```
import numpy as np
import pandas as pd


import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))



df=pd.read_csv('/content/Heart_Disease_Prediction.csv')


df
```

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

270 rows × 14 columns

```
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
```

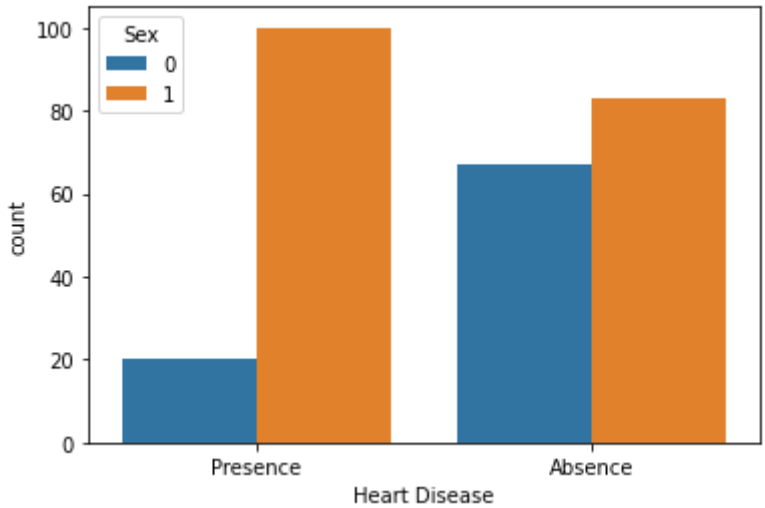
```
 6   EKG results              270 non-null    int64
 7   Max HR                   270 non-null    int64
 8   Exercise angina          270 non-null    int64
 9   ST depression            270 non-null    float64
 10  Slope of ST              270 non-null    int64
 11  Number of vessels fluro  270 non-null    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
```

```
df.isnull().sum()
```

```
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
```

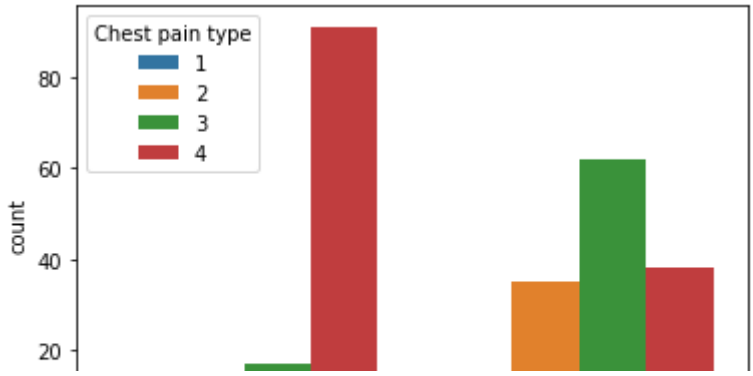
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x=df['Heart Disease'],hue='Sex',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6ee0510>
```



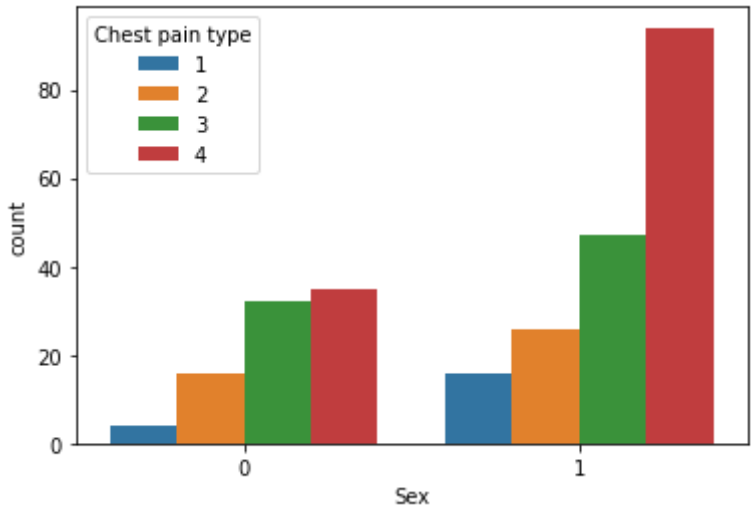

```
sns.countplot(x=df['Heart Disease'],hue='Chest pain type',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6c6f690>
```



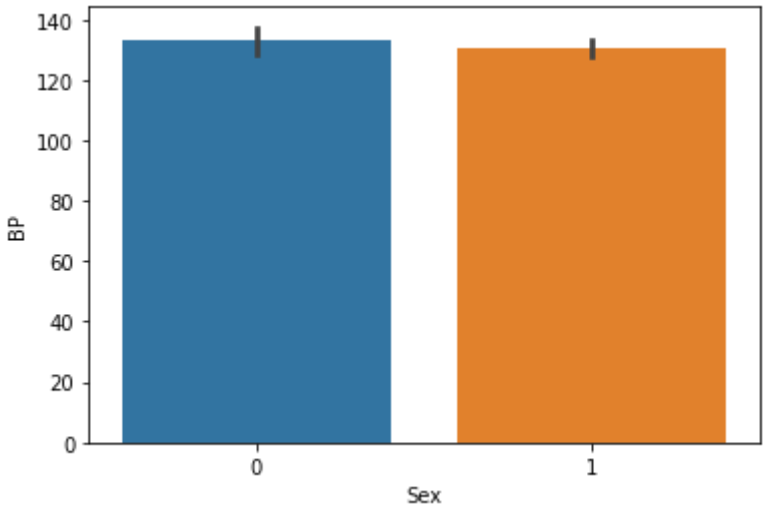

```
sns.countplot(x=df['Sex'],hue='Chest pain type',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d67c4690>
```



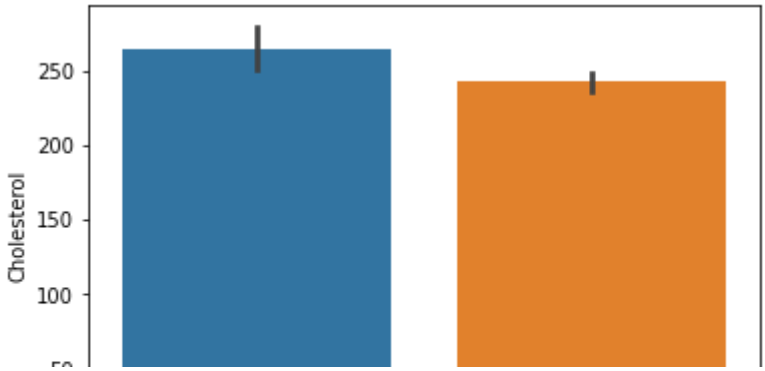

```
sns.barplot(x=df['Sex'],y=df['BP'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d67b2190>
```



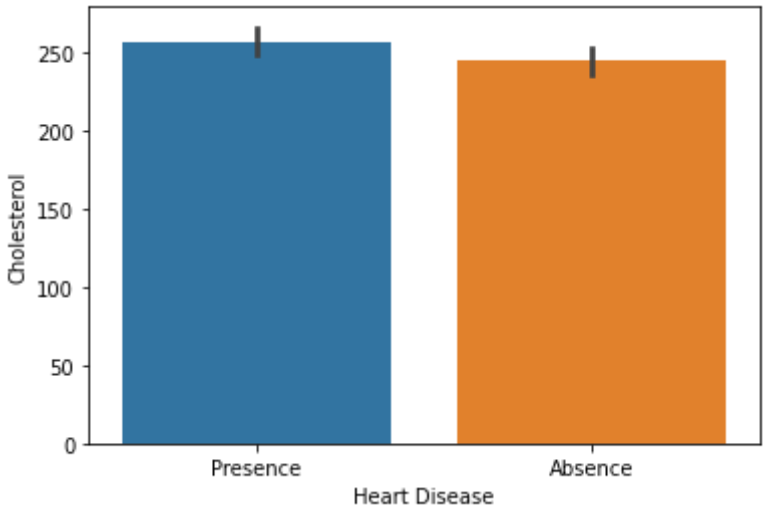

```
sns.barplot(x=df['Sex'],y=df['Cholesterol'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d669cb50>
```



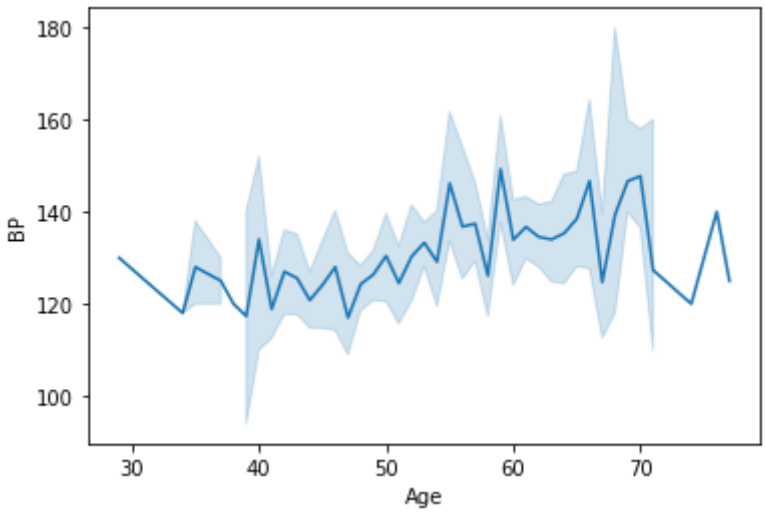

```
sns.barplot(x=df['Heart Disease'],y=df['Cholesterol'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d66150d0>
```



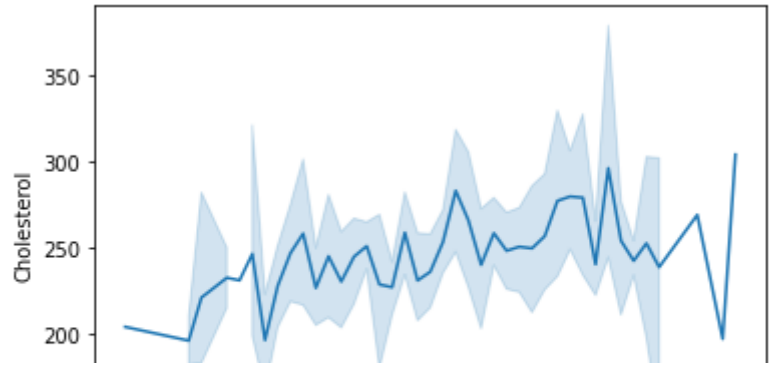

```
sns.lineplot(x=df['Age'],y=df['BP'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d65ec910>
```



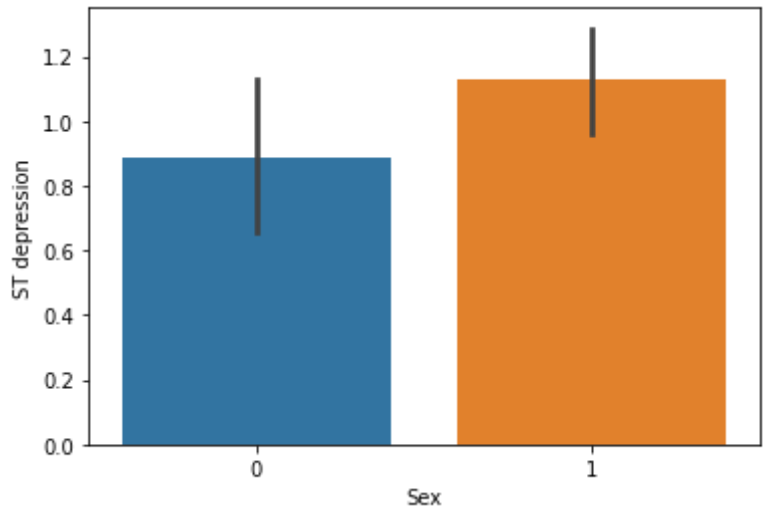

```
sns.lineplot(x=df['Age'],y=df['Cholesterol'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6510e90>
```



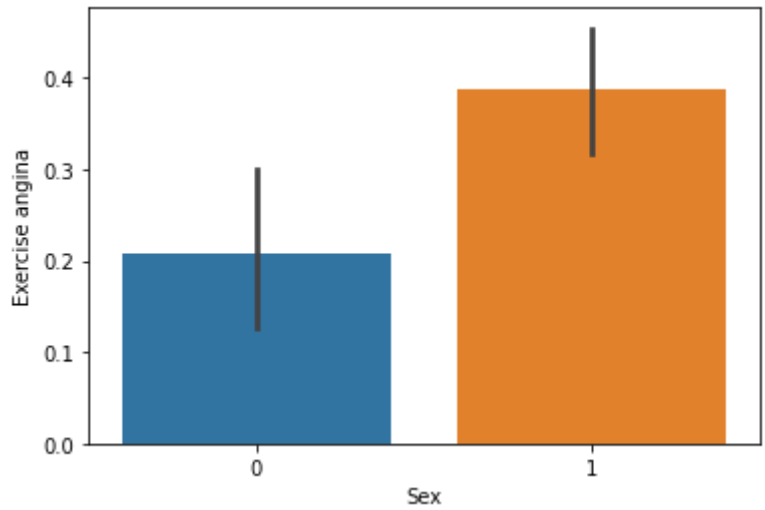

```
sns.barplot(x=df['Sex'],y=df['ST depression'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6503e90>
```



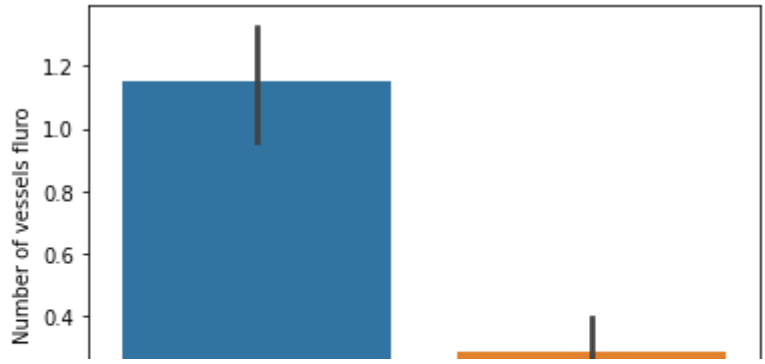

```
sns.barplot(x=df['Sex'],y=df['Exercise angina'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6455b50>
```



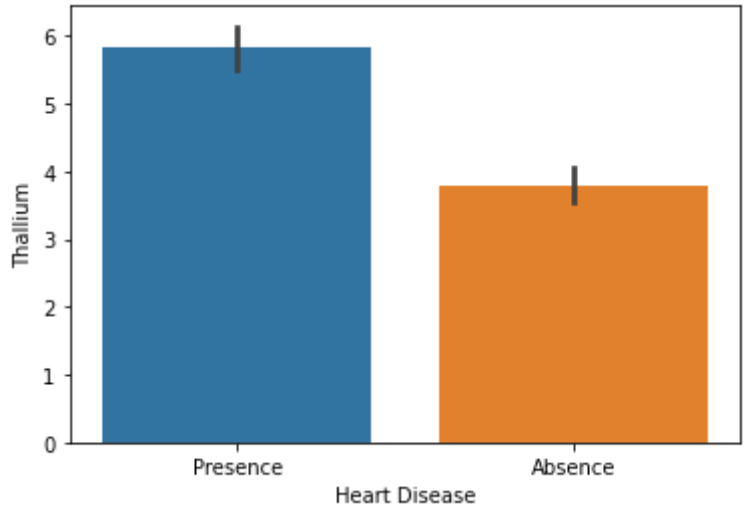

```
sns.barplot(x=df['Heart Disease'],y=df['Number of vessels fluro'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d64374d0>
```



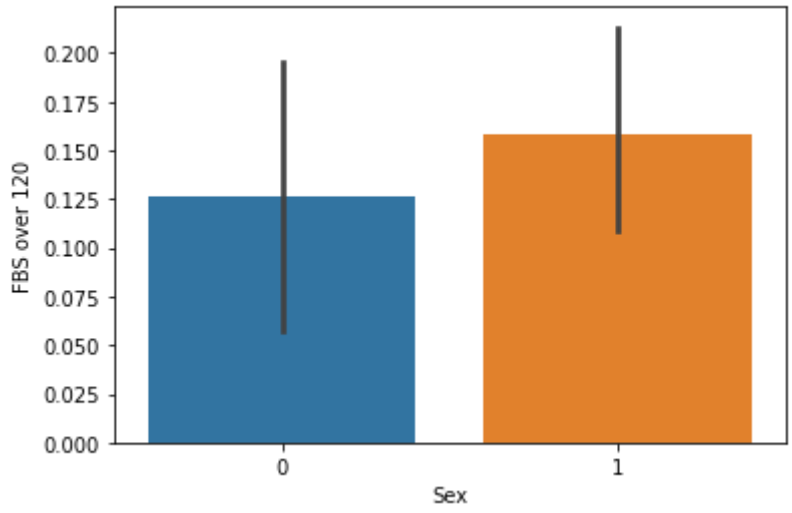

```
sns.barplot(x=df['Heart Disease'],y=df['Thallium'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d63a6690>
```



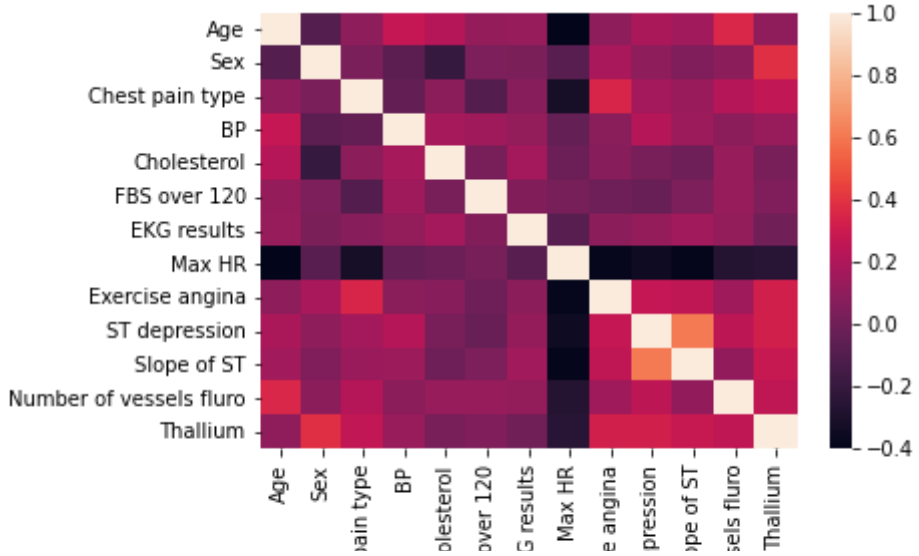

```
sns.barplot(x=df['Sex'],y=df['FBS over 120'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6371450>
```




```
sns.heatmap(df.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84d6626190>
```




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


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


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)


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)
score_1
```

```
0.7777777777777778
```

```
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)
score_2
```

```
0.7592592592592593
```

```
from xgboost import XGBClassifier
xgb=XGBClassifier()
xgb.fit(x_train,y_train)
pred_3=xgb.predict(x_test)
score_3=accuracy_score(y_test,pred_3)
score_3
```

```
0.7777777777777778
```

```
from sklearn.neighbors import KNeighborsClassifier
list_1=[]
for i in range(1,21):
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    preds=knn.predict(x_test)
    scores=accuracy_score(y_test,preds)
    list_1.append(scores)
max(list_1)
```

```
0.7037037037037037
```