In [1]:
```
import pandas as pd
import seaborn as sns
sns.set(color_codes=True)
from sklearn.preprocessing import LabelEncoder
import numpy as np
from pandas_datareader import data
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.datasets import load_boston
from sklearn.model_selection import GridSearchCV
```

In [2]:
```
pip install pandas-datareader
```

```
Requirement already satisfied: pandas-datareader in c:\users\priya\anaconda3\li
b\site-packages (0.10.0)
Requirement already satisfied: pandas>=0.23 in c:\users\priya\anaconda3\lib\sit
e-packages (from pandas-datareader) (1.2.4)
Requirement already satisfied: lxml in c:\users\priya\anaconda3\lib\site-packag
es (from pandas-datareader) (4.6.3)
Requirement already satisfied: requests>=2.19.0 in c:\users\priya\anaconda3\lib
\site-packages (from pandas-datareader) (2.25.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\priya\anacond
a3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2.8.1)
Requirement already satisfied: numpy>=1.16.5 in c:\users\priya\anaconda3\lib\si
te-packages (from pandas>=0.23->pandas-datareader) (1.19.5)
Requirement already satisfied: pytz>=2017.3 in c:\users\priya\anaconda3\lib\sit
e-packages (from pandas>=0.23->pandas-datareader) (2021.1)
Requirement already satisfied: six>=1.5 in c:\users\priya\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7.3->pandas>=0.23->pandas-datareader) (1.15.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\priya\anaconda3\lib\sit
e-packages (from requests>=2.19.0->pandas-datareader) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\priya\anaconda3\l
ib\site-packages (from requests>=2.19.0->pandas-datareader) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\priya\anaconda3\li
b\site-packages (from requests>=2.19.0->pandas-datareader) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\priya\anaconda
3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (1.26.4)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
df = pd.read_csv('heart-statlog_csv.csv', na_values = ['?'])
```

In [4]:
```
df.head()
```

Out[4]:

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electro
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	

In [5]:
```
df.shape
```

Out[5]:
```
(270, 14)
```

In [6]:
```
df.isna().sum()
```

Out[6]:
```
age                                0
sex                                0
chest                              0
resting_blood_pressure             0
serum_cholestoral                  0
fasting_blood_sugar                0
resting_electrocardiographic_results    0
maximum_heart_rate_achieved        0
exercise_induced_angina            0
oldpeak                            0
slope                              0
number_of_major_vessels            0
thal                               0
class                              0
dtype: int64
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   age                                   270 non-null    int64  
 1   sex                                   270 non-null    int64  
 2   chest                                 270 non-null    int64  
 3   resting_blood_pressure                270 non-null    int64  
 4   serum_cholestoral                     270 non-null    int64  
 5   fasting_blood_sugar                   270 non-null    int64  
 6   resting_electrocardiographic_results  270 non-null    int64  
 7   maximum_heart_rate_achieved           270 non-null    int64  
 8   exercise_induced_angina               270 non-null    int64  
 9   oldpeak                               270 non-null    float64
 10  slope                                 270 non-null    int64  
 11  number_of_major_vessels               270 non-null    int64  
 12  thal                                  270 non-null    int64  
 13  class                                 270 non-null    object 
dtypes: float64(1), int64(12), object(1)
memory usage: 29.7+ KB
```

In [8]:
```
le=LabelEncoder()
label=le.fit_transform(df["class"])
```

In [9]:
```
le.classes_
```

Out[9]:
```
array(['absent', 'present'], dtype=object)
```

In [10]:
```
Data=df.drop("class",axis='columns')
```

In [11]:
```
Data.head()
```

Out[11]:

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electro
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	

In [12]:
```
Data["class"]=label
```

In [13]:
```
Data
```

Out[13]:

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_elec
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	
...	...	...	...	...	...	...	
265	52	1	3	172	199	1	
266	44	1	2	120	263	0	
267	56	0	2	140	294	0	
268	57	1	4	140	192	0	
269	67	1	4	160	286	0	

270 rows × 14 columns

In [14]:
```
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

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

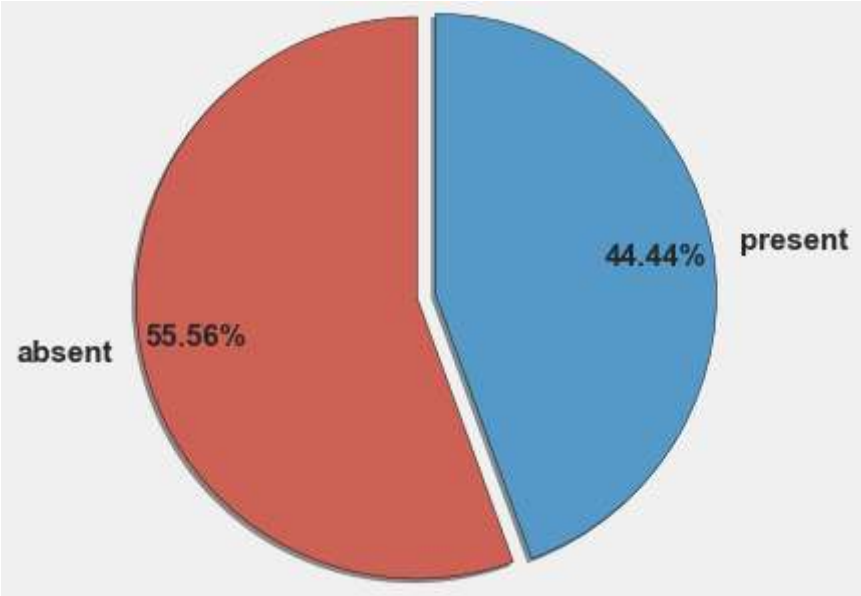
Out[15]:
```
array(['present', 'absent'], dtype=object)
```

In [16]:
```
work_count = df['class'].value_counts().tolist()
```

In [17]:
```
work_label = df['class'].value_counts().index
```

In [18]:
```
colors = ['#CD6155','#5499C7','#AF7AC5','#48C9B0','#52BE80','#F4D03F']
```

In [19]:
```
plt.pie(work_count, labels = work_label,
        autopct = '%1.2f%%',
        colors = colors[:3],
        wedgeprops = {'edgecolor':'k'},
        textprops = {'fontweight':'bold', 'size':15},
        shadow = True,
        explode = [0.1, 0],
        startangle = 90,
        pctdistance = 0.8,
        radius=1.5)
plt.show()
plt.savefig(r'E:\overleaf charts\barchart.png')
```




```
<Figure size 432x288 with 0 Axes>
```

In [20]:
```
df.value_counts()[:20]
```

Out[20]:
```
age  sex  chest  resting_blood_pressure  serum_cholestoral  fasting_blood_sugar  resting_electrocardiographic_results  maximum_heart_rate_achieved  exercise_induced_angina  oldpeak  slope  number_of_major_vessels  thal    class
29   1    2      130                     204                0                    2                                     202                          0                        0.0      1      0                        3       absent  1
59   1    4      170                     326                0                    2                                     140                          1                        3.4      3      0                        7       present 1
58   1    4      128                     259                0                    2                                     130                          1                        3.0      2      2                        7       present 1
                 146                     218                0                    0                                     105                          0                        2.0      2      1                        7       present 1
                 150                     270                0                    2                                     111                          1                        0.8      1      0                        7       present 1
59   0    4      174                     249                0                    0                                     143                          1                        0.0      2      0                        3       present 1
     1    1      160                     273                0                    2                                     125                          0                        0.0      1      0                        3       present 1
                 170                     288                0                    2                                     159                          0                        0.2      2      0                        7       present 1
                 178                     270                0                    2                                     145                          0                        4.2      3      0                        7       absent  1
                  2      140             221                0                    0                                     164                          1                        0.0      1      0                        3       absent  1
                  3      126             218                1                    0                                     134                          0                        2.2      2      1                        6       present 1
                 150                     212                1                    0                                     157                          0                        1.6      1      0                        3       absent  1
                  4      110             239                0                    2                                     142                          1                        1.2      2      1                        7       present 1
                 135                     234                0                    0                                     161                          0                        0.5      2      0                        7       absent  1
                 138                     271                0                    2                                     182                          0                        0.0      1      0                        3       absent  1
                 140                     177                0                    0                                     162                          1                        0.0      1      1                        7       present 1
60   0    1      150                     240                0                    0                                     171                          0                        0.9      1      0                        3       absent  1
```

```
58   1    4      125                  300                  0
2                                  171                        0
0.0      1      2                     7     present    1
60   0    3      102                  318                  0
0                                  160                        0
0.0      1      1                     3     absent     1
                 120                  178                  1
0                                  96                         0
0.0      1      0                     3     absent     1
dtype: int64
```

In [22]:
```
X = df.drop(['class'],axis=1)
y = df['class']
```

In [23]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0
```

In [24]:
```
print('Distribution of traget variable in training set')
print(y_train.value_counts())

print('Distribution of traget variable in test set')
print(y_test.value_counts())
```

```
Distribution of traget variable in training set
absent     120
present     96
Name: class, dtype: int64
Distribution of traget variable in test set
absent     30
present    24
Name: class, dtype: int64
```

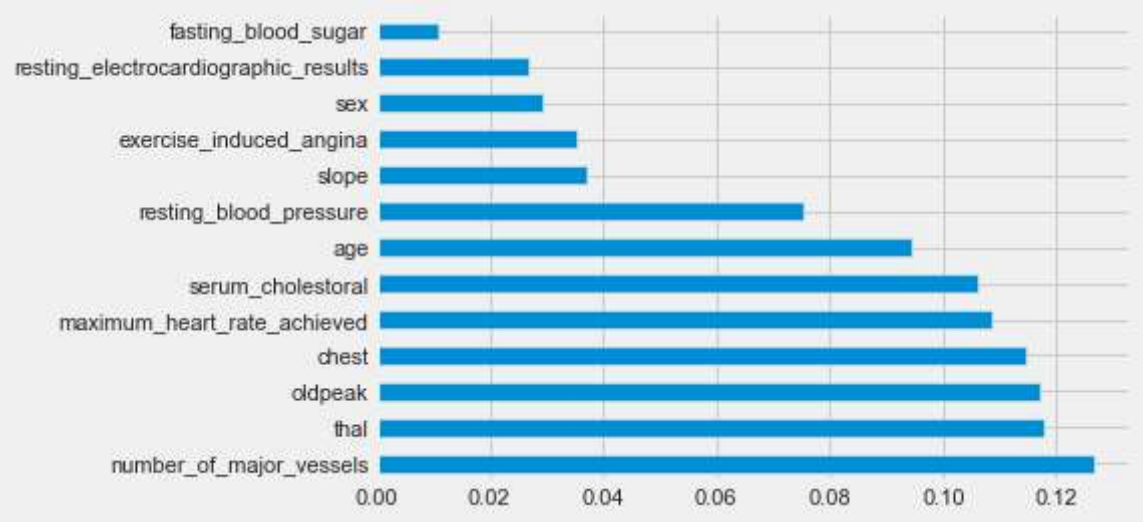
In [25]:
```
print('------------Training Set------------------')
print(X_train.shape)
print(y_train.shape)

print('------------Test Set------------------')
print(X_test.shape)
print(y_test.shape)
```

```
------------Training Set------------------
(216, 13)
(216,)
------------Test Set------------------
(54, 13)
(54,)
```

In [43]:
```
feat_importances = pd.Series(rf_ent.feature_importances_, index=X_train.columns)
feat_importances.nlargest(20).plot(kind='barh')
```

Out[43]: <AxesSubplot:>



In [49]:
```
X = df.iloc[:,0:12]
y = df.iloc[:,12]
```

In [51]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_st
```

In [52]:
```
rt = DecisionTreeRegressor(criterion = 'mse', max_depth=5)
```

In [53]:
```
rt.fit(X_train,y_train)
```

Out[53]: DecisionTreeRegressor(max_depth=5)

In [54]:
```
y_pred = rt.predict(X_test)
r2_score(y_test,y_pred)
```

Out[54]: 0.10740364193337304

In [55]:
```
param_grid = {
    'max_depth':[2,4,8,10,None],
    'criterion':['mse','mae'],
    'max_features':[0.25,0.5,1.0],
    'min_samples_split':[0.25,0.5,1.0]
}
```

In [56]:
```
reg = GridSearchCV(DecisionTreeRegressor(),param_grid=param_grid)
```

In [57]:
```
reg.fit(X_train,y_train)
```

Out[57]:
```
GridSearchCV(estimator=DecisionTreeRegressor(),
             param_grid={'criterion': ['mse', 'mae'],
                         'max_depth': [2, 4, 8, 10, None],
                         'max_features': [0.25, 0.5, 1.0],
                         'min_samples_split': [0.25, 0.5, 1.0]})
```

In [36]:
```
reg.best_score_
```

Out[36]:
```
0.16345729494299172
```

In [38]:
```
reg.best_params_
```

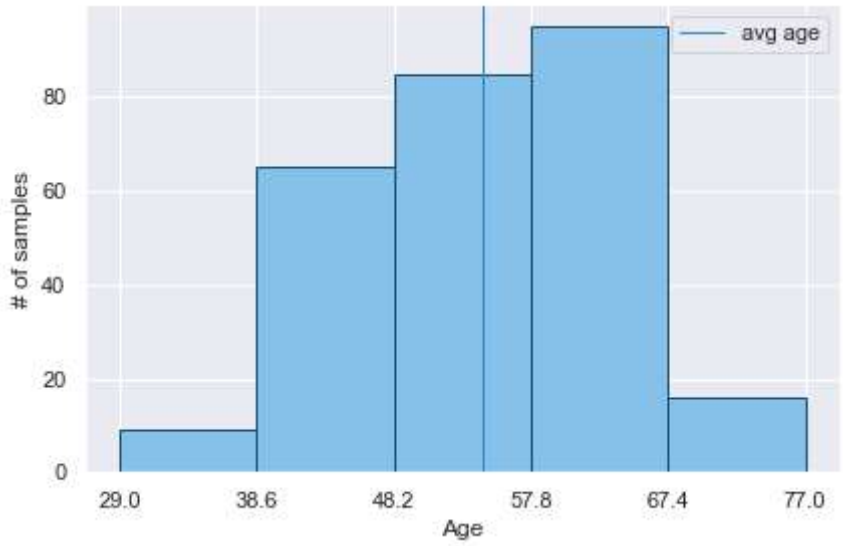
Out[38]:
```
{'criterion': 'mse',
 'max_depth': 2,
 'max_features': 0.25,
 'min_samples_split': 1.0}
```

In [25]:
```
count, bin_edges = np.histogram(df['age'], bins = 5)
```

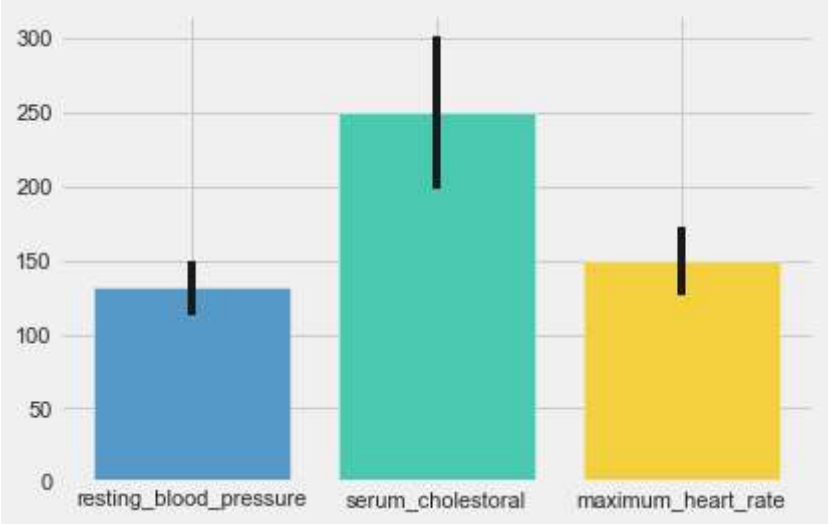
In [ ]:
```

```

In [26]:
```
age_mean = df['age'].mean()
plt.hist(df['age'], color = '#85C1E9', ec = '#1B4F72', bins = 5)
plt.xlabel('Age')
plt.ylabel('# of samples')
plt.xticks(bin_edges)
plt.axvline(age_mean, label = 'avg age',
            color = '#2E86C1', linewidth = 1)
plt.legend()
plt.show()
```



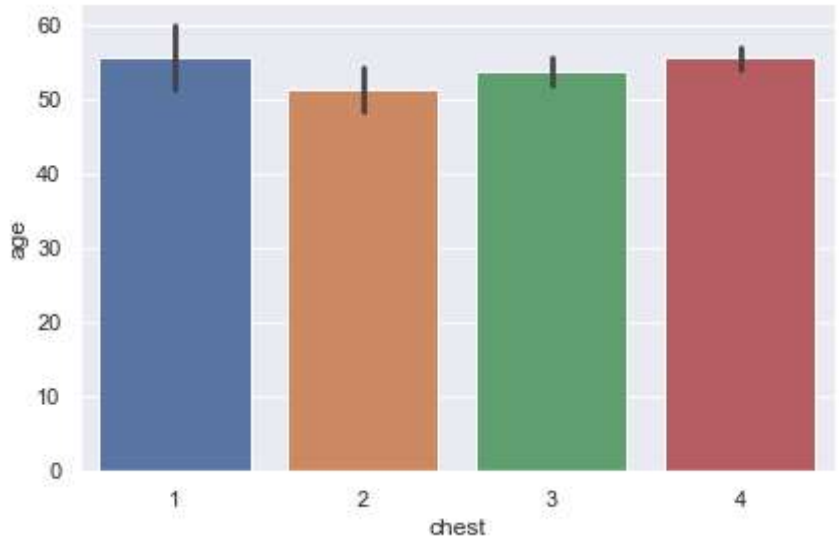
In [20]:
```
plt.bar('resting_blood_pressure', df['resting_blood_pressure'].mean(), yerr = df[
       color = colors[1])
plt.bar('serum_cholestoral', df['serum_cholestoral'].mean(), yerr = df['serum_cho
       color = colors[3])
plt.bar('maximum_heart_rate', df['maximum_heart_rate_achieved'].mean(), yerr = df
       color = colors[5])
plt.show()
```



In [27]:
```
sns.barplot(df['chest'], df['age'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

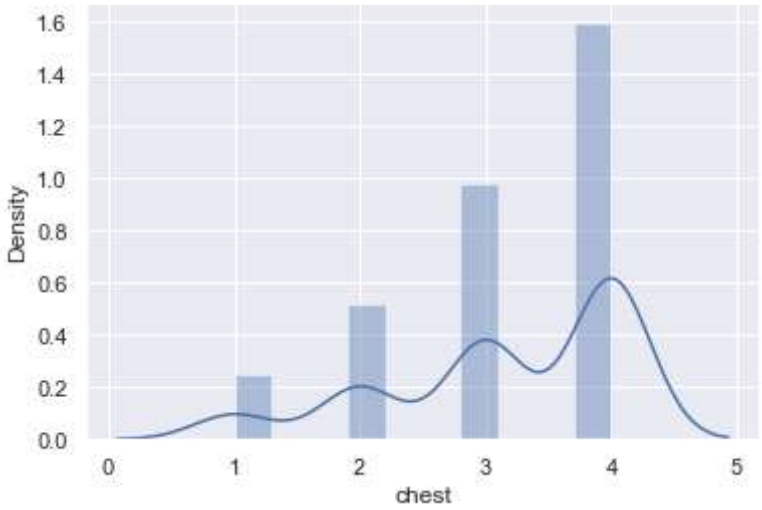
Out[27]:
```
<AxesSubplot:xlabel='chest', ylabel='age'>
```



In [36]:
```
sns.distplot(df['chest'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

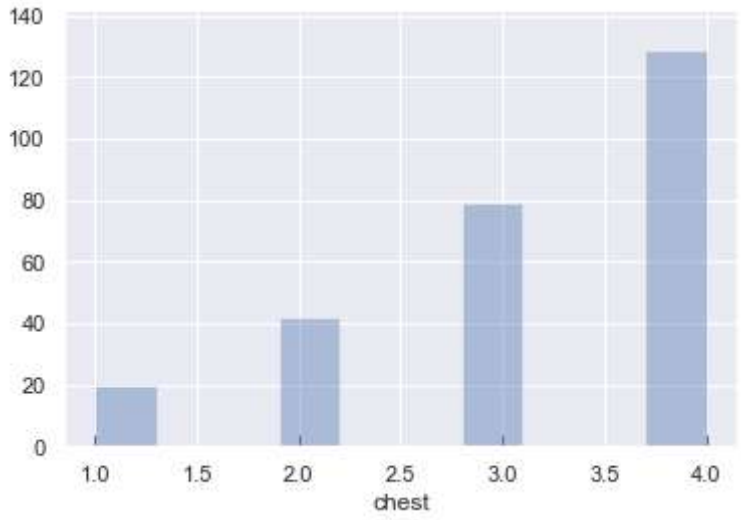
Out[36]: <AxesSubplot:xlabel='chest', ylabel='Density'>



In [37]:
```
sns.distplot(df['chest'], kde=False, rug=True)
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future v
ersion. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\distributions.py:2056: Futur
eWarning: The `axis` variable is no longer used and will be removed. Instead, a
ssign variables directly to `x` or `y`.
  warnings.warn(msg, FutureWarning)
```

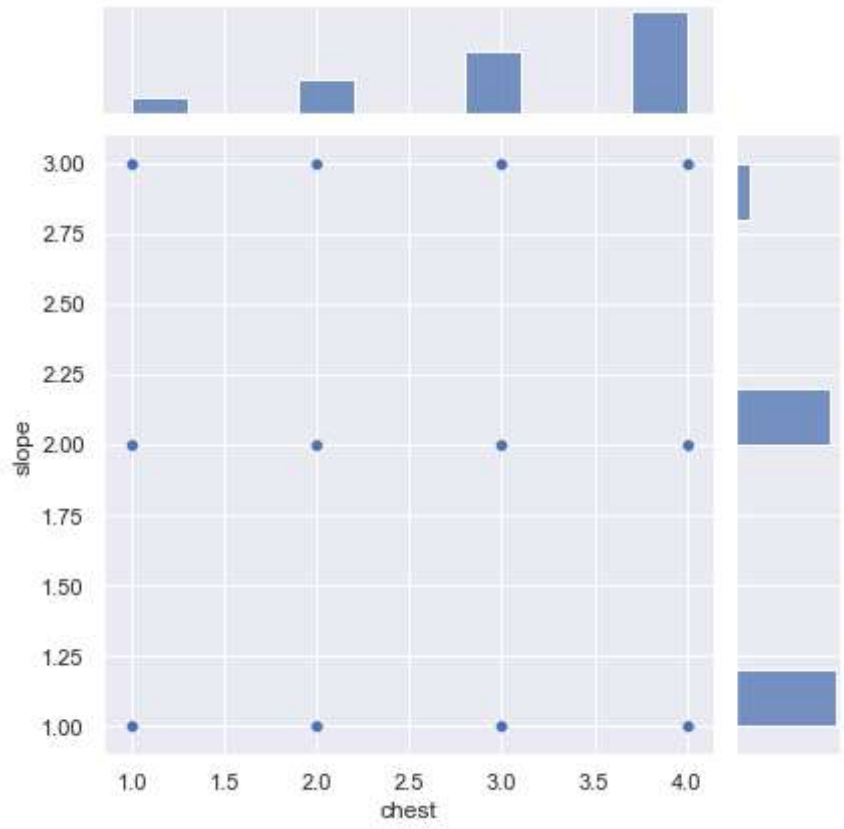
Out[37]: <AxesSubplot:xlabel='chest'>



In [38]:
```
sns.jointplot(df['chest'], df['slope'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

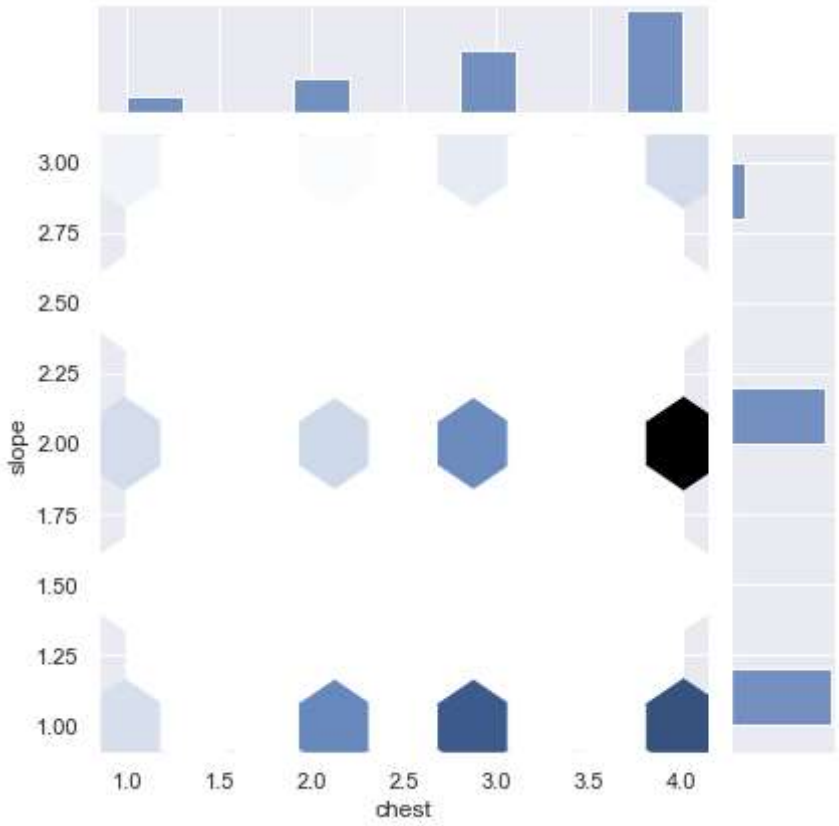
Out[38]: <seaborn.axisgrid.JointGrid at 0x1d86d103160>



In [39]:
```
sns.jointplot(df['chest'], df['slope'], kind="hex")
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

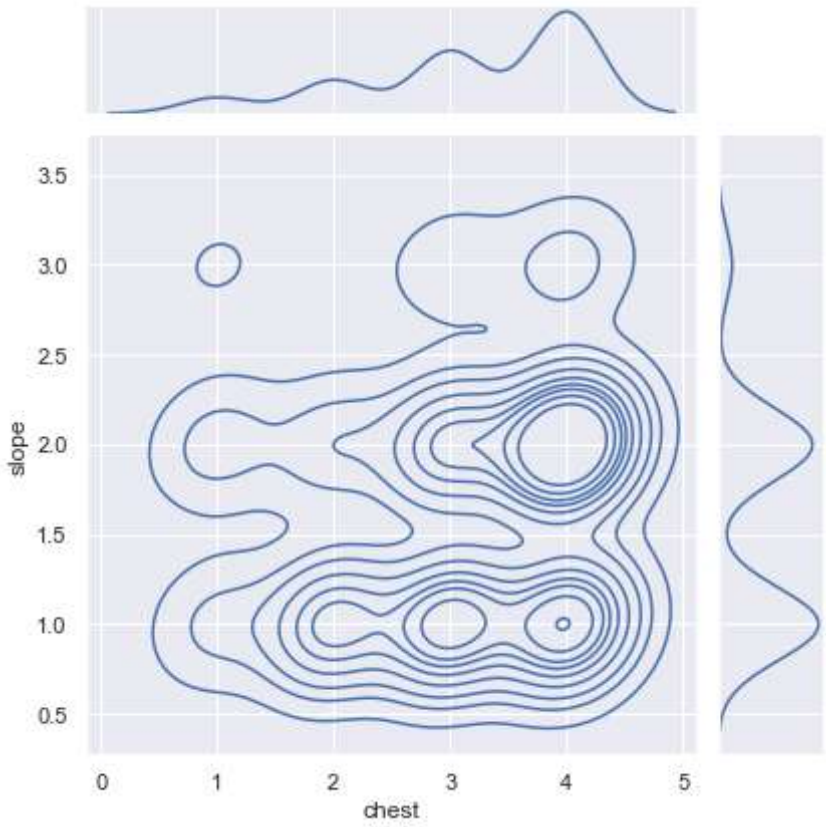
Out[39]:
```
<seaborn.axisgrid.JointGrid at 0x1d86d0223d0>
```



In [40]:
```
sns.jointplot(df['chest'], df['slope'], kind="kde")
```

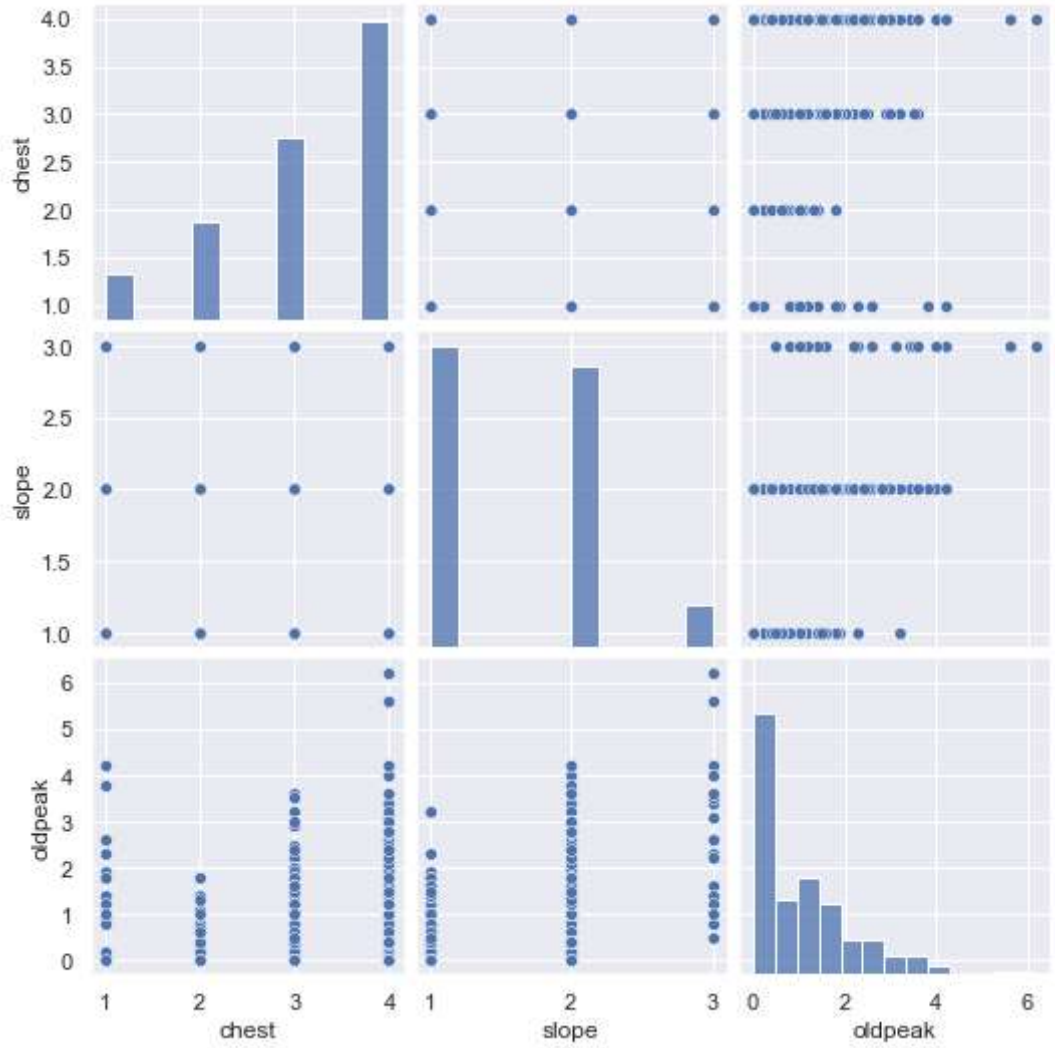
```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[40]: <seaborn.axisgrid.JointGrid at 0x1d86d14aa90>



In [41]:
```
sns.pairplot(df[['chest', 'slope', 'oldpeak']])
```

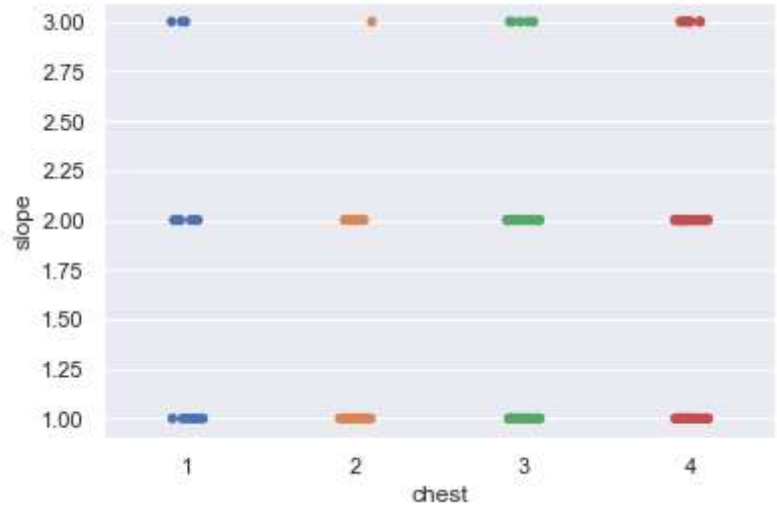
Out[41]: <seaborn.axisgrid.PairGrid at 0x1d86e3d14f0>



In [42]:
```
sns.stripplot(df['chest'], df['slope'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

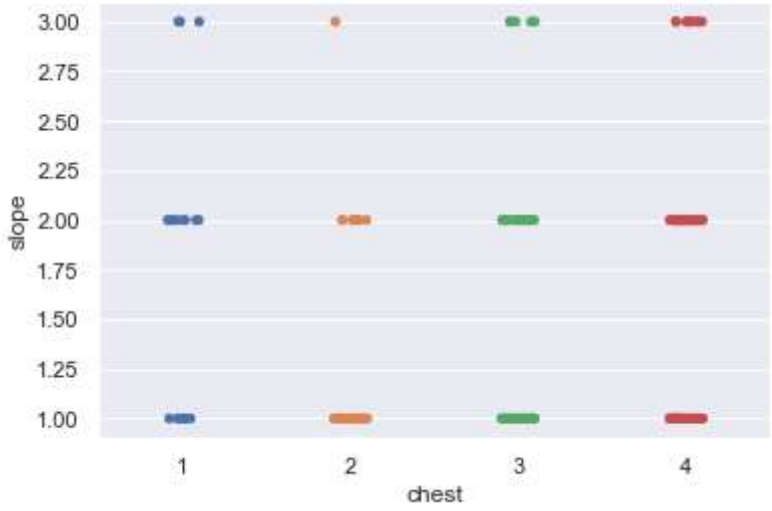
Out[42]: <AxesSubplot:xlabel='chest', ylabel='slope'>



In [43]:
```
sns.stripplot(df['chest'], df['slope'], jitter = True)
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

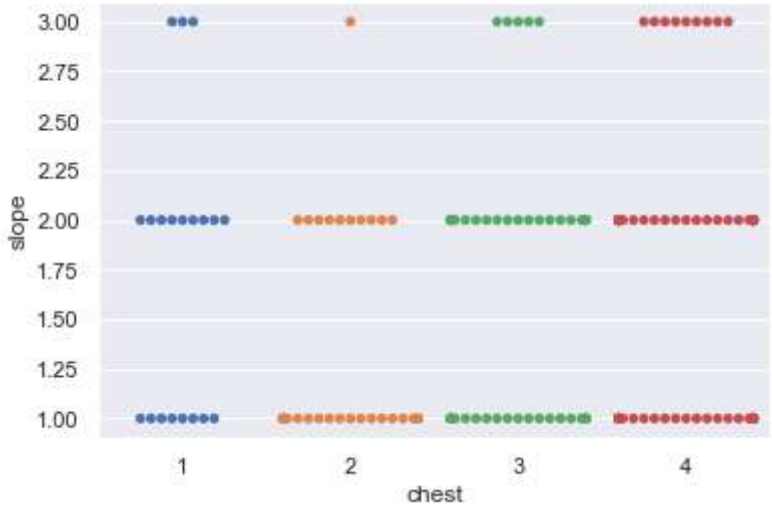
Out[43]: <AxesSubplot:xlabel='chest', ylabel='slope'>



In [44]:
```
sns.swarmplot(df['chest'], df['slope'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWar
ning: 42.9% of the points cannot be placed; you may want to decrease the size o
f the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWar
ning: 60.8% of the points cannot be placed; you may want to decrease the size o
f the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWar
ning: 72.9% of the points cannot be placed; you may want to decrease the size o
f the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

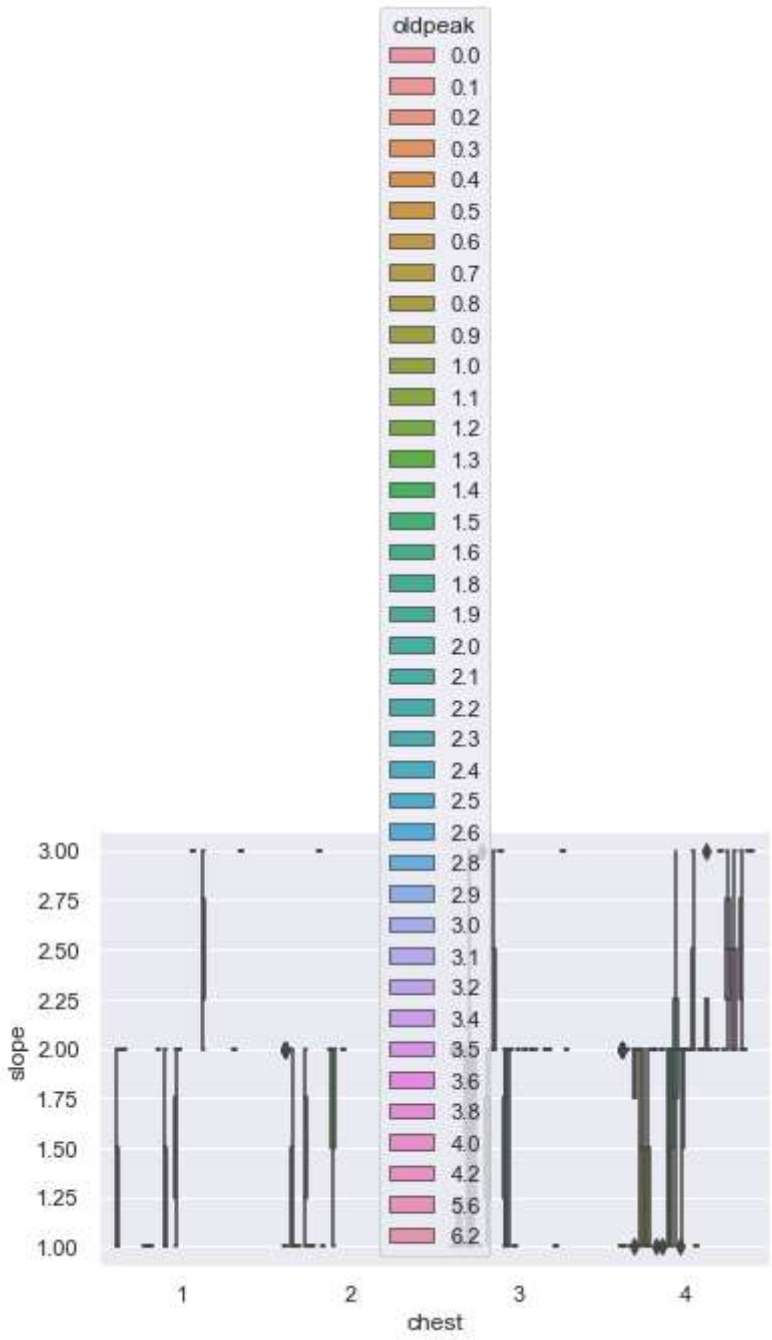
Out[44]: <AxesSubplot:xlabel='chest', ylabel='slope'>



In [45]:
```
sns.boxplot(df['chest'], df['slope'], hue=df['oldpeak'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

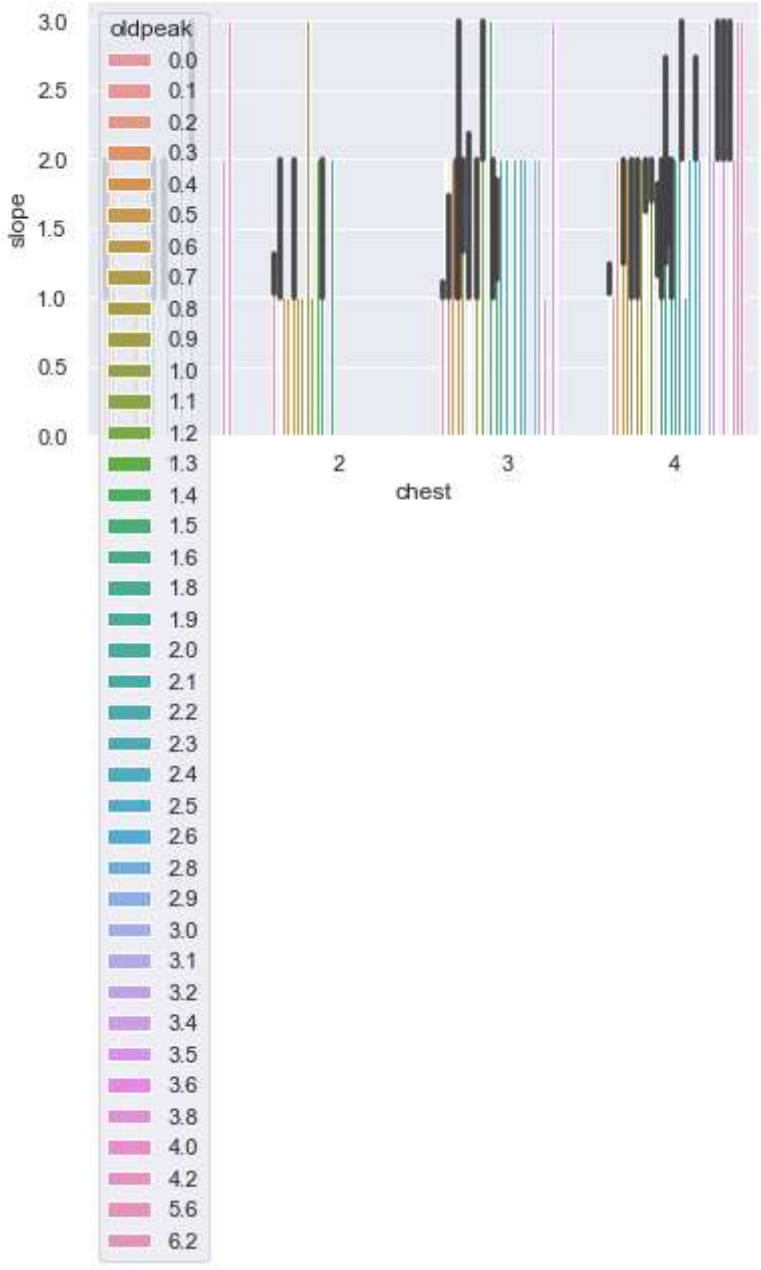
Out[45]: <AxesSubplot:xlabel='chest', ylabel='slope'>



In [46]:
```
sns.barplot(df['chest'], df['slope'], hue=df['oldpeak'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

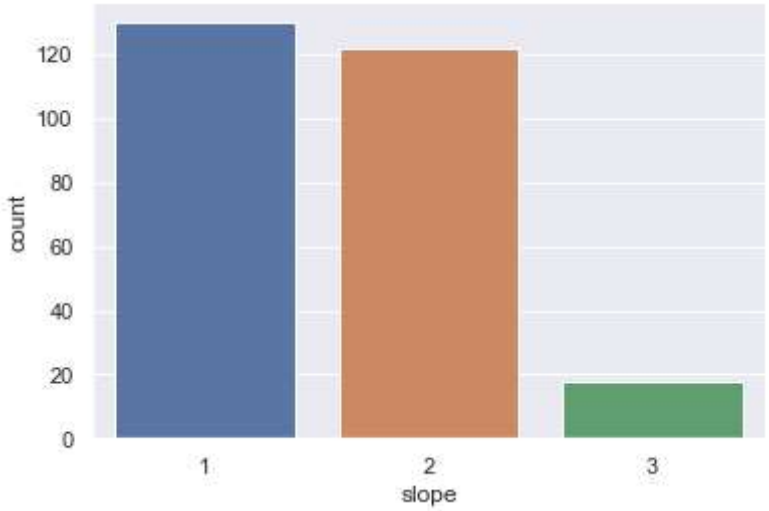
Out[46]: <AxesSubplot:xlabel='chest', ylabel='slope'>



In [47]:
```
sns.countplot(df['slope'])
```

```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

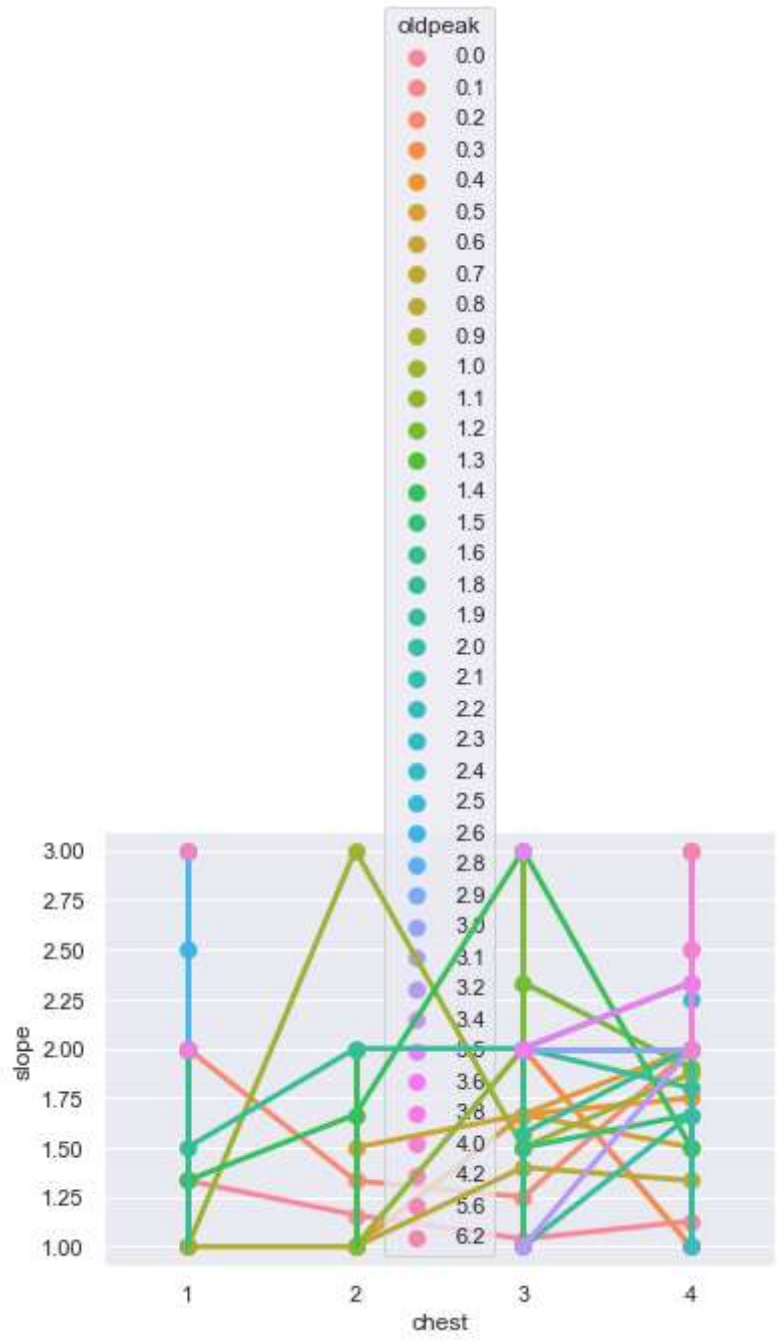
Out[47]: <AxesSubplot:xlabel='slope', ylabel='count'>



In [48]:
```
sns.pointplot(df['chest'], df['slope'], hue=df['oldpeak'])
```

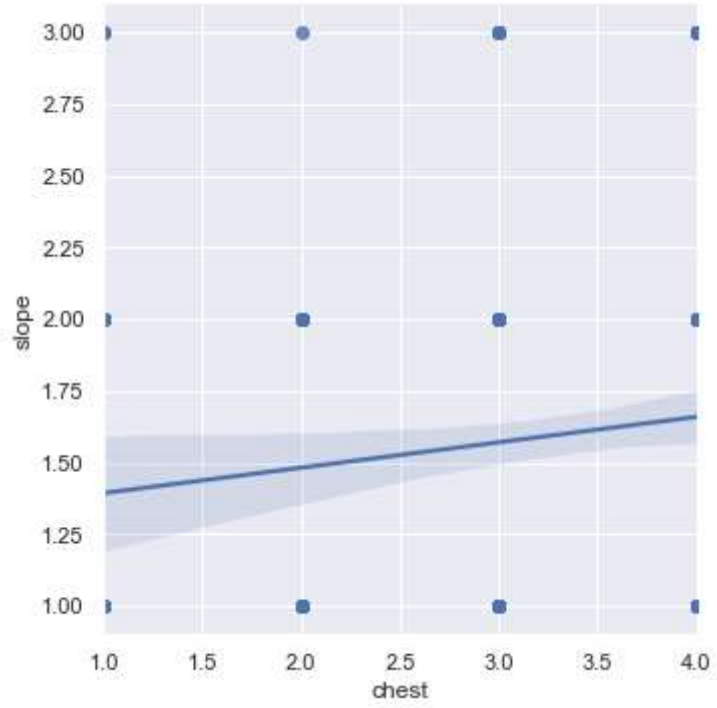
```
C:\Users\Priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[48]: <AxesSubplot:xlabel='chest', ylabel='slope'>



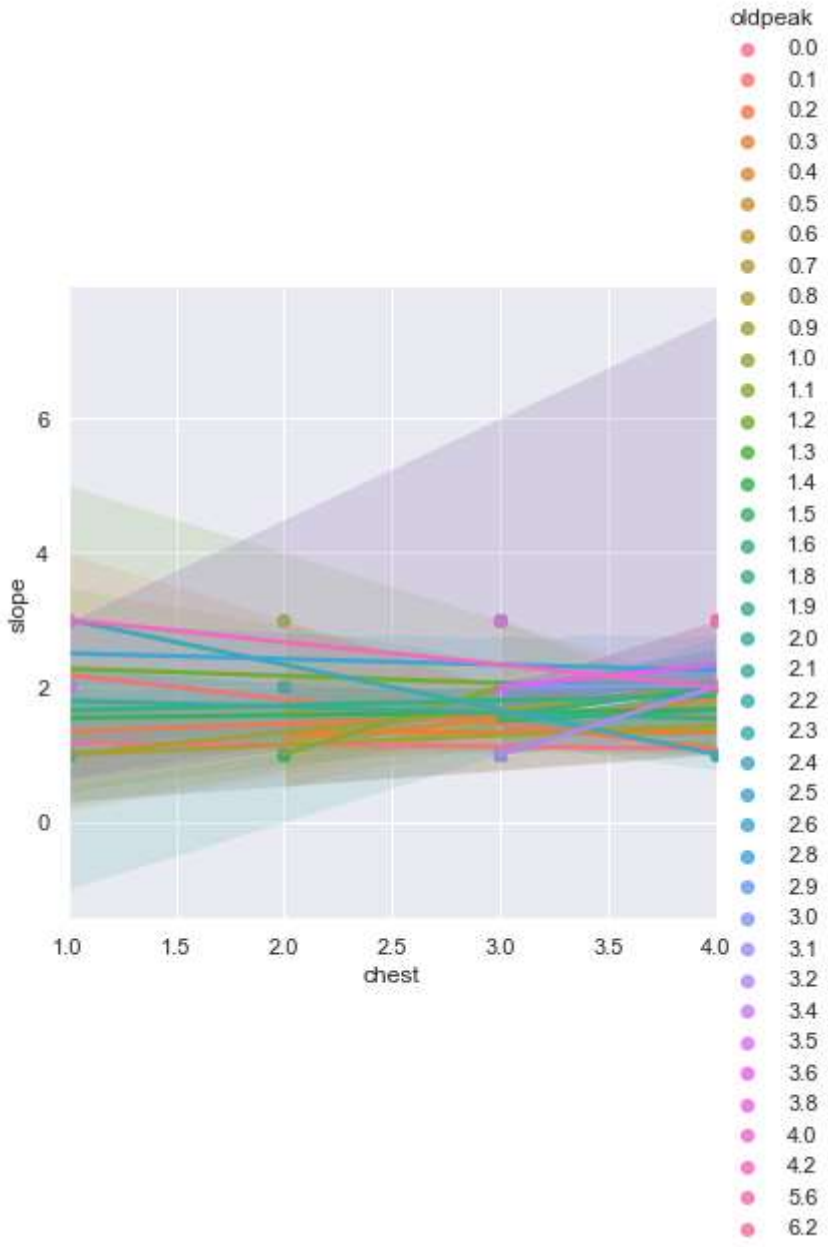
In [49]:
```
sns.lmplot(x="chest", y="slope", data=df)
```

Out[49]: <seaborn.axisgrid.FacetGrid at 0x1d86faca5b0>



In [50]:
```
sns.lmplot(x="chest", y="slope", hue="oldpeak", data=df)
```

Out[50]: <seaborn.axisgrid.FacetGrid at 0x1d86f783e50>



In [ ]:

In [ ]: