

heart disease prediction project

it is very important and rare case for medical department.. generally it won't possible in python but machine learning code gives us best preference...

.....

```
In [9]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [11]: df = pd.read_excel('C:\\Users\\ACER\\Downloads\\heart.xls')
print(df)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	63	1	3	145	233	1	0	150	0	2.3	
1	37	1	2	130	250	0	1	187	0	3.5	
2	41	0	1	130	204	0	0	172	0	1.4	
3	56	1	1	120	236	0	1	178	0	0.8	
4	57	0	0	120	354	0	1	163	1	0.6	
..	
298	57	0	0	140	241	0	1	123	1	0.2	
299	45	1	3	110	264	0	1	132	0	1.2	
300	68	1	0	144	193	1	1	141	0	3.4	
301	57	1	0	130	131	0	1	115	1	1.2	
302	57	0	1	130	236	0	0	174	0	0.0	

	slope	ca	thal	target
0	0	0	1	1
1	0	0	2	1
2	2	0	2	1
3	2	0	2	1
4	2	0	2	1
..
298	1	0	3	0
299	1	0	3	0
300	1	2	3	0
301	1	1	3	0
302	1	1	2	0

[303 rows x 14 columns]

display elements

```
In [12]: df.head()
```

```
Out[12]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

check last elements

```
In [13]: df.tail()
```

```
Out[13]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0



shape of all elements

```
In [15]: df.shape
```

```
Out[15]: (303, 14)
```

get overall statistics

```
In [16]: df.describe()
```

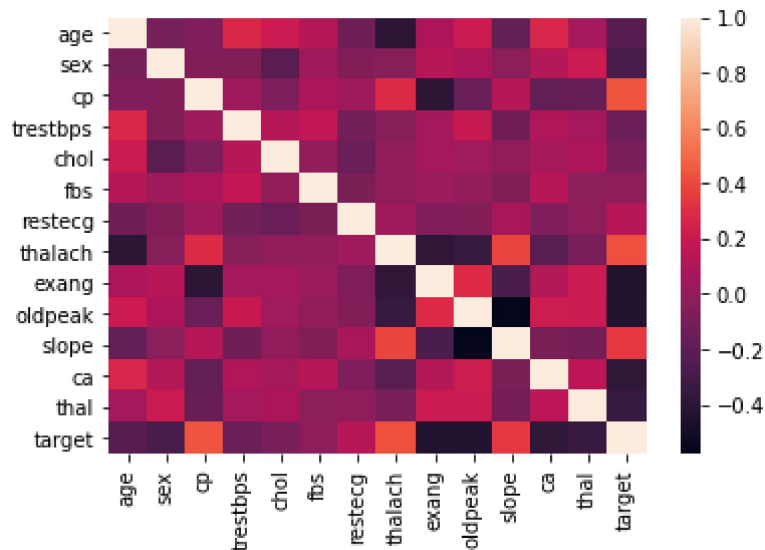
```
Out[16]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	tl
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.6
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.9
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.5
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.0
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.0
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.0

draw correlation matrix

```
In [18]: sns.heatmap(df.corr())
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1242f70>
```

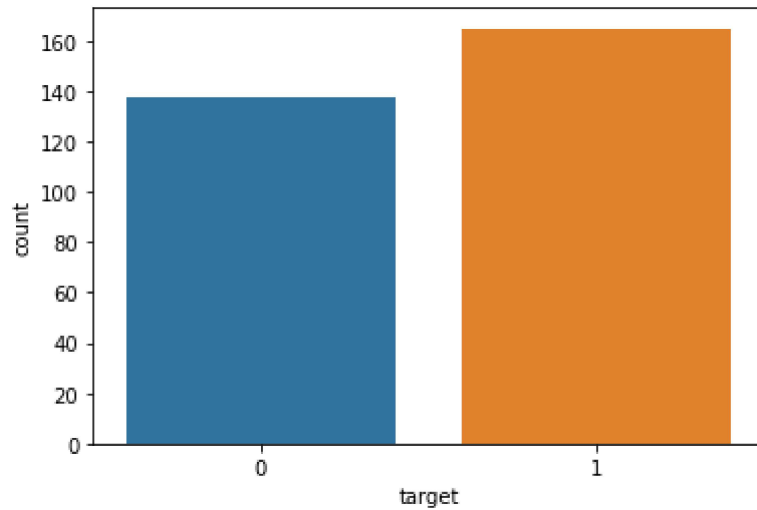


now we analysis it

1. how many people are suffering / not from heart disease

```
In [19]: sns.countplot(df['target'])
```

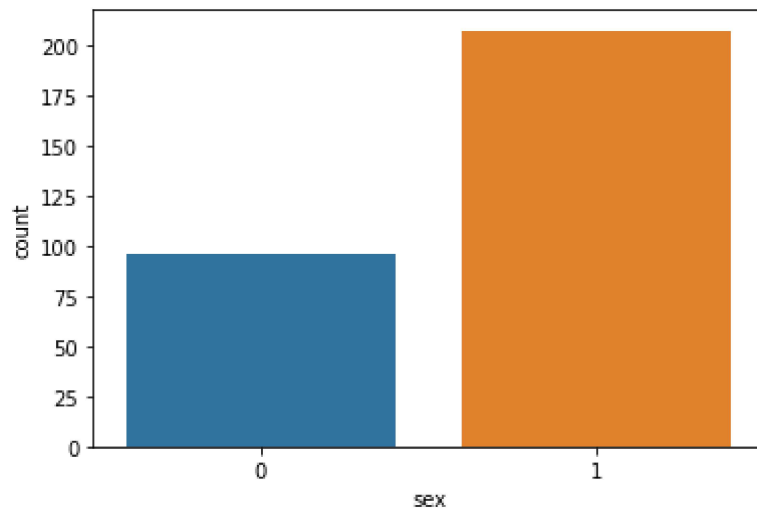
```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1395070>
```



2. find gender distribution

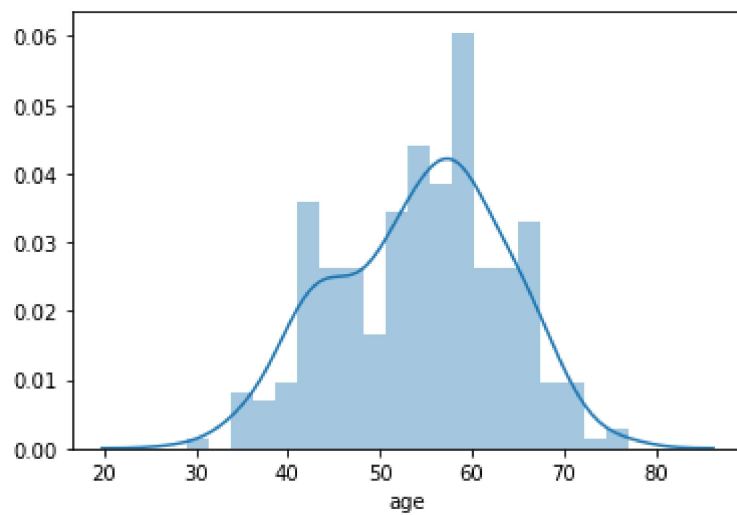
```
In [28]: sns.countplot(df['sex'])
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x636bf10>
```



3.check age distribution

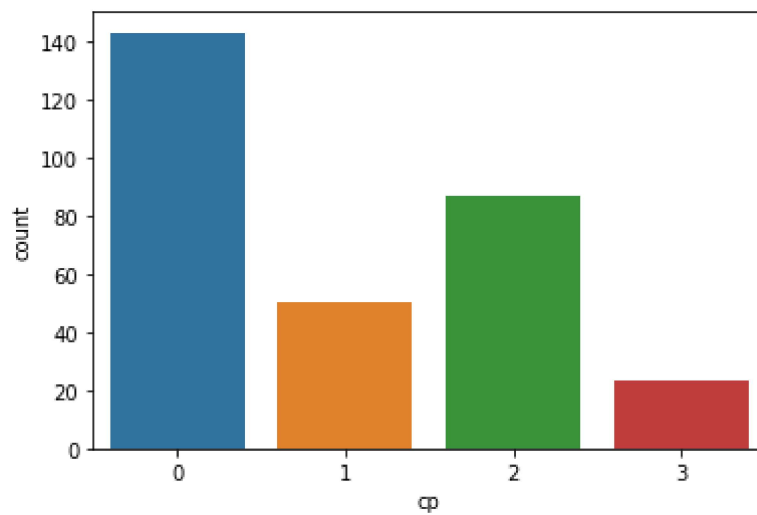
```
In [29]: sns.distplot(df['age'],bins=20)  
plt.show()
```



4.check type of chest pain

```
In [30]: sns.countplot(df['cp'])
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x9db89b8>
```



->value 0:typical angina

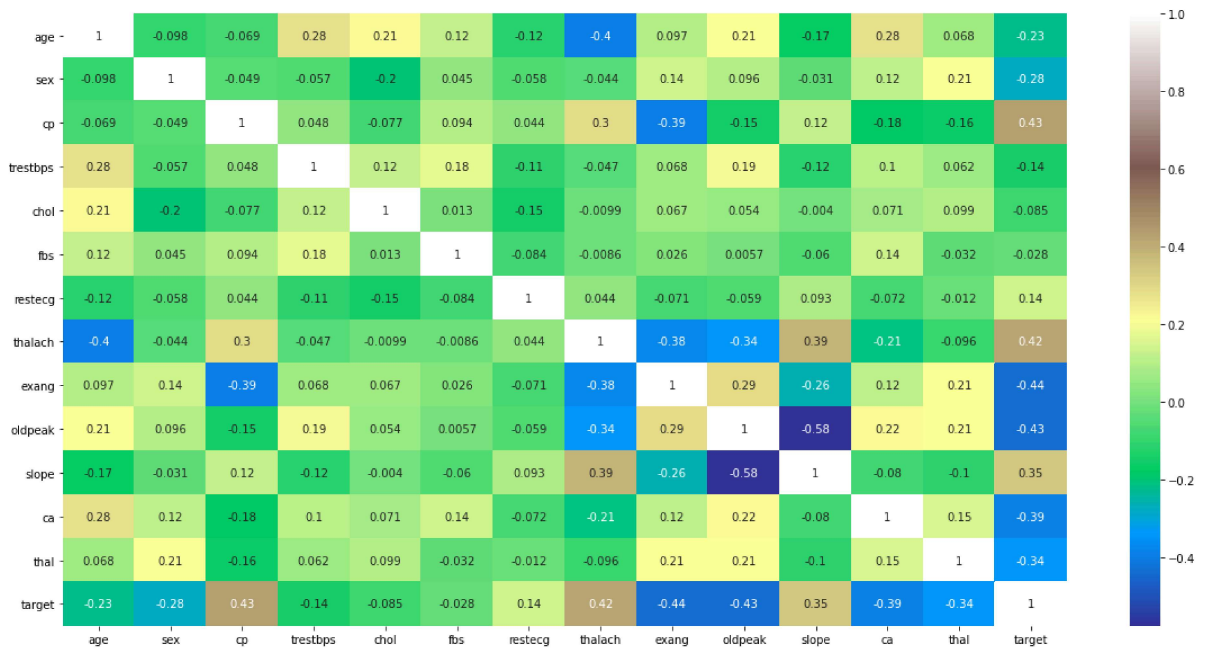
->value 1:atypical angina

->value 2:non-anginal pain

->value 3:asymptomatic

```
In [34]: plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot=True,cmap='terrain')
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x134ad30>
```



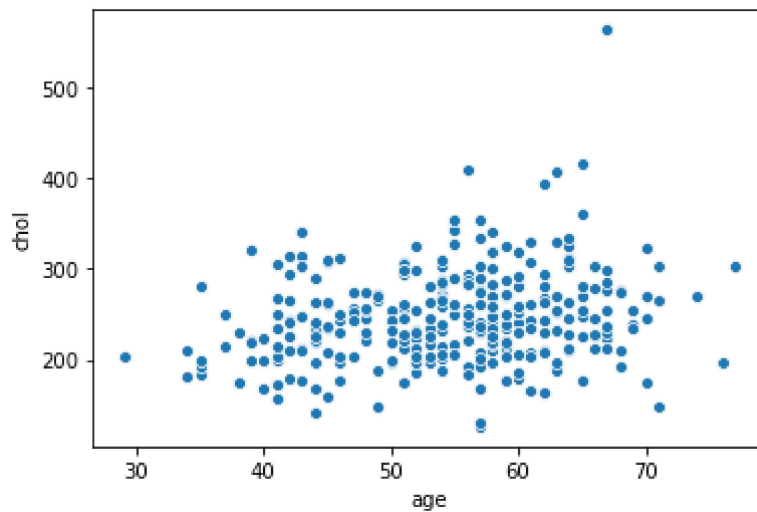
derive with calculation

```
In [46]: df.isnull().sum()
```

```
Out[46]: age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

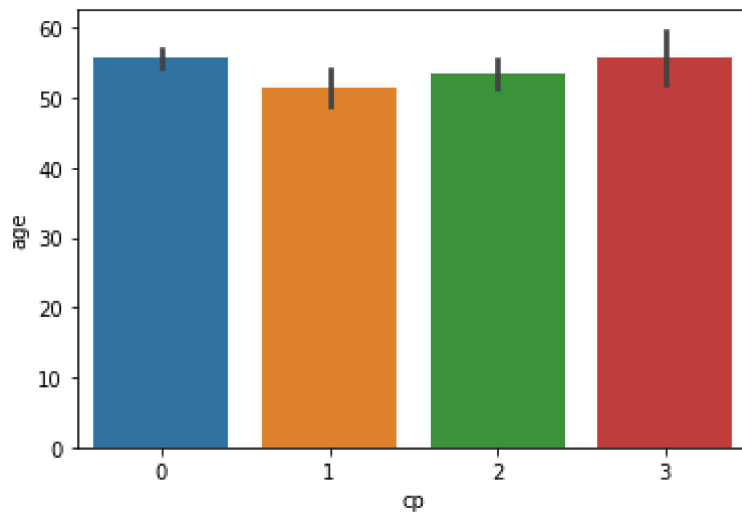
```
In [47]: sns.scatterplot(x='age',y='chol',data=df)
```

```
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2a84c0>
```



```
In [48]: sns.barplot(x='cp',y='age',data=df)
```

```
Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x18b24958>
```



train test split

```
In [50]: lr.score(xtest,ytest)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-50-2d288b51dcf9> in <module>
----> 1 lr.score(xtest,ytest)

NameError: name 'lr' is not defined
```

```
In [51]: print(df.info())
```

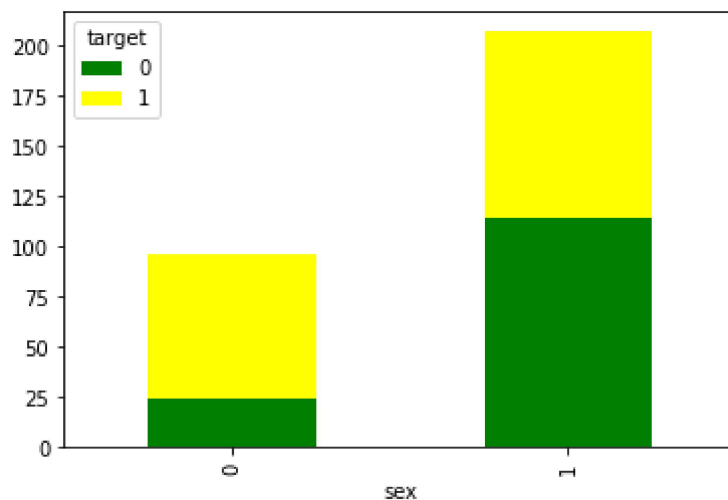
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   age         303 non-null    int64
 1   sex         303 non-null    int64
 2   cp          303 non-null    int64
 3   trestbps    303 non-null    int64
 4   chol        303 non-null    int64
 5   fbs         303 non-null    int64
 6   restecg     303 non-null    int64
 7   thalach     303 non-null    int64
 8   exang       303 non-null    int64
 9   oldpeak     303 non-null    float64
10   slope       303 non-null    int64
11   ca          303 non-null    int64
12   thal        303 non-null    int64
13   target      303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
None
```

```
In [52]: gen = pd.crosstab(df['sex'],df['target'])
print(gen)
```

```
target    0    1
sex
0         24   72
1        114   93
```

```
In [53]: gen.plot(kind='bar',stacked=True,color=['green','yellow'],grid=False)
```

```
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x18e43928>
```



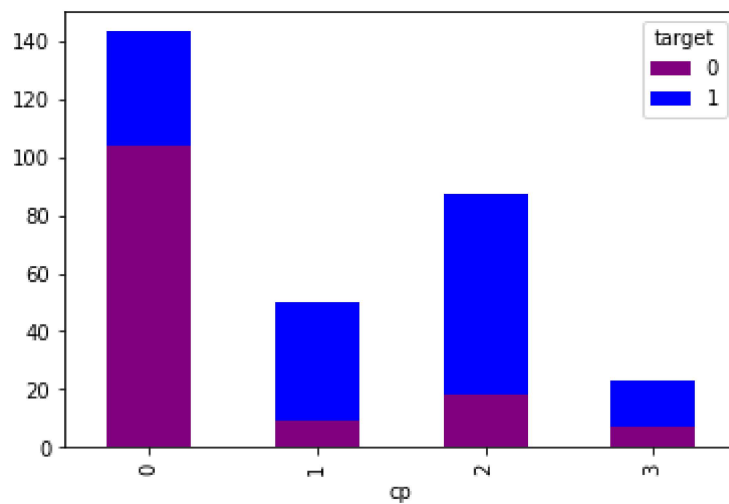

```
In [56]: chest_pain = pd.crosstab(df['cp'],df['target'])
chest_pain
```

```
Out[56]:
```

	target	0	1
cp			
0	104	39	
1	9	41	
2	18	69	
3	7	16	

```
In [58]: chest_pain.plot(kind='bar',stacked=True,color=['purple','blue'],grid=False)
```

```
Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x15681988>
```

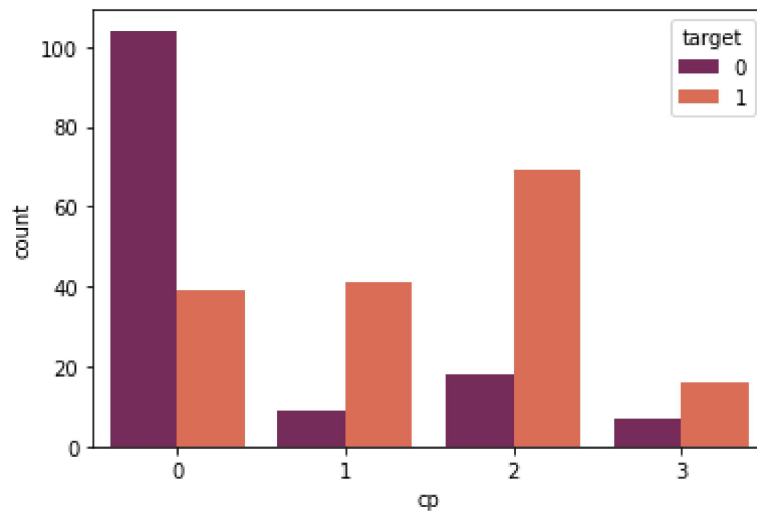


```
In [61]: df['cp'].value_counts()
```

```
Out[61]: 0    143
         2     87
         1     50
         3     23
         Name: cp, dtype: int64
```

```
In [65]: sns.countplot( x='cp',hue='target',data=df,palette='rocket')
```

```
Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0x181c4fb8>
```

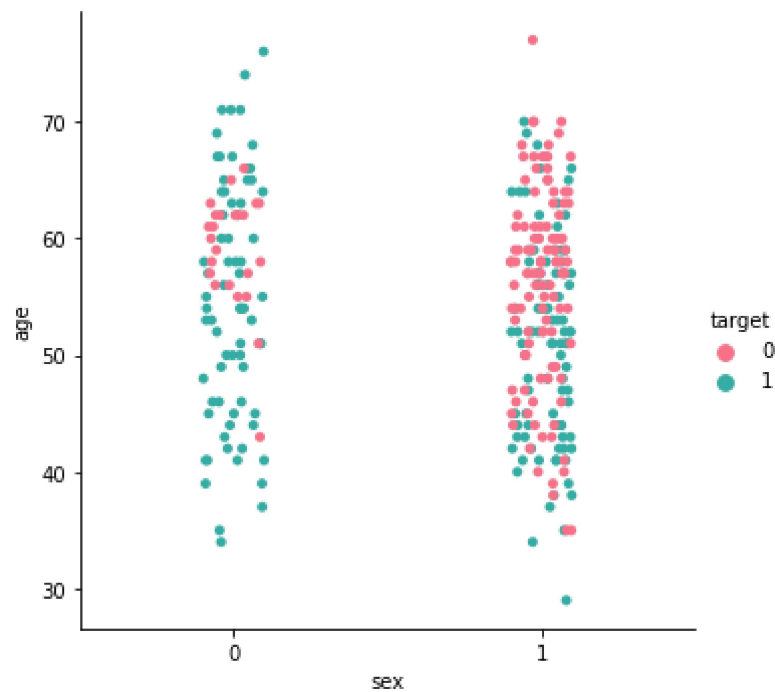


```
In [66]: df['sex'].value_counts()
```

```
Out[66]: 1    207  
         0     96  
         Name: sex, dtype: int64
```

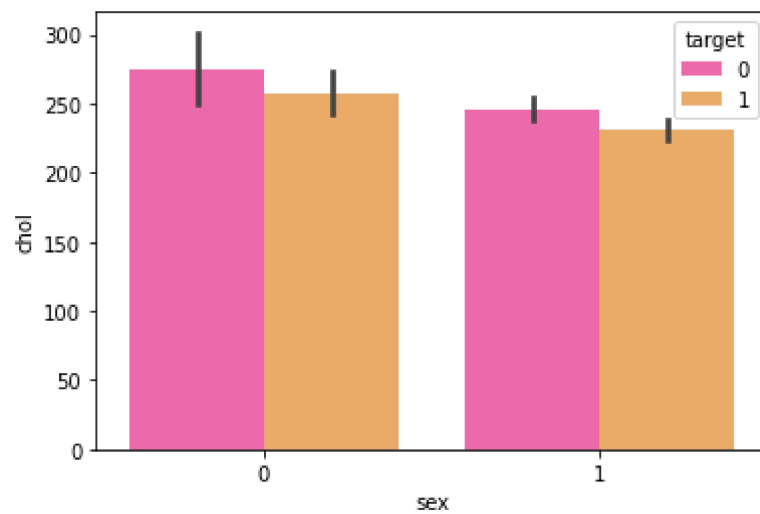
```
In [67]: sns.catplot(data=df,x='sex',y='age',hue='target',palette='husl')
```

```
Out[67]: <seaborn.axisgrid.FacetGrid at 0x1562ed48>
```



```
In [68]: sns.barplot(data=df,x='sex',y='chol',hue='target',palette='spring')
```

```
Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x1913a730>
```



Done by:-

K.K.Sreevalli