AIRLINE ANALYSIS : EDA

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
flight_price = pd.read_excel('flight_price.xlsx')
flight_price.head()
```

Out[5]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

In [6]:
```
df = flight_price.copy()
```

In [7]:
```
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
```

In [8]:
```
df.isna().sum()
```

Out[8]:
```
Airline           0
Date_of_Journey   0
Source            0
Destination       0
Route             1
Dep_Time          0
Arrival_Time      0
Duration          0
Total_Stops       1
Additional_Info   0
Price             0
dtype: int64
```

In [9]:
```
df.head()
```

Out[9]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

a) Date_of_Journey - Separate Date into date, month, year and type cast to int

In [10]:
```
# Separating

df['Date'] = df['Date_of_Journey'].str.split('/').str[0]
df['Month'] = df['Date_of_Journey'].str.split('/').str[1]
df['Year'] = df['Date_of_Journey'].str.split('/').str[2]

df.head(2)
```

Out[10]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	03	2019
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	05	2019

In [11]:
```
# Type casting

df['Date'] = df['Date'].astype(int)
df['Month'] = df['Month'].astype(int)
df['Year'] = df['Year'].astype(int)
```

In [12]:
```
df.dtypes
```

Out[12]:
```
Airline            object
Date_of_Journey    object
Source             object
Destination        object
Route              object
Dep_Time           object
Arrival_Time       object
Duration           object
Total_Stops        object
Additional_Info    object
Price               int64
Date                int64
Month               int64
Year                int64
dtype: object
```

In [13]:
```
# Drop Date_of_Journey Column

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

In [14]:
```
df.head(2)
```

Out[14]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	2019
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5	2019

b) Arrival_Time - Separate Hours and Minutes, typecast to int

In [15]:
```
# Remove extra dates after time

df['Arrival_Time'] = df['Arrival_Time'].str.split(' ').str[0]
```

In [16]:
```
df.head(2)
```

Out[16]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10	2h 50m	non-stop	No info	3897	24	3	2019
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5	2019

In [17]:
```
df['Arrival_Hour'] = df['Arrival_Time'].str.split(':').str[0]
df['Arrival_Minute'] = df['Arrival_Time'].str.split(':').str[1]
```

In [18]:
```
df.head(2)
```

Out[18]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10	2h 50m	non-stop	No info	3897	24	3	2019	01	10
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5	2019	13	15

In [19]:
```
# Typecast
```

```
df['Arrival_Hour'] = df['Arrival_Hour'].astype(int)
df['Arrival_Minute'] = df['Arrival_Minute'].astype(int)
```

In [20]:
```
df.dtypes
```

Out[20]:
```
Airline           object
Source            object
Destination       object
Route             object
Dep_Time          object
Arrival_Time      object
Duration          object
Total_Stops       object
Additional_Info   object
Price              int64
Date               int64
Month              int64
Year               int64
Arrival_Hour       int64
Arrival_Minute     int64
dtype: object
```

In [21]:
```
# Drop Arrival Time Column

df.drop('Arrival_Time', axis=1, inplace=True)
df.head(2)
```

Out[21]:

	Airline	Source	Destination	Route	Dep_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	2h 50m	non-stop	No info	3897	24	3	2019	1	10
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	7h 25m	2 stops	No info	7662	1	5	2019	13	15

c) Dep_Time - Separate Hours and Minutes, typecast to int

In [22]:
```
df['Dep_Hour'] = df['Dep_Time'].str.split(':').str[0]
df['Dep_Min'] = df['Dep_Time'].str.split(':').str[1]
```

In [23]:
```
df.head(2)
```

Out[23]:

	Airline	Source	Destination	Route	Dep_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute	Dep_Hour	Dep_Min
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	2h 50m	non-stop	No info	3897	24	3	2019	1	10	22	20
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	7h 25m	2 stops	No info	7662	1	5	2019	13	15	05	50

In [24]:
```
# Typecast

df['Dep_Hour'] = df['Dep_Hour'].astype(int)
df['Dep_Min'] = df['Dep_Min'].astype(int)
```

In [25]:
```
df.dtypes
```

Out[25]:
```
Airline            object
Source             object
Destination        object
Route              object
Dep_Time           object
Duration           object
Total_Stops        object
Additional_Info    object
Price               int64
Date                int64
Month               int64
Year                int64
Arrival_Hour        int64
Arrival_Minute      int64
Dep_Hour            int64
Dep_Min             int64
dtype: object
```

In [26]:
```
# Drop Dep_Time column

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

In [27]:
```
df.head(2)
```

Out[27]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute	Dep_Hour	Dep_Min
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24	3	2019	1	10	22	20
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	1	5	2019	13	15	5	50

d) Duration - Separate Hours and Minutes, typecast to int

In [28]:
```
# If Duration has hrs, split it else return 0

df['Duration_hr'] = df['Duration'].apply(lambda x:x.split('h')[0] if 'h' in x else '0')

# If Duration has m replace with blank, else 0

df['Duration_min'] = df['Duration'].apply(lambda x: x.split('h')[-1].replace('m', '') if 'm' in x else '0')
```

In [29]:
```
df.head(2)
```

Out[29]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute	Dep_Hour	Dep_Min	Duration_hr	Duration_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24	3	2019	1	10	22	20	2	50
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	1	5	2019	13	15	5	50	7	25

In [30]:
```
# Typecast

df[['Duration_hr', 'Duration_min']] = df[['Duration_hr', 'Duration_min']].astype(int)
```

In [31]:
```
df.dtypes
```

Out[31]:
```
Airline            object
Source             object
Destination        object
Route              object
Duration           object
Total_Stops        object
Additional_Info    object
Price               int64
Date                int64
Month               int64
Year                int64
Arrival_Hour        int64
Arrival_Minute      int64
Dep_Hour            int64
Dep_Min             int64
Duration_hr         int64
Duration_min        int64
dtype: object
```

In [32]:
```
# Drop Duration

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

In [33]:
```
df.head(2)
```

Out[33]:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute	Dep_Hour	Dep_Min	Duration_hr	Duration_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	non-stop	No info	3897	24	3	2019	1	10	22	20	2	50
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2 stops	No info	7662	1	5	2019	13	15	5	50	7	25

e) Total_Stops - Fill NaN values and label Encode

In [34]:
```
df['Total_Stops'].unique()
```

Out[34]:
```
array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
      dtype=object)
```

In [35]:
```
mode = df['Total_Stops'].mode()[0]
mode
```

Out[35]:
```
'1 stop'
```

In [36]:
```
# Fill Null value using mode

df['Total_Stops'] = df['Total_Stops'].fillna(mode)
```

In [37]:
```
# Label Encode - Convert objects into int using Map

df['Total_Stops'] = df['Total_Stops'].map({'non-stop':0, '1 stop':1, '2 stops':2, '3 stops':3, '4 stops':4})
```

In [38]:
```
df.head(2)
```

Out[38]:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute	Dep_Hour	Dep_Min	Duration_hr	Duration_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	0	No info	3897	24	3	2019	1	10	22	20	2	50
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2	No info	7662	1	5	2019	13	15	5	50	7	25

f) Route - Drop it as Source and Destination is given

In [39]:
```
df.drop('Route', axis=1, inplace=True)
```

In [40]:
```
df.head(2)
```

Out[40]:

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival_Hour	Arrival_Minute	Dep_Hour	Dep_Min	Duration_hr	Duration_min
0	IndiGo	Banglore	New Delhi	0	No info	3897	24	3	2019	1	10	22	20	2	50
1	Air India	Kolkata	Banglore	2	No info	7662	1	5	2019	13	15	5	50	7	25

In [41]:
```
df.dtypes
```

Out[41]:
```
Airline            object
Source             object
Destination        object
Total_Stops         int64
Additional_Info    object
Price               int64
Date                int64
Month               int64
Year                int64
Arrival_Hour        int64
Arrival_Minute      int64
Dep_Hour            int64
Dep_Min             int64
Duration_hr         int64
Duration_min        int64
dtype: object
```

g) Final Check

In [42]:
```
display(df['Destination'].unique())
display(df['Source'].unique())
```

```
array(['New Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Delhi', 'Hyderabad'],
      dtype=object)
array(['Banglore', 'Kolkata', 'Delhi', 'Chennai', 'Mumbai'], dtype=object)
```

In [43]:
```
df['Destination'] = df['Destination'].replace('Delhi', 'New Delhi')
df['Source'] = df['Source'].replace('Delhi', 'New Delhi')
```

In [44]:
```
df = df.drop(df[(df['Duration_hr'] == 0) & (df['Duration_min'] == 5)].index)
```