Importing pndas and loading the DataSet

```python
import pandas as pd
df=pd.read_csv('/content/Heart_Disease_Prediction.csv')
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 270,\n  \"fields\": [\
n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 9,\n        \"min\": 29,\n
\"max\": 77,\n        \"num_unique_values\": 41,\n        \"samples\":
[\n          50,\n          71,\n          60\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Sex\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Chest pain type\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 4,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          3,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"BP\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 17,\n        \"min\": 94,\n
\"max\": 200,\n        \"num_unique_values\": 47,\n
\"samples\": [\n          156,\n          200\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Cholesterol\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
51,\n        \"min\": 126,\n        \"max\": 564,\n
\"num_unique_values\": 144,\n        \"samples\": [\n          255,\n
229\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"FBS
over 120\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"EKG
results\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          2,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Max
HR\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 23,\n        \"min\": 71,\n        \"max\": 202,\n
\"num_unique_values\": 90,\n        \"samples\": [\n          96,\n
139\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Exercise angina\",\n      \"properties\": {\n        \"dtype\":
```

```
\"number\",\n        \"std\": 0,\n         \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"ST depression\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1.1452098393779975,\n
\"min\": 0.0,\n        \"max\": 6.2,\n        \"num_unique_values\":
39,\n        \"samples\": [\n          2.1,\n          3.5\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Slope of ST\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n         \"min\": 1,\n
\"max\": 3,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          2,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Number of vessels fluro\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 3,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          0,\n          2\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Thallium\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n         \"min\": 3,\n
\"max\": 7,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          3,\n          7\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Heart Disease\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"Absence\",\n          \"Presence\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

Displaying Last five rows

```
df.tail()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8,\n        \"min\": 44,\n
\"max\": 67,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          44,\n          67,\n          56\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Sex\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Chest pain type\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 2,\n
\"max\": 4,\n        \"num_unique_values\": 3,\n        \"samples\":
```

```
[\n          3,\n          2\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"BP\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 20,\n        \"min\": 120,\n
\"max\": 172,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          120,\n          160\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Cholesterol\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
48,\n        \"min\": 192,\n        \"max\": 294,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          263,\n
286\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"FBS
over 120\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"EKG
results\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          2,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Max
HR\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 24,\n        \"min\": 108,\n        \"max\": 173,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          173,\n
108\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Exercise angina\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"ST depression\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.6348228099241552,\n
\"min\": 0.0,\n        \"max\": 1.5,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          0.0,\n          1.5\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Slope of ST\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 1,\n        \"max\": 2,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          2,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Number of vessels fluro\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 0,\n
\"max\": 3,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          3,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
```

```
\"column\": \"Thallium\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2,\n        \"min\": 3,\n
\"max\": 7,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          7,\n          3\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Heart Disease\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"Presence\",\n          \"Absence\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

Find No.of rows & columns

```
df.shape
```

```
(270, 14)
```

All the columns

```
for i in df.columns:
  print(i)
print("No.of Columns:",len(df.columns))
```

```
Age
Sex
Chest pain type
BP
Cholesterol
FBS over 120
EKG results
Max HR
Exercise angina
ST depression
Slope of ST
Number of vessels fluro
Thallium
Heart Disease
No.of Columns: 14
```

Get Paties name with Age>=30

```
df.loc[df['Age']>30,'Sex']
```

```
0      1
1      0
2      1
3      1
4      0
      ..
```

```
265    1
266    1
267    0
268    1
269    1
Name: Sex, Length: 269, dtype: int64
```

Data with age>=30 & sex==1

```
df.loc[(df['Age']>=30) & (df['Sex']==1),['BP','Sex']]
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 182,\n  \"fields\": [\
n    {\n      \"column\": \"BP\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 16,\n        \"min\": 94,\n
\"max\": 192,\n        \"num_unique_values\": 41,\n
\"samples\": [\n          156,\n          144,\n          134\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Sex\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\":
1,\n        \"max\": 1,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```

Display Information of DataSet using info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      270 non-null    int64  
 1   Sex                      270 non-null    int64  
 2   Chest pain type          270 non-null    int64  
 3   BP                       270 non-null    int64  
 4   Cholesterol              270 non-null    int64  
 5   FBS over 120             270 non-null    int64  
 6   EKG results              270 non-null    int64  
 7   Max HR                   270 non-null    int64  
 8   Exercise angina          270 non-null    int64  
 9   ST depression            270 non-null    float64
 10  Slope of ST              270 non-null    int64  
 11  Number of vessels fluro  270 non-null    int64  
 12  Thallium                 270 non-null    int64  
 13  Heart Disease            270 non-null    object 
dtypes: float64(1), int64(12), object(1)
memory usage: 29.7+ KB
```

Identify Numerical & Categorical features

```
numerical_features = df.select_dtypes(include=["int64",
"float64"]).columns
print("Numerical Features:")
print(numerical_features)

Numerical Features:
Index(['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over
120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium'],
      dtype='object')

categorical_features = df.select_dtypes(include=["object",
"category"]).columns
print("\nCategorical Features:")
print(categorical_features)


Categorical Features:
Index(['Heart Disease'], dtype='object')
```

DataSet of all Columns

```
df.dtypes

Age                          int64
Sex                          int64
Chest pain type              int64
BP                           int64
Cholesterol                  int64
FBS over 120                 int64
EKG results                  int64
Max HR                       int64
Exercise angina              int64
ST depression              float64
Slope of ST                  int64
Number of vessels fluro      int64
Thallium                     int64
Heart Disease               object
dtype: object
```

Check for Missing values

```
df.isnull().sum()
print("It explain about No.of Missing values in each record")

It explain about No.of Missing values in each record
```

```
  QUESTION 3
```

Display Statistical Analysis

```
print("Statistical SUmmary:(mean/median/mode)")
print(df.describe())

Statistical SUmmary:(mean/median/mode)
              Age         Sex  Chest pain type          BP  
Cholesterol  \
count  270.000000  270.000000       270.000000  270.000000  
270.000000   
mean    54.433333    0.677778         3.174074  131.344444  
249.659259   
std      9.109067    0.468195         0.950090   17.861608   
51.686237   
min     29.000000    0.000000         1.000000   94.000000  
126.000000   
25%     48.000000    0.000000         3.000000  120.000000  
213.000000   
50%     55.000000    1.000000         3.000000  130.000000  
245.000000   
75%     61.000000    1.000000         4.000000  140.000000  
280.000000   
max     77.000000    1.000000         4.000000  200.000000  
564.000000   

       FBS over 120  EKG results      Max HR  Exercise angina  ST 
depression  \
count    270.000000   270.000000  270.000000       270.000000     
270.00000   
mean       0.148148     1.022222  149.677778         0.329630       
1.05000   
std        0.355906     0.997891   23.165717         0.470952       
1.14521   
min        0.000000     0.000000   71.000000         0.000000       
0.00000   
25%        0.000000     0.000000  133.000000         0.000000       
0.00000   
50%        0.000000     2.000000  153.500000         0.000000       
0.80000   
75%        0.000000     2.000000  166.000000         1.000000       
1.60000   
max        1.000000     2.000000  202.000000         1.000000       
6.20000   

       Slope of ST  Number of vessels fluro    Thallium  
count   270.000000               270.000000  270.000000  
mean      1.585185                 0.670370    4.696296  
```

```
std       0.614390                         0.943896    1.940659
min       1.000000                         0.000000    3.000000
25%       1.000000                         0.000000    3.000000
50%       2.000000                         0.000000    3.000000
75%       2.000000                         1.000000    7.000000
max       3.000000                         3.000000    7.000000
```

Calculate mean,median & standard deviation of Age

```
import numpy as np
mean=np.mean(df['Age'])
print("\nmean:",mean)
median=np.median(df['Age'])
print("\nMedian:",median)
std=np.std(df['Age'])
print("Standard Deviation :",std)


mean: 54.43333333333333

Median: 55.0
Standard Deviation : 9.092182234083173
```

Calculate max,min of Cholestrol

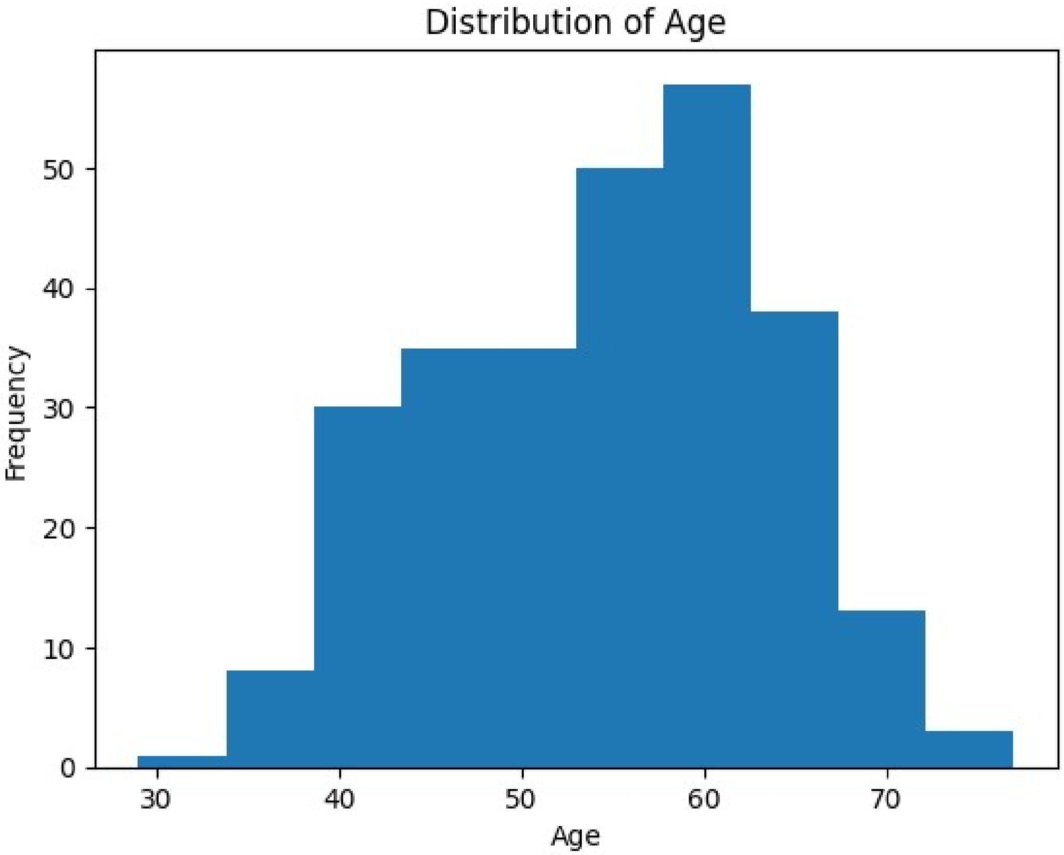
```
print("Max=",np.max(df['Cholesterol']))
print("Min=",np.min(df['Cholesterol']))

Max= 564
Min= 126

  QUESTION 4
```

Count No.of Patients with & without heart Disease

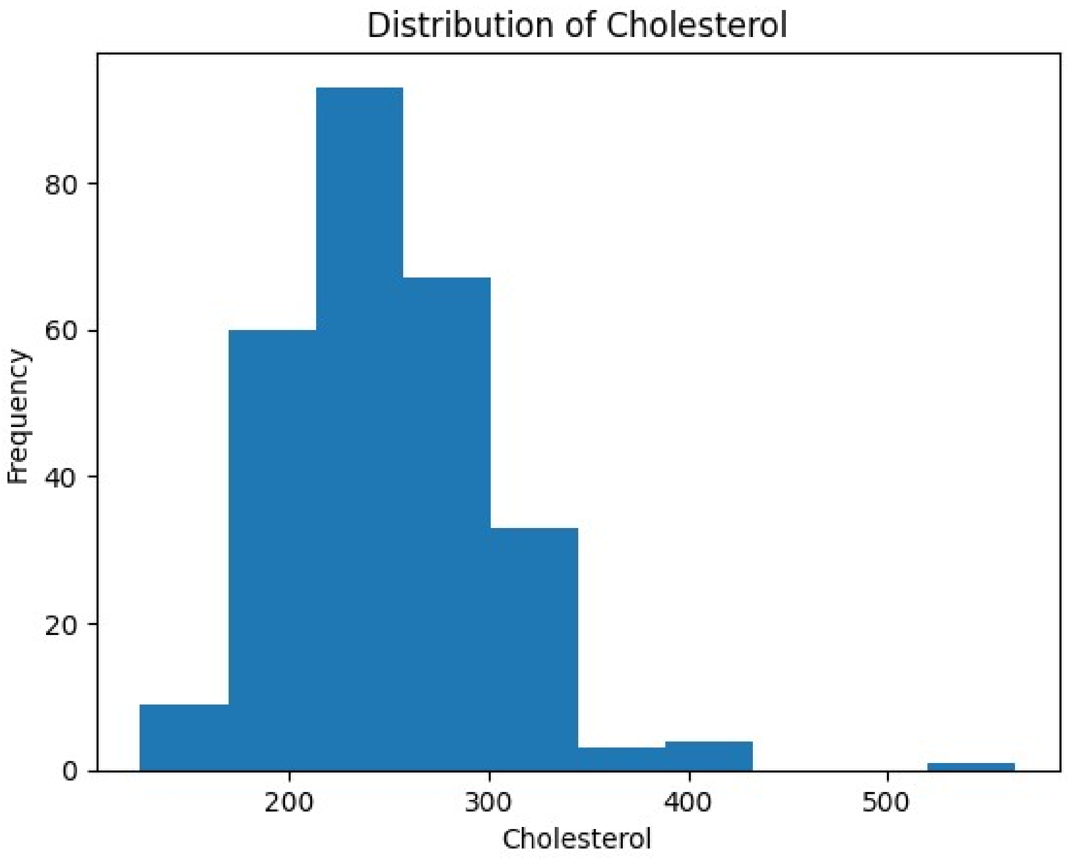
```
print(df['Heart Disease'].value_counts())

Heart Disease
Absence     150
Presence    120
Name: count, dtype: int64
```

Plot Histogram od Age column

```
import matplotlib.pyplot as plt
plt.figure()
plt.hist(df['Age'],bins=10)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.show()
```



```
import matplotlib.pyplot as plt
plt.figure()
plt.hist(df['Cholesterol'],bins=10)
plt.xlabel('Cholesterol')
plt.ylabel('Frequency')
plt.title('Distribution of Cholesterol')
plt.show()
```



QUESTION 5

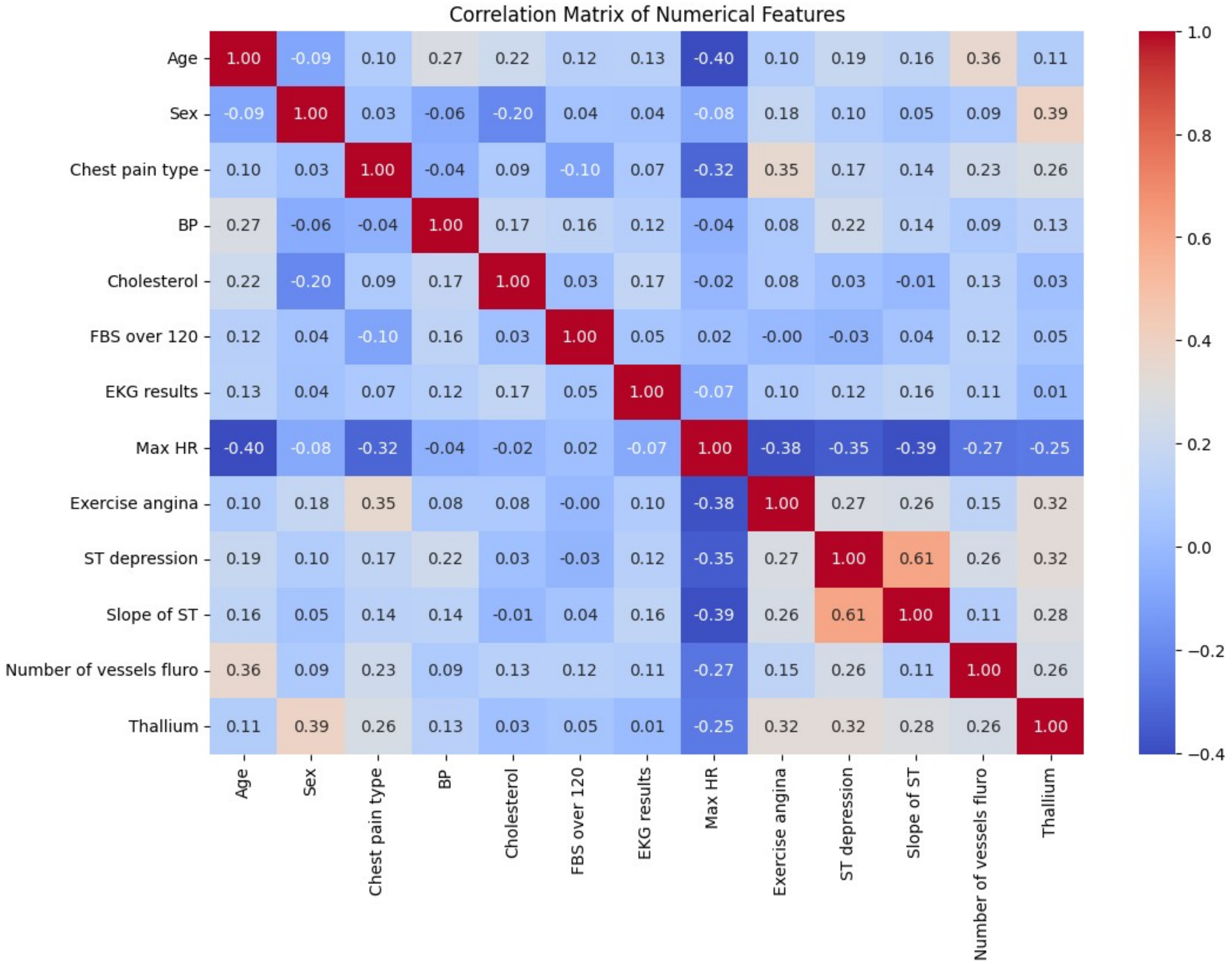
```
# Assignment 2: Relationship and Correlation Analysis
# Heart Disease Dataset

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Load the dataset
df = pd.read_csv("/content/Heart_Disease_Prediction.csv")

# Display first few rows
print("First 5 rows of the dataset:")
print(df.head())

# 2. Identify numerical and categorical features
numerical_features = df.select_dtypes(include=["int64",
"float64"]).columns
categorical_features = df.select_dtypes(include=["object",
"category"]).columns

print("\nNumerical Features:")
```

```python
print(numerical_features)

print("\nCategorical Features:")
print(categorical_features)

# 3. Compute correlation matrix (numerical features only)
corr_matrix = df[numerical_features].corr()

print("\nCorrelation Matrix:")
print(corr_matrix)

# 4. Plot correlation heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix of Numerical Features")
plt.show()

# 5. Identify highly correlated variables (absolute correlation > 0.5)
high_corr = corr_matrix.abs()
high_corr_pairs = high_corr.unstack().sort_values(ascending=False)

# Remove self-correlation
high_corr_pairs = high_corr_pairs[high_corr_pairs < 1]

print("\nHighly Correlated Feature Pairs (|correlation| > 0.5):")
print(high_corr_pairs[high_corr_pairs > 0.5])

# 6. Scatter plot: Age vs Maximum Heart Rate
plt.figure(figsize=(7, 5))
sns.scatterplot(x="Age", y="MaxHR", data=df)
plt.title("Scatter Plot: Age vs Maximum Heart Rate")
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.show()

# 7. Interpretation (printed for reference)
print("\nInterpretation:")
print("- Age and Maximum Heart Rate show a negative correlation.")
print("- As age increases, the maximum heart rate generally
decreases.")
print("- Features like Oldpeak and ExerciseAngina tend to show
positive correlation with HeartDisease.")

print("\nConclusion:")
print("Relationships between medical parameters were explored using
correlation analysis and visualization techniques.")
```

```
First 5 rows of the dataset:
   Age  Sex  Chest pain type   BP  Cholesterol  FBS over 120  EKG
results  \
0   70    1                4  130          322             0
2
1   67    0                3  115          564             0
2
2   57    1                2  124          261             0
0
3   64    1                4  128          263             0
0
4   74    0                2  120          269             0
2

   Max HR  Exercise angina  ST depression  Slope of ST  \
0     109                0            2.4            2
1     160                0            1.6            2
2     141                0            0.3            1
3     105                1            0.2            2
4     121                1            0.2            1

   Number of vessels fluro  Thallium Heart Disease
0                        3         3      Presence
1                        0         7       Absence
2                        0         7      Presence
3                        1         7       Absence
4                        1         3       Absence

Numerical Features:
Index(['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over
120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium'],
      dtype='object')

Categorical Features:
Index(['Heart Disease'], dtype='object')

Correlation Matrix:
                              Age       Sex  Chest pain type        BP
\
Age                      1.000000 -0.094401         0.096920  0.273053

Sex                     -0.094401  1.000000         0.034636 -0.062693

Chest pain type          0.096920  0.034636         1.000000 -0.043196

BP                       0.273053 -0.062693        -0.043196  1.000000

Cholesterol              0.220056 -0.201647         0.090465  0.173019

FBS over 120             0.123458  0.042140        -0.098537  0.155681
```

```
EKG results              0.128171  0.039253        0.074325  0.116157

Max HR                  -0.402215 -0.076101       -0.317682 -0.039136

Exercise angina          0.098297  0.180022        0.353160  0.082793

ST depression            0.194234  0.097412        0.167244  0.222800

Slope of ST              0.159774  0.050545        0.136900  0.142472

Number of vessels fluro  0.356081  0.086830        0.225890  0.085697

Thallium                 0.106100  0.391046        0.262659  0.132045


                         Cholesterol  FBS over 120  EKG results    Max
HR  \
Age                         0.220056      0.123458     0.128171 -
0.402215
Sex                        -0.201647      0.042140     0.039253 -
0.076101
Chest pain type             0.090465     -0.098537     0.074325 -
0.317682
BP                          0.173019      0.155681     0.116157 -
0.039136
Cholesterol                 1.000000      0.025186     0.167652 -
0.018739
FBS over 120                0.025186      1.000000     0.053499
0.022494
EKG results                 0.167652      0.053499     1.000000 -
0.074628
Max HR                     -0.018739      0.022494    -0.074628
1.000000
Exercise angina             0.078243     -0.004107     0.095098 -
0.380719
ST depression               0.027709     -0.025538     0.120034 -
0.349045
Slope of ST                -0.005755      0.044076     0.160614 -
0.386847
Number of vessels fluro     0.126541      0.123774     0.114368 -
0.265333
Thallium                    0.028836      0.049237     0.007337 -
0.253397

                         Exercise angina  ST depression  Slope of
ST  \
Age                             0.098297       0.194234     0.159774

Sex                             0.180022       0.097412     0.050545
```

```
Chest pain type                0.353160    0.167244    0.136900

BP                             0.082793    0.222800    0.142472

Cholesterol                    0.078243    0.027709   -0.005755

FBS over 120                  -0.004107   -0.025538    0.044076

EKG results                    0.095098    0.120034    0.160614

Max HR                        -0.380719   -0.349045   -0.386847

Exercise angina                1.000000    0.274672    0.255908

ST depression                  0.274672    1.000000    0.609712

Slope of ST                    0.255908    0.609712    1.000000

Number of vessels fluro        0.153347    0.255005    0.109498

Thallium                       0.321449    0.324333    0.283678


                         Number of vessels fluro  Thallium
Age                                     0.356081  0.106100
Sex                                     0.086830  0.391046
Chest pain type                         0.225890  0.262659
BP                                      0.085697  0.132045
Cholesterol                             0.126541  0.028836
FBS over 120                            0.123774  0.049237
EKG results                             0.114368  0.007337
Max HR                                 -0.265333 -0.253397
Exercise angina                         0.153347  0.321449
ST depression                           0.255005  0.324333
Slope of ST                             0.109498  0.283678
Number of vessels fluro                 1.000000  0.255648
Thallium                                0.255648  1.000000
```



```
Highly Correlated Feature Pairs (|correlation| > 0.5):
ST depression  Slope of ST      0.609712
Slope of ST    ST depression    0.609712
dtype: float64

----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
/tmp/ipython-input-3314100734.py in <cell line: 0>()
     47 # 6. Scatter plot: Age vs Maximum Heart Rate
     48 plt.figure(figsize=(7, 5))
---> 49 sns.scatterplot(x="Age", y="MaxHR", data=df)
     50 plt.title("Scatter Plot: Age vs Maximum Heart Rate")
     51 plt.xlabel("Age")

/usr/local/lib/python3.12/dist-packages/seaborn/relational.py in
scatterplot(data, x, y, hue, size, style, palette, hue_order,
hue_norm, sizes, size_order, size_norm, markers, style_order, legend,
ax, **kwargs)
    613 ):
    614
--> 615     p = _ScatterPlotter(
    616         data=data,
    617         variables=dict(x=x, y=y, hue=hue, size=size,
style=style),

/usr/local/lib/python3.12/dist-packages/seaborn/relational.py in
__init__(self, data, variables, legend)
    394         )
    395
--> 396         super().__init__(data=data, variables=variables)
    397
    398         self.legend = legend

/usr/local/lib/python3.12/dist-packages/seaborn/_base.py in
__init__(self, data, variables)
    632         # information for numeric axes would be information
about log scales.
    633         self._var_ordered = {"x": False, "y": False}  # alt.,
used DefaultDict
--> 634         self.assign_variables(data, variables)
    635
    636         # TODO Lots of tests assume that these are called to
initialize the

/usr/local/lib/python3.12/dist-packages/seaborn/_base.py in
assign_variables(self, data, variables)
    677             # to centralize / standardize data consumption
logic.
    678             self.input_format = "long"
--> 679             plot_data = PlotData(data, variables)
    680             frame = plot_data.frame
    681             names = plot_data.names

/usr/local/lib/python3.12/dist-packages/seaborn/_core/data.py in
__init__(self, data, variables)
     56
     57         data = handle_data_source(data)
---> 58         frame, names, ids = self._assign_variables(data,
variables)
     59
     60         self.frame = frame

/usr/local/lib/python3.12/dist-packages/seaborn/_core/data.py in
_assign_variables(self, data, variables)
    230                 else:
    231                     err += "An entry with this name does not
appear in `data`."
```

```
--> 232                 raise ValueError(err)
    233
    234             else:

ValueError: Could not interpret value `MaxHR` for `y`. An entry with
this name does not appear in `data`.

<Figure size 700x500 with 0 Axes>
```