

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

```
In [2]: df=pd.read_csv(r"C:\Users\91903\Downloads\framingham.csv")
df
```

Out[2]:

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

Out[3]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	d
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	



In [4]: `df.tail()`

Out[4]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
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 [5]: `df.shape`

Out[5]: (4238, 16)

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
```

In [7]: `df.describe()`

Out[7]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevale
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	423
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	

In [8]: `df.isnull().sum()`

Out[8]:

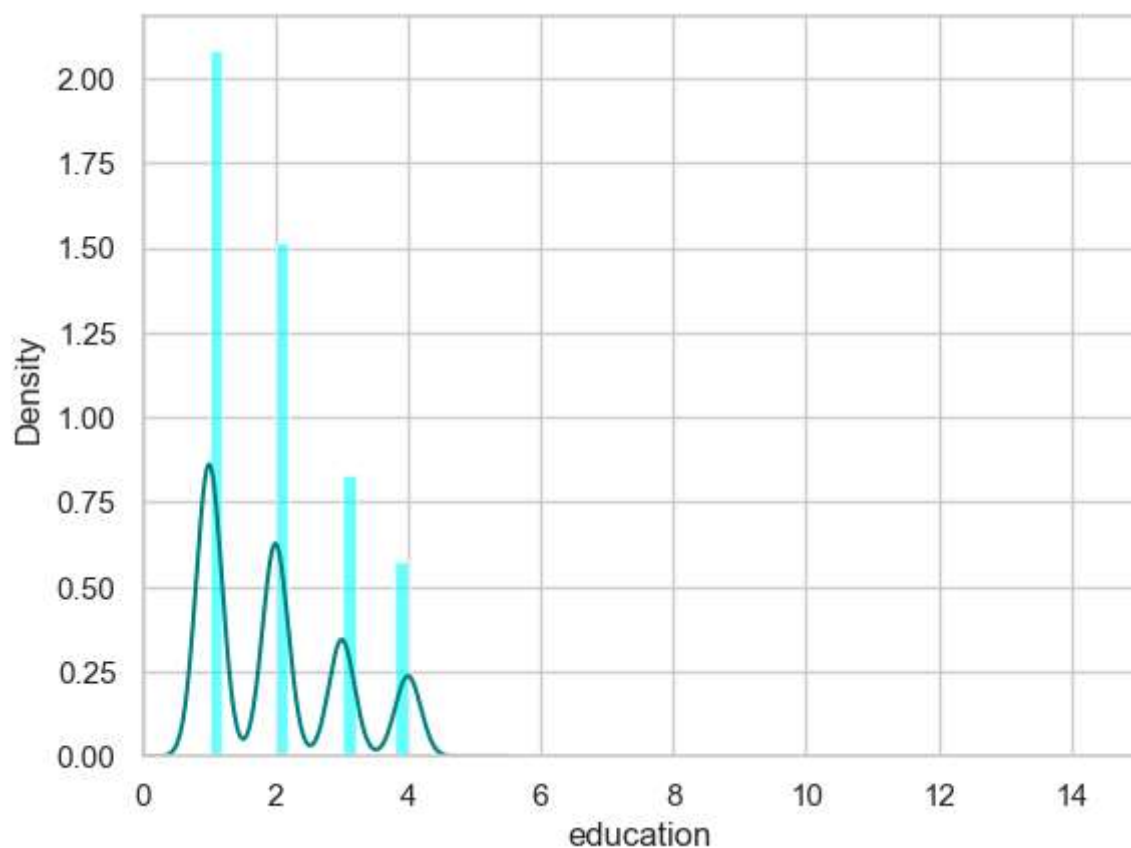
male	0
age	0
education	105
currentSmoker	0
cigsPerDay	29
BPMeds	53
prevalentStroke	0
prevalentHyp	0
diabetes	0
totChol	50
sysBP	0
diaBP	0
BMI	19
heartRate	1
glucose	388
TenYearCHD	0

dtype: int64

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

```
Out[9]: male                True
age                True
education          True
currentSmoker      True
cigsPerDay         True
BPMeds             True
prevalentStroke    True
prevalentHyp       True
diabetes           True
totChol            True
sysBP              True
diaBP              True
BMI                True
heartRate          True
glucose            True
TenYearCHD         True
dtype: bool
```

```
In [10]: ax=df["education"].hist(bins=15,density=True,stacked=True,color='cyan',alpha=0.5)
df["education"].plot(kind='density',color='teal')
ax.set(xlabel='education')
plt.xlim(-0,15)
plt.show()
```



```
In [11]: print(df["education"].mean(skipna=True))  
         print(df["education"].median(skipna=True))
```

```
1.9789499153157513  
2.0
```

```
In [12]: print((df['glucose'].isnull().sum()/df.shape[0]*100))
```

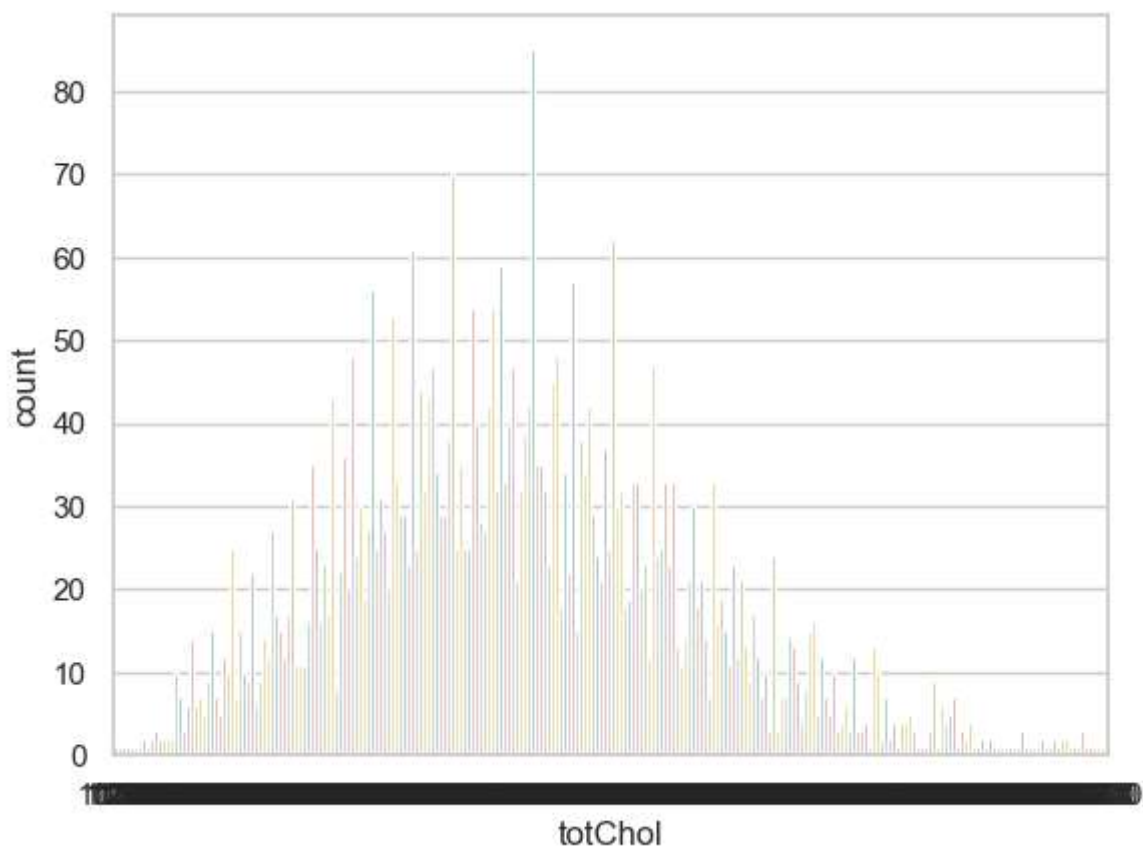
```
9.155261915998112
```

```
In [13]: print((df['totChol'].isnull().sum()/df.shape[0]*100))
```

```
1.1798017932987257
```

```
In [14]: 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 [15]: print(df['totChol'].value_counts().idxmax())
```

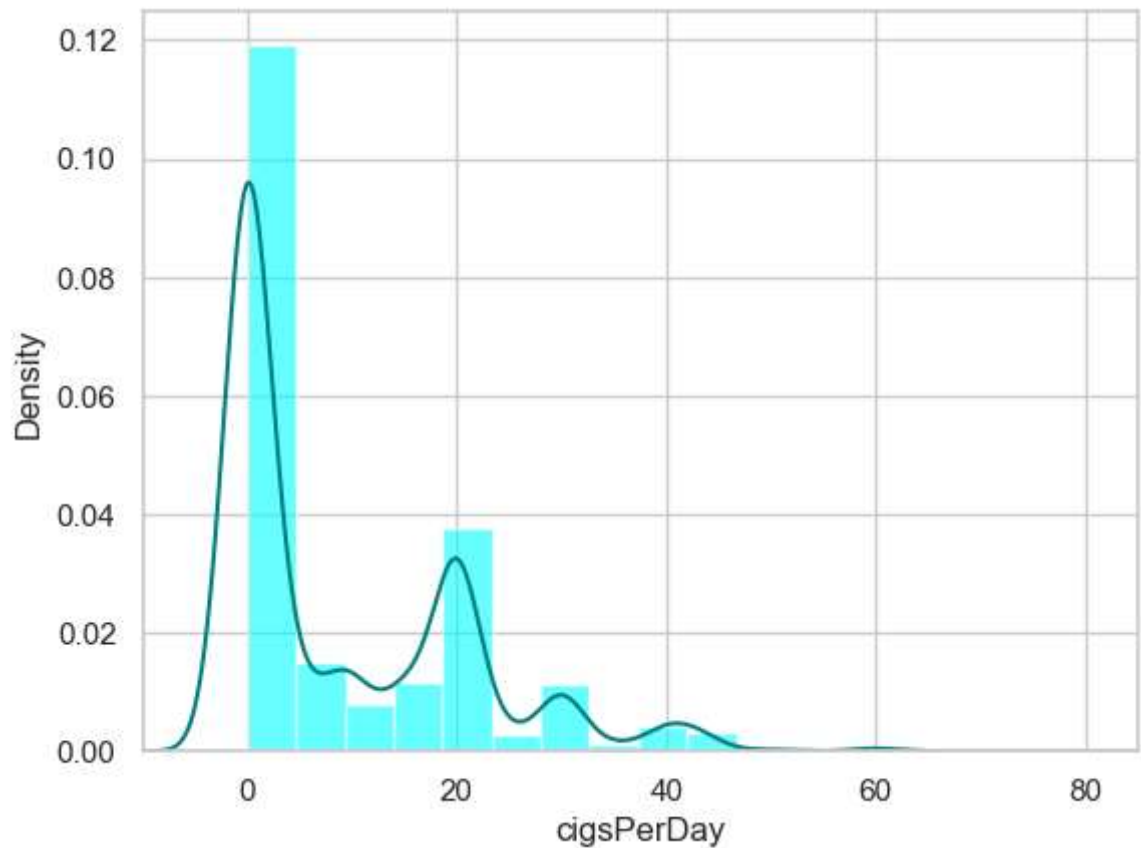
```
240.0
```

```
In [16]: 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 [17]: data.isnull().sum()
```

```
Out[17]: 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 [18]: ax=df["cigsPerDay"].hist(bins=15,density=True,stacked=True,color='cyan',alpha=
df["cigsPerDay"].plot(kind='density',color='teal')
ax.set(xlabel='cigsPerDay')
plt.xlim(-10,85)
plt.show()
```



```
In [19]: print(df["cigsPerDay"].mean(skipna=True))  
print(df["cigsPerDay"].median(skipna=True))
```

```
9.003088619624615  
0.0
```

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

```
1.2505899008966492
```

```
In [21]: print((df['BMI'].isnull().sum()/df.shape[0]*100))
```

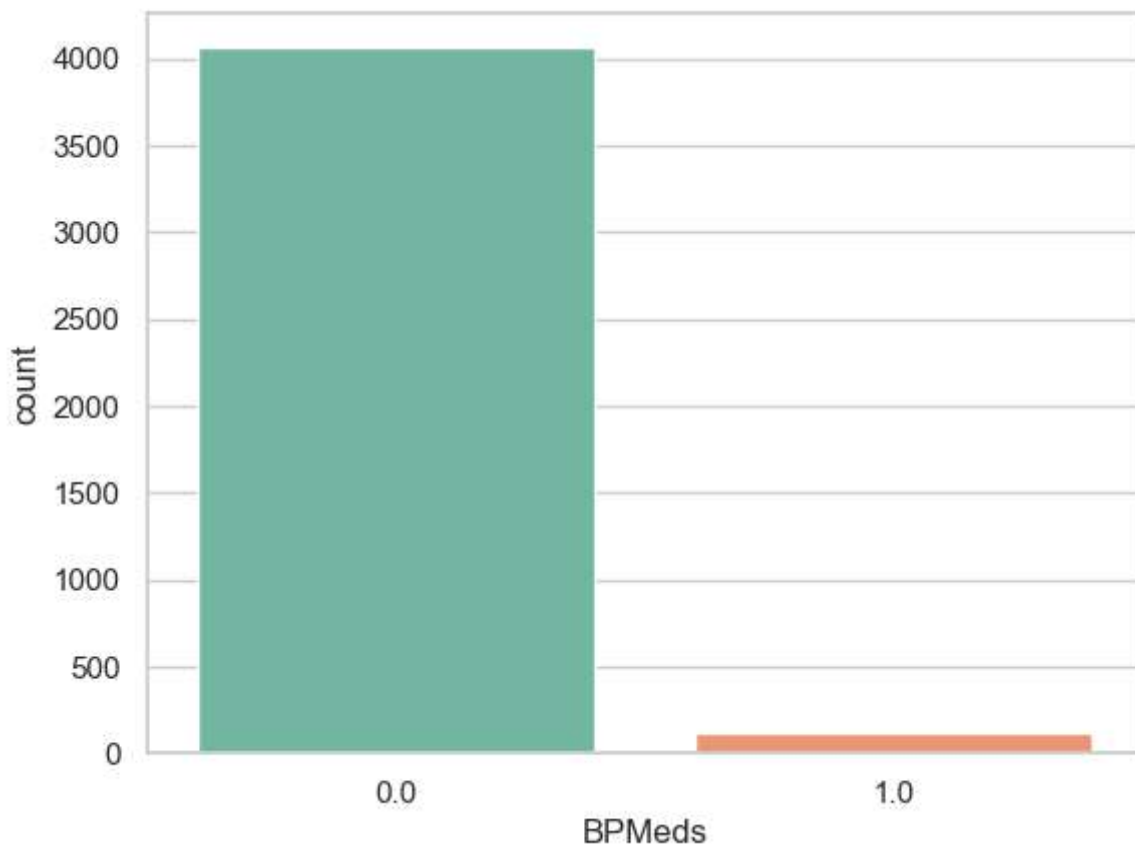
```
0.4483246814535158
```

```
In [22]: print((df['heartRate'].isnull().sum()/df.shape[0]*100))
```

```
0.023596035865974516
```

```
In [23]: 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 [24]: print(df['heartRate'].value_counts().idxmax())
```

75.0

```
In [25]: data=df.copy()
data["cigsPerDay"].fillna(df["cigsPerDay"].median(skipna=True),inplace=True)
data["BPMeds"].fillna(df["BPMeds"].median(skipna=True),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 [26]: data.isnull().sum()
```

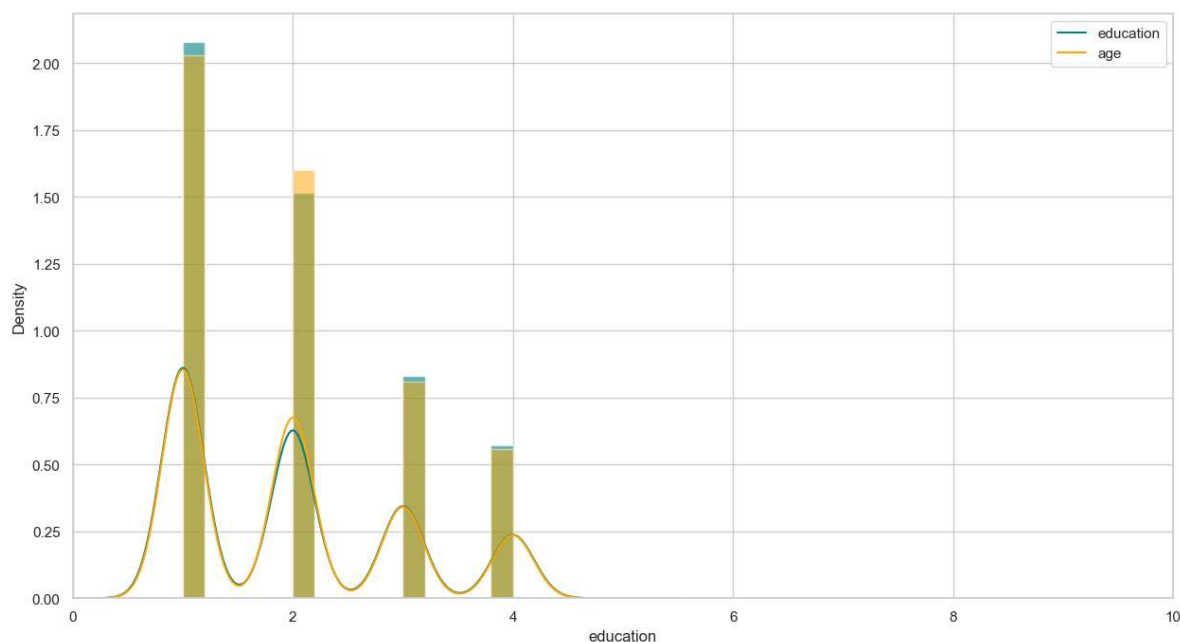
```
Out[26]: male                0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
TenYearCHD         0
dtype: int64
```

```
In [27]: data.head()
```

```
Out[27]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	d
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	

```
In [28]: plt.figure(figsize=(15,8))
ax=df["education"].hist(bins=15,density=True,stacked=True,color='teal',alpha=0.5)
df["education"].plot(kind='density',color='teal')
ax=data["education"].hist(bins=15,density=True,stacked=True,color='orange',alpha=0.5)
data["education"].plot(kind='density',color='orange')
ax.legend(["education","age"])
ax.set(xlabel='education')
plt.xlim(-0,10)
plt.show()
```



```
In [29]: 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 [30]: 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()
```

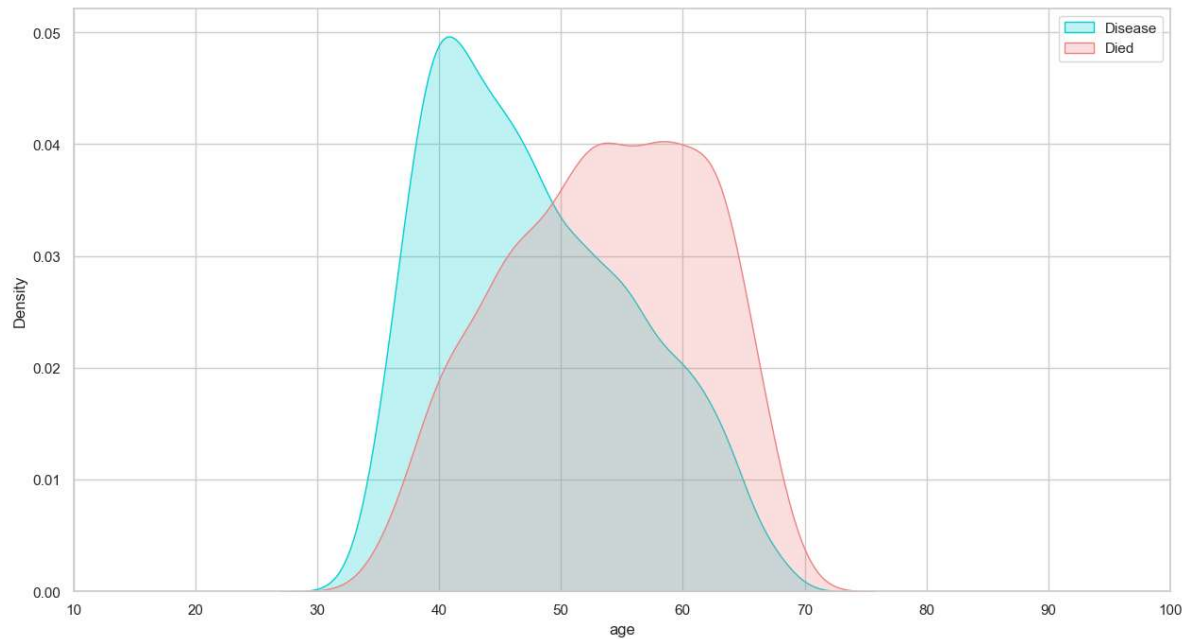
Out[30]:

	age	education	cigsPerDay	BPMeds	diabetes	Disease	currentSmoker_0	currentSmoker_1
0	39	4.0	0.0	0.0	0	1	True	False
1	46	2.0	0.0	0.0	0	1	True	False
2	48	1.0	20.0	0.0	0	1	False	True
3	61	3.0	30.0	0.0	0	0	False	True
4	46	3.0	23.0	0.0	0	1	False	True

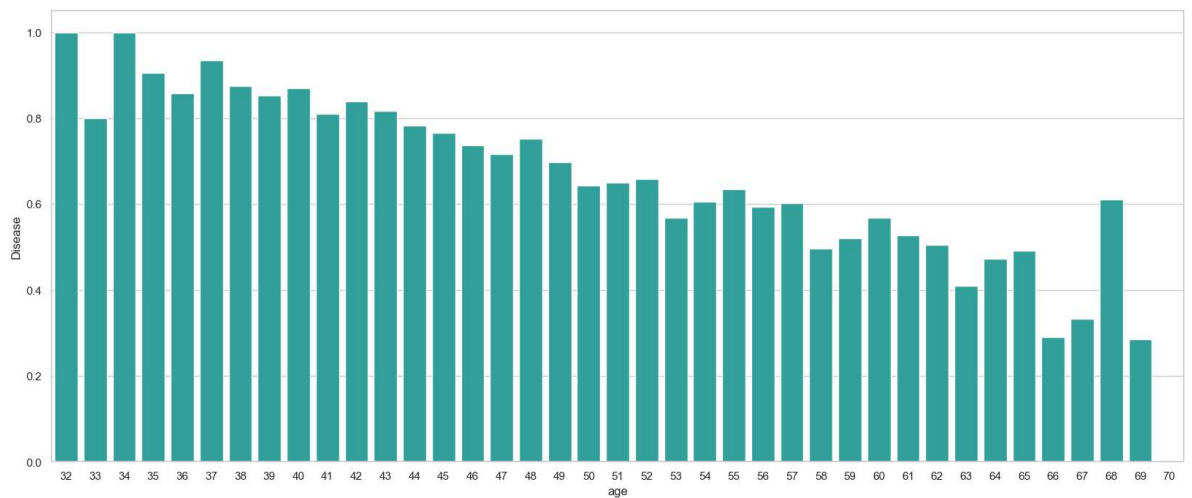
5 rows × 490 columns

Exploratory Data Analysis

```
In [31]: 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",
plt.legend(['Disease', 'Died'])
ax.set(xlabel='age')
plt.xlim(10,100)
plt.show()
```



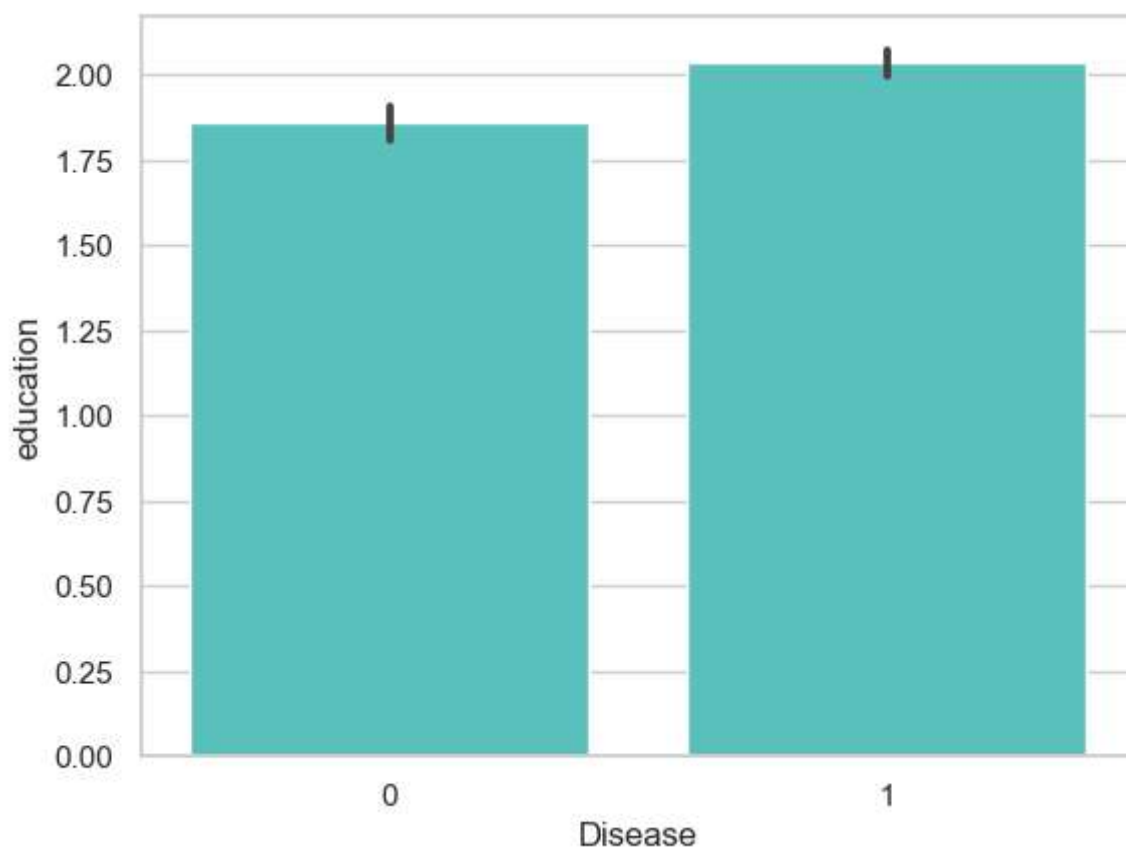
```
In [32]: plt.figure(figsize=(20,8))
avg_survival_byage = final_train[["age", "Disease"]].groupby(['age'], as_index=False)
g = sns.barplot(x='age', y='Disease', data=avg_survival_byage, color="LightSeaGreen")
plt.show()
```



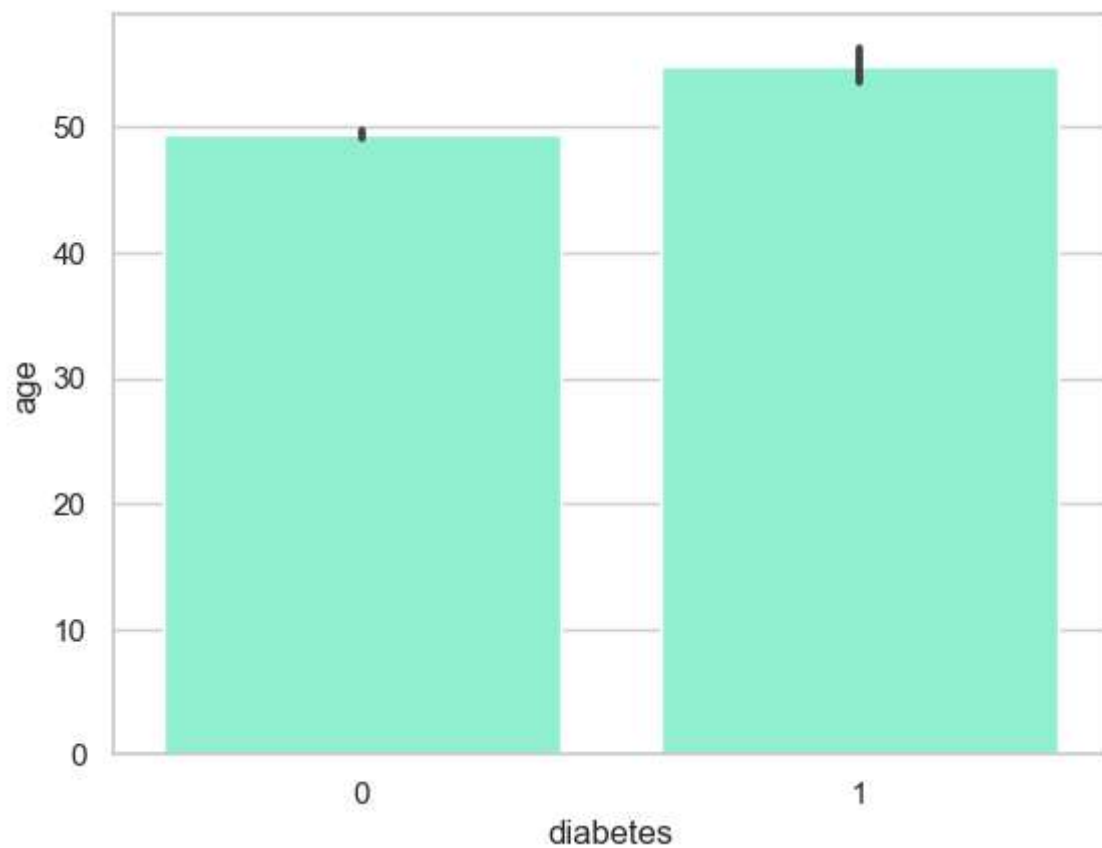
```
In [33]: final_train['IsMinor']=np.where(final_train['age']<=16, 1, 0)
print(final_train['IsMinor'])
```

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



```
In [35]: 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 []: