

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
data = pd.read_csv("Heart.csv")
data.head(10)
```

Out[2]:

	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	37	1	2	130	250	0	1	187	0	3.5	0	0	2	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	

In [3]:

```
data.columns
```

Out[3]:

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
      'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [4]:

```
data.describe()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

In [5]:

```
# summary of dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age          303 non-null int64
sex          303 non-null int64
cp           303 non-null int64
trestbps     303 non-null int64
chol         303 non-null int64
fbs          303 non-null int64
restecg      303 non-null int64
thalach      303 non-null int64
exang        303 non-null int64
oldpeak      303 non-null float64
slope        303 non-null int64
ca           303 non-null int64
thal         303 non-null int64
target       303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

In [6]:

```
data.dtypes
```

Out[6]:

```
age          int64
sex          int64
cp           int64
trestbps     int64
chol         int64
fbs          int64
restecg      int64
thalach      int64
exang        int64
oldpeak      float64
slope        int64
ca           int64
thal         int64
target       int64
dtype: object
```

In [7]:

```
data.shape
```

Out[7]:

```
(303, 14)
```

Check the number of unique values in target variable¶

In [8]:

```
data['target'].unique()
```

Out[8]:

```
array([1, 0])
```

So, the unique values are 1 and 0. (1 stands for presence of heart disease and 0 for absence of heart disease).

Data Exploration

Question Number 1. How many are suffering from heart disease? Also plot the stats.

In [9]:

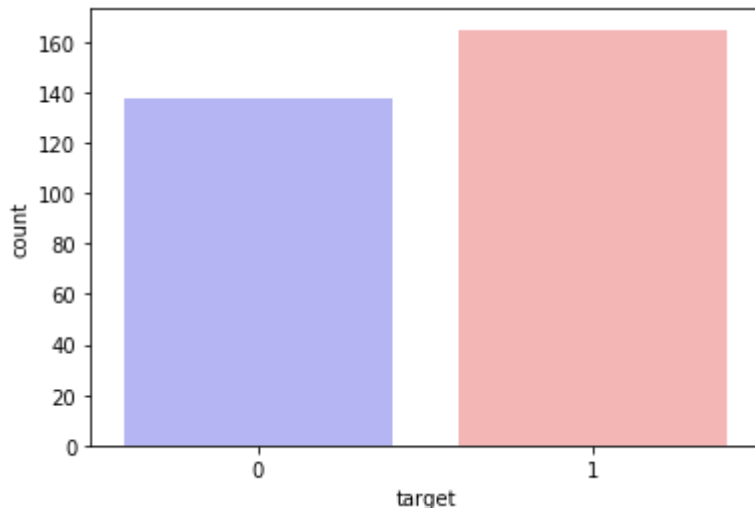
```
data.target.value_counts()
```

Out[9]:

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

In [10]:

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



Question Number 2. How many males and females have heart disease out of total?

In [11]:

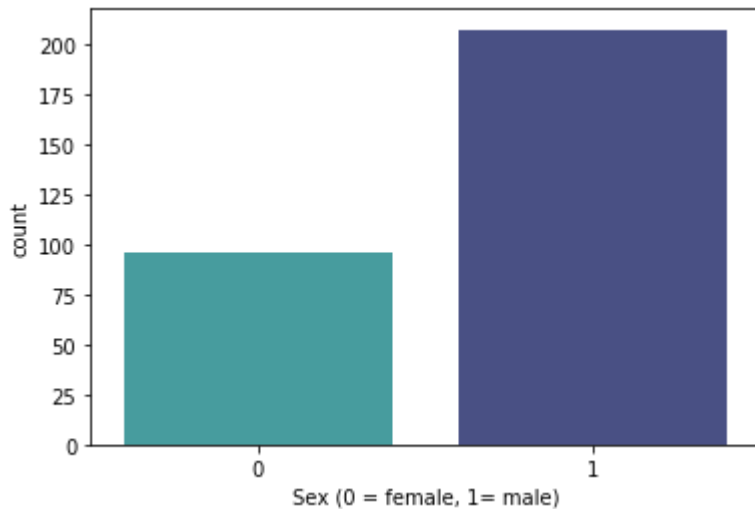
```
countNoDisease = len(data[data.target == 0])
countHaveDisease = len(data[data.target == 1])
print("Percentage of Patients Haven't Heart Disease: {:.2f}%".format((countNoDisease /
(len(data.target))*100)))
print("Percentage of Patients Have Heart Disease: {:.2f}%".format((countHaveDisease / (
len(data.target))*100)))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

Percentage of Patients Haven't Heart Disease: 45.54%

Percentage of Patients Have Heart Disease: 54.46%

In [12]:

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



In [13]:

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

Percentage of Female Patients: 31.68%

Percentage of Male Patients: 68.32%

In [14]:

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

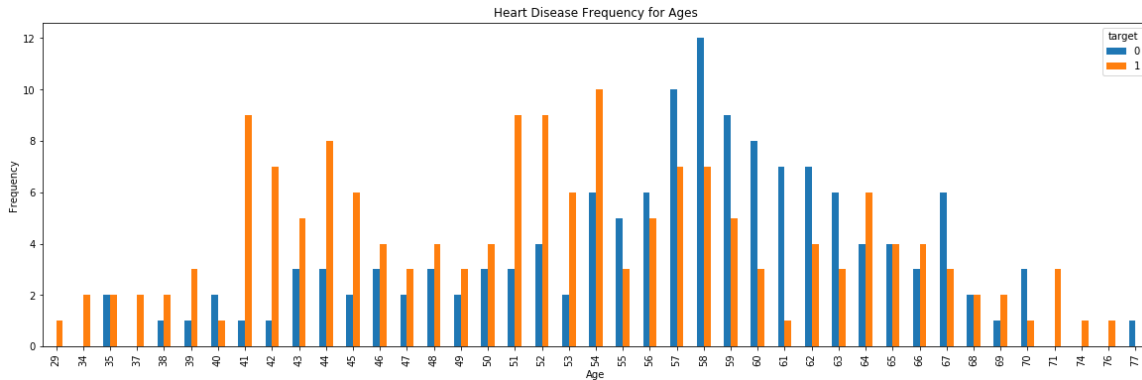
Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalac
target								
0	56.601449	0.826087	0.478261	134.398551	251.086957	0.159420	0.449275	139.10144
1	52.496970	0.563636	1.375758	129.303030	242.230303	0.139394	0.593939	158.46666

3. Visualize frequency distribution of the thalach variable and find what's the heart rate and heart disease relation?

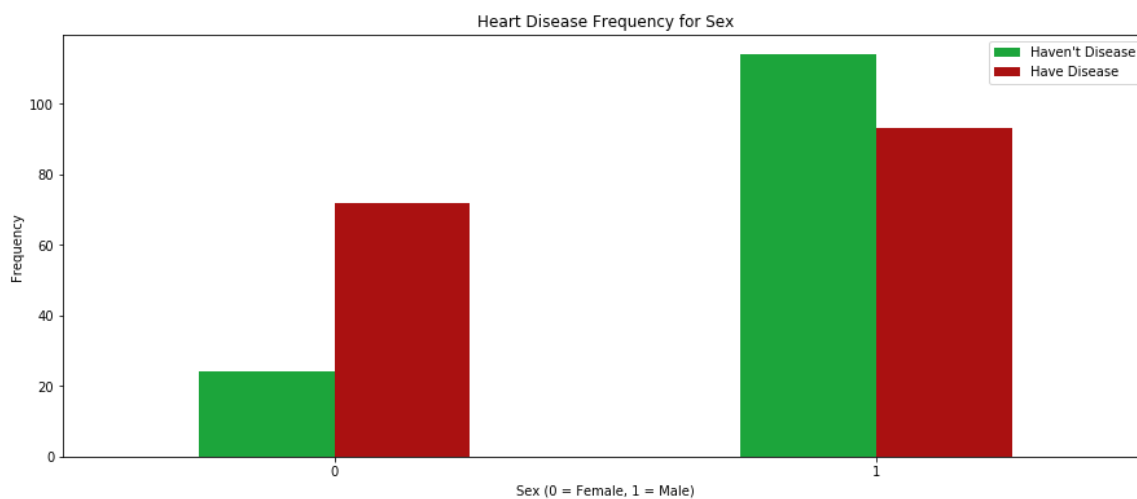
In [15]:

```
pd.crosstab(data.age,data.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()
```



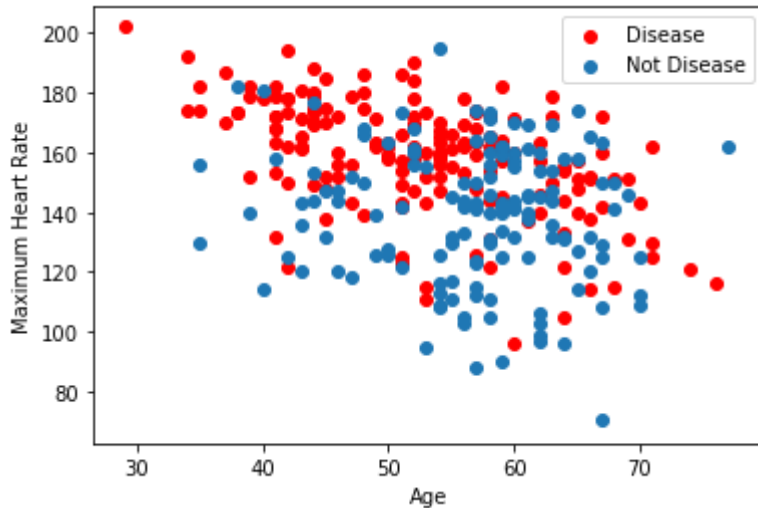
In [16]:

```
pd.crosstab(data.sex,data.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()
```



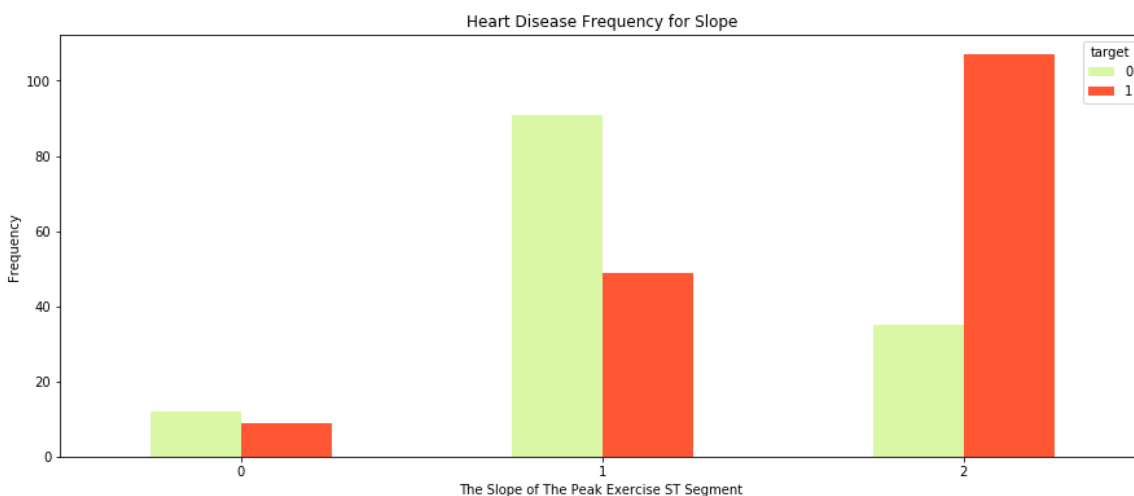
In [17]:

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



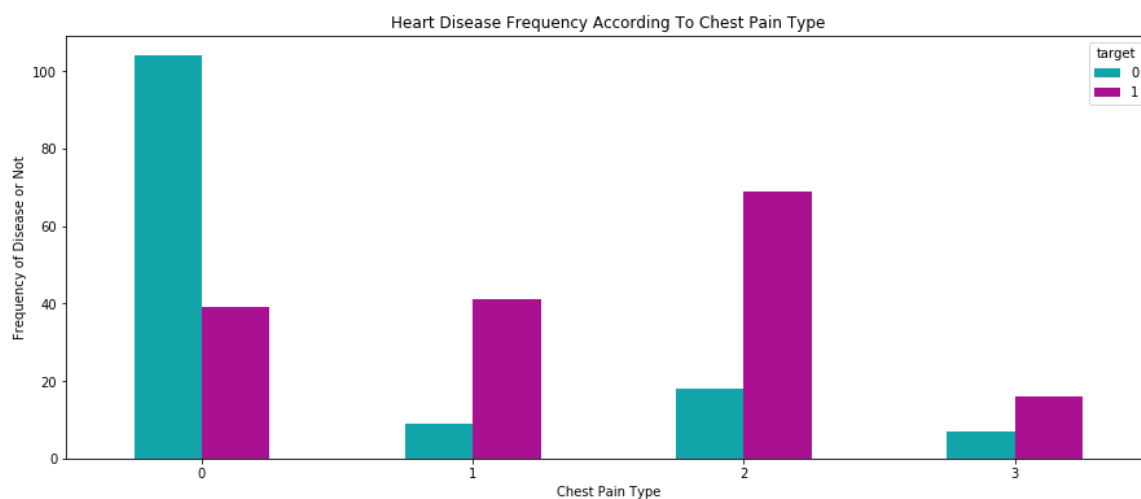
In [18]:

```
pd.crosstab(data.slope,data.target).plot(kind="bar",figsize=(15,6),color=[ '#DAF7A6', '#F5733' ])
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()
```



In [19]:

```
pd.crosstab(data.cp,data.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()
```



In []: