task5

July 31, 2024

[26]:
```
#Importing all the libraries that we need.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[27]:
```
#Importing our dataset.
df = pd.read_csv('C:\\Users\\hp\\Downloads\\heart.csv')
```

[28]:
```
#Checking first five rows by calling df.head()
df.head()
```

[28]:

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

[29]:
```
df.tail()
```

[29]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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				

```
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
```

[30]:
```
#Take a look at the columns names.
df.columns.values
```

[30]:
```
array(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
       'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype=object)
```

[31]:
```
#Checking for null values
df.isna().sum()
```

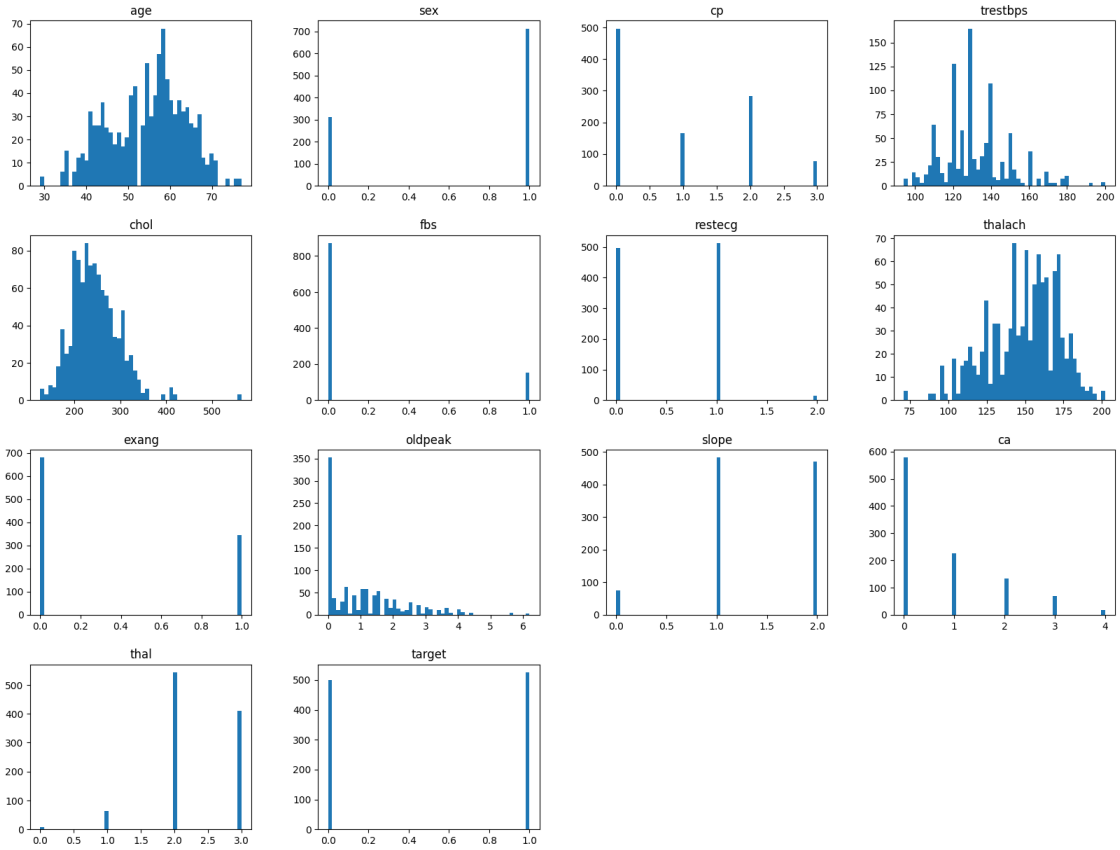
[31]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

[32]:
```
#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
```

[33]:
```
#Plotting histogram of all numeric values
df.hist(bins = 50, grid = False, figsize = (20,15));
```



[34]:
```
#Genrating descriptive statistics.
df.describe()
```

[34]:

	age	sex	cp	trestbps	chol	\
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	
mean	54.434146	0.695610	0.942439	131.611707	246.00000	
std	9.072290	0.460373	1.029641	17.516718	51.59251	
min	29.000000	0.000000	0.000000	94.000000	126.00000	

```
25%      48.000000     0.000000     0.000000   120.000000   211.00000
50%      56.000000     1.000000     1.000000   130.000000   240.00000
75%      61.000000     1.000000     2.000000   140.000000   275.00000
max      77.000000     1.000000     3.000000   200.000000   564.00000

               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
```

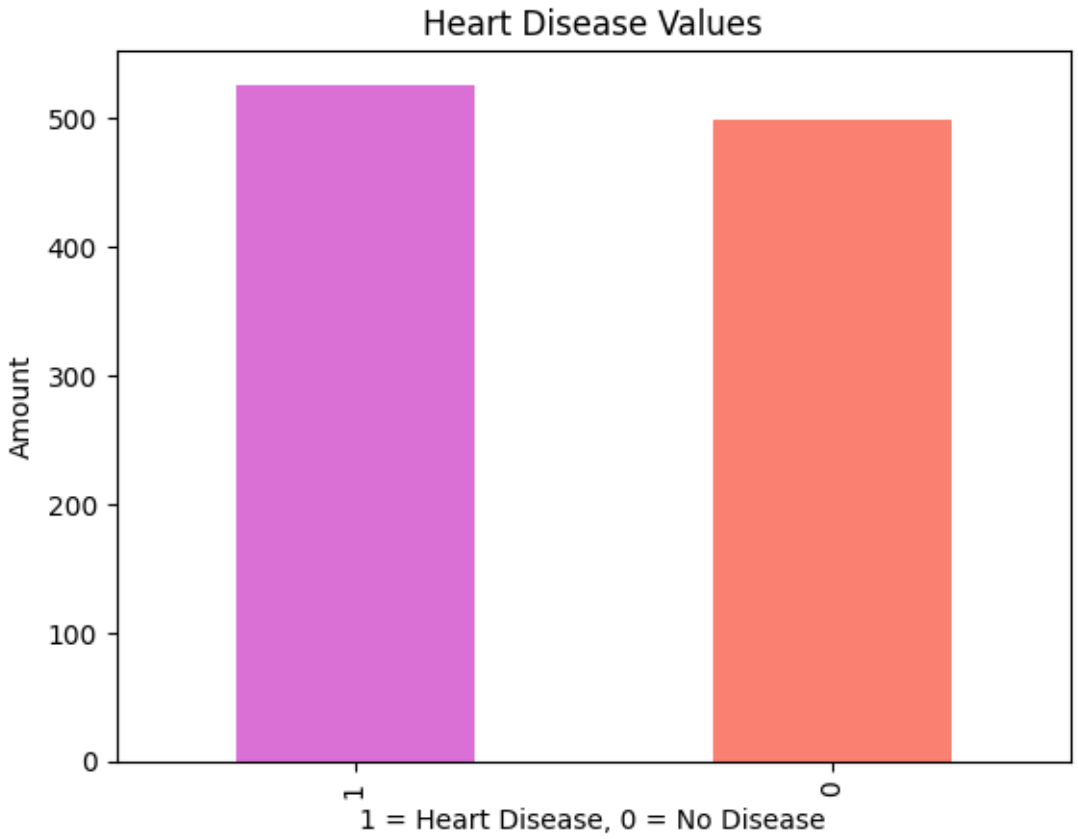
[35]:
```
questions = ["1. How many people have heart disease and how many people doesn't␣
 ↪have heart disease?",
             "2. People of which sex has most heart disease?",
             "3. People of which sex has which type of chest pain most?",
             "4. People with which chest pain are most pron to have heart␣
 ↪disease?"]
questions
```

[35]:
```
["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?']
```

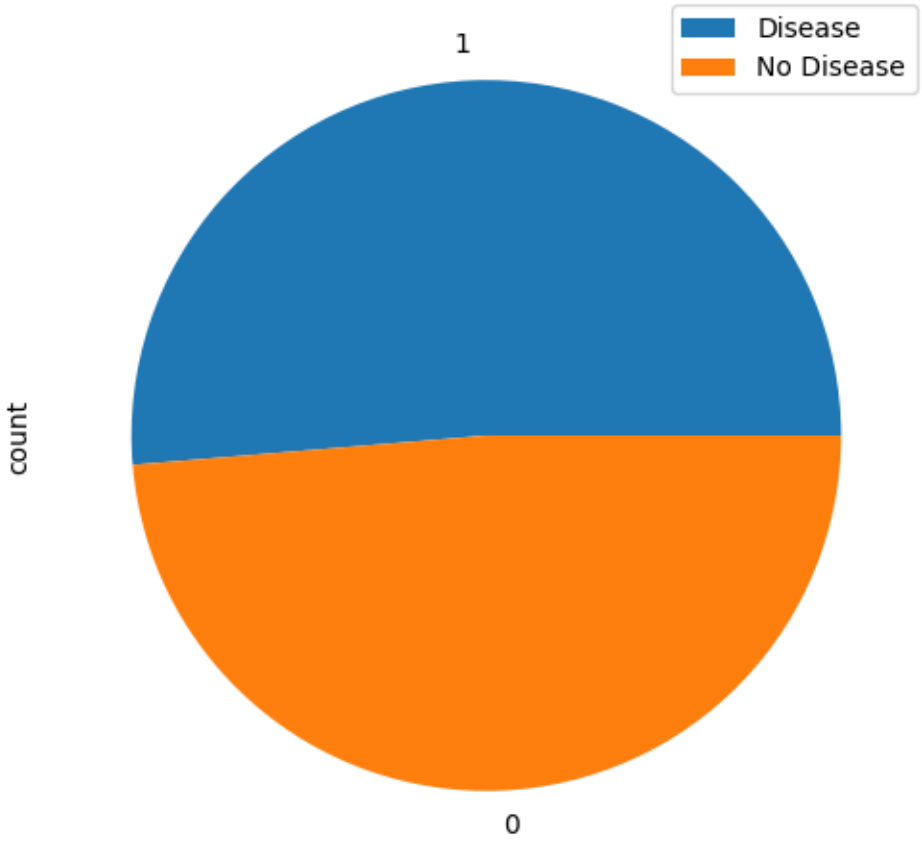
[36]:
```
#Lets find the answer of first question.

#1.How many people have heart disease and how many people doesn't have heart␣
 ↪disease?

#Getting the Values
df.target.value_counts()
```

[36]:
```
target
1    526
0    499
Name: count, dtype: int64
```

[37]:
```
#Plotting bar chart
df.target.value_counts().plot(kind = 'bar', color = ["orchid", "salmon"])
plt.title("Heart Disease Values")
plt.xlabel("1 = Heart Disease, 0 = No Disease")
plt.ylabel("Amount");
```



[38]:
```
#Plotting a  pie chart
df.target.value_counts().plot(kind = 'pie',  figsize = (8,6))
plt.legend(["Disease", "No Disease"]);
```

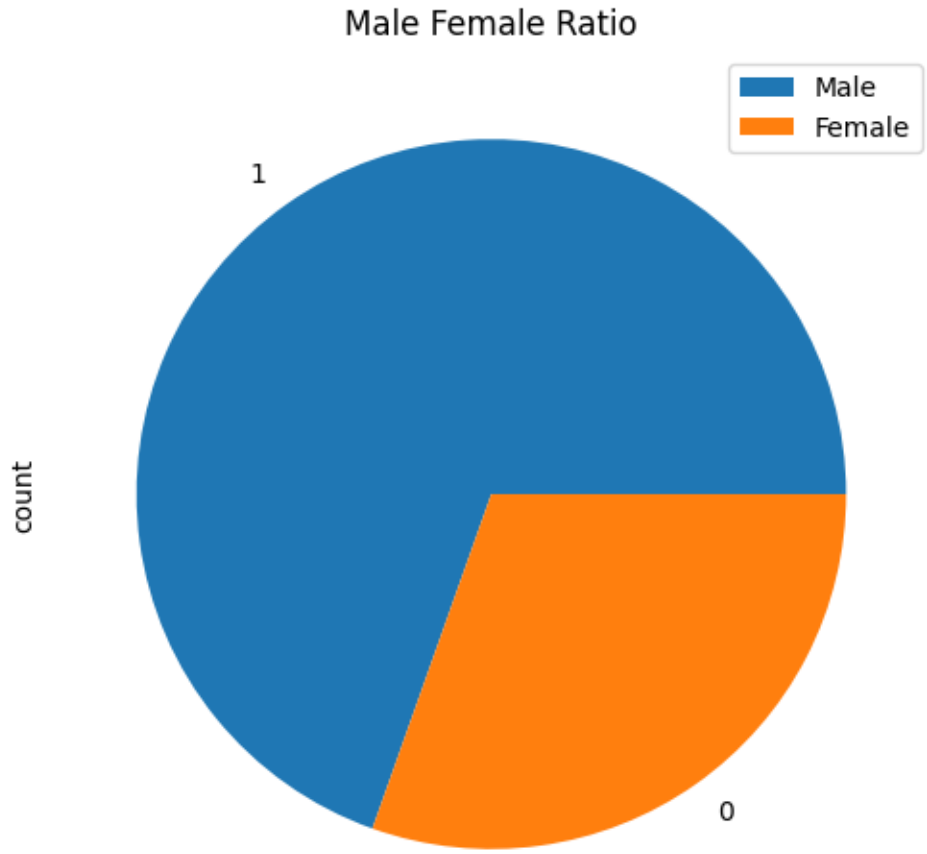


[39]:
```
#Sex column part
#0 represent female
#1 represent male

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

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

[39]:
```
sex
1    713
0    312
Name: count, dtype: int64
```

[40]:
```
#Plotting a pie chart
df.sex.value_counts().plot(kind = 'pie', figsize = (8,6))
```

```
plt.title("Male Female Ratio")
plt.legend(['Male', 'Female']);
```

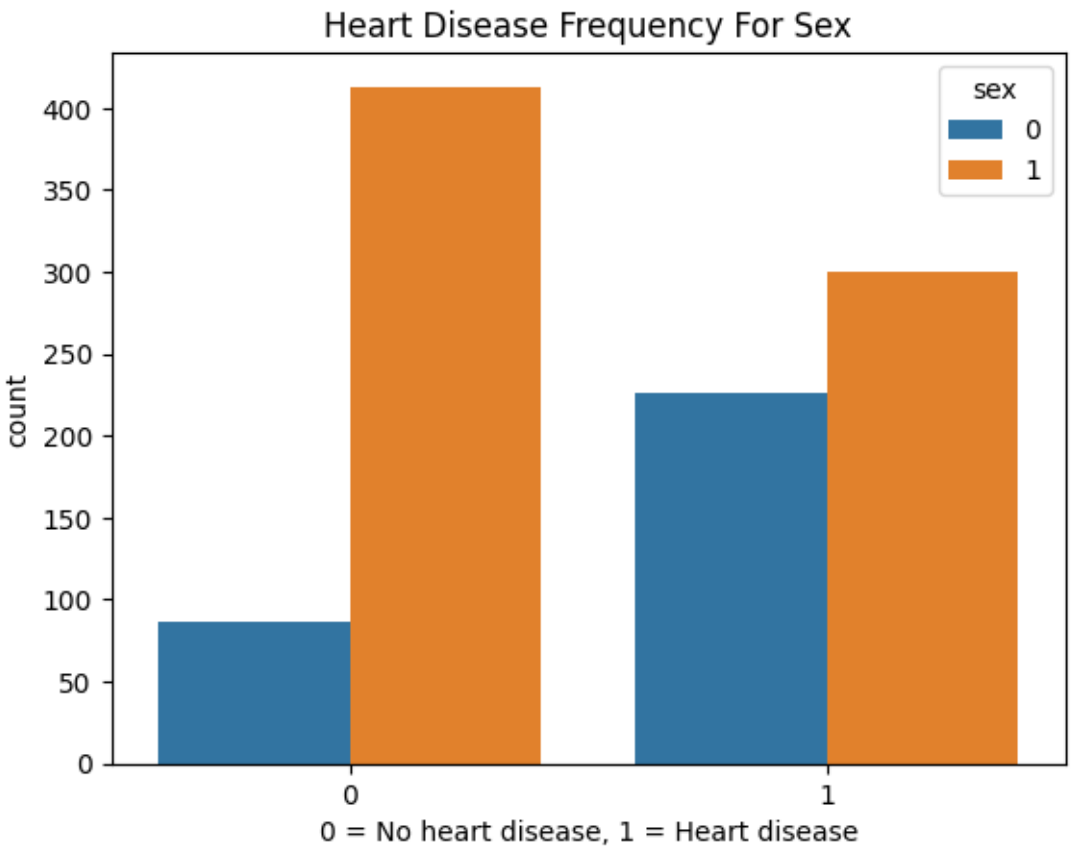


[41]:
```
#Lets's find the answer of our 2nd question
#2. People of which sex has most heart disease?
pd.crosstab(df.target, df.sex)
```

[41]:

sex	0	1
target		
0	86	413
1	226	300

[42]:
```
sns.countplot(x = 'target', data = df, hue = 'sex')
plt.title("Heart Disease Frequency For Sex")
plt.xlabel("0 = No heart disease, 1 = Heart disease");
```

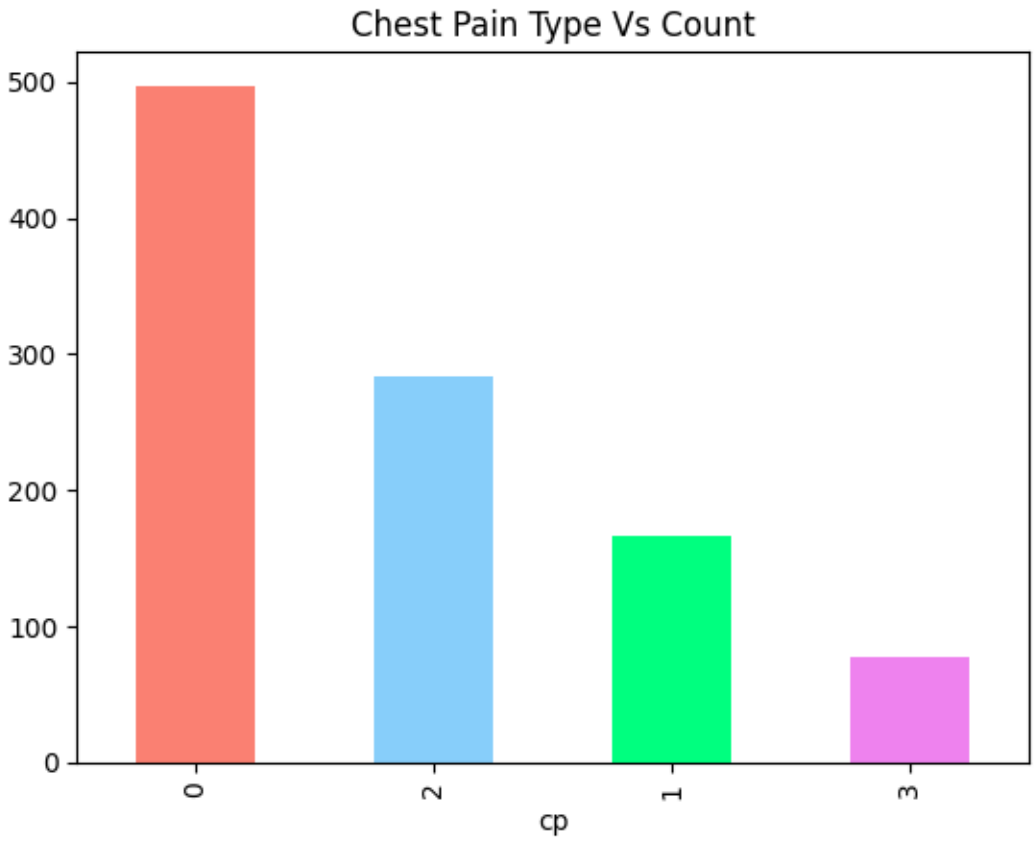


[43]:
```
#Number of male is more than double in our dataset than female

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

[44]:
```
#let's move to the 3rd question

#3. People of which sex has which type of chest pain most?

#Counting values for different chest pain

df.cp.value_counts()
```

[44]:
```
cp
0    497
2    284
1    167
3     77
Name: count, dtype: int64
```

[46]:
```
#Plotting a bar chart
df.cp.value_counts().plot(kind = 'bar', color = ['salmon', 'lightskyblue',␣
 ↪'springgreen', 'violet'])
plt.title("Chest Pain Type Vs Count");
```

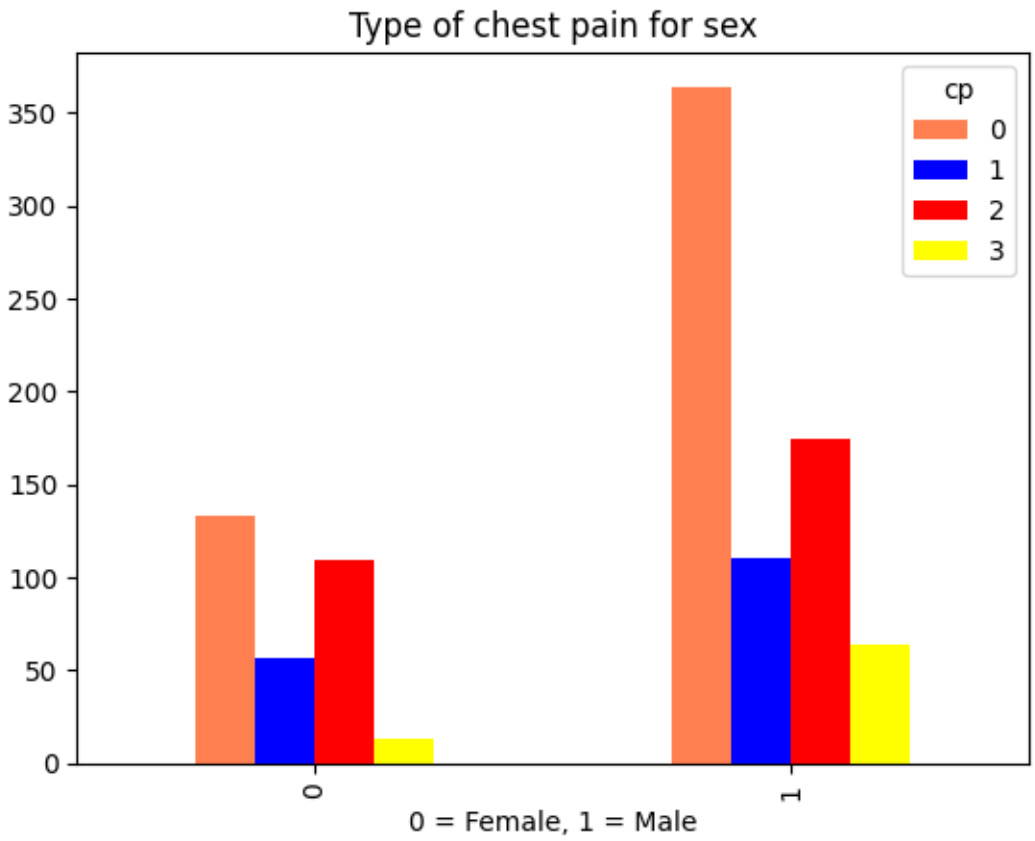


[47]:
```
pd.crosstab(df.sex, df.cp)
```

[47]:
```
cp      0    1    2   3
sex
0     133   57  109  13
1     364  110  175  64
```

[48]:
```
pd.crosstab(df.sex, df.cp).plot(kind = 'bar', color = ['coral', 'blue', 'red',␣
 ↪'yellow'])
plt.title("Type of chest pain for sex")
plt.xlabel('0 = Female, 1 = Male')
```

[48]:
```
Text(0.5, 0, '0 = Female, 1 = Male')
```

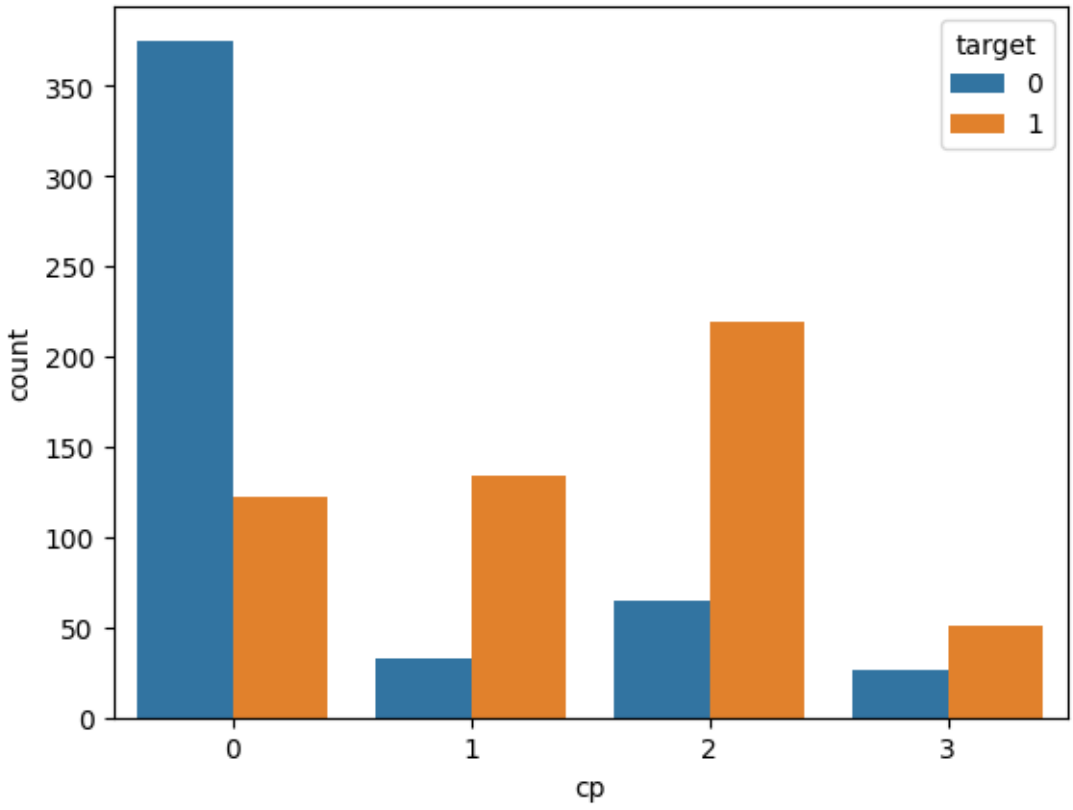


[ ]:
```
#Most of male has type 0 chest pain and least of male has type 4 pain.

#Incase of female type 0 and type 2 percentage is almost same.
```

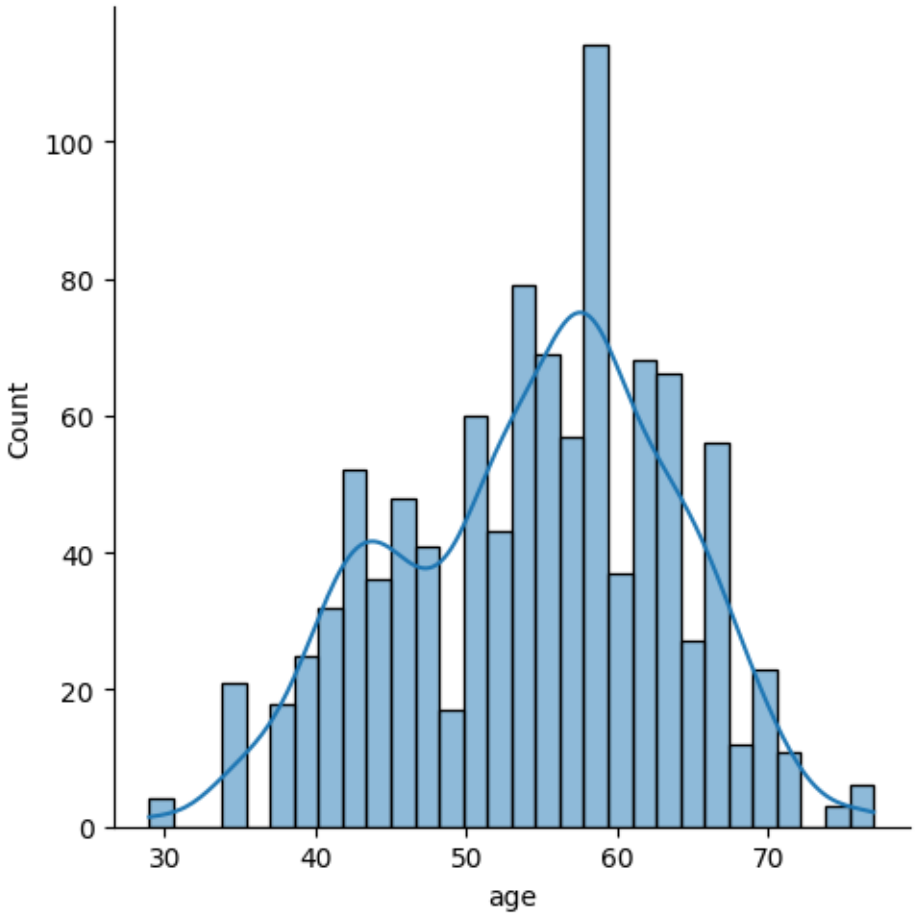
[49]:
```
#let's move to the 4th question
#4. People with which chest pain are most pron to have heart disease?
pd.crosstab(df.cp, df.target)
```

[49]:
```
target    0    1
cp
0       375  122
1        33  134
2        65  219
3        26   51
```

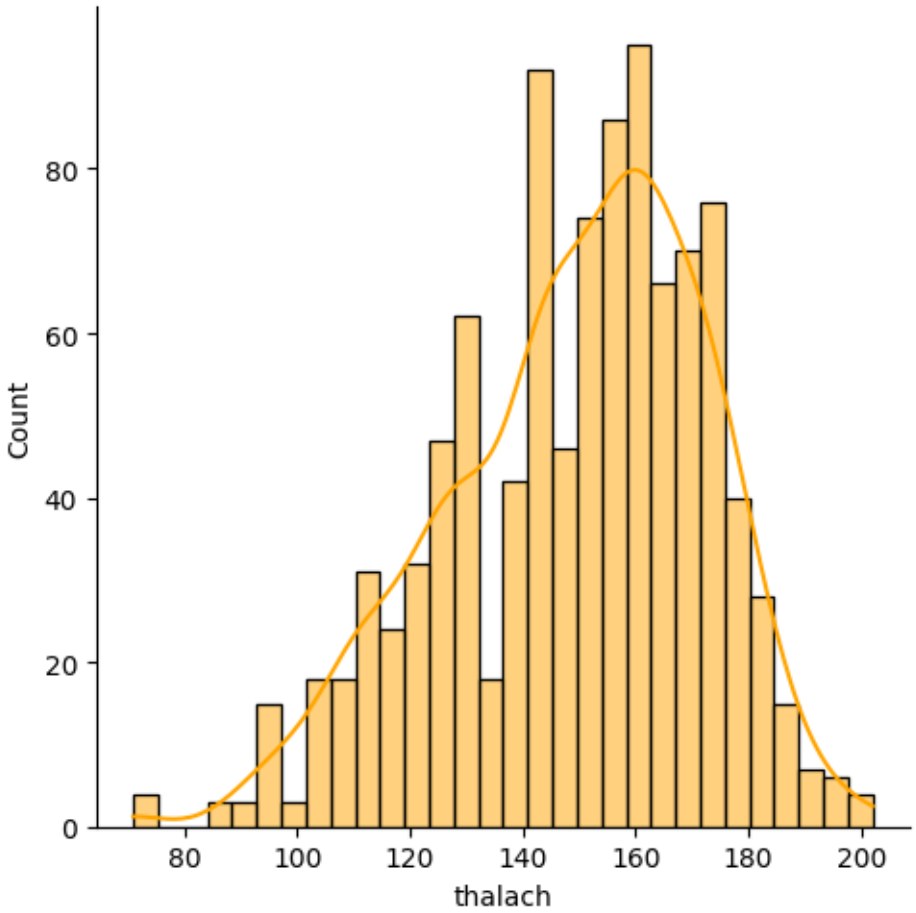
[51]:
```
sns.countplot(x = 'cp', data = df, hue = 'target');
```



[52]:
```
#Most of the people who has type 0 chest pain has less chance of heart disease.

#And we see the opposite for other types.

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

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



[53]:
```
# 58-59 year old people are most in the dataset

#Let's plot another distribution plot for 'Maximum heart rate'
sns.displot(x = 'thalach', data = df, bins = 30, kde = True, color = 'orange');
```



[ ]:
```
#From this plot we get a clear overview above Maximum heart rate represented by␣
↪thalach
```