example

March 29, 2023

1 Basic on dataframe

[110]:

```
import pandas as pd
mydataset = {
  'cars': ["BMW", "Volvo", "Ford"],
  'passings': [3, 7, 2]
}
print(pd.DataFrame(mydataset))

data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
print(pd.DataFrame(data, index=["Feb", "Mar", "Apr"]))
```

```
    cars  passings
0    BMW         3
1  Volvo         7
2   Ford         2
     calories  duration
Feb       420        50
Mar       380        40
Apr       390        45
```

2 check the pandas.Series()

[111]:

```
myvar = pd.Series([2,4,6], index = ["x", "y", "z"])
print(myvar.to_string)



calories = {"day1": {"Cal": 23, "Steps": 10_000}, "day2": 380, "day3": 390}
myvar = pd.Series(calories)
print(myvar)
```

```
<bound method Series.to_string of x    2
y    4
z    6
dtype: int64>
```

```
day1    {'Cal': 23, 'Steps': 10000}
day2                            380
day3                            390
dtype: object
```

3 Operations on Example-data.csv

[112]:

```
import pandas as pd

#You can check your system's maximum rows with the pd.options.display.max_rows␣
 ↪statement.
print(f"MAX_ROWS = {pd.options.display.max_rows}")

df = pd.read_csv("example-data.csv")

print(df)
```

```
MAX_ROWS = 60
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
```

3.1 lets print complete dataset

[113]:

```
# lets print all the first 30 dataset in string format
print(df.head(30).to_string())
# print out the dataset information
print(df.info())
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
```

```
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
17        45     90       112       NaN
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
27        60    103       132       NaN
28        60    100       132     280.0
29        60    102       129     380.3
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  169 non-null    int64
 1   Pulse     169 non-null    int64
 2   Maxpulse  169 non-null    int64
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
```

Inference from dateframe.info() is calories has `169 - 164` no of NULL entries

3.2 Lets remove the NULL entries

[114]:

```python
print(df.head(30).dropna().to_string()) # removed the NULL entries
# the above command didn't change the orginal dataframe

# to change the original dataframe
df.dropna(inplace=True)
print(df.head(30).to_string())
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
```

```
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
28        60    100       132     280.0
29        60    102       129     380.3
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
```

```
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
28        60    100       132     280.0
29        60    102       129     380.3
30        60     92       115     243.0
31        45     90       112     180.1
```

3.3 replace the missing values from STANDARD GLOBAL CONSTRANT

[115]:

```
# if we want to replace all NaN entries to some data (contrain: other␣
 ↪attributes must be of the type we replace the data with)

df1 = pd.read_csv("example-data.csv")

printDf = df1.fillna(999999)
print(printDf.head(30).to_string())

# we want to replace Nan specific to a given attribute
df = pd.read_csv("example-data.csv")
df["Calories"].fillna(1300, inplace=True)
print(df.head(30).to_string())
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
17        45     90       112  999999.0
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
```

```
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
27        60    103       132  999999.0
28        60    100       132     280.0
29        60    102       129     380.3
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
17        45     90       112    1300.0
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
27        60    103       132    1300.0
28        60    100       132     280.0
29        60    102       129     380.3
```

3.4 replace the mean, median, mode

[116]:

```
df = pd.read_csv("example-data.csv")
# x = df["Calories"].mean()
# x = df["Calories"].median()
x = df["Calories"].mode()[0]
```

```
df["Calories"].fillna(x, inplace = True)
print(df.head(30).to_string())
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
17        45     90       112     300.0
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
27        60    103       132     300.0
28        60    100       132     280.0
29        60    102       129     380.3
```

4 Operations in Dataset data.csv

[117]:

```
df = pd.read_csv("data.csv")
df['Date'] = pd.to_datetime(df['Date'], format="%d-%m-%Y")
print(df.to_string())
```

```
    Duration  Pulse  Maxpulse  Calories       Date
0         60    110       130     409.1 2020-12-01
1         60    117       145     479.0 2020-12-02
2         60    103       135     340.0 2020-12-03
3         45    109       175     282.4 2020-12-04
4         45    117       148     406.0 2020-12-05
```

```
5         60    102        127     300.0 2020-12-06
6         60    110        136     374.0 2020-12-07
7         45    104        134     253.3 2020-12-08
8         45    104        134     253.3 2020-12-08
9         30    109        133     195.1 2020-12-09
10        60     98        124     269.0 2020-12-10
11        60    103        147     329.3 2020-12-11
12        60    100        120     250.7 2020-12-12
13        60    106        128     345.3 2020-12-13
14        60    104        132     379.3 2020-12-14
15        60     98        123     275.0 2020-12-15
16        60     98        120     215.2 2020-12-16
17        60    100        120     300.0 2020-12-17
18        45     90        112       NaN 2020-12-18
19        60    103        123     323.0 2020-12-19
20        45     97        125     243.0 2020-12-20
21        60    108        131     364.2 2020-12-21
22        45    100        119     282.0 2020-12-22
23        60    130        101     300.0 2020-12-23
24        45    105        132     246.0 2020-12-24
25        60    102        126     334.5 2020-12-25
26        60    100        120     250.0 2020-12-26
27        60     92        118     241.0 2020-12-27
28        60    103        132       NaN 2020-12-28
29        60    100        132     280.0 2020-12-29
30        60    102        129     380.3 2020-12-30
```

4.1 drop the row having Nan

[118]:
```
print(df.dropna().to_string())
```

```
    Duration  Pulse  Maxpulse  Calories        Date
0         60    110        130     409.1 2020-12-01
1         60    117        145     479.0 2020-12-02
2         60    103        135     340.0 2020-12-03
3         45    109        175     282.4 2020-12-04
4         45    117        148     406.0 2020-12-05
5         60    102        127     300.0 2020-12-06
6         60    110        136     374.0 2020-12-07
7         45    104        134     253.3 2020-12-08
8         45    104        134     253.3 2020-12-08
9         30    109        133     195.1 2020-12-09
10        60     98        124     269.0 2020-12-10
11        60    103        147     329.3 2020-12-11
12        60    100        120     250.7 2020-12-12
13        60    106        128     345.3 2020-12-13
14        60    104        132     379.3 2020-12-14
15        60     98        123     275.0 2020-12-15
```

```
16        60     98       120     215.2 2020-12-16
17        60    100       120     300.0 2020-12-17
19        60    103       123     323.0 2020-12-19
20        45     97       125     243.0 2020-12-20
21        60    108       131     364.2 2020-12-21
22        45    100       119     282.0 2020-12-22
23        60    130       101     300.0 2020-12-23
24        45    105       132     246.0 2020-12-24
25        60    102       126     334.5 2020-12-25
26        60    100       120     250.0 2020-12-26
27        60     92       118     241.0 2020-12-27
29        60    100       132     280.0 2020-12-29
30        60    102       129     380.3 2020-12-30
```

4.2 replacing data in specific row and attribute

[119]:

```
df.loc[6, 'Duration'] = 500000
print(df)
```

```
    Duration  Pulse  Maxpulse  Calories        Date
0         60    110       130     409.1 2020-12-01
1         60    117       145     479.0 2020-12-02
2         60    103       135     340.0 2020-12-03
3         45    109       175     282.4 2020-12-04
4         45    117       148     406.0 2020-12-05
5         60    102       127     300.0 2020-12-06
6     500000    110       136     374.0 2020-12-07
7         45    104       134     253.3 2020-12-08
8         45    104       134     253.3 2020-12-08
9         30    109       133     195.1 2020-12-09
10        60     98       124     269.0 2020-12-10
11        60    103       147     329.3 2020-12-11
12        60    100       120     250.7 2020-12-12
13        60    106       128     345.3 2020-12-13
14        60    104       132     379.3 2020-12-14
15        60     98       123     275.0 2020-12-15
16        60     98       120     215.2 2020-12-16
17        60    100       120     300.0 2020-12-17
18        45     90       112       NaN 2020-12-18
19        60    103       123     323.0 2020-12-19
20        45     97       125     243.0 2020-12-20
21        60    108       131     364.2 2020-12-21
22        45    100       119     282.0 2020-12-22
23        60    130       101     300.0 2020-12-23
24        45    105       132     246.0 2020-12-24
25        60    102       126     334.5 2020-12-25
26        60    100       120     250.0 2020-12-26
27        60     92       118     241.0 2020-12-27
```

```
28        60    103       132       NaN 2020-12-28
29        60    100       132     280.0 2020-12-29
30        60    102       129     380.3 2020-12-30
```

[120]:

```
print(df.info())
for x in df.index:
  if df.loc[x, 'Maxpulse'] > 120:
    df.loc[x, "Maxpulse"] = 120
print(df.to_string())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  31 non-null     int64
 1   Pulse     31 non-null     int64
 2   Maxpulse  31 non-null     int64
 3   Calories  29 non-null     float64
 4   Date      31 non-null     datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(3)
memory usage: 1.3 KB
None
    Duration  Pulse  Maxpulse  Calories       Date
0         60    110       120     409.1 2020-12-01
1         60    117       120     479.0 2020-12-02
2         60    103       120     340.0 2020-12-03
3         45    109       120     282.4 2020-12-04
4         45    117       120     406.0 2020-12-05
5         60    102       120     300.0 2020-12-06
6     500000    110       120     374.0 2020-12-07
7         45    104       120     253.3 2020-12-08
8         45    104       120     253.3 2020-12-08
9         30    109       120     195.1 2020-12-09
10        60     98       120     269.0 2020-12-10
11        60    103       120     329.3 2020-12-11
12        60    100       120     250.7 2020-12-12
13        60    106       120     345.3 2020-12-13
14        60    104       120     379.3 2020-12-14
15        60     98       120     275.0 2020-12-15
16        60     98       120     215.2 2020-12-16
17        60    100       120     300.0 2020-12-17
18        45     90       112       NaN 2020-12-18
19        60    103       120     323.0 2020-12-19
20        45     97       120     243.0 2020-12-20
21        60    108       120     364.2 2020-12-21
22        45    100       119     282.0 2020-12-22
23        60    130       101     300.0 2020-12-23
```

```
24        45    105       120     246.0 2020-12-24
25        60    102       120     334.5 2020-12-25
26        60    100       120     250.0 2020-12-26
27        60     92       118     241.0 2020-12-27
28        60    103       120       NaN 2020-12-28
29        60    100       120     280.0 2020-12-29
30        60    102       120     380.3 2020-12-30
```

4.3 remove duplicates

[121]:

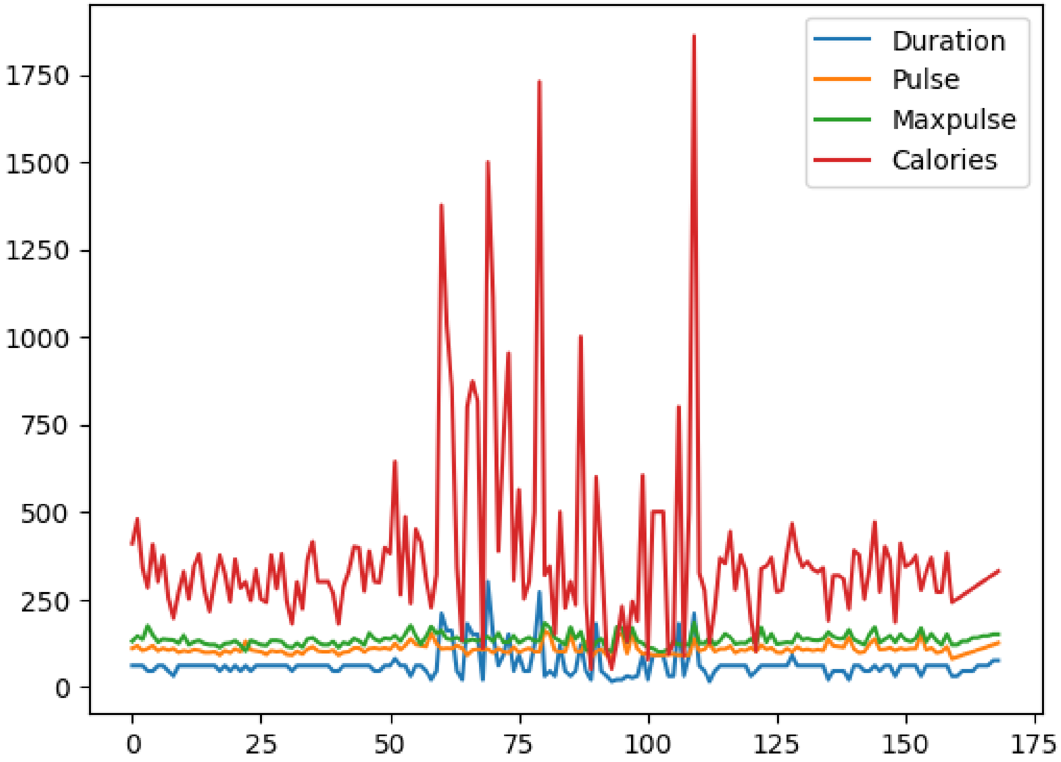
```
print(df.drop_duplicates().to_string())
```

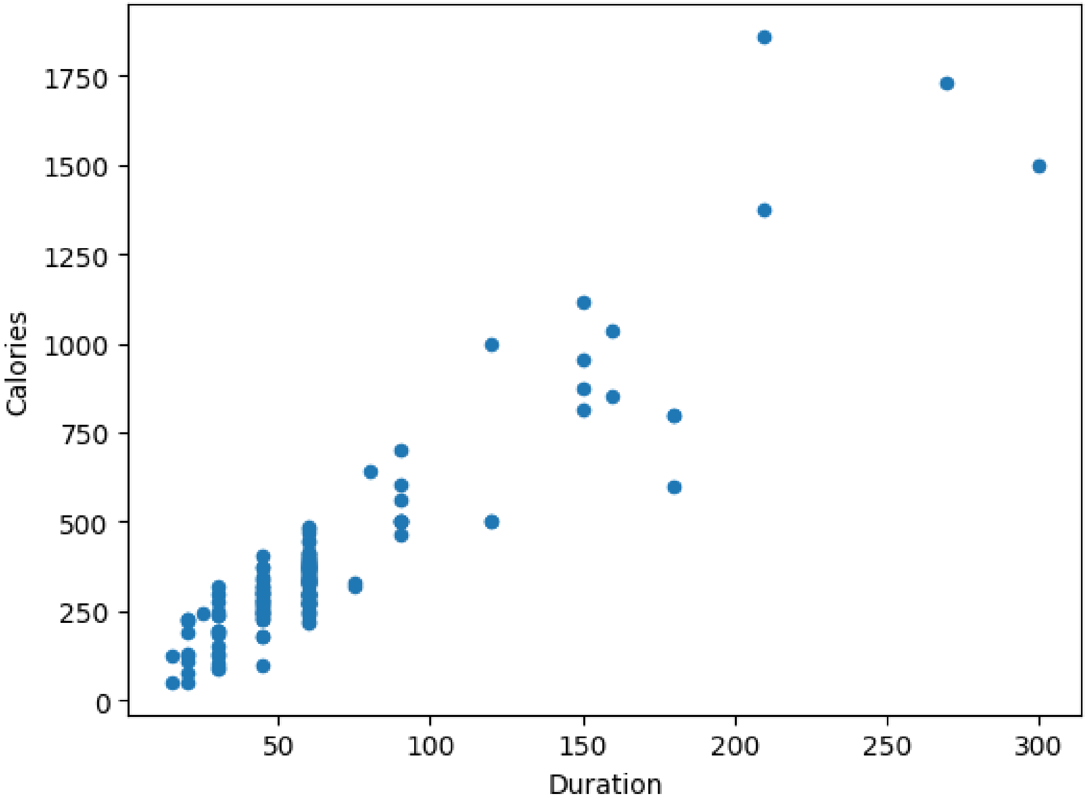
```
    Duration  Pulse  Maxpulse  Calories       Date
0         60    110       120     409.1 2020-12-01
1         60    117       120     479.0 2020-12-02
2         60    103       120     340.0 2020-12-03
3         45    109       120     282.4 2020-12-04
4         45    117       120     406.0 2020-12-05
5         60    102       120     300.0 2020-12-06
6     500000    110       120     374.0 2020-12-07
7         45    104       120     253.3 2020-12-08
9         30    109       120     195.1 2020-12-09
10        60     98       120     269.0 2020-12-10
11        60    103       120     329.3 2020-12-11
12        60    100       120     250.7 2020-12-12
13        60    106       120     345.3 2020-12-13
14        60    104       120     379.3 2020-12-14
15        60     98       120     275.0 2020-12-15
16        60     98       120     215.2 2020-12-16
17        60    100       120     300.0 2020-12-17
18        45     90       112       NaN 2020-12-18
19        60    103       120     323.0 2020-12-19
20        45     97       120     243.0 2020-12-20
21        60    108       120     364.2 2020-12-21
22        45    100       119     282.0 2020-12-22
23        60    130       101     300.0 2020-12-23
24        45    105       120     246.0 2020-12-24
25        60    102       120     334.5 2020-12-25
26        60    100       120     250.0 2020-12-26
27        60     92       118     241.0 2020-12-27
28        60    103       120       NaN 2020-12-28
29        60    100       120     280.0 2020-12-29
30        60    102       120     380.3 2020-12-30
```

[122]:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('example-data.csv')
```

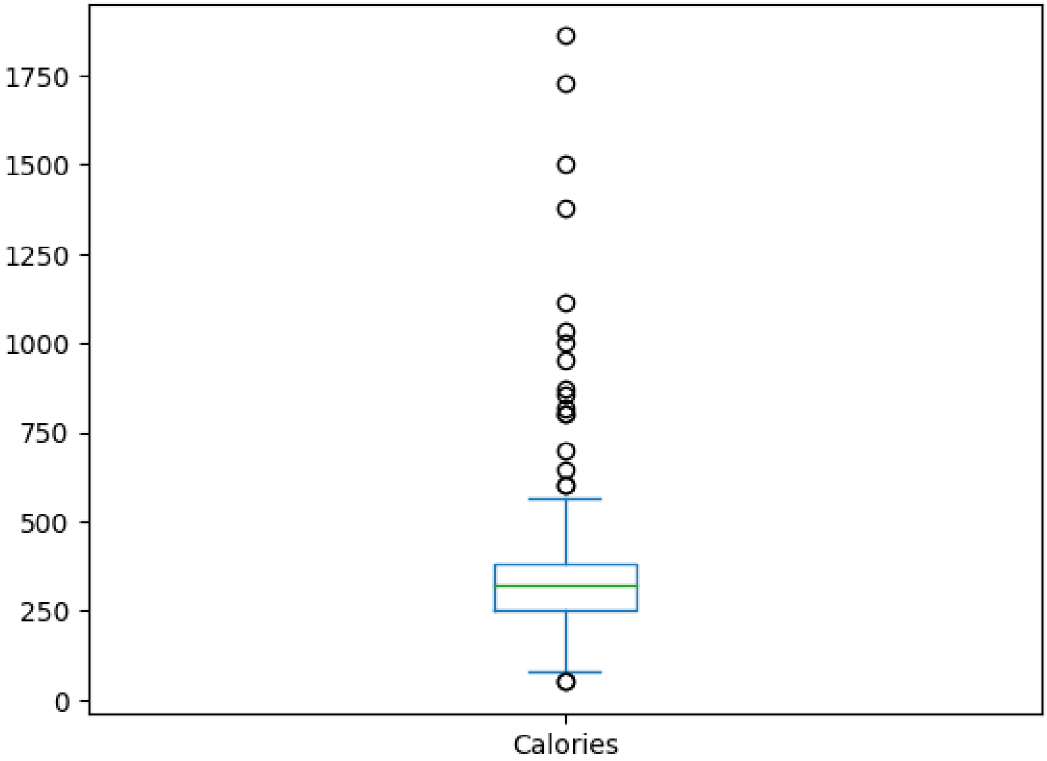
```
x = df["Calories"].mean()
df["Calories"].fillna(x, inplace=True)
df.plot()
df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')
plt.show()
plt.show()
```





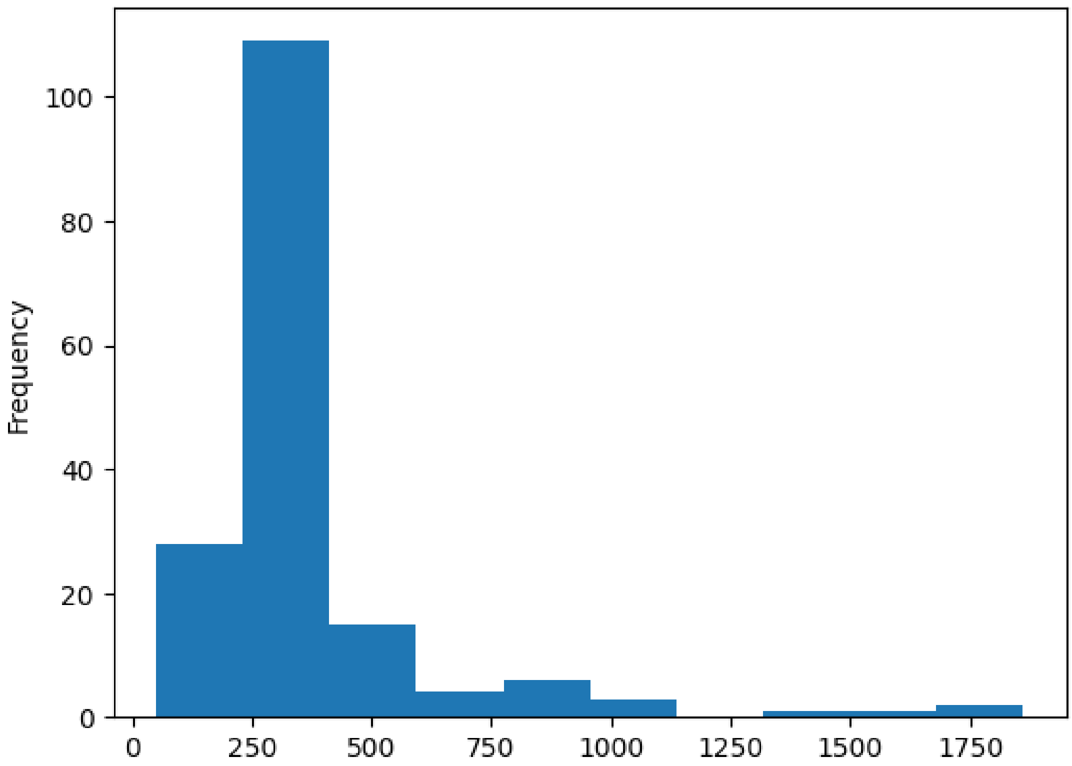
[123]:
```
# Histogram Plot
df["Calories"].plot(kind="box")

plt.plot()
```

[123]:
```
[]
```



```
[124]: df["Calories"].plot(kind="hist")
```

```
[124]: <Axes: ylabel='Frequency'>
```



positive skewed for `calories`