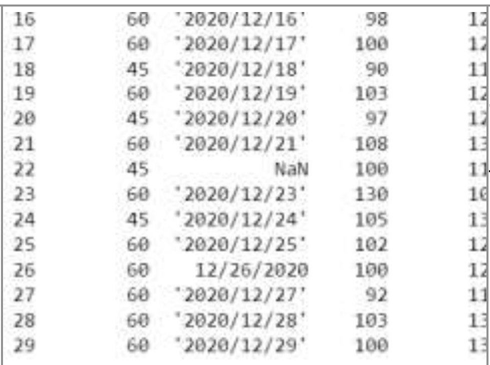
Cleaning data	<<CleaningData.csv>>

Empty Cells	Dropna() to remove empty cells Fillna() to fill something in empty cells

Dropna()



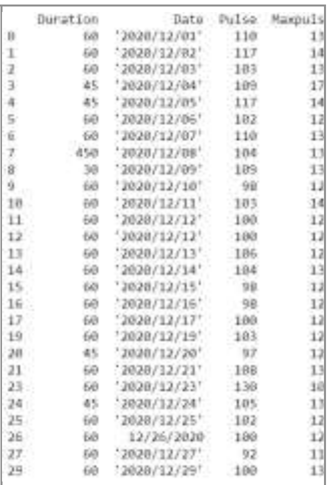
```
16      60  '2020/12/16'    98    12
17      60  '2020/12/17'   100    12
18      45  '2020/12/18'    90    11
19      60  '2020/12/19'   103    12
20      45  '2020/12/20'    97    12
21      60  '2020/12/21'   108    13
22      45           NaN   100    11
23      60  '2020/12/23'   130    1(
24      45  '2020/12/24'   105    13
25      60  '2020/12/25'   102    12
26      60    12/26/2020   100    12
27      60  '2020/12/27'    92    11
28      60  '2020/12/28'   103    13
29      60  '2020/12/29'   100    13
```

```
import pandas as pd

df = pd.read_csv('CleaningData.csv')

new_df = df.dropna()

print(new_df.to_string())
```



```
    Duration          Date  Pulse  Maxpuls
0         60  '2020/12/01'    110       13
1         60  '2020/12/02'    117       14
2         60  '2020/12/03'    103       13
3         45  '2020/12/04'    109       17
4         45  '2020/12/05'    117       14
5         60  '2020/12/06'    102       12
6         60  '2020/12/07'    110       13
7        450  '2020/12/08'    104       13
8         30  '2020/12/09'    109       13
9         60  '2020/12/10'     98       12
10        60  '2020/12/11'    103       14
11        60  '2020/12/12'    100       12
12        60  '2020/12/12'    100       12
13        60  '2020/12/13'    106       12
14        60  '2020/12/14'    104       13
15        60  '2020/12/15'     98       12
16        60  '2020/12/16'     98       12
17        60  '2020/12/17'    100       12
19        60  '2020/12/19'    103       12
20        45  '2020/12/20'     97       12
21        60  '2020/12/21'    108       13
23        60  '2020/12/23'    130       10
24        45  '2020/12/24'    105       13
25        60  '2020/12/25'    102       12
26        60    12/26/2020    100       12
27        60  '2020/12/27'     92       11
29        60  '2020/12/29'    100       13
```

Note= We can see its replacing new dataframe not the original one

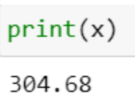
To replace null values in original dataframe use inplace=True	import pandas as pd
	df = pd.read_csv('data.csv')
	df.dropna(inplace = True)
	print(df.to_string())

Fillna()	import pandas as pd
	df = pd.read_csv('data.csv')
	df.fillna(130, inplace = True)
	454rf

Since we have less data its not good to remove entire row

For numerical values we will calculate mean value for coulmns

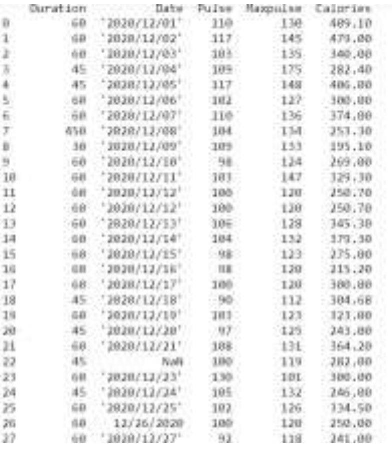
```
x =
df["Calories"].mean()
df["Calories"].fillna(x,
inplace = True)
```



```
print(x)
```

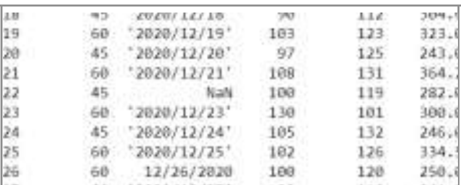
```
304.68
```

We got x value as 304.68 and it will filled with null values in df



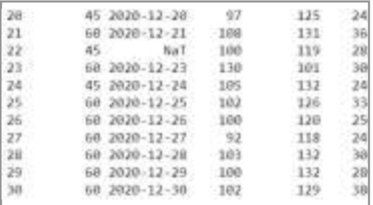
```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130    409.10
1         60  '2020/12/02'    117       145    479.00
2         60  '2020/12/03'    103       135    340.00
3         45  '2020/12/04'    109       175    282.40
4         45  '2020/12/05'    117       148    406.00
5         60  '2020/12/06'    102       127    300.00
6         60  '2020/12/07'    110       136    374.00
7        450  '2020/12/08'    104       134    253.30
8         30  '2020/12/09'    109       133    195.10
9         60  '2020/12/10'     98       124    269.00
10        60  '2020/12/11'    103       147    329.30
11        60  '2020/12/12'    100       120    250.70
12        60  '2020/12/12'    100       120    250.70
13        60  '2020/12/13'    106       128    345.30
14        60  '2020/12/14'    104       132    379.30
15        60  '2020/12/15'     98       123    275.00
16        60  '2020/12/16'     98       120    215.20
17        60  '2020/12/17'    100       120    300.00
18        45  '2020/12/18'     90       112    304.68
19        60  '2020/12/19'    103       123    323.00
20        45  '2020/12/20'     97       125    243.00
21        60  '2020/12/21'    108       131    364.20
22        45           NaN    100       119    282.00
23        60  '2020/12/23'    130       101    300.00
24        45  '2020/12/24'    105       132    246.00
25        60  '2020/12/25'    102       126    334.50
26        60    12/26/2020    100       120    250.00
27        60  '2020/12/27'     92       118    241.00
```

Our dataset has wrong date format



```
18        45  '2020/12/18'     90       112    304.
19        60  '2020/12/19'    103       123    323.
20        45  '2020/12/20'     97       125    243.
21        60  '2020/12/21'    108       131    364.
22        45           NaN    100       119    282.
23        60  '2020/12/23'    130       101    300.
24        45  '2020/12/24'    105       132    246.
25        60  '2020/12/25'    102       126    334.
26        60    12/26/2020    100       120    250.
```

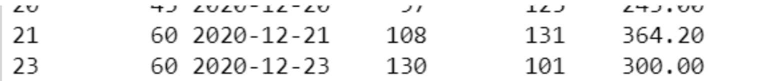
```
df['Date'] =
pd.to_datetime(df['Date'])
print(df.to_string())
```



```
20        45 2020-12-20     97       125   24
21        60 2020-12-21    108       131   36
22        45        NaT    100       119   28
23        60 2020-12-23    130       101   30
24        45 2020-12-24    105       132   24
25        60 2020-12-25    102       126   33
26        60 2020-12-26    100       120   25
27        60 2020-12-27     92       118   24
28        60 2020-12-28    103       132   38
29        60 2020-12-29    100       132   28
30        60 2020-12-30    102       129   38
```

Its better to drop null values in date

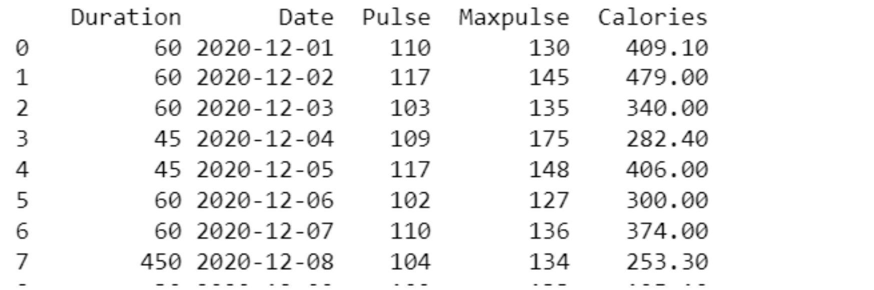
```
df.dropna(subset=['Date'],
inplace = True)
```



```
20        45 2020-12-20     97       125    243.00
21        60 2020-12-21    108       131    364.20
23        60 2020-12-23    130       101    300.00
```

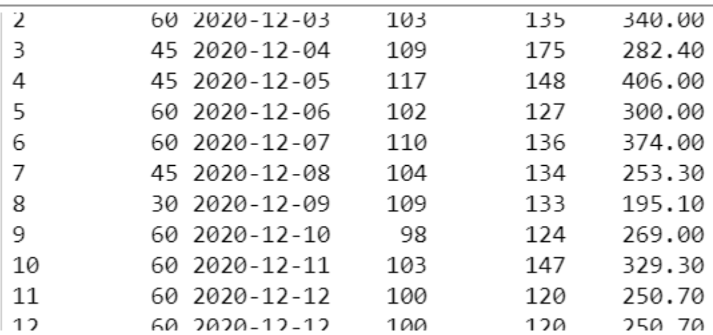
So 22 value is removed

Wrong data or unmatchable to respective column values



```
    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130    409.10
1         60 2020-12-02    117       145    479.00
2         60 2020-12-03    103       135    340.00
3         45 2020-12-04    109       175    282.40
4         45 2020-12-05    117       148    406.00
5         60 2020-12-06    102       127    300.00
6         60 2020-12-07    110       136    374.00
7        450 2020-12-08    104       134    253.30
```

Here one value contains 450 m which may not be appropriate so we will reduce it two 45 using loc print(df.to_string())

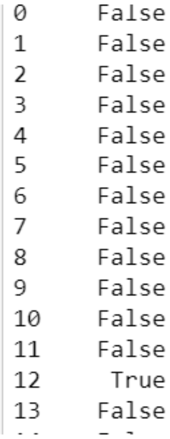


```
2           60  2020-12-03     103         135      340.00
3           45  2020-12-04     109         175      282.40
4           45  2020-12-05     117         148      406.00
5           60  2020-12-06     102         127      300.00
6           60  2020-12-07     110         136      374.00
7           45  2020-12-08     104         134      253.30
8           30  2020-12-09     109         133      195.10
9           60  2020-12-10      98         124      269.00
10          60  2020-12-11     103         147      329.30
11          60  2020-12-12     100         120      250.70
12          60  2020-12-12     100         120      250.70
```

Remove Duplicates

To check duplicate values

Df.duplicated()

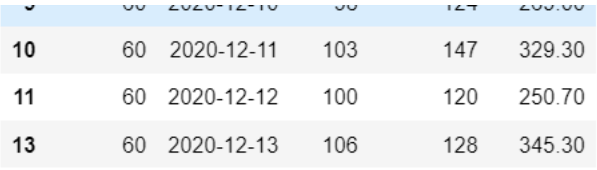


```
0      False
1      False
2      False
3      False
4      False
5      False
6      False
7      False
8      False
9      False
10     False
11     False
12      True
13     False
```

12 row has dublicate values

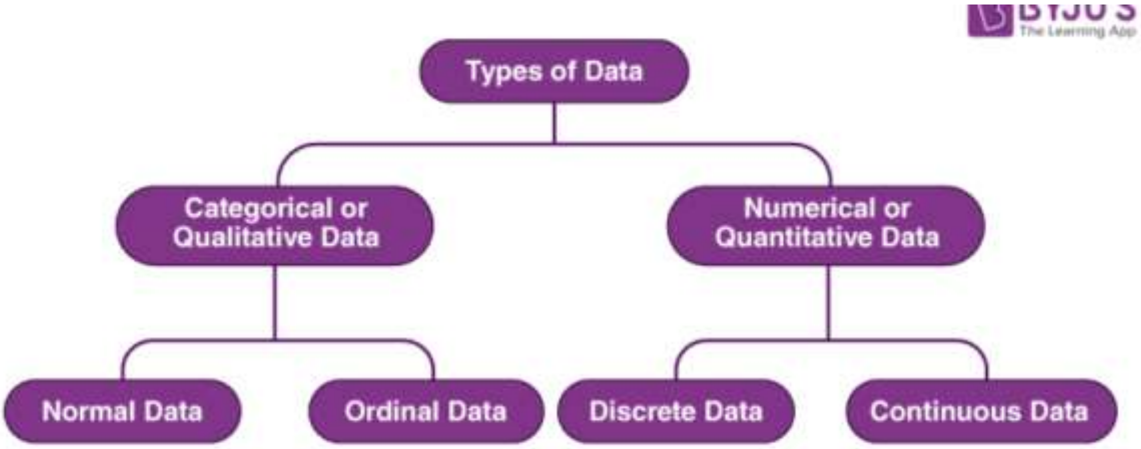
df.drop_duplicates()

To drop dublicates



10	60	2020-12-11	103	147	329.30
11	60	2020-12-12	100	120	250.70
13	60	2020-12-13	106	128	345.30

Data


BYJU'S
The Learning App
Types of Data
Categorical or Qualitative Data
Numerical or Quantitative Data
Normal Data
Ordinal Data
Discrete Data
Continuous Data