In [ ]:
```
#Importing all the libraries that we need
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [ ]:
```
#importing our data
df = pd.read_csv('heart.csv')
```

In [ ]:
```
#checking first five rows
df.head()
```

Out[ ]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2

In [ ]:
```
# checking last five rows
df.tail()
```

Out[ ]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
1020	59	1	1	140	221	0	1	164	1	0.0	2	0
1021	60	1	0	125	258	0	0	141	1	2.8	1	1
1022	47	1	0	110	275	0	0	118	1	1.0	1	1
1023	50	0	0	110	254	0	0	159	0	0.0	2	0
1024	54	1	0	120	188	0	1	113	0	1.4	1	1

In [ ]:
```
#t ake a look at column names
df.columns.values
```

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

In [ ]:
```
# checking for null values
df.isna().sum()
```

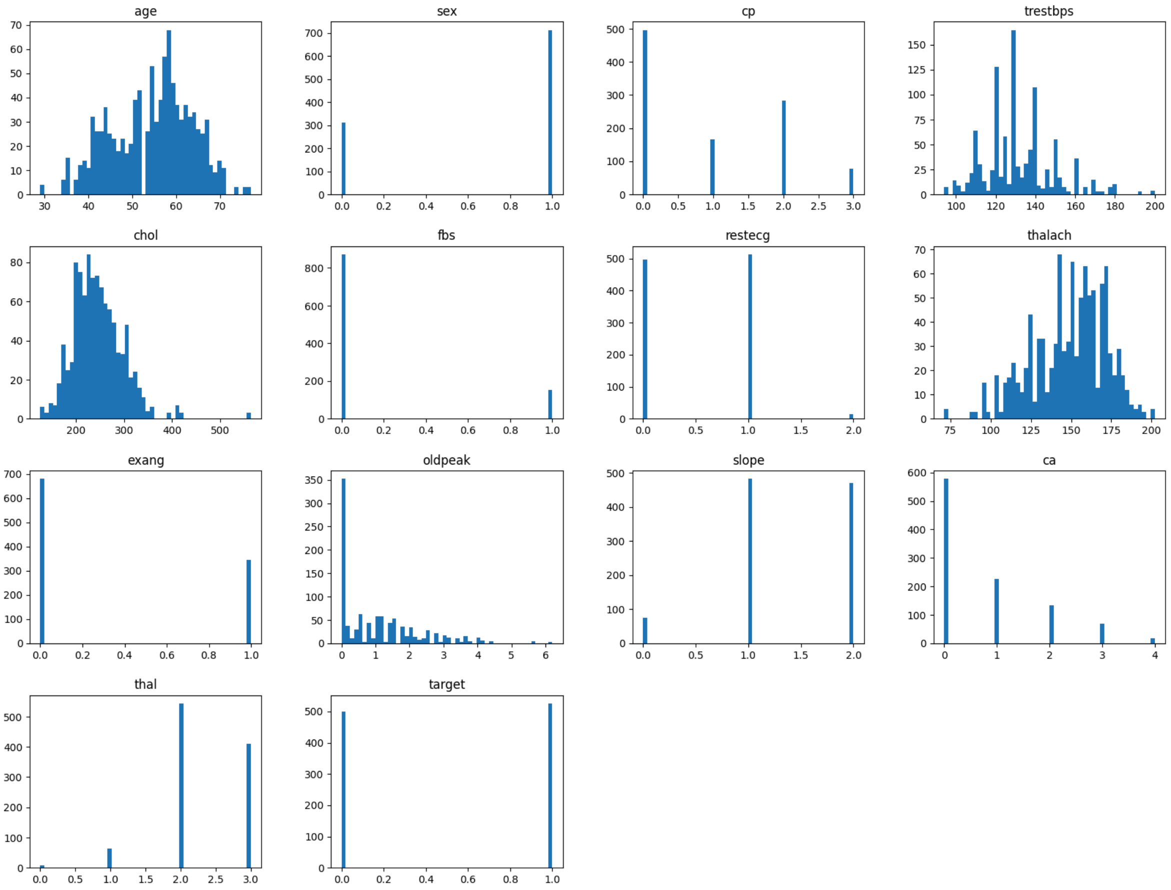
```
Out[ ]: age         0
        sex         0
        cp          0
        trestbps    0
        chol        0
        fbs         0
        restecg     0
        thalach     0
        exang       0
        oldpeak     0
        slope       0
        ca          0
        thal        0
        target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

```
In [ ]: # concise summary of our dataset
        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
```

```
In [ ]: #plotting histogram of all numeric values
        df.hist(bins=50, grid= False, figsize = (20,15))
```

```
Out[ ]: 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)
```



In [ ]:
```python
# genetrate descriptive statistics
df.describe()
```

Out[ ]:

	age	sex	cp	trestbps	chol	fbs	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	10
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	

In [ ]:
```python
questions = ["1. How many people have heart disease and how many people doesn't
"2. People of which sex has most heart disease?",
"3. People of which sex has which type of chest pain most?",
"4. People with which chest pain are most pron to have heart disease?"]

questions
```

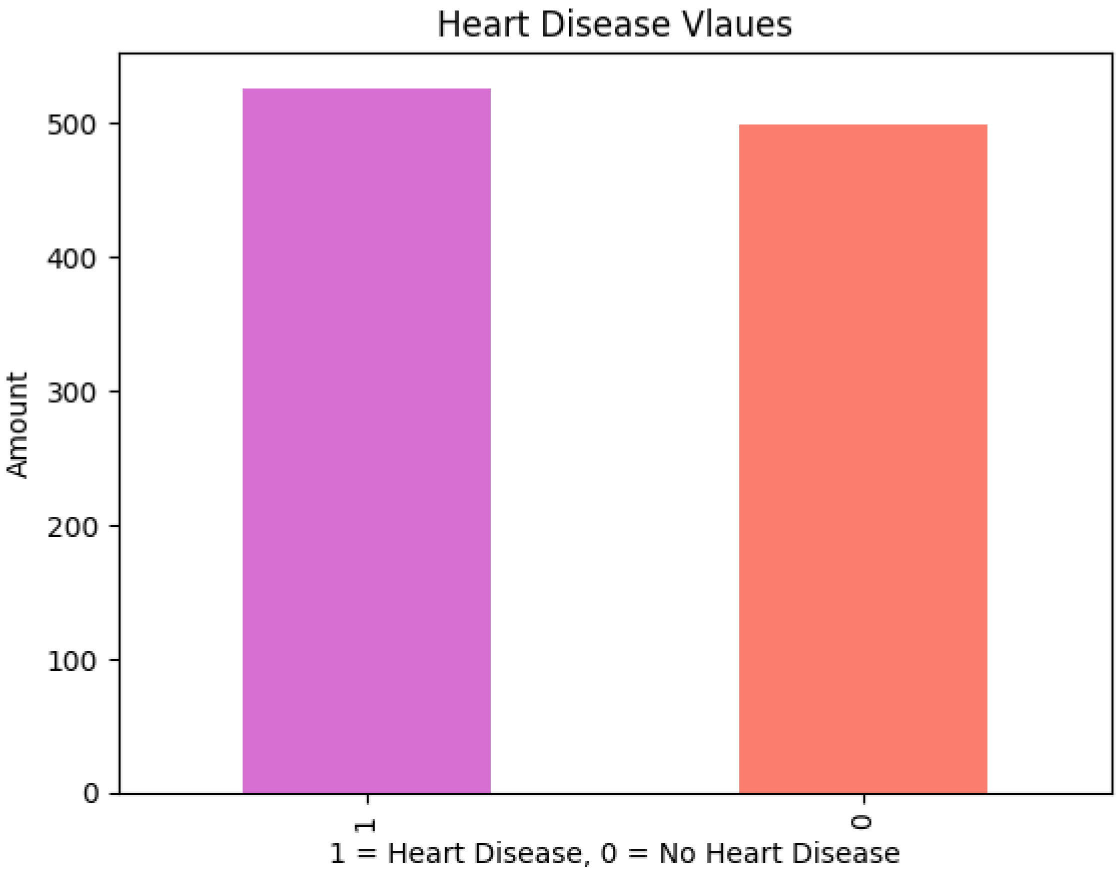
Out[ ]:
```
["1. How many people have heart disease and how many people doesn't have heart
disease? ",
 '2. People of which sex has most heart disease?',
 '3. People of which sex has which type of chest pain most?',
 '4. People with which chest pain are most pron to have heart disease?']
```

In [ ]:
```
#1. How many people have heart disease and how many people doesn't have heart di

df.target.value_counts()
```

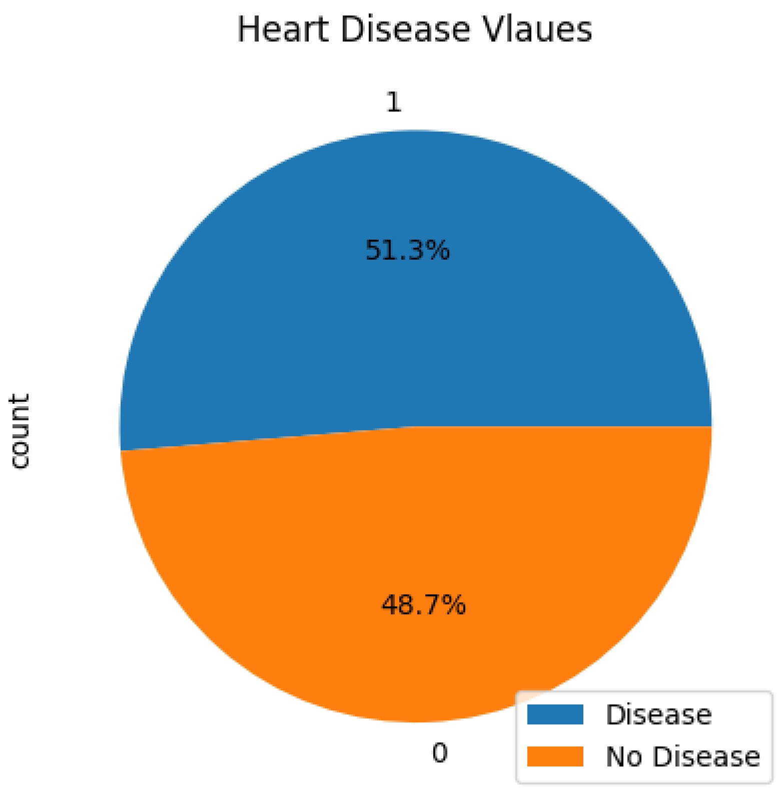
Out[ ]:
```
target
1    526
0    499
Name: count, dtype: int64
```

In [ ]:
```
# plotting bar chart
df.target.value_counts().plot(kind = "bar", color=["orchid", "salmon"])
plt.title("Heart Disease Vlaues")
plt.xlabel("1 = Heart Disease, 0 = No Heart Disease")
plt.ylabel("Amount")
```

Out[ ]:
```
Text(0, 0.5, 'Amount')
```



In [ ]:
```
# plotting a pie chart
df.target.value_counts().plot(kind = "pie", autopct = "%.1f%%")
plt.title("Heart Disease Vlaues")
plt.legend(["Disease", "No Disease"])
plt.show()
```



In [ ]:

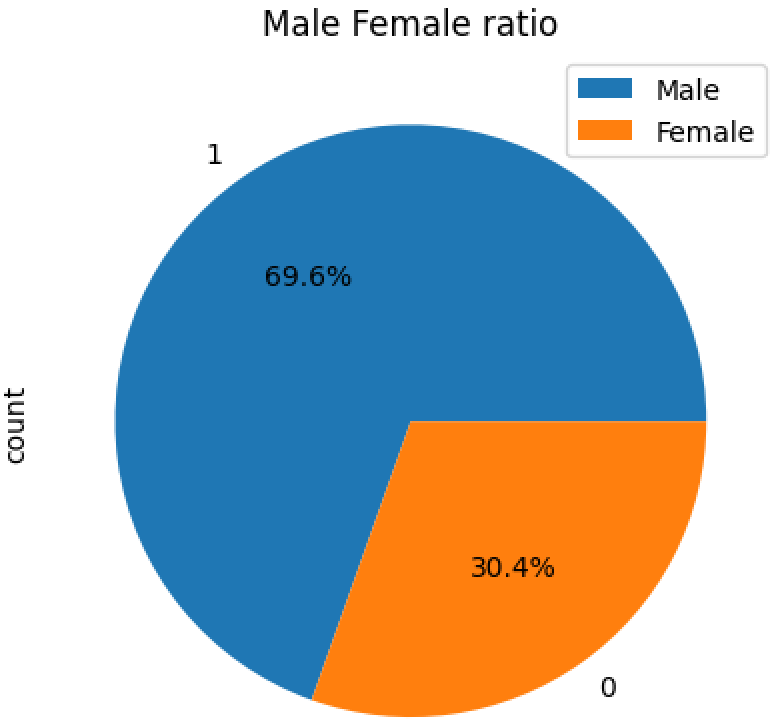
```
# '0' represent 'Female'
# '1' represent 'Male'
# SEX column part

# '0' represent 'No Disease'
# '1' represent 'Disease'
# Target column part

# Now let's check how many "Male" and "Female" are there in the dataset
df.sex.value_counts()
```

Out[ ]:

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

In [ ]:

```
#plotting a pie chart
df.sex.value_counts().plot(kind = "pie", autopct = "%.1f%%")
plt.title("Male Female ratio")
plt.legend(["Male", "Female"])
plt.show()
```



In [ ]:

```
# Let's find the answer of 2nd question.

# 2. People of which sex has most heart disease?

pd.crosstab(df.target, df.sex)
```

Out[ ]:

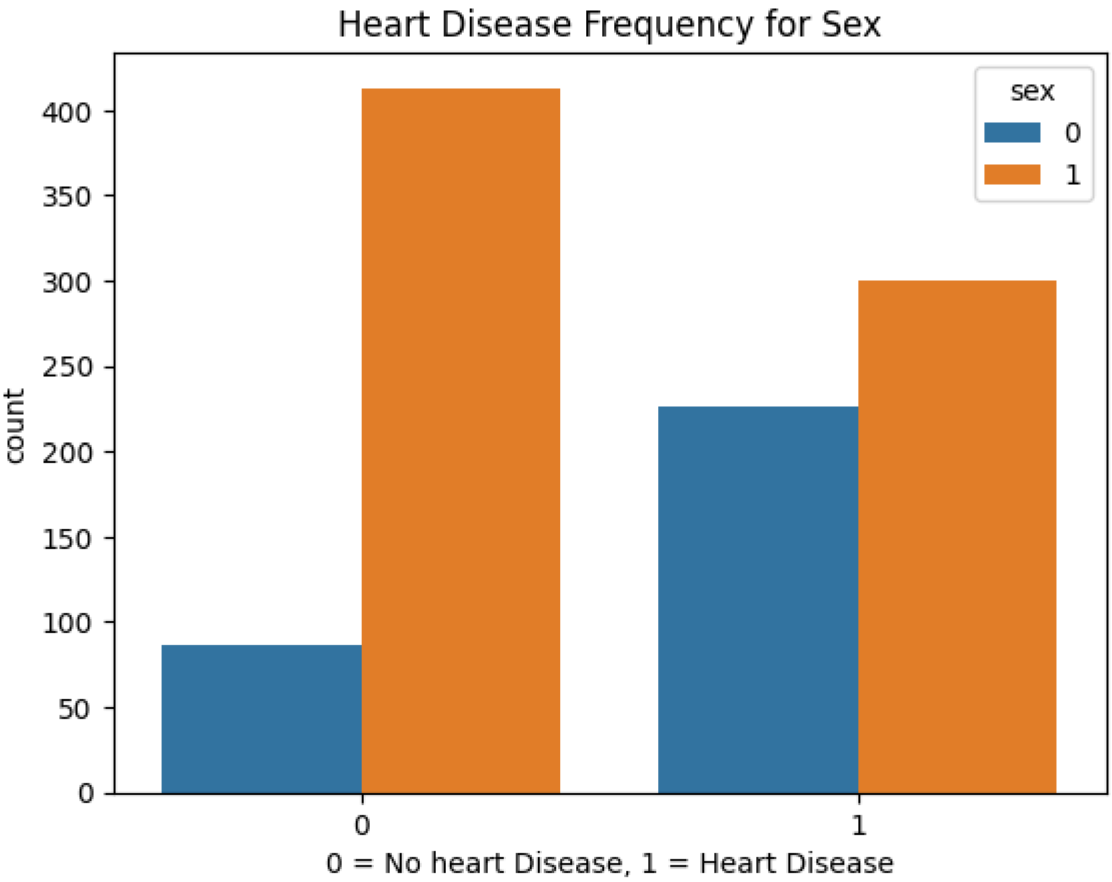
sex	0	1
target		
0	86	413
1	226	300

In [ ]:

```
sns.countplot(x = 'target', data = df, hue = "sex")
plt.title("Heart Disease Frequency for Sex")
plt.xlabel("0 = No heart Disease, 1 = Heart Disease")
```

Out[ ]:

```
Text(0.5, 0, '0 = No heart Disease, 1 = Heart Disease')
```



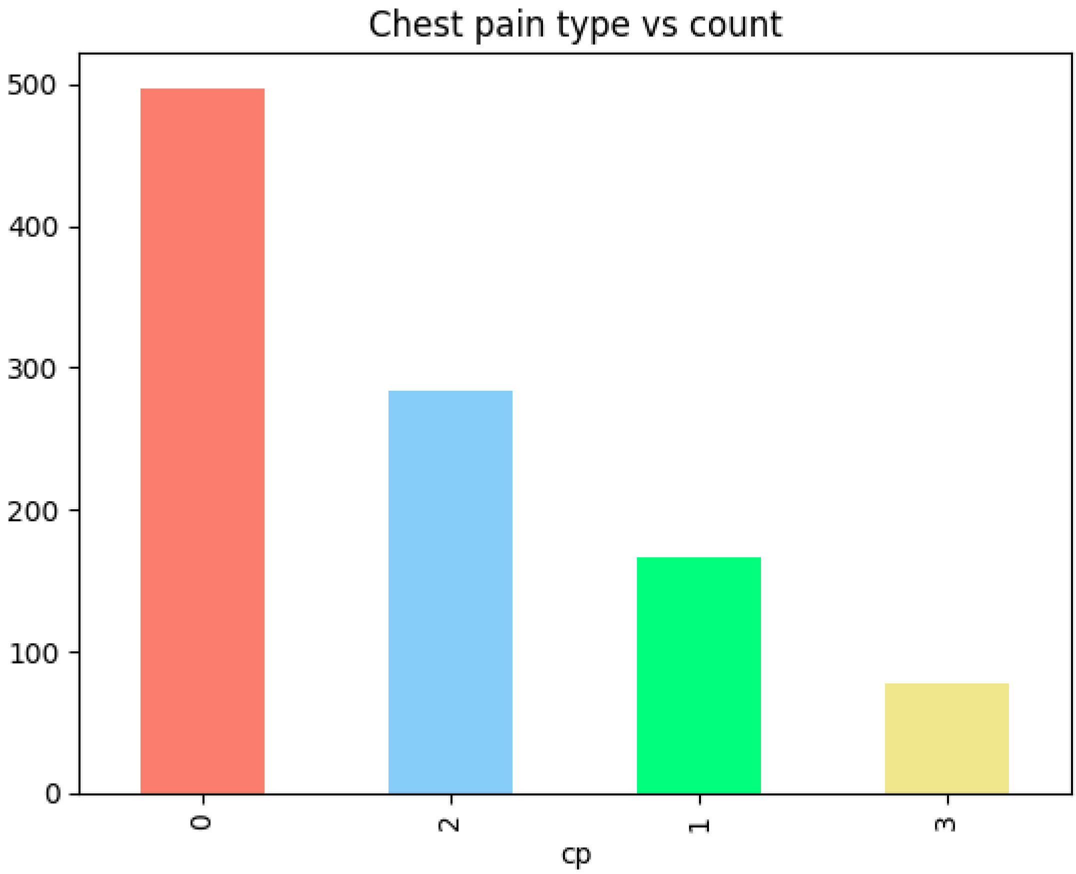
In [ ]:
```
# Number of male is more than double in our dataset than female

# More than "45% male" has heart disease and "75% female" has heart disease
```

In [ ]:
```
#Let's move to question 3
# 3. People of which sex has which type of chest pain most?

#counting values for different chest pain
df.cp.value_counts()
```

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

In [ ]:
```
# plotting a bar chart
df.cp.value_counts().plot(kind = "bar", color = ["salmon", "lightskyblue", "spri
plt.title("Chest pain type vs count")
```

Out[ ]:
```
Text(0.5, 1.0, 'Chest pain type vs count')
```



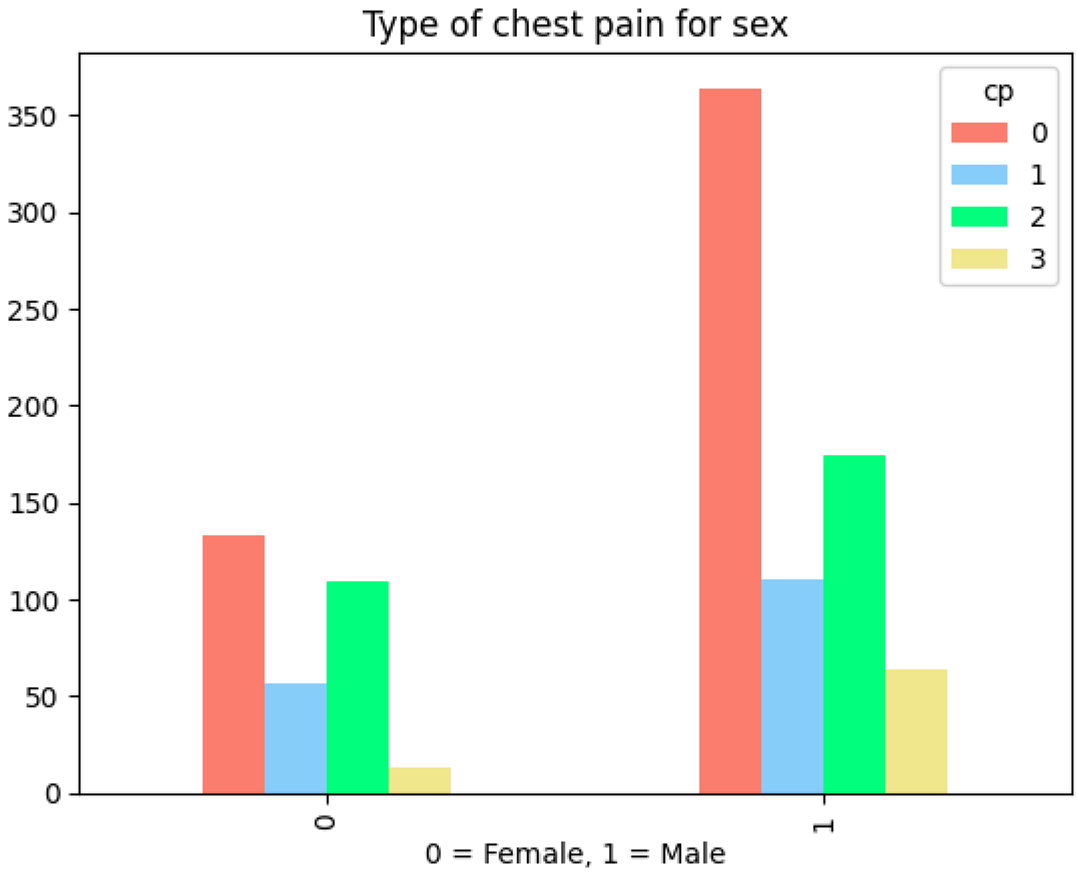
```
In [ ]: pd.crosstab(df.sex, df.cp)
```

Out[ ]:

cp	0	1	2	3
sex				
0	133	57	109	13
1	364	110	175	64

```
In [ ]: pd.crosstab(df.sex, df.cp).plot(kind = "bar", color = ["salmon", "lightskyblue",
        plt.title("Type of chest pain for sex")
        plt.xlabel("0 = Female, 1 = Male")
```

```
Out[ ]: Text(0.5, 0, '0 = Female, 1 = Male')
```



In [ ]:

```
# Most of the "male" has "type 0" chest pain and least of the "Male" has "type 4
# In case of "female" "type 0" and "type 2" percentage is almost same
```

In [ ]:

```
#Now question 4:

#4. People with which chest pain are most pron to have heart disease?

pd.crosstab(df.cp, df.target)
```

Out[ ]:

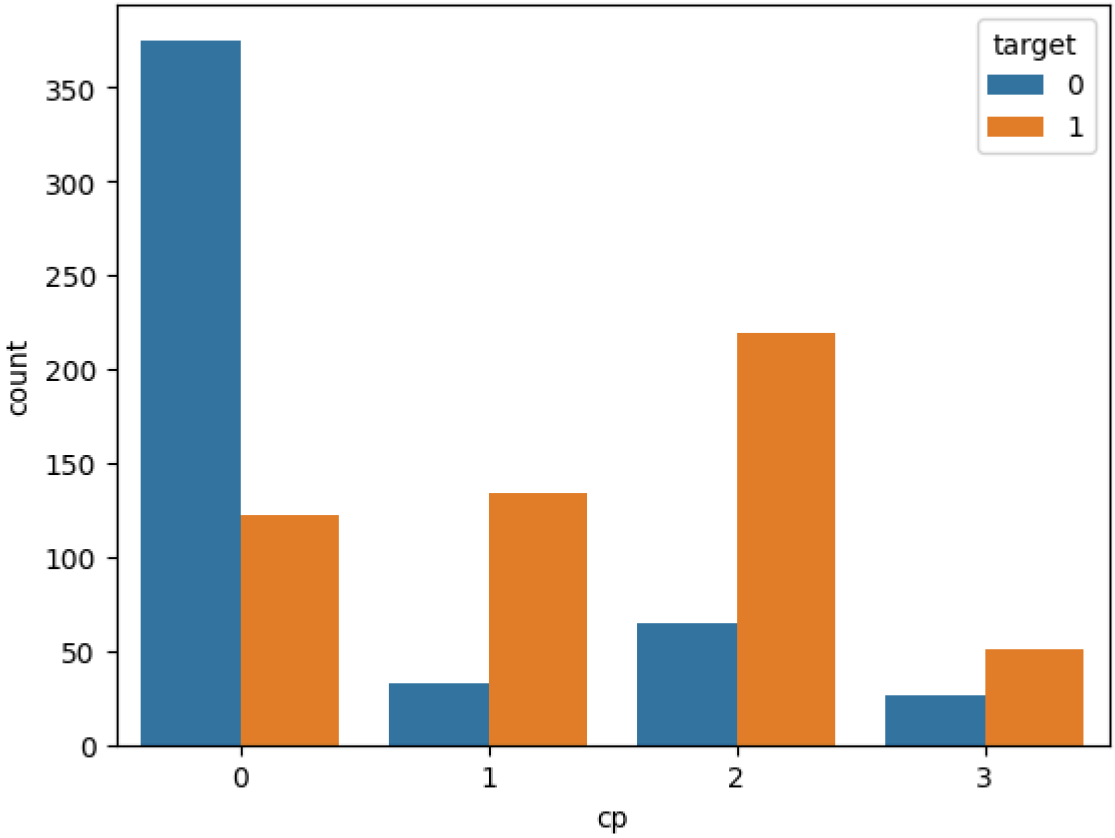
target	**0**	**1**
cp		
0	375	122
1	33	134
2	65	219
3	26	51

In [ ]:

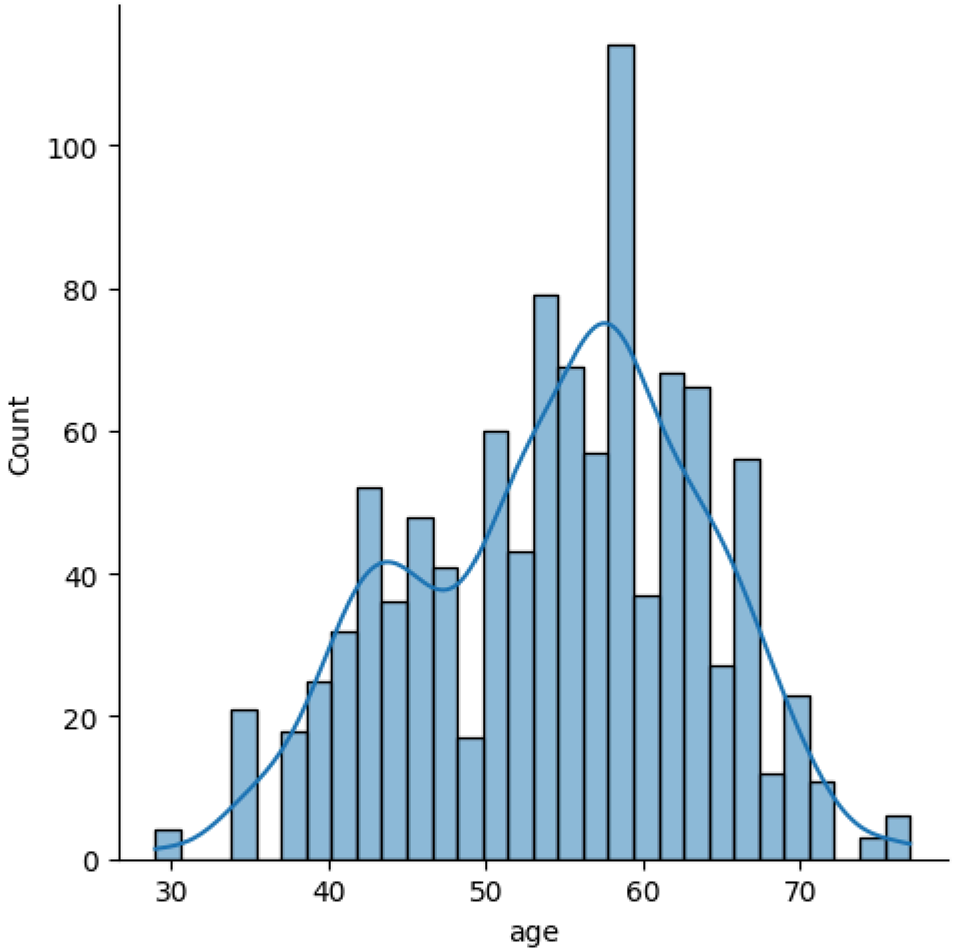
```
sns.countplot(x = "cp", data = df, hue = "target")
```

Out[ ]:

```
<Axes: xlabel='cp', ylabel='count'>
```



```
In [ ]: # most of the people who has "type 0" chest pain has less chance of heart diseas

        # And we see the opposite for other types.

        #Now Let's take a look at our age column

        # Create a distribution plot with normal distribution curve
        sns.displot(x = "age", data = df, bins = 30, kde = True)

Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1dd80139600>
```



In [ ]:
```
# From this plot we get a clear overview about Maximum heart rate represented by
```

In [ ]:
```
# Now lest do some more question
```

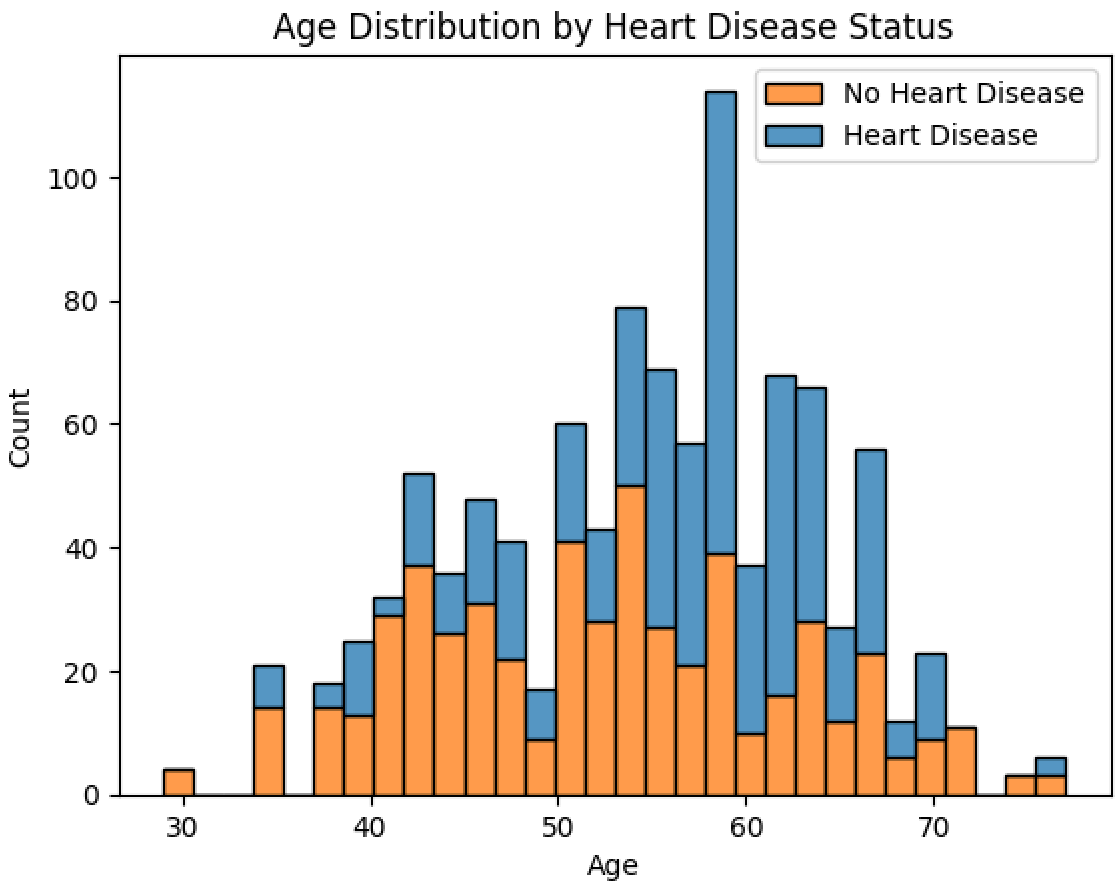
In [ ]:
```
more_questions = ["5. What is the distribution of age among people with and with
                  "6. How does cholesterol level vary between people with and wi
                  "7. What is the relationship between fasting blood sugar and h
                  "8. How does maximum heart rate (thalach) relate to heart dise
                  "9. What is the impact of exercise-induced angina on heart dis
                  "10. What is the distribution of resting blood pressure (trest


more_questions
```

Out[ ]:
```
['5. What is the distribution of age among people with and without heart diseas
e?',
 '6. How does cholesterol level vary between people with and without heart dise
ase?',
 '7. What is the relationship between fasting blood sugar and heart disease?',
 '8. How does maximum heart rate (thalach) relate to heart disease?',
 '9. What is the impact of exercise-induced angina on heart disease?',
 '10. What is the distribution of resting blood pressure (trestbps) in people w
ith and without heart disease?']
```

In [ ]:
```
# 5. What is the distribution of age among people with and without heart disease

# Distribution of age
age_distribution = df.groupby('target')['age'].describe()
print(age_distribution)
```

```
        count       mean       std   min   25%   50%   75%   max
target
0       499.0  56.569138  7.908153  35.0  52.0  58.0  62.0  77.0
1       526.0  52.408745  9.631804  29.0  44.0  52.0  59.0  76.0
```

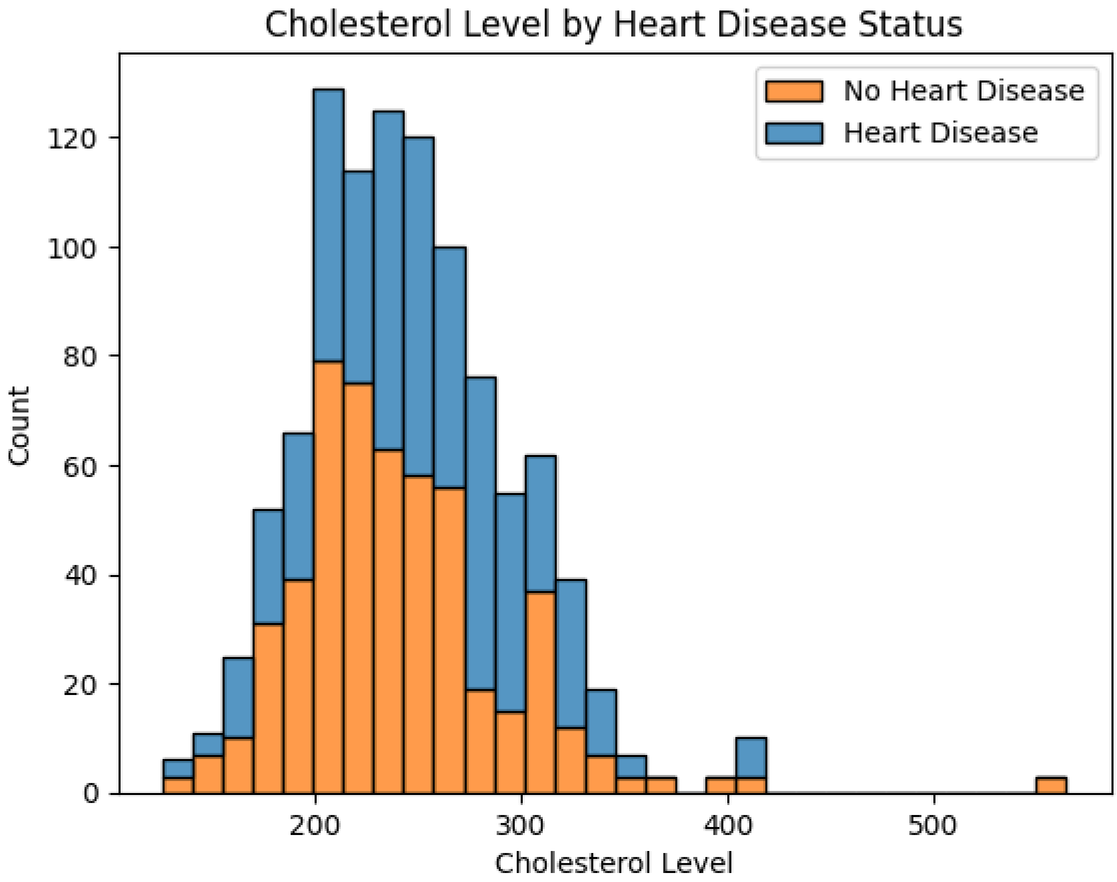
In [ ]:
```
# Distribution of age
sns.histplot(data=df, x='age', hue='target', multiple='stack', bins=30)
plt.title('Age Distribution by Heart Disease Status')
plt.xlabel('Age')
plt.ylabel('Count')
plt.legend(['No Heart Disease', 'Heart Disease'])
plt.show()
```



In [ ]:
```
# 6. How does cholesterol level vary between people with and without heart disea

# Cholesterol level distribution
cholesterol_distribution = df.groupby('target')['chol'].describe()
print(cholesterol_distribution)
```

```
        count        mean        std    min    25%    50%     75%    max
target
0       499.0  251.292585  49.558924  131.0  217.0  249.0  284.00  409.0
1       526.0  240.979087  53.010345  126.0  208.0  234.0  265.75  564.0
```

In [ ]:
```
# Cholesterol level distribution
sns.histplot(data=df, x='chol', hue='target', multiple='stack', bins=30)
plt.title('Cholesterol Level by Heart Disease Status')
plt.xlabel('Cholesterol Level')
plt.ylabel('Count')
plt.legend(['No Heart Disease', 'Heart Disease'])
plt.show()
```



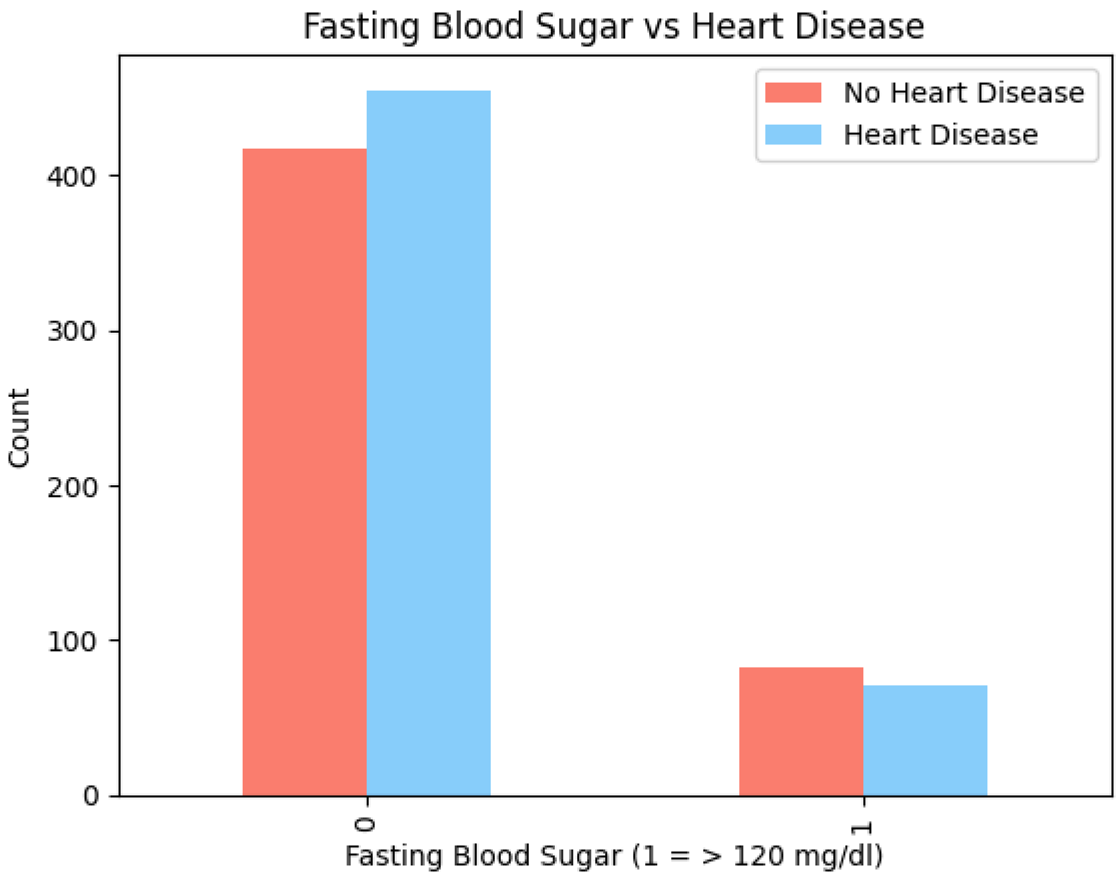
In [ ]:

```
# 7. What is the relationship between fasting blood sugar and heart disease?

# Fasting blood sugar vs heart disease
fbs_relationship = pd.crosstab(df.fbs, df.target)
print(fbs_relationship)
```

```
target    0    1
fbs
0       417  455
1        82   71
```

In [ ]:

```
# Fasting blood sugar vs heart disease
pd.crosstab(df.fbs, df.target).plot(kind="bar", color=["salmon", "lightskyblue"]
plt.title('Fasting Blood Sugar vs Heart Disease')
plt.xlabel('Fasting Blood Sugar (1 = > 120 mg/dl)')
plt.ylabel('Count')
plt.legend(['No Heart Disease', 'Heart Disease'])
plt.show()
```



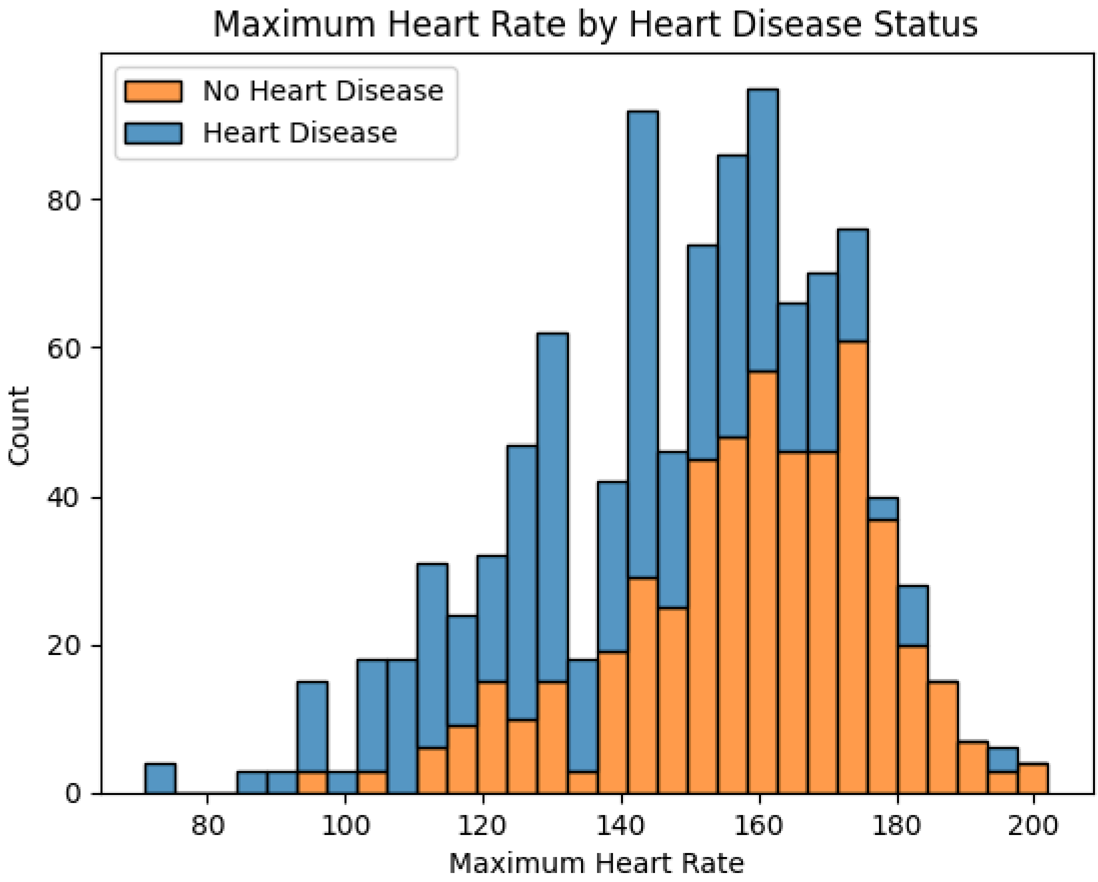
In [ ]:

```
# 8. How does maximum heart rate (thalach) relate to heart disease?

# Maximum heart rate distribution
thalach_distribution = df.groupby('target')['thalach'].describe()
print(thalach_distribution)
```

```
        count        mean        std   min    25%    50%    75%    max
target
0       499.0  139.130261  22.565235  71.0  125.0  142.0  156.0  195.0
1       526.0  158.585551  19.096928  96.0  149.0  161.5  172.0  202.0
```

In [ ]:

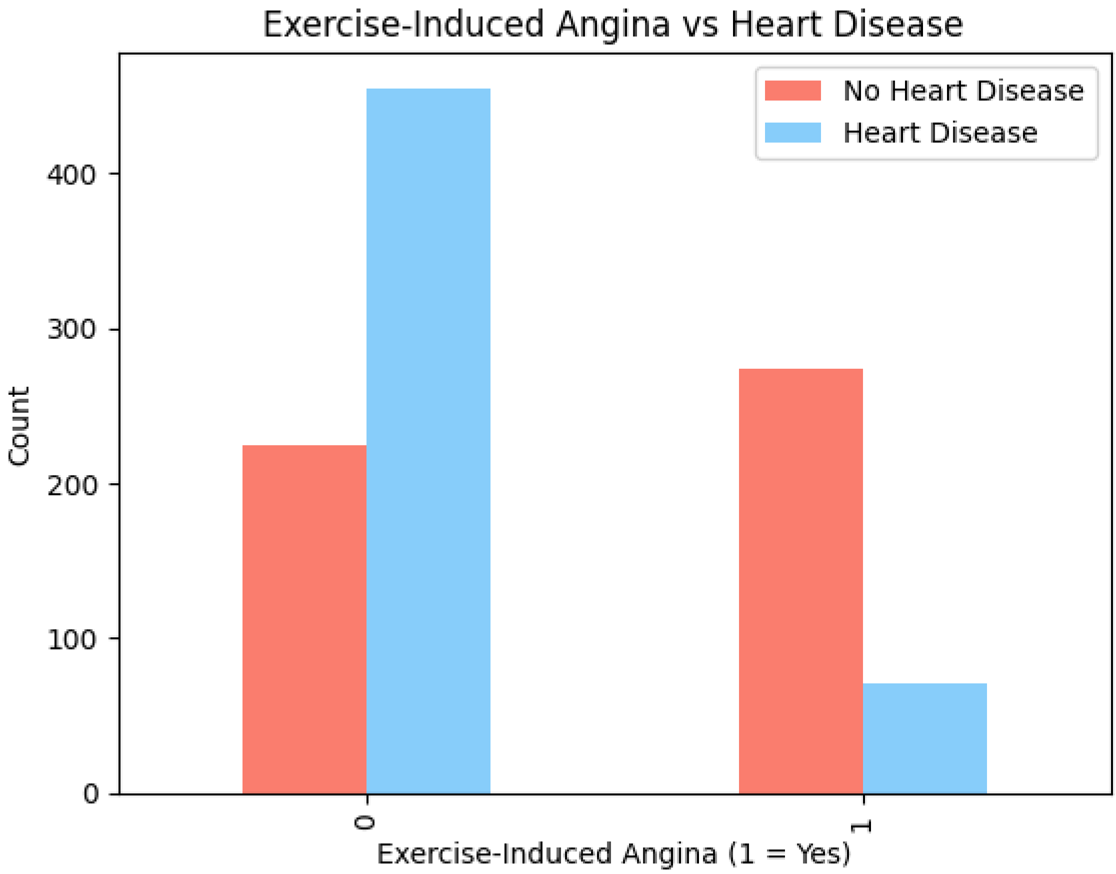
```
# Maximum heart rate distribution
sns.histplot(data=df, x='thalach', hue='target', multiple='stack', bins=30)
plt.title('Maximum Heart Rate by Heart Disease Status')
plt.xlabel('Maximum Heart Rate')
plt.ylabel('Count')
plt.legend(['No Heart Disease', 'Heart Disease'])
plt.show()
```



```
In [ ]: # 9. What is the impact of exercise-induced angina on heart disease?

        # Exercise-induced angina vs heart disease
        exang_relationship = pd.crosstab(df.exang, df.target)
        print(exang_relationship)
```

```
target    0    1
exang           
0       225  455
1       274   71
```

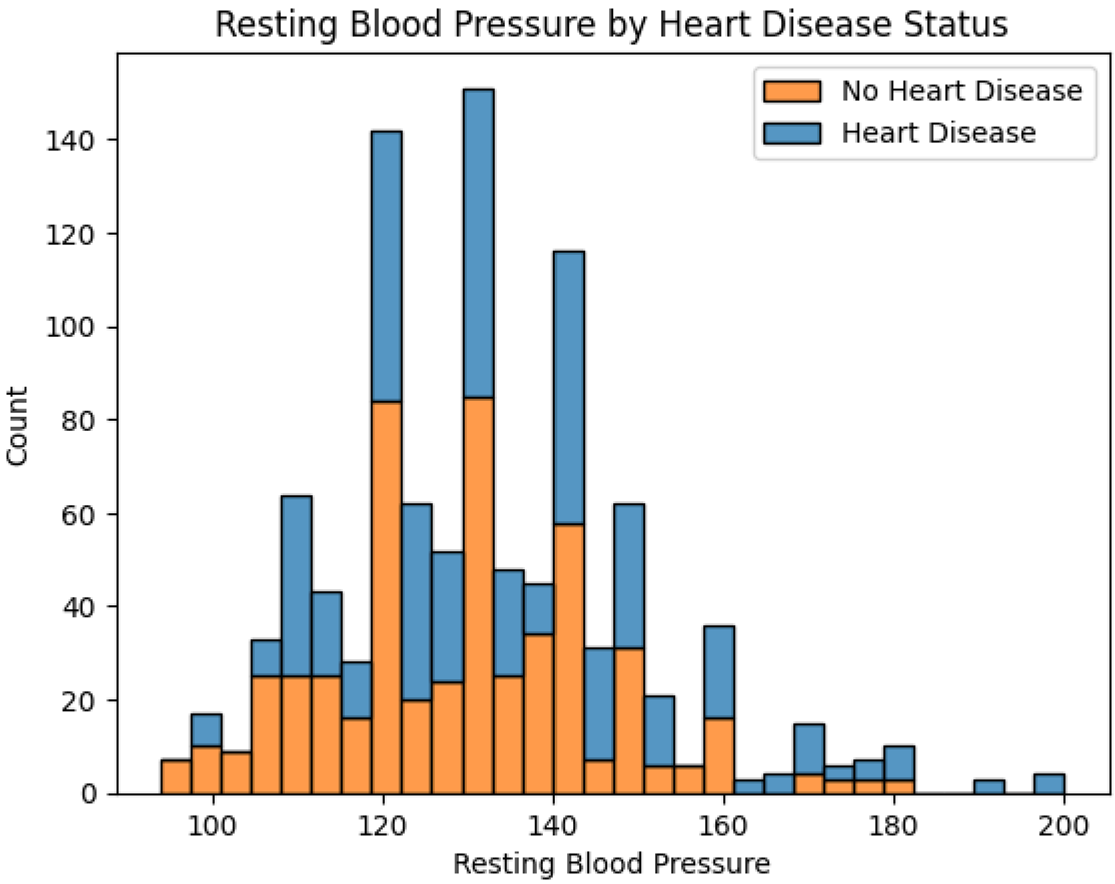
```
In [ ]: # Exercise-induced angina vs heart disease
        pd.crosstab(df.exang, df.target).plot(kind="bar", color=["salmon", "lightskyblue
        plt.title('Exercise-Induced Angina vs Heart Disease')
        plt.xlabel('Exercise-Induced Angina (1 = Yes)')
        plt.ylabel('Count')
        plt.legend(['No Heart Disease', 'Heart Disease'])
        plt.show()
```



In [ ]:
```
# 10. What is the distribution of resting blood pressure (trestbps) in people wi

# Resting blood pressure distribution
trestbps_distribution = df.groupby('target')['trestbps'].describe()
print(trestbps_distribution)
```

```
        count        mean        std    min    25%    50%    75%    max
target
0       499.0  134.106212  18.576736  100.0  120.0  130.0  144.0  200.0
1       526.0  129.245247  16.112188   94.0  120.0  130.0  140.0  180.0
```

In [ ]:
```
# Resting blood pressure distribution
sns.histplot(data=df, x='trestbps', hue='target', multiple='stack', bins=30)
plt.title('Resting Blood Pressure by Heart Disease Status')
plt.xlabel('Resting Blood Pressure')
plt.ylabel('Count')
plt.legend(['No Heart Disease', 'Heart Disease'])
plt.show()
```


Resting Blood Pressure by Heart Disease Status
No Heart Disease
Heart Disease
Count
0
20
40
60
80
100
120
140
100
120
140
160
180
200
Resting Blood Pressure