Heart Disease

PROJECT 1

By Faiz Khan

First We Have To Import Libraries That We Are Going To Use In Our Project

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings("ignore")
```

Now we have to load DataSet

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

```
In [3]: df
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

*cp- It is a condition that cause diastolic heart failure and can be potentially curable *trestbps-Also known as resting blood pressure,is a factor that can be used to predict heart disease risk *chol-cholesterol *fbs-Fasting blood sugar *restecg-resting electrocardiograpic measurement(0=normal,1=having st-t,2=showing probable or definate left ventricular hypertrophy) *thalch-Person's maximun hearthrate *exang=Excercise included angina(1=yes,0=no) *oldpeak-is a number that measures ST depression caused by excercise relative to rest on an electrocardiogram *slope=slope of st segment/heartrate *ca=calcium in your hear artries *thal-Thalassemia is an inherited disorder that can cause heart problems

Applying Some Python Methods

```
In [4]: df.describe()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.00
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.75
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.03
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.00
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.00
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.00
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.00
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.00

```
In [5]: 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  1025 non-null   int64
 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
```

Duplicate values and Value Counts

```
In [6]: df.duplicated()
```

```
Out[6]: 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 [7]: df[1020:1025]
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

```
In [8]: df.target.value_counts()
```

```
Out[8]: 1    526
        0    499
```

click to expand output; double click to hide output

```
In [9]: df.sex.value_counts()
```

```
Out[9]: 1    713
        0    312
        Name: sex, dtype: int64
```

```
In [10]: df.age.value_counts()
```

```
Out[10]: 58    68
         57    57
         54    53
         59    46
         52    43
         51    39
         56    39
         62    37
         60    37
         44    36
```

```
Name: age, dtype: int64
```

```
In [11]: df.groupby(["age","sex"])["target"].unique()
```

```
Out[11]: age  sex
         29   1         [1]
         34   0         [1]
              1         [1]
         35   0         [1]
              1      [0, 1]
                      ...
         70   1      [0, 1]
         71   0         [1]
         74   0         [1]
         76   0         [1]
         77   1         [0]
         Name: target, Length: 73, dtype: object
```

```
In [12]: df.oldpeak.value_counts()
```

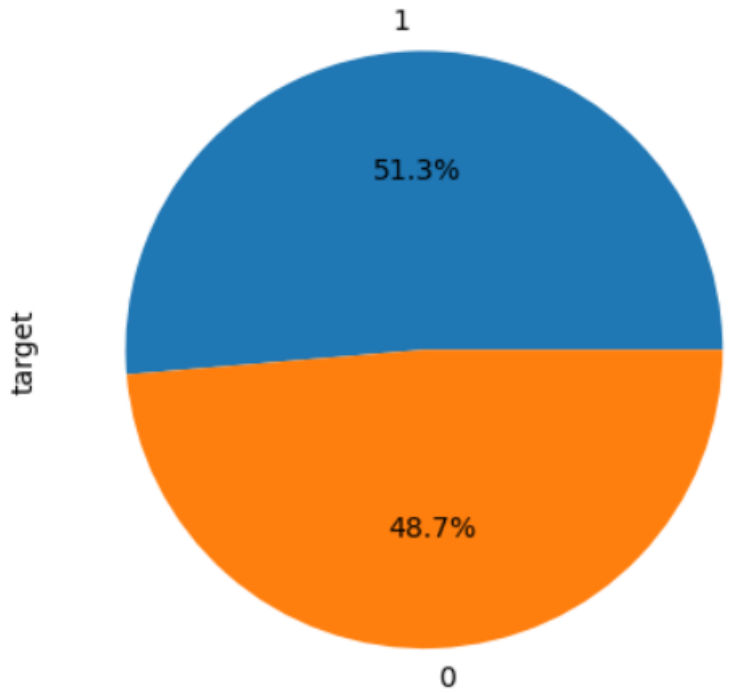
```
Out[12]: 0.0    329
         1.2     58
         1.0     51
         0.6     47
         0.8     44
         1.4     44
         1.6     37
```

```
In [13]: df.restecg.value_counts()
```

```
Out[13]: 1    513
         0    497
         2     15
         Name: restecg, dtype: int64
```

Data Visualisation With Python

```
In [14]: df.target.value_counts().plot(kind="pie",autopct="%1.1f%%")
Out[14]: <Axes: ylabel='target'>
```

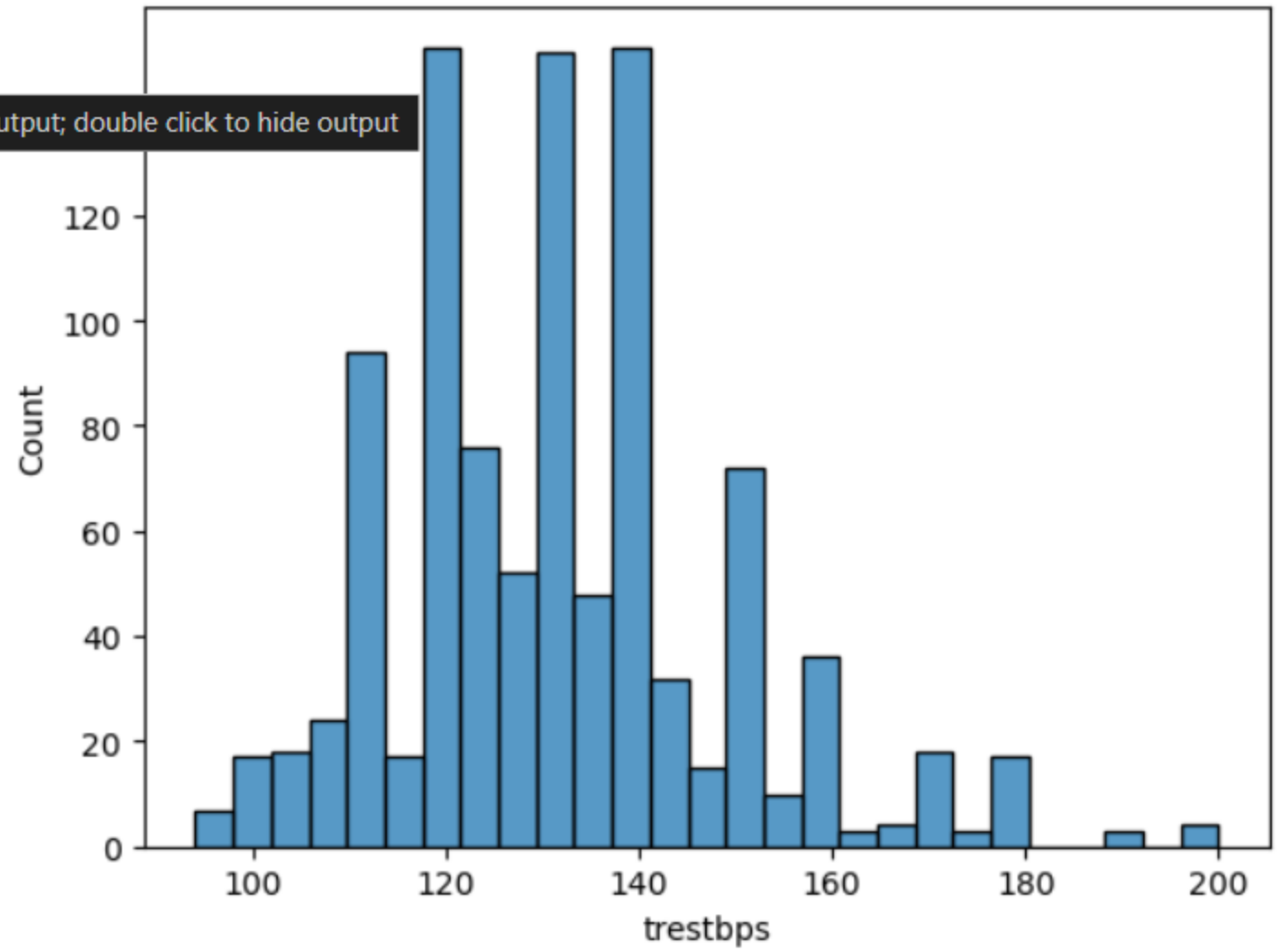



```
In [15]: df
```

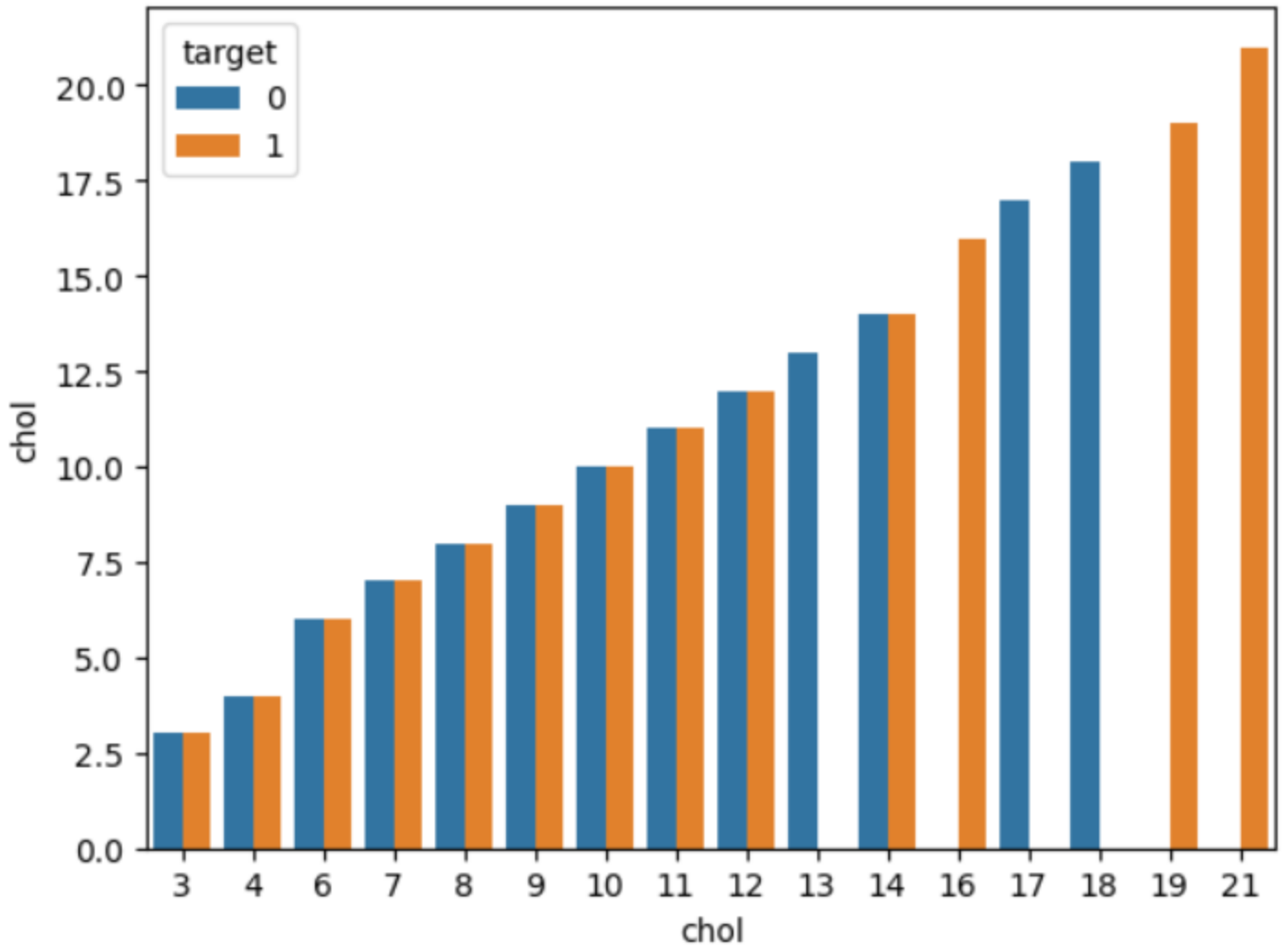
```
[16]: sns.histplot(x=df.trestbps)
```

```
t[16]: <Axes: xlabel='trestbps', ylabel='Count'>
```

o expand output; double click to hide output

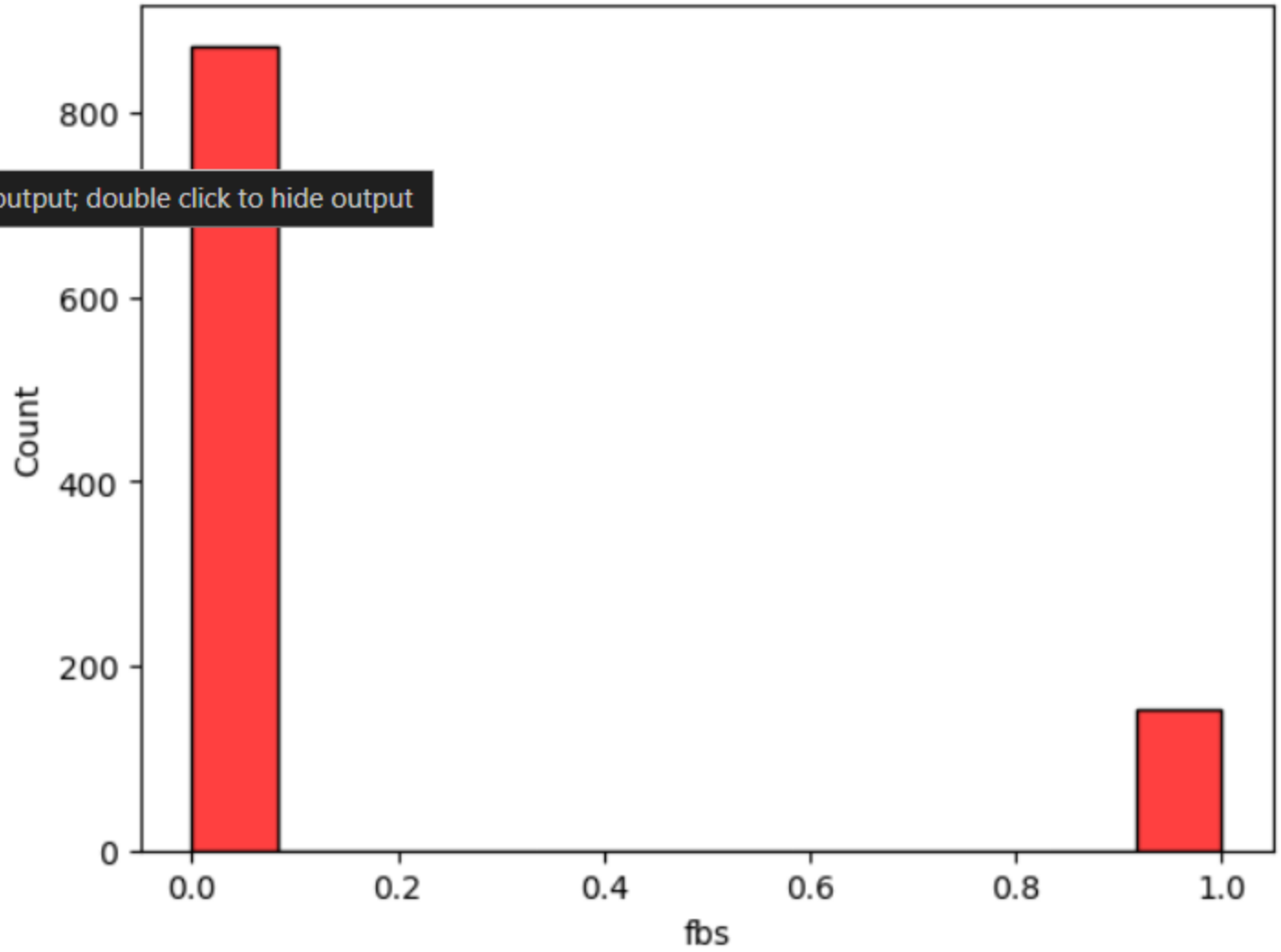


```
In [17]: sns.barplot(x=df.chol.value_counts(),y=df.chol.value_counts(),hue=df.target)
Out[17]: <Axes: xlabel='chol', ylabel='chol'>
```



In [18]:
```
sns.histplot(df.fbs,color="red")
```

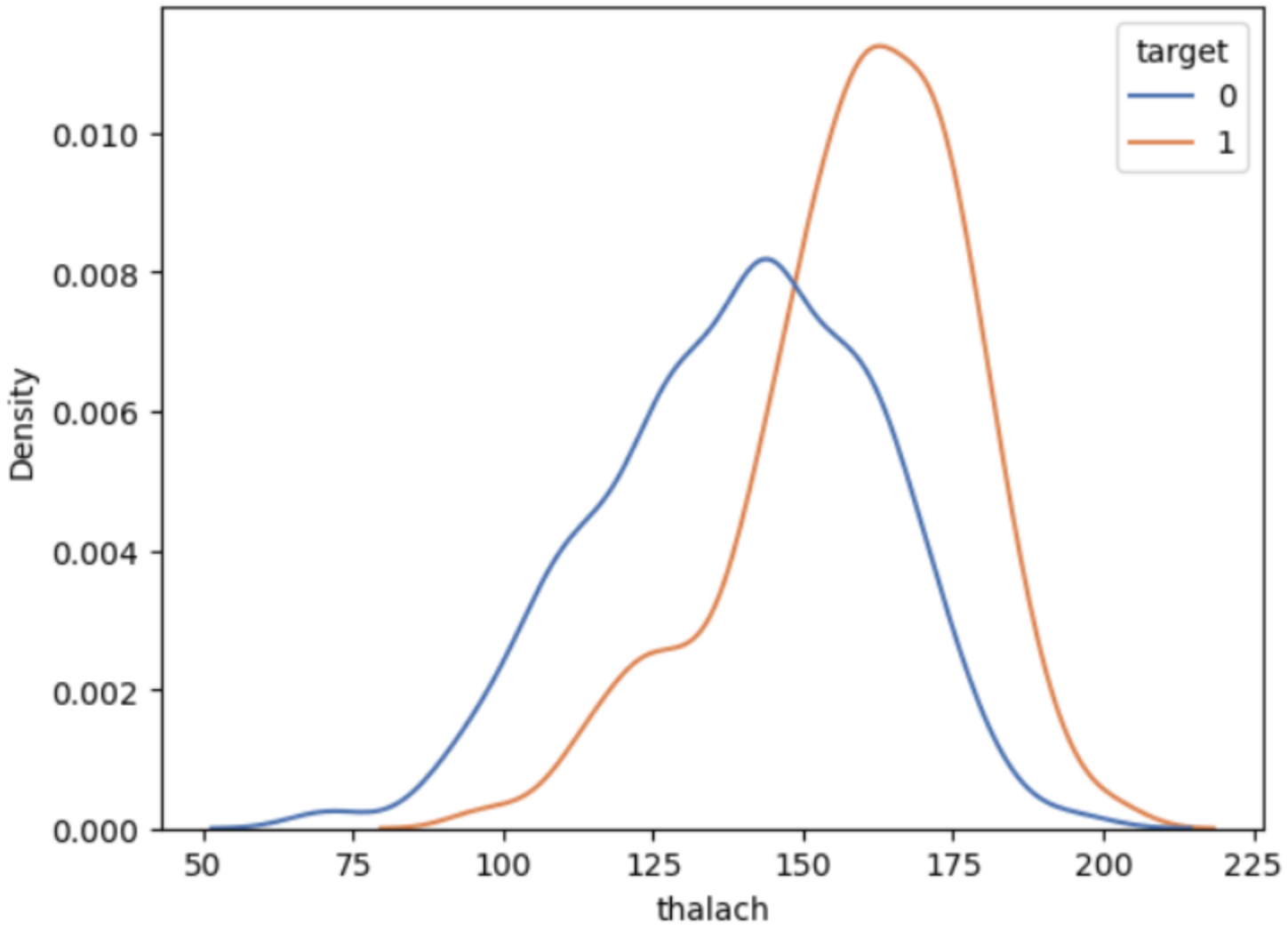
Out[18]:
```
<Axes: xlabel='fbs', ylabel='Count'>
```



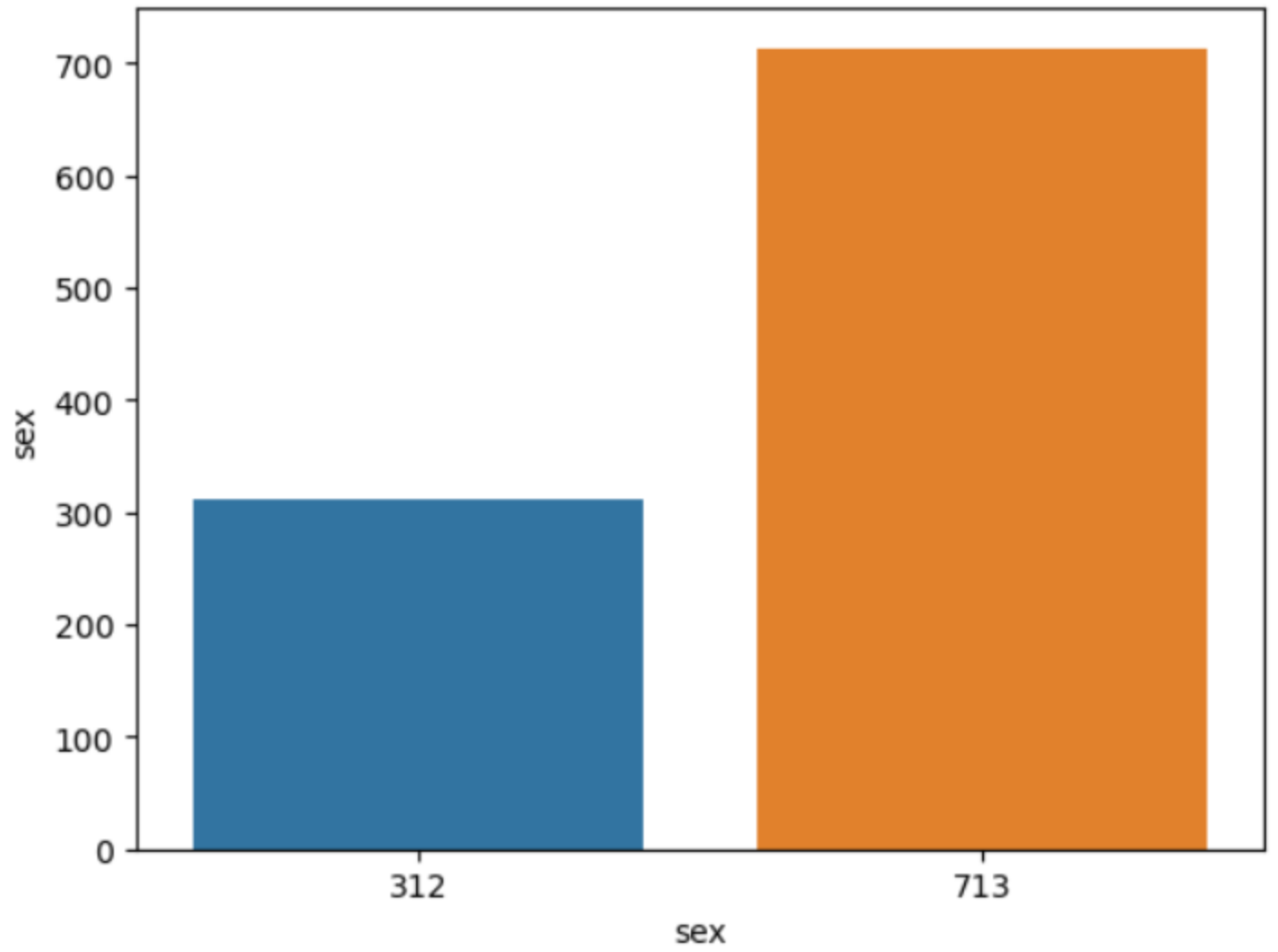
click to expand output; double click to hide output

In [19]:
```
sns.kdeplot(x=df.thalach,hue=df.target,palette="deep")
```

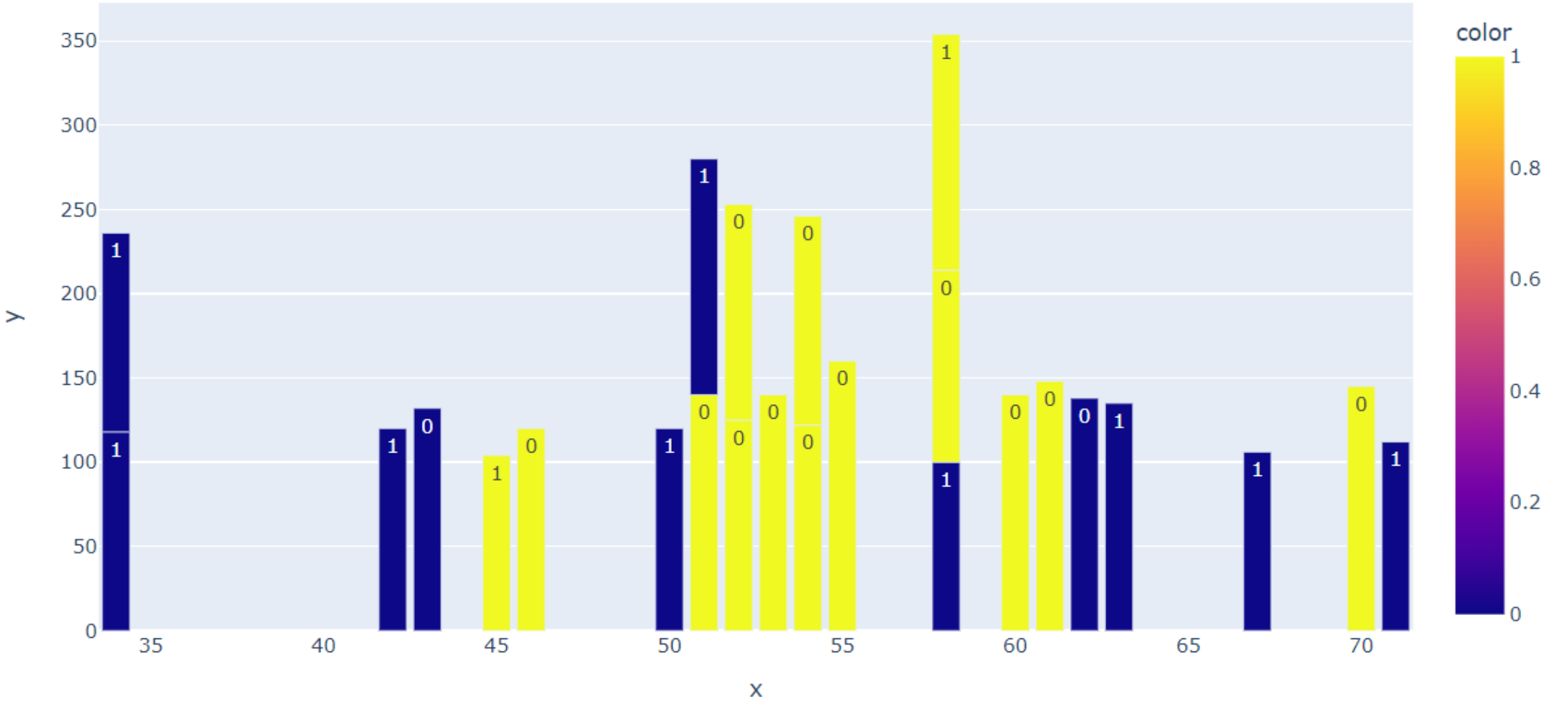
Out[19]: <Axes: xlabel='thalach', ylabel='Density'>



```
In [20]: sns.barplot(x=df.sex.value_counts(),y=df.sex.value_counts())
Out[20]: <Axes: xlabel='sex', ylabel='sex'>
```



In [4]:
```
px.bar(x=df.age.head(25),y=df.trestbps.head(25),color=df.sex.head(25),text=df.target.head(25))
```



Now we have to apply Ml Algorithm

```
In [24]: x=df.drop("target",axis=1)

In [25]: y=df.target

In [27]: from sklearn.model_selection import train_test_split

In [28]: x_test,x_train,y_test,y_train=train_test_split(x,y,test_size=0.25,random_state=44)

In [29]: x_test.shape,x_train.shape
Out[29]: ((768, 13), (257, 13))

In [30]: from sklearn.preprocessing import OrdinalEncoder

In [31]: oe=OrdinalEncoder()

In [32]: oe
Out[32]: OrdinalEncoder()
         In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
         On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [33]: x_test[["trestbps","chol","thalach"]]=oe.fit_transform(x_test[["trestbps","chol","thalach"]])

In [34]: x_train[["trestbps","chol","thalach"]]=oe.fit_transform(x_train[["trestbps","chol","thalach"]])
```

In [35]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [36]:
```
rfc=RandomForestClassifier()
```

In [37]:
```
rfc
```

Out[37]:
```
RandomForestClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [38]:
```
rfc.fit(x_train,y_train)
```

Out[38]:
```
RandomForestClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [39]:
```
y_pred=rfc.predict(x_test)
```

In [41]:
```
from sklearn.metrics import accuracy_score
```

In [42]:
```
accuracy_score(y_pred,y_test).round(2)
```

Out[42]:
```
0.92
```

Tableau'Dashboard of Heart Disease Dataset

Tableau Public - Project 1

File Data Worksheet Dashboard Story Analysis Map Format Server Window Help

Entire View | Show Me

Dashboard | Layout

Default
Phone
Device Preview

Size
Fit to height: width: 1320

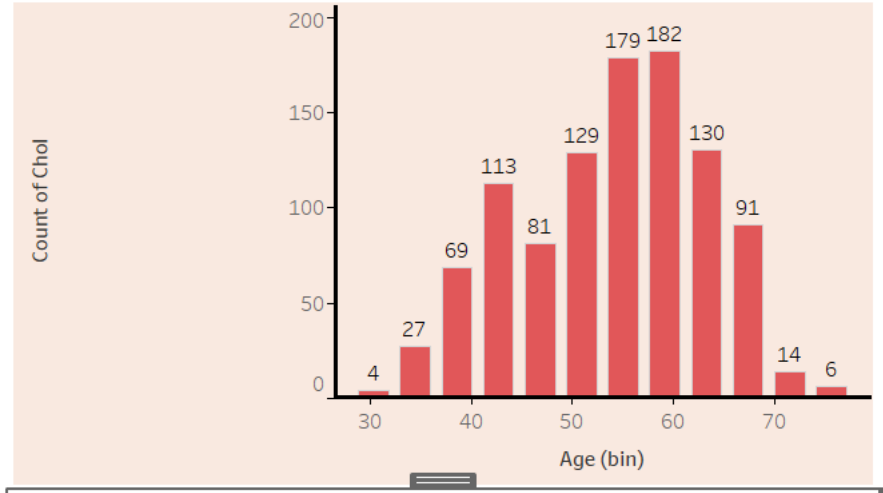
Sheets
- Sheet 1
- Sheet 2
- Sheet 3
- Sheet 4

Objects
- Horizontal Container
- Vertical Container
- Text
- Extension
- Ask Data
- Data Story
- Image

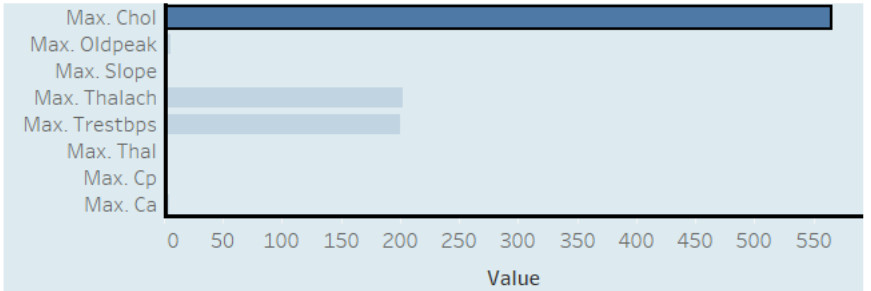
Tiled | Floating

Show dashboard title

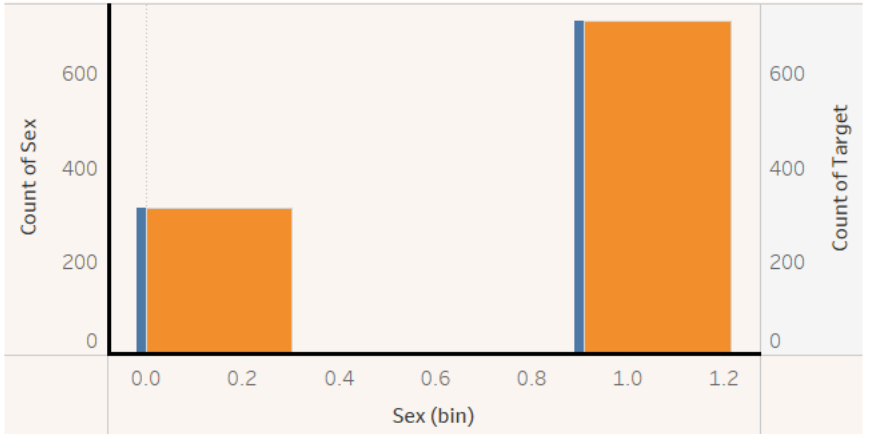
Cholesterol Level According To Diffrent Age Groups



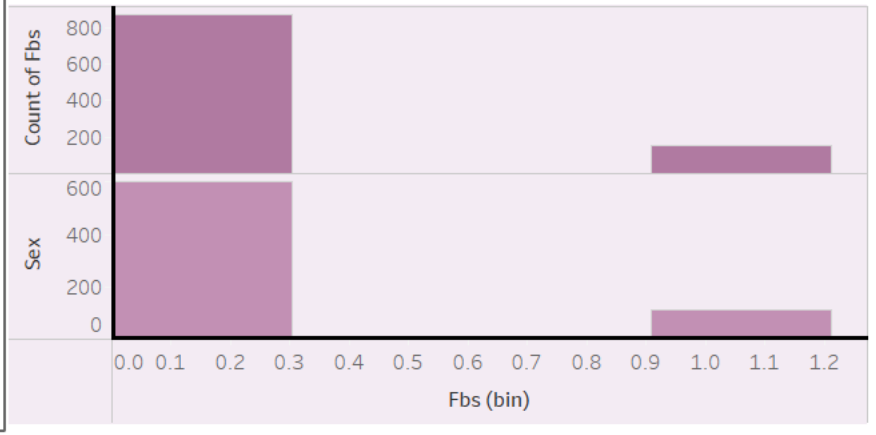
All Symthoms with their Maximum numbers



Number Of people Having Heart Disease According To Gender



Number of People who has Heart Disease But Not A Diabetes



Data Source | Sheet 1 | Sheet 2 | Sheet 3 | Sheet 4 | Dashboard 1

4 marks | 1 row by 1 column | SUM of CNT(Sex): 1,025 | Faiz Khan

Import Points

- People who have high cholesterol can have more chances of having Heart Disease
- People who have Diabetes are mostly a Heart patient
- Men category have more heart diseases than women
- Age group from 45-65 suffers most from heart disease and also most of them have high cholesterol and also they are suffering from diabetes