Problem Statement:

Healthcare Data Exploration –

Visualize patient data like blood pressure, sugar levels, and weight to identify health trends

Name: ASHOK GANGWAR

Roll No.: 202401100400057

Date of Submission: 10-03-2025

Introduction

In modern healthcare, data-driven analysis plays a crucial role in monitoring and improving patient health. This report explores patient data, focusing on key health indicators such as blood pressure, sugar levels, and weight. By visualizing these parameters, we aim to identify trends that could help in preventive healthcare and better medical decision-making. The dataset consists of synthetic patient records, allowing us to analyze variations and potential risk factors among individuals.

Methodology

1. **Data Collection:**
   The dataset consists of patient health metrics, including blood pressure, sugar levels, and weight. The data is preprocessed to handle missing values and inconsistencies.

2. Data Cleaning:

Duplicate entries are removed, and missing values are handled using imputation techniques.

3. Data Analysis:

Statistical analysis is performed to understand the distribution of each health parameter.

4. Visualization Techniques:

- Line graphs for tracking trends over time.
- Scatter plots to analyze correlations between parameters.
- Histograms for the distribution of health metrics.

5. Conclusion and Findings:

Insights drawn from the data are interpreted to suggest healthcare improvements.

CODE:

```
# Import required libraries for EDA and visualization

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


# Load the dataset (assuming it's a CSV file)

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/synthetic_healthcare_data_150_patients.csv')


# Display the first few rows of the dataset

print(df.head())


# Get general information (data types, non-null counts)

print(df.info())


# Get summary statistics for the numeric columns

# This provides the mean, standard deviation, min, max, and other statistics to understand the data distribution

print(df.describe())


# Checking for missing values in the dataset

# If missing values are found, you might want to either drop them or fill them with a strategy (e.g., mean or median)

print("\nMissing Values:")

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

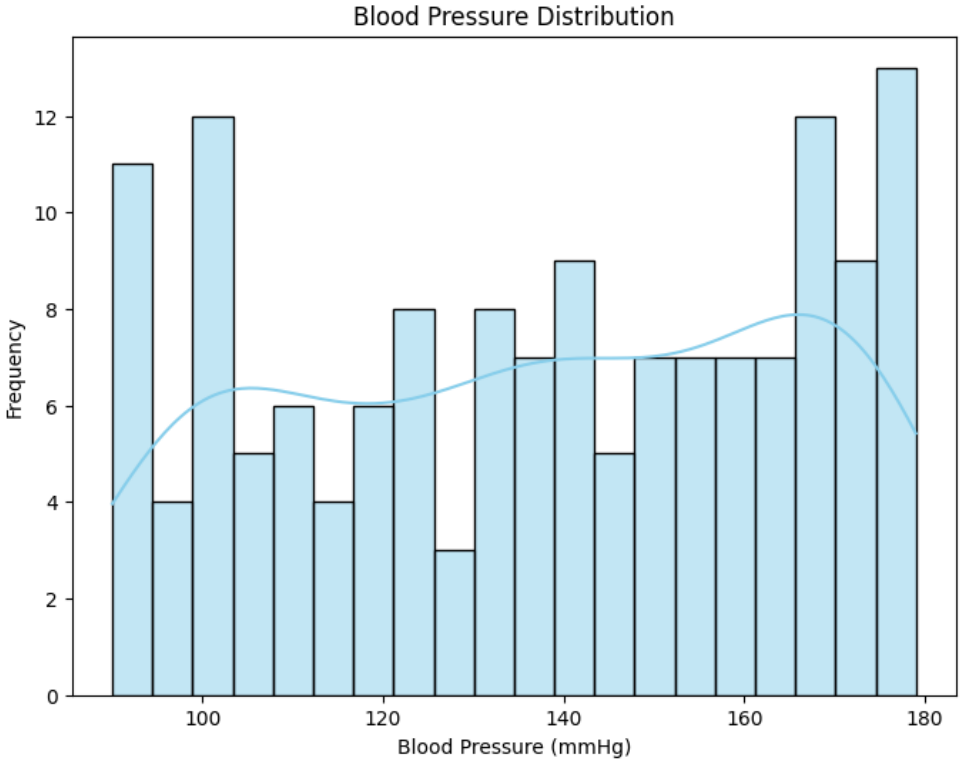
```
df['Blood_Pressure_mmHg'].fillna(df['Blood_Pressure_mmHg'].median(),
inplace=True)
```

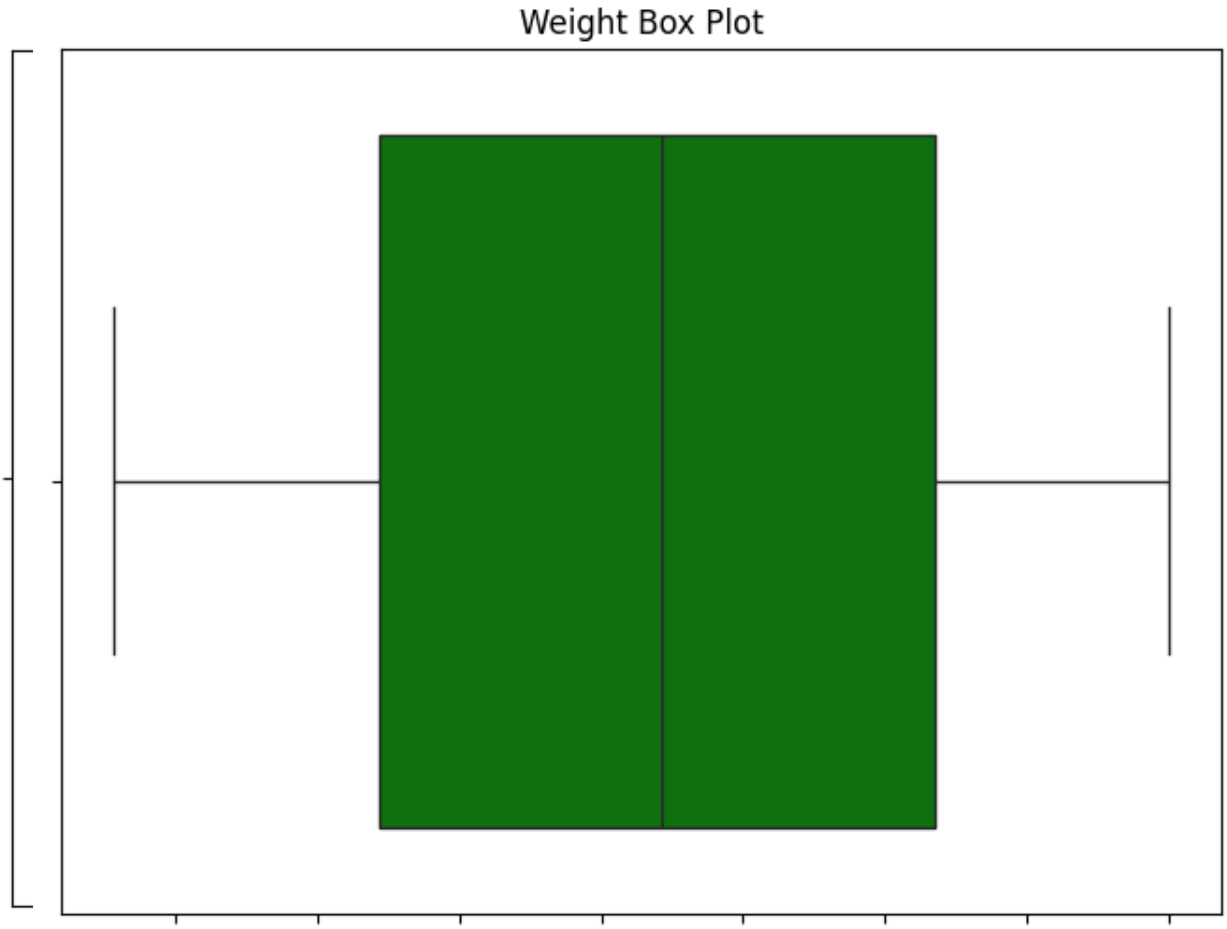
```
#Data Visualization - Visualize the distribution of key health metrics

# Blood Pressure Distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['Blood_Pressure_mmHg'], kde=True, color='skyblue', bins=20)
plt.title('Blood Pressure Distribution')
plt.xlabel('Blood Pressure (mmHg)')
plt.ylabel('Frequency')
plt.show()

# Sugar Level Distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['Sugar_Level_mg_dl'], kde=True, color='orange', bins=20)
plt.title('Sugar Level Distribution')
plt.xlabel('Sugar Level (mg/dL)')
plt.ylabel('Frequency')
plt.show()

# Weight Distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['Weight_kg'], kde=True, color='green', bins=20)
plt.title('Weight Distribution')
plt.xlabel('Weight (kg)')
plt.ylabel('Frequency')
```

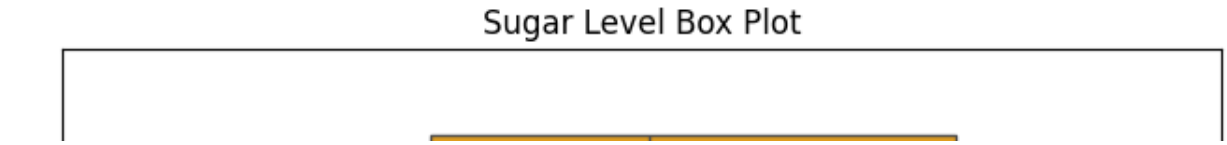
```
plt.show()

# Box plot for Blood Pressure to identify any outliers in the data
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Blood_Pressure_mmHg'], color='skyblue')
plt.title('Blood Pressure Box Plot')
plt.xlabel('Blood Pressure (mmHg)')
plt.show()


# Sugar Level Box Plot
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Sugar_Level_mg_dl'], color='orange')
plt.title('Sugar Level Box Plot')
plt.xlabel('Sugar Level (mg/dL)')
plt.show()

# Weight Box Plot
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Weight_kg'], color='green')
plt.title('Weight Box Plot')
plt.xlabel('Weight (kg)')
plt.show()

# Pair Plot to explore relationships
sns.pairplot(df[['Age', 'Weight_kg', 'Blood_Pressure_mmHg',
'Sugar_Level_mg_dl', 'BMI']])
plt.suptitle('Pair Plot of Key Health Metrics', y=1.02)
```

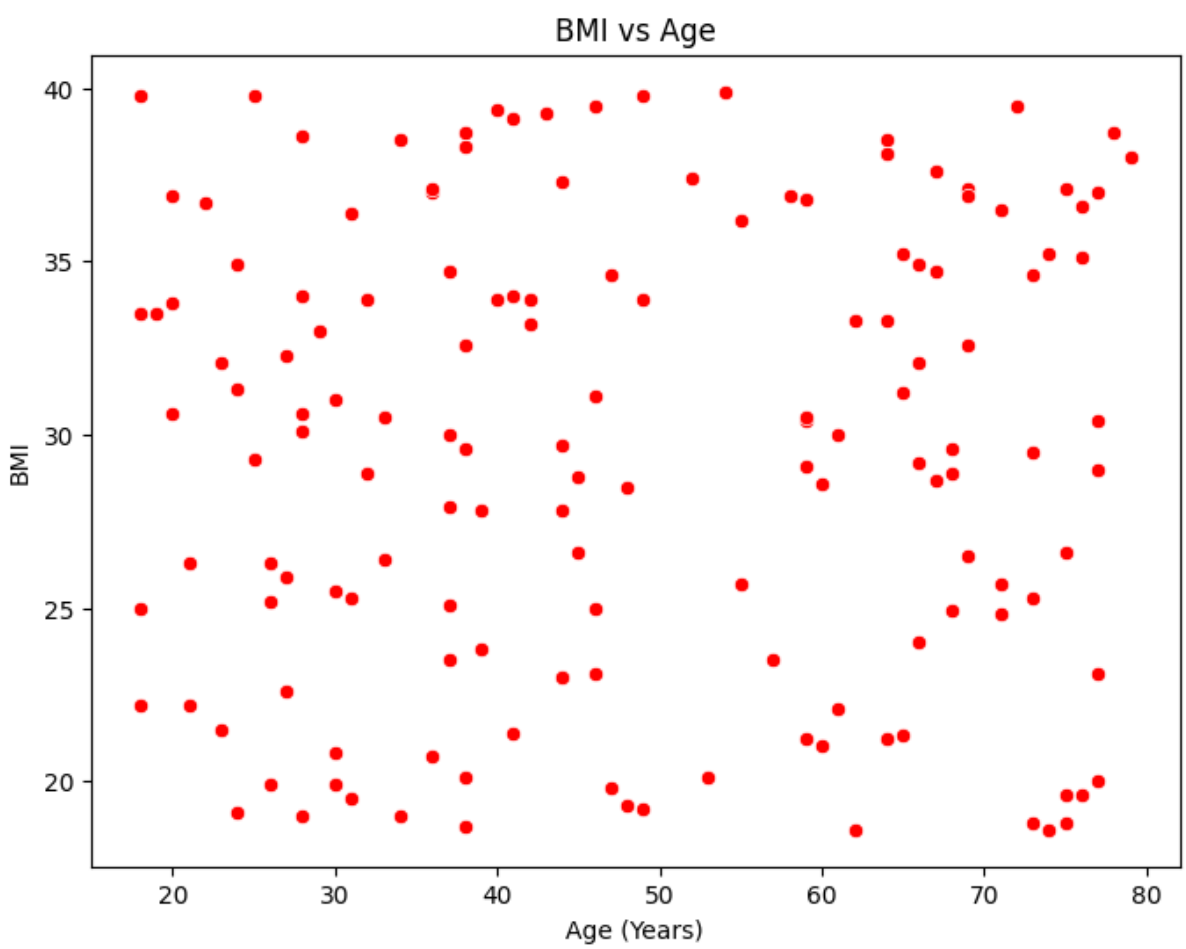
```
plt.show()

# Scatter Plot between Blood Pressure and Sugar Levels
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df['Blood_Pressure_mmHg'], y=df['Sugar_Level_mg_dl'], color='purple')
plt.title('Blood Pressure vs Sugar Levels')
plt.xlabel('Blood Pressure (mmHg)')
plt.ylabel('Sugar Level (mg/dL)')
plt.show()

# BMI Distribution Based on Age
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df['Age'], y=df['BMI'], color='red')
plt.title('BMI vs Age')
plt.xlabel('Age (Years)')
plt.ylabel('BMI')
plt.show()
```

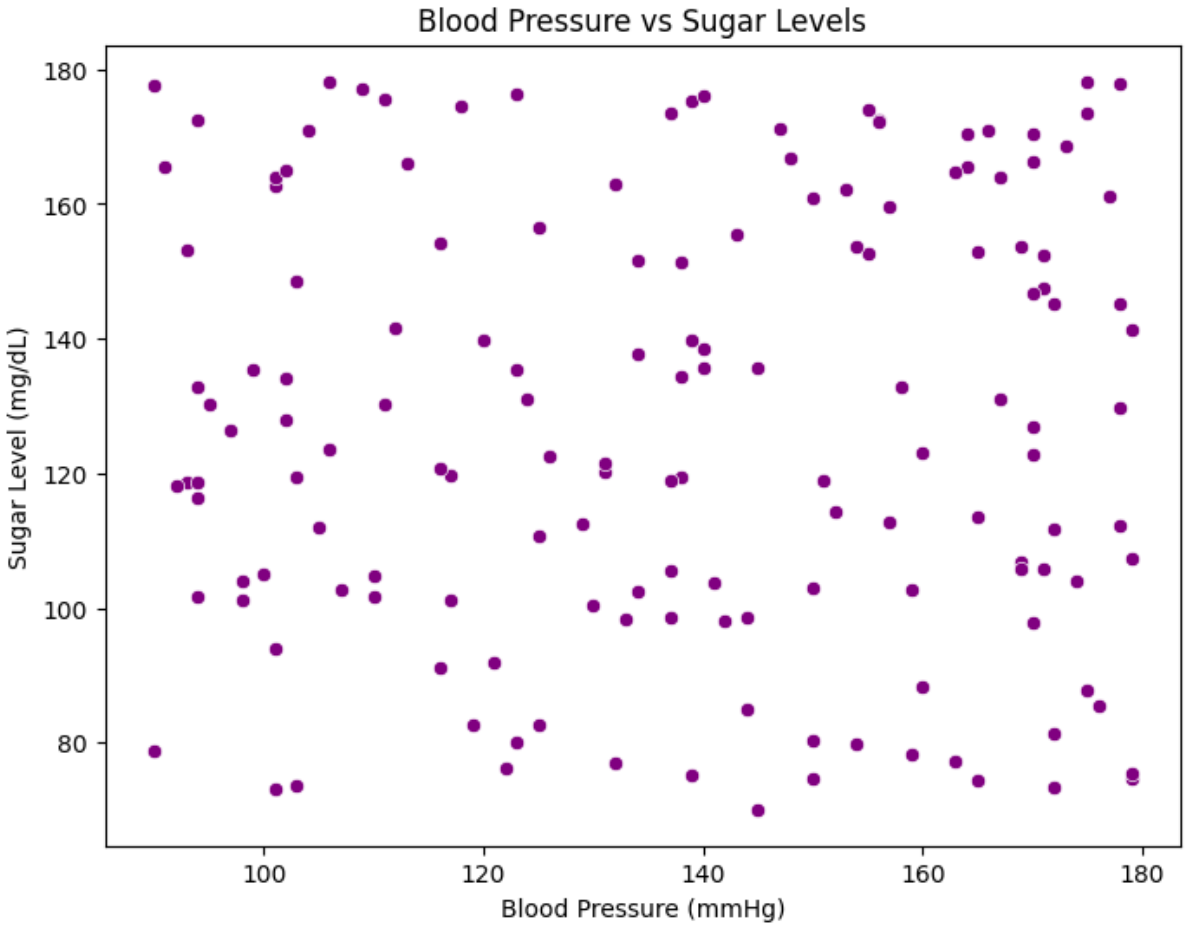
Output/Result

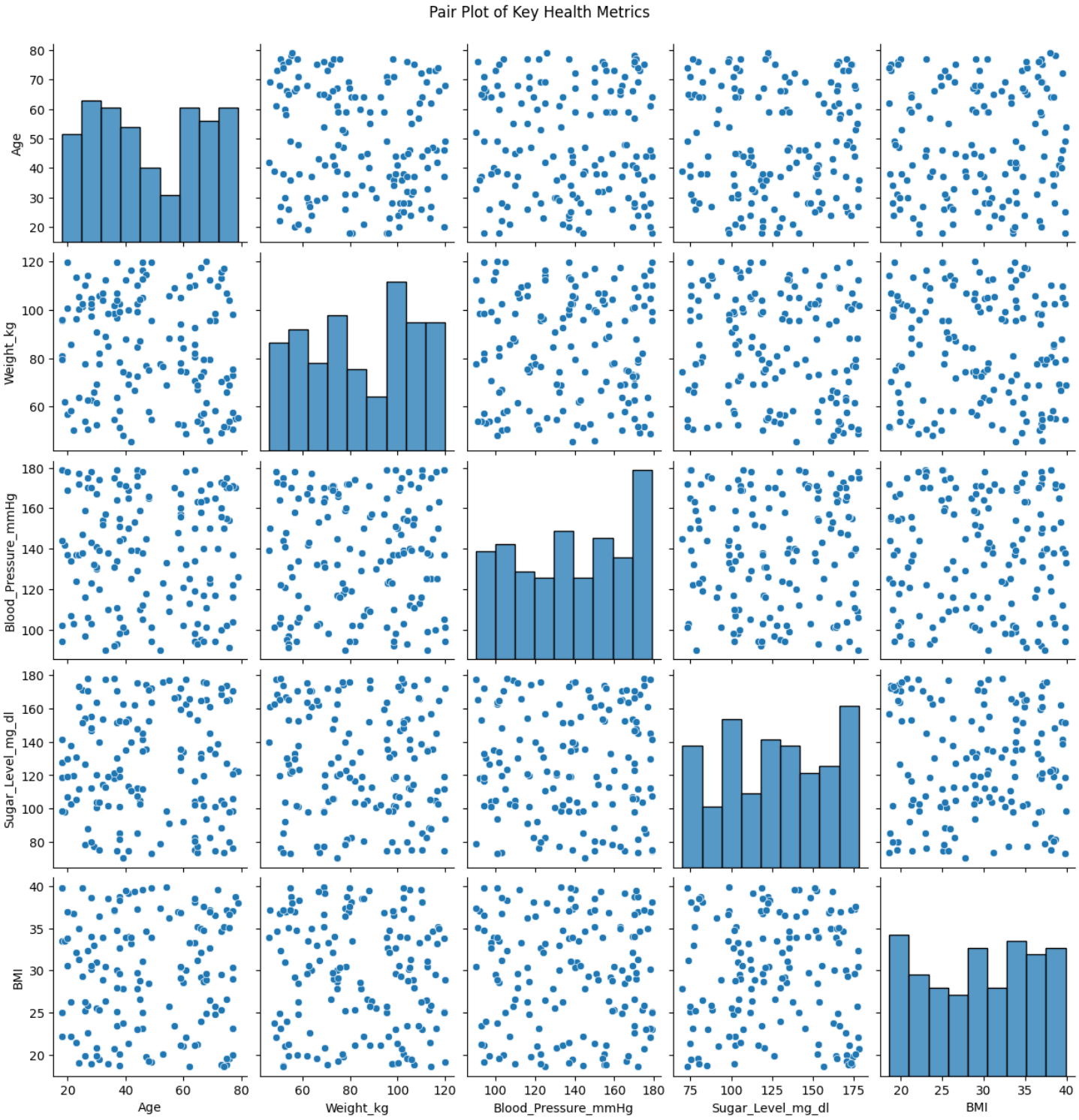



Weight Box Plot


Sugar Level Box Plot


BMI vs Age
BMI
Age (Years)
40
35
30
25
20
20
30
40
50
60
70
80


Blood Pressure vs Sugar Levels
Sugar Level (mg/dL)
180
160
140
120
100
80
100
120
140
160
180
Blood Pressure (mmHg)


Pair Plot of Key Health Metrics
Age
Weight_kg
Blood_Pressure_mmHg
Sugar_Level_mg_dl
BMI

References/Credits

- Dataset Source: Synthetic healthcare data.
- Python Libraries: Pandas, Matplotlib.
- External Resources: Articles on healthcare data visualization and trends.