

**SOURCE
CODE**

Importing the libraries

```
In [66]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing data

```
In [67]: data = pd.read_csv(r"C:\Users\Pracheeta\Desktop\training\Filtered_Flight_Data.csv")
data.head(10)
```

C:\Users\Pracheeta\AppData\Local\Temp\ipykernel_2216\969922091.py:1: DtypeWarning: Columns (11) have mixed types. Specify dtype option on import or set low_memory=False.

```
data = pd.read_csv(r"C:\Users\Pracheeta\Desktop\training\Filtered_Flight_Data.csv")
```

Out[67]:

	Unnamed: 0	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	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954

Data copy

```
In [68]: datacopy = data
datacopy.head(10)
```

Out[68]:

	Unnamed: 0	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	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954

Analysis 1

Plotting data with sub-data and charts

```
In [69]: # Percentage of flights available vs airline
flight_count = datacopy.groupby(['airline'], as_index = False)['flight'].count()
flight_count.rename(columns = {'flight': 'Count_Flights', 'airline': 'Airline'}, inplace = True)
flight_count

# Total flights
```

```
In [69]: # Percentage of flights available vs airline
flight_count = datacopy.groupby(['airline'], as_index = False)['flight'].count()
flight_count.rename(columns = {'flight':'Count_Flights', 'airline':'Airline'}, inplace = True)
flight_count

# Total flights
total_flights = flight_count['Count_Flights'].sum()
flight_count['Percentage'] = ((flight_count['Count_Flights']/total_flights) * 100).round(2)
flight_count.sort_values(by = 'Count_Flights', ascending = False, inplace = True)
flight_count
```

Out[69]:

	Airline	Count_Flights	Percentage
5	Vistara	128727	42.74
1	Air_India	81060	26.91
3	Indigo	43128	14.32
2	GO_FIRST	23176	7.69
0	AirAsia	16100	5.35
4	SpiceJet	9015	2.99

```
In [70]: # Percentage of flights available vs city
flight_count_city = datacopy.groupby(['source_city'], as_index = False)['flight'].count()
flight_count_city.rename(columns = {'source_city':'City', 'flight':'Count_Flights'}, inplace = True)
flight_count_city

#Total flights
total_flights_city = flight_count_city['Count_Flights'].sum()
flight_count_city['Percentage'] = ((flight_count_city['Count_Flights']/total_flights_city) * 100).round(2)
flight_count_city.sort_values(by = 'Count_Flights', ascending = False, inplace = True)
flight_count_city
```

Out[70]:

	City	Count_Flights	Percentage
--	------	---------------	------------

Out[70]:

	City	Count_Flights	Percentage
2	Delhi	61394	20.38
5	Mumbai	60896	20.22
0	Bangalore	52061	17.28
4	Kolkata	46347	15.39
3	Hyderabad	40806	13.55
1	Chennai	39702	13.18

```
In [72]: # Plotting the above data
plot, axis = plt.subplots(nrows = 1,ncols = 2,figsize = (20,5))

# Plotting barplot
sns.barplot(x = "Airline", y = "Percentage", data = flight_count, ax = axis[0], palette = 'magma')
axis[0].set_xlabel('Airline', fontsize=14)
axis[0].set_ylabel('Count of Flights (%)', fontsize=14)
axis[0].tick_params(axis='both',labelsize=13)

#axis[0].set_xticklabels(axis[0].get_xticklabels(), rotation=90, fontsize=13)

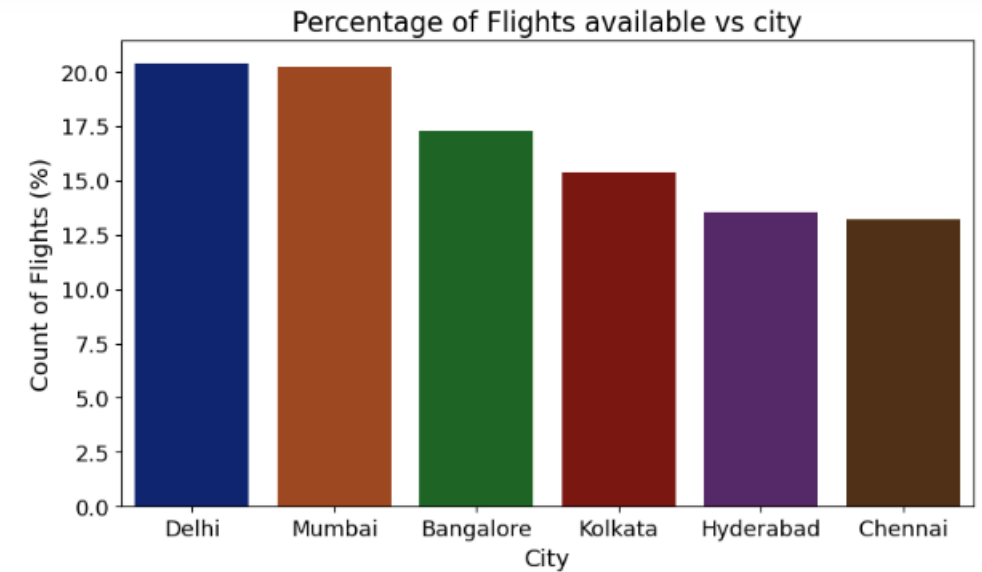
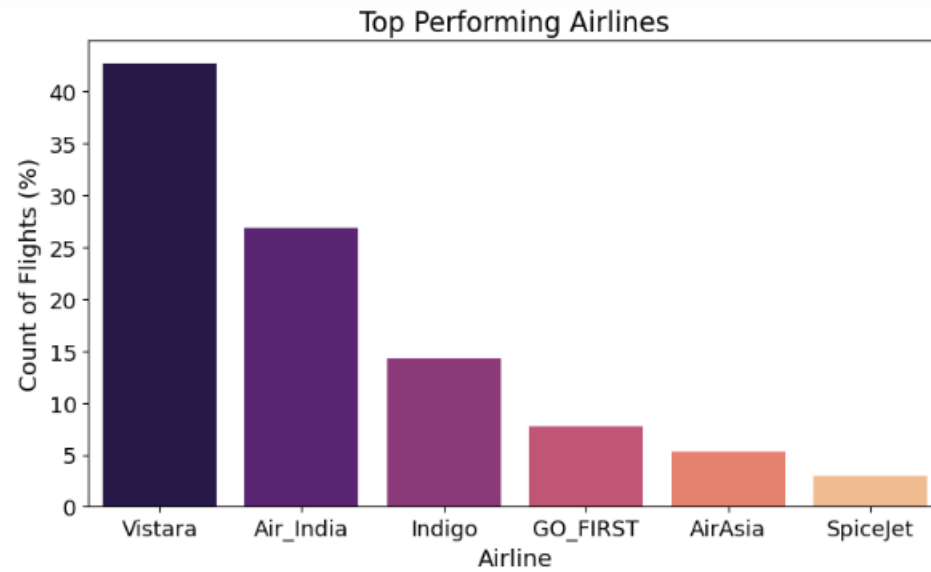
axis[0].set_title('Top Performing Airlines', fontsize = 16)
sns.barplot(x = "City", y = "Percentage", data = flight_count_city, ax = axis[1],
            palette = 'dark')
axis[1].set_xlabel('City',fontsize=14)
axis[1].set_ylabel('Count of Flights (%)', fontsize=14)
axis[1].tick_params(axis='both', labelsize=13)

#axis[1].set_xticklabels(axis[1].get_xticklabels(), rotation=90, fontsize=13)
axis[1].set_title('Percentage of Flights available vs city', fontsize = 16)

plt.show()
```

Top Performing Airlines

Percentage of Flights available vs city



Conclusion

- Vistara Flights are more in number than other Flights.
- Availability of flights of Delhi and Mumbai are more in number.

```
In [73]: datacopy.head(10)
```

Out[73]:

	Unnamed: 0	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	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954

Analysis 2

```
In [74]: # Departure_time vs source City vs Flight_Count
```

```
flight_count_city = datacopy.groupby(['source_city', 'departure_time'], as_index = False)['flight'].count()
flight_count_city.rename(columns = {'source_city': 'City', 'departure_time': 'Departure_time', 'flight': 'Count_Flights'}, inplace =
flight_count_city
```

Out[74]:

	City	Departure_time	Count_Flights
0	Bangalore	Afternoon	5183
1	Bangalore	Early_Morning	13611

```
In [75]: # For better analysis, Replace Early_Morning -> Morning, Late_Night -> Night... (Departure)
flight_count_city['Departure_time'].replace({'Early_Morning':'Morning', 'Late_Night':'Night'}, inplace = True)
flight_count_city
```

Out[75]:

	City	Departure_time	Count_Flights
0	Bangalore	Afternoon	5183
1	Bangalore	Morning	13611
2	Bangalore	Evening	14243
3	Bangalore	Night	457
4	Bangalore	Morning	12323
5	Bangalore	Night	6244
6	Chennai	Afternoon	5905
7	Chennai	Morning	9568
8	Chennai	Evening	5546
9	Chennai	Night	72
10	Chennai	Morning	10841
11	Chennai	Night	7770
12	Delhi	Afternoon	11246
13	Delhi	Morning	12254
14	Delhi	Evening	16809
15	Delhi	Night	357
16	Delhi	Morning	13686
17	Delhi	Night	7042
18	Hyderabad	Afternoon	7221
19	Hvderabad	Morning	8524


```
In [76]: data_FC_Departure = flight_count_city.groupby(['City', 'Departure_time'], as_index = False)['Count_Flights'].sum()  
data_FC_Departure
```

Out[76]:

	City	Departure_time	Count_Flights
0	Bangalore	Afternoon	5183
1	Bangalore	Evening	14243
2	Bangalore	Morning	25934
3	Bangalore	Night	6701
4	Chennai	Afternoon	5905
5	Chennai	Evening	5546
6	Chennai	Morning	20409
7	Chennai	Night	7842
8	Delhi	Afternoon	11246
9	Delhi	Evening	16809
10	Delhi	Morning	25940
11	Delhi	Night	7399
12	Hyderabad	Afternoon	7221
13	Hyderabad	Evening	5991
14	Hyderabad	Morning	18447
15	Hyderabad	Night	9147
16	Kolkata	Afternoon	7863
17	Kolkata	Evening	9594
18	Kolkata	Morning	20198
19	Kolkata	Night	8692
20	Mumbai	Afternoon	10486

```
In [77]: # Arrival time vs Destination City vs Flight_Count
flight_count_Arrival_city = datacopy.groupby(['destination_city', 'arrival_time'], as_index = False)['flight'].count()
flight_count_Arrival_city.rename(columns = {'destinantion_city': 'destination_City', 'arrival_time': 'Arrival_Time', 'flight': 'Cour
flight_count_Arrival_city
```

Out[77]:

	destination_city	Arrival_Time	Count_Flights
0	Bangalore	Afternoon	4827
1	Bangalore	Early_Morning	1823
2	Bangalore	Evening	13937
3	Bangalore	Late_Night	3176
4	Bangalore	Morning	11246
5	Bangalore	Night	16059
6	Chennai	Afternoon	2731
7	Chennai	Early_Morning	3481
8	Chennai	Evening	9318
9	Chennai	Late_Night	975
10	Chennai	Morning	10680
11	Chennai	Night	13183
12	Delhi	Afternoon	10206
13	Delhi	Early_Morning	536
14	Delhi	Evening	13940
15	Delhi	Late_Night	6296
16	Delhi	Morning	10338
17	Delhi	Night	16044
18	Hyderabad	Afternoon	6463

```
In [78]: flight_count_Arrival_city['Arrival_Time'].replace({'Early_Morning':'Morning', 'Late_Night':'Night'}, inplace = True)
flight_count_Arrival_city
```

Out[78]:

	destination_city	Arrival_Time	Count_Flights
0	Bangalore	Afternoon	4827
1	Bangalore	Morning	1823
2	Bangalore	Evening	13937
3	Bangalore	Night	3176
4	Bangalore	Morning	11246
5	Bangalore	Night	16059
6	Chennai	Afternoon	2731
7	Chennai	Morning	3481
8	Chennai	Evening	9318
9	Chennai	Night	975
10	Chennai	Morning	10680
11	Chennai	Night	13183
12	Delhi	Afternoon	10206
13	Delhi	Morning	536
14	Delhi	Evening	13940
15	Delhi	Night	6296
16	Delhi	Morning	10338
17	Delhi	Night	16044
18	Hyderabad	Afternoon	6463
19	Hyderabad	Morning	3920
20	Hyderabad	Evening	10771

```
In [79]: data_FC_Departure1 = flight_count_Arrival_city.groupby(['destination_city', 'Arrival_Time'], as_index = False)['Count_Flights'].sort_values(by=['destination_city', 'Count_Flights'], ascending = False)
data_FC_Departure1
```

Out[79]:

	destination_city	Arrival_Time	Count_Flights
23	Mumbai	Night	23527
22	Mumbai	Morning	14373
21	Mumbai	Evening	12717
20	Mumbai	Afternoon	8531
17	Kolkata	Evening	17917
18	Kolkata	Morning	14359
19	Kolkata	Night	11718
16	Kolkata	Afternoon	5540
15	Hyderabad	Night	14913
14	Hyderabad	Morning	11581
13	Hyderabad	Evening	10771
12	Hyderabad	Afternoon	6463
11	Delhi	Night	22340
9	Delhi	Evening	13940
10	Delhi	Morning	10874
8	Delhi	Afternoon	10206
6	Chennai	Morning	14161
7	Chennai	Night	14158
5	Chennai	Evening	9318

```
[379]: # Plotting the above data...
plot, axis = plt.subplots(nrows = 2, ncols = 1, figsize = (22, 12))

# Plotting barplot...
sns.barplot(x = "City", y = "Count_Flights", data = data_FC_Departure, ax = axis[0], hue = 'Departure_time',
            palette='viridis')
axis[0].set_xlabel('Source City', fontsize=14)
axis[0].set_ylabel('Count_Flights', fontsize=14)
axis[0].tick_params(axis='both', labelsize=13)
axis[0].set_title('Departure_time vs Source City vs Flight_Count', fontsize = 18)

sns.barplot(x = "Destination_City", y = "Count_Flights", data = data_FC_Departure1, ax = axis[1], hue = 'Arrival_Time',
            palette='BuPu')
axis[1].set_xlabel('Destination_City', fontsize=14)
axis[1].set_ylabel('Count_Flights', fontsize=14)
axis[1].tick_params(axis='both', labelsize=13)
axis[1].set_title('Arrival_time vs Destination_City vs Flight_Count', fontsize = 18)

plt.show()
```



Conclusion

- People prefer to take morning flight from their respective source_city.

In [82]: `datacopy.head(10)`

Out[82]:

	Unnamed: 0	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	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954

Analysis 3

```
In [83]: # Flight count vs Flight class in different airline
data_Airline_class = datacopy.groupby(['airline', 'class'], as_index = False)['flight'].count()
data_Airline_class
```

Analysis 3

```
In [83]: # Flight count vs Flight class in different airline
data_Airline_class = datacopy.groupby(['airline', 'class'], as_index = False)['flight'].count()
data_Airline_class
```

Out[83]:

	airline	class	flight
0	AirAsia	Economy	16100
1	Air_India	Business	32996
2	Air_India	Economy	48064
3	GO_FIRST	Economy	23176
4	Indigo	Economy	43128
5	SpiceJet	Economy	9015
6	Vistara	Business	61293
7	Vistara	Economy	67434

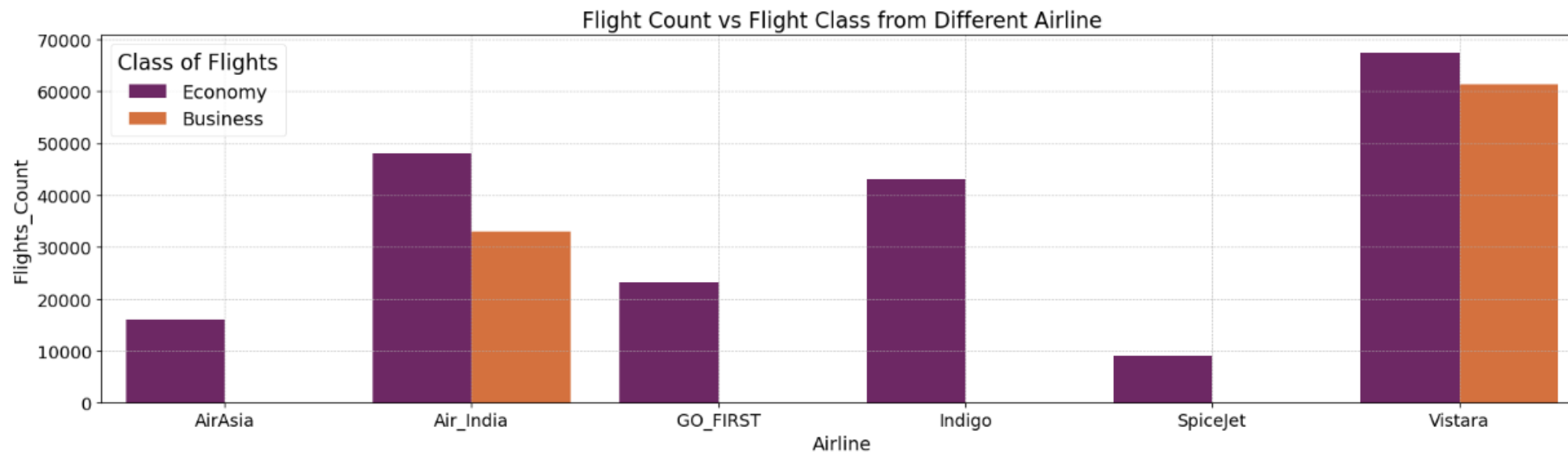
```
In [84]: # Plotting the above data
plot, axis = plt.subplots(nrows = 1, ncols = 1, figsize = (20,5))

# Plotting barplot
sns.barplot(x = "airline", y = "flight", data = data_Airline_class, ax = axis, hue = 'class', palette = 'inferno')
axis.set_xlabel('Airline', fontsize=14)
axis.set_ylabel('Flights_Count', fontsize=14)
axis.tick_params(axis='both', labelsize=13)
axis.grid(visible = True, which='both', linestyle='--', linewidth=0.5)
axis.set_title('Flight Count vs Flight Class from Different Airline', fontsize = 16)

# Change the value of legends
legend = axis.legend(title='Class of Flights', title_fontsize= '16', fontsize= '14')
legend.get_frame().set_linewidth(0.5)
```



```
plt.show()
```



Conclusion

- Passengers prefer to choose business class exclusively from air india and vistara.
- Among these, vistara is the most preferred airline for business class
- Similarly, vistara also holds the highest preference for economy class among travellers.

```
In [85]: datacopy.head(10)
```

```
Out[85]:
```

	Unnamed: 0	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	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956

Analysis 4

In [86]:

```
datacopy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301206 entries, 0 to 301205
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0            301206 non-null  int64
1   airline               301206 non-null  object
2   flight               301206 non-null  object
3   source_city           301206 non-null  object
4   departure_time        301206 non-null  object
5   stops                301206 non-null  object
6   arrival_time          301206 non-null  object
7   destination_city      301206 non-null  object
8   class                 301206 non-null  object
9   duration              301206 non-null  float64
10  days_left             301206 non-null  int64
11  price                 301206 non-null  object
dtypes: float64(1), int64(2), object(9)
memory usage: 27.6+ MB
```

In [87]:

```
# changing datatype of price
datacopy['price'].value_counts()
```

Out[87]:

```
54608    1445
2339     1442
54684    1390
60978    1383
60508    1230
...
14374      1
5048       1
8075       1
13203      1
15983      1
Name: price, Length: 12270, dtype: int64
```

```
In [88]: # Some non numeric values are present so we have to convert it to null
datacopy['price'] = pd.to_numeric(datacopy['price'], errors='coerce')

# Total null value set
datacopy['price'].isnull().sum()
```

Out[88]: 2

```
In [89]: # Changing null values to mode values
Mod_price = datacopy['price'].mode()[0]
Mod_price
```

Out[89]: 54608.0

```
In [90]: # Setting Null values to Mod_Price
datacopy['price'] = datacopy['price'].fillna(Mod_price)
```

```
In [91]: # Total null value have changed so no null valued
datacopy['price'].isnull().sum()
```

Out[91]: 0

```
In [92]: # Data changed of price column
datacopy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301206 entries, 0 to 301205
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  -
 0   Unnamed: 0         301206 non-null  int64
 1   airline            301206 non-null  object
 2   flight             301206 non-null  object
 3   source_city        301206 non-null  object
 4   departure_time     301206 non-null  object
 5   stops              301206 non-null  object
 6   arrival_time       301206 non-null  object
 7   destination_city   301206 non-null  object
 8   class              301206 non-null  object
 9   duration            301206 non-null  float64
10   days left          301206 non-null  int64
```

```
In [93]: # Top 5 flights that gets travel more
flight_travel = datacopy['flight'].value_counts().reset_index()
flight_travel.columns = ['flight', 'Flight_Count']
flight_travel = flight_travel.head(5)
flight_travel
```

Out[93]:

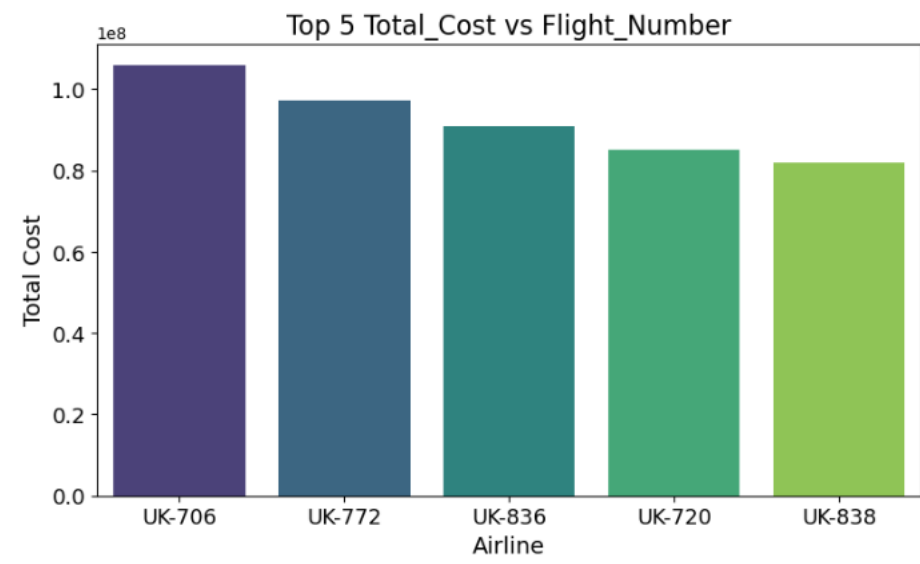
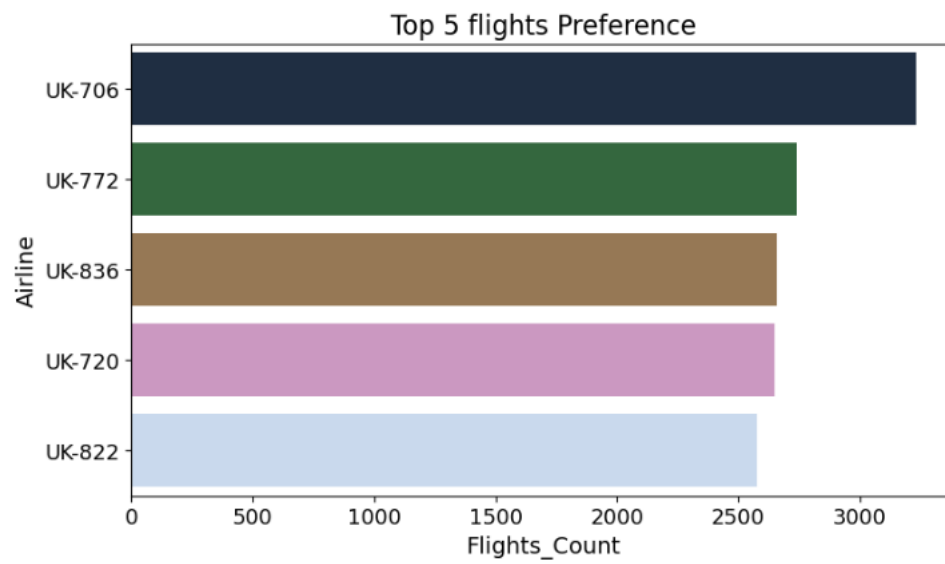
	flight	Flight_Count
0	UK-706	3235
1	UK-772	2741
2	UK-836	2657
3	UK-720	2650
4	UK-822	2575

```
In [94]: # Top 5 flights total revenue vs flight number
data_F_Price = datacopy.groupby(['flight'], as_index = False)['price'].sum()
data_F_Price = data_F_Price.sort_values(by = ['price'], ascending = False)
top5_data_F_Price = data_F_Price.head(5)
top5_data_F_Price
```

Out[94]:

	flight	price
1442	UK-706	105871560.0
1454	UK-772	97212901.0
1490	UK-836	91016350.0
1445	UK-720	85182167.0
1492	UK-838	82050784.0

```
In [150]: plot, axis = plt.subplots(nrows = 1, ncols = 2, figsize = (20,5))
sns.barplot(x = 'Flight_Count', y = "Flight", data = flight_travel, ax = axis[0], orient = 'h', palette='cubehelix')
axis[0].set_xlabel('Flights_count', fontsize=14)
axis[0].set_ylabel('Airline', fontsize=14)
axis[0].tick_params(axis='both', labelsize=13)
axis[0].set_title('Top 5 flights Preference', fontsize = 16)
```



Conclusion

- The airline UK-706 operates a greater number of flights.
- UK-822 has a higher number of flights compared to UK-838, yet UK-838 generates more revenue than UK-822.
- UK-706 and UK-772 exhibit a dramatic variation in the number of flights. However, the total cost remains relatively stable between the two.

```
In [96]: datacopy.size
```

```
Out[96]: 3614472
```

Analysis 4

```
In [97]: len = datacopy.shape[0]  
pd.options.display.max_rows = len
```

```
In [136]: datacopy['flight'].value_counts()
```

AI-640	950
UK-945	931
UK-954	931
UK-933	925
UK-977	925
AI-773	917
UK-993	911
AI-610	891
UK-902	870
UK-951	866
UK-943	835
UK-941	828
AI-526	818
AI-525	815
UK-898	813
UK-975	782
AI-732	767
UK-877	757
UK-657	755
AI-766	753

In [137]: `datacopy.head(10)`

Out[137]:

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price	Total_stops
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953.0	0
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953.0	0
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956.0	0
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955.0	0
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955.0	0
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955.0	0
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060.0	0
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060.0	0
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954.0	0
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954.0	0

```
In [142]: # Common destination cities from each source city
Common_Destination = datacopy['destination_city'].value_counts().reset_index()
Common_Destination.rename(columns={'destination_city': 'Destination_City', 'count': 'flight_Count'}, inplace=True)
Common_Destination.sort_values
Common_Destination
```

Out[142]:

	index	Destination_City	flight_Count
0	Mumbai	59148	
1	Delhi	57360	
2	Bangalore	51068	
3	Kolkata	49534	
4	Hyderabad	43728	
5	Chennai	40368	

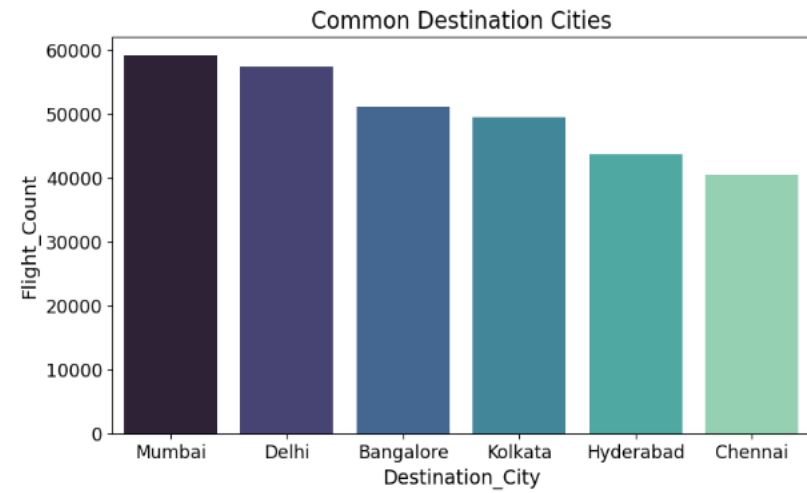
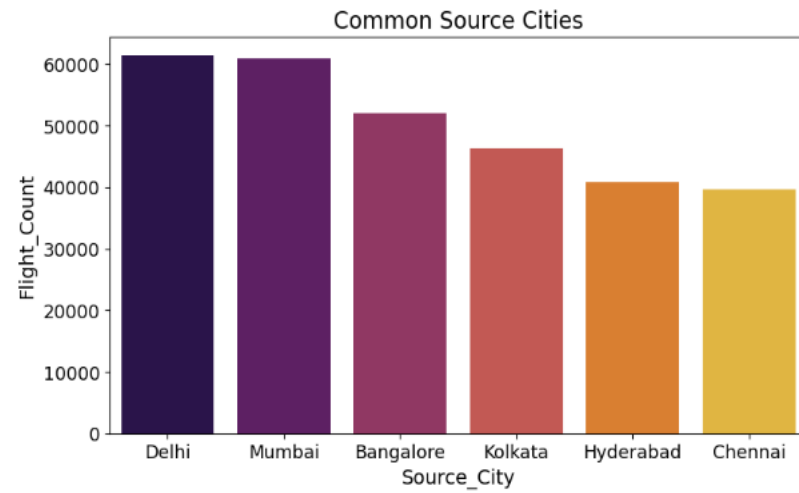
```
In [143]: # Common source cities
Common_Source = datacopy['source_city'].value_counts().reset_index()
Common_Source.rename(columns={'source_city': 'Source_City', 'count': 'Flight_Count'}, inplace=True)
Common_Source.sort_values
Common_Source
```

Out[143]:

	index	Source_City
0	Delhi	61394
1	Mumbai	60896
2	Bangalore	52061
3	Kolkata	46347
4	Hyderabad	40806
5	Chennai	39702

```
In [152]: plot, axis = plt.subplots(nrows = 1, ncols = 2, figsize = (20,5))
sns.barplot(x = 'Source_City', y = "Flight_count", data = Common_Source, ax = axis[0], palette='inferno')
axis[0].set_xlabel('Source_City', fontsize=14)
axis[0].set_ylabel('Flight_Count', fontsize=14)
axis[0].tick_params(axis='both', labelsize=13)
axis[0].set_title('Common Source Cities', fontsize = 16)

sns.barplot(x = 'Destination_City', y = "Flight_count", data = Common_Destination, ax = axis[1], palette='mako')
axis[1].set_xlabel('Destination_City', fontsize=14)
axis[1].set_ylabel('Flight_Count', fontsize=14)
axis[1].tick_params(axis='both', labelsize=13)
axis[1].set_title('Common Destination Cities', fontsize=16)
plt.show
```

Conclusion

- The most frequent source city is Delhi.
- The most common destination city is Mumbai.

```
In [109]: datacopy.head(10)
```

```
Out[109]:
```

	Unnamed: 0	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	5953.0
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953.0
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956.0
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955.0
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955.0
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955.0
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060.0
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060.0
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954.0
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954.0

Analysis 6

```
In [110]: # Airlines offer the fastest routes for common destination
fastest_routes = datacopy.groupby(['destination_city', 'airline'], as_index=False)['duration'].min()
fastest_routes.rename(columns={'duration': 'Fastest_Duration'}, inplace=True)
fastest_routes = fastest_routes.sort_values(by=['destination_city', 'Fastest_Duration'])
fastest_routes
```

```
Out[110]:
```

	destination_city	airline	Fastest_Duration
3	Bangalore	Indigo	0.83
0	Bangalore	AirAsia	0.92
1	Bangalore	Air_India	1.00
2	Bangalore	GO_FIRST	1.00
5	Bangalore	Vistara	1.00
4	Bangalore	SpiceJet	1.33
9	Chennai	Indigo	0.83

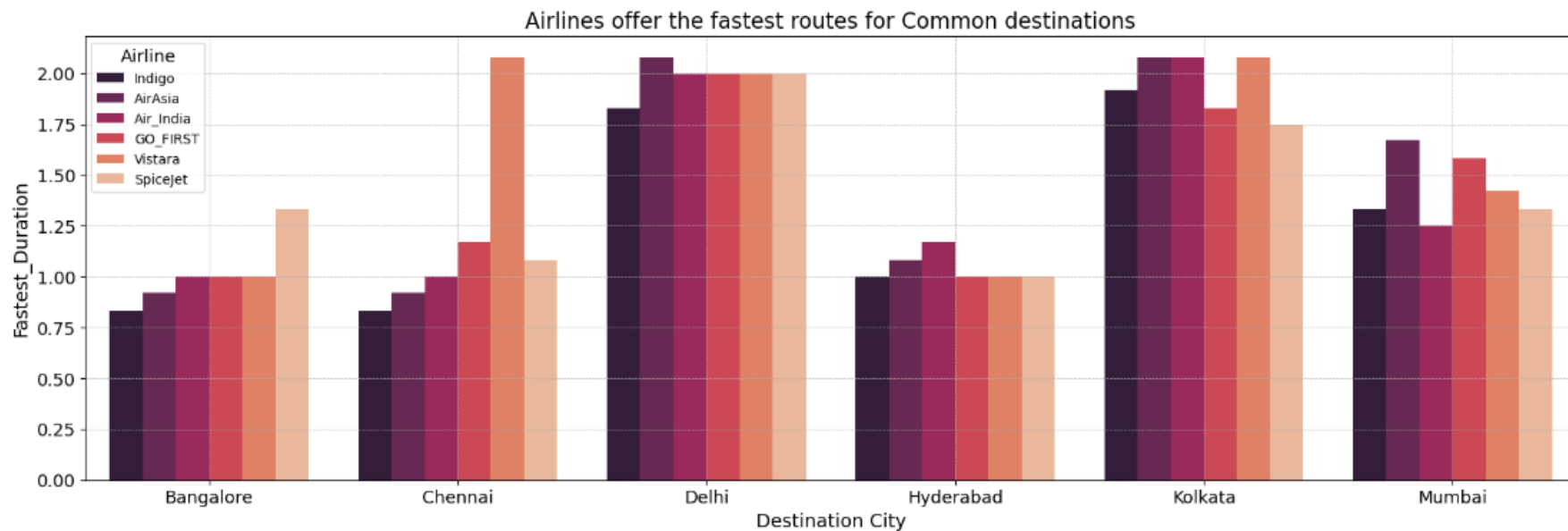
```

In [111]: # Plotting the above data
plot, axis = plt.subplots(nrows = 1, ncols = 1, figsize = (20,6))

# Plotting barplot
sns.barplot(x = "destination_city", y = "Fastest_Duration", data = fastest_routes, ax = axis, hue = 'airline', palette='rocket')
axis.set_xlabel('Destination City', fontsize=14)
axis.set_ylabel('Fastest_Duration', fontsize=14)
axis.tick_params(axis='both', labels=13)
axis.legend(loc='upper left', title = 'Airline', title_fontsize = '13', fontsize = '10')
axis.grid(visible = True, which='both', linestyle='--', linewidth=0.5, zorder=1)
axis.set_title('Airlines offer the fastest routes for Common destinations', fontsize = 16)

plt.show()

```



Conclusion

- For Bangalore, the fastest flight is offered by SpiceJet.
- For Chennai, Vistara provides the fastest flights.
- For Delhi, AirAsia has the quickest flight.
- For Hyderabad, Air India offers the fastest flight.
- For Kolkata, the fastest flights are available with AirAsia, Air India, and Vistara.
- For Mumbai, AirAsia provides the fastest flight.

In [112]: `datacopy.head(10)`

Out[112]:

	Unnamed: 0	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	5953.0
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953.0
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956.0
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955.0
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955.0
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955.0
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060.0
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060.0
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954.0
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954.0

Analysis 7

In [113]: `datacopy['stops'].unique()`

Out[113]: `array(['zero', 'one', 'two_or_more'], dtype=object)`

```
In [114]: conditions = [
            (datacopy['stops'] == 'zero'),
            (datacopy['stops'] == 'one'),
            (datacopy['stops'] == 'two_or_more')
          ]
          choices = [0,1,2]
```

```
In [115]: # Adding an extra column for stops
          datacopy['Total_stops'] = np.select(conditions, choices, default = 0)
          datacopy.head(10)
```

Out[115]:

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price	Total_stops
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953.0	0
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953.0	0
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956.0	0
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955.0	0
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955.0	0
5	5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955.0	0
6	6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060.0	0
7	7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060.0	0
8	8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954.0	0
9	9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954.0	0

```
In [116]: datacopy['Total_stops'].unique()
```

Out[116]: array([0, 1, 2])

```
In [117]: stops = datacopy.groupby(['airline'], as_index=False)['Total_stops'].mean()
          stops = stops.sort_values(by=['Total_stops'], ascending = False)
          stops
```

Out[117]:

	airline	Total_stops
0	AirAsia	0.988323

```
In [118]: # Plotting the above data
plot, axis = plt.subplots(nrows = 1, ncols = 1, figsize = (8,4.2))

# Plotting barplot
original_palette = sns.color_palette('Blues')
# Reverse the palette
reversed_palette = original_palette[::-1]
sns.barplot(x = "airline", y = "Total_stops", data = stops, ax = axis, palette = reversed_palette)
axis.set_xlabel('Airline', fontsize=14)
axis.set_ylabel('Total_stops', fontsize=14)
axis.tick_params(axis='both', labels=13)
axis.set_title('Mean Stops of Airlines', fontsize = 16)

plt.show()
```

