Airlines Flights Data Analysis with Python - DSL

```
# import the required Python libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
data = pd.read_csv("/content/airlines_flights_data.csv")
```

```
data
```

	index	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_lef
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
			UK-								

Cleaning the data

```
# Remove the 'index' column

data.drop( columns = 'index', inplace = True)


data
```

	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953
1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
...	...	...	...	...	...	...	...	...	...	...	...
300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265
300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105
		UK-									

```
# Get some Info about the dataset

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 11 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   airline           300153 non-null  object
 1   flight            300153 non-null  object
 2   source_city       300153 non-null  object
 3   departure_time    300153 non-null  object
 4   stops             300153 non-null  object
 5   arrival_time      300153 non-null  object
```

```
 6   destination_city  300153 non-null  object
 7   class             300153 non-null  object
 8   duration          300153 non-null  float64
 9   days_left         300153 non-null  int64
 10  price             300153 non-null  int64
dtypes: float64(1), int64(2), object(8)
memory usage: 25.2+ MB
```

```
# Get Statistical summary about the dataset
data.describe()
```

	duration	days_left	price
count	300153.000000	300153.000000	300153.000000
mean	12.221021	26.004751	20889.660523
std	7.191997	13.561004	22697.767366
min	0.830000	1.000000	1105.000000
25%	6.830000	15.000000	4783.000000
50%	11.250000	26.000000	7425.000000
75%	16.170000	38.000000	42521.000000
max	49.830000	49.000000	123071.000000

```
# check for the missing values in any column
data.isnull().sum()
```

	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

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

```
data.head()
```

	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953
1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955

```
# Checking how many Airlines are in the dataset

data['airline'].nunique()
```

```
6
```

```
# Showing the names of the Airlines in the dataset

data['airline'].unique()
```

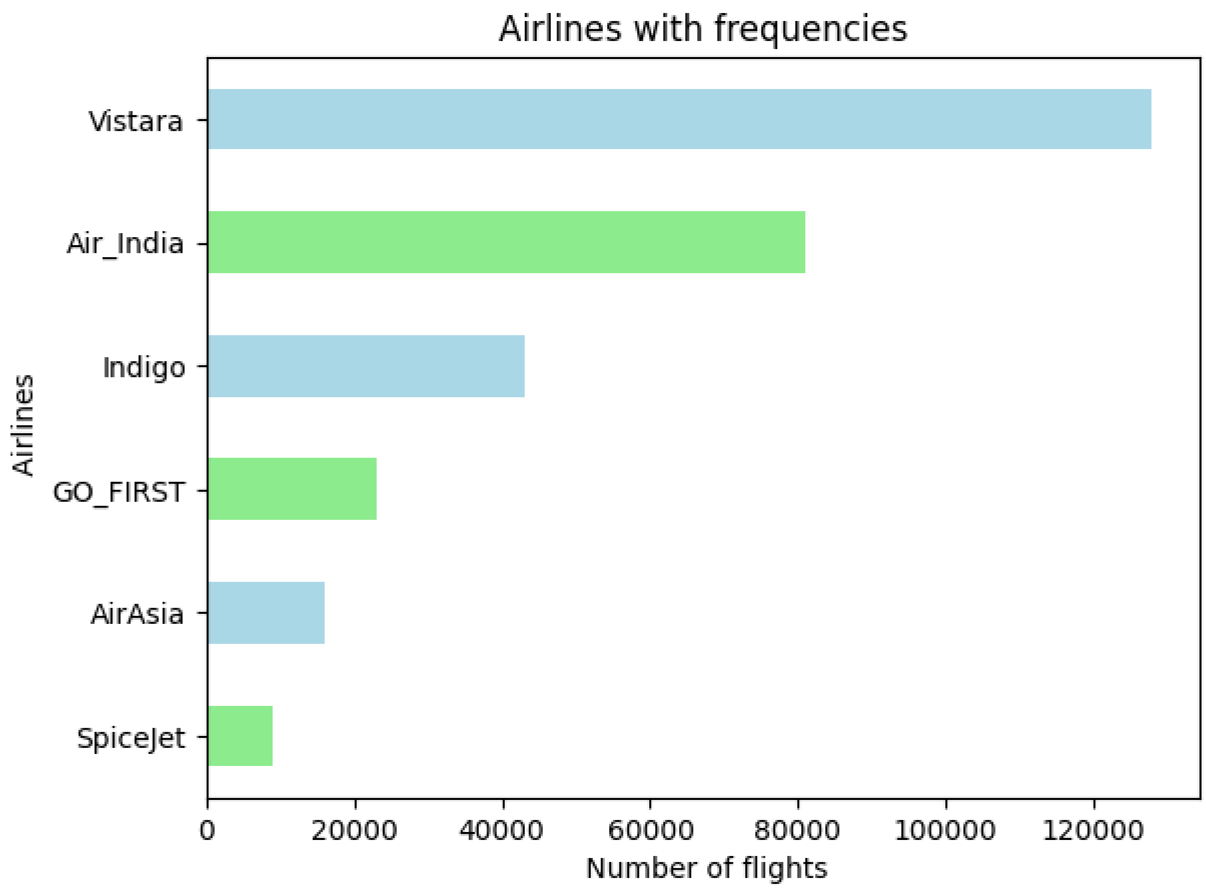
```
array(['SpiceJet', 'AirAsia', 'Vistara', 'GO_FIRST', 'Indigo',
       'Air_India'], dtype=object)
```

```
# Showing all the Airlines with their frequencies

data['airline'].value_counts()
```

airline	count
Vistara	127859
Air_India	80892
Indigo	43120
GO_FIRST	23173
AirAsia	16098
SpiceJet	9011

dtype: int64

```
# Showing all the Airlines with their Number of Flights in Horizontal Bar Graph

data['airline'].value_counts(ascending=True).plot.barh( color = ['lightgreen', 'lightblue'])

plt.title("Airlines with frequencies")

plt.xlabel(" Number of flights")

plt.ylabel(" Airlines")

plt.show()
```



Q.2. Show Bar Graphs representing the Departure Time & Arrival Time

```
# Showing the Departure Time for the flights

data['departure_time'].value_counts()
```

departure_time	count
Morning	71146
Early_Morning	66790
Evening	65102
Night	48015
Afternoon	47794
Late_Night	1306

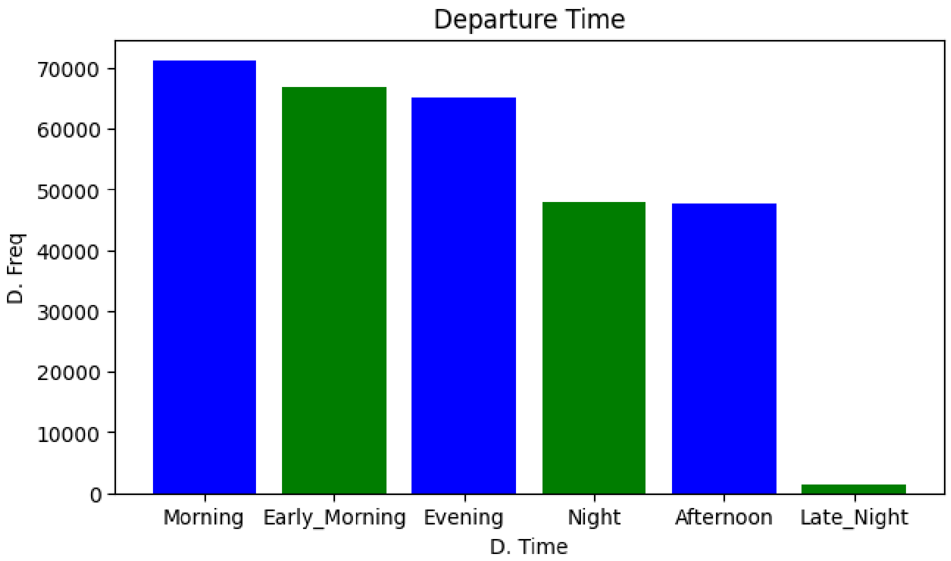
dtype: int64

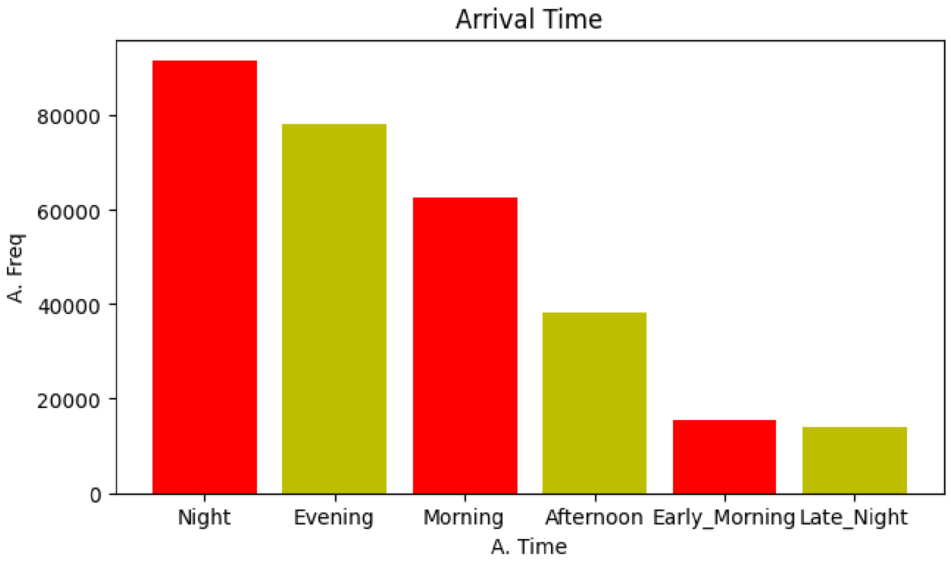
```
# Showing the Arrival Time for the flights

data['arrival_time'].value_counts()
```

arrival_time	count
Night	91538
Evening	78323
Morning	62735
Afternoon	38139
Early_Morning	15417
Late_Night	14001

dtype: int64

```
# Showing the Departure Time & Arrival Time for the flights with their counts

plt.figure(figsize = (16,4))

plt.subplot(1,2,1)

plt.bar( data['departure_time'].value_counts().index , data['departure_time'].value_counts().values, color = ['b', 'g'] )
plt.title("Departure Time")
plt.xlabel("D. Time")
plt.ylabel("D. Freq")

plt.subplot(1,2,2)

plt.bar( data['arrival_time'].value_counts().index, data['arrival_time'].value_counts().values, color = ['r', 'y'])
plt.title("Arrival Time")
plt.xlabel("A. Time")
plt.ylabel("A. Freq")

plt.show()
```





Q.3. Show Bar Graphs representing the Source City & Destination City

```
# Showing the Source City of the flights

data['source_city'].value_counts()
```

source_city	count
Delhi	61343
Mumbai	60896
Bangalore	52061
Kolkata	46347
Hyderabad	40806
Chennai	38700

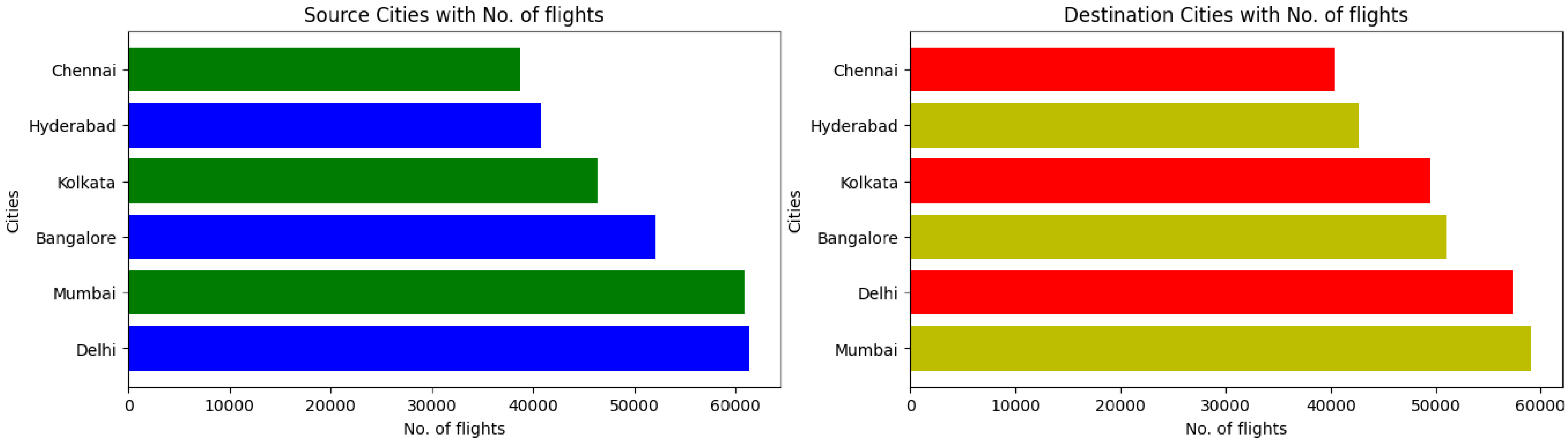
dtype: int64

```
# Showing the Destination City of the flights

data['destination_city'].value_counts()
```

destination_city	count
Mumbai	59097
Delhi	57360
Bangalore	51068
Kolkata	49534
Hyderabad	42726
Chennai	40368

dtype: int64

```
# Showing the Source City & Destination City for the flights with their counts

plt.figure( figsize= (16,4))

plt.subplot(1,2,1)

plt.barh( data['source_city'].value_counts().index , data['source_city'].value_counts().values, color = ['b', 'g'])
plt.title("Source Cities with No. of flights")
plt.ylabel("Cities")
plt.xlabel("No. of flights")

plt.subplot(1,2,2)

plt.barh( data['destination_city'].value_counts().index , data['destination_city'].value_counts().values, color = ['y', 'r'])
plt.title("Destination Cities with No. of flights")
plt.ylabel("Cities")
plt.xlabel("No. of flights")

plt.show()
```



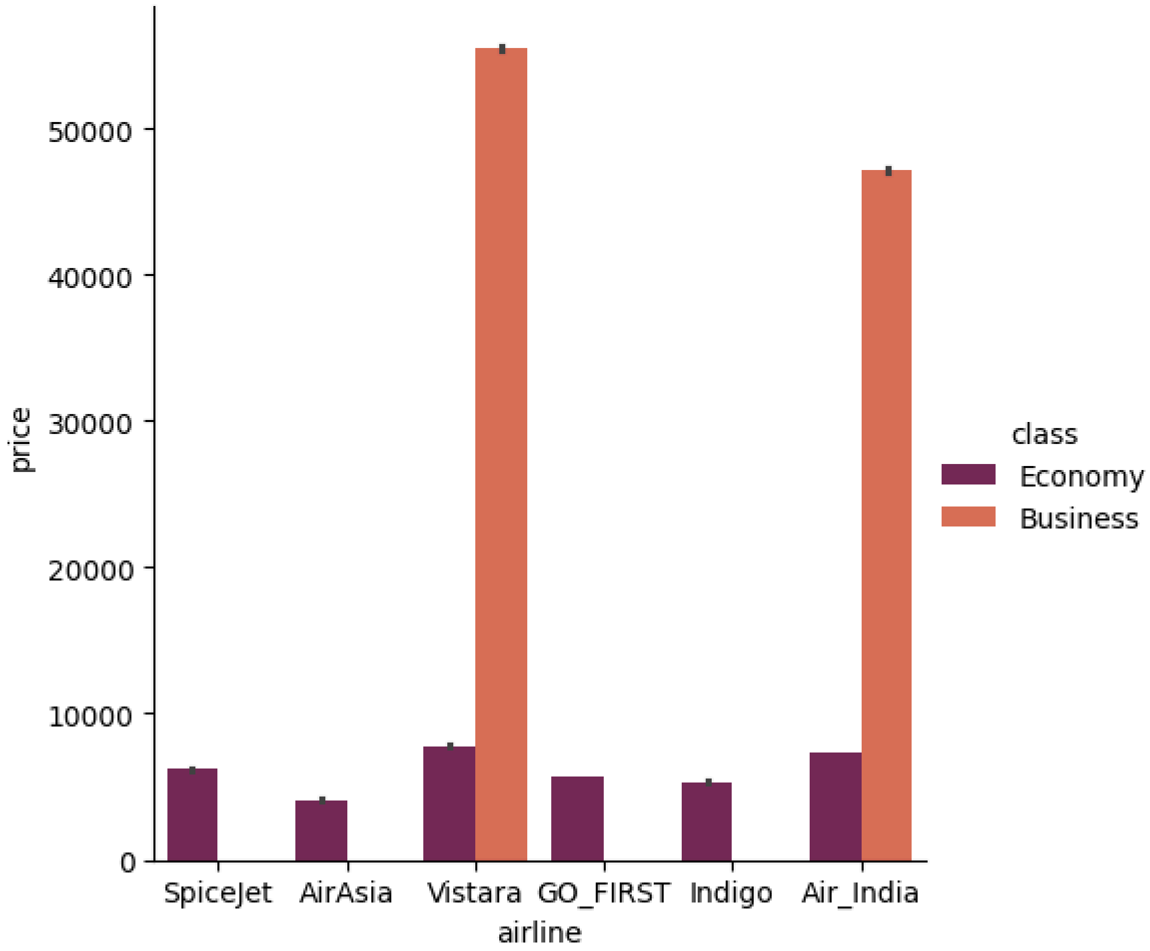
Q.4. Does price varies with airlines ?

```
# Grouping the airlines and checking their mean price

data.groupby('airline')['price'].mean()
```

	price
airline	
AirAsia	4091.072742
Air_India	23507.019112
GO_FIRST	5652.007595
Indigo	5324.216303
SpiceJet	6179.278881
Vistara	30396.536302

dtype: float64

```
# Drawing a Categorical Plot showing the Mean Ticket Price for each Airline

sns.catplot( x = 'airline', y = 'price', kind = 'bar', palette = 'rocket', data = data, hue = 'class')

plt.show()
```



Q.5. Does ticket price change based on the departure time and arrival time?

```
# Checking the Mean Ticket Price based on the Departure Times

data.groupby('departure_time')['price'].mean()
```

departure_time	price
Afternoon	18179.203331
Early_Morning	20370.676718
Evening	21232.361894
Late_Night	9295.299387
Morning	21630.760254
Night	23062.146808

dtype: float64

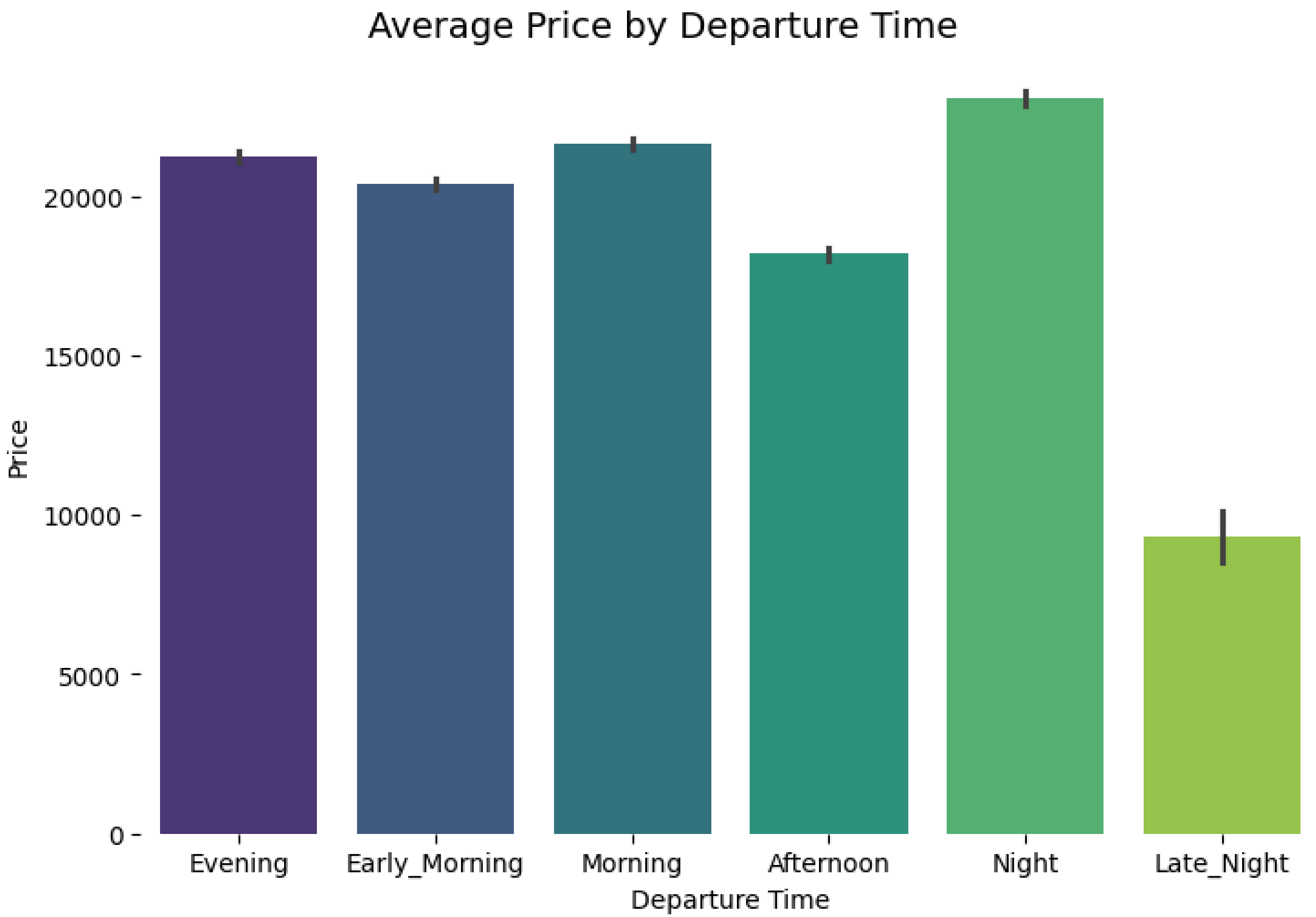
```
# Checking the Mean Ticket Price based on the Arrival Times

data.groupby('arrival_time')['price'].mean()
```

arrival_time	price
Afternoon	18494.598993
Early_Morning	14993.139521
Evening	23044.371615
Late_Night	11284.906078
Morning	22231.076098
Night	21586.758341

dtype: float64

```
# Create the plot
g = sns.catplot(
    x='departure_time',
    y='price',
    kind='bar',
    data=data,
    palette='viridis',   # colorful palette
    height=5,
    aspect=1.5
)

# Customize labels and title
g.set_axis_labels("Departure Time", "Price")
g.fig.suptitle("Average Price by Departure Time", fontsize=14, y=1.02)

# Remove extra spines for cleaner look
sns.despine(left=True, bottom=True)

plt.show()
```

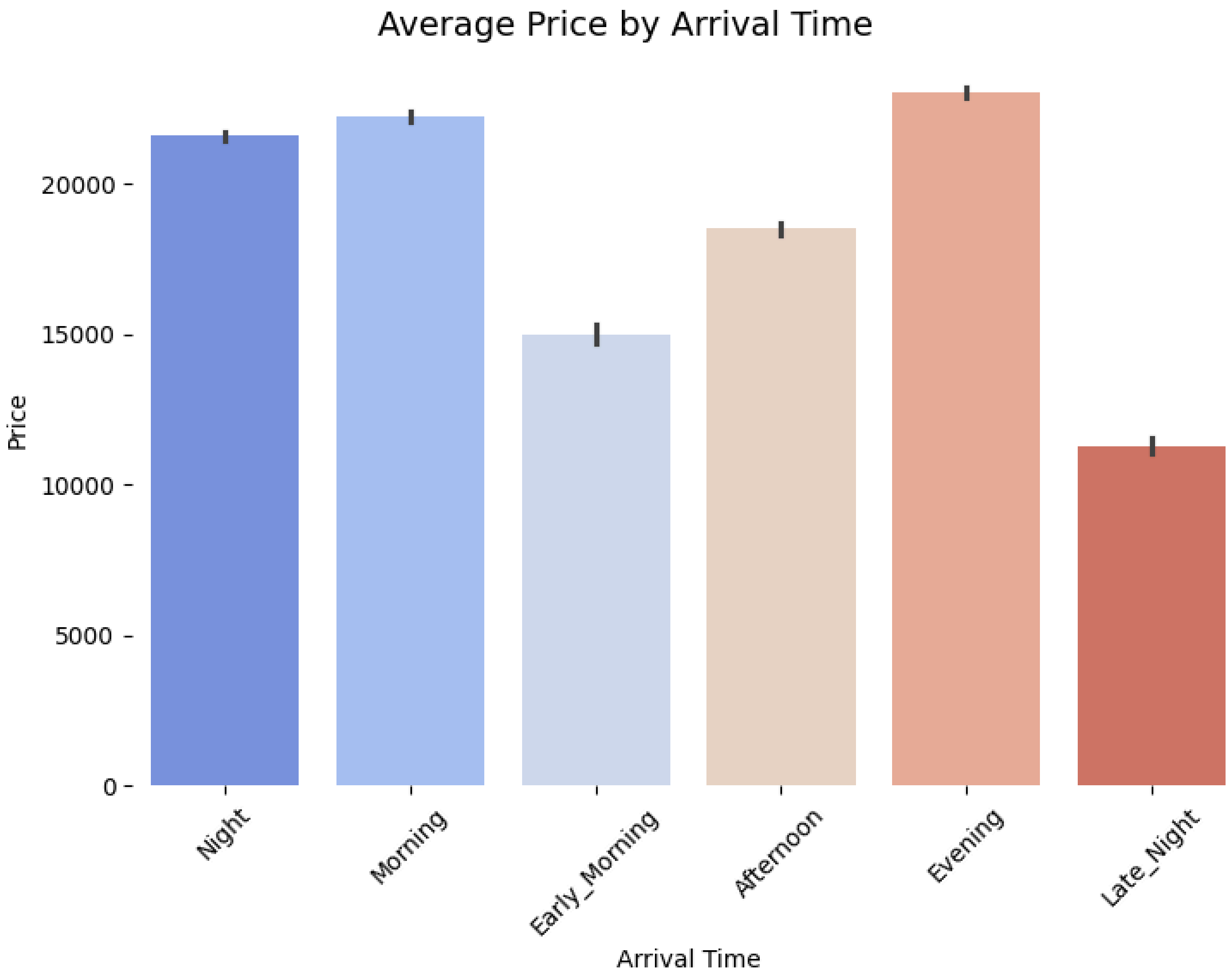
```
/tmp/ipython-input-1835243693.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set

  g = sns.catplot(
```



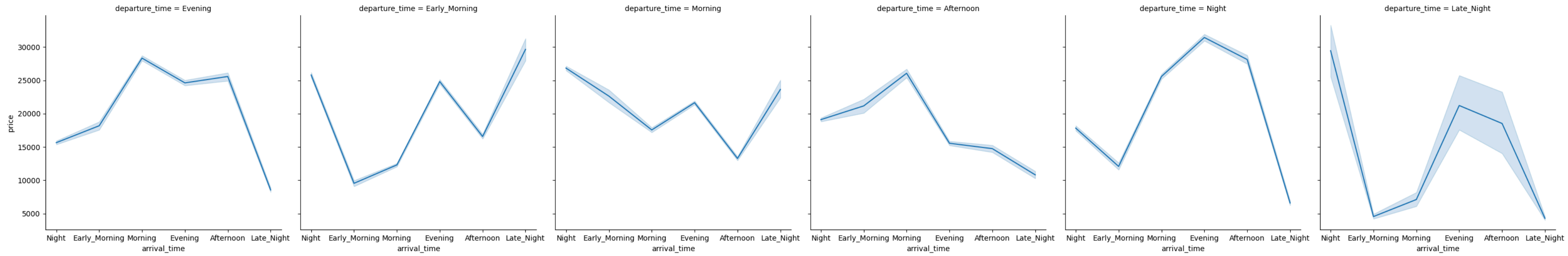

```
# Create the plot
g = sns.catplot(
    x='arrival_time',
    y='price',
```

```
        kind='bar',
        data=data,
        palette='coolwarm',   # distinct color palette
        height=5,
        aspect=1.5
    )

    # Customize labels and title
    g.set_axis_labels("Arrival Time", "Price")
    g.fig.suptitle("Average Price by Arrival Time", fontsize=14, y=1.02)

    # Rotate x-axis labels if they overlap
    g.set_xticklabels(rotation=45)

    # Remove extra spines
    sns.despine(left=True, bottom=True)

    plt.show()
```

```
/tmp/ipython-input-2748057074.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set

  g = sns.catplot(
```



```python
sns.relplot( x = 'arrival_time', y = 'price', data = data, col = 'departure_time', kind = 'line')

plt.show()
```



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

```python
# Checking the Mean Ticket Price for each Source City

data.groupby('source_city')['price'].mean()
```

source_city	price
Bangalore	21469.460575
Chennai	21995.339871
Delhi	18951.326639
Hyderabad	20155.623879
Kolkata	21746.235679
Mumbai	21483.818839

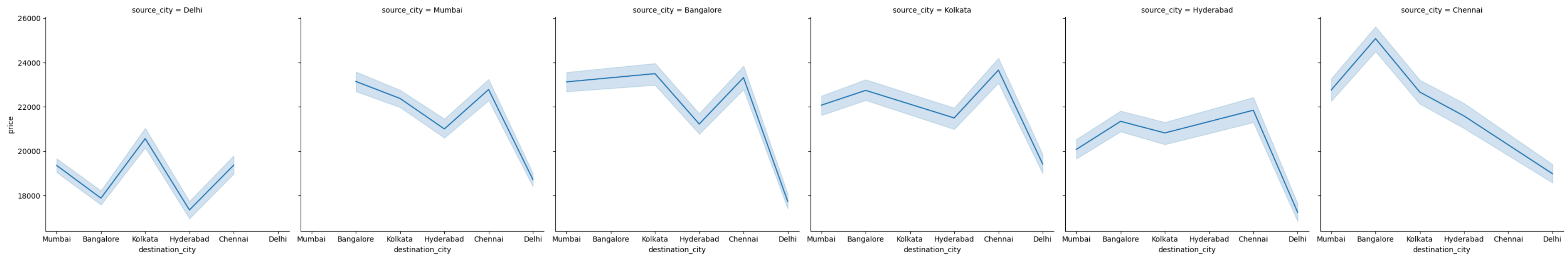
dtype: float64

```
# Checking the Mean Ticket Price for each Destination City

data.groupby('destination_city')['price'].mean()
```

destination_city	price
Bangalore	21593.955784
Chennai	21953.323969
Delhi	18436.767870
Hyderabad	20427.661284
Kolkata	21959.557556
Mumbai	21372.529469

dtype: float64

```
sns.relplot( x = 'destination_city', y = 'price', data = data, col = "source_city", kind = 'line')

plt.show()
```



Q.7. How is the price affected when tickets are bought in just 1 or 2 days before departure?

```
data['days_left'].nunique()
```

```
49
```

```
data['days_left'].unique()
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

```
# Checking the Mean Ticket Price for different days_left

data.groupby('days_left')['price'].mean()
```

	price
days_left	
1	21591.867151
2	30211.299801
3	28976.083569
4	25730.905653
5	26679.773368
6	24856.493902
7	25588.367351
8	24895.883995
9	25726.246072
10	25572.819134
11	22990.656070
12	22505.803322
13	22498.885384
14	22678.002363
15	21952.540852
16	20503.546237
17	20386.353949
18	19987.445168
19	19507.677375
20	19699.983390
21	19430.494058