

Group No. - 37

**Data Science and Analysis Project on**

**Heart Disease Prediction**

**Presented by: -**

**Gaurav Dilip Shripad**

**Heart Disease Prediction**

* **TABLE OF CONTENT**

|  |  |
| --- | --- |
| **1.** | TABLE OF FIGURES |
| **2.** | ABSTRACT |
| **3.** | CHAPTERS |
| **4.** | COMPLETE CODE |
| **5.** | CONCLUSION |

1. **TABLE OF FIGURES / TABLES**
2. **ABSTRACT**

We are creating a Data Science and Analysis Project on Heart Disease Prediction. The project uses raw data in form of .csv and .xls files and transforms it into a Data Analysis. This project is attempt of data analysing on Heart Disease Prediction with the help of Data Science and Data Analysis in python code. This analysis will help us to find the basis behind common notions about the disease prediction found between which age from purely a dataset perspective.

The data used for this prosect is across the file-

Heart.csv

For the analysis we use some libraries such as pandas, sklearn, numpy, matplotlib, seaborn, plotly, cufflinks, os, accuracy\_score, plotly.offlineinit\_notebook\_mode, plot and iplot.

In the project, we import fie and explore the parameters in file. By using some user-defined functions use to clean data.

We also filter off the unwanted entries, thus making our data perfect for analysis.

Then we use the data from the data set in dataframes various graphs for further analysis. For the heart disease, we use age-wise, sex-wise,

Chest pain-wise, cholesterol-wise, etc.

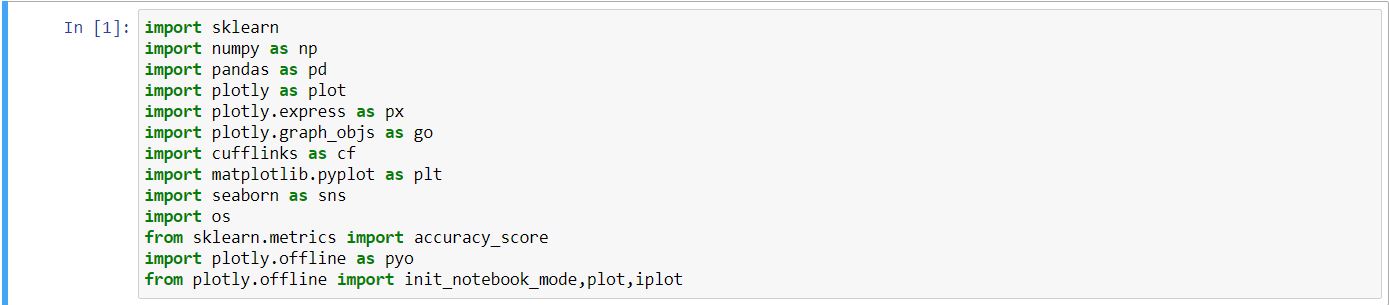
For analysis we use various graphs such as sns.barplot, sns.bar, px.bar, sns.displot, sns.pairplot, etc.

For visualising we use various types of maps such as correlation heat maps, various type of barplot, displot, figures, pairplots.

For further analysis we use the concept of sorting x and y, Train and Test, Decision tree, predictions, and concept of KNN.

1. **CHAPTERS**

Importing data visualization libraries in python



* **numpy: To work with arrays.**
* **pandas: To work with csv files and dataframes**
* **matplotlib: To create charts using pyplot**
* **sklearn:** **It features various**[**classification**](https://en.wikipedia.org/wiki/Statistical_classification)**,**[**regression**](https://en.wikipedia.org/wiki/Regression_analysis)**and**[**clustering**](https://en.wikipedia.org/wiki/Cluster_analysis)**algorithms**
* **plotly:** **open-source plotting library that supports** **different types charts and graphs**
* **plotly.express:** **The plotly. express module (usually imported as px) contains functions that can create entire figures at once, and is referred to as Plotly Express or PX. Plotly Express is a built-in part of the plotly library, and is the recommended starting point for creating most common figures.**
* **plotly.graph\_objs: to create, manipulate and render graphical figures (i.e. charts, plots, maps and diagrams) represented by data structures also referred to as figures.**
* **cufflinks: cufflink connects plotly with pandas to create graphs and charts of dataframes directly.**
* **seaborn:  It provides a high-level interface for drawing attractive and informative statistical graphics.**

Import dataset

In order to display the plot inside the notebook, we need to initiate plotly’s notebook mode as follows –

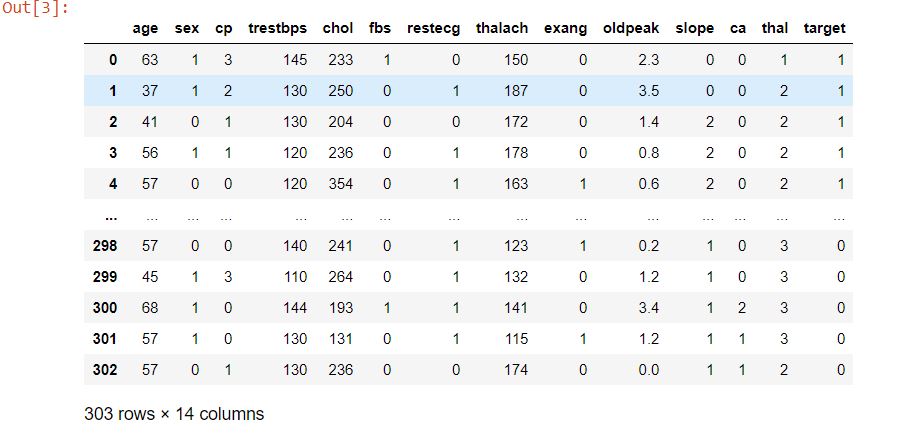


Keep rest of the script as it is and run the notebook cell by pressing **Shift+Enter**. Graph will be displayed offline inside the notebook itself.

I saved it to my working directory with the name as heart.csv



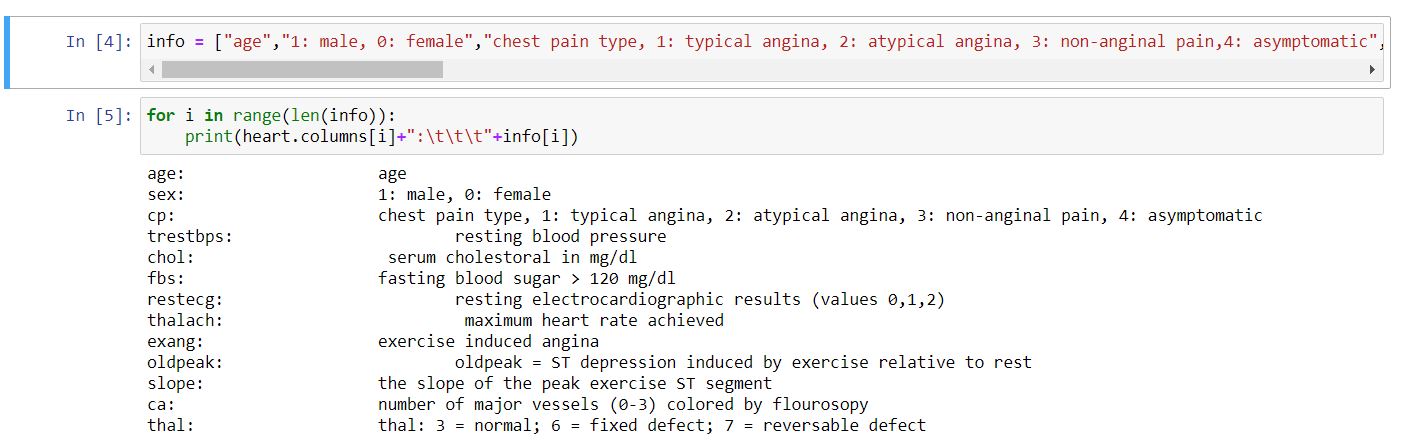
By using ‘heart’ it shows the file’s inside document



which has a total of 303 rows with respect to 14 columns.

By using the info

info = ["age","1: male, 0: female","chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain,4: asymptomatic","resting blood pressure"," serum cholestoral in mg/dl","fasting blood sugar > 120 mg/dl","resting electrocardiographic results (values 0,1,2)"," maximum heart rate achieved","exercise induced angina","oldpeak = ST depression induced by exercise relative to rest","the slope of the peak exercise ST segment","number of major vessels (0-3) colored by flourosopy","thal: 3 = normal; 6 = fixed defect; 7 = reversable defect"]



It shows the output as the full form allotted to the dataset short forms

age: age

sex: 1: male, 0: female

cp: chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain,4: asymptomatic

trestbps: resting blood pressure

chol: serum cholestoral in mg/dl

fbs: fasting blood sugar > 120 mg/dl

restecg: resting electrocardiographic results (values 0,1,2)

thalach: maximum heart rate achieved

exang: exercise induced angina

oldpeak: 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: thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

By using the heart['target'] it sorts the data base and shows an output as

0 1

1 1

2 1

3 1

4 1

..

298 0

299 0

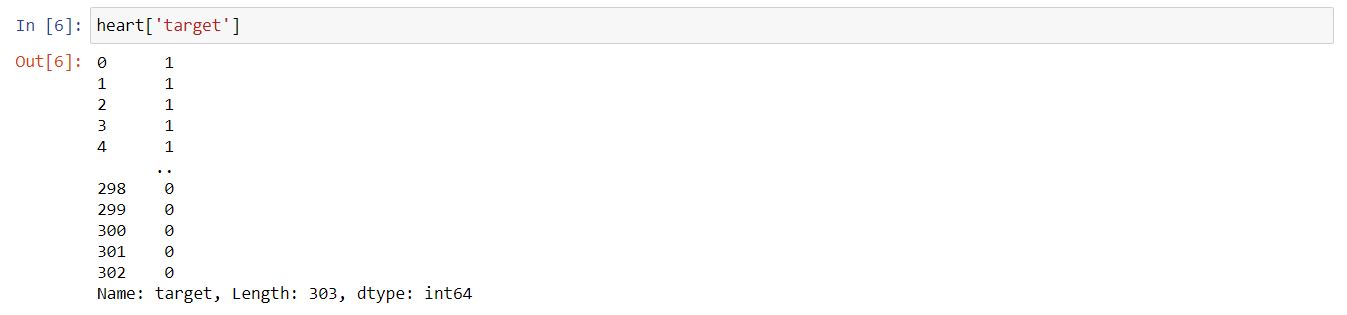
300 0

301 0

302 0

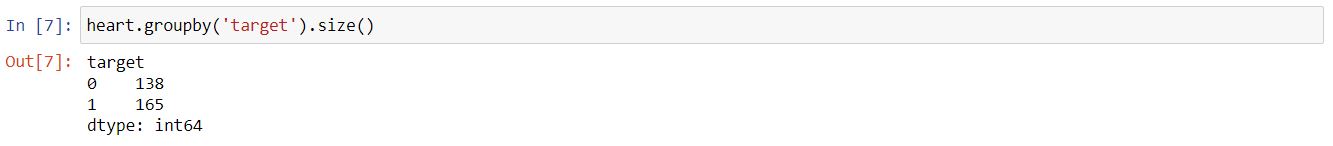
Name: target, Length: 303, dtype: int64

Which let us to describe the person has heart disease or not.



heart.groupby('target').size()

tell us the size means total how many no of people are infected by heart disease.



Which shows the result as

target

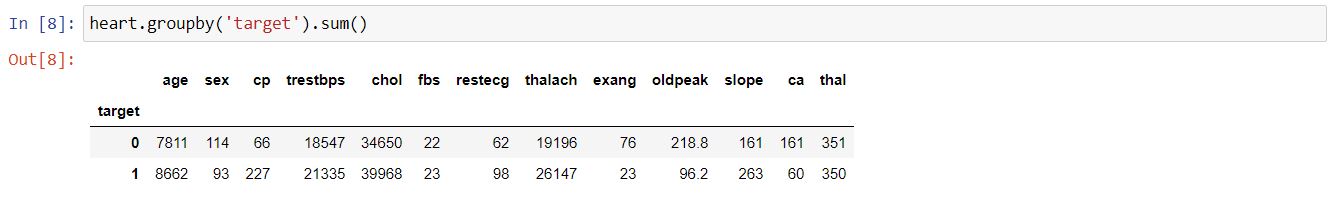
0 138

1 165

dtype: int64

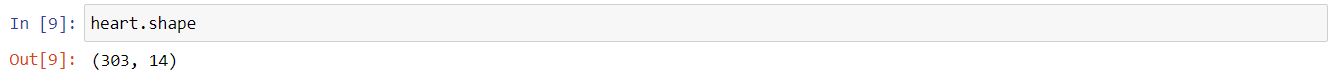
By using heart.groupby('target').sum()

It shows the chart which define the no of person infected by heart disease

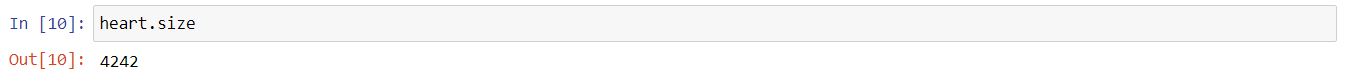


Which help us to understand by using table format.

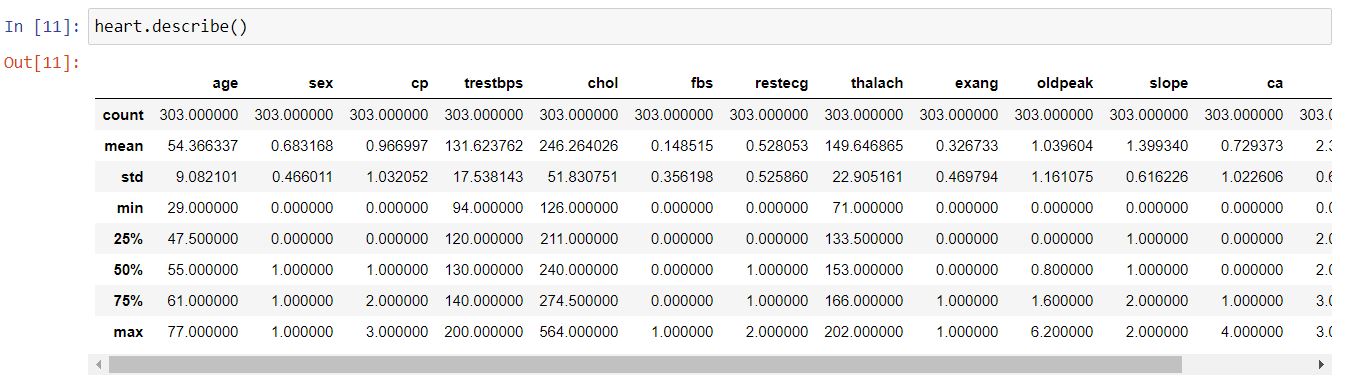
heart. shape is use to describe number of rows and columns.



heart.size is used to describe the total amount of cells present.

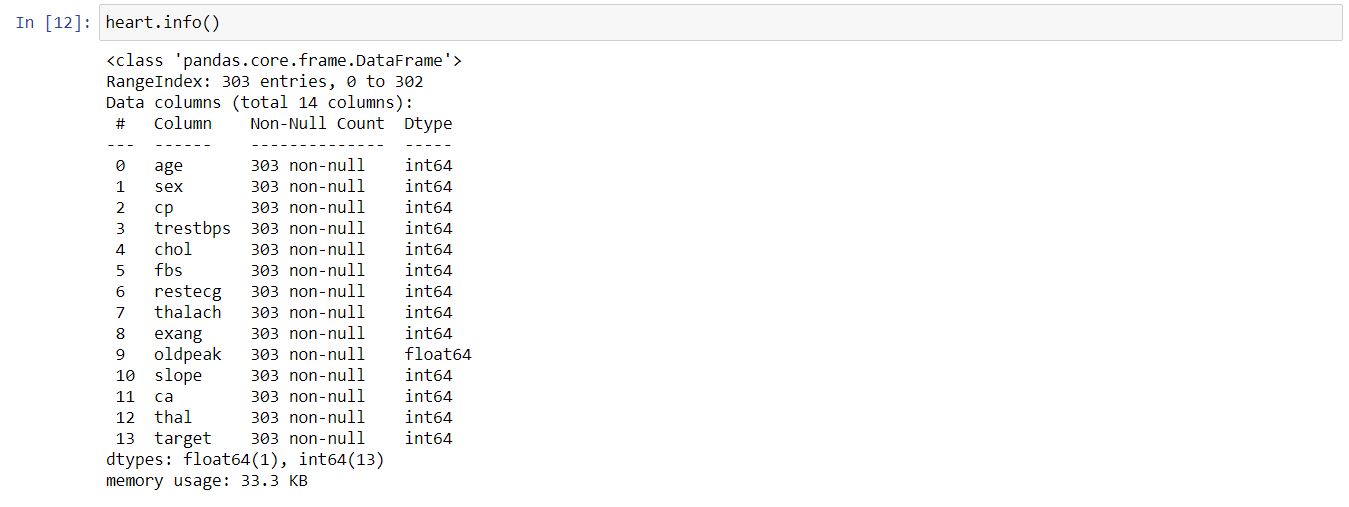


heart.describe() is used to describe the total count, mean ,min ,std ,25% ,50% ,75% ,maximum.



Which shows the result in a tabular form.

For information on data I used heart.info()



As we can see from the output above, there are a total of 13 features and 1 target variable. Also, there are no missing values so we don’t need to take care of any null values.

By using heart['target'].unique() it shows the array and data type of target.



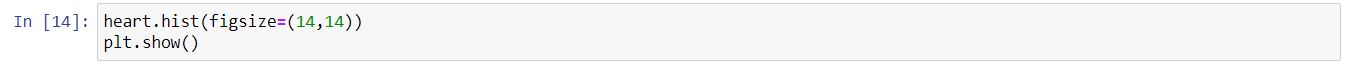
Which shows the output as array([1,0],dtype=int64).

Understanding the graphs

By using the

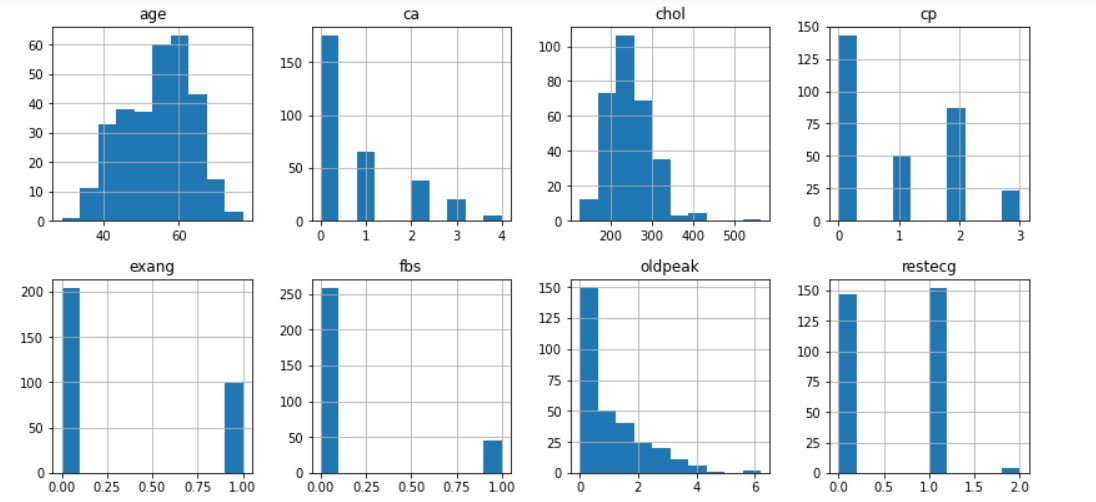
heart.hist(figsize=(14,14))

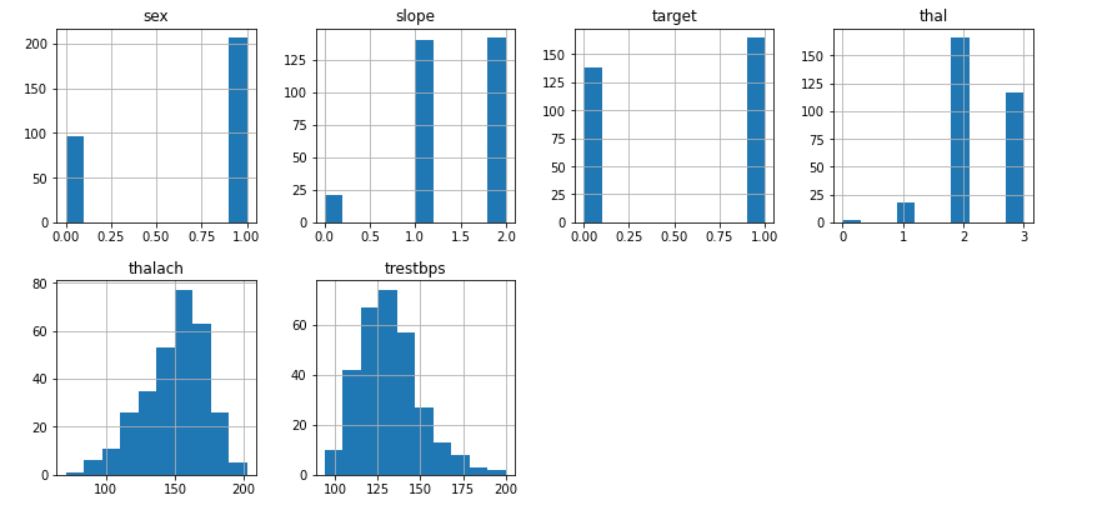
plt.show()



it shows the histogram,

It shows how each feature and label is distributed along different ranges, which further confirms the need for scaling. Next, wherever you see discrete bars, it basically means that each of these is actually a categorical variable. We will need to handle these categorical variables before applying Machine Learning. Our target labels have two classes, 0 for no disease and 1 for disease.

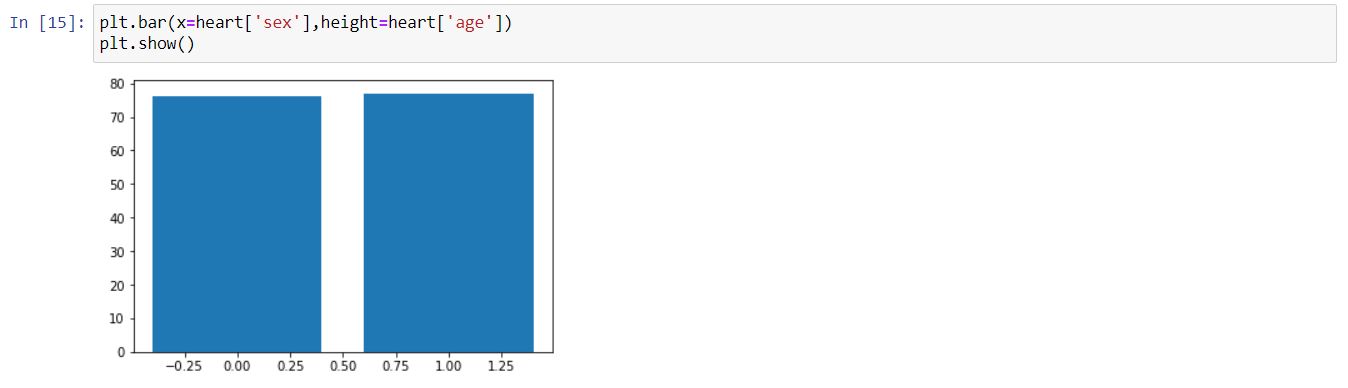




By using ,

plt.bar(x=heart['sex'],height=heart['age'])

plt.show()

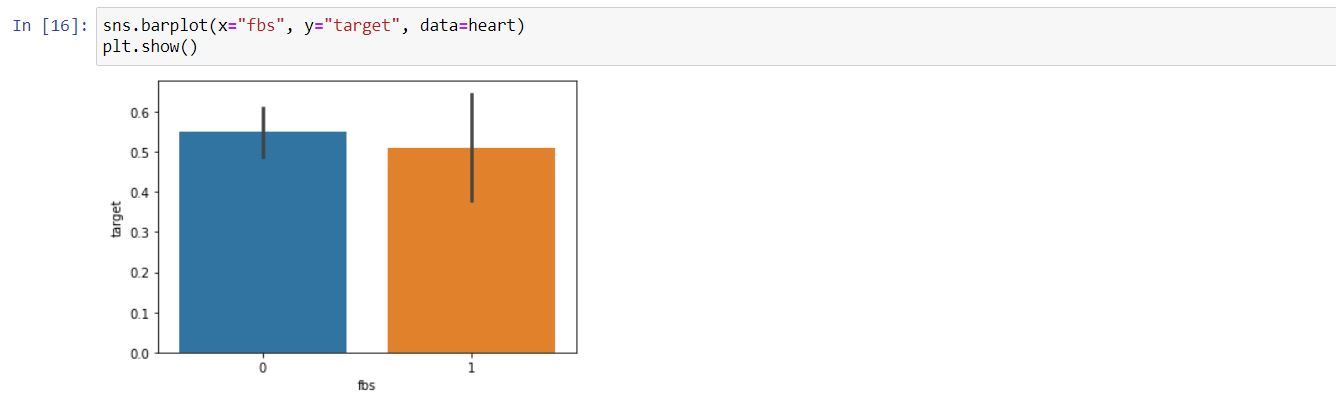


It shows the bar plot on which x axis has sex, and y axis has age.

By using,

sns.barplot(x="fbs", y="target", data=heart)

plt.show()



It shows the bar plot on which x axis has fbs, and y axis has target with respect to data=heart.

By using,

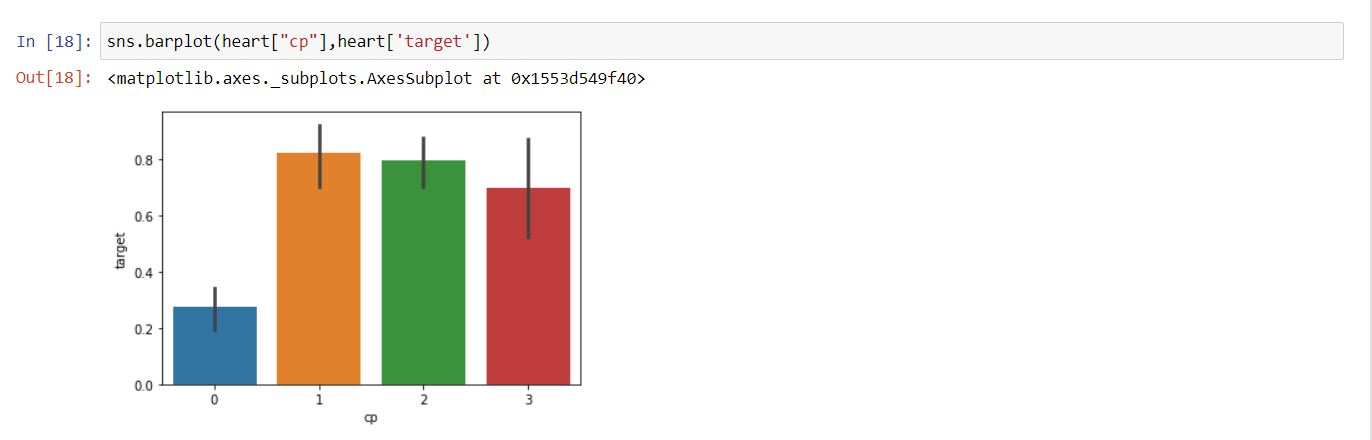
sns.barplot(x=heart['sex'],y=heart['age'],hue=heart['target'])



It shows the bar plot on which x axis has sex, and y axis has age with hue=heart[‘target’] which help us to tally data consequently.

By using,

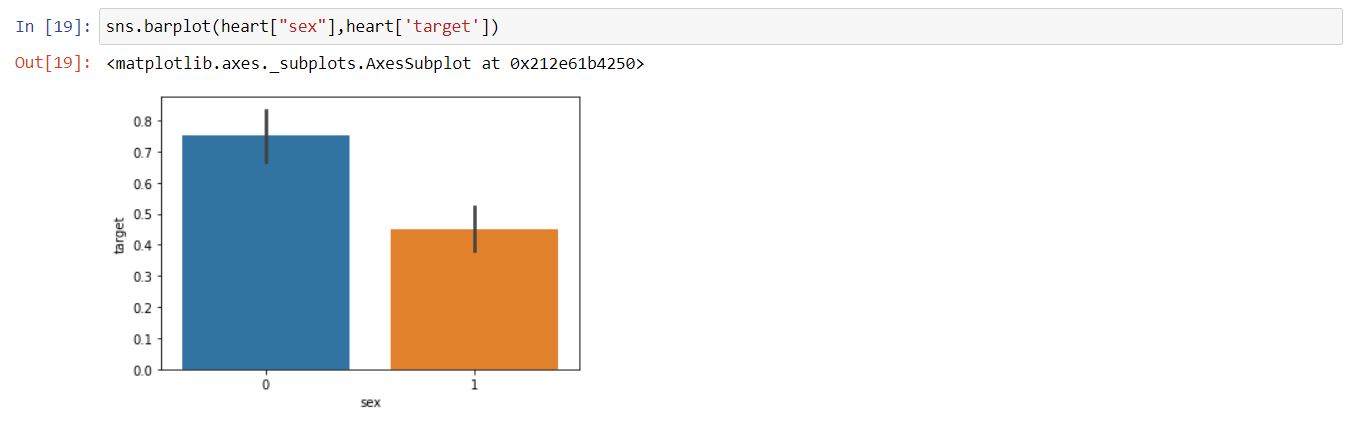
sns.barplot(heart["cp"],heart['target'])



It shows the bar plot on which x axis has cp, and y axis has target.

By using,

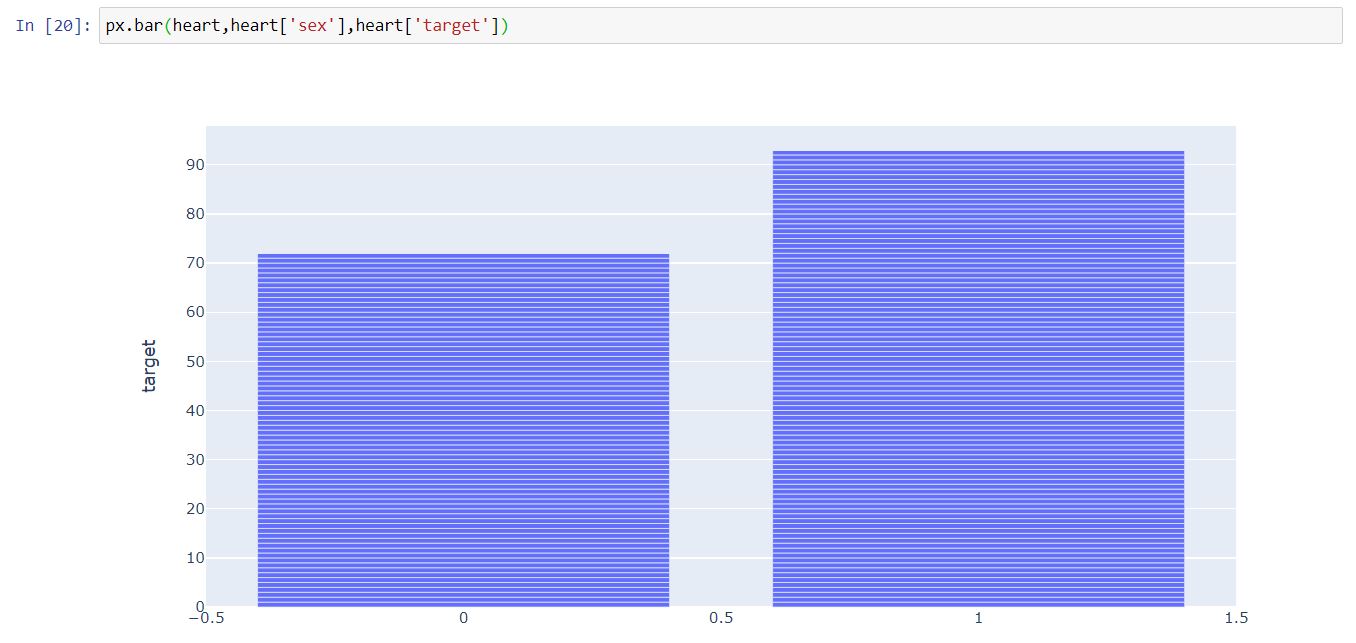
sns.barplot(heart["sex"],heart['target'])



It shows the bar plot on which x axis has sex, and y axis has target.

By using,

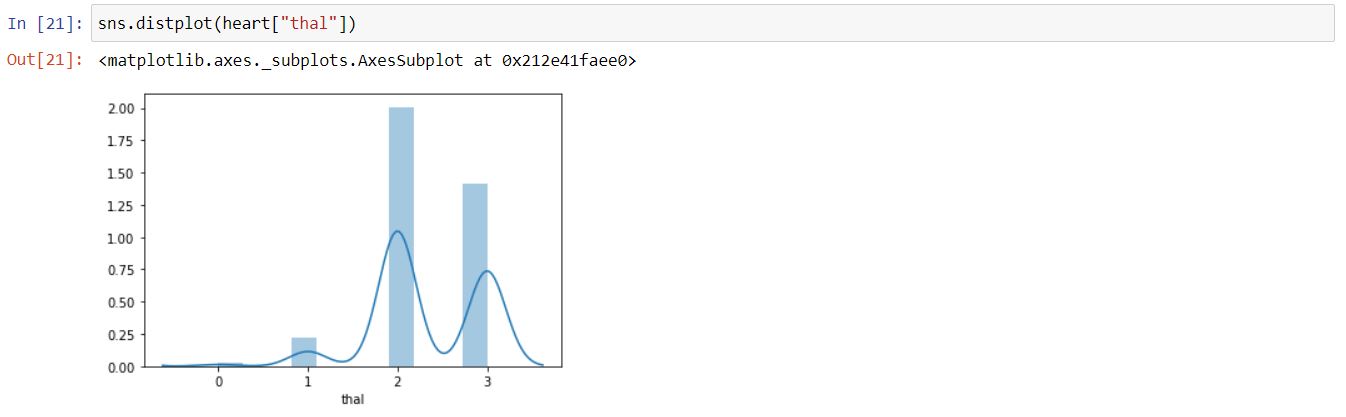
px.bar(heart,heart['sex'],heart['target'])



This is a special type of bar plot which let us know any data at any point on the graph by taking the specific target and scrolling the Corser at the point which let us know the information.

By using,

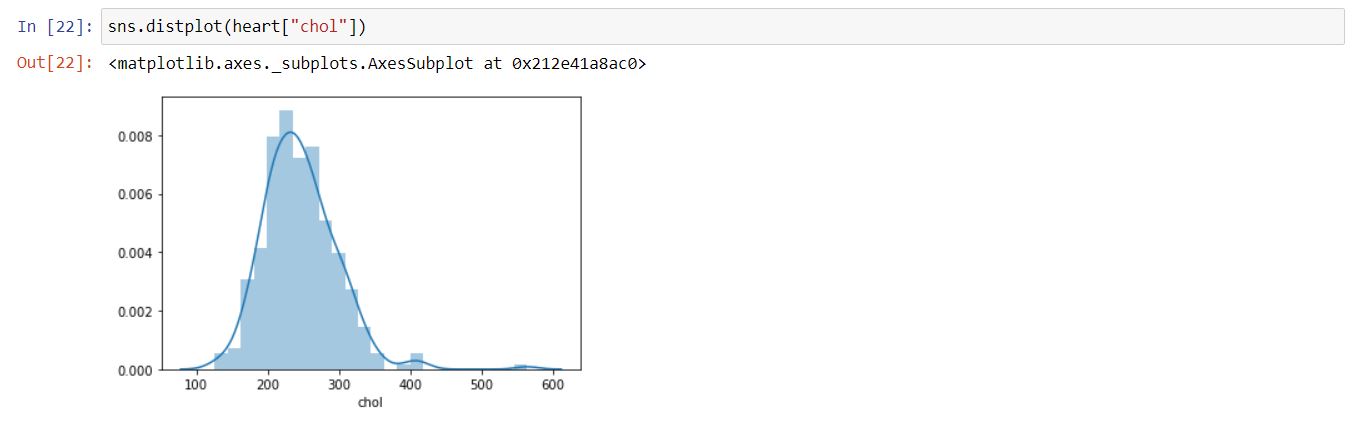
sns.distplot(heart["thal"])



This type of bar graph shows the information in bar form as well as in the wave form.

By using,

sns.distplot(heart["chol"])



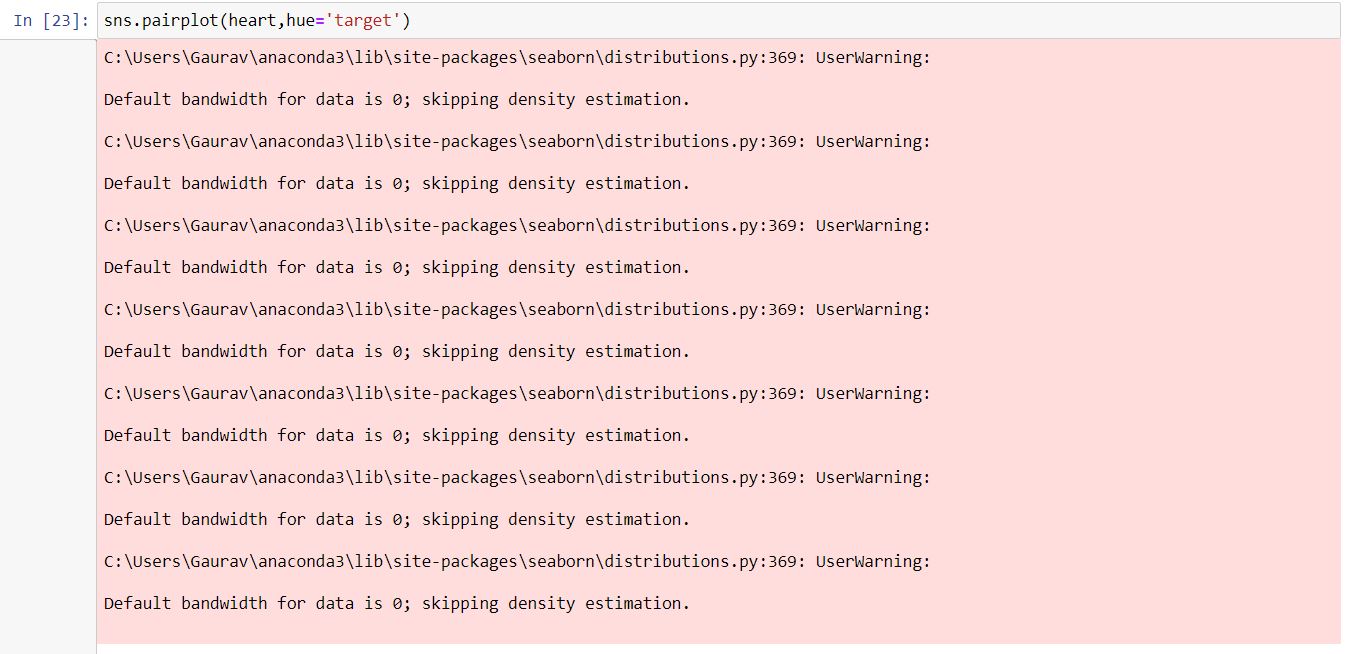
This type of bar graph shows the information in bar form as well as in the wave form.

In this graph we can see a specific pattern because of multiple quantity.

By using,

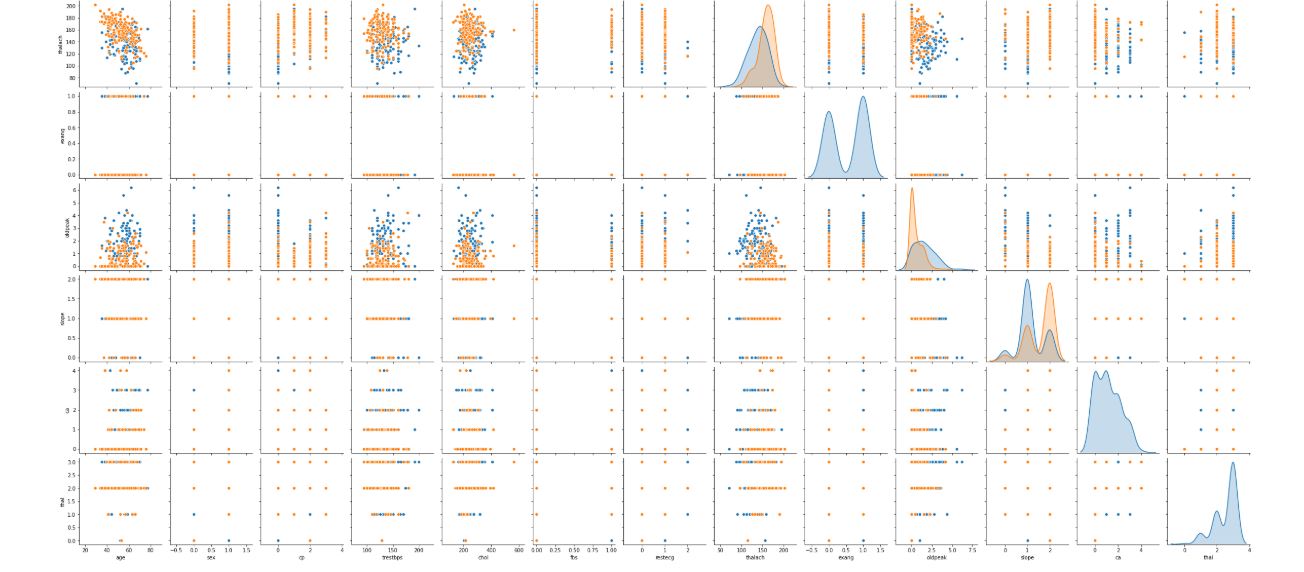
sns.pairplot(heart,hue='target')

this type of code shows multiple graphs at a single glance.



Because of multiple graphs it shows multiple warnings as well.



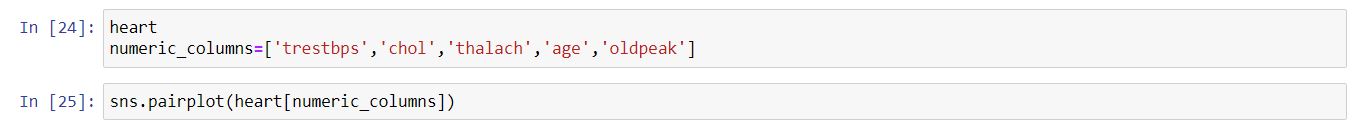


By using,

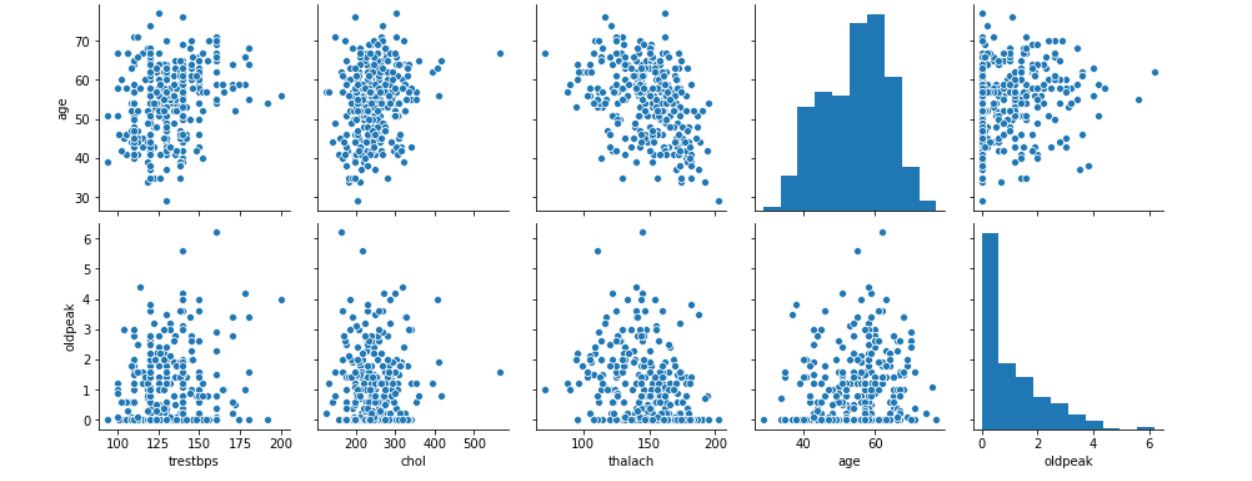
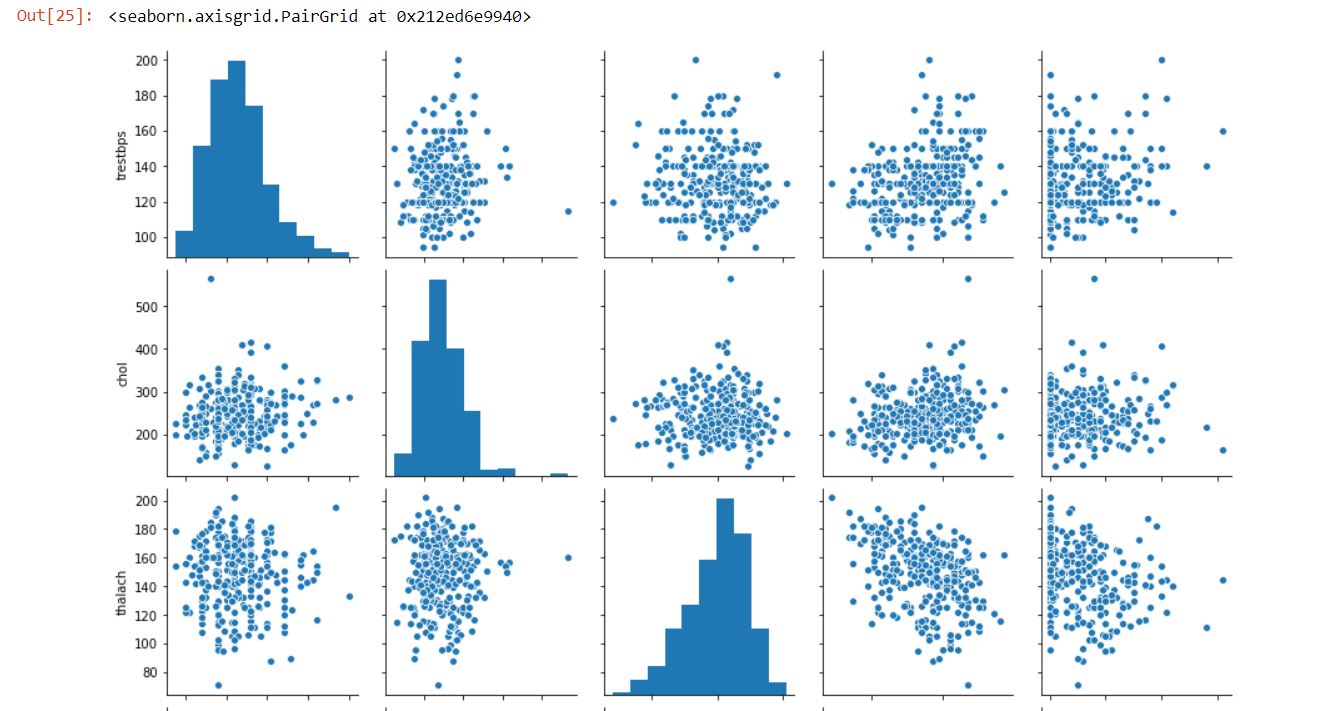
heart

numeric\_columns=['trestbps','chol','thalach','age','oldpeak']

sns.pairplot(heart[numeric\_columns])



it shows multiple program which tally the specific information fed to it in this case 'trestbps','chol','thalach','age','oldpeak',are tally with rest of the target.



We can see that trestbps is tally with others such as oldpeak, age, etc.

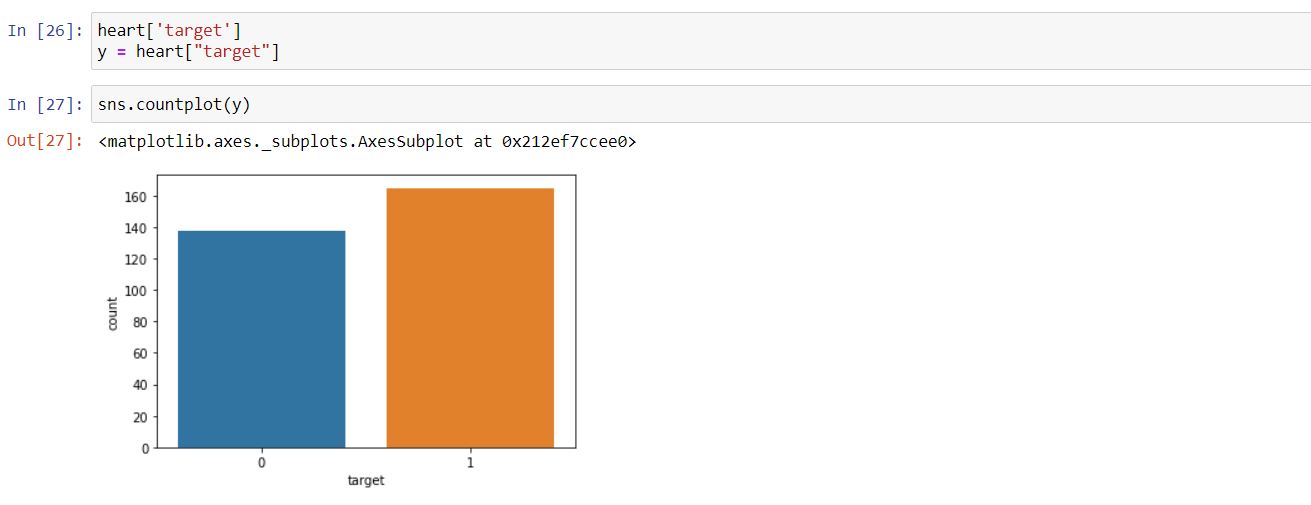
By using,

heart['target']

y = heart["target"]

sns.countplot(y)

this type of code also shows bar graph,with respect to count.



By using,

target\_temp = heart.target.value\_counts()

print(target\_temp)



The program gives the total count value with respect to target and also gives its data type.

Here the output shows,

1 165

0 138

Name: target, dtype: int64

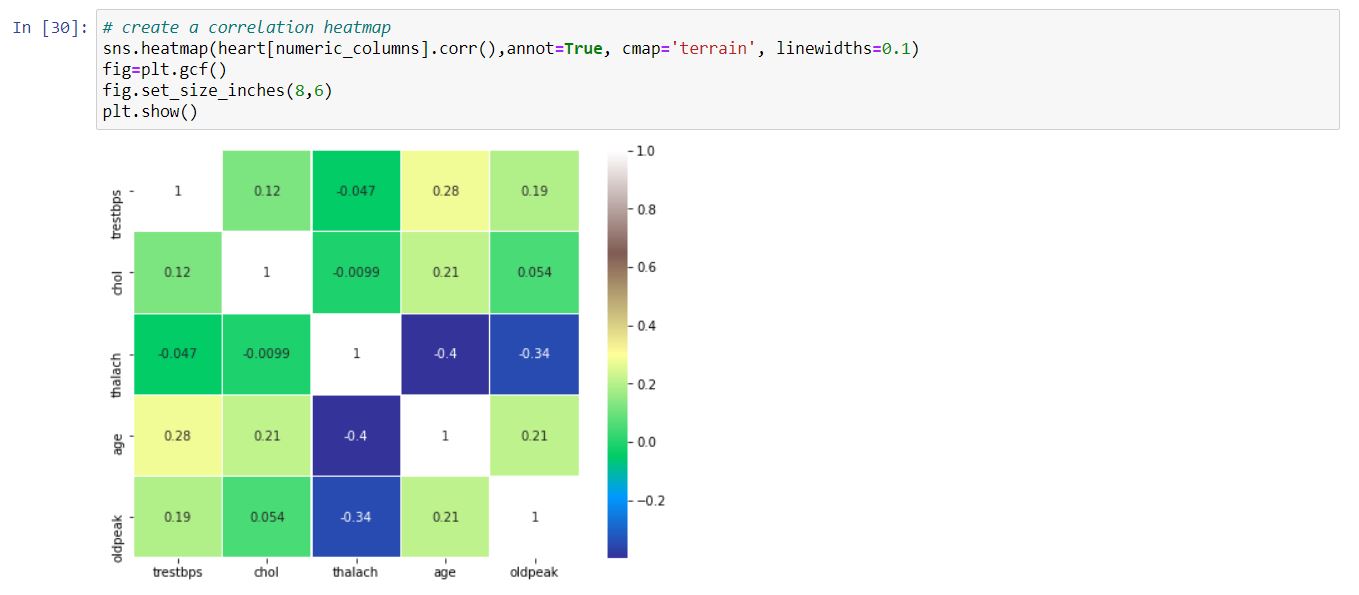
By creating the correlation heat map,

sns.heatmap(heart[numeric\_columns].corr(),annot=True, cmap='terrain', linewidths=0.1)

fig=plt.gcf()

fig.set\_size\_inches(8,6)

plt.show()



It’s easy to see that there is no single feature that has a very high correlation with our target value. Also, some of the features have a negative correlation with the target value and some have positive.

By creating four displot consequently,



plt.figure(figsize=(12,10))

plt.subplot(221)

sns.distplot(heart[heart['target']==0].age)

plt.title('Age of patients without heart disease')

plt.subplot(222)

sns.distplot(heart[heart['target']==1].age)

plt.title('Age of patients with heart disease')

plt.subplot(223)

sns.distplot(heart[heart['target']==0].thalach )

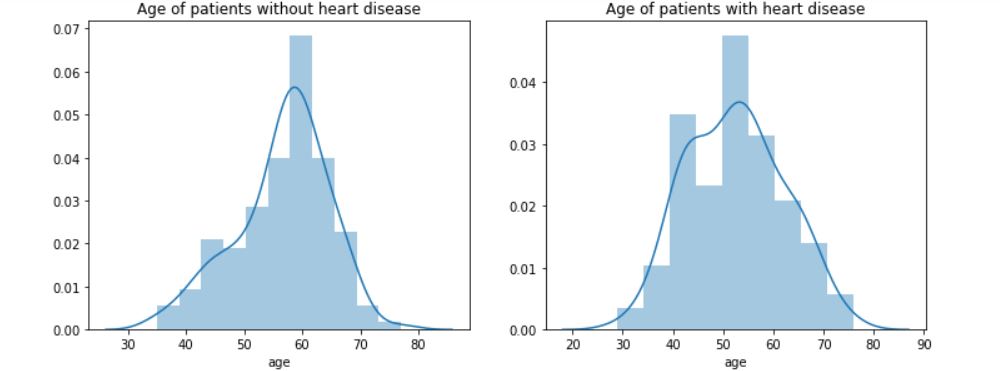
plt.title('Max heart rate of patients without heart disease')

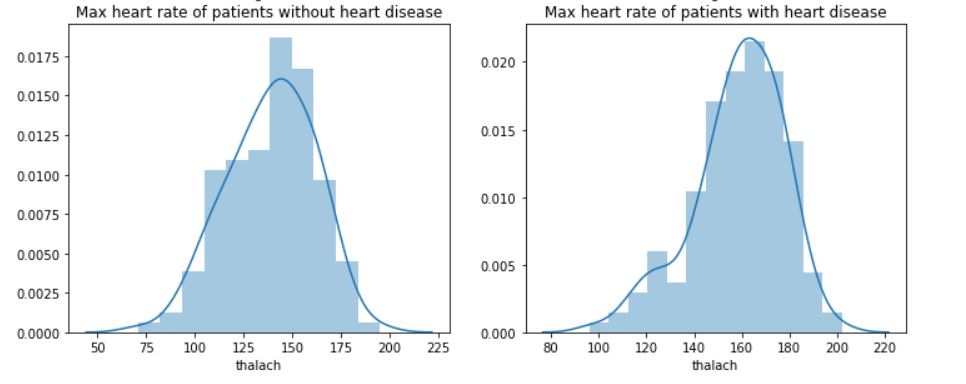
plt.subplot(224)

sns.distplot(heart[heart['target']==1].thalach )

plt.title('Max heart rate of patients with heart disease')

plt.show()

this type of code gives each an every graph a specific changes given by me to different graphs. 



By using,

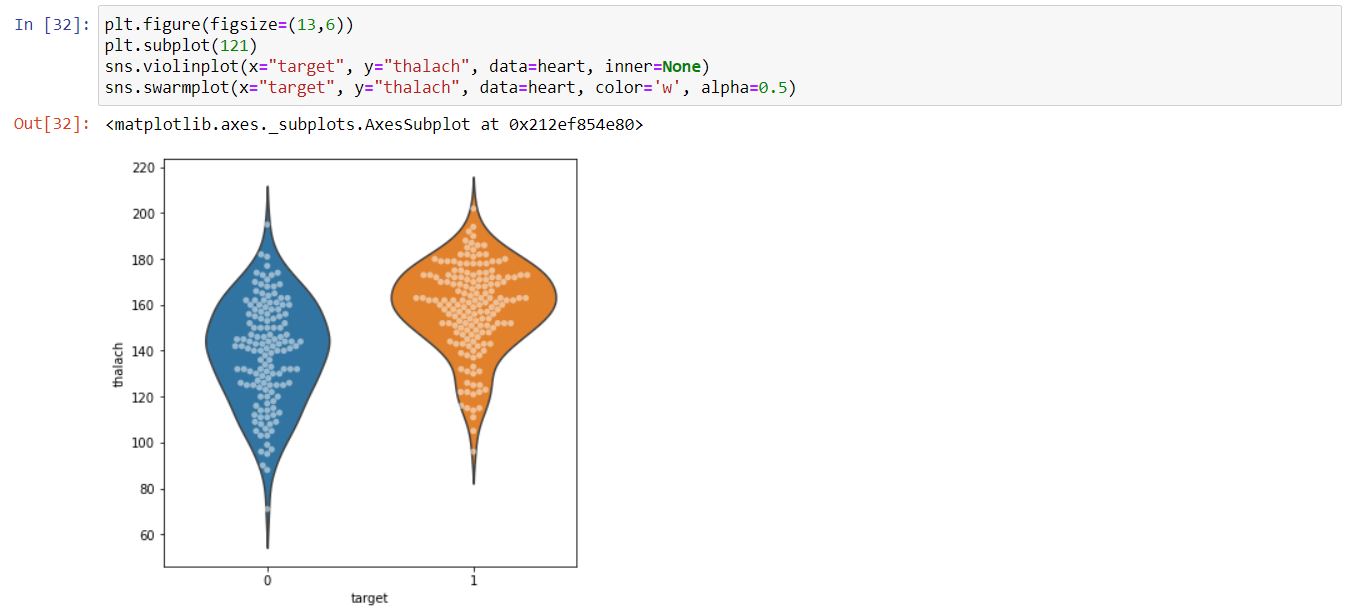
plt.figure(figsize=(13,6))

plt.subplot(121)

sns.violinplot(x="target", y="thalach", data=heart, inner=None)

sns.swarmplot(x="target", y="thalach", data=heart, color='w', alpha=0.5)

gives the a graph a figure ,shape which helps to analysis the graph more accurately.



As we can see that the graph has been plot between target and thalach because of which the graph changes its shape and size according information.

By removing the shape texture it shows the specific amount of data without any regression.

By using,

plt.subplot(122)

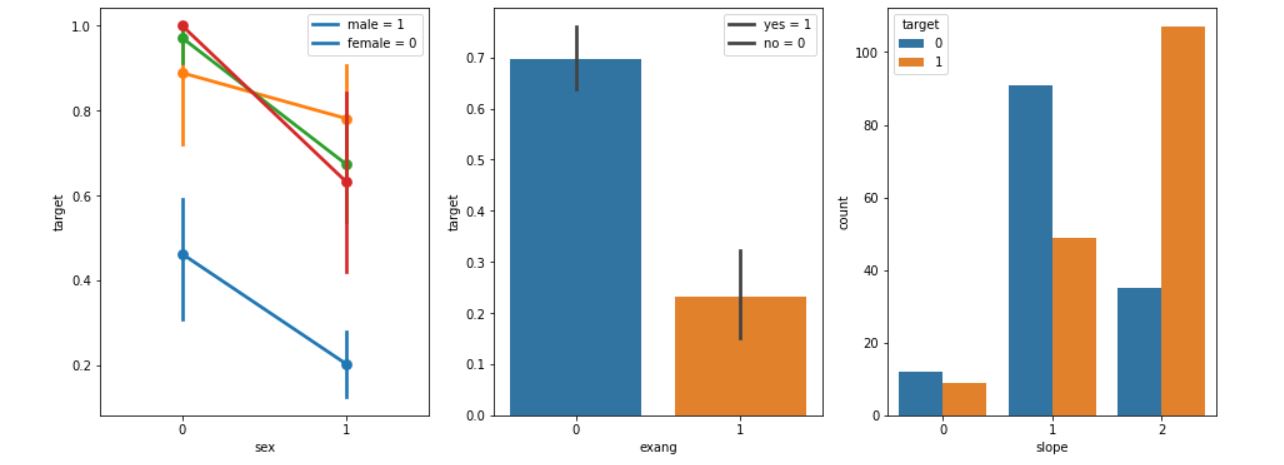
sns.swarmplot(x="target", y="thalach", data=heart)

plt.show()



By using pairplot, and two barplot,

Using this code tell us that it has some variations such as 4-5 line tell us that the code is specifically return for pair plot and rest of it return for barplot.



Code:

plt.figure(figsize=(16,6))

plt.subplot(131)

sns.pointplot(x="sex", y="target", hue='cp', data=heart)

plt.legend(['male = 1', 'female = 0'])

plt.subplot(132)

sns.barplot(x="exang", y="target", data=heart)

plt.legend(['yes = 1', 'no = 0'])

plt.subplot(133)

sns.countplot(x="slope", hue='target', data=heart)

plt.show()