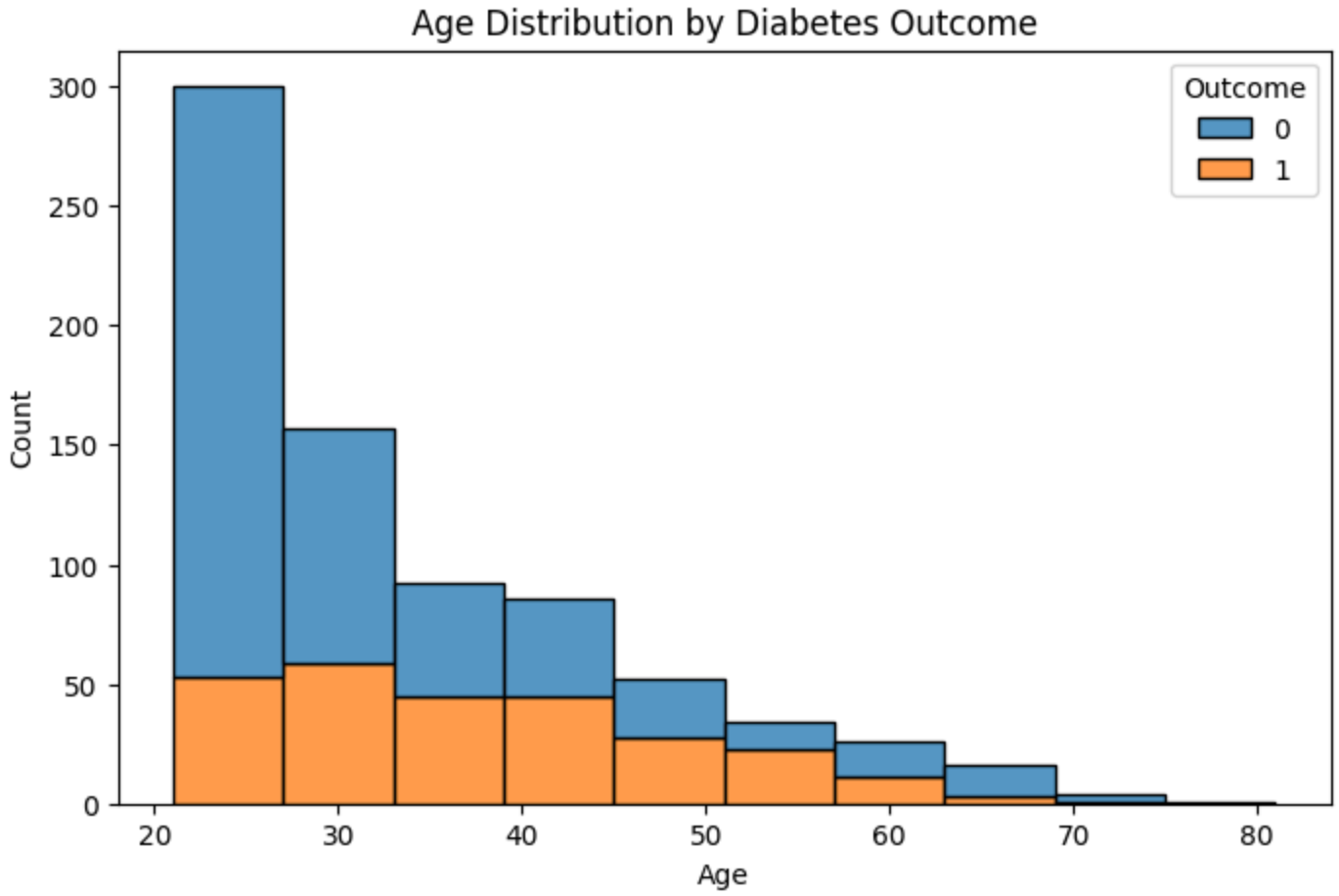
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv("diabetes.csv")

# Display basic dataset information
print("Dataset Summary:")
print(data.info())

# Handle missing or invalid values (replace zeros with the median of each column)
features_to_adjust = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']
for feature in features_to_adjust:
    median_value = data.loc[data[feature] != 0, feature].median()
    data[feature].replace(0, median_value, inplace=True)

# General statistics
print("\nStatistical Overview:")
print(data.describe())

# Filter data for individuals with high glucose and obesity
high_glucose_bmi = data[(data['Glucose'] > 140) & (data['BMI'] > 30)]
print(f"\nNumber of individuals with high glucose and obesity: {len(high_glucose_bmi)}")

# Categorize individuals based on age groups
age_bins = [20, 30, 40, 50, 60, 100]
age_labels = ["20-30", "30-40", "40-50", "50-60", "60+"]
data['AgeCategory'] = pd.cut(data['Age'], bins=age_bins, labels=age_labels)

# Calculate average and median values for each age group
age_analysis = data.groupby('AgeCategory').agg(['mean', 'median'])
print("\nAge Group Analysis:")
print(age_analysis)

# Sort data based on glucose levels and retrieve the top 10
top_glucose = data.nlargest(10, 'Glucose')[['Glucose', 'Age', 'BMI']]
print("\nTop 10 Individuals with Highest Glucose Levels:")
```

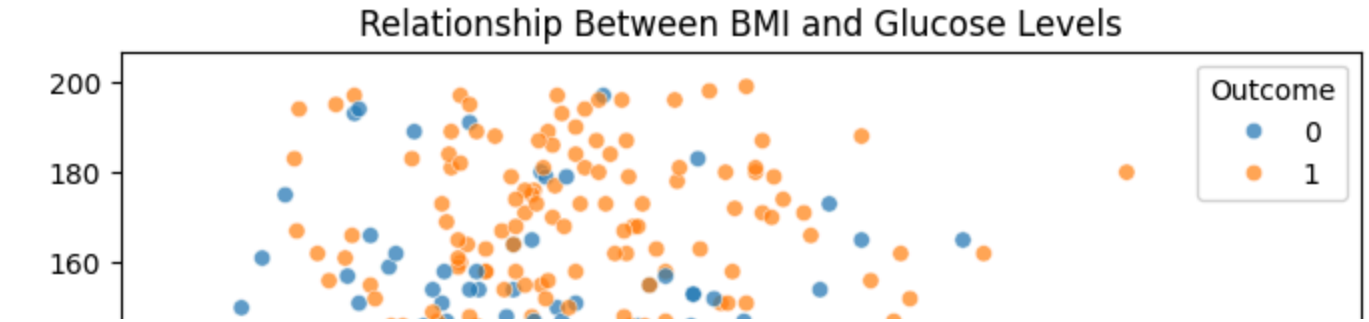
```
print("\nTop 10 individuals with highest Glucose Levels:")
print(top_glucose)

# Visualize diabetes outcome distribution by age
plt.figure(figsize=(8, 5))
sns.histplot(data=data, x='Age', hue='Outcome', multiple='stack', bins=10)
plt.title("Age Distribution by Diabetes Outcome")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()

# Scatter plot for BMI vs. glucose levels
plt.figure(figsize=(8, 5))
sns.scatterplot(data=data, x='BMI', y='Glucose', hue='Outcome', alpha=0.7)
plt.title("Relationship Between BMI and Glucose Levels")
plt.xlabel("BMI")
plt.ylabel("Glucose")
plt.show()

print("\nData analysis completed successfully!")
```

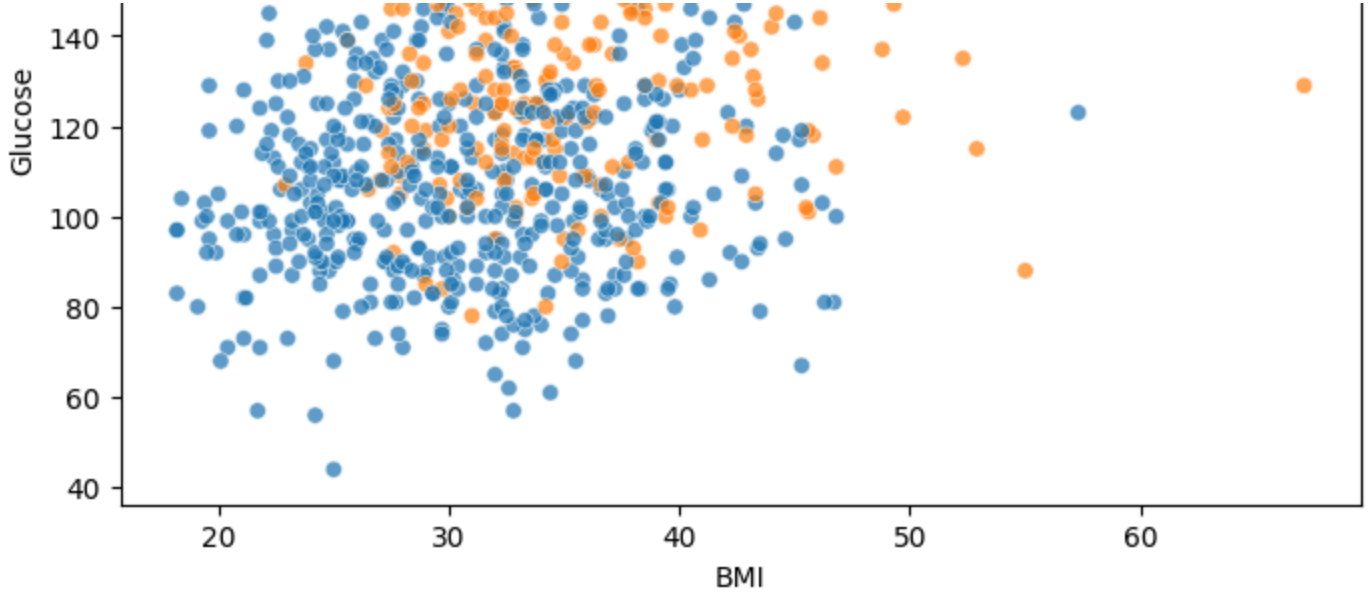
```
Dataset Summary:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None

Statistical Overview:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  121.656250      72.386719      29.108073  140.671875
std       3.369578   30.438286      12.096642       8.791221   86.383060
min       0.000000   44.000000      24.000000       7.000000   14.000000
25%       1.000000   99.750000      64.000000      25.000000  121.500000
50%       3.000000  117.000000      72.000000      29.000000  125.000000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    32.455208                  0.471876   33.240885    0.348958
std      6.875177                  0.331329   11.760232    0.476951
min     18.200000                  0.078000   21.000000    0.000000
25%     27.500000                  0.243750   24.000000    0.000000
50%     32.300000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000

Number of individuals with high glucose and obesity: 148
```

```
Age Group Analysis:
            Pregnancies            Glucose        BloodPressure         \
                   mean median        mean median          mean median
AgeCategory
20-30          2.007194    2.0  115.016787  111.0     69.122302   70.0
30-40          5.273885    5.0  126.923567  123.0     73.942675   74.0
40-50          7.123894    7.0  125.920354  125.0     77.336283   76.0
50-60          6.518519    7.0  141.148148  138.0     80.055556   78.0
60+            4.851852    5.0  136.740741  136.0     77.703704   78.0

            SkinThickness            Insulin               BMI         \
                     mean median        mean median       mean median
AgeCategory
20-30           28.055156   29.0  134.366906  125.0  32.019185   31.6
30-40           30.515924   29.0  142.789809  125.0  32.886624   32.3
40-50           30.424779   29.0  138.814159  125.0  34.501770   33.8
50-60           29.148148   29.0  193.907407  125.0  31.712963   32.8
60+             31.592593   29.0  127.037037  125.0  29.600000   28.8

            DiabetesPedigreeFunction              Age          Outcome
                                mean median       mean median      mean median
AgeCategory
20-30                       0.446827  0.364  24.599520   24.0  0.215827    0.0
30-40                       0.536994  0.432  35.146497   35.0  0.484076    0.0
40-50                       0.452593  0.341  44.327434   44.0  0.566372    1.0
50-60                       0.529111  0.476  54.981481   54.5  0.574074    1.0
60+                         0.446333  0.419  65.740741   65.0  0.259259    0.0

Top 10 Individuals with Highest Glucose Levels:
     Glucose  Age   BMI
661      199   22  42.9
561      198   28  41.3
8        197   53  30.5
228      197   31  36.7
408      197   39  25.9
579      197   62  34.7
22       196   41  39.8
206      196   57  37.5
359      196   29  36.5
498      195   55  25.1
<ipython-input-2-1b19f9567982>:16: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series throug
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we
```

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

  data[feature].replace(0, median_value, inplace=True)
<ipython-input-2-1b19f9567982>:32: FutureWarning: The default of observed=False is deprecated and will be changed to Tr
  age_analysis = data.groupby('AgeCategory').agg(['mean', 'median'])
```








```
Data analysis completed successfully!
```