airline-project-2

August 10, 2025

[ ]:
```
This Airline Project created by Fahad Tofeeq
```

Installation And Import Process

[118]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[11]:
```
import matplotlib.pyplot as plt
```

[3]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in d:\python\lib\site-packages (2.3.1)
Requirement already satisfied: numpy>=1.26.0 in d:\python\lib\site-packages
(from pandas) (2.3.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\fahad
tofeeq\appdata\roaming\python\python313\site-packages (from pandas)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in d:\python\lib\site-packages (from
pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in d:\python\lib\site-packages
(from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\fahad
tofeeq\appdata\roaming\python\python313\site-packages (from python-
dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

[4]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in d:\python\lib\site-packages (2.3.2)
Note: you may need to restart the kernel to use updated packages.
```

[5]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in d:\python\lib\site-packages
(3.10.5)
Requirement already satisfied: contourpy>=1.0.1 in d:\python\lib\site-packages
```

```
(from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in d:\python\lib\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in d:\python\lib\site-packages
(from matplotlib) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in d:\python\lib\site-packages
(from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in d:\python\lib\site-packages (from
matplotlib) (2.3.2)
Requirement already satisfied: packaging>=20.0 in d:\python\lib\site-packages
(from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in d:\python\lib\site-packages (from
matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in d:\python\lib\site-packages
(from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\fahad
tofeeq\appdata\roaming\python\python313\site-packages (from matplotlib)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\fahad
tofeeq\appdata\roaming\python\python313\site-packages (from python-
dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

[6]:

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in d:\python\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in d:\python\lib\site-
packages (from seaborn) (2.3.2)
Requirement already satisfied: pandas>=1.2 in d:\python\lib\site-packages (from
seaborn) (2.3.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in d:\python\lib\site-
packages (from seaborn) (3.10.5)
Requirement already satisfied: contourpy>=1.0.1 in d:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.3)
Requirement already satisfied: cycler>=0.10 in d:\python\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in d:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in d:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in d:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (25.0)
Requirement already satisfied: pillow>=8 in d:\python\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in d:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\fahad
tofeeq\appdata\roaming\python\python313\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in d:\python\lib\site-packages (from
pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in d:\python\lib\site-packages
(from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\fahad
tofeeq\appdata\roaming\python\python313\site-packages (from python-
dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

[7]:

```
pip install openpyxl
```

```
Requirement already satisfied: openpyxl in d:\python\lib\site-packages (3.1.5)
Requirement already satisfied: et-xmlfile in d:\python\lib\site-packages (from
openpyxl) (2.0.0)
Note: you may need to restart the kernel to use updated packages.
```

Loading Data

Check Directory Contents:

[13]:

```
import os
print(os.listdir())
```

```
['.ipynb_checkpoints', 'Airline project.ipynb', 'flight_analysis_output
(1).xlsx']
```

Load DataFrame:

[12]:

```
df = pd.read_excel("C:/Users/Fahad Tofeeq/Downloads/flight_analysis_output (1).
 ↪xlsx")
df
```

[12]:

```
        Unnamed: 0   index          airline   flight source_city  \
0                0       0              PIA  SG-8709   Islamabad
1                1       1              PIA  SG-8157   Islamabad
2                2       2         Air Blue   I5-764   Islamabad
3                3       3  Shaheen Airline   UK-995   Islamabad
4                4       4  Shaheen Airline   UK-963   Islamabad
…              …       …              …        …           …
300148      300148  300148  Shaheen Airline   UK-822      Quetta
300149      300149  300149  Shaheen Airline   UK-826      Quetta
300150      300150  300150  Shaheen Airline   UK-832      Quetta
300151      300151  300151  Shaheen Airline   UK-828      Quetta
300152      300152  300152  Shaheen Airline   UK-822      Quetta

       departure_time stops   arrival_time destination_city    class  \
0             Evening  zero          Night           Multan  Economy
1       Early_Morning  zero        Morning           Multan  Economy
2       Early_Morning  zero  Early_Morning           Multan  Economy
```

```
3             Morning  zero      Afternoon           Multan   Economy
4             Morning  zero        Morning           Multan   Economy
…               …   …            …              …       …
300148        Morning   one        Evening               GB  Business
300149      Afternoon   one          Night               GB  Business
300150  Early_Morning   one          Night               GB  Business
300151  Early_Morning   one        Evening               GB  Business
300152        Morning   one        Evening               GB  Business

        duration  days_left  price columns
0           2.17          1   5953  object
1           2.33          1   5953  object
2           2.17          1   5956  object
3           2.25          1   5955  object
4           2.33          1   5955  object
…          …        …    …     …
300148     10.08         49  69265  object
300149     10.42         49  77105  object
300150     13.83         49  79099  object
300151     10.00         49  81585  object
300152     10.08         49  81585  object

[300153 rows x 14 columns]
```

- Verify Columns:

[14]:
```
print(df.columns.tolist())
```

```
['Unnamed: 0', 'index', 'airline', 'flight', 'source_city', 'departure_time',
'stops', 'arrival_time', 'destination_city', 'class', 'duration', 'days_left',
'price', 'columns']
```

Cleaning Data

[15]:
```
df.isnull().sum()
```

[15]:
```
Unnamed: 0          0
index               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
columns             0
dtype: int64
```

Fill Missing Values:

```
[19]: df['duration']= df['duration'].fillna(df['duration'].mean())
      print(df)
```

```
        Unnamed: 0   index          airline   flight source_city  \
0                0       0              PIA  SG-8709   Islamabad
1                1       1              PIA  SG-8157   Islamabad
2                2       2         Air Blue   I5-764   Islamabad
3                3       3  Shaheen Airline   UK-995   Islamabad
4                4       4  Shaheen Airline   UK-963   Islamabad
…            …     …              …      …         …
300148      300148  300148  Shaheen Airline   UK-822      Quetta
300149      300149  300149  Shaheen Airline   UK-826      Quetta
300150      300150  300150  Shaheen Airline   UK-832      Quetta
300151      300151  300151  Shaheen Airline   UK-828      Quetta
300152      300152  300152  Shaheen Airline   UK-822      Quetta

       departure_time stops   arrival_time destination_city     class  \
0             Evening  zero          Night           Multan   Economy
1       Early_Morning  zero        Morning           Multan   Economy
2       Early_Morning  zero  Early_Morning           Multan   Economy
3             Morning  zero      Afternoon           Multan   Economy
4             Morning  zero        Morning           Multan   Economy
…               …   …            …              …       …
300148        Morning   one        Evening               GB  Business
300149      Afternoon   one          Night               GB  Business
300150  Early_Morning   one          Night               GB  Business
300151  Early_Morning   one        Evening               GB  Business
300152        Morning   one        Evening               GB  Business

        duration  days_left  price columns
0           2.17          1   5953  object
1           2.33          1   5953  object
2           2.17          1   5956  object
3           2.25          1   5955  object
4           2.33          1   5955  object
…          …        …    …     …
300148     10.08         49  69265  object
300149     10.42         49  77105  object
300150     13.83         49  79099  object
300151     10.00         49  81585  object
300152     10.08         49  81585  object
```

```
[300153 rows x 14 columns]
```

[23]:
```
df['stops'] = df['stops'].fillna('unknown')
df.head()
```

[23]:
```
   Unnamed: 0  index          airline   flight source_city departure_time  \
0           0      0              PIA  SG-8709   Islamabad        Evening
1           1      1              PIA  SG-8157   Islamabad  Early_Morning
2           2      2         Air Blue   I5-764   Islamabad  Early_Morning
3           3      3  Shaheen Airline   UK-995   Islamabad        Morning
4           4      4  Shaheen Airline   UK-963   Islamabad        Morning

  stops   arrival_time destination_city    class  duration  days_left  price  \
0  zero          Night           Multan  Economy      2.17          1   5953
1  zero        Morning           Multan  Economy      2.33          1   5953
2  zero  Early_Morning           Multan  Economy      2.17          1   5956
3  zero      Afternoon           Multan  Economy      2.25          1   5955
4  zero        Morning           Multan  Economy      2.33          1   5955

  columns
0  object
1  object
2  object
3  object
4  object
```

- Convert Data Types:

Check Data types

[24]:
```
df.dtypes
```

[24]:
```
Unnamed: 0            int64
index                 int64
airline              object
flight               object
source_city          object
departure_time       object
stops                object
arrival_time         object
destination_city     object
class                object
duration            float64
days_left             int64
price                 int64
columns              object
dtype: object
```

[25]:

```
df['price'] =  df['price'].astype(float)
df['duration'] =  df['duration'].astype(float)
df.dtypes
```

[25]:

```
Unnamed: 0            int64
index                 int64
airline              object
flight               object
source_city          object
departure_time       object
stops                object
arrival_time         object
destination_city     object
class                object
duration            float64
days_left             int64
price               float64
columns              object
dtype: object
```

[26]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 14 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Unnamed: 0        300153 non-null  int64
 1   index             300153 non-null  int64
 2   airline           300153 non-null  object
 3   flight            300153 non-null  object
 4   source_city       300153 non-null  object
 5   departure_time    300153 non-null  object
 6   stops             300153 non-null  object
 7   arrival_time      300153 non-null  object
 8   destination_city  300153 non-null  object
 9   class             300153 non-null  object
 10  duration          300153 non-null  float64
 11  days_left         300153 non-null  int64
 12  price             300153 non-null  float64
 13  columns           300153 non-null  object
dtypes: float64(2), int64(3), object(9)
memory usage: 32.1+ MB
```

[27]:

```
df.describe()
```

```
[27]:          Unnamed: 0          index       duration      days_left  \
      count  300153.000000  300153.000000  300153.000000  300153.000000
      mean   150076.000000  150076.000000      12.221021      26.004751
      std     86646.852011   86646.852011       7.191997      13.561004
      min         0.000000       0.000000       0.830000       1.000000
      25%     75038.000000   75038.000000       6.830000      15.000000
      50%    150076.000000  150076.000000      11.250000      26.000000
      75%    225114.000000  225114.000000      16.170000      38.000000
      max    300152.000000  300152.000000      49.830000      49.000000

                     price
      count  300153.000000
      mean    20889.660523
      std     22697.767366
      min      1105.000000
      25%      4783.000000
      50%      7425.000000
      75%     42521.000000
      max    123071.000000
```

```
[28]: df.columns
```

```
[28]: Index(['Unnamed: 0', 'index', 'airline', 'flight', 'source_city',
             'departure_time', 'stops', 'arrival_time', 'destination_city', 'class',
             'duration', 'days_left', 'price', 'columns'],
            dtype='object')
```

```
[29]: df['destination_city'].unique()
```

```
[29]: array(['Multan', 'Lahore', 'Karachi', 'GB', 'Islamabad', 'Quetta'],
            dtype=object)
```

```
[30]: df['class'].unique()
```

```
[30]: array(['Economy', 'Business'], dtype=object)
```

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

```
[31]: Unnamed: 0          0
      index               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
columns               0
dtype: int64
```

[32]:
```
import os
print(os.listdir())
```

```
['.ipynb_checkpoints', 'Airline project.ipynb', 'flight_analysis_output
(1).xlsx']
```

[33]:
```
print(df.columns.tolist())
```

```
['Unnamed: 0', 'index', 'airline', 'flight', 'source_city', 'departure_time',
'stops', 'arrival_time', 'destination_city', 'class', 'duration', 'days_left',
'price', 'columns']
```

Standardize Formats

[34]:
```
df['duration'] = df['duration'].astype(str) + 'hrs'
print(df['duration'].head())
```

```
0    2.17hrs
1    2.33hrs
2    2.17hrs
3    2.25hrs
4    2.33hrs
Name: duration, dtype: object
```

Remove Unnecessary Columns (e.g., columns):

[36]:
```
df = df.drop(['columns','Unnamed: 0'], axis=1, errors='ignore')
print(df.columns.tolist())
```

```
['index', 'airline', 'flight', 'source_city', 'departure_time', 'stops',
'arrival_time', 'destination_city', 'class', 'duration', 'days_left', 'price']
```

Exploring Data

DataFrame Type:

[37]:
```
print(type(df))
print(isinstance(df, pd.DataFrame))
```

```
<class 'pandas.core.frame.DataFrame'>
True
```

Column Data Types:

[38]:
```
print(df.dtypes)
```

```
index                 int64
airline              object
flight               object
source_city          object
departure_time       object
stops                object
arrival_time         object
destination_city     object
class                object
duration             object
days_left             int64
price               float64
dtype: object
```

Filter Object Columns:

[39]:
```
object_columns = df.select_dtypes(include=['object']).columns.tolist()
print("Object columns:", object_columns)
```

```
Object columns: ['airline', 'flight', 'source_city', 'departure_time', 'stops',
'arrival_time', 'destination_city', 'class', 'duration']
```

Filter Float columns

[43]:
```
columns = df.columns[1:5]
columns
```

[43]:
```
Index(['airline', 'flight', 'source_city', 'departure_time'], dtype='object')
```

[40]:
```
float_columns = df.select_dtypes(include= ['float']).columns.tolist()
print(f"Float Column is :{float_columns}")
```

```
Float Column is :['price']
```

Int columns

[44]:
```
int_columns = df.select_dtypes(include = ["int"]).columns.tolist()
print(f"int Column is :{int_columns}")
```

```
int Column is :['index', 'days_left']
```

Summary Statistics:

[45]:
```
print(df.describe())
print(df.describe(include='object'))
```

```
               index      days_left          price
count  300153.000000  300153.000000  300153.000000
mean   150076.000000      26.004751   20889.660523
```

```
std     86646.852011      13.561004   22697.767366
min         0.000000       1.000000    1105.000000
25%     75038.000000      15.000000    4783.000000
50%    150076.000000      26.000000    7425.000000
75%    225114.000000      38.000000   42521.000000
max    300152.000000      49.000000  123071.000000
                airline  flight source_city departure_time   stops  \
count            300153  300153      300153         300153  300153
unique                6    1561           6              6       3
top     Shaheen Airline  UK-706   Islamabad        Morning     one
freq             127859    3235       61343          71146  250863

       arrival_time destination_city    class duration
count        300153           300153   300153   300153
unique            6                6        2      476
top           Night           Multan  Economy  2.17hrs
freq          91538            59097   206666     4242
```

```
[78]: df.columns
```

```
[78]: Index(['Unnamed: 0', 'index', 'airline', 'flight', 'source_city',
             'departure_time', 'stops', 'arrival_time', 'destination_city', 'class',
             'duration', 'days_left', 'price'],
            dtype='object')
```

```
[46]: df = df.drop('Unnamed: 0', axis= 1 ,errors='ignore')
      df
```

```
[46]:          index          airline   flight source_city departure_time stops  \
      0            0              PIA  SG-8709   Islamabad        Evening  zero
      1            1              PIA  SG-8157   Islamabad  Early_Morning  zero
      2            2         Air Blue   I5-764   Islamabad  Early_Morning  zero
      3            3  Shaheen Airline   UK-995   Islamabad        Morning  zero
      4            4  Shaheen Airline   UK-963   Islamabad        Morning  zero
      …        …              …      …         …            …   …
      300148  300148  Shaheen Airline   UK-822      Quetta        Morning   one
      300149  300149  Shaheen Airline   UK-826      Quetta      Afternoon   one
      300150  300150  Shaheen Airline   UK-832      Quetta  Early_Morning   one
      300151  300151  Shaheen Airline   UK-828      Quetta  Early_Morning   one
      300152  300152  Shaheen Airline   UK-822      Quetta        Morning   one

               arrival_time destination_city     class  duration  days_left    price
      0               Night           Multan   Economy   2.17hrs          1   5953.0
      1             Morning           Multan   Economy   2.33hrs          1   5953.0
      2       Early_Morning           Multan   Economy   2.17hrs          1   5956.0
      3           Afternoon           Multan   Economy   2.25hrs          1   5955.0
      4             Morning           Multan   Economy   2.33hrs          1   5955.0
```

```
…            …              …       …        …           …      …
300148       Evening                GB  Business  10.08hrs        49  69265.0
300149         Night                GB  Business  10.42hrs        49  77105.0
300150         Night                GB  Business  13.83hrs        49  79099.0
300151       Evening                GB  Business   10.0hrs        49  81585.0
300152       Evening                GB  Business  10.08hrs        49  81585.0

[300153 rows x 12 columns]
```

Unique Values:

[47]:
```
print(df['airline'].unique())
print(df['stops'].value_counts())
```

```
['PIA' 'Air Blue' 'Shaheen Airline' 'Serene Air' 'Air Sial' 'Fly Jinnah']
stops
one            250863
zero            36004
two_or_more     13286
Name: count, dtype: int64
```

Selecting and Filtering Data

[50]:
```
selected_columns = df.iloc[:, 0:5]
print(selected_columns.head())
```

```
   index          airline   flight source_city departure_time
0      0              PIA  SG-8709    Islamabad        Evening
1      1              PIA  SG-8157    Islamabad  Early_Morning
2      2         Air Blue   I5-764    Islamabad  Early_Morning
3      3  Shaheen Airline   UK-995    Islamabad        Morning
4      4  Shaheen Airline   UK-963    Islamabad        Morning
```

[55]:
```
print(df[['airline', 'flight', 'duration', 'price']].head())
```

```
           airline   flight duration   price
0              PIA  SG-8709  2.17hrs  5953.0
1              PIA  SG-8157  2.33hrs  5953.0
2         Air Blue   I5-764  2.17hrs  5956.0
3  Shaheen Airline   UK-995  2.25hrs  5955.0
4  Shaheen Airline   UK-963  2.33hrs  5955.0
```

[56]:
```
high_price = df[df['price'] > 10000]
print(high_price[['airline', 'price']].head())
```

```
            airline    price
28              PIA  10260.0
29         Air Sial  10470.0
30       Serene Air  10575.0
```

```
31          Air Sial  10838.0
32   Shaheen Airline  12150.0
```

```
[84]: filtered = df[(df['price'] > 10000) & (df['class'] == 'Business')]
      print(filtered[['airline', 'class', 'price']].head())
```

```
          airline     class    price
206666  Fly Jinnah  Business  25612.0
206667  Fly Jinnah  Business  25612.0
206668  Fly Jinnah  Business  42220.0
206669  Fly Jinnah  Business  44450.0
206670  Fly Jinnah  Business  46690.0
```

```
[57]: df.columns
```

```
[57]: Index(['index', 'airline', 'flight', 'source_city', 'departure_time', 'stops',
             'arrival_time', 'destination_city', 'class', 'duration', 'days_left',
             'price'],
            dtype='object')
```

```
[58]: df['destination_city'].unique()
```

```
[58]: array(['Multan', 'Lahore', 'Karachi', 'GB', 'Islamabad', 'Quetta'],
            dtype=object)
```

```
[64]: new1 = df[df['destination_city'].isin (['Islamabad','Lahore','Quetta'])]
      print(new1[[ 'airline', 'flight','price']])
```

```
                airline   flight    price
9982                PIA   SG-191   7425.0
9983           Air Blue  I5-1529   7426.0
9984    Shaheen Airline   UK-815   7425.0
9985    Shaheen Airline   UK-801   7425.0
9986    Shaheen Airline   UK-819   7425.0
…                   …       …       …
295924       Fly Jinnah   AI-440  60260.0
295925       Fly Jinnah   AI-539  60260.0
295926       Fly Jinnah   AI-430  60260.0
295927       Fly Jinnah   AI-539  60260.0
295928       Fly Jinnah   AI-430  60260.0

[148796 rows x 3 columns]
```

```
[65]: df['class'].unique()
```

```
[65]: array(['Economy', 'Business'], dtype=object)
```

[66]:

```
premium = df[(df["price"]>10000) & (df['class']== "Business")]
premium[['airline', 'class', 'price']]
```

[66]:

```
                airline     class    price
206666       Fly Jinnah  Business  25612.0
206667       Fly Jinnah  Business  25612.0
206668       Fly Jinnah  Business  42220.0
206669       Fly Jinnah  Business  44450.0
206670       Fly Jinnah  Business  46690.0
…                   …         …       …
300148  Shaheen Airline  Business  69265.0
300149  Shaheen Airline  Business  77105.0
300150  Shaheen Airline  Business  79099.0
300151  Shaheen Airline  Business  81585.0
300152  Shaheen Airline  Business  81585.0

[93487 rows x 3 columns]
```

[94]:

```
Economy =df[(df['price']<10000) & (df['class']=='Economy')]
print(Economy[['airline','class','price']])
```

```
                airline    class   price
0                   PIA  Economy  5953.0
1                   PIA  Economy  5953.0
2              Air Blue  Economy  5956.0
3       Shaheen Airline  Economy  5955.0
4       Shaheen Airline  Economy  5955.0
…                   …        …      …
206661  Shaheen Airline  Economy  7697.0
206662  Shaheen Airline  Economy  7709.0
206663  Shaheen Airline  Economy  8640.0
206664  Shaheen Airline  Economy  8640.0
206665  Shaheen Airline  Economy  8640.0

[173653 rows x 3 columns]
```

[96]:

```
Economy.describe()
```

[96]:

```
               index      days_left          price
count  173653.000000  173653.000000  173653.000000
mean   102819.423569      29.129223    5248.050906
std     59696.424959      12.205192    1898.246310
min         0.000000       1.000000    1105.000000
25%     50622.000000      20.000000    3999.000000
50%    101455.000000      30.000000    5176.000000
75%    155458.000000      39.000000    6450.000000
max    206665.000000      49.000000    9999.000000
```

[67]:

```
df['price_days'] = df['price'] * df['days_left']
print(df[['price', 'days_left', 'price_days']].head())
```

```
    price  days_left  price_days
0  5953.0          1      5953.0
1  5953.0          1      5953.0
2  5956.0          1      5956.0
3  5955.0          1      5955.0
4  5955.0          1      5955.0
```

[97]:

```
df['price_days'] = df['price'] * df['days_left']
print(df[['price','days_left','price_days',]])
```

```
          price  days_left  price_days
0        5953.0          1      5953.0
1        5953.0          1      5953.0
2        5956.0          1      5956.0
3        5955.0          1      5955.0
4        5955.0          1      5955.0
…         …        …         …
300148  69265.0         49   3393985.0
300149  77105.0         49   3778145.0
300150  79099.0         49   3875851.0
300151  81585.0         49   3997665.0
300152  81585.0         49   3997665.0

[300153 rows x 3 columns]
```

[68]:

```
new_column_data = df['price'] * df['days_left']
df.insert(loc=2, column='price_days_inserted', value=new_column_data)
print(df.head())
```

```
   index         airline  price_days_inserted   flight source_city  \
0      0             PIA               5953.0  SG-8709   Islamabad
1      1             PIA               5953.0  SG-8157   Islamabad
2      2        Air Blue               5956.0   I5-764   Islamabad
3      3  Shaheen Airline               5955.0   UK-995   Islamabad
4      4  Shaheen Airline               5955.0   UK-963   Islamabad

  departure_time stops   arrival_time destination_city    class duration  \
0        Evening  zero          Night          Multan  Economy  2.17hrs
1  Early_Morning  zero        Morning          Multan  Economy  2.33hrs
2  Early_Morning  zero  Early_Morning          Multan  Economy  2.17hrs
3        Morning  zero      Afternoon          Multan  Economy  2.25hrs
4        Morning  zero        Morning          Multan  Economy  2.33hrs

   days_left    price  price_days
0          1   5953.0      5953.0
```

```
1           1  5953.0      5953.0
2           1  5956.0      5956.0
3           1  5955.0      5955.0
4           1  5955.0      5955.0
```

[126]:

```
y= df.columns.tolist()
print(y)
for i in y:
    print(i)
```

```
['index', 'airline', 'price_days_inserted', 'flight', 'source_city',
'departure_time', 'stops', 'arrival_time', 'destination_city', 'class',
'duration', 'days_left', 'price', 'price_days']
index
airline
price_days_inserted
flight
source_city
departure_time
stops
arrival_time
destination_city
class
duration
days_left
price
price_days
```

[129]:

```
avg_price = df.groupby('airline')['price'].mean()
print(avg_price)
```

```
airline
Air Blue             4091.072742
Air Sial             5324.216303
Fly Jinnah          23507.019112
PIA                  6179.278881
Serene Air           5652.007595
Shaheen Airline     30396.536302
Name: price, dtype: float64
```

[132]:

```
df.columns
```

[132]:

```
Index(['index', 'airline', 'price_days_inserted', 'flight', 'source_city',
       'departure_time', 'stops', 'arrival_time', 'destination_city', 'class',
       'duration', 'days_left', 'price', 'price_days'],
      dtype='object')
```

[69]:

```
# Convert duration to numeric
df['duration'] = df['duration'].str.replace('hrs', '').astype(float)
# Group and aggregate
stats = df.groupby('class').agg({'price': 'mean', 'duration': 'mean'})
print(stats)
```

```
                price   duration
class
Business  52540.081124  13.704274
Economy    6572.342383  11.550060
```

[140]:

```
by_city_class = df.groupby(['source_city', 'class'])['price'].mean()
print(by_city_class)
```

```
source_city  class
GB           Business    52740.521251
             Economy      6358.920050
Islamabad    Business    48697.983457
             Economy      6288.585423
Karachi      Business    53723.539687
             Economy      6582.057019
Lahore       Business    56607.860536
             Economy      7458.632354
Multan       Business    50370.466200
             Economy      6235.770619
Quetta       Business    54140.530529
             Economy      6606.279699
Name: price, dtype: float64
```

[84]:

```
df.select_dtypes(include = ['float']).columns.tolist()
```

[84]:

```
['price_days_inserted', 'duration', 'price', 'price_days']
```

[141]:

```
df.columns
```

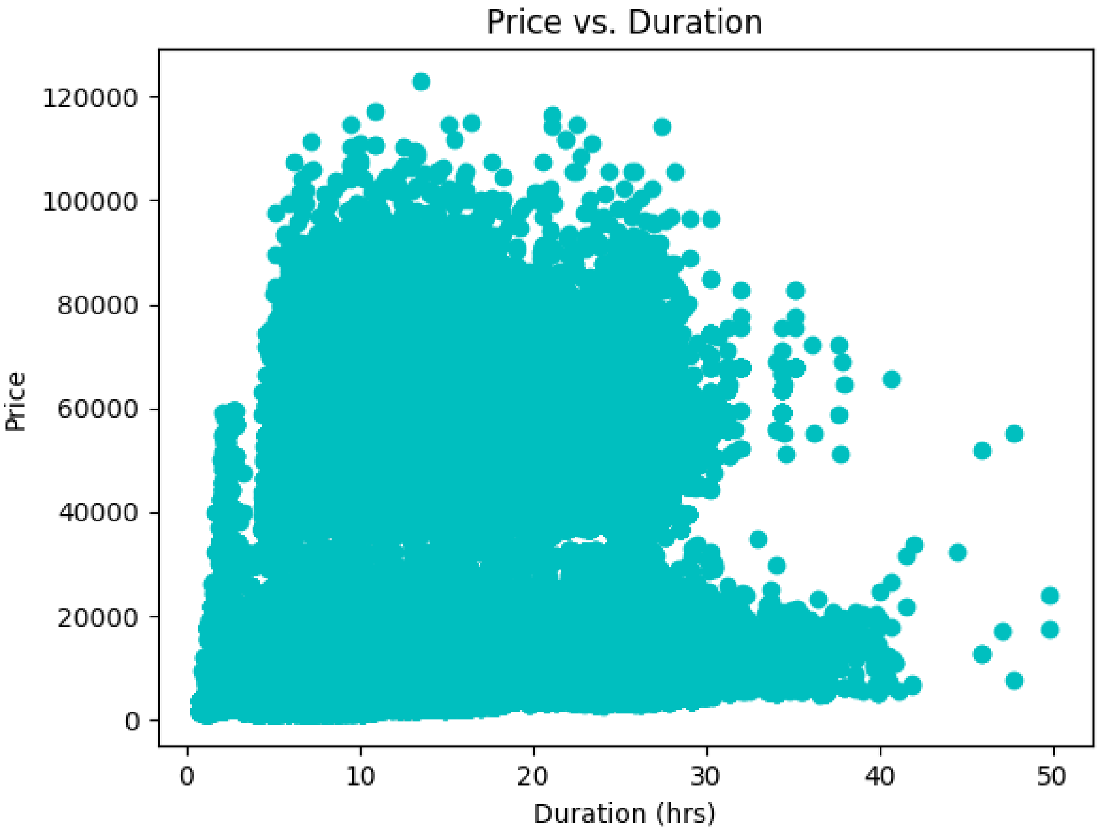
[141]:

```
Index(['index', 'airline', 'price_days_inserted', 'flight', 'source_city',
       'departure_time', 'stops', 'arrival_time', 'destination_city', 'class',
       'duration', 'days_left', 'price', 'price_days'],
      dtype='object')
```

Visualizing Data • Price vs. Duration Scatter Plot:

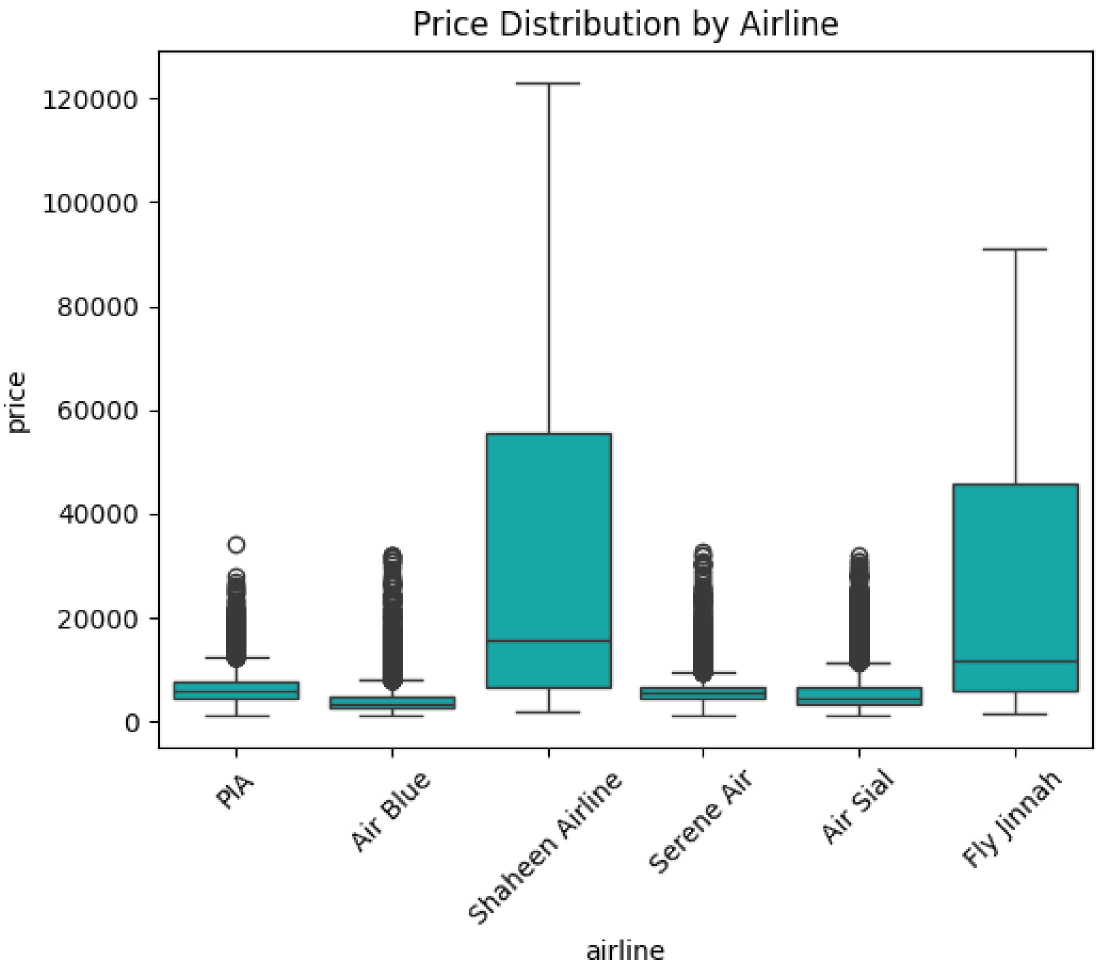
[95]:

```
import matplotlib.pyplot as plt
c = ["g",'r','b']
plt.scatter(df['duration'].astype(float), df['price'],color = 'c')
plt.xlabel('Duration (hrs)')
plt.ylabel('Price')
plt.title('Price vs. Duration')
plt.show()
```



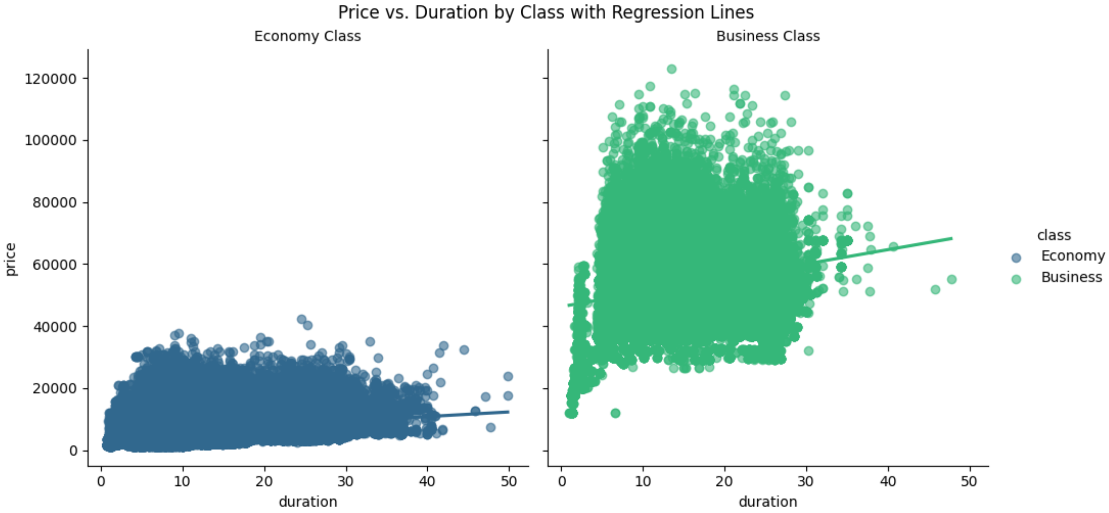
[99]:

```
import seaborn as sns
c =('g','r','b','m','p')
sns.boxplot(x='airline', y='price', data=df, color = 'c')
plt.title('Price Distribution by Airline')
plt.xticks(rotation=45)
plt.show()
```

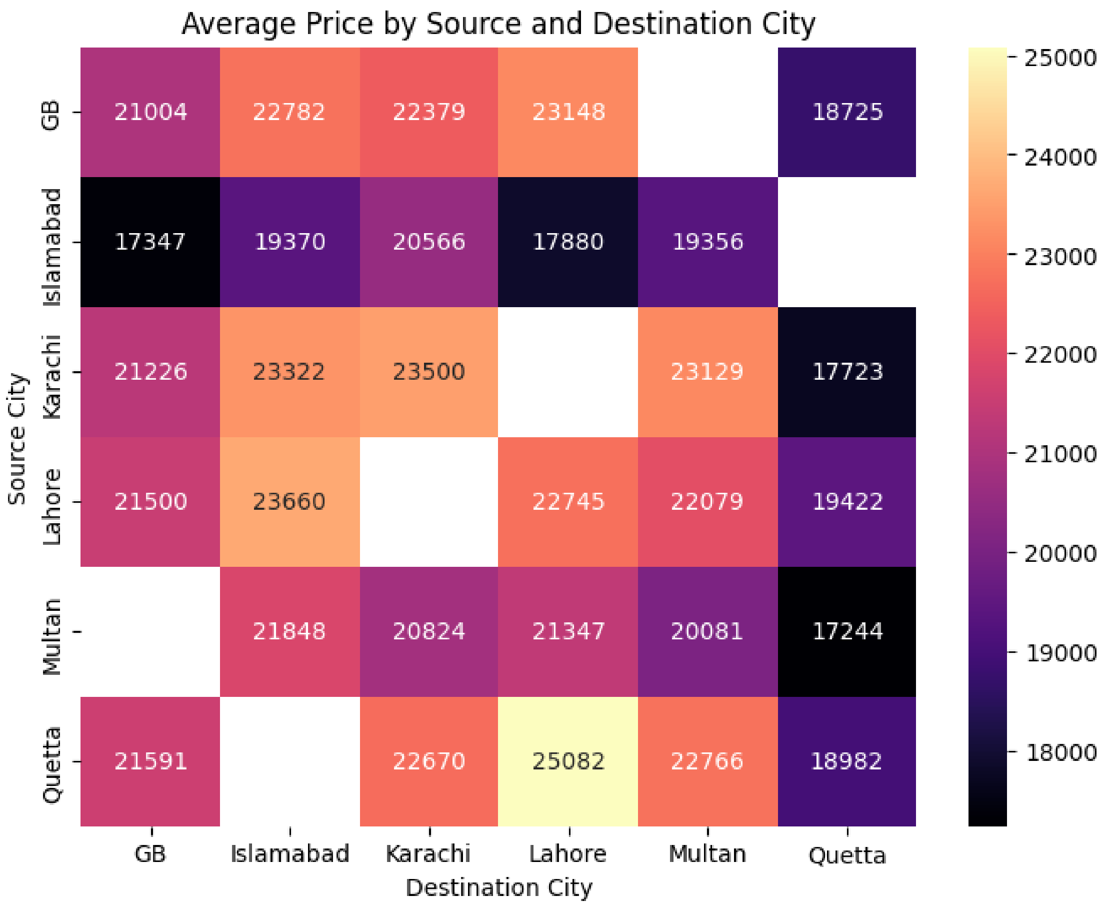


[100]:

```
# FacetGrid with scatter and regression
g = sns.FacetGrid(df, col='class', hue='class', palette='viridis', height=5)
g.map(sns.regplot, 'duration', 'price', scatter_kws={'alpha':0.6})
g.add_legend()
g.set_titles(col_template='{col_name} Class')
plt.suptitle('Price vs. Duration by Class with Regression Lines', y=1.02)
plt.show()
```



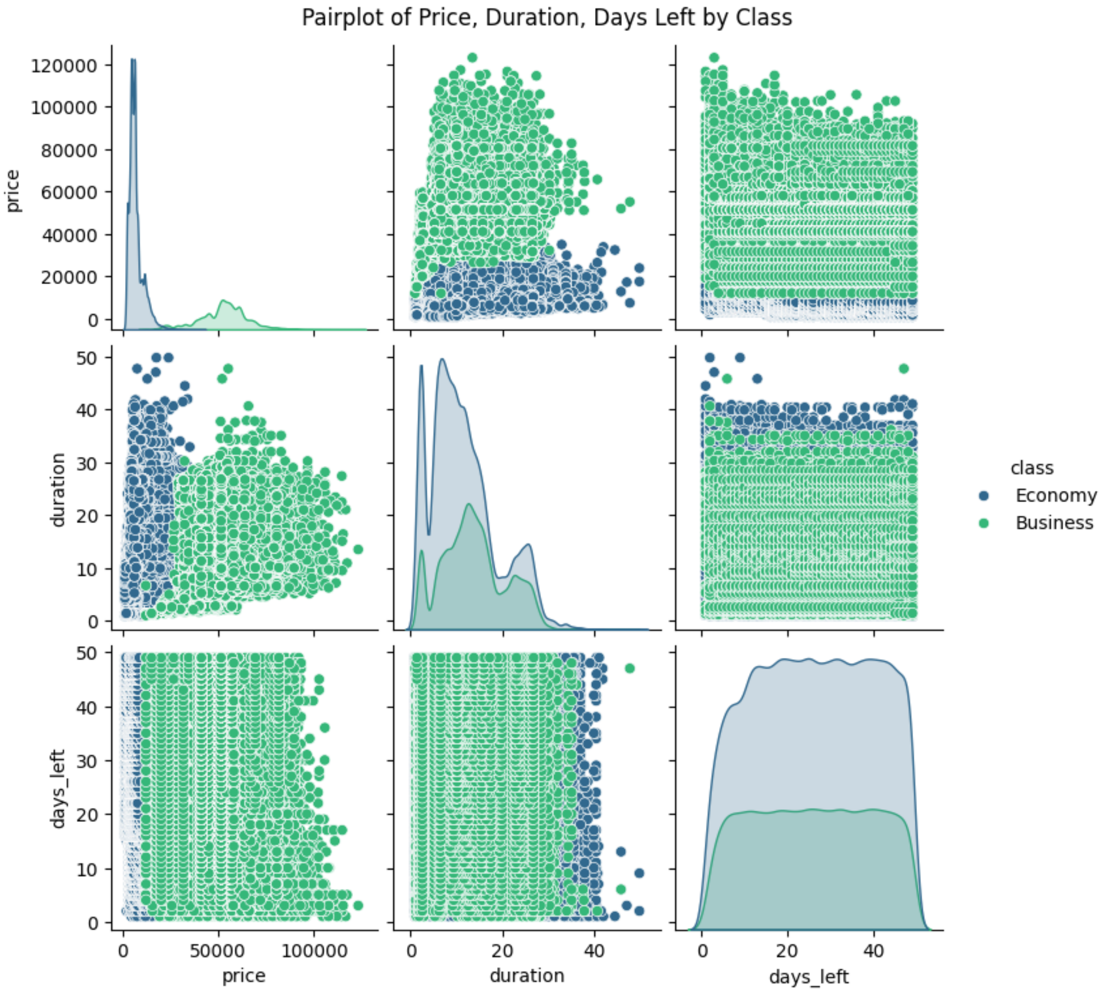
[133]:
```
# Pivot data for heatmap
pivot_price = df.pivot_table(values='price', index='source_city',␣
 ↪columns='destination_city', aggfunc='mean')

# Create heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(pivot_price, annot=True, fmt='.0f', cmap='magma')
plt.title('Average Price by Source and Destination City')
plt.xlabel('Destination City')
plt.ylabel('Source City')
plt.show()
```



Description: Visualizes mean prices as a matrix, with darker colors indicating higher prices. Annotations show exact values.

Pairplot for Numeric Variables (Price, Duration, Days Left)

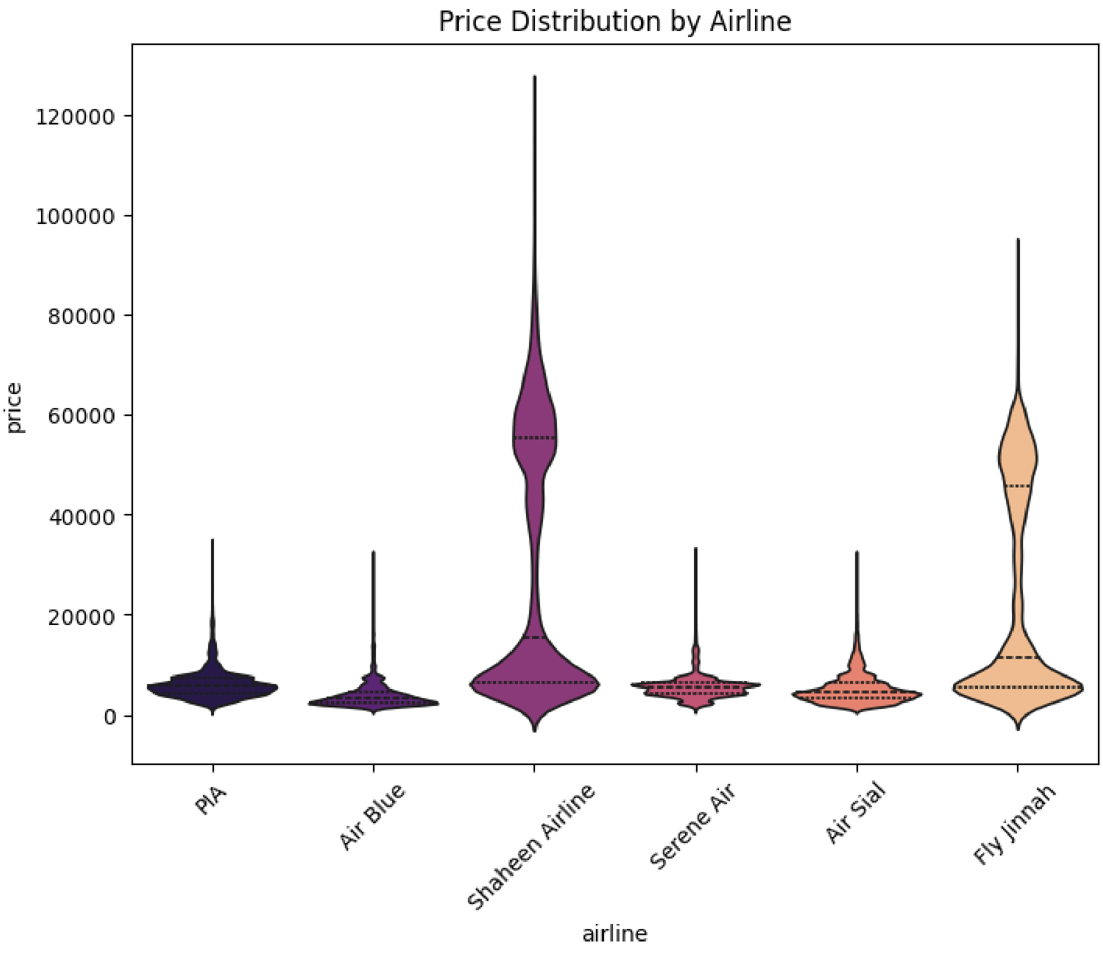
[109]:
```
# Pairplot with hue by class
sns.pairplot(df[['price', 'duration', 'days_left', 'class']], hue='class',␣
 ↪palette='viridis')
plt.suptitle('Pairplot of Price, Duration, Days Left by Class', y=1.02)
plt.show()
```



Description: Shows pairwise relationships (e.g., price vs. duration) and distributions, with colors distinguishing Economy vs. Business. Useful for correlation spotting

Violin Plot for Price Distribution by Airline

[113]:
```
# Violin plot with hue
plt.figure(figsize=(8, 6))
sns.violinplot(x='airline', y='price', hue='airline', data=df, palette='magma',␣
  ↪inner='quartile', legend=False)
plt.title('Price Distribution by Airline')
plt.xticks(rotation=45)
plt.show()
```

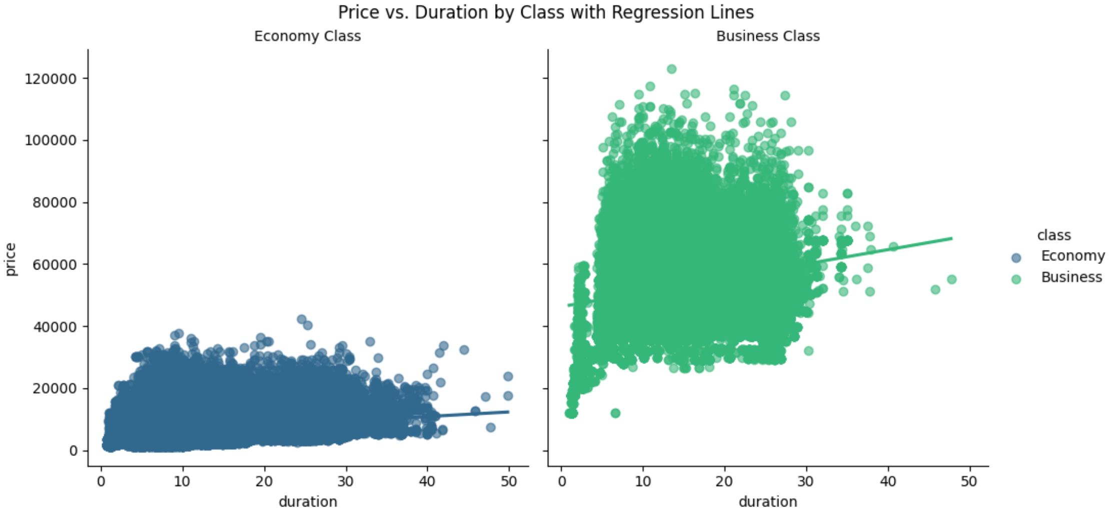


Description: Displays price density and quartiles for each airline, highlighting variability and outliers

FacetGrid: Scatter Plot of Price vs. Duration by Class

[114]:

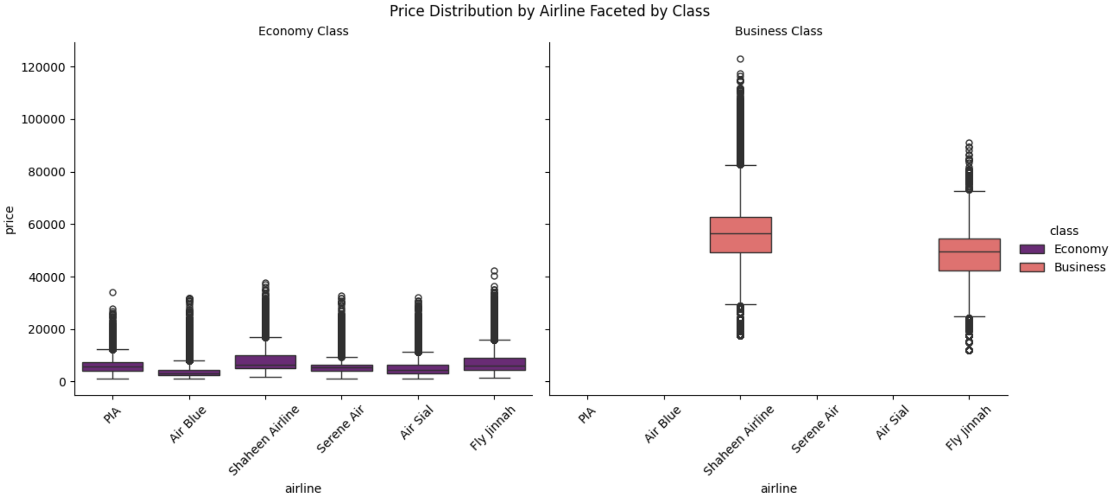
```
# FacetGrid with scatter and regression
g = sns.FacetGrid(df, col='class', hue='class', palette='viridis', height=5)
g.map(sns.regplot, 'duration', 'price', scatter_kws={'alpha':0.6})
g.add_legend()
g.set_titles(col_template='{col_name} Class')
plt.suptitle('Price vs. Duration by Class with Regression Lines', y=1.02)
plt.show()
```



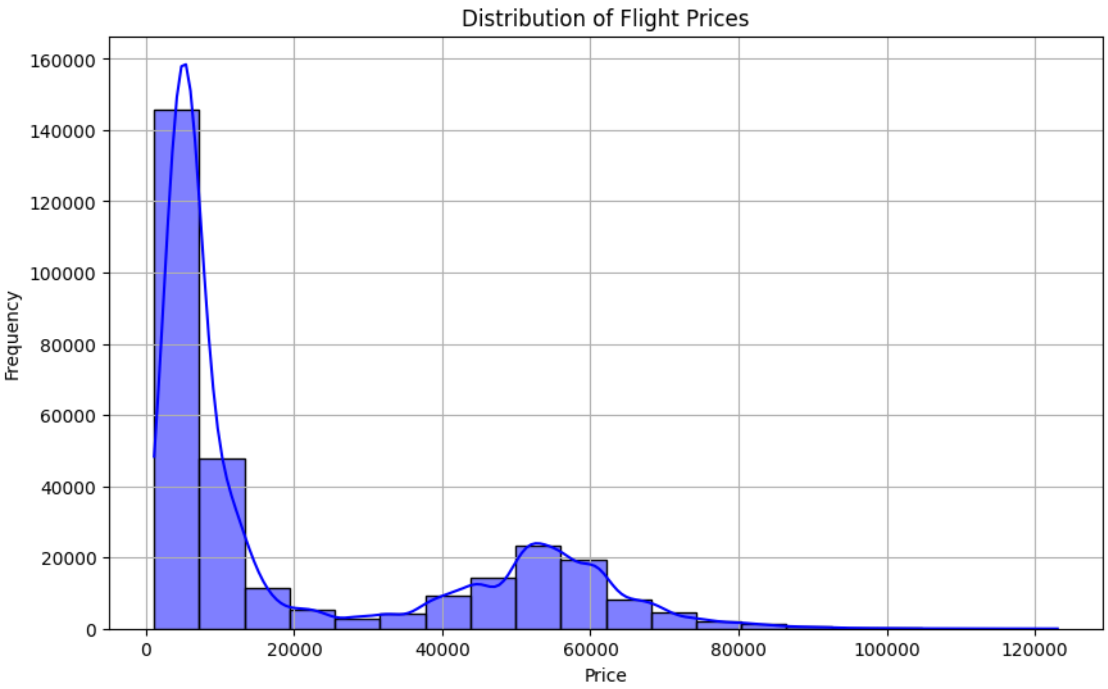
Description: Separate panels for each class showing price-duration relationship with trend lines, useful for comparing segments.

Time Series Line Plot: Average Price by Departure Hour

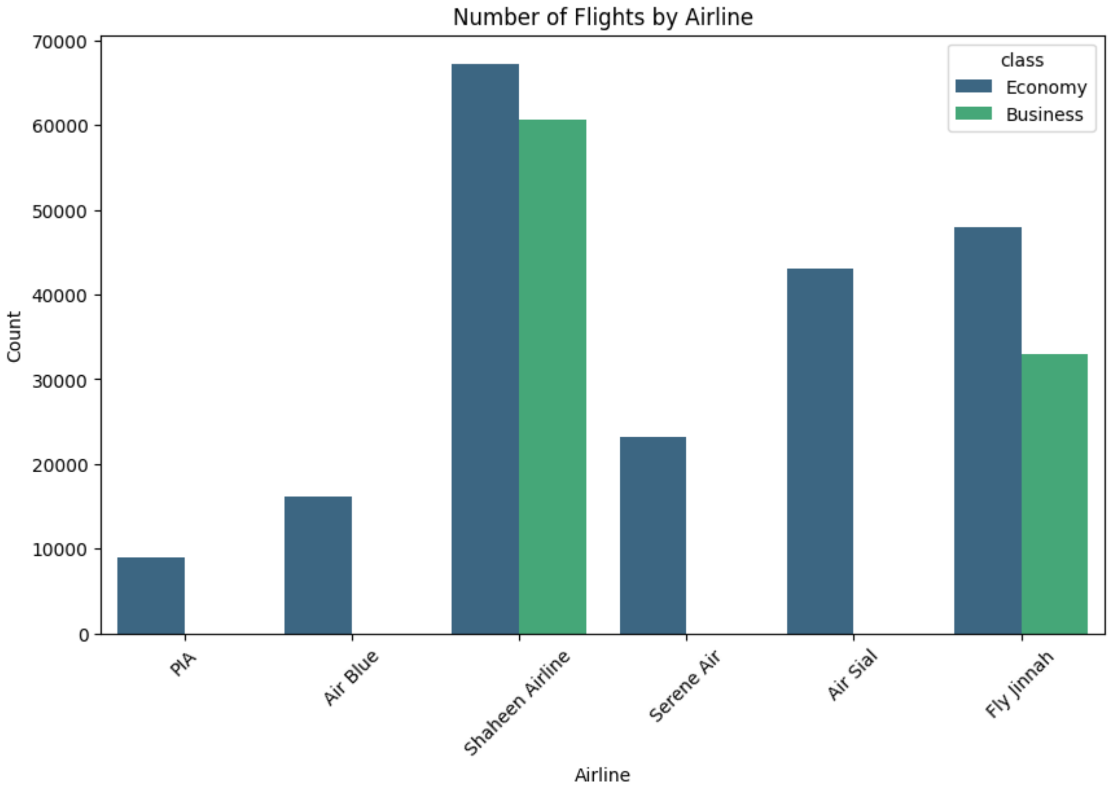
[132]:
```
# Faceted boxplot
g = sns.catplot(x='airline', y='price', col='class', data=df,␣
↪kind='box',hue='class', palette='magma', height=5, aspect=1.2)
g.set_xticklabels(rotation=45)
g.set_titles(col_template='{col_name} Class')
plt.suptitle('Price Distribution by Airline Faceted by Class', y=1.02)
plt.show()
```



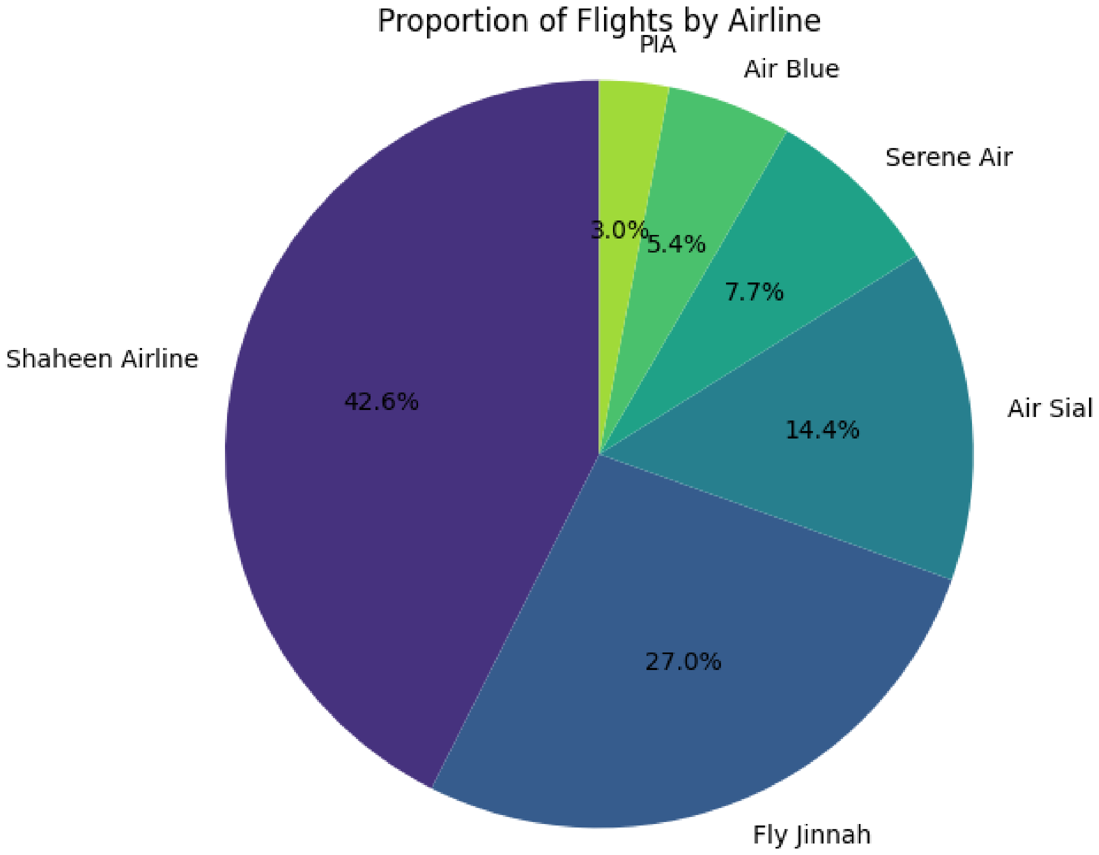
[121]:
```
# Histogram for price
plt.figure(figsize=(10, 6))
sns.histplot(df['price'], bins=20, kde=True, color='blue')
plt.title('Distribution of Flight Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



[124]:
```
# Bar plot for flight counts by airline
plt.figure(figsize=(10, 6))
sns.countplot(x='airline', data=df,hue= 'class', palette='viridis')
plt.title('Number of Flights by Airline')
plt.xlabel('Airline')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



```
[131]: airline_counts = df['airline'].value_counts()
       plt.figure(figsize=(6, 6))
       plt.pie(airline_counts, labels=airline_counts.index, autopct='%1.1f%%',␣
        ↪startangle=90, colors=sns.color_palette('viridis', len(airline_counts)))
       plt.title('Proportion of Flights by Airline')
       plt.axis('equal')  # Equal aspect ratio for circular pie
       plt.show()
```



Proportion of Flights by Airline

[ ]: