9913_DWM_Exp3

August 13, 2024

```
[ ]: #1) Load the libraries
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
```

```
[ ]: #2) Download the data set from kaggle/ other sources
     dataset = "heart.csv"
```

```
[ ]: #3) Read the file -select appropriate file read function according to data type␣
       ↪of file
     df = pd.read_csv(dataset)
```

```
[ ]: #4) Display attributes in the data set-10 samples.
     print (df.head(10))
```

```
   age  sex  cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
0   63    1   3     145.0  233.0    1        0    150.0      0      2.3
1   37    1   2     130.0  250.0    0        1    187.0      0      3.5
2   41    0   1     130.0    NaN    0        0    172.0      0      1.4
3   56    1   1     120.0    NaN    0        1    178.0      0      0.8
4   57    0   0     120.0  354.0    0        1    163.0      1      0.6
5   57    1   0       NaN  192.0    0        1    148.0      0      0.4
6   56    0   1       NaN  294.0    0        0    153.0      0      1.3
7   44    1   1     120.0  263.0    0        1      NaN      0      0.0
8   52    1   2     172.0  199.0    1        1      NaN      0      0.5
9   57    1   2     150.0  168.0    0        1    174.0      0      1.6

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

[ ]:

```python
#5) Describe the attributes name, count no of values, and find min, max, data type, range, quartile, percentile, box plot and outliers.
#Describing the attributes
summary = df.describe(include='all')

summary
```

[ ]:

```
              age         sex          cp    trestbps        chol         fbs  \
count  303.000000  303.000000  303.000000  301.000000  301.000000  303.000000
mean    54.366337    0.683168    0.966997  131.568106  246.438538    0.148515
std      9.082101    0.466011    1.032052   17.583122   51.942279    0.356198
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000
25%     47.500000    0.000000    0.000000  120.000000  211.000000    0.000000
50%     55.000000    1.000000    1.000000  130.000000  241.000000    0.000000
75%     61.000000    1.000000    2.000000  140.000000  275.000000    0.000000
max     77.000000    1.000000    3.000000  200.000000  564.000000    1.000000

          restecg     thalach      exang     oldpeak       slope          ca  \
count  303.000000  301.000000  303.000000  303.000000  303.000000  303.000000
mean     0.528053  149.528239    0.326733    1.039604    1.399340    0.729373
std      0.525860   22.930403    0.469794    1.161075    0.616226    1.022606
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  133.000000    0.000000    0.000000    1.000000    0.000000
50%      1.000000  152.000000    0.000000    0.800000    1.000000    0.000000
75%      1.000000  166.000000    1.000000    1.600000    2.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    2.000000    4.000000

             thal      target
count  303.000000  303.000000
mean     2.313531    0.544554
std      0.612277    0.498835
min      0.000000    0.000000
25%      2.000000    0.000000
50%      2.000000    1.000000
75%      3.000000    1.000000
max      3.000000    1.000000
```

[ ]:

```python
#Min, Max and range
description = df.describe()

min_vals = description.loc['min']
max_vals = description.loc['max']
range_vals = max_vals - min_vals

min_vals, max_vals, range_vals
```

```
[ ]: (age          29.0
      sex           0.0
      cp            0.0
      trestbps     94.0
      chol        126.0
      fbs           0.0
      restecg       0.0
      thalach      71.0
      exang         0.0
      oldpeak       0.0
      slope         0.0
      ca            0.0
      thal          0.0
      target        0.0
      Name: min, dtype: float64,
      age          77.0
      sex           1.0
      cp            3.0
      trestbps    200.0
      chol        564.0
      fbs           1.0
      restecg       2.0
      thalach     202.0
      exang         1.0
      oldpeak       6.2
      slope         2.0
      ca            4.0
      thal          3.0
      target        1.0
      Name: max, dtype: float64,
      age          48.0
      sex           1.0
      cp            3.0
      trestbps    106.0
      chol        438.0
      fbs           1.0
      restecg       2.0
      thalach     131.0
      exang         1.0
      oldpeak       6.2
      slope         2.0
      ca            4.0
      thal          3.0
      target        1.0
      dtype: float64)
```

[ ]:

```
#Count and data type
value_counts = df.count()
data_types = df.dtypes

value_counts, data_types
```

[ ]:

```
(age         303
 sex         303
 cp          303
 trestbps    301
 chol        301
 fbs         303
 restecg     303
 thalach     301
 exang       303
 oldpeak     303
 slope       303
 ca          303
 thal        303
 target      303
 dtype: int64,
 age           int64
 sex           int64
 cp            int64
 trestbps    float64
 chol        float64
 fbs           int64
 restecg       int64
 thalach     float64
 exang         int64
 oldpeak     float64
 slope         int64
 ca            int64
 thal          int64
 target        int64
 dtype: object)
```

[ ]:

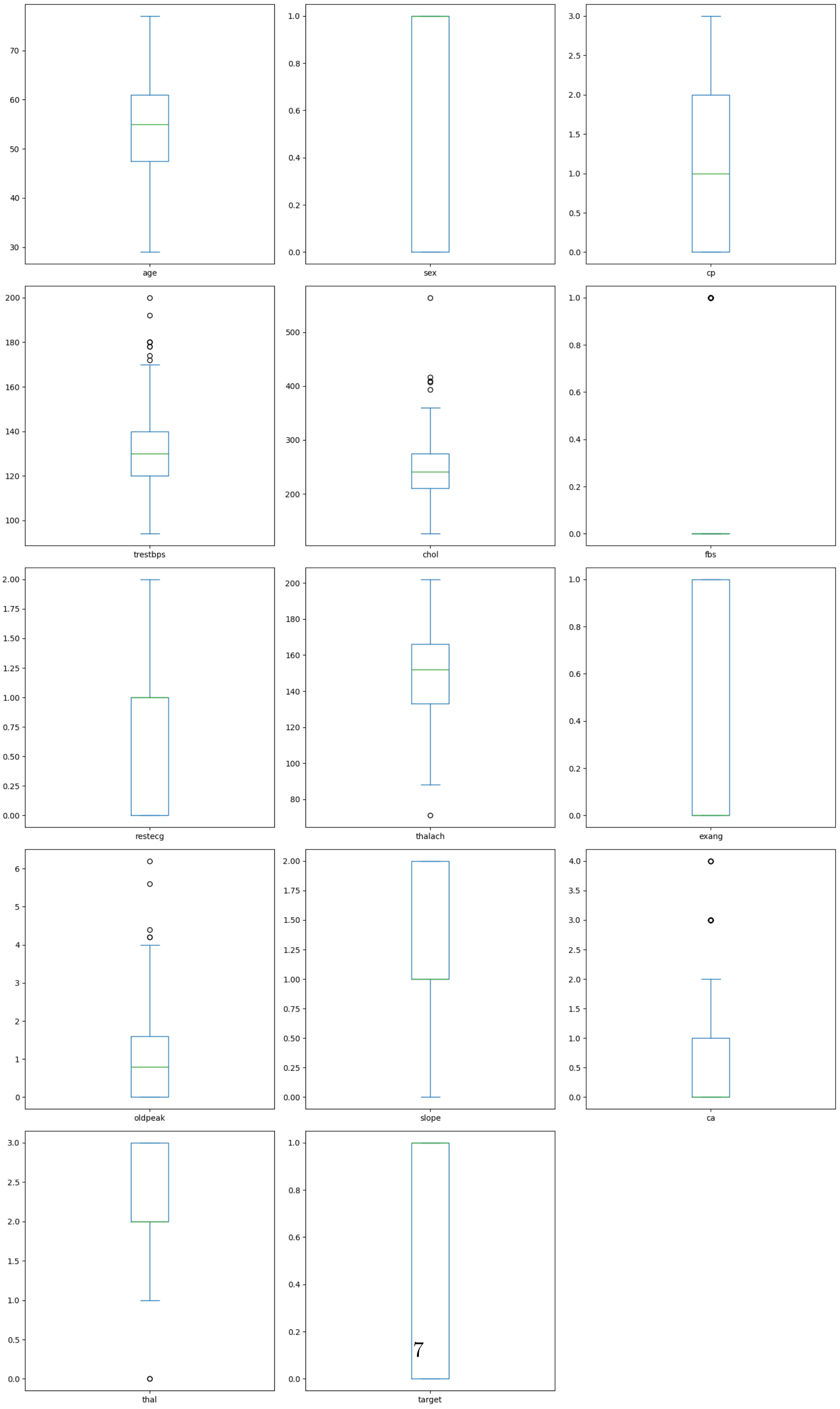
```
#Quartile, percentile and outlier
quartiles = df.quantile([0.25, 0.5, 0.75])
percentiles = df.quantile([0.1, 0.25, 0.5, 0.75, 0.9])
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = ((df < lower_bound) | (df > upper_bound))
```

```
quartiles, percentiles, outliers
```

```
[ ]: (       age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
 0.25  47.5  0.0  0.0     120.0  211.0  0.0      0.0    133.0    0.0      0.0
 0.50  55.0  1.0  1.0     130.0  241.0  0.0      1.0    152.0    0.0      0.8
 0.75  61.0  1.0  2.0     140.0  275.0  0.0      1.0    166.0    1.0      1.6

       slope   ca  thal  target
 0.25    1.0  0.0   2.0     0.0
 0.50    1.0  0.0   2.0     1.0
 0.75    2.0  1.0   3.0     1.0  ,
         age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
 0.10  42.0  0.0  0.0     110.0  188.0  0.0      0.0    116.0    0.0      0.0
 0.25  47.5  0.0  0.0     120.0  211.0  0.0      0.0    133.0    0.0      0.0
 0.50  55.0  1.0  1.0     130.0  241.0  0.0      1.0    152.0    0.0      0.8
 0.75  61.0  1.0  2.0     140.0  275.0  0.0      1.0    166.0    1.0      1.6
 0.90  66.0  1.0  2.0     152.0  309.0  1.0      1.0    177.0    1.0      2.8

       slope   ca  thal  target
 0.10    1.0  0.0   2.0     0.0
 0.25    1.0  0.0   2.0     0.0
 0.50    1.0  0.0   2.0     1.0
 0.75    2.0  1.0   3.0     1.0
 0.90    2.0  2.0   3.0     1.0  ,
         age    sex     cp  trestbps   chol    fbs  restecg  thalach  exang  \
 0     False  False  False     False  False   True    False    False  False
 1     False  False  False     False  False  False    False    False  False
 2     False  False  False     False  False  False    False    False  False
 3     False  False  False     False  False  False    False    False  False
 4     False  False  False     False  False  False    False    False  False
 ..      …      …      …         …      …      …        …        …      …
 298   False  False  False     False  False  False    False    False  False
 299   False  False  False     False  False  False    False    False  False
 300   False  False  False     False  False   True    False    False  False
 301   False  False  False     False  False  False    False    False  False
 302   False  False  False     False  False  False    False    False  False

      oldpeak  slope     ca   thal  target
 0      False  False  False  False   False
 1      False  False  False  False   False
 2      False  False  False  False   False
 3      False  False  False  False   False
 4      False  False  False  False   False
 ..       …      …      …      …       …
 298    False  False  False  False   False
 299    False  False  False  False   False
 300    False  False  False  False   False
 301    False  False  False  False   False
 302    False  False  False  False   False

 [303 rows x 14 columns])
```

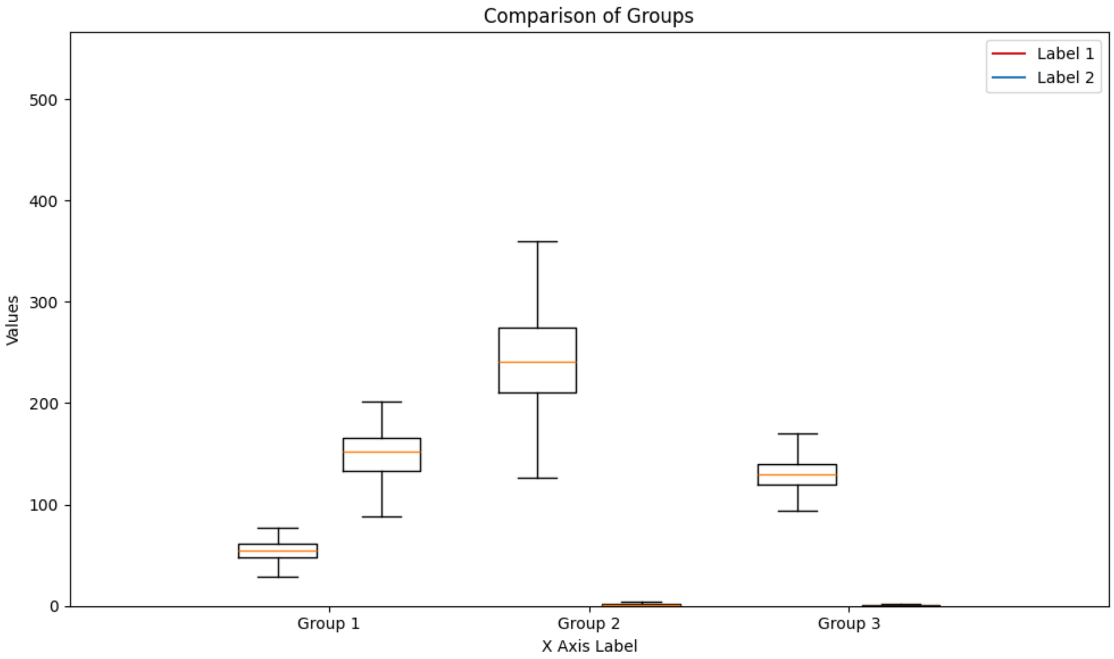
[ ]:

```
#Box plot
num_cols = len(df.select_dtypes(include=[np.number]).columns)

rows = (num_cols // 3) + 1 if num_cols % 3 != 0 else num_cols // 3
cols = 3 if num_cols > 3 else num_cols

df.plot(kind='box', figsize=(cols * 5, rows * 5), subplots=True, layout=(rows,␣
 ↪cols), sharex=False, sharey=False)
plt.tight_layout()
plt.show()
```

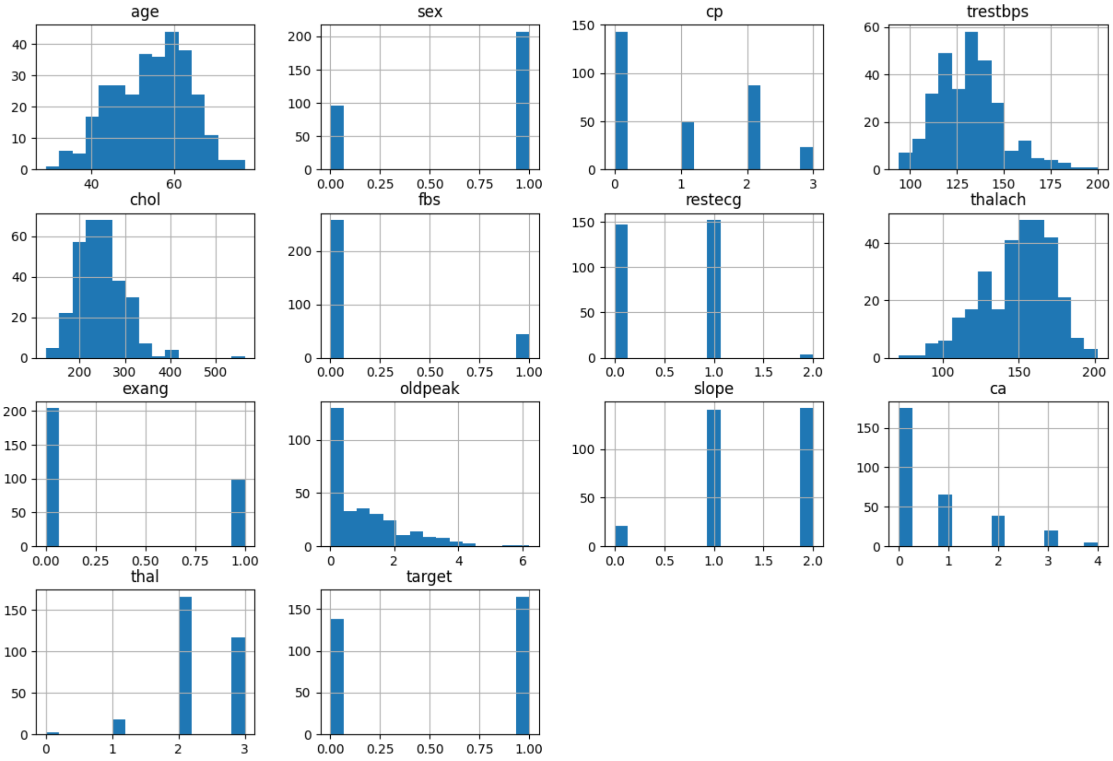

age
30
40
50
60
70
sex
0.0
0.2
0.4
0.6
0.8
1.0
cp
0.0
0.5
1.0
1.5
2.0
2.5
3.0
trestbps
100
120
140
160
180
200
chol
200
300
400
500
fbs
0.0
0.2
0.4
0.6
0.8
1.0
restecg
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
thalach
80
100
120
140
160
180
200
exang
0.0
0.2
0.4
0.6
0.8
1.0
oldpeak
0
1
2
3
4
5
6
slope
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
ca
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
thal
0.0
0.5
1.0
1.5
2.0
2.5
3.0
target
0.0
0.2
0.4
0.6
0.8
1.0

[ ]:

```
#Box plot 2
df_1 = [df['age'].dropna().values, df['chol'].dropna().values, df['trestbps'].
 ↪dropna().values]
df_2 = [df['thalach'].dropna().values, df['oldpeak'].dropna().values, df['ca'].
 ↪dropna().values]

ticks = ['Group 1', 'Group 2', 'Group 3']
plt.figure(figsize=(10, 6))

bpl = plt.boxplot(df_1, positions=np.array(range(len(df_1)))*2.0-0.4, sym='',␣
 ↪widths=0.6)
bpr = plt.boxplot(df_2, positions=np.array(range(len(df_2)))*2.0+0.4, sym='',␣
 ↪widths=0.6)

plt.plot([], c='#D7191C', label='Label 1')
plt.plot([], c='#2C7BB6', label='Label 2')
plt.title('Comparison of Groups')
plt.xlabel('X Axis Label')
plt.ylabel('Values')
plt.legend()
plt.xticks(range(0, len(ticks) * 2, 2), ticks)
plt.xlim(-2, len(ticks) * 2)
plt.ylim(0, max([df['age'].max(), df['chol'].max(), df['trestbps'].max(),␣
 ↪df['thalach'].max(), df['oldpeak'].max(), df['ca'].max()]) + 2)
plt.tight_layout()

plt.show()
```

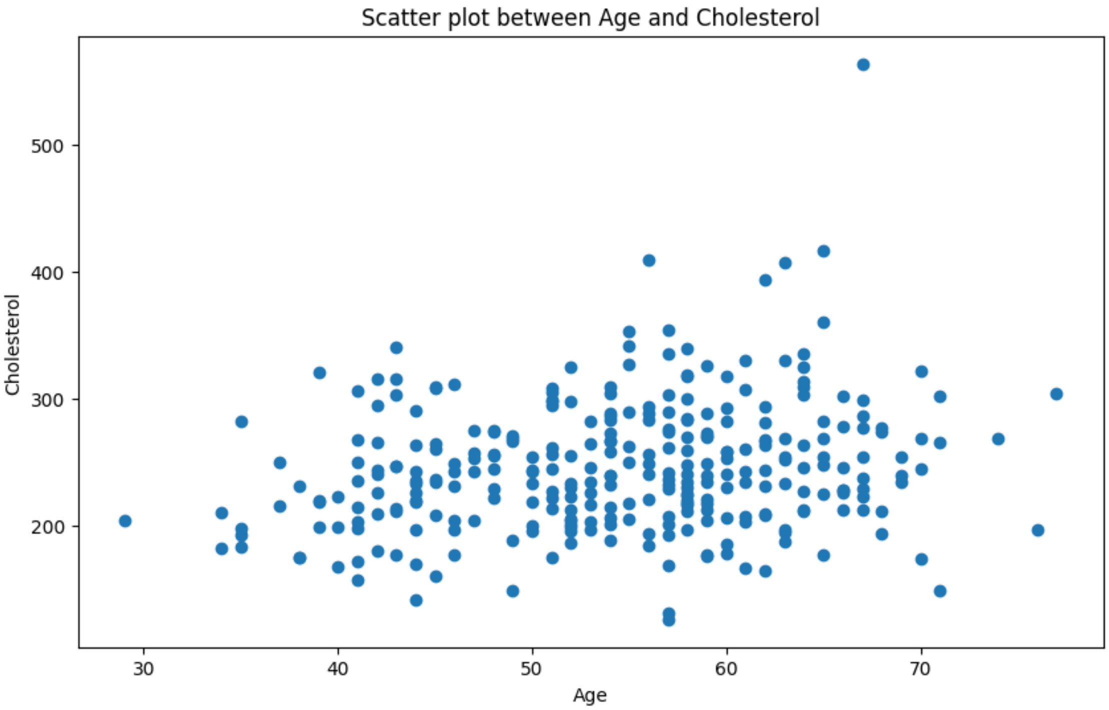


```
[ ]: #6) Give visualization of statistical description of data - in form of␣
     ↪histogram, scatter plot, pie chart
     #Histogram
     df.hist(figsize=(15, 10), bins=15)
     plt.show()
```

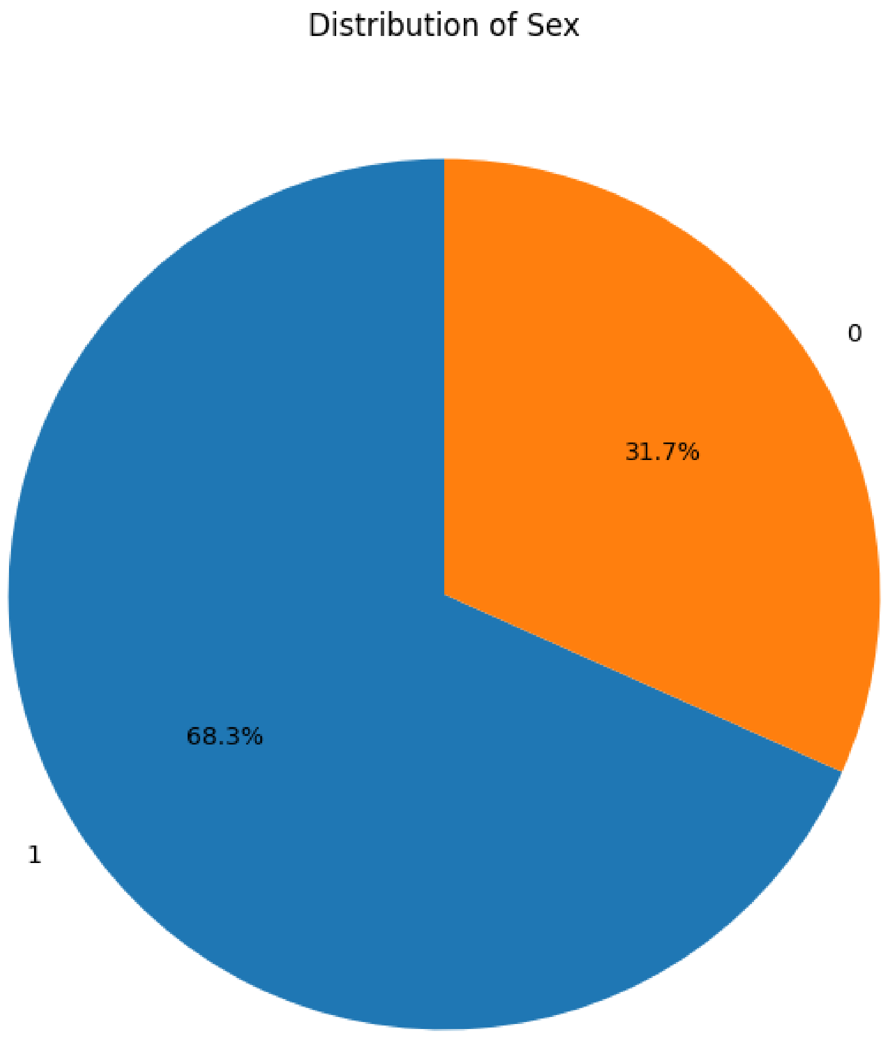


[ ]:

```
#Scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(df['age'], df['chol'])
plt.title('Scatter plot between Age and Cholesterol')
plt.xlabel('Age')
plt.ylabel('Cholesterol')
plt.show()
```

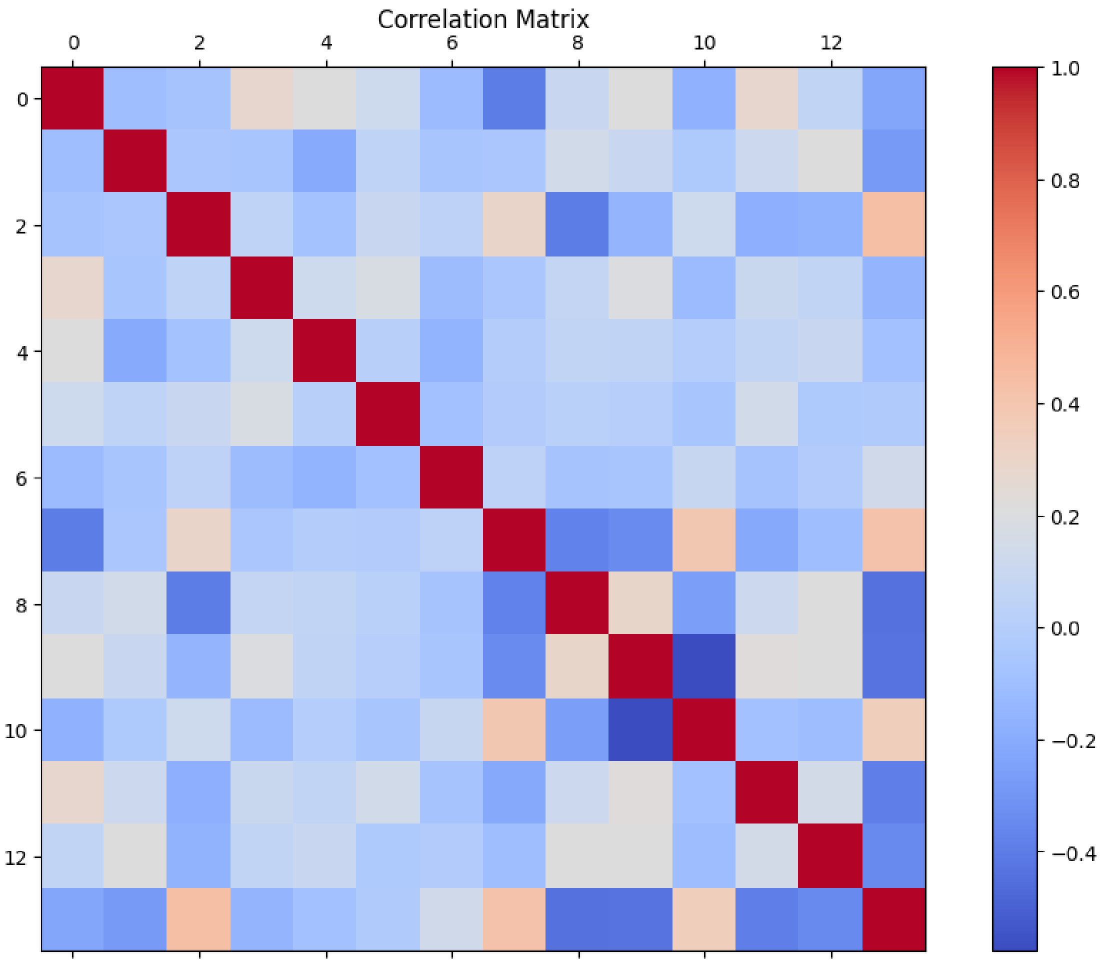


[ ]:
```
#Pie chart
df['sex'].value_counts().plot.pie(autopct='%1.1f%%', figsize=(8, 8), startangle=90)
plt.title('Distribution of Sex')
plt.ylabel('')
plt.show()
```



[ ]:

```
#Correlation matrix
correlation_matrix = df.corr()

plt.figure(figsize=(12, 8))
plt.matshow(correlation_matrix, fignum=1, cmap='coolwarm')
plt.colorbar()
plt.title('Correlation Matrix', pad=20)
plt.show()

correlation_matrix
```



```
[ ]:            age       sex        cp  trestbps      chol       fbs  \
age        1.000000 -0.098447 -0.068653  0.278869  0.210825  0.121308
sex       -0.098447  1.000000 -0.049353 -0.055802 -0.202548  0.045032
cp        -0.068653 -0.049353  1.000000  0.049158 -0.076887  0.094444
trestbps   0.278869 -0.055802  0.049158  1.000000  0.123419  0.179098
chol       0.210825 -0.202548 -0.076887  0.123419  1.000000  0.011906
fbs        0.121308  0.045032  0.094444  0.179098  0.011906  1.000000
restecg   -0.116211 -0.058196  0.044421 -0.114364 -0.153806 -0.084189
thalach   -0.395959 -0.047732  0.294994 -0.049358 -0.005942 -0.011602
exang      0.096801  0.141664 -0.394280  0.069997  0.064854  0.025665
oldpeak    0.210013  0.096093 -0.149230  0.193998  0.054733  0.005747
slope     -0.168814 -0.030711  0.119717 -0.119670 -0.000748 -0.059894
ca         0.276326  0.118261 -0.181053  0.103915  0.068297  0.137979
thal       0.068001  0.210041 -0.161736  0.067036  0.097272 -0.032019
target    -0.225439 -0.280937  0.433798 -0.148363 -0.082481 -0.028046

            restecg   thalach     exang   oldpeak     slope        ca  \
age      -0.116211 -0.395959  0.096801  0.210013 -0.168814  0.276326
sex      -0.058196 -0.047732  0.141664  0.096093 -0.030711  0.118261
cp        0.044421  0.294994 -0.394280 -0.149230  0.119717 -0.181053
trestbps -0.114364 -0.049358  0.069997  0.193998 -0.119670  0.103915
chol     -0.153806 -0.005942  0.064854  0.054733 -0.000748  0.068297
fbs      -0.084189 -0.011602  0.025665  0.005747 -0.059894  0.137979
restecg   1.000000  0.039654 -0.070733 -0.058770  0.093045 -0.072042
thalach   0.039654  1.000000 -0.376640 -0.341647  0.383786 -0.210293
exang    -0.070733 -0.376640  1.000000  0.288223 -0.257748  0.115739
oldpeak  -0.058770 -0.341647  0.288223  1.000000 -0.577537  0.222682
slope     0.093045  0.383786 -0.257748 -0.577537  1.000000 -0.080155
ca       -0.072042 -0.210293  0.115739  0.222682 -0.080155  1.000000
thal     -0.011981 -0.102924  0.206754  0.210244 -0.104764  0.151832
target    0.137230  0.419090 -0.436757 -0.430696  0.345877 -0.391724

              thal    target
age       0.068001 -0.225439
sex       0.210041 -0.280937
cp       -0.161736  0.433798
trestbps  0.067036 -0.148363
chol      0.097272 -0.082481
fbs      -0.032019 -0.028046
restecg  -0.011981  0.137230
thalach  -0.102924  0.419090
exang     0.206754 -0.436757
oldpeak   0.210244 -0.430696
slope    -0.104764  0.345877
ca        0.151832 -0.391724
thal      1.000000 -0.344029
target   -0.344029  1.000000
```

[ ]:

```
#8) Identify missing values and outlier and fill them with average.
missing_values = df.isnull().sum()

df_filled = df.fillna(df.mean())

Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

df_no_outliers = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).
 ↪any(axis=1)]

df_filled_outliers = df.where(~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 *␣
 ↪IQR))).any(axis=1), df.mean(), axis=1)
```

```
missing_values, df_filled.head(), df_no_outliers.head(), df_filled_outliers.
 ↪head()
```

```
[ ]: (age         0
      sex         0
      cp          0
      trestbps    2
      chol        2
      fbs         0
      restecg     0
      thalach     2
      exang       0
      oldpeak     0
      slope       0
      ca          0
      thal        0
      target      0
      dtype: int64,
         age  sex  cp  trestbps        chol  fbs  restecg  thalach  exang  oldpeak  \
      0   63    1   3     145.0  233.000000    1        0    150.0      0      2.3
      1   37    1   2     130.0  250.000000    0        1    187.0      0      3.5
      2   41    0   1     130.0  246.438538    0        0    172.0      0      1.4
      3   56    1   1     120.0  246.438538    0        1    178.0      0      0.8
      4   57    0   0     120.0  354.000000    0        1    163.0      1      0.6

         slope  ca  thal  target
      0      0   0     1       1
      1      0   0     2       1
      2      2   0     2       1
      3      2   0     2       1
      4      2   0     2       1  ,
         age  sex  cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
      1   37    1   2     130.0  250.0    0        1    187.0      0      3.5
      2   41    0   1     130.0    NaN    0        0    172.0      0      1.4
      3   56    1   1     120.0    NaN    0        1    178.0      0      0.8
      4   57    0   0     120.0  354.0    0        1    163.0      1      0.6
      5   57    1   0       NaN  192.0    0        1    148.0      0      0.4

         slope  ca  thal  target
      1      0   0     2       1
      2      2   0     2       1
      3      2   0     2       1
      4      2   0     2       1
      5      1   0     1       1  ,
               age       sex        cp    trestbps        chol       fbs   restecg  \
      0  54.366337  0.683168  0.966997  131.568106  246.438538  0.148515  0.528053
      1  37.000000  1.000000  2.000000  130.000000  250.000000  0.000000  1.000000
      2  41.000000  0.000000  1.000000  130.000000         NaN  0.000000  0.000000
      3  56.000000  1.000000  1.000000  120.000000         NaN  0.000000  1.000000
      4  57.000000  0.000000  0.000000  120.000000  354.000000  0.000000  1.000000

            thalach     exang   oldpeak    slope        ca      thal    target
      0  149.528239  0.326733  1.039604  1.39934  0.729373  2.313531  0.544554
      1  187.000000  0.000000  3.500000  0.00000  0.000000  2.000000  1.000000
      2  172.000000  0.000000  1.400000  2.00000  0.000000  2.000000  1.000000
      3  178.000000  0.000000  0.800000  2.00000  0.000000  2.000000  1.000000
      4  163.000000  1.000000  0.600000  2.00000  0.000000  2.000000  1.000000  )
```