

ASSIGNMENT

**STUDENT NAME** **:** **UZAIR AHMED**

**STUDENT IDENTITY :** **CSC-21F-130**

**COURSE :** **ARTIFICIAL INTELLIGENCE**

**PROGRAM : BSCS (MORNING)**

**SECTION :** **6TH ‘C’**

**COURSE COORDINATOR** **:** **MISS AQSA UMAR**.

PREDICT SURVIVAL OF PATIENTS WITH HEART FAILURE

ANALYSIS:

Importing the libraries in notebook:

import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

import seaborn as sns

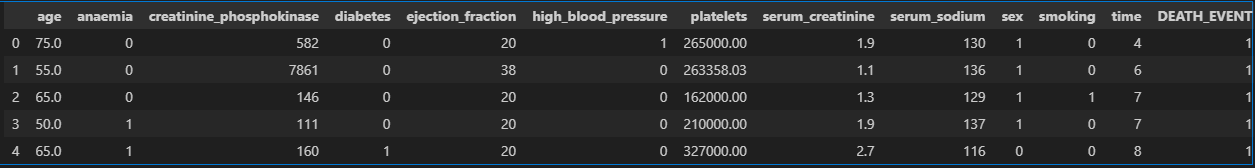
this code reads a CSV file named 'heart\_failure\_clinical\_records\_dataset.csv' into a pandas DataFrame and assigns it to the variable df.

df = pd.read\_csv('heart\_failure\_clinical\_records\_dataset.csv')

df.head() displays the first few rows of the DataFrame df. By default, it shows the first 5 rows, but you can specify a different number of rows to view within the parenthesis, like df.head(10) for the first 10 rows.

df.head()

output:



df.shape returns a tuple containing the dimensions of the DataFrame df. This tuple represents the number of rows and columns in the data, respectively.

df.shape

output:

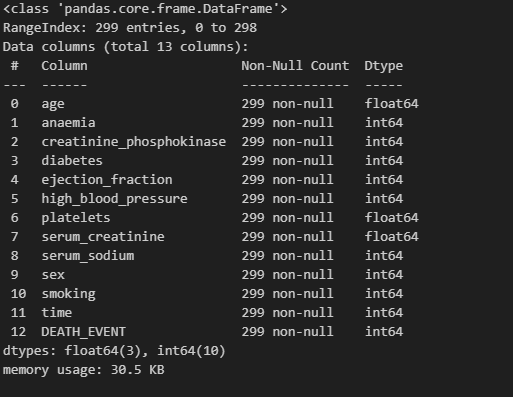
C:\Users\3TEE\Downloads\out1.PNG

df.info() provides a concise summary of the DataFrame df's structure and information.

pen\_spark

df.info()

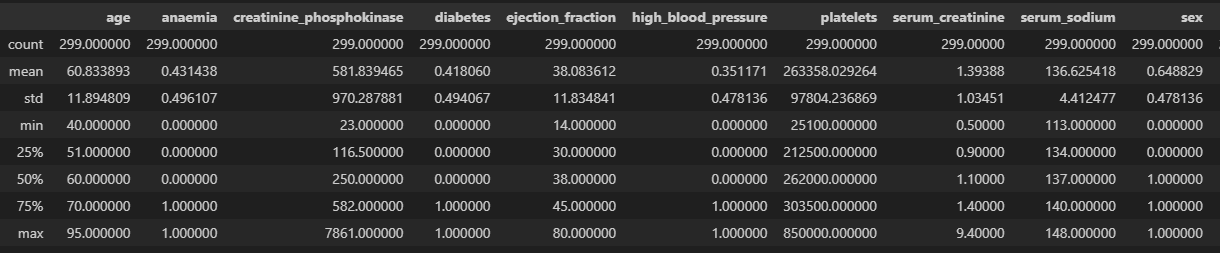
output:



df.describe() summary of descriptive statistics for the DataFrame df

df.describe()

output:





directly converting the entire code block for data cleaning with boxplots into a single line isn't possible in Python due to the limitations of for loops and plotting functionalities.

However, you can achieve a similar outcome using a list comprehension and the pandas.DataFrame.plot

# data cleaning

for col in df.columns:

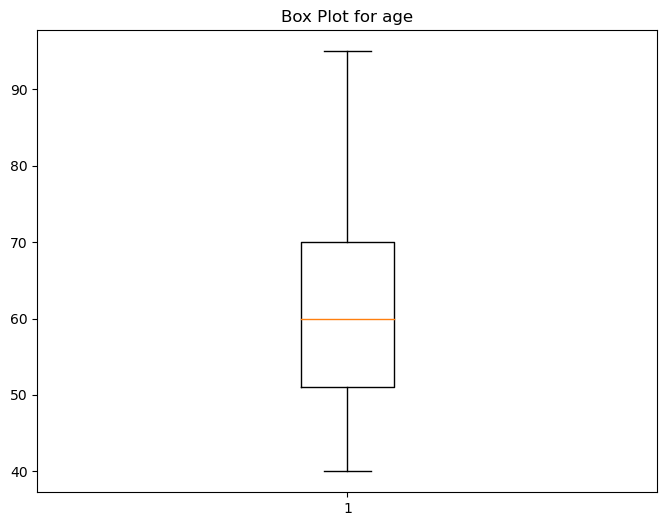
    plt.figure(figsize=(8, 6))

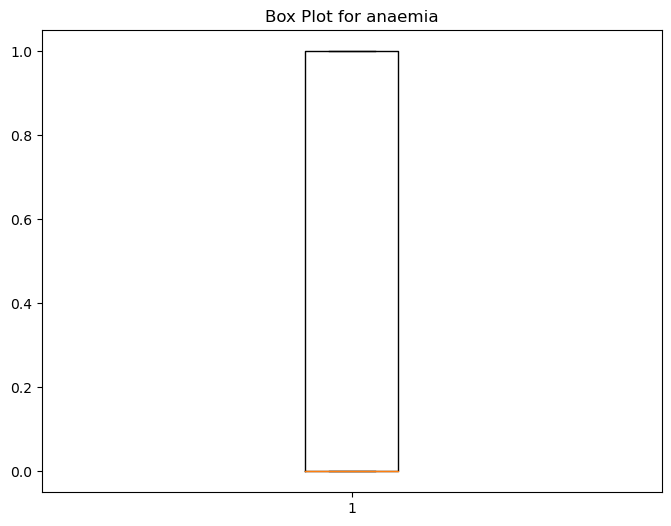
    plt.boxplot(df[col])

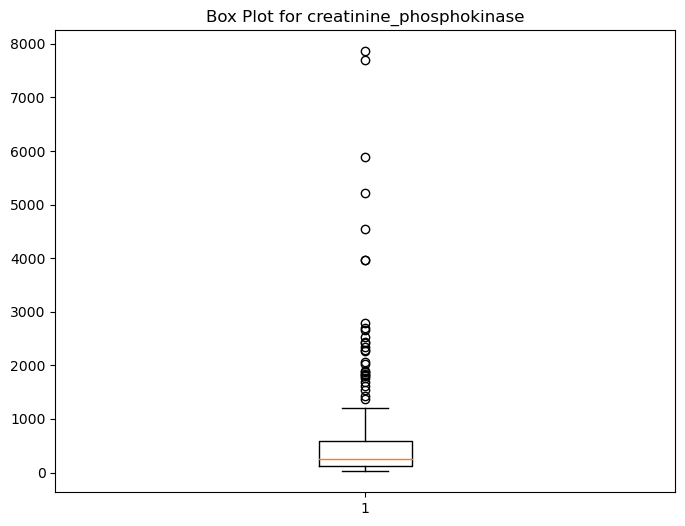
    plt.title(f'Box Plot for {col}')

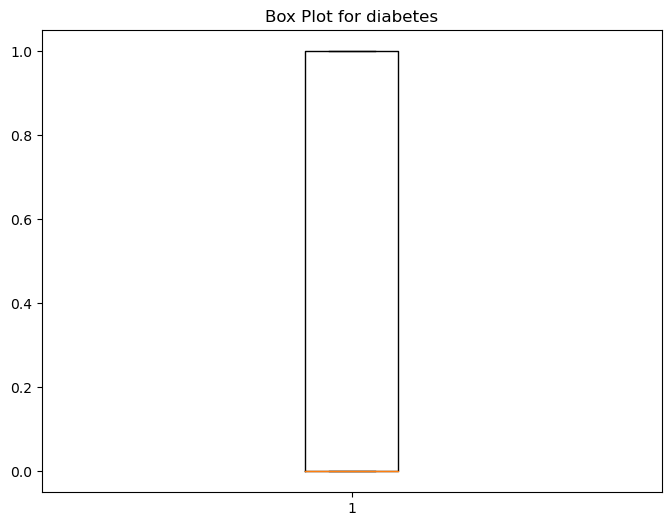
    plt.show()

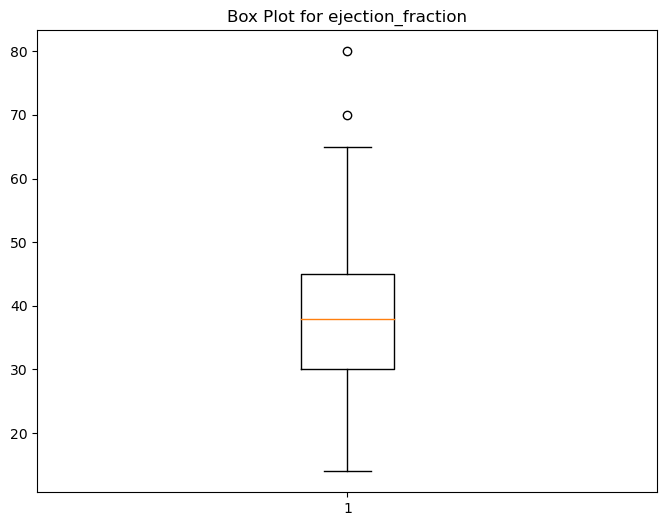
output:

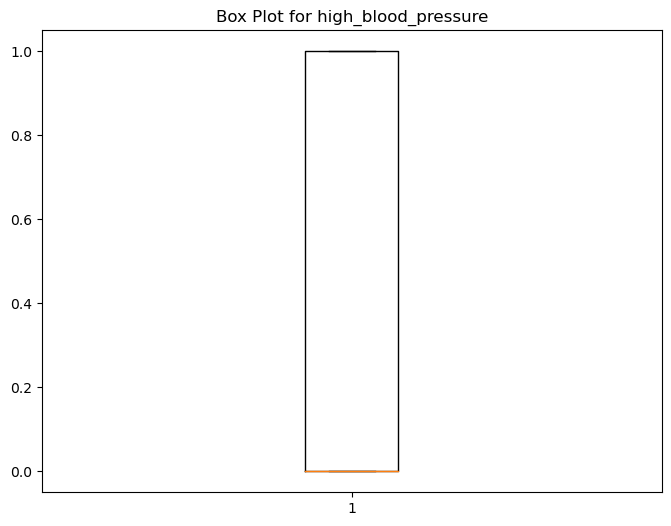


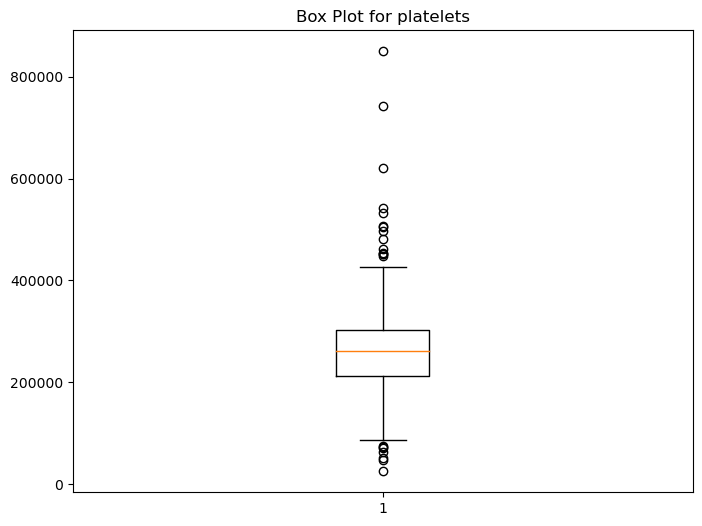


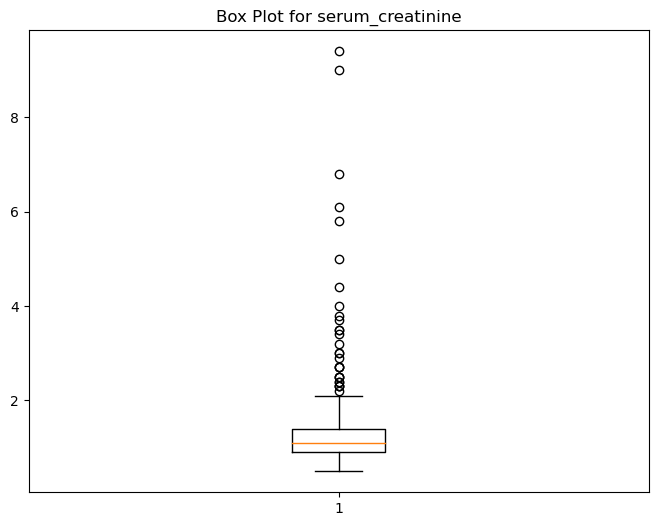


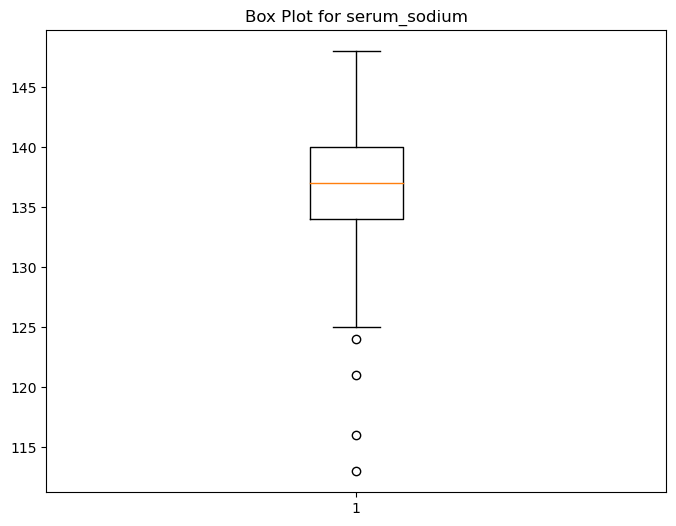


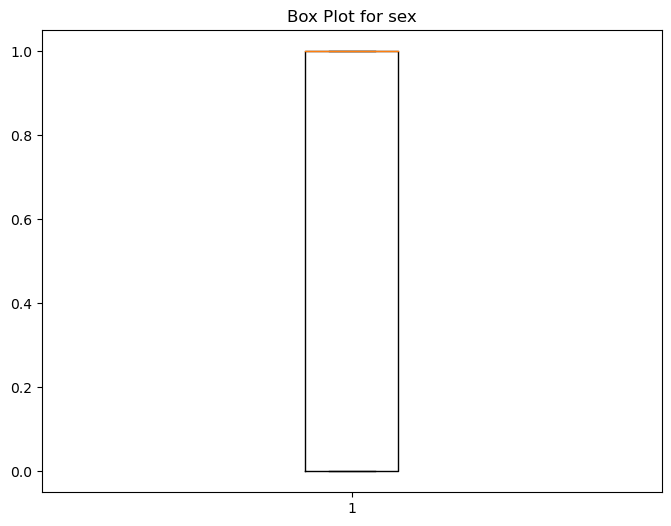


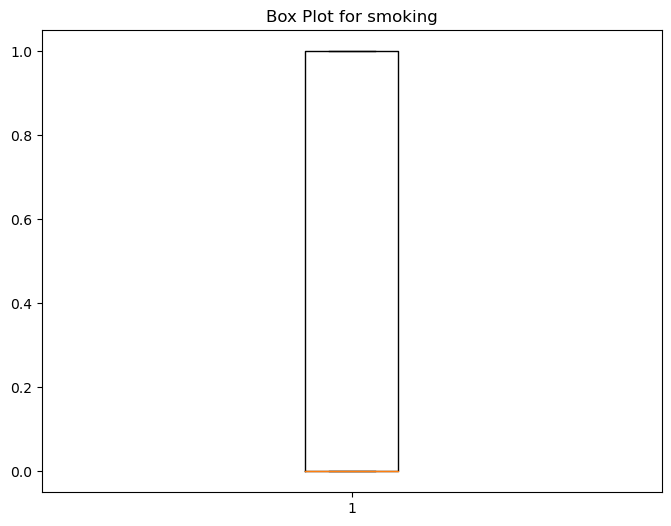


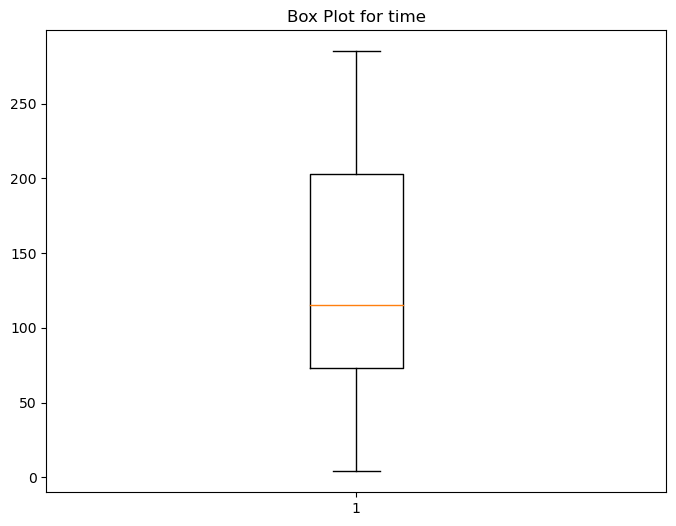


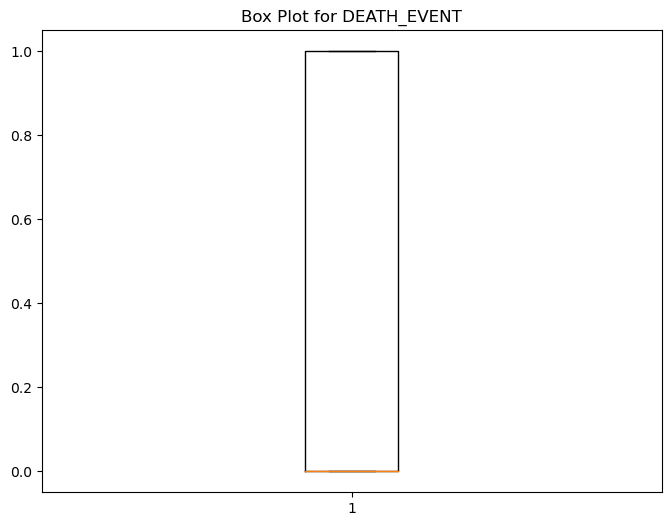












While directly converting the entire function for outlier removal into a single line in Python isn't ideal due to readability concerns

def remove\_outliers(df):

    for col in df.select\_dtypes(include='number').columns:

        q1 = df[col].quantile(0.25)

        q3 = df[col].quantile(0.75)

        iqr = q3 - q1

        lower\_bound = q1 - 1.5 \* iqr

        upper\_bound = q3 + 1.5 \* iqr

        outliers=(df[col] < lower\_bound ) | (df[col] > upper\_bound)

        df=df[~outliers]

    return df

# Remove outliers

df = remove\_outliers(df)

Directly converting the entire code block for boxplots into a single line in Python isn't ideal due to readability limitations.

for col in df.columns:

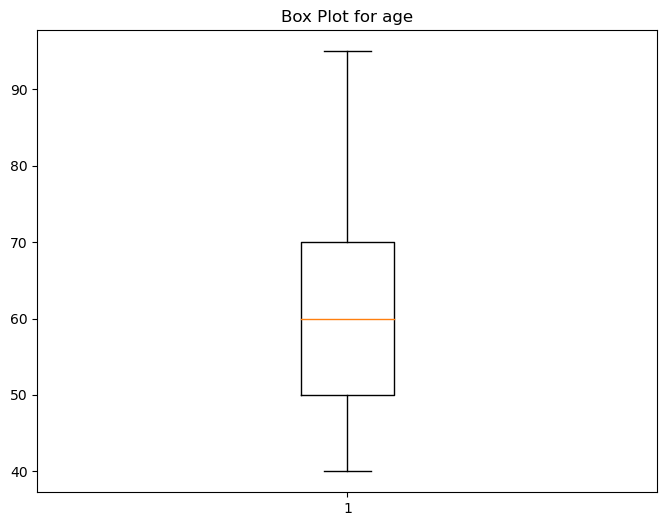
    plt.figure(figsize=(8, 6))

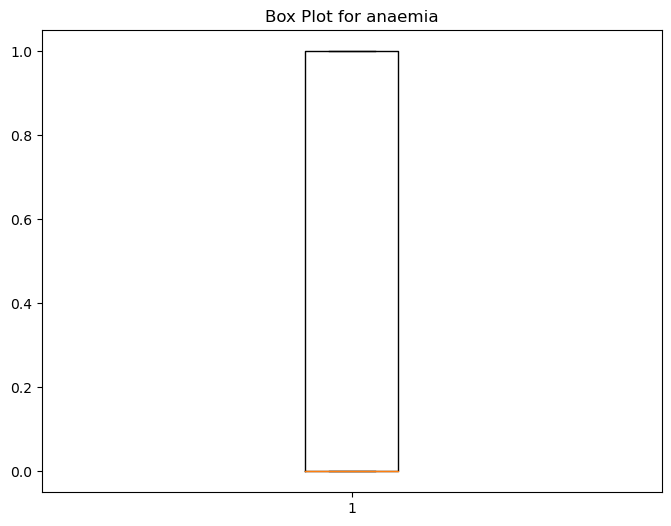
    plt.boxplot(df[col])

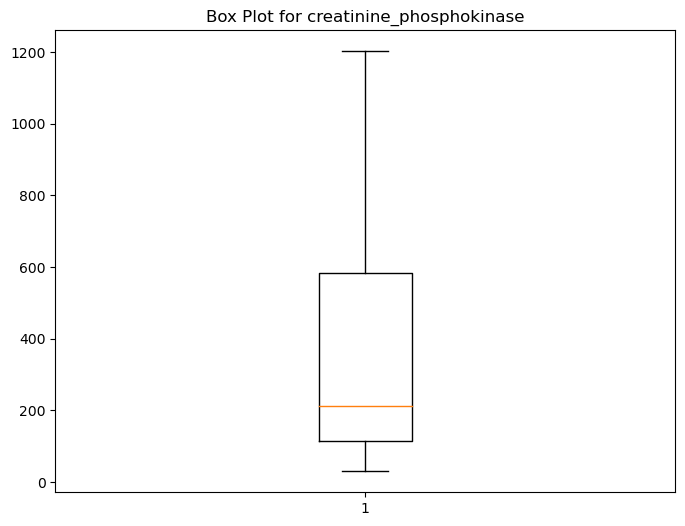
    plt.title(f'Box Plot for {col}')

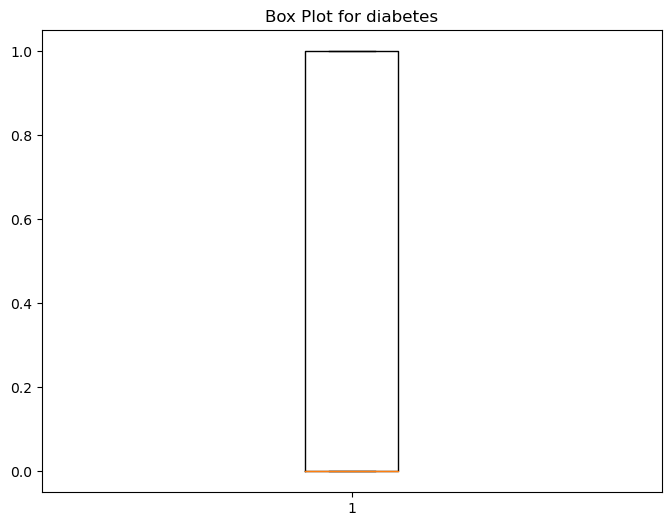
    plt.show()

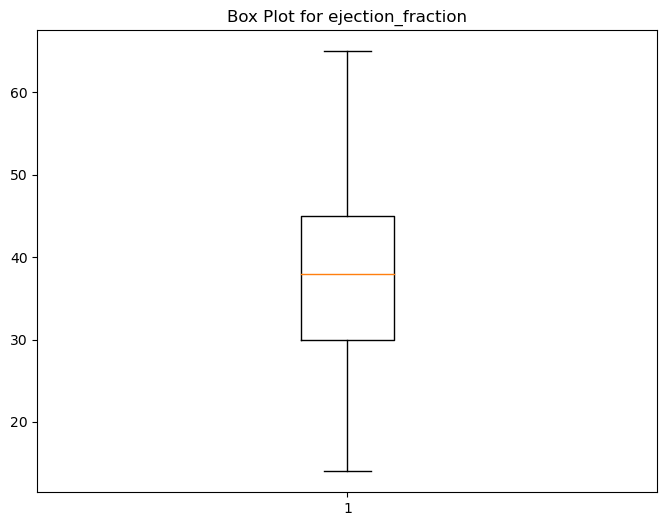
output:

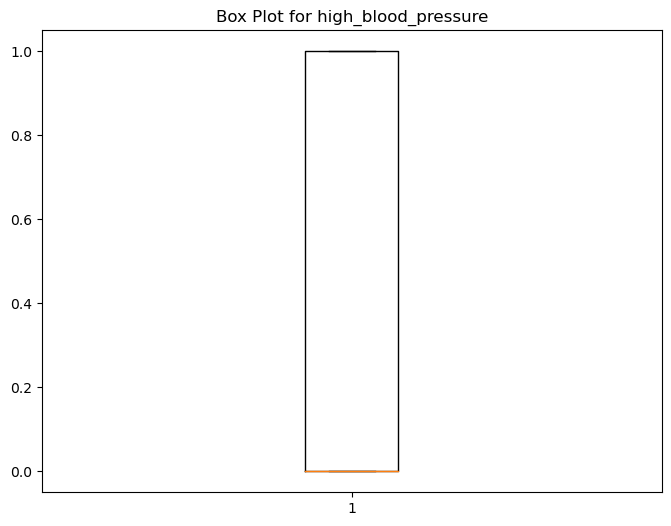


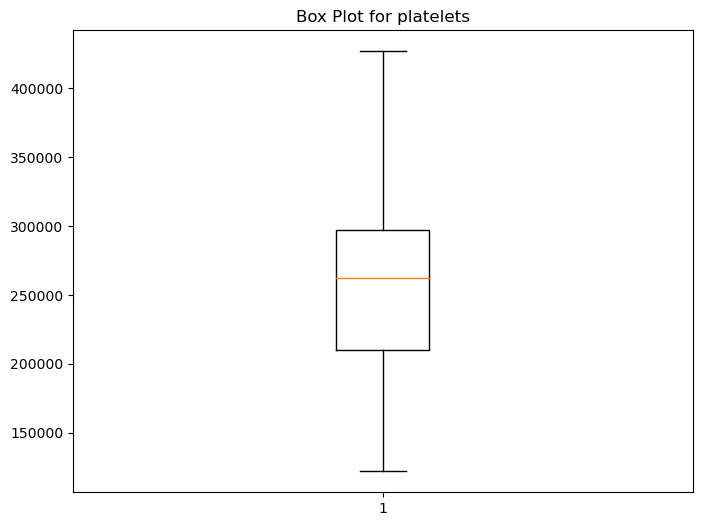


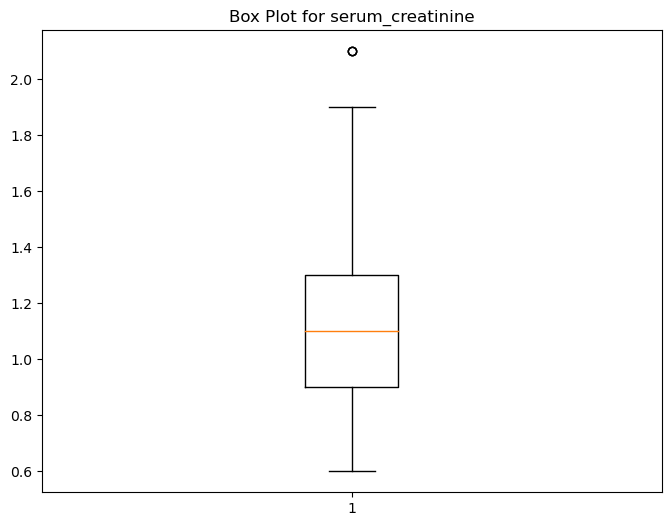


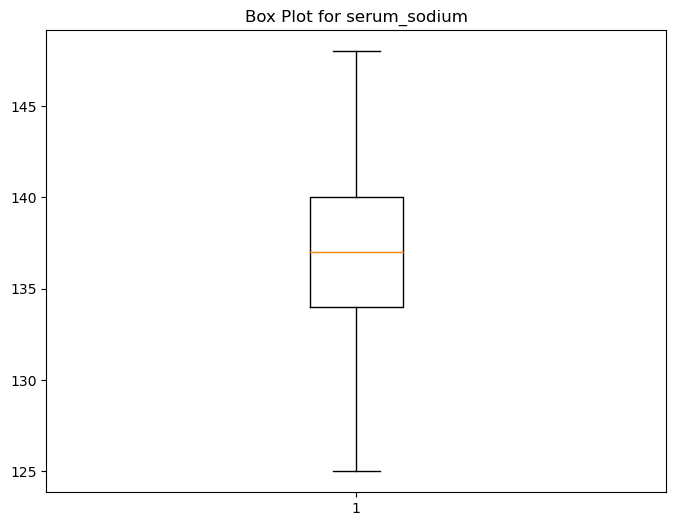


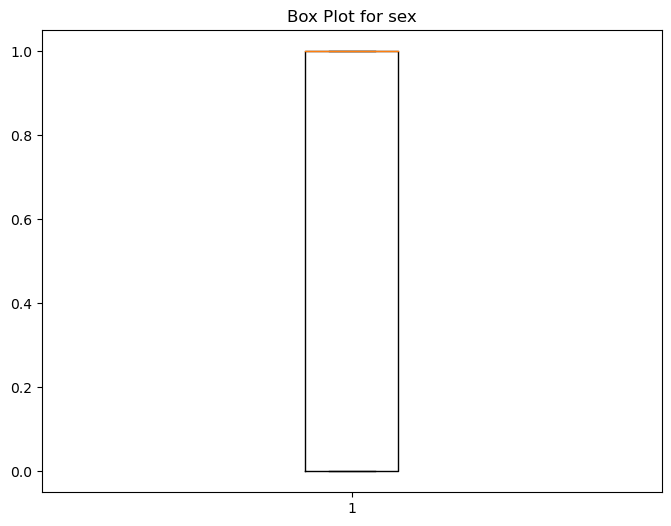


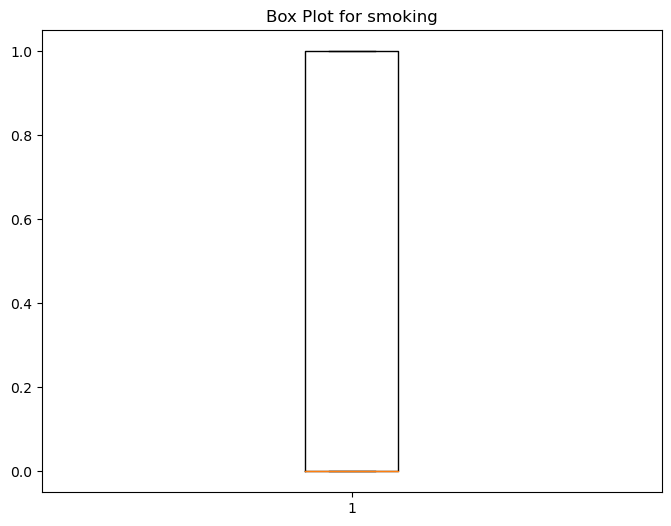


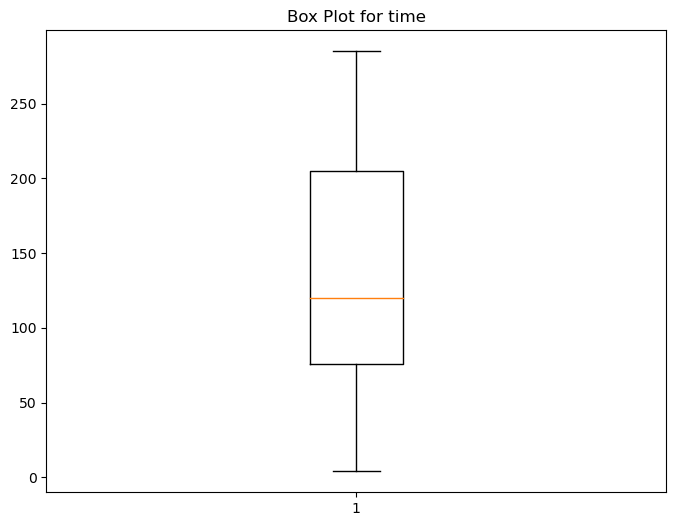


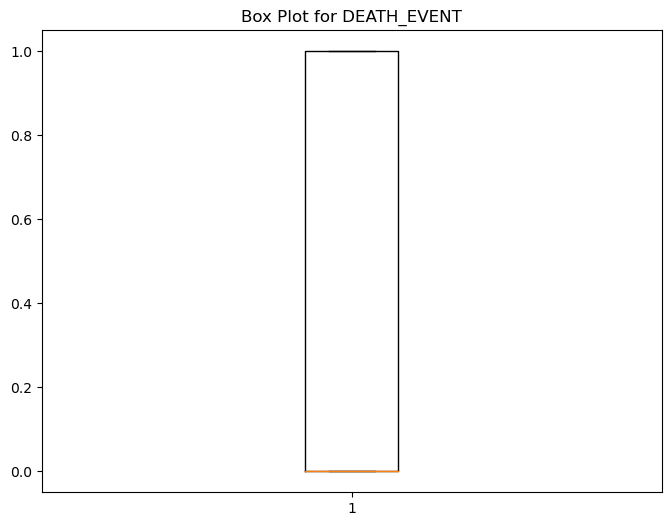












print the shape of a DataFrame (number of rows, columns) in a single line using the .shape attribute

df.shape

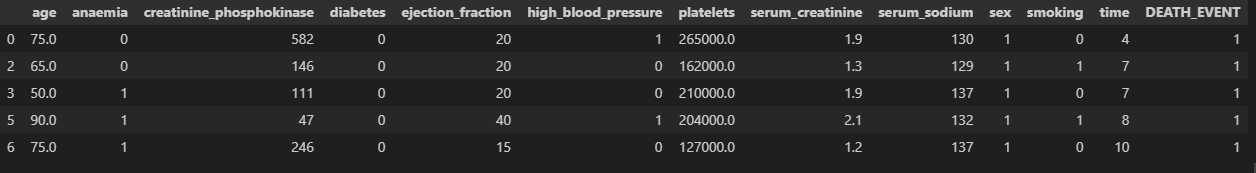
output:

C:\Users\3TEE\Downloads\out.PNG

first few rows of a DataFrame df using df.head() in a single line

df.head()

output:



one-line version of the code you provided to create a histogram of the 'age' column in DataFrame df using

seaborn:

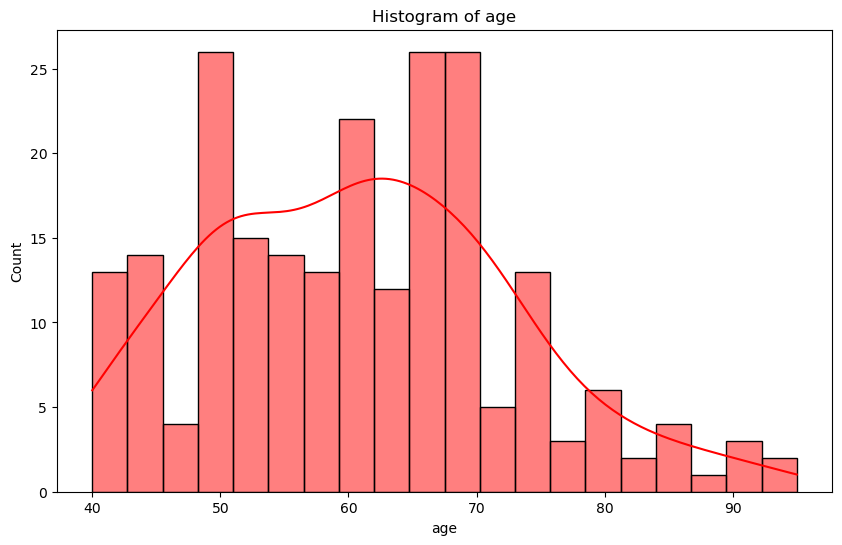
plt.figure(figsize=(10,6))

sns.histplot(df['age'], kde = True , color='red', bins = 20)

plt.title('Histogram of age')

plt.show()

output:



Determine if more than half of the people in the Data-Frame df have anemia.

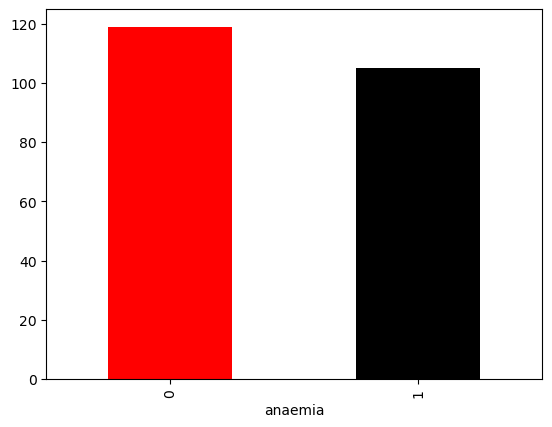
# anaemia

df['anaemia'].value\_counts().plot(kind = 'bar',color=['red','black'])

plt.show()

# more than half of the people have anaemia

Output:



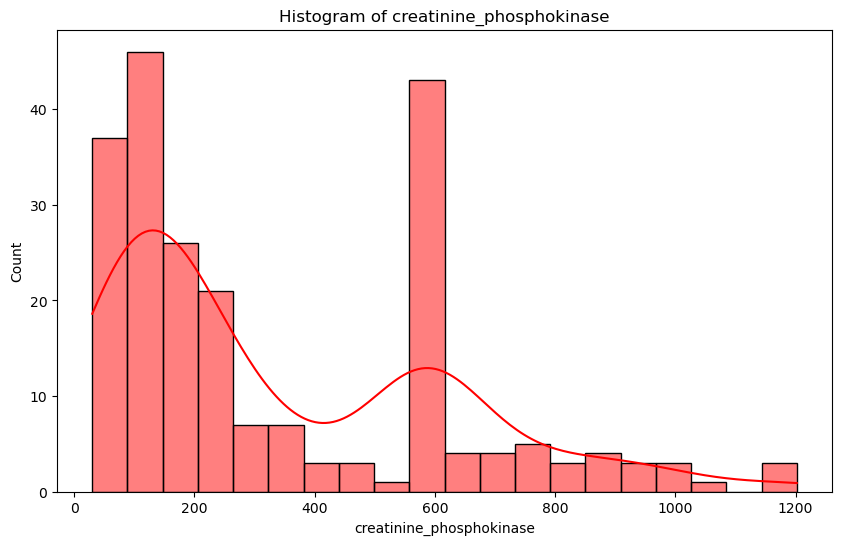
create a histogram of the 'creatinine\_phosphokinase' column in DataFrame df using seaborn:

plt.figure(figsize=(10,6))

sns.histplot(df['creatinine\_phosphokinase'], kde = True , color='red', bins = 20)

plt.title('Histogram of creatinine\_phosphokinase')

plt.show()

output: 

To create a correlation heatmap using seaborn:

# heatmap

plt.figure(figsize=(10, 8))

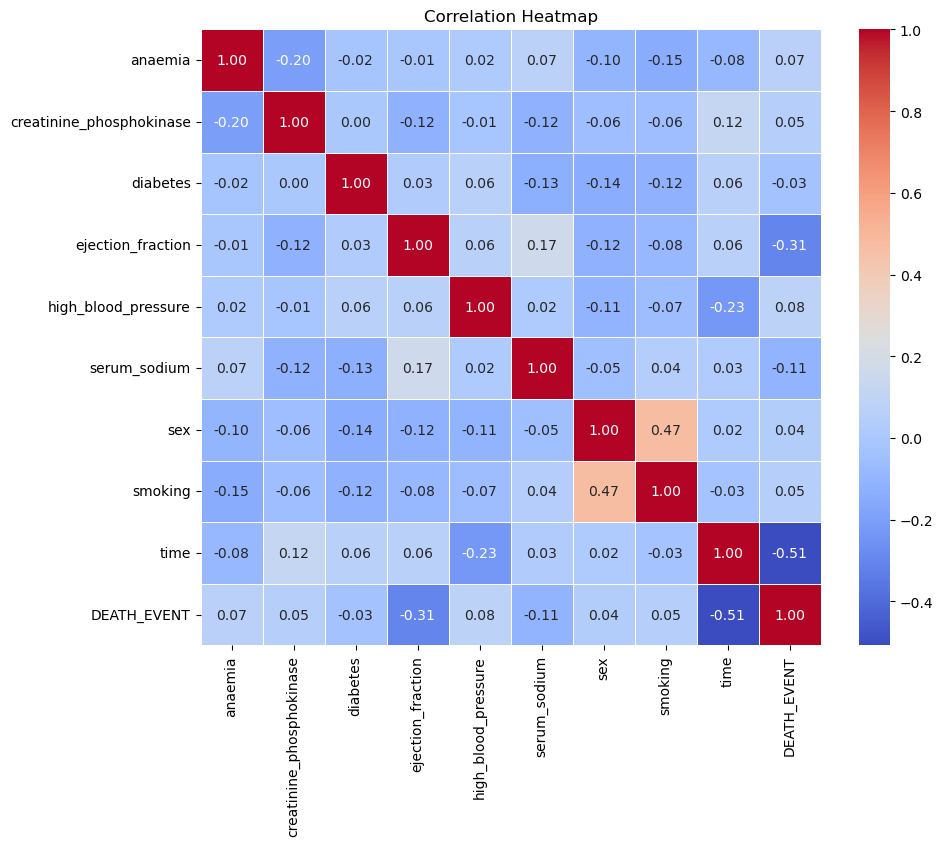
num\_cols = df.select\_dtypes('int','float')

heatmap = sns.heatmap(num\_cols.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)

heatmap.set\_title('Correlation Heatmap')

plt.show()

output:

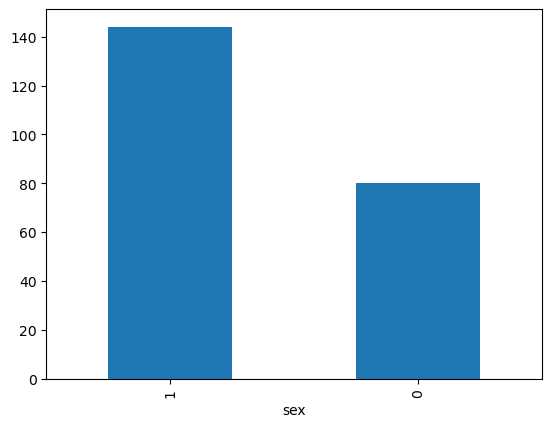


To create a bar chart showing the counts of each sex in DataFrame df:

df['sex'].value\_counts().plot(kind = 'bar')

plt.show()

output:



create a scatterplot of 'age' vs. 'platelets' in DataFrame df using seaborn:

# age vs platelets

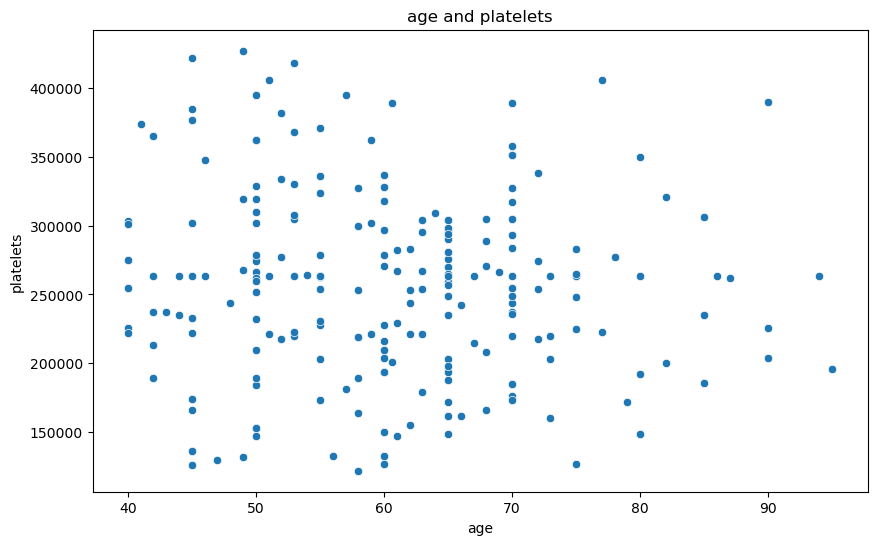
plt.figure(figsize=(10,6))

sns.scatterplot(x = df['age'], y = df['platelets'])

plt.title('age and platelets')

plt.show()

output:



A bar chart showing the average platelet count for each sex in DataFrame df,

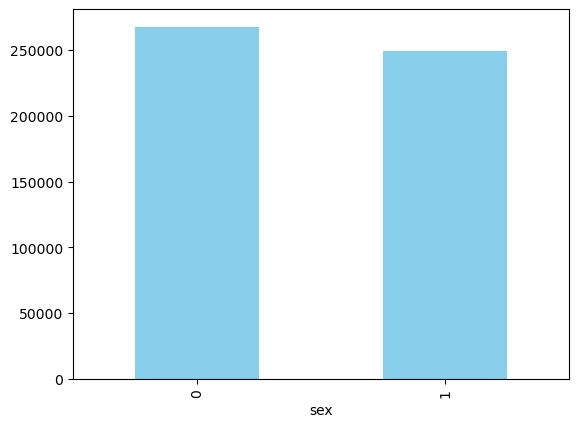
task as the original code block, providing a concise way to visualize the average platelet count distribution across genders.

# avg platelets count vs gender

df.groupby('sex')['platelets'].mean().plot(kind = 'bar' , color = 'skyblue')

plt.show()

output:



References:

Dataset :  
  
BMC Medical Informatics and Decision Making 20, 16 (2020)

Link: <https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-1023-5>

GitHub:

Link : <https://github.com/Uzi130/heart-failure-dataset>