Table of Contents

Project Introduction

The chosen dataset is taken from kaggle/datasets and is related to the **heart disease analysis of a patient**. The **aim** of this project is to make different analysis on the heart disease dataset to explore the analysis of patients. There is an attribute '**target**' which shows the presence of heart disease in a patient. Value **1** shows the disease, and the Value **0** shows no disease. The dataset holds **1025 rows** and **14 columns** initially. We will perform different operations on this dataset to make an analysis of this dataset.

This dataset was created by:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach, and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

Following is the information regarding the columns in a dataset:

1. age: patient age.
2. sex: 0 is for female and 1 for female.
3. cp: It is the type of chest pain of a patient. It has 4 types of pains.
    a. Value 0: asymptomatic
    b. Value 1: atypical angina
    c. Value 2: non-anginal pain
    d. Value 3: typical angina
4. trestbps: It is the resting of patient blood pressure.
5. chol: Showing the serum cholesterol in mg/dl.
6. fbs: It is the blood sugar fasting > 120 mg/dl.
7. restecg: Shows resting electrocardiographic results. It has values 0, 1, and 2.
8. thalach: This is the maximum heart rate of a patient.
9. exang: This attribute is for exercise induced angina.
10. oldpeak: It is the ST depression induced relative to rest by exercise.
11. slope: It is the slope of the peak exercise ST segment.
12. ca: This shows the number of major vessels covered by fluoroscopy. It is numbered from 0-3.
13. thal: It is the thalassemia value where 0 shows the normal, 1 shows the fixed-defect, and 2 shows the reversable defect.
14. target: It is the disease status of patient. 0 shows no disease and 1 shows the disease.

Analysis Objectives

To analyze and visualize the data of chosen dataset, following are the questions we have defined:

1. Describe the attributes of chosen dataset using the statistical methods showing the statistical information, and the complete statistical chart for numerical attributes.
2. What percentage of patients have a heart disease in the age less than 45? Show their details.
3. Which type of chest pain is reported by the patients which have fixed defect in their heart? Is it all asymptomatic?

4. What are the average cholesterol values for both the males and females, which have diabetes, and which do not have diabetes?
5. Sort the dataset from older to younger patient and display it.
6. Show the maximum heart rate in ascending order of the females which have heart-disease.
7. Calculate the person’s average resting blood pressure against each resting electrocardiographic result group in ascending order.
8. Display the number of males and number of females with heart disease, and without heart disease.
9. Display the percentage of each type of chest pain of the patients who have heart disease using pie chart. Which type of pain is the most? Highlight that.
10. Display the cholesterol level range for the patients above age 50 of both with the heart disease, and without heart disease using scatter plot. Also display the average cholesterol level.

Data Acquisition and Cleaning

Python code for reading the dataset

```
# Importing necessary library
import pandas as pd

# Reading the chosen dataset
heart = pd.read_csv('./Data/heart.csv')
# Displaying the dataset as a dataframe
heart
```

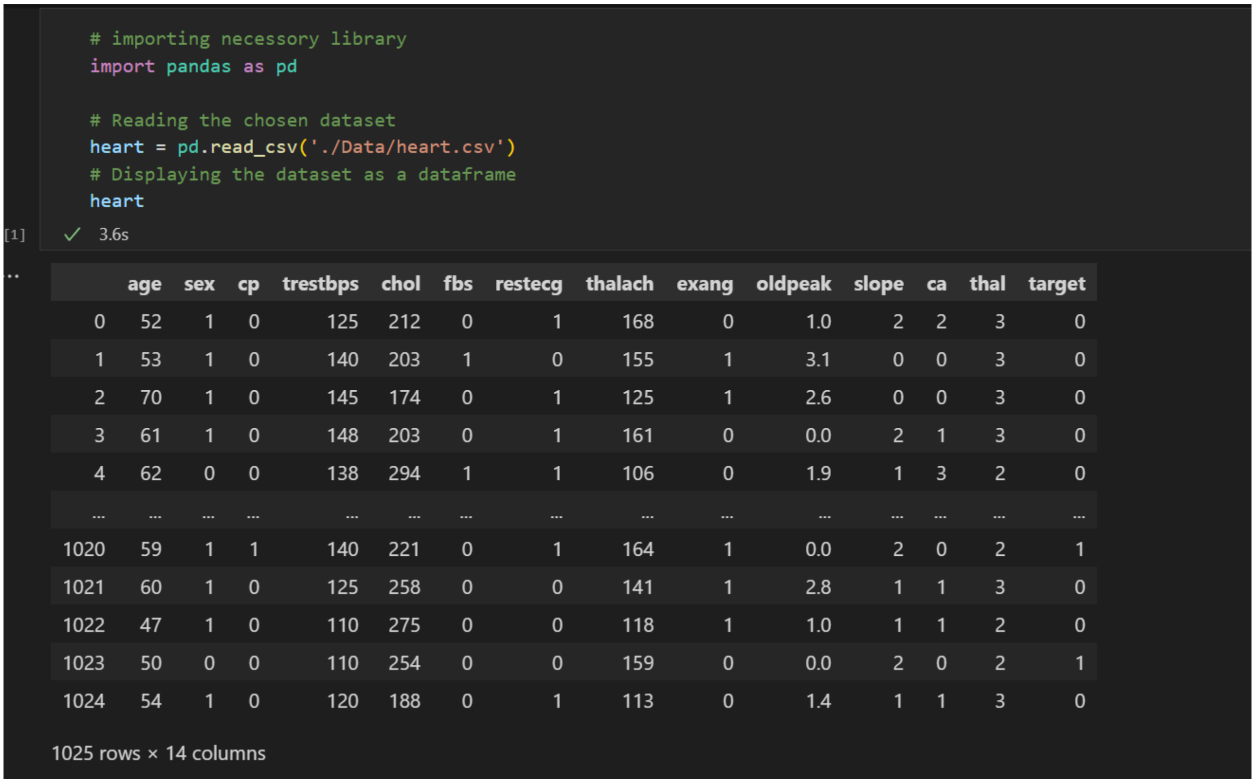
Python Code for cleaning the dataset

```
# Cleaning the dataset
heart.drop_duplicates(inplace=True, ignore_index=True)

# Why Cleaning unnecessary data
print("Why Cleaning needed: There were 1025 records in the dataset initially. But the patients were not 1025 as dataset contained the data of one patient more than once. The cleaning of data was needed to manipulate the dataset which contain the details of different patients. We don't want to analyze single patient more than once that's why we do not need any duplicate values.")

# Displaying the cleaned dataset
heart
```

Why Cleaning is needed for selected Dataset

There were 1025 records in the dataset initially. But the patients were not 1025 as dataset contained the data of one patient more than once. The cleaning of data was needed to manipulate the dataset which contain the details of different patients. We don't want to analyze single patient more than once that's why we do not need any duplicate values.

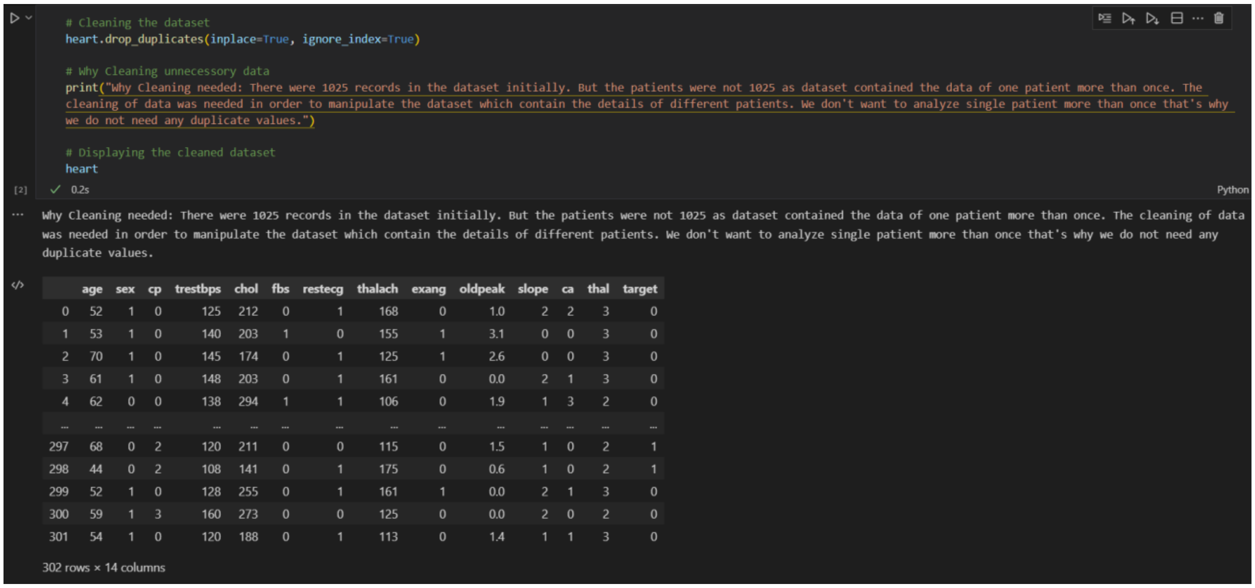
Screenshot of code



```python
# importing necessory library
import pandas as pd

# Reading the chosen dataset
heart = pd.read_csv('./Data/heart.csv')
# Displaying the dataset as a dataframe
heart
```

	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



```python
# Cleaning the dataset
heart.drop_duplicates(inplace=True, ignore_index=True)

# Why Cleaning unnecessory data
print("Why Cleaning needed: There were 1025 records in the dataset initially. But the patients were not 1025 as dataset contained the data of one patient more than once. The cleaning of data was needed in order to manipulate the dataset which contain the details of different patients. We don't want to analyze single patient more than once that's why we do not need any duplicate values.")

# Displaying the cleaned dataset
heart
```

```
Why Cleaning needed: There were 1025 records in the dataset initially. But the patients were not 1025 as dataset contained the data of one patient more than once. The cleaning of data
was needed in order to manipulate the dataset which contain the details of different patients. We don't want to analyze single patient more than once that's why we do not need any
duplicate values.
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
297	68	0	2	120	211	0	0	115	0	1.5	1	0	2	1
298	44	0	2	108	141	0	1	175	0	0.6	1	0	2	1
299	52	1	0	128	255	0	1	161	1	0.0	2	1	3	0
300	59	1	3	160	273	0	0	125	0	0.0	2	0	2	0
301	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

302 rows × 14 columns

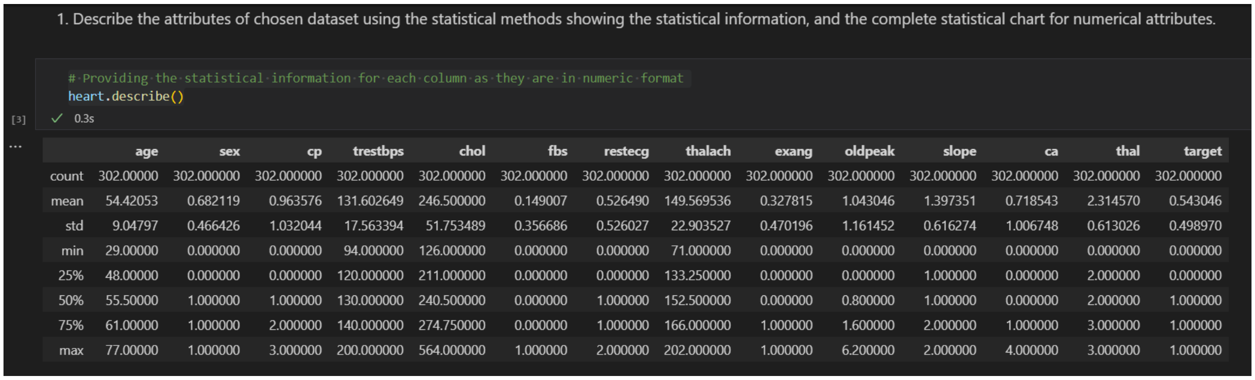
Data and Exploratory Analysis

Code for Question 1

```python
# Providing the statistical information for each column as they are in numeric
format
heart.describe()
```

```
# Providing the information about each attribute in a dataset
heart.info()
```

Screenshot for Question 1

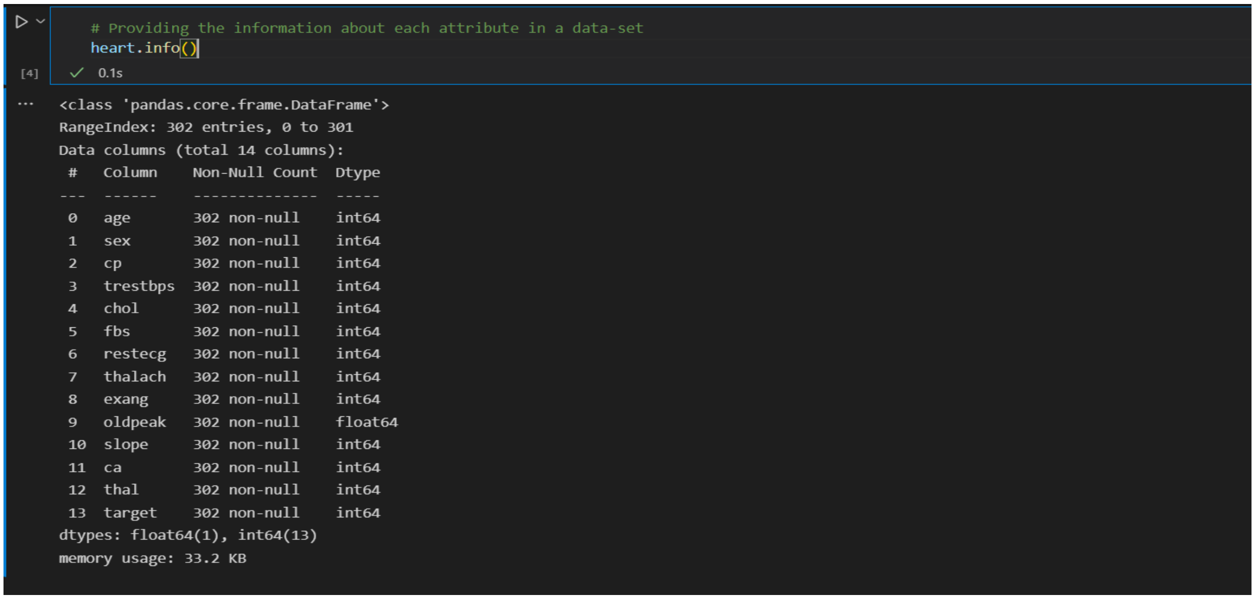


1. Describe the attributes of chosen dataset using the statistical methods showing the statistical information, and the complete statistical chart for numerical attributes.

```
# Providing the statistical information for each column as they are in numeric format
heart.describe()
```

[3] ✓ 0.3s

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	302.00000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000
mean	54.42053	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.569536	0.327815	1.043046	1.397351	0.718543	2.314570	0.543046
std	9.04797	0.466426	1.032044	17.563394	51.753489	0.356686	0.526027	22.903527	0.470196	1.161452	0.616274	1.006748	0.613026	0.498970
min	29.00000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.00000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.250000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.50000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.500000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.00000	1.000000	2.000000	140.000000	274.750000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.00000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000



```
# Providing the information about each attribute in a data-set
heart.info()
```

[4] ✓ 0.1s

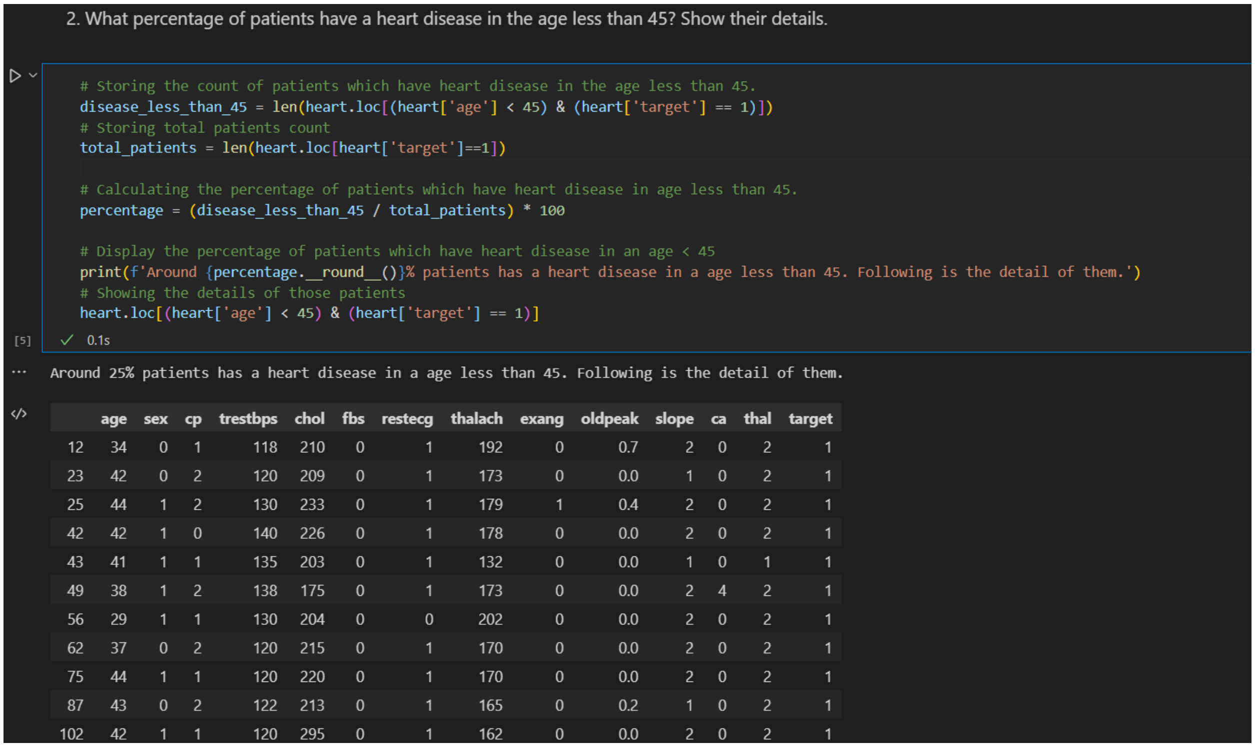
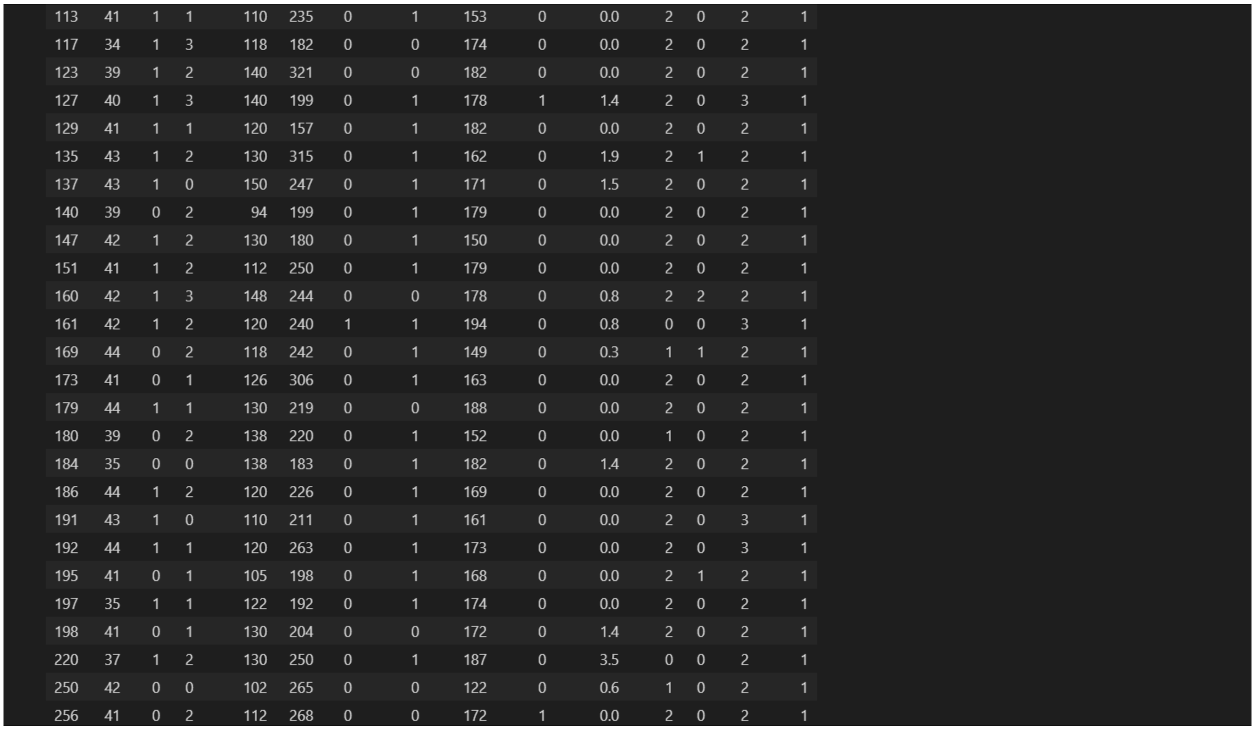
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 302 entries, 0 to 301
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       302 non-null    int64  
 1   sex       302 non-null    int64  
 2   cp        302 non-null    int64  
 3   trestbps  302 non-null    int64  
 4   chol      302 non-null    int64  
 5   fbs       302 non-null    int64  
 6   restecg   302 non-null    int64  
 7   thalach   302 non-null    int64  
 8   exang     302 non-null    int64  
 9   oldpeak   302 non-null    float64
 10  slope     302 non-null    int64  
 11  ca        302 non-null    int64  
 12  thal      302 non-null    int64  
 13  target    302 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

Code for Question 2

```
# Storing the count of patients which have heart disease in the age less than 45.
disease_less_than_45 = len(heart.loc[(heart['age'] < 45) & (heart['target'] ==
1)])
# Storing total patients count
total_patients = len(heart.loc[heart['target']==1])

# Calculating the percentage of patients which have heart disease in age less
than 45.
percentage = (disease_less_than_45 / total_patients) * 100

# Display the percentage of patients which have heart disease in an age < 45
print(f'Around {percentage.__round__()}% patients has a heart disease in a age
less than 45. Following is the detail of them.')
```

```
# Showing the details of those patients
heart.loc[(heart['age'] < 45) & (heart['target'] == 1)]
```

Screenshot for Question 2

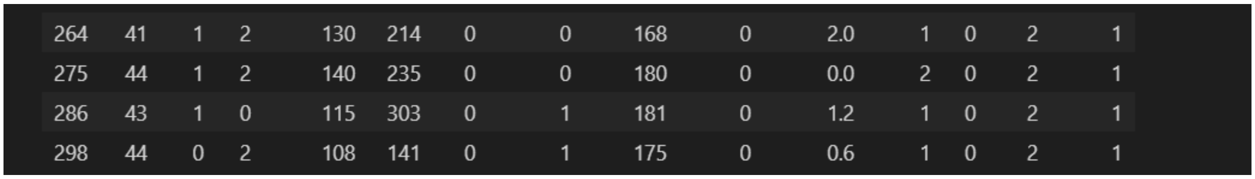
2. What percentage of patients have a heart disease in the age less than 45? Show their details.

```
# Storing the count of patients which have heart disease in the age less than 45.
disease_less_than_45 = len(heart.loc[(heart['age'] < 45) & (heart['target'] == 1)])
# Storing total patients count
total_patients = len(heart.loc[heart['target']==1])

# Calculating the percentage of patients which have heart disease in age less than 45.
percentage = (disease_less_than_45 / total_patients) * 100

# Display the percentage of patients which have heart disease in an age < 45
print(f'Around {percentage.__round__()}% patients has a heart disease in a age less than 45. Following is the detail of them.')
# Showing the details of those patients
heart.loc[(heart['age'] < 45) & (heart['target'] == 1)]
```

[5] ✓ 0.1s

```
Around 25% patients has a heart disease in a age less than 45. Following is the detail of them.
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
12	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
23	42	0	2	120	209	0	1	173	0	0.0	1	0	2	1
25	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	42	1	0	140	226	0	1	178	0	0.0	2	0	2	1
43	41	1	1	135	203	0	1	132	0	0.0	1	0	1	1
49	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1
56	29	1	1	130	204	0	0	202	0	0.0	2	0	2	1
62	37	0	2	120	215	0	1	170	0	0.0	2	0	2	1
75	44	1	1	120	220	0	1	170	0	0.0	2	0	2	1
87	43	0	2	122	213	0	1	165	0	0.2	1	0	2	1
102	42	1	1	120	295	0	1	162	0	0.0	2	0	2	1
113	41	1	1	110	235	0	1	153	0	0.0	2	0	2	1
117	34	1	3	118	182	0	0	174	0	0.0	2	0	2	1
123	39	1	2	140	321	0	0	182	0	0.0	2	0	2	1
127	40	1	3	140	199	0	1	178	1	1.4	2	0	3	1
129	41	1	1	120	157	0	1	182	0	0.0	2	0	2	1
135	43	1	2	130	315	0	1	162	0	1.9	2	1	2	1
137	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
140	39	0	2	94	199	0	1	179	0	0.0	2	0	2	1
147	42	1	2	130	180	0	1	150	0	0.0	2	0	2	1
151	41	1	2	112	250	0	1	179	0	0.0	2	0	2	1
160	42	1	3	148	244	0	0	178	0	0.8	2	2	2	1
161	42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
169	44	0	2	118	242	0	1	149	0	0.3	1	1	2	1
173	41	0	1	126	306	0	1	163	0	0.0	2	0	2	1
179	44	1	1	130	219	0	0	188	0	0.0	2	0	2	1
180	39	0	2	138	220	0	1	152	0	0.0	1	0	2	1
184	35	0	0	138	183	0	1	182	0	1.4	2	0	2	1
186	44	1	2	120	226	0	1	169	0	0.0	2	0	2	1
191	43	1	0	110	211	0	1	161	0	0.0	2	0	3	1
192	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
195	41	0	1	105	198	0	1	168	0	0.0	2	1	2	1
197	35	1	1	122	192	0	1	174	0	0.0	2	0	2	1
198	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
220	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
250	42	0	0	102	265	0	0	122	0	0.6	1	0	2	1
256	41	0	2	112	268	0	0	172	1	0.0	2	0	2	1

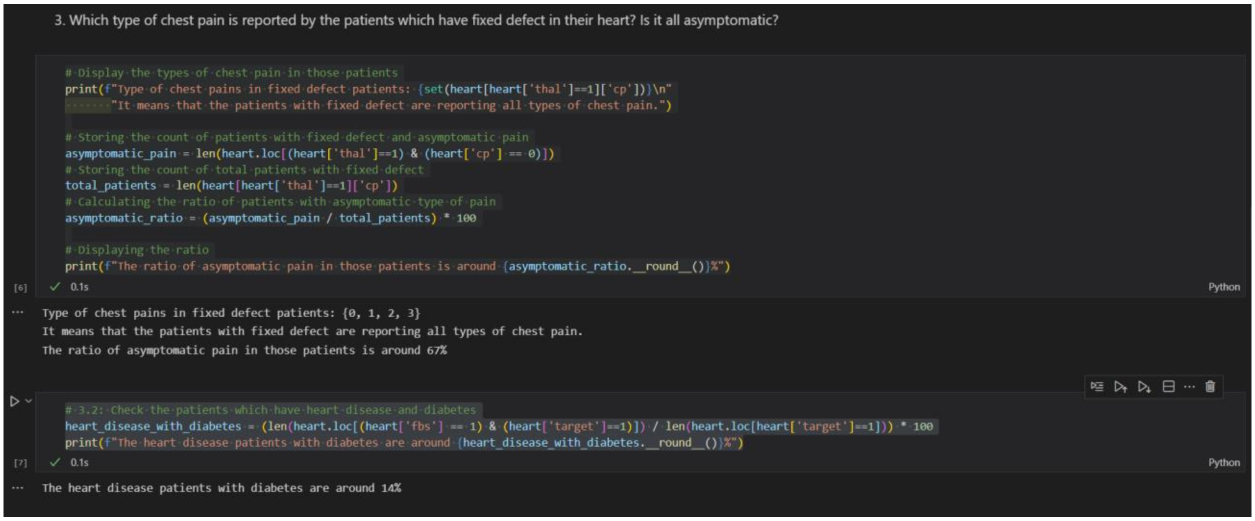


264	41	1	2	130	214	0	0	168	0	2.0	1	0	2	1
275	44	1	2	140	235	0	0	180	0	0.0	2	0	2	1
286	43	1	0	115	303	0	1	181	0	1.2	1	0	2	1
298	44	0	2	108	141	0	1	175	0	0.6	1	0	2	1

Code for Question 3

```
# Display the types of chest pain in those patients
print(f"Type of chest pains in fixed defect patients:
{set(heart[heart['thal']==1]['cp'])}\n"
       "It means that the patients with fixed defect are reporting all types of
chest pain.")

# Storing the count of patients with fixed defect and asymptomatic pain
asymptomatic_pain = len(heart.loc[(heart['thal']==1) & (heart['cp'] == 0)])
# Storing the count of total patients with fixed defect
total_patients = len(heart[heart['thal']==1]['cp'])
# Calculating the ratio of patients with asymptomatic type of pain
asymptomatic_ratio = (asymptomatic_pain / total_patients) * 100

# Displaying the ratio
print(f"The ratio of asymptomatic pain in those patients is around
{asymptomatic_ratio.__round__()}%")
# 3.2: Check the patients which have heart disease and diabetes
heart_disease_with_diabetes = (len(heart.loc[(heart['fbs'] == 1) &
(heart['target']==1)]) / len(heart.loc[heart['target']==1])) * 100
print(f"The heart disease patients with diabetes are around
{heart_disease_with_diabetes.__round__()}%")
```

Screenshot for Question 3



3. Which type of chest pain is reported by the patients which have fixed defect in their heart? Is it all asymptomatic?

```
# Display the types of chest pain in those patients
print(f"Type of chest pains in fixed defect patients: {set(heart[heart['thal']==1]['cp'])}\n"
      "It means that the patients with fixed defect are reporting all types of chest pain.")

# Storing the count of patients with fixed defect and asymptomatic pain
asymptomatic_pain = len(heart.loc[(heart['thal']==1) & (heart['cp'] == 0)])
# Storing the count of total patients with fixed defect
total_patients = len(heart[heart['thal']==1]['cp'])
# Calculating the ratio of patients with asymptomatic type of pain
asymptomatic_ratio = (asymptomatic_pain / total_patients) * 100

# Displaying the ratio
print(f"The ratio of asymptomatic pain in those patients is around {asymptomatic_ratio.__round__()}%")
```

```
Type of chest pains in fixed defect patients: {0, 1, 2, 3}
It means that the patients with fixed defect are reporting all types of chest pain.
The ratio of asymptomatic pain in those patients is around 67%
```

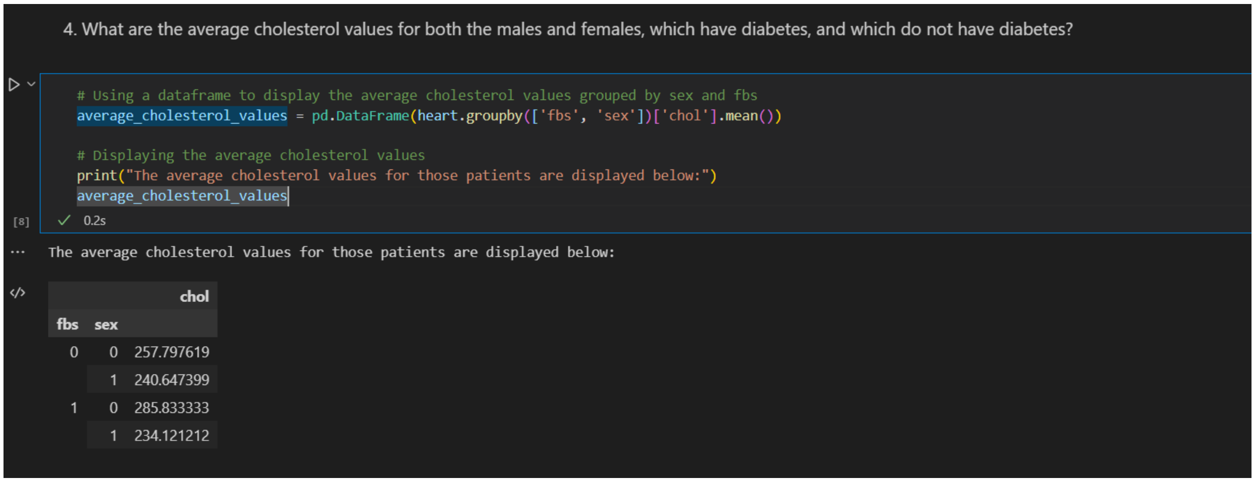
```
# 3.2: Check the patients which have heart disease and diabetes
heart_disease_with_diabetes = (len(heart.loc[(heart['fbs'] == 1) & (heart['target']==1)]) / len(heart.loc[heart['target']==1])) * 100
print(f"The heart disease patients with diabetes are around {heart_disease_with_diabetes.__round__()}%")
```

```
The heart disease patients with diabetes are around 14%
```

Code for Question 4

```
# Using a dataframe to display the average cholesterol values grouped by sex and fbs
average_cholesterol_values = pd.DataFrame(heart.groupby(['fbs', 'sex'])['chol'].mean())

# Displaying the average cholesterol values
print("The average cholesterol values for those patients are displayed below:")
average_cholesterol_values
```

Screenshot for Question 4



4. What are the average cholesterol values for both the males and females, which have diabetes, and which do not have diabetes?

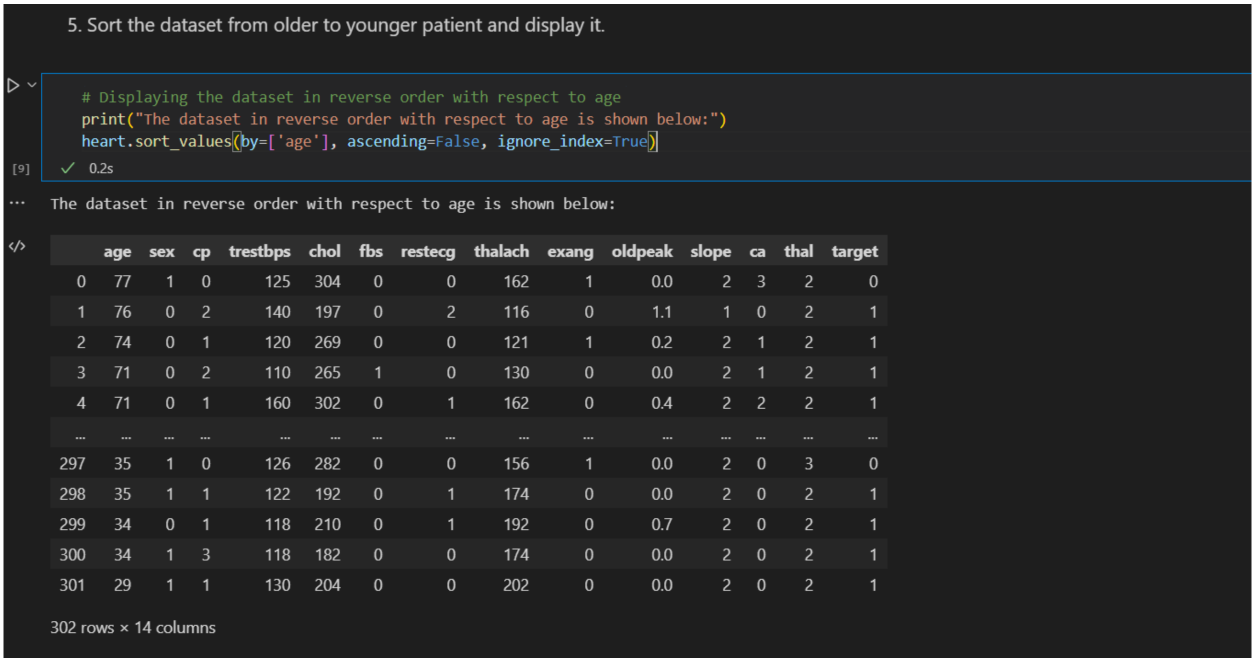
```
# Using a dataframe to display the average cholesterol values grouped by sex and fbs
average_cholesterol_values = pd.DataFrame(heart.groupby(['fbs', 'sex'])['chol'].mean())

# Displaying the average cholesterol values
print("The average cholesterol values for those patients are displayed below:")
average_cholesterol_values
```

```
The average cholesterol values for those patients are displayed below:

              chol
fbs  sex
  0    0  257.797619
       1  240.647399
  1    0  285.833333
       1  234.121212
```

Code for Question 5

```
# Displaying the dataset in reverse order with respect to age
print("The dataset in reverse order with respect to age is shown below:")
heart.sort_values(by=['age'], ascending=False, ignore_index=True)
```

Screenshot for Question 5



5. Sort the dataset from older to younger patient and display it.

```
# Displaying the dataset in reverse order with respect to age
print("The dataset in reverse order with respect to age is shown below:")
heart.sort_values(by=['age'], ascending=False, ignore_index=True)
```

[9] ✓ 0.2s

```
The dataset in reverse order with respect to age is shown below:
```

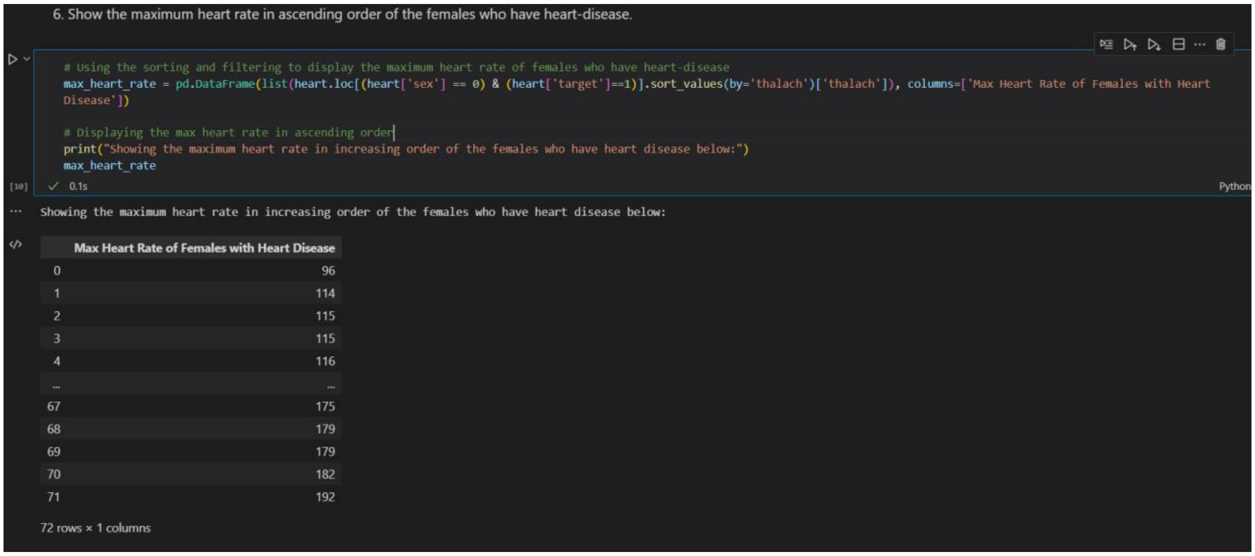
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	77	1	0	125	304	0	0	162	1	0.0	2	3	2	0
1	76	0	2	140	197	0	2	116	0	1.1	1	0	2	1
2	74	0	1	120	269	0	0	121	1	0.2	2	1	2	1
3	71	0	2	110	265	1	0	130	0	0.0	2	1	2	1
4	71	0	1	160	302	0	1	162	0	0.4	2	2	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
297	35	1	0	126	282	0	0	156	1	0.0	2	0	3	0
298	35	1	1	122	192	0	1	174	0	0.0	2	0	2	1
299	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
300	34	1	3	118	182	0	0	174	0	0.0	2	0	2	1
301	29	1	1	130	204	0	0	202	0	0.0	2	0	2	1

302 rows × 14 columns

Code for Question 6

```
# Using the sorting and filtering to display the maximum heart rate of females who have heart-disease
max_heart_rate = pd.DataFrame(list(heart.loc[(heart['sex'] == 0) & (heart['target']==1)].sort_values(by='thalach')['thalach']), columns=['Max Heart Rate of Females with Heart Disease'])

# Displaying the max heart rate in ascending order
print("Showing the maximum heart rate in increasing order of the females who have heart disease below:")
max_heart_rate
```

Screenshot for Question 6



6. Show the maximum heart rate in ascending order of the females who have heart-disease.

```python
# Using the sorting and filtering to display the maximum heart rate of females who have heart-disease
max_heart_rate = pd.DataFrame(list(heart.loc[(heart['sex'] == 0) & (heart['target']==1)].sort_values(by='thalach')['thalach']), columns=['Max Heart Rate of Females with Heart Disease'])

# Displaying the max heart rate in ascending order
print("Showing the maximum heart rate in increasing order of the females who have heart disease below:")
max_heart_rate
```

[10] ✓ 0.1s Python

```
Showing the maximum heart rate in increasing order of the females who have heart disease below:
```

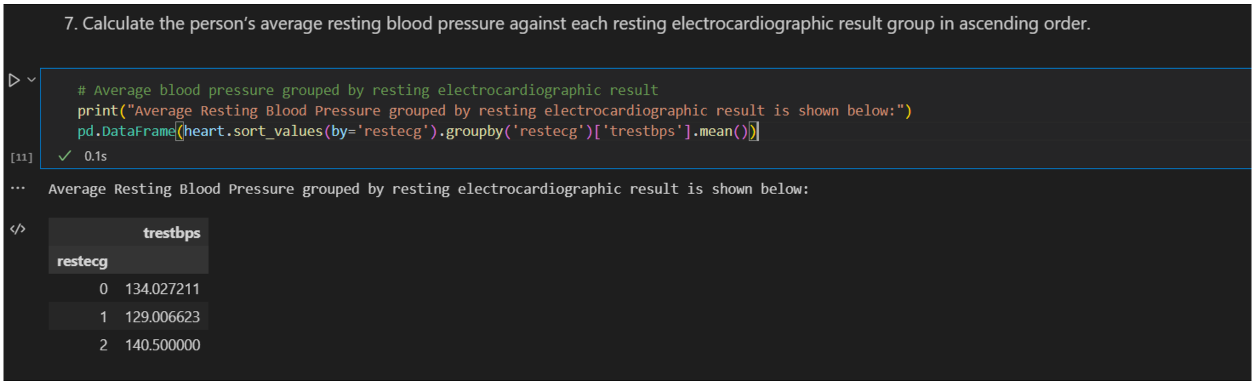
	Max Heart Rate of Females with Heart Disease
0	96
1	114
2	115
3	115
4	116
...	...
67	175
68	179
69	179
70	182
71	192

72 rows × 1 columns

Code for Question 7

```python
# Average blood pressure grouped by resting electrocardiographic result
print("Average Resting Blood Pressure grouped by resting electrocardiographic result is shown below:")
pd.DataFrame(heart.sort_values(by='restecg').groupby('restecg')['trestbps'].mean())
```

Screenshot for Question 7



7. Calculate the person's average resting blood pressure against each resting electrocardiographic result group in ascending order.

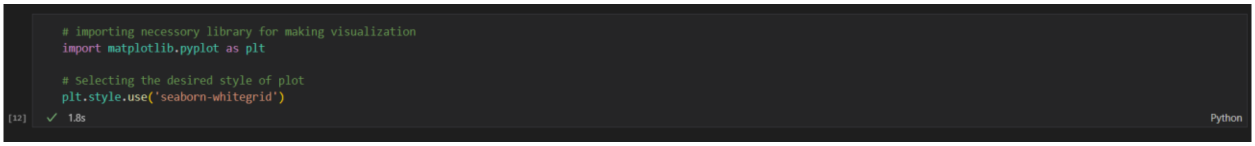
```python
# Average blood pressure grouped by resting electrocardiographic result
print("Average Resting Blood Pressure grouped by resting electrocardiographic result is shown below:")
pd.DataFrame(heart.sort_values(by='restecg').groupby('restecg')['trestbps'].mean())
```

[11] ✓ 0.1s

```
Average Resting Blood Pressure grouped by resting electrocardiographic result is shown below:
```

	trestbps
restecg	
0	134.027211
1	129.006623
2	140.500000

Data Visualization



```python
# importing necessory library for making visualization
import matplotlib.pyplot as plt

# Selecting the desired style of plot
plt.style.use('seaborn-whitegrid')
```

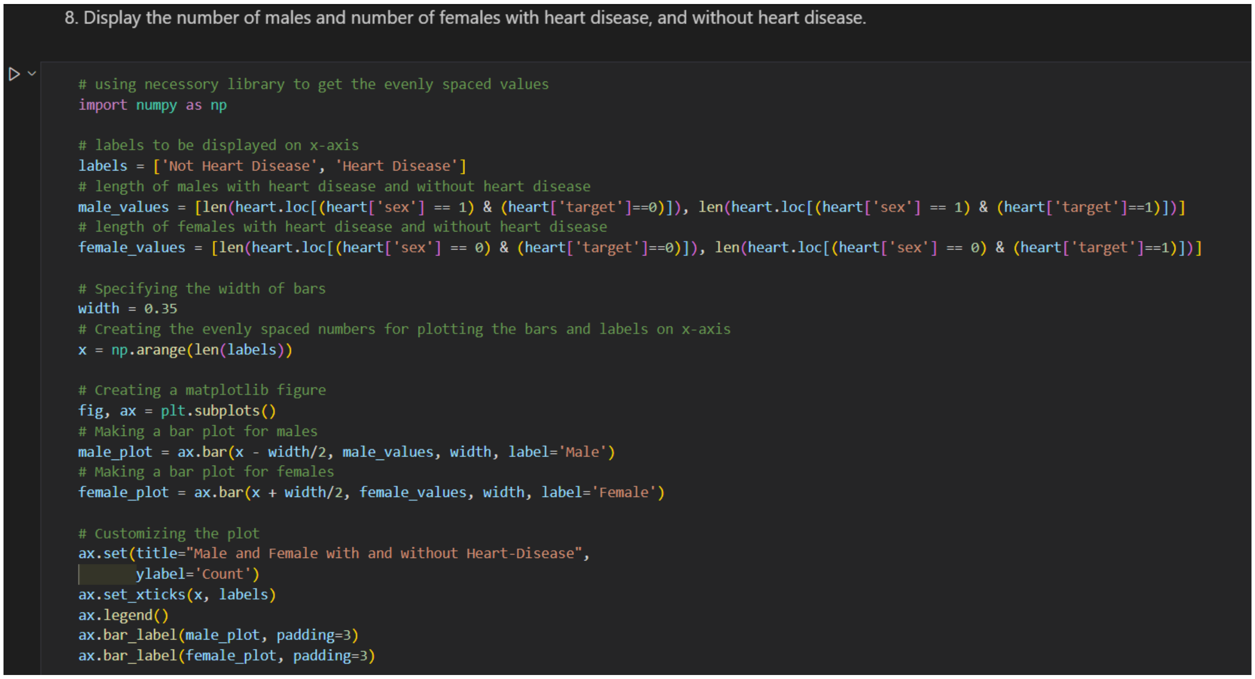
[12] ✓ 1.8s Python

Code for Question 8

```python
# using necessory library to get the evenly spaced values
import numpy as np
```

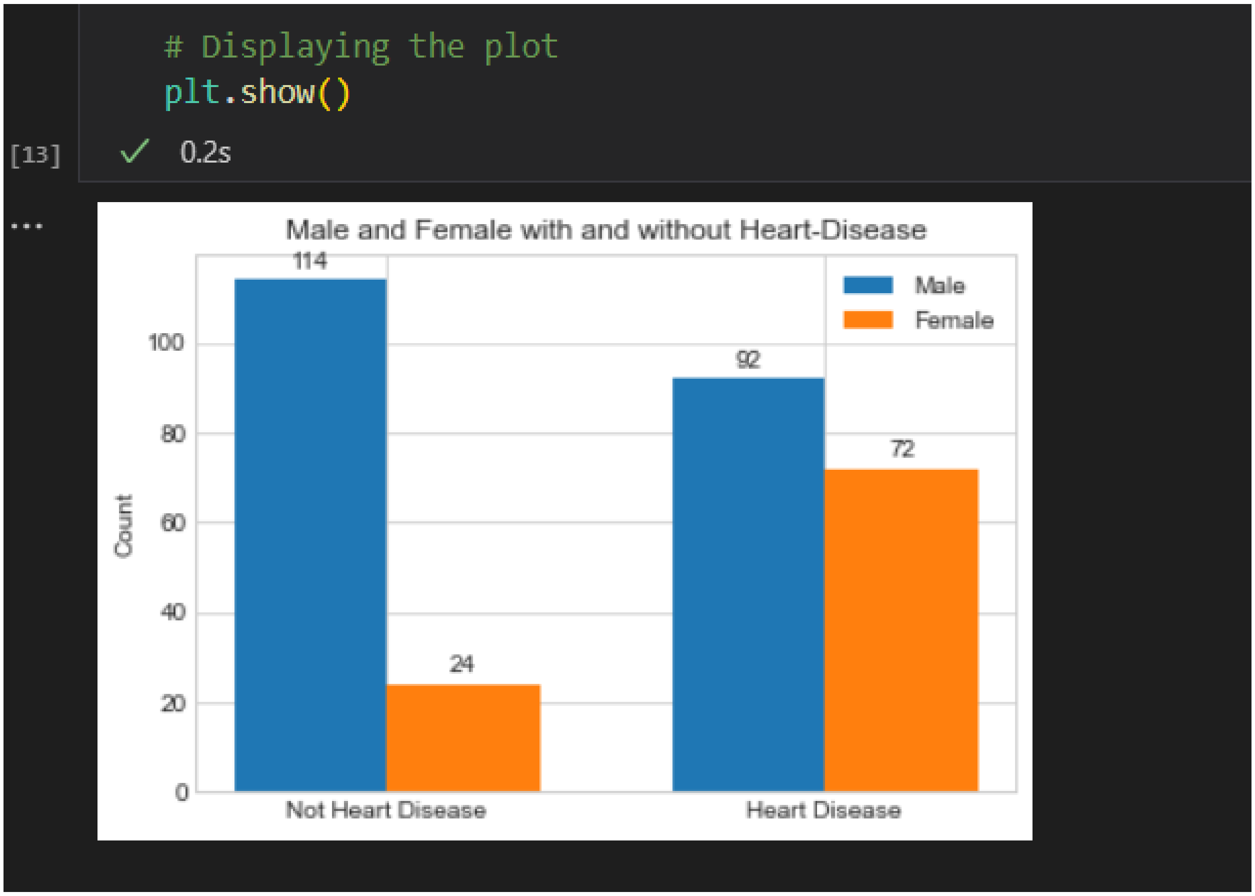
```python
# labels to be displayed on x-axis
labels = ['Not Heart Disease', 'Heart Disease']
# length of males with heart disease and without heart disease
male_values = [len(heart.loc[(heart['sex'] == 1) & (heart['target']==0)]),
len(heart.loc[(heart['sex'] == 1) & (heart['target']==1)])]
# length of females with heart disease and without heart disease
female_values = [len(heart.loc[(heart['sex'] == 0) & (heart['target']==0)]),
len(heart.loc[(heart['sex'] == 0) & (heart['target']==1)])]

# Specifying the width of bars
width = 0.35
# Creating the evenly spaced numbers for plotting the bars and labels on x-axis
x = np.arange(len(labels))

# Creating a matplotlib figure
fig, ax = plt.subplots()
# Making a bar plot for males
male_plot = ax.bar(x - width/2, male_values, width, label='Male')
# Making a bar plot for females
female_plot = ax.bar(x + width/2, female_values, width, label='Female')

# Customizing the plot
ax.set(title="Male and Female with and without Heart-Disease",
       ylabel='Count')
ax.set_xticks(x, labels)
ax.legend()
ax.bar_label(male_plot, padding=3)
ax.bar_label(female_plot, padding=3)

# Displaying the plot
plt.show()
```

Screenshot for Question 8



8. Display the number of males and number of females with heart disease, and without heart disease.

```
# using necessary library to get the evenly spaced values
import numpy as np

# labels to be displayed on x-axis
labels = ['Not Heart Disease', 'Heart Disease']
# length of males with heart disease and without heart disease
male_values = [len(heart.loc[(heart['sex'] == 1) & (heart['target']==0)]), len(heart.loc[(heart['sex'] == 1) & (heart['target']==1)])]
# length of females with heart disease and without heart disease
female_values = [len(heart.loc[(heart['sex'] == 0) & (heart['target']==0)]), len(heart.loc[(heart['sex'] == 0) & (heart['target']==1)])]

# Specifying the width of bars
width = 0.35
# Creating the evenly spaced numbers for plotting the bars and labels on x-axis
x = np.arange(len(labels))

# Creating a matplotlib figure
fig, ax = plt.subplots()
# Making a bar plot for males
male_plot = ax.bar(x - width/2, male_values, width, label='Male')
# Making a bar plot for females
female_plot = ax.bar(x + width/2, female_values, width, label='Female')

# Customizing the plot
ax.set(title="Male and Female with and without Heart-Disease",
       ylabel='Count')
ax.set_xticks(x, labels)
ax.legend()
ax.bar_label(male_plot, padding=3)
ax.bar_label(female_plot, padding=3)
```



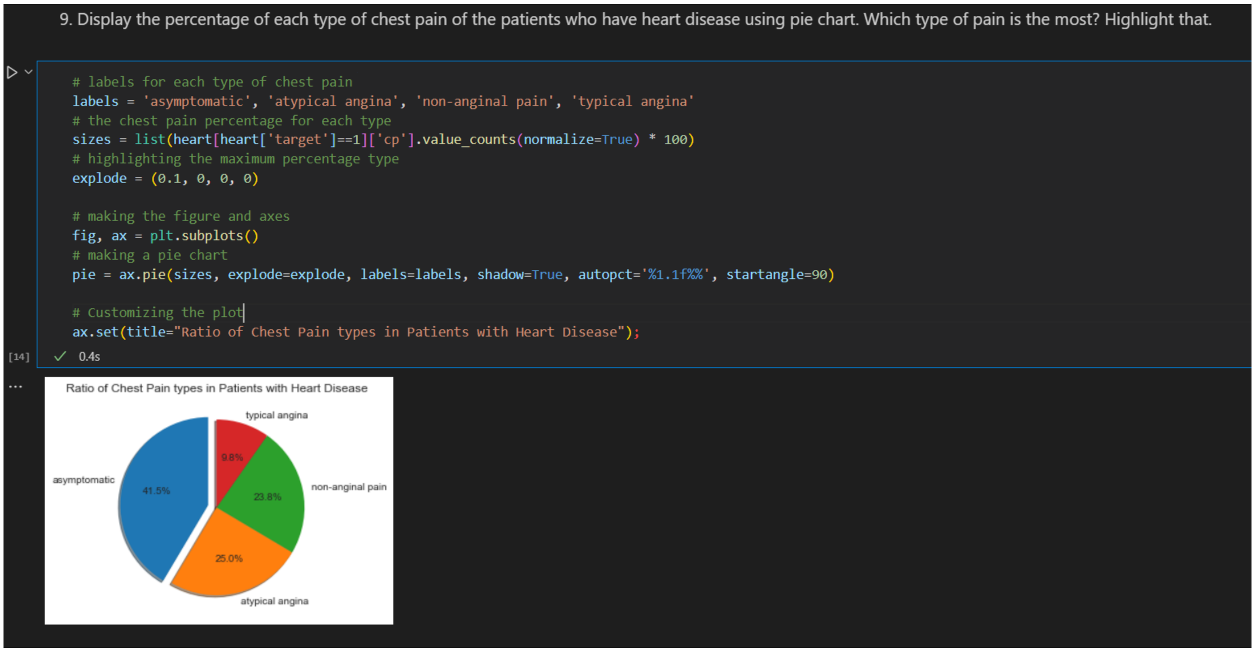
```
# Displaying the plot
plt.show()
```

[13] ✓ 0.2s

Code for Question 9

```
# labels for each type of chest pain
labels = 'asymptomatic', 'atypical angina', 'non-anginal pain', 'typical angina'
# the chest pain percentage for each type
sizes = list(heart[heart['target']==1]['cp'].value_counts(normalize=True) * 100)
# highlighting the maximum percentage type
explode = (0.1, 0, 0, 0)

# making the figure and axes
fig, ax = plt.subplots()
# making a pie chart
pie = ax.pie(sizes, explode=explode, labels=labels, shadow=True,
autopct='%1.1f%%', startangle=90)

# Customizing the plot
ax.set(title="Ratio of Chest Pain types in Patients with Heart Disease");
```

Screenshots for Question 9



9. Display the percentage of each type of chest pain of the patients who have heart disease using pie chart. Which type of pain is the most? Highlight that.

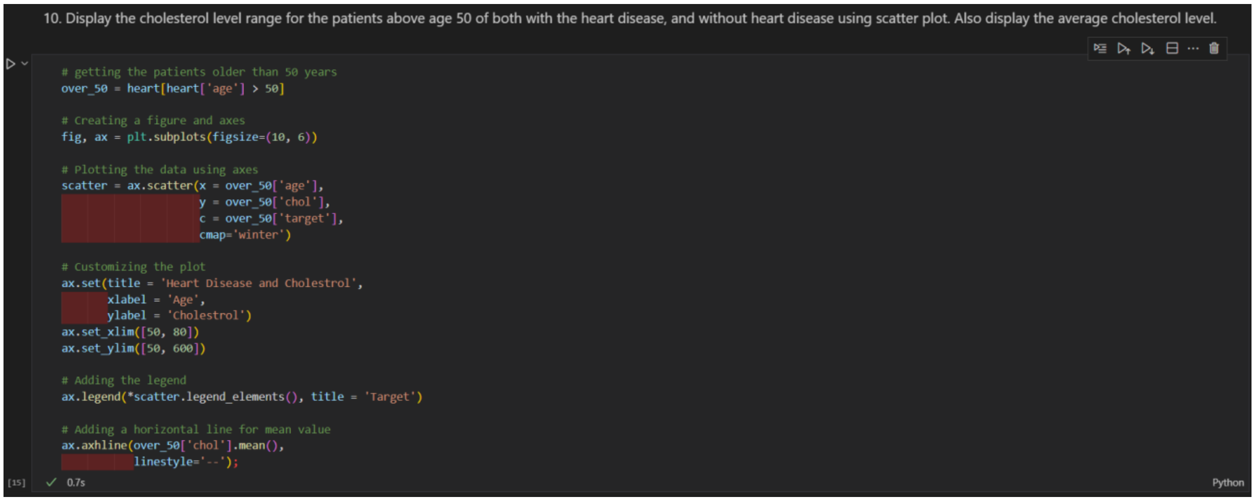
```
# labels for each type of chest pain
labels = 'asymptomatic', 'atypical angina', 'non-anginal pain', 'typical angina'
# the chest pain percentage for each type
sizes = list(heart[heart['target']==1]['cp'].value_counts(normalize=True) * 100)
# highlighting the maximum percentage type
explode = (0.1, 0, 0, 0)

# making the figure and axes
fig, ax = plt.subplots()
# making a pie chart
pie = ax.pie(sizes, explode=explode, labels=labels, shadow=True, autopct='%1.1f%%', startangle=90)

# Customizing the plot
ax.set(title="Ratio of Chest Pain types in Patients with Heart Disease");
```

[14] ✓ 0.4s

Code for Question 10

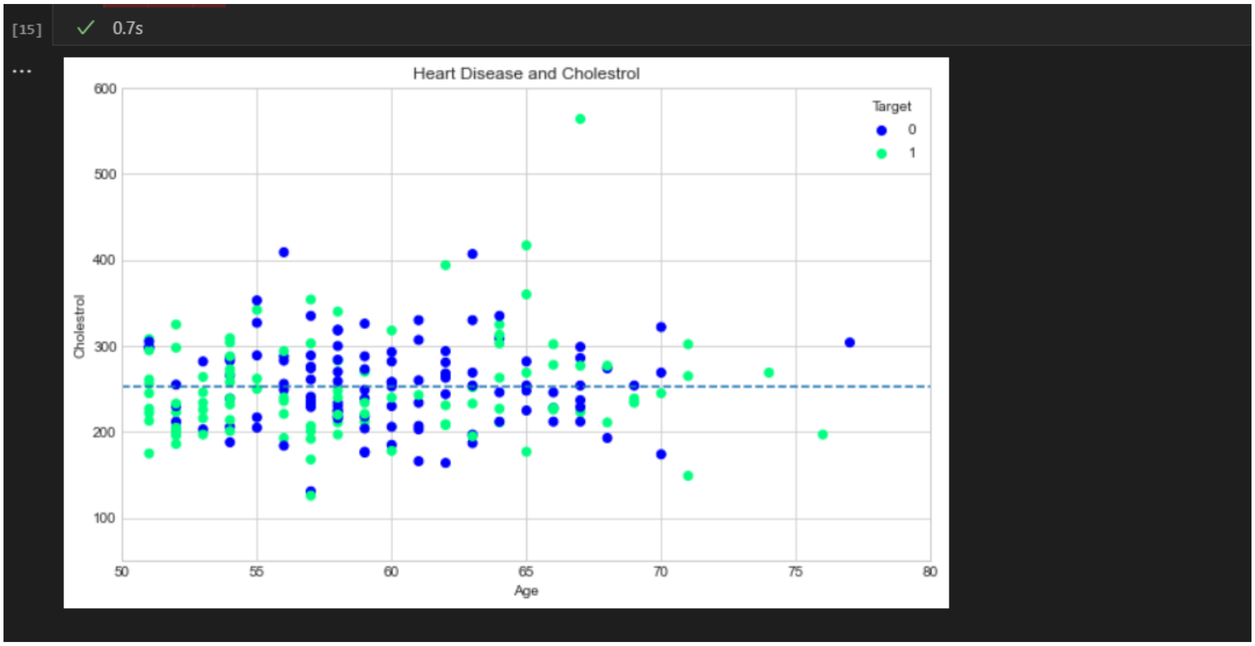
```
# getting the patients older than 50 years
over_50 = heart[heart['age'] > 50]

# Creating a figure and axes
fig, ax = plt.subplots(figsize=(10, 6))

# Plotting the data using axes
scatter = ax.scatter(x = over_50['age'],
                     y = over_50['chol'],
                     c = over_50['target'],
                     cmap='winter')

# Customizing the plot
ax.set(title = 'Heart Disease and Cholestrol',
       xlabel = 'Age',
       ylabel = 'Cholestrol')
ax.set_xlim([50, 80])
ax.set_ylim([50, 600])

# Adding the legend
ax.legend(*scatter.legend_elements(), title = 'Target')

# Adding a horizontal line for mean value
ax.axhline(over_50['chol'].mean(),
           linestyle='--');
```

Screenshot for Question 10



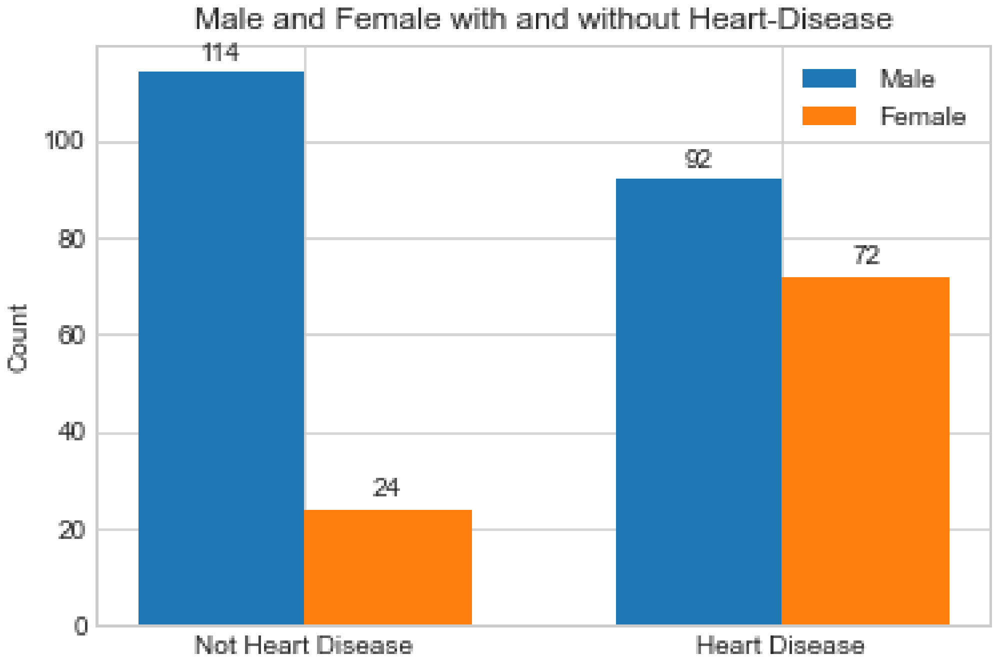
10. Display the cholesterol level range for the patients above age 50 of both with the heart disease, and without heart disease using scatter plot. Also display the average cholesterol level.

```
# getting the patients older than 50 years
over_50 = heart[heart['age'] > 50]

# Creating a figure and axes
fig, ax = plt.subplots(figsize=(10, 6))

# Plotting the data using axes
scatter = ax.scatter(x = over_50['age'],
                     y = over_50['chol'],
                     c = over_50['target'],
                     cmap='winter')

# Customizing the plot
ax.set(title = 'Heart Disease and Cholestrol',
       xlabel = 'Age',
       ylabel = 'Cholestrol')
ax.set_xlim([50, 80])
ax.set_ylim([50, 600])

# Adding the legend
ax.legend(*scatter.legend_elements(), title = 'Target')

# Adding a horizontal line for mean value
ax.axhline(over_50['chol'].mean(),
           linestyle='--');
```

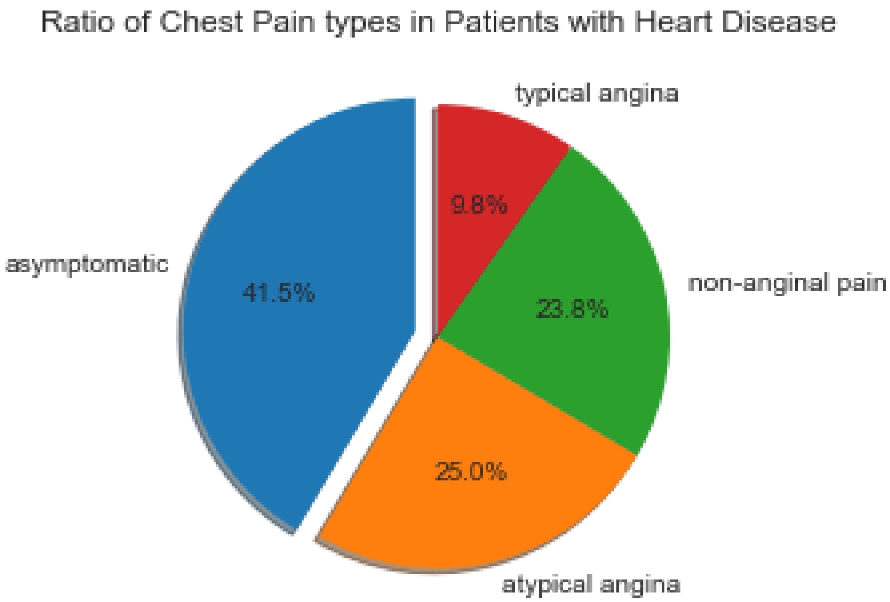


Visualizations as separate Images

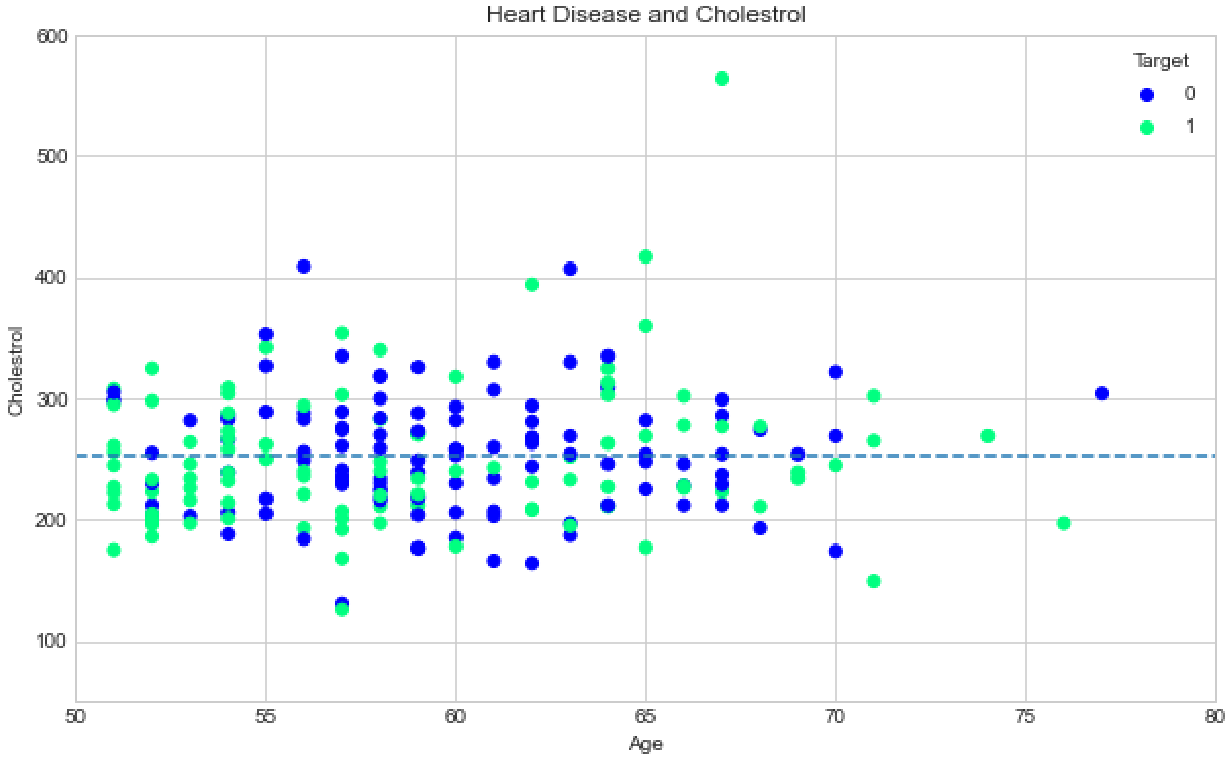
Question 8



Question 9



Question 10



Executive Summary

Introduction

This project is related to the heart disease of the patient. The dataset is manipulated, and the analysis is done to check the symptoms, conditions, and behavior of different attributes that contribute to making sound decisions regarding the patients and the disease. Skill in useful libraries like pandas and matplotlib is used to get helpful information. We have patients of different ages whose data has been taken as input, and the goal state is determined. Thirteen additional attributes of the patient helped to analyze the heart disease prediction.

Key Findings

The dataset was read successfully and stored in the pandas' data frame using the pandas' library. Then different analyses were made by following the specific criteria in each task. The team was able to sort the patient details, filter the patients based on some condition(s), and successfully group the patients into different categories. We were also able to visualize the parties regarding the patients as we have made a chart showing the males and females with and without the disease. Some exciting information related to the chest pain types of the patients has been visualized using the pie chart. We also used the scatter plot to keep track of the patients' cholesterol levels concerning their age, and the action is differentiating the patients with heart disease and without heart disease using legends. We used the mean () to collect the average value of the different attributes at different places to make quick decisions. We can now make some valuable recommendations based on these findings.

Solution

From our key findings, we have concluded that a relatively more minor number of patients have heart disease before their 50s. So, heart disease has been reported after 50 in most cases. We have seen that the person said an issue with the minimum age of 29 and a maximum of 77 in our dataset, which means that before the age of 30, there is an exceptional case that the person would have heart disease. Also, many people who do not have pain in the chest have normal thalassemia value. Otherwise, if a person has any chest pain, they will also have some problem with thalassemia value. But we have seen that more than 50% of the person with the fixed defect has asymptomatic chest pain. Our findings show that females have less cholesterol value if they don't have diabetes. It increases in the case of having diabetes. The male's cholesterol usually doesn't fluctuate much with diabetes. We have also seen that the ratio of heart disease in males is more than the disease in females. Our analysis shows that the cholesterol value of the patients is between 200 and 300 for most of the patients. This is regardless of the disease in a patient.

Our analysis has provided us with different helpful information on which we have made some predictions and the decisions about different categories of patients. We are now able to make some strong recommendations like if a person with age less than 30 is having some issue in heart or breathing etc., there is a rare chance that they would have heart disease. And the probability in females becomes even less. We can also recommend that if a person has heart disease, there is less chance that they would have diabetes as well. Different other recommendations can be made because the information from the dataset justifies them. The data analysis has helped a lot in analyzing the dataset to make some good decisions.

References

https://archive.ics.uci.edu/ml/datasets/heart%2BDisease

https://pandas.pydata.org/docs/index.html

https://matplotlib.org/