




## #Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set_style('whitegrid')
```

## #Extracting CSV Dataset From System using Pandas Library

```
data=pd.read_csv('Heart Disease data.csv')
data
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
<b>0</b>	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	
<b>1</b>	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	
<b>2</b>	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	
<b>3</b>	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	
<b>4</b>	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
<b>1020</b>	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1	
<b>1021</b>	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0	
<b>1022</b>	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0	
<b>1023</b>	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1	
<b>1024</b>	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0	

1025 rows × 14 columns

Next steps:

[Generate code with data](#)[View recommended plots](#)

```
#All Columns in the Dataset
```

```
data.columns
```

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

```
#Checking NULL 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
```

```
#Percentage of people having Heart Disease
```

```
target=data.groupby('target').size()
```

```
target
```



```
target  
0      499  
1      526  
dtype: int64
```

```
#Converting Numerical Data into Categorical Data
```

```
def heart_disease(row):
    if row==0:
        return 'Absence'
    elif row==1:
        return 'Presence'
```

```
#Applying converted data into our dataset with new column - Heart_Disease
```

```
data['Heart_Disease']=data['target'].apply(heart_disease)
data.head()
```

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

Next steps:

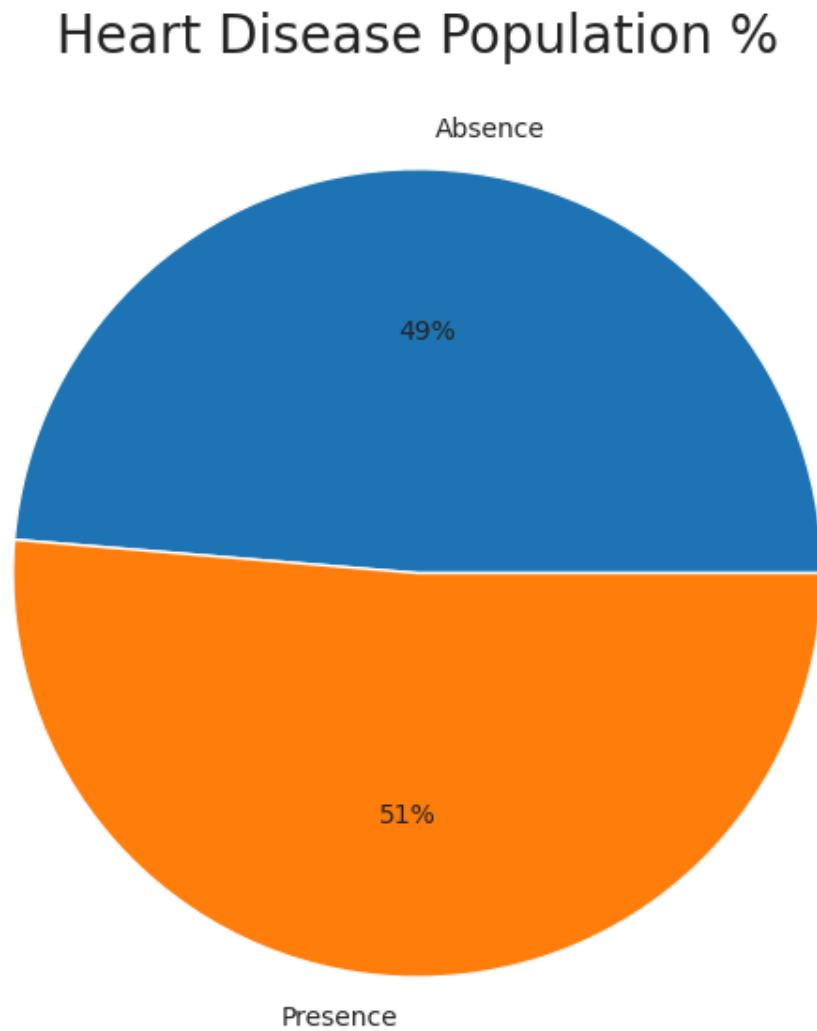
[Generate code with data](#)
[View recommended plots](#)

```
hd=data.groupby('Heart_Disease')['target'].count()
hd
```

```
Heart_Disease
Absence      499
Presence     526
Name: target, dtype: int64
```

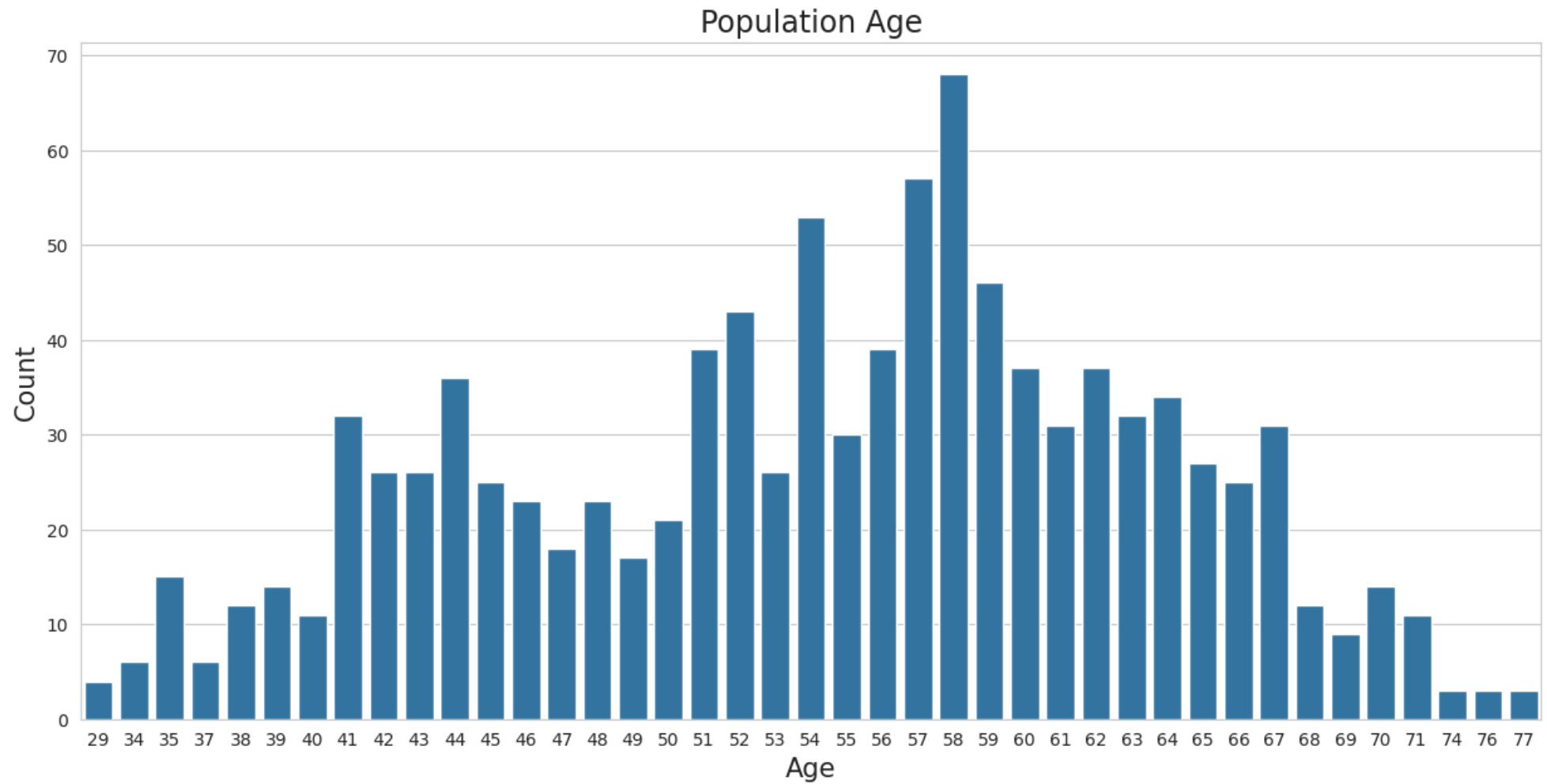
```
#Pie Chart Creation of Heart Disease Population % using Matplotlib
```

```
plt.figure(figsize=(10,7))  
plt.pie(hd, labels=['Absence','Presence'], autopct='%0.0f%%')  
plt.title('Heart Disease Population %', fontsize=20)  
plt.show()
```



#Countplot Creation of Population Age using Matplotlib and Seaborn

```
plt.figure(figsize=(15,7))
sns.countplot(x='age', data=data)
plt.title('Population Age', fontsize=17)
plt.xlabel('Age', fontsize=15)
plt.ylabel('Count', fontsize=15)
plt.show()
```



### #Statistical Analysis

```
Min_Age=data['age'].min()
Max_Age=data['age'].max()
Mean_Age=data['age'].mean()
print("Minimum Age =",Min_Age)
print("Maximum Age =",Max_Age)
print("Mean Age =",Mean_Age)

    Minimum Age = 29
    Maximum Age = 77
    Mean Age = 54.43414634146342
```

### #Categorical Analysis

```
Young_Ages=data[(data['age']>=29) & (data['age']<40)]
Middle_Ages=data[(data['age']>=40) & (data['age']<55)]
Elderly_Ages=data[(data['age']>55)]
print('Young Ages =',len(Young_Ages))
print('Middle Ages =',len(Middle_Ages))
print('Elderly Ages =',len(Elderly_Ages))

    Young Ages = 57
    Middle Ages = 419
    Elderly Ages = 519
```

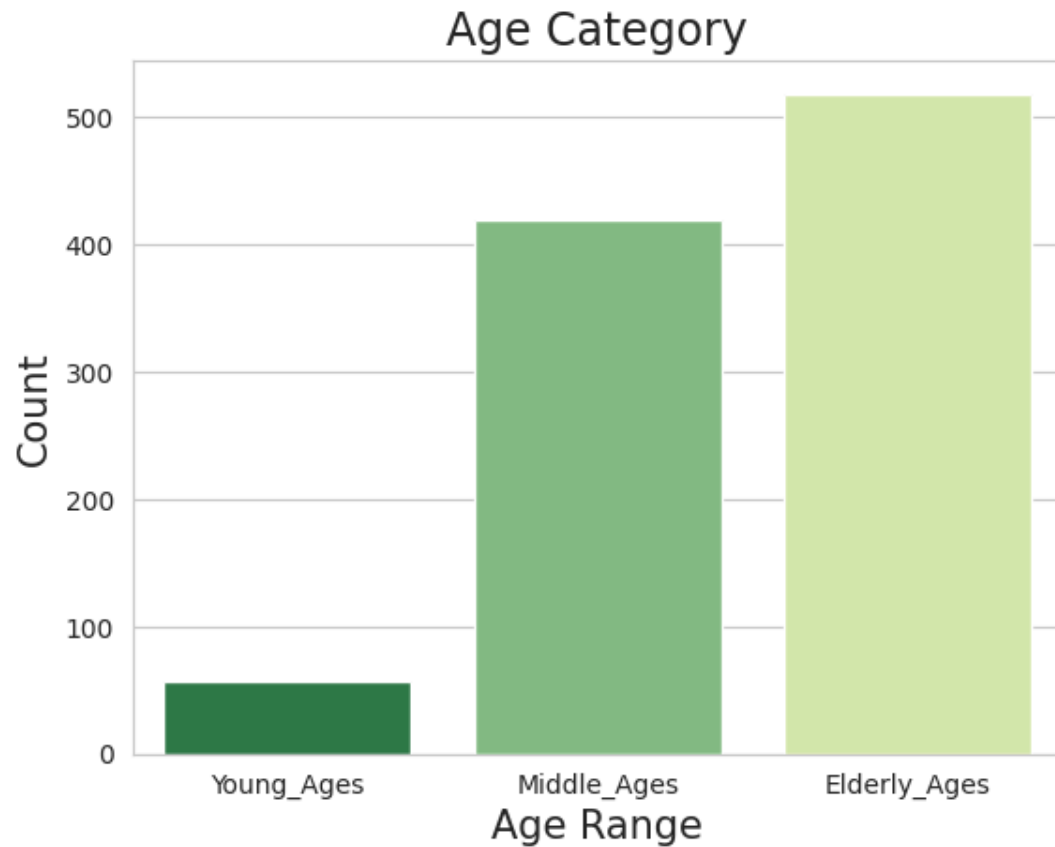
### #Bar Plot Creation of Age Category using Matplotlib and Seaborn

```
sns.barplot(x=['Young_Ages','Middle_Ages','Elderly_Ages'], y=[len(Young_Ages), len(Middle_Ages), len(Elderly_Ages)], palette='YlGn_r')
plt.title('Age Category', fontsize=17)
plt.xlabel('Age Range', fontsize=15)
plt.ylabel('Count', fontsize=15)
plt.show()
```

```
<ipython-input-17-5f1f8f273de1>:3: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`
```

```
sns.barplot(x=['Young_Ages', 'Middle_Ages', 'Elderly_Ages'], y=[len(Young_Ages), len(Middle_Ages), len(Elderly_Ages)], palette='YlGn_r')
```





```
#Converting Numerical Data into Categorical Data
```

```
def gender(row):  
    if row==1:  
        return 'Male'  
    elif row==0:  
        return 'Female'
```

```
#Applying converted data into our dataset with new column - sex1
```

```
data['sex1']=data['sex'].apply(gender)
data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Heart_Disease	sex1	
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	Absence	Male	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	Absence	Male	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	Absence	Male	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	Absence	Male	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	Absence	Female	

Next steps:

[Generate code with data](#)
[View recommended plots](#)

```
#Converting Numerical Data into Categorical Data
```

```
def age_range(row):
    if row>=29 and row<40:
        return 'Young Age'
    elif row>=40 and row<55:
        return 'Middle Age'
    elif row>55:
        return 'Elder Age'
```

```
#Applying converted data into our dataset with new column - Age_Range
```

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



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Heart_Disease	sex1	Age_Range
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	Absence	Male	Middle Age
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	Absence	Male	Middle Age
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	Absence	Male	Elder Age
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	Absence	Male	Elder Age
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	Absence	Female	Elder Age



Next steps:

[Generate code with data](#)[View recommended plots](#)

#Swarm Plot Creation of Gender Based Age Category using Matplotlib and Seaborn

```

plt.figure(figsize=(10,7))
sns.swarmplot(x='Age_Range', y='age', hue='sex1', data=data, order=['Young Age','Middle Age','Elder Age'], palette='Oranges_r')
plt.title('Gender Based Age Category', fontsize=17)
plt.xlabel('Age Category', fontsize=15)
plt.ylabel('Age', fontsize=15)
plt.show()

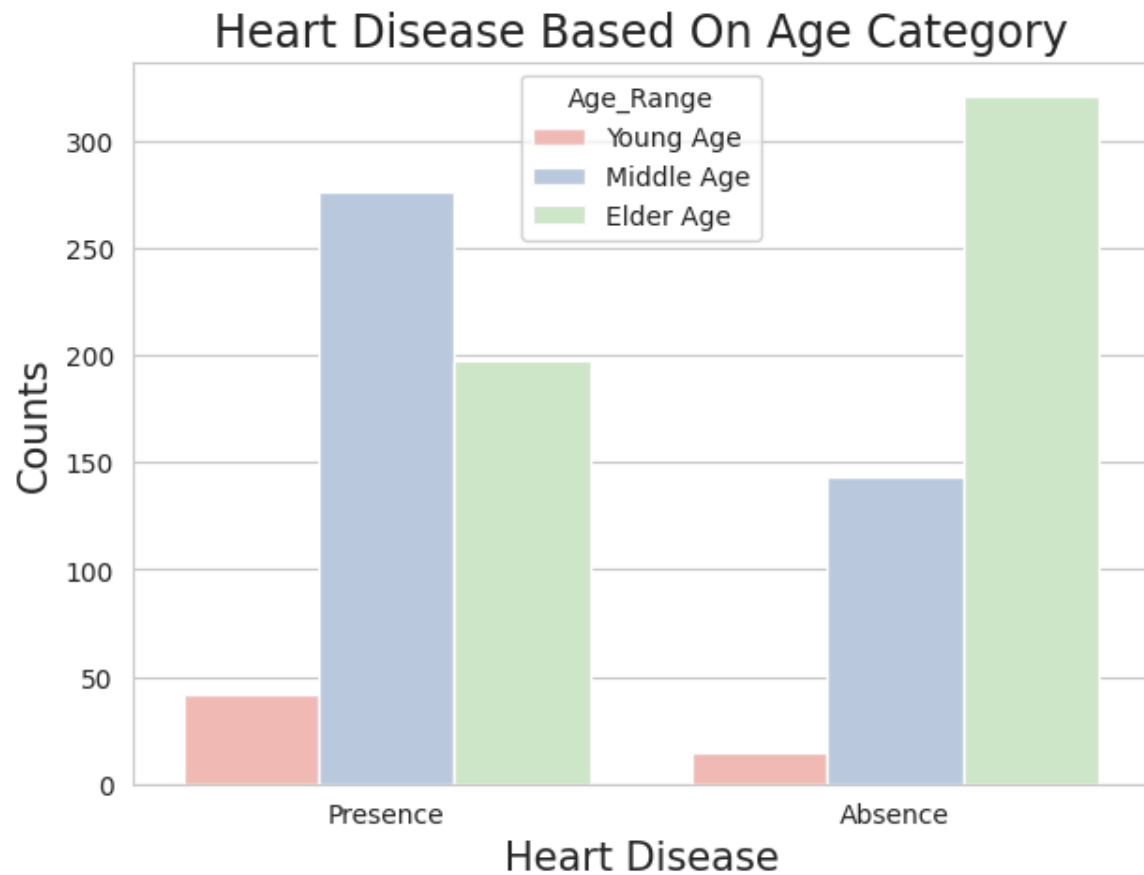
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398: UserWarning: 10.4% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398: UserWarning: 13.8% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398: UserWarning: 23.5% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
```



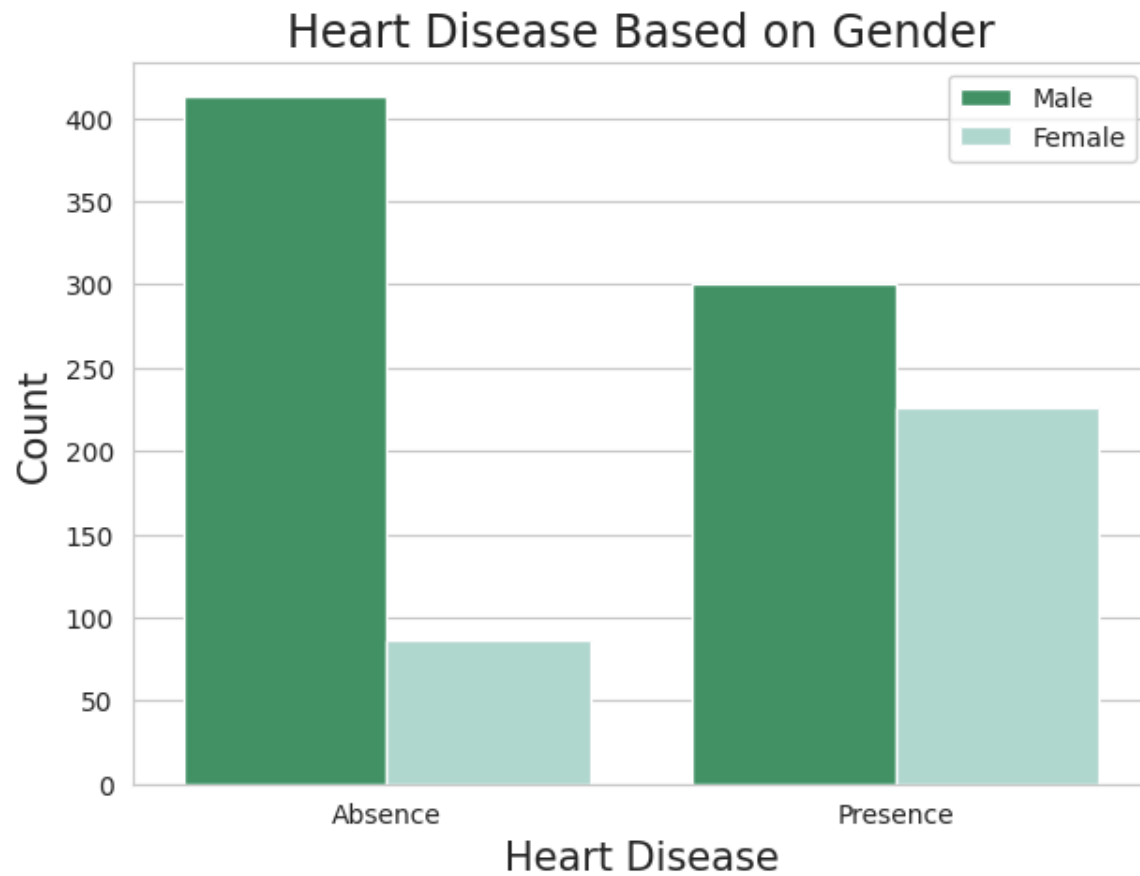
#Count Plot Creation of Heart Disease Based On Age Category using Matplotlib and Seaborn

```
plt.figure(figsize=(7,5))
hue_order=['Young Age', 'Middle Age', 'Elder Age']
sns.countplot(x='Heart_Disease', hue='Age_Range', data=data, order=['Presence','Absence'], hue_order=hue_order, palette='Pastel1')
plt.title('Heart Disease Based On Age Category', fontsize=17)
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Counts', fontsize=15)
plt.show()
```



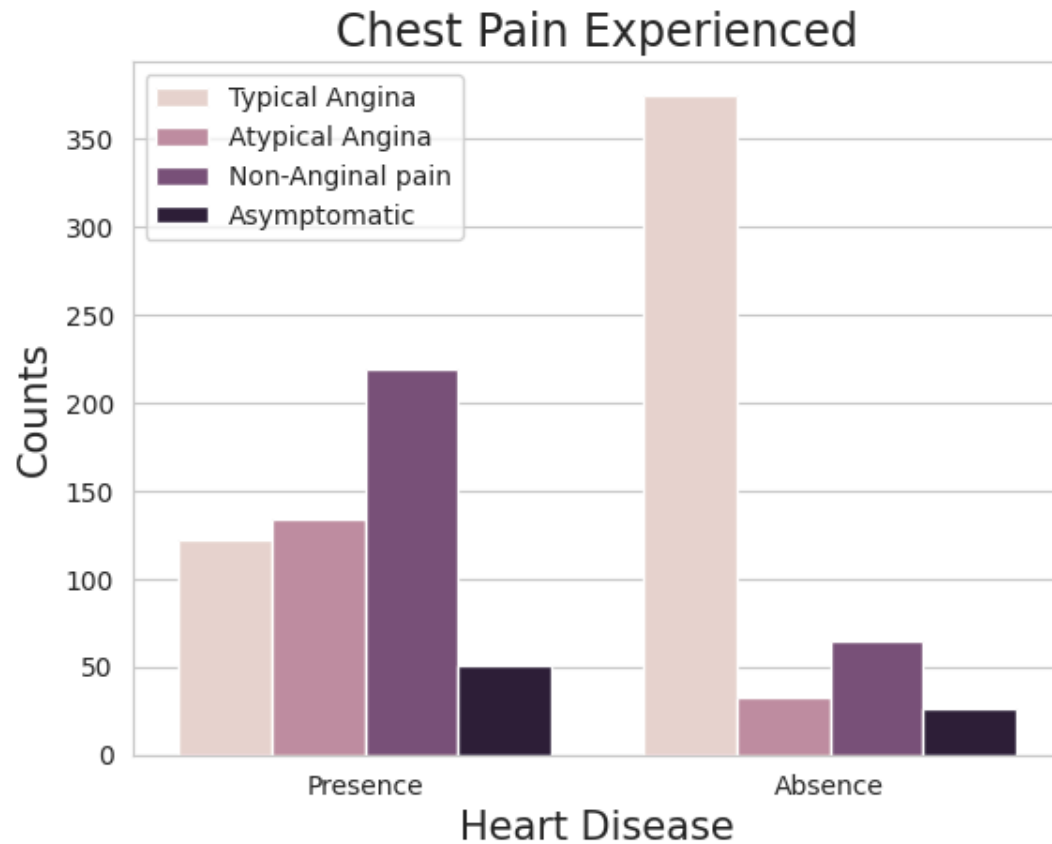
```
#Count Plot Creation of Heart Disease Based on Gender using Matplotlib and Seaborn
```

```
plt.figure(figsize=(7,5))  
sns.countplot(x=data['Heart_Disease'], hue='sex1', data=data, palette='BuGn_r')  
plt.xlabel('Heart Disease', fontsize=15)  
plt.ylabel('Count',fontsize=15)  
plt.legend(labels=['Male','Female'])  
plt.title('Heart Disease Based on Gender',fontsize=17)  
plt.show()
```



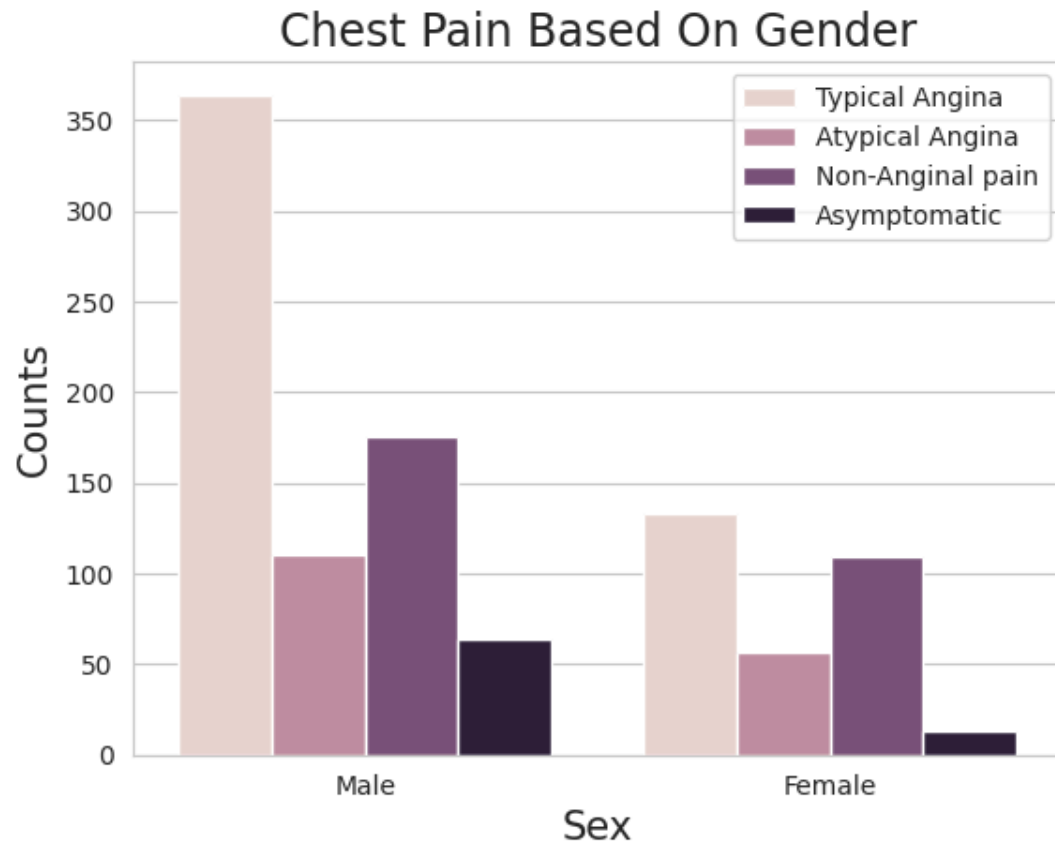
#Count Plot Creation of Chest Pain Experienced using Matplotlib and Seaborn

```
sns.countplot(x=data['Heart_Disease'], hue='cp', data=data, order=['Presence', 'Absence'])  
plt.title('Chest Pain Experienced', fontsize=17)  
plt.xlabel('Heart Disease', fontsize=15)  
plt.ylabel('Counts', fontsize=15)  
plt.legend(labels=['Typical Angina', 'Atypical Angina', 'Non-Anginal pain', 'Asymptomatic'])  
plt.show()
```



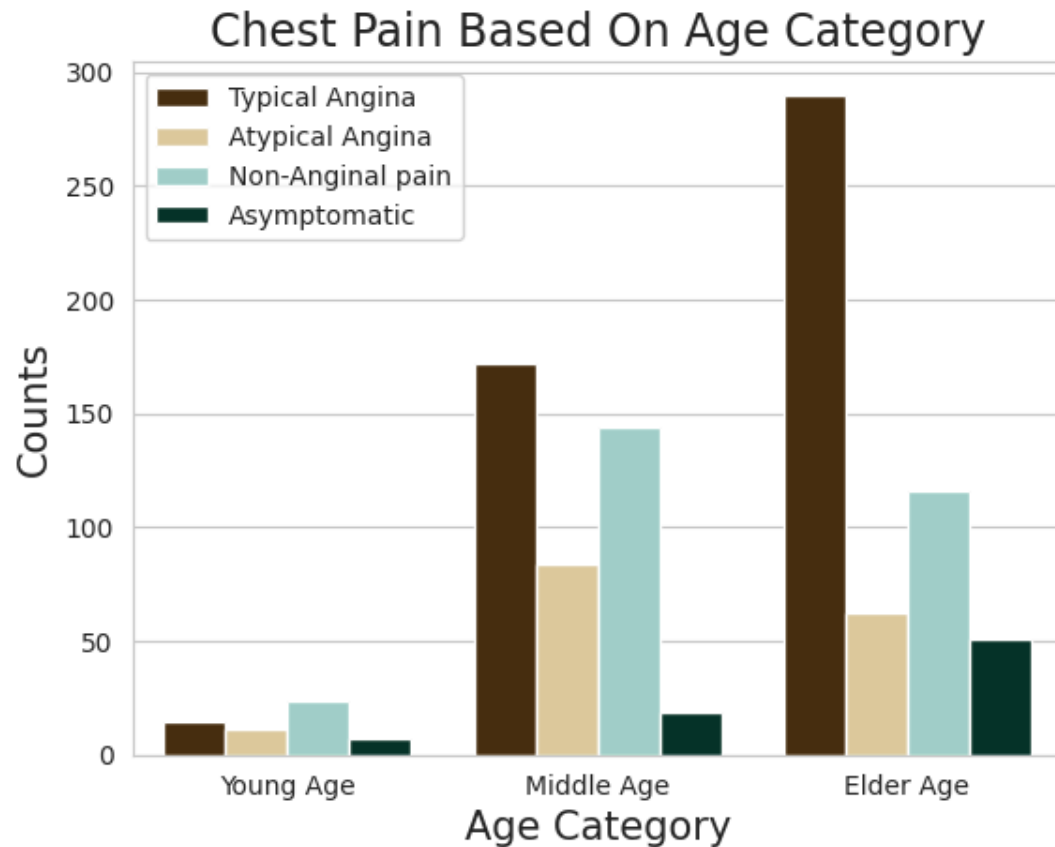
```
#Count Plot Creation of Chest Pain Based On Gender using Matplotlib and Seaborn
```

```
sns.countplot(x=data['sex1'], hue='cp', data=data)
plt.title('Chest Pain Based On Gender', fontsize=17)
plt.xlabel('Sex', fontsize=15)
plt.ylabel('Counts', fontsize=15)
plt.legend(labels=['Typical Angina', 'Atypical Angina', 'Non-Anginal pain', 'Asymptomatic'])
plt.show()
```



```
#Count Plot Creation of Chest Pain Based On Age Category using Matplotlib and Seaborn
```

```
sns.countplot(x=data['Age_Range'], hue='cp', data=data, order=['Young Age', 'Middle Age', 'Elder Age'], palette='BrBG')  
plt.title('Chest Pain Based On Age Category', fontsize=17)  
plt.xlabel('Age Category', fontsize=15)  
plt.ylabel('Counts', fontsize=15)  
plt.legend(labels=['Typical Angina', 'Atypical Angina', 'Non-Anginal pain', 'Asymptomatic'])  
plt.show()
```



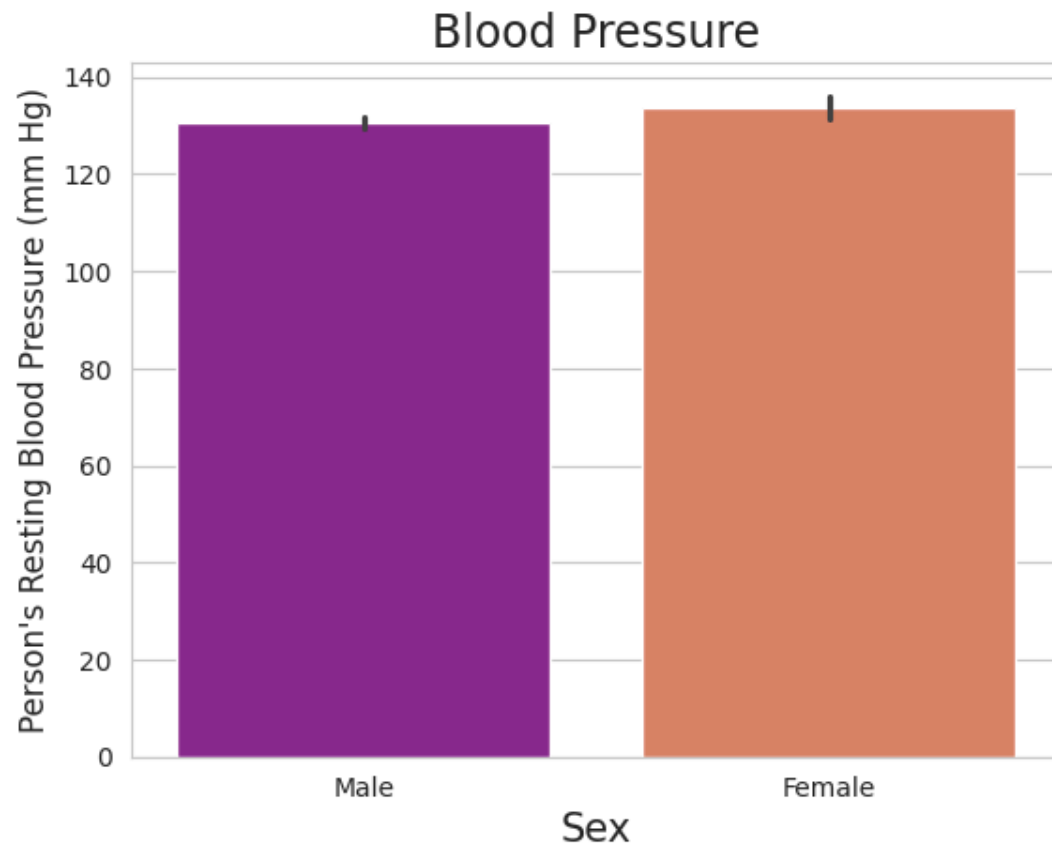
```
#Bar Plot Creation of Person's Resting Blood Pressure (mm Hg) using Matplotlib and Seaborn
```

```
sns.barplot(x='sex1', y='trestbps', data=data, palette='plasma')  
plt.title("Blood Pressure", fontsize=17)  
plt.xlabel('Sex',fontsize=15)  
plt.ylabel("Person's Resting Blood Pressure (mm Hg)", fontsize=12)  
plt.show()
```

<ipython-input-29-37e940b55034>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(x='sex1', y='trestbps', data=data, palette='plasma')
```





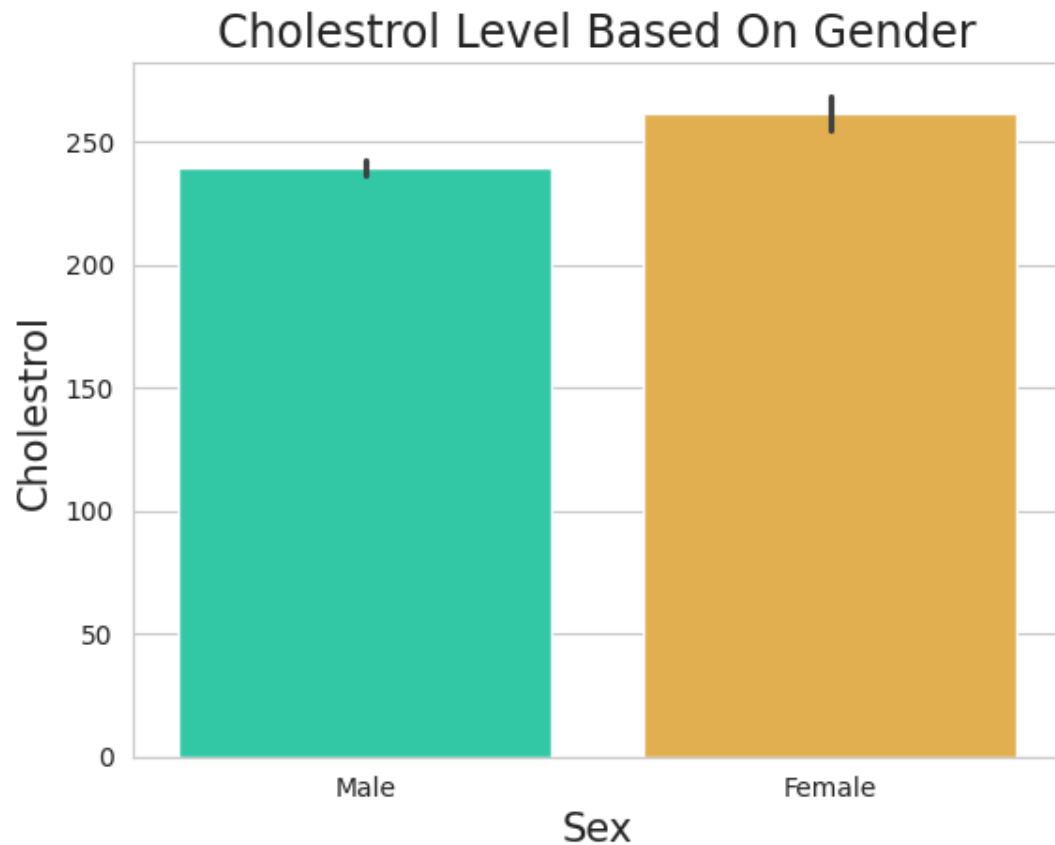
```
#Bar Plot Creation of Cholestrol Level Based On Gender using Matplotlib and Seaborn
```

```
sns.barplot(x='sex1', y='chol', data=data, palette='turbo')  
plt.title("Cholestrol Level Based On Gender", fontsize=17)  
plt.xlabel('Sex',fontsize=15)  
plt.ylabel("Cholestrol", fontsize=15)  
plt.show()
```

<ipython-input-30-4fe2024e27b1>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(x='sex1', y='chol', data=data, palette='turbo')
```



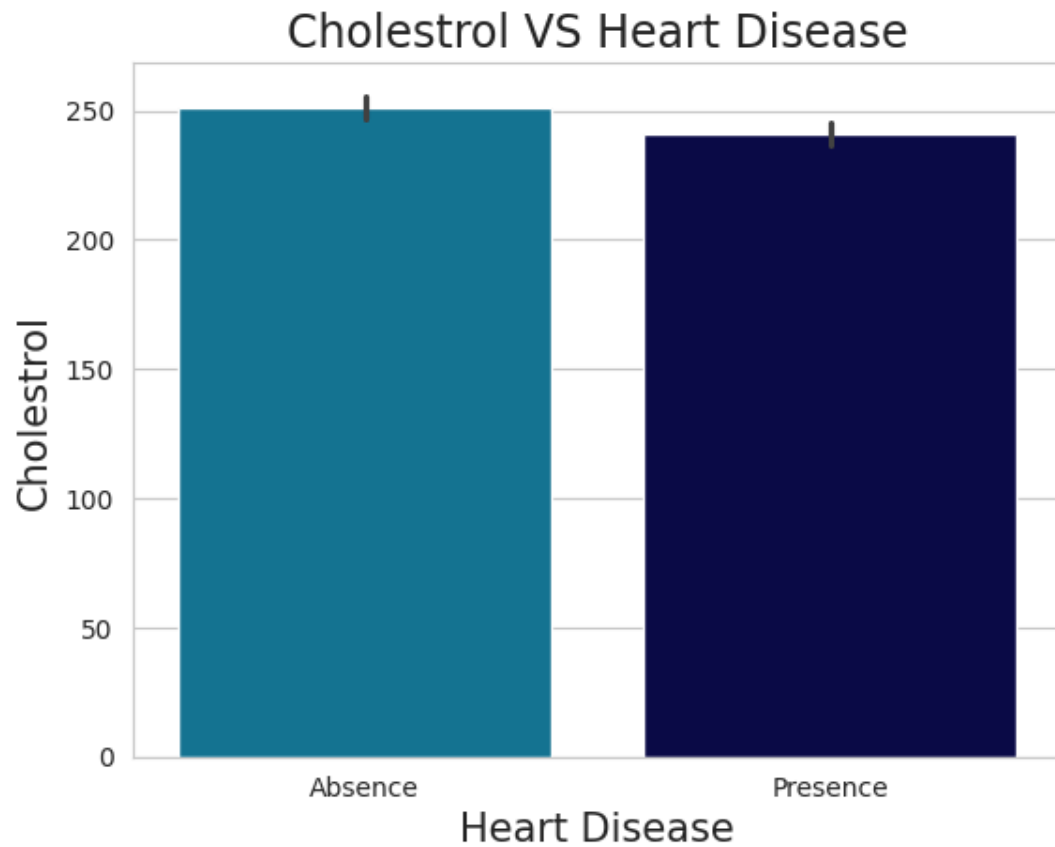
```
#Bar Plot Creation of Cholestrol VS Heart Disease using Matplotlib and Seaborn
```

```
sns.barplot(x='Heart_Disease', y='chol', data=data, palette='ocean_r')  
plt.title('Cholestrol VS Heart Disease', fontsize=17)  
plt.xlabel('Heart Disease', fontsize=15)  
plt.ylabel('Cholestrol', fontsize=15)  
plt.show()
```

<ipython-input-31-8bf9809f64cc>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(x='Heart_Disease', y='chol', data=data, palette='ocean_r')
```



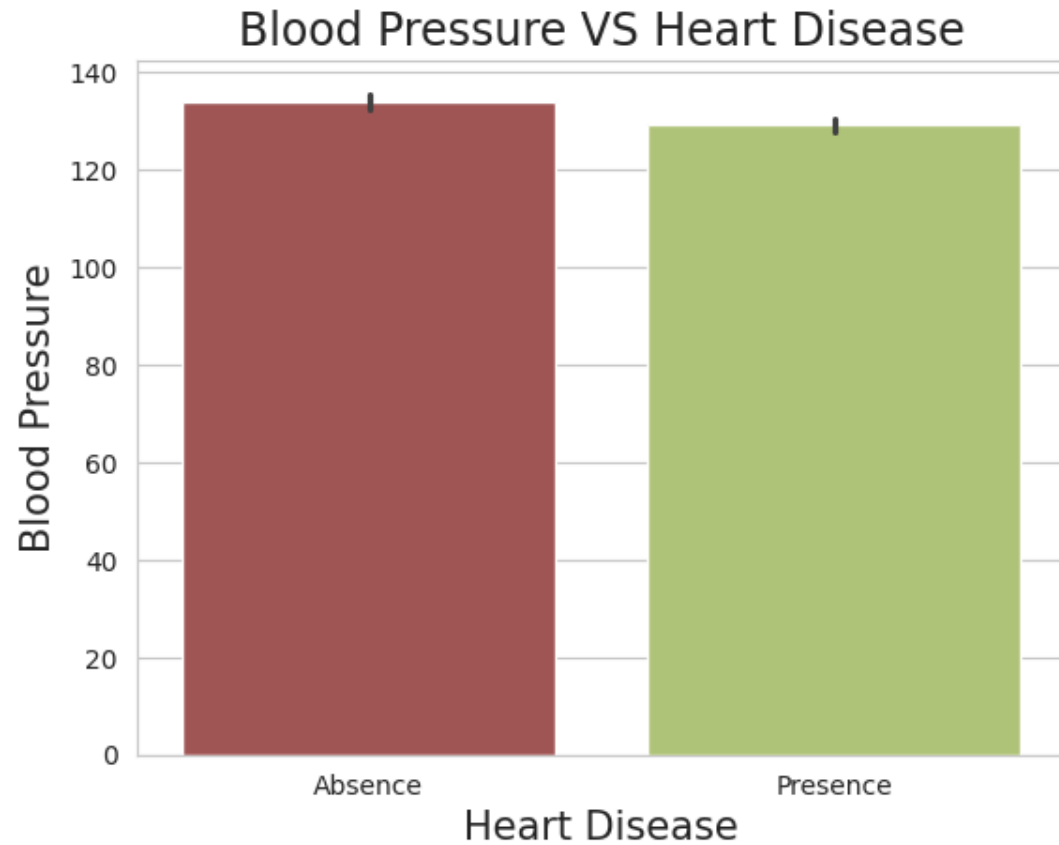
```
#Bar Plot Creation of Blood Pressure VS Heart Disease using Matplotlib and Seaborn
```

```
sns.barplot(x='Heart_Disease', y='trestbps', data=data, palette='tab20b_r')  
plt.title('Blood Pressure VS Heart Disease', fontsize=17)  
plt.xlabel('Heart Disease', fontsize=15)  
plt.ylabel('Blood Pressure', fontsize=15)  
plt.show()
```

<ipython-input-32-df099a8656be>:3: FutureWarning:

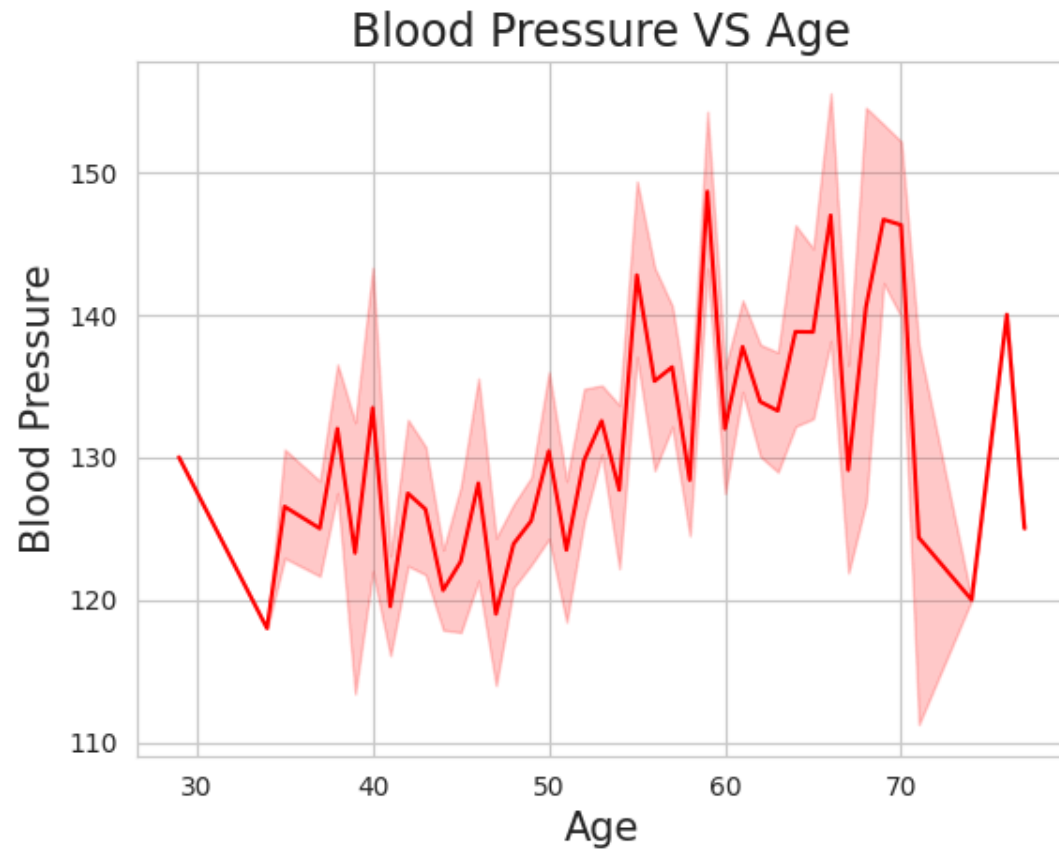
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(x='Heart_Disease', y='trestbps', data=data, palette='tab20b_r')
```



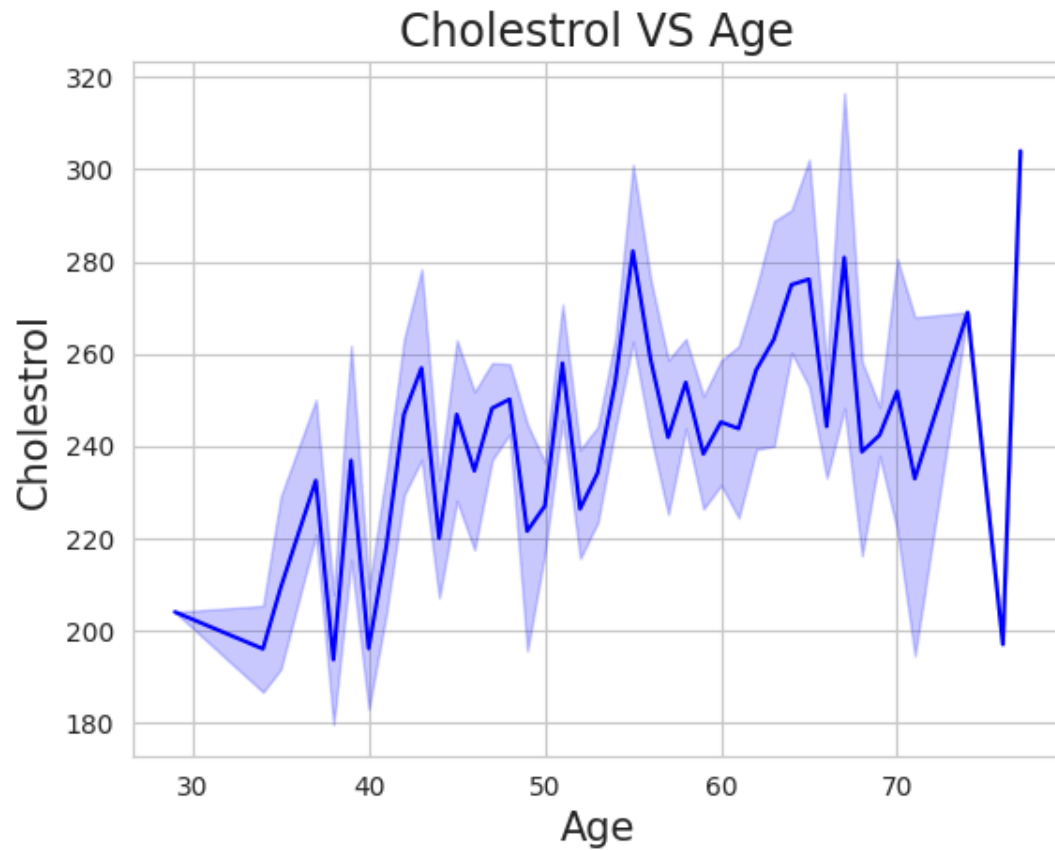
```
#Line Plot Creation of Blood Pressure VS Age using Matplotlib and Seaborn
```

```
sns.lineplot(x='age', y='trestbps', data=data, color='r')  
plt.title('Blood Pressure VS Age', fontsize=17)  
plt.xlabel('Age', fontsize=15)  
plt.ylabel('Blood Pressure', fontsize=15)  
plt.show()
```



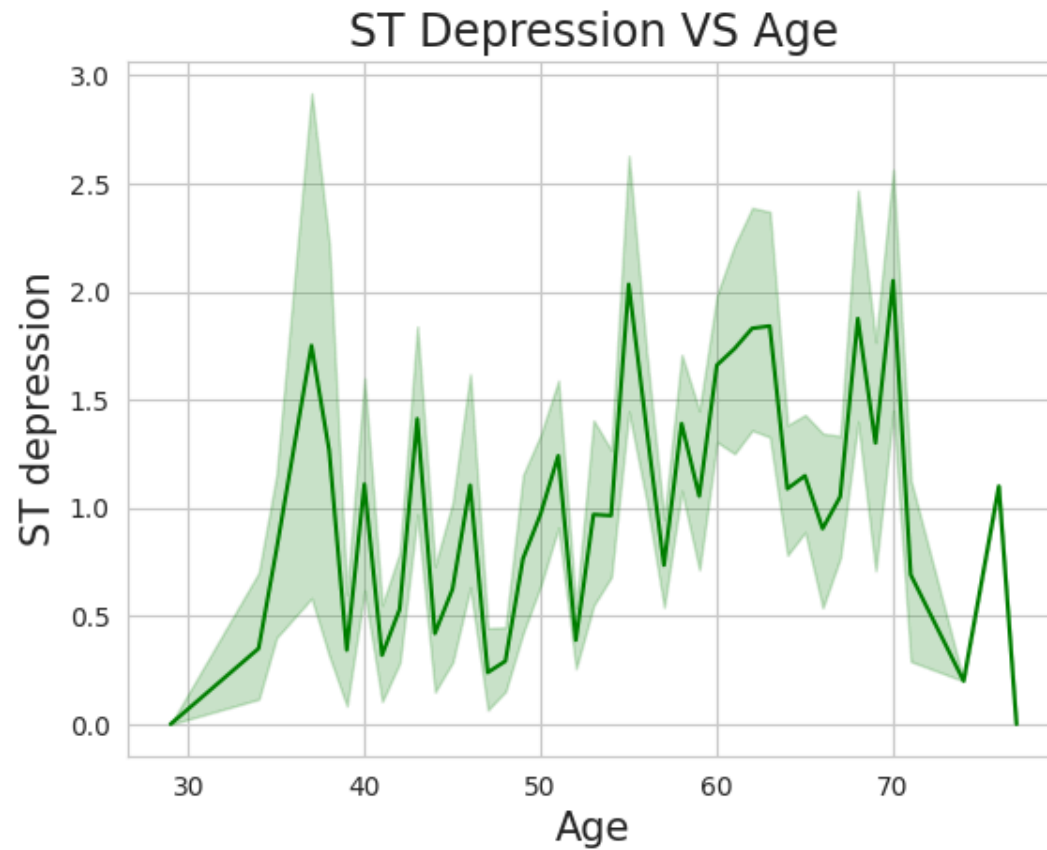
```
#Line Plot Creation of Cholestrol VS Age using Matplotlib and Seaborn
```

```
sns.lineplot(x='age', y='chol', data=data, color='b')  
plt.title('Cholestrol VS Age', fontsize=17)  
plt.xlabel('Age', fontsize=15)  
plt.ylabel('Cholestrol', fontsize=15)  
plt.show()
```



```
#Line Plot Creation of ST Depression VS Age using Matplotlib and Seaborn
```

```
sns.lineplot(x='age', y='oldpeak', data=data, color='g')  
plt.title('ST Depression VS Age', fontsize=17)  
plt.xlabel('Age', fontsize=15)  
plt.ylabel('ST depression', fontsize=15)  
plt.show()
```



```
#Bar Plot Creation of ST depression VS Heart Disease using Matplotlib and Seaborn
```

```
sns.barplot(x='sex1', y='oldpeak', data=data, palette='twilight_r')  
plt.title('ST depression VS Heart Disease', fontsize=17)  
plt.xlabel('Sex', fontsize=15)  
plt.ylabel('ST depression', fontsize=15)  
plt.show()
```

ython-input-36-df7084e7e09e>:3: FutureWarning:

Using `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False`.

```
sns.barplot(x='sex1', y='oldpeak', data=data, palette='twilight_r')
```



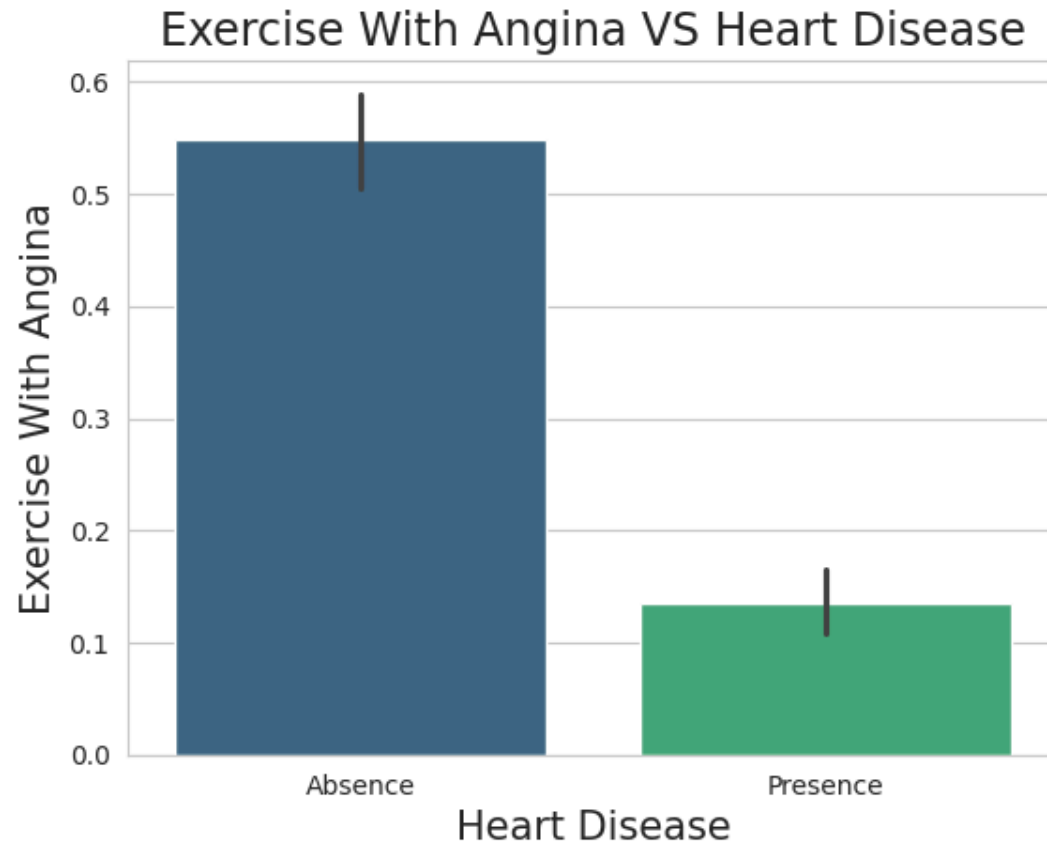
#Bar Plot Creation of Exercise With Angina VS Heart Disease using Matplotlib and Seaborn

```
sns.barplot(x='Heart_Disease', y='exang', data=data, palette='viridis')
plt.title('Exercise With Angina VS Heart Disease', fontsize=17)
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Exercise With Angina', fontsize=15)
plt.show()
```

<ipython-input-37-e1ce3acd7cb2>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(x='Heart_Disease', y='exang', data=data, palette='viridis')
```





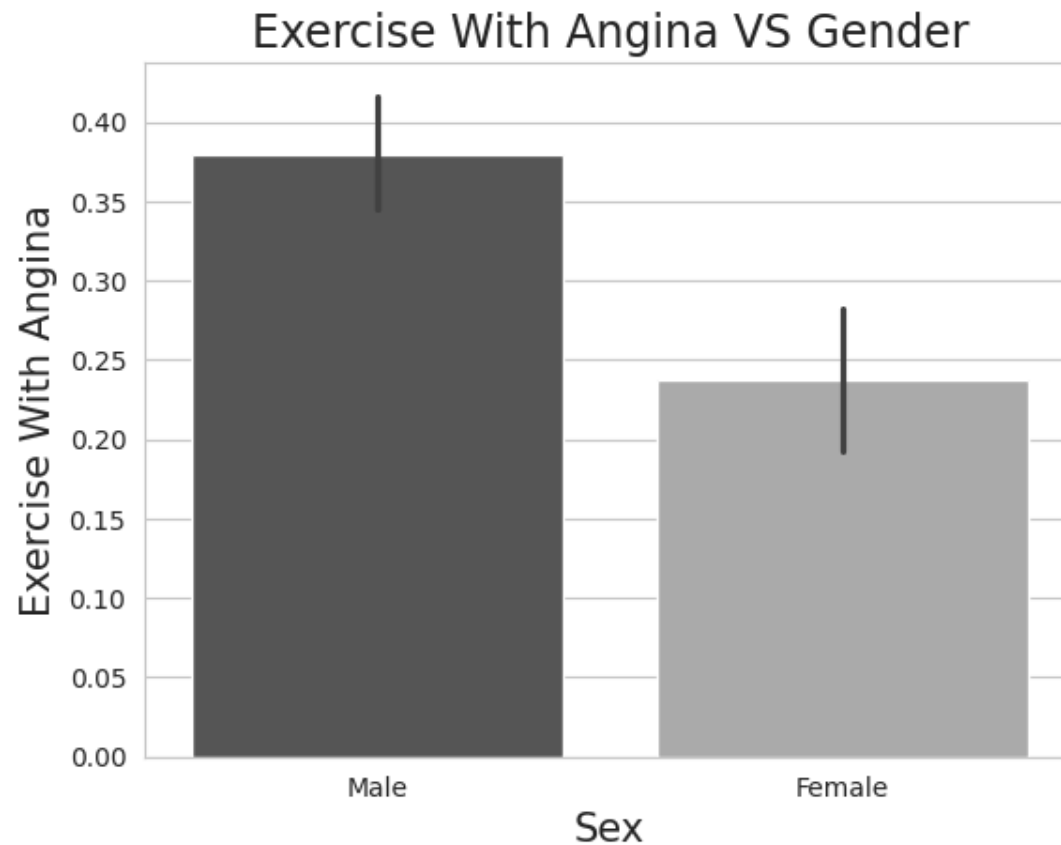
```
#Bar Plot Creation of Exercise With Angina VS Gender using Matplotlib and Seaborn
```

```
sns.barplot(x='sex1', y='exang', data=data, palette='binary_r')  
plt.title('Exercise With Angina VS Gender', fontsize=17)  
plt.xlabel('Sex', fontsize=15)  
plt.ylabel('Exercise With Angina', fontsize=15)  
plt.show()
```

<ipython-input-38-8c8cd9640718>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(x='sex1', y='exang', data=data, palette='binary_r')
```



```
#Bar Plot Creation of Fasting Blood Sugar VS Gender using Matplotlib and Seaborn
```

```
sns.barplot(y='fbs', x='sex1', data=data, palette='hsv')  
plt.title(' Fasting Blood Sugar VS Gender', fontsize=17)  
plt.xlabel('Sex', fontsize=15)  
plt.ylabel('Fasting Blood Sugar', fontsize=15)  
plt.show()
```

<ipython-input-39-b2f433eda42c>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.barplot(y='fbs', x='sex1', data=data, palette='hsv')
```

