UCI Heart Disease Data Analysis and Data Visualization

Data Analysis and Data Visualization Project

Department of Computer Science-Data Science

QIS College of Engineering & Technology

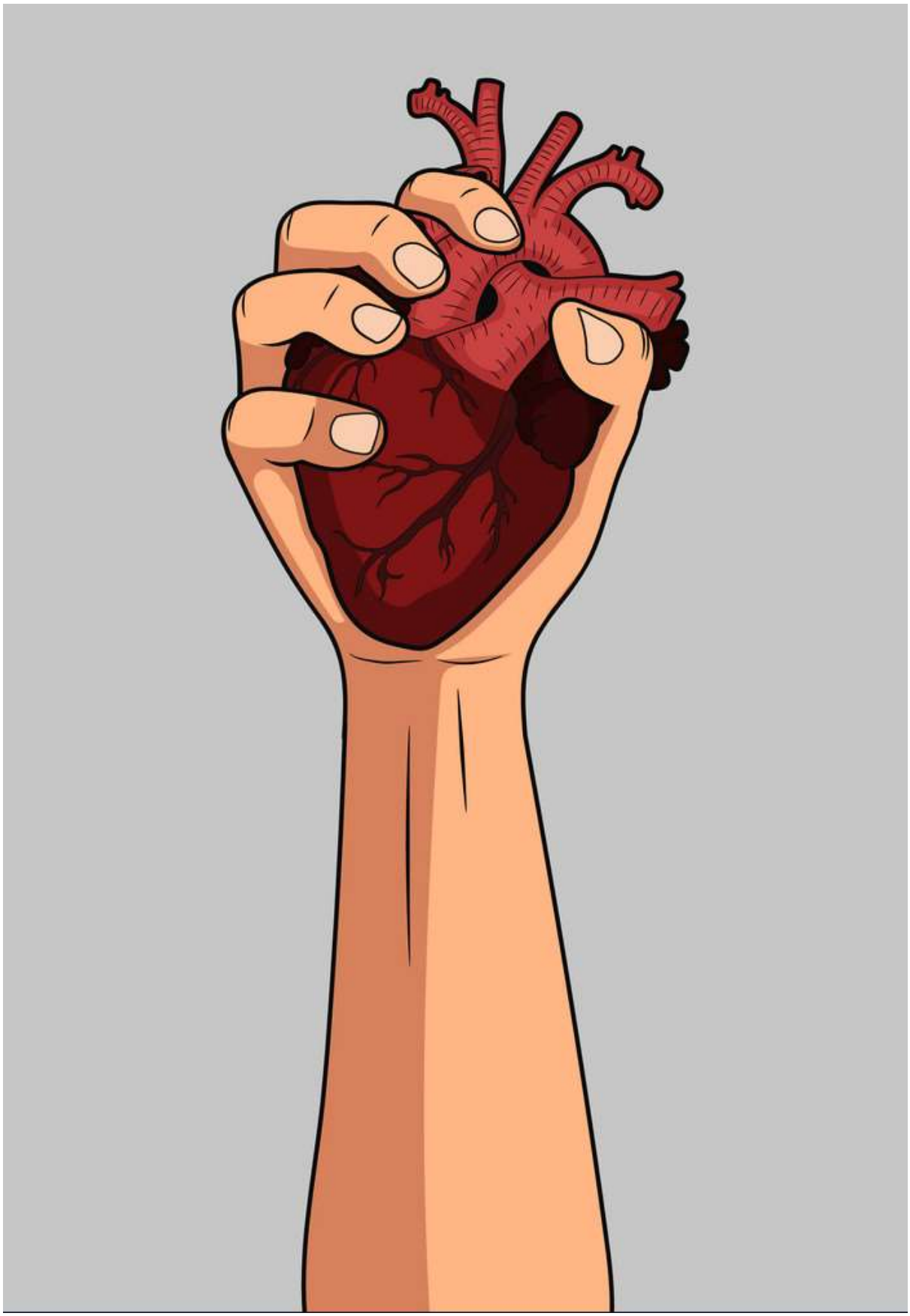
2022

Submitted by:

- GOTHULA PUJITHA(20491A4421)
- DAGGUPATI TRIVENI (20491A4452)
- PALUVADI BHAVANA (20491A4453)
- KANDIMALLA JOSHNAVI (20491A4462)
- SHAIK NAJINI (20491A4407)
- KAKARLAPUDI SAISUMA (20491A4427)
- SYED RUHIPARVEEN (20491A4447)
- RAMACHANDRUNI ANJANEYA SHANMUKH (20491A4413)
- BOGGAVARAPU LIKHITHACHOWDARY (20491A4415)
- SHAIK AYESHA (20491A4445)
- MADDIBOINA KOUSHIK (20491A4441)

Supervisor:

Dr.M.Suresh



Contents

ACKNOWLEDGEMENT 1

Heart Disease: Project Description 2

Objective

Background

Let`s talk Data: Data Description 3

Description

Packages Required 3

Packages

Load Dataset 4

Description for the columns in Dataset 5

Data Cleaning 7

Are there any missing values?

Exploratory Analysis 8

1. Find the mean for every column
2. Find the mode for every column
3. Find the minimum value in every column
4. Find the maximum value in every column
5. Find total missing values of each column with a single statement.
6. Find row wise mean
7. Find row wise minimum
8. Find row wise maximum
9. Find unique values of each column
10. Find Duplicate values
11. Drop Duplicate values

12. Find the patients whose age is greater than 60 and not suffering from any heart disease.
13. Dropping a column in the dataset
14. Find the patients who don't have diabetes but have a heart disease.
15. Find the patients whose ECG is normal don't have exercise induced angina but still have a heart rate less than 100
16. Find the correlation between Cholesterol measure and Severity Index
17. Show an example for Binning.
18. Show how to map Thalassemia Index with Thalassemia Severity.
19. Plot pie chart representing different types of Chest Pain.
20. Plot a bar graph depicting the count of people suffering from different types of thalassemia.
21. Plot pie charts depicting Severity Index in different regions.
22. Plot scatter plot depicting Max Heart Rate and Resting Blood Pressure among Males and females.
23. Plot a box plot showing the values (Cholesterol Measure values) which are too much deviated from the data.
24. Plot a histogram depicting Severity Index which is greater than 0 among different age groups.

Summary **37**

References **41**

ACKNOWLEDGEMENT

We express our sincere indebtedness towards our guide Dr M Suresh, Department of Computer Science, QIS COLLEGE OF ENGINEERING AND TECHNOLOGY for his invaluable guidance, suggestions, and supervision throughout the work. Without his kind patronage and guidance, the project would not have taken shape. We would also like to express our gratitude and sincere regards for his kind approval of the project, time to time counselling, and advice.

UCI Heart Disease: Project Description

OBJECTIVE

The objective of this analysis is to explore the UCI Heart Disease dataset through different visualizations and answer interesting questions to get more insights about the dataset.

BACKGROUND

Your heart is the main organ of your cardiovascular system, a network of blood vessels that pumps blood throughout your body. It also works with other body systems to control your heart rate and blood pressure.

A healthy heart is central to overall good health. Embracing a healthy lifestyle at any age can prevent heart disease and lower your risk for a heart attack or stroke. You are never too old or too young to begin taking care of your heart. True, the younger you begin making healthy choices, the longer you can reap the benefits. But swapping good habits for bad to promote good health can make a difference, even if you've already suffered a heart attack.

Let's Talk Data: Data Description

Now a days heart diseases has become one of the main deaths causing element for human race. It is because of lack of awareness on our health condition. A patient will face heart attack only when he is suffering from different health problems for the long time. Because of not identifying this type of problems leads to heart attacks. One of the major tasks on this dataset is to predict based on the given attributes of a patient that whether that particular person has heart disease or not and other is the experimental task to diagnose and find out various insights from this dataset which could help in understanding the problem more.

The data set includes the following features:

Each row displays the information about the selected patients and each columns displays the attributes required for our analysis and predictions.

Packages Required

1. Pandas: pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labelled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

2. We used a tool called Tableau, for easier analysis of graphs.

Loading Data Set

Here, we use some inbuilt methods to import the csv files

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data)
```

And place the path of the file.

Output:

```
      id  age     sex        dataset  ...        slope  ca               thal num
0      1   63    Male      Cleveland  ...  downsloping   0       fixed defect   0
1      2   67    Male      Cleveland  ...         flat   3             normal   2
2      3   67    Male      Cleveland  ...         flat   2  reversable defect   1
3      4   37    Male      Cleveland  ...  downsloping   0             normal   0
4      5   41  Female      Cleveland  ...    upsloping   0             normal   0
..   ...  ...     ...            ...  ...          ...  ..                ...  ..
915  916   54  Female  VA Long Beach  ...         flat   0               NaN   1
916  917   62    Male  VA Long Beach  ...         flat   0               NaN   0
917  918   55    Male  VA Long Beach  ...         flat   0       fixed defect   2
918  919   58    Male  VA Long Beach  ...         flat   0               NaN   0
919  920   62    Male  VA Long Beach  ...         flat   0               NaN   1

[920 rows x 16 columns]
```

The dataset consists of 920 rows and 16 columns.

DESCRIPTION For the Columns in the Data set

COLUMN 1: ID

This column represents the unique identity number for the patients.

COLUMN 2: AGE

This column represents the age of the patient in years.

COLUMN 3: SEX

This column represents the sex of the patient in Boolean

COLUMN 4: DATASET

This column represents the place of the patient.

It contains 4 types:

1. Cleveland
2. Hungary
3. Switzerland
4. VA Long Beach

COLUMN 5: CPT

This column represents the Chest Pain Type of the patient.

It contains 4 types:

1. Asymptomatic
2. Atypical angina
3. Non anginal
4. Typical angina

COLUMN 6: RESTBP

This column represents the Resting Blood Pressure of the patient in Numbers. (mmHg)

COLUMN 7: **SCHOL**

This column represents the serum cholesterol in Numbers (mg/dl).

COLUMN 8: **FBS**

This column represents the Fasting Blood Sugar in Numbers (mg/dl).

COLUMN 9: **RESTECG**

This column represents the Resting electrocardiographic results.

It contains 4 types:

1. Lv hypertrophy
2. Normal
3. St-t abnormality
4. (Blanks)

COLUMN 10: **MHRACH**

This column represents the Maximum Heart Rate achieved by the patient in Numbers. (BPM)

COLUMN 11: **EXANG**

This column represents the exercise induced in angina in Boolean Values

COLUMN 12: **OLDPEAK**

This column represents ST depression induced by exercise relative to rest. In Float Values.

COLUMN 13: **SLOPE**

This column represents the slope of the peak exercise ST segment
1. upsloping
2: flat
3: down sloping
4: Blanks

COLUMN 14: **CA**

This column represents number of major vessels (0-3) colored by fluoroscopy in numbers

COLUMN 15: **THAL**

This column represents type of Thalassemia index.

It contains 4 types:

1. Normal
2. Fixed Defect
3. Reversible Defect
4. Blanks

COLUMN 16: **NUM**

This column represents diagnosis of heart disease and severity index (angiographic disease status).

It contains range from 0-5.

Data Cleaning

The empty spaces in the columns are being filled with random numbers in the dataset. We had made some changes in the dataset. Please observe.

Exploratory Analysis

1. **Find the mean for every column**
   Mean is nothing but average

   Mean=Sum of observations/Total no. of observations.

   Here, we had used the .mean() method for calculating average.

   i. Mean for the age column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['age'].mean())
```

   Output:

```
53.51086956521739
```

   ii. The Mean for the RestBp column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restbp'].mean())
```

   Output:

```
132.26728110599078
```

iii. The Mean for the SChol column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['schol'].mean())
```

Output:

```
199.13033707865168
```

iv. The Mean for the MHRAch column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['mhrach'].mean())
```

Output:

```
137.5456647398844
```

v. The Mean for the OldPeak Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['oldpeak'].mean())
```

Output:

```
0.8195652173913044
```

2. **Find the mode for every column**
   Mode is most frequently occurred value in the given column.

   To find the mode , we use .mode () method

i. The Mode of the Age Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['age'].mode())
```

Output:

```
0    54
Name: age, dtype: int64
```

ii. The Mode of the Sex Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['sex'].mode())
```

Output:

```
0    Male
Name: sex, dtype: object
```

iii. The Mode of the Cpt Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['cpt'].mode())
```

Output:

```
0    asymptomatic
Name: cpt, dtype: object
```

iv. The Mode of the RestBp Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restbp'].mode())
```

Output:

```
0    120.0
Name: restbp, dtype: float64
```

v. The Mode of the Schol Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['schol'].mode())
```

Output:

```
0    0.0
Name: schol, dtype: float64
```

vi. The Mode of the RestECG Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restecg'].mode())
```

Output:

```
0    normal
Name: restecg, dtype: object
```

vii. The Mode of the MHRAch Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['mhrach'].mode())
```

Output:

```
0    150.0
Name: mhrach, dtype: float64
```

viii. The Mode of the Old peak Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['oldpeak'].mode())
```

Output:

```
0    0.0
Name: oldpeak, dtype: float64
```

ix. The Mode of the Slope column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['slope'].mode())
```

Output:

```
0    flat
Name: slope, dtype: object
```

x. The Mode of the CA column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['ca'].mode())
```

Output:

```
0    0
Name: ca, dtype: int64
```

xi. The Mode of the Thal column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['thal'].mode())
```

Output:

```
0    normal
Name: thal, dtype: object
```

xii. The Mode of the Num column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['num'].mode())
```

Output:

```
0    0
Name: num, dtype: int64
```

3. Find the Minimum in every column

Minimum is the least value in the given column
It is calculated using .min() method.

i. The Minimum in the age column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['age'].min())
```

Output:

```
28
```

ii. The Minimum in the RestBp column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restbp'].min())
```

Output:

```
80.0
```

iii. The Minimum in the Schol column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['schol'].min())
```

Output:

```
0.0
```

iv. The Minimum in the MHRAch column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['mhrach'].min())
```

Output:

```
60.0
```

v. The Minimum in the Old Peak column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['oldpeak'].min())
```

Output:

```
-2.6
```

vi. The Minimum in the CA column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['ca'].min())
```

Output:

```
0
```

vii. The Minimum in the NUM column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['num'].min())
```

Output:

```
0
```

4. **Find the Maximum in every column**
   Maximum is the highest value in the given column
   It can be calculated by using .max() method.

i. The Maximum in the age column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['age'].max())
```

Output:

```
77
```

ii. The Maximum in the RestBp column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restbp'].max())
```

Output:

```
200.0
```

iii. The Maximum in the Schol column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['schol'].max())
```

Output:

```
603.0
```

iv. The Maximum in the MHRAch column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['mhrach'].max())
```

Output:

```
202.0
```

v. The Maximum in the Old Peak column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['oldpeak'].max())
```

Output:

```
6.2
```

vi. The Maximum in the CA column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['ca'].max())
```

Output:

```
4
```

vii. The Maximum in the NUM column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['num'].max())
```

Output:

```
4
```

5. **Find total missing values of each column with a single statement.**

   Here, we use isnull(). It gives Boolean value.
   And we used sum() to calculate total True values.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.isnull().sum())
```

Output:

```
id            0
age           0
sex           0
dataset       0
cpt           0
restbp       52
schol        30
fbs          90
restecg       2
mhrach       55
exang        55
oldpeak       0
slope         0
ca            0
thal        486
num           0
dtype: int64
```

6. Find Row wise mean

To find mean row wise, we use axis=1

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.mean(axis=1))
```

Output:

```
0         74.287500
1         78.687500
2         69.200000
3         76.437500
4         69.175000
             ...
915      198.125000
916      186.333333
917      177.500000
918      227.000000
919      181.250000
Length: 920, dtype: float64
```

7. Find Row wise Minimum

To Find row wise min we use axis=1

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.min(axis=1))
```

Output:

```
0      0.0
1      1.5
2      1.0
3      0.0
4      0.0
      ...
915    0.0
916    0.0
917    0.0
918    0.0
919    0.0
Length: 920, dtype: float64
```

8. Find Row wise Maximum

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.max(axis=1))
```

Output:

```
0       233.0
1       286.0
2       229.0
3       250.0
4       204.0
        ...
915     916.0
916     917.0
917     918.0
918     919.0
919     920.0
Length: 920, dtype: float64
```

9. Find unique values of each column

We use .unique() command on each column

i. For Age Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['age'].unique())
```

Output:

```
[63 67 37 41 56 62 57 53 44 52 48 54 49 64 58 60 50 66 43 40 69 59 42 55
 61 65 71 51 46 45 39 68 47 34 35 29 70 77 38 74 76 28 30 31 32 33 36 72
 73 75]
```

ii. For Sex Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['sex'].unique())
```

Output:

```
['Male' 'Female']
```

iii. For dataset column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['dataset'].unique())
```

Output:

```
['Cleveland' 'Hungary' 'Switzerland' 'VA Long Beach']
```

iv. For cpt column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['cpt'].unique())
```

Output:

```
['typical angina' 'asymptomatic' 'non-anginal' 'atypical angina']
```

v. For Rest Bp column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restbp'].unique())
```

Output:

```
[145. 160. 120. 130. 140. 172. 150. 110. 132. 117. 135. 112. 105. 124.
 125. 142. 128. 170. 155. 104. 180. 138. 108. 134. 122. 115. 118. 100.
 200.  94. 165. 102. 152. 101. 126. 174. 148. 178. 158. 192. 129. 144.
 123. 136. 146. 106. 156. 154. 114. 164.  98. 190.  nan 113.  92.  95.
  80. 185. 116. 153. 111.  96. 127.]
```

vi. For schol column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['schol'].unique())
'
```

Output:

```
[233. 286. 229. 250. 204. 236. 268. 354. 254. 203. 192. 294. 256. 263.
 199. 168. 239. 275. 266. 211. 283. 284. 224. 206. 219. 340. 226. 247.
 167. 230. 335. 234. 177. 276. 353. 243. 225. 302. 212. 330. 175. 417.
 197. 198. 290. 253. 172. 273. 213. 305. 216. 304. 188. 282. 185. 232.
 326. 231. 269. 267. 248. 360. 258. 308. 245. 270. 208. 264. 321. 274.
 325. 235. 257. 164. 141. 252. 255. 201. 222. 260. 182. 303. 265. 309.
 307. 249. 186. 341. 183. 407. 217. 288. 220. 209. 227. 261. 174. 281.
 221. 205. 240. 289. 318. 298. 564. 246. 322. 299. 300. 293. 277. 214.
 207. 223. 160. 394. 184. 315. 409. 244. 195. 196. 126. 313. 259. 200.
 262. 215. 228. 193. 271. 210. 327. 149. 295. 306. 178. 237. 218. 242.
 319. 166. 180. 311. 278. 342. 169. 187. 157. 176. 241. 131. 132.  nan
 161. 173. 194. 297. 292. 339. 147. 291. 358. 412. 238. 163. 280. 202.
 328. 129. 190. 179. 272. 100. 468. 320. 312. 171. 365. 344.  85. 347.
 251. 287. 156. 117. 466. 338. 529. 392. 329. 355. 603. 404. 518. 285.
 279. 388. 336. 491. 331. 393.   0. 153. 316. 458. 384. 349. 142. 181.
 310. 170. 369. 165. 337. 333. 139. 385.]
```

vii. For FBS Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['fbs'].unique())
```

Output:

```
[False True nan]
```

viii. For RestEcg column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['restecg'].unique())
```

Output:

```
['lv hypertrophy' 'normal' 'st-t abnormality' nan]
```

ix. For MHRAch column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['mhrach'].unique())
```

Output:

```
[150. 108. 129. 187. 172. 178. 160. 163. 147. 155. 148. 153. 142. 173.
 162. 174. 168. 139. 171. 144. 132. 158. 114. 151. 161. 179. 120. 112.
 137. 157. 169. 165. 123. 128. 152. 140. 188. 109. 125. 131. 170. 113.
  99. 177. 141. 180. 111. 143. 182. 156. 115. 149. 145. 146. 175. 186.
 185. 159. 130. 190. 136.  97. 127. 154. 133. 126. 202. 103. 166. 164.
 184. 124. 122.  96. 138.  88. 105. 194. 195. 106. 167.  95. 192. 117.
 121. 116.  71. 118. 181. 134.  90.  98. 176. 135. 110.  nan 100.  87.
 102.  92.  91.  82. 119.  94. 104.  60.  83.  63.  70.  77.  72.  78.
  86.  93.  67.  84.  80. 107.  69.  73.]
```

x. For Exang column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['exang'].unique())
```

Output:

```
[False True nan]
```

xi. For oldpeak Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['oldpeak'].unique())
```

Output:

```
[ 2.3  1.5  2.6  3.5  1.4  0.8  3.6  0.6  3.1  0.4  1.3  0.   0.5  1.6
  1.   1.2  0.2  1.8  3.2  2.4  2.   2.5  2.2  2.8  3.   3.4  6.2  4.
  5.6  2.9  0.1  2.1  1.9  4.2  0.9  1.1  3.8  0.7  0.3  4.4  5.  -1.1
 -1.5 -0.1 -2.6 -0.7 -2.  -1.   1.7 -0.8 -0.5 -0.9  3.7]
```

xii. For slope Column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['slope'].unique())
```

Output:

```
['downsloping' 'flat' 'upsloping']
```

xiii. For CA column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv"
print(data['ca'].unique())
```

Output:

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

xiv. For thal column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['thal'].unique())
```

Output:

```
['fixed defect' 'normal' 'reversable defect' nan]
```

xv. For num column

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data['num'].unique())
```

Output:

```
[0 2 1 3 4]
```

10. Find Duplicate Values

Here, we use .duplicate() command to check the duplicate values and .sum() to count the number of duplicate values

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.duplicated().sum())
```

Output:

```
"C:\Users\shanm\PycharmProjects\Heart Disease\venv\Scripts\python.exe"
0
```

11. Drop Duplicate Values

In case of any duplicate values, to remove them we use drop_duplicates() method to delete them.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.drop_duplicates())
```

Output:

```
      id  age     sex        dataset  ...        slope  ca               thal num
0      1   63    Male      Cleveland  ...  downsloping   0       fixed defect   0
1      2   67    Male      Cleveland  ...         flat   3             normal   2
2      3   67    Male      Cleveland  ...         flat   2  reversable defect   1
3      4   37    Male      Cleveland  ...  downsloping   0             normal   0
4      5   41  Female      Cleveland  ...    upsloping   0             normal   0
..   ...  ...     ...            ...  ...          ...  ..                ...  ..
915  916   54  Female  VA Long Beach  ...         flat   0               NaN   1
916  917   62    Male  VA Long Beach  ...         flat   0               NaN   0
917  918   55    Male  VA Long Beach  ...         flat   0       fixed defect   2
918  919   58    Male  VA Long Beach  ...         flat   0               NaN   0
919  920   62    Male  VA Long Beach  ...         flat   0               NaN   1

[920 rows x 16 columns]
```

12. Find the patients whose age is greater than 60 and not suffering from any heart disease.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
x=data[(data['age']>60)&(data['num']==0)]
print(x)
```

0utput:

	id	age	sex	dataset	...	slope	ca	thal	num
0	1	63	Male	Cleveland	...	downsloping	0	fixed defect	0
20	21	64	Male	Cleveland	...	flat	0	normal	0
27	28	66	Female	Cleveland	...	downsloping	0	normal	0
30	31	69	Female	Cleveland	...	upsloping	2	normal	0
39	40	61	Male	Cleveland	...	flat	0	normal	0
42	43	71	Female	Cleveland	...	upsloping	2	normal	0
48	49	65	Female	Cleveland	...	upsloping	1	normal	0
51	52	65	Male	Cleveland	...	upsloping	0	reversable defect	0
70	71	65	Female	Cleveland	...	upsloping	0	normal	0
75	76	65	Female	Cleveland	...	upsloping	0	normal	0
90	91	66	Male	Cleveland	...	flat	0	normal	0
92	93	62	Male	Cleveland	...	flat	3	reversable defect	0
94	95	63	Female	Cleveland	...	upsloping	0	normal	0
129	130	62	Female	Cleveland	...	upsloping	0	normal	0
152	153	67	Female	Cleveland	...	flat	0	reversable defect	0
159	160	68	Male	Cleveland	...	upsloping	1	reversable defect	0
173	174	62	Female	Cleveland	...	flat	0	normal	0
185	186	63	Female	Cleveland	...	upsloping	2	normal	0
194	195	68	Female	Cleveland	...	flat	0	normal	0
196	197	69	Male	Cleveland	...	flat	1	normal	0
201	202	64	Female	Cleveland	...	upsloping	0	normal	0
203	204	64	Female	Cleveland	...	upsloping	0	reversable defect	0
218	219	64	Female	Cleveland	...	flat	2	normal	0
227	228	67	Female	Cleveland	...	upsloping	1	normal	0
233	234	74	Female	Cleveland	...	upsloping	1	normal	0
249	250	62	Male	Cleveland	...	upsloping	0	normal	0
252	253	64	Male	Cleveland	...	flat	1	reversable defect	0
256	257	67	Female	Cleveland	...	upsloping	2	normal	0

```
490  491   62    Male        Hungary  ...    upsloping   0                NaN   0
706  707   65    Male    Switzerland  ...    upsloping   0                NaN   0
717  718   72    Male    Switzerland  ...         flat   2                NaN   0
724  725   66    Male  VA Long Beach  ...         flat   0                NaN   0
725  726   66    Male  VA Long Beach  ...    upsloping   0                NaN   0
735  736   62    Male  VA Long Beach  ...         flat   0                NaN   0
738  739   63    Male  VA Long Beach  ...  downsloping   0                NaN   0
743  744   74    Male  VA Long Beach  ...         flat   0                NaN   0
771  772   63  Female  VA Long Beach  ...         flat   0                NaN   0
778  779   62    Male  VA Long Beach  ...         flat   0                NaN   0
792  793   65    Male  VA Long Beach  ...         flat   0                NaN   0
819  820   63    Male  VA Long Beach  ...         flat   0                NaN   0
824  825   64    Male  VA Long Beach  ...         flat   0                NaN   0
831  832   61    Male  VA Long Beach  ...         flat   0                NaN   0
848  849   61  Female  VA Long Beach  ...         flat   0  reversable defect   0
```

```
824  825   64    Male  VA Long Beach  ...         flat   0                NaN   0
831  832   61    Male  VA Long Beach  ...         flat   0                NaN   0
848  849   61  Female  VA Long Beach  ...         flat   0  reversable defect   0
853  854   68    Male  VA Long Beach  ...         flat   0                NaN   0
855  856   62    Male  VA Long Beach  ...         flat   0                NaN   0
860  861   75    Male  VA Long Beach  ...  downsloping   0  reversable defect   0
887  888   69    Male  VA Long Beach  ...         flat   0                NaN   0
894  895   63    Male  VA Long Beach  ...         flat   0                NaN   0
909  910   68    Male  VA Long Beach  ...         flat   0             normal   0
916  917   62    Male  VA Long Beach  ...         flat   0                NaN   0

[60 rows x 16 columns]
```

13. Dropping a column in the dataset

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(data.drop('id',axis=1))
```

Output:

```
      age     sex        dataset  ... ca               thal  num
0      63    Male      Cleveland  ...  0       fixed defect    0
1      67    Male      Cleveland  ...  3             normal    2
2      67    Male      Cleveland  ...  2  reversable defect    1
3      37    Male      Cleveland  ...  0             normal    0
4      41  Female      Cleveland  ...  0             normal    0
..    ...     ...            ...  ... ..                ...  ...
915    54  Female  VA Long Beach  ...  0               NaN    1
916    62    Male  VA Long Beach  ...  0               NaN    0
917    55    Male  VA Long Beach  ...  0       fixed defect    2
918    58    Male  VA Long Beach  ...  0               NaN    0
919    62    Male  VA Long Beach  ...  0               NaN    1

[920 rows x 15 columns]
```

14. Find the patients who don't have diabetes but have a heart disease.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
x=data[(data['fbs']==False)&(data['num']>0)]
print(x)
```

Output:

```
      id  age     sex        dataset  ...        slope  ca               thal num
1      2   67    Male      Cleveland  ...         flat   3             normal   2
2      3   67    Male      Cleveland  ...         flat   2  reversable defect   1
6      7   62  Female      Cleveland  ...  downsloping   2             normal   3
8      9   63    Male      Cleveland  ...         flat   1  reversable defect   2
16    17   48    Male      Cleveland  ...  downsloping   0  reversable defect   1
..   ...  ...     ...            ...  ...          ...  ..               ...  ..
907  908   58    Male  VA Long Beach  ...         flat   0              NaN   2
908  909   74    Male  VA Long Beach  ...  downsloping   0              NaN   2
913  914   62    Male  VA Long Beach  ...         flat   0              NaN   1
914  915   46    Male  VA Long Beach  ...         flat   0             normal   2
919  920   62    Male  VA Long Beach  ...         flat   0              NaN   1

[339 rows x 16 columns]
```

15. Find the correlation between Cholesterol Measure and Severity Index.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
corr=data['schol'].corr(data['num'])
print(corr)
```

Output:

```
-0.2315471497051458
```

16. Find the patients whose ECG is normal don't have exercise induced angina but still have a heart rate less than 100.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
x=data[(data['restecg']=="normal")&(data['exang']==False)&(data['mhrach']<100)]
print(x)
```

Output:

```
      id  age     sex        dataset  ...        slope  ca               thal num
114  115   62  Female      Cleveland  ...         flat   1  reversable defect   2
244  245   60  Female      Cleveland  ...    upsloping   0             normal   0
245  246   67    Male      Cleveland  ...         flat   0             normal   2
381  382   46  Female        Hungary  ...         flat   0                NaN   0
474  475   57  Female        Hungary  ...         flat   0                NaN   0
627  628   51    Male    Switzerland  ...         flat   0                NaN   4
631  632   51    Male    Switzerland  ...         flat   0             normal   2
637  638   53    Male    Switzerland  ...         flat   0             normal   3
651  652   55    Male    Switzerland  ...         flat   0  reversable defect   2
652  653   56    Male    Switzerland  ...         flat   0  reversable defect   0
653  654   56    Male    Switzerland  ...         flat   0  reversable defect   2
670  671   59    Male    Switzerland  ...         flat   0       fixed defect   3
685  686   61    Male    Switzerland  ...         flat   0                NaN   3
708  709   66  Female    Switzerland  ...         flat   0  reversable defect   1
800  801   58    Male  VA Long Beach  ...  downsloping   0                NaN   0
854  855   55    Male  VA Long Beach  ...         flat   0                NaN   3
893  894   74    Male  VA Long Beach  ...         flat   0                NaN   1
903  904   56    Male  VA Long Beach  ...         flat   0  reversable defect   1

[18 rows x 16 columns]
```

17. Show an example for Binning

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
print(pd.qcut(data['schol'],q=4))
```

Output:

```
0       (223.0, 268.0]
1       (268.0, 603.0]
2       (223.0, 268.0]
3       (223.0, 268.0]
4       (175.0, 223.0]
           ...
915     (268.0, 603.0]
916    (-0.001, 175.0]
917     (175.0, 223.0]
918     (268.0, 603.0]
919     (223.0, 268.0]
Name: schol, Length: 920, dtype: category
Categories (4, interval[float64, right]): [(-0.001, 175.0] < (175.0, 223.0] < (223.0, 268.0] <
                                           (268.0, 603.0]]
```

18. Show how to Map Thalassemia Index with Thalassemia Severity

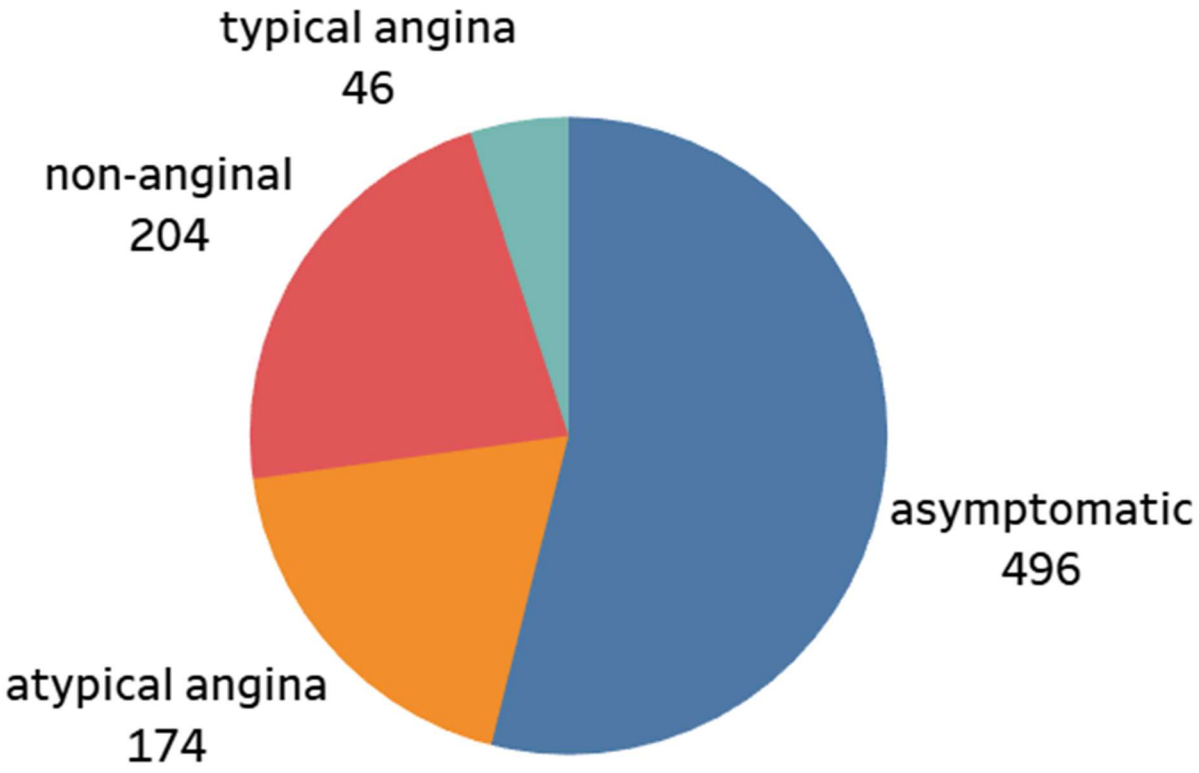
```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
thal_index={'normal':'3','fixed defect':'6','reversable defect':'7'}
st=data['thal']
data['num']=st.map(thal_index)
print(data)
```

Output:

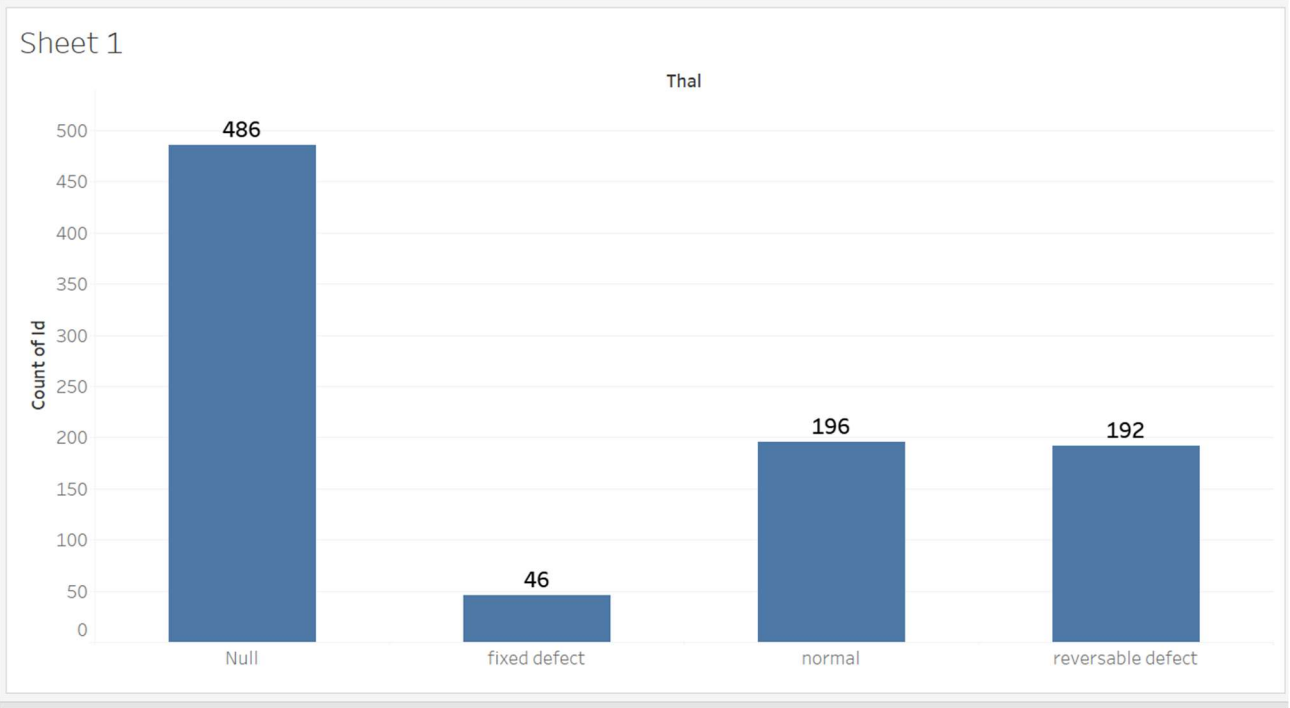
```
      id  age     sex        dataset  ...        slope  ca               thal  num
0      1   63    Male      Cleveland  ...  downsloping   0       fixed defect    6
1      2   67    Male      Cleveland  ...         flat   3             normal    3
2      3   67    Male      Cleveland  ...         flat   2  reversable defect    7
3      4   37    Male      Cleveland  ...  downsloping   0             normal    3
4      5   41  Female      Cleveland  ...    upsloping   0             normal    3
..   ...  ...     ...            ...  ...          ...  ..                ...  ...
915  916   54  Female  VA Long Beach  ...         flat   0               NaN  NaN
916  917   62    Male  VA Long Beach  ...         flat   0               NaN  NaN
917  918   55    Male  VA Long Beach  ...         flat   0       fixed defect    6
918  919   58    Male  VA Long Beach  ...         flat   0               NaN  NaN
919  920   62    Male  VA Long Beach  ...         flat   0               NaN  NaN

[920 rows x 16 columns]
```

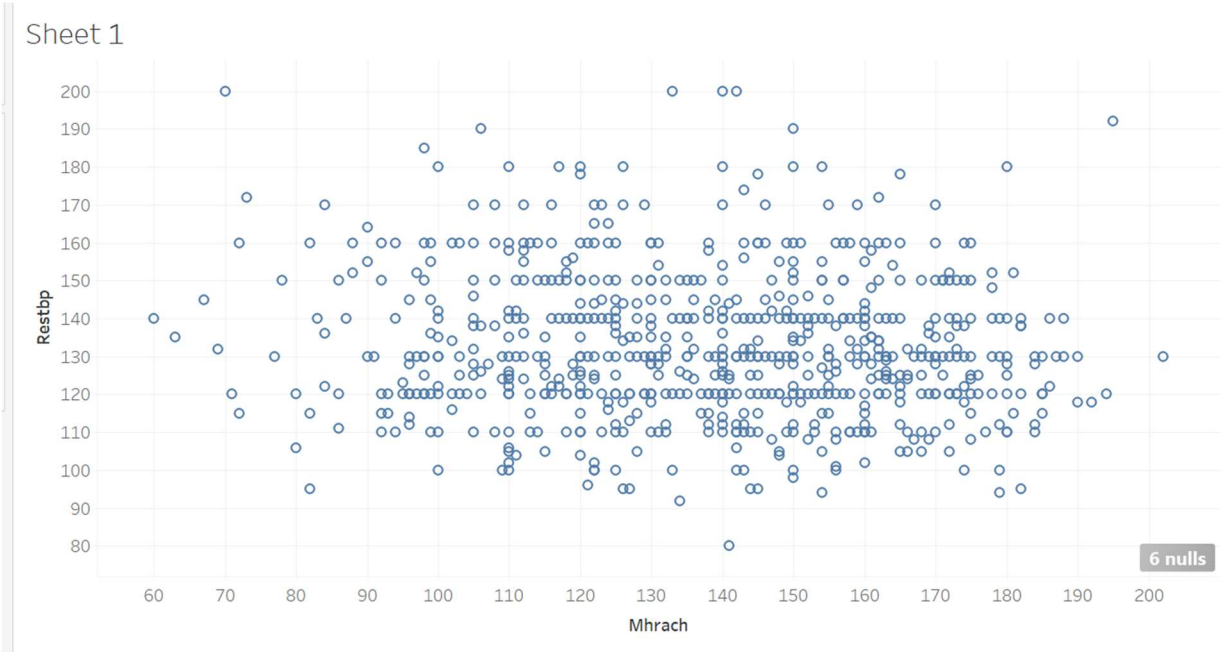
19. Plot a pie chart representing different types of chest pain among the people.



20. Plot a bar graph depicting the count of people suffering from different types of thalassemia.

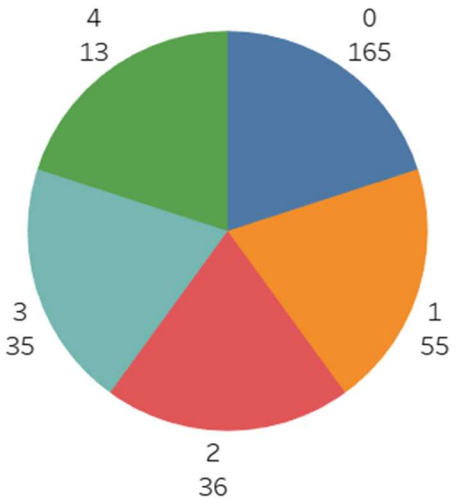


21. Plot a scatter plot depicting Max Heart Rate and Resting Blood Pressure among males and female

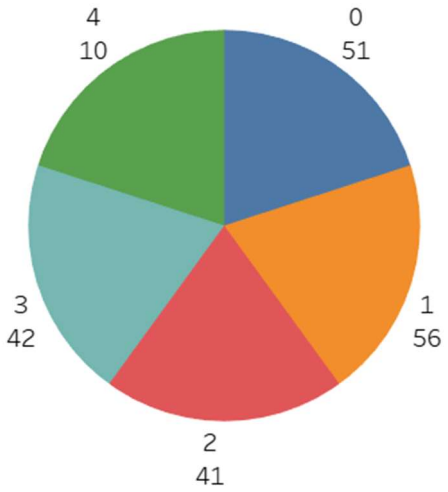


22. Plot pie charts depicting Severity Index in different regions.

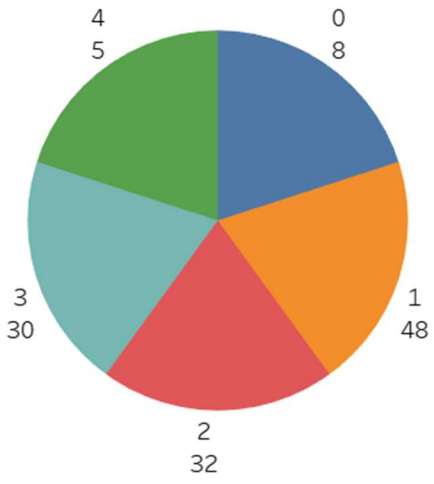
Dataset
Cleveland



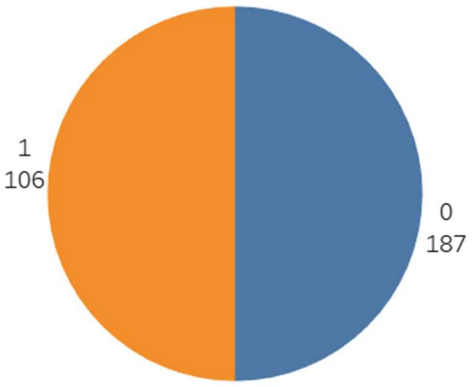
Dataset
VA Long Beach



Dataset
Switzerland

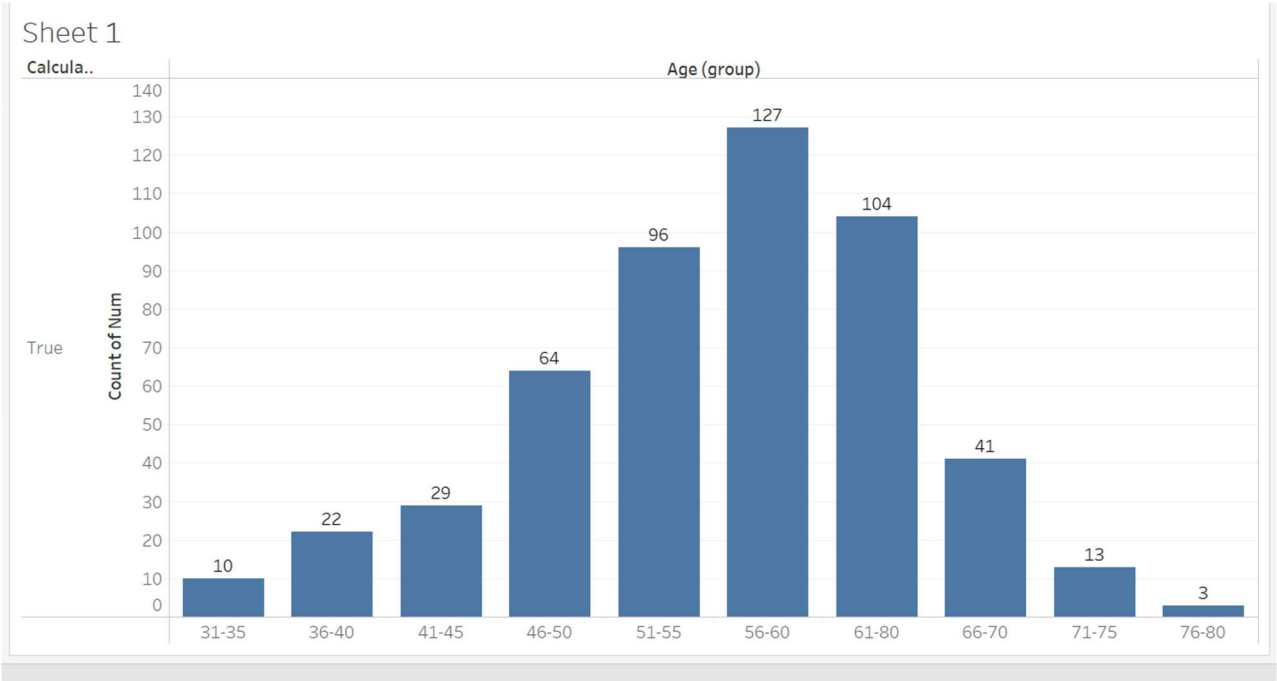


Dataset
Hungary



23. Plot a box plot showing the values (Cholesterol measure values) which are too much deviated from the data

24. Plot a histogram depicting Severity Index which is greater than 0 among different age groups.



Summary

i. Minimum values in each column

Age	**28**
Resting Blood Pressure	**80**
Serum Cholesterol	**0.0**
Maximum Heart Rate	**60**
Old Peak	**-2.6**
Cardiac Fluoroscopy	**0**
Severity Index	**0**

ii. Maximum values in each column

Age	**77**
Resting Blood Pressure	**200**
Serum Cholesterol	**603**
Maximum Heart Rate	**202**
Old Peak	**6.2**
Cardiac Fluoroscopy	**3**
Severity Index	**4**

iii. Mean values in each column

Age	**53.45**
Resting Blood Pressure	**132.13**
Serum Cholesterol	**199.13**
Maximum Heart Rate	**137.54**
Old Peak	**0.8196**

iv. Mode values in each column

Age	**54**
Chest Pain Type	**Asymptomatic**
Sex	**Male**
Resting Blood Pressure	**120**
Serum Cholesterol	**0**
Rest ECG	**Normal**
Maximum Heart Rate	**150**
Old Peak	**0**
slope	**Flat**
Cardiac Fluoroscopy	**0**
Thalassemia Index	**Normal**
Severity Index	**0**

According to the considered dataset,

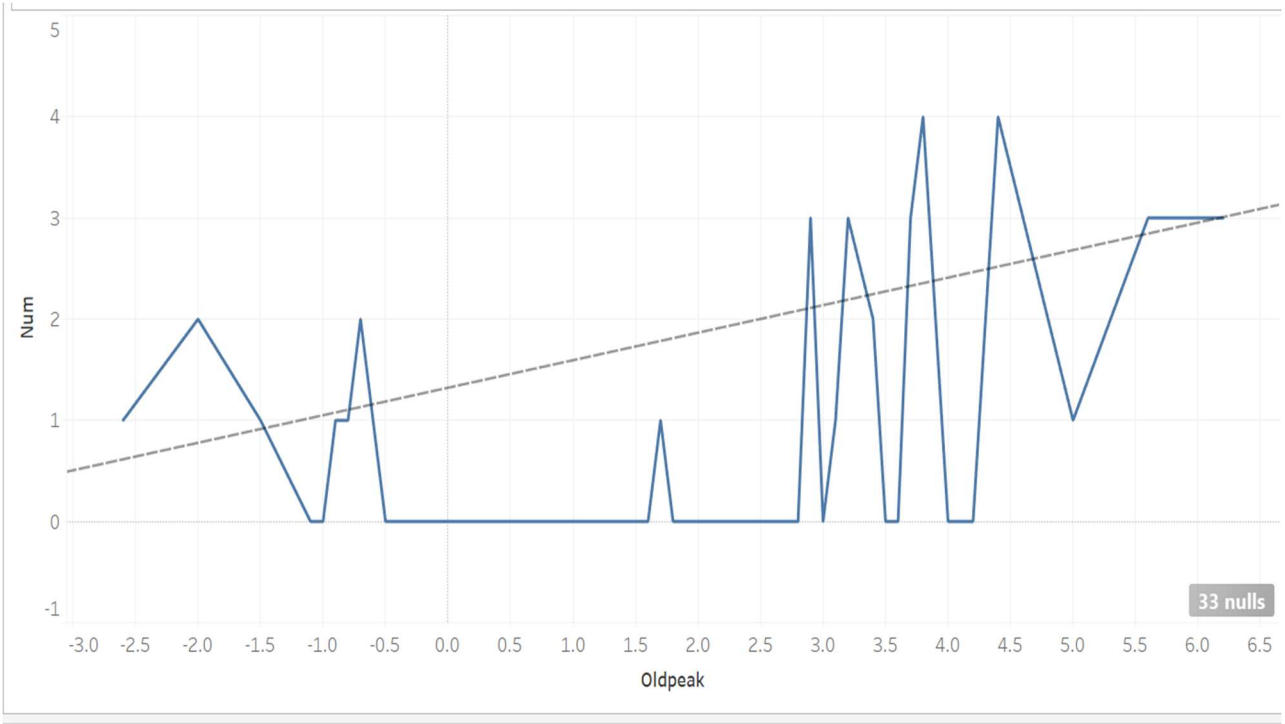
- The correlation between the CA (Colour By Fluoroscopy) and Num (Severity Index) is positive.
- In cardiac catheterization, fluoroscopy is used to help the healthcare provider see the flow of blood through the coronary arteries. It can check for arterial blockages.
- It indicates that having high CA (Colour By Fluoroscopy) leads to increase in Rate of Heart Disease.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
corr=data['ca'].corr(data['num'])
print(corr)
```

Output:

```
0.2617970896790229
```

Plotting Graph:



Likewise,

- The correlation between the OId Peak and Severity Index is also positive.
- Old Peak means ST depression induced by exercise relative to rest.
- It indicates that having High Old peak value lead to having heart disease.

```
import pandas as pd
data=pd.read_csv("C:/Users/shanm/Downloads/heart_disease_uci (1).csv")
corr=data['oldpeak'].corr(data['num'])
print(corr)
```

Output:

```
0.39332674660668293
```

References

The dataset can be downloaded from https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data