Mini-Project 1 Ryan Neighbor ANA 500

import matplotlib.pyplot as plt import pandas as pd import numpy as np !pip install openpyxl

In [55]:
```
# 1a Acquire - identify data sets, retrieve data, query data
# Designate File Path
file_path = r"C:\Users\Nanoo\OneDrive\Desktop\ANA 500\heart disease.xlsx"

# Use read_excel for .xlsx files
df = pd.read_excel(file_path)
```

In [57]:
```
# 1b Acquire - identify data sets, retrieve data, query data
# Check First 5 entrys
df.head()
```

Out[57]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	Exer
0	40	M	ATA	140	289	0	Normal	172	
1	49	F	NAP	160	180	0	Normal	156	
2	37	M	ATA	130	283	0	ST	98	
3	48	F	ASY	138	214	0	Normal	108	
4	54	M	NAP	150	195	0	Normal	122	

In [59]:
```
# 1c Acquire - identify data sets, retrieve data, query data
# Missing Value Check
df.isnull().sum()
```

Out[59]:
```
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```

In [61]:
```
# 1d Acquire - identify data sets, retrieve data, query data
# DF Info Check
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             918 non-null    int64
 1   Sex             918 non-null    object
 2   ChestPainType   918 non-null    object
 3   RestingBP       918 non-null    int64
 4   Cholesterol     918 non-null    int64
 5   FastingBS       918 non-null    int64
 6   RestingECG      918 non-null    object
 7   MaxHR           918 non-null    int64
 8   ExerciseAngina  918 non-null    object
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object
 11  HeartDisease    918 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

In [69]:

```python
# 1e Acquire – identify data sets, retrieve data, query data
# Dissect Categorical Variable Data Types by looping thhrough each Column and Print
categorical_cols = ['Sex', 'ChestPainType', 'RestingECG', 'ExerciseAngina', 'ST_Slo

for col in categorical_cols:
    print(f"\nColumn: {col}")
    print(df[col].value_counts())
```

```
Column: Sex
Sex
M    725
F    193
Name: count, dtype: int64

Column: ChestPainType
ChestPainType
ASY    496
NAP    203
ATA    173
TA      46
Name: count, dtype: int64

Column: RestingECG
RestingECG
Normal    552
LVH       188
ST        178
Name: count, dtype: int64

Column: ExerciseAngina
ExerciseAngina
N    547
Y    371
Name: count, dtype: int64

Column: ST_Slope
ST_Slope
Flat    460
Up      395
Down     63
Name: count, dtype: int64
```

In [71]:

```python
# 1e Acquire – identify data sets, retrieve data, query data
# Dissect Categorical Variable Data Types (Autodetect Method)
categorical_cols = df.select_dtypes(include=['object', 'category']).columns

for col in categorical_cols:
    print(f"\nColumn: {col}")
    print(df[col].value_counts())
```

```
Column: Sex
Sex
M    725
F    193
Name: count, dtype: int64

Column: ChestPainType
ChestPainType
ASY    496
NAP    203
ATA    173
TA      46
Name: count, dtype: int64

Column: RestingECG
RestingECG
Normal    552
LVH       188
ST        178
Name: count, dtype: int64

Column: ExerciseAngina
ExerciseAngina
N    547
Y    371
Name: count, dtype: int64

Column: ST_Slope
ST_Slope
Flat    460
Up      395
Down     63
Name: count, dtype: int64
```

In [73]:

```
# 1f Acquire – identify data sets, retrieve data, query data

# List View of Columns/Variables for better Understanding
print(df.columns.tolist())
```

```
['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol', 'FastingBS', 'RestingEC
G', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope', 'HeartDisease']
```

In [77]:

```
# 2a Prepare – explore and pre-process

# Summary Statistics
df.describe
```

Out[77]:
```
<bound method NDFrame.describe of      Age Sex ChestPainType  RestingBP  Cholester
ol  FastingBS RestingECG  \
0     40   M           ATA        140          289          0     Normal
1     49   F           NAP        160          180          0     Normal
2     37   M           ATA        130          283          0         ST
3     48   F           ASY        138          214          0     Normal
4     54   M           NAP        150          195          0     Normal
..   ...  ..           ...        ...          ...        ...        ...
913   45   M            TA        110          264          0     Normal
914   68   M           ASY        144          193          1     Normal
915   57   M           ASY        130          131          0     Normal
916   57   F           ATA        130          236          0        LVH
917   38   M           NAP        138          175          0     Normal

     MaxHR ExerciseAngina  Oldpeak ST_Slope  HeartDisease
0      172              N      0.0       Up             0
1      156              N      1.0     Flat             1
2       98              N      0.0       Up             0
3      108              Y      1.5     Flat             1
4      122              N      0.0       Up             0
..     ...            ...      ...      ...           ...
913    132              N      1.2     Flat             1
914    141              N      3.4     Flat             1
915    115              Y      1.2     Flat             1
916    174              N      0.0     Flat             1
917    173              N      0.0       Up             0

[918 rows x 12 columns]>
```

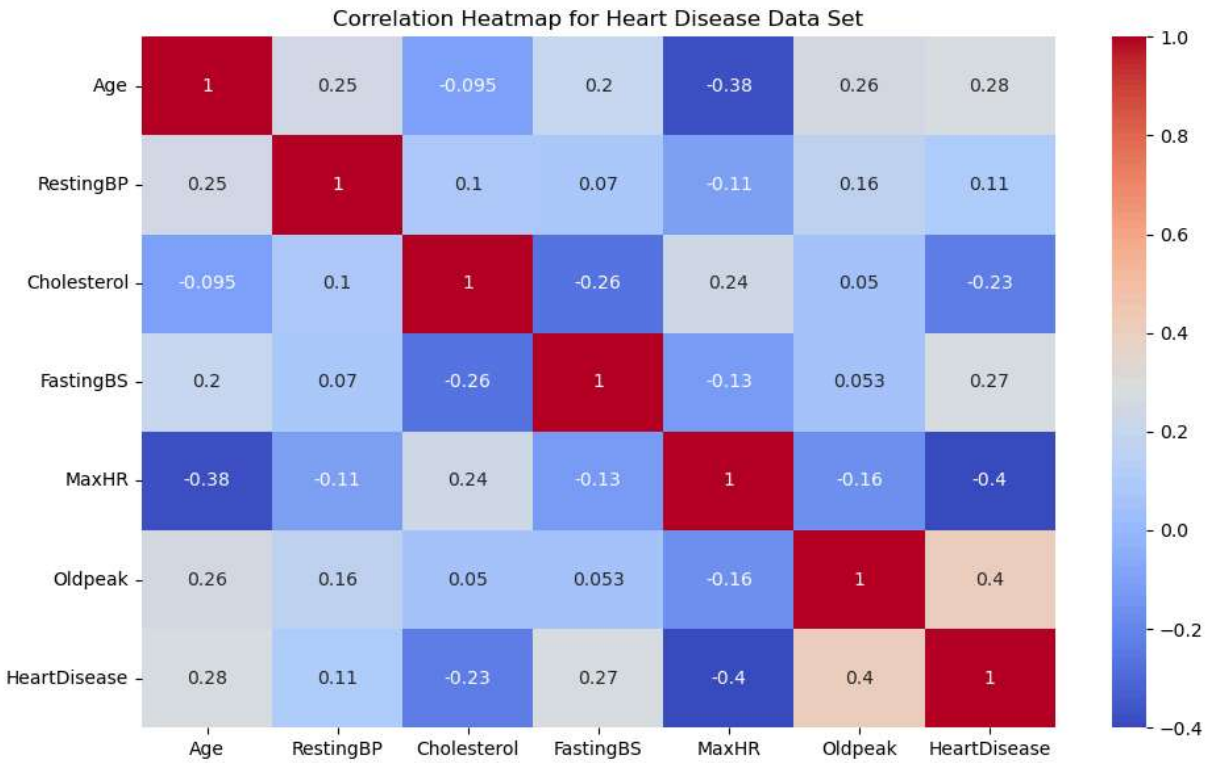
In [81]:
```python
# 2b Prepare – explore and pre-process

# Summary Statistics
df[['Age','Cholesterol','MaxHR', 'FastingBS', 'Oldpeak']].describe()
```

Out[81]:

	Age	Cholesterol	MaxHR	FastingBS	Oldpeak
count	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	198.799564	136.809368	0.233115	0.887364
std	9.432617	109.384145	25.460334	0.423046	1.066570
min	28.000000	0.000000	60.000000	0.000000	-2.600000
25%	47.000000	173.250000	120.000000	0.000000	0.000000
50%	54.000000	223.000000	138.000000	0.000000	0.600000
75%	60.000000	267.000000	156.000000	0.000000	1.500000
max	77.000000	603.000000	202.000000	1.000000	6.200000

In [85]:
```python
# 2c Prepare – explore and pre-process
import seaborn as sns
import matplotlib.pyplot as plt
```

```
#Correlation Heatmap for Continuous Data
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap for Heart Disease Data Set")
plt.tight_layout()
plt.show()
```



In [87]:
```
# 2d Prepare – explore and pre-process

# Exploratory Filtering
df_filtered = df[(df['Age'] > 50) & (df['Cholesterol'] < 200)]
```

In [89]:
```
# 2e Prepare – explore and pre-process
# Summary Statistics
df_filtered[['Age','Cholesterol']].describe()
```

Out[89]:

	Age	Cholesterol
count	222.000000	222.000000
mean	59.243243	65.504505
std	5.823523	85.598862
min	51.000000	0.000000
25%	55.000000	0.000000
50%	59.000000	0.000000
75%	63.000000	170.750000
max	77.000000	199.000000

In [95]:

```
# 2f Prepare – explore and pre-process

# Normalize Continuous Values
# List of numeric columns to normalize
numeric_cols = ['Age', 'Cholesterol', 'MaxHR', 'FastingBS', 'Oldpeak']

# Apply normalization
df_normalized = df[numeric_cols].apply(lambda x: round((x - x.min()) / (x.max() - x

# Added normalized columns back to the original DataFrame
for col in df_normalized.columns:
    df[f'{col}_normalized'] = df_normalized[col]
```

In [97]:

```
# 2e Prepare – explore and pre-process

# Summary Statistics Normalized Dataframe
df_normalized.describe()
```

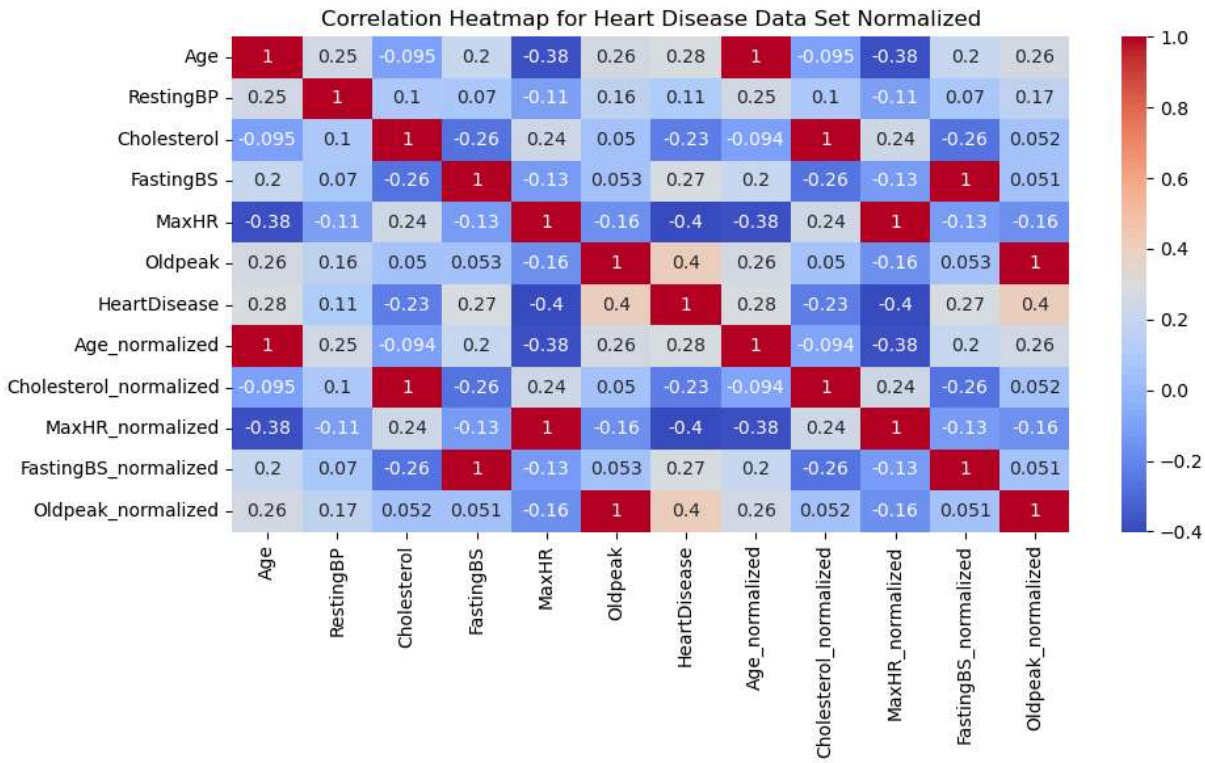
Out[97]:

	Age	Cholesterol	MaxHR	FastingBS	Oldpeak
count	918.000000	918.000000	918.000000	918.000000	918.000000
mean	0.520327	0.329597	0.540512	0.233115	0.398170
std	0.192296	0.181389	0.179295	0.423046	0.119703
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.390000	0.290000	0.420000	0.000000	0.300000
50%	0.530000	0.370000	0.550000	0.000000	0.360000
75%	0.650000	0.440000	0.680000	0.000000	0.470000
max	1.000000	1.000000	1.000000	1.000000	1.000000

In [101…

```
# 2f Prepare – explore and pre-process

#Correlation Heatmap for Normalized Data
```

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap for Heart Disease Data Set Normalized")
plt.tight_layout()
plt.show()
```



When comparing both non & normalized data for correlation: Age vs MaxHR shows promise Heart Disease vs MaxHR shows promise Cholesterol vs fastingBS shows promise Cholesterol vs MaxHR shows promise

In [105…

```
# 2g Prepare – explore and pre-process


# Bin Age into 4 equal groups
df['AgeGroup'] = pd.cut(df['Age'], bins=[0, 25, 50, 75, 100],
                        labels=['0-25', '26-50', '51-75', '76-100'],
                        right=True)

# Group by AgeGroups and summarize Cholesterol

df.groupby('AgeGroup')['Cholesterol'].agg(['mean', 'max', 'min', 'count'])
```

```
C:\Users\Nanoo\AppData\Local\Temp\ipykernel_23496\880737948.py:11: FutureWarning: Th
e default of observed=False is deprecated and will be changed to True in a future ve
rsion of pandas. Pass observed=False to retain current behavior or observed=True to
adopt the future default and silence this warning.
  df.groupby('AgeGroup')['Cholesterol'].agg(['mean', 'max', 'min', 'count'])
```

Out[105…

AgeGroup	mean	max	min	count
0–25	NaN	NaN	NaN	0
26–50	213.373418	529.0	0.0	316
51–75	191.115385	603.0	0.0	598
76–100	196.250000	304.0	113.0	4

In [ ]: