```
import pandas as pd import
seaborn as sns import
matplotlib.pyplot as plt
```

Step 1: Extract Data

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

```
heart_disease_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 row s × 14 columns

```
apping = {0: 'female', 1: 'male'}
```

```
_diseas e_data['sex'] = heart_disease_data['sex'].map(sex_mapping)
```

```
_diseas e_data
```

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

1025 rows × 14 columns

```
sex_m
```

```
heart
```

```
heart
```

```
Start coding or generate with AI.
```

Calculate overall heart disease rate

```
# Calculate overall heart disease rate overall_heart_disease_rate
= heart_disease_data['target'].mean()
```

```
overall_heart_disease_rate
```

```
0.5131707317073171
```

Group data by gender and calculate heart disease rates for each gender

```
heart_disease_by_gender = heart_disease_data.groupby('sex')['target'].mean()
```

```
heart_disease_by_gender
```

```
sex          female
0.724359       male
0.420757
Name: target, dtype: float64
```

De ne the bins for age groups

```
bins = [0, 30, 40, 50, 60, 70, 80]
```

De ne labels for the age groups

```
labels = ['0-29', '30-39', '40-49', '50-59', '60-69', '70-79']
```

Create a new column 'age_group' in the DataFrame

```
heart_disease_data['age_group'] = pd.cut(heart_disease_data['age'], bins=bins, labels=labels, right=False)
```

```
print(heart_disease_data[['age', 'age_group']].head())
```

```
   age age_group
0   52     50-59
1   53     50-59
2   70     70-79
3   61     60-69
4   62     60-69
```

```
heart_disease_data
```

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

1025 rows × 15 columns

```
age_group_counts = heart_disease_data['age_group'].value_counts()
```

```
age_group_counts
```

```
age_group 50-59    422
60-69      275
40-49      237
30-39       53
70-79       34
0-29         4
Name: count, dtype: int64
```

```
df = pd.DataFrame(heart_disease_data) df
```

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

1025 rows × 15 columns

```
counts = df.isna().sum()
```

```
counts
```

```
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
age_group    0
dtype: int64
```

```
num_rows = df.shape[0] num_rows
```

```
1025
```

```
num_columns = df.shape[1] num_columns
```

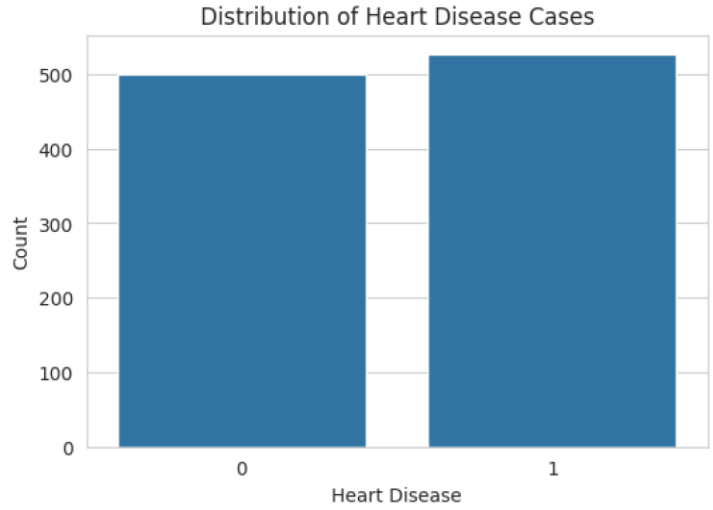
```
15
```

```
Start coding or generate with AI.
```

```
counts = heart_disease_data['target'].value_counts() print("Counts:",
counts)
```

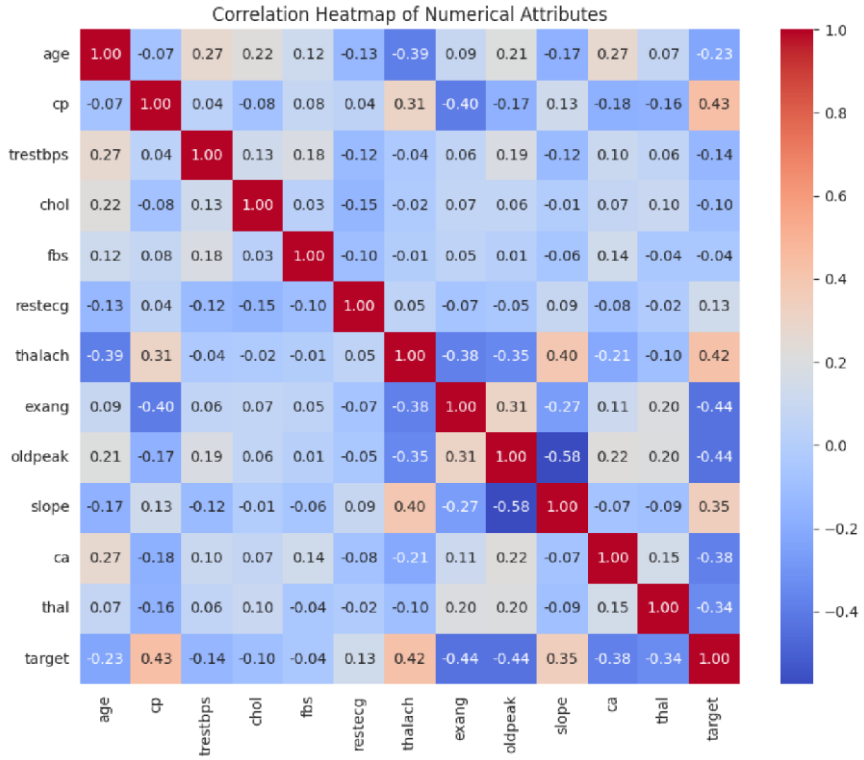
```
Counts: target
1    526
0    499
Name: count, dtype: int64
```

```
plt.figure(figsize=(6, 4)) sns.countplot(x='target',
data=heart_disease_data) plt.title('Distribution of
Heart Disease Cases') plt.xlabel('Heart Disease')
plt.ylabel('Count') plt.show()
```

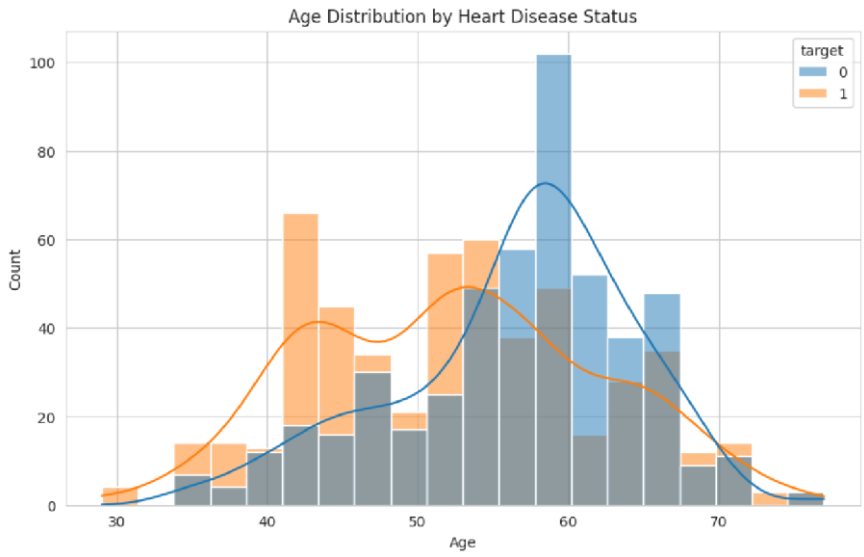



```
numeric_columns = heart_disease_data.select_dtypes(include=['number'])
```

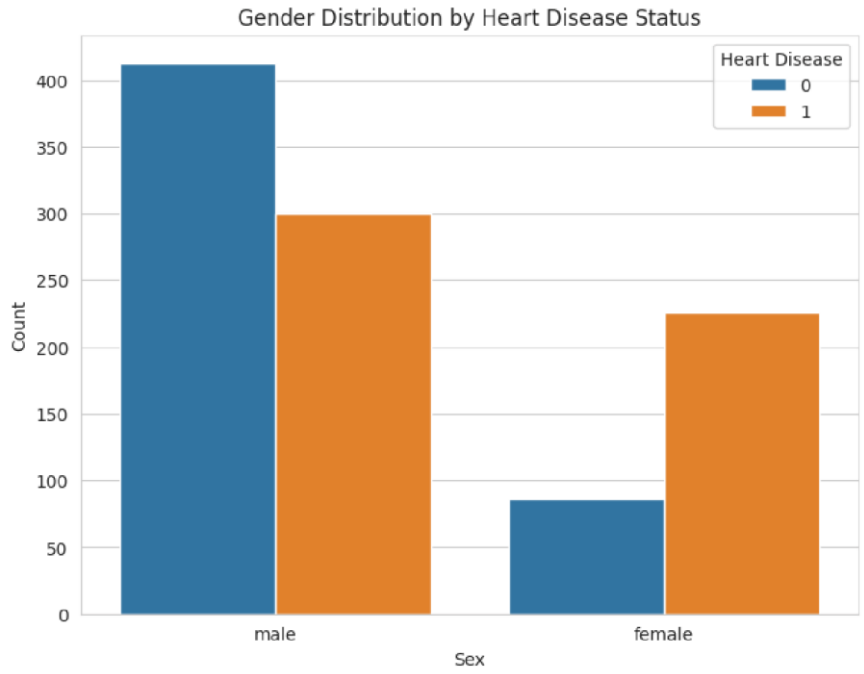
```
plt.figure(figsize=(10, 8)) sns.heatmap(numeric_columns.corr(), annot=True,
cmap='coolwarm', fmt='.2f') plt.title('Correlation Heatmap of Numerical
Attributes') plt.show()
```




```
plt.figure(figsize=(10, 6)) sns.histplot(data=heart_disease_data, x='age',
hue='target', kde=True, bins=20) plt.title('Age Distribution by Heart Disease
Status') plt.xlabel('Age') plt.ylabel('Count') plt.show()
```




```
plt.figure(figsize=(8, 6)) sns.countplot(x='sex', hue='target',
data=heart_disease_data) plt.title('Gender Distribution by
Heart Disease Status') plt.xlabel('Sex') plt.ylabel('Count')
plt.legend(title='Heart Disease') plt.show()
```

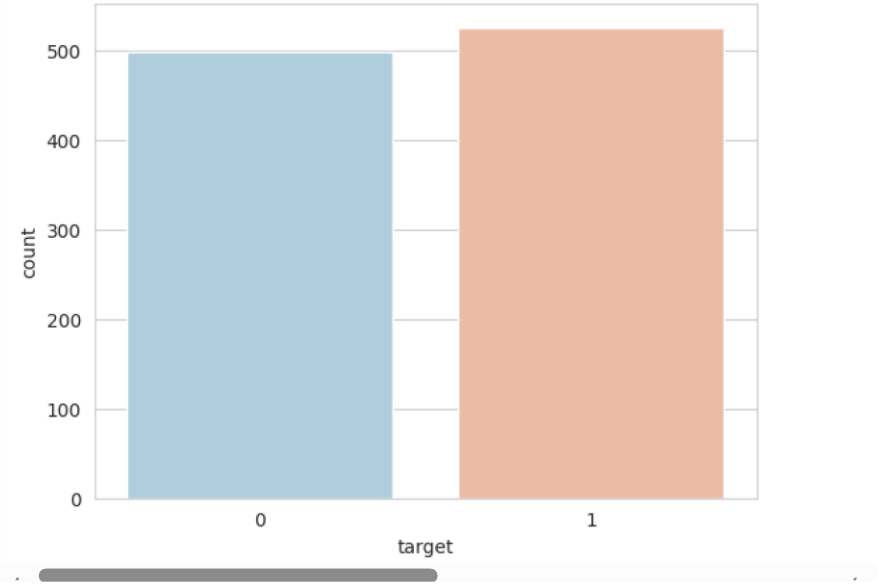



```
sns.pairplot(heart_disease_data, hue='target', diag_kind='hist') plt.show()
```

```
heart_disease_data['target'].value_counts()
```

```
target 1
526
0    499
Name: count, dtype: int64
```

```
sns.set_style('whitegrid') sns.countplot(x='target',data=heart_disease_data,palette='RdBu_r')
<ipython-input-75-5e8111ca5ebb>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.

sns.countplot(x='target',data=heart_disease_data,palette='RdBu_r')

<Axes: xlabel='target', ylabel='count'>
```



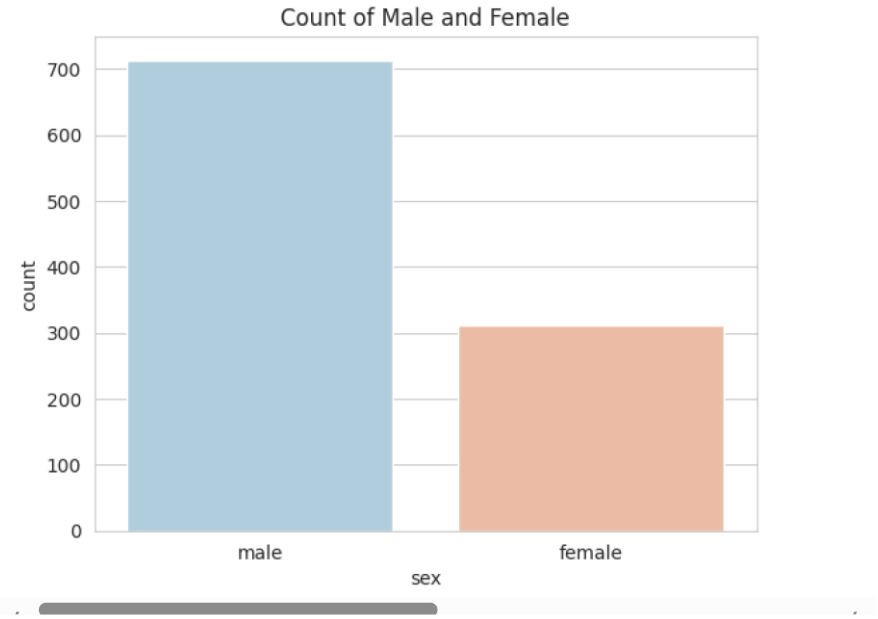
Finds the Counts of males and Females

```
heart_disease_data['sex'].value_counts()
```

```
sex       male
713     female
312
Name: count, dtype: int64
```

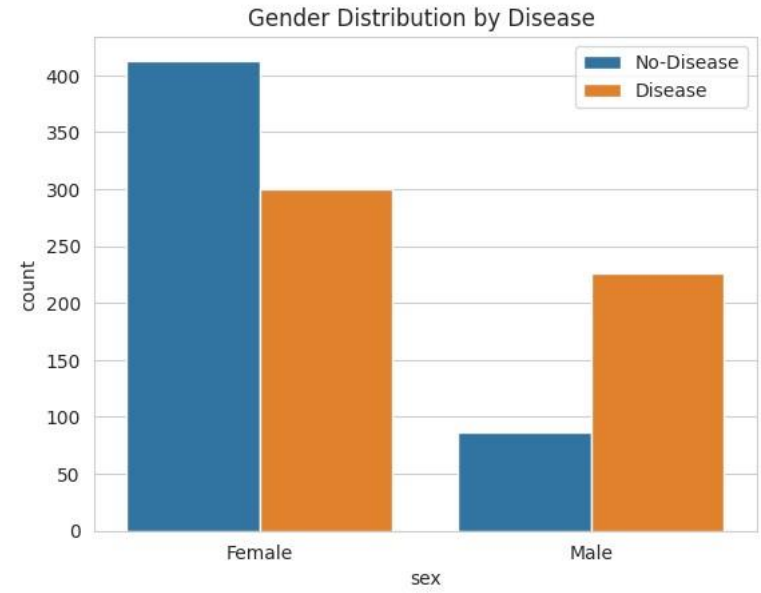
```
sns.set_style('whitegrid')
sns.countplot(x='sex',data=heart_disease_data,palette='RdBu_r') plt.title('Count
of Male and Female')

<ipython-input-77-b9ccb3f0da73>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.
sns.countplot(x='sex',data=heart_disease_data,palette='RdBu_r')
```

```
Text(0.5, 1.0, 'Count of Male and Female')
```



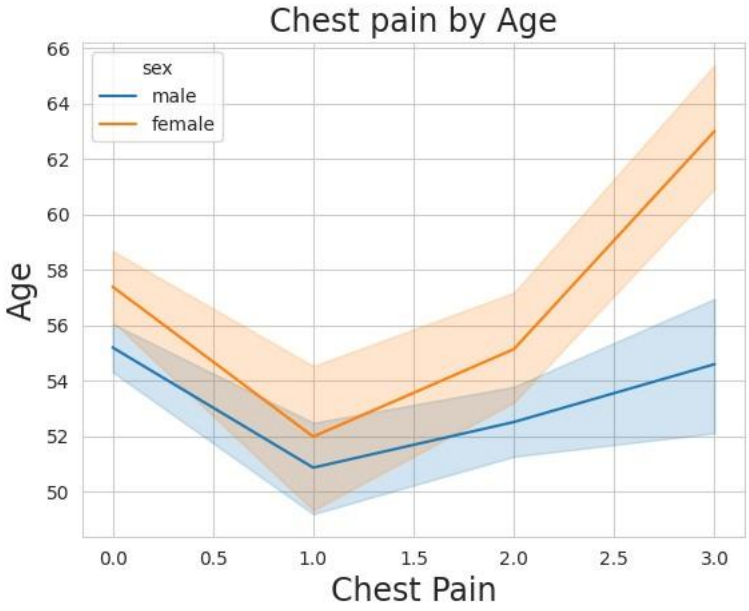
Gender wise

```
sns.countplot(data=heart_disease_data,x='sex', hue='target')
plt.xticks([1,0],['Male','Female']) plt.legend(labels=['No-
Disease','Disease']) plt.title('Gender Distribution by Disease
') plt.show()
```



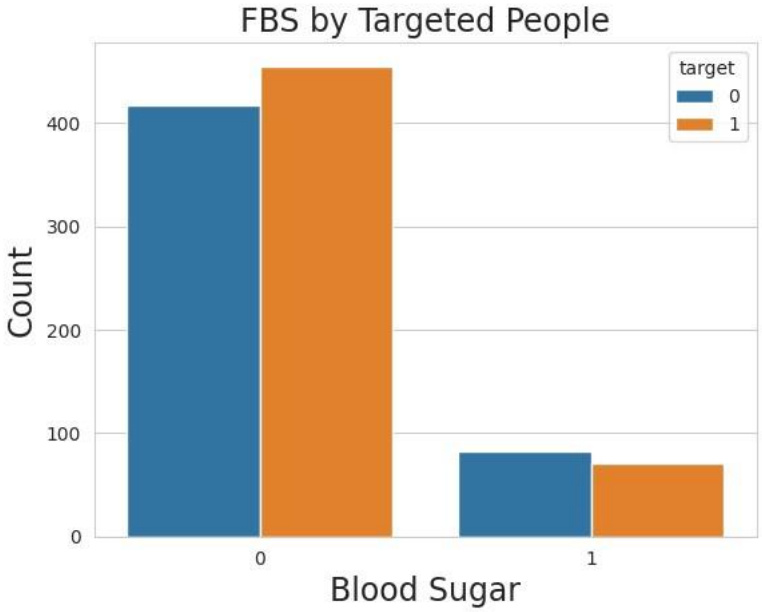

```
sns.lineplot(x='cp',data=heart_disease_data,y='age', hue='sex')
plt.xlabel('Chest Pain',fontsize=17)
plt.ylabel('Age',fontsize=17) plt.title('Chest pain by
Age',fontsize=17)
```

```
Text(0.5, 1.0, 'Chest pain by Age')
```



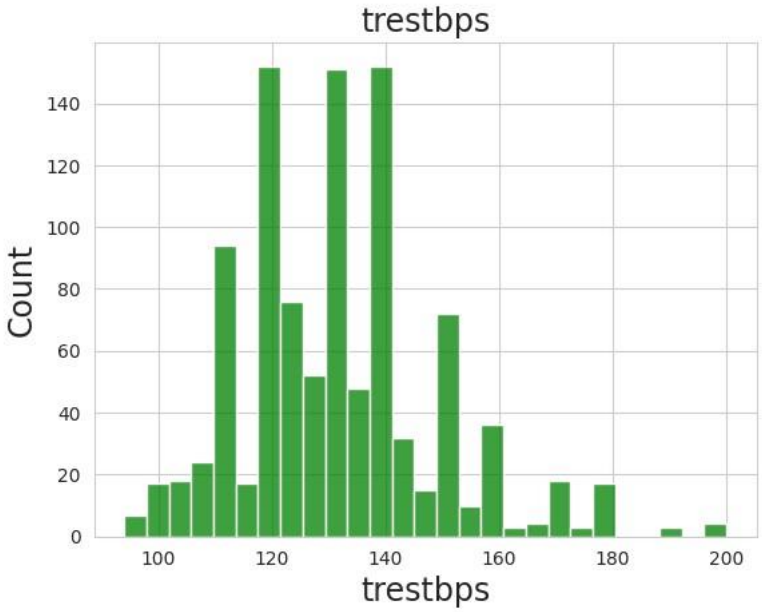

```
sns.countplot(data=heart_disease_data,x='fbs', hue='target')
plt.xlabel('Blood Sugar',fontsize=17)
plt.ylabel('Count',fontsize=17) plt.title('FBS by Targeted
People',fontsize=17)
```

```
Text(0.5, 1.0, 'FBS by Targeted People')
```




```
sns.histplot(x='trestbps',data=heart_disease_data, color='green')
plt.show plt.xlabel('trestbps',fontsize=17)
plt.ylabel('Count',fontsize=17) plt.title('trestbps',fontsize=17)
```

```
Text(0.5, 1.0, 'trestbps')
```




```
sns.histplot(x='chol',data=heart_disease_data,color='red')
plt.xlabel('Cholesterol',fontsize=18)
plt.ylabel('count',fontsize=18)
plt.title('Cholesterol',fontsize=18) plt.show
```