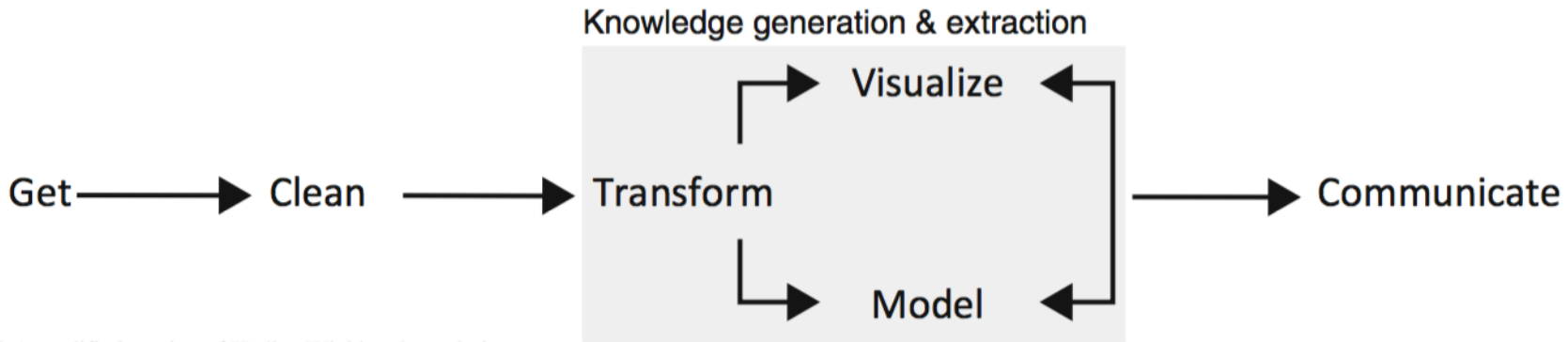
Manejo de datos en R (II)

Introducción a la Línea de Comandos para Análisis Bioinformáticos

03 de Marzo, 2020

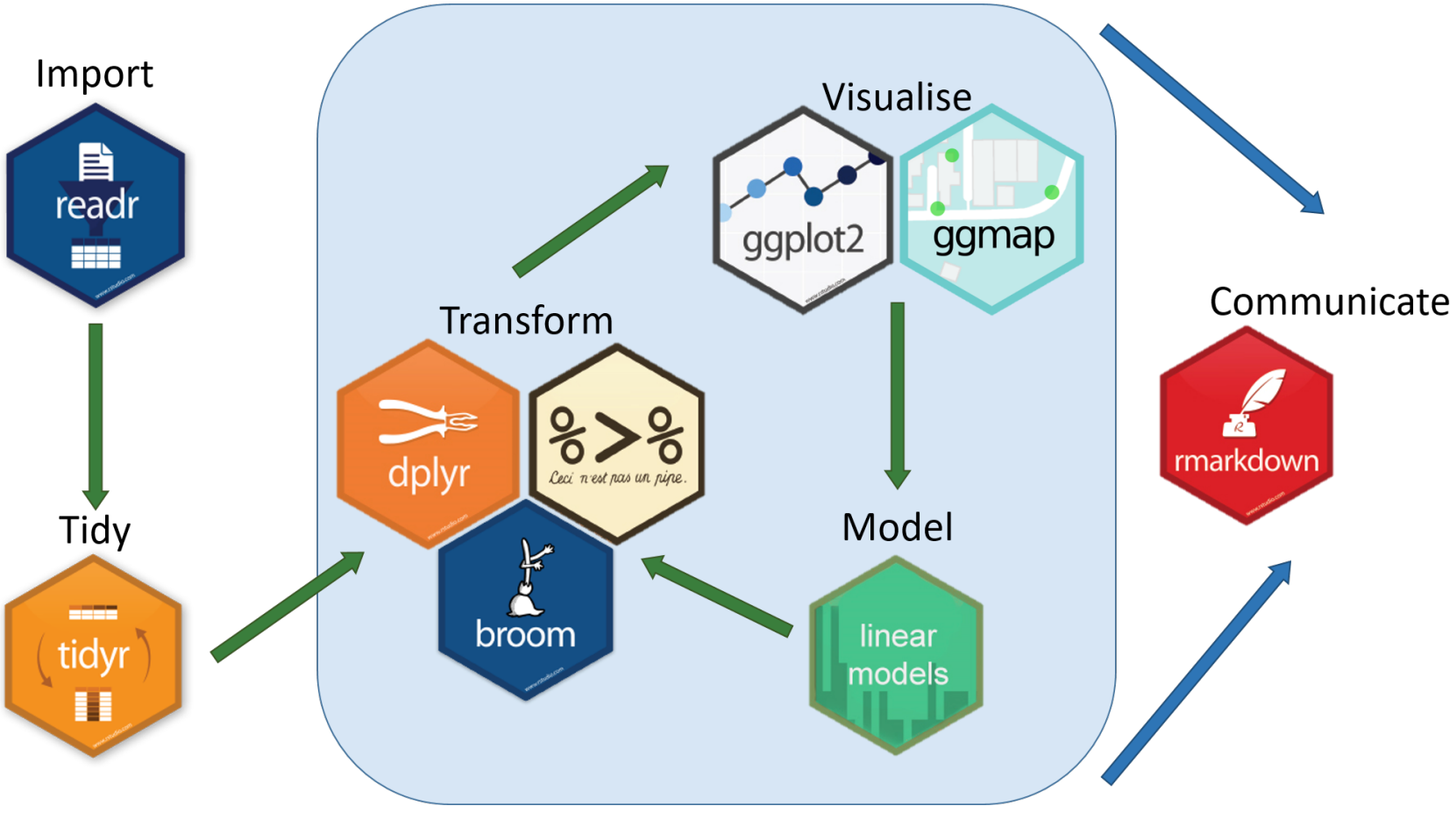
Análisis reproducible



Operaciones sobre datos

- Cargar datos *crudos*/Guardar datos finales y tablas de interés.
- Filtrar datos (con criterio).
- Unir datos que vienen de diferentes fuentes, referentes a un mismo conjunto estudiado.
- Hacer modificaciones: crear *tags*, correcciones ortográficas, filas y columnas de tablas, etc. . .
- Generar nuevos datos: obtener promedios, medianas, aplicar funciones de librerías.
- Dejar anotado y reportar lo hecho.

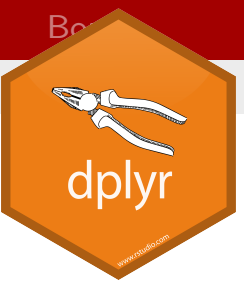

tidyverse


Import
readr
Tidy
tidyr
Transform
dplyr
%>%
Leci n'est pas un pipe.
broom
Visualise
ggplot2
ggmap
Model
linear models
Communicate
rmarkdown

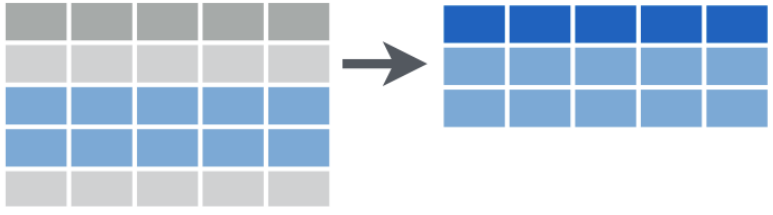

```r
library(tibble)

as_tibble(iris)
```

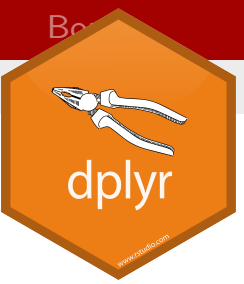
```
## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.1         3.5          1.4         0.2 setosa
##  2          4.9         3            1.4         0.2 setosa
##  3          4.7         3.2          1.3         0.2 setosa
##  4          4.6         3.1          1.5         0.2 setosa
##  5          5           3.6          1.4         0.2 setosa
##  6          5.4         3.9          1.7         0.4 setosa
##  7          4.6         3.4          1.4         0.3 setosa
##  8          5           3.4          1.5         0.2 setosa
##  9          4.4         2.9          1.4         0.2 setosa
## 10          4.9         3.1          1.5         0.1 setosa
## # ... with 140 more rows
```



Subset Observations (Rows)



Subset Variables (Columns)



Combine Data Sets

a

x1	x2
A	1
B	2
C	3

+

b

x1	x3
A	T
B	F
D	T

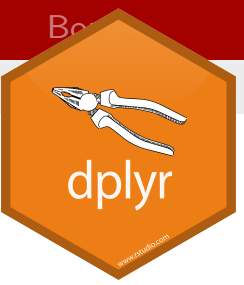
=

x1	x2	x3
A	1	T
B	2	F
C	3	NA

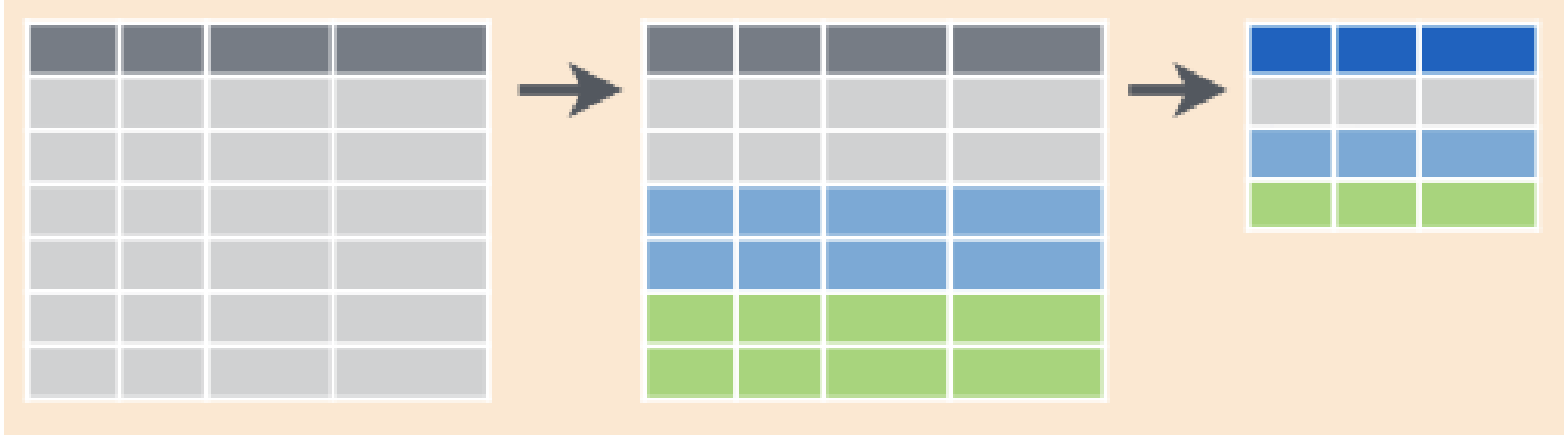
x1	x3	x2
A	T	1
B	F	2
D	T	NA

x1	x2	x3
A	1	T
B	2	F

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T



- Con dplyr es posible dividir el análisis de la tabla según una columna, y luego operar sobre en base a esto

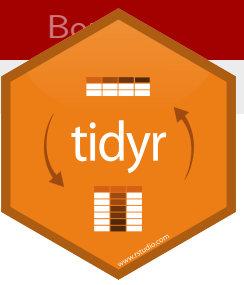


Tablas... todas dan igual?

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

Tablas... todas dan igual?

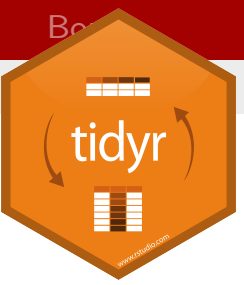
country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T



- Podemos llevar una tabla a formato alargado

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

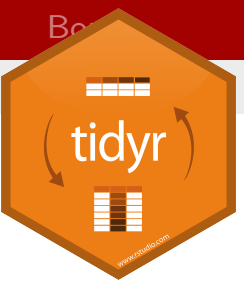
country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K



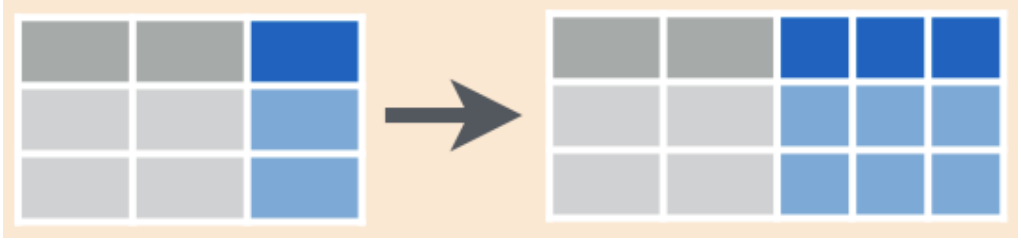
- O llevarla a un formato ancho

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

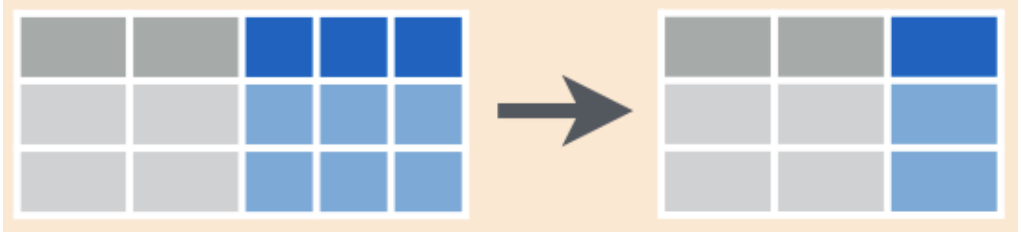
country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

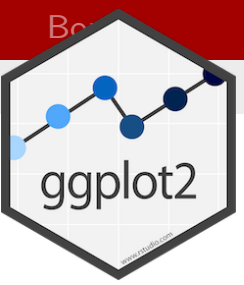


- Podemos separar valores en celdas

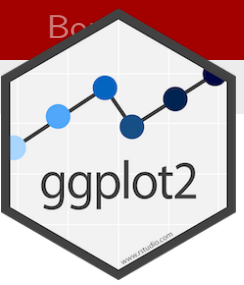


- O unirlos

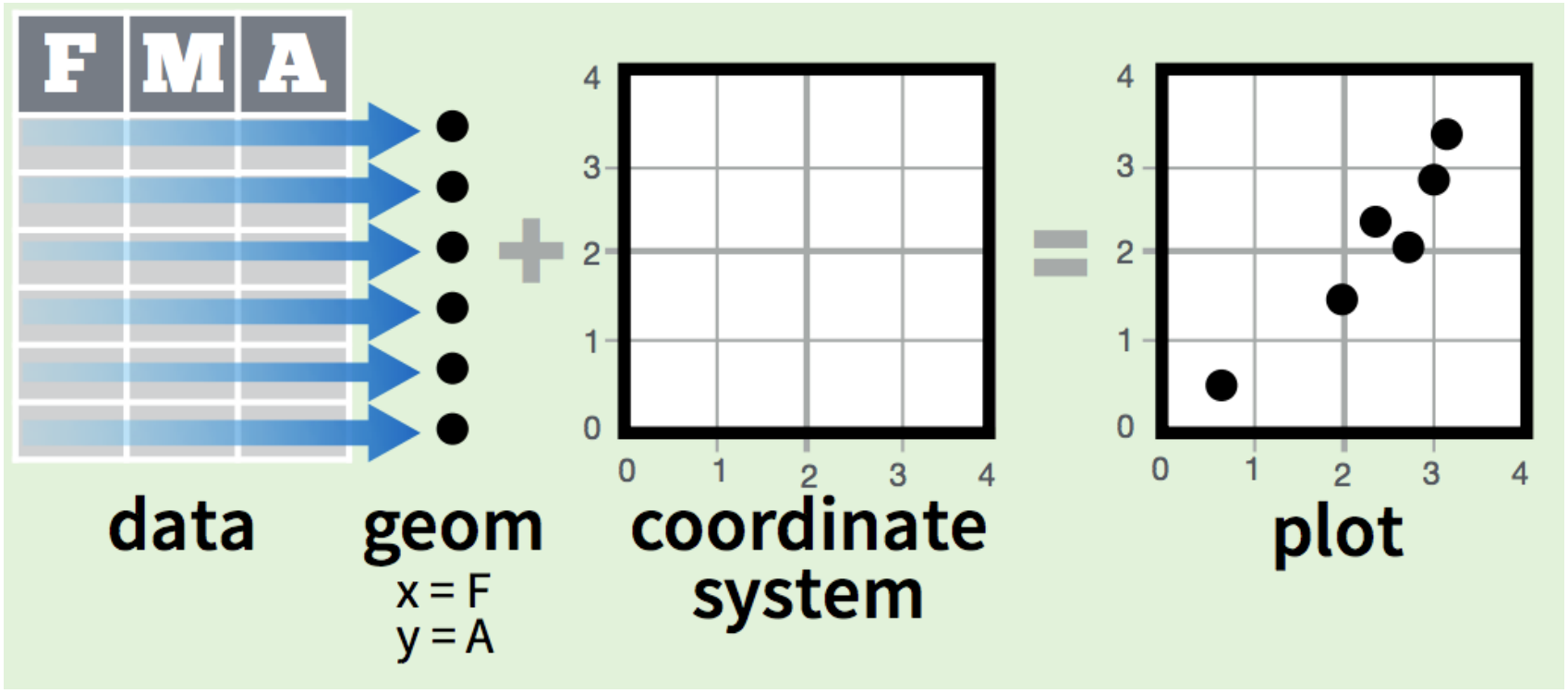


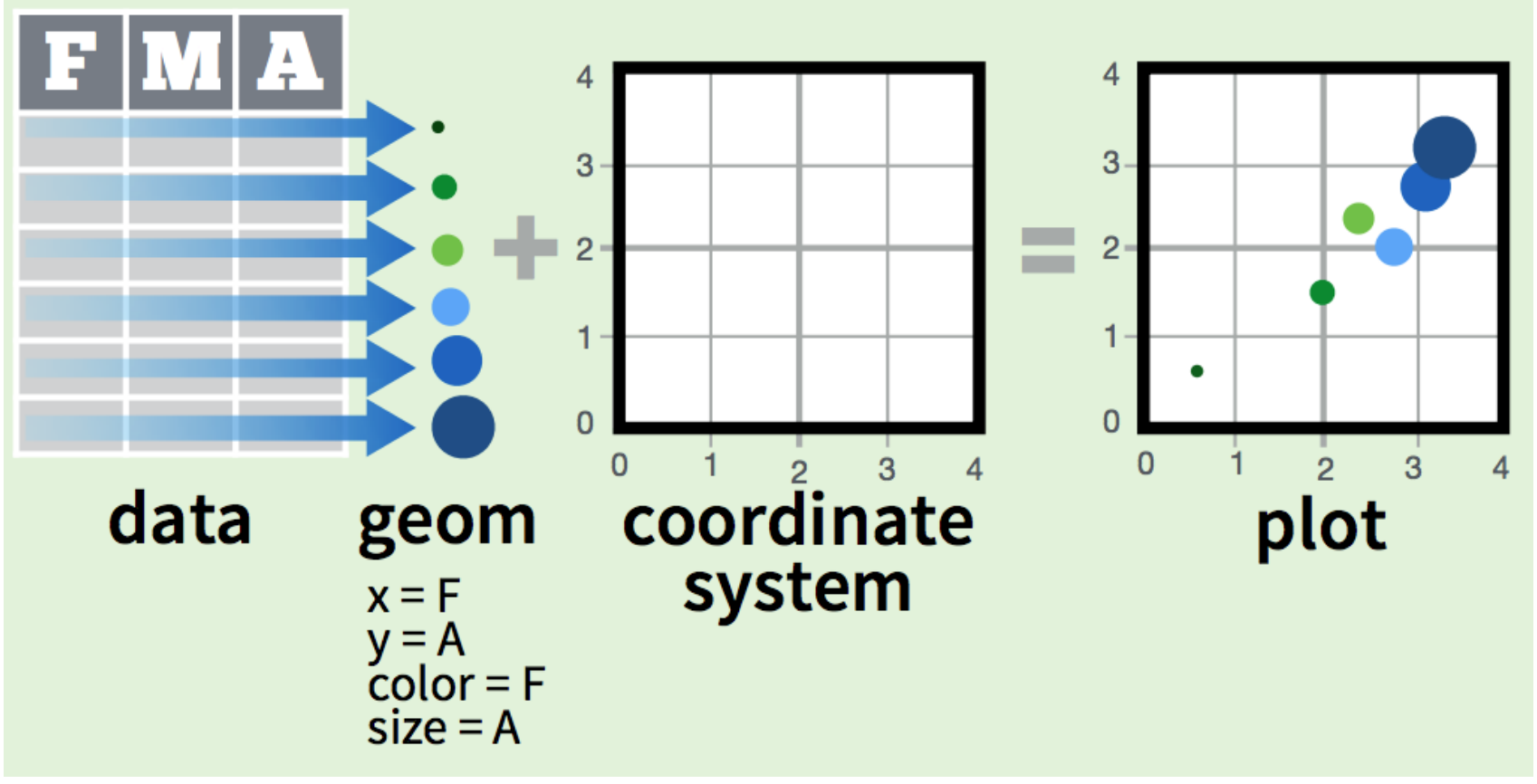


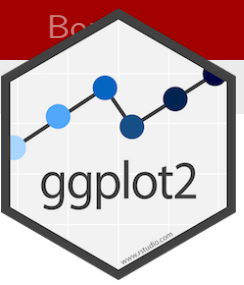
- Para realizar un gráfico preciso especificar:
  - Los **datos** sobre los que trabajo
  - Un sistema de coordenadas
  - Una especificación de qué representa cada dato a nivel **estético**
  - Una **forma geométrica** para representar estos datos
- Además, podríamos considerar
  - Especificar **funciones** que operen sobre nuestros datos, agrupándolos o transformándolos (pasa en histogramas, por ejemplo)
  - **Subdivisiones** de nuestros datos en base a factores.



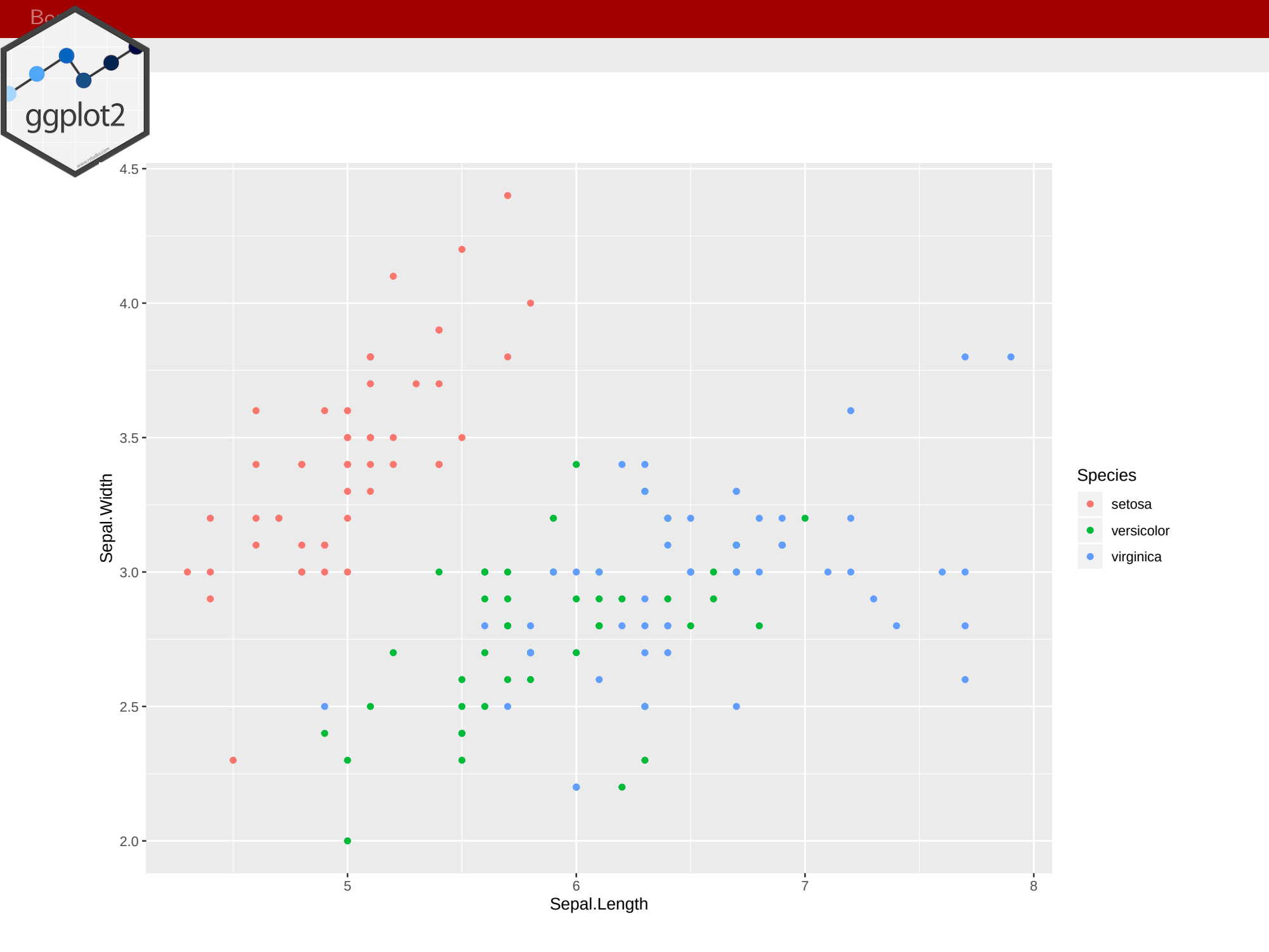
- Para realizar un gráfico preciso especificar:
  - Los **datos** sobre los que trabajo -> **ggplot()**
  - Un sistema de coordenadas -> **ggplot()**
  - Una especificación de qué representa cada dato a nivel **estético** -> **aes()**
  - Una **forma geométrica** para representar estos datos -> **geom()**
- Además, podríamos considerar
  - Especificar **funciones** que operen sobre nuestros datos, agrupándolos o transformándolos (pasa en histogramas, por ejemplo) -> **stat()**
  - **Subdivisiones** de nuestros datos en base a factores. -> **facet_wrap**


F
M
A
4
3
2
1
0
0
1
2
3
4
data
geom
x = F
y = A
coordinate
system
plot


F
M
A
data
geom
x = F
y = A
color = F
size = A
coordinate system
plot
0
1
2
3
4




```
library(tidyverse)
  # se grafica Sepal.Length vs Sepal.Width,
  # coloreando segun Species
ggplot(data = iris,
           aes(x = Sepal.Length,
               y = Sepal.Width,
               color = Species,
               fill = Species)) +
  # se grafica utilizando puntos
  geom_point()
```


ggplot2
Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
Sepal.Length
Species
setosa
versicolor
virginica


Ceci n'est pas une pipe.

- Es el *pipe* de R.
- El uso es exactamente igual al '|' de Bash.
- Un único detalle: se utiliza **.** para hacer referencia a resultados intermedios en un pipe.

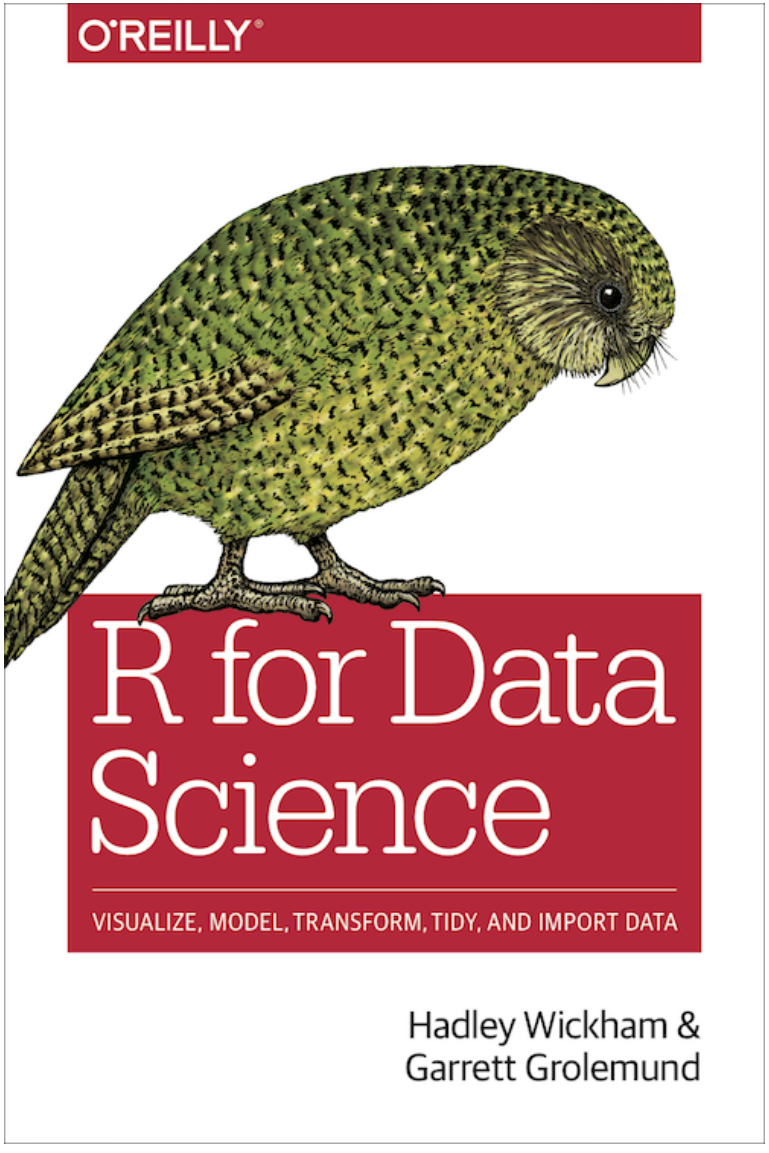

```r
# con magrittr
library(magrittr)

Sepal.Width.Median = iris %>% .$Sepal.Width %>% median(.)
```

¿Donde encuentro info sobre estos paquetes?

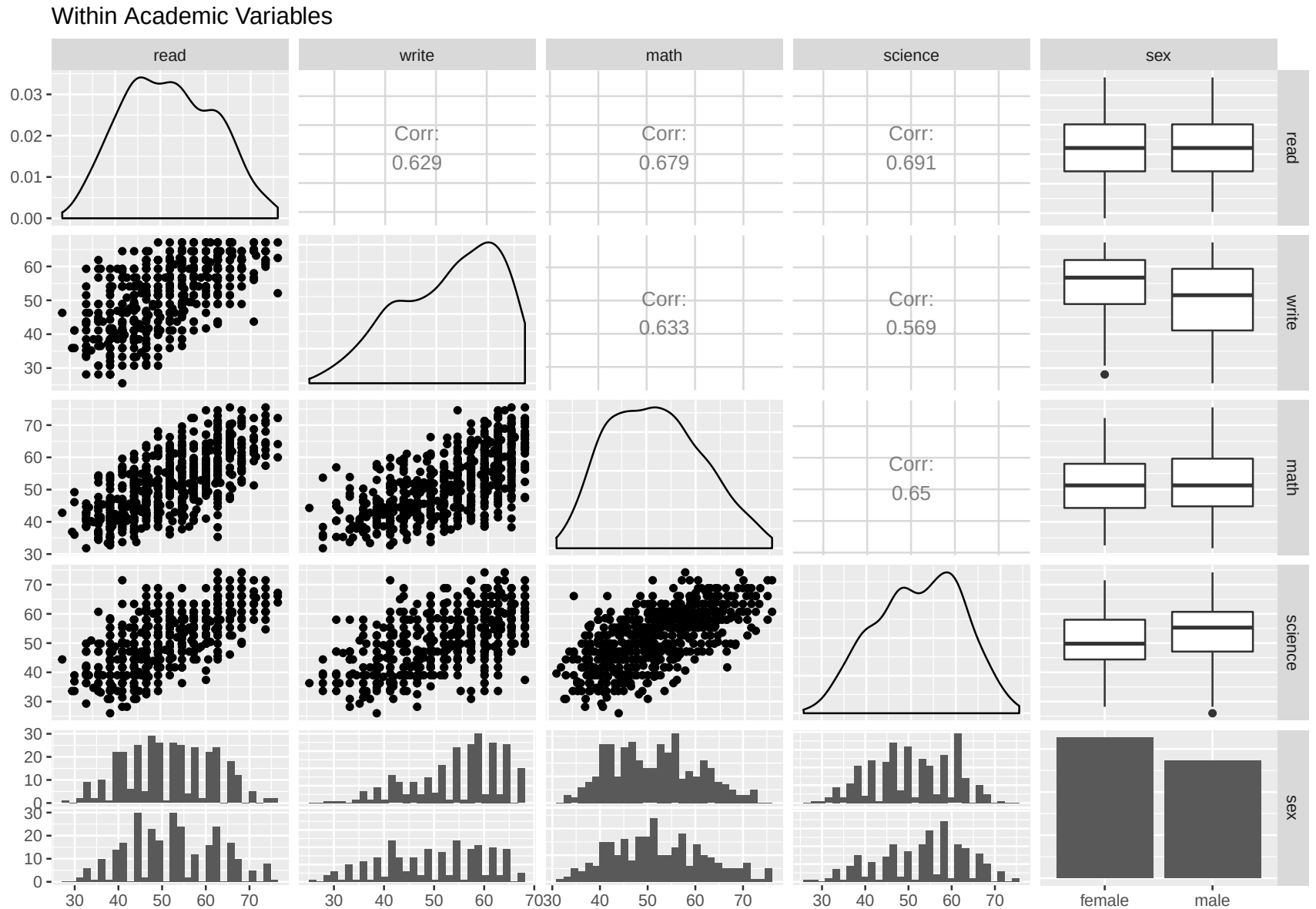
- Cheatsheets
- Vignettes

¿Donde encuentro info sobre estos paquetes?

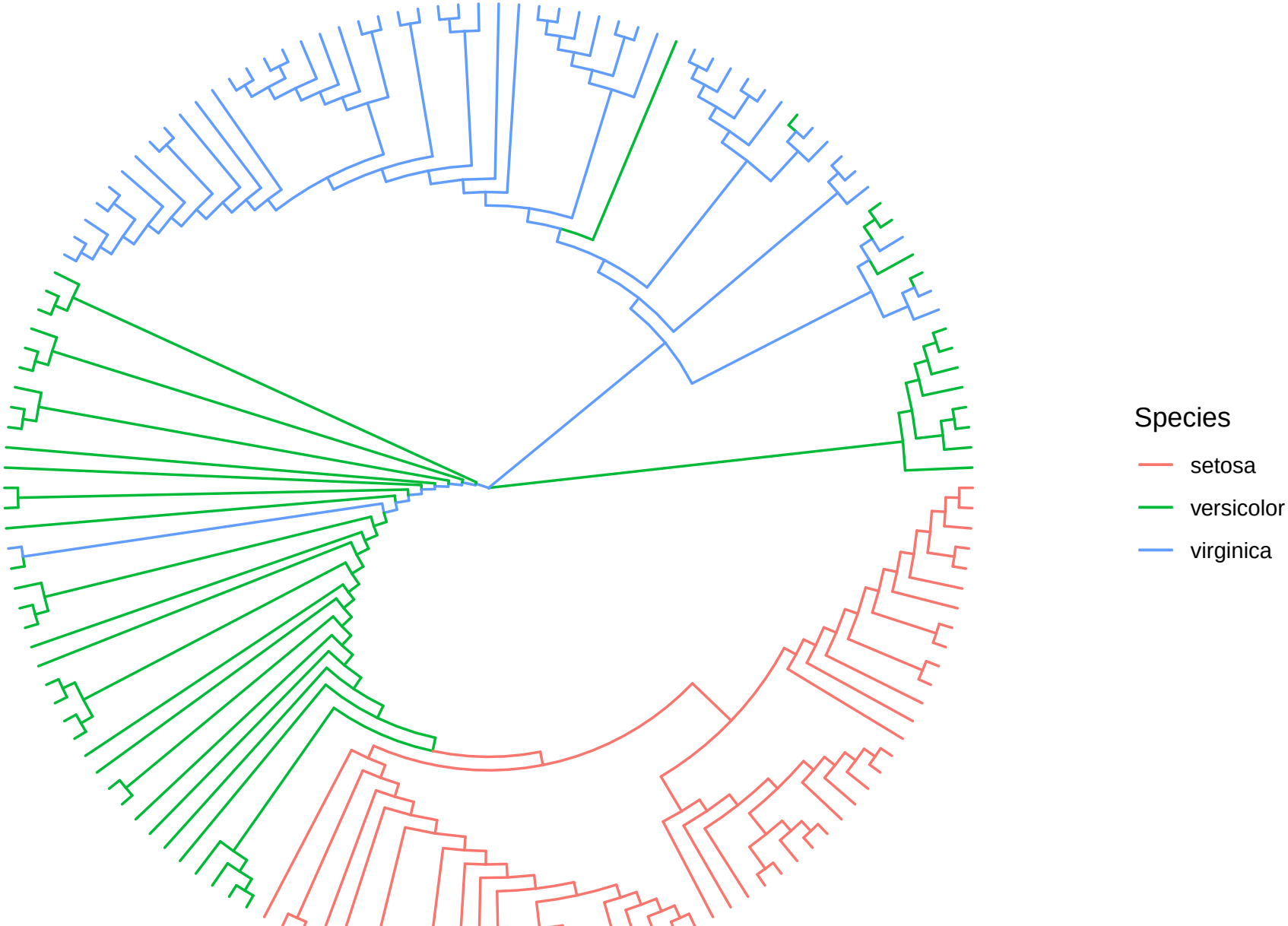


Bonus

GGally: análisis exploratorios y otros

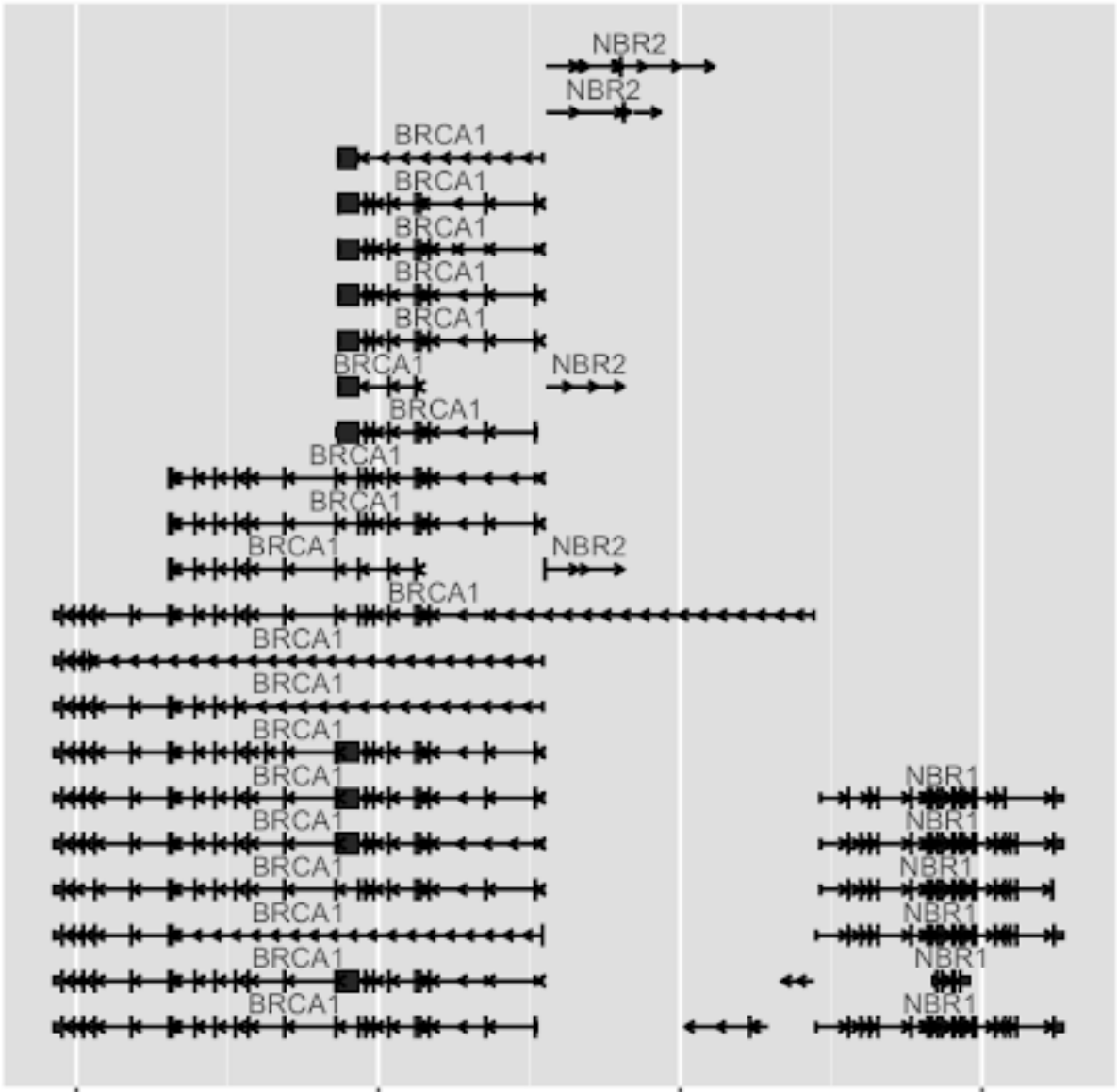


Filogenética: librería ggtree



Genómica: BioCircos/**rcirclize** y gggnomics, ggbio

Genómica: BioCircos/rcirclize y gggnomics, **ggbio**



Genómica: BioCircos/rcirclize y gggnomics, **ggbio**