


Start coding or [generate](#) with AI.

Load data

```
import pandas as pd
df = pd.read_csv('simulated_flights.csv')
df['scheduled_departure'] = pd.to_datetime(df['scheduled_departure'])
df.head()
```



	flight_number	origin	destination	scheduled_departure	status	delay_min
0	DXB0001	DEL	DXB	2024-04-16 16:03:00	Diverted	
1	DXB0002	JFK	DXB	2024-04-17 05:05:00	Cancelled	
2	DXB0003	MUM	DXB	2024-04-17 06:27:00	On Time	
3	DXB0004	JFK	DXB	2024-04-16 20:10:00	On Time	
4	DXB0005	JFK	DXB	2024-04-16 11:11:00	On Time	


Next steps:

[Generate code with df](#)

 [View recommended plots](#)

[New interactive sheet](#)

Question: What are the data types of each column, and how many entries do we have?
df.info()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 6 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   flight_number                        1500 non-null   object
1   origin                              1500 non-null   object
2   destination                          1500 non-null   object
3   scheduled_departure                  1500 non-null   datetime64[ns]
4   status                              1500 non-null   object
5   delay_minutes                       1500 non-null   int64
dtypes: datetime64[ns](1), int64(1), object(4)
memory usage: 70.4+ KB
```

```
affected_flights = df[df['status'].isin(['Delayed', 'Cancelled', 'Diverted'])]
num_affected = len(affected_flights)
print(f"Number of affected flights: {num_affected}")
```



```
Number of affected flights: 757
```

```
# Question: What is the breakdown of the different types of disruptions (how many of each)?
disruption_counts = affected_flights['status'].value_counts()
print("Breakdown of disruptions:")
print(disruption_counts)
```

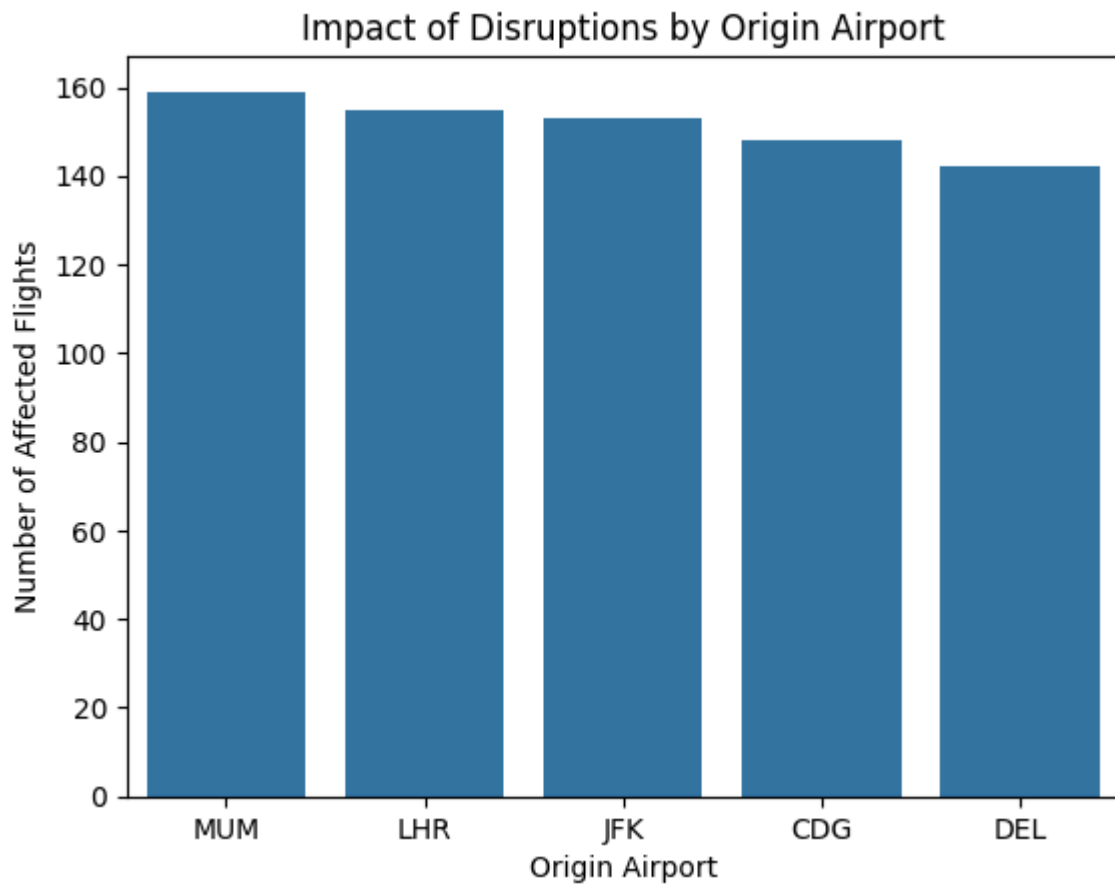
```
⇒ Breakdown of disruptions:
status
Delayed      389
Diverted     216
Cancelled    152
Name: count, dtype: int64
```

```
# Question: Which origin airports were most affected by the disruptions?
origin_impact = affected_flights['origin'].value_counts()
print("Impact by origin airport:")
print(origin_impact)
```

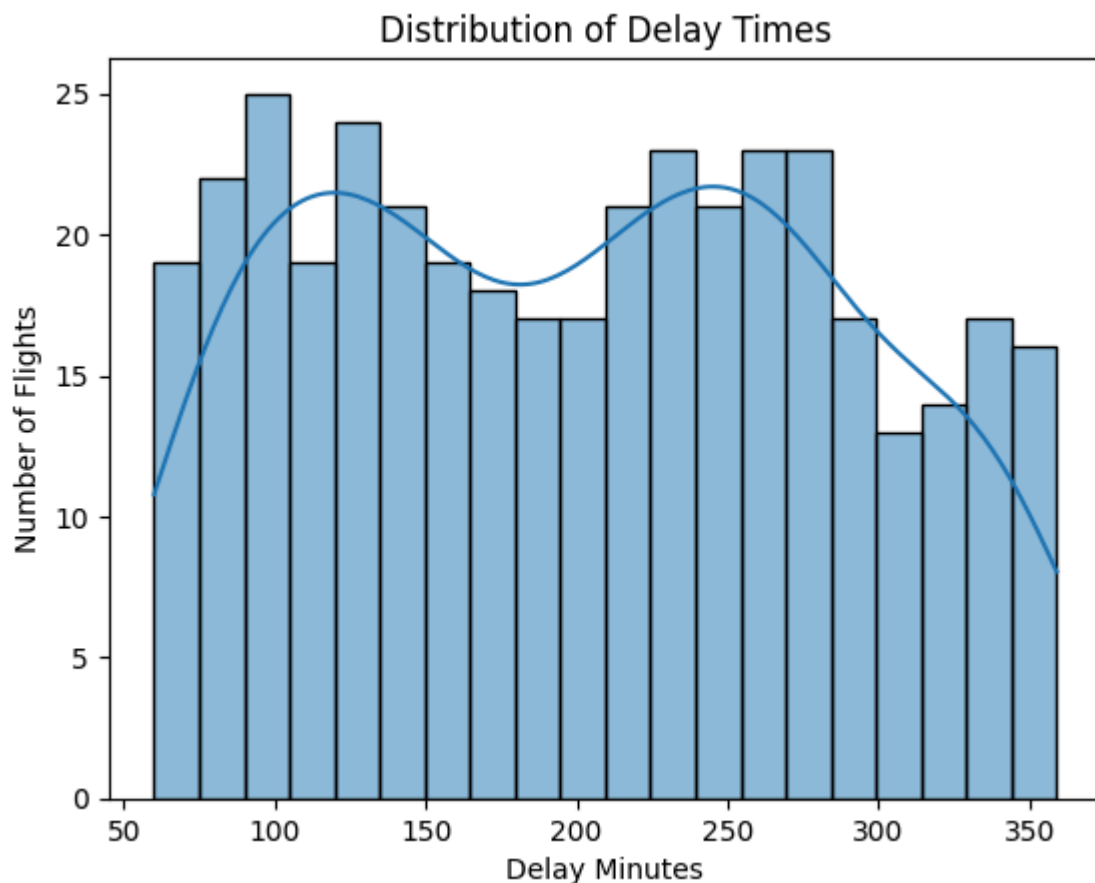
```
⇒ Impact by origin airport:
origin
MUM      159
LHR      155
JFK      153
CDG      148
DEL      142
Name: count, dtype: int64
```

```
# Question: Let's visualize the impact by origin airport using a bar chart.
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.barplot(x=origin_impact.index, y=origin_impact.values)
plt.xlabel("Origin Airport")
plt.ylabel("Number of Affected Flights")
plt.title("Impact of Disruptions by Origin Airport")
plt.show()
```



```
# Question: What is the distribution of delay times for the delayed flights?
delayed_flights = df[df['status'] == 'Delayed']
sns.histplot(delayed_flights['delay_minutes'], bins=20, kde=True) #kde for kernel
plt.xlabel("Delay Minutes")
plt.ylabel("Number of Flights")
plt.title("Distribution of Delay Times")
plt.show()
```



```
# Question: How many flights were affected during the simulated storm period (12
storm_flights = df[df['scheduled_departure'].dt.hour.between(12, 18)]
storm_affected = storm_flights[storm_flights['status'].isin(['Delayed', 'Cancelled'])]
print(f"Number of affected flights during storm: {len(storm_affected)}")
print(storm_affected['status'].value_counts())
```



```
Number of affected flights during storm: 456
status
Delayed      238
Diverted     112
Cancelled    106
Name: count, dtype: int64
```

```
# Question: Which flights need to be rescheduled (Cancelled or Diverted)?
rescheduleable_flights = df[df['status'].isin(['Cancelled', 'Diverted'])].copy()
rescheduleable_flights
```



	flight_number	origin	destination	scheduled_departure	status	delay_r
0	DXB0001	DEL	DXB	2024-04-16 16:03:00	Diverted	
1	DXB0002	JFK	DXB	2024-04-17 05:05:00	Cancelled	
13	DXB0014	CDG	DXB	2024-04-17 03:03:00	Diverted	
19	DXB0020	DEL	DXB	2024-04-16 12:44:00	Diverted	
25	DXB0026	MUM	DXB	2024-04-16 15:39:00	Diverted	

Next
steps:

code

rescheduleable_flights

recommended

New interactive sheet

1489 DXB1490 MUM DXB 2024-04-16 20:07:00 Cancelled

```
# Question: We want to reschedule these flights within the next 6 weeks. How do w
start_date = pd.to_datetime('2024-04-17')
end_date = start_date + pd.Timedelta(weeks=6)
reschedule_dates = pd.date_range(start=start_date, end=end_date, freq='7D')
print(f"Rescheduling dates: {reschedule_dates}")
```



```
Rescheduling dates: DatetimeIndex(['2024-04-17', '2024-04-24', '2024-05-01',
                                     '2024-05-15', '2024-05-22', '2024-05-29'],
                                     dtype='datetime64[ns]', freq='7D')
```

```
# Question: Let's create a list of rescheduleable_flights
num_flights = len(rescheduleable_flights)
num_dates = len(reschedule_dates)
rescheduleable_flights['rescheduled_departure'] = [reschedule_dates[i % num_dates]
rescheduleable_flights['status'] = 'Rescheduled'
```

```
#remove the old flights
df = df[~df['status'].isin(['Cancelled', 'Diverted'])]
df = pd.concat([df, rescheduleable_flights], ignore_index=True)
```

```
# List rescheduled days
df['rescheduled_day'] = df[df['status'] == 'Rescheduled']['rescheduled_departure']
print(df['rescheduled_day'].value_counts())
```



```
rescheduled_day
2024-04-17    53
2024-04-24    53
2024-05-01    53
2024-05-08    53
2024-05-15    52
2024-05-22    52
2024-05-29    52
Name: count, dtype: int64
```

```
# Question: After rescheduling, how many flights are still marked as Cancelled or I
remaining_flights = df[df['status'].isin(['Cancelled', 'Diverted'])]
num_remaining = len(remaining_flights)
```