In [52]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [69]:
```
df = pd.read_csv('dirtydata.csv')
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       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
```

In [54]:
```
df.shape
```

Out[54]: (32, 5)

In [55]:
```
df.describe()
```


Out[55]:

	Duration	Pulse	Maxpulse	Calories
count	32.000000	32.000000	32.000000	30.000000
mean	68.437500	103.500000	128.500000	266.013333
std	70.039591	7.832933	12.998759	164.876415
min	30.000000	90.000000	101.000000	-300.000000
25%	60.000000	100.000000	120.000000	247.000000
50%	60.000000	102.500000	127.500000	282.200000
75%	60.000000	106.500000	132.250000	343.975000
max	450.000000	130.000000	175.000000	479.000000

In [56]:
```
df.isnull().sum()
```

Out[56]:
```
Duration    0
Date        1
Pulse       0
Maxpulse    0
Calories    2
dtype: int64
```

In [57]:
```
df.dtypes
```

Out[57]:
```
Duration      int64
Date         object
Pulse         int64
Maxpulse      int64
Calories    float64
dtype: object
```

In [70]:
```
df['Calories'] = df['Calories'].abs()
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	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

In [59]:
```
x = df['Calories'].mean()
x
```

Out[59]: 304.68

In [71]:
```
# Filling null values
df['Calories'].fillna(x, inplace=True)
print(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	20201226	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

In [72]:
```
# Convert the date into a "Date" format
df['Date'] = pd.to_datetime(df['Date'], format='mixed')
print(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	NaT	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

In [73]:
```
# 7th location's duration value is 450 which is an outlier
df.loc[7, 'Duration'] = 45
print(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	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
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	NaT	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

In [63]:
```
# Finding duplicates
df.duplicated().sum()
```

Out[63]: 1

In [74]:
```
df.drop_duplicates(inplace=True)
print(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	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
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	NaT	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

In [76]:
```
data = pd.read_csv('nba.csv')
data
```


Out[76]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2-Jun	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	6-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	5-Jun	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	5-Jun	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	10-Jun	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	PF	20	10-Jun	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	3-Jun	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	1-Jun	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	3-Jul	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	Jul-00	231	Kansas	947276.0

457 rows × 9 columns

In [77]:
```
data['Position'].value_counts()
```

Out[77]:
```
Position
SG    102
PF    100
PG     92
SF     85
C      78
Name: count, dtype: int64
```

In [78]:
```
data['Position'].replace(['SG','PF','PG','SF','C'], [1,2,3,4,5], inplace=True)
data
```

Out[78]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	3	25	2-Jun	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	4	25	6-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	1	27	5-Jun	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	1	22	5-Jun	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	2	29	10-Jun	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	2	20	10-Jun	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	3	26	3-Jun	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	3	24	1-Jun	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	5	26	3-Jul	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	5	26	Jul-00	231	Kansas	947276.0

457 rows × 9 columns

In [79]:
```
category = pd.cut(data.Age, bins=[19,25,30,35,45], labels=['A','B','C','D'])
data.insert(5, 'Age_group', category)
data
```


Out[79]:

	Name	Team	Number	Position	Age	Age_group	Height	Weight	College	
0	Avery Bradley	Boston Celtics	0	3	25	A	2-Jun	180	Texas	77
1	Jae Crowder	Boston Celtics	99	4	25	A	6-Jun	235	Marquette	67
2	John Holland	Boston Celtics	30	1	27	B	5-Jun	205	Boston University	
3	R.J. Hunter	Boston Celtics	28	1	22	A	5-Jun	185	Georgia State	11
4	Jonas Jerebko	Boston Celtics	8	2	29	B	10-Jun	231	NaN	50
...	...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	2	20	A	10-Jun	234	Kentucky	22
453	Shelvin Mack	Utah Jazz	8	3	26	B	3-Jun	203	Butler	24
454	Raul Neto	Utah Jazz	25	3	24	A	1-Jun	179	NaN	9
455	Tibor Pleiss	Utah Jazz	21	5	26	B	3-Jul	256	NaN	29
456	Jeff Withey	Utah Jazz	24	5	26	B	Jul-00	231	Kansas	9

457 rows × 10 columns