statistics-lab-8-assignment

November 27, 2023

Q1. Take death_rate_of_countries_and_its_causes dataset. Do the EDA and analyse following:

a) Top 10 countries with the highest and lowest air pollution.

[3]:
```
import pandas as pd

data = pd.read_csv('death_rate_of_countries_and_its_causes.csv')
print(data.head())
print("*"*100)

#top 10 highest air pollution
top_10_highest_air_pollution = data.nlargest(10, 'Outdoor air␣
 ↪pollution')[['Entity', 'Outdoor air pollution']]
print("Top 10 countries with the highest outdoor air pollution:")
print(top_10_highest_air_pollution)
print("*"*100)

#top 10 lowest air pollution
top_10_lowest_air_pollution = data.nsmallest(10, 'Outdoor air␣
 ↪pollution')[['Entity', 'Outdoor air pollution']]
print("\nTop 10 countries with the lowest outdoor air pollution:")
print(top_10_lowest_air_pollution)
```

```
       Entity Code  Year  Outdoor air pollution  \
0  Afghanistan  AFG  1990                   3169
1  Afghanistan  AFG  1991                   3222
2  Afghanistan  AFG  1992                   3395
3  Afghanistan  AFG  1993                   3623
4  Afghanistan  AFG  1994                   3788

   High systolic blood pressure  Diet high in sodium   \
0                         25633                  1045
1                         25872                  1055
2                         26309                  1075
3                         26961                  1103
4                         27658                  1134

   Diet low in whole grains  Alochol use  Diet low in fruits  \
```

```
0                     7077          356               3185
1                     7149          364               3248
2                     7297          376               3351
3                     7499          389               3480
4                     7698          399               3610

   Unsafe water source  …  High body mass index  Unsafe sanitation  \
0                 3702  …                  9518               2798
1                 4309  …                  9489               3254
2                 5356  …                  9528               4042
3                 7152  …                  9611               5392
4                 7192  …                  9675               5418

   No access to handwashing facility  Drug use  Low bone mineral density  \
0                               4825       174                       389
1                               5127       188                       389
2                               5889       211                       393
3                               7007       232                       411
4                               7421       247                       413

   Vitamin A deficiency  Child stunting  Discontinued breastfeeding  \
0                  2016            7686                         107
1                  2056            7886                         121
2                  2100            8568                         150
3                  2316            9875                         204
4                  2665           11031                         204

   Non-exclusive breastfeeding  Iron deficiency
0                         2216              564
1                         2501              611
2                         3053              700
3                         3726              773
4                         3833              812

[5 rows x 31 columns]
**************************************************************************
********************
Top 10 countries with the highest outdoor air pollution:
     Entity  Outdoor air pollution
6629  World                4506193
6628  World                4341493
6627  World                4238086
6626  World                4218230
6625  World                4191433
6624  World                4100011
6623  World                4009365
6622  World                3887311
6621  World                3778458
```

```
6620  World                3673797
****************************************************************************
********************
```

```
Top 10 countries with the lowest outdoor air pollution:
     Entity  Outdoor air pollution
4200   Niue                      0
4201   Niue                      0
4202   Niue                      0
4203   Niue                      0
4204   Niue                      0
4205   Niue                      0
4206   Niue                      0
4210   Niue                      0
4211   Niue                      0
4212   Niue                      0
```

b)Top 10 countries with the highest and lowest blood pressure.

[4]:
```
#Top 10 countries with the highest blood pressure
top_10_highest_blood_pressure = data.nlargest(10, 'High systolic blood␣
 ↪pressure')[['Entity', 'High systolic blood pressure']]
print("Top 10 countries with the highest blood pressure:")
print(top_10_highest_blood_pressure)
print("*"*100)

#Top 10 countries with the lowest blood pressure
top_10_lowest_blood_pressure = data.nsmallest(10, 'High systolic blood␣
 ↪pressure')[['Entity', 'High systolic blood pressure']]
print("\nTop 10 countries with the lowest blood pressure:")
print(top_10_lowest_blood_pressure)
```

```
Top 10 countries with the highest blood pressure:
     Entity  High systolic blood pressure
6629  World                      10845595
6628  World                      10589176
6627  World                      10351721
6626  World                      10190860
6625  World                       9999740
6624  World                       9765811
6623  World                       9622330
6622  World                       9470980
6621  World                       9324965
6620  World                       9181355
****************************************************************************
********************
```

```
Top 10 countries with the lowest blood pressure:
```

```
       Entity  High systolic blood pressure
6000  Tokelau                             2
6020  Tokelau                             2
6021  Tokelau                             2
6022  Tokelau                             2
6023  Tokelau                             2
6024  Tokelau                             2
6025  Tokelau                             2
6026  Tokelau                             2
6001  Tokelau                             3
6002  Tokelau                             3
```

c) Top 10 countries with the highest and lowest low whole grain diet.

[5]:
```
# Top 10 countries with the highest low whole grain diet
top_10_highest_low_whole_grain_diet = data.nlargest(10, 'Diet low in whole␣
 ↪grains')[['Entity', 'Diet low in whole grains']]
print("Top 10 countries with the highest low whole grain diet:")
print(top_10_highest_low_whole_grain_diet)
print("*"*100)

# Top 10 countries with the lowest low whole grain diet
top_10_lowest_low_whole_grain_diet = data.nsmallest(10, 'Diet low in whole␣
 ↪grains')[['Entity', 'Diet low in whole grains']]
print("\nTop 10 countries with the lowest low whole grain diet:")
print(top_10_lowest_low_whole_grain_diet)
```

```
Top 10 countries with the highest low whole grain diet:
     Entity  Diet low in whole grains
6629  World                   1844836
6628  World                   1802323
6627  World                   1758037
6626  World                   1730299
6625  World                   1702413
6624  World                   1659528
6623  World                   1639467
6622  World                   1614793
6621  World                   1591653
6620  World                   1569687
****************************************************************************
********************

Top 10 countries with the lowest low whole grain diet:
       Entity  Diet low in whole grains
6023  Tokelau                         0
6024  Tokelau                         0
6025  Tokelau                         0
6026  Tokelau                         0
```

```
4200     Niue                          1
4201     Niue                          1
4202     Niue                          1
4203     Niue                          1
4204     Niue                          1
4205     Niue                          1
```

d) Top 10 countries with the highest and lowest alcohol use.

[6]:
```
# Top 10 countries with the highest alcohol use
top_10_highest_alcohol_use = data.nlargest(10, 'Alochol use')[['Entity',␣
 ↪'Alochol use']]
print("Top 10 countries with the highest alcohol use:")
print(top_10_highest_alcohol_use)
print("*"*100)

# Top 10 countries with the lowest alcohol use
top_10_lowest_alcohol_use = data.nsmallest(10, 'Alochol use')[['Entity',␣
 ↪'Alochol use']]
print("\nTop 10 countries with the lowest alcohol use:")
print(top_10_lowest_alcohol_use)
```

```
Top 10 countries with the highest alcohol use:
     Entity  Alochol use
6629  World      2441973
6628  World      2393321
6627  World      2355671
6626  World      2332374
6625  World      2304268
6624  World      2270896
6623  World      2258003
6620  World      2249855
6622  World      2247304
6621  World      2235098
****************************************************************************************
********************

Top 10 countries with the lowest alcohol use:
     Entity  Alochol use
4213   Niue            0
4214   Niue            0
4215   Niue            0
4216   Niue            0
4217   Niue            0
4218   Niue            0
4219   Niue            0
4220   Niue            0
4221   Niue            0
```

```
4222   Niue            0
```

e) Top 10 countries with the highest and lowest diet low in fruits.

[7]:
```
# Top 10 countries with the highest diet low in fruits
top_10_highest_low_fruit_diet = data.nlargest(10, 'Diet low in␣
 ↪fruits')[['Entity', 'Diet low in fruits']]
print("Top 10 countries with the highest diet low in fruits:")
print(top_10_highest_low_fruit_diet)
print("*"*100)

# Top 10 countries with the lowest diet low in fruits
top_10_lowest_low_fruit_diet = data.nsmallest(10, 'Diet low in␣
 ↪fruits')[['Entity', 'Diet low in fruits']]
print("\nTop 10 countries with the lowest diet low in fruits:")
print(top_10_lowest_low_fruit_diet)
```

```
Top 10 countries with the highest diet low in fruits:
     Entity  Diet low in fruits
6629  World             1046015
6628  World             1027421
6627  World             1008138
6626  World              996499
6625  World              983682
6624  World              967794
6623  World              964259
6622  World              959393
6621  World              954611
6620  World              950119
****************************************************************************
********************

Top 10 countries with the lowest diet low in fruits:
       Entity  Diet low in fruits
6000  Tokelau                   0
6001  Tokelau                   0
6002  Tokelau                   0
6003  Tokelau                   0
6004  Tokelau                   0
6005  Tokelau                   0
6006  Tokelau                   0
6007  Tokelau                   0
6008  Tokelau                   0
6009  Tokelau                   0
```

2. Explore predictive modelling techniques on inbuilt datasets.

i) Classification Example using the Iris Dataset:

[12]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier,DecisionTreeRegressor
from sklearn.metrics import accuracy_score, classification_report

iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

model =  DecisionTreeClassifier(random_state=42)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of the classification in %: ",accuracy * 100)
cr = classification_report(y_test,y_pred)
print("*"*60)
print(cr)
```

```
Accuracy of the classification in %:  100.0
************************************************************
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

ii) Regression using the Diabetes Dataset:

[14]:
```
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

diabetes = load_diabetes()
X = diabetes.data
```

```
y = diabetes.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=34)

dr_model = DecisionTreeRegressor(random_state=42)
dr_model.fit(X_train, y_train)

y_pred = dr_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("root mean square error : ", mse)
r2 = r2_score(y_test, y_pred)

N = len(y_test)
p = X_test.shape[1]
adj_r2 = 1 - ((1 - r2) * (N - 1) / (N - p - 1))
print("Adjusted R2: ", adj_r2)
```

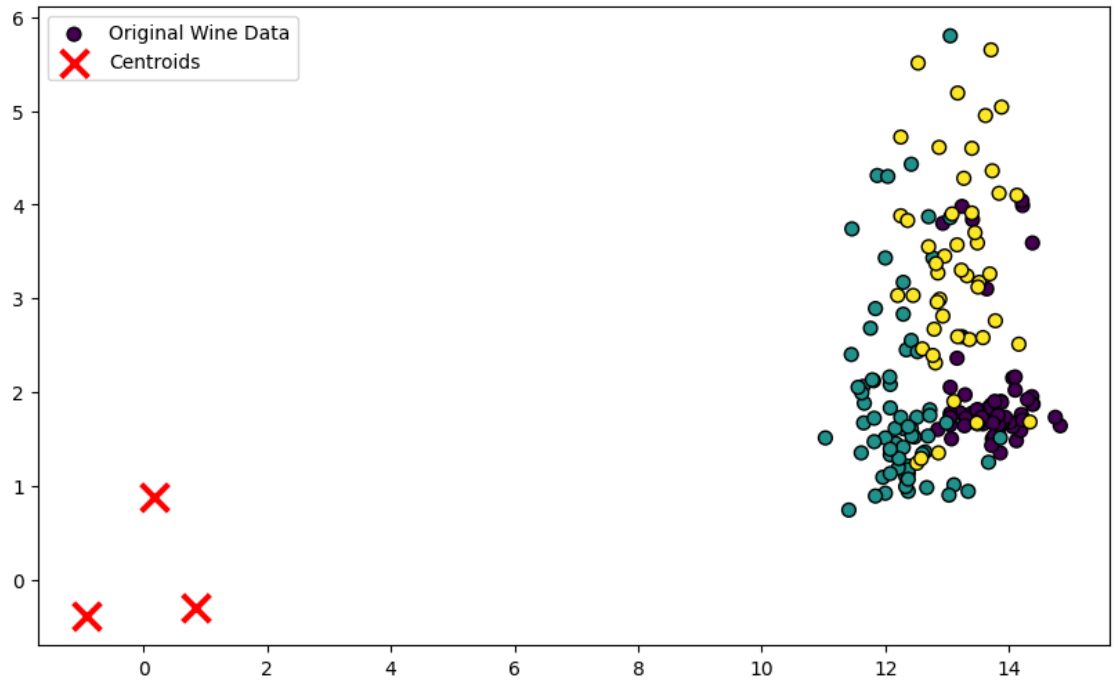
```
root mean square error :  6963.011235955056
Adjusted R2:  -0.48485660027559496
```

iii)Clustering using the Wine Dataset:

[18]:
```
from sklearn.datasets import load_wine
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

wine = load_wine()
X = wine.data
y = wine.target

scaler = StandardScaler()
X_std = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=3, random_state=40)
kmeans.fit(X_std)

plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, edgecolors='k', label="Original Wine␣
 ↪Data")
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,␣
 ↪1],marker='x', s=200, linewidth=3, color="red", label="Centroids")
plt.legend()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
```

```
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```



Q.3 Use Flight_data dataset. Find the factors affecting flight delay.

a) Show the number of flights with delay more than 60 minutes.

[23]:
```
import pandas as pd

flight_data = pd.read_csv('Flight_data.csv')
print(flight_data.head())
print("*"*100)

# Number of flights with delay more than 60 minutes
num_flights_delay = flight_data[flight_data['Delay Minutes'] > 60].shape[0]
print("Number of flights with delay more than 60 minutes:", num_flights_delay)
```

```
       Departure City    Arrival City       Departure Date  Flight Duration  \
0          Wilsonstad  Lake Johnmouth  2023-05-02 20:11:09          1.27581
1           New Brent      Port Wanda  2023-04-21 00:10:14          1.27581
2  South Samanthaberg  Lake Meganside  2023-05-12 15:16:31          0.72111
3      Lake Gracefurt       Jamesberg  2023-06-13 20:53:09         -0.94299
4         Owenborough     Kelleymouth  2023-05-15 23:06:14         -0.38829

   Delay Minutes  Customer ID              Name Booking Class  \
```

```
0             120         3769      Daniel Oliver      Business
1              35         3529       Deborah Hall       Economy
2              67         1303          Mary York       Economy
3              72         2965  Christina Sanchez       Economy
4             101         8779       Dustin Owens       Economy

  Frequent Flyer Status    Route  Ticket Price  Competitor Price    Demand  \
0                  Gold  MEL-BNE    370.638128        382.947396 -0.932755
1              Platinum  BNE-SYD    114.529016        394.583641 -1.005569
2              Platinum  MEL-BNE    164.468018        479.832444  1.761384
3                  Gold  BNE-SYD    318.903167        286.301632 -0.520139
4                Silver  BNE-SYD    389.971051        407.463316 -0.665768

  Origin Destination  Profitability  Loyalty Points  Churned
0    MEL         LHR       0.632226            4245     True
1    MEL         SIN       1.265026             833     True
2    MEL         LAX       1.141651            2568     True
3    MEL         LAX       1.129291             284     True
4    BNE         SIN       1.218239            2805     True
********************************************************************************
********************
Number of flights with delay more than 60 minutes: 47
```

b)Take the average flight price.

[24]:
```
average_flight_price = flight_data['Ticket Price'].mean()
print("Average Flight Price:", average_flight_price)
```

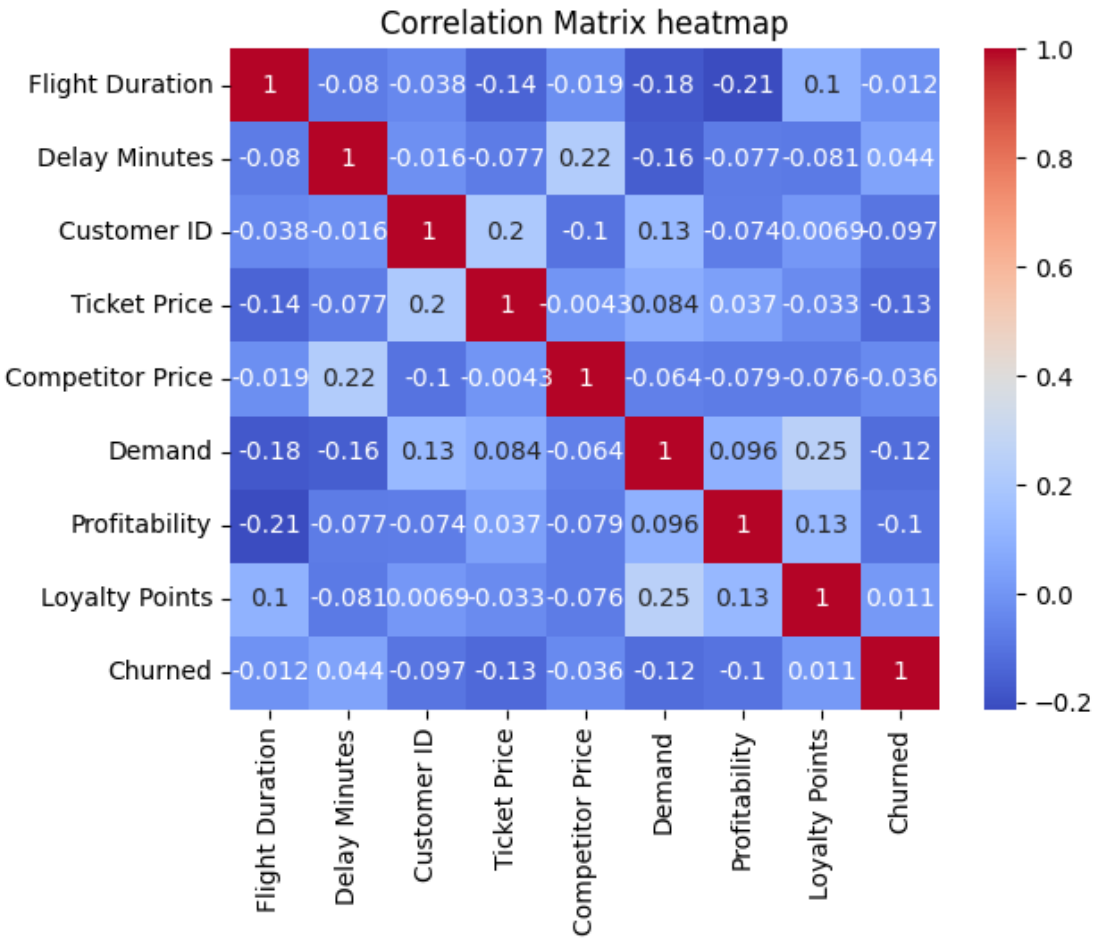
```
Average Flight Price: 303.33716947201077
```

c)Find the correlation matrix of all columns.

[32]:
```
import seaborn as sns

correlation_matrix = flight_data.corr()
print("Correlation Matrix: ", correlation_matrix)

print("*"*100)
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix heatmap')
plt.show()
```

```
<ipython-input-32-f5dbd70afb15>:3: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  correlation_matrix = flight_data.corr()

Correlation Matrix:                   Flight Duration  Delay Minutes  Customer
ID  Ticket Price  \
```

```
Flight Duration          1.000000      -0.079940    -0.038388     -0.143712
Delay Minutes           -0.079940       1.000000    -0.016202     -0.076690
Customer ID             -0.038388      -0.016202     1.000000      0.200419
Ticket Price            -0.143712      -0.076690     0.200419      1.000000
Competitor Price        -0.019251       0.221503    -0.101162     -0.004318
Demand                  -0.176788      -0.164059     0.129577      0.083749
Profitability           -0.213067      -0.077177    -0.073501      0.036509
Loyalty Points           0.101041      -0.080907     0.006875     -0.032742
Churned                 -0.012225       0.044465    -0.096623     -0.131737

                  Competitor Price    Demand  Profitability  Loyalty Points  \
Flight Duration          -0.019251 -0.176788      -0.213067        0.101041
Delay Minutes             0.221503 -0.164059      -0.077177       -0.080907
Customer ID              -0.101162  0.129577      -0.073501        0.006875
Ticket Price             -0.004318  0.083749       0.036509       -0.032742
Competitor Price          1.000000 -0.064048      -0.079324       -0.076321
Demand                   -0.064048  1.000000       0.095876        0.247524
Profitability            -0.079324  0.095876       1.000000        0.125386
Loyalty Points           -0.076321  0.247524       0.125386        1.000000
Churned                  -0.036213 -0.121961      -0.103235        0.011370

                   Churned
Flight Duration  -0.012225
Delay Minutes     0.044465
Customer ID      -0.096623
Ticket Price     -0.131737
Competitor Price -0.036213
Demand           -0.121961
Profitability    -0.103235
Loyalty Points    0.011370
Churned           1.000000
********************************************************************************
********************
```



d)Take the count of different booking class.

[29]:
```
booking_class_counts = flight_data['Booking Class'].value_counts()
print("Count of Different Booking Classes:",booking_class_counts)
```

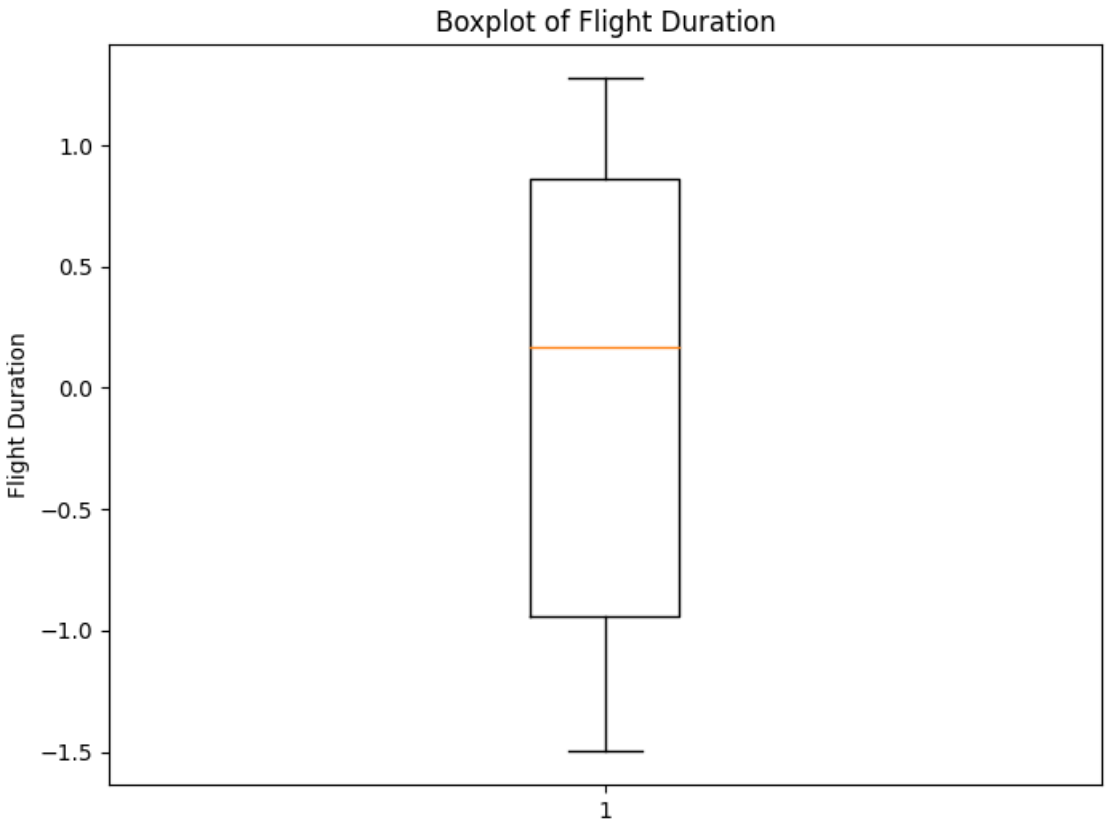
```
Count of Different Booking Classes: Economy     36
First       34
Business    30
Name: Booking Class, dtype: int64
```

e)Find outliers in flight duration.

[31]:
```
import matplotlib.pyplot as plt

# calculate Q1, Q2 & IQR
Q1 = flight_data['Flight Duration'].quantile(0.25)
Q3 = flight_data['Flight Duration'].quantile(0.75)
IQR = Q3 - Q1
```

```
#bound of outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

#outliers
outliers = flight_data[(flight_data['Flight Duration'] < lower_bound) |␣
 ↪(flight_data['Flight Duration'] > upper_bound)]
print("Outliers in Flight Duration:",outliers)
print("*"*100)

flight_duration = flight_data['Flight Duration']
plt.figure(figsize=(8, 6))
plt.boxplot(flight_duration)
plt.title('Boxplot of Flight Duration')
plt.ylabel('Flight Duration')
plt.show()
```

```
Outliers in Flight Duration: Empty DataFrame
Columns: [Departure City, Arrival City, Departure Date, Flight Duration, Delay
Minutes, Customer ID, Name, Booking Class, Frequent Flyer Status, Route, Ticket
Price, Competitor Price, Demand, Origin, Destination, Profitability, Loyalty
Points, Churned]
Index: []
****************************************************************************
********************
```


Boxplot of Flight Duration
Flight Duration
1.0
0.5
0.0
−0.5
−1.0
−1.5
1