Data Mining

Exercise 2

Name: Chethan Kashyap Bangalore Muralidhara

Date: 09/11/2023

Completed Exercise: 1,2,3,4,5(All)

Question 1:

The code file will be attached as well.

Question 1:

- the types of variables: 'UVA', 'US', 'PVR', 'PMU', 'CYM', 'PTR', 'MUC', 'SS', 'UVJ', 'SSY', 'CLU', 'DV', 'USY', 'AGE'

UVA - binary - statistics: mode

US - Low 0–6, high 7–20 Categorical --> statistics: median and means

PVR - Categorical - statistics: median and mean

PMU - Categorical - statistics: median and mean

CYM - Binary - statistics:mode

PTR - Categorical - statistics: median and mean

MUCP - Negative ≤0, positive >0 = Categorical - statistics: median and mean

SS - binary - statistics: mode

UVJ - binary - statistics: mode

SSY binary - statistics: mode

CLU - binary - statistics: mode

DV binary - statistics: mode

USY- binary - statistics: mode

Age - Nominal - statistics: mode

How many different diagnoses the data contains? [0 1 2 3 4]

5 diagnoses total

The average age is 52.327586206896555 of all patients

```
In [ ]: AGE_COLUMN = pd.to_numeric(df['AGE'], errors='coerce')
```

```
In [ ]: print(AGE_COLUMN.mean())
```

```
52.327586206896555
```

Question 2:

Question 2: We find that only AGE has missing values so we group by the data by DIAGNOSI now missing values are replaced according to groups of DIAGNOSI

In [ ]:
```
for columns in df.columns:
    print(f'Column {columns}, has missing values: {df[columns].isna().any()}')
```

```
Column NO, has missing values: False
Column DIAGNOSI, has missing values: False
Column UVA, has missing values: False
Column US, has missing values: False
Column PVR, has missing values: False
Column PMU, has missing values: False
Column CYM, has missing values: False
Column PTR, has missing values: False
Column MUC, has missing values: False
Column SS, has missing values: False
Column UVJ, has missing values: False
Column SSY, has missing values: False
Column CLU, has missing values: False
Column DV, has missing values: False
Column USY, has missing values: False
Column AGE, has missing values: False
```

In [ ]:
```
df['AGE'].unique()
```

Out[13]:
```
array(['62', '46', '84', '53', '73', '63', '60', '40', '55', '67', '59',
       '48', '44', '27', '51', '45', '41', '50', '33', ' ', '57', '47',
       '37', '54', '65', '42', '64', '43', '36', '58', '32', '34', '49',
       '39', '72', '66', '38', '31', '35', '56', '30', '70', '69', '61',
       '52', '74', '79', '68', '86', '75', '71', '76', '78', '81', '26',
       '89', '77', '29', '28'], dtype=object)
```

In [ ]:
```
df['AGE'] = pd.to_numeric(df['AGE'], errors='coerce')
```

In [ ]:
```
df['AGE'].isna().any()
```

Out[15]:
```
True
```

In [ ]:
```
df.groupby(['DIAGNOSI'])['AGE'].mean()
```

Out[16]:
```
DIAGNOSI
0    50.444795
1    55.014388
2    55.545455
3    58.200000
4    53.944444
Name: AGE, dtype: float64
```

In [ ]:
```
df["AGE"] = df.groupby("DIAGNOSI")['AGE'].transform(lambda x: x.fillna(x.mean()))
```

In [ ]:
```
df['AGE'].isna().any()
```

Out[18]:
```
False
```

Question 3:

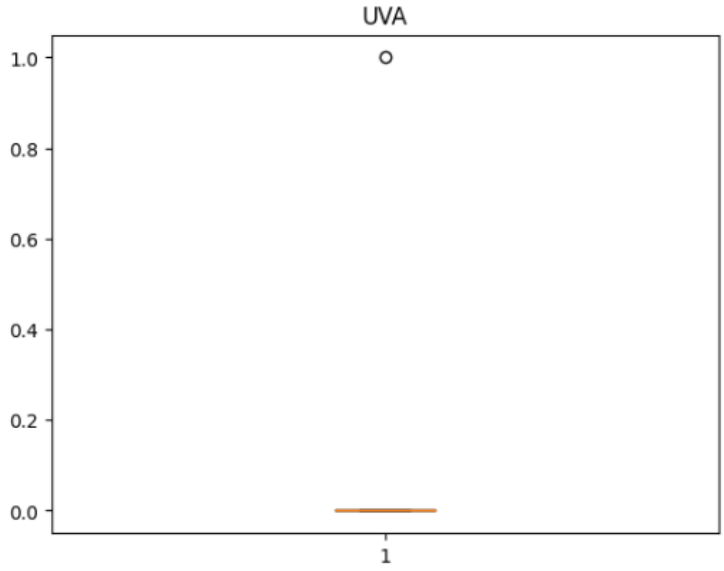
Question 3: Box plot for UVA, US, CYM and PTR Histogram for Age

What can you say about age distribution over all cases in the data?

0 and 1 looks Normal 2, 3, and 4 looks left skew

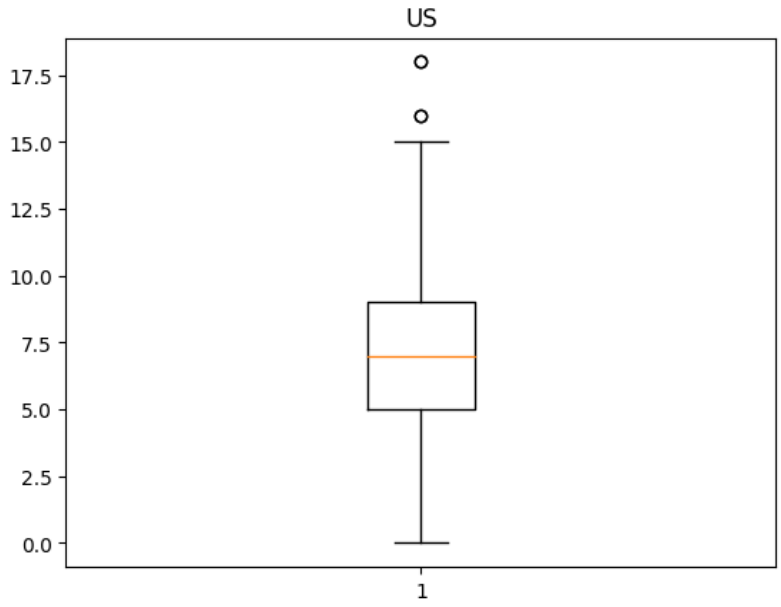
In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:
```
# Creating plot
df['UVA'] = pd.to_numeric(df['UVA'], errors='coerce')
plt.boxplot(df['UVA'].dropna())
# show plot
plt.title('UVA')
plt.show()
```

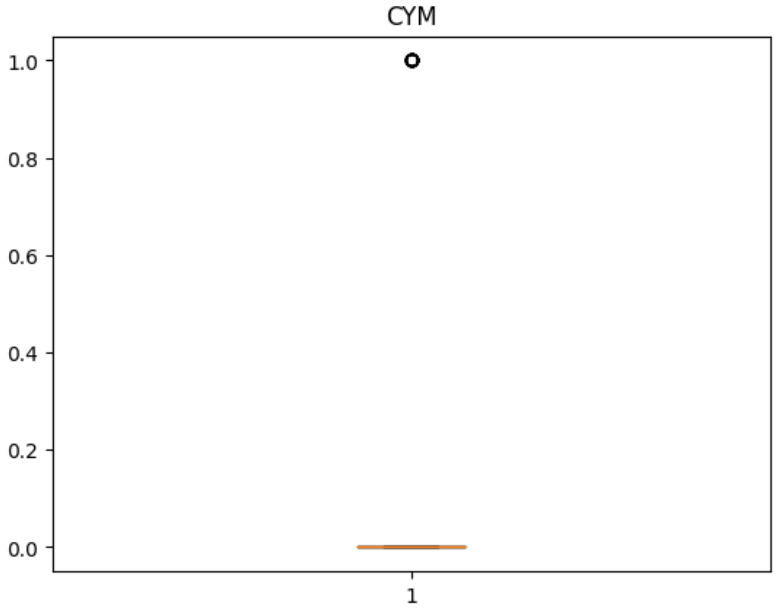


In [ ]:

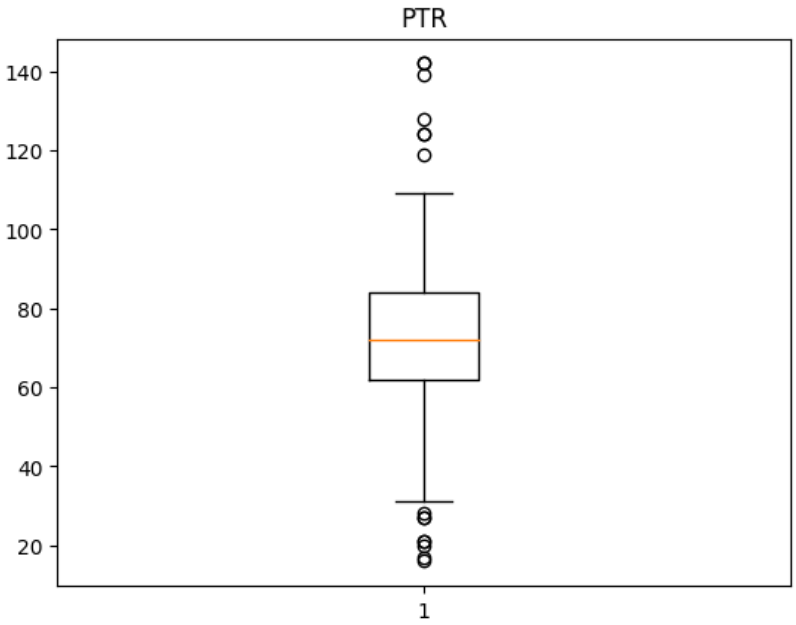
```
# Creating plot
df['US'] = pd.to_numeric(df['US'], errors='coerce')
plt.boxplot(df['US'].dropna())
# show plot
plt.title('US')
plt.show()
```



In [ ]:
```
# Creating plot
df['CYM'] = pd.to_numeric(df['CYM'], errors='coerce')
plt.boxplot(df['CYM'].dropna())
# show plot
plt.title('CYM')
plt.show()
```

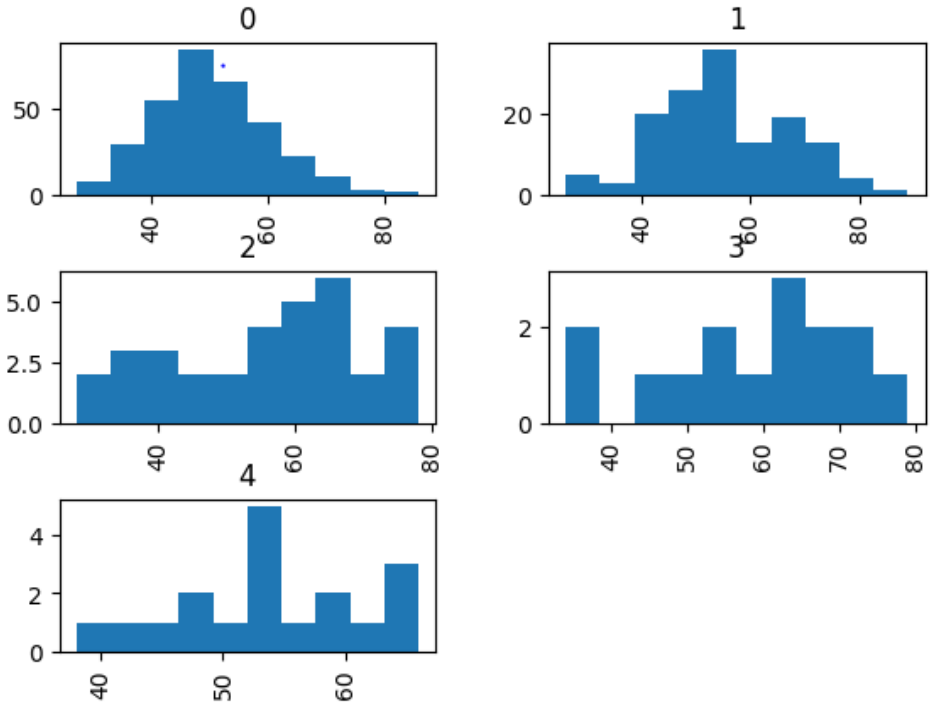


In [ ]:
```
# Creating plot
df['PTR'] = pd.to_numeric(df['PTR'], errors='coerce')
plt.boxplot(df['PTR'].dropna())
# show plot
plt.title('PTR')
plt.show()
```



```
In [ ]: df.hist(by="DIAGNOSI", column='AGE')
```

```
Out[24]: array([[<Axes: title={'center': '0'}>, <Axes: title={'center': '1'}>],
                [<Axes: title={'center': '2'}>, <Axes: title={'center': '3'}>],
                [<Axes: title={'center': '4'}>, <Axes: >]], dtype=object)
```



Question 4:

Question 4:

row 2 and 269: 8.0, row 2 and 393: 2.0, and row 269-393: 6.0

So it seems that row 2 and row 393 are the closest in terms of distance

What kind of problems you may encounter when you use Euclidean distance measure? if the data is not metric then it might not perform well

In addition to Euclidean distance, there are plenty of others. What other distance measures you have heard of?

Manhattan or Block city (or Hamming) distance, Tshebyschev distance, Minkowski distances, Mahalanobis distance

```
In [ ]: row_2_age = df.iloc[1]['AGE']
        row_269_age = df.iloc[268]['AGE']
        row_393_age = df.iloc[392]['AGE']
```

```
In [ ]: import math
```

```
In [ ]: distance_1 = math.sqrt((row_2_age-row_269_age)**2)
        distance_2 = math.sqrt((row_2_age-row_393_age)**2)
        distance_3 = math.sqrt((row_269_age-row_393_age)**2)
```

```
In [ ]: print(f'row 2 and 269: {distance_1}, row 2 and 393: {distance_2}, and row 269-393: {distance_3}')
```

```
row 2 and 269: 8.0, row 2 and 393: 2.0, and row 269-393: 6.0
```

Question 5:

```
distance_1 = num1-num2
distance_2 = num1-num3
distance_3 = num2-num3
sum_of_abs_value_of_differences = abs(distance_1) + abs(distance_2) + abs(distance_3)
%the main factor is we take the absolute values of the distance
```