

dsbda7

March 13, 2024

[16]: `!pip install seaborn`

```
Requirement already satisfied: seaborn in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from seaborn) (1.25.1)
Requirement already satisfied: pandas>=1.2 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from seaborn) (2.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from seaborn) (3.8.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.1.0)
Requirement already satisfied: cycler>=0.10 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.42.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (23.1)
Requirement already satisfied: pillow>=8 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.0.9)
```

Requirement already satisfied: python-dateutil>=2.7 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pandas>=1.2->seaborn) (2023.3)

Requirement already satisfied: tzdata>=2022.1 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pandas>=1.2->seaborn) (2023.3)

Requirement already satisfied: six>=1.5 in c:\users\namya\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)

[notice] A new release of pip is available: 23.3.1 -> 24.0

[notice] To update, run: C:\Users\namya\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip

```
[17]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
```

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

```
[19]: 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

```

```
[20]: df.describe()
```

```

[20]:
count    age      sex      cp      trestbps      chol \
mean     54.434146  0.695610  0.942439  131.611707  246.00000
std       9.072290  0.460373  1.029641  17.516718   51.59251
min      29.000000  0.000000  0.000000  94.000000  126.00000
25%      48.000000  0.000000  0.000000  120.000000  211.00000
50%      56.000000  1.000000  1.000000  130.000000  240.00000
75%      61.000000  1.000000  2.000000  140.000000  275.00000
max      77.000000  1.000000  3.000000  200.000000  564.00000

count    fbs      restecg      thalach      exang      oldpeak \
mean     0.149268  0.529756  149.114146  0.336585   1.071512
std       0.356527  0.527878  23.005724  0.472772   1.175053
min       0.000000  0.000000  71.000000  0.000000   0.000000
25%       0.000000  0.000000  132.000000  0.000000   0.000000
50%       0.000000  1.000000  152.000000  0.000000   0.800000
75%       0.000000  1.000000  166.000000  1.000000   1.800000
max       1.000000  2.000000  202.000000  1.000000   6.200000

count    slope      ca      thal      target
mean     1.385366  0.754146  2.323902  0.513171
std       0.617755  1.030798  0.620660  0.500070
min       0.000000  0.000000  0.000000  0.000000
25%       1.000000  0.000000  2.000000  0.000000
50%       1.000000  0.000000  2.000000  1.000000
75%       2.000000  1.000000  3.000000  1.000000
max       2.000000  4.000000  3.000000  1.000000

```

```
[21]: df.head()
```

```

[21]:
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope \
0   52   1   0     125    212   0         1     168     0       1.0     2
1   53   1   0     140    203   1         0     155     1       3.1     0
2   70   1   0     145    174   0         1     125     1       2.6     0
3   61   1   0     148    203   0         1     161     0       0.0     2
4   62   0   0     138    294   1         1     106     0       1.9     1

   ca  thal  target
0   2     3       0

```

1	0	3	0
2	0	3	0
3	1	3	0
4	3	2	0

```
[22]: df.target.value_counts()
```

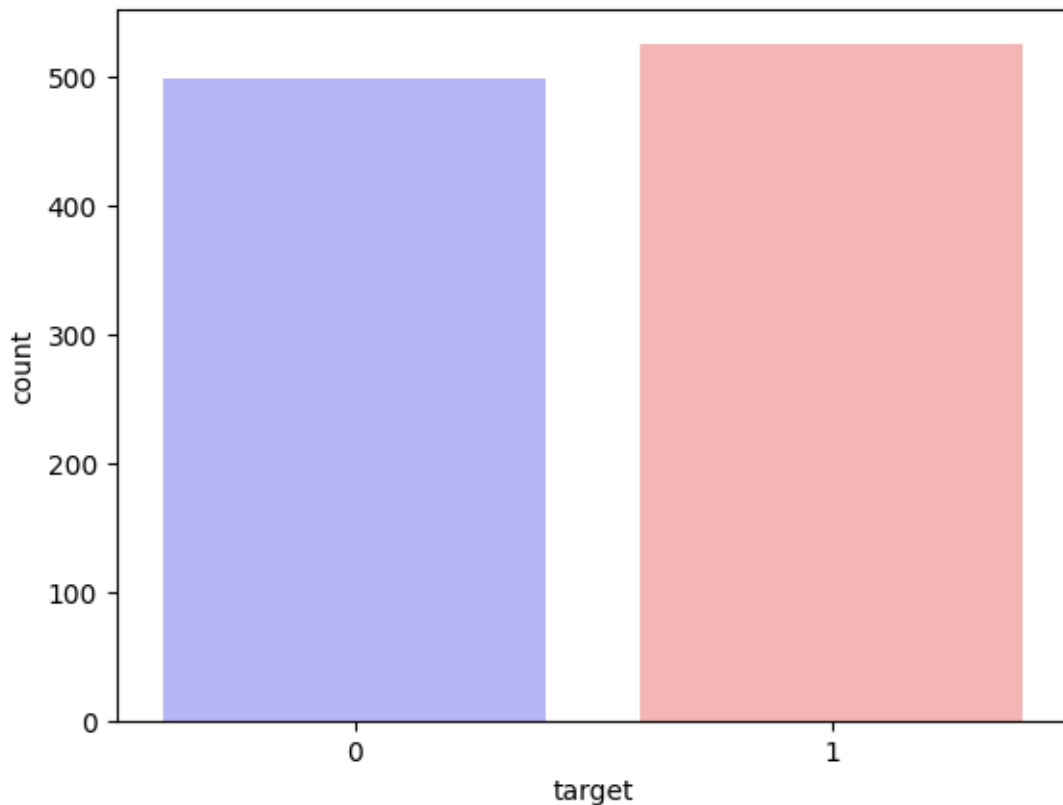
```
[22]: target
1      526
0      499
Name: count, dtype: int64
```

```
[24]: sn.countplot(x="target", data=df, palette="bwr")
plt.show()
```

C:\Users\namya\AppData\Local\Temp\ipykernel_27516\3182577542.py: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 `legend=False` for the same effect.

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



```
[27]: countNoDisease = len(df[df.target == 0])
countHaveDisease = len(df[df.target == 1])
print("Percentage of Patients Haven't Heart Disease: ", (countNoDisease /
↳ len(df.target))*100, "%")
print("Percentage of Patients Have Heart Disease: ", (countHaveDisease /
↳ len(df.target))*100, "%")
```

Percentage of Patients Haven't Heart Disease: 48.68292682926829 %

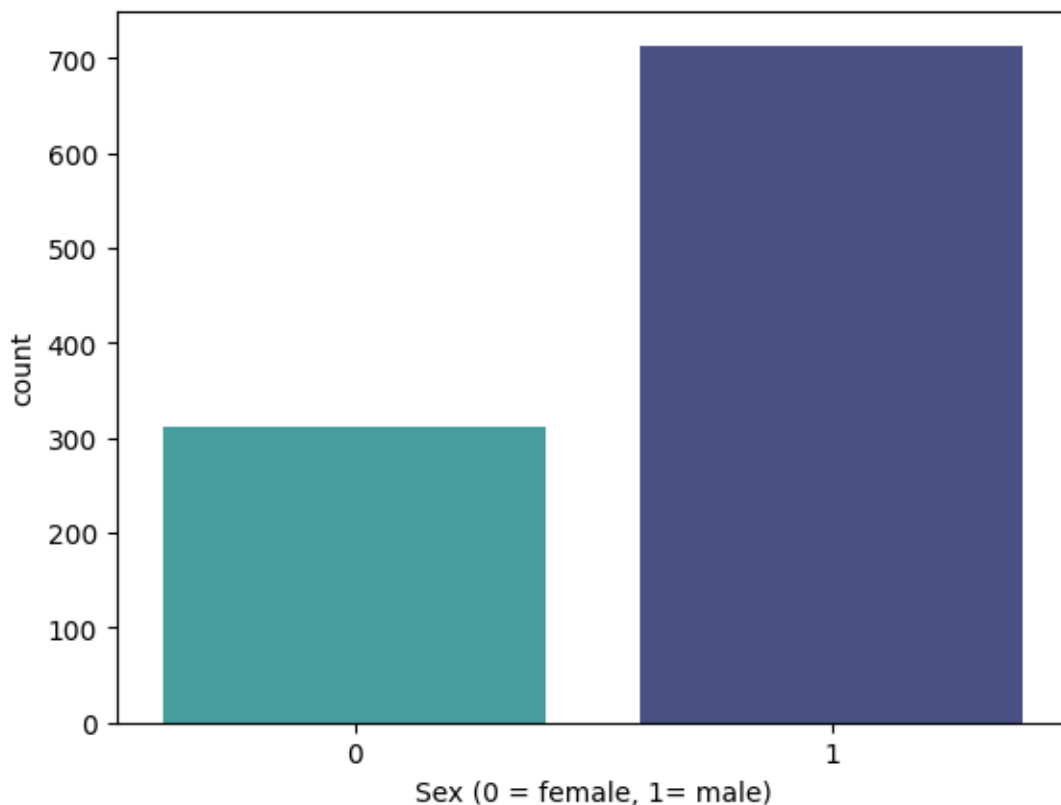
Percentage of Patients Have Heart Disease: 51.31707317073171 %

```
[28]: sn.countplot(x='sex', data=df, palette="mako_r")
plt.xlabel("Sex (0 = female, 1= male)")
plt.show()
```

C:\Users\namya\AppData\Local\Temp\ipykernel_27516\930773917.py: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 `legend=False` for the same effect.

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



```
[30]: countFemale = len(df[df.sex == 0])
countMale = len(df[df.sex == 1])
print("Percentage of Female Patients:",(countFemale / (len(df.sex))*100), "%")
print("Percentage of Male Patients:",(countMale / (len(df.sex))*100), "%")
```

Percentage of Female Patients: 30.4390243902439 %
Percentage of Male Patients: 69.5609756097561 %

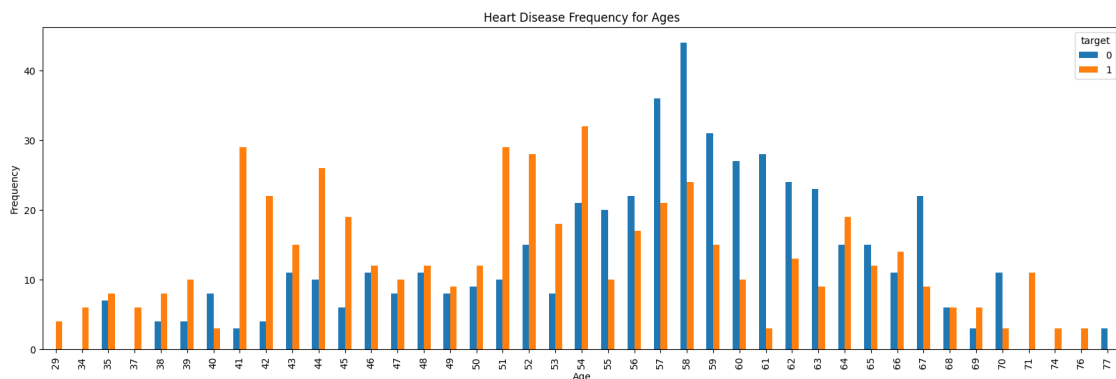
```
[31]: df.groupby('target').mean()
```

```
[31]:
```

	age	sex	cp	trestbps	chol	fbs	\
target							
0	56.569138	0.827655	0.482966	134.106212	251.292585	0.164329	
1	52.408745	0.570342	1.378327	129.245247	240.979087	0.134981	

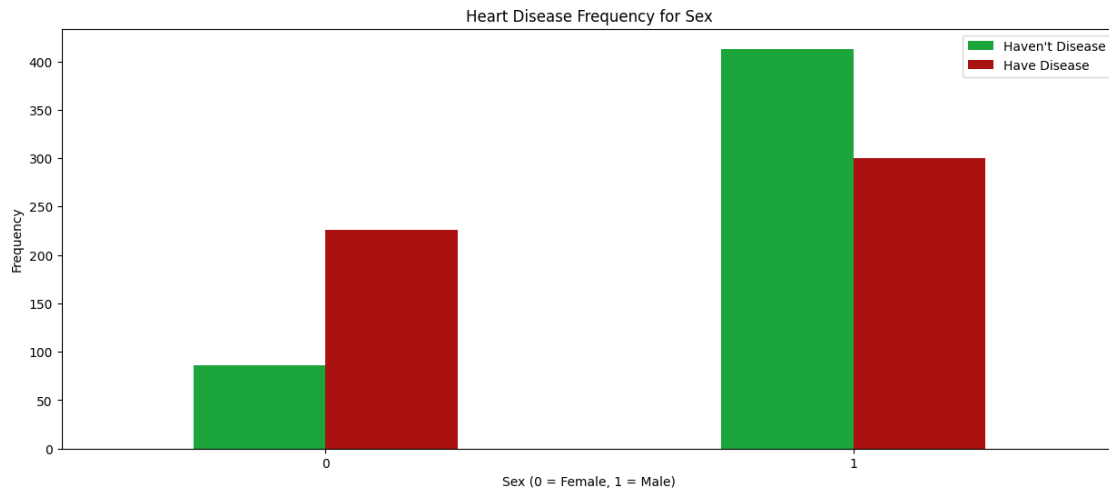
	restecg	thalach	exang	oldpeak	slope	ca	thal
target							
0	0.456914	139.130261	0.549098	1.600200	1.166333	1.158317	2.539078
1	0.598859	158.585551	0.134981	0.569962	1.593156	0.370722	2.119772

```
[33]: 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('heartDiseaseAndAges.png')
plt.show()
```

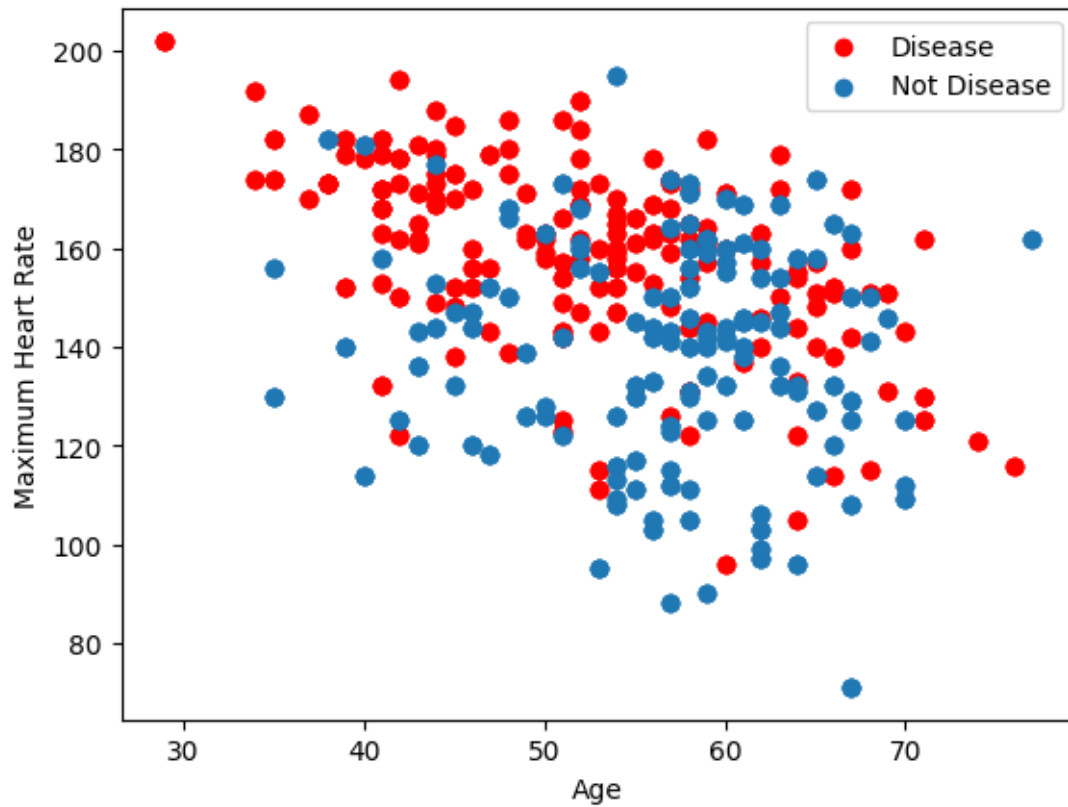


```
[34]: pd.crosstab(df.sex,df.target).
      .plot(kind="bar",figsize=(15,6),color=['#1CA53B','#AA1111' ])
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
```

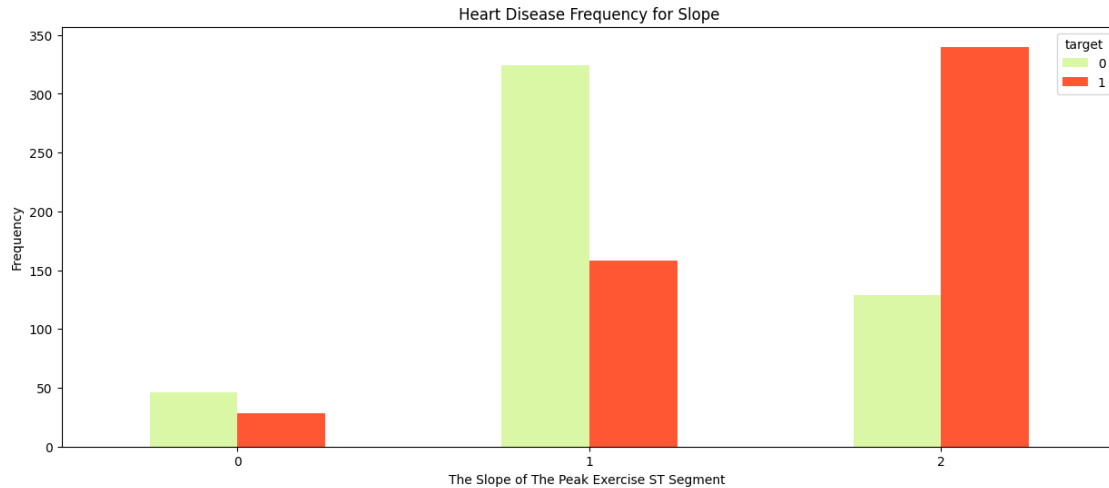
```
plt.xticks(rotation=0)
plt.legend(["Haven't Disease", "Have Disease"])
plt.ylabel('Frequency')
plt.show()
```



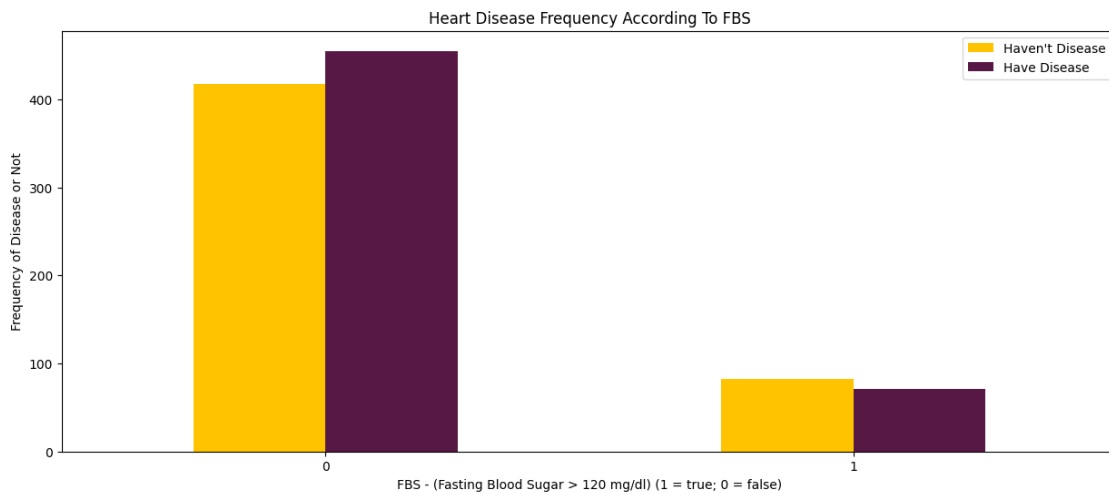
```
[35]: plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="red")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Disease", "Not Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.show()
```



```
[36]: 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()
```

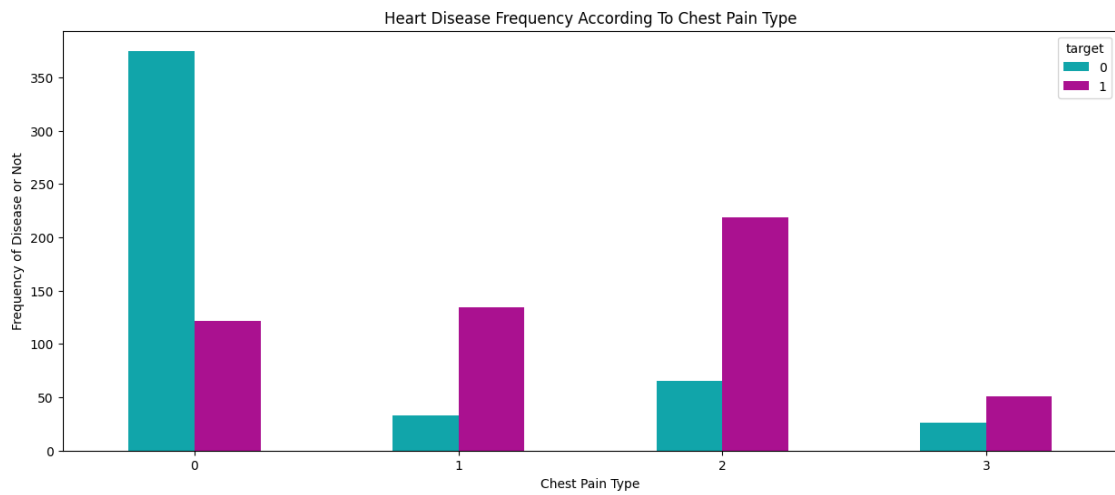



```
[37]: pd.crosstab(df.fbs,df.target).
      ↪plot(kind="bar",figsize=(15,6),color=['#FFC300','#581845' ])
plt.title('Heart Disease Frequency According To FBS')
plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation = 0)
plt.legend(["Haven't Disease", "Have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



```
[38]: 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()
```



```
[ ]:
```