In [559…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

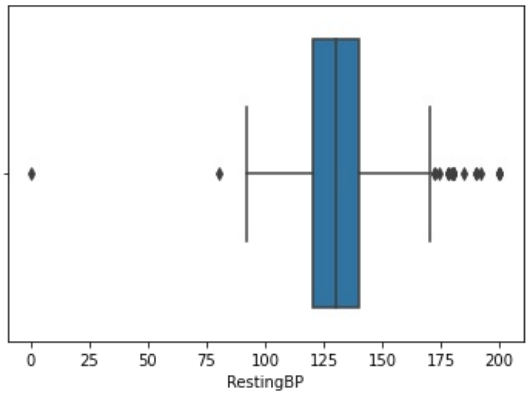
In [560…
```
df=pd.read_csv('F:\\NEW_DATSET\\heart.csv')
df.head()
```

Out[560…

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

In [561…
```
sns.boxplot(x='RestingBP',data=df)
```

Out[561… <AxesSubplot:xlabel='RestingBP'>



removing outlier

In [562…
```
q1=df.RestingBP.quantile(0.25)
q1
```

Out[562… 120.0

q1

In [563…
```
q2=df.RestingBP.quantile(0.75)
q2
```

Out[563… 140.0

In [564…
```
iqr=q2-q1
cut_off=iqr*1.5
iqr,cut_off
```

Out[564… (20.0, 30.0)

In [565…
```
upper,lower=q2+cut_off,q1-cut_off
df=df[(df.RestingBP>lower)&(df.RestingBP<upper)]
df.describe()
```

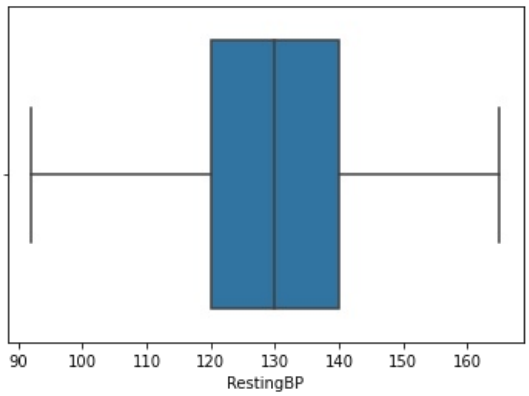
Out[565…

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	876.000000	876.000000	876.000000	876.000000	876.000000	876.000000	876.000000
mean	53.335616	130.492009	198.644977	0.229452	137.067352	0.869635	0.546804
std	9.469671	15.176634	108.722171	0.420721	25.374071	1.055420	0.498089
min	28.000000	92.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.000000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	222.000000	0.000000	138.000000	0.500000	1.000000
75%	60.000000	140.000000	266.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	165.000000	603.000000	1.000000	202.000000	6.200000	1.000000

In [566…

```
sns.boxplot(x='RestingBP',data=df)
```

Out[566… <AxesSubplot:xlabel='RestingBP'>



In [567…

```
print(upper,lower)
```

```
170.0 90.0
```

In [568…

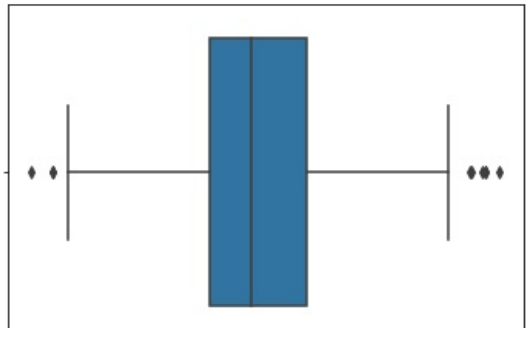
```
q1=df.Cholesterol.quantile(0.25)
q2=df.Cholesterol.quantile(0.75)
iqr=q2-q1
cut_off=iqr*1.5
upper,lower=q2+cut_off,q1-cut_off



print(upper,lower)
```

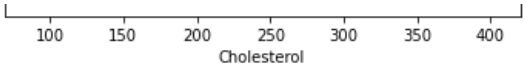
```
405.5 33.5
```

In [569…

```
df=df[(df.Cholesterol>lower)&(df.Cholesterol<upper)]

sns.boxplot(x='Cholesterol',data=df)
```

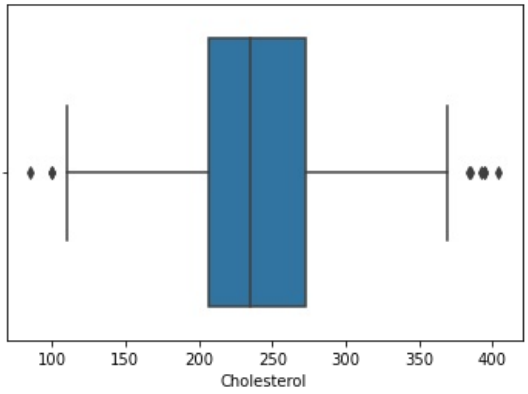
Out[569… <AxesSubplot:xlabel='Cholesterol'>





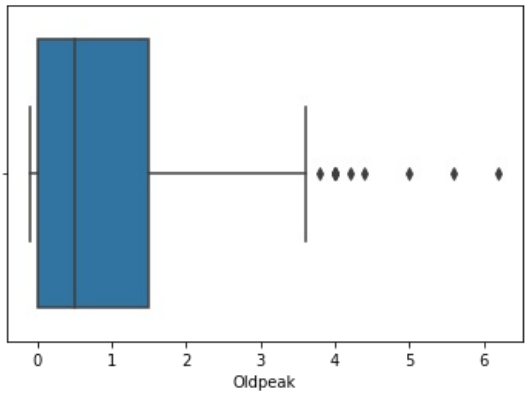
In [570…
```
sns.boxplot(x='Cholesterol',data=df)
```

Out[570…
```
<AxesSubplot:xlabel='Cholesterol'>
```



In [571…
```
sns.boxplot(x='Oldpeak',data=df)
```

Out[571…
```
<AxesSubplot:xlabel='Oldpeak'>
```

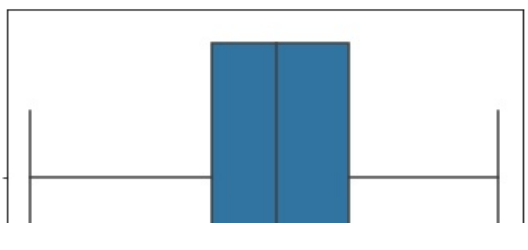


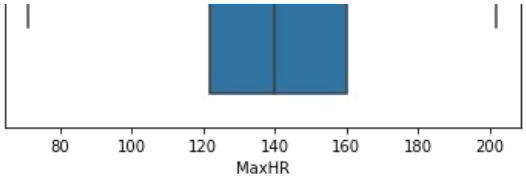
In [572…
```
q1=df.MaxHR.quantile(0.25)
q2=df.MaxHR.quantile(0.75)
iqr=q2-q1
cut_off=iqr*1.5
upper,lower=q2+cut_off,q1-cut_off



print(upper,lower)
```

```
217.0 65.0
```

In [573…
```
df=df=df[(df.MaxHR>lower)&(df.MaxHR<upper)]
df.describe()

sns.boxplot(x='MaxHR',data=df)
```

Out[573…
```
<AxesSubplot:xlabel='MaxHR'>
```





In [574…
```
df.head()
```

Out[574…

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

one hot encoding

In [575…
```
df['Sex']=pd.get_dummies(df['Sex'],drop_first=True)
df.head()
```

Out[575…

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	1	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	0	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	1	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	0	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	1	NAP	150	195	0	Normal	122	N	0.0	Up	0

In [576…
```
chestpaintype=pd.get_dummies(df['ChestPainType'],drop_first=True)
rstingecg=pd.get_dummies(df['RestingECG'],drop_first=True)
exercise=pd.get_dummies(df['ExerciseAngina'],drop_first=True)
stslope=pd.get_dummies(df['ST_Slope'],drop_first=True)

df_n=pd.concat([chestpaintype,rstingecg,exercise,stslope,df],axis=1)
```

In [577…
```
df_n.head()
```

Out[577…

	ATA	NAP	TA	Normal	ST	Y	Flat	Up	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngir
0	1	0	0	1	0	0	0	1	40	1	ATA	140	289	0	Normal	172	
1	0	1	0	1	0	0	1	0	49	0	NAP	160	180	0	Normal	156	
2	1	0	0	0	1	0	0	1	37	1	ATA	130	283	0	ST	98	
3	0	0	0	1	0	1	1	0	48	0	ASY	138	214	0	Normal	108	
4	0	1	0	1	0	0	0	1	54	1	NAP	150	195	0	Normal	122	

In [578…
```
df_n.drop(['ChestPainType','RestingECG','ExerciseAngina','ST_Slope'],axis=1,inplace=True)
df_n.head()
```

Out[578…

	ATA	NAP	TA	Normal	ST	Y	Flat	Up	Age	Sex	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
0	1	0	0	1	0	0	0	1	40	1	140	289	0	172	0.0	0
1	0	1	0	1	0	0	1	0	49	0	160	180	0	156	1.0	1
2	1	0	0	0	1	0	0	1	37	1	130	283	0	98	0.0	0
3	0	0	0	1	0	1	1	0	48	0	138	214	0	108	1.5	1
4	0	1	0	1	0	0	0	1	54	1	150	195	0	122	0.0	0

In [579…
```
df_n.describe()
```

Out[579…

	ATA	NAP	TA	Normal	ST	Y	Flat	Up	Age	Sex	RestingBP	Cl
count	702.000000	702.000000	702.000000	702.000000	702.000000	702.000000	702.000000	702.000000	702.000000	702.000000	702.000000	70
mean	0.226496	0.232194	0.052707	0.603989	0.163818	0.378917	0.468661	0.480057	52.710826	0.762108	131.007123	23
std	0.418862	0.422533	0.223606	0.489416	0.370374	0.485463	0.499373	0.499958	9.492718	0.426096	14.705233	5
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	28.000000	0.000000	92.000000	8
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	46.000000	1.000000	120.000000	20
50%	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	54.000000	1.000000	130.000000	23
75%	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	59.000000	1.000000	140.000000	27
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	77.000000	1.000000	165.000000	40

In [580…
```
x=df_n.drop(['HeartDisease'],axis=1)
y=df_n['HeartDisease']
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x=sc.fit_transform(x)

from sklearn.model_selection import train_test_split
x_tr,x_te,y_tr,y_te= train_test_split(x,y,test_size=0.2)
```

In [ ]:
```

```

In [581…
```
from sklearn.svm import SVC
```

In [583…
```
model=SVC(kernel='linear')
model.fit(x_tr,y_tr)
model.score(x_te,y_te)
```

Out[583… 0.8865248226950354

In [584…
```
from mlxtend.plotting import plot_decision_regions
```

In [588…
```
y_te
```

Out[588…
```
821    0
666    0
541    1
559    1
425    1
      ..
268    1
478    1
170    0
849    0
784    1
Name: HeartDisease, Length: 141, dtype: int64
```

In [ ]:
```

```