

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

```
1 df = pd.read_csv('/content/airlines_flights_data.csv')
2 df
```

			index	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left
0	0	SpiceJet	SG-8709		Delhi		Evening	zero	Night	Mumbai	Economy	2.17	
1	1	SpiceJet	SG-8157		Delhi		Early_Morning	zero	Morning	Mumbai	Economy	2.33	
2	2	AirAsia	I5-764		Delhi		Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	
3	3	Vistara	UK-995		Delhi		Morning	zero	Afternoon	Mumbai	Economy	2.25	
4	4	Vistara	UK-963		Delhi		Morning	zero	Morning	Mumbai	Economy	2.33	
...	...	...	...		...		...	...	...	...	...	...	.
300148	300148	Vistara	UK-822		Chennai		Morning	one	Evening	Hyderabad	Business	10.08	4
300149	300149	Vistara	UK-826		Chennai		Afternoon	one	Night	Hyderabad	Business	10.42	4
300150	300150	Vistara	UK-832		Chennai		Early_Morning	one	Night	Hyderabad	Business	13.83	4
300151	300151	Vistara	UK-828		Chennai		Early_Morning	one	Evening	Hyderabad	Business	10.00	4
300152	300152	Vistara	UK-822		Chennai		Morning	one	Evening	Hyderabad	Business	10.08	4

300153 rows × 14 columns

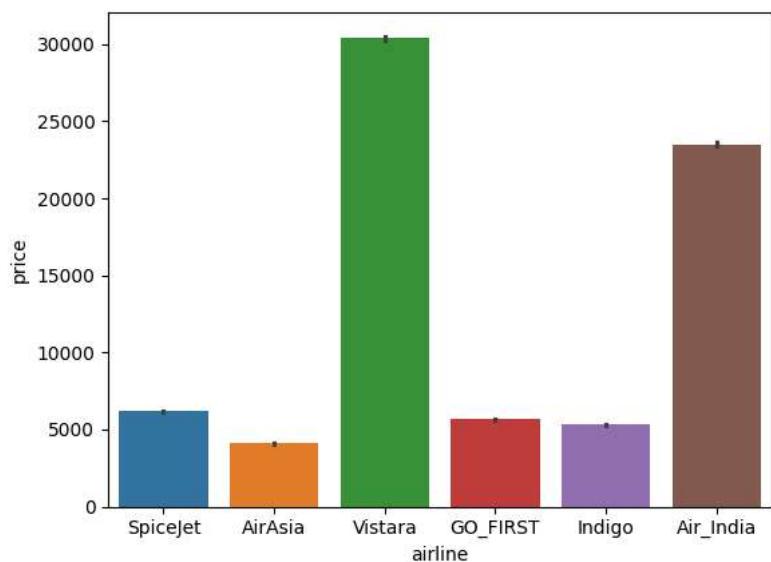
```
1 df = df.drop(['Unnamed: 12','Unnamed: 13'],axis = 1)
2 df
```

			index	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left
0	0	SpiceJet	SG-8709		Delhi		Evening	zero	Night	Mumbai	Economy	2.17	
1	1	SpiceJet	SG-8157		Delhi		Early_Morning	zero	Morning	Mumbai	Economy	2.33	
2	2	AirAsia	I5-764		Delhi		Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	
3	3	Vistara	UK-995		Delhi		Morning	zero	Afternoon	Mumbai	Economy	2.25	
4	4	Vistara	UK-963		Delhi		Morning	zero	Morning	Mumbai	Economy	2.33	
...	...	...	...		...		...	...	...	...	...	...	.
300148	300148	Vistara	UK-822		Chennai		Morning	one	Evening	Hyderabad	Business	10.08	4
300149	300149	Vistara	UK-826		Chennai		Afternoon	one	Night	Hyderabad	Business	10.42	4
			I1K-										

### Price Distribution across Airlines

```
1 sns.barplot(data = df,x = 'airline',y = 'price',hue = 'airline')
```

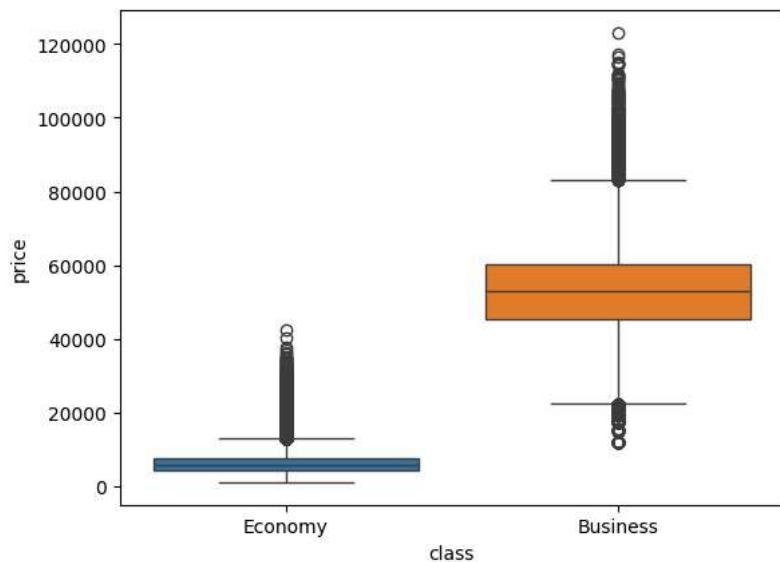
<Axes: xlabel='airline', ylabel='price'>



### Average Price: Economy vs. Business Class

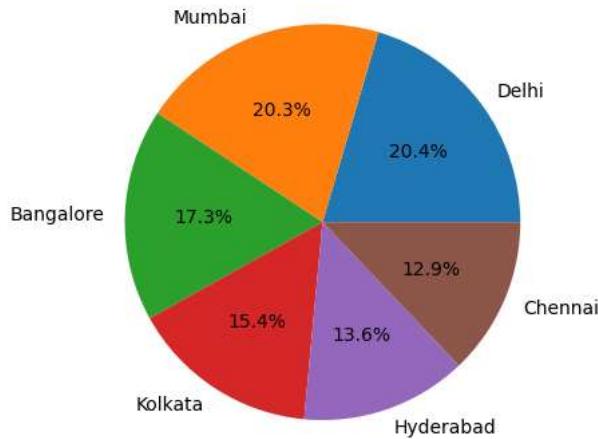
```
1 sns.boxplot(data = df,x = 'class',y = 'price',hue = 'class')
```

<Axes: xlabel='class', ylabel='price'>



### Busiest Source & Destination Cities

```
1 count1 = df['source_city'].value_counts()
2 labels = count1.index
3 category = count1.values
4 plt.pie(category,labels = labels,autopct = '%1.1f%')
5 plt.show()
```

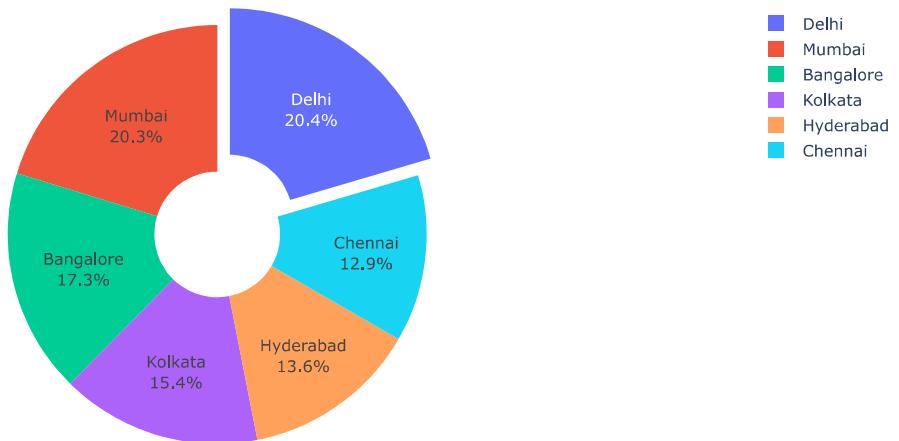


```

1 import plotly.graph_objects as go
2 a = df['source_city'].value_counts()
3 labels = a.index
4 value = a.values
5 fig = go.Figure(data=[go.Pie(labels=labels,values=a.values,hole=0.3)])
6 fig.update_traces(textinfo='percent+label',pull=[0.1,0])
7 fig.update_layout(title_text='Source Donut Chart')
8 fig.show()

```

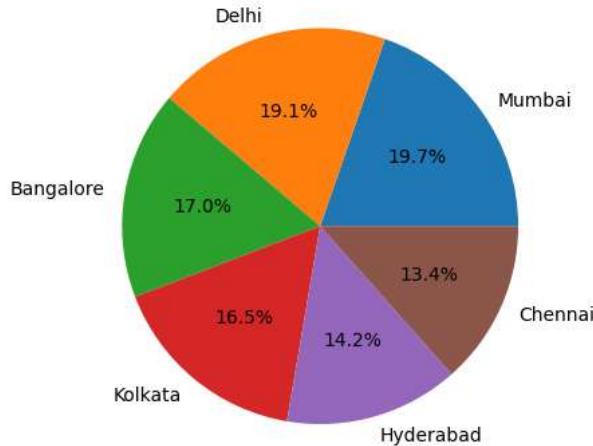
Source Donut Chart



```

1 count2 = df['destination_city'].value_counts()
2 labels = count2.index
3 catogery = count2.values
4 plt.pie(catogery,labels = labels,autopct='%1.1f%%')
5 plt.show()

```

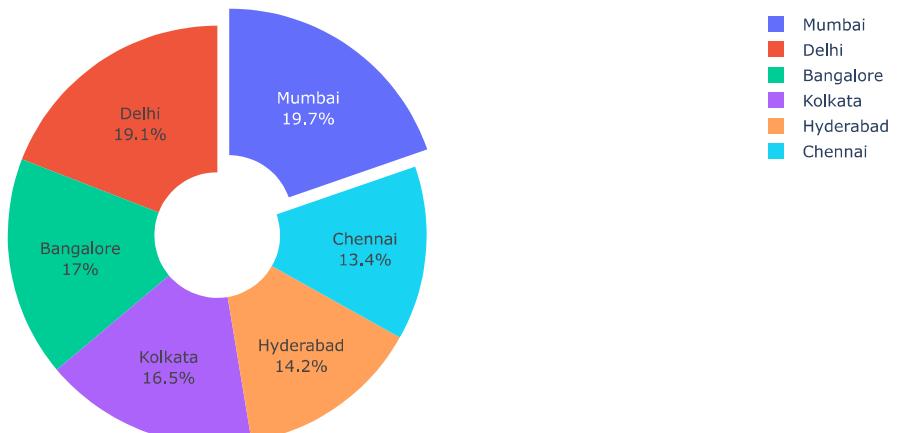


```

1 a = df['destination_city'].value_counts()
2 labels = a.index
3 value = a.values
4 fig = go.Figure(data=[go.Pie(labels=labels,values=a.values, hole=0.3)])
5 fig.update_traces(textinfo='percent+label', pull=[0.1,0])
6 fig.update_layout(title_text='Source Donut Chart')
7 fig.show()

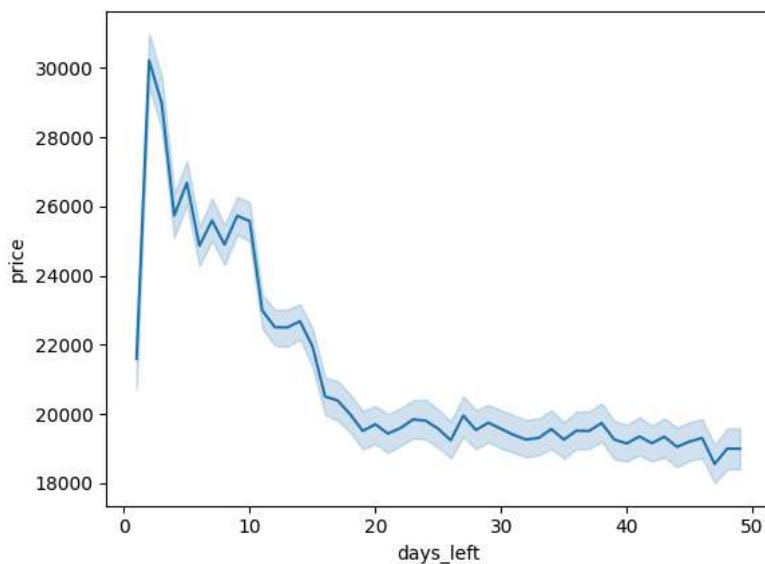
```

Source Donut Chart

**Days Left vs. Ticket Price Trend**

```
1 sns.lineplot(data = df,x = 'days_left',y = 'price')
```

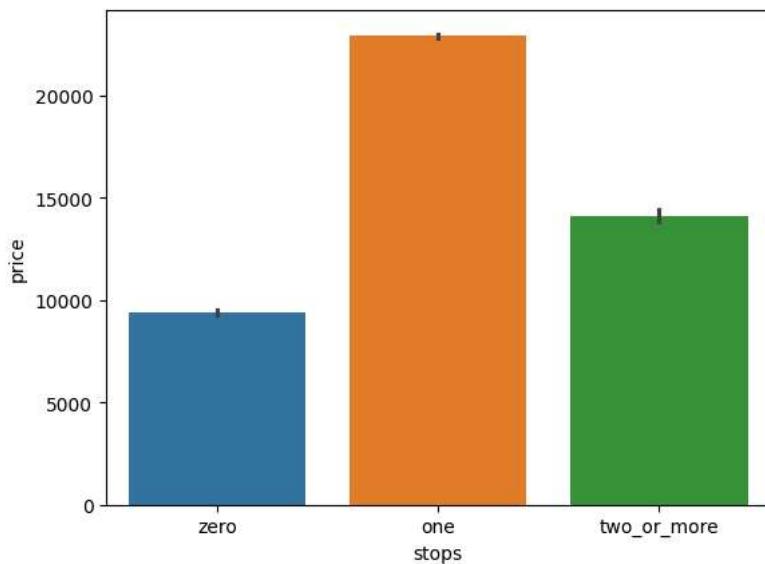
```
<Axes: xlabel='days_left', ylabel='price'>
```



### Stops vs. Price Comparison

```
1 sns.barplot(data = df,x = 'stops',y = 'price',hue = 'stops')
```

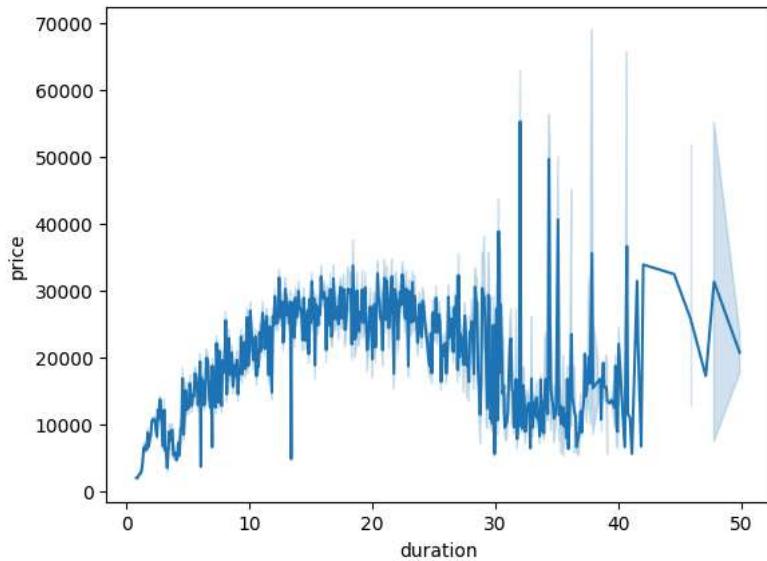
```
<Axes: xlabel='stops', ylabel='price'>
```



### Duration Impact on Ticket Pricing

```
1 sns.lineplot(data = df,x = 'duration',y = 'price')
```

```
<Axes: xlabel='duration', ylabel='price'>
```



```
1 avg = df.groupby('duration')['price'].mean()  
2 avg
```

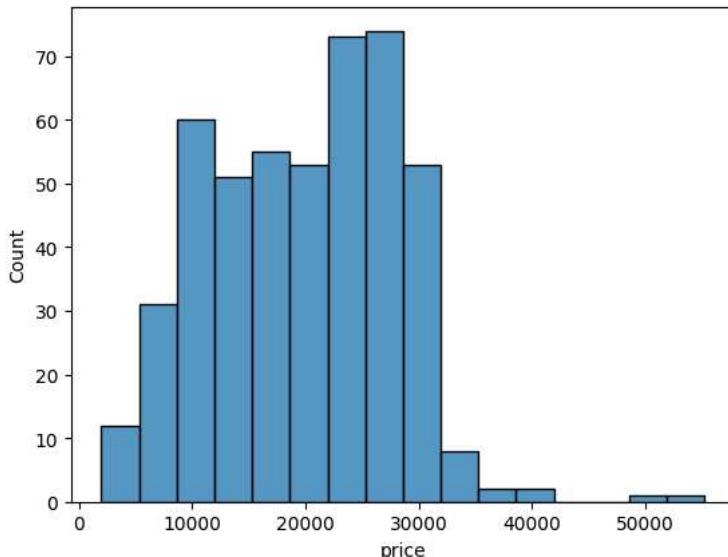
```
      price  
duration  
0.83    1973.557692  
0.92    2003.538462  
1.00    2266.063660  
1.08    2589.311031  
1.17    2632.209375  
...       ...  
44.50   32439.000000  
45.83   25851.333333  
47.08   17223.000000  
47.75   31291.500000  
49.83   20714.500000
```

476 rows × 1 columns

**dtype:** float64

```
1 sns.histplot(data = avg)
```

```
<Axes: xlabel='price', ylabel='Count'>
```



```
1 import sqlite3
```

```
1 conn = sqlite3.connect("airflight.db")
```

```
1 cursor = conn.cursor()
```

```
1 df.to_sql('airflight',conn,if_exists = 'replace',index=True)
```

```
300153
```

```
1 from tabulate import tabulate
```

```
1 #1. Find the average ticket price per airline.
2 cursor.execute('''SELECT airline, AVG(price) FROM airflight GROUP BY airline
3 ''')
4 rows = cursor.fetchall()
5 columns = [desc[0] for desc in cursor.description]
6 print(tabulate(rows,headers = columns,tablefmt='grid'))
```

airline	AVG(price)
AirAsia	4091.07
Air_India	23507
GO_FIRST	5652.01
Indigo	5324.22
SpiceJet	6179.28
Vistara	30396.5

```
1 #2. List the top 5 most expensive routes (source to destination).
```

```
2 cursor.execute("SELECT source_city,destination_city,price FROM airflight ORDER BY price DESC LIMIT 5")
3 rows = cursor.fetchall()
4 columns = [desc[0] for desc in cursor.description]
5 print(tabulate(rows,headers = columns,tablefmt='grid'))
```

source_city	destination_city	price
Kolkata	Delhi	123071
Delhi	Kolkata	117307
Delhi	Kolkata	116562

Hyderabad	Mumbai	115211
Kolkata	Hyderabad	114705

```
1 #3. Which airline offers the lowest average ticket price for Business class?
```

```
2 cursor.execute('''SELECT airline,class,AVG(price) as avg_price
3 FROM airflight
4 WHERE class = 'Business'
5 ORDER BY avg_price DESC
6 LIMIT 1''')
7 rows = cursor.fetchall()
8 columns = [desc[0] for desc in cursor.description]
9 print(tabulate(rows,headers = columns,tablefmt='grid'))
```

airline	class	avg_price
Air_India	Business	52540.1

```
1 #4. Find routes with more than 50 flights and their average ticket prices.
```

```
2 cursor.execute('''SELECT source_city,destination_city,AVG(price) as price,COUNT(flight) as flight_count
3 FROM airflight
4 GROUP BY flight
5 HAVING flight_count > 50
6 ORDER BY flight_count DESC
7 ;''')
8 rows = cursor.fetchall()
9 columns = [desc[0] for desc in cursor.description]
10 print(tabulate(rows,headers = columns,tablefmt='grid'))
```

source_city	destination_city	price	flight_count
Delhi	Mumbai	5284.84	38415
Delhi	Mumbai	32726.9	3235
Kolkata	Delhi	35466.2	2741
Kolkata	Delhi	32144.2	2650
Chennai	Delhi	33194.1	2542
Chennai	Delhi	27586.7	2468
Chennai	Delhi	28283.5	2440
Hyderabad	Delhi	26982.6	2423
Chennai	Delhi	31499.7	2404
Chennai	Delhi	26276.5	2335
Hyderabad	Delhi	31690.9	2329
Hyderabad	Delhi	25993.8	2307
Hyderabad	Delhi	27126.6	2285
Chennai	Delhi	33914.1	2255
Hyderabad	Delhi	30987.6	2204
Hyderabad	Delhi	34373.8	2199
Chennai	Delhi	29435.5	2156
Kolkata	Delhi	31244.9	2145
Kolkata	Delhi	31781.9	2140
Kolkata	Delhi	30966.2	2136
Mumbai	Delhi	24767.5	2016
Kolkata	Delhi	32345.8	2015
Kolkata	Delhi	28284.7	1833

Bangalore	Delhi	27117.5	1795
Bangalore	Delhi	36221.9	1761
Hyderabad	Delhi	32095.8	1730
Bangalore	Delhi	37141.3	1695
Chennai	Delhi	38812.0	1600

```

1 #5. Compare average price difference between Economy and Business class per airline.
2 cursor.execute('''SELECT airline,
3                         AVG(CASE WHEN class = 'Economy' THEN price END) AS avg_economy,
4                         AVG(CASE WHEN class = 'Business' THEN price END) AS avg_business,
5                         (AVG(CASE WHEN class = 'Business' THEN price END) -
6                         AVG(CASE WHEN class = 'Economy' THEN price END)) AS price_diff
7 FROM airflight
8 GROUP BY airline
9 ORDER BY price_diff
10 ''')
11 rows = cursor.fetchall()
12 columns = [desc[0] for desc in cursor.description]
13 print(tabulate(rows,headers = columns,tablefmt='grid'))

```

airline	avg_economy	avg_business	price_diff
AirAsia	4091.07		
GO_FIRST	5652.01		
Indigo	5324.22		
SpiceJet	6179.28		
Air_India	7313.68	47131	39817.4
Vistara	7806.94	55477	47670.1

```

1 #6. Which cities have the most incoming flights?
2 cursor.execute("""SELECT destination_city,COUNT(*) as no_of_planes
3 FROM airflight
4 GROUP BY destination_city
5 ORDER BY no_of_planes DESC
6 LIMIT 1""")
7 rows = cursor.fetchall()
8 columns = [desc[0] for desc in cursor.description]
9 print(tabulate(rows,headers = columns,tablefmt='grid'))

```

destination_city	no_of_planes
Mumbai	59097

```

1 #7. Which cities have the most outgoing flights?
2 cursor.execute('''SELECT source_city,COUNT(source_city) AS no_of_flights FROM airflight
3 GROUP BY source_city
4 ORDER BY no_of_flights DESC
5 LIMIT 1''')
6 rows = cursor.fetchall()
7 columns = [desc[0] for desc in cursor.description]
8 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

source_city	no_of_flights
Delhi	61343

1 Start coding or generate with AI.

```

1 #8. Find the busiest route (most number of flights).
2 cursor.execute('''SELECT source_city,destination_city,COUNT(flight) AS no_of_flights FROM airflight GROUP BY destination_ci
3 rows = cursor.fetchall()
4 columns = [desc[0] for desc in cursor.description]
5 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

source_city	destination_city	no_of_flights
Delhi	Mumbai	15289

```

1 #9. List the top 3 airlines with cheapest average prices for each route.
2 cursor.execute('''SELECT source_city,destination_city,airline,avg_price
3 FROM(
4   SELECT source_city,destination_city,airline,AVG(price) AS avg_price,
5   ROW_NUMBER()
6   OVER(PARTITION BY source_city,destination_city ORDER BY AVG(price)) as rk
7   FROM airflight
8   GROUP BY source_city,destination_city,airline)AS ranked
9   WHERE rk<=3 ''')
10 rows = cursor.fetchall()
11 columns = [desc[0] for desc in cursor.description]
12 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

source_city	destination_city	airline	avg_price
Bangalore	Chennai	AirAsia	2073.04
Bangalore	Chennai	Indigo	2363.33
Bangalore	Chennai	SpiceJet	2613.31
Bangalore	Delhi	AirAsia	4887.09
Bangalore	Delhi	GO_FIRST	5524.7
Bangalore	Delhi	Indigo	6273.5
Bangalore	Hyderabad	AirAsia	2931.49
Bangalore	Hyderabad	Indigo	3220.66
Bangalore	Hyderabad	SpiceJet	4338
Bangalore	Kolkata	AirAsia	4443.47
Bangalore	Kolkata	SpiceJet	5297.97
Bangalore	Kolkata	GO_FIRST	5794.16
Bangalore	Mumbai	AirAsia	3342.39
Bangalore	Mumbai	Indigo	5112.27
Bangalore	Mumbai	GO_FIRST	5533.8
Chennai	Bangalore	AirAsia	1914.76
Chennai	Bangalore	SpiceJet	2426.31
Chennai	Bangalore	Indigo	2845.48
Chennai	Delhi	AirAsia	3697.31
Chennai	Delhi	GO_FIRST	4994.12
Chennai	Delhi	Indigo	5250.7
Chennai	Hyderabad	GO_FIRST	1613.11
Chennai	Hyderabad	AirAsia	2053.18
Chennai	Hyderabad	SpiceJet	3291.73
Chennai	Kolkata	AirAsia	3682.34
Chennai	Kolkata	SpiceJet	4672.56
Chennai	Kolkata	Indigo	5667.82
Chennai	Mumbai	AirAsia	2691.1

```
1 df.head()
```

	index	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	595
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	595
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	595
3	3	Vistara	UK-	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	595

```

1 #10. Find flights with duration more than 5 hours but priced below average.
2 cursor.execute(''':SELECT * FROM airflight WHERE duration>5 AND price<(SELECT AVG(price) FROM airflight) ''')
3 rows = cursor.fetchall()
4 columns = [desc[0] for desc in cursor.description]
5 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

level_0	index	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
18	18	AirAsia	I5-747	Delhi	Evening	one	Early_Morning	Mumbai				
19	19	AirAsia	I5-747	Delhi	Evening	one	Morning	Mumbai				
20	20	GO_FIRST	G8-266	Delhi	Early_Morning	one	Evening	Mumbai				
21	21	GO_FIRST	G8-101	Delhi	Early_Morning	one	Night	Mumbai				
22	22	GO_FIRST	G8-103	Delhi	Evening	one	Morning	Mumbai				
27	27	AirAsia	I5-744	Delhi	Morning	one	Afternoon	Mumbai				
30	30	GO_FIRST	G8-165	Delhi	Early_Morning	one	Afternoon	Mumbai				
31	31	Indigo	0.00E+00	Delhi	Afternoon	one	Evening	Mumbai				
32	32	Vistara	UK-813	Delhi	Evening	one	Morning	Mumbai				
33	33	Vistara	UK-817	Delhi	Evening	one	Morning	Mumbai				
34	34	Vistara	UK-819	Delhi	Afternoon	one	Morning	Mumbai				
35	35	Vistara	UK-801	Delhi	Morning	one	Morning	Mumbai				
36	36	Vistara	UK-815	Delhi	Morning	one	Morning	Mumbai				
37	37	Air_India	AI-453	Delhi	Early_Morning	one	Afternoon	Mumbai				
39	39	SpiceJet	SG-2976	Delhi	Evening	one	Morning	Mumbai				
40	40	Air_India	AI-504	Delhi	Night	one	Morning	Mumbai				
41	41	Air_India	AI-502	Delhi	Afternoon	one	Morning	Mumbai				
42	42	Air_India	AI-506	Delhi	Morning	one	Morning	Mumbai				
43	43	Air_India	AI-803	Delhi	Early_Morning	one	Morning	Mumbai				
44	44	Air_India	AI-479	Delhi	Night	one	Afternoon	Mumbai				
45	45	SpiceJet	SG-339	Delhi	Evening	one	Afternoon	Mumbai				
47	47	Vistara	UK-627	Delhi	Afternoon	one	Afternoon	Mumbai				
48	48	AirAsia	I5-784	Delhi	Evening	one	Early_Morning	Mumbai				
49	49	Air_India	AI-9643	Delhi	Evening	one	Afternoon	Mumbai				
50	50	Air_India	AI-540	Delhi	Night	one	Morning	Mumbai				
51	51	Air_India	AI-429	Delhi	Morning	one	Morning	Mumbai				
52	52	Air_India	AI-439	Delhi	Early_Morning	one	Morning	Mumbai				

```

1 #11. Which flight class shows the highest average price variation (std deviation)?
2 cursor.execute(''': SELECT T1.class, CAST(SUM((price - avg_price)*(price - avg_price)) AS REAL) / COUNT(price) AS price_vari
3 FROM airflight AS T1
4 JOIN (
5     SELECT class, AVG(price) AS avg_price

```

```

6   FROM airflight
7   GROUP BY class
8 ) AS avg_prices ON T1.class = avg_prices.class
9 GROUP BY T1.class
10 ORDER BY price_variation DESC
11 LIMIT 1; ''')
12 rows = cursor.fetchall()
13 columns = [desc[0] for desc in cursor.description]
14 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

class	price_variation
Business	1.68201e+08

```

1 #12. Find how ticket prices vary with number of days left before departure.
2 cursor.execute('''SELECT days_left,AVG(price) as avg_price FROM airflight GROUP BY days_left''')
3 rows = cursor.fetchall()
4 columns = [desc[0] for desc in cursor.description]
5 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

days_left	avg_price
1	21591.9
2	30211.3
3	28976.1
4	25730.9
5	26679.8
6	24856.5
7	25588.4
8	24895.9
9	25726.2
10	25572.8
11	22990.7
12	22505.8
13	22498.9
14	22678
15	21952.5
16	20503.5
17	20386.4
18	19987.4
19	19507.7
20	19700
21	19430.5
22	19590.7
23	19840.9
24	19803.9
25	19571.6
26	19238.3
27	19950.9
28	19535

	index	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	SpiceJet	SG- ~~~	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953

```

1 #13. Which time of day (Morning, Evening, etc.) has the highest average ticket price?
2 cursor.execute('''SELECT departure_time, AVG(price) as avg_price FROM airflight GROUP BY departure_time ORDER BY avg_price DESC''')
3 rows = cursor.fetchall()
4 columns = [desc[0] for desc in cursor.description]
5 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

departure_time	avg_price
Night	23062.1

```

1 #14. Find Price trends by airline and travel class
2 cursor.execute('''SELECT airline,class,price FROM airflight
3 GROUP BY airline,class
4 ORDER BY price'''')
5 rows = cursor.fetchall()
6 columns = [desc[0] for desc in cursor.description]
7 print(tabulate(rows,headers = columns,tablefmt = 'grid'))

```

airline	class	price
SpiceJet	Economy	5953
GO_FIRST	Economy	5954
Air_India	Economy	5955
Indigo	Economy	5955
Vistara	Economy	5955
AirAsia	Economy	5956