```
import pandas as pd
import numpy as np
import seaborn as sns
from scipy.stats import zscore,ttest_ind
import matplotlib.pyplot as plt
import statsmodels.stats.api as sm
import  cv2
from sklearn.cluster import KMeans
```

```
file=pd.read_csv("heart_disease_uci.csv")
file.head()
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150.0
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108.0
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	normal	187.0
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172.0

```
file=file.drop_duplicates()
file.shape
```

```
(920, 16)
```

```
file.isnull().sum()
```

```
id            0
age           0
sex           0
dataset       0
cp            0
trestbps     59
chol         30
fbs          90
restecg       2
thalch       55
exang        55
oldpeak      62
slope       309
ca          611
thal        486
num           0
dtype: int64
```

```
file.dtypes
```

```
id            int64
age           int64
sex          object
dataset      object
cp           object
trestbps    float64
chol        float64
fbs          object
restecg      object
thalch      float64
exang        object
```

```
oldpeak     float64
slope        object
ca          float64
thal         object
num           int64
dtype: object
```

```
for i in file.columns:
  if any(file[i].isnull()):
        if file[i].dtype=='int64' or file[i].dtype=='float64' :
                 file[i].fillna(file[i].mean(),inplace=True)
        if file[i].dtype=='object':
                 file[i].fillna(file[i].mode().iloc[0],inplace=True)
```

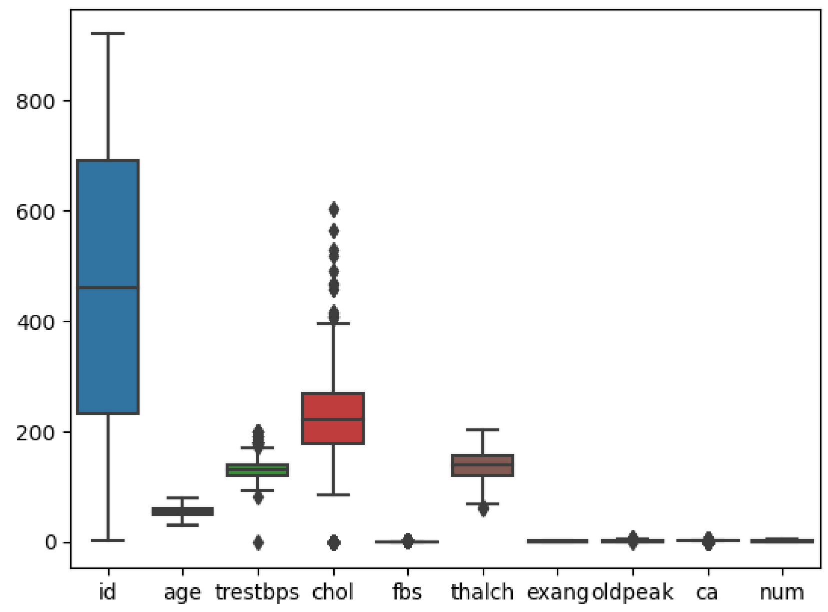
```
file.isnull().sum()
```

```
id          0
age         0
sex         0
dataset     0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalch      0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
num         0
dtype: int64
```

Removing Outliers:

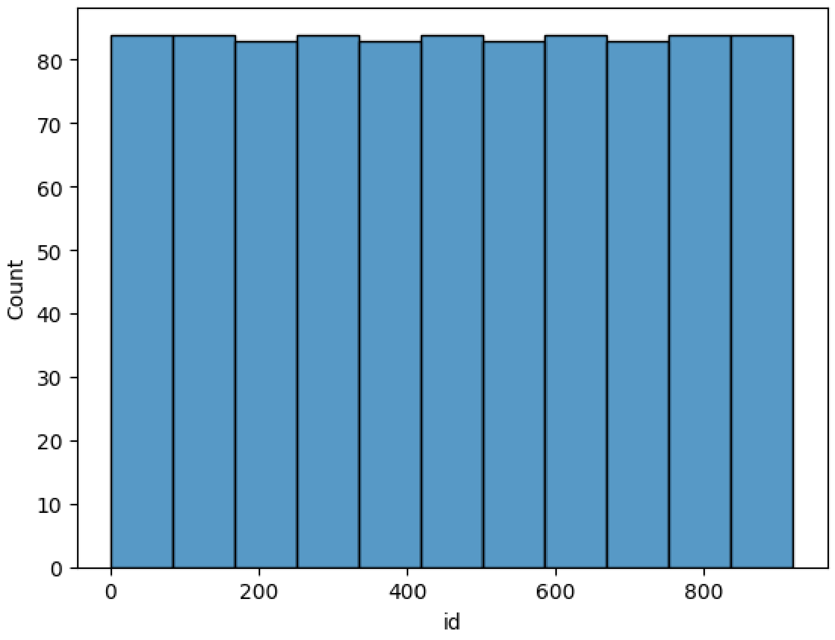
```
def remove_outliers(data, threshold=3):
 z_scores=zscore(data)
 outlier= np.abs(z_scores) > threshold
 cleaned_data = data[~outlier]
 return cleaned_data
```

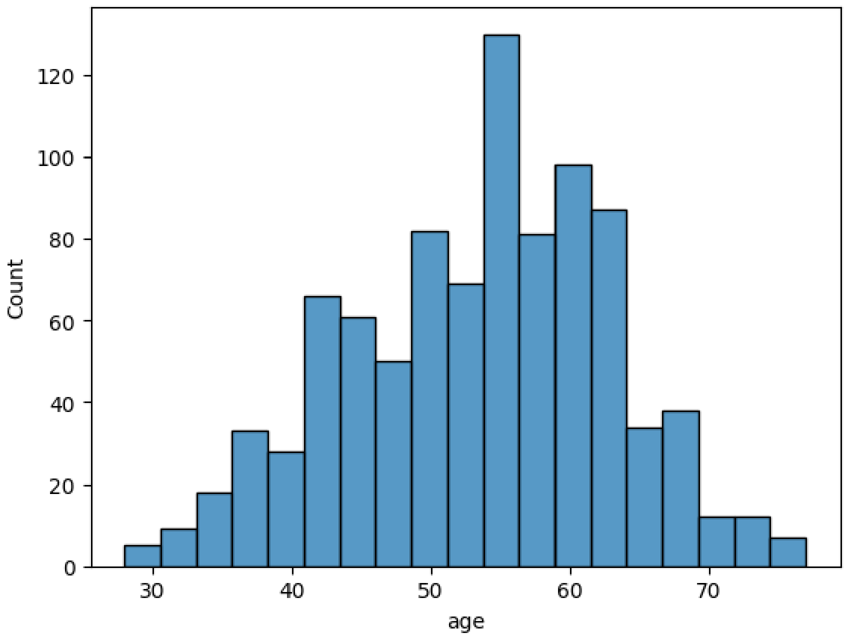
```
sns.boxplot(data=file)
plt.show(  )
remove_outliers(file.select_dtypes(include=['int','float']),threshold=3)
```

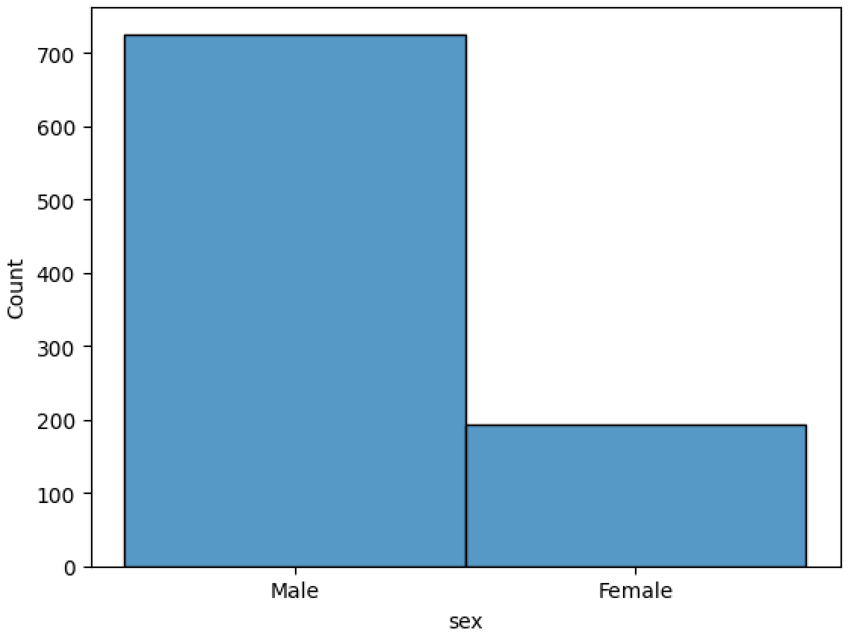


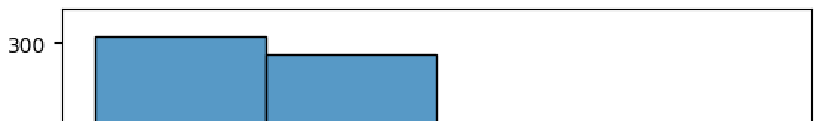
```
   id  age    trestbps   chol      thalch   oldpeak         ca   num
```

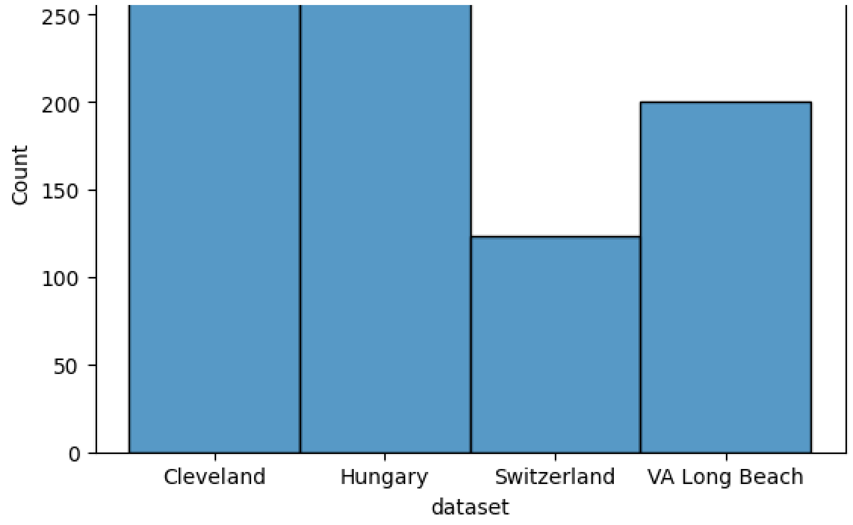
```
for i in file.columns:
  sns.histplot(file[i])
  plt.show()
```

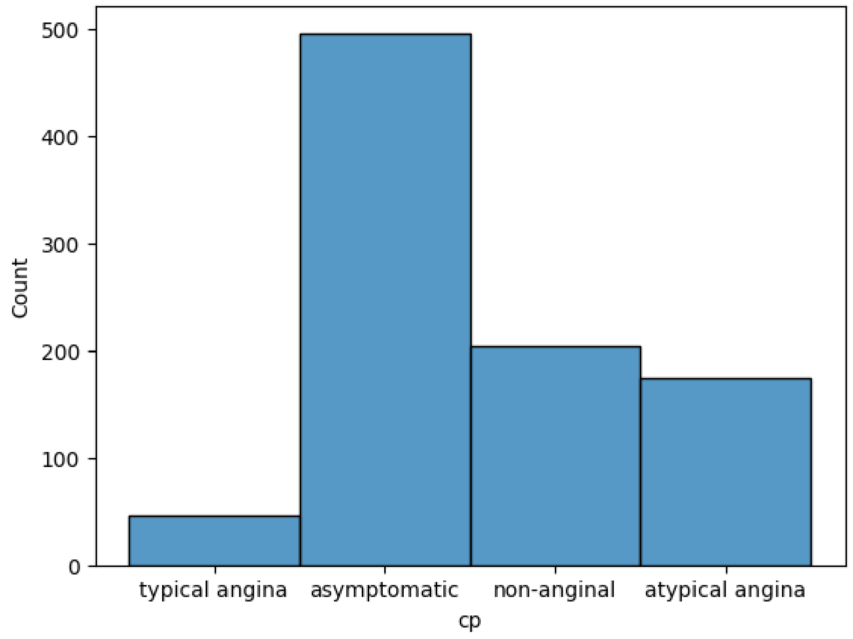

Count
id
0
200
400
600
800
0
10
20
30
40
50
60
70
80

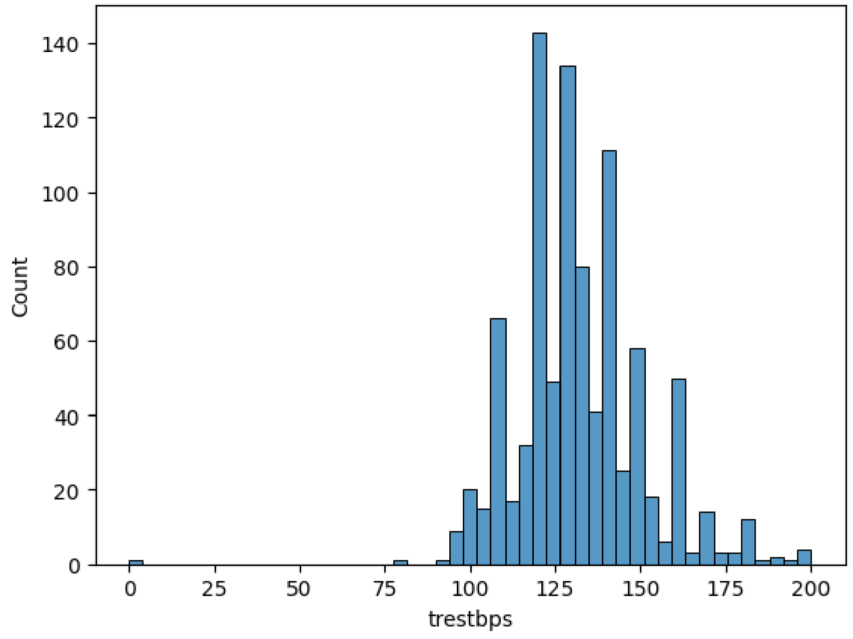

Count
age
30
40
50
60
70
0
20
40
60
80
100
120

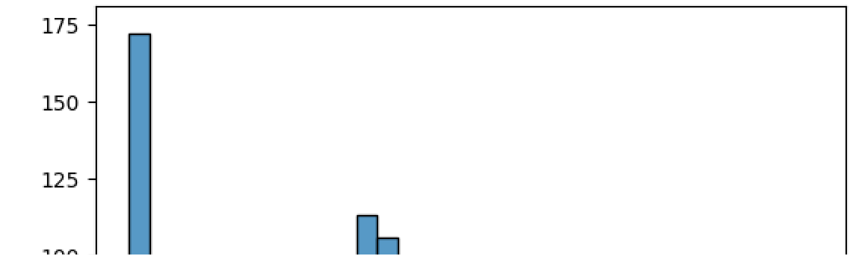

Count
sex
Male
Female
0
100
200
300
400
500
600
700

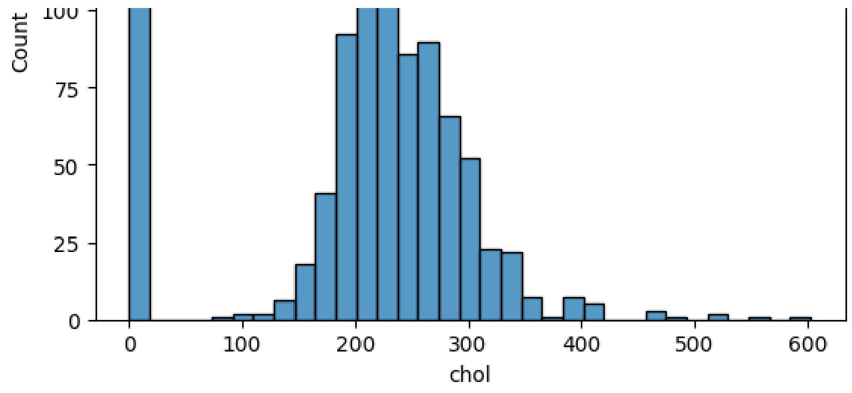

300


Count
250
200
150
100
50
0
Cleveland
Hungary
Switzerland
VA Long Beach
dataset

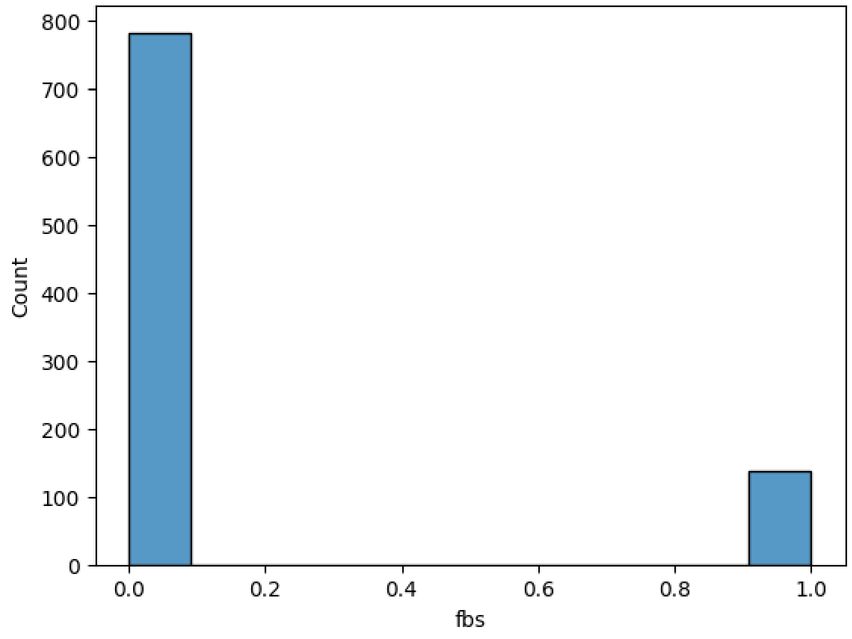

Count
500
400
300
200
100
0
typical angina
asymptomatic
non-anginal
atypical angina
cp

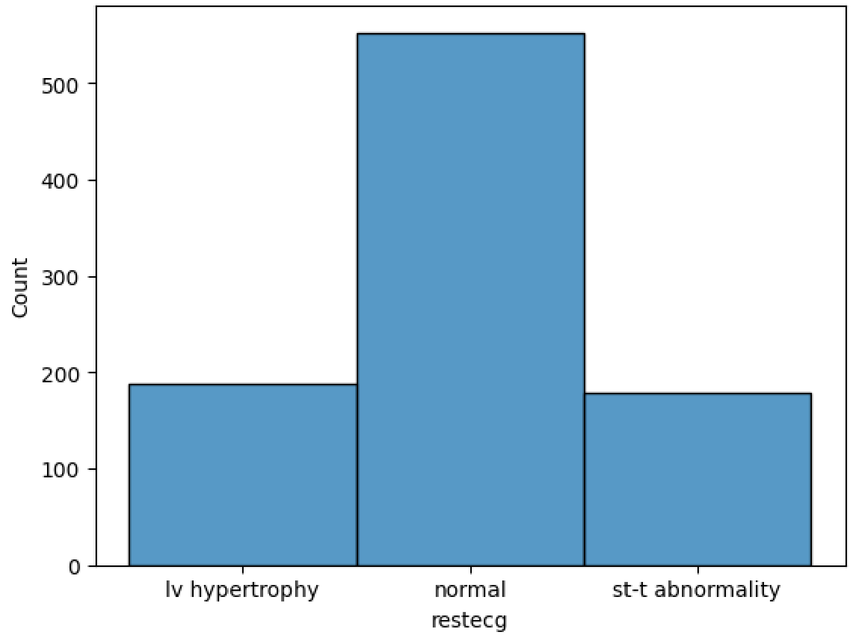

Count
140
120
100
80
60
40
20
0
0
25
50
75
100
125
150
175
200
trestbps

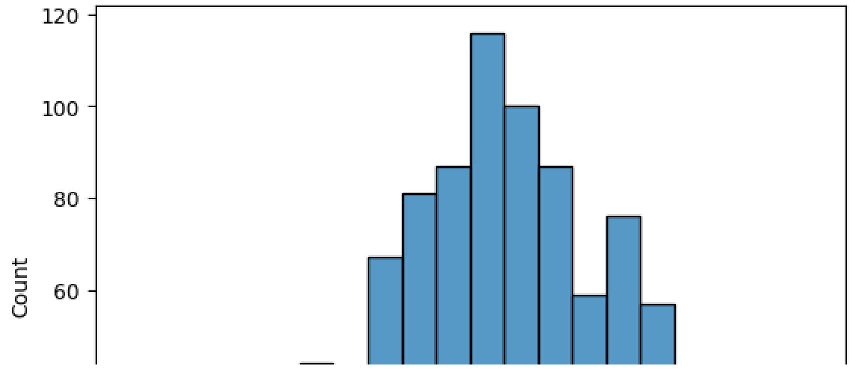

175
150
125

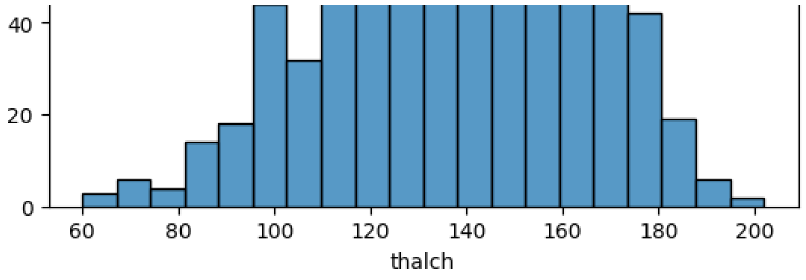


```
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool to
```

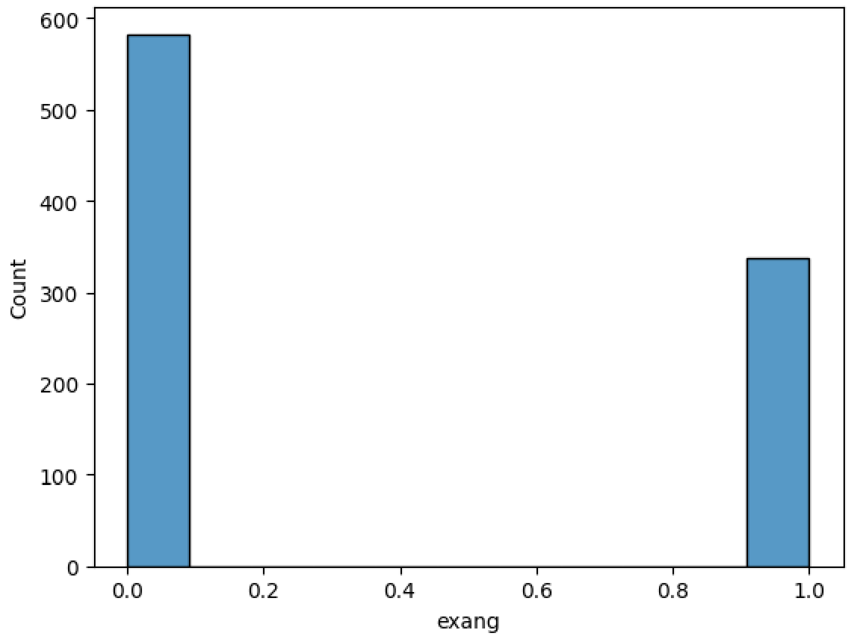


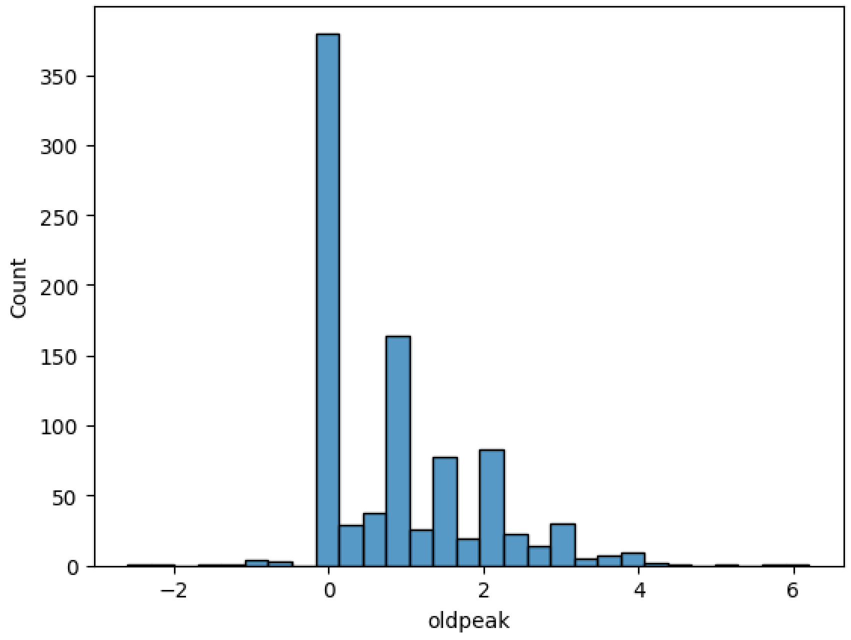


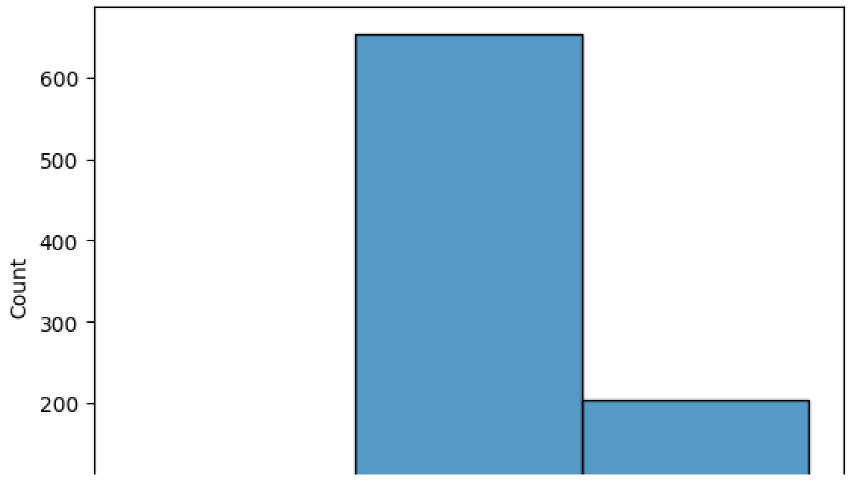


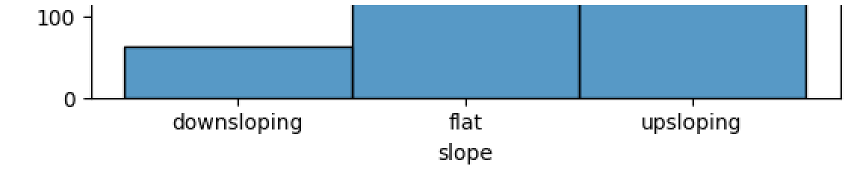


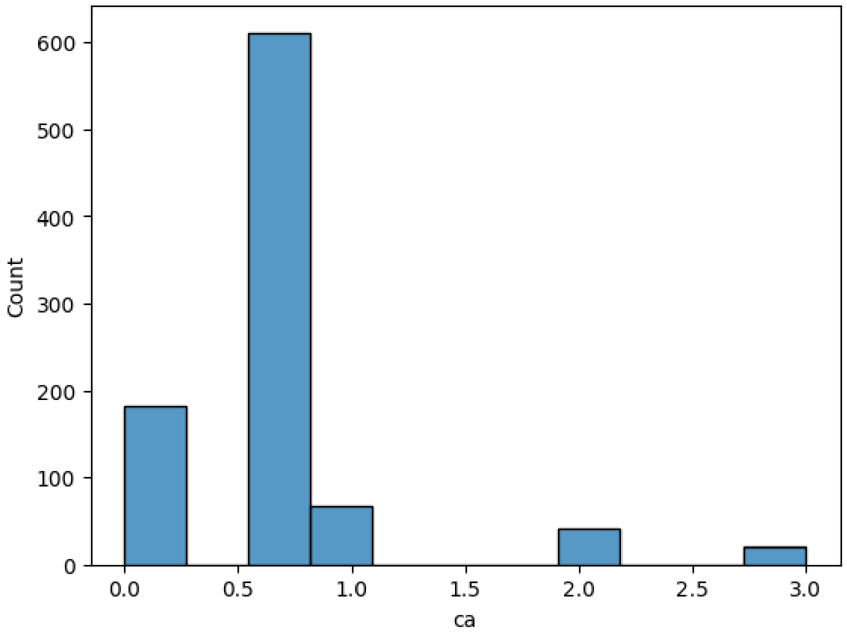
```
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool to
```

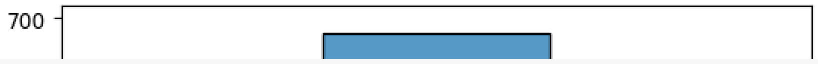






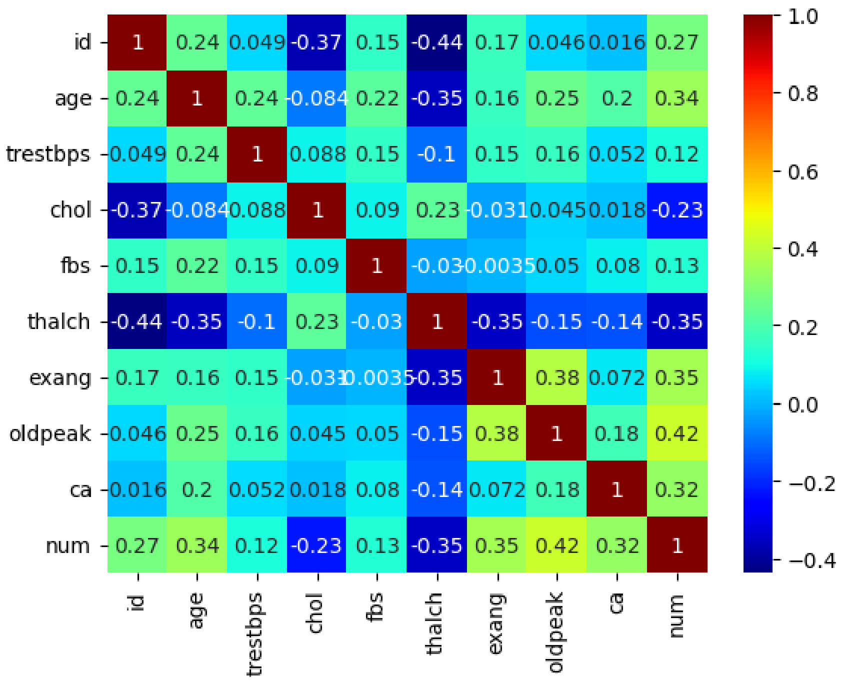






```
correlation=file.corr()
sns.heatmap(correlation,annot=True,cmap='jet' )
```

```
<ipython-input-97-5c4c79bf2f5f>:1: FutureWarning: The default value of numeric_onl
  correlation=file.corr()
<Axes: >
```



	id	age	trestbps	chol	fbs	thalch	exang	oldpeak	ca	num
id	1	0.24	0.049	-0.37	0.15	-0.44	0.17	0.046	0.016	0.27
age	0.24	1	0.24	-0.084	0.22	-0.35	0.16	0.25	0.2	0.34
trestbps	0.049	0.24	1	0.088	0.15	-0.1	0.15	0.16	0.052	0.12
chol	-0.37	-0.084	0.088	1	0.09	0.23	-0.031	0.045	0.018	-0.23
fbs	0.15	0.22	0.15	0.09	1	-0.03	0.0035	0.05	0.08	0.13
thalch	-0.44	-0.35	-0.1	0.23	-0.03	1	-0.35	-0.15	-0.14	-0.35
exang	0.17	0.16	0.15	-0.031	0.0035	-0.35	1	0.38	0.072	0.35
oldpeak	0.046	0.25	0.16	0.045	0.05	-0.15	0.38	1	0.18	0.42
ca	0.016	0.2	0.052	0.018	0.08	-0.14	0.072	0.18	1	0.32
num	0.27	0.34	0.12	-0.23	0.13	-0.35	0.35	0.42	0.32	1

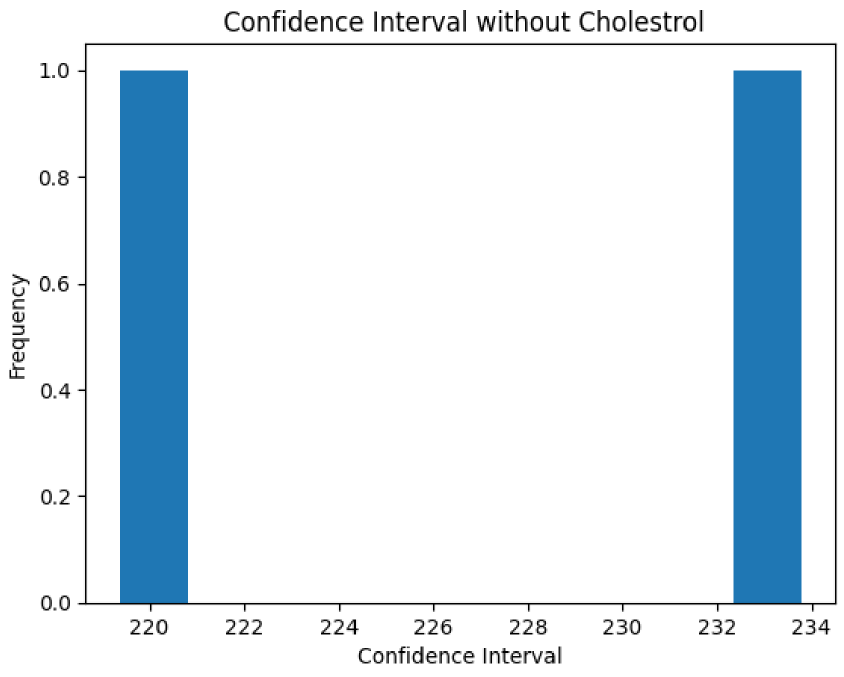
```
chol_with_disease = file[file['num'] > 0]['chol']
chol_without_disease = file[file['num'] == 0]['chol']
t_statistic, p_value = ttest_ind(chol_with_disease, chol_without_disease)
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis. There is a significant difference in cholesterol levels between patients with hea
else:
    print("Fail to reject the null hypothesis. There is no significant difference in cholesterol levels between patients
```

```
Reject the null hypothesis. There is a significant difference in cholesterol levels between patients with heart di
```

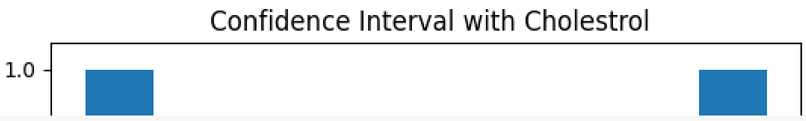
```
confident_interval_with_cholesterol = sm.DescrStatsW(chol_with_disease).tconfint_mean()
confident_interval_without_cholesterol = sm.DescrStatsW(chol_without_disease).tconfint_mean()
print("Confidence Interval for Cholesterol with heart disease= ", confident_interval_with_cholesterol)
print("Confidence Interval for Cholesterol without heart disease= ", confident_interval_without_cholesterol)
```

```
Confidence Interval for Cholesterol with heart disease=  (165.98214381850937, 187.9566919687801)
Confidence Interval for Cholesterol without heart disease=  (219.37078787273725, 233.77994888879255)
```

```
plt.hist(confident_interval_without_cholesterol,bins=10)
plt.xlabel("Confidence Interval")
plt.ylabel("Frequency")
plt.title(" Confidence Interval without Cholestrol")
plt.show()
```

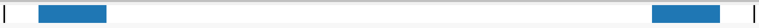


```
plt.hist(confident_interval_with_cholesterol,bins=10)
plt.xlabel("Confidence Interval")
plt.ylabel("Frequency")
plt.title(" Confidence Interval with Cholestrol")
plt.show()
```



```
BP_with_disease = file[file['num'] > 0]['trestbps']
BP_without_disease = file[file['num'] == 0]['trestbps']
t_statistic, p_value = ttest_ind(BP_with_disease, BP_without_disease)
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis. There is a significant difference in blood pressure levels between patients with
else:
    print("Fail to reject the null hypothesis. There is no significant difference in blood pressure levels between patie
```

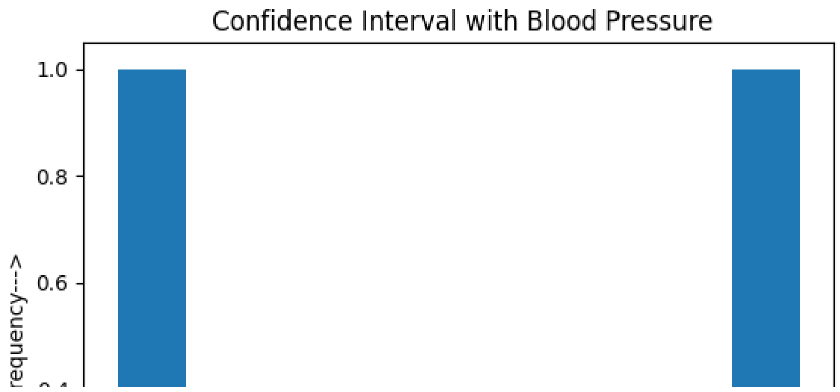
```
Reject the null hypothesis. There is a significant difference in blood pressure levels between patients with and w
```



```
confident_interval_with_BP = sm.DescrStatsW(BP_with_disease).tconfint_mean()
confident_interval_without_BP = sm.DescrStatsW(BP_without_disease ).tconfint_mean()
print("Confidence Interval for Cholesterol with heart disease= ",confident_interval_with_BP)
print("Confidence Interval for Cholesterol without heart disease= ",confident_interval_without_BP)
```

```
Confidence Interval for Cholesterol with heart disease=  (132.11707570160883, 135.55743810218763)
Confidence Interval for Cholesterol without heart disease=  (128.4249918770566, 131.6170912549321)
```

```
plt.hist(confident_interval_with_BP,bins=10)
plt.xlabel("Confidence Interval--->")
plt.ylabel("Frequency--->")
plt.title(" Confidence Interval with Blood Pressure")
plt.show()
plt.hist(confident_interval_without_BP,bins=10)
plt.xlabel("Confidence Interval--->")
plt.ylabel("Frequency--->")
plt.title(" Confidence Interval without Blood Pressure")
plt.show()
```



TASK#2:

```
image_file=cv2.imread("image.jpg")
plt.title("Original Image:")
plt.imshow(image_file)
plt.axis('off')
plt.show()
image_file.shape
```

Original Image:

```
(1080, 1920, 3)
```

Interpretation of image_file.shape:

->Here 1080 represents the heigth of image pixels.

->And 1920 represents the length/width of image pixels(horizontally)

->3=RGB color.

Converting Into 2D shape:

```
two_D_image_file=image_file.reshape(-1,3)
two_D_image_file.shape
```

```
(2073600, 3)
```

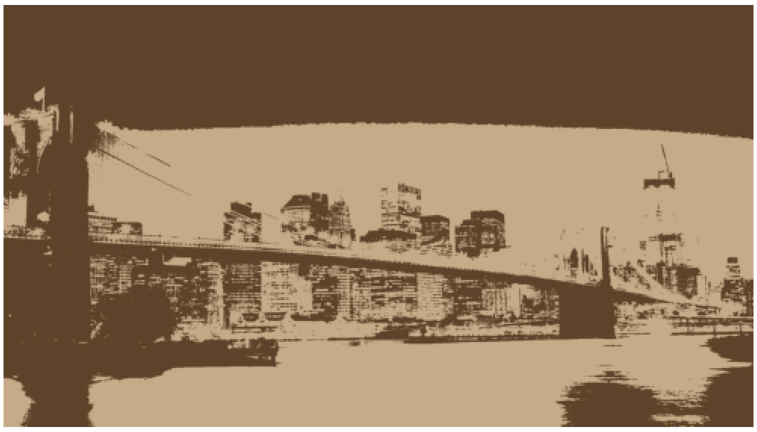
Defining values in K:

```
K=[2,3,5,10,15,20]
for i in K:
  kmeans=KMeans(n_clusters=i,random_state=0)
  kmeans.fit(two_D_image_file)
```

```
labels=kmeans.labels_
centroid_value=kmeans.cluster_centers_[labels]
original_image_shape=centroid_value.reshape(image_file.shape)
plt.title('Image with K=' + str(i) + ' value')
plt.imshow(original_image_shape.astype(np.uint8))
plt.savefig('Image with cluster K='+str(i)+' .jpg',bbox_inches='tight', transparent=True,pad_inches=0)
plt.show()
plt.show()
```

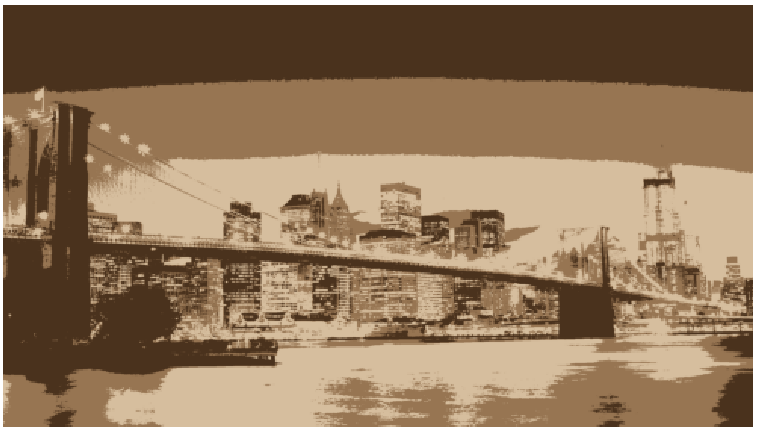
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarn
  warnings.warn(
```

Image with K=2 value



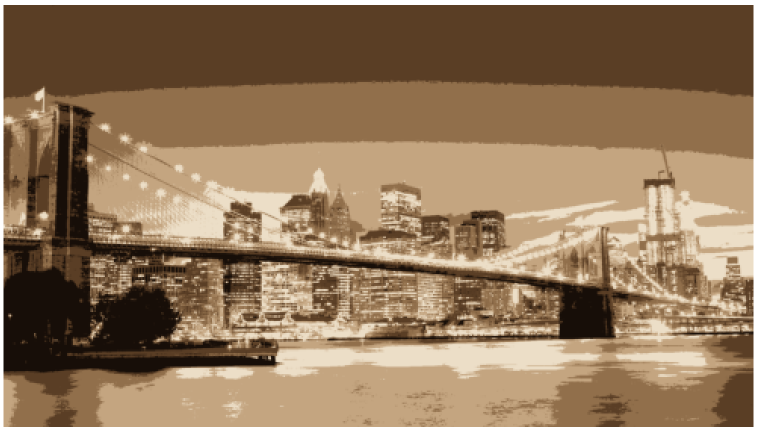
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarn
  warnings.warn(
```

Image with K=3 value



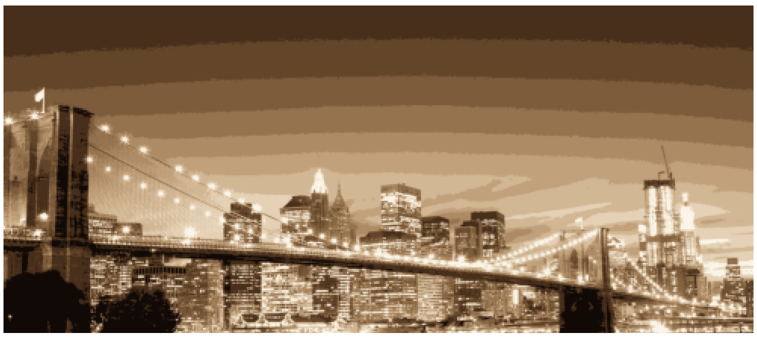
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarn
  warnings.warn(
```

Image with K=5 value



```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarn
  warnings.warn(
```

Image with K=10 value



here