Lecture 20
Combining Data Sets

Why Combine Data?

Sometimes, information is spread across multiple data sets.

Why Combine Data?

Sometimes, information is spread across multiple data sets.

For example, suppose we want to know which manufacturer's planes made the most flights in November 2013.

Why Combine Data?

Sometimes, information is spread across multiple data sets.

For example, suppose we want to know which manufacturer's planes made the most flights in November 2013.

One data set contains information about flights in Nov. 2013...

Why Combine Data?

Sometimes, information is spread across multiple data sets.

For example, suppose we want to know which manufacturer's planes made the most flights in November 2013.

One data set contains information about flights in Nov. 2013...

```
import pandas as pd
data_dir = "https://datasci112.stanford.edu/data/nycflights13/"
df_flights = pd.read_csv(f"{data_dir}/flights11.csv")
df_flights
```

Why Combine Data?

Sometimes, information is spread across multiple data sets.

For example, suppose we want to know which manufacturer's planes made the most flights in November 2013.

One data set contains information about flights in Nov. 2013...

```
import pandas as pd
data_dir = "https://datasci112.stanford.edu/data/nycflights13/"
df_flights = pd.read_csv(f"{data_dir}/flights11.csv")
df_flights
```

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

Fields in flights11.csv:
year — Yıl month — Ay day — Gün
dep_time — Gerçek kalkış saati
sched_dep_time —Planlanan kalkış
dep_delay — Kalkış gecikmesi (min)
arr_time — Gerçek varış
sched_arr_time — Planlanan varış
arr_delay — Varış gecikmesi(min)
carrier — Havayolu kodu
flight — Uçuş numarası
tailnum — Uçağın kuyruk numarası
origin — Kalkış havaalanı
dest — Varış havaalanı
air_time — Uçuş süresi(min)
distance — Mesafe (mil)
hour — Saat
minute — Dakika
time_hour — Tarih–saat birleşimi

Why Combine Data?

...while another contains information about planes.

Why Combine Data?

...while another contains information about planes.

```
df_planes = pd.read_csv(f"{data_dir}/planes.csv")
df_planes
```

Why Combine Data?

...while another contains information about planes.

```
df_planes = pd.read_csv(f"{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

Manufacturer info in planes.csv

flights info in the flights11.cvs table

tailnum → Uçağın kuyruk numarası (kimlik)
year → Uçağın üretim yılı
type → Uçak tipi
manufacturer → Üretici firma
model → Uçak modeli
engines → Motor sayısı
seats → Koltuk sayısı
speed → Seyir hızı
engine → Motor tipi

Why Combine Data?

...while another contains information about planes.

```
df_planes = pd.read_csv(f"{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

In order to answer the question of which manufacturer made the most flights, we have to join these two data sets together.

1. Joining on a Key
2. Joining on Multiple Keys
3. Recap

Keys

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

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

df_planes

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

3322 rows × 9 columns

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

df_planes

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

3322 rows × 9 columns

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

df_planes

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

3322 rows × 9 columns

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

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

df_planes

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

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.

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

df_planes

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

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.

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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 rows × 18 columns

df_planes

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

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 DataFrames on a key.

Joining on a Key

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

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

on="tailnum" → ortak anahtar sütun

Joining on a Key

The Pandas function .merge() can be used to join two DataFrames 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	11	1	2108.0	2056	...	EMB-145XR	2	55	NaN	Turbo-fan
1	2013	11	2	1421.0	1430	...	EMB-145XR	2	55	NaN	Turbo-fan
2	2013	11	3	1218.0	1226	...	EMB-145XR	2	55	NaN	Turbo-fan
3	2013	11	3	1725.0	1729	...	EMB-145XR	2	55	NaN	Turbo-fan
4	2013	11	4	633.0	635	...	EMB-145XR	2	55	NaN	Turbo-fan
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1649.0	1655	...	737-8H4	2	140	NaN	Turbo-fan
23296	2013	11	30	1650.0	1700	...	CL-600-2B19	2	55	NaN	Turbo-fan
23297	2013	11	30	1308.0	1315	...	CL-600-2B19	2	55	NaN	Turbo-fan
23298	2013	11	30	1858.0	1900	...	CL-600-2B19	2	55	NaN	Turbo-fan
23299	2013	11	30	2106.0	2033	...	717-200	2	100	NaN	Turbo-fan

23300 rows × 26 columns

on="tailnum" → ortak anahtar sütun

Joining on a Key

The Pandas function .merge() can be used to join two DataFrames 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	11	1	2108.0	2056	...	EMB-145XR	2	55	NaN	Turbo-fan
1	2013	11	2	1421.0	1430	...	EMB-145XR	2	55	NaN	Turbo-fan
2	2013	11	3	1218.0	1226	...	EMB-145XR	2	55	NaN	Turbo-fan
3	2013	11	3	1725.0	1729	...	EMB-145XR	2	55	NaN	Turbo-fan
4	2013	11	4	633.0	635	...	EMB-145XR	2	55	NaN	Turbo-fan
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1649.0	1655	...	737-8H4	2	140	NaN	Turbo-fan
23296	2013	11	30	1650.0	1700	...	CL-600-2B19	2	55	NaN	Turbo-fan
23297	2013	11	30	1308.0	1315	...	CL-600-2B19	2	55	NaN	Turbo-fan
23298	2013	11	30	1858.0	1900	...	CL-600-2B19	2	55	NaN	Turbo-fan
23299	2013	11	30	2106.0	2033	...	717-200	2	100	NaN	Turbo-fan

23300 rows x 26 columns

- Joining two data frames results in a *wider* data frame, with more columns.

Joining on a Key

The Pandas function `.merge()` can be used to join two DataFrames 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	11	1	2108.0	2056	...	EMB-145XR	2	55	NaN	Turbo-fan
1	2013	11	2	1421.0	1430	...	EMB-145XR	2	55	NaN	Turbo-fan
2	2013	11	3	1218.0	1226	...	EMB-145XR	2	55	NaN	Turbo-fan
3	2013	11	3	1725.0	1729	...	EMB-145XR	2	55	NaN	Turbo-fan
4	2013	11	4	633.0	635	...	EMB-145XR	2	55	NaN	Turbo-fan
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1649.0	1655	...	737-8H4	2	140	NaN	Turbo-fan
23296	2013	11	30	1650.0	1700	...	CL-600-2B19	2	55	NaN	Turbo-fan
23297	2013	11	30	1308.0	1315	...	CL-600-2B19	2	55	NaN	Turbo-fan
23298	2013	11	30	1858.0	1900	...	CL-600-2B19	2	55	NaN	Turbo-fan
23299	2013	11	30	2106.0	2033	...	717-200	2	100	NaN	Turbo-fan

23300 rows x 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	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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

Overlapping Column Names

df_flights

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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

Overlapping Column Names

df_flights

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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.

Overlapping Column Names

df_flights

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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.

Overlapping Column Names

df_flights

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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.columns
```

Overlapping Column Names

df_flights

	year	month	day	dep_time	sched_dep_time	...	air_time	distance	hour	minute	tailnum
0	2013	11	1	2108.0	2056	...	179.0	1167	20	56	N10156
1	2013	11	1	1154.0	1200	...	102.0	541	12	0	N102UW
2	2013	11	1	854.0	829	...	162.0	946	8	29	N10575
3	2013	11	1	1643.0	1505	...	104.0	594	15	5	N10575
4	2013	11	1	603.0	600	...	53.0	282	6	0	N11109
...	...	...	...	...	...	...	...	...	...	...	...
23295	2013	11	30	1337.0	1340	...	153.0	1076	13	40	N994DL
23296	2013	11	30	802.0	807	...	140.0	1069	8	7	N995DL
23297	2013	11	30	1544.0	1550	...	150.0	1069	15	50	N995DL
23298	2013	11	30	850.0	900	...	117.0	762	9	0	N996DL
23299	2013	11	30	1959.0	2000	...	115.0	762	20	0	N999DN

23300 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.columns
```

```
Index(['year_flight', 'month', 'day', 'dep_time', 'sched_dep_time',
       'dep_delay', 'arr_time', 'sched_arr_time', 'arr_delay', 'carrier',
       'flight', 'origin', 'dest', 'air_time', 'distance', 'hour', 'minute',
       'tailnum', 'year_plane', 'type', 'manufacturer', 'model', 'engines',
       'seats', 'speed', 'engine'])
```

Analyzing the Joined Data

Now that we have joined the two data sets, we can answer the question: which manufacturer’s planes made the most flights?

Analyzing the Joined Data

Now that we have joined the two data sets, we can answer the question: which manufacturer's planes made the most flights?

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

```
BOEING                6557
EMBRAER               5175
AIRBUS                3954
AIRBUS INDUSTRIE      3456
BOMBARDIER INC        2632
                      ...
MARZ BARRY               1
AVIAT AIRCRAFT INC       1
PAIR MIKE E              1
LEBLANC GLENN T          1
STEWART MACO             1
Name: manufacturer, Length: 29, dtype: int64
```

Joining to Weather Data

What weather factors cause flight delays?

Joining to Weather Data

What weather factors cause flight delays?

To answer this question, we need to join the flights data to weather data. Here is a data set containing hourly weather data at each airport in 2013.

Joining to Weather Data

What weather factors cause flight delays?

To answer this question, we need to join the flights data to weather data. Here is a data set containing hourly weather data at each airport in 2013.

```
df_weather = pd.read_csv(f"{data_dir}/weather.csv")
df_weather
```

origin → Havaalanı (EWR, JFK, LGA)
year → Yıl month → Ay day → Gün hour → Saat
temp → Sıcaklık (Fahrenheit)
dewp → Çiy noktası
humid → Nem
wind_dir → Rüzgâr yönü
wind_speed → Rüzgâr hızı
wind_gust → Rüzgâr hamlesi
precip → Yağış miktarı
pressure → Hava basıncı
visib → Görüş mesafesi
time_hour → Tarih–saat (timestamp)

Joining to Weather Data

What weather factors cause flight delays?

To answer this question, we need to join the flights data to weather data. Here is a data set containing hourly weather data at each airport in 2013.

```
df_weather = pd.read_csv(f"{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

Joining to Weather Data

What weather factors cause flight delays?

To answer this question, we need to join the flights data to weather data. Here is a data set containing hourly weather data at each airport in 2013.

```
df_weather = pd.read_csv(f"{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?

Joining to Weather Data

What weather factors cause flight delays?

To answer this question, we need to join the flights data to weather data. Here is a data set containing hourly weather data at each airport in 2013.

```
df_weather = pd.read_csv(f"{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

Weather is measured:
- at a specific airport
- at a specific time

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 to the weather data on year, month, day, and hour.

A Key with Multiple Columns

Let’s start by looking at flights out of JFK. We need to join to the weather data 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
```

df_flights[df_flights["origin"] == "JFK"]
Sadece JFK havaalanından kalkan uçuşlar filtrelenir.

df_weather[df_weather["airport"] == "JFK"]
Sadece JFK havaalanına ait hava durumu kayıtları alınır.

on=("year","month","day","hour")
Uçuş ve hava durumu aynı zaman diliminde eşleştirilir.

df_jfk
JFK’den kalkan uçuşlar
O saatteki JFK hava koşulları eklenmiş halde

A Key with Multiple Columns

Let's start by looking at flights out of JFK. We need to join to the weather data 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	11	1	1154.0	1200	...	18.41248	NaN	0.0	1002.8	10.0
1	2013	11	1	1351.0	1229	...	18.41248	NaN	0.0	1002.8	10.0
2	2013	11	1	1203.0	1200	...	18.41248	NaN	0.0	1002.8	10.0
3	2013	11	1	1159.0	1200	...	18.41248	NaN	0.0	1002.8	10.0
4	2013	11	1	1246.0	1200	...	18.41248	NaN	0.0	1002.8	10.0
...	...	...	...	...	...	...	...	...	...	...	...
7396	2013	11	30	1055.0	1100	...	5.75390	NaN	0.0	1039.8	10.0
7397	2013	11	30	2351.0	2359	...	4.60312	NaN	0.0	1028.9	10.0
7398	2013	11	30	2354.0	2359	...	4.60312	NaN	0.0	1028.9	10.0
7399	2013	11	30	11.0	2359	...	4.60312	NaN	0.0	1028.9	10.0
7400	2013	11	30	544.0	540	...	8.05546	NaN	0.0	1041.7	10.0

7401 rows × 28 columns

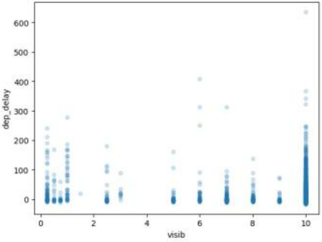
Let’s see how visibility affects departure delays.

Let's see how visibility affects departure delays.

```
df_jfk.plot.scatter(x="visib", y="dep_delay", alpha=0.2)
```

Let's see how visibility affects departure delays.

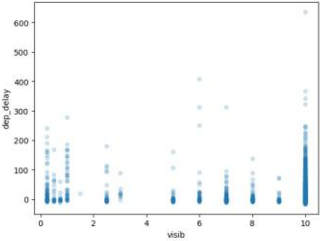
```
df_jfk.plot.scatter(x="visib", y="dep_delay", alpha=0.2)
```



x ekseni: görüş mesafesi
y ekseni: kalkış gecikmesi

Let's see how visibility affects departure delays.

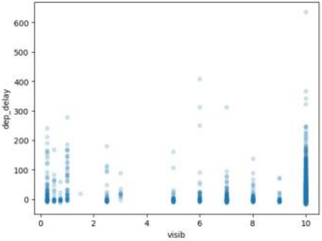
```
df_jfk.plot.scatter(x="visib", y="dep_delay", alpha=0.2)
```




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

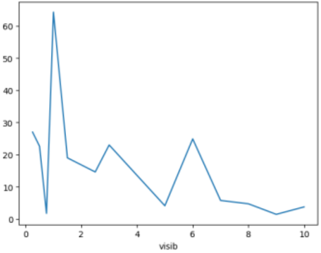
Let’s see how visibility affects departure delays.

```
df_jfk.plot.scatter(x="visib", y="dep_delay", alpha=0.2)
```



x ekseni: görüş mesafesi
y ekseni: kalkış gecikmesi

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



groupby("visib") → groups the data by visibility

["dep_delay"].mean() → computes the mean departure delay for each visibility level

plot.line() → displays the result as a line plot

Purpose: to see how average departure delay changes with visibility.

e.g. visib= about 1 mile delay about 60mins

Joining on Keys with Different Names

Sometimes, the join keys have different names in the two data sets. This happens if the data sets come from different sources.

Joining on Keys with Different Names

Sometimes, the join keys have different names in the two data sets. This happens if the data sets come from different sources.

For example, if we want to join the (entire) flights data to the weather data, we would need to include the airport in the key.

Joining on Keys with Different Names

Sometimes, the join keys have different names in the two data sets. This happens if the data sets come from different sources.

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

Joining on Keys with Different Names

Sometimes, the join keys have different names in the two data sets. This happens if the data sets come from different sources.

For example, if we want to join the (entire) flights data to the weather data, we would 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 column names in the **left** and **right** data frames.

Joining on Keys with Different Names

Sometimes, the join keys have different names in the two data sets. This happens if the data sets come from different sources.

For example, if we want to join the (entire) flights data to the weather data, we would 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 column names 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"))
```

Joining on Keys with Different Names

Let’s complete this analysis in a Colab.

What We Have Learned Today

In Pandas, `df_left.merge(df_right, ...)` can be used to join two DataFrames, giving you access to variables from both data sets.

What We Have Learned Today

- In Pandas, `df_left.merge(df_right, ...)` can be used to join two
- DataFrames, giving you access to variables from both data sets.
    - Usually, we join the *foreign key* of one data set to the *primary key* of another.

What We Have Learned Today

- In Pandas, `df_left.merge(df_right, ...)` can be used to join two
- DataFrames, giving you access to variables from both data sets.
  - Usually, we join the *foreign key* of one data set to the *primary key* of another.
  - If the keys have the same names, we use `df_left.merge(df_right, on=...)`. Note that `on=` may be a single column name or a list of column names.

What We Have Learned Today

- In Pandas, `df_left.merge(df_right, ...)` can be used to join two
- DataFrames, giving you access to variables from both data sets.
  - Usually, we join the *foreign key* of one data set to the *primary key* of another.
  - If the keys have the same names, we use `df_left.merge(df_right, on=...)`. Note that `on=` may be a single column name or a list of column names.
  - If the keys have different names, we use
  - `df_left.merge(df_right, left_on=..., right_on=...)`.

What We Have Learned Today

- In Pandas, `df_left.merge(df_right, ...)` can be used to join two
- DataFrames, giving you access to variables from both data sets.
    - Usually, we join the *foreign key* of one data set to the *primary key* of another.
    - If the keys have the same names, we use `df_left.merge(df_right, on=...)`. Note that `on=` may be a single column name or a list of column names.
    - If the keys have different names, we use
        - `df_left.merge(df_right, left_on=..., right_on=...)`.
    - Overlapping columns that are not keys will have a suffix appended, which can be customized using `df_left.merge(df_right, ..., suffixes=...)`.

What We Haven't Learned Today

What We Haven't Learned Today

- what happens when a key is missing from the left or right data set

What We Haven't Learned Today

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

What We Haven't Learned Today

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

We'll address these issues in the next lecture.