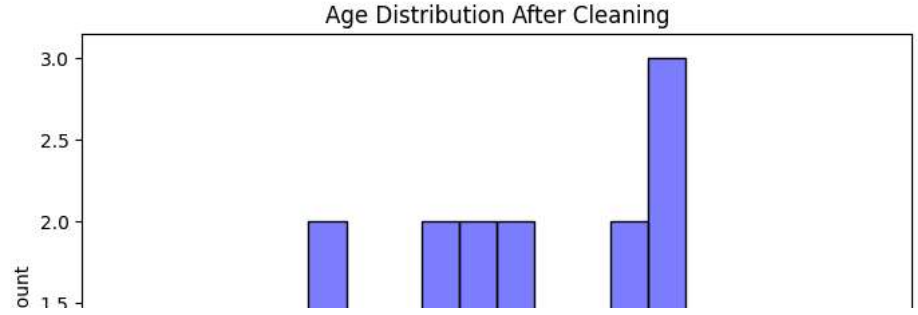
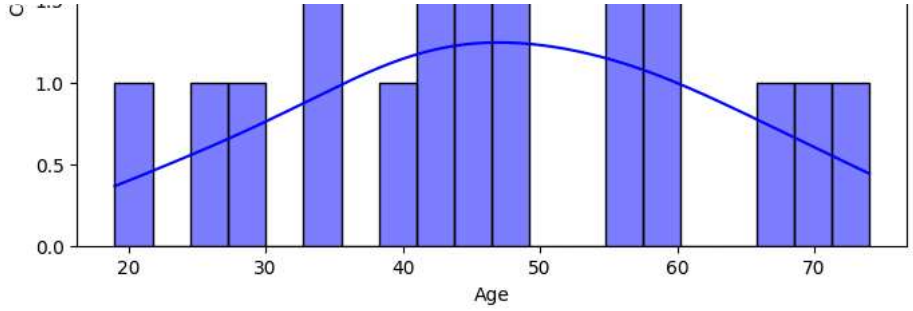
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
import shutil


# Load the dataset
file_path = '/content/healthcare_data.csv'  # Ensure this matches the uploaded file name
df = pd.read_csv(file_path)


# Display basic information
print("Dataset Overview:")
print(df.info())
print("\nFirst 5 rows:")
print(df.head())

# Visualize missing values
plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cmap='viridis', cbar=False)
plt.title('Missing Values Heatmap')
plt.show()

# Handling missing values
# Fill numerical columns with median, categorical columns with mode
for col in df.columns:
    if df[col].dtype == 'object':
        df[col].fillna(df[col].mode()[0], inplace=True)
    else:
        df[col].fillna(df[col].median(), inplace=True)

# Removing duplicates
duplicate_count = df.duplicated().sum()
print(f"\nNumber of duplicate rows removed: {duplicate_count}")
df.drop_duplicates(inplace=True)

# Checking inconsistencies (example: standardizing categorical values)
if 'Gender' in df.columns:
    df.replace({'Male': 'M', 'Female': 'F', 'male': 'M', 'female': 'F'}, inplace=True)

# Identifying outliers using IQR method and capping extreme values
for col in df.select_dtypes(include=np.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
    df[col] = np.where(df[col] < lower_bound, lower_bound, df[col])
    df[col] = np.where(df[col] > upper_bound, upper_bound, df[col])

# Summary after cleaning
print("\nData after Cleaning:")
print(df.info())

# Visualization: Distribution of a numerical column (e.g., Age if exists)
if 'Age' in df.columns:
    plt.figure(figsize=(8,5))
    sns.histplot(df['Age'], bins=20, kde=True, color='blue')
    plt.title('Age Distribution After Cleaning')
    plt.xlabel('Age')
    plt.ylabel('Count')
    plt.show()

# Visualization: Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Heatmap of Numeric Features')
plt.show()

# Visualization: Boxplot for Outlier Analysis
if 'Age' in df.columns:
    plt.figure(figsize=(8,5))
    sns.boxplot(x=df['Age'], color='red')
    plt.title('Boxplot of Age After Outlier Handling')
    plt.show()

# Additional Visualizations
# Count plot for categorical features
for col in df.select_dtypes(include=['object']).columns:
    plt.figure(figsize=(8, 5))
```

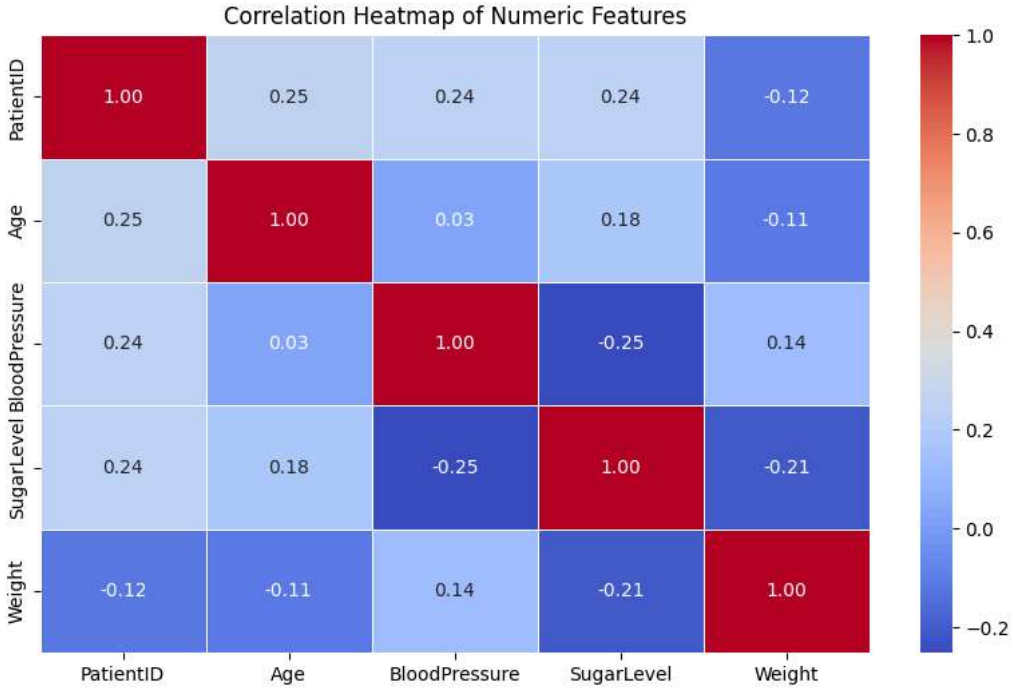
```
        sns.countplot(x=df[col], palette='Set2')
        plt.title(f'Count Plot of {col}')
        plt.xticks(rotation=45)
        plt.show()

    # Pairplot for numerical data
    sns.pairplot(df.select_dtypes(include=np.number))
    plt.show()

    # Violin plot for distribution analysis
    for col in df.select_dtypes(include=np.number).columns:
        plt.figure(figsize=(8, 5))
        sns.violinplot(y=df[col], palette='muted')
        plt.title(f'Violin Plot of {col}')
        plt.show()

    # Save cleaned dataset
    df.to_csv('cleaned_healthcare_data.csv', index=False)
    shutil.move('cleaned_healthcare_data.csv', '/content/cleaned_healthcare_data.csv')
    files.download('/content/cleaned_healthcare_data.csv')
```

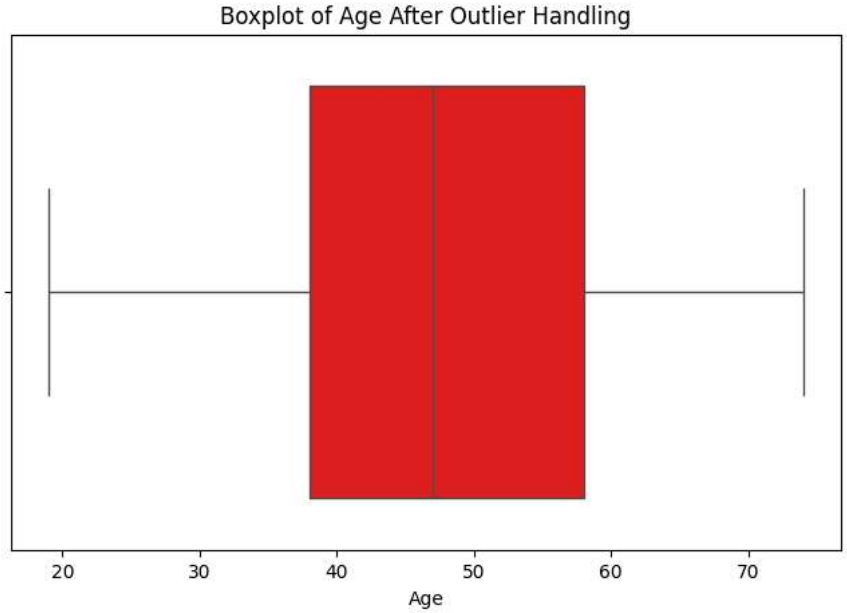
```
Dataset Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   PatientID      20 non-null     float64
 1   Age            20 non-null     float64
 2   BloodPressure  20 non-null     float64
 3   SugarLevel     20 non-null     float64
 4   Weight         20 non-null     float64
dtypes: float64(5)
memory usage: 932.0 bytes
None

First 5 rows:
   PatientID   Age  BloodPressure  SugarLevel      Weight
0        1.0  44.0          118.0   87.892495  105.568034
1        2.0  39.0          109.0  177.321803  105.703426
2        3.0  49.0          149.0  144.148273   77.787070
3        4.0  58.0          121.0   90.355404  115.244784
4        5.0  35.0          109.0  126.421800   70.383790
```

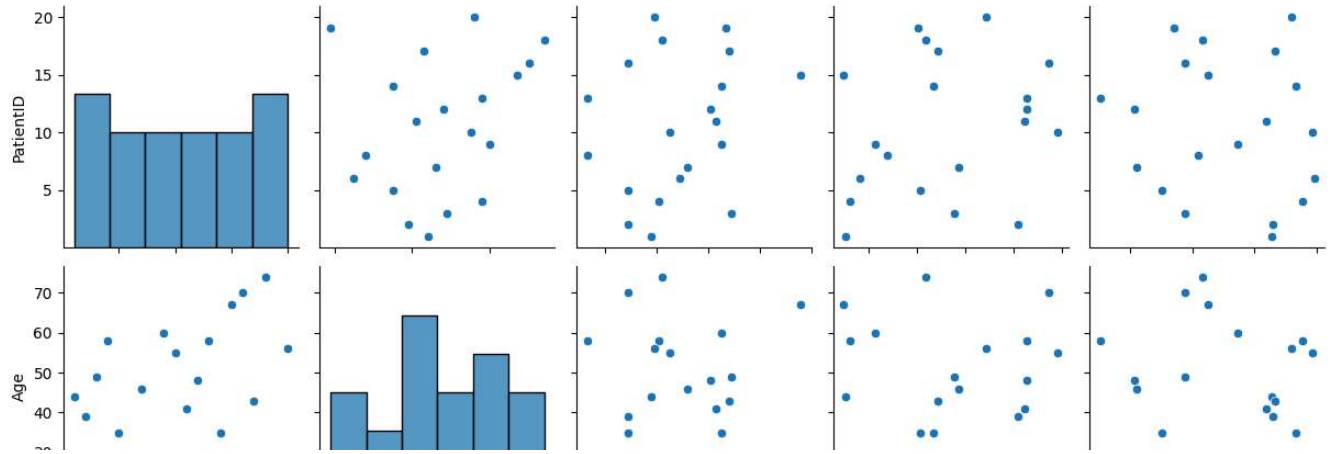

```
<ipython-input-7-c1c22ff2f849>:19: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col


  df[col].fillna(df[col].median(), inplace=True)

Number of duplicate rows removed: 0

Data after Cleaning:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   PatientID      20 non-null     float64
 1   Age            20 non-null     float64
 2   BloodPressure  20 non-null     float64
 3   SugarLevel     20 non-null     float64
 4   Weight         20 non-null     float64
dtypes: float64(5)
memory usage: 932.0 bytes
None
```

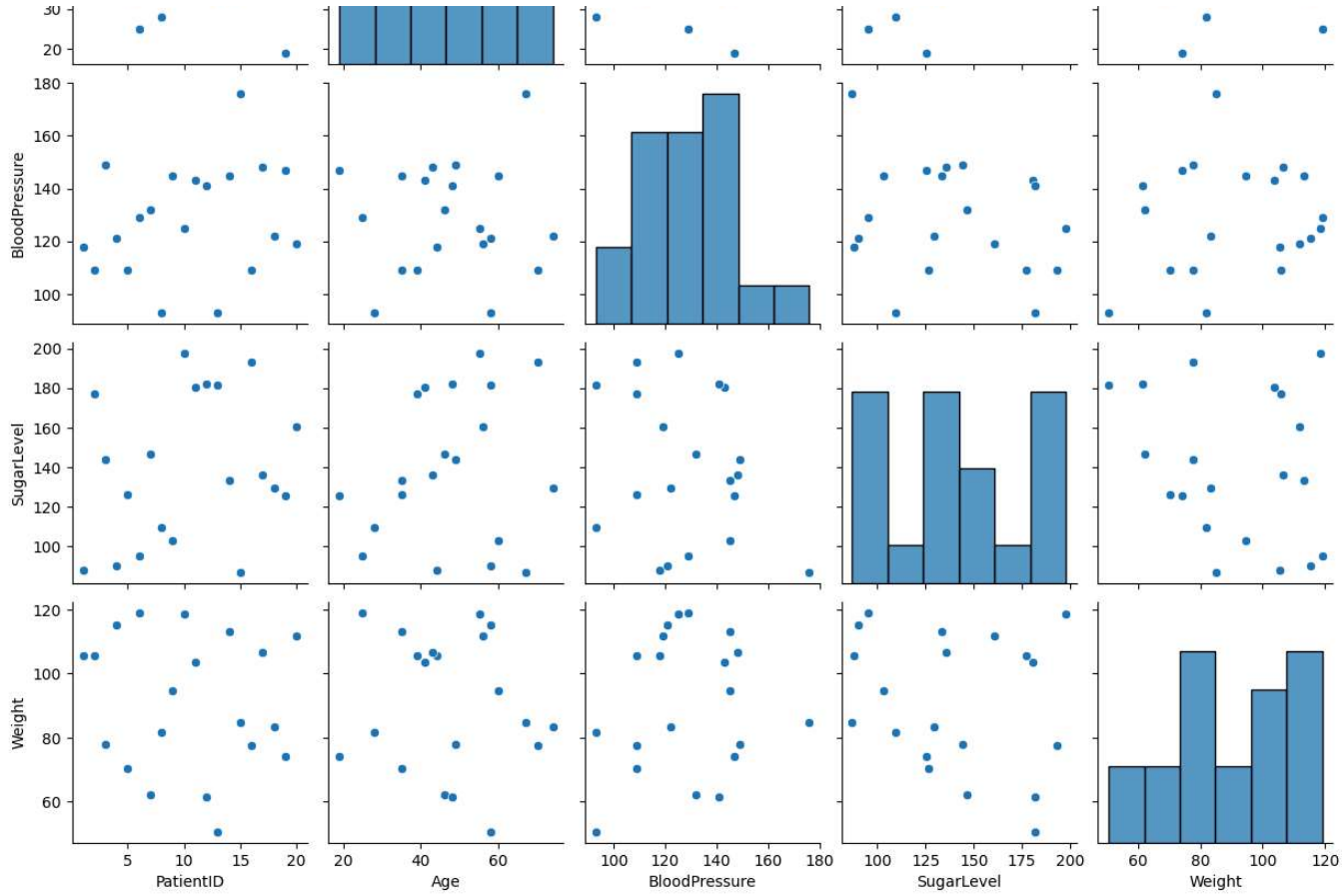



1.0
0.5
0.0
20
30
40
50
60
70
Age

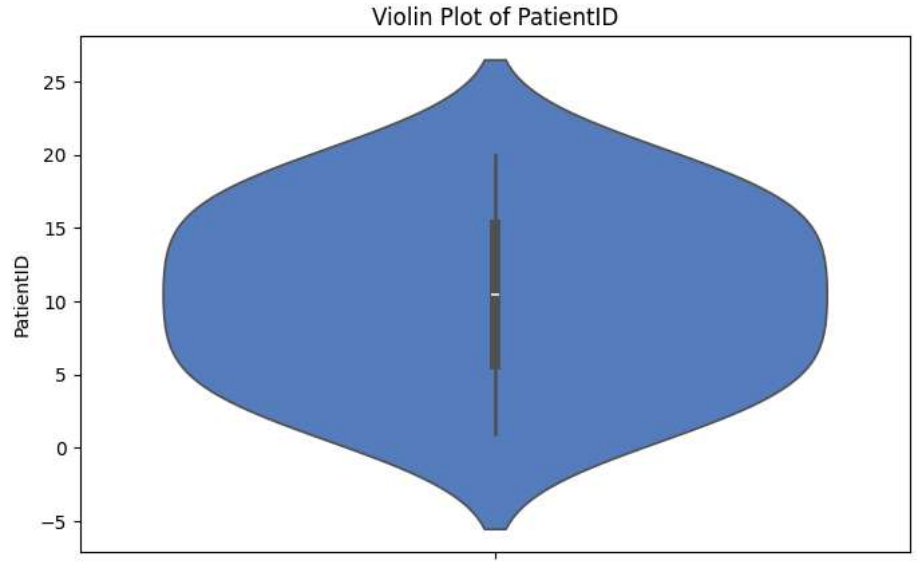

Correlation Heatmap of Numeric Features
PatientID
Age
BloodPressure
SugarLevel
Weight
1.00 0.25 0.24 0.24 -0.12
0.25 1.00 0.03 0.18 -0.11
0.24 0.03 1.00 -0.25 0.14
0.24 0.18 -0.25 1.00 -0.21
-0.12 -0.11 0.14 -0.21 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Boxplot of Age After Outlier Handling
20
30
40
50
60
70
Age

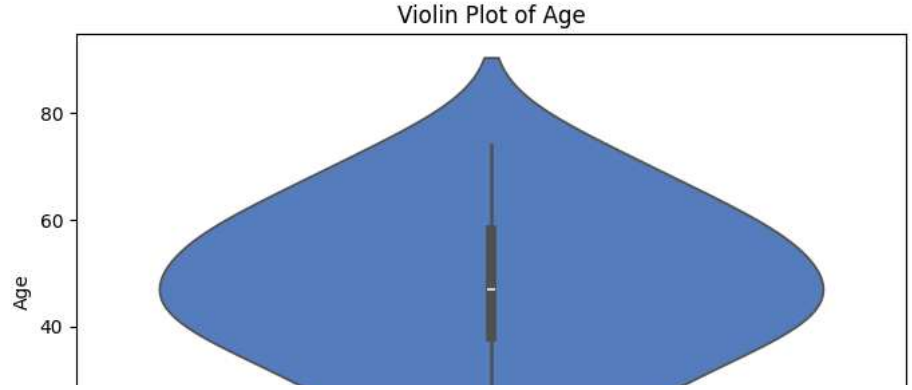

PatientID
Age
20
15
10
5
70
60
50
40

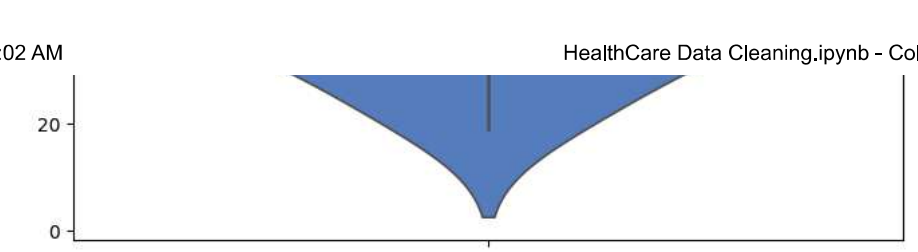



```
<ipython-input-7-c1c22ff2f849>:82: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.violinplot(y=df[col], palette='muted')
```

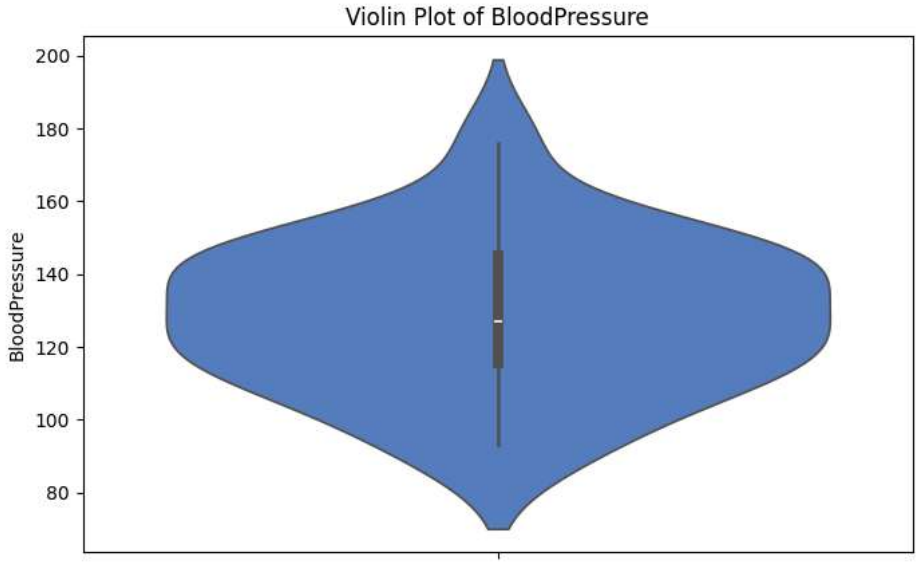



```
<ipython-input-7-c1c22ff2f849>:82: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.violinplot(y=df[col], palette='muted')
```

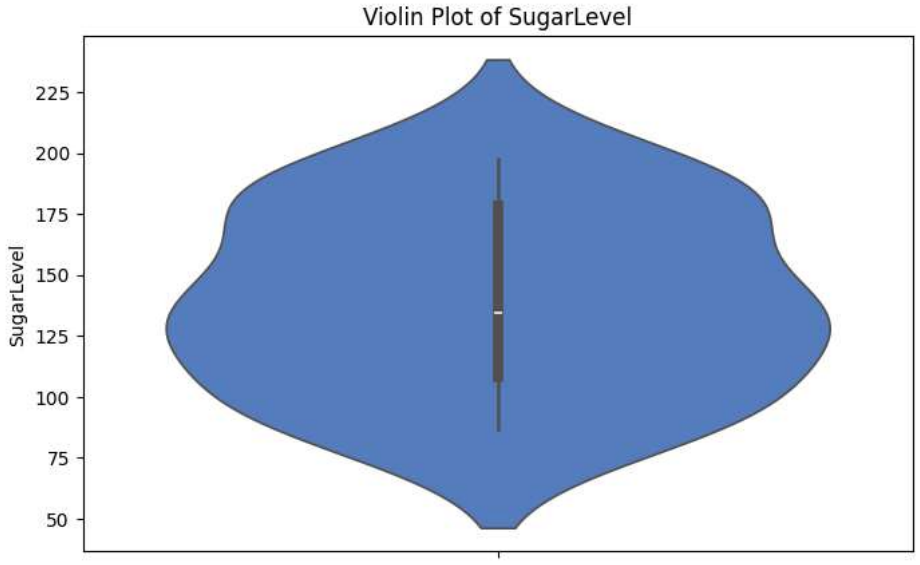





```
<ipython-input-7-c1c22ff2f849>:82: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.violinplot(y=df[col], palette='muted')
```




```
<ipython-input-7-c1c22ff2f849>:82: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.violinplot(y=df[col], palette='muted')
```




```
<ipython-input-7-c1c22ff2f849>:82: FutureWarning:
```