In [1]:

```python
#import all the libraries that we need.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```python
#importing our dataset.
from google.colab import drive
drive.mount('/content/drive')

data_path = '/content/drive/My Drive/diabetes_data.csv'  # Replace with your file path
df = pd.read_csv(data_path)
```

```
Mounted at /content/drive
```

In [3]:

```python
#Checking first five rows by calling df.head()
df.head()
```

Out[3]:

	PatientID	Age	Gender	Ethnicity	SocioeconomicStatus	EducationLevel	BMI	Smoking	AlcoholConsu
0	6000	44	0	1	2	1	32.985284	1	4.
1	6001	51	1	0	1	2	39.916764	0	1.
2	6002	89	1	0	1	3	19.782251	0	1.
3	6003	21	1	1	1	2	32.376881	1	1.
4	6004	27	1	0	1	3	16.808600	0	15.

5 rows × 46 columns

In [4]:

```python
df.tail()
```

Out[4]:

	PatientID	Age	Gender	Ethnicity	SocioeconomicStatus	EducationLevel	BMI	Smoking	AlcoholCo
1874	7874	37	0	0	2	2	20.811137	0	
1875	7875	80	1	0	2	2	27.694312	0	
1876	7876	38	1	0	0	2	35.640824	0	
1877	7877	43	0	1	2	0	32.423016	0	
1878	7878	85	1	0	2	2	33.145119	0	

5 rows × 46 columns

In [5]:

```python
#Take a look at the column names.
df.columns.values
```

Out[5]:

```
array(['PatientID', 'Age', 'Gender', 'Ethnicity', 'SocioeconomicStatus',
       'EducationLevel', 'BMI', 'Smoking', 'AlcoholConsumption',
       'PhysicalActivity', 'DietQuality', 'SleepQuality',
       'FamilyHistoryDiabetes', 'GestationalDiabetes',
       'PolycysticOvarySyndrome', 'PreviousPreDiabetes', 'Hypertension',
       'SystolicBP', 'DiastolicBP', 'FastingBloodSugar', 'HbA1c',
       'SerumCreatinine', 'BUNLevels', 'CholesterolTotal',
       'CholesterolLDL', 'CholesterolHDL', 'CholesterolTriglycerides',
       'AntihypertensiveMedications', 'Statins',
       'AntidiabeticMedications', 'FrequentUrination', 'ExcessiveThirst',
       'UnexplainedWeightLoss', 'FatigueLevels', 'BlurredVision',
       'SlowHealingSores', 'TinglingHandsFeet', 'QualityOfLifeScore',
        'HeavyMetalsExposure', 'OccupationalExposureChemicals',
        'WaterQuality', 'MedicalCheckupsFrequency', 'MedicationAdherence',
        'HealthLiteracy', 'Diagnosis', 'DoctorInCharge'], dtype=object)
```

In [7]:

```python
#Checking for null values
df.isna().sum()
```

Out[7]:

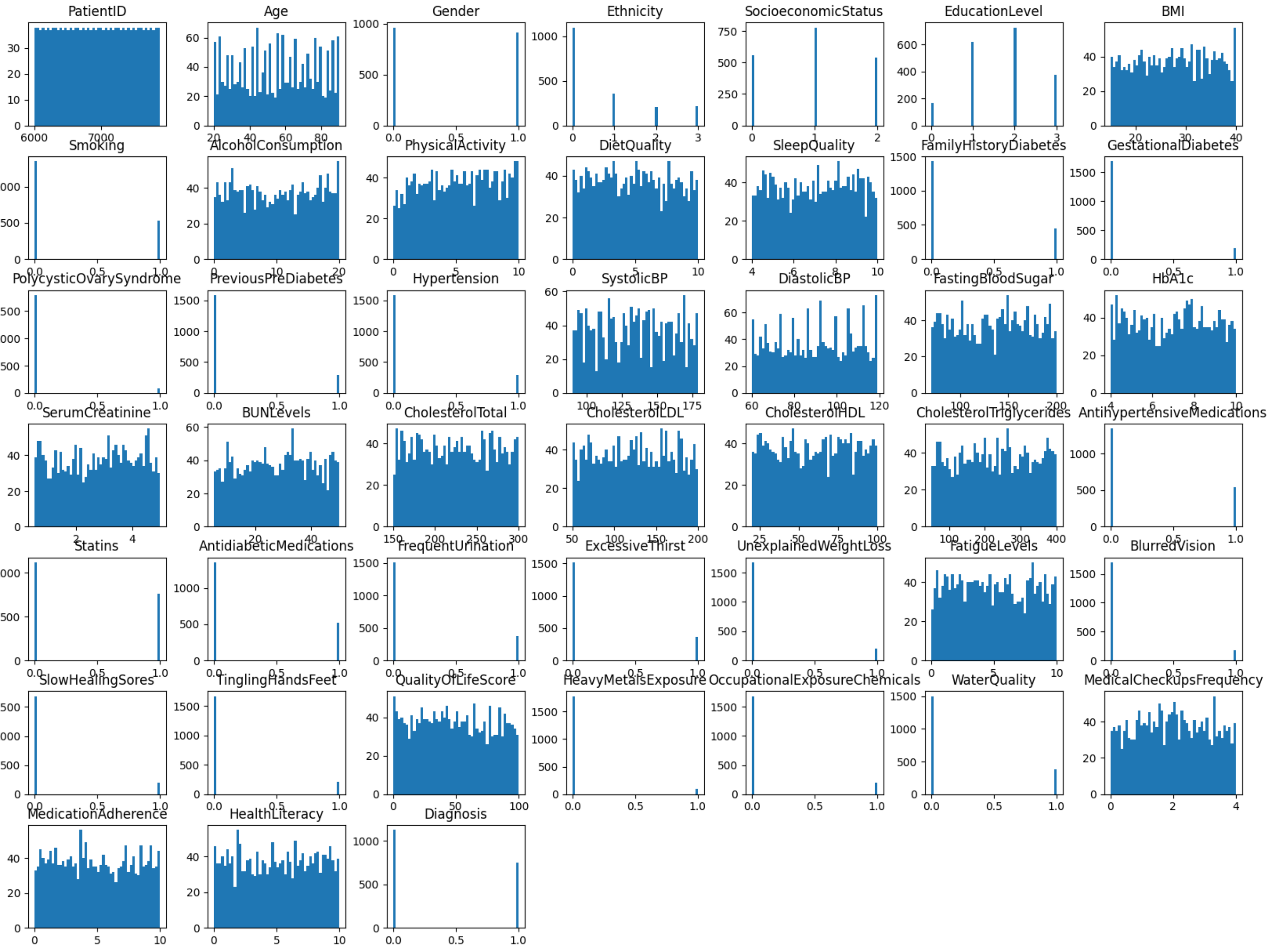
	0
PatientID	0
Age	0
Gender	0
Ethnicity	0
SocioeconomicStatus	0
EducationLevel	0
BMI	0
Smoking	0
AlcoholConsumption	0
PhysicalActivity	0
DietQuality	0
SleepQuality	0
FamilyHistoryDiabetes	0
GestationalDiabetes	0
PolycysticOvarySyndrome	0
PreviousPreDiabetes	0
Hypertension	0
SystolicBP	0
DiastolicBP	0
FastingBloodSugar	0
HbA1c	0
SerumCreatinine	0
BUNLevels	0
CholesterolTotal	0
CholesterolLDL	0
CholesterolHDL	0
CholesterolTriglycerides	0
AntihypertensiveMedications	0
Statins	0
AntidiabeticMedications	0
FrequentUrination	0
ExcessiveThirst	0
UnexplainedWeightLoss	0
FatigueLevels	0
BlurredVision	0

SlowHealingSores	0
TinglingHandsFeet	0
QualityOfLifeScore	0
HeavyMetalsExposure	0
OccupationalExposureChemicals	0
WaterQuality	0
MedicalCheckupsFrequency	0
MedicationAdherence	0
HealthLiteracy	0
Diagnosis	0
DoctorInCharge	0

dtype: int64

In [8]:

```
#plotting histogram of all numeric values
df.hist(bins = 50, grid = False ,figsize=(20,15) );
```



In [9]:

```
#Generating descriptive statistics.
df.describe()
```

Out[9]:

	PatientID	Age	Gender	Ethnicity	SocioeconomicStatus	EducationLevel	BMI
count	1879.000000	1879.000000	1879.000000	1879.000000	1879.000000	1879.000000	1879.000000
mean	6939.000000	55.043108	0.487493	0.755721	0.992017	1.699308	27.687601

std	542.564896	20.515839	0.499977	1.047558	0.764940	0.885665	7.190975
min	6000.000000	20.000000	0.000000	0.000000	0.000000	0.000000	15.025898
25%	6469.500000	38.000000	0.000000	0.000000	0.000000	1.000000	21.469981
50%	6939.000000	55.000000	0.000000	0.000000	1.000000	2.000000	27.722988
75%	7408.500000	73.000000	1.000000	1.000000	2.000000	2.000000	33.856460
max	7878.000000	90.000000	1.000000	3.000000	2.000000	3.000000	39.998811

8 rows × 45 columns

In [61]:

```
questions = ["1. How does the distribution of BloodPressure vary between individuals wit
             "2. How does the prevalence of the 'Diagnosis' (assuming this indicates a d
             "3. Is there a relationship between Fasting Blood Sugar and lifestyle facto
             "4. How are patients distributed across different education levels?",
             "5. What is the distribution of Quality of Life Scores among the patients?"
             "6. How does the distribution of Diabeties values vary for individuals with
             "7. How are fasting blood sugar levels distributed in the overall populatio

]

questions
```

Out[61]:

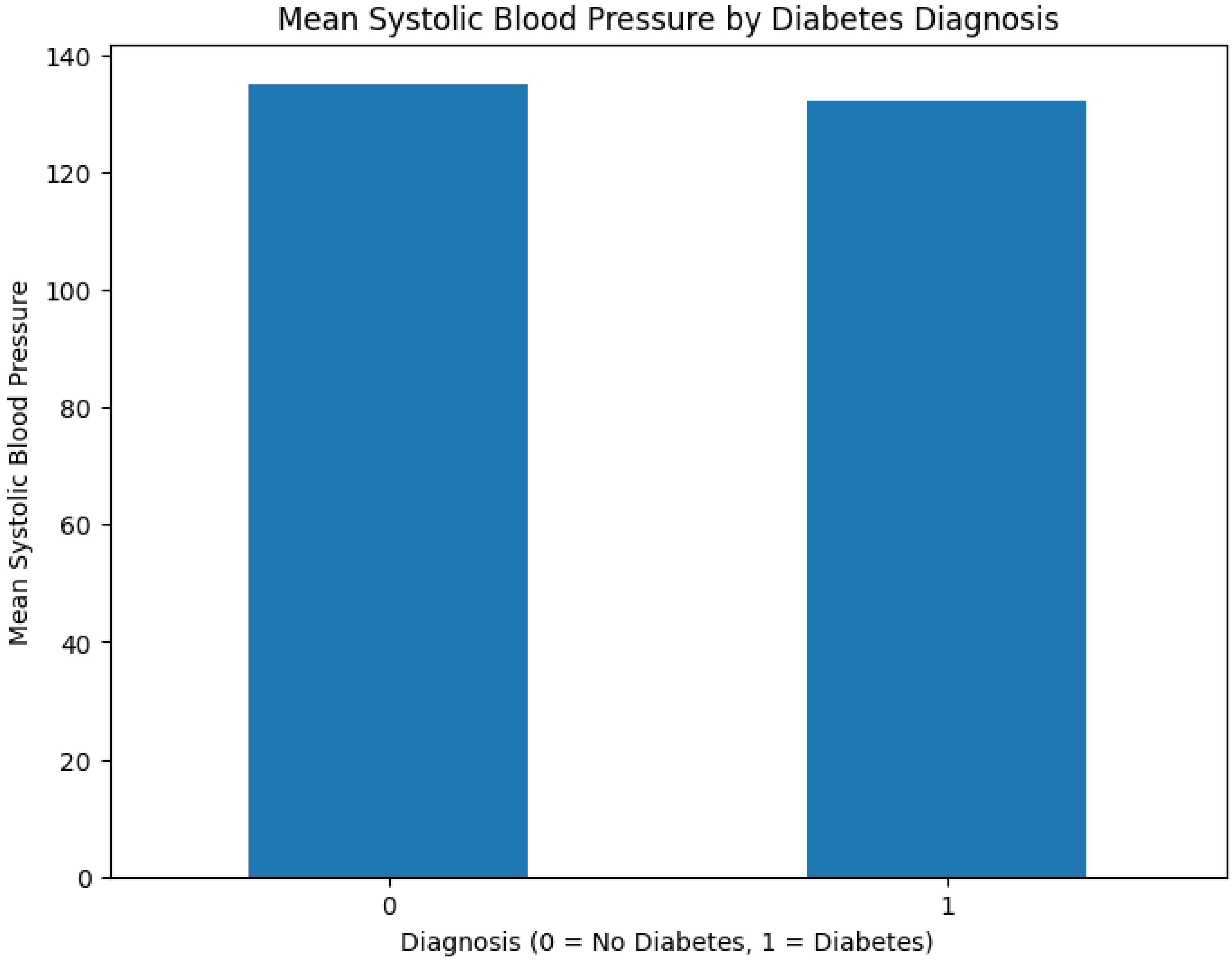
```
['1. How does the distribution of BloodPressure vary between individuals with and withou
t diabetes?',
 "2. How does the prevalence of the 'Diagnosis' (assuming this indicates a diabetes diag
nosis) differ between males and females?",
 '3. Is there a relationship between Fasting Blood Sugar and lifestyle factors (Smoking,
Alcohol Consumption, Physical Activity, Diet Quality)?',
 '4. How are patients distributed across different education levels?',
 '5. What is the distribution of Quality of Life Scores among the patients?',
 '6. How does the distribution of Diabeties values vary for individuals with and without
a family history of diabetes?',
 '7. How are fasting blood sugar levels distributed in the overall population?']
```

In [11]:

```
#Let's check the 1st question

#1.How does the distribution of BloodPressure vary between individuals with and without
```

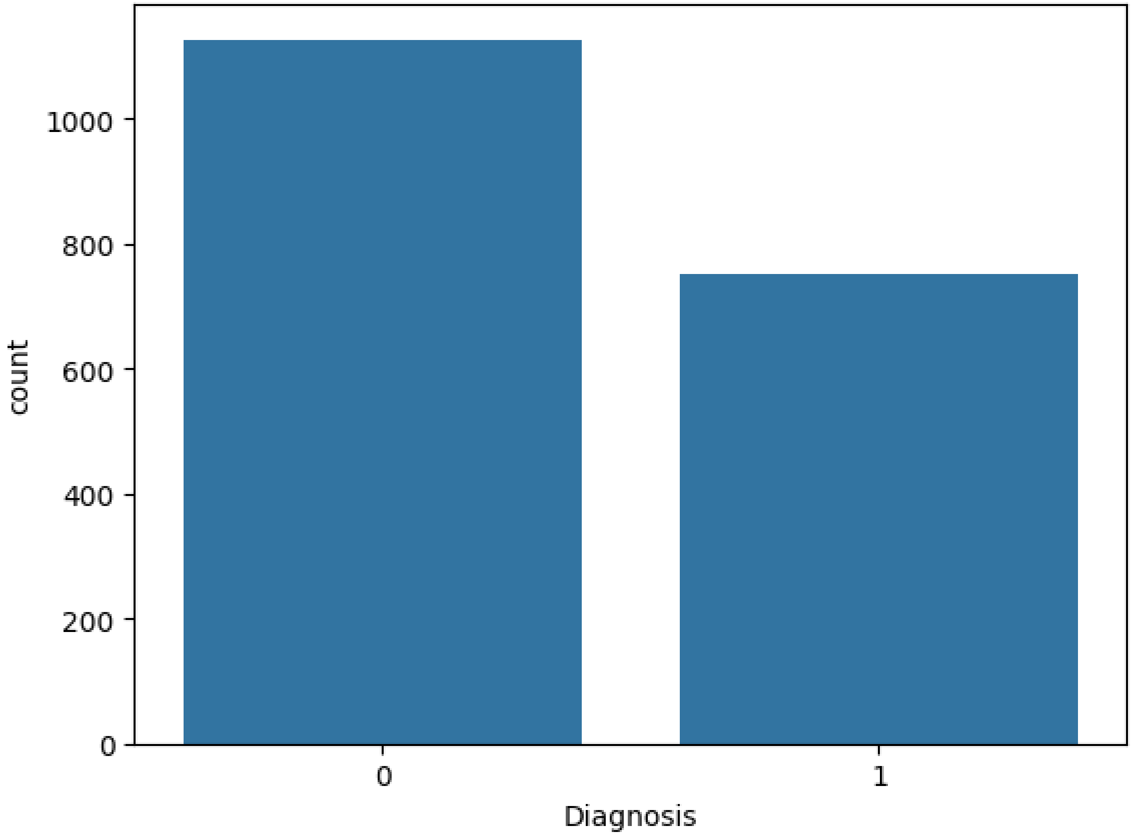
In [12]:

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

# Calculate mean blood pressure for each group
mean_systolic = df.groupby('Diagnosis')['SystolicBP'].mean()

# Create bar plot
plt.figure(figsize=(8, 6))
mean_systolic.plot(kind='bar')
plt.title('Mean Systolic Blood Pressure by Diabetes Diagnosis')
plt.xlabel('Diagnosis (0 = No Diabetes, 1 = Diabetes)')
plt.ylabel('Mean Systolic Blood Pressure')
plt.xticks(rotation=0)  # Rotate x-axis labels for readability
plt.show()
```



In [13]:

```
sns.countplot(x = 'Diagnosis', data = df)
# blood pressure varience with and without diebeties
plt.show()
```

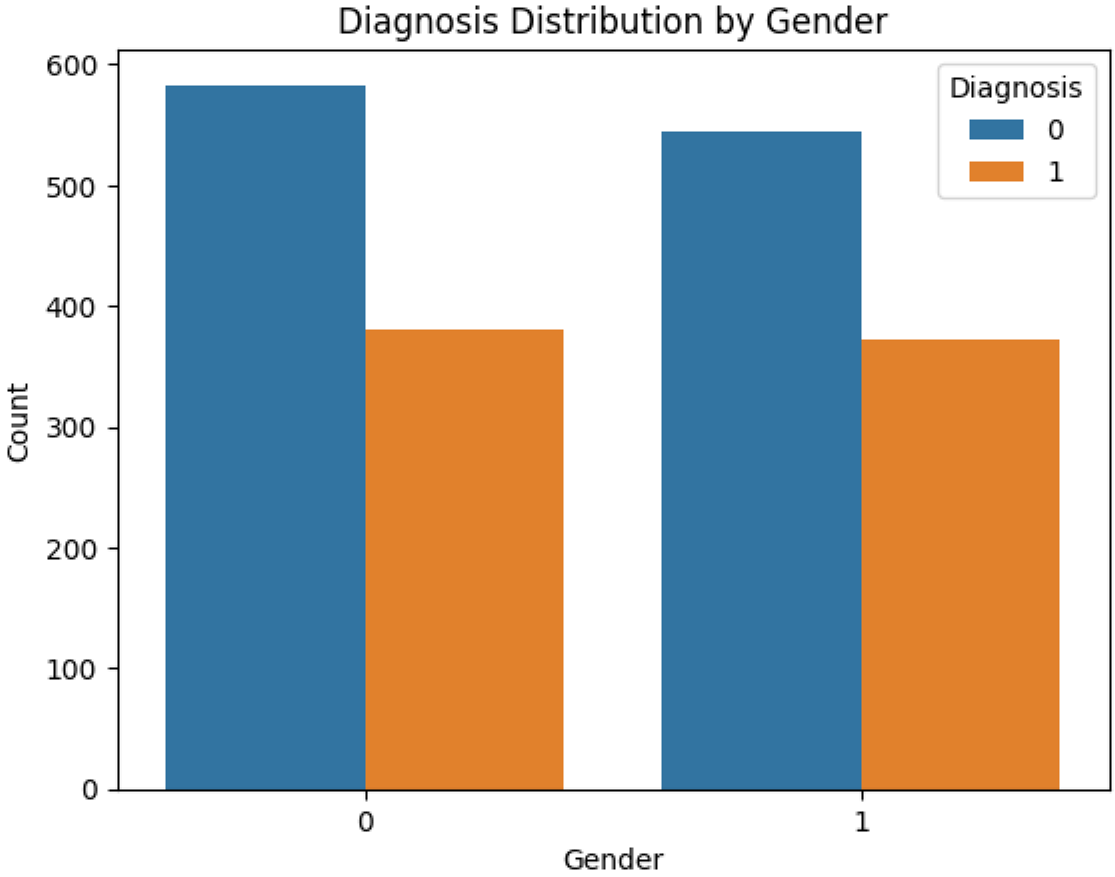


In [ ]:

```
#lets see the next question no 2

#How does the prevalence of the 'Diagnosis' (assuming this indicates a diabetes diagnosi
```

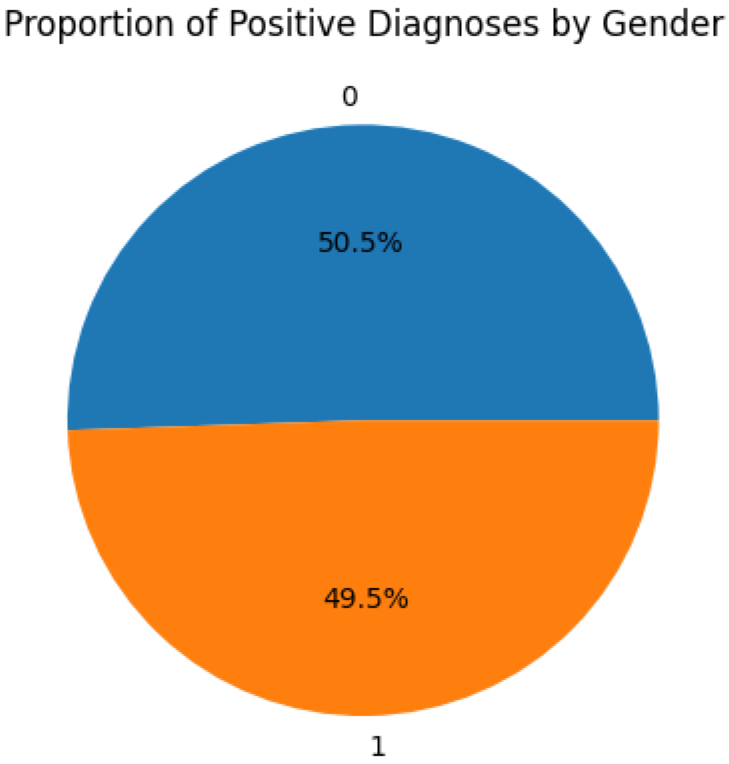
In [20]:

```
sns.countplot(x='Gender', hue='Diagnosis', data=df)
plt.title('Diagnosis Distribution by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



In [23]:

```
gender_diagnosis = df.groupby('Gender')['Diagnosis'].sum()
plt.pie(gender_diagnosis, labels=gender_diagnosis.index, autopct='%1.1f%%')
plt.title('Proportion of Positive Diagnoses by Gender')
plt.show()
```



In [ ]:

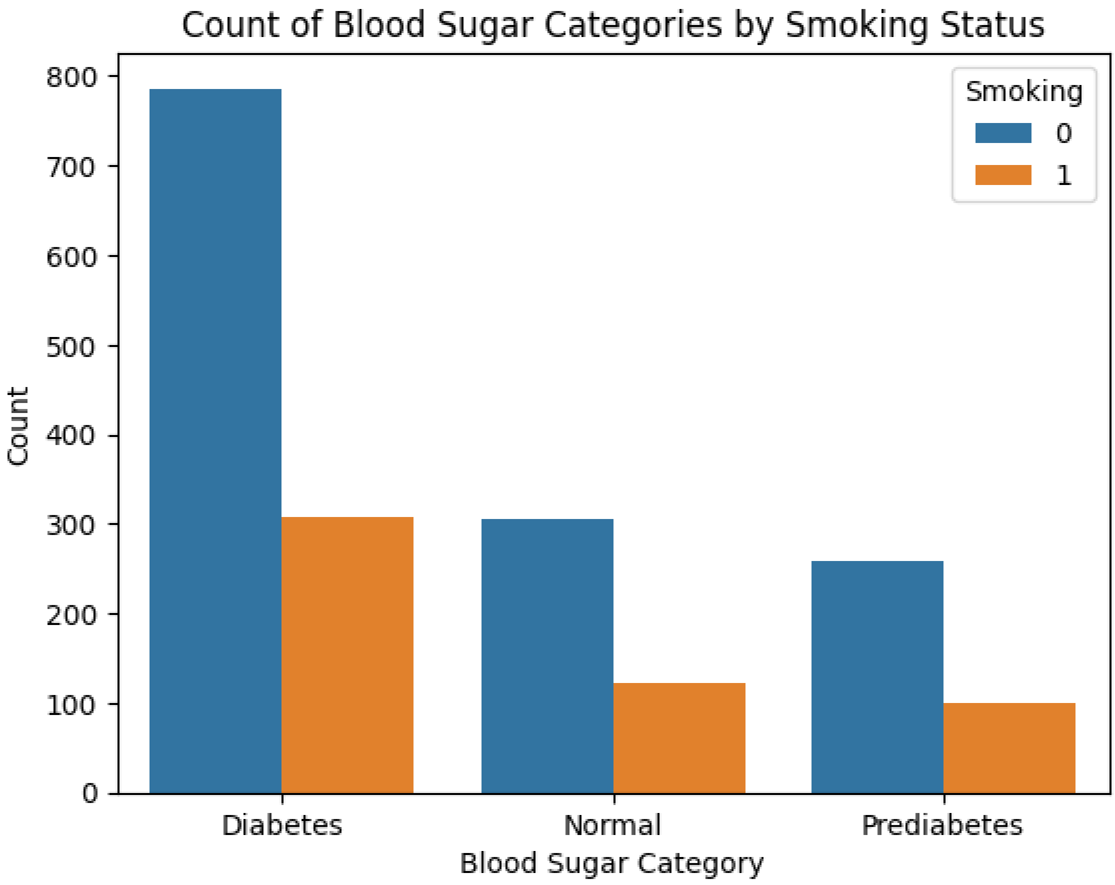
```
#let's see question 3

#Is there a relationship between Fasting Blood Sugar and lifestyle factors (Smoking, Alc
```

In [ ]:

```
def categorize_blood_sugar(value):
  if value < 100:
    return 'Normal'
  elif value < 126:
    return 'Prediabetes'
  else:
    return 'Diabetes'

df['BloodSugarCategory'] = df['FastingBloodSugar'].apply(categorize_blood_sugar)
```

In [36]:

```
sns.countplot(x='BloodSugarCategory', hue='Smoking', data=df)
plt.title('Count of Blood Sugar Categories by Smoking Status')
plt.xlabel('Blood Sugar Category')
plt.ylabel('Count')
plt.show()
```

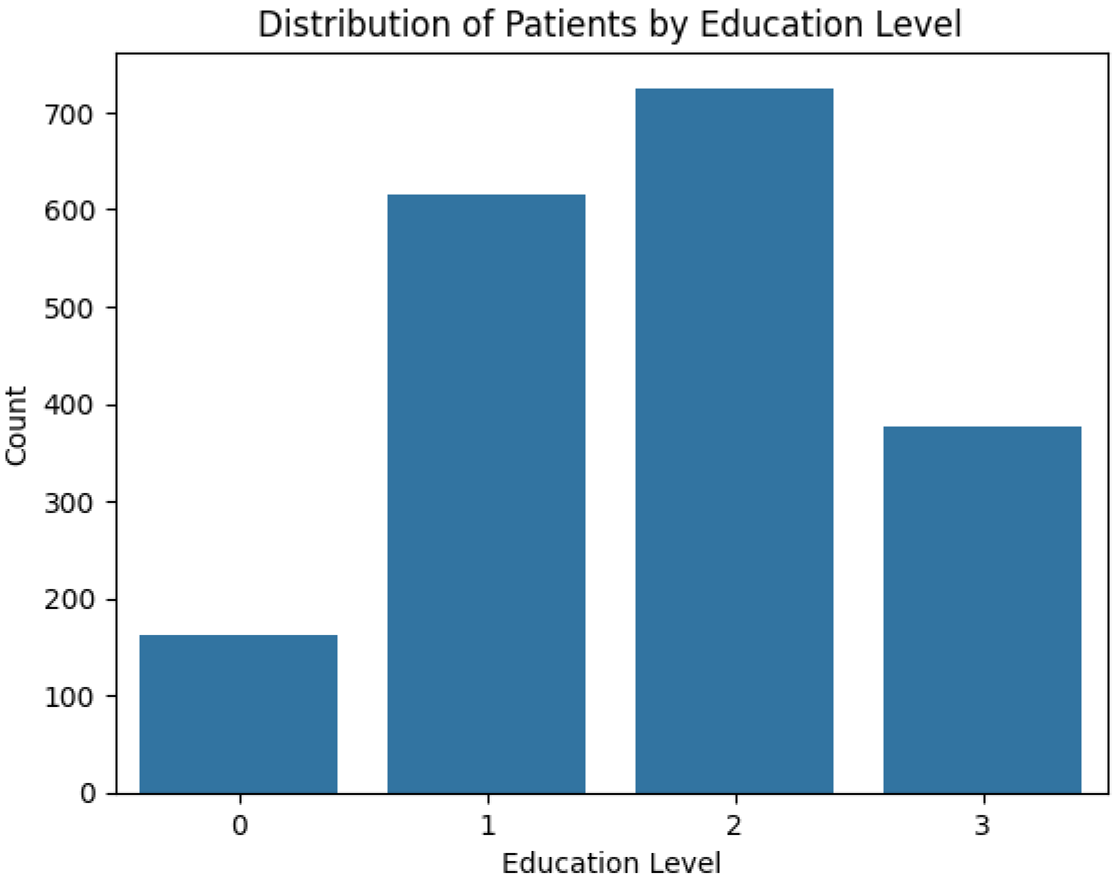


In [ ]:

```
#let's see question 4

#How are patients distributed across different education levels?
```

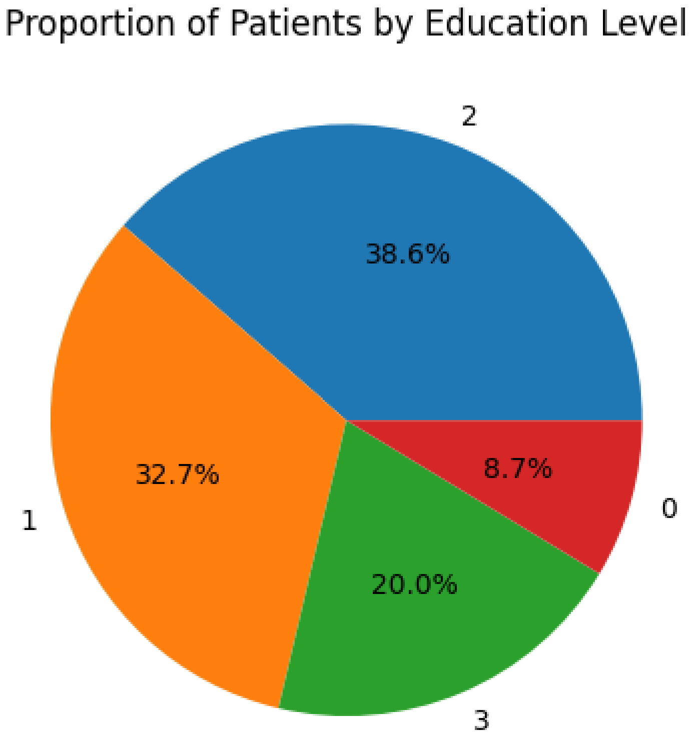
In [27]:

```
sns.countplot(x='EducationLevel', data=df)
plt.title('Distribution of Patients by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.show()
```



In [40]:

```
# Count the occurrences of each education level
education_counts = df['EducationLevel'].value_counts()

# Plot the pie chart
plt.pie(education_counts, labels=education_counts.index, autopct='%1.1f%%')
plt.title('Proportion of Patients by Education Level')
plt.show()
```

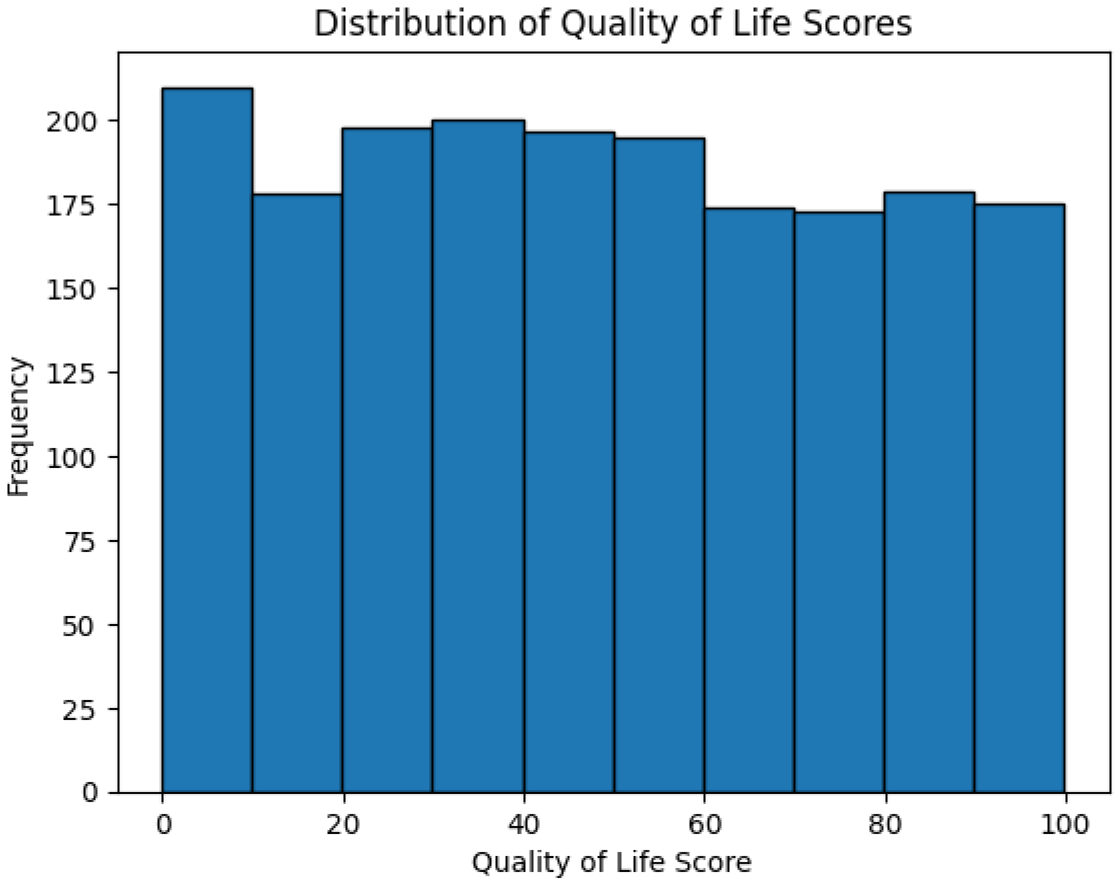


In [ ]:

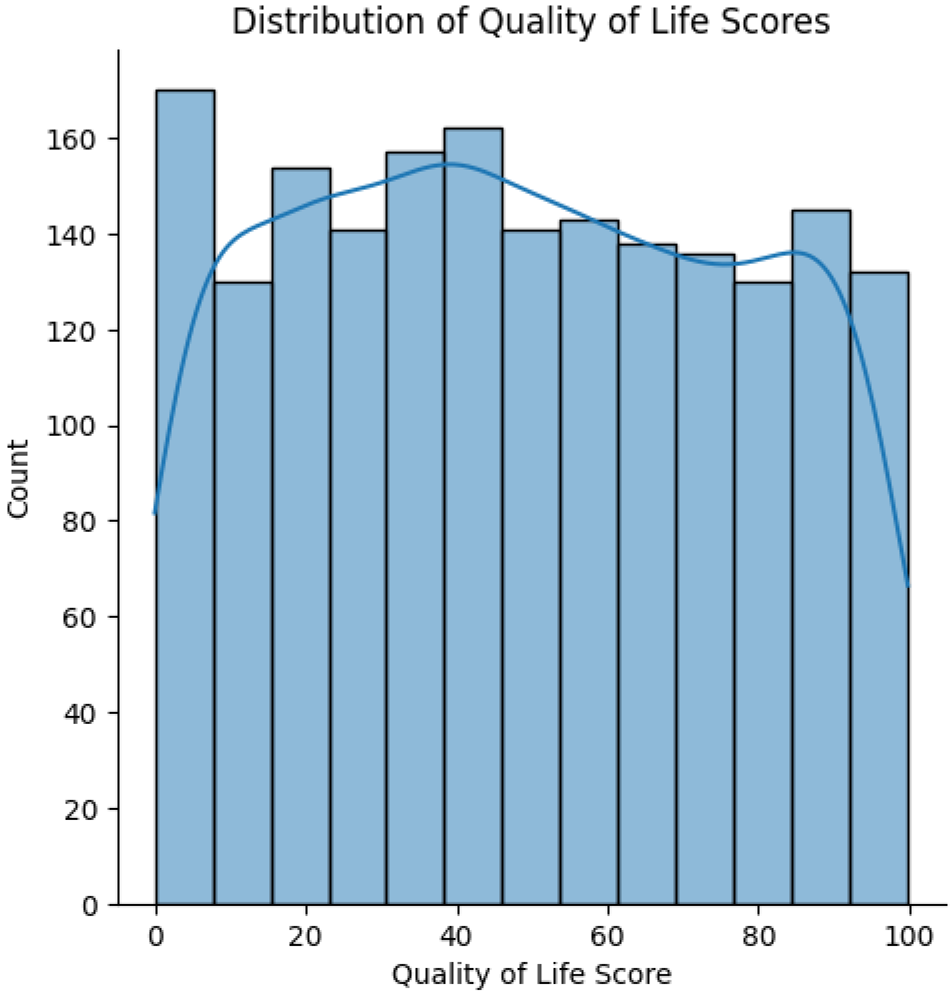
```
#next question 5
```

```
#What is the distribution of Quality of Life Scores among the patients?
```

In [45]:
```
plt.hist(df['QualityOfLifeScore'], bins=10, edgecolor='black')
plt.title('Distribution of Quality of Life Scores')
plt.xlabel('Quality of Life Score')
plt.ylabel('Frequency')
plt.show()
```



In [43]:
```
sns.displot(df['QualityOfLifeScore'], kde=True)
plt.title('Distribution of Quality of Life Scores')
plt.xlabel('Quality of Life Score')
plt.show()
```

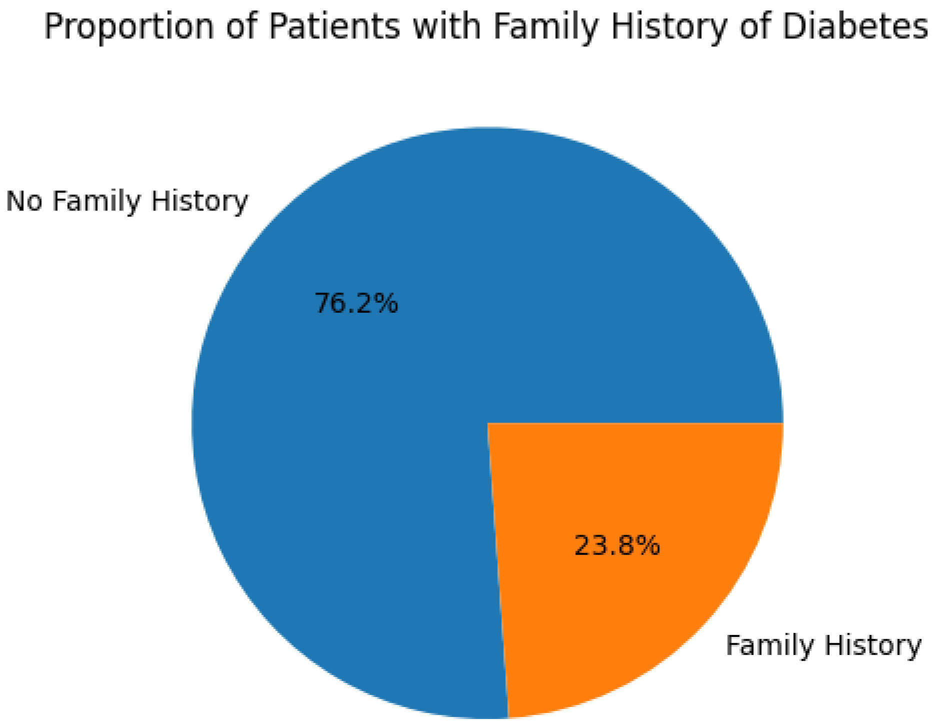


In [ ]:

```
# now going to the next question no 6

# 6.How does the distribution of Diabetes  vary for individuals with and without a famil
```

In [28]:

```
family_history_counts = df['FamilyHistoryDiabetes'].value_counts()
plt.pie(family_history_counts, labels=['No Family History', 'Family History'], autopct='
plt.title('Proportion of Patients with Family History of Diabetes')
plt.show()
```

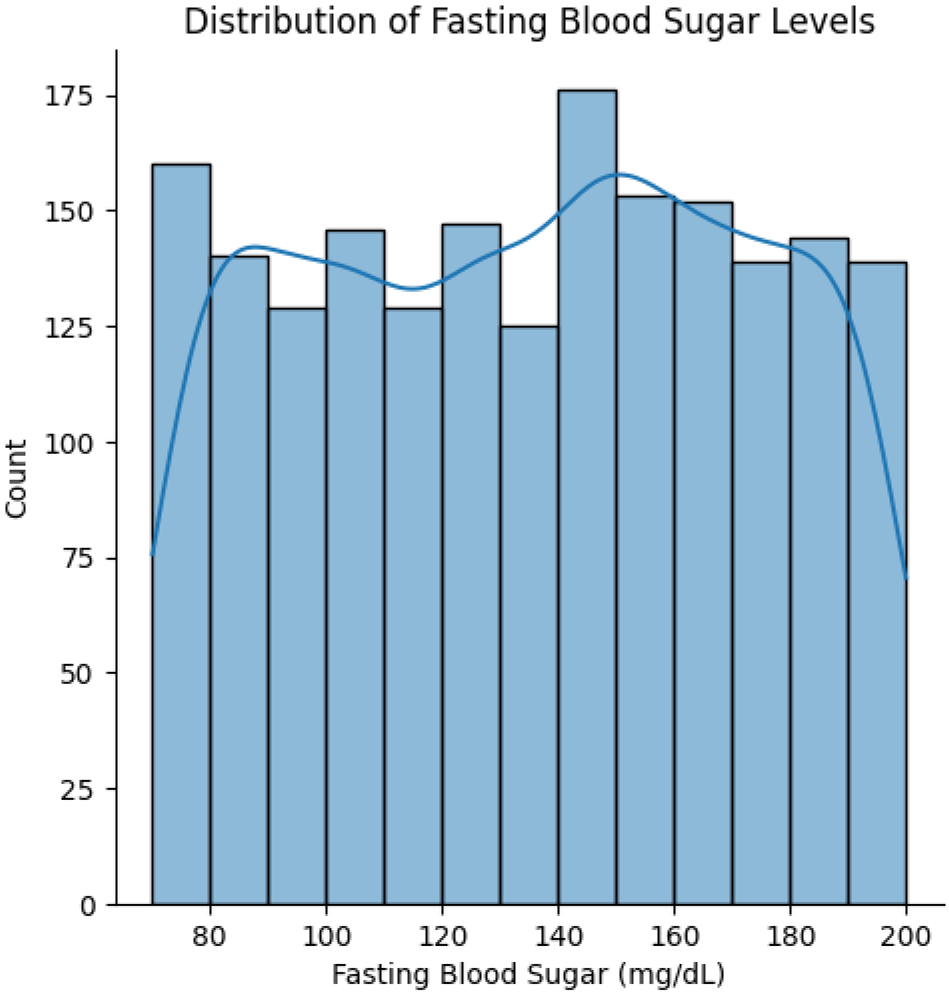


In [ ]:

```
#question 7

#How are fasting blood sugar levels distributed in the overall population?
```

In [53]:

```
sns.displot(df['FastingBloodSugar'], kde=True)
plt.title('Distribution of Fasting Blood Sugar Levels')
plt.xlabel('Fasting Blood Sugar (mg/dL)')
plt.show()
```

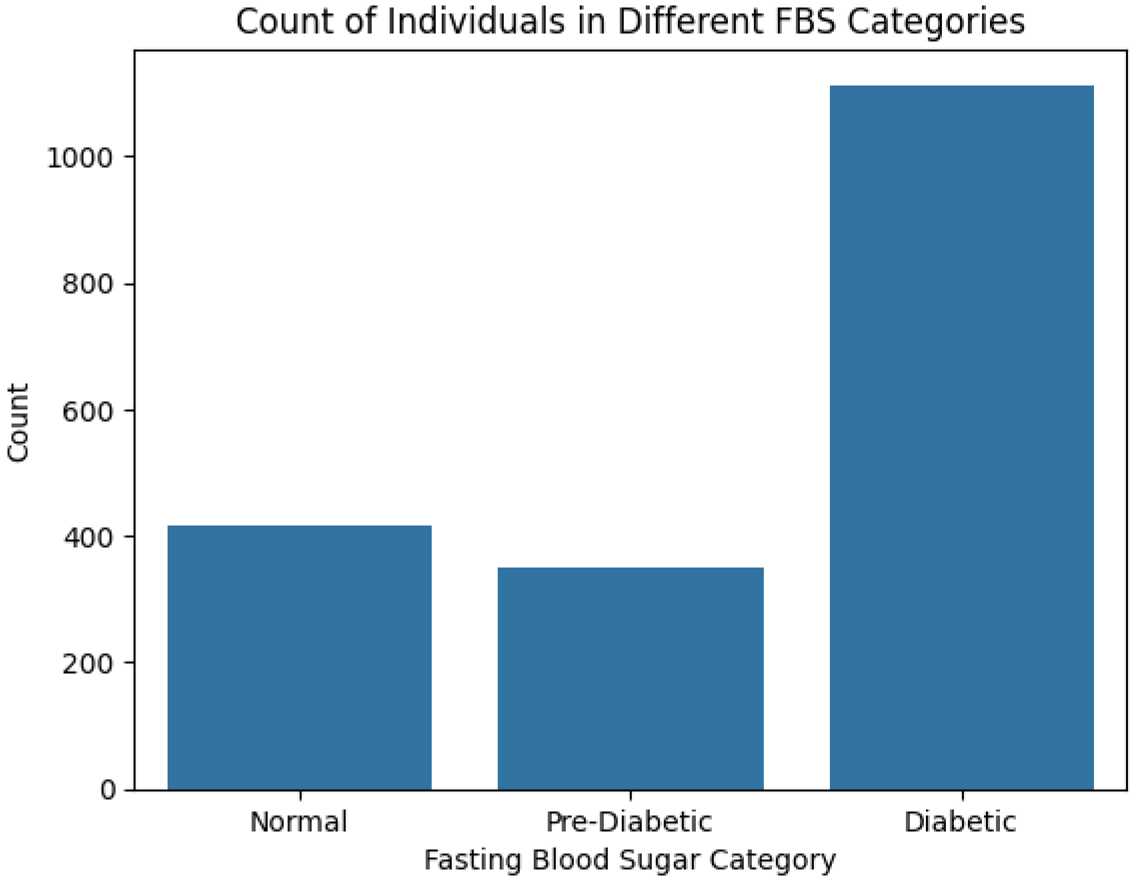


In [ ]:

```
# How many individuals fall into different categories of fasting blood sugar level?
```

In [56]:

```
# Example categories (adjust based on medical guidelines)
df['FBS_Category'] = pd.cut(df['FastingBloodSugar'],
                             bins=[0, 99, 125, float('inf')],
                             labels=['Normal', 'Pre-Diabetic', 'Diabetic'])

sns.countplot(x='FBS_Category', data=df)
plt.title('Count of Individuals in Different FBS Categories')
plt.xlabel('Fasting Blood Sugar Category')
plt.ylabel('Count')
plt.show()
```


Count of Individuals in Different FBS Categories
Count
0
200
400
600
800
1000
Normal
Pre-Diabetic
Diabetic
Fasting Blood Sugar Category