```
from google.colab import files
Uploaded = files.Upload()


Capstone Project: EDA on Diabetes Dataset (Q1 to Q20)

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

Load the dataset

df = pd.read_csv("diabetes.csv")

Q1: How many rows and columns are there in the dataset?

print("Q1:", df.shape)

Q2: What are the data types of each column?

print("Q2:\n", df.dtypes)

Q3: Are there any missing or zero values in critical columns?

print("Q3:\n", df[['Glucose', 'BMI', 'Insulin']].isin([0]).sum())

Q4: Distribution of the Outcome variable

print("Q4:\n", df['Outcome'].value_counts())

Q5: Histograms for Glucose, BMI, and Age

for col in ['Glucose', 'BMI', 'Age']: sns.histplot(df[col], kde=True) plt.title(f'Histogram

Q6: Feature with highest mean and lowest std

print("Q6 - Highest mean:", df.mean().idxmax()) print("Q6 - Lowest std:", df.std().idxmin()

Q7: Correlation between Glucose and BMI

sns.scatterplot(x='Glucose', y='BMI', hue='Outcome', data=df) plt.title('Glucose vs BMI') p

Q8: Average Age of diabetic vs. non-diabetic patients

print("Q8:\n", df.groupby('Outcome')['Age'].mean())

Q9: Average Insulin levels for patients with and without diabetes

print("Q9:\n", df.groupby('Outcome')['Insulin'].mean())

Q10: Boxplot of BMI grouped by Outcome

sns.boxplot(x='Outcome', y='BMI', data=df) plt.title('BMI by Outcome') plt.show()

Q11: Heatmap of feature correlations

corr = df.corr() sns.heatmap(corr, annot=True, cmap='coolwarm') plt.title('Correlation Heat

Q12: Pairplot for Glucose, Insulin, BMI

sns.pairplot(df[['Glucose', 'Insulin', 'BMI', 'Outcome']], hue='Outcome') plt.show()
```

```
Q13: Boxplot to identify outliers

for col in ['SkinThickness', 'Insulin']: sns.boxplot(y=col, data=df) plt.title(f'Boxplot of

Q14: Mean Glucose level in AgeGroup bins

bins = [0, 30, 40, 50, 60, df['Age'].max()] labels = ['<30', '30-40', '40-50', '50-60', '60

Q15: BMI Category with highest proportion of diabetic patients

def bmi_category(bmi): if bmi < 18.5: return 'Underweight' elif bmi < 25: return 'Normal' e

Q16: Percentage of people over 50 who are diabetic

print("Q16:\n", (df[df['Age'] > 50]['Outcome'].mean() * 100).round(2), "%")

Q17: Pivot table of avg BMI and Glucose for AgeGroup and Outcome

print("Q17:\n", df.pivot_table(values=['BMI', 'Glucose'], index='AgeGroup', columns='Outcom

Q18: Pairplot to separate diabetic and non-diabetic

sns.pairplot(df, vars=['Glucose', 'BMI', 'Age', 'Insulin'], hue='Outcome') plt.show()

Q19: Variable showing strongest trend with Outcome

sns.boxplot(x='Outcome', y='Glucose', data=df) plt.title('Glucose vs Outcome') plt.show()

Q20: Multicollinearity check using correlation matrix

sns.heatmap(df.corr(), annot=True, cmap='coolwarm') plt.title('Multicollinearity - Correlat
```