Kalyani Dhondge

Internship iNeuron project

Heart Disease Diagnostic Analysis

Importing Required Libraries

In [11]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Extracting CSV Dataset From System using Pandas Library

In [285…

```
df=pd.read_csv('heart_disease_dataset.csv')
df
```

Out[285…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	63	1	1	145	233	1	2	150	0	2.3	3	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3
2	67	1	4	120	229	0	2	129	1	2.6	2	2
3	37	1	3	130	250	0	0	187	0	3.5	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
298	45	1	1	110	264	0	0	132	0	1.2	2	0
299	68	1	4	144	193	1	0	141	0	3.4	2	2
300	57	1	4	130	131	0	0	115	1	1.2	2	1
301	57	0	2	130	236	0	2	174	0	0.0	2	1
302	38	1	3	138	175	0	0	173	0	0.0	1	-100000

303 rows × 14 columns

In [286…

```
df.head()
```

Out[286…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	n
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3

In [287…

```
df.tail()
```

Out[287…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
298	45	1	1	110	264	0	0	132	0	1.2	2	0
299	68	1	4	144	193	1	0	141	0	3.4	2	2
300	57	1	4	130	131	0	0	115	1	1.2	2	1
301	57	0	2	130	236	0	2	174	0	0.0	2	1
302	38	1	3	138	175	0	0	173	0	0.0	1	-100000

In [288…

```
#Checking Not null values
df.info()

# We can see that majority of the variables are of int64 type and are non-
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  num       303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

There are 14 features in Dataset

1. age: The person's age in years

2. sex: The person's sex (1 = male, 0 = female)

3. cp: The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic)

4. trestbps: The person's resting blood pressure (mm Hg on admission to the hospital)

5. chol: The person's cholesterol measurement in mg/dl

6. fbs: The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)

7. restecg: Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)

8. thalach: The person's maximum heart rate achieved

9. exang: Exercise induced angina (1 = yes; 0 = no)

10. oldpeak: ST depression induced by exercise relative to rest

11)slope: the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)

12)ca: The number of major vessels (0-3)

13)thal: A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

14)num: Heart disease (0 = no, 1 = yes)

In [289…

```
# On closer analysis of the dataset it is visible that there are some attr
# but they are categorical variables having a specific number of classes.
```

In [290…

```
df.shape

# The dataset contains 303 records and 14 different attributes / variables
```

Out[290…

```
(303, 14)
```

In [291…

```
df.describe()

# The describe() function gives the statistical summary of the numberical
```

Out[291…

	age	sex	cp	trestbps	chol	fbs	res
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.00
mean	54.438944	0.679868	3.158416	131.689769	246.693069	0.148515	0.99
std	9.038662	0.467299	0.960126	17.599748	51.776918	0.356198	0.99

	age	sex	cp	trestbps	chol	fbs	res
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.00
25%	48.000000	0.000000	3.000000	120.000000	211.000000	0.000000	0.00
50%	56.000000	1.000000	3.000000	130.000000	241.000000	0.000000	1.00
75%	61.000000	1.000000	4.000000	140.000000	275.000000	0.000000	2.00
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.00

Percentage of people having Heart Disease

In [292…

```
num=df.groupby('num').size()
num
```

Out[292…

```
num
0    164
1    139
dtype: int64
```

In [293…

```
def heart_d(r):
    if r==0:
        return 'Absence'
    elif r==1:
        return 'Presence'
```

In [294…

```
#Applying converted data into our dataset with new column - Heart_Disease

df['Heart_Disease']=df['num'].apply(heart_d)
df.head()
```

Out[294…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	n
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	

In [295…

```
hd=df.groupby('Heart_Disease')['num'].count()
hd
```

Out[295…

```
Heart_Disease
Absence     164
Presence    139
Name: num, dtype: int64
```

In [296…

```
#Converting Numerical Data into Categorical Data

def gen(r):
    if r==1:
        return 'Male'
    elif r==0:
        return 'Female'
```

In [297…

```
#Applying converted data into our dataset with new column - sex1

df['sex1']=df['sex'].apply(gen)
df.head()
```

Out[297…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	n
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	

In [298…

```
#Converting Numerical Data into Categorical Data

def age_rng(r):
    if r>=29 and r<40:
        return 'Young Age'
    elif r>=40 and r<55:
        return 'Middle Age'
    elif r>55:
        return 'Elder Age'
```

In [299…

```
#Applying converted data into our dataset with new column - Age_Range

df['Age_Range']=df['age'].apply(age_rng)
df.head()
```

Out[299…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	n
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	

Exploratory Data Analysis

In [300…

```
!pip install pandas-profiling
```

```
Requirement already satisfied: pandas-profiling in c:\users\gigabyte\anaco
nda3\lib\site-packages (3.1.0)
Requirement already satisfied: tqdm>=4.48.2 in c:\users\gigabyte\anaconda3
\lib\site-packages (from pandas-profiling) (4.59.0)
```

```
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.2
5.3 in c:\users\gigabyte\anaconda3\lib\site-packages (from pandas-profilin
g) (1.2.4)
Requirement already satisfied: htmlmin>=0.1.12 in c:\users\gigabyte\anacon
da3\lib\site-packages (from pandas-profiling) (0.1.12)
Requirement already satisfied: pydantic>=1.8.1 in c:\users\gigabyte\anacon
da3\lib\site-packages (from pandas-profiling) (1.9.0)
Requirement already satisfied: visions[type_image_path]==0.7.4 in c:\users
\gigabyte\anaconda3\lib\site-packages (from pandas-profiling) (0.7.4)
Requirement already satisfied: tangled-up-in-unicode==0.1.0 in c:\users\gi
gabyte\anaconda3\lib\site-packages (from pandas-profiling) (0.1.0)
Requirement already satisfied: requests>=2.24.0 in c:\users\gigabyte\anaco
nda3\lib\site-packages (from pandas-profiling) (2.27.1)
Requirement already satisfied: multimethod>=1.4 in c:\users\gigabyte\anaco
nda3\lib\site-packages (from pandas-profiling) (1.6)
Requirement already satisfied: scipy>=1.4.1 in c:\users\gigabyte\anaconda3
\lib\site-packages (from pandas-profiling) (1.6.2)
Requirement already satisfied: jinja2>=2.11.1 in c:\users\gigabyte\anacond
a3\lib\site-packages (from pandas-profiling) (3.0.3)
Requirement already satisfied: markupsafe~=2.0.1 in c:\users\gigabyte\anac
onda3\lib\site-packages (from pandas-profiling) (2.0.1)
Requirement already satisfied: joblib~=1.0.1 in c:\users\gigabyte\anaconda
3\lib\site-packages (from pandas-profiling) (1.0.1)
Requirement already satisfied: PyYAML>=5.0.0 in c:\users\gigabyte\anaconda
3\lib\site-packages (from pandas-profiling) (5.4.1)
Requirement already satisfied: missingno>=0.4.2 in c:\users\gigabyte\anaco
nda3\lib\site-packages (from pandas-profiling) (0.5.0)
Requirement already satisfied: seaborn>=0.10.1 in c:\users\gigabyte\anacon
da3\lib\site-packages (from pandas-profiling) (0.11.1)
Requirement already satisfied: numpy>=1.16.0 in c:\users\gigabyte\anaconda
3\lib\site-packages (from pandas-profiling) (1.22.1)
Requirement already satisfied: phik>=0.11.1 in c:\users\gigabyte\anaconda3
\lib\site-packages (from pandas-profiling) (0.12.0)
Requirement already satisfied: matplotlib>=3.2.0 in c:\users\gigabyte\anac
onda3\lib\site-packages (from pandas-profiling) (3.3.4)
Requirement already satisfied: attrs>=19.3.0 in c:\users\gigabyte\anaconda
3\lib\site-packages (from visions[type_image_path]==0.7.4->pandas-profilin
g) (20.3.0)
Requirement already satisfied: networkx>=2.4 in c:\users\gigabyte\anaconda
3\lib\site-packages (from visions[type_image_path]==0.7.4->pandas-profilin
g) (2.5)
Requirement already satisfied: Pillow in c:\users\gigabyte\anaconda3\lib\s
ite-packages (from visions[type_image_path]==0.7.4->pandas-profiling) (6.
2.1)
Requirement already satisfied: imagehash in c:\users\gigabyte\anaconda3\li
b\site-packages (from visions[type_image_path]==0.7.4->pandas-profiling)
(4.2.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
c:\users\gigabyte\anaconda3\lib\site-packages (from matplotlib>=3.2.0->pan
das-profiling) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\gigabyte\a
naconda3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (2.
8.1)
Requirement already satisfied: cycler>=0.10 in c:\users\gigabyte\anaconda3
\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\gigabyte\anac
onda3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (1.3.1)
Requirement already satisfied: six in c:\users\gigabyte\anaconda3\lib\site
-packages (from cycler>=0.10->matplotlib>=3.2.0->pandas-profiling) (1.13.
0)
Requirement already satisfied: decorator>=4.3.0 in c:\users\gigabyte\anaco
nda3\lib\site-packages (from networkx>=2.4->visions[type_image_path]==0.7.
4->pandas-profiling) (5.0.6)
Requirement already satisfied: pytz>=2017.3 in c:\users\gigabyte\anaconda3
```

```
\lib\site-packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3->p
andas-profiling) (2021.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\giga
byte\anaconda3\lib\site-packages (from pydantic>=1.8.1->pandas-profiling)
(3.7.4.3)
Requirement already satisfied: idna<4,>=2.5 in c:\users\gigabyte\anaconda3
\lib\site-packages (from requests>=2.24.0->pandas-profiling) (2.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\gigab
yte\anaconda3\lib\site-packages (from requests>=2.24.0->pandas-profiling)
(2.0.9)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\gigabyte
\anaconda3\lib\site-packages (from requests>=2.24.0->pandas-profiling) (1.
24.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\gigabyte\ana
conda3\lib\site-packages (from requests>=2.24.0->pandas-profiling) (2019.
9.11)
Requirement already satisfied: PyWavelets in c:\users\gigabyte\anaconda3\l
ib\site-packages (from imagehash->visions[type_image_path]==0.7.4->pandas-
profiling) (1.1.1)
```

In [301…

```python
from pandas_profiling import ProfileReport
```

In [302…

```python
prof=ProfileReport(df,title="Heart_Dataset_Profile_Report_Before_Cleanup.h
```

In [303…

```python
prof
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[303…

Data Transformation

In [304…

```python
# We calculate the median and mode of ca and thal columns to be replaced i
# From below calculation, we come to know that median and mode for both 'c
# So we will replace the -100000 with 0.0 in 'ca' and 3.0 in 'thal'
```

In [305…

```python
df.median()
```

Out[305…

```
age          56.0
sex           1.0
cp            3.0
trestbps    130.0
chol        241.0
fbs           0.0
restecg       1.0
thalach     153.0
exang         0.0
oldpeak       0.8
slope         2.0
ca            0.0
thal          3.0
num           0.0
dtype: float64
```

In [306…

```
# replacing 0 in 'ca' where value = -100000

ca_median = int(df['ca'].median())
df.loc[df.ca == -100000, 'ca'] = np.nan
df.fillna(ca_median,inplace=True)
```

In [307…

```
df['ca']
```

Out[307…

```
0      0.0
1      3.0
2      2.0
3      0.0
4      0.0
      ...
298    0.0
299    2.0
300    1.0
301    1.0
302    0.0
Name: ca, Length: 303, dtype: float64
```

In [308…

```
# Replacing 3 in 'thal' where value = -100000

thal_median = int(df['thal'].median())
df.loc[df.thal == -100000, 'thal'] = np.nan
df.fillna(thal_median,inplace=True)
```

In [309…

```
df['thal']
```

Out[309…

```
0      6.0
1      3.0
2      7.0
3      3.0
4      3.0
      ...
298    7.0
299    7.0
300    7.0
301    3.0
302    3.0
Name: thal, Length: 303, dtype: float64
```

In [310…

```
#Checking Data Types
```

In [311…

```
df.dtypes
```

Out[311…

```
age                   int64
sex                   int64
cp                    int64
trestbps              int64
chol                  int64
fbs                   int64
restecg               int64
thalach               int64
exang                 int64
oldpeak             float64
```

```
slope               int64
ca                float64
thal              float64
num                 int64
Heart_Disease      object
sex1               object
Age_Range          object
dtype: object
```

In [312…

```
# Converting the numeric columns to categorical

df = df.astype({"sex":'category',
                                "cp":'category',
                                "fbs":'category',
                                "restecg":'category',
                                "exang":'category',
                                "slope":'category',
                                "ca":'category',
                                "thal":'category',
                                "num":'category'})
```

In [313…

```
df.dtypes
```

Out[313…

```
age                   int64
sex                category
cp                 category
trestbps              int64
chol                  int64
fbs                category
restecg            category
thalach               int64
exang              category
oldpeak             float64
slope              category
ca                 category
thal               category
num                category
Heart_Disease        object
sex1                 object
Age_Range            object
dtype: object
```

In [314…

```
df
```

Out[314…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0
1	67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0
2	67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0
3	37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0
4	41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	45	1	1	110	264	0	0	132	0	1.2	2	0.0	7.0
299	68	1	4	144	193	1	0	141	0	3.4	2	2.0	7.0
300	57	1	4	130	131	0	0	115	1	1.2	2	1.0	7.0

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
301	57	0	2	130	236	0	2	174	0	0.0	2	1.0	3.0
302	38	1	3	138	175	0	0	173	0	0.0	1	0.0	3.0

303 rows × 17 columns

In [315…

```
df.dtypes[df.dtypes=='category']
```

Out[315…

```
sex         category
cp          category
fbs         category
restecg     category
exang       category
slope       category
ca          category
thal        category
num         category
dtype: object
```

In [316…

```
df.describe()
# After changing datatypes, only numeric values are reflected in summary b
```

Out[316…

	age	trestbps	chol	thalach	oldpeak
count	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.438944	131.689769	246.693069	149.607261	1.039604
std	9.038662	17.599748	51.776918	22.875003	1.161075
min	29.000000	94.000000	126.000000	71.000000	0.000000
25%	48.000000	120.000000	211.000000	133.500000	0.000000
50%	56.000000	130.000000	241.000000	153.000000	0.800000
75%	61.000000	140.000000	275.000000	166.000000	1.600000
max	77.000000	200.000000	564.000000	202.000000	6.200000

In [317…

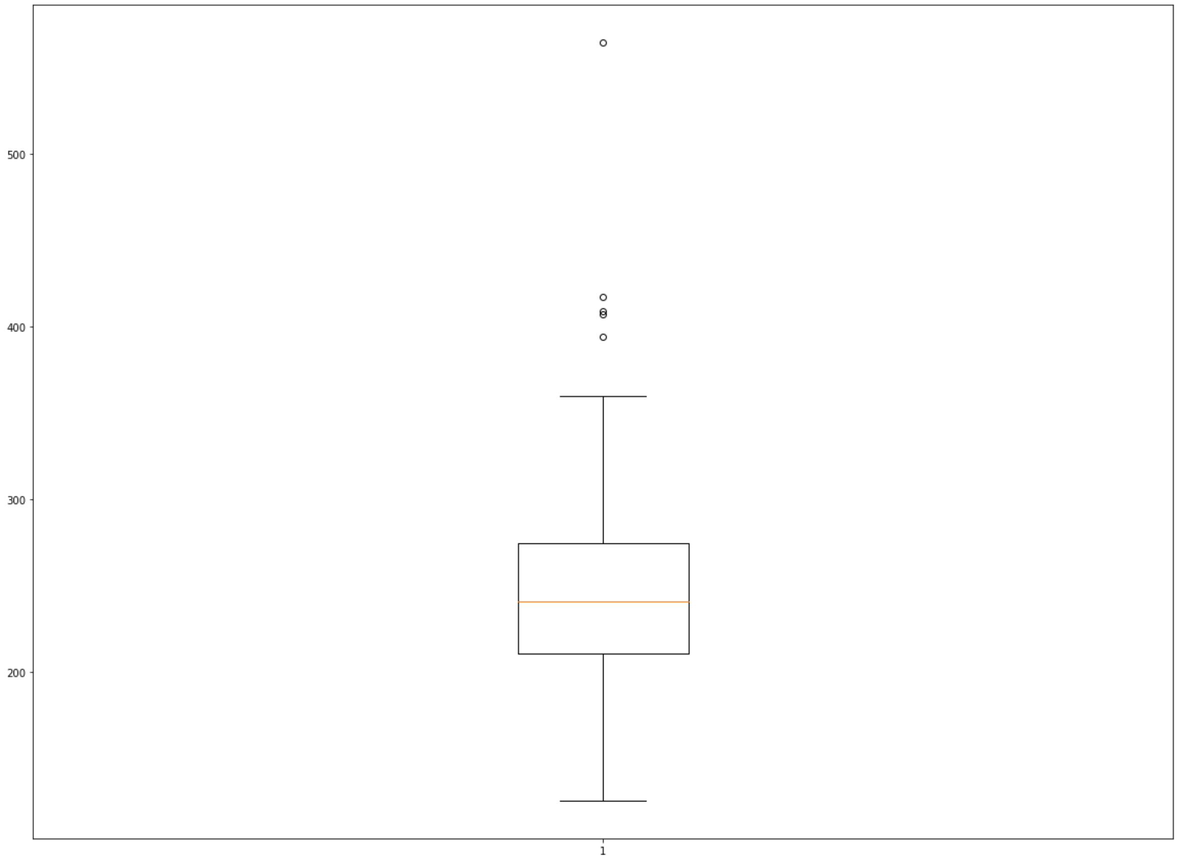
```
# Boxplot before outlier treatment for proper visualization of the outlier
```

In [318…

```
plt.subplots(figsize=(20,15))
plt.boxplot(data=df, x='chol')
```

Out[318…

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a787b2b310>,
  <matplotlib.lines.Line2D at 0x1a787b2a400>],
 'caps': [<matplotlib.lines.Line2D at 0x1a787b2abb0>,
  <matplotlib.lines.Line2D at 0x1a787b2a4c0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a787b2bbe0>],
 'medians': [<matplotlib.lines.Line2D at 0x1a787972310>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a787972b80>],
 'means': []}
```



In [319…

```
df['chol'].mean()
```

Out[319…

```
246.69306930693068
```

In [320…

```
#Detecting Outliers using Inter Quartile Range
#Finding The data located in First Quartile and Third Quartile
#If the data point significantly differs from other cluster of data points
```

In [321…

```
outliers_chol = []
def Find_Outliers(data):
    data = sorted(data)
    Q1 = np.percentile(data,25)
    Q3 = np.percentile(data,75)

    IQR = Q3-Q1
    l_bound = Q1-(1.5*IQR)
    u_bound = Q3+(1.5*IQR)

    for j in data:
        if (j < l_bound or j > u_bound):
            outliers_chol.append(j)
    return outliers_chol

outliers_chol = Find_Outliers(df['chol'])
print("Outliers from IQR method for chol column: ", outliers_chol)
```

```
Outliers from IQR method for chol column:  [394, 407, 409, 417, 564]
```

In [ ]:

```

```

In [322…

```
#Replacing the outliers in the chol column with the mean
```

In [323…

```
for i in outliers_chol:
    df['chol'] = np.where(df['chol'] == i, df['chol'].mean(), df['chol'])
```

In [324…

```
df
```

Out[324…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tha
0	63	1	1	145	233.0	1	2	150	0	2.3	3	0.0	6.
1	67	1	4	160	286.0	0	2	108	1	1.5	2	3.0	3.
2	67	1	4	120	229.0	0	2	129	1	2.6	2	2.0	7.
3	37	1	3	130	250.0	0	0	187	0	3.5	3	0.0	3.
4	41	0	2	130	204.0	0	2	172	0	1.4	1	0.0	3.
...	...	...	...	...	...	...	...	...	...	...	...	...	.
298	45	1	1	110	264.0	0	0	132	0	1.2	2	0.0	7.
299	68	1	4	144	193.0	1	0	141	0	3.4	2	2.0	7.
300	57	1	4	130	131.0	0	0	115	1	1.2	2	1.0	7.
301	57	0	2	130	236.0	0	2	174	0	0.0	2	1.0	3.
302	38	1	3	138	175.0	0	0	173	0	0.0	1	0.0	3.

303 rows × 17 columns

In [ ]:

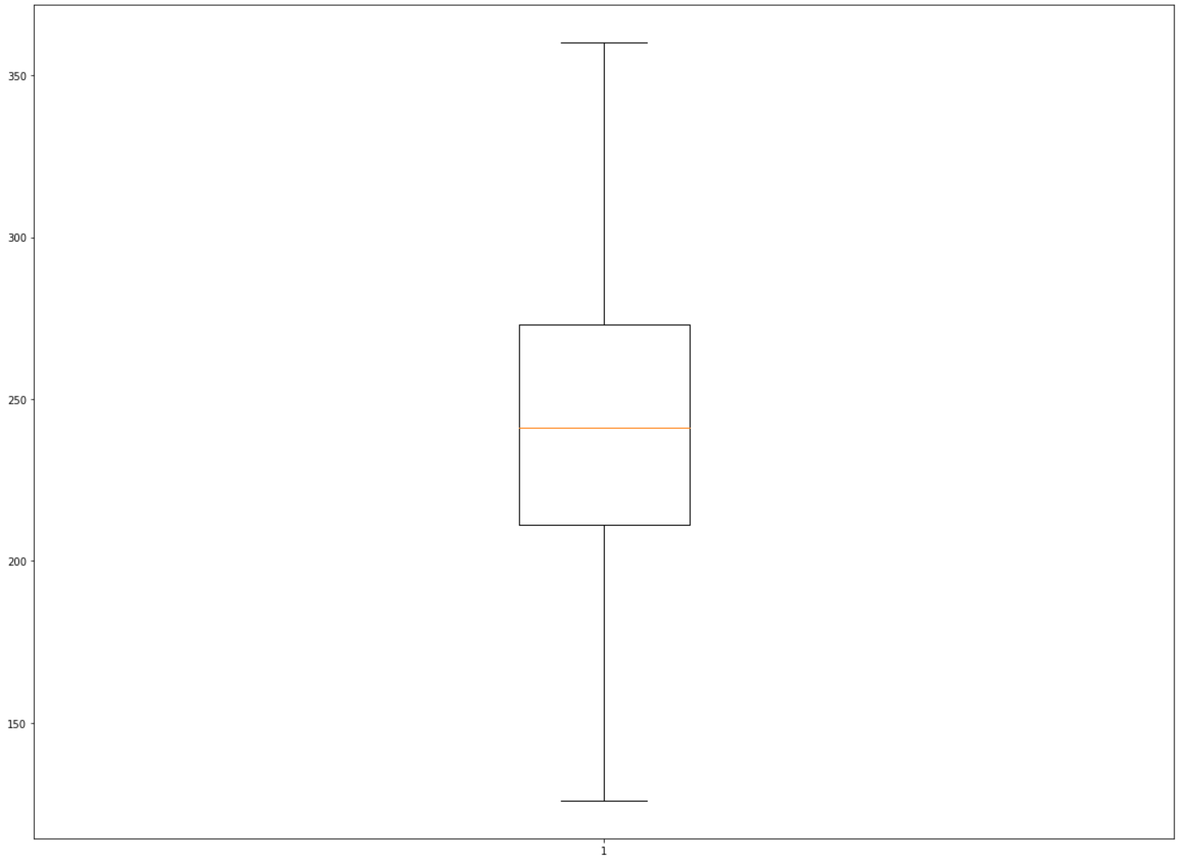
In [325…

```
# Boxplot after removing outlier

plt.subplots(figsize=(20,15))
plt.boxplot(data=df, x='chol')
```

Out[325…

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a7fb0aee80>,
  <matplotlib.lines.Line2D at 0x1a7fb0bf220>],
 'caps': [<matplotlib.lines.Line2D at 0x1a7fb0bf580>,
  <matplotlib.lines.Line2D at 0x1a7fb0bf8e0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a7fb0aeb20>],
 'medians': [<matplotlib.lines.Line2D at 0x1a7fb0bfc40>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a7fb0bffa0>],
 'means': []}
```



In [ ]:

In [326…

```
# Replacing zeros with mean in 'oldpeak' column
```

In [327…

```
df['oldpeak']
```

Out[327…

```
0      2.3
1      1.5
2      2.6
3      3.5
4      1.4
      ...
298    1.2
299    3.4
300    1.2
301    0.0
302    0.0
Name: oldpeak, Length: 303, dtype: float64
```

In [328…

```
df['oldpeak'] = np.where(df['oldpeak'] == 0, df['oldpeak'].mean(), df['old
```

In [329…

```
# Checking the oldpeak column after replacing zeros with mean
```

In [330…

```
df['oldpeak']
```

Out[330…

```
0      2.300000
1      1.500000
2      2.600000
3      3.500000
```

```
4        1.400000
           ...
298      1.200000
299      3.400000
300      1.200000
301      1.039604
302      1.039604
Name: oldpeak, Length: 303, dtype: float64
```

In [331…

```
#checking if there are any remaining null values
```

In [332…

```
df['oldpeak'].isna().count()

# now we can see there are no zero value in oldpeak column
```

Out[332…

```
303
```

In [333…

```
# If we observe the oldpeak distribution it is skewed
# So we perform Log transformation to remove skewness from 'oldpeak' colum
```

In [334…

```
oldpeak_log = np.log(df['oldpeak'])
oldpeak_log
```

Out[334…

```
0        0.832909
1        0.405465
2        0.955511
3        1.252763
4        0.336472
           ...
298      0.182322
299      1.223775
300      0.182322
301      0.038840
302      0.038840
Name: oldpeak, Length: 303, dtype: float64
```

Viewing the cleaned data set

In [335…

```
df
```

Out[335…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	1	1	145	233.0	1	2	150	0	2.300000	3	0.0	6
1	67	1	4	160	286.0	0	2	108	1	1.500000	2	3.0	3
2	67	1	4	120	229.0	0	2	129	1	2.600000	2	2.0	7
3	37	1	3	130	250.0	0	0	187	0	3.500000	3	0.0	3
4	41	0	2	130	204.0	0	2	172	0	1.400000	1	0.0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	45	1	1	110	264.0	0	0	132	0	1.200000	2	0.0	7
299	68	1	4	144	193.0	1	0	141	0	3.400000	2	2.0	7

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
300	57	1	4	130	131.0	0	0	115	1	1.200000	2	1.0	7
301	57	0	2	130	236.0	0	2	174	0	1.039604	2	1.0	3
302	38	1	3	138	175.0	0	0	173	0	1.039604	1	0.0	3

303 rows × 17 columns

In [336…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 17 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   age            303 non-null    int64
 1   sex            303 non-null    category
 2   cp             303 non-null    category
 3   trestbps       303 non-null    int64
 4   chol           303 non-null    float64
 5   fbs            303 non-null    category
 6   restecg        303 non-null    category
 7   thalach        303 non-null    int64
 8   exang          303 non-null    category
 9   oldpeak        303 non-null    float64
 10  slope          303 non-null    category
 11  ca             303 non-null    category
 12  thal           303 non-null    category
 13  num            303 non-null    category
 14  Heart_Disease  303 non-null    object
 15  sex1           303 non-null    object
 16  Age_Range      303 non-null    object
dtypes: category(9), float64(2), int64(3), object(3)
memory usage: 23.0+ KB
```

In [337…

```
df.describe()
```

Out[337…

	age	trestbps	chol	thalach	oldpeak
count	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.438944	131.689769	243.515787	149.607261	1.379277
std	9.038662	17.599748	44.689381	22.875003	0.937039
min	29.000000	94.000000	126.000000	71.000000	0.100000
25%	48.000000	120.000000	211.000000	133.500000	1.000000
50%	56.000000	130.000000	241.000000	153.000000	1.039604
75%	61.000000	140.000000	273.000000	166.000000	1.600000
max	77.000000	200.000000	360.000000	202.000000	6.200000

In [338…

```
for x in df.dtypes[df.dtypes=='category'].index:
    print(x+":")
    print(pd.Categorical(df[x]))
    print()
```

```
sex:
[1, 1, 1, 1, 0, ..., 1, 1, 1, 0, 1]
Length: 303
Categories (2, int64): [0, 1]

cp:
[1, 4, 4, 3, 2, ..., 1, 4, 4, 2, 3]
Length: 303
Categories (4, int64): [1, 2, 3, 4]

fbs:
[1, 0, 0, 0, 0, ..., 0, 1, 0, 0, 0]
Length: 303
Categories (2, int64): [0, 1]

restecg:
[2, 2, 2, 0, 2, ..., 0, 0, 0, 2, 0]
Length: 303
Categories (3, int64): [0, 1, 2]

exang:
[0, 1, 1, 0, 0, ..., 0, 0, 1, 0, 0]
Length: 303
Categories (2, int64): [0, 1]

slope:
[3, 2, 2, 3, 1, ..., 2, 2, 2, 2, 1]
Length: 303
Categories (3, int64): [1, 2, 3]

ca:
[0.0, 3.0, 2.0, 0.0, 0.0, ..., 0.0, 2.0, 1.0, 1.0, 0.0]
Length: 303
Categories (4, float64): [0.0, 1.0, 2.0, 3.0]

thal:
[6.0, 3.0, 7.0, 3.0, 3.0, ..., 7.0, 7.0, 7.0, 3.0, 3.0]
Length: 303
Categories (3, float64): [3.0, 6.0, 7.0]

num:
[0, 1, 1, 0, 0, ..., 1, 1, 1, 1, 0]
Length: 303
Categories (2, int64): [0, 1]

C:\Users\Gigabyte\anaconda3\lib\site-packages\pandas\io\formats\format.py:
1405: FutureWarning: Index.ravel returning ndarray is deprecated; in a fut
ure version this will return a view on self.
  for val, m in zip(values.ravel(), mask.ravel())
C:\Users\Gigabyte\anaconda3\lib\site-packages\pandas\io\formats\format.py:
1405: FutureWarning: Index.ravel returning ndarray is deprecated; in a fut
ure version this will return a view on self.
  for val, m in zip(values.ravel(), mask.ravel())
```

In [339…

```python
# Summary for categorical variables
df[df.dtypes[df.dtypes=='category'].index].describe()
```

Out[339…

	sex	cp	fbs	restecg	exang	slope	ca	thal	num
count	303	303	303	303	303	303	303.0	303.0	303
unique	2	4	2	3	2	3	4.0	3.0	2

	sex	cp	fbs	restecg	exang	slope	ca	thal	num
top	1	4	0	0	0	1	0.0	3.0	0
freq	206	144	258	151	204	142	180.0	168.0	164

In [340…

```
df.dtypes
```

Out[340…

```
age                  int64
sex               category
cp                category
trestbps             int64
chol               float64
fbs               category
restecg           category
thalach              int64
exang             category
oldpeak            float64
slope             category
ca                category
thal              category
num               category
Heart_Disease       object
sex1                object
Age_Range           object
dtype: object
```

In [ ]:

```

```

EDA after cleaning the data

In [341…

```
prof=ProfileReport(df,title="Heart_Dataset_Profile_Report_Before_Cleanup.h
```

In [342…

```
prof
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[342…

Exporting the Preprocessed dataset into a csv file for further analysis

In [343…

```
# Now we will export the preprocessed dataset to a csv file with no row in
# Output file: preprocessed_heart_disease_dataset.csv

df.to_csv('preprocessed_heart_disease_dataset.csv',index = False)
```

In [344…

```
df
```

Out[344…

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	1	1	145	233.0	1	2	150	0	2.300000	3	0.0	6
1	67	1	4	160	286.0	0	2	108	1	1.500000	2	3.0	3
2	67	1	4	120	229.0	0	2	129	1	2.600000	2	2.0	7
3	37	1	3	130	250.0	0	0	187	0	3.500000	3	0.0	3
4	41	0	2	130	204.0	0	2	172	0	1.400000	1	0.0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	45	1	1	110	264.0	0	0	132	0	1.200000	2	0.0	7
299	68	1	4	144	193.0	1	0	141	0	3.400000	2	2.0	7
300	57	1	4	130	131.0	0	0	115	1	1.200000	2	1.0	7
301	57	0	2	130	236.0	0	2	174	0	1.039604	2	1.0	3
302	38	1	3	138	175.0	0	0	173	0	1.039604	1	0.0	3

303 rows × 17 columns

In [ ]: