```
#Import neccesary Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv("D:\\Data_Science_Intern\\Heart Disease data\\Heart 
Disease data.csv")

#details of rows and column
df.head()
```

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

   ca  thal  target  
0   2     3       0  
1   0     3       0  
2   0     3       0  
3   1     3       0  
4   3     2       0  
```

```
df.shape
```

```
(1025, 14)
```

```
#Description of data
df.describe()
```

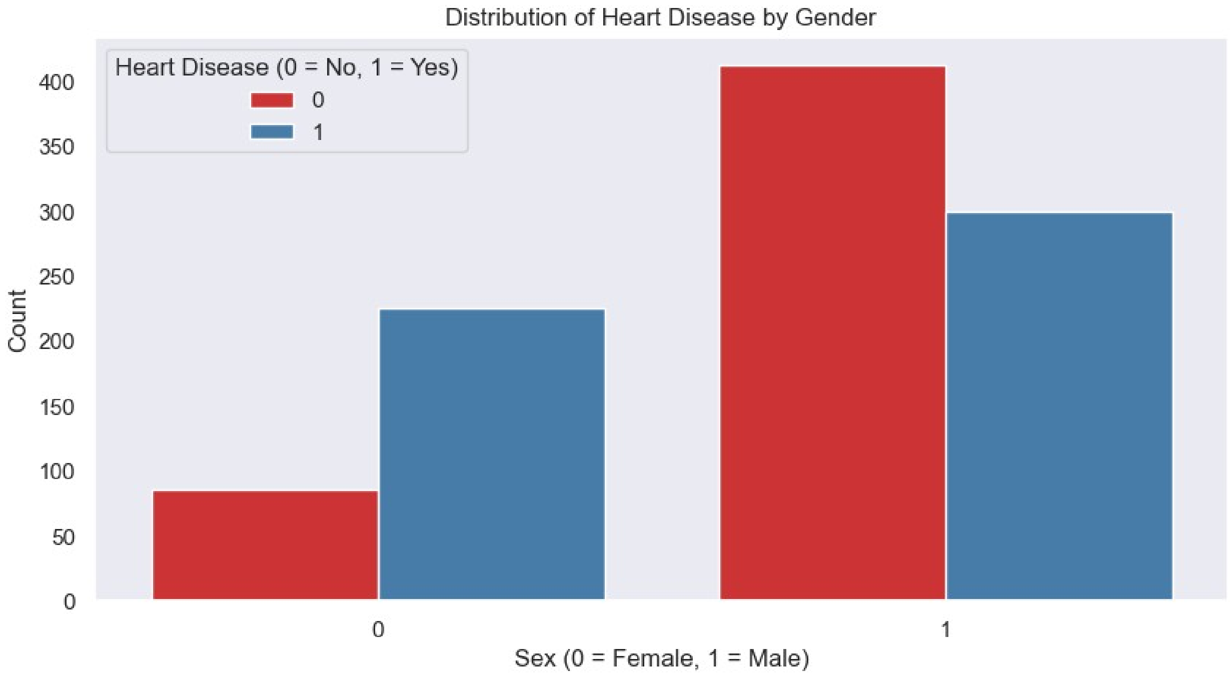
```
                age          sex           cp     trestbps         chol  
\
count  1025.000000  1025.000000  1025.000000  1025.000000  1025.00000

mean     54.434146     0.695610     0.942439   131.611707   246.00000

std       9.072290     0.460373     1.029641    17.516718    51.59251

min      29.000000     0.000000     0.000000    94.000000   126.00000

25%      48.000000     0.000000     0.000000   120.000000   211.00000

50%      56.000000     1.000000     1.000000   130.000000   240.00000
```

```
75%      61.000000     1.000000     2.000000   140.000000   275.00000

max      77.000000     1.000000     3.000000   200.000000   564.00000


                fbs      restecg      thalach        exang      oldpeak
\
count   1025.000000  1025.000000  1025.000000  1025.000000  1025.000000

mean       0.149268     0.529756   149.114146     0.336585     1.071512

std        0.356527     0.527878    23.005724     0.472772     1.175053

min        0.000000     0.000000    71.000000     0.000000     0.000000

25%        0.000000     0.000000   132.000000     0.000000     0.000000

50%        0.000000     1.000000   152.000000     0.000000     0.800000

75%        0.000000     1.000000   166.000000     1.000000     1.800000

max        1.000000     2.000000   202.000000     1.000000     6.200000


              slope           ca         thal       target
count   1025.000000  1025.000000  1025.000000  1025.000000
mean       1.385366     0.754146     2.323902     0.513171
std        0.617755     1.030798     0.620660     0.500070
min        0.000000     0.000000     0.000000     0.000000
25%        1.000000     0.000000     2.000000     0.000000
50%        1.000000     0.000000     2.000000     1.000000
75%        2.000000     1.000000     3.000000     1.000000
max        2.000000     4.000000     3.000000     1.000000
```

```
# Check for missing values
missing_values = df.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

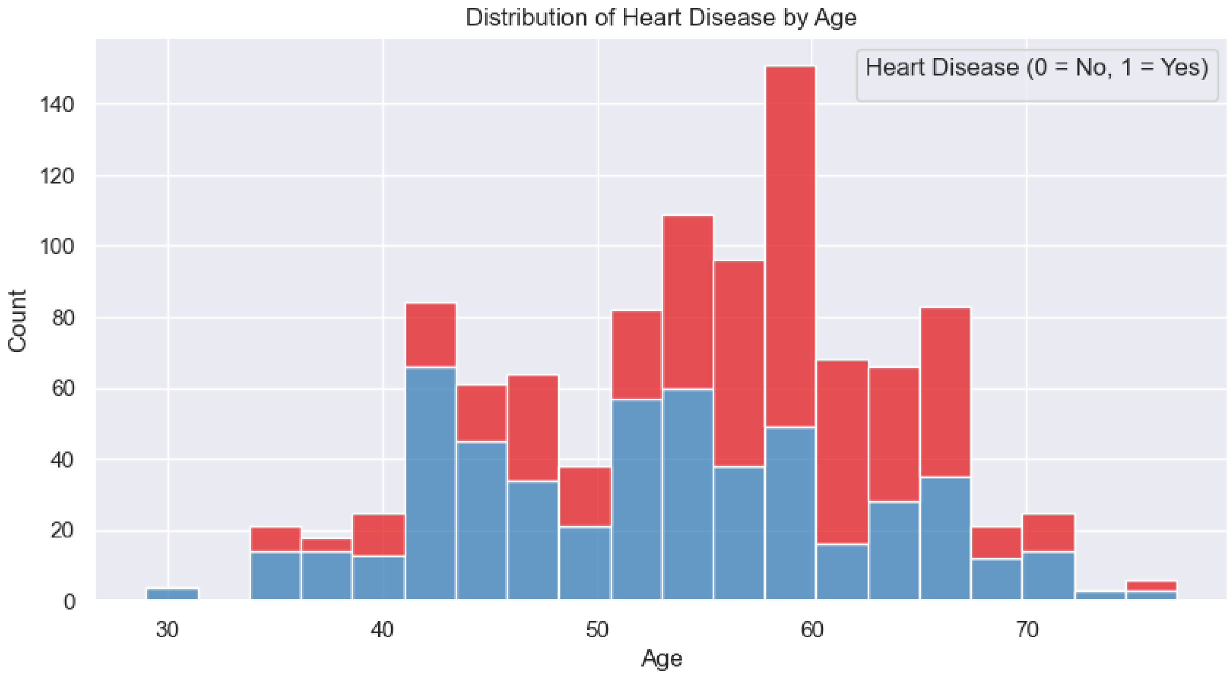
```
Missing values in each column:
 age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
```

```
ca          0
thal        0
target      0
dtype: int64

# Check data types
data_types = df.dtypes
print("\nData types of each column:\n", data_types)


Data types of each column:
 age              int64
sex              int64
cp               int64
trestbps         int64
chol             int64
fbs              int64
restecg          int64
thalach          int64
exang            int64
oldpeak        float64
slope            int64
ca               int64
thal             int64
target           int64
dtype: object

# Set plot style
sns.set(style="dark")

#Convert target data into string
df['target'] = df['target'].astype(str)

# Distribution of heart disease by gender using countplot
plt.figure(figsize=(10, 5))
sns.countplot(x='sex', hue='target', data=df, palette='Set1')
plt.title('Distribution of Heart Disease by Gender')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.ylabel('Count')
plt.legend(title='Heart Disease (0 = No, 1 = Yes)')
plt.show()
```

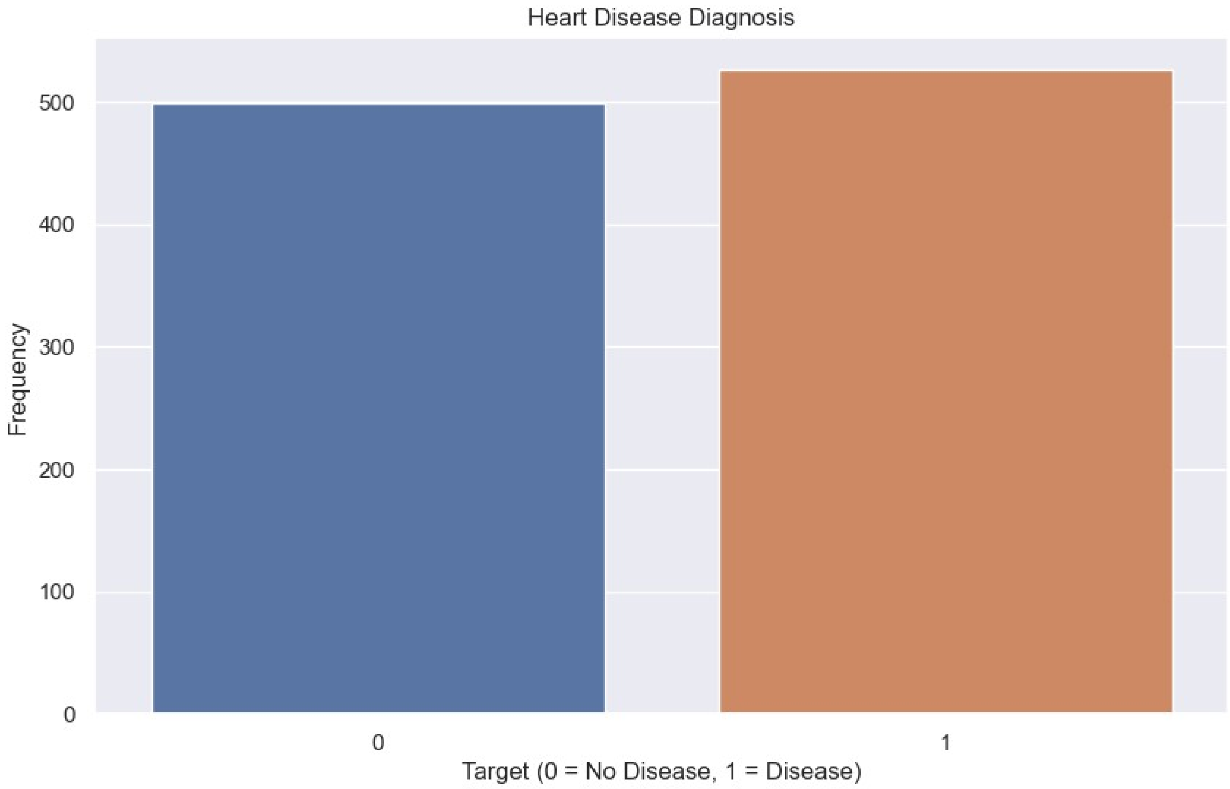


```
sns.set(style="darkgrid")

#Distribution of heart disease by age using histogram ploting
plt.figure(figsize=(10, 5))
sns.histplot(data=df, x='age', hue='target', multiple='stack',
palette='Set1', bins=20)
plt.title('Distribution of Heart Disease by Age')
plt.xlabel('Age')
plt.ylabel('Count')
plt.legend(title='Heart Disease (0 = No, 1 = Yes)')
plt.show()
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.
```

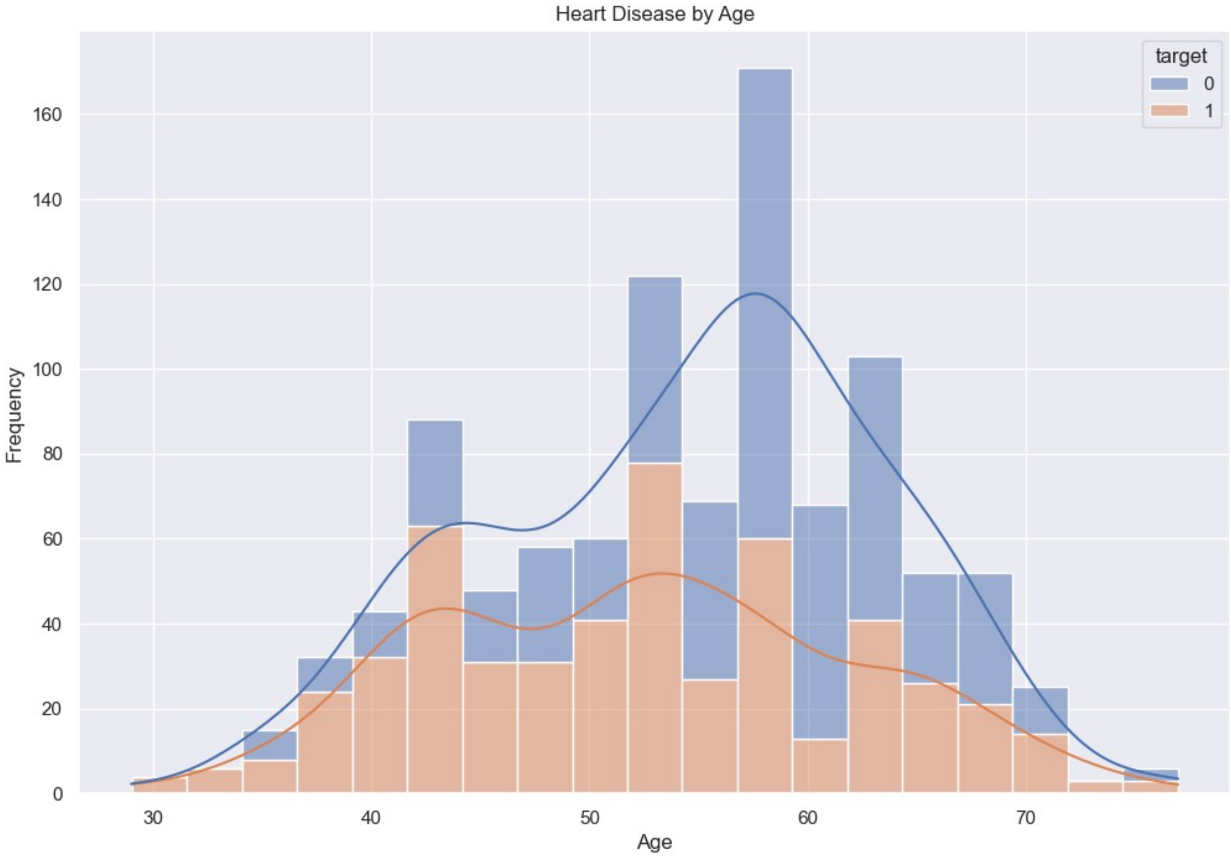


```
# Plot the distribution of target variable using countplot
plt.figure(figsize=(10, 6))
sns.countplot(x='target', data=df)
plt.title('Heart Disease Diagnosis')
plt.xlabel('Target (0 = No Disease, 1 = Disease)')
plt.ylabel('Frequency')
plt.show()
```

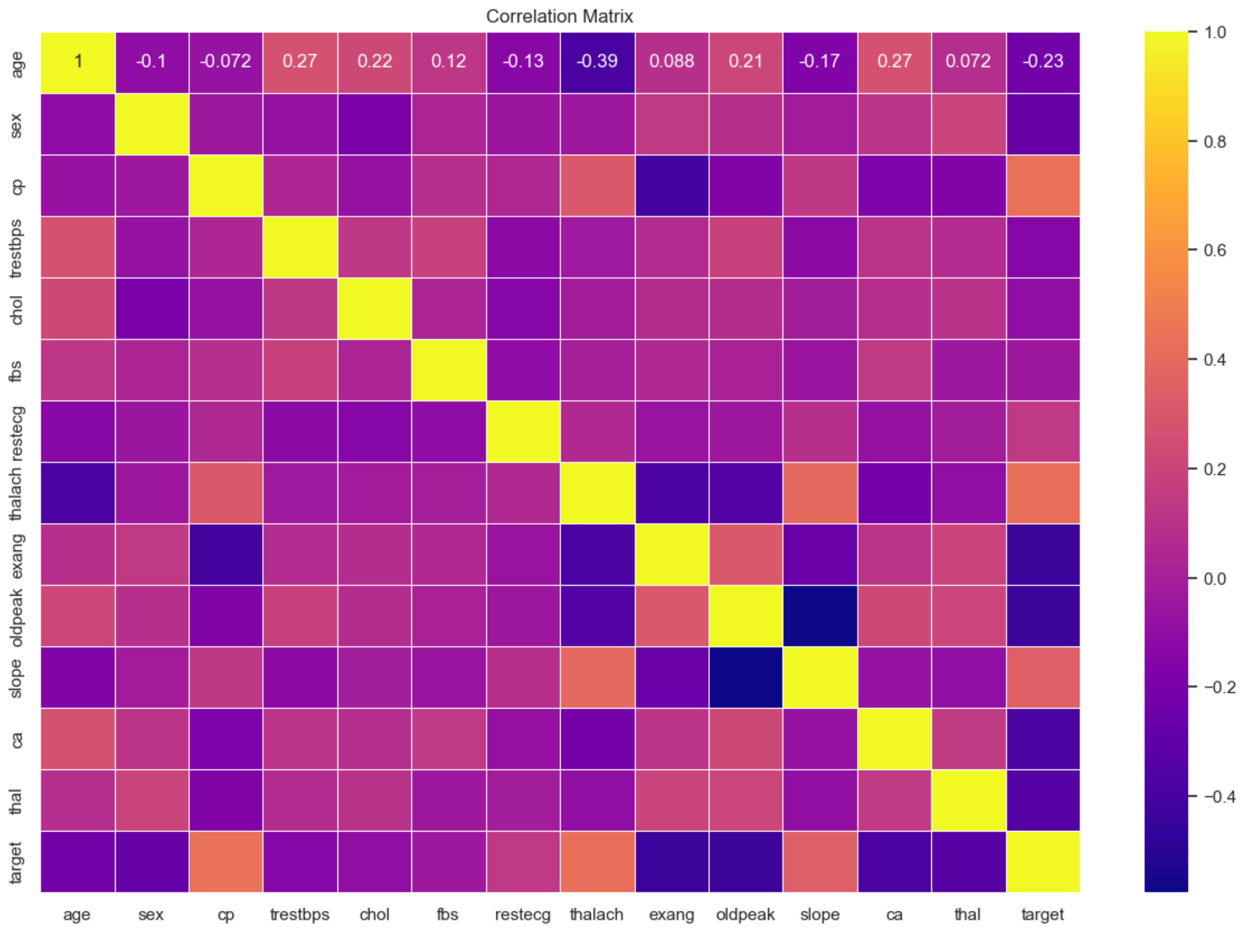


```
# Plot the relationship between age and heart disease using histogram
ploting
plt.figure(figsize=(12, 8))
sns.histplot(data=df, x='age', hue='target', multiple='stack',
kde=True)
plt.title('Heart Disease by Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
#Correlation matrix using heatmap
plt.figure(figsize=(15, 10))
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='plasma',
linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```

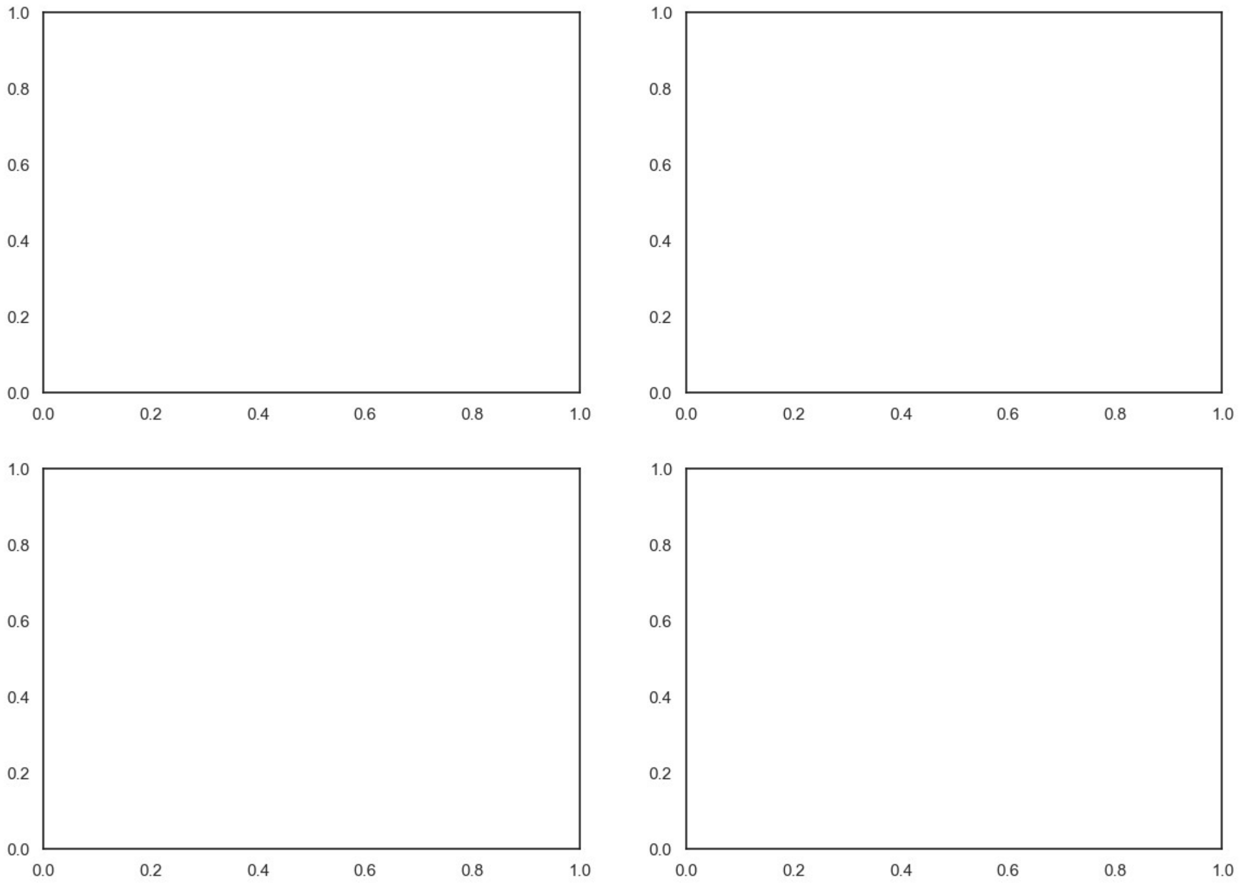


```
# Calculate the correlation matrix
correlation_matrix = df.corr()
sns.set(style="white")

print(correlation_matrix)
```

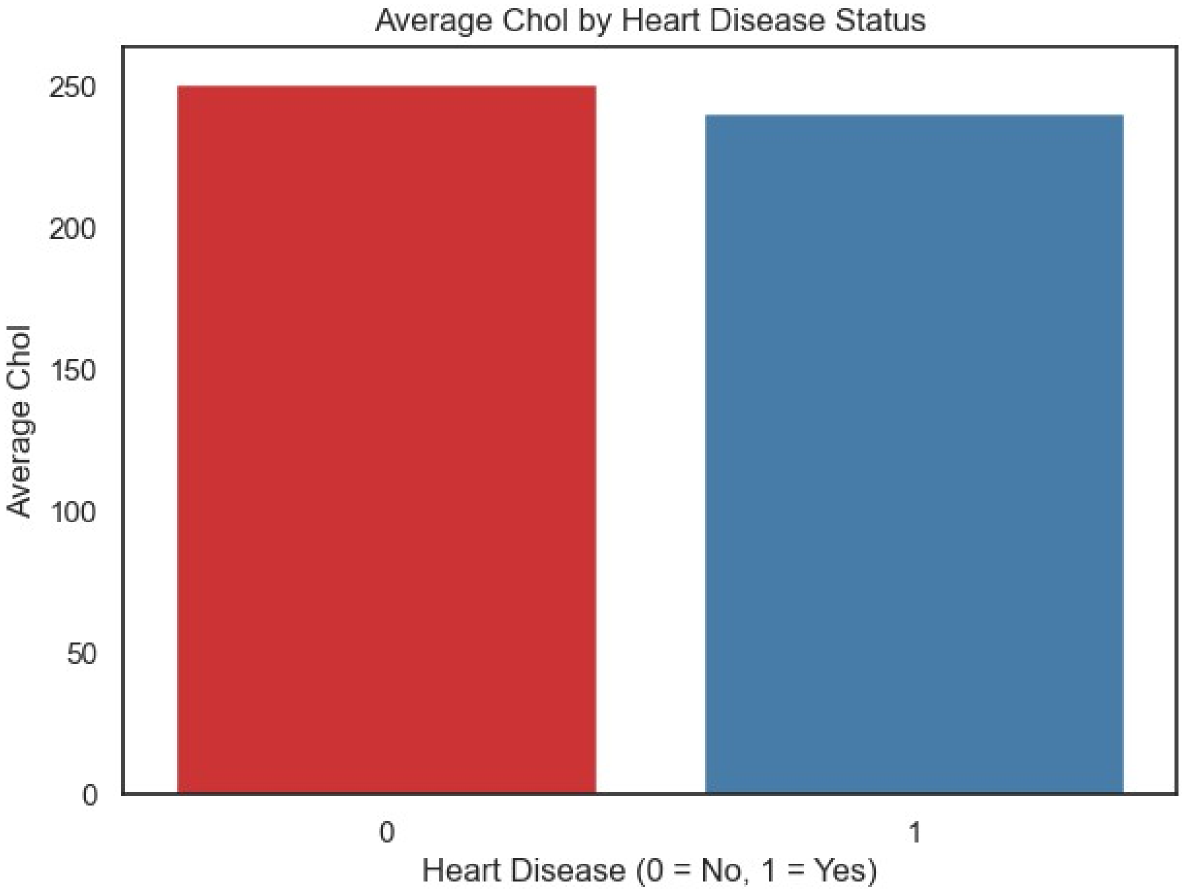
```
               age       sex        cp  trestbps      chol
fbs  \
age       1.000000 -0.103240 -0.071966  0.271121  0.219823  0.121243

sex      -0.103240  1.000000 -0.041119 -0.078974 -0.198258  0.027200

cp       -0.071966 -0.041119  1.000000  0.038177 -0.081641  0.079294

trestbps  0.271121 -0.078974  0.038177  1.000000  0.127977  0.181767

chol      0.219823 -0.198258 -0.081641  0.127977  1.000000  0.026917

fbs       0.121243  0.027200  0.079294  0.181767  0.026917  1.000000

restecg  -0.132696 -0.055117  0.043581 -0.123794 -0.147410 -0.104051
```

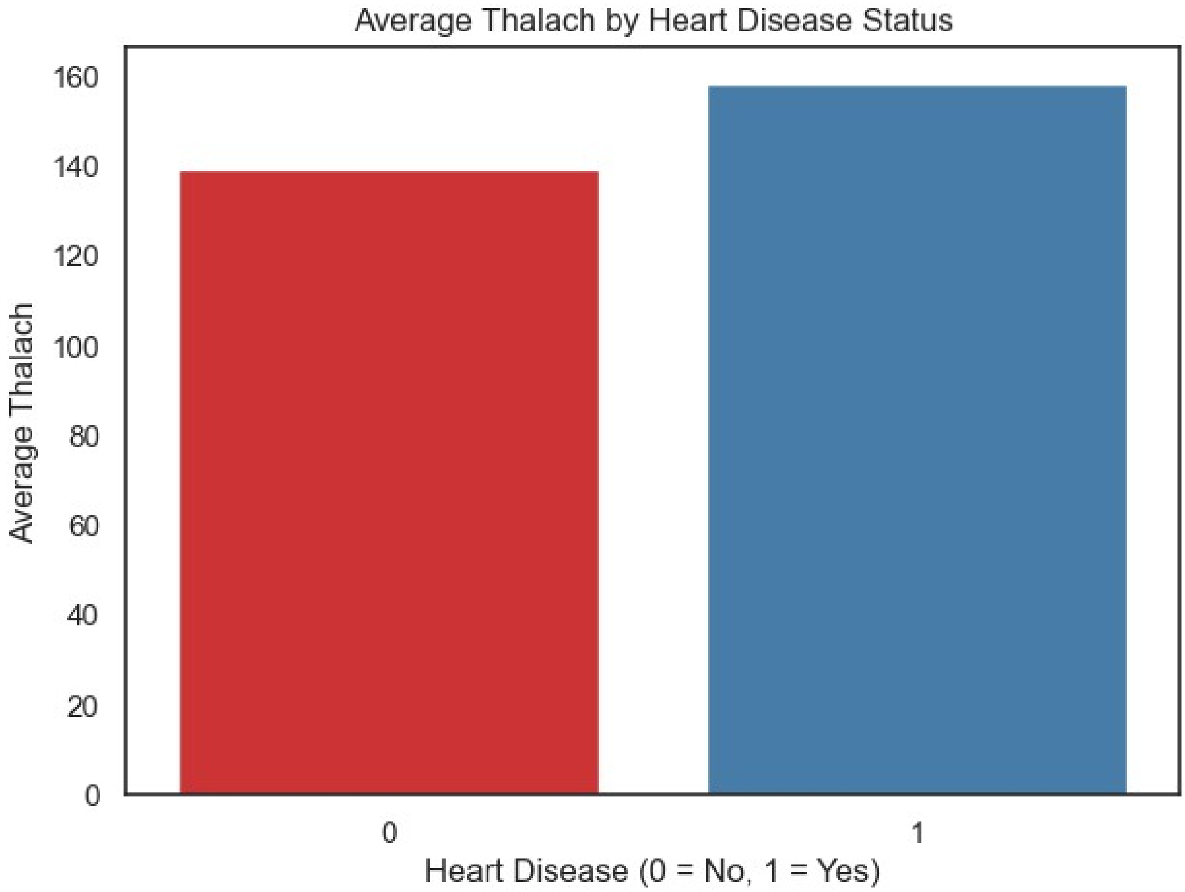
```
thalach   -0.390227 -0.049365  0.306839 -0.039264 -0.021772 -0.008866

exang      0.088163  0.139157 -0.401513  0.061197  0.067382  0.049261

oldpeak    0.208137  0.084687 -0.174733  0.187434  0.064880  0.010859

slope     -0.169105 -0.026666  0.131633 -0.120445 -0.014248 -0.061902

ca         0.271551  0.111729 -0.176206  0.104554  0.074259  0.137156

thal       0.072297  0.198424 -0.163341  0.059276  0.100244 -0.042177

target    -0.229324 -0.279501  0.434854 -0.138772 -0.099966 -0.041164


           restecg   thalach     exang   oldpeak     slope
ca  \
age       -0.132696 -0.390227  0.088163  0.208137 -0.169105  0.271551

sex       -0.055117 -0.049365  0.139157  0.084687 -0.026666  0.111729

cp         0.043581  0.306839 -0.401513 -0.174733  0.131633 -0.176206

trestbps  -0.123794 -0.039264  0.061197  0.187434 -0.120445  0.104554

chol      -0.147410 -0.021772  0.067382  0.064880 -0.014248  0.074259

fbs       -0.104051 -0.008866  0.049261  0.010859 -0.061902  0.137156

restecg    1.000000  0.048411 -0.065606 -0.050114  0.086086 -0.078072

thalach    0.048411  1.000000 -0.380281 -0.349796  0.395308 -0.207888

exang     -0.065606 -0.380281  1.000000  0.310844 -0.267335  0.107849

oldpeak   -0.050114 -0.349796  0.310844  1.000000 -0.575189  0.221816

slope      0.086086  0.395308 -0.267335 -0.575189  1.000000 -0.073440

ca        -0.078072 -0.207888  0.107849  0.221816 -0.073440  1.000000

thal      -0.020504 -0.098068  0.197201  0.202672 -0.094090  0.149014

target     0.134468  0.422895 -0.438029 -0.438441  0.345512 -0.382085


              thal    target
age       0.072297 -0.229324
sex       0.198424 -0.279501
cp       -0.163341  0.434854
trestbps  0.059276 -0.138772
```

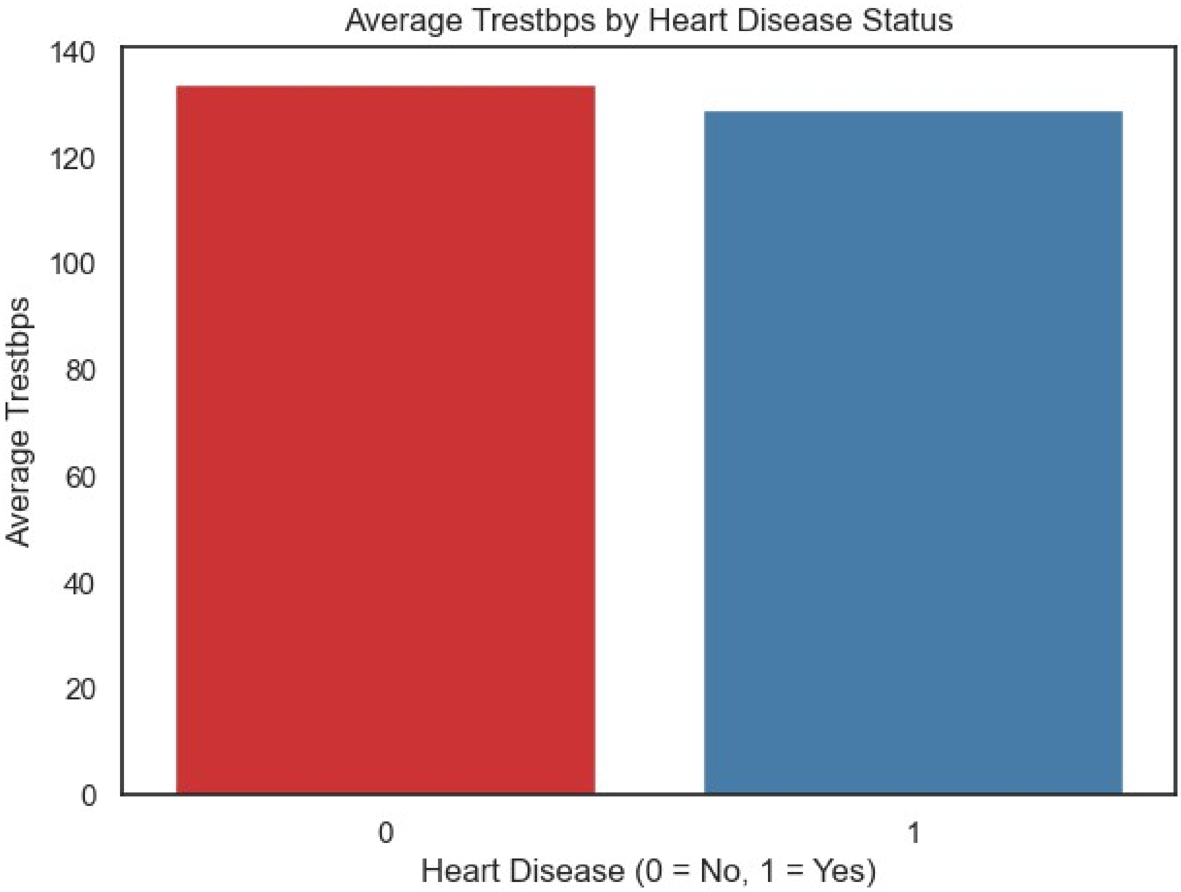
```
chol      0.100244 -0.099966
fbs      -0.042177 -0.041164
restecg  -0.020504  0.134468
thalach  -0.098068  0.422895
exang     0.197201 -0.438029
oldpeak   0.202672 -0.438441
slope    -0.094090  0.345512
ca        0.149014 -0.382085
thal      1.000000 -0.337838
target   -0.337838  1.000000

# Compute key metrics
average_metrics = df.groupby('target').mean()
# Metrics to display
metrics = ['chol', 'thalach', 'trestbps', 'oldpeak']

print(average_metrics)

              age       sex        cp    trestbps        chol
fbs  \
target

0       56.569138  0.827655  0.482966  134.106212  251.292585
0.164329
1       52.408745  0.570342  1.378327  129.245247  240.979087
0.134981

         restecg     thalach     exang   oldpeak     slope        ca
thal
target

0       0.456914  139.130261  0.549098  1.600200  1.166333  1.158317
2.539078
1       0.598859  158.585551  0.134981  0.569962  1.593156  0.370722
2.119772

#Average metrics by heart disease status
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
```

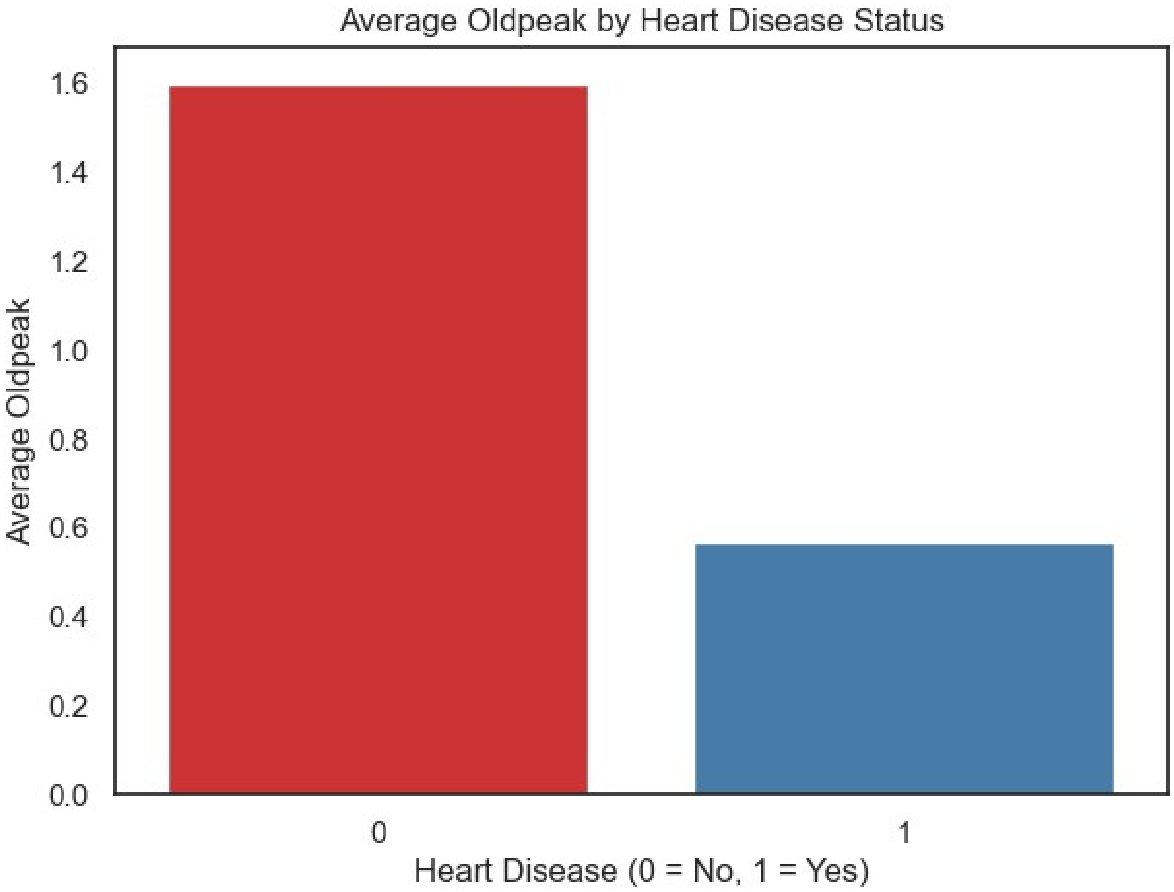


```
for metric in metrics:
    plt.figure(figsize=(7, 5))
    sns.barplot(x=average_metrics.index.astype(str),
y=average_metrics[metric], palette='Set1')
    plt.title(f'Average {metric.capitalize()} by Heart Disease
Status')
    plt.xlabel('Heart Disease (0 = No, 1 = Yes)')
    plt.ylabel(f'Average {metric.capitalize()}')
    plt.show()
```


Average Chol by Heart Disease Status
Average Chol
250
200
150
100
50
0
0
1
Heart Disease (0 = No, 1 = Yes)


Average Thalach by Heart Disease Status
Average Thalach
160
140
120
100
80
60
40
20
0
0
1
Heart Disease (0 = No, 1 = Yes)


Average Trestbps by Heart Disease Status
Average Trestbps
140
120
100
80
60
40
20
0
0
1
Heart Disease (0 = No, 1 = Yes)


Average Oldpeak by Heart Disease Status
Average Oldpeak
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
Heart Disease (0 = No, 1 = Yes)