

## task-5

```
[51]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
[52]: df=pd.read_csv("C:\\Users\\kumar\\Desktop\\DATA SETS\\heart-1.csv")
```

```
[53]: df.head(10)
```

```
[53]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	\
0	52	1	0	125	212	0	1	168	0	1.0	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	
4	62	0	0	138	294	1	1	106	0	1.9	1	
5	58	0	0	100	248	0	0	122	0	1.0	1	
6	58	1	0	114	318	0	2	140	0	4.4	0	
7	55	1	0	160	289	0	0	145	1	0.8	1	
8	46	1	0	120	249	0	0	144	0	0.8	2	
9	54	1	0	122	286	0	0	116	1	3.2	1	

	ca	thal	target
0	2	3	0
1	0	3	0
2	0	3	0
3	1	3	0
4	3	2	0
5	0	2	1
6	3	1	0
7	1	3	0
8	0	3	0
9	2	2	0

```
[54]: df.tail()
```

```
[54]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
1020	59	1	1	140	221	0	1	164	1	0.0	
1021	60	1	0	125	258	0	0	141	1	2.8	
1022	47	1	0	110	275	0	0	118	1	1.0	
1023	50	0	0	110	254	0	0	159	0	0.0	
1024	54	1	0	120	188	0	1	113	0	1.4	

	slope	ca	thal	target
1020	2	0	2	1
1021	1	1	3	0
1022	1	1	2	0
1023	2	0	2	1
1024	1	1	3	0

```
[55]: df.columns.values
```

```
[55]: array(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
        'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
        dtype=object)
```

```
[56]: df.isna().sum()
```

```
[56]: 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
target         0
dtype: int64
```

```
[57]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         1025 non-null   int64
1   sex         1025 non-null   int64
```

```

2   cp          1025 non-null   int64
3   trestbps    1025 non-null   int64
4   chol        1025 non-null   int64
5   fbs         1025 non-null   int64
6   restecg     1025 non-null   int64
7   thalach     1025 non-null   int64
8   exang       1025 non-null   int64
9   oldpeak     1025 non-null   float64
10  slope       1025 non-null   int64
11  ca          1025 non-null   int64
12  thal        1025 non-null   int64
13  target      1025 non-null   int64

```

dtypes: float64(1), int64(13)

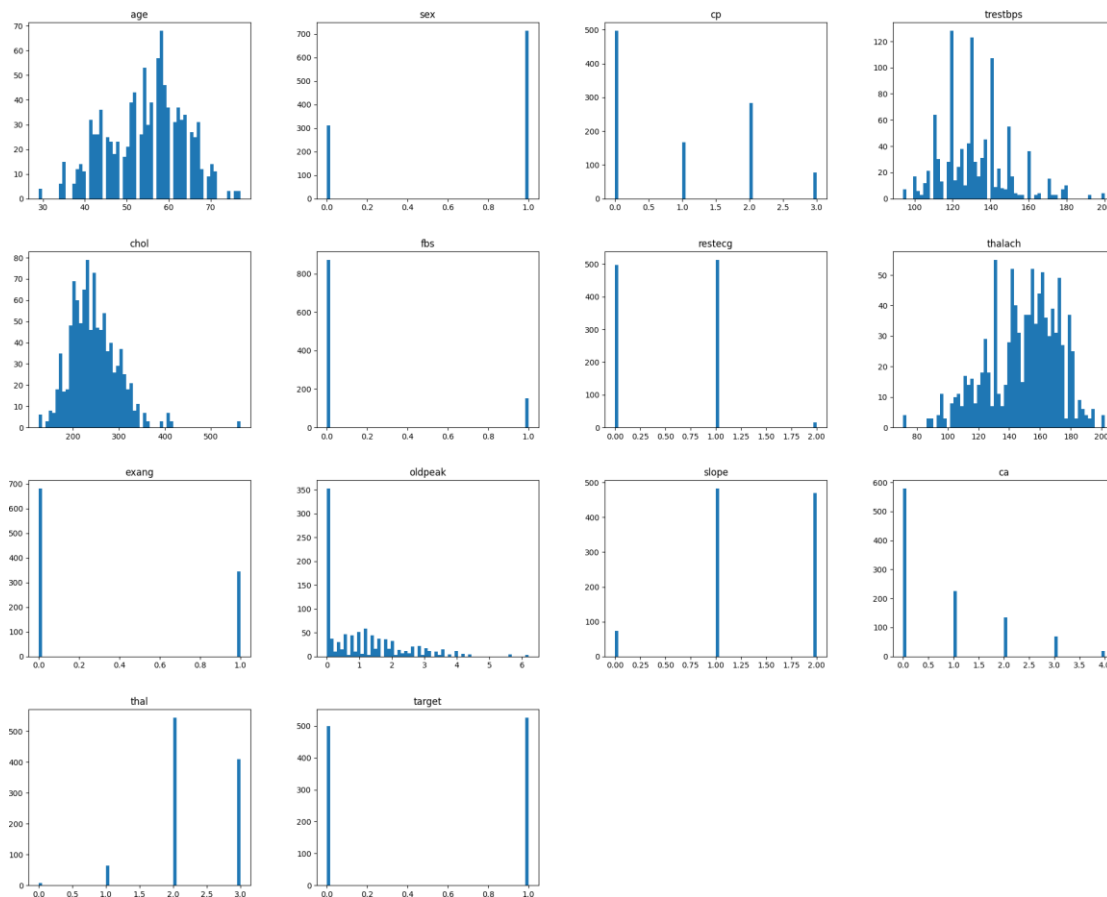
memory usage: 112.2 KB

```
[58]: df.hist(bins=60, grid=False, figsize=(25,20))
```

```

[58]: array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'sex'}>,
  <Axes: title={'center': 'cp'}>,
  <Axes: title={'center': 'trestbps'}>],
  [<Axes: title={'center': 'chol'}>,
  <Axes: title={'center': 'fbs'}>,
  <Axes: title={'center': 'restecg'}>,
  <Axes: title={'center': 'thalach'}>],
  [<Axes: title={'center': 'exang'}>,
  <Axes: title={'center': 'oldpeak'}>,
  <Axes: title={'center': 'slope'}>,
  <Axes: title={'center': 'ca'}>],
  [<Axes: title={'center': 'thal'}>,
  <Axes: title={'center': 'target'}>, <Axes: >, <Axes: >]],
  dtype=object)

```



[59]: df.describe()

```
[59]:
```

	age	sex	cp	trestbps	chol	\
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	
mean	54.434146	0.695610	0.942439	131.611707	246.000000	
std	9.072290	0.460373	1.029641	17.516718	51.59251	
min	29.000000	0.000000	0.000000	94.000000	126.000000	
25%	48.000000	0.000000	0.000000	120.000000	211.000000	
50%	56.000000	1.000000	1.000000	130.000000	240.000000	
75%	61.000000	1.000000	2.000000	140.000000	275.000000	
max	77.000000	1.000000	3.000000	200.000000	564.000000	

	fbs	restecg	thalach	exang	oldpeak	\
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	
mean	0.149268	0.529756	149.114146	0.336585	1.071512	
std	0.356527	0.527878	23.005724	0.472772	1.175053	
min	0.000000	0.000000	71.000000	0.000000	0.000000	
25%	0.000000	0.000000	132.000000	0.000000	0.000000	
50%	0.000000	1.000000	152.000000	0.000000	0.800000	

75%	0.000000	1.000000	166.000000	1.000000	1.800000
max	1.000000	2.000000	202.000000	1.000000	6.200000

	slope	ca	thal	target
count	1025.000000	1025.000000	1025.000000	1025.000000
mean	1.385366	0.754146	2.323902	0.513171
std	0.617755	1.030798	0.620660	0.500070
min	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	2.000000	0.000000
50%	1.000000	0.000000	2.000000	1.000000
75%	2.000000	1.000000	3.000000	1.000000
max	2.000000	4.000000	3.000000	1.000000

```
[60]: questions = ["1. Which sex group has more heart disease ?"
                  "2. How many people have heart disease ?"
                  "3. What is the distribution of chest pain among patients with_
↳different thalassemia level ?"
                  "4. How does age effect the heart desease ?"
                  "5. What is the distribution of chest pain among heart disease_
↳patients ?"
                  "6. What is the relationship between fbs and heart disease ?"
                  "7. How does heart disease vary with maximum heart rate achived ?"]
questions
```

```
[60]: ['1. Which sex group has more heart disease ?2. How many people have heart
disease ?3. What is the distribution of chest pain among patients with different
thalassemia level ?4. How does age effect the heart desease ?5. What is the
distribution of chest pain among heart disease patients ?6. What is the
relationship between fbs and heart disease ?7. How does heart disease vary with
maximum heart rate achived ?']
```

```
[61]: # 1=male
      # 0=female

df.sex.value_counts()
```

```
[61]: sex
1      713
0      312
Name: count, dtype: int64
```

```
[62]: # 1=heart disease
      # 0=no heart disease

df.target.value_counts()
```

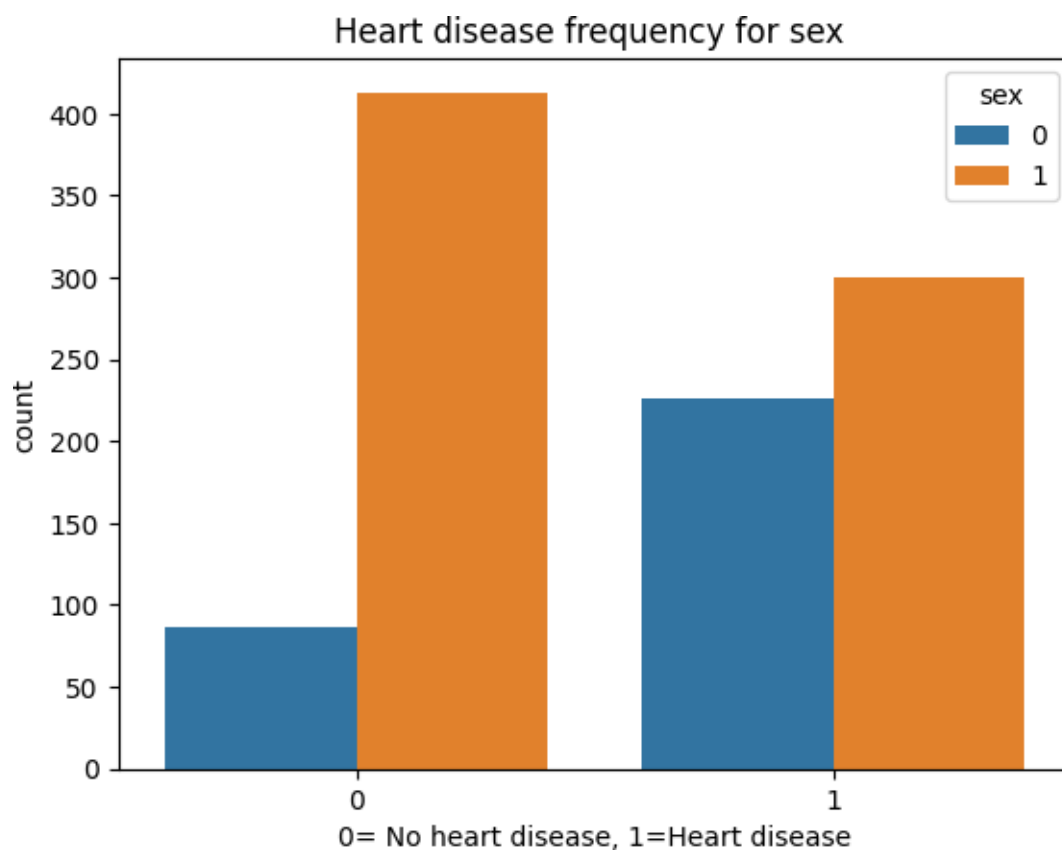
```
[62]: target
      1    526
      0    499
      Name: count, dtype: int64
```

```
[73]: pd.crosstab(df.target, df.sex)
```

```
[73]: sex      0      1
      target
      0      86  413
      1     226  300
```

```
[74]: #1. Which sex group has more heart disease ?
      sns.countplot(x="target", data=df, hue="sex")
      plt.title("Heart disease frequency for sex ")
      plt.xlabel("0= No heart disease, 1=Heart disease")
```

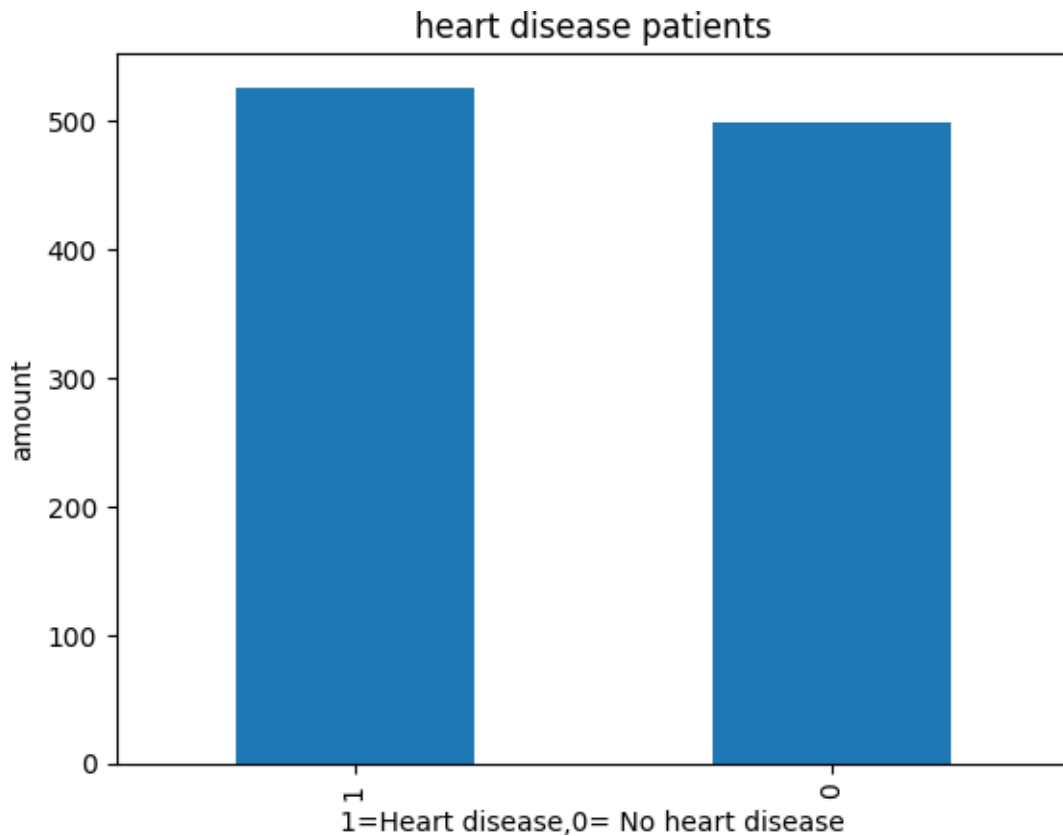
```
[74]: Text(0.5, 0, '0= No heart disease, 1=Heart disease')
```



[75]: *# 2. How many people have heart disease ?*

```
df.target.value_counts().plot(kind="bar")  
plt.title("heart disease patients")  
plt.xlabel("1=Heart disease,0= No heart disease")  
plt.ylabel("amount")
```

[75]: Text(0, 0.5, 'amount')

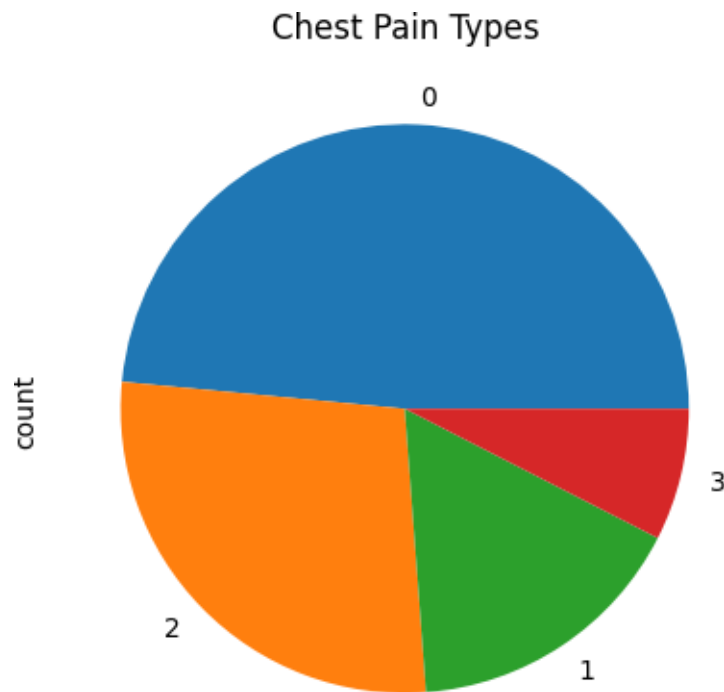


[76]: df.cp.value\_counts()

```
[76]: cp  
0    497  
2    284  
1    167  
3     77  
Name: count, dtype: int64
```

```
[77]: df.cp.value_counts().plot(kind="pie")  
plt.title("Chest Pain Types")
```

[77]: Text(0.5, 1.0, 'Chest Pain Types')



[78]: #3. What is the distribution of chest pain among patients with different thalassemia level ?

```
sns.barplot(x='cp', y='thal',data=df,palette='rocket' )  
plt.title("chest pain patient among different thalassemia level")  
plt.xlabel('Chest Pain')  
plt.ylabel('Thalassemia')
```

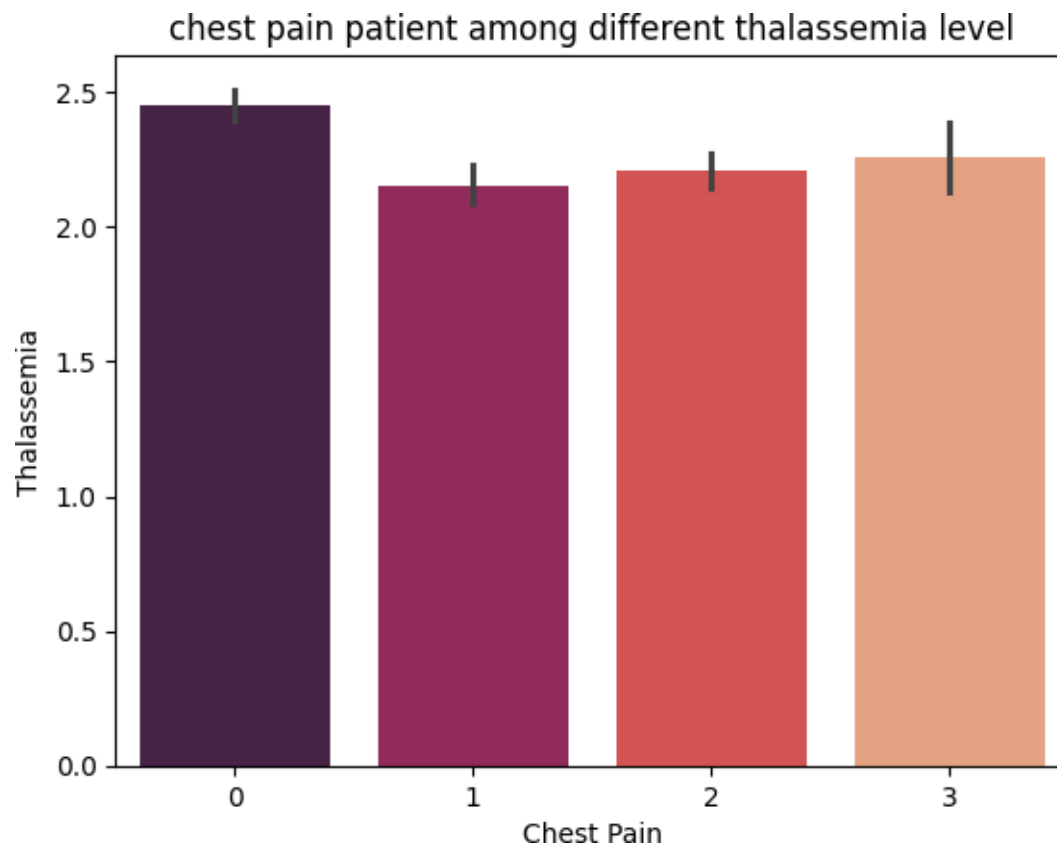
C:\Users\kumar\AppData\Local\Temp\ipykernel\_29464\132151036.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='cp', y='thal',data=df,palette='rocket' )
```

[78]: Text(0, 0.5, 'Thalassemia')

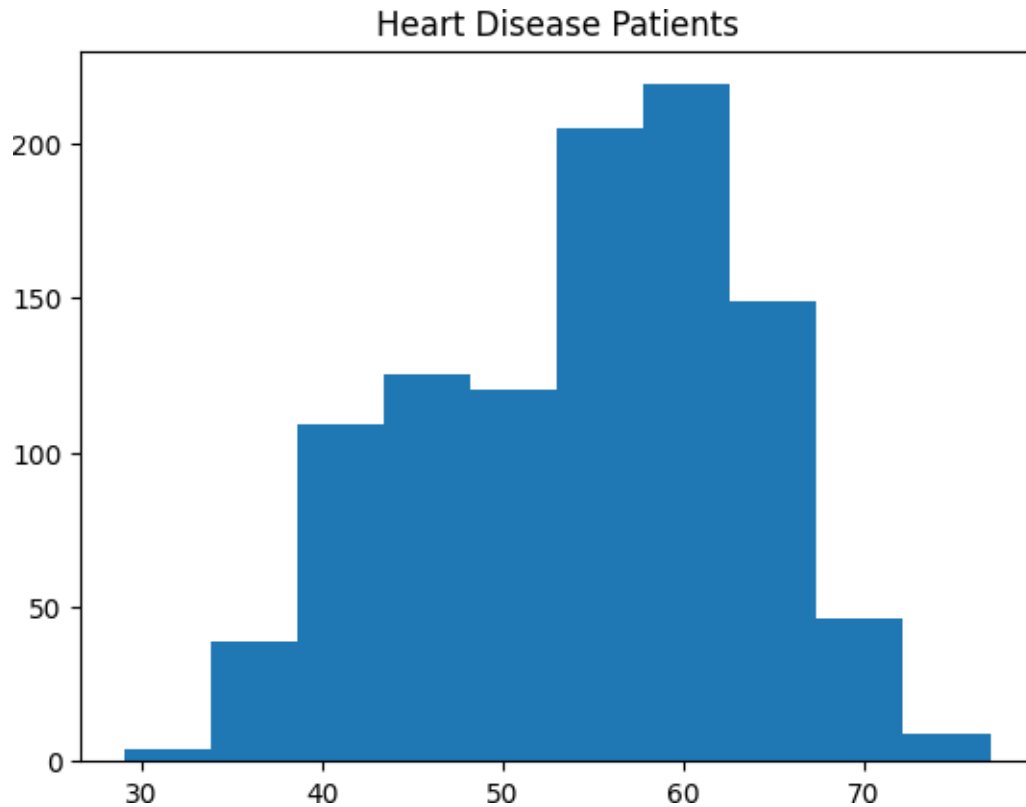




[79]: # 4. How does age effect the heart disease ?

```
plt.hist(df['age'])  
plt.title('Heart Disease Patients')
```

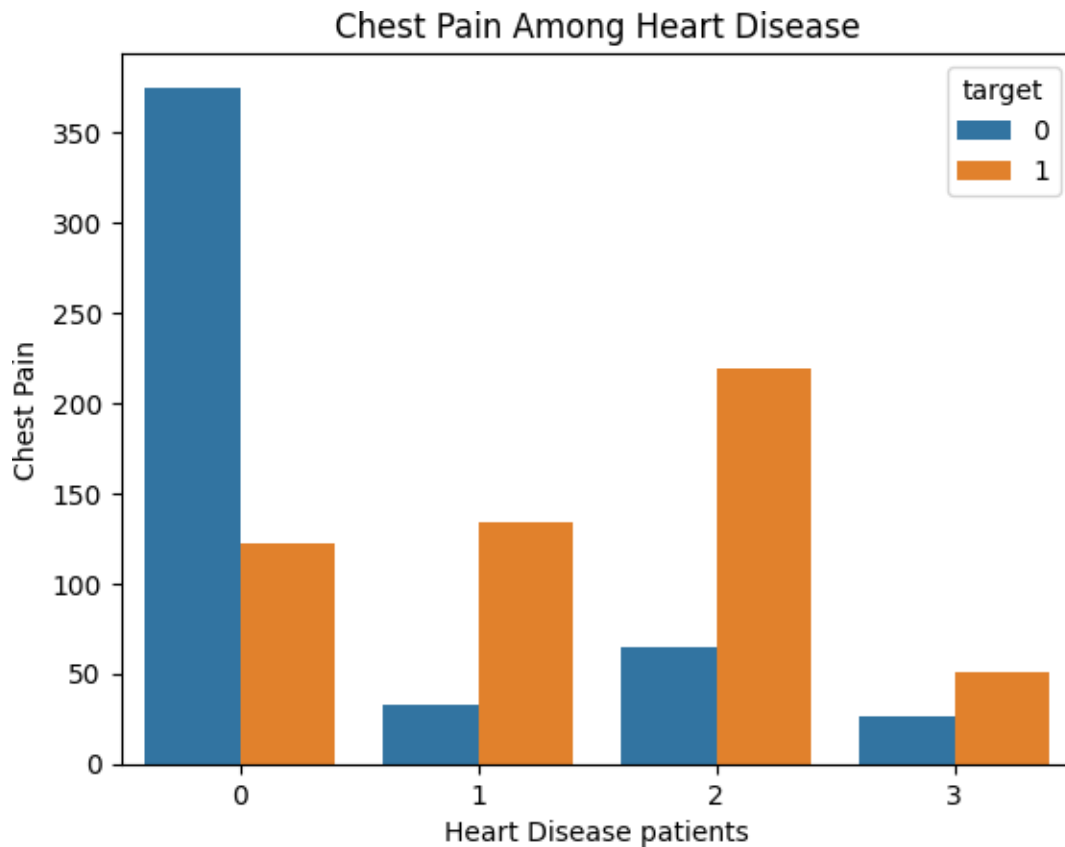
[79]: Text(0.5, 1.0, 'Heart Disease Patients')



[80]: *# 5. What is the distribution of chest pain among heart disease patients ?*

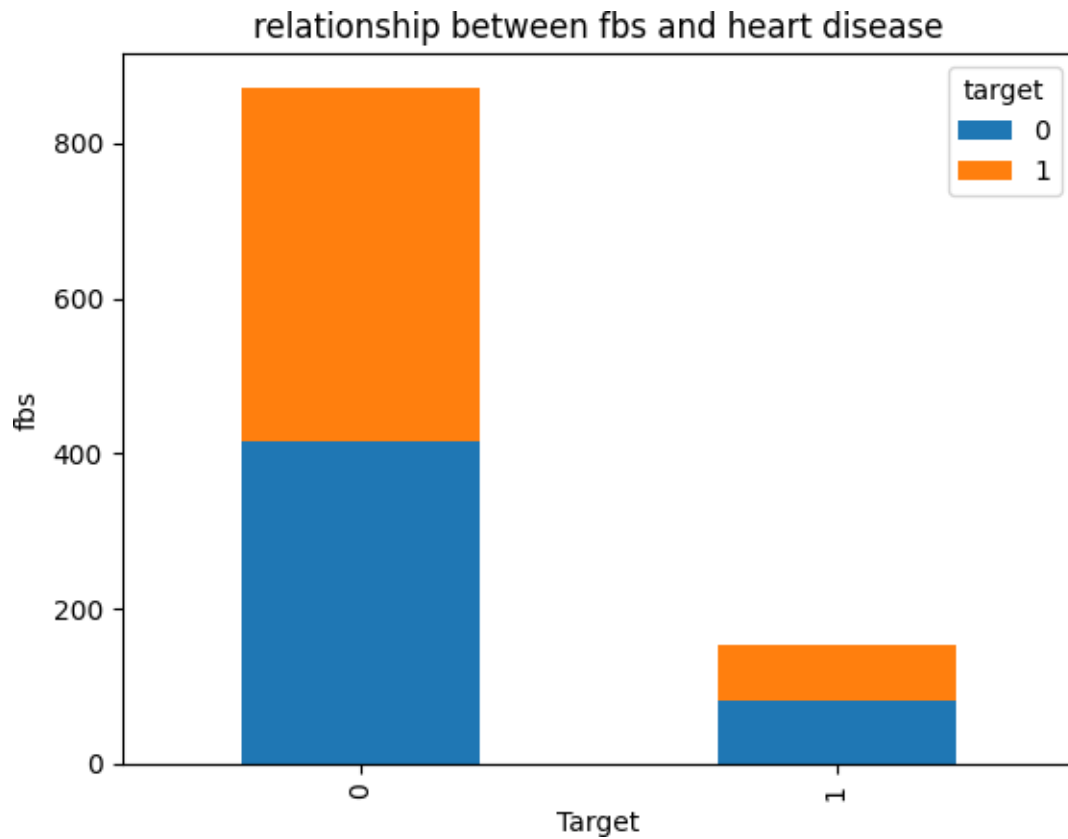
```
sns.countplot(x='cp', hue='target', data=df)
plt.title('Chest Pain Among Heart Disease')
plt.xlabel('Heart Disease patients')
plt.ylabel('Chest Pain')
```

[80]: Text(0, 0.5, 'Chest Pain')



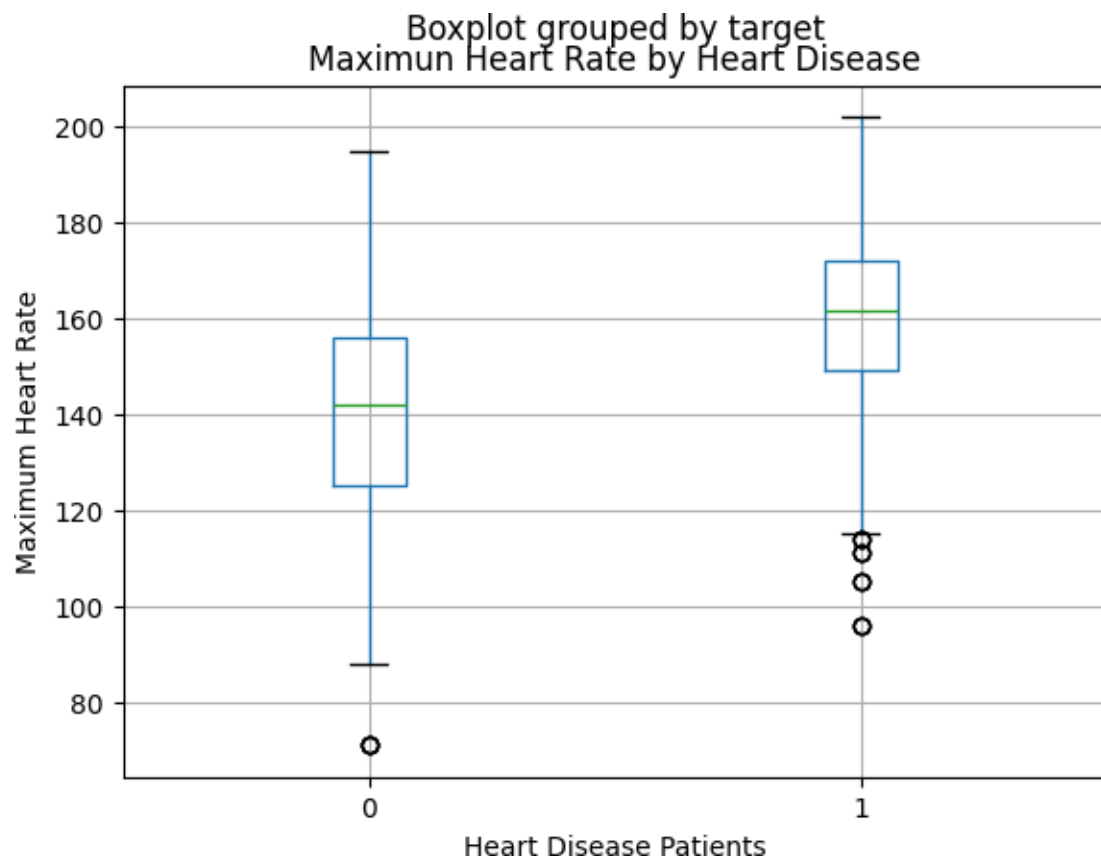
[81]: # 6. What is the relationship between fbs and heart disease ?

```
fbs_target_counts= df.groupby('fbs')['target'].value_counts().unstack().  
    ↪ fillna(0)  
fbs_target_counts.plot(kind='bar',stacked=True)  
plt.title('relationship between fbs and heart disease')  
plt.ylabel('fbs')  
plt.xlabel('Target')  
plt.show()
```



[82]: # 7. How does heart disease vary with maximum heart rate achieved ?

```
df.boxplot(column="thalach", by="target")  
plt.title("Maximun Heart Rate by Heart Disease")  
plt.ylabel("Maximum Heart Rate")  
plt.xlabel("Heart Disease Patients")  
plt.show()
```



[ ]: