Heart Disease Diagnostic Analysis

Details of the Dataset :

Attribute Information:

age sex

chest pain type (4 values)

resting blood pressure

serum cholestoral in mg/dl

fasting blood sugar > 120 mg/dl

resting electrocardiographic results (values 0,1,2)

maximum heart rate achieved

exercise induced angina

oldpeak = ST depression induced by exercise relative to rest the slope of the peak exercise ST segment

number of major vessels (0-3) colored by flourosopy

thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

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

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  No file chosen        Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to
enable
```

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

Data Exploration

```
df.head()
```

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

```
df.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000

```
df.columns
```

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
missing_values = df.isnull().sum()
print("Missing Values\n", missing_values)
```

```
Missing Values
 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
```

```
duplicate_values = df.duplicated().sum()
print('Number of duplicates', duplicate_values)
```

```
Number of duplicates 723
```

```
df = df.drop_duplicates()
```

```
duplicate_values = df.duplicated().sum()
print('Number of duplicates', duplicate_values)
```

```
Number of duplicates 0
```

Data Analysis

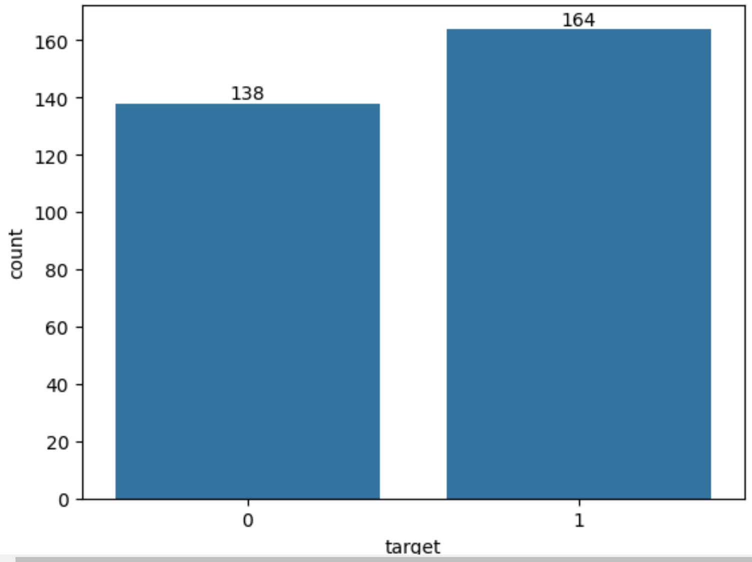
Number of people with heart disease

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

```
        count
target
     1    164
     0    138

dtype: int64
```

```
hd = sns.countplot(x=df['target'])
for i in hd.containers:
    hd.bar_label(i,)
    plt.show()
```



From here, we can see that more than half of the population have Heart disease.

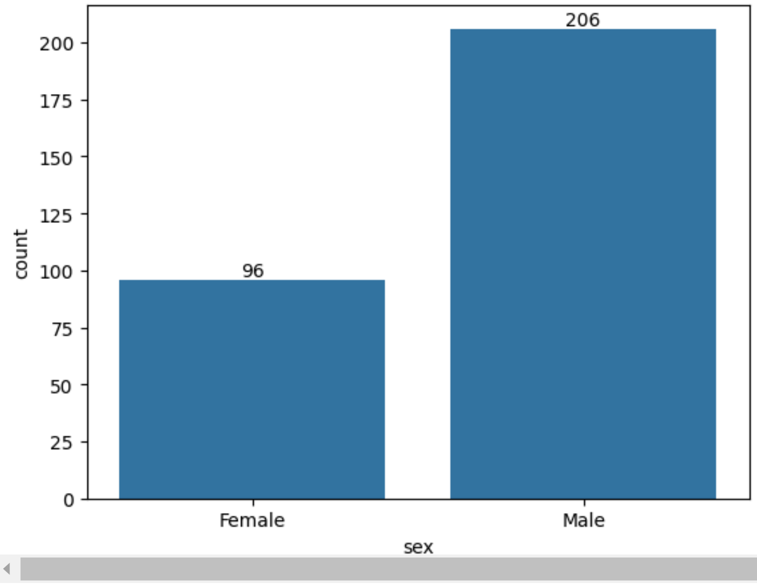
Number of Male and Female

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

```
     count
sex
  1    206
  0     96

dtype: int64
```

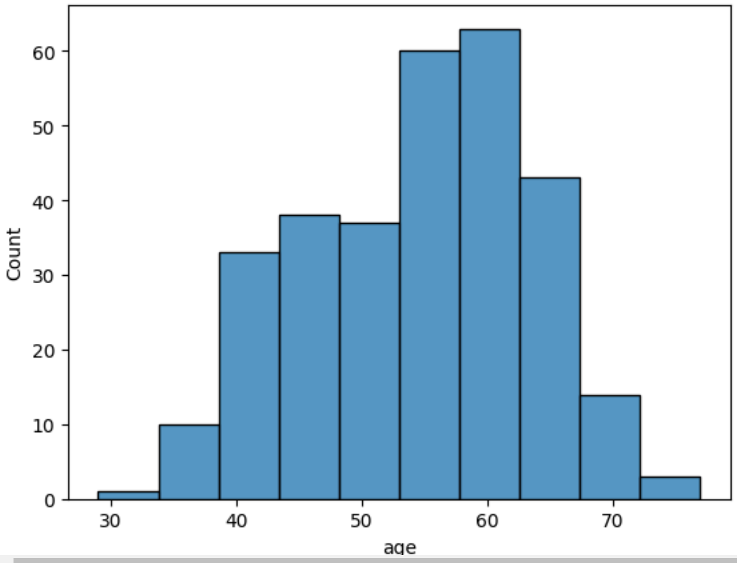
```
gd = sns.countplot(x=df['sex'])
for i in gd.containers:
    gd.bar_label(i,)
plt.xticks([0,1],['Female','Male'])
plt.show()
```



We can conclude, Males suffer more from Heart Disease than Females.

Age Distribution

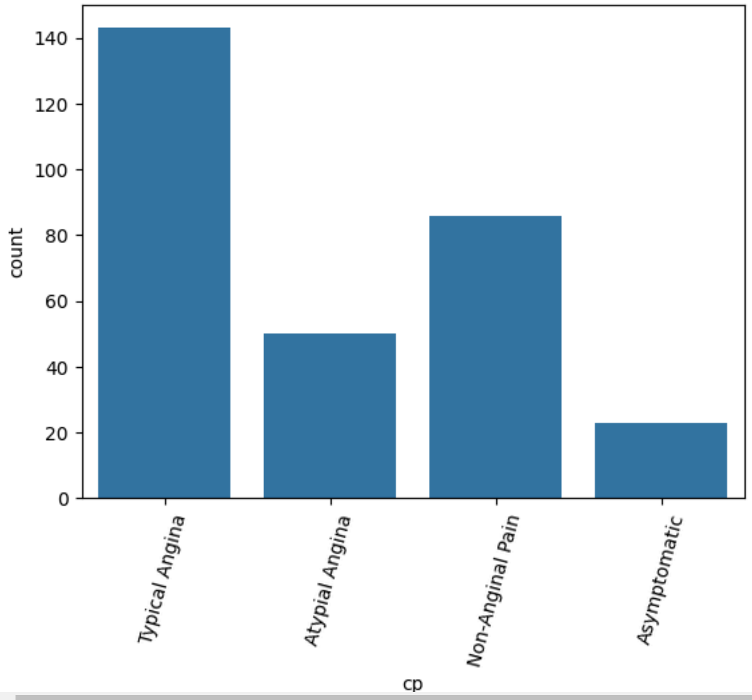
```
sns.histplot(df['age'],bins=10)
plt.show()
```



From the above graph, we can observe that maximum number of people lie between the age of 50-60.

Chest Pain Types

```
sns.countplot(x=df['cp'])
plt.xticks([0,1,2,3],['Typical Angina','Atypial Angina','Non-Anginal Pain','Asymptomatic'])
plt.xticks(rotation=75)
plt.show()
```

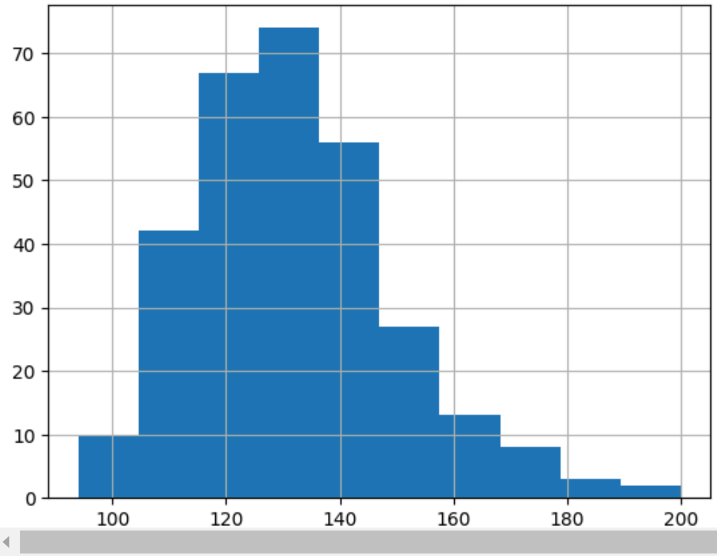


From the above plot, we can clearly observe that Chest Pain Type(0), i.e. Typical Angina is more common.

Resting Blood Pressure Distribution

```
df['trestbps'].hist()
```

```
<Axes: >
```



Resting Blood Pressure Vs. Sex

```
g= sns.FacetGrid(df,hue='sex',aspect=4)
g.map(sns.kdeplot,'trestbps',shade=True)
plt.legend(labels=['Male','Female'])
plt.show()
```

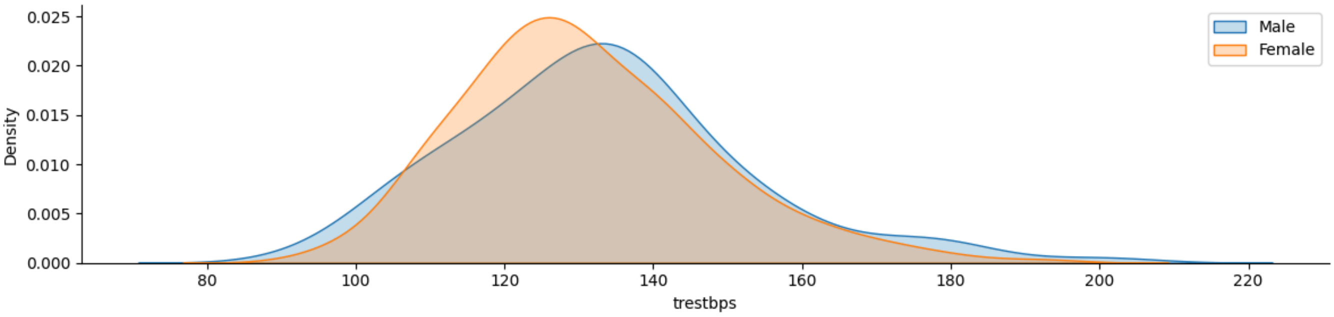
```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
```

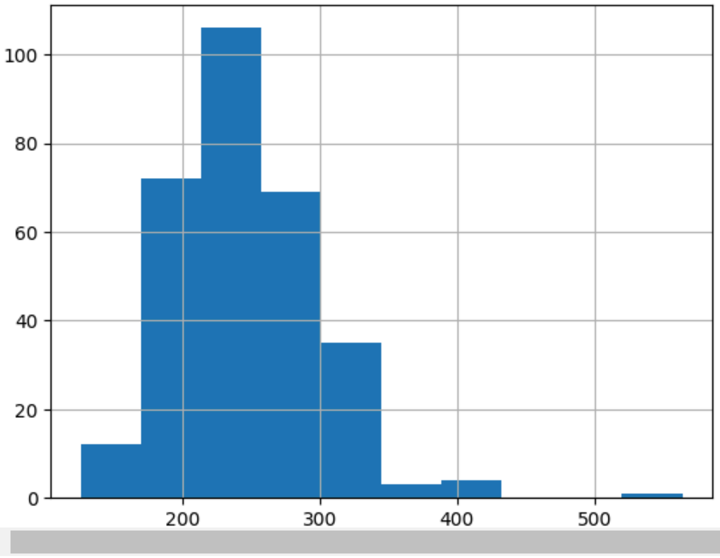


From the above plot, we can observe that the Women have lower resting blood pressure as compared to men. For Women, around 120 and for Men, little less than 140.

Distribution of Serum Cholestrol

```
df['chol'].hist()
```

```
<Axes: >
```

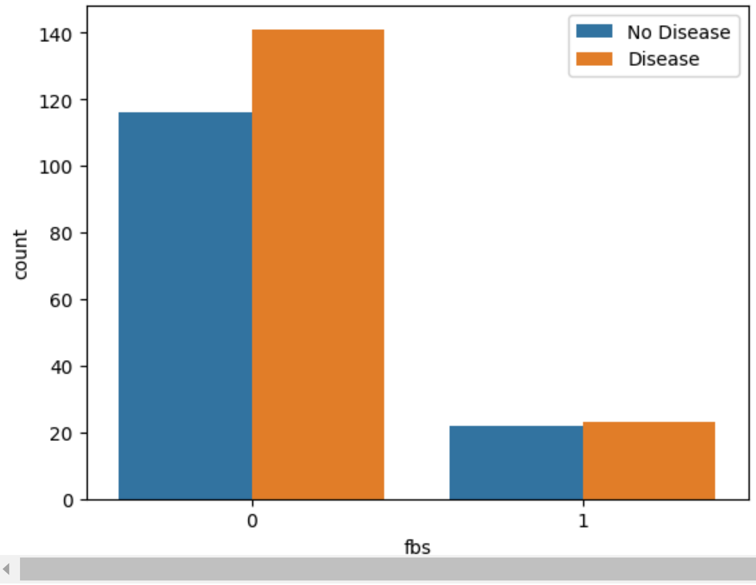


Chest Pain distribution as per target variable

```
sns.countplot(x='cp',hue='target',data=df)
plt.legend(labels=['No Disease','Disease'])
plt.show()
```

Fasting Blood Pressure distribution as per target variable

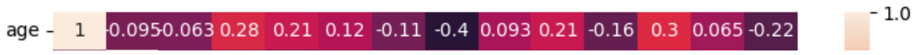
```
sns.countplot(x='fbs',hue ='target',data=df)
plt.legend(labels=['No Disease','Disease'])
plt.show()
```



Correlation

```
plt.figure(figsize=(9,5))
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```


age