

Heart Disease Analysis

Domain - Healthcare

Members - Anusha.C (221121101018) & Jashwanth.G (221121101062)

```
In [44]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set_style('whitegrid')
```

```
In [3]: data=pd.read_csv(r'C:\Users\Jashwanth\OneDrive\Documents\GRT\heart_disease_dataset.csv')
data.head()
```

```
Out[3]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0

```
In [4]: data.columns
```

```
Out[4]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
              'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num'],
              dtype='object')
```

Features in Dataset -

age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, num.

```
In [5]: #Checking NULL Values
data.isnull().sum()
```

```
Out[5]: 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
num            0
dtype: int64
```

```
In [6]: #Percentage of people having Heart Disease
num=data.groupby('num').size()
num
```

```
Out[6]: num
0      164
1      139
dtype: int64
```

```
In [7]: #Converting Numerical Data into Categorical Data
def heart_disease(row):
    if row==0:
        return 'Absence'
    elif row==1:
        return 'Presence'
```

```
In [8]: #Applying converted data into our dataset with new column - Heart_Disease
data['Heart_Disease']=data['num'].apply(heart_disease)
data.head()
```

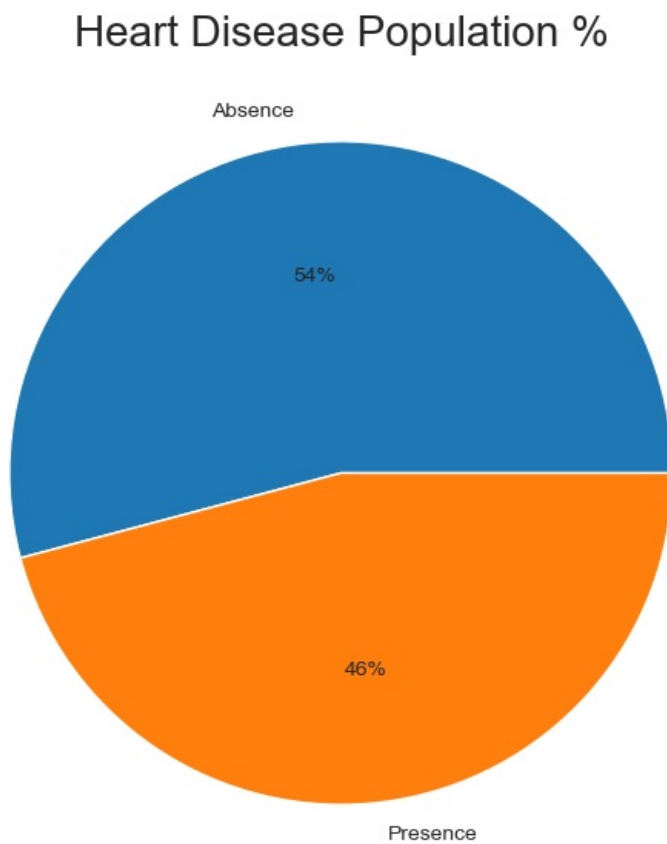
```
Out[8]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	Heart_Disease
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0	Absence
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1	Presence
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1	Presence
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0	Absence
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0	Absence

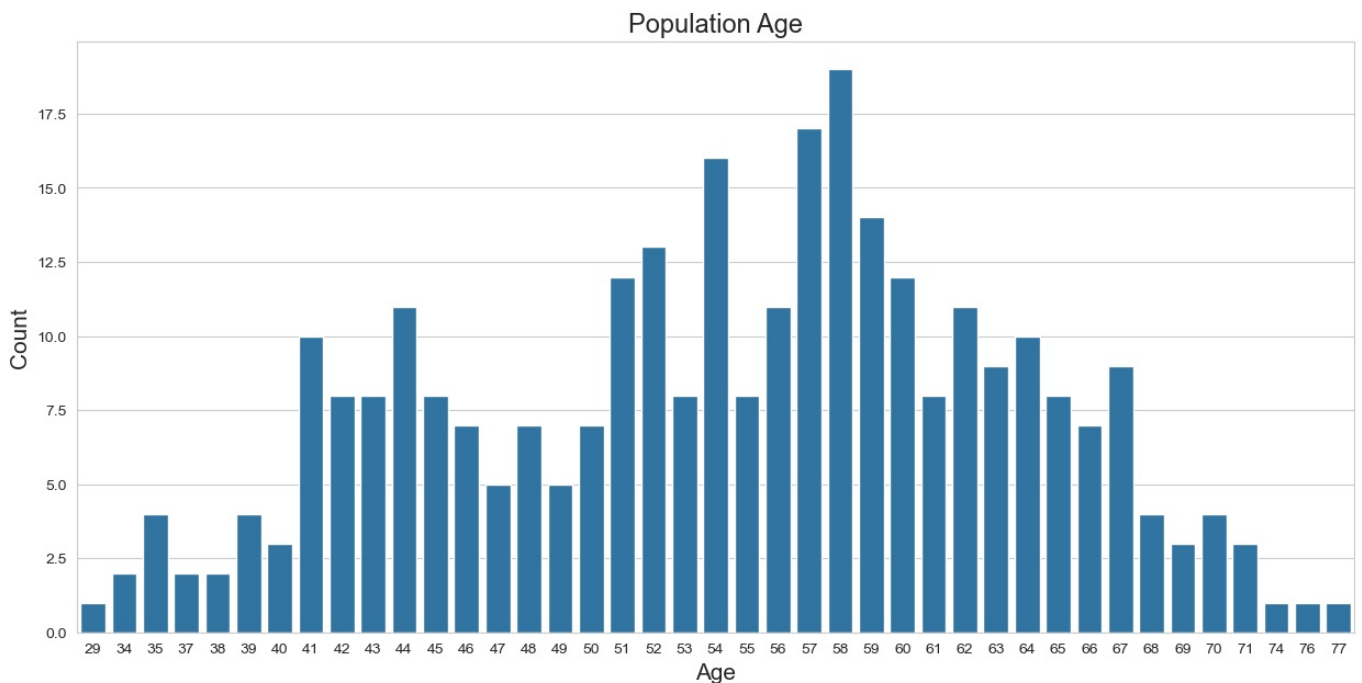
```
In [9]: hd=data.groupby('Heart_Disease')['num'].count()
hd
```

```
Out[9]: Heart_Disease
Absence    164
Presence    139
Name: num, dtype: int64
```

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



```
In [11]: 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()
```



```
In [12]: #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 = 54.43894389438944
```

```
In [13]: #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 = 15
Middle Ages = 128
Elderly Ages = 152
```

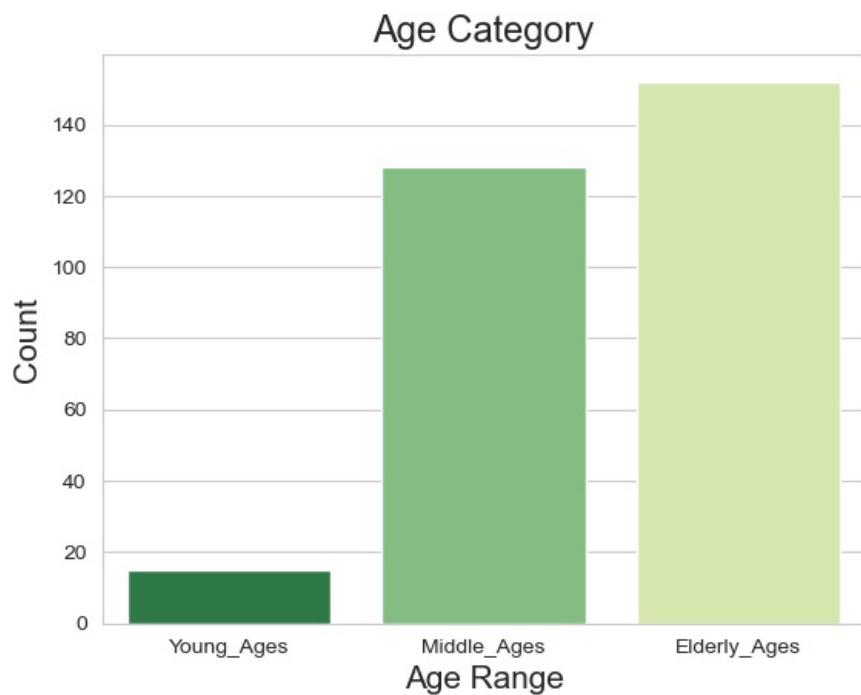
```
In [14]: #Bar Plot Creation of Age Category using Matplotlib and Seaborn

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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\3903353695.py: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=False` for the same effect.

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



```
In [15]: def gender(row):
        if row==1:
            return 'Male'
        elif row==0:
            return 'Female'
```

```
In [16]: #Applying converted data into our dataset with new column - sex1
```

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

```
Out[16]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	Heart_Disease	sex1
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0	Absence	Male
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1	Presence	Male
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1	Presence	Male
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0	Absence	Male
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0	Absence	Female

```
In [17]: #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'
```

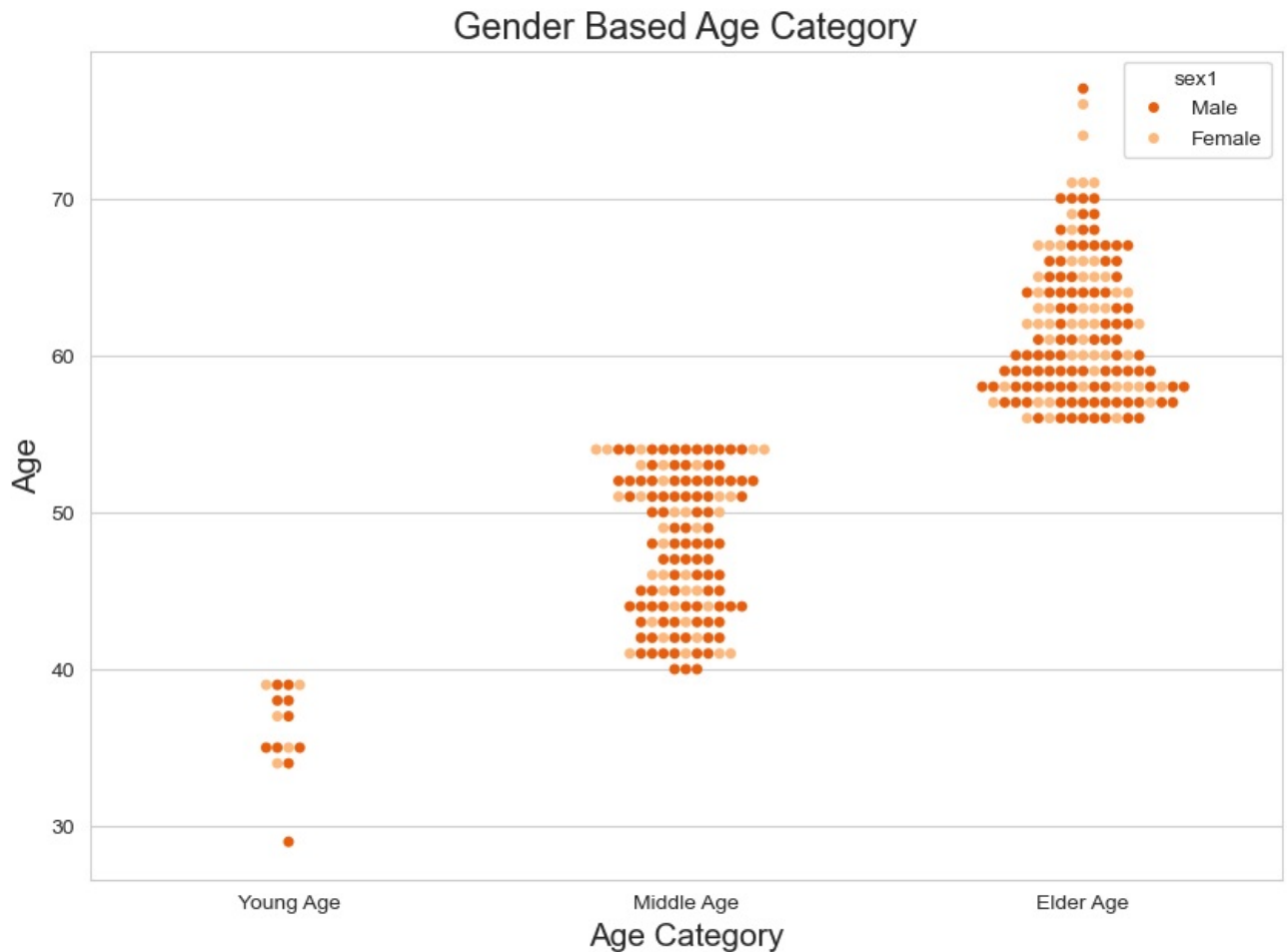
```
In [18]: #Applying converted data into our dataset with new column - Age_Range
```

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

Out[18]:	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	Heart_Disease	sex1	Age_Range
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0	Absence	Male	Elder Age
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1	Presence	Male	Elder Age
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1	Presence	Male	Elder Age
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0	Absence	Male	Young Age
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0	Absence	Female	Middle Age

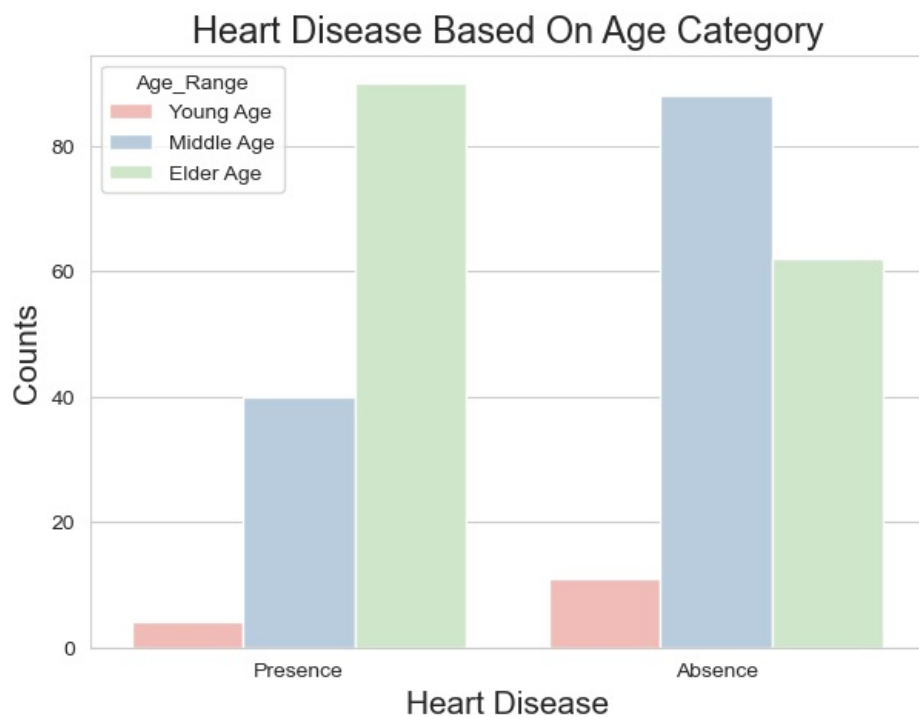
```
In [19]: #Swarm Plot Creation of Gender Based Age Category using Matplotlib and Seaborn

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



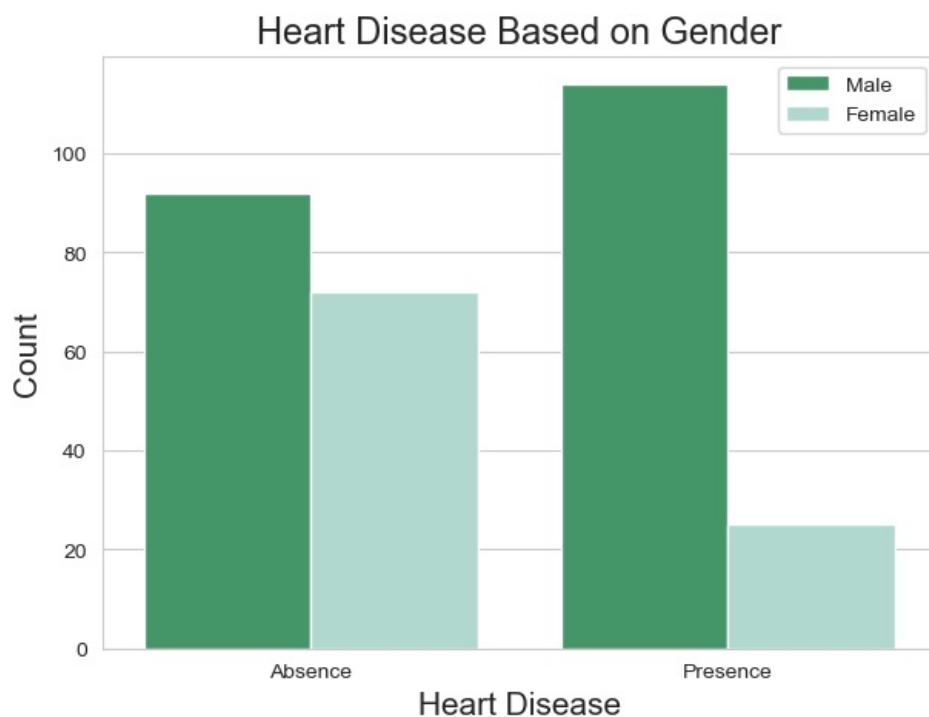
```
In [20]: #Count Plot Creation of Heart Disease Based On Age Category using Matplotlib and Seaborn

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='magma')
plt.title('Heart Disease Based On Age Category', fontsize=17)
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Counts', fontsize=15)
plt.show()
```



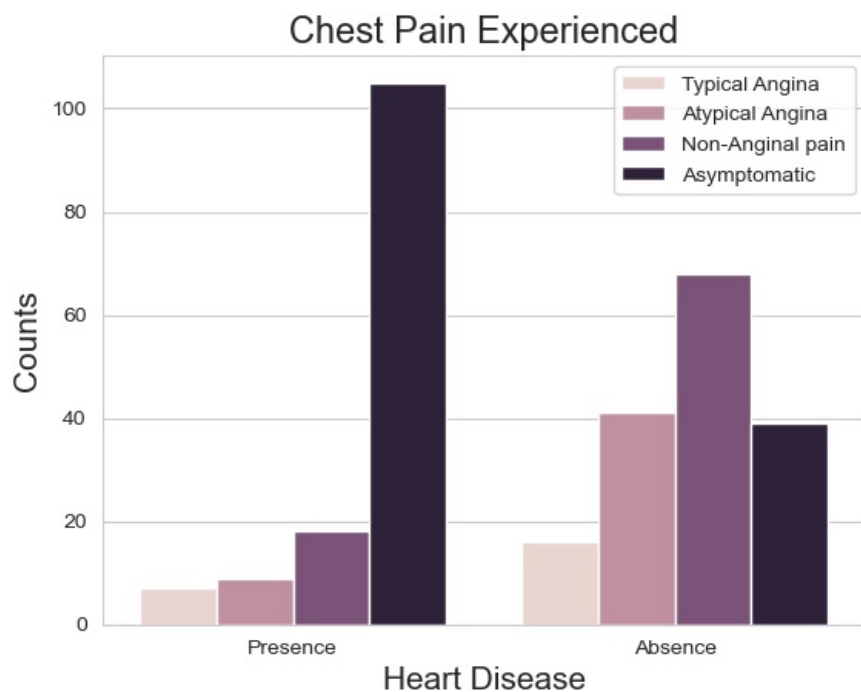
In [21]: *#Count Plot Creation of Heart Disease Based on Gender using Matplotlib and Seaborn*

```
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()
```



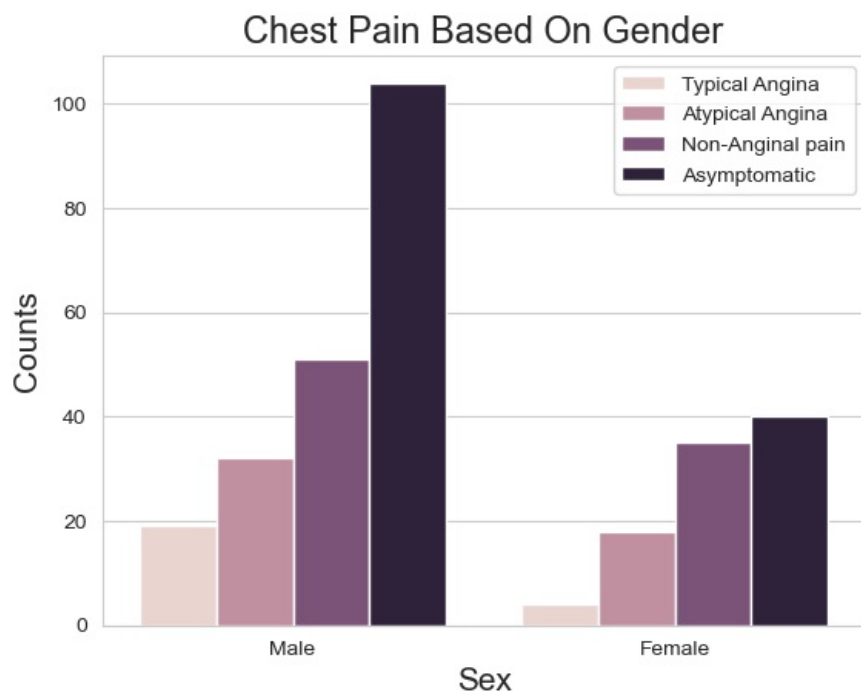
In [22]: *#Count Plot Creation of Chest Pain Experienced using Matplotlib and Seaborn*

```
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()
```



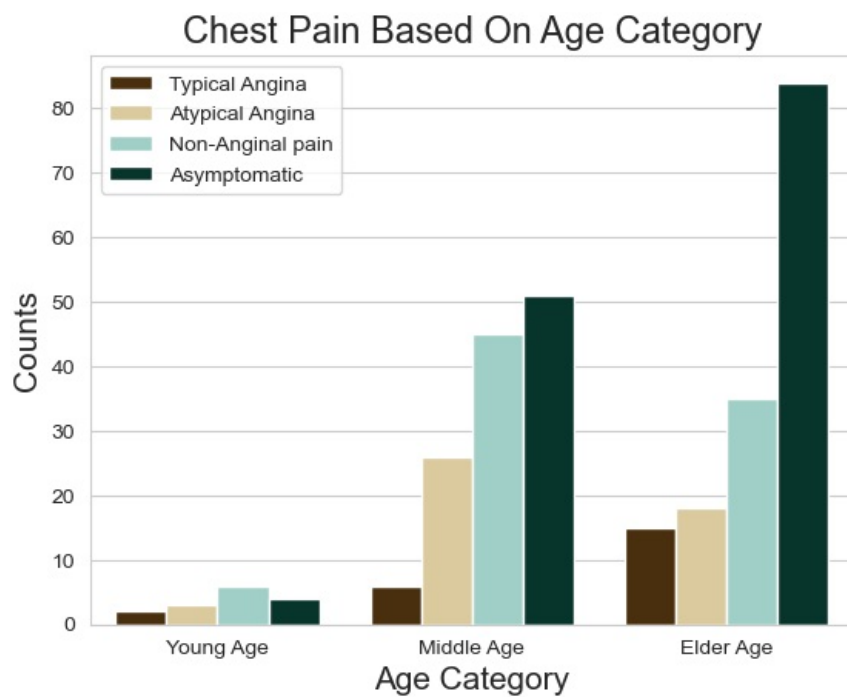
In [23]: #Count Plot Creation of Chest Pain Based On Gender using Matplotlib and Seaborn

```
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()
```



In [24]: #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=
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()
```



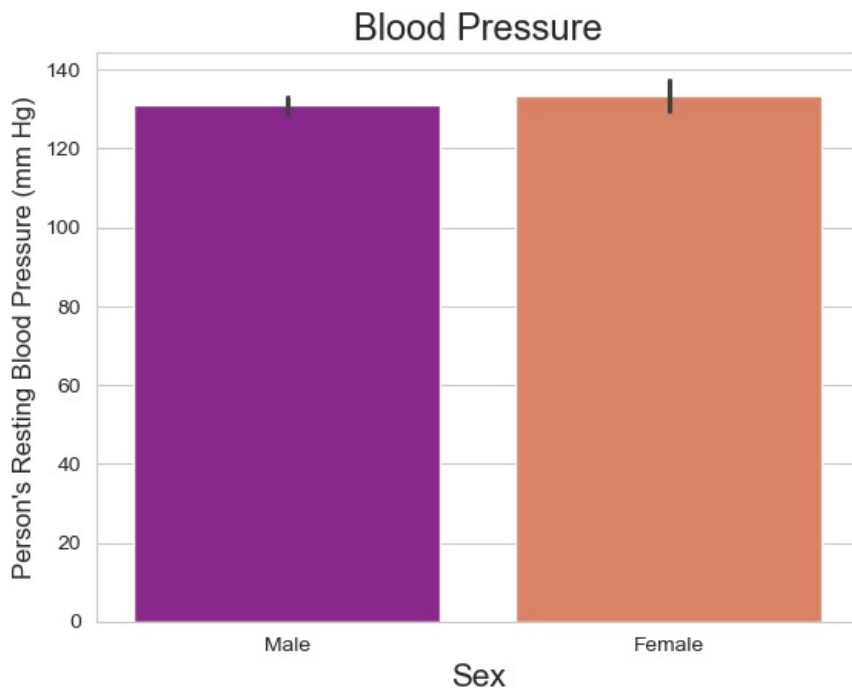
In [25]: *#Bar Plot Creation of Person's Resting Blood Pressure (mm Hg) using Matplotlib and Seaborn*

```
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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\3738778419.py: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=False` for the same effect.

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



In [26]: *#Bar Plot Creation of Cholesterol Level Based On Gender using Matplotlib and Seaborn*

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


C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\2623607158.py: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=False` for the same effect.

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



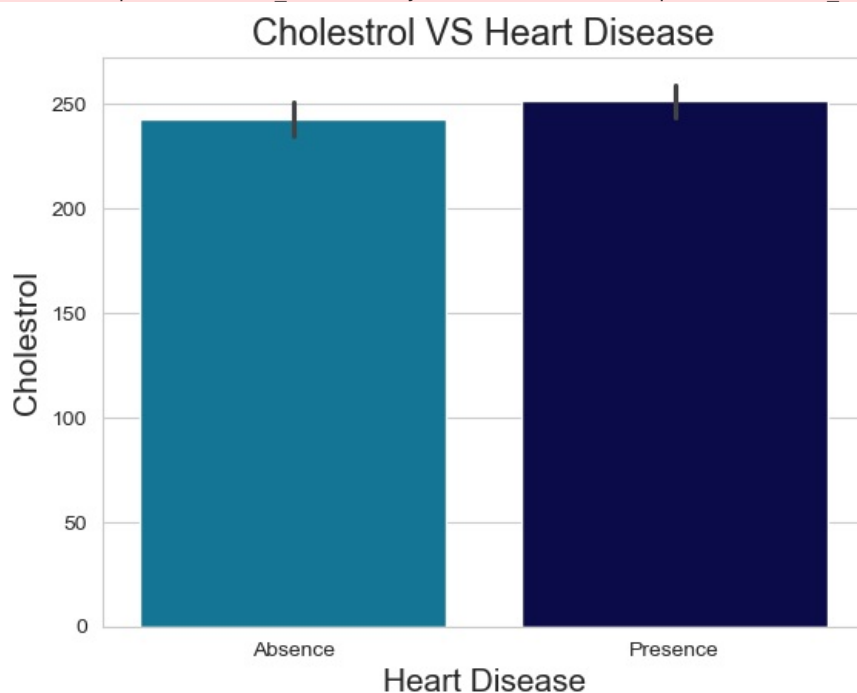
In [27]: *#Bar Plot Creation of Cholestrol VS Heart Disease using Matplotlib and Seaborn*

```
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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\2825866973.py: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=False` for the same effect.

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



In [28]: *#Bar Plot Creation of Blood Pressure VS Heart Disease using Matplotlib and Seaborn*

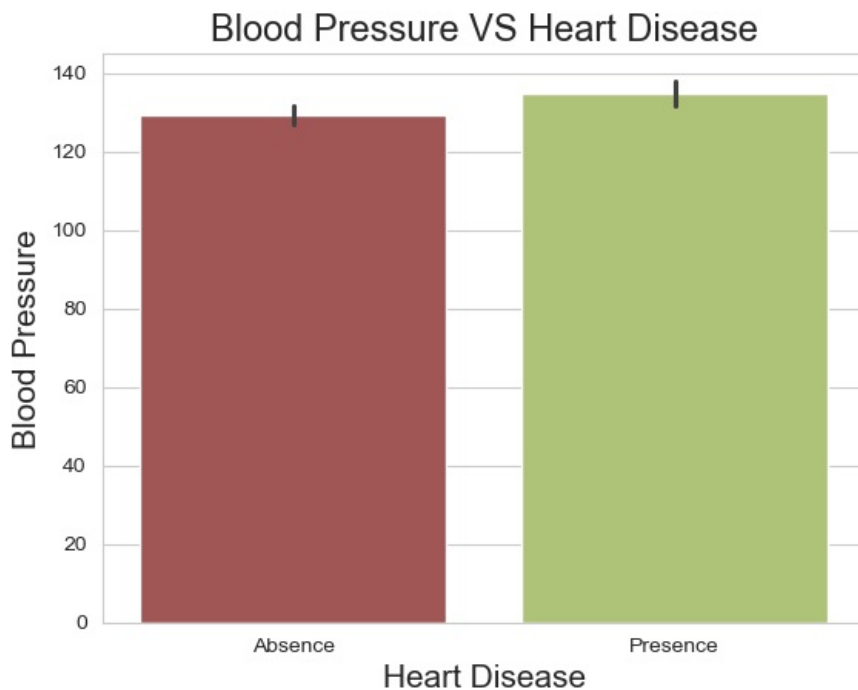
```
sns.barplot(x='Heart_Disease', y='trestbps', data=data, palette='tab20b_r')
plt.title('Blood Pressure VS Heart Disease', fontsize=17)
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Blood Pressure', fontsize=15)
```

```
plt.show()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\2124966136.py:3: FutureWarning:

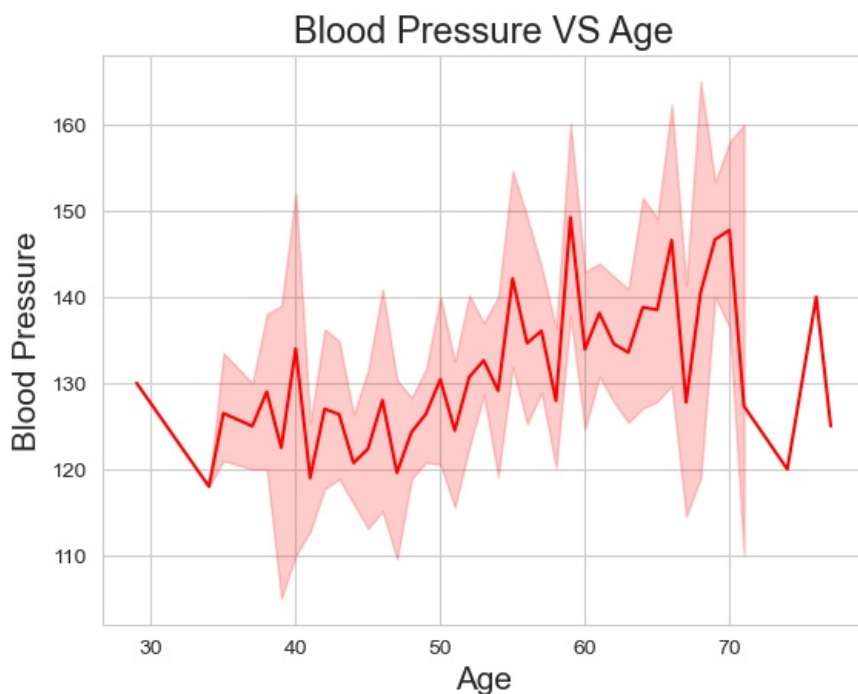
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

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



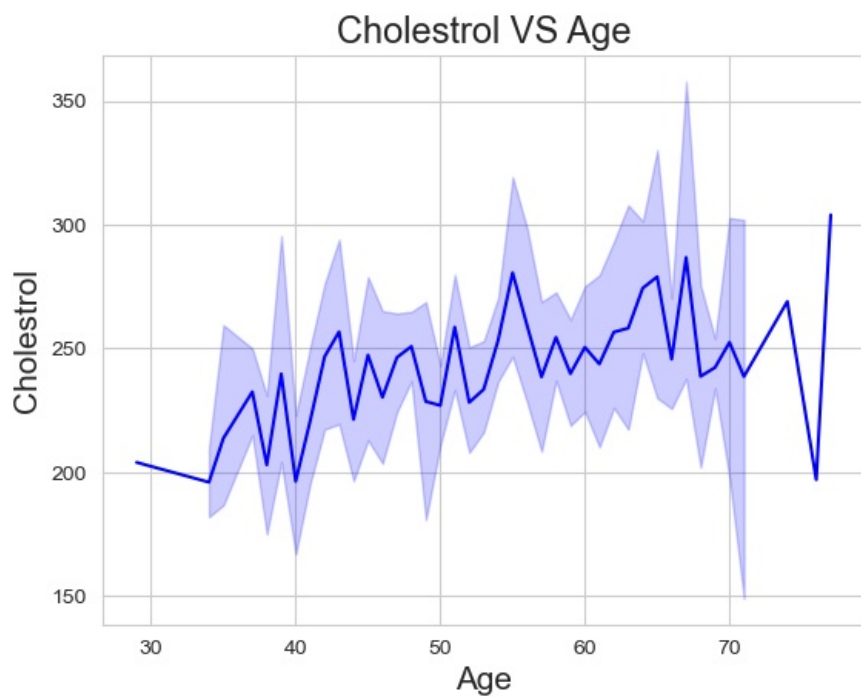
In [29]: #Line Plot Creation of Blood Pressure VS Age using Matplotlib and Seaborn

```
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()
```



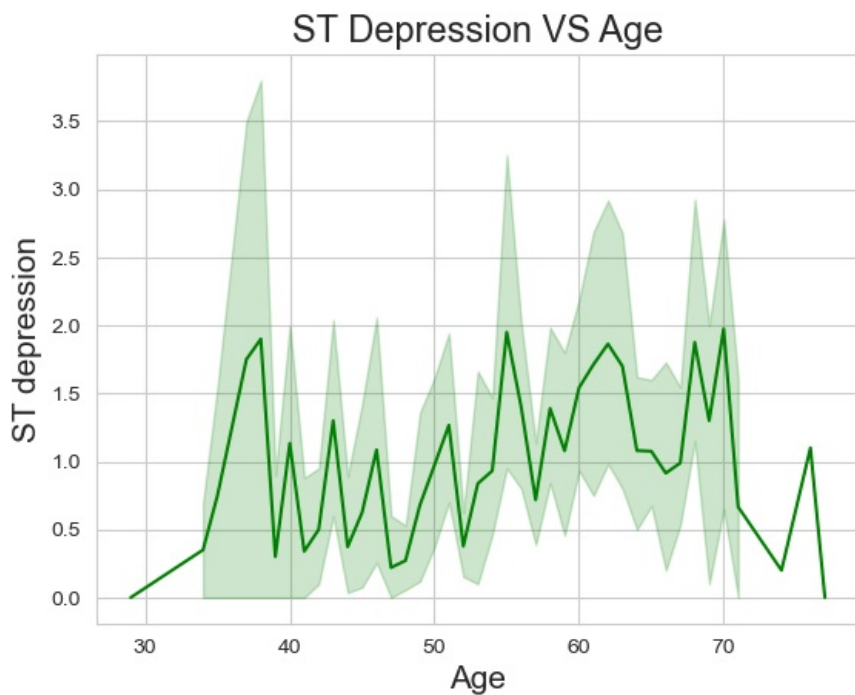
In [30]: #Line Plot Creation of Cholestrol VS Age using Matplotlib and Seaborn

```
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()
```



In [31]: *#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()
```



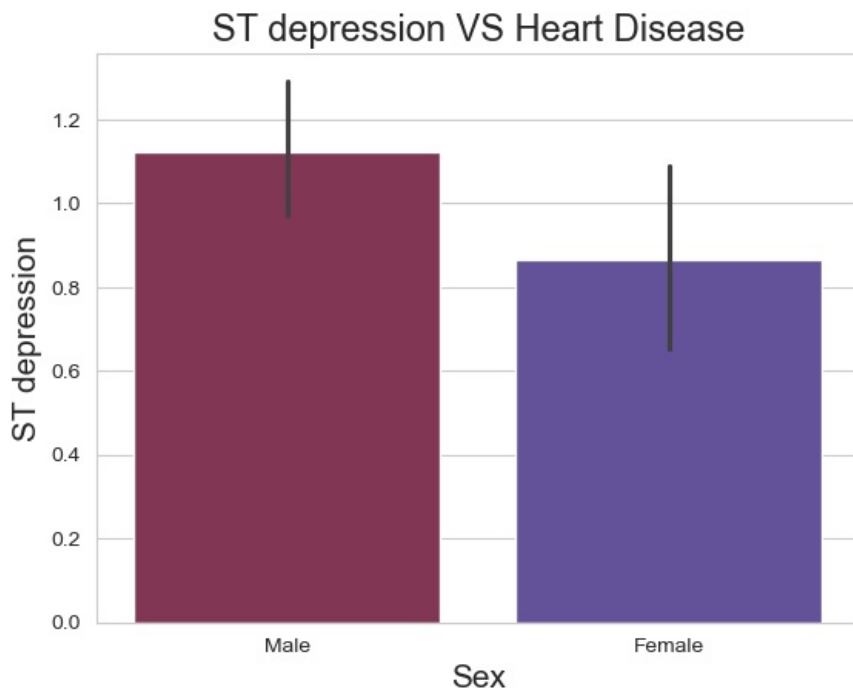
In [32]: *#Bar Plot Creation of ST depression VS Heart Disease using Matplotlib and Seaborn*

```
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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\201252497.py: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=False` for the same effect.

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



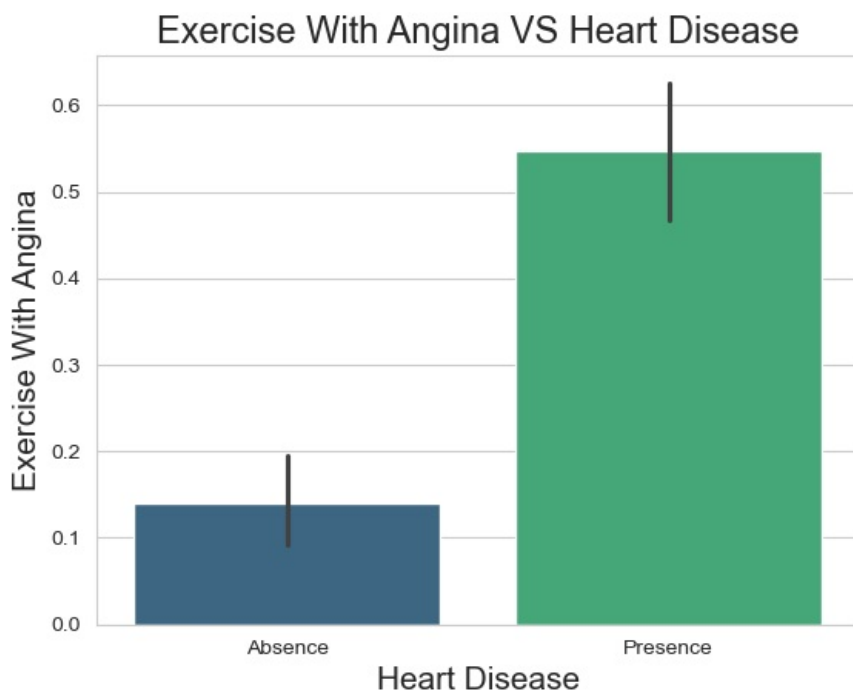
In [33]: #Bar Plot Creation of Exercise With Angina VS Heart Disease using Matplotlib and Seaborn

```
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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\2959183411.py: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=False` for the same effect.

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



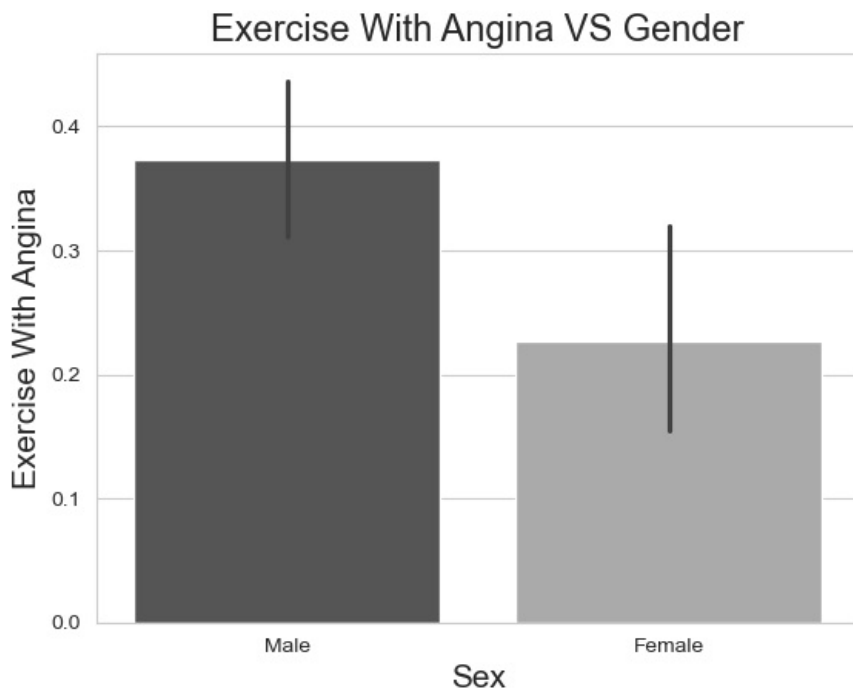
In [34]: #Bar Plot Creation of Exercise With Angina VS Gender using Matplotlib and Seaborn

```
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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\389153480.py: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=False` for the same effect.

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



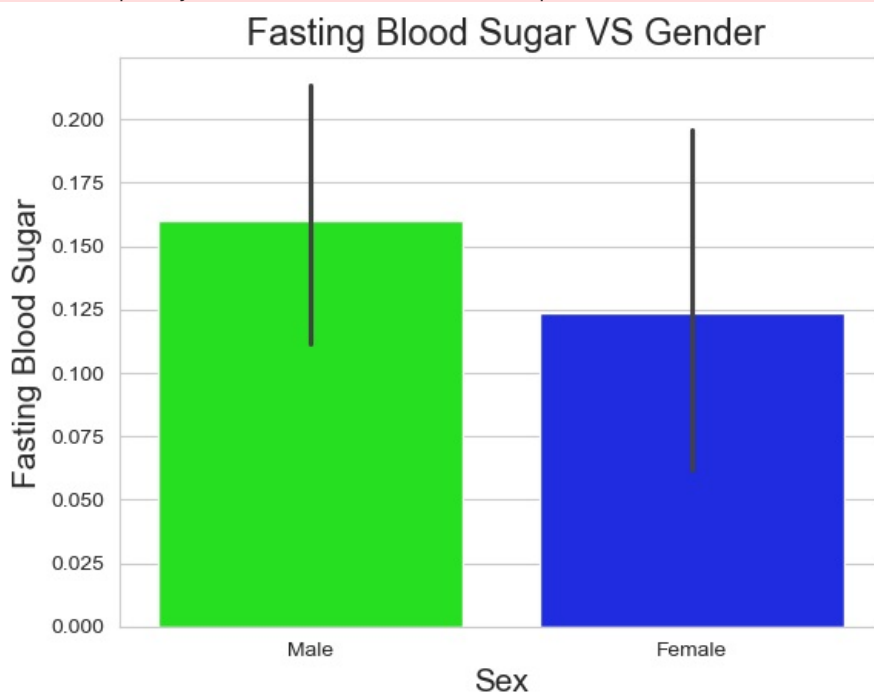
In [41]: *#Bar Plot Creation of Fasting Blood Sugar VS Gender using Matplotlib and Seaborn*

```
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()
```

C:\Users\Jashwanth\AppData\Local\Temp\ipykernel_18640\3171021298.py: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=False` for the same effect.

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



In [50]: **import** matplotlib.pyplot **as** plt
import seaborn **as** sns
import pandas **as** pd

```
# Assuming 'data' is your DataFrame
# Select only numeric columns
```

```

numeric_data = data.select_dtypes(include=['number'])

# Compute the correlation matrix
corr_matrix = numeric_data.corr()

# Plot the heatmap
plt.figure(figsize=(16,9))
sns.heatmap(corr_matrix, annot=True, linewidth=3, cmap='coolwarm')
plt.show()

```



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js