In [1]:
```
import numpy as np
import pandas as pd
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\HP\Downloads\archive (1).zip")
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.shape
```

Out[3]:
```
(4238, 16)
```

In [4]:
```
df.describe
```

Out[4]:
```
<bound method NDFrame.describe of       male  age  education  currentSmoker
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    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 [5]:
```
df.info()
```

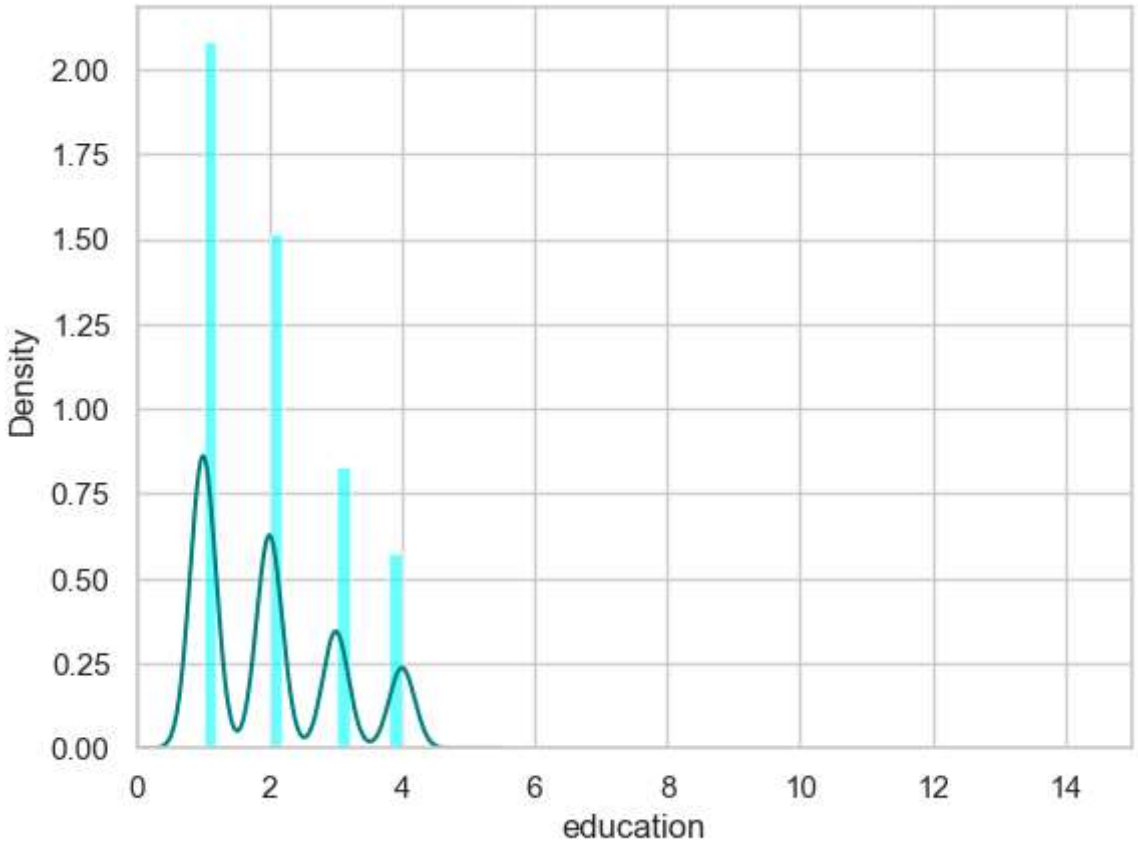
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        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 [6]:
```
df.isnull().sum()
```

Out[6]:
```
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 [7]:
```
ax=df['education'].hist(bins=15,density=True,stacked=True,color='cyan',alpha=0
df["education"].plot(kind='density',color='teal')
ax.set(xlabel="education")
plt.xlim(-0,15)
plt.show()
```



In [8]:
```
print(df['education'].mean(skipna=True))
print(df['education'].median(skipna=True))
```

```
1.9789499153157513
2.0
```

In [9]:
```
print(df['glucose'].isnull().sum()/df.shape[0]*100)
```

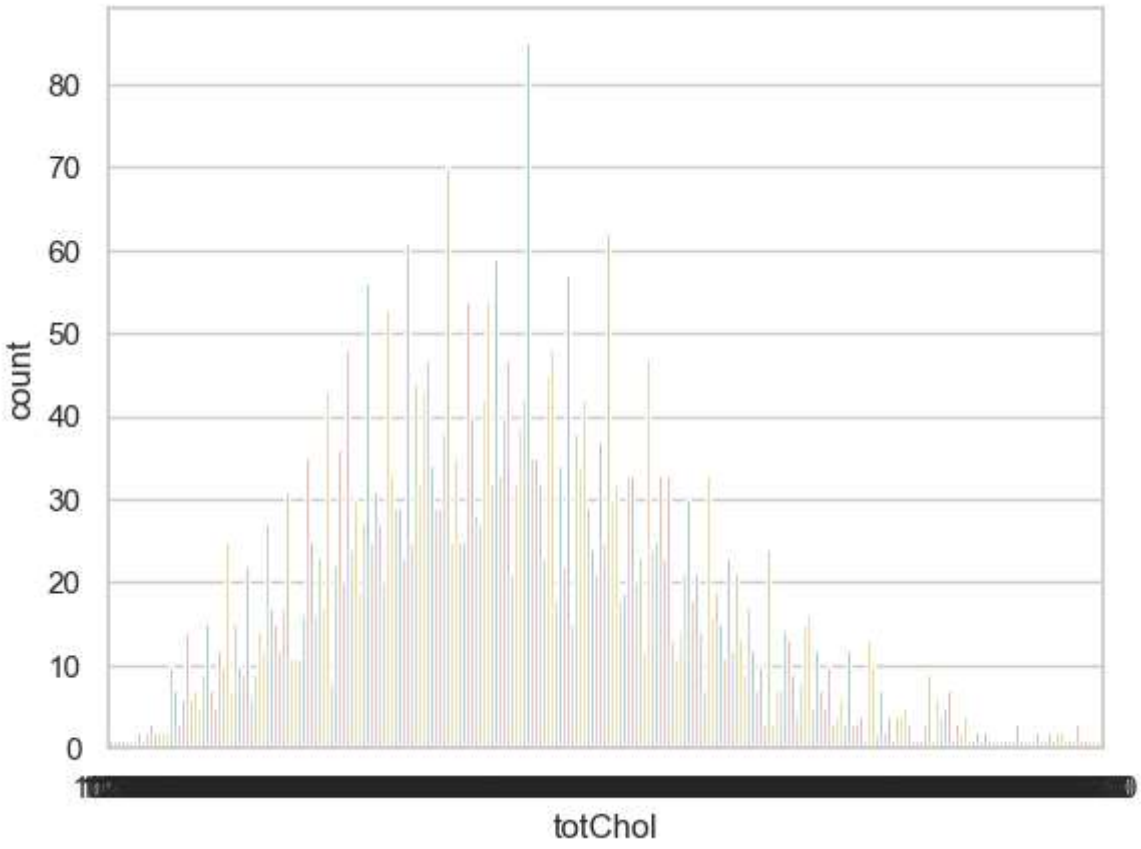
```
9.155261915998112
```

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

```
1.1798017932987257
```

In [11]:
```
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 [12]:
```
print(df['totChol'].value_counts().idxmax())
```

```
240.0
```

In [13]:
```
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 [14]:
```
data.isnull().sum()
```

Out[14]:
```
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 [15]:
```
ax=df["clgsPerDay"].hist(bins=15,density=True,stacked=True,color='cyan',alpha=
df["clgsPerDay"].plot(kind='density',color='teal')
ax.set(xlabel='clgsPerDay')
plt.xlim(-10,85)
plt.show()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core
\indexes\base.py:3652, in Index.get_loc(self, key)
   3651 try:
-> 3652     return self._engine.get_loc(casted_key)
   3653 except KeyError as err:

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\_libs
\index.pyx:147, in pandas._libs.index.IndexEngine.get_loc()

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\_libs
\index.pyx:176, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:7080, in pandas._libs.hashtable.
PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.
PyObjectHashTable.get_item()

KeyError: 'clgsPerDay'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[15], line 1
----> 1 ax=df["clgsPerDay"].hist(bins=15,density=True,stacked=True,color='cya
n',alpha=0.5)
      2 df["clgsPerDay"].plot(kind='density',color='teal')
      3 ax.set(xlabel='clgsPerDay')

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core
\frame.py:3761, in DataFrame.__getitem__(self, key)
   3759 if self.columns.nlevels > 1:
   3760     return self._getitem_multilevel(key)
-> 3761 indexer = self.columns.get_loc(key)
   3762 if is_integer(indexer):
   3763     indexer = [indexer]

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core
\indexes\base.py:3654, in Index.get_loc(self, key)
   3652     return self._engine.get_loc(casted_key)
   3653 except KeyError as err:
-> 3654     raise KeyError(key) from err
   3655 except TypeError:
   3656     # If we have a listlike key, _check_indexing_error will raise
   3657     #  InvalidIndexError. Otherwise we fall through and re-raise
   3658     #  the TypeError.
   3659     self._check_indexing_error(key)

KeyError: 'clgsPerDay'
```

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

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

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

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

In [ ]:
```
print(df['BPMeds'].value_counts())
sns.countplot(x='BPMeds',data=df,palette='set2')
plt.show()
```

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

In [19]:
```
data=df.copy()
data["clgsPerDay"].fillna(df["clgsPerDay"].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)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core
\indexes\base.py:3652, in Index.get_loc(self, key)
   3651 try:
-> 3652     return self._engine.get_loc(casted_key)
   3653 except KeyError as err:

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\_libs
\index.pyx:147, in pandas._libs.index.IndexEngine.get_loc()

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\_libs
\index.pyx:176, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:7080, in pandas._libs.hashtable.
PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.
PyObjectHashTable.get_item()

KeyError: 'clgsPerDay'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[19], line 2
      1 data=df.copy()
----> 2 data["clgsPerDay"].fillna(df["clgsPerDay"].median(skipna=True),inplac
e=True)
      3 data["BPMeds"].fillna(df["BPMeds"].median(skipna=True),inplace=True)
      4 data['education'].fillna(df["education"].median(skipna=True),inplace=
True)

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core
\frame.py:3761, in DataFrame.__getitem__(self, key)
   3759 if self.columns.nlevels > 1:
   3760     return self._getitem_multilevel(key)
-> 3761 indexer = self.columns.get_loc(key)
   3762 if is_integer(indexer):
   3763     indexer = [indexer]

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core
\indexes\base.py:3654, in Index.get_loc(self, key)
   3652     return self._engine.get_loc(casted_key)
   3653 except KeyError as err:
-> 3654     raise KeyError(key) from err
   3655 except TypeError:
   3656     # If we have a listlike key, _check_indexing_error will raise
   3657     #  InvalidIndexError. Otherwise we fall through and re-raise
   3658     #  the TypeError.
   3659     self._check_indexing_error(key)

KeyError: 'clgsPerDay'
```

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

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

In [ ]:

In [ ]:

In [ ]: