



Sesión 2

Manipulación de tablas en R

Campamento de invierno EGOB | UC | 28 de julio, 2023

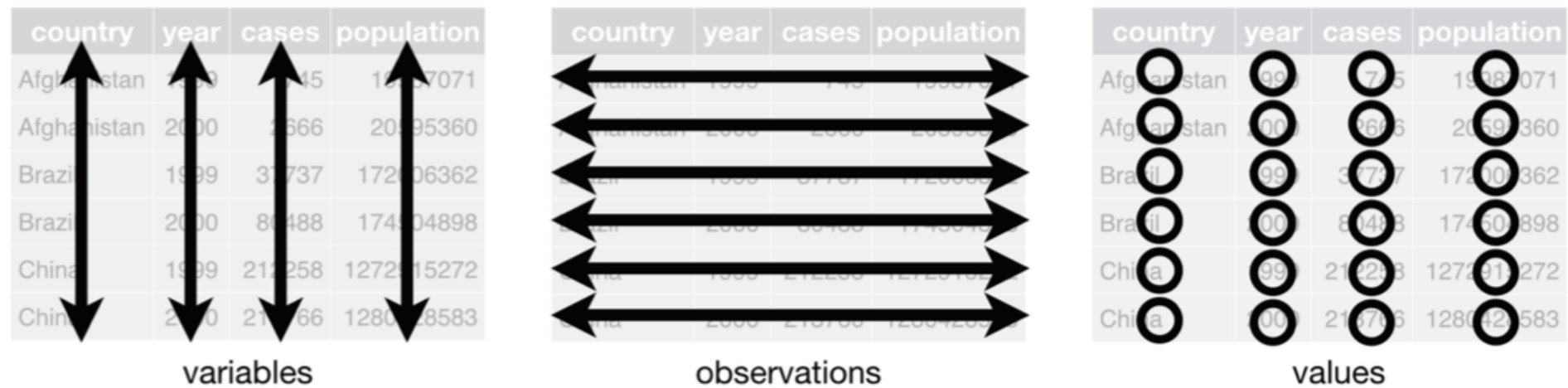
👤 **José D. Conejeros** | ✉ jdconejeros@uc.cl

Guía

1. Tablas de datos
2. Manipulación con dplyr
3. Fundir tablas de datos
4. Transformaciones globales
5. Manipulación de variables

Tablas de datos

Recordemos lo que es una tabla de datos



Manipulación de datos con dplyr

Una de las claves para trabajar con dplyr son los pipes: %>% o ▷

dplyr functions work with pipes and expect **tidy data**. In tidy data:



Each **variable** is in its own **column**

&



Each **observation**, or **case**, is in its own **row**

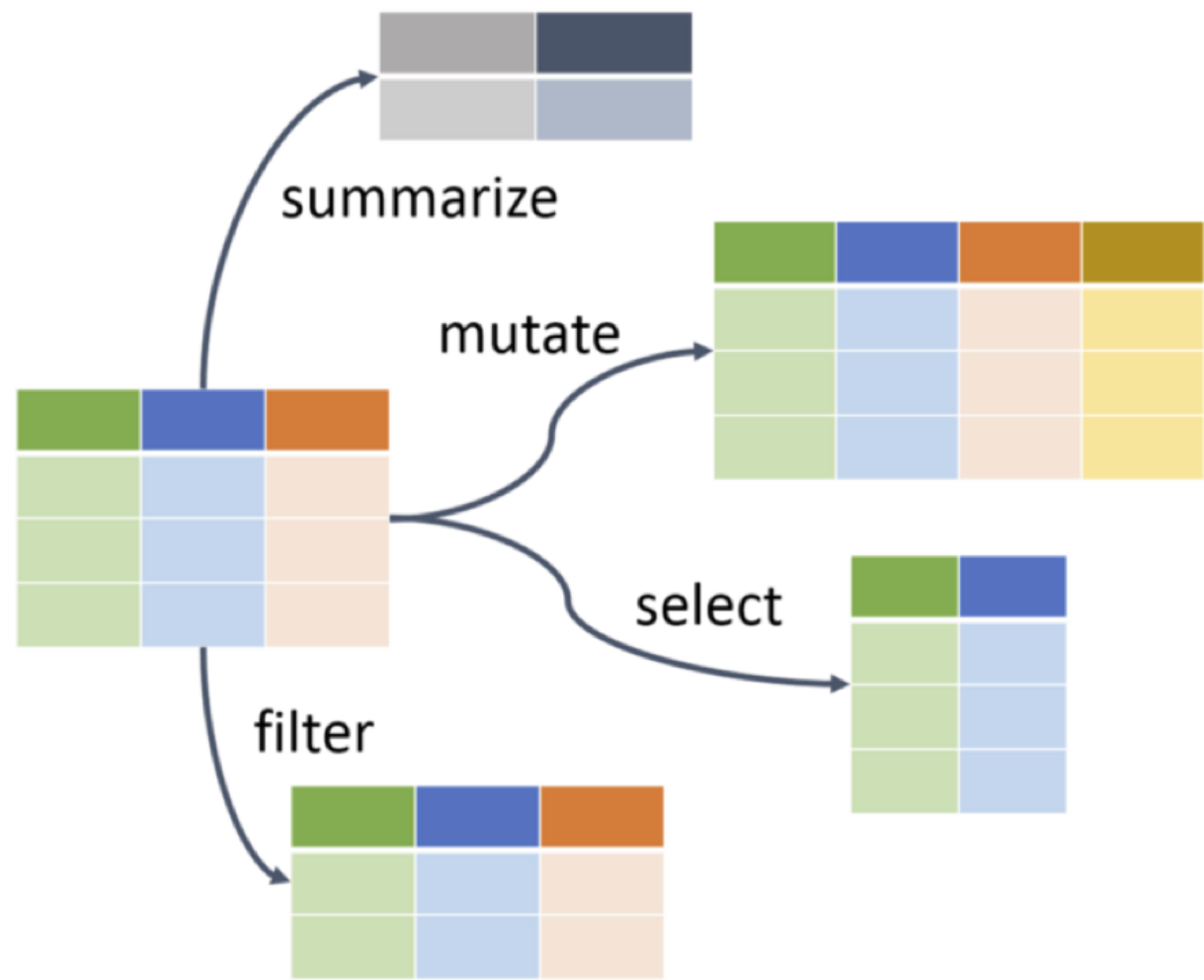


pipes

x %>% f(y)
becomes **f(x, y)**

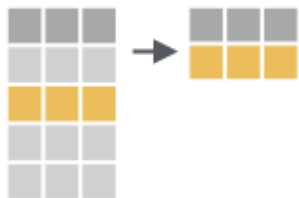
- El primer argumento siempre es un tibble o data.frame
- El resto de los argumentos indican los parametros de lo que queremos hacer
- El resultado siempre tiene estructura de tibble o data frame

Transformación usando vectores



Transformación usando vectores

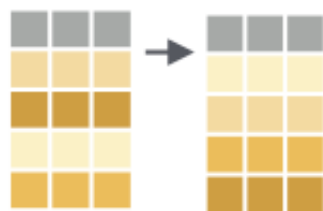
Row functions return a subset of rows as a new table.



filter(.data, ..., .preserve = FALSE) Extract rows that meet logical criteria.
`filter(mtcars, mpg > 20)`



select(.data, ...) Extract columns as a table.
`select(mtcars, mpg, wt)`

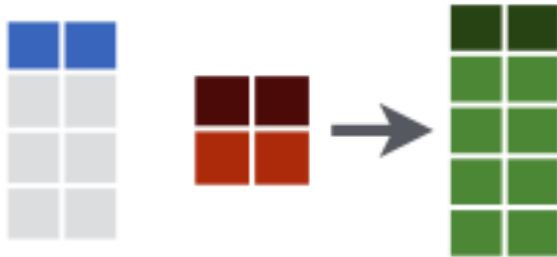


arrange(.data, ..., .by_group = FALSE) Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.
`arrange(mtcars, mpg)`
`arrange(mtcars, desc(mpg))`

Fundir tablas de datos

Unión por filas

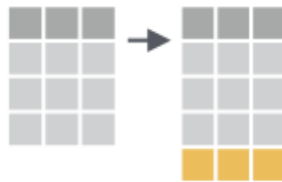
rbind - Bind rows.



			A	B	C	
		X	a	t	1	
			b	u	2	
			A	B	C	
			c	v	3	
			d	w	4	
			DF	A	B	C
			x	a	t	1
			x	b	u	2
			y	c	v	3
			y	d	w	4

bind_rows(..., .id = NULL)

Returns tables one on top of the other as a single table. Set .id to a column name to add a column of the original table names (as pictured).



add_row(.data, ..., .before = NULL, .after = NULL)

Add one or more rows to a table.

`add_row(cars, speed = 1, dist = 1)`

Fundir tablas de datos

Unión por columnas

A	B	C	D
a	t	1	3
b	u	2	2
c	v	3	NA

left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ..., keep = FALSE, na_matches = "na") Join matching values from y to x.

A	B	C	D
a	t	1	3
b	u	2	2
d	w	NA	1

right_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ..., keep = FALSE, na_matches = "na") Join matching values from x to y.

A	B	C	D
a	t	1	3
b	u	2	2

inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ..., keep = FALSE, na_matches = "na") Join data. Retain only rows with matches.

A	B	C	D
a	t	1	3
b	u	2	2
c	v	3	NA
d	w	NA	1

full_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ..., keep = FALSE, na_matches = "na") Join data. Retain all values, all rows.

Fundir tablas de datos

Unión por columnas

A	B.x	C	B.y	D
a	t	1	t	3
b	u	2	u	2
c	v	3	NA	NA

Use **by = c("col1", "col2", ...)** to specify one or more common columns to match on.

`left_join(x, y, by = "A")`

A.x	B.x	C	A.y	B.y
a	t	1	d	w
b	u	2	b	u
c	v	3	a	t

Use a named vector, **by = c("col1" = "col2")**, to match on columns that have different names in each table.

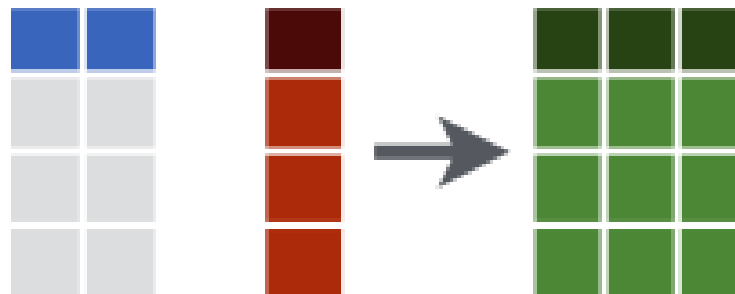
`left_join(x, y, by = c("C" = "D"))`

A1	B1	C	A2	B2
a	t	1	d	w
b	u	2	b	u
c	v	3	a	t

Use **suffix** to specify the suffix to give to unmatched columns that have the same name in both tables.

`left_join(x, y, by = c("C" = "D"),
suffix = c("1", "2"))`

cbind - Bind columns.



Transformaciones globales

Reorganización

Reshape Data - Pivot data to reorganize values into a new layout.

table4a

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

pivot_longer(data, cols, names_to = "name", values_to = "value", values_drop_na = FALSE)
"Lengthen" data by collapsing several columns into two. Column names move to a new names_to column and values to a new values_to column.

```
pivot_longer(table4a, cols = 2:3, names_to = "year", values_to = "cases")
```

table2

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

pivot_wider(data, names_from = "name", values_from = "value")
The inverse of pivot_longer(). "Widen" data by expanding two columns into several. One column provides the new column names, the other the values.

```
pivot_wider(table2, names_from = type, values_from = count)
```


Transformaciones globales

Separación

Split Cells - Use these functions to split or combine cells into individual, isolated values.

table5

country	century	year
A	19	99
A	20	00
B	19	99
B	20	00




country	year
A	1999
A	2000
B	1999
B	2000

unite(data, col, ..., sep = "_", remove = TRUE, na.rm = FALSE) Collapse cells across several columns into a single column.

`unite(table5, century, year, col = "year", sep = "")`

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M




country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172
B	2000	80K	174

separate(data, col, into, sep = "[^:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", ...) Separate each cell in a column into several columns. Also **extract()**.

`separate(table3, rate, sep = "/",
into = c("cases", "pop"))`

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M



country	year	rate
A	1999	0.7K
A	1999	19M
A	2000	2K
A	2000	20M
B	1999	37K
B	1999	172M
B	2000	80K
B	2000	174M

separate_rows(data, ..., sep = "[^:alnum:].]+", convert = FALSE) Separate each cell in a column into several rows.

`separate_rows(table3, rate, sep = "/")`

Manipulación de variables

Generación de nuevas variables

MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).



`mutate(.data, ..., .keep = "all", .before = NULL, .after = NULL)` Compute new column(s). Also **`add_column()`**, **`add_count()`**, and **`add_tally()`**.
`mutate(mtcars, gpm = 1 / mpg)`



`transmute(.data, ...)` Compute new column(s), drop others.
`transmute(mtcars, gpm = 1 / mpg)`

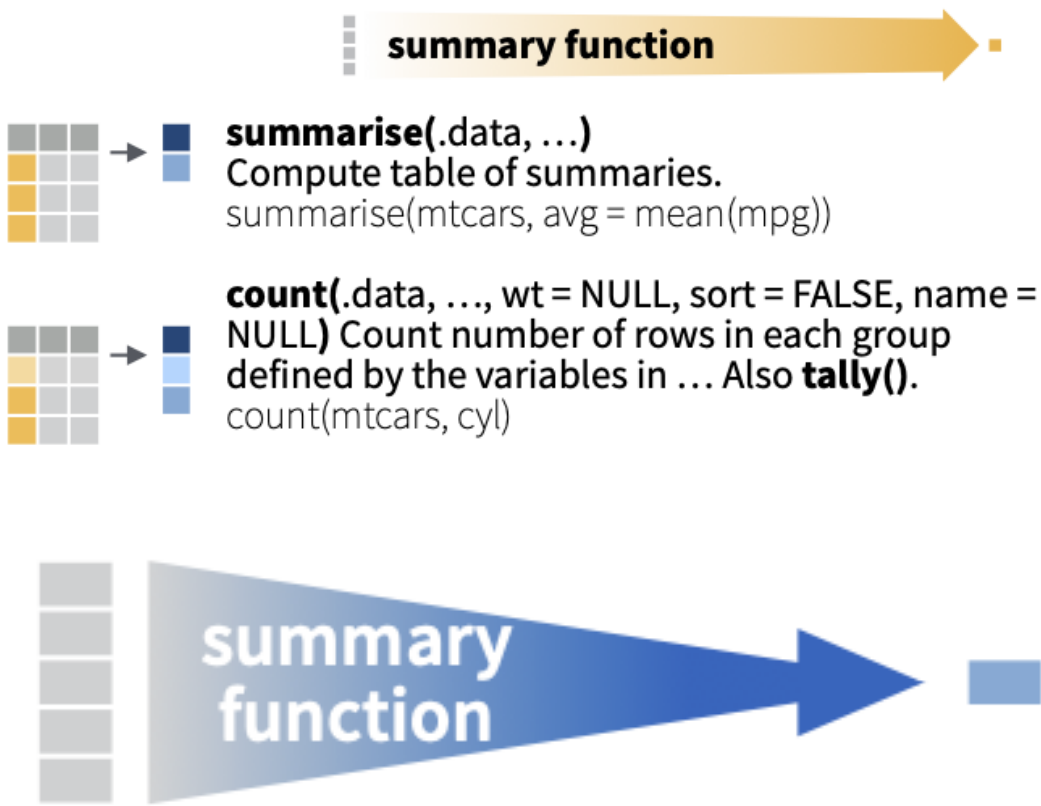


`rename(.data, ...)` Rename columns. Use **`rename_with()`** to rename with a function.
`rename(cars, distance = dist)`

Análisis de variables

Descriptivos

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).



Análisis de variables

Descriptivos

summarise() applies summary functions to columns to create a new table. Summary functions take vectors as input and return single values as output.



COUNT

dplyr::n() - number of values/rows
dplyr::n_distinct() - # of uniques
sum(!is.na()) - # of non-NA's

POSITION

mean() - mean, also **mean(!is.na())**
median() - median

LOGICAL

mean() - proportion of TRUE's
sum() - # of TRUE's

ORDER

dplyr::first() - first value
dplyr::last() - last value
dplyr::nth() - value in nth location of vector

RANK

quantile() - nth quantile
min() - minimum value
max() - maximum value

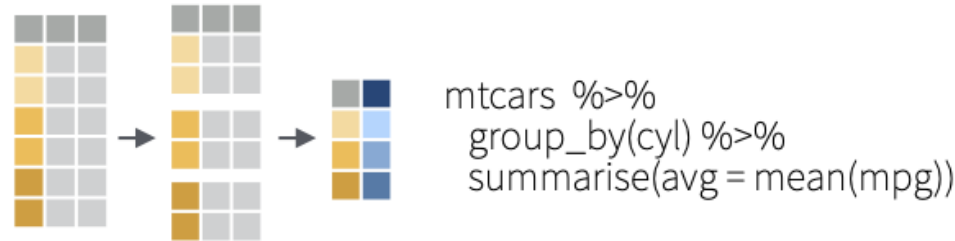
SPREAD

IQR() - Inter-Quartile Range
mad() - median absolute deviation
sd() - standard deviation
var() - variance

Análisis de variables:

Agrupamiento por columnas

Use **group_by**(.data, ..., .add = FALSE, .drop = TRUE) to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.

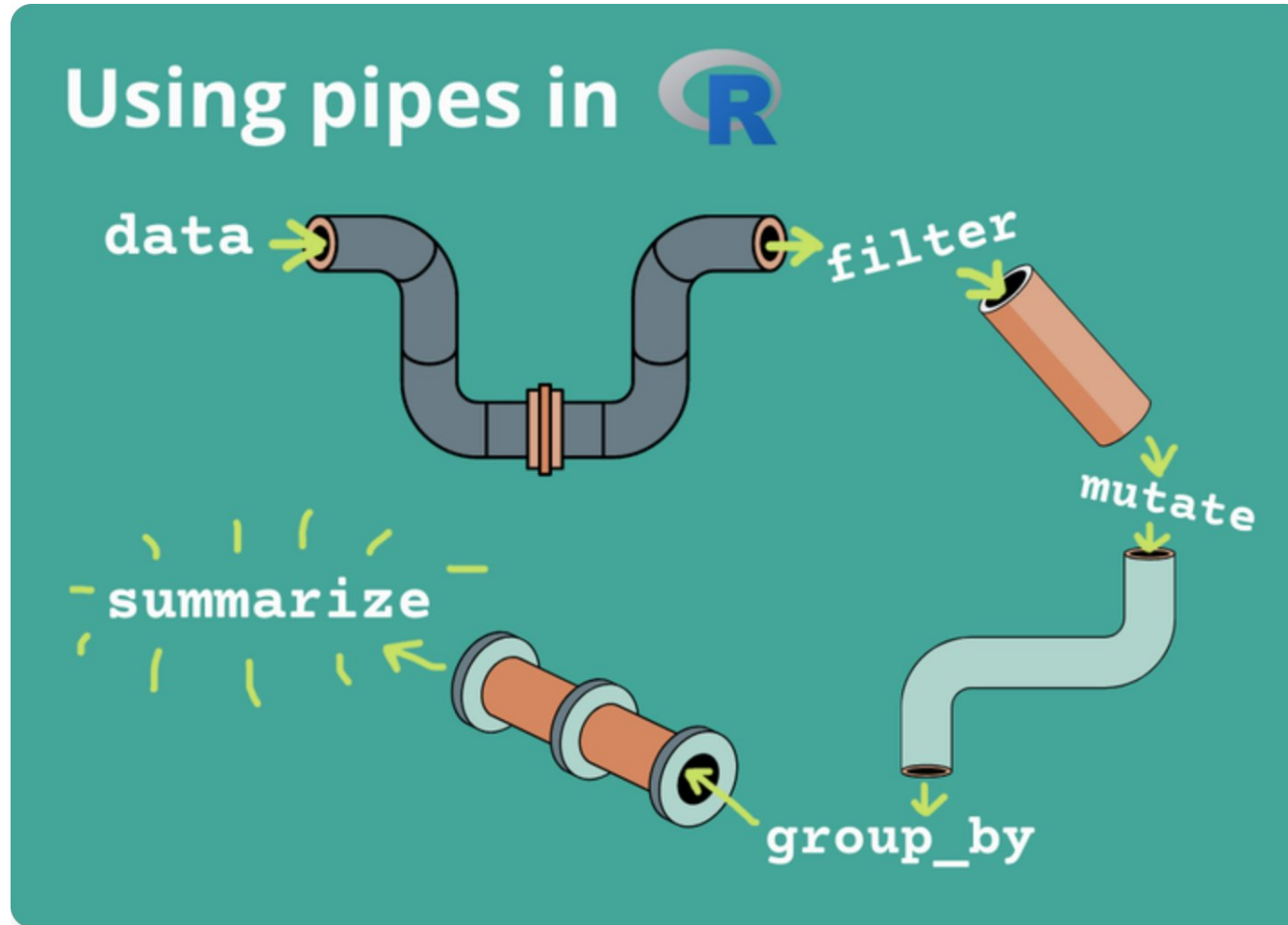


Use **rowwise**(.data, ...) to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidyr cheat sheet for list-column workflow.



ungroup(x, ...) Returns ungrouped copy of table.
`ungroup(g_mtcars)`

Uso de tuberías



Referencias

Wickham, H., & Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc.". Cap. 9 al 12. Recurso en línea: <https://r4ds.hadley.nz/>

Urdinez, F., & Cruz, A. (2020). R for Political Data Science: A Practical Guide. CRC Press. Cap. 1 al 4. Recurso en línea en español: <https://arcruz0.github.io/libroadp/>

Posit Cheatsheets ("hojas de trucos"): <https://posit.co/resources/cheatsheets/?type=posit-cheatsheets/>

Página oficial de Tidyverse: <https://www.tidyverse.org/>



Sesión 2

Manipulación de tablas

28 de julio, 2023

👤 **José D. Conejeros** | ✉ jdconejeros@uc.cl | 🌐 JDConejeros