UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

HEALTH DATA ANALYSIS PROJECT

Student Name: Arpita Nayak **UID:** 24MCI10001
Branch: MCA (AIML) **Section/Group:** 1/A
Semester:1 **Date of Performance:** 28/10/2024
Subject Name: Python Programming **Subject Code:** 24CAH-606

Q. Project on Health Data Analysis

- **Collect health-related data such as patient records, vital signs, and medical test results.**
- **Use NumPy for data manipulation, cleaning, and filtering operations.**
- **Visualize patient demographics using histograms or bar plots.**
- **Create scatter plots to analyze correlations between variables like age, blood pressure, and cholesterol levels etc depending on data available**

1. **Aim/Overview of the project:**

   - The primary objective of this project is to conduct a comprehensive health data analysis by utilizing Python programming language and its libraries, particularly NumPy, Matplotlib, Pandas and Seaborn.

   - The project aims to explore health-related data encompassing patient records, vital signs, and medical test results, thereby providing insights into various health parameters.

   - The visualization techniques employed will facilitate a better understanding of patient demographics and the relationships between different health variables, ultimately contributing to the assessment of heart disease risk factors.

2. Task to be done:

a) **Data Collection and Preparation**: Gather health-related data, including patient records, vital signs, and medical test results, to establish a robust dataset for analysis.

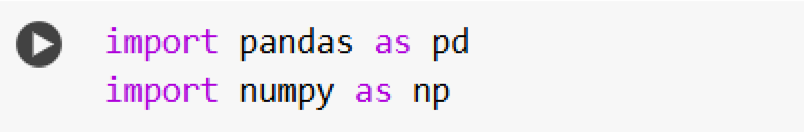
b) **Data Manipulation**: Utilize NumPy for data cleaning, transformation, and filtering operations to ensure data integrity and accuracy.

c) **Data Visualization**: Employ Matplotlib and Seaborn libraries to create a variety of visualizations, including:

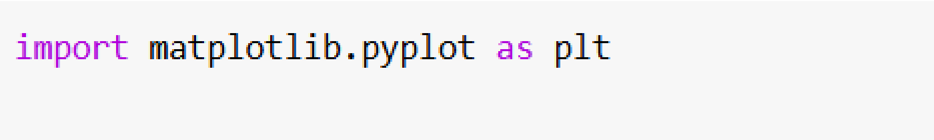
- **Scatter Plots**: Illustrate the correlation between age and cholesterol levels while examining the prevalence of heart disease.
- **Histograms**: Display the distribution of cholesterol levels among the patient population.
- **Bar Plots**: Present average cholesterol levels segmented by heart disease status.
- **Box Plots**: Compare resting blood pressure across different heart disease statuses.
- **Violin Plots**: Analyze the distribution of age against resting blood pressure.
- **Pie Charts**: Visualize the distribution of heart disease status among male and female patients.
- **Heatmaps**: Explore the interaction between age groups, cholesterol levels, and heart disease prevalence.

3. Codes and outputs:

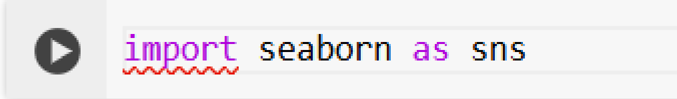
- **IMPORTING NECESSARY LIBRARIES:**



```
import pandas as pd
import numpy as np
```



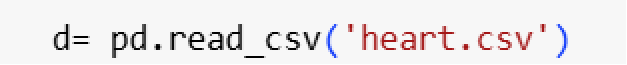
```
import matplotlib.pyplot as plt
```



```
import seaborn as sns
```

- **SAVING DATASET TO A VARIABLE FOR FURTHER USE:**

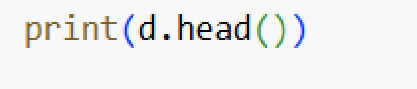
CODE:



```
d= pd.read_csv('heart.csv')
```

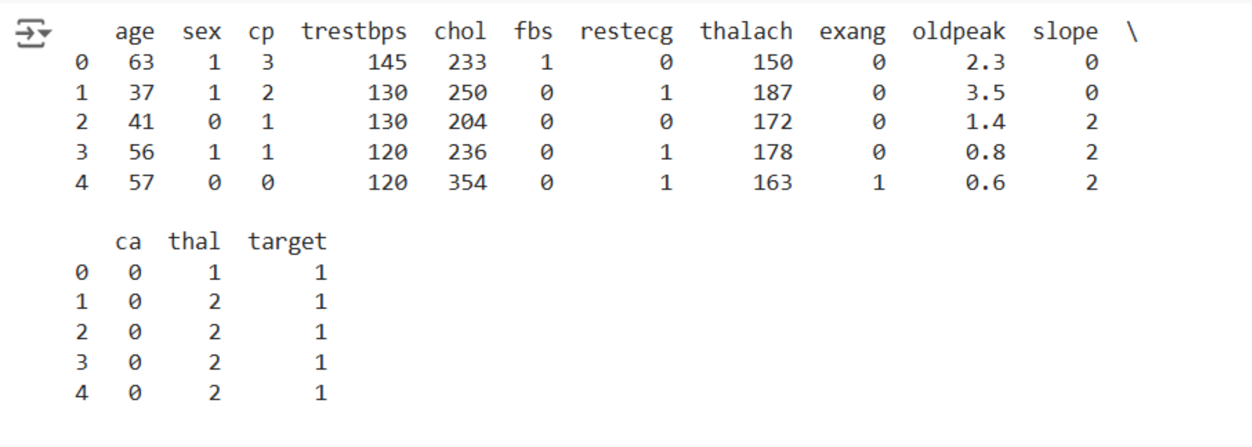
- **PRINTING HEAD OF THE DATASET(BY DEFAULT FIRST 5 ROWS OF EACH COLUMN IS DISPLAYED):**

CODE:



```
print(d.head())
```

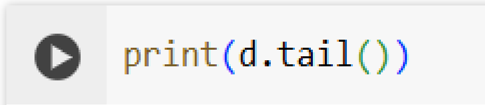
OUTPUT:



```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   3       145   233    1        0      150      0      2.3      0   
1   37    1   2       130   250    0        1      187      0      3.5      0   
2   41    0   1       130   204    0        0      172      0      1.4      2   
3   56    1   1       120   236    0        1      178      0      0.8      2   
4   57    0   0       120   354    0        1      163      1      0.6      2   

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

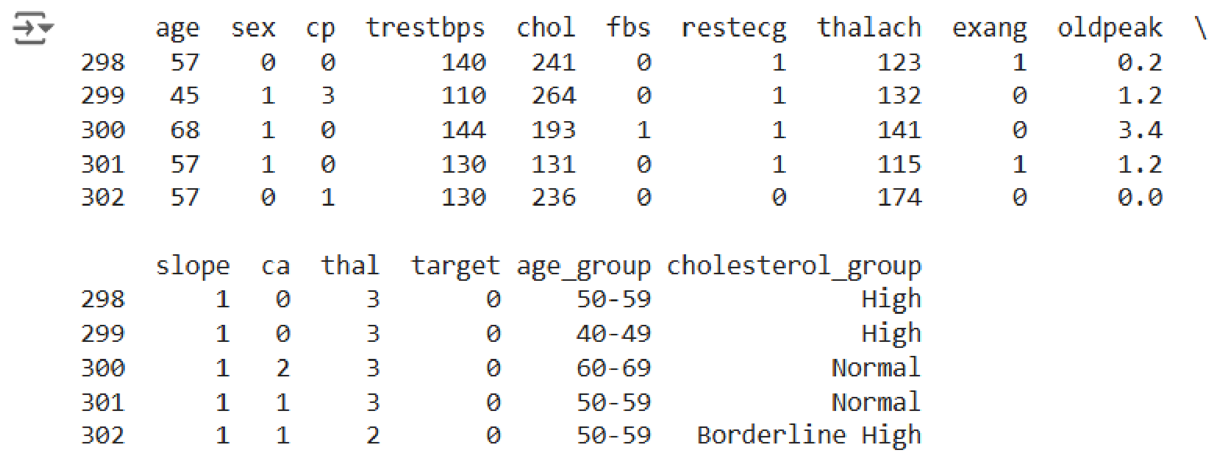
- **PRINTING TAIL OF THE DATASET(BY DEFAULT LAST 5 ROWS OF EACH COLUMN IS DISPLAYED):**

CODE:



```
print(d.tail())
```

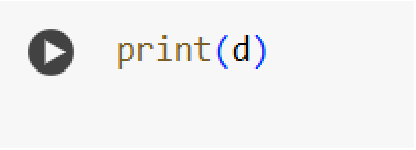
OUTPUT:



```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
298   57    0   0       140   241    0        1      123      1      0.2   
299   45    1   3       110   264    0        1      132      0      1.2   
300   68    1   0       144   193    1        1      141      0      3.4   
301   57    1   0       130   131    0        1      115      1      1.2   
302   57    0   1       130   236    0        0      174      0      0.0   

     slope  ca  thal  target age_group cholesterol_group  
298      1   0     3       0     50-59              High  
299      1   0     3       0     40-49              High  
300      1   2     3       0     60-69            Normal  
301      1   1     3       0     50-59            Normal  
302      1   1     2       0     50-59   Borderline High  
```

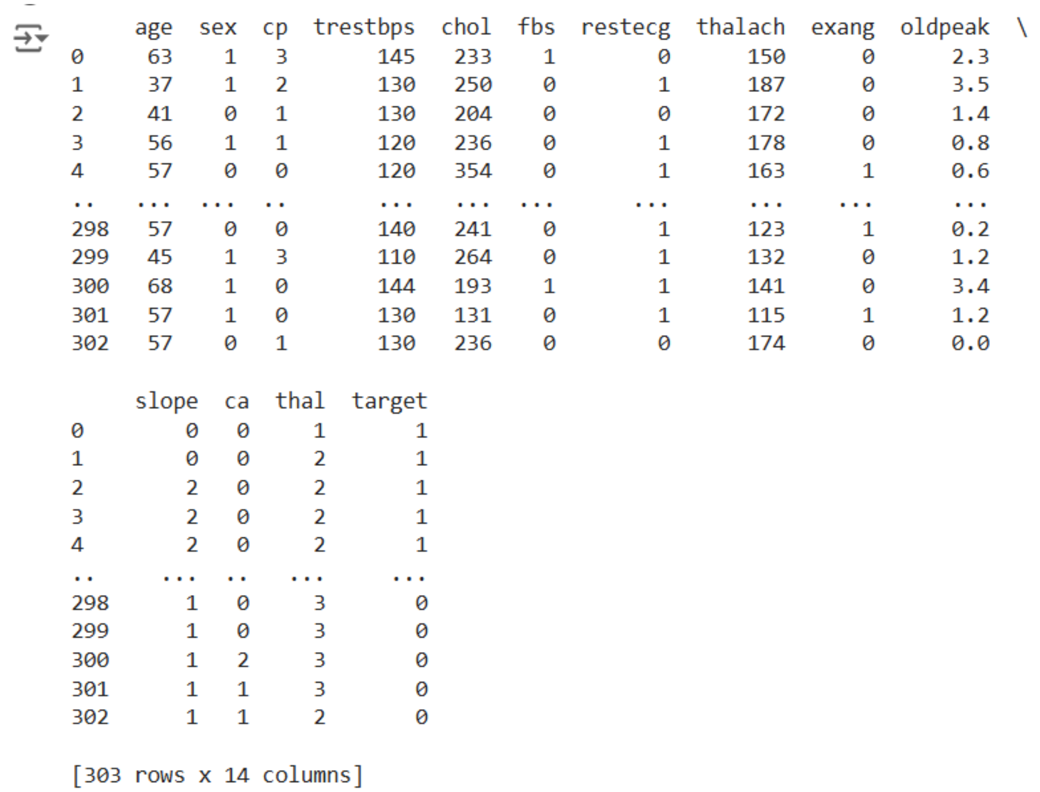
- **<u>PRINTING FULL DATASET:</u>**

CODE:



```
print(d)
```

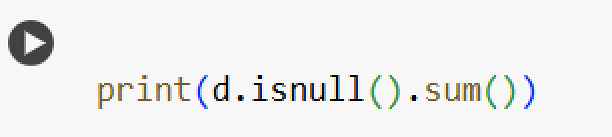
OUTPUT:



```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3   
1     37    1   2       130   250    0        1      187      0      3.5   
2     41    0   1       130   204    0        0      172      0      1.4   
3     56    1   1       120   236    0        1      178      0      0.8   
4     57    0   0       120   354    0        1      163      1      0.6   
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...   
298   57    0   0       140   241    0        1      123      1      0.2   
299   45    1   3       110   264    0        1      132      0      1.2   
300   68    1   0       144   193    1        1      141      0      3.4   
301   57    1   0       130   131    0        1      115      1      1.2   
302   57    0   1       130   236    0        0      174      0      0.0   

     slope  ca  thal  target  
0        0   0     1       1  
1        0   0     2       1  
2        2   0     2       1  
3        2   0     2       1  
4        2   0     2       1  
..     ...  ..   ...     ...  
298      1   0     3       0  
299      1   0     3       0  
300      1   2     3       0  
301      1   1     3       0  
302      1   1     2       0  

[303 rows x 14 columns]
```

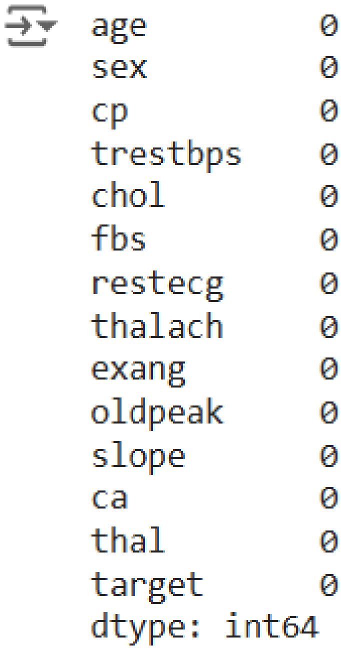
- **PRINTING COUNT OF NULL VALUES IN EACH COLUMN**

CODE:



```
print(d.isnull().sum())
```

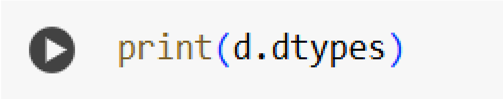
OUTPUT:



```
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
```

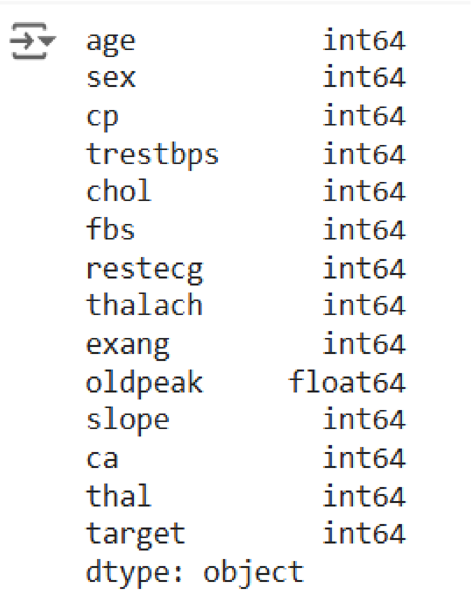
- **PRINTING DATATYPES:**

CODE:



```
print(d.dtypes)
```

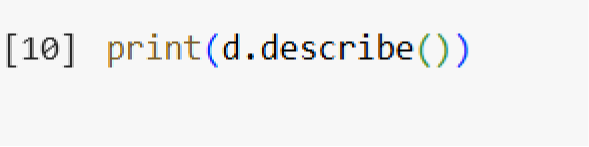
OUTPUT:



```
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
```

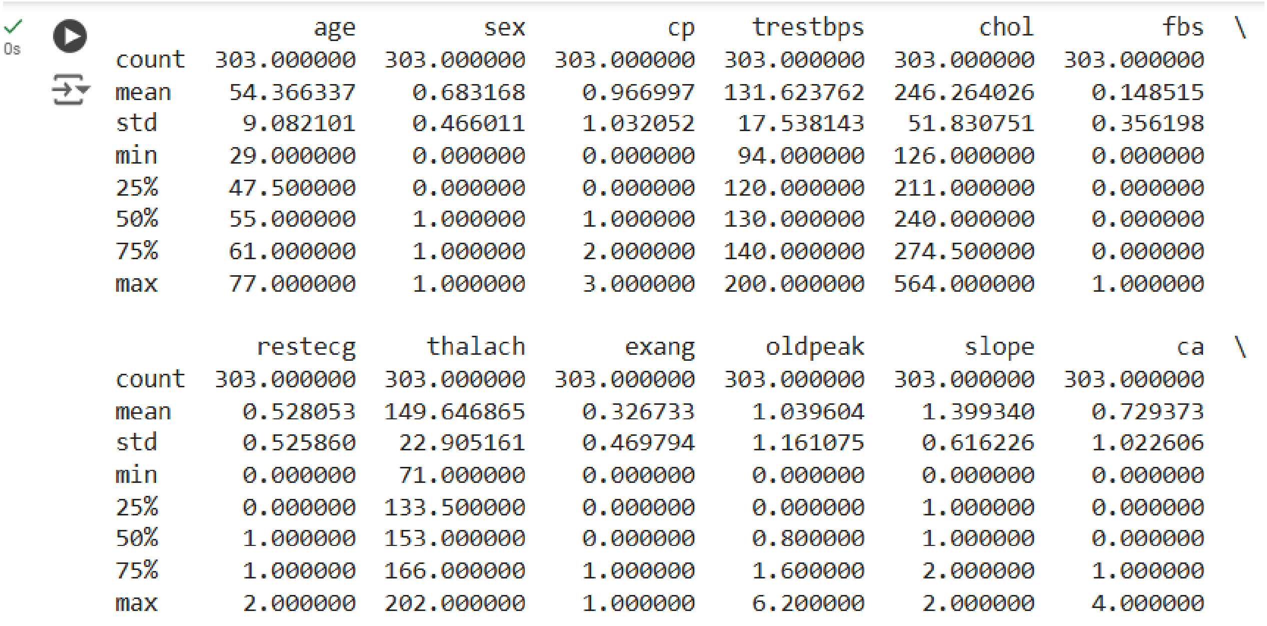
- **PRINTING SUMMARY OF THE DATASET(LIKE MIN, MAX, STANDARD DEVIATION, COUNT, QUANTILES,ETC):**

CODE:

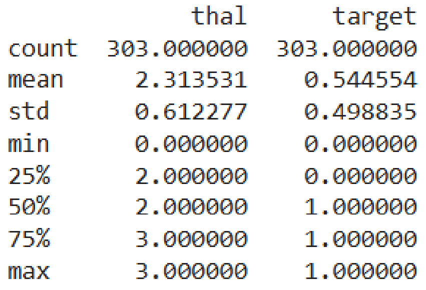


```
[10] print(d.describe())
```

OUTPUT:



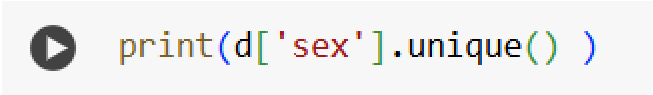
```
              age         sex          cp    trestbps        chol         fbs  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean    54.366337    0.683168    0.966997  131.623762  246.264026    0.148515
std      9.082101    0.466011    1.032052   17.538143   51.830751    0.356198
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000
25%     47.500000    0.000000    0.000000  120.000000  211.000000    0.000000
50%     55.000000    1.000000    1.000000  130.000000  240.000000    0.000000
75%     61.000000    1.000000    2.000000  140.000000  274.500000    0.000000
max     77.000000    1.000000    3.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean     0.528053  149.646865    0.326733    1.039604    1.399340    0.729373
std      0.525860   22.905161    0.469794    1.161075    0.616226    1.022606
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  133.500000    0.000000    0.000000    1.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    1.000000    0.000000
75%      1.000000  166.000000    1.000000    1.600000    2.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    2.000000    4.000000
```



```
             thal      target
count  303.000000  303.000000
mean     2.313531    0.544554
std      0.612277    0.498835
min      0.000000    0.000000
25%      2.000000    0.000000
50%      2.000000    1.000000
75%      3.000000    1.000000
max      3.000000    1.000000
```

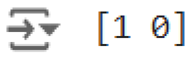
- **<u>PRINTING UNIQUE VALUES OF SOME COLUMNS IN ORDER TO CHECK IF THE DATA IS CORRECT OR NOT:</u>**

CODE:



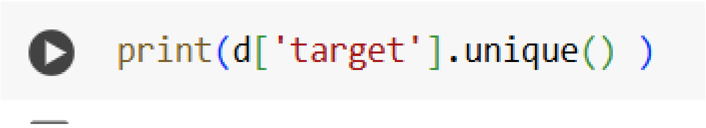
```
print(d['sex'].unique() )
```

OUTPUT:



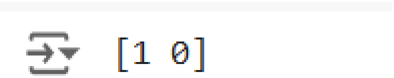
```
[1 0]
```

CODE:



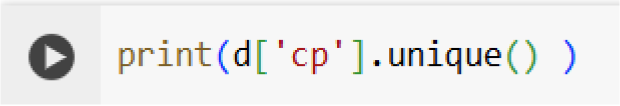
```
print(d['target'].unique() )
```

OUTPUT:



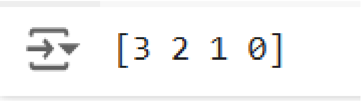
```
[1 0]
```

CODE:



```
print(d['cp'].unique() )
```

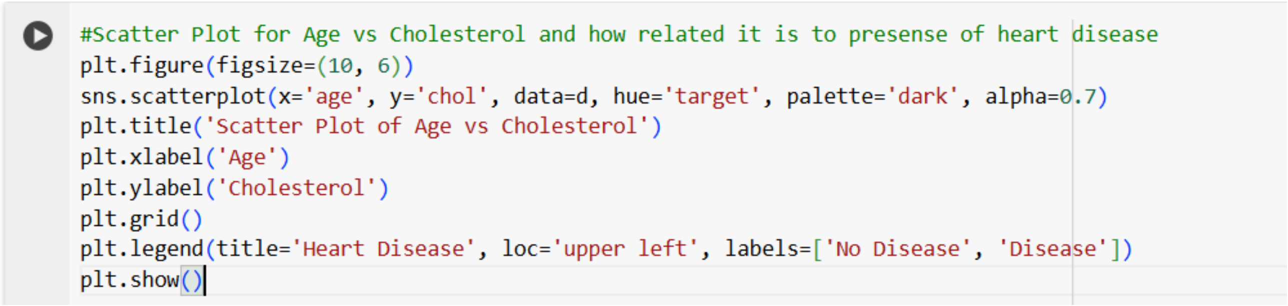
OUTPUT:



```
[3 2 1 0]
```

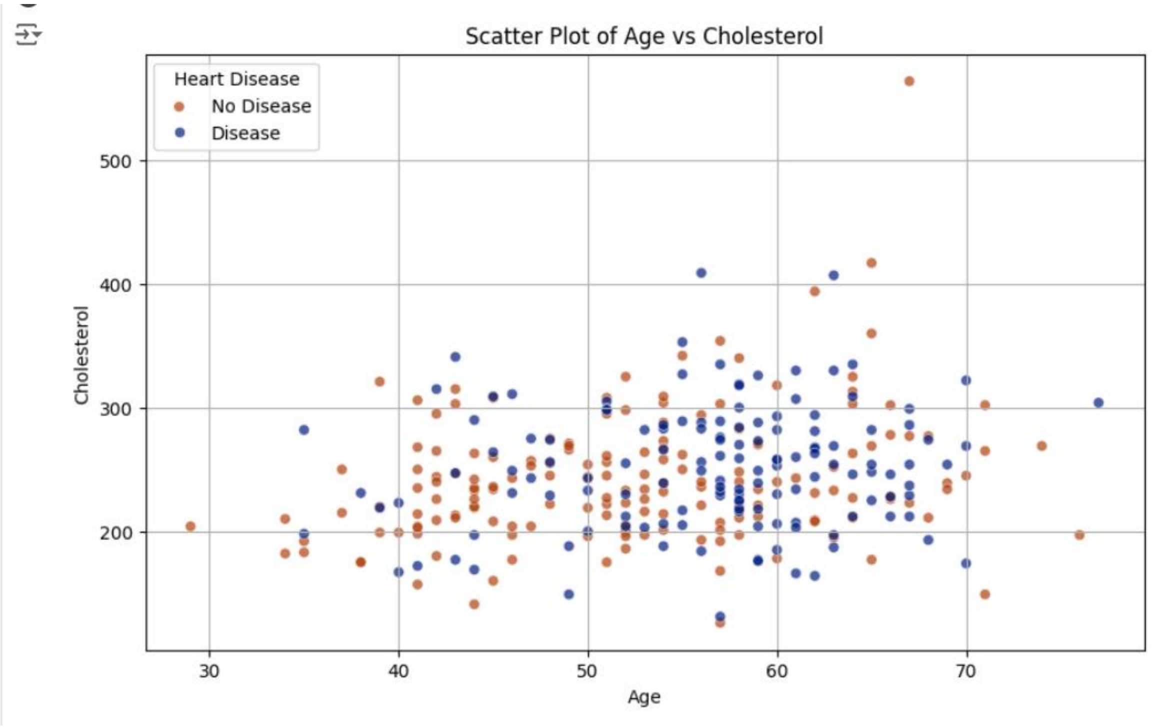
- **SCATTER PLOT FOR AGE VS CHOLESTEROL AND HOW RELATED IT IS TO PRESENSE OF HEART DISEASE:**

CODE:



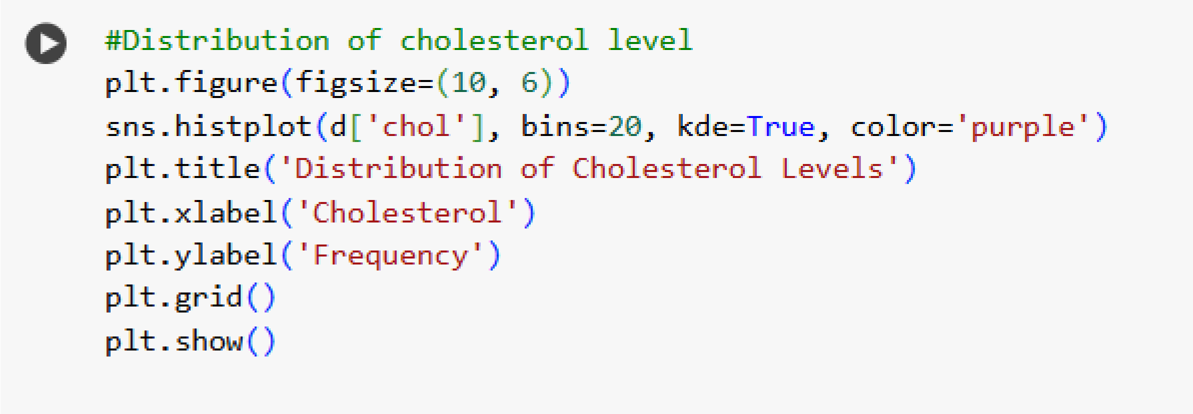
```
#Scatter Plot for Age vs Cholesterol and how related it is to presense of heart disease
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age', y='chol', data=d, hue='target', palette='dark', alpha=0.7)
plt.title('Scatter Plot of Age vs Cholesterol')
plt.xlabel('Age')
plt.ylabel('Cholesterol')
plt.grid()
plt.legend(title='Heart Disease', loc='upper left', labels=['No Disease', 'Disease'])
plt.show()
```

OUTPUT:



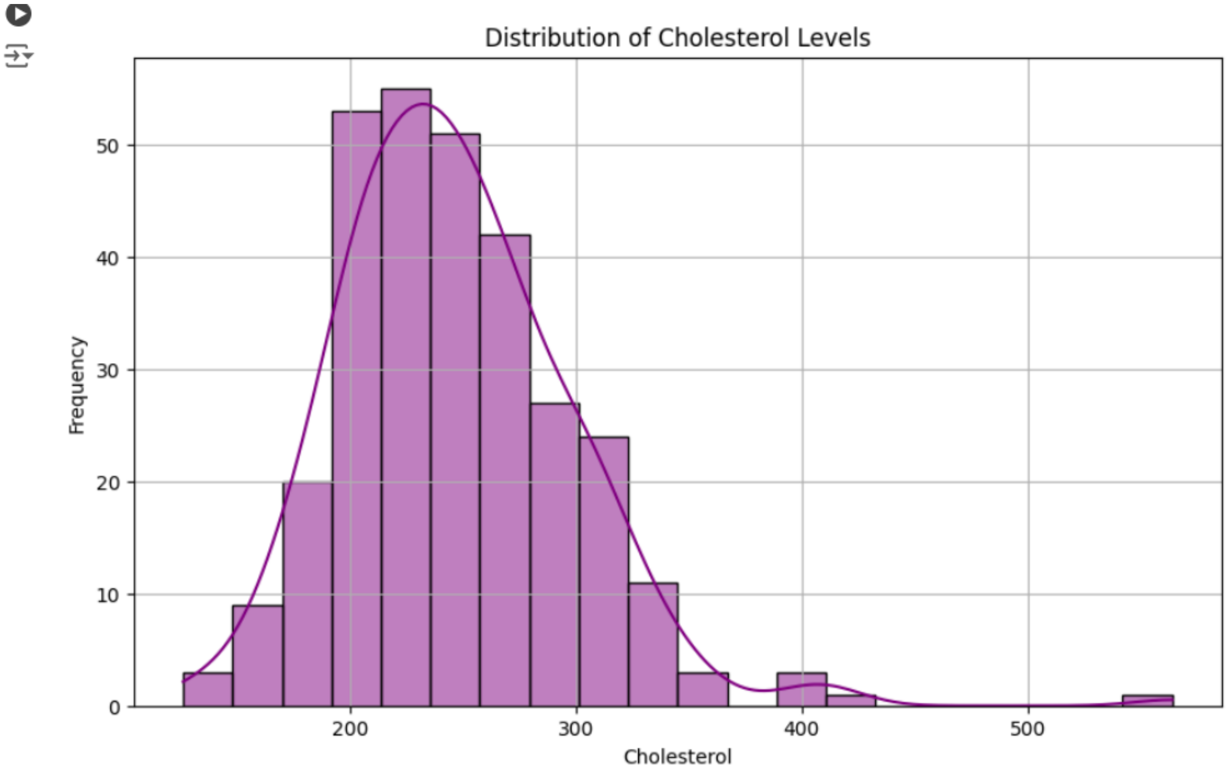
- **HISTOGRAM TO DEPICT DISTRIBUTION OF CHOLESTROL LEVELS:**

CODE:



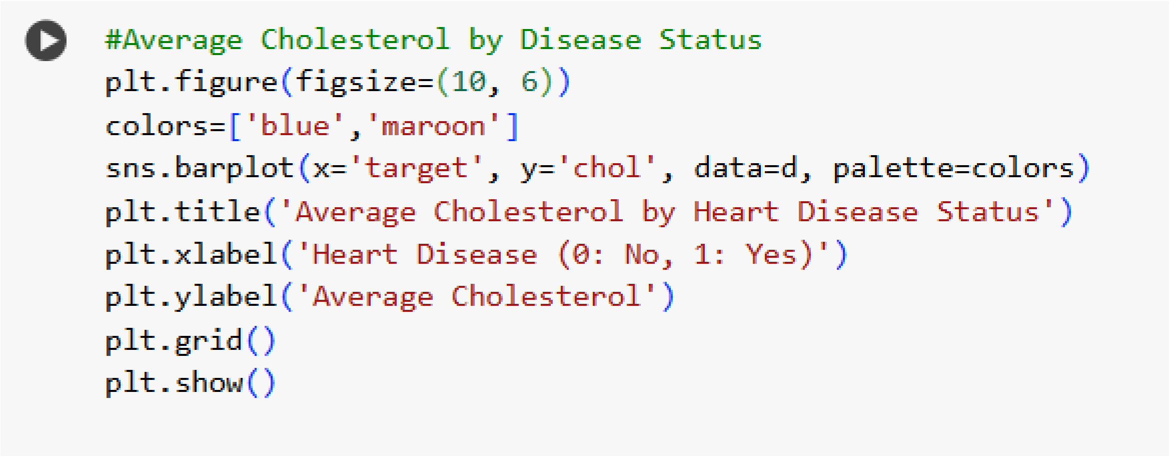
```
#Distribution of cholesterol level
plt.figure(figsize=(10, 6))
sns.histplot(d['chol'], bins=20, kde=True, color='purple')
plt.title('Distribution of Cholesterol Levels')
plt.xlabel('Cholesterol')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```

OUTPUT:



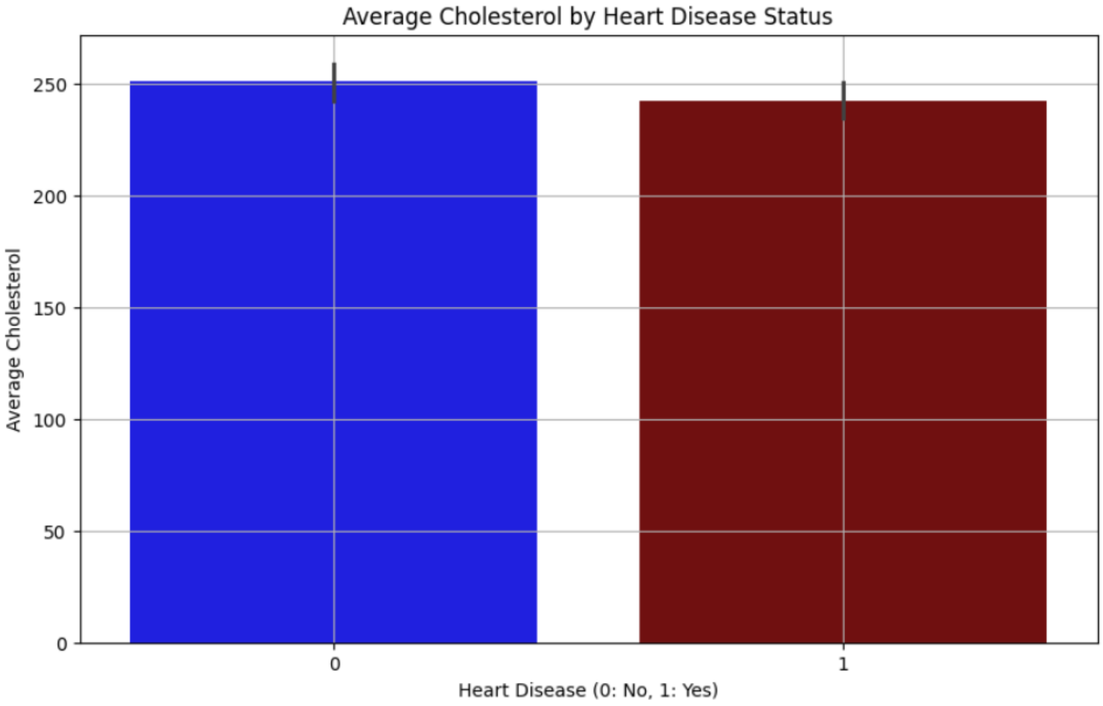
- **BAR PLOT TO SHOW RELATION BETWEEN AVERAGE CHOLESTROL LEVEL AND THE OCCURRENCE OF HEART DISEASE:**

CODE:



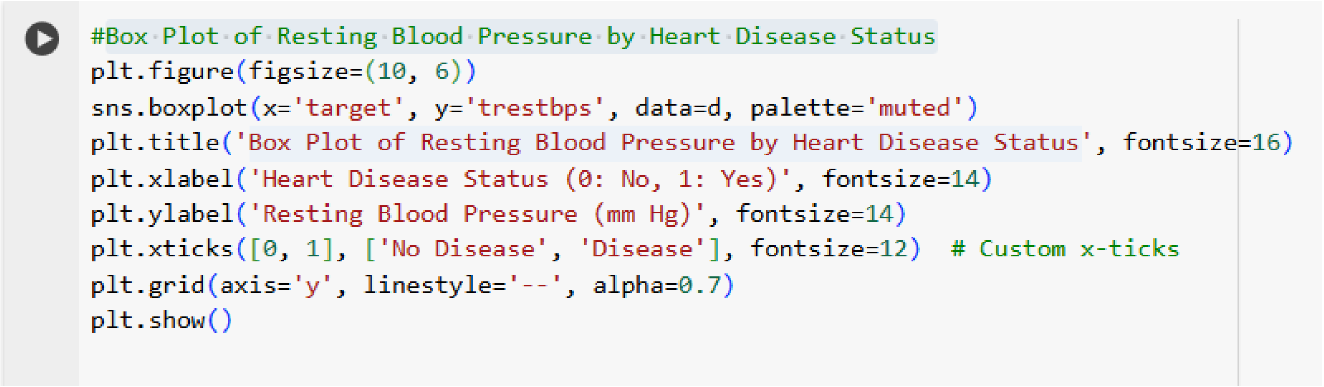
```
#Average Cholesterol by Disease Status
plt.figure(figsize=(10, 6))
colors=['blue','maroon']
sns.barplot(x='target', y='chol', data=d, palette=colors)
plt.title('Average Cholesterol by Heart Disease Status')
plt.xlabel('Heart Disease (0: No, 1: Yes)')
plt.ylabel('Average Cholesterol')
plt.grid()
plt.show()
```

OUTPUT:



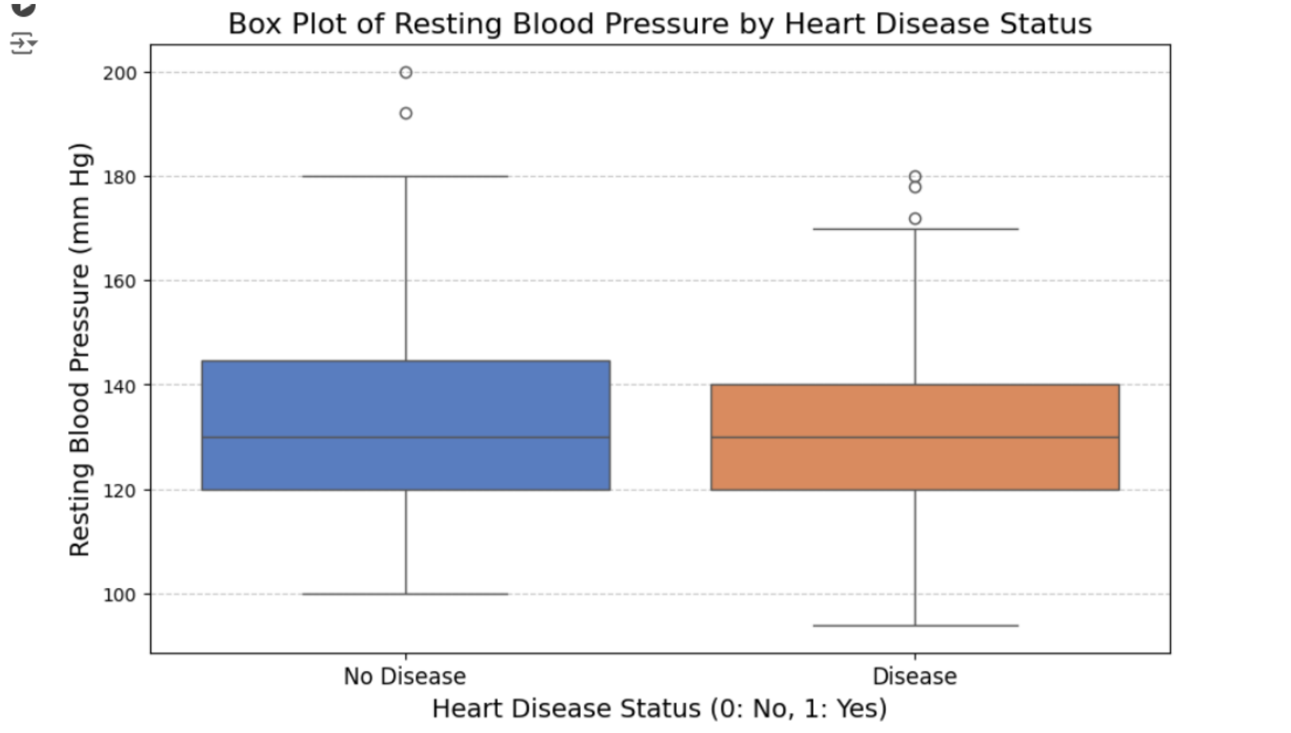
- **BOX PLOT FOR CHECKING RELATION BETWEEN RESTING BLOOD PRESSURE AND OCCURANCE HEART DISEASE :**

CODE:



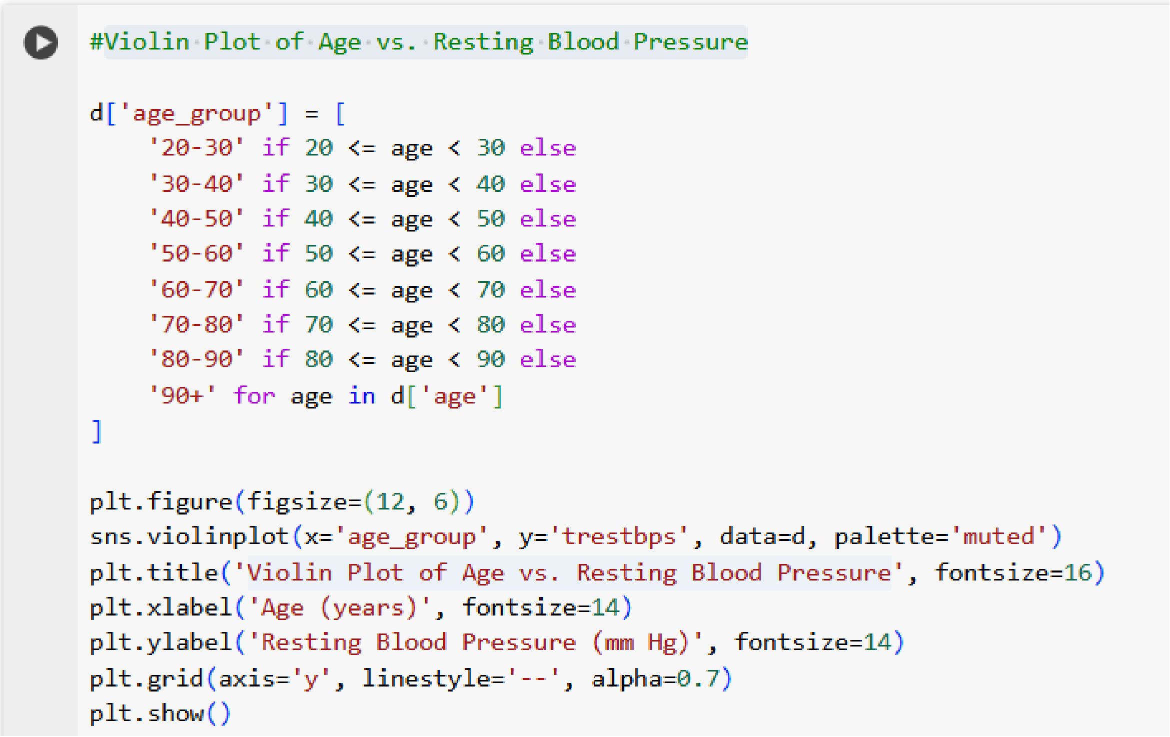
```
#Box Plot of Resting Blood Pressure by Heart Disease Status
plt.figure(figsize=(10, 6))
sns.boxplot(x='target', y='trestbps', data=d, palette='muted')
plt.title('Box Plot of Resting Blood Pressure by Heart Disease Status', fontsize=16)
plt.xlabel('Heart Disease Status (0: No, 1: Yes)', fontsize=14)
plt.ylabel('Resting Blood Pressure (mm Hg)', fontsize=14)
plt.xticks([0, 1], ['No Disease', 'Disease'], fontsize=12)  # Custom x-ticks
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

OUTPUT:



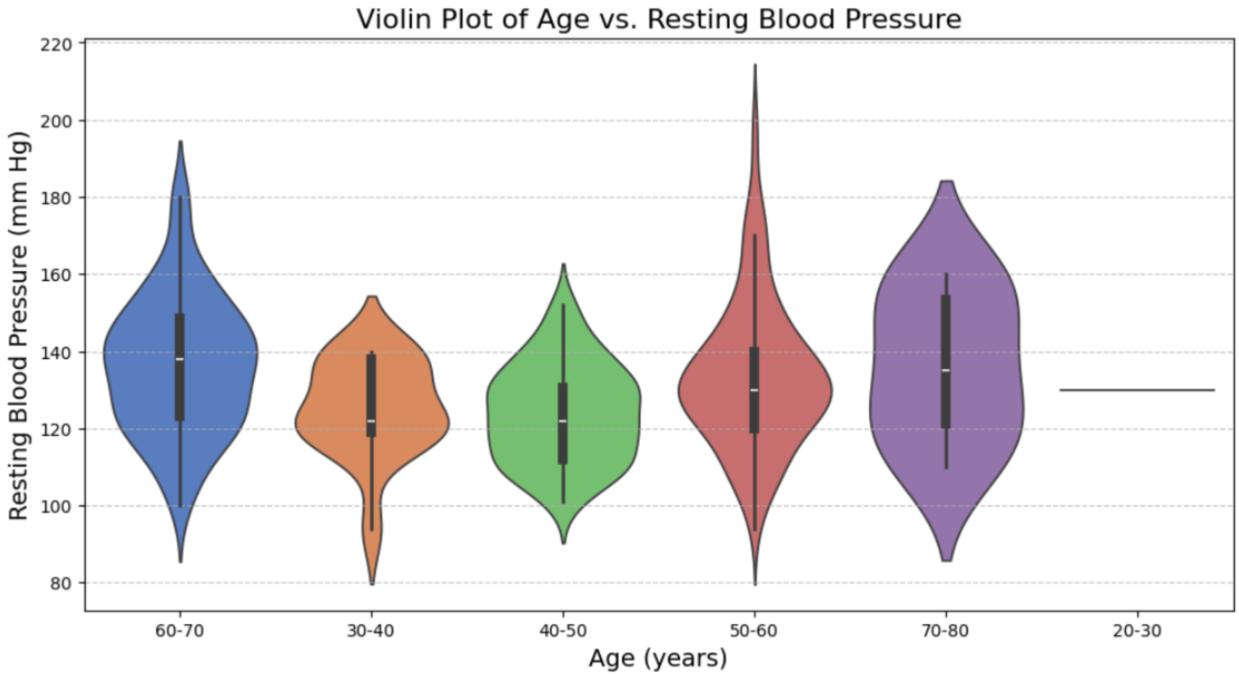
- **VIOLIN PLOT FOR AGE VS. RESTING BLOOD PRESSURE:**

CODE:



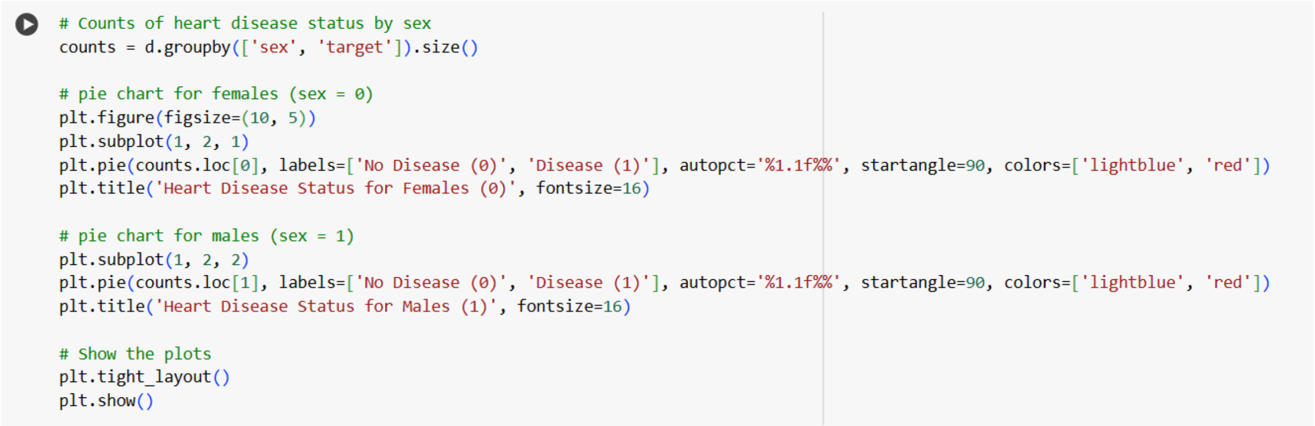
```
#Violin Plot of Age vs. Resting Blood Pressure

d['age_group'] = [
    '20-30' if 20 <= age < 30 else
    '30-40' if 30 <= age < 40 else
    '40-50' if 40 <= age < 50 else
    '50-60' if 50 <= age < 60 else
    '60-70' if 60 <= age < 70 else
    '70-80' if 70 <= age < 80 else
    '80-90' if 80 <= age < 90 else
    '90+' for age in d['age']
]

plt.figure(figsize=(12, 6))
sns.violinplot(x='age_group', y='trestbps', data=d, palette='muted')
plt.title('Violin Plot of Age vs. Resting Blood Pressure', fontsize=16)
plt.xlabel('Age (years)', fontsize=14)
plt.ylabel('Resting Blood Pressure (mm Hg)', fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

OUTPUT:



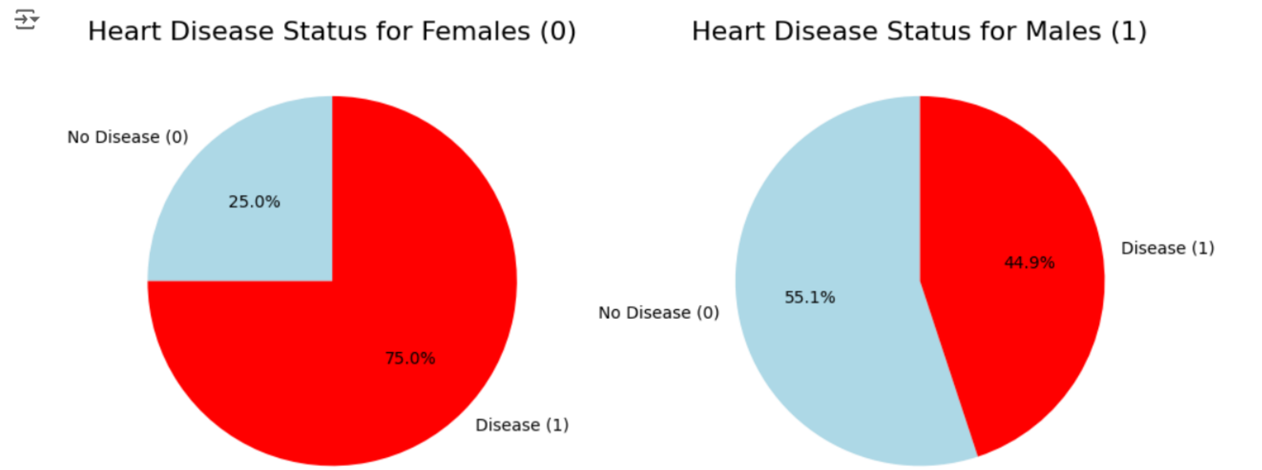
- **PIE CHARTS TO DEPICT PERCENTAGE OF PEOPLE OF HEART DISEASE IN MALE AND FEMALE:**

CODE:



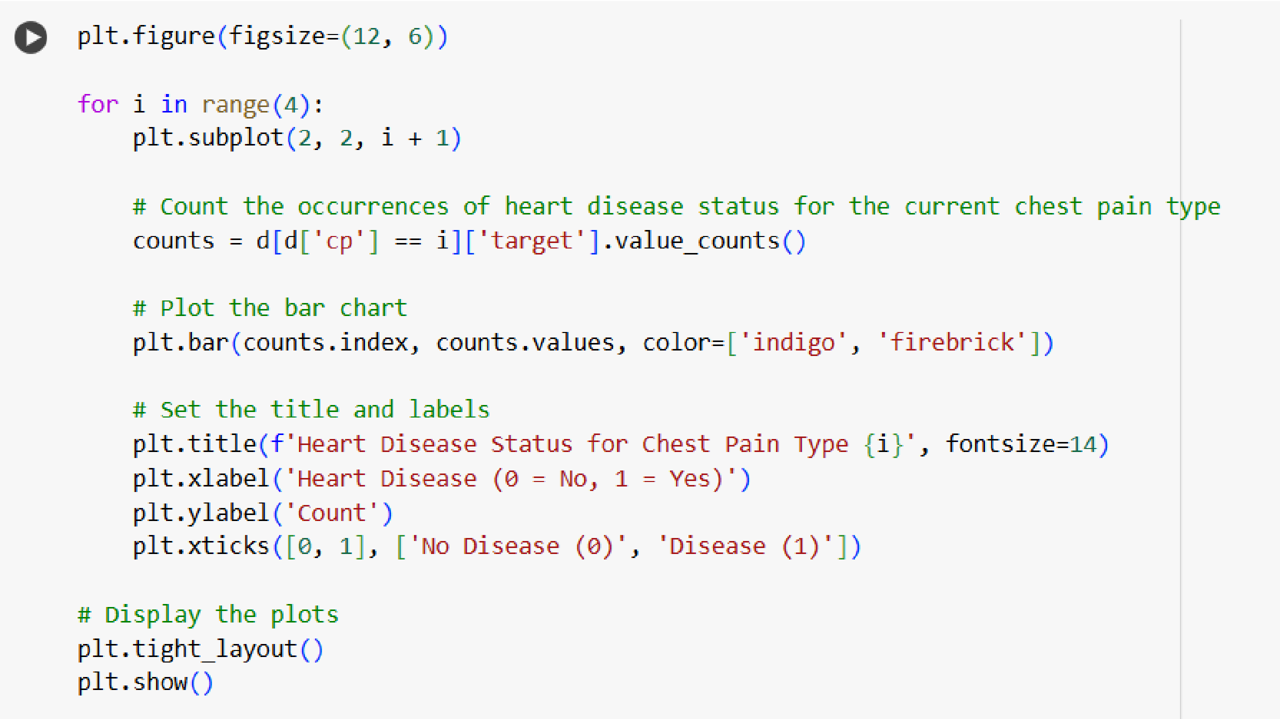
```
# Counts of heart disease status by sex
counts = d.groupby(['sex', 'target']).size()

# pie chart for females (sex = 0)
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.pie(counts.loc[0], labels=['No Disease (0)', 'Disease (1)'], autopct='%1.1f%%', startangle=90, colors=['lightblue', 'red'])
plt.title('Heart Disease Status for Females (0)', fontsize=16)

# pie chart for males (sex = 1)
plt.subplot(1, 2, 2)
plt.pie(counts.loc[1], labels=['No Disease (0)', 'Disease (1)'], autopct='%1.1f%%', startangle=90, colors=['lightblue', 'red'])
plt.title('Heart Disease Status for Males (1)', fontsize=16)

# Show the plots
plt.tight_layout()
plt.show()
```

OUTPUT:



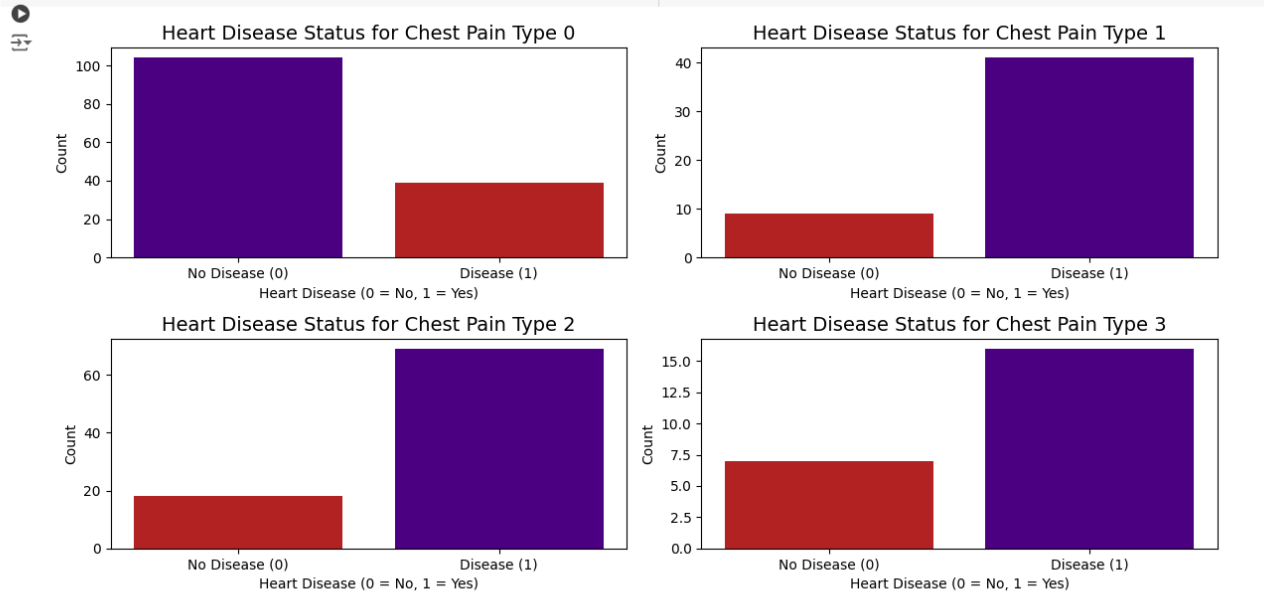
- **SUB BAR PLOTS TO SEE RELATION BETWEEN EACH TYPES OF CHEST PAIN AND THE CHANCES OF THEM HAVING A HEART DISEASE:**

CODE:



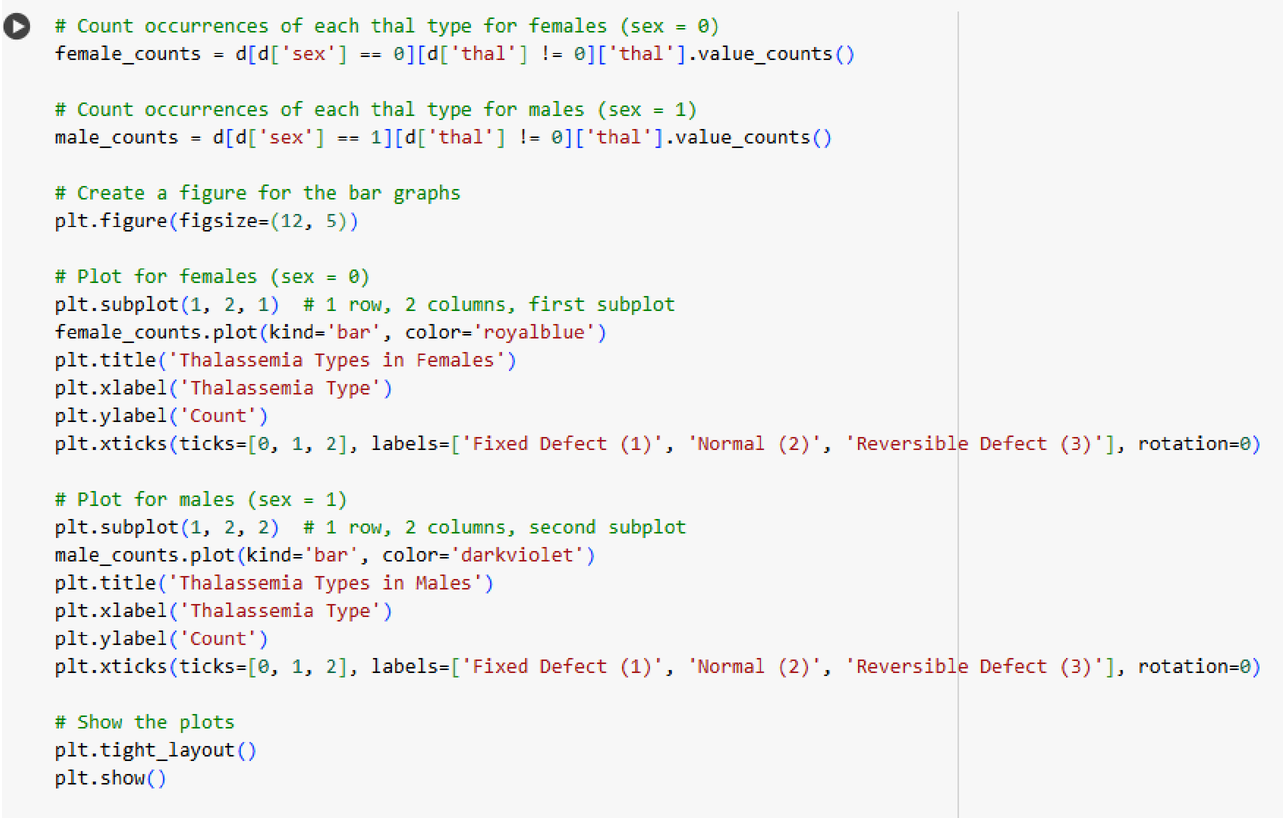
```
plt.figure(figsize=(12, 6))

for i in range(4):
    plt.subplot(2, 2, i + 1)

    # Count the occurrences of heart disease status for the current chest pain type
    counts = d[d['cp'] == i]['target'].value_counts()

    # Plot the bar chart
    plt.bar(counts.index, counts.values, color=['indigo', 'firebrick'])

    # Set the title and labels
    plt.title(f'Heart Disease Status for Chest Pain Type {i}', fontsize=14)
    plt.xlabel('Heart Disease (0 = No, 1 = Yes)')
    plt.ylabel('Count')
    plt.xticks([0, 1], ['No Disease (0)', 'Disease (1)'])

# Display the plots
plt.tight_layout()
plt.show()
```

OUTPUT:



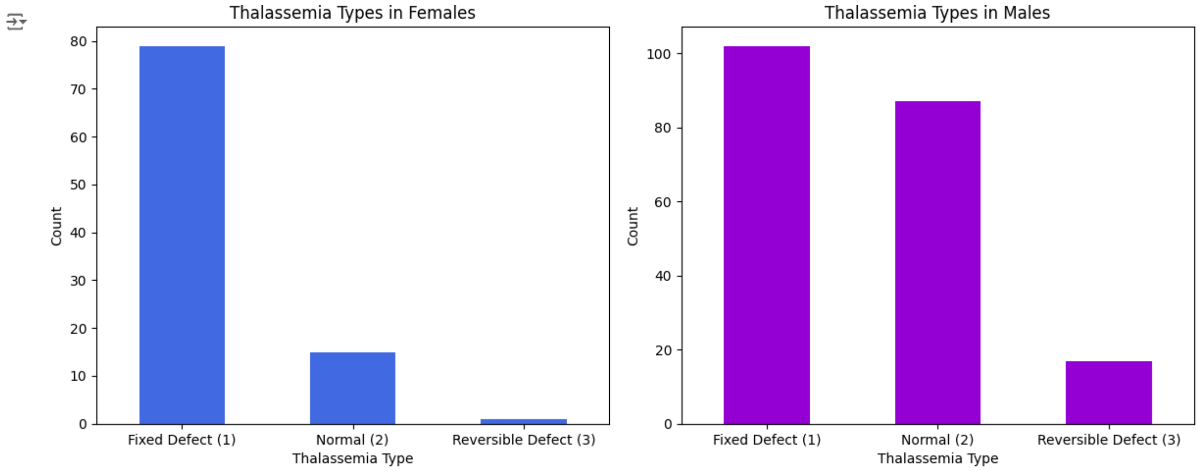
- **DEPICTING FREQUENCY OF MALES AND FEMALES HAVING DIFFERENT THALASSEMIA TYPES USING BAR SUBPLOTS:**

CODE:



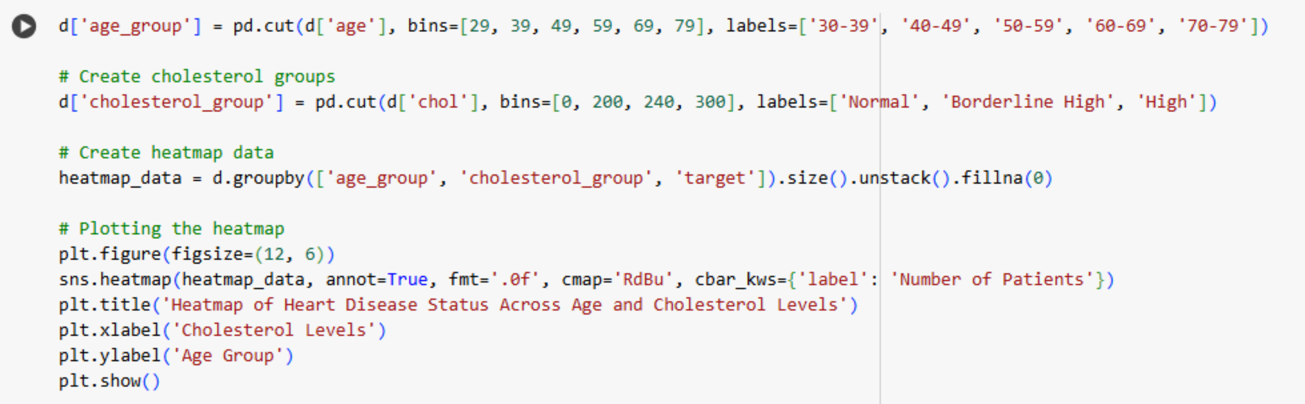
```
# Count occurrences of each thal type for females (sex = 0)
female_counts = d[d['sex'] == 0][d['thal'] != 0]['thal'].value_counts()

# Count occurrences of each thal type for males (sex = 1)
male_counts = d[d['sex'] == 1][d['thal'] != 0]['thal'].value_counts()

# Create a figure for the bar graphs
plt.figure(figsize=(12, 5))

# Plot for females (sex = 0)
plt.subplot(1, 2, 1)  # 1 row, 2 columns, first subplot
female_counts.plot(kind='bar', color='royalblue')
plt.title('Thalassemia Types in Females')
plt.xlabel('Thalassemia Type')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1, 2], labels=['Fixed Defect (1)', 'Normal (2)', 'Reversible Defect (3)'], rotation=0)

# Plot for males (sex = 1)
plt.subplot(1, 2, 2)  # 1 row, 2 columns, second subplot
male_counts.plot(kind='bar', color='darkviolet')
plt.title('Thalassemia Types in Males')
plt.xlabel('Thalassemia Type')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1, 2], labels=['Fixed Defect (1)', 'Normal (2)', 'Reversible Defect (3)'], rotation=0)

# Show the plots
plt.tight_layout()
plt.show()
```

OUTPUT:



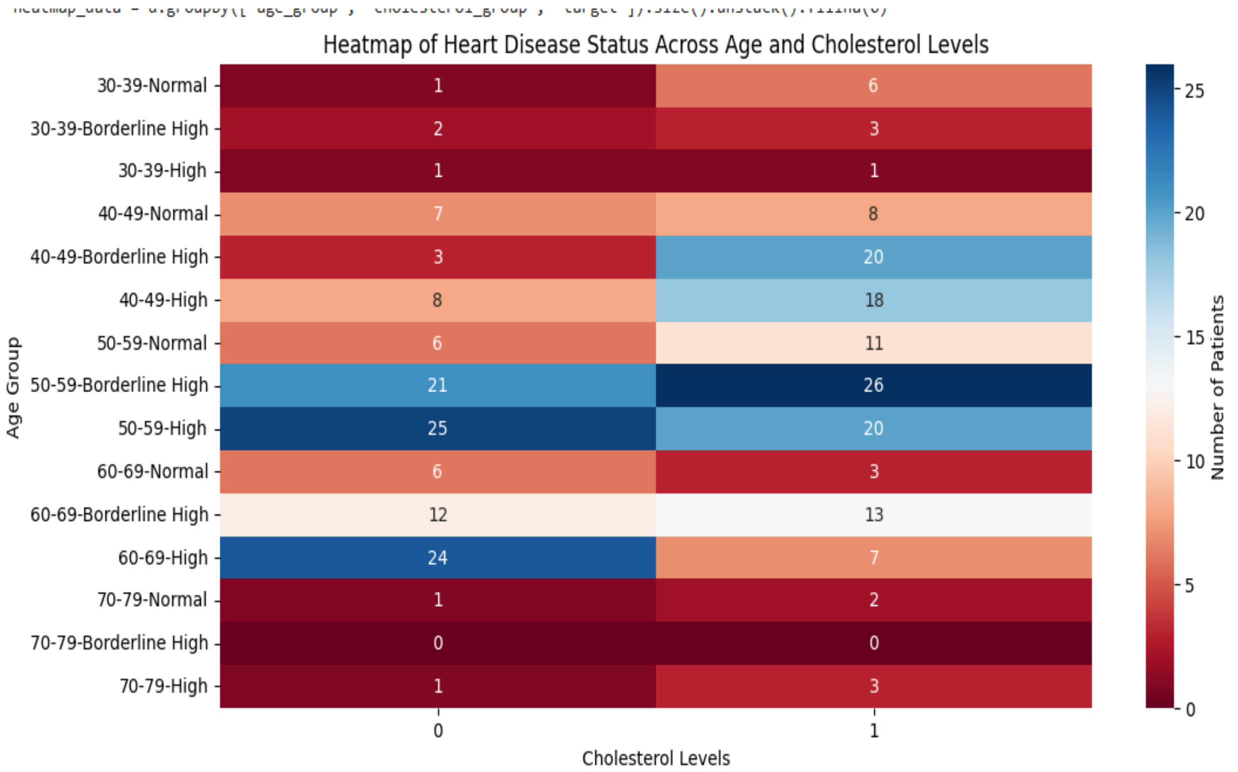
- **HEATMAP OF HEART DISEASE STATUS ACROSS AGE AND CHOLESTEROL LEVELS:**

CODE:



```
d['age_group'] = pd.cut(d['age'], bins=[29, 39, 49, 59, 69, 79], labels=['30-39', '40-49', '50-59', '60-69', '70-79'])

# Create cholesterol groups
d['cholesterol_group'] = pd.cut(d['chol'], bins=[0, 200, 240, 300], labels=['Normal', 'Borderline High', 'High'])

# Create heatmap data
heatmap_data = d.groupby(['age_group', 'cholesterol_group', 'target']).size().unstack().fillna(0)

# Plotting the heatmap
plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data, annot=True, fmt='.0f', cmap='RdBu', cbar_kws={'label': 'Number of Patients'})
plt.title('Heatmap of Heart Disease Status Across Age and Cholesterol Levels')
plt.xlabel('Cholesterol Levels')
plt.ylabel('Age Group')
plt.show()
```

OUTPUT:



4. Conclusions and Summary:

a) SCATTER PLOT FOR AGE VS CHOLESTEROL AND HOW RELATED IT IS TO PRESENCE OF HEART DISEASE

The scatter plot elucidates the correlation between age and cholesterol levels while highlighting the presence of heart disease among the participants. **It shows that people having cholesterol in range 200-300mg/dL have more chance of having a heart disease**.

b) HISTOGRAM TO DEPICT DISTRIBUTION OF CHOLESTROL LEVELS:

The histogram reveals the distribution of cholesterol levels within the patient cohort, facilitating an understanding of cholesterol prevalence. **It shows that more people have cholesterol in range 200-300mg/dL.**

c) BAR PLOT TO SHOW RELATION BETWEEN AVERAGE CHOLESTROL LEVEL AND THE OCCURRENCE OF HEART DISEASE:

The bar plot demonstrates the average cholesterol levels categorized by heart disease status, providing insights into the impact of heart disease on cholesterol. **It shows that cholesterol level doesn’t have significant impact on occurrence of heart disease.**

d) BOX PLOT FOR CHECKING RELATION BETWEEN RESTING BLOOD PRESSURE AND OCCURANCE HEART DISEASE :

The box plot showcases the variability in resting blood pressure across different heart disease statuses, enabling comparisons between the two groups. **It shows that high blood pressure doesn’t have direct correlation to occurrence of heart disease.**

e) VIOLIN PLOT FOR AGE VS. RESTING BLOOD PRESSURE:

The violin plot offers a comprehensive view of the distribution of resting blood pressure against heart disease status, accentuating any differences in distribution.

f) PIE CHARTS TO DEPICT PERCENTAGE OF PEOPLE OF HEART DISEASE IN MALE AND FEMALE:

The pie charts illustrating the percentage of heart disease among males and females revealed a striking difference in the distribution of heart disease status. **It shows more percentage of women have heart disease as compared to men. Thus we can conclude that women have higher chance of having a heart disease that men.**

g) SUB BAR PLOTS TO SEE RELATION BETWEEN EACH TYPES OF CHEST PAIN AND THE CHANCES OF THEM HAVING A HEART DISEASE:

The sub bar plots depicting the relationship between various types of chest pain (categorically labeled from 0 to 3) and the corresponding chances of heart disease revealed critical insights. Each plot showcased the number of individuals with and without heart disease for each type of chest pain. **Notably, individuals experiencing typical angina (chest pain type 1) exhibited the highest frequency of heart disease, whereas those with atypical angina (chest pain type 2) showed a relatively lower incidence.**

h) DEPICTING FREQUENCY OF MALES AND FEMALES HAVING DIFFERENT THALASSEMIA TYPES USING BAR SUBPLOTS:

The bar subplots depicting the frequency of thalassemia types among males and females highlighted distinct patterns in the prevalence of this condition**. Both male and female plots illustrated that the majority of patients exhibited normal thalassemia**, with fewer individuals diagnosed with fixed or reversible defects. **However, the male subplot showed a slightly higher occurrence of fixed defects compared to females.**

i) HEATMAP OF HEART DISEASE STATUS ACROSS AGE AND CHOLESTEROL LEVELS

The heatmap illustrated the relationship between age groups and cholesterol levels in relation to heart disease prevalence. Key observations include:

- **Increased Risk with Age:** Older age groups (60-69 and 70-79) showed a higher incidence of heart disease.
- **Cholesterol Impact:** High cholesterol levels were consistently linked to increased heart disease rates, indicating that even borderline high levels can pose risks.
- **Combined Analysis:** The data emphasized that older individuals with high cholesterol face significant heart disease risks, highlighting the need for regular health monitoring.

5. Learning Outcomes:

❖ **Data Manipulation Skills:** Learnt to utilize libraries such as NumPy and Pandas for efficient data cleaning, manipulation, and transformation, enhancing the ability to handle real-world health datasets.

- **Visualization Techniques:** Gained proficiency in using Matplotlib and Seaborn to create various visualizations, including scatter plots, bar charts, and heatmaps, to effectively communicate insights derived from health data.

- **Statistical Analysis Understanding:** Developed an understanding of statistical concepts by analyzing correlations between health metrics (e.g., cholesterol levels, age) and the presence of heart disease, fostering critical thinking skills in interpreting data.

- **Interpretation of Medical Data:** Enhanced the ability to interpret medical data through visual representations, which aids in identifying trends and patterns that are crucial for health risk assessments and decision-making.

- **Application of Data Analysis in Healthcare:** Recognized the significance of data analysis in healthcare settings, learning how data-driven insights can influence patient management strategies and promote proactive healthcare measures.