

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
df=pd.read_csv('/content/Heart_Disease_Prediction.csv')
df
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Presence

```
df.head()
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence

```
df.tail()
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence

```
df.shape
```

```
(270, 14)
```

```
df.columns
```

```
Index(['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium', 'Heart Disease'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Age              270 non-null    int64  
 1   Sex              270 non-null    int64  
 2   Chest pain type 270 non-null    int64  
 3   BP               270 non-null    int64  
 4   Cholesterol      270 non-null    int64  
 5   FBS over 120     270 non-null    int64  
 6   EKG results      270 non-null    int64  
 7   Max HR           270 non-null    int64  
 8   Exercise angina  270 non-null    int64  
 9   ST depression    270 non-null    float64 
 10  Slope of ST      270 non-null    int64  
 11  Number of vessels fluro 270 non-null    int64  
 12  Thallium          270 non-null    int64  
 13  Heart Disease    270 non-null    object  
dtypes: float64(1), int64(12), object(1)
```

```
df.dtypes
```

	0
Age	int64
Sex	int64
Chest pain type	int64
BP	int64
Cholesterol	int64
FBS over 120	int64
EKG results	int64
Max HR	int64
Exercise angina	int64
ST depression	float64
Slope of ST	int64
Number of vessels fluro	int64
Thallium	int64
Heart Disease	object

```
dtype: object
```

```
numerical_features = df.select_dtypes(include=['int64','float64'])
numerical_features.columns
```

```
Index(['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium'],
      dtype='object')
```

```
categorical_features = df.select_dtypes(include=['object'])
categorical_features.columns
```

```
Index(['Heart Disease'], dtype='object')
```

```
df.isnull().sum() #check column wise for null value
```

	0
<b>Age</b>	0
<b>Sex</b>	0
<b>Chest pain type</b>	0
<b>BP</b>	0
<b>Cholesterol</b>	0
<b>FBS over 120</b>	0
<b>EKG results</b>	0
<b>Max HR</b>	0
<b>Exercise angina</b>	0
<b>ST depression</b>	0
<b>Slope of ST</b>	0
<b>Number of vessels fluro</b>	0
<b>Thallium</b>	0
<b>Heart Disease</b>	0

**dtype:** int64

```
df[df.isnull().any(axis=1)] #row wise check any null value
```

Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels	Thallium	Heart Disease
-----	-----	-----------------	----	-------------	--------------	-------------	--------	-----------------	---------------	-------------	-------------------	----------	---------------

```
df.describe() #avg statistical summary of each column
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels	Thallium	Heart Disease
<b>count</b>	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	
<b>mean</b>	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	1.050000	0.470952	1.145214	0.000000	0.000000
<b>std</b>	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
<b>min</b>	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
<b>25%</b>	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
<b>50%</b>	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.800000
<b>75%</b>	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	1.600000	0.000000	0.000000	0.000000	0.000000
<b>77.000000</b>	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

```
age_array = df['Age'].to_numpy() #to convert a column into an array
age_array
```

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

```
age_array_2d = df['Age'].to_numpy().reshape(-1,1) #to convert a column into a 2d array
age_array_2d
```

```
array([[70],  
       [67],  
       [57],  
       [64],  
       [74],  
       [65],  
       [56],  
       [59],  
       [60],  
       [63],  
       [59],  
       [53],  
       [44],  
       [61],  
       [57],  
       [71],  
       [46],  
       [53],  
       [64],  
       [40],  
       [67],  
       [48],  
       [43],  
       [47],  
       [54],  
       [48],  
       [46],  
       [51],  
       [58],  
       [71],  
       [57],  
       [66],  
       [37],  
       [59],  
       [50],  
       [48],  
       [61],  
       [59],  
       [42],  
       [48],  
       [40],  
       [62],  
       [44],  
       [46],  
       [59],  
       [58],  
       [49],  
       [44],  
       [66],  
       [65],  
       [42],  
       [52],  
       [65],  
       [63],  
       [45],  
       [41],  
       [61],  
       [60],
```

```
mean_age = np.mean(age_array_2d)  
median_age = np.median(age_array)  
std_age= np.std(age_array)  
print(mean_age,',',median_age,',',std_age)
```

```
54.4333333333333 , 55.0 , 9.092182234083177
```

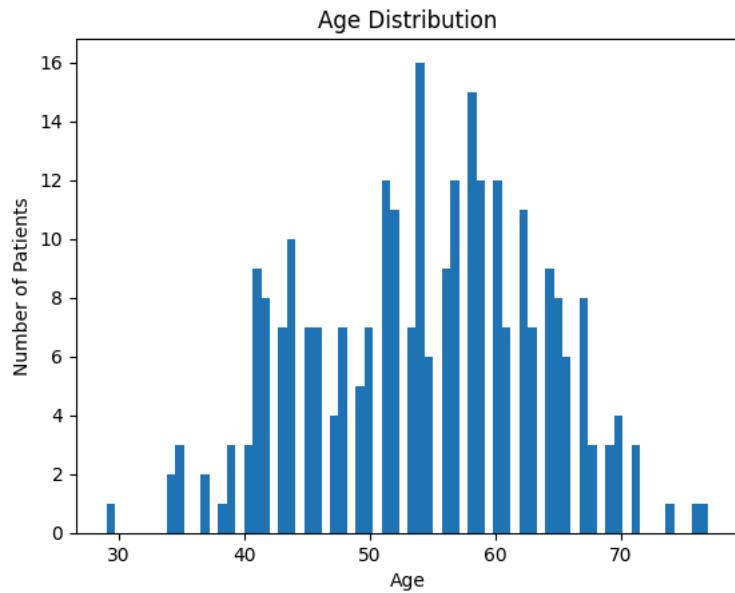
```
min_cholesterol = np.min(df['Cholesterol'])  
max_cholesterol = np.max(df['Cholesterol'])  
print(min_cholesterol)  
print(max_cholesterol)
```

```
126  
564
```

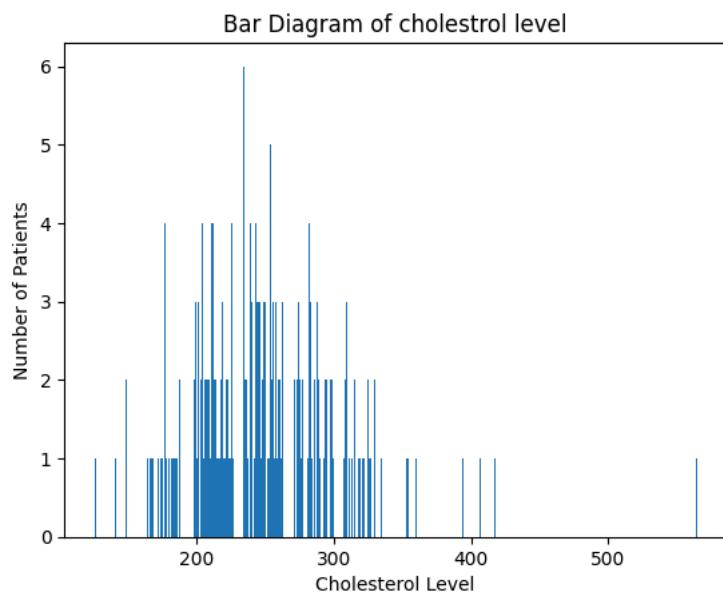
```
df['Heart Disease'].value_counts()
```

```
count  
Heart Disease  
Absence      150  
Presence     120  
dtype: int64
```

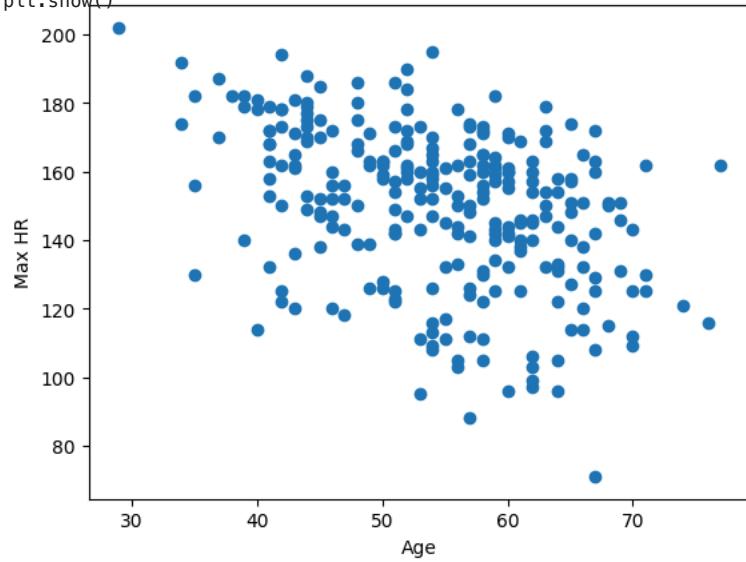
```
plt.hist(age_array,bins=70)  
plt.xlabel('Age')  
plt.ylabel('Number of Patients')  
plt.title('Age Distribution')  
plt.show()
```



```
cholesterol_array=df['Cholesterol'].to_numpy()  
values,counts=np.unique(cholesterol_array,return_counts=True)  
plt.bar(values,counts)  
plt.xlabel('Cholesterol Level')  
plt.ylabel('Number of Patients')  
plt.title('Bar Diagram of cholesterol level')  
plt.show()
```



```
plt.scatter(df['Age'],df['Max HR'])
plt.xlabel('Age')
plt.ylabel('Max HR')
plt.show()
```



```
df_new = df.drop(['Heart Disease'],axis=1)
df_new
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3

270 rows × 13 columns

Start coding or [generate](#) with AI.