Load and explore the dataset

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

In [2]:
```
data = pd.read_csv("C:/Users/Administrator/Desktop/UST campus training/Dataset/Hea
data.head(10)
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0
2	66	0	0	150	226	0	0	114	0	2.6	2	0	0
3	65	1	0	138	282	1	2	174	0	1.4	1	1	0
4	64	1	0	110	211	0	2	144	1	1.8	1	0	0
5	64	1	0	170	227	0	2	155	0	0.6	1	0	2
6	63	1	0	145	233	1	2	150	0	2.3	2	0	1
7	61	1	0	134	234	0	0	145	0	2.6	1	2	0
8	60	0	0	150	240	0	0	171	0	0.9	0	0	0
9	59	1	0	178	270	0	2	145	0	4.2	2	0	2

In [14]:
```
data.columns
```

Out[14]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'condition'],
      dtype='object')
```

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

Out[3]:
```
age            int64
sex            int64
cp             int64
trestbps       int64
chol           int64
fbs            int64
restecg        int64
thalach        int64
exang          int64
oldpeak      float64
slope          int64
ca             int64
thal           int64
condition      int64
dtype: object
```

In [6]:
```
data['condition'].value_counts()
```

Out[6]:
```
condition
0    160
1    137
Name: count, dtype: int64
```

Calculating the number of missing values in the dataset

In [9]:
```
data.isna().sum()
```

Out[9]:
```
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
condition    0
dtype: int64
```

Summary statistics

In [15]:
```
data.describe()
```

Out[15]:

	age	sex	cp	trestbps	chol	fbs	reste
count	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.0000
mean	54.542088	0.676768	2.158249	131.693603	247.350168	0.144781	0.9966
std	9.049736	0.468500	0.964859	17.762806	51.997583	0.352474	0.9949
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.0000
25%	48.000000	0.000000	2.000000	120.000000	211.000000	0.000000	0.0000
50%	56.000000	1.000000	2.000000	130.000000	243.000000	0.000000	1.0000
75%	61.000000	1.000000	3.000000	140.000000	276.000000	0.000000	2.0000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.0000

Gender Distribution Analysis

Count the number of males and females

In [26]:
```
gender = data['sex'].value_counts()
gender.index = ['Male','Female']
gender
```

Out[26]:
```
Male      201
Female     96
Name: count, dtype: int64
```

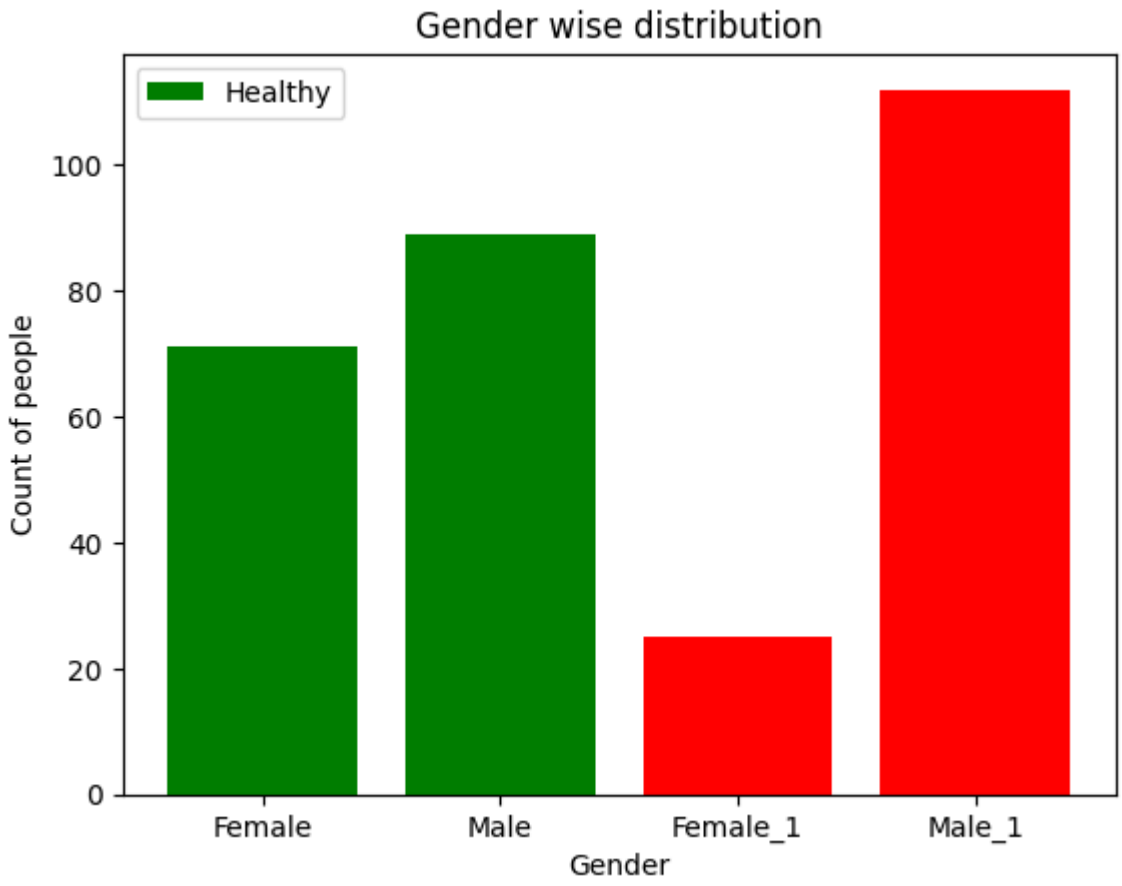
In [25]:
```
```

Out[25]:
```
sex
1    201
0     96
Name: count, dtype: int64
```

In [42]:
```
# Perentage distribution using numpy
gender_dev = data.groupby(['condition','sex'])['sex'].value_counts()
gender_dev
```

Out[42]:
```
condition  sex
0          0       71
           1       89
1          0       25
           1      112
Name: count, dtype: int64
```

In [177]:
```
categories = ['Female','Male','Female_1','Male_1']
plt.bar(categories,gender_dev.values , color = ['green','green','red','red'])
plt.title('Gender wise distribution')
plt.xlabel('Gender')
plt.legend(['Healthy','Has disease'])
plt.ylabel('Count of people')
plt.show()
```



Age analysis

In [50]:
```
min_age = data['age'].min()
max_age = data['age'].max()
avg_age = data['age'].mean()
median_age = data['age'].median()
print(f'Minimum age in dataset is : {min_age}')
print(f'Maximum age in dataset is : {max_age}')
print(f'Average age in dataset is : {round(avg_age)}')
print(f'Median age in dataset is : {median_age}')
```

```
Minimum age in dataset is : 29
Maximum age in dataset is : 77
Average age in dataset is : 55
Median age in dataset is : 56.0
```

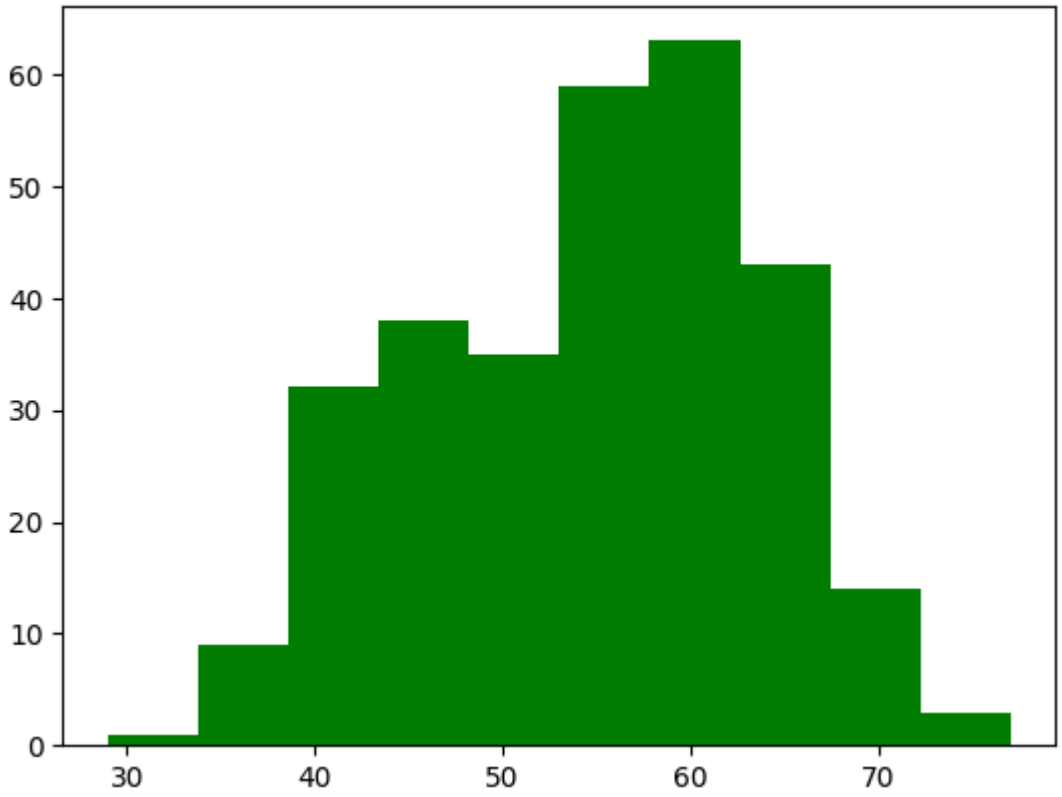
In [59]:
```
min_age = data[data['condition']==1]['age'].min()
max_age = data[data['condition']==1]['age'].max()
avg_age = data[data['condition']==1]['age'].mean()
median_age = data[data['condition']==1]['age'].median()

print(f" Min age of patient with disease is {min_age}")
print(f" Maximum age of patient with disease is {max_age}")
print(f" Average age of patient with disease is {round(avg_age)}")
print(f" Median age of patient with disease is {median_age}")
```

```
Min age of patient with disease is 35
Maximum age of patient with disease is 77
Average age of patient with disease is 57
Median age of patient with disease is 58.0
```

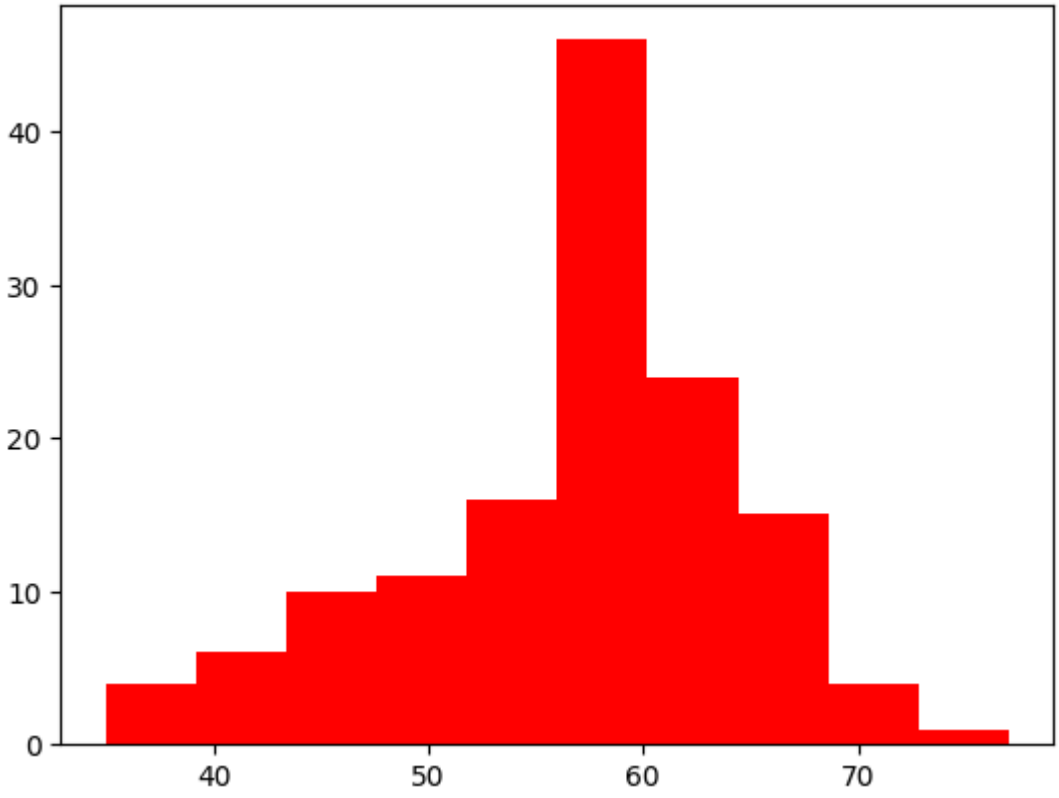
In [61]:
```
plt.hist(data['age'], color='green', bins=10)
```

Out[61]:
```
(array([ 1.,  9., 32., 38., 35., 59., 63., 43., 14.,  3.]),
 array([29. , 33.8, 38.6, 43.4, 48.2, 53. , 57.8, 62.6, 67.4, 72.2, 77. ]),
 <BarContainer object of 10 artists>)
```



In [62]:
```
plt.hist(data[data['condition']==1]['age'], bins = 10, color = 'red')
```

Out[62]:
```
(array([ 4.,  6., 10., 11., 16., 46., 24., 15.,  4.,  1.]),
 array([35. , 39.2, 43.4, 47.6, 51.8, 56. , 60.2, 64.4, 68.6, 72.8, 77. ]),
 <BarContainer object of 10 artists>)
```



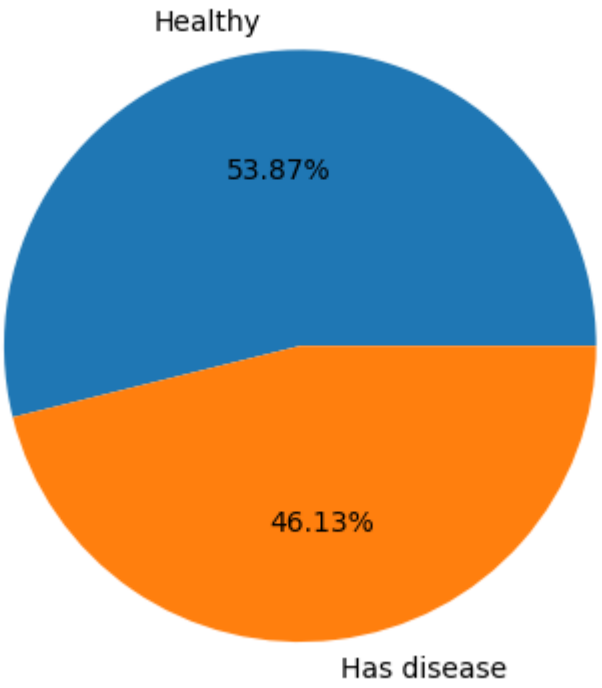
4. Target Variable Analysis

In [65]:
```
patients_with_without = data['condition'].value_counts()
patients_with_without
```

Out[65]:
```
condition
0    160
1    137
Name: count, dtype: int64
```

In [71]:
```
plt.pie(patients_with_without, labels=['Healthy','Has disease'],autopct='%2.2f%%')
```

Out[71]:
```
([<matplotlib.patches.Wedge at 0x2da4cd87050>,
  <matplotlib.patches.Wedge at 0x2da4cd86bd0>],
 [Text(-0.13347885143430296, 1.0918715108563732, 'Healthy'),
  Text(0.13347885143430332, -1.0918715108563732, 'Has disease')],
 [Text(-0.07280664623689252, 0.5955662786489309, '53.87%'),
  Text(0.07280664623689272, -0.5955662786489309, '46.13%')])
```



In [13]:
```
disease_percentage = (len(data[data['condition']==1])/len(data))*100
disease_percentage
```

Out[13]: 46.12794612794613

In [79]:
```
data.corr()
```

Out[79]:

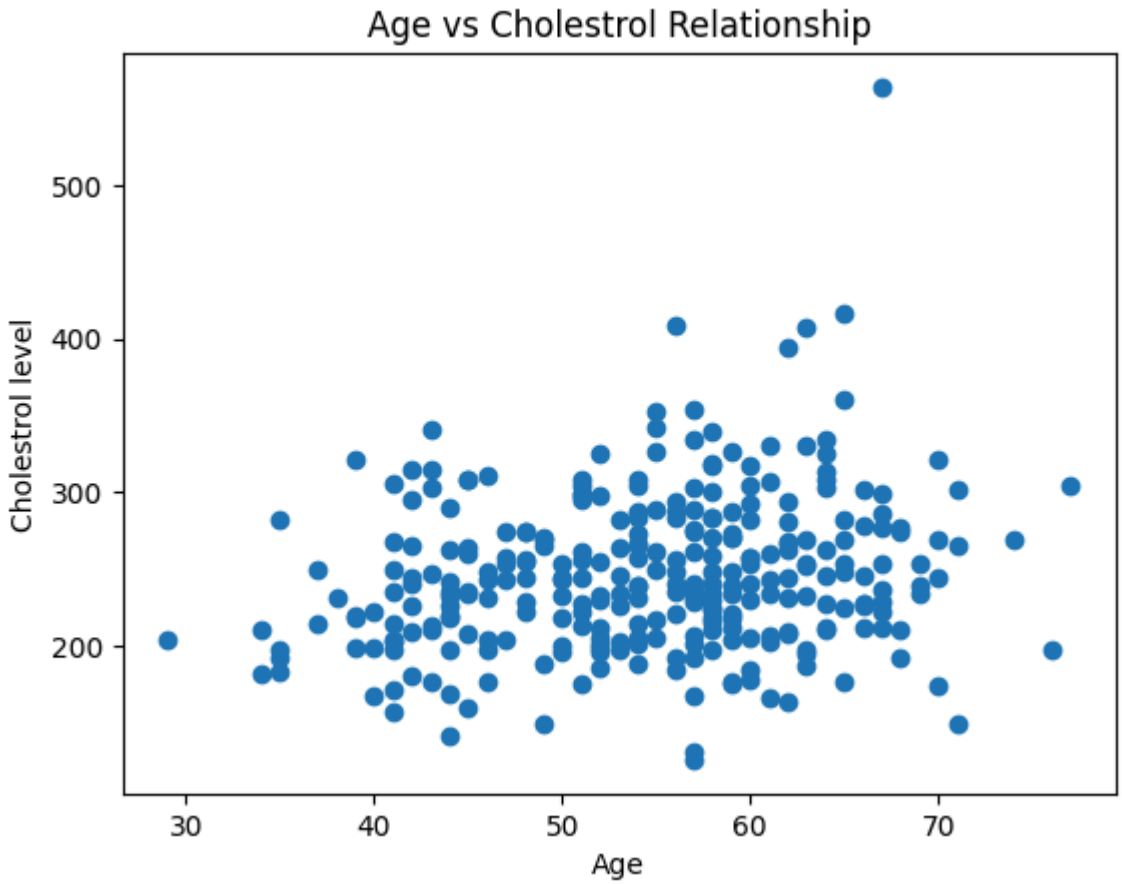
	age	sex	cp	trestbps	chol	fbs	restecg	
age	1.000000	-0.092399	0.110471	0.290476	0.202644	0.132062	0.149917	-(
sex	-0.092399	1.000000	0.008908	-0.066340	-0.198089	0.038850	0.033897	-(
cp	0.110471	0.008908	1.000000	-0.036980	0.072088	-0.057663	0.063905	-(
trestbps	0.290476	-0.066340	-0.036980	1.000000	0.131536	0.180860	0.149242	-(
chol	0.202644	-0.198089	0.072088	0.131536	1.000000	0.012708	0.165046	-(
fbs	0.132062	0.038850	-0.057663	0.180860	0.012708	1.000000	0.068831	-(
restecg	0.149917	0.033897	0.063905	0.149242	0.165046	0.068831	1.000000	-(
thalach	-0.394563	-0.060496	-0.339308	-0.049108	-0.000075	-0.007842	-0.072290	
exang	0.096489	0.143581	0.377525	0.066691	0.059339	-0.000893	0.081874	-(
oldpeak	0.197123	0.106567	0.203244	0.191243	0.038596	0.008311	0.113726	-(
slope	0.159405	0.033345	0.151079	0.121172	-0.009215	0.047819	0.135141	-(
ca	0.362210	0.091925	0.235644	0.097954	0.115945	0.152086	0.129021	-(
thal	0.120795	0.370556	0.266275	0.130612	0.023441	0.051038	0.013612	-(
condition	0.227075	0.278467	0.408945	0.153490	0.080285	0.003167	0.166343	-(

In [14]:
```
print(data['age'].corr(data['chol']))
```

```
0.20264354584662683
```

In [82]:
```
plt.scatter(data['age'],data['chol'])
plt.xlabel('Age')
plt.ylabel('Cholestrol level')
plt.title('Age vs Cholestrol Relationship')
```

Out[82]: Text(0.5, 1.0, 'Age vs Cholestrol Relationship')



6. Chest Pain type vs disease

In [83]:
```
cp_group = data.groupby('cp')['condition'].value_counts().unstack()
cp_group
```

Out[83]:

condition	0	1
cp		
0	16	7
1	40	9
2	65	18
3	39	103

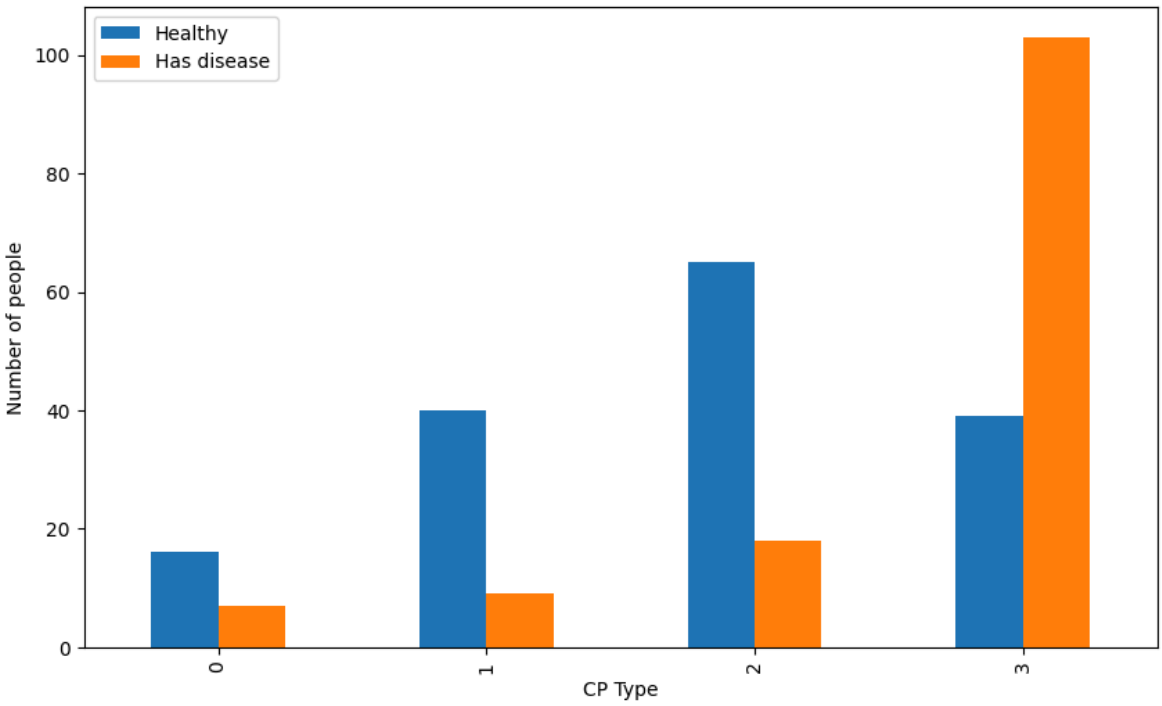
In [9]:
```
cp_group_values = data.groupby('cp')['condition'].mean()*100
cp_group_values
```

Out[9]:
```
cp
0    30.434783
1    18.367347
2    21.686747
3    72.535211
Name: condition, dtype: float64
```

In [89]:
```
#plt.bar(cp_group.index, cp_group.values)

cp_group.plot(kind='bar', figsize=(10, 6))
plt.legend(['Healthy','Has disease'])
plt.xlabel('CP Type')
plt.ylabel('Number of people')
```

Out[89]:
```
Text(0, 0.5, 'Number of people')
```



From the above graph we can interpret that the type 3 cp is more dangerous

In [11]:
```
# Assuming 'summary' is the DataFrame from the previous step
riskiest_cp = cp_group_values.idxmax()
highest_rate = cp_group_values.max()

print(f"The most risky chest pain type is Type {riskiest_cp} with a rate of {highe
```

```
The most risky chest pain type is Type 3 with a rate of 72.54
```

7. Average cholestrol by gender

In [95]:
```
gender = data.groupby('sex')['chol'].mean()
gender
```

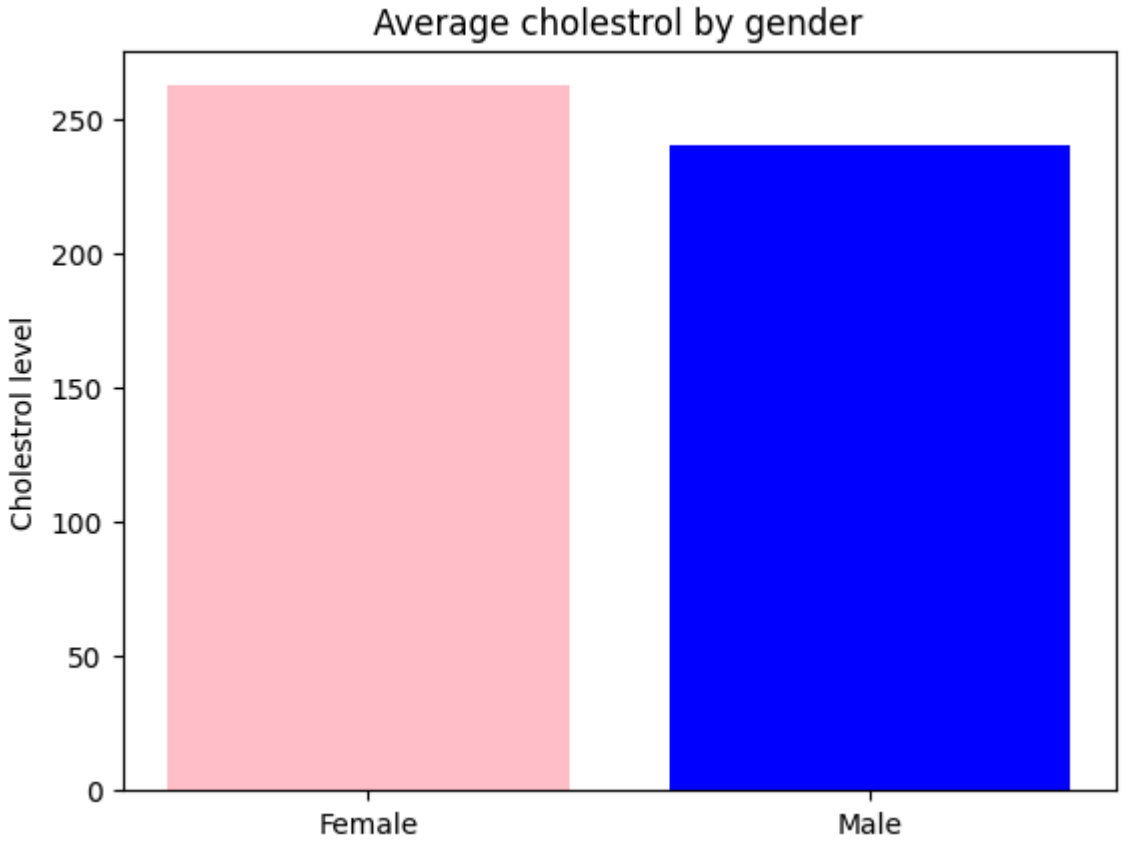
Out[95]:
```
sex
0    262.229167
1    240.243781
Name: chol, dtype: float64
```

In [15]:
```
print(data['chol'].mean())
```

```
247.35016835016836
```

In [97]:
```
plt.bar(['Female','Male'],gender.values, color=['Pink','Blue'])
plt.ylabel('Cholestrol level')
plt.title('Average cholestrol by gender')
```

Out[97]: Text(0.5, 1.0, 'Average cholestrol by gender')



8. Resting blood pressure analysis

In [101]:
```
avg_bp = data['trestbps'].mean()
print('Average blood pressure of dataset is ',round(avg_bp,2))
```

```
Average blood pressure of dataset is  131.69
```

In [105]:
```
bpgt140 = data[data['trestbps']>140]
print(f"Number of people with bp greater than 140 is {len(bpgt140)}")
bpgt140.head(10)
```

```
Number of people with bp greater than 140 is 66
```

Out[105]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	69	1	0	160	234	1	2	131	0	0.1	1	1	
2	66	0	0	150	226	0	0	114	0	2.6	2	0	
5	64	1	0	170	227	0	2	155	0	0.6	1	0	
6	63	1	0	145	233	1	2	150	0	2.3	2	0	
8	60	0	0	150	240	0	0	171	0	0.9	0	0	
9	59	1	0	178	270	0	2	145	0	4.2	2	0	
10	59	1	0	170	288	0	2	159	0	0.2	1	0	
11	59	1	0	160	273	0	2	125	0	0.0	0	0	
13	58	0	0	150	283	1	2	162	0	1.0	0	0	
16	52	1	0	152	298	1	0	178	0	1.2	1	0	

In [106]:
```
highest_bp = data['trestbps'].max()
print(highest_bp)
```

```
200
```

In [113]:
```
bp180 = data[data['trestbps']>180]['condition'].value_counts()
bp180
```

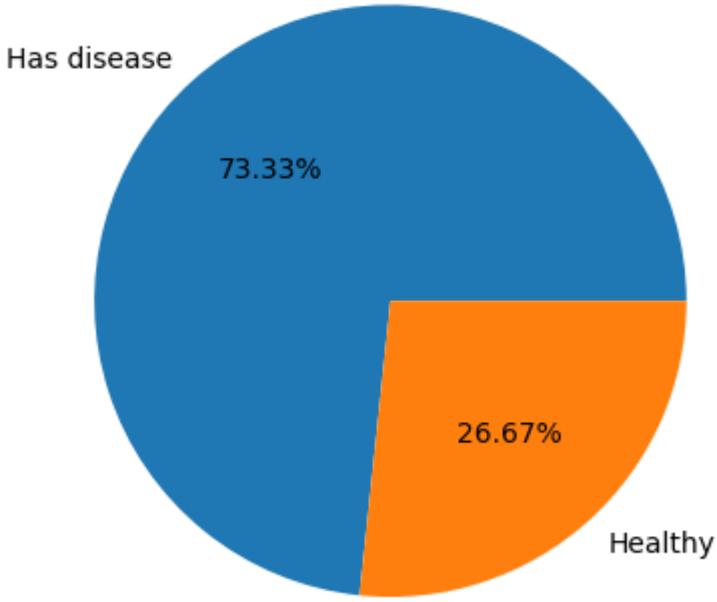
Out[113]:
```
condition
1    2
Name: count, dtype: int64
```

In [114]:
```
bp160 = data[data['trestbps']>160]['condition'].value_counts()
bp160
```

Out[114]:
```
condition
1    11
0     4
Name: count, dtype: int64
```

In [117]:
```
plt.pie(bp160.values,labels=['Has disease','Healthy'],autopct='%2.2f%%')
```

Out[117]:
```
([<matplotlib.patches.Wedge at 0x2da4f17cce0>,
  <matplotlib.patches.Wedge at 0x2da4f1785f0>],
 [Text(-0.7360437078139774, 0.817459271271329, 'Has disease'),
  Text(0.7360437843500347, -0.8174592023579401, 'Healthy')],
 [Text(-0.4014783860803513, 0.4458868752389067, '73.33%'),
  Text(0.4014784278272916, -0.4458868376497855, '26.67%')])
```



9.Maximum heartrate vs disease

In [119]:
```
avgHR_1 = data[data['condition']==1]['thalach'].mean()
print(avgHR_1)
```

```
139.1094890510949
```

In [120]:
```
avgHR_0 = data[data['condition']==0]['thalach'].mean()
print(avgHR_0)
```

```
158.58125
```

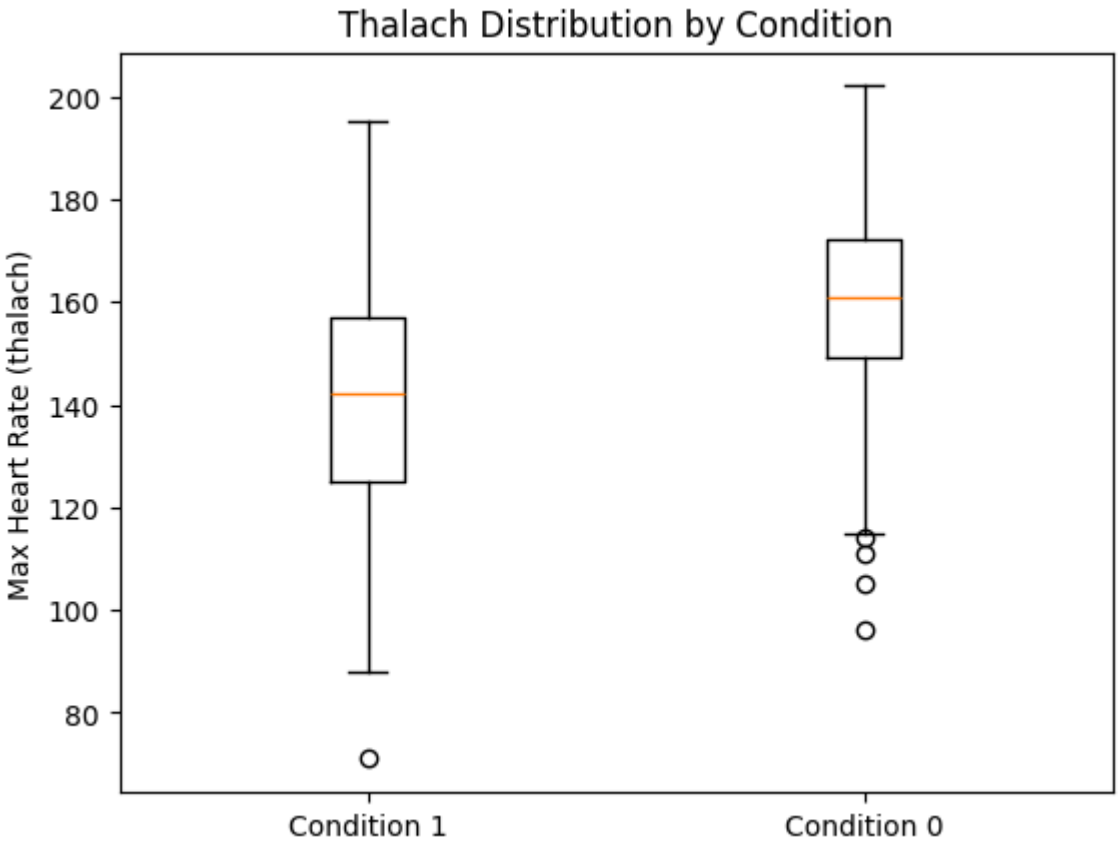
In [130]:
```
boxplotdata = [data[data['condition']==1]['thalach'],
               data[data['condition']==0]['thalach']]

plt.boxplot(boxplotdata, labels=['Condition 1', 'Condition 0'])

plt.title('Thalach Distribution by Condition')
plt.ylabel('Max Heart Rate (thalach)')
```

```
C:\Users\Administrator\AppData\Local\Temp\ipykernel_8504\3969326452.py:5: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be dropped in 3.11.
  plt.boxplot(boxplotdata, labels=['Condition 1', 'Condition 0'])
```

Out[130]:
```
Text(0, 0.5, 'Max Heart Rate (thalach)')
```



10. Exercise Induced Angina Impact

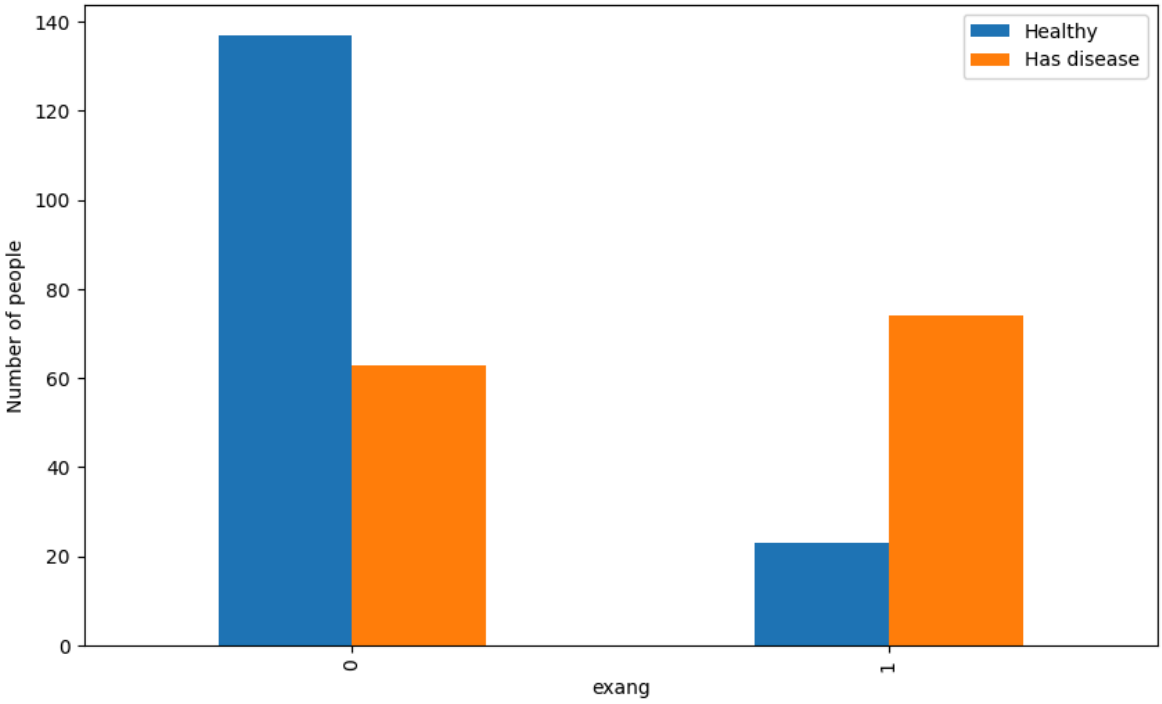
In [136]:
```
exang_data = data.groupby('exang')['condition'].value_counts().unstack()
exang_data
```

Out[136]:

condition	0	1
exang		
0	137	63
1	23	74

In [138]:
```
exang_data.plot(kind='bar', figsize=(10, 6))
plt.legend(['Healthy','Has disease'])
plt.xlabel('exang')
plt.ylabel('Number of people')
```

Out[138]:
```
Text(0, 0.5, 'Number of people')
```



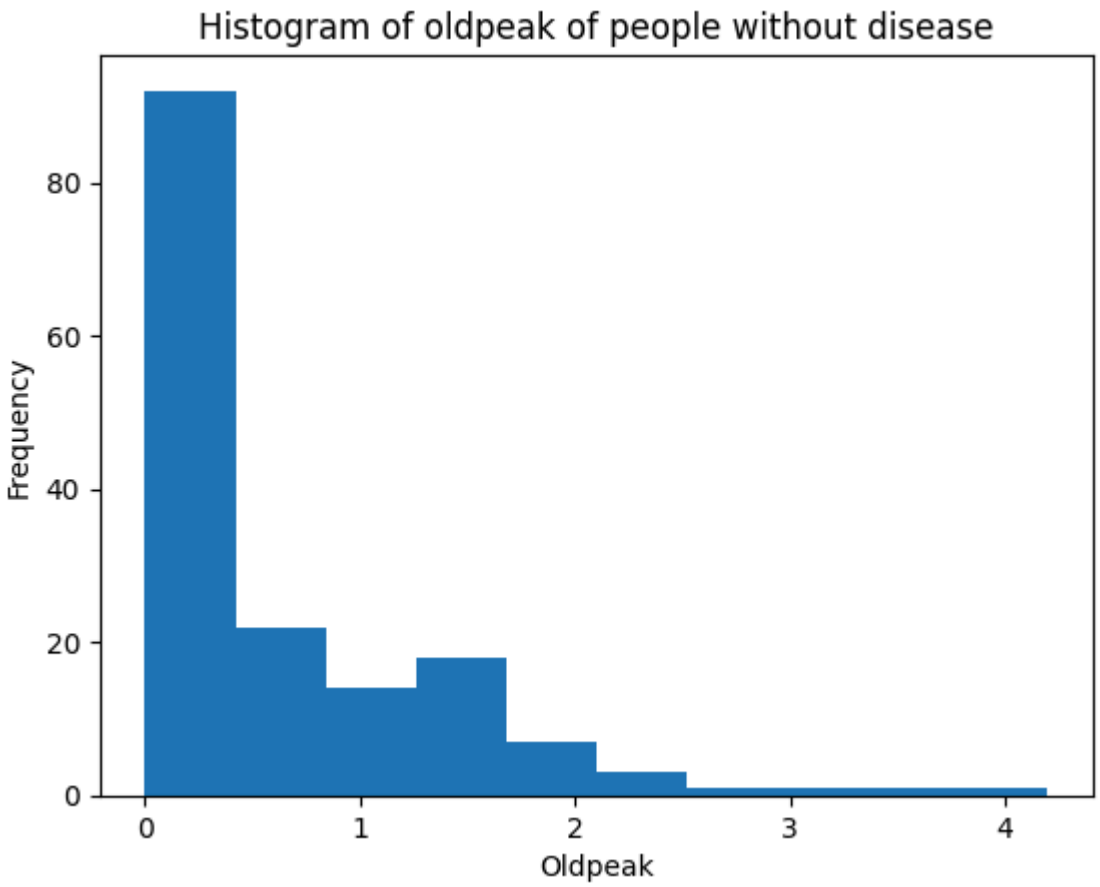
11. ST Depression (oldpeak) Analysis

In [141]:
```
oldpeak = data.groupby('condition')['oldpeak'].mean()
oldpeak
```

Out[141]:
```
condition
0    0.598750
1    1.589051
Name: oldpeak, dtype: float64
```

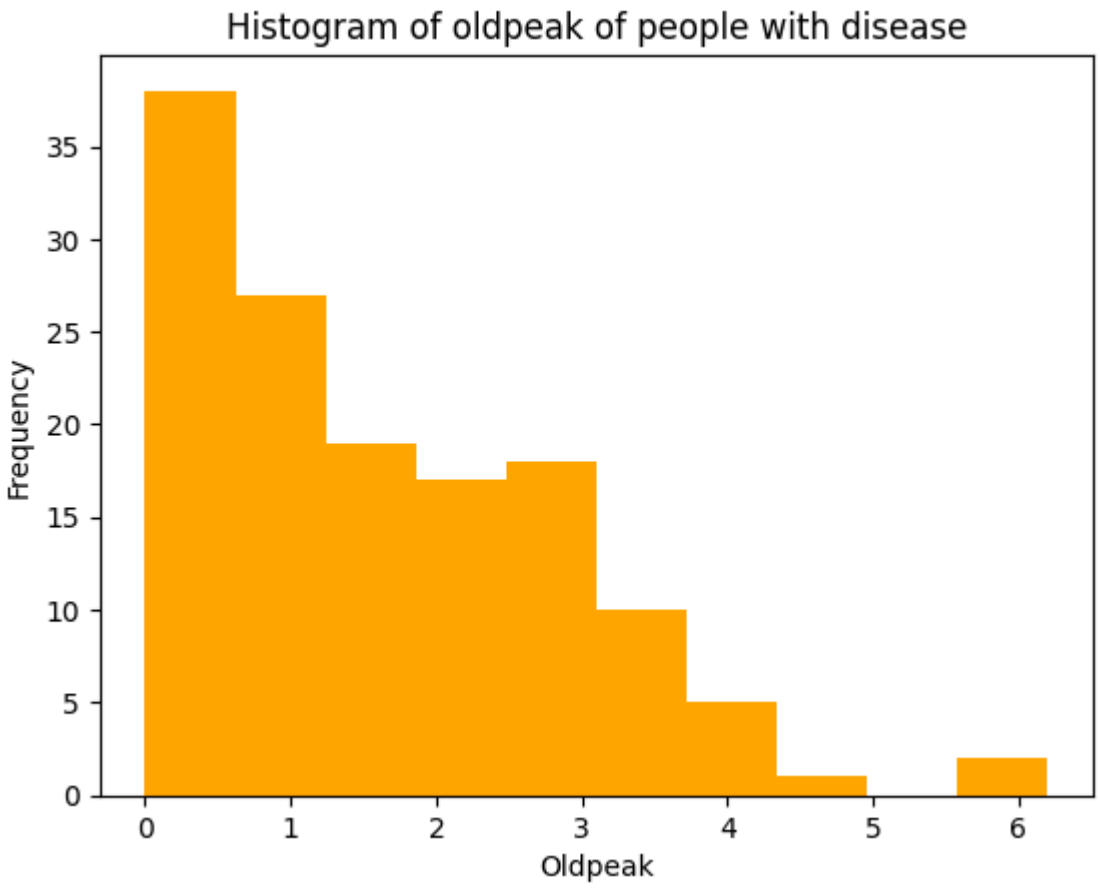
In [143]:
```
plt.hist(data[data['condition']==0]['oldpeak'])
plt.xlabel('Oldpeak')
plt.ylabel('Frequency')
plt.title('Histogram of oldpeak of people without disease')
```

Out[143]:
```
Text(0.5, 1.0, 'Histogram of oldpeak of people without disease')
```



In [144]:
```
plt.hist(data[data['condition']==1]['oldpeak'], color='orange')
plt.xlabel('Oldpeak')
plt.ylabel('Frequency')
plt.title('Histogram of oldpeak of people with disease')
```

Out[144]:
```
Text(0.5, 1.0, 'Histogram of oldpeak of people with disease')
```



People who has heart disease has a higher value of ST depression

12. Number of Major Vessels (ca) Impact

In [145]:
```
ca_group = data.groupby('ca')['condition'].value_counts()
ca_group
```

Out[145]:
```
ca  condition
0   0            129
    1             45
1   1             44
    0             21
2   1             31
    0              7
3   1             17
    0              3
Name: count, dtype: int64
```

In [167]:
```
#x = (len(ca_group[ca_group['condition']==1, ca_group['ca']==0])/len(ca_group['ca
x = data[data['ca'] == 0]['condition'].mean() * 100
x
```

Out[167]:
```
25.862068965517242
```

In [168]:
```
y = data[data['ca'] == 1]['condition'].mean() * 100
y
```

Out[168]:
```
67.6923076923077
```

In [169]:
```
z = data[data['ca'] == 2]['condition'].mean() * 100
z
```

Out[169]: 81.57894736842105

In [171]:
```
a = data[data['ca'] == 3]['condition'].mean() * 100
a
```

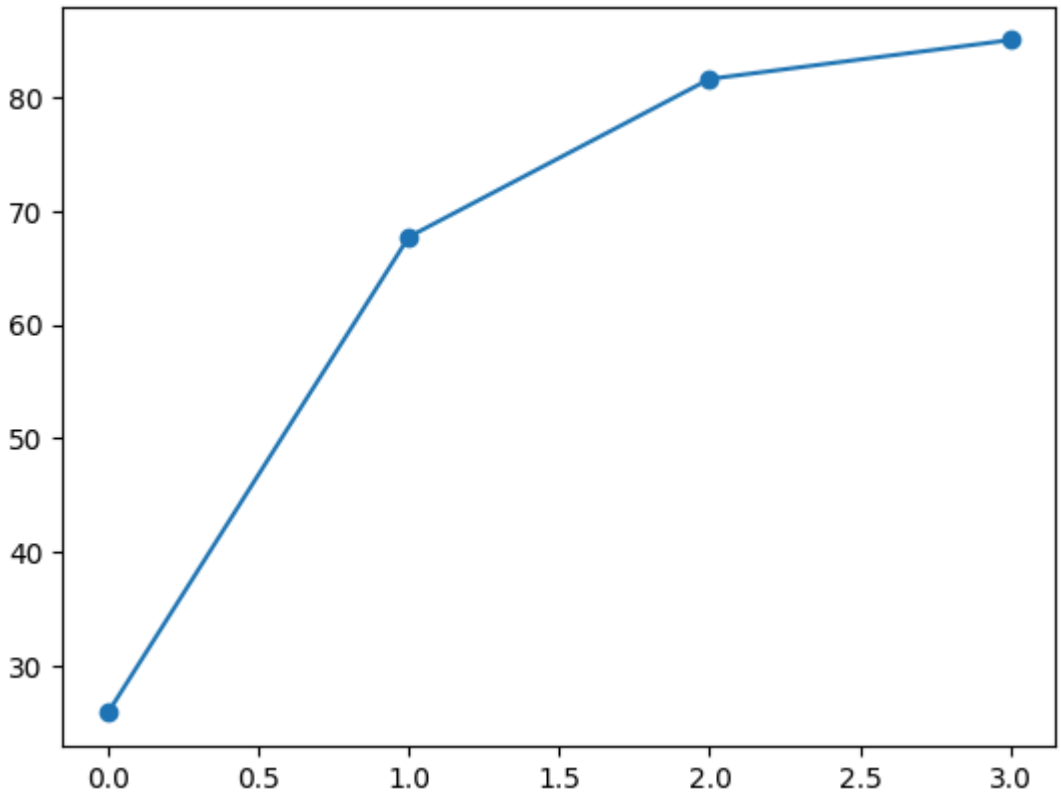
Out[171]: 85.0

In [147]:
```
disease_prob  = ca_group.groupby('ca').mean()
disease_prob
```

Out[147]:
```
ca
0    87.0
1    32.5
2    19.0
3    10.0
Name: count, dtype: float64
```

In [173]:
```
plt.plot([0,1,2,3],[x,y,z,a], linestyle='-',marker='o')
```

Out[173]: [<matplotlib.lines.Line2D at 0x2da51e4af90>]

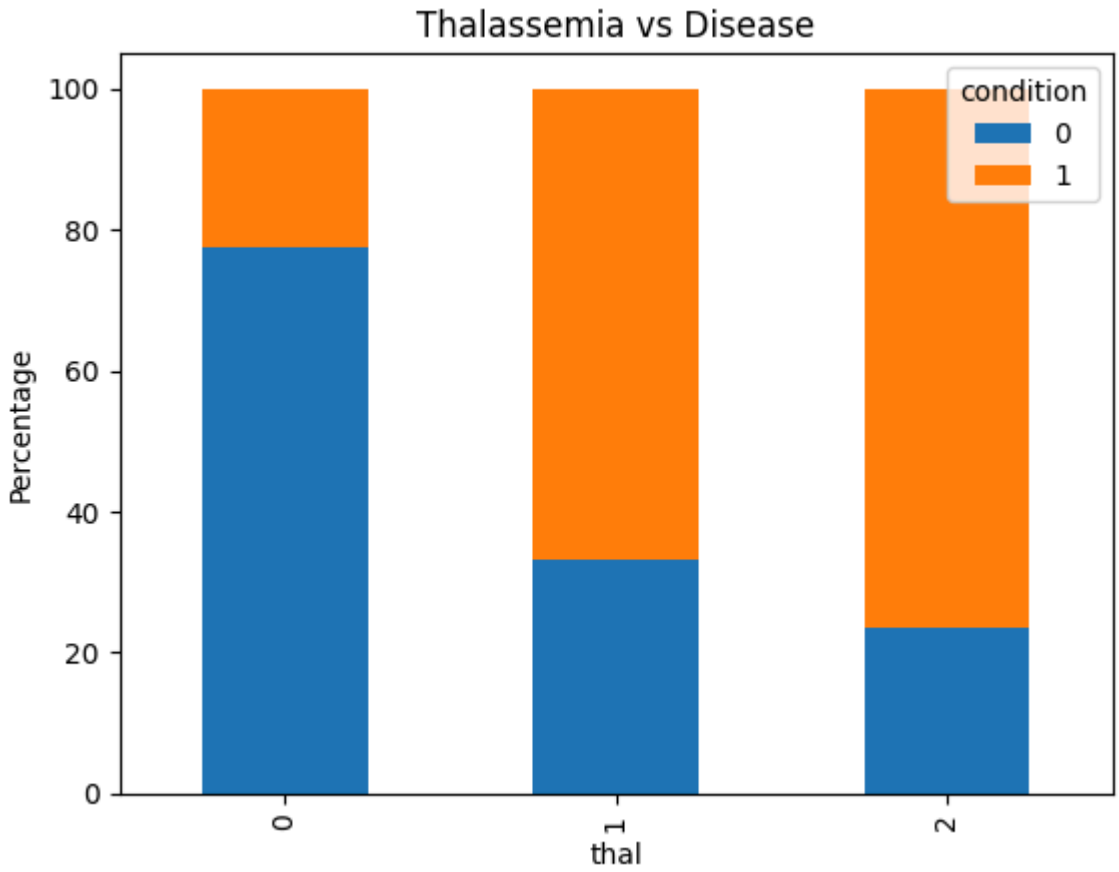


13. Thalassemia vs Disease

In [174]:
```
cross= pd.crosstab(data['thal'], data['condition'])

cross = cross.div(cross.sum(axis=1), axis=0) * 100

cross.plot(kind='bar', stacked=True, title="Thalassemia vs Disease")
plt.ylabel('Percentage')
plt.show()
```



In [ ]:

In [ ]:

14. Multi-Factor Risk Analysis

In [154]:
```
agegt50 = data[data['age'] > 50]
agegt50
```

Out[154]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tl
0	69	1	0	160	234	1	2	131	0	0.1	1	1	
1	69	0	0	140	239	0	0	151	0	1.8	0	2	
2	66	0	0	150	226	0	0	114	0	2.6	2	0	
3	65	1	0	138	282	1	2	174	0	1.4	1	1	
4	64	1	0	110	211	0	2	144	1	1.8	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
258	52	1	3	125	212	0	0	168	0	1.0	0	2	
259	51	0	3	130	305	0	0	142	1	1.2	1	0	
260	51	1	3	140	298	0	0	122	1	4.2	1	3	
261	51	1	3	140	261	0	2	186	1	0.0	0	0	
262	51	1	3	140	299	0	0	173	1	1.6	0	0	

205 rows × 15 columns

In [157]:
```
cholegt240 = data[data['chol'] > 240]
cholegt240
```

Out[157]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tl
3	65	1	0	138	282	1	2	174	0	1.4	1	1	
9	59	1	0	178	270	0	2	145	0	4.2	2	0	
10	59	1	0	170	288	0	2	159	0	0.2	1	0	
11	59	1	0	160	273	0	2	125	0	0.0	0	0	
13	58	0	0	150	283	1	2	162	0	1.0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
284	43	1	3	115	303	0	0	181	0	1.2	1	0	
285	43	1	3	150	247	0	0	171	0	1.5	0	0	
287	42	0	3	102	265	0	2	122	0	0.6	1	0	
288	42	1	3	136	315	0	0	125	1	1.8	1	0	
296	35	1	3	126	282	0	2	156	1	0.0	0	0	

151 rows × 15 columns

In [158]:
```
bpgt140 = data[data['trestbps'] > 140]
bpgt140
```

Out[158]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tl
0	69	1	0	160	234	1	2	131	0	0.1	1	1	
2	66	0	0	150	226	0	0	114	0	2.6	2	0	
5	64	1	0	170	227	0	2	155	0	0.6	1	0	
6	63	1	0	145	233	1	2	150	0	2.3	2	0	
8	60	0	0	150	240	0	0	171	0	0.9	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
263	50	1	3	150	243	0	2	128	0	2.6	1	0	
264	50	1	3	144	200	0	2	126	1	0.9	1	0	
277	45	1	3	142	309	0	2	147	1	0.0	1	3	
285	43	1	3	150	247	0	0	171	0	1.5	0	0	
292	40	1	3	152	223	0	0	181	0	0.0	0	0	

66 rows × 15 columns

In [162]:
```
x = (len(agegt50[agegt50['condition']==1])/len(agegt50))*100
print(f"The percentage of people with disease having age > 50 is : {round(x,2)}")
```

```
The percentage of people with disease having age > 50 is : 53.17
```

In [164]:
```
x = (len(cholegt240[cholegt240['condition']==1])/len(cholegt240))*100
print(f"The percentage of people with disease having cholestrol > 240 is : {round(
```

```
The percentage of people with disease having cholestrol > 240 is : 52.32
```

In [165]:
```
bpgt140
x = (len(bpgt140[bpgt140['condition']==1])/len(bpgt140))*100
print(f"The percentage of people with disease having bp > 140 is : {round(x,2)}")
```

```
The percentage of people with disease having bp > 140 is : 59.09
```

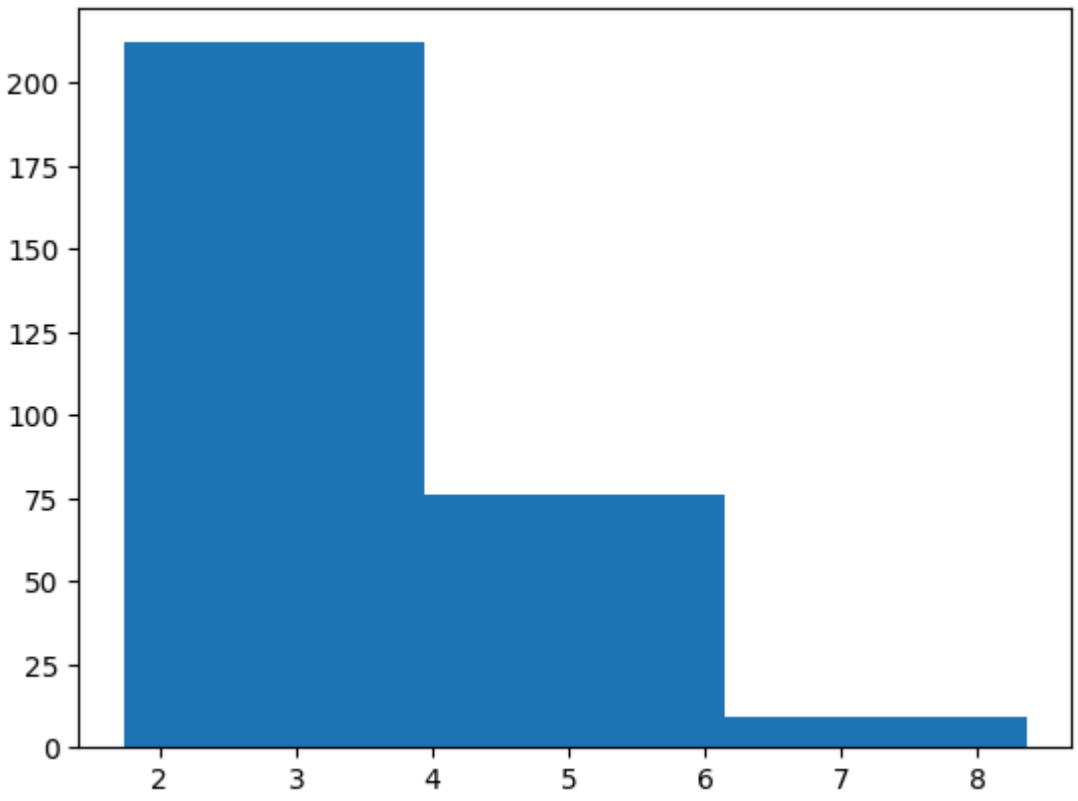
15. Create Risk Score (Custom Analysis)

In [150]:
```
data['risk_score'] = (data['chol']/200) + (data['trestbps']/120) + (data['oldpeak'
data['risk_score']
```

Out[150]:
```
0      2.603333
1      4.161667
2      4.980000
3      3.960000
4      3.771667
         ...
292    2.381667
293    3.278333
294    3.590000
295    3.465000
296    2.460000
Name: risk_score, Length: 297, dtype: float64
```

In [151]:
```
plt.hist(data['risk_score'], bins=3)
```

Out[151]:
```
(array([212.,  76.,   9.]),
 array([1.73333333, 3.94      , 6.14666667, 8.35333333]),
 <BarContainer object of 3 artists>)
```

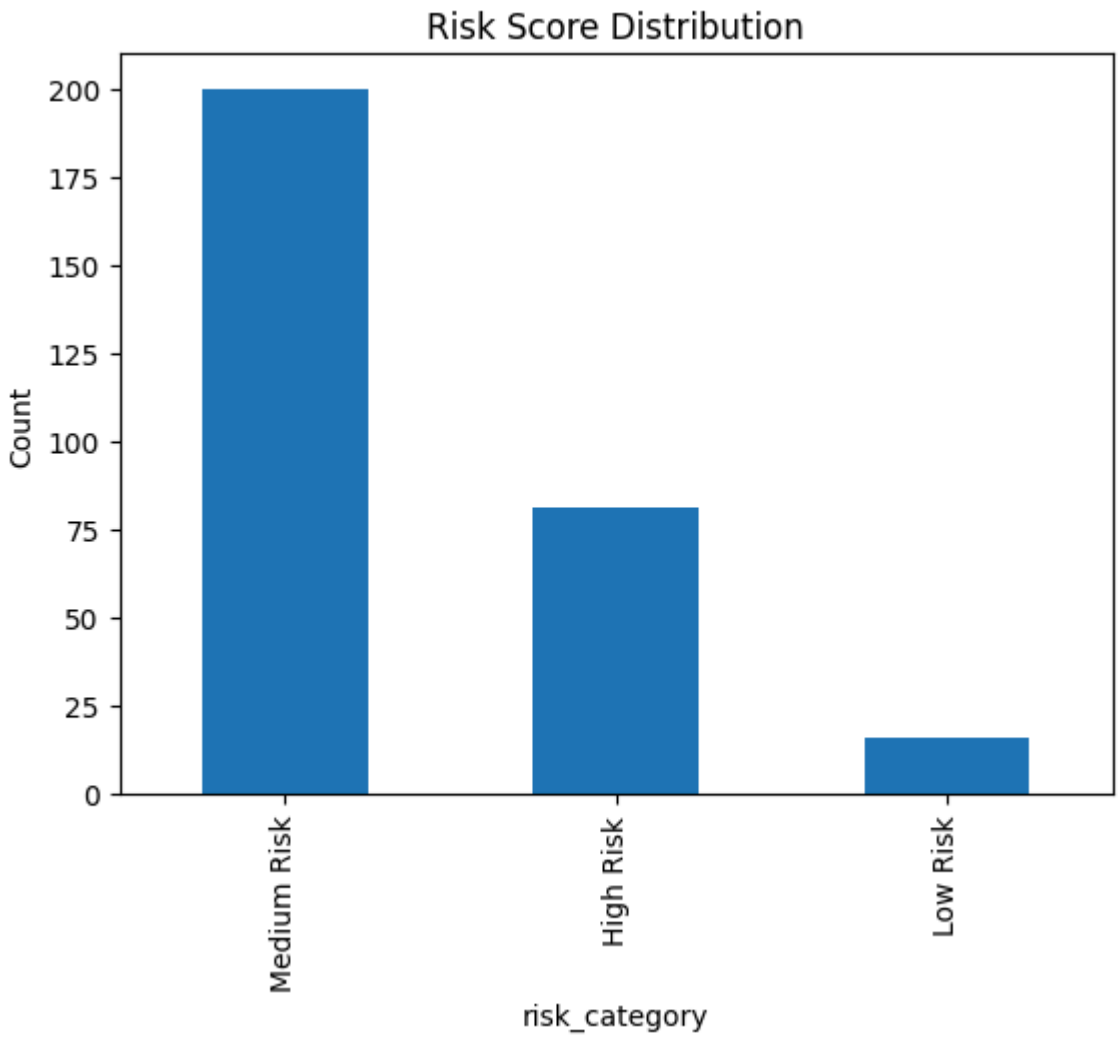


In [180]:
```
#data['risk_score'] = (data['chol'] / 200) + (data['trestbps'] / 120) + data['oldp

conditions = [
    (data['risk_score'] < 2),
    (data['risk_score'] >= 2) & (data['risk_score'] < 4),
    (data['risk_score'] >= 4)
]
choices = ['Low Risk', 'Medium Risk', 'High Risk']

data['risk_category'] = np.select(conditions, choices)


data['risk_category'].value_counts().plot(kind='bar', title="Risk Score Distributi
plt.ylabel('Count')
plt.show()
```



In [16]:
```
data.corr()
```

Out[16]:

	age	sex	cp	trestbps	chol	fbs	restecg	
age	1.000000	-0.092399	0.110471	0.290476	0.202644	0.132062	0.149917	-(
sex	-0.092399	1.000000	0.008908	-0.066340	-0.198089	0.038850	0.033897	-(
cp	0.110471	0.008908	1.000000	-0.036980	0.072088	-0.057663	0.063905	-(
trestbps	0.290476	-0.066340	-0.036980	1.000000	0.131536	0.180860	0.149242	-(
chol	0.202644	-0.198089	0.072088	0.131536	1.000000	0.012708	0.165046	-(
fbs	0.132062	0.038850	-0.057663	0.180860	0.012708	1.000000	0.068831	-(
restecg	0.149917	0.033897	0.063905	0.149242	0.165046	0.068831	1.000000	-(
thalach	-0.394563	-0.060496	-0.339308	-0.049108	-0.000075	-0.007842	-0.072290	1
exang	0.096489	0.143581	0.377525	0.066691	0.059339	-0.000893	0.081874	-(
oldpeak	0.197123	0.106567	0.203244	0.191243	0.038596	0.008311	0.113726	-(
slope	0.159405	0.033345	0.151079	0.121172	-0.009215	0.047819	0.135141	-(
ca	0.362210	0.091925	0.235644	0.097954	0.115945	0.152086	0.129021	-(
thal	0.120795	0.370556	0.266275	0.130612	0.023441	0.051038	0.013612	-(
condition	0.227075	0.278467	0.408945	0.153490	0.080285	0.003167	0.166343	-(

Does cholesterol strongly impact heart disease?

```
It has a very small positive corelation, ie a person with
high cholestrol has a small chance of having heart disease.
```

Is male population more vulnerable? Yes based on the data provided men have a higher chance of having heart disease.

Does exercise-induced angina significantly increase risk? Yes exercise-induced angina can significantly increase the risk of heart disease as they have a very strong positive corelation.

Which feature has strongest correlation with disease? thal has the strongest correlation with having a heart disease.