

Joining and Pivoting

Advanced data manipulation

Download the section 6 .Rmd handout to
STAT240/lecture/06-join-pivot.

Download two datasets to STAT240/data:

- grocery-list.csv
- supermarket.csv

Material in this section is covered by Chapter 8 on
the notes website.

Joining combines information from two dataframes.

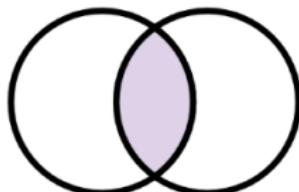
- **Mutating joins** append columns together
- **Filtering joins** keeps rows based on another df

Mutating join arguments:

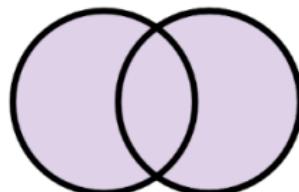
- Two data frames
- by = Names of columns to join

`left_join()`, `right_join()`, `full_join()`,
`inner_join()`

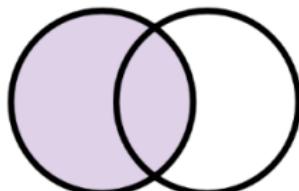
Which dataset is given “priority”?



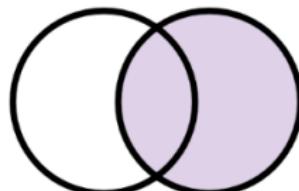
`inner_join(x, y)`



`full_join(x, y)`



`left_join(x, y)`



`right_join(x, y)`

from tavareshugo.github.io

`left_join(x, y)` keeps all rows in x.

- x is “nailed down”
- Then y columns are added
- Can induce NA in y's columns

`right_join(x, y)` keeps all rows in y.

Be mindful of column names!

- Best practice: provide “by”
- R will try to match names

If the dataframes have no columns in common, and by is not given, we get an error.

`full_join()` keeps all rows from both dataframes.

- Like `left_join()`, but all rows get added
- Can induce NA values for columns in x or y

This is not the same as “stacking” the dataframes.

`inner_join()` keeps only rows appearing in both dataframes.

- Does not induce NA values

`full_join()` and `inner_join()` are symmetric.

Predict what will happen when joining
band_instruments and band_members.

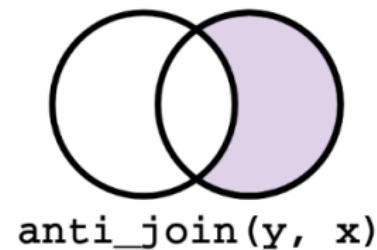
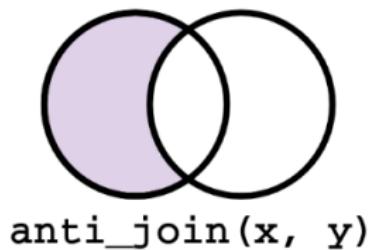
- How many rows and columns will there be?
- Will there be any NAs?

Uncomment the lines to see if you were right.

Filtering joins remove rows of the x dataset.

- `semi_join()` keep rows that also appear in y
- `anti_join()` keep rows that don't appear in y

No columns from y appear in the output.



from tavareshugo.github.io

Now predict the output when filter-joining
band_instruments and band_members.

- How many rows and columns will there be?
- Will there be any NAs?

Uncomment the lines to see if you were right.

Pivoting changes the shape of the dataframe while retaining all of its information.

Datasets can be wide or long, depending on how we want to structure the rows.

	a	b	c
1			
2			

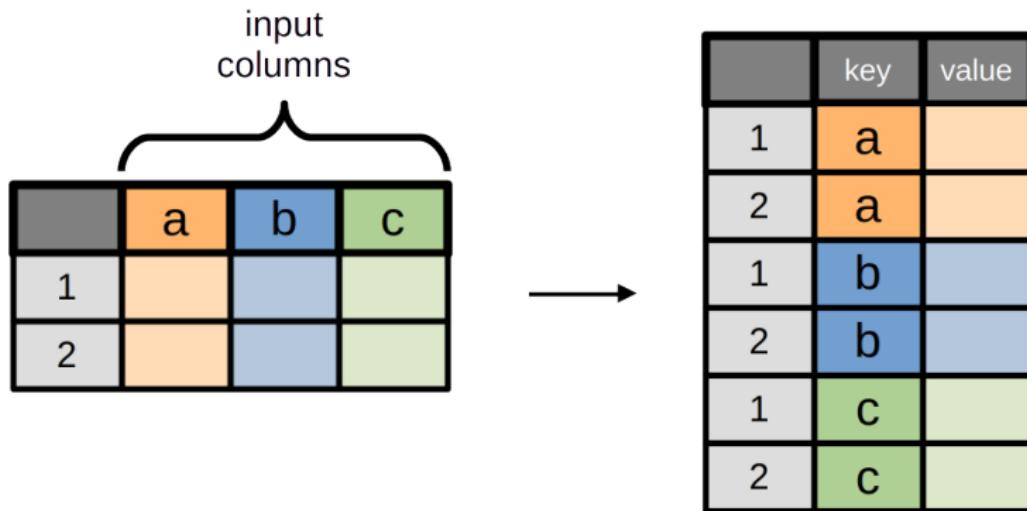
wide form

1	a	
2	a	
1	b	
2	b	
1	c	
2	c	

long form

Switch between them with `pivot_longer()` and `pivot_wider()`.

pivot_longer():



`pivot_longer()` increases rows and decreases columns.

- We specify existing columns

Those columns will be merged into one long column.

pivot_wider():

input names input values

The diagram illustrates the `pivot_wider()` function. On the left, an input data frame is shown in long format. It has columns for 'key' and 'value'. The 'key' column contains values 'a', 'a', 'b', 'b', 'c', 'c'. The 'value' column contains values '1', '2', '1', '2', '1', '2'. Above the first two columns, there are two curly braces: one for the 'key' column and another for the 'value' column, both labeled 'input names' and 'input values' respectively. An arrow points from this input frame to an output frame on the right. The output frame is in wide format. It has four columns: 'key' (gray), 'a' (orange), 'b' (blue), and 'c' (green). The 'key' column contains values '1' and '2'. The 'a' column contains values '1' and '2'. The 'b' column contains values '1' and '2'. The 'c' column contains values '1' and '2'.

	key	value
1	a	1
2	a	2
1	b	1
2	b	2
1	c	1
2	c	2

→

	key	a	b	c
1		1	1	1
2		2	2	2

`pivot_wider()` decreases rows and increases columns.

- We specify the column to split and the values for the new columns

These become the names and values of the new columns.