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


df=pd.read_excel('flight_price.xlsx')
df.head()
```

Out[4]:

	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 [5]:
```python
df.info()   #information about the data
```

```
<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 [6]:
```python
df.shape
```

Out[6]:
```
(10683, 11)
```

In [7]:
```python
#finding the null values
df.isnull().sum()
```

Out[7]:
```
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 [8]:
```python
df.dropna(inplace=True)
```

In [9]:
```python
df.isnull().sum()
```

Out[9]:
```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              0
Dep_Time           0
Arrival_Time       0
Duration           0
Total_Stops        0
Additional_Info    0
Price              0
dtype: int64
```

In [10]:
```python
df.dtypes
```

Out[10]:
```
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
dtype: object
```

In [12]:
```python
def change_into_datetime(col):
    df[col]=pd.to_datetime(df[col])
```

In [13]:
```python
df.columns
```

Out[13]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

In [14]:
```python
for i in ['Date_of_Journey','Dep_Time', 'Arrival_Time']:
    change_into_datetime(i)
```

```
C:\Users\rautv\AppData\Local\Temp\ipykernel_20604\3204248598.py:2: UserWarning: Parsing dates in %d/%m/%Y format when dayfirst=False (the default) was specified. Pass `dayfirst=True` or specify a format to silence this warning.
  df[col]=pd.to_datetime(df[col])
C:\Users\rautv\AppData\Local\Temp\ipykernel_20604\3204248598.py:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df[col]=pd.to_datetime(df[col])
C:\Users\rautv\AppData\Local\Temp\ipykernel_20604\3204248598.py:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df[col]=pd.to_datetime(df[col])
```

In [15]:
```python
df.dtypes
```

Out[15]:
```
Airline                    object
Date_of_Journey    datetime64[ns]
Source                     object
Destination                object
Route                      object
Dep_Time           datetime64[ns]
Arrival_Time       datetime64[ns]
Duration                   object
Total_Stops                object
Additional_Info            object
Price                       int64
dtype: object
```

In [16]:
```python
df['journey_day']=df['Date_of_Journey'].dt.day
df['journey_month']=df['Date_of_Journey'].dt.month
```

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

Out[18]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	journey_day	journey_month
0	IndiGo	2019-03-24	Banglore	New Delhi	BLR → DEL	2024-01-22 22:20:00	2024-03-22 01:10:00	2h 50m	non-stop	No info	3897	24	3
1	Air India	2019-05-01	Kolkata	Banglore	CCU → IXR → BBI → BLR	2024-01-22 05:50:00	2024-01-22 13:15:00	7h 25m	2 stops	No info	7662	1	5
2	Jet Airways	2019-06-09	Delhi	Cochin	DEL → LKO → BOM → COK	2024-01-22 09:25:00	2024-06-10 04:25:00	19h	2 stops	No info	13882	9	6
3	IndiGo	2019-05-12	Kolkata	Banglore	CCU → NAG → BLR	2024-01-22 18:05:00	2024-01-22 23:30:00	5h 25m	1 stop	No info	6218	12	5
4	IndiGo	2019-03-01	Banglore	New Delhi	BLR → NAG → DEL	2024-01-22 16:50:00	2024-01-22 21:35:00	4h 45m	1 stop	No info	13302	1	3

In [19]:
```python
df.drop('Date_of_Journey', axis=1, inplace=True)
```

In [28]:
```python
df['Duration_hours'] = df['Duration'].str.extract('(\d+)h').fillna(0).astype(int)
df['Duration_minutes'] = df['Duration'].str.extract('(\d+)m').fillna(0).astype(int)

# If you want to convert hours and minutes to a single duration in minutes, you can add a new column
df['Total_duration_minutes'] = df['Duration_hours'] * 60 + df['Duration_minutes']
```

```
<>:4: SyntaxWarning: invalid escape sequence '\d'
<>:5: SyntaxWarning: invalid escape sequence '\d'
<>:4: SyntaxWarning: invalid escape sequence '\d'
<>:5: SyntaxWarning: invalid escape sequence '\d'
C:\Users\rautv\AppData\Local\Temp\ipykernel_20604\3953932600.py:4: SyntaxWarning: invalid escape sequence '\d'
  df['Duration_hours'] = df['Duration'].str.extract('(\d+)h').fillna(0).astype(int)
C:\Users\rautv\AppData\Local\Temp\ipykernel_20604\3953932600.py:5: SyntaxWarning: invalid escape sequence '\d'
  df['Duration_minutes'] = df['Duration'].str.extract('(\d+)m').fillna(0).astype(int)
```

In [ ]:

In [30]:
```python
df.head()
```

Out[30]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	journey_day	journey_month	Duration_hours	Duration_minutes	Total_duration_minutes
0	IndiGo	Banglore	New Delhi	BLR → DEL	2024-01-22 22:20:00	2024-03-22 01:10:00	2h 50m	non-stop	No info	3897	24	3	2	50	170
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2024-01-22 05:50:00	2024-01-22 13:15:00	7h 25m	2 stops	No info	7662	1	5	7	25	445
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	2024-01-22 09:25:00	2024-06-10 04:25:00	19h	2 stops	No info	13882	9	6	19	0	1140
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	2024-01-22 18:05:00	2024-01-22 23:30:00	5h 25m	1 stop	No info	6218	12	5	5	25	325
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	2024-01-22 16:50:00	2024-01-22 21:35:00	4h 45m	1 stop	No info	13302	1	3	4	45	285

In [34]:
```python
df.dtypes
```

Out[34]:
```
Airline                           object
Source                            object
Destination                       object
Route                             object
Dep_Time                  datetime64[ns]
Arrival_Time              datetime64[ns]
Duration                          object
Total_Stops                       object
Additional_Info                   object
Price                              int64
journey_day                        int32
journey_month                      int32
Duration_hours                     int32
Duration_minutes                   int32
Total_duration_minutes             int32
dtype: object
```

In [35]:
```python
# function for extracting hour and minutes
def extract_hour(data,col):
    data[col+'_hour']=data[col].dt.hour

def extract_min(data,col):
    data[col+'_min']=data[col].dt.minute


def drop_col(data,col):
    data.drop(col,axis=1,inplace=True)
```

In [36]:
```python
#call the function
# Departure time is when a plane leaves the gate.
# Similar to Date_of_Journey we can extract values from Dep_Time
extract_hour(df,'Dep_Time')

#extracting minutes
extract_min(df,'Dep_Time')

#drop the column
drop_col(df,'Dep_Time')
```

In [37]:
```python
#extracting hour
extract_hour(df,'Arrival_Time')

#extracting min
extract_min(df,'Arrival_Time')


#drop the column
drop_col(df,'Arrival_Time')
```

In [38]:
```python
df.head(10)
```

Out[38]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	journey_day	journey_month	Duration_hours	Duration_minutes	Total_duration_minutes	Dep_Time_hour	Dep_Time_min	Arrival_Time_hour	Arrival_Time_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24	3	2	50	170	22	20	1	10
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	1	5	7	25	445	5	50	13	15
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	19h	2 stops	No info	13882	9	6	19	0	1140	9	25	4	25
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	5h 25m	1 stop	No info	6218	12	5	5	25	325	18	5	23	30
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	4h 45m	1 stop	No info	13302	1	3	4	45	285	16	50	21	35
5	SpiceJet	Kolkata	Banglore	CCU → BLR	2h 25m	non-stop	No info	3873	24	6	2	25	145	9	0	11	25
6	Jet Airways	Banglore	New Delhi	BLR → BOM → DEL	15h 30m	1 stop	In-flight meal not included	11087	12	3	15	30	930	18	55	10	25
7	Jet Airways	Banglore	New Delhi	BLR → BOM → DEL	21h 5m	1 stop	No info	22270	1	3	21	5	1265	8	0	5	5
8	Jet Airways	Banglore	New Delhi	BLR → BOM → DEL	25h 30m	1 stop	In-flight meal not included	11087	12	3	25	30	1530	8	55	10	25
9	Multiple carriers	Delhi	Cochin	DEL → BOM → COK	7h 50m	1 stop	No info	8625	27	5	7	50	470	11	25	19	15

In [48]:
```python
def hour(x):
    return x.split(' ')[0][0:-1]

def minutes(x):
    return x.split(' ')[1][0:-1]
```

In [53]:
```python
df.drop('Duration', axis=1, inplace=True)
```

In [54]:
```python
df.head()
```

Out[54]:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Price	journey_day	journey_month	Duration_hours	Duration_minutes	Dep_Time_hour	Dep_Time_min	Arrival_Time_hour	Arrival_Time_min	dur_hour
0	IndiGo	Banglore	New Delhi	BLR → DEL	non-stop	No info	3897	24	3	2	50	22	20	1	10	2
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2 stops	No info	7662	1	5	7	25	5	50	13	15	7
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	2 stops	No info	13882	9	6	19	0	9	25	4	25	19
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	1 stop	No info	6218	12	5	5	25	18	5	23	30	5
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	1 stop	No info	13302	1	3	4	45	16	50	21	35	4

In [57]:
```python
df.dtypes
```

Out[57]:
```
Airline              object
Source               object
Destination          object
Route                object
Total_Stops          object
Additional_Info      object
Price                 int64
journey_day           int32
journey_month         int32
Duration_hours        int32
Duration_minutes      int32
Dep_Time_hour         int32
Dep_Time_min          int32
Arrival_Time_hour     int32
Arrival_Time_min      int32
dur_hour             object
dtype: object
```

In [58]:
```python
df['dur_hour'] = df['dur_hour'].astype(int)
```

In [59]:
```python
df.dtypes
```

Out[59]:
```
Airline              object
Source               object
Destination          object
Route                object
Total_Stops          object
Additional_Info      object
Price                 int64
journey_day           int32
journey_month         int32
Duration_hours        int32
Duration_minutes      int32
Dep_Time_hour         int32
Dep_Time_min          int32
Arrival_Time_hour     int32
Arrival_Time_min      int32
dur_hour              int32
dtype: object
```

In [60]:
```python
column=[column for column in df.columns if df[column].dtype=='object']
column
```

Out[60]:
```
['Airline', 'Source', 'Destination', 'Route', 'Total_Stops', 'Additional_Info']
```

In [61]:
```python
continuous_col =[column for column in df.columns if df[column].dtype!='object']
continuous_col
```

Out[61]:
```
['Price',
 'journey_day',
 'journey_month',
 'Duration_hours',
 'Duration_minutes',
 'Dep_Time_hour',
 'Dep_Time_min',
 'Arrival_Time_hour',
 'Arrival_Time_min',
 'dur_hour']
```

In [63]:
```python
categorical = df[column]
categorical.head()
```

Out[63]:

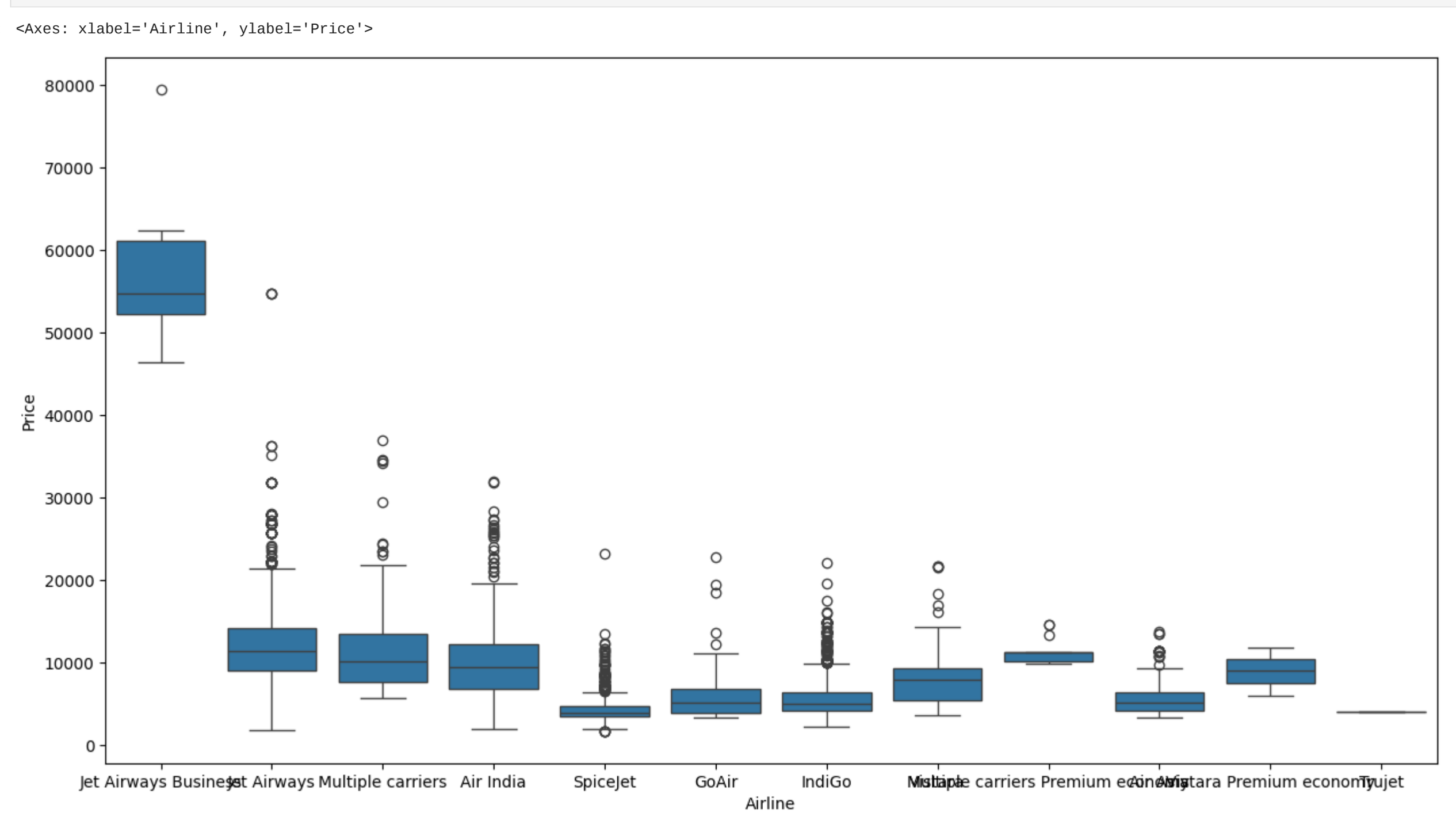
	Airline	Source	Destination	Route	Total_Stops	Additional_Info
0	IndiGo	Banglore	New Delhi	BLR → DEL	non-stop	No info
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2 stops	No info
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	2 stops	No info
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	1 stop	No info
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	1 stop	No info

In [64]:
```python
categorical['Airline'].value_counts()
```

Out[64]:
```
Airline
Jet Airways                          3849
IndiGo                               2053
Air India                            1751
Multiple carriers                    1196
SpiceJet                              818
Vistara                               479
Air Asia                              319
GoAir                                 194
Multiple carriers Premium economy      13
Jet Airways Business                    6
Vistara Premium economy                 3
Trujet                                  1
Name: count, dtype: int64
```

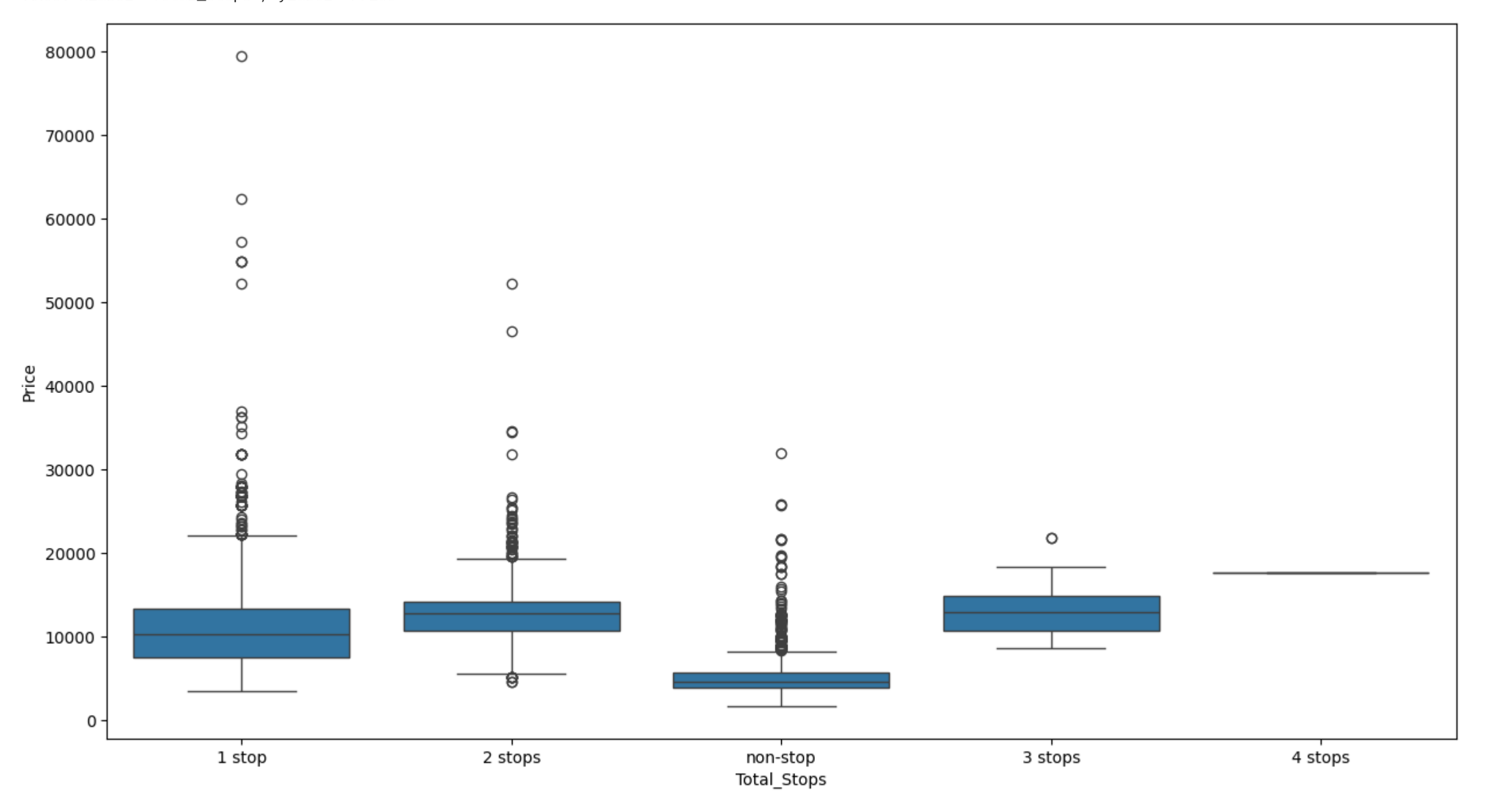
In [70]:
```python
plt.figure(figsize=(15,8))
sns.boxplot(x='Airline',y='Price',data=df.sort_values('Price',ascending=False))
```

Out[70]:
```
<Axes: xlabel='Airline', ylabel='Price'>
```

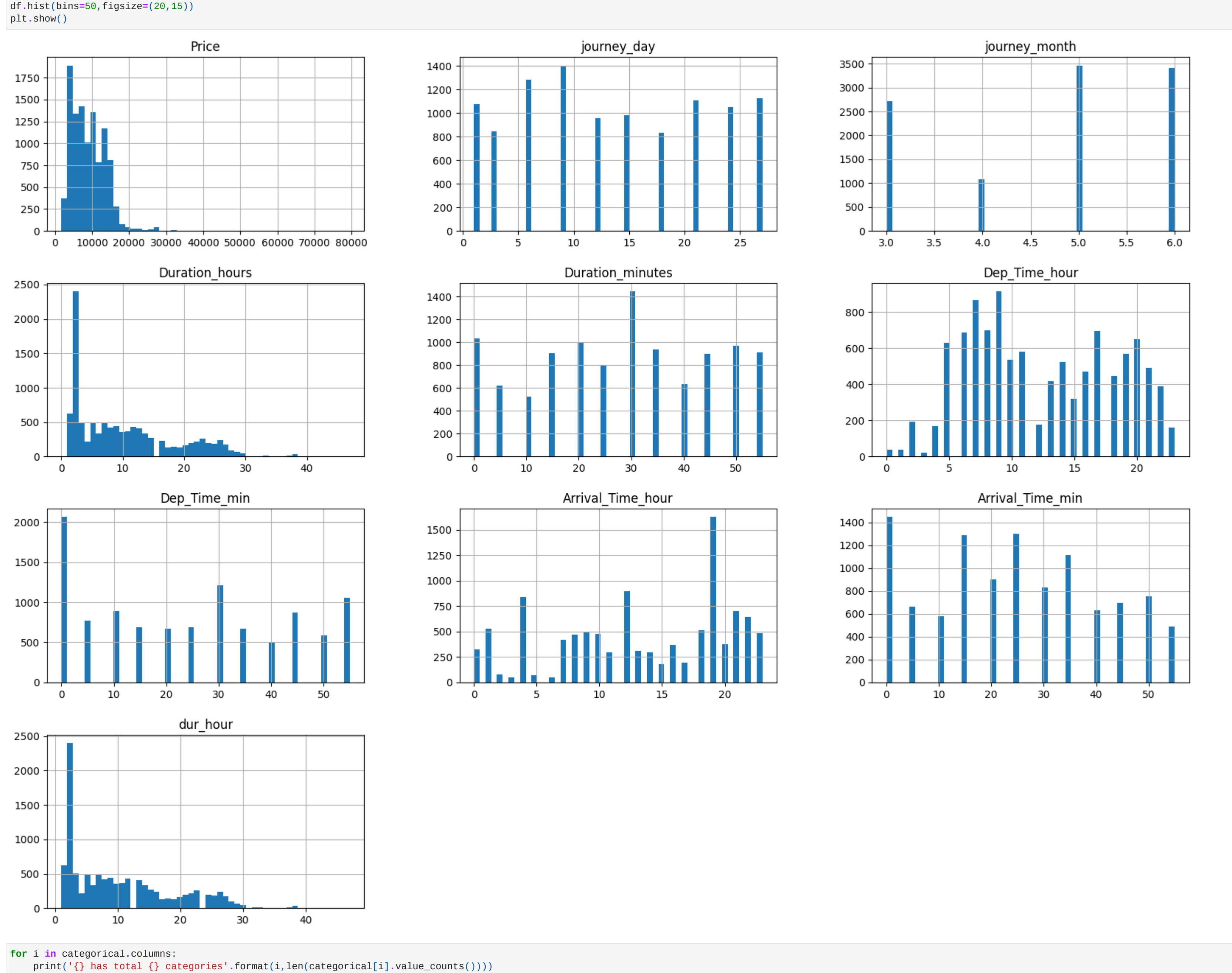


In [71]:
```python
plt.figure(figsize=(15,8))
sns.boxplot(x='Total_Stops',y='Price',data=df.sort_values('Price',ascending=False))
```

Out[71]:
```
<Axes: xlabel='Total_Stops', ylabel='Price'>
```



In [72]:
```python
#sns.set_style('seaborn')
df.hist(bins=50,figsize=(20,15))
plt.show()
```



In [73]:
```python
for i in categorical.columns:
    print('{} has total {} categories'.format(i,len(categorical[i].value_counts())))
```

```
Airline has total 12 categories
Source has total 5 categories
Destination has total 6 categories
Route has total 128 categories
Total_Stops has total 5 categories
Additional_Info has total 10 categories
```

In [ ]: