DATA ANALYSE

1. Load the file

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
fp_df = pd.read_csv('/content/Clean_Dataset.csv')
```

2. Print first 5 rows of data

```
fp_df.head()
```

	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

3. Print last 5 rows of data

```
fp_df.tail()
```

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	pri
300148	300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	692
300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	771
300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	790

4. Cleaning the data for missing values, null values etc.

```
fp_df.isnull().sum()
```

	0
Unnamed: 0	0
airline	0
flight	0
source_city	0
departure_time	0
stops	0
arrival_time	0
destination_city	0
class	0
duration	0
days_left	0
price	0

dtype: int64

6. Get some info about the data

```
fp_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Unnamed: 0        300153 non-null  int64  
 1   airline           300153 non-null  object 
 2   flight            300153 non-null  object 
 3   source_city       300153 non-null  object 
 4   departure_time    300153 non-null  object 
 5   stops             300153 non-null  object 
 6   arrival_time      300153 non-null  object 
 7   destination_city  300153 non-null  object 
 8   class             300153 non-null  object 
 9   duration          300153 non-null  float64
 10  days_left         300153 non-null  int64  
 11  price             300153 non-null  int64  
dtypes: float64(1), int64(3), object(8)
memory usage: 27.5+ MB
```

7. Get some description about the data

```
fp_df.describe()
```

	Unnamed: 0	duration	days_left	price
count	300153.000000	300153.000000	300153.000000	300153.000000
mean	150076.000000	12.221021	26.004751	20889.660523
std	86646.852011	7.191997	13.561004	22697.767366
min	0.000000	0.830000	1.000000	1105.000000
25%	75038.000000	6.830000	15.000000	4783.000000
50%	150076.000000	11.250000	26.000000	7425.000000
75%	225114.000000	16.170000	38.000000	42521.000000
max	300152.000000	49.830000	49.000000	123071.000000

DATA VISUALIZATION

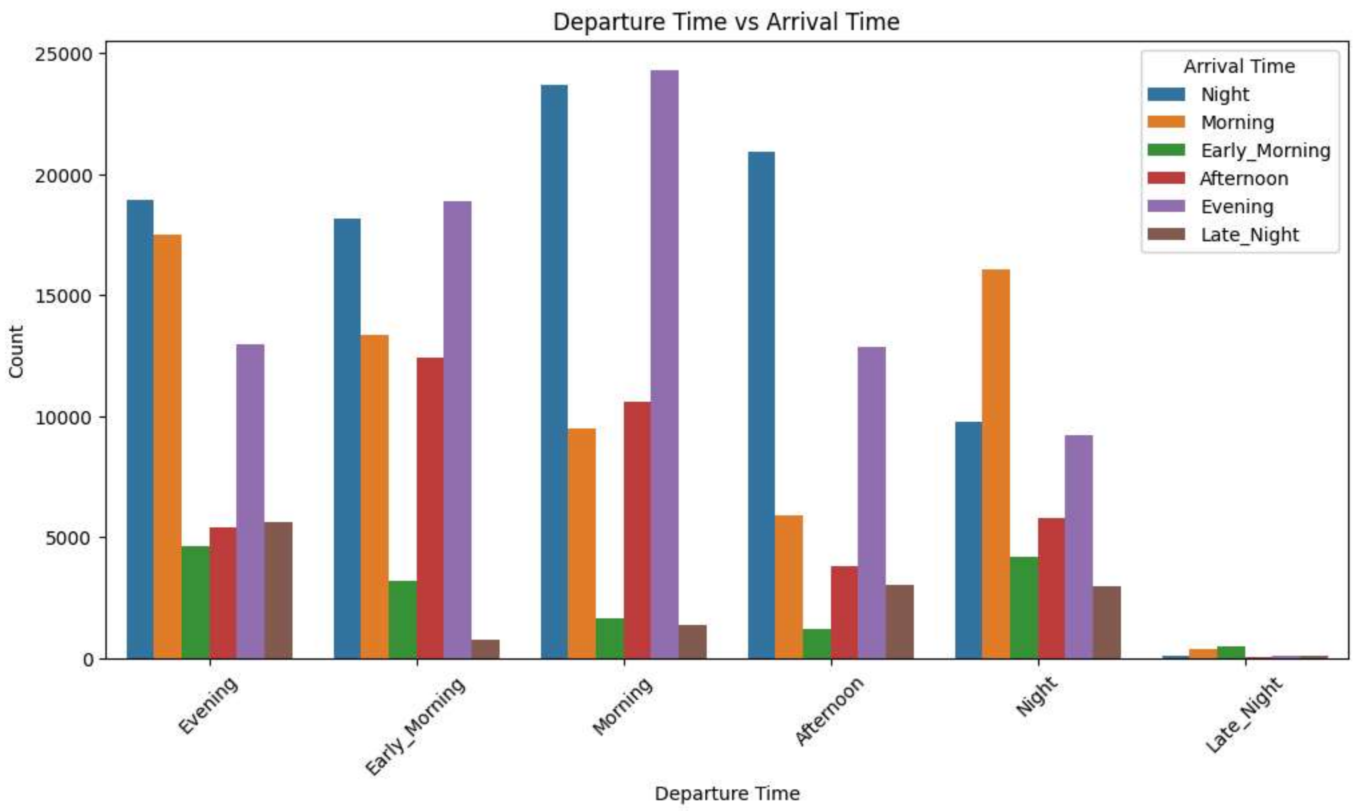
1. What are the airlines in the dataset, accompanied by their frequencies?

```
airline_counts = fp_df['airline'].value_counts()
print(airline_counts)
```

```
airline
Vistara      127859
Air_India     80892
Indigo        43120
GO_FIRST      23173
AirAsia       16098
SpiceJet       9011
Name: count, dtype: int64
```

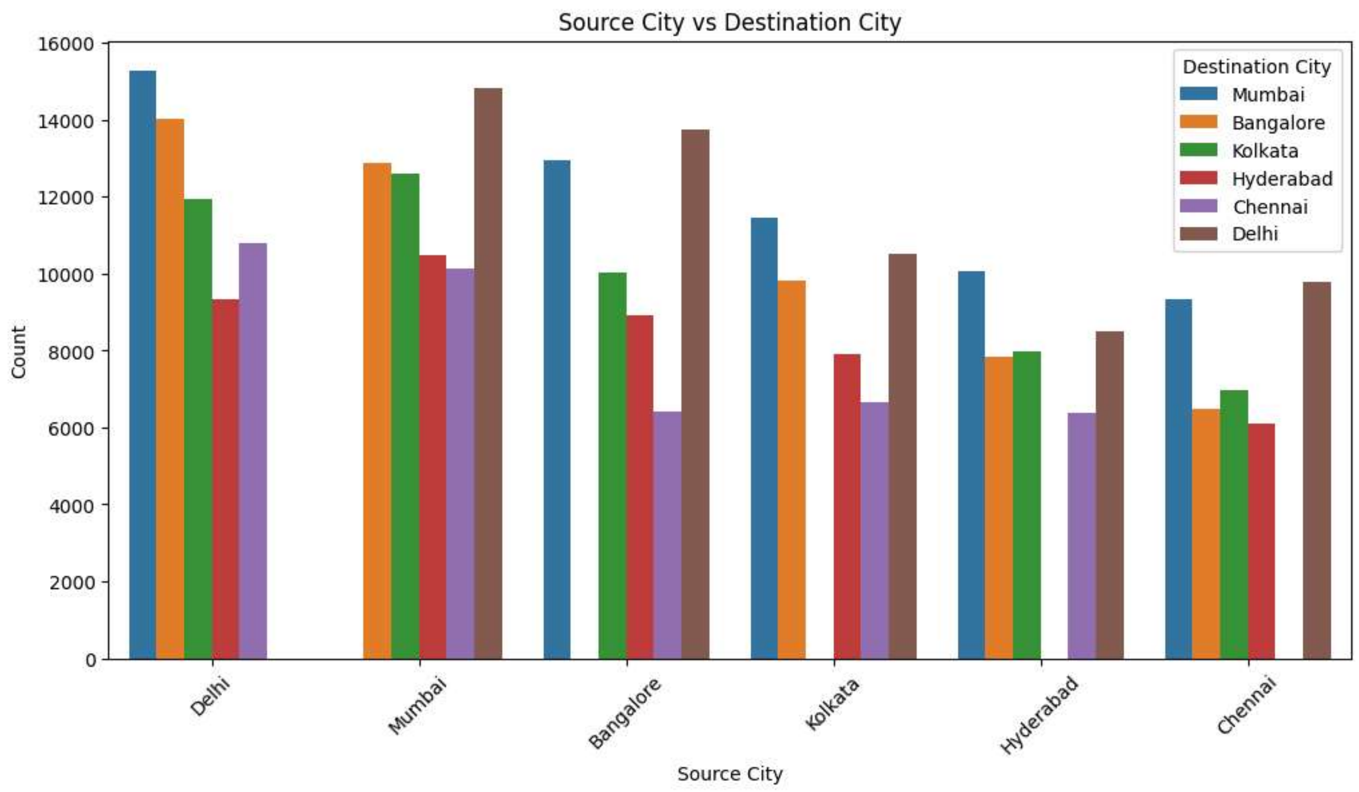
2. Departure time against Arrival time using barplot.

```
plt.figure(figsize=(12, 6))
sns.countplot(x='departure_time', hue='arrival_time', data=fp_df)
plt.xlabel('Departure Time')
plt.ylabel('Count')
plt.title('Departure Time vs Arrival Time')
plt.xticks(rotation=45)
plt.legend(title='Arrival Time')
plt.show()
```



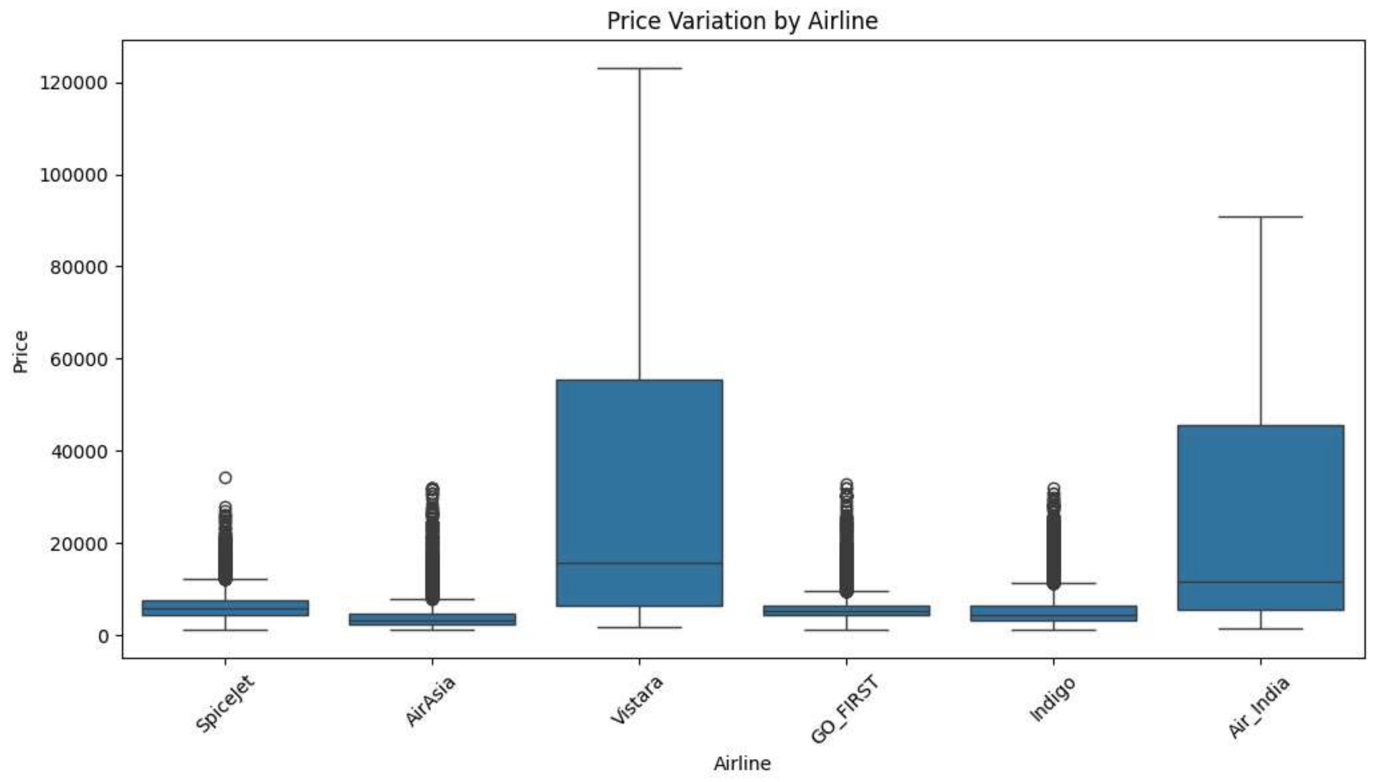
3. Source city against Destination city

```
plt.figure(figsize=(12, 6))
sns.countplot(x='source_city', hue='destination_city', data=fp_df)
plt.xlabel('Source City')
plt.ylabel('Count')
plt.title('Source City vs Destination City')
plt.xticks(rotation=45)
plt.legend(title='Destination City')
plt.show()
```



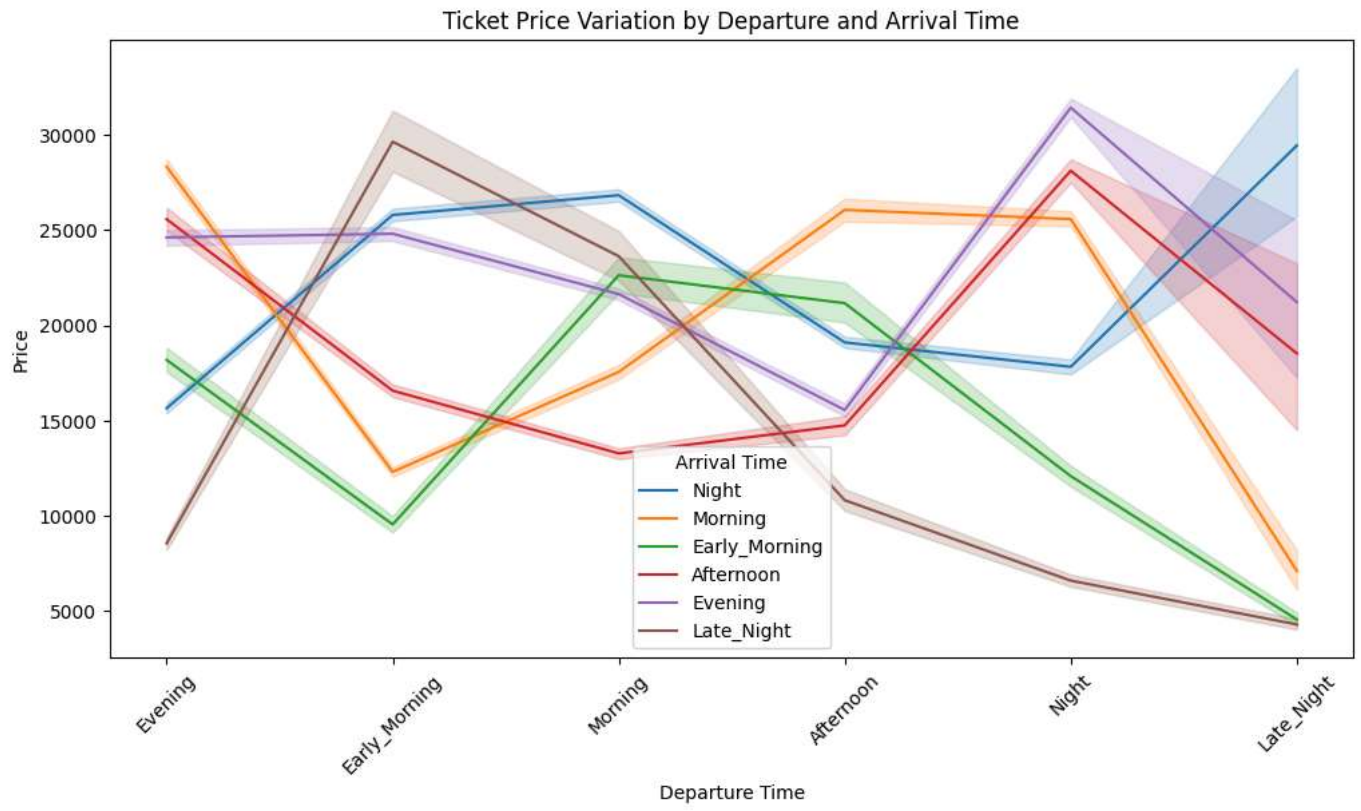
4. Does price vary with Airlines?

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='airline', y='price', data=fp_df)
plt.xlabel('Airline')
plt.ylabel('Price')
plt.title('Price Variation by Airline')
plt.xticks(rotation=45)
plt.show()
```



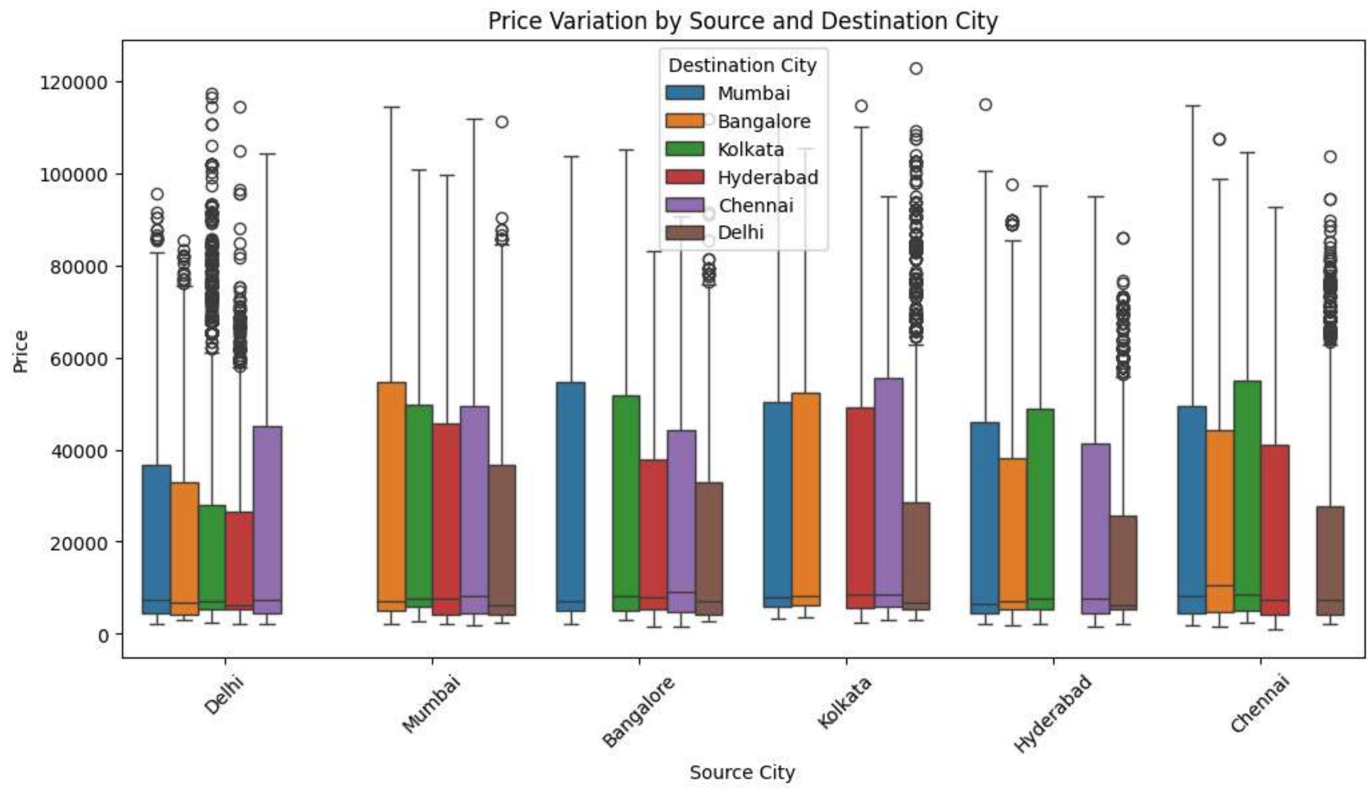
5. Does ticket price change based on the departure time and arrival time using line plot?

```
plt.figure(figsize=(12, 6))
sns.lineplot(x='departure_time', y='price', hue='arrival_time', data=fp_df)
plt.xlabel('Departure Time')
plt.ylabel('Price')
plt.title('Ticket Price Variation by Departure and Arrival Time')
plt.xticks(rotation=45)
plt.legend(title='Arrival Time')
plt.show()
```



6. How the price changes with change in Source and Destination?

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='source_city', y='price', hue='destination_city', data=fp_df)
plt.xlabel('Source City')
plt.ylabel('Price')
plt.title('Price Variation by Source and Destination City')
plt.xticks(rotation=45)
plt.legend(title='Destination City')
plt.show()
```



7. Duration of travel vs city

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='source_city', y='duration', data=fp_df)
plt.xlabel('Source City')
plt.ylabel('Duration')
plt.title('Duration of Travel by Source City')
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(x='destination_city', y='duration', data=fp_df)
plt.xlabel('Destination City')
plt.ylabel('Duration')
plt.title('Duration of Travel by Destination City')
plt.xticks(rotation=45)
plt.show()
```