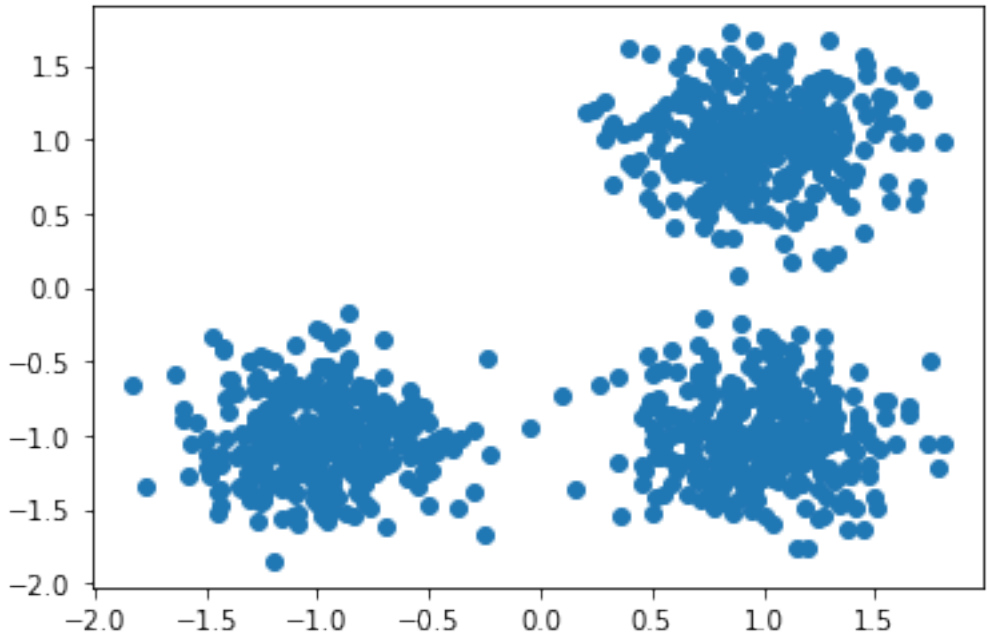
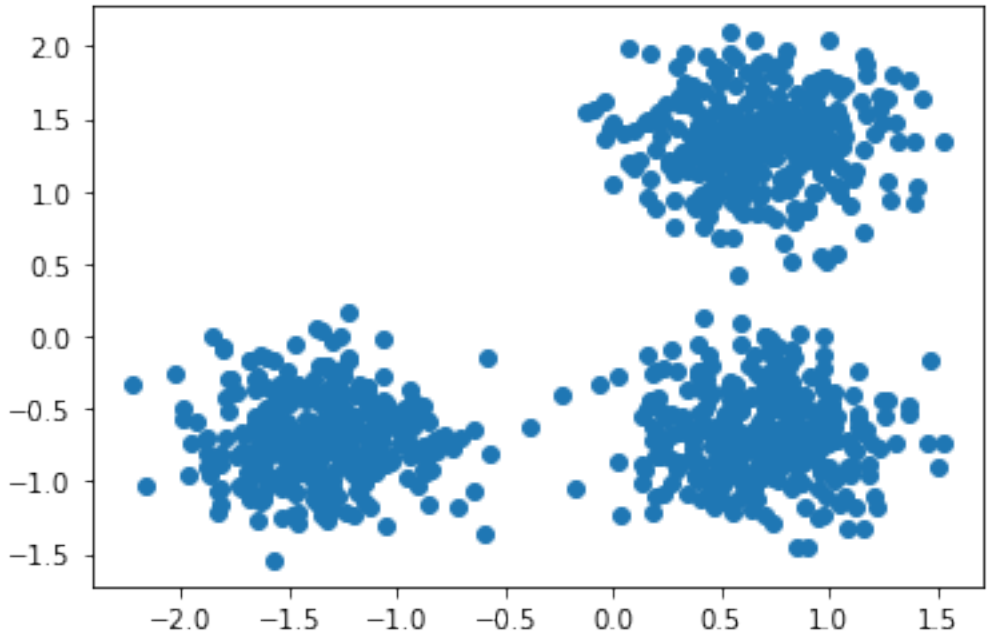
```
#Assignment 3
#1- Using Make_blob generate data of 1000 data points with three
cluster apply kmeans on it
#with k = 3 and use the metrics and get the accuracy (For Accuracy
take reference of DBSCAN
#evaluation)
#● Apply DBscan on Cust Segmentation Data

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

1000/3


centers=[[1,1],[-1,-1],[1,-1]]
x,labels_true=make_blobs(n_samples=1000,centers=centers,cluster_std=0.
3,random_state=0)

set(labels_true)

{0, 1, 2}

labels_true.shape

(1000,)

x.shape

(1000, 2)

x

array([[ 0.68543411,  0.57399462],
       [ 1.5292157 ,  1.12004716],
       [ 0.85126135, -0.63502669],
       ...,
       [-0.8178665 , -1.31445112],
       [-0.51052077, -0.88667225],
       [ 0.70827913, -0.59613368]])

plt.scatter(x[:,0],x[:,1])

<matplotlib.collections.PathCollection at 0x21b76729cd0>
```




```
x=StandardScaler().fit_transform(x)
plt.scatter(x[:,0],x[:,1])
plt.show()
```




```
x=np.nan_to_num(x)
Clus_dataset = StandardScaler().fit_transform(x)
Clus_dataset.shape
```

```
(1000, 2)
```

```
Clus_dataset
```

```
array([[ 0.36960469,  0.92478301],
       [ 1.24084714,  1.48033771],
       [ 0.54082878, -0.30527713],
       ...,
       [-1.1826208 , -0.9965246 ],
       [-0.86527251, -0.56130172],
       [ 0.39319321, -0.26570733]])
```

```
from sklearn.cluster import KMeans
k_means = KMeans(init="k-means++", n_clusters =3, n_init=100)
k_means.fit(x)
```

```
KMeans(n_clusters=3, n_init=100)
```

```
x.shape
```

```
(1000, 2)
```

```
labels = k_means.labels_
print(labels[0:5])
print(labels.shape)
print("number of classes= ",set(labels))
```

```
[2 2 0 1 0]
(1000,)
number of classes=  {0, 1, 2}
```

```
db =DBSCAN(eps=0.3,min_samples=15).fit(x)
labels=db.labels_
```

```
print()
print(set(labels))
print(set(labels_true))
```

```
{0, 1, 2, -1}
{0, 1, 2}
```

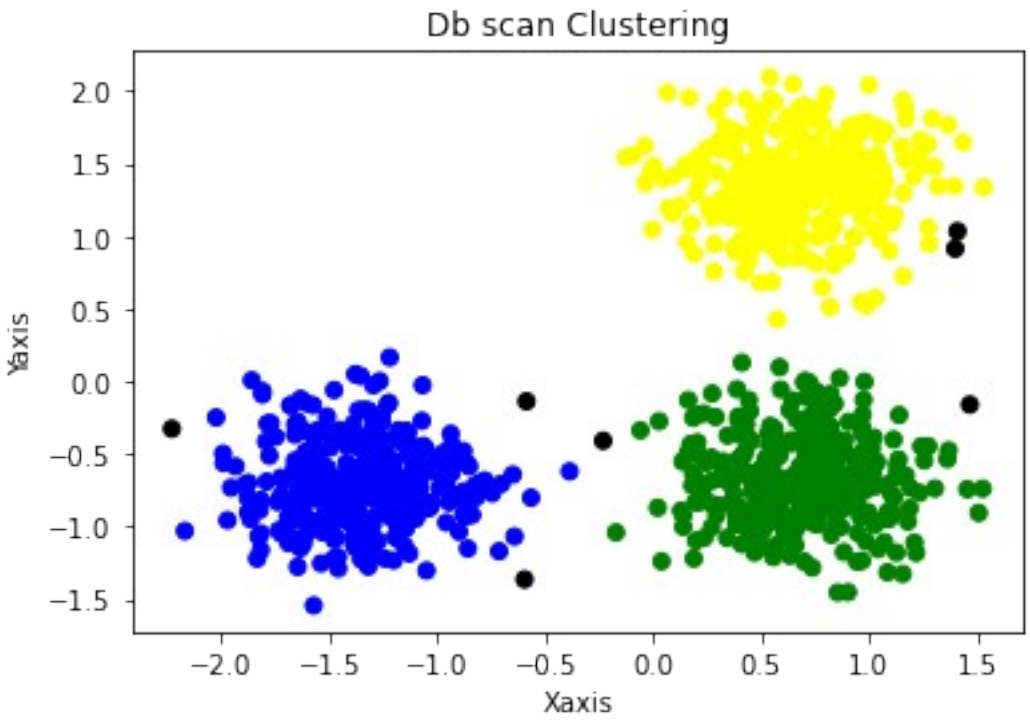
```
print(f"Homogeneity :
{metrics.homogeneity_score(labels_true,db.labels_)}")
```

```
Homogeneity :0.9931250065143498
```

```
set(db.labels_)
```

```
{-1, 0, 1, 2}
```

```
colormap = np.array(['yellow',"green","blue","black"])
plt.scatter(x[:,0],x[:,1],color = colormap[db.labels_])
plt.xlabel('Xaxis')
plt.ylabel('Yaxis')
plt.title("Db scan Clustering")
plt.show()
```




```
#2-Using dirtydata.csv Demonstrate all the techniques for removing the
null values
#● Replace by MEAN
#● Replace by MEDIAN
#● Replace by MODE
#● Replace by ARBITUARY VALUE
#● Replace by 0
df=pd.read_csv("E:\dirtydata.csv")
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

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  32 non-null     int64
 1   Date      31 non-null     object
 2   Pulse     32 non-null     int64
 3   Maxpulse  32 non-null     int64
 4   Calories  30 non-null     float64
dtypes: float64(1), int64(3), object(1)
memory usage: 1.4+ KB

df.isnull().sum()

Duration    0
Date        1
Pulse       0
Maxpulse    0
Calories    2
dtype: int64

len(df)

32

df.dropna()

    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
```

```
df.replace(np.nan,0)
```

```
    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       0.0
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             0    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       0.0
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
```

```
df[df['Date'].isnull()]
```

```
    Duration Date  Pulse  Maxpulse  Calories
22        45  NaN    100       119     282.0
```

```
df[df['Calories'].isnull()]
```

```
    Duration          Date  Pulse  Maxpulse  Calories
18        45  '2020/12/18'     90       112       NaN
28        60  '2020/12/28'    103       132       NaN
```

```
def impute_nan(df,variable,value):
    df[variable+"_mean"]=df[variable].fillna(value)
```

```
Calories_mean=df.Calories.mean()
Calories_mean
```

```
304.68
```

```
impute_nan(df,'Calories',Calories_mean)
```

```
df
```

```
    Duration          Date  Pulse  Maxpulse  Calories  Calories_mean
0         60  '2020/12/01'    110       130     409.1         409.10
1         60  '2020/12/02'    117       145     479.0         479.00
2         60  '2020/12/03'    103       135     340.0         340.00
3         45  '2020/12/04'    109       175     282.4         282.40
4         45  '2020/12/05'    117       148     406.0         406.00
5         60  '2020/12/06'    102       127     300.0         300.00
6         60  '2020/12/07'    110       136     374.0         374.00
7        450  '2020/12/08'    104       134     253.3         253.30
8         30  '2020/12/09'    109       133     195.1         195.10
9         60  '2020/12/10'     98       124     269.0         269.00
10        60  '2020/12/11'    103       147     329.3         329.30
11        60  '2020/12/12'    100       120     250.7         250.70
12        60  '2020/12/12'    100       120     250.7         250.70
13        60  '2020/12/13'    106       128     345.3         345.30
14        60  '2020/12/14'    104       132     379.3         379.30
15        60  '2020/12/15'     98       123     275.0         275.00
16        60  '2020/12/16'     98       120     215.2         215.20
17        60  '2020/12/17'    100       120     300.0         300.00
```

```
18        45  '2020/12/18'     90       112       NaN         304.68
19        60  '2020/12/19'    103       123     323.0         323.00
20        45  '2020/12/20'     97       125     243.0         243.00
21        60  '2020/12/21'    108       131     364.2         364.20
22        45           NaN    100       119     282.0         282.00
23        60  '2020/12/23'    130       101     300.0         300.00
24        45  '2020/12/24'    105       132     246.0         246.00
25        60  '2020/12/25'    102       126     334.5         334.50
26        60      20201226    100       120     250.0         250.00
27        60  '2020/12/27'     92       118     241.0         241.00
28        60  '2020/12/28'    103       132       NaN         304.68
29        60  '2020/12/29'    100       132     280.0         280.00
30        60  '2020/12/30'    102       129     380.3         380.30
31        60  '2020/12/31'     92       115     243.0         243.00
```

```
def impute_nan_median(df,variable,value):
    df[variable+"_medain"]=df[variable].fillna(value)

median=df.Calories.median()
median
print("The meadin from Calories coloumn which is avaibale in df
dataframe is",median)
```

```
The meadin from Calories coloumn which is avaibale in df dataframe is
291.2
```

```
impute_nan_median(df,'Calories',median)
df
```

```
    Duration          Date  Pulse  Maxpulse  Calories
Calories_mean  \
0         60  '2020/12/01'    110       130     409.1         409.10

1         60  '2020/12/02'    117       145     479.0         479.00

2         60  '2020/12/03'    103       135     340.0         340.00

3         45  '2020/12/04'    109       175     282.4         282.40

4         45  '2020/12/05'    117       148     406.0         406.00

5         60  '2020/12/06'    102       127     300.0         300.00

6         60  '2020/12/07'    110       136     374.0         374.00

7        450  '2020/12/08'    104       134     253.3         253.30

8         30  '2020/12/09'    109       133     195.1         195.10

9         60  '2020/12/10'     98       124     269.0         269.00
```

```
10        60  '2020/12/11'    103       147     329.3         329.30
11        60  '2020/12/12'    100       120     250.7         250.70
12        60  '2020/12/12'    100       120     250.7         250.70
13        60  '2020/12/13'    106       128     345.3         345.30
14        60  '2020/12/14'    104       132     379.3         379.30
15        60  '2020/12/15'     98       123     275.0         275.00
16        60  '2020/12/16'     98       120     215.2         215.20
17        60  '2020/12/17'    100       120     300.0         300.00
18        45  '2020/12/18'     90       112       NaN         304.68
19        60  '2020/12/19'    103       123     323.0         323.00
20        45  '2020/12/20'     97       125     243.0         243.00
21        60  '2020/12/21'    108       131     364.2         364.20
22        45           NaN    100       119     282.0         282.00
23        60  '2020/12/23'    130       101     300.0         300.00
24        45  '2020/12/24'    105       132     246.0         246.00
25        60  '2020/12/25'    102       126     334.5         334.50
26        60      20201226    100       120     250.0         250.00
27        60  '2020/12/27'     92       118     241.0         241.00
28        60  '2020/12/28'    103       132       NaN         304.68
29        60  '2020/12/29'    100       132     280.0         280.00
30        60  '2020/12/30'    102       129     380.3         380.30
31        60  '2020/12/31'     92       115     243.0         243.00

    Calories_medain
0             409.1
```

```
1             479.0
2             340.0
3             282.4
4             406.0
5             300.0
6             374.0
7             253.3
8             195.1
9             269.0
10            329.3
11            250.7
12            250.7
13            345.3
14            379.3
15            275.0
16            215.2
17            300.0
18            291.2
19            323.0
20            243.0
21            364.2
22            282.0
23            300.0
24            246.0
25            334.5
26            250.0
27            241.0
28            291.2
29            280.0
30            380.3
31            243.0
```

```
def impute_nan_mode(df,variable,value):
    df[variable+"_mode"]=df[variable].fillna(value)

mode =df.Calories.mode()
type(mode)
mode
```

```
0    300.0
Name: Calories, dtype: float64
```

```
mode[0]
```

```
300.0
```

```
print(mode)
impute_nan_mode(df,'Calories',mode[0])
```

```
0    300.0
Name: Calories, dtype: float64
```

```
df
```

	Duration	Date	Pulse	Maxpulse	Calories	Calories_mean
0	60	'2020/12/01'	110	130	409.1	409.10
1	60	'2020/12/02'	117	145	479.0	479.00
2	60	'2020/12/03'	103	135	340.0	340.00
3	45	'2020/12/04'	109	175	282.4	282.40
4	45	'2020/12/05'	117	148	406.0	406.00
5	60	'2020/12/06'	102	127	300.0	300.00
6	60	'2020/12/07'	110	136	374.0	374.00
7	450	'2020/12/08'	104	134	253.3	253.30
8	30	'2020/12/09'	109	133	195.1	195.10
9	60	'2020/12/10'	98	124	269.0	269.00
10	60	'2020/12/11'	103	147	329.3	329.30
11	60	'2020/12/12'	100	120	250.7	250.70
12	60	'2020/12/12'	100	120	250.7	250.70
13	60	'2020/12/13'	106	128	345.3	345.30
14	60	'2020/12/14'	104	132	379.3	379.30
15	60	'2020/12/15'	98	123	275.0	275.00
16	60	'2020/12/16'	98	120	215.2	215.20
17	60	'2020/12/17'	100	120	300.0	300.00
18	45	'2020/12/18'	90	112	NaN	304.68
19	60	'2020/12/19'	103	123	323.0	323.00
20	45	'2020/12/20'	97	125	243.0	243.00
21	60	'2020/12/21'	108	131	364.2	364.20

```
22        45           NaN    100       119     282.0         282.00

23        60  '2020/12/23'    130       101     300.0         300.00

24        45  '2020/12/24'    105       132     246.0         246.00

25        60  '2020/12/25'    102       126     334.5         334.50

26        60      20201226    100       120     250.0         250.00

27        60  '2020/12/27'     92       118     241.0         241.00

28        60  '2020/12/28'    103       132       NaN         304.68

29        60  '2020/12/29'    100       132     280.0         280.00

30        60  '2020/12/30'    102       129     380.3         380.30

31        60  '2020/12/31'     92       115     243.0         243.00
```

	Calories_medain	Calories_mode
0	409.1	409.1
1	479.0	479.0
2	340.0	340.0
3	282.4	282.4
4	406.0	406.0
5	300.0	300.0
6	374.0	374.0
7	253.3	253.3
8	195.1	195.1
9	269.0	269.0
10	329.3	329.3
11	250.7	250.7
12	250.7	250.7
13	345.3	345.3
14	379.3	379.3
15	275.0	275.0
16	215.2	215.2
17	300.0	300.0
18	291.2	300.0
19	323.0	323.0
20	243.0	243.0
21	364.2	364.2
22	282.0	282.0
23	300.0	300.0
24	246.0	246.0
25	334.5	334.5
26	250.0	250.0

```
27            241.0          241.0
28            291.2          300.0
29            280.0          280.0
30            380.3          380.3
31            243.0          243.0
```

```
df['arb_num_calories']=df['Calories'].fillna(250)
df['arb_num_date']=df['Date'].fillna("'2020/12/22'")
df
```

```
    Duration          Date  Pulse  Maxpulse  Calories
Calories_mean  \
0          60  '2020/12/01'    110       130     409.1          409.10

1          60  '2020/12/02'    117       145     479.0          479.00

2          60  '2020/12/03'    103       135     340.0          340.00

3          45  '2020/12/04'    109       175     282.4          282.40

4          45  '2020/12/05'    117       148     406.0          406.00

5          60  '2020/12/06'    102       127     300.0          300.00

6          60  '2020/12/07'    110       136     374.0          374.00

7         450  '2020/12/08'    104       134     253.3          253.30

8          30  '2020/12/09'    109       133     195.1          195.10

9          60  '2020/12/10'     98       124     269.0          269.00

10         60  '2020/12/11'    103       147     329.3          329.30

11         60  '2020/12/12'    100       120     250.7          250.70

12         60  '2020/12/12'    100       120     250.7          250.70

13         60  '2020/12/13'    106       128     345.3          345.30

14         60  '2020/12/14'    104       132     379.3          379.30

15         60  '2020/12/15'     98       123     275.0          275.00

16         60  '2020/12/16'     98       120     215.2          215.20

17         60  '2020/12/17'    100       120     300.0          300.00

18         45  '2020/12/18'     90       112       NaN          304.68
```

19	60	'2020/12/19'	103	123	323.0	323.00
20	45	'2020/12/20'	97	125	243.0	243.00
21	60	'2020/12/21'	108	131	364.2	364.20
22	45	NaN	100	119	282.0	282.00
23	60	'2020/12/23'	130	101	300.0	300.00
24	45	'2020/12/24'	105	132	246.0	246.00
25	60	'2020/12/25'	102	126	334.5	334.50
26	60	20201226	100	120	250.0	250.00
27	60	'2020/12/27'	92	118	241.0	241.00
28	60	'2020/12/28'	103	132	NaN	304.68
29	60	'2020/12/29'	100	132	280.0	280.00
30	60	'2020/12/30'	102	129	380.3	380.30
31	60	'2020/12/31'	92	115	243.0	243.00

	Calories_medain	Calories_mode	arb_num_calories	arb_num_date
0	409.1	409.1	409.1	'2020/12/01'
1	479.0	479.0	479.0	'2020/12/02'
2	340.0	340.0	340.0	'2020/12/03'
3	282.4	282.4	282.4	'2020/12/04'
4	406.0	406.0	406.0	'2020/12/05'
5	300.0	300.0	300.0	'2020/12/06'
6	374.0	374.0	374.0	'2020/12/07'
7	253.3	253.3	253.3	'2020/12/08'
8	195.1	195.1	195.1	'2020/12/09'
9	269.0	269.0	269.0	'2020/12/10'
10	329.3	329.3	329.3	'2020/12/11'
11	250.7	250.7	250.7	'2020/12/12'
12	250.7	250.7	250.7	'2020/12/12'
13	345.3	345.3	345.3	'2020/12/13'
14	379.3	379.3	379.3	'2020/12/14'
15	275.0	275.0	275.0	'2020/12/15'
16	215.2	215.2	215.2	'2020/12/16'
17	300.0	300.0	300.0	'2020/12/17'
18	291.2	300.0	250.0	'2020/12/18'
19	323.0	323.0	323.0	'2020/12/19'

20	243.0	243.0	243.0	'2020/12/20'
21	364.2	364.2	364.2	'2020/12/21'
22	282.0	282.0	282.0	'2020/12/22'
23	300.0	300.0	300.0	'2020/12/23'
24	246.0	246.0	246.0	'2020/12/24'
25	334.5	334.5	334.5	'2020/12/25'
26	250.0	250.0	250.0	20201226
27	241.0	241.0	241.0	'2020/12/27'
28	291.2	300.0	250.0	'2020/12/28'
29	280.0	280.0	280.0	'2020/12/29'
30	380.3	380.3	380.3	'2020/12/30'
31	243.0	243.0	243.0	'2020/12/31'