**21l 5367**

**Ibtisam ali**

**6 A**

**Q1:**

from google.colab import files

# Upload the CSV file

uploaded = files.upload()

for filename in uploaded.keys():

    print(f'Uploaded file: {filename}')

import pandas as pd

filename = list(uploaded.keys())[0]

df = pd.read\_csv(filename)

print(df.head())

print("First few rows of the dataset:")

print(df.head())

# Check the data types of each column

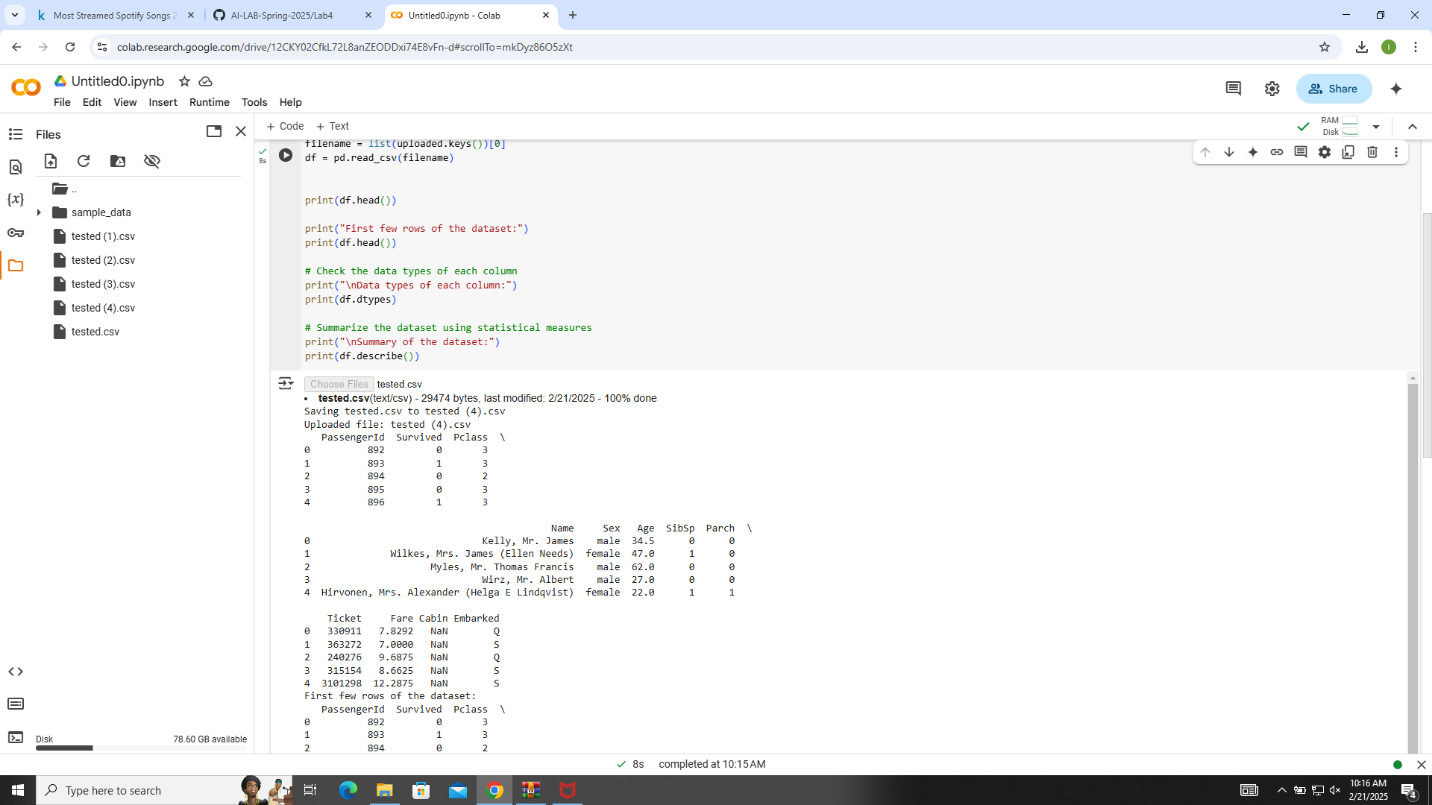
print("\nData types of each column:")

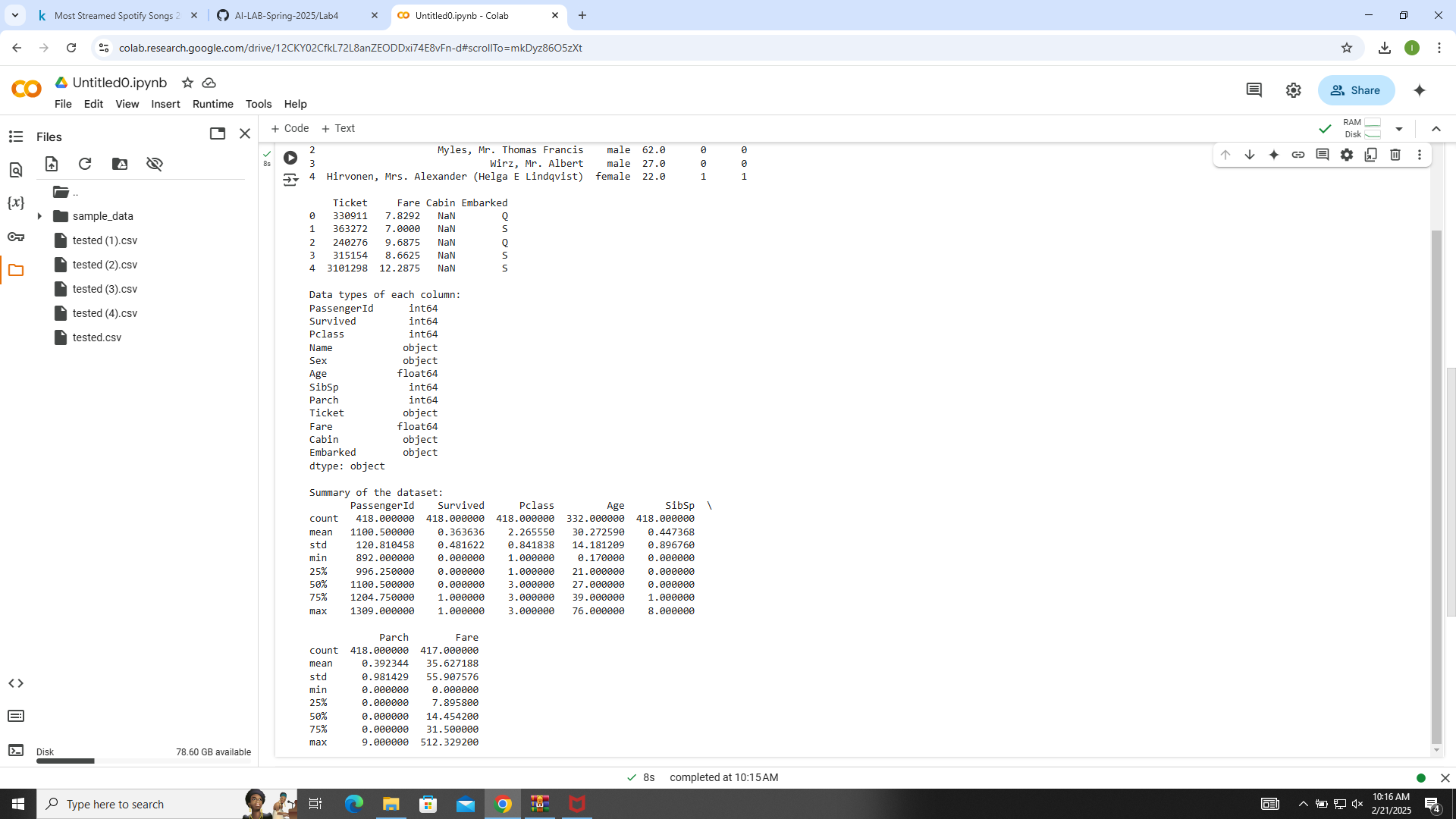
print(df.dtypes)

# Summarize the dataset using statistical measures

print("\nSummary of the dataset:")

print(df.describe())





**Q2:**

# Identify missing values

print("Identifying missing values:")

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

# Step 2: Clean the Dataset

    if df[column].dtype == 'object':

        df[column].fillna(df[column].mode()[0], inplace=True)

    else:

        df[column].fillna(df[column].mean(), inplace=True)

print("\nAfter handling missing values:")

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

#

def remove\_outliers(data, column):

    Q1 = data[column].quantile(0.25)

    Q3 = data[column].quantile(0.75)

    IQR = Q3 - Q1

    lower\_bound = Q1 - 1.5 \* IQR

    upper\_bound = Q3 + 1.5 \* IQR

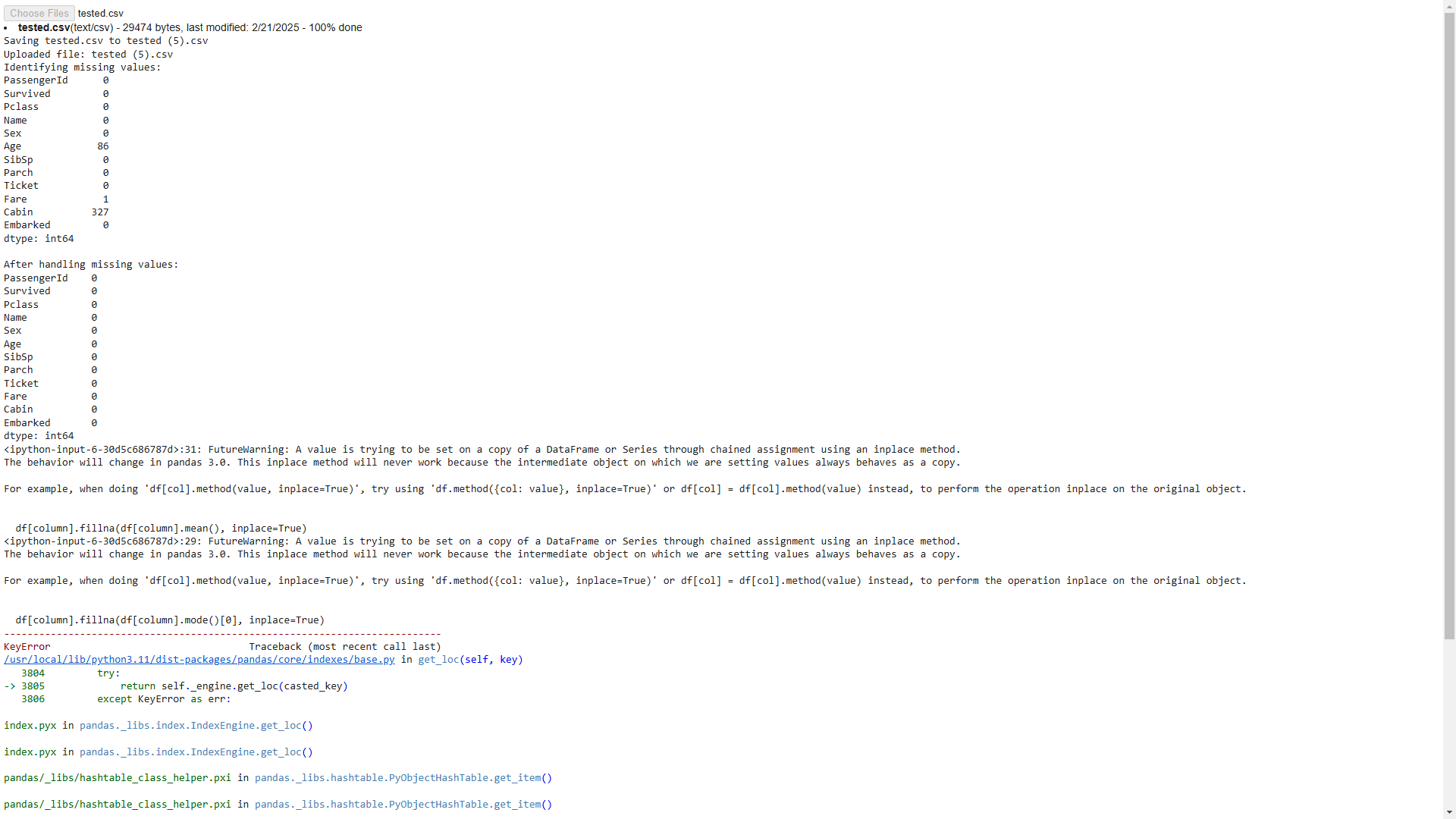
    return data[(data[column] >= lower\_bound) & (data[column] <= upper\_bound)]

df = df[df['Streams'] >= 0]

df = remove\_outliers(df, 'Streams')

print("\nCleaned dataset summary:")

print(df.describe())



**Q3,4:**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Univariate Analysis

numerical\_columns = df.select\_dtypes(include=['float64', 'int64']).columns

for column in numerical\_columns:

    plt.figure(figsize=(10, 5))

    sns.histplot(df[column], kde=True)

    plt.title(f'Histogram of {column}')

    plt.show()

for column in numerical\_columns:

    plt.figure(figsize=(10, 5))

    sns.boxplot(x=df[column])

    plt.title(f'Box Plot of {column}')

    plt.show()

# Bivariate Analysis

for i in range(len(numerical\_columns)):

    for j in range(i + 1, len(numerical\_columns)):

        plt.figure(figsize=(10, 5))

        sns.scatterplot(x=df[numerical\_columns[i]], y=df[numerical\_columns[j]])

        plt.title(f'Scatter Plot between {numerical\_columns[i]} and {numerical\_columns[j]}')

        plt.show()

plt.figure(figsize=(12, 8))

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Heatmap of Correlation Matrix')

plt.show()

