pandas

August 9, 2024

[17]:
```
# Pandas == Panel Data
# Version Check
import pandas as pd

print(pd.__version__)
```

```
2.2.2
```

[18]:
```
# Sample Data Set
data = {"Car": ["BMW", "Mercedes", "Nissan"], "Pass": [3, 7, 2]}
myData = pd.DataFrame(data)
print(myData)
```

```
        Car  Pass
0       BMW     3
1  Mercedes     7
2    Nissan     2
```

[19]:
```
# Series --> Coloumn in a Table
myData = pd.Series(data["Pass"])
print(myData)
print(type(myData))
print(myData[0])
```

```
0    3
1    7
2    2
dtype: int64
<class 'pandas.core.series.Series'>
3
```

[20]:
```
# Series --> Coloumn in a Table
# Custom Indexing or Labeling
myData = pd.Series(data["Pass"], index=["x", "y", "z"])
print(myData)
print(myData["y"])
```

```
x    3
```

```
y    7
z    2
dtype: int64
7
```

[21]:
```
# Series For Dictionary Type Data
myData = pd.Series(data)
print(myData)
```

```
Car     [BMW, Mercedes, Nissan]
Pass                  [3, 7, 2]
dtype: object
```

[22]:
```
# Data Frame
data = {"Id": [4, 7, 2, 8], "Duration": [100, 45, 98, 55]}
myData = pd.DataFrame(data)
print(myData)
```

```
   Id  Duration
0   4       100
1   7        45
2   2        98
3   8        55
```

[25]:
```
# Loacte Row
data = {"Id": [4, 7, 2, 8], "Duration": [100, 45, 98, 55]}
myData = pd.DataFrame(data)

# Index
print(myData.loc[0])

# List of Index
print(myData.loc[[0, 2]])
```

```
Id           4
Duration    100
Name: 0, dtype: int64
   Id  Duration
0   4       100
2   2        98
```

[29]:
```
# Data Frame Cutom Indexing
data = {"Id": [4, 7, 2, 8], "Duration": [100, 45, 98, 55]}
customIndex = ["P1", "P2", "P3", "P4"]
myData = pd.DataFrame(data, index=customIndex)
print(myData)
print(myData.loc["P3"])
```

```
    Id  Duration
P1   4       100
P2   7        45
P3   2        98
P4   8        55
Id           2
Duration    98
Name: P3, dtype: int64
```

[34]:
```
# Load Files Into DataFrame
df = pd.read_csv("data.csv")
print(df)
print(df.loc[4])
print(df.loc[[1, 2]])
print(pd.Series(df.loc[4]))
```

```
     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     300.4
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
Duration     45.0
Pulse       117.0
Maxpulse    148.0
Calories    406.0
Name: 4, dtype: float64
   Duration  Pulse  Maxpulse  Calories
1        60    117       145     479.0
2        60    103       135     340.0
Duration     45.0
Pulse       117.0
Maxpulse    148.0
Calories    406.0
Name: 4, dtype: float64
```

[35]:
```
# Print The Entire DataFrame
df = pd.read_csv("data.csv")
print(df.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	300.5
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
30	60	92	115	243.0
31	45	90	112	180.1
32	60	101	124	299.0
33	60	93	113	223.0
34	60	107	136	361.0
35	60	114	140	415.0
36	60	102	127	300.5
37	60	100	120	300.1
38	60	100	120	300.0
39	45	104	129	266.0
40	45	90	112	180.1
41	60	98	126	286.0
42	60	100	122	329.4
43	60	111	138	400.0
44	60	111	131	397.0
45	60	99	119	273.0
46	60	109	153	387.6

47	45	111	136	300.0
48	45	108	129	298.0
49	60	111	139	397.6
50	60	107	136	380.2
51	80	123	146	643.1
52	60	106	130	263.0
53	60	118	151	486.0
54	30	136	175	238.0
55	60	121	146	450.7
56	60	118	121	413.0
57	45	115	144	305.0
58	20	153	172	226.4
59	45	123	152	321.0
60	210	108	160	1376.0
61	160	110	137	1034.4
62	160	109	135	853.0
63	45	118	141	341.0
64	20	110	130	131.4
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
68	20	106	136	110.4
69	300	108	143	1500.2
70	150	97	129	1115.0
71	60	109	153	387.6
72	90	100	127	700.0
73	150	97	127	953.2
74	45	114	146	304.0
75	90	98	125	563.2
76	45	105	134	251.0
77	45	110	141	300.0
78	120	100	130	500.4
79	270	100	131	1729.0
80	30	159	182	319.2
81	45	149	169	344.0
82	30	103	139	151.1
83	120	100	130	500.0
84	45	100	120	225.3
85	30	151	170	300.1
86	45	102	136	234.0
87	120	100	157	1000.1
88	45	129	103	242.0
89	20	83	107	50.3
90	180	101	127	600.1
91	45	107	137	NaN
92	30	90	107	105.3
93	15	80	100	50.5
94	20	150	171	127.4

95	20	151	168	229.4
96	30	95	128	128.2
97	25	152	168	244.2
98	30	109	131	188.2
99	90	93	124	604.1
100	20	95	112	77.7
101	90	90	110	500.0
102	90	90	100	500.0
103	90	90	100	500.4
104	30	92	108	92.7
105	30	93	128	124.0
106	180	90	120	800.3
107	30	90	120	86.2
108	90	90	120	500.3
109	210	137	184	1860.4
110	60	102	124	325.2
111	45	107	124	275.0
112	15	124	139	124.2
113	45	100	120	225.3
114	60	108	131	367.6
115	60	108	151	351.7
116	60	116	141	443.0
117	60	97	122	277.4
118	60	105	125	NaN
119	60	103	124	332.7
120	30	112	137	193.9
121	45	100	120	100.7
122	60	119	169	336.7
123	60	107	127	344.9
124	60	111	151	368.5
125	60	98	122	271.0
126	60	97	124	275.3
127	60	109	127	382.0
128	90	99	125	466.4
129	60	114	151	384.0
130	60	104	134	342.5
131	60	107	138	357.5
132	60	103	133	335.0
133	60	106	132	327.5
134	60	103	136	339.0
135	20	136	156	189.0
136	45	117	143	317.7
137	45	115	137	318.0
138	45	113	138	308.0
139	20	141	162	222.4
140	60	108	135	390.0
141	60	97	127	NaN
142	45	100	120	250.4

```
143        45    122        149     335.4
144        60    136        170     470.2
145        45    106        126     270.8
146        60    107        136     400.0
147        60    112        146     361.9
148        30    103        127     185.0
149        60    110        150     409.4
150        60    106        134     343.0
151        60    109        129     353.2
152        60    109        138     374.0
153        30    150        167     275.8
154        60    105        128     328.0
155        60    111        151     368.5
156        60     97        131     270.4
157        60    100        120     270.4
158        60    114        150     382.8
159        30     80        120     240.9
160        30     85        120     250.4
161        45     90        130     260.4
162        45     95        130     270.0
163        45    100        140     280.9
164        60    105        140     290.8
165        60    110        145     300.4
166        60    115        145     310.2
167        75    120        150     320.4
168        75    125        150     330.4
```

[46]:
```
# Systems Maximum Rows
df = pd.read_csv("data.csv")
print(pd.options.display.max_rows)
print(pd.options.display.max_columns)
print(df)
```

```
60
20
     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     300.4
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
```

[169 rows x 4 columns]

[48]:
```
# Increase The Max Row
df = pd.read_csv("data.csv")
pd.options.display.max_rows = 9999
print(df)
```

```
     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     300.5
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
30         60     92       115     243.0
31         45     90       112     180.1
32         60    101       124     299.0
33         60     93       113     223.0
34         60    107       136     361.0
35         60    114       140     415.0
36         60    102       127     300.5
37         60    100       120     300.1
38         60    100       120     300.0
39         45    104       129     266.0
```

40	45	90	112	180.1
41	60	98	126	286.0
42	60	100	122	329.4
43	60	111	138	400.0
44	60	111	131	397.0
45	60	99	119	273.0
46	60	109	153	387.6
47	45	111	136	300.0
48	45	108	129	298.0
49	60	111	139	397.6
50	60	107	136	380.2
51	80	123	146	643.1
52	60	106	130	263.0
53	60	118	151	486.0
54	30	136	175	238.0
55	60	121	146	450.7
56	60	118	121	413.0
57	45	115	144	305.0
58	20	153	172	226.4
59	45	123	152	321.0
60	210	108	160	1376.0
61	160	110	137	1034.4
62	160	109	135	853.0
63	45	118	141	341.0
64	20	110	130	131.4
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
68	20	106	136	110.4
69	300	108	143	1500.2
70	150	97	129	1115.0
71	60	109	153	387.6
72	90	100	127	700.0
73	150	97	127	953.2
74	45	114	146	304.0
75	90	98	125	563.2
76	45	105	134	251.0
77	45	110	141	300.0
78	120	100	130	500.4
79	270	100	131	1729.0
80	30	159	182	319.2
81	45	149	169	344.0
82	30	103	139	151.1
83	120	100	130	500.0
84	45	100	120	225.3
85	30	151	170	300.1
86	45	102	136	234.0
87	120	100	157	1000.1

88	45	129	103	242.0
89	20	83	107	50.3
90	180	101	127	600.1
91	45	107	137	NaN
92	30	90	107	105.3
93	15	80	100	50.5
94	20	150	171	127.4
95	20	151	168	229.4
96	30	95	128	128.2
97	25	152	168	244.2
98	30	109	131	188.2
99	90	93	124	604.1
100	20	95	112	77.7
101	90	90	110	500.0
102	90	90	100	500.0
103	90	90	100	500.4
104	30	92	108	92.7
105	30	93	128	124.0
106	180	90	120	800.3
107	30	90	120	86.2
108	90	90	120	500.3
109	210	137	184	1860.4
110	60	102	124	325.2
111	45	107	124	275.0
112	15	124	139	124.2
113	45	100	120	225.3
114	60	108	131	367.6
115	60	108	151	351.7
116	60	116	141	443.0
117	60	97	122	277.4
118	60	105	125	NaN
119	60	103	124	332.7
120	30	112	137	193.9
121	45	100	120	100.7
122	60	119	169	336.7
123	60	107	127	344.9
124	60	111	151	368.5
125	60	98	122	271.0
126	60	97	124	275.3
127	60	109	127	382.0
128	90	99	125	466.4
129	60	114	151	384.0
130	60	104	134	342.5
131	60	107	138	357.5
132	60	103	133	335.0
133	60	106	132	327.5
134	60	103	136	339.0
135	20	136	156	189.0

```
136        45    117        143     317.7
137        45    115        137     318.0
138        45    113        138     308.0
139        20    141        162     222.4
140        60    108        135     390.0
141        60     97        127       NaN
142        45    100        120     250.4
143        45    122        149     335.4
144        60    136        170     470.2
145        45    106        126     270.8
146        60    107        136     400.0
147        60    112        146     361.9
148        30    103        127     185.0
149        60    110        150     409.4
150        60    106        134     343.0
151        60    109        129     353.2
152        60    109        138     374.0
153        30    150        167     275.8
154        60    105        128     328.0
155        60    111        151     368.5
156        60     97        131     270.4
157        60    100        120     270.4
158        60    114        150     382.8
159        30     80        120     240.9
160        30     85        120     250.4
161        45     90        130     260.4
162        45     95        130     270.0
163        45    100        140     280.9
164        60    105        140     290.8
165        60    110        145     300.4
166        60    115        145     310.2
167        75    120        150     320.4
168        75    125        150     330.4
```

[52]:
```
# Read JSON
df = pd.read_json("data.json")
print(df)
```

```
     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     300.5
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
30	60	92	115	243.0
31	45	90	112	180.1
32	60	101	124	299.0
33	60	93	113	223.0
34	60	107	136	361.0
35	60	114	140	415.0
36	60	102	127	300.5
37	60	100	120	300.1
38	60	100	120	300.0
39	45	104	129	266.0
40	45	90	112	180.1
41	60	98	126	286.0
42	60	100	122	329.4
43	60	111	138	400.0
44	60	111	131	397.0
45	60	99	119	273.0
46	60	109	153	387.6
47	45	111	136	300.0
48	45	108	129	298.0
49	60	111	139	397.6
50	60	107	136	380.2
51	80	123	146	643.1
52	60	106	130	263.0
53	60	118	151	486.0
54	30	136	175	238.0
55	60	121	146	450.7
56	60	118	121	413.0

57	45	115	144	305.0
58	20	153	172	226.4
59	45	123	152	321.0
60	210	108	160	1376.0
61	160	110	137	1034.4
62	160	109	135	853.0
63	45	118	141	341.0
64	20	110	130	131.4
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
68	20	106	136	110.4
69	300	108	143	1500.2
70	150	97	129	1115.0
71	60	109	153	387.6
72	90	100	127	700.0
73	150	97	127	953.2
74	45	114	146	304.0
75	90	98	125	563.2
76	45	105	134	251.0
77	45	110	141	300.0
78	120	100	130	500.4
79	270	100	131	1729.0
80	30	159	182	319.2
81	45	149	169	344.0
82	30	103	139	151.1
83	120	100	130	500.0
84	45	100	120	225.3
85	30	151	170	300.1
86	45	102	136	234.0
87	120	100	157	1000.1
88	45	129	103	242.0
89	20	83	107	50.3
90	180	101	127	600.1
91	45	107	137	NaN
92	30	90	107	105.3
93	15	80	100	50.5
94	20	150	171	127.4
95	20	151	168	229.4
96	30	95	128	128.2
97	25	152	168	244.2
98	30	109	131	188.2
99	90	93	124	604.1
100	20	95	112	77.7
101	90	90	110	500.0
102	90	90	100	500.0
103	90	90	100	500.4
104	30	92	108	92.7

105	30	93	128	124.0
106	180	90	120	800.3
107	30	90	120	86.2
108	90	90	120	500.3
109	210	137	184	1860.4
110	60	102	124	325.2
111	45	107	124	275.0
112	15	124	139	124.2
113	45	100	120	225.3
114	60	108	131	367.6
115	60	108	151	351.7
116	60	116	141	443.0
117	60	97	122	277.4
118	60	105	125	NaN
119	60	103	124	332.7
120	30	112	137	193.9
121	45	100	120	100.7
122	60	119	169	336.7
123	60	107	127	344.9
124	60	111	151	368.5
125	60	98	122	271.0
126	60	97	124	275.3
127	60	109	127	382.0
128	90	99	125	466.4
129	60	114	151	384.0
130	60	104	134	342.5
131	60	107	138	357.5
132	60	103	133	335.0
133	60	106	132	327.5
134	60	103	136	339.0
135	20	136	156	189.0
136	45	117	143	317.7
137	45	115	137	318.0
138	45	113	138	308.0
139	20	141	162	222.4
140	60	108	135	390.0
141	60	97	127	NaN
142	45	100	120	250.4
143	45	122	149	335.4
144	60	136	170	470.2
145	45	106	126	270.8
146	60	107	136	400.0
147	60	112	146	361.9
148	30	103	127	185.0
149	60	110	150	409.4
150	60	106	134	343.0
151	60	109	129	353.2
152	60	109	138	374.0

```
153        30    150        167     275.8
154        60    105        128     328.0
155        60    111        151     368.5
156        60     97        131     270.4
157        60    100        120     270.4
158        60    114        150     382.8
159        30     80        120     240.9
160        30     85        120     250.4
161        45     90        130     260.4
162        45     95        130     270.0
163        45    100        140     280.9
164        60    105        140     290.8
165        60    110        145     300.4
166        60    115        145     310.2
167        75    120        150     320.4
168        75    125        150     330.4
```

[63]:
```
# Analyzing Data Frames
df = pd.read_json("data.json")
print(df.head())
print(df.head(10))
print(df.tail())
print(df.tail(10))
```

```
   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
   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     300.5
6        60    110       136     374.0
7        45    104       134     253.3
8        30    109       133     195.1
9        60     98       124     269.0
     Duration  Pulse  Maxpulse  Calories
164        60    105       140     290.8
165        60    110       145     300.4
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
     Duration  Pulse  Maxpulse  Calories
```

```
159        30     80        120     240.9
160        30     85        120     250.4
161        45     90        130     260.4
162        45     95        130     270.0
163        45    100        140     280.9
164        60    105        140     290.8
165        60    110        145     300.4
166        60    115        145     310.2
167        75    120        150     320.4
168        75    125        150     330.4
```

[66]:
```
# Data Frame Info
df = pd.read_csv("data.csv")
print(df.info())
```

```
<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
```

[69]:
```
# Cleaning Empty Cells
df = pd.read_csv("new_data.csv")

# This Will Remove The Empty Cell Containing Rows
new_df = df.dropna()
# Here new_df is a copy of df
print(new_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
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    2020/12/26    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
```

[71]:
```
# Cleaning Empty Cells
df = pd.read_csv("new_data.csv")

# This Will Remove The Empty Cell Containing Rows
df.dropna(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
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    2020/12/26    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
```

[72]:
```
# Replace Empty Values
df = pd.read_csv("new_data.csv")

# This Will Replace The Empty Values With 130
df.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           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    2020/12/26    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
```

[74]:
```
# Replace Empty Values
df = pd.read_csv("new_data.csv")

# This Will Replace The Empty Values With 130
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    2020/12/26    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

C:\Users\Md Abdullah Emon\AppData\Local\Temp\ipykernel_12740\877189009.py:5:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
```

```
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


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

[76]:
```
# Replace Empty Values
df = pd.read_csv("new_data.csv")

# This Will Replace The Empty Values With 130
x = df["Calories"].mean()
print(x)
df["Calories"].fillna(x, inplace=True)
print(df)
```

```
304.68
    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    2020/12/26    100       120    250.00
27        60  '2020/12/27'     92       118    241.00
```

```
28        60  '2020/12/28'    103       132    304.68
29        60  '2020/12/29'    100       132    280.00
30        60  '2020/12/30'    102       129    380.30
31        60  '2020/12/31'     92       115    243.00

C:\Users\Md Abdullah Emon\AppData\Local\Temp\ipykernel_12740\2859649932.py:7:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


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

[90]:
```
# Cleaning Data of Wrong Format

df = pd.read_csv("new_data.csv")
# df["Date"] = pd.to_datetime(df["Date"])
# print(df)
```

[96]:
```
# Replace Value
df = pd.read_csv("new_data.csv")

df.loc[0, "Duration"] = 1
print(df.loc[0])

# Info of Index
print(df.index)

# Setting The Value of Duration >45 by 100
for x in df.index:
    if df.loc[x, "Duration"] > 45:
        df.loc[x, "Duration"] = 100
print(df)
```

```
Duration              1
Date        '2020/12/01'
Pulse                110
Maxpulse             130
Calories           409.1
Name: 0, dtype: object
RangeIndex(start=0, stop=32, step=1)
    Duration          Date  Pulse  Maxpulse  Calories
```

	Duration	Date	Pulse	Maxpulse	Calories
0	1	'2020/12/01'	110	130	409.1
1	100	'2020/12/02'	117	145	479.0
2	100	'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	100	'2020/12/06'	102	127	300.0
6	100	'2020/12/07'	110	136	374.0
7	100	'2020/12/08'	104	134	253.3
8	30	'2020/12/09'	109	133	195.1
9	100	'2020/12/10'	98	124	269.0
10	100	'2020/12/11'	103	147	329.3
11	100	'2020/12/12'	100	120	250.7
12	100	'2020/12/12'	100	120	250.7
13	100	'2020/12/13'	106	128	345.3
14	100	'2020/12/14'	104	132	379.3
15	100	'2020/12/15'	98	123	275.0
16	100	'2020/12/16'	98	120	215.2
17	100	'2020/12/17'	100	120	300.0
18	45	'2020/12/18'	90	112	NaN
19	100	'2020/12/19'	103	123	323.0
20	45	'2020/12/20'	97	125	243.0
21	100	'2020/12/21'	108	131	364.2
22	45	NaN	100	119	282.0
23	100	'2020/12/23'	130	101	300.0
24	45	'2020/12/24'	105	132	246.0
25	100	'2020/12/25'	102	126	334.5
26	100	2020/12/26	100	120	250.0
27	100	'2020/12/27'	92	118	241.0
28	100	'2020/12/28'	103	132	NaN
29	100	'2020/12/29'	100	132	280.0
30	100	'2020/12/30'	102	129	380.3
31	100	'2020/12/31'	92	115	243.0

[103]:
```
# Removing The Rows Which Has A Greater Calorie Than 300
df = pd.read_csv("new_data.csv")

for i in df.index:
    if df.loc[i, "Calories"] > 300:
        df.drop(i, inplace=True)
print(df)
```

	Duration	Date	Pulse	Maxpulse	Calories
3	45	'2020/12/04'	109	175	282.4
5	60	'2020/12/06'	102	127	300.0
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
11	60	'2020/12/12'	100	120	250.7

```
12        60  '2020/12/12'    100       120     250.7
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
20        45  '2020/12/20'     97       125     243.0
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
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
31        60  '2020/12/31'     92       115     243.0
```

[107]:
```
# Removing Duplicates
df = pd.read_csv("new_data.csv")
print(df.duplicated())

df.drop_duplicates(inplace=True)
print(df)
```

```
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
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
```

```
26    False
27    False
28    False
29    False
30    False
31    False
dtype: bool
```

	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	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