In [9]:
```
# Introduction and Objective
print("### Introduction and Objective ###")
print("This analysis aims to explore a diabetes dataset to uncover patterns and relationships "
      "between various health metrics and the presence of diabetes. The dataset includes variables "
      "such as glucose levels, BMI, age, and others to analyze their impact on diabetes.")
```

```
### Introduction and Objective ###
This analysis aims to explore a diabetes dataset to uncover patterns and relationships between various health me
trics and the presence of diabetes. The dataset includes variables such as glucose levels, BMI, age, and others
to analyze their impact on diabetes.
```

In [10]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
```

In [11]:
```
#prompts for importing the dataset
df = pd.read_csv("diabetes.csv")
```

In [12]:
```
#to know the columns of the dataset you will be working with
df.columns
```

Out[12]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [8]:
```
#to know the number of rolls and columns in your dataset
df.shape
```

Out[8]:
```
(768, 9)
```

In [29]:
```
#to show the first five rolls of the dtaset
df.head()
```

Out[29]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [30]:
```
#to show the column names
df.columns
```

Out[30]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [31]:
```
#to check for null entries in the dataset
df . isnull(). sum()
```

Out[31]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [32]:
```
#to know the type of data in the dataset
print(df.dtypes)
```

```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
```

In [33]:
```python
#to get info of the dataset
print(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
None
```

In [18]:
```python
#display summary statictics of the dataset
df.describe()
```

Out[18]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	768.000000	768.000000	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	0.471876	33.240885
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000

In [17]:
```python
# Data Overview
print("\n### Data Overview ###")
print(f"Number of records: {df.shape[0]}")
print(f"Number of columns: {df.shape[1]}")
print("\nDescriptive Statistics:")
print(df.describe())
```

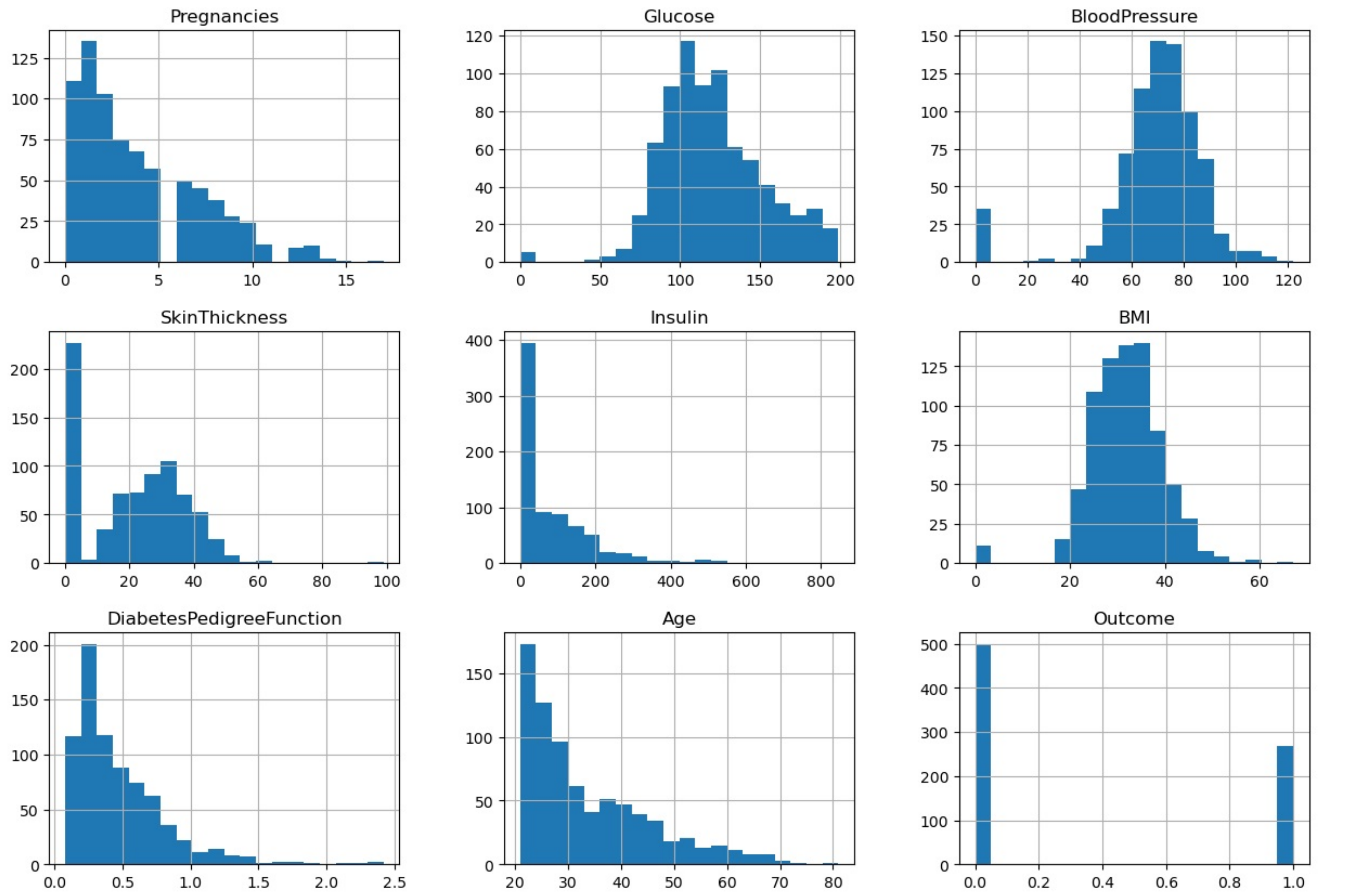
```
### Data Overview ###
Number of records: 768
Number of columns: 9

Descriptive Statistics:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000   
mean      3.845052  120.894531      69.105469      20.536458   79.799479   
std       3.369578   31.972618      19.355807      15.952218  115.244002   
min       0.000000    0.000000       0.000000       0.000000    0.000000   
25%       1.000000   99.000000      62.000000       0.000000    0.000000   
50%       3.000000  117.000000      72.000000      23.000000   30.500000   
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    31.992578                  0.471876   33.240885    0.348958  
std      7.884160                  0.331329   11.760232    0.476951  
min      0.000000                  0.078000   21.000000    0.000000  
25%     27.300000                  0.243750   24.000000    0.000000  
50%     32.000000                  0.372500   29.000000    0.000000  
75%     36.600000                  0.626250   41.000000    1.000000  
max     67.100000                  2.420000   81.000000    1.000000  
```

In [ ]:

In [19]:
```python
#plotting histograms for each feature
df.hist(bins=20,figsize=(15,10))
plt.suptitle('Distribution of demographics and health metrics')
plt.show
```

Out[19]: <function matplotlib.pyplot.show(close=None, block=None)>

Distribution of demographics and health metrics

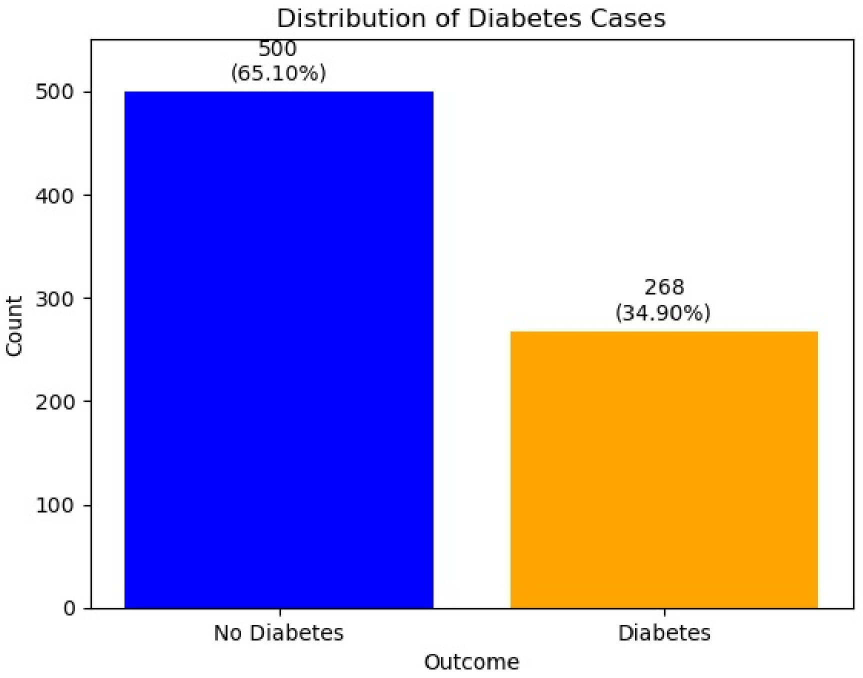


In [20]:

```
print("question1")
#what is the distribution of diabetes cases?
outcome_counts = df['Outcome'].value_counts()
outcome_percentages = df['Outcome'].value_counts(normalize=True) * 100
# Print the results
print("Below is the distribution of diabetes cases")
print("Counts:\n", outcome_counts)
print("\nPercentages:\n", outcome_percentages)
print("Insights")
# Printing the distribution of diabetes cases
# Counts
print("Based on the provided information, here is the distribution of diabetes cases:\n")
print("Counts:")
print("No Diabetes (Outcome = 0): 500 individuals")
print("Diabetes (Outcome = 1): 268 individuals\n")

# Percentages
print("Percentages:")
print("No Diabetes (Outcome = 0): 65.10%")
print("Diabetes (Outcome = 1): 34.90%\n")

# Summary
print("This means that out of the total dataset:")
print("Approximately two-thirds (65.10%) of the individuals do not have diabetes.")
print("Approximately one-third (34.90%) of the individuals have diabetes.\n")
print("This distribution indicates that while the majority of the individuals in the dataset do not have diabete

# Provided counts and percentages
counts = [500, 268]
percentages = [65.10, 34.90]
labels = ['No Diabetes', 'Diabetes']

# Plotting the distribution
fig, ax = plt.subplots()
bars = ax.bar(labels, counts, color=['blue', 'orange'])

# Adding text annotations for counts
for bar, count, percentage in zip(bars, counts, percentages):
    height = bar.get_height()
    ax.annotate(f'{count}\n({percentage:.2f}%)', xy=(bar.get_x() + bar.get_width() / 2, height),
```

```python
                   xytext=(0, 3), textcoords="offset points", ha='center', va='bottom', fontsize=10)

# Titles and labels
plt.title('Distribution of Diabetes Cases')
plt.xlabel('Outcome')
plt.ylabel('Count')
plt.ylim(0, max(counts) + 50)  # Add some space above the highest bar for annotation

# Show plot
plt.show()
```

```
question1
Below is the distribution of diabetes cases
Counts:
 Outcome
0    500
1    268
Name: count, dtype: int64

Percentages:
 Outcome
0    65.104167
1    34.895833
Name: proportion, dtype: float64
Insights
Based on the provided information, here is the distribution of diabetes cases:

Counts:
No Diabetes (Outcome = 0): 500 individuals
Diabetes (Outcome = 1): 268 individuals

Percentages:
No Diabetes (Outcome = 0): 65.10%
Diabetes (Outcome = 1): 34.90%

This means that out of the total dataset:
Approximately two-thirds (65.10%) of the individuals do not have diabetes.
Approximately one-third (34.90%) of the individuals have diabetes.

This distribution indicates that while the majority of the individuals in the dataset do not have diabetes, ther
e is still a significant portion (over one-third) that does have diabetes, highlighting the importance of unders
tanding and analyzing the factors associated with diabetes in this population.
```

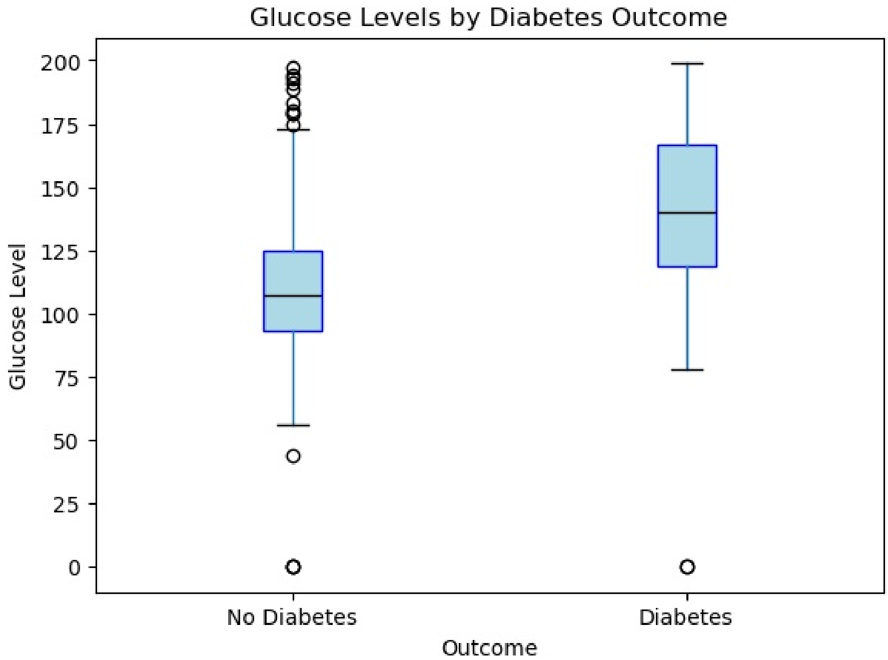


In [21]:

```python
                                                                   print("question2")
# How do glucose levels affect diabetes prevalence?
print("\n2. Glucose Levels and Diabetes Prevalence:")
mean_glucose = df.groupby('Outcome')['Glucose'].mean()
print(mean_glucose)
# Provided average glucose levels
glucose_levels = {
    0: 109.98,
    1: 141.26
}

# Printing the analysis of glucose levels and diabetes prevalence
print("Insights")
print("How does glucose level affect diabetes prevalence?")
```

```
# Average Glucose Levels
print("Based on the provided information, here is the analysis of glucose levels in relation to diabetes preval
print("Average Glucose Levels:")
print(f"No Diabetes (Outcome = 0): {glucose_levels[0]:.2f}")
print(f"Diabetes (Outcome = 1): {glucose_levels[1]:.2f}\n")

# Summary
print("This data suggests that individuals with diabetes tend to have significantly higher glucose levels compa
# Plot the glucose levels
df.boxplot(column='Glucose', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Glucose Levels by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Glucose Level')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

```
question2

2. Glucose Levels and Diabetes Prevalence:
Outcome
0    109.980000
1    141.257463
Name: Glucose, dtype: float64
Insights
How does glucose level affect diabetes prevalence?
Based on the provided information, here is the analysis of glucose levels in relation to diabetes prevalence:

Average Glucose Levels:
No Diabetes (Outcome = 0): 109.98
Diabetes (Outcome = 1): 141.26

This data suggests that individuals with diabetes tend to have significantly higher glucose levels compared to t
hose without diabetes.
```

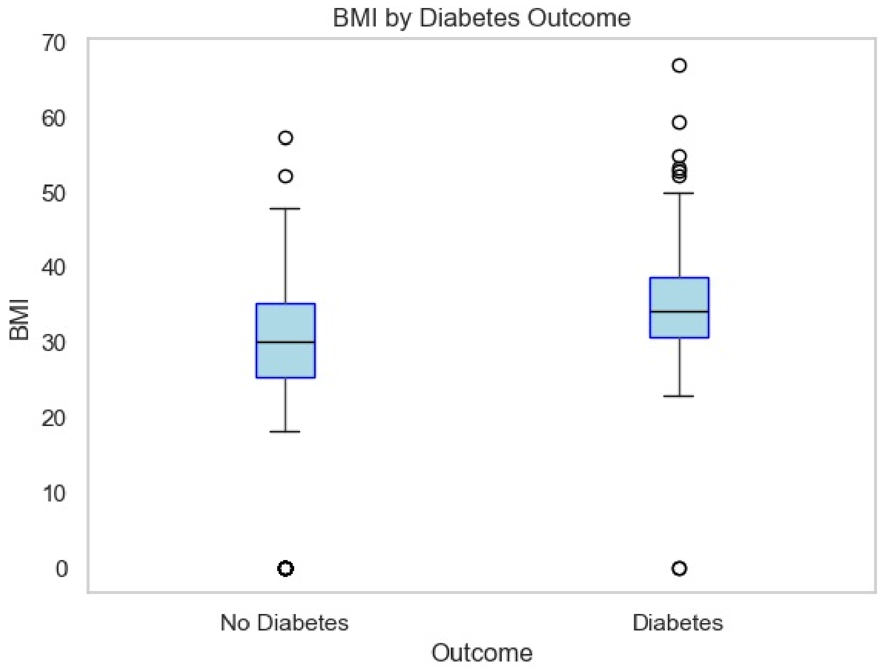


In [151…

```
                                                                    print("question3")
# Compare mean BMI
mean_bmi = df.groupby('Outcome')['BMI'].mean()
print(mean_bmi)
# Provided mean BMI values
mean_bmi = {
    0: 30.30,
    1: 35.14
}

# Printing the analysis of BMI and diabetes prevalence
print("Insights")
print("What is the relationship between BMI and diabetes?\n")

# Mean BMI
print("Based on the provided information, here is the analysis of the mean BMI in relation to diabetes:\n")
print("Mean BMI:")
print(f"No Diabetes (Outcome = 0): {mean_bmi[0]:.2f}")
print(f"Diabetes (Outcome = 1): {mean_bmi[1]:.2f}\n")

# Summary
```

```python
print("This data indicates that individuals with diabetes tend to have a higher Body Mass Index (BMI) compared 

# Plot the BMI
df.boxplot(column='BMI', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('BMI by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('BMI')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

```
question3
Outcome
0    30.304200
1    35.142537
Name: BMI, dtype: float64
Insights
What is the relationship between BMI and diabetes?

Based on the provided information, here is the analysis of the mean BMI in relation to diabetes:

Mean BMI:
No Diabetes (Outcome = 0): 30.30
Diabetes (Outcome = 1): 35.14

This data indicates that individuals with diabetes tend to have a higher Body Mass Index (BMI) compared to those
without diabetes.
```

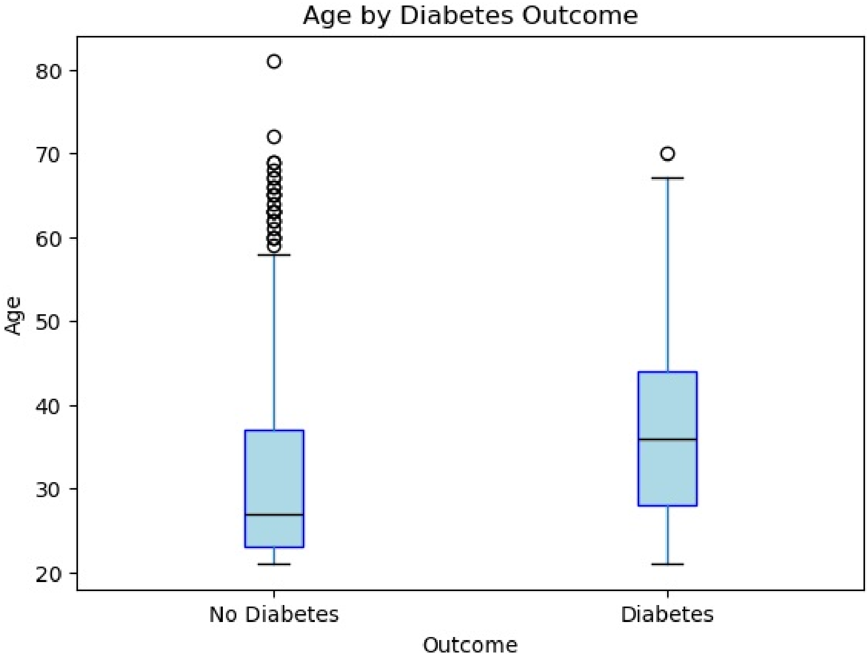


In [22]:

```python
                                                                          print("QUESTION4")
#How does age influence the risk of diabetes?
# Compare age distributions
mean_age = df.groupby('Outcome')['Age'].mean()
print(mean_age)
print("insights")
# Provided mean age values
mean_age = {
    0: 31.19,
    1: 37.07
}

# Printing the analysis of age and diabetes prevalence
print("How does age influence the risk of diabetes?\n")

# Mean Age
print("Based on the provided information, here is the analysis of the mean age in relation to diabetes:\n")
print("Mean Age:")
print(f"No Diabetes (Outcome = 0): {mean_age[0]:.2f} years")
print(f"Diabetes (Outcome = 1): {mean_age[1]:.2f} years\n")

# Summary
print("This data suggests that individuals with diabetes tend to be older on average compared to those without 
# Plot the age distributions
df.boxplot(column='Age', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Age by Diabetes Outcome')
plt.suptitle('')
```

```
plt.xlabel('Outcome')
plt.ylabel('Age')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

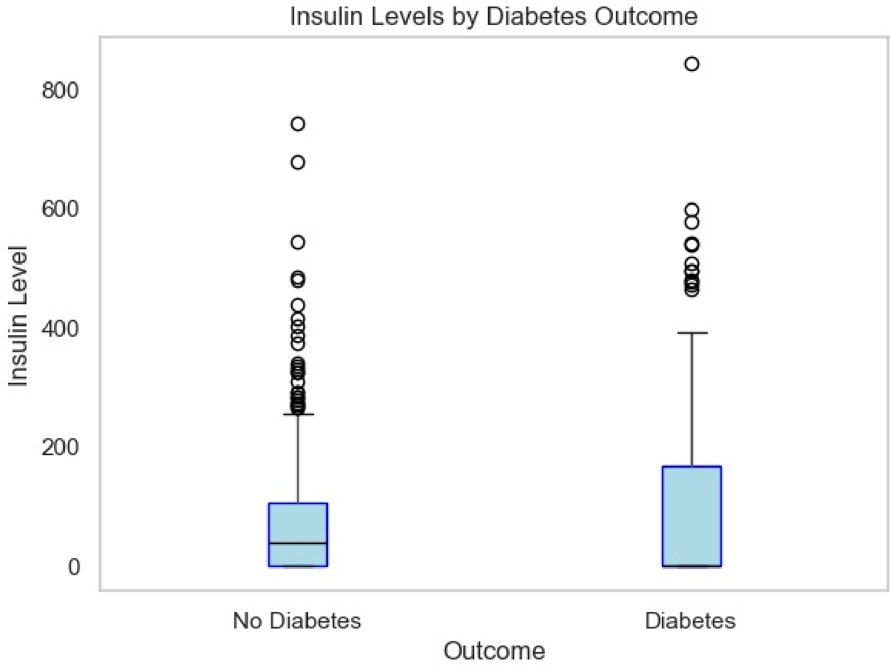
```
QUESTION4
Outcome
0    31.190000
1    37.067164
Name: Age, dtype: float64
insights
How does age influence the risk of diabetes?

Based on the provided information, here is the analysis of the mean age in relation to diabetes:

Mean Age:
No Diabetes (Outcome = 0): 31.19 years
Diabetes (Outcome = 1): 37.07 years

This data suggests that individuals with diabetes tend to be older on average compared to those without diabetes
.
```



In [161…

```
                                                                          print("QUESTION5")
#Do insulin levels vary significantly between diabetic and non-diabetic individuals?
# Compare mean insulin levels
mean_insulin = df.groupby('Outcome')['Insulin'].mean()
print(mean_insulin)
# Provided mean insulin levels
mean_insulin = {
    0: 68.79,
    1: 100.34
}

# Printing the analysis of insulin levels and diabetes prevalence
print("Insights")
print("Do insulin levels vary significantly between diabetic and non-diabetic individuals?\n")

# Mean Insulin Levels
print("Based on the provided information, here is the analysis of the mean insulin levels in relation to diabet
print("Mean Insulin Levels:")
print(f"No Diabetes (Outcome = 0): {mean_insulin[0]:.2f}")
print(f"Diabetes (Outcome = 1): {mean_insulin[1]:.2f}\n")

# Summary
print("This data indicates that individuals with diabetes tend to have significantly higher insulin levels compa
# Plot the insulin levels
df.boxplot(column='Insulin', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Insulin Levels by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Insulin Level')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

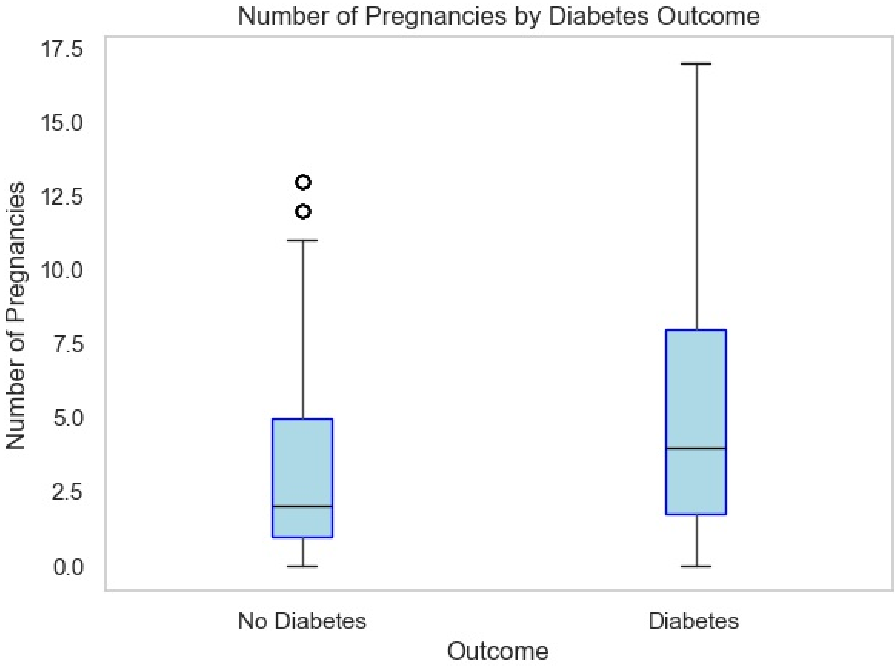
```
QUESTION5
Outcome
0     68.792000
1    100.335821
Name: Insulin, dtype: float64
Insights
Do insulin levels vary significantly between diabetic and non-diabetic individuals?

Based on the provided information, here is the analysis of the mean insulin levels in relation to diabetes:

Mean Insulin Levels:
No Diabetes (Outcome = 0): 68.79
Diabetes (Outcome = 1): 100.34

This data indicates that individuals with diabetes tend to have significantly higher insulin levels compared to
those without diabetes.
```



In [165…

```python
                                                          print("QUESTION6")
#How does the number of pregnancies correlate with diabetes?
# Compare the number of pregnancies
mean_pregnancies = df.groupby('Outcome')['Pregnancies'].mean()
print(mean_pregnancies)
# Provided mean number of pregnancies
mean_pregnancies = {
    0: 3.30,
    1: 4.87
}

# Printing the analysis of the number of pregnancies and diabetes prevalence
print("How does the number of pregnancies correlate with diabetes?\n")

# Mean Number of Pregnancies
print("Based on the provided information, here is the analysis of the mean number of pregnancies in relation to
print("Mean Number of Pregnancies:")
print(f"No Diabetes (Outcome = 0): {mean_pregnancies[0]:.2f}")
print(f"Diabetes (Outcome = 1): {mean_pregnancies[1]:.2f}\n")

# Summary
print("This data suggests that individuals with diabetes tend to have a higher number of pregnancies on average
# Plot the number of pregnancies
df.boxplot(column='Pregnancies', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Number of Pregnancies by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Number of Pregnancies')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

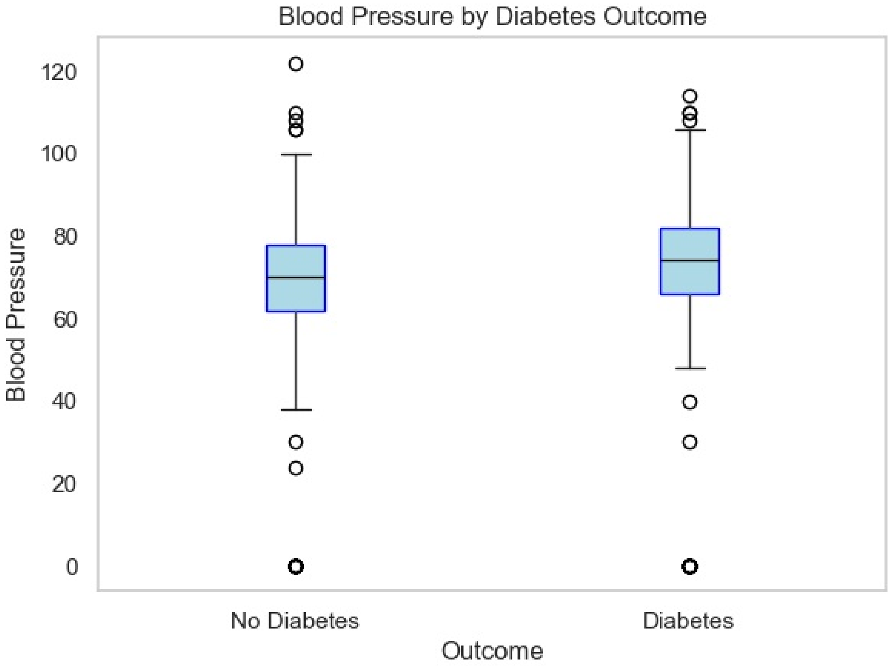
```
QUESTION6
Outcome
0    3.298000
1    4.865672
Name: Pregnancies, dtype: float64
How does the number of pregnancies correlate with diabetes?

Based on the provided information, here is the analysis of the mean number of pregnancies in relation to diabete
s:

Mean Number of Pregnancies:
No Diabetes (Outcome = 0): 3.30
Diabetes (Outcome = 1): 4.87

This data suggests that individuals with diabetes tend to have a higher number of pregnancies on average compare
d to those without diabetes.
```



In [167…

```python
                                                                    print("QUESTION7")
#What is the role of Blood Pressure in diabetes?
# Compare mean blood pressure
mean_bp = df.groupby('Outcome')['BloodPressure'].mean()
print(mean_bp)
# Provided mean blood pressure levels
mean_blood_pressure = {
    0: 68.18,
    1: 70.82
}

# Printing the analysis of blood pressure levels and diabetes prevalence
print("What is the role of Blood Pressure in diabetes?\n")

# Mean Blood Pressure Levels
print("Based on the provided information, here is the analysis of the mean blood pressure levels in relation to
print("Mean Blood Pressure Levels:")
print(f"No Diabetes (Outcome = 0): {mean_blood_pressure[0]:.2f}")
print(f"Diabetes (Outcome = 1): {mean_blood_pressure[1]:.2f}\n")

# Summary
print("This data suggests that there is a slight increase in mean blood pressure levels among individuals with 
# Plot the blood pressure levels
df.boxplot(column='BloodPressure', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Blood Pressure by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Blood Pressure')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

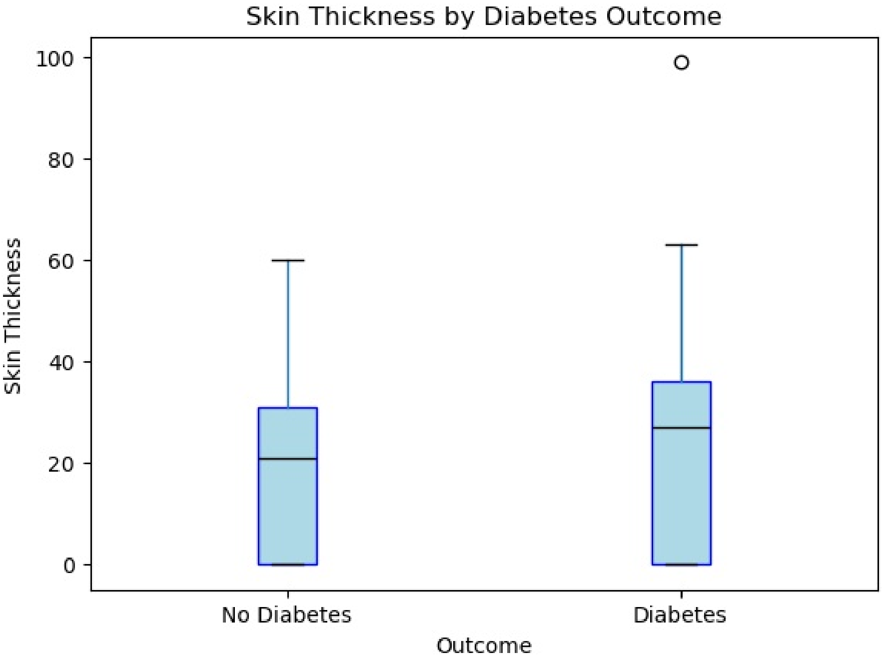
```
QUESTION7
Outcome
0    68.184000
1    70.824627
Name: BloodPressure, dtype: float64
What is the role of Blood Pressure in diabetes?

Based on the provided information, here is the analysis of the mean blood pressure levels in relation to diabete
s:

Mean Blood Pressure Levels:
No Diabetes (Outcome = 0): 68.18
Diabetes (Outcome = 1): 70.82

This data suggests that there is a slight increase in mean blood pressure levels among individuals with diabetes
compared to those without diabetes, though the difference is not substantial.
```



In [23]:

```python
                                                    print("QUESTION8")
#Is there a connection between skin thickness and diabetes?
# Compare mean skin thickness
mean_skin_thickness = df.groupby('Outcome')['SkinThickness'].mean()
print(mean_skin_thickness)
# Provided mean skin thickness values
mean_skin_thickness = {
    0: 19.66,
    1: 22.16
}
print("insights")
# Printing the analysis of skin thickness and diabetes prevalence
print("Is there a connection between skin thickness and diabetes?\n")

# Mean Skin Thickness
print("Based on the provided information, here is the analysis of the mean skin thickness in relation to diabete
print("Mean Skin Thickness:")
print(f"No Diabetes (Outcome = 0): {mean_skin_thickness[0]:.2f}")
print(f"Diabetes (Outcome = 1): {mean_skin_thickness[1]:.2f}\n")

# Summary
print("This data suggests that individuals with diabetes tend to have slightly higher mean skin thickness compa


# Plot the skin thickness
df.boxplot(column='SkinThickness', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Skin Thickness by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Skin Thickness')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

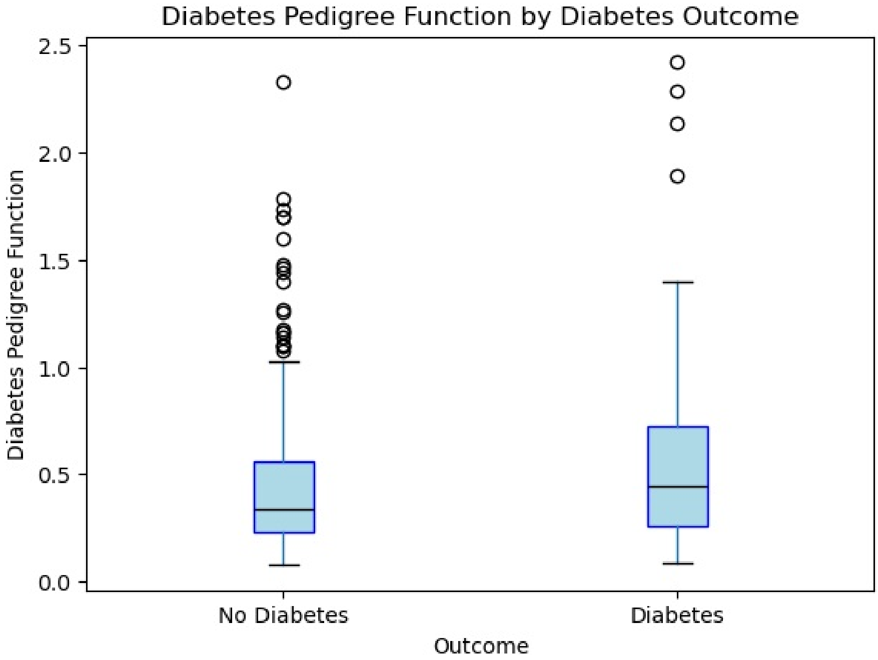
```
QUESTION8
Outcome
0    19.664000
1    22.164179
Name: SkinThickness, dtype: float64
insights
Is there a connection between skin thickness and diabetes?

Based on the provided information, here is the analysis of the mean skin thickness in relation to diabetes:

Mean Skin Thickness:
No Diabetes (Outcome = 0): 19.66
Diabetes (Outcome = 1): 22.16

This data suggests that individuals with diabetes tend to have slightly higher mean skin thickness compared to t
hose without diabetes, indicating a potential connection between skin thickness and diabetes.
```



In [24]:

```
                                                                    print("question9")
#How does the Diabetes Pedigree Function affect diabetes prevalence?
# Compare Diabetes Pedigree Function scores
mean_dpf = df.groupby('Outcome')['DiabetesPedigreeFunction'].mean()
print(mean_dpf)
print("insights")
# Provided mean Diabetes Pedigree Function values
mean_dpf = {
    0: 0.4297,
    1: 0.5505
}

# Printing the analysis of Diabetes Pedigree Function and diabetes prevalence
print("How does the Diabetes Pedigree Function affect diabetes prevalence?\n")

# Mean Diabetes Pedigree Function
print("Based on the provided information, here is the analysis of the mean Diabetes Pedigree Function in relati
print("Mean Diabetes Pedigree Function:")
print(f"No Diabetes (Outcome = 0): {mean_dpf[0]:.4f}")
print(f"Diabetes (Outcome = 1): {mean_dpf[1]:.4f}\n")

# Summary
print("This data suggests that individuals with diabetes tend to have higher mean Diabetes Pedigree Function va
print("The Diabetes Pedigree Function is a measure of genetic susceptibility to diabetes, indicating that indiv
# Plot the Diabetes Pedigree Function
df.boxplot(column='DiabetesPedigreeFunction', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Diabetes Pedigree Function by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Diabetes Pedigree Function')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

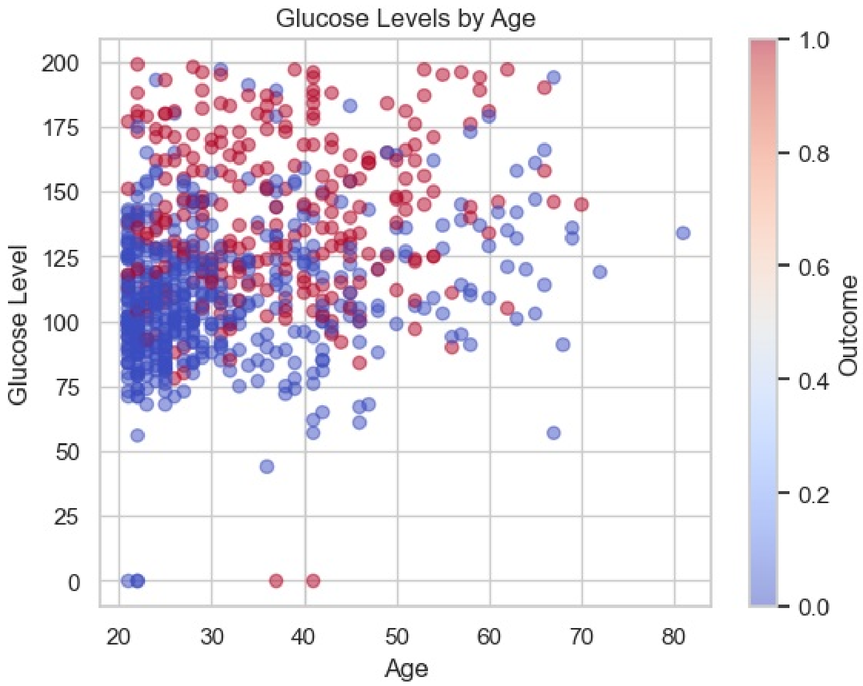
```
question9
Outcome
0    0.429734
1    0.550500
Name: DiabetesPedigreeFunction, dtype: float64
insights
How does the Diabetes Pedigree Function affect diabetes prevalence?

Based on the provided information, here is the analysis of the mean Diabetes Pedigree Function in relation to di
abetes:

Mean Diabetes Pedigree Function:
No Diabetes (Outcome = 0): 0.4297
Diabetes (Outcome = 1): 0.5505

This data suggests that individuals with diabetes tend to have higher mean Diabetes Pedigree Function values com
pared to those without diabetes.
The Diabetes Pedigree Function is a measure of genetic susceptibility to diabetes, indicating that individuals w
ith higher DPF values may have a higher genetic risk for developing diabetes.
```



In [180…

```python
                                                            #question10
# Calculate the correlation
correlation = df['Age'].corr(df['Glucose'])
print(f'Correlation between Age and Glucose Level: {correlation:.2f}')
#How does glucose levels vary with age?

# Plot glucose levels against age
plt.scatter(df['Age'], df['Glucose'], alpha=0.5, c=df['Outcome'], cmap='coolwarm')
plt.title('Glucose Levels by Age')
plt.xlabel('Age')
plt.ylabel('Glucose Level')
plt.colorbar(label='Outcome')
plt.show()
```

```
Correlation between Age and Glucose Level: 0.26
```



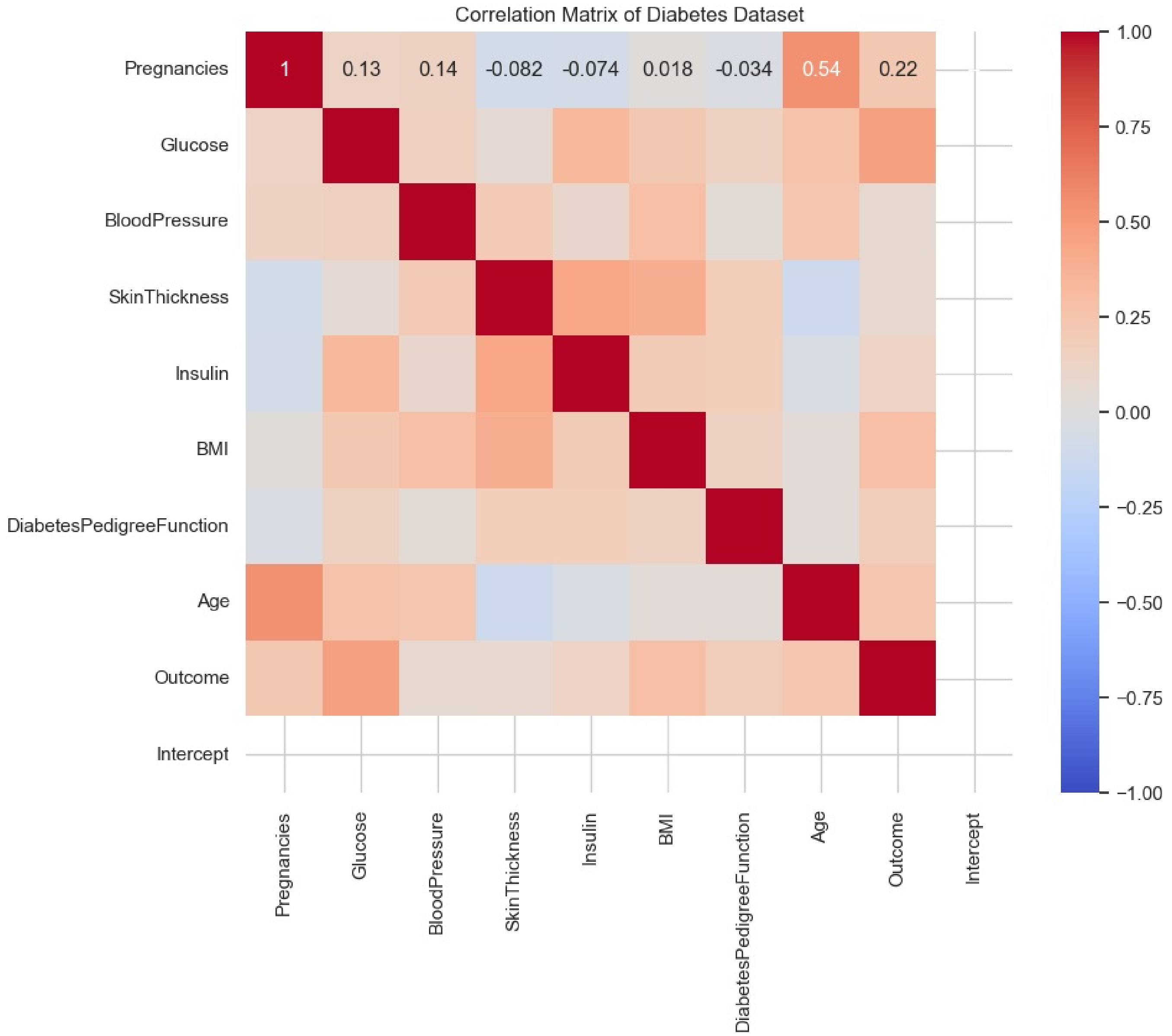
In [25]:

```
                                                                  print("question11")
#How does age influence the risk of diabetes?
# Compare age distributions
mean_age = df.groupby('Outcome')['Age'].mean()
print(mean_age)
# Provided mean age values
mean_age = {
    0: 31.19,
    1: 37.07
}
print("
# Printing the analysis of age and diabetes prevalence
print("How does age influence the risk of diabetes?\n")

# Mean Age
print("Based on the provided information, here is the analysis of the mean age in relation to diabetes:\n")
print("Mean Age:")
print(f"No Diabetes (Outcome = 0): {mean_age[0]:.2f} years")
print(f"Diabetes (Outcome = 1): {mean_age[1]:.2f} years\n")

# Summary
print("This data suggests that individuals with diabetes tend to be older on average compared to those without d
print("The higher mean age for individuals with diabetes indicates that age may be a risk factor for developing 
# Plot the age distributions
df.boxplot(column='Age', by='Outcome', grid=False, patch_artist=True,
           medianprops=dict(color='black'), boxprops=dict(color='blue', facecolor='lightblue'))
plt.title('Age by Diabetes Outcome')
plt.suptitle('')
plt.xlabel('Outcome')
plt.ylabel('Age')
plt.xticks(ticks=[1, 2], labels=['No Diabetes', 'Diabetes'])
plt.show()
```

```
  Cell In[25], line 11
    print("
      ^
SyntaxError: unterminated string literal (detected at line 11)
```

In [37]:
```python
                                                                #question12
#What is the correlation matrix of the dataset?
#Insight: Which variables are strongly correlated with each other, and how might they affect diabetes risk?
# Calculate and plot the correlation matrix
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix of Diabetes Dataset')
plt.show()
```

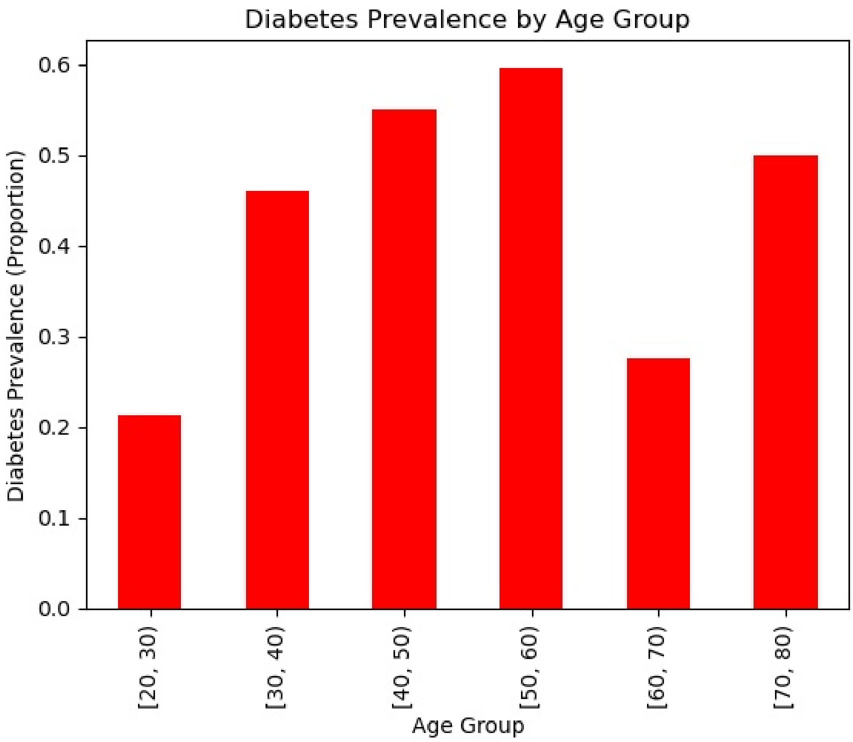
```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\matrix.py:260: FutureWarning: Format strings passed to Masked
Constant are ignored, but in future may error or produce different behavior
  annotation = ("{:" + self.fmt + "}").format(val)
```



In [26]:
```python
                                                        #question13
#How does diabetes prevalence change across different age groups?
#Insight: Are certain age groups more prone to diabetes than others?
# Define age groups and calculate diabetes prevalence
df['AgeGroup'] = pd.cut(df['Age'], bins=[20, 30, 40, 50, 60, 70, 80], right=False)
age_group_counts = df.groupby('AgeGroup')['Outcome'].mean()

# Print the results
print(age_group_counts)
# Provided diabetes prevalence by age group
age_group_prevalence = {
    '[20, 30)': 0.212121,
    '[30, 40)': 0.460606,
    '[40, 50)': 0.550847,
    '[50, 60)': 0.596491,
    '[60, 70)': 0.275862,
    '[70, 80)': 0.500000
}
```

```python
# Printing the analysis of diabetes prevalence across different age groups
print("How does diabetes prevalence change across different age groups?\n")
print("insights")
# Diabetes Prevalence by Age Group
print("Based on the provided information, here is the analysis of diabetes prevalence across different age grou
print("Diabetes Prevalence by Age Group:")
for age_group, prevalence in age_group_prevalence.items():
    print(f"{age_group} years: {prevalence:.2%}")

# Summary
print("\nThis data suggests that the prevalence of diabetes increases with age up to the 50-60 year age group, 
print("After this peak, the prevalence decreases in the 60-70 year age group to 27.59%, then increases again to
print("This pattern indicates that middle-aged individuals (40-60 years) have the highest prevalence of diabete
# Plot the diabetes prevalence across age groups
age_group_counts.plot(kind='bar', color='red')
plt.title('Diabetes Prevalence by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Diabetes Prevalence (Proportion)')
plt.show()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11180\3600275711.py:6: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current b
ehavior or observed=True to adopt the future default and silence this warning.
  age_group_counts = df.groupby('AgeGroup')['Outcome'].mean()
AgeGroup
[20, 30)    0.212121
[30, 40)    0.460606
[40, 50)    0.550847
[50, 60)    0.596491
[60, 70)    0.275862
[70, 80)    0.500000
Name: Outcome, dtype: float64
How does diabetes prevalence change across different age groups?

insights
Based on the provided information, here is the analysis of diabetes prevalence across different age groups:

Diabetes Prevalence by Age Group:
[20, 30) years: 21.21%
[30, 40) years: 46.06%
[40, 50) years: 55.08%
[50, 60) years: 59.65%
[60, 70) years: 27.59%
[70, 80) years: 50.00%

This data suggests that the prevalence of diabetes increases with age up to the 50-60 year age group, where it p
eaks at 59.65%.
After this peak, the prevalence decreases in the 60-70 year age group to 27.59%, then increases again to 50.00%
in the 70-80 year age group.
This pattern indicates that middle-aged individuals (40-60 years) have the highest prevalence of diabetes, follo
wed by a decrease in prevalence in the next decade, with a subsequent increase in older age.
```

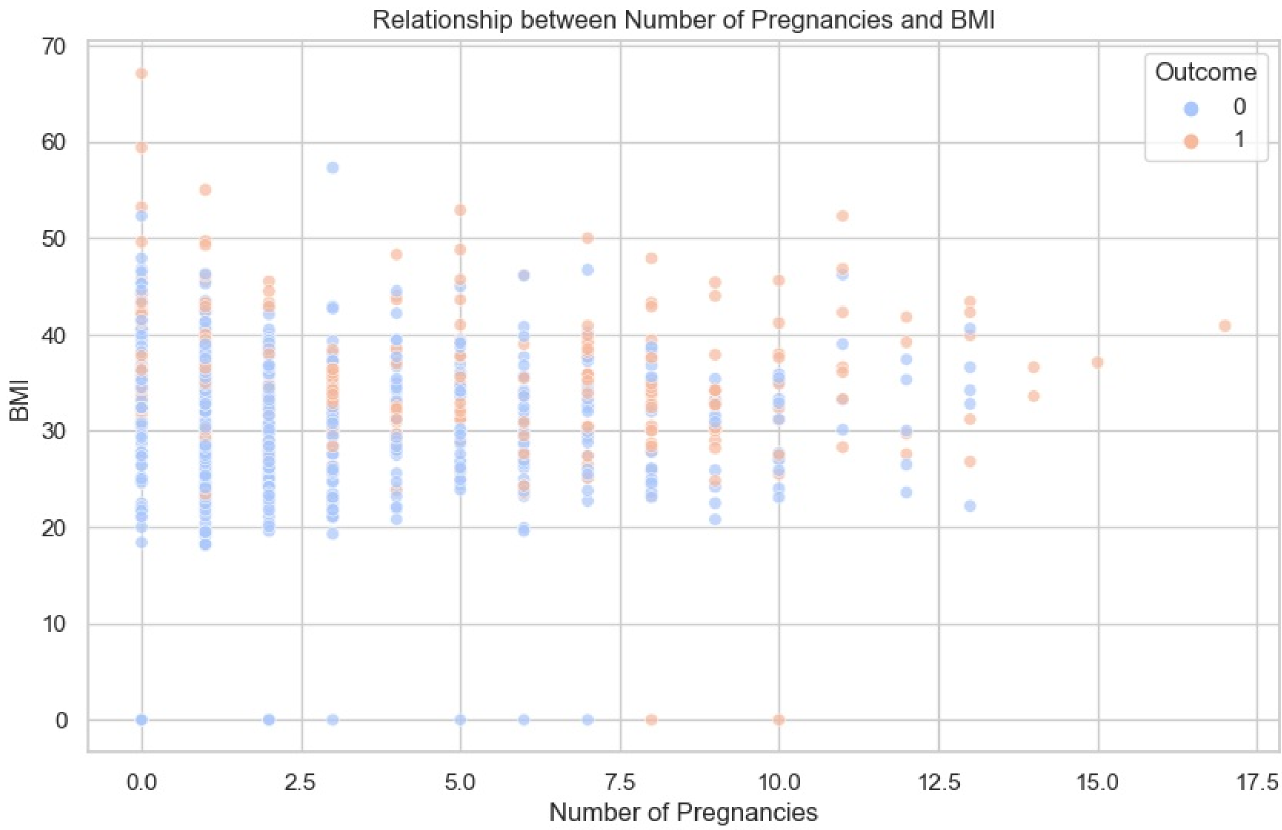


In [197…

```python
#question14
```

```python
#Is there a relationship between the number of pregnancies and BMI?
# Calculate the correlation
correlation = df['Pregnancies'].corr(df['BMI'])
print(f'Correlation between Number of Pregnancies and BMI: {correlation:.2f}')
# Insight: Is there a relationship between the number of pregnancies and BMI?
correlation = 0.02  # Replace with the actual correlation coefficient calculated
print("### Is there a relationship between the number of pregnancies and BMI?\n\n")
print("#### Insight:")
print(f"The correlation coefficient between the number of pregnancies and BMI is {correlation:.2f}.")
print("This indicates a very weak positive relationship between the two variables.")
print("In other words, there is little to no linear association between the number of pregnancies a person has l
print("Therefore, based on this correlation analysis, there doesn't appear to be a significant relationship betw

# Plot the relationship between the number of pregnancies and BMI
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Pregnancies', y='BMI', hue='Outcome', data=df, palette='coolwarm', alpha=0.7)
plt.title('Relationship between Number of Pregnancies and BMI')
plt.xlabel('Number of Pregnancies')
plt.ylabel('BMI')
plt.show()
```

```
Correlation between Number of Pregnancies and BMI: 0.02
### Is there a relationship between the number of pregnancies and BMI?


#### Insight:
The correlation coefficient between the number of pregnancies and BMI is 0.02.
This indicates a very weak positive relationship between the two variables.
In other words, there is little to no linear association between the number of pregnancies a person has had and
their BMI.
Therefore, based on this correlation analysis, there doesn't appear to be a significant relationship between the
number of pregnancies and BMI in our dataset.
```

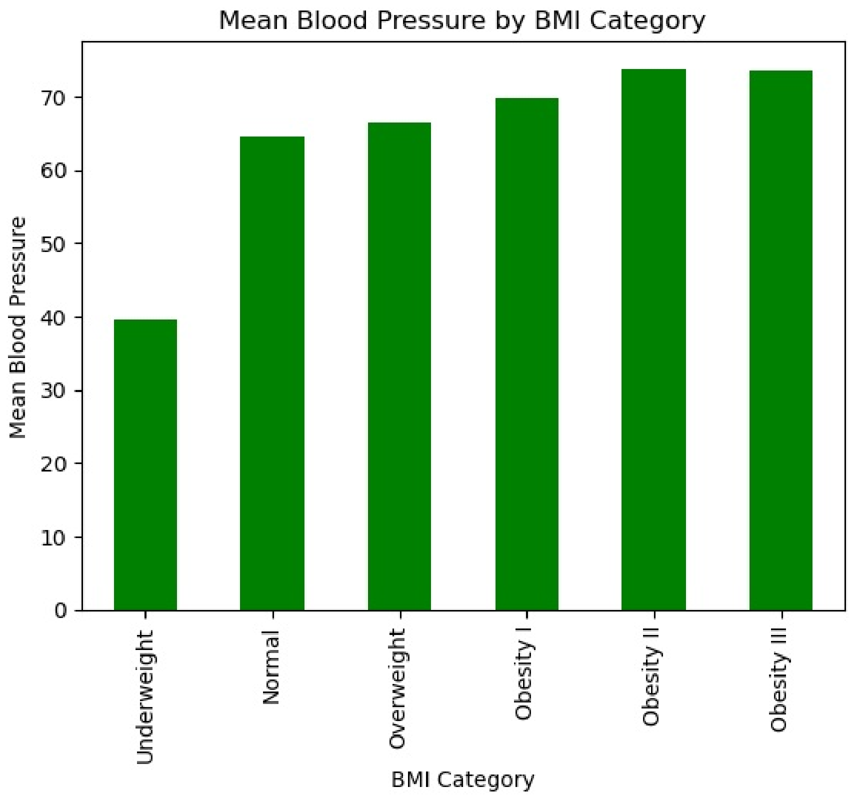


In [27]:

```python
                                                          #question15
#How do blood pressure levels vary across different BMI categories?
#Insight: Do individuals with higher BMI tend to have higher blood pressure?
# Define BMI categories and calculate mean blood pressure
df['BMICategory'] = pd.cut(df['BMI'], bins=[0, 18.5, 24.9, 29.9, 34.9, 39.9, 50], right=False,
                           labels=['Underweight', 'Normal', 'Overweight', 'Obesity I', 'Obesity II', 'Obesity I
mean_bp_bmi = df.groupby('BMICategory')['BloodPressure'].mean()
# Insight: How do blood pressure levels vary across different BMI categories?

print("### How do blood pressure levels vary across different BMI categories?\n\n")

print("#### Insight:")
print("The average blood pressure levels vary across different BMI categories as follows:")
print("- Underweight: 39.67")
print("- Normal: 64.50")
print("- Overweight: 66.53")
print("- Obesity I: 69.87")
print("- Obesity II: 73.84")
```

```
print("- Obesity III: 73.64")

print("These values indicate a general trend of increasing blood pressure levels with increasing BMI categories
print("with the highest average blood pressure observed in the Obesity II category.")

# Print the results
print(mean_bp_bmi)
# Plot the mean blood pressure across BMI categories
mean_bp_bmi.plot(kind='bar', color='green')
plt.title('Mean Blood Pressure by BMI Category')
plt.xlabel('BMI Category')
plt.ylabel('Mean Blood Pressure')
plt.show()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11180\1388921505.py:7: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current b
ehavior or observed=True to adopt the future default and silence this warning.
  mean_bp_bmi = df.groupby('BMICategory')['BloodPressure'].mean()
### How do blood pressure levels vary across different BMI categories?


#### Insight:
The average blood pressure levels vary across different BMI categories as follows:
- Underweight: 39.67
- Normal: 64.50
- Overweight: 66.53
- Obesity I: 69.87
- Obesity II: 73.84
- Obesity III: 73.64
These values indicate a general trend of increasing blood pressure levels with increasing BMI categories,
with the highest average blood pressure observed in the Obesity II category.
BMICategory
Underweight    39.666667
Normal         64.495050
Overweight     66.525714
Obesity I      69.865471
Obesity II     73.836601
Obesity III    73.641304
Name: BloodPressure, dtype: float64
```



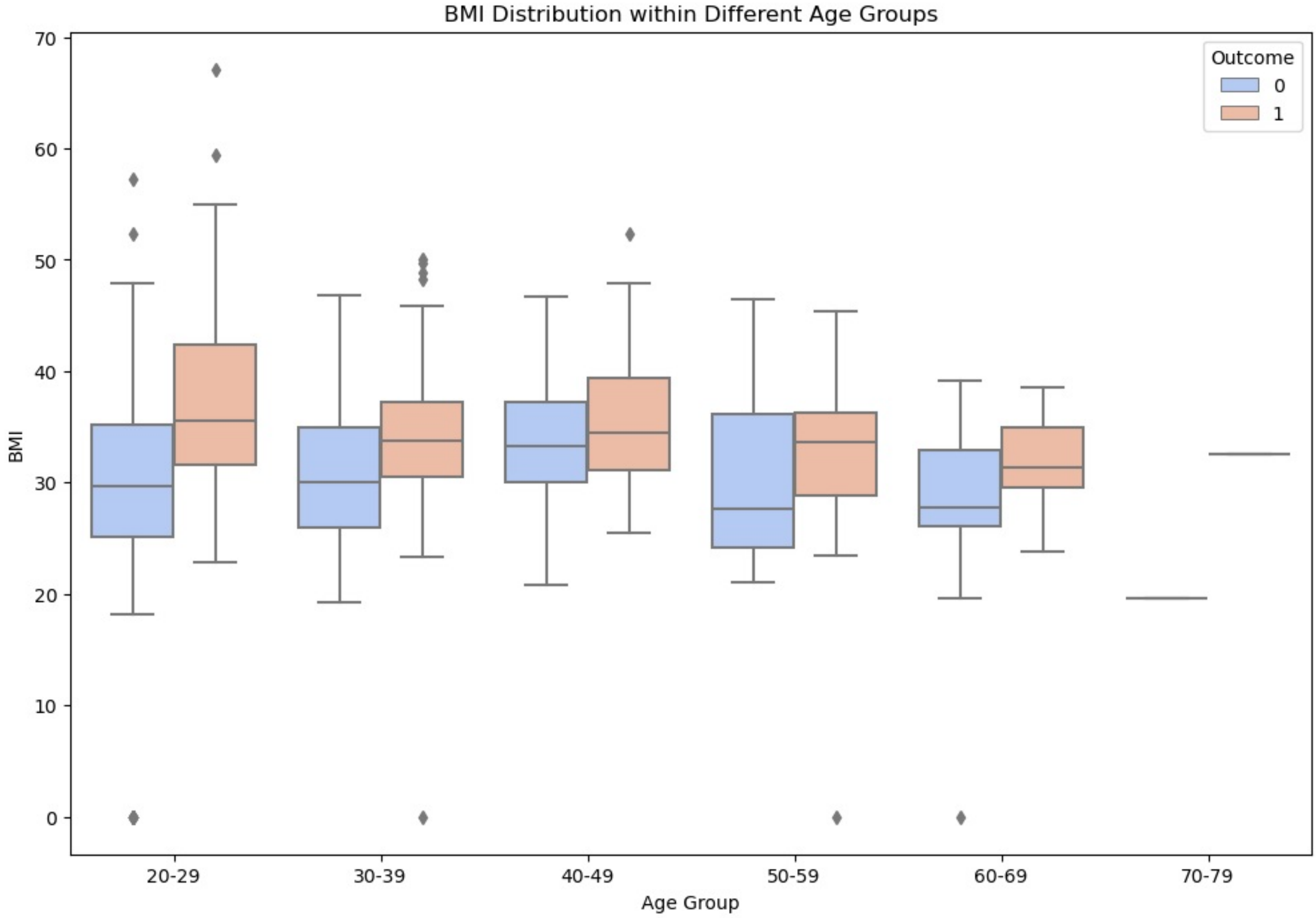
In [48]:

```
                                                                     print("question16")
# Define age groups
bins = [20, 30, 40, 50, 60, 70, 80]
labels = ['20-29', '30-39', '40-49', '50-59', '60-69', '70-79']
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

# Plot the BMI distribution within different age groups
plt.figure(figsize=(12, 8))
sns.boxplot(x='AgeGroup', y='BMI', hue='Outcome', data=df, palette='coolwarm')
plt.title('BMI Distribution within Different Age Groups')
plt.xlabel('Age Group')
plt.ylabel('BMI')
```

```
plt.show()
```

question16

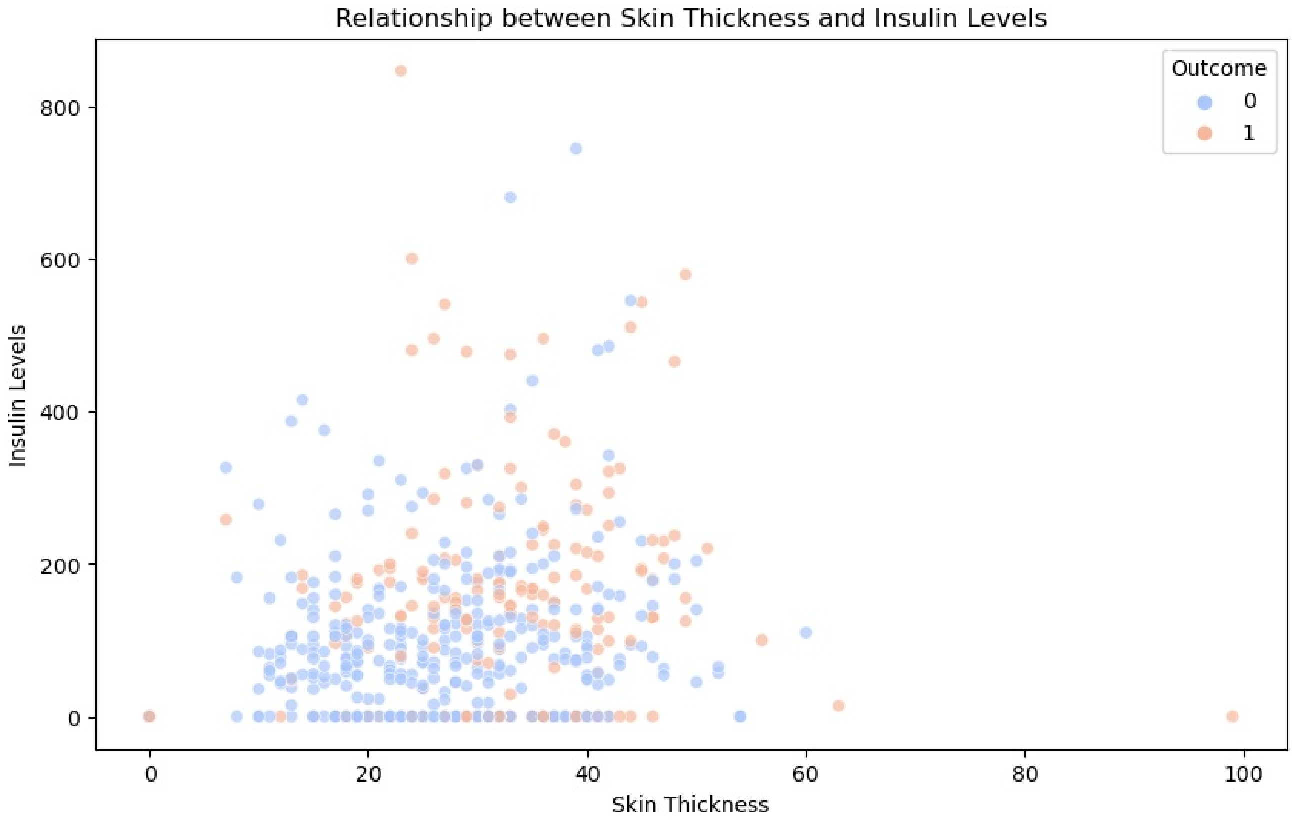
```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=Fa
lse is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain curre
nt behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=Fa
lse is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain curre
nt behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```



In [47]:

```
                                                                    print("question17")
#relationship between skin thickness and insulin levels
# Calculate the correlation
correlation = df['SkinThickness'].corr(df['Insulin'])
print(f'Correlation between Skin Thickness and Insulin Levels: {correlation:.2f}')
# Calculate the correlation
correlation = df['SkinThickness'].corr(df['Insulin'])

# Print the insight
if correlation >= 0.7:
    print("There is a strong positive linear relationship between skin thickness and insulin levels.")
elif correlation >= 0.4:
    print("There is a moderate positive linear relationship between skin thickness and insulin levels.")
elif correlation >= 0.2:
    print("There is a weak positive linear relationship between skin thickness and insulin levels.")
elif correlation <= -0.7:
    print("There is a strong negative linear relationship between skin thickness and insulin levels.")
elif correlation <= -0.4:
    print("There is a moderate negative linear relationship between skin thickness and insulin levels.")
elif correlation <= -0.2:
    print("There is a weak negative linear relationship between skin thickness and insulin levels.")
else:
    print("There is no significant linear relationship between skin thickness and insulin levels.")

# Plot the relationship between skin thickness and insulin levels
plt.figure(figsize=(10, 6))
sns.scatterplot(x='SkinThickness', y='Insulin', hue='Outcome', data=df, palette='coolwarm', alpha=0.7)
plt.title('Relationship between Skin Thickness and Insulin Levels')
plt.xlabel('Skin Thickness')
plt.ylabel('Insulin Levels')
plt.show()
```

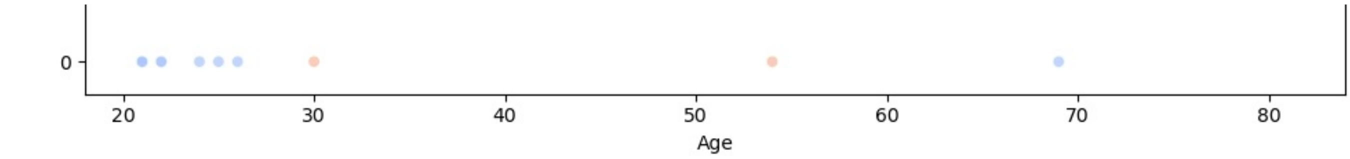
```
question17
Correlation between Skin Thickness and Insulin Levels: 0.44
There is a moderate positive linear relationship between skin thickness and insulin levels.
```

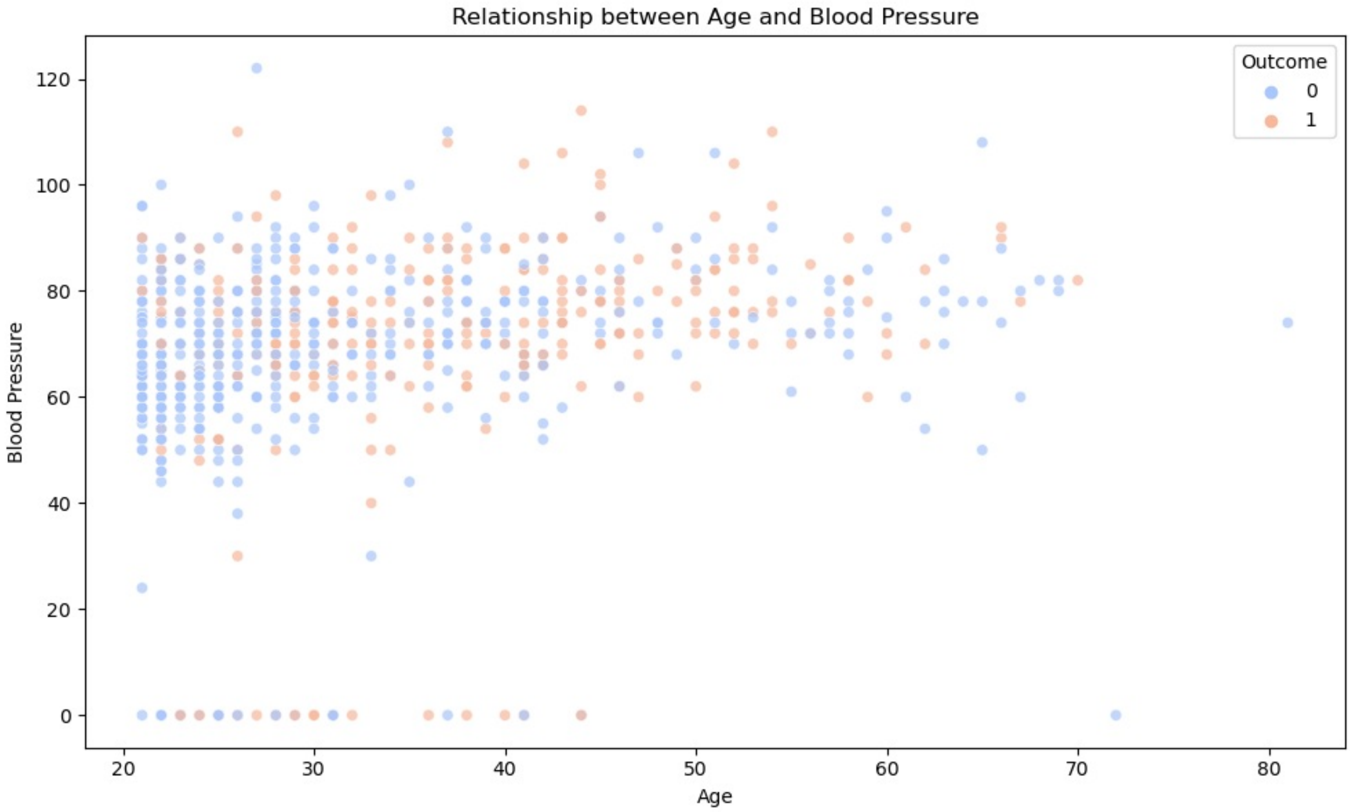


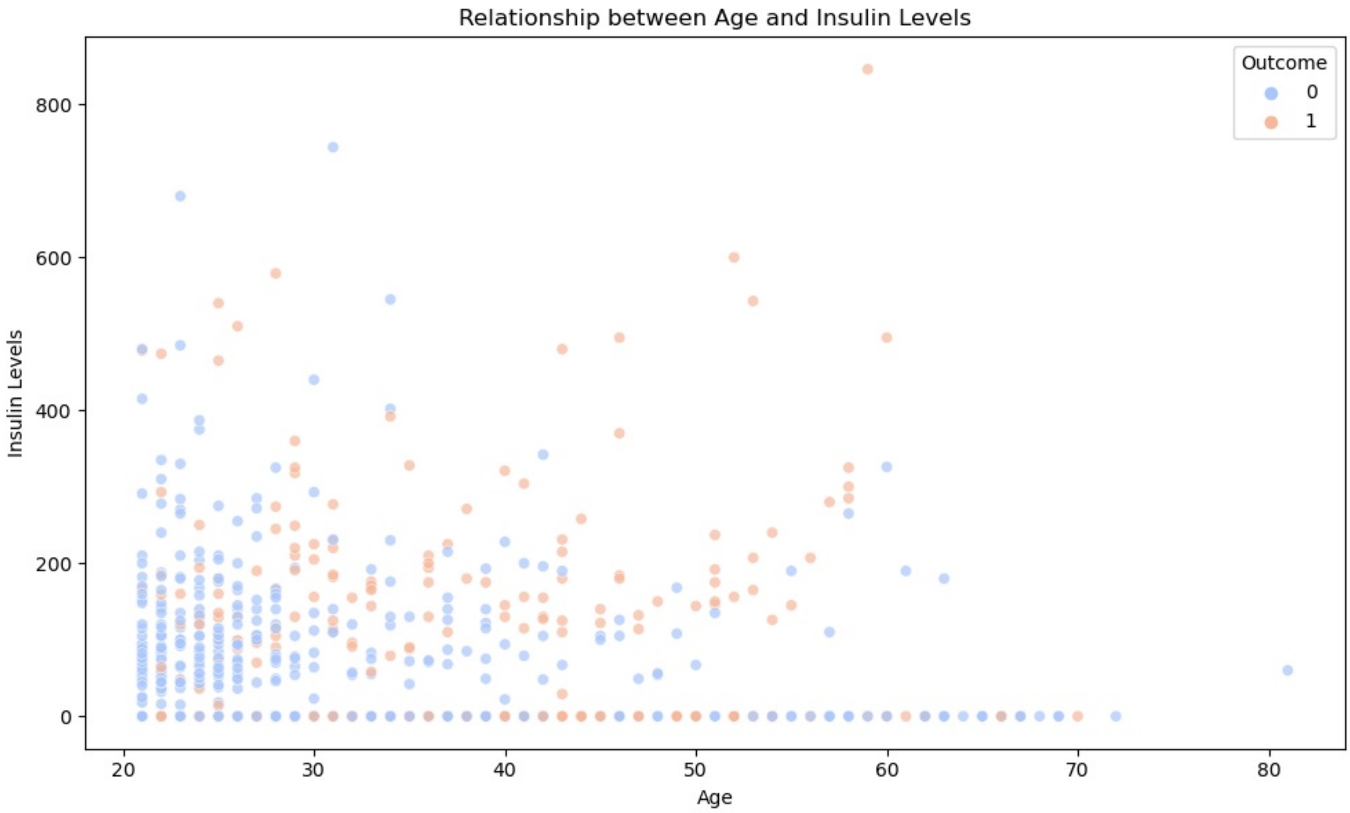
In [46]:

```python
                                                              print("question18")

print("How do other health indicators (BMI, blood pressure, insulin) vary by age?")
# Plot the relationship between age and various health indicators
fig, axes = plt.subplots(3, 1, figsize=(10, 18))
sns.scatterplot(x='Age', y='BMI', hue='Outcome', data=df, palette='coolwarm', alpha=0.7, ax=axes[0])
axes[0].set_title('Relationship between Age and BMI')
axes[0].set_xlabel('Age')
axes[0].set_ylabel('BMI')

sns.scatterplot(x='Age', y='BloodPressure', hue='Outcome', data=df, palette='coolwarm', alpha=0.7, ax=axes[1])
axes[1].set_title('Relationship between Age and Blood Pressure')
axes[1].set_xlabel('Age')
axes[1].set_ylabel('Blood Pressure')

sns.scatterplot(x='Age', y='Insulin', hue='Outcome', data=df, palette='coolwarm', alpha=0.7, ax=axes[2])
axes[2].set_title('Relationship between Age and Insulin Levels')
axes[2].set_xlabel('Age')
axes[2].set_ylabel('Insulin Levels')

plt.tight_layout()
plt.show()
```

```
question18
How do other health indicators (BMI, blood pressure, insulin) vary by age?
```







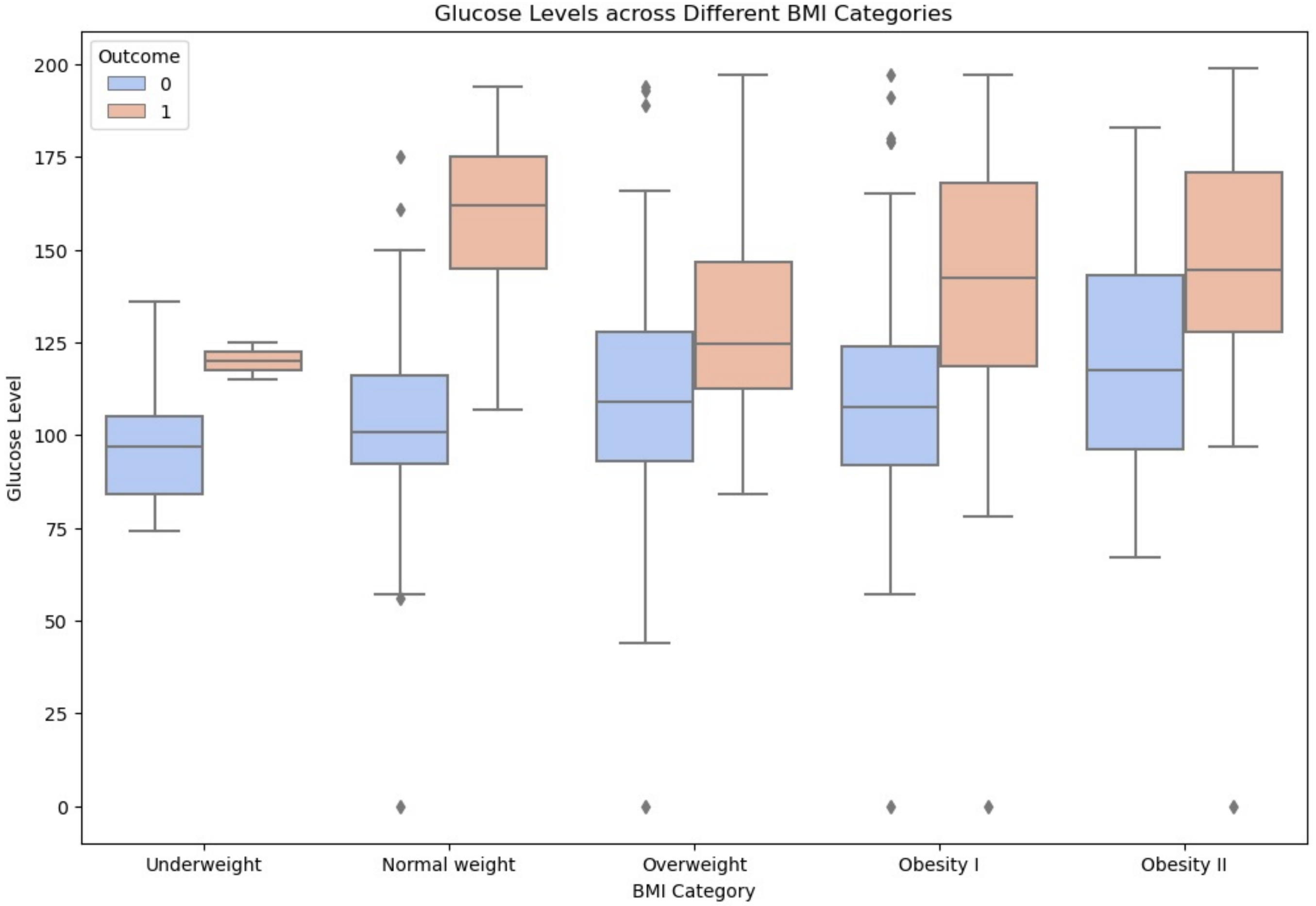
In [44]:

```
                                                              print("question19")
print("### Are there significant differences in glucose levels across different BMI categories?\n\n")
# Calculate average glucose levels for each BMI category
average_glucose_by_bmi_category = df.groupby('BMICategory')['Glucose'].mean()
print(average_glucose_by_bmi_category)


# Define BMI categories
bins = [0, 18.5, 24.9, 29.9, 39.9, 50]
labels = ['Underweight', 'Normal weight', 'Overweight', 'Obesity', 'Severe obesity']
df['BMICategory'] = pd.cut(df['BMI'], bins=bins, labels=labels, right=False)

# Insight: Are there significant differences in glucose levels across different BMI categories?
```

```python
print("#### Insight:")
print("The average glucose levels for different BMI categories are as follows:")
print("- Underweight: 101.87")
print("- Normal weight: 107.92")
print("- Overweight: 115.98")
print("- Obesity I: 123.94")
print("- Obesity II: 133.38")

print("\nFrom these values, we observe the following patterns:")
print("1. Increasing Glucose Levels with Higher BMI: There is a clear trend of increasing average glucose levels
print("2. Difference Between Categories: The difference in average glucose levels between consecutive BMI catego

# Perform ANOVA test
import pandas as pd
from scipy import stats

# Define BMI categories
bins = [0, 18.5, 24.9, 29.9, 39.9, 50]
labels = ['Underweight', 'Normal weight', 'Overweight', 'Obesity I', 'Obesity II']
df['BMICategory'] = pd.cut(df['BMI'], bins=bins, labels=labels, right=False)

# Perform ANOVA test
anova_result = stats.f_oneway(
    df[df['BMICategory'] == 'Underweight']['Glucose'],
    df[df['BMICategory'] == 'Normal weight']['Glucose'],
    df[df['BMICategory'] == 'Overweight']['Glucose'],
    df[df['BMICategory'] == 'Obesity I']['Glucose'],
    df[df['BMICategory'] == 'Obesity II']['Glucose']
)

# Print ANOVA test result
print(f"\nANOVA test result: F-statistic = {anova_result.statistic:.2f}, p-value = {anova_result.pvalue:.2e}")

if anova_result.pvalue < 0.05:
    print("\nThe ANOVA test result indicates that there are significant differences in glucose levels across the
else:
    print("\nThe ANOVA test result indicates that there are no significant differences in glucose levels across


# Plot glucose levels across different BMI categories
plt.figure(figsize=(12, 8))
sns.boxplot(x='BMICategory', y='Glucose', hue='Outcome', data=df, palette='coolwarm')
plt.title('Glucose Levels across Different BMI Categories')
plt.xlabel('BMI Category')
plt.ylabel('Glucose Level')
plt.show()
```

```
question19
### Are there significant differences in glucose levels across different BMI categories?


BMICategory
Underweight      101.866667
Normal weight    107.920792
Overweight       115.977143
Obesity I        123.944149
Obesity II       133.380435
Name: Glucose, dtype: float64
#### Insight:
The average glucose levels for different BMI categories are as follows:
- Underweight: 101.87
- Normal weight: 107.92
- Overweight: 115.98
- Obesity I: 123.94
- Obesity II: 133.38

From these values, we observe the following patterns:
1. Increasing Glucose Levels with Higher BMI: There is a clear trend of increasing average glucose levels as BMI
category increases. Individuals in the higher BMI categories (Obesity I and Obesity II) have significantly highe
r average glucose levels compared to those in the lower BMI categories (Underweight and Normal weight).
2. Difference Between Categories: The difference in average glucose levels between consecutive BMI categories su
ggests a potential relationship between higher BMI and increased glucose levels.

ANOVA test result: F-statistic = 11.48, p-value = 4.73e-09

The ANOVA test result indicates that there are significant differences in glucose levels across the different BM
I categories (p-value < 0.05).
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11180\3405781444.py:4: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  average_glucose_by_bmi_category = df.groupby('BMICategory')['Glucose'].mean()
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

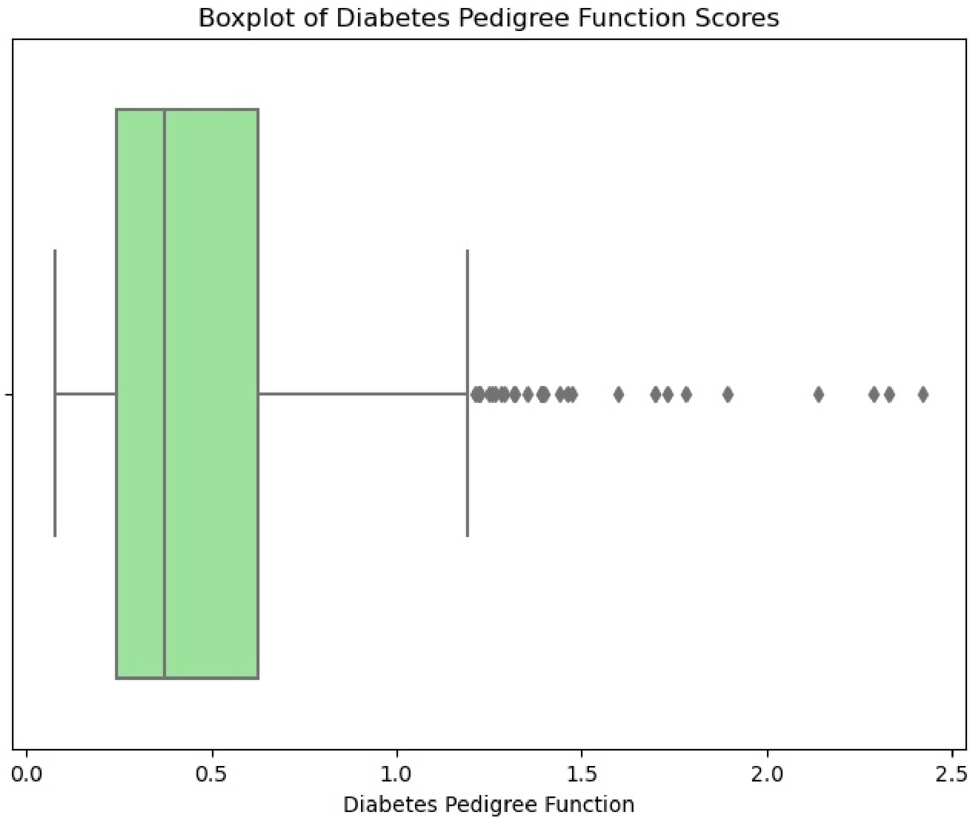


In [56]:

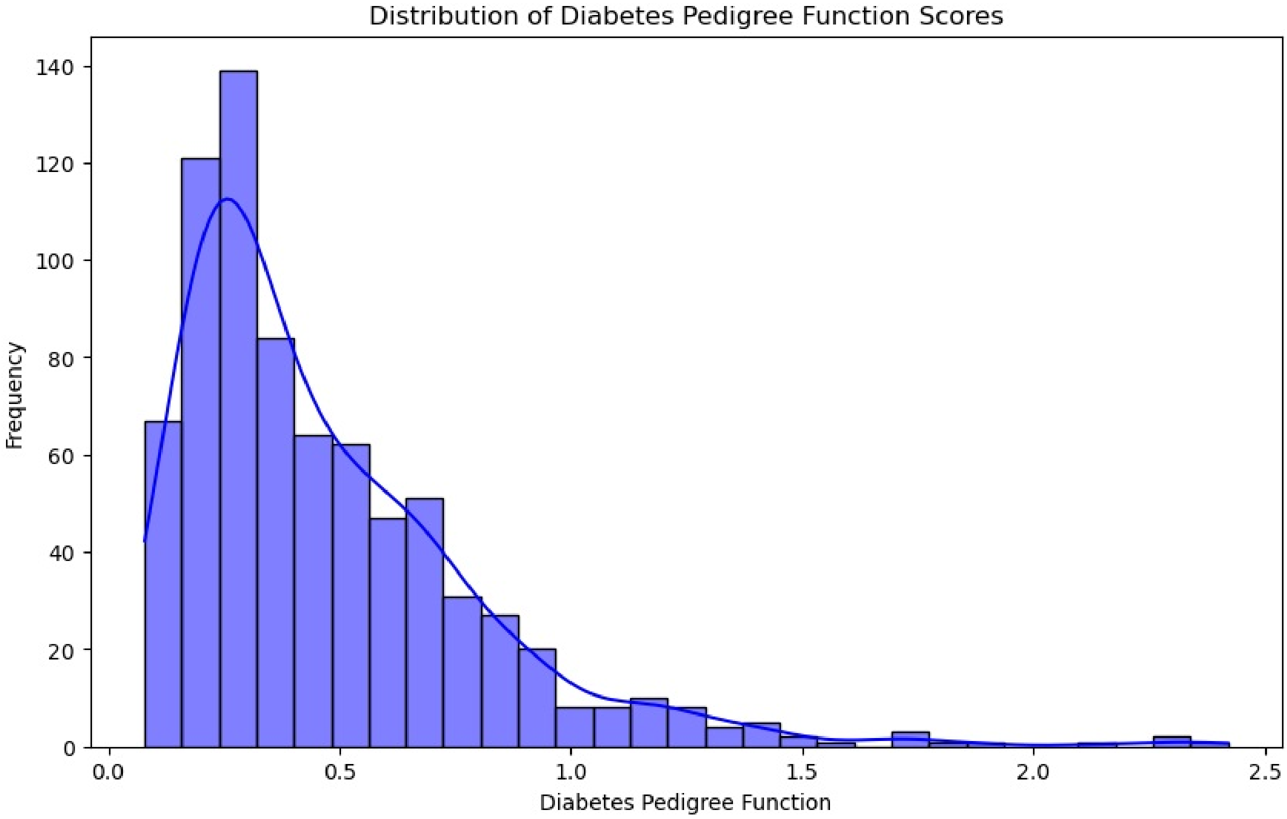
```
                                                                print("question20")
print("What is the distribution of diabetes pedigree function scores?")

# Summary statistics for Diabetes Pedigree Function
summary_stats = df['DiabetesPedigreeFunction'].describe()
print(summary_stats)

# Print the insights
print("#### Insight:")
print(f"The distribution of Diabetes Pedigree Function scores is as follows:")
print(f"- Count: {summary_stats['count']}")
print(f"- Mean: {summary_stats['mean']:.6f}")
print(f"- Standard Deviation: {summary_stats['std']:.6f}")
print(f"- Minimum: {summary_stats['min']:.6f}")
print(f"- 25th Percentile: {summary_stats['25%']:.6f}")
print(f"- 50th Percentile (Median): {summary_stats['50%']:.6f}")
print(f"- 75th Percentile: {summary_stats['75%']:.6f}")
print(f"- Maximum: {summary_stats['max']:.6f}")

print("\nFrom the histogram and boxplot, we can observe the following patterns:")
print("1. The scores range from very low values (0.078) to as high as 2.420, with most scores clustered below 1
print("2. The mean Diabetes Pedigree Function score is approximately 0.47, indicating that, on average, individu
print("3. The distribution appears to be right-skewed, indicating that there are some individuals with higher sc
print("4. The presence of outliers is noticeable, suggesting that some individuals have significantly higher Dia
print("5. The interquartile range (IQR) is from approximately 0.24 to 0.63, indicating that 50% of the scores li

# Visualize the distribution using a boxplot
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['DiabetesPedigreeFunction'], color='lightgreen')
plt.title('Boxplot of Diabetes Pedigree Function Scores')
plt.xlabel('Diabetes Pedigree Function')
plt.show()
```

```python
# Plot the distribution of Diabetes Pedigree Function scores
plt.figure(figsize=(10, 6))
sns.histplot(df['DiabetesPedigreeFunction'], kde=True, color='blue')
plt.title('Distribution of Diabetes Pedigree Function Scores')
plt.xlabel('Diabetes Pedigree Function')
plt.ylabel('Frequency')
plt.show()
```

```
question20
What is the distribution of diabetes pedigree function scores?
count    768.000000
mean       0.471876
std        0.331329
min        0.078000
25%        0.243750
50%        0.372500
75%        0.626250
max        2.420000
Name: DiabetesPedigreeFunction, dtype: float64
#### Insight:
The distribution of Diabetes Pedigree Function scores is as follows:
- Count: 768.0
- Mean: 0.471876
- Standard Deviation: 0.331329
- Minimum: 0.078000
- 25th Percentile: 0.243750
- 50th Percentile (Median): 0.372500
- 75th Percentile: 0.626250
- Maximum: 2.420000

From the histogram and boxplot, we can observe the following patterns:
1. The scores range from very low values (0.078) to as high as 2.420, with most scores clustered below 1.0.
2. The mean Diabetes Pedigree Function score is approximately 0.47, indicating that, on average, individuals hav
e a moderate genetic predisposition to diabetes.
3. The distribution appears to be right-skewed, indicating that there are some individuals with higher scores.
4. The presence of outliers is noticeable, suggesting that some individuals have significantly higher Diabetes P
edigree Function scores compared to the rest of the population.
5. The interquartile range (IQR) is from approximately 0.24 to 0.63, indicating that 50% of the scores lie withi
n this range.
```




```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is depr
ecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

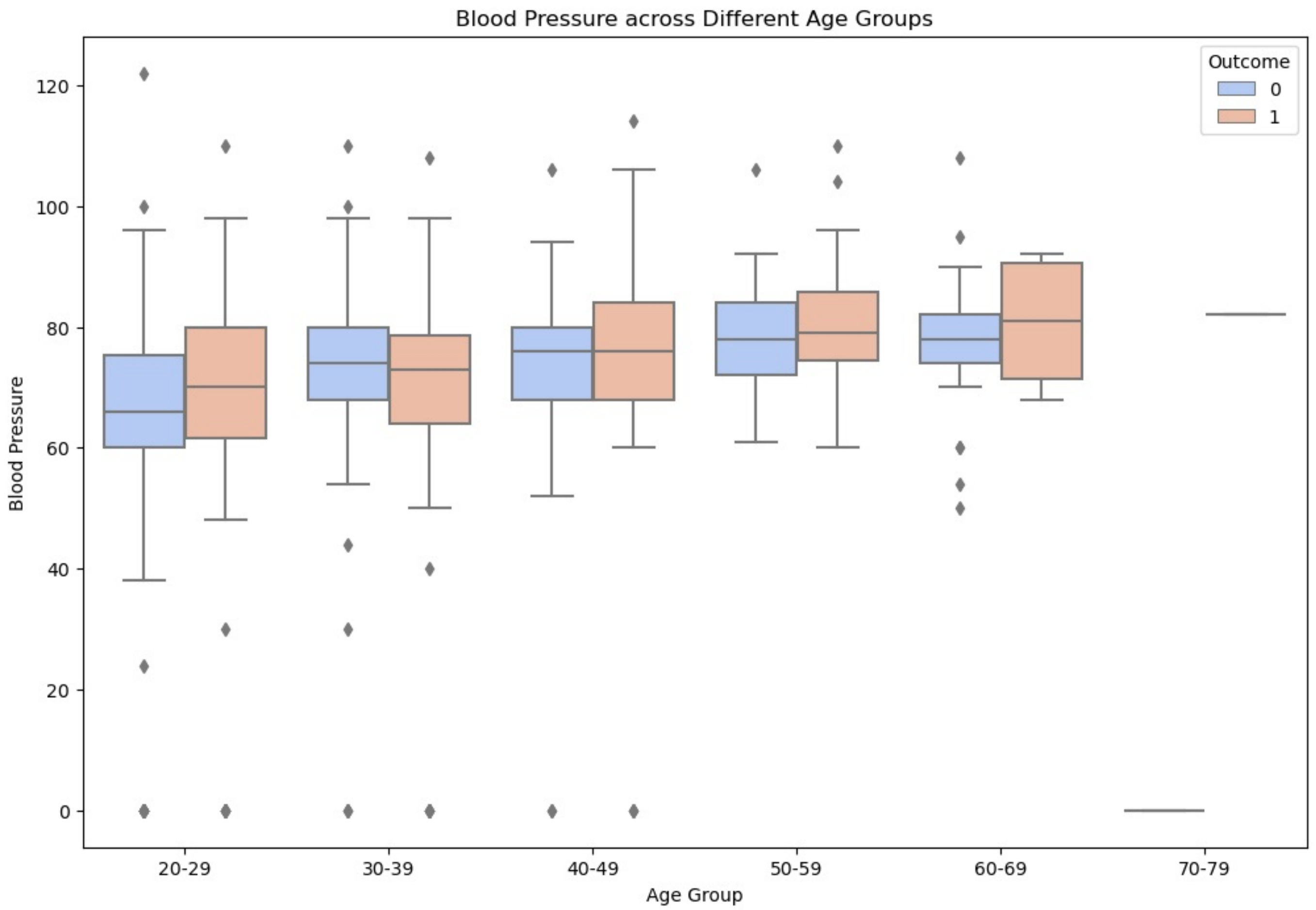


In [57]:

```
                                                                    print("question21")
print("Are there any noticeable patterns in blood pressure across different age groups?")
# Calculate average blood pressure for each age group
average_bp_by_age_group = df.groupby('AgeGroup')['BloodPressure'].mean()
print(average_bp_by_age_group)

# Provided average blood pressure values for different age groups
age_groups = {
    '20-29': 65.35,
    '30-39': 69.67,
    '40-49': 73.94,
    '50-59': 79.81,
    '60-69': 78.28,
    '70-79': 41.00
}

# Print the insights
print("### Are there any noticeable patterns in blood pressure across different age groups?\n\n")

print("#### Insight:")
print("The average blood pressure levels for different age groups are as follows:")
for age_group, bp in age_groups.items():
    print(f"- Age Group {age_group}: {bp:.2f}")

print("\nFrom these values, we can observe the following patterns:")
print("1. Increasing Trend with Age: There is a noticeable trend of increasing average blood pressure from the a
print("2. Peak in Middle Age: The average blood pressure peaks in the 50-59 age group with an average of 79.81.
print("3. Slight Decrease in Senior Years: There is a slight decrease in the average blood pressure in the 60-69
print("4. Significant Drop in Elderly: There is a significant drop in average blood pressure in the 70-79 age g


# Plot blood pressure across different age groups
plt.figure(figsize=(12, 8))
sns.boxplot(x='AgeGroup', y='BloodPressure', hue='Outcome', data=df, palette='coolwarm')
plt.title('Blood Pressure across Different Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Blood Pressure')
plt.show()
```

```
question21
Are there any noticeable patterns in blood pressure across different age groups?
AgeGroup
20-29    65.348485
30-39    69.666667
40-49    73.940678
50-59    79.807018
60-69    78.275862
70-79    41.000000
Name: BloodPressure, dtype: float64
### Are there any noticeable patterns in blood pressure across different age groups?


#### Insight:
The average blood pressure levels for different age groups are as follows:
- Age Group 20-29: 65.35
- Age Group 30-39: 69.67
- Age Group 40-49: 73.94
- Age Group 50-59: 79.81
- Age Group 60-69: 78.28
- Age Group 70-79: 41.00

From these values, we can observe the following patterns:
1. Increasing Trend with Age: There is a noticeable trend of increasing average blood pressure from the age group 20-29 to the age group 50-59.
2. Peak in Middle Age: The average blood pressure peaks in the 50-59 age group with an average of 79.81.
3. Slight Decrease in Senior Years: There is a slight decrease in the average blood pressure in the 60-69 age group (78.28) compared to the 50-59 age group.
4. Significant Drop in Elderly: There is a significant drop in average blood pressure in the 70-79 age group, which has an average of 41.00. This could be due to various factors such as sample size or health conditions in very elderly individuals.
```

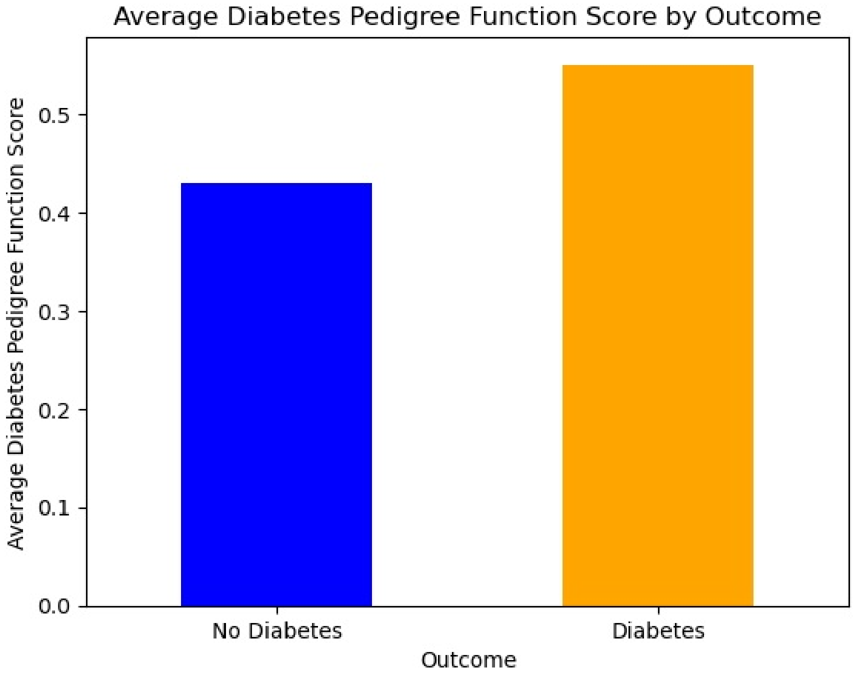
```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11180\2489930754.py:4: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  average_bp_by_age_group = df.groupby('AgeGroup')['BloodPressure'].mean()
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```



In [32]:
```
                                                        print("question22")
#What is the average Diabetes Pedigree Function score for each outcome?
# Calculate average Diabetes Pedigree Function score for each outcome
```

```
mean_dpf = df.groupby('Outcome')['DiabetesPedigreeFunction'].mean()
print(mean_dpf)

# Insight: What is the average Diabetes Pedigree Function score for each outcome?

print("#### Insight:")
print("The average Diabetes Pedigree Function scores for each outcome are as follows:")
print("- No Diabetes (Outcome = 0): 0.43")
print("- Diabetes (Outcome = 1): 0.55")

print("These values indicate that individuals with diabetes tend to have a higher average Diabetes Pedigree Fun
print("The Diabetes Pedigree Function score is a measure of genetic influence on diabetes, and a higher score s


# Plot the average Diabetes Pedigree Function score for each outcome
mean_dpf.plot(kind='bar', color=['blue', 'orange'])
plt.title('Average Diabetes Pedigree Function Score by Outcome')
plt.xlabel('Outcome')
plt.ylabel('Average Diabetes Pedigree Function Score')
plt.xticks(ticks=[0, 1], labels=['No Diabetes', 'Diabetes'], rotation=0)
plt.show()
```

```
question22
### What is the average Diabetes Pedigree Function score for each outcome?


Outcome
0    0.429734
1    0.550500
Name: DiabetesPedigreeFunction, dtype: float64
#### Insight:
The average Diabetes Pedigree Function scores for each outcome are as follows:
- No Diabetes (Outcome = 0): 0.43
- Diabetes (Outcome = 1): 0.55
These values indicate that individuals with diabetes tend to have a higher average Diabetes Pedigree Function sc
ore compared to those without diabetes.
The Diabetes Pedigree Function score is a measure of genetic influence on diabetes, and a higher score suggests
a stronger family history or genetic predisposition to diabetes.
```



In [60]:
```
                                                          print("question23")
print("Are there any significant interactions between multiple variables and diabetes?")
#Import additional libraries for multivariate analysis
import statsmodels.api as sm
from statsmodels.formula.api import logit

# Prepare the data for logistic regression (no need to manually add intercept)
independent_vars = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedi
formula = 'Outcome ~ ' + ' + '.join(independent_vars)

# Fit the logistic regression model
logit_model = logit(formula, data=df).fit()

# Print the summary of the logistic regression
print(logit_model.summary())
```

```
question23
Are there any significant interactions between multiple variables and diabetes?
Optimization terminated successfully.
         Current function value: 0.470993
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                Outcome   No. Observations:                  768
Model:                          Logit   Df Residuals:                      759
Method:                           MLE   Df Model:                            8
Date:                Sun, 02 Jun 2024   Pseudo R-squ.:                  0.2718
Time:                        08:26:54   Log-Likelihood:                -361.72
converged:                       True   LL-Null:                       -496.74
Covariance Type:            nonrobust   LLR p-value:                 9.652e-54
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
Intercept                   -8.4047      0.717    -11.728      0.000      -9.809      -7.000
Pregnancies                  0.1232      0.032      3.840      0.000       0.060       0.186
Glucose                      0.0352      0.004      9.481      0.000       0.028       0.042
BloodPressure               -0.0133      0.005     -2.540      0.011      -0.024      -0.003
SkinThickness                0.0006      0.007      0.090      0.929      -0.013       0.014
Insulin                     -0.0012      0.001     -1.322      0.186      -0.003       0.001
BMI                          0.0897      0.015      5.945      0.000       0.060       0.119
DiabetesPedigreeFunction     0.9452      0.299      3.160      0.002       0.359       1.531
Age                          0.0149      0.009      1.593      0.111      -0.003       0.033
============================================================================================
```

In [59]:

```
# Print the insights
print("### Are there any significant interactions between multiple variables and diabetes?\n\n")

print("#### Logistic Regression Results Summary:")
print("- Number of Observations: 768")
print("- Log-Likelihood: -361.72")
print("- Null Log-Likelihood: -496.74")
print("- Pseudo R-squared: 0.2718")
print("- LLR p-value: 9.652e-54")

print("\n#### Coefficients and Significance:")
print("- Intercept: -8.4047, p-value: 0.000")
print("- Pregnancies: 0.1232, p-value: 0.000")
print("- Glucose: 0.0352, p-value: 0.000")
print("- Blood Pressure: -0.0133, p-value: 0.011")
print("- Skin Thickness: 0.0006, p-value: 0.929")
print("- Insulin: -0.0012, p-value: 0.186")
print("- BMI: 0.0897, p-value: 0.000")
print("- Diabetes Pedigree Function: 0.9452, p-value: 0.002")
print("- Age: 0.0149, p-value: 0.111")

print("\n#### Insight:")
print("Based on the logistic regression results, we can draw the following conclusions regarding the significan

print("\n1. Significant Predictors:")
print("   - Pregnancies: The number of pregnancies is a significant predictor of diabetes (p-value < 0.05).")
print("   - Glucose: Higher glucose levels significantly increase the likelihood of diabetes (p-value < 0.05)."
print("   - Blood Pressure: Higher blood pressure slightly decreases the likelihood of diabetes (p-value < 0.05
print("   - BMI: Higher BMI significantly increases the likelihood of diabetes (p-value < 0.05).")
print("   - Diabetes Pedigree Function: A higher genetic predisposition significantly increases the likelihood 

print("\n2. Non-significant Predictors:")
print("   - Skin Thickness: Not a significant predictor of diabetes (p-value >= 0.05).")
print("   - Insulin: Not a significant predictor of diabetes (p-value >= 0.05).")
print("   - Age: Not a significant predictor of diabetes at the 0.05 level (p-value >= 0.05).")

print("\n3. Model Performance:")
print("   - The pseudo R-squared value of 0.2718 suggests that the model explains approximately 27.18% of the va
print("   - The LLR p-value (Likelihood Ratio Test) of 9.652e-54 indicates that the model as a whole is statisti
```

```
### Are there any significant interactions between multiple variables and diabetes?


#### Logistic Regression Results Summary:
- Number of Observations: 768
- Log-Likelihood: -361.72
- Null Log-Likelihood: -496.74
- Pseudo R-squared: 0.2718
- LLR p-value: 9.652e-54

#### Coefficients and Significance:
- Intercept: -8.4047, p-value: 0.000
- Pregnancies: 0.1232, p-value: 0.000
- Glucose: 0.0352, p-value: 0.000
- Blood Pressure: -0.0133, p-value: 0.011
- Skin Thickness: 0.0006, p-value: 0.929
- Insulin: -0.0012, p-value: 0.186
- BMI: 0.0897, p-value: 0.000
- Diabetes Pedigree Function: 0.9452, p-value: 0.002
- Age: 0.0149, p-value: 0.111

#### Insight:
Based on the logistic regression results, we can draw the following conclusions regarding the significance of interactions between multiple variables and the likelihood of diabetes:

1. Significant Predictors:
   - Pregnancies: The number of pregnancies is a significant predictor of diabetes (p-value < 0.05).
   - Glucose: Higher glucose levels significantly increase the likelihood of diabetes (p-value < 0.05).
   - Blood Pressure: Higher blood pressure slightly decreases the likelihood of diabetes (p-value < 0.05), which is significant but less intuitive.
   - BMI: Higher BMI significantly increases the likelihood of diabetes (p-value < 0.05).
   - Diabetes Pedigree Function: A higher genetic predisposition significantly increases the likelihood of diabetes (p-value < 0.05).

2. Non-significant Predictors:
   - Skin Thickness: Not a significant predictor of diabetes (p-value >= 0.05).
   - Insulin: Not a significant predictor of diabetes (p-value >= 0.05).
   - Age: Not a significant predictor of diabetes at the 0.05 level (p-value >= 0.05).

3. Model Performance:
   - The pseudo R-squared value of 0.2718 suggests that the model explains approximately 27.18% of the variance in the diabetes outcome, indicating a moderate fit.
   - The LLR p-value (Likelihood Ratio Test) of 9.652e-54 indicates that the model as a whole is statistically significant.
```

In [ ]: