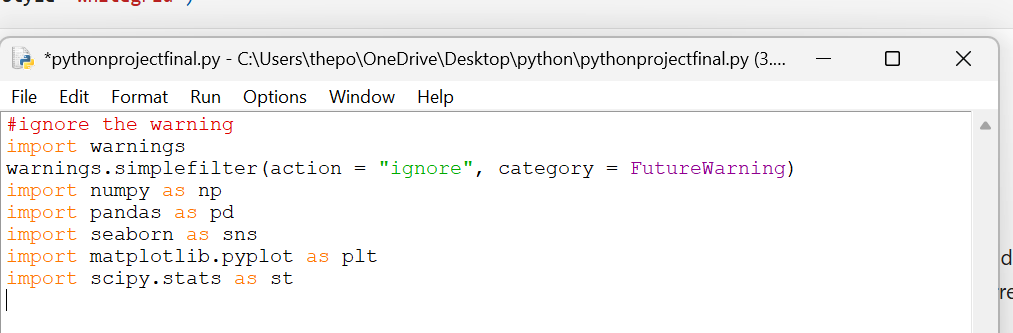
**UK TRAIN RIDES**

**(2020-2025)**

* + - Name – Geetanjali
    - Reg – no – 12323935
    - Roll-no- 65
    - Section – K23ED
    - In This project I have covered almost every point of python libraries including NumPy pandas mat plot and seaborn
    - The Website from which I have taken this dataset is –

https://mavenanalytics.io/data-playground

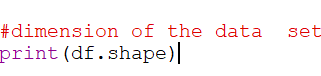
* + - This project is based on the UK Train Rides dataset from Jan to April 2024.
* 1. Importing the warnings and python libraries in idle python --  
   
* 2. Importing the data set



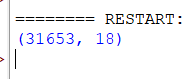
# Using Pandas

* 3. Overview of the data set

. check the dimension of the data set for that we have use shape attribute



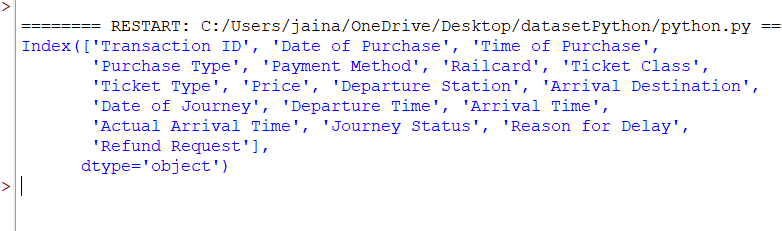
Ans – The output of the code is the



 . check the columns of the dataset for that I used attribute



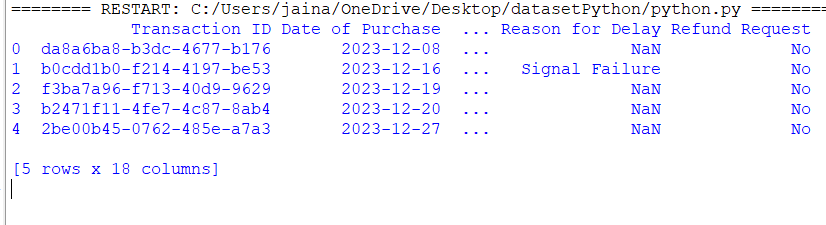
Ans – The output of the code is the



. check the top 5 row of the dataset



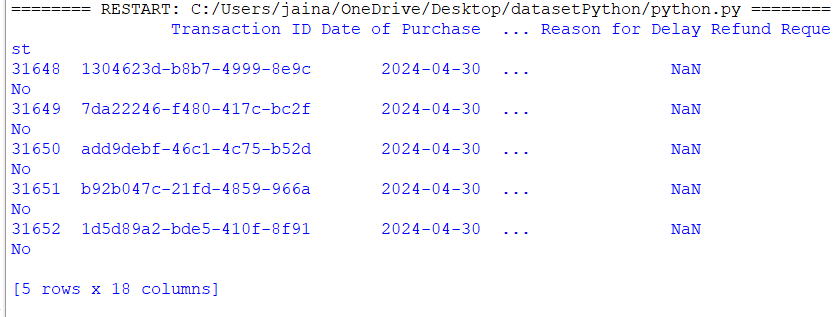
Ans – The output of the code is



. check the list 5 rows of the dataset



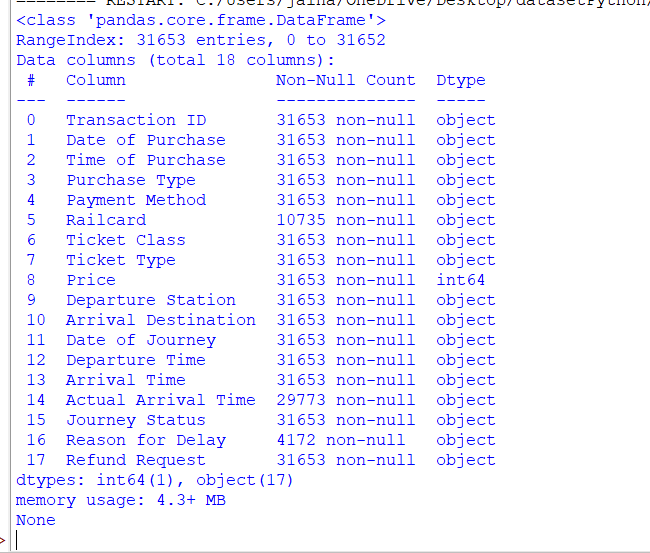
Ans – The output of the code is



. checking all the information of the dataset and details then we use info function

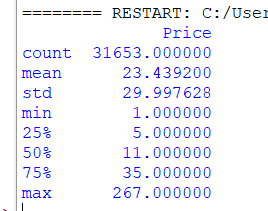


Ans – The output of the code is



. checking for the describe method it will give you the summary of the invention



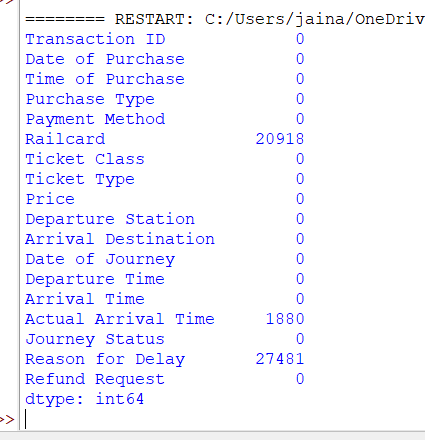
Ans - 

* 4. **Check for anomalies in the dataset**

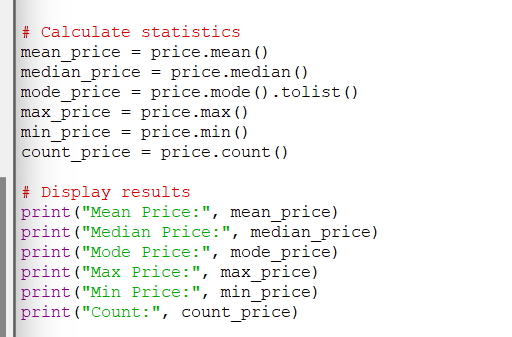
. check for missing numeric values Check for the missing number in the dataset and their sum



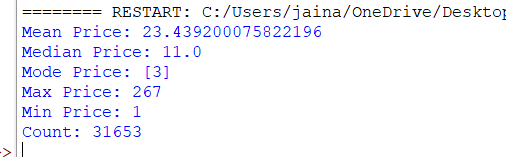
Ans – The output is



* 5. Checking for the max value min values median mode count and sum in one pic



Ans - The output of the code is



* 6. Checking for the cleaning of the dataset A close up of a word

  AI-generated content may be incorrect.

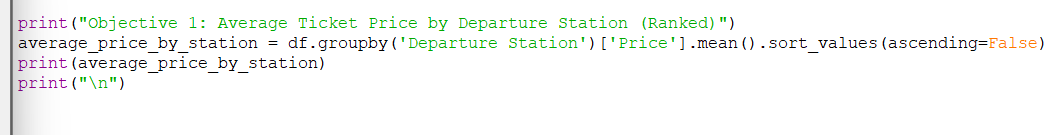
Ans – The output of the code is the



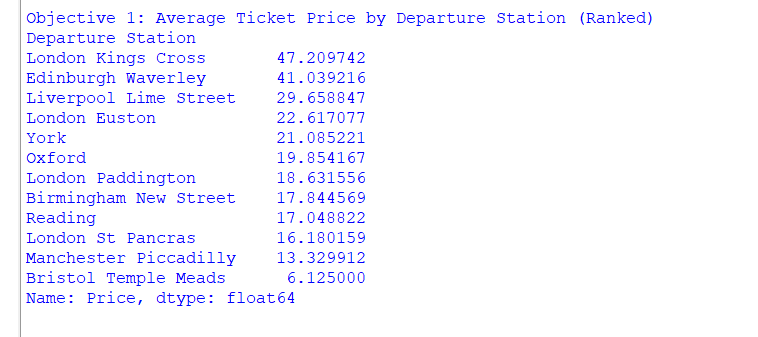
# CREATION OF NUMPY ARRAY

OBJECTIVE 1:

Calculate the Average ticket price by Department , Station and rank them.

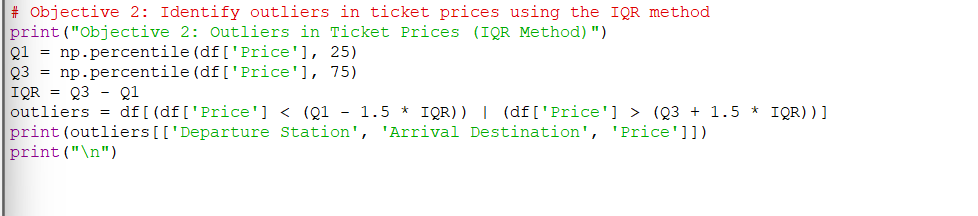


Ans – The output of the code is

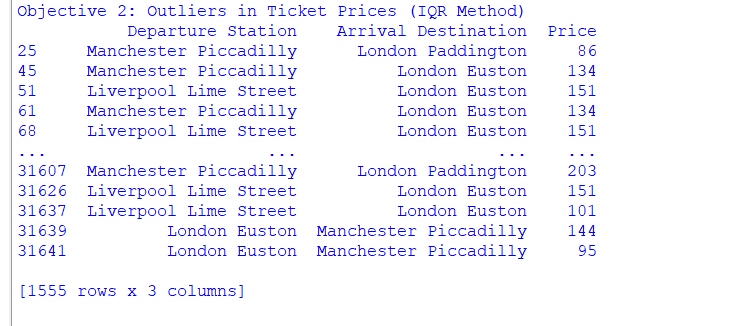


OBJECTIVE 2:

Identify outliers in ticket price using the IQR method.

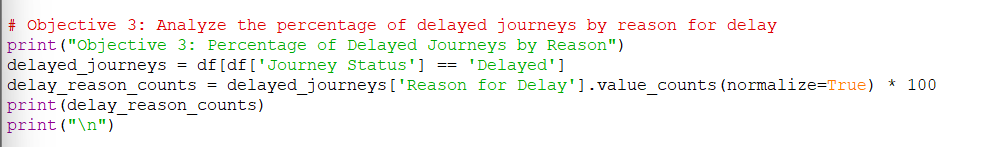


Ans – The output of the code is

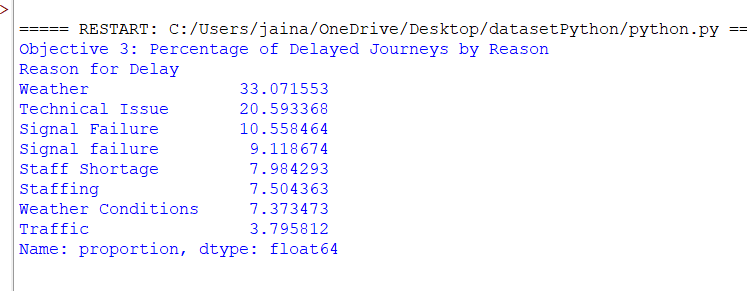


OBJECTIVE 3:

Analyze the percentage of delay journeys by reason for delay.

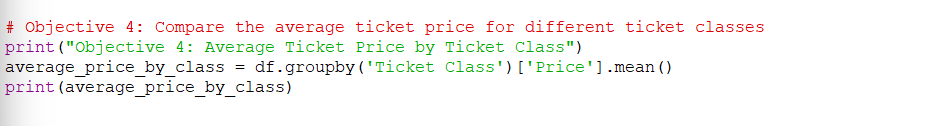


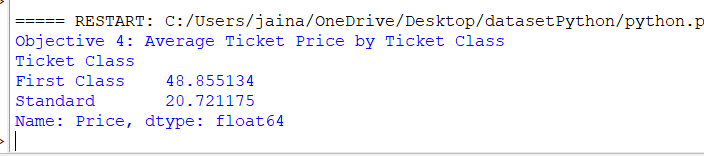
Ans – The output of the code is



OBJECTIVE 4:

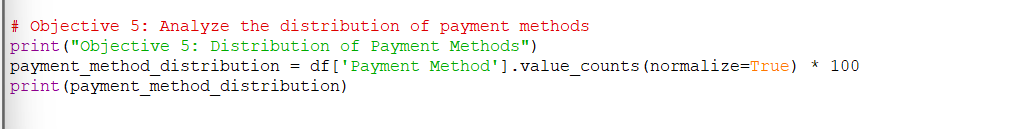
Compare the average ticket price for different ticket classes :



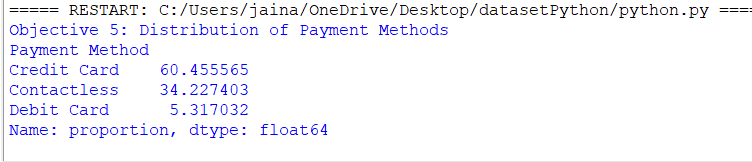
Ans – The output of the code is 

OBJECTIVE 5:

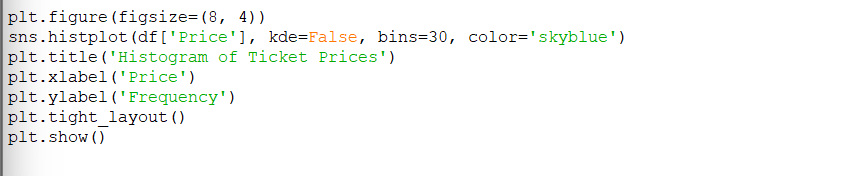
Analyze the distribution of payment method :

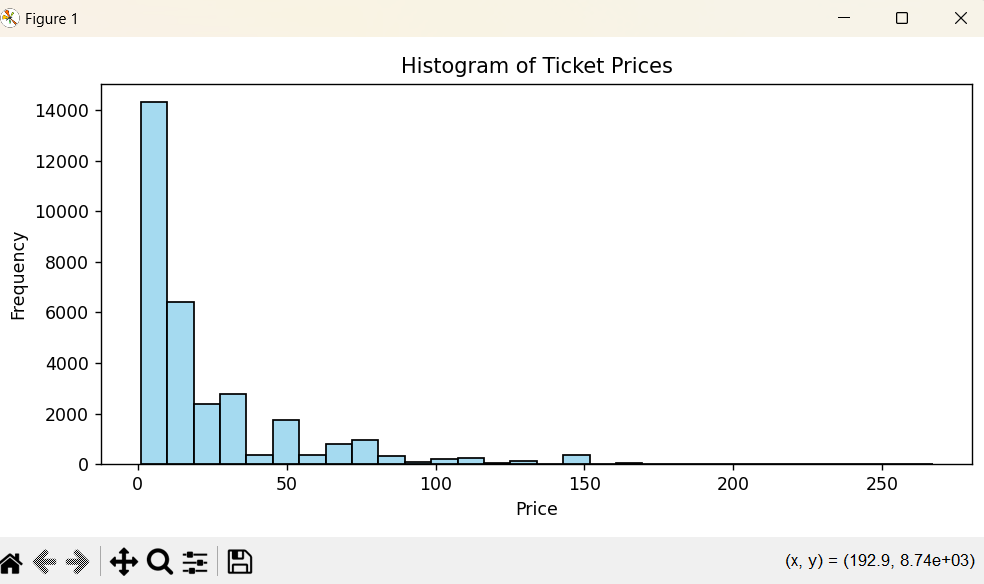


Ans-



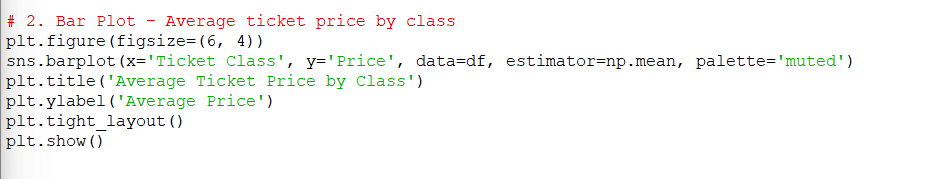
1. HISTOGRAM:



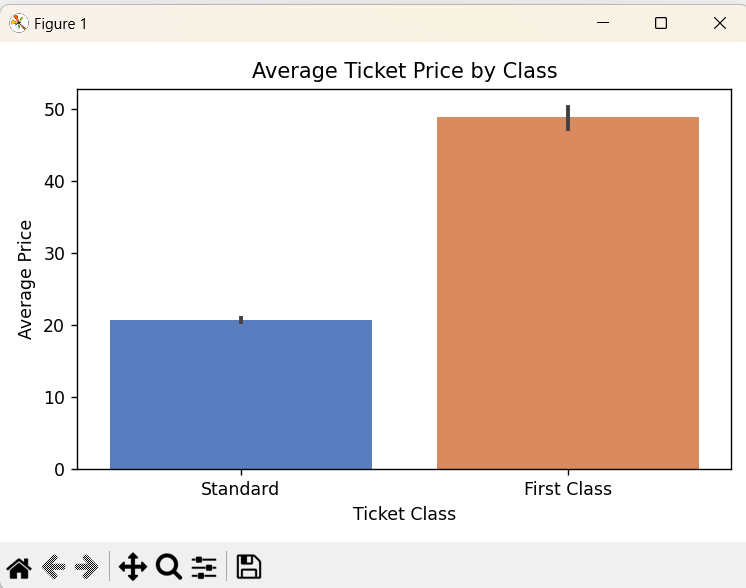


1. BAR PLOT:

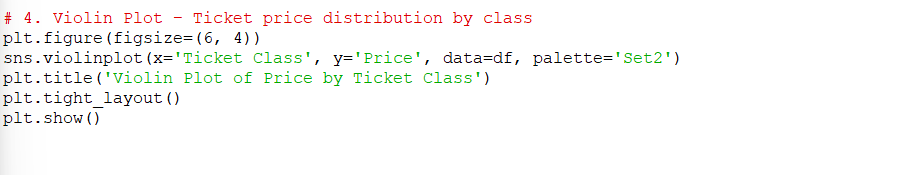
Creating a Bar plot for “CRM Cd” grouped by “AREA NAME”:



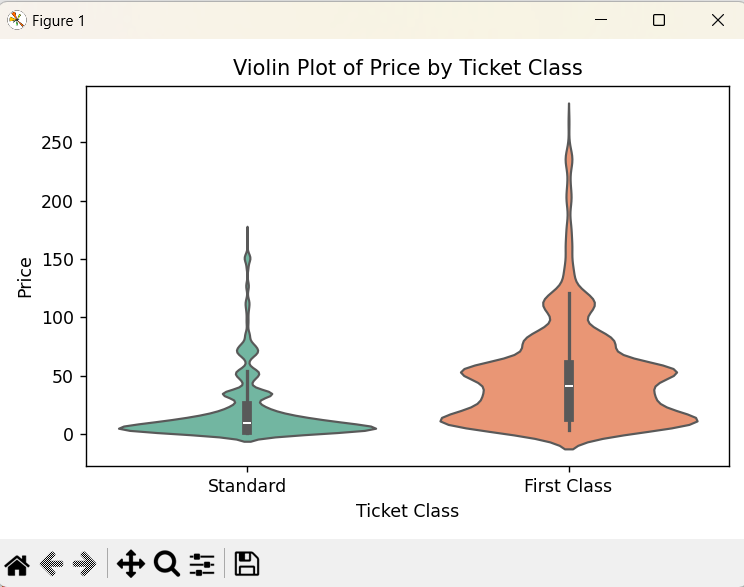
Ans – The output of the code is



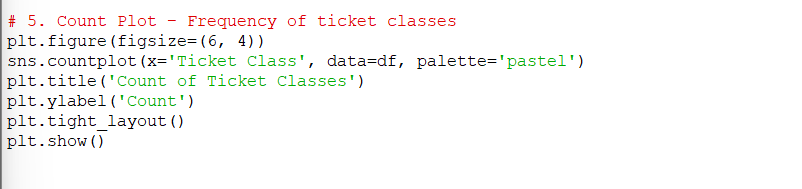
1. violin plot :



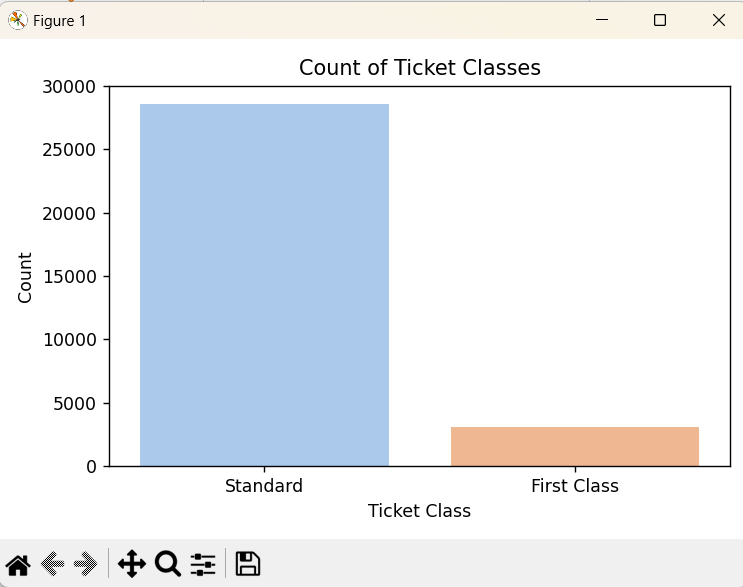
Ans – The output of the code is



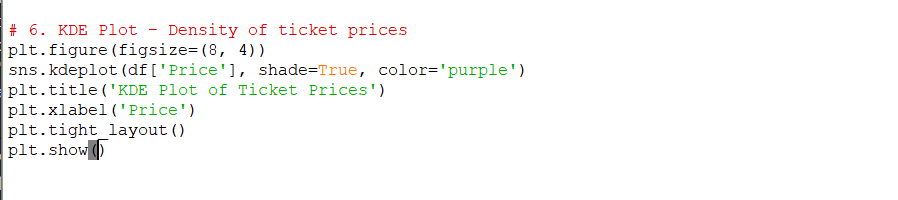
1. count plot:



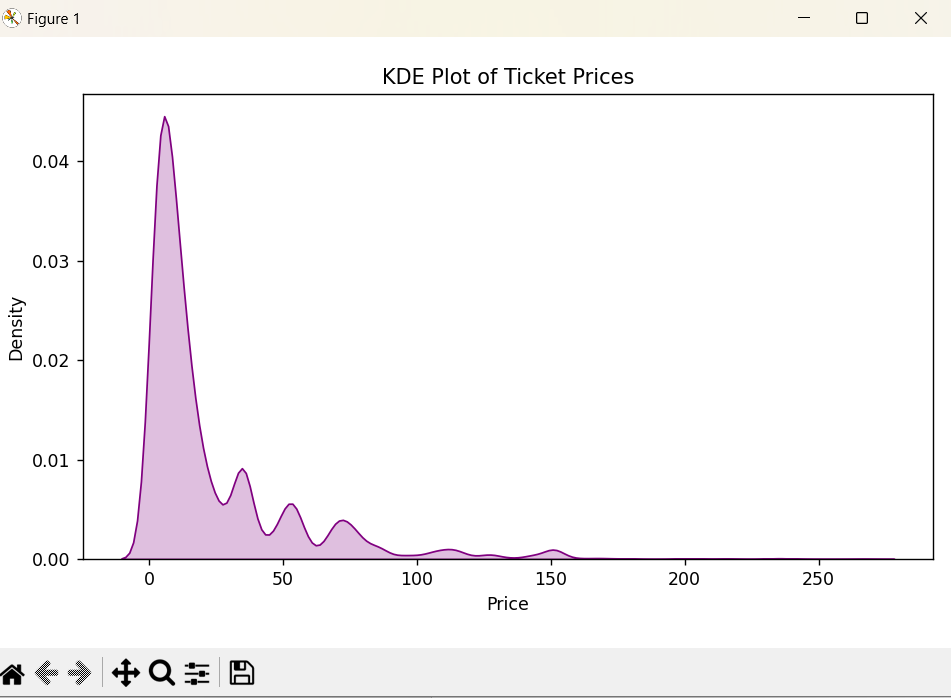
Ans – The output of the code is the



1. KDE plot:



Ans:



BELOW I HAVE WRITTEN ALL THE CODE WHICH I HAVE WRITE IN IDLE PYTHON

import warnings

warnings.simplefilter(action="ignore", category=FutureWarning)

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import scipy.stats as st

# Importing the dataset

df = pd.read\_csv("railway.csv")

# Checking the dimensions of the dataset

#print(df.shape)

# Listing the columns of the dataset

#print(df.columns)

# Viewing the first five rows of the dataset

#print(df.head())

# Viewing the last five rows of the dataset

#print(df.tail())

# Viewing all the information about the dataset

# print(df.info())

# Viewing all the information about the dataset

#print(df.info())

# Descriptive statistics of the dataset

#print(df.describe())

# Checking for missing values in the dataset and their total count

#print(df.isnull().sum())

# Viewing the maximum values in the dataset

#print(df.max())

# Viewing the minimum values in the dataset

#print(df.min())

# Viewing the median values in the dataset

#print(df.median())

# Viewing the mean values in the dataset

#print(df.mean())

# Viewing the mode values in the dataset

#print(df.mode())

# Counting non-null values in each column

#print(df.count())

'''

# Select the numerical column

price = df["Price"]

# Calculate statistics

mean\_price = price.mean()

median\_price = price.median()

mode\_price = price.mode().tolist()

max\_price = price.max()

min\_price = price.min()

count\_price = price.count()

# Display results

print("Mean Price:", mean\_price)

print("Median Price:", median\_price)

print("Mode Price:", mode\_price)

print("Max Price:", max\_price)

print("Min Price:", min\_price)

print("Count:", count\_price)

'''

# Cleaning the dataset by dropping rows with missing values

#print(df.dropna(inplace=True))

'''

print("Objective 1: Average Ticket Price by Departure Station (Ranked)")

average\_price\_by\_station = df.groupby('Departure Station')['Price'].mean().sort\_values(ascending=False)

print(average\_price\_by\_station)

print("\n")

'''

'''

# Objective 2: Identify outliers in ticket prices using the IQR method

print("Objective 2: Outliers in Ticket Prices (IQR Method)")

Q1 = np.percentile(df['Price'], 25)

Q3 = np.percentile(df['Price'], 75)

IQR = Q3 - Q1

outliers = df[(df['Price'] < (Q1 - 1.5 \* IQR)) | (df['Price'] > (Q3 + 1.5 \* IQR))]

print(outliers[['Departure Station', 'Arrival Destination', 'Price']])

print("\n")

'''

'''

# Objective 3: Analyze the percentage of delayed journeys by reason for delay

print("Objective 3: Percentage of Delayed Journeys by Reason")

delayed\_journeys = df[df['Journey Status'] == 'Delayed']

delay\_reason\_counts = delayed\_journeys['Reason for Delay'].value\_counts(normalize=True) \* 100

print(delay\_reason\_counts)

print("\n")

# Objective 4: Compare the average ticket price for different ticket classes

print("Objective 4: Average Ticket Price by Ticket Class")

average\_price\_by\_class = df.groupby('Ticket Class')['Price'].mean()

print(average\_price\_by\_class)

#Histogram:

average\_price\_by\_class.plot(kind='bar', title='Average Ticket Price by Class', color='skyblue')

plt.xlabel('Ticket Class')

plt.ylabel('Average Price')

plt.tight\_layout()

plt.show()

plt.figure(figsize=(8, 4))

sns.histplot(df['Price'], kde=False, bins=30, color='skyblue')

plt.title('Histogram of Ticket Prices')

plt.xlabel('Price')

plt.ylabel('Frequency')

plt.tight\_layout()

plt.show()

# Objective 5: Analyze the distribution of payment methods

print("Objective 5: Distribution of Payment Methods")

payment\_method\_distribution = df['Payment Method'].value\_counts(normalize=True) \* 100

print(payment\_method\_distribution)

# 2. Bar Plot – Average ticket price by class

plt.figure(figsize=(6, 4))

sns.barplot(x='Ticket Class', y='Price', data=df, estimator=np.mean, palette='muted')

plt.title('Average Ticket Price by Class')

plt.ylabel('Average Price')

plt.tight\_layout()

plt.show()

# 4. Violin Plot – Ticket price distribution by class

plt.figure(figsize=(6, 4))

sns.violinplot(x='Ticket Class', y='Price', data=df, palette='Set2')

plt.title('Violin Plot of Price by Ticket Class')

plt.tight\_layout()

plt.show()

# 5. Count Plot – Frequency of ticket classes

plt.figure(figsize=(6, 4))

sns.countplot(x='Ticket Class', data=df, palette='pastel')

plt.title('Count of Ticket Classes')

plt.ylabel('Count')

plt.tight\_layout()

plt.show()

'''

# 6. KDE Plot – Density of ticket prices

plt.figure(figsize=(8, 4))

sns.kdeplot(df['Price'], shade=True, color='purple')

plt.title('KDE Plot of Ticket Prices')

plt.xlabel('Price')

plt.tight\_layout()

plt.show()