

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [3]: data = pd.read_csv('Heart Disease.csv')
data.head(10)

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
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0

age:sex:chest pain type (4 values) value 0:typical angina value 1:atypical angina value 2:non-anginal pain value 3:asymptomatic trestbps: resting blood pressure(in mm hg on admission to the hospital) chol:serum cholestrol in mg/dl fbs: (fasting blood sugar>120 mg/dl)(1 = true;0 = false) restecg:resting eletrocardiographic results value 0:normal value 1:having ST-T wave abnormality(T wave inversions and/or ST elevation or depression of >0.05mV) value 2:showing probable or definite left ventricular hypertrophy by Estes criteria thalach:maximum heart rate achieved exercise induced angina(1=yes;0 = no) oldpeak =ST depression induced by exercise ST segment value 1:upslloping value 2:flat value 3:downslloping ca:number of major vessel(0-3)colored by fluoroscopy thal: 3 = normal;6 = fixed defect;7 = reversible defect target: 0=less chance of heart attack, 1=more chance of heart attack

last 5 rows

```
In [4]: data.tail()
```

```
Out[4]:
```

	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

Find shape of our Dataset(Number of Rows and Columns)

```
In [5]: print("Number of Rows",data.shape[0])
print("Number of Columns",data.shape[1])

Number of Rows 1025
Number of Columns 14
```

Get information about our Dataset like Toatal Number Row,Total Number of columns,Datatype of Each column And memory Requirement

```
In [6]: data.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
 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(1), int64(13)
memory usage: 112.2 KB
```

Check Null values in the Dataset

```
In [8]: data.isnull().sum()
```

```
Out[8]:
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
```

check for Duplicate Date and drop them

```
In [9]: data_dup = data.duplicated().any()
print(data_dup)

True

In [10]: data = data.drop_duplicates()

In [12]: data.shape

Out[12]:
(302, 14)
```

Get overall Statistics about the Dataset

```
In [13]: data.describe()
```

```
Out[13]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	302.00000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000
mean	54.42053	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.569536	0.327815	1.043046	1.397351	0.718543	2.314570	0.543046
std	9.04797	0.466426	1.032044	17.563394	51.753489	0.356686	0.528027	22.903527	0.470196	1.161452	0.616274	1.006748	0.613026	0.498970
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	45.000000	0.000000	1.000000	120.000000	211.000000	0.000000	0.000000	133.250000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.500000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.500000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.750000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

