

DATE:- 2-6-2023 __HEART DISEASE DATA SET USING **LOGISTIC REGRESSION**

```
In [ ]: 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")
sns.set(style="whitegrid", color_codes=True)
import warnings
warnings.simplefilter(action='ignore')
```

```
In [ ]: df=pd.read_csv(r"/content/framingham.csv")
df
```

Out[]:

	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	1
4	0	46	3.0	1	23.0	0.0	0	0
...
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
4235	0	48	2.0	1	20.0	NaN	0	0
4236	0	44	1.0	1	15.0	0.0	0	0
4237	0	52	2.0	0	0.0	0.0	0	0

4238 rows × 16 columns

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

Out[]:

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

```
In [ ]: df.shape
```

```
Out[ ]: (4238, 16)
```

In []: df.describe

```
Out[ ]: <bound method NDFrame.describe of
erDay BPMeds \
0      1   39      4.0
1      0   46      2.0
2      1   48      1.0
3      0   61      3.0
4      0   46      3.0
...
4233    1   50      1.0
4234    1   51      3.0
4235    0   48      2.0
4236    0   44      1.0
4237    0   52      2.0

  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]>
```

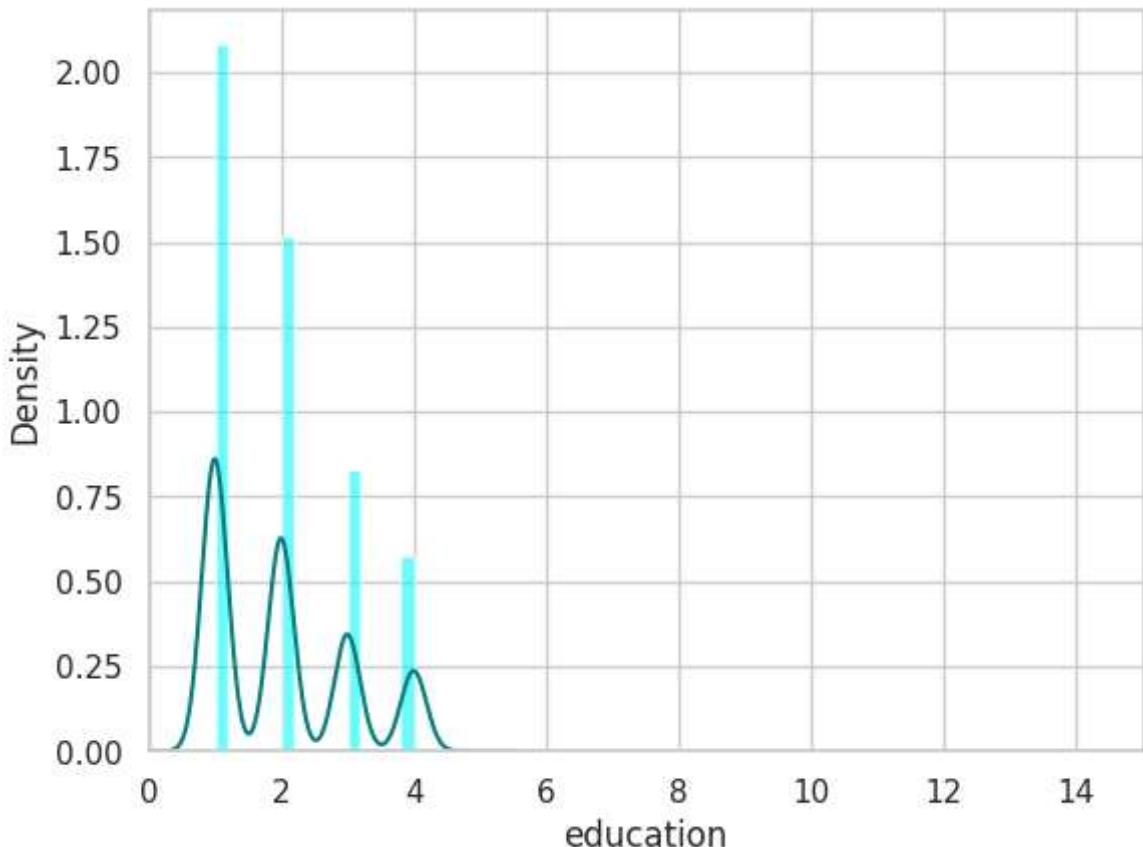
In []: df.info()

```
<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
```

In []: df.isna().any()

```
Out[ ]: male      False
age       False
education True
currentSmoker False
cigsPerDay True
BPMeds    True
prevalentStroke False
prevalentHyp False
diabetes   False
totChol    True
sysBP     False
diaBP     False
BMI       True
heartRate  True
glucose    True
TenYearCHD False
dtype: bool
```

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

```
1.9789499153157513
2.0
```

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

```
9.155261915998112
```

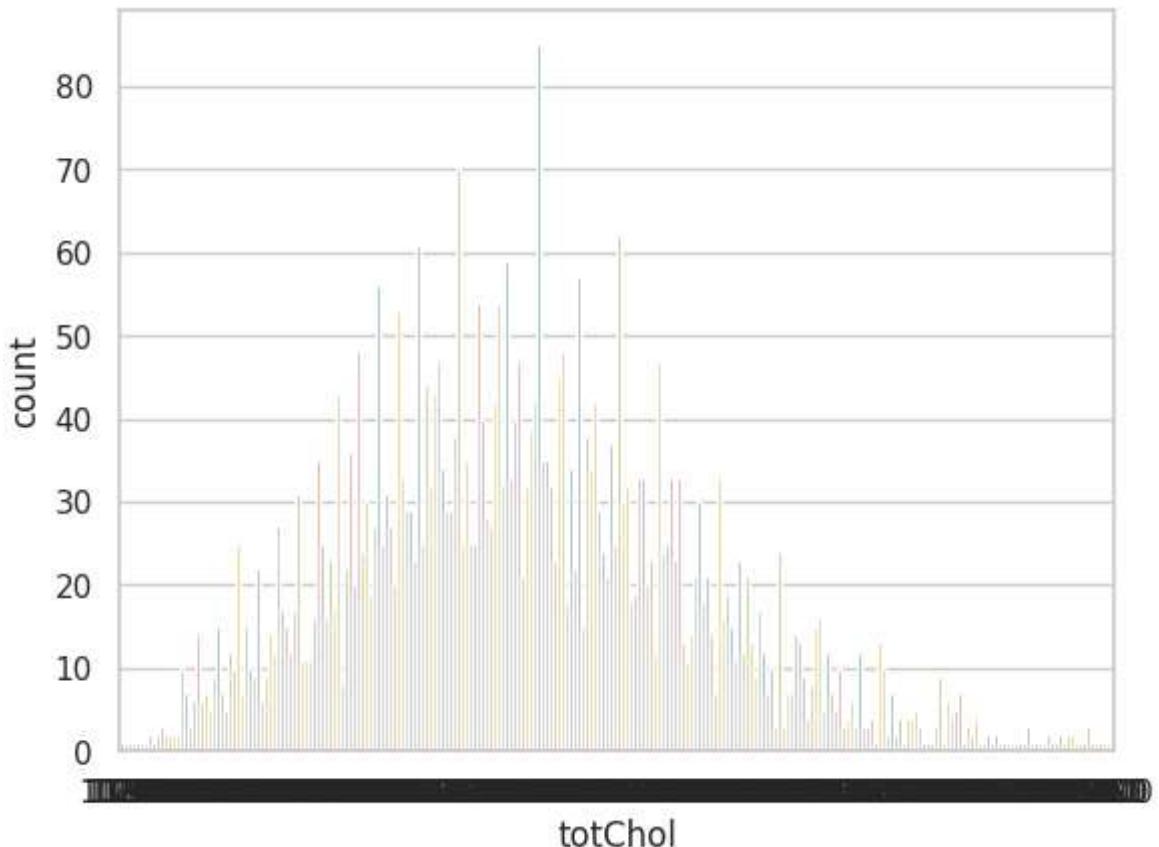
```
In [ ]: print(df['totChol'].isnull().sum()/df.shape[0]*100)
```

```
1.1798017932987257
```

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

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: totChol, Length: 248, dtype: int64



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

```
240.0
```

```
In [ ]: 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 [ ]: data.isna().sum()
```

```
Out[ ]:
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 [ ]: pd.set_option('display.max_rows',4238)
pd.set_option('display.max_columns',16)
```

```
In [ ]: pd.set_option('display.width',50)
```

```
In [ ]: print('This DataFrame has %d Rows and %d Columns'%(df.shape))
```

This DataFrame has 4238 Rows and 16 Columns

```
In [ ]: features_matrix=df.iloc[:,0:15]
target_vector=df.iloc[:, -2]
```

```
In [ ]: print('The Features Matrix Has %d Rows And %d Column(s)'%(features_matrix.shape))
print('The Target Matrix Has %d Rows And %d Column(s)'%np.array(target_vector).re
```

The Features Matrix Has 4238 Rows And 15 Column(s)
The Target Matrix Has 4238 Rows And 1 Column(s)

```
In [ ]: df['education'].mean()
```

```
Out[ ]: 1.9789499153157513
```

```
In [ ]: df['cigsPerDay'].mean()
```

```
Out[ ]: 9.003088619624615
```

```
In [ ]: df['heartRate'].median()
```

```
Out[ ]: 75.0
```

```
In [ ]: df['BPMed'] .mean()
```

```
Out[ ]: 0.02962962962962963
```

```
In [ ]: df["glucose"].fillna(df["glucose"].median(skipna=True),inplace=True)
df
```

Out[]:	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	1
4	0	46	3.0	1	23.0	0.0	0	0
5	0	43	2.0	0	0.0	0.0	0	1
6	0	63	1.0	0	0.0	0.0	0	0
7	0	45	2.0	1	20.0	0.0	0	0
8	1	52	1.0	0	0.0	0.0	0	1
9	1	43	1.0	1	30.0	0.0	0	1
10	0	50	1.0	0	0.0	0.0	0	0
11	0	43	2.0	0	0.0	0.0	0	0
12	1	46	1.0	1	15.0	0.0	0	1
13	0	41	3.0	0	0.0	1.0	0	1
14	0	39	2.0	1	9.0	0.0	0	0
15	0	38	2.0	1	20.0	0.0	0	1
16	1	48	3.0	1	10.0	0.0	0	1
17	0	46	2.0	1	20.0	0.0	0	0
18	0	38	2.0	1	5.0	0.0	0	0
19	1	41	2.0	0	0.0	0.0	0	0
20	0	42	2.0	1	30.0	0.0	0	0
21	0	43	1.0	0	0.0	0.0	0	0
22	0	52	1.0	0	0.0	0.0	0	0
23	0	52	3.0	1	20.0	0.0	0	0
24	1	44	2.0	1	30.0	0.0	0	1
25	1	47	4.0	1	20.0	0.0	0	0
26	0	60	1.0	0	0.0	0.0	0	0
27	1	35	2.0	1	20.0	0.0	0	1
28	0	61	3.0	0	0.0	0.0	0	1
29	0	60	1.0	0	0.0	0.0	0	0
30	1	36	4.0	1	35.0	0.0	0	0
31	1	43	4.0	1	43.0	0.0	0	0
32	0	59	1.0	0	0.0	0.0	0	1
33	1	61	NaN	1	5.0	0.0	0	0
34	1	54	1.0	1	20.0	0.0	0	1
35	1	37	2.0	0	0.0	0.0	0	1

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
4212	1	48	2.0	1	43.0	0.0	0	1
4213	1	48	2.0	0	0.0	0.0	0	0
4214	0	56	3.0	1	5.0	0.0	0	0
4215	0	63	1.0	0	0.0	0.0	0	1
4216	0	59	3.0	1	20.0	0.0	0	0
4217	0	63	1.0	0	0.0	0.0	0	1
4218	0	52	1.0	0	0.0	0.0	0	0
4219	1	48	1.0	0	0.0	0.0	0	0
4220	1	60	1.0	0	0.0	0.0	0	0
4221	1	50	1.0	0	0.0	0.0	0	0
4222	1	53	3.0	0	0.0	0.0	0	1
4223	1	56	4.0	0	0.0	1.0	0	1
4224	1	47	2.0	1	3.0	0.0	0	0
4225	1	45	4.0	1	43.0	0.0	0	0
4226	1	58	1.0	0	0.0	0.0	0	0
4227	1	43	4.0	1	20.0	0.0	0	0
4228	0	50	1.0	0	0.0	0.0	0	1
4229	0	51	3.0	1	20.0	0.0	0	1
4230	0	56	1.0	1	3.0	0.0	0	1
4231	1	58	3.0	0	0.0	0.0	0	1
4232	1	68	1.0	0	0.0	0.0	0	1
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
4235	0	48	2.0	1	20.0	NaN	0	0
4236	0	44	1.0	1	15.0	0.0	0	0
4237	0	52	2.0	0	0.0	0.0	0	0

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

```
Out[ ]: 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        0
         TenYearCHD    0
         dtype: int64
```

```
In [ ]: df['education'].fillna(df['education'].median(skipna=True),inplace=True)
df['totChol'].fillna(df['totChol'].median(skipna=True),inplace=True)
df['BMI'].fillna(df['BMI'].median(skipna=True),inplace=True)
df['heartRate'].fillna(df['heartRate'].median(skipna=True),inplace=True)
df['BPMeds'].fillna(df['BPMeds'].median(skipna=True),inplace=True)
df['cigsPerDay'].fillna(df['cigsPerDay'].median(skipna=True),inplace=True)
```

```
In [ ]: df.isna().any()
```

```
Out[ ]: male          False
         age          False
         education    False
         currentSmoker False
         cigsPerDay    False
         BPMeds       False
         prevalentStroke False
         prevalentHyp   False
         diabetes      False
         totChol       False
         sysBP         False
         diaBP         False
         BMI          False
         heartRate     False
         glucose        False
         TenYearCHD    False
         dtype: bool
```

```
In [ ]: df.drop('glucose',axis=1,inplace=True)
```

```
In [ ]: df.isna().any()
```

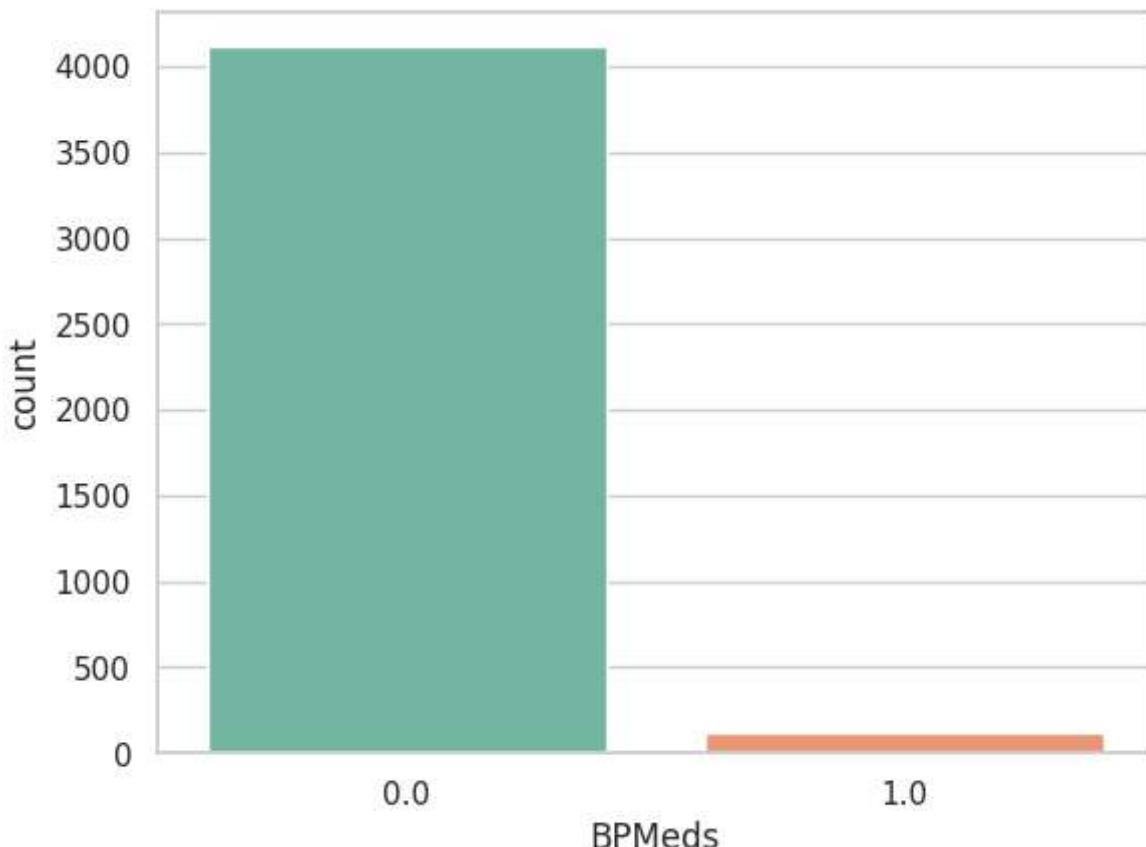
```
Out[ ]: male      False
         age       False
         education  False
         currentSmoker  False
         cigsPerDay   False
         BPMeds     False
         prevalentStroke  False
         prevalentHyp   False
         diabetes    False
         totChol     False
         sysBP       False
         diabP       False
         BMI        False
         heartRate   False
         TenYearCHD  False
         dtype: bool
```

```
In [ ]: print(df["cigsPerDay"].mean(skipna=True))
print(df["cigsPerDay"].median(skipna=True))
print((df['BPMeds'].isnull().sum()/df.shape[0]*100))
print((df['BMI'].isnull().sum()/df.shape[0]*100))
print((df['heartRate'].isnull().sum()/df.shape[0]*100))
```

```
8.941481831052384
0.0
0.0
0.0
0.0
```

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

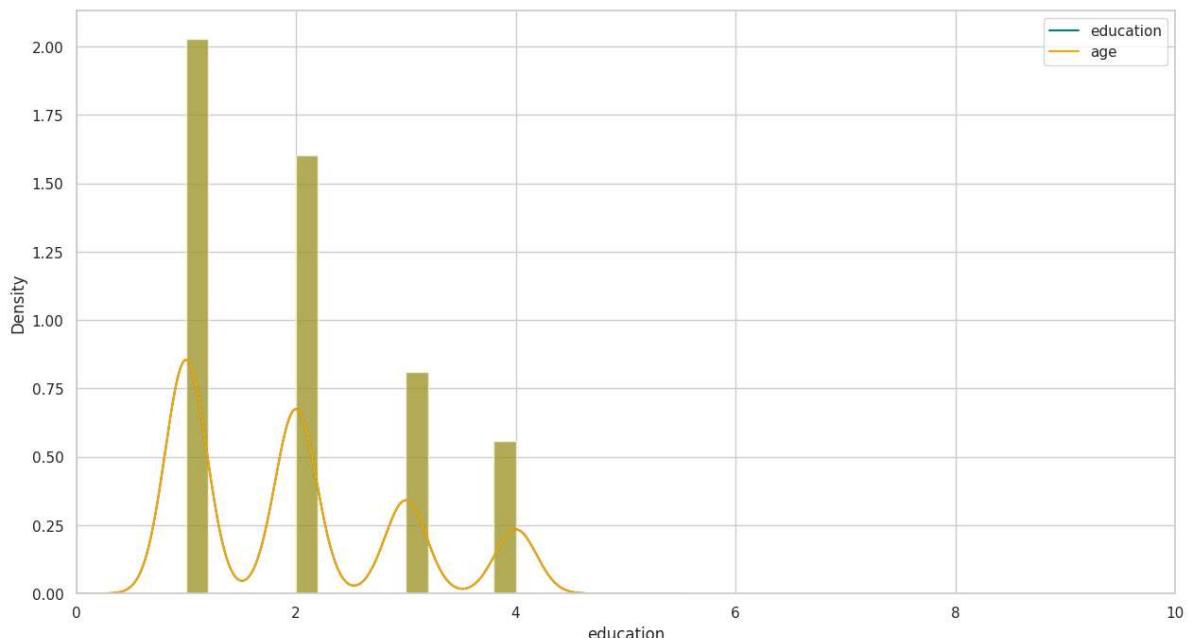
```
0.0    4114
1.0    124
Name: BPMeds, dtype: int64
```



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

75.0

```
In [ ]: 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 [ ]: 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)
```

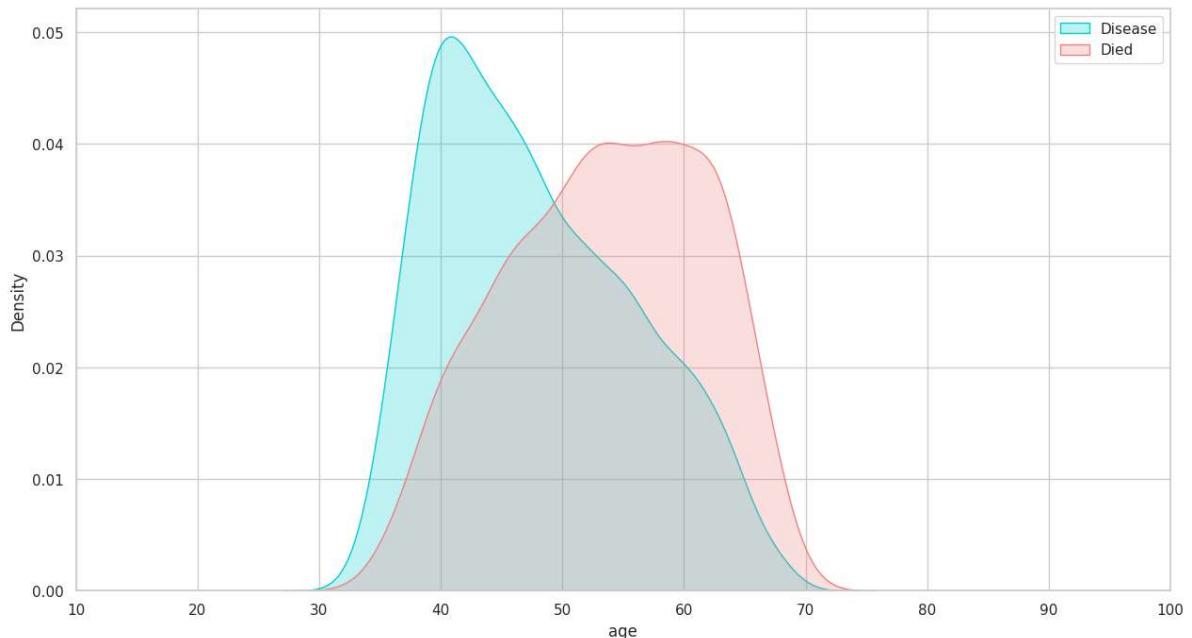
```
In [ ]: training=pd.get_dummies(data,columns=["currentSmoker","totChol","sysBP"])
training.drop('TenYearCHD',axis=1,inplace=True)
training.drop('male',axis=1,inplace=True)
training.drop('diaBP',axis=1,inplace=True)
final_train=training
final_train.head()
```

	age	education	cigsPerDay	BPMeds	diabetes	BMI	heartRate	Disease	...	sysBP_220.0	sys
0	39	4.0	0.0	0.0	0	26.97	80.0	1	...	0	
1	46	2.0	0.0	0.0	0	28.73	95.0	1	...	0	
2	48	1.0	20.0	0.0	0	25.34	75.0	1	...	0	
3	61	3.0	30.0	0.0	0	28.58	65.0	0	...	0	
4	46	3.0	23.0	0.0	0	23.10	85.0	1	...	0	

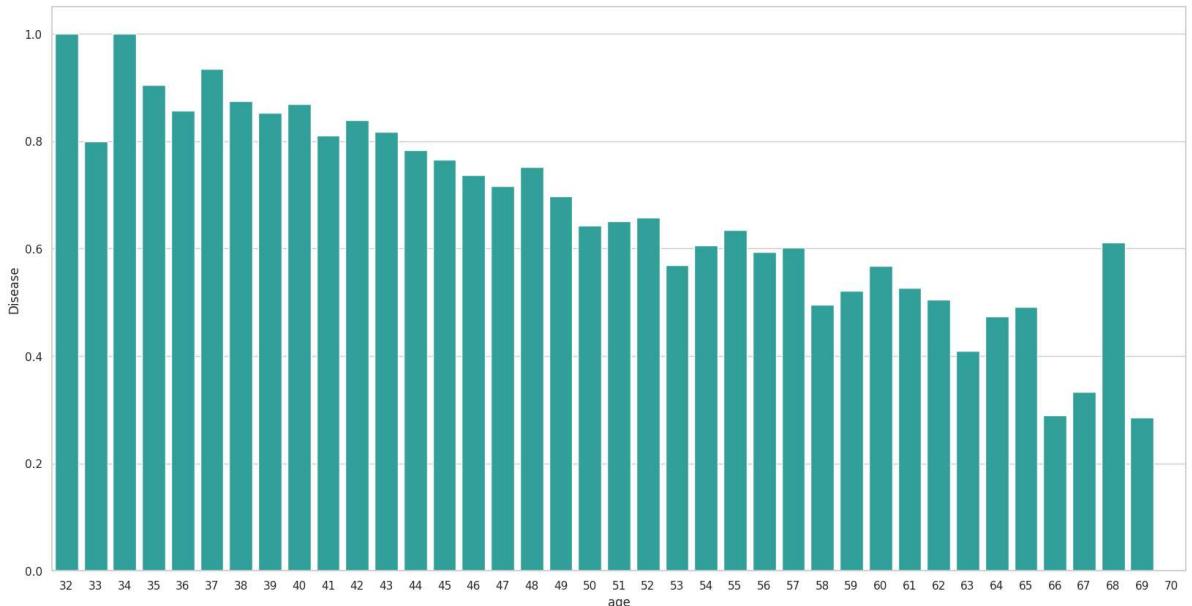
5 rows × 492 columns

EXPLORATORY DATA ANALYSIS

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



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

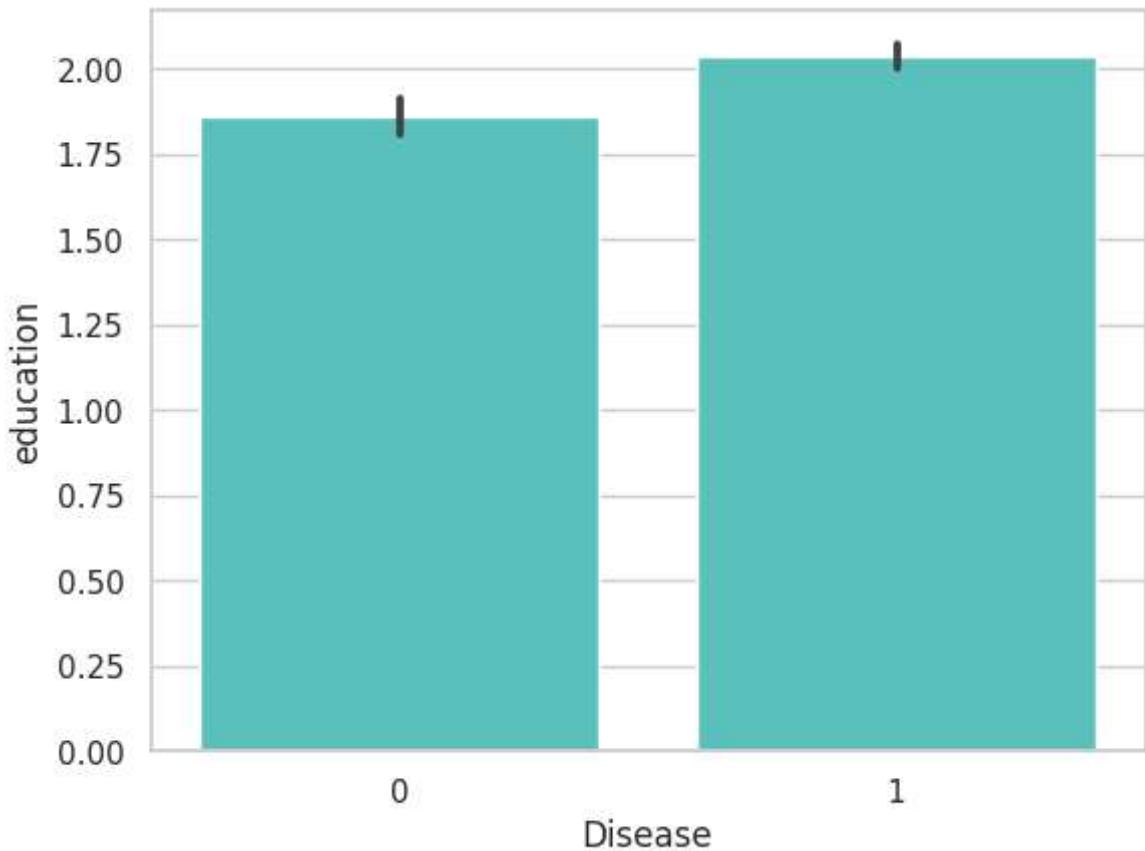


```
In [ ]: final_train['IsMinor']=np.where(final_train['age']<=16,1,0)
print(final_train['IsMinor'])
```

0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0
34	0
35	0
36	0
37	0
38	0
39	0
40	0
41	0
42	0
43	0
44	0
45	0
46	0
47	0
48	0
49	0
50	0
51	0
52	0
53	0
54	0
55	0
56	0
57	0
58	0
59	0
60	0
61	0
62	0
63	0

```
4224    0  
4225    0  
4226    0  
4227    0  
4228    0  
4229    0  
4230    0  
4231    0  
4232    0  
4233    0  
4234    0  
4235    0  
4236    0  
4237    0  
Name: IsMinor, dtype: int64
```

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



```
In [ ]: import seaborn as sns  
import matplotlib.pyplot as plt  
sns.barplot(x='diabetes',y='age',data=df,color="aquamarine")  
plt.show()
```

