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

In [11]:
```
df = pd.read_csv("heart.csv")
```

In [12]:
```
df.head()
```

Out[12]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	tar
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 [13]:
```
df.tail()
```

Out[13]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3

In [14]:
```
df.columns.values
```

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

In [15]:
```
df.isna().sum()
```

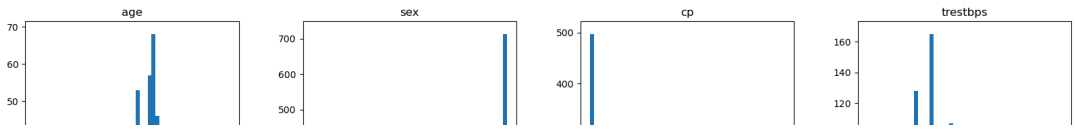
Out[15]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [16]:
```
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 [17]:
```
df.hist(bins=50, grid=False,figsize=(20,25))
```

Out[17]:
```
array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'se
x'}>,
        <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)
```



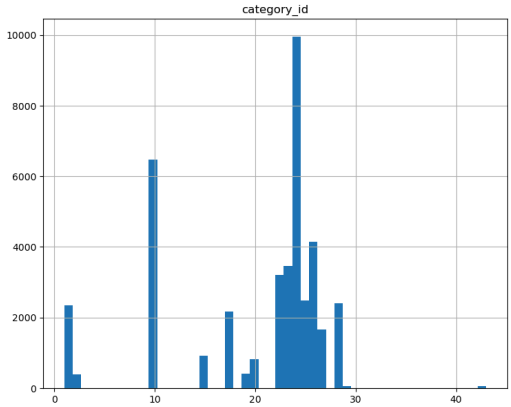
In [18]:
```
df.describe()
```

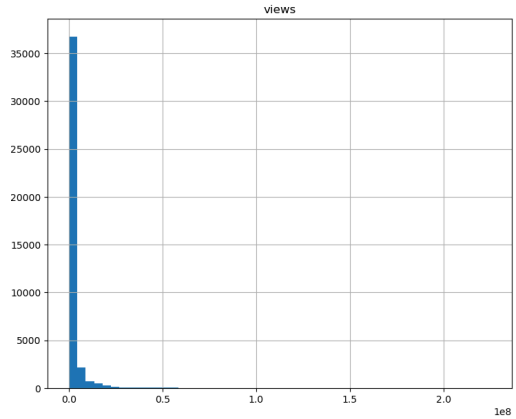
Out[18]:

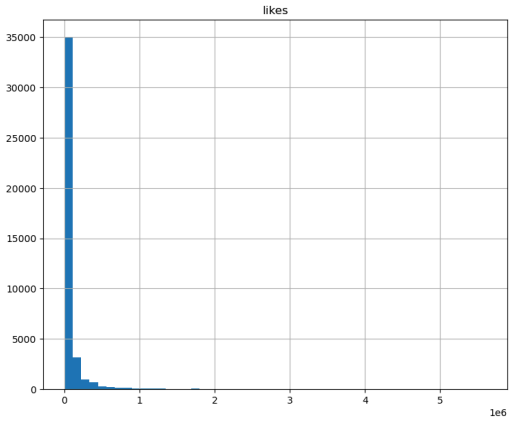
	age	sex	cp	trestbps	chol	fbs	r
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.(
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.5
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.5
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.(
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.(
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.(
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.(
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.(

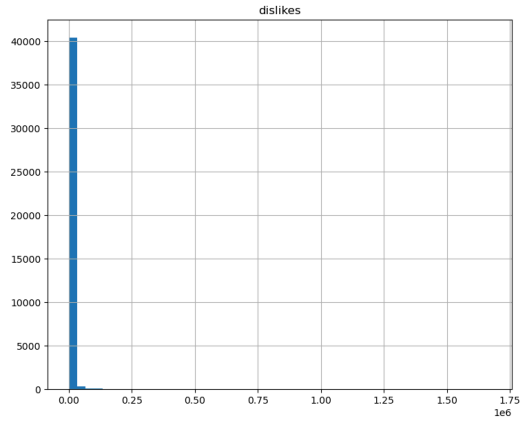
In [10]:
```
df.hist(bins=50, grid=True,figsize=(20,25))
```

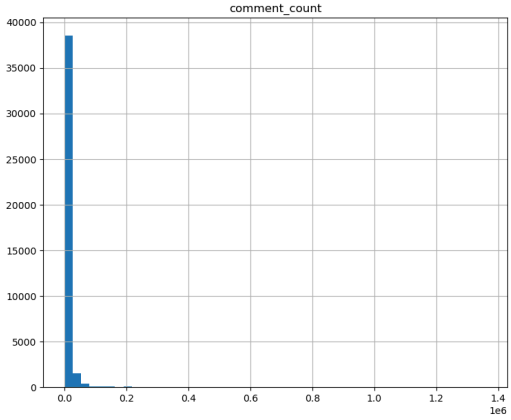
Out[10]:
```
array([[<Axes: title={'center': 'category_id'}>,
        <Axes: title={'center': 'views'}>],
       [<Axes: title={'center': 'likes'}>,
        <Axes: title={'center': 'dislikes'}>],
       [<Axes: title={'center': 'comment_count'}>, <Axes: >]],
      dtype=object)
```











In [19]:
```
questions = ["1. How many people have heart disease and how many people do
             "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


questions
```

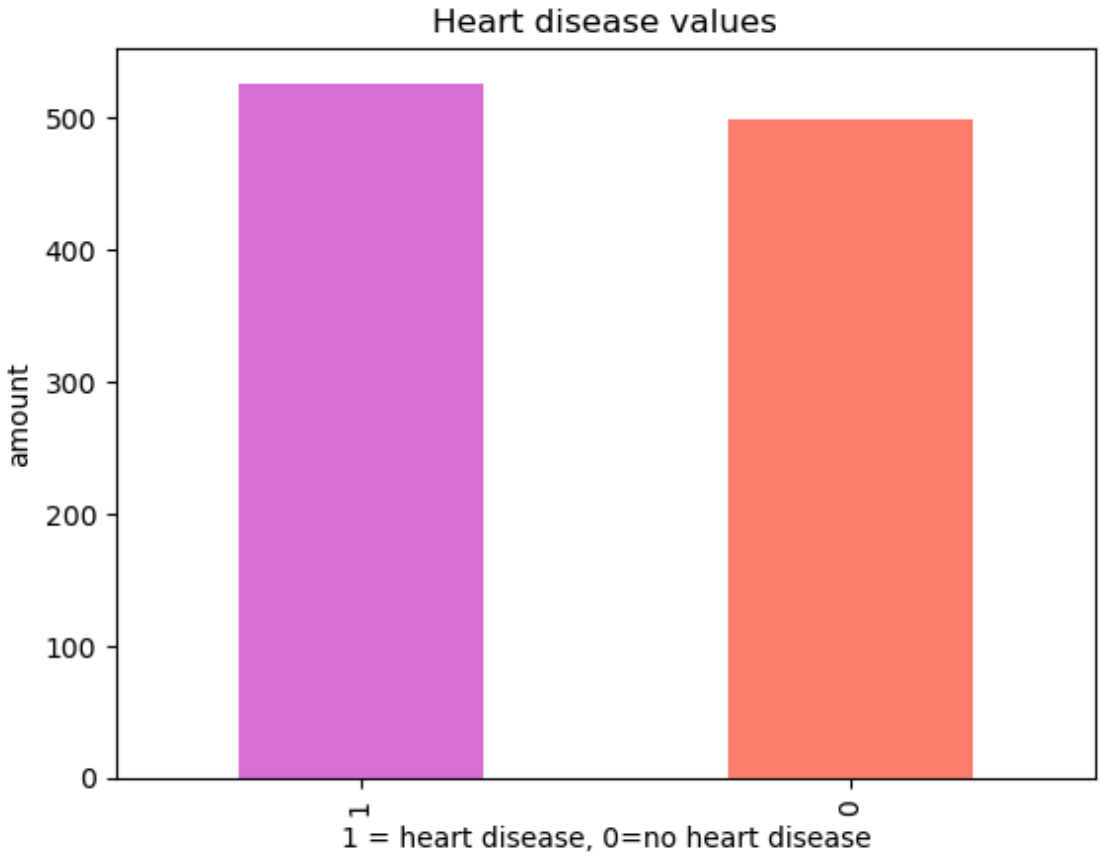
Out[19]:
```
["1. How many people have heart disease and how many people doesn't have
heart disease?",
 '2. People of which sex has most heart disease?',
 '3. People of which sex has which type of chest pain most?',
 '4. People with which chest pain are most pron to have heart disease?']
```

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

Out[20]:
```
target
1    526
0    499
Name: count, dtype: int64
```

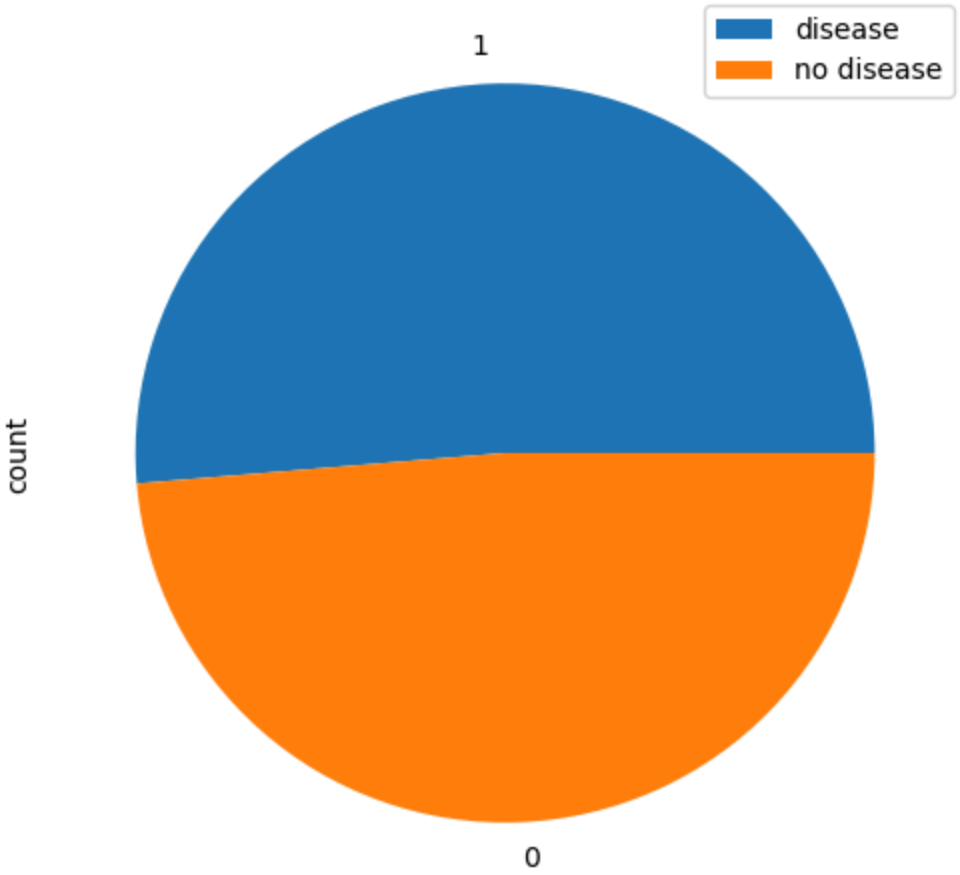
In [21]:
```
df.target.value_counts().plot(kind ='bar', color=["orchid","salmon"])
plt.title("Heart disease values")
plt.xlabel("1 = heart disease, 0=no heart disease")
plt.ylabel("amount")
```

Out[21]:
```
Text(0, 0.5, 'amount')
```



In [22]:
```
df.target.value_counts().plot(kind='pie', figsize=(8,6))
plt.legend(["disease","no disease"])
```

Out[22]: <matplotlib.legend.Legend at 0x26f5a288d90>

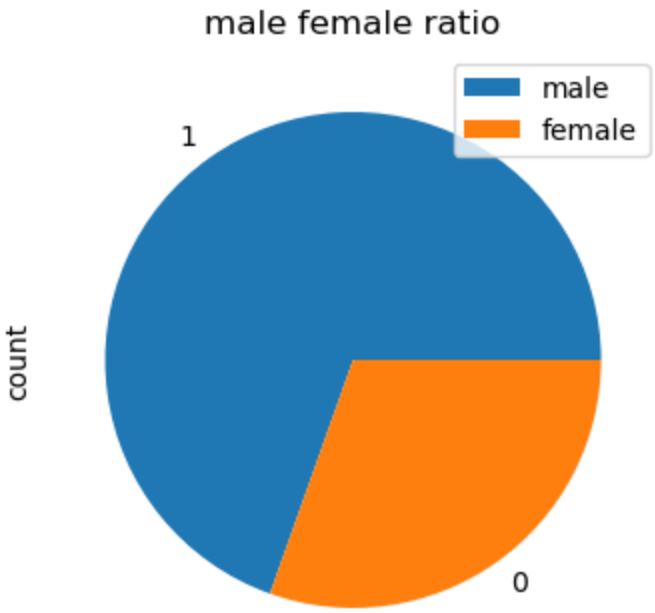


In [23]:
```
df.sex.value_counts()
```

Out[23]:
```
sex
1    713
0    312
Name: count, dtype: int64
```

In [24]:
```
df.sex.value_counts().plot(kind='pie',figsize=(4,6))
plt.title('male female ratio')
plt.legend(['male','female'])
```

Out[24]: <matplotlib.legend.Legend at 0x26f5a55e410>



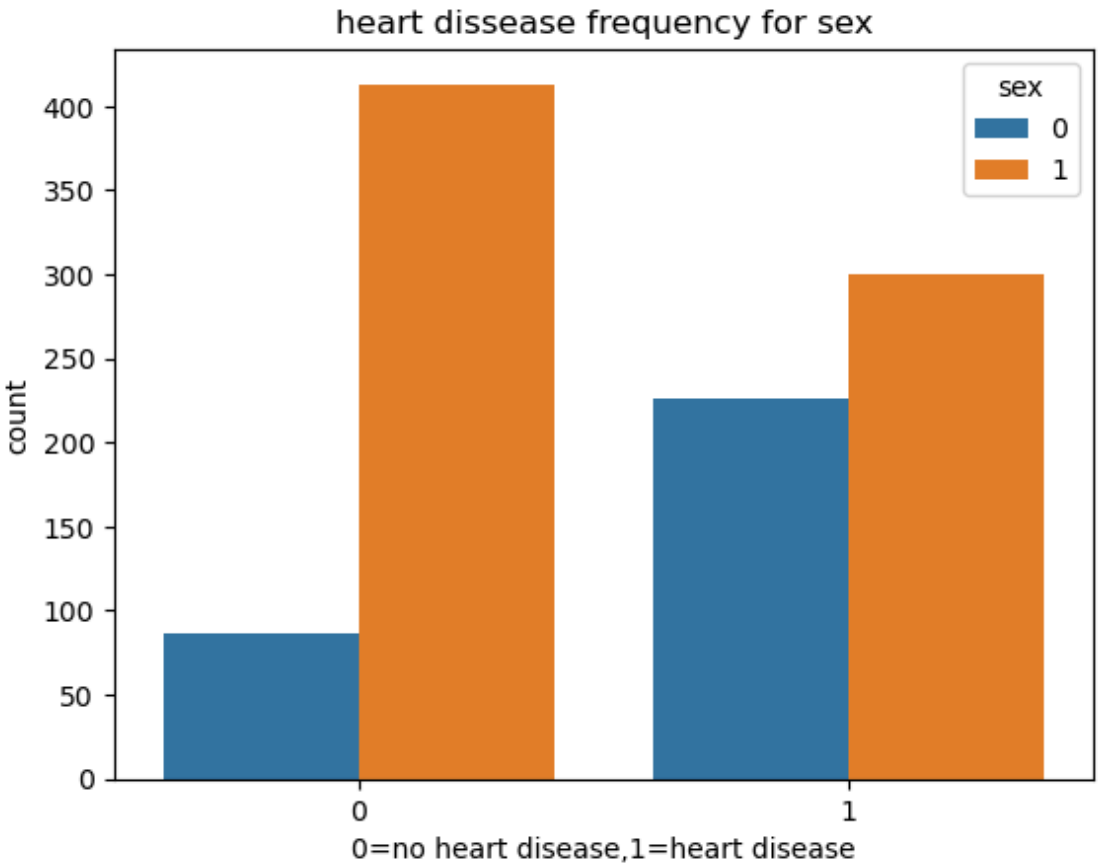
In [25]:
```
pd.crosstab(df.target,df.sex)
```

Out[25]:

sex	0	1
target		
0	86	413
1	226	300

In [26]:
```
sns.countplot(x='target', data = df, hue='sex')
plt.title("heart dissease frequency for sex")
plt.xlabel("0=no heart disease,1=heart disease")
```

Out[26]: Text(0.5, 0, '0=no heart disease,1=heart disease')



In [27]:
```
df.cp.value_counts()
```

Out[27]:
```
cp
0    497
2    284
1    167
3     77
Name: count, dtype: int64
```

In [35]:
```
pd.crosstab.value_counts.plot(kind='bar', color=['salmon','lightskyblue',
plt.title('chest pain type vs count')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[35], line 1
----> 1 pd.crosstab.value_counts.plot(kind='bar', color=['salmon','lightskyblue', 'springgreen', 'khaki'])
      2 plt.title('chest pain type vs count')

AttributeError: 'function' object has no attribute 'value_counts'
```

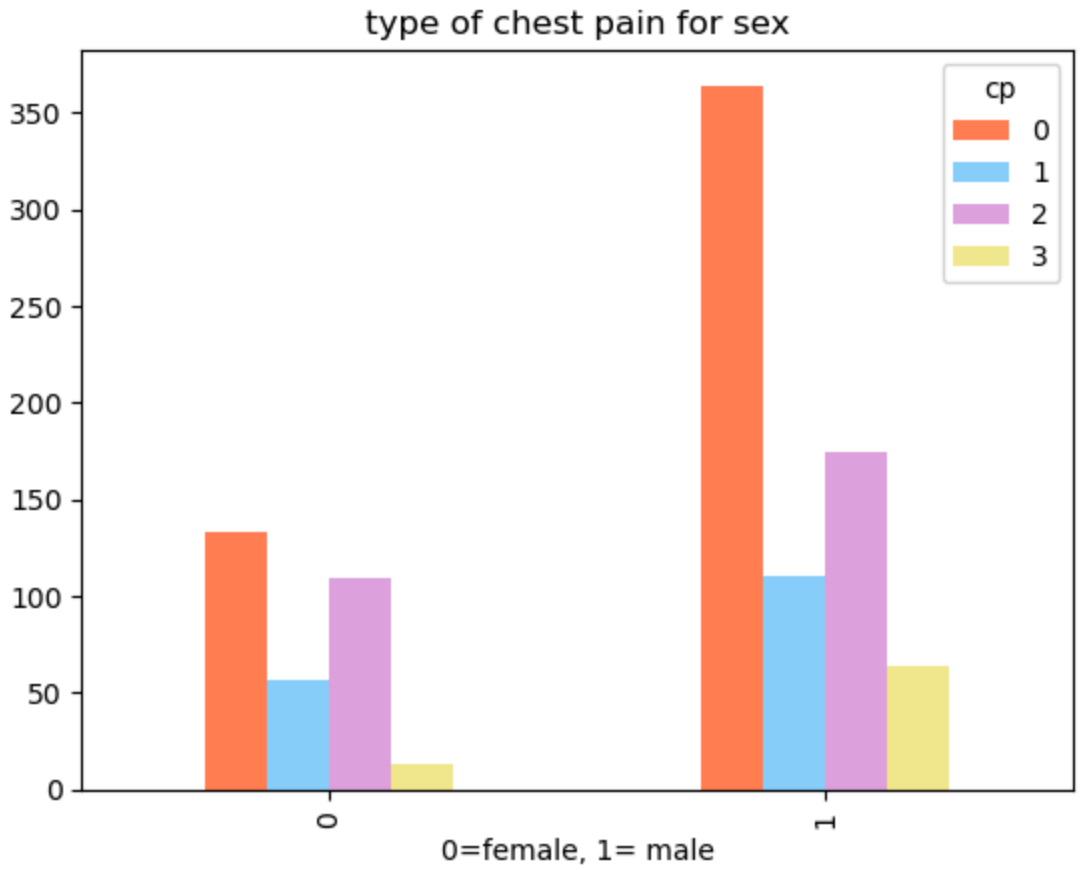
In [32]:
```
pd.crosstab(df.sex,df.cp)
```

Out[32]:

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

In [33]:
```
pd.crosstab(df.sex,df.cp).plot(kind='bar', color=['coral', 'lightskyblue',
plt.title('type of chest pain for sex')
plt.xlabel('0=female, 1= male')
```

Out[33]: Text(0.5, 0, '0=female, 1= male')



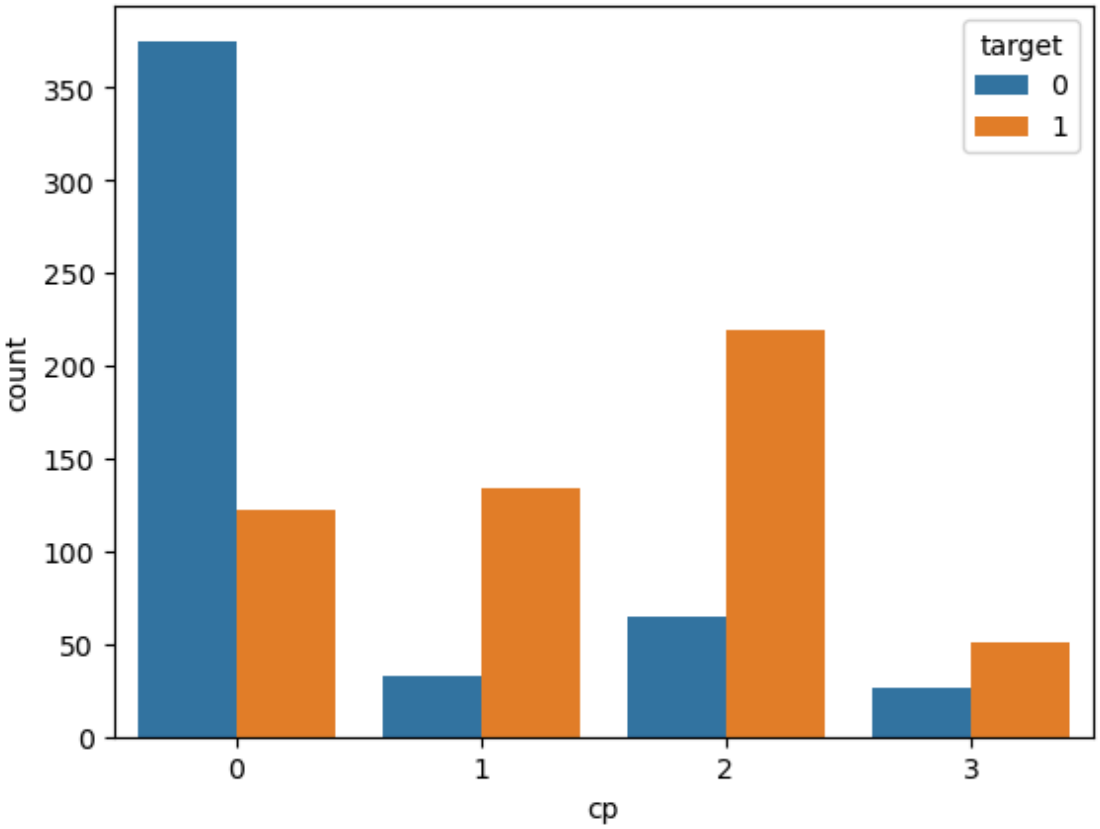
In [36]:
```
pd.crosstab(df.cp, df.target)
```

Out[36]:

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

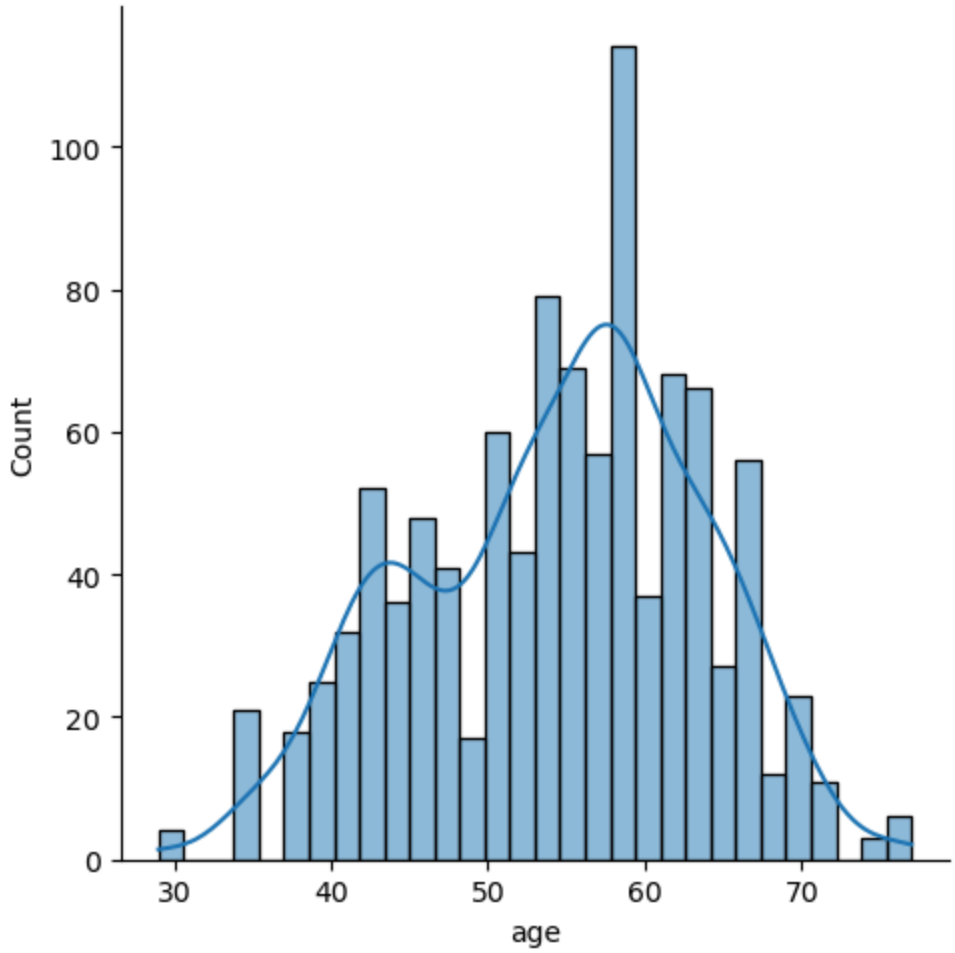
In [38]:
```
sns.countplot(x = 'cp', data =df, hue='target')
```

Out[38]: <Axes: xlabel='cp', ylabel='count'>



In [39]:
```
sns.displot(x='age', data = df, bins=30, kde =True);
```

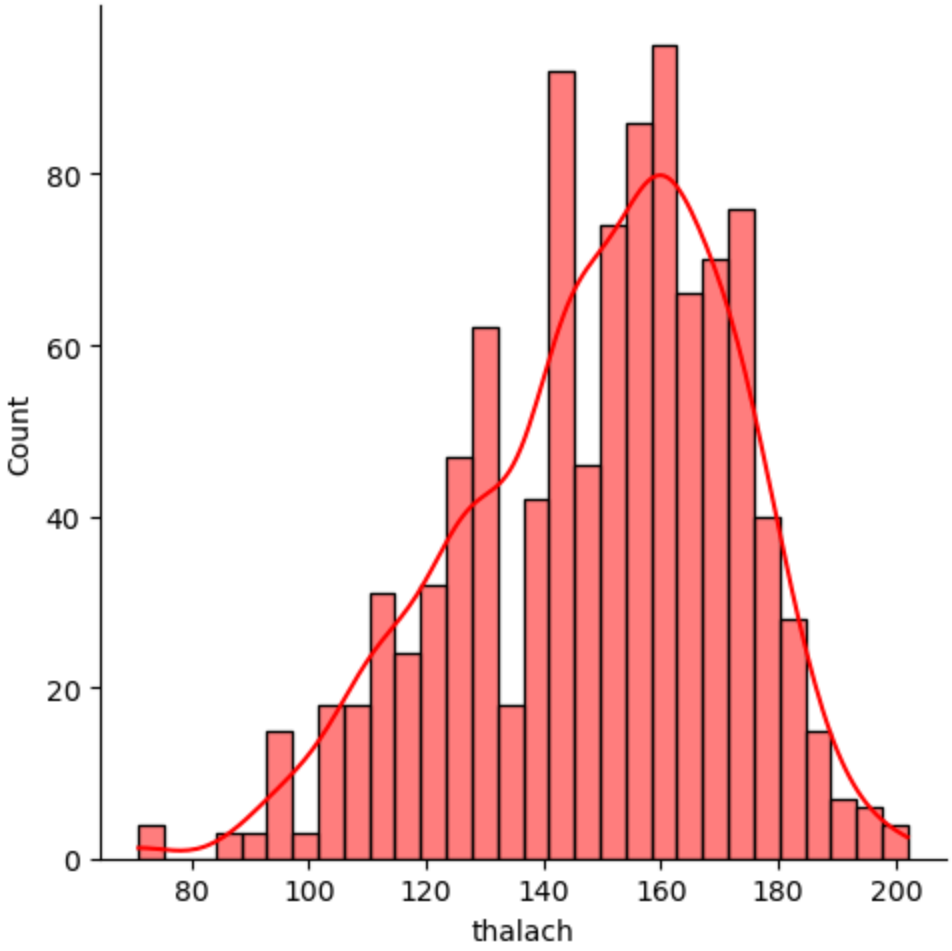
```
C:\Users\lohit\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserW
arning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [40]:
```
sns.displot(x='thalach', data =df, bins =30, kde=True, color='red')
```

```
C:\Users\lohit\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserW
arning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[40]:
```
<seaborn.axisgrid.FacetGrid at 0x26f601df950>
```



In [ ]:
```

```