**Description:**

This task involves using the Pandas library to manipulate data.

1. **Importing Libraries & Loading the CSV file:**

```
[2]: import pandas as pd

     # Load the CSV file into a DataFrame
     file_path = '01.Data Cleaning and Preprocessing.csv'
     df = pd.read_csv(file_path)
```

```
[ ]:
```

2. **Displaying Initial Rows and DataFrame Information:**

```
[3]: print("First few rows of the DataFrame:")
     print(df.head())

     print("\nDataFrame Information:")
     print(df.info())
```

```
First few rows of the DataFrame:
  Observation  Y-Kappa  ChipRate  BF-CMratio  BlowFlow  ChipLevel4   \
0    31-00:00    23.10    16.520     121.717  1177.607      169.805
1    31-01:00    27.60    16.810      79.022  1328.360      341.327
2    31-02:00    23.19    16.709      79.562  1329.407      239.161
3    31-03:00    23.60    16.478      81.011  1334.877      213.527
4    31-04:00    22.90    15.618      93.244  1334.168      243.131

   T-upperExt-2   T-lowerExt-2    UCZAA  WhiteFlow-4   ...  SteamFlow-4   \
0       358.282        329.545  1.443       599.253  ...        67.122
1       351.050        329.067  1.549       537.201  ...        60.012
2       350.022        329.260  1.600       549.611  ...        61.304
3       350.938        331.142  1.604       623.362  ...        68.496
4       351.640        332.709    NaN       638.672  ...        70.022

   Lower-HeatT-3  Upper-HeatT-3   ChipMass-4   WeakLiquorF   BlackFlow-2   \
0       329.432        303.099      175.964      1127.197      1319.039
1       330.823        304.879      163.202       665.975      1297.317
2       329.140        303.383      164.013       677.534      1327.072
3       328.875        302.254      181.487       767.853      1324.461
4       328.352        300.954      183.929       888.448      1343.424

   WeakWashF   SteamHeatF-3   T-Top-Chips-4   SulphidityL-4
0     257.325         54.612         252.077             NaN
1     241.182         46.603         251.406           29.11
2     237.272         51.795         251.335             NaN
3     239.478         54.846         250.312           29.02
4     215.372         54.186         249.916           29.01
```

```
[5 rows x 23 columns]

DataFrame Information:
<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
```

```
 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      322 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: 57.0+ KB
None
```

3. **Filtering Data Based on Conditions:**

```
[4]: filtered_df = df[df['Y-Kappa'] > 25]
     print("\nFiltered DataFrame (Y-Kappa > 25):")
     print(filtered_df.head())
```

```
Filtered DataFrame (Y-Kappa > 25):
   Observation  Y-Kappa  ChipRate  BF-CMratio  BlowFlow  ChipLevel4   \
1     31-01:00    27.60    16.810      79.022  1328.360      341.327
12    31-11:00    26.62    15.467      84.447  1334.255      386.971
13    31-12:00    27.20    16.083      82.839  1332.331      366.855
15    31-14:00    25.40    16.425      72.924  1197.775      118.821
40     1-15:00    27.10    13.558      83.117  1175.417      289.256

    T-upperExt-2   T-lowerExt-2    UCZAA  WhiteFlow-4   ...  SteamFlow-4   \
1         351.050        329.067  1.549       537.201  ...        60.012
12        349.392        321.021  1.428       531.250  ...        59.407
13        350.094        327.439  1.486       527.893  ...        60.271
15        350.765        329.799  1.635       585.011  ...        65.474
40        339.168        318.386  1.360       480.184  ...        48.568

    Lower-HeatT-3  Upper-HeatT-3   ChipMass-4   WeakLiquorF   BlackFlow-2   \
1         330.823        304.879      163.202       665.975      1297.317
12        330.284        303.248      156.797       799.947      1299.782
13        330.023        302.883      160.562       771.158      1299.974
15        329.773        302.884      175.646       756.154      1300.037
40        318.228        294.850      131.537       744.659       996.046

    WeakWashF   SteamHeatF-3   T-Top-Chips-4   SulphidityL-4
1     241.182         46.603         251.406           29.11
12    118.901         46.597         251.721             NaN
13    153.647         47.175         251.767           30.18
15    401.418         54.628         251.009           30.41
40    118.899         41.985         253.450             NaN

[5 rows x 23 columns]
```

```
]:
```

4. **Handling Missing Values:**

```
[5]: filled_df = df.fillna(df.mean())
print("\nDataFrame with missing values filled with mean:")
print(filled_df.head())
```

```
<ipython-input-5-11716a1e1554>:1: FutureWarning: The default value of numeric_only
In a future version, it will default to False. In addition, specifying 'numeric_on
ly valid columns or specify the value of numeric_only to silence this warning.
  filled_df = df.fillna(df.mean())

DataFrame with missing values filled with mean:
  Observation  Y-Kappa  ChipRate  BF-CMratio  BlowFlow  ChipLevel4   \
0    31-00:00    23.10    16.520     121.717  1177.607      169.805
1    31-01:00    27.60    16.810      79.022  1328.360      341.327
2    31-02:00    23.19    16.709      79.562  1329.407      239.161
3    31-03:00    23.60    16.478      81.011  1334.877      213.527
4    31-04:00    22.90    15.618      93.244  1334.168      243.131

   T-upperExt-2   T-lowerExt-2      UCZAA  WhiteFlow-4   ...  SteamFlow-4   \
0        358.282        329.545  1.44300       599.253  ...        67.122
1        351.050        329.067  1.54900       537.201  ...        60.012
2        350.022        329.260  1.60000       549.611  ...        61.304
3        350.938        331.142  1.60400       623.362  ...        68.496
4        351.640        332.709  1.49201       638.672  ...        70.022

   Lower-HeatT-3  Upper-HeatT-3   ChipMass-4   WeakLiquorF   BlackFlow-2   \
```

```
   Lower-HeatT-3  Upper-HeatT-3   ChipMass-4   WeakLiquorF   BlackFlow-2   \
0        329.432        303.099      175.964      1127.197      1319.039
1        330.823        304.879      163.202       665.975      1297.317
2        329.140        303.383      164.013       677.534      1327.072
3        328.875        302.254      181.487       767.853      1324.461
4        328.352        300.954      183.929       888.448      1343.424

   WeakWashF   SteamHeatF-3   T-Top-Chips-4   SulphidityL-4
0     257.325         54.612         252.077       30.411671
1     241.182         46.603         251.406       29.110000
2     237.272         51.795         251.335       30.411671
3     239.478         54.846         250.312       29.020000
4     215.372         54.186         249.916       29.010000

[5 rows x 23 columns]
```

```
]:
```

5. **Calculating Summary Statistics:**

```
[6]: summary_stats = df.describe()
     print("\nSummary statistics for numeric columns:")
     print(summary_stats)
```

```
Summary statistics for numeric columns:
          Y-Kappa    ChipRate  BF-CMratio     BlowFlow  ChipLevel4   \
count  324.000000  319.000000  307.000000   308.000000   323.000000
mean    20.635370   14.347937   87.464456  1237.837614   258.164483
std      3.070036    1.499095    7.995012   100.593735    87.987452
min     12.170000    9.983000   68.645000     0.000000     0.000000
25%     18.382500   13.358000   81.823000  1193.215250   213.527000
50%     20.845000   14.308000   86.739000  1273.138500   271.792000
75%     23.032500   15.517000   92.372000  1289.196000   321.680000
max     27.600000   16.958000  121.717000  1351.240000   419.014000

       T-upperExt-2  T-lowerExt-2         UCZAA  WhiteFlow-4  AAWhiteSt-4   \
count     322.000000    322.000000  299.000000   323.000000   173.000000
mean      356.904295    324.020180    1.492010   591.732260     6.140410
std         9.209290      7.621402    0.105923    67.016351     0.081609
min       339.168000    284.633000    1.182000   405.111000     5.890000
25%       350.241250    321.420000    1.431500   540.989500     6.089000
50%       356.843000    325.669000    1.498000   592.895000     6.135000
75%       362.242250    329.175000    1.560500   639.480500     6.199000
max       399.135000    337.012000    1.747000   731.394000     6.340000
```

```
       ...  SteamFlow-4  Lower-HeatT-3  Upper-HeatT-3  ChipMass-4   \
count  ...   323.000000     322.000000     322.000000  323.000000
mean   ...    66.668285     325.567820     300.525699  162.222322
std    ...     5.708587       4.609862       4.568484   14.160688
min    ...    48.568000     318.051000     293.312000  113.922000
25%    ...    62.518000     321.385500     296.513250  153.032500
50%    ...    67.429000     324.741000     299.126000  163.690000
75%    ...    71.522000     329.845250     304.244750  172.555000
max    ...    76.147000     333.854000     311.146000  189.268000

       WeakLiquorF  BlackFlow-2  WeakWashF  SteamHeatF-3  T-Top-Chips-4   \
count   323.000000   322.000000  323.000000    322.000000     323.000000
mean    873.828941  1175.917016  263.543068     49.696907     251.240087
std     122.073521   149.334010  163.666942      4.551909       1.283432
min     486.938000   838.948000    0.000000     35.510000     248.359000
25%     792.019500  1044.817500  134.649000     46.389750     250.312000
50%     865.254000  1150.221500  269.193000     50.277000     251.380000
75%     965.286500  1319.021250  405.563000     53.294250     252.323500
max    1226.277000  1395.767000  715.715000     63.332000     254.122000
```

| | SulphidityL-4 |
|---|---|
| count | 173.000000 |
| mean | 30.411671 |
| std | 0.701317 |
| min | 29.010000 |
| 25% | 29.970000 |
| 50% | 30.370000 |
| 75% | 30.820000 |
| max | 32.840000 |

[8 rows x 22 columns]