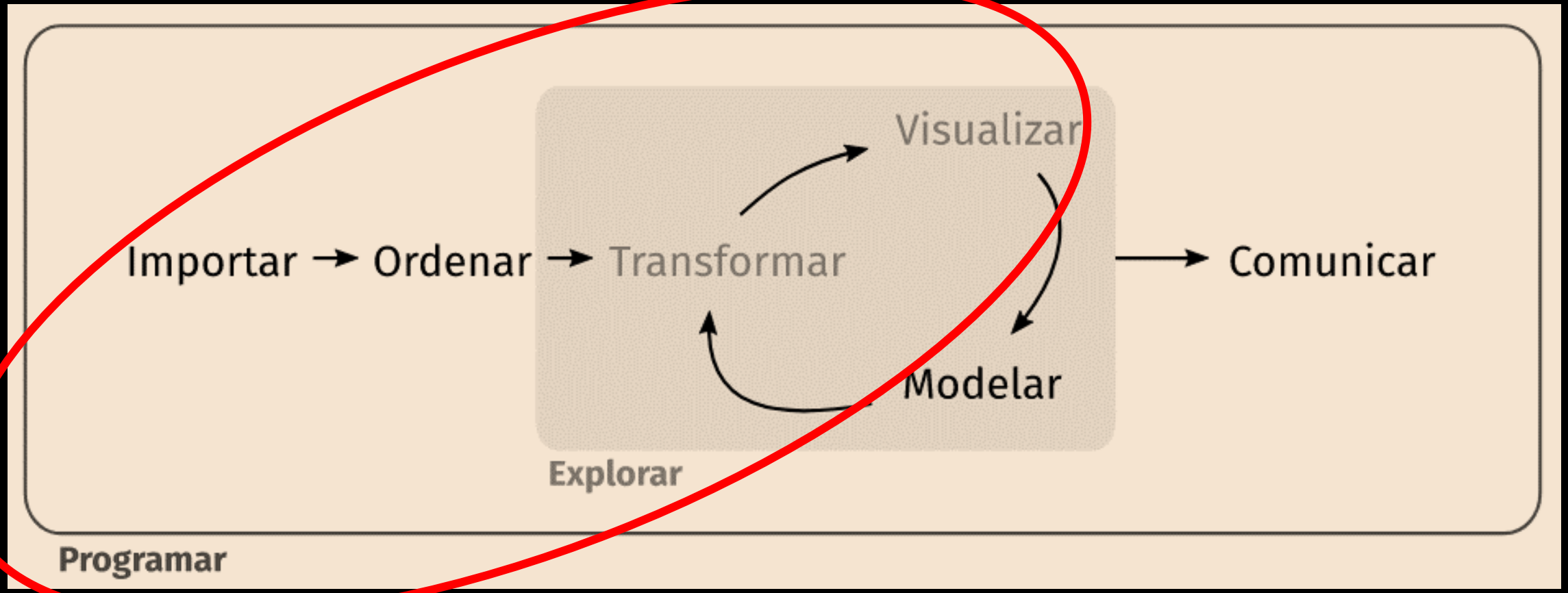
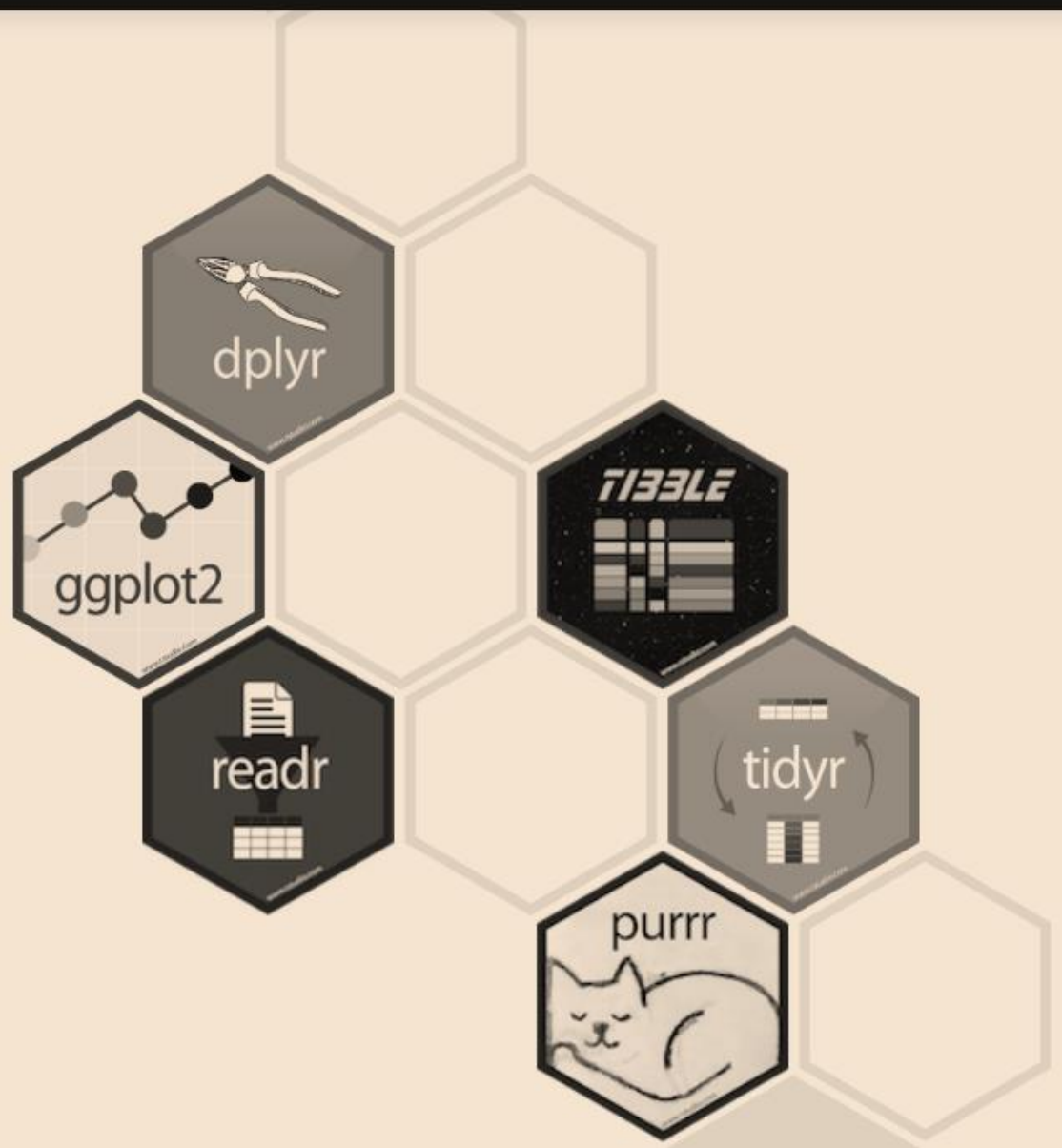


https://es.r4ds.hadley.nz/explora-introduccion.html

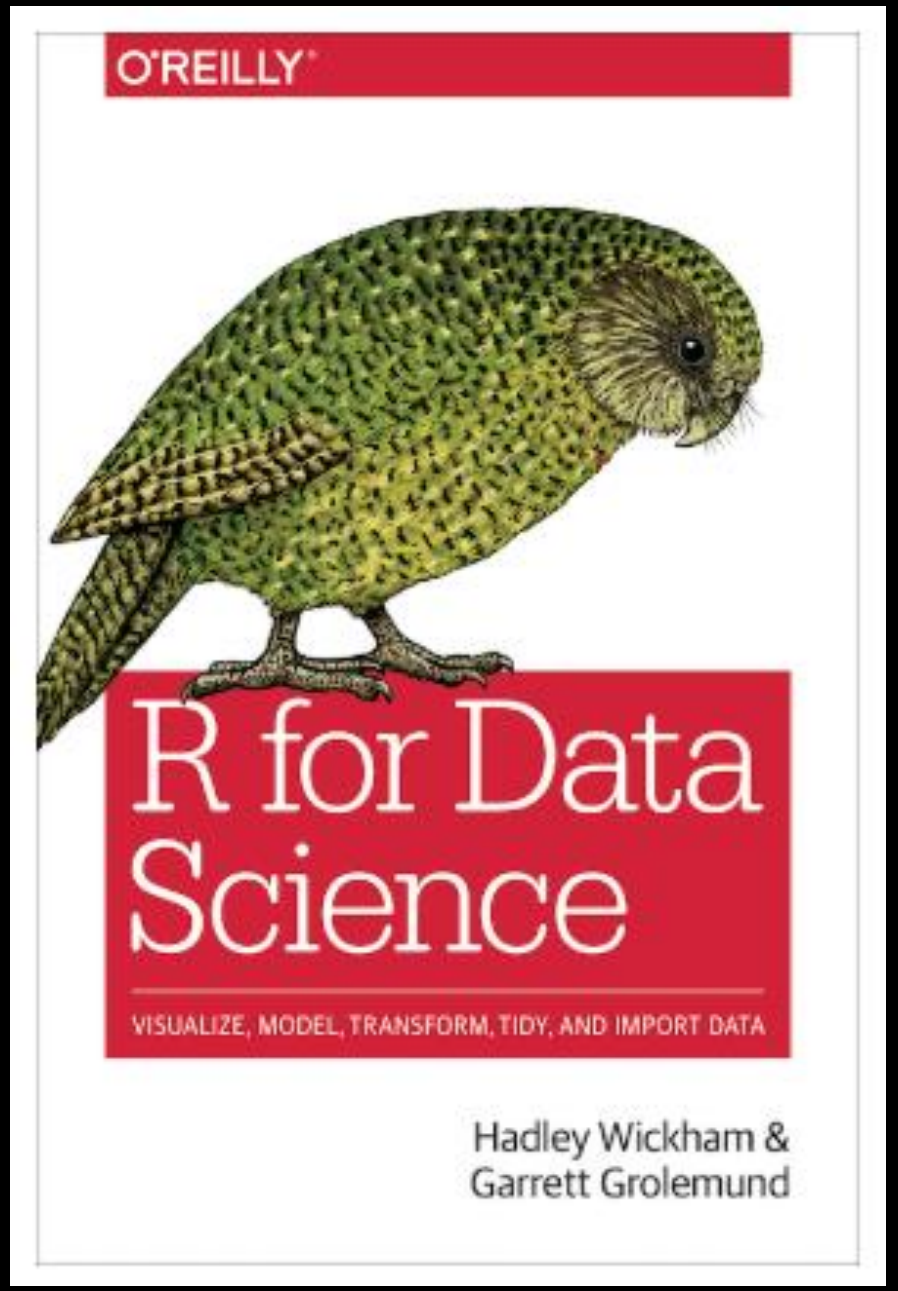
Tidyverse



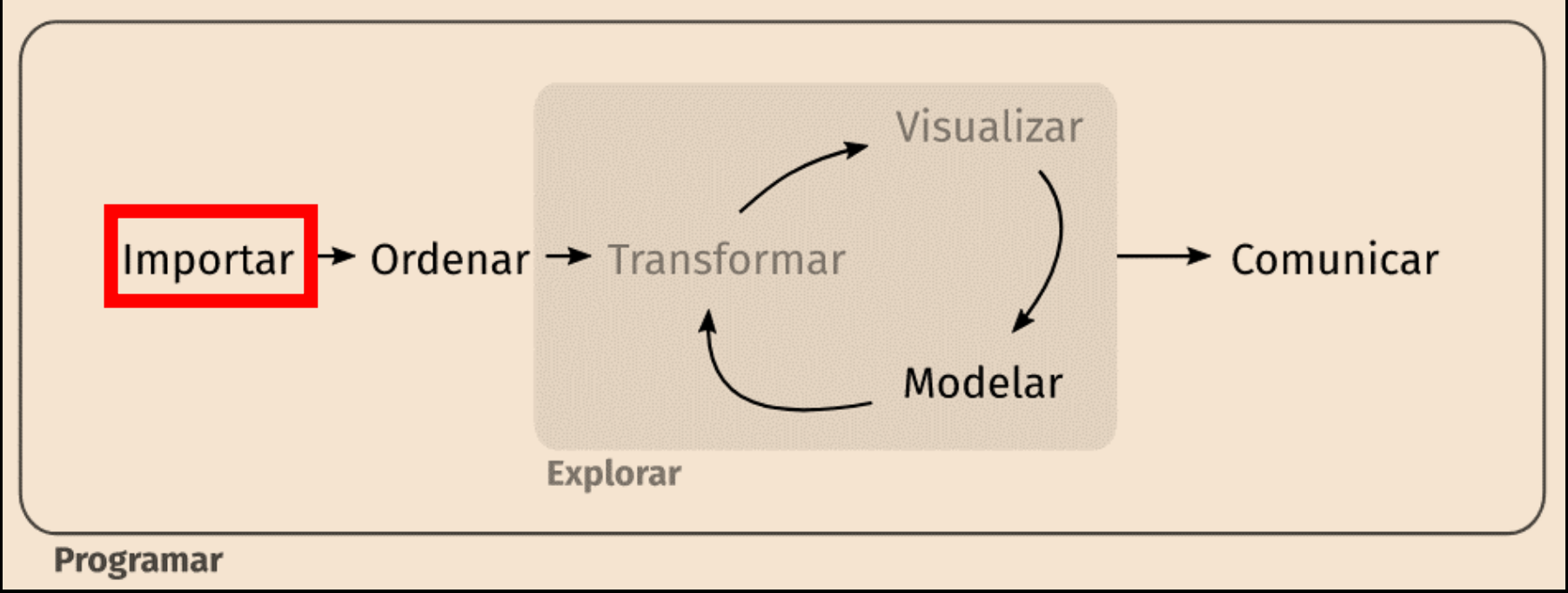
https://www.tidyverse.org/

El ***tidyverse*** es una colección de paquetes R diseñados para la ciencia de datos. Todos los paquetes comparten un diseño, filosofía, gramática y estructuras de datos subyacente.

```
install.packages("tidyverse")
```



https://r4ds.had.co.nz/

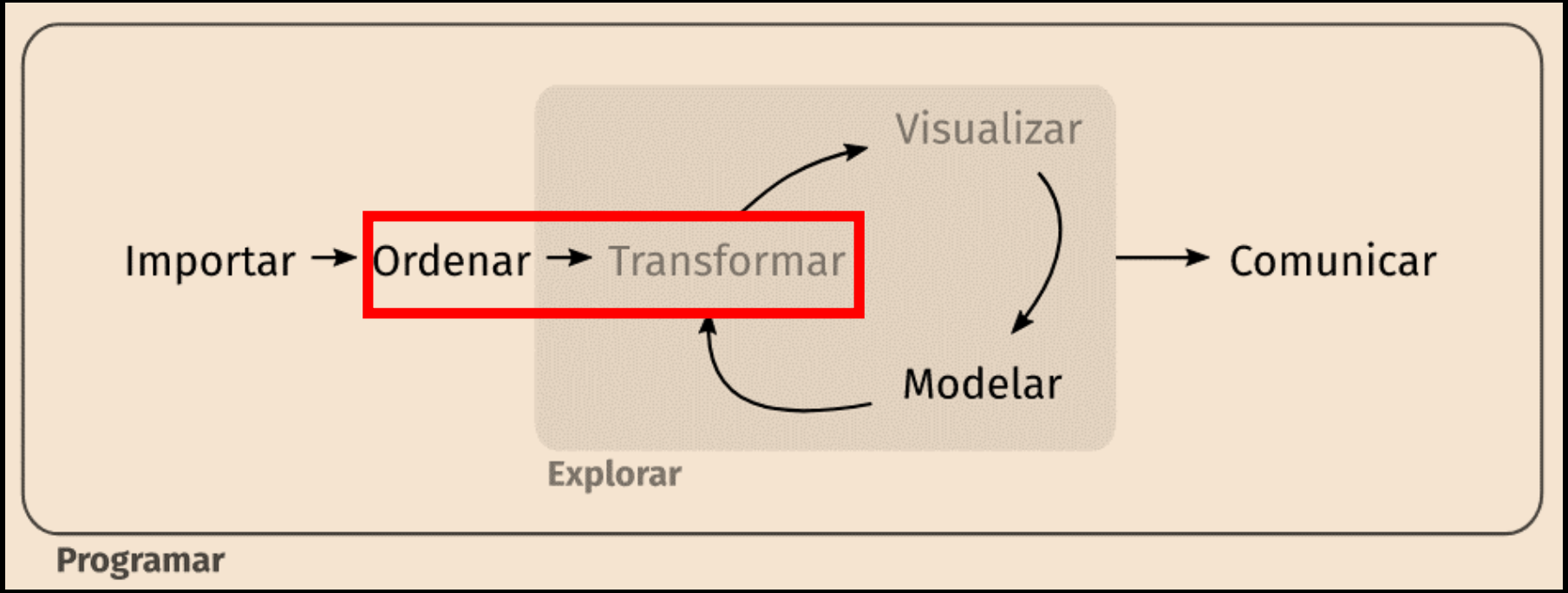

Importar
Ordenar
Transformar
Visualizar
Modelar
Comunicar
Explorar
Programar


```
read_tsv()
read_csv()
read_csv2()
```


```
read_excel()
```

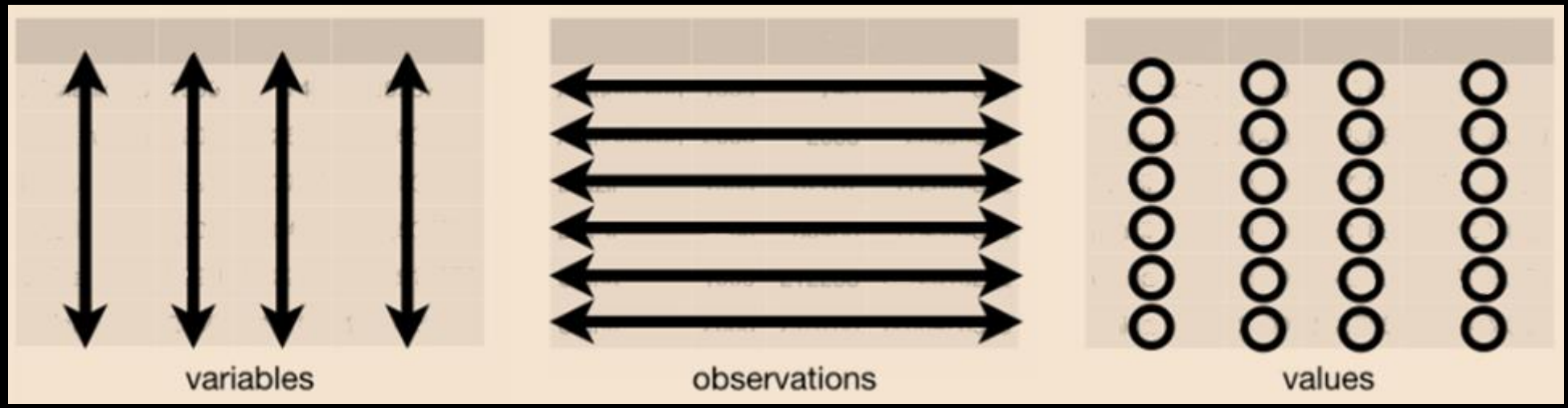

```
read_stata()
read_spss()
read_sas()
```

“Manipulación de datos” (Data Wrangling)



“Datos ordenados” (*Tidy Data*)

- Cada columna es una variable
- Cada fila es una observación
- Cada valor tiene su propia celda



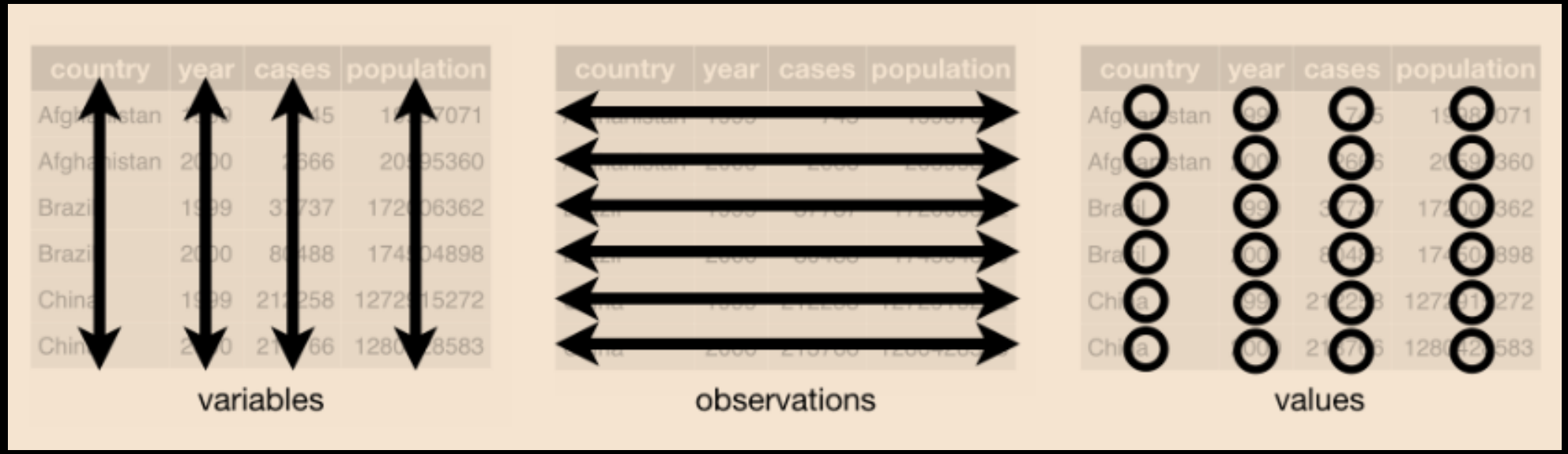
¿Cuál corresponde a datos “tidy”?

	country	year	type	count
	<chr>	<int>	<chr>	<int>
1	Afghanistan	1999	cases	745
2	Afghanistan	1999	population	19987071
3	Afghanistan	2000	cases	2666
4	Afghanistan	2000	population	20595360
5	Brazil	1999	cases	37737
6	Brazil	1999	population	172006362
7	Brazil	2000	cases	80488
8	Brazil	2000	population	174504898
9	China	1999	cases	212258
10	China	1999	population	1272915272
11	China	2000	cases	213766
12	China	2000	population	1280428583

	country	year	cases	population
	<chr>	<int>	<int>	<int>
1	Afghanistan	1999	745	19987071
2	Afghanistan	2000	2666	20595360
3	Brazil	1999	37737	172006362
4	Brazil	2000	80488	174504898
5	China	1999	212258	1272915272
6	China	2000	213766	1280428583

	country	year	rate
*	<chr>	<int>	<chr>
1	Afghanistan	1999	745/19987071
2	Afghanistan	2000	2666/20595360
3	Brazil	1999	37737/172006362
4	Brazil	2000	80488/174504898
5	China	1999	212258/1272915272
6	China	2000	213766/1280428583

	country	`1999`	`2000`
*	<chr>	<int>	<int>
1	Afghanistan	19987071	20595360
2	Brazil	172006362	174504898
3	China	1272915272	1280428583



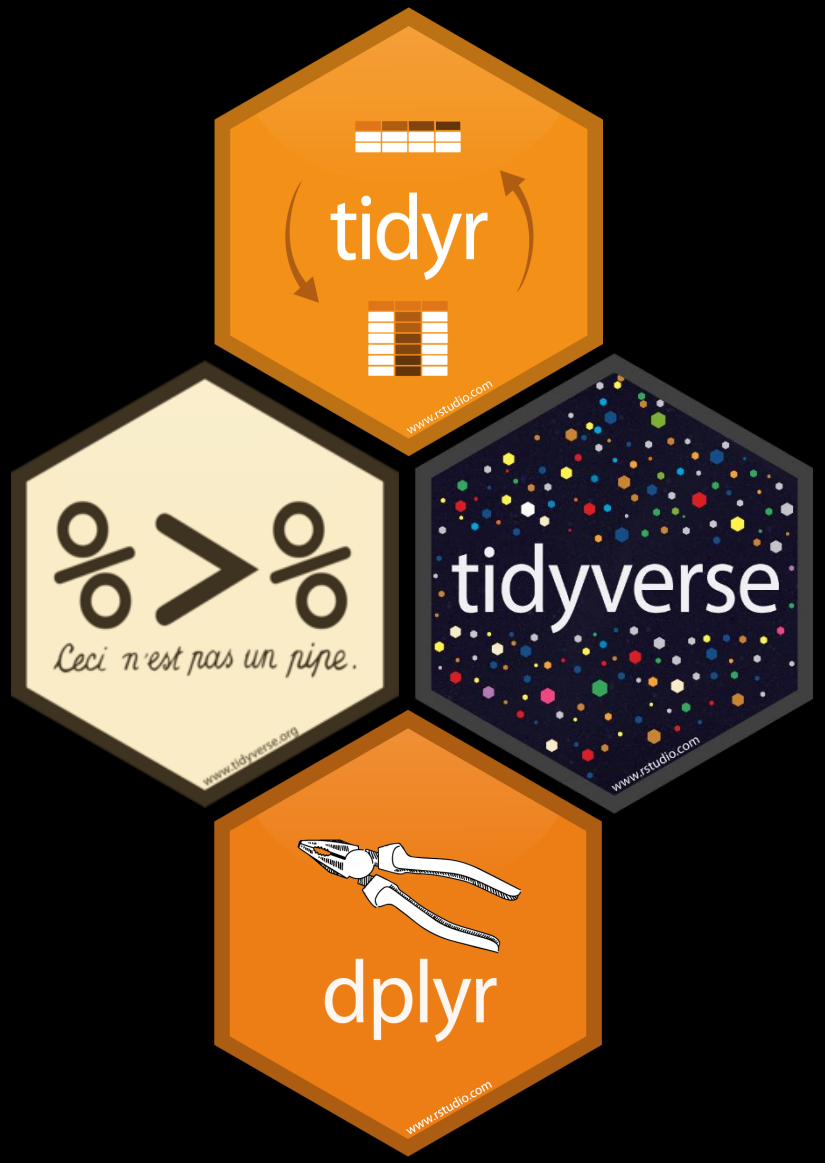
	country	year	cases	population
	<chr>	<int>	<int>	<int>
1	Afghanistan	1999	745	19987071
2	Afghanistan	2000	2666	20595360
3	Brazil	1999	37737	172006362
4	Brazil	2000	80488	174504898
5	China	1999	212258	1272915272
6	China	2000	213766	1280428583

¿Por qué ordenar los datos así?

- Bueno tener una sola forma consistente de almacenamiento de datos
- Ventajas para explotar la forma de trabajo en R (vectores)

Manipulación de datos con “tidyverse”

- **tidyr**
  - gather()
  - spread()
  - separate()
  - unite()
- **dplyr**
  - filter()
  - arrange()
  - select()
  - mutate()
  - summarise()
    - group_by()
- **magrittr**
  - %>% (pipe)



"Happy families are all alike; every unhappy family is unhappy in its own way."

— **Leo Tolstoy**

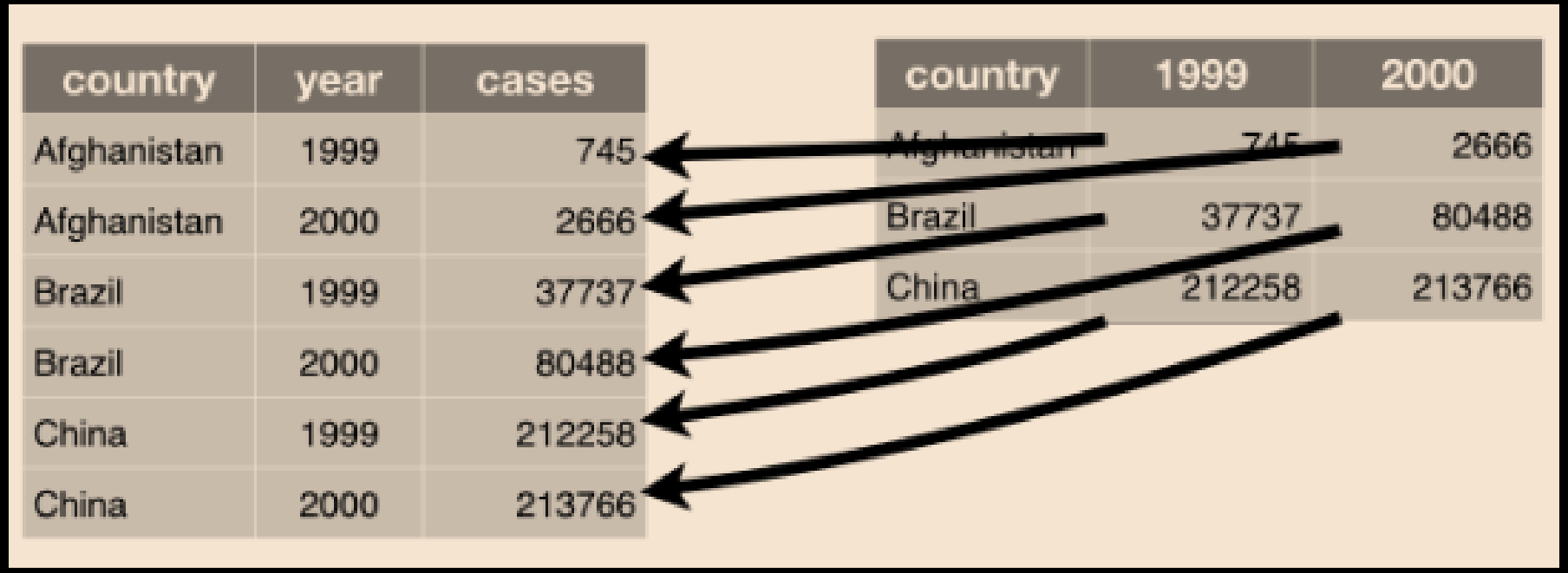
"Tidy datasets are all alike, but every messy dataset is messy in its own way."

— **Hadley Wickham**

- **tidyr**
  - gather()
  - spread()
  - separate()
  - unite()

pivot_longer()

- Usar cuando nombres de variables corresponden a valores

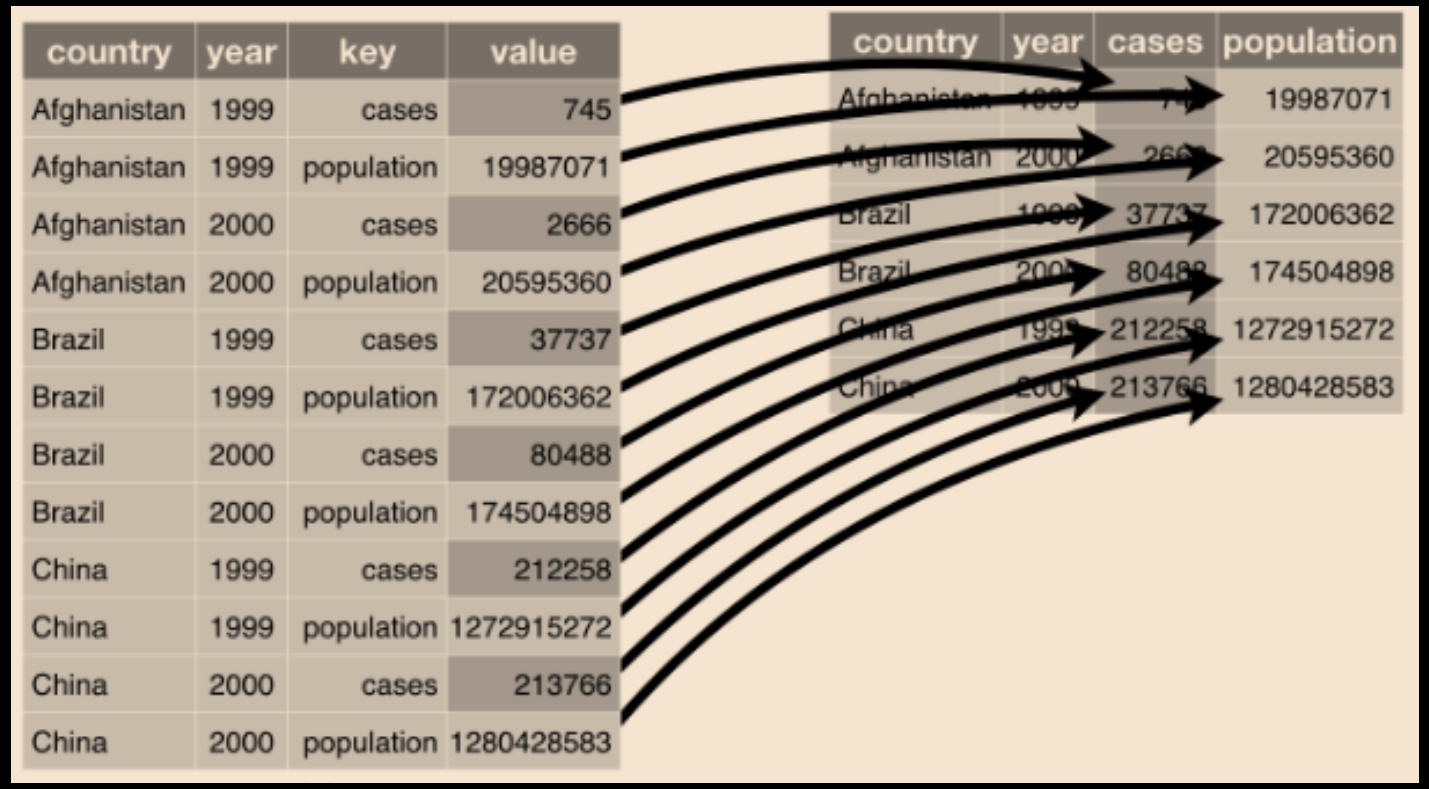


country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

pivot_wider()

- Función opuesta a `pivot_longer()`
  - Usar cuando información sobre una observación está “repartida” entre varias filas



country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

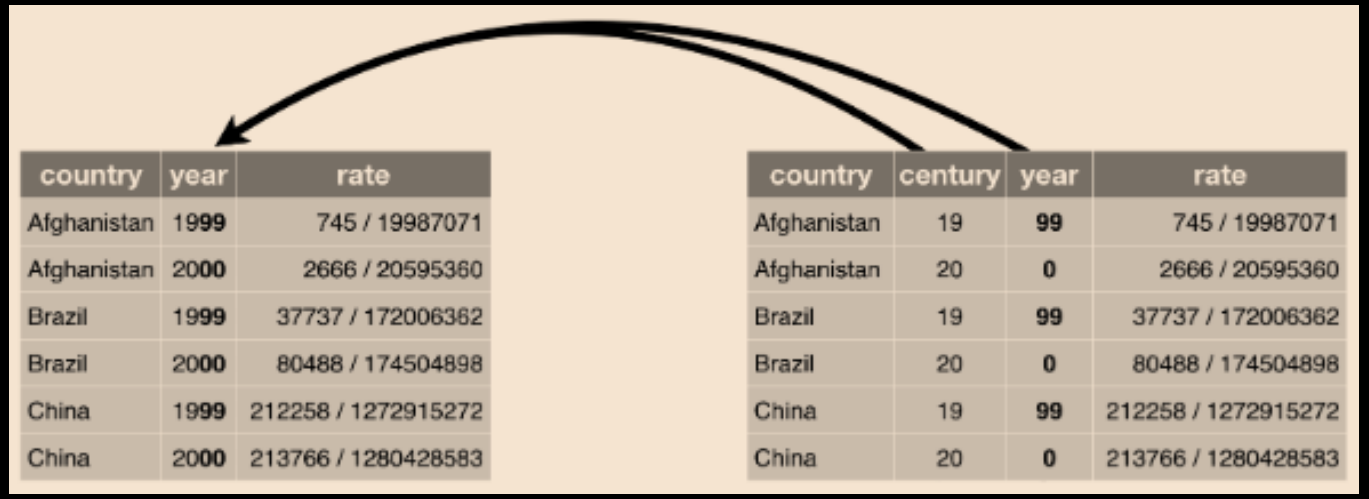
`separate()`/`unite()`

- `separate()` separa una columna en múltiples

country	year	rate
Afghanistan	1999	**745** / 19987071
Afghanistan	2000	**2666** / 20595360
Brazil	1999	**37737** / 172006362
Brazil	2000	**80488** / 174504898
China	1999	**212258** / 1272915272
China	2000	**213766** / 1280428583

country	year	cases	population
Afghanistan	1999	**745**	19987071
Afghanistan	2000	**2666**	20595360
Brazil	1999	**37737**	172006362
Brazil	2000	**80488**	174504898
China	1999	**212258**	1272915272
China	2000	**213766**	1280428583

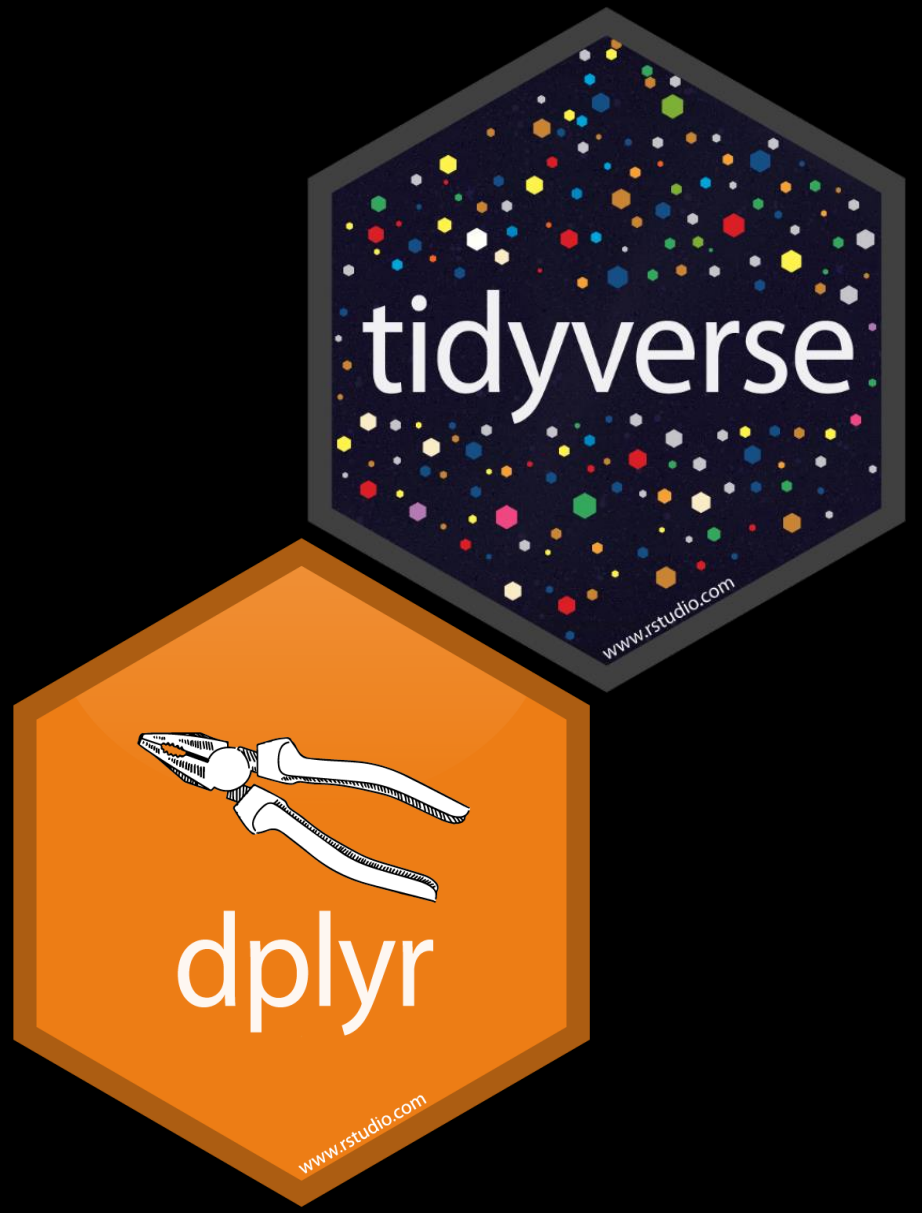
- `unite()` combina múltiples columnas en una



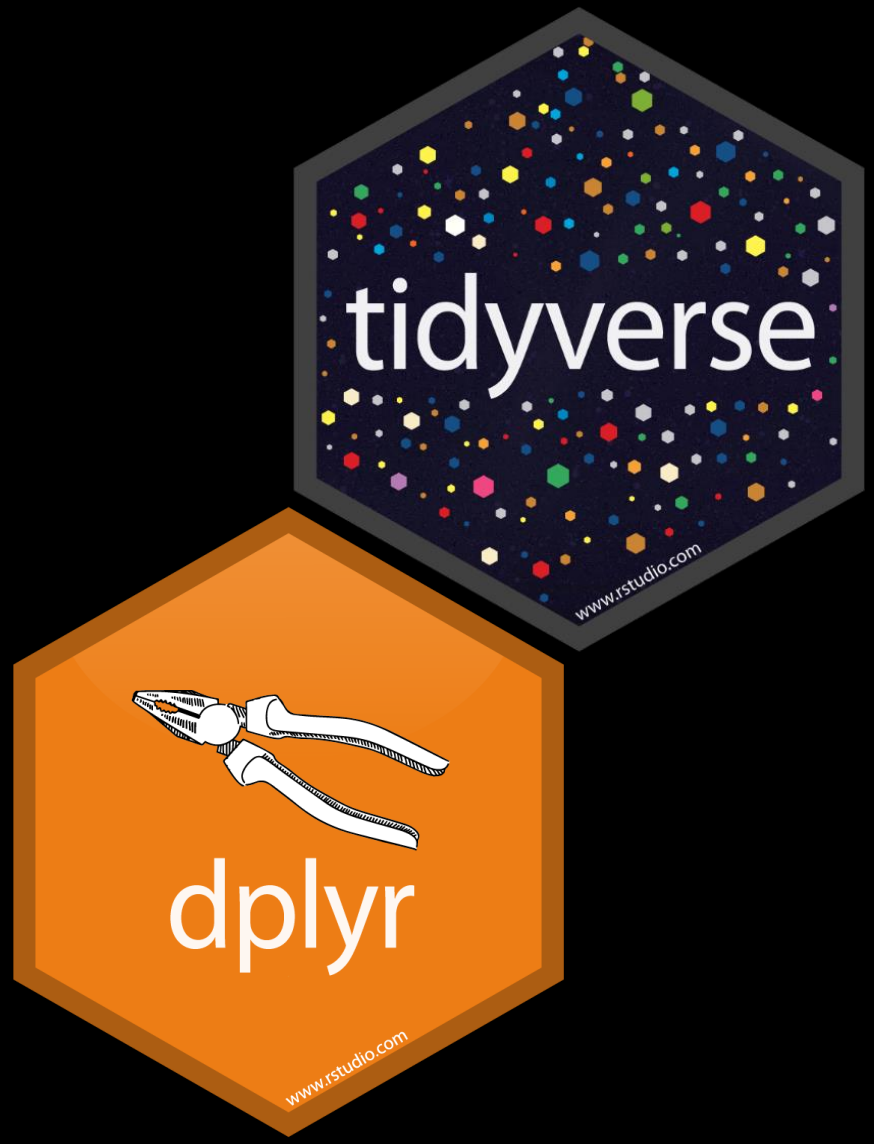
country	year	rate
Afghanistan	**1999**	745 / 19987071
Afghanistan	**2000**	2666 / 20595360
Brazil	**1999**	37737 / 172006362
Brazil	**2000**	80488 / 174504898
China	**1999**	212258 / 1272915272
China	**2000**	213766 / 1280428583

country	century	year	rate
Afghanistan	19	**99**	745 / 19987071
Afghanistan	20	**0**	2666 / 20595360
Brazil	19	**99**	37737 / 172006362
Brazil	20	**0**	80488 / 174504898
China	19	**99**	212258 / 1272915272
China	20	**0**	213766 / 1280428583

- **dplyr**
  - filter()
  - arrange()
  - select()
  - mutate()
  - summarise()

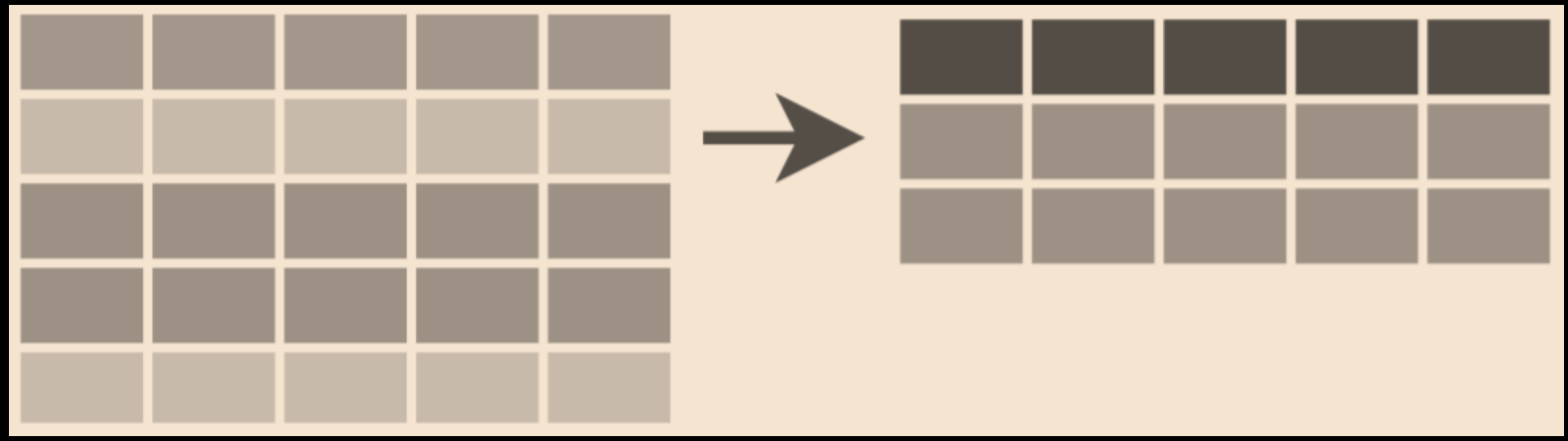


- **dplyr**
  - filter()
  - arrange()
  - select()
  - mutate()
  - group_by()
  - summarise()



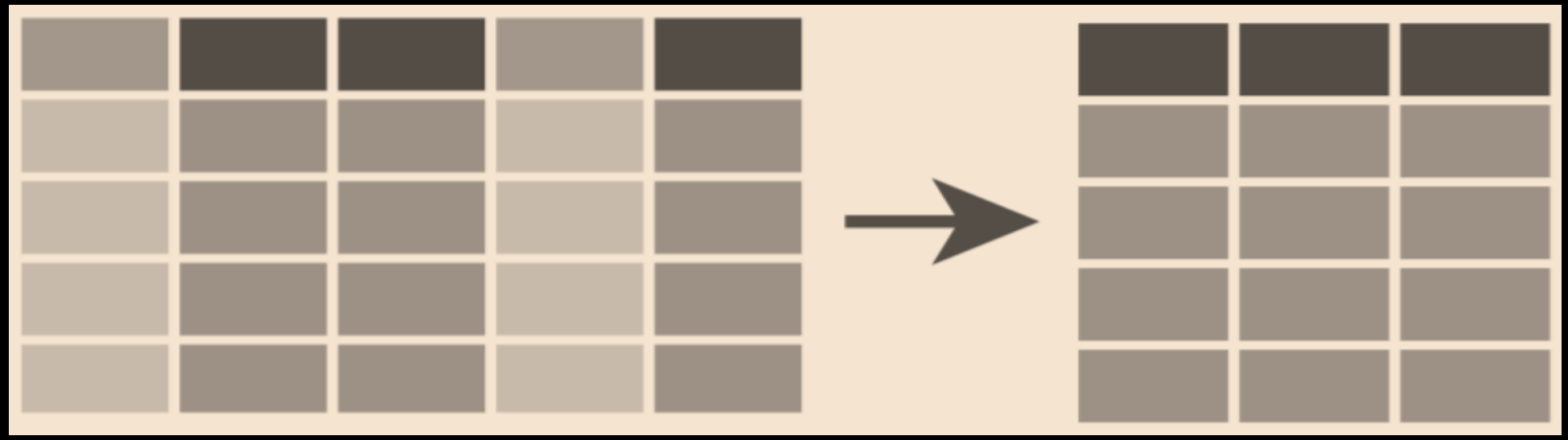
filter()

R Base	D[D$subject == 4 & D$trial == 10,]
dplyr	filter(D, subject == 4, trial == 10)



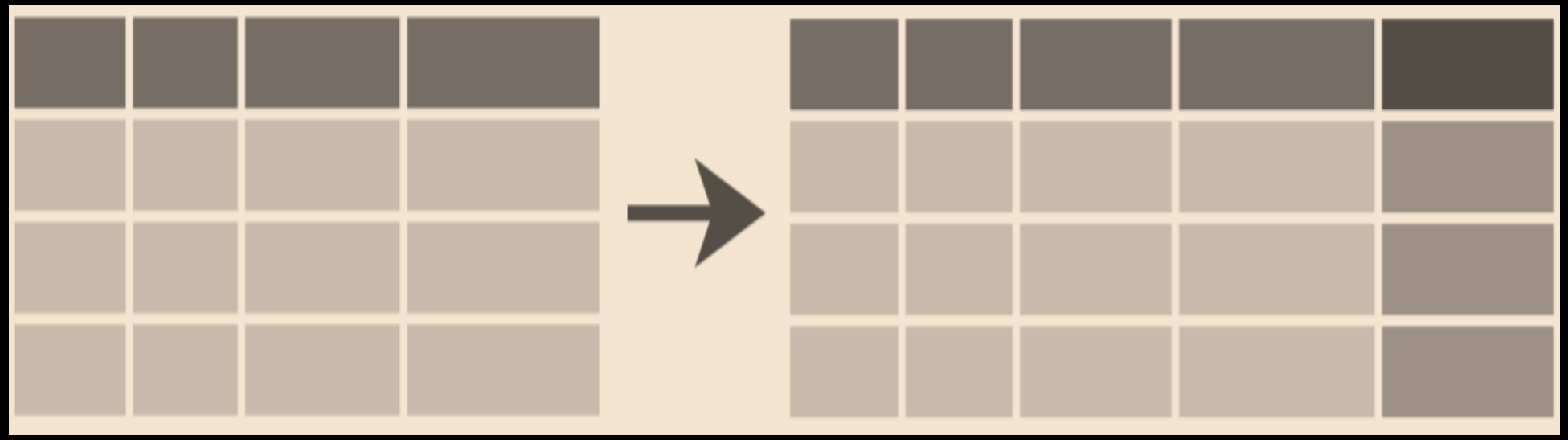
select()

R Base	`D[,c("subject", "trial")]`
dplyr	`select(D, subject, trial)`



mutate()

R Base	`D$freq <- D$count/D$time`
dplyr	`D <- mutate(D, freq = count/time)`



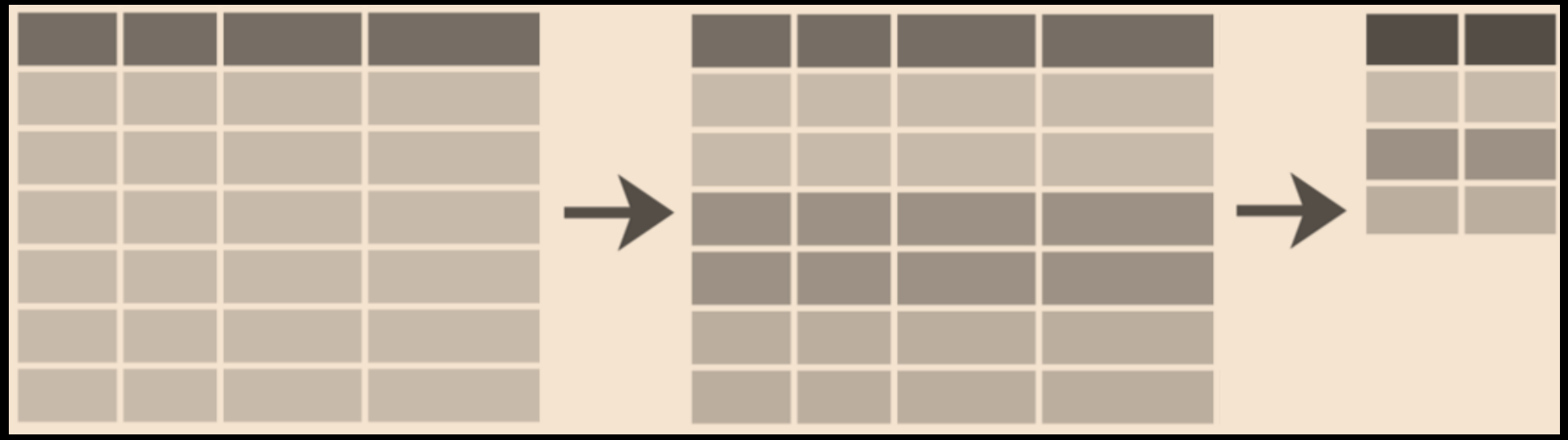
arrange()

R Base	D[order(D$subject, D$trial),]
dplyr	arrange(D, subject, trial)

Id	subject	trial	varX
10002	1	1	4
30005	1	2	6
10003	1	3	4
20002	2	1	3
10001	2	2	7
10008	2	3	3
30002	3	1	1
20005	3	2	4

group_by() - summarise()

R Base	`aggregate(D$RT, list(subject = D$Subject), mean)`
dplyr	`a <- group_by(D, subject)` `summarise(a, totalcount = sum(count))`



magrittr - %>%


%>%
Ceci n'est pas un pipe.
www.tidyverse.org
tidyverse
www.rstudio.com

```
salgo_de_casa(me_visto(me_levanto(despierto(Yo))))
```

```
Yo %>%
despierto() %>%
me_levanto() %>%
me_visto() %>%
salgo_de_casa()
```

```r
resultado_final <- sqrt(mean(abs(x)))


absoluto <- abs(x)
promedio <- mean(absoluto)
resultado_final <- sqrt(promedio)


Resultado_final <- x %>% abs %>% mean %>% sqrt
```

¿Cómo funciona %>%?

```
f(x)                    =    x %>% f

f(x,y)                  =    x %>% f(y)

f(y,x)                  =    x %>% f(y,.)

h(g(f(x)))              =    x %>% f %>% g %>% h

filter(df, v1 == 2)     =    df %>% filter(v1 == 2)
```