Lecture 20
Combining Datasets

Dennis Sun
Stanford University
DATASCI / STATS 112

March 1, 2023

Why Combine Datasets?

Sometimes, information is spread across multiple datasets.

For example, suppose we want to know which manufacturer's planes are most common.

One dataset may contain information about flights…

```
import pandas as pd
data_dir = "http://dlsun.github.io/stats112/data/nycflights13/"
df_flights = pd.read_csv(data_dir + "flights.csv")
df_flights
```

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	1	1	517.0	515	...	227.0	1400	5	15	N14228
1	2013	1	1	533.0	529	...	227.0	1416	5	29	N24211
2	2013	1	1	542.0	540	...	160.0	1089	5	40	N619AA
3	2013	1	1	544.0	545	...	183.0	1576	5	45	N804JB
4	2013	1	1	554.0	600	...	116.0	762	6	0	N668DN
...	...	...	...	...	...	...	...	...	...	...	...
336771	2013	9	30	NaN	1455	...	NaN	213	14	55	NaN
336772	2013	9	30	NaN	2200	...	NaN	198	22	0	NaN
336773	2013	9	30	NaN	1210	...	NaN	764	12	10	N535MQ
336774	2013	9	30	NaN	1159	...	NaN	419	11	59	N511MQ
336775	2013	9	30	NaN	840	...	NaN	431	8	40	N839MQ

336776 rows × 18 columns

Why Combine Datasets?

...while another contains information about planes.

```
df_planes = pd.read_csv(data_dir + "planes.csv")
df_planes
```

	tailnum	year	type	manufacturer	model	engines	seats	speed	engine
0	N10156	2004.0	Fixed wing multi engine	EMBRAER	EMB-145XR	2	55	NaN	Turbo-fan
1	N102UW	1998.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan
2	N103US	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan
3	N104UW	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan
4	N10575	2002.0	Fixed wing multi engine	EMBRAER	EMB-145LR	2	55	NaN	Turbo-fan
...	...	...	...	...	...	...	...	...	...
3317	N997AT	2002.0	Fixed wing multi engine	BOEING	717-200	2	100	NaN	Turbo-fan
3318	N997DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS AIRCRAFT CO	MD-88	2	142	NaN	Turbo-fan
3319	N998AT	2002.0	Fixed wing multi engine	BOEING	717-200	2	100	NaN	Turbo-fan
3320	N998DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88	2	142	NaN	Turbo-jet
3321	N999DN	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88	2	142	NaN	Turbo-jet

3322 rows × 9 columns

Joining multiple datasets is one way to fulfill the data collection "complexity" requirement for the final project!

1. Joining Datasets on a Key
2. Keys with Multiple Columns
3. Summary
4. Reminders

1 Joining Datasets on a Key

2 Keys with Multiple Columns

3 Summary

4 Reminders

Keys

Planes are uniquely identified by their *tail number* (`tailnum`).

`df_flights`

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	1	1	517.0	515	...	227.0	1400	5	15	N14228
1	2013	1	1	533.0	529	...	227.0	1416	5	29	N24211
2	2013	1	1	542.0	540	...	160.0	1089	5	40	N619AA
3	2013	1	1	544.0	545	...	183.0	1576	5	45	N804JB
4	2013	1	1	554.0	600	...	116.0	762	6	0	N668DN
...	...	...	...	...	...	...	...	...	...	...	...
336771	2013	9	30	NaN	1455	...	NaN	213	14	55	NaN
336772	2013	9	30	NaN	2200	...	NaN	198	22	0	NaN
336773	2013	9	30	NaN	1210	...	NaN	764	12	10	N535MQ
336774	2013	9	30	NaN	1159	...	NaN	419	11	59	N511MQ
336775	2013	9	30	NaN	840	...	NaN	431	8	40	N839MQ

336776 rows × 18 columns

`df_planes`

	tailnum	year	type	manufacturer	model
0	N10156	2004.0	Fixed wing multi engine	EMBRAER	EMB-145XR
1	N102UW	1998.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214
2	N103US	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214
3	N104UW	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214
4	N10575	2002.0	Fixed wing multi engine	EMBRAER	EMB-145LR
...	...	...	...	...	...
3317	N997AT	2002.0	Fixed wing multi engine	BOEING	717-200
3318	N997DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS AIRCRAFT CO	MD-88
3319	N998AT	2002.0	Fixed wing multi engine	BOEING	717-200
3320	N998DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88
3321	N999DN	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88

3322 rows × 9 columns

`tailnum` is a foreign key of `df_flights`. It points to the primary key of another table.

`tailnum` is the primary key of `df_planes`. It uniquely identifies a plane.

A **primary key** is a column (or a set of columns) that uniquely identifies observations in a data frame.

A **foreign key** is a column (or a set of columns) that points to the primary key of another data frame.

Joining on a Key

The Pandas function `.merge()` can be used to join two `DataFrame`s on a key.

```
df_joined = df_flights.merge(df_planes, on="tailnum")
df_joined
```

	year_x	month	day	dep_time	sched_dep_time	...	model	engines	seats	speed	engine
0	2013	1	1	517.0	515	...	737-824	2	149	NaN	Turbo-fan
1	2013	1	8	1435.0	1440	...	737-824	2	149	NaN	Turbo-fan
2	2013	1	9	717.0	700	...	737-824	2	149	NaN	Turbo-fan
3	2013	1	9	1143.0	1144	...	737-824	2	149	NaN	Turbo-fan
4	2013	1	13	835.0	824	...	737-824	2	149	NaN	Turbo-fan
...	...	...	...	...	...	...	...	...	...	...	...
284165	2013	9	20	1758.0	1805	...	CL-600-2C10	2	80	NaN	Turbo-fan
284166	2013	9	22	1759.0	1805	...	CL-600-2C10	2	80	NaN	Turbo-fan
284167	2013	9	23	1759.0	1805	...	CL-600-2C10	2	80	NaN	Turbo-fan
284168	2013	9	24	1751.0	1805	...	CL-600-2C10	2	80	NaN	Turbo-fan
284169	2013	9	28	712.0	720	...	737-890	2	149	NaN	Turbo-fan

284170 rows × 26 columns

- Joining two data frames results in a *wider* data frame, with more columns.
- What's the deal with `year_x`?

Overlapping Column Names

`df_flights`

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	1	1	517.0	515	...	227.0	1400	5	15	N14228
1	2013	1	1	533.0	529	...	227.0	1416	5	29	N24211
2	2013	1	1	542.0	540	...	160.0	1089	5	40	N619AA
3	2013	1	1	544.0	545	...	183.0	1576	5	45	N804JB
4	2013	1	1	554.0	600	...	116.0	762	6	0	N668DN
...	...	...	...	...	...	...	...	...	...	...	...
336771	2013	9	30	NaN	1455	...	NaN	213	14	55	NaN
336772	2013	9	30	NaN	2200	...	NaN	198	22	0	NaN
336773	2013	9	30	NaN	1210	...	NaN	764	12	10	N535MQ
336774	2013	9	30	NaN	1159	...	NaN	419	11	59	N511MQ
336775	2013	9	30	NaN	840	...	NaN	431	8	40	N839MQ

336776 rows × 18 columns

`df_planes`

	tailnum	year	type	manufacturer	model
0	N10156	2004.0	Fixed wing multi engine	EMBRAER	EMB-145XR
1	N102UW	1998.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214
2	N103US	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214
3	N104UW	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214
4	N10575	2002.0	Fixed wing multi engine	EMBRAER	EMB-145LR
...	...	...	...	...	...
3317	N997AT	2002.0	Fixed wing multi engine	BOEING	717-200
3318	N997DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS AIRCRAFT CO	MD-88
3319	N998AT	2002.0	Fixed wing multi engine	BOEING	717-200
3320	N998DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88
3321	N999DN	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88

3322 rows × 9 columns

Both data frames contain a column named `year`, but we did not join on this as a key.

By default, Pandas adds the suffixes `_x` and `_y` to overlapping column names, but this can be customized.

```
df_joined = df_flights.merge(df_planes, on="tailnum",
                             suffixes=("_flight", "_plane"))
df_joined
```

	year_flight	month	day	dep_time	sched_dep_time	...	model	engines	seats	speed	engine
0	2013	1	1	517.0	515	...	737-824	2	149	NaN	Turbo-fan
1	2013	1	8	1435.0	1440	...	737-824	2	149	NaN	Turbo-fan
2	2013	1	9	717.0	700	...	737-824	2	149	NaN	Turbo-fan

Analyzing the Joined Data

Now that we have joined the datasets, we can answer the question: which manufacturer's planes are most common?

```
df_joined["manufacturer"].value_counts()
```

```
BOEING                          82912
EMBRAER                         66068
AIRBUS                          47302
AIRBUS INDUSTRIE                40891
BOMBARDIER INC                  28272
...
```

Maybe it would be better to count *passengers* instead of flights. We don't have data on the number of passengers, but we can approximate it by the number of seats on each plane.

```
(df_joined.groupby("manufacturer")["seats"].sum().
 sort_values(ascending=False))
```

```
manufacturer
BOEING                          14418976
AIRBUS                           9624241
AIRBUS INDUSTRIE                 7522805
EMBRAER                          2803680
BOMBARDIER INC                   2239015
...
```

1 Joining Datasets on a Key

2 Keys with Multiple Columns

3 Summary

4 Reminders

Joining to Weather Data

What weather factors cause flight delays?

We need to join the flights data to weather data. Here is a dataset containing hourly weather data at each airport.

```
df_weather = pd.read_csv(data_dir + "weather.csv")
df_weather
```

	airport	year	month	day	hour	...	wind_speed	wind_gust	precip	pressure	visib
0	EWR	2013	1	1	1	...	10.35702	NaN	0.0	1012.0	10.0
1	EWR	2013	1	1	2	...	8.05546	NaN	0.0	1012.3	10.0
2	EWR	2013	1	1	3	...	11.50780	NaN	0.0	1012.5	10.0
3	EWR	2013	1	1	4	...	12.65858	NaN	0.0	1012.2	10.0
4	EWR	2013	1	1	5	...	12.65858	NaN	0.0	1011.9	10.0
...	...	...	...	...	...	...	...	...	...	...	...
26110	LGA	2013	12	30	14	...	13.80936	21.86482	0.0	1017.1	10.0
26111	LGA	2013	12	30	15	...	17.26170	21.86482	0.0	1018.8	10.0
26112	LGA	2013	12	30	16	...	14.96014	23.01560	0.0	1019.5	10.0
26113	LGA	2013	12	30	17	...	17.26170	NaN	0.0	1019.9	10.0
26114	LGA	2013	12	30	18	...	18.41248	NaN	0.0	1020.9	10.0

26115 rows × 14 columns

What is the primary key of this data set?

(airport, year, month, day, hour)

A Key with Multiple Columns

Let's start by looking at flights out of JFK. We need to join on year, month, day, and hour.

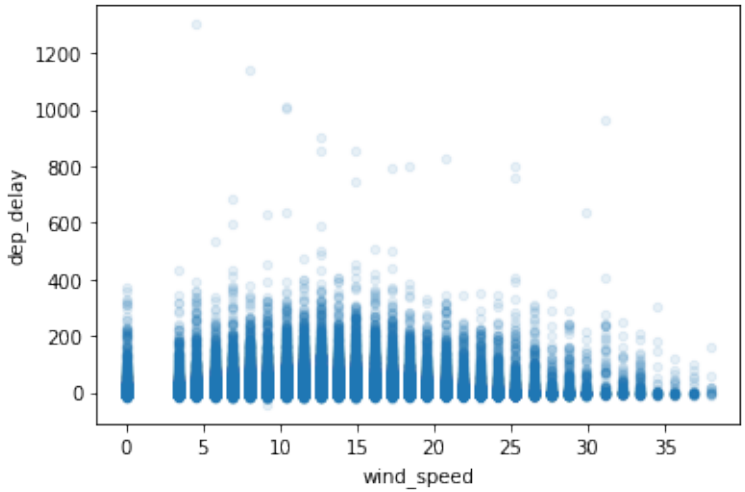
```
df_jfk = df_flights[df_flights["origin"] == "JFK"].merge(
    df_weather[df_weather["airport"] == "JFK"],
    on=("year", "month", "day", "hour"))
df_jfk
```

	year	month	day	dep_time	sched_dep_time	...	wind_speed	wind_gust	precip	pressure	visib
0	2013	1	1	542.0	540	...	14.96014	NaN	0.0	1012.1	10.0
1	2013	1	1	544.0	545	...	14.96014	NaN	0.0	1012.1	10.0
2	2013	1	1	559.0	559	...	14.96014	NaN	0.0	1012.1	10.0
3	2013	1	1	557.0	600	...	13.80936	NaN	0.0	1012.6	10.0
4	2013	1	1	558.0	600	...	13.80936	NaN	0.0	1012.6	10.0
...	...	...	...	...	...	...	...	...	...	...	...
110728	2013	9	30	2240.0	2245	...	9.20624	NaN	0.0	1016.5	10.0
110729	2013	9	30	2240.0	2250	...	9.20624	NaN	0.0	1016.5	10.0
110730	2013	9	30	2241.0	2246	...	9.20624	NaN	0.0	1016.5	10.0
110731	2013	9	30	2307.0	2255	...	9.20624	NaN	0.0	1016.5	10.0
110732	2013	9	30	2349.0	2359	...	9.20624	NaN	0.0	1016.3	10.0

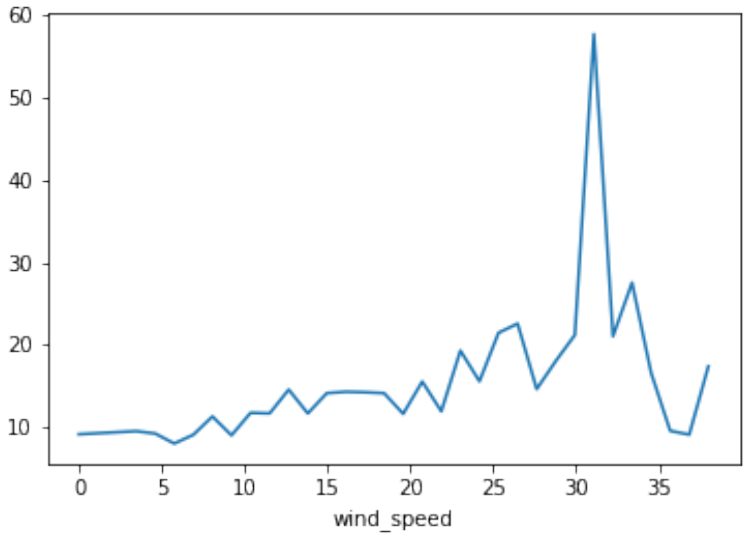
110733 rows × 28 columns

Let's see if the wind affects departure delays.

```
df_jfk.plot.scatter(x="wind_speed", y="dep_delay", alpha=0.1)
```




```
df_jfk.groupby("wind_speed")["dep_delay"].mean().plot.line()
```



Joining on Keys with Different Names

Sometimes, the keys you want to join have different names in the two datasets. This usually happens if the datasets come from different sources.

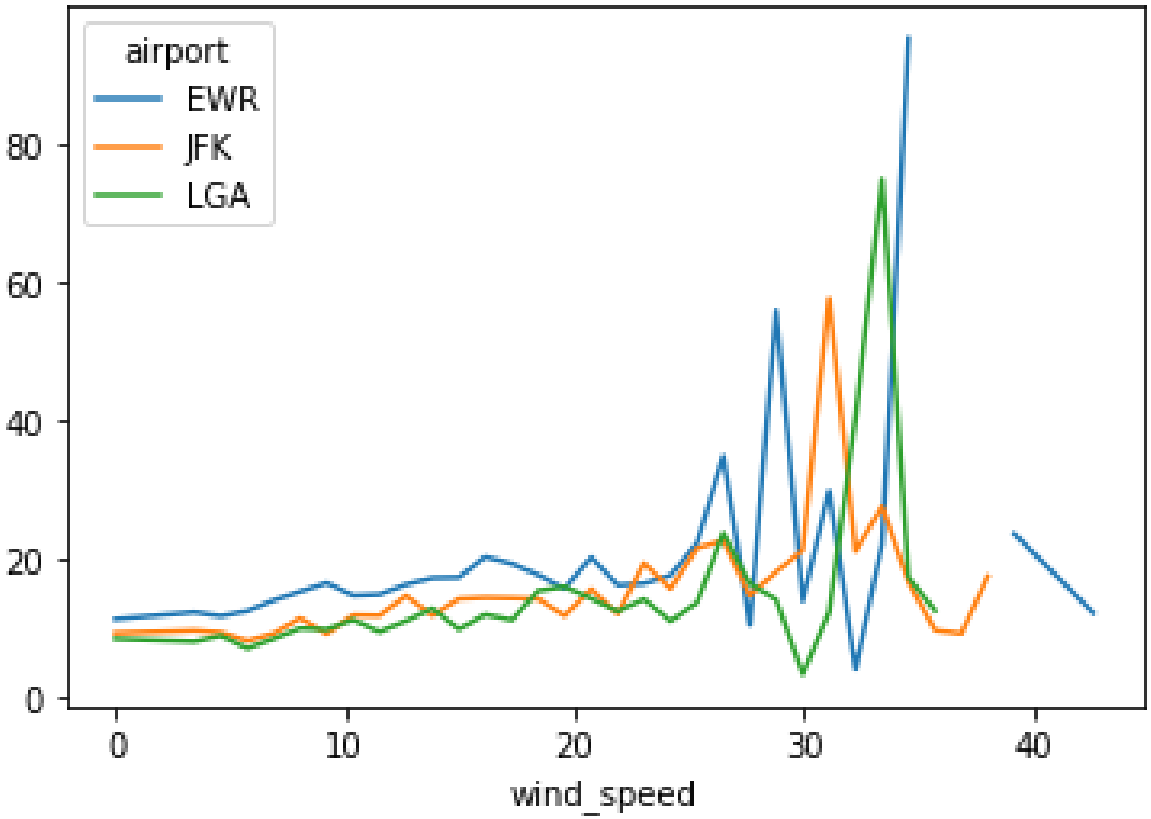
For example, if we want to join the (entire) flights data to the weather data, we need to include the airport in the key.
But the airport is called `"origin"` in `df_flights` and `"airport"` in `df_weather`.

The `.merge()` function provides `left_on=` and `right_on=` arguments for specifying different keys in the **left** and **right** data frames.

```
df_flights_weather = df_flights.merge(
    df_weather,
    left_on=("origin", "year", "month", "day", "hour"),
    right_on=("airport", "year", "month", "day", "hour"))
```

Note that both `"origin"` and `"airport"` appear as columns in the joined data frame (even though they are the same).

```
(df_flights_weather
 .groupby(["airport", "wind_speed"])["dep_delay"].mean()
 .unstack("airport")
 .plot.line())
```



1 Joining Datasets on a Key

2 Keys with Multiple Columns

3 Summary

4 Reminders

What We Have Learned Today

In Pandas, `df1.merge(df2, ...)` can be used to join two `DataFrame`s, giving you access to variables from both datasets.

- Usually, we join the *foreign key* of one dataset to the *primary key* of another.
- If the keys have the same names, we use `df1.merge(df2, on=...)`. Note that `on=` may be a single column name or a list of column names.
- If the keys have different names, we use `df1.merge(df2, left_on=..., right_on=...)`.
- Overlapping columns that are not keys will have a suffix appended, which can be customized using `df1.merge(df2, ..., suffixes=...)`.

What We Haven't Learned Today

- what happens when a key is missing from the left or right dataset
- what happens when you join using a column (or columns) that is not a primary key

We'll talk about these issues next time.

1. Joining Datasets on a Key
2. Keys with Multiple Columns
3. Summary
4. Reminders

Homework Reminders

- Don't forget to do the Colab ("Baby Names") for section tomorrow.
- Assignment 6 due Friday. **No extensions** because we need to post solutions (so that you can study for your exam).

Exam 2 Reminders

- Exam 2 is in class next Monday. Same policy as last time (1 page of handwritten notes allowed).
- The exam includes material up to Monday (hierarchical clustering). It does not include material from today or Friday.
- I have posted a practice exam. Solutions will be posted later in the week.
- We have also posted solutions to all the assignments and will post solutions to Assignments 5 and 6 before the exam.

Project Reminders

- Sign up for a final project presentation here: [link to form].
- The final project files are due on Canvas on Wednesday 3/22 at 11:59 PM.