**Lab 1**

**Data understanding and preprocessing**

1. Write a program to find the various statistical measures (mean median, model, etc) based on the sample dataset.

Statistical measures are crucial for understanding the distribution and characteristics of a dataset. This program calculates fundamental statistics such as mean, median, mode, and more, providing insights into the central tendency and spread of the data. These measures serve as the foundation for further data analysis and mining tasks.

Program:

import pandas as pd

import numpy as np

dataset\_url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv'

# Define the column names

columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

# Load the dataset into a DataFrame

pima = pd.read\_csv(dataset\_url, names=columns)

# To display the first few rows of the dataset to understand its structure

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

print(pima.head(), "\n")

# Calculate the mean of each column

mean\_values = pima.mean()

print("Mean of each column:")

print(mean\_values, "\n")

# Calculate the median of each column

median\_values = pima.median()

print("Median of each column:")

print(median\_values, "\n")

# Calculate the mode of each column

mode\_values = pima.mode().iloc[0] # Taking the first mode value if there are multiple modes

print("Mode of each column:")

print(mode\_values, "\n")

summary\_statistics = pima.describe()

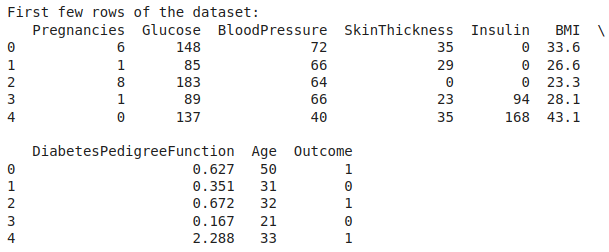
print("Summary statistics of the dataset:")

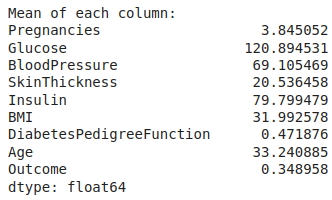
print(summary\_statistics)

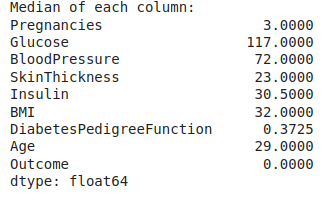
Explanation:

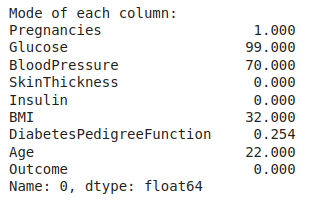
This code performs basic exploratory data analysis (EDA) on the Pima Indians Diabetes dataset. It begins by importing the necessary libraries (pandas and numpy) and loading the dataset into a DataFrame, assigning appropriate column names. The dataset's structure is inspected by displaying the first few rows. It then computes and displays the mean, median, and mode for each column, which are key measures of central tendency. Finally, a summary of statistics (e.g., count, mean, min, max, etc.) for all columns is generated using the describe method to give an overview of the dataset.

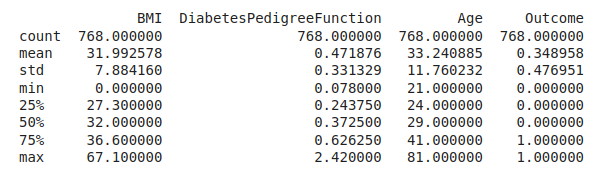
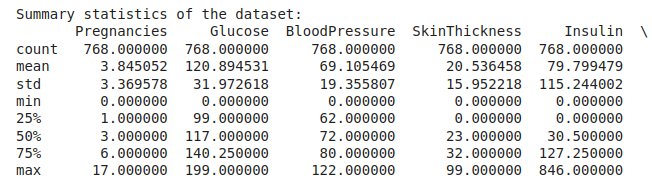
Output:











1. Write a program to illustrate the data visualization based on your own sample dataset.

Data visualization transforms raw data into graphical representations, making patterns and trends easier to interpret. This program demonstrates how to create visualizations such as bar charts, line plots, and scatter plots using a sample dataset. These visual tools are essential for effective data analysis and decision-making.

Program:

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

import numpy as np

dataset\_url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv'

columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

# Load the dataset into a DataFrame

pima = pd.read\_csv(dataset\_url, names=columns)

# Histogram for each column

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

pima.hist(bins=15, figsize=(15, 10), layout=(3, 3))

plt.suptitle('Histogram of Each Feature')

plt.tight\_layout(rect=[0, 0, 1, 0.96])

plt.show()

# Boxplot for each column

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

sns.boxplot(data=pima, orient='h', palette='Set2')

plt.title('Boxplot of Each Feature')

plt.show()

# Heatmap for correlation between the features

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

correlation\_matrix = pima.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

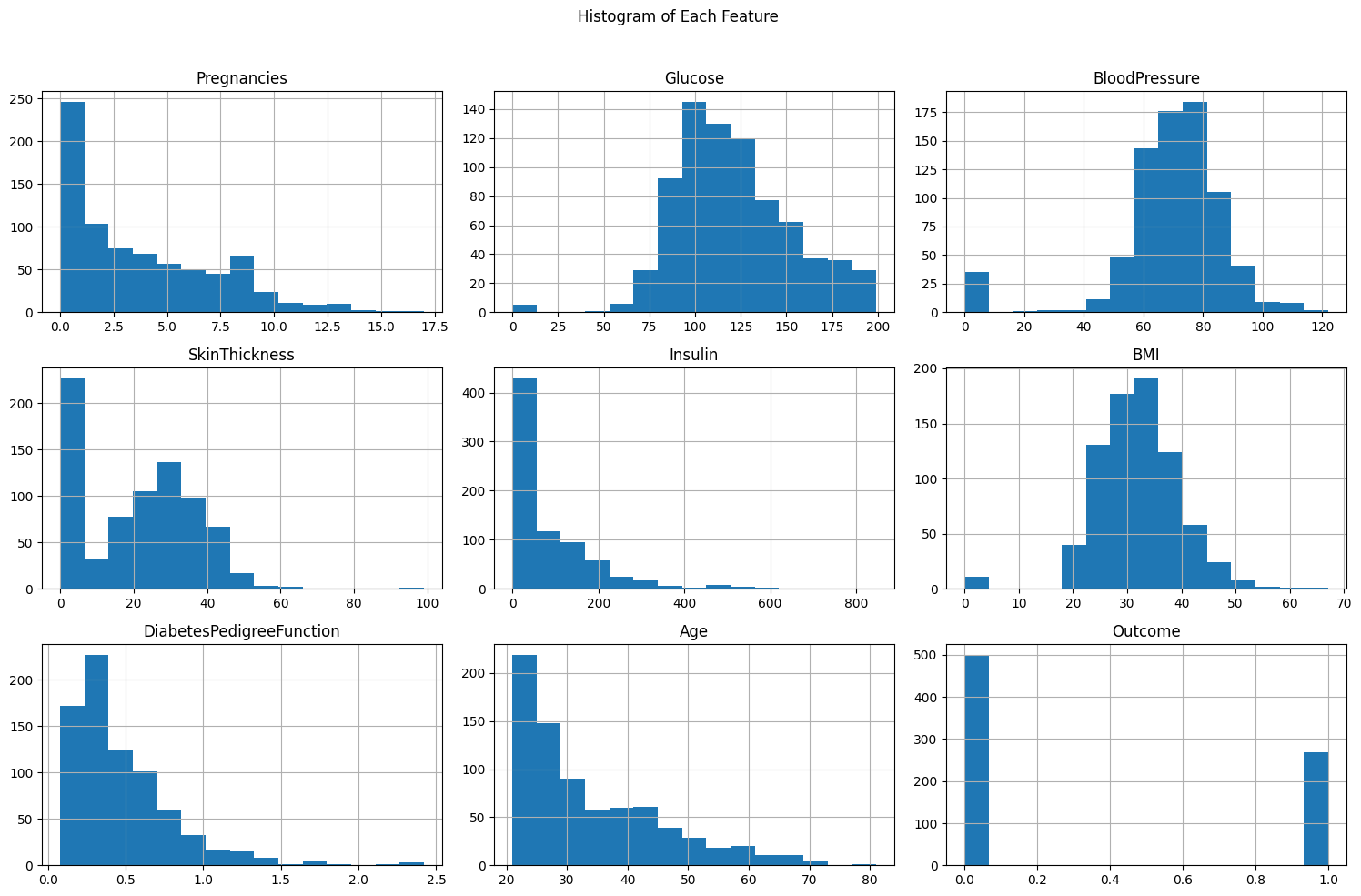
plt.title('Heatmap of Correlation Matrix')

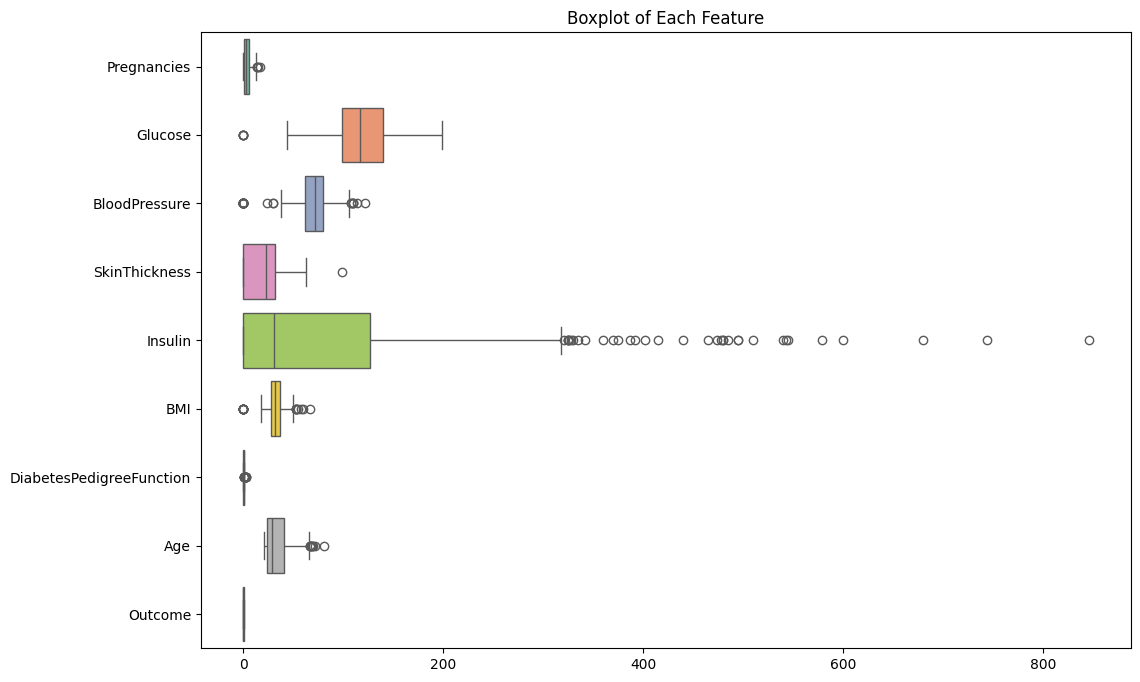
plt.show()

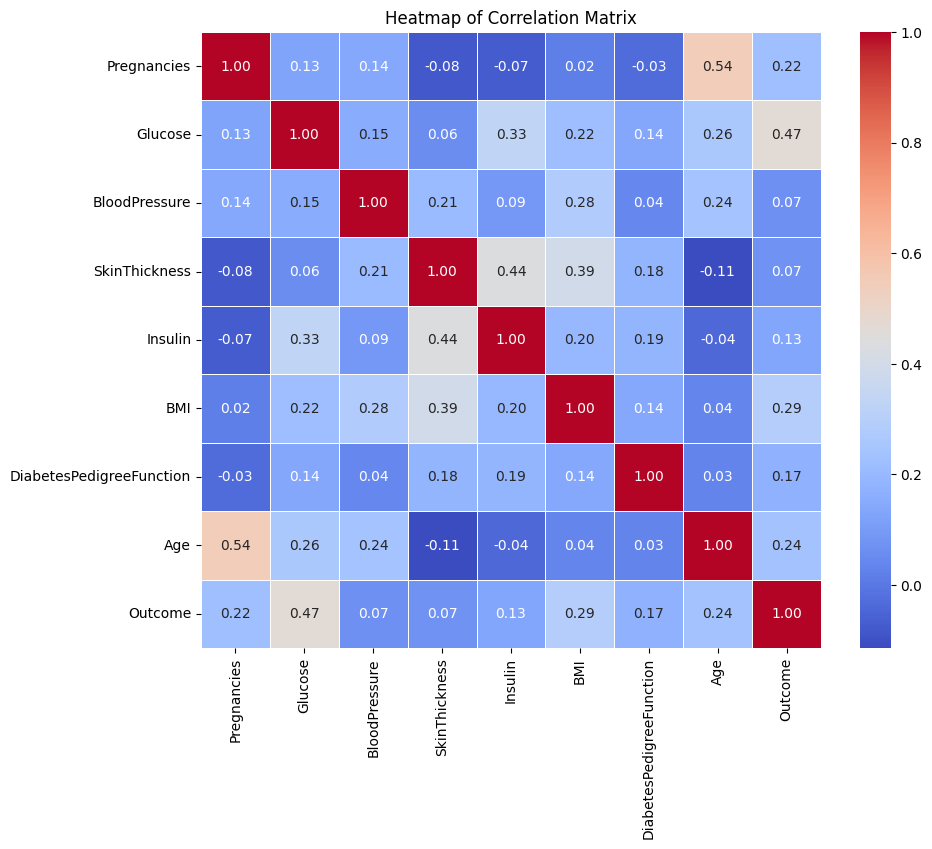
Explanation:

This code visualizes the Pima Indians Diabetes dataset using histograms, boxplots, and a heatmap to explore feature distributions and relationships. The histograms provide an overview of the frequency distribution of each feature. Boxplots are used to display the spread, central tendency, and potential outliers in the dataset. The heatmap illustrates the correlation matrix, highlighting the relationships between features using color gradients and numerical values. Together, these visualizations offer insights into the data's structure and potential patterns.

Output:







1. Write a program to perform some data preprocessing (discretization, sampling, remove duplicate, outlier, handle missing, noise handle, etc) in your own sample data set.

Data preprocessing involves preparing raw data by cleaning and transforming it into a format suitable for analysis. This program demonstrates various preprocessing techniques such as discretization, sampling, removing duplicates, handling missing values, detecting outliers, and noise handling. These steps are essential to improve the quality and accuracy of data for mining and analysis tasks.

Program:

import numpy as np

import pandas as pd

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import MinMaxScaler

# Load the dataset

dataset\_url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv'

columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

pima = pd.read\_csv(dataset\_url, names=columns)

print("Data before preprocessing:\n", pima.head())

# Handle missing values

# Replace zeros in 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', and 'BMI' with the column mean

columns\_to\_impute = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']

imputer = SimpleImputer(missing\_values=0, strategy='mean')

pima[columns\_to\_impute] = imputer.fit\_transform(pima[columns\_to\_impute])

# Remove duplicate rows

pima = pima.drop\_duplicates()

# Normalize the data

scaler = MinMaxScaler()

pima.iloc[:, :] = scaler.fit\_transform(pima)

print("\nData after preprocessing:\n", pima.head())

Explanation:

This code performs data preprocessing on the Pima Indians Diabetes dataset. It begins by loading the dataset and inspecting the data before any processing. Missing values are handled by replacing zeros in specific columns (Glucose, BloodPressure, SkinThickness, Insulin, BMI) with the column mean using SimpleImputer. Duplicate rows are removed to ensure the data is unique. Finally, the data is normalized using MinMaxScaler to scale the features to a range of 0 to 1, ensuring consistent value ranges for all features. The result is the preprocessed dataset, ready for further analysis or modeling.

Output:

