


```
[3]: df.head()
```

5 rows x 46 columns

5 rows x 46 columns

```
[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',
```

```
[6]: df.isna().sum()
```

```
[6]: 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
```

```
[7]: df.info()
```

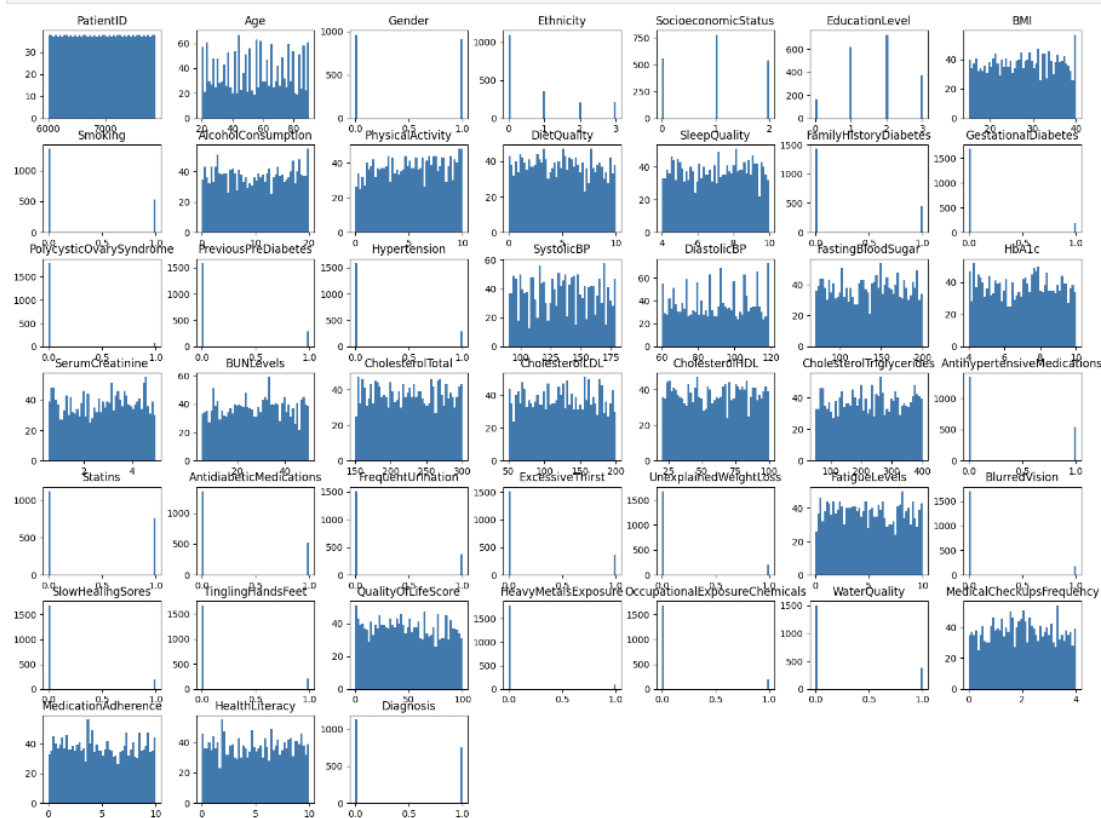
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1879 entries, 0 to 1878
Data columns (total 46 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   PatientID                                1879 non-null   int64
```



Search



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



```
[9]: df.describe()
```

```
[9]: PatientID  Age  Gender  Ethnicity  SocioeconomicStatus  EducationLevel  BMI  Smoking  AlcoholConsumption  PhysicalActivity  ... :
```

```
[9]: df.describe()
```

	PatientID	Age	Gender	Ethnicity	SocioeconomicStatus	EducationLevel	BMI	Smoking	AlcoholConsumption	PhysicalActivity	
count	1879.000000	1879.000000	1879.000000	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	0.281533	10.096587	5.200790	...
std	542.564896	20.515839	0.499977	1.047558	0.764940	0.885665	7.190975	0.449866	5.914216	2.857012	...
min	6000.000000	20.000000	0.000000	0.000000	0.000000	0.000000	15.025898	0.000000	0.000928	0.004089	...
25%	6469.500000	38.000000	0.000000	0.000000	0.000000	1.000000	21.469981	0.000000	4.789725	2.751022	...
50%	6939.000000	55.000000	0.000000	0.000000	1.000000	2.000000	27.722988	0.000000	10.173865	5.249002	...
75%	7408.500000	73.000000	1.000000	1.000000	2.000000	2.000000	33.856460	1.000000	15.285359	7.671402	...
max	7878.000000	90.000000	1.000000	3.000000	2.000000	3.000000	39.998811	1.000000	19.996231	9.993893	...

8 rows x 45 columns

```
[10]: ques=["1. How many people have sugar disease and how many people doesn't have sugar disease?",
          "2. People of which sex has more heart disease?",
          "3. People of which sex have MedicalCheckupsFrequency?",
          "4. People of which chest pain are mmost pron to have heart disease?"]
      ques
```

```
[10]: ["1. How many people have sugar disease and how many people doesn't have sugar disease?",
        '2. People of which sex has more heart disease?',
        '3. People of which sex have which type of chest pain?',
        '4. People of which chest pain are mmost pron to have heart disease?']
```

```
[12]: #to find the ans of first ques
      #getting the values
```

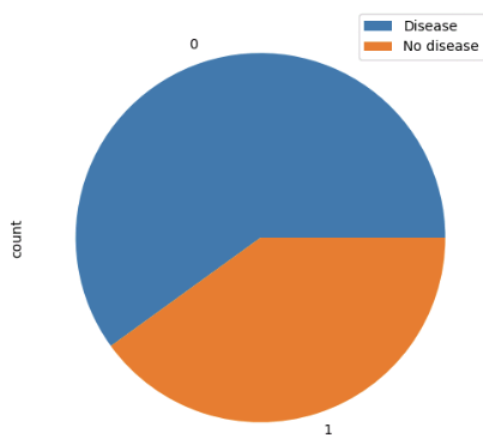
```
df.Diagnosis.value_counts()
```

```
[12]: Diagnosis
0    1127
1     752
      Name: count, dtype: int64
```

```
[14]: #plotting bar chart
      df.Diagnosis.value_counts().plot(kind='bar',color=["orchid","salmon"])
      plt.title("Sugar Disease Values")
      plt.xlabel("1= sugar Disease,0= No sugar Disease")
      plt.ylabel("num");
```



```
[15]: #plotting pie chart
df.Diagnosis.value_counts().plot(kind='pie',figsize=(8,6))
plt.legend(["Disease","No disease"]);
```



```
[16]: # 0 for Female
# 1 for male
# sex column part
#0 represent no disease
# 1 for disesase
#target column part

df.Gender.value_counts()
```

```
[16]: Gender
0    963
1    916
Name: count, dtype: int64
```

```
[17]: #plotting pie chart

df.Gender.value_counts().plot(kind='pie',figsize=(8,6))
plt.title('Male Female ratio')
plt.legend(['Male','Female']);
```

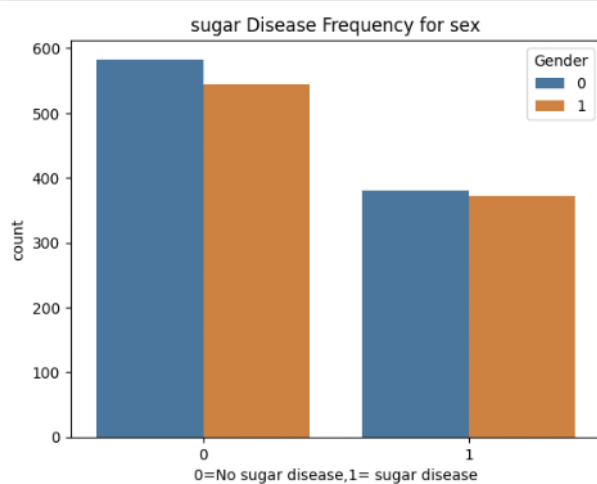
Male Female ratio



```
[19]: #for 2 ques
pd.crosstab(df.Diagnosis,df.Gender)
```

```
[19]: Gender    0    1
      Diagnosis
      0  583  544
      1  380  372
```

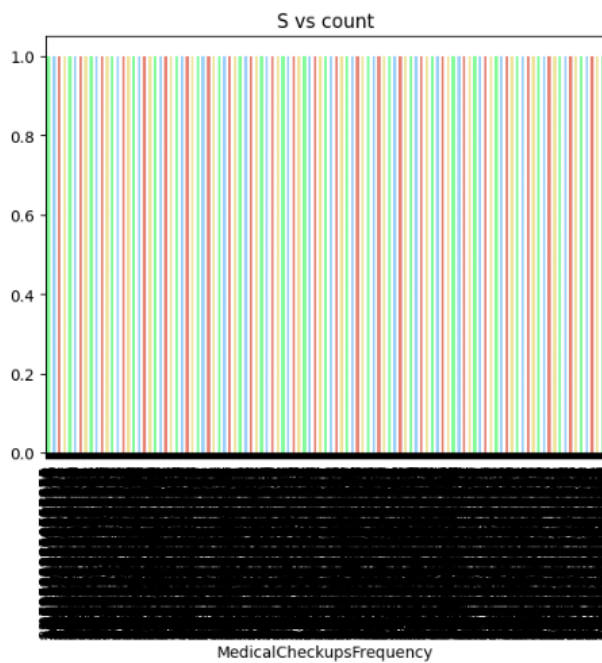
```
[21]: sns.countplot(x='Diagnosis',data=df,hue='Gender')
plt.title("sugar Disease Frequency for sex")
plt.xlabel("0=No sugar disease,1= sugar disease");
```



```
[22]: #3
df.MedicalCheckupsFrequency.value_counts()
```

```
[22]: MedicalCheckupsFrequency
3.070583    1
1.782724    1
3.381070    1
2.701019    1
1.409056    1
..
2.048066    1
2.410958    1
```

```
[23]: df.MedicalCheckupsFrequency.value_counts().plot(kind='bar',color=['salmon','lightskyblue','springgreen','khaki'])
plt.title('S vs count');
```



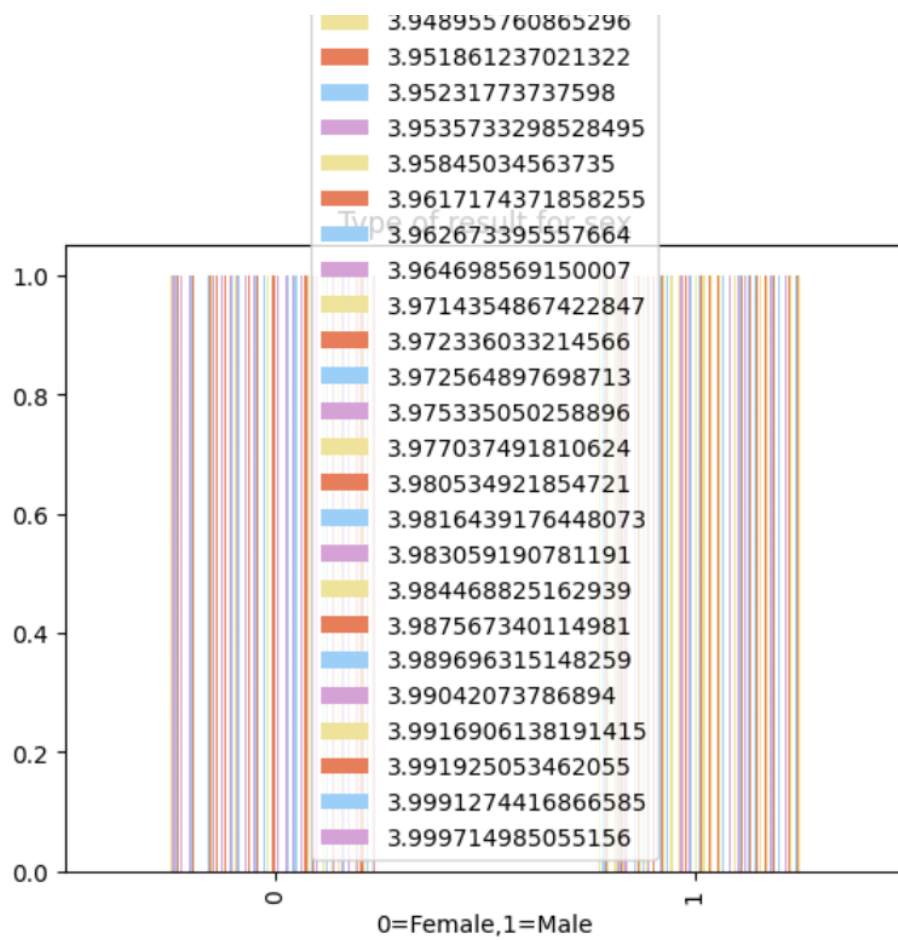
```
[24]: pd.crosstab(df.Gender,df.MedicalCheckupsFrequency)
```

```
[24]: MedicalCheckupsFrequency  0.004013  0.008091  0.009260  0.009428  0.014202  0.016009  0.016215  0.019152  0.027725  0.028107  ...  3.981644  3.983059  3.984469
Gender
0      0      1      0      0      1      1      1      0      1      1      0 ...      0      1      1
1      1      0      1      1      0      0      0      1      0      0      1 ...      1      0      0
```

2 rows × 1879 columns



```
[25]: pd.crosstab(df.Gender,df.MedicalCheckupsFrequency).plot(kind='bar',color=['coral','lightskyblue','plum','khaki'])
plt.title("Type of result for sex")
plt.xlabel('0=Female,1=Male');
```



[]: