In [1]:
```
# importing libraries
import numpy as np
import pandas as pd
```

In [2]:
```
# importing dataset
data = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\Internships\Main Flow Internsh
data.head(10)
```

Out[2]:

	Observation	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	UCZAA	W
0	31-00.00	23.10	16.520	121.717	1177.607	169.805	358.282	329.545	1.443	
1	31-01.00	27.60	16.810	79.022	1328.360	341.327	351.050	329.067	1.549	
2	31-02.00	23.19	16.709	79.562	1329.407	239.161	350.022	329.260	1.600	
3	31-03.00	23.60	16.478	81.011	1334.877	213.527	350.938	331.142	1.604	
4	31-04.00	22.90	15.618	93.244	1334.168	243.131	351.640	332.709	NaN	
5	1-08.00	14.23	15.350	85.518	1171.604	198.538	344.014	325.195	1.436	
6	1-09.00	13.49	13.700	98.186	1243.688	116.275	346.208	326.982	1.434	
7	31-06.00	22.65	14.100	91.887	1307.852	288.989	352.321	331.162	1.468	
8	31-07.00	22.50	14.233	97.249	1346.900	330.325	352.687	328.894	1.480	
9	31-08.00	24.70	13.850	96.208	1334.892	362.511	352.372	327.358	1.515	

10 rows × 23 columns

In [3]:
```
# details about the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 324 entries, 0 to 323
Data columns (total 23 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Observation     324 non-null    object
 1   Y-Kappa         324 non-null    float64
 2   ChipRate        319 non-null    float64
 3   BF-CMratio      307 non-null    float64
 4   BlowFlow        308 non-null    float64
 5   ChipLevel4      323 non-null    float64
 6   T-upperExt-2    322 non-null    float64
 7   T-lowerExt-2    322 non-null    float64
 8   UCZAA           299 non-null    float64
 9   WhiteFlow-4     323 non-null    float64
 10  AAWhiteSt-4     173 non-null    float64
 11  AA-Wood-4       323 non-null    float64
 12  ChipMoisture-4  323 non-null    float64
 13  SteamFlow-4     323 non-null    float64
 14  Lower-HeatT-3   322 non-null    float64
 15  Upper-HeatT-3   322 non-null    float64
 16  ChipMass-4      323 non-null    float64
 17  WeakLiquorF     323 non-null    float64
 18  BlackFlow-2     323 non-null    float64
 19  WeakWashF       323 non-null    float64
 20  SteamHeatF-3    322 non-null    float64
 21  T-Top-Chips-4   323 non-null    float64
 22  SulphidityL-4   173 non-null    float64
dtypes: float64(22), object(1)
memory usage: 58.3+ KB
```

In [4]:
```
# size of the data set;
data.shape
```

Out[4]: (324, 23)

In [5]:
```
# checking null values()
data.isnull()
```

Out[5]:

	Observation	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	UCZAA
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	True
...	...	...	...	...	...	...	...	...	...
319	False	False	False	False	False	False	False	False	False
320	False	False	False	False	False	False	False	False	False
321	False	False	False	False	False	False	False	False	False
322	False	False	False	False	False	False	False	False	False
323	False	False	False	False	False	False	False	False	False

324 rows × 23 columns

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
Observation       0
Y-Kappa           0
ChipRate          5
BF-CMratio       17
BlowFlow         16
ChipLevel4        1
T-upperExt-2      2
T-lowerExt-2      2
UCZAA            25
WhiteFlow-4       1
AAWhiteSt-4     151
AA-Wood-4         1
ChipMoisture-4    1
SteamFlow-4       1
Lower-HeatT-3     2
Upper-HeatT-3     2
ChipMass-4        1
WeakLiquorF       1
BlackFlow-2       2
WeakWashF         1
SteamHeatF-3      2
T-Top-Chips-4     1
SulphidityL-4   151
dtype: int64
```

In [7]:
```
# persentage of null data that is present per column
(data.isnull().sum()/data.shape[0])*100
```

Out[7]:
```
Observation      0.000000
Y-Kappa          0.000000
ChipRate         1.543210
BF-CMratio       5.246914
BlowFlow         4.938272
ChipLevel4       0.308642
T-upperExt-2     0.617284
T-lowerExt-2     0.617284
UCZAA            7.716049
WhiteFlow-4      0.308642
AAWhiteSt-4     46.604938
AA-Wood-4        0.308642
ChipMoisture-4   0.308642
SteamFlow-4      0.308642
Lower-HeatT-3    0.617284
Upper-HeatT-3    0.617284
ChipMass-4       0.308642
WeakLiquorF      0.308642
BlackFlow-2      0.617284
WeakWashF        0.308642
SteamHeatF-3     0.617284
T-Top-Chips-4    0.308642
SulphidityL-4   46.604938
dtype: float64
```

In [8]:
```
# total persentage of null data in the entire data set
(data.isnull().sum().sum()/(data.shape[0]*data.shape[1]))*100
```

Out[8]: 5.179817498658078

In [9]:
```
data.columns
```

Out[9]:
```
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')
```

In [10]:
```
# handling missing values
# removing colums with maximum numbers of null values present
data.drop(columns=['AAWhiteSt-4 ','SulphidityL-4 '],inplace=True)
```

In [11]:
```
# checking how many colums remain
data.shape
```

Out[11]: (324, 21)

In [12]:
```
# filling null values with average values
  # collecting all numerical datatype columns
lst = data.select_dtypes(include="float64").columns
lst
```

Out[12]:
```
Index(['Y-Kappa', 'ChipRate', 'BF-CMratio', 'BlowFlow', 'ChipLevel4 ',
       'T-upperExt-2 ', 'T-lowerExt-2 ', 'UCZAA', 'WhiteFlow-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 '],
      dtype='object')
```

In [13]:
```
# fill the null values with mean values
for i in lst:
    data[i].fillna(data[i].mean(),inplace=True)
```

In [14]:
```
# checking missing values
data.isnull().sum()
```

Out[14]:
```
Observation      0
Y-Kappa          0
ChipRate         0
BF-CMratio       0
BlowFlow         0
ChipLevel4       0
T-upperExt-2     0
T-lowerExt-2     0
UCZAA            0
WhiteFlow-4      0
AA-Wood-4        0
ChipMoisture-4   0
SteamFlow-4      0
Lower-HeatT-3    0
Upper-HeatT-3    0
ChipMass-4       0
WeakLiquorF      0
BlackFlow-2      0
WeakWashF        0
SteamHeatF-3     0
T-Top-Chips-4    0
dtype: int64
```

In [15]:
```
data.drop(columns=["Observation"],inplace=True)
```

In [16]:
```
# removing outlier
# function to remove outlier using IQR
def remove_outliers(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    lower_lmt = Q1 - 1.5*IQR
    upper_lmt = Q3 + 1.5*IQR
    data_out = data[~((data<lower_lmt)|(data>upper_lmt)).any(axis=1)]
    return data_out
data_cleaned = remove_outliers(data)
data_cleaned.head()
```

Out[16]:

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	UCZAA	WhiteFlow-4	W
1	27.60	16.810	79.022	1328.360	341.327	351.050	329.067	1.54900	537.201	16
2	23.19	16.709	79.562	1329.407	239.161	350.022	329.260	1.60000	549.611	16
3	23.60	16.478	81.011	1334.877	213.527	350.938	331.142	1.60400	623.362	16
4	22.90	15.618	93.244	1334.168	243.131	351.640	332.709	1.49201	638.672	16
5	14.23	15.350	85.518	1171.604	198.538	344.014	325.195	1.43600	628.245	15

In [17]:
```
data.shape
```

Out[17]: (324, 20)

In [18]:
```
# showing all the statistical measures
data_cleaned.describe()
```

Out[18]:

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	T
count	261.000000	261.000000	261.000000	261.000000	261.000000	261.000000	261.000000	2
mean	20.795747	14.671888	86.223664	1256.048101	259.986492	356.753713	325.202893	
std	3.036493	1.285011	6.752980	49.136231	72.555861	7.513533	5.704286	
min	12.480000	10.833000	68.645000	1084.083000	52.241000	340.222000	310.421000	
25%	18.640000	13.850000	81.279000	1220.750000	215.022000	350.938000	321.956000	
50%	20.900000	14.683000	85.518000	1278.006000	267.787000	356.787000	326.178000	
75%	23.190000	15.708000	91.706000	1289.992000	311.776000	361.477000	329.260000	
max	27.600000	16.958000	105.911000	1351.240000	394.234000	375.047000	337.012000	

In [ ]:
```

```