Project Title: *Healthcare Data Exploration – Visualizing Patient Data to Identify Health Trends*

Introduction

Overview

Healthcare data analysis plays a crucial role in monitoring patient health and identifying risks early. By analyzing key health metrics such as **blood pressure, sugar levels, and weight**, patterns and trends can be identified to assist in preventive care and medical decision-making.

Objectives

- To collect and analyze patient health data.
- To identify trends in blood pressure, sugar levels, and weight.
- To visualize data using Python libraries such as **Pandas, Matplotlib, and Seaborn**.
- To understand the correlation between different health parameters.

Scope of Study

The project focuses on **data visualization techniques** to analyze healthcare trends. It does not provide medical diagnosis or treatment recommendations but aims to present insightful trends through data analysis.

Methodology

Data Collection

- Patient health metrics (blood pressure, sugar levels, and weight) are collected from **synthetic sample data** for this study.
- The dataset is structured and stored in a **Pandas DataFrame** for further analysis.

Data Preprocessing

- Handling of **missing values** (if any).
- Data **normalization and standardization** to improve visualization accuracy.

Data Analysis & Visualization

- **Descriptive statistics** such as mean, median, and standard deviation.
- **Histograms** to show the distribution of each health parameter.

- **Pairplots** to visualize relationships between variables.
- **Heatmaps** to identify correlations among blood pressure, sugar levels, and weight.

Tools & Technologies Used

- **Python**: Primary language for data analysis.
- **Pandas**: Data manipulation and handling.
- **Matplotlib & Seaborn**: Libraries for data visualization.

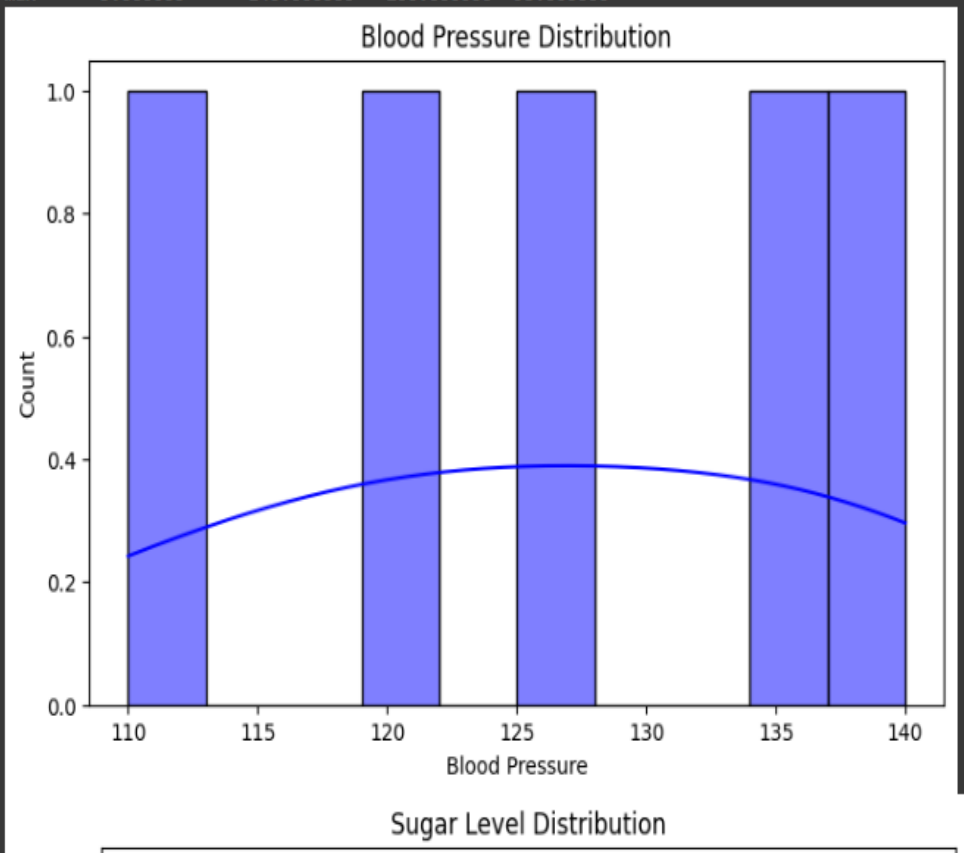
CODE :

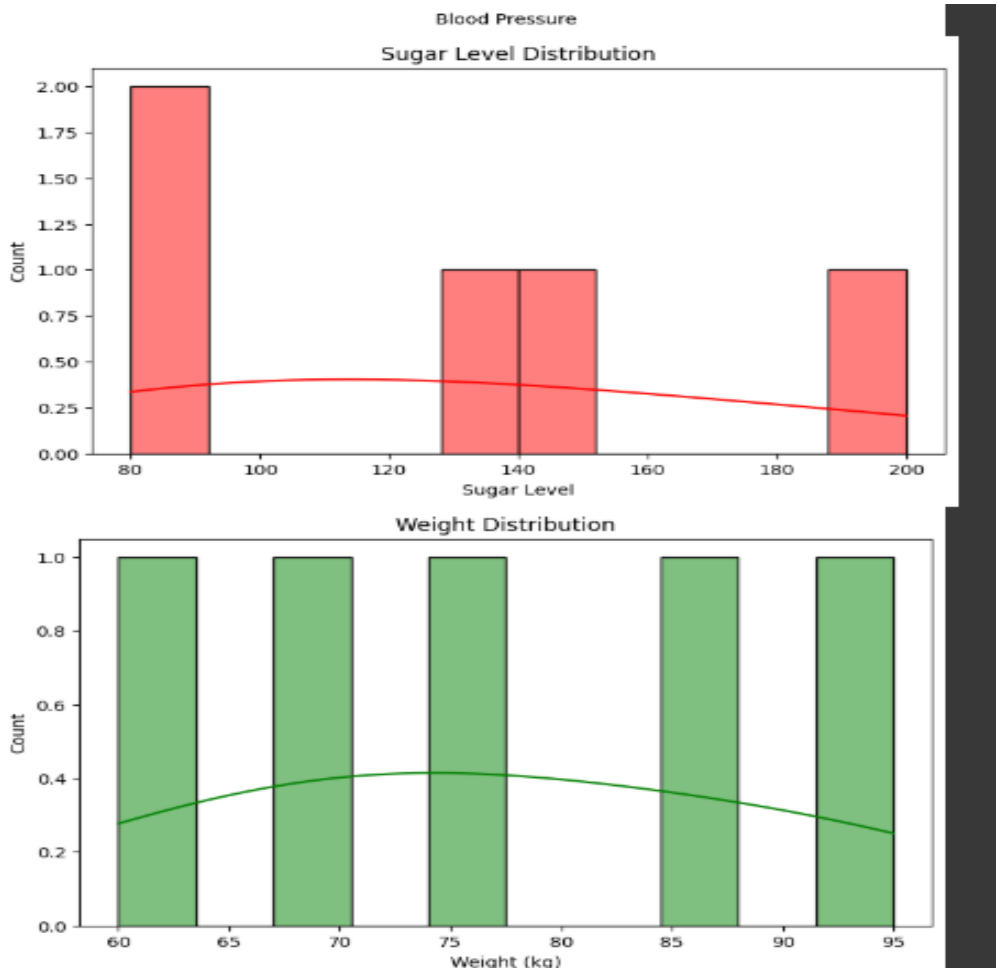
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data (You can replace this with actual patient data)
data = {
   "Patient_ID": [1, 2, 3, 4, 5],
   "Blood_Pressure": [120, 135, 110, 140, 125],
   "Sugar_Level": [90, 150, 80, 200, 130],
   "Weight": [70, 85, 60, 95, 75]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Display basic statistics
print("Basic Statistics:\n", df.describe())

# Visualizing Blood Pressure
plt.figure(figsize=(8, 5))
sns.histplot(df["Blood_Pressure"], kde=True, bins=10, color='blue')
plt.title("Blood Pressure Distribution")
plt.xlabel("Blood Pressure")
plt.ylabel("Count")
plt.show()


# Visualizing Sugar Levels

plt.figure(figsize=(8, 5))

sns.histplot(df["Sugar_Level"], kde=True, bins=10, color='red')

plt.title("Sugar Level Distribution")

plt.xlabel("Sugar Level")

plt.ylabel("Count")

plt.show()
```

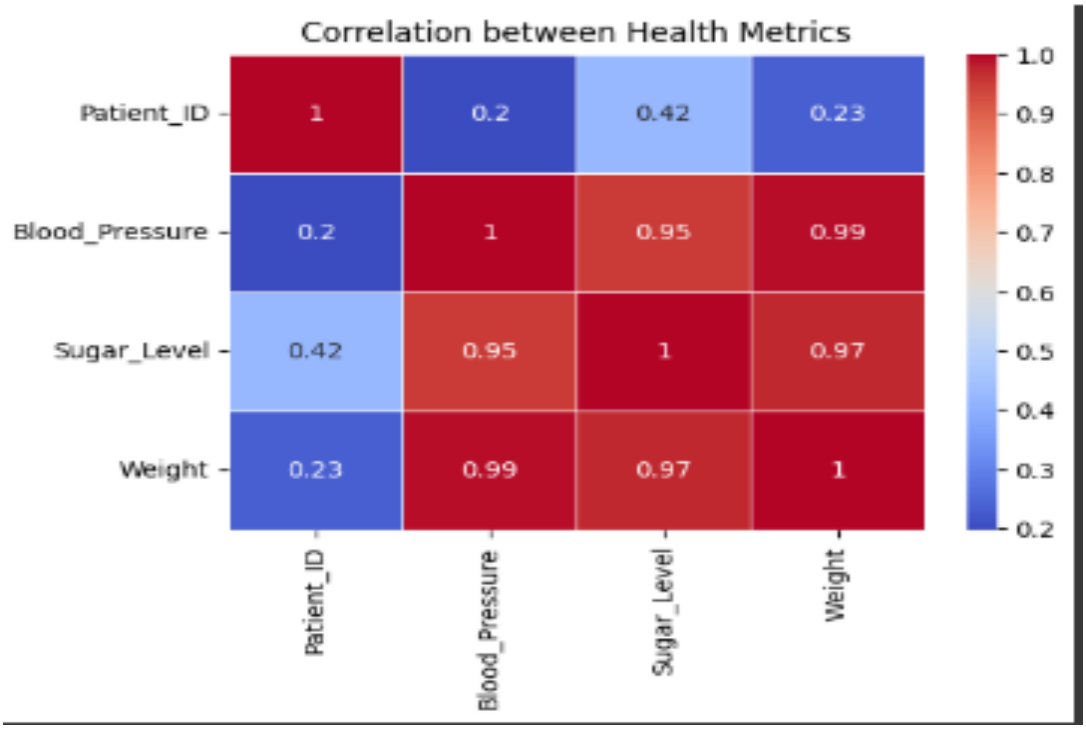
```
# Visualizing Weight
plt.figure(figsize=(8, 5))
sns.histplot(df["Weight"], kde=True, bins=10, color='green')
plt.title("Weight Distribution")
plt.xlabel("Weight (kg)")
plt.ylabel("Count")
plt.show()

# Pairplot to see relationships
sns.pairplot(df, diag_kind='kde')
plt.show()

# Correlation Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation between Health Metrics")
plt.show()
```

Basic Statistics:

	Patient_ID	Blood_Pressure	Sugar_Level	Weight
count	5.000000	5.000000	5.000000	5.000000
mean	3.000000	126.000000	130.000000	77.000000
std	1.581139	11.937336	48.476799	13.509256
min	1.000000	110.000000	80.000000	60.000000
25%	2.000000	120.000000	90.000000	70.000000
50%	3.000000	125.000000	130.000000	75.000000
75%	4.000000	135.000000	150.000000	85.000000
max	5.000000	140.000000	200.000000	95.000000




Blood Pressure
Sugar Level Distribution
Count
Sugar Level
Weight Distribution
Count
Weight (kg)


Correlation between Health Metrics
Patient_ID
Blood_Pressure
Sugar_Level
Weight
1
0.2
0.42
0.23
0.2
1
0.95
0.99
0.42
0.95
1
0.97
0.23
0.99
0.97
1