DiabetesAnalysis

March 23, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

[2]:
```
# Load dataset
df = pd.read_csv("diabetes.csv")
```

[3]:
```
# Display basic info and first few rows
print("Dataset Info:")
print(df.info())
print("\nFirst 5 Rows:")
print(df.head())
```

```
Dataset Info:
<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

First 5 Rows:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
```

```
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

[4]:
```
# Descriptive statistics
desc_stats = df.describe().T
print("\nDescriptive Statistics:")
print(desc_stats)
```

```
Descriptive Statistics:
                          count        mean         std     min       25%  \
Pregnancies               768.0    3.845052    3.369578   0.000   1.00000
Glucose                   768.0  120.894531   31.972618   0.000  99.00000
BloodPressure             768.0   69.105469   19.355807   0.000  62.00000
SkinThickness             768.0   20.536458   15.952218   0.000   0.00000
Insulin                   768.0   79.799479  115.244002   0.000   0.00000
BMI                       768.0   31.992578    7.884160   0.000  27.30000
DiabetesPedigreeFunction  768.0    0.471876    0.331329   0.078   0.24375
Age                       768.0   33.240885   11.760232  21.000  24.00000
Outcome                   768.0    0.348958    0.476951   0.000   0.00000

                             50%        75%     max
Pregnancies                3.0000    6.00000   17.00
Glucose                  117.0000  140.25000  199.00
BloodPressure             72.0000   80.00000  122.00
SkinThickness             23.0000   32.00000   99.00
Insulin                   30.5000  127.25000  846.00
BMI                       32.0000   36.60000   67.10
DiabetesPedigreeFunction   0.3725    0.62625    2.42
Age                       29.0000   41.00000   81.00
Outcome                    0.0000    1.00000    1.00
```

[5]:
```
# Calculate mode for each column
modes = df.mode().iloc[0]
print("\nMode for each column:")
print(modes)
```

```
Mode for each column:
Pregnancies                 1.000
Glucose                    99.000
```

```
BloodPressure               70.000
SkinThickness                0.000
Insulin                      0.000
BMI                         32.000
DiabetesPedigreeFunction     0.254
Age                         22.000
Outcome                      0.000
Name: 0, dtype: float64
```

[6]:
```
# Calculate skewness and kurtosis for each column
skewness = df.skew()
kurtosis = df.kurtosis()
print("\nSkewness of each column:")
print(skewness)
print("\nKurtosis of each column:")
print(kurtosis)
```

```
Skewness of each column:
Pregnancies                 0.901674
Glucose                     0.173754
BloodPressure              -1.843608
SkinThickness               0.109372
Insulin                     2.272251
BMI                        -0.428982
DiabetesPedigreeFunction    1.919911
Age                         1.129597
Outcome                     0.635017
dtype: float64

Kurtosis of each column:
Pregnancies                 0.159220
Glucose                     0.640780
BloodPressure               5.180157
SkinThickness              -0.520072
Insulin                     7.214260
BMI                         3.290443
DiabetesPedigreeFunction    5.594954
Age                         0.643159
Outcome                    -1.600930
dtype: float64
```

[7]:
```
# Correlation matrix
corr_matrix = df.corr()
print("\nCorrelation Matrix:")
print(corr_matrix)
```

```
Correlation Matrix:
```

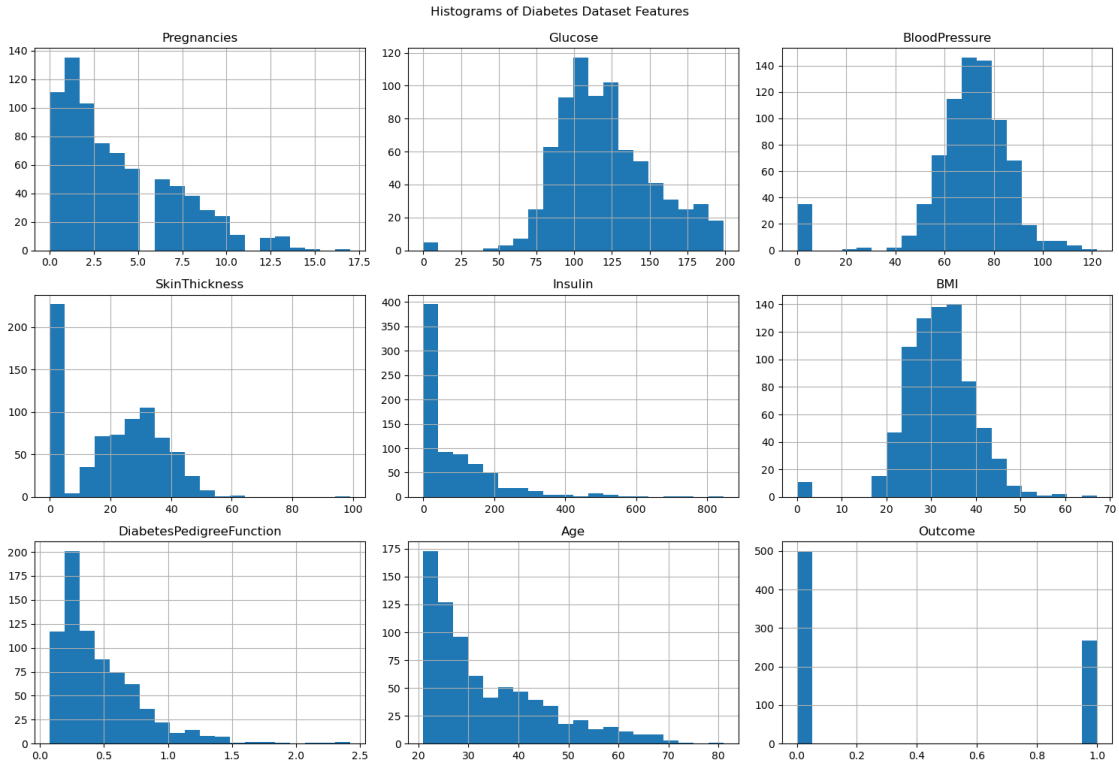
```
                           Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                   1.000000  0.129459       0.141282      -0.081672
Glucose                       0.129459  1.000000       0.152590       0.057328
BloodPressure                 0.141282  0.152590       1.000000       0.207371
SkinThickness                -0.081672  0.057328       0.207371       1.000000
Insulin                      -0.073535  0.331357       0.088933       0.436783
BMI                           0.017683  0.221071       0.281805       0.392573
DiabetesPedigreeFunction     -0.033523  0.137337       0.041265       0.183928
Age                           0.544341  0.263514       0.239528      -0.113970
Outcome                       0.221898  0.466581       0.065068       0.074752

                            Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies               -0.073535  0.017683                 -0.033523
Glucose                    0.331357  0.221071                  0.137337
BloodPressure              0.088933  0.281805                  0.041265
SkinThickness              0.436783  0.392573                  0.183928
Insulin                    1.000000  0.197859                  0.185071
BMI                        0.197859  1.000000                  0.140647
DiabetesPedigreeFunction   0.185071  0.140647                  1.000000
Age                       -0.042163  0.036242                  0.033561
Outcome                    0.130548  0.292695                  0.173844

                                Age   Outcome
Pregnancies                0.544341  0.221898
Glucose                    0.263514  0.466581
BloodPressure              0.239528  0.065068
SkinThickness             -0.113970  0.074752
Insulin                   -0.042163  0.130548
BMI                        0.036242  0.292695
DiabetesPedigreeFunction   0.033561  0.173844
Age                        1.000000  0.238356
Outcome                    0.238356  1.000000
```

[8]:
```python
# Hypothesis Testing: t-test for Glucose levels between Outcome groups
group0 = df[df['Outcome'] == 0]['Glucose']
group1 = df[df['Outcome'] == 1]['Glucose']
t_stat, p_val = stats.ttest_ind(group0, group1, nan_policy='omit')
print("\nT-test for Glucose levels between Outcome groups:")
print(f"t-statistic: {t_stat:.4f}, p-value: {p_val:.4f}")
```

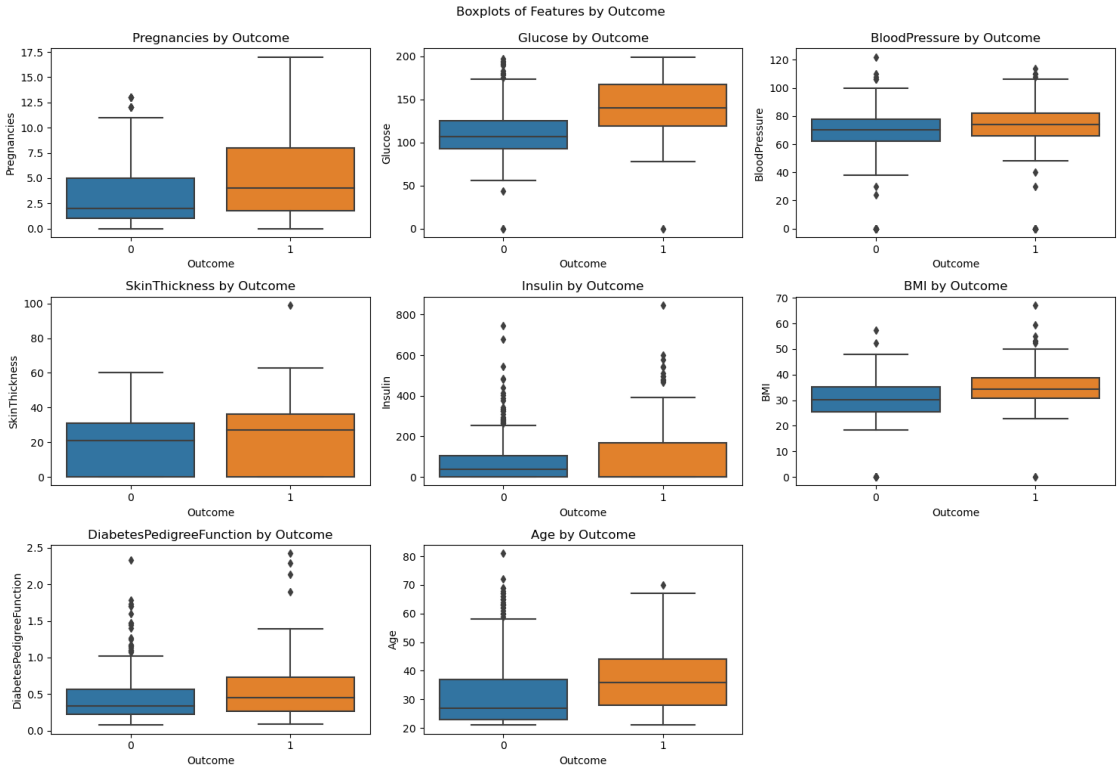
```
T-test for Glucose levels between Outcome groups:
t-statistic: -14.6001, p-value: 0.0000
```

[9]:
```python
# Visualization 1: Histograms for each variable
df.hist(bins=20, figsize=(15, 10))
plt.tight_layout()
plt.suptitle("Histograms of Diabetes Dataset Features", y=1.02)
```

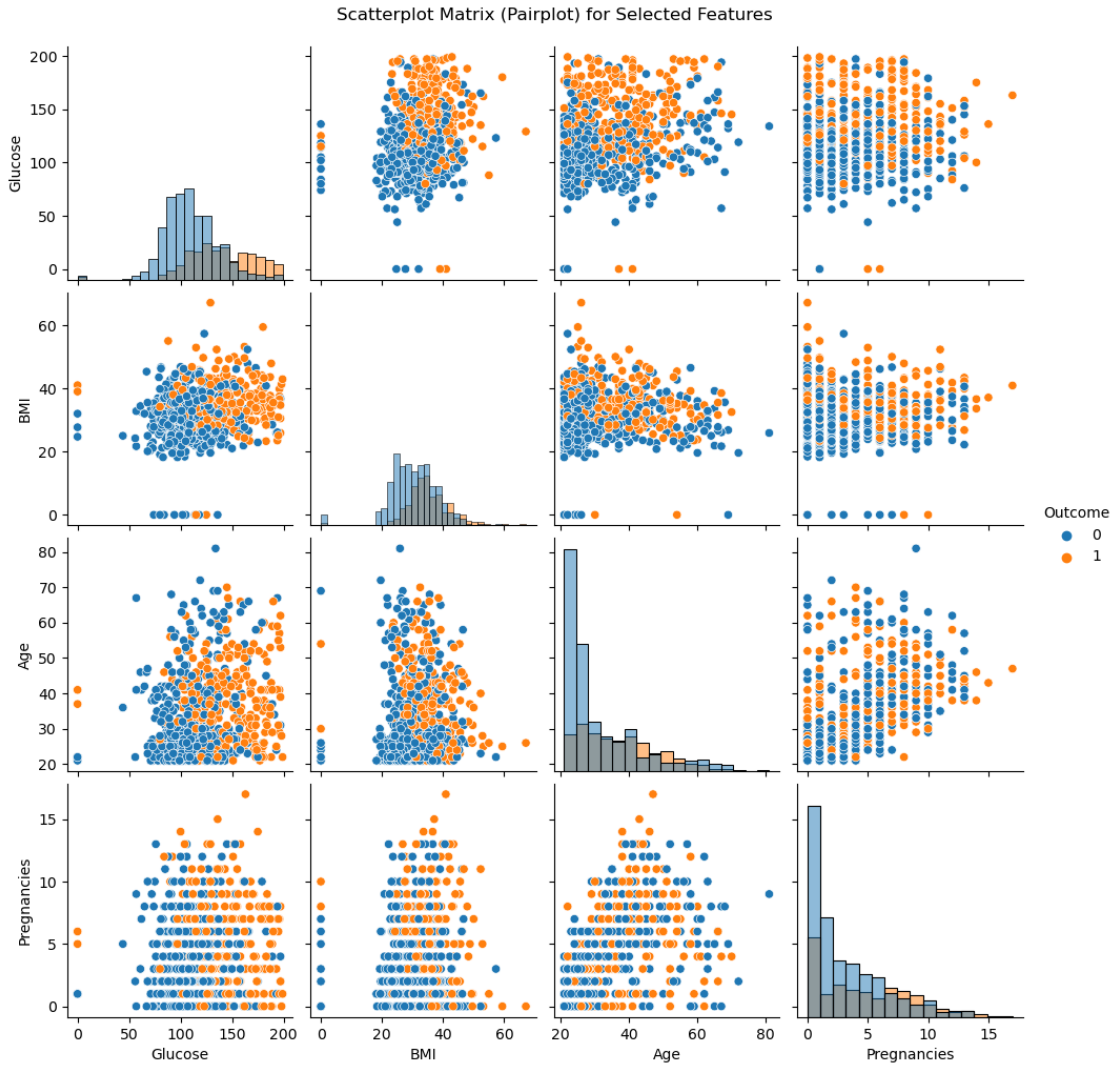
```
plt.show()
```



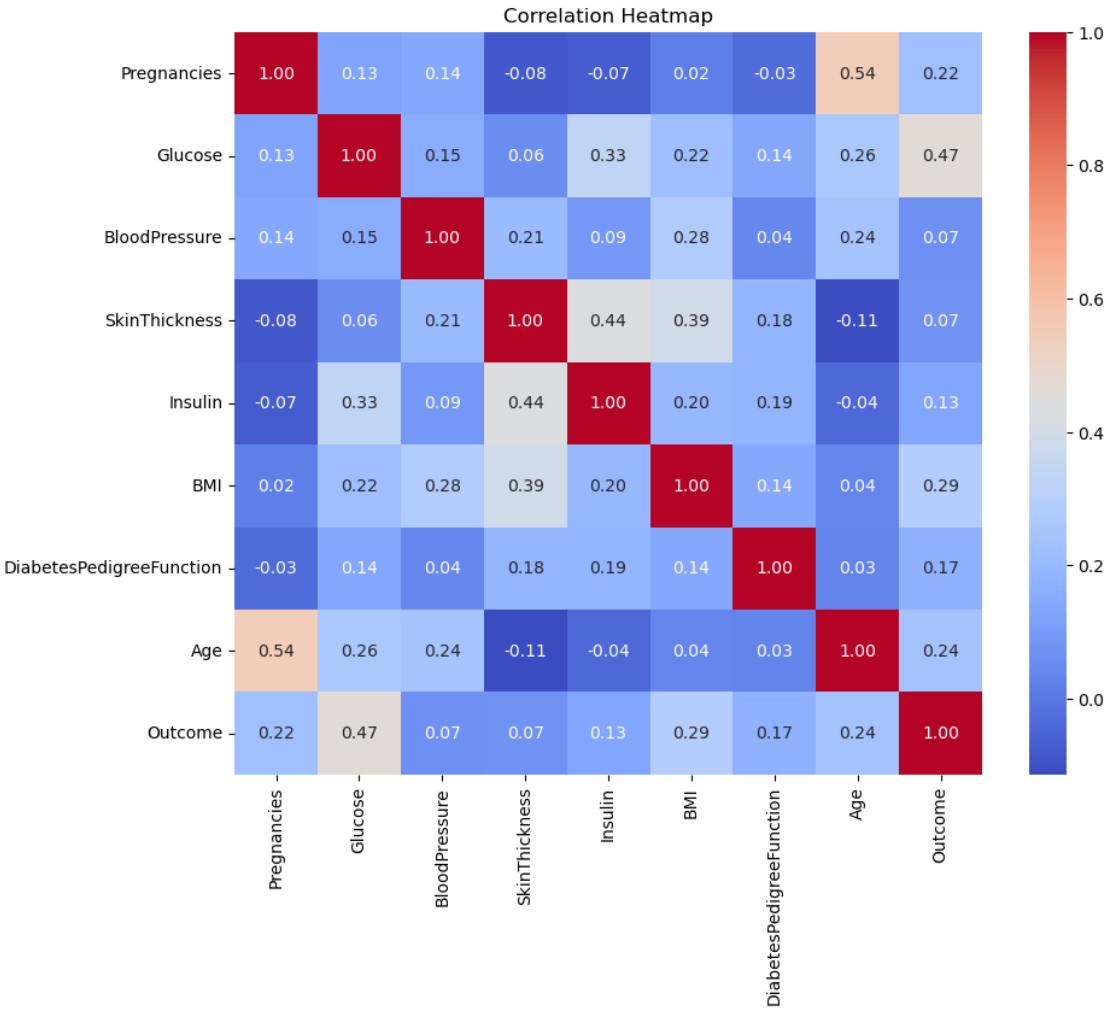
[10]:
```
# Visualization 2: Boxplots for each feature by Outcome
plt.figure(figsize=(15, 10))
for idx, col in enumerate(df.columns[:-1], 1):  # Excluding Outcome
    plt.subplot(3, 3, idx)
    sns.boxplot(x='Outcome', y=col, data=df)
    plt.title(f"{col} by Outcome")
plt.tight_layout()
plt.suptitle("Boxplots of Features by Outcome", y=1.02)
plt.show()
```



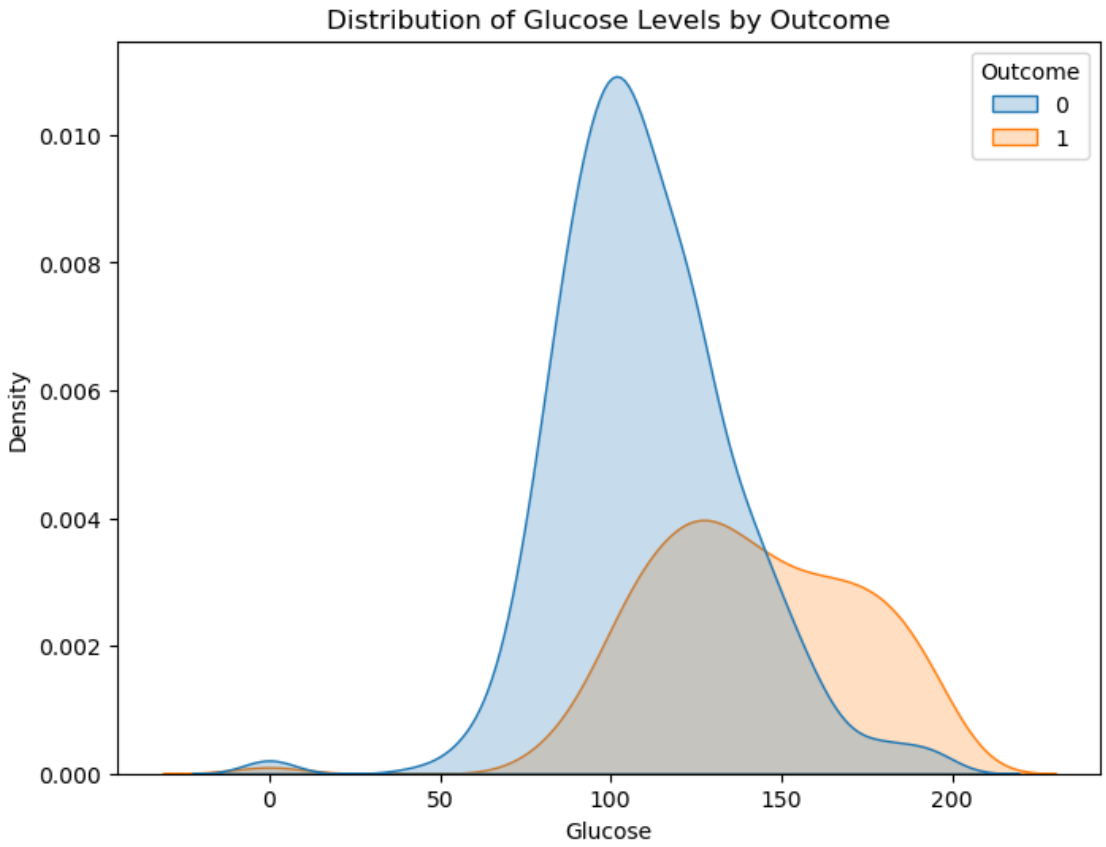
[11]:
```
# Visualization 3: Scatterplot Matrix (Pairplot) for a subset of features
# (Using a subset to avoid memory issues)
features_subset = ["Glucose", "BMI", "Age", "Pregnancies"]
sns.pairplot(df[features_subset + ["Outcome"]], hue="Outcome", diag_kind="hist")
plt.suptitle("Scatterplot Matrix (Pairplot) for Selected Features", y=1.02)
plt.show()
```



[12]:
```
# Visualization 4: Correlation Heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



[13]:
```
# Visualization 5: KDE Plot for Glucose Distribution by Outcome
plt.figure(figsize=(8, 6))
sns.kdeplot(data=df, x="Glucose", hue="Outcome", fill=True)
plt.title("Distribution of Glucose Levels by Outcome")
plt.show()
```


Distribution of Glucose Levels by Outcome
Outcome
0
1
Density
0.010
0.008
0.006
0.004
0.002
0.000
0
50
100
150
200
Glucose

[ ]: