Heart disease diagnostic analysis

```
#Extracting zip file


from zipfile import ZipFile


data1=ZipFile('/content/Heart Disease data.zip')


files=data1.namelist()
print(files)
```

```
['Heart Disease data/', 'Heart Disease data/Dataset Details.txt', 'Heart Disease data
```

```
data1.extract('Heart Disease data/Heart Disease data.csv')
```

```
'/content/Heart Disease data/Heart Disease data.csv'
```

```
import pandas as pd
data=pd.read_csv('/content/Heart Disease data/Heart Disease data.csv')


#Importing libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline


# Display all the columns of the dataframe
data.columns
```

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

```
# Print shape of the dataframe
data.shape
```

```
(1025, 14)
```

```
# First 5 rows of dataset
data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tha
0	52	1	0	125	212	0	1	168	0	1.0	2	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	

```
# Check missing values
data.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Observation:- There is no missing value in any of the column

```
def heart_diseases(value):
    '''function to convert numerical features to categorical feature'''
    if value==1:
      return 'absence'
    else:
      return 'presence'


#Adding new column to dataset of converted data
data['heart_disease']=data['target'].apply(heart_diseases)


data
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	52	1	0	125	212	0	1	168	0	1.0	2	2
1	53	1	0	140	203	1	0	155	1	3.1	0	0

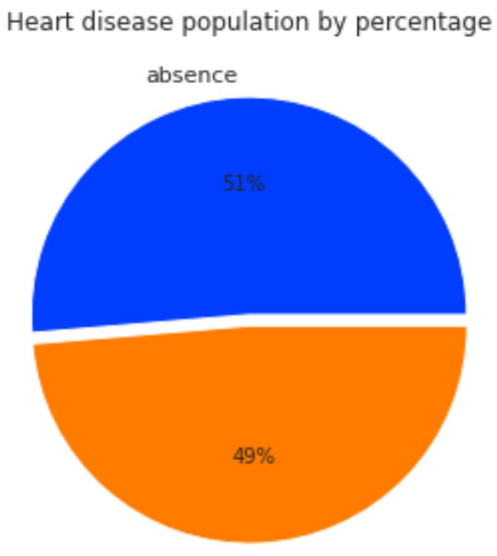
2	70	1	0	145	174	0	1	125	1	2.6	0	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1
4	62	0	0	138	294	1	1	106	0	1.9	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0
1021	60	1	0	125	258	0	0	141	1	2.8	1	1
1022	47	1	0	110	275	0	0	118	1	1.0	1	1
1023	50	0	0	110	254	0	0	159	0	0.0	2	0
1024	54	1	0	120	188	0	1	113	0	1.4	1	1

1025 rows × 15 columns

```
#Count number of people having heart disease and not having heart disease
hd=data.groupby('heart_disease')['target'].count()
hd
```

```
heart_disease
absence     526
presence    499
Name: target, dtype: int64
```

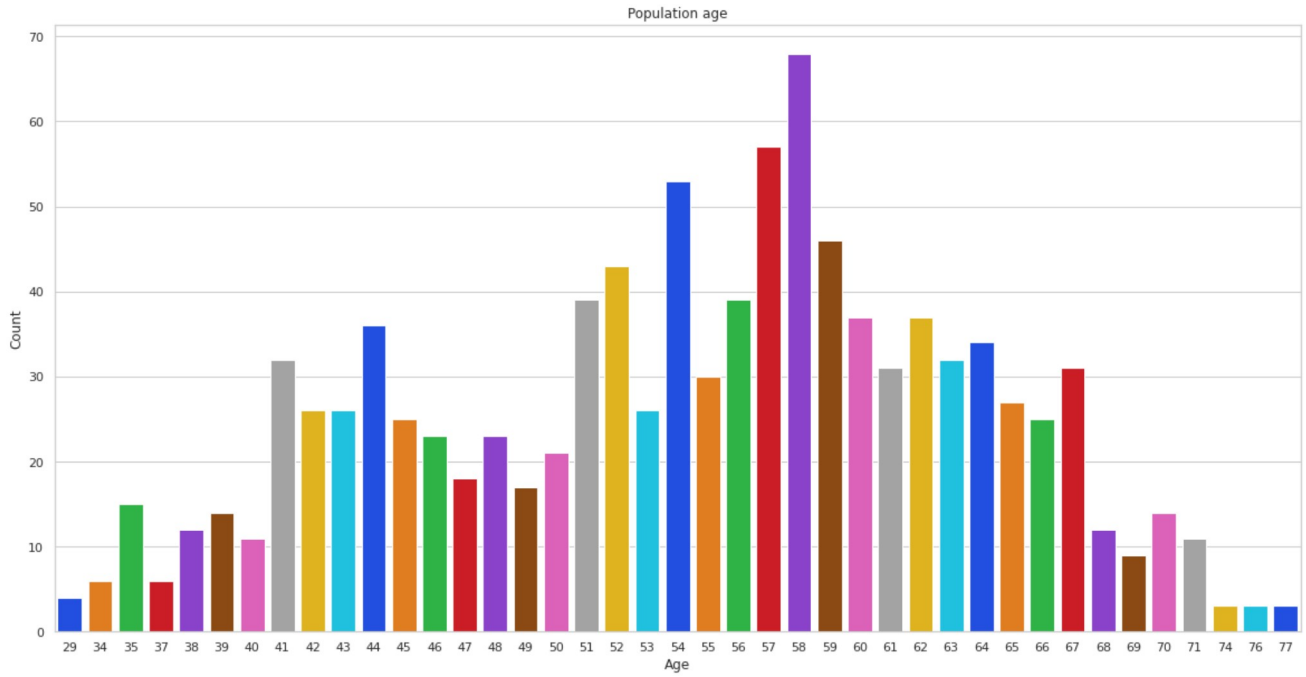
```
# Plot the chart of above using seaborn and matplotlib
plt.figure(figsize=(5,5))
clrs= sns.color_palette('bright')
explode=[0,0.05]
plt.pie(hd,labels=['absence','presence'],autopct="%0.0f%%",colors=clrs,explode=explode)
plt.title('Heart disease population by percentage')
plt.show()
```



presence

From the above observation,people having heart disease (49%) are less than people who do not have heart disease

```
#Plotting countplot of population age using matplotlib and seaborn
plt.figure(figsize=(20,10))
plt.title('Population age')
sns.countplot(x='age',data=data,palette='bright')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



We can observe the count of population according to their age (young, middle_age and elder

people)

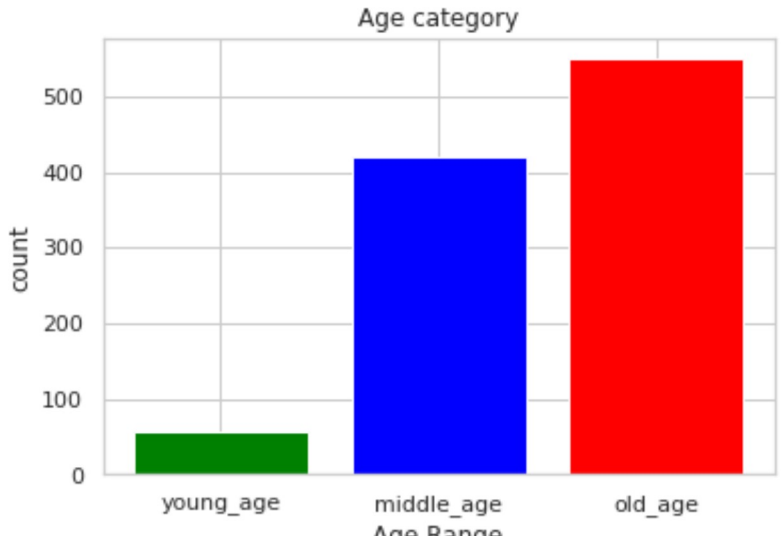
```
# To find the minimum, maximum and average of the population age using statistical analysis
min_age=data['age'].min()
max_age=data['age'].max()
mean_age=data['age'].mean()
print(f"The minimum age is {min_age}")
print(f"The maximum age is {max_age}")
print(f"The average age is {np.round(mean_age,2)}")
```

```
The minimum age is 29
The maximum age is 77
The average age is 54.43
```

```
#To divide the population age in different categories
young_age=data[(data['age']>=29)&(data['age']<40)]
middle_age=data[(data['age']>=40)&(data['age']<55)]
old_age=data[(data['age']>=55)]
print(f"Number of young age people={len(young_age)}")
print(f"Number of middle age people={len(middle_age)}")
print(f"Number of old age people={len(old_age)}")
```

```
Number of young age people=57
Number of middle age people=419
Number of old age people=549
```

```
#Bar plot using matplotlib and seaborn for different categories of population age
cat=['young_age','middle_age','old_age']
_count=[len(young_age),len(middle_age),len(old_age)]
plt.bar(cat,_count,color=['green','blue','red'])
plt.title('Age category')
plt.xlabel('Age Range')
plt.ylabel('count')
plt.show()
```



From the above plot we observe that old age population is more than the middle age and young age population. And there is least population of young age.

```
# Converting the numerical data into categorical data

def age_range(row):
    '''converting population age into range of age'''
    if row>=29 and row<40:
      return 'YoungAge'
    elif row>=40 and row<55:
      return 'MiddleAge'
    else:
      return 'OldAge'


#Applying converted data in our dataset

data['ageRange']=data['age'].apply(age_range)
data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tha
0	52	1	0	125	212	0	1	168	0	1.0	2	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	

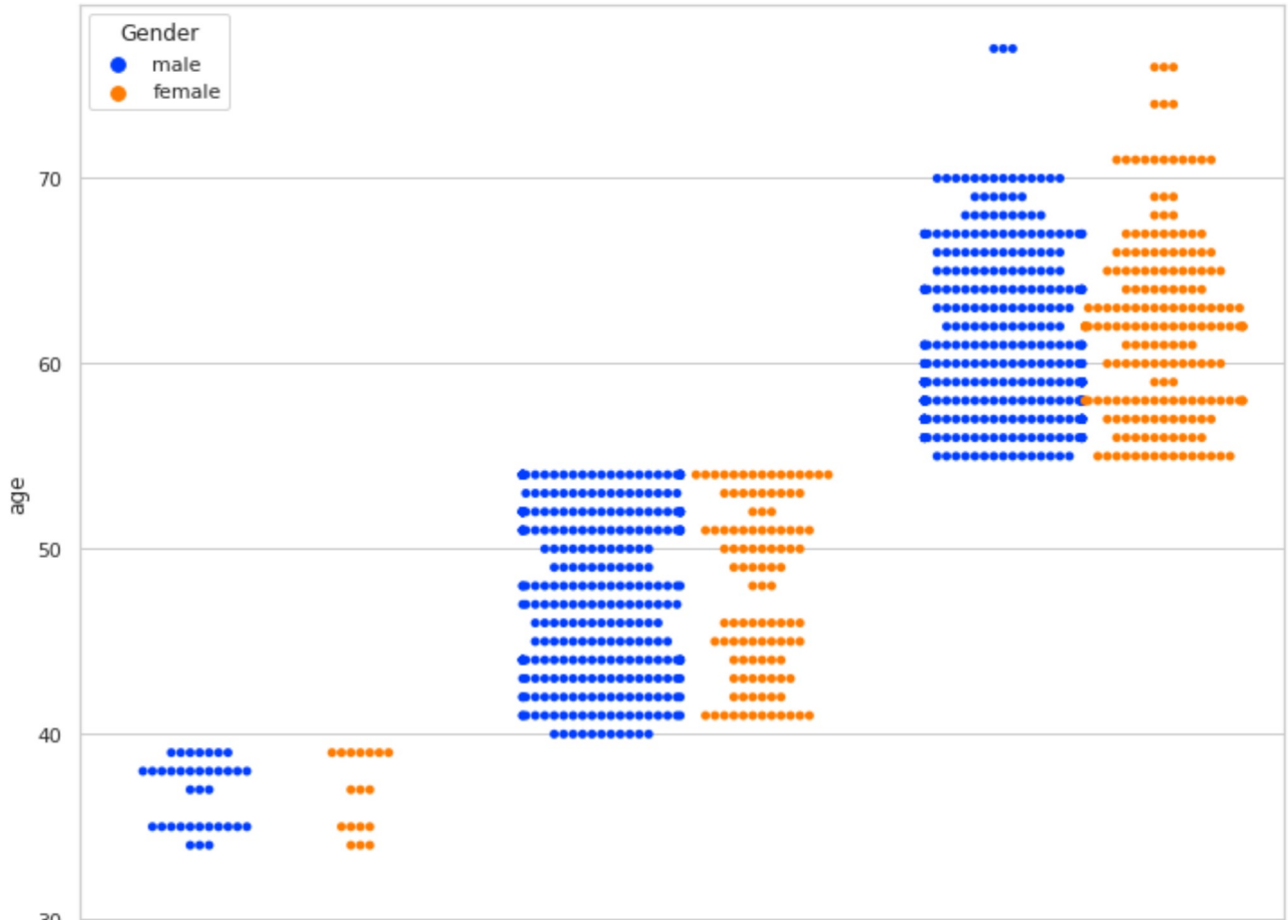
```
#Converting numerical data into categorical

def _sex(row):
    if row==1:
        return 'male'
    else:
        return 'female'


#Applying converted data in our dataset

data['Gender']=data['sex'].apply(_sex)
data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tha

0	52	1	0	125	212	0	1	168	0	1.0	2	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	

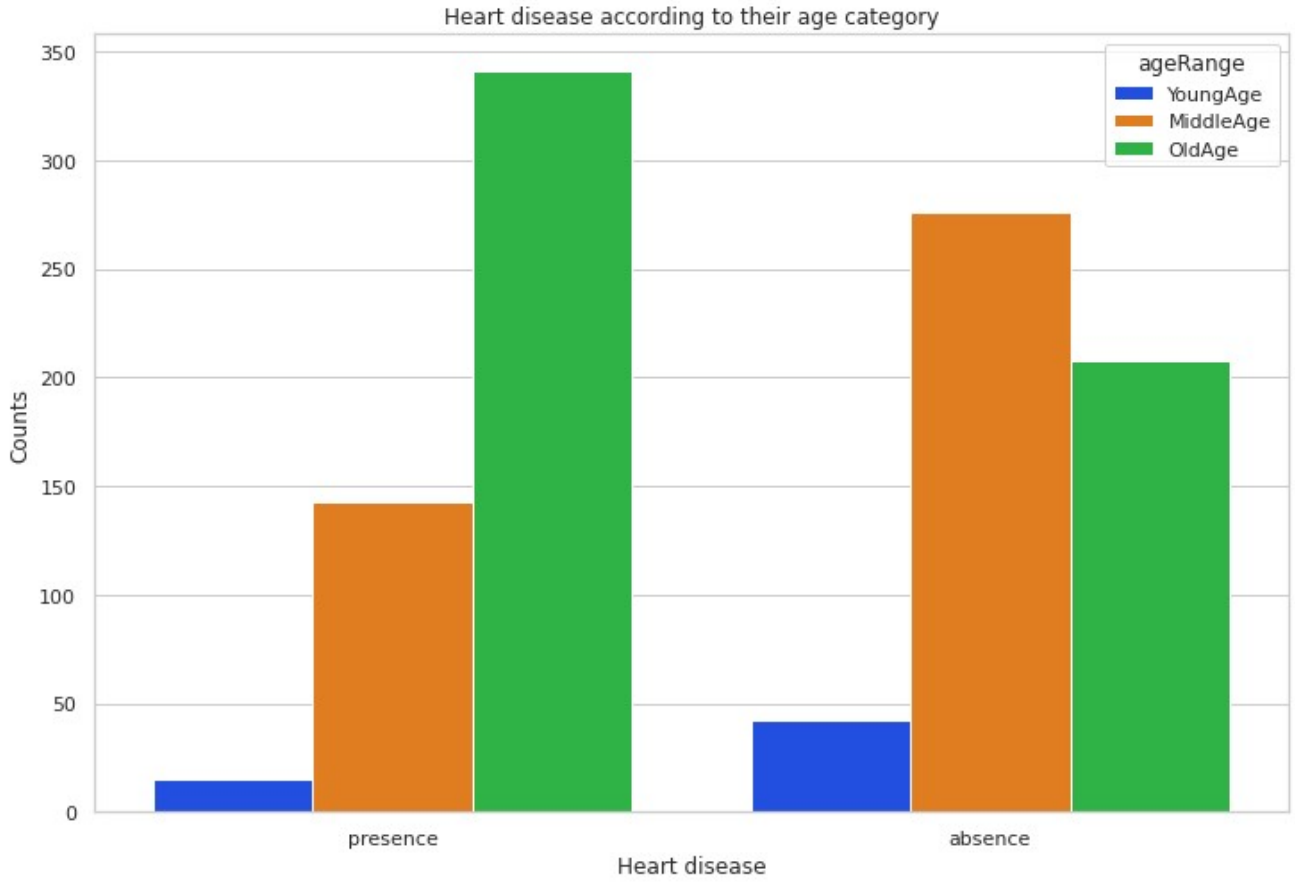
```
#Scatter plot creation Age category vs gender using matplotlib

plt.figure(figsize=(12,10))
x=data.ageRange
y=data.age
sns.set(style='whitegrid',palette='bright')
sns.swarmplot(x,y,hue='Gender',data=data,dodge=True,order=['YoungAge','MiddleAge','OldAge'])
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:1296: UserWarning: 25.3
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:1296: UserWarning: 33.8
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:1296: UserWarning: 5.9%
  warnings.warn(msg, UserWarning)
```



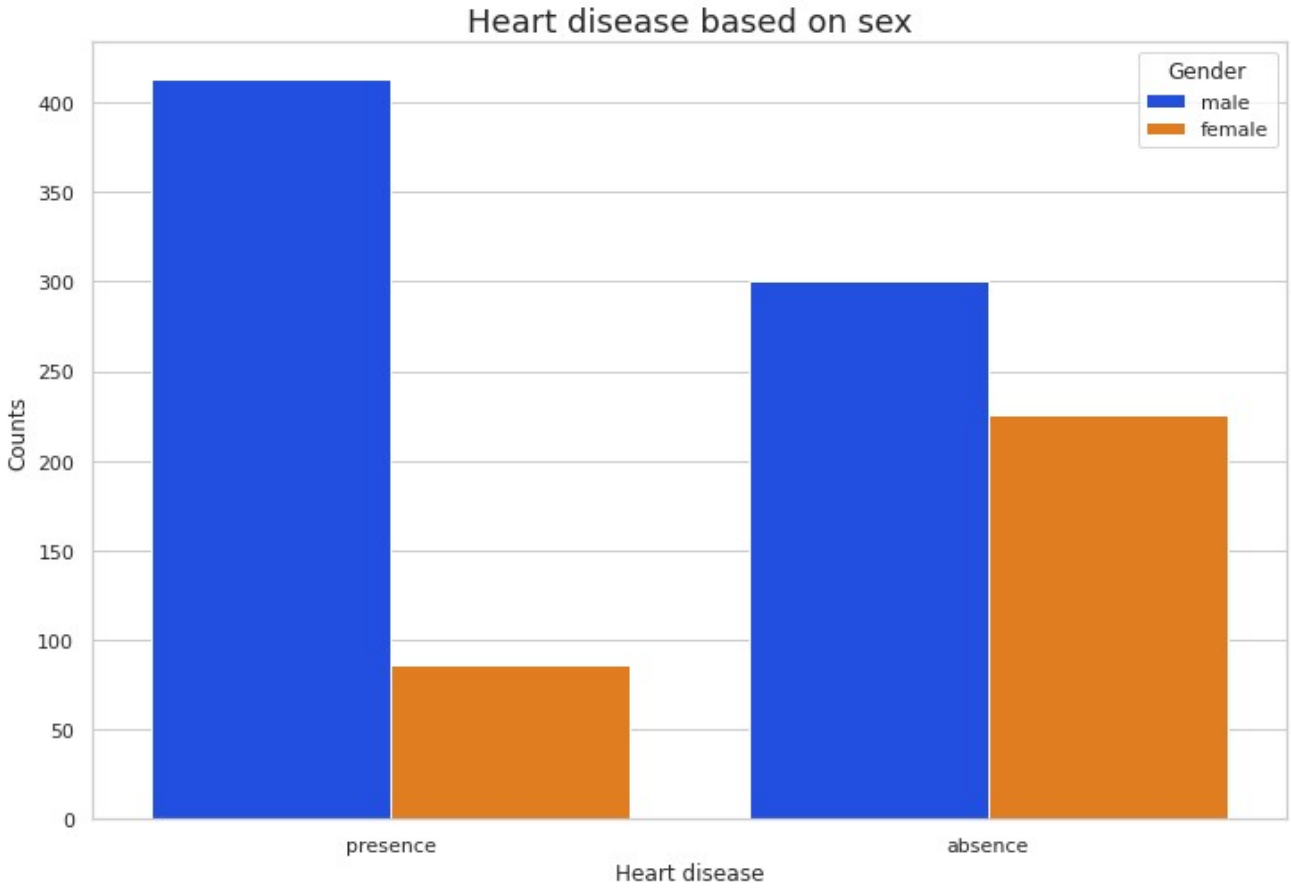
In the given dataset, number of male population is more than the female population in each age group

```
#Countplot of heart disease according to their age category

plt.figure(figsize=(12,8))
hue_orders = ['YoungAge','MiddleAge','OldAge']
plt.title("Heart disease according to their age category")
sns.countplot(x='heart_disease',hue='ageRange',data=data,hue_order=hue_orders,palette='brig
plt.xlabel("Heart disease")
plt.ylabel("Counts")
plt.show()
```



Old Age people are most affected by Heart disease and young age are least affected

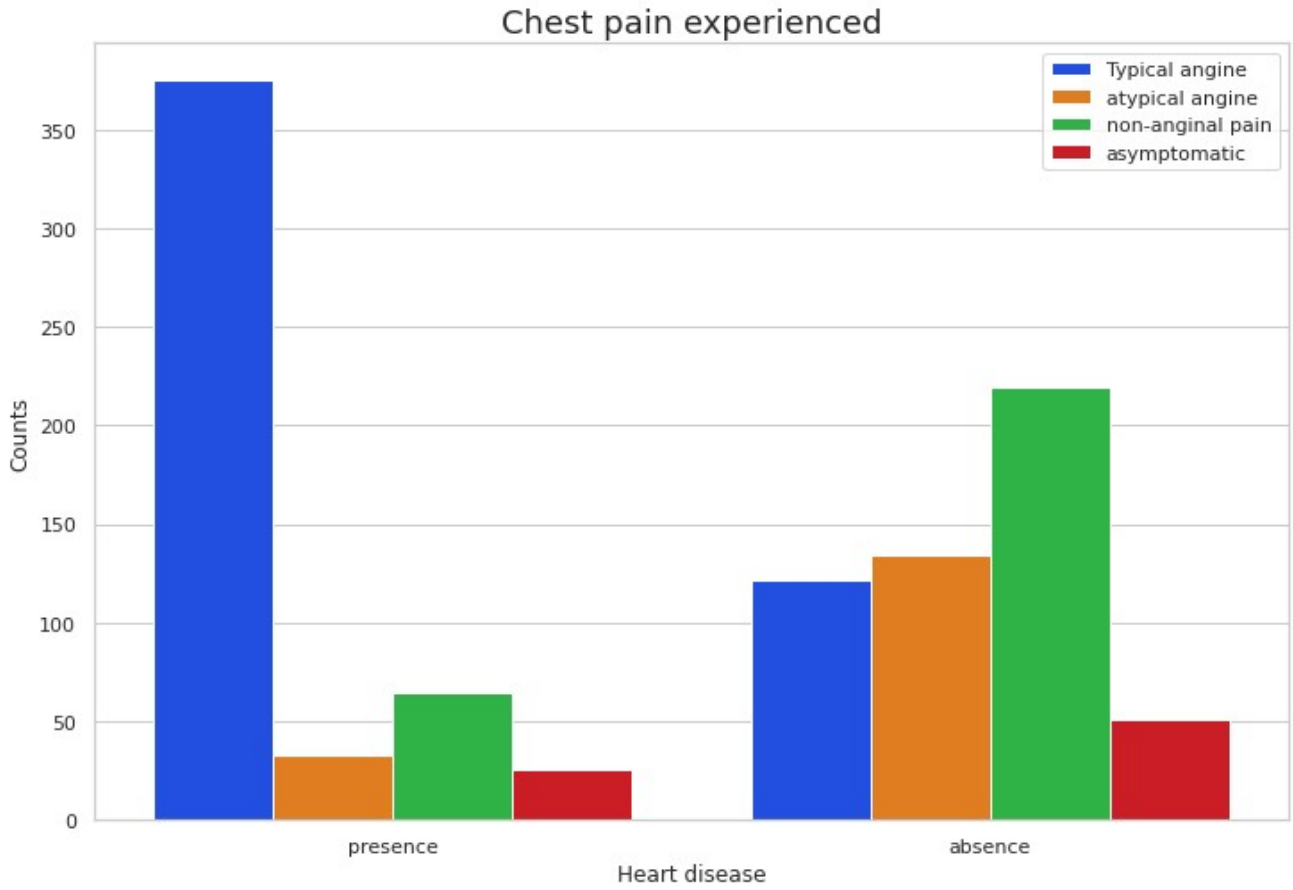
```
#Countplot for heart disease based on sex

plt.figure(figsize=(12,8))
plt.title("Heart disease based on sex",fontsize=18)
sns.countplot(x=data['heart_disease'],hue='Gender',data=data,palette='bright')
plt.xlabel("Heart disease")
plt.ylabel("Counts")
plt.show()
```



From the above graph it is observed that there are more number of males affected by heart disease as compared to females

```
#Countplot based on chest pain experienced

plt.figure(figsize=(12,8))
plt.title("Chest pain experienced",fontsize=18)
sns.countplot(x=data['heart_disease'],hue='cp',data=data,palette='bright')
```

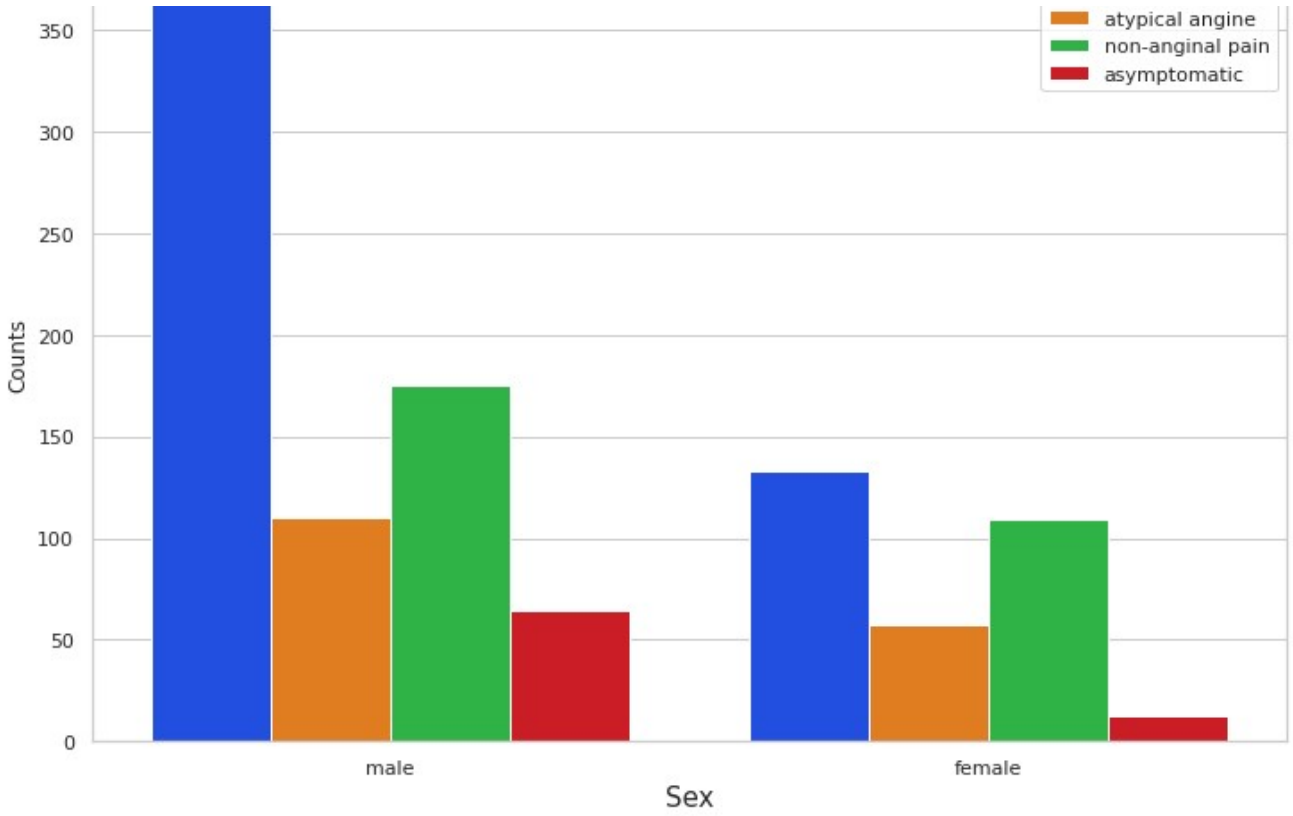
```
plt.legend(labels=['Typical angine','atypical angine','non-anginal pain','asymptomatic'])
plt.xlabel("Heart disease")
plt.ylabel("Counts")
plt.show()
```



It is observed that people having typical anginal chest pain have higher chance of heart disease

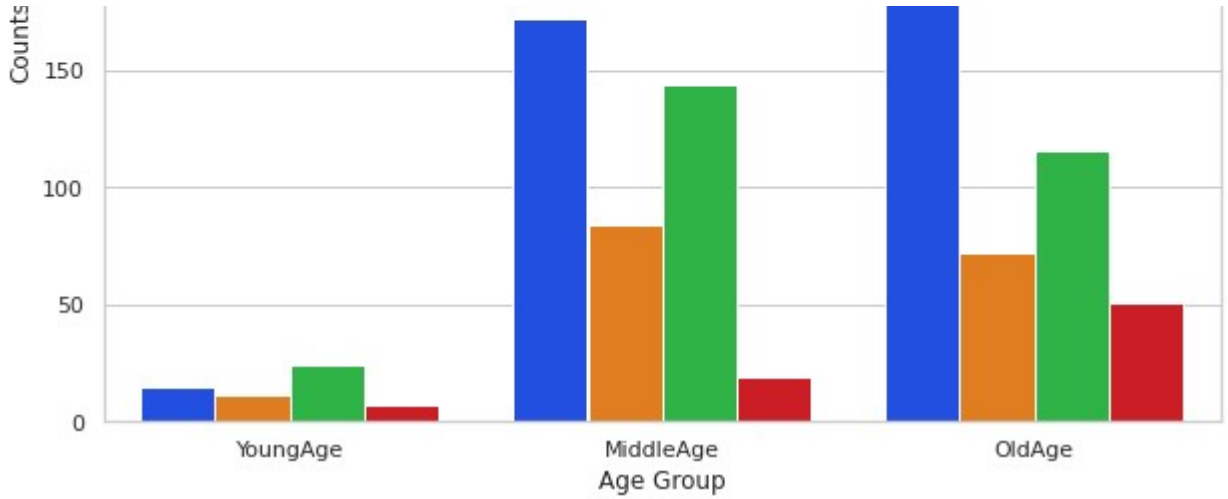
```
#Count plot for chest pain according to gender

plt.figure(figsize=(12,8))
plt.title("Chest pain based on sex",fontsize=18)
sns.countplot('Gender',hue='cp',data=data,palette='bright')
plt.legend(labels=['Typical angine','atypical angine','non-anginal pain','asymptomatic'])
plt.xlabel("Sex",fontsize=15)
plt.ylabel("Counts")
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass
  warnings.warn(
```



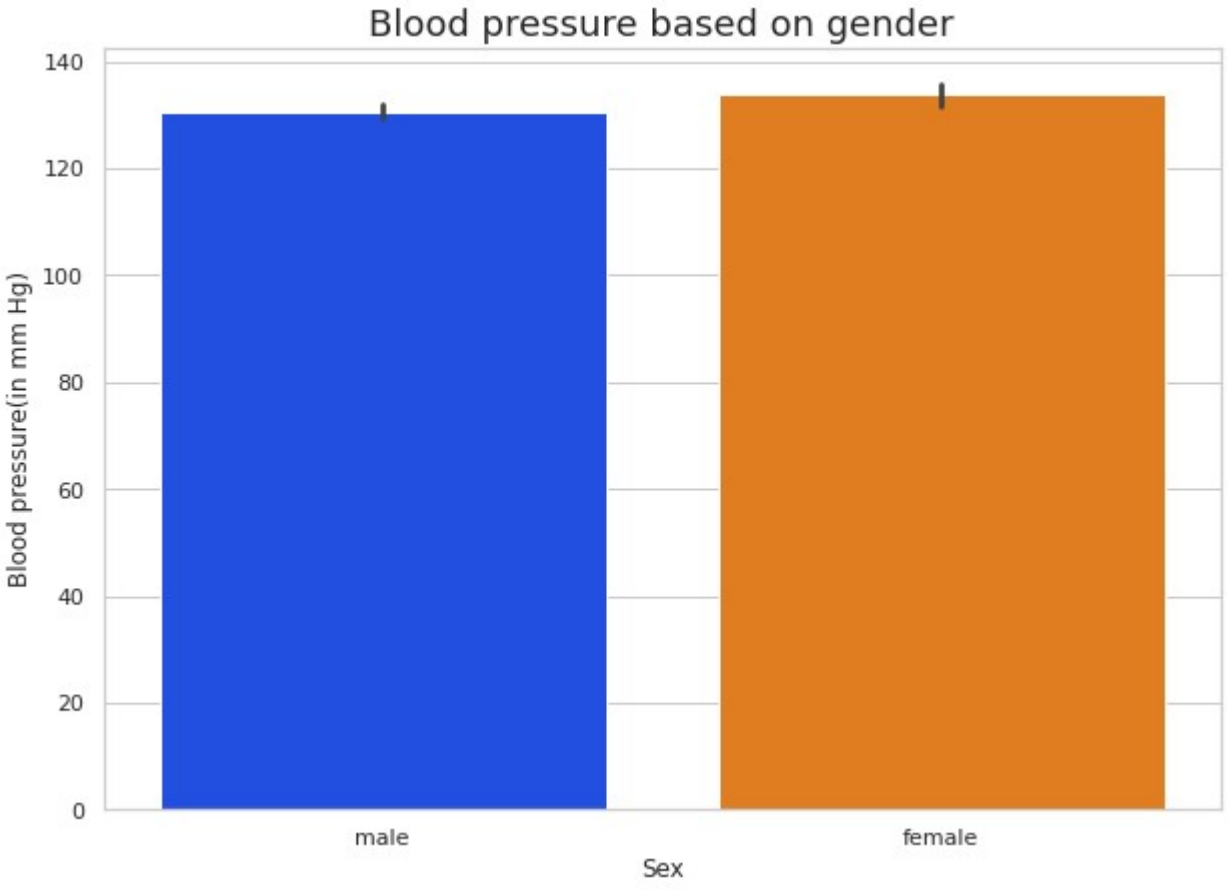
It is observed that higher number of males are suffering from typical angine chest pain

```
#Countplot for chest pain vs age group using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("Chest pain based on age group",fontsize=18)
sns.countplot(x=data['ageRange'],hue='cp',data=data,palette='bright',order=['YoungAge','Mi
plt.legend(labels=['Typical angine','atypical angine','non-anginal pain','asymptomatic'])
plt.xlabel("Age Group")
plt.ylabel("Counts")
plt.show()
```



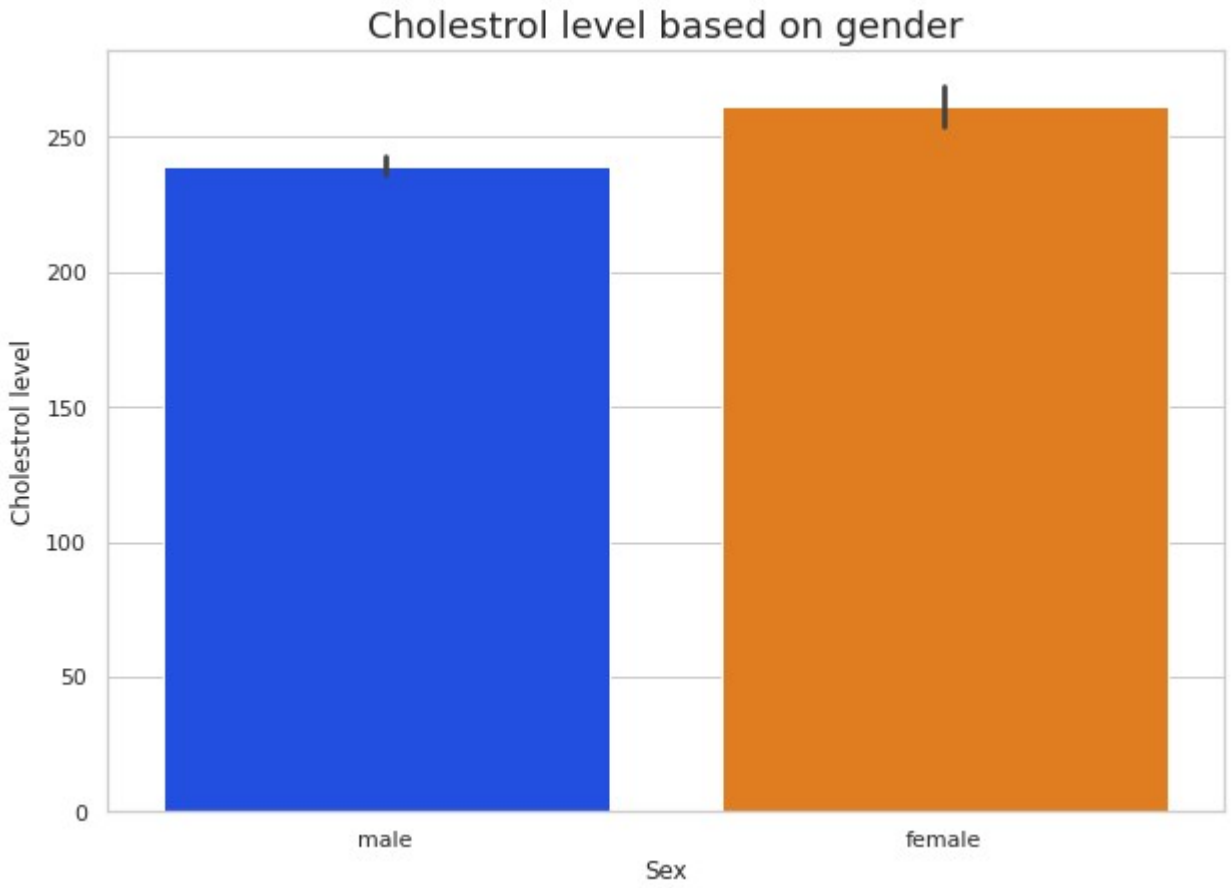
It seems that old age group have higher typical angine chest pain

```
#Bar graph for Restin blood pressure (trestbs)(in mm Hg) based on gender using matplotlib a

plt.figure(figsize=(10,7))
plt.title("Blood pressure based on gender",fontsize=18)
sns.barplot(x='Gender',y='trestbps',data=data,palette='bright')
plt.xlabel("Sex")
plt.ylabel("Blood pressure(in mm Hg)")
plt.show()
```



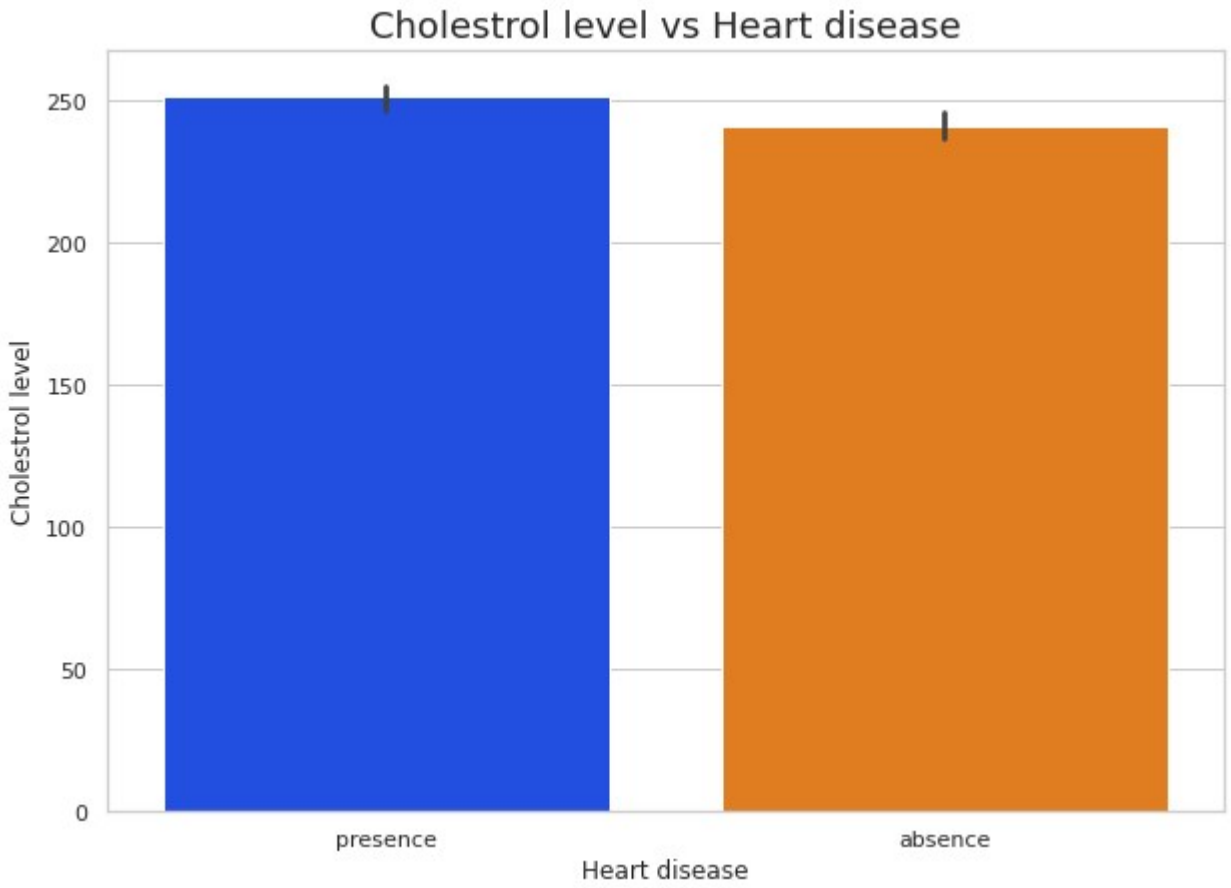
It is observed that person's Resting blood pressure is almost same for males and females

```
#Bar Graph for Cholestrol level based on gender using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("Cholestrol level based on gender",fontsize=18)
sns.barplot(x='Gender',y='chol',data=data,palette='bright')
plt.xlabel("Sex")
plt.ylabel("Cholestrol level")
plt.show()
```



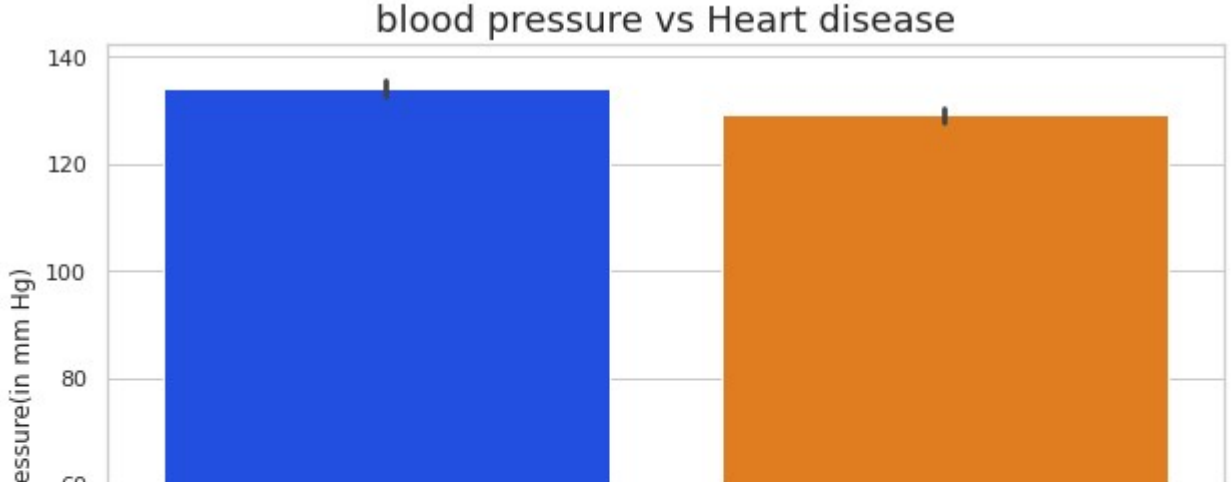
Cholestrol level is little bit more in females as compared to males

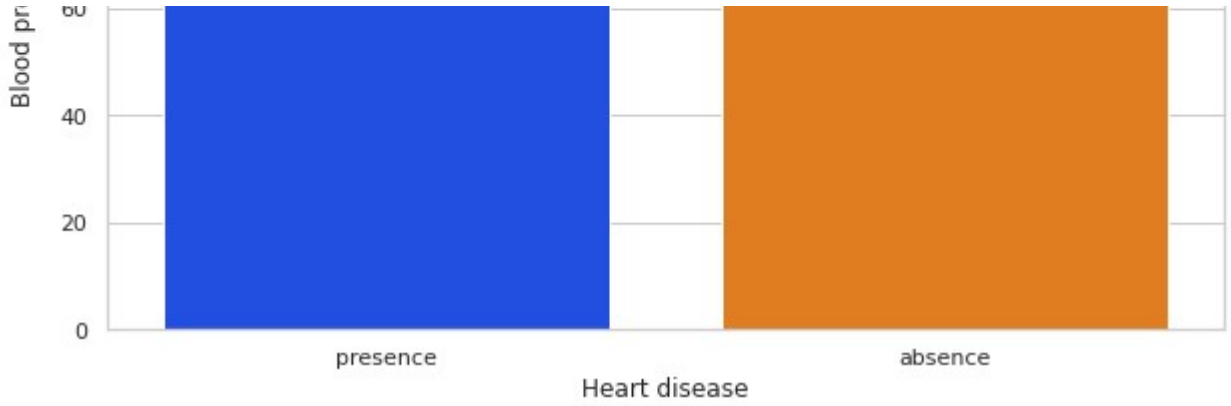
```
#Bar plot for Cholestrol level vs Heart disease using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("Cholestrol level vs Heart disease",fontsize=18)
sns.barplot(x='heart_disease',y='chol',data=data,palette='bright')
plt.xlabel("Heart disease")
plt.ylabel("Cholestrol level")
plt.show()
```



Cholestrol level might have affect on heart disease

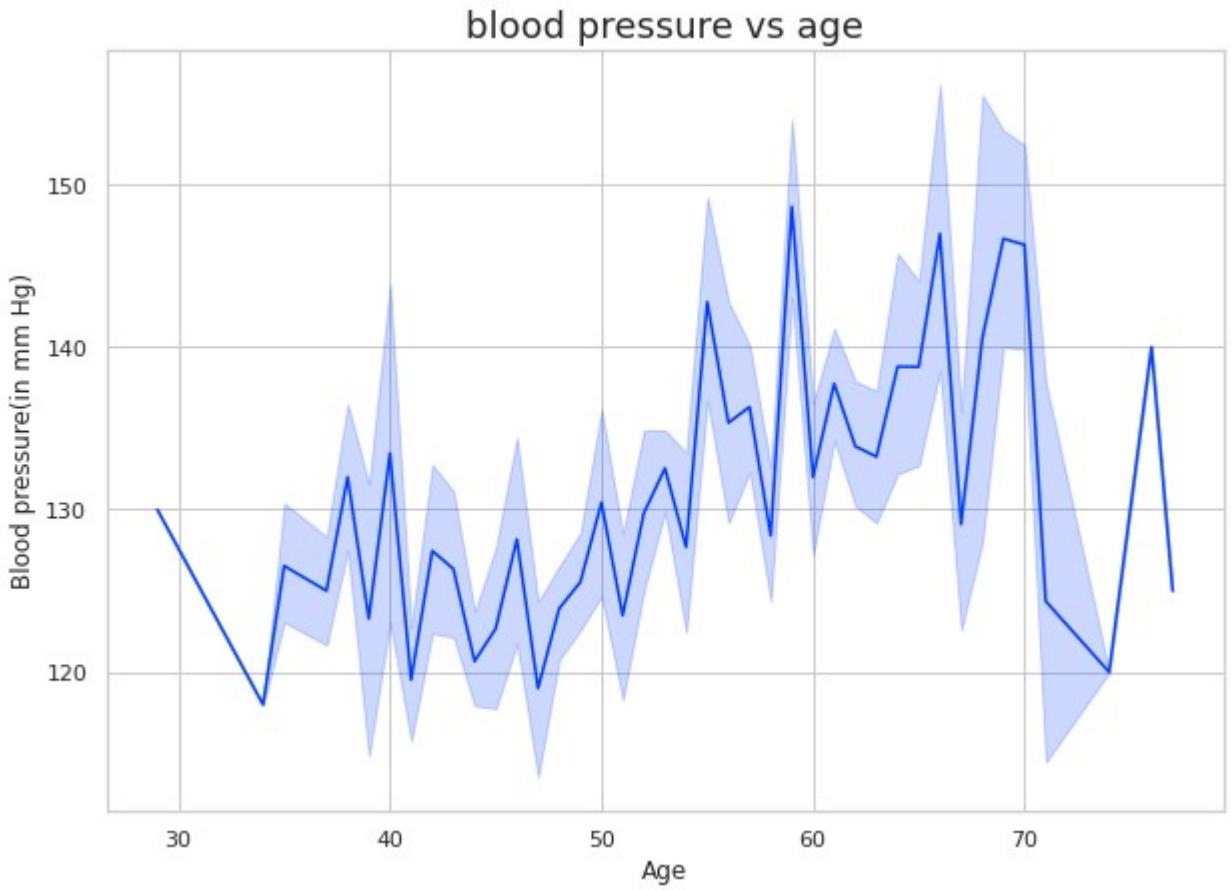
```
#Bar Graph for blood pressure vs Heart disease using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("blood pressure vs Heart disease",fontsize=18)
sns.barplot(x='heart_disease',y='trestbps',data=data,palette='bright')
plt.xlabel("Heart disease")
plt.ylabel("Blood pressure(in mm Hg)")
plt.show()
```





Blood pressure might have affect on heart disease

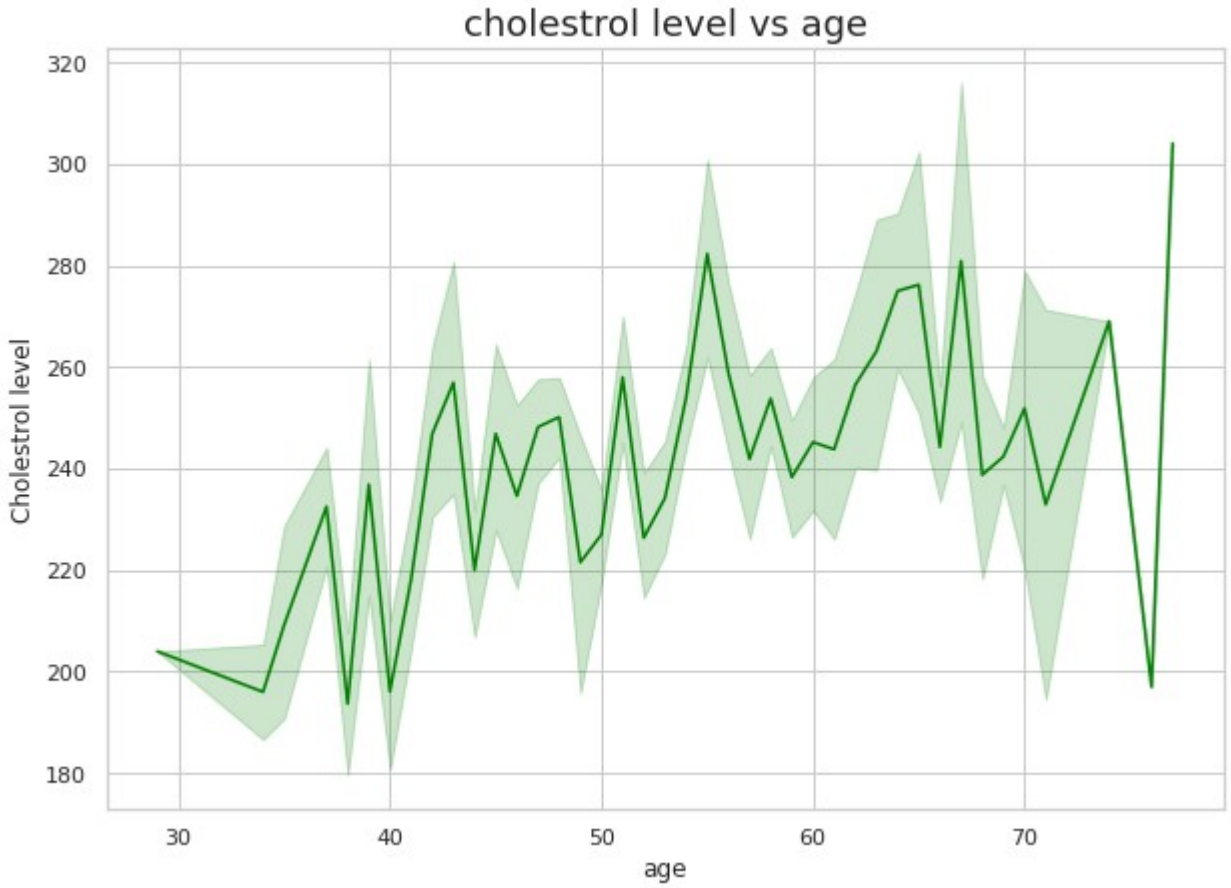
```
#Line plot for blood pressure vs age using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("blood pressure vs age",fontsize=18)
sns.lineplot(x='age',y='trestbps',data=data,palette='bright')
plt.xlabel("Age")
plt.ylabel("Blood pressure(in mm Hg)")
plt.show()
```



Here we observe that blood pressure is normal from 30-50 and after 50 it increses gradually to

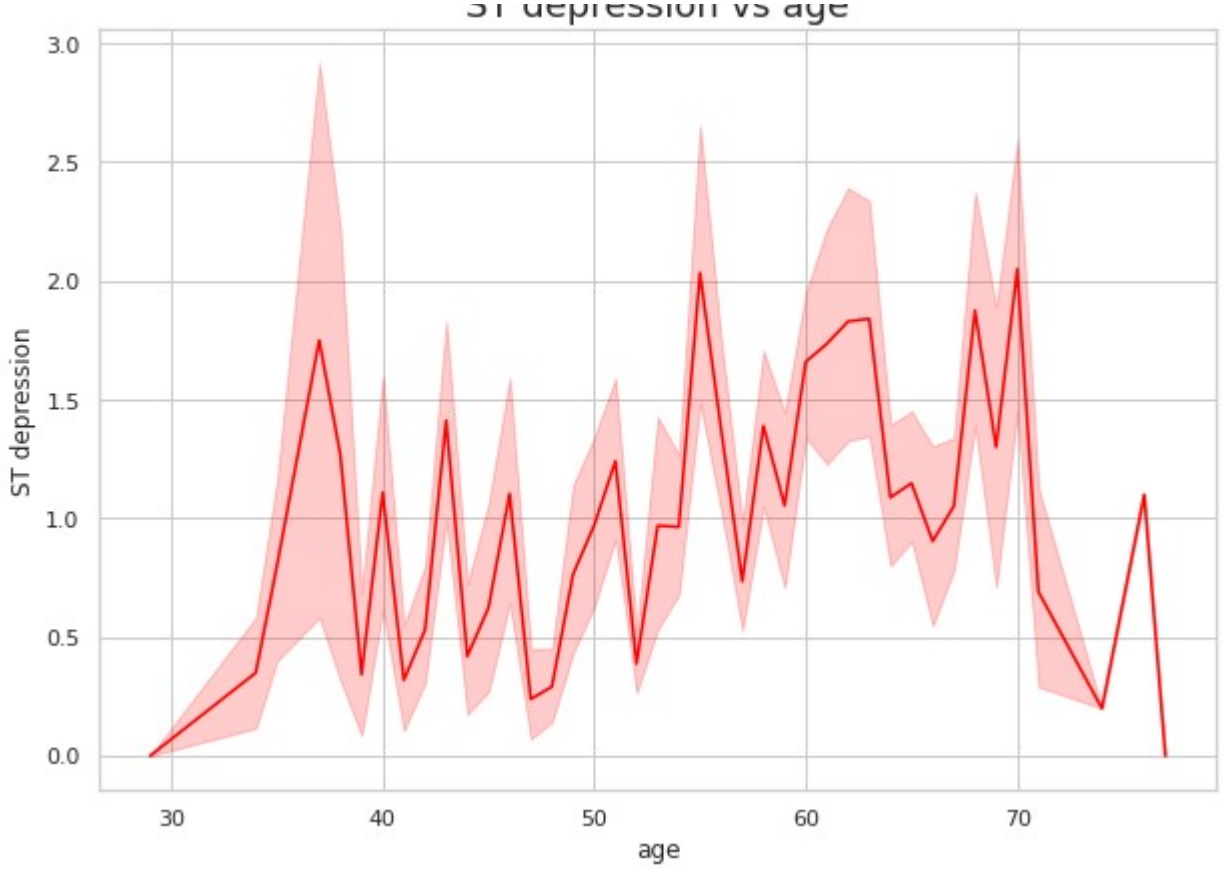
age of 60. And after 60 it is fluctuating drastically

```
#Line plot for cholestrol level vs age using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("cholestrol level vs age",fontsize=18)
sns.lineplot(x='age',y='chol',data=data,color='green')
plt.xlabel("age")
plt.ylabel("Cholestrol level")
plt.show()
```



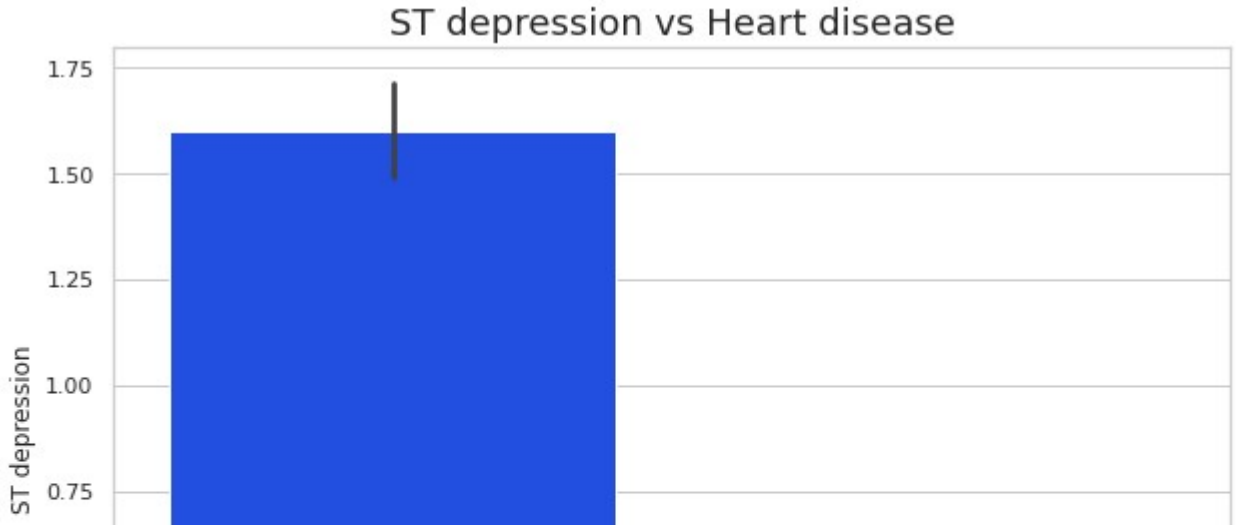
Cholestrol level is start increasing at the age of 50

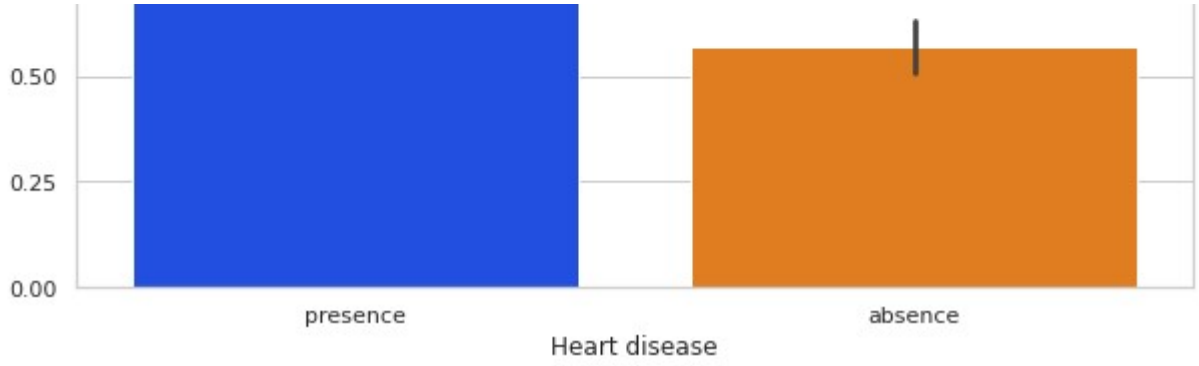
```
#Line plot for ST depression vs age using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("ST depression vs age",fontsize=18)
sns.lineplot(x='age',y='oldpeak',data=data,color='red')
plt.xlabel("age")
plt.ylabel("ST depression")
plt.show()
```



Depression is quite high in the age group of 30 to 40 and 50-60.For 40-50 it remains stable

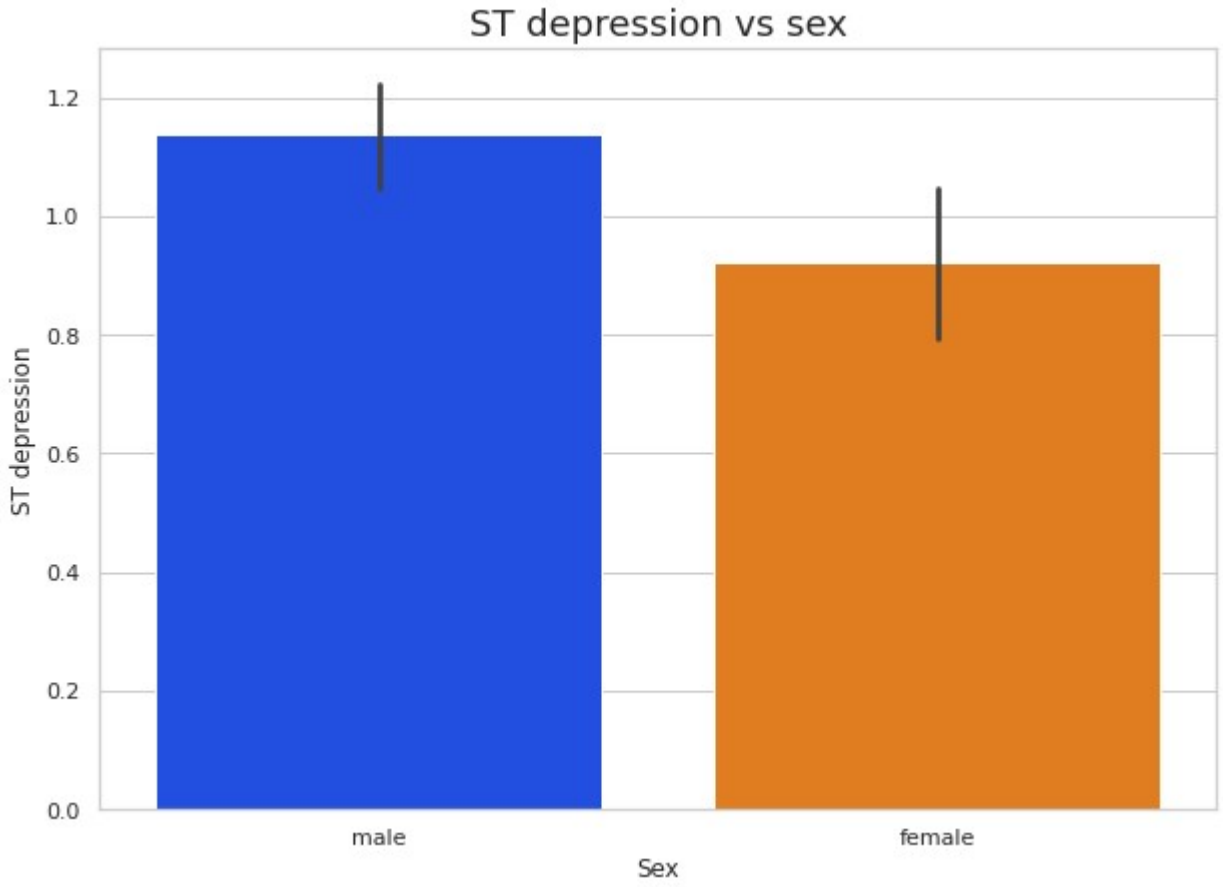
```
#Bar plot for ST depression vs Heart disease using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("ST depression vs Heart disease",fontsize=18)
sns.barplot(x='heart_disease',y='oldpeak',data=data,palette='bright')
plt.xlabel("Heart disease")
plt.ylabel("ST depression")
plt.show()
```





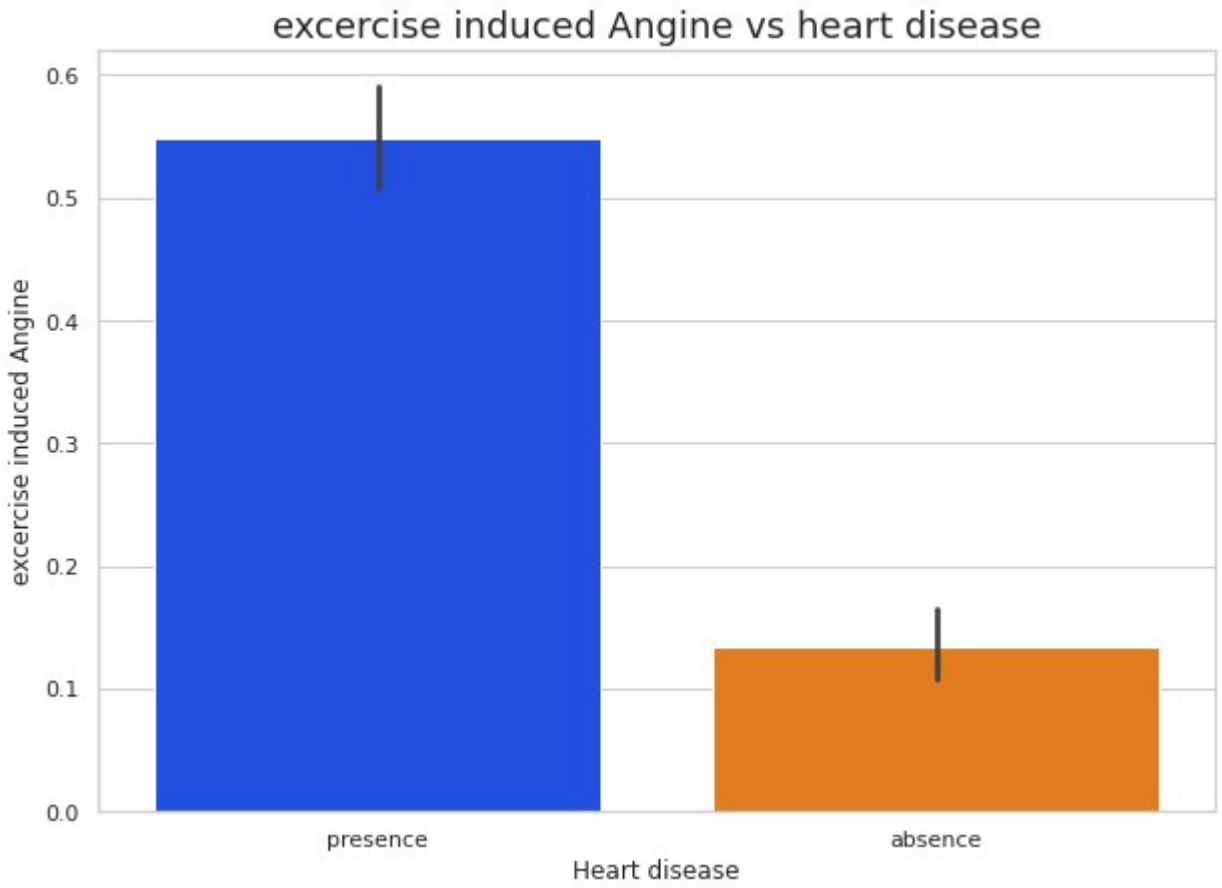
People with ST depression have higher chance of heart disease

```
#Bar·graph·for·ST·depression·vs·Gender·using·matplotlib·and·seaborn

plt.figure(figsize=(10,7))
plt.title("ST·depression·vs·sex",fontsize=18)
sns.barplot(x='Gender',y='oldpeak',data=data,palette='bright')
plt.xlabel("Sex")
plt.ylabel("ST·depression")
plt.show()
```



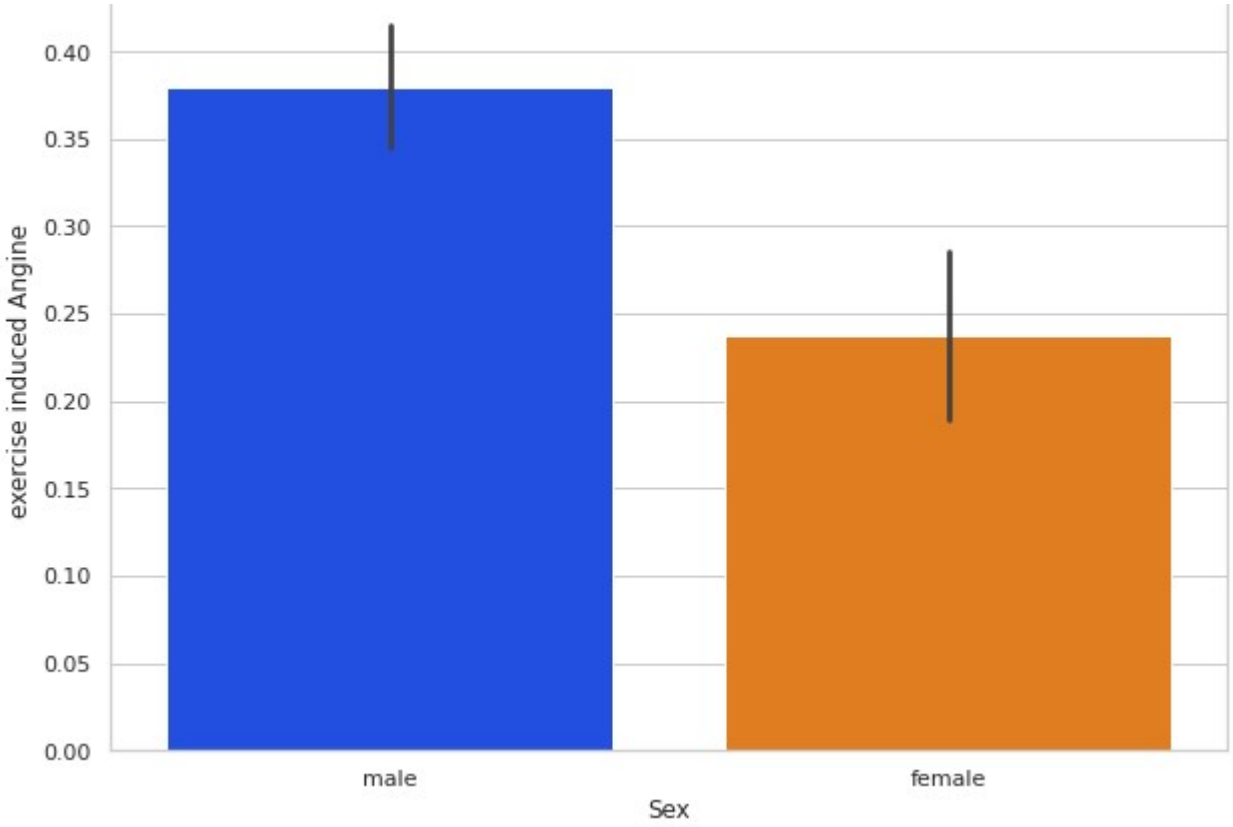
It is observed that more number of males are prone to ST depression as compared to females

```
#Bar·graph·for·excercise·induced·Angine·vs·heart·disease·using·matplotlib·and·seaborn

plt.figure(figsize=(10,7))
plt.title("excercise·induced·Angine·vs·heart·disease",fontsize=18)
sns.barplot(x='heart_disease',y='exang',data=data,palette='bright')
plt.xlabel("Heart·disease")
plt.ylabel("excercise·induced·Angine")
plt.show()
```



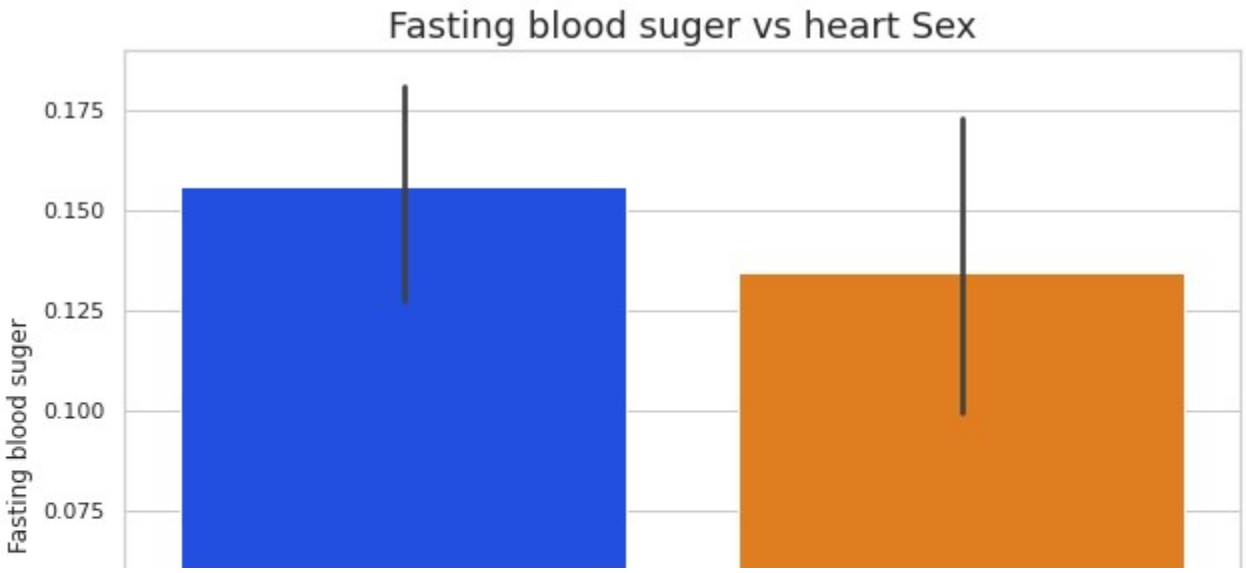
It is observed that if the people already suffered with angine then exercise will make it worse condition for him/her

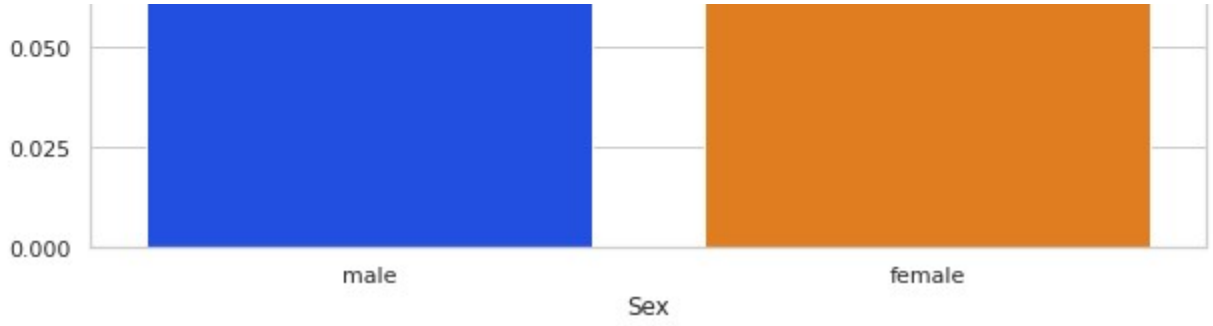
```
#Bar graph for excercise induced Angine vs gender using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("exercise induced Angine vs Sex",fontsize=18)
sns.barplot(x='Gender',y='exang',data=data,palette='bright')
plt.xlabel("Sex")
plt.ylabel("exercise induced Angine")
plt.show()
```



Male having high exercise induced Angine

```
#Bar graph for Fasting blood suger vs heart Gender using matplotlib and seaborn

plt.figure(figsize=(10,7))
plt.title("Fasting blood suger vs heart Sex",fontsize=18)
sns.barplot(x='Gender',y='fbs',data=data,palette='bright')
plt.xlabel("Sex")
plt.ylabel("Fasting blood suger ")
plt.show()
```

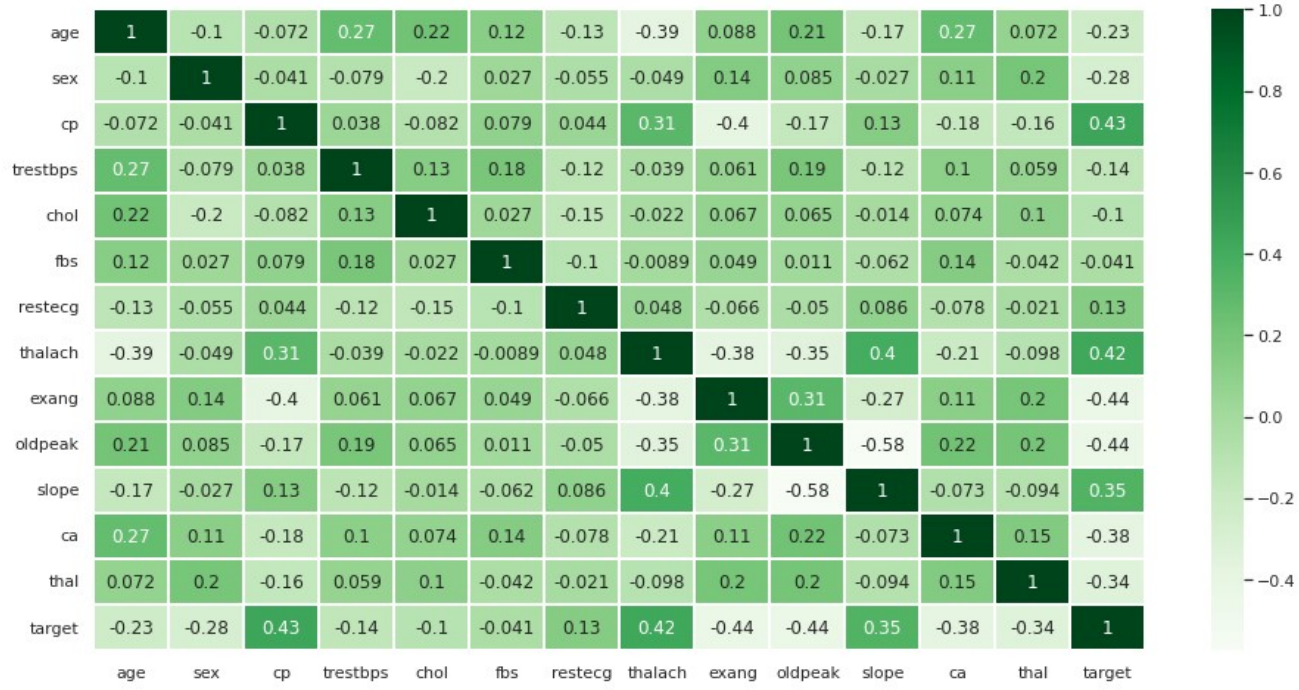




It is observed that males having high fasting blood sugar>120mg/dl

```
#Heatmap using seaborn

plt.figure(figsize=(16,8))
sns.heatmap(data.corr(),annot=True,linewidths=2,linecolor='white',cmap='Greens')
```

```
<AxesSubplot:>
```




```
data
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	52	1	0	125	212	0	1	168	0	1.0	2	2

1	53	1	0	140	203	1	0	155	1	3.1	0	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1
4	62	0	0	138	294	1	1	106	0	1.9	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0
1021	60	1	0	125	258	0	0	141	1	2.8	1	1
1022	47	1	0	110	275	0	0	118	1	1.0	1	1
1023	50	0	0	110	254	0	0	159	0	0.0	2	0
1024	54	1	0	120	188	0	1	113	0	1.4	1	1

1025 rows × 17 columns