Computing in Context: Fall 2024

Lab 10 | Split-Apply-Combine

Dr Sarah Ashcroft-Jones | Mailman School of Public Health

Main operations in pandas

Combine information across tables (**join, anti-join**)

- **Join**: e.g., combine tables from multiple sources
- **Anti-join**: e.g., outliers, exclude them from the existing table

Compute summary tables (**split-apply-combine**)

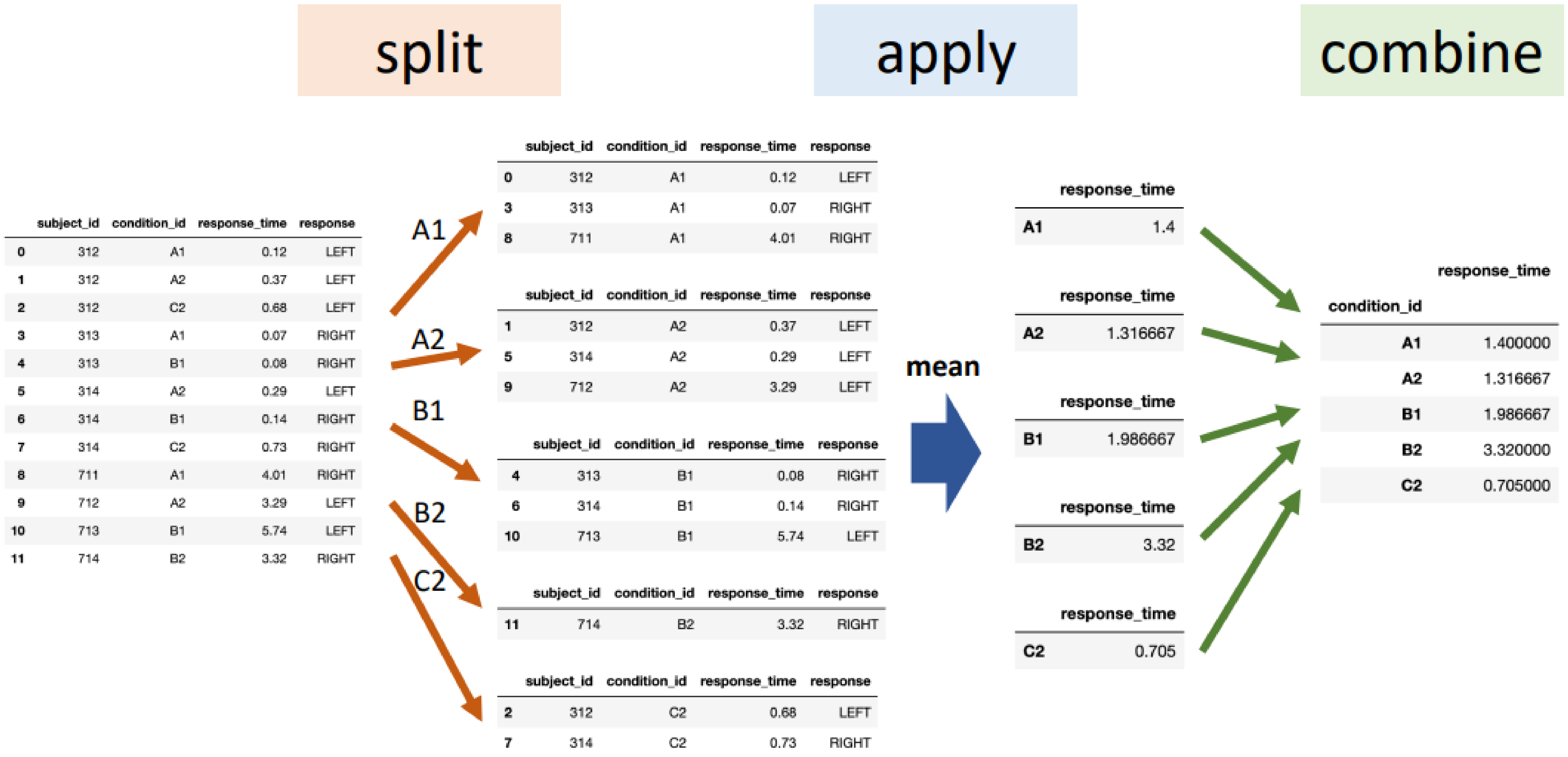
- E.g., compute average measurement

How to split-apply-combine?

Most tabular data tools (including Python) have a way to vectorize the standard split-apply-combine operations, using a “group-by” command

In addition, Pandas has the “pivot-table” command that can be used to simplify the creation of more complex summary tables (we have seen this before).

The basic structure of most analyses



	subject_id	condition_id	response_time	response
0	312	A1	0.12	LEFT
1	312	A2	0.37	LEFT
2	312	C2	0.68	LEFT
3	313	A1	0.07	RIGHT
4	313	B1	0.08	RIGHT
5	314	A2	0.29	LEFT
6	314	B1	0.14	RIGHT
7	314	C2	0.73	RIGHT
8	711	A1	4.01	RIGHT
9	712	A2	3.29	LEFT
10	713	B1	5.74	LEFT
11	714	B2	3.32	RIGHT

	subject_id	condition_id	response_time	response
0	312	A1	0.12	LEFT
3	313	A1	0.07	RIGHT
8	711	A1	4.01	RIGHT

	subject_id	condition_id	response_time	response
1	312	A2	0.37	LEFT
5	314	A2	0.29	LEFT
9	712	A2	3.29	LEFT

	subject_id	condition_id	response_time	response
4	313	B1	0.08	RIGHT
6	314	B1	0.14	RIGHT
10	713	B1	5.74	LEFT

	subject_id	condition_id	response_time	response
11	714	B2	3.32	RIGHT

	subject_id	condition_id	response_time	response
2	312	C2	0.68	LEFT
7	314	C2	0.73	RIGHT

	response_time
A1	1.4

	response_time
A2	1.316667

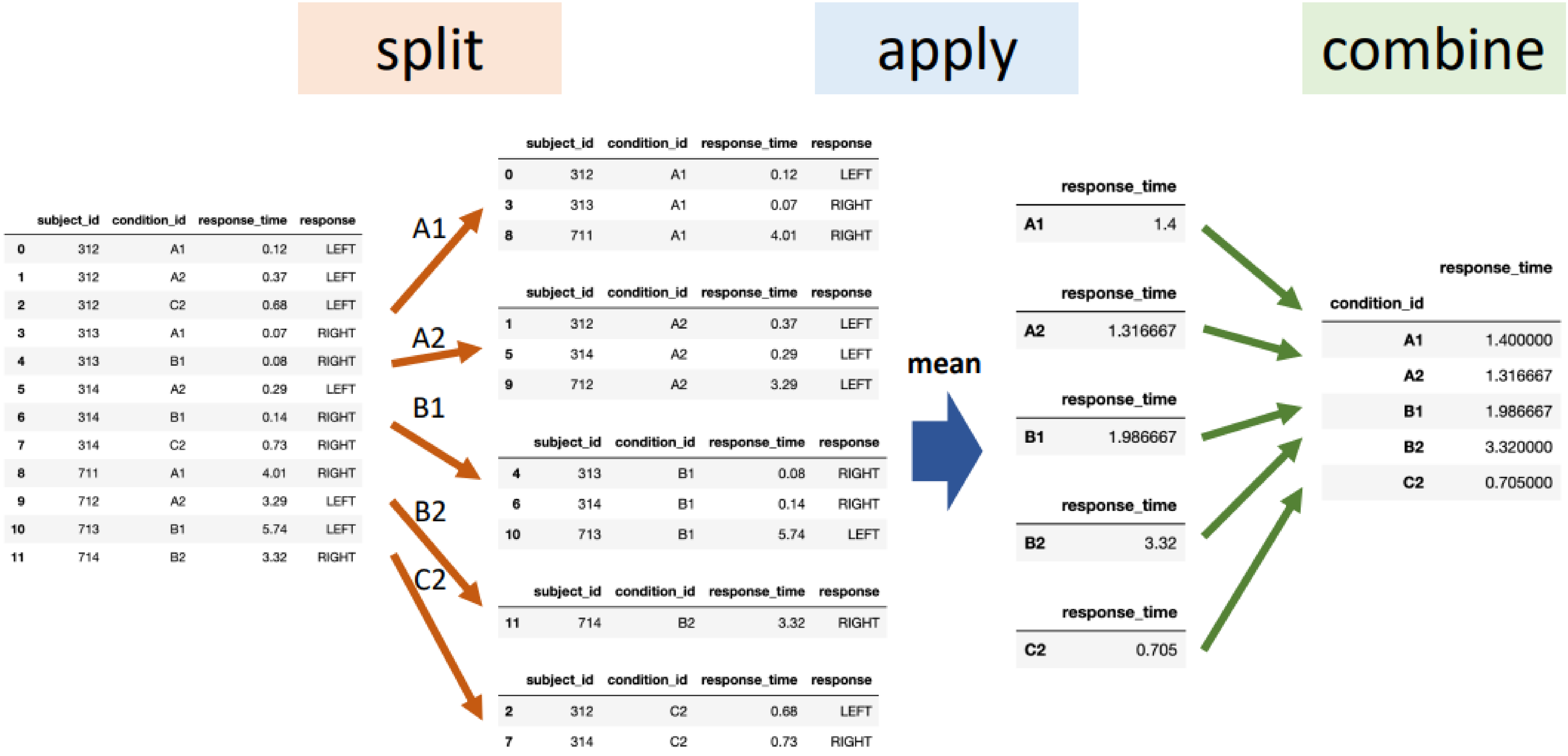
	response_time
B1	1.986667

	response_time
B2	3.32

	response_time
C2	0.705

	response_time
condition_id	
A1	1.400000
A2	1.316667
B1	1.986667
B2	3.320000
C2	0.705000

```
df.groupby('condition_id')['response_time'].mean()
```



	subject_id	condition_id	response_time	response
0	312	A1	0.12	LEFT
1	312	A2	0.37	LEFT
2	312	C2	0.68	LEFT
3	313	A1	0.07	RIGHT
4	313	B1	0.08	RIGHT
5	314	A2	0.29	LEFT
6	314	B1	0.14	RIGHT
7	314	C2	0.73	RIGHT
8	711	A1	4.01	RIGHT
9	712	A2	3.29	LEFT
10	713	B1	5.74	LEFT
11	714	B2	3.32	RIGHT

	subject_id	condition_id	response_time	response
0	312	A1	0.12	LEFT
3	313	A1	0.07	RIGHT
8	711	A1	4.01	RIGHT

	subject_id	condition_id	response_time	response
1	312	A2	0.37	LEFT
5	314	A2	0.29	LEFT
9	712	A2	3.29	LEFT

	subject_id	condition_id	response_time	response
4	313	B1	0.08	RIGHT
6	314	B1	0.14	RIGHT
10	713	B1	5.74	LEFT

	subject_id	condition_id	response_time	response
11	714	B2	3.32	RIGHT

	subject_id	condition_id	response_time	response
2	312	C2	0.68	LEFT
7	314	C2	0.73	RIGHT

	response_time
A1	1.4

	response_time
A2	1.316667

	response_time
B1	1.986667

	response_time
B2	3.32

	response_time
C2	0.705

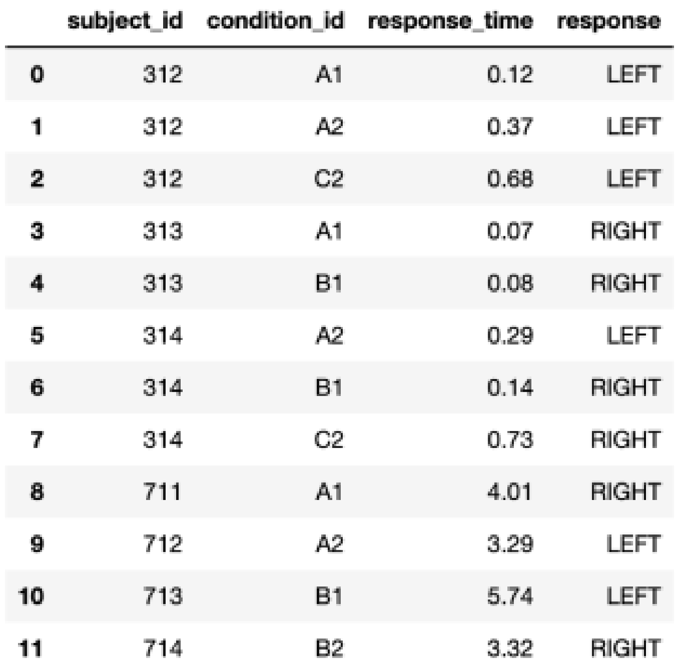
	response_time
condition_id	
A1	1.400000
A2	1.316667
B1	1.986667
B2	3.320000
C2	0.705000

```
data.pivot_table(
    index='condition_id', columns='response',
    values='response_time', aggfunc='mean',
)
```

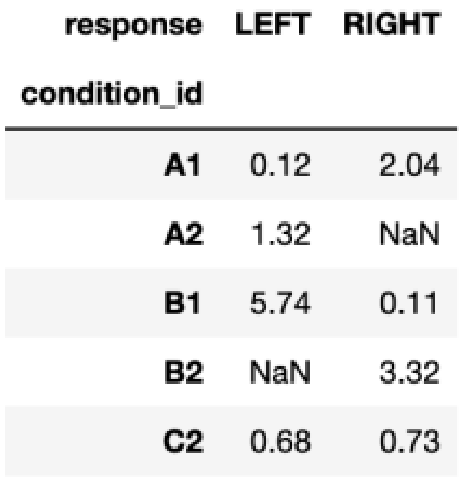
split

apply

combine



	subject_id	condition_id	response_time	response
0	312	A1	0.12	LEFT
1	312	A2	0.37	LEFT
2	312	C2	0.68	LEFT
3	313	A1	0.07	RIGHT
4	313	B1	0.08	RIGHT
5	314	A2	0.29	LEFT
6	314	B1	0.14	RIGHT
7	314	C2	0.73	RIGHT
8	711	A1	4.01	RIGHT
9	712	A2	3.29	LEFT
10	713	B1	5.74	LEFT
11	714	B2	3.32	RIGHT



response	**LEFT**	**RIGHT**
condition_id		
A1	0.12	2.04
A2	1.32	NaN
B1	5.74	0.11
B2	NaN	3.32
C2	0.68	0.73

Hands on practice:
download Jupyter Notebooks + data