In [27]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

In [5]:
```
import os
os.chdir("C:/Users/admin/Desktop/DATASET")
```

In [6]:
```
df=pd.read_csv('Heart_Disease_Prediction.csv')
```

In [7]:
```
df
```

Out[7]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels flurc
0	70	1	4	130	322	0	2	109	0	2.4	2	3
1	67	0	3	115	564	0	2	160	0	1.6	2	0
2	57	1	2	124	261	0	0	141	0	0.3	1	0
3	64	1	4	128	263	0	0	105	1	0.2	2	1
4	74	0	2	120	269	0	2	121	1	0.2	1	1
...	...	...	...	...	...	...	...	...	...	...	...	..
265	52	1	3	172	199	1	0	162	0	0.5	1	0
266	44	1	2	120	263	0	0	173	0	0.0	1	0
267	56	0	2	140	294	0	2	153	0	1.3	2	0
268	57	1	4	140	192	0	0	148	0	0.4	2	0
269	67	1	4	160	286	0	2	108	1	1.5	2	3

270 rows × 14 columns

In [8]:
```
df.describe()
```

Out[8]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Ma
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.0(
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.67
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.16
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.0(
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.0(
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.5(
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.0(
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.0(

In [9]:
```
df.columns
```

Out[9]:
```
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')
```

In [11]:
```
df.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
```

In [12]:
```
df.head()
```

Out[12]:

	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
0	70	1	4	130	322	0	2	109	0	2.4	2	3
1	67	0	3	115	564	0	2	160	0	1.6	2	0
2	57	1	2	124	261	0	0	141	0	0.3	1	0
3	64	1	4	128	263	0	0	105	1	0.2	2	1
4	74	0	2	120	269	0	2	121	1	0.2	1	1

In [13]:
```
df.tail()
```

Out[13]:

	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
265	52	1	3	172	199	1	0	162	0	0.5	1	0
266	44	1	2	120	263	0	0	173	0	0.0	1	0
267	56	0	2	140	294	0	2	153	0	1.3	2	0
268	57	1	4	140	192	0	0	148	0	0.4	2	0
269	67	1	4	160	286	0	2	108	1	1.5	2	3

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
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
```

In [15]:
```
df.corr()
```

Out[15]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR
Age	1.000000	-0.094401	0.096920	0.273053	0.220056	0.123458	0.128171	-0.402215
Sex	-0.094401	1.000000	0.034636	-0.062693	-0.201647	0.042140	0.039253	-0.076101
Chest pain type	0.096920	0.034636	1.000000	-0.043196	0.090465	-0.098537	0.074325	-0.317682
BP	0.273053	-0.062693	-0.043196	1.000000	0.173019	0.155681	0.116157	-0.039136
Cholesterol	0.220056	-0.201647	0.090465	0.173019	1.000000	0.025186	0.167652	-0.018739
FBS over 120	0.123458	0.042140	-0.098537	0.155681	0.025186	1.000000	0.053499	0.022494
EKG results	0.128171	0.039253	0.074325	0.116157	0.167652	0.053499	1.000000	-0.074628
Max HR	-0.402215	-0.076101	-0.317682	-0.039136	-0.018739	0.022494	-0.074628	1.000000
Exercise angina	0.098297	0.180022	0.353160	0.082793	0.078243	-0.004107	0.095098	-0.380719
ST depression	0.194234	0.097412	0.167244	0.222800	0.027709	-0.025538	0.120034	-0.349045
Slope of ST	0.159774	0.050545	0.136900	0.142472	-0.005755	0.044076	0.160614	-0.386847
Number of vessels fluro	0.356081	0.086830	0.225890	0.085697	0.126541	0.123774	0.114368	-0.265333
Thallium	0.106100	0.391046	0.262659	0.132045	0.028836	0.049237	0.007337	-0.253397

In [16]:
```
df.cov()
```

Out[16]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Ma
Age	82.975093	-0.402602	0.838786	44.426394	103.605452	0.400248	1.165056	-84.8
Sex	-0.402602	0.219207	0.015407	-0.524287	-4.879719	0.007022	0.018340	-0.8
Chest pain type	0.838786	0.015407	0.902671	-0.733044	4.442434	-0.033320	0.070467	-6.9
BP	44.426394	-0.524287	-0.733044	319.037051	159.731185	0.989674	2.070384	-16.1
Cholesterol	103.605452	-4.879719	4.442434	159.731185	2671.467107	0.463307	8.647005	-22.4
FBS over 120	0.400248	0.007022	-0.033320	0.989674	0.463307	0.126669	0.019000	0.1
EKG results	1.165056	0.018340	0.070467	2.070384	8.647005	0.019000	0.995787	-1.7
Max HR	-84.874721	-0.825403	-6.992028	-16.193432	-22.437340	0.185461	-1.725155	536.6
Exercise angina	0.421685	0.039694	0.158020	0.696448	1.904557	-0.000688	0.044692	-4.1
ST depression	2.026208	0.052230	0.181970	4.557435	1.640149	-0.010409	0.137175	-9.2
Slope of ST	0.894176	0.014539	0.079912	1.563486	-0.182762	0.009638	0.098472	-5.5
Number of vessels fluro	3.061586	0.038373	0.202575	1.444816	6.173510	0.041581	0.107724	-5.8
Thallium	1.875589	0.355308	0.484290	4.577117	2.892414	0.034008	0.014209	-11.3

In [17]:
```
df.dtypes
```

Out[17]:
```
Age                        int64
Sex                        int64
Chest pain type            int64
BP                         int64
Cholesterol                int64
FBS over 120               int64
EKG results                int64
Max HR                     int64
Exercise angina            int64
ST depression            float64
Slope of ST                int64
Number of vessels fluro    int64
Thallium                   int64
Heart Disease             object
dtype: object
```

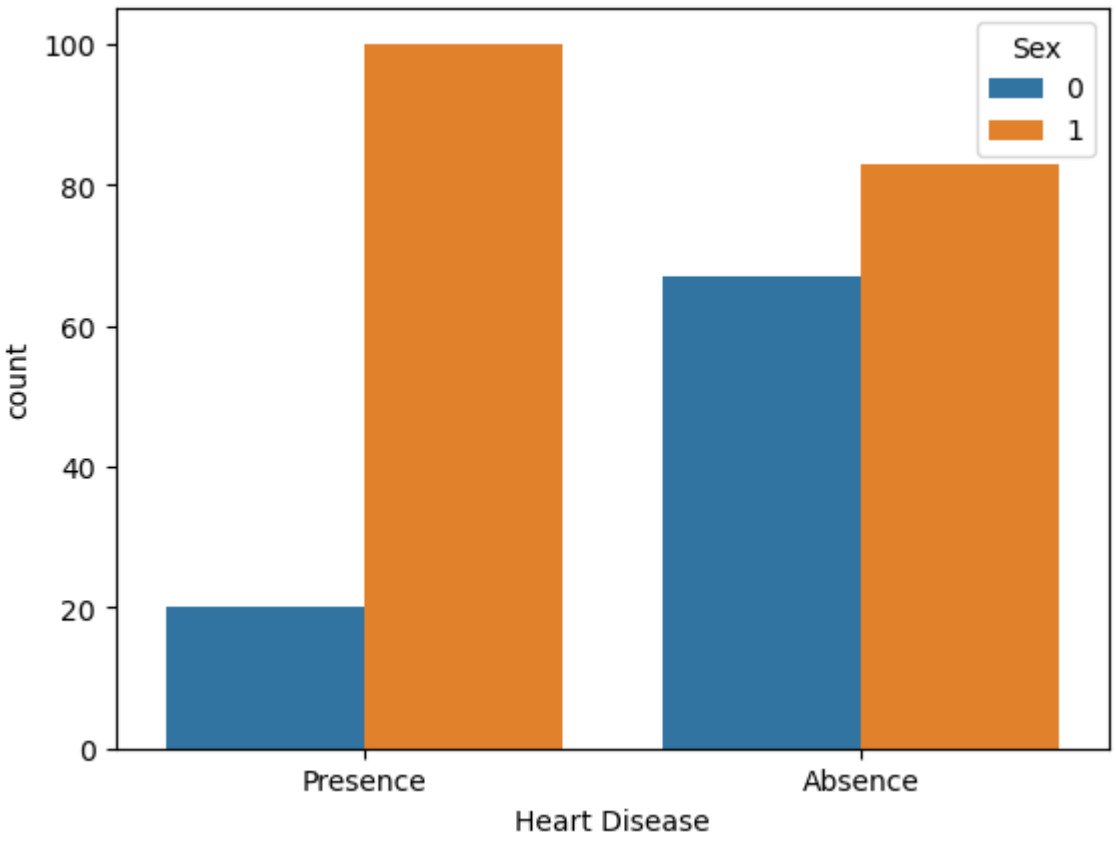
In [18]:
```
df.shape
```

Out[18]:
```
(270, 14)
```

In [19]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

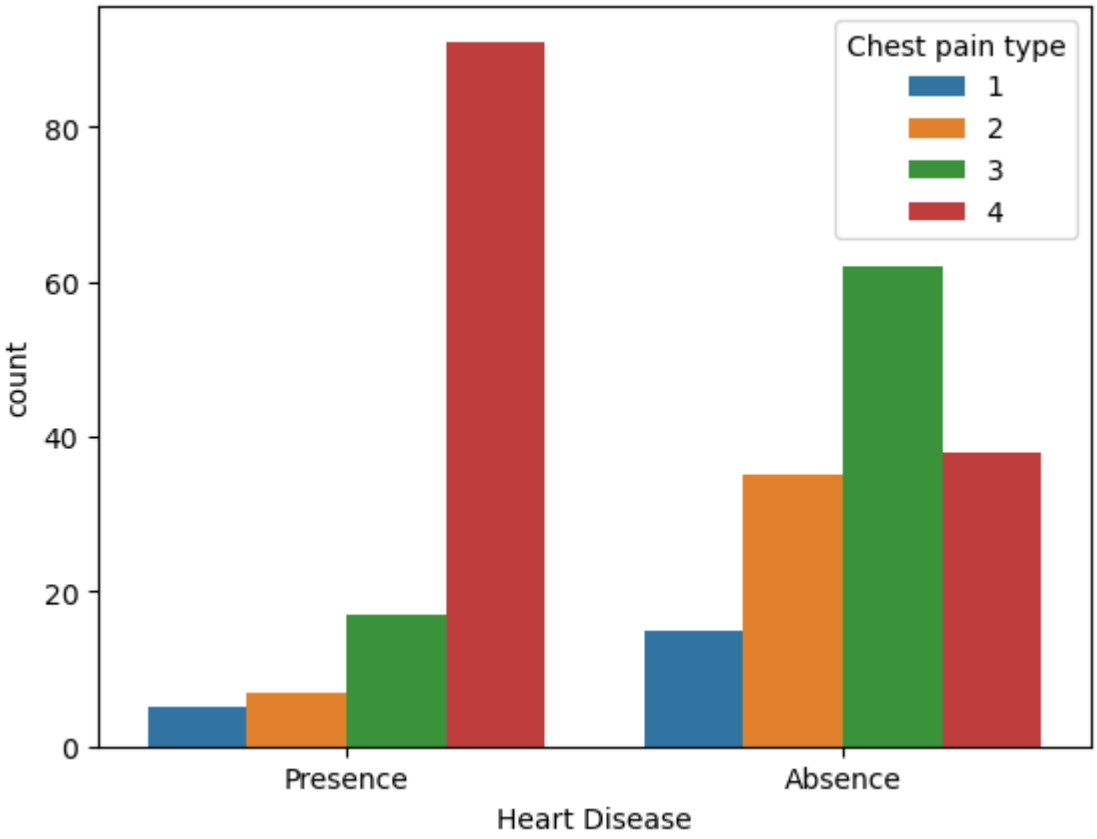
In [35]:
```
sns.countplot(x=df['Heart Disease'],hue='Sex',data=df)
```

Out[35]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='count'>
```



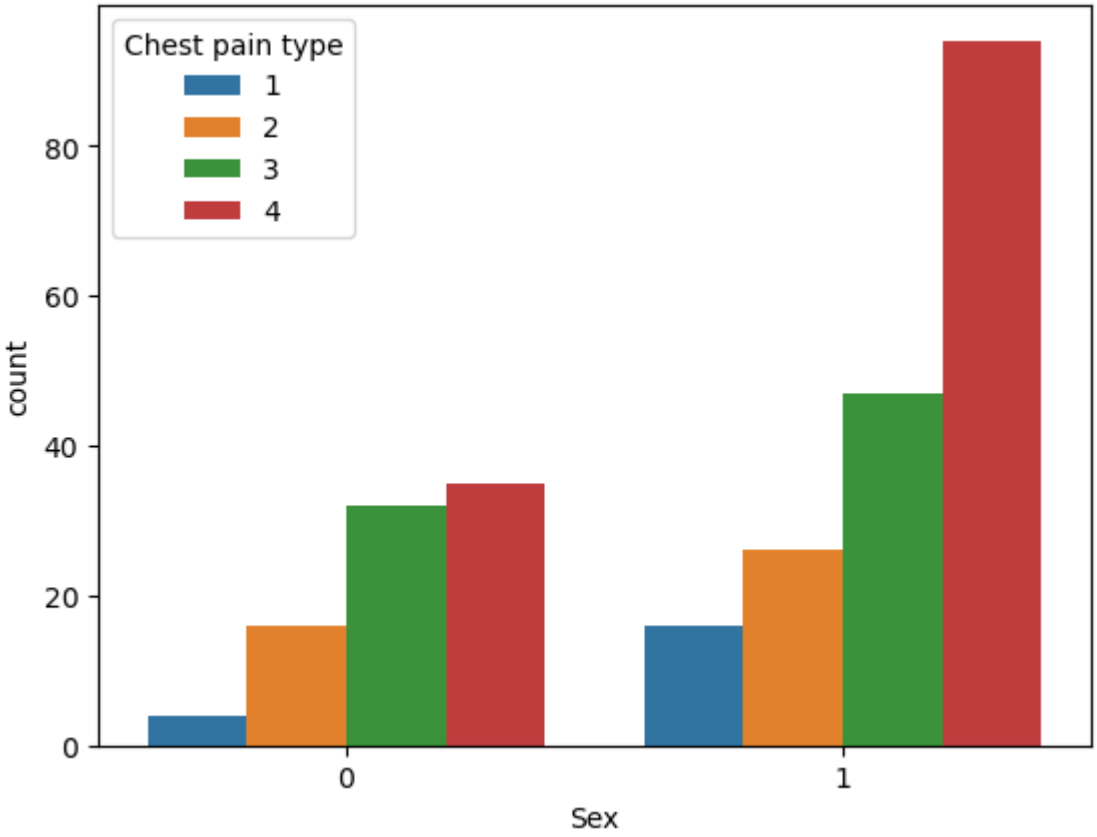
In [36]:
```
sns.countplot(x=df['Heart Disease'],hue='Chest pain type',data=df)
```

Out[36]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='count'>
```



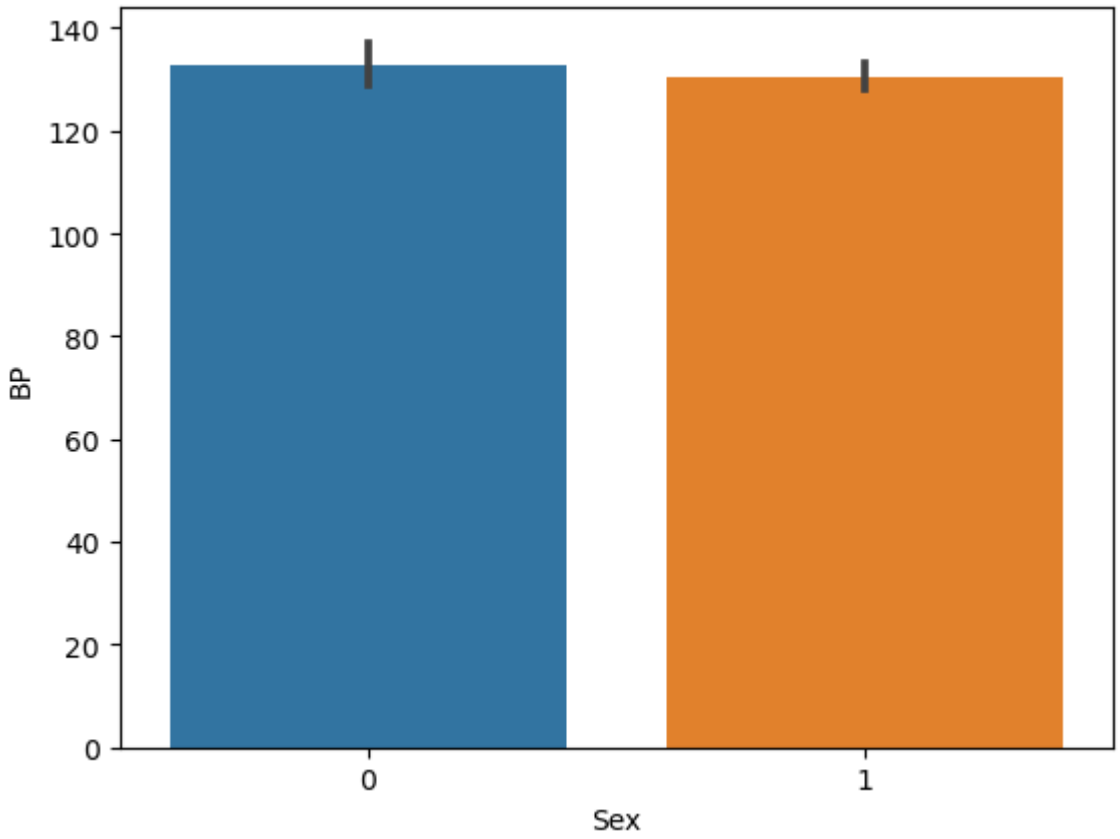
In [37]:
```
sns.countplot(x=df['Sex'],hue='Chest pain type',data=df)
```

Out[37]:
```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```



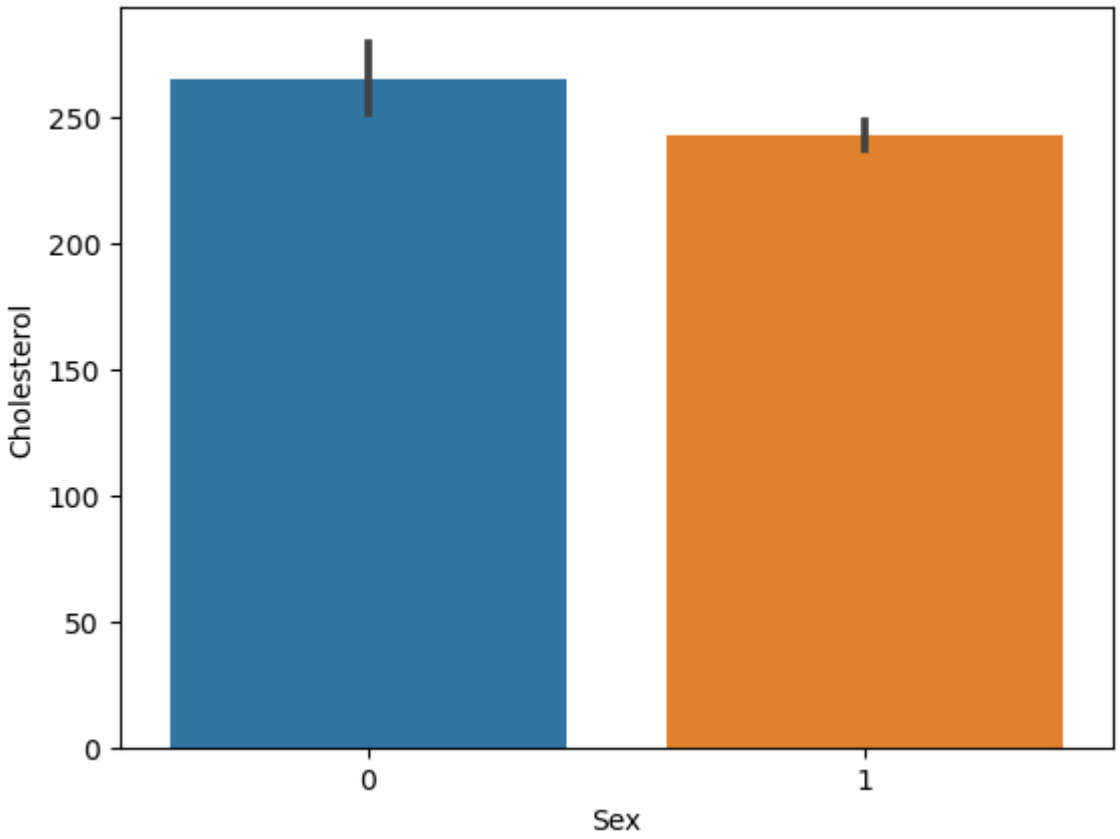
In [38]:
```
sns.barplot(x=df['Sex'],y=df['BP'],data=df)
```

Out[38]:
```
<AxesSubplot:xlabel='Sex', ylabel='BP'>
```



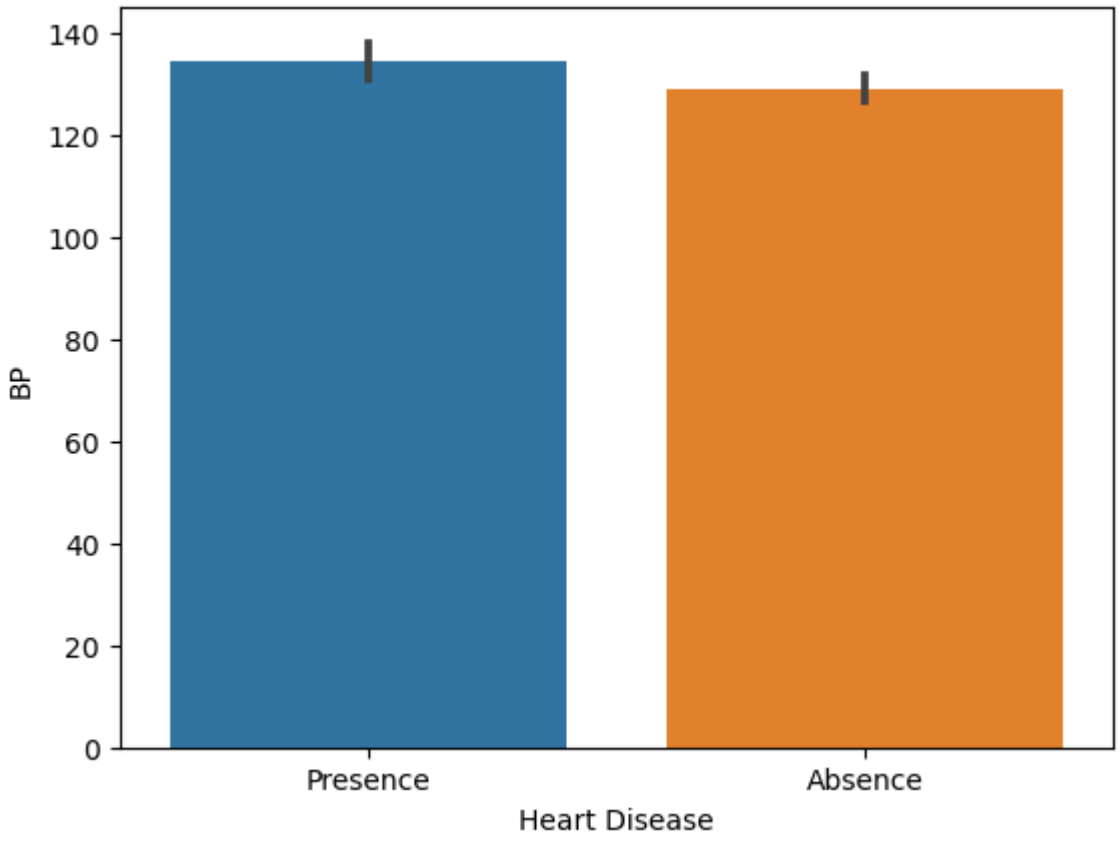
In [39]:
```
sns.barplot(x=df['Sex'],y=df['Cholesterol'],data=df)
```

Out[39]:
```
<AxesSubplot:xlabel='Sex', ylabel='Cholesterol'>
```



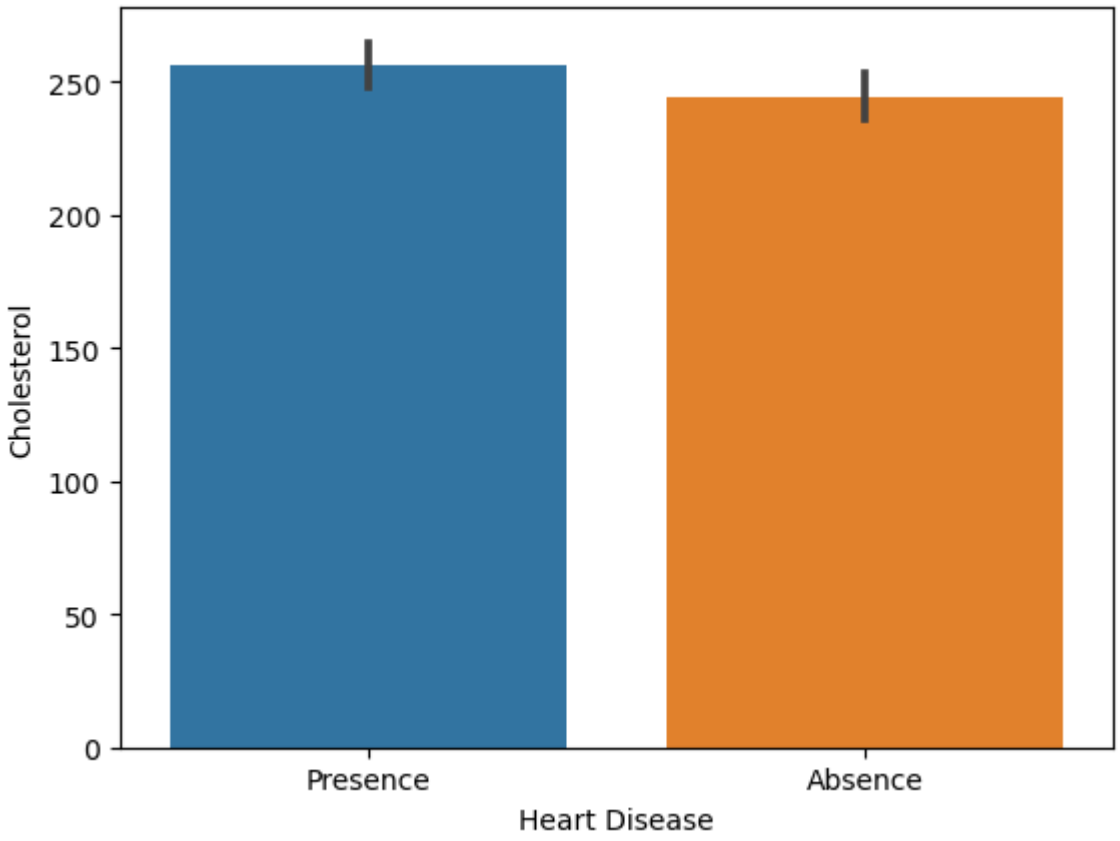
In [42]:
```
sns.barplot(x=df['Heart Disease'],y=df['BP'],data=df)
```

Out[42]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='BP'>
```



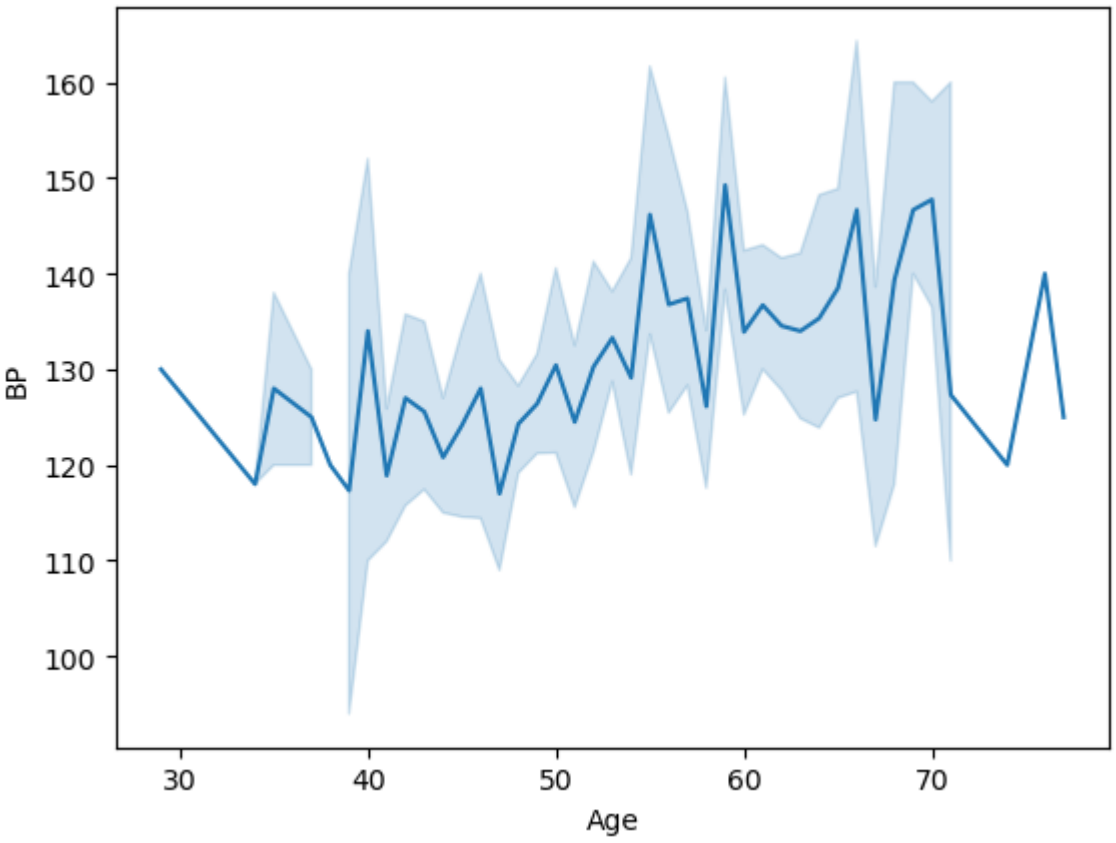
In [43]:
```
sns.barplot(x=df['Heart Disease'],y=df['Cholesterol'],data=df)
```

Out[43]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='Cholesterol'>
```



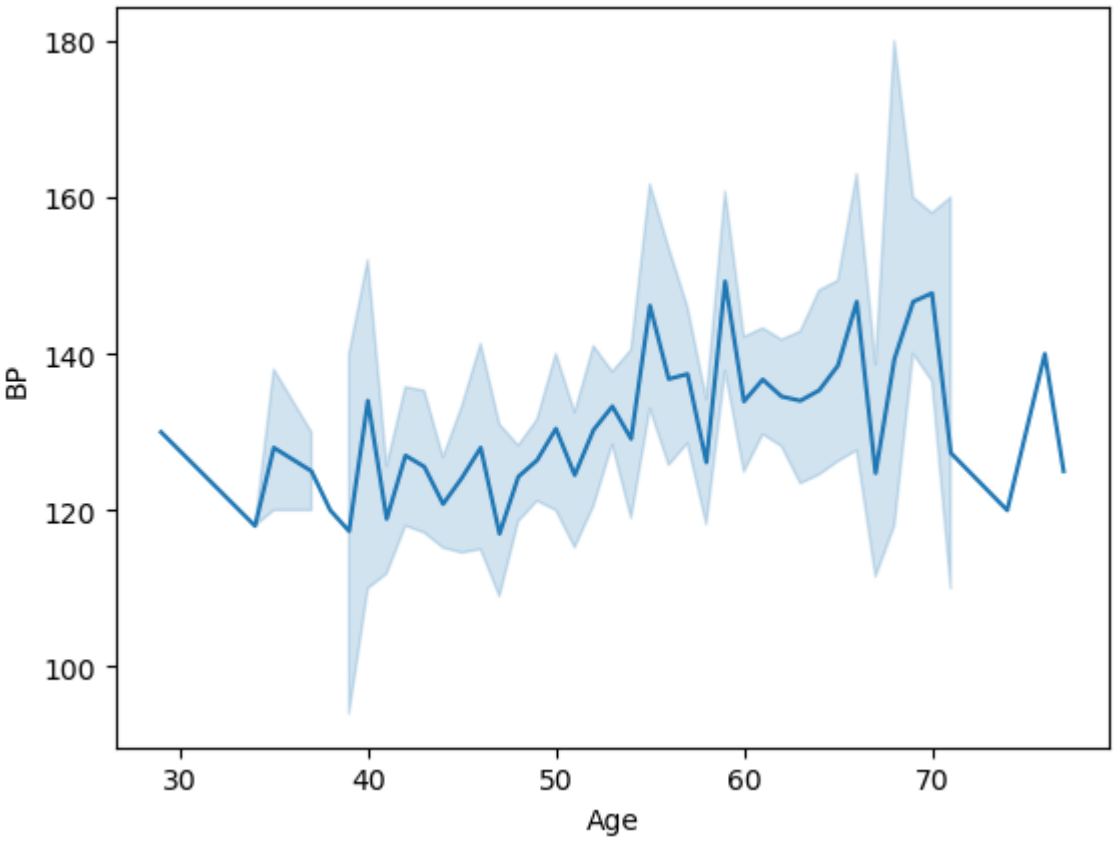
In [44]:
```
sns.lineplot(x=df['Age'],y=df['BP'],data=df)
```

Out[44]:
```
<AxesSubplot:xlabel='Age', ylabel='BP'>
```



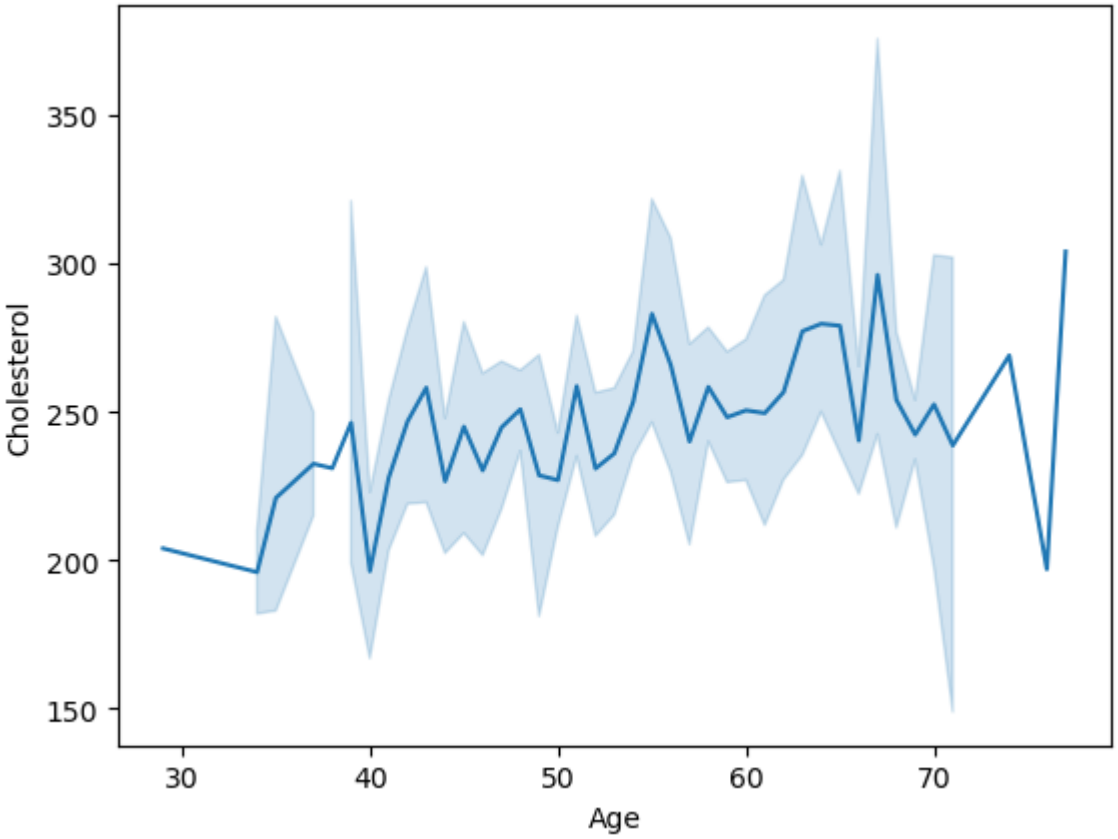
In [45]:
```
sns.lineplot(x=df['Age'],y=df['BP'],data=df)
```

Out[45]:
```
<AxesSubplot:xlabel='Age', ylabel='BP'>
```



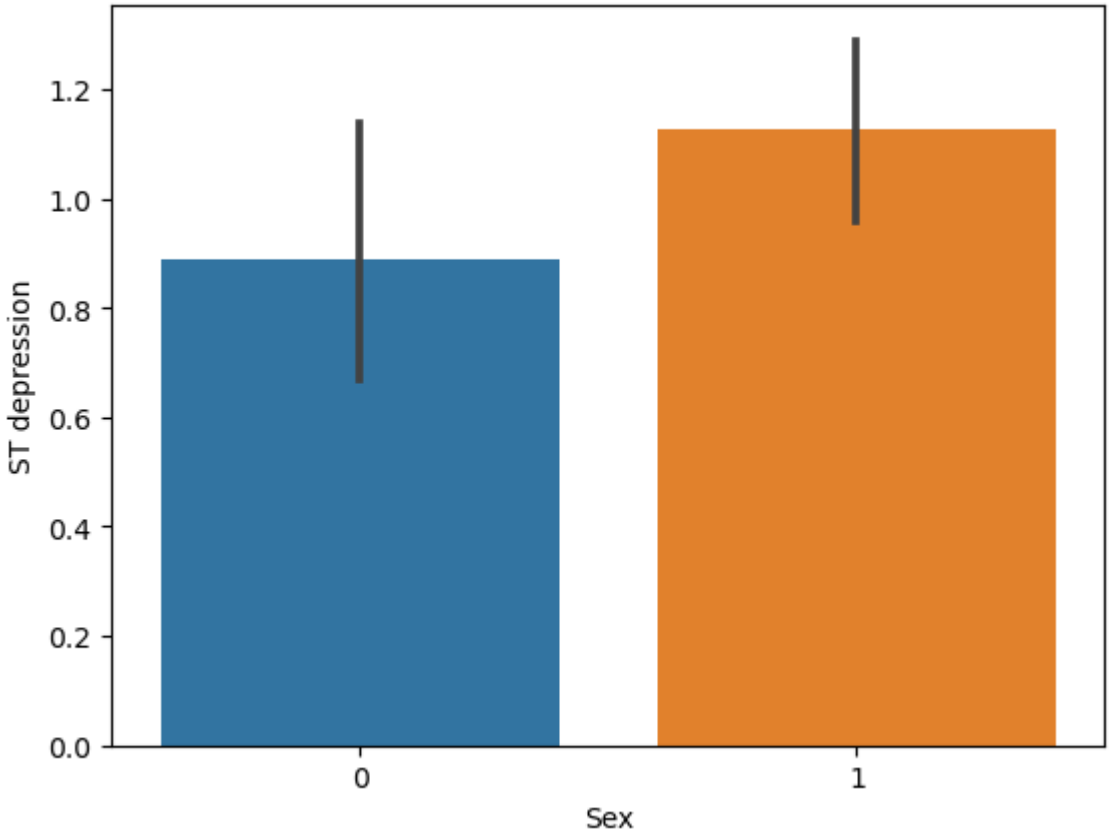
In [46]:
```
sns.lineplot(x=df['Age'],y=df['Cholesterol'],data=df)
```

Out[46]:
```
<AxesSubplot:xlabel='Age', ylabel='Cholesterol'>
```



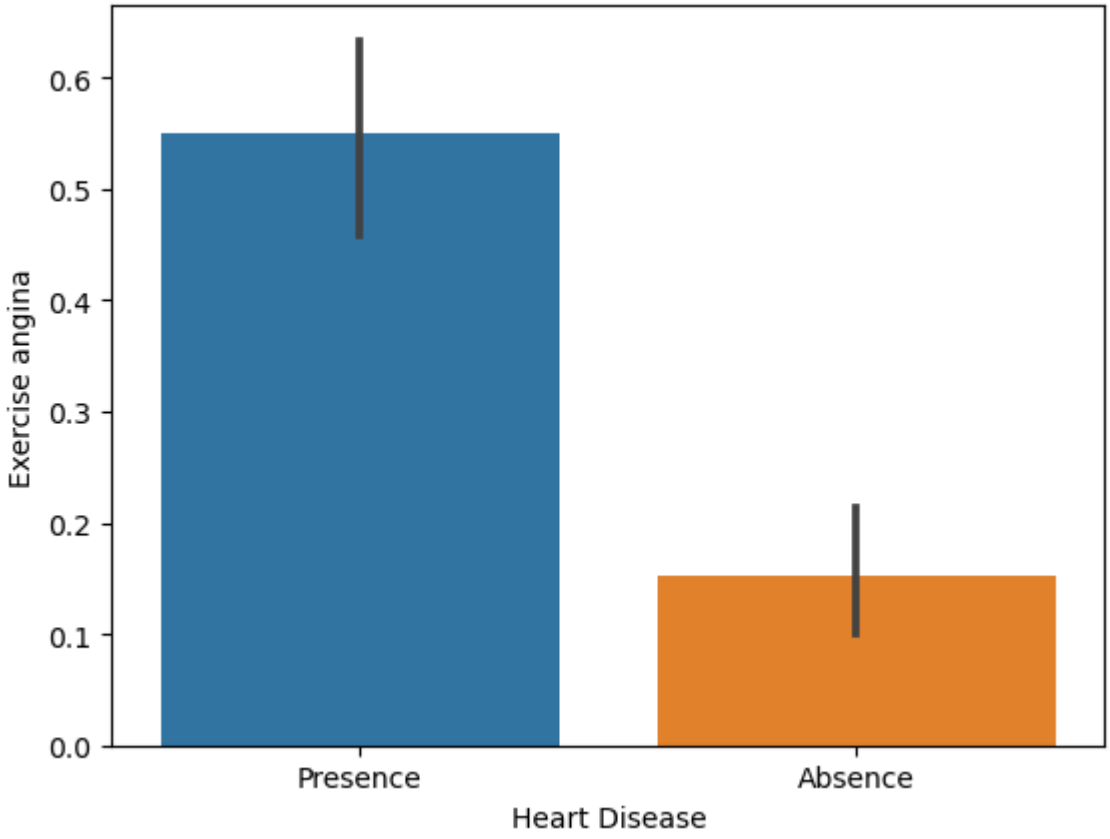
In [47]:
```
sns.barplot(x=df['Sex'],y=df['ST depression'],data=df)
```

Out[47]:
```
<AxesSubplot:xlabel='Sex', ylabel='ST depression'>
```



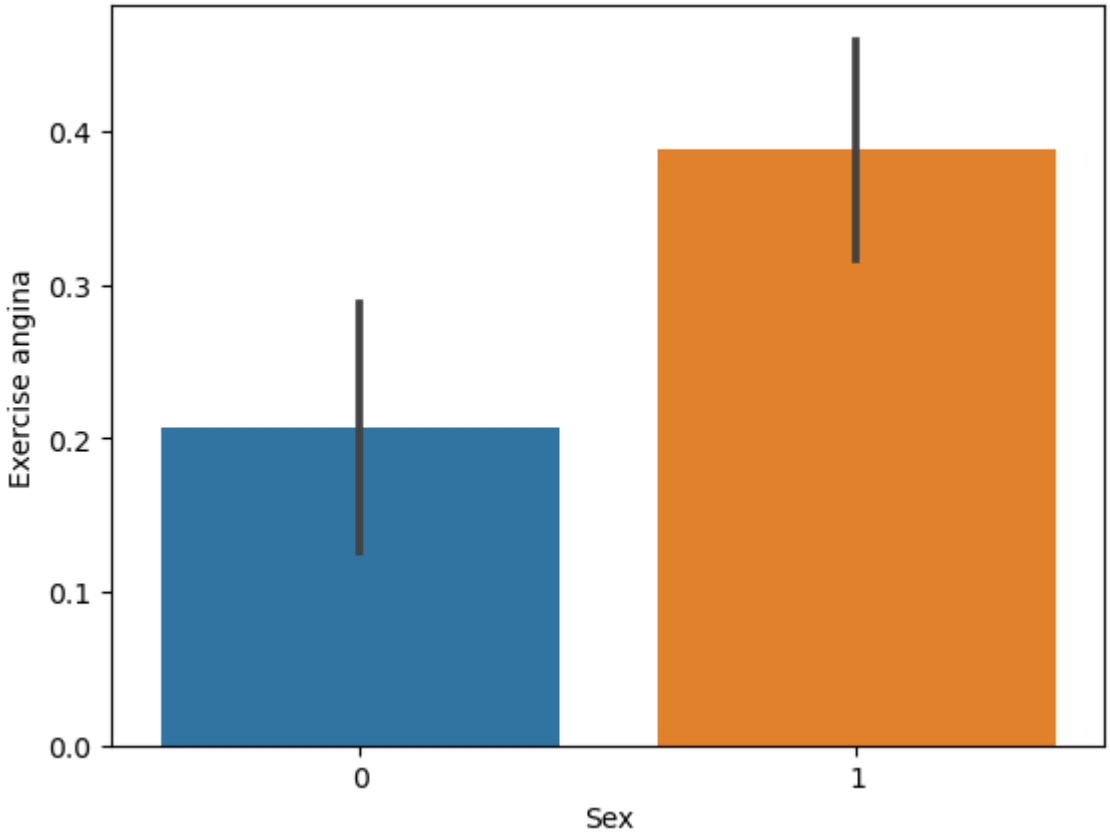
In [48]:
```
sns.barplot(x=df['Heart Disease'],y=df['Exercise angina'],data=df)
```

Out[48]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='Exercise angina'>
```



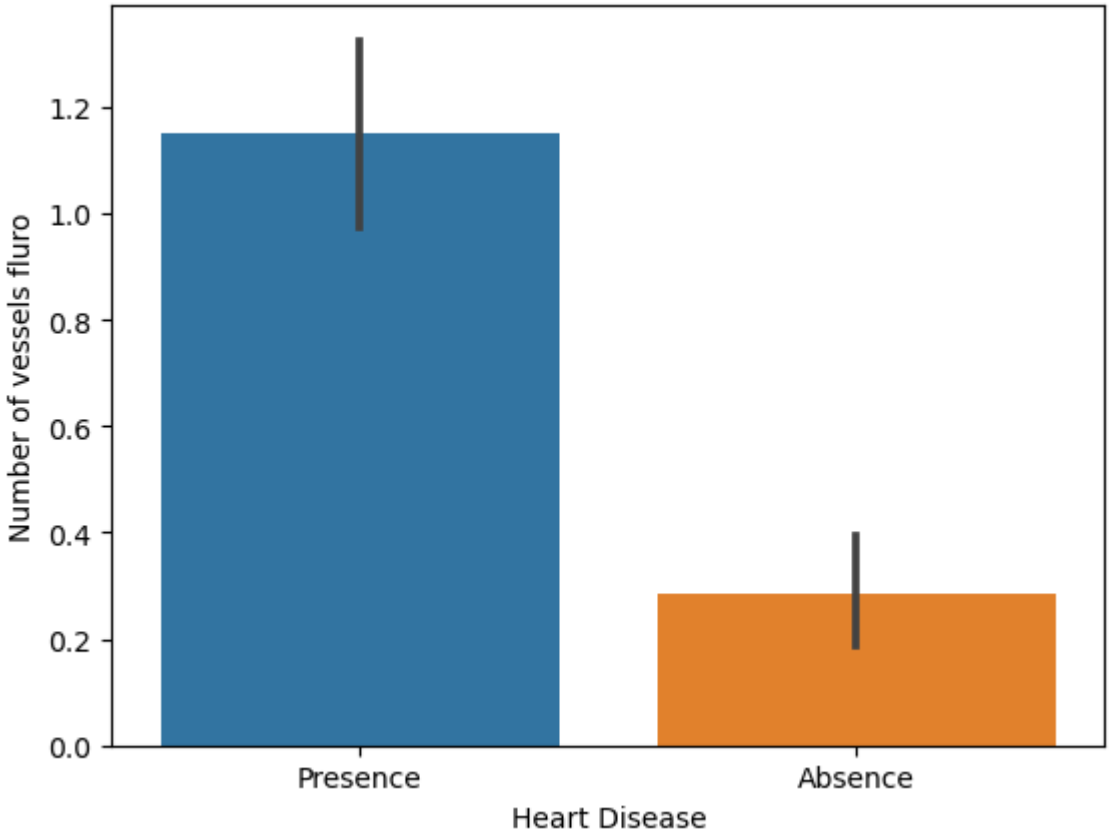
In [49]:
```
sns.barplot(x=df['Sex'],y=df['Exercise angina'],data=df)
```

Out[49]:
```
<AxesSubplot:xlabel='Sex', ylabel='Exercise angina'>
```



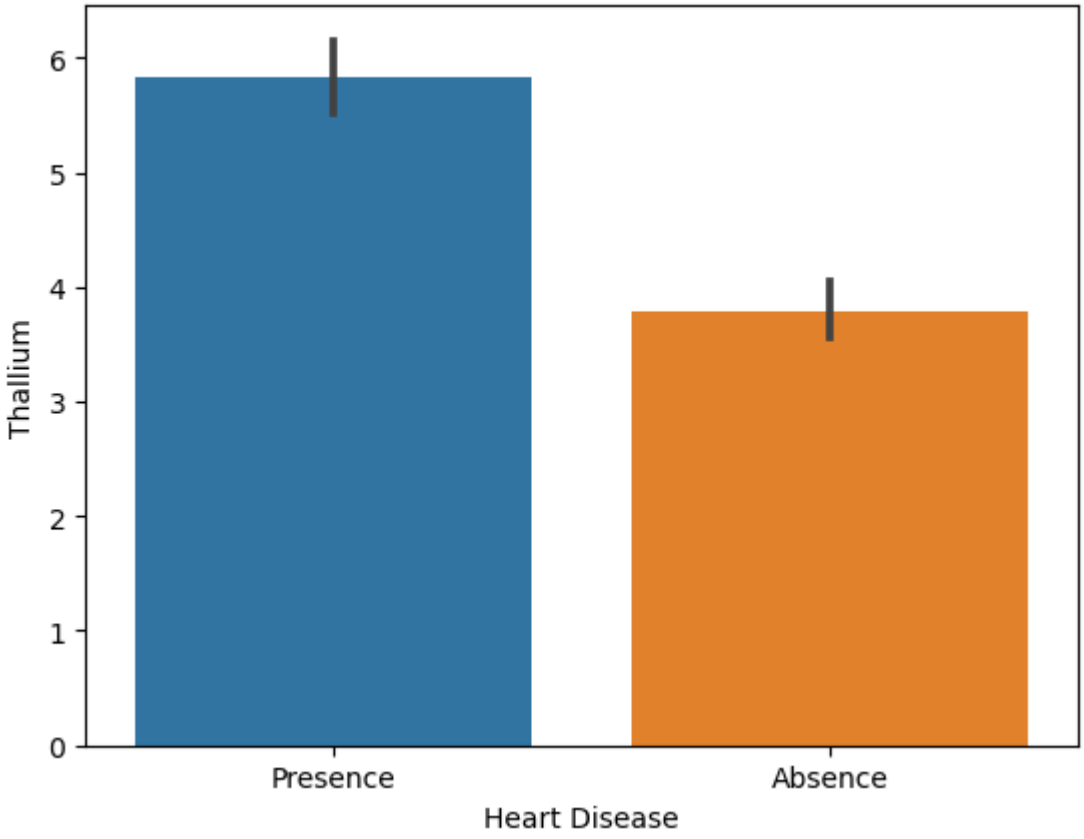
In [50]:
```
sns.barplot(x=df['Heart Disease'],y=df['Number of vessels fluro'],data=df)
```

Out[50]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='Number of vessels fluro'>
```



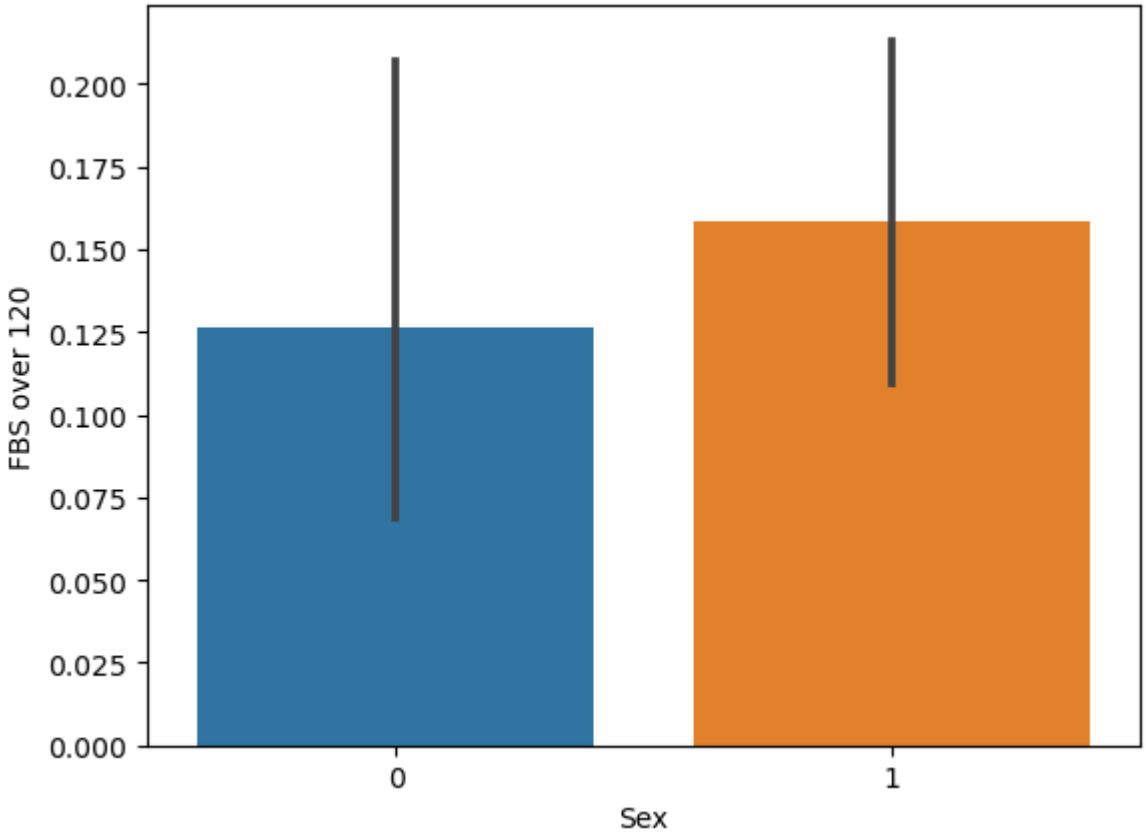
In [51]:
```
sns.barplot(x=df['Heart Disease'],y=df['Thallium'],data=df)
```

Out[51]:
```
<AxesSubplot:xlabel='Heart Disease', ylabel='Thallium'>
```



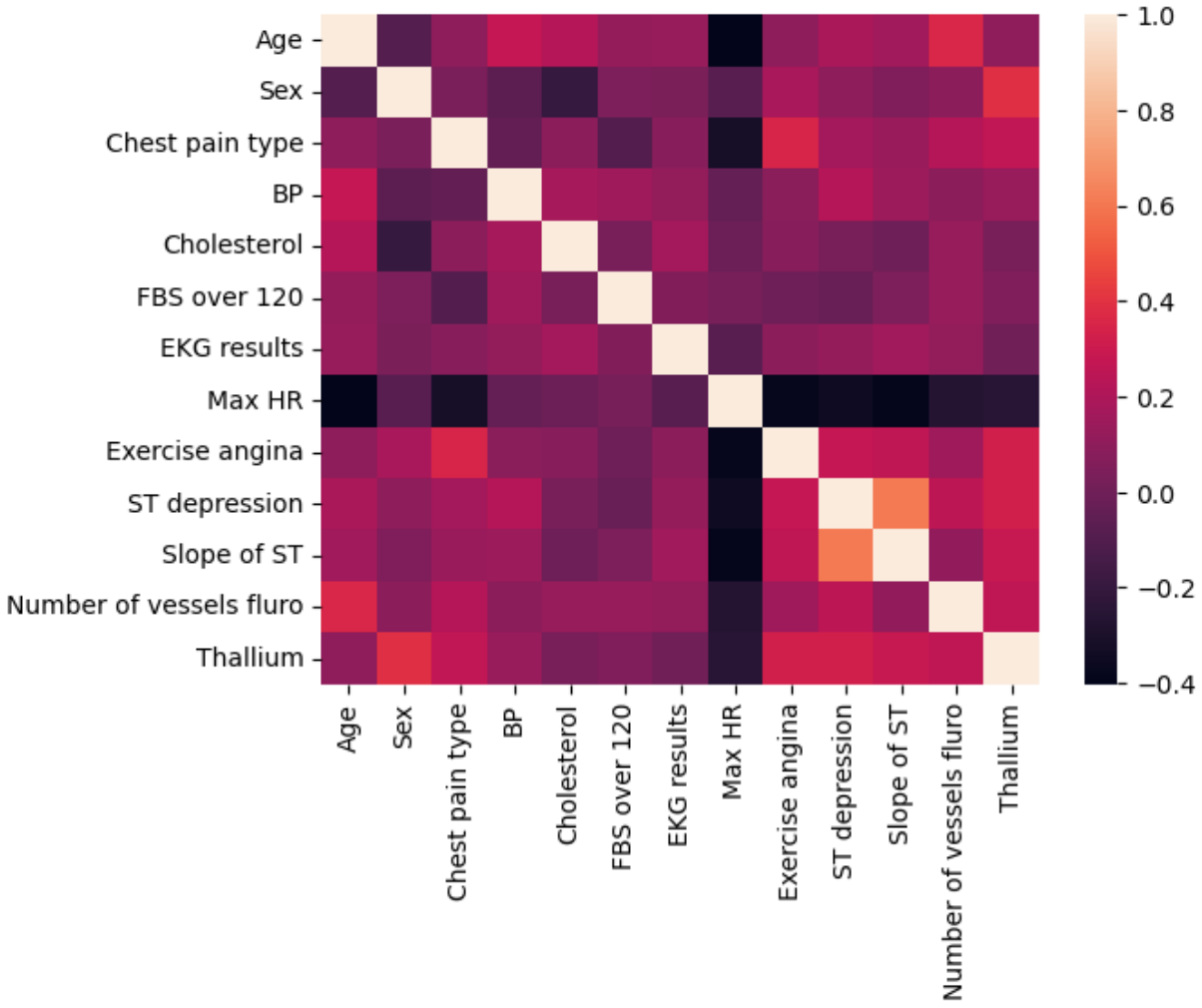
In [52]:
```
sns.barplot(x=df['Sex'],y=df['FBS over 120'],data=df)
```

Out[52]:
```
<AxesSubplot:xlabel='Sex', ylabel='FBS over 120'>
```



In [53]:
```
sns.heatmap(df.corr())
```

Out[53]: <AxesSubplot:>



In [54]:
```
from sklearn.preprocessing import LabelEncoder,StandardScaler
le=LabelEncoder()
df['Heart Disease']=le.fit_transform(df['Heart Disease'])
```

In [55]:
```
y=df['Heart Disease']
x=df.drop(['Heart Disease'],axis=1)
```

In [56]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0,test_size=0.2)
```

In [57]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
lr=LogisticRegression(max_iter=10000)
lr.fit(x_train,y_train)
pred_1=lr.predict(x_test)
score_1=accuracy_score(y_test,pred_1)
```

In [58]:
```
score_1
```

Out[58]:
```
0.7777777777777778
```

In [59]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
pred_2=rfc.predict(x_test)
score_2=accuracy_score(y_test,pred_2)
```

In [60]:
```
score_2
```

Out[60]:
```
0.7592592592592593
```

In [64]:
```
max(list_1)
```

Out[64]:
```
0.7037037037037037
```

In [ ]: