In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

import os
print(os.listdir())

import warnings
warnings.filterwarnings('ignore')
```

```
['.anaconda', '.android', '.atom', '.bash_history', '.conda', '.condarc', '.con
tinuum', '.gitconfig', '.idlerc', '.ipynb_checkpoints', '.ipython', '.jupyter',
'.matplotlib', '.ms-ad', '.virtualenvs', '3D Objects', 'abalone.csv', 'anaconda
3', 'Anaconda3-2022.05-Windows-x86_64.exe', 'AppData', 'Application Data', 'ass
ignment 4.ipynb', 'auducity', 'ChromeSetup.exe', 'Contacts', 'Cookies', 'Creati
ve Cloud Files', 'd4ac4633ebd6440fa397b84f1bc94a3c.7z', 'DA_Assignment_3_Pytho
n.ipynb', 'Documents', 'Downloads', 'Exploratory data analysis.ipynb', 'Favorit
es', 'filmora-idco_setup_full1901.exe', 'Git-2.37.3-64-bit.exe', 'Heart_Disease
_Prediction.csv', 'HP_M1130_M1210_MFP_Full_Solution-v20180815-10158769.exe', 'i
nittk.ini', 'inst.ini', 'kav21.1.15.500en_26175.exe', 'Links', 'Local Setting
s', 'MicrosoftEdgeBackups', 'Music', 'My Documents', 'my-venv-1', 'NetHood', 'N
ox_share', 'NTUSER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LOG2', 'NTUSER.DAT{e04c
8bc8-36e4-11ec-aaca-f0ebe5b2f12a}.TM.blf', 'NTUSER.DAT{e04c8bc8-36e4-11ec-aaca-
f0ebe5b2f12a}.TMContainer00000000000000000001.regtrans-ms', 'NTUSER.DAT{e04c8bc
8-36e4-11ec-aaca-f0ebe5b2f12a}.TMContainer00000000000000000002.regtrans-ms', 'n
tuser.ini', 'nuuid.ini', 'OneDrive', 'Pipfile', 'PrintHood', 'Recent', 'Saved G
ames', 'scikit_learn_data', 'Searches', 'SendTo', 'source', 'Start Menu', 'Temp
lates', 'Untitled.ipynb', 'Untitled1.ipynb', 'Untitled2.ipynb', 'Untitled3.ipyn
b', 'Untitled4.ipynb', 'useruid.ini', 'Videos', 'vMixDesktopCaptureNDI.zip', 'v
mlogs']
```

In [2]:
```
dataset = pd.read_csv("Heart_Disease_Prediction.csv")
```

In [3]:
```
type(dataset)
```

Out[3]:
```
pandas.core.frame.DataFrame
```

In [4]:
```
dataset.shape
```

Out[4]:
```
(270, 14)
```

In [5]:
```
1 dataset.head(5)
```

Out[5]:

	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 [6]:
```
1 dataset.sample(5)
```

Out[6]:

	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
43	46	1	2	101	197	1	0	156	0	0.0	1	0
175	62	0	4	138	294	1	0	106	0	1.9	2	3
24	54	0	2	132	288	1	2	159	1	0.0	1	1
99	50	0	2	120	244	0	0	162	0	1.1	1	0
143	51	1	1	125	213	0	2	125	1	1.4	1	1

In [7]:
```
1 dataset.describe()
```

Out[7]:

	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	270.00
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	71.00
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.00
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.50
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.00
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.00

In [8]:
```
1 dataset.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 [9]:
```
info = ["age","1: male, 0: female","chest pain type, 1: typical angina, 2: a



for i in range(len(info)):
    print(dataset.columns[i]+":\t\t\t"+info[i])
```

```
Age:                    age
Sex:                    1: male, 0: female
Chest pain type:                        chest pain type, 1: typical angina, 2:
atypical angina, 3: non-anginal pain, 4: asymptomatic
BP:                     resting blood pressure
Cholesterol:                     serum cholestoral in mg/dl
FBS over 120:                   fasting blood sugar > 120 mg/dl
EKG results:                    resting electrocardiographic results (values 0,
1,2)
Max HR:                  maximum heart rate achieved
Exercise angina:                        exercise induced angina
ST depression:                  oldpeak = ST depression induced by exercise rel
ative to rest
Slope of ST:                    the slope of the peak exercise ST segment
Number of vessels fluro:                        number of major vessels (0-3) c
olored by flourosopy
Thallium:                       thal: 3 = normal; 6 = fixed defect; 7 = reversa
ble defect
```

In [18]:
```
dataset["Heart Disease"].describe()
```

Out[18]:
```
count         270
unique          2
top       Absence
freq          150
Name: Heart Disease, dtype: object
```

In [19]:
```
dataset["Heart Disease"].unique()
```

Out[19]:
```
array(['Presence', 'Absence'], dtype=object)
```

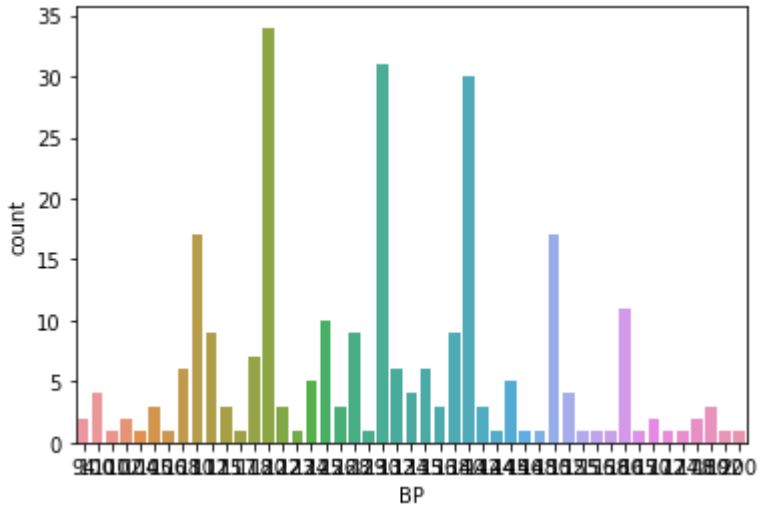
In [21]:
```
1 print(dataset.corr()["BP"].abs().sort_values(ascending=False))
```

```
BP                          1.000000
Age                         0.273053
ST depression               0.222800
Cholesterol                 0.173019
FBS over 120                0.155681
Slope of ST                 0.142472
Thallium                    0.132045
EKG results                 0.116157
Number of vessels fluro     0.085697
Exercise angina             0.082793
Sex                         0.062693
Chest pain type             0.043196
Max HR                      0.039136
Name: BP, dtype: float64
```

In [15]:
```
y = dataset["BP"]

sns.countplot(y)


target_temp = dataset.BP.value_counts()

print(target_temp)
```

```
120    34
130    31
140    30
110    17
150    17
160    11
125    10
138     9
112     9
128     9
118     7
132     6
135     6
108     6
124     5
145     5
134     4
100     4
152     4
180     3
136     3
115     3
105     3
126     3
122     3
142     3
94      2
102     2
178     2
170     2
104     1
200     1
155     1
101     1
129     1
123     1
192     1
117     1
158     1
146     1
156     1
165     1
174     1
144     1
106     1
148     1
```

```
172     1
Name: BP, dtype: int64
```



In [24]:
```
print("Percentage of patience with heart problems: "+str(y.where(y==1).count
print("Percentage of patience with heart problems: "+str(y.where(y==0).count

```

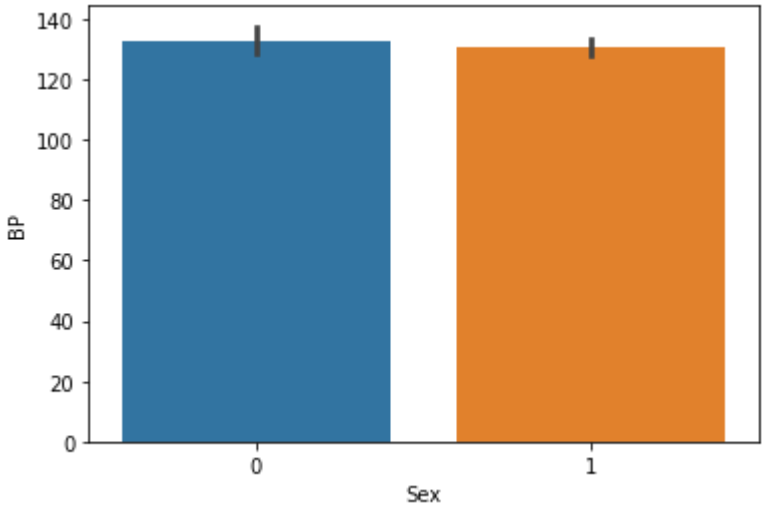
```
Percentage of patience with heart problems: 0.0
Percentage of patience with heart problems: 0.0
```

In [27]:
```
dataset["Sex"].unique()
```

Out[27]:
```
array([1, 0], dtype=int64)
```

In [28]:
```
sns.barplot(dataset["Sex"],y)
```

Out[28]:
```
<AxesSubplot:xlabel='Sex', ylabel='BP'>
```

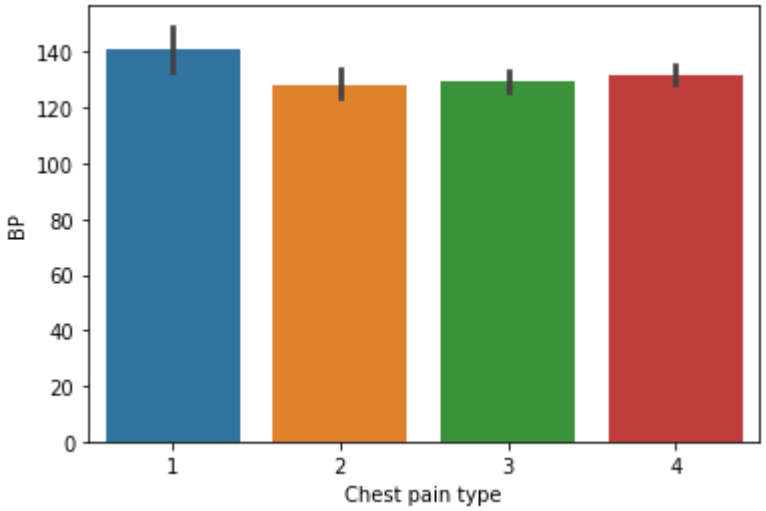


In [31]:
```
#Analysing the 'Chest Pain Type' feature
dataset["Chest pain type"].unique()
```

Out[31]:
```
array([4, 3, 2, 1], dtype=int64)
```

In [32]:
```
1 sns.barplot(dataset["Chest pain type"],y)
```

Out[32]: <AxesSubplot:xlabel='Chest pain type', ylabel='BP'>



In [34]:
```
1 dataset["FBS over 120"].describe()
```

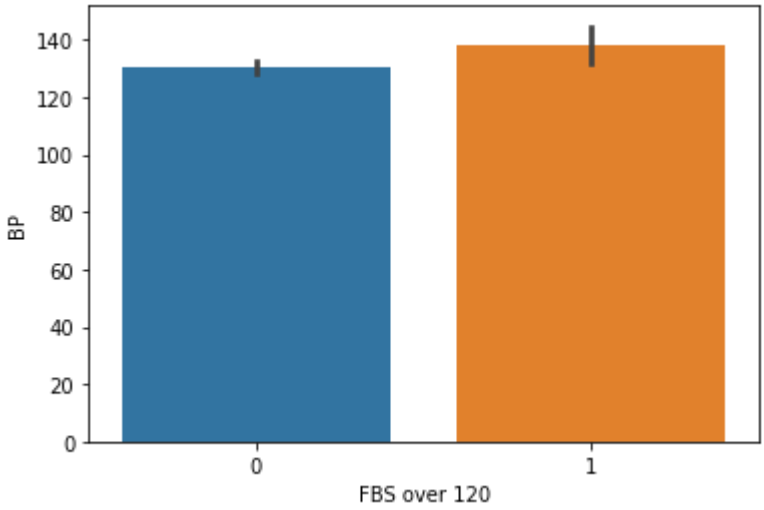
Out[34]:
```
count    270.000000
mean       0.148148
std        0.355906
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max        1.000000
Name: FBS over 120, dtype: float64
```

In [35]:
```
1 dataset["FBS over 120"].unique()
```

Out[35]:
```
array([0, 1], dtype=int64)
```

In [36]:
```
1 sns.barplot(dataset["FBS over 120"],y)
```

Out[36]: <AxesSubplot:xlabel='FBS over 120', ylabel='BP'>

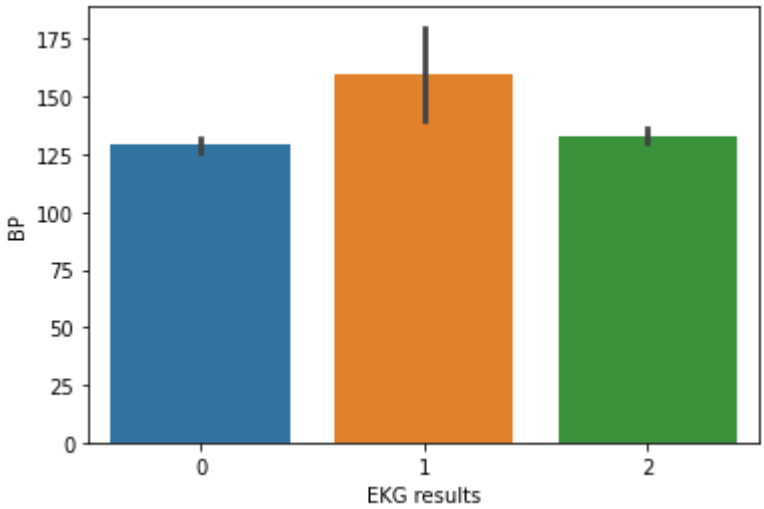


In [38]:
```
1 #Analysing the restecg feature
2 dataset["EKG results"].unique()
```

Out[38]: array([2, 0, 1], dtype=int64)

In [39]:
```
1 sns.barplot(dataset["EKG results"],y)
```

Out[39]: <AxesSubplot:xlabel='EKG results', ylabel='BP'>

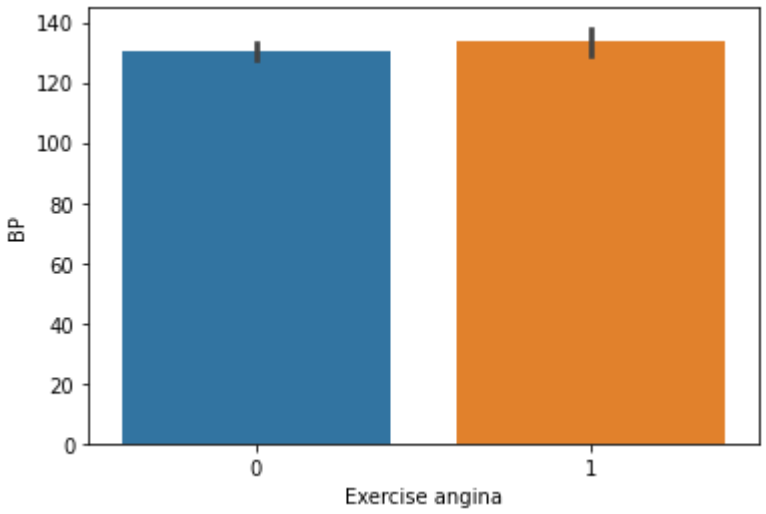


In [40]:
```
1 #Analysing the 'exang' feature
2 dataset["Exercise angina"].unique()
```

Out[40]: array([0, 1], dtype=int64)

In [41]:
```
1 sns.barplot(dataset["Exercise angina"],y)
```

Out[41]: <AxesSubplot:xlabel='Exercise angina', ylabel='BP'>

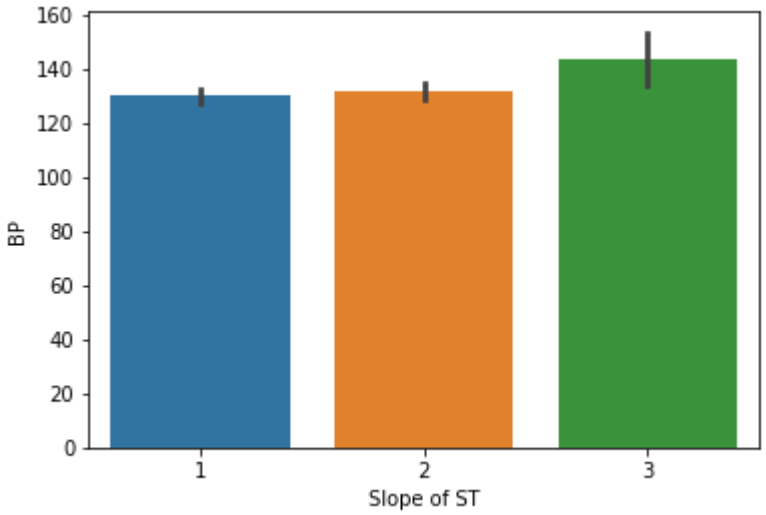


In [42]:
```
1 #Analysing the Slope feature
2 dataset["Slope of ST"].unique()
```

Out[42]: array([2, 1, 3], dtype=int64)

In [43]:
```
1 sns.barplot(dataset["Slope of ST"],y)
```

Out[43]: <AxesSubplot:xlabel='Slope of ST', ylabel='BP'>

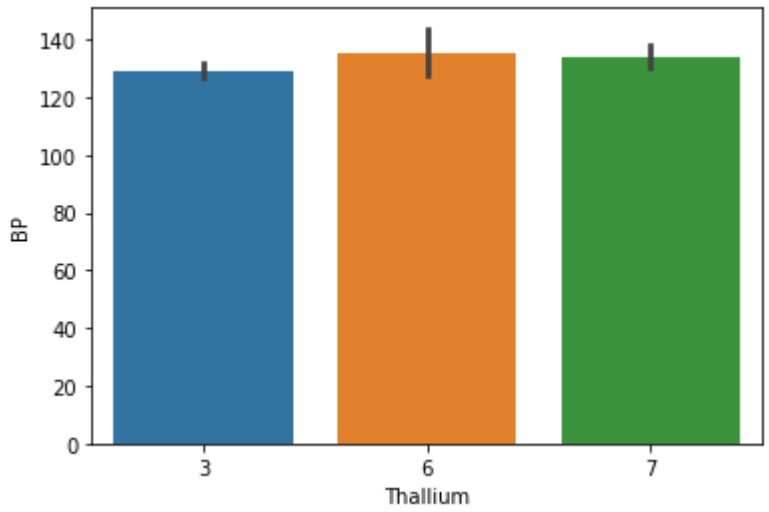


In [44]:
```
1 ### Analysing the 'thal' feature
2 dataset["Thallium"].unique()
```

Out[44]: array([3, 7, 6], dtype=int64)

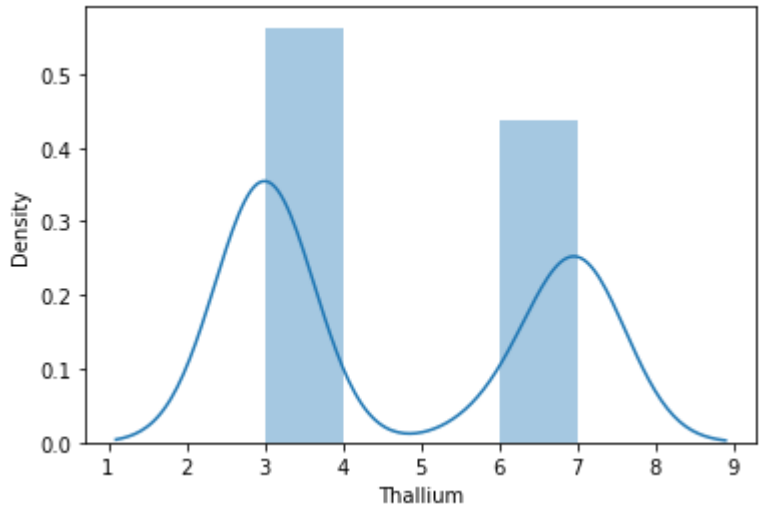
In [45]:
```
1 sns.barplot(dataset["Thallium"],y)
```

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



In [46]:
```
1 sns.distplot(dataset["Thallium"])
```

Out[46]: <AxesSubplot:xlabel='Thallium', ylabel='Density'>



In [51]:
```
1  from sklearn.model_selection import train_test_split
2
3  predictors = dataset.drop("BP",axis=1)
4  target = dataset["BP"]
5
6  X_train,X_test,Y_train,Y_test = train_test_split(predictors,target,test_size
7
8  from sklearn.model_selection import train_test_split
9
10 predictors = dataset.drop("BP",axis=1)
11 target = dataset["BP"]
12
13 X_train,X_test,Y_train,Y_test = train_test_split(predictors,target,test_size
```

In [52]:
```
1 X_train.shape
```

Out[52]: (216, 13)

In [53]:
```
1 X_test.shape
```

Out[53]: (54, 13)

In [54]:
```
1 Y_train.shape
```

Out[54]: (216,)

In [55]:
```
1 Y_test.shape
```

Out[55]: (54,)

In [ ]:
```
1
```