Exploratory Data Analysis on Diabetes Dataset

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

In [2]:
```
df=pd.read_csv("diabetes - diabetes.csv")
df
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFuncti
0	6	148	72	35	0	33.6	0.6
1	1	85	66	29	0	26.6	0.3
2	8	183	64	0	0	23.3	0.6
3	1	89	66	23	94	28.1	0.1
4	0	137	40	35	168	43.1	2.2
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.1
764	2	122	70	27	0	36.8	0.3
765	5	121	72	23	112	26.2	0.2
766	1	126	60	0	0	30.1	0.3
767	1	93	70	31	0	30.4	0.3

768 rows × 9 columns

In [3]:
```
df.shape
```

Out[3]:
```
(768, 9)
```

*The dataset contains 768 rows and 9 columns, representing medical data of female patients with diabetes test results.

In [4]:
```
df.head(10)
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
5	5	116	74	0	0	25.6	0.201
6	3	78	50	32	88	31.0	0.248
7	10	115	0	0	0	35.3	0.134
8	2	197	70	45	543	30.5	0.158
9	8	125	96	0	0	0.0	0.232

In [5]:
```
df.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
```

*No missing values detected. All numeric features.

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [7]:
```
df.duplicated().sum()
```

Out[7]:
```
0
```

- No missing or duplicate row

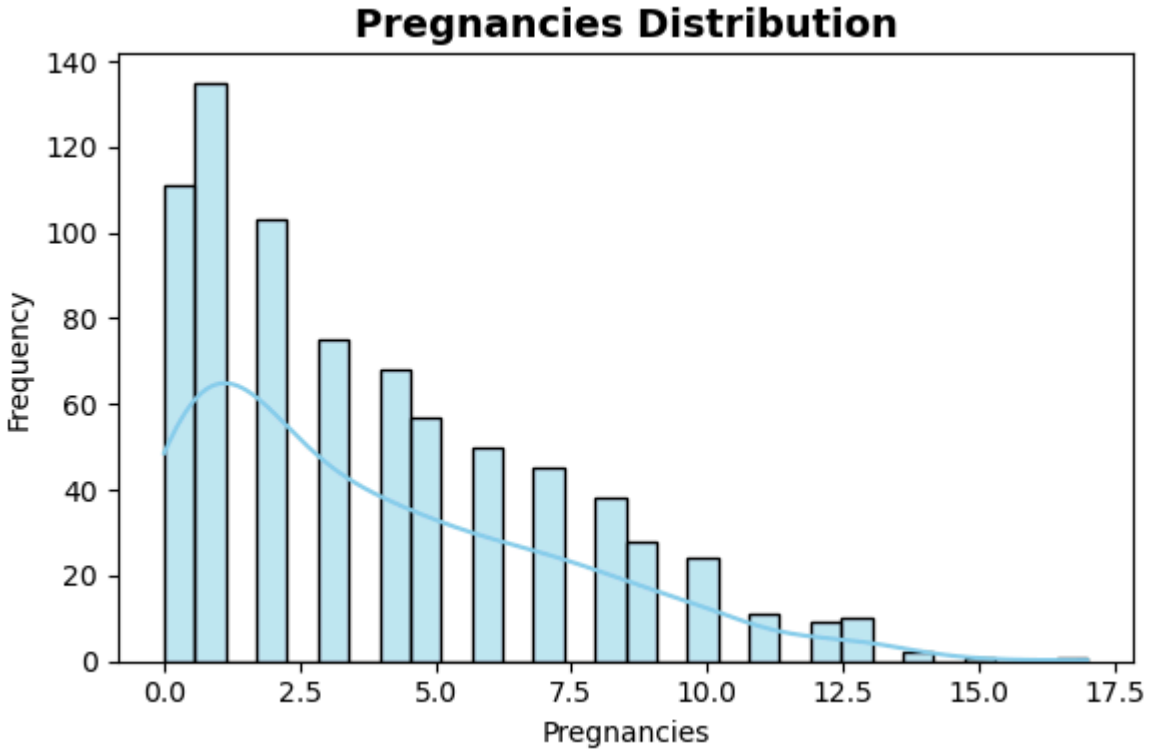
In [10]:
```
df.describe()
```

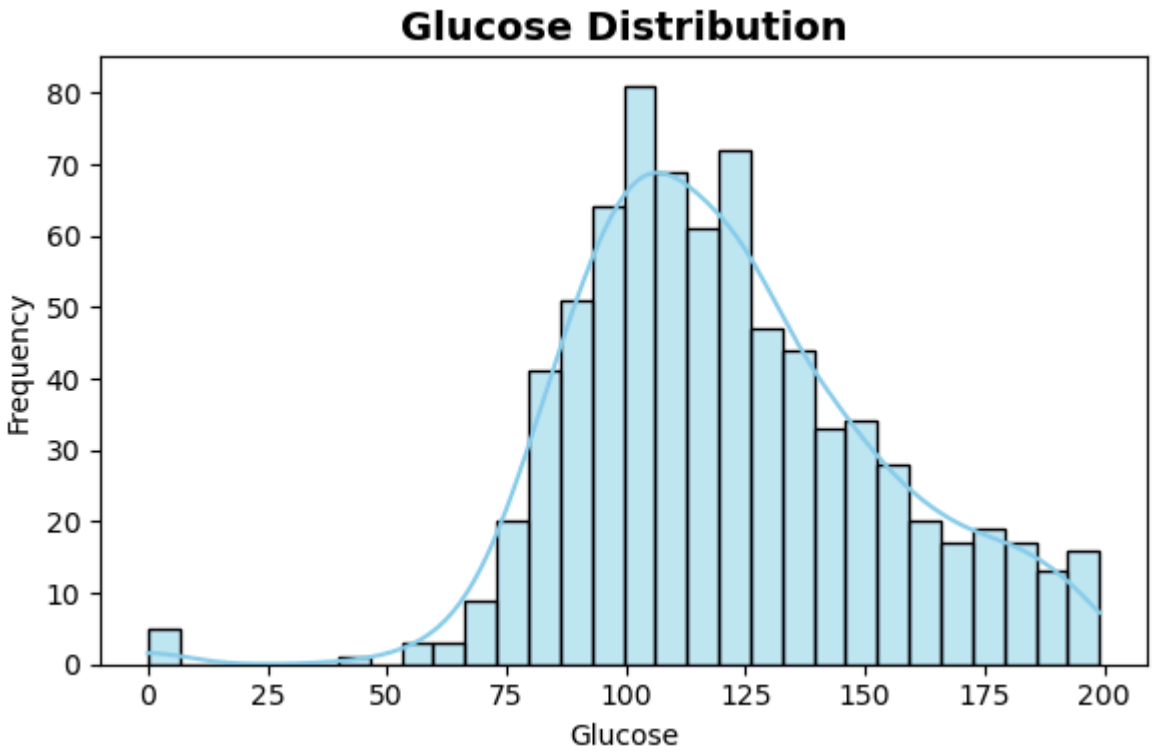
Out[10]:

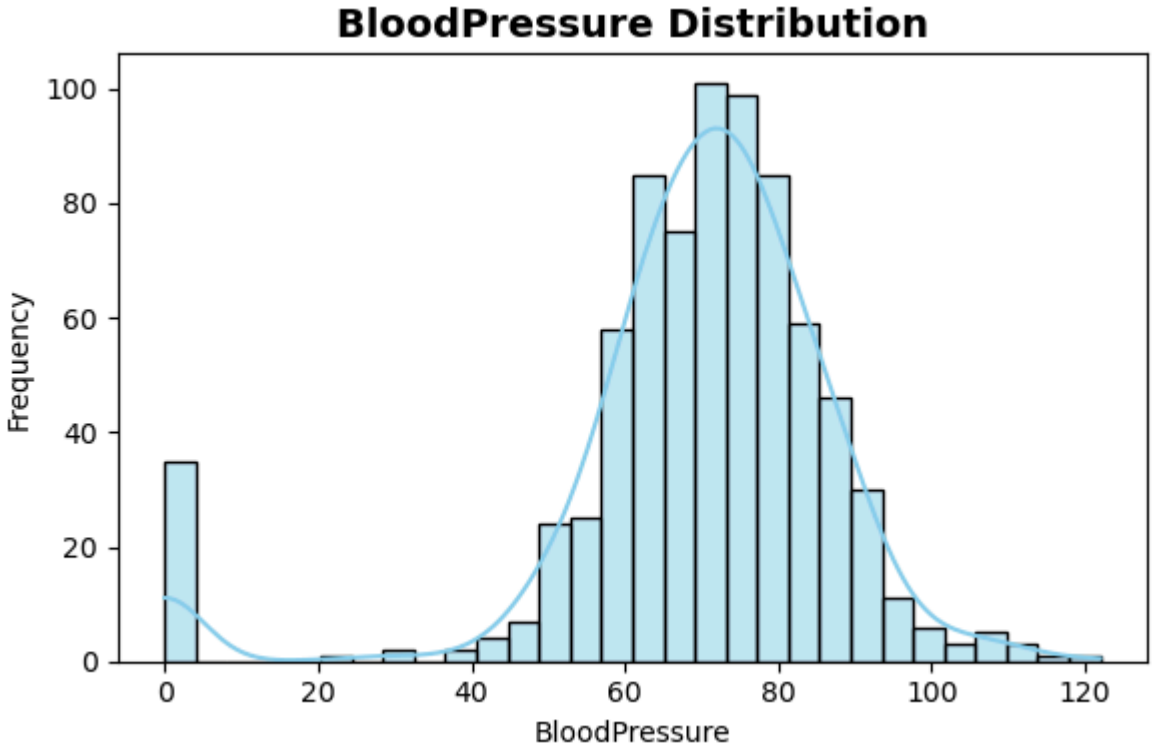
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabete
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

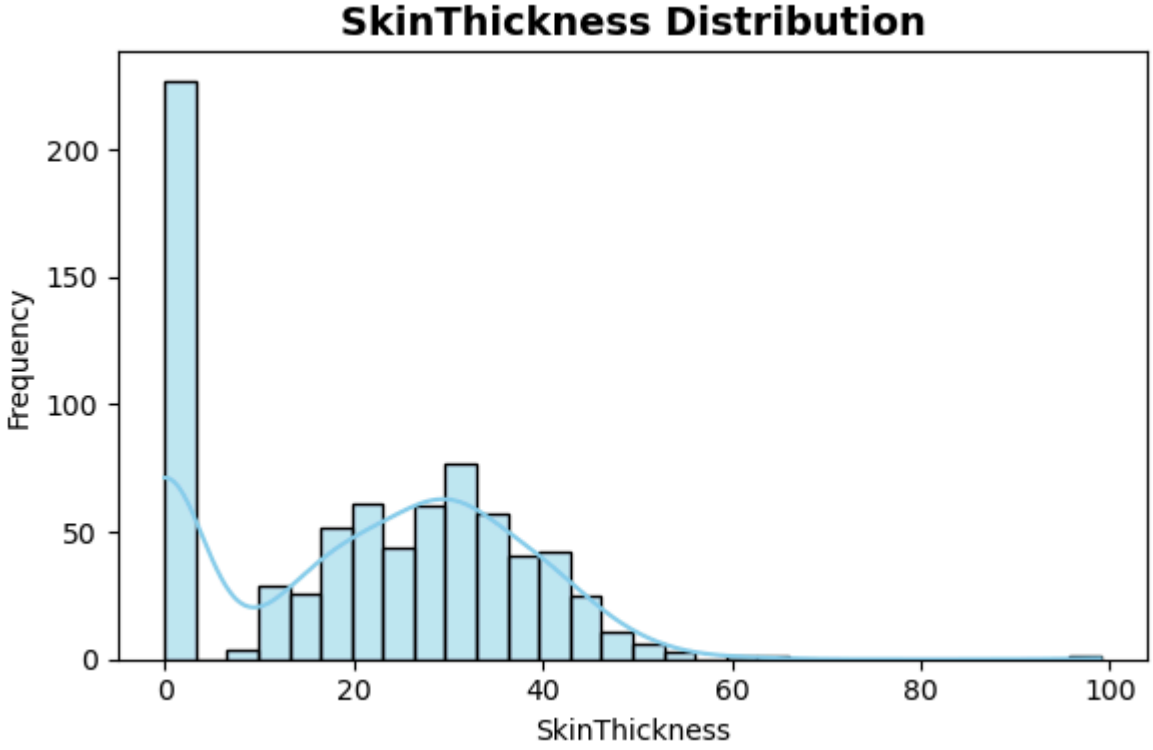
Univariate Analysis

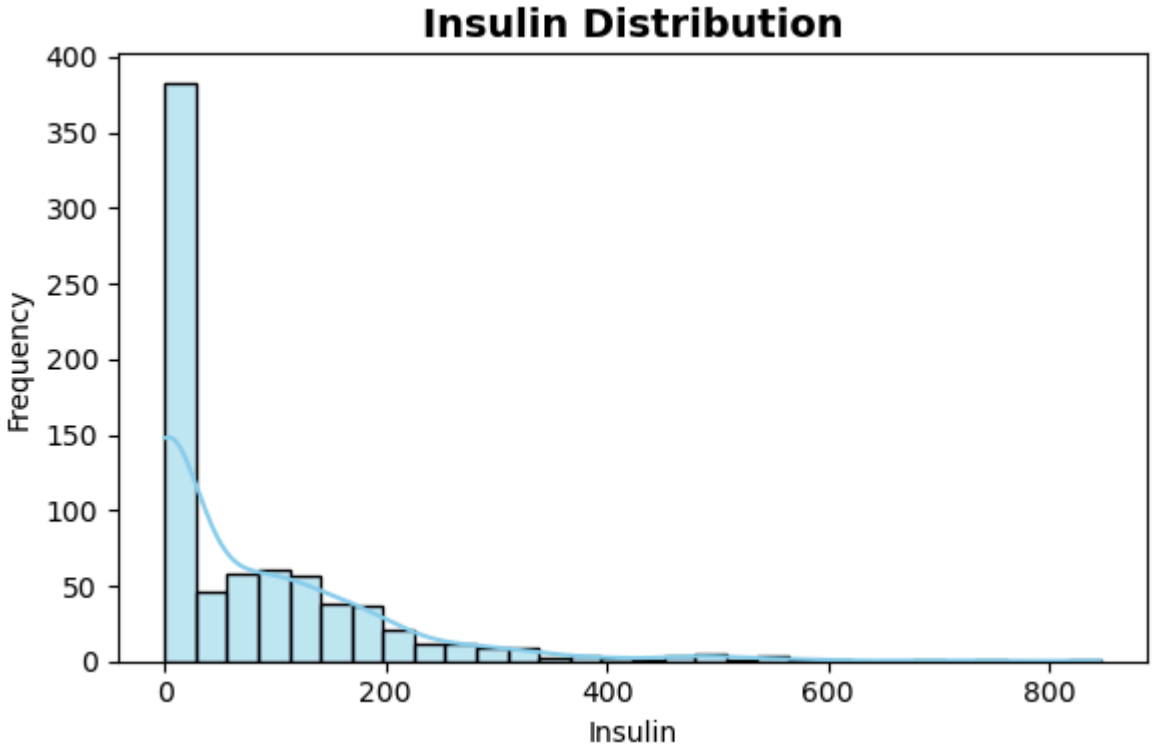
In [11]:
```
num_cols = df.columns[:-1]   # all columns except Outcome

for col in num_cols:
    plt.figure(figsize=(6,4))
    sns.histplot(df[col], bins=30, kde=True, color='skyblue', edgecolor='black
    plt.title(f'{col} Distribution', fontsize=14, fontweight='bold')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.tight_layout()
    plt.show()
```

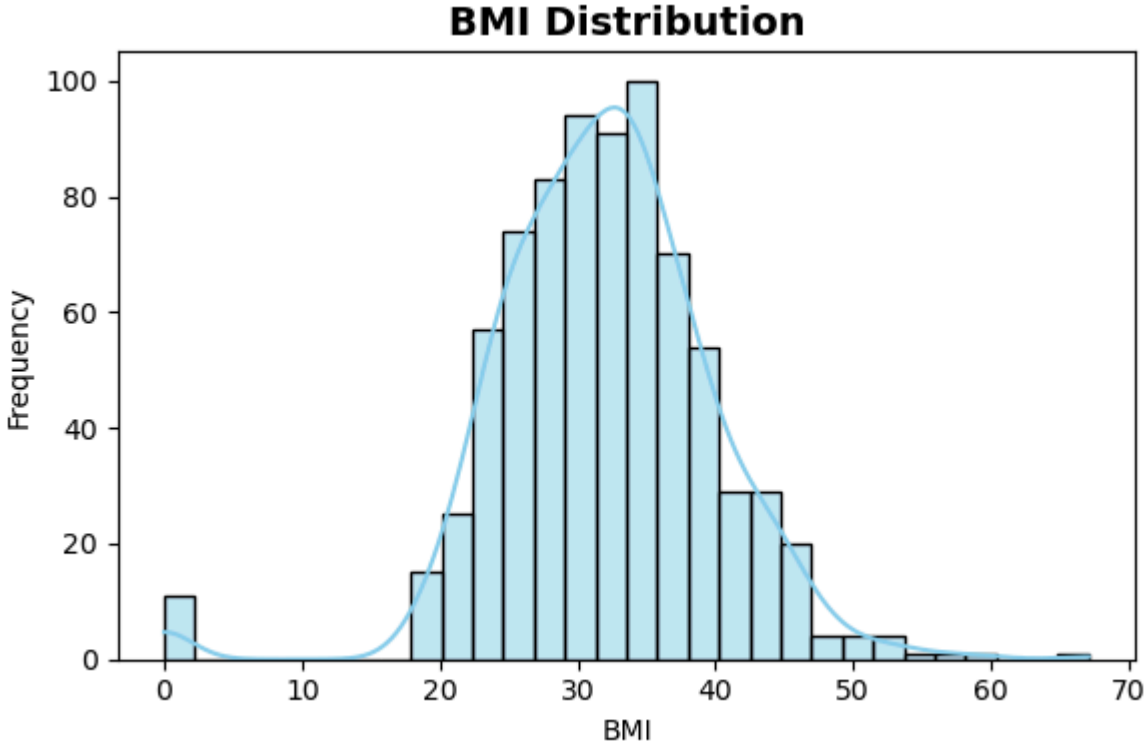


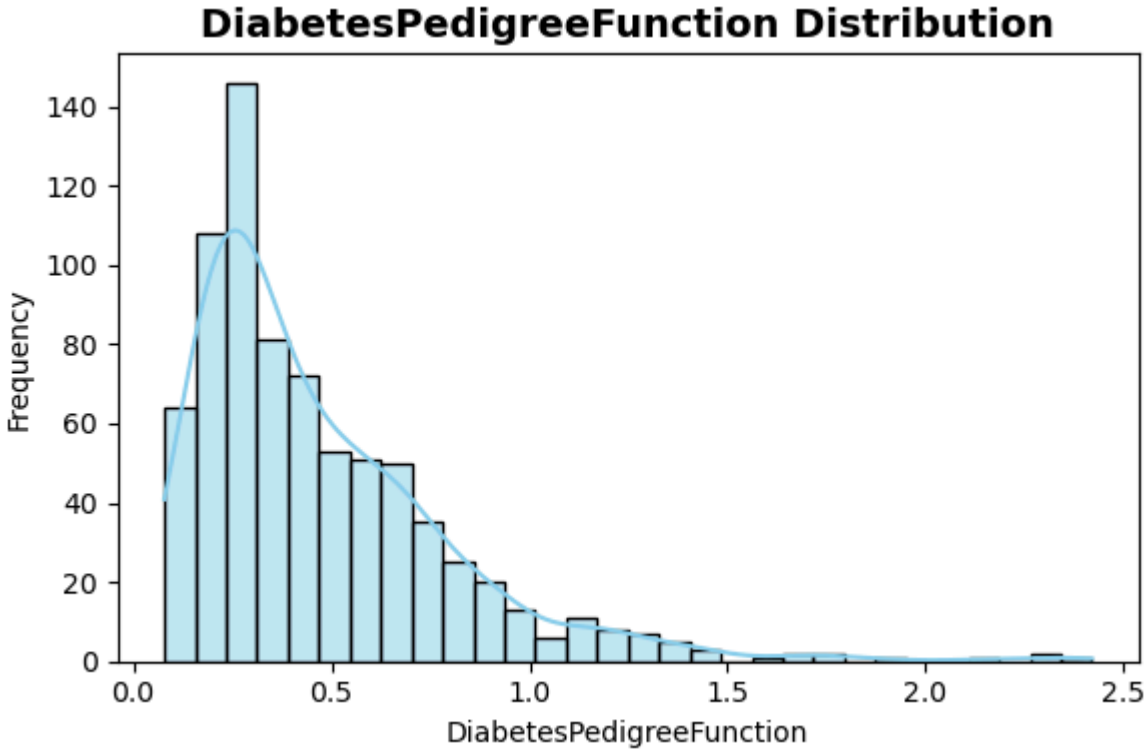


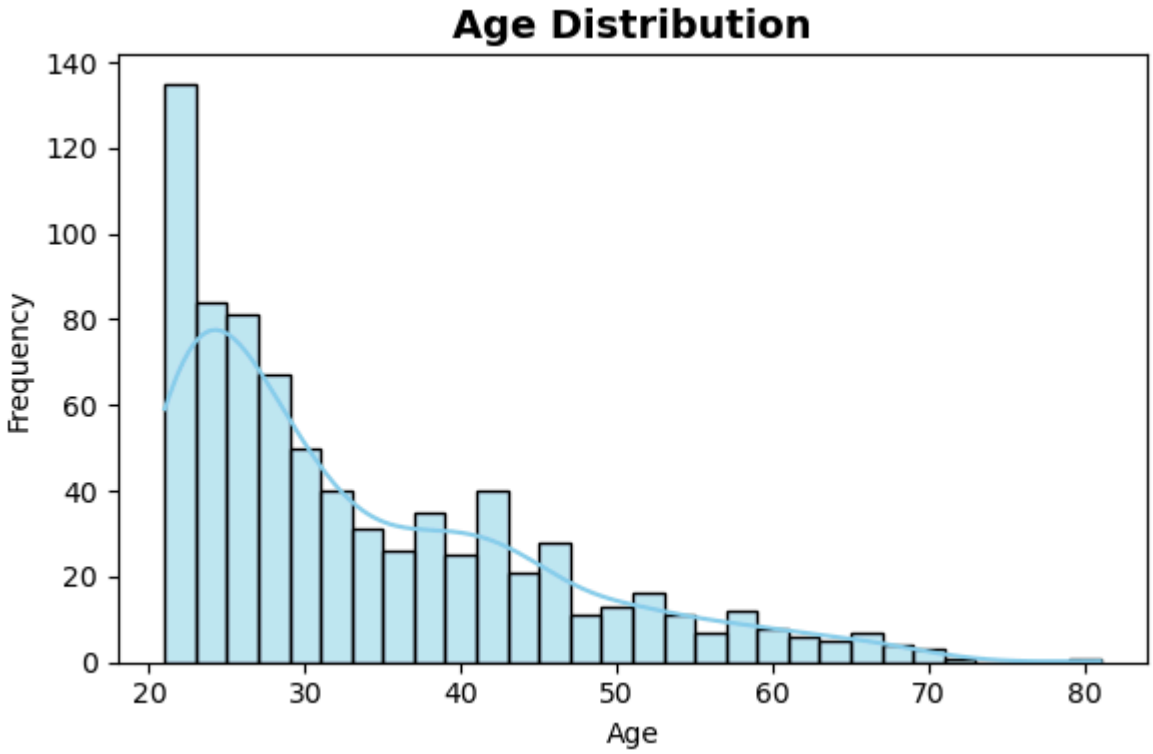

BloodPressure Distribution
Frequency
BloodPressure
0
20
40
60
80
100
120


SkinThickness Distribution
Frequency
SkinThickness
0
50
100
150
200
0
20
40
60
80
100


Insulin Distribution
Frequency
Insulin
0
50
100
150
200
250
300
350
400
0
200
400
600
800

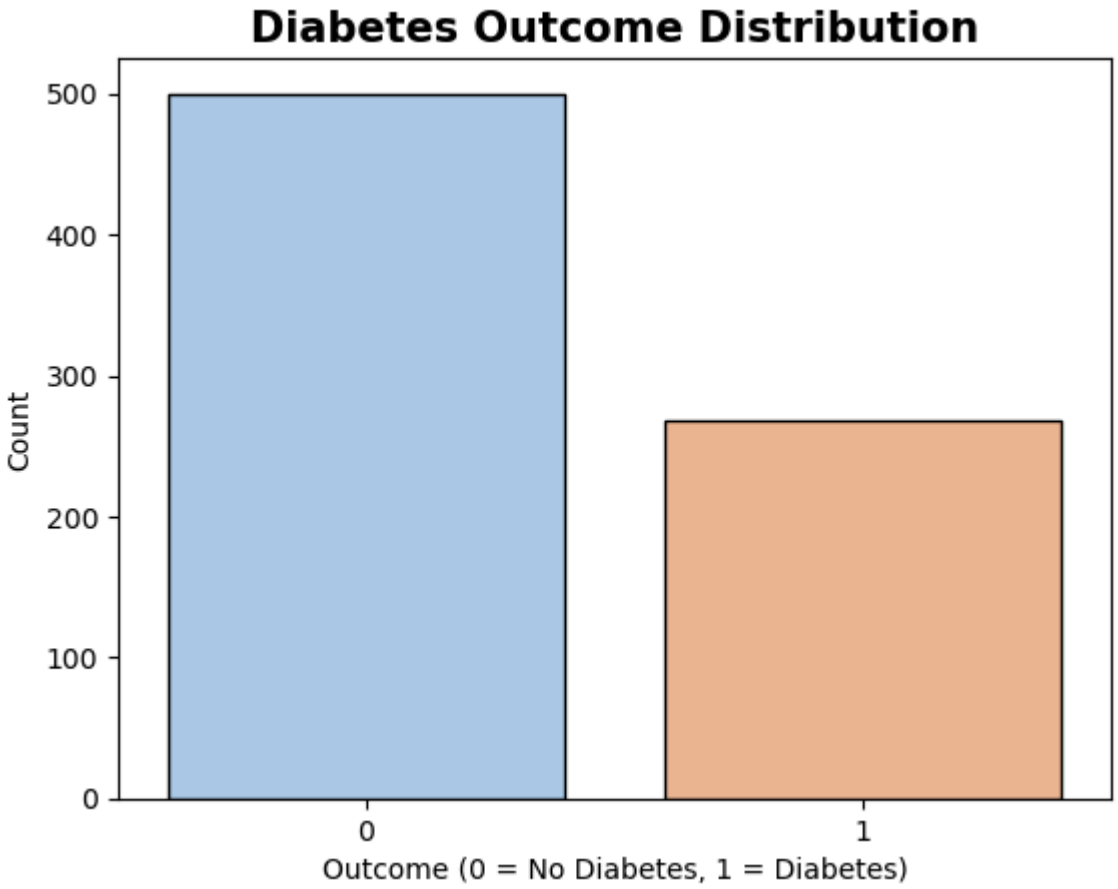

BMI Distribution
Frequency
BMI
0
20
40
60
80
100
0
10
20
30
40
50
60
70





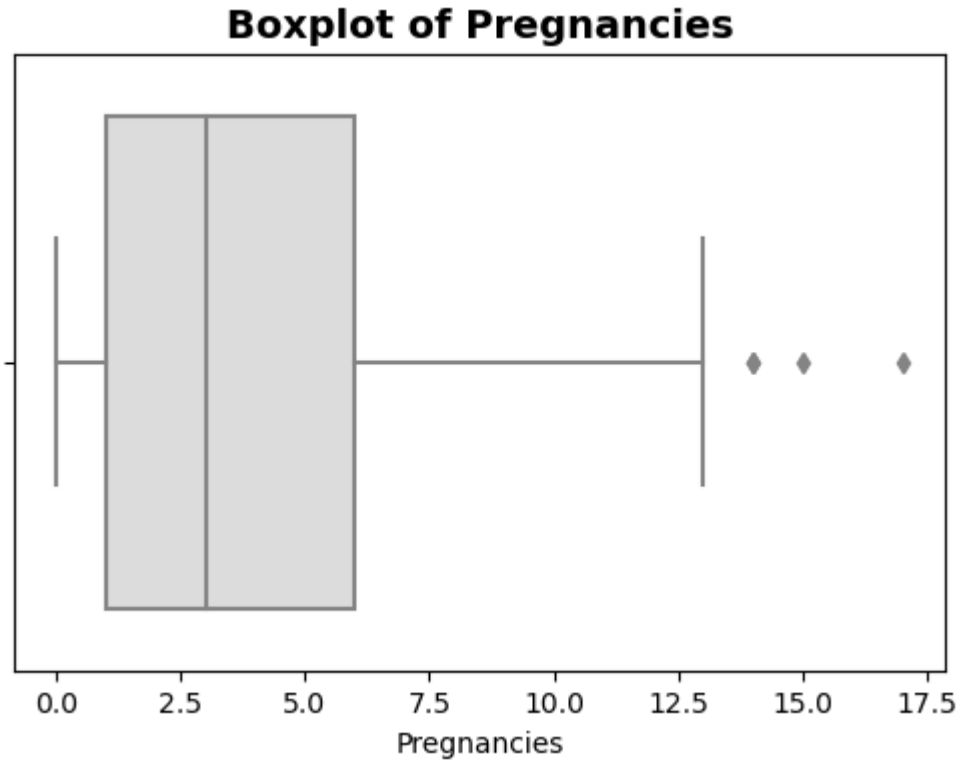
*Glucose, BMI, and Age are slightly right-skewed. Insulin has many 0s — heavy skew. Pregnancies mostly between 0–10.

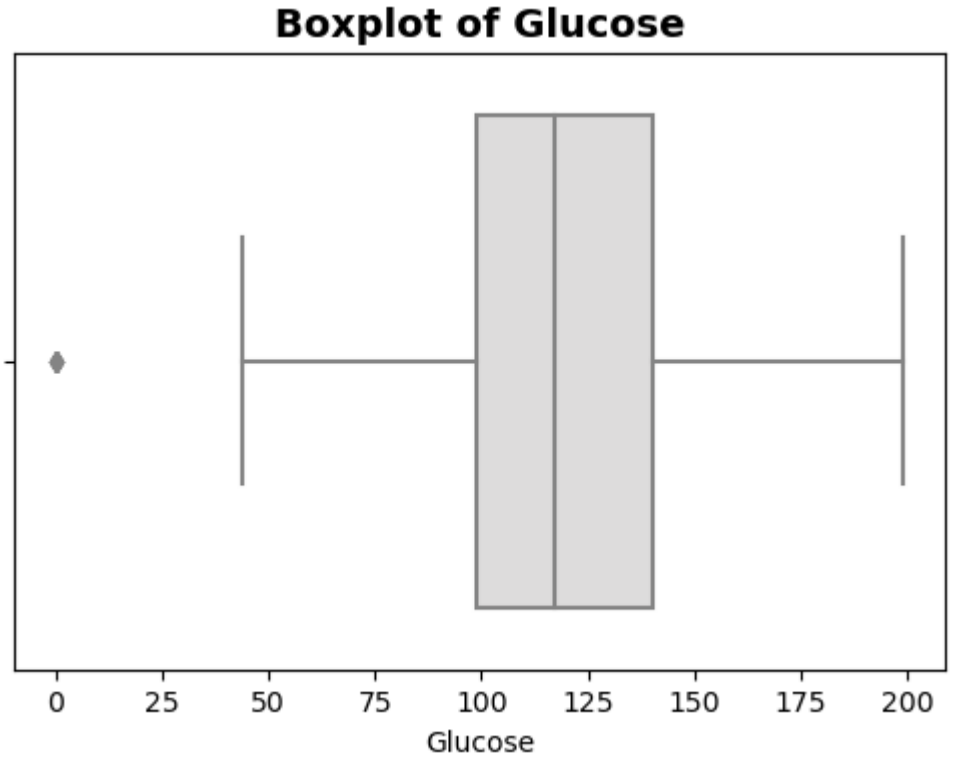
In [12]:
```
# Categorical variable outcomes
sns.countplot(data=df, x='Outcome', palette='pastel', edgecolor='black')
plt.title("Diabetes Outcome Distribution", fontsize=15, fontweight='bold')
plt.xlabel("Outcome (0 = No Diabetes, 1 = Diabetes)")
plt.ylabel("Count")
plt.show()
```

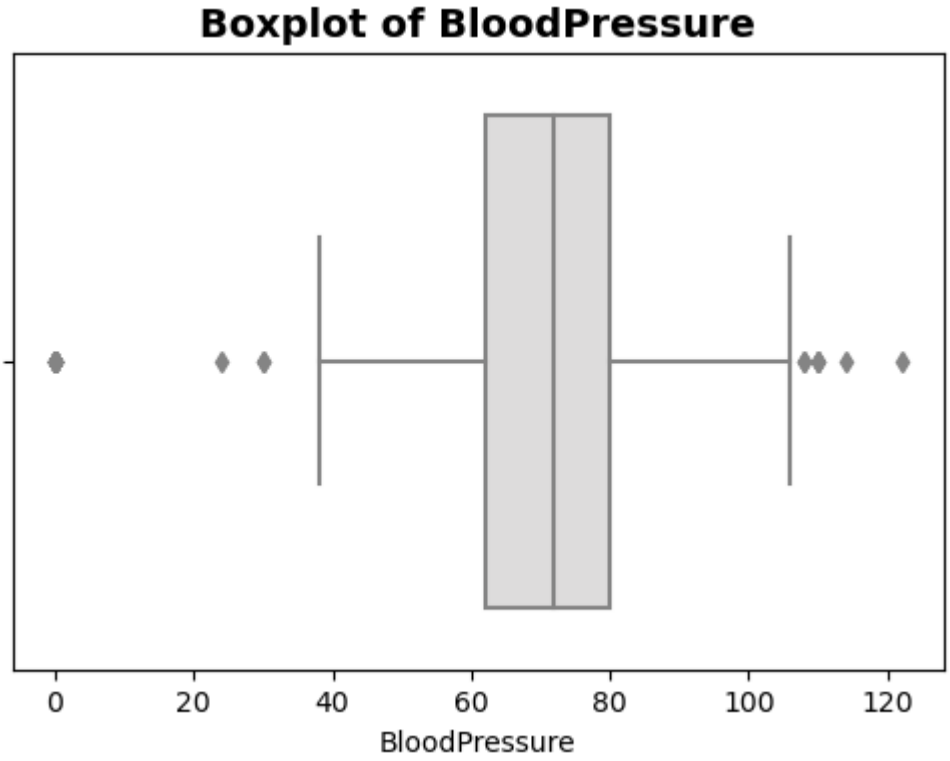


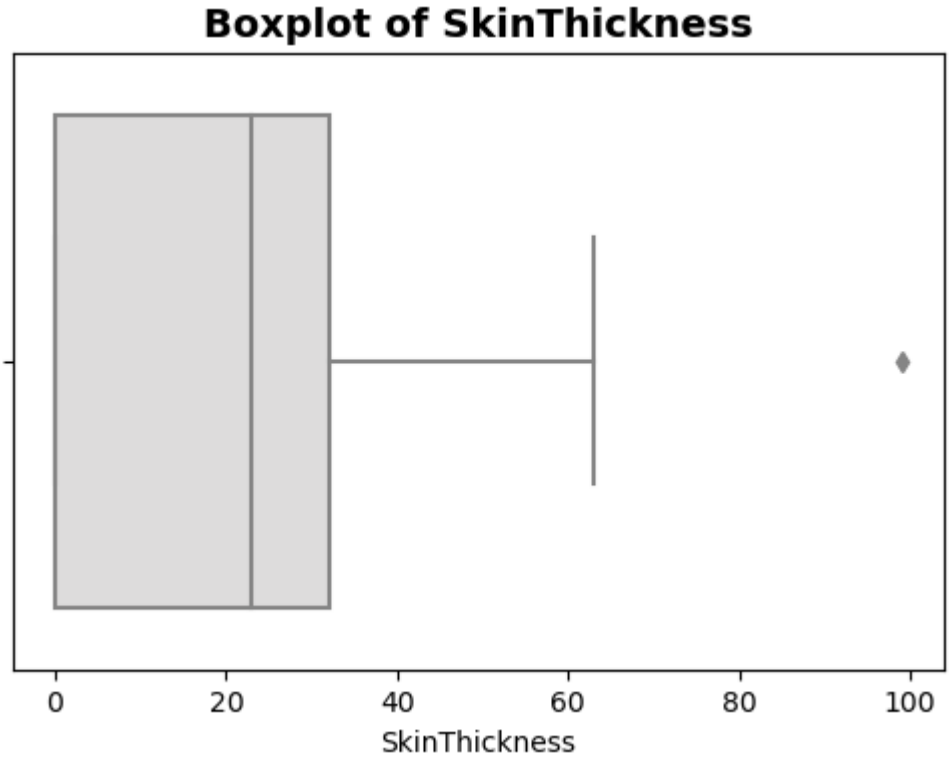
*Around 35% diabetic and 65% non-diabetic.

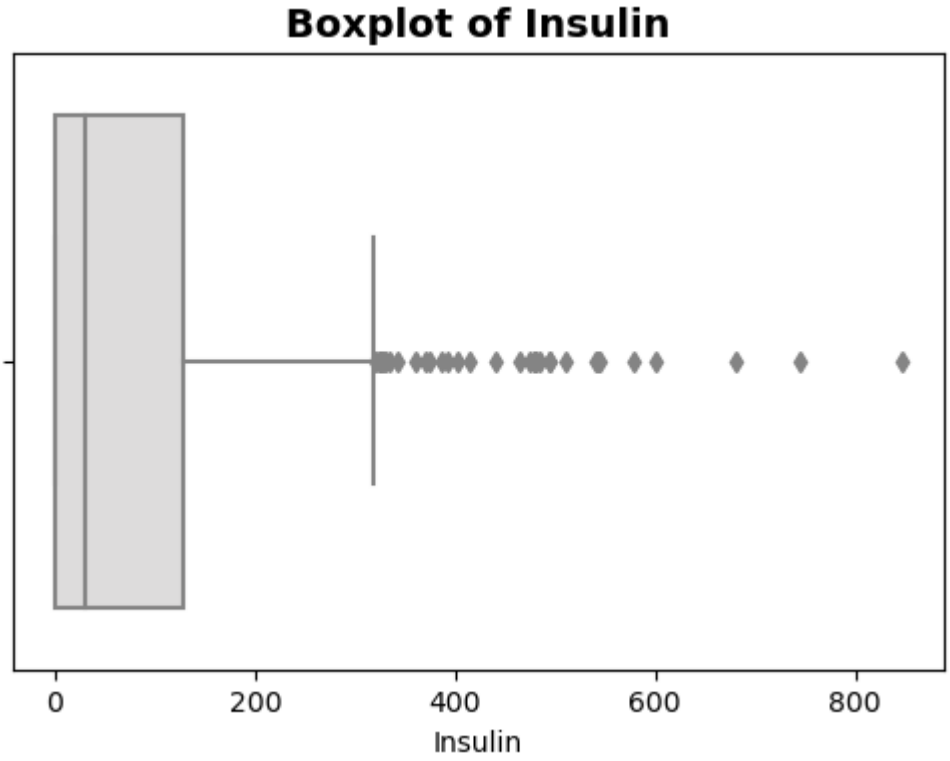
In [13]:
```
#Boxplot- Detect outliers
for col in num_cols:
    plt.figure(figsize=(6,4))
    sns.boxplot(x=df[col], palette='coolwarm')
    plt.title(f'Boxplot of {col}', fontsize=14, fontweight='bold')
    plt.show()
```

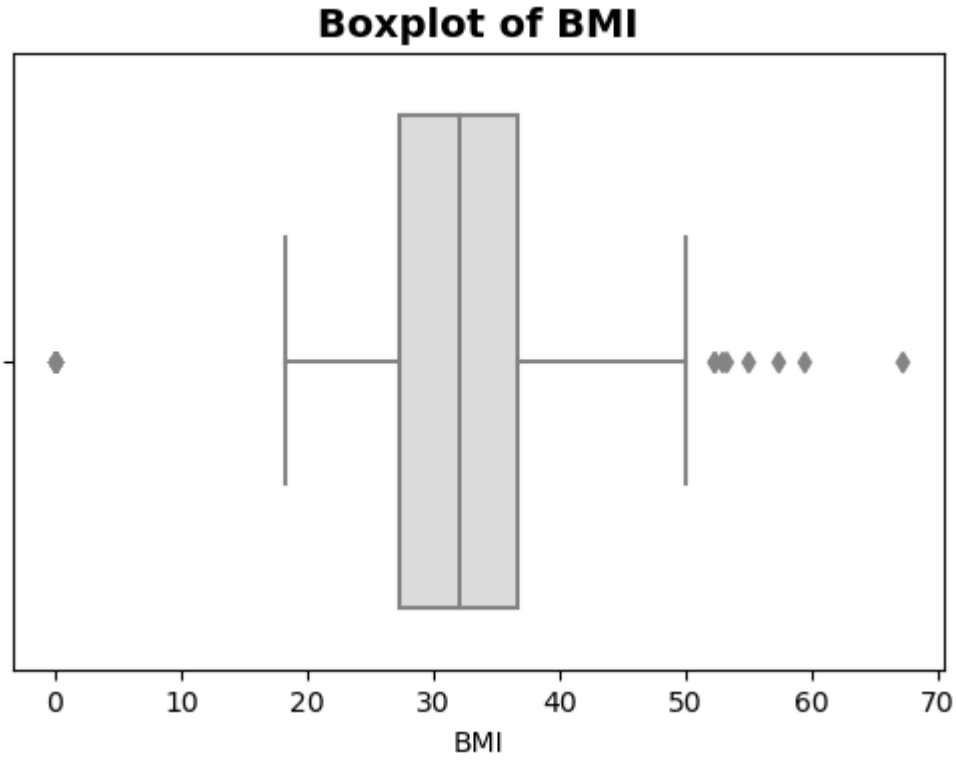


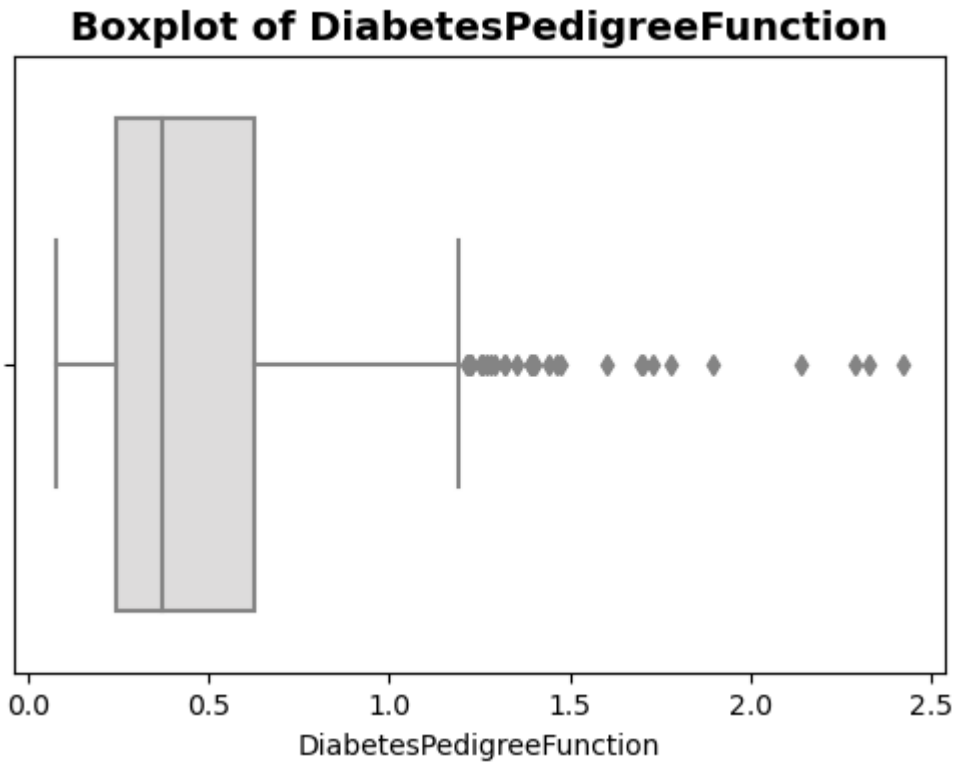


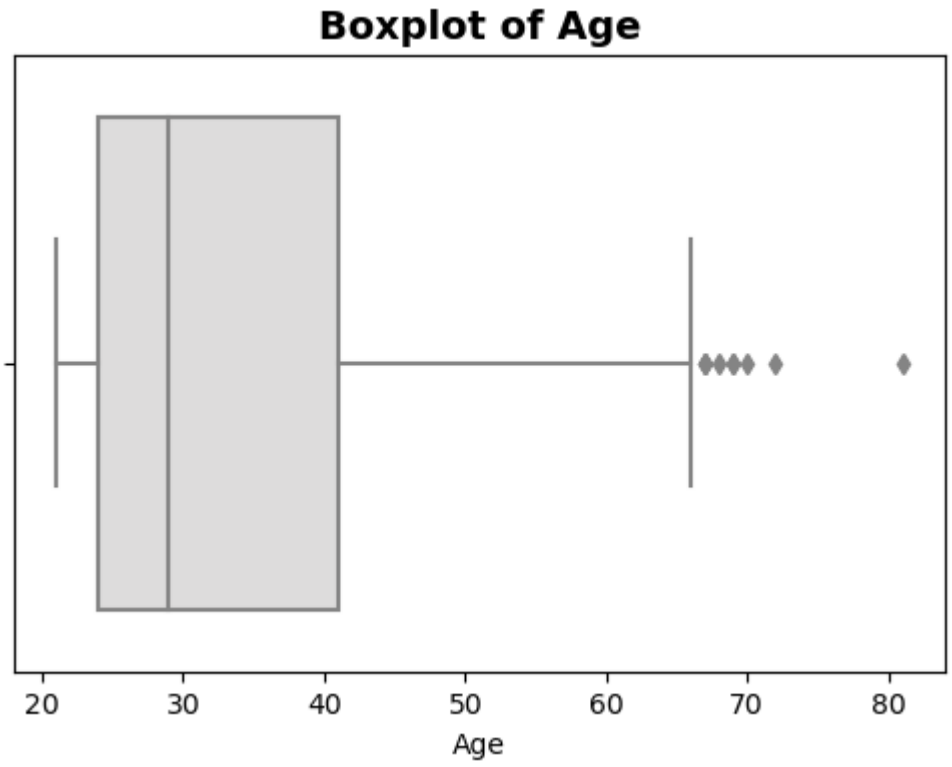

Boxplot of BloodPressure
0
20
40
60
80
100
120
BloodPressure


Boxplot of SkinThickness
0
20
40
60
80
100
SkinThickness


Boxplot of Insulin
0
200
400
600
800
Insulin


Boxplot of BMI
0
10
20
30
40
50
60
70
BMI

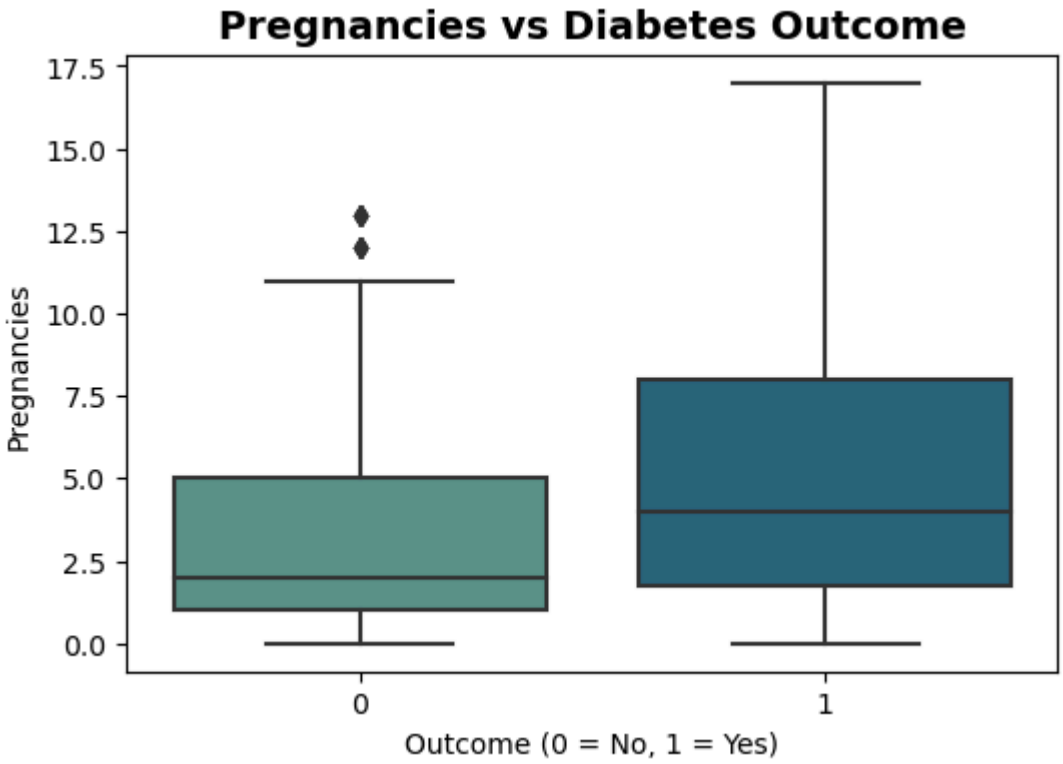


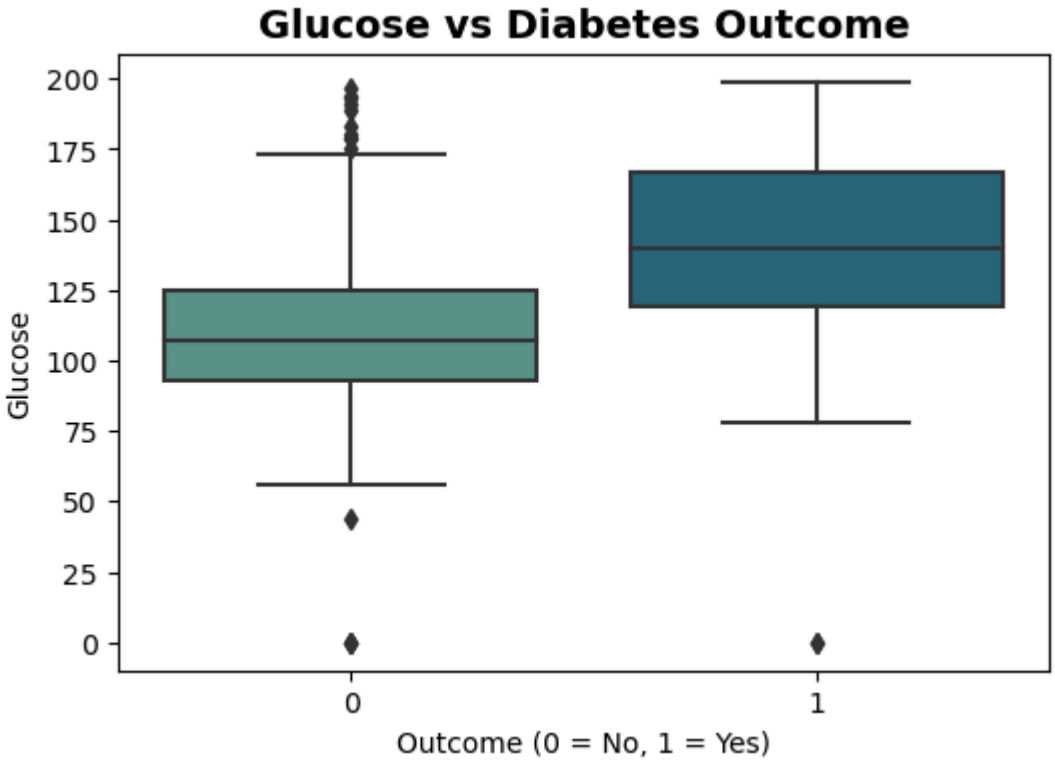


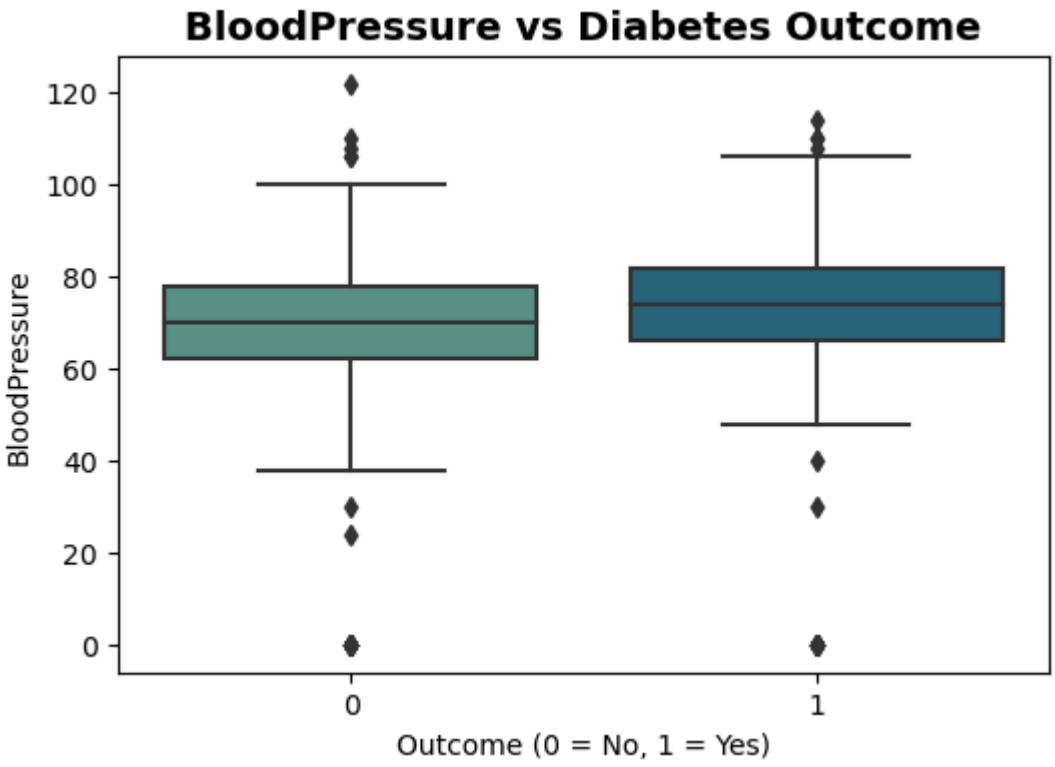
* ⚠️ Insulin, SkinThickness, and BloodPressure show strong outliers.

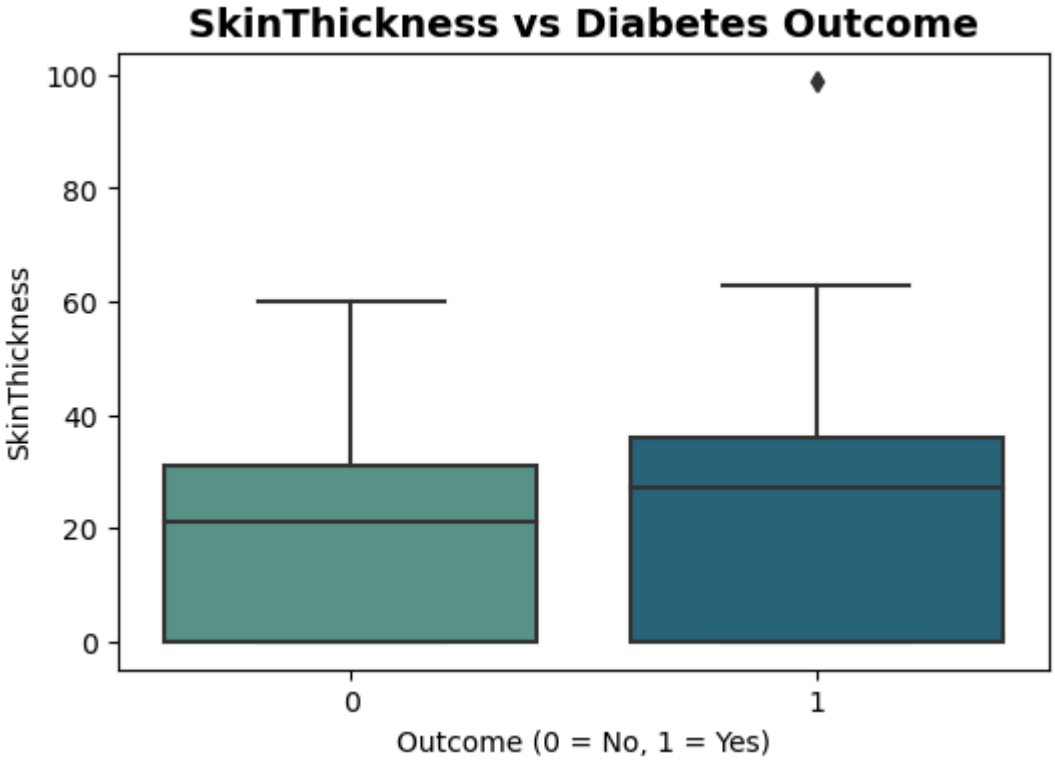
Bivariate analysis

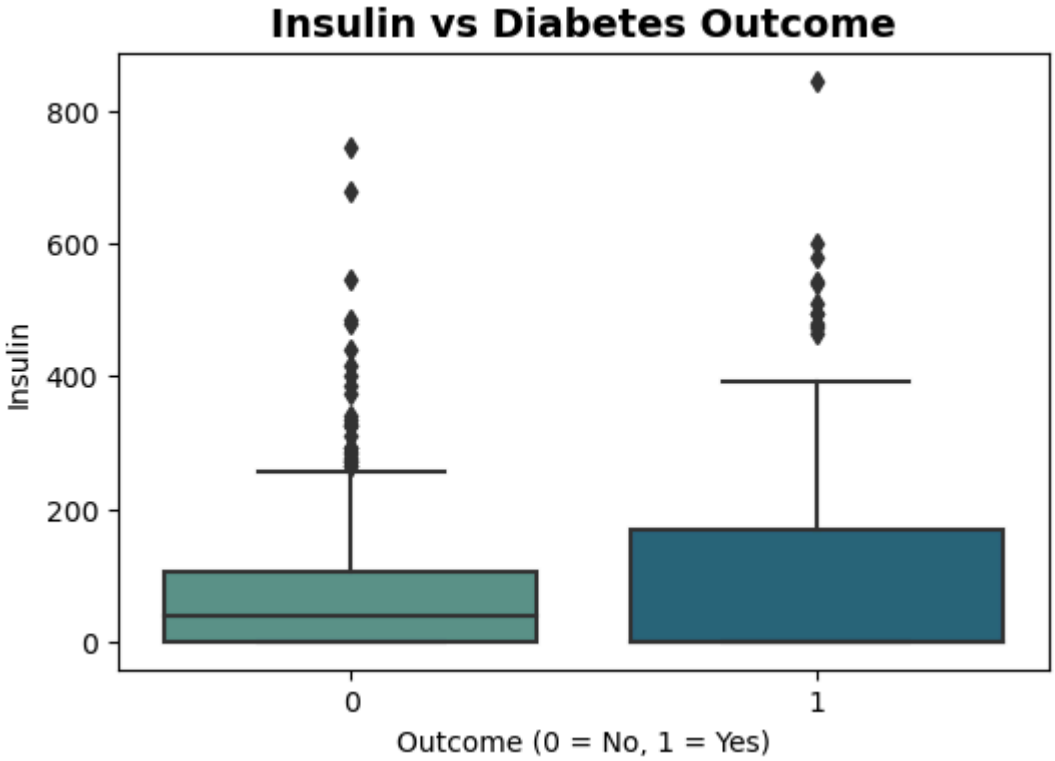
In [14]:
```
#Bivariate Analysis (Relationship with Outcome)
for col in num_cols:
    plt.figure(figsize=(6,4))
    sns.boxplot(data=df, x='Outcome', y=col, palette='crest')
    plt.title(f'{col} vs Diabetes Outcome', fontsize=14, fontweight='bold')
    plt.xlabel('Outcome (0 = No, 1 = Yes)')
    plt.ylabel(col)
    plt.show()
```

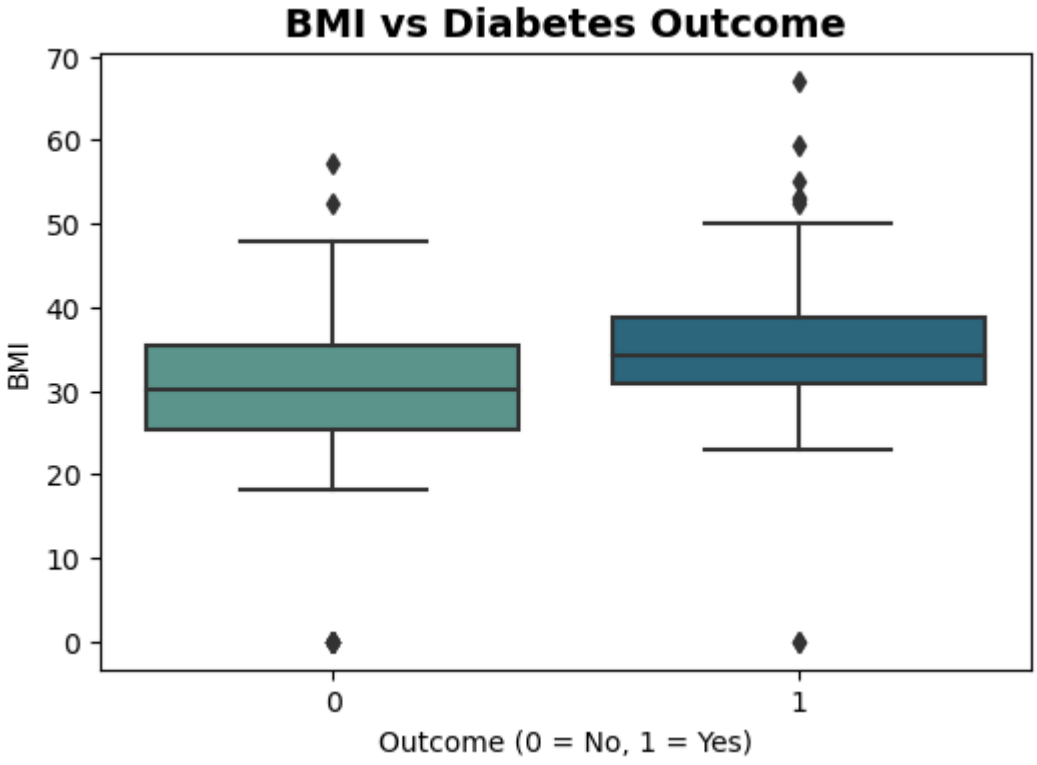


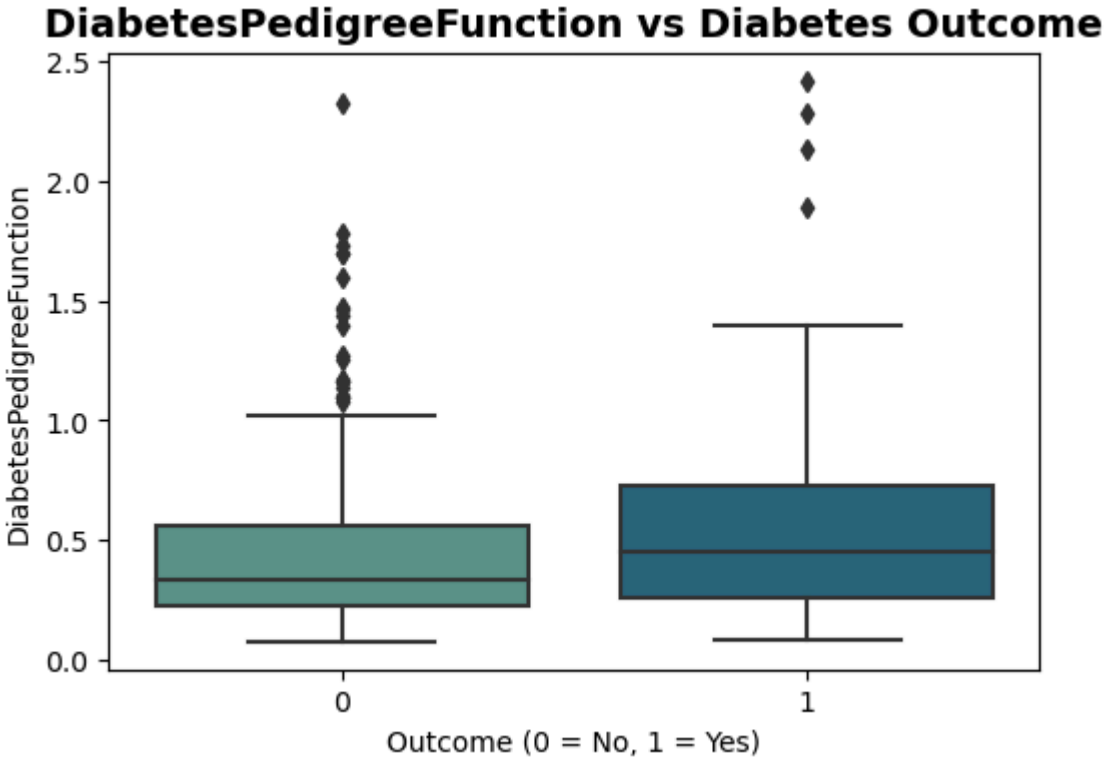


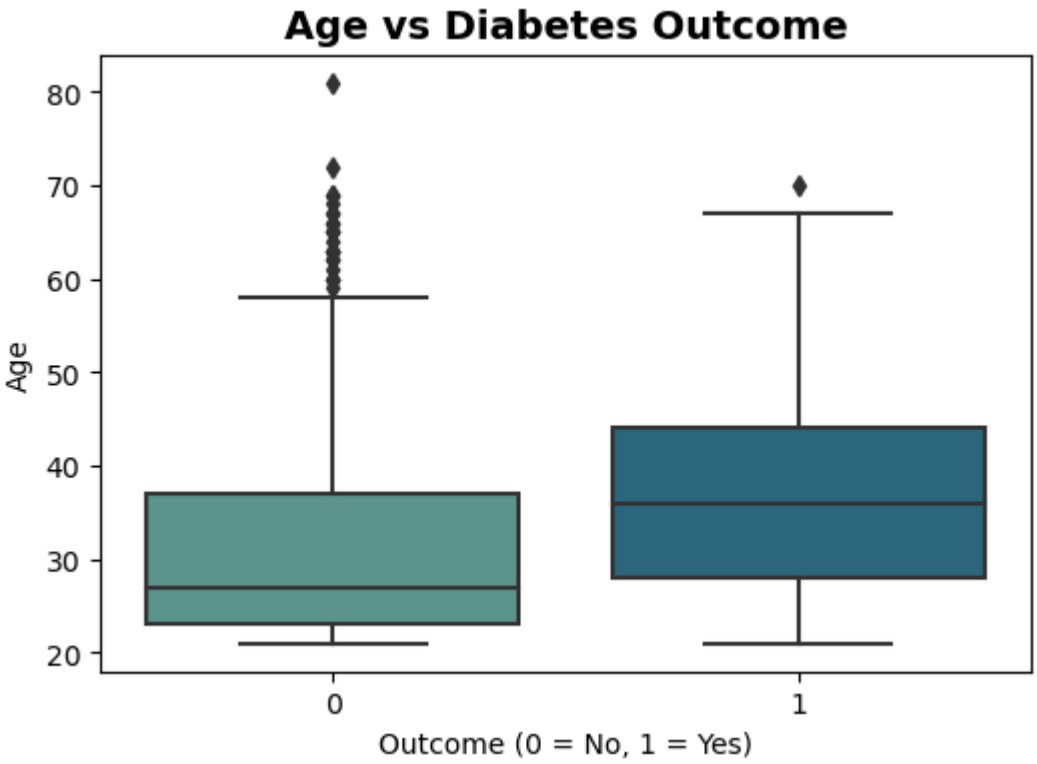

BloodPressure vs Diabetes Outcome
BloodPressure
120
100
80
60
40
20
0
0
1
Outcome (0 = No, 1 = Yes)


SkinThickness vs Diabetes Outcome
SkinThickness
100
80
60
40
20
0
0
1
Outcome (0 = No, 1 = Yes)


Insulin vs Diabetes Outcome
Insulin
800
600
400
200
0
0
1
Outcome (0 = No, 1 = Yes)

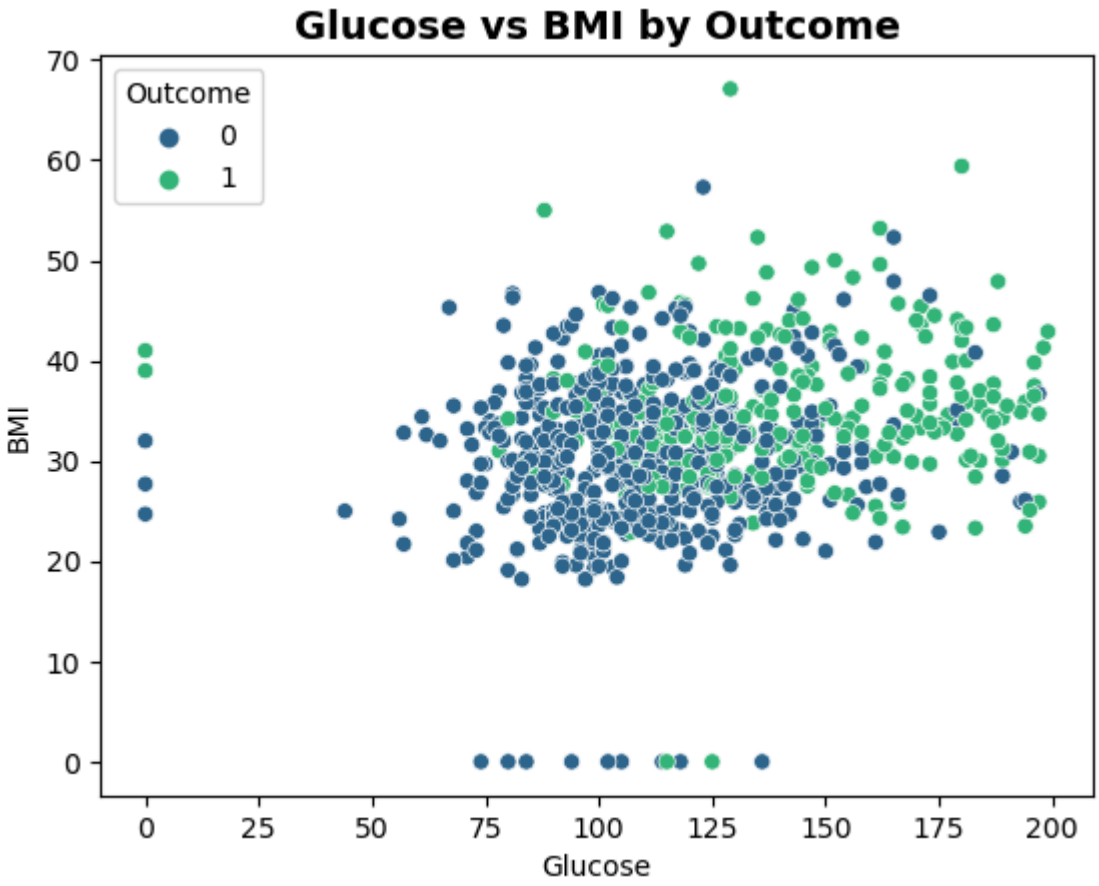

BMI vs Diabetes Outcome
BMI
70
60
50
40
30
20
10
0
0
1
Outcome (0 = No, 1 = Yes)

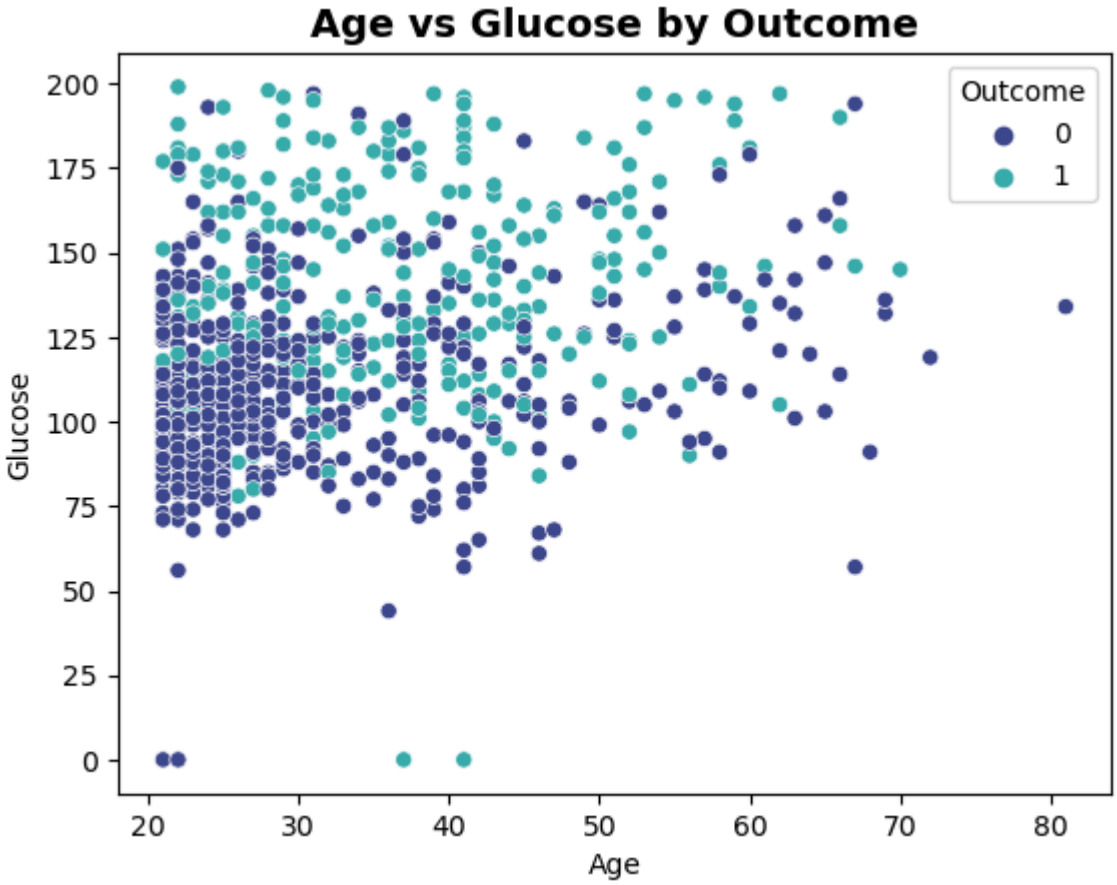




*Diabetic patients tend to have higher Glucose, BMI, and Age. Pregnancies also slightly higher for diabetic group.

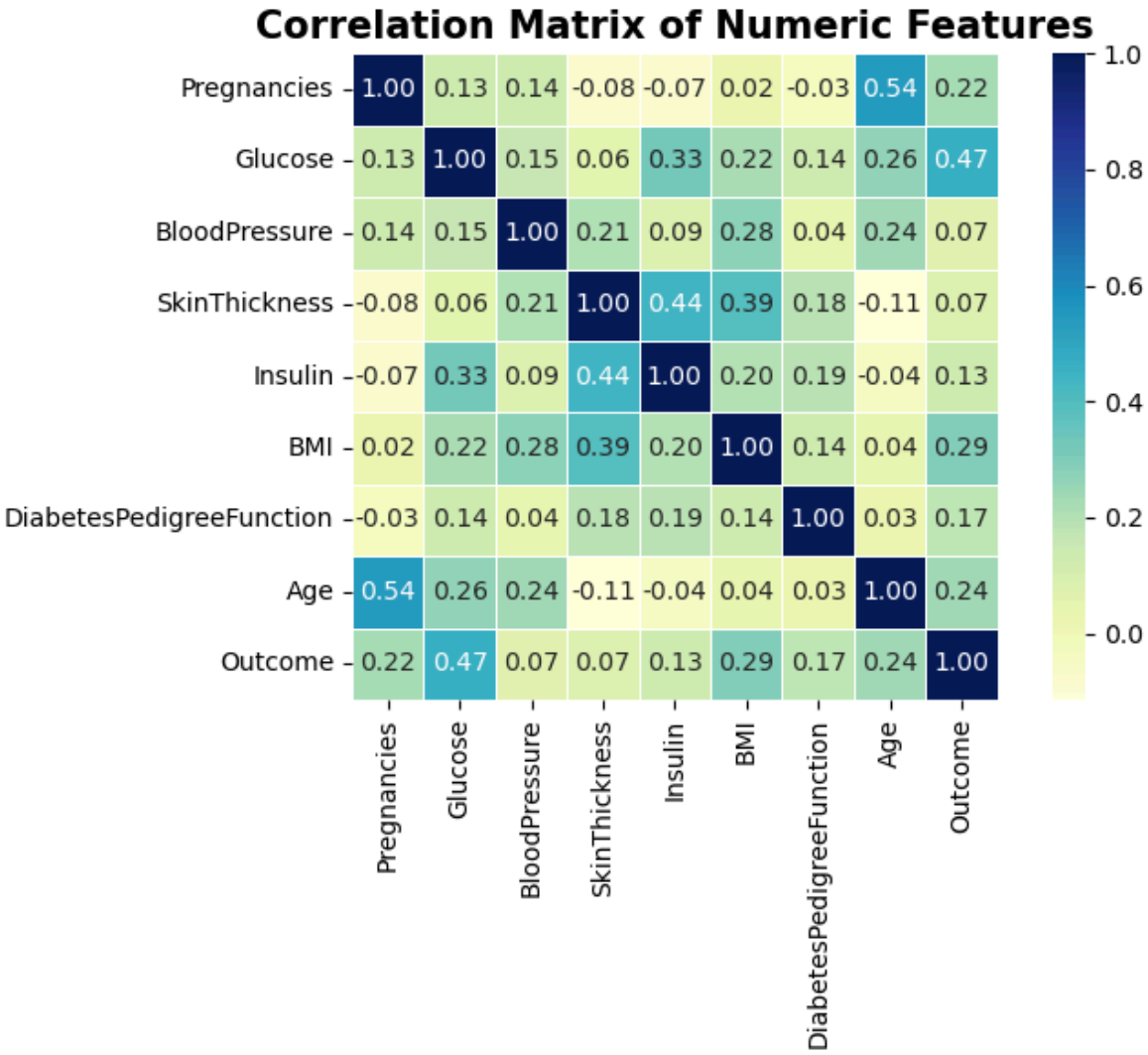
In [15]:
```
# scatter releationships
sns.scatterplot(data=df, x='Glucose', y='BMI', hue='Outcome', palette='viridis
plt.title("Glucose vs BMI by Outcome", fontsize=14, fontweight='bold')
plt.show()

sns.scatterplot(data=df, x='Age', y='Glucose', hue='Outcome', palette='mako')
plt.title("Age vs Glucose by Outcome", fontsize=14, fontweight='bold')
plt.show()
```





*Clear clustering — diabetic patients have higher glucose and higher BMI.

In [16]:
```
#Correleation Heatmap
plt.figure(figsize=(8,6))
corr = df.corr()
sns.heatmap(corr, annot=True, cmap='YlGnBu', fmt=".2f", linewidths=0.5, square
plt.title("Correlation Matrix of Numeric Features", fontsize=15, fontweight='b
plt.tight_layout()
plt.show()
```



*Glucose has the strongest correlation with Outcome (≈ 0.47) BMI, Age, and Pregnancies are also moderately correlated. Insulin and SkinThickness have weak correlation due to missing/zero issues.

*Final Conclusion — Diabetes Data EDA

After performing detailed EDA:

Dataset has 768 patients, all numeric features.

Glucose, BMI, Age, and Pregnancies are strong indicators of diabetes.

Several unrealistic zeros (for BloodPressure, Insulin, etc.) → must handle before modeling.

Strong positive correlation between Glucose and Outcome.

In [ ]: