This dataset was obtained from data.world.com (Heart Disease Prediction). It was downloaded via Excel. The data was then cleaned. The Gender and Chest Pain Type column initially had numbers, these were replaced with the correct values using =IF and (highlight) CTRL+F functions. Converting what would have been viewed as Quantitative variables into Categorical Variables. (This text was written using the 'Markdown' tab)

In [75]:

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

In [49]:

```
import os
os.getcwd()
```

Out[49]:

```
'C:\\Users\\Team Knowhow\\Documents'
```

In [50]:

```
os.chdir('C:\\Users\\Team Knowhow\\Documents')
os.getcwd()
```

Out[50]:

```
'C:\\Users\\Team Knowhow\\Documents'
```

In [51]:

```
heart = pd.read_csv('Heart_Disease_Prediction.csv')
print(heart.head(2))
```

```
   Age  Gender    Chest pain type   BP  Cholesterol  FBS over 120  EKG results  \
0   70    Male       asymptomatic  130          322             0            2   
1   67  Female  non-anginal pain  115          564             0            2   

   Max HR  Exercise angina  ST depression  Slope of ST  \
0     109                0            2.4            2   
1     160                0            1.6            2   

   Number of vessels fluro  Thallium Heart Disease  
0                        3         3      Presence  
1                        0         7       Absence  
```

In [52]:

```
heart.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   Gender                   270 non-null    object 
 2   Chest pain type          270 non-null    object 
 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(10), object(3)
memory usage: 29.7+ KB
```

In [53]:

```
#Look at the unqiue values in a DataFrame column
print(heart['Heart Disease'].unique())
```

```
['Presence' 'Absence']
```

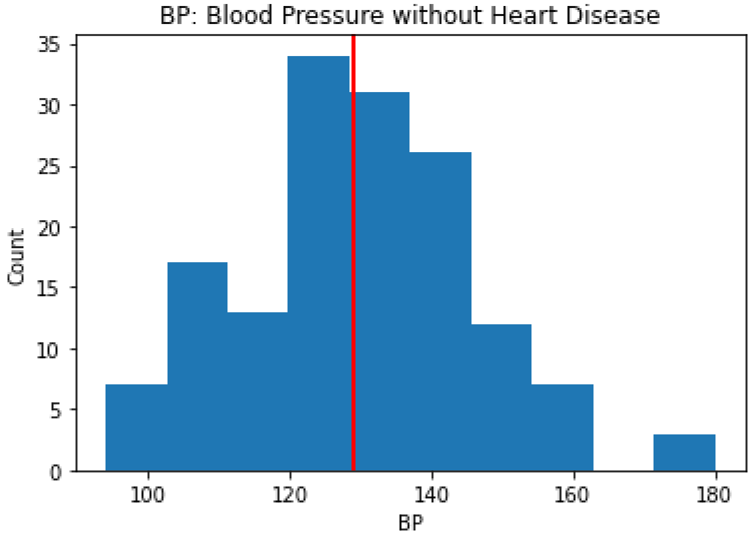
In [54]:

```
#We will create two new DF one with and one without HD.
heart_dis_abs = heart[heart['Heart Disease'] == "Absence"]
heart_dis_pre = heart[heart['Heart Disease'] == "Presence"]
```

In [55]:

```
average_bp_abs = np.mean(heart_dis_abs['BP'])
print("The avearge Blood Pressure for people without HD in this dataset is: " + str(averag
plt.hist(heart_dis_abs['BP'], bins=10)
plt.title("BP: Blood Pressure without Heart Disease")
plt.xlabel("BP")
plt.ylabel("Count")
plt.axvline(average_bp_abs, color='r', linestyle='solid', linewidth=2, label='Mean')
plt.show()
```

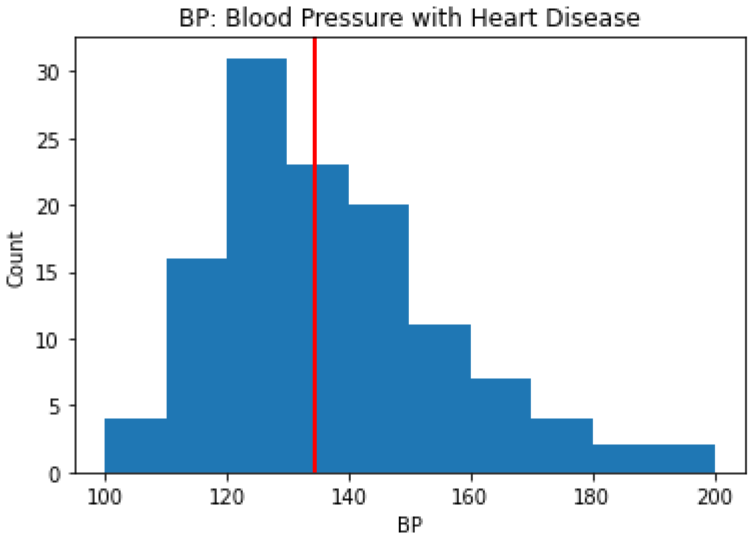
```
The avearge Blood Pressure for people without HD in this dataset is: 128.86666666666667
```



In [56]:

```
average_bp_pre = np.mean(heart_dis_pre['BP'])
print("The avearge Blood Pressure for people with HD in this dataset is: " + str(average_
plt.hist(heart_dis_pre['BP'], bins=10)
plt.title("BP: Blood Pressure with Heart Disease")
plt.xlabel("BP")
plt.ylabel("Count")
plt.axvline(average_bp_pre, color='r', linestyle='solid', linewidth=2, label='Mean')
plt.show()
```

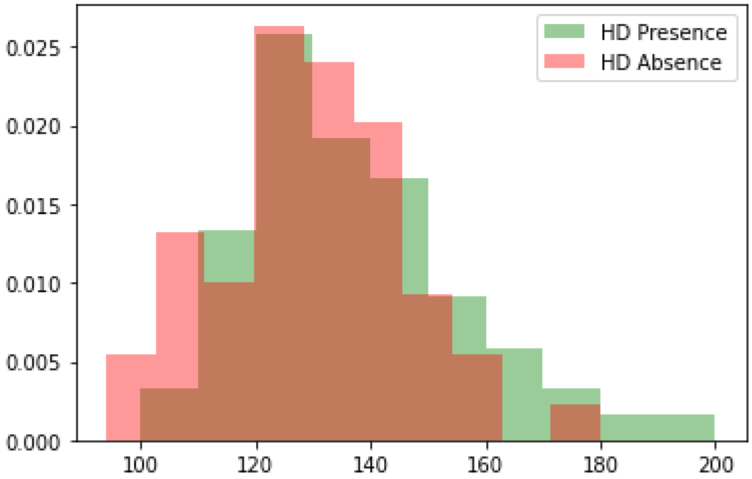
```
The avearge Blood Pressure for people with HD in this dataset is: 134.44166666666666
```



In [57]:

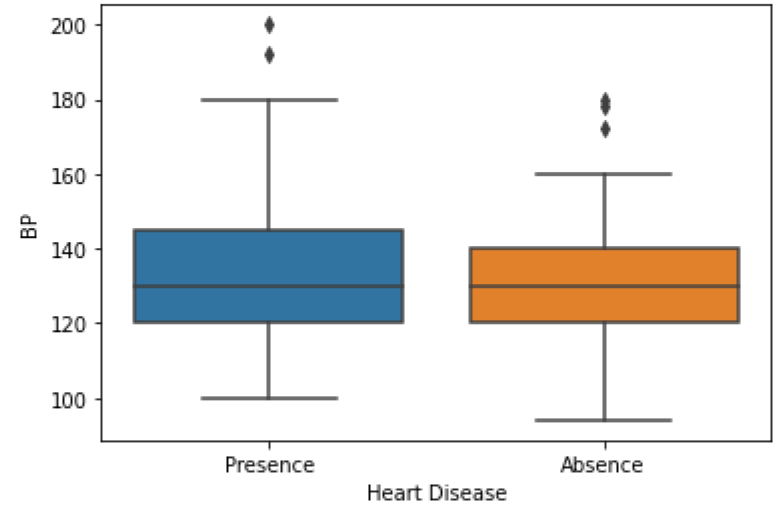
```
#We can see that the histogram is Positive Right Skewed. Thus, the Mean > Median > Mode
#If the data set's lower bounds are extremely low relative to the rest of the data, this w

#Overlapping Histograms
plt.hist(heart_dis_pre['BP'], color='Green', density=True, alpha=0.4, bins=10, label='HD P
plt.hist(heart_dis_abs['BP'], color='Red', density=True, alpha=0.4, bins=10, label='HD Abs
plt.title("Overlapping Histograms to compate the BP for thioose with and without Heart Dis
plt.legend()
plt.show()
```

Overlapping Histograms to compate the BP for thioose with and without Heart Disease.



In [58]:

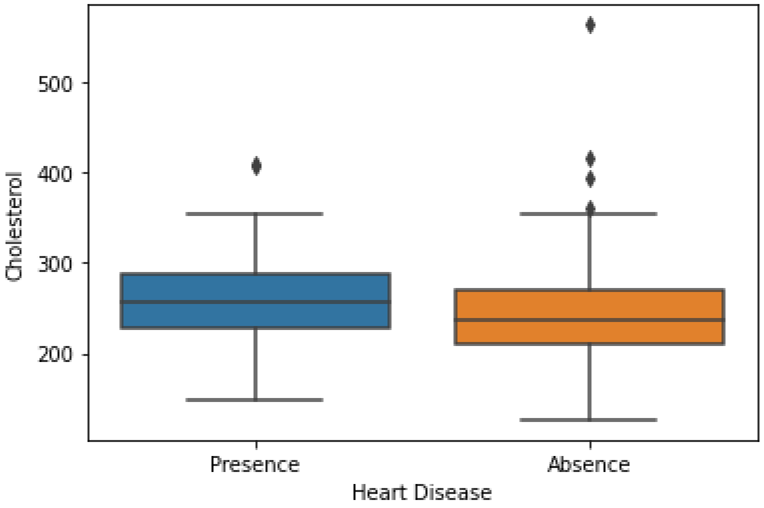
```
#Use Boxplots to quickly compare the Blood Pressure for those with and without Heart Disea
sns.boxplot(data=heart, x='Heart Disease', y='BP')
plt.title("A Boxplot to compare the Blood Pressure of those with and without Heart Disease
plt.show()
```



A Boxplot to compare the Blood Pressure of those with and without Heart Disease

In [59]:

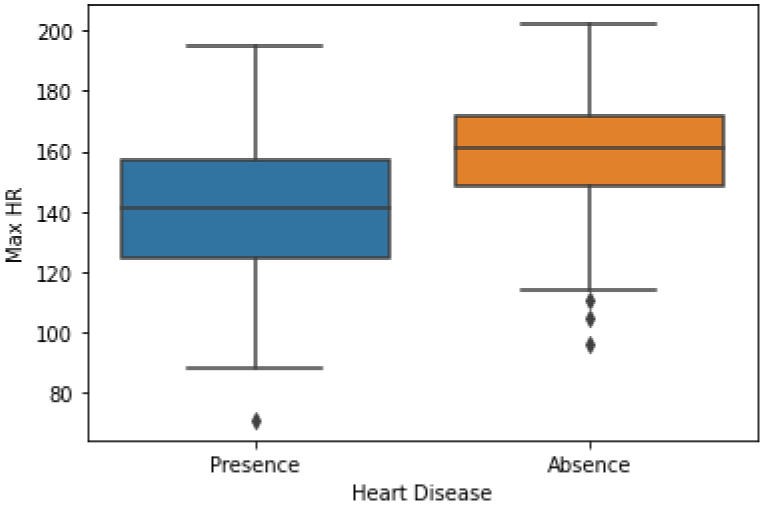
```
sns.boxplot(data=heart, x='Heart Disease', y='Cholesterol')
plt.title("A Boxplot to compare the Cholesterol of those with and without Heart Disease")
plt.show()
```



A Boxplot to compare the Cholesterol of those with and without Heart Disease

In [60]:

```
sns.boxplot(data=heart, x='Heart Disease', y='Max HR')
plt.title("A Boxplot to compare the Max Heart Rate of those with and without Heart Disease
plt.show()
```



A Boxplot to compare the Max Heart Rate of those with and without Heart Disease

In [72]:

```
#groupby function for summary statistics, which automatically calculates the mean for all
heart.groupby('Gender').mean()
```

Out[72]:

	Age	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro
Gender										
Female	55.678161	132.965517	264.747126	0.126437	0.965517	152.229885	0.206897	0.888506	1.540230	0.551724
Male	53.841530	130.573770	242.486339	0.158470	1.049180	148.464481	0.387978	1.126776	1.606557	0.726776

In [73]:

```
heart.groupby('Gender').median()
```

Out[73]:

	Age	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
Gender											
Female	57	132	263	0	0	158	0	0.6	1	0	3
Male	54	130	239	0	2	150	0	0.9	2	0	6

In [63]:

```
heart.groupby('Heart Disease').mean()
```

Out[63]:

	Age	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro
Heart Disease										
Absence	52.706667	128.866667	244.213333	0.153333	0.860	158.333333	0.153333	0.622667	1.400000	0.286667
Presence	56.591667	134.441667	256.466667	0.141667	1.225	138.858333	0.550000	1.584167	1.816667	1.150000

In [64]:

```
#Hypothesis Test, 2-Sample-T-Tests are the most commonly used hypothesis tests.

#We will use the 3rd boxplot graph to guide us. The difference between the Max HR shown fo
#Is this true of the enire population?
```

In [65]:

```
print("NULL/H0: There is NO difference in the mean Cholesterol level (among patient who do
print("ALTERNATIVE/H1: There IS a difference in the mean Cholesterol (among patient who do
```

```
NULL/H0: There is NO difference in the mean Cholesterol level (among patient who do and do
not have Heart Disease).
ALTERNATIVE/H1: There IS a difference in the mean Cholesterol (among patient who do and do
not have Heart Disease).
```

In [66]:

```
print("Another way of saying the NULL: true mean of cholesterol for people with hd - true
```

```
Another way of saying the NULL: true mean of cholesterol for people with hd - true mean of
cholesterol for people without hr = 0
```

In [67]:

```
heart[['Cholesterol', 'Heart Disease']].head()
```

Out[67]:

	Cholesterol	Heart Disease
0	322	Presence
1	564	Absence
2	261	Presence
3	263	Absence
4	269	Absence

In [68]:

```
#heart_dis_abs heart_dis_pre
from scipy.stats import ttest_ind
tstat, pval = ttest_ind(heart_dis_abs["Cholesterol"], heart_dis_pre["Cholesterol"])
print(pval)
```

```
0.05273888557034281
```

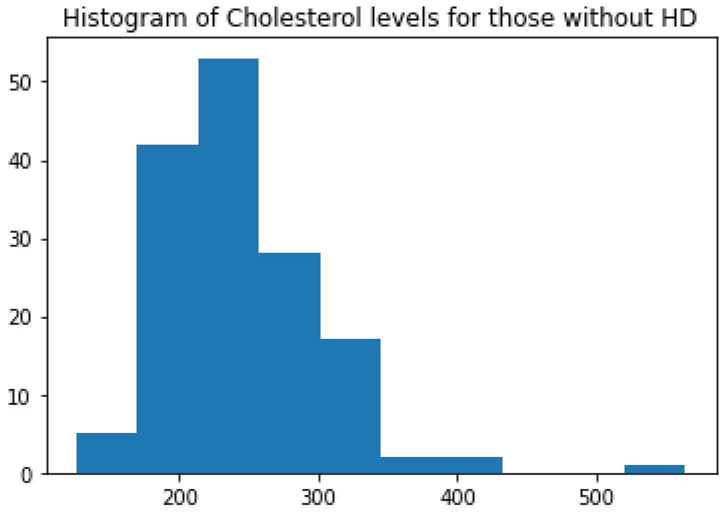
In [69]:

```
print("The P-Val is > 0.05, thus we accept the Null and reject the Alternative.")
```

```
The P-Val is > 0.05, thus we accept the Null and reject the Alternative.
```

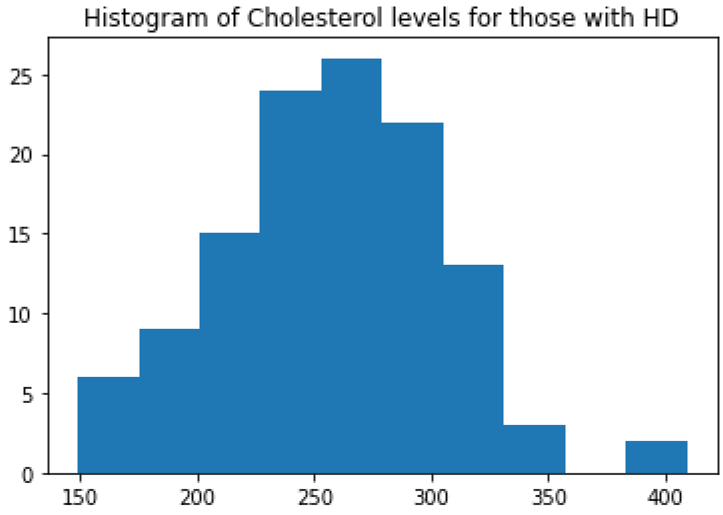
In [70]:

```
plt.hist(heart_dis_abs["Cholesterol"])
plt.title("Histogram of Cholesterol levels for those without HD")
plt.show()
```



In [71]:

```
plt.hist(heart_dis_pre["Cholesterol"])
plt.title("Histogram of Cholesterol levels for those with HD")
plt.show()
```



In [74]:
```
heart['Chest pain type'].unique()
```

Out[74]:
```
array(['asymptomatic', 'non-anginal pain', 'atypical angina',
       'typical angina'], dtype=object)
```

In [85]:
```
#ANOVA test on Chest Pain Type and Max HR

maxhr_asymptomatic = heart['Max HR'][heart["Chest pain type"] == 'asymptomatic']
maxhr_non_anginal = heart['Max HR'][heart["Chest pain type"] == "non-anginal pain"]
maxhr_atypical_angina = heart['Max HR'][heart["Chest pain type"] == "atypical angina"]
maxhr_typical_angina = heart['Max HR'][heart["Chest pain type"] == "typical angina"]
```

In [87]:
```
print(maxhr_typical_angina.head())
```

```
13    145
18    144
19    178
37    125
63    171
Name: Max HR, dtype: int64
```

In [89]:
```
from scipy.stats import f_oneway

fstat, pval = f_oneway(maxhr_asymptomatic, maxhr_non_anginal, maxhr_atypical_angina, maxhr
print("P-Value for ANOVA is " + str(pval))
```

```
P-Value for ANOVA is 4.219911049988753e-08
```

In [94]:
```
#Following the ANOVA test, which pairs of variables are different.
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey_results = pairwise_tukeyhsd(heart['Max HR'], heart['Chest pain type'], 0.05)
print(tukey_results)
#If reject is TRUE we conclude that there is a significant difference between those groups
```

```
           Multiple Comparison of Means - Tukey HSD, FWER=0.05
=========================================================================
     group1           group2      meandiff p-adj   lower    upper  reject
-------------------------------------------------------------------------
    asymptomatic  atypical angina  20.3461  0.001  10.3667 30.3254   True
    asymptomatic non-anginal pain  14.1782  0.001   6.1532 22.2032   True
    asymptomatic   typical angina  15.6984 0.0153   2.1993 29.1976   True
 atypical angina non-anginal pain  -6.1679 0.4484 -16.8949  4.5591  False
 atypical angina   typical angina  -4.6476 0.8434 -19.9085 10.6133  False
non-anginal pain   typical angina   1.5203    0.9 -12.5407 15.5812  False
-------------------------------------------------------------------------
```

In [95]:

```python
#Chi-Square Test. Are the outcomes of two categorical variables associated?
table = pd.crosstab(heart['Gender'], heart['Chest pain type'])
```

In [99]:

```python
from scipy.stats import chi2_contingency

chi2, pval, dof, expected = chi2_contingency(table)
print("The P-Value for the Chi2 is " + str(pval))
```

```
The P-Value for the Chi2 is 0.10947278040318617
```

In [ ]: