Importing the libraries

In [5]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Dipak Mani

In [6]:

```
# Load the dataset
df = pd.read_excel('flight_price.xlsx')
```

In [7]:

```
df.head()
```

Out[7]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Pr
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13

In [8]:

```
## Shape of the DataSet
df.shape
```

Out[8]:

```
(10683, 11)
```

In [9]:

```
## Summary of the DataSet
df.describe()
```

Out[9]:

	Price
count	10683.000000
mean	9087.064121
std	4611.359167
min	1759.000000
25%	5277.000000
50%	8372.000000
75%	12373.000000
max	79512.000000

In [10]:

```
# Check null and Dtypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64 
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

Exploring Data

In [11]:

```
# define numerical & categorical columns
numeric_features=[feature for feature in df.columns if df[feature].dtype != 'O']
categorical_features=[feature for feature in df.columns if df[feature].dtype == 'O']

#print columns
print(f'We have {len(numeric_features)} numerical features :{numeric_features}')
print(f'We have {len(categorical_features)} categorical features :{categorical_features}'
)
```

```
We have 1 numerical features :['Price']
We have 10 categorical features :['Airline', 'Date_of_Journey', 'Source', 'Destination',
'Route', 'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops', 'Additional_Info']
```

Feature Information Airline: Name of the Airline from which the Ticket is Booked. Date_of_Journey: Date of Journey of the Traveller. Source: Source from which the Airline Would Departure. Destination: Destination to Which Airline Would Arrive. Route: Route of the Airline from Source to Destination. Dep_Time: Time at which Flight Would Departure from the Source. Arrival_Time: Time at which Flight Would Arrive at the Destination. Duration: Duration that Airline Takes to fly from Source to Destination. Total_Stops: Total No of Stops that Airline takes Between Source and Destination. Additional_Info: Any Additional Info about the Airline. Price: Fare of the Ticket to fly from Source to Destination.

In [12]:

In [12]:

```
# proportion of count data of each categorical columns
for col in categorical_features:
    print(df[col].value_counts(normalize=True)*100)
    print('----------------------------')
```

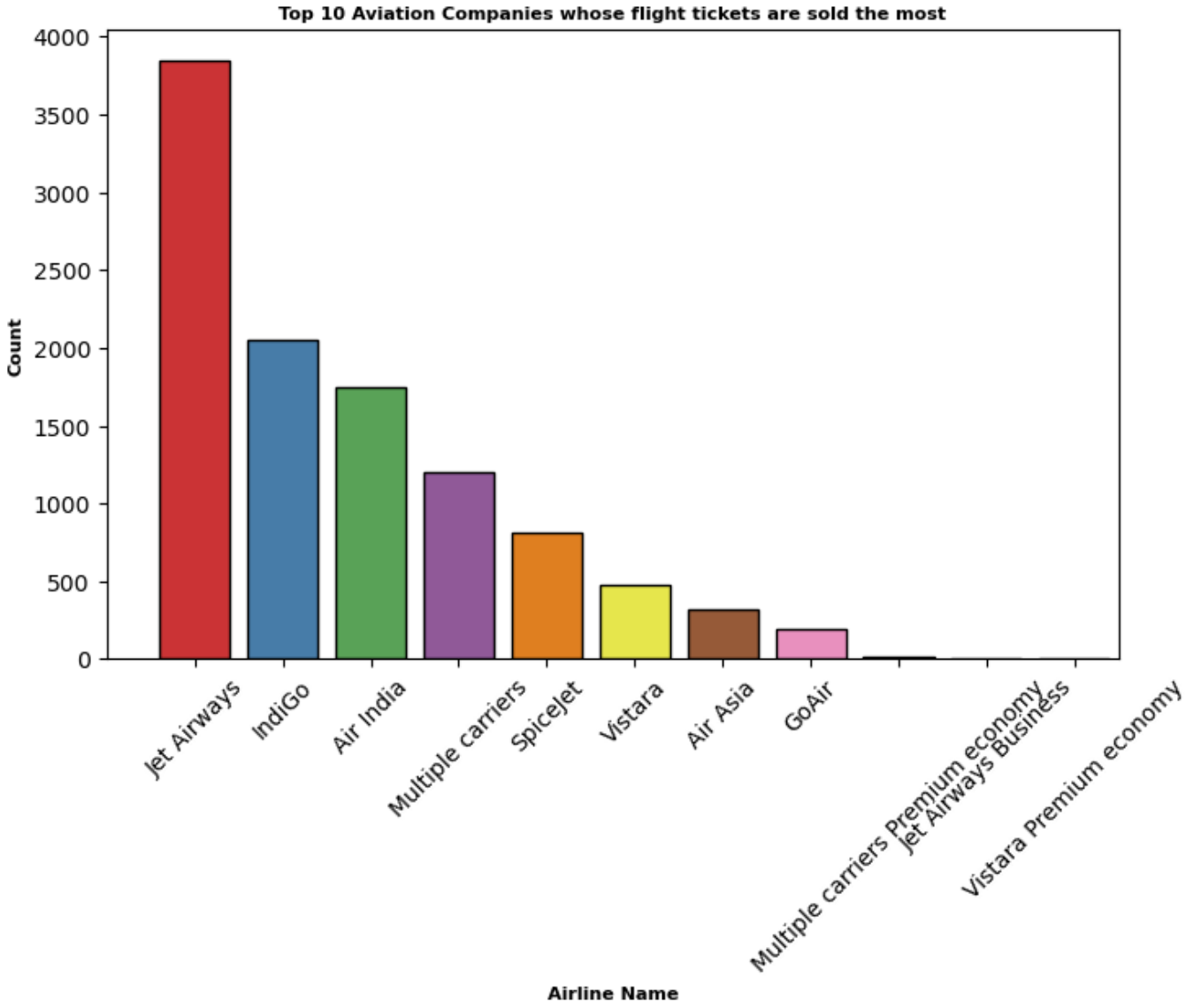
```
Jet Airways                          36.029205
IndiGo                               19.217448
Air India                            16.399888
Multiple carriers                    11.195357
SpiceJet                              7.657025
Vistara                               4.483759
Air Asia                              2.986053
GoAir                                 1.815969
Multiple carriers Premium economy     0.121689
Jet Airways Business                  0.056164
Vistara Premium economy               0.028082
Trujet                                0.009361
Name: Airline, dtype: float64
----------------------------
18/05/2019    4.717776
6/06/2019     4.708415
21/05/2019    4.652251
9/06/2019     4.633530
12/06/2019    4.614809
9/05/2019     4.530563
21/03/2019    3.959562
15/05/2019    3.791070
27/05/2019    3.575775
27/06/2019    3.323037
24/06/2019    3.285594
1/06/2019     3.201348
3/06/2019     3.117102
15/06/2019    3.070299
24/03/2019    3.023495
6/03/2019     2.883085
27/03/2019    2.798839
24/05/2019    2.677151
6/05/2019     2.639708
1/05/2019     2.592905
12/05/2019    2.424413
1/04/2019     2.405691
3/03/2019     2.040625
9/03/2019     1.872133
15/03/2019    1.516428
18/03/2019    1.460264
01/03/2019    1.422821
12/03/2019    1.329215
9/04/2019     1.170083
3/04/2019     1.029673
21/06/2019    1.020313
18/06/2019    0.982870
09/03/2019    0.954788
6/04/2019     0.936067
03/03/2019    0.907985
06/03/2019    0.889263
27/04/2019    0.879903
24/04/2019    0.861181
3/05/2019     0.842460
15/04/2019    0.833099
21/04/2019    0.767575
18/04/2019    0.627165
12/04/2019    0.589722
1/03/2019     0.439951
Name: Date_of_Journey, dtype: float64
----------------------------
Delhi       42.469344
Kolkata     26.874473
Banglore    20.565384
Mumbai       6.524385
Chennai      3.566414
Name: Source, dtype: float64
```

```
---------------------------
Cochin       42.469344
Banglore     26.874473
Delhi        11.841243
New Delhi     8.724141
Hyderabad     6.524385
Kolkata       3.566414
Name: Destination, dtype: float64
---------------------------
DEL → BOM → COK          22.243026
BLR → DEL                14.529114
CCU → BOM → BLR           9.164950
CCU → BLR                 6.777757
BOM → HYD                 5.813518
                           ...
CCU → VTZ → BLR           0.009362
CCU → IXZ → MAA → BLR     0.009362
BOM → COK → MAA → HYD     0.009362
BOM → CCU → HYD           0.009362
BOM → BBI → HYD           0.009362
Name: Route, Length: 128, dtype: float64
---------------------------
18:55    2.181035
17:00    2.124871
07:05    1.918937
10:00    1.900215
07:10    1.890855
          ...
16:25    0.009361
01:35    0.009361
21:35    0.009361
04:15    0.009361
03:00    0.009361
Name: Dep_Time, Length: 222, dtype: float64
---------------------------
19:00           3.959562
21:00           3.369840
19:15           3.117102
16:10           1.441543
12:35           1.142001
                  ...
00:25 02 Jun    0.009361
08:55 13 Mar    0.009361
11:05 19 May    0.009361
12:30 22 May    0.009361
21:20 13 Mar    0.009361
Name: Arrival_Time, Length: 1343, dtype: float64
---------------------------
2h 50m     5.148367
1h 30m     3.613217
2h 45m     3.154545
2h 55m     3.154545
2h 35m     3.079659
            ...
31h 30m    0.009361
30h 25m    0.009361
42h 5m     0.009361
4h 10m     0.009361
47h 40m    0.009361
Name: Duration, Length: 368, dtype: float64
---------------------------
1 stop      52.658678
non-stop    32.681146
2 stops     14.229545
3 stops      0.421269
4 stops      0.009362
Name: Total_Stops, dtype: float64
---------------------------
No info                         78.114762
In-flight meal not included     18.552841
No check-in baggage included     2.995413
1 Long layover                   0.177853
```

```
Change airports                 0.065525
Business class                  0.037443
No Info                         0.028082
1 Short layover                 0.009361
Red-eye flight                  0.009361
2 Long layover                  0.009361
Name: Additional_Info, dtype: float64
----------------------------
```

In [13]:

```
# Top 10 Aviation companies whoose flight tickets are sold most
plt.subplots(figsize=(8,5))
sns.countplot(x="Airline", data=df,ec = "black",palette="Set1",order = df['Airline'].val
ue_counts().index)
plt.title("Top 10 Aviation Companies whose flight tickets are sold the most", weight="bol
d",fontsize=8, pad=5)
plt.ylabel("Count", weight="bold", fontsize=8)
plt.xlabel("Airline Name", weight="bold", fontsize=8)
plt.xticks(rotation= 45)
plt.xlim(-1,10.5)
plt.show()
```



Check mean price of Jet Airways whose flight tickets are sold the most¶

In [14]:

```
jet_airways = df[df['Airline'] == 'Jet Airways']['Price'].mean()
print(f'The mean price of Jet Airways Flight Tickets  is {jet_airways:.2f} Rupees')
```

```
The mean price of Jet Airways Flight Tickets  is 11643.92 Rupees
```

Costliest Aviation Companies and Costliest Flight Tickets

In [15]:

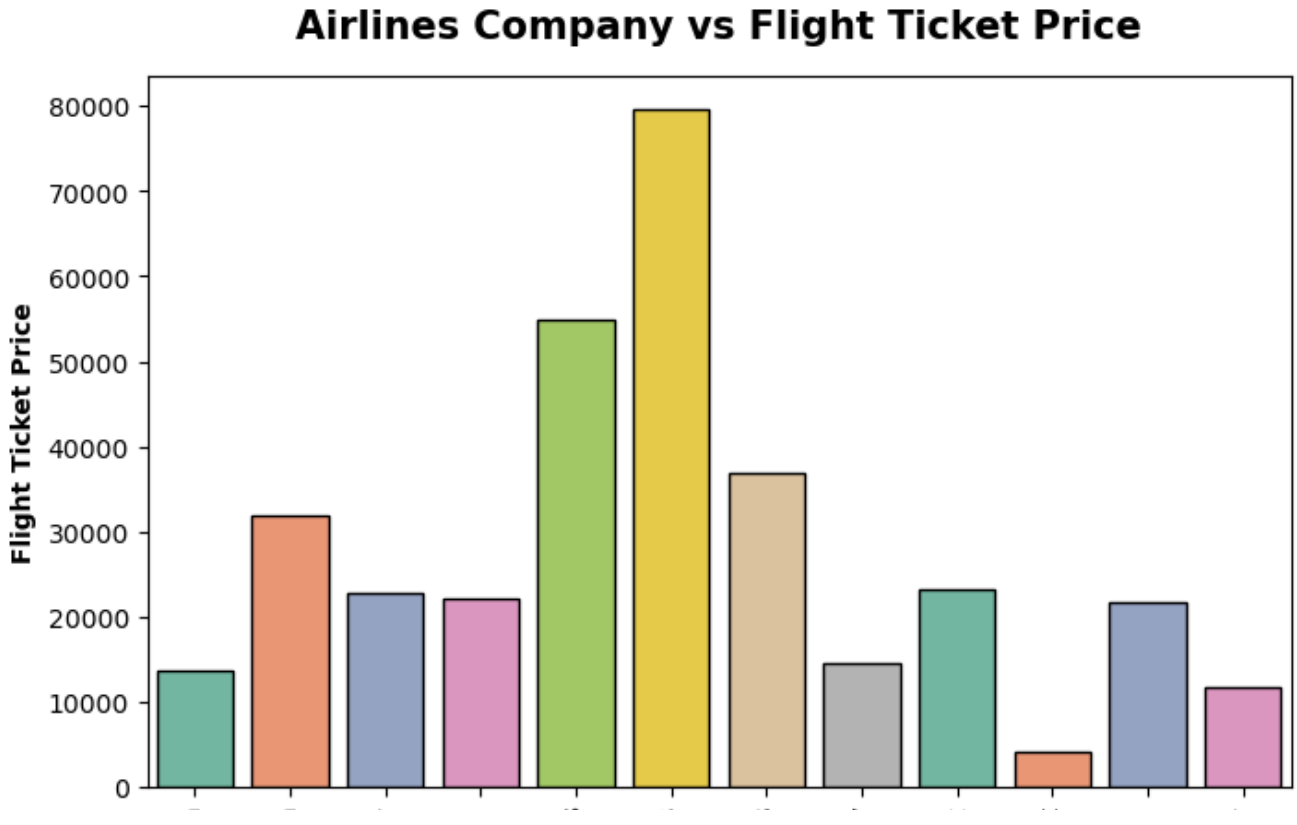
```
aviation_company_airline = df.groupby('Airline').Price.max()
aviation_company= aviation_company_airline.to_frame().sort_values('Price',ascending=False
)[0:10]
aviation_company
```

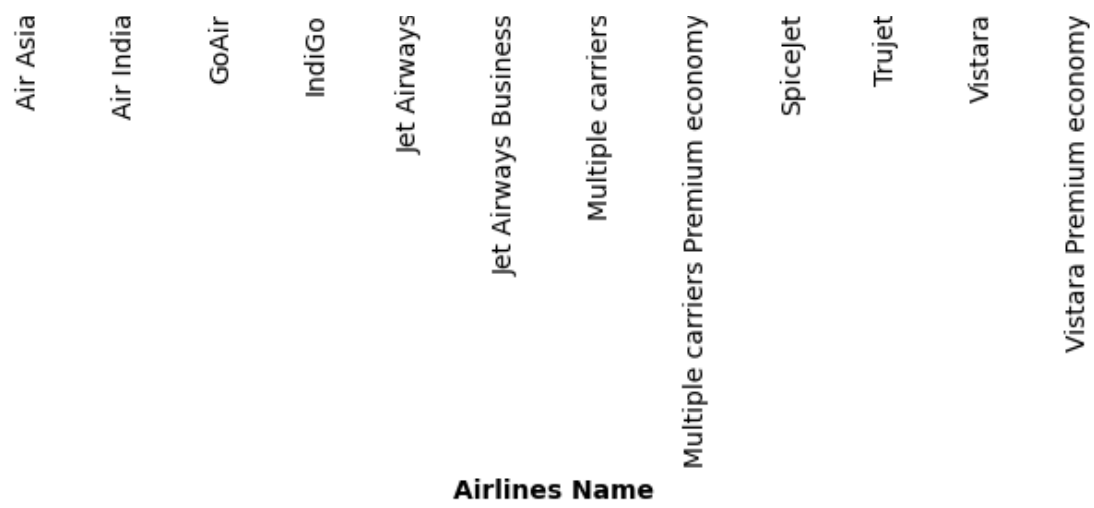
Out[15]:

Airline	Price
Jet Airways Business	79512
Jet Airways	54826
Multiple carriers	36983
Air India	31945
SpiceJet	23267
GoAir	22794
IndiGo	22153
Vistara	21730
Multiple carriers Premium economy	14629
Air Asia	13774

In [16]:

```
# Graph Airlines Companies Vs Flight Ticket Price
plt.subplots(figsize=(8,5))
sns.barplot(x=aviation_company_airline.index, y=aviation_company_airline.values,ec = "bl
ack",palette="Set2")
plt.title("Airlines Company vs Flight Ticket Price", weight="bold",fontsize=15, pad=15)
plt.ylabel("Flight Ticket Price", weight="bold", fontsize=10)
plt.xlabel("Airlines Name", weight="bold", fontsize=10)
plt.xticks(rotation=90)
plt.show()
```





In [17]:

```
# Airline is a categorical feature
df['Airline'].unique()
```

Out[17]:

```
array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
       'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
       'Vistara Premium economy', 'Jet Airways Business',
       'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

In [18]:

```
df['Date_of_Journey']=pd.to_datetime(df['Date_of_Journey'],infer_datetime_format=True)
```

In [19]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   Airline          10683 non-null  object        
 1   Date_of_Journey  10683 non-null  datetime64[ns]
 2   Source           10683 non-null  object        
 3   Destination      10683 non-null  object        
 4   Route            10682 non-null  object        
 5   Dep_Time         10683 non-null  object        
 6   Arrival_Time     10683 non-null  object        
 7   Duration         10683 non-null  object        
 8   Total_Stops      10682 non-null  object        
 9   Additional_Info  10683 non-null  object        
 10  Price            10683 non-null  int64         
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 918.2+ KB
```

In [20]:

```
df.head(2)
```

Out[20]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Pric
0	IndiGo	2019-03-24	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	389
1	Air India	2019-05-01	Kolkata	Banglore	CCU → IXR → BBI →	05:50	13:15	7h 25m	2 stops	No info	766

Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Pric

In [21]:

```
# Split the Date_of_Journey columns into separately as Day, Month, Year
df['Day']=df['Date_of_Journey'].dt.day
df['Month']=df['Date_of_Journey'].dt.month
df['Year']=df['Date_of_Journey'].dt.year
```

In [22]:

```
df.head()
```

Out[22]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	P
0	IndiGo	2019-03-24	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3
1	Air India	2019-05-01	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7
2	Jet Airways	2019-06-09	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13
3	IndiGo	2019-05-12	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6
4	IndiGo	2019-03-01	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13

In [23]:

```
# Drop Date_of_Journey column
df.drop('Date_of_Journey',axis=1,inplace=True)
```

In [24]:

```
df.head()
```

Out[24]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Day	Month
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5
	Jet			DEL → LKO								

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Day	Month
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	9	6
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	12	5
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	1	3

 # Extract day,month,year from the string df['Date']=df['Date_of_Journey'].apply(lambda x:x.split("/")[0]) df['Month']=df['Date_of_Journey'].apply(lambda x:x.split("/")[1]) df['Year']=df['Date_of_Journey'].apply(lambda x:x.split("/")[2])

In [25]:

```
df.head(2)
```

Out[25]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Day	Month	Y
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	2
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5	2

In [26]:

```
# Feature Engineering Process
# Split the Dep_Time column into two columns Dept_Hour and Dept_Min
df['Dept_Hour']=df['Dep_Time'].str.split(':').str[0]
df['Dept_Min']=df['Dep_Time'].str.split(':').str[1]
```

In [27]:

```
df.head()
```

Out[27]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Day	Month
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	9	6
3	IndiGo	Kolkata	Banglore	CCU →								

3	IndiGo	Kolkata	Banglore	NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6318	12	5
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	1	3

In [28]:

```
# Drop Dep_Time feature
df.drop('Dep_Time',axis=1,inplace=True)
```

In [29]:

```
# More information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Source           10683 non-null  object
 2   Destination      10683 non-null  object
 3   Route            10682 non-null  object
 4   Arrival_Time     10683 non-null  object
 5   Duration         10683 non-null  object
 6   Total_Stops      10682 non-null  object
 7   Additional_Info  10683 non-null  object
 8   Price            10683 non-null  int64 
 9   Day              10683 non-null  int64 
 10  Month            10683 non-null  int64 
 11  Year             10683 non-null  int64 
 12  Dept_Hour        10683 non-null  object
 13  Dept_Min         10683 non-null  object
dtypes: int64(4), object(10)
memory usage: 1.1+ MB
```

In [30]:

```
# Convert Object column into integer(int)
df['Dept_Hour']=df['Dept_Hour'].astype(int)
df['Dept_Min']=df['Dept_Min'].astype(int)
```

In [31]:

```
df.head(2)
```

Out[31]:

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_H
0	IndiGo	Banglore	New Delhi	BLR → DEL	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	2019	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	13:15	7h 25m	2 stops	No info	7662	1	5	2019	

In [32]:

```
# Arrival_Time column splits in Arrival_hour and Arrival_min
df['Arrival_Time']=df['Arrival_Time'].apply(lambda x: x.split(' ')[0])
```

In [33]:

```
df['Arrival_hour']=df['Arrival_Time'].str.split(':').str[0]
df['Arrival_min']=df['Arrival_Time'].str.split(':').str[1]
```

In [34]:

```
# Type change object into int
df['Arrival_hour']=df['Arrival_hour'].astype(int)
df['Arrival_min']=df['Arrival_min'].astype(int)
```

In [35]:

```
# Drop arrival Time column
df.drop('Arrival_Time',axis=1,inplace=True)
```

In [36]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Source           10683 non-null  object
 2   Destination      10683 non-null  object
 3   Route            10682 non-null  object
 4   Duration         10683 non-null  object
 5   Total_Stops      10682 non-null  object
 6   Additional_Info  10683 non-null  object
 7   Price            10683 non-null  int64 
 8   Day              10683 non-null  int64 
 9   Month            10683 non-null  int64 
 10  Year             10683 non-null  int64 
 11  Dept_Hour        10683 non-null  int32 
 12  Dept_Min         10683 non-null  int32 
 13  Arrival_hour     10683 non-null  int32 
 14  Arrival_min      10683 non-null  int32 
dtypes: int32(4), int64(4), object(7)
memory usage: 1.1+ MB
```

In [37]:

```
# Total stop column
df['Total_Stops'].unique()
```

Out[37]:

```
array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
      dtype=object)
```

In [38]:

```
# Categorical values divide
df['Total_Stops'].value_counts()
```

Out[38]:

```
1 stop      5625
non-stop    3491
2 stops     1520
3 stops       45
4 stops        1
Name: Total_Stops, dtype: int64
```

In [39]:

```
# Null value present in thr Total_Stops column
```

```
df['Total_Stops'].isnull().sum()
```

Out[39]:

```
1
```

In [40]:

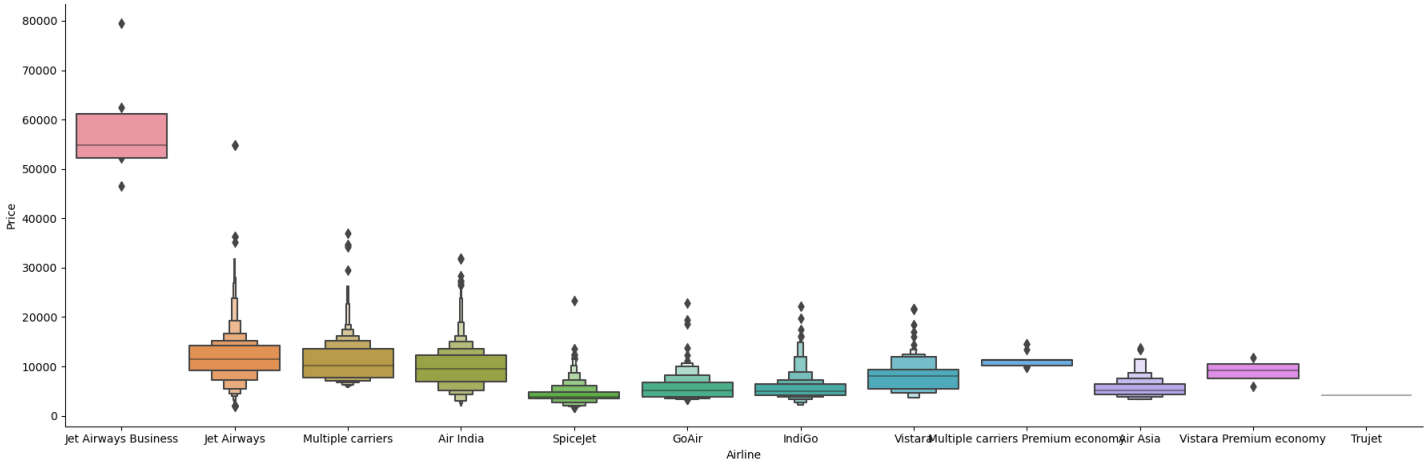
```
# Row of null value
df[df['Total_Stops'].isnull()]
```

Out[40]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_M
9039	Air India	Delhi	Cochin	NaN	23h 40m	NaN	No info	7480	6	5	2019	9	

In [41]:

```
# CatPlot between Airline vs Price
sns.catplot(y = "Price", x = "Airline", data = df.sort_values("Price", ascending = False
), kind="boxen", height = 6, aspect = 3)
plt.show()
```



In [42]:

```
# Mapping technique use for convert categorical to numerical
df['Total_Stops']=df['Total_Stops'].map({'non-stop':0,'1 stop':1,'2 stops':2,'3 stops':3
,'4 stops':4,'nan':1})
```

In [43]:

```
df.head()
```

Out[43]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_I
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	0.0	No info	3897	24	3	2019	22	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2.0	No info	7662	1	5	2019	5	
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	19h	2.0	No info	13882	9	6	2019	9	

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_I
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	5h 25m	1.0	No info	6218	12	5	2019	18	
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	4h 45m	1.0	No info	13302	1	3	2019	16	

In [44]:

```
# Split the duration_hour column
df['duration_hour']=df['Duration'].str.split(' ').str[0].str.split('h').str[0]
```

In [45]:

```
df.head()
```

Out[45]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_I
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	0.0	No info	3897	24	3	2019	22	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2.0	No info	7662	1	5	2019	5	
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	19h	2.0	No info	13882	9	6	2019	9	
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	5h 25m	1.0	No info	6218	12	5	2019	18	
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	4h 45m	1.0	No info	13302	1	3	2019	16	

In [46]:

```
df[df['duration_hour']=='5m']
```

Out[46]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_
6474	Air India	Mumbai	Hyderabad	BOM → GOI → PNQ → HYD	5m	2.0	No info	17327	6	3	2019	16	

In [47]:

```
#Drop row number is 6474
df.drop(6474,axis=0,inplace=True)
```

In [48]:

```
df['duration_min']=df['Duration'].str.split(' ').str[1].str.split('m').str[0]
```

In [49]:

```
df['duration_min']
```

Out[49]:

```
0          50
1          25
2         NaN
3          25
4          45
         ...
10678      30
10679      35
10680     NaN
10681      40
10682      20
Name: duration_min, Length: 10682, dtype: object
```

In [50]:

```
# Fill null values with 0
df['duration_min']=df['duration_min'].fillna(0)
```

In [51]:

```
# Drop Duration
df.drop('Duration',axis=1,inplace=True)
```

In [52]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10682 entries, 0 to 10682
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10682 non-null  object
 1   Source           10682 non-null  object
 2   Destination      10682 non-null  object
 3   Route            10681 non-null  object
 4   Total_Stops      10681 non-null  float64
 5   Additional_Info  10682 non-null  object
 6   Price            10682 non-null  int64 
 7   Day              10682 non-null  int64 
 8   Month            10682 non-null  int64 
 9   Year             10682 non-null  int64 
 10  Dept_Hour        10682 non-null  int32 
 11  Dept_Min         10682 non-null  int32 
 12  Arrival_hour     10682 non-null  int32 
 13  Arrival_min      10682 non-null  int32 
 14  duration_hour    10682 non-null  object
 15  duration_min     10682 non-null  object
dtypes: float64(1), int32(4), int64(4), object(7)
memory usage: 1.2+ MB
```

In [53]:

```
# Change dtype object into int
df['duration_hour']=df['duration_hour'].astype(int)
df['duration_min']=df['duration_min'].astype(int)
```

In [54]:

```
"Dipak Mani".split(" ")
```

Out[54]:

```
['Dipak', 'Mani']
```

In [55]:

```
"Dipak Mani".split(" ")[0]
```

Out[55]:

```
'Dipak'
```

In [56]:

```
"Dipak Mani".split(" ")[1]
```

Out[56]:

```
'Mani'
```

In [57]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10682 entries, 0 to 10682
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10682 non-null  object
 1   Source           10682 non-null  object
 2   Destination      10682 non-null  object
 3   Route            10681 non-null  object
 4   Total_Stops      10681 non-null  float64
 5   Additional_Info  10682 non-null  object
 6   Price            10682 non-null  int64 
 7   Day              10682 non-null  int64 
 8   Month            10682 non-null  int64 
 9   Year             10682 non-null  int64 
 10  Dept_Hour        10682 non-null  int32 
 11  Dept_Min         10682 non-null  int32 
 12  Arrival_hour     10682 non-null  int32 
 13  Arrival_min      10682 non-null  int32 
 14  duration_hour    10682 non-null  int32 
 15  duration_min     10682 non-null  int32 
dtypes: float64(1), int32(6), int64(4), object(5)
memory usage: 1.1+ MB
```

In [58]:

```
# Unique
df.Airline.unique()
```

Out[58]:

```
array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
       'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
       'Vistara Premium economy', 'Jet Airways Business',
       'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

In [59]:

```
df.head()
```

Out[59]:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_Min	Arriva
				BLR									

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_Min	Arriva
0	IndiGo	Banglore	New Delhi	BLR → DEL	0.0	No info	3897	24	3	2019	22	20	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2.0	No info	7662	1	5	2019	5	50	
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	2.0	No info	13882	9	6	2019	9	25	
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	1.0	No info	6218	12	5	2019	18	5	
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	1.0	No info	13302	1	3	2019	16	50	

In [60]:

```
# Target guided ordinal

df.groupby('Airline')['Price'].mean().sort_values()
```

Out[60]:

```
Airline
Trujet                                4140.000000
SpiceJet                              4338.284841
Air Asia                              5590.260188
IndiGo                                5673.682903
GoAir                                 5861.056701
Vistara                               7796.348643
Vistara Premium economy               8962.333333
Air India                             9606.804112
Multiple carriers                    10902.678094
Multiple carriers Premium economy    11418.846154
Jet Airways                          11643.923357
Jet Airways Business                 58358.666667
Name: Price, dtype: float64
```

In [61]:

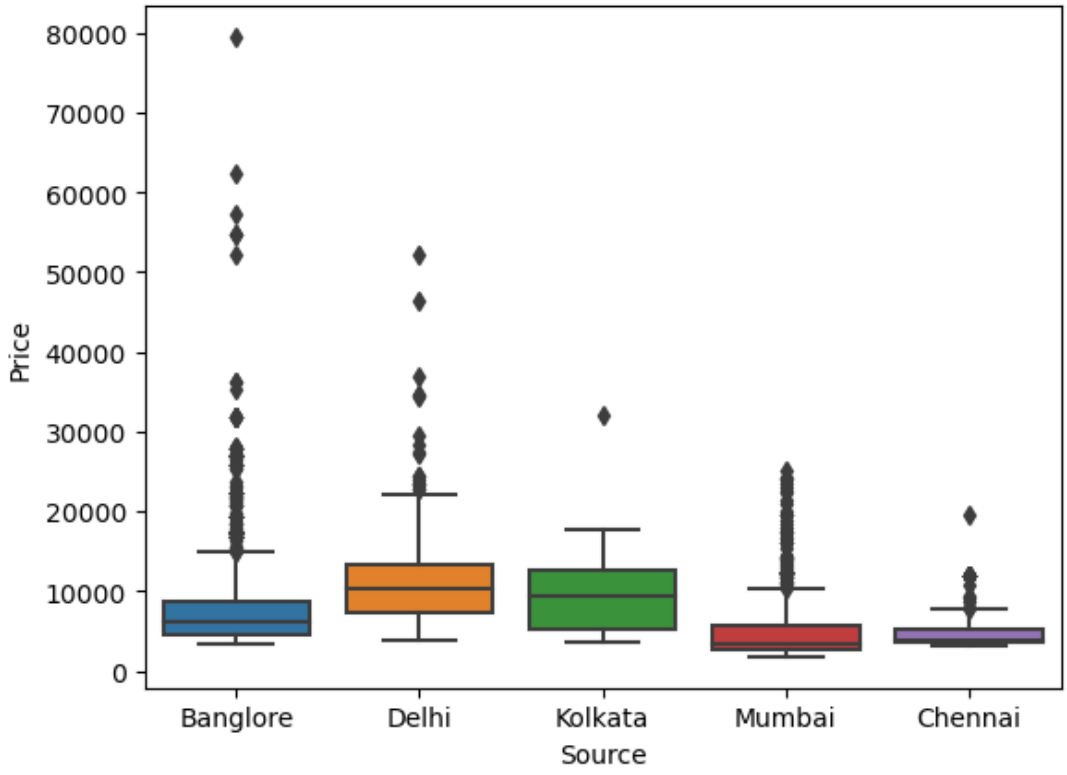
```
#OneHotEncoding -----> Nominal data
Airline = df[["Airline"]]
Airline = pd.get_dummies(df['Airline'],drop_first=False)
Airline.head()
```

Out[61]:

	Air Asia	Air India	GoAir	IndiGo	Jet Airways	Jet Airways Business	Multiple carriers	Multiple carriers Premium economy	SpiceJet	Trujet	Vistara	Vistara Premium economy
0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0

In [62]:

```
sns.boxplot(y = "Price", x = "Source", data = df.sort_values("Price", ascending = False)
)
plt.show()
```



In [63]:

```
from sklearn.preprocessing import OneHotEncoder
```

In [64]:

```
ohe = OneHotEncoder()
```

In [65]:

```
ohe.fit_transform(df[['Airline']]).toarray()
```

Out[65]:

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 1., 0., ..., 0., 0., 0.]])
```

In [66]:

```
## Replacing target guided ordinal encoding
def replace_airline_with_mean(df):
    mean_prices = df.groupby('Airline')['Price'].mean().sort_values()
    df['Airline'] = df['Airline'].apply(lambda x: mean_prices[x])
    return df

df = replace_airline_with_mean(df)
df.head()
```

Out[66]:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_Min
0	5673.682903	Banglore	New Delhi	BLR →	0.0	No info	3897	24	3	2019	22	20

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	Day	Month	Year	Dept_Hour	Dept_Min
1	9606.804112	Kolkata	Banglore	CCU → IXR → BBI → BLR	2.0	No info	7662	1	5	2019	5	50
2	11643.923357	Delhi	Cochin	DEL → LKO → BOM → COK	2.0	No info	13882	9	6	2019	9	25
3	5673.682903	Kolkata	Banglore	CCU → NAG → BLR	1.0	No info	6218	12	5	2019	18	5
4	5673.682903	Banglore	New Delhi	BLR → NAG → DEL	1.0	No info	13302	1	3	2019	16	50

In [67]:

```
pd.DataFrame(ohe.fit_transform(df[['Airline']]).toarray(),columns=ohe.get_feature_names()
)
```

Out[67]:

	x0_4140.0	x0_4338.284841075794	x0_5590.260188087775	x0_5673.68290306868	x0_5861.056701030928	x0_7796.34864300
0	0.0	0.0	0.0	1.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	1.0	0.0	
4	0.0	0.0	0.0	1.0	0.0	
...	...	...	...	...	...	
10677	0.0	0.0	1.0	0.0	0.0	
10678	0.0	0.0	0.0	0.0	0.0	
10679	0.0	0.0	0.0	0.0	0.0	
10680	0.0	0.0	0.0	0.0	0.0	
10681	0.0	0.0	0.0	0.0	0.0	

10682 rows × 12 columns