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

In [2]:

```
df=pd.read_csv(r"C:\Users\DELL E5490\Downloads\heart disease.csv")
df
```

Out[2]:

```
      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
 ...   ...  ...        ...            ...         ...     ...              ...
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	prevalentHy
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  currentSmok
er  cigsPerDay  BPMeds
0        1   39        4.0              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    B
MI
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 [7]:

```
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 MISSING VALUES

In [8]:

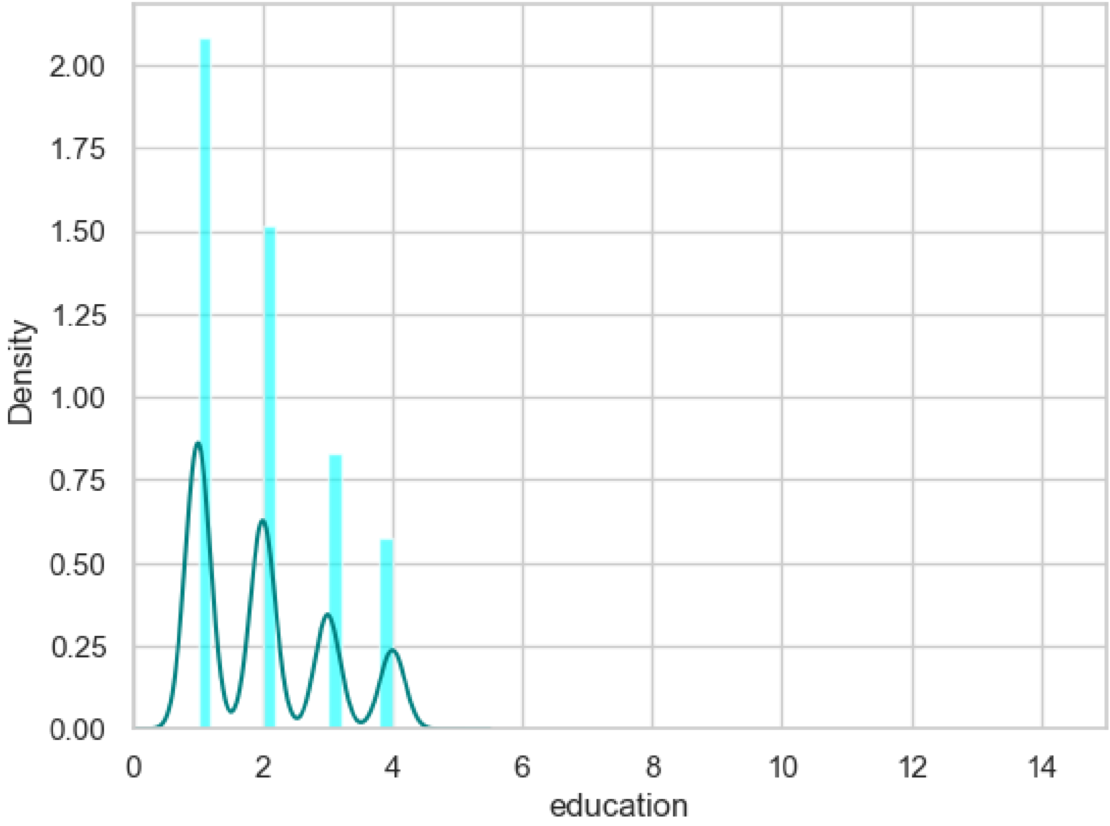
```
df.isnull().sum()
```

Out[8]:

```
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 [12]:

```
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 [13]:

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

```
1.9789499153157513
2.0
```

In [14]:

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

```
9.155261915998112
```

In [15]:

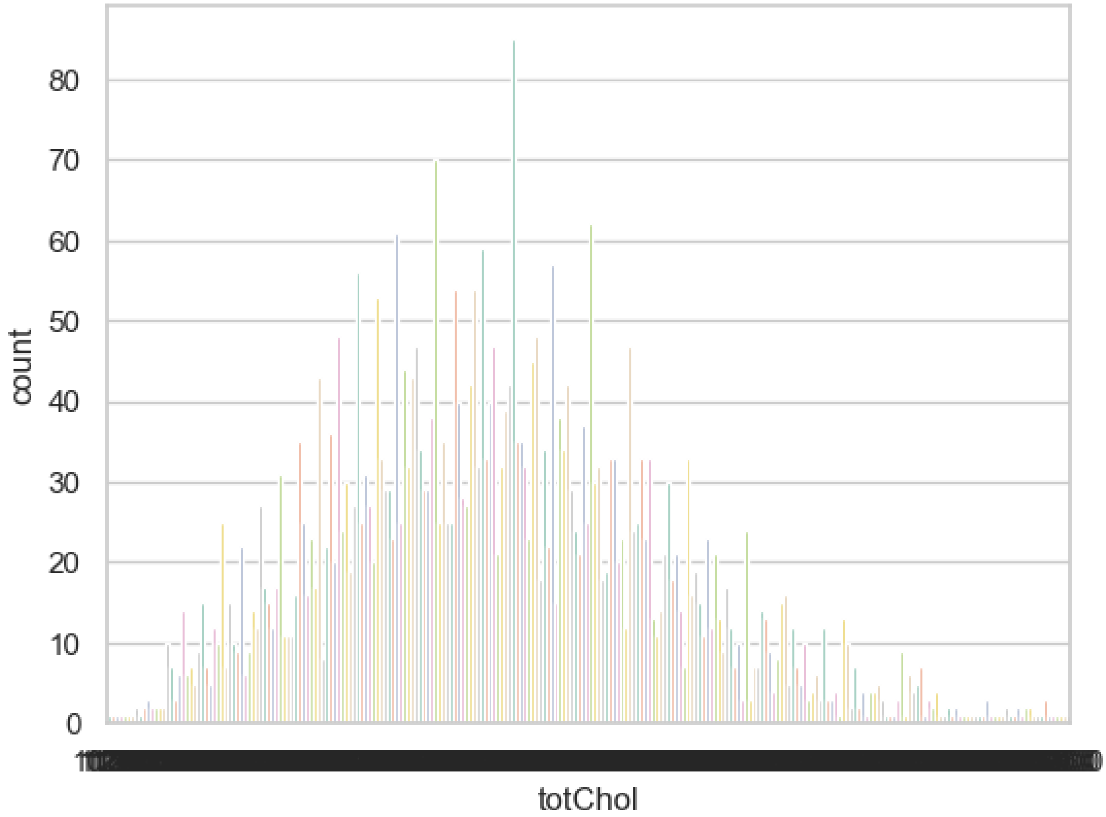
```
print(df['totChol'].isnull().sum()/df.shape[0]*100)
```

```
1.1798017932987257
```

In [16]:

```
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 [17]:

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

```
240.0
```

In [18]:

```
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 [19]:

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

Out[19]:

```
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 [20]:

```
pd.set_option('display.max_rows',4238)
pd.set_option('display.max_columns',16)
```

In [21]:

```
pd.set_option('display.width',50)
```

In [22]:

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

```
This DataFrame has 4238 Rows and 16 Columns
```

In [23]:

```
features_matrix=df.iloc[:,0:15]
```

In [24]:

```
target_vector=df.iloc[:,-2]
```

In [25]:

```
print('The Features Matrix Has %d Rows And %d Column(s)'%(features_matrix.shape))
```

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

In [28]:

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

385	1	39	2.0	0	0.0	0.0	0	0
386	0	63	2.0	0	0.0	0.0	0	1
387	0	55	3.0	0	0.0	0.0	0	1
388	0	39	2.0	0	0.0	0.0	0	0
389	0	39	2.0	1	20.0	0.0	0	0
390	1	51	1.0	1	20.0	0.0	0	0
391	0	54	1.0	0	0.0	0.0	0	0
392	0	48	3.0	0	0.0	0.0	0	0
393	1	51	1.0	1	20.0	0.0	0	0
394	0	65	2.0	0	0.0	0.0	0	1
395	0	65	2.0	0	0.0	NaN	0	1
396	1	39	3.0	0	0.0	0.0	0	0

In [29]:

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

```
9.003088619624615
0.0
```

In [30]:

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

```
1.2505899008966492
```

In [31]:

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

```
0.4483246814535158
```

In [32]:

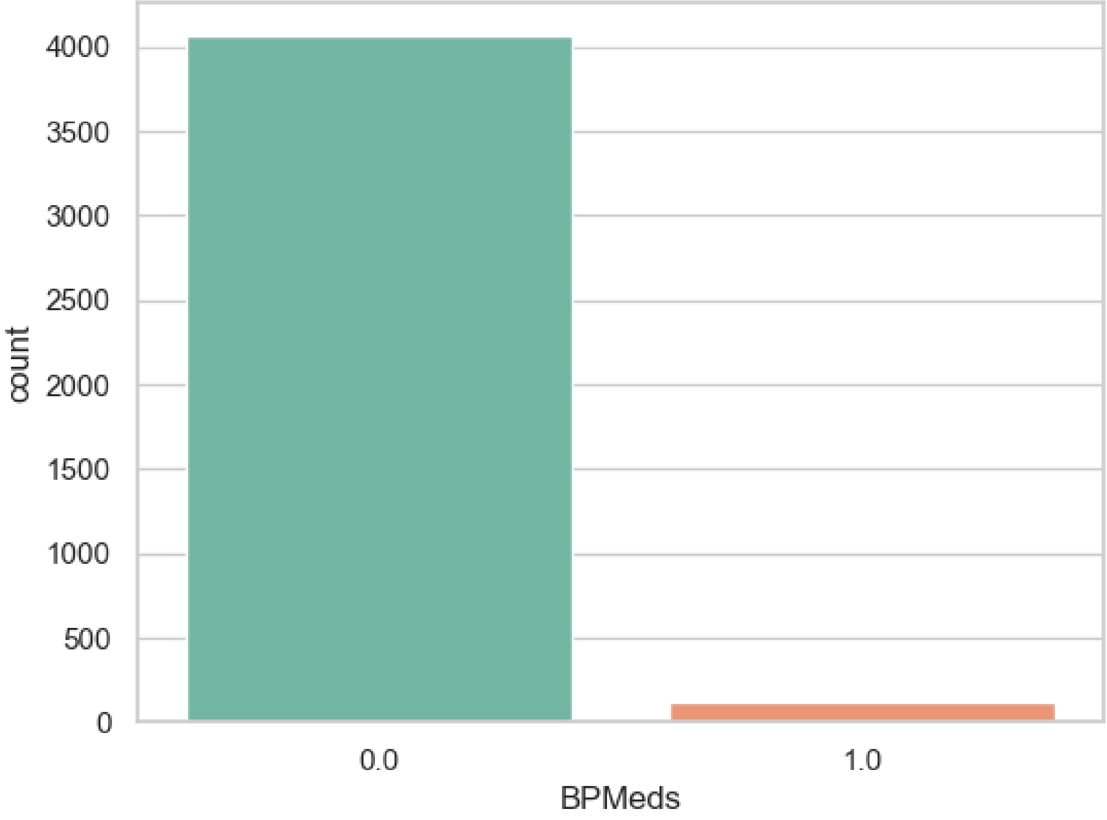
```
print((df['heartRate'].isnull().sum()/df.shape[0]*100))
```

```
0.023596035865974516
```

In [33]:

```
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
```



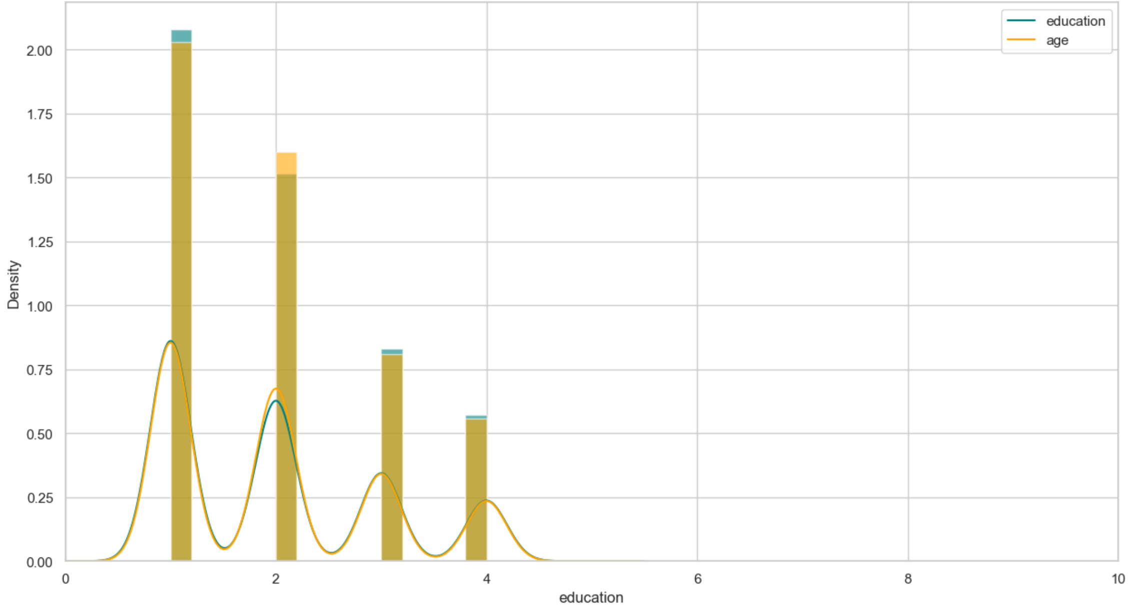
In [34]:

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

```
75.0
```

In [36]:

```
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 [37]:

```
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 [38]:

```
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[38]:

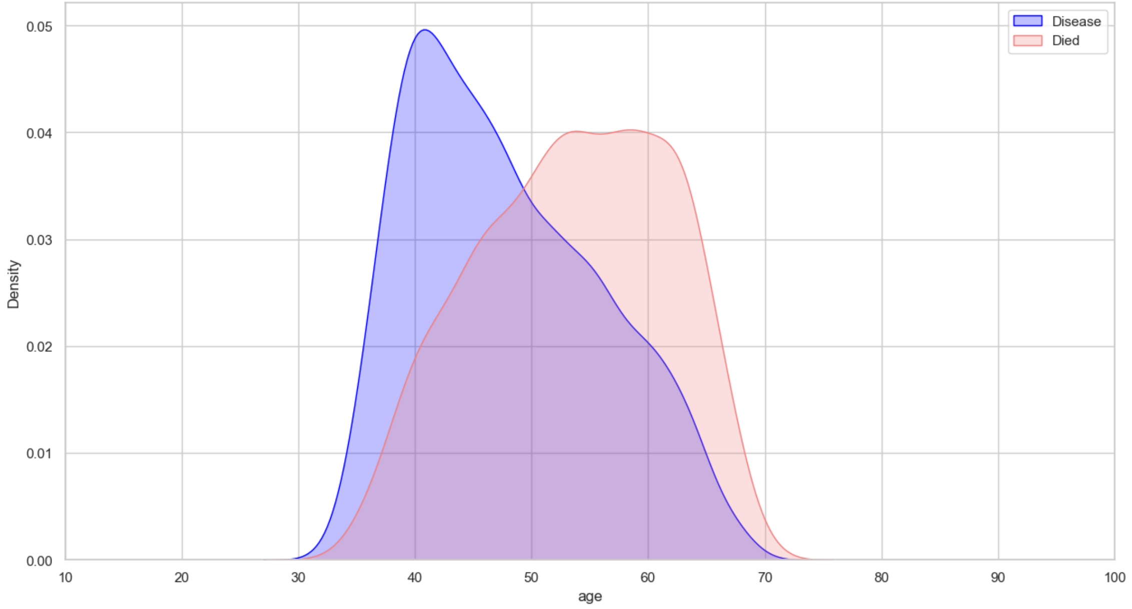
	age	education	cigsPerDay	BPMeds	diabetes	BMI	heartRate	Disease	...	sysBP_220
0	39	4.0	0.0	0.0	0	26.97	80.0	1	...	Fal
1	46	2.0	0.0	0.0	0	28.73	95.0	1	...	Fal
2	48	1.0	20.0	0.0	0	25.34	75.0	1	...	Fal
3	61	3.0	30.0	0.0	0	28.58	65.0	0	...	Fal
4	46	3.0	23.0	0.0	0	23.10	85.0	1	...	Fal

5 rows × 492 columns

EXPLORATORY DATA ANALYSIS

In [46]:

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

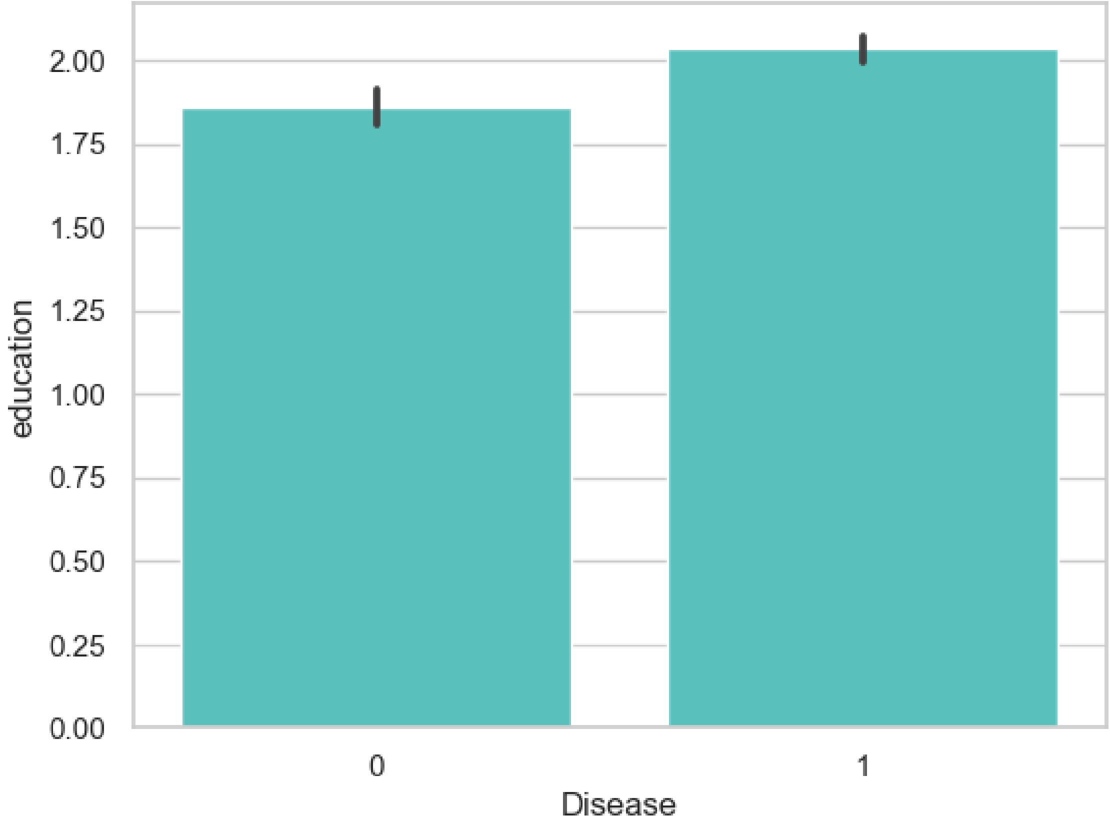


In [51]:

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

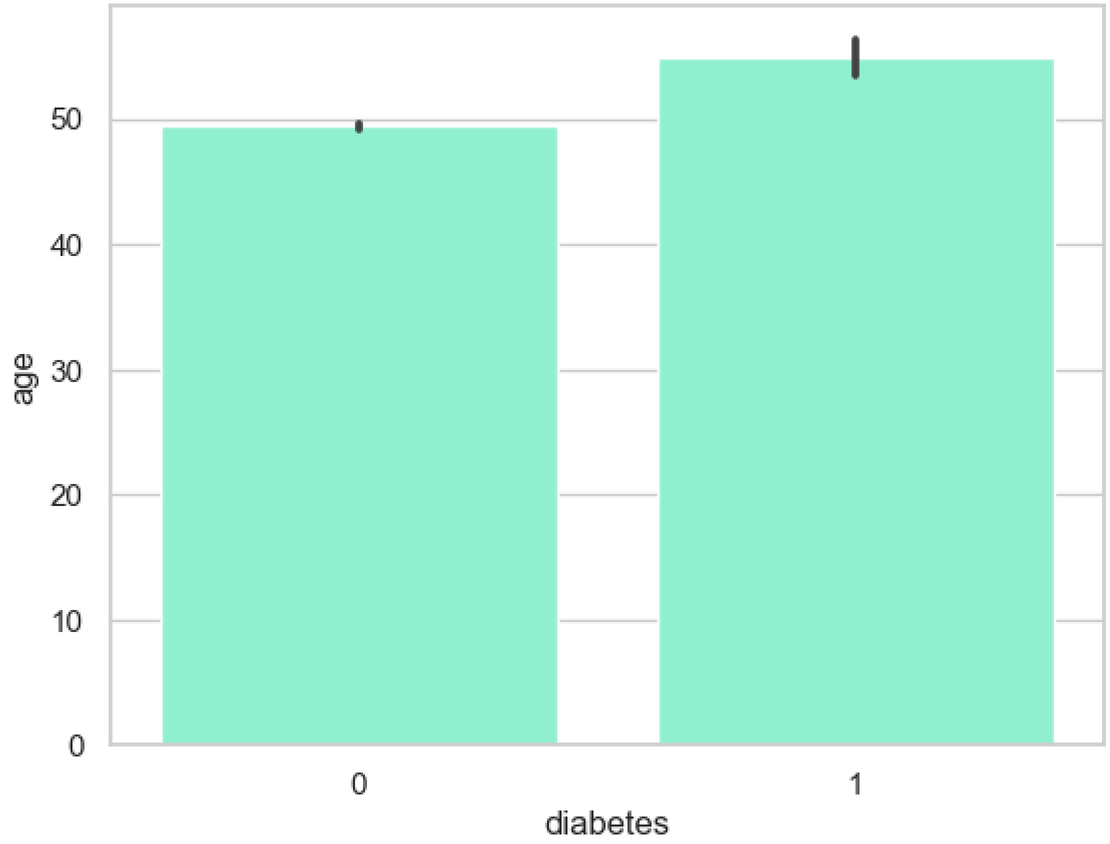
```
85      0
86      0
87      0
88      0
89      0
90      0
91      0
92      0
93      0
94      0
95      0
96      0
97      0
98      0
99      0
100     0
101     0
102     0
103     0
104     0
```

In [53]:



In [55]:

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


age
diabetes