**This example uses the Pandas library for data manipulation and Seaborn for visualization.**

# Importing necessary libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Load the dataset

# Assuming the dataset is stored in a file named 'diabetes\_dataset.csv'

data = pd.read\_csv('diabetes\_dataset.csv')

# Data preprocessing (cleaning and structuring the dataset)

# For the purpose of this example, we assume that the dataset has been preprocessed

# Generating key visualizations

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

(Tested: No functional data to use, updated with functioning code)   
 # Visualization 1: Scatter Plot of HbA1c vs. Diabetic Microvascular Complications

# Clean and convert 'HbA1c (mmol/mol)' column to numeric

df3\_clean['HbA1c (mmol/mol)'] = pd.to\_numeric(df3\_clean['HbA1c (mmol/mol)'], errors='coerce')

# Clean and convert 'Diabetic Microvascular Complications' column to numeric

# For example, if '/' indicates missing or invalid data, you can replace it with NaN

df3\_clean['Diabetic Microvascular Complications'] = df3\_clean['Diabetic Microvascular Complications'].replace('/', np.nan)

df3\_clean['Diabetic Microvascular Complications'] = pd.to\_numeric(df3\_clean['Diabetic Microvascular Complications'], errors='coerce')

sns.scatterplot(x='HbA1c (mmol/mol)', y='Diabetic Microvascular Complications', data=df3\_clean)

plt.title('Scatter Plot of HbA1c vs. Diabetic Microvascular Complications')

plt.show()

(Tested: Functions as is)

# Visualization 2: Bar Chart of Hypoglycemia Incidence

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

sns.countplot(x='Hypoglycemia (yes/no)', data=df3\_clean)

plt.title('Bar Chart of Hypoglycemia Incidence')

plt.show()

(Tested: Code works as is, does not plot any data)

# Visualization 3: Box Plot of Fasting Plasma Glucose by Presence of Microvascular Complications

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

sns.boxplot(x='Diabetic Microvascular Complications', y='Fasting Plasma Glucose (mg/dl)', data=data)

plt.title('Box Plot of Fasting Plasma Glucose by Presence of Microvascular Complications')

plt.show()

(Tested: May work once cleanup is complete. Code is updated)

# Visualization 4: Line Plot of HbA1c Levels Over Time

# Assuming a column 'Duration of diabetes (years)' in the dataset

# Replace '/' with NaN in 'Merge DoD (years)' column

df3\_clean['Merge DoD (years)'] = pd.to\_numeric(df3\_clean['Merge DoD (years)'], errors='coerce')

# Remove rows with NaN values in 'Merge DoD (years)' column

df3\_clean.dropna(subset=['Merge DoD (years)'], inplace=True)

# Plot the line plot after cleaning the data

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

sns.lineplot(x='Merge DoD (years)', y='HbA1c (mmol/mol)', data=df3\_clean)

plt.title('Line Plot of HbA1c Levels Over Time')

plt.xlabel('Merge DoD (years)')

plt.ylabel('HbA1c (mmol/mol)')

plt.show()

(Tested: Code works as is. Looks good, will be better once clean up is complete.)

# Visualization 5: Histogram of Age Distribution by Microvascular Complications

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

sns.histplot(data=df3\_clean, x='Age (years)', hue='Diabetic Microvascular Complications', multiple='stack')

plt.title('Histogram of Age Distribution by Microvascular Complications')

plt.show()

(Tested: Updated with functional code. Will look better after clean update)

# Visualization 6: Heatmap of Correlation Matrix

# Drop non-numeric columns before calculating the correlation matrix

df\_numeric = df3\_clean.select\_dtypes(include=['number'])

# Calculate the correlation matrix

correlation\_matrix = df\_numeric.corr()

# Plot the heatmap of the correlation matrix

plt.figure(figsize=(12, 10))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Heatmap of Correlation Matrix')

plt.show()

(Tested: Works great, needs some spacing/labeling work done on the graph.)

# Visualization 7: Box Plot of BMI by Microvascular Complications

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

sns.boxplot(x='Diabetic Microvascular Complications', y='BMI (kg/m2)', data=df3\_clean)

plt.title('Box Plot of BMI by Microvascular Complications')

plt.show()

# Visualization 8: Stacked Bar Chart of Diabetic Macrovascular Complications by Gender and HbA1c Levels

# Convert numeric values in 'Gender (Female=1, Male=2)' column to meaningful labels

df3\_clean['Gender (Female=1, Male=2)'] = df3\_clean['Gender (Female=1, Male=2)'].map({1: 'Female', 2: 'Male'})

(Tested: Works as is, just needs spacing and labeling updates and nan clean up)

# Visualization 8: Stacked Bar Chart of Diabetic Macrovascular Complications by Gender and HbA1c Levels

# Convert numeric values in 'Gender (Female=1, Male=2)' column to meaningful labels

df3\_clean['Gender (Female=1, Male=2)'] = df3\_clean['Gender (Female=1, Male=2)'].map({1: 'Female', 2: 'Male'})

plt.figure(figsize=(20, 12))

sns.countplot(x='HbA1c (mmol/mol)', hue='Gender (Female=1, Male=2)', data=df3\_clean)

plt.title('Stacked Bar Chart of Diabetic Macrovascular Complications by Gender and HbA1c Levels')

plt.show()