In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv("heart1.csv")
```

In [5]:
```
df.shape
```

Out[5]: (1025, 14)

In [7]:
```
df
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	52	1	0	125.0	212	0	1	168	0	1.0	2	2
1	52	1	0	125.0	87	0	1	168	0	1.0	2	2
2	70	1	0	145.0	174	0	1	125	1	2.6	0	0
3	61	1	0	148.0	203	0	1	161	0	0.0	2	1
4	62	0	0	138.0	294	1	1	106	0	1.9	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140.0	221	0	1	164	1	0.0	2	0
1021	60	1	0	125.0	258	0	0	141	1	2.8	1	1
1022	47	1	0	110.0	275	0	0	118	1	1.0	1	1
1023	50	0	0	110.0	254	0	0	159	0	0.0	2	0
1024	54	1	0	120.0	188	0	1	113	0	1.4	1	1

1025 rows × 14 columns

In [9]:
```
print(df.to_string())
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125.0	212	0	1	168	0	1.0	2	2	3	0
1	52	1	0	125.0	87	0	1	168	0	1.0	2	2	3	0
2	70	1	0	145.0	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148.0	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138.0	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100.0	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	NaN	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160.0	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120.0	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122.0	286	0	0	116	1	3.2	1	2	2	0
10	71	0	0	112.0	149	0	1	125	0	1.6	1	0	2	1
11	43	0	0	132.0	341	1	0	136	1	3.0	1	0	3	0
12	34	0	1	118.0	210	0	1	192	0	0.7	2	0	2	1
13	51	1	0	140.0	298	0	1	122	1	4.2	1	3	3	0
14	52	1	0	128.0	204	1	1	156	1	1.0	1	0	0	0
15	34	0	1	118.0	210	0	1	192	0	0.7	2	0	2	1
16	51	0	2	140.0	308	0	0	142	0	1.5	2	1	2	1
17	54	1	0	124.0	266	0	0	109	1	2.2	1	1	3	0
18	50	0	1	120.0	244	0	1	162	0	1.1	2	0	2	1
19	58	1	2	140.0	211	1	0	165	0	0.0	2	0	2	1
20	60	1	2	140.0	185	0	0	155	0	3.0	1	0	2	0
21	67	0	0	106.0	223	0	1	142	0	0.3	2	2	2	1
22	45	1	0	104.0	208	0	0	148	1	3.0	1	0	2	1
23	63	0	2	135.0	252	0	0	172	0	0.0	2	0	2	1
24	42	0	2	120.0	209	0	1	173	0	0.0	1	0	2	1
25	61	0	0	145.0	307	0	0	146	1	1.0	1	0	3	0
26	44	1	2	130.0	233	0	1	179	1	0.4	2	0	2	1
27	58	0	1	136.0	319	1	0	152	0	0.0	2	2	2	0
28	56	1	2	130.0	256	1	0	142	1	0.6	1	1	1	0

29	55	0	0	180.0	327	0	2	117	1	3.4	1	0	2	0
30	44	1	0	120.0	169	0	1	144	1	2.8	0	0	1	0
31	50	0	1	120.0	244	0	1	162	0	1.1	2	0	2	1
32	57	1	0	130.0	131	0	1	115	1	1.2	1	1	3	0
33	70	1	2	160.0	269	0	1	112	1	2.9	1	1	3	0
34	50	1	2	129.0	196	0	1	163	0	0.0	2	0	2	1
35	46	1	2	150.0	231	0	1	147	0	3.6	1	0	2	0
36	51	1	3	125.0	213	0	0	125	1	1.4	2	1	2	1
37	59	1	0	138.0	271	0	0	182	0	0.0	2	0	2	1
38	64	1	0	128.0	263	0	1	105	1	0.2	1	1	3	1
39	57	1	2	128.0	229	0	0	150	0	0.4	1	1	3	0
40	65	0	2	160.0	360	0	0	151	0	0.8	2	0	2	1
41	54	1	2	120.0	258	0	0	147	0	0.4	1	0	3	1
42	61	0	0	130.0	330	0	0	169	0	0.0	2	0	2	0
43	46	1	0	120.0	249	0	0	144	0	0.8	2	0	3	0
44	55	0	1	132.0	342	0	1	166	0	1.2	2	0	2	1
45	42	1	0	140.0	226	0	1	178	0	0.0	2	0	2	1
46	41	1	1	135.0	203	0	1	132	0	0.0	1	0	1	1
47	66	0	0	178.0	228	1	1	165	1	1.0	1	2	3	0
48	66	0	2	146.0	278	0	0	152	0	0.0	1	1	2	1
49	60	1	0	117.0	230	1	1	160	1	1.4	2	2	3	0
50	58	0	3	150.0	283	1	0	162	0	1.0	2	0	2	1
51	57	0	0	140.0	241	0	1	123	1	0.2	1	0	3	0
52	38	1	2	138.0	175	0	1	173	0	0.0	2	4	2	1
53	49	1	2	120.0	188	0	1	139	0	2.0	1	3	3	0
54	55	1	0	140.0	217	0	1	111	1	5.6	0	0	3	0
55	55	1	0	140.0	217	0	1	111	1	5.6	0	0	3	0
56	56	1	3	120.0	193	0	0	162	0	1.9	1	0	3	1
57	48	1	1	130.0	245	0	0	180	0	0.2	1	0	2	1
58	67	1	2	152.0	212	0	0	150	0	0.8	1	0	3	0

```
1019   47    1   0     112.0   204    0        1      143      0      0.1      2
0     2       1
1020   59    1   1     140.0   221    0        1      164      1      0.0      2
0     2       1
1021   60    1   0     125.0   258    0        0      141      1      2.8      1
1     3       0
1022   47    1   0     110.0   275    0        0      118      1      1.0      1
1     2       0
1023   50    0   0     110.0   254    0        0      159      0      0.0      2
0     2       1
1024   54    1   0     120.0   188    0        1      113      0      1.4      1
1     3       0
```

In [11]:
```
df.head()
```

Out[11]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125.0	212	0	1	168	0	1.0	2	2	3
1	52	1	0	125.0	87	0	1	168	0	1.0	2	2	3
2	70	1	0	145.0	174	0	1	125	1	2.6	0	0	3
3	61	1	0	148.0	203	0	1	161	0	0.0	2	1	3
4	62	0	0	138.0	294	1	1	106	0	1.9	1	3	2

In [13]:
```
df.head(7)
```

Out[13]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125.0	212	0	1	168	0	1.0	2	2	3
1	52	1	0	125.0	87	0	1	168	0	1.0	2	2	3
2	70	1	0	145.0	174	0	1	125	1	2.6	0	0	3
3	61	1	0	148.0	203	0	1	161	0	0.0	2	1	3
4	62	0	0	138.0	294	1	1	106	0	1.9	1	3	2
5	58	0	0	100.0	248	0	0	122	0	1.0	1	0	2
6	58	1	0	NaN	318	0	2	140	0	4.4	0	3	1

In [15]:
```
df.tail()
```

Out[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
1020	59	1	1	140.0	221	0	1	164	1	0.0	2	0
1021	60	1	0	125.0	258	0	0	141	1	2.8	1	1
1022	47	1	0	110.0	275	0	0	118	1	1.0	1	1
1023	50	0	0	110.0	254	0	0	159	0	0.0	2	0
1024	54	1	0	120.0	188	0	1	113	0	1.4	1	1

In [17]:
```
df.tail(3)
```

Out[17]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
1022	47	1	0	110.0	275	0	0	118	1	1.0	1	1
1023	50	0	0	110.0	254	0	0	159	0	0.0	2	0
1024	54	1	0	120.0	188	0	1	113	0	1.4	1	1

In [19]:
```
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  1024 non-null   float64
 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(2), int64(12)
memory usage: 112.2 KB
```

In [21]:
```
df.describe()
```

Out[21]:

	age	sex	cp	trestbps	chol	fbs	1
count	1025.000000	1025.000000	1025.000000	1024.000000	1025.000000	1025.000000	
mean	54.433171	0.695610	0.942439	131.614258	245.886829	0.148293	
std	9.072498	0.460373	1.029641	17.515881	51.813677	0.355563	
min	29.000000	0.000000	0.000000	94.000000	87.000000	0.000000	
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	
50%	56.000000	1.000000	1.000000	130.000000	240.000000	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	

In [23]:
```
df.dtypes
```

Out[23]:
```
age           int64
sex           int64
cp            int64
trestbps    float64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [25]:
```
df['age'].mean()
```

Out[25]:
```
54.433170731707314
```

In [27]:
```
df['chol'].median()
```

Out[27]:
```
240.0
```

In [29]:
```
df['trestbps'].min()
```

Out[29]:
```
94.0
```

In [31]:
```
df['trestbps'].max()
```

Out[31]:
```
200.0
```

In [33]:
```
df[df==0].count()
```

Out[33]:
```
age          0
sex        312
cp         497
trestbps     0
chol         0
fbs        873
restecg    496
thalach      0
exang      681
oldpeak    329
slope       73
ca         577
thal         7
target     499
dtype: int64
```

In [35]:
```
df['age'].sum()
```

Out[35]:
```
55794
```

In [37]:
```
df['age'].count()
```

Out[37]:
```
1025
```

In [39]:
```
df.isna()
```

Out[39]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
1020	False	False	False	False	False	False	False	False	False	False	False
1021	False	False	False	False	False	False	False	False	False	False	False
1022	False	False	False	False	False	False	False	False	False	False	False
1023	False	False	False	False	False	False	False	False	False	False	False
1024	False	False	False	False	False	False	False	False	False	False	False

1025 rows × 14 columns

In [41]:
```
df.isnull().sum()
```

Out[41]:
```
age         0
sex         0
cp          0
trestbps    1
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [43]:
```
df=df.fillna(df.median())
```

In [45]:
```
df.isna().sum()
```

Out[45]:
```
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 [47]:
```
df.duplicated()
```

Out[47]:
```
0       False
1       False
2       False
3       False
4       False
        ...
1020     True
1021     True
1022     True
1023     True
1024     True
Length: 1025, dtype: bool
```

In [49]:
```
df.duplicated().sum()
```

Out[49]:
```
721
```

In [51]:
```
df=df.drop_duplicates()
```

In [53]:
```
df.shape
```

Out[53]: (304, 14)

In [55]:
```
df
```

Out[55]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	52	1	0	125.0	212	0	1	168	0	1.0	2	2	
1	52	1	0	125.0	87	0	1	168	0	1.0	2	2	
2	70	1	0	145.0	174	0	1	125	1	2.6	0	0	
3	61	1	0	148.0	203	0	1	161	0	0.0	2	1	
4	62	0	0	138.0	294	1	1	106	0	1.9	1	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	
723	68	0	2	120.0	211	0	0	115	0	1.5	1	0	
733	44	0	2	108.0	141	0	1	175	0	0.6	1	0	
739	52	1	0	128.0	255	0	1	161	1	0.0	2	1	
843	59	1	3	160.0	273	0	0	125	0	0.0	2	0	
878	54	1	0	120.0	188	0	1	113	0	1.4	1	1	

304 rows × 14 columns

In [57]:
```
df.dtypes
```

Out[57]:
```
age           int64
sex           int64
cp            int64
trestbps    float64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [59]:
```
df=df.astype({'trestbps':'int','oldpeak':'int'})
```

In [61]:
```
df.dtypes
```

Out[61]:
```
age         int64
sex         int64
cp          int64
trestbps    int32
chol        int64
fbs         int64
restecg     int64
thalach     int64
exang       int64
oldpeak     int32
slope       int64
ca          int64
thal        int64
target      int64
dtype: object
```

In [63]:
```
import matplotlib.pyplot as plt
```

In [64]:
```
from sklearn.model_selection import train_test_split
```

In [65]:
```
x=df.drop('target',axis='columns')
```

In [67]:
```
x
```

Out[67]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	52	1	0	125	212	0	1	168	0	1	2	2	
1	52	1	0	125	87	0	1	168	0	1	2	2	
2	70	1	0	145	174	0	1	125	1	2	0	0	
3	61	1	0	148	203	0	1	161	0	0	2	1	
4	62	0	0	138	294	1	1	106	0	1	1	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	
723	68	0	2	120	211	0	0	115	0	1	1	0	
733	44	0	2	108	141	0	1	175	0	0	1	0	
739	52	1	0	128	255	0	1	161	1	0	2	1	
843	59	1	3	160	273	0	0	125	0	0	2	0	
878	54	1	0	120	188	0	1	113	0	1	1	1	

304 rows × 13 columns

In [68]:
```
y=df['target']
```

In [69]:
```
y
```

Out[69]:
```
0      0
1      0
2      0
3      0
4      0
      ..
723    1
733    1
739    0
843    0
878    0
Name: target, Length: 304, dtype: int64
```

In [70]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [77]:
```
x_train.shape
```

Out[77]:
```
(228, 13)
```

In [79]:
```
x_test.shape
```

Out[79]:
```
(76, 13)
```

In [81]:
```
y_train.shape
```

Out[81]:
```
(228,)
```

In [83]:
```
y_test.shape
```

Out[83]:
```
(76,)
```

In [85]:
```
y_test.shape
```

Out[85]:
```
(76,)
```

In [87]:
```
from sklearn.linear_model import LogisticRegression
```

In [88]:
```
reg = LogisticRegression()
```

In [91]:
```
reg.fit(x_train,y_train)
```

```
C:\Users\PRATIK\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[91]:
```
LogisticRegression()
```

In [93]:
```
y_predict=reg.predict(x_test)
```

In [95]:
```
y_predict.shape
```

Out[95]: (76,)

In [97]:
```
from sklearn.metrics import accuracy_score
```

In [99]:
```
print(accuracy_score(y_test,y_predict))
```

```
0.8552631578947368
```

In [101…
```
from sklearn.metrics import classification_report
```

In [103…
```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.85      0.77      0.81        30
           1       0.86      0.91      0.88        46

    accuracy                           0.86        76
   macro avg       0.85      0.84      0.85        76
weighted avg       0.86      0.86      0.85        76
```

In [105…
```
from sklearn.metrics import confusion_matrix
```

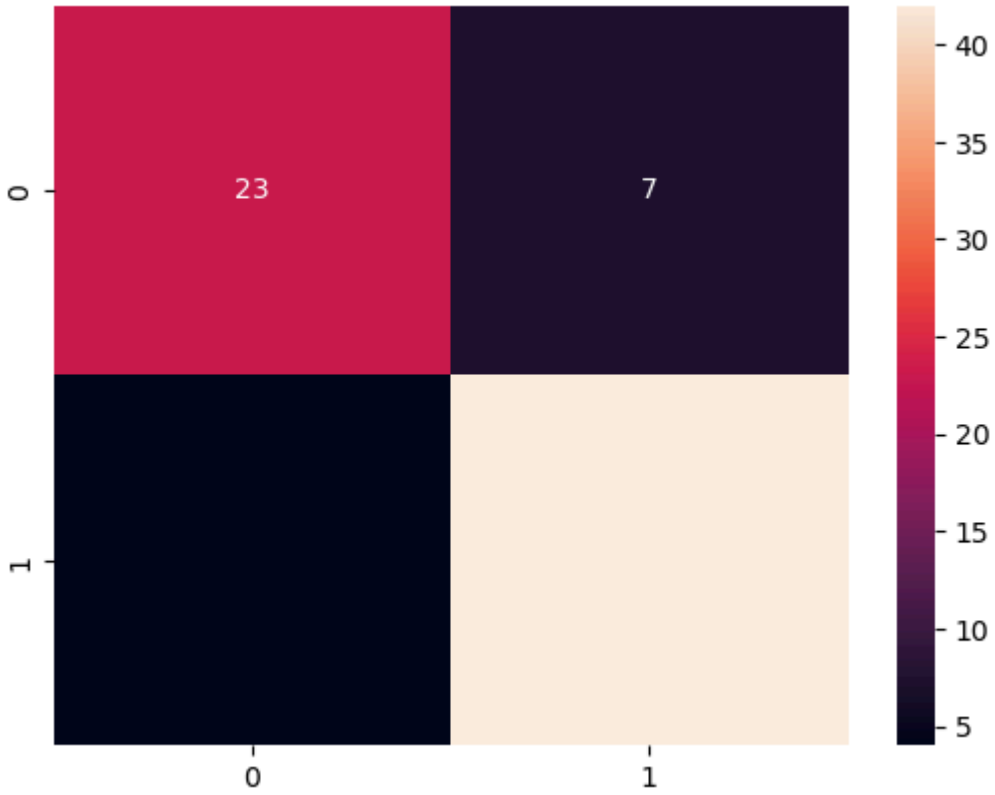
In [107…
```
print(confusion_matrix(y_test,y_predict))
```

```
[[23  7]
 [ 4 42]]
```

In [109…
```
import seaborn as sns
```

In [111…
```
sns.heatmap(confusion_matrix(y_test,y_predict),annot=True)
```

Out[111… <Axes: >



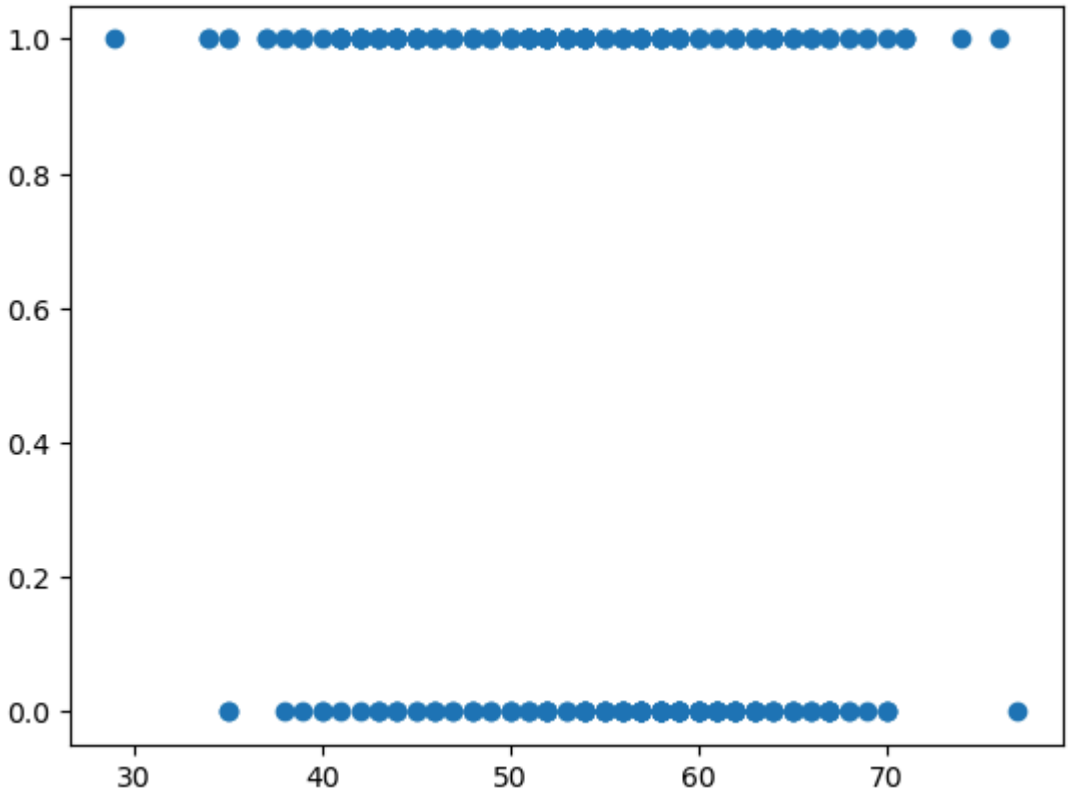
In [113…
```
import matplotlib.pyplot as plt
```

In [115…
```
x=df['age']
```

In [117…
```
y=df['target']
```

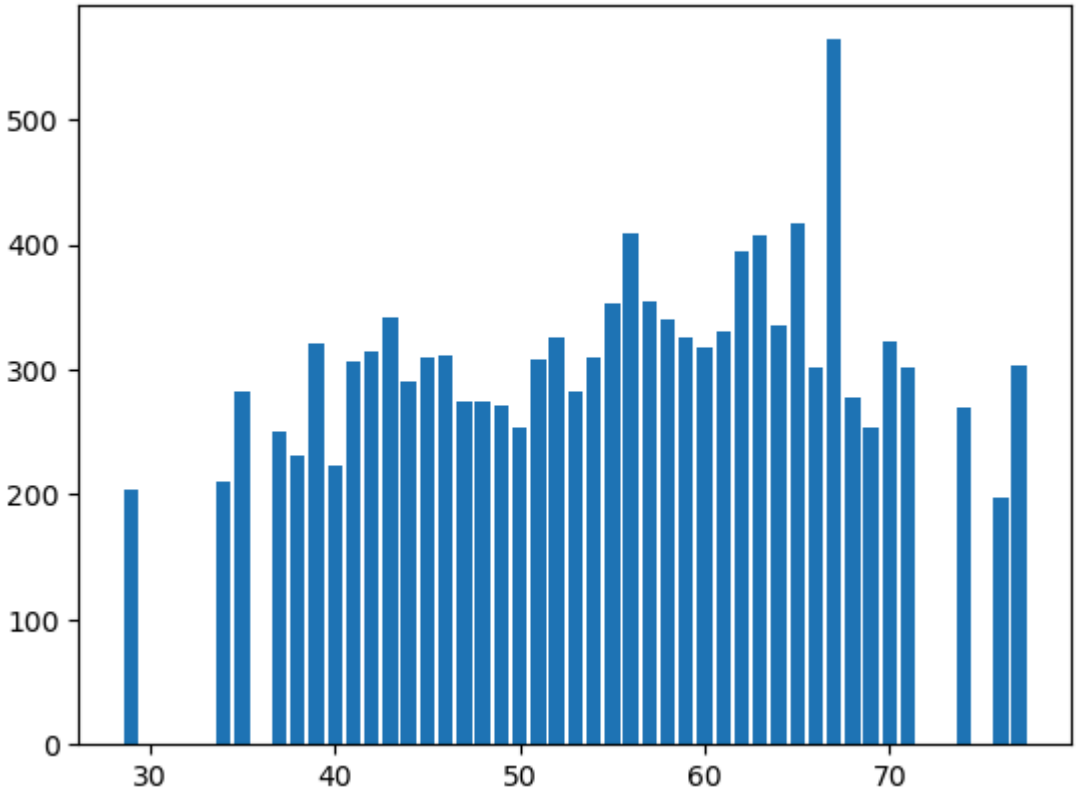
In [119…
```
plt.scatter(x,y)
```

Out[119…
```
<matplotlib.collections.PathCollection at 0x2430569e390>
```



In [121…
```
plt.bar(df['age'],df['chol'])
```

Out[121…
```
<BarContainer object of 304 artists>
```



In [ ]:

In [ ]:

In [ ]: