economy-class-analysis

February 3, 2024

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv('economy.csv')
df.head()
```

[2]:

	date	airline	ch_code	num_code	dep_time	from	time_taken	stop	arr_time	to	price
0	11-02-2022	SpiceJet	SG	8709	18:55	Delhi	02h 10m	non-stop	21:05	Mumbai	5,953
1	11-02-2022	SpiceJet	SG	8157	06:20	Delhi	02h 20m	non-stop	08:40	Mumbai	5,953
2	11-02-2022	AirAsia	I5	764	04:25	Delhi	02h 10m	non-stop	06:35	Mumbai	5,956
3	11-02-2022	Vistara	UK	995	10:20	Delhi	02h 15m	non-stop	12:35	Mumbai	5,955
4	11-02-2022	Vistara	UK	963	08:50	Delhi	02h 20m	non-stop	11:10	Mumbai	5,955

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206774 entries, 0 to 206773
Data columns (total 11 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   date        206774 non-null  object
 1   airline     206774 non-null  object
 2   ch_code     206774 non-null  object
 3   num_code    206774 non-null  int64 
 4   dep_time    206774 non-null  object
 5   from        206774 non-null  object
 6   time_taken  206774 non-null  object
 7   stop        206774 non-null  object
 8   arr_time    206774 non-null  object
```

```
 9   to          206774 non-null  object
 10  price       206774 non-null  object
dtypes: int64(1), object(10)
memory usage: 17.4+ MB
```

[4]:
```
df.isnull().sum()
```

[4]:
```
date          0
airline       0
ch_code       0
num_code      0
dep_time      0
from          0
time_taken    0
stop          0
arr_time      0
to            0
price         0
dtype: int64
```

[5]:
```
def rename_columns(df):
    df.rename(columns={'from':"Origin", 'to':'Destination'}, inplace=True)
rename_columns(df)
```

[6]:
```
df.head()
```

[6]:

	date	airline	ch_code	num_code	dep_time	Origin	time_taken	stop	arr_time	Destination	price
0	11-02-2022	SpiceJet	SG	8709	18:55	Delhi	02h 10m	non-stop	21:05	Mumbai	5,953
1	11-02-2022	SpiceJet	SG	8157	06:20	Delhi	02h 20m	non-stop	08:40	Mumbai	5,953
2	11-02-2022	AirAsia	I5	764	04:25	Delhi	02h 10m	non-stop	06:35	Mumbai	5,956
3	11-02-2022	Vistara	UK	995	10:20	Delhi	02h 15m	non-stop	12:35	Mumbai	5,955
4	11-02-2022	Vistara	UK	963	08:50	Delhi	02h 20m	non-stop	11:10	Mumbai	5,955

[7]:
```
df.drop(columns=['ch_code','num_code'], inplace=True)
```

[8]:
```
df.head(1)
```

[8]:

	date	airline	dep_time	Origin	time_taken	stop	arr_time	Destination	price
0	11-02-2022	SpiceJet	18:55	Delhi	02h 10m	non-stop	21:05	Mumbai	5,953

```
  Destination  price
0      Mumbai  5,953
```

[9]:
```
df['price'] = df['price'].str.replace(',','').astype(int) # remove comma in␣
↪price
```

[10]:
```
df['price'].value_counts()
```

[10]:
```
2339     1442
5949     1196
5955     1138
6489     1082
6067     1063
         …
10598       1
11333       1
17468       1
6071        1
6541        1
Name: price, Length: 9819, dtype: int64
```

0.1 date

[11]:
```
df['date'] = pd.to_datetime(df['date'])
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_3256\3532345252.py:1: UserWarning:
Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was
specified. This may lead to inconsistently parsed dates! Specify a format to
ensure consistent parsing.
  df['date'] = pd.to_datetime(df['date'])
```

[12]:
```
df.head(1)
```

[12]:
```
        date   airline dep_time Origin time_taken       stop arr_time  \
0 2022-11-02  SpiceJet    18:55  Delhi    02h 10m  non-stop     21:05

  Destination  price
0      Mumbai   5953
```

[13]:
```
df['Day'] = df['date'].dt.day
df['Month'] = df['date'].dt.month
```

[14]:
```
df.head(1)
```

[14]:
```
        date   airline dep_time Origin time_taken       stop arr_time  \
0 2022-11-02  SpiceJet    18:55  Delhi    02h 10m  non-stop     21:05
```

```
  Destination  price  Day  Month
0      Mumbai   5953    2     11
```

[15]:
```
df.drop('date', axis=1, inplace=True)
```

[16]:
```
df.head(1)
```

[16]:
```
    airline dep_time Origin time_taken       stop arr_time Destination  price  \
0  SpiceJet    18:55  Delhi    02h 10m  non-stop     21:05      Mumbai   5953

   Day  Month
0    2     11
```

0.2 duration

[17]:
```
df.rename(columns ={'time_taken':'Duration'}, inplace=True)
```

[18]:
```
df.head(1)
```

[18]:
```
    airline dep_time Origin Duration       stop arr_time Destination  price  \
0  SpiceJet    18:55  Delhi  02h 10m  non-stop     21:05      Mumbai   5953

   Day  Month
0    2     11
```

[19]:
```
def convert_duration(duration):
    try:
        if len(duration.split()) ==2:
            hours = int(duration.split()[0][: -1])
            minutes = int(duration.split()[1][:-1])
            return hours * 60 + minutes
        else:
            return int(float(duration[: -1])* 60)
    except ValueError:
        return 0
```

[20]:
```
df['Duration'] = df['Duration'].apply(convert_duration)
df.head()
```

[20]:
```
    airline dep_time Origin  Duration       stop arr_time Destination  price  \
0  SpiceJet    18:55  Delhi       130  non-stop     21:05      Mumbai   5953
1  SpiceJet    06:20  Delhi       140  non-stop     08:40      Mumbai   5953
2   AirAsia    04:25  Delhi       130  non-stop     06:35      Mumbai   5956
3   Vistara    10:20  Delhi       135  non-stop     12:35      Mumbai   5955
4   Vistara    08:50  Delhi       140  non-stop     11:10      Mumbai   5955

   Day  Month
```

```
0    2     11
1    2     11
2    2     11
3    2     11
4    2     11
```

[21]:
```
df['Duration'].value_counts()
```

[21]:
```
130     3081
135     2912
165     2006
125     1969
170     1724
        …
2865       1
2445       1
2175       1
2495       1
2490       1
Name: Duration, Length: 477, dtype: int64
```

0.3 departure time and arrival time

[22]:
```
df['dep_time'] = pd.to_datetime(df['dep_time'])
df['arr_time'] = pd.to_datetime(df['arr_time'])
df.dtypes
```

[22]:
```
airline                object
dep_time       datetime64[ns]
Origin                 object
Duration                int64
stop                   object
arr_time       datetime64[ns]
Destination            object
price                   int32
Day                     int64
Month                   int64
dtype: object
```

[23]:
```
# convert hours and minutes column
df['dep_time_hour'] = df['dep_time'].dt.hour
df['dep_time_minute'] = df['dep_time'].dt.minute

df['arr_time_hour'] = df['arr_time'].dt.hour
df['arr_time_minute'] = df['arr_time'].dt.minute
```

[24]:
```
df.head(2)
```

[24]:
```
     airline            dep_time Origin  Duration       stop  \
0  SpiceJet 2024-01-27 18:55:00  Delhi       130  non-stop
1  SpiceJet 2024-01-27 06:20:00  Delhi       140  non-stop

             arr_time Destination  price  Day  Month  dep_time_hour  \
0 2024-01-27 21:05:00      Mumbai   5953    2     11             18
1 2024-01-27 08:40:00      Mumbai   5953    2     11              6

   dep_time_minute  arr_time_hour  arr_time_minute
0               55             21                5
1               20              8               40
```

[25]:
```
df.drop(['dep_time','arr_time'], axis=1, inplace=True)
df.head(2)
```

[25]:
```
     airline Origin  Duration       stop Destination  price  Day  Month  \
0  SpiceJet  Delhi       130  non-stop       Mumbai   5953    2     11
1  SpiceJet  Delhi       140  non-stop       Mumbai   5953    2     11

   dep_time_hour  dep_time_minute  arr_time_hour  arr_time_minute
0             18               55             21                5
1              6               20              8               40
```

0.4 stop

[26]:
```
df['stop'].unique()
```

[26]:
```
array(['non-stop ',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\t\t\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia IXU\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Chennai\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Indore\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia RPR\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '2+-stop',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Lucknow\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia GOP\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Raipur\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Nagpur\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Surat\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia
Hyderabad\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia STV\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia IDR\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia VTZ\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia Guwahati\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia NAG\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\tVia
```

```
Vishakhapatnam\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia NDC\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia IXE\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Patna\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia PAT\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia
Mangalore\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Mumbai\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia KLH\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia MYQ\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Kolkata\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Delhi\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Kolhapur\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia JRG\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia
Bhubaneswar\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Ranchi\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia IXR\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia JGB\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia BBI\n\t\t\t\t\t\t\t\t\t\t\t\t\t',
       '1-stop\n\t\t\t\t\t\t\t\t\t\t\t\t\tVia Mysore\n\t\t\t\t\t\t\t\t\t\t\t\t\t'],
      dtype=object)
```

[27]:
```
def convert_stop(stop):
    if 'non-stop' in stop.lower():
        return 0
    elif 'stop' in stop.lower():
        stop_count = 0
        for char in stop:
            if char.isdigit():
                stop_count= int(char)
                break
        return stop_count
    else:
        return -1
```

[28]:
```
df['stop_code'] = df['stop'].apply(convert_stop)
```

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

[29]:

	airline	Origin	Duration	stop	Destination	price	Day	Month	\
0	SpiceJet	Delhi	130	non-stop	Mumbai	5953	2	11	
1	SpiceJet	Delhi	140	non-stop	Mumbai	5953	2	11	
2	AirAsia	Delhi	130	non-stop	Mumbai	5956	2	11	
3	Vistara	Delhi	135	non-stop	Mumbai	5955	2	11	
4	Vistara	Delhi	140	non-stop	Mumbai	5955	2	11	

```
   dep_time_hour  dep_time_minute  arr_time_hour  arr_time_minute  stop_code
0             18               55             21                5          0
1              6               20              8               40          0
2              4               25              6               35          0
3             10               20             12               35          0
4              8               50             11               10          0
```

[30]:
```
df['stop_code'].value_counts()
```

[30]:
```
1    166627
0     27942
2     12205
Name: stop_code, dtype: int64
```

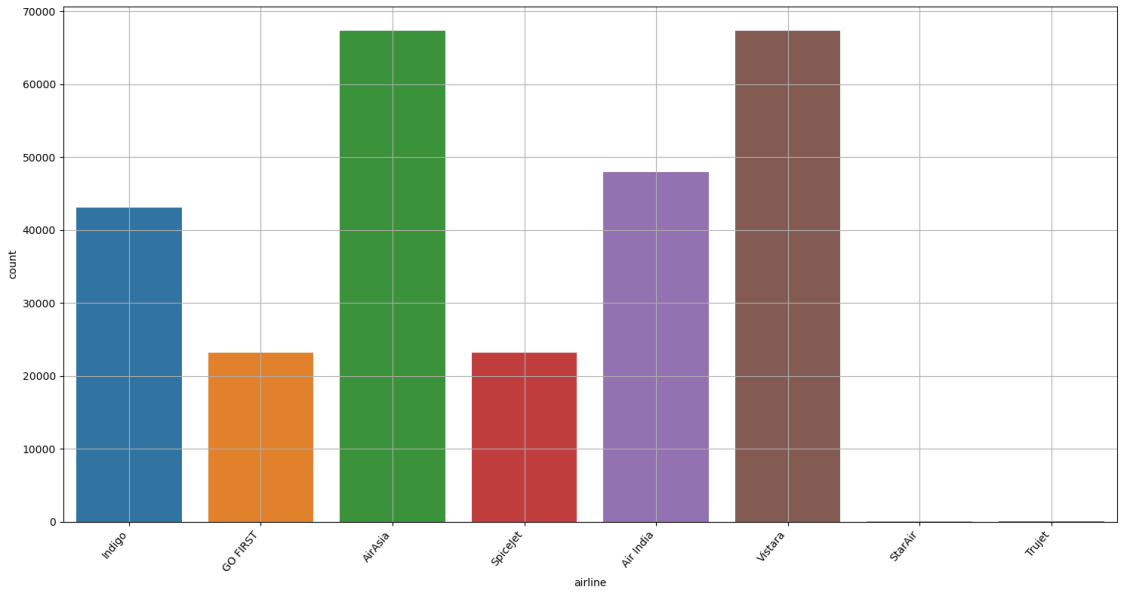
[31]:
```
df.drop(['stop'], axis=1, inplace=True)
```

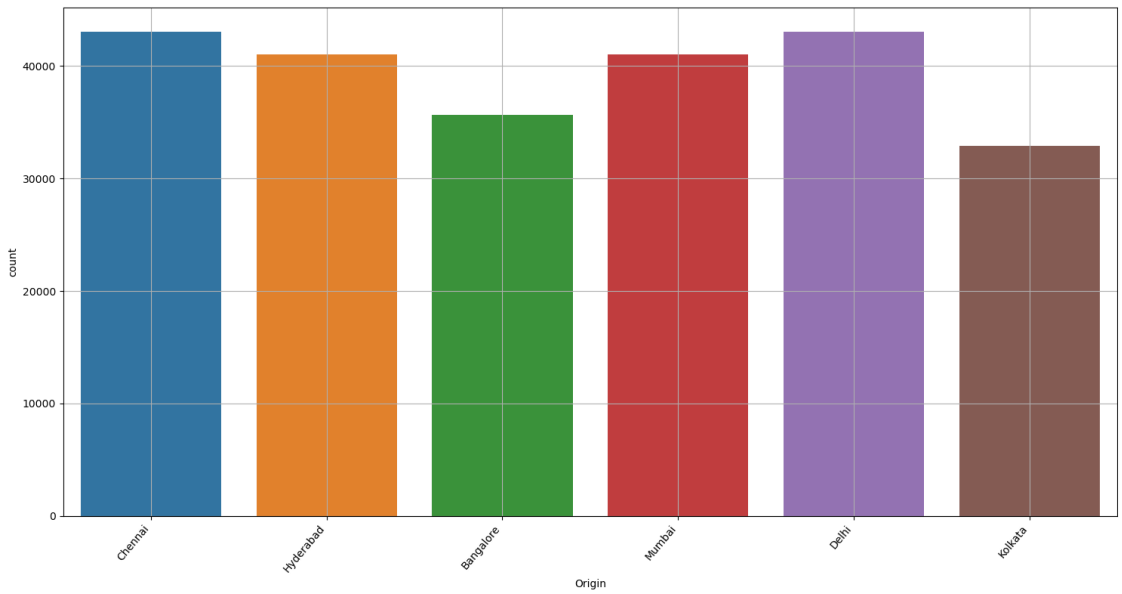
[32]:
```
df.head(1)
```

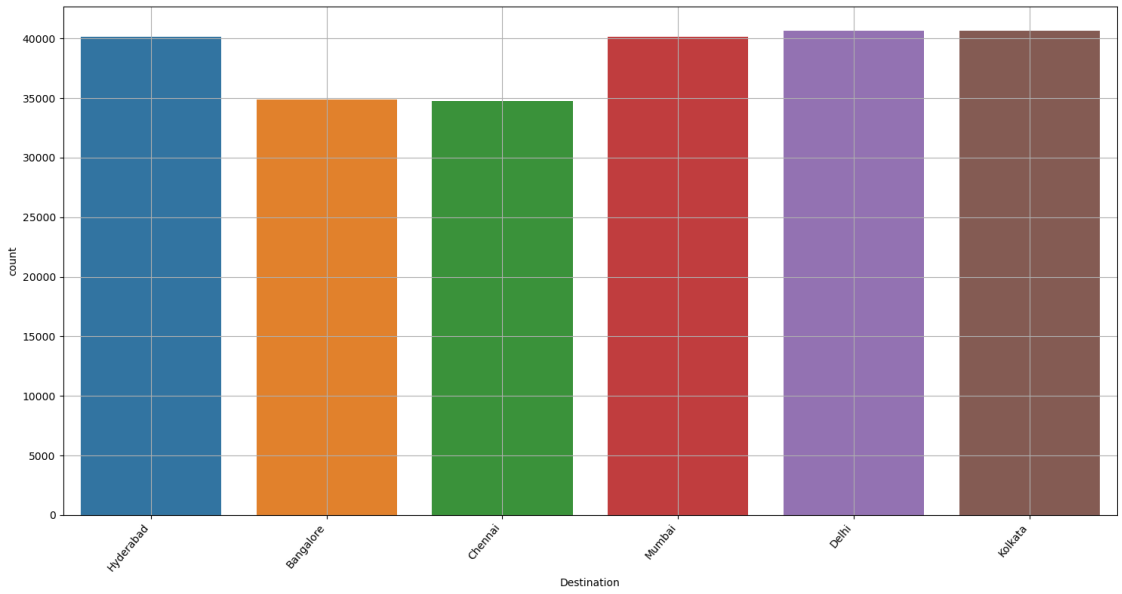
[32]:
```
    airline Origin  Duration Destination  price  Day  Month  dep_time_hour  \
0  SpiceJet  Delhi       130      Mumbai   5953    2     11             18

   dep_time_minute  arr_time_hour  arr_time_minute  stop_code
0               55             21                5          0
```

[33]:
```
for I in ['airline','Origin','Destination']:
    plt.figure(figsize=(15,8))
    sns.countplot(data=df, x=I)
    ax = sns.countplot(x=I, data=df.sort_values('price', ascending=True))
    plt.xticks(rotation=50, ha='right')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
    print('\n\n')
```


70000
60000
50000
40000
30000
20000
10000
0
count
Indigo
GO FIRST
AirAsia
SpiceJet
Air India
Vistara
StarAir
Trujet
airline


40000
30000
20000
10000
0
count
Chennai
Hyderabad
Bangalore
Mumbai
Delhi
Kolkata
Origin

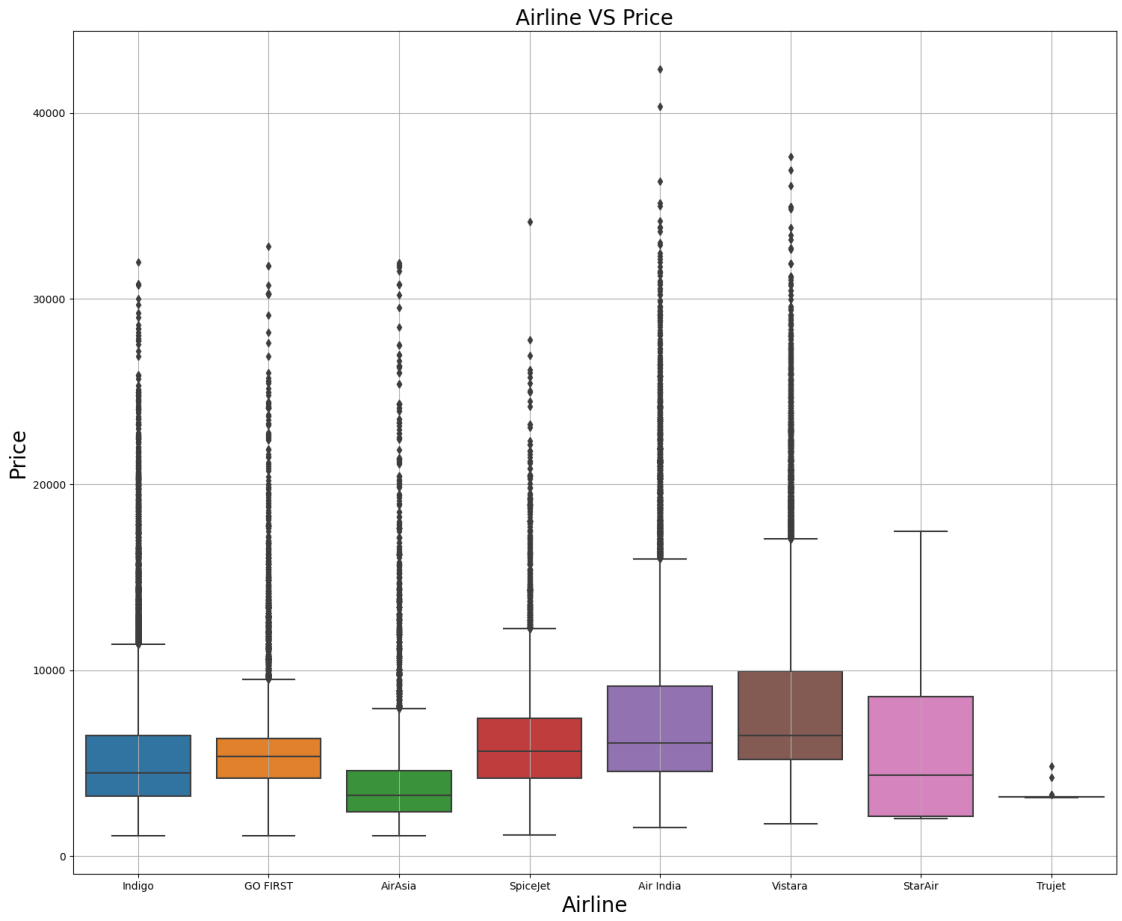


[34]:
```
df['airline'].value_counts()
```

[34]:
```
Vistara      67270
Air India    47996
Indigo       43120
GO FIRST     23177
AirAsia      16098
SpiceJet      9011
StarAir         61
Trujet          41
Name: airline, dtype: int64
```

[35]:
```
plt.figure(figsize=(15,8))
sns.barplot(x='airline', y='price', data=df.sort_values('price',␣
 ↪ascending=False), palette='viridis')
plt.tight_layout()
plt.xlabel('Airline', fontsize=20)
plt.ylabel('Price', fontsize=20)
plt.title('Airline VS Price', fontsize=20)
plt.grid()
plt.show()
```

[36]:
```
plt.figure(figsize=(15,12))
sns.boxplot(x='airline', y='price', data=df.sort_values('price',␣
 ↪ascending=True))
plt.tight_layout()
plt.xlabel('Airline', fontsize=20)
plt.ylabel('Price', fontsize=20)
plt.title('Airline VS Price', fontsize=20)
plt.grid()
plt.show()
```



[37]:
```
df.groupby('airline').describe()['price'].sort_values('mean', ascending=True).T
```

[37]:

airline	Trujet	AirAsia	StarAir	Indigo	GO FIRST
count	41.000000	16098.000000	61.000000	43120.000000	23177.000000
mean	3244.634146	4091.072742	4932.655738	5324.216303	5651.930578
std	305.184596	2824.055172	3487.874935	3268.894831	2513.669305
min	3124.000000	1105.000000	2000.000000	1105.000000	1105.000000
25%	3166.000000	2361.000000	2115.000000	3219.000000	4205.000000
50%	3196.000000	3276.000000	4337.000000	4453.000000	5336.000000
75%	3196.000000	4589.000000	8582.000000	6489.000000	6324.000000
max	4844.000000	31917.000000	17482.000000	31952.000000	32803.000000

airline	SpiceJet	Air India	Vistara
count	9011.000000	47996.000000	67270.000000
mean	6179.278881	7313.730144	7806.943645
std	2999.630406	3989.516123	3854.572559
min	1106.000000	1526.000000	1714.000000
25%	4197.000000	4556.000000	5206.000000

```
50%       5654.000000   6082.000000   6461.000000
75%       7412.000000   9135.000000   9945.000000
max      34158.000000  42349.000000  37646.000000
```

[38]:
```
list1 = ['Origin', 'Destination']
for x in list1:
    print(df[[x]].value_counts(),'\n\n\n')
```

```
Origin
Delhi        43029
Mumbai       41045
Bangalore    35665
Kolkata      32874
Hyderabad    27990
Chennai      26171
dtype: int64



Destination
Delhi          40654
Mumbai         40118
Bangalore      34914
Kolkata        34777
Hyderabad      29101
Chennai        27210
dtype: int64
```

[39]:
```
df = pd.get_dummies(df)
```

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

[40]:

	Duration	price	Day	Month	dep_time_hour	dep_time_minute	arr_time_hour	\
0	130	5953	2	11	18	55	21	
1	140	5953	2	11	6	20	8	
2	130	5956	2	11	4	25	6	
3	135	5955	2	11	10	20	12	
4	140	5955	2	11	8	50	11	

	arr_time_minute	stop_code	airline_Air India	…	Origin_Delhi	\
0	5	0	0	…	1	
1	40	0	0	…	1	
2	35	0	0	…	1	
3	35	0	0	…	1	

```
4               10          0                   0  …              1

   Origin_Hyderabad  Origin_Kolkata  Origin_Mumbai  Destination_Bangalore  \
0                 0               0              0                      0
1                 0               0              0                      0
2                 0               0              0                      0
3                 0               0              0                      0
4                 0               0              0                      0

   Destination_Chennai  Destination_Delhi  Destination_Hyderabad  \
0                    0                  0                      0
1                    0                  0                      0
2                    0                  0                      0
3                    0                  0                      0
4                    0                  0                      0

   Destination_Kolkata  Destination_Mumbai
0                    0                   1
1                    0                   1
2                    0                   1
3                    0                   1
4                    0                   1

[5 rows x 29 columns]
```

[41]:
```
from sklearn.model_selection import train_test_split
```

[42]:
```
x = df.drop('price', axis=1)
y = df['price']
```

[43]:
```
x.head()
```

[43]:
```
   Duration  Day  Month  dep_time_hour  dep_time_minute  arr_time_hour  \
0       130    2     11             18               55             21
1       140    2     11              6               20              8
2       130    2     11              4               25              6
3       135    2     11             10               20             12
4       140    2     11              8               50             11

   arr_time_minute  stop_code  airline_Air India  airline_AirAsia  …  \
0                5          0                  0                0  …
1               40          0                  0                0  …
2               35          0                  0                1  …
3               35          0                  0                0  …
4               10          0                  0                0  …

   Origin_Delhi  Origin_Hyderabad  Origin_Kolkata  Origin_Mumbai  \
```

```
0              1                  0                 0               0
1              1                  0                 0               0
2              1                  0                 0               0
3              1                  0                 0               0
4              1                  0                 0               0

   Destination_Bangalore  Destination_Chennai  Destination_Delhi  \
0                      0                    0                  0
1                      0                    0                  0
2                      0                    0                  0
3                      0                    0                  0
4                      0                    0                  0

   Destination_Hyderabad  Destination_Kolkata  Destination_Mumbai
0                      0                    0                   1
1                      0                    0                   1
2                      0                    0                   1
3                      0                    0                   1
4                      0                    0                   1

[5 rows x 28 columns]
```

[44]:
```
y.head()
```

[44]:
```
0    5953
1    5953
2    5956
3    5955
4    5955
Name: price, dtype: int32
```

[45]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2,␣
  ↪random_state=42)
print('x_train',x_train.shape)
print('x_test',x_test.shape)
print('y_train',y_train.shape)
print('y_test',y_test.shape)
```

```
x_train (165419, 28)
x_test (41355, 28)
y_train (165419,)
y_test (41355,)
```

1 RandomForestRegressor

[46]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

rf_reg = RandomForestRegressor(n_estimators=50, random_state=42)
rf_reg.fit(x_train, y_train.ravel())

y_pred = rf_reg.predict(x_test)

mse_1 = mean_squared_error(y_test, y_pred)
r2_1 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse_1:.2f}')
print(f'R-squared: {r2_1:.2f}')
```

```
Mean Squared Error: 1591296.81
R-squared: 0.89
```

2 LinearRegression

[47]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
lin_reg_model = LinearRegression()

lin_reg_model.fit(x_train,y_train)

lin_reg_pred = lin_reg_model.predict(x_test)

MSE_2 = mean_squared_error(y_test,lin_reg_pred)
R2_2 = r2_score(y_test, lin_reg_pred)

print(f'Mean Squared Error: {MSE_2:.2f}')
print(f'R_squared: {R2_2:.2f}')
```

```
Mean Squared Error: 10904791.34
R_squared: 0.22
```

[ ]:
```

```