In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
import math
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.datasets import make_classification
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
%matplotlib inline
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
from ydata_profiling import ProfileReport
import math
```

In [34]:

```
original_data=pd.read_csv(r"C:\Users\DELL\Downloads\archive\diabetes_binary_
```

In [3]:

```
data=pd.read_csv(r"C:\Users\DELL\Downloads\archive\diabetes_binary_health_ir
data
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDis
0	0.0	0	1.0	1	15.0	1.0	0.0	0.0
1	1.0	1	0.0	1	28.0	0.0	0.0	1.0
2	1.0	1	1.0	1	33.0	0.0	0.0	0.0
3	1.0	0	1.0	1	29.0	0.0	1.0	1.0
4	0.0	0	0.0	1	24.0	1.0	0.0	0.0
...	...	...	...	...	...	...	...	...
236373	1.0	1	1.0	1	21.0	0.0	0.0	0.0
236374	0.0	1	0.0	1	25.0	1.0	0.0	0.0
236375	0.0	0	1.0	1	31.0	0.0	0.0	0.0
236376	0.0	1	0.0	1	24.0	0.0	0.0	0.0
236377	0.0	0	1.0	1	32.0	0.0	0.0	0.0

```
236378 rows × 22 columns
```

In [4]:

```
profile=ProfileReport(data)
profile
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

In [5]:

```
data.head
```

```
<bound method NDFrame.head of         Diabetes_binary  HighBP  HighChol  CholCheck   BMI  Smoker
0                   0.0       0     1.0          1  15.0     1.0     0.0
1                   1.0       1     0.0          1  28.0     0.0     0.0
2                   1.0       1     1.0          1  33.0     0.0     0.0
3                   1.0       0     1.0          1  29.0     0.0     1.0
4                   0.0       0     0.0          1  24.0     1.0     0.0
...                 ...     ...     ...        ...   ...     ...     ...
236373              1.0       1     1.0          1  21.0     0.0     0.0
236374              0.0       1     0.0          1  25.0     1.0     0.0
236375              0.0       0     1.0          1  31.0     0.0     0.0
236376              0.0       1     0.0          1  24.0     0.0     0.0
236377              0.0       0     1.0          1  32.0     0.0     0.0

        HeartDiseaseorAttack  PhysActivity  Fruits  ...  AnyHealthcare  \
0                        0.0             0       1  ...              1
1                        1.0             0       1  ...              1
2                        0.0             1       1  ...              1
3                        1.0             1       1  ...              1
4                        0.0             0       0  ...              1
...                      ...           ...     ...  ...            ...
236373                   0.0             1       1  ...              1
236374                   0.0             1       1  ...              0
236375                   0.0             1       1  ...              1
236376                   0.0             1       1  ...              1
236377                   0.0             1       0  ...              1

        NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex  Age  \
0               0.0      5.0      10.0      20.0       0.0    0   11
1               0.0      2.0       0.0       0.0       0.0    0   11
2               0.0      2.0      10.0       0.0       0.0    0    9
3               0.0      5.0       0.0      30.0       1.0    1   12
4               0.0      3.0       0.0       0.0       1.0    1   13
...             ...      ...       ...       ...       ...  ...  ...
236373          0.0      4.0       0.0       0.0       0.0    1   10
236374          1.0      2.0      20.0       0.0       0.0    0    3
236375          0.0      2.0       0.0       0.0       0.0    1    7
236376          0.0      2.0       0.0       0.0       0.0    1   10
236377          0.0      1.0       2.0       2.0       0.0    0    6

        Education  Income
0             4.0     5.0
1             4.0     3.0
2             4.0     7.0
3             3.0     4.0
4             5.0     6.0
...           ...     ...
236373        2.0     3.0
236374        4.0     5.0
236375        6.0    10.0
236376        4.0     6.0
236377        6.0     6.0

[236378 rows x 22 columns]>
```

In [6]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236378 entries, 0 to 236377
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Diabetes_binary       236378 non-null  float64
 1   HighBP                236378 non-null  int64  
 2   HighChol              236378 non-null  float64
 3   CholCheck             236378 non-null  int64  
 4   BMI                   236378 non-null  float64
 5   Smoker                236378 non-null  float64
 6   Stroke                236378 non-null  float64
 7   HeartDiseaseorAttack  236378 non-null  float64
 8   PhysActivity          236378 non-null  int64  
 9   Fruits                236378 non-null  int64  
 10  Veggies               236378 non-null  int64  
 11  HvyAlcoholConsump     236378 non-null  int64  
 12  AnyHealthcare         236378 non-null  int64  
 13  NoDocbcCost           236378 non-null  float64
 14  GenHlth               236378 non-null  float64
 15  MentHlth              236378 non-null  float64
 16  PhysHlth              236378 non-null  float64
 17  DiffWalk              236378 non-null  float64
 18  Sex                   236378 non-null  int64  
 19  Age                   236378 non-null  int64  
 20  Education             236378 non-null  float64
 21  Income                236378 non-null  float64
dtypes: float64(13), int64(9)
memory usage: 39.7 MB
```

In [7]:

```
data.shape
```

```
(236378, 22)
```

In [8]:

```
data.describe()
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker
count	236378.000000	236378.000000	236378.000000	236378.000000	236378.000000	236378.0
mean	0.142010	0.418558	0.402059	0.963347	28.953579	0.411997
std	0.349061	0.493324	0.490315	0.187909	6.552055	0.492196
min	0.000000	0.000000	0.000000	0.000000	12.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000	24.000000	0.000000
50%	0.000000	0.000000	0.000000	1.000000	28.000000	0.000000
75%	0.000000	1.000000	1.000000	1.000000	32.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	99.000000	1.000000

8 rows × 22 columns

preprocessing

In [9]:

```
data["Diabetes_binary"] = data["Diabetes_binary"].astype(int)
data["HighBP"] = data["HighBP"].astype(int)
data["HighChol"] = data["HighChol"].astype(int)
data["CholCheck"] = data["CholCheck"].astype(int)
data["BMI"] = data["BMI"].astype(int)
data["Smoker"] = data["Smoker"].astype(int)
data["Stroke"] = data["Stroke"].astype(int)
data["HeartDiseaseorAttack"] = data["HeartDiseaseorAttack"].astype(int)
data["PhysActivity"] = data["PhysActivity"].astype(int)
data["Fruits"] = data["Fruits"].astype(int)
data["Veggies"] = data["Veggies"].astype(int)
data["HvyAlcoholConsump"] = data["HvyAlcoholConsump"].astype(int)
data["AnyHealthcare"] = data["AnyHealthcare"].astype(int)
data["NoDocbcCost"] = data["NoDocbcCost"].astype(int)
data["GenHlth"] = data["GenHlth"].astype(int)
data["MentHlth"] = data["MentHlth"].astype(int)
data["PhysHlth"] = data["PhysHlth"].astype(int)
data["DiffWalk"] = data["DiffWalk"].astype(int)
data["Sex"] = data["Sex"].astype(int)
data["Age"] = data["Age"].astype(int)
data["Education"] = data["Education"].astype(int)
data["Income"] =data["Income"].astype(int)
```

```
In [10]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236378 entries, 0 to 236377
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Diabetes_binary       236378 non-null  int32
 1   HighBP                236378 non-null  int32
 2   HighChol              236378 non-null  int32
 3   CholCheck             236378 non-null  int32
 4   BMI                   236378 non-null  int32
 5   Smoker                236378 non-null  int32
 6   Stroke                236378 non-null  int32
 7   HeartDiseaseorAttack  236378 non-null  int32
 8   PhysActivity          236378 non-null  int32
 9   Fruits                236378 non-null  int32
 10  Veggies               236378 non-null  int32
 11  HvyAlcoholConsump     236378 non-null  int32
 12  AnyHealthcare         236378 non-null  int32
 13  NoDocbcCost           236378 non-null  int32
 14  GenHlth               236378 non-null  int32
 15  MentHlth              236378 non-null  int32
 16  PhysHlth              236378 non-null  int32
 17  DiffWalk              236378 non-null  int32
 18  Sex                   236378 non-null  int32
 19  Age                   236378 non-null  int32
 20  Education             236378 non-null  int32
 21  Income                236378 non-null  int32
dtypes: int32(22)
memory usage: 19.8 MB
```

```
In [11]: data.head()
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseo
0	0	0	1	1	15	1	0	0
1	1	1	0	1	28	0	0	1
2	1	1	1	1	33	0	0	0
3	1	0	1	1	29	0	1	1
4	0	0	0	1	24	1	0	0

5 rows × 22 columns

In [12]:

```
data.describe()
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker
count	236378.000000	236378.000000	236378.000000	236378.000000	236378.000000	236378.0
mean	0.142010	0.418558	0.402059	0.963347	28.953579	0.411997
std	0.349061	0.493324	0.490315	0.187909	6.552055	0.492196
min	0.000000	0.000000	0.000000	0.000000	12.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000	24.000000	0.000000
50%	0.000000	0.000000	0.000000	1.000000	28.000000	0.000000
75%	0.000000	1.000000	1.000000	1.000000	32.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	99.000000	1.000000

```
8 rows × 22 columns
```

In [13]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236378 entries, 0 to 236377
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Diabetes_binary       236378 non-null  int32
 1   HighBP                236378 non-null  int32
 2   HighChol              236378 non-null  int32
 3   CholCheck             236378 non-null  int32
 4   BMI                   236378 non-null  int32
 5   Smoker                236378 non-null  int32
 6   Stroke                236378 non-null  int32
 7   HeartDiseaseorAttack  236378 non-null  int32
 8   PhysActivity          236378 non-null  int32
 9   Fruits                236378 non-null  int32
 10  Veggies               236378 non-null  int32
 11  HvyAlcoholConsump     236378 non-null  int32
 12  AnyHealthcare         236378 non-null  int32
 13  NoDocbcCost           236378 non-null  int32
 14  GenHlth               236378 non-null  int32
 15  MentHlth              236378 non-null  int32
 16  PhysHlth              236378 non-null  int32
 17  DiffWalk              236378 non-null  int32
 18  Sex                   236378 non-null  int32
 19  Age                   236378 non-null  int32
 20  Education             236378 non-null  int32
 21  Income                236378 non-null  int32
dtypes: int32(22)
memory usage: 19.8 MB
```

In [14]:

```
data.isnull().sum ()
```

```
Diabetes_binary         0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
HeartDiseaseorAttack    0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

In [15]:

```
unique_values = {}
for col in data.columns:
    unique_values[col]=data[col].value_counts().shape[0]
    pd.DataFrame(unique_values,index=['unique value count']).transpose()
```

In [16]:

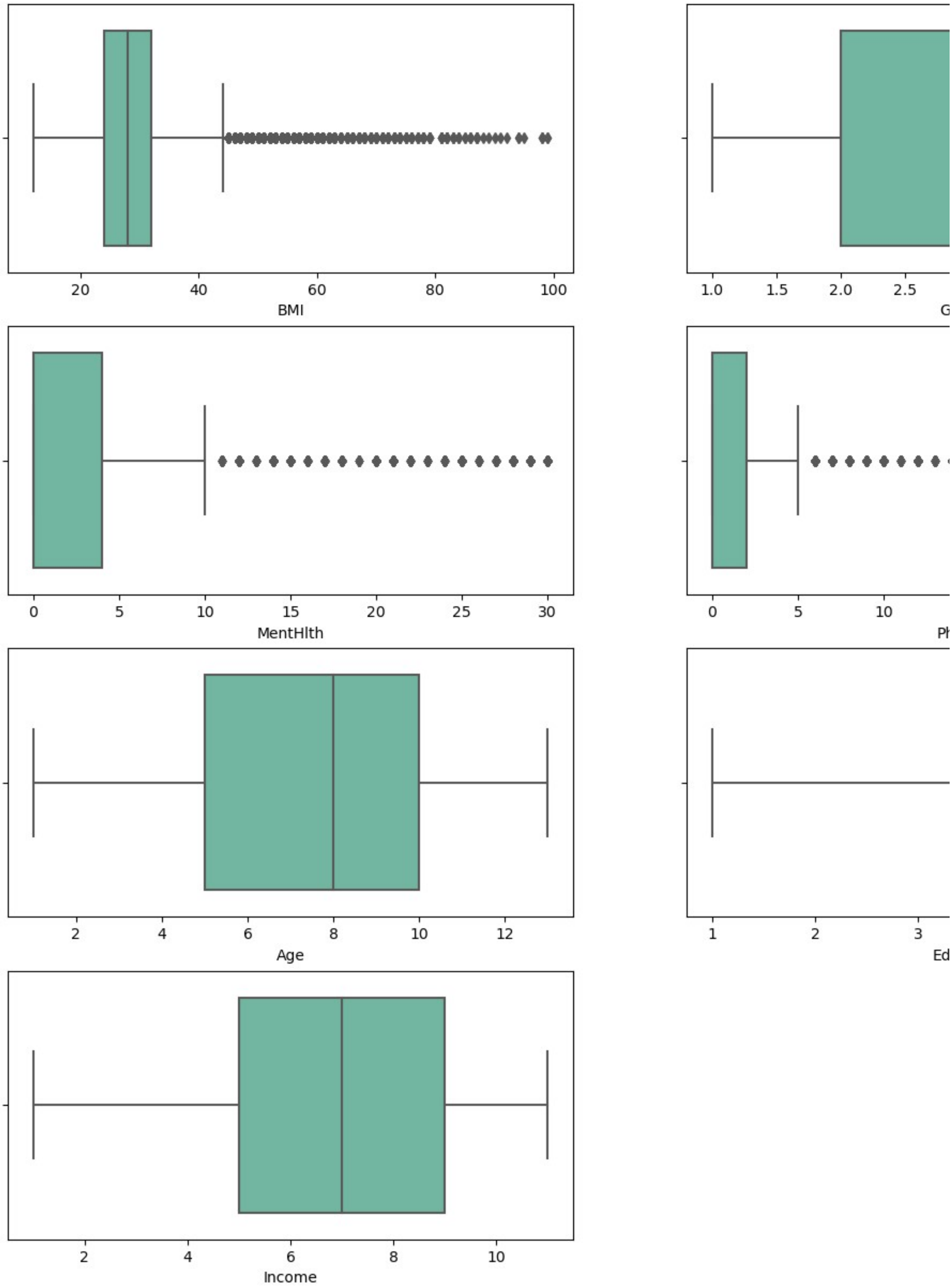
```
def v_counts(dataframe):
    for i in dataframe:
        print(dataframe[i].value_counts())
        print("-------------------------")
```

```
In [17]: v_counts(data)
```

```
0    202810
1     33568
Name: Diabetes_binary, dtype: int64
--------------------------
0    137440
1     98938
Name: HighBP, dtype: int64
--------------------------
0    141340
1     95038
Name: HighChol, dtype: int64
--------------------------
1    227714
0      8664
Name: CholCheck, dtype: int64
--------------------------
27    21733
26    17930
24    17175
28    15225
25    14722
      ...
89        1
90        1
```

In [18]:

```
plt.figure(figsize = (15,15))
for i,col in enumerate(['BMI', 'GenHlth', 'MentHlth', 'PhysHlth', 'Age','Edu
    plt.subplot(4,2,i+1)
    sns.boxplot(x = col, data = data ,palette='Set2')
plt.show()
```



check and drop duplicated data

In [19]:
```
data.duplicated().sum()
```

```
13135
```

In [20]:
```
data.drop_duplicates(inplace = True)
```

In [21]:
```
data.duplicated().sum()
```

```
0
```

In [22]:
```
data.shape
```

```
(223243, 22)
```

some codes are helpful us in EDA PART

In [23]:
```
data2=data.copy()
```

data2.Age[data2['Age'] == 1] = '18 to 24' data2.Age[data2['Age'] == 2] = '25 to '30 to 34' data2.Age[data2['Age'] == 4] = '35 to 39' data2.Age[data2['Age'] == 5 data2.Age[data2['Age'] == 6] = '45 to 49' data2.Age[data2['Age'] == 7] = '50 to '55 to 59' data2.Age[data2['Age'] == 9] = '60 to 64' data2.Age[data2['Age'] == 1 data2.Age[data2['Age'] == 11] = '70 to 74' data2.Age[data2['Age'] == 12] = '75 13] = '80 or older'

data2.Diabetes_012[data2['Diabetes_012'] == 0] = 'No Diabetes' data2.Diabet 1] = 'Diabetes'

data2.HighBP[data2['HighBP'] == 0] = 'No High' data2.HighBP[data2['HighBP']

data2.HighChol[data2['HighChol'] == 0] = 'No High Cholesterol' data2.HighCho Cholesterol'

data2.CholCheck[data2['CholCheck'] == 0] = 'No Cholesterol Check in 5 Years data2.CholCheck[data2['CholCheck'] == 1] = 'Cholesterol Check in 5 Years'

data2.Smoker[data2['Smoker'] == 0] = 'No' data2.Smoker[data2['Smoker'] == 1

data2.Stroke[data2['Stroke'] == 0] = 'No' data2.Stroke[data2['Stroke'] == 1] = 'Y data2.HeartDiseaseorAttack[data2['HeartDiseaseorAttack'] == 0] = 'No' data2.HeartDiseaseorAttack[data2['HeartDiseaseorAttack'] == 1] = 'Yes'

data2.PhysActivity[data2['PhysActivity'] == 0] = 'No' data2.PhysActivity[data2['

data2.Fruits[data2['Fruits'] == 0] = 'No' data2.Fruits[data2['Fruits'] == 1] = 'Yes'

data2.Veggies[data2['Veggies'] == 0] = 'No' data2.Veggies[data2['Veggies'] ==

data2.HvyAlcoholConsump[data2['HvyAlcoholConsump'] == 0] = 'No' data2.HvyAlcoholConsump[data2['HvyAlcoholConsump'] == 1] = 'Yes'

data2.AnyHealthcare[data2['AnyHealthcare'] == 0] = 'No' data2.AnyHealthcare
'Yes'

data2.NoDocbcCost[data2['NoDocbcCost'] == 0] = 'No' data2.NoDocbcCost[da

data2.GenHlth[data2['GenHlth'] == 1] = 'Excellent' data2.GenHlth[data2['GenH
data2.GenHlth[data2['GenHlth'] == 3] = 'Good' data2.GenHlth[data2['GenHlth']
data2.GenHlth[data2['GenHlth'] == 5] = 'Poor'

data2.DiffWalk[data2['DiffWalk'] == 0] = 'No' data2.DiffWalk[data2['DiffWalk'] ==

data2.Sex[data2['Sex'] == 0] = 'Female' data2.Sex[data2['Sex'] == 1] = 'Male'

data2.Education[data2['Education'] == 1] = 'Never Attended School' data2.Edu
'Elementary' data2.Education[data2['Education'] == 3] = 'Junior High School' d
== 4] = 'Senior High School' data2.Education[data2['Education'] == 5] = 'Under
data2.Education[data2['Education'] == 6] = 'Magister'

data2.Income[data2['Income'] == 1] = 'Less Than
$10,000' data2.Income[data2['Income'] == 2] =' LessThan$10,000' dat

'Less Than $10,000' data2.Income[data2['Income'] == 4] =' LessThan$

data2.Income[data2['Income'] == 5] = 'Less Than
$35,000' data2.Income[data2['Income'] == 6] =' LessThan$50,000' dat

'Less Than $75,000' data2.Income[data2['Income'] == 8] =' LessThan$

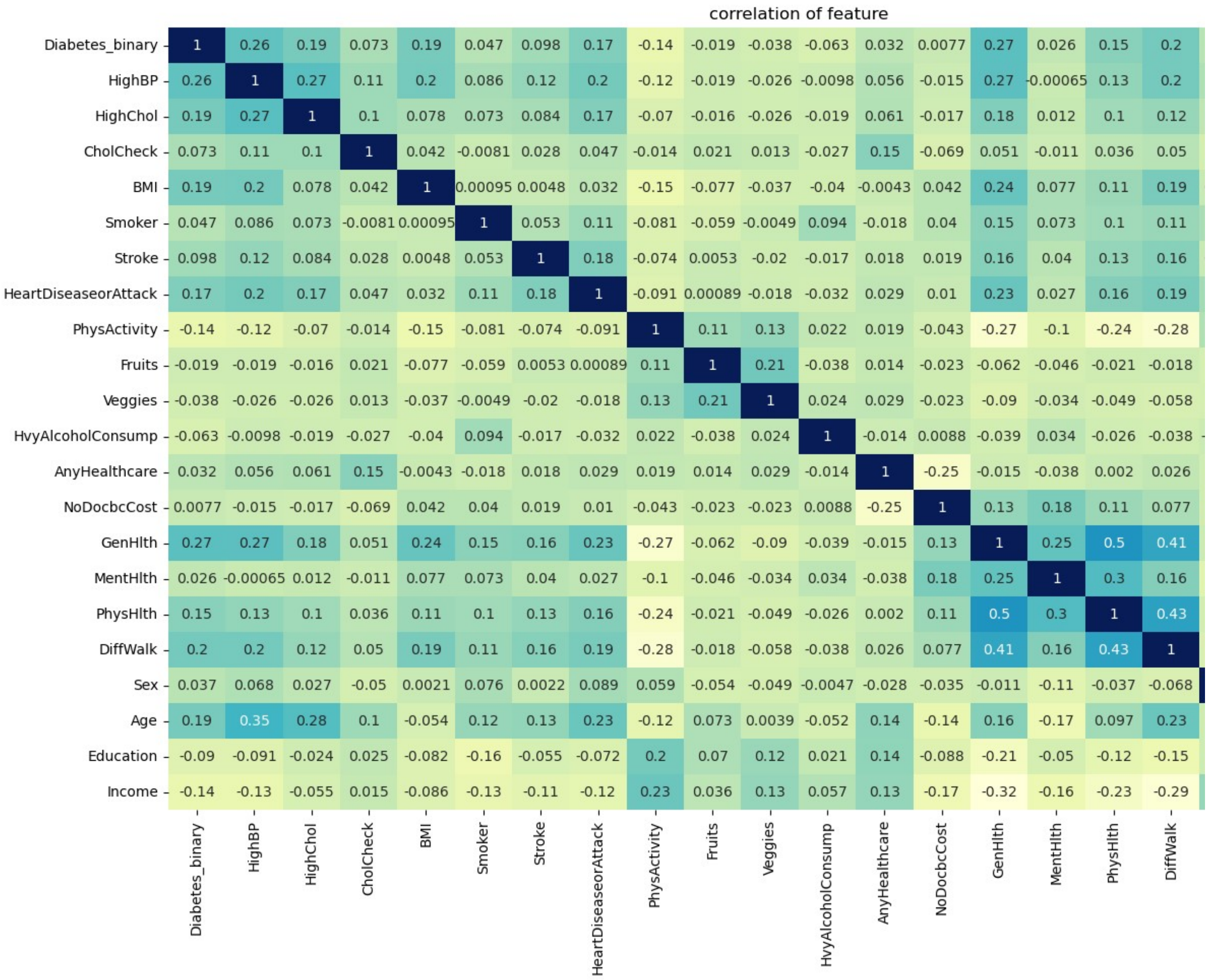
data2.Income[data2['Income'] == 9] = 'Less Than
$150,000' data2.Income[data2['Income'] == 10] =' LessThan$200,000'
11] = '$200,000 or More'

EDA

In [24]:

```
plt.figure(figsize=(20,10))
sns.heatmap(data.corr(),annot=True,cmap='YlGnBu')
plt.title("correlation of feature")
```

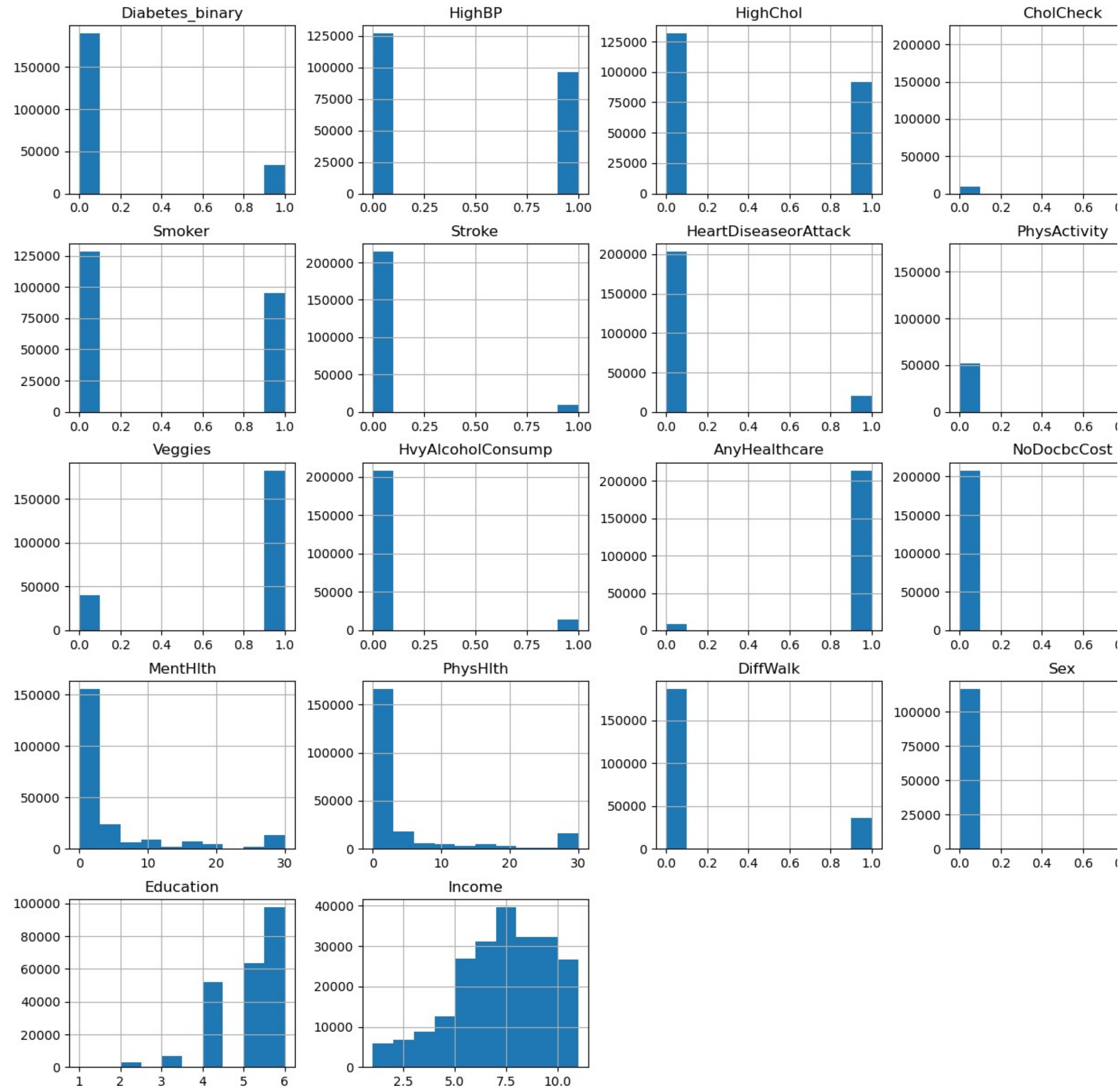
```
Text(0.5, 1.0, 'correlation of feature')
```



	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk
Diabetes_binary	1	0.26	0.19	0.073	0.19	0.047	0.098	0.17	-0.14	-0.019	-0.038	-0.063	0.032	0.0077	0.27	0.026	0.15	0.2
HighBP	0.26	1	0.27	0.11	0.2	0.086	0.12	0.2	-0.12	-0.019	-0.026	-0.0098	0.056	-0.015	0.27	-0.00065	0.13	0.2
HighChol	0.19	0.27	1	0.1	0.078	0.073	0.084	0.17	-0.07	-0.016	-0.026	-0.019	0.061	-0.017	0.18	0.012	0.1	0.12
CholCheck	0.073	0.11	0.1	1	0.042	-0.0081	0.028	0.047	-0.014	0.021	0.013	-0.027	0.15	-0.069	0.051	-0.011	0.036	0.05
BMI	0.19	0.2	0.078	0.042	1	0.00095	0.0048	0.032	-0.15	-0.077	-0.037	-0.04	-0.0043	0.042	0.24	0.077	0.11	0.19
Smoker	0.047	0.086	0.073	-0.0081	0.00095	1	0.053	0.11	-0.081	-0.059	-0.0049	0.094	-0.018	0.04	0.15	0.073	0.1	0.11
Stroke	0.098	0.12	0.084	0.028	0.0048	0.053	1	0.18	-0.074	0.0053	-0.02	-0.017	0.018	0.019	0.16	0.04	0.13	0.16
HeartDiseaseorAttack	0.17	0.2	0.17	0.047	0.032	0.11	0.18	1	-0.091	0.00089	-0.018	-0.032	0.029	0.01	0.23	0.027	0.16	0.19
PhysActivity	-0.14	-0.12	-0.07	-0.014	-0.15	-0.081	-0.074	-0.091	1	0.11	0.13	0.022	0.019	-0.043	-0.27	-0.1	-0.24	-0.28
Fruits	-0.019	-0.019	-0.016	0.021	-0.077	-0.059	0.0053	0.00089	0.11	1	0.21	-0.038	0.014	-0.023	-0.062	-0.046	-0.021	-0.018
Veggies	-0.038	-0.026	-0.026	0.013	-0.037	-0.0049	-0.02	-0.018	0.13	0.21	1	0.024	0.029	-0.023	-0.09	-0.034	-0.049	-0.058
HvyAlcoholConsump	-0.063	-0.0098	-0.019	-0.027	-0.04	0.094	-0.017	-0.032	0.022	-0.038	0.024	1	-0.014	0.0088	-0.039	0.034	-0.026	-0.038
AnyHealthcare	0.032	0.056	0.061	0.15	-0.0043	-0.018	0.018	0.029	0.019	0.014	0.029	-0.014	1	-0.25	-0.015	-0.038	0.002	0.026
NoDocbcCost	0.0077	-0.015	-0.017	-0.069	0.042	0.04	0.019	0.01	-0.043	-0.023	-0.023	0.0088	-0.25	1	0.13	0.18	0.11	0.077
GenHlth	0.27	0.27	0.18	0.051	0.24	0.15	0.16	0.23	-0.27	-0.062	-0.09	-0.039	-0.015	0.13	1	0.25	0.5	0.41
MentHlth	0.026	-0.00065	0.012	-0.011	0.077	0.073	0.04	0.027	-0.1	-0.046	-0.034	0.034	-0.038	0.18	0.25	1	0.3	0.16
PhysHlth	0.15	0.13	0.1	0.036	0.11	0.1	0.13	0.16	-0.24	-0.021	-0.049	-0.026	0.002	0.11	0.5	0.3	1	0.43
DiffWalk	0.2	0.2	0.12	0.05	0.19	0.11	0.16	0.19	-0.28	-0.018	-0.058	-0.038	0.026	0.077	0.41	0.16	0.43	1
Sex	0.037	0.068	0.027	-0.05	0.0021	0.076	0.0022	0.089	0.059	-0.054	-0.049	-0.0047	-0.028	-0.035	-0.011	-0.11	-0.037	-0.068
Age	0.19	0.35	0.28	0.1	-0.054	0.12	0.13	0.23	-0.12	0.073	0.0039	-0.052	0.14	-0.14	0.16	-0.17	0.097	0.23
Education	-0.09	-0.091	-0.024	0.025	-0.082	-0.16	-0.055	-0.072	0.2	0.07	0.12	0.021	0.14	-0.088	-0.21	-0.05	-0.12	-0.15
Income	-0.14	-0.13	-0.055	0.015	-0.086	-0.13	-0.11	-0.12	0.23	0.036	0.13	0.057	0.13	-0.17	-0.32	-0.16	-0.23	-0.29

In [25]:

```
data.hist(figsize=(20,15));
```



In [26]:

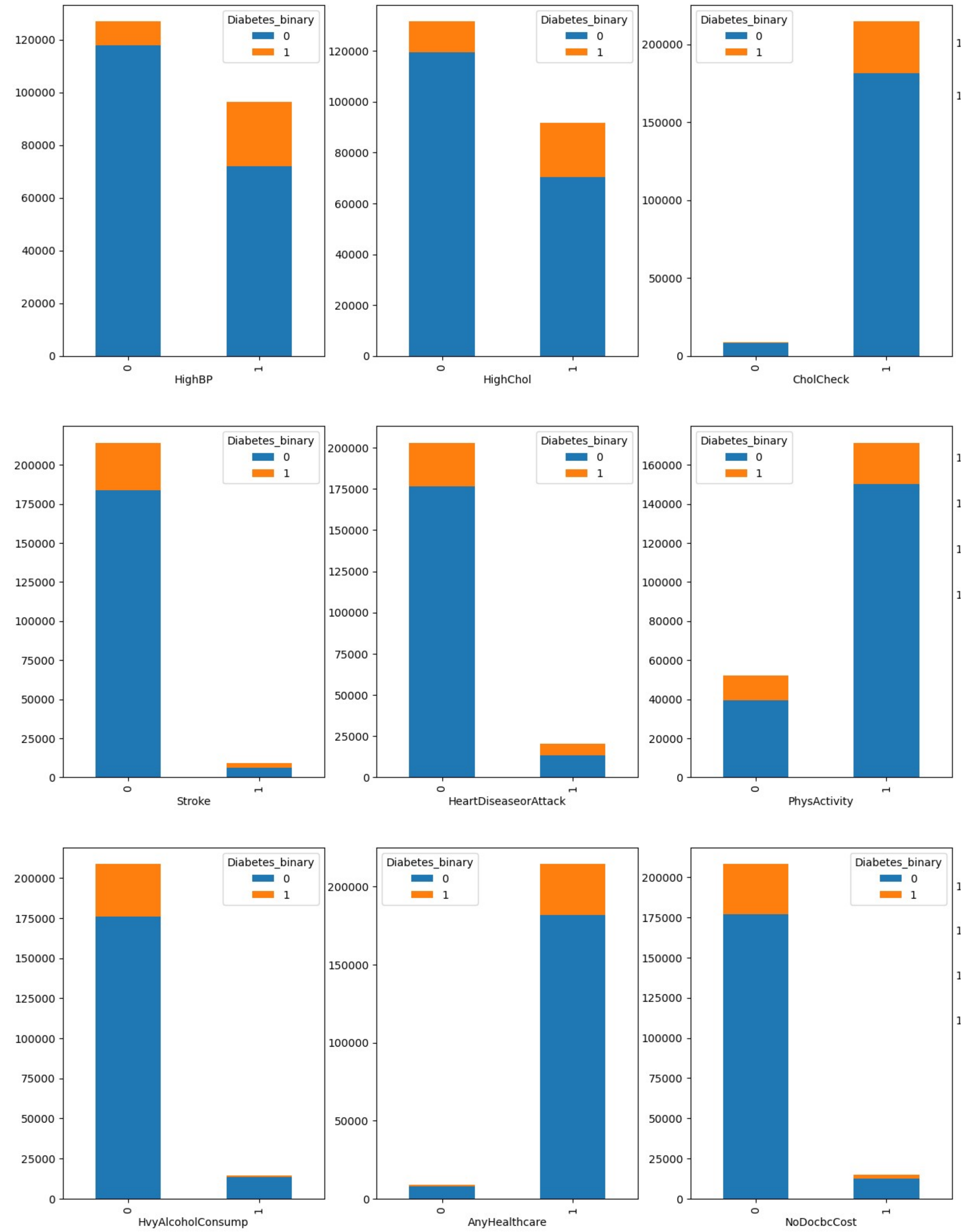
```
cols = ['HighBP','HighChol','CholCheck','Smoker','Stroke','HeartDiseaseorAtt
        'AnyHealthcare','NoDocbcCost','DiffWalk']
```

In [27]:

```
def create_plot_pivot(data2, x_column):
    """ Create a pivot table for satisfaction versus another rating for easy
    _df_plot = data2.groupby([x_column, 'Diabetes_binary']).size() \
    .reset_index().pivot(columns='Diabetes_binary', index=x_column, values=0
    return _df_plot
```

In [28]:

```
fig,ax=plt.subplots(3,4,figsize=(20,20))
axe=ax.ravel()
c=len(cols)
for i in range(c):
    create_plot_pivot(data2,cols[i]).plot(kind='bar',stacked=True,ax=axe[i])
    axe[i].set_xlabel(cols[i])
fig.show()
```



In [29]:

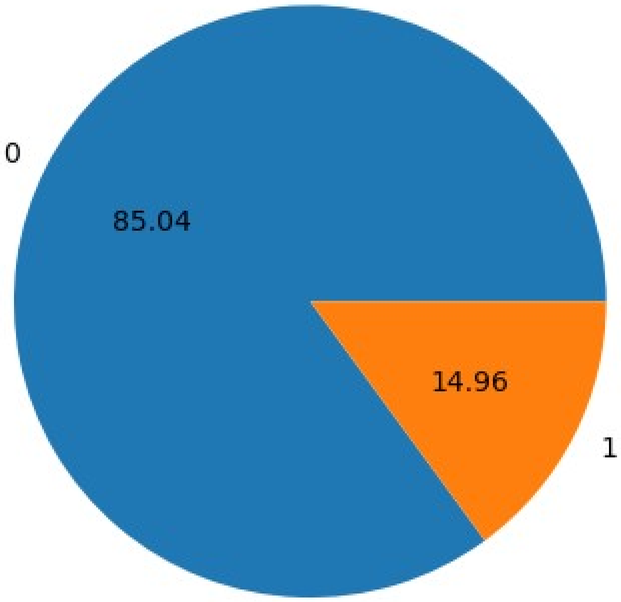
```
data2['Diabetes_binary'].value_counts()
```

```
0    189848
1     33395
Name: Diabetes_binary, dtype: int64
```

In [30]:

```
import matplotlib.pyplot as plt

# Assuming data2["Diabetes_012"] contains categorical values
labels = data2["Diabetes_binary"].value_counts().index
plt.pie(data2["Diabetes_binary"].value_counts(), labels=labels, autopct='%.0
plt.show()
```



In [31]:

```
data["Diabetes_binary_str"]= data["Diabetes_binary"].replace({0:"Non-Diabeti
```

In [32]:

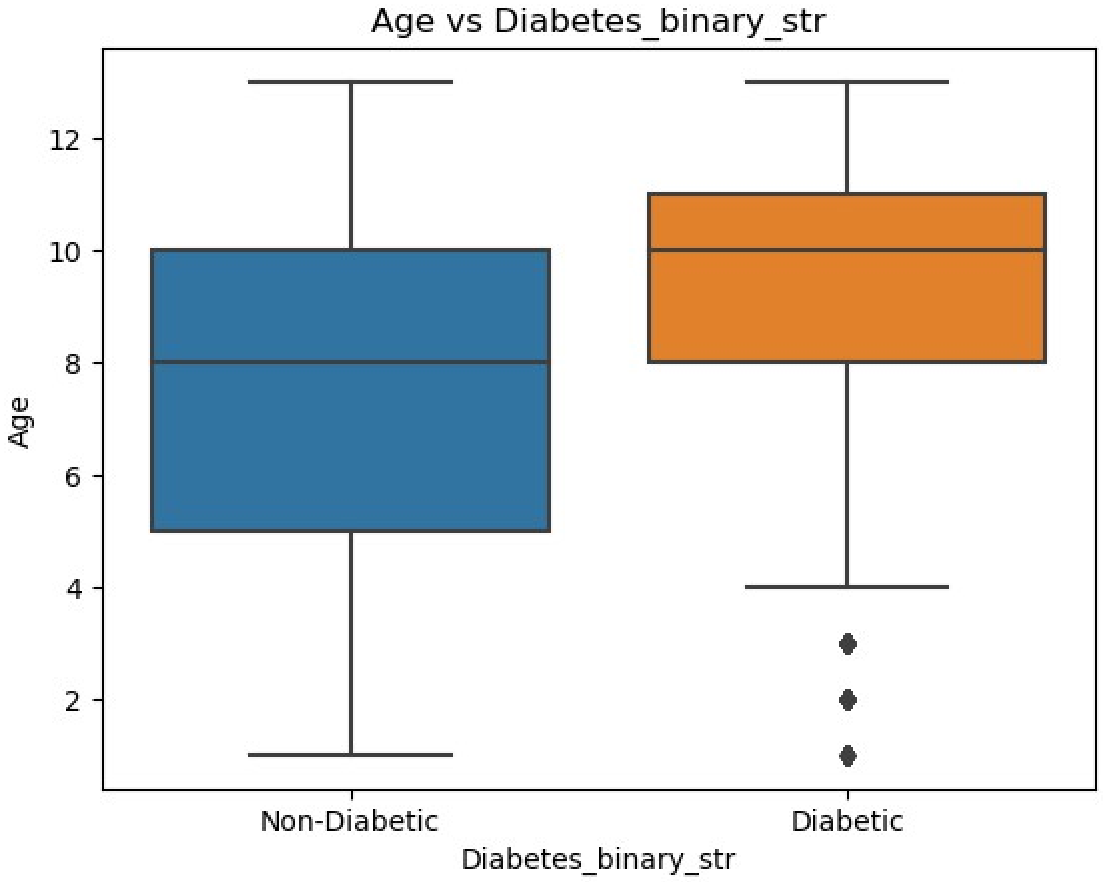
```
data
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDis
0	0	0	1	1	15	1	0	0
1	1	1	0	1	28	0	0	1
2	1	1	1	1	33	0	0	0
3	1	0	1	1	29	0	1	1
4	0	0	0	1	24	1	0	0
...	...	...	...	...	...	...	...	...
236373	1	1	1	1	21	0	0	0
236374	0	1	0	1	25	1	0	0
236375	0	0	1	1	31	0	0	0
236376	0	1	0	1	24	0	0	0
236377	0	0	1	1	32	0	0	0

```
223243 rows × 23 columns
```

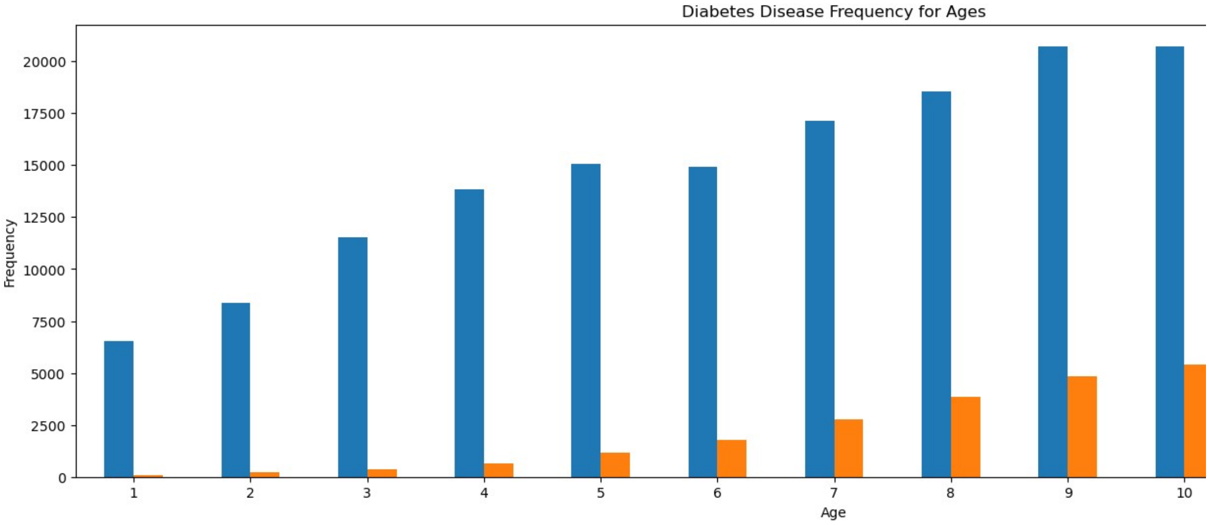
In [72]:

```
sns.boxplot( x="Diabetes_binary_str", y='Age',data=data)
plt.title('Age vs Diabetes_binary_str')
plt.show()
```



In [73]:

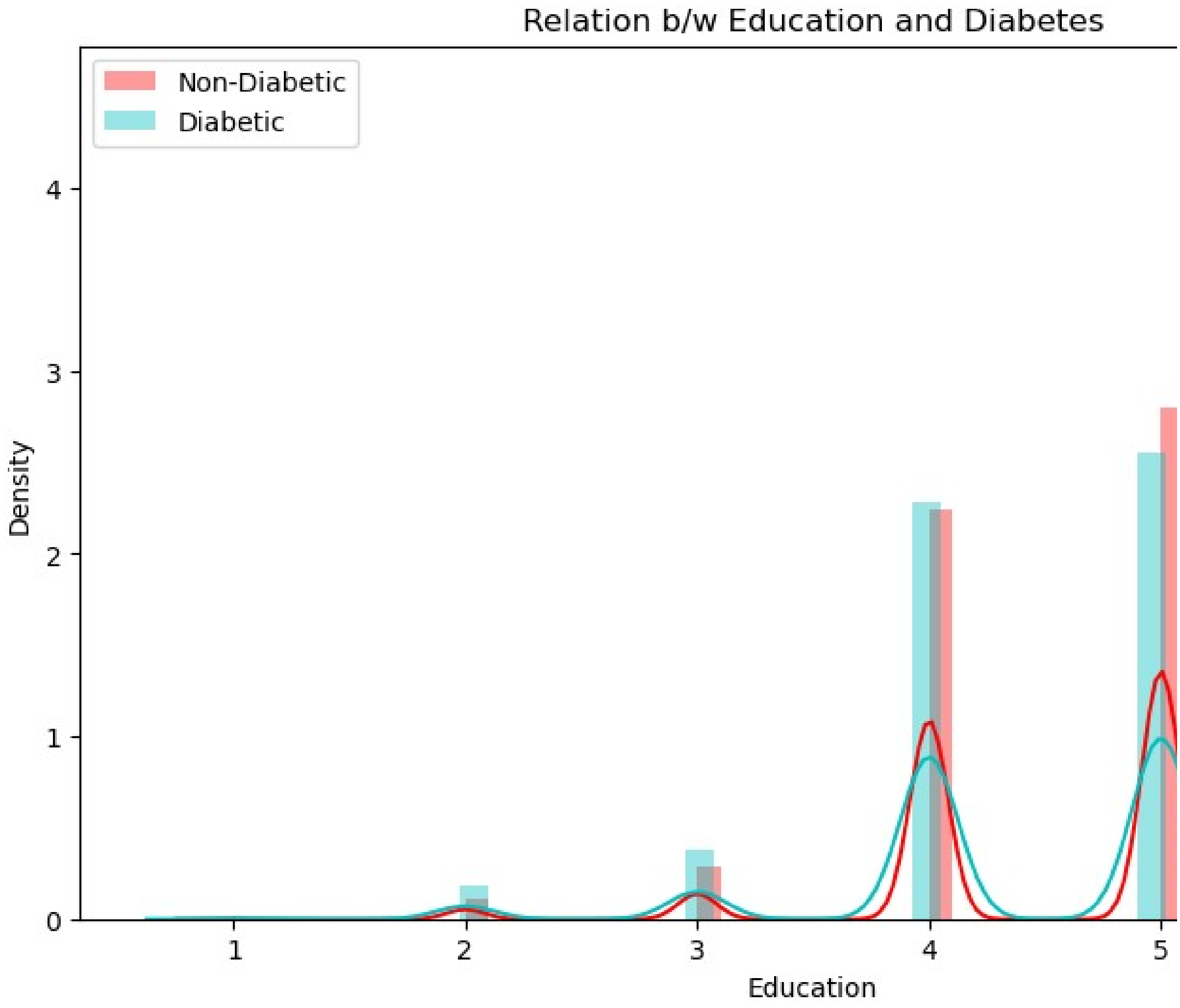
```
pd.crosstab(data2.Age,data2.Diabetes_binary).plot(kind="bar",figsize=(20,6))
plt.title('Diabetes Disease Frequency for Ages')
plt.xlabel('Age')
plt.xticks(rotation=0)
plt.ylabel('Frequency')
plt.show()
```



In [74]:

```
plt.figure(figsize=(10,6))
sns.distplot(data.Education[data.Diabetes_binary==0],color="r",label="Non-Di
sns.distplot(data.Education[data.Diabetes_binary==1],color="c",label="Diabet
plt.title("Relation b/w Education and Diabetes")
plt.legend()
```

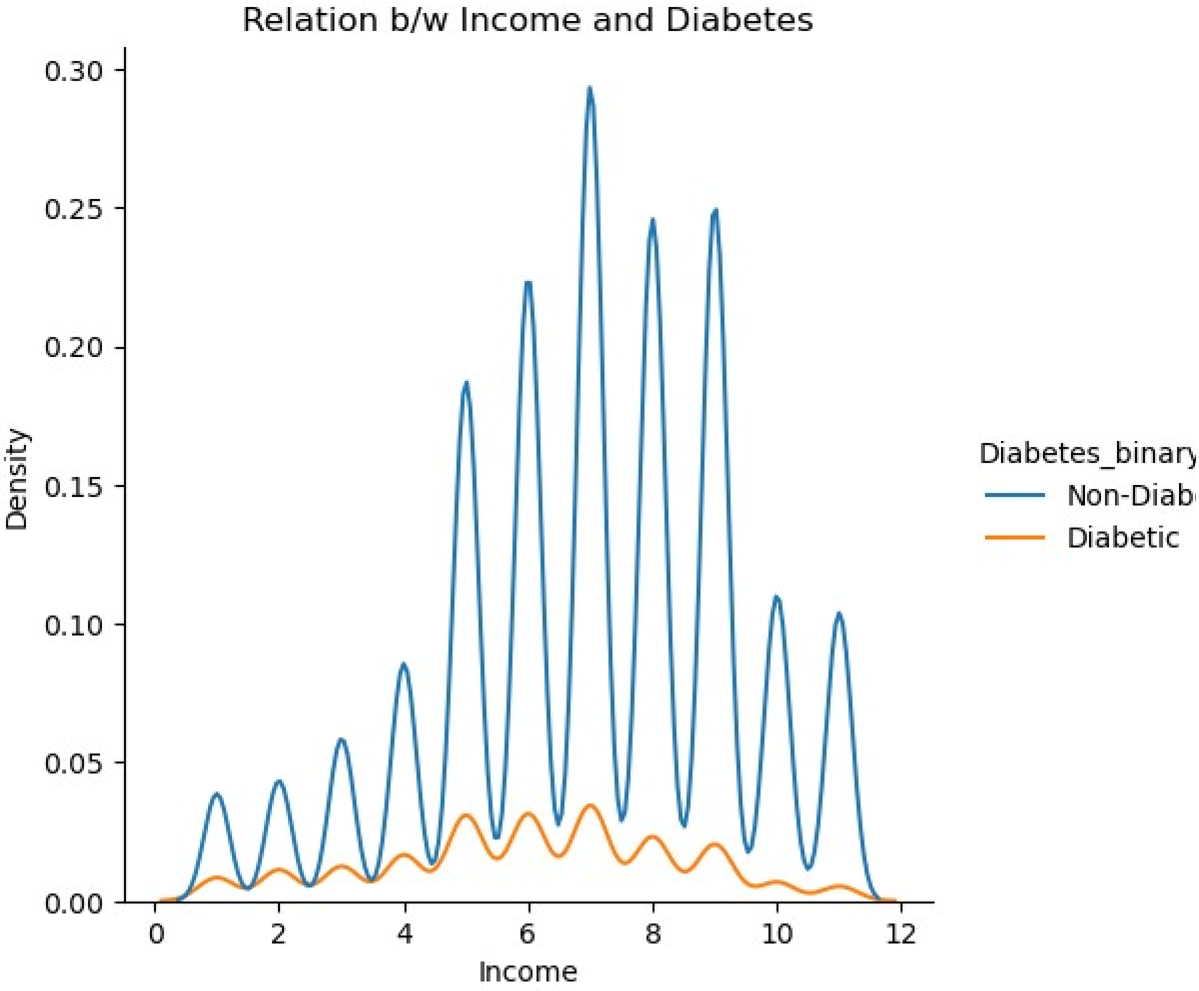
```
<matplotlib.legend.Legend at 0x20000497850>
```



```
In [75]: plt.figure(figsize=(20,10))
         sns.displot(data=data,x="Income",hue="Diabetes_binary_str",kind="kde")
         plt.title("Relation b/w Income and Diabetes")
```

```
Text(0.5, 1.0, 'Relation b/w Income and Diabetes')
```

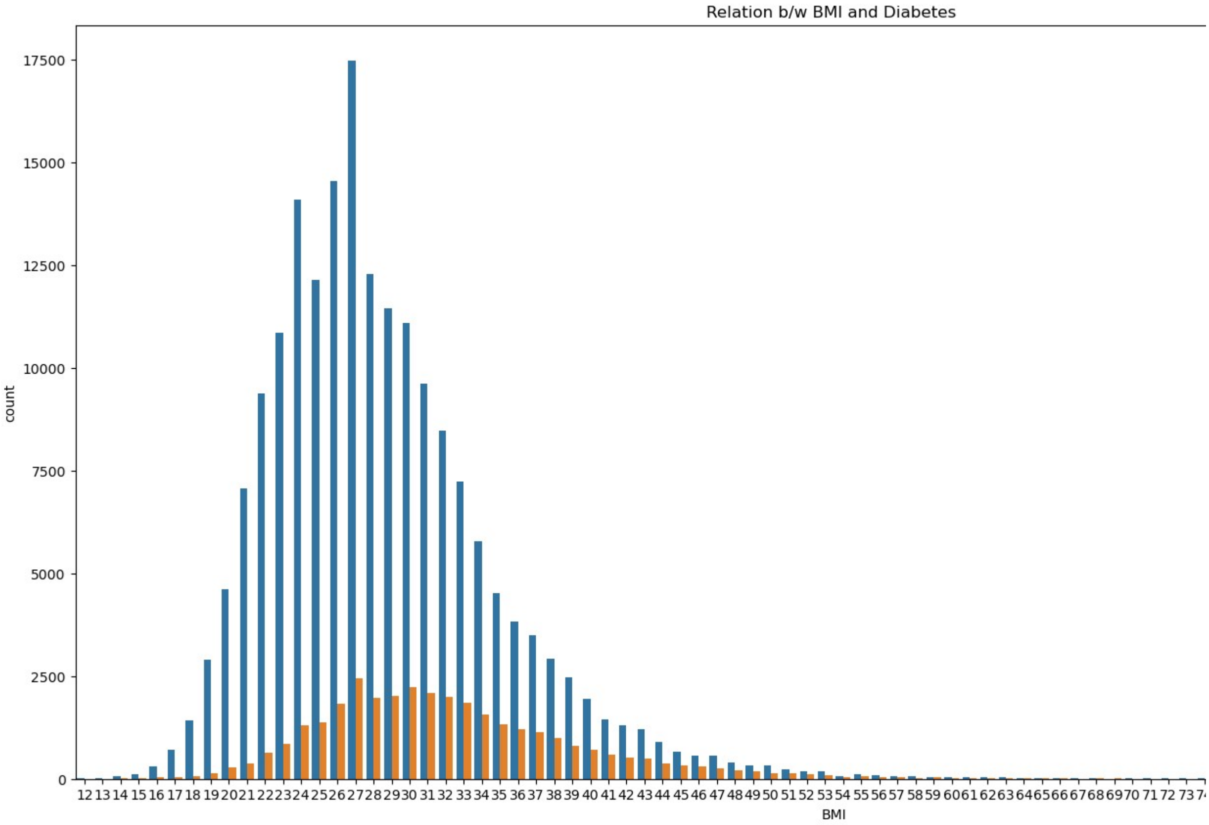
```
<Figure size 2000x1000 with 0 Axes>
```



In [76]:

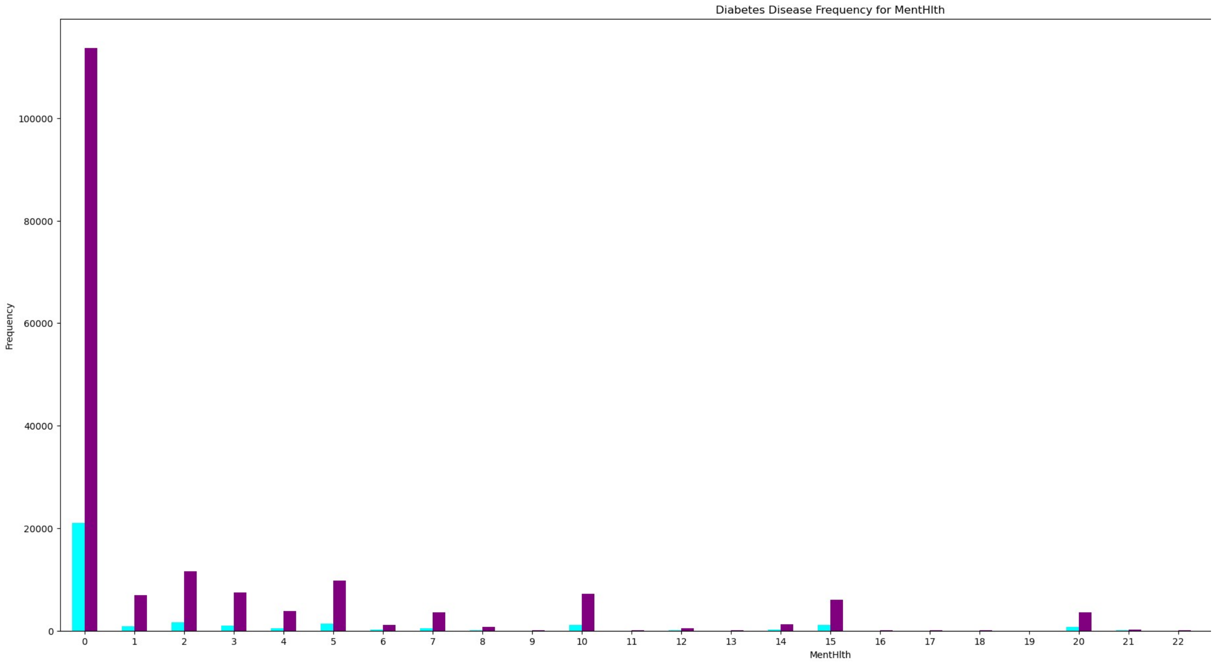
```
plt.figure(figsize=(20,10))
sns.countplot(data,x="BMI",hue="Diabetes_binary_str")
plt.title("Relation b/w BMI and Diabetes ")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2001d497850>
```



In [77]:

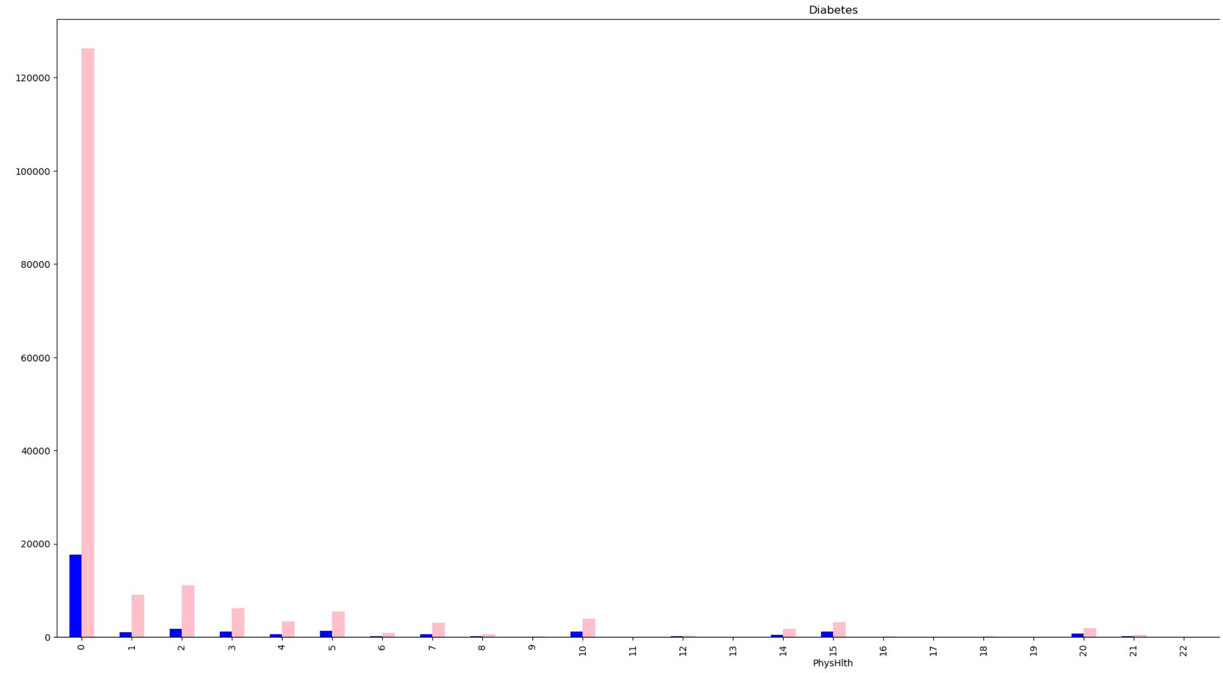
```
pd.crosstab(data.MentHlth,data.Diabetes_binary_str).plot(kind="bar",figsize=
plt.title("Diabetes Disease Frequency for MentHlth")
plt.xlabel('MentHlth')
plt.xticks(rotation=0)
plt.ylabel('Frequency')
plt.show()
```



In [78]:

```
pd.crosstab(data.PhysHlth,data.Diabetes_binary_str).plot(kind="bar",figsize=
plt.title('Diabetes')
```

```
Text(0.5, 1.0, 'Diabetes')
```

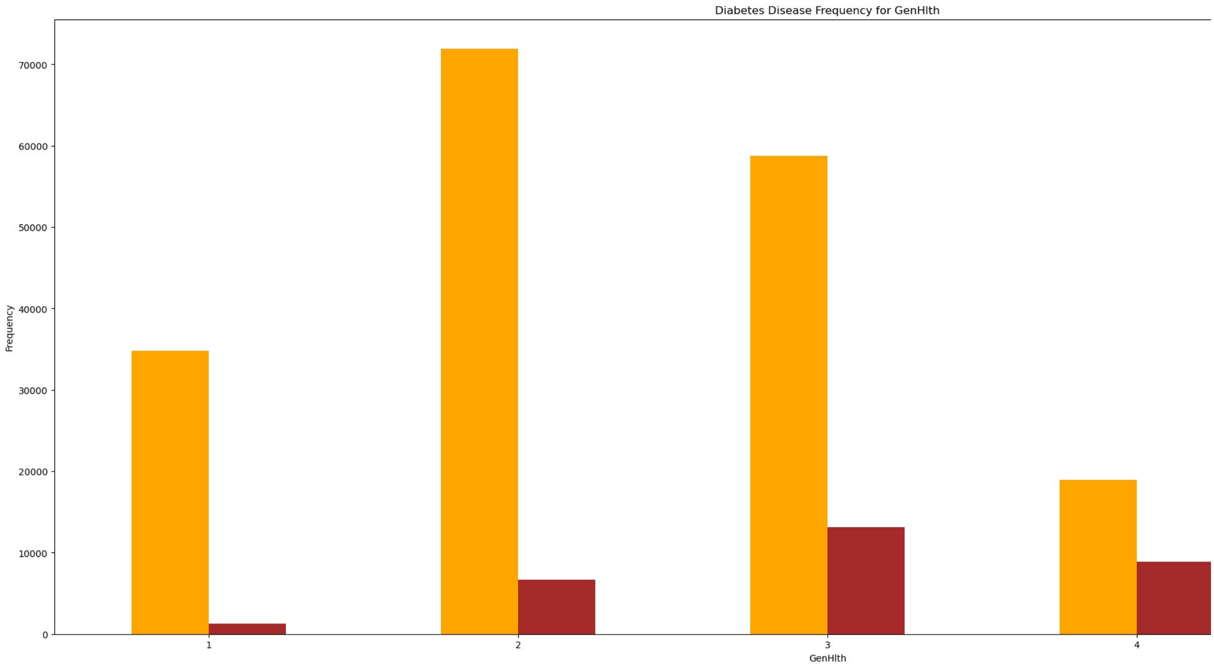


In [79]:

```
import pandas as pd
```

In [80]:

```
pd.crosstab(data.GenHlth,data.Diabetes_binary).plot(kind='bar',figsize=(30,1
plt.title('Diabetes Disease Frequency for GenHlth')
plt.xticks(rotation=0)
plt.ylabel('Frequency')
plt.show()
```



Feature Selections

data.drop(['Diabetes_binary','Diabetes_binary_str'],axis=1).corrwith(data.Diabe
(20,8) ,title="Correlation with Diabetes_binary",color='Purple');

VIF Test

In [81]:

```
import pandas as pd
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant  # Make sure to import add_
```

In [ ]:

In [36]:

```
def calc_VIF(x):
    vif=pd.DataFrame()
    vif['variables']=x.columns
    vif["VIF"]=[variance_inflation_factor(x.values,i) for i in range(x.shape
    return(vif)
X=add_constant(original_data)
ds=pd.Series([variance_inflation_factor(X.values,i) for i in range(X.shape[1
print(ds)
```

```
const                   122.078775
Diabetes_binary           1.187221
HighBP                    1.332464
HighChol                  1.175202
CholCheck                 1.048253
BMI                       1.174940
Smoker                    1.098596
Stroke                    1.069496
HeartDiseaseorAttack      1.158434
PhysActivity              1.200475
Fruits                    1.082997
Veggies                   1.084989
HvyAlcoholConsump         1.025476
AnyHealthcare             1.129791
NoDocbcCost               1.143437
GenHlth                   1.716120
MentHlth                  1.231285
PhysHlth                  1.536002
DiffWalk                  1.473067
Sex                       1.074073
Age                       1.425164
Education                 1.296028
Income                    1.454653
dtype: float64
```

In [37]:

```
X=original_data.iloc[:,1:]
Y=original_data.iloc[:,0]
```

In [38]:

```
fs=SelectKBest(score_func=f_classif,k=10)
X_selected=fs.fit_transform(X,Y)
print(X_selected.shape)
```

```
(236378, 10)
```

In [39]:

```
pd.DataFrame(X_selected).head(3)
```

	0	1	2	3	4	5	6	7	8	9
0	0.0	1.0	15.0	0.0	0.0	5.0	20.0	0.0	11.0	5.0
1	1.0	0.0	28.0	1.0	0.0	2.0	0.0	0.0	11.0	3.0
2	1.0	1.0	33.0	0.0	1.0	2.0	0.0	0.0	9.0	7.0

Chi Square

In [44]:

```
BestFeatures = SelectKBest(score_func=chi2,k=10)
fit=BestFeatures.fit(X,Y)
df_scores = pd.DataFrame(fit.scores_)
df_columns=pd.DataFrame(X.columns)
f_scores =pd.concat([df_columns,df_scores],axis=1)
f_scores.columns = ['Feature','Score']
f_scores
```

	Feature	Score
0	HighBP	9315.114284
1	HighChol	5487.828992
2	CholCheck	40.755005
3	BMI	14129.012544
4	Smoker	451.292612
5	Stroke	2343.124171
6	HeartDiseaseorAttack	6638.154494
7	PhysActivity	1142.297495
8	Fruits	73.710068
9	Veggies	86.314236
10	HvyAlcoholConsump	724.271381
11	AnyHealthcare	6.774205
12	NoDocbcCost	39.266735
13	GenHlth	7847.210017
14	MentHlth	4865.335431
15	PhysHlth	109774.699301
16	DiffWalk	8782.595881
17	Sex	147.641195
18	Age	11483.365130
19	Education	430.918003
20	Income	4417.444380

In [46]:

```
print(f_scores.nlargest(16,'Score'))
```

```
                 Feature          Score
15              PhysHlth  109774.699301
3                    BMI   14129.012544
18                   Age   11483.365130
0                 HighBP    9315.114284
16              DiffWalk    8782.595881
13               GenHlth    7847.210017
6   HeartDiseaseorAttack    6638.154494
1               HighChol    5487.828992
14              MentHlth    4865.335431
20                Income    4417.444380
5                 Stroke    2343.124171
7           PhysActivity    1142.297495
10     HvyAlcoholConsump     724.271381
4                 Smoker     451.292612
19             Education     430.918003
17                   Sex     147.641195
```

In [ ]: