```
#PROJECT -1 [HEART DISEASE ANALYSIS]
import pandas as pd
import numpy as np
data=pd.read_csv("Heart Disease data.csv")
data
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
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

1025 rows × 14 columns

```
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
```

```
data.notnull().sum()
```

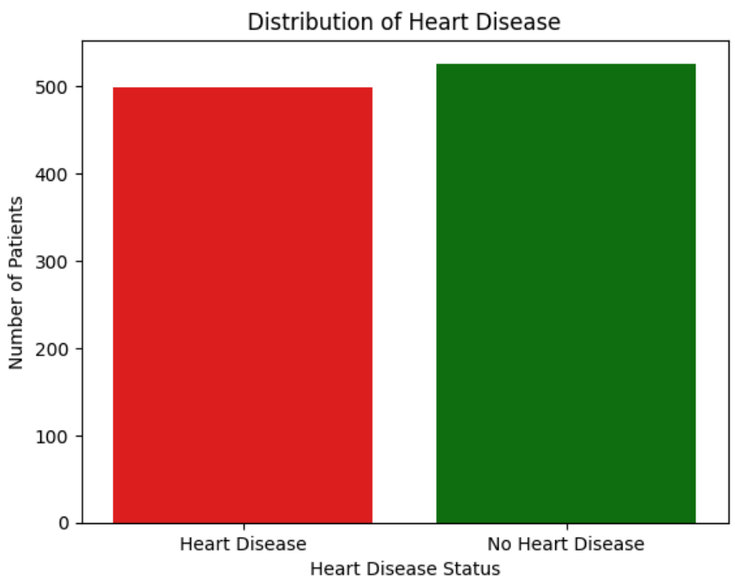
```
age         1025
sex         1025
cp          1025
trestbps    1025
chol        1025
fbs         1025
restecg     1025
thalach     1025
exang       1025
oldpeak     1025
slope       1025
ca          1025
thal        1025
target      1025
dtype: int64
```

```
print(data.head())
print(data.describe())
```

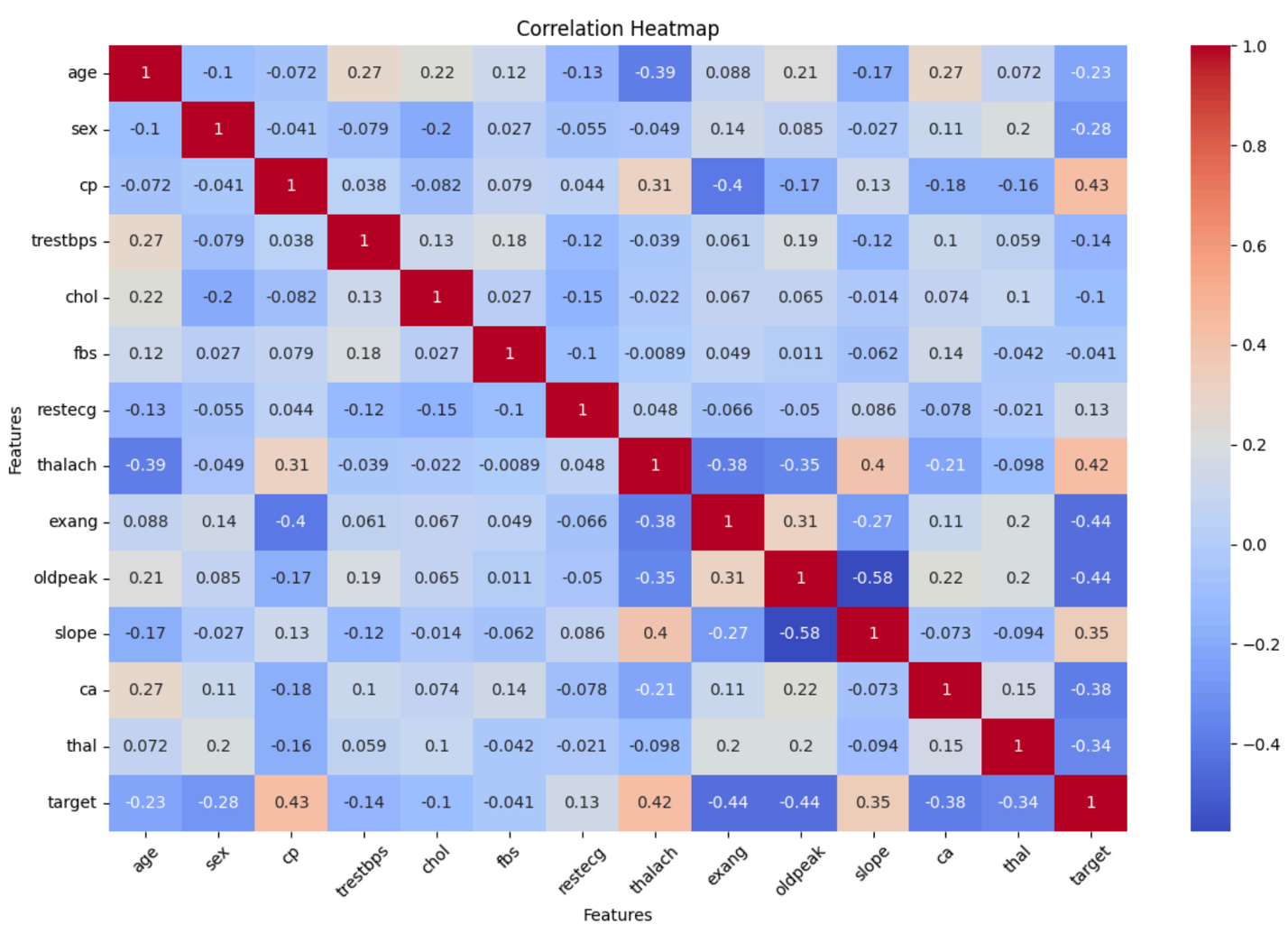
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   52    1   0       125   212    0        1      168      0      1.0      2
1   53    1   0       140   203    1        0      155      1      3.1      0
2   70    1   0       145   174    0        1      125      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
               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
```

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

# Analyze key metrics and relationships
# 1. Target variable distribution (presence of heart disease)
sns.countplot(x='target', data=data, hue_order=[1, 0],  # Specify order for labels
              palette=['red', 'green'])  # Set custom color palette
labels = ["Heart Disease", "No Heart Disease"]  # Define custom labels
plt.xlabel('Heart Disease Status')  # Update x-axis label
plt.ylabel('Number of Patients')  # Update y-axis label
plt.xticks(ticks=[0, 1], labels=labels)  # Set custom labels on x-axis ticks
plt.title('Distribution of Heart Disease')
plt.show()
```

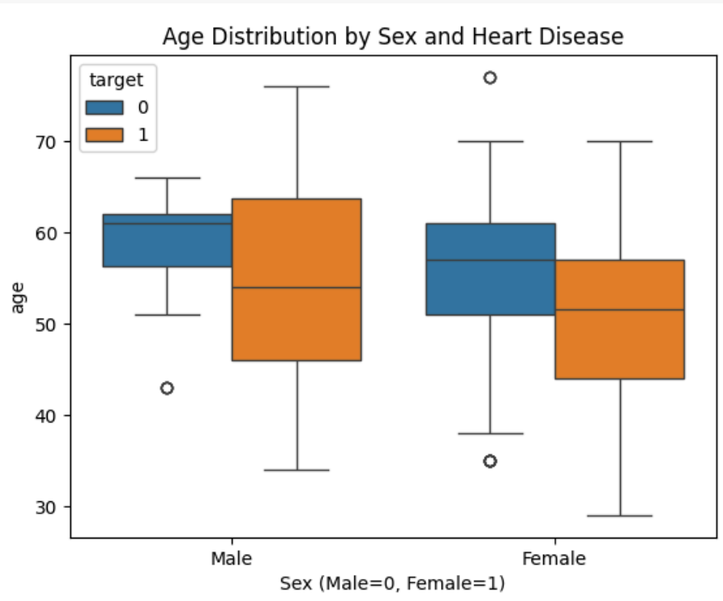
```
<ipython-input-11-ccbaea7c6bf4>:6: FutureWarning:

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.countplot(x='target', data=data, hue_order=[1, 0],  # Specify order for labels
```



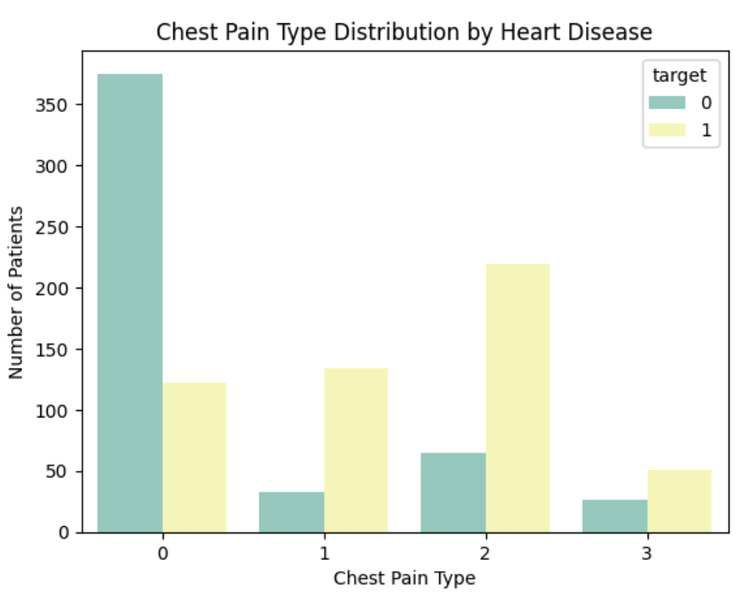

```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix
correlation = data.corr()

# Create a larger figure size for better readability
plt.figure(figsize=(12, 8))  # Adjust width and height as needed

# Generate the heatmap with annotations
sns.heatmap(correlation, annot=True, cmap='coolwarm')

#  labels and title
plt.xlabel('Features')
plt.ylabel('Features')
plt.title('Correlation Heatmap')

# Rotate x-axis labels for better visibility
plt.xticks(rotation=45)
plt.tight_layout()  # Adjust spacing between elements

plt.show()
```




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

# Continuous vs. target variable
sns.boxplot(x='sex', y='age', data=data, hue='target')

#  x-axis labels
plt.xlabel('Sex (Male=0, Female=1)')  # Informative label with coding scheme
plt.xticks([0, 1], ['Male', 'Female'])  # Set custom labels for sex categories

plt.title('Age Distribution by Sex and Heart Disease')
plt.show()
```

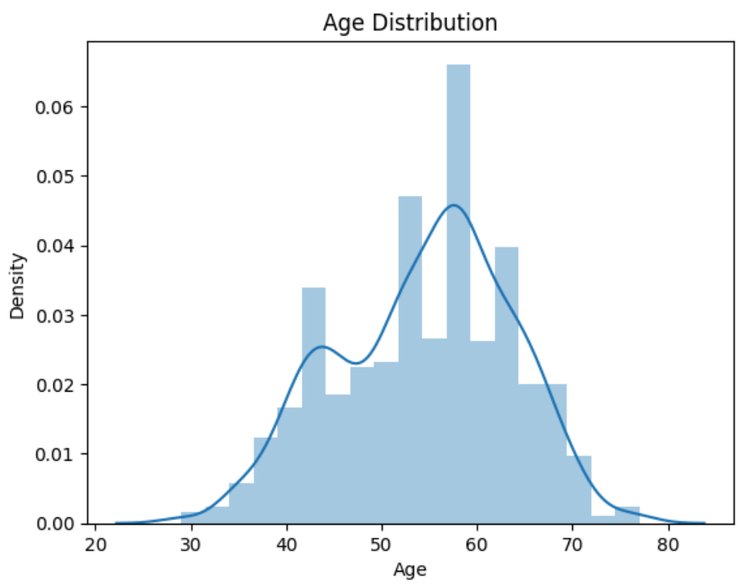


```
import matplotlib.pyplot as plt
import seaborn as sns

# Load your heart disease dataset (replace 'heart.csv' with your actual file path)
data=pd.read_csv("Heart Disease data.csv")

# Assuming the 'cp' feature represents chest pain type with encoded values (0, 1, 2, 3)
# Check the dataset documentation or variable descriptions for the actual chest pain type labels.


chest_pain_labels = {

      0: 'sensation of Chest Pain',
      1: 'slight Chest Pain',
      2: 'mild chest pain',
      3: 'severe chest pain'
}

# Analyze key metrics and relationships
# 3. Categorical vs. target variable (heart disease)
sns.countplot(x='cp', hue='target', data=data, order=chest_pain_labels.keys(),  # Maintain order
              palette='Set3')  # Consider a color palette for better distinction
plt.xlabel('Chest Pain Type')
plt.ylabel('Number of Patients')
plt.title('Chest Pain Type Distribution by Heart Disease')
plt.show()
```

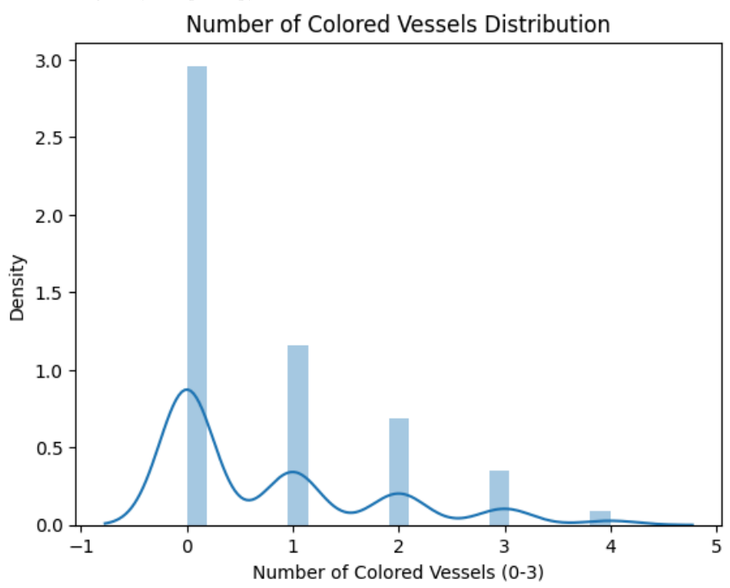



```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'age' and 'ca' are column names
sns.distplot(data['age'])
plt.xlabel('Age')
plt.ylabel('Density')
plt.title('Age Distribution')
plt.show()

sns.distplot(data['ca'])
plt.xlabel('Number of Colored Vessels (0-3)')
plt.ylabel('Density')
plt.title('Number of Colored Vessels Distribution')
plt.show()
```

```
<ipython-input-11-31462c768fcb>:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['age'])
```

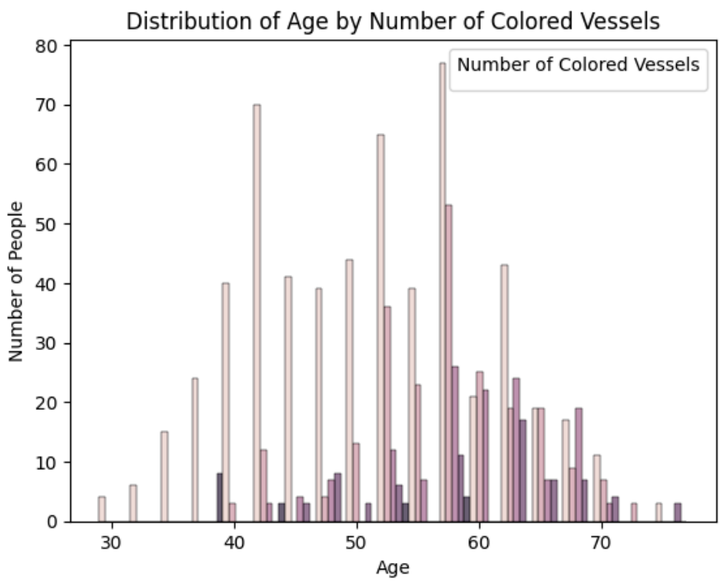



```
<ipython-input-11-31462c768fcb>:11: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['ca'])
```

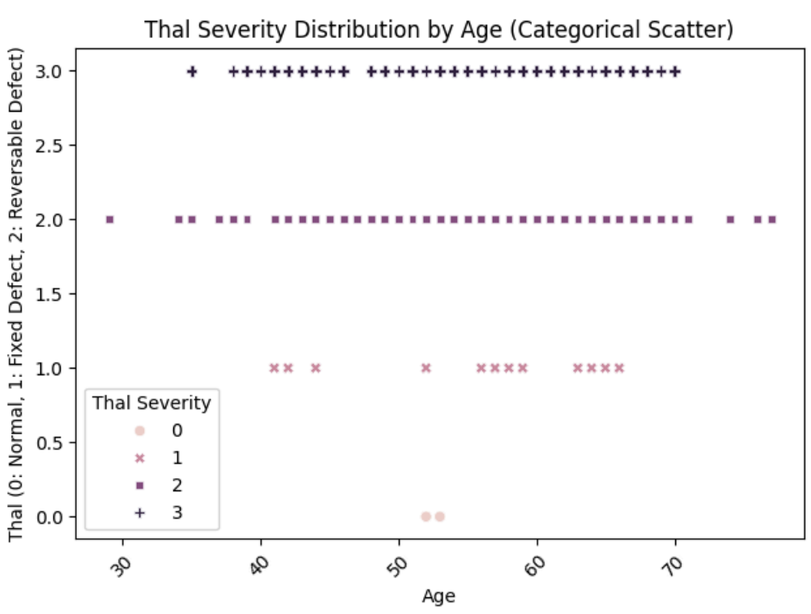


```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'age' and 'ca' are column names
sns.histplot(x='age', hue='ca', data=data, multiple='dodge', alpha=0.7)  # Adjust alpha for transparency
plt.xlabel('Age')
plt.ylabel('Number of People')
plt.title('Distribution of Age by Number of Colored Vessels')
plt.legend(title='Number of Colored Vessels')
plt.show()
```

```
WARNING:matplotlib.legend: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.
```




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

# Assuming 'age' and 'thal' are column names
sns.scatterplot(
    x='age',
    y='thal',
    hue='thal',  # Use 'thal' as the hue variable for coloring by category
    data=data,
    style='thal',  # Use 'thal' as the style variable for marker shapes
)
plt.xlabel('Age')
plt.ylabel('Thal (0: Normal, 1: Fixed Defect, 2: Reversable Defect)')
plt.title('Thal Severity Distribution by Age (Categorical Scatter)')
plt.xticks(rotation=45)  #x-axis
plt.legend(title='Thal Severity')  # legend
plt.tight_layout()
plt.show()
```




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

# Assuming 'sex' and 'oldpeak' are column names
sns.boxplot(
    x='sex',
    y='oldpeak',
    showmeans=True,
    data=data
)  # Show mean as a point within each box
plt.xlabel('Sex (0: male, 1: Female)')
plt.ylabel('Oldpeak (ST depression)')
plt.title('Oldpeak Distribution by Sex (Side-by-Side Box Plots)')
plt.xticks([0, 1], ['male', 'Female'])  # Set custom x-axis labels for clarity
plt.tight_layout()
plt.show()
```