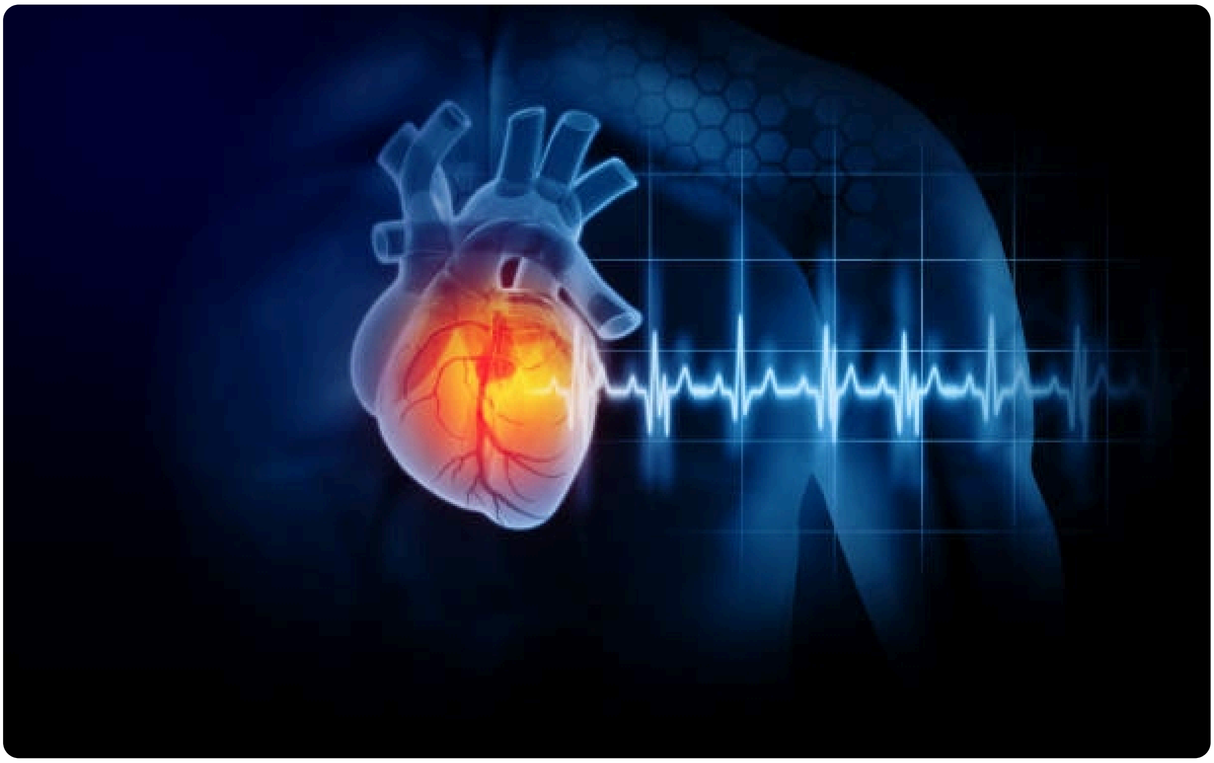


Heart Stroke Image

Extensive Analysis & Visualization with Python By SRI SAI PRAVEEN KATTA

Heart disease or **Cardiovascular disease (CVD)** is a class of diseases that involve the heart or blood vessels. Cardiovascular diseases are the leading cause of death globally. This is true in all areas of the world except Africa. Together CVD resulted in 17.9 million deaths (32.1%) in 2015. Deaths, at a given age, from CVD are more common and have been increasing in much of the developing world, while rates have declined in most of the developed world since the 1970s.

So, in this kernel, I have conducted **Exploratory Data Analysis** or **EDA** of the heart disease dataset. **Exploratory Data Analysis** or **EDA** is a critical first step in analyzing a new dataset. The primary objective of EDA is to analyze the data for distribution, outliers and anomalies in the dataset. It enable us to direct specific testing of the hypothesis. It includes analysing the data to find the distribution of data, its main characteristics, identifying patterns and visualizations. It also provides tools for hypothesis generation by visualizing and understanding the data through graphical representation.

I hope you learn and enjoy this kernel.

So, your upvote would be highly appreciated.

Table of Contents

The table of contents for this project are as follows: -

1. Introduction to EDA

Several questions come to mind when we come across a new dataset. The below list shed light on some of these questions:-

• What is the distribution of the dataset?

• Are there any missing numerical values, outliers or anomalies in the dataset?

• What are the underlying assumptions in the dataset?

• Whether there exists relationships between variables in the dataset?

• How to be sure that our dataset is ready for input in a machine learning algorithm?

• How to select the most suitable algorithm for a given dataset?

So, how do we get answer to the above questions?

2. Objectives of EDA

The objectives of the EDA are as follows:-

i. To get an overview of the distribution of the dataset.

ii. Check for missing numerical values, outliers or other anomalies in the dataset.

iii. Discover patterns and relationships between variables in the dataset.

iv. Check the underlying assumptions in the dataset.

3. Types of EDA

EDA is generally cross-classified in two ways. First, each method is either non-graphical or graphical. Second, each method is either univariate or multivariate (usually bivariate). The non-graphical methods provide insight into the characteristics and the distribution of the variable(s) of interest. So, non-graphical methods involve calculation of summary statistics while graphical methods include summarizing the data diagrammatically.

There are four types of exploratory data analysis (EDA) based on the above cross-classification methods. Each of these types of EDA are described below:-

i. Univariate non-graphical EDA

The objective of the univariate non-graphical EDA is to understand the sample distribution and also to make some initial conclusions about population distributions. Outlier detection is also a part of this analysis.

ii. Multivariate non-graphical EDA

Multivariate non-graphical EDA techniques show the relationship between two or more variables in the form of either cross-tabulation or statistics.

iii. Univariate graphical EDA

In addition to finding the various sample statistics of univariate distribution (discussed above), we also look graphically at the distribution of the sample. The non-graphical methods are quantitative and objective. They do not give full picture of the data. Hence, we need graphical methods, which are more qualitative in nature and presents an overview of the data.

iv. Multivariate graphical EDA

There are several useful multivariate graphical EDA techniques, which are used to look at the distribution of multivariate data. These are as follows:-

- Side-by-Side Boxplots
- Scatterplots
- Heat Maps and 3-D Surface Plots

4. IMPORT LIBRARIES

Libraries have been imported at the start of the script.

5. IMPORT DATASET

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	63	1	3	145	233	1	0	150	0	2.3	0	0
1	37	1	2	130	250	0	1	187	0	3.5	0	0
2	41	0	1	130	204	0	0	172	0	1.4	2	0
3	56	1	1	120	236	0	1	178	0	0.8	2	0
4	57	0	0	120	354	0	1	163	1	0.6	2	0

6. Exploratory Data Analysis

"Check shape of the dataset

- It is a good idea to first check the shape of the dataset."

```
(303, 14)
```

Now, we can see that the dataset contains 303 instances and 14 variables.

Preview the Dataset

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	63	1	3	145	233	1	0	150	0	2.3	0	0
1	37	1	2	130	250	0	1	187	0	3.5	0	0
2	41	0	1	130	204	0	0	172	0	1.4	2	0
3	56	1	1	120	236	0	1	178	0	0.8	2	0
4	57	0	0	120	354	0	1	163	1	0.6	2	0
5	57	1	0	140	192	0	1	148	0	0.4	1	0
6	56	0	1	140	294	0	0	153	0	1.3	1	0
7	44	1	1	120	263	0	1	173	0	0	2	0
8	52	1	2	172	199	1	1	162	0	0.5	2	0
9	57	1	2	150	168	0	1	174	0	1.6	2	0

Summary of the dataset

```
None
```

Dataset description

- The dataset contains several columns which are as follows -
  - age : age in years
  - sex : (1 = male; 0 = female)
  - cp : chest pain type
  - trestbps : resting blood pressure (in mm Hg on admission to the hospital)
  - chol : serum cholestoral in mg/dl
  - fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
  - restecg : resting electrocardiographic results
  - thalach : maximum heart rate achieved

- exang : exercise induced angina (1 = yes; 0 = no)
- oldpeak : ST depression induced by exercise relative to rest
- slope : the slope of the peak exercise ST segment
- ca : number of major vessels (0-3) colored by flourosopy
- thal : 3 = normal; 6 = fixed defect; 7 = reversable defect
- target : 1 or 0

Check the data types of columns

- The above `heart.info()` command gives us the number of filled values along with the data types of columns.
- If we simply want to check the data type of a particular column, we can use the following command.

	0
age	int64
sex	int64
cp	int64
trestbps	int64
chol	int64
fbs	int64
restecg	int64
thalach	int64
exang	int64
oldpeak	float64

Important points about dataset

- `sex` is a character variable. Its data type should be object. But it is encoded as (1 = male; 0 = female). So, its data type is given as int64.
- Same is the case with several other variables - `fbs` , `exang` and `target` .
- `fbs (fasting blood sugar)` should be a character variable as it contains only 0 and 1 as values (1 = true; 0 = false). As it contains only 0 and 1 as values, so its data type is given as int64.
- `exang (exercise induced angina)` should also be a character variable as it contains only 0 and 1 as values (1 = yes; 0 = no). It also contains only 0 and 1 as values, so its data type is given as int64.

- `target` should also be a character variable. But, it also contains 0 and 1 as values. So, its data type is given as int64.

Statistical properties of dataset

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
count	303	303	303	303	303	303	303	303	303	303	303
mean	54.3663	0.6832	0.967	131.6238	246.264	0.1485	0.5281	149.6469	0.3267	1.0396	1.3993
std	9.0821	0.466	1.0321	17.5381	51.8308	0.3562	0.5259	22.9052	0.4698	1.1611	0.6162
min	29	0	0	94	126	0	0	71	0	0	0
25%	47.5	0	0	120	211	0	0	133.5	0	0	1
50%	55	1	1	130	240	0	1	153	0	0.8	1
75%	61	1	2	140	274.5	0	1	166	1	1.6	2
max	77	1	3	200	564	1	2	202	1	6.2	2

Important points to note

- The above command `heart.describe()` helps us to view the statistical properties of numerical variables. It excludes character variables.
- If we want to view the statistical properties of character variables, we should run the following command -

  `heart.describe(include=['object'])`

- If we want to view the statistical properties of all the variables, we should run the following command -

  `heart.describe(include='all')`

View Column Names

0
age
sex
cp
trestbps
chol
fbs
restecg
thalach
exang
oldpeak

7. Univariate analysis

Analysis of `target` feature variable

- Our feature variable of interest is `target`.
- It refers to the presence of heart disease in the patient.
- It is integer valued as it contains two integers 0 and 1 - (0 stands for absence of heart disease and 1 for presence of heart disease).
- So, in this section, I will analyze the `target` variable.

Check the number of unique values in `target` variable

`2`

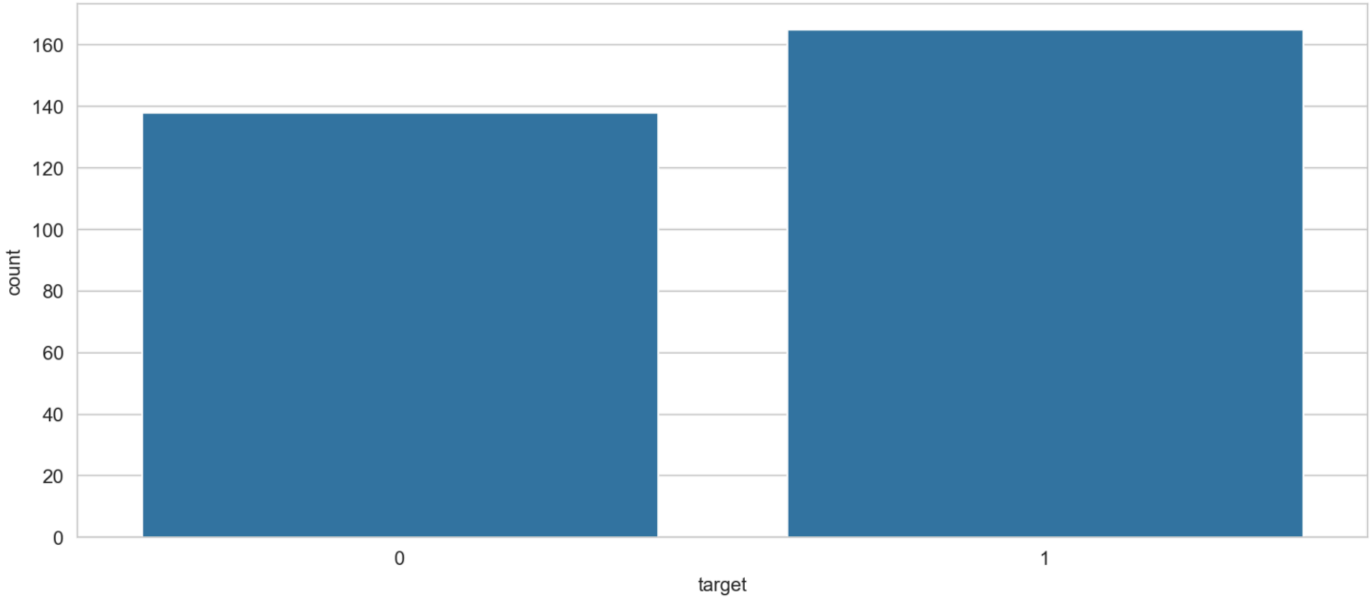
value
1
0

Frequency distribution of `target` variable

target	count
1	165
0	138

- `1` stands for presence of heart disease. So, there are 165 patients suffering from heart disease.
- Similarly, `0` stands for absence of heart disease. So, there are 138 patients who do not have any heart disease.
- We can visualize this information below.

Visualize frequency distribution of `target` variable



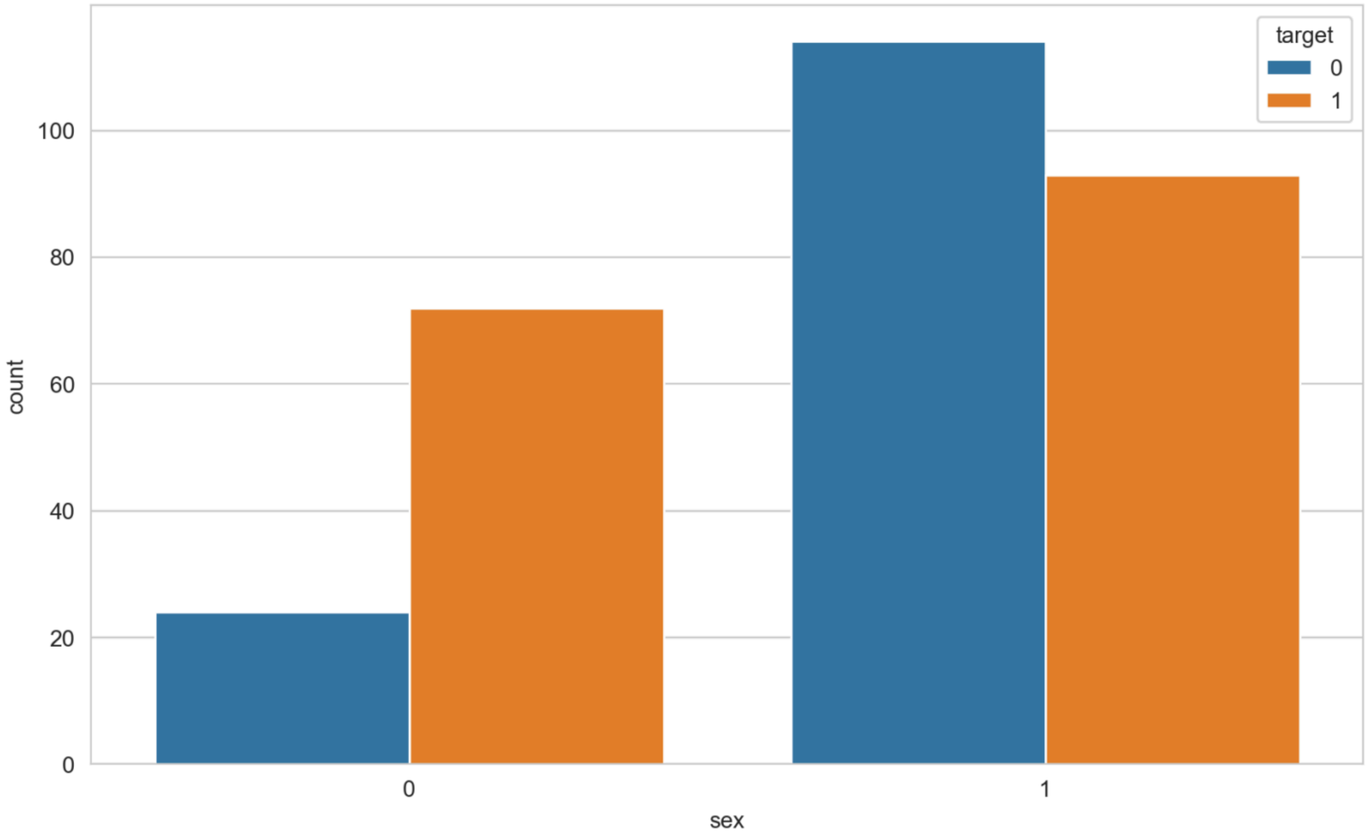
Interpretation

- The above plot confirms the findings that -
    - There are 165 patients suffering from heart disease, and
    - There are 138 patients who do not have any heart disease.

Frequency distribution of `target` variable wrt `sex`

sex	target	count
0	1	72
0	0	24
1	0	114
1	1	93

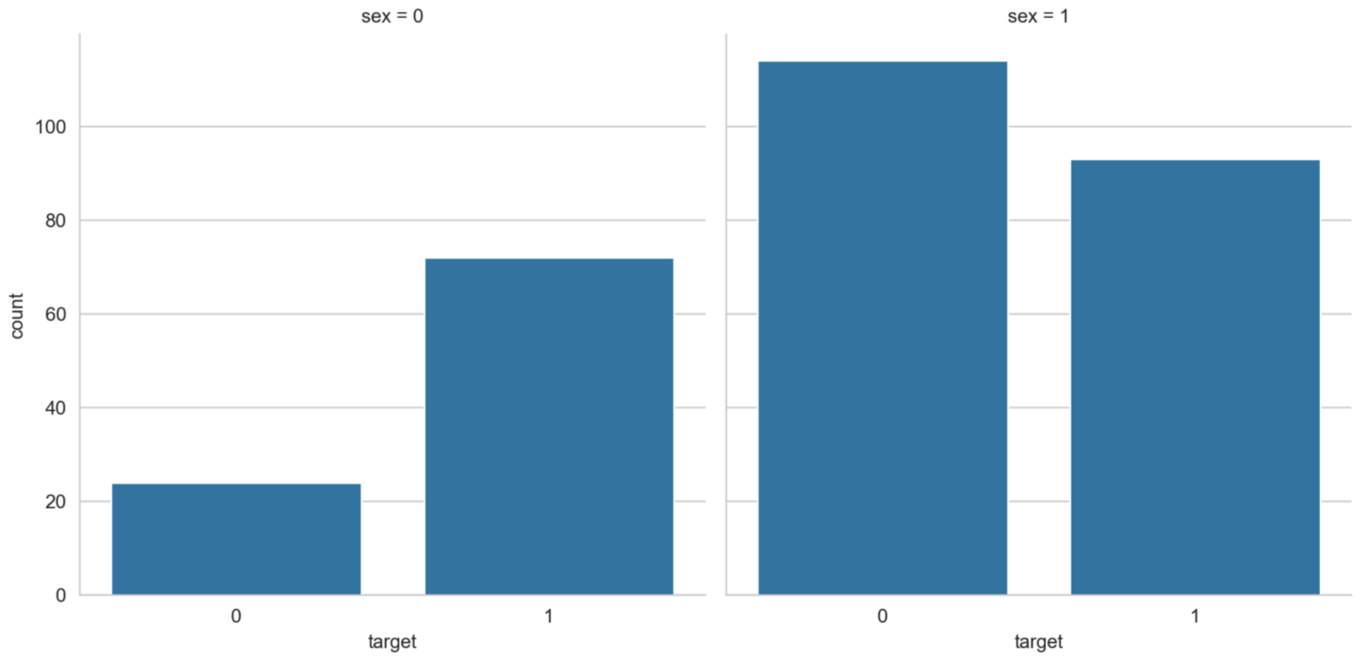
- `sex` variable contains two integer values 1 and 0 : (1 = male; 0 = female).
- `target` variable also contains two integer values 1 and 0 : (1 = Presence of heart disease; 0 = Absence of heart disease)
- So, out of 96 females - 72 have heart disease and 24 do not have heart disease.
- Similarly, out of 207 males - 93 have heart disease and 114 do not have heart disease.
- We can visualize this information below.



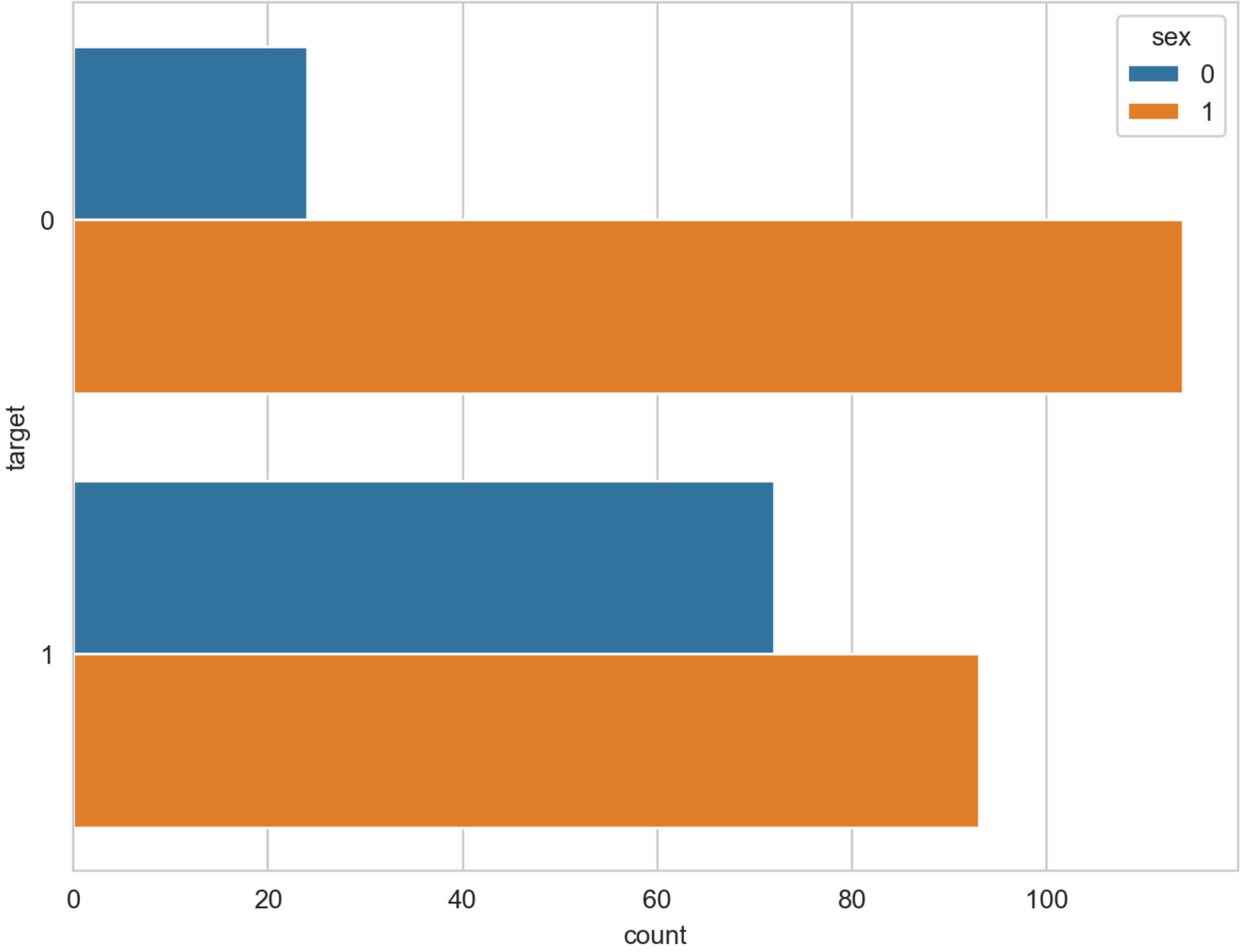
Interpretation

- We can see that the values of `target` variable are plotted wrt `sex` : (1 = male; 0 = female).
- `target` variable also contains two integer values 1 and 0 : (1 = Presence of heart disease; 0 = Absence of heart disease)

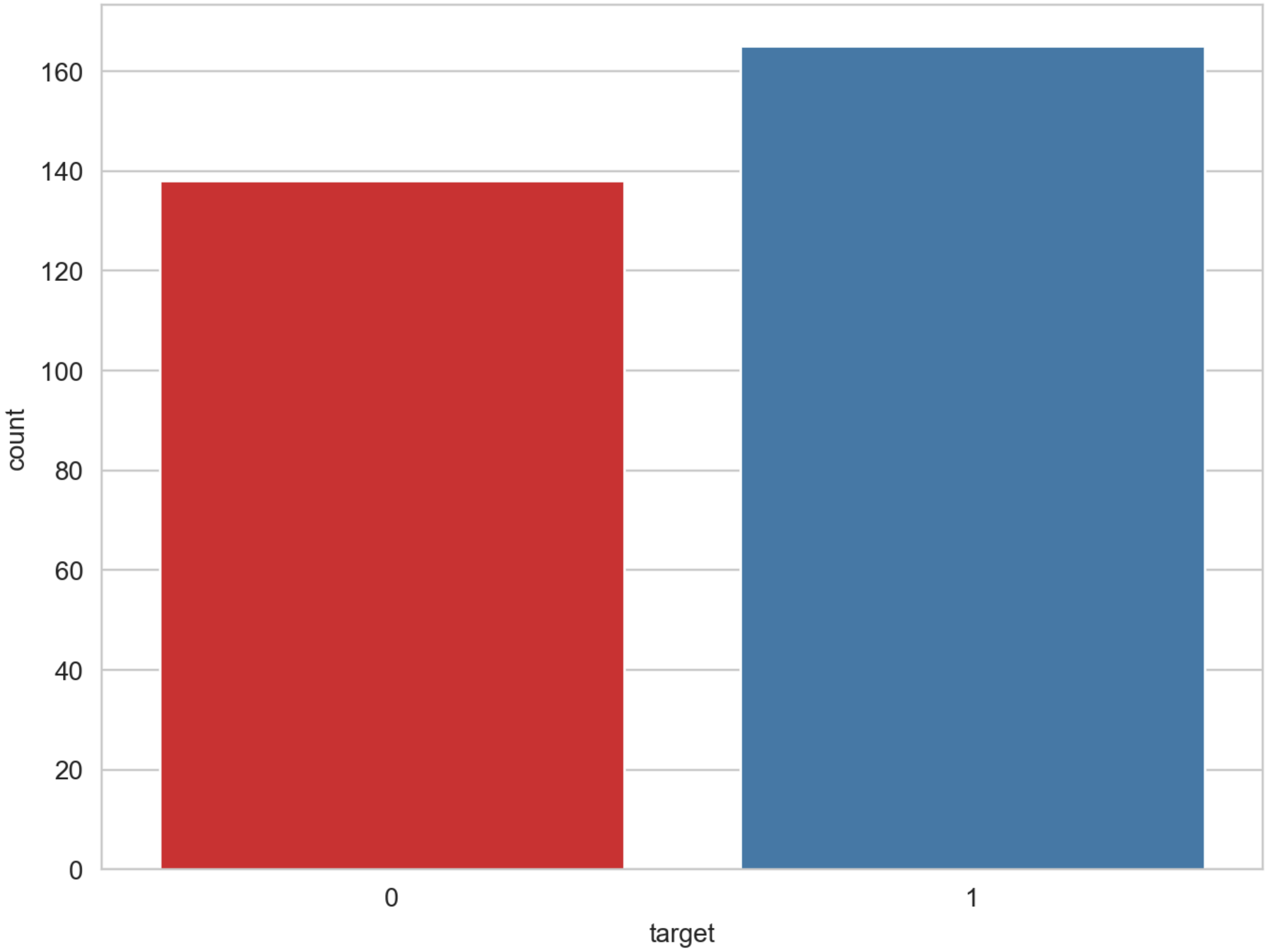
- The above plot confirms our findings that -
    - Out of 96 females - 72 have heart disease and 24 do not have heart disease.
    - Similarly, out of 207 males - 93 have heart disease and 114 do not have heart disease.

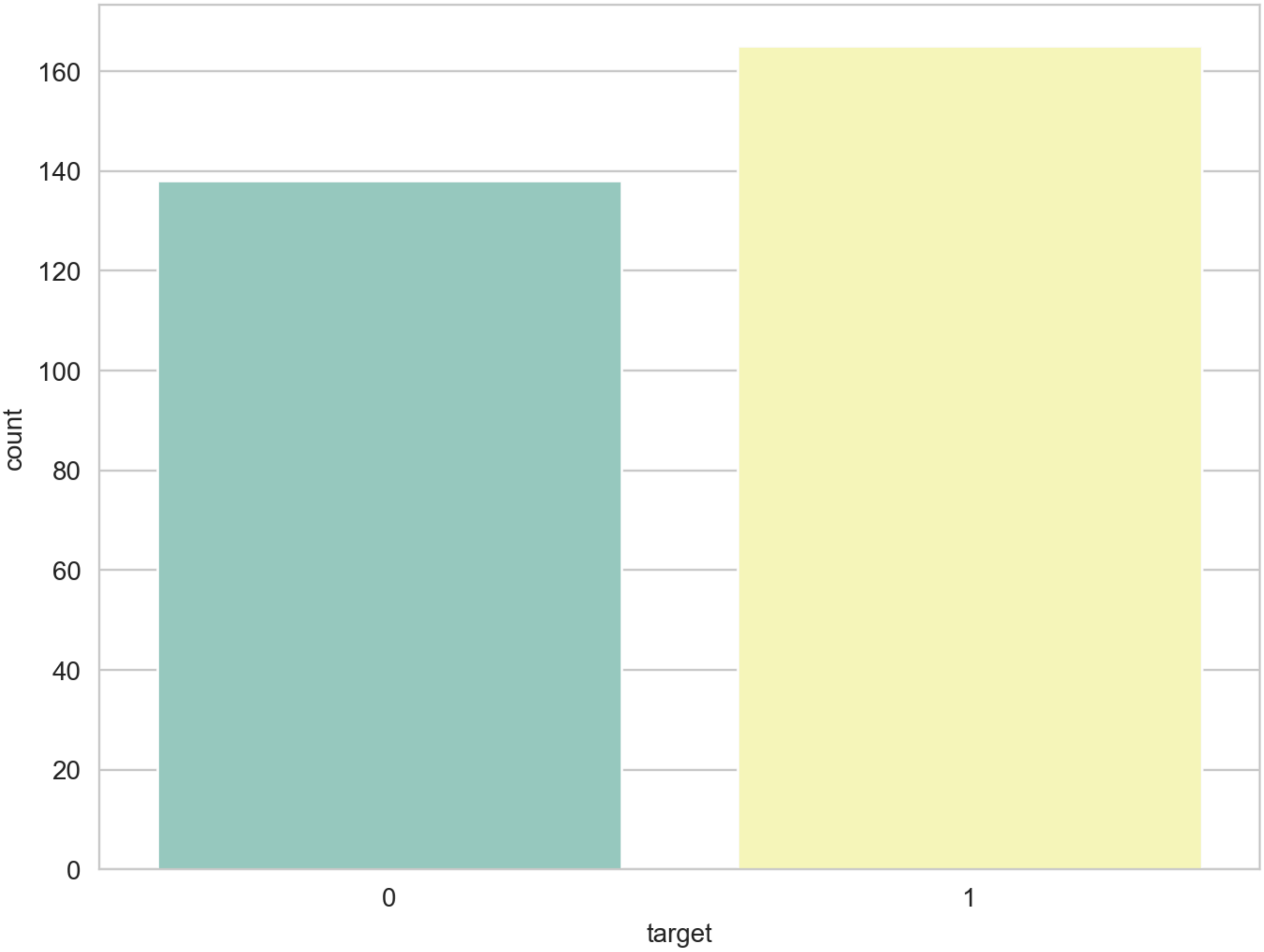


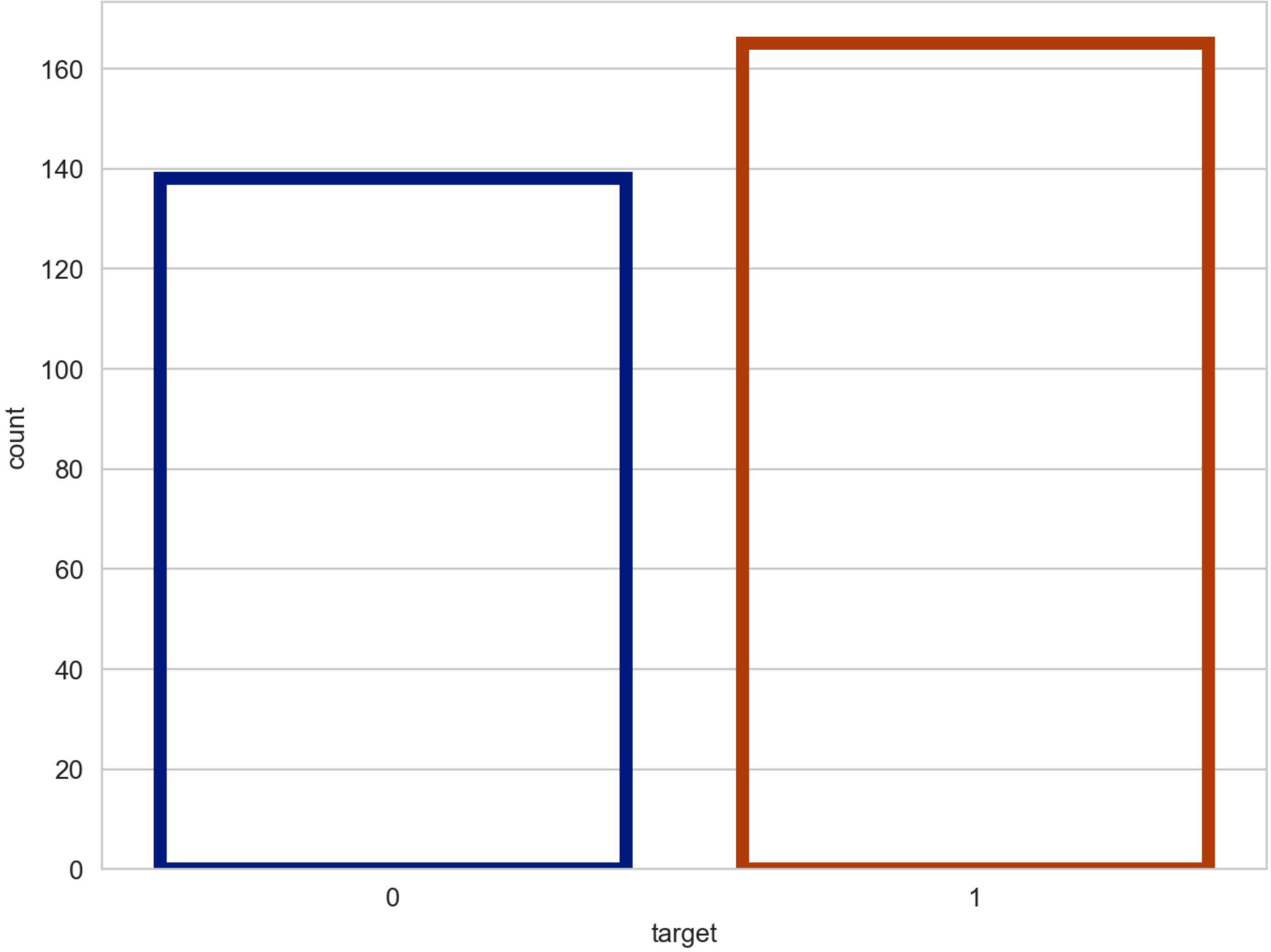
- The above plot segregate the values of `target` variable and plot on two different columns labelled as (sex = 0, sex = 1).
- I think it is more convinient way of interpret the plots. We can plot the bars horizontally as follows :



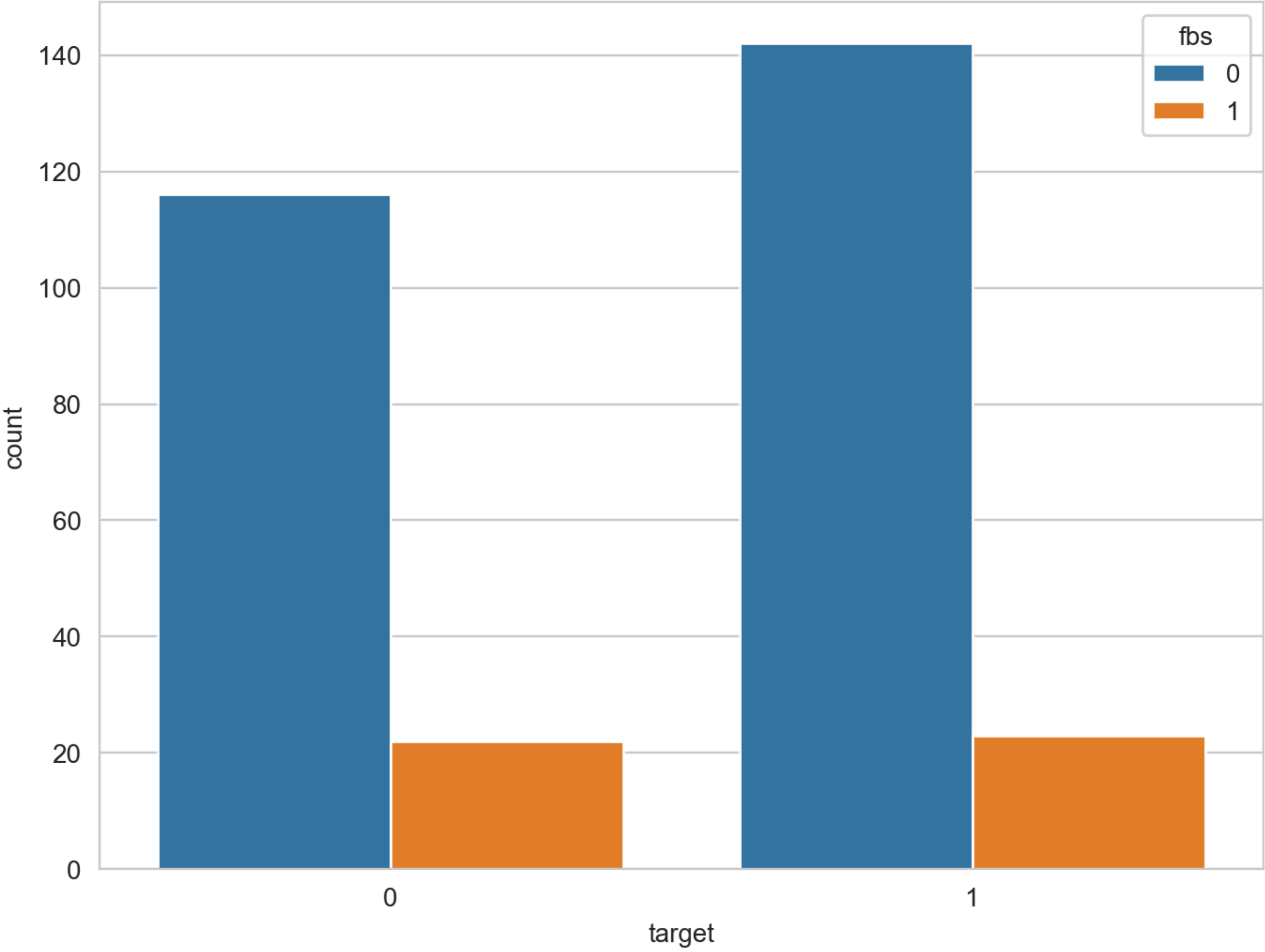
We can use a different color palette as follows :

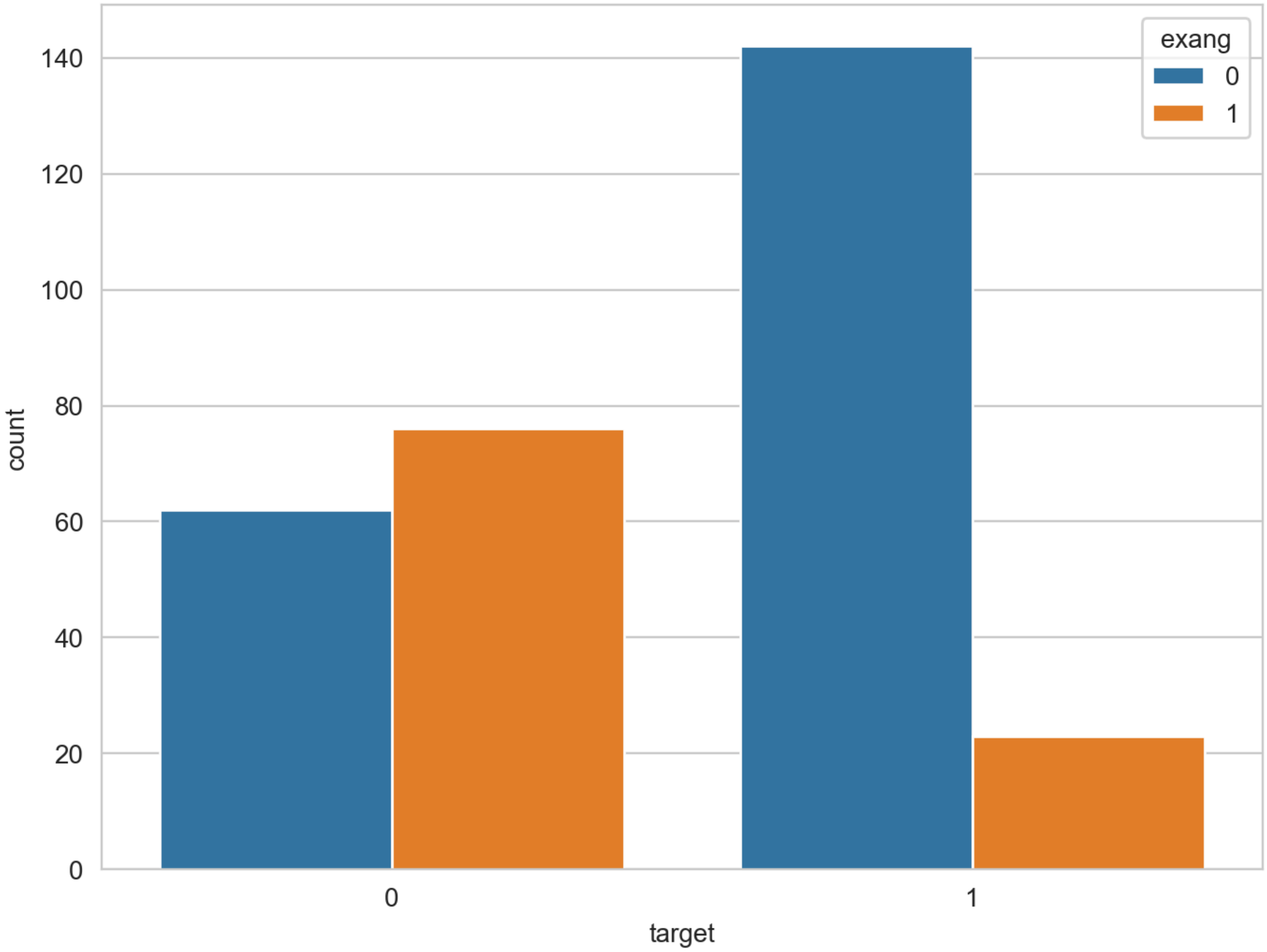

160
140
120
100
80
60
40
20
0
count
0
1
target


160
140
120
100
80
60
40
20
0
count
0
1
target



- I have visualize the `target` values distribution wrt `sex`.
- We can follow the same principles and visualize the `target` values distribution wrt `fbs (fasting blood sugar)` and `exang (exercise induced angina)`.


fbs
0
1
count
140
120
100
80
60
40
20
0
0
1
target



Findings of Univariate Analysis

Findings of univariate analysis are as follows:-

- Our feature variable of interest is `target`.
- It refers to the presence of heart disease in the patient.
- It is integer valued as it contains two integers 0 and 1 - (0 stands for absence of heart disease and 1 for presence of heart disease).
- `1` stands for presence of heart disease. So, there are 165 patients suffering from heart disease.
- Similarly, `0` stands for absence of heart disease. So, there are 138 patients who do not have any heart disease.
- There are 165 patients suffering from heart disease, and
- There are 138 patients who do not have any heart disease.
- Out of 96 females - 72 have heart disease and 24 do not have heart disease.
- Similarly, out of 207 males - 93 have heart disease and 114 do not have heart disease.

8. Bivariate Analysis

Estimate correlation coefficients

Our dataset is very small. So, I will compute the standard correlation coefficient (also called Pearson's r) between every pair of attributes. I will compute it using the `heart.corr()` method as follows:-

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slop
age	1	-0.0984	-0.0687	0.2794	0.2137	0.1213	-0.1162	-0.3985	0.0968	0.21	-0.16
sex	-0.0984	1	-0.0494	-0.0568	-0.1979	0.045	-0.0582	-0.044	0.1417	0.0961	-0.03
cp	-0.0687	-0.0494	1	0.0476	-0.0769	0.0944	0.0444	0.2958	-0.3943	-0.1492	0.11
trestbps	0.2794	-0.0568	0.0476	1	0.1232	0.1775	-0.1141	-0.0467	0.0676	0.1932	-0.12
chol	0.2137	-0.1979	-0.0769	0.1232	1	0.0133	-0.151	-0.0099	0.067	0.054	-0.0
fbs	0.1213	0.045	0.0944	0.1775	0.0133	1	-0.0842	-0.0086	0.0257	0.0057	-0.05
restecg	-0.1162	-0.0582	0.0444	-0.1141	-0.151	-0.0842	1	0.0441	-0.0707	-0.0588	0.0
thalach	-0.3985	-0.044	0.2958	-0.0467	-0.0099	-0.0086	0.0441	1	-0.3788	-0.3442	0.38
exang	0.0968	0.1417	-0.3943	0.0676	0.067	0.0257	-0.0707	-0.3788	1	0.2882	-0.25
oldpeak	0.21	0.0961	-0.1492	0.1932	0.054	0.0057	-0.0588	-0.3442	0.2882	1	-0.57

The target variable is `target`. So, we should check how each attribute correlates with the `target` variable.

	target
target	1
cp	0.4338
thalach	0.4217
slope	0.3459
restecg	0.1372
fbs	-0.028
chol	-0.0852
trestbps	-0.1449
age	-0.2254
sex	-0.2809

Interpretation of correlation coefficient

- The correlation coefficient ranges from -1 to +1.
- When it is close to +1, this signifies that there is a strong positive correlation. So, we can see that there is no variable which has strong positive correlation with `target` variable.
- When it is close to -1, it means that there is a strong negative correlation. So, we can see that there is no variable which has strong negative correlation with `target` variable.
- When it is close to 0, it means that there is no correlation. So, there is no correlation between `target` and `fbs` .
- We can see that the `cp` and `thalach` variables are mildly positively correlated with `target` variable. So, I will analyze the interaction between these features and `target` variable.

Analysis of `target` and `cp` variable

Explore `cp` variable

- `cp` stands for chest pain type.
- First, I will check number of unique values in `cp` variable.

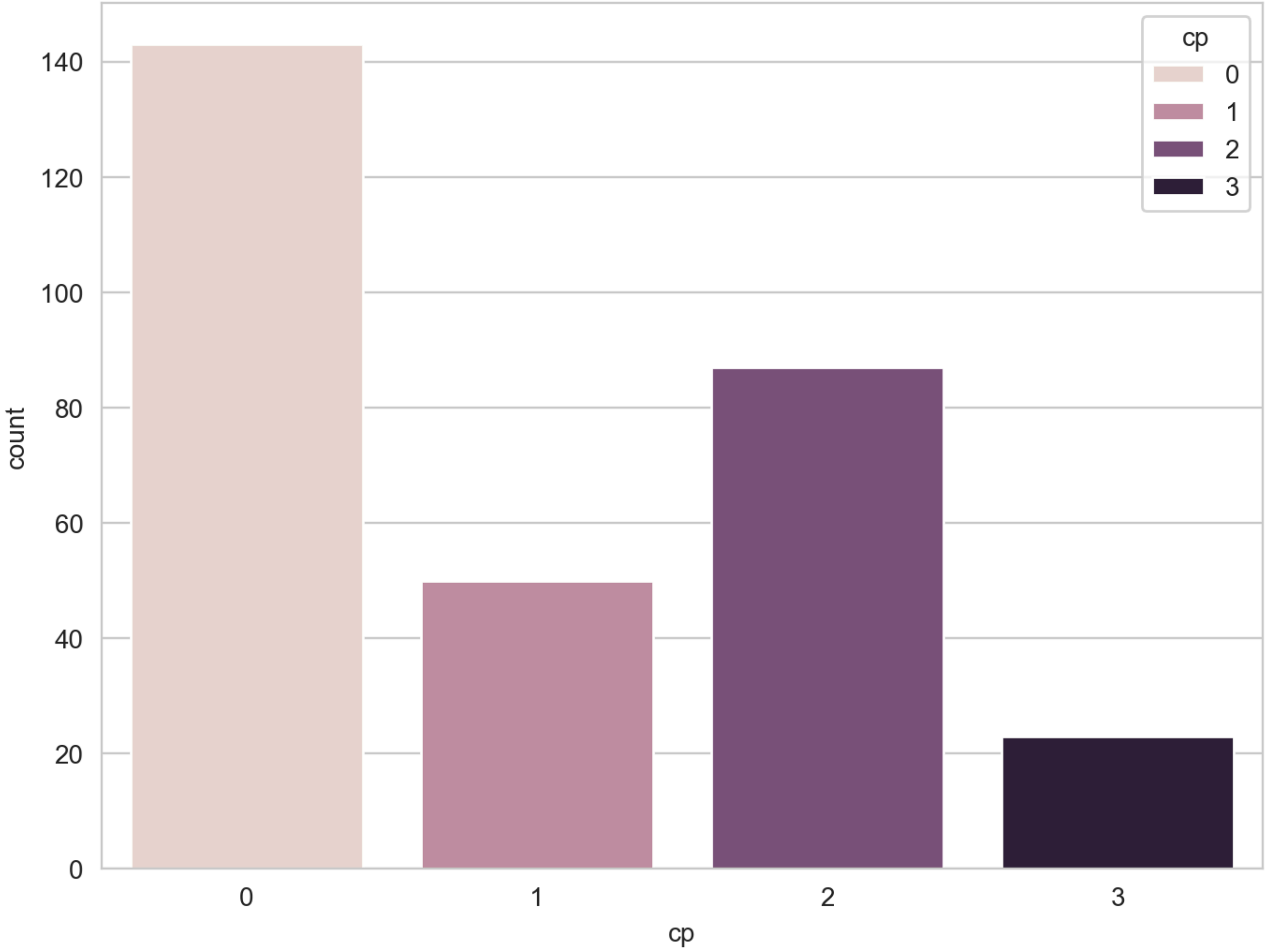
```
4
```

So, there are 4 unique values in `cp` variable. Hence, it is a categorical variable.

cp	count
0	143
2	87
1	50
3	23

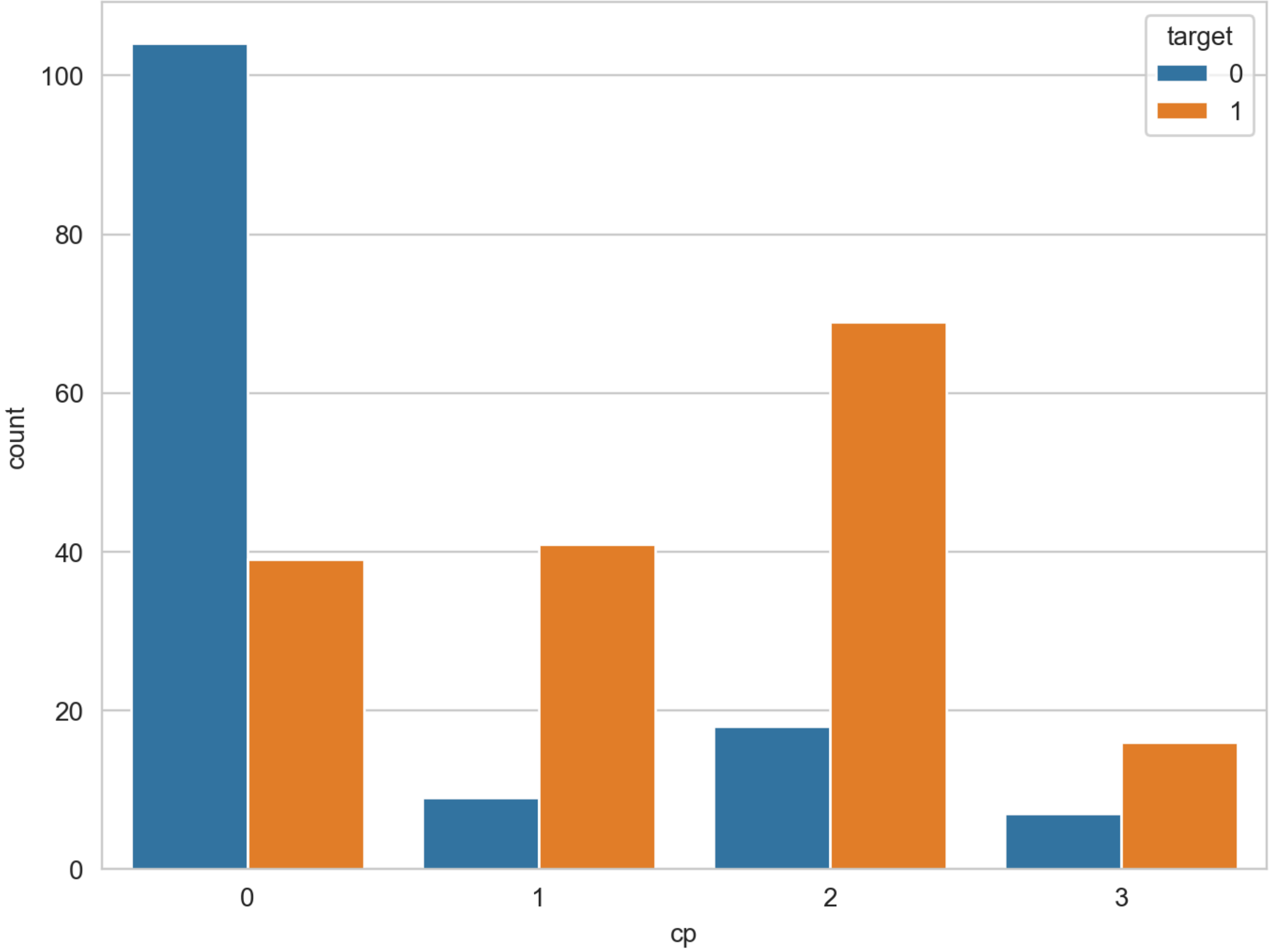
- It can be seen that `cp` is a categorical variable and it contains 4 types of values - 0, 1, 2 and 3.

Visualize the frequency distribution of `cp` variable

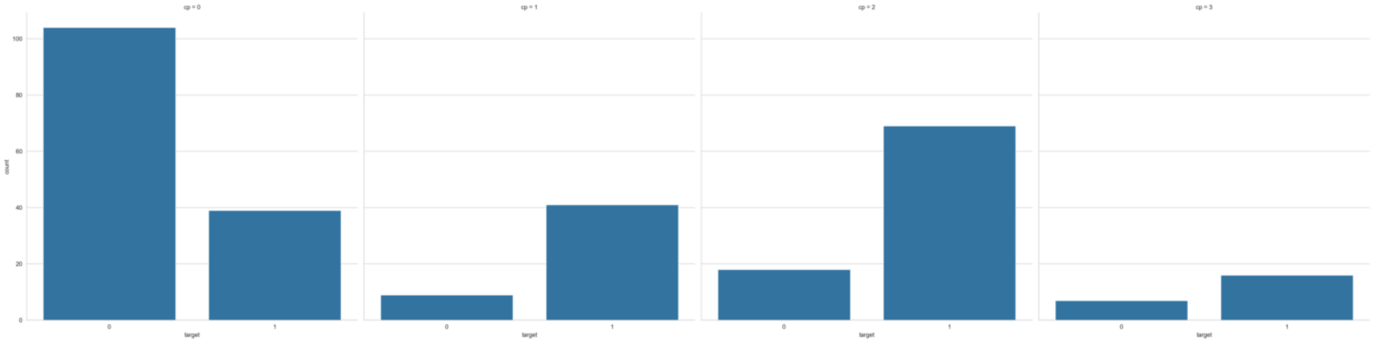


cp	target	count
0	0	104
0	1	39
1	1	41
1	0	9
2	1	69
2	0	18
3	1	16
3	0	7

- `cp` variable contains four integer values 0, 1, 2 and 3.
- `target` variable contains two integer values 1 and 0 : (1 = Presence of heart disease; 0 = Absence of heart disease)
- So, the above analysis gives `target` variable values categorized into presence and absence of heart disease and groupby `cp` variable values.
- We can visualize this information below.



- We can see that the values of `target` variable are plotted wrt `cp` .
- `target` variable contains two integer values 1 and 0 : (1 = Presence of heart disease; 0 = Absence of heart disease)
- The above plot confirms our above findings,



Analysis of `target` and `thalach` variable

Explore `thalach` variable

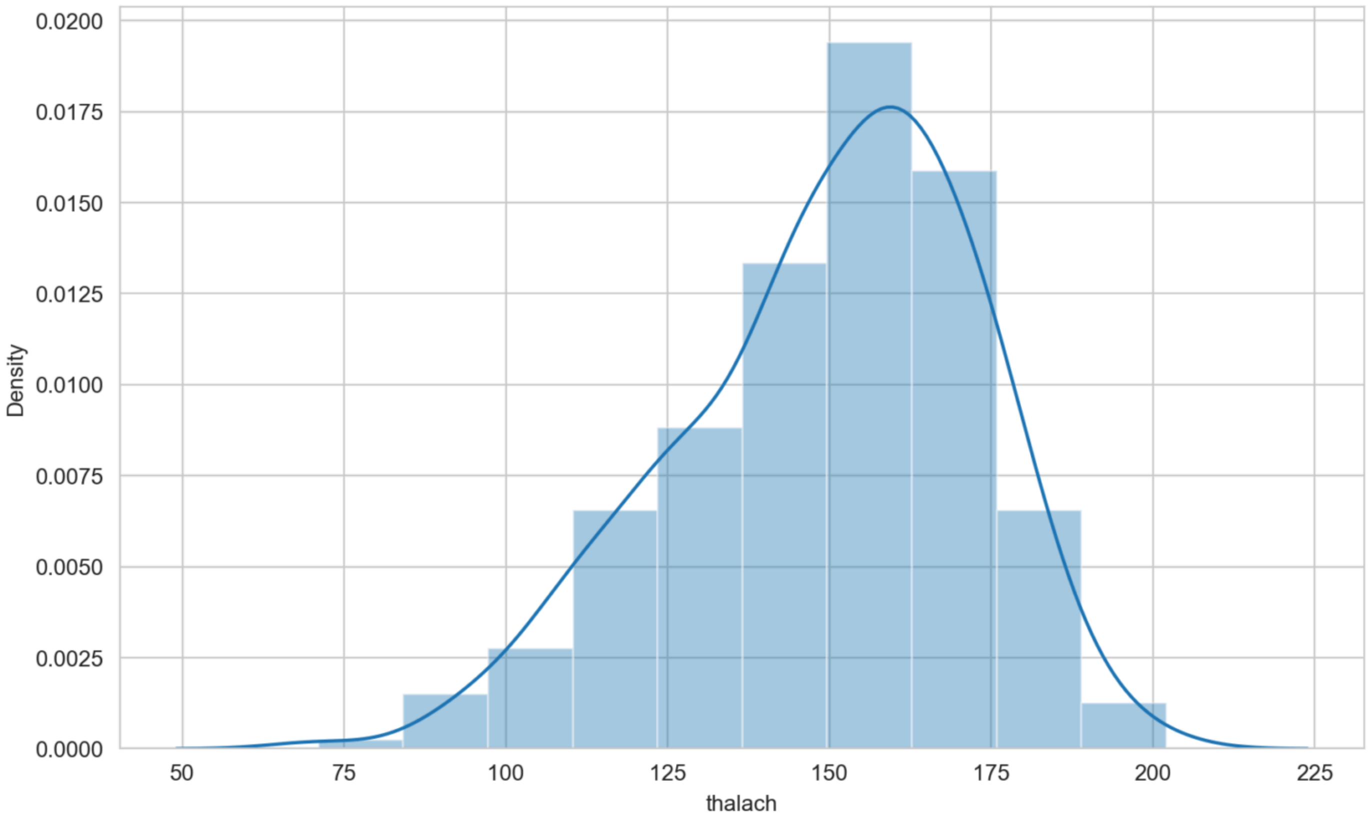
- `thalach` stands for maximum heart rate achieved.

- I will check number of unique values in `thalach` variable

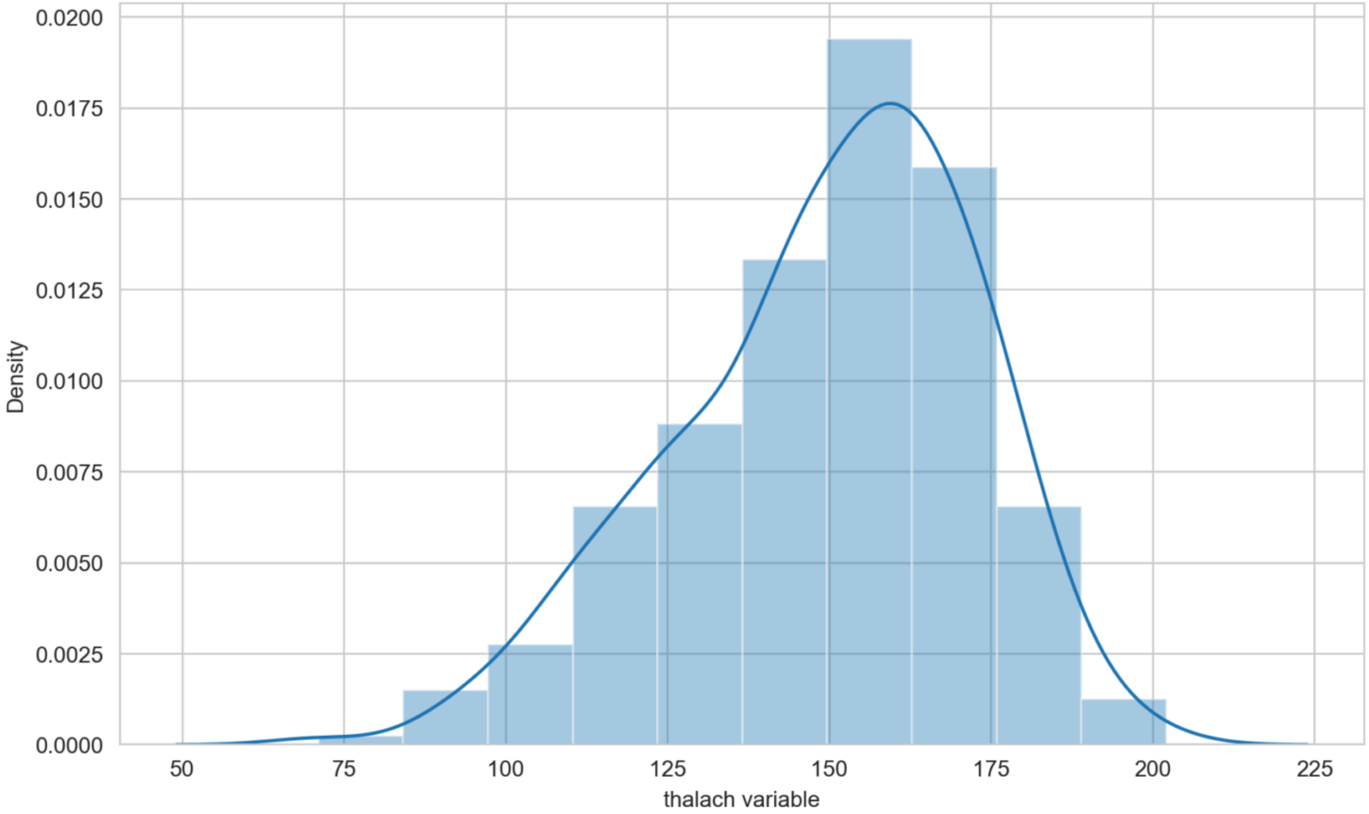
```
91
```

- So, number of unique values in `thalach` variable is 91. Hence, it is numerical variable.

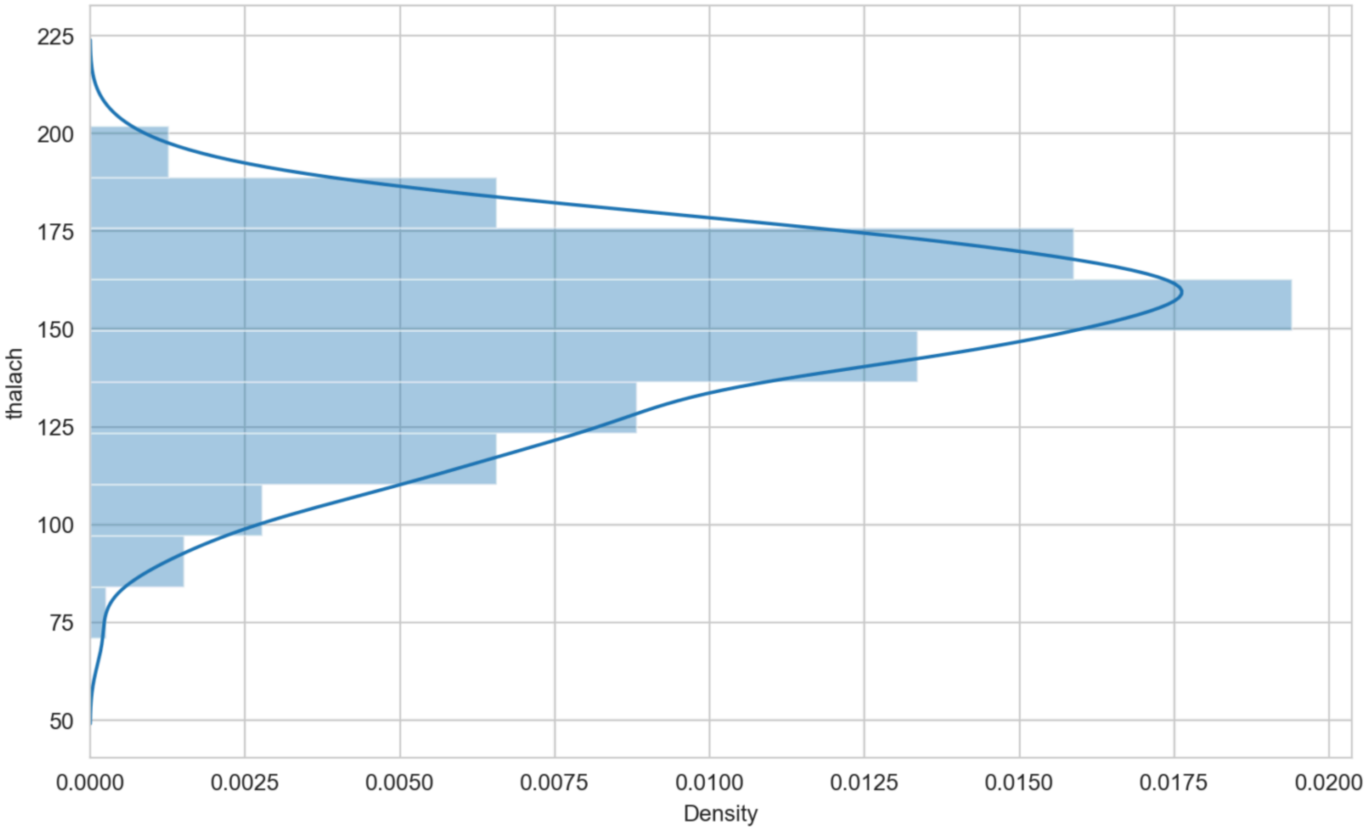
value
150
187
172
178
163
148
153
173
162
174



We can use Pandas series object to get an informative axis label :



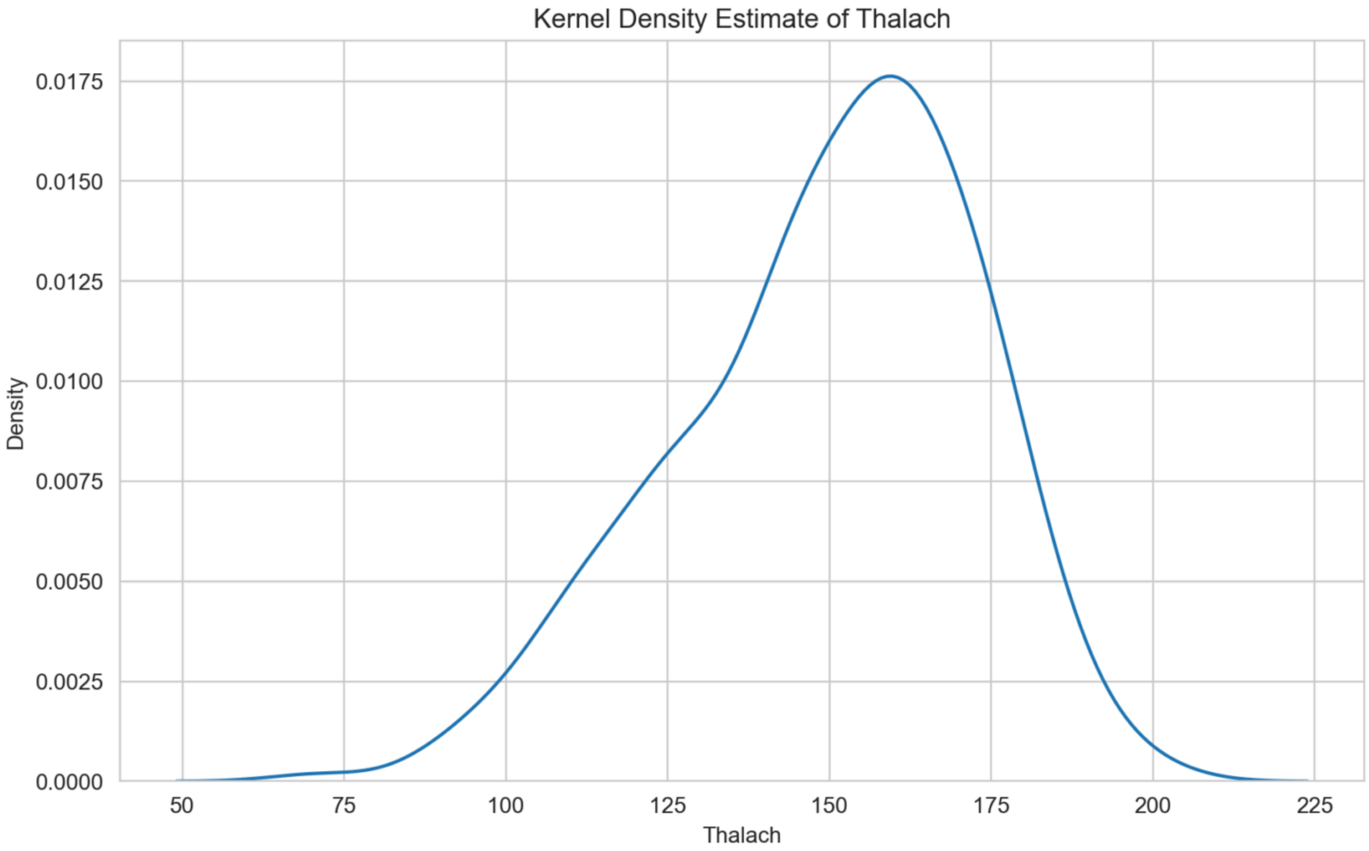
We can plot the distribution on the vertical



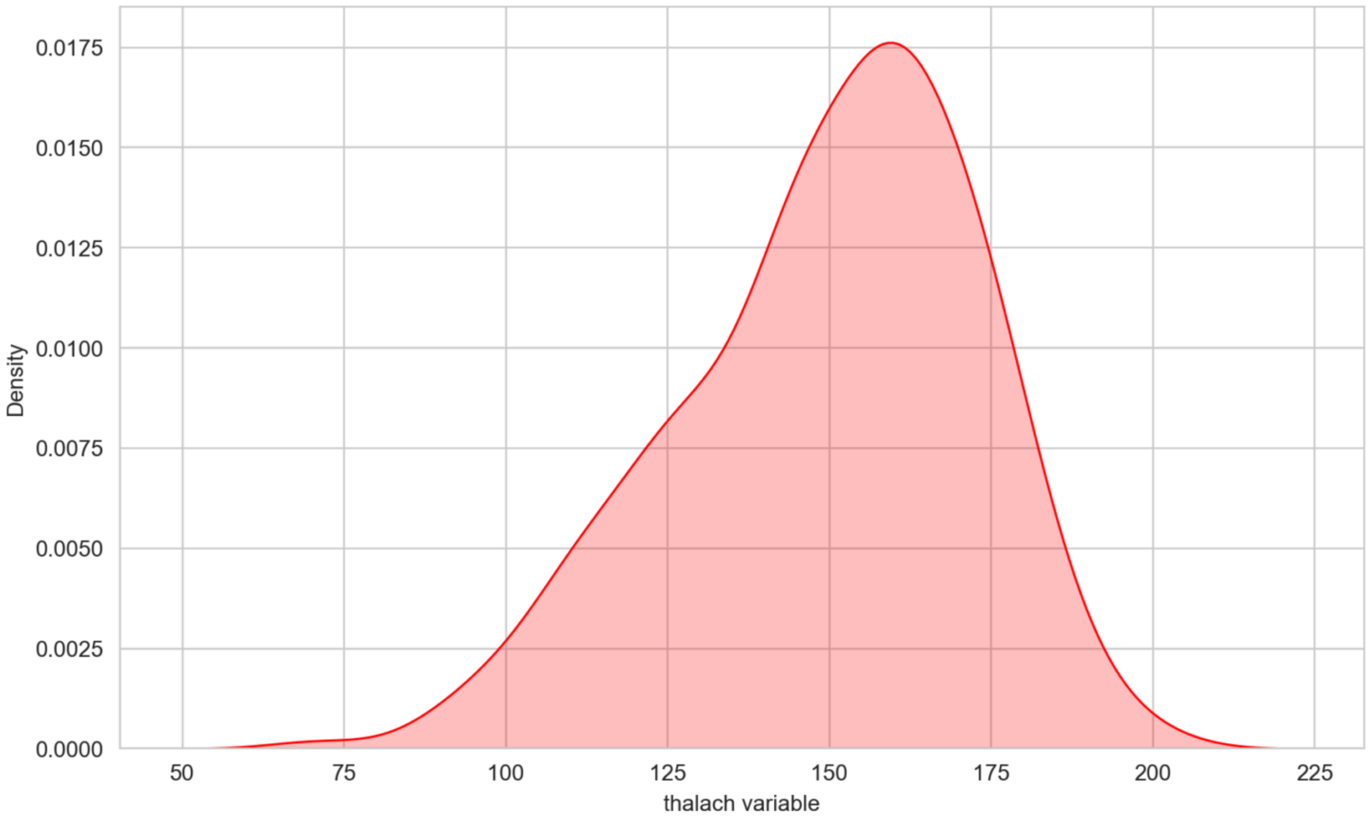
Seaborn Kernel Density Estimation (KDE) Plot

- The kernel density estimate (KDE) plot is a useful tool for plotting the shape of a distribution.
- The KDE plot plots the density of observations on one axis with height along the other axis.

- We can plot a KDE plot as follows :

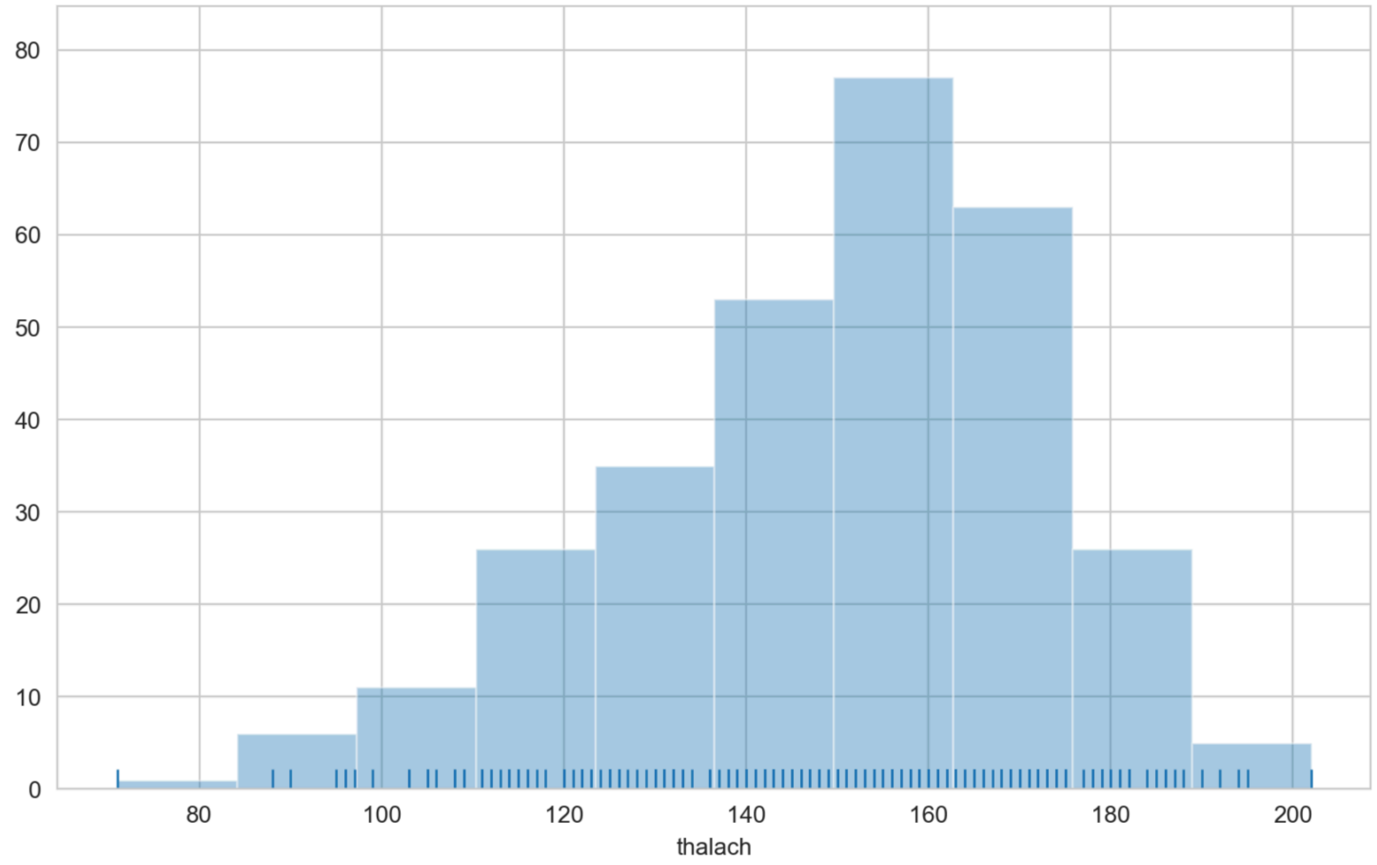


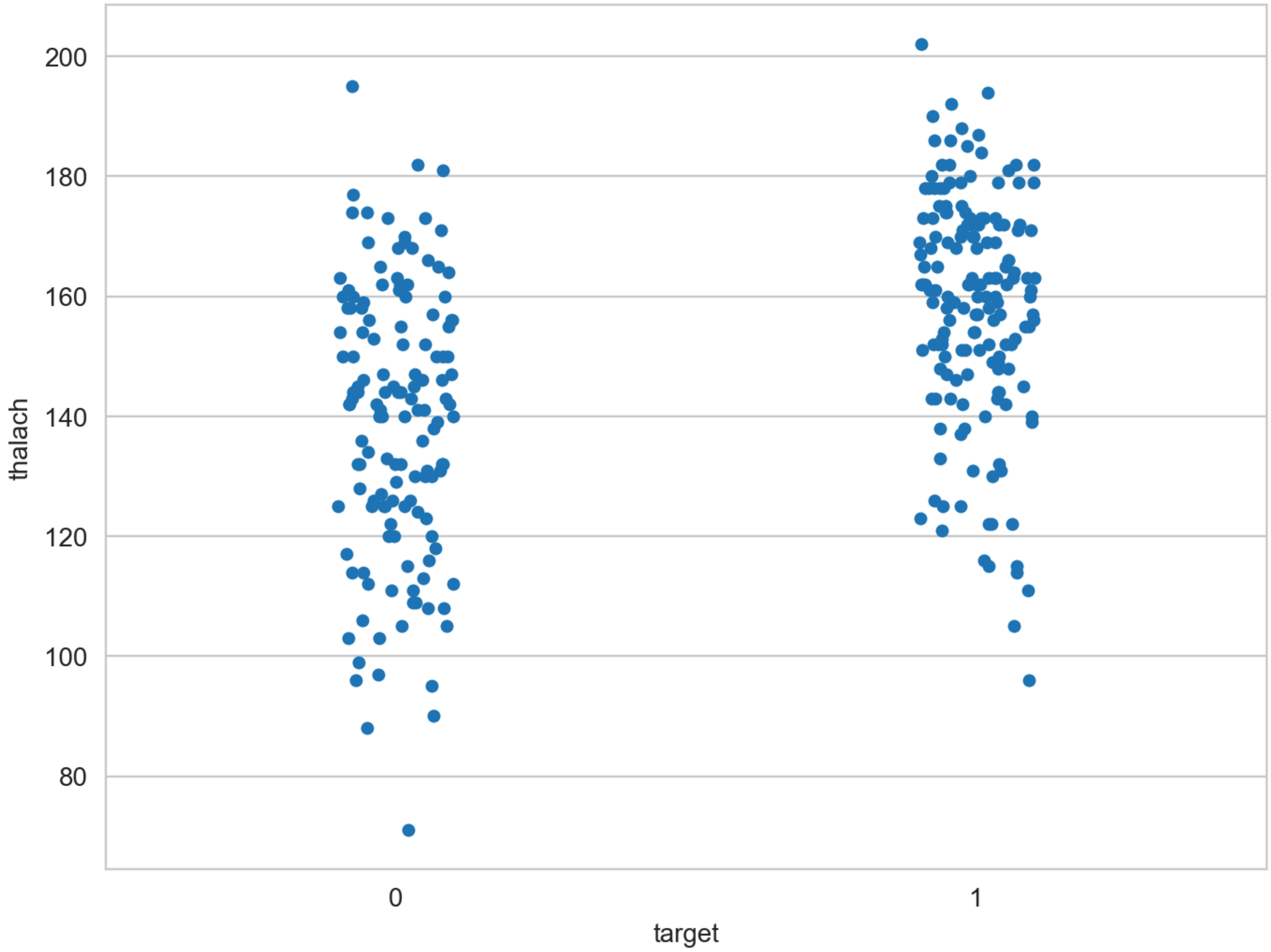
We can shade under the density curve and use a different color



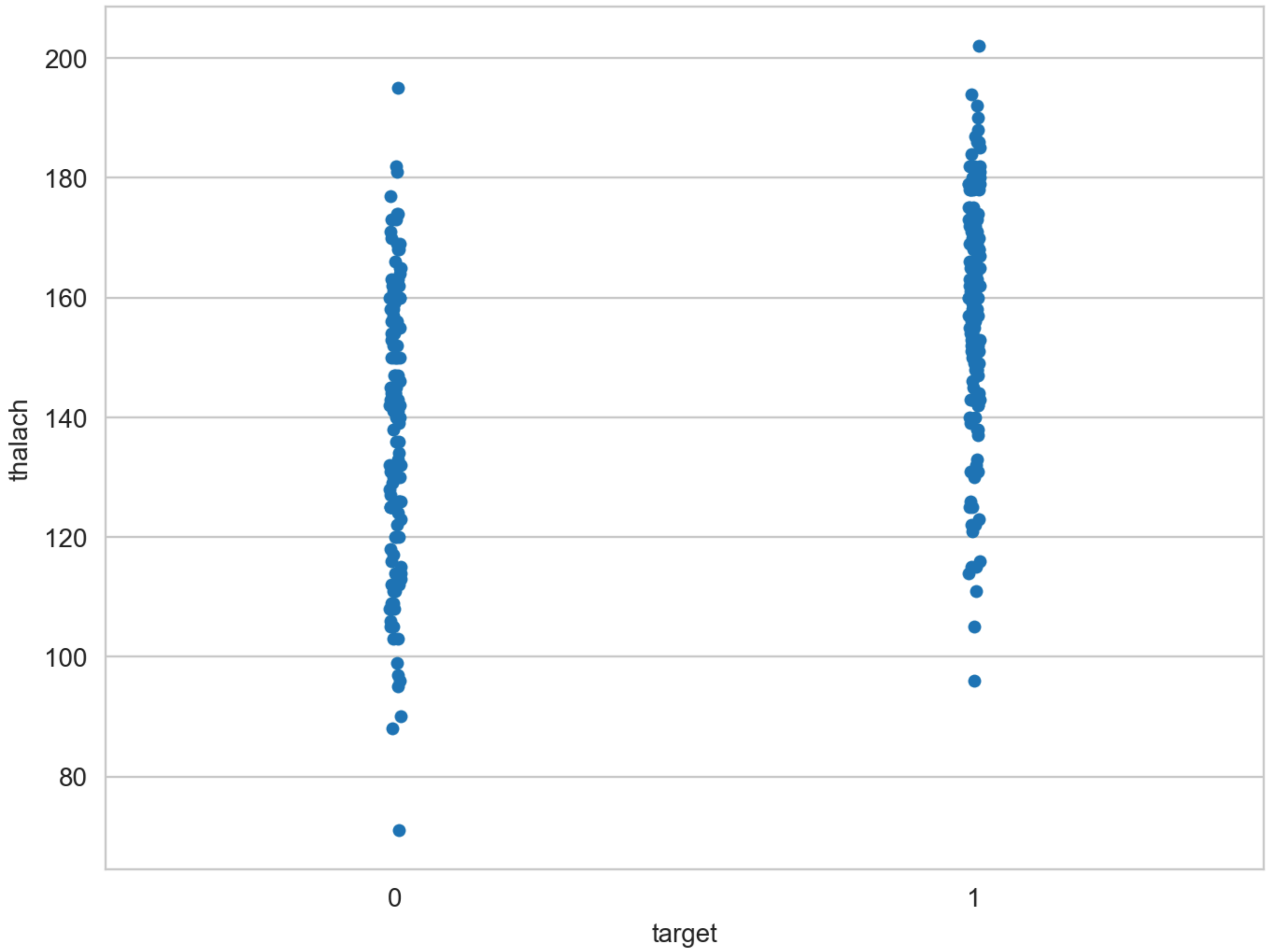
Histogram

- A histogram represents the distribution of data by forming bins along the range of the data and then drawing bars to show the number of observations that fall in each bin.
- We can plot a histogram as follows :

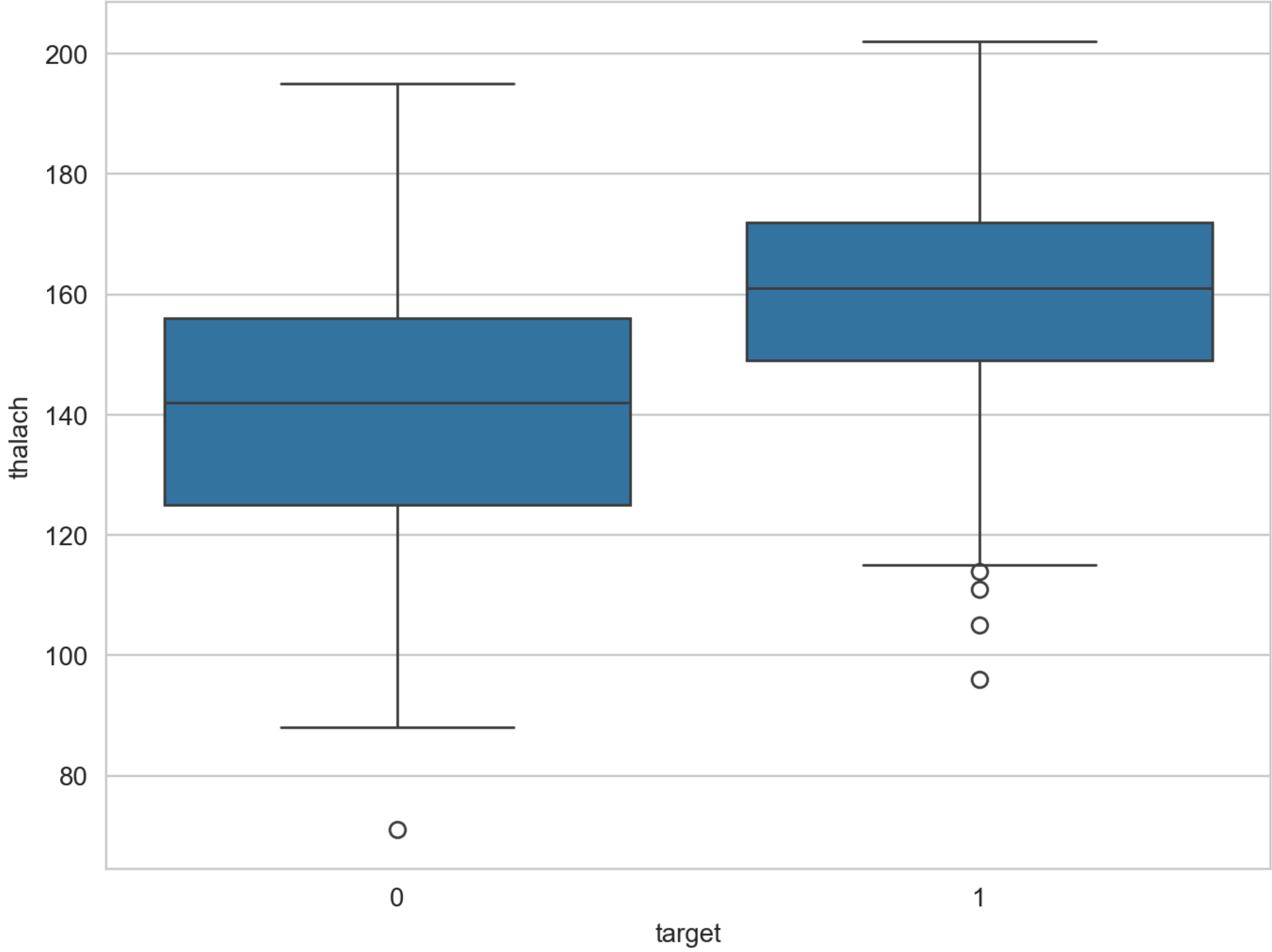




- We can see that those people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0). We can add jitter to bring out the distribution



Visualize distribution of `thalach` variable wrt `target` with boxplot



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 –

- 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).

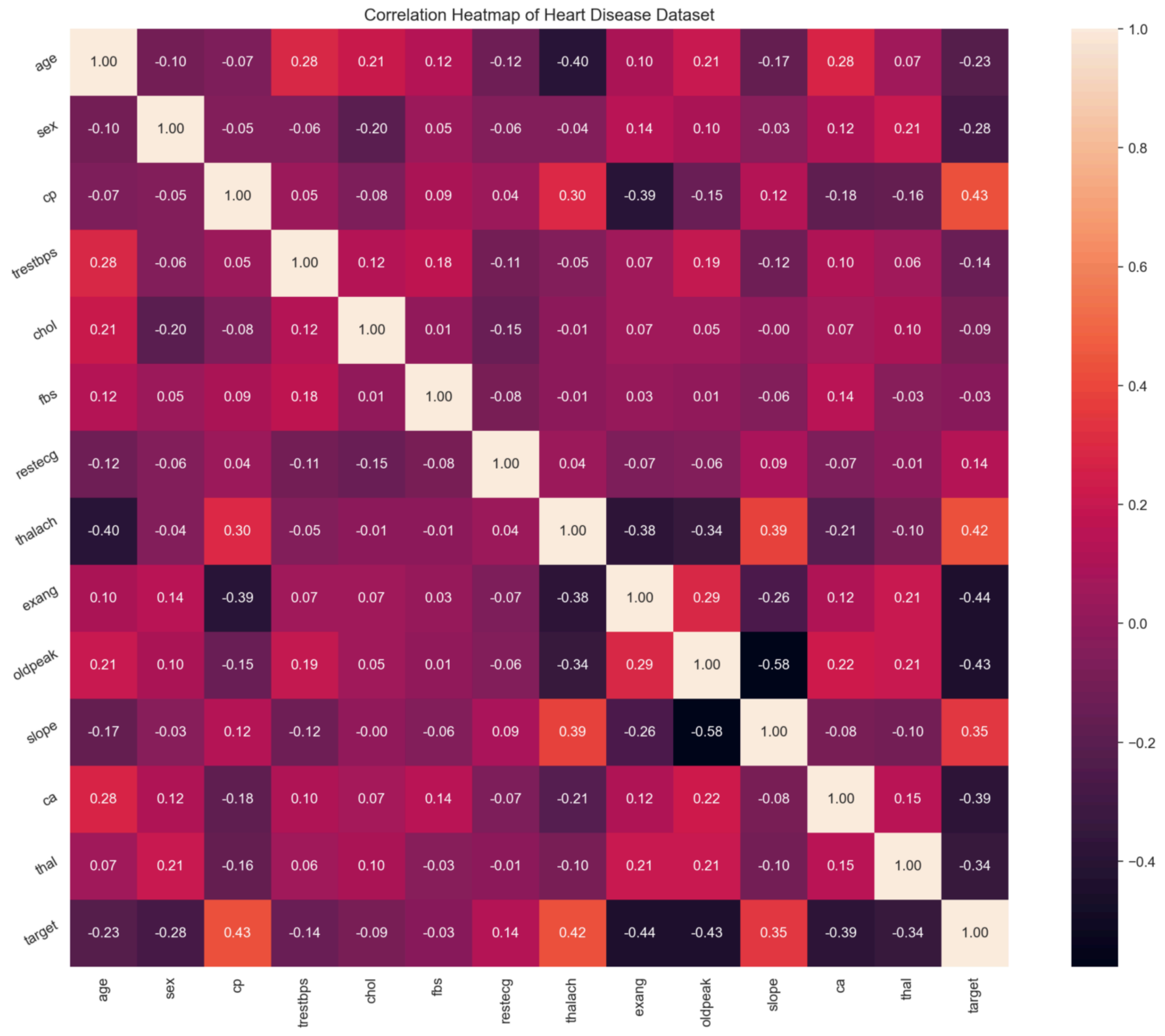
- The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).

9. 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`.


Correlation Heatmap of Heart Disease Dataset

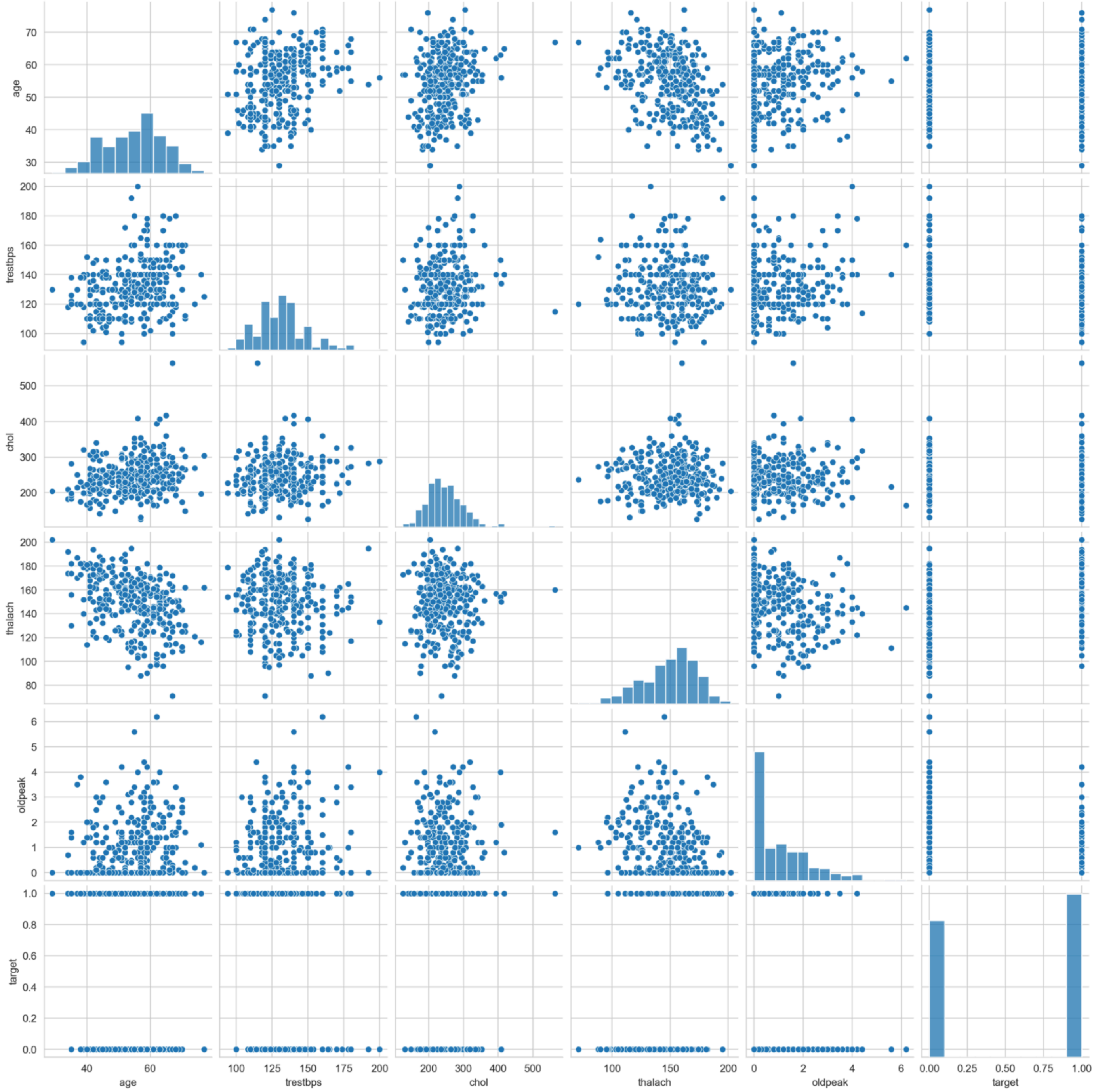
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.00	-0.10	-0.07	0.28	0.21	0.12	-0.12	-0.40	0.10	0.21	-0.17	0.28	0.07	-0.23
sex	-0.10	1.00	-0.05	-0.06	-0.20	0.05	-0.06	-0.04	0.14	0.10	-0.03	0.12	0.21	-0.28
cp	-0.07	-0.05	1.00	0.05	-0.08	0.09	0.04	0.30	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trestbps	0.28	-0.06	0.05	1.00	0.12	0.18	-0.11	-0.05	0.07	0.19	-0.12	0.10	0.06	-0.14
chol	0.21	-0.20	-0.08	0.12	1.00	0.01	-0.15	-0.01	0.07	0.05	-0.00	0.07	0.10	-0.09
fbs	0.12	0.05	0.09	0.18	0.01	1.00	-0.08	-0.01	0.03	0.01	-0.06	0.14	-0.03	-0.03
restecg	-0.12	-0.06	0.04	-0.11	-0.15	-0.08	1.00	0.04	-0.07	-0.06	0.09	-0.07	-0.01	0.14
thalach	-0.40	-0.04	0.30	-0.05	-0.01	-0.01	0.04	1.00	-0.38	-0.34	0.39	-0.21	-0.10	0.42
exang	0.10	0.14	-0.39	0.07	0.07	0.03	-0.07	-0.38	1.00	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.10	-0.15	0.19	0.05	0.01	-0.06	-0.34	0.29	1.00	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.03	0.12	-0.12	-0.00	-0.06	0.09	0.39	-0.26	-0.58	1.00	-0.08	-0.10	0.35
ca	0.28	0.12	-0.18	0.10	0.07	0.14	-0.07	-0.21	0.12	0.22	-0.08	1.00	0.15	-0.39
thal	0.07	0.21	-0.16	0.06	0.10	-0.03	-0.01	-0.10	0.21	0.21	-0.10	0.15	1.00	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.09	-0.03	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1.00

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 weakly negatively correlated (correlation coefficient = -0.34).

Pair Plot



- 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 will check relationships between these variables.

Analysis of `age` and other variables

Check the number of unique values in `age` variable

```
41
```

View statistical summary of `age` variable##

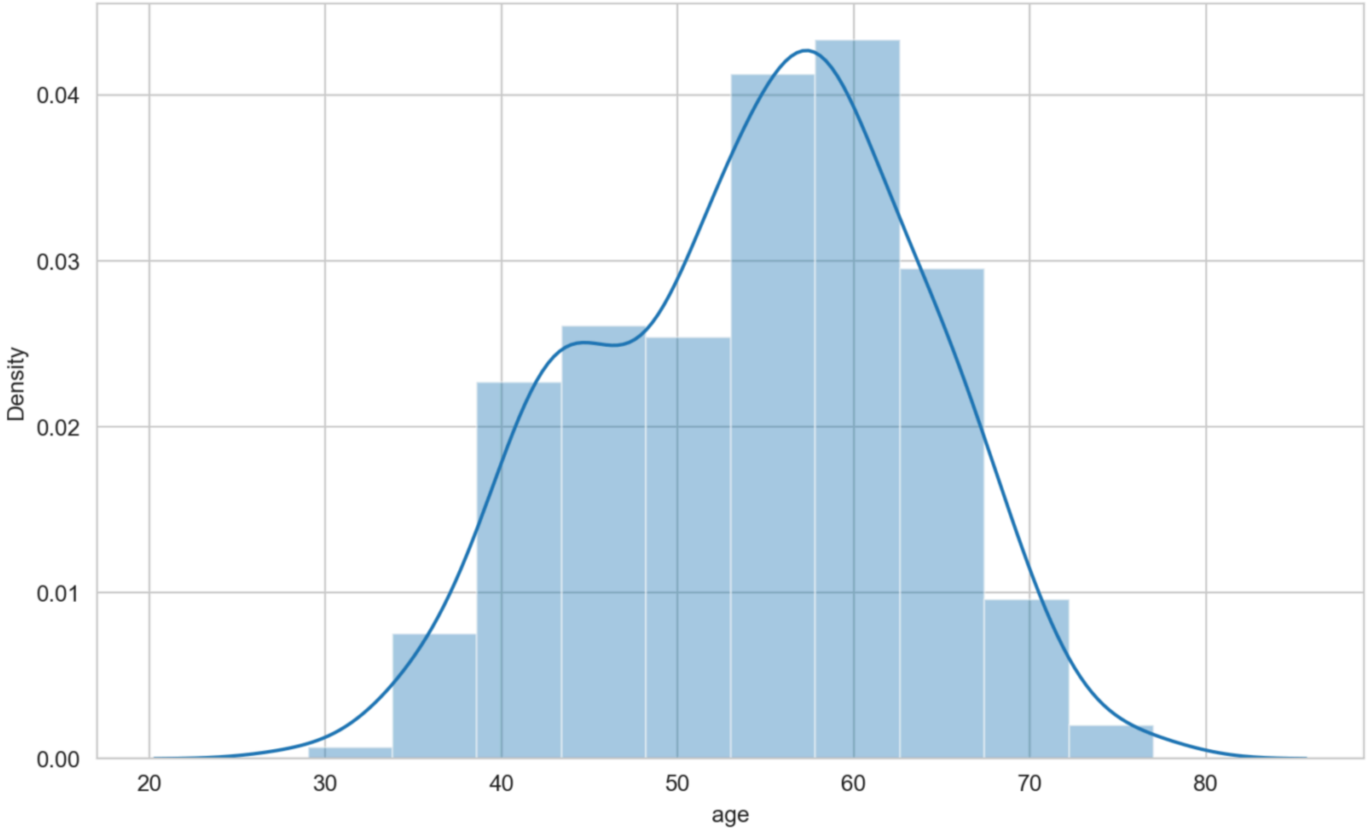
	age
count	303
mean	54.3663
std	9.0821
min	29
25%	47.5
50%	55
75%	61
max	77

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.

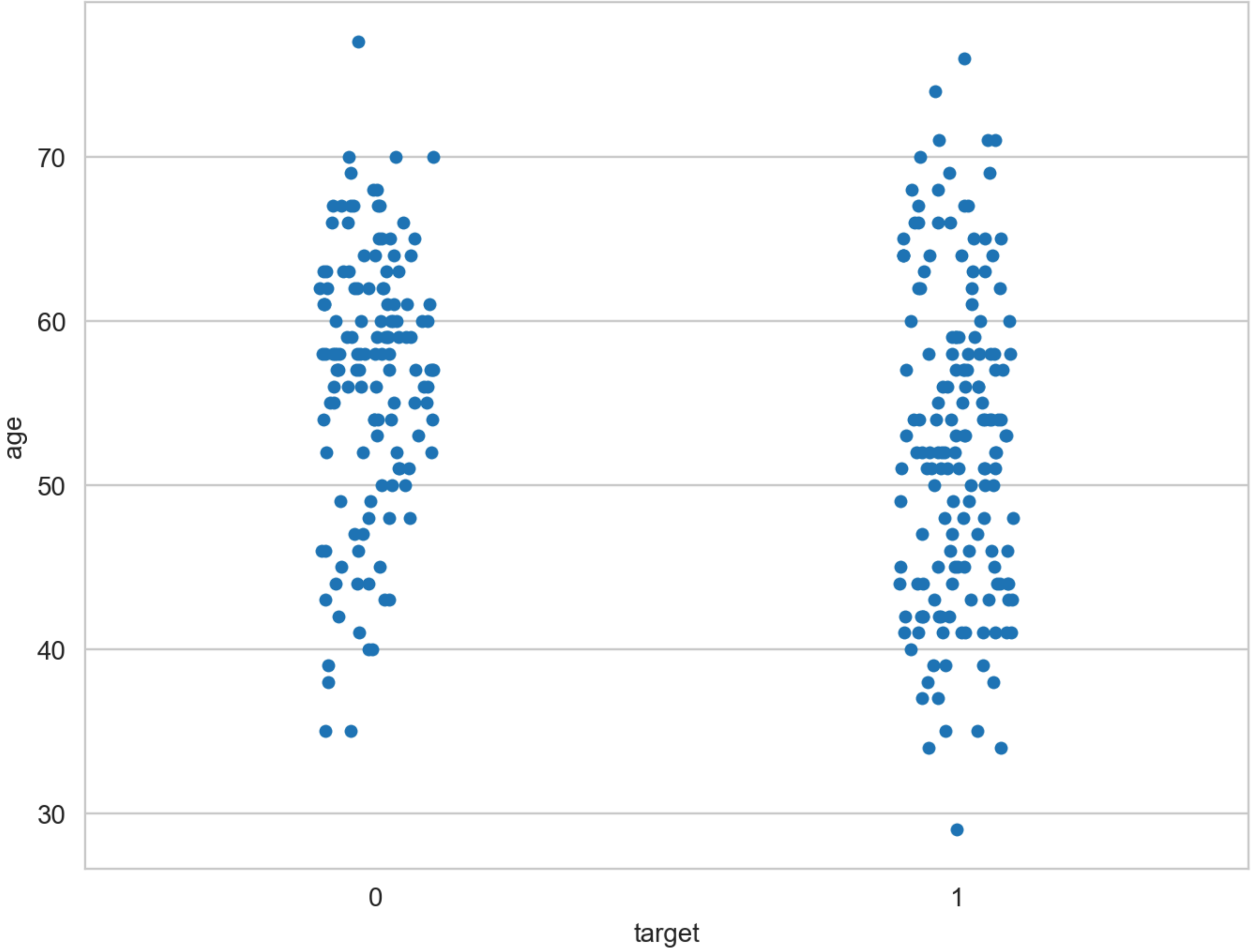


Interpretation

- The `age` variable distribution is approximately normal.

Analyze `age` and `target` variable

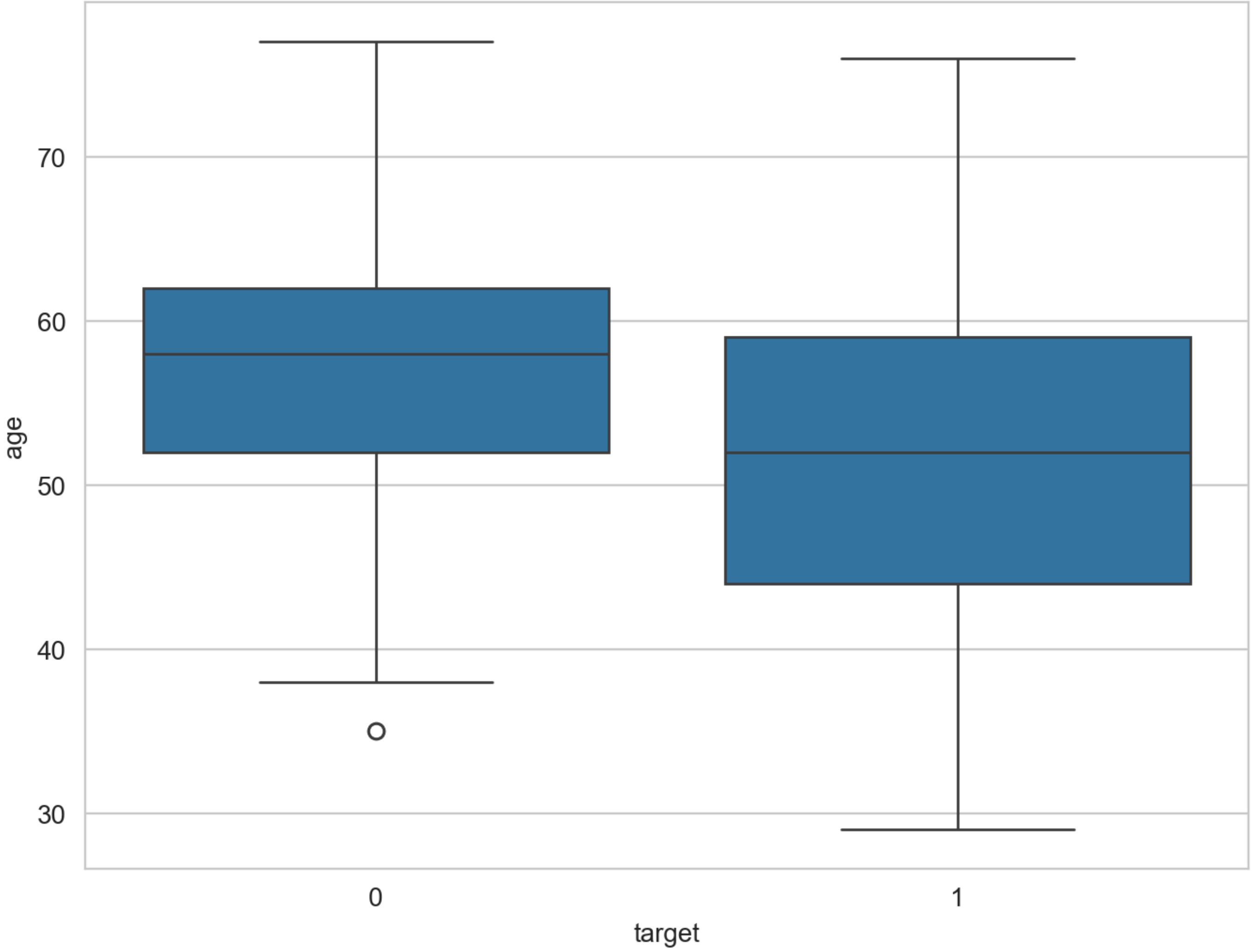
Visualize frequency distribution of `age` variable wrt `target`



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.

Visualize distribution of `age` variable wrt `target` with boxplot

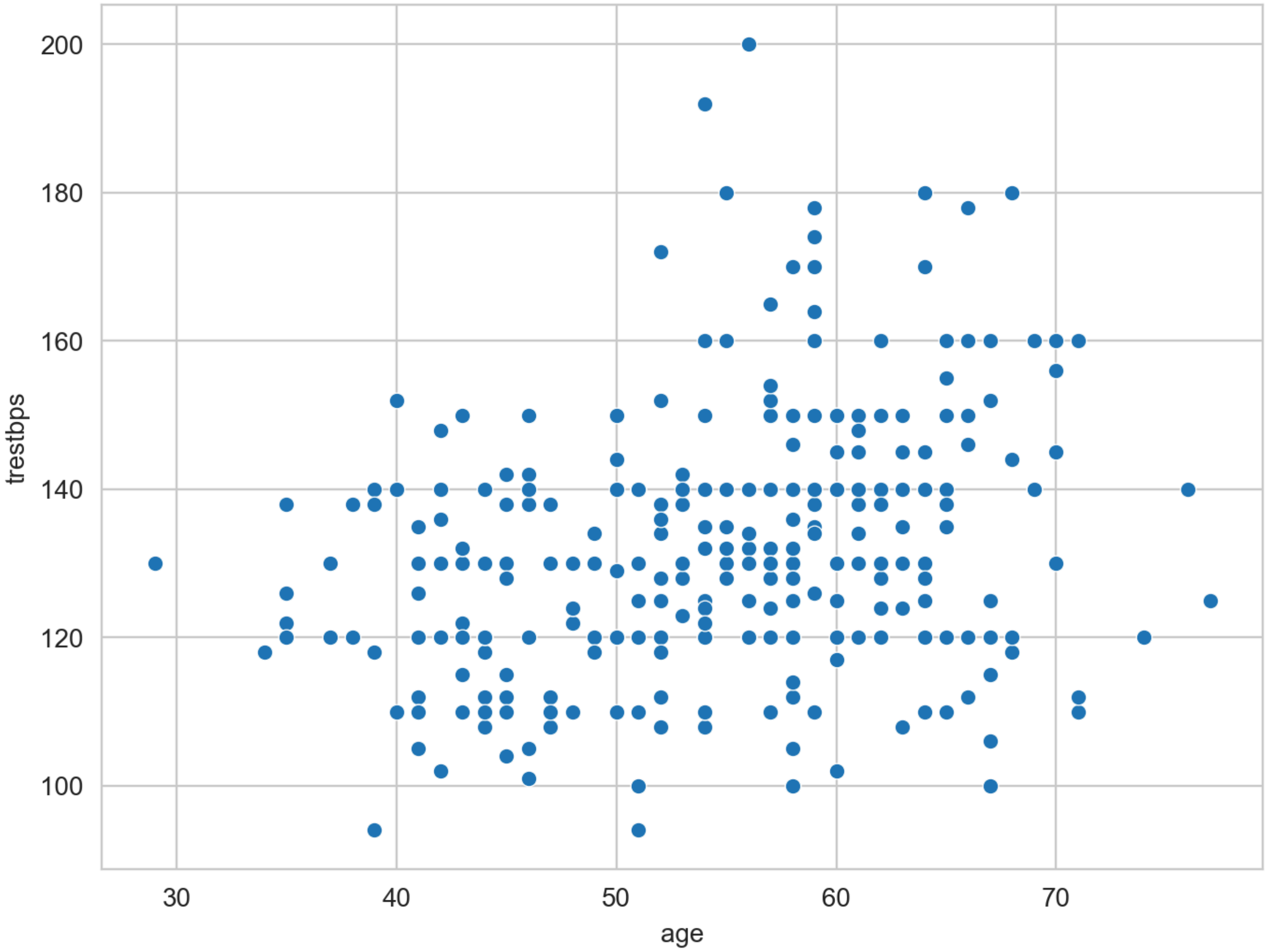


Interpretation

- 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

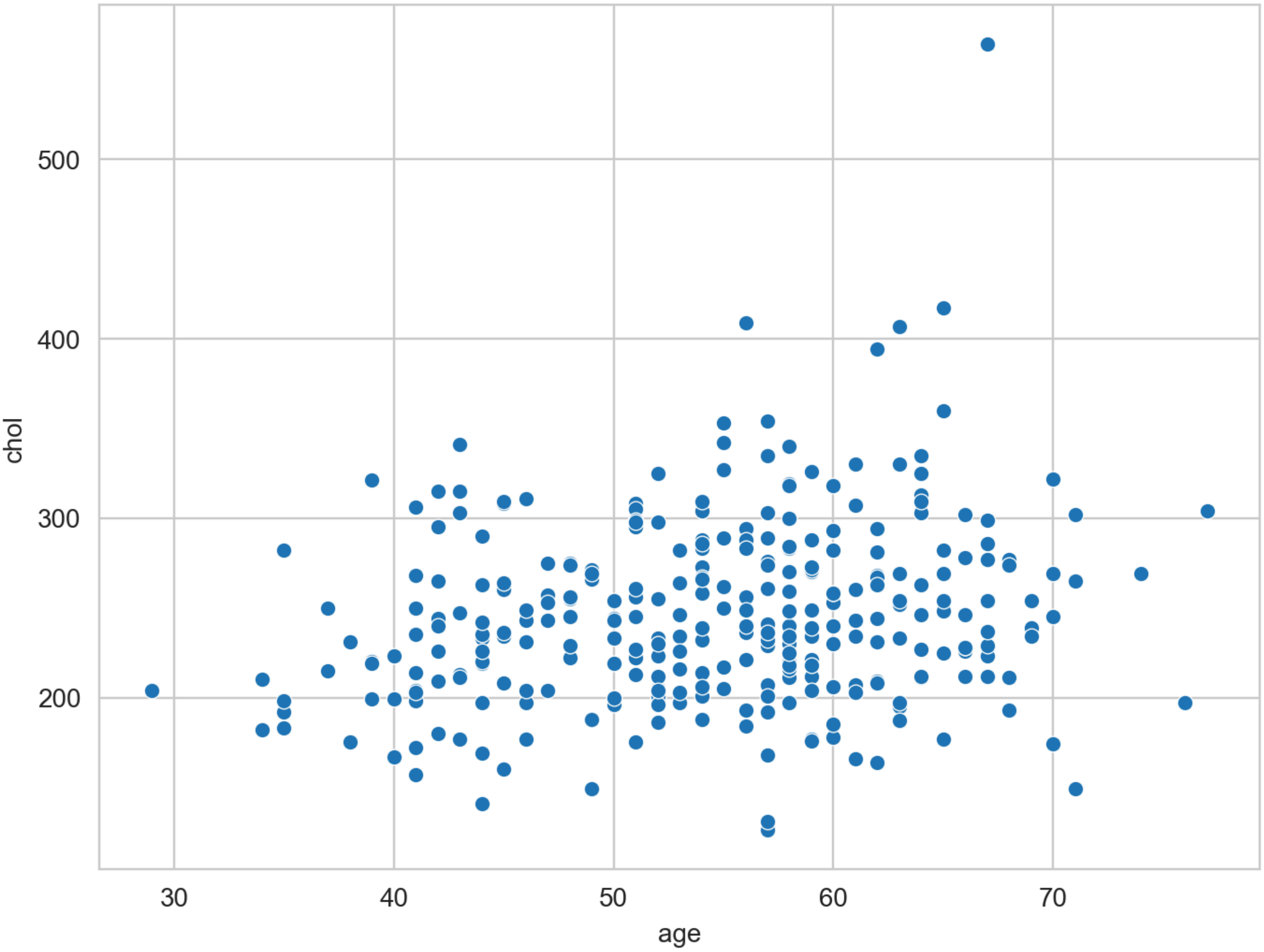
I will plot a scatterplot to visualize the relationship between `age` and `trestbps` variable.

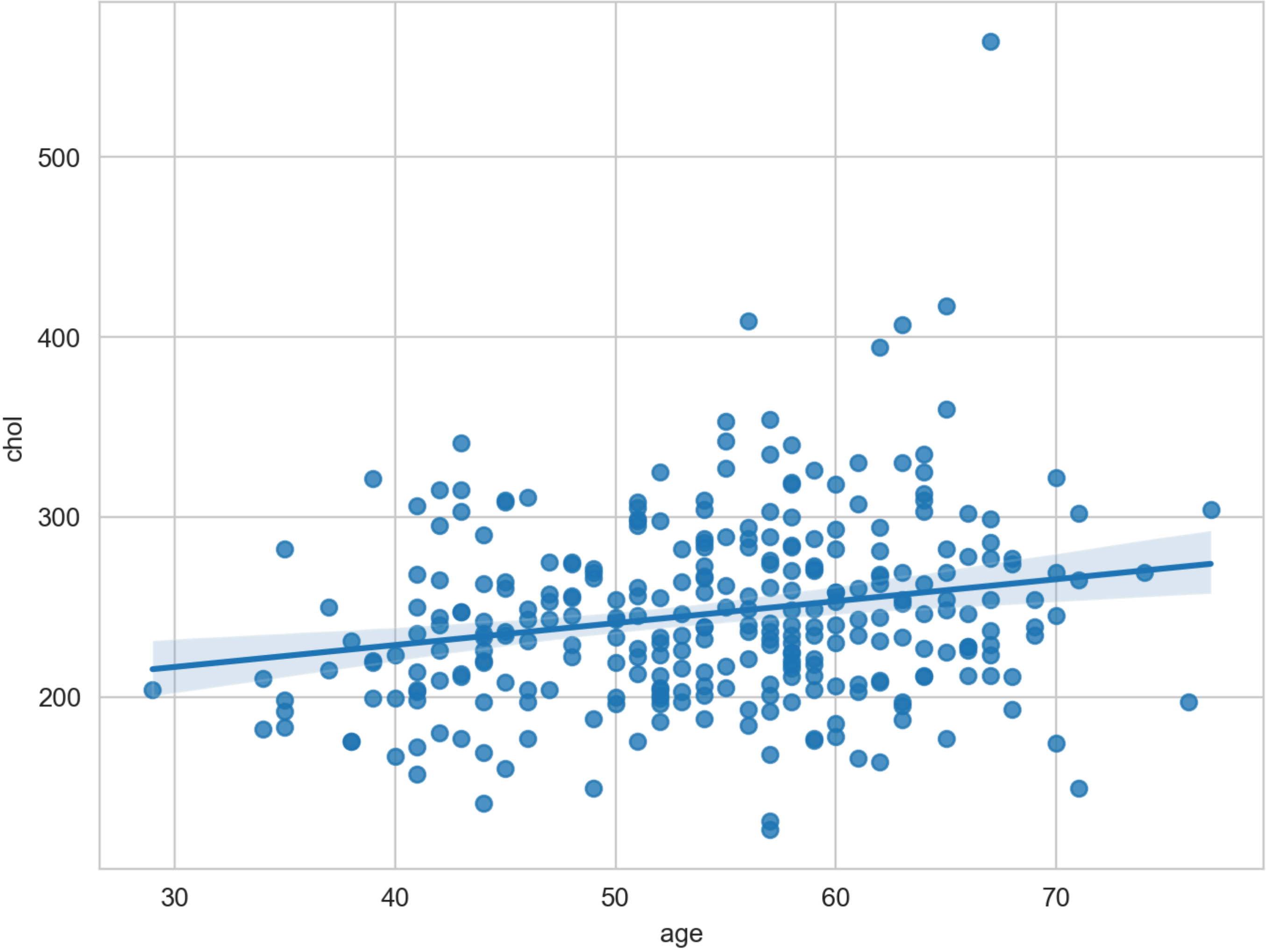


Interpretation

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

Analyze `age` and `chol` variable

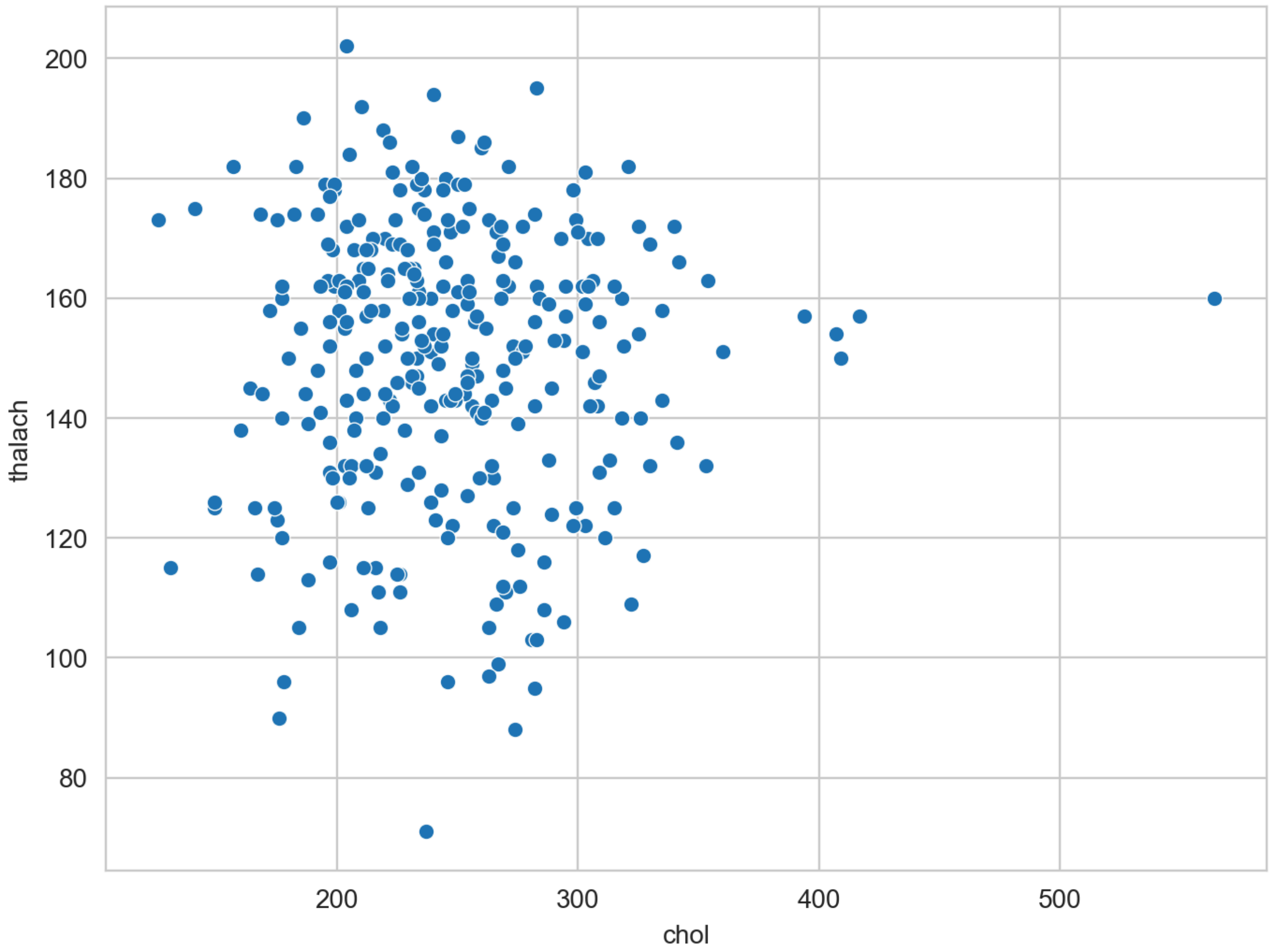

500
400
300
200
chol
30
40
50
60
70
age

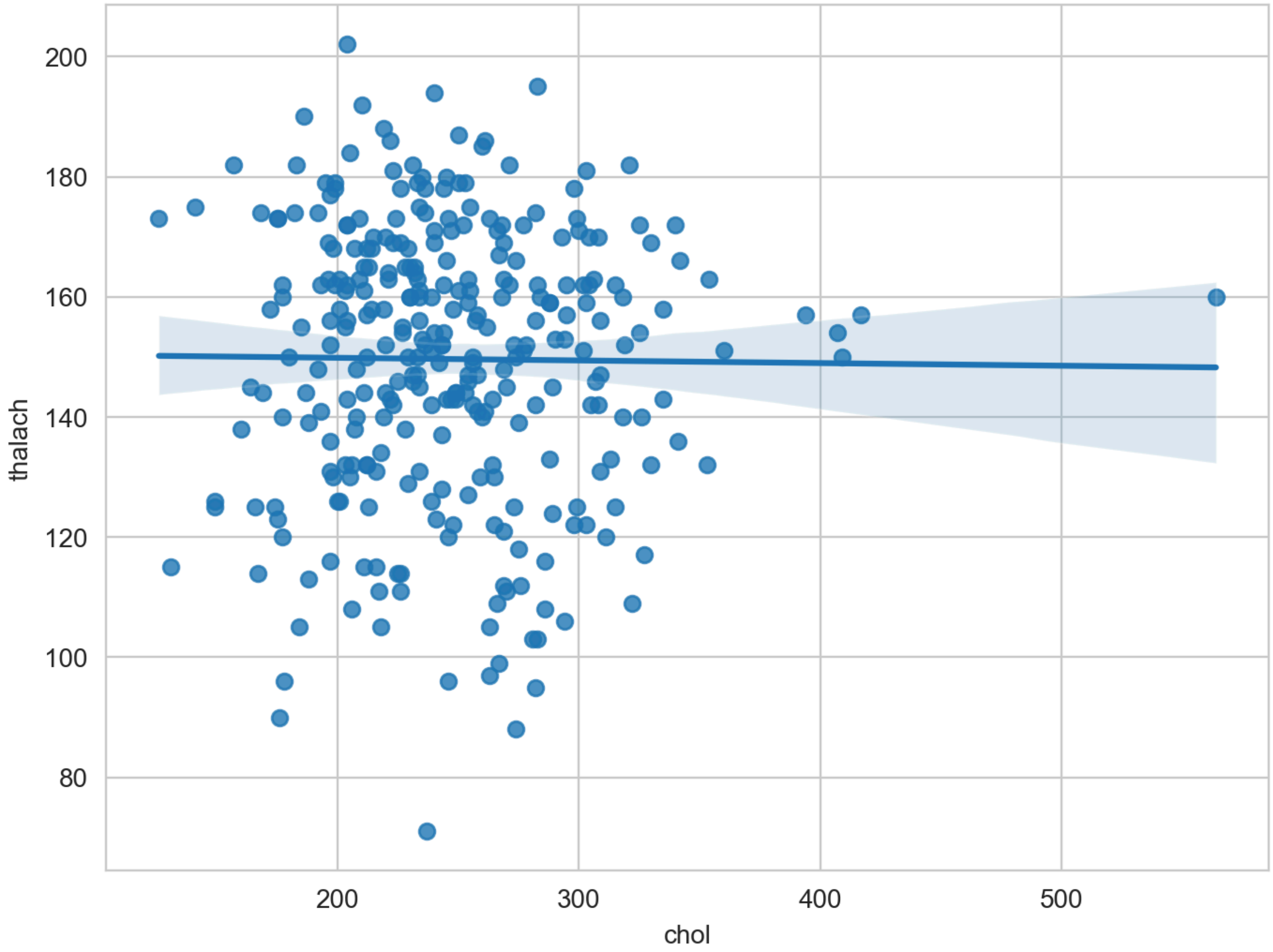


Interpretation

- The above plot confirms that there is a slightly positive correlation between `age` and `chol` variables.

Analyze `chol` and `thalach` variable


thalach
200
180
160
140
120
100
80
chol
200
300
400
500



Interpretation

- The above plot shows that there is no correlation between `chol` and `thalach` variable.

10. Dealing with missing values

- In Pandas missing data is represented by two values:
  - **None**: None is a Python singleton object that is often used for missing data in Python code.
  - **NaN** : NaN (an acronym for Not a Number), is a special floating-point value recognized by all systems that use the standard IEEE floating-point representation.
- There are different methods in place on how to detect missing values.

Pandas isnull() and notnull() functions

- Pandas offers two functions to test for missing data - `isnull()` and `notnull()` . These are simple functions that return a boolean value indicating whether the passed in argument value is in fact missing data.

- Below, I will list some useful commands to deal with missing values.

Useful commands to detect missing values

- **heart.isnull()**

The above command checks whether each cell in a dataframe contains missing values or not. If the cell contains missing value, it returns True otherwise it returns False.

- **heart.isnull().sum()**

The above command returns total number of missing values in each column in the dataframe.

- **heart.isnull().sum().sum()**

It returns total number of missing values in the dataframe.

- **heart.isnull().mean()**

It returns percentage of missing values in each column in the dataframe.

- **heart.isnull().any()**

It checks which column has null values and which has not. The columns which has null values returns TRUE and FALSE otherwise.

- **heart.isnull().any().any()**

It returns a boolean value indicating whether the dataframe has missing values or not. If dataframe contains missing values it returns TRUE and FALSE otherwise.

- **heart.isnull().values.any()**

It checks whether a particular column has missing values or not. If the column contains missing values, then it returns TRUE otherwise FALSE.

- **heart.isnull().values.sum()**

It returns the total number of missing values in the dataframe.

	0
age	0
sex	0
cp	0
trestbps	0
chol	0
fbs	0
restecg	0
thalach	0
exang	0
oldpeak	0

We can see that there are no missing values in the dataset.

11. Check with ASSERT statement

- We must confirm that our dataset has no missing values.
- We can write an **assert statement** to verify this.
- We can use an assert statement to programmatically check that no missing, unexpected 0 or negative values are present.
- This gives us confidence that our code is running properly.
- **Assert statement** will return nothing if the value being tested is true and will throw an AssertionError if the value is false.
- **Asserts**
  - assert 1 == 1 (return Nothing if the value is True)
  - assert 1 == 2 (return AssertionError if the value is False)
- The above two commands do not throw any error. Hence, it is confirmed that there are no missing or negative values in the dataset.
- All the values are greater than or equal to zero.

12. Outlier detection

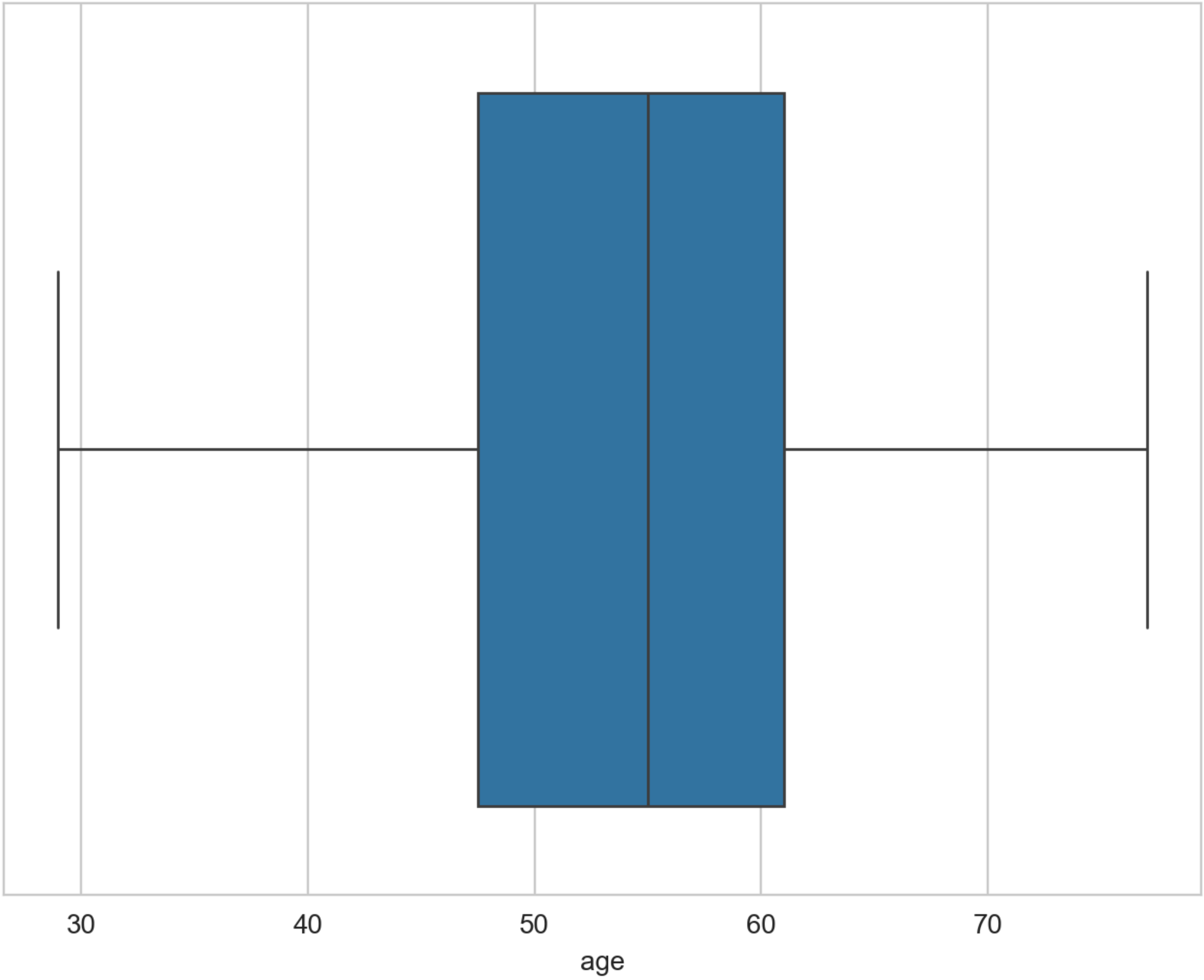
I will make boxplots to visualise outliers in the continuous numerical variables : -

`age` , `trestbps` , `chol` , `thalach` and `oldpeak` variables.

`age` variable

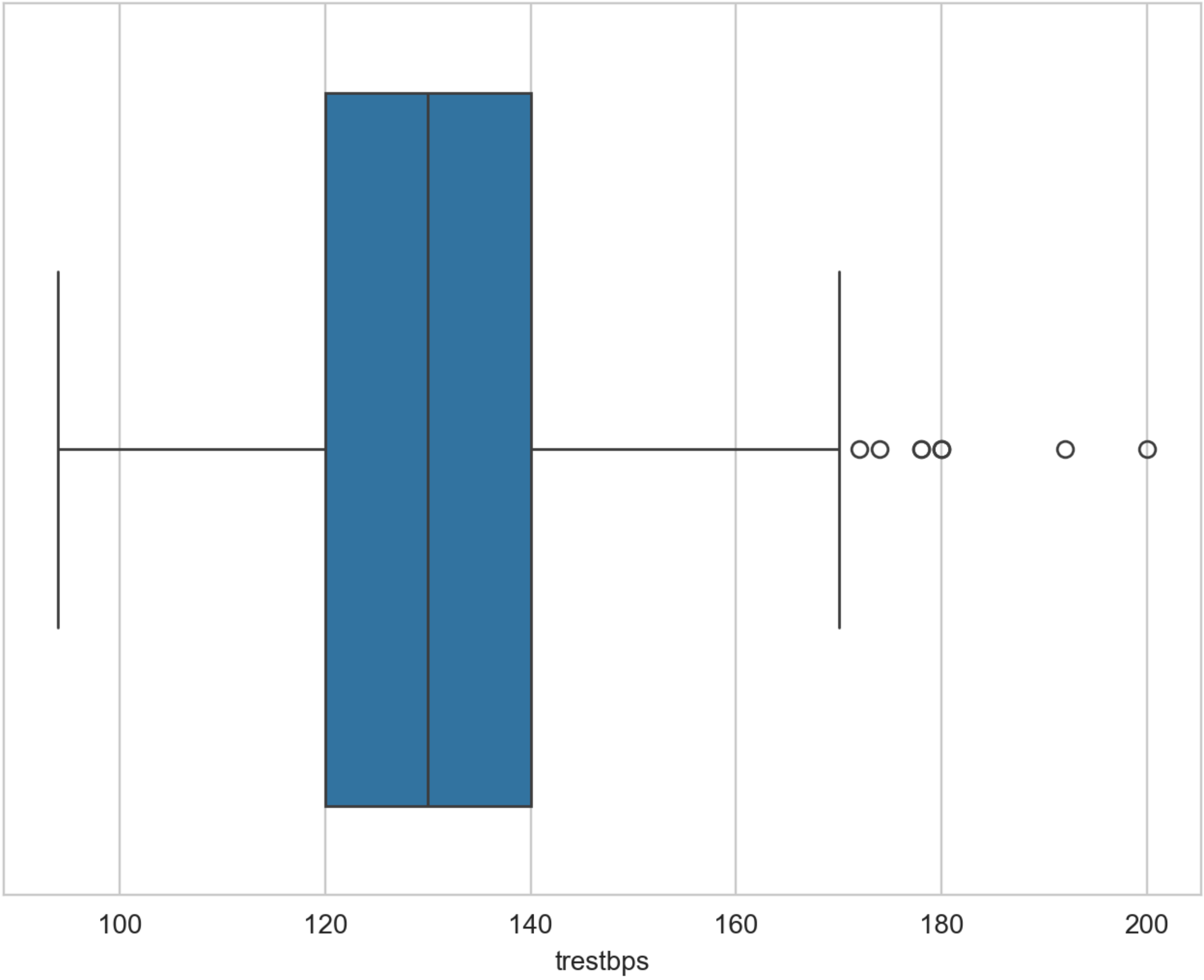
	age
count	303
mean	54.3663
std	9.0821
min	29
25%	47.5
50%	55
75%	61
max	77

Box-plot of `age` variable



`trestbps` variable

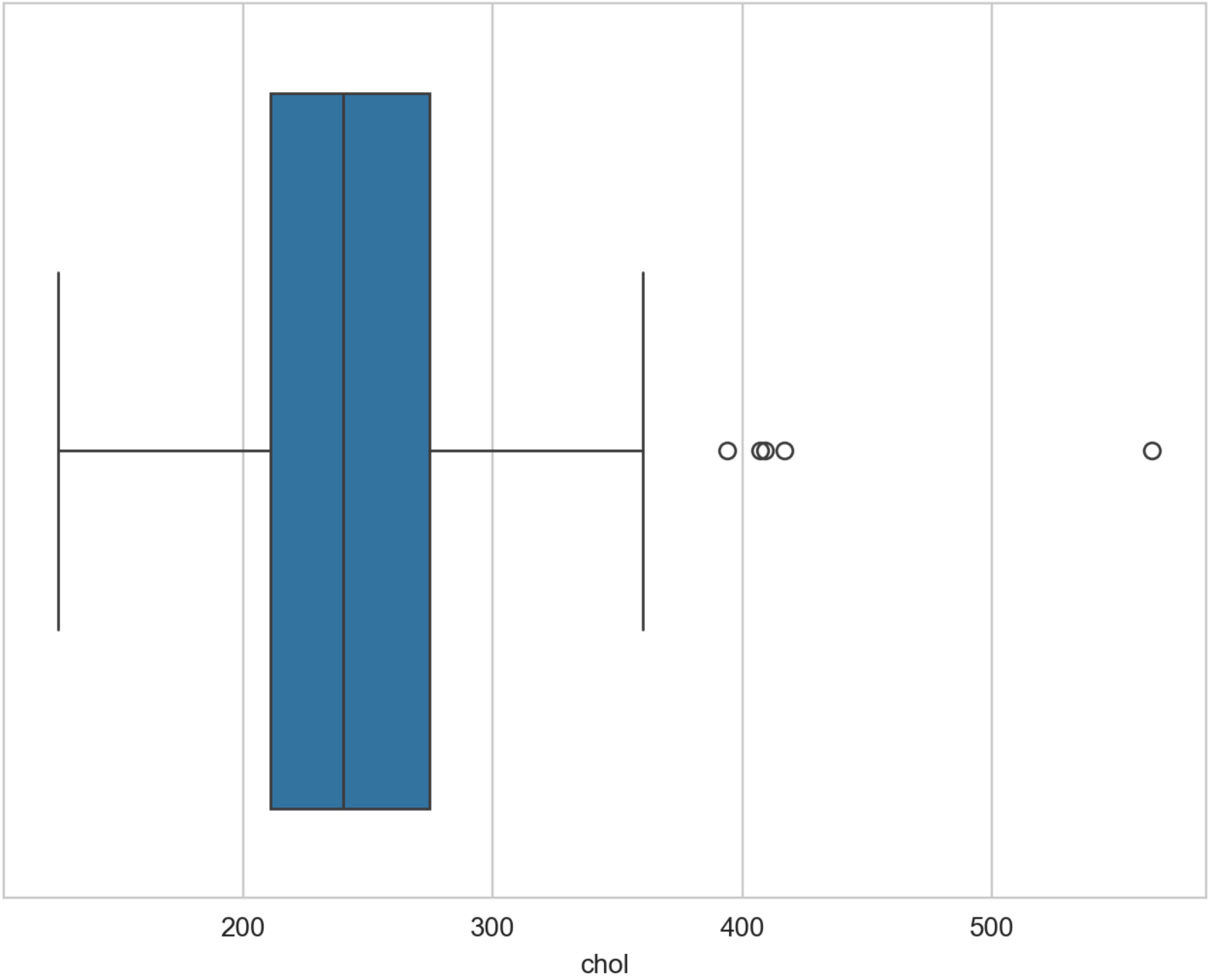
	trestbps
count	303
mean	131.6238
std	17.5381
min	94
25%	120
50%	130
75%	140
max	200



`chol` variable

	chol
count	303
mean	246.264
std	51.8308
min	126
25%	211
50%	240
75%	274.5
max	564

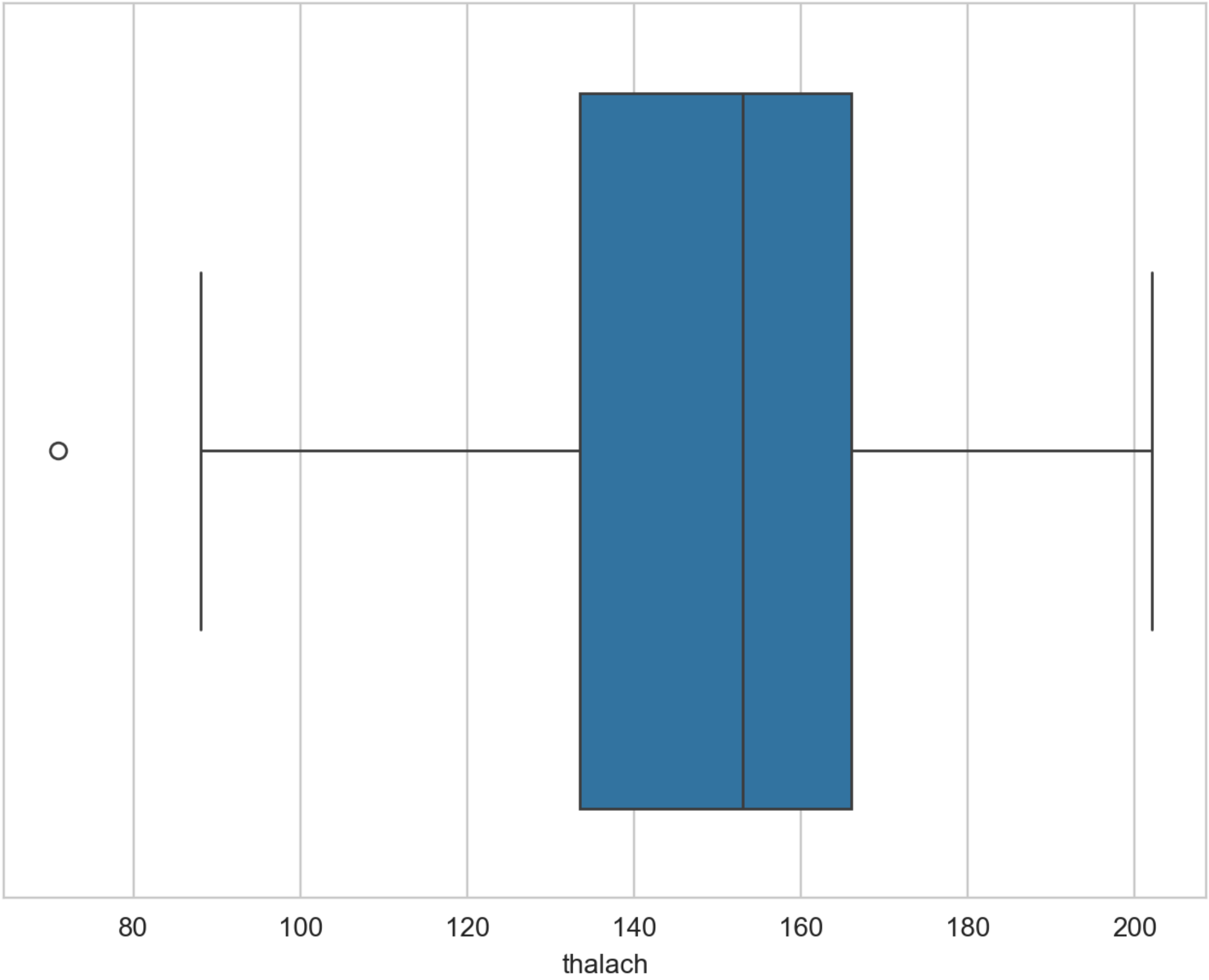
Box-plot of `chol` variable



`thalach` variable

	thalach
count	303
mean	149.6469
std	22.9052
min	71
25%	133.5
50%	153
75%	166
max	202

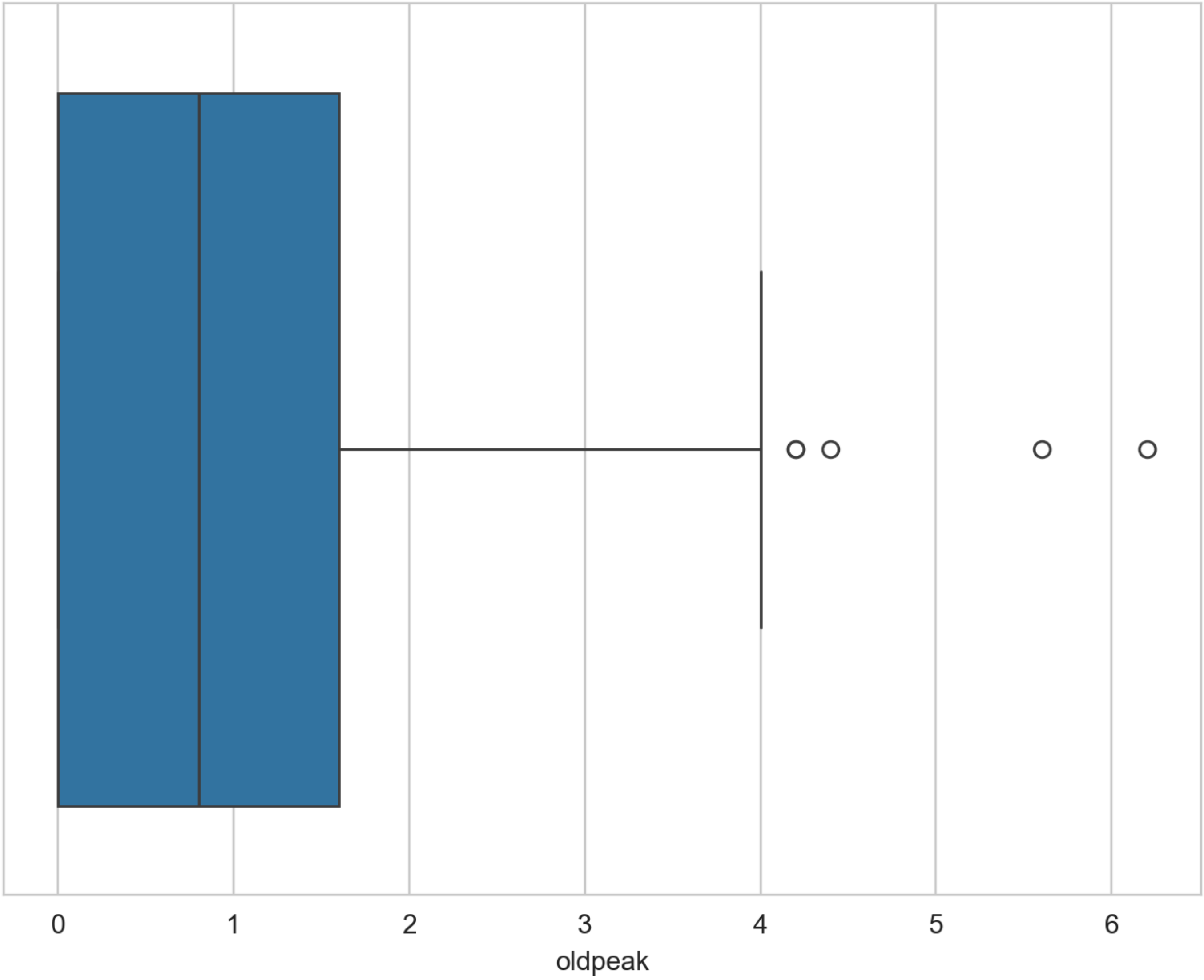
Box-plot of `thalach` variable



`oldpeak` variable

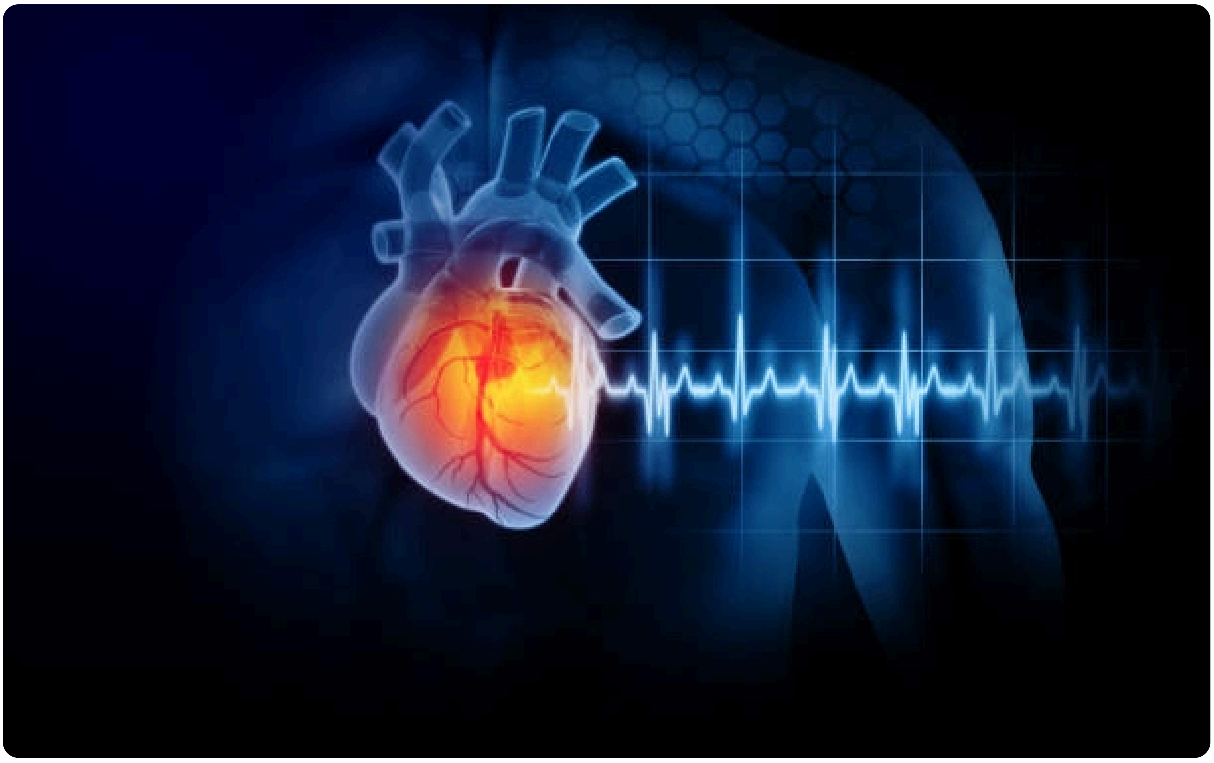
	oldpeak
count	303
mean	1.0396
std	1.1611
min	0
25%	0
50%	0.8
75%	1.6
max	6.2

Box-plot of `oldpeak` variable



Findings

- The `age` variable does not contain any outlier.
- `trestbps` variable contains outliers to the right side.
- `chol` variable also contains outliers to the right side.
- `thalach` variable contains a single outlier to the left side.
- `oldpeak` variable contains outliers to the right side.
- Those variables containing outliers needs further investigation.



Heart Stroke Image