

```
In [1]: import pandas as pd
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
In [2]: data=pd.read_csv("C:\\\\Users\\\\harsh\\\\Downloads\\\\heart.csv")  
data.head()
```

```
Out[2]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

age sex chest pain type (4 values) resting blood pressure serum cholestorol in mg/dl fasting blood sugar > 120 mg/dl resting electrocardiographic results (values 0,1,2) maximum heart rate achieved exercise induced angina oldpeak = ST depression induced by exercise relative to rest the slope of the peak exercise ST segment number of major vessels (0-3) colored by flourosopy thal: 0 = normal; 1 = fixed defect; 2 = reversable defect The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

```
In [3]: data.shape
```

```
Out[3]: (1025, 14)
```

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count Dtype  
--- 
 0   age        1025 non-null   int64  
 1   sex         1025 non-null   int64  
 2   cp          1025 non-null   int64  
 3   trestbps   1025 non-null   int64  
 4   chol        1025 non-null   int64  
 5   fbs         1025 non-null   int64  
 6   restecg    1025 non-null   int64  
 7   thalach     1025 non-null   int64  
 8   exang       1025 non-null   int64  
 9   oldpeak     1025 non-null   float64 
 10  slope       1025 non-null   int64  
 11  ca          1025 non-null   int64  
 12  thal        1025 non-null   int64  
 13  target      1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
In [5]: data.isnull().sum()
```

```
Out[5]: 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 [6]: data.duplicated().sum()
```

```
Out[6]: 723
```

```
In [7]: data.drop_duplicates(inplace=True)
```

```
In [8]: data
```

```
Out[8]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...
723	68	0	2	120	211	0	0	115	0	1.5	1	0	2	1
733	44	0	2	108	141	0	1	175	0	0.6	1	0	2	1
739	52	1	0	128	255	0	1	161	1	0.0	2	1	3	0
843	59	1	3	160	273	0	0	125	0	0.0	2	0	2	0
878	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

302 rows × 14 columns

```
In [9]: data.describe()
```

Out[9]:

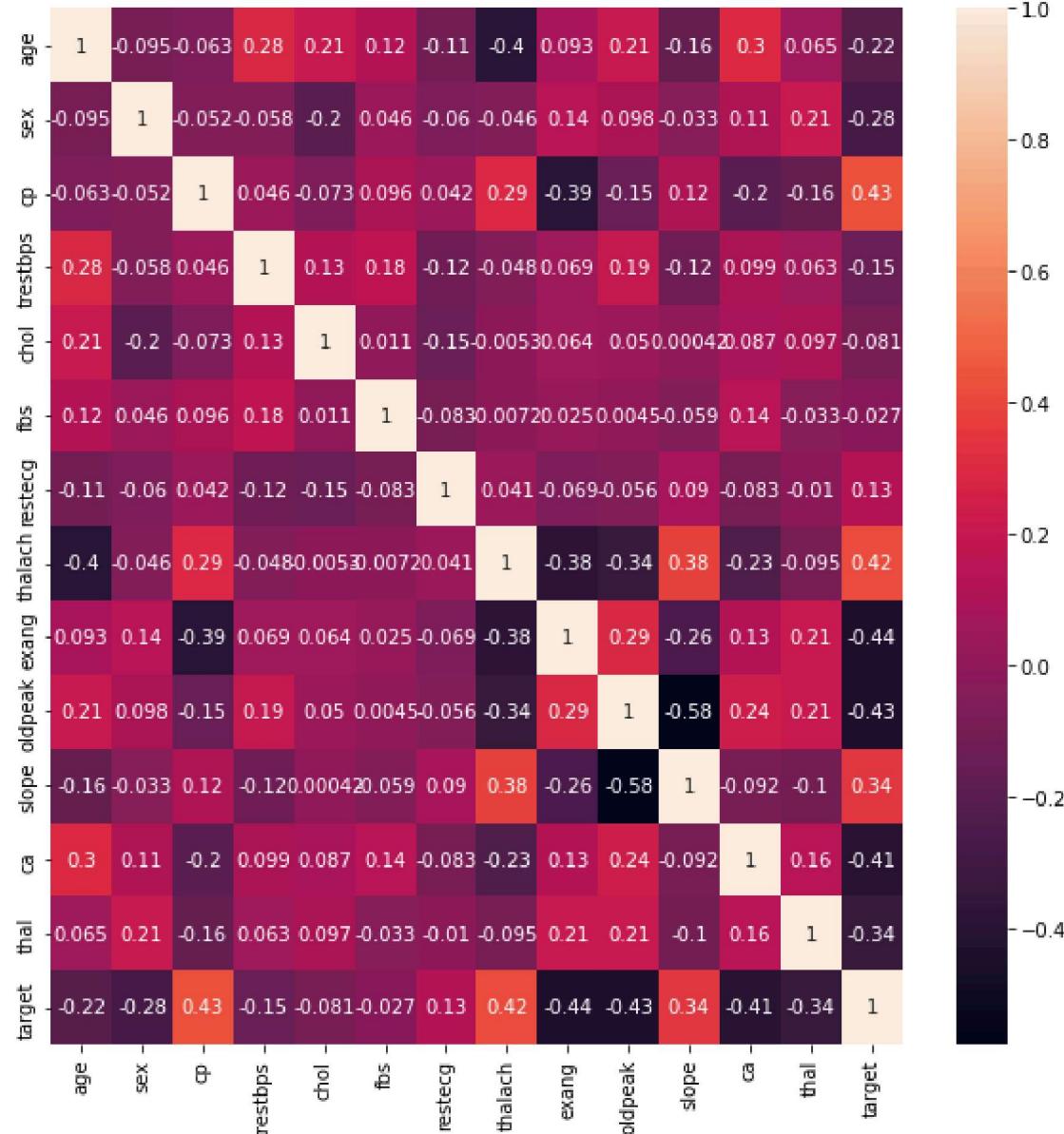
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	31
count	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000
mean	54.42053	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.569536	0.327815	1.043046	1.397351	0.718543	31
std	9.04797	0.466426	1.032044	17.563394	51.753489	0.356686	0.526027	22.903527	0.470196	1.161452	0.616274	1.006748	31
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	31
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.250000	0.000000	0.000000	1.000000	0.000000	31
50%	55.500000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.500000	0.000000	0.800000	1.000000	0.000000	31
75%	61.000000	1.000000	2.000000	140.000000	274.750000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	31
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	31

In [10]: `data.corr()`

Out[10]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	1
age	1.000000	-0.094962	-0.063107	0.283121	0.207216	0.119492	-0.111590	-0.395235	0.093216	0.206040	-0.164124	0.302261	0.065317	-0.2
sex	-0.094962	1.000000	-0.051740	-0.057647	-0.195571	0.046022	-0.060351	-0.046439	0.143460	0.098322	-0.032990	0.113060	0.211452	-0.2
cp	-0.063107	-0.051740	1.000000	0.046486	-0.072682	0.096018	0.041561	0.293367	-0.392937	-0.146692	0.116854	-0.195356	-0.160370	0.4
trestbps	0.283121	-0.057647	0.046486	1.000000	0.125256	0.178125	-0.115367	-0.048023	0.068526	0.194600	-0.122873	0.099248	0.062870	-0.1
chol	0.207216	-0.195571	-0.072682	0.125256	1.000000	0.011428	-0.147602	-0.005308	0.064099	0.050086	0.000417	0.086878	0.096810	-0.0
fbs	0.119492	0.046022	0.096018	0.178125	0.011428	1.000000	-0.083081	-0.007169	0.024729	0.004514	-0.058654	0.144935	-0.032752	-0.0
restecg	-0.111590	-0.060351	0.041561	-0.115367	-0.147602	-0.083081	1.000000	0.041210	-0.068807	-0.056251	0.090402	-0.083112	-0.010473	0.1
thalach	-0.395235	-0.046439	0.293367	-0.048023	-0.005308	-0.007169	0.041210	1.000000	-0.377411	-0.342201	0.384754	-0.228311	-0.094910	0.4
exang	0.093216	0.143460	-0.392937	0.068526	0.064099	0.024729	-0.068807	-0.377411	1.000000	0.286766	-0.256106	0.125377	0.205826	-0.4
oldpeak	0.206040	0.098322	-0.146692	0.194600	0.050086	0.004514	-0.056251	-0.342201	0.286766	1.000000	-0.576314	0.236560	0.209090	-0.4
slope	-0.164124	-0.032990	0.116854	-0.122873	0.000417	-0.058654	0.090402	0.384754	-0.256106	-0.576314	1.000000	-0.092236	-0.103314	0.3
ca	0.302261	0.113060	-0.195356	0.099248	0.086878	0.144935	-0.083112	-0.228311	0.125377	0.236560	-0.092236	1.000000	0.160085	-0.4
thal	0.065317	0.211452	-0.160370	0.062870	0.096810	-0.032752	-0.010473	-0.094910	0.205826	0.209090	-0.103314	0.160085	1.000000	-0.3
target	-0.221476	-0.283609	0.432080	-0.146269	-0.081437	-0.026826	0.134874	0.419955	-0.435601	-0.429146	0.343940	-0.408992	-0.343101	1.0





In [16]: `data.head(1)`

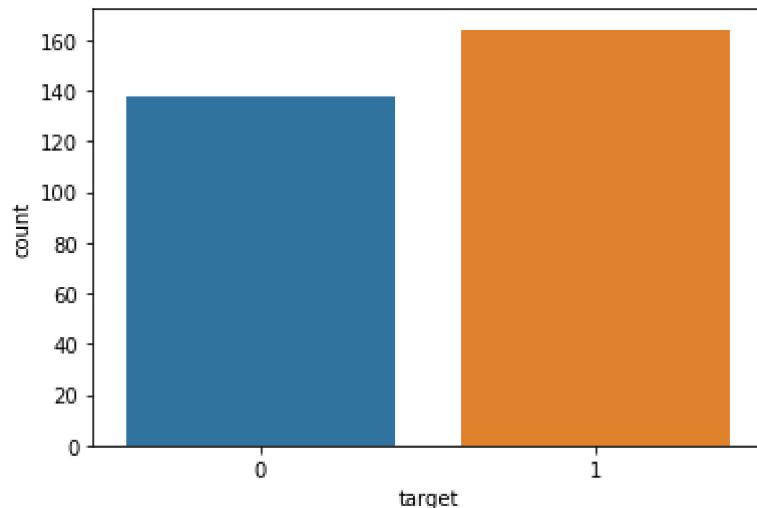
```
Out[16]:    age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0      52     1     0      125    212     0      1     168     0      1.0      2      2      3      0
```

```
In [21]: data['target'].value_counts()
```

```
Out[21]: 1    164
0    138
Name: target, dtype: int64
```

```
In [20]: sns.countplot(x='target', data=data)
```

```
Out[20]: <AxesSubplot:xlabel='target', ylabel='count'>
```

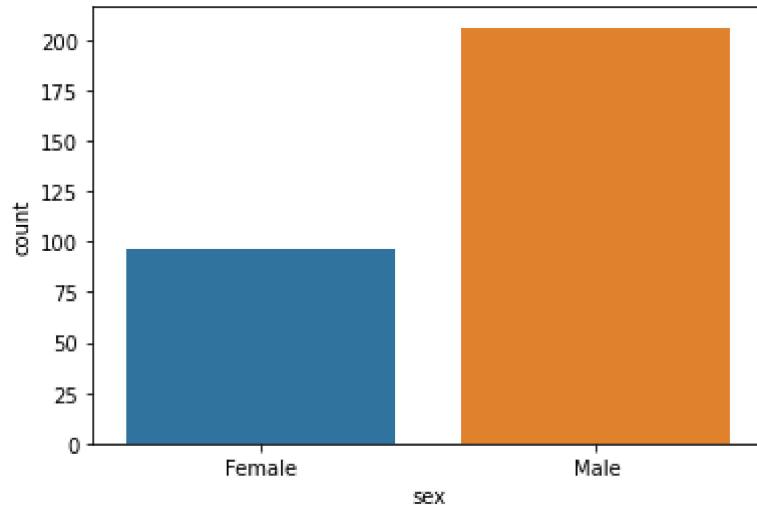


```
In [22]: data['sex'].value_counts()
```

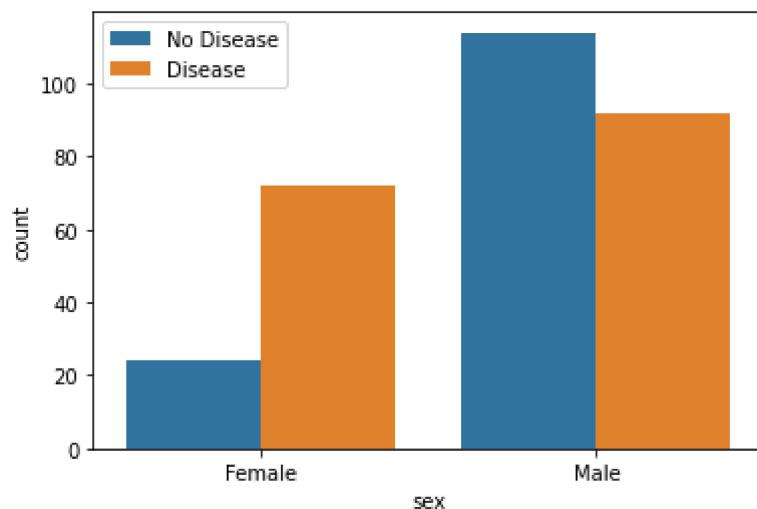
```
Out[22]: 1    206
0     96
Name: sex, dtype: int64
```

```
In [30]: sns.countplot(x='sex', data=data)
plt.xticks([0,1],['Female','Male'])
```

```
Out[30]: ([
```



```
In [33]: sns.countplot(x='sex', data=data, hue='target')
plt.xticks([0,1],['Female','Male'])
plt.legend(['No Disease', 'Disease'])
plt.show()
```



```
In [ ]:
```

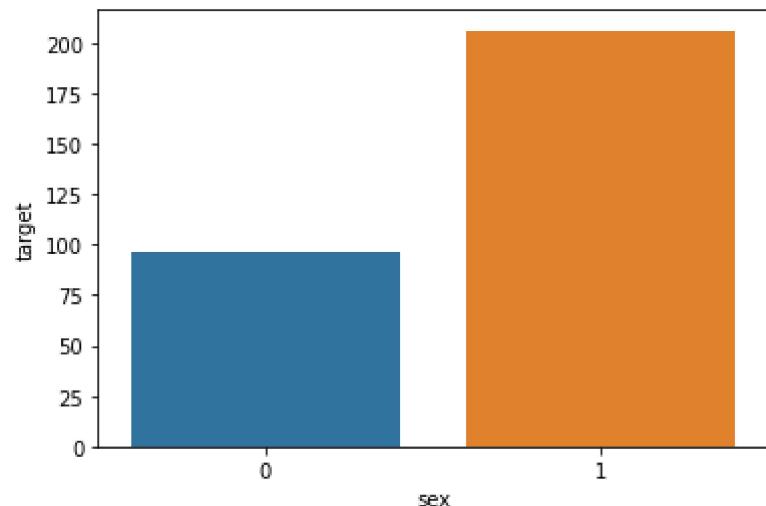
```
In [45]: df=data.groupby('sex').count()[['target']]
df
```

```
Out[45]: target
```

sex	target
0	96
1	206

```
In [46]: sns.barplot(x=df.index,y='target',data=df)
```

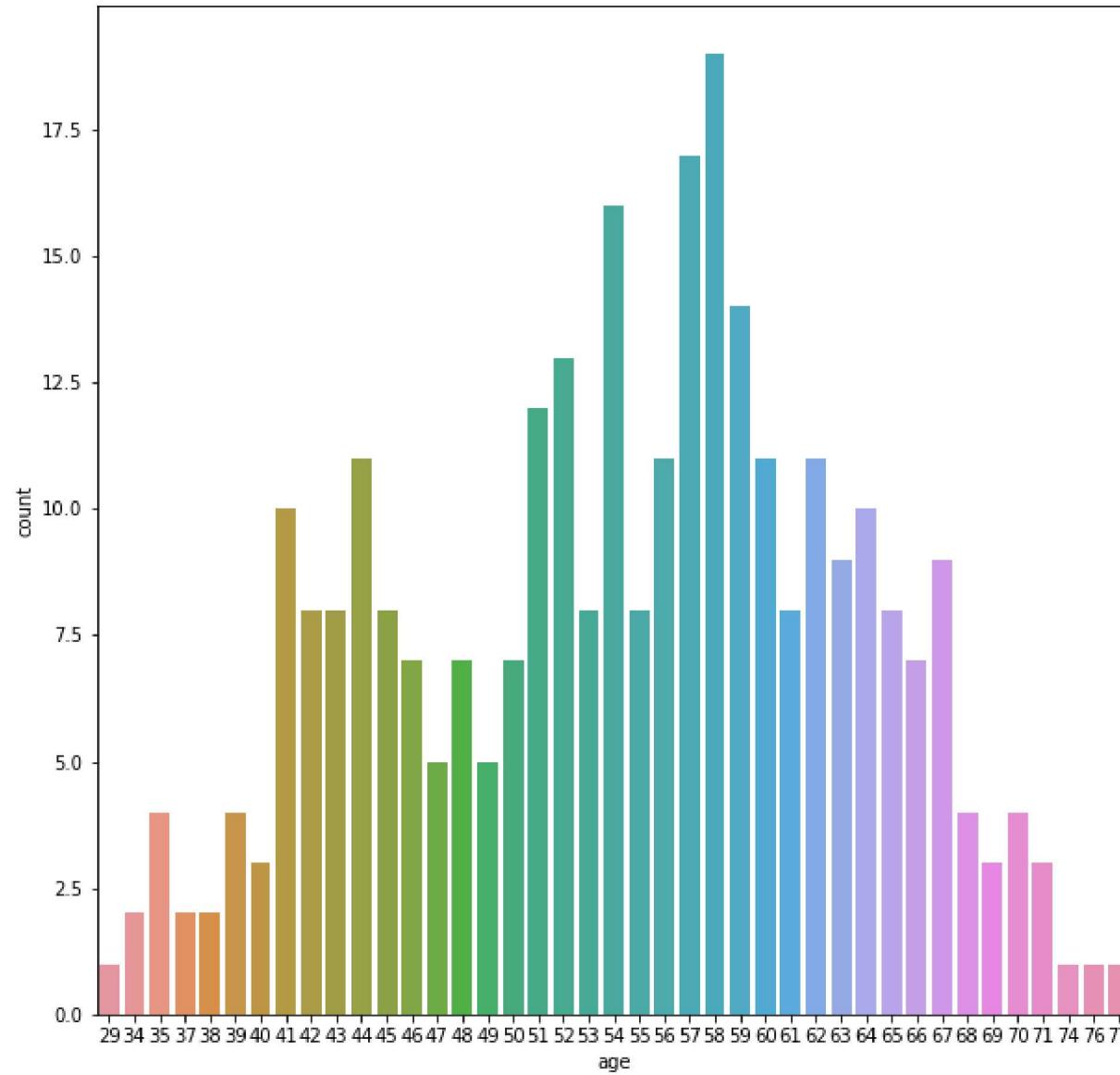
```
Out[46]: <AxesSubplot:xlabel='sex', ylabel='target'>
```



```
In [ ]:
```

```
In [50]: plt.figure(figsize=(10,10))
sns.countplot(x='age',data=data)
```

```
Out[50]: <AxesSubplot:xlabel='age', ylabel='count'>
```



```
In [53]: data['age'].value_counts().head()
```

```
Out[53]:
```

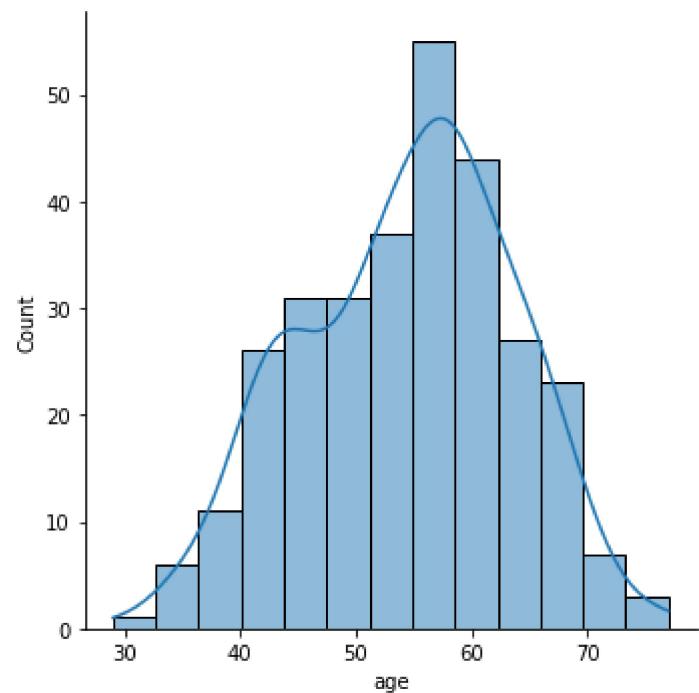
58	19
57	17
54	16
59	14
52	13

```
Name: age, dtype: int64
```

```
In [ ]:
```

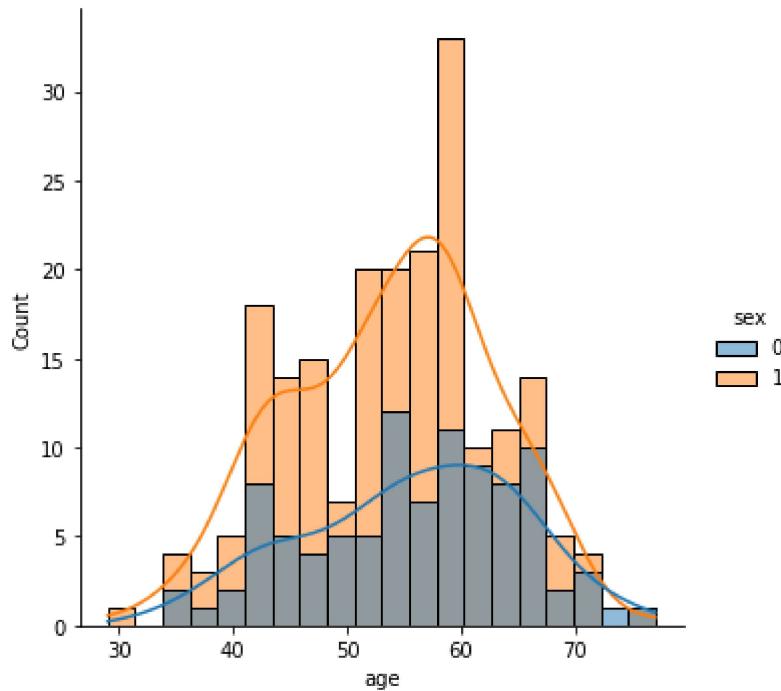
```
In [68]: sns.displot(x='age', data=data, kde=True)
```

```
Out[68]: <seaborn.axisgrid.FacetGrid at 0x11490775bb0>
```



```
In [70]: sns.displot(x='age', data=data, kde=True, bins=20, hue='sex')
```

```
Out[70]: <seaborn.axisgrid.FacetGrid at 0x114906dc9d0>
```



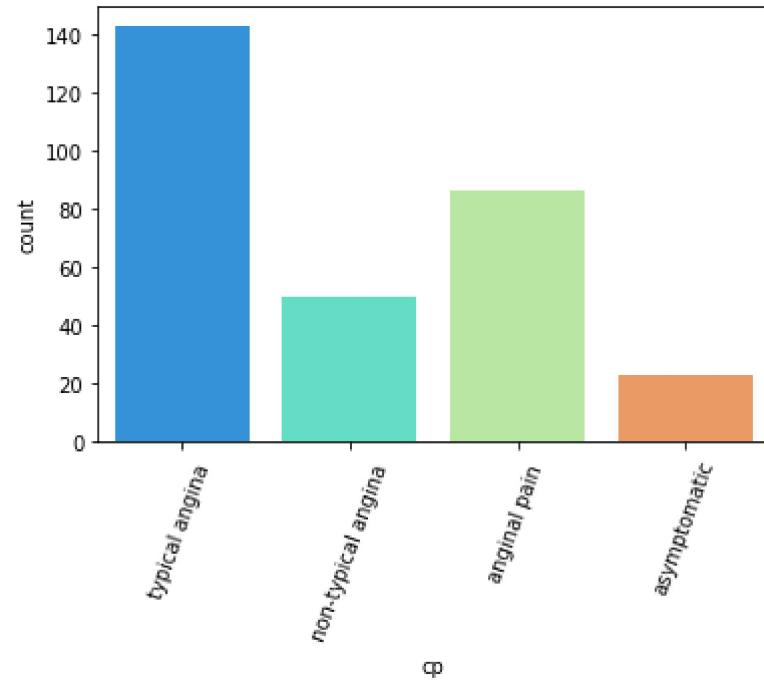
```
In [72]: data.head(1)
```

```
Out[72]:    age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0      52     1    0       125   212     0      1        0     168      0      1.0      2      2      3      0
```

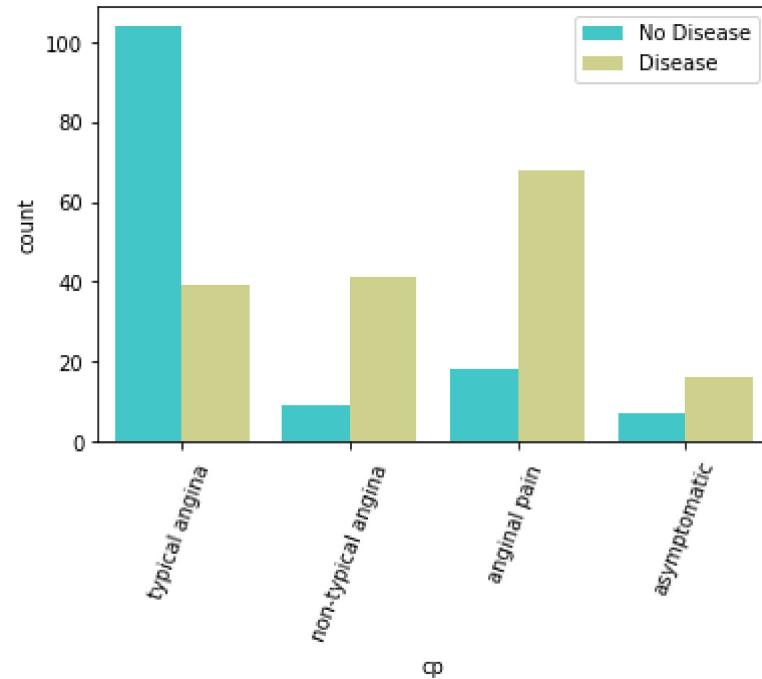
```
In [74]: data['cp'].value_counts()
```

```
Out[74]: 0    143
2     86
1     50
3     23
Name: cp, dtype: int64
```

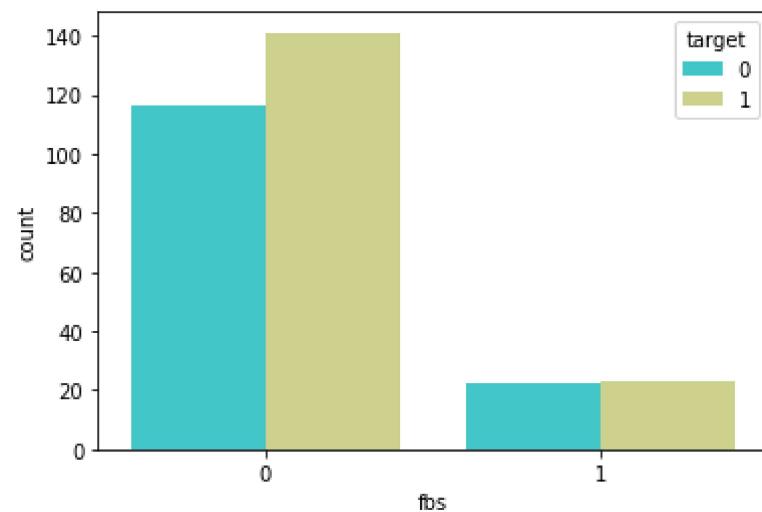
```
In [79]: sns.countplot(x='cp', data=data, palette='rainbow')
plt.xticks([0,1,2,3],['typical angina ',' non-typical angina ', 'anginal pain ','asymptomatic'],rotation=70)
plt.show()
```



```
In [81]: sns.countplot(x='cp',data=data,palette='rainbow',hue='target')
plt.xticks([0,1,2,3],['typical angina ',' non-typical angina ',' anginal pain ','asymptomatic'],rotation=70)
plt.legend(['No Disease','Disease'])
plt.show()
```



```
In [82]: sns.countplot(x='fbs', data=data, palette='rainbow', hue='target')
plt.show()
```



```
In [84]: data.head(1)
```

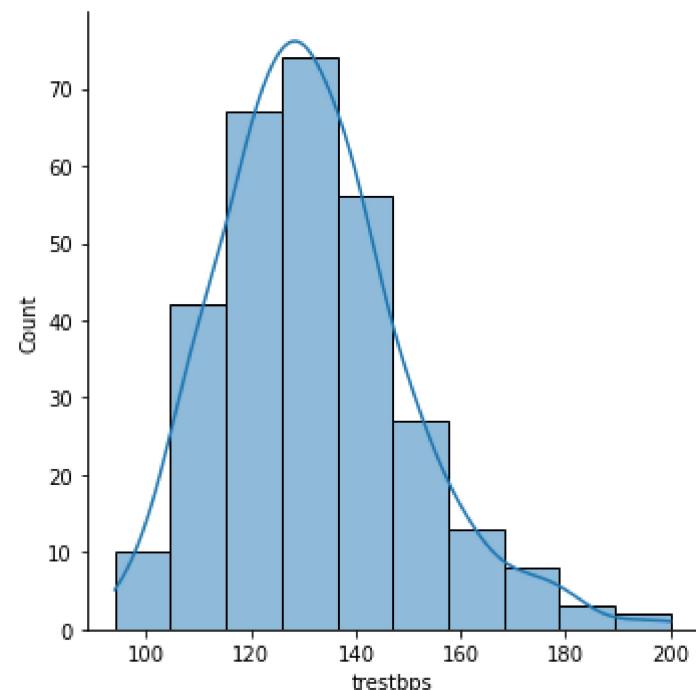
```
Out[84]:   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0      52     1    0       125    212     0      1       168      0      1.0      2      2      3      0
```

```
In [86]: data['trestbps'].head()
```

```
Out[86]: 0    125
1    140
2    145
3    148
4    138
Name: trestbps, dtype: int64
```

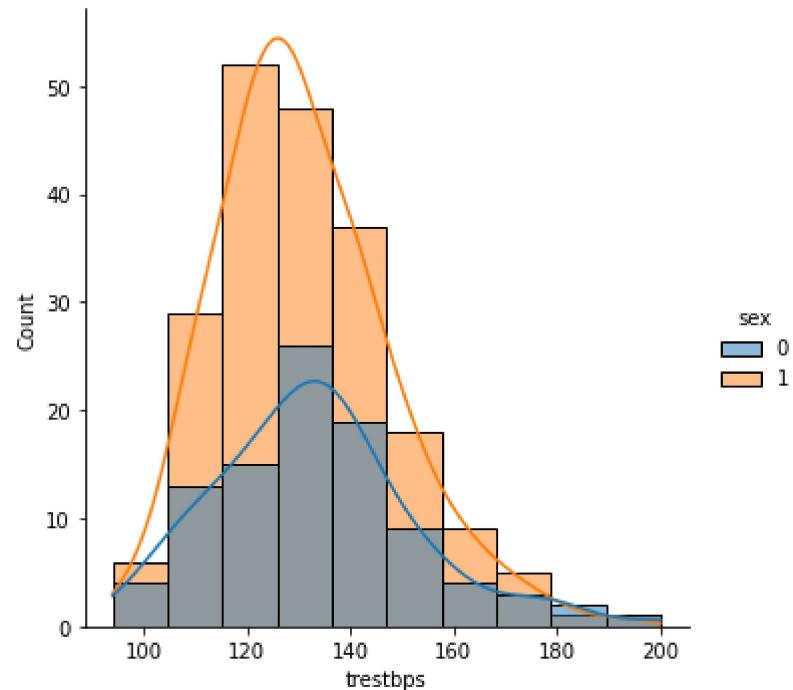
```
In [90]: sns.displot(x='trestbps', data=data, kde=True, bins=10)
```

```
Out[90]: <seaborn.axisgrid.FacetGrid at 0x11491d1c640>
```



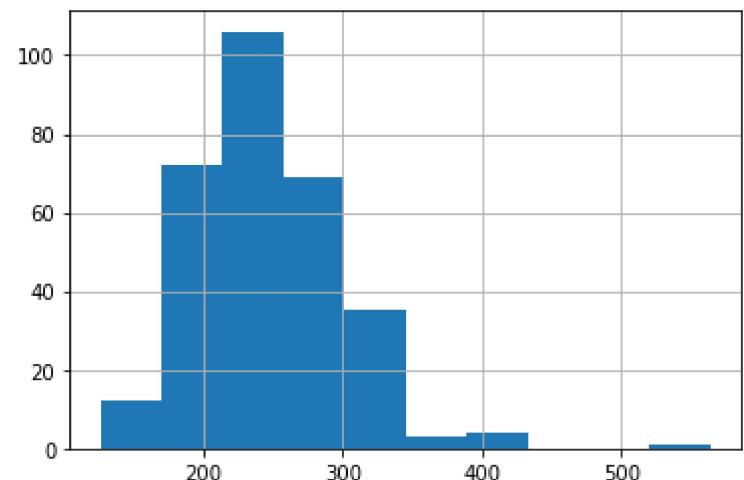
```
In [91]: sns.displot(x='trestbps', data=data, hue='sex', kde=True, bins=10)
```

```
Out[91]: <seaborn.axisgrid.FacetGrid at 0x11491c89a90>
```



```
In [92]: data['chol'].hist()
```

```
Out[92]: <AxesSubplot:>
```



```
In [93]: data.columns
```

```
Out[93]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
   'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
              dtype='object')
```

```
In [94]: cat=[]
cont=[]
for i in data.columns:
    if data[i].nunique()<=10:
        cat.append(i)
    else:
        cont.append(i)
```

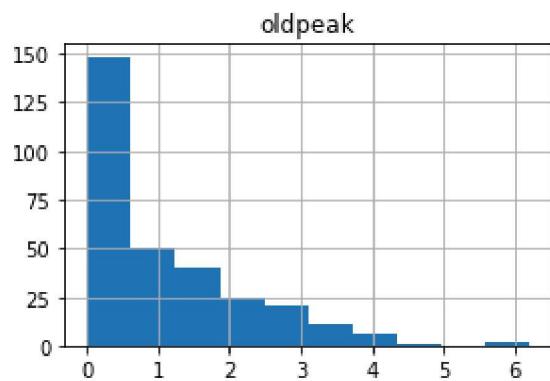
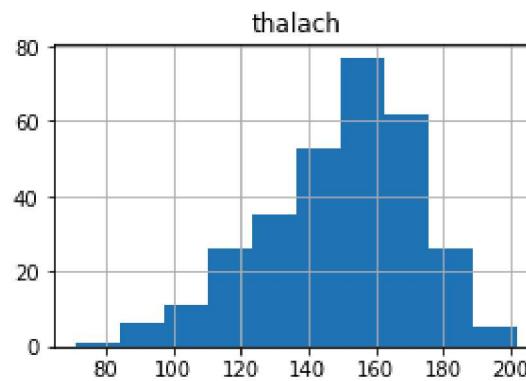
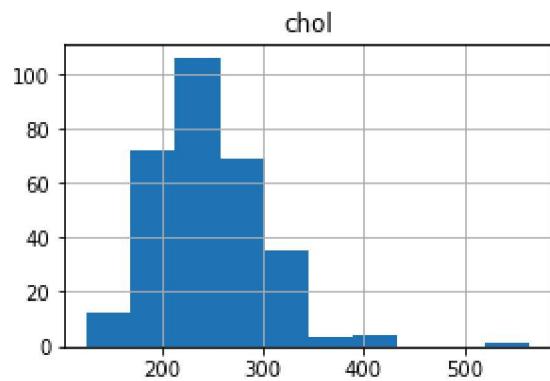
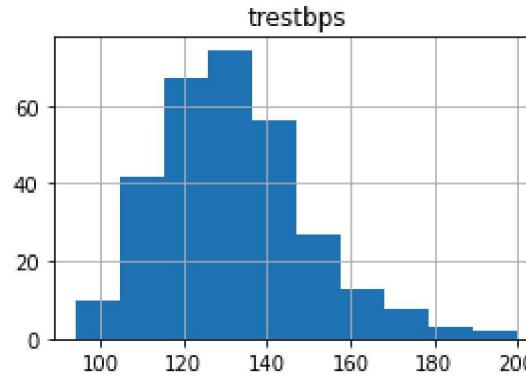
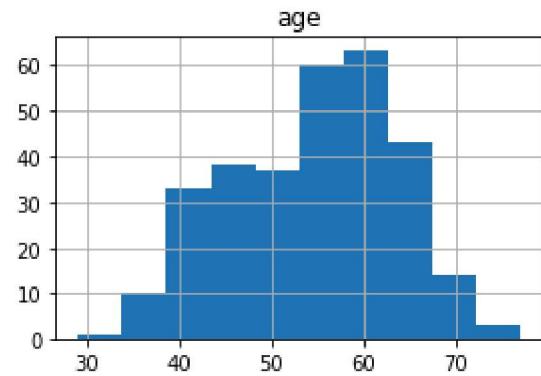
```
In [95]: cat
```

```
Out[95]: ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

```
In [96]: cont
```

```
Out[96]: ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

```
In [102...]: data.hist(cont,figsize=(10,10))
plt.show()
```



In []:

In []: