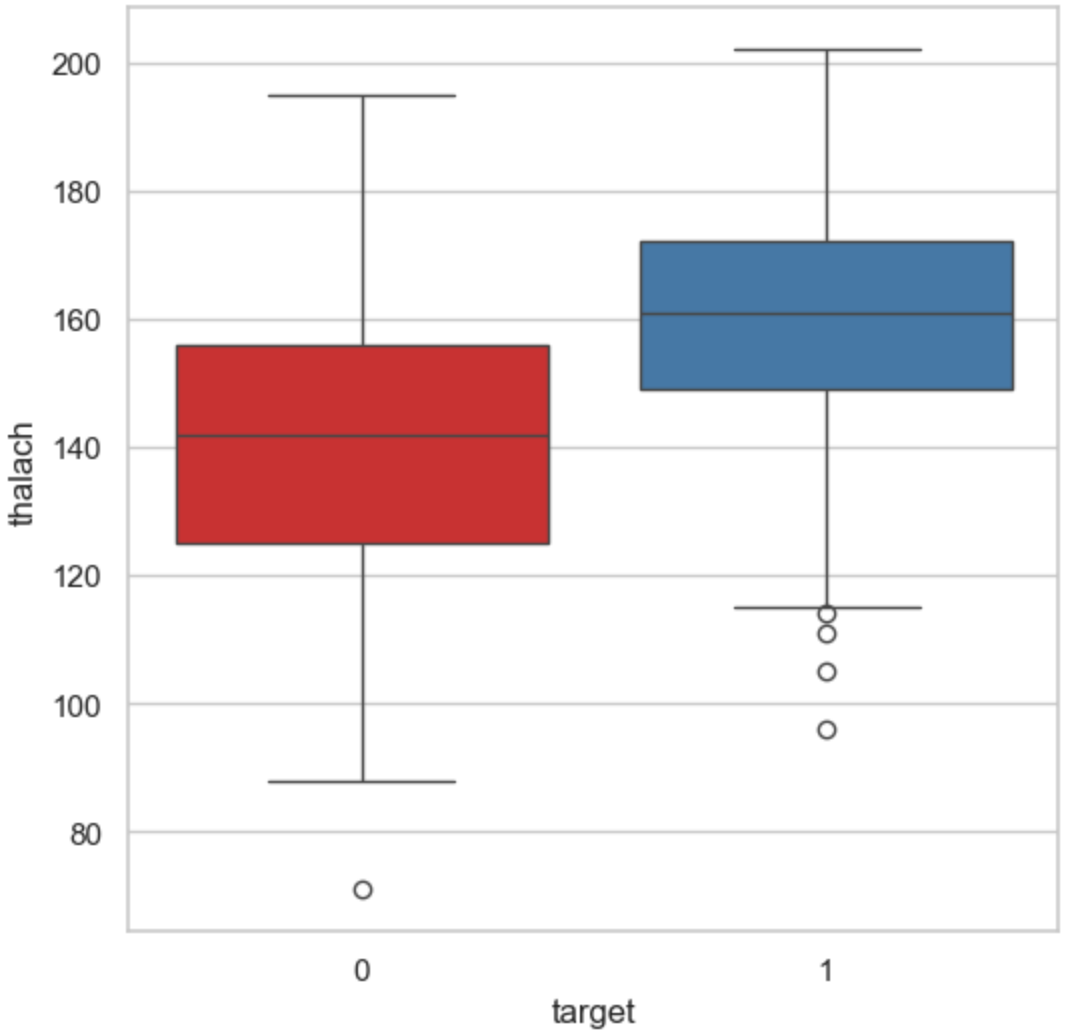


Interpretation

The above boxplot confirms our finding that people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).

Findings of Bivariate Analysis

Findings of Bivariate Analysis are as follows –

- There is no variable which has strong positive correlation with `target` variable.
- There is no variable which has strong negative correlation with `target` variable.
- There is no correlation between `target` and `fbs`.
- The `cp` and `thalach` variables are mildly positively correlated with `target` variable.
- We can see that the `thalach` variable is slightly negatively skewed.

- The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).

- The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).

#Multivariate analysis

- The objective of the multivariate analysis is to discover patterns and relationships in the dataset.

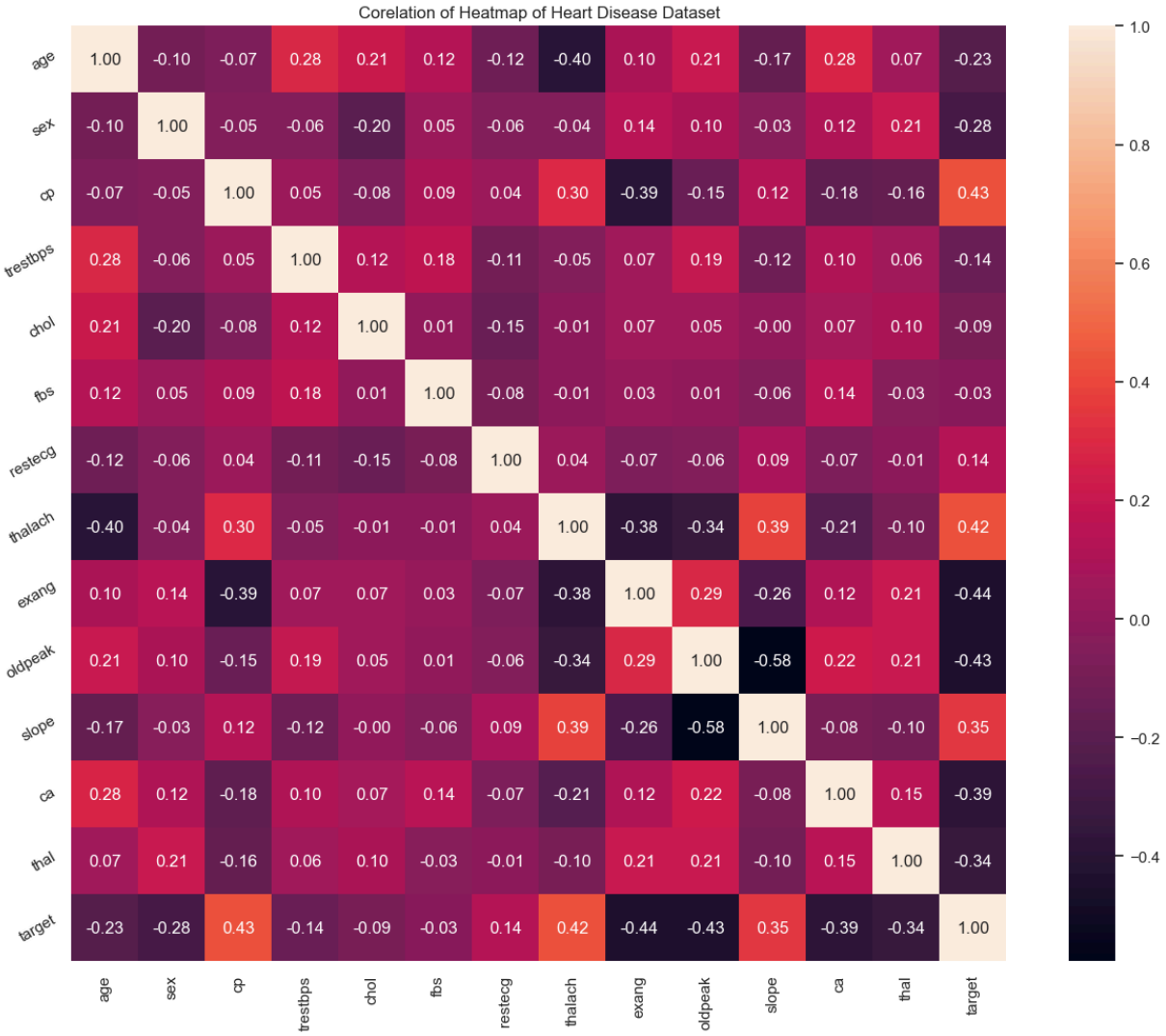
Discover patterns and relationships

- An important step in EDA is to discover patterns and relationships between variables in the dataset.
- I will use `heat map` and `pair plot` to discover the patterns and relationships in the dataset.
- First of all, I will draw a `heat map`.

#Heat map

In [113…

```
plt.figure(figsize=(16,12))
plt.title("Corelation of Heatmap of Heart Disease Dataset")
a= sns.heatmap(correlation, square=True, annot=True, fmt='.2f',linecolor='white')
a.set_xticklabels(a.get_xticklabels(), rotation=90)
a.set_yticklabels(a.get_yticklabels(), rotation=30)
plt.show()
```



- correlation: This should be a DataFrame or matrix containing the correlation values.
- `square=True` : This makes each cell square-shaped.
- `annot=True` : This displays the actual correlation values inside the heatmap cells.
- `fmt='.2f'` : This controls the format of the annotations, limiting them to two decimal places.
- `linecolor='white'` : Adds white grid lines between the cells.

Interpretation

From the above correlation heat map, we can conclude that :-

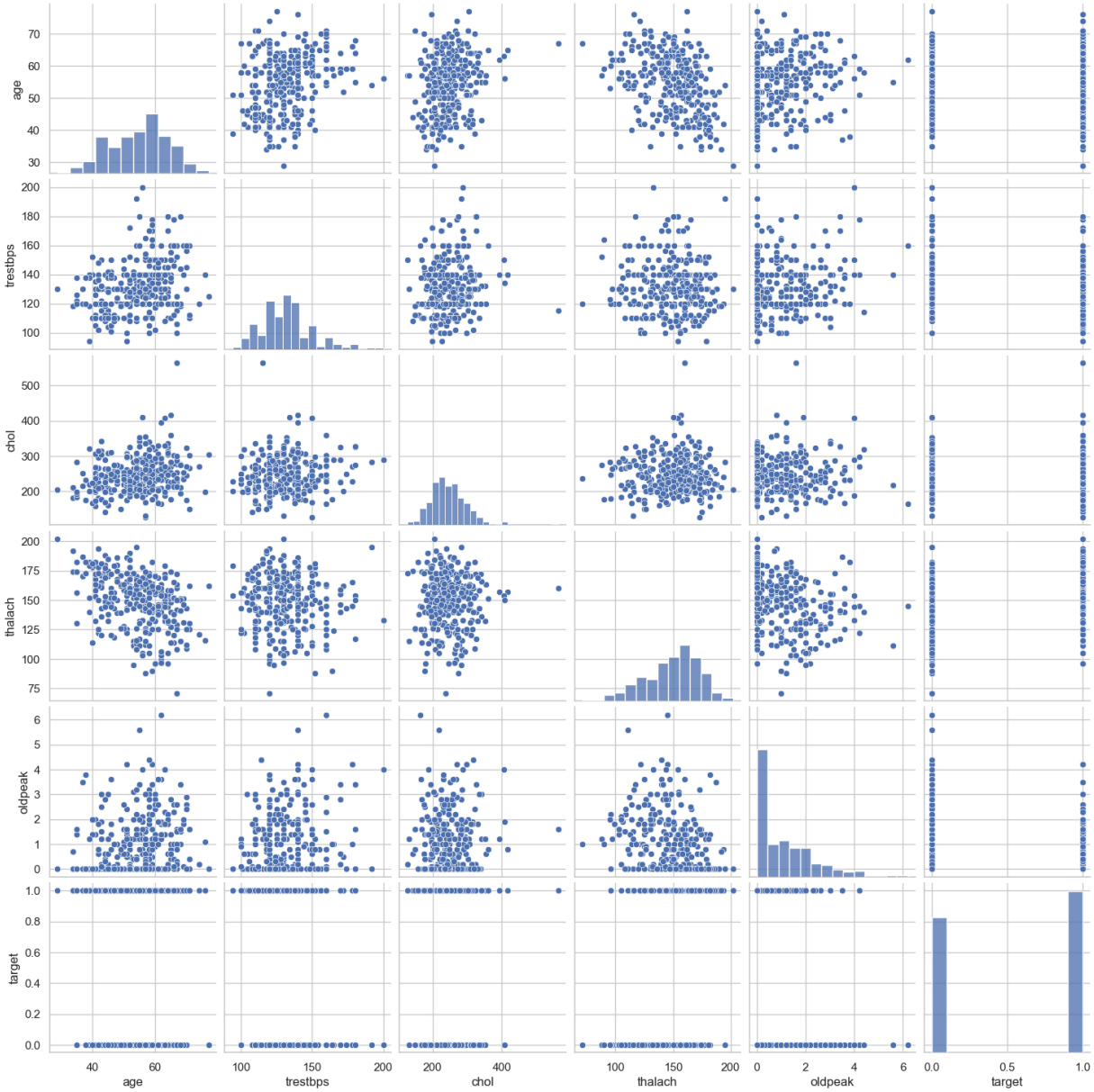
- `target` and `cp` variable are mildly positively correlated (correlation coefficient = 0.43).
- `target` and `thalach` variable are also mildly positively correlated (correlation coefficient = 0.42).
- `target` and `slope` variable are weakly positively correlated (correlation coefficient = 0.35).

- `target` and `exang` variable are mildly negatively correlated (correlation coefficient = -0.44).
- `target` and `oldpeak` variable are also mildly negatively correlated (correlation coefficient = -0.43).
- `target` and `ca` variable are weakly negatively correlated (correlation coefficient = -0.39).
- `target` and `thal` variable are also waekly negatively correlated (correlation coefficient = -0.34).

#Pair Plot

In [117…

```
num_var =['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'target']
sns.pairplot(df[num_var], kind='scatter',diag_kind='hist')
plt.show()
```



- df[num_var]: DataFrame df se specified numerical columns ko select karta hai.
- kind='scatter': Off-diagonal subplots par scatter plots banata hai, jo pairs of variables ke beech relation dikhata hai.
- diag_kind='hist': Diagonal plots par histograms banata hai jo har variable ki distribution ko show karta hai.

Comment

- I have defined a variable `num_var`. Here `age`, `trestbps`, `chol`, `thalach` and `oldpeak` are numerical variables and `target` is the categorical variable.
- So, I wll check relationships between these variables.

Pairplot Kya Dikhata Hai:

- Scatter Plots: Diagonal ke bahar ke subplots do variables ke beech relationship ko dikhate hain. Jaise, 'age' aur 'thalach' ke beech ka plot.
- Histograms: Diagonal par histograms har variable ki distribution ko dikhate hain. Jaise, 'age' ke diagonal plot par age distribution dikhegi.

Analysis of `Age` and other veriables

Check the number of unique values in `age` variable

In [123…
```
df['age'].nunique()
```

Out[123…
```
41
```

#view statistical summary of age veriable

In [125…
```
df['age'].describe()
```

Out[125…
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

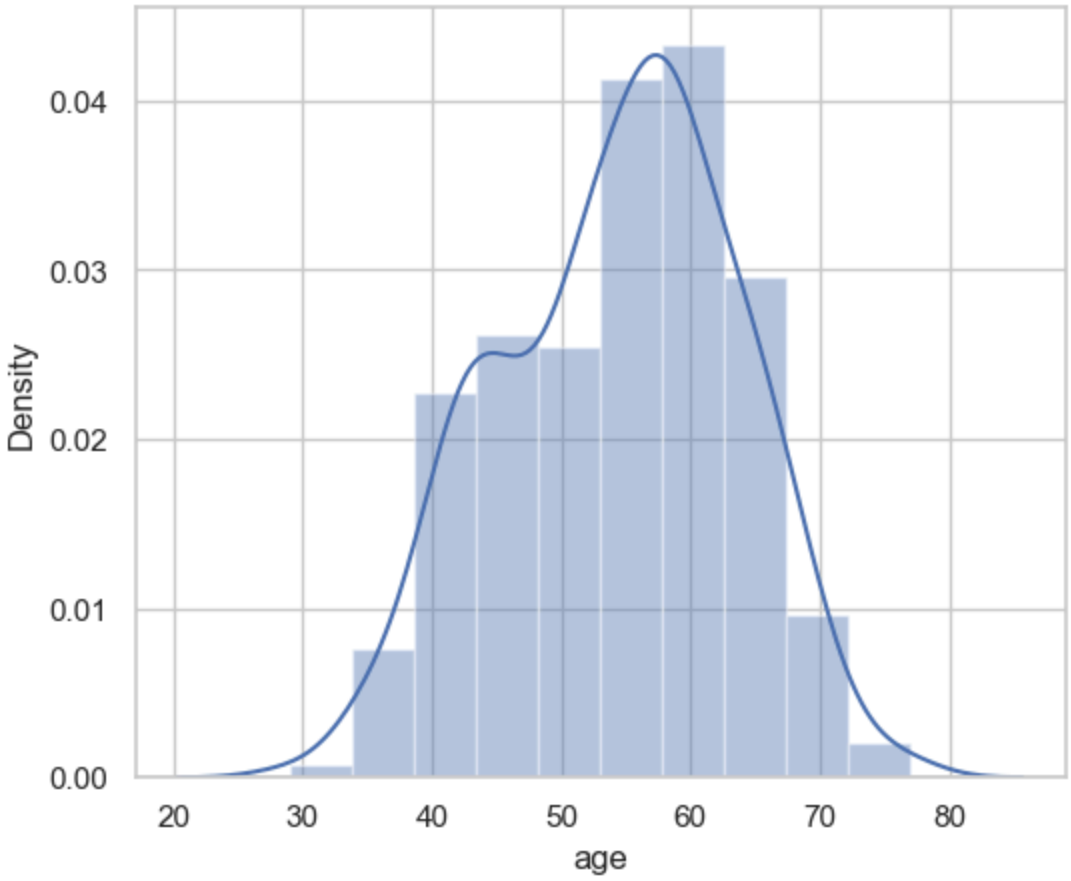
Interpretation

- The mean value of the `age` variable is 54.37 years.
- The minimum and maximum values of `age` are 29 and 77 years.

Plot the distribution of `age` variable

Now, I will plot the distribution of `age` variable to view the statistical properties.

In [128…
```
f, ax = plt.subplots(figsize=(6,5))
x = df['age']
ax = sns.distplot(x,bins=10)
plt.show()
```



Interpretation

- The `age` variable distribution is approximately normal.

Analyze `age` and `target` variable

Visualize frequency distribution of `age` variable wrt `target`

In [132…
```
df['age'].unique()
```

Out[132…
```
array([63, 37, 41, 56, 57, 44, 52, 54, 48, 49, 64, 58, 50, 66, 43, 69, 59,
       42, 61, 40, 71, 51, 65, 53, 46, 45, 39, 47, 62, 34, 35, 29, 55, 60,
       67, 68, 74, 76, 70, 38, 77], dtype=int64)
```

In [133…
```
df['age'].nunique()
```

Out[133…
```
41
```

In [134…
```
df.groupby('age')['target'].value_counts()
```

Out[134…

```
age  target
29   1         1
34   1         2
35   0         2
     1         2
37   1         2
               ..
70   1         1
71   1         3
74   1         1
76   1         1
77   0         1
Name: count, Length: 75, dtype: int64
```

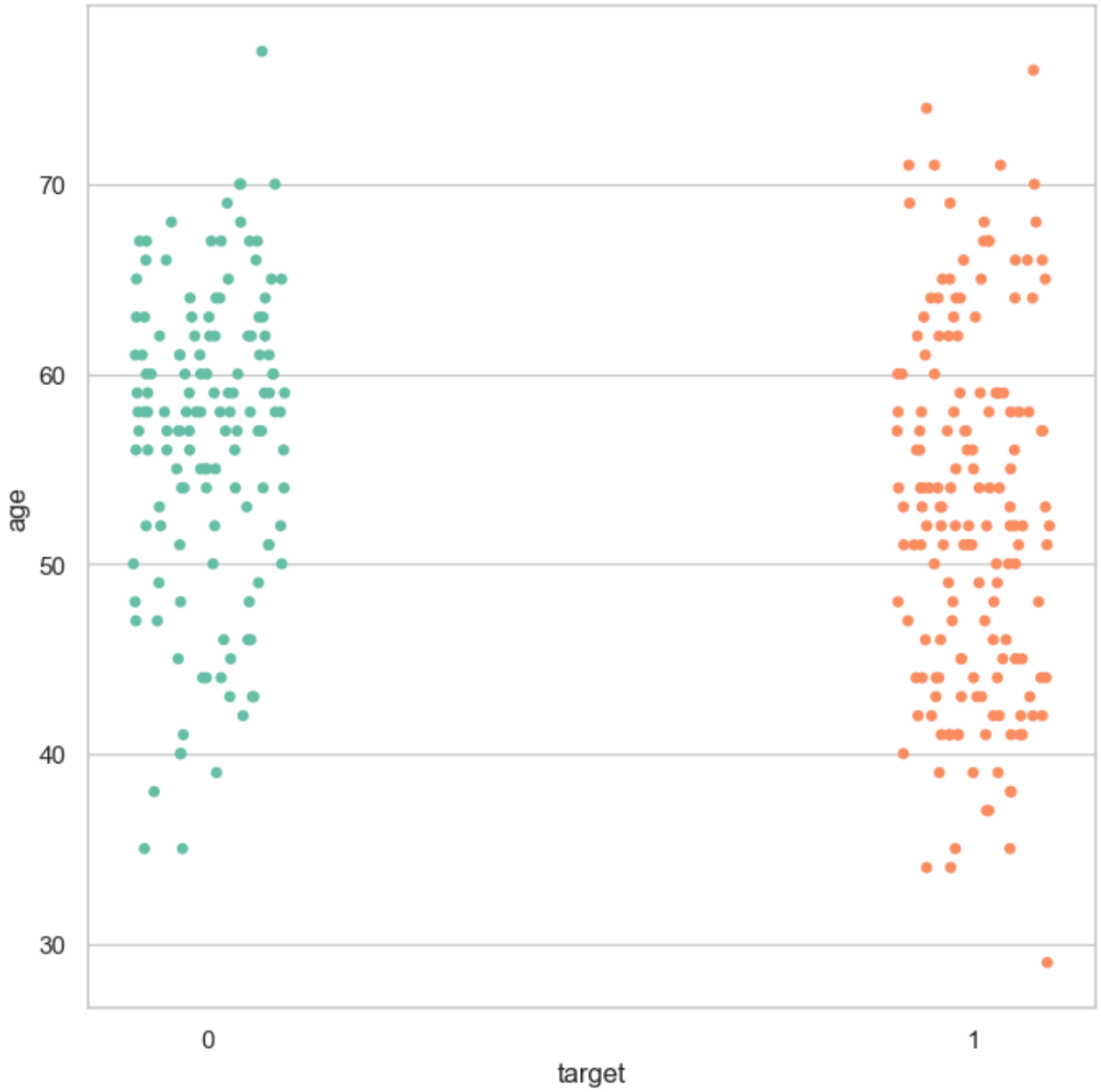
- 29 1 1: For age 29, there is 1 occurrence where the target is 1.
- 34 1 2: For age 34, there are 2 occurrences where the target is 1.
- 35 0 2: For age 35, there are 2 occurrences where the target is 0.
- 35 1 2: For age 35, there are 2 occurrences where the target is 1.

In [136…

```
f,ax = plt.subplots(figsize=(8,8))
sns.stripplot(x='target',y='age',data= df, palette='Set2')
plt.plot()
```

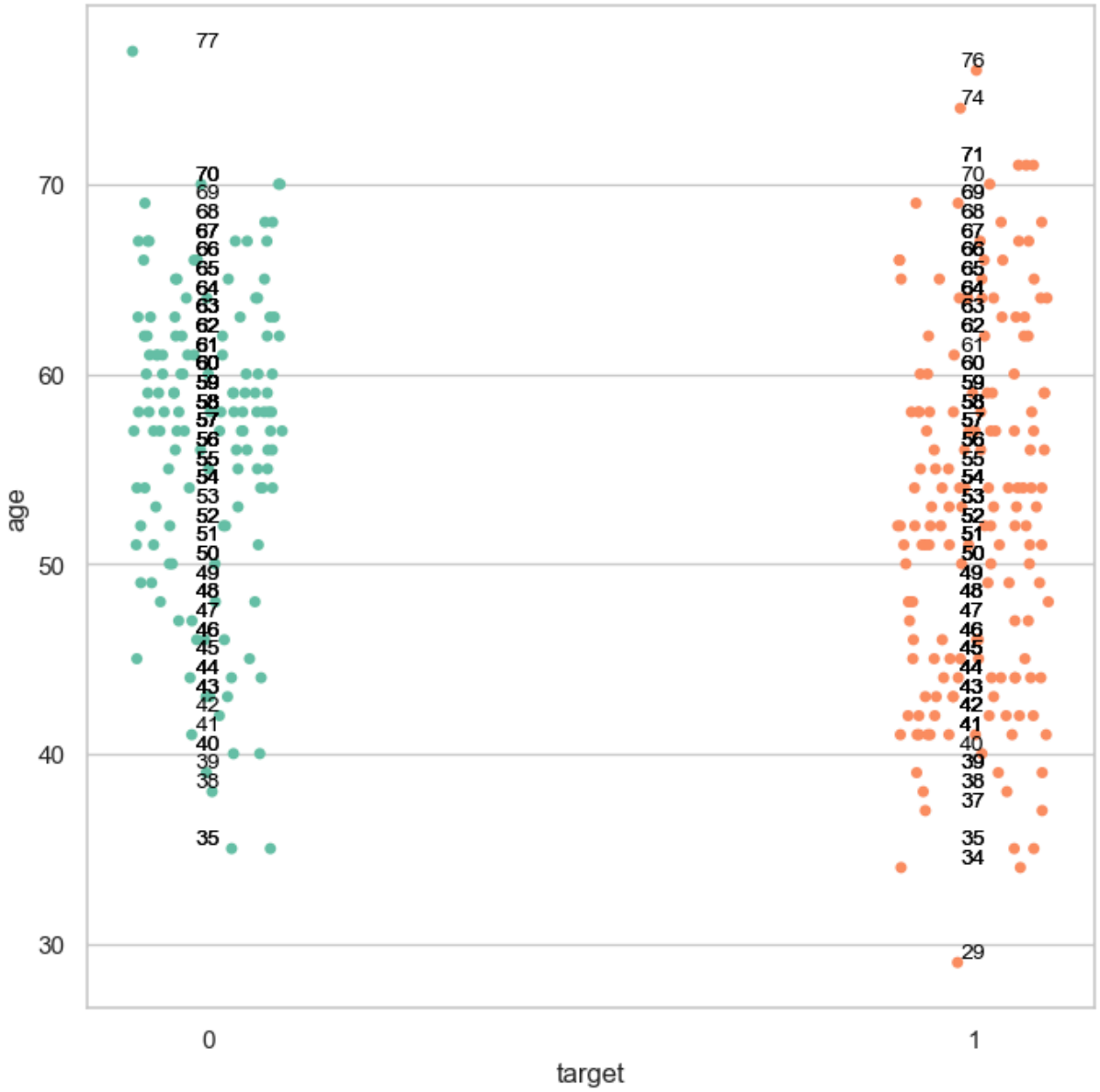
Out[136…

```
[]
```



In [137…
```
f,ax = plt.subplots(figsize=(8,8))
sns.stripplot(x='target',y='age',data= df, palette='Set2')
for i in range(len(df)):
    ax.text(df['target'].iloc[i], df['age'].iloc[i],
            str(df['age'].iloc[i]),
            color='black', ha='center', va='bottom', fontsize=10)
plt.plot()
```

Out[137…
```
[]
```

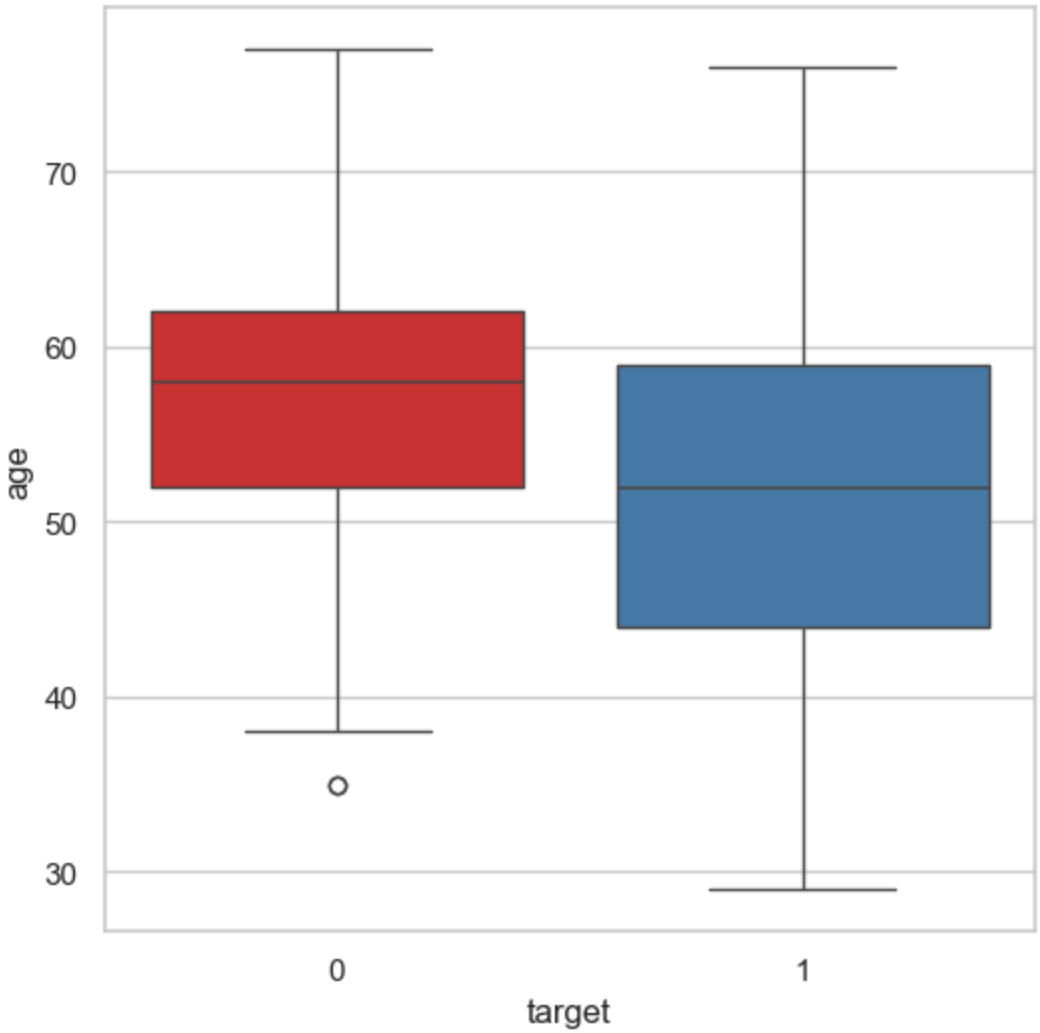


Interpretation

- We can see that the people suffering from heart disease (target = 1) and people who are not suffering from heart disease (target = 0) have comparable ages.

In [139…

```
f, ax = plt.subplots(figsize=(6,6))
sns.boxplot(x='target', y='age', data=df,palette='Set1')
plt.show()
```



Interpretation

- The above boxplot tells two different things :
  - The mean age of the people who have heart disease is less than the mean age of the people who do not have heart disease.
  - The dispersion or spread of age of the people who have heart disease is greater than the dispersion or spread of age of the people who do not have heart disease.

Analyze `age` and `trestbps` variable

trestbps : resting blood pressure (in mm Hg on admission to the hospital)

- I will plot a scatterplot to visualize the relationship between `age` and `trestbps` variable.

In [144…
```
df.groupby('age')['trestbps'].value_counts()
```

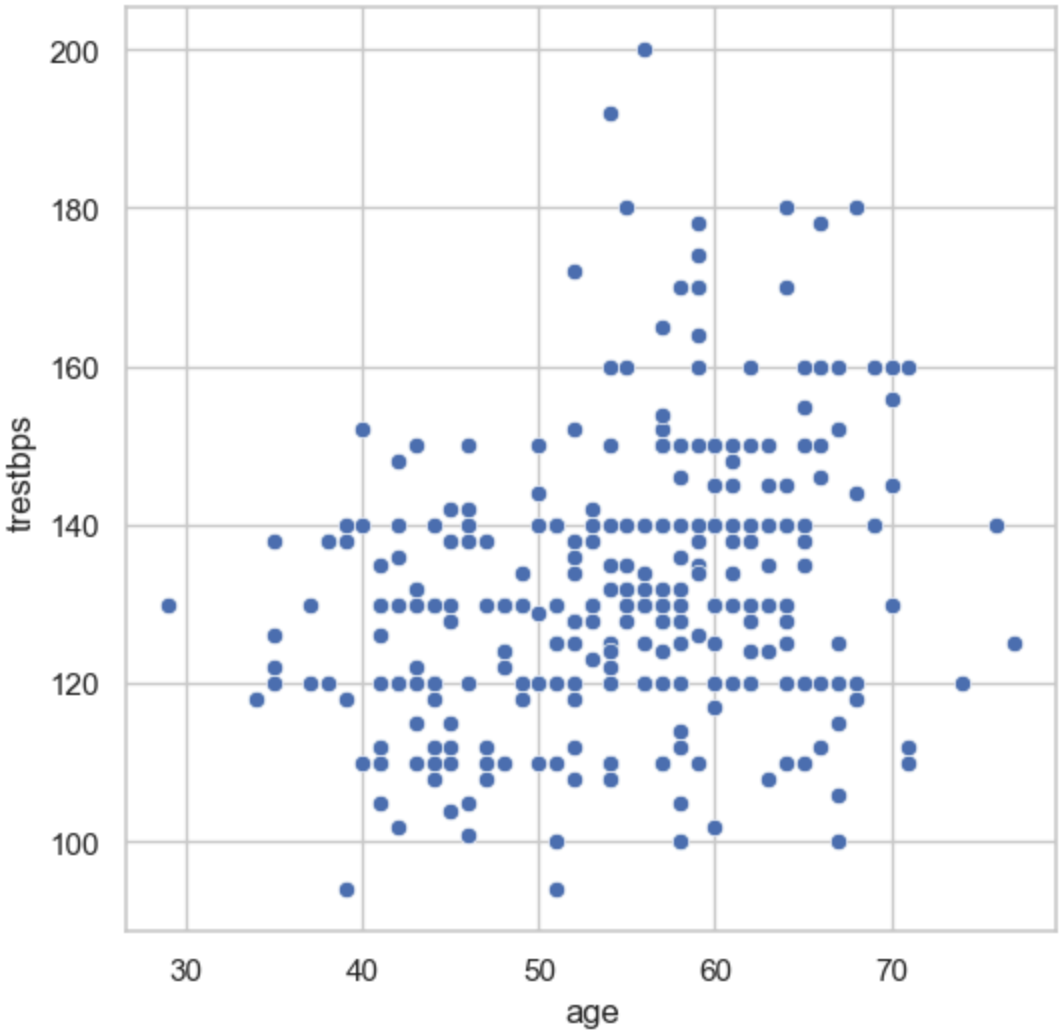
Out[144…

```
age  trestbps
29   130         1
34   118         2
35   120         1
     122         1
     126         1
                ..
71   112         1
     160         1
74   120         1
76   140         1
77   125         1
Name: count, Length: 242, dtype: int64
```

- 29 130 1: At age 29, there is 1 person with a resting blood pressure of 130 mm Hg.
- 34 118 2: At age 34, there are 2 people with a resting blood pressure of 118 mm Hg.
- 35 120 1: At age 35, there is 1 person with a resting blood pressure of 120 mm Hg.
- 35 122 1: At age 35, there is 1 person with a resting blood pressure of 122 mm Hg.

In [146…

```
f,ax = plt.subplots(figsize=(6,6))
sns.scatterplot(x='age', y='trestbps', data=df,palette='Set1')
plt.show()
```

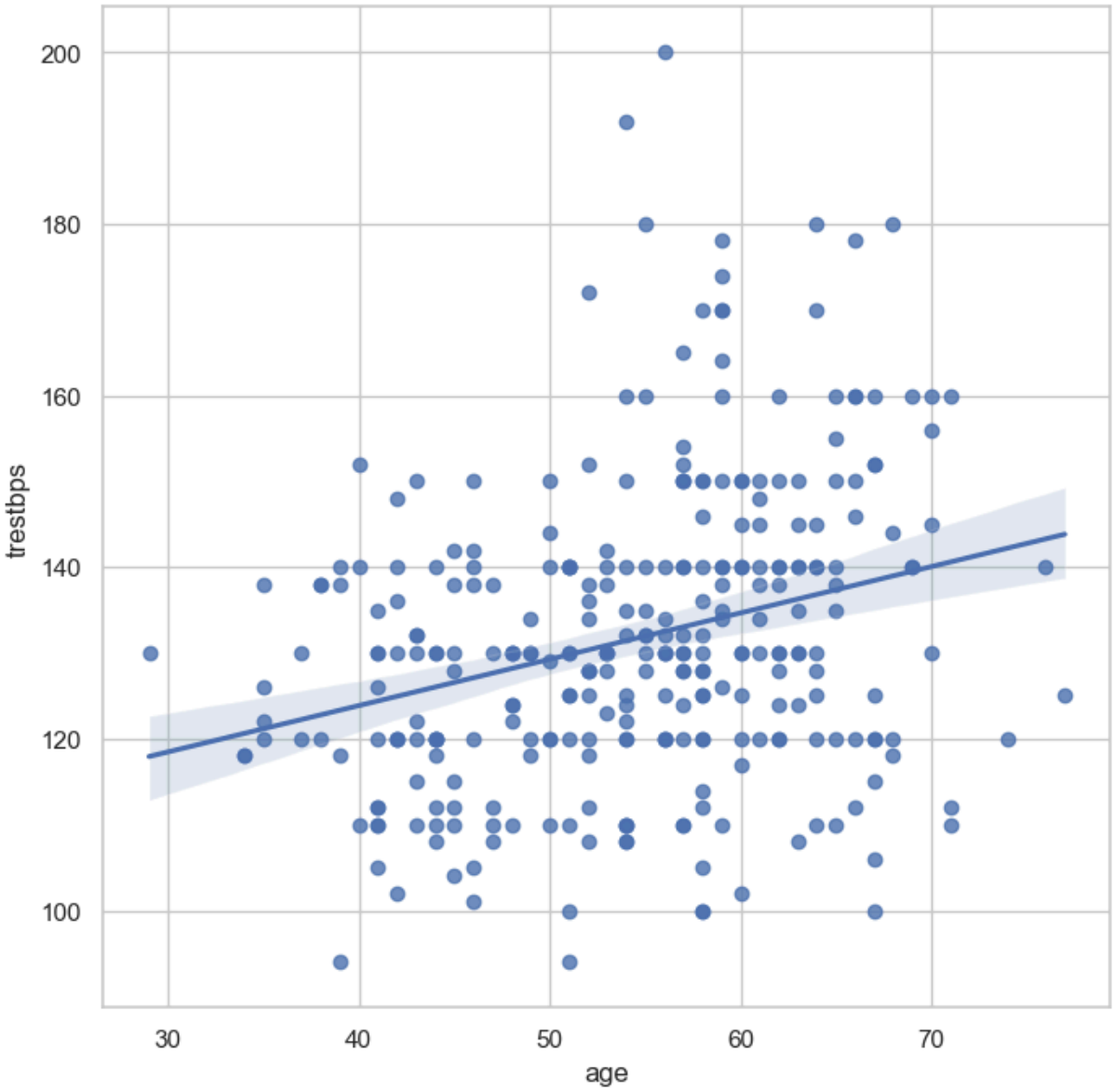


Interpretation

- The above scatter plot shows that there is no correlation between `age` and `trestbps` variable.

In [148…

```
f, ax = plt.subplots(figsize=(8,8))
sns.regplot(x='age',y='trestbps',data=df)
plt.show()
```



Interpretation

- The above line shows that linear regression model is not good fit to the data.

Analyze `age` and `chol` variable

- chol : serum cholestoral in mg/dl

In [152…

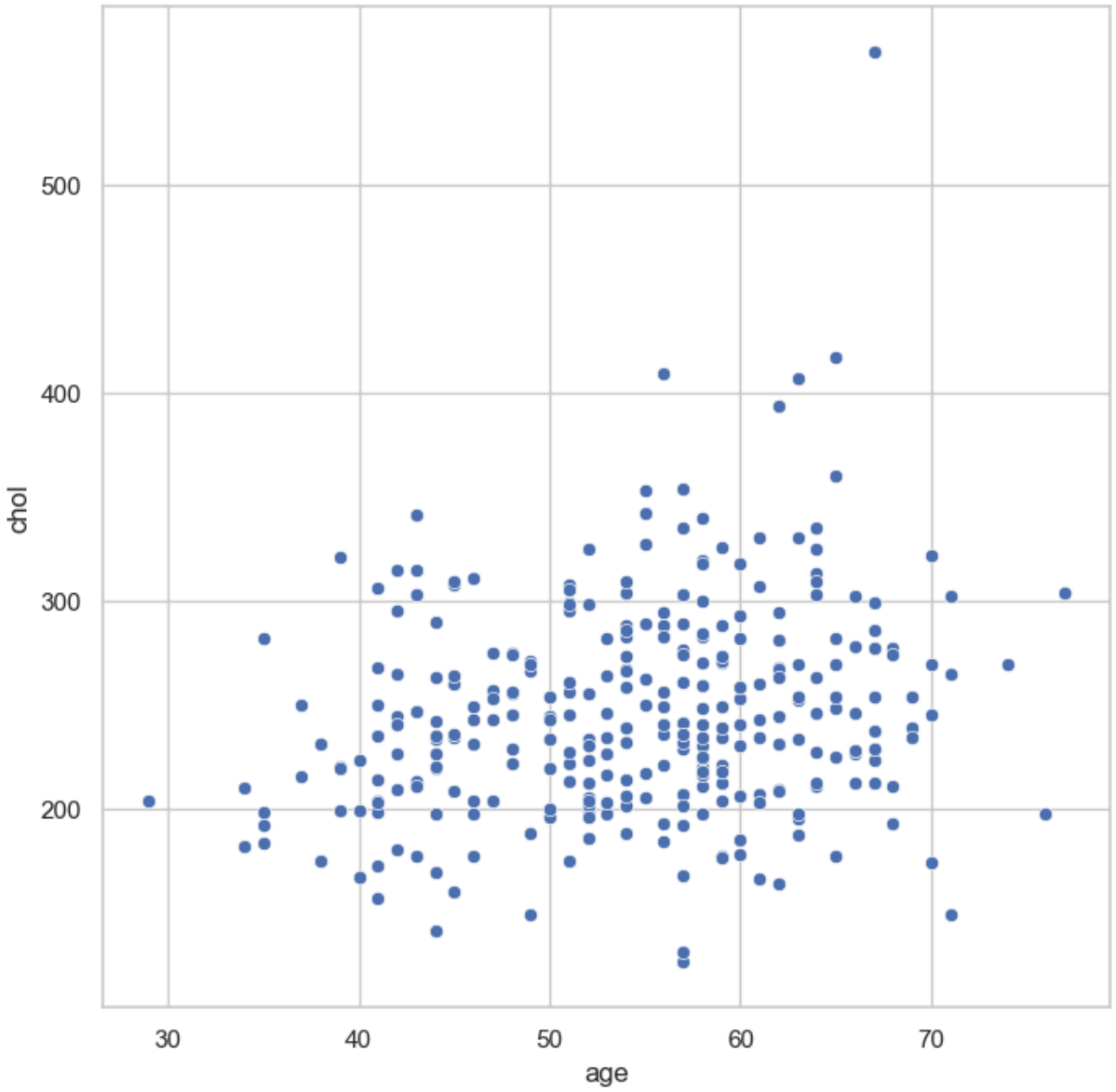
```
df.groupby('age')['chol'].value_counts()
```

Out[152…

```
age  chol
29   204     1
34   182     1
     210     1
35   183     1
     192     1
             ..
71   265     1
     302     1
74   269     1
76   197     1
77   304     1
Name: count, Length: 298, dtype: int64
```

In [153…

```
f, ax = plt.subplots(figsize=(8,8))
sns.scatterplot(x='age',y='chol', data=df)
plt.show()
```



In [154…

```
f, ax = plt.subplots(figsize=(8,8))
sns.regplot(x='age',y='chol',data=df)
```