1- Data Manipulation

Curso Ciencia de datos en R

Escuela de Doctorado, Universidad de Alcalá, Mayo-Junio 2021

Prof. Sara Villén Pérez

0 – Intro to R

1 - Data manipulation

2 – Data visualization

3 – Automation

for

while

function

CURSO
CIENCIA DE DATOS EN R

Packages

- R Base Package: {base}
- Tidyverse Family:
  - {readr}: import and export data
  - {tibble}: enhanced dataframes
  - {dplyr}: manipulate data
  - {tidyr}: tidy data
  - {magrittr}: concatenate operations
  - {forcats}: factors
  - {stringr}: character strings
  - {lubridate}: dates and times
  - {ggplot2}: graphics
- Others

1 – Data manipulation

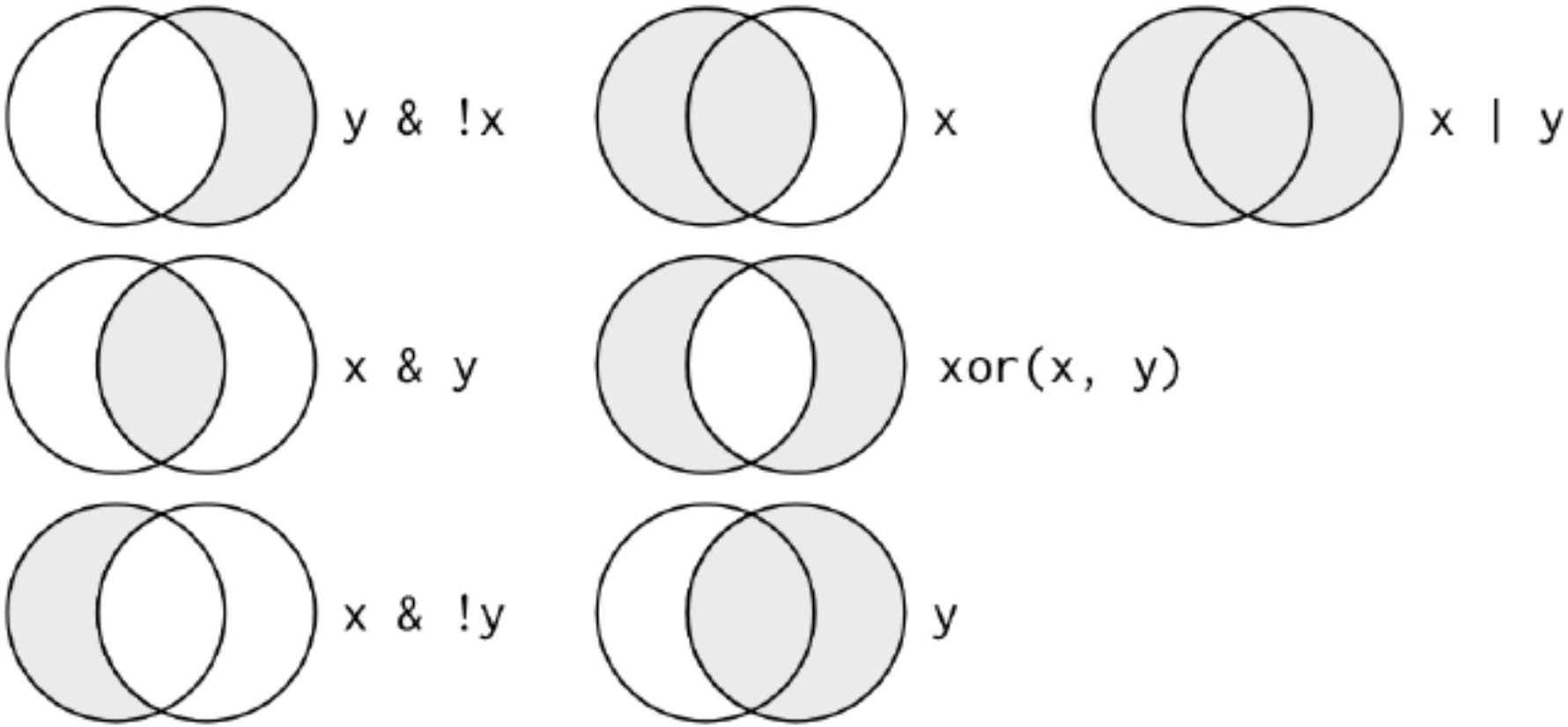
From “raw”
to useful data

1. Data type and structure
2. Data exploration
3. Data subsetting
4. Data editing
5. Data reorganization
6. Tidy / reshape data
7. Aggregate and analyze data
8. Concatenate operations
9. Special data-type manipulation
10. Data importation / exportation

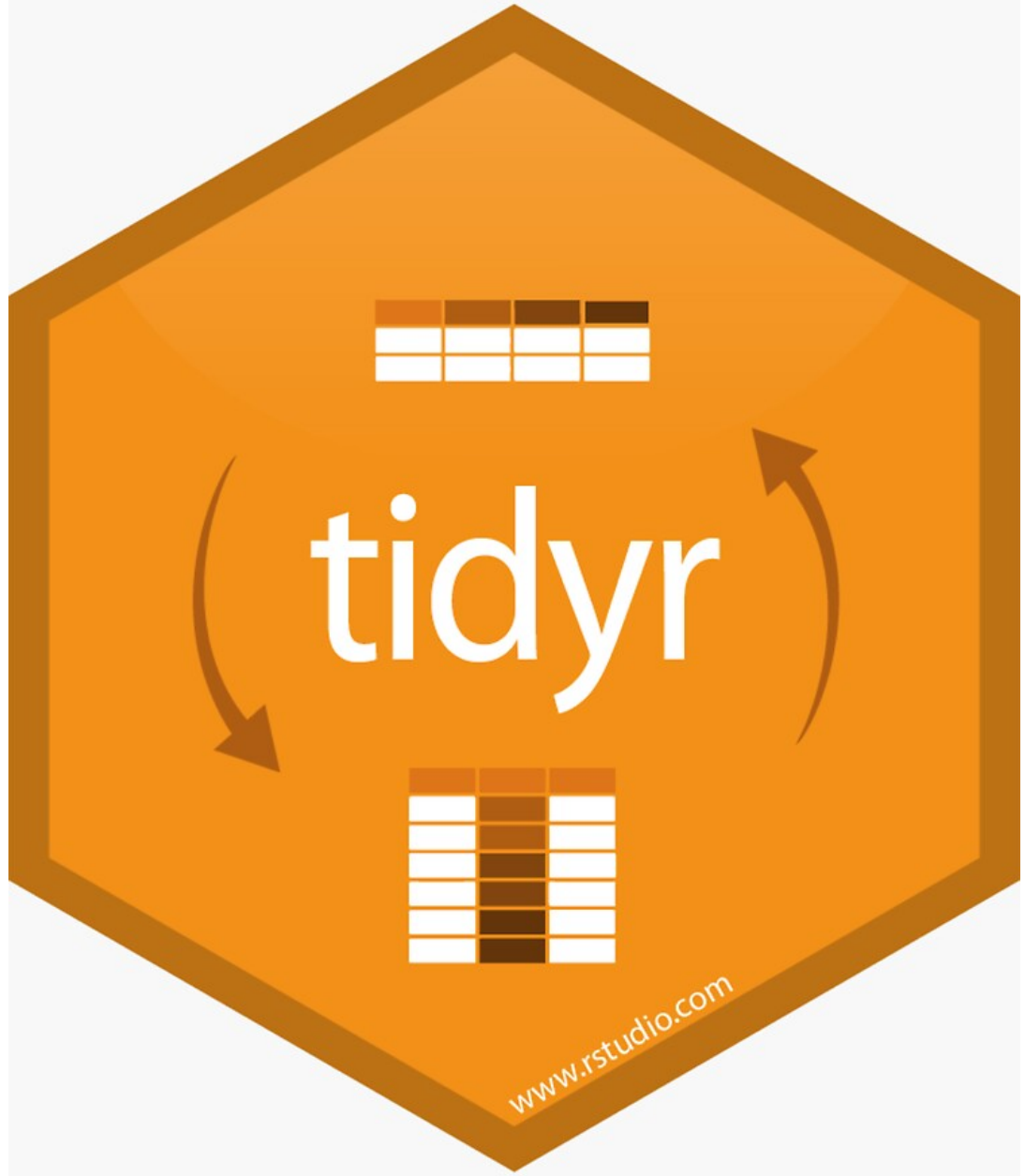
Data structure

	Homogeneous	**Heterogeneous**
1-dimension	Atomic Vector	List
2-dimensions	Matrix	Data Frame
n-dimensions	Array	

Boolean operators

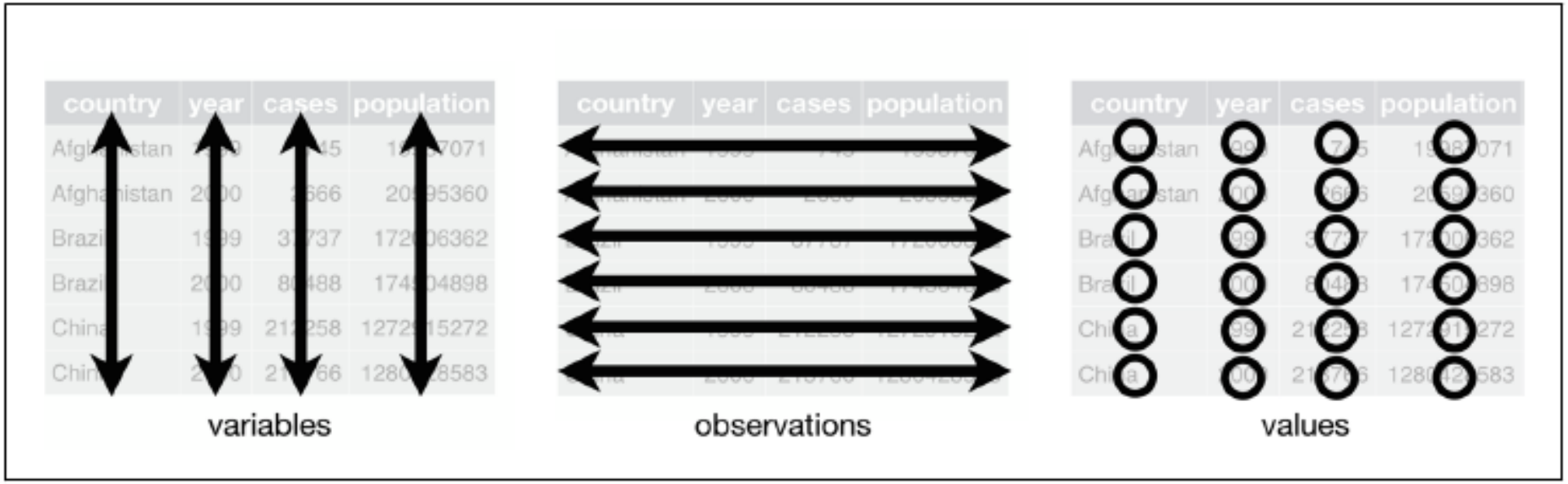


R for Data Science (2016) Hadley Wickham & Garrett Grolemund


tidyr
www.rstudio.com

The 3 principles of tidy data

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.



R for Data Science (2016) Hadley Wickham & Garrett Grolemund

Which one is tidy here?

WIDE FORMAT

Id	Variable 1	Variable 2	Variable 3
1	100	A	7
2	200	B	8
3	300	C	9

LONG FORMAT

Id	Variable name	Value
1	Variable 1	100
2	Variable 1	200
3	Variable 1	300
1	Variable 2	A
2	Variable 2	B
3	Variable 2	C
1	Variable 3	7
2	Variable 3	8
3	Variable 3	9

WIDE FORMAT
(Tidy)

Id	Variable 1	Variable 2	Variable 3
1	100	A	7
2	200	B	8
3	300	C	9

LONG FORMAT
(NOT tidy)

Id	Variable name	Value
1	Variable 1	100
2	Variable 1	200
3	Variable 1	300
1	Variable 2	A
2	Variable 2	B
3	Variable 2	C
1	Variable 3	7
2	Variable 3	8
3	Variable 3	9

Which one is tidy here?

(Repeated measures design)

WIDE FORMAT

Id	Time 1	Time 2	Time 3
1	100	400	700
2	200	500	800
3	300	600	900

LONG FORMAT

Id	Time	Value
1	1	100
2	1	200
3	1	300
1	2	400
2	2	500
3	2	600
1	3	700
2	3	800
3	3	900

WIDE FORMAT (NOT tidy)

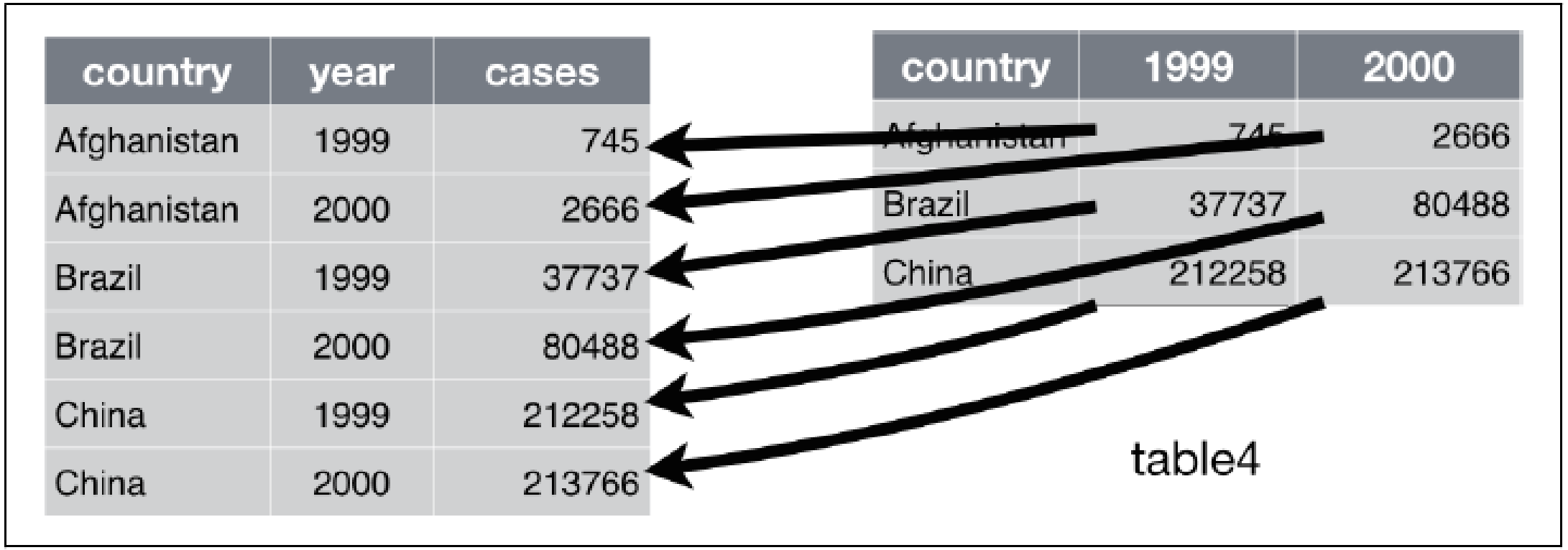
(Repeated measures design)

Id	Time 1	Time 2	Time 3
1	100	400	700
2	200	500	800
3	300	600	900

LONG FORMAT (Tidy)

Id	Time	Value
1	1	100
2	1	200
3	1	300
1	2	400
2	2	500
3	2	600
1	3	700
2	3	800
3	3	900

gather(): wide to long format



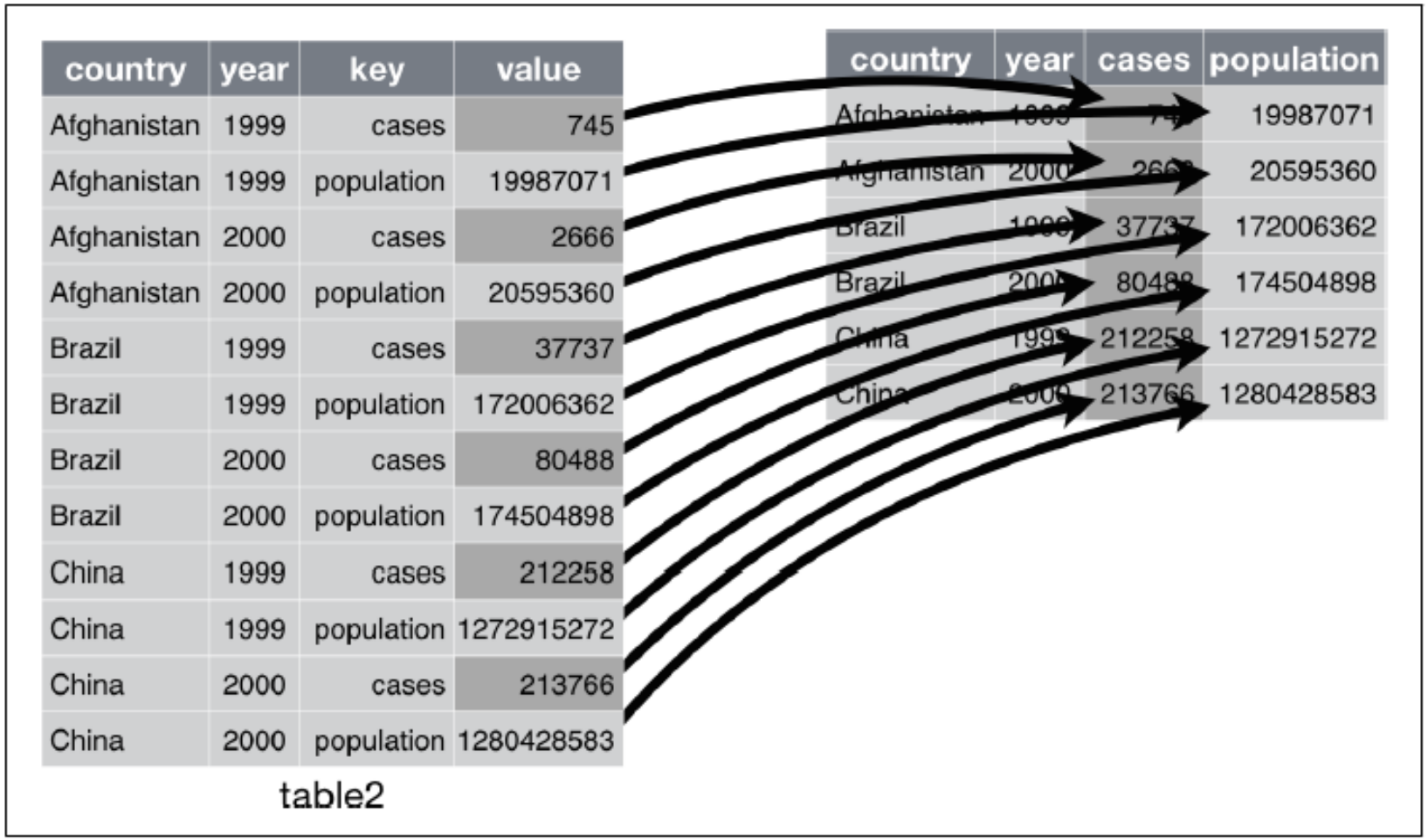
country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

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

table4

R for Data Science (2016) Hadley Wickham & Garrett Grolemund

spread(): long to wide format



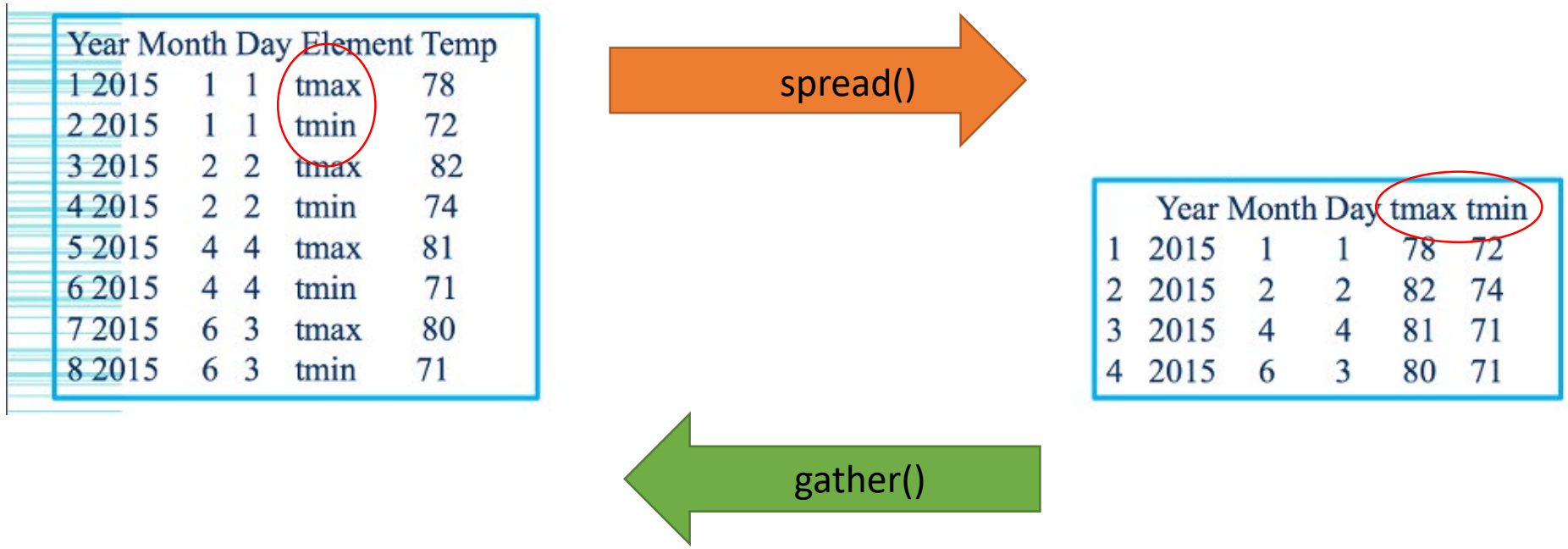
country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999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

table2

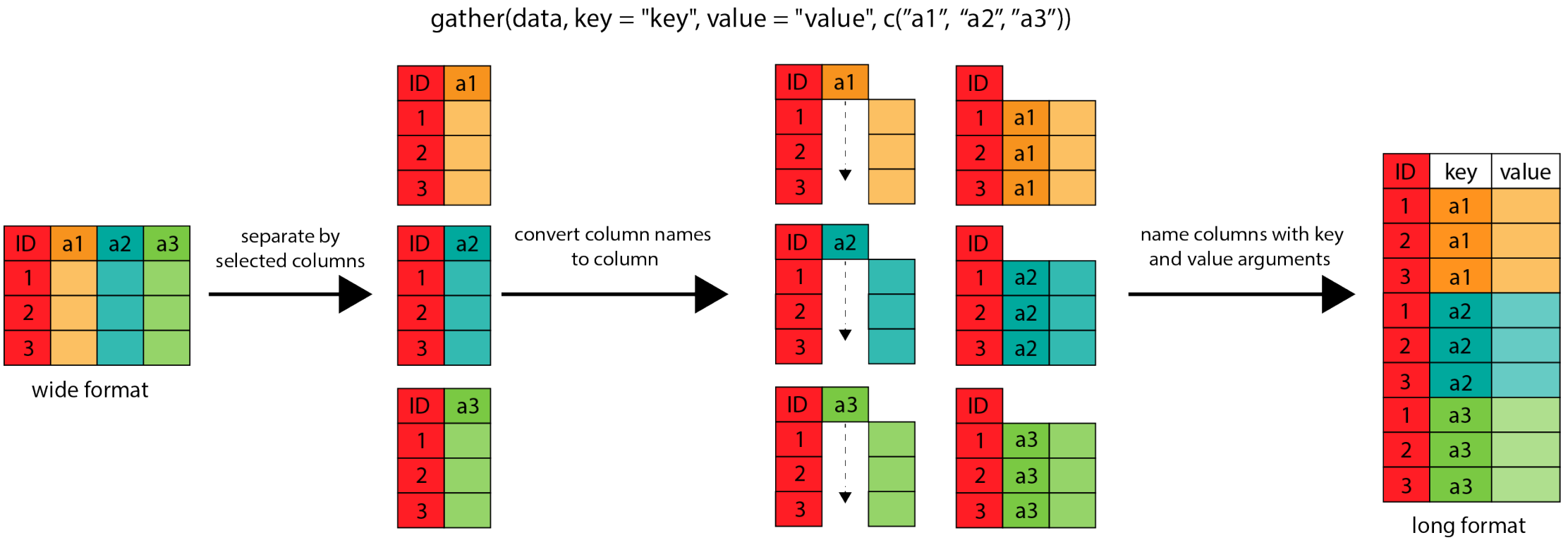
country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

R for Data Science (2016) Hadley Wickham & Garrett Grolemund

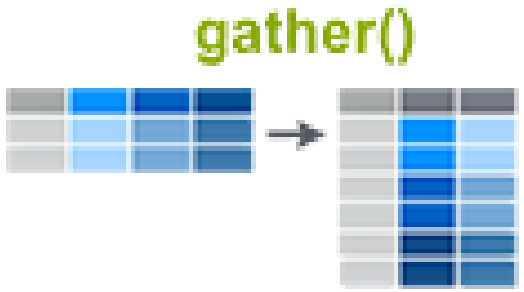
gather() & spread()

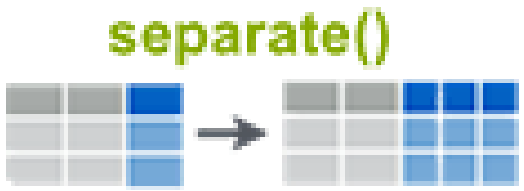


gather()

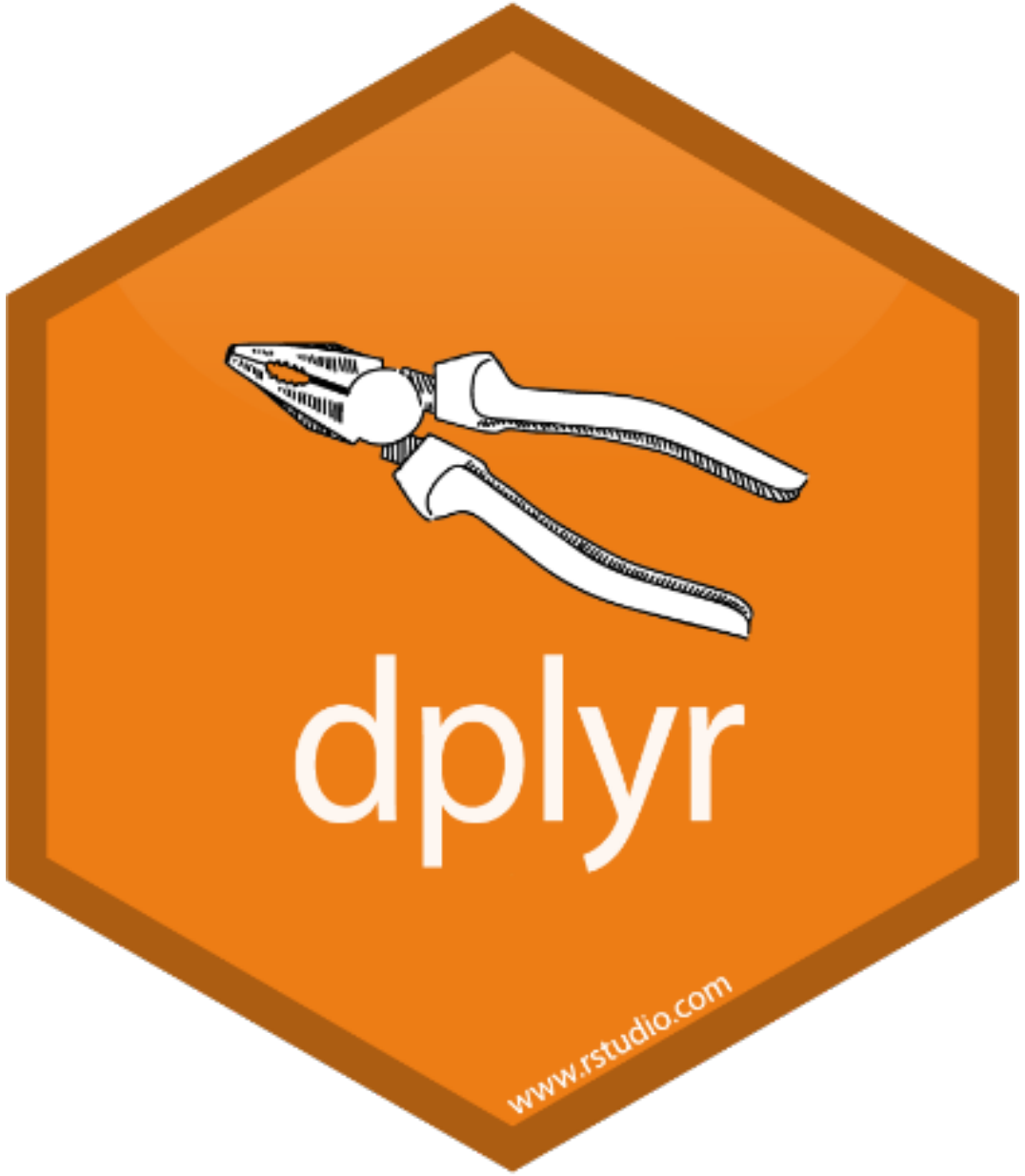

gather(data, key = "key", value = "value", c("a1", "a2", "a3"))
ID a1 a2 a3
1
2
3
wide format
separate by selected columns
ID a1
ID a2
ID a3
convert column names to column
name columns with key and value arguments
ID key value
long format

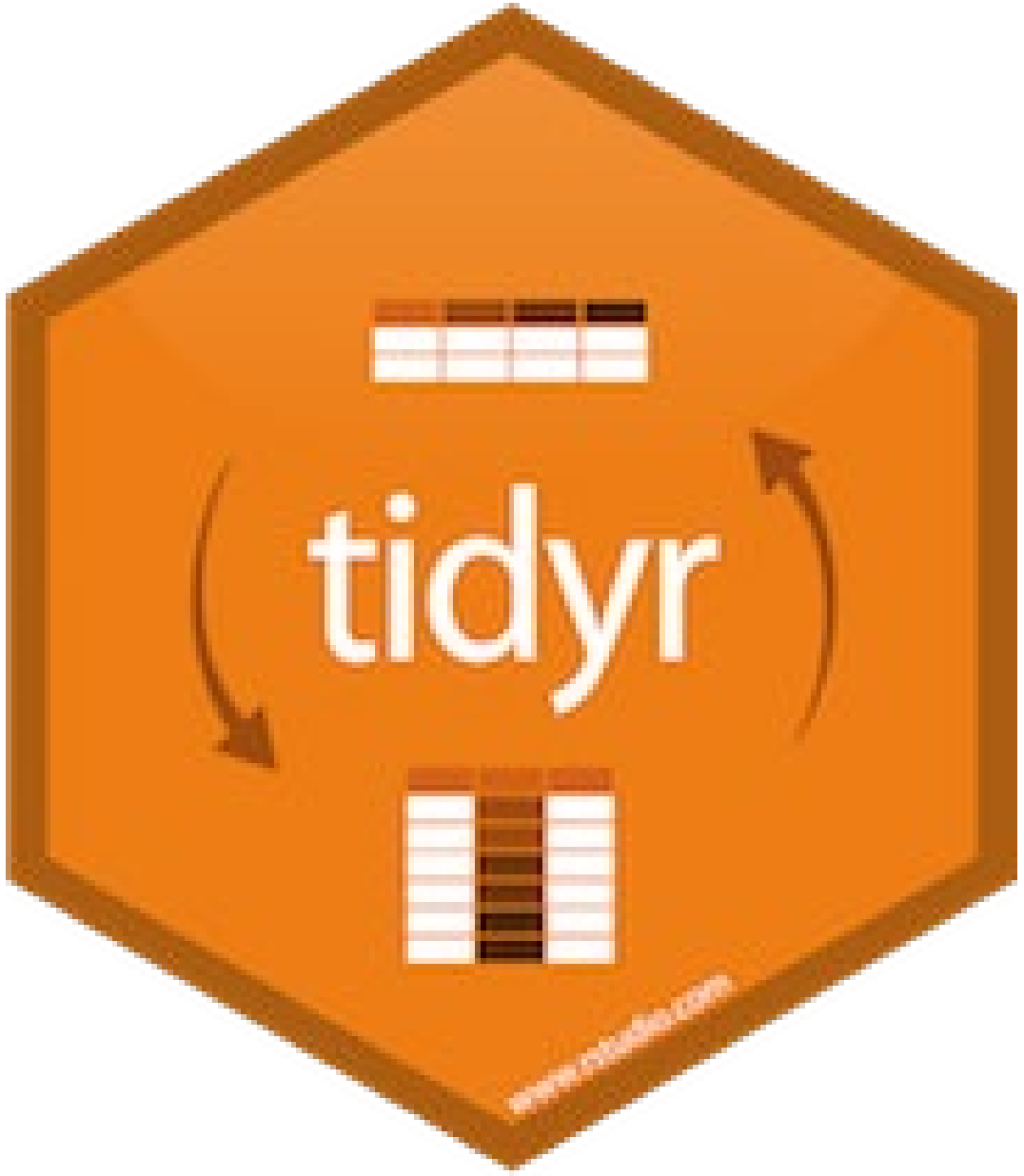
{tidyr}





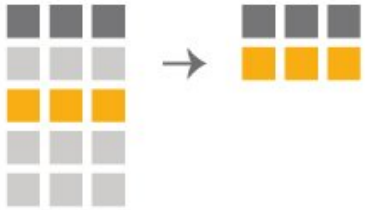
- **gather()**: collapse multiple columns into key-pair values
- **spread()**: reverse of gather. Separate one column into multiple
- **separate()**: separate one column into multiple
- **unite()**: unite multiple columns into one


dplyr
www.rstudio.com


tidyr

{dplyr} & {tidyr}

filter: extract rows that meet logical criteria
filter(Site == „Alpine“)



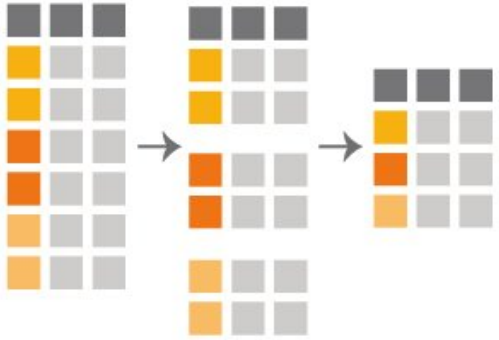
select: extract columns as table
select(Site)

mutate: compute new columns
mutate(Richness = n())

arrange: order rows
arrange(Cover)

addapted from RStudio Cheat Sheet; rstudio.com/resources/cheatsheets/

Use **group_by** to create a "grouped" copy of a table. Use **summarise** to compute table of summary.
... group_by(Site) %>% summarise(meantrait = mean(trait))



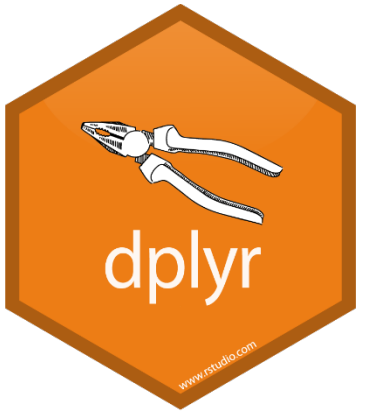
left_join: combine tables
left_join(A, B, by = c(“Taxon“)

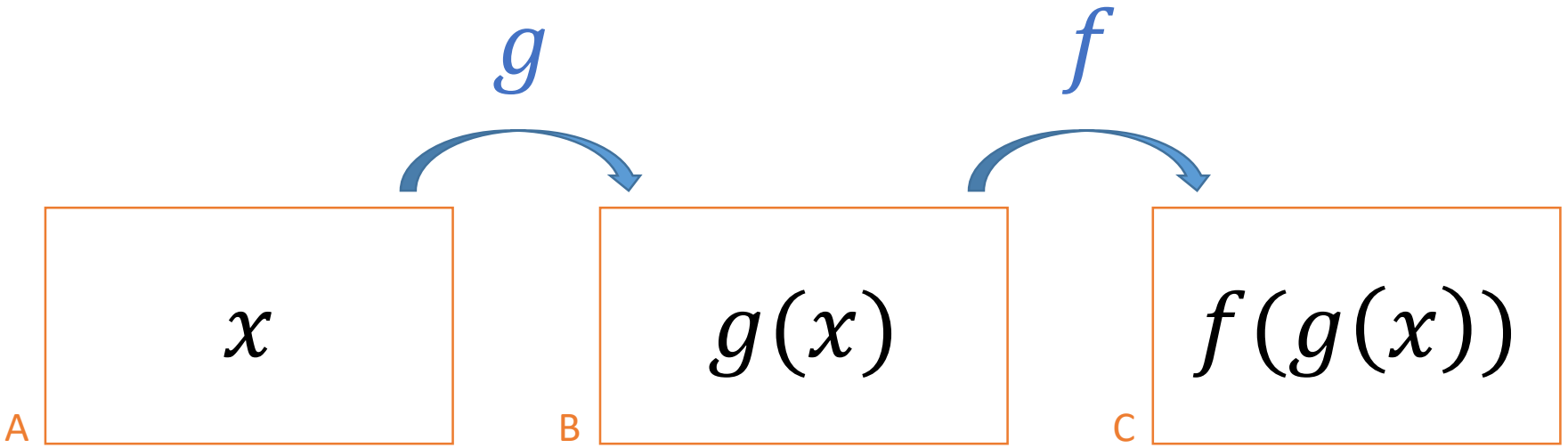
Use **gather** and **spread** to reorganize the values of a table into a new layout.
gather(data, key, value)

spread(data, key, value)

%>%
magrittr

Ceci n'est pas un pipe.





Opciones para concatenar funciones:

1. Usando paréntesis: $f(g(x))$
2. Usando objetos intermedios:

```
B <- g(x)
C <- f(B)
```

3. Sobrescribiendo el objeto original:

```
x <- g(x)
x <- f(x)
```

4. Usando pipes:

```
x %>%
  g() %>%
    f()
```

Toolbox for data wrangling in R

`magrittr` **%>%**

`readr`

`tidyr`

`dplyr`

`load` `%>%` `reshape` `%>%` `manipulate`

data wrangling

adapted from H. Wickham

Data %>% group_by() %>% summarize()

```
gapminder %>%
group_by(a) %>%
summarize(mean_b=mean(b))
```