

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import pandas as pd
df=pd.read_csv('heart.csv')
```

```
df.describe()
```




	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.529756	149.114146	0.336500
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472700
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	132.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	152.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000	166.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000

```
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
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
```

```
df
```



	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

Next steps:

[Generate code with df](#)[New interactive sheet](#)

```
import pandas as pd
df1=pd.read_csv('Heart_Disease_Prediction.csv')
```

df1.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)
memory usage: 29.7+ KB
```

df1.describe()

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	dep
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	27
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952	
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	

df1.isnull().sum()

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

dtype: int64

df1

	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

270 rows × 14 columns

Next steps: [Generate code with df1](#) [New interactive sheet](#)

df1.nunique()

	0
Age	41
Sex	2
Chest pain type	4
BP	47
Cholesterol	144
FBS over 120	2
EKG results	3
Max HR	90
Exercise angina	2
ST depression	39
Slope of ST	3
Number of vessels fluro	4
Thallium	3
Heart Disease	2

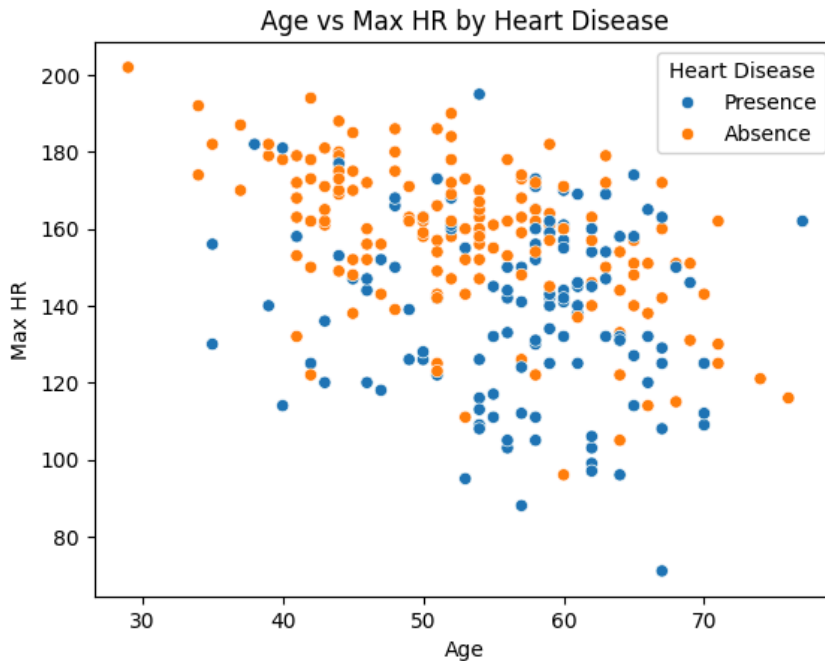
dtype: int64

df1["Smoking"].value_counts()

Show hidden output

Next steps: [Explain error](#)

```
import matplotlib.pyplot as plt
sns.scatterplot(data=df1, x='Age', y='Max HR', hue='Heart Disease')
plt.title('Age vs Max HR by Heart Disease')
plt.xlabel('Age')
plt.ylabel('Max HR')
plt.show()
```



```
df1.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)
memory usage: 29.7+ KB
```

```
## encoding
```

```
df1.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
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

Next steps: [Generate code with df1](#) [New interactive sheet](#)

df1.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')
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
df1['Heart Disease']=le.fit_transform(df1['Heart Disease'])
```

df1

	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	1
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	0
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	1
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	0
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	0
...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	0
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	0
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	0
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	0
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1

270 rows × 14 columns

Next steps: [Generate code with df1](#) [New interactive sheet](#)

```
x=df1.drop('Heart Disease',axis=1)
y=df1['Heart Disease']
```

```
###train test split
```

```
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
##impletation of model
```

```
from sklearn.linear_model import LogisticRegression  
lr=LogisticRegression()  
lr.fit(x_train,y_train)
```

[Show hidden output](#)

```
## scaling
```

```
from sklearn.preprocessing import StandardScaler  
sc=StandardScaler()  
x_train=sc.fit_transform(x_train)  
x_test=sc.transform(x_test)
```

```
x=df1.drop('Heart Disease',axis=1)  
y=df1['Heart Disease']
```

```
x
```

	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
Next steps: Generate code with x New interactive sheet													

y																
Heart Disease			2	140	294	0	2	153	0	1.3	2	0	3			
0			1	4	140	192	0	0	148	0	0.4	2	0	6		
269			67	1	0	4	160	286	0	2	108	1	1.5	2	3	3
2			1													
3			0													
Distributions																