Data Analysis using Python-Task5

April 14, 2024

```
[2]: # 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
```

```
[3]: # Importing our dataset
     df = pd.read_csv("C:\\Program Files\\PostgreSQL\\16\\data\\data_copy\\heart.
      ↪csv")
```

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

```
[3]:    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

        ca  thal  target
     0   2     3       0
     1   0     3       0
     2   0     3       0
     3   1     3       0
     4   3     2       0
```

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

```
[4]:       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
```

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

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

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

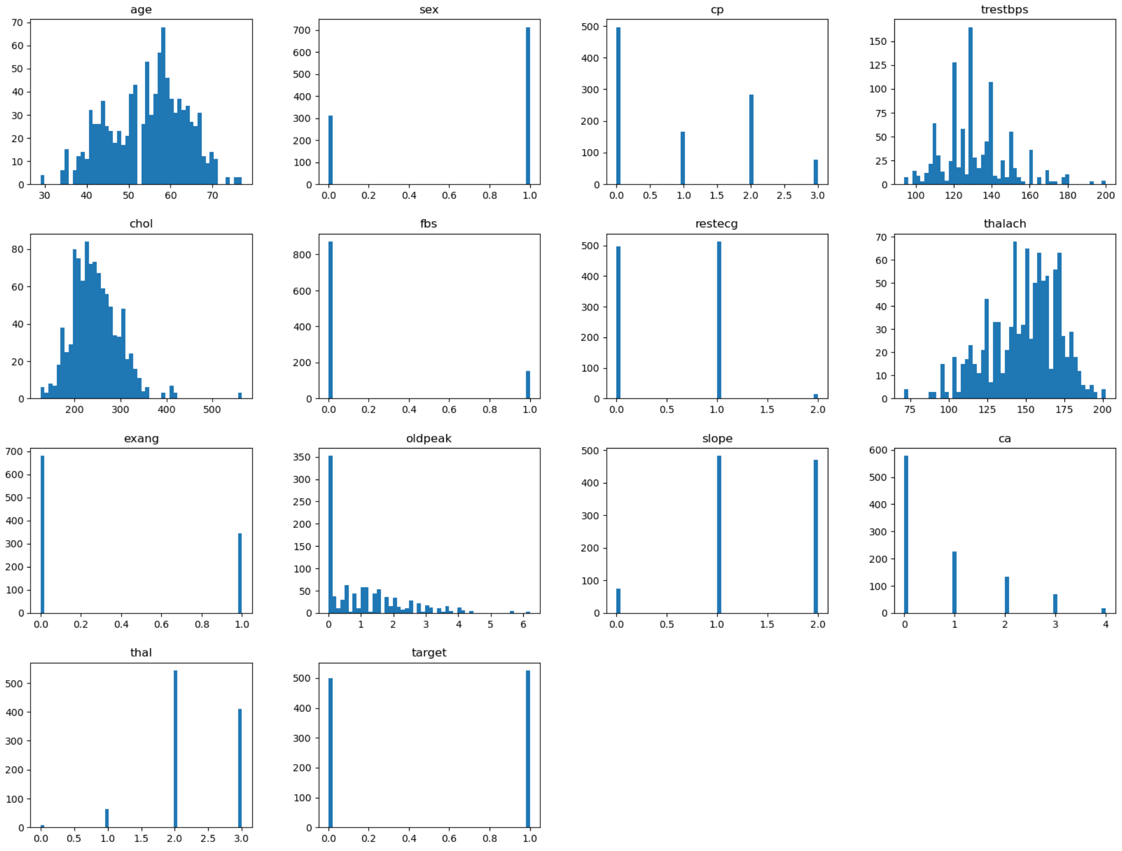
```
[12]: age         0
      sex         0
      cp          0
      trestbps    0
      chol        0
      fbs         0
      restecg     0
      thalach     0
      exang       0
      oldpeak     0
      slope       0
      ca          0
      thal        0
      target      0
      dtype: int64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

```
[13]: # 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
```

[14]:

```
# Plotting histogram of all numeric value
df.hist(bins = 50, grid = False, figsize = (20,15));
```



[15]:

```
# Generating descriptive statistics
df.describe()
```

[15]:

```
               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
```

[10]:

```
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?",
             "5. People which having high Cholestrol for Heart Disease?",
             "6. People which coronary artery(increase the risk of heart␣
 ↪attacks) for heart disease?",
             "7. People which having trestbps(high and normal blood pressure)␣
 ↪for heart disease?"]

questions
```

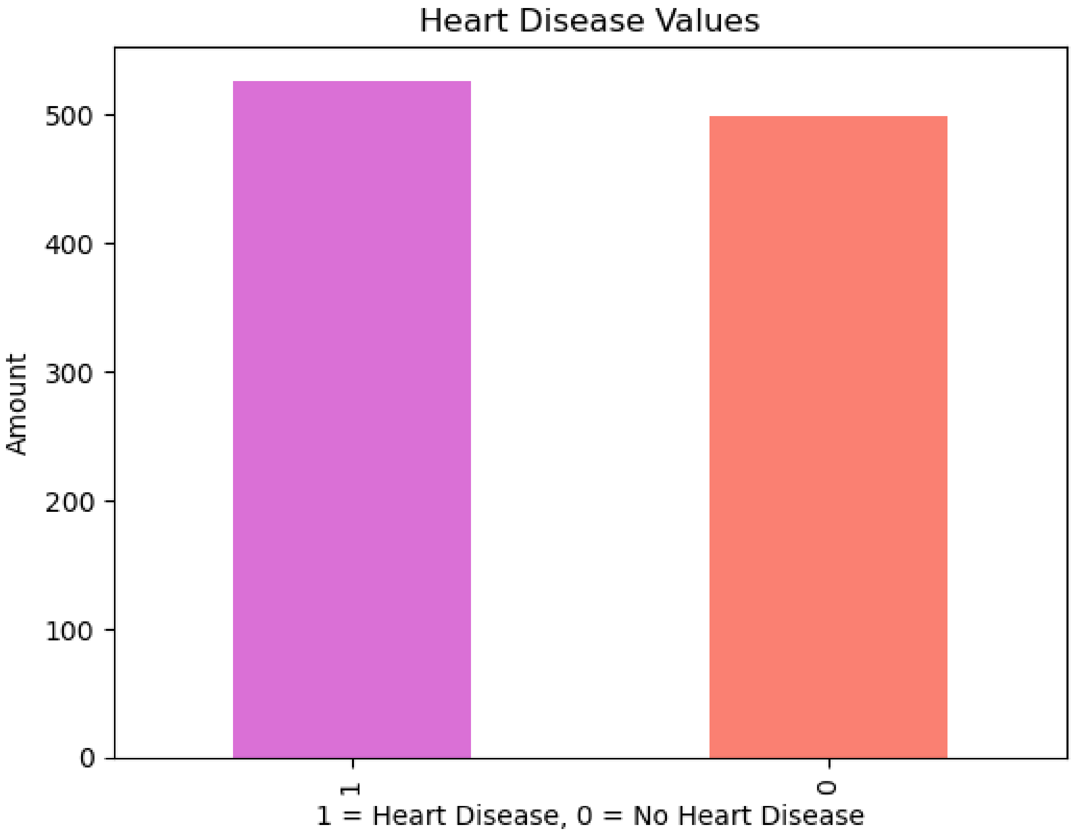
[10]:

```
["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?',
 '5. People which having high Cholestrol for Heart Disease?',
 '6. People which coronary artery(increase the risk of heart attacks) for heart
disease?',
 '7. People which having trestbps(high and normal blood pressure) for heart
disease?']
```

```
[23]: # Let's 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()
```

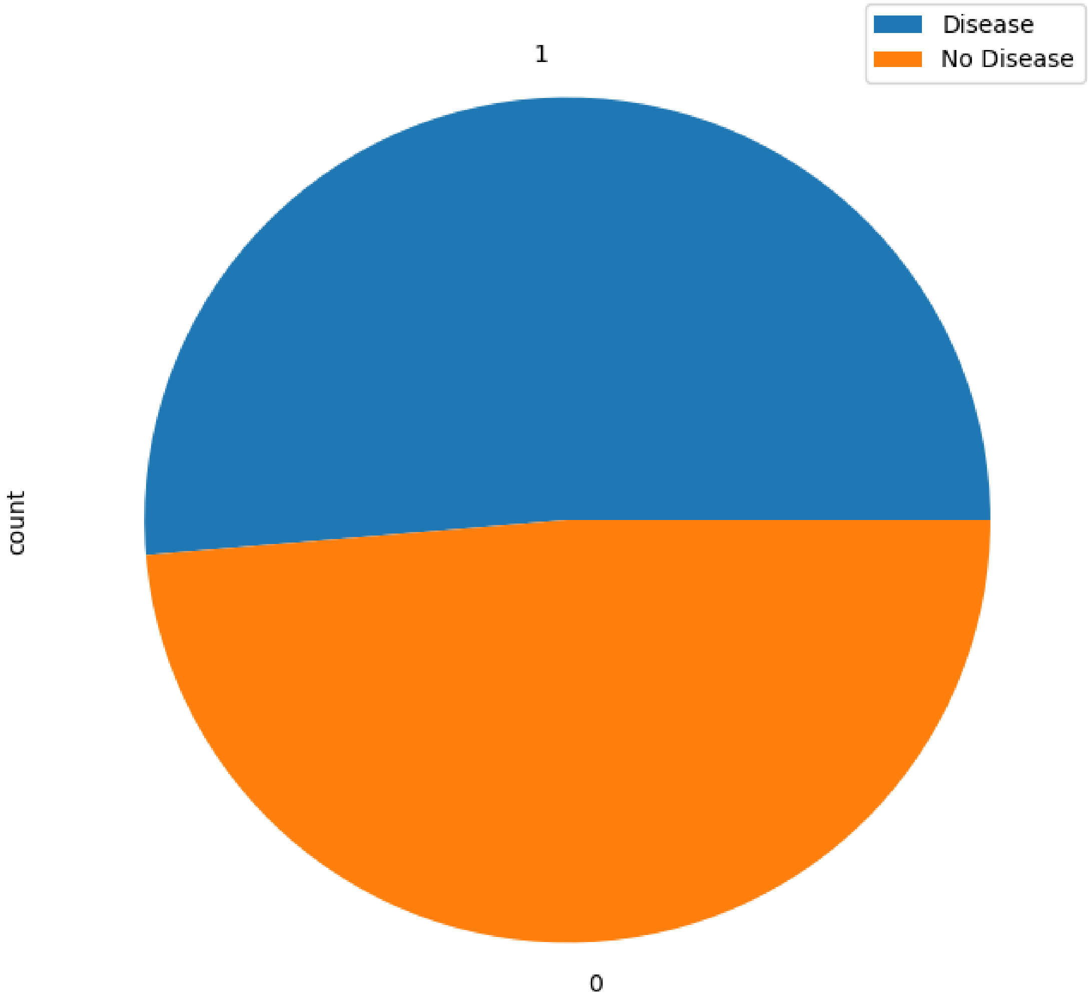
```
[23]: target
      1    526
      0    499
      Name: count, dtype: int64
```

```
[25]: # 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 Heart Disease")
      plt.ylabel("Amount");
```



[27]:

```python
# Plotting a pie chart
df.target.value_counts().plot(kind = 'pie', figsize = (15,8))
plt.legend(["Disease", "No Disease"]);
```



[28]:

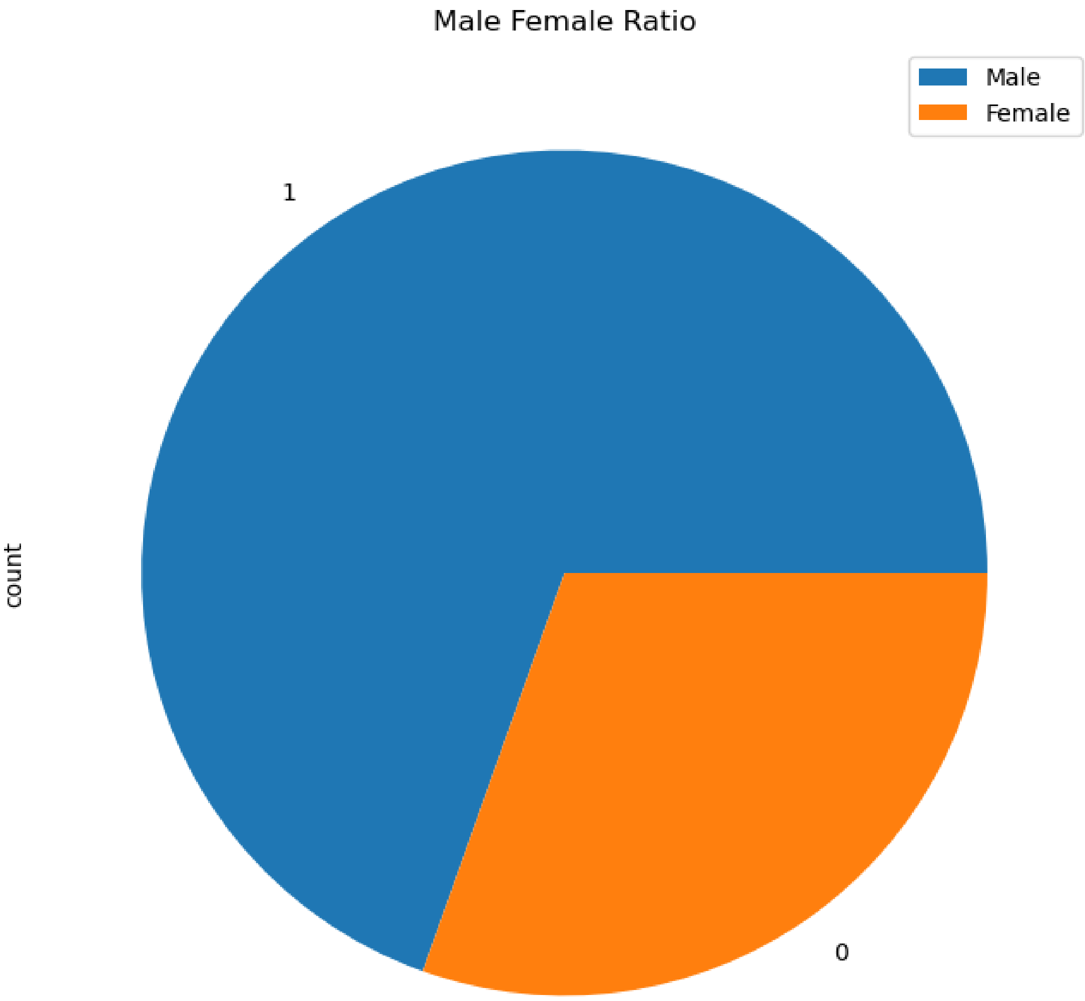
```python
# '0' represent 'Female'
# '1' represent 'Male'
# '0' represent 'No Disease'
# '1' represent 'Disease'

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

[28]:

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

[30]:

```
# Plotting a pie chart
df.sex.value_counts().plot(kind = 'pie', figsize = (12,8))
plt.title('Male Female Ratio')
plt.legend(['Male', 'Female']);
```



[5]:

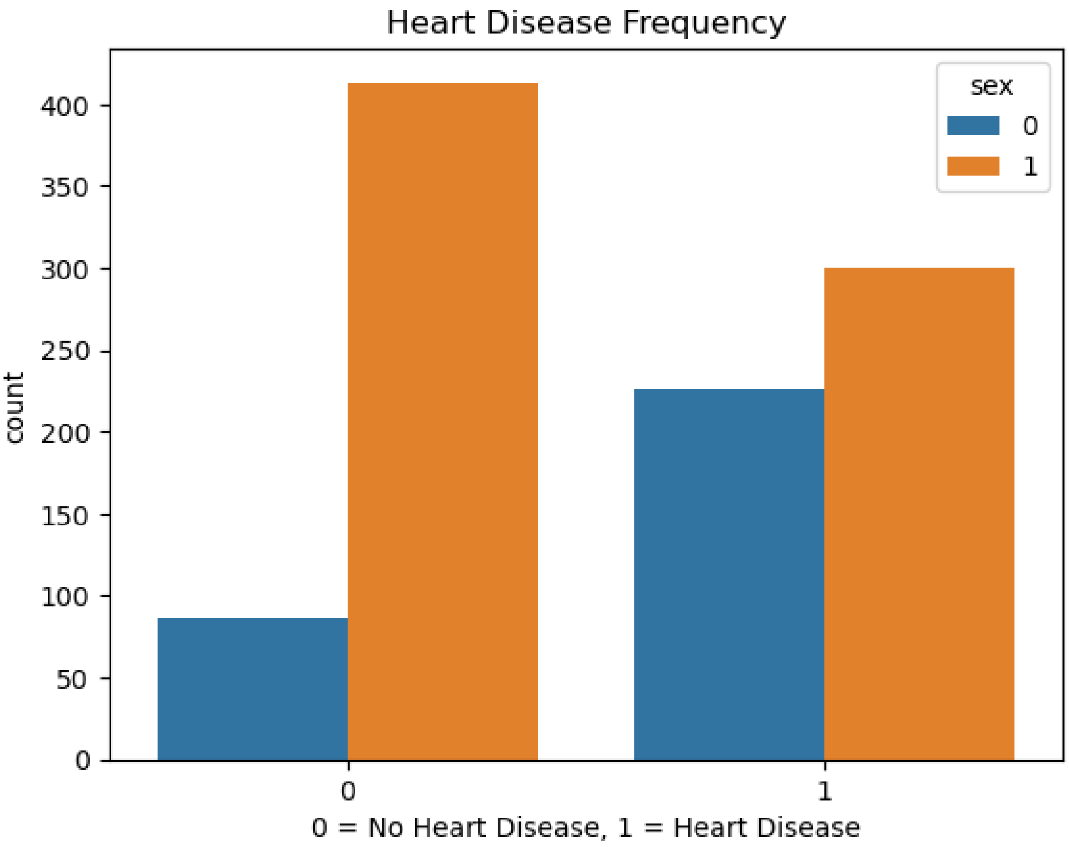
```
# Let's find the answer of our 2nd question.
# 2.People of which sex has most heart disease?
pd.crosstab(df.target, df.sex)
```

[5]:

```
sex       0    1
target
0        86  413
1       226  300
```

[6]:

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



[7]:

```
# Number of male is more than double in our dataset than Female
# More than '45% male' has heart disease and '75% Female' has heart dsease.
```

[8]:

```
# 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()
```

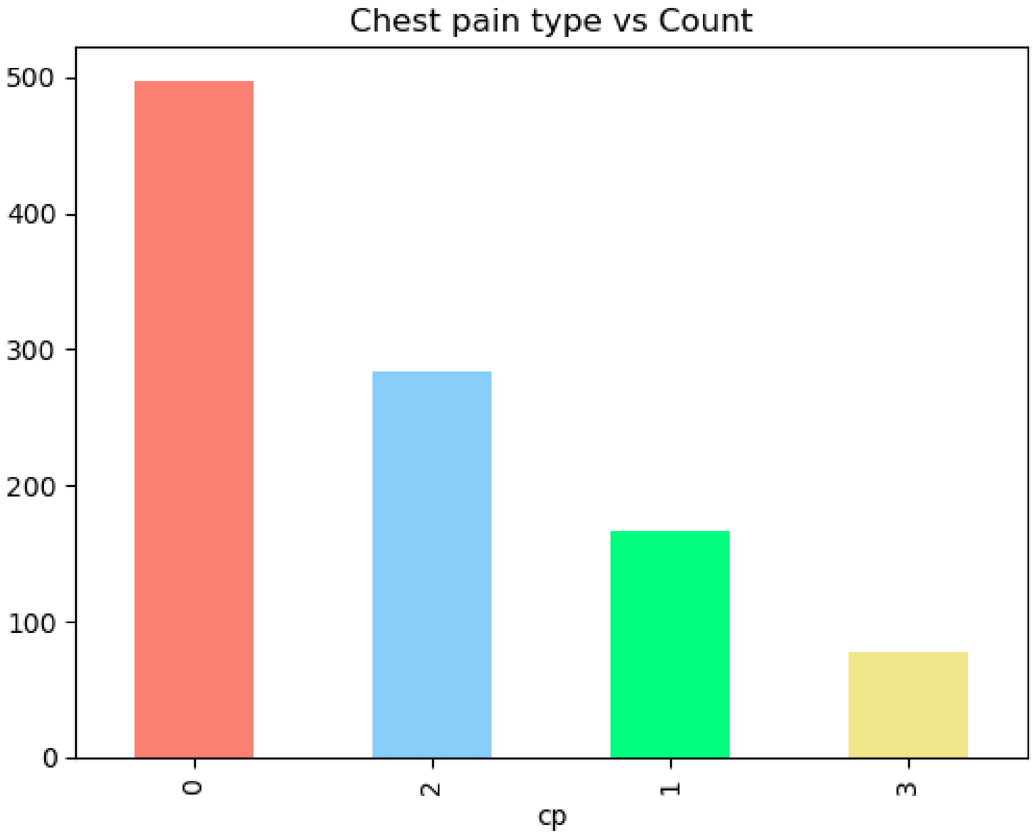
[8]:

```
cp
0    497
2    284
```

```
1    167
3     77
Name: count, dtype: int64
```

[10]:

```
# Plotting a bar chart
df.cp.value_counts().plot(kind = 'bar', color =␣
 ↪['salmon','Lightskyblue','springgreen','khaki'])
plt.title('Chest pain type vs Count');
```



[11]:

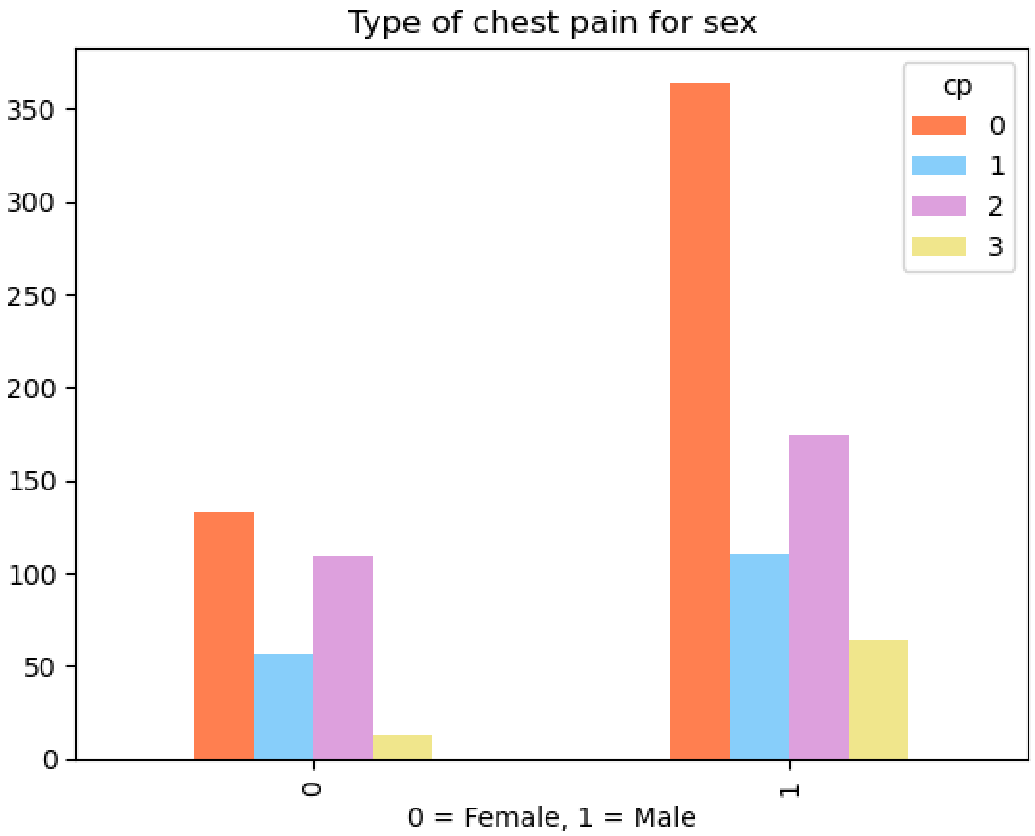
```
pd.crosstab(df.sex, df.cp)
```

[11]:

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

[16]:

```
pd.crosstab(df.sex, df.cp).plot(kind = 'bar', color = ['coral', 'lightskyblue',␣
 ↪'plum', 'khaki'])
plt.title('Type of chest pain for sex')
plt.xlabel('0 = Female, 1 = Male');
```

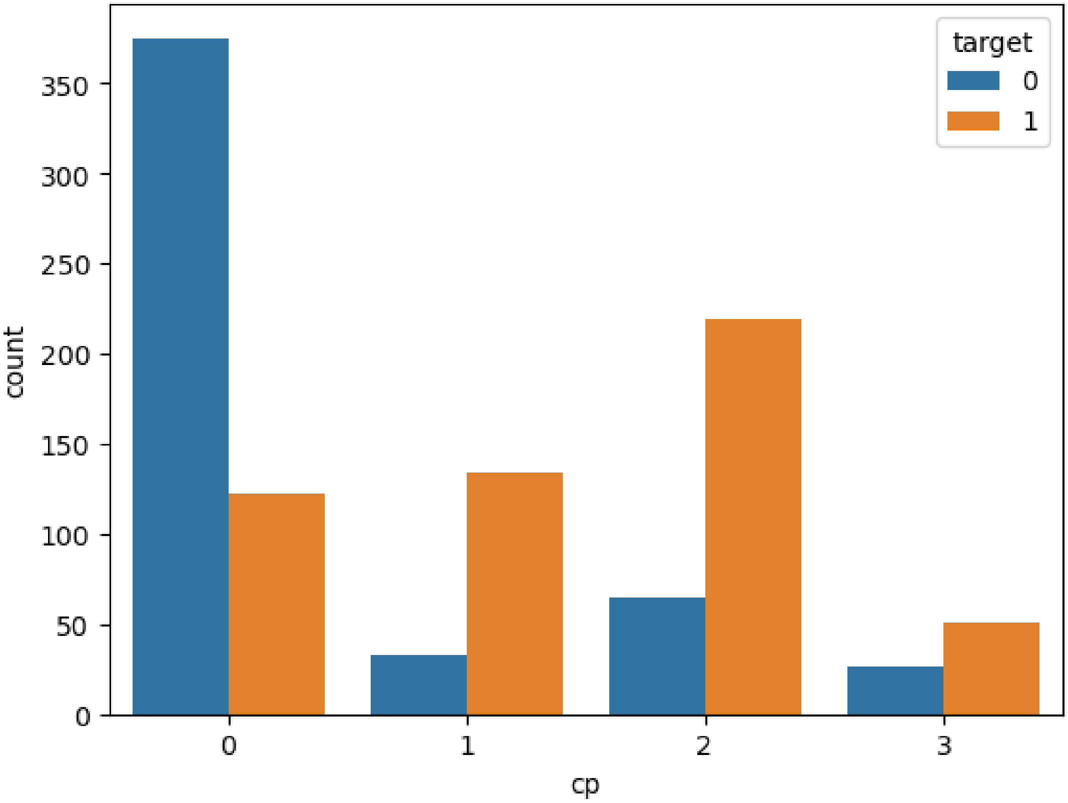


```
[17]: # Most of 'Male' has 'type 0' chest pain and least of 'Male' has 'type 4' pain
      # In case of 'Female' 'type 0' and 'type 1' percentage is almost same.
```

```
[19]: # Now question 4?
      # 4. 'People with which chest pain are most pron to have heart disease?'
      pd.crosstab(df.cp, df.target)
```

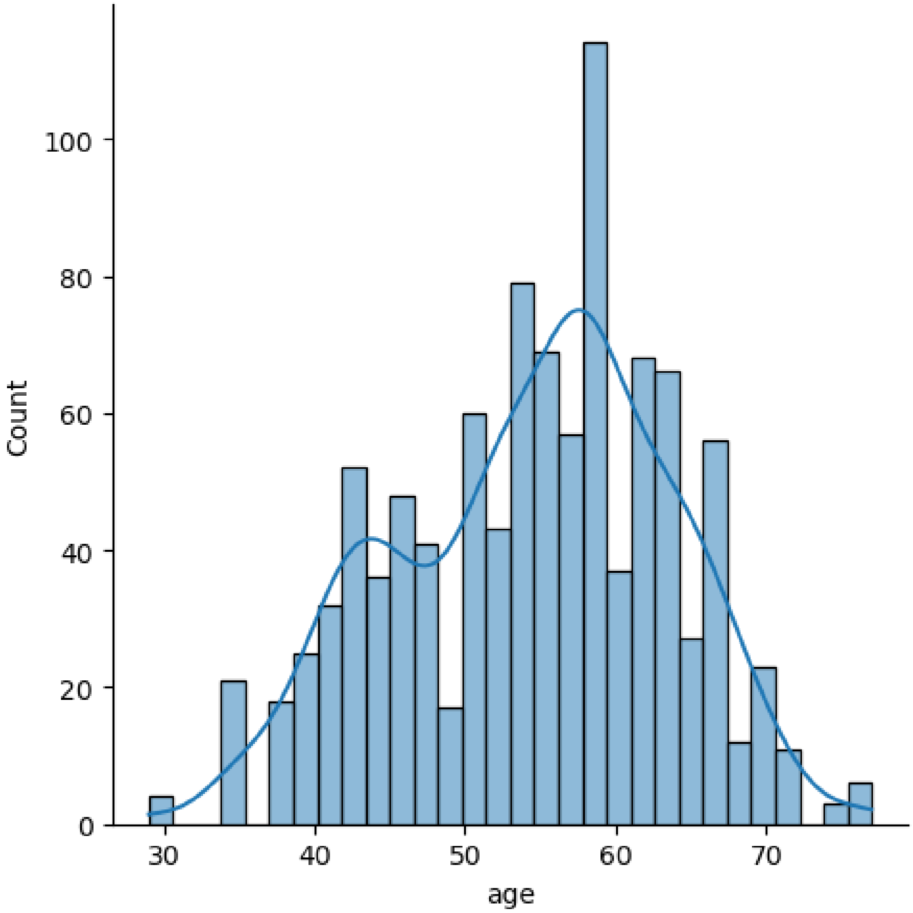
```
[19]: target    0    1
      cp
      0       375  122
      1        33  134
      2        65  219
      3        26   51
```

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



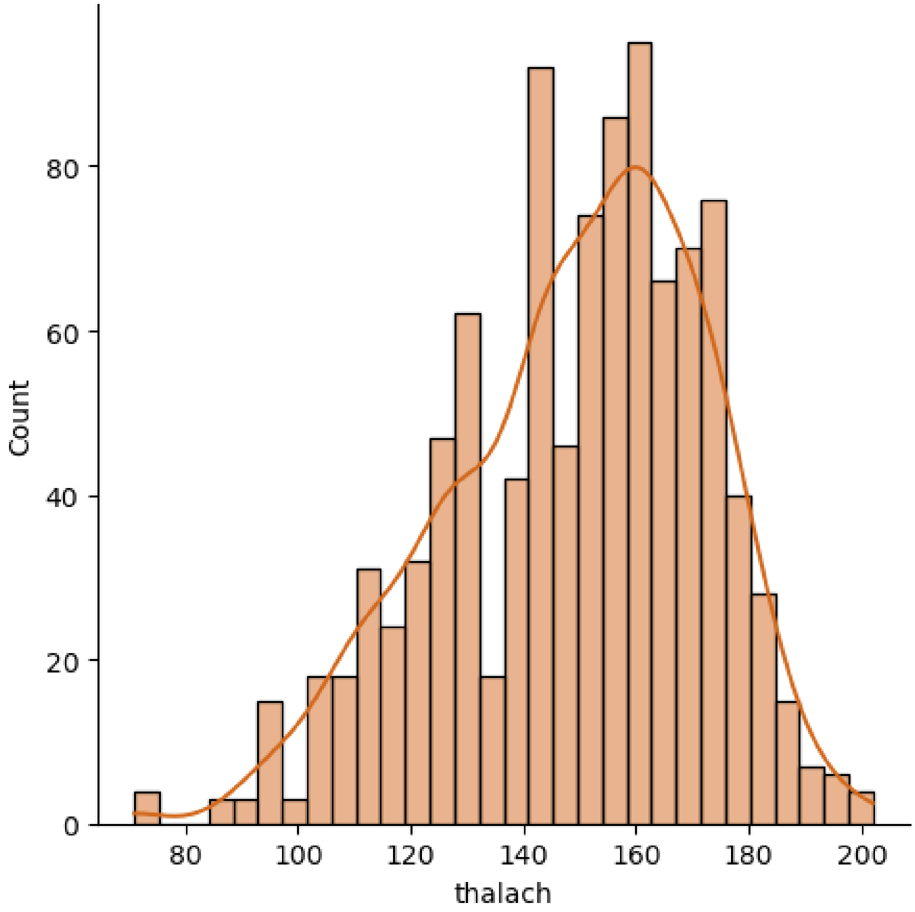
[24]:
```
# 'Most of people who has 'type a' chest pain has less chance of heart disease'.
# Add we see the opposite for other types.

# 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);
```

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



[25]:
```
# '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 =␣
 ↪'chocolate');
```

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



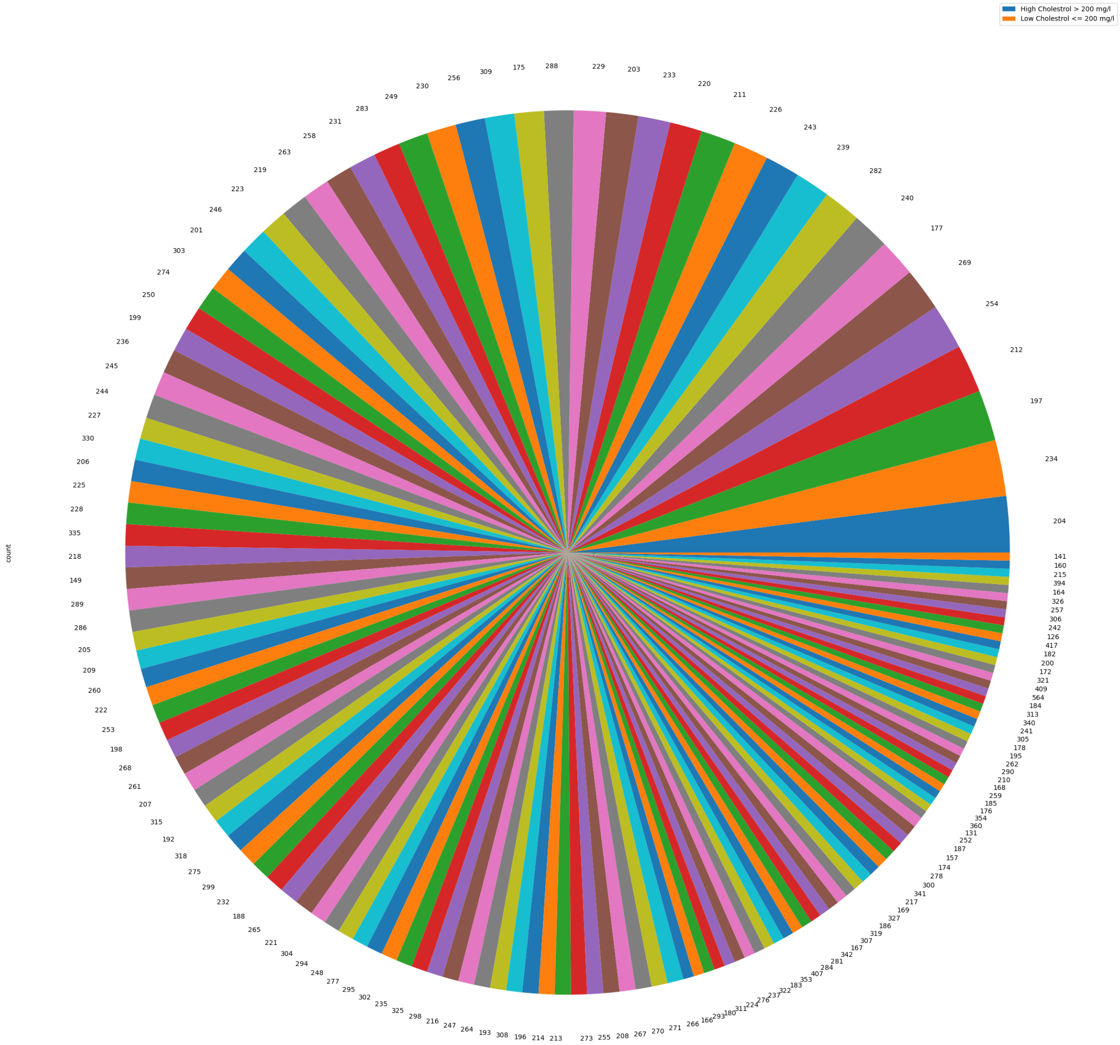
[13]:
```
# From this plot we get a clear overview about Maximum heart rate represent by␣
↪'thalach'.
```

[14]:
```
# 5.'People which having high Cholestrol for Heart Disease?'
pd.crosstab(df.chol, df.sex)
```

[14]:
```
sex   0  1
chol
126   0  3
131   0  3
141   3  0
149   4  4
157   0  4
…    .. ..
394   3  0
407   4  0
409   3  0
417   3  0
```

```
564   3  0

[152 rows x 2 columns]
```

[9]:
```
# Plotting a pie chart
df.chol.value_counts().plot(kind = 'pie', figsize = (50,30))
plt.legend(["High Cholestrol > 200 mg/l","Low Cholestrol <= 200 mg/l"]);
```

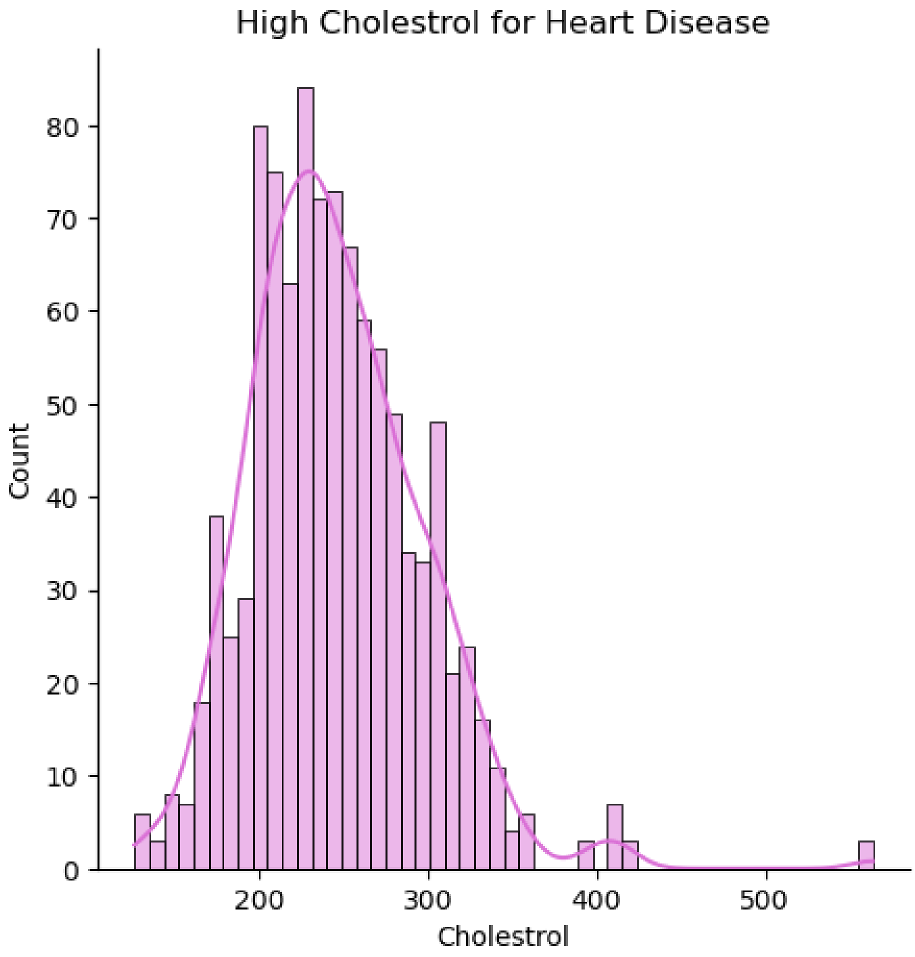


[12]:
```
sns.displot(x = 'chol', data = df, bins = 50, kde = True, color = "orchid");
plt.title('High Cholestrol for Heart Disease')
plt.xlabel('Cholestrol')
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
```

```
  self._figure.tight_layout(*args, **kwargs)
```

```
[12]: Text(0.5, 9.444444444444438, 'Cholestrol')
```

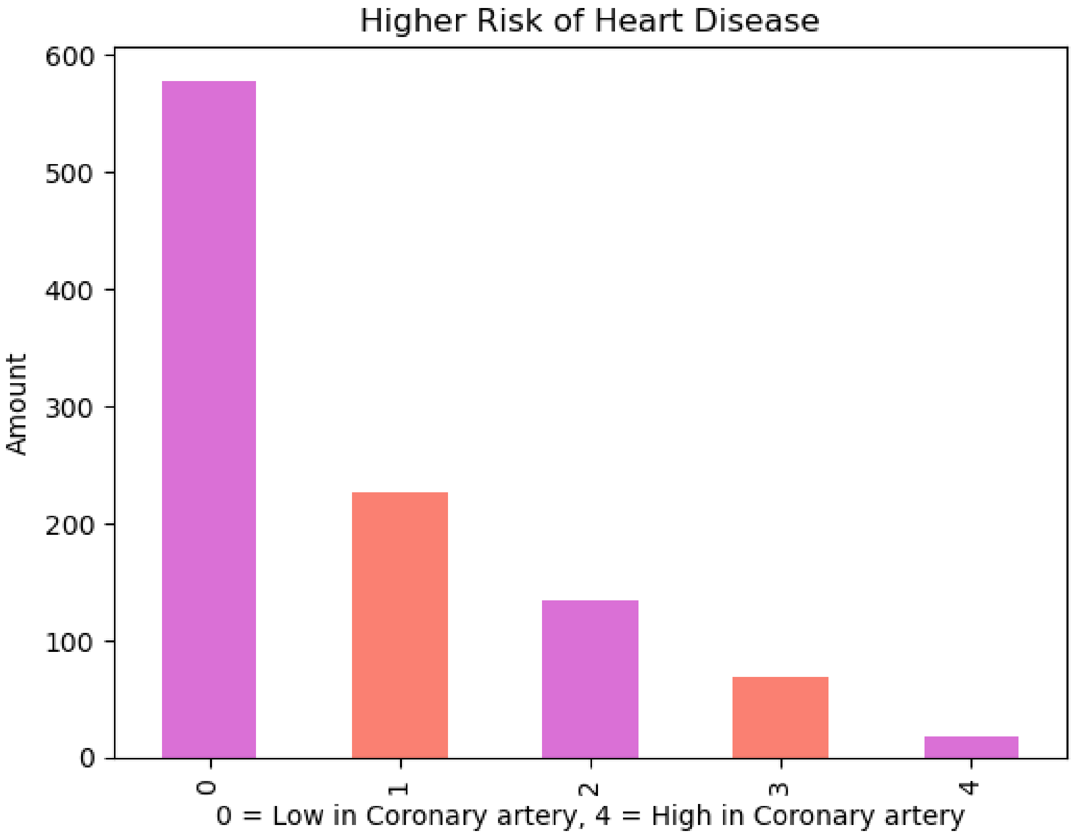


```
[15]: # 6.'People which coronary artery(increase the risk of heart attacks) for heart␣
       ↪disease?'
      # Getting the values
      df.ca.value_counts()
```

```
[15]: ca
      0    578
      1    226
      2    134
      3     69
      4     18
      Name: count, dtype: int64
```

[16]:

```
# Plotting bar chart
df.ca.value_counts().plot(kind = 'bar', color = ["orchid", "salmon"])
plt.title("Higher Risk of Heart Disease")
plt.xlabel("0 = Low in Coronary artery, 4 = High in Coronary artery")
plt.ylabel("Amount");
```



[17]:

```
# Most of 'Male' has higher risk of heart attacks compared to 'Female'.
pd.crosstab(df.ca, df.sex)
```

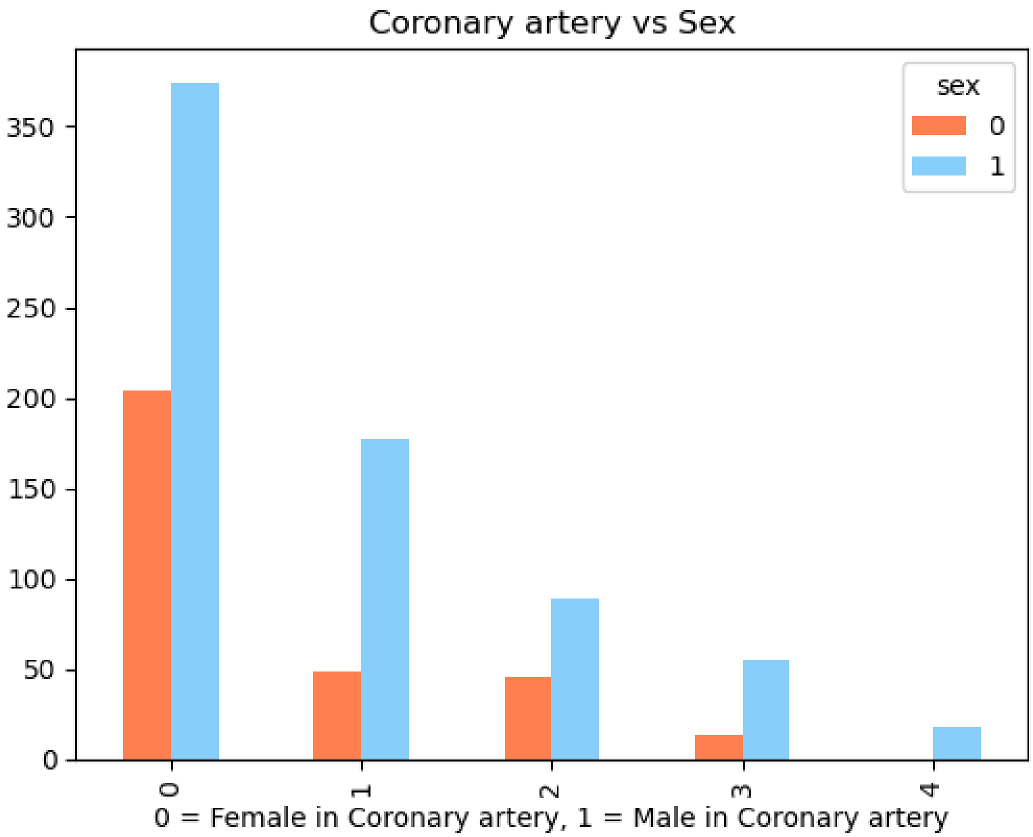
[17]:

```
sex    0    1
ca
0    204  374
1     49  177
2     45   89
3     14   55
4      0   18
```

[18]:

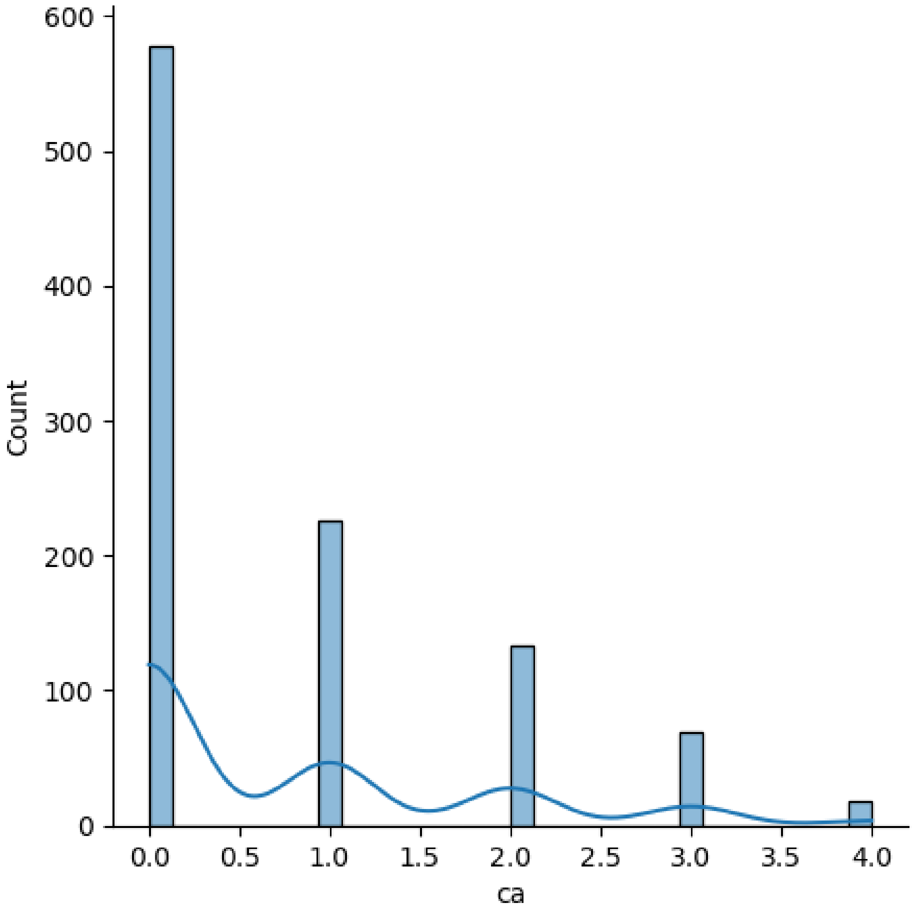
```
pd.crosstab(df.ca, df.sex).plot(kind = 'bar', color = ['coral','lightskyblue'])
plt.title('Coronary artery vs Sex')
plt.xlabel('0 = Female in Coronary artery, 1 = Male in Coronary artery')
```

```
[18]: Text(0.5, 0, '0 = Female in Coronary artery, 1 = Male in Coronary artery')
```



```
[19]: # Most of people having high risk of ca in male for heart disease
      # Now Let's take look at our ca column
      # Create a distribution plot with normal distribution curve
      sns.displot(x = 'ca', data = df, bins = 30, kde = True);
```

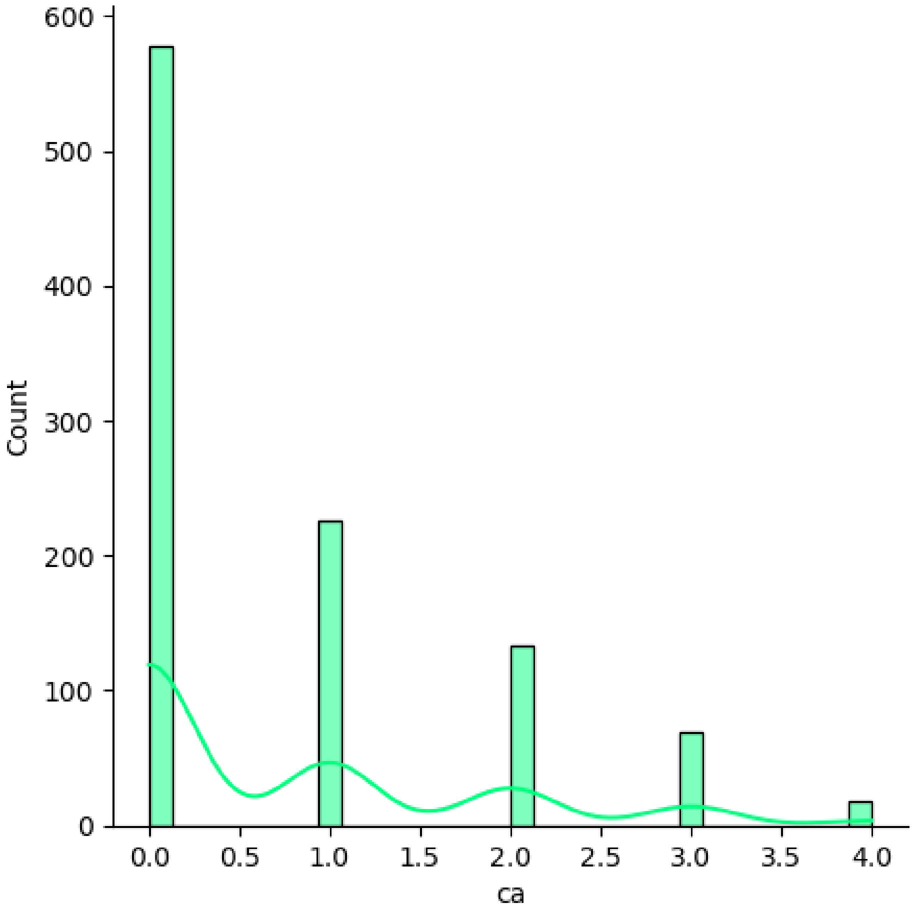
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[20]:

```
# '35-71' year old people are most in ca in the dataset.
# Let's plot another distribution plot for 'Maximum heart attacks'
sns.displot(x = 'ca', data = df, bins = 30, kde = True, color = 'springgreen');
```

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



```
[4]: # 7.'People which having trestbps(high and normal blood pressure) for heart␣
       ↪disease?'
     pd.crosstab(df.trestbps, df.sex)
```

```
[4]: sex        0   1
     trestbps
     94         3   4
     100        3  11
     101        0   3
     102        6   0
     104        0   3
     105        6   3
     106        3   0
     108       10  11
     110       10  54
     112       10  20
     114        0   4
     115        3   6
```

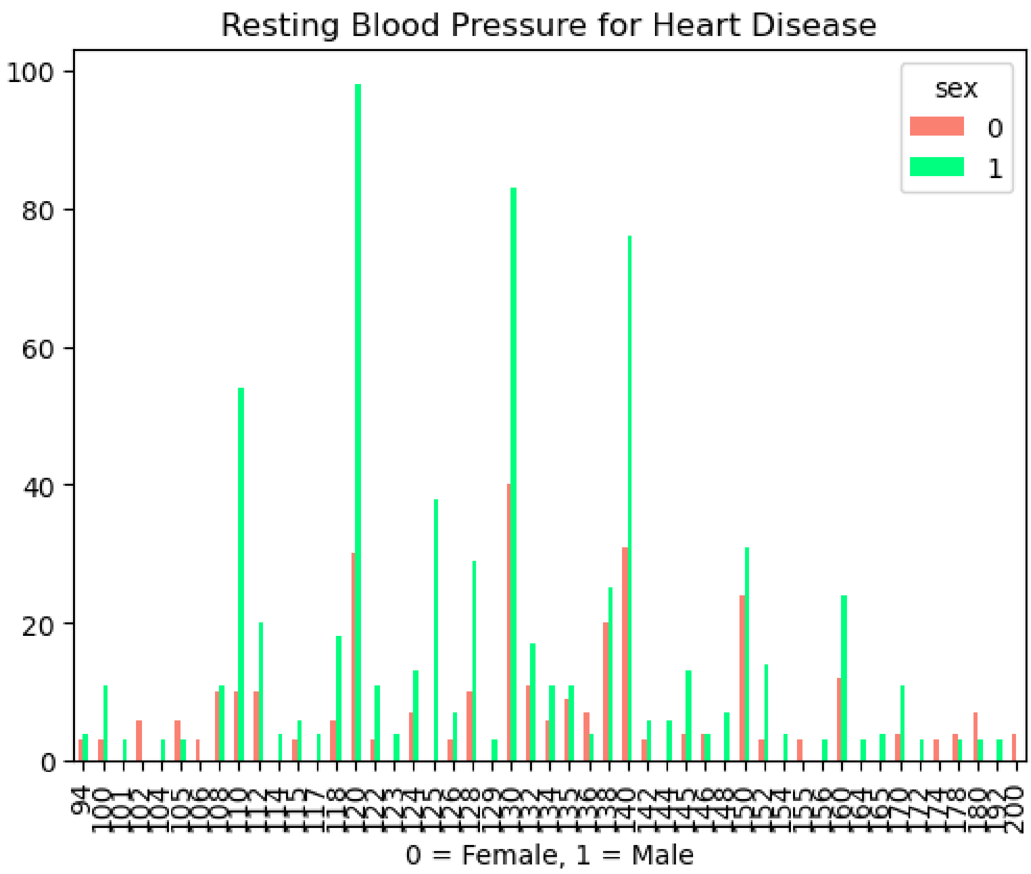
```
117          0   4
118          6  18
120         30  98
122          3  11
123          0   4
124          7  13
125          0  38
126          3   7
128         10  29
129          0   3
130         40  83
132         11  17
134          6  11
135          9  11
136          7   4
138         20  25
140         31  76
142          3   6
144          0   6
145          4  13
146          4   4
148          0   7
150         24  31
152          3  14
154          0   4
155          3   0
156          0   3
160         12  24
164          0   3
165          0   4
170          4  11
172          0   3
174          3   0
178          4   3
180          7   3
192          0   3
200          4   0
```

[5]:

```
pd.crosstab(df.trestbps, df.sex).plot(kind = 'bar', color =␣
 ↪['salmon','springgreen'])
plt.title('Resting Blood Pressure for Heart Disease')
plt.xlabel('0 = Female, 1 = Male')
```

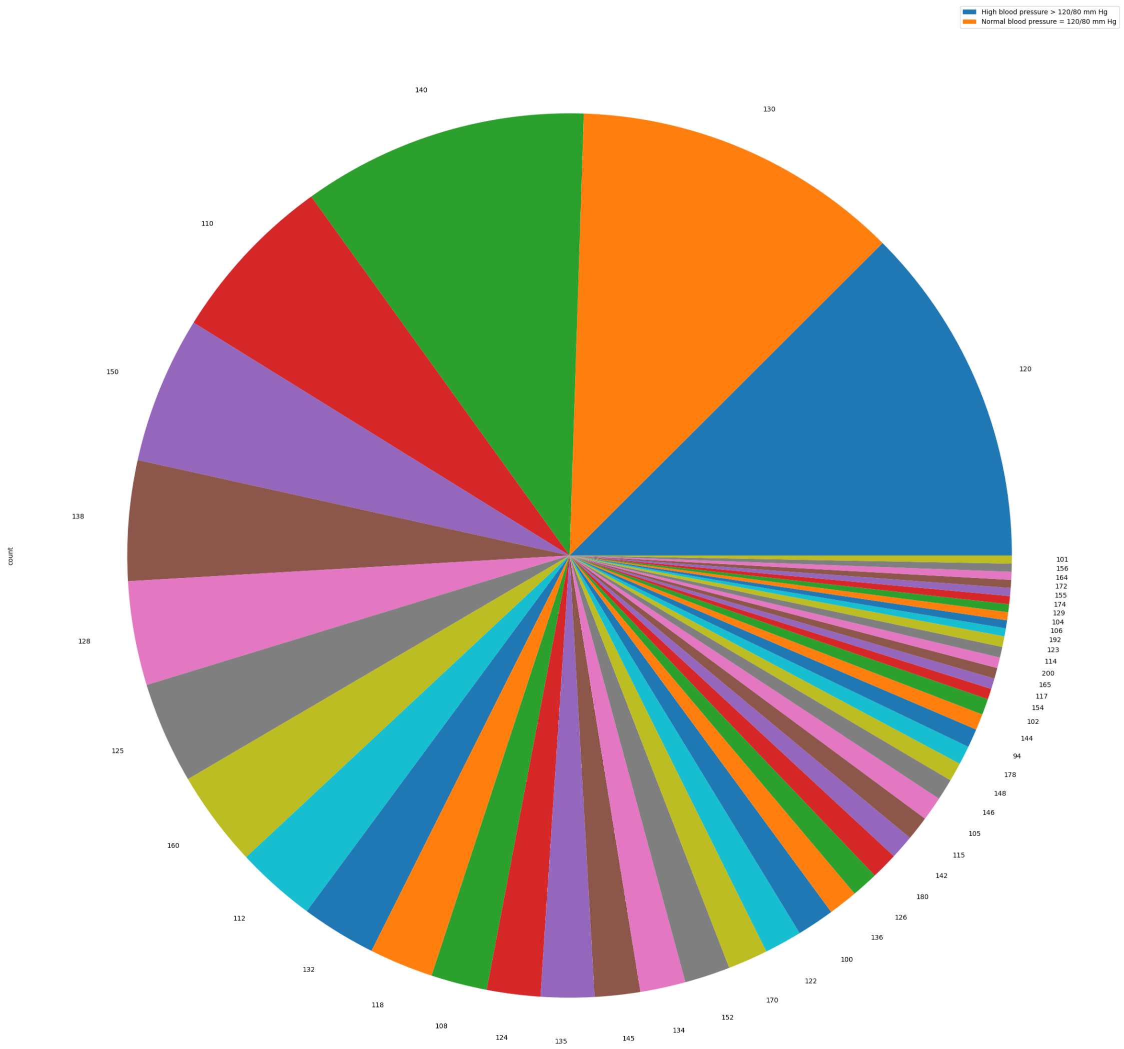
[5]:

```
Text(0.5, 0, '0 = Female, 1 = Male')
```



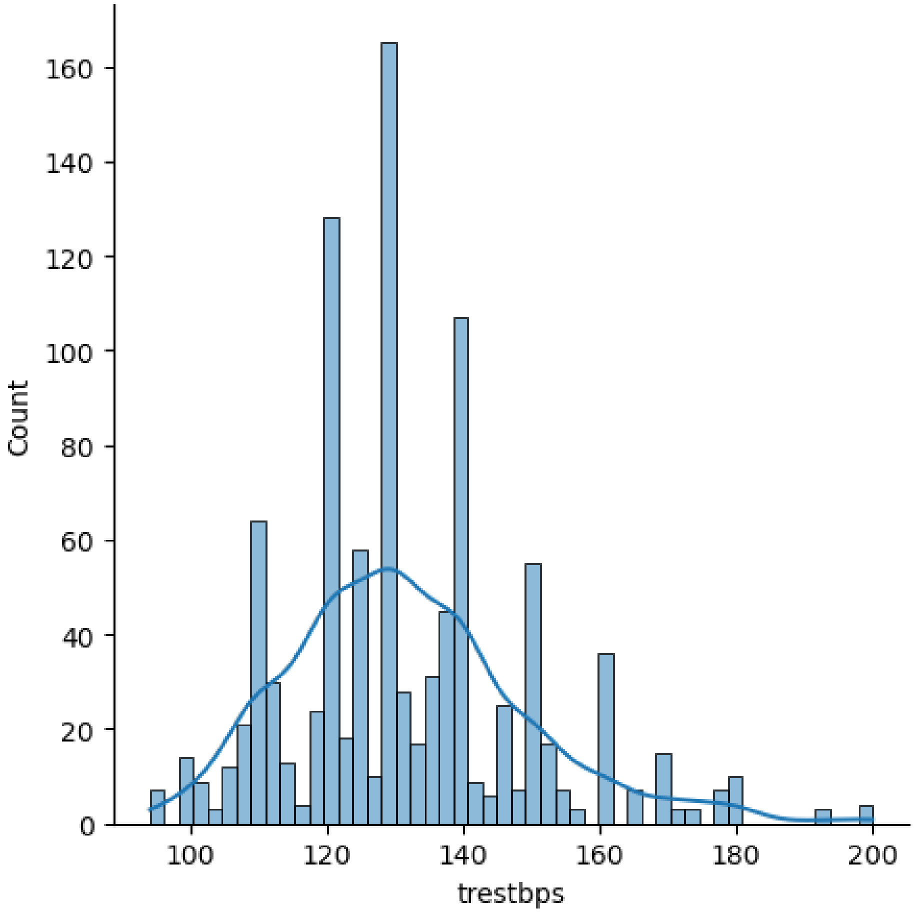
```
[6]: df.trestbps.value_counts().plot(kind ='pie', figsize = (50, 30))
     plt.legend(["High blood pressure > 120/80 mm Hg","Normal blood pressure = 120/
      ↪80 mm Hg"])
```

```
[6]: <matplotlib.legend.Legend at 0x2416fe02510>
```



[7]:
```
# Most of people who has High blood pressure is 'Male' compared to 'Female' of chance of heart disease.
# Now Let's take look at our trestbps column.
# Create a distribution plot with normal distrubution curve.
sns.displot(x = 'trestbps', data = df, bins = 50, kde = True);
```

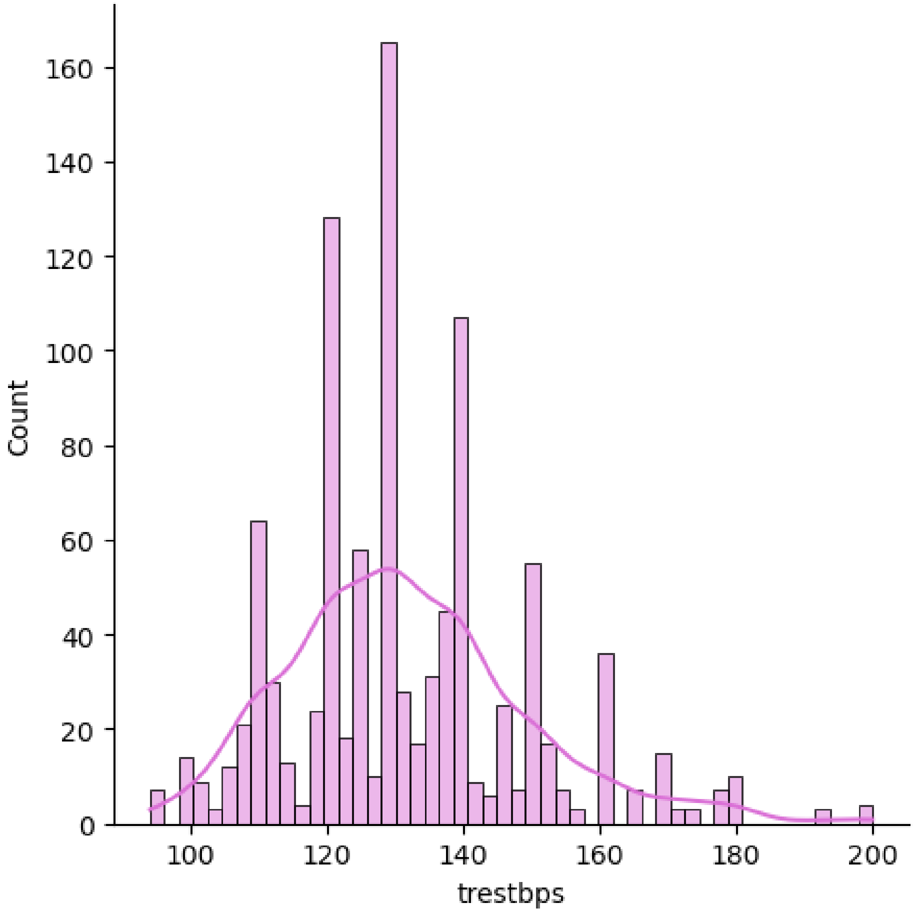
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[8]:

```
# '51-70' year old people are high blood pressure in the dataset.
# Let's plot another distribution plot for 'Maximum blood pressure'
sns.displot(x = 'trestbps', data = df, bins = 50, kde = True, color = 'orchid');
```

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


Count
0
20
40
60
80
100
120
140
160
100
120
140
160
180
200
trestbps

```
[ ]:
```