```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
import pandas as pd
data = pd.read_csv('C:\\Users\\admin\\Downloads\\01.Data Cleaning and
Preprocessing.csv')
type(data)
<class 'pandas.core.frame.DataFrame'>
data.info
<bound method DataFrame.info of        Observation  Y-Kappa  ...  T-Top-Chips-4
SulphidityL-4
0       31-00:00    23.10  ...        252.077           NaN
1       31-01:00    27.60  ...        251.406         29.11
2       31-02:00    23.19  ...        251.335           NaN
3       31-03:00    23.60  ...        250.312         29.02
4       31-04:00    22.90  ...        249.916         29.01
..           ...      ...  ...            ...           ...
319     10-16:00    23.75  ...        252.947         30.86
320      9-19:00    19.80  ...        252.092         30.70
321      9-20:00    23.01  ...        252.438           NaN
322      9-21:00    24.32  ...        253.176         31.13
323      9-22:00    25.75  ...        253.216           NaN

[324 rows x 23 columns]>
data.describe()
           Y-Kappa     ChipRate  ...  T-Top-Chips-4   SulphidityL-4
count   324.000000   319.000000  ...     323.000000      173.000000
mean     20.635370    14.347937  ...     251.240087       30.411671
std       3.070036     1.499095  ...       1.283432        0.701317
min      12.170000     9.983000  ...     248.359000       29.010000
25%      18.382500    13.358000  ...     250.312000       29.970000
50%      20.845000    14.308000  ...     251.380000       30.370000
75%      23.032500    15.517000  ...     252.323500       30.820000
max      27.600000    16.958000  ...     254.122000       32.840000

[8 rows x 22 columns]
data=data.drop_duplicates()
data
     Observation  Y-Kappa  ...  T-Top-Chips-4   SulphidityL-4
0       31-00:00    23.10  ...        252.077           NaN
1       31-01:00    27.60  ...        251.406         29.11
2       31-02:00    23.19  ...        251.335           NaN
3       31-03:00    23.60  ...        250.312         29.02
4       31-04:00    22.90  ...        249.916         29.01
..           ...      ...  ...            ...           ...
298     12-09:00    20.90  ...        251.833         30.29
299     12-10:00    24.98  ...        251.614         30.47
300     12-11:00    21.00  ...        251.197           NaN
301     12-12:00    21.40  ...        251.324         30.46
307     31-05:00    20.89  ...        250.084           NaN

[301 rows x 23 columns]
data.isnull().sum()
Observation              0
```

```
Y-Kappa              0
ChipRate             4
BF-CMratio          14
BlowFlow            13
ChipLevel4           1
T-upperExt-2         1
T-lowerExt-2         1
UCZAA               24
WhiteFlow-4          1
AAWhiteSt-4        141
AA-Wood-4            1
ChipMoisture-4       1
SteamFlow-4          1
Lower-HeatT-3        1
Upper-HeatT-3        1
ChipMass-4           1
WeakLiquorF          1
BlackFlow-2          1
WeakWashF            1
SteamHeatF-3         1
T-Top-Chips-4        1
SulphidityL-4      141
dtype: int64
```

`data.notnull()`

```
     Observation   Y-Kappa  ...  T-Top-Chips-4  SulphidityL-4
0           True      True  ...           True          False
1           True      True  ...           True           True
2           True      True  ...           True          False
3           True      True  ...           True           True
4           True      True  ...           True           True
..           ...       ...  ...            ...            ...
298         True      True  ...           True           True
299         True      True  ...           True           True
300         True      True  ...           True          False
301         True      True  ...           True           True
307         True      True  ...           True          False

[301 rows x 23 columns]
```

`data.isnull().sum().sum()`

`np.int64(352)`

`data2 = data.fillna(value=0)`

`data2`

```
     Observation   Y-Kappa  ...  T-Top-Chips-4  SulphidityL-4
0       31-00:00     23.10  ...        252.077           0.00
1       31-01:00     27.60  ...        251.406          29.11
2       31-02:00     23.19  ...        251.335           0.00
3       31-03:00     23.60  ...        250.312          29.02
4       31-04:00     22.90  ...        249.916          29.01
..           ...       ...  ...            ...            ...
298     12-09:00     20.90  ...        251.833          30.29
299     12-10:00     24.98  ...        251.614          30.47
300     12-11:00     21.00  ...        251.197           0.00
301     12-12:00     21.40  ...        251.324          30.46
307     31-05:00     20.89  ...        250.084           0.00
```

```
[301 rows x 23 columns]
data2.isnull().sum().sum()
np.int64(0)
data
     Observation  Y-Kappa  ...  T-Top-Chips-4  SulphidityL-4
0       31-00:00    23.10  ...        252.077            NaN
1       31-01:00    27.60  ...        251.406          29.11
2       31-02:00    23.19  ...        251.335            NaN
3       31-03:00    23.60  ...        250.312          29.02
4       31-04:00    22.90  ...        249.916          29.01
..           ...      ...  ...            ...            ...
298     12-09:00    20.90  ...        251.833          30.29
299     12-10:00    24.98  ...        251.614          30.47
300     12-11:00    21.00  ...        251.197            NaN
301     12-12:00    21.40  ...        251.324          30.46
307     31-05:00    20.89  ...        250.084            NaN

[301 rows x 23 columns]
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
data2.columns
Index(['Observation', 'Y-Kappa', 'ChipRate', 'BF-CMratio', 'BlowFlow',
       'ChipLevel4 ', 'T-upperExt-2 ', 'T-lowerExt-2  ', 'UCZAA',
       'WhiteFlow-4 ', 'AAWhiteSt-4 ', 'AA-Wood-4  ', 'ChipMoisture-4 ',
       'SteamFlow-4 ', 'Lower-HeatT-3', 'Upper-HeatT-3 ', 'ChipMass-4 ',
       'WeakLiquorF ', 'BlackFlow-2 ', 'WeakWashF ', 'SteamHeatF-3 ',
       'T-Top-Chips-4 ', 'SulphidityL-4 '],
      dtype='object')
data2.drop(['Observation'], axis=1, inplace=True)
data2.columns
Index(['Y-Kappa', 'ChipRate', 'BF-CMratio', 'BlowFlow', 'ChipLevel4 ',
       'T-upperExt-2 ', 'T-lowerExt-2  ', 'UCZAA', 'WhiteFlow-4 ',
       'AAWhiteSt-4 ', 'AA-Wood-4  ', 'ChipMoisture-4 ', 'SteamFlow-4 ',
       'Lower-HeatT-3', 'Upper-HeatT-3 ', 'ChipMass-4 ', 'WeakLiquorF ',
       'BlackFlow-2 ', 'WeakWashF ', 'SteamHeatF-3 ', 'T-Top-Chips-4 ',
       'SulphidityL-4 '],
      dtype='object')
Q1= data2.quantile(0.25)
Q3= data2.quantile(0.75)
IQR=Q3-Q1
print(IQR)
Y-Kappa                4.550
ChipRate               2.233
BF-CMratio            10.912
BlowFlow              96.766
ChipLevel4           105.868
T-upperExt-2          11.994
T-lowerExt-2           7.609
UCZAA                  0.152
WhiteFlow-4          100.098
AAWhiteSt-4            6.143
AA-Wood-4             1.486
```

```
ChipMoisture-4         2.186
SteamFlow-4            8.840
Lower-HeatT-3          8.585
Upper-HeatT-3          7.852
ChipMass-4           19.347
WeakLiquorF         180.613
BlackFlow-2         280.829
WeakWashF           267.219
SteamHeatF-3          6.903
T-Top-Chips-4         2.044
SulphidityL-4        30.420
dtype: float64
>>> data2=data2[~((data2<(Q1-1.5*IQR))|(data2>(Q3+1.5*IQR))).any(axis=1)]
>>> data2
     Y-Kappa  ChipRate  ...  T-Top-Chips-4  SulphidityL-4
1      27.60    16.810  ...        251.406          29.11
2      23.19    16.709  ...        251.335           0.00
3      23.60    16.478  ...        250.312          29.02
5      14.23    15.350  ...        249.580          30.34
6      13.49    13.700  ...        248.741           0.00
..       ...       ...  ...            ...            ...
276    22.70    15.517  ...        252.216          29.59
296    20.50    13.358  ...        252.423          30.43
297    20.40    14.233  ...        252.311           0.00
298    20.90    15.167  ...        251.833          30.29
307    20.89    14.308  ...        250.084           0.00

[226 rows x 22 columns]
>>> data2.describe()
          Y-Kappa    ChipRate  ...  T-Top-Chips-4  SulphidityL-4
count  226.000000  226.000000  ...     226.000000     226.000000
mean    20.690487   14.673491  ...     251.177779      15.391987
std      2.982916    1.297369  ...       1.221296      15.297984
min     12.480000   10.833000  ...     248.359000       0.000000
25%     18.457500   13.850000  ...     250.290750       0.000000
50%     20.775000   14.729000  ...     251.233000      29.065000
75%     23.010000   15.708000  ...     252.240000      30.437500
max     27.600000   16.958000  ...     254.122000      32.840000

[8 rows x 22 columns]
```