DATA CLEANING PROCESS USING PYTHON

i)Data Cleaning process using Data_set.csv:

Import Required Libraries from Python Library:

```
import pandas as pd
import numpy as np
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
```

Reading the file and display first five data:

```
df=pd.read_csv("Data_set.csv")
df.head()
```

```
                show_name      country  num_episodes
aired_on  \
0                     NaN  South Korea            16     Friday,
Saturday
1                     NaN  South Korea            16     Friday,
Saturday
2  Descendants of the Sun  South Korea            16  Wednesday,
Thursday
3       Boys Over Flowers  South Korea            25      Monday,
Tuesday
4                       W  South Korea            16  Wednesday,
Thursday

  original_network  rating  current_overall_rank
lifetime_popularity_rank  \
0              tvN     8.9                  33.0
1
1             jTBC     8.7                  89.0
2
2             KBS2     8.7                  77.0
3
3             KBS2     7.7                2249.0
4
4              MBC     8.5                 201.0
5

   watchers
0  111706.0
1  100950.0
2   96318.0
3   94228.0
4   92121.0
```

Data set Information:

```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   show_name                 96 non-null     object 
 1   country                   100 non-null    object 
 2   num_episodes              100 non-null    int64  
 3   aired_on                  99 non-null     object 
 4   original_network          99 non-null     object 
 5   rating                    96 non-null     float64
 6   current_overall_rank      97 non-null     float64
 7   lifetime_popularity_rank  100 non-null    int64  
 8   watchers                  97 non-null     float64
dtypes: float64(3), int64(2), object(4)
memory usage: 7.2+ KB
```

```
       num_episodes     rating  current_overall_rank  \
count    100.000000  96.000000             97.000000   
mean      18.980000   8.293750            731.247423   
std        6.846041   0.424714            857.597007   
min        8.000000   7.300000              2.000000   
25%       16.000000   8.100000            194.000000   
50%       16.000000   8.300000            441.000000   
75%       20.000000   8.600000            806.000000   
max       50.000000   9.100000           3788.000000   

       lifetime_popularity_rank       watchers  
count                100.000000      97.000000  
mean                  51.650000   52994.907216  
std                   30.133164   17551.028458  
min                    1.000000   34523.000000  
25%                   25.750000   39545.000000  
50%                   51.500000   46963.000000  
75%                   77.250000   63140.000000  
max                  103.000000  111706.000000  
```

Handling Missing values and check Null Values

```
df.isnull()
df.isnull().sum()
```

```
show_name                 4
country                   0
num_episodes              0
```

```
aired_on                    1
original_network            1
rating                      4
current_overall_rank        3
lifetime_popularity_rank    0
watchers                    3
dtype: int64
```

Filling the Missing Values with 0

```
df1=df.fillna(0)
df1
```

```
                    show_name      country  num_episodes
aired_on  \
0                           0  South Korea            16     Friday,
Saturday
1                           0  South Korea            16     Friday,
Saturday
2     Descendants of the Sun  South Korea            16  Wednesday,
Thursday
3          Boys Over Flowers  South Korea            25      Monday,
Tuesday
4                           W  South Korea            16  Wednesday,
Thursday
..                        ...          ...           ...
...
95  Shut Up: Flower Boy Band  South Korea            16      Monday,
Tuesday
96                     Blood  South Korea            20      Monday,
Tuesday
97        Chicago Typewriter  South Korea            16     Friday,
Saturday
98      Sungkyunkwan Scandal  South Korea            20      Monday,
Tuesday
99                   Vagabond  South Korea            16     Friday,
Saturday

   original_network  rating  current_overall_rank
lifetime_popularity_rank  \
0               tvN     8.9                  33.0
1
1              jTBC     8.7                  89.0
2
2              KBS2     8.7                  77.0
3
3              KBS2     7.7                2249.0
4
4               MBC     8.5                 201.0
```

```
5
..              ...     ...                   ...
...
95               tvN     8.1                  806.0
99
96              KBS2     7.4                 3271.0
100
97               tvN     8.8                   51.0
101
98              KBS2     8.2                  605.0
102
99     SBS, Netflix     8.5                  238.0
103

    watchers
0   111706.0
1   100950.0
2    96318.0
3    94228.0
4    92121.0
..       ...
95   34668.0
96   34666.0
97       0.0
98   34615.0
99   34523.0

[100 rows x 9 columns]
```

Forward Fill using ffill()

```
df_ffill=df.ffill()
df_ffill

                    show_name      country  num_episodes
aired_on  \
0                         NaN  South Korea            16     Friday,
Saturday
1                         NaN  South Korea            16     Friday,
Saturday
2     Descendants of the Sun  South Korea            16  Wednesday,
Thursday
3          Boys Over Flowers  South Korea            25      Monday,
Tuesday
4                           W  South Korea            16  Wednesday,
Thursday
..                        ...          ...           ...
...
95  Shut Up: Flower Boy Band  South Korea            16      Monday,
```

```
Tuesday
96                     Blood  South Korea            20      Monday,
Tuesday
97         Chicago Typewriter  South Korea            16     Friday,
Saturday
98      Sungkyunkwan Scandal  South Korea            20      Monday,
Tuesday
99                   Vagabond  South Korea            16     Friday,
Saturday

   original_network  rating  current_overall_rank
lifetime_popularity_rank  \
0                tvN     8.9                  33.0
1
1               jTBC     8.7                  89.0
2
2               KBS2     8.7                  77.0
3
3               KBS2     7.7                2249.0
4
4                MBC     8.5                 201.0
5
..               ...     ...                   ...
...
95               tvN     8.1                 806.0
99
96              KBS2     7.4                3271.0
100
97               tvN     8.8                  51.0
101
98              KBS2     8.2                 605.0
102
99     SBS, Netflix     8.5                 238.0
103

    watchers
0   111706.0
1   100950.0
2    96318.0
3    94228.0
4    92121.0
..       ...
95   34668.0
96   34666.0
97   34666.0
98   34615.0
99   34523.0

[100 rows x 9 columns]
```

Backward fill using bfill()

```
df_bfill=df.bfill()
df_bfill
```

```
                    show_name      country  num_episodes
aired_on  \
0     Descendants of the Sun  South Korea            16     Friday,
Saturday
1     Descendants of the Sun  South Korea            16     Friday,
Saturday
2     Descendants of the Sun  South Korea            16  Wednesday,
Thursday
3          Boys Over Flowers  South Korea            25      Monday,
Tuesday
4                          W  South Korea            16  Wednesday,
Thursday
..                       ...          ...           ...
...
95  Shut Up: Flower Boy Band  South Korea            16      Monday,
Tuesday
96                     Blood  South Korea            20      Monday,
Tuesday
97        Chicago Typewriter  South Korea            16     Friday,
Saturday
98      Sungkyunkwan Scandal  South Korea            20      Monday,
Tuesday
99                   Vagabond  South Korea            16     Friday,
Saturday

   original_network  rating  current_overall_rank
lifetime_popularity_rank  \
0                tvN     8.9                  33.0
1
1              jTBC     8.7                  89.0
2
2              KBS2     8.7                  77.0
3
3              KBS2     7.7                2249.0
4
4               MBC     8.5                 201.0
5
..              ...     ...                   ...
...
95              tvN     8.1                 806.0
99
96             KBS2     7.4                3271.0
100
97              tvN     8.8                  51.0
101
```

```
98             KBS2     8.2                  605.0
102
99     SBS, Netflix     8.5                  238.0
103

    watchers
0   111706.0
1   100950.0
2    96318.0
3    94228.0
4    92121.0
..       ...
95   34668.0
96   34666.0
97   34615.0
98   34615.0
99   34523.0

[100 rows x 9 columns]
```

Filling the missing values with mean values:

```
df['rating']=df['rating'].fillna(df['rating'].mean())
df['watchers']=df['watchers'].fillna(df['watchers'].mean())
df
```

```
                    show_name      country  num_episodes
aired_on  \
0                         NaN  South Korea            16     Friday,
Saturday
1                         NaN  South Korea            16     Friday,
Saturday
2     Descendants of the Sun  South Korea            16  Wednesday,
Thursday
3          Boys Over Flowers  South Korea            25      Monday,
Tuesday
4                           W  South Korea            16  Wednesday,
Thursday
..                        ...          ...           ...
...
95  Shut Up: Flower Boy Band  South Korea            16      Monday,
Tuesday
96                     Blood  South Korea            20      Monday,
Tuesday
97        Chicago Typewriter  South Korea            16     Friday,
Saturday
98      Sungkyunkwan Scandal  South Korea            20      Monday,
Tuesday
99                   Vagabond  South Korea            16     Friday,
```

```
Saturday

   original_network  rating  current_overall_rank
lifetime_popularity_rank  \
0                tvN     8.9                  33.0
1
1               jTBC     8.7                  89.0
2
2               KBS2     8.7                  77.0
3
3               KBS2     7.7                2249.0
4
4                MBC     8.5                 201.0
5
..               ...     ...                   ...
...
95               tvN     8.1                 806.0
99
96              KBS2     7.4                3271.0
100
97               tvN     8.8                  51.0
101
98              KBS2     8.2                 605.0
102
99     SBS, Netflix     8.5                 238.0
103

         watchers
0   111706.000000
1   100950.000000
2    96318.000000
3    94228.000000
4    92121.000000
..            ...
95   34668.000000
96   34666.000000
97   52994.907216
98   34615.000000
99   34523.000000

[100 rows x 9 columns]
```

Deleting the rows which contains atleast one missing values:

```
df_dropna=df.dropna()
df_dropna

                          show_name      country  num_episodes  \
2            Descendants of the Sun  South Korea            16
```

```
3                  Boys Over Flowers  South Korea             25
4                                  W  South Korea             16
5       You Who Came from the Stars  South Korea             21
6   Weightlifting Fairy Kim Bok Joo  South Korea             16
..                              ...          ...            ...
95         Shut Up: Flower Boy Band  South Korea             16
96                           Blood  South Korea             20
97                Chicago Typewriter  South Korea             16
98              Sungkyunkwan Scandal  South Korea             20
99                          Vagabond  South Korea             16

                aired_on original_network  rating  current_overall_rank
\
2   Wednesday, Thursday             KBS2     8.7                  77.0

3       Monday, Tuesday             KBS2     7.7                2249.0

4   Wednesday, Thursday              MBC     8.5                 201.0

5   Wednesday, Thursday              SBS     8.6                 112.0

6   Wednesday, Thursday              MBC     8.8                  40.0

..                  ...              ...     ...                   ...

95      Monday, Tuesday              tvN     8.1                 806.0

96      Monday, Tuesday             KBS2     7.4                3271.0

97     Friday, Saturday              tvN     8.8                  51.0

98      Monday, Tuesday             KBS2     8.2                 605.0

99     Friday, Saturday     SBS, Netflix     8.5                 238.0


    lifetime_popularity_rank      watchers
2                          3  96318.000000
3                          4  94228.000000
4                          5  92121.000000
5                          6  91360.000000
6                          7  91330.000000
..                       ...           ...
95                        99  34668.000000
96                       100  34666.000000
97                       101  52994.907216
98                       102  34615.000000
99                       103  34523.000000

[92 rows x 9 columns]
```

Save the cleaned data in new file:

```
df_dropna.to_csv('exp1 data set.csv',index=False)
```

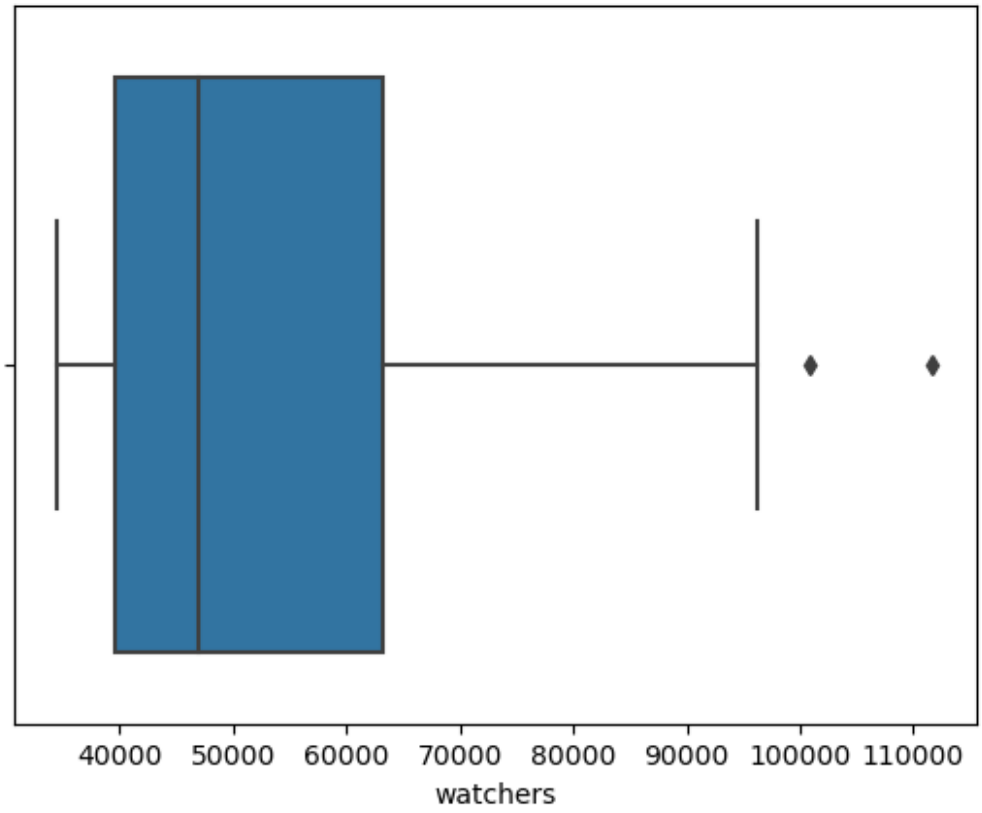
Detecting the outliers for Data_set.csv file:

Using IQR method:

```
df=pd.read_csv("Data_set.csv")
```

Using Box Plot for Detecting Outliers:

```
sns.boxplot(x=df['watchers'])
plt.show()
```



Calculate Q1 and Q3 to perform Q3-Q1

```
Q1=df['watchers'].quantile(0.25)
Q3=df['watchers'].quantile(0.75)
IQR=Q3-Q1
print(f"The IQR value is {IQR}")
```

```
The IQR value is 23595.0
```

Detecting outliers:

```
outliers = df[(df['watchers'] < (Q1 - 1.5 * IQR)) |
              (df['watchers'] > (Q3 + 1.5 * IQR))]


print(outliers)
```

```
  show_name      country  num_episodes          aired_on
original_network  \
0       NaN  South Korea            16  Friday, Saturday
tvN
1       NaN  South Korea            16  Friday, Saturday
jTBC

   rating  current_overall_rank  lifetime_popularity_rank  watchers
0     8.9                  33.0                         1  111706.0
1     8.7                  89.0                         2  100950.0
```

Removing Outliers

```
removed_outliers=df[~((df['watchers']<(Q1-1.5*IQR)) |
                      (df['watchers']>(Q3+1.5*IQR)))]
removed_outliers
```

```
                          show_name      country  num_episodes  \
2            Descendants of the Sun  South Korea            16
3                 Boys Over Flowers  South Korea            25
4                                 W  South Korea            16
5       You Who Came from the Stars  South Korea            21
6   Weightlifting Fairy Kim Bok Joo  South Korea            16
..                              ...          ...           ...
95         Shut Up: Flower Boy Band  South Korea            16
96                           Blood  South Korea            20
97             Chicago Typewriter  South Korea            16
98            Sungkyunkwan Scandal  South Korea            20
99                        Vagabond  South Korea            16

               aired_on original_network  rating  current_overall_rank
\
2   Wednesday, Thursday             KBS2     8.7                  77.0

3       Monday, Tuesday             KBS2     7.7                2249.0

4   Wednesday, Thursday              MBC     8.5                 201.0

5   Wednesday, Thursday              SBS     8.6                 112.0
```

```
6   Wednesday, Thursday              MBC     8.8                  40.0

..                  ...              ...     ...                   ...

95      Monday, Tuesday              tvN     8.1                 806.0

96      Monday, Tuesday             KBS2     7.4                3271.0

97     Friday, Saturday              tvN     8.8                  51.0

98      Monday, Tuesday             KBS2     8.2                 605.0

99     Friday, Saturday     SBS, Netflix     8.5                 238.0


    lifetime_popularity_rank  watchers
2                          3   96318.0
3                          4   94228.0
4                          5   92121.0
5                          6   91360.0
6                          7   91330.0
..                       ...       ...
95                        99   34668.0
96                       100   34666.0
97                       101       NaN
98                       102   34615.0
99                       103   34523.0

[98 rows x 9 columns]
```

Calculate Outliers using Z Score Method using current_overall_rank Column

```
z_score=np.abs(stats.zscore(df['rating'].dropna()))
z_score
```

```
0     1.434926
1     0.961548
2     0.961548
3     1.405340
4     0.488171
        ...
95    0.458585
96    2.115406
97    1.198237
98    0.221896
99    0.488171
Name: rating, Length: 96, dtype: float64
```

Detecting Outliers

```
threshold=3
mask = np.zeros(len(df), dtype=bool)
mask[df['rating'].dropna().index] = z_score > threshold
outliers = df[mask]
print('outliers')
print(outliers)
```

```
outliers
Empty DataFrame
Columns: [show_name, country, num_episodes, aired_on,
original_network, rating, current_overall_rank,
lifetime_popularity_rank, watchers]
Index: []
```

Removing Outliers

```
mask = np.ones(len(df), dtype=bool)
mask[df['rating'].dropna().index] = z_score <= threshold
df_cleaned = df[mask]
df_cleaned
```

```
                    show_name      country  num_episodes
aired_on  \
0                         NaN  South Korea            16     Friday,
Saturday
1                         NaN  South Korea            16     Friday,
Saturday
2     Descendants of the Sun  South Korea            16  Wednesday,
Thursday
3          Boys Over Flowers  South Korea            25      Monday,
Tuesday
4                          W  South Korea            16  Wednesday,
Thursday
..                       ...          ...           ...
...
95  Shut Up: Flower Boy Band  South Korea            16      Monday,
Tuesday
96                     Blood  South Korea            20      Monday,
Tuesday
97        Chicago Typewriter  South Korea            16     Friday,
Saturday
98      Sungkyunkwan Scandal  South Korea            20      Monday,
Tuesday
99                   Vagabond  South Korea            16     Friday,
Saturday

   original_network  rating  current_overall_rank
lifetime_popularity_rank  \
0               tvN     8.9                  33.0
```

```
1
1              jTBC     8.7                   89.0
2
2              KBS2     8.7                   77.0
3
3              KBS2     7.7                 2249.0
4
4               MBC     8.5                  201.0
5
..              ...     ...                    ...
...
95              tvN     8.1                  806.0
99
96             KBS2     7.4                 3271.0
100
97              tvN     8.8                   51.0
101
98             KBS2     8.2                  605.0
102
99     SBS, Netflix     8.5                  238.0
103

    watchers
0   111706.0
1   100950.0
2    96318.0
3    94228.0
4    92121.0
..       ...
95   34668.0
96   34666.0
97       NaN
98   34615.0
99   34523.0

[100 rows x 9 columns]
```

i)Data Cleaning process using heights.csv:

Import required Libraries:

```
import pandas as pd
import numpy as np
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
```

Reading the heights.csv file:

```
df=pd.read_csv("heights (1).csv")
df.head()
```

```
    name  height
0  mohan     5.9
1  maria     5.2
2  sakib     5.1
3    tao     5.5
4  virat     4.9
```

Dataset information using info() and describe()

```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   name    14 non-null     object 
 1   height  14 non-null     float64
dtypes: float64(1), object(1)
memory usage: 352.0+ bytes
```

```
          height
count  14.000000
mean    6.050000
std     2.779804
min     1.200000
25%     5.250000
50%     5.550000
75%     6.175000
max    14.500000
```

Check whether the Data Contains any missing values

```
df.isnull()
df.isnull().sum()
```

```
name      0
height    0
dtype: int64
```

Filling the Missing values with 0 if it exists:

```
df_isnull=df.fillna(0)
df_isnull
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
6    dmitry     6.2
7    selena     6.5
8      john     7.1
9     imran    14.5
10     jose     6.1
11  deepika     5.6
12   yoseph     1.2
13    binod     5.5
```

Filling the missing values with forward fill:

```
df_ffill=df.ffill()
df_ffill
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
6    dmitry     6.2
7    selena     6.5
8      john     7.1
9     imran    14.5
10     jose     6.1
11  deepika     5.6
12   yoseph     1.2
13    binod     5.5
```

Filling the missing value with backward fill:

```
df_bfill=df.bfill()
df_bfill
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
```

```
6    dmitry     6.2
7    selena     6.5
8      john     7.1
9     imran    14.5
10     jose     6.1
11  deepika     5.6
12   yoseph     1.2
13    binod     5.5
```

Filling the row with mean value:

```
df['height']=df['height'].fillna(df['height'].mean())
df
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
6    dmitry     6.2
7    selena     6.5
8      john     7.1
9     imran    14.5
10     jose     6.1
11  deepika     5.6
12   yoseph     1.2
13    binod     5.5
```

Dropping the missing values using dropna()

```
df_dropna=df.dropna()
df_dropna
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
6    dmitry     6.2
7    selena     6.5
8      john     7.1
9     imran    14.5
10     jose     6.1
11  deepika     5.6
```

```
12   yoseph     1.2
13    binod     5.5
```

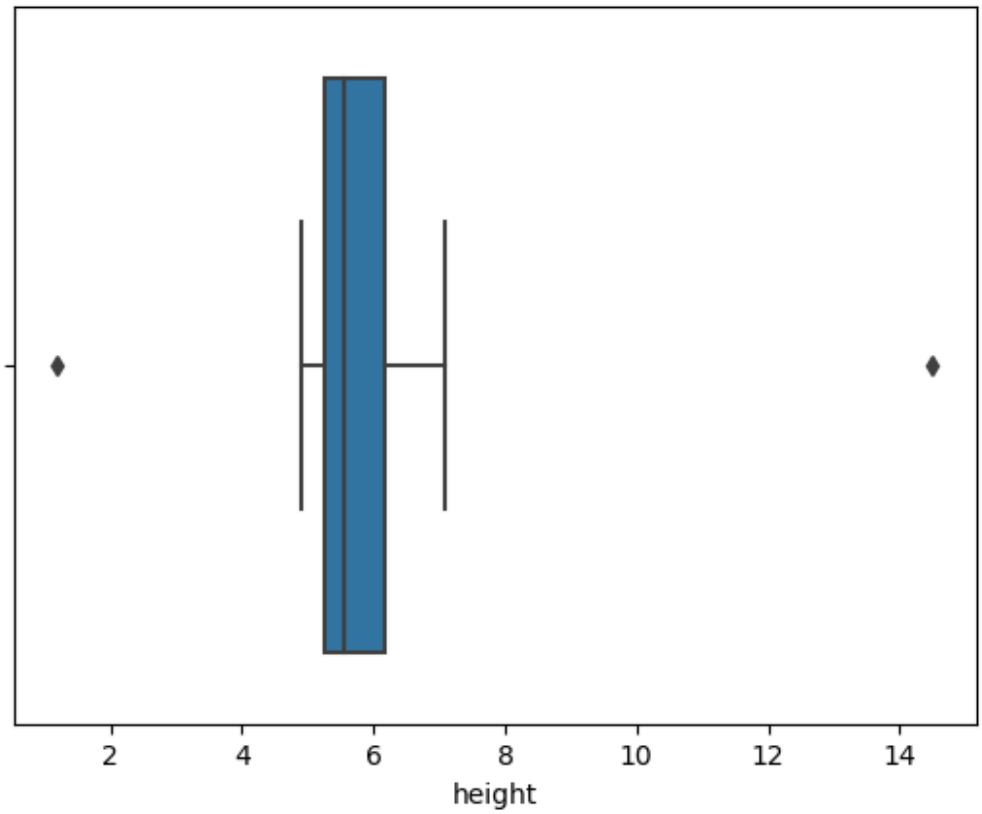
Save the cleaned data to new file named cleaned_heights.csv

```
df_dropna.to_csv("cleaned_heights.csv",index=False)
```

Detecting the outliers using IQR method:

Using Boxplot method

```
sns.boxplot(x=df['height'])
plt.show()
```



Calculating IQR by finding Q1 and Q3:

```
Q1=df['height'].quantile(0.25)
Q3=df['height'].quantile(0.75)
IQR=Q3-Q1
print("The IQR value is",IQR)

The IQR value is 0.9249999999999998
```

Detecting the outliers

```
outliers_iqr=df[(df['height']<(Q1-1.5*IQR))|
                (df['height']>(Q3+1.5*IQR))]
outliers_iqr
```

```
      name  height
9    imran    14.5
12  yoseph     1.2
```

Removing Outliers:

```
removed_outliers=df[~((df['height']<(Q1-1.5*IQR))|
                      (df['height']>(Q3+1.5*IQR)))]
removed_outliers
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
6    dmitry     6.2
7    selena     6.5
8      john     7.1
10     jose     6.1
11  deepika     5.6
13    binod     5.5
```

Calculate Z score using Z-Score Method:

```
z_score=np.abs(stats.zscore(df['height']))
z_score
```

```
0     0.055998
1     0.317320
2     0.354652
3     0.205325
4     0.429315
5     0.242656
6     0.055998
7     0.167993
8     0.391983
9     3.154532
10    0.018666
11    0.167993
12    1.810589
13    0.205325
Name: height, dtype: float64
```

Detecting outliers:

```
threshold=3
outlier=df[z_score>threshold]
print('outlier')
outlier
```

```
outlier
```

```
    name  height
9  imran    14.5
```

Removing the outliers:

```
cleaned=df[z_score<=threshold]
cleaned
```

```
       name  height
0     mohan     5.9
1     maria     5.2
2     sakib     5.1
3       tao     5.5
4     virat     4.9
5    khusbu     5.4
6    dmitry     6.2
7    selena     6.5
8      john     7.1
10     jose     6.1
11  deepika     5.6
12   yoseph     1.2
13    binod     5.5
```

RESULT:

Thus,we have cleaned the data and removed the outliers in heights.csv

iii) Data Cleaning process using Loan_data.csv file

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

Reading the Data in the Loan_dat.csv file:

```
df=pd.read_csv("Loan_data.csv")
df.head()
```

```
    Loan_ID Gender Married Dependents     Education Self_Employed  \
0  LP001015   Male     Yes          0      Graduate            No
1  LP001022   Male     Yes          1      Graduate            No
2  LP001031   Male     Yes          2      Graduate            No
3  LP001035   Male     Yes          2      Graduate            No
4  LP001051   Male      No          0  Not Graduate            No

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             5720                  0       110.0             360.0
1             3076               1500       126.0             360.0
2             5000               1800       208.0             360.0
3             2340               2546       100.0             360.0
4             3276                  0        78.0             360.0

   Credit_History Property_Area
0             1.0         Urban
1             1.0         Urban
2             1.0         Urban
3             NaN         Urban
4             1.0         Urban
```

Data set Information using info() and describe()

```
df.info()
df.describe()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            367 non-null    object
 1   Gender             356 non-null    object
 2   Married            367 non-null    object
 3   Dependents         357 non-null    object
 4   Education          367 non-null    object
 5   Self_Employed      344 non-null    object
 6   ApplicantIncome    367 non-null    int64
 7   CoapplicantIncome  367 non-null    int64
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   361 non-null    float64
 10  Credit_History     338 non-null    float64
 11  Property_Area      367 non-null    object
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB

       ApplicantIncome  CoapplicantIncome  LoanAmount
Loan_Amount_Term  \
count       367.000000         367.000000  362.000000
361.000000
```

```
mean       4805.599455         1569.577657  136.132597
342.537396
std        4910.685399         2334.232099   61.366652
65.156643
min           0.000000            0.000000   28.000000
6.000000
25%        2864.000000            0.000000  100.250000
360.000000
50%        3786.000000         1025.000000  125.000000
360.000000
75%        5060.000000         2430.500000  158.000000
360.000000
max       72529.000000        24000.000000  550.000000
480.000000

       Credit_History
count      338.000000
mean         0.825444
std          0.380150
min          0.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          1.000000
```

Checking the missing values:

```
df.isnull()
df.isnull().sum()
```

```
Loan_ID               0
Gender               11
Married               0
Dependents           10
Education             0
Self_Employed        23
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

Filling the missing values with 0:

```
df_0=df.fillna(0)
df_0
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No
1    LP001022   Male     Yes          1      Graduate            No
2    LP001031   Male     Yes          2      Graduate            No
3    LP001035   Male     Yes          2      Graduate            No
4    LP001051   Male      No          0  Not Graduate            No
..        ...    ...     ...        ...          ...           ...
362  LP002971   Male     Yes         3+  Not Graduate           Yes
363  LP002975   Male     Yes          0      Graduate            No
364  LP002980   Male      No          0      Graduate            No
365  LP002986   Male     Yes          0      Graduate            No
366  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               5720                  0       110.0             360.0

1               3076               1500       126.0             360.0

2               5000               1800       208.0             360.0

3               2340               2546       100.0             360.0

4               3276                  0        78.0             360.0

..               ...                ...         ...               ...

362             4009               1777       113.0             360.0

363             4158                709       115.0             360.0

364             3250               1993       126.0             360.0

365             5000               2393       158.0             360.0

366             9200                  0        98.0             180.0


     Credit_History Property_Area
0               1.0         Urban
1               1.0         Urban
2               1.0         Urban
3               0.0         Urban
4               1.0         Urban
..              ...           ...
362             1.0         Urban
363             1.0         Urban
364             0.0     Semiurban
365             1.0         Rural
366             1.0         Rural
```

```
[367 rows x 12 columns]
```

Filling the missing values with ffill:

```
df_ffill=df.ffill()
df_ffill
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No   
1    LP001022   Male     Yes          1      Graduate            No   
2    LP001031   Male     Yes          2      Graduate            No   
3    LP001035   Male     Yes          2      Graduate            No   
4    LP001051   Male      No          0  Not Graduate            No   
..        ...    ...     ...        ...          ...           ...   
362  LP002971   Male     Yes         3+  Not Graduate           Yes   
363  LP002975   Male     Yes          0      Graduate            No   
364  LP002980   Male      No          0      Graduate            No   
365  LP002986   Male     Yes          0      Graduate            No   
366  LP002989   Male      No          0      Graduate           Yes   

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5720                  0       110.0             360.0   
1               3076               1500       126.0             360.0   
2               5000               1800       208.0             360.0   
3               2340               2546       100.0             360.0   
4               3276                  0        78.0             360.0   
..               ...                ...         ...               ...   
362             4009               1777       113.0             360.0   
363             4158                709       115.0             360.0   
364             3250               1993       126.0             360.0   
365             5000               2393       158.0             360.0   
366             9200                  0        98.0             180.0   

     Credit_History Property_Area  
0               1.0         Urban  
1               1.0         Urban  
2               1.0         Urban  
```

```
3               1.0         Urban
4               1.0         Urban
..              ...           ...
362             1.0         Urban
363             1.0         Urban
364             1.0     Semiurban
365             1.0         Rural
366             1.0         Rural

[367 rows x 12 columns]
```

Filling the missing values with backward fill:

```
df_bfill=df.bfill()
df_bfill
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No
1    LP001022   Male     Yes          1      Graduate            No
2    LP001031   Male     Yes          2      Graduate            No
3    LP001035   Male     Yes          2      Graduate            No
4    LP001051   Male      No          0  Not Graduate            No
..        ...    ...     ...        ...          ...           ...
362  LP002971   Male     Yes         3+  Not Graduate           Yes
363  LP002975   Male     Yes          0      Graduate            No
364  LP002980   Male      No          0      Graduate            No
365  LP002986   Male     Yes          0      Graduate            No
366  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               5720                  0       110.0             360.0

1               3076               1500       126.0             360.0

2               5000               1800       208.0             360.0

3               2340               2546       100.0             360.0

4               3276                  0        78.0             360.0

..               ...                ...         ...               ...

362             4009               1777       113.0             360.0

363             4158                709       115.0             360.0

364             3250               1993       126.0             360.0

365             5000               2393       158.0             360.0
```

```
366              9200                   0        98.0              180.0


     Credit_History Property_Area
0               1.0         Urban
1               1.0         Urban
2               1.0         Urban
3               1.0         Urban
4               1.0         Urban
..              ...           ...
362             1.0         Urban
363             1.0         Urban
364             1.0     Semiurban
365             1.0         Rural
366             1.0         Rural

[367 rows x 12 columns]
```

Filling the missing values with mean value:

```
df['Credit_History']=df['Credit_History'].fillna(df['Credit_History'].
mean())
df
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No
1    LP001022   Male     Yes          1      Graduate            No
2    LP001031   Male     Yes          2      Graduate            No
3    LP001035   Male     Yes          2      Graduate            No
4    LP001051   Male      No          0  Not Graduate            No
..        ...    ...     ...        ...           ...           ...
362  LP002971   Male     Yes         3+  Not Graduate           Yes
363  LP002975   Male     Yes          0      Graduate            No
364  LP002980   Male      No          0      Graduate            No
365  LP002986   Male     Yes          0      Graduate            No
366  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               5720                  0       110.0             360.0

1               3076               1500       126.0             360.0

2               5000               1800       208.0             360.0

3               2340               2546       100.0             360.0

4               3276                  0        78.0             360.0
```

```
..                ...                 ...         ...                ...

362              4009                1777       113.0              360.0

363              4158                 709       115.0              360.0

364              3250                1993       126.0              360.0

365              5000                2393       158.0              360.0

366              9200                   0        98.0              180.0


     Credit_History Property_Area
0          1.000000         Urban
1          1.000000         Urban
2          1.000000         Urban
3          0.825444         Urban
4          1.000000         Urban
..              ...           ...
362        1.000000         Urban
363        1.000000         Urban
364        0.825444     Semiurban
365        1.000000         Rural
366        1.000000         Rural

[367 rows x 12 columns]
```

Dropping the missing values:

```
df_dropna=df.dropna()
df_dropna
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No
1    LP001022   Male     Yes          1      Graduate            No
2    LP001031   Male     Yes          2      Graduate            No
3    LP001035   Male     Yes          2      Graduate            No
4    LP001051   Male      No          0  Not Graduate            No
..        ...    ...     ...        ...           ...           ...
362  LP002971   Male     Yes         3+  Not Graduate           Yes
363  LP002975   Male     Yes          0      Graduate            No
364  LP002980   Male      No          0      Graduate            No
365  LP002986   Male     Yes          0      Graduate            No
366  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               5720                  0       110.0             360.0
```

```
1                3076                1500       126.0               360.0
2                5000                1800       208.0               360.0
3                2340                2546       100.0               360.0
4                3276                   0        78.0               360.0
..                ...                 ...         ...                 ...
362              4009                1777       113.0               360.0
363              4158                 709       115.0               360.0
364              3250                1993       126.0               360.0
365              5000                2393       158.0               360.0
366              9200                   0        98.0               180.0


     Credit_History Property_Area
0          1.000000         Urban
1          1.000000         Urban
2          1.000000         Urban
3          0.825444         Urban
4          1.000000         Urban
..              ...           ...
362        1.000000         Urban
363        1.000000         Urban
364        0.825444     Semiurban
365        1.000000         Rural
366        1.000000         Rural

[314 rows x 12 columns]
```

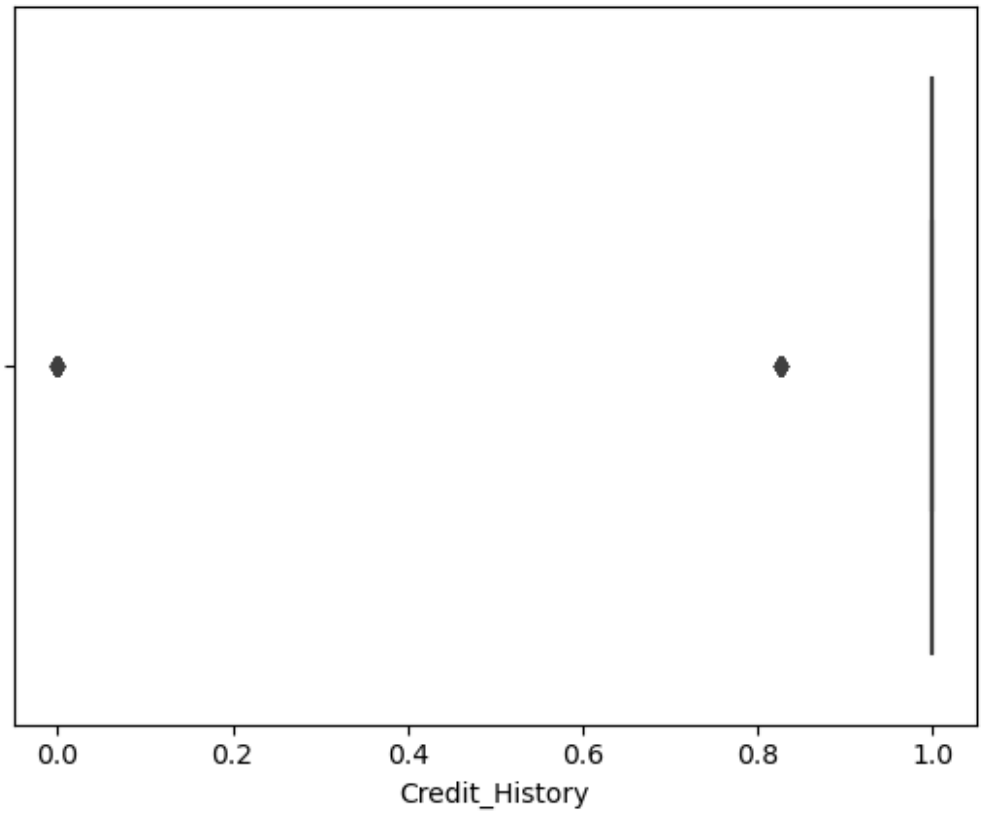
Save the cleaned data to new file named 'cleaned_Loan_data.csv'

```
df_dropna.to_csv("cleaned_Loan_data.csv",index=False)
```

Outlier Detection Using IQR Method:

Using BOXPLOT Method:

```
sns.boxplot(x=df['Credit_History'])
plt.show()
```



Calculate IQR

```
Q1=df['Credit_History'].quantile(0.25)
Q3=df['Credit_History'].quantile(0.75)
IQR=Q3-Q1
print("IQR value is",IQR)
```

```
IQR value is 0.0
```

Detecting outliers

```
outliers = df[(df['Credit_History'] < (Q1 - 1.5 * IQR)) |
              (df['Credit_History'] > (Q3 + 1.5 * IQR))]
outliers
```

```
      Loan_ID  Gender Married Dependents     Education
Self_Employed  \
3    LP001035    Male     Yes          2      Graduate             No

7    LP001056    Male     Yes          2  Not Graduate             No

12   LP001083    Male      No         3+      Graduate             No

13   LP001094    Male     Yes          2      Graduate            NaN
```

```
25   LP001153    Male      No          0      Graduate             No
..        ...     ...     ...        ...           ...            ...
351  LP002901    Male      No          0      Graduate             No
354  LP002921    Male     Yes         3+  Not Graduate             No
358  LP002954    Male     Yes          2  Not Graduate             No
360  LP002965  Female     Yes          0      Graduate             No
364  LP002980    Male      No          0      Graduate             No

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
3               2340               2546       100.0             360.0
7               3881                  0       147.0             360.0
12              4166                  0        40.0             180.0
13             12173                  0       166.0             360.0
25                 0              24000       148.0             360.0
..               ...                ...         ...               ...
351             2283              15000       106.0             360.0
354             5316                187       158.0             180.0
358             3132                  0        76.0             360.0
360             8550               4255        96.0             360.0
364             3250               1993       126.0             360.0

     Credit_History Property_Area
3          0.825444         Urban
7          0.000000         Rural
12         0.825444         Urban
13         0.000000     Semiurban
25         0.000000         Rural
..              ...           ...
351        0.825444         Rural
354        0.000000     Semiurban
358        0.825444         Rural
360        0.825444         Urban
```

```
364        0.825444     Semiurban

[88 rows x 12 columns]
```

Removing outliers:

```
df_cleaned=df[~((df['Credit_History'] < (Q1 - 1.5 * IQR)) |
                (df['Credit_History'] > (Q3 + 1.5 * IQR)))]
df_cleaned
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No
1    LP001022   Male     Yes          1      Graduate            No
2    LP001031   Male     Yes          2      Graduate            No
4    LP001051   Male      No          0  Not Graduate            No
5    LP001054   Male     Yes          0  Not Graduate           Yes
..        ...    ...     ...        ...           ...           ...
361  LP002969   Male     Yes          1      Graduate            No
362  LP002971   Male     Yes         3+  Not Graduate           Yes
363  LP002975   Male     Yes          0      Graduate            No
365  LP002986   Male     Yes          0      Graduate            No
366  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               5720                  0       110.0             360.0

1               3076               1500       126.0             360.0

2               5000               1800       208.0             360.0

4               3276                  0        78.0             360.0

5               2165               3422       152.0             360.0

..               ...                ...         ...               ...

361             2269               2167        99.0             360.0

362             4009               1777       113.0             360.0

363             4158                709       115.0             360.0

365             5000               2393       158.0             360.0

366             9200                  0        98.0             180.0


     Credit_History Property_Area
0               1.0         Urban
```

```
1                1.0         Urban
2                1.0         Urban
4                1.0         Urban
5                1.0         Urban
..               ...           ...
361              1.0     Semiurban
362              1.0         Urban
363              1.0         Urban
365              1.0         Rural
366              1.0         Rural

[279 rows x 12 columns]
```

Detecting outliers using Z-Score Method:

Calculate Z Score

```
z_score=np.abs(stats.zscore(df['Credit_History']))
z_score
```

```
0      0.47918
1      0.47918
2      0.47918
3      0.00000
4      0.47918
        ...
362    0.47918
363    0.47918
364    0.00000
365    0.47918
366    0.47918
Name: Credit_History, Length: 367, dtype: float64
```

Detecting Outliers

```
threshold=3
outliers=df[z_score>threshold]
print('outliers')
outliers
```

```
outliers
```

```
Empty DataFrame
Columns: [Loan_ID, Gender, Married, Dependents, Education,
Self_Employed, ApplicantIncome, CoapplicantIncome, LoanAmount,
Loan_Amount_Term, Credit_History, Property_Area]
Index: []
```

Removing outliers

```
cleaned=df[z_score<=threshold]
cleaned
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001015   Male     Yes          0      Graduate            No   
1    LP001022   Male     Yes          1      Graduate            No   
2    LP001031   Male     Yes          2      Graduate            No   
3    LP001035   Male     Yes          2      Graduate            No   
4    LP001051   Male      No          0  Not Graduate            No   
..        ...    ...     ...        ...          ...           ...   
362  LP002971   Male     Yes         3+  Not Graduate           Yes   
363  LP002975   Male     Yes          0      Graduate            No   
364  LP002980   Male      No          0      Graduate            No   
365  LP002986   Male     Yes          0      Graduate            No   
366  LP002989   Male      No          0      Graduate           Yes   

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5720                  0       110.0             360.0   
1               3076               1500       126.0             360.0   
2               5000               1800       208.0             360.0   
3               2340               2546       100.0             360.0   
4               3276                  0        78.0             360.0   
..               ...                ...         ...               ...   
362             4009               1777       113.0             360.0   
363             4158                709       115.0             360.0   
364             3250               1993       126.0             360.0   
365             5000               2393       158.0             360.0   
366             9200                  0        98.0             180.0   

     Credit_History Property_Area  
0          1.000000         Urban  
1          1.000000         Urban  
2          1.000000         Urban  
3          0.825444         Urban  
4          1.000000         Urban  
..              ...           ...  
362        1.000000         Urban  
363        1.000000         Urban  
364        0.825444     Semiurban  
```

```
365         1.000000          Rural
366         1.000000          Rural

[367 rows x 12 columns]
```

Result:

Thus,the we have cleaned the data and removed the outliers in Loan_data.csv .

iv)Data cleaning process using iris.csv

Import required libraries

```
import numpy as np
import pandas as pd
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
```

Read the Data file

```
df=pd.read_csv("iris.csv")
df.head()
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

Dataset Information:

```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

Checking Whether the data contain any empty values:

```
df.isnull()
df.isnull().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Filling the missing values with 0:

```
df_fillna=df.fillna(0)
df_fillna
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

Fill the data using ffill

```
df_ffill=df.ffill()
df_ffill
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
```

```
1              4.9           3.0            1.4           0.2     setosa
2              4.7           3.2            1.3           0.2     setosa
3              4.6           3.1            1.5           0.2     setosa
4              5.0           3.6            1.4           0.2     setosa
..             ...           ...            ...           ...        ...
145            6.7           3.0            5.2           2.3  virginica
146            6.3           2.5            5.0           1.9  virginica
147            6.5           3.0            5.2           2.0  virginica
148            6.2           3.4            5.4           2.3  virginica
149            5.9           3.0            5.1           1.8  virginica

[150 rows x 5 columns]
```

Fill the data using bfill

```
df_bfill=df.bfill()
df_bfill
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

Filling the data with mean value

```
df['sepal_length']=df['sepal_length'].fillna(df['sepal_length'].mean()
)
df
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
```

```
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

Dropping the missing values:

```
df_drop=df.dropna()
df_drop
```

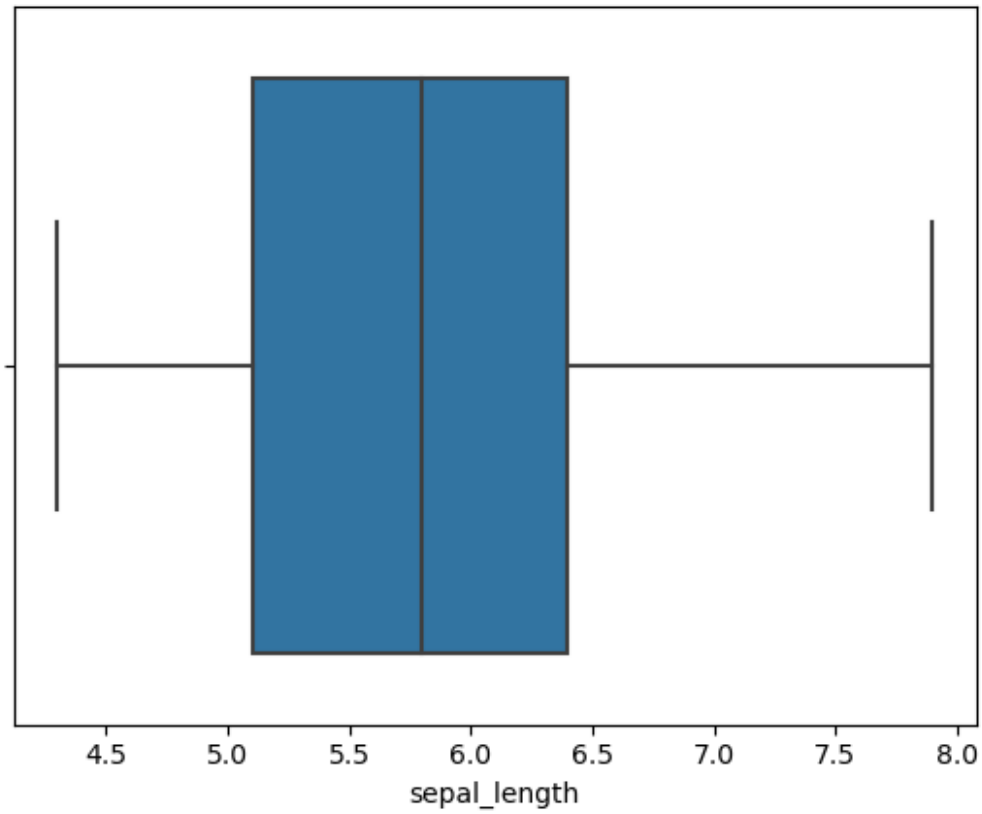
```
     sepal_length  sepal_width  petal_length  petal_width     species
0             5.1          3.5           1.4          0.2      setosa
1             4.9          3.0           1.4          0.2      setosa
2             4.7          3.2           1.3          0.2      setosa
3             4.6          3.1           1.5          0.2      setosa
4             5.0          3.6           1.4          0.2      setosa
..            ...          ...           ...          ...         ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

Save the cleaned data in a new file:

```
df_drop.to_csv('cleaned_iris.csv',index=False)
```

Outlier detection using IQR Method:

```
sns.boxplot(x=df['sepal_length'])
plt.show()
```



Calculate IQR value:

```
Q1=df['sepal_length'].quantile(0.25)
Q3=df['sepal_length'].quantile(0.75)
IQR=Q3-Q1
print("IQR value is",IQR)
```

```
IQR value is 1.3000000000000007
```

Detecting Outliers:

```
outliers=df[(df['sepal_length']<(Q1-1.5*IQR))|
(df['sepal_length']>(Q3+1.5*IQR))]
outliers
```

```
Empty DataFrame
Columns: [sepal_length, sepal_width, petal_length, petal_width,
species]
Index: []
```

Removing outliers

```
cleaned=df[~((df['sepal_length']<(Q1-1.5*IQR))|
             (df['sepal_length']>(Q3+1.5*IQR)))]
cleaned
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

Detecting Outliers using Z Score method:

```
z_score=np.abs(stats.zscore(df['sepal_length']))
z_score
```

```
0      0.900681
1      1.143017
2      1.385353
3      1.506521
4      1.021849
         ...
145    1.038005
146    0.553333
147    0.795669
148    0.432165
149    0.068662
Name: sepal_length, Length: 150, dtype: float64
```

Detecting outliers:

```
threshold=3
outlier=df[z_score>threshold]
print("Outliers:")
outlier
```

```
Outliers:

Empty DataFrame
Columns: [sepal_length, sepal_width, petal_length, petal_width,
species]
Index: []
```

Removing outliers:

```
cleaned=df[z_score<=threshold]
cleaned
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

RESULT:

Thus we have cleaned the data and removed the outliers by detection using IQR and Z-score method.

Data Cleaning Process using SAMPLEIDS.csv

Import Required Libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

Reading the Data Set :

```
df=pd.read_csv("SAMPLEIDS.csv")
df.head()
```

```
   SNO    REGNO    NAME         DOB  GENDER      ADDRESS    M1    M2
M3  \
0    1  1220121    ARUN  2000-02-10    MALE    THANDALAM  82.0  81.0
90.0
1    2  1220122    BABU  1999-01-25    MALE  KANCHIPURAM  56.0  61.0
80.0
2    3  1220123  CHARAN  2000.09.21    MALE    THANDALAM   NaN  59.0
60.0
3    4  1220124    DEVA  2000-11-09    MALE   POONAMALEE  74.0  79.0
80.0
```

```
4    5  1220125   ESTER  2000-11-21  FEMALE      CHITHUR  92.0  95.0
96.0

     M4  TOTAL         AVG
0   NaN    NaN         NaN
1  56.0  253.0   84.333333
2  70.0    NaN    0.000000
3  74.0  307.0  102.333333
4  92.0  375.0  125.000000
```

Data Set Information:

```
df.info()
df.describe()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 12 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   SNO      21 non-null     int64
 1   REGNO    21 non-null     int64
 2   NAME     20 non-null     object
 3   DOB      21 non-null     object
 4   GENDER   20 non-null     object
 5   ADDRESS  20 non-null     object
 6   M1       18 non-null     float64
 7   M2       19 non-null     float64
 8   M3       17 non-null     float64
 9   M4       18 non-null     float64
 10  TOTAL    16 non-null     float64
 11  AVG      20 non-null     float64
dtypes: float64(6), int64(2), object(4)
memory usage: 2.1+ KB

             SNO         REGNO         M1         M2         M3
M4  \
count  21.000000  2.100000e+01  18.000000  19.000000  17.000000
18.000000
mean   10.333333  1.220130e+06  73.666667  74.315789  79.529412
73.166667
std     5.816643  5.816643e+00  17.580069  15.836149  13.010177
17.426315
min     1.000000  1.220121e+06  34.000000  45.000000  50.000000
34.000000
25%     6.000000  1.220126e+06  64.750000  62.500000  70.000000
65.500000
50%    10.000000  1.220130e+06  77.500000  77.000000  80.000000
75.000000
```

```
75%    15.000000  1.220135e+06  85.500000  86.500000  90.000000
85.500000
max    20.000000  1.220140e+06  96.000000  96.000000  96.000000
96.000000

            TOTAL         AVG
count   16.000000   20.000000
mean   272.750000   72.733333
std    102.048681   48.017127
min      0.000000    0.000000
25%    216.250000   40.750000
50%    304.000000   78.666667
75%    349.500000  113.333333
max    383.000000  127.666667
```

Handling the missing values:

```
df.isnull()
df.isnull().sum()

SNO         0
REGNO       0
NAME        1
DOB         0
GENDER      1
ADDRESS     1
M1          3
M2          2
M3          4
M4          3
TOTAL       5
AVG         1
dtype: int64
```

Fill the missing values with 0

```
df_fillna=df.fillna(0)
df_fillna

    SNO    REGNO       NAME          DOB  GENDER       ADDRESS    M1
M2    M3  \
0     1  1220121       ARUN  2000-02-10    MALE     THANDALAM  82.0
81.0  90.0
1     2  1220122       BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
2     3  1220123     CHARAN  2000.09.21    MALE     THANDALAM   0.0
59.0  60.0
3     4  1220124       DEVA  2000-11-09    MALE    POONAMALEE  74.0
79.0  80.0
```

```
4     5  1220125     ESTER  2000-11-21  FEMALE      CHITHUR  92.0
95.0  96.0
5     6  1220126   FARHANA  1999-03-05  FEMALE    THANDALAM  91.0
88.0  90.0
6     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
7     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128      HEMA  1999-01-25  FEMALE   POONAMALEE  95.0
96.0  90.0
9     9  1220129     INDRA  2000.09.21  FEMALE  KANCHIPURAM  64.0
0.0   0.0
10   10  1220130    JAHITH  2000-11-09    MALE    THANDALAM  34.0
45.0  50.0
11   11  1220131      KANI  2000-11-21  FEMALE      CHITHUR  96.0
95.0  96.0
12   12  1220132  LATHESSH  1999-03-05    MALE    THANDALAM   0.0
68.0  70.0
13   13  1220133      MANI  2000-10-02    MALE  KANCHIPURAM  71.0
76.0   0.0
14   14  1220134      NANI    20001109    MALE   POONAMALEE  79.0
77.0  80.0
15   15  1220135         0    19990125       0            0   0.0
0.0   0.0
16   16  1220136   PRATHAP    20000921    MALE  KANCHIPURAM  86.0
84.0  90.0
17   17  1220137     RAGHU    20001109    MALE   POONAMALEE  67.0
64.0  70.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
86.0  90.0
19   19  1220139   SARVESH    19990305    MALE    THANDALAM  84.0
87.0   0.0
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0

      M4  TOTAL         AVG
0    0.0    0.0    0.000000
1   56.0  253.0   84.333333
2   70.0    0.0    0.000000
3   74.0  307.0  102.333333
4   92.0  375.0  125.000000
5   91.0  360.0  120.000000
6   49.0  219.0   73.000000
7   49.0  219.0   73.000000
8   95.0  376.0  125.333333
9   64.0    0.0    0.000000
10  34.0  163.0   54.333333
11  96.0  383.0  127.666667
12  70.0  208.0   69.333333
```

```
13  71.0    0.0    0.000000
14  79.0  315.0  105.000000
15   0.0    0.0    0.000000
16  86.0  346.0  115.333333
17   0.0  201.0   67.000000
18  81.0  338.0  112.666667
19  84.0    0.0    0.000000
20  76.0  301.0  100.333333
```

Forward fill

```
df_ffill=df.ffill()
df_ffill
```

```
    SNO    REGNO      NAME          DOB  GENDER      ADDRESS    M1
M2    M3  \
0     1  1220121      ARUN  2000-02-10    MALE    THANDALAM  82.0
81.0  90.0
1     2  1220122      BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
2     3  1220123    CHARAN  2000.09.21    MALE    THANDALAM  56.0
59.0  60.0
3     4  1220124      DEVA  2000-11-09    MALE   POONAMALEE  74.0
79.0  80.0
4     5  1220125     ESTER  2000-11-21  FEMALE      CHITHUR  92.0
95.0  96.0
5     6  1220126   FARHANA  1999-03-05  FEMALE    THANDALAM  91.0
88.0  90.0
6     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
7     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128      HEMA  1999-01-25  FEMALE   POONAMALEE  95.0
96.0  90.0
9     9  1220129     INDRA  2000.09.21  FEMALE  KANCHIPURAM  64.0
96.0  90.0
10   10  1220130    JAHITH  2000-11-09    MALE    THANDALAM  34.0
45.0  50.0
11   11  1220131      KANI  2000-11-21  FEMALE      CHITHUR  96.0
95.0  96.0
12   12  1220132  LATHESSH  1999-03-05    MALE    THANDALAM  96.0
68.0  70.0
13   13  1220133      MANI  2000-10-02    MALE  KANCHIPURAM  71.0
76.0  70.0
14   14  1220134      NANI    20001109    MALE   POONAMALEE  79.0
77.0  80.0
15   15  1220135      NANI    19990125    MALE   POONAMALEE  79.0
77.0  80.0
16   16  1220136   PRATHAP    20000921    MALE  KANCHIPURAM  86.0
```

```
84.0  90.0
17   17  1220137     RAGHU    20001109    MALE   POONAMALEE  67.0
64.0  70.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
86.0  90.0
19   19  1220139   SARVESH    19990305    MALE    THANDALAM  84.0
87.0  90.0
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0

      M4  TOTAL         AVG
0    NaN    NaN         NaN
1   56.0  253.0   84.333333
2   70.0  253.0    0.000000
3   74.0  307.0  102.333333
4   92.0  375.0  125.000000
5   91.0  360.0  120.000000
6   49.0  219.0   73.000000
7   49.0  219.0   73.000000
8   95.0  376.0  125.333333
9   64.0  376.0    0.000000
10  34.0  163.0   54.333333
11  96.0  383.0  127.666667
12  70.0  208.0   69.333333
13  71.0  208.0    0.000000
14  79.0  315.0  105.000000
15  79.0    0.0    0.000000
16  86.0  346.0  115.333333
17  86.0  201.0   67.000000
18  81.0  338.0  112.666667
19  84.0  338.0    0.000000
20  76.0  301.0  100.333333
```

Backward fill

```
df_bfill=df.bfill()
df_bfill

    SNO    REGNO      NAME         DOB  GENDER      ADDRESS    M1
M2    M3  \
0     1  1220121      ARUN  2000-02-10    MALE    THANDALAM  82.0
81.0  90.0
1     2  1220122      BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
2     3  1220123    CHARAN  2000.09.21    MALE    THANDALAM  74.0
59.0  60.0
3     4  1220124      DEVA  2000-11-09    MALE   POONAMALEE  74.0
79.0  80.0
4     5  1220125     ESTER  2000-11-21  FEMALE      CHITHUR  92.0
```

```
95.0  96.0
5     6  1220126   FARHANA  1999-03-05  FEMALE    THANDALAM  91.0
88.0  90.0
6     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
7     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128      HEMA  1999-01-25  FEMALE   POONAMALEE  95.0
96.0  90.0
9     9  1220129     INDRA  2000.09.21  FEMALE  KANCHIPURAM  64.0
45.0  50.0
10   10  1220130    JAHITH  2000-11-09    MALE    THANDALAM  34.0
45.0  50.0
11   11  1220131      KANI  2000-11-21  FEMALE      CHITHUR  96.0
95.0  96.0
12   12  1220132  LATHESSH  1999-03-05    MALE    THANDALAM  71.0
68.0  70.0
13   13  1220133      MANI  2000-10-02    MALE  KANCHIPURAM  71.0
76.0  80.0
14   14  1220134      NANI    20001109    MALE   POONAMALEE  79.0
77.0  80.0
15   15  1220135   PRATHAP    19990125    MALE  KANCHIPURAM  86.0
84.0  90.0
16   16  1220136   PRATHAP    20000921    MALE  KANCHIPURAM  86.0
84.0  90.0
17   17  1220137     RAGHU    20001109    MALE   POONAMALEE  67.0
64.0  70.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
86.0  90.0
19   19  1220139   SARVESH    19990305    MALE    THANDALAM  84.0
87.0  80.0
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0

      M4  TOTAL         AVG
0   56.0  253.0   84.333333
1   56.0  253.0   84.333333
2   70.0  307.0    0.000000
3   74.0  307.0  102.333333
4   92.0  375.0  125.000000
5   91.0  360.0  120.000000
6   49.0  219.0   73.000000
7   49.0  219.0   73.000000
8   95.0  376.0  125.333333
9   64.0  163.0    0.000000
10  34.0  163.0   54.333333
11  96.0  383.0  127.666667
12  70.0  208.0   69.333333
13  71.0  315.0    0.000000
```

```
14  79.0  315.0  105.000000
15  86.0    0.0    0.000000
16  86.0  346.0  115.333333
17  81.0  201.0   67.000000
18  81.0  338.0  112.666667
19  84.0  301.0    0.000000
20  76.0  301.0  100.333333
```

Filling the missing values with mean value:

```
df['TOTAL']=df['TOTAL'].fillna(df['TOTAL'].mean())
df['AVG']=df['AVG'].fillna(df['AVG'].mean())
df
```

```
    SNO    REGNO       NAME          DOB  GENDER       ADDRESS    M1
M2    M3  \
0     1  1220121       ARUN  2000-02-10    MALE     THANDALAM  82.0
81.0  90.0
1     2  1220122       BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
2     3  1220123     CHARAN  2000.09.21    MALE     THANDALAM   NaN
59.0  60.0
3     4  1220124       DEVA  2000-11-09    MALE    POONAMALEE  74.0
79.0  80.0
4     5  1220125      ESTER  2000-11-21  FEMALE       CHITHUR  92.0
95.0  96.0
5     6  1220126    FARHANA  1999-03-05  FEMALE     THANDALAM  91.0
88.0  90.0
6     7  1220127       GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
7     7  1220127       GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128       HEMA  1999-01-25  FEMALE    POONAMALEE  95.0
96.0  90.0
9     9  1220129      INDRA  2000.09.21  FEMALE  KANCHIPURAM  64.0
NaN   NaN
10   10  1220130     JAHITH  2000-11-09    MALE     THANDALAM  34.0
45.0  50.0
11   11  1220131       KANI  2000-11-21  FEMALE       CHITHUR  96.0
95.0  96.0
12   12  1220132   LATHESSH  1999-03-05    MALE     THANDALAM   NaN
68.0  70.0
13   13  1220133       MANI  2000-10-02    MALE  KANCHIPURAM  71.0
76.0   NaN
14   14  1220134       NANI    20001109    MALE    POONAMALEE  79.0
77.0  80.0
15   15  1220135        NaN    19990125     NaN           NaN   NaN
NaN   NaN
16   16  1220136    PRATHAP    20000921    MALE  KANCHIPURAM  86.0
```

```
84.0  90.0
17   17  1220137     RAGHU    20001109    MALE   POONAMALEE  67.0
64.0  70.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
86.0  90.0
19   19  1220139   SARVESH    19990305    MALE    THANDALAM  84.0
87.0   NaN
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0

      M4   TOTAL         AVG
0    NaN  272.75   72.733333
1   56.0  253.00   84.333333
2   70.0  272.75    0.000000
3   74.0  307.00  102.333333
4   92.0  375.00  125.000000
5   91.0  360.00  120.000000
6   49.0  219.00   73.000000
7   49.0  219.00   73.000000
8   95.0  376.00  125.333333
9   64.0  272.75    0.000000
10  34.0  163.00   54.333333
11  96.0  383.00  127.666667
12  70.0  208.00   69.333333
13  71.0  272.75    0.000000
14  79.0  315.00  105.000000
15   NaN    0.00    0.000000
16  86.0  346.00  115.333333
17   NaN  201.00   67.000000
18  81.0  338.00  112.666667
19  84.0  272.75    0.000000
20  76.0  301.00  100.333333
```

Droping the missing value:

```
df_dropna=df.dropna()
df_dropna

    SNO    REGNO      NAME         DOB  GENDER      ADDRESS    M1
M2    M3  \
1     2  1220122      BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
3     4  1220124      DEVA  2000-11-09    MALE   POONAMALEE  74.0
79.0  80.0
4     5  1220125     ESTER  2000-11-21  FEMALE      CHITHUR  92.0
95.0  96.0
5     6  1220126   FARHANA  1999-03-05  FEMALE    THANDALAM  91.0
88.0  90.0
6     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
```

```
51.0  70.0
7     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128      HEMA  1999-01-25  FEMALE   POONAMALEE  95.0
96.0  90.0
10   10  1220130    JAHITH  2000-11-09    MALE    THANDALAM  34.0
45.0  50.0
11   11  1220131      KANI  2000-11-21  FEMALE      CHITHUR  96.0
95.0  96.0
14   14  1220134      NANI    20001109    MALE   POONAMALEE  79.0
77.0  80.0
16   16  1220136   PRATHAP    20000921    MALE  KANCHIPURAM  86.0
84.0  90.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
86.0  90.0
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0
```

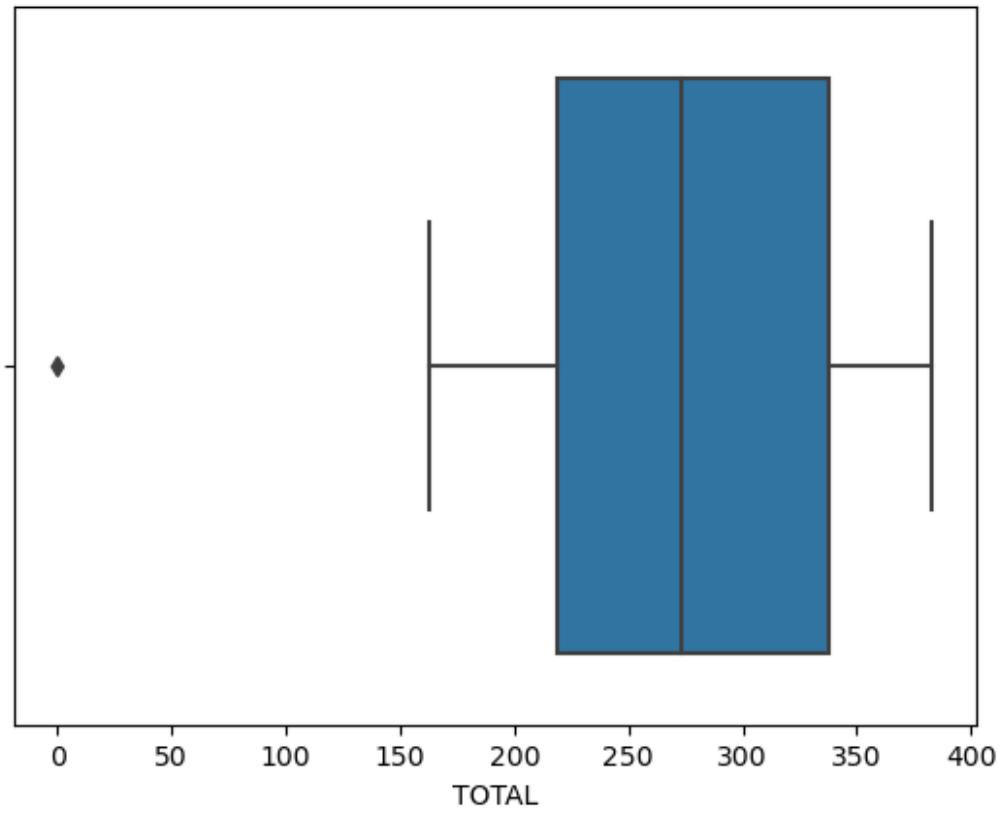
```
      M4  TOTAL         AVG
1   56.0  253.0   84.333333
3   74.0  307.0  102.333333
4   92.0  375.0  125.000000
5   91.0  360.0  120.000000
6   49.0  219.0   73.000000
7   49.0  219.0   73.000000
8   95.0  376.0  125.333333
10  34.0  163.0   54.333333
11  96.0  383.0  127.666667
14  79.0  315.0  105.000000
16  86.0  346.0  115.333333
18  81.0  338.0  112.666667
20  76.0  301.0  100.333333
```

Save cleaned data :

```
df_dropna.to_csv("cleaned_SAMPLEIDS.csv",index=False)
```

Outlier detection using IQR method:

```
sns.boxplot(x=df['TOTAL'])
plt.show()
```



Calculate IQR Value:

```
Q1=df['TOTAL'].quantile(0.25)
Q3=df['TOTAL'].quantile(0.75)
IQR=Q3-Q1
print("IQR value is",IQR)
```

```
IQR value is 119.0
```

Detecting Outliers:

```
outlier=df[(df['TOTAL']<(Q1-1.5*IQR))|(df['TOTAL']>(Q3+1.5*IQR))]
outlier
```

```
    SNO    REGNO NAME       DOB GENDER ADDRESS  M1  M2  M3  M4  TOTAL
AVG
15   15  1220135  NaN  19990125    NaN     NaN NaN NaN NaN NaN    0.0
0.0
```

Removing Outlier:

```
cleaned=df[~((df['TOTAL']<(Q1-1.5*IQR))|(df['TOTAL']>(Q3+1.5*IQR)))]
cleaned
```

```
    SNO    REGNO      NAME          DOB  GENDER      ADDRESS    M1
M2    M3  \
0     1  1220121      ARUN  2000-02-10    MALE    THANDALAM  82.0
81.0  90.0
1     2  1220122      BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
2     3  1220123    CHARAN  2000.09.21    MALE    THANDALAM   NaN
59.0  60.0
3     4  1220124      DEVA  2000-11-09    MALE   POONAMALEE  74.0
79.0  80.0
4     5  1220125     ESTER  2000-11-21  FEMALE      CHITHUR  92.0
95.0  96.0
5     6  1220126   FARHANA  1999-03-05  FEMALE    THANDALAM  91.0
88.0  90.0
6     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
7     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128      HEMA  1999-01-25  FEMALE   POONAMALEE  95.0
96.0  90.0
9     9  1220129     INDRA  2000.09.21  FEMALE  KANCHIPURAM  64.0
NaN   NaN
10   10  1220130    JAHITH  2000-11-09    MALE    THANDALAM  34.0
45.0  50.0
11   11  1220131      KANI  2000-11-21  FEMALE      CHITHUR  96.0
95.0  96.0
12   12  1220132  LATHESSH  1999-03-05    MALE    THANDALAM   NaN
68.0  70.0
13   13  1220133      MANI  2000-10-02    MALE  KANCHIPURAM  71.0
76.0   NaN
14   14  1220134      NANI    20001109    MALE   POONAMALEE  79.0
77.0  80.0
16   16  1220136   PRATHAP    20000921    MALE  KANCHIPURAM  86.0
84.0  90.0
17   17  1220137     RAGHU    20001109    MALE   POONAMALEE  67.0
64.0  70.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
86.0  90.0
19   19  1220139   SARVESH    19990305    MALE    THANDALAM  84.0
87.0   NaN
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0

      M4   TOTAL         AVG
0    NaN  272.75   72.733333
1   56.0  253.00   84.333333
2   70.0  272.75    0.000000
3   74.0  307.00  102.333333
4   92.0  375.00  125.000000
5   91.0  360.00  120.000000
```

```
6   49.0  219.00   73.000000
7   49.0  219.00   73.000000
8   95.0  376.00  125.333333
9   64.0  272.75    0.000000
10  34.0  163.00   54.333333
11  96.0  383.00  127.666667
12  70.0  208.00   69.333333
13  71.0  272.75    0.000000
14  79.0  315.00  105.000000
16  86.0  346.00  115.333333
17   NaN  201.00   67.000000
18  81.0  338.00  112.666667
19  84.0  272.75    0.000000
20  76.0  301.00  100.333333
```

Detecting Outliers using Z Score Method:

```
z_score=np.abs(stats.zscore(df['TOTAL']))
z_score
```

```
0     0.000000
1     0.228994
2     0.000000
3     0.397116
4     1.185550
5     1.011631
6     0.623211
7     0.623211
8     1.197145
9     0.000000
10    1.272510
11    1.278307
12    0.750752
13    0.000000
14    0.489873
15    3.162433
16    0.849306
17    0.831914
18    0.756549
19    0.000000
20    0.327548
Name: TOTAL, dtype: float64
```

Detecting Outliers:

```
threshold=3
outlier=df[z_score>threshold]
print("Outliers")
outlier
```

```
Outliers

    SNO    REGNO NAME       DOB GENDER ADDRESS  M1  M2  M3  M4  TOTAL
AVG
15   15  1220135  NaN  19990125    NaN     NaN NaN NaN NaN NaN    0.0
0.0
```

Removing Outliers:

```
cleaned=df[z_score<=threshold]
cleaned

    SNO    REGNO      NAME         DOB  GENDER      ADDRESS    M1
M2    M3  \
0     1  1220121      ARUN  2000-02-10    MALE    THANDALAM  82.0
81.0  90.0
1     2  1220122      BABU  1999-01-25    MALE  KANCHIPURAM  56.0
61.0  80.0
2     3  1220123    CHARAN  2000.09.21    MALE    THANDALAM   NaN
59.0  60.0
3     4  1220124      DEVA  2000-11-09    MALE   POONAMALEE  74.0
79.0  80.0
4     5  1220125     ESTER  2000-11-21  FEMALE      CHITHUR  92.0
95.0  96.0
5     6  1220126   FARHANA  1999-03-05  FEMALE    THANDALAM  91.0
88.0  90.0
6     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
7     7  1220127      GANI  2000-10-02    MALE  KANCHIPURAM  49.0
51.0  70.0
8     8  1220128      HEMA  1999-01-25  FEMALE   POONAMALEE  95.0
96.0  90.0
9     9  1220129     INDRA  2000.09.21  FEMALE  KANCHIPURAM  64.0
NaN   NaN
10   10  1220130    JAHITH  2000-11-09    MALE    THANDALAM  34.0
45.0  50.0
11   11  1220131      KANI  2000-11-21  FEMALE      CHITHUR  96.0
95.0  96.0
12   12  1220132  LATHESSH  1999-03-05    MALE    THANDALAM   NaN
68.0  70.0
13   13  1220133      MANI  2000-10-02    MALE  KANCHIPURAM  71.0
76.0   NaN
14   14  1220134      NANI    20001109    MALE   POONAMALEE  79.0
77.0  80.0
16   16  1220136   PRATHAP    20000921    MALE  KANCHIPURAM  86.0
84.0  90.0
17   17  1220137     RAGHU    20001109    MALE   POONAMALEE  67.0
64.0  70.0
18   18  1220138     RATHI    20001121  FEMALE  KANCHIPURAM  81.0
```

```
86.0  90.0
19   19  1220139   SARVESH    19990305    MALE    THANDALAM  84.0
87.0   NaN
20   20  1220140  SANTHOSH    20001002    MALE  KANCHIPURAM  76.0
69.0  80.0
```

	M4	TOTAL	AVG
0	NaN	272.75	72.733333
1	56.0	253.00	84.333333
2	70.0	272.75	0.000000
3	74.0	307.00	102.333333
4	92.0	375.00	125.000000
5	91.0	360.00	120.000000
6	49.0	219.00	73.000000
7	49.0	219.00	73.000000
8	95.0	376.00	125.333333
9	64.0	272.75	0.000000
10	34.0	163.00	54.333333
11	96.0	383.00	127.666667
12	70.0	208.00	69.333333
13	71.0	272.75	0.000000
14	79.0	315.00	105.000000
16	86.0	346.00	115.333333
17	NaN	201.00	67.000000
18	81.0	338.00	112.666667
19	84.0	272.75	0.000000
20	76.0	301.00	100.333333

RESULT:

Thus we have cleaned the data and removed the outliers by detection using IQR and Z-score method.