

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

#Load the file
Flight_df=pd.read_csv('/content/Clean_Dataset.csv')
Flight_df
```

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price	<div><div></div></div>	
	0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953	<div><div></div></div>
	1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953	<div><div></div></div>
	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	
	...	...	...	...	...	...	...	...	...	...	...	...	...	
	300148	300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265	
	300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105	
	300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	79099	
	300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	Evening	Hyderabad	Business	10.00	49	81585	
	300152	300152	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	81585	

300153 rows × 12 columns

```
#First five rows of data
Flight_df.head()
```

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price	<div><div></div><div></div></div>
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953	<div><div></div><div></div></div>
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	

```
#Last five rows of data
Flight_df.tail()
```

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price	<div><div></div><div></div></div>	
	300148	300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265	<div><div></div><div></div></div>
	300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105	
	300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	79099	
	300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	Evening	Hyderabad	Business	10.00	49	81585	
	300152	300152	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	81585	

```
#Missing values
Flight_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

```
#Reset index column
Flight_df.rename(columns={'Unnamed: 0':"index"},inplace=True)
print(Flight_df)
```

	index	airline	flight	source_city	departure_time	stops	\
	0	0	SpiceJet	SG-8709	Delhi	Evening	zero
	1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero
	2	2	AirAsia	I5-764	Delhi	Early_Morning	zero
	3	3	Vistara	UK-995	Delhi	Morning	zero
	4	4	Vistara	UK-963	Delhi	Morning	zero
	...	...	...	...	...	...	...
	300148	300148	Vistara	UK-822	Chennai	Morning	one
	300149	300149	Vistara	UK-826	Chennai	Afternoon	one
	300150	300150	Vistara	UK-832	Chennai	Early_Morning	one
	300151	300151	Vistara	UK-828	Chennai	Early_Morning	one
	300152	300152	Vistara	UK-822	Chennai	Morning	one

	arrival_time	destination_city	class	duration	days_left	price
0	Night	Mumbai	Economy	2.17	1	5953
1	Morning	Mumbai	Economy	2.33	1	5953
2	Early_Morning	Mumbai	Economy	2.17	1	5956
3	Afternoon	Mumbai	Economy	2.25	1	5955
4	Morning	Mumbai	Economy	2.33	1	5955
...	...	...	...	...	...	...
300148	Evening	Hyderabad	Business	10.08	49	69265
300149	Night	Hyderabad	Business	10.42	49	77105
300150	Night	Hyderabad	Business	13.83	49	79099
300151	Evening	Hyderabad	Business	10.00	49	81585
300152	Evening	Hyderabad	Business	10.08	49	81585

[300153 rows x 12 columns]

```
#Get info
Flight_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   index                 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
```

```
#Remove in data index column
Flight_df.drop(['index'],axis=1,inplace=True)
Flight_df.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

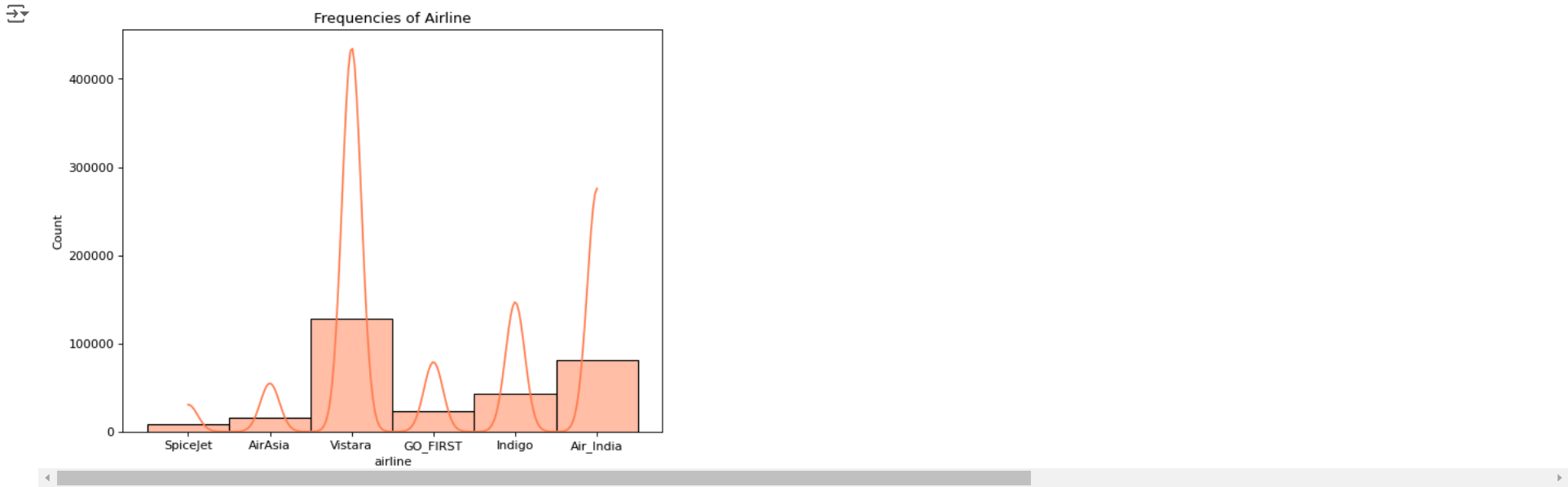
```
#Get info
Flight_df.info()
```

```
<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
```

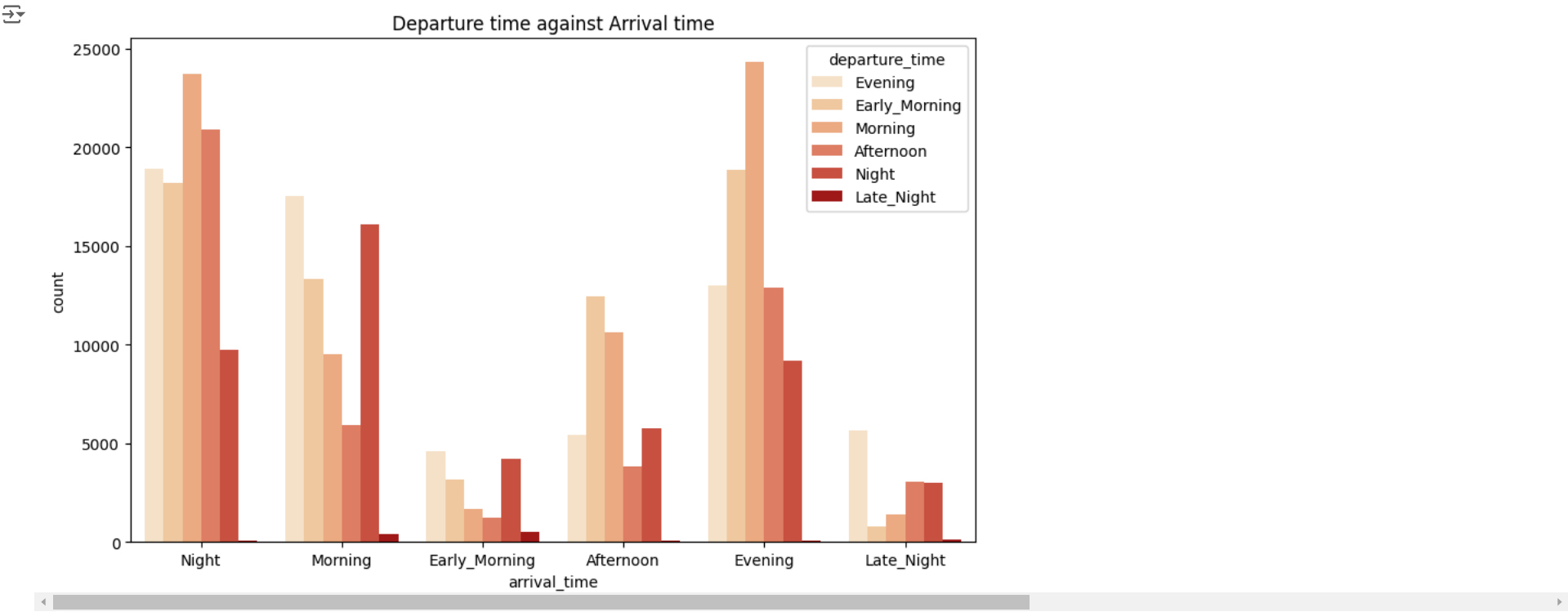
```
#Get description
Flight_df.describe()
```

	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

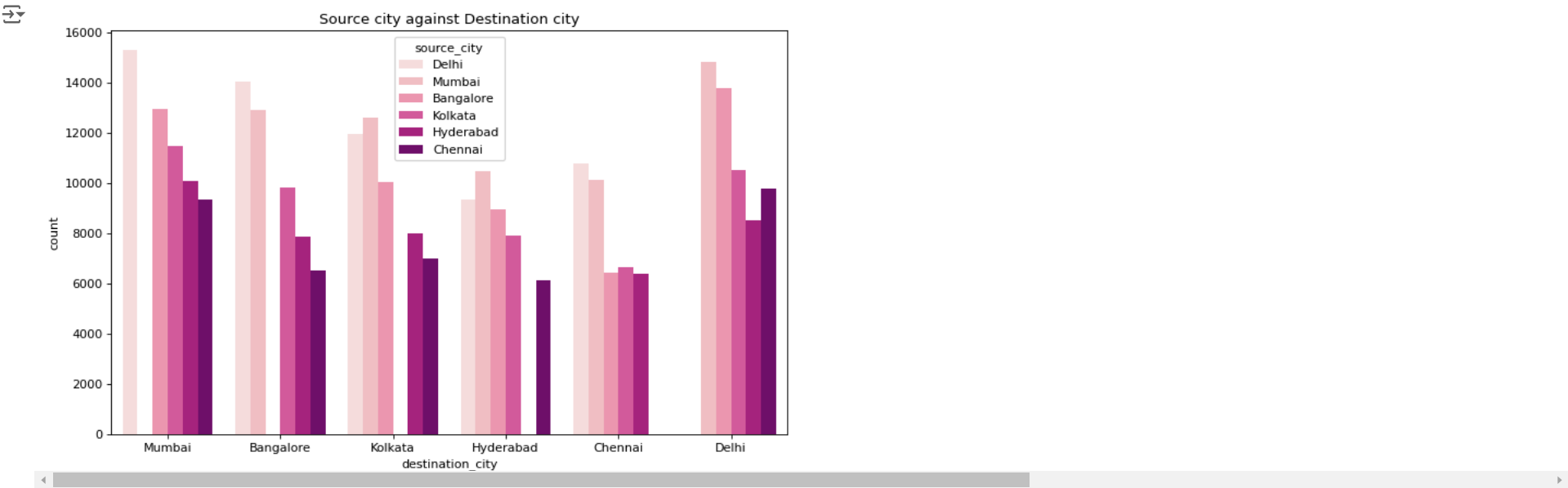
```
#1.Frequency for Airline
plt.figure(figsize=(8,6),dpi=80)
#Create Histogram
sns.histplot(Flight_df['airline'],kde=True,color='coral').set(title='Frequencies of Airline')
plt.show()
```



```
#2.Departure time against Arrival time
plt.figure(figsize=(10,6))
#Create countplot
sns.countplot(data=Flight_df, x='arrival_time',hue='departure_time',palette='OrRd')
plt.title('Departure time against Arrival time')
plt.show()
```



```
#3.Source city against Destination city
plt.figure(figsize=(10,6),dpi=80)
#Create countplot
sns.countplot(data=Flight_df, x='destination_city',hue='source_city',palette='RdPu')
plt.title('Source city against Destination city')
plt.show()
```

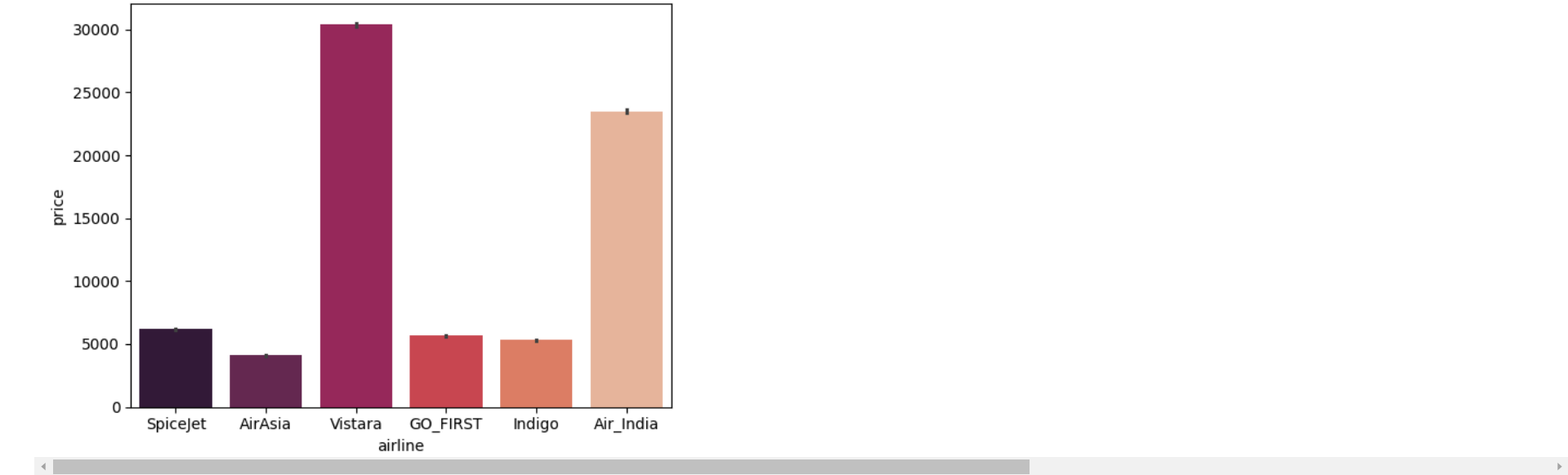


```
#4.Does price vary with Airlines?
#Create Barplot
sns.barplot(data=Flight_df,x='airline',y='price',palette='rocket')
plt.show()
```

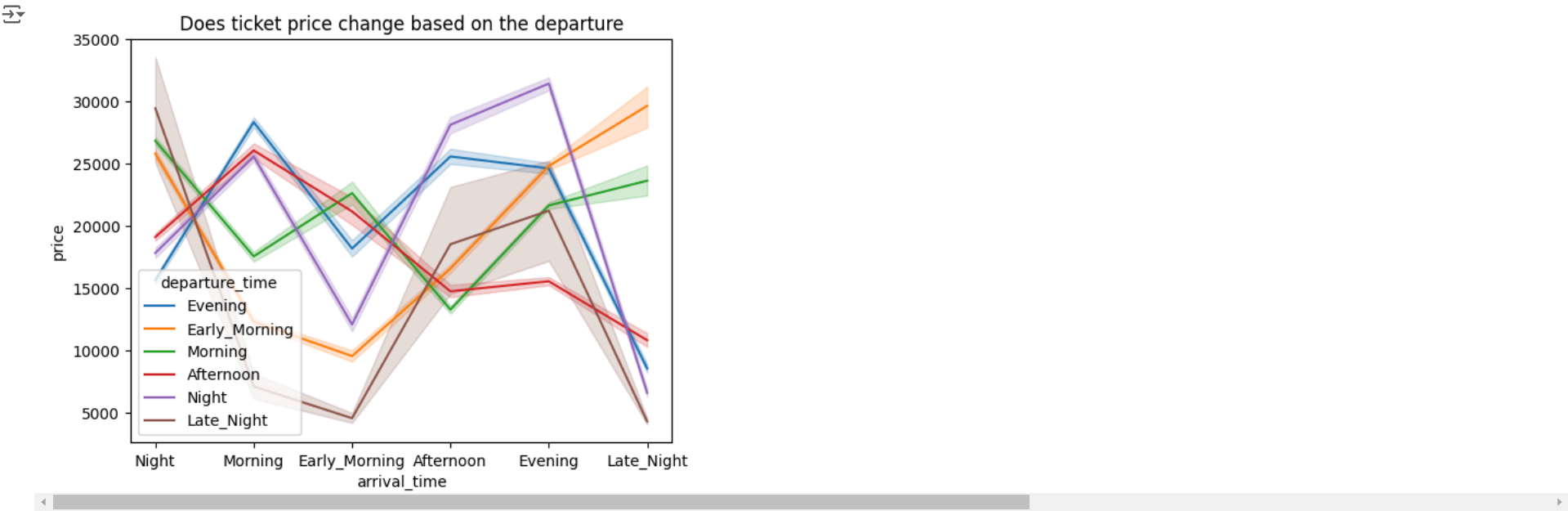
↗ <ipython-input-21-7a6602191895>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(data=Flight_df,x='airline',y='price',palette='rocket')
```

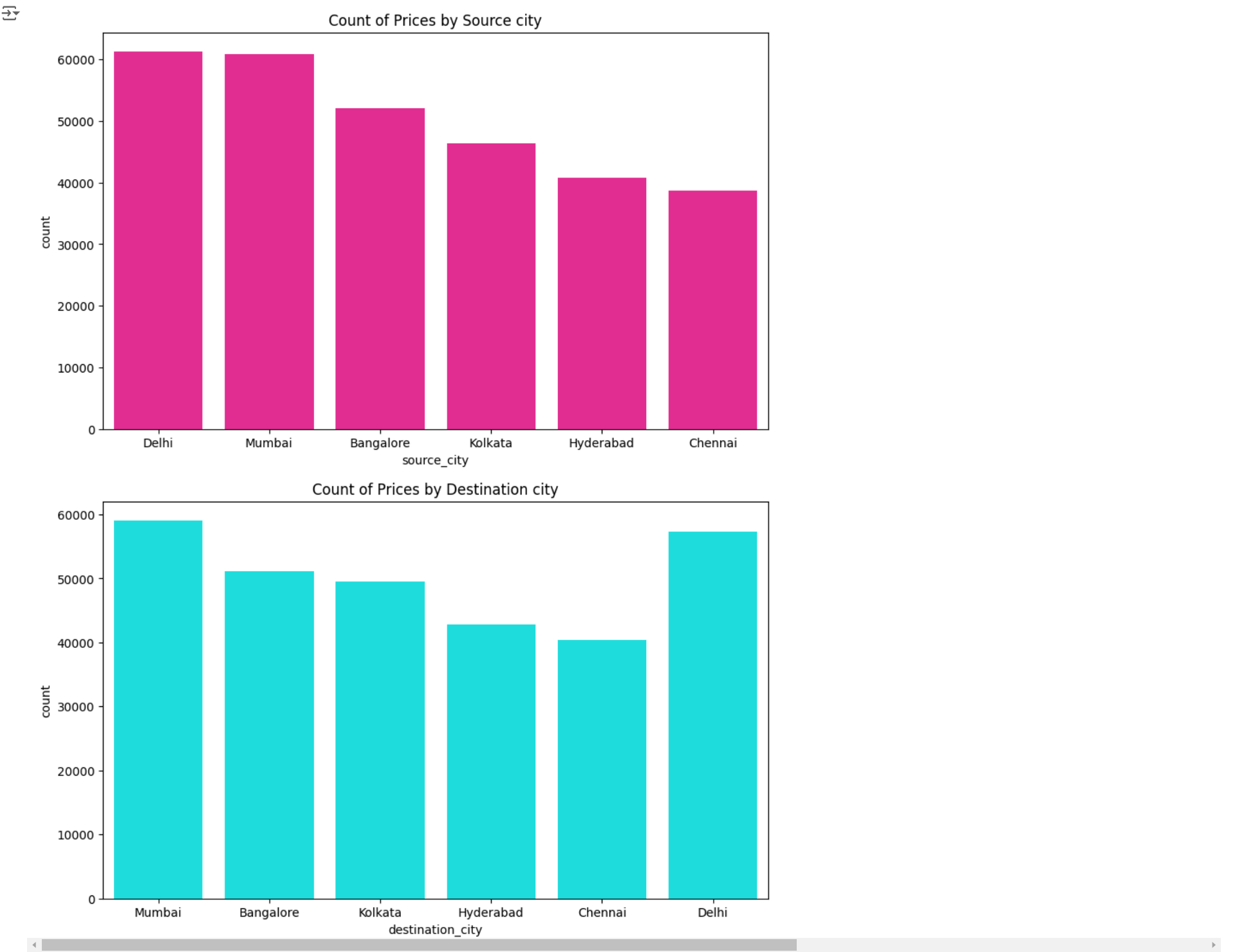


```
#5.Does ticket price change based on the departure time and arrival time using line plot
#Create lineplot
sns.lineplot(data=Flight_df, x='arrival_time',y='price',hue='departure_time')
plt.title('Does ticket price change based on the departure')
plt.show()
```



```
#6.How the price changes with change in Source and Destination city
#Create a countplot for Sorce city
plt.figure(figsize=(10, 6))
#Create Countplot
sns.countplot(data=Flight_df, x='source_city',color='deeppink')
plt.title('Count of Prices by Source city')
plt.show()
```

```
# Create a countplot for Destination city
plt.figure(figsize=(10, 6))
#Create Countplot
sns.countplot(data=Flight_df, x='destination_city',color='cyan')
plt.title('Count of Prices by Destination city')
plt.show()
```

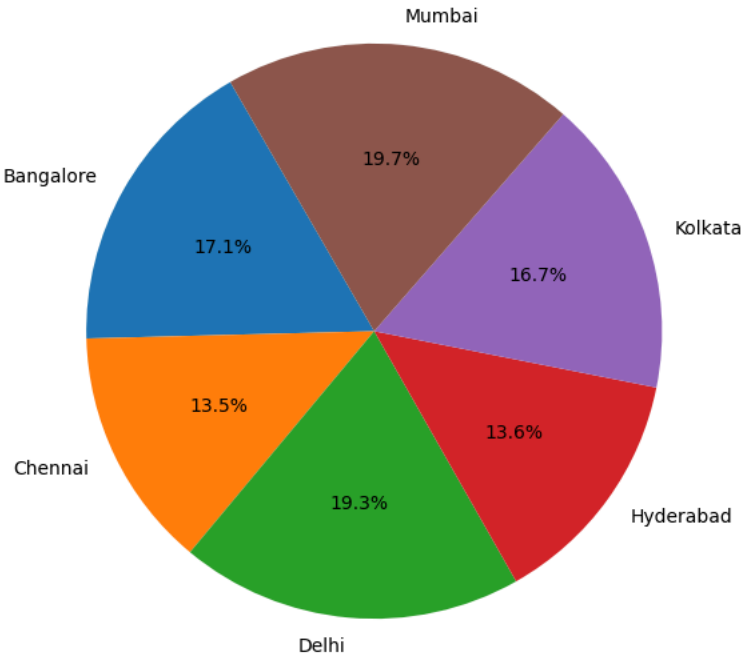


```
#7.Duration of travel vs Source city
Duration=Flight_df.groupby('source_city')['duration'].sum()
plt.figure(figsize=(10, 7))
# Create Piechart
plt.pie(Duration, labels=Duration.index, autopct='%1.1f%%', startangle=120)
plt.title('Duration of Travel vs Source city')
plt.show()
```

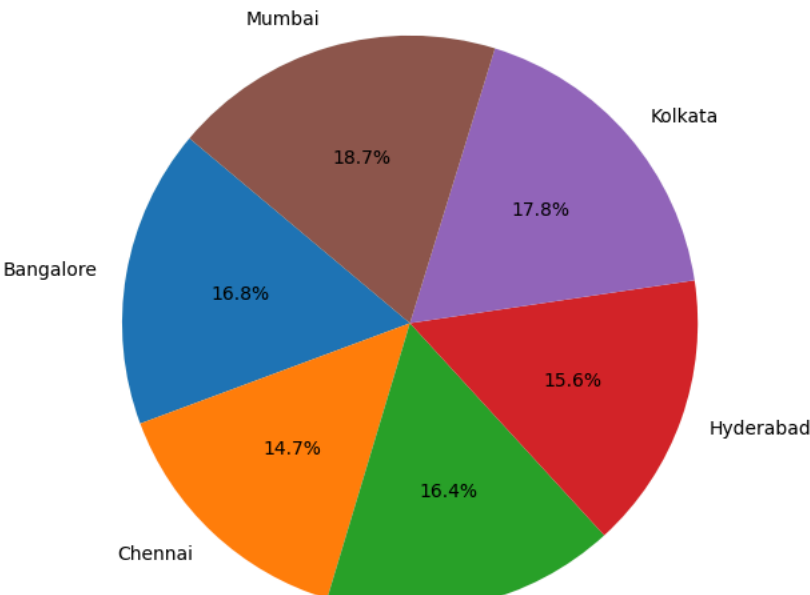
```
#7.Duration of travel vs Destination city
Duration= Flight_df.groupby('destination_city')['duration'].sum()
plt.figure(figsize=(10, 7))
# Create Piechart
plt.pie(Duration, labels=Duration.index, autopct='%1.1f%%', startangle=140)
plt.title('Duration of Travel vs Destination city')
plt.show()
```



Duration of Travel vs Source city



Duration of Travel vs Destination city



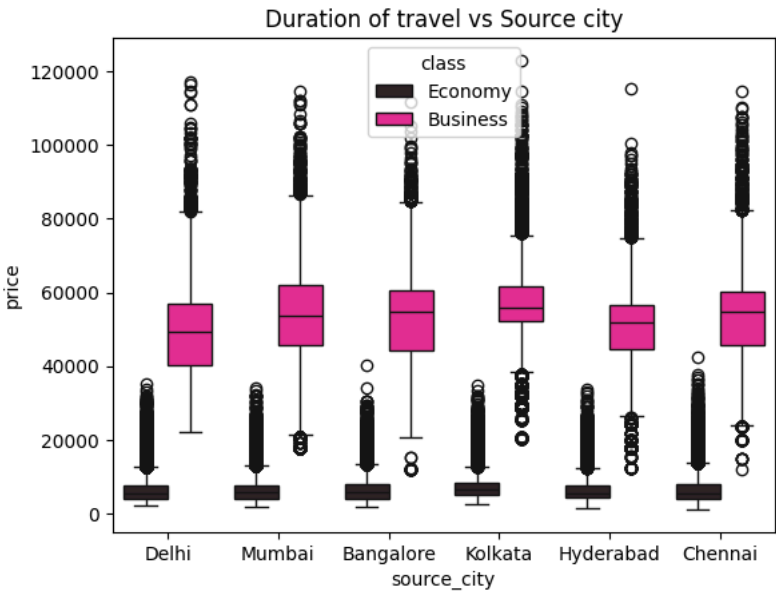
```
#8.Show the high price with class type for Source city
#Create box plot
sns.boxplot(data=Flight_df, x='source_city',y='price',hue='class',color='deeppink')
plt.title('Duration of travel vs Source city')
plt.show()
```

```
#Show the high price with class type for Destination city
#Create box plot
sns.boxplot(data=Flight_df, x='destination_city',y='price',hue='class',color='deeppink')
plt.title('Duration of travel vs Destination city')
plt.show()
```



```
<ipython-input-19-da88db9c6d80>:3: FutureWarning:
Setting a gradient palette using color= is deprecated and will be removed in v0.14.0. Set `palette='dark:deeppink'` for the same effect.

sns.boxplot(data=Flight_df, x='source_city',y='price',hue='class',color='deeppink')
```



```
<ipython-input-19-da88db9c6d80>:9: FutureWarning:
Setting a gradient palette using color= is deprecated and will be removed in v0.14.0. Set `palette='dark:deeppink'` for the same effect.

sns.boxplot(data=Flight_df, x='destination_city',y='price',hue='class',color='deeppink')
```

