

HEART DISEASE

PROBLEM STATEMENT: TO PREDICT AND ANALYZE

```
In [1]: import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="white")#white background style for Seaborn plots
sns.set(style="whitegrid",color_codes=True)
import warnings
warnings.simplefilter(action='ignore')
```

```
In [2]: df=pd.read_csv(r"C:\Users\manis\Downloads\framingham_heart_disease.csv")
df
```

```
Out[2]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	pre
0	1	39	4.0	0	0.0	0.0	0	
1	0	46	2.0	0	0.0	0.0	0	
2	1	48	1.0	1	20.0	0.0	0	
3	0	61	3.0	1	30.0	0.0	0	
4	0	46	3.0	1	23.0	0.0	0	
...
4233	1	50	1.0	1	1.0	0.0	0	
4234	1	51	3.0	1	43.0	0.0	0	
4235	0	48	2.0	1	20.0	NaN	0	
4236	0	44	1.0	1	15.0	0.0	0	
4237	0	52	2.0	0	0.0	0.0	0	

4238 rows × 16 columns

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

Out[3]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalen
0	1	39	4.0	0	0.0	0.0	0	
1	0	46	2.0	0	0.0	0.0	0	
2	1	48	1.0	1	20.0	0.0	0	
3	0	61	3.0	1	30.0	0.0	0	
4	0	46	3.0	1	23.0	0.0	0	



In [4]: `df.shape`

Out[4]: (4238, 16)

In [5]: `df.describe`

Out[5]: <bound method NDFrame.describe of

	male	age	education	currentSmoker	ci
gsPerDay					
BPMeds					
0	1	39	4.0	0	0.0 0.0 0.0 \
1	0	46	2.0	0	0.0 0.0
2	1	48	1.0	1	20.0 0.0
3	0	61	3.0	1	30.0 0.0
4	0	46	3.0	1	23.0 0.0
...
4233	1	50	1.0	1	1.0 0.0
4234	1	51	3.0	1	43.0 0.0
4235	0	48	2.0	1	20.0 NaN
4236	0	44	1.0	1	15.0 0.0
4237	0	52	2.0	0	0.0 0.0

	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI
0	0	0	0	195.0	106.0	70.0	26.97 \
1	0	0	0	250.0	121.0	81.0	28.73
2	0	0	0	245.0	127.5	80.0	25.34
3	0	1	0	225.0	150.0	95.0	28.58
4	0	0	0	285.0	130.0	84.0	23.10
...
4233	0	1	0	313.0	179.0	92.0	25.97
4234	0	0	0	207.0	126.5	80.0	19.71
4235	0	0	0	248.0	131.0	72.0	22.00
4236	0	0	0	210.0	126.5	87.0	19.16
4237	0	0	0	269.0	133.5	83.0	21.47

	heartRate	glucose	TenYearCHD
0	80.0	77.0	0
1	95.0	76.0	0
2	75.0	70.0	0
3	65.0	103.0	1
4	85.0	85.0	0
...
4233	66.0	86.0	1
4234	65.0	68.0	0
4235	84.0	86.0	0
4236	86.0	NaN	0
4237	80.0	107.0	0

[4238 rows x 16 columns]>

In [6]: df.info()

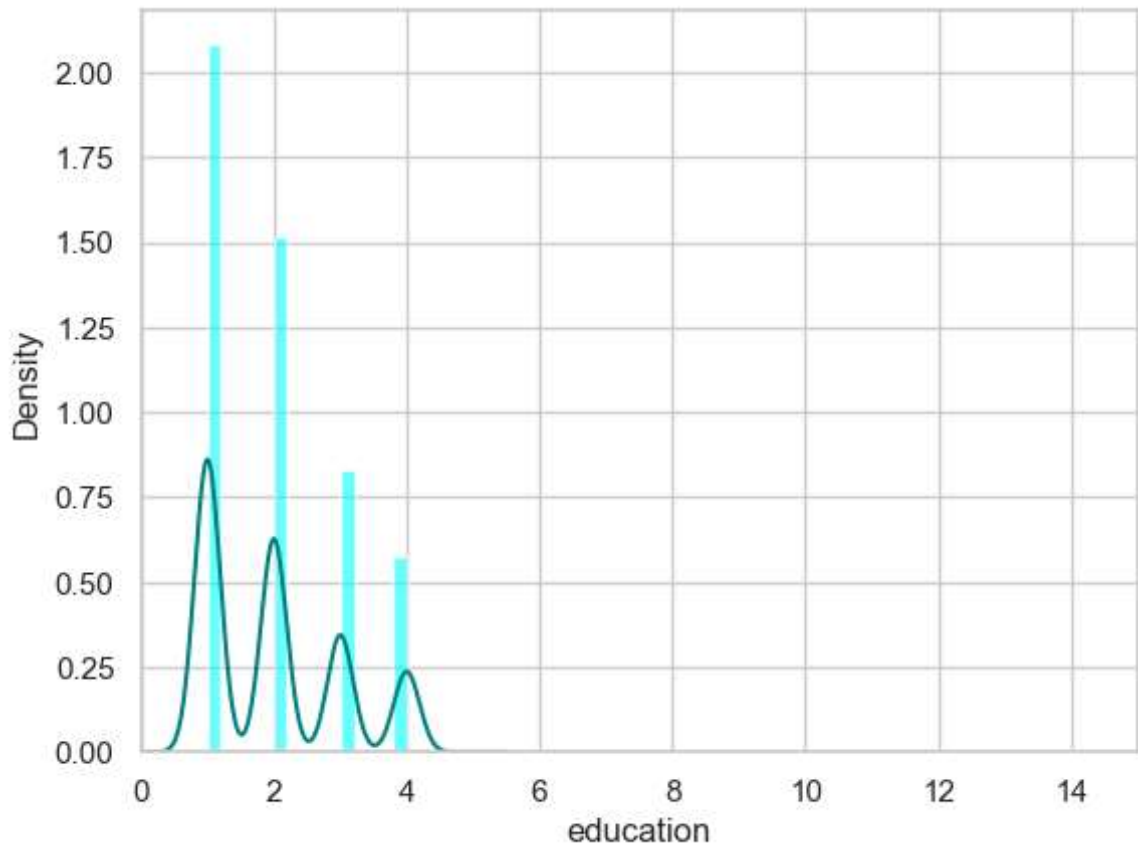
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   male                  4238 non-null   int64
1   age                   4238 non-null   int64
2   education             4133 non-null   float64
3   currentSmoker         4238 non-null   int64
4   cigsPerDay            4209 non-null   float64
5   BPMeds                4185 non-null   float64
6   prevalentStroke       4238 non-null   int64
7   prevalentHyp          4238 non-null   int64
8   diabetes              4238 non-null   int64
9   totChol               4188 non-null   float64
10  sysBP                 4238 non-null   float64
11  diaBP                 4238 non-null   float64
12  BMI                   4219 non-null   float64
13  heartRate             4237 non-null   float64
14  glucose               3850 non-null   float64
15  TenYearCHD            4238 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

TO FIND THE MISSING VALUES

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

```
Out[7]: male                0
age                0
education          105
currentSmoker      0
cigsPerDay         29
BPMeds             53
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            50
sysBP              0
diaBP              0
BMI               19
heartRate          1
glucose            388
TenYearCHD         0
dtype: int64
```

```
In [8]: ax=df["education"].hist(bins=15,density=True,stacked=True,color='cyan',alpha=0.6)
df["education"].plot(kind='density',color='teal')
ax.set(xlabel='education')
plt.xlim(-0,15)
plt.show()
```



```
In [9]: print(df["education"].mean(skipna=True))
        print(df["education"].median(skipna=True))
```

```
1.9789499153157513
2.0
```

```
In [10]: print((df['glucose'].isnull().sum()/df.shape[0]*100))
```

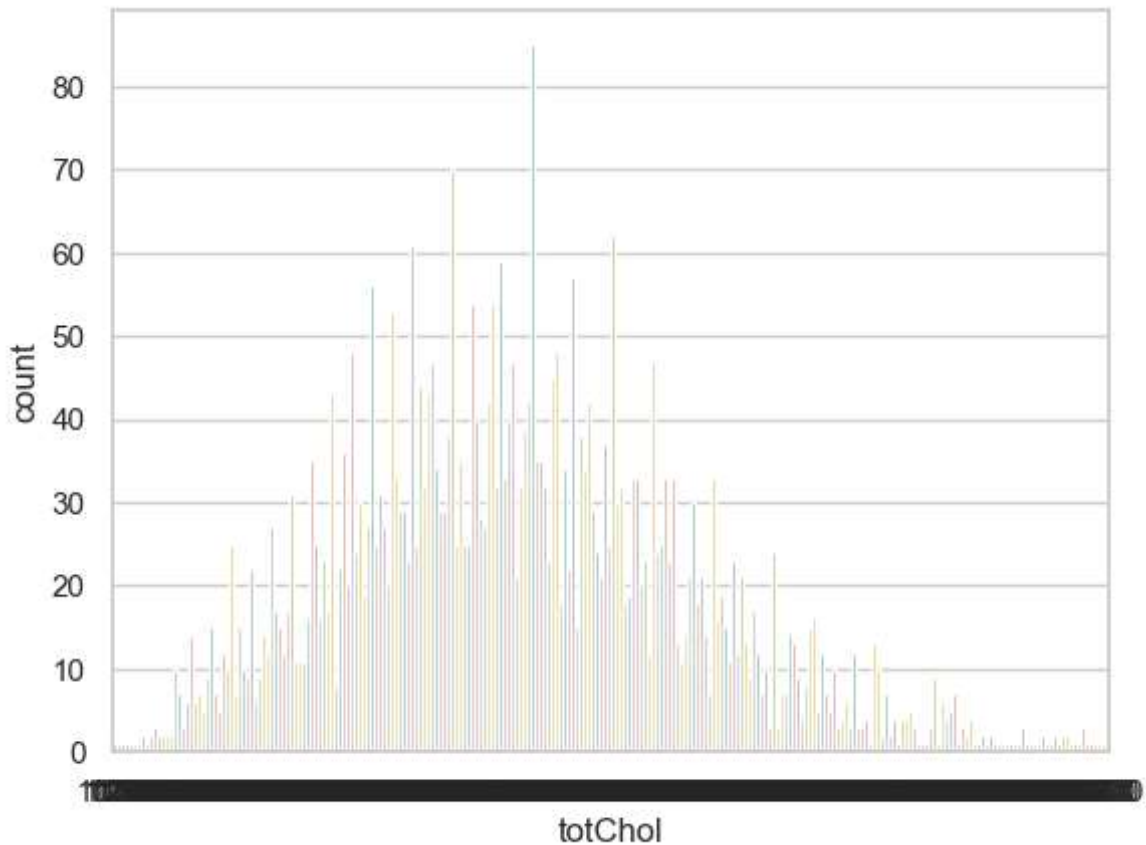
```
9.155261915998112
```

```
In [11]: print((df['totChol'].isnull().sum()/df.shape[0]*100))
```

```
1.1798017932987257
```

```
In [12]: print(df['totChol'].value_counts())
        sns.countplot(x='totChol',data=df,palette='Set2')
        plt.show()
```

```
totChol
240.0    85
220.0    70
260.0    62
210.0    61
232.0    59
..
392.0     1
405.0     1
359.0     1
398.0     1
119.0     1
Name: count, Length: 248, dtype: int64
```



```
In [13]: print(df['totChol'].value_counts().idxmax())
```

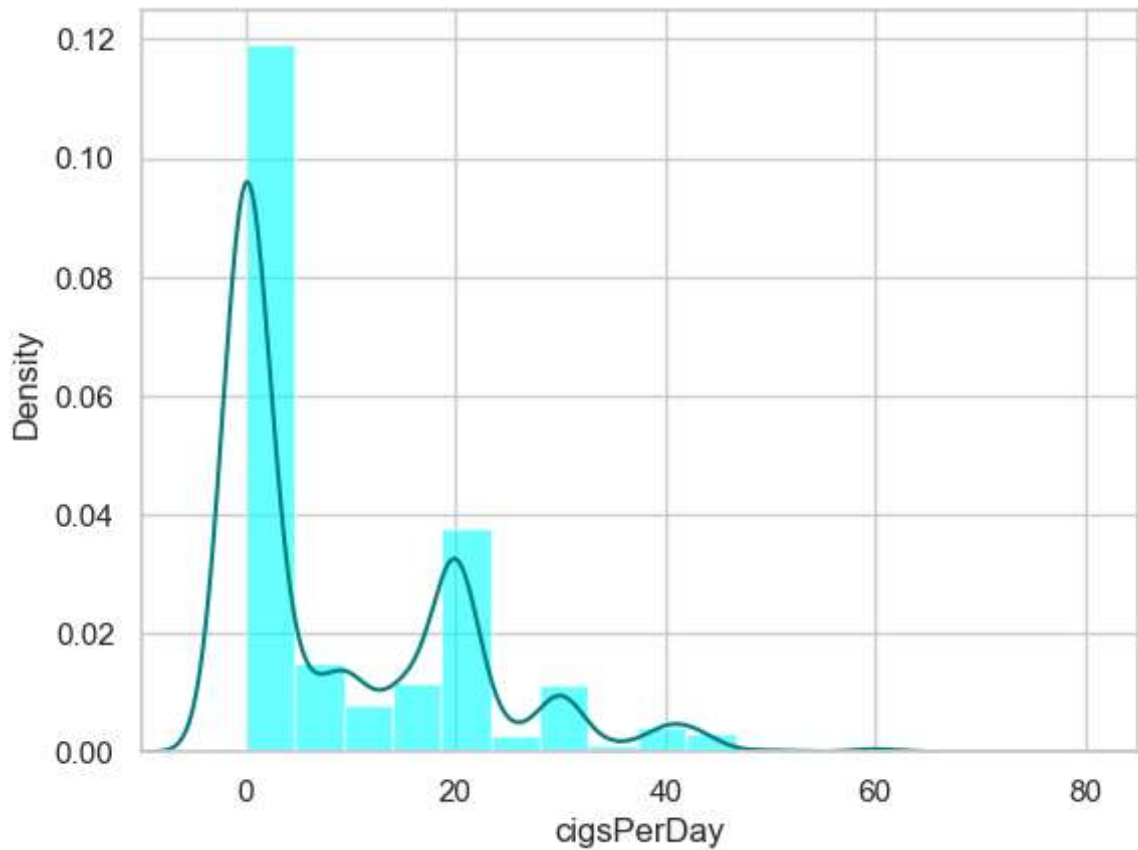
240.0

```
In [14]: data=df.copy()
data["education"].fillna(df["education"].median(skipna=True),inplace=True)
data["totChol"].fillna(df["totChol"].value_counts().idxmax(),inplace=True)
data.drop('glucose',axis=1,inplace=True)
```

```
In [15]: data.isnull().sum()
```

```
Out[15]: male          0
age              0
education        0
currentSmoker    0
cigsPerDay       29
BPMeds           53
prevalentStroke  0
prevalentHyp     0
diabetes         0
totChol          0
sysBP            0
diaBP            0
BMI              19
heartRate        1
TenYearCHD       0
dtype: int64
```

```
In [16]: ax=df["cigsPerDay"].hist(bins=15,density=True,stacked=True,color='cyan',alpha=0.
df["cigsPerDay"].plot(kind='density',color='teal')
ax.set(xlabel='cigsPerDay')
plt.xlim(-10,85)
plt.show()
```



```
In [17]: print(df["cigsPerDay"].mean(skipna=True))  
         print(df["cigsPerDay"].median(skipna=True))
```

```
9.003088619624615  
0.0
```

```
In [18]: print((df['BPMeds'].isnull().sum()/df.shape[0]*100))
```

```
1.2505899008966492
```

```
In [19]: print((df['BMI'].isnull().sum()/df.shape[0]*100))
```

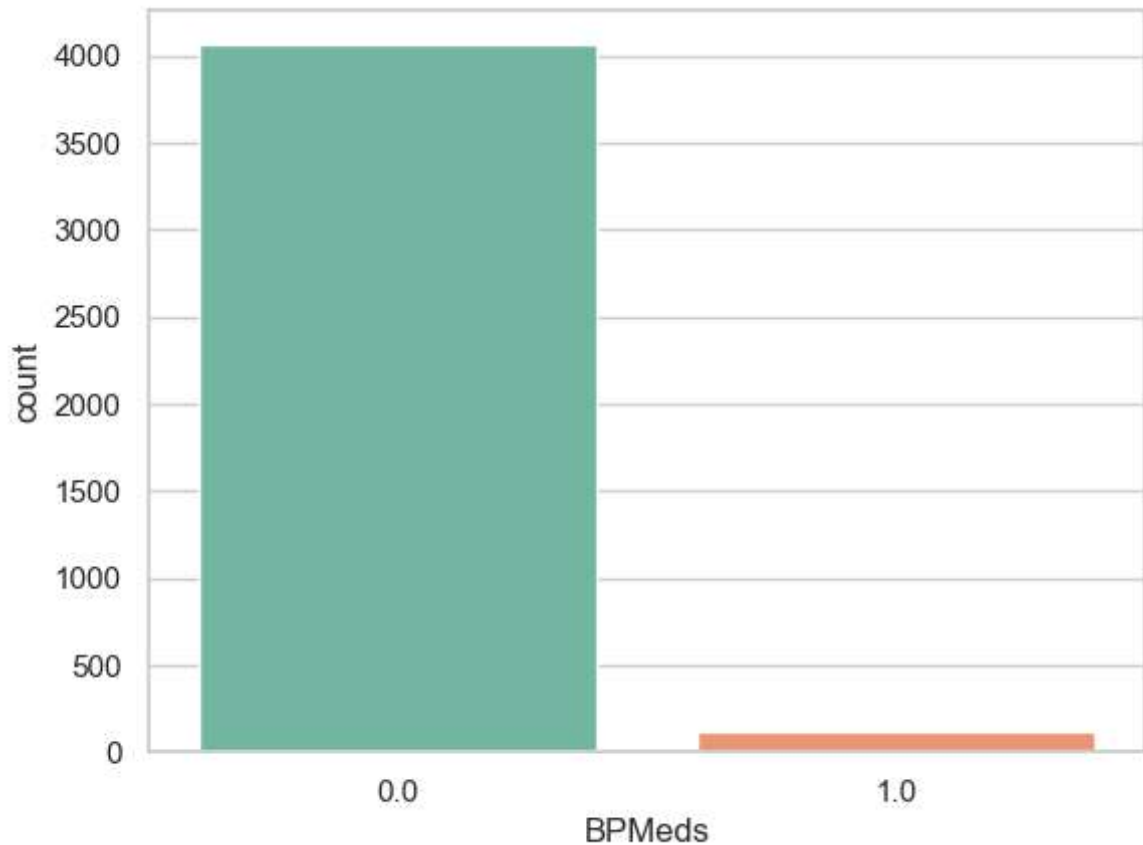
```
0.4483246814535158
```

```
In [20]: print((df['heartRate'].isnull().sum()/df.shape[0]*100))
```

```
0.023596035865974516
```

```
In [21]: print(df['BPMeds'].value_counts())  
         sns.countplot(x='BPMeds',data=df,palette='Set2')  
         plt.show()
```

```
BPMeds  
0.0    4061  
1.0     124  
Name: count, dtype: int64
```



```
In [22]: print(df['heartRate'].value_counts().idxmax())
```

75.0

```
In [23]: data=df.copy()
data["cigsPerDay"].fillna(df["cigsPerDay"].median(skipna=True),inplace=True)
data["BPMeds"].fillna(df["BPMeds"].value_counts().idxmax(),inplace=True)
data["education"].fillna(df["education"].median(skipna=True),inplace=True)
data["totChol"].fillna(df["totChol"].value_counts().idxmax(),inplace=True)
data.drop('glucose',axis=1,inplace=True)
data.drop('BMI',axis=1,inplace=True)
data.drop('heartRate',axis=1,inplace=True)
```

```
In [24]: data.isnull().sum()
```

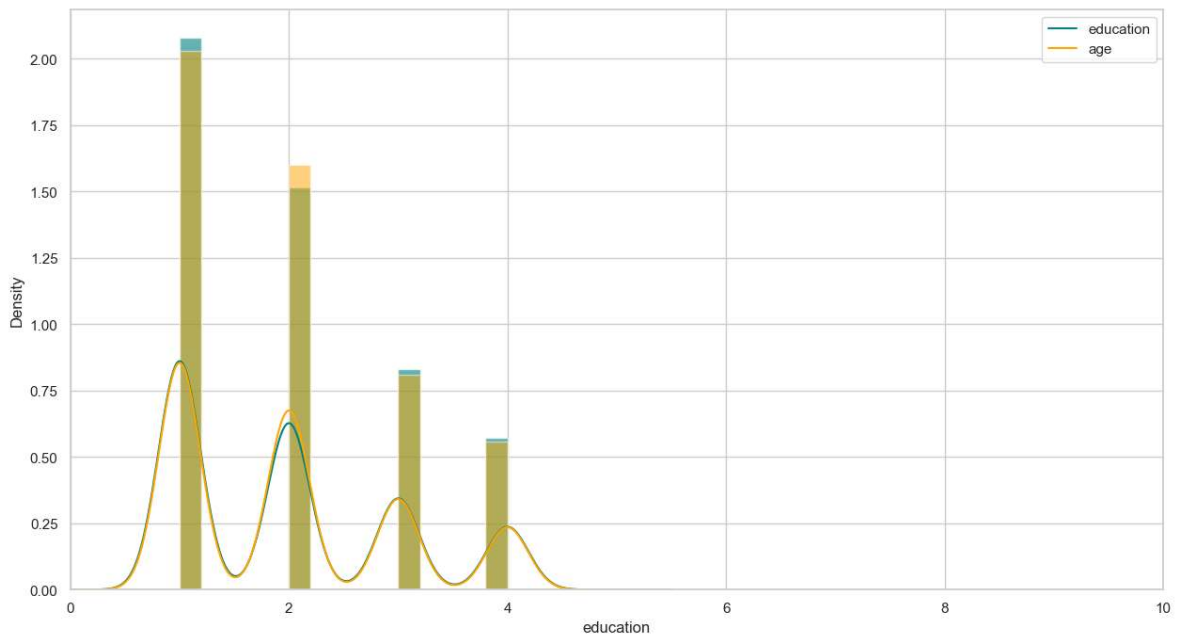
```
Out[24]: male          0
age          0
education     0
currentSmoker 0
cigsPerDay    0
BPMeds        0
prevalentStroke 0
prevalentHyp  0
diabetes      0
totChol       0
sysBP         0
diaBP         0
TenYearCHD    0
dtype: int64
```

```
In [25]: data.head()
```


Out[25]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.0	0	0
1	0	46	2.0	0	0.0	0.0	0	0
2	1	48	1.0	1	20.0	0.0	0	0
3	0	61	3.0	1	30.0	0.0	0	0
4	0	46	3.0	1	23.0	0.0	0	0

```
In [26]: plt.figure(figsize=(15,8))
ax=df["education"].hist(bins=15,density=True,stacked=True,color='teal',alpha=0.6)
df["education"].plot(kind='density',color='teal')
ax=data["education"].hist(bins=15,density=True,stacked=True,color='orange',alpha=0.6)
data["education"].plot(kind='density',color='orange')
ax.legend(["education","age"])
ax.set(xlabel='education')
plt.xlim(-0,10)
plt.show()
```



```
In [27]: data['Disease']=np.where((data["prevalentHyp"]+data["prevalentStroke"])>0,0,1)
data.drop('prevalentHyp',axis=1,inplace=True)
data.drop('prevalentStroke',axis=1,inplace=True)
```

```
In [28]: training=pd.get_dummies(data,columns=["currentSmoker","totChol","sysBP"])
training.drop('TenYearCHD',axis=1,inplace=True)
training.drop('male',axis=1,inplace=True)
training.drop('diaBP',axis=1,inplace=True)
final_train=training
final_train.head()
```

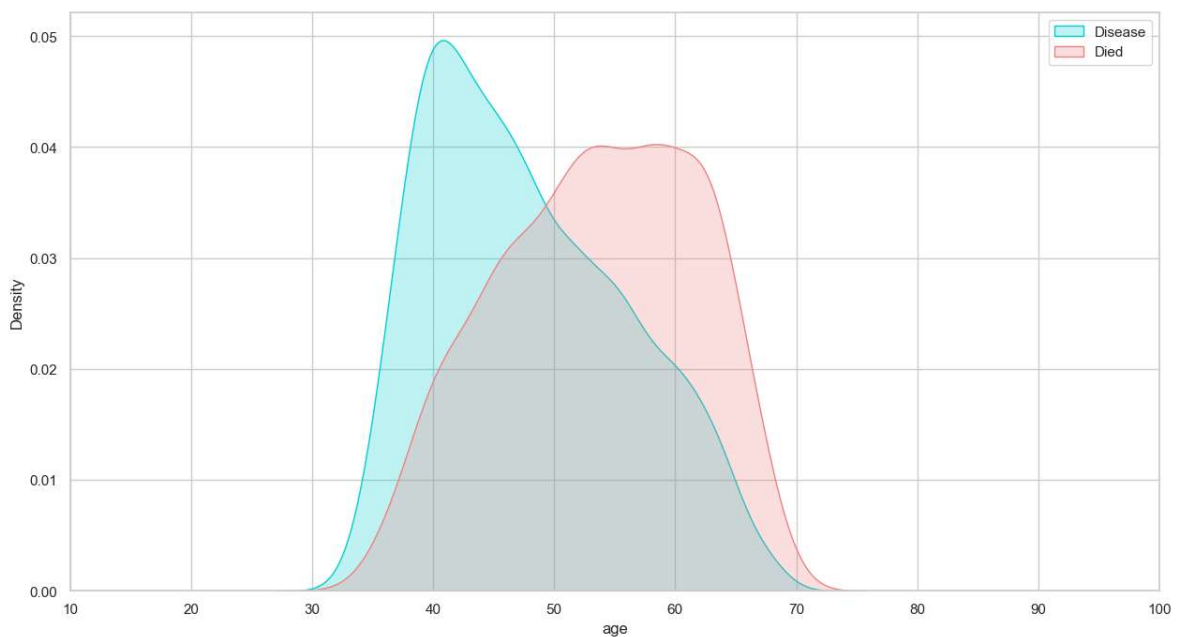
Out[28]:

	age	education	cigsPerDay	BPMeds	diabetes	Disease	currentSmoker_0	currentSr
0	39	4.0	0.0	0.0	0	1	True	
1	46	2.0	0.0	0.0	0	1	True	
2	48	1.0	20.0	0.0	0	1	False	
3	61	3.0	30.0	0.0	0	0	False	
4	46	3.0	23.0	0.0	0	1	False	

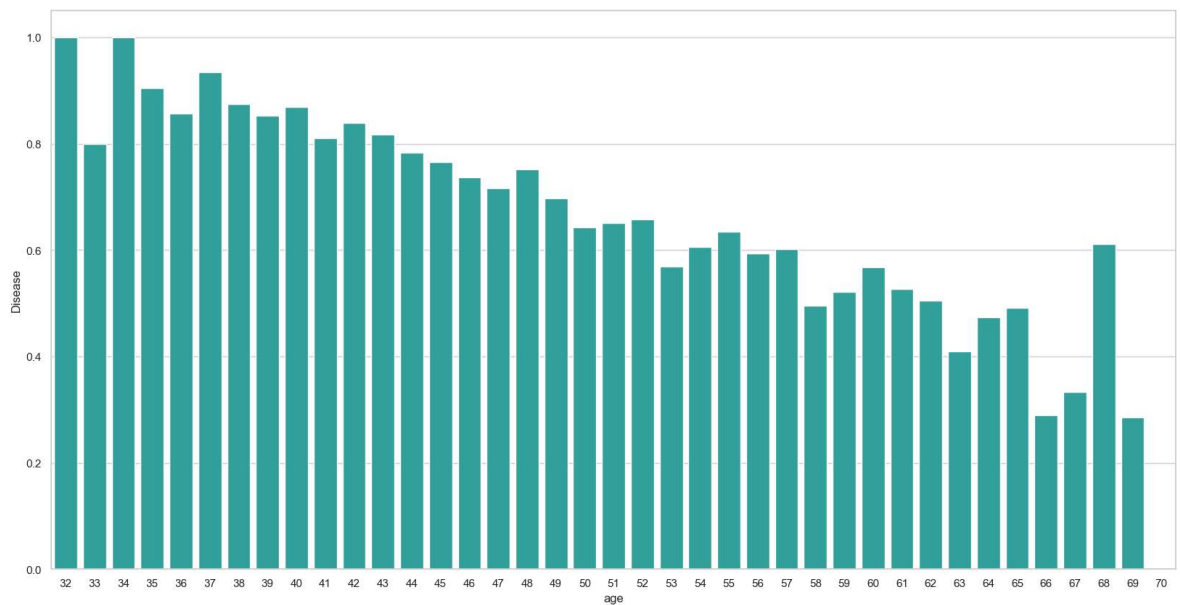
5 rows × 490 columns

EXPLORATORY DATA ANALYSIS

```
In [56]: plt.figure(figsize=(15,8))
ax = sns.kdeplot(final_train["age"][final_train.Disease == 1], color="darkturqu
sns.kdeplot(final_train["age"][final_train.Disease == 0], color="lightcoral", sh
plt.legend(['Disease', 'Died'])
ax.set(xlabel='age')
plt.xlim(10,100)
plt.show()
```



```
In [61]: plt.figure(figsize=(20,10))
avg_survival_byage = final_train[["age", "Disease"]].groupby(['age'], as_index=F
g = sns.barplot(x='age', y='Disease', data=avg_survival_byage, color="LightSeaGr
plt.show()
```

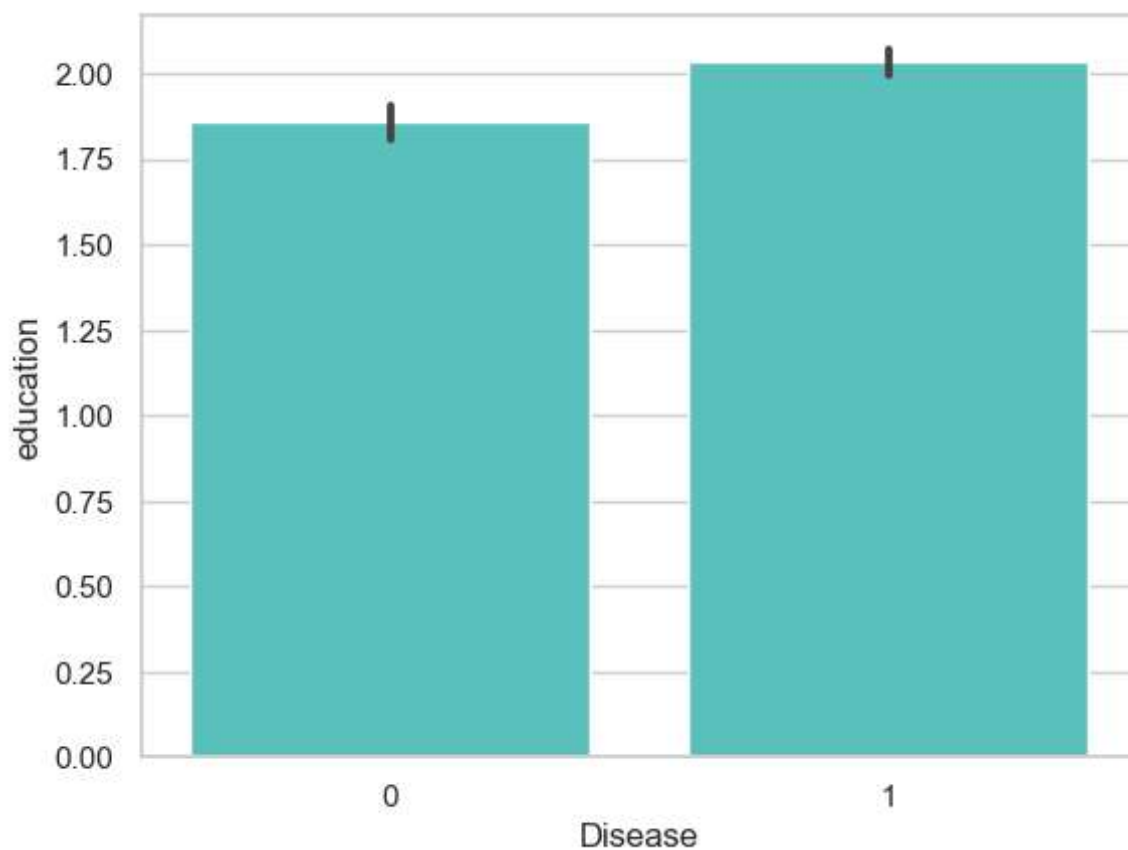


```
In [63]: final_train['IsMinor']=np.where(final_train['age']<=16, 1, 0)
print(final_train['IsMinor'])
```

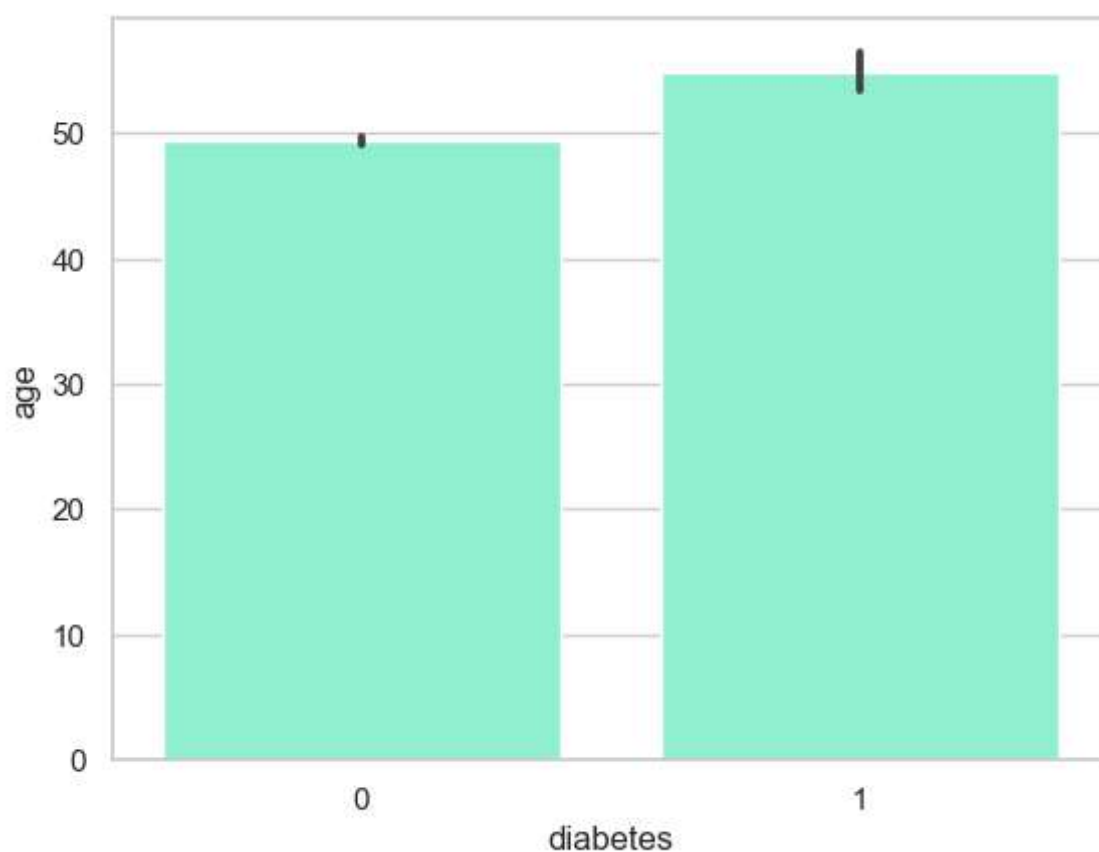
```
0      0
1      0
2      0
3      0
4      0
..
4233   0
4234   0
4235   0
4236   0
4237   0
```

Name: IsMinor, Length: 4238, dtype: int32

```
In [66]: sns.barplot(x='Disease', y='education', data=final_train, color="mediumturquoise")
plt.show()
```



```
In [69]: import seaborn as sns
import matplotlib.pyplot as plt
# Assuming 'train_df' is your DataFrame containing the data
sns.barplot(x='diabetes', y='age', data=df, color='aquamarine')
plt.show()
```



In []: