


```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
df = pd.read_csv("/heart.csv")
```

```
df
```




	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Next steps:

 [View recommended plots](#)

```
df.head(10)
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

Next steps:

 [View recommended plots](#)

```
df.target.value_counts()
```

```
target
1    165
0    138
Name: count, dtype: int64

sns.countplot(x="target", data=df, palette="bwr")
plt.show()
```

 **McAfee** | WebAdvisor

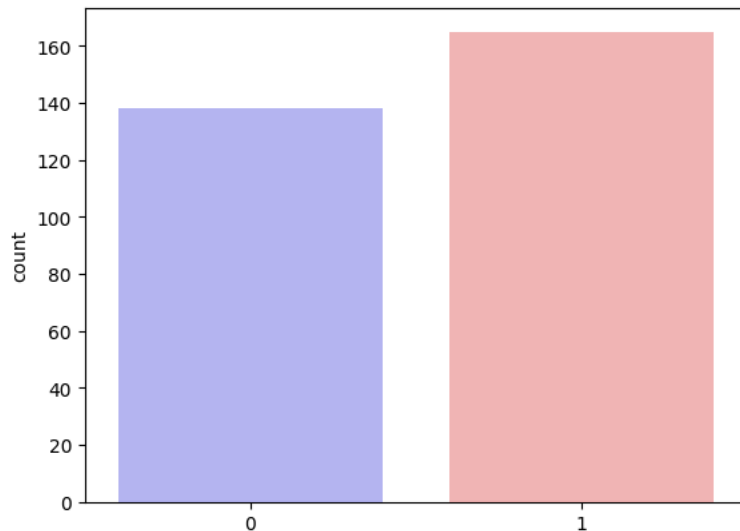


Your download's being scanned.
We'll let you know if there's an issue.

<ipython-input-11-64d5be856ef3>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

```
sns.countplot(x="target", data=df, palette="bwr")
```



```
countNoDisease = len(df[df.target==0])
countHaveDisease = len(df[df.target==1])
print("Percentage of Patients Haven't Heart Disease: {:.2f}%".format((countNoDisease/(len(df.target))*100)))
print("Percentage of Patients Have Heart Disease: {:.2f}%".format((countHaveDisease/(len(df.target))*100)))
```

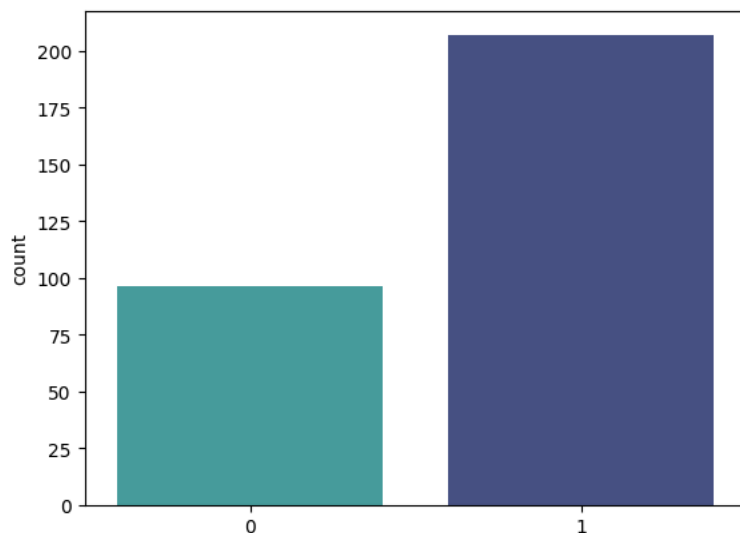
Percentage of Patients Haven't Heart Disease: 45.54%
Percentage of Patients Have Heart Disease: 54.46%

```
sns.countplot(x='sex',data=df,palette="mako_r")
plt.xlabel("Sex (0=Female, 1=Male)")
plt.show()
```

<ipython-input-13-e63205455ec5>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

```
sns.countplot(x='sex',data=df,palette="mako_r")
```



```
countFemale = len(df[df.sex==0])
countMale = len(df[df.sex==1])
print("Percentage of Female Patients: {:.2f}%".format((countFemale/(len(df.sex))*100)))
print("Percentage of Male Patients: {:.2f}%".format((countMale/(len(df.sex))*100)))
```

Percentage of Female Patients: 31.68%
Percentage of Male Patients: 68.32%

```
df.groupby('target').mean()
```



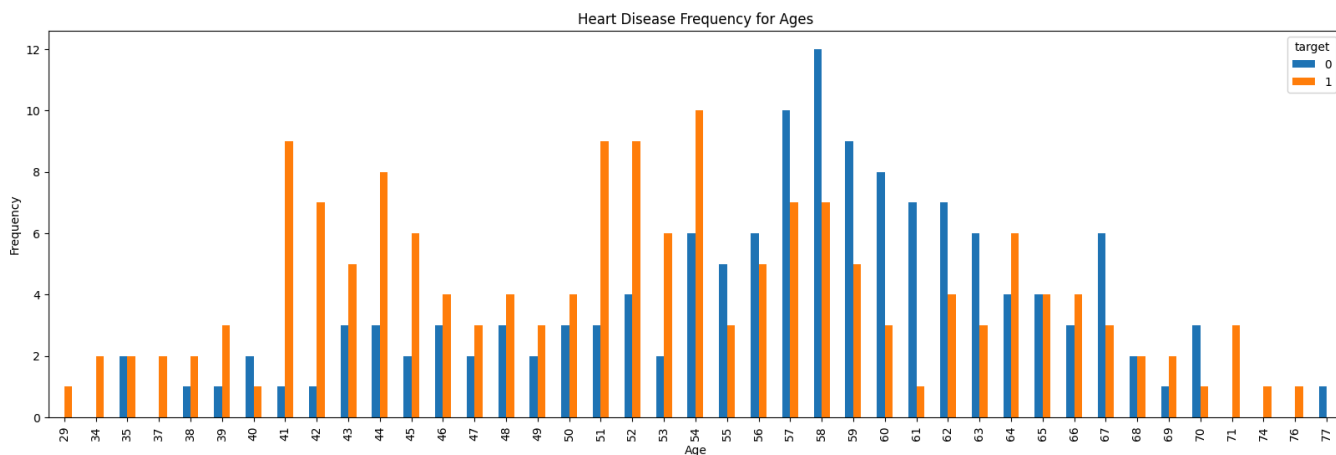
McAfee | WebAdvisor



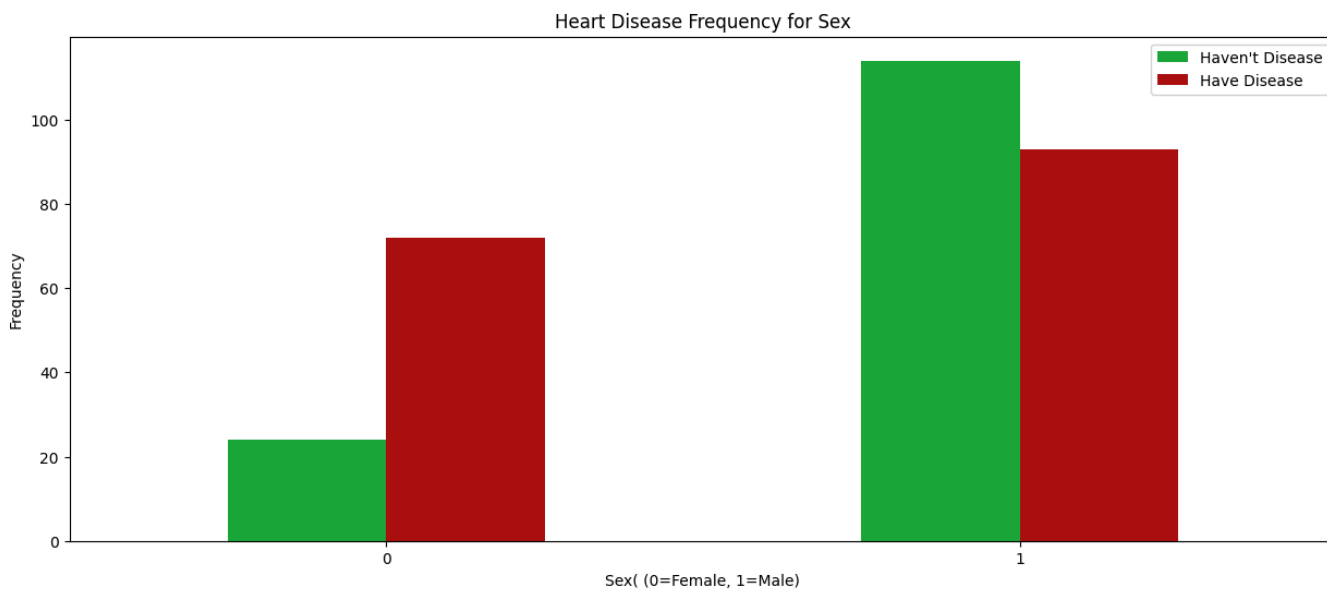
Your download's being scanned.
We'll let you know if there's an issue.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
target													
0	56.601449	0.826087	0.478261	134.398551	251.086957	0.159420	0.449275	139.101449	0.550725	1.585507	1.166667	1.166667	2.5

```
pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
plt.title("Heart Disease Frequency for Ages")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.savefig("HeartDiseasesAndAges.png")
plt.show()
```



```
pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6),color=["#1CA53B","#AA1111"])
plt.title("Heart Disease Frequency for Sex")
plt.xlabel("Sex( 0=Female, 1=Male)")
plt.xticks(rotation=0)
plt.legend(["Haven't Disease","Have Disease"])
plt.ylabel("Frequency")
plt.show()
```

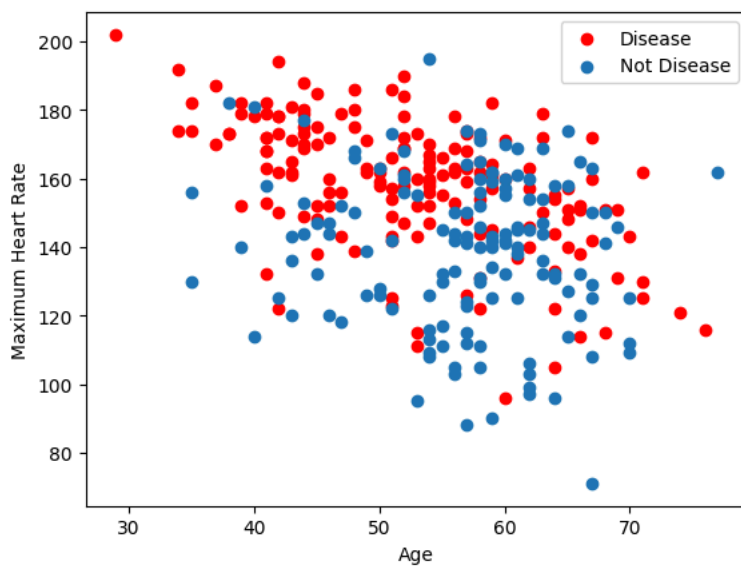


McAfee | WebAdvisor

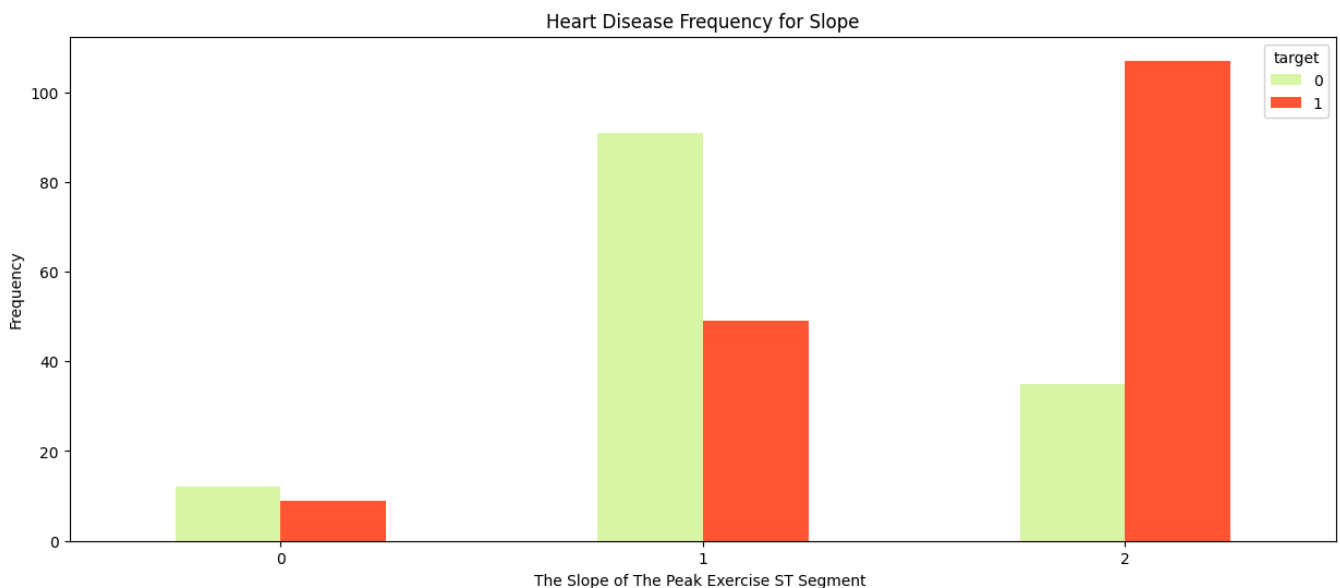


Your download's being scanned.
We'll let you know if there's an issue.

```
plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="red")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Disease", "Not Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.show()
```



```
pd.crosstab(df.slope,df.target).plot(kind="bar",figsize=(15,6),color=[ '#DAF7A6', '#FF5733' ])
plt.title('Heart Disease Frequency for Slope')
plt.xlabel('The Slope of The Peak Exercise ST Segment ')
plt.xticks(rotation = 0)
plt.ylabel('Frequency')
plt.show()
```



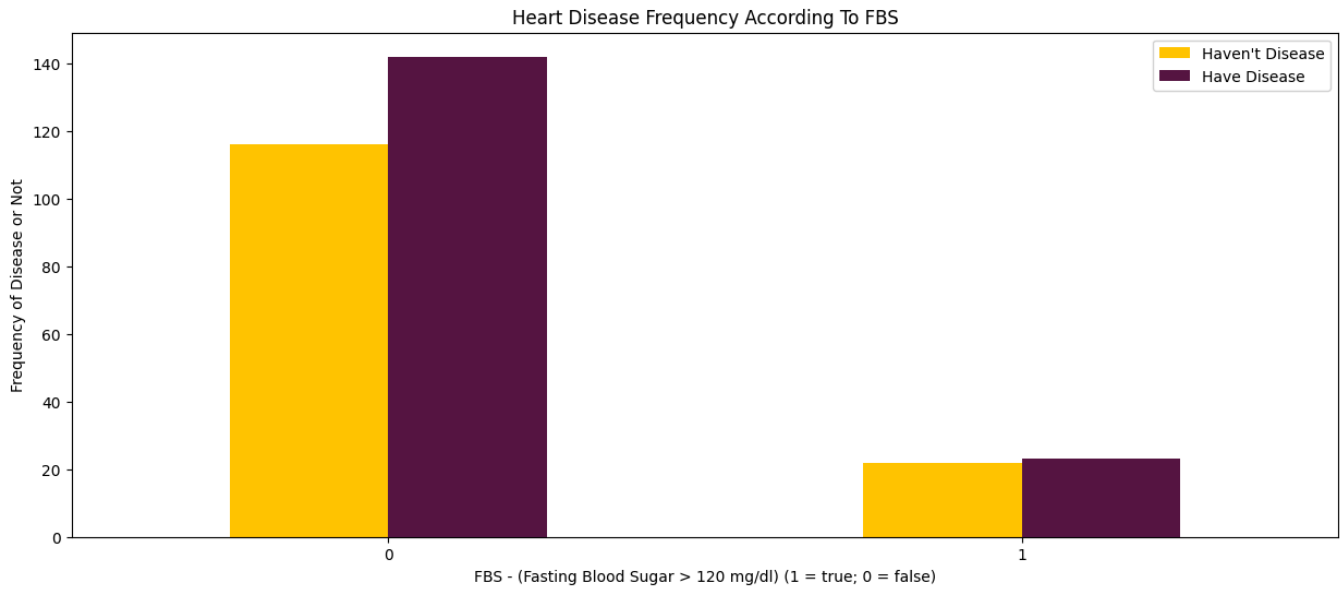
```
pd.crosstab(df.fbs,df.target).plot(kind="bar",figsize=(15,6),color=[ '#FFC300', '#581845' ])
plt.title('Heart Disease Frequency According To FBS')
plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation = 0)
plt.legend(["Haven't Disease", "Have Disease"])
plt.ylabel('Frequency of Disease or Not')
plt.show()
```



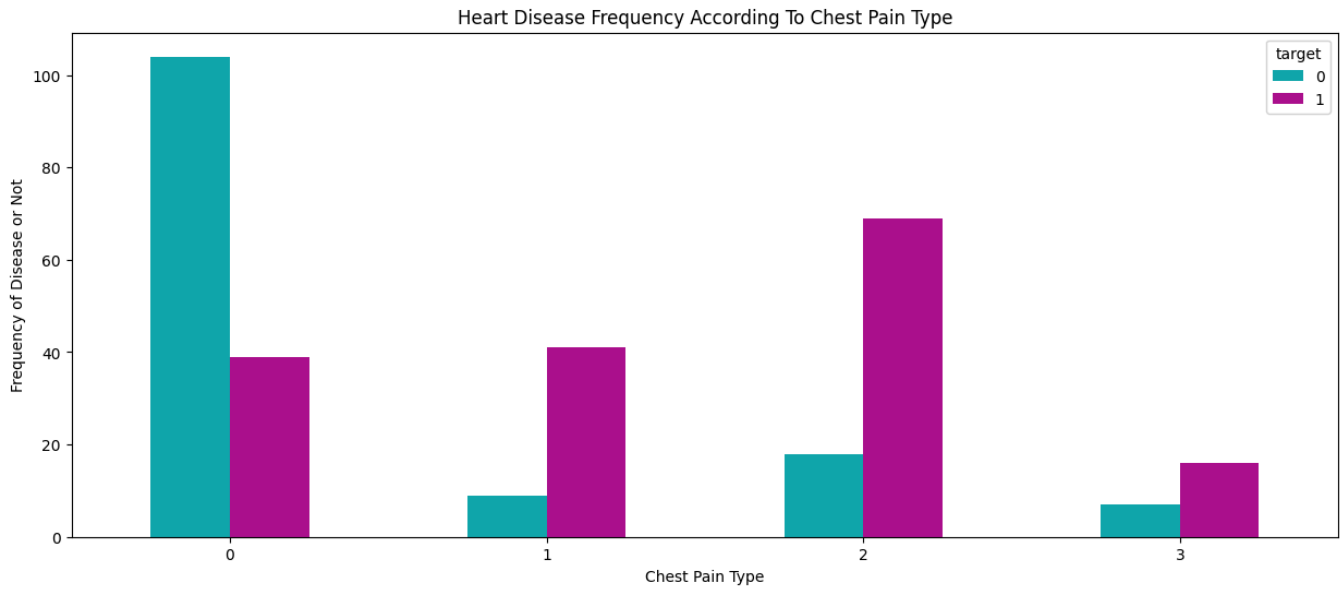
McAfee | WebAdvisor



Your download's being scanned.
We'll let you know if there's an issue.



```
pd.crosstab(df.cp,df.target).plot(kind="bar",figsize=(15,6),color=[ '#11A5AA', '#AA1190' ])
plt.title('Heart Disease Frequency According To Chest Pain Type')
plt.xlabel('Chest Pain Type')
plt.xticks(rotation = 0)
plt.ylabel('Frequency of Disease or Not')
plt.show()
```



Your download's being scanned.
We'll let you know if there's an issue.