PANDAS TUTORIAL

June 29, 2022

[1]:
```
#PANDAS FOR DS..ANYTHING LEFT CHECK OUT THE NOTEBOOK
```

[2]:
```
#what is called as dataframe:
#its a cmbination of both columns and rows and it will be shown in representive␣
 ↪data.
```

[3]:
```
#creating dataframe
import pandas as pd
import numpy as np
demodataframe=pd.DataFrame(np.arange(0,20).
 ↪reshape(5,4),index=['row1','row2','row3','row4','row5'],columns=['cl1','cl2','cl3','cl4'])
print(demodataframe)
```

```
      cl1  cl2  cl3  cl4
row1    0    1    2    3
row2    4    5    6    7
row3    8    9   10   11
row4   12   13   14   15
row5   16   17   18   19
```

[4]:
```
import pandas as pd
import numpy as np

df=pd.read_csv('mercedesbenz.csv')
print(df.head)
# .head function prints the first 5 contents of the csv file
```

```
<bound method NDFrame.head of         ID       y  X0 X1  X2 X3 X4  X5 X6 X8  …
X375  X376  X377  X378  \
0        0  130.81   k  v  at  a  d   u  j  o  …     0     0     1     0
1        6   88.53   k  t  av  e  d   y  l  o  …     1     0     0     0
2        7   76.26  az  w   n  c  d   x  j  x  …     0     0     0     0
3        9   80.62  az  t   n  f  d   x  l  e  …     0     0     0     0
4       13   78.02  az  v   n  f  d   h  d  n  …     0     0     0     0
…     …     …  …  …  …  …  …  …  …  …  …  …  …  …  …  …  …  …  …  …  …
4204  8405  107.39  ak  s  as  c  d  aa  d  q  …     1     0     0     0
4205  8406  108.77   j  o   t  d  d  aa  h  h  …     0     1     0     0
4206  8412  109.22  ak  v   r  a  d  aa  g  e  …     0     0     1     0
4207  8415   87.48  al  r   e  f  d  aa  l  u  …     0     0     0     0
```

```
4208  8417  110.85   z  r  ae  c  d  aa  g  w  …     1     0     0     0

      X379  X380  X382  X383  X384  X385
0        0     0     0     0     0     0
1        0     0     0     0     0     0
2        0     0     1     0     0     0
3        0     0     0     0     0     0
4        0     0     0     0     0     0
…    …   …   …   …   …   …
4204     0     0     0     0     0     0
4205     0     0     0     0     0     0
4206     0     0     0     0     0     0
4207     0     0     0     0     0     0
4208     0     0     0     0     0     0

[4209 rows x 378 columns]>
```

[5]:
```
print(df.info())
#it gives the info about the above csv file..all the range and entries to int␣
 ↪and float datas
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4209 entries, 0 to 4208
Columns: 378 entries, ID to X385
dtypes: float64(1), int64(369), object(8)
memory usage: 12.1+ MB
None
```

[6]:
```
print(df.describe())
#this function is used to print only the int float values ad count mean but not␣
 ↪any string characters
```

```
                ID            y          X10     X11          X12  \
count  4209.000000  4209.000000  4209.000000  4209.0  4209.000000
mean   4205.960798   100.669318     0.013305     0.0     0.075077
std    2437.608688    12.679381     0.114590     0.0     0.263547
min       0.000000    72.110000     0.000000     0.0     0.000000
25%    2095.000000    90.820000     0.000000     0.0     0.000000
50%    4220.000000    99.150000     0.000000     0.0     0.000000
75%    6314.000000   109.010000     0.000000     0.0     0.000000
max    8417.000000   265.320000     1.000000     0.0     1.000000

               X13          X14          X15          X16          X17  …  \
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000  …
mean      0.057971     0.428130     0.000475     0.002613     0.007603  …
std       0.233716     0.494867     0.021796     0.051061     0.086872  …
min       0.000000     0.000000     0.000000     0.000000     0.000000  …
25%       0.000000     0.000000     0.000000     0.000000     0.000000  …
50%       0.000000     0.000000     0.000000     0.000000     0.000000  …
```

```
75%       0.000000     1.000000     0.000000     0.000000     0.000000  …
max       1.000000     1.000000     1.000000     1.000000     1.000000  …

              X375         X376         X377         X378         X379  \
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000
mean      0.318841     0.057258     0.314802     0.020670     0.009503
std       0.466082     0.232363     0.464492     0.142294     0.097033
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       1.000000     0.000000     1.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

              X380         X382         X383         X384         X385
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000
mean      0.008078     0.007603     0.001663     0.000475     0.001426
std       0.089524     0.086872     0.040752     0.021796     0.037734
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

[8 rows x 370 columns]
```

[7]:
```
print(df['X0'].value_counts())
#this shows that in the column X0 how many times the character in the column␣
 ↪has repeated
```

```
z     360
ak    349
y     324
ay    313
t     306
x     300
o     269
f     227
n     195
w     182
j     181
az    175
aj    151
s     106
ap    103
h      75
d      73
al     67
v      36
```

```
af     35
m      34
ai     34
e      32
ba     27
at     25
a      21
ax     19
aq     18
i      18
am     18
u      17
aw     16
l      16
ad     14
au     11
k      11
b      11
as     10
r      10
bc      6
ao      4
c       3
aa      2
q       2
ab      1
g       1
ac      1
Name: X0, dtype: int64
```

[8]:
```
print(df[df['y']>100])
#this shows that in the column Y..the values above 100 is printed
```

```
        ID       y  X0 X1  X2 X3 X4  X5 X6 X8  …  X375  X376  X377  X378  \
0        0  130.81   k  v  at  a  d   u  j  o  …     0     0     1     0
6       24  128.76  al  r   e  f  d   f  h  s  …     0     0     0     0
8       27  108.67   w  s  as  e  d   f  i  h  …     1     0     0     0
9       30  126.99   j  b  aq  c  d   f  a  e  …     0     0     1     0
10      31  102.09   h  r   r  f  d   f  h  p  …     0     0     0     0
…     …      …  …  .. ..  .. .. ..  .. .. ..  …   …  …   …   …   …
4202  8402  123.34  ap  l   s  c  d  aa  d  r  …     0     0     0     0
4204  8405  107.39  ak  s  as  c  d  aa  d  q  …     1     0     0     0
4205  8406  108.77   j  o   t  d  d  aa  h  h  …     0     1     0     0
4206  8412  109.22  ak  v   r  a  d  aa  g  e  …     0     0     1     0
4208  8417  110.85   z  r  ae  c  d  aa  g  w  …     1     0     0     0

      X379  X380  X382  X383  X384  X385
0        0     0     0     0     0     0
```

```
6          0     0     0     0     0     0
8          0     0     0     0     0     0
9          0     0     0     0     0     0
10         0     0     0     0     0     0
…     …    …     …     …     …     …
4202       0     0     0     0     0     0
4204       0     0     0     0     0     0
4205       0     0     0     0     0     0
4206       0     0     0     0     0     0
4208       0     0     0     0     0     0

[2004 rows x 378 columns]
```

[9]:
```
#stringIO module is an in memory file like object.this object can be used as an␣
 ↪input or output .
#in built functions in stringIO are:
#string='This is initial string.'
#StringIO.getvalue(): This function returns the entire content of the file.
#xyz = StringIO(string) : this basically creates a given string into a file␣
 ↪that means treated like a file
#print(xyz.read()) : this will read the file
#xyz.write(" Welcome to geeksforgeeks."):we can add or write in this file using␣
 ↪this function.
#xyz.seek(0): this will basically fix the cursor at 0..
#print('The string after writing is:', xyz.read()):this will read and print the␣
 ↪file from cursor 0 to the end.

#therefore we already have a csv file called as 'Test1.csv'..we are going to␣
 ↪implement the above information using the
#given csv file

from io import StringIO , BytesIO
give = ('c0lumn1,column2,column3,column4\n'
'row1,0,1,2,3\n'
'row2,4,5,6,7\n'
'row3,8,9,10,11\n'
'row4,12,13,14,15\n'
'row5,16,17,18,19\n'

)
```

[10]:
```
type(give)
#so this code basically says that the type of above data(give)is basically of␣
 ↪string type
#since its written in strings
```

[10]:
```
str
```

[11]:
```
pd.read_csv(StringIO(give))
# we are using thr abve code read_csv to read the stringIo file of the data called'give'
```

[11]:

	c0lumn1	column2	column3	column4
row1	0	1	2	3
row2	4	5	6	7
row3	8	9	10	11
row4	12	13	14	15
row5	16	17	18	19

[12]:
```
#to read specific colums from above csv file
df=pd.read_csv(StringIO(give), usecols=['column2','column4'])
print(df)
#now the output only prints the column2 and column4
```

	column2	column4
row1	1	3
row2	5	7
row3	9	11
row4	13	15
row5	17	19

[13]:
```
type(give)
```

[13]:
```
str
```

[14]:
```
#so basically the data give is in string format from the code line of 21
#to change the dtype from string to object or int you have to follow the code line from 23 to 27
df=pd.read_csv(StringIO(give),dtype=object)
```

[15]:
```
print(df)
```

	c0lumn1	column2	column3	column4
row1	0	1	2	3
row2	4	5	6	7
row3	8	9	10	11
row4	12	13	14	15
row5	16	17	18	19

[16]:
```
print(df['c0lumn1'])
```

```
row1     0
row2     4
row3     8
row4    12
```

```
row5    16
Name: c0lumn1, dtype: object
```

[17]:
```
#now we can also change the datatype of each columns in the data 'give' by␣
 ↪using the below code
df=pd.read_csv(StringIO(give),dtype={'column2':object,'column4':int})
print(df)
```

```
      c0lumn1  column2  column3  column4
row1        0        1        2        3
row2        4        5        6        7
row3        8        9       10       11
row4       12       13       14       15
row5       16       17       18       19
```

[18]:
```
print(df.dtypes)
#now you can see we changed the dtype of each columns using the above code ,␣
 ↪below output explains
```

```
c0lumn1     int64
column2    object
column3     int64
column4     int32
dtype: object
```

[19]:
```
#cleaning data
```

[20]:
```
#data cleaning means fixing bad data in our data set
#1 cleaning empty cells
#2 cleaning wrong format
#3 cleaning wrong data
#4 removing duplicates
```

[21]:
```
import pandas as pd
actual_read=pd.read_csv('dirtydata.csv')
print(actual_read)
#notice you have null values at row 22,18 and 28
#to remove that you have to clean
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
7        450  '2020/12/08'    104       134     253.3
8         30  '2020/12/09'    109       133     195.1
```

```
9          60  '2020/12/10'     98        124     269.0
10         60  '2020/12/11'    103        147     329.3
11         60  '2020/12/12'    100        120     250.7
12         60  '2020/12/12'    100        120     250.7
13         60  '2020/12/13'    106        128     345.3
14         60  '2020/12/14'    104        132     379.3
15         60  '2020/12/15'     98        123     275.0
16         60  '2020/12/16'     98        120     215.2
17         60  '2020/12/17'    100        120     300.0
18         45  '2020/12/18'     90        112       NaN
19         60  '2020/12/19'    103        123     323.0
20         45  '2020/12/20'     97        125     243.0
21         60  '2020/12/21'    108        131     364.2
22         45           NaN    100        119     282.0
23         60  '2020/12/23'    130        101     300.0
24         45  '2020/12/24'    105        132     246.0
25         60  '2020/12/25'    102        126     334.5
26         60      20201226    100        120     250.0
27         60  '2020/12/27'     92        118     241.0
28         60  '2020/12/28'    103        132       NaN
29         60  '2020/12/29'    100        132     280.0
30         60  '2020/12/30'    102        129     380.3
31         60  '2020/12/31'     92        115     243.0
```

[22]:
```
#cleaning empty cells
#first we will download a file with empy cells so we can practise how to cleam
import pandas as pd
dfemo= pd.read_csv('dirtydata.csv')

dfemo.dropna(inplace = True)

print(dfemo.to_string())
#the dropna(inplace = True) will NOT return a new DataFrame, but it will remove␣
 ↪all rows containg NULL values from the original DataFrame.
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
7        450  '2020/12/08'    104       134     253.3
8         30  '2020/12/09'    109       133     195.1
9         60  '2020/12/10'     98       124     269.0
10        60  '2020/12/11'    103       147     329.3
11        60  '2020/12/12'    100       120     250.7
```

12	60	'2020/12/12'	100	120	250.7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379.3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300.0
19	60	'2020/12/19'	103	123	323.0
20	45	'2020/12/20'	97	125	243.0
21	60	'2020/12/21'	108	131	364.2
23	60	'2020/12/23'	130	101	300.0
24	45	'2020/12/24'	105	132	246.0
25	60	'2020/12/25'	102	126	334.5
26	60	20201226	100	120	250.0
27	60	'2020/12/27'	92	118	241.0
29	60	'2020/12/29'	100	132	280.0
30	60	'2020/12/30'	102	129	380.3
31	60	'2020/12/31'	92	115	243.0

[23]:
```
#to replace new values into the null value areas you have to use command␣
 ↪fillna()
#The fillna() method allows us to replace empty cells with a value
import pandas as pd
demofemo = pd.read_csv('dirtydata.csv')

demofemo.fillna(130, inplace = True)
print(demofemo)
```

	Duration	Date	Pulse	Maxpulse	Calories
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	406.0
5	60	'2020/12/06'	102	127	300.0
6	60	'2020/12/07'	110	136	374.0
7	450	'2020/12/08'	104	134	253.3
8	30	'2020/12/09'	109	133	195.1
9	60	'2020/12/10'	98	124	269.0
10	60	'2020/12/11'	103	147	329.3
11	60	'2020/12/12'	100	120	250.7
12	60	'2020/12/12'	100	120	250.7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379.3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300.0
18	45	'2020/12/18'	90	112	130.0
19	60	'2020/12/19'	103	123	323.0

```
20        45  '2020/12/20'     97       125     243.0
21        60  '2020/12/21'    108       131     364.2
22        45           130    100       119     282.0
23        60  '2020/12/23'    130       101     300.0
24        45  '2020/12/24'    105       132     246.0
25        60  '2020/12/25'    102       126     334.5
26        60      20201226    100       120     250.0
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132     130.0
29        60  '2020/12/29'    100       132     280.0
30        60  '2020/12/30'    102       129     380.3
31        60  '2020/12/31'     92       115     243.0
```

[24]:
```
#to replace only for specified columns
#we are replacing value only in the specified column called as colories in the␣
 ↪above data.
import pandas as pd
df = pd.read_csv('dirtydata.csv')

df["Calories"].fillna(130, inplace = True)
print(df)
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
7        450  '2020/12/08'    104       134     253.3
8         30  '2020/12/09'    109       133     195.1
9         60  '2020/12/10'     98       124     269.0
10        60  '2020/12/11'    103       147     329.3
11        60  '2020/12/12'    100       120     250.7
12        60  '2020/12/12'    100       120     250.7
13        60  '2020/12/13'    106       128     345.3
14        60  '2020/12/14'    104       132     379.3
15        60  '2020/12/15'     98       123     275.0
16        60  '2020/12/16'     98       120     215.2
17        60  '2020/12/17'    100       120     300.0
18        45  '2020/12/18'     90       112     130.0
19        60  '2020/12/19'    103       123     323.0
20        45  '2020/12/20'     97       125     243.0
21        60  '2020/12/21'    108       131     364.2
22        45           NaN    100       119     282.0
23        60  '2020/12/23'    130       101     300.0
24        45  '2020/12/24'    105       132     246.0
```

```
25        60  '2020/12/25'    102       126     334.5
26        60      20201226    100       120     250.0
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132     130.0
29        60  '2020/12/29'    100       132     280.0
30        60  '2020/12/30'    102       129     380.3
31        60  '2020/12/31'     92       115     243.0
```

[25]:
```
#cleaning data of wrong format.
#Cells with data of wrong format can make it difficult, or even impossible, to␣
 ↪analyze data.
#To fix it, you have two options: remove the rows, or convert all cells in the␣
 ↪columns into the same format.
```

[26]:
```
#now you can see in the row 22 we have date in the wrong format
#to fix that you have to use the command to_datetime by following the below code
import pandas as pd
demofemo = pd.read_csv('dirtydata.csv')
demofemo['Date']=pd.to_datetime(demofemo['Date'])
print(demofemo)
#now if you look at the output the date at the row 22 is fixed
```

```
    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130     409.1
1         60 2020-12-02    117       145     479.0
2         60 2020-12-03    103       135     340.0
3         45 2020-12-04    109       175     282.4
4         45 2020-12-05    117       148     406.0
5         60 2020-12-06    102       127     300.0
6         60 2020-12-07    110       136     374.0
7        450 2020-12-08    104       134     253.3
8         30 2020-12-09    109       133     195.1
9         60 2020-12-10     98       124     269.0
10        60 2020-12-11    103       147     329.3
11        60 2020-12-12    100       120     250.7
12        60 2020-12-12    100       120     250.7
13        60 2020-12-13    106       128     345.3
14        60 2020-12-14    104       132     379.3
15        60 2020-12-15     98       123     275.0
16        60 2020-12-16     98       120     215.2
17        60 2020-12-17    100       120     300.0
18        45 2020-12-18     90       112       NaN
19        60 2020-12-19    103       123     323.0
20        45 2020-12-20     97       125     243.0
21        60 2020-12-21    108       131     364.2
22        45        NaT    100       119     282.0
23        60 2020-12-23    130       101     300.0
24        45 2020-12-24    105       132     246.0
```

```
25        60 2020-12-25    102       126     334.5
26        60 2020-12-26    100       120     250.0
27        60 2020-12-27     92       118     241.0
28        60 2020-12-28    103       132       NaN
29        60 2020-12-29    100       132     280.0
30        60 2020-12-30    102       129     380.3
31        60 2020-12-31     92       115     243.0
```

[27]:
```
#now wrong data doesnt mean wrong format or null values..we have may registered␣
 ↪the value wrong.
#now if you see the row 7 we have value 450 instead of 45..to do that we have␣
 ↪command called as:
##modifying and accessing rows and columns.
#.loc() means specified columns
#.iloc() you can specify the index and location
```

[28]:
```
import pandas as pd
demofemo = pd.read_csv('dirtydata.csv')
demofemo.loc[7, 'Duration'] = 45
print(demofemo)
```

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

```
26        60      20201226    100        120     250.0
27        60  '2020/12/27'     92        118     241.0
28        60  '2020/12/28'    103        132       NaN
29        60  '2020/12/29'    100        132     280.0
30        60  '2020/12/30'    102        129     380.3
31        60  '2020/12/31'     92        115     243.0
```

[29]:
```
#that is one way of doing it
#analyzing big data we can use loop
#in this below code we are looping throuh the duration ..if the value i greater␣
 ↪than 120 then we are setting it to 120
import pandas as pd

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

for x in df.index:
  if df.loc[x, "Duration"] > 120:
    df.loc[x, "Duration"] = 120

print(df.to_string())
```

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

```
26        60      20201226    100        120     250.0
27        60  '2020/12/27'     92        118     241.0
28        60  '2020/12/28'    103        132       NaN
29        60  '2020/12/29'    100        132     280.0
30        60  '2020/12/30'    102        129     380.3
31        60  '2020/12/31'     92        115     243.0
```

[30]:
```
#we can delete the row where the duration is greater than 120
import pandas as pd
df=pd.read_csv('dirtydata.csv')
for x in df.index:
  if df.loc[x, "Duration"] > 120:
    df.drop(x, inplace = True)

#remember to include the 'inplace = True' argument to make the changes in the␣
 ↪original DataFrame object instead of returning a copy

print(df.to_string())
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
8         30  '2020/12/09'    109       133     195.1
9         60  '2020/12/10'     98       124     269.0
10        60  '2020/12/11'    103       147     329.3
11        60  '2020/12/12'    100       120     250.7
12        60  '2020/12/12'    100       120     250.7
13        60  '2020/12/13'    106       128     345.3
14        60  '2020/12/14'    104       132     379.3
15        60  '2020/12/15'     98       123     275.0
16        60  '2020/12/16'     98       120     215.2
17        60  '2020/12/17'    100       120     300.0
18        45  '2020/12/18'     90       112       NaN
19        60  '2020/12/19'    103       123     323.0
20        45  '2020/12/20'     97       125     243.0
21        60  '2020/12/21'    108       131     364.2
22        45           NaN    100       119     282.0
23        60  '2020/12/23'    130       101     300.0
24        45  '2020/12/24'    105       132     246.0
25        60  '2020/12/25'    102       126     334.5
26        60      20201226    100       120     250.0
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132       NaN
```

```
29        60  '2020/12/29'    100        132     280.0
30        60  '2020/12/30'    102        129     380.3
31        60  '2020/12/31'     92        115     243.0
```

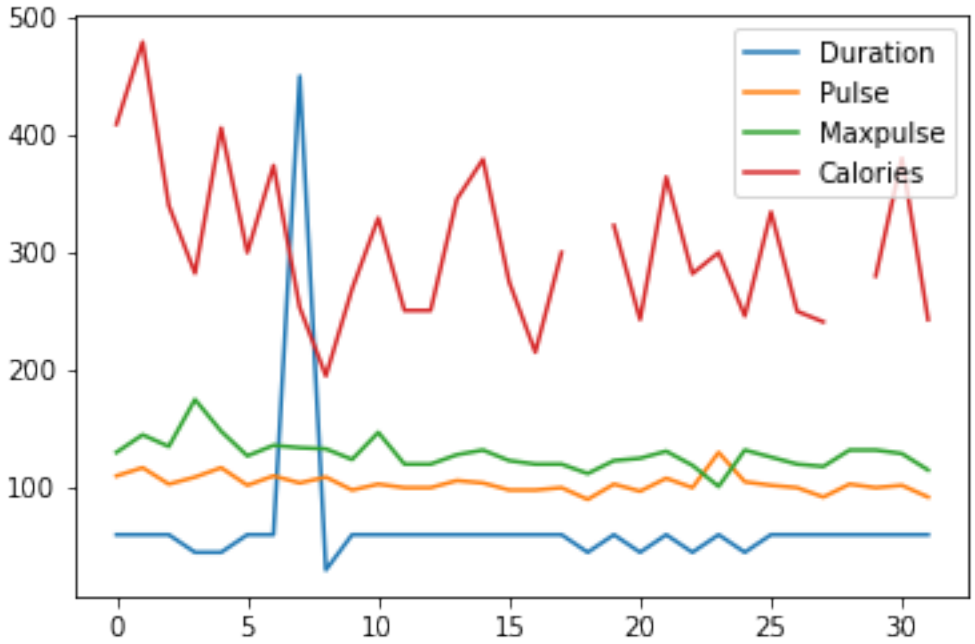
[31]:
```
#removing duplicates
#duplicate values are the values which have been registered more than one time
#to remove that we have to follow the below code
import pandas as pd

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

df.drop_duplicates(inplace = True)

print(df.to_string())

#Notice that row 12 has been removed from the result
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
7        450  '2020/12/08'    104       134     253.3
8         30  '2020/12/09'    109       133     195.1
9         60  '2020/12/10'     98       124     269.0
10        60  '2020/12/11'    103       147     329.3
11        60  '2020/12/12'    100       120     250.7
13        60  '2020/12/13'    106       128     345.3
14        60  '2020/12/14'    104       132     379.3
15        60  '2020/12/15'     98       123     275.0
16        60  '2020/12/16'     98       120     215.2
17        60  '2020/12/17'    100       120     300.0
18        45  '2020/12/18'     90       112       NaN
19        60  '2020/12/19'    103       123     323.0
20        45  '2020/12/20'     97       125     243.0
21        60  '2020/12/21'    108       131     364.2
22        45           NaN    100       119     282.0
23        60  '2020/12/23'    130       101     300.0
24        45  '2020/12/24'    105       132     246.0
25        60  '2020/12/25'    102       126     334.5
26        60      20201226    100       120     250.0
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132       NaN
29        60  '2020/12/29'    100       132     280.0
30        60  '2020/12/30'    102       129     380.3
```

```
31        60  '2020/12/31'     92       115     243.0
```

[32]:
```
#pandas has some inbuilt visualization
#this concept is interlinked with matplot lib
#first we have to read a csv file
reading = pd.read_csv('dirtydata.csv')
reading.plot()
```

[32]: <AxesSubplot:>



[33]:
```
show_data = pd.read_csv('dirtydata.csv')
print(show_data)
#this data shown below is plotted above in a visualization
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
7        450  '2020/12/08'    104       134     253.3
8         30  '2020/12/09'    109       133     195.1
9         60  '2020/12/10'     98       124     269.0
10        60  '2020/12/11'    103       147     329.3
```

```
11        60  '2020/12/12'    100       120     250.7
12        60  '2020/12/12'    100       120     250.7
13        60  '2020/12/13'    106       128     345.3
14        60  '2020/12/14'    104       132     379.3
15        60  '2020/12/15'     98       123     275.0
16        60  '2020/12/16'     98       120     215.2
17        60  '2020/12/17'    100       120     300.0
18        45  '2020/12/18'     90       112       NaN
19        60  '2020/12/19'    103       123     323.0
20        45  '2020/12/20'     97       125     243.0
21        60  '2020/12/21'    108       131     364.2
22        45           NaN    100       119     282.0
23        60  '2020/12/23'    130       101     300.0
24        45  '2020/12/24'    105       132     246.0
25        60  '2020/12/25'    102       126     334.5
26        60      20201226    100       120     250.0
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132       NaN
29        60  '2020/12/29'    100       132     280.0
30        60  '2020/12/30'    102       129     380.3
31        60  '2020/12/31'     92       115     243.0
```

[ ]: