Heart Attack Prediction

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Data collection :-

In [2]:
```
df=pd.read_csv(r"C:\Users\HP\Downloads\Heart\data.csv",na_values=['?'])
df
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	28	1	2	130.0	132.0	0.0	2.0	185.0	0.0	0.0	NaN	NaN	NaN	0
1	29	1	2	120.0	243.0	0.0	0.0	160.0	0.0	0.0	NaN	NaN	NaN	0
2	29	1	2	140.0	NaN	0.0	0.0	170.0	0.0	0.0	NaN	NaN	NaN	0
3	30	0	1	170.0	237.0	0.0	1.0	170.0	0.0	0.0	NaN	NaN	6.0	0
4	31	0	2	100.0	219.0	0.0	1.0	150.0	0.0	0.0	NaN	NaN	NaN	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
289	52	1	4	160.0	331.0	0.0	0.0	94.0	1.0	2.5	NaN	NaN	NaN	1
290	54	0	3	130.0	294.0	0.0	1.0	100.0	1.0	0.0	2.0	NaN	NaN	1
291	56	1	4	155.0	342.0	1.0	0.0	150.0	1.0	3.0	2.0	NaN	NaN	1
292	58	0	2	180.0	393.0	0.0	0.0	110.0	1.0	1.0	2.0	NaN	7.0	1
293	65	1	4	130.0	275.0	0.0	1.0	115.0	1.0	1.0	2.0	NaN	NaN	1

294 rows × 14 columns

2.Data Exploration:-

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

Out[3]:
```
(294, 14)
```

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

Out[4]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num'],
      dtype='object')
```

In [5]:
```
df['age'].unique()
```

Out[5]:
```
array([28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
       45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
       62, 63, 65, 66], dtype=int64)
```

In [6]:
```
df['sex'].unique()
```

Out[6]:
```
array([1, 0], dtype=int64)
```

In [7]:
```
df['fbs'].unique()
```

Out[7]:
```
array([ 0., nan,  1.])
```

In [8]:
```
df['chol'].unique()
```

Out[8]:
```
array([132., 243.,  nan, 237., 219., 198., 225., 254., 298., 161., 214.,
       220., 160., 167., 308., 264., 166., 340., 209., 260., 211., 173.,
       283., 194., 223., 315., 275., 297., 292., 182., 200., 204., 241.,
       339., 147., 273., 307., 289., 215., 281., 250., 184., 245., 291.,
       295., 269., 196., 268., 228., 358., 201., 249., 266., 186., 207.,
       218., 412., 224., 238., 230., 163., 240., 280., 257., 263., 276.,
       284., 195., 227., 253., 187., 202., 328., 168., 216., 129., 190.,
       188., 179., 210., 272., 180., 100., 259., 468., 274., 320., 221.,
       309., 312., 171., 208., 246., 305., 217., 365., 344., 394., 256.,
       326., 277., 270., 229.,  85., 347., 251., 222., 287., 318., 213.,
       294., 193., 271., 156., 267., 282., 117., 466., 247., 226., 265.,
       206., 288., 303., 338., 248., 306., 529., 392., 231., 329., 355.,
       233., 242., 603., 255., 172., 175., 290., 341., 234., 342., 404.,
       518., 285., 279., 388., 164., 336., 491., 205., 212., 331., 393.])
```

In [9]:
```
df['fbs'].unique()
```

Out[9]:
```
array([ 0., nan,  1.])
```

In [10]:
```
df['thalach'].unique()
```

Out[10]:
```
array([185., 160., 170., 150., 165., 184., 155., 190., 168., 180., 178.,
       172., 130., 142.,  98., 158., 129., 146., 145., 120., 106., 132.,
       140., 138., 167., 188., 144., 137., 136., 152., 175., 176., 118.,
       154., 115., 135., 122., 110.,  90., 116., 174., 125.,  nan, 148.,
       100., 164., 139., 127., 162., 112., 134., 114., 128., 126., 124.,
       153., 166., 103., 156.,  87., 102.,  92.,  99., 121.,  91., 108.,
        96.,  82., 105., 143., 119.,  94.])
```

In [11]:
```
df['cp'].unique()
```

Out[11]:
```
array([2, 1, 3, 4], dtype=int64)
```

In [12]:
```
df['restecg'].unique()
```

Out[12]:
```
array([ 2.,  0.,  1., nan])
```

In [13]:
```
df['slope'].unique()
```

Out[13]:
```
array([nan,  2.,  1.,  3.])
```

In [14]:
```
df['ca'].unique()
```

Out[14]:
```
array([nan,  0.])
```

In [15]:
```
df['thal'].unique()
```

Out[15]:
```
array([nan,  6.,  3.,  7.])
```

In [16]:
```
df['exang'].unique()
```

Out[16]:
```
array([ 0.,  1., nan])
```

In [17]:
```
df.head()
```

Out[17]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	28	1	2	130.0	132.0	0.0	2.0	185.0	0.0	0.0	NaN	NaN	NaN	0
1	29	1	2	120.0	243.0	0.0	0.0	160.0	0.0	0.0	NaN	NaN	NaN	0
2	29	1	2	140.0	NaN	0.0	0.0	170.0	0.0	0.0	NaN	NaN	NaN	0
3	30	0	1	170.0	237.0	0.0	1.0	170.0	0.0	0.0	NaN	NaN	6.0	0
4	31	0	2	100.0	219.0	0.0	1.0	150.0	0.0	0.0	NaN	NaN	NaN	0

In [18]:
```
df.tail()
```

Out[18]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
289	52	1	4	160.0	331.0	0.0	0.0	94.0	1.0	2.5	NaN	NaN	NaN	1
290	54	0	3	130.0	294.0	0.0	1.0	100.0	1.0	0.0	2.0	NaN	NaN	1
291	56	1	4	155.0	342.0	1.0	0.0	150.0	1.0	3.0	2.0	NaN	NaN	1
292	58	0	2	180.0	393.0	0.0	0.0	110.0	1.0	1.0	2.0	NaN	7.0	1
293	65	1	4	130.0	275.0	0.0	1.0	115.0	1.0	1.0	2.0	NaN	NaN	1

3. Data Cleaning:-

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 14 columns):
age         294 non-null int64
sex         294 non-null int64
cp          294 non-null int64
trestbps    293 non-null float64
chol        271 non-null float64
fbs         286 non-null float64
restecg     293 non-null float64
thalach     293 non-null float64
exang       293 non-null float64
oldpeak     294 non-null float64
slope       104 non-null float64
ca          3 non-null float64
thal        28 non-null float64
num         294 non-null int64
dtypes: float64(10), int64(4)
memory usage: 32.3 KB
```

In [20]:
```
df.shape
```

Out[20]: (294, 14)

In [21]:
```
df.drop(columns=['ca'],inplace=True)
```

In [22]:
```
df.shape
```

Out[22]: (294, 13)

In [23]:
```
df.isnull().sum()
```

Out[23]:
```
age          0
sex          0
cp           0
trestbps     1
chol        23
fbs          8
restecg      1
thalach      1
exang        1
oldpeak      0
slope      190
thal       266
num          0
dtype: int64
```

age,sex,cp,old peak,num dont contain any null value .

In [24]:
```
df.skew()
```

Out[24]:
```
age        -0.284261
sex        -1.010104
cp         -0.218171
trestbps    0.738270
chol        1.433386
fbs         3.390521
restecg     1.983402
thalach    -0.086919
exang       0.857867
oldpeak     1.548824
slope      -1.809518
thal       -0.958747
num         0.583857
dtype: float64
```

In [25]:
```
df['thalach'].fillna(df['thalach'].mean(),inplace=True)
```

In [26]:
```
df.fillna(df.median(),inplace=True)
```

In [27]:
```
df.isnull().sum()
```

Out[27]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
thal        0
num         0
dtype: int64
```

In [28]:
```
df.duplicated().sum()
```

Out[28]:
```
1
```

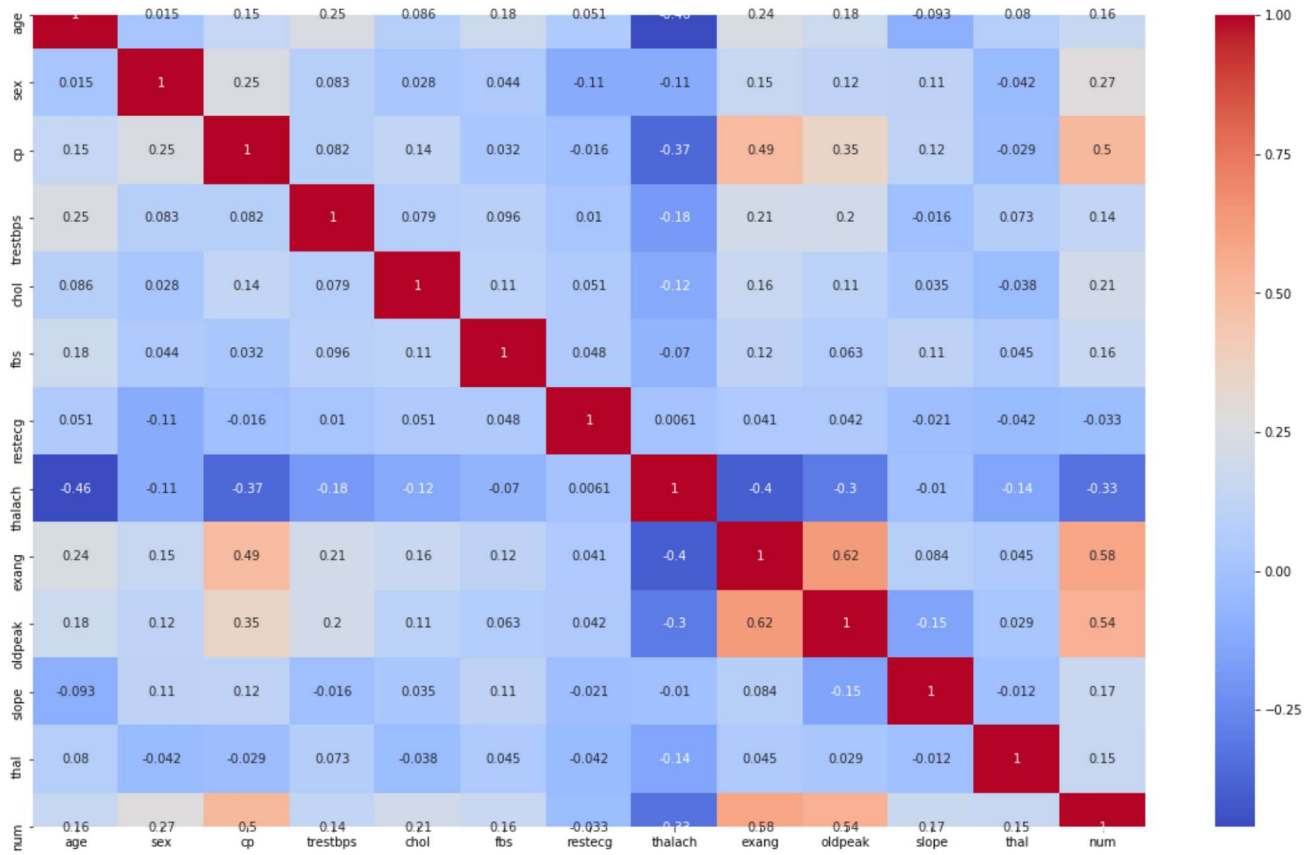
In [29]:
```
df.drop_duplicates(inplace=True)
```

In [30]:
```
df.shape
```

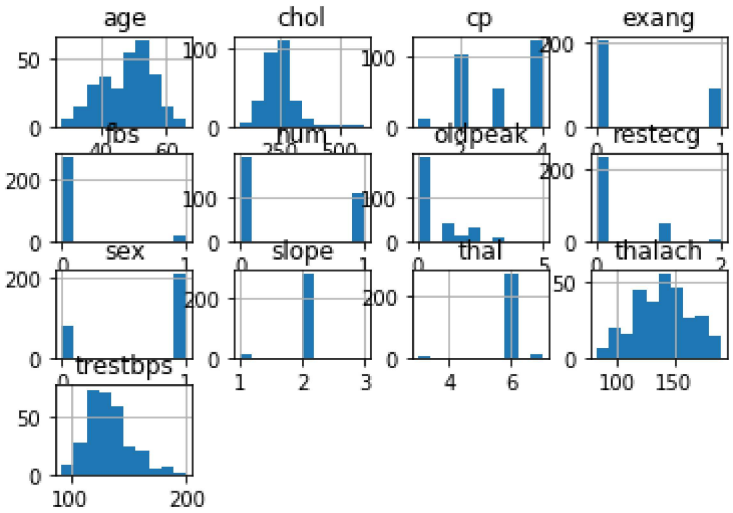
Out[30]:
```
(293, 13)
```

4.Feature Selection:-

In [31]:
```
cor=df.corr()
plt.figure(figsize=(20,12))
sns.heatmap(cor,annot=True,cmap='coolwarm')
plt.show()
```



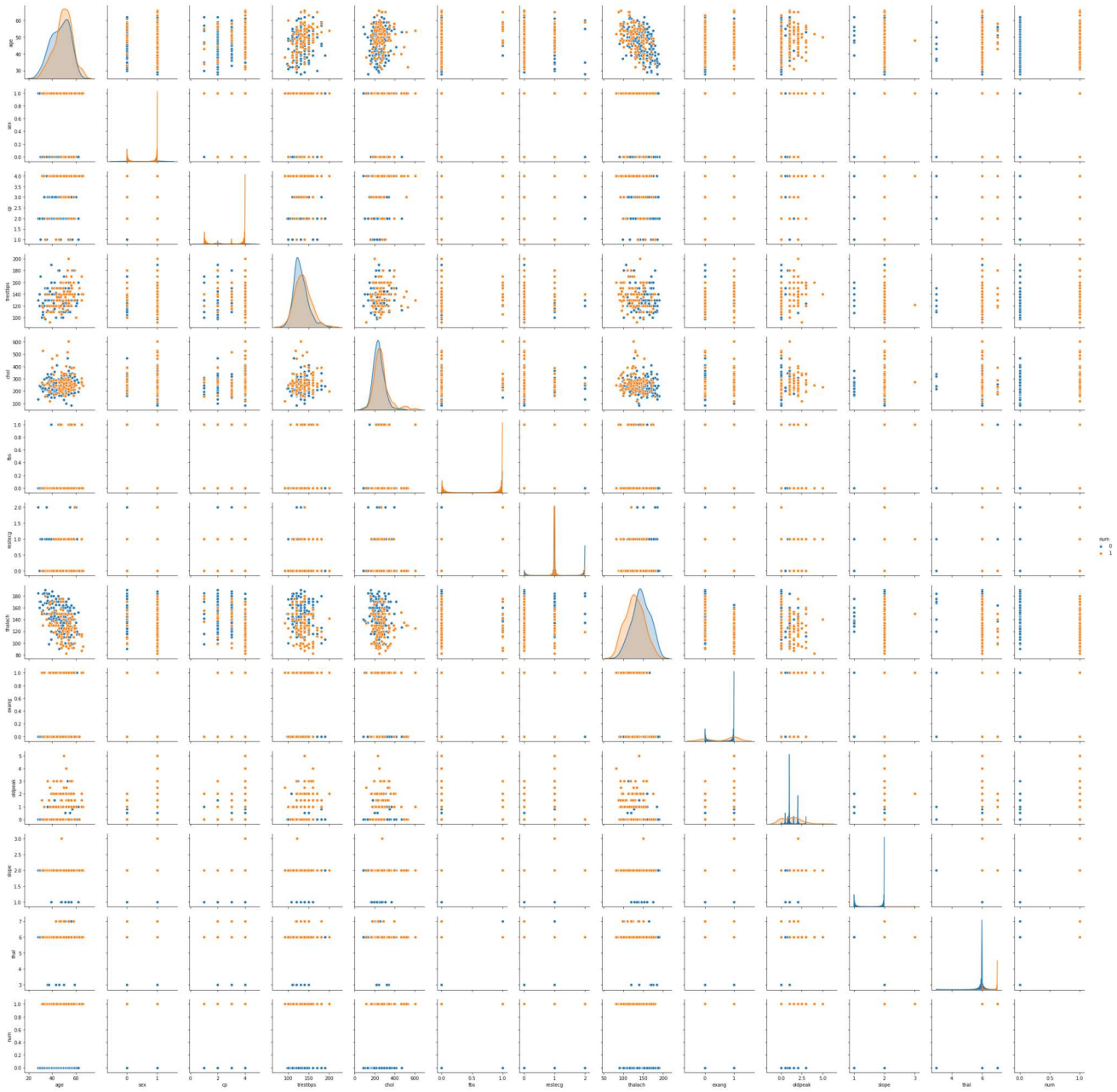
In [32]:
```
df.hist()
plt.figure(figsize=(20,12))
plt.show()
```



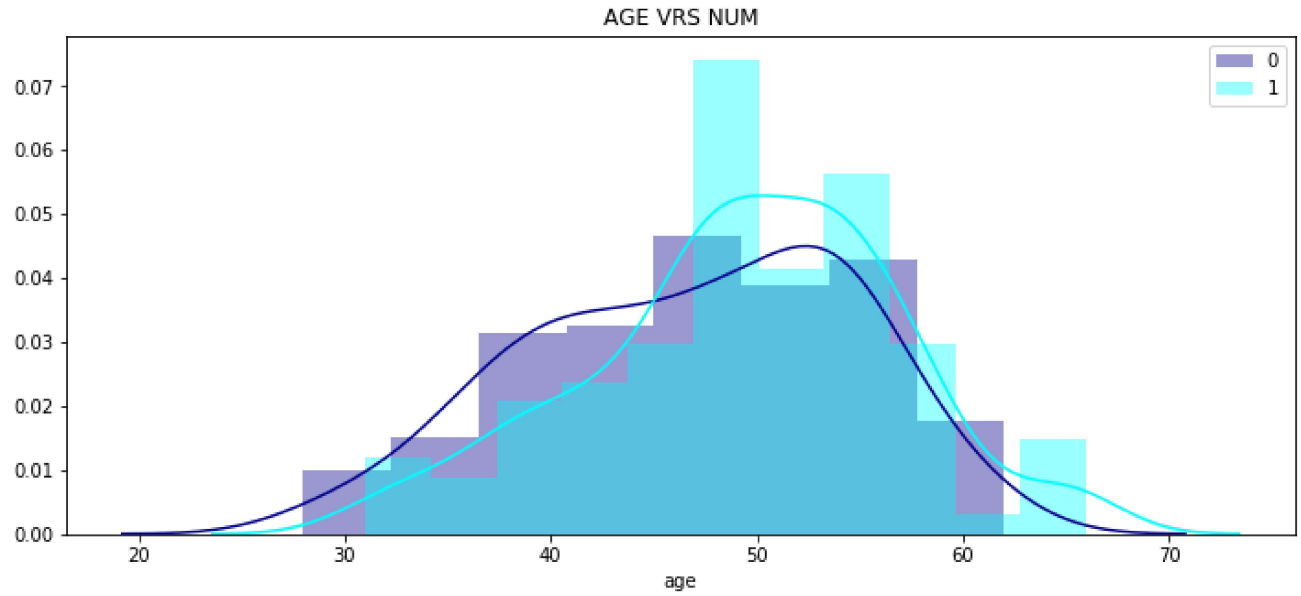

```
<Figure size 1440x864 with 0 Axes>
```

In [33]:
```
sns.pairplot(df,hue='num')
plt.show()
```

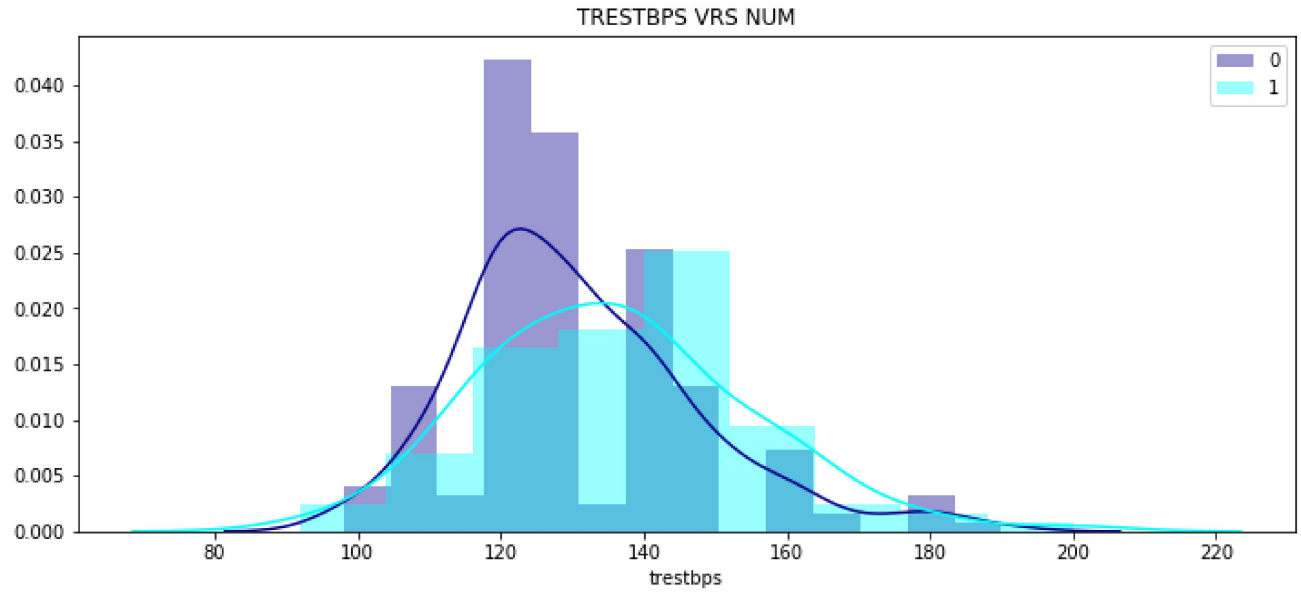
```
C:\Users\HP\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py:487: Runtim
eWarning: invalid value encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\Users\HP\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetools.py:34: Ru
ntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```



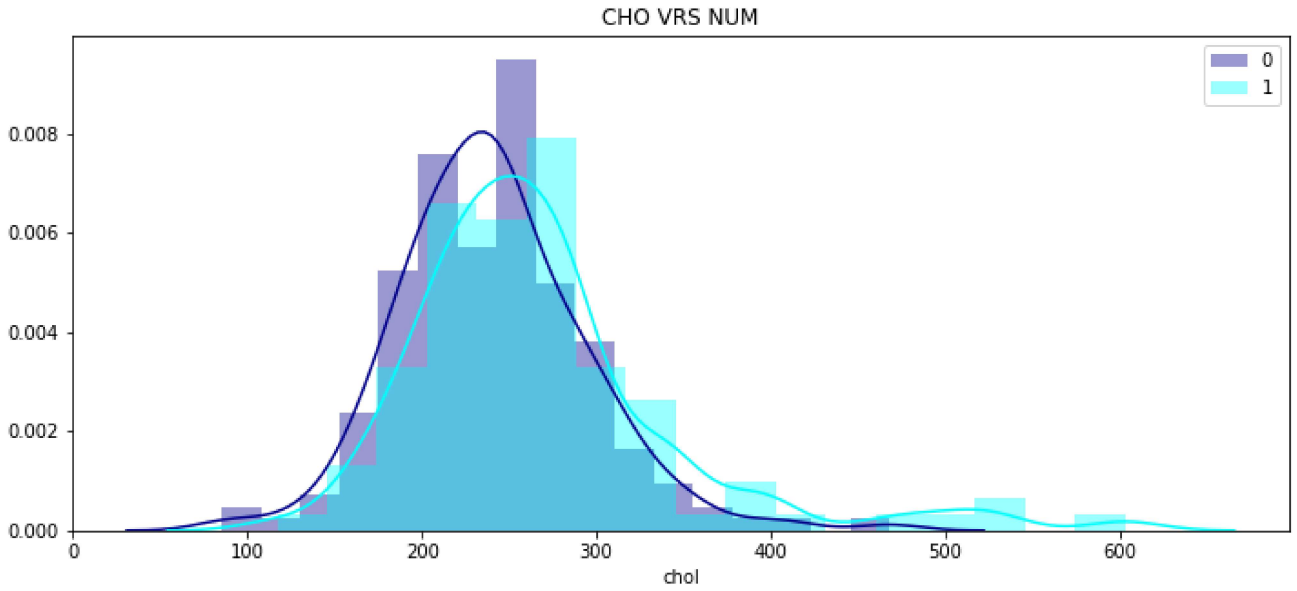
In [34]:
```
plt.figure(figsize=(12,5))
plt.title("AGE VRS NUM")
sns.distplot(df.age[df.num==0],color="darkblue")
sns.distplot(df.age[df.num==1],color="cyan")
plt.legend(['0','1'])
plt.show()
```



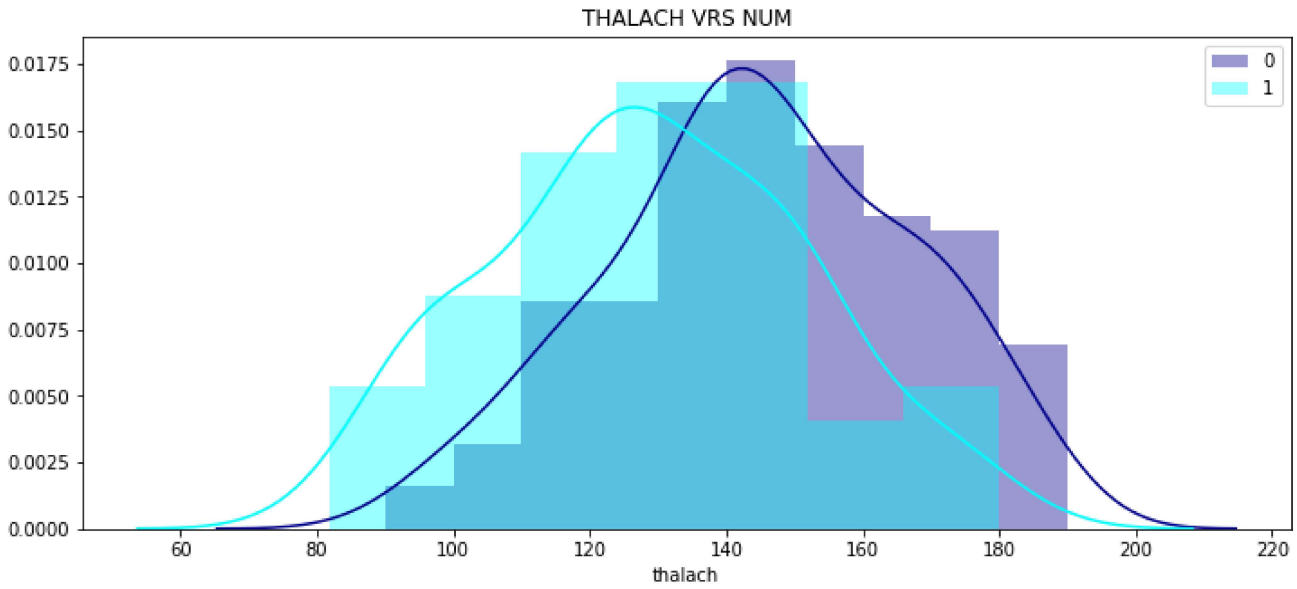
In [35]:
```
plt.figure(figsize=(12,5))
plt.title("TRESTBPS VRS NUM")
sns.distplot(df.trestbps[df.num==0],color="darkblue")
sns.distplot(df.trestbps[df.num==1],color="cyan")
plt.legend(['0','1'])
plt.show()
```



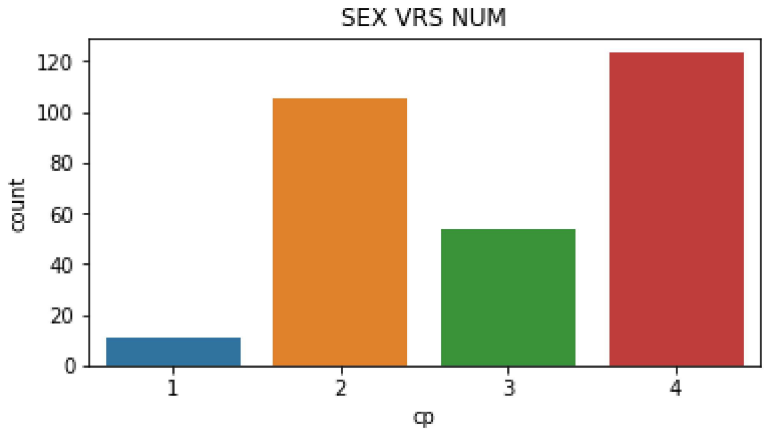
In [36]:
```
plt.figure(figsize=(12,5))
plt.title("CHO VRS NUM")
sns.distplot(df.chol[df.num==0],color="darkblue")
sns.distplot(df.chol[df.num==1],color="cyan")
plt.legend(['0','1'])
plt.show()
```

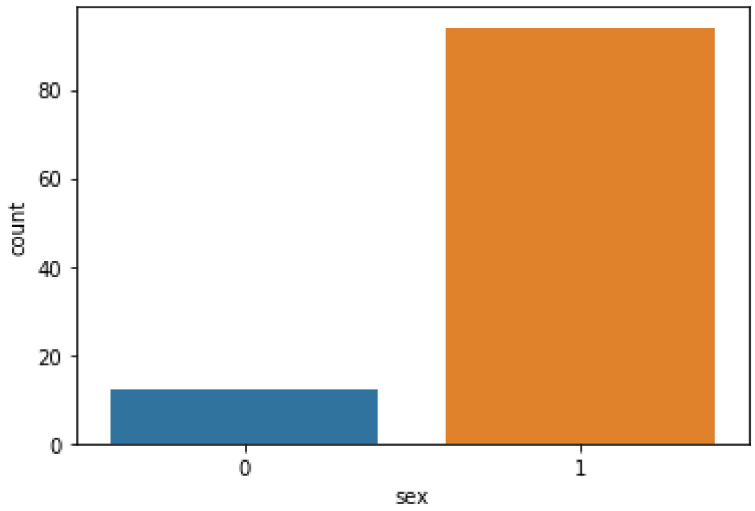


In [37]:
```
plt.figure(figsize=(12,5))
plt.title("THALACH VRS NUM")
sns.distplot(df.thalach[df.num==0],color="darkblue")
sns.distplot(df.thalach[df.num==1],color="cyan")
plt.legend(['0','1'])
plt.show()
```

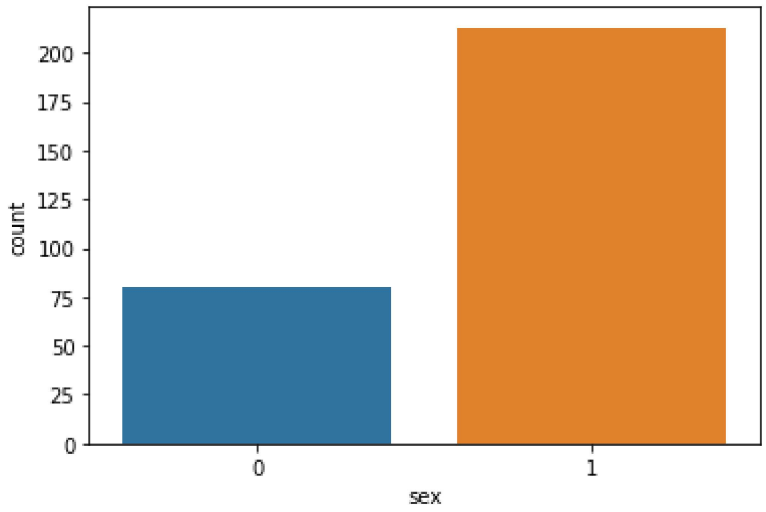


In [38]:
```
plt.figure(figsize=(6,3))
plt.title("SEX VRS NUM")
sns.countplot(df.cp)
plt.show()
sns.countplot(df.sex[df.num==1])
plt.show()
sns.countplot(df.sex)
```

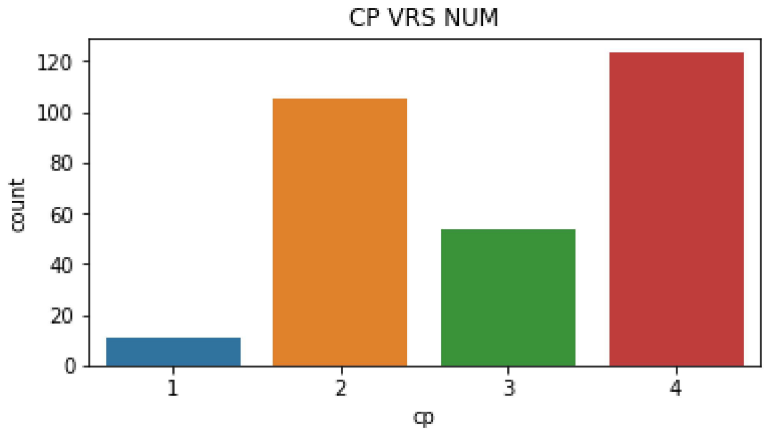


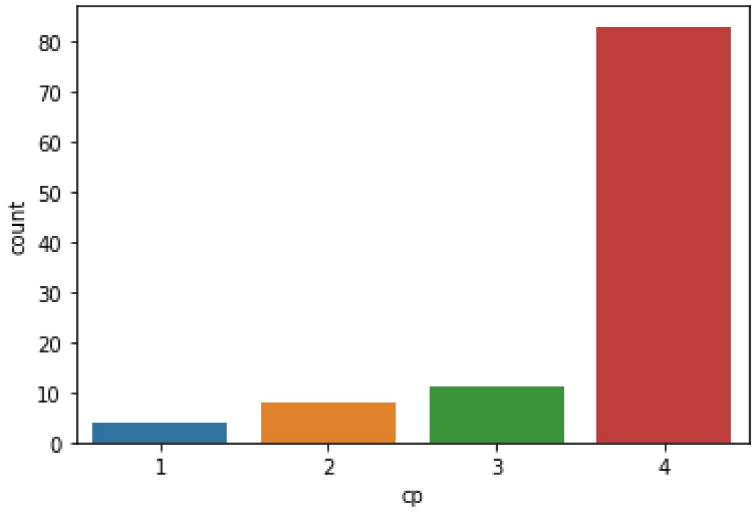


Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x1ae613ab248>

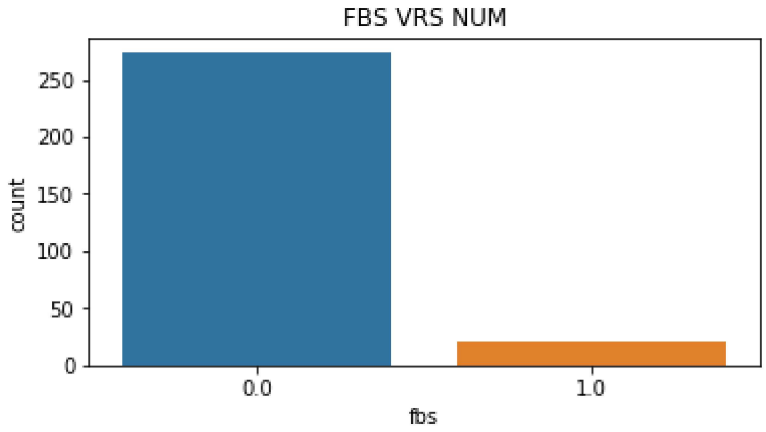


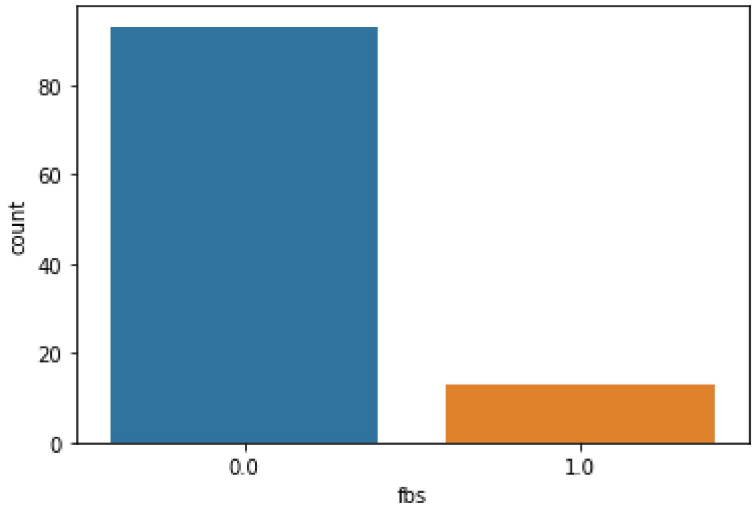
In [39]:
```
plt.figure(figsize=(6,3))
plt.title("CP VRS NUM")
sns.countplot(df.cp)
plt.show()
sns.countplot(df.cp[df.num==1])
plt.show()
```



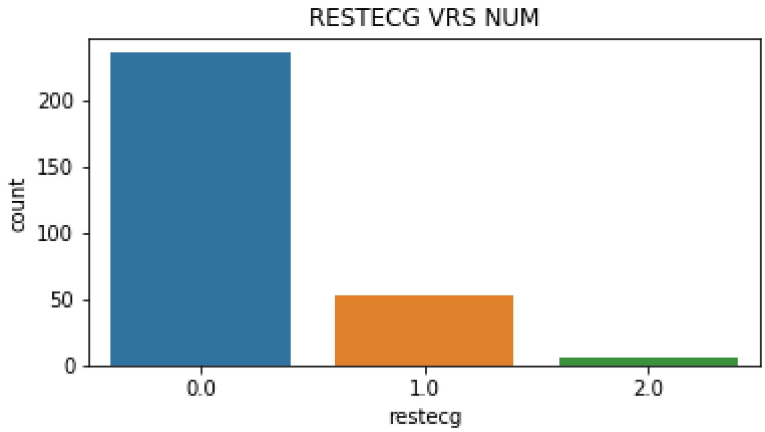


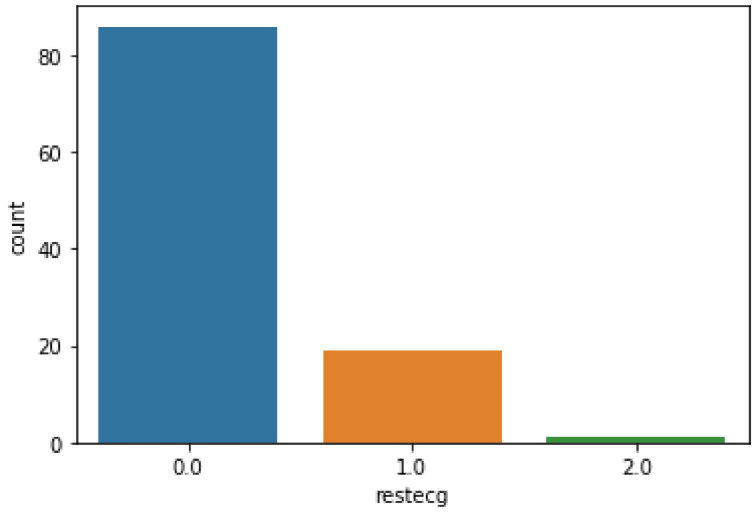
In [40]:
```
plt.figure(figsize=(6,3))
plt.title("FBS VRS NUM")
sns.countplot(df.fbs)
plt.show()
sns.countplot(df.fbs[df.num==1])
plt.show()
```



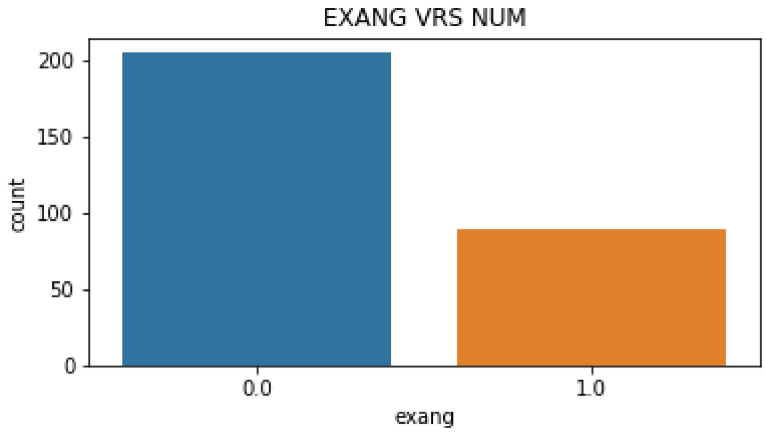


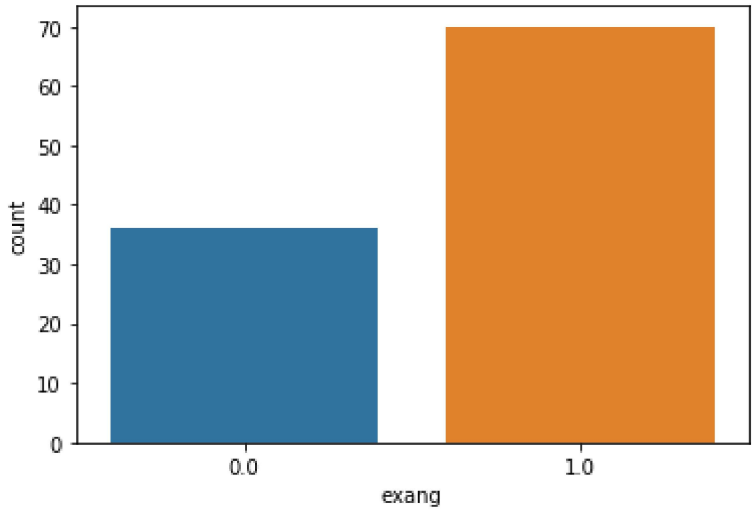
In [41]:
```
plt.figure(figsize=(6,3))
plt.title("RESTECG VRS NUM")
sns.countplot(df.restecg)
plt.show()
sns.countplot(df.restecg[df.num==1])
plt.show()
```



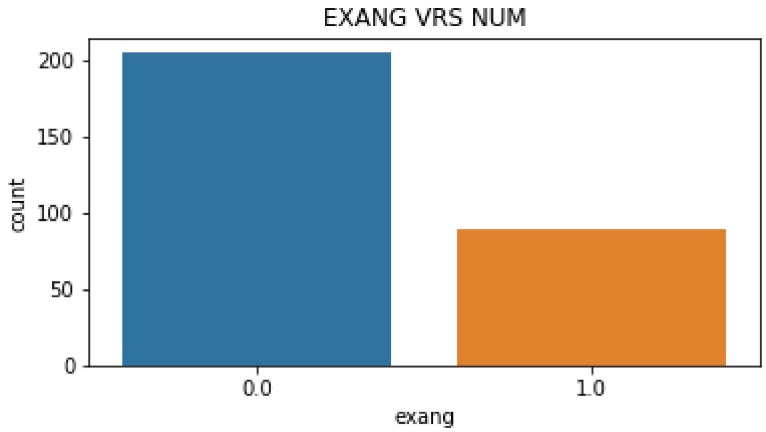


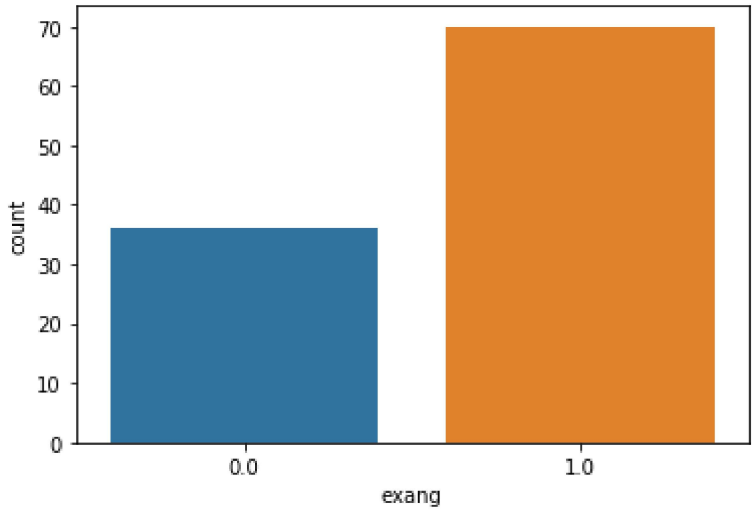
In [42]:
```
plt.figure(figsize=(6,3))
plt.title("EXANG VRS NUM")
sns.countplot(df.exang)
plt.show()
sns.countplot(df.exang[df.num==1])
plt.show()
```



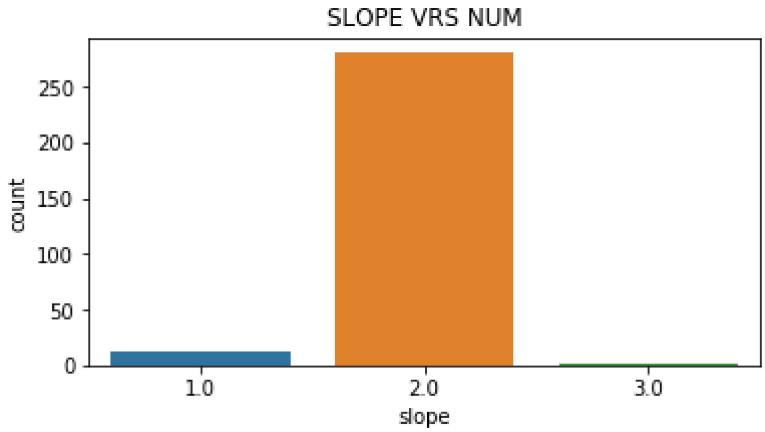


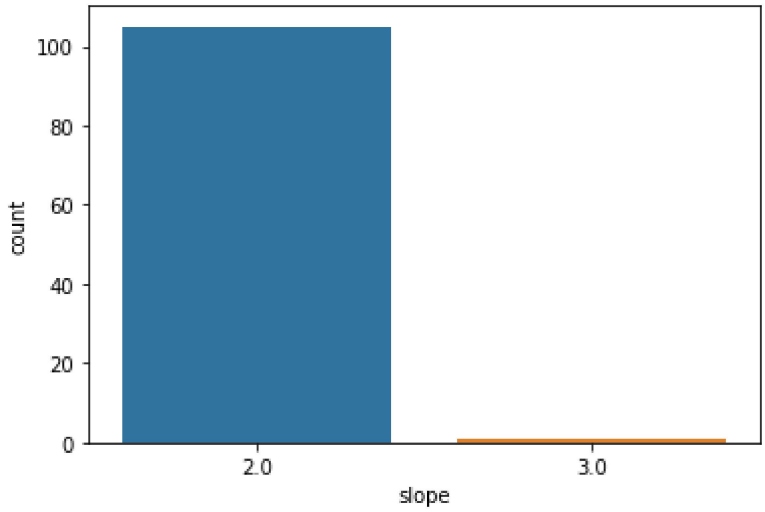
In [43]:
```
plt.figure(figsize=(6,3))
plt.title("EXANG VRS NUM")
sns.countplot(df.exang)
plt.show()
sns.countplot(df.exang[df.num==1])
plt.show()
```



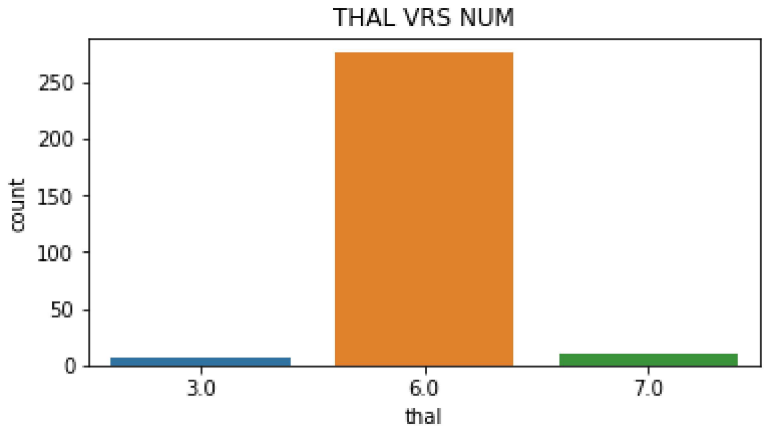


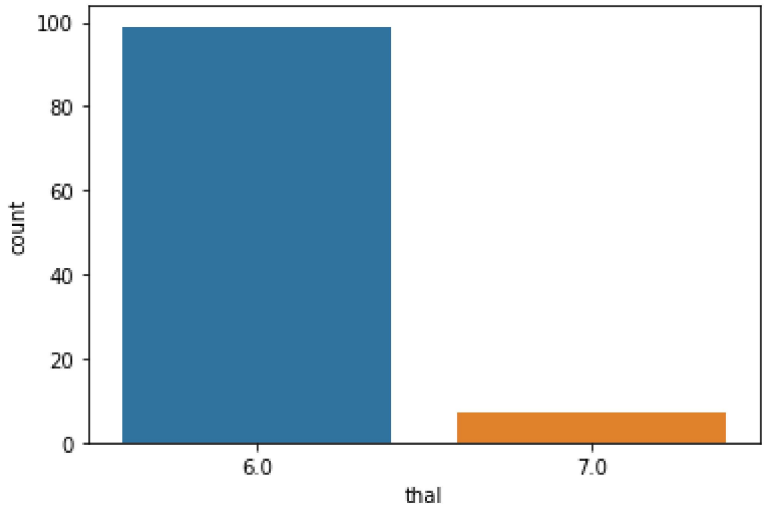
In [44]:
```
plt.figure(figsize=(6,3))
plt.title("SLOPE VRS NUM")
sns.countplot(df.slope)
plt.show()
sns.countplot(df.slope[df.num==1])
plt.show()
```



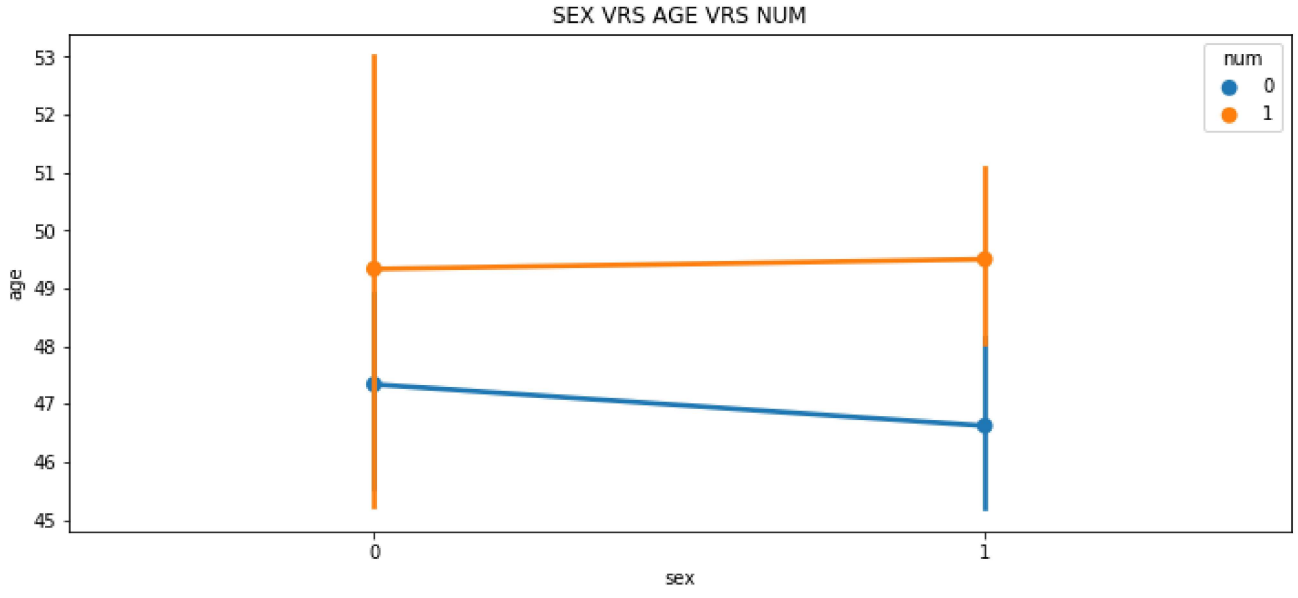


In [45]:
```
plt.figure(figsize=(6,3))
plt.title("THAL VRS NUM")
sns.countplot(df.thal)
plt.show()
sns.countplot(df.thal[df.num==1])
plt.show()
```

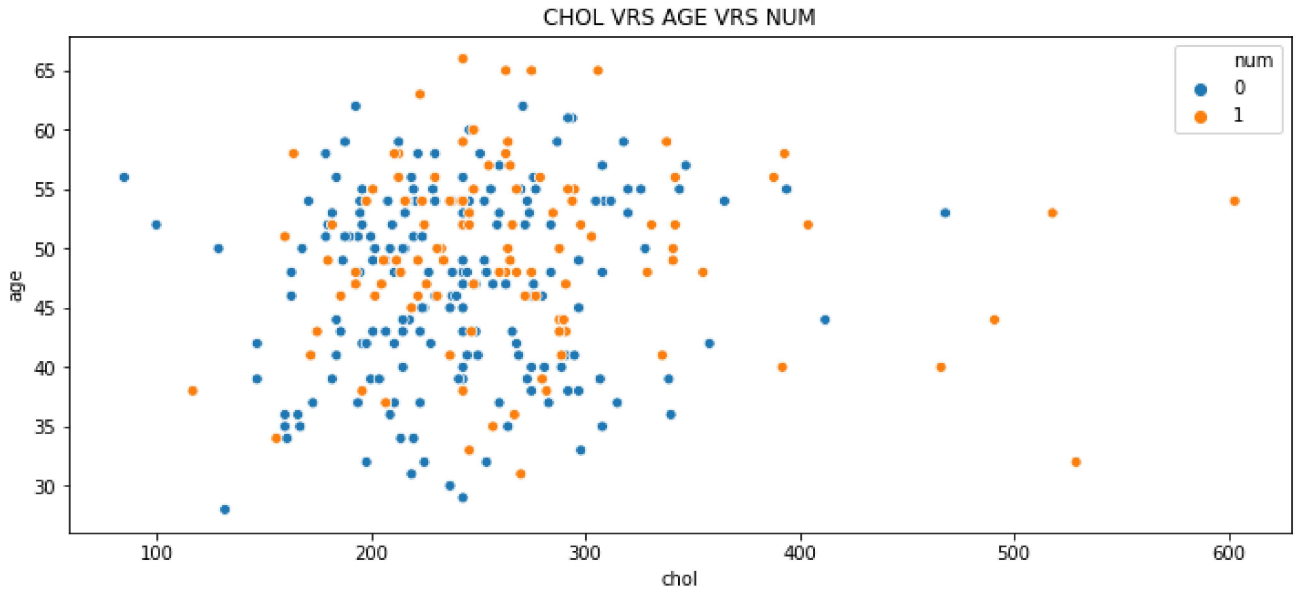




In [46]:
```
plt.figure(figsize=(12,5))
plt.title("SEX VRS AGE VRS NUM")
sns.pointplot(x = 'sex',y='age',hue='num',data=df)
plt.show()
```



In [47]:
```
plt.figure(figsize=(12,5))
plt.title("CHOL VRS AGE VRS NUM")
sns.scatterplot(x='chol',y='age',hue='num',data=df)
plt.show()
```



Split the data into train and test set:-

In [48]:
```
x=df[['age', 'sex', 'cp','trestbps', 'chol', 'fbs','restecg', 'thalach', 'exang', 'oldpeak', 'slope','thal']]
y=df['num']
print(x)
print(y)
```

```
     age  sex  cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
0     28    1   2     130.0  132.0  0.0      2.0    185.0    0.0      0.0
1     29    1   2     120.0  243.0  0.0      0.0    160.0    0.0      0.0
2     29    1   2     140.0  243.0  0.0      0.0    170.0    0.0      0.0
3     30    0   1     170.0  237.0  0.0      1.0    170.0    0.0      0.0
4     31    0   2     100.0  219.0  0.0      1.0    150.0    0.0      0.0
..   ...  ...  ..       ...    ...  ...      ...      ...    ...      ...
289   52    1   4     160.0  331.0  0.0      0.0     94.0    1.0      2.5
290   54    0   3     130.0  294.0  0.0      1.0    100.0    1.0      0.0
291   56    1   4     155.0  342.0  1.0      0.0    150.0    1.0      3.0
292   58    0   2     180.0  393.0  0.0      0.0    110.0    1.0      1.0
293   65    1   4     130.0  275.0  0.0      1.0    115.0    1.0      1.0

     slope  thal
0      2.0   6.0
1      2.0   6.0
2      2.0   6.0
3      2.0   6.0
4      2.0   6.0
..     ...   ...
289    2.0   6.0
290    2.0   6.0
291    2.0   6.0
292    2.0   7.0
293    2.0   6.0

[293 rows x 12 columns]
0      0
1      0
2      0
3      0
4      0
      ..
289    1
290    1
291    1
292    1
293    1
Name: num, Length: 293, dtype: int64
```

In [49]:
```
# split data into train and test
from sklearn.model_selection import train_test_split
xtr,xts,ytr,yts = train_test_split(x,y,test_size=0.2)
# we have to split the data into 80% as train and 20% as test so we have specified test_size as 0.2
print(x.shape)
print(xtr.shape)
print(xts.shape)
print(y.shape)
print(ytr.shape)
print(yts.shape)
```

```
(293, 12)
(234, 12)
(59, 12)
(293,)
(234,)
(59,)
```

SVM

In [50]:
```
from sklearn.svm import SVC
```

In [51]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

In [52]:
```
svc_model=SVC()
```

In [53]:
```
svc_model.fit(xtr,ytr)
```

```
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid thi
s warning.
  "avoid this warning.", FutureWarning)
```

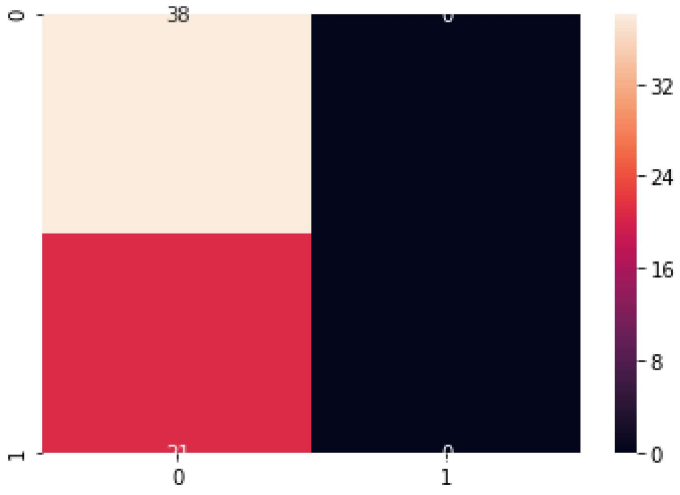
Out[53]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [54]:
```
y_pred=svc_model.predict(xts)
```

In [55]:
```
cm=confusion_matrix(yts,y_pred)
```

In [56]:
```
sns.heatmap(cm,annot=True)
```

Out[56]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ae6337bf88>
```



Normalisation(1):-

In [57]:
```
min_train = xtr.min()
min_train
```

Out[57]:
```
age         28.0
sex          0.0
cp           1.0
trestbps    92.0
chol        85.0
fbs          0.0
restecg      0.0
thalach     82.0
exang        0.0
oldpeak      0.0
slope        1.0
thal         3.0
dtype: float64
```

In [58]:
```
range_train = (xtr - min_train).max()
range_train
```

Out[58]:
```
age          38.0
sex           1.0
cp            3.0
trestbps    108.0
chol        444.0
fbs           1.0
restecg       2.0
thalach     108.0
exang         1.0
oldpeak       5.0
slope         1.0
thal          4.0
dtype: float64
```

In [60]:
```
X_train_scaled = (xtr - min_train)/range_train
X_train_scaled
```

Out[60]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	thal
76	0.447368	1.0	1.000000	0.259259	0.315315	0.0	0.0	0.537037	0.0	0.0	1.0	0.75
115	0.578947	1.0	1.000000	0.537037	0.292793	0.0	0.0	0.537037	1.0	0.0	1.0	0.75
142	0.684211	0.0	0.333333	0.259259	0.423423	0.0	0.0	0.629630	0.0	0.3	1.0	0.75
212	0.578947	0.0	0.666667	0.444444	0.457207	0.0	0.0	0.537037	1.0	0.0	1.0	1.00
210	0.552632	1.0	0.666667	0.212963	0.405405	0.0	0.0	0.861111	0.0	0.0	1.0	0.75
...	...	...	...	...	...	...	...	...	...	...	...	...
34	0.289474	1.0	0.333333	0.351852	0.355856	0.0	0.0	0.351852	0.0	0.0	1.0	0.75
250	0.789474	1.0	1.000000	0.351852	0.400901	0.0	0.0	0.537037	1.0	0.4	1.0	0.75
251	0.342105	1.0	1.000000	0.351852	0.195946	0.0	0.5	0.444444	0.0	0.4	1.0	0.75
242	0.684211	1.0	1.000000	0.351852	0.353604	0.0	0.0	0.083333	1.0	0.2	1.0	0.75
4	0.078947	0.0	0.333333	0.074074	0.301802	0.0	0.5	0.629630	0.0	0.0	1.0	0.75

234 rows × 12 columns

In [61]:
```
min_test = xts.min()
range_test = (xts - min_test).max()
X_test_scaled = (xts - min_test)/range_test
```

In [62]:
```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix

svc_model = SVC()
svc_model.fit(X_train_scaled, ytr)
```

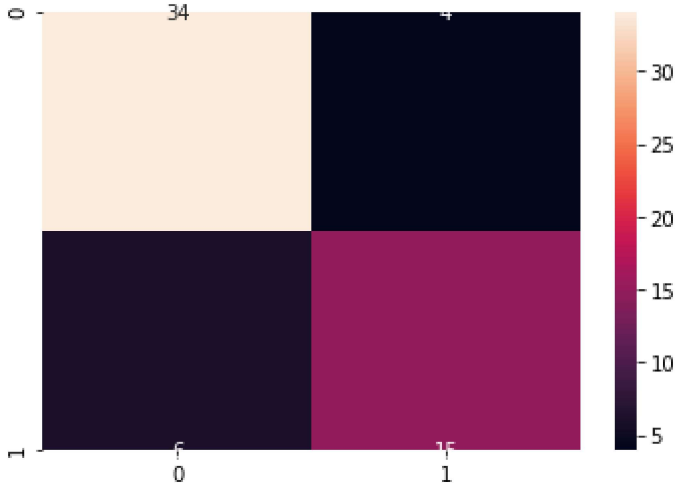
```
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid thi
s warning.
  "avoid this warning.", FutureWarning)
```

Out[62]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [63]:
```
y_predict = svc_model.predict(X_test_scaled)
cm = confusion_matrix(yts, y_predict)

sns.heatmap(cm,annot=True,fmt="d")
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ae63509f08>
```



In [64]:
```
print(classification_report(yts,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.85      0.89      0.87        38
           1       0.79      0.71      0.75        21

    accuracy                           0.83        59
   macro avg       0.82      0.80      0.81        59
weighted avg       0.83      0.83      0.83        59
```

Normalisation(2):

In [65]:
```
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel': ['rb
f']}
```

In [66]:
```
from sklearn.model_selection import GridSearchCV
grid = GridSearchCV(SVC(),param_grid,refit=True,verbose=4)
grid.fit(X_train_scaled,ytr)
```

```
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:1978: Futu
reWarning: The default value of cv will change from 3 to 5 in version 0.22. Specify
it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    0.0s remaining:    0.0s
```

```
Fitting 3 folds for each of 16 candidates, totalling 48 fits
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.759, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.808, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.766, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.641, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.636, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.641, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.636, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.641, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.636, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.810, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.833, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.805, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.810, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.821, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.792, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.633, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.641, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.636, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.633, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.641, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.636, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.709, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.833, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.818, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.823, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.833, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.792, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.810, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.821, total=   0.0s
```

```
[CV] C=10, gamma=0.01, kernel=rbf ....................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.792, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ...................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.633, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ...................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.641, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ...................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.636, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf ......................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.734, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf ......................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.821, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf ......................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.779, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ....................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.797, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ....................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.859, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ....................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.818, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ...................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.835, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ...................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.833, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ...................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.792, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf ..................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.810, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf ..................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.821, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf ..................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.792, total=   0.0s

[Parallel(n_jobs=1)]: Done  48 out of  48 | elapsed:    0.3s finished
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:814: Depr
ecationWarning: The default of the `iid` parameter will change from True to False in
version 0.22 and will be removed in 0.24. This will change numeric results when test
-set sizes are unequal.
  DeprecationWarning)
```

Out[66]:
```
GridSearchCV(cv='warn', error_score='raise-deprecating',
             estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='auto_deprecated', kernel='rbf', max_iter=-1,
                           probability=False, random_state=None, shrinking=True,
                           tol=0.001, verbose=False),
             iid='warn', n_jobs=None,
             param_grid={'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001],
                         'kernel': ['rbf']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=4)
```

In [67]:
```
grid.best_params_
```

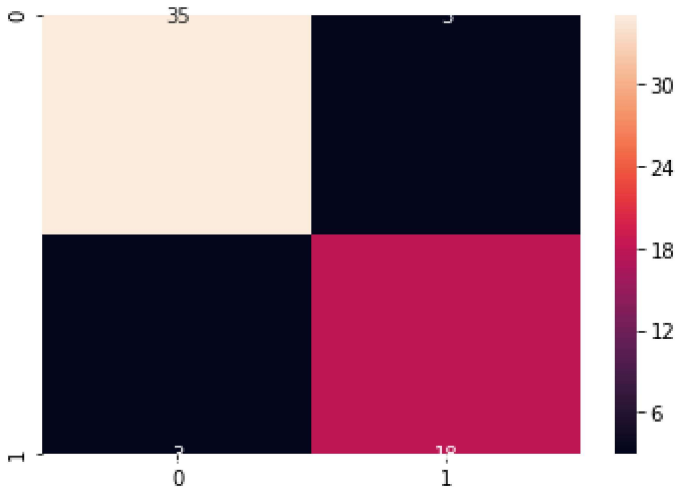
Out[67]:
```
{'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}
```

In [68]:
```
grid.best_estimator_
```

Out[68]:
```
SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

In [69]:
```
grid_predictions = grid.predict(X_test_scaled)
cm = confusion_matrix(yts, grid_predictions)
sns.heatmap(cm, annot=True)
```

Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x1ae6458cbc8>



In [70]:
```
print(classification_report(yts,grid_predictions))
```

```
              precision    recall  f1-score   support

           0       0.92      0.92      0.92        38
           1       0.86      0.86      0.86        21

    accuracy                           0.90        59
   macro avg       0.89      0.89      0.89        59
weighted avg       0.90      0.90      0.90        59
```