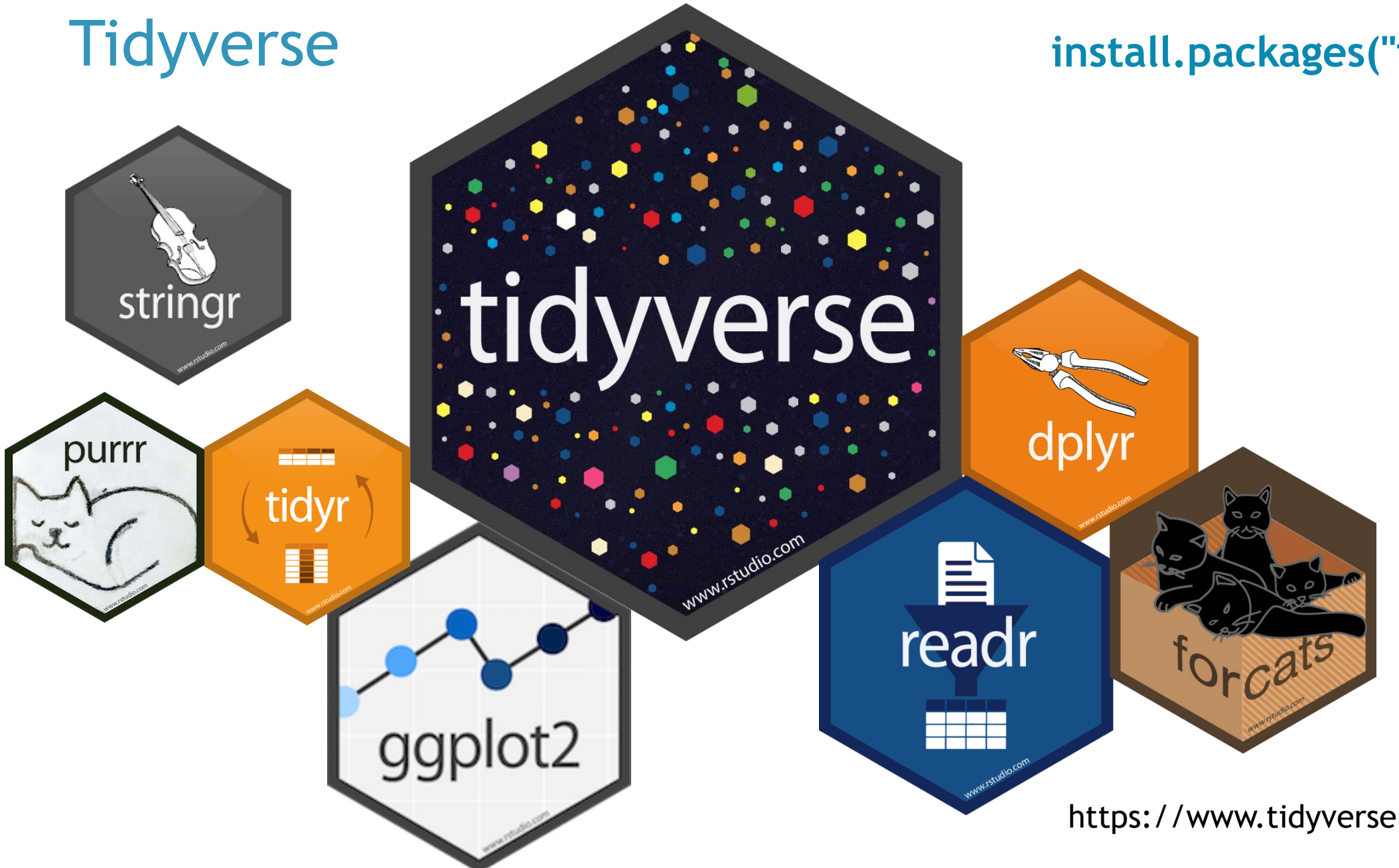
```
plot(diamonds$carat, diamonds$price)
```

tidyverse

```
library(ggplot2)
ggplot(diamonds, aes(x = carat,  y = price)) +
  geom_point()
```

Tidyverse

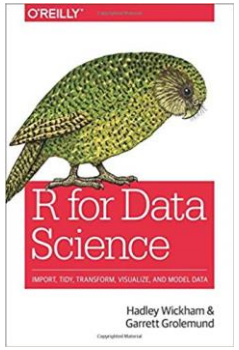
install.packages("tidyverse")



https://www.tidyverse.org/packages/

El **tidyverse** es una colección de paquetes R diseñados para ciencia de datos.

Todos los paquetes comparten una filosofía de diseño subyacente, gramática y estructuras de datos.



http://r4ds.had.co.nz/

Tidyverse

Tidyverse y r base pueden coexistir

- **ggplot2**, which implements the grammar of graphics. You can use it to visualize your data.

- **dplyr** is a grammar of data manipulation. You can use it to solve the most common data manipulation challenges.

- **tidyr** helps you to create tidy data or data where each variable is in a column, each observation is a row end each value is a cell.

- **readr** is a fast and friendly way to read rectangular data.

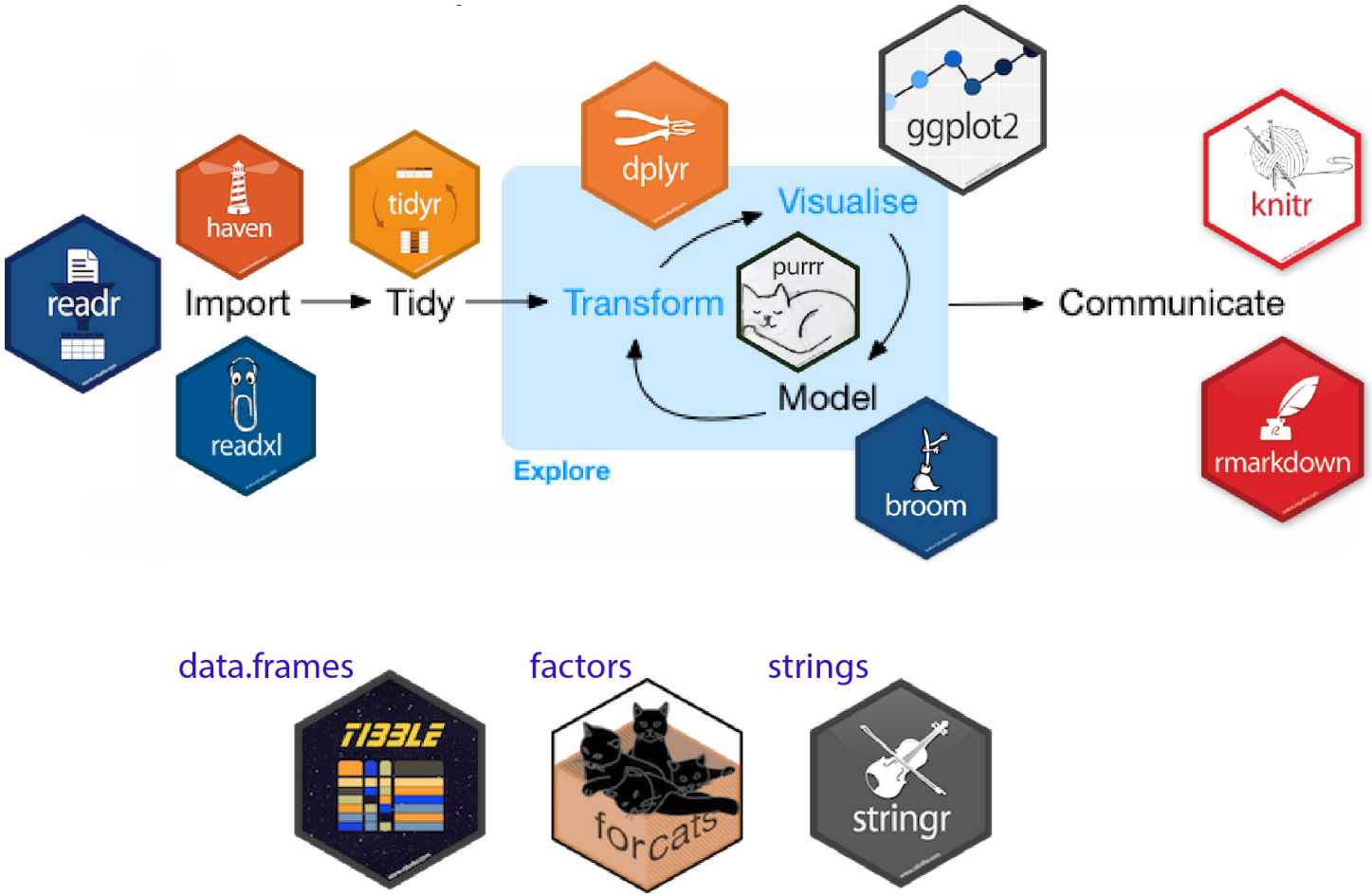
- **purrr** enhances R's functional programming (FP) toolkit by providing a complete and consistent set of tools for working with functions and vectors.

- **tibble** is a modern re-imaginging of the data frame.

- **stringr** provides a cohesive set of functions designed to make working with strings as easy as posssible

- **forcats** provide a suite of useful tools that solve common problems with factors.

Tidyverse
ejemplo



https://lsru.github.io/tv_course/lecture01_intro.html#1

Los paquetes del tidyverse se escriben diferente, tienen una sintaxis particular

R Syntax Comparison : : CHEAT SHEET

Dollar sign syntax

`goal(data$x, data$y)`

SUMMARY STATISTICS:

one continuous variable:
```
mean(mtcars$mpg)
```

one categorical variable:
```
table(mtcars$cyl)
```

two categorical variables:
```
table(mtcars$cyl, mtcars$am)
```

one continuous, one categorical:
```
mean(mtcars$mpg[mtcars$cyl==4])
mean(mtcars$mpg[mtcars$cyl==6])
mean(mtcars$mpg[mtcars$cyl==8])
```

PLOTTING:

one continuous variable:
```
hist(mtcars$disp)

boxplot(mtcars$disp)
```

one categorical variable:
```
barplot(table(mtcars$cyl))
```

two continuous variables:
```
plot(mtcars$disp, mtcars$mpg)
```

two categorical variables:
```
mosaicplot(table(mtcars$am, mtcars$cyl))
```

one continuous, one categorical:
```
histogram(mtcars$disp[mtcars$cyl==4])
histogram(mtcars$disp[mtcars$cyl==6])
histogram(mtcars$disp[mtcars$cyl==8])

boxplot(mtcars$disp[mtcars$cyl==4])
boxplot(mtcars$disp[mtcars$cyl==6])
boxplot(mtcars$disp[mtcars$cyl==8])
```

WRANGLING:

subsetting:
```
mtcars[mtcars$mpg>30, ]
```

making a new variable:
```
mtcars$efficient[mtcars$mpg>30] <- TRUE
mtcars$efficient[mtcars$mpg<30] <- FALSE
```

Formula syntax

`goal(y~x|z, data=data, group=w)`

SUMMARY STATISTICS:

one continuous variable:
```
mosaic::mean(~mpg, data=mtcars)
```

one categorical variable:
```
mosaic::tally(~cyl, data=mtcars)
```

two categorical variables:
```
mosaic::tally(cyl~am, data=mtcars)
```

one continuous, one categorical:
```
mosaic::mean(mpg~cyl, data=mtcars)
```

tilde

PLOTTING:

one continuous variable:
```
lattice::histogram(~disp, data=mtcars)

lattice::bwplot(~disp, data=mtcars)
```

one categorical variable:
```
mosaic::bargraph(~cyl, data=mtcars)
```

two continuous variables:
```
lattice::xyplot(mpg~disp, data=mtcars)
```

two categorical variables:
```
mosaic::bargraph(~am, data=mtcars, group=cyl)
```

one continuous, one categorical:
```
lattice::histogram(~disp|cyl, data=mtcars)

lattice::bwplot(cyl~disp, data=mtcars)
```

The variety of R syntaxes give you many ways to "say" the same thing

read across the cheatsheet to see how different syntaxes approach the same problem

Tidyverse syntax

`data %>% goal(x)`

SUMMARY STATISTICS:

one continuous variable:
```
mtcars %>% dplyr::summarize(mean(mpg))
```

one categorical variable:
```
mtcars %>% dplyr::group_by(cyl) %>%
  dplyr::summarize(n())
```

the pipe

two categorical variables:
```
mtcars %>% dplyr::group_by(cyl, am) %>%
  dplyr::summarize(n())
```

one continuous, one categorical:
```
mtcars %>%  dplyr::group_by(cyl) %>%
   dplyr::summarize(mean(mpg))
```

PLOTTING:

one continuous variable:
```
ggplot2::qplot(x=mpg, data=mtcars, geom = "histogram")

ggplot2::qplot(y=disp, x=1, data=mtcars, geom="boxplot"
```

one categorical variable:
```
ggplot2::qplot(x=cyl, data=mtcars, geom="bar")
```

two continuous variables:
```
ggplot2::qplot(x=disp, y=mpg, data=mtcars, geom="point"
```

two categorical variables:
```
ggplot2::qplot(x=factor(cyl), data=mtcars, geom="bar")
       facet_grid(.~am)
```

one continuous, one categorical:
```
ggplot2::qplot(x=disp, data=mtcars, geom = "histogram"
       facet_grid(.~cyl)

ggplot2::qplot(y=disp, x=factor(cyl), data=mtcars,
               geom="boxplot")
```

WRANGLING:

subsetting:
```
mtcars %>% dplyr::filter(mpg>30)
```

making a new variable:
```
mtcars <- mtcars %>%
dplyr::mutate(efficient = if_else(mpg>30,  TRUE, FALSE
```

Algunos paquetes usan pipes

%>%
es un operador que nos permite encadenar las acciones que queremos aplicar

La mayoría de los paquetes del **tidyverse** usa pipes

https://es.r4ds.hadley.nz/pipes.html

Pipes

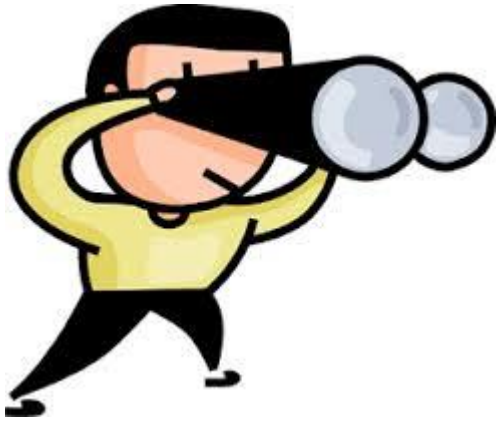

```
Yo %>%                    dataset %>%
  despertar() %>%           funcion1() %>%
  duchar() %>%              funcion2() %>%
  vestir() %>%              funcion3() %>%
  desayunar() %>%           funcion4() %>%
  trabajar()                funcion5()
```

Los pipes son mayoritariamente usados para reescribir una secuencia lineal corta de operaciones.

https://twitter.com/WeAreRLadies/status/1172576445794803713?s=19

Entonces

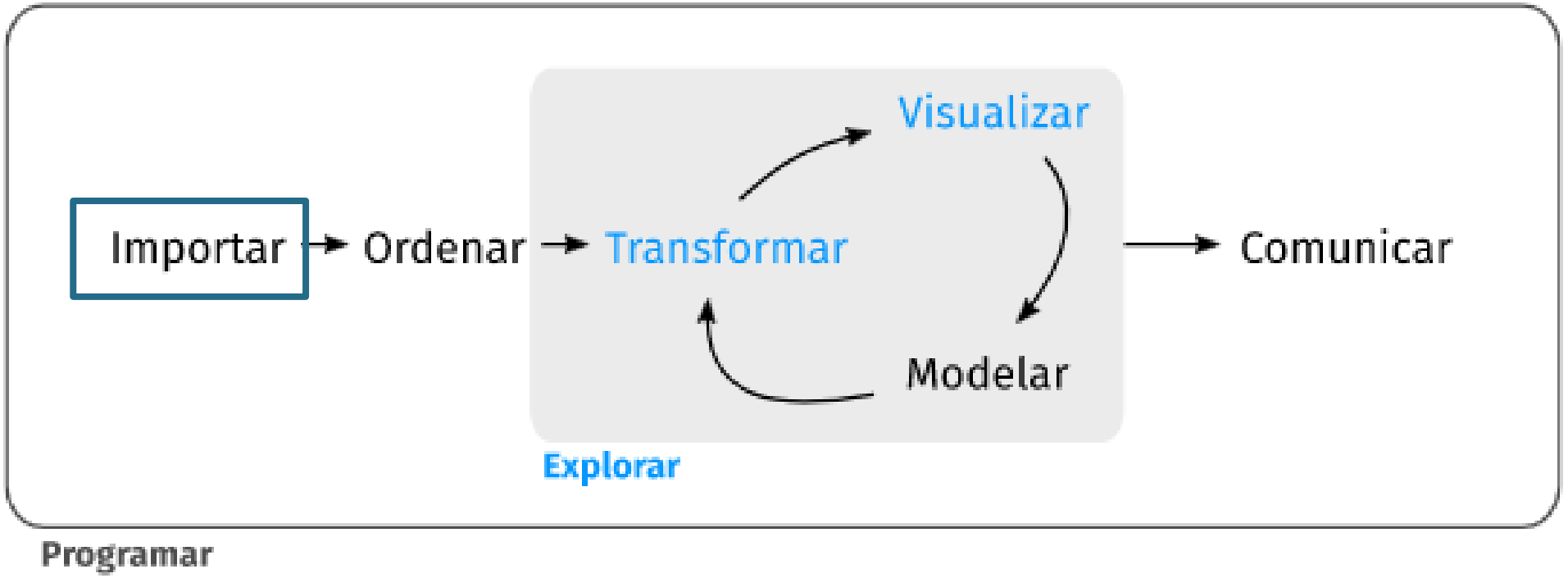
- ¿Tidyverse? -> incluir explícitamente en la búsqueda (ej: ggplot2) o seleccionar las respuestas que encuentro
- Tener en cuenta la fecha de la respuesta
- Buscar en inglés va a dar más resultados de búsqueda
- Busca ayudar en las comunidades, foros y twitter



Google

Importar datos

Importar datos



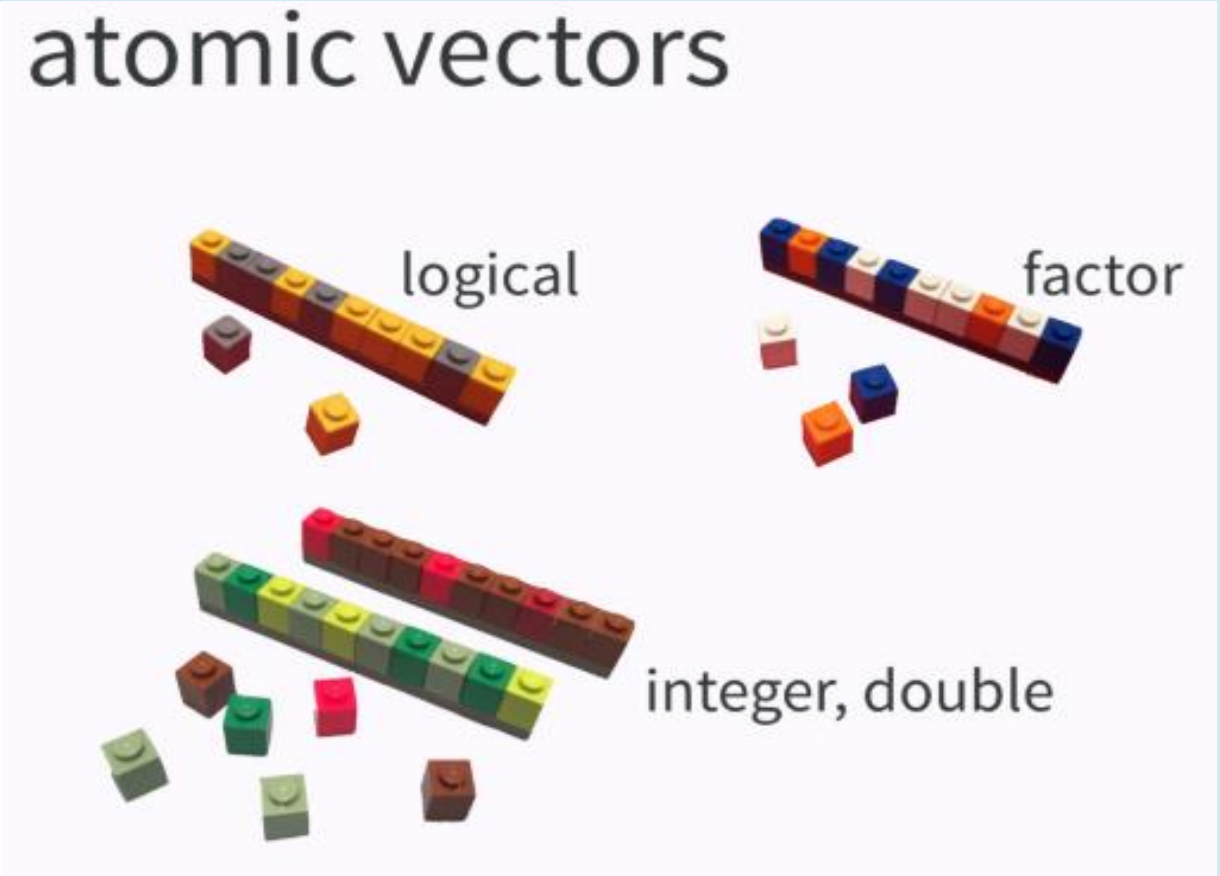
https://es.r4ds.hadley.nz/explora-introduccion.html

Vector

- Un vector es esencialmente una lista ordenada de cosas

Condición especial:

- Todo en el vector debe ser el mismo tipo de datos (double, integer, complex, logical o character)



https://speakerdeck.com/jennybc/data-rectangling

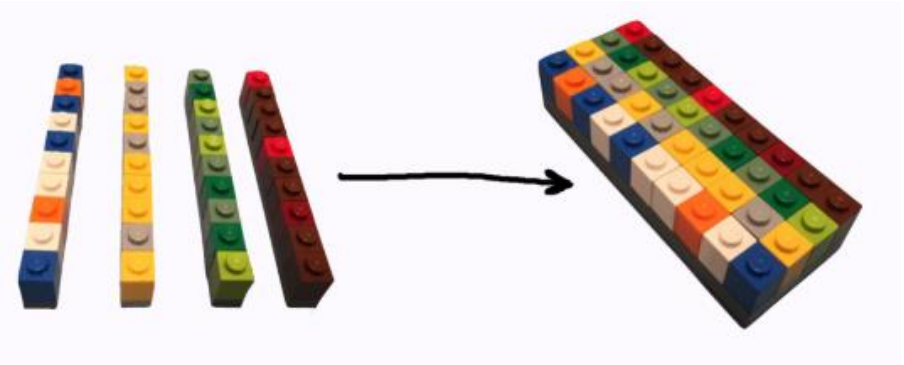
http://swcarpentry.github.io/r-novice-gapminder/04-data-structures-part1/index.html

Analogía de Jenny Bryan
https://twitter.com/jennybryan

data.frame

ESTRUCTURA DE DATOS

estructura que R sabe cómo construir a partir
de los tipos de datos básicos.



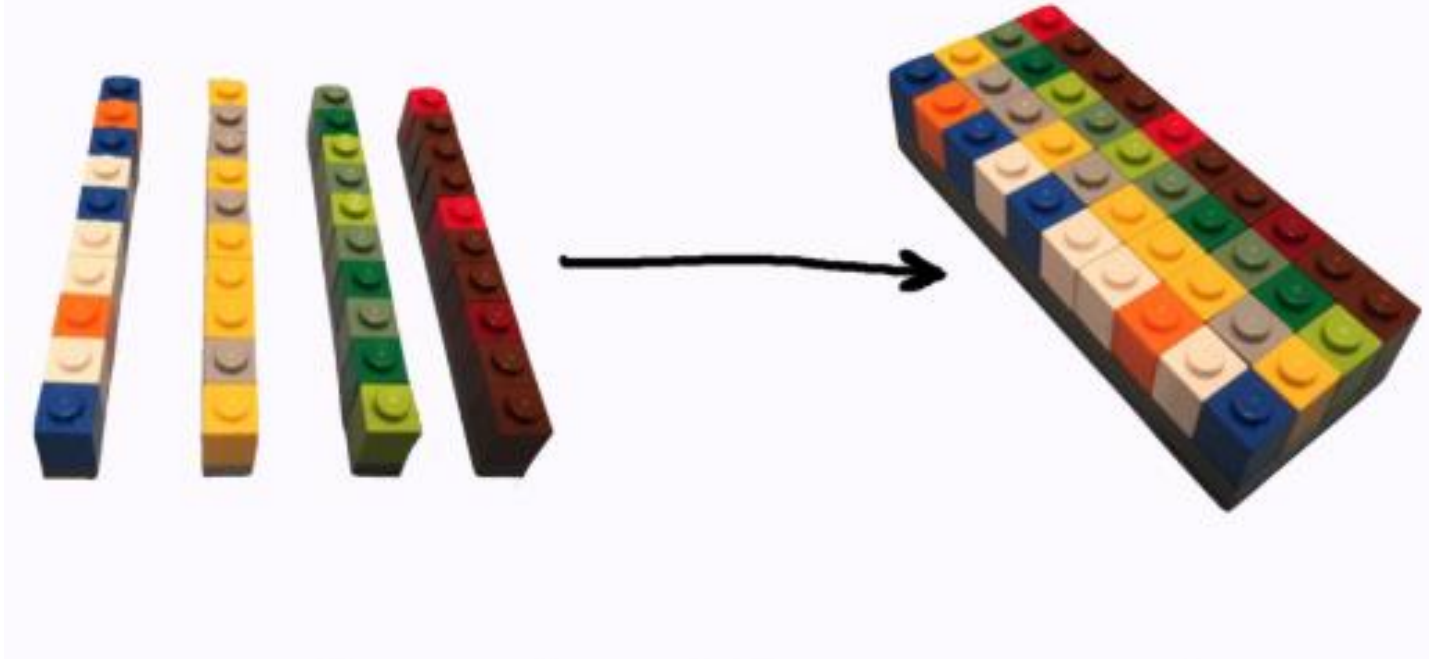
http://swcarpentry.github.io/r-novice-gapminder/04-data-structures-part1/index.html

Analogía de Jenny Bryan
https://twitter.com/jennybryan

data.frame

las columnas de los data.frames son todos vectores

Por eso R cada columna tiene un mismo tipo de datos.



¿Vectores de un mismo largo? -> DATA FRAME

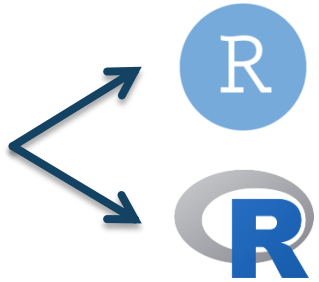
https://speakerdeck.com/jennybc/data-rectangling

http://swcarpentry.github.io/r-novice-gapminder/04-data-structures-part1/index.html

Analogía de Jenny Bryan
https://twitter.com/jennybryan

Trabajando con datos tabulares

- Leer datos desde archivos externos



- Generar datos tabulares con R -> data.frame()
- Los paquetes pueden tener bases de datos precargadas

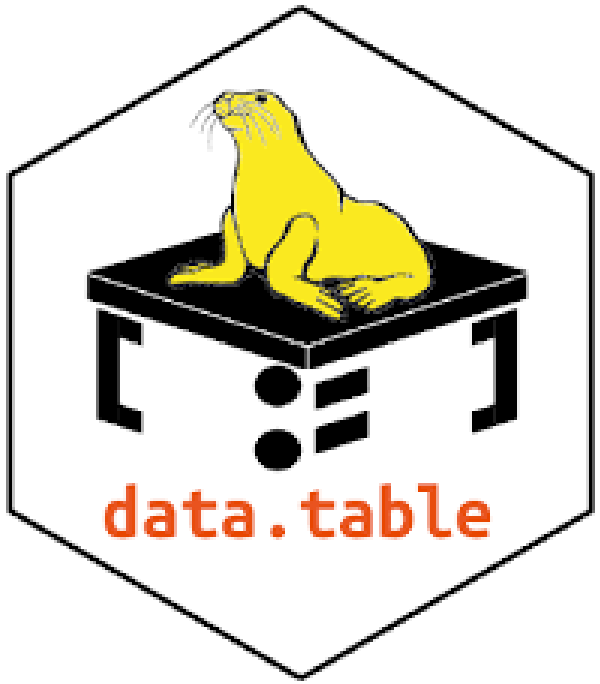
Tengo mis archivos en formato .xlsx

El paquete **readxl** facilita la obtención de datos desde Excel hacia R.

R Studio IDE tiene integrado este paquete

https://readxl.tidyverse.org/

Otras consideraciones



Si tengo que importar muchos datos puedo usar *data.table::fread()*. Es más rápido.

¡Manos a la obra!

- Escritura Paquete::funcion()
- Importar una base de datos
- Escribir con pipes

https://flor14.github.io/r_cai_2019 -> **df_pipes_tidyverse.R**

Mate break

Foto: gentiliza Mauro Lepore