In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
raw_data=pd.read_csv('Bellafit_tracker.csv')
```

In [3]:
```
raw_data.head()
```

Out[3]:

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	SedentaryMinutes	Calori
0	1503960366	04-12-2016	13162	8.50	8.50	728	19
1	1503960366	4/13/2016	10735	6.97	6.97	776	17
2	1503960366	4/14/2016	10460	6.74	6.74	1218	17
3	1503960366	4/15/2016	9762	6.28	6.28	726	17
4	1503960366	4/16/2016	12669	8.16	8.16	773	18

In [4]:
```
raw_data.info()
```

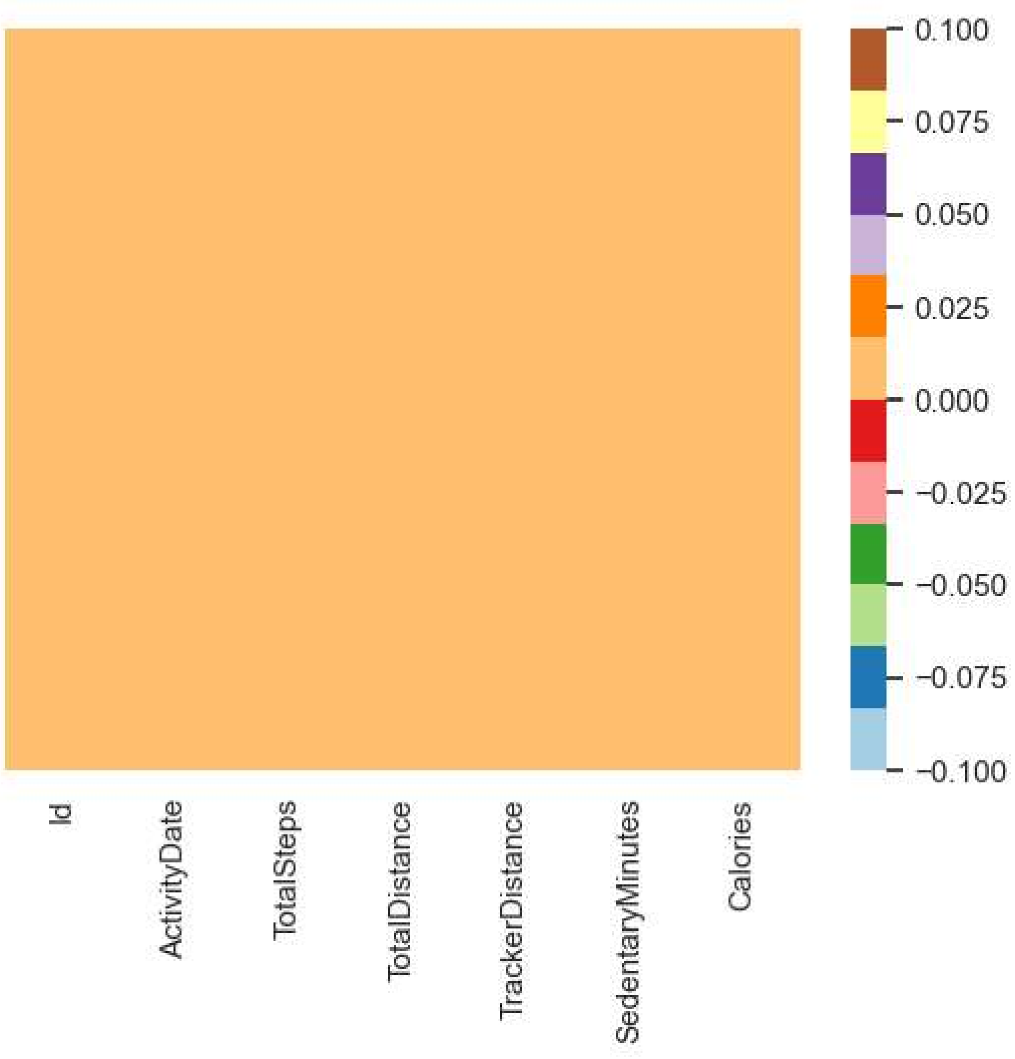
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 629 entries, 0 to 628
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Id                629 non-null    int64  
 1   ActivityDate      629 non-null    object 
 2   TotalSteps        629 non-null    int64  
 3   TotalDistance     629 non-null    float64
 4   TrackerDistance   629 non-null    float64
 5   SedentaryMinutes  629 non-null    int64  
 6   Calories          629 non-null    int64  
dtypes: float64(2), int64(4), object(1)
memory usage: 34.5+ KB
```

In [5]:
```
raw_data.isnull().sum()
```

Out[5]:
```
Id                  0
ActivityDate        0
TotalSteps          0
TotalDistance       0
TrackerDistance     0
SedentaryMinutes    0
Calories            0
dtype: int64
```

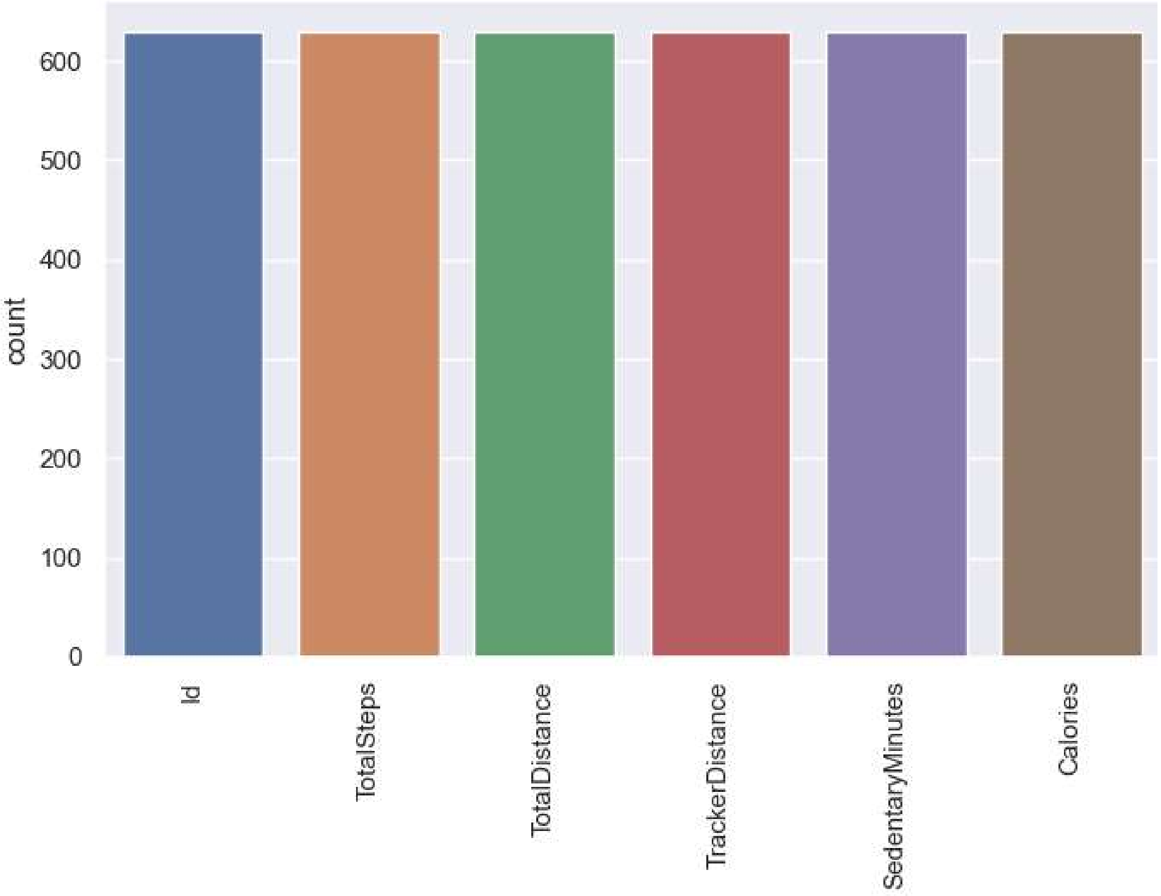
In [6]:
```
sns.heatmap(raw_data.isnull(),yticklabels=False,cmap="Paired")
```

Out[6]:
```
<Axes: >
```



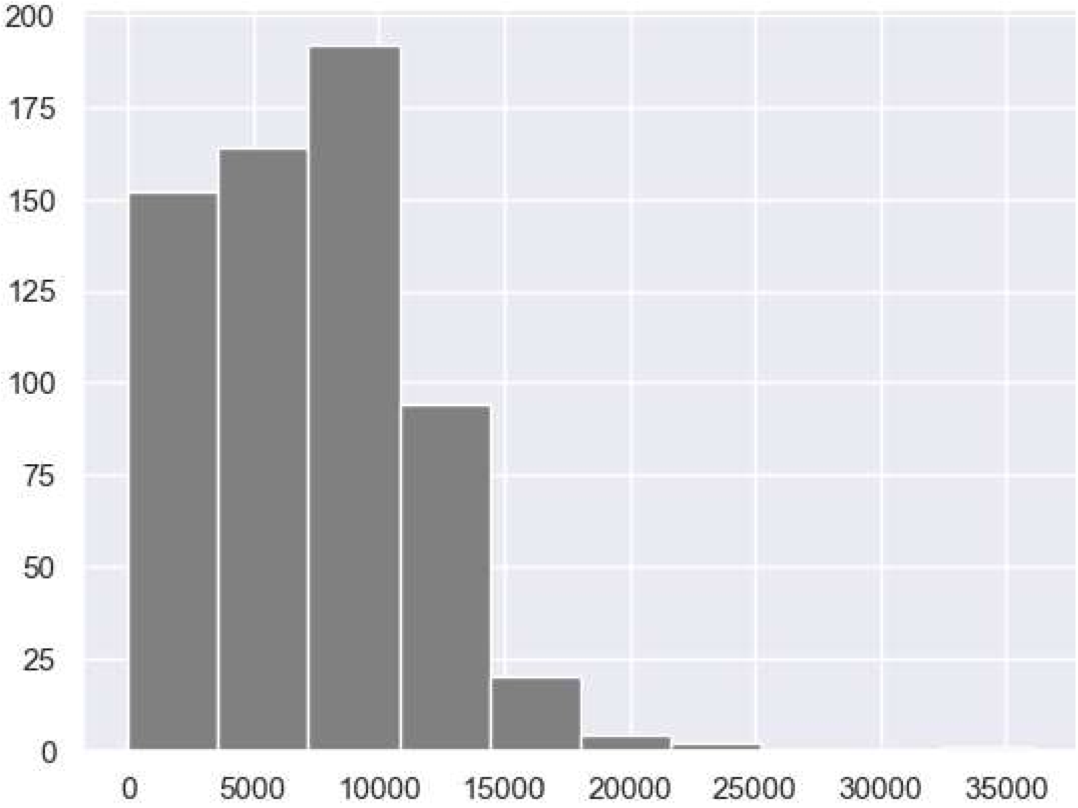
In [7]:
```
plt.figure(figsize=(8,5))
sns.countplot(raw_data)
plt.xticks(rotation=90)
```

Out[7]:
```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Id'),
  Text(1, 0, 'TotalSteps'),
  Text(2, 0, 'TotalDistance'),
  Text(3, 0, 'TrackerDistance'),
  Text(4, 0, 'SedentaryMinutes'),
  Text(5, 0, 'Calories')])
```



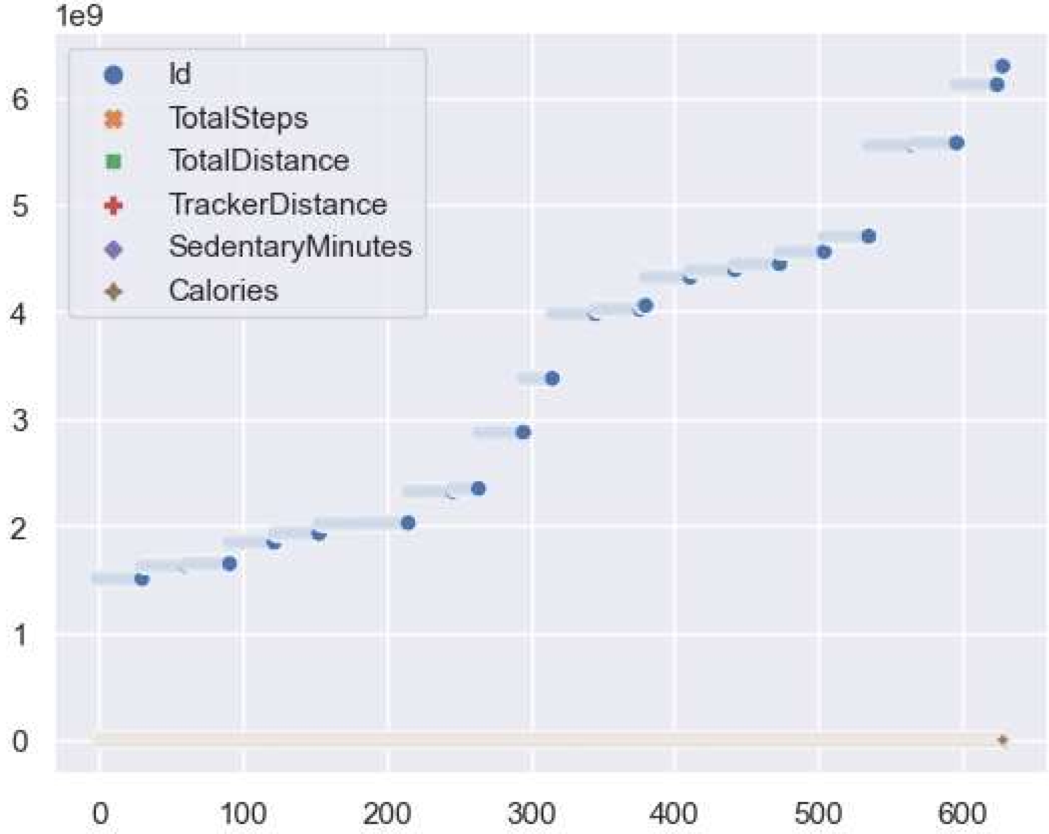
```
In [8]: plt.hist(raw_data['TotalSteps'], bins=10, color='grey')
```

```
Out[8]: (array([152., 164., 192.,  94.,  20.,   4.,   2.,   0.,   0.,   1.]),
         array([8.00000e+00, 3.60910e+03, 7.21020e+03, 1.08113e+04, 1.44124e+04,
                1.80135e+04, 2.16146e+04, 2.52157e+04, 2.88168e+04, 3.24179e+04,
                3.60190e+04]),
         <BarContainer object of 10 artists>)
```



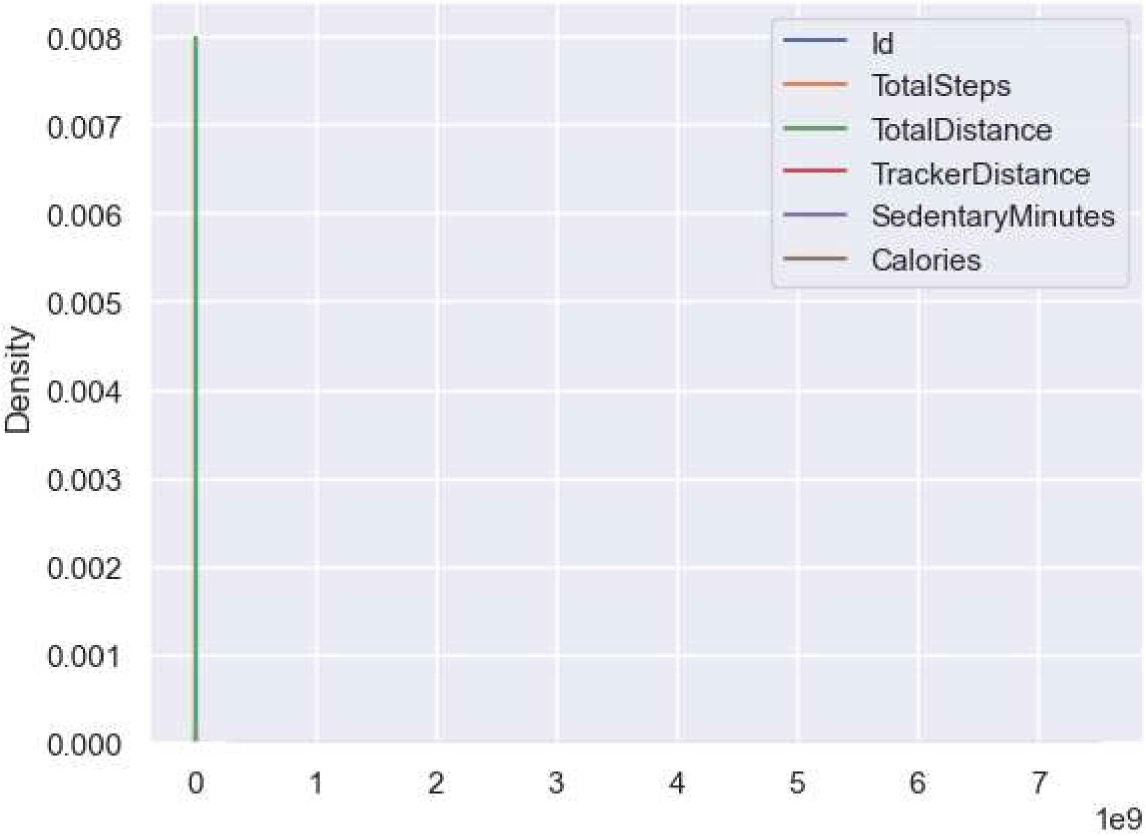
In [9]:
```
sns.scatterplot(raw_data)
```

Out[9]:
```
<Axes: >
```



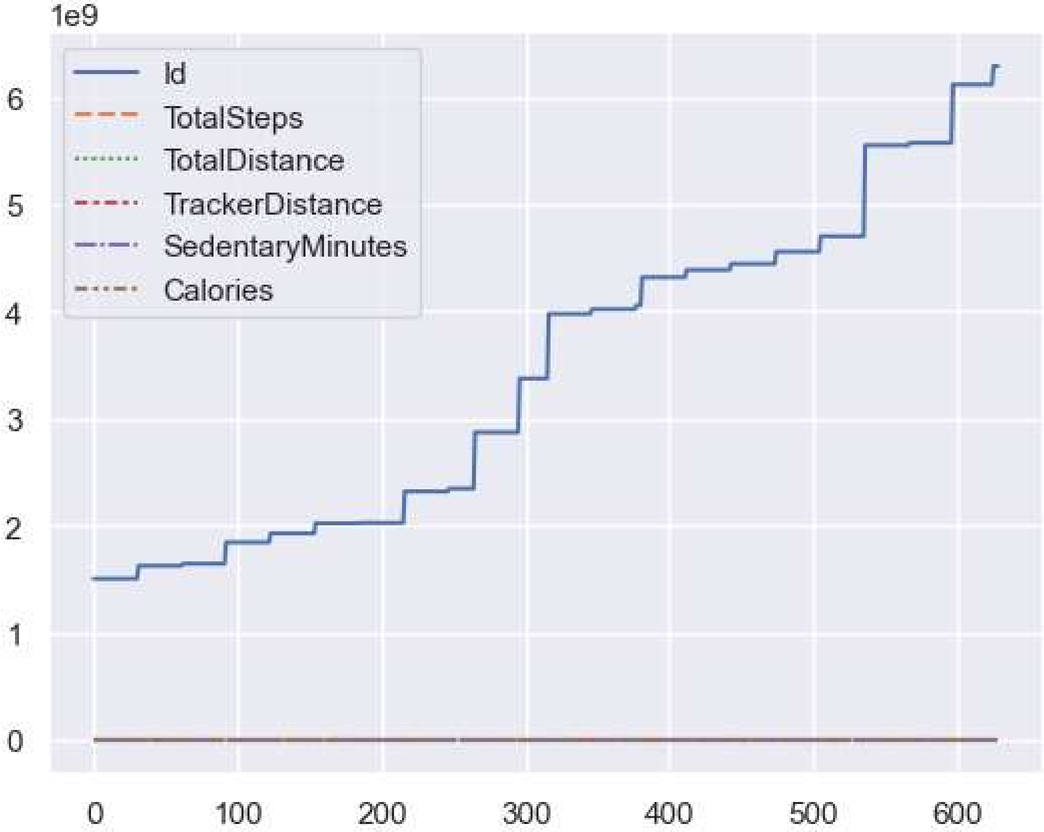
In [10]:
```
sns.kdeplot(raw_data)
```

Out[10]:
```
<Axes: ylabel='Density'>
```

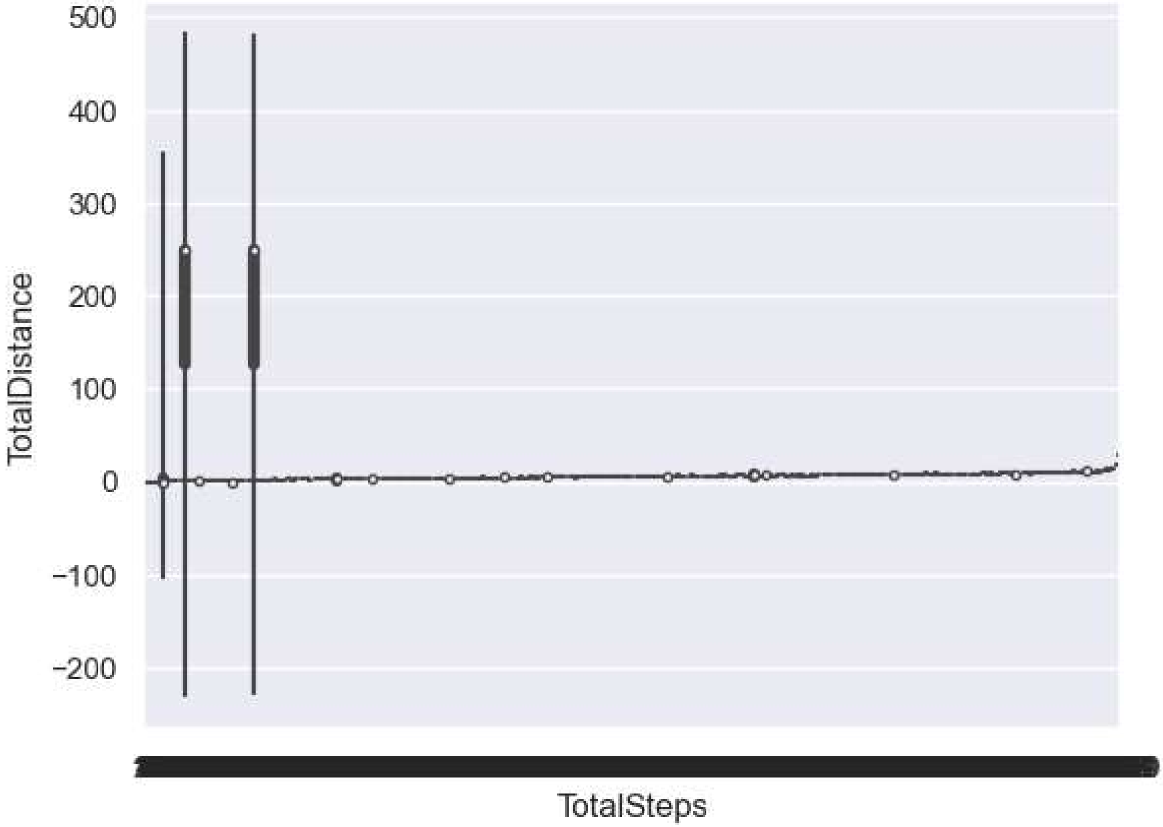


```
In [11]: sns.lineplot(raw_data)
```

```
Out[11]: <Axes: >
```



In [12]:
```
sns.violinplot(x='TotalSteps',y='TotalDistance',data=raw_data)
plt.show()
```



In [13]:
```
raw_data.dtypes
```

Out[13]:
```
Id                    int64
ActivityDate         object
TotalSteps            int64
TotalDistance       float64
TrackerDistance     float64
SedentaryMinutes      int64
Calories              int64
dtype: object
```

In [14]:

```
raw_data
```

Out[14]:

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	SedentaryMinutes	Cal
0	1503960366	04-12-2016	13162	8.50	8.50	728	
1	1503960366	4/13/2016	10735	6.97	6.97	776	
2	1503960366	4/14/2016	10460	6.74	6.74	1218	
3	1503960366	4/15/2016	9762	6.28	6.28	726	
4	1503960366	4/16/2016	12669	8.16	8.16	773	
...	...	...	...	...	...	...	
624	6117666160	05-09-2016	4477	3.38	3.38	125	
625	6290855005	04-12-2016	4562	3.45	3.45	1241	
626	6290855005	4/13/2016	7142	5.40	5.40	1090	
627	6290855005	4/14/2016	7671	5.80	5.80	1077	
628	6290855005	4/15/2016	9501	7.18	7.18	1112	

629 rows × 7 columns

In [15]:

```
from scipy.stats import zscore
z_scores = zscore(raw_data['TotalDistance'])
z_score_outliers=(z_scores<-3)|(z_scores>3)
z_score_outlier_rows=raw_data[z_score_outliers]
print("outliers detected by Z-score:",z_score_outlier_rows)
```

```
outliers detected by Z-score:            Id ActivityDate  TotalSteps  Total
Distance  TrackerDistance  \
127  1927972279    4/16/2016         980          250.0            250.0
128  1927972279    4/17/2016         980          250.0            250.0
140  1927972279    4/29/2016        2704          250.0            250.0
141  1927972279    4/30/2016        2704          250.0            250.0
347  4020332650    4/13/2016         250          250.0            250.0
365  4020332650   05-01-2016         250          250.0            250.0
379  4057192912    4/14/2016         250          250.0            250.0
524  4702921684   05-01-2016         250          250.0            250.0

     SedentaryMinutes  Calories
127              1440      2064
128              1440      2063
140              1440      2063
141              1440      2064
347              1440      1981
365              1440      1980
379              1440      1776
524              1440      2017
```

In [16]:

```
x=(z_scores>-3)&(z_scores<3)
df1=raw_data[x]
print(df1)
```

```
             Id ActivityDate  TotalSteps  TotalDistance  TrackerDistance  \
0    1503960366   04-12-2016       13162           8.50             8.50   
1    1503960366    4/13/2016       10735           6.97             6.97   
2    1503960366    4/14/2016       10460           6.74             6.74   
3    1503960366    4/15/2016        9762           6.28             6.28   
4    1503960366    4/16/2016       12669           8.16             8.16   
..          ...          ...         ...            ...              ...   
624  6117666160   05-09-2016        4477           3.38             3.38   
625  6290855005   04-12-2016        4562           3.45             3.45   
626  6290855005    4/13/2016        7142           5.40             5.40   
627  6290855005    4/14/2016        7671           5.80             5.80   
628  6290855005    4/15/2016        9501           7.18             7.18   

     SedentaryMinutes  Calories  
0                 728      1985  
1                 776      1797  
2                1218      1776  
3                 726      1745  
4                 773      1863  
..                ...       ...  
624               125      1248  
625              1241      2560  
626              1090      2905  
627              1077      2952  
628              1112      2896  

[621 rows x 7 columns]
```

In [17]:

```
from scipy.stats import zscore
z_scores = zscore(df1['TotalSteps'])
z_score_outliers=(z_scores<-3)|(z_scores>3)
z_score_outlier_rows=df1[z_score_outliers]
print("outliers detected by Z-score:",z_score_outlier_rows)
```

```
outliers detected by Z-score:              Id ActivityDate  TotalSteps  Total
Distance  TrackerDistance  \
50   1624580081   05-01-2016       36019      28.030001        28.030001   
251  2347167796    4/16/2016       22244      15.080000        15.080000   
437  4388161847   05-07-2016       22770      17.540001        17.540001   

     SedentaryMinutes  Calories  
50               1020      2690  
251               968      2670  
437               508      4022  
```

In [18]:
```
x=(z_scores>-3)&(z_scores<3)
dff=df1[x]
print(dff)
```

```
             Id ActivityDate  TotalSteps  TotalDistance  TrackerDistance  \
0    1503960366   04-12-2016       13162           8.50             8.50   
1    1503960366    4/13/2016       10735           6.97             6.97   
2    1503960366    4/14/2016       10460           6.74             6.74   
3    1503960366    4/15/2016        9762           6.28             6.28   
4    1503960366    4/16/2016       12669           8.16             8.16   
..          ...          ...         ...            ...              ...   
624  6117666160   05-09-2016        4477           3.38             3.38   
625  6290855005   04-12-2016        4562           3.45             3.45   
626  6290855005    4/13/2016        7142           5.40             5.40   
627  6290855005    4/14/2016        7671           5.80             5.80   
628  6290855005    4/15/2016        9501           7.18             7.18   

     SedentaryMinutes  Calories  
0                 728      1985  
1                 776      1797  
2                1218      1776  
3                 726      1745  
4                 773      1863  
..                ...       ...  
624               125      1248  
625              1241      2560  
626              1090      2905  
627              1077      2952  
628              1112      2896  

[618 rows x 7 columns]
```

In [26]:
```
dff.drop('ActivityDate', axis=1, inplace=True)
```

In [27]:
```
from sklearn.preprocessing import StandardScaler
scale = StandardScaler().fit(dff)
dff = scale.transform(dff)
features_scaled = pd.DataFrame( dff, columns= raw_data.columns)
features_scaled.head()
```

Out[27]:

	Id	TotalSteps	TotalDistance	TrackerDistance	SedentaryMinutes	Calories
0	-1.304116	1.452212	1.175632	1.175632	-0.769412	-0.264036
1	-1.304116	0.862949	0.654844	0.654844	-0.609900	-0.549606
2	-1.304116	0.796180	0.576555	0.576555	0.858944	-0.581504
3	-1.304116	0.626709	0.419978	0.419978	-0.776059	-0.628593
4	-1.304116	1.332514	1.059901	1.059901	-0.619869	-0.449352

In [28]:
```
from sklearn.cluster import KMeans
```

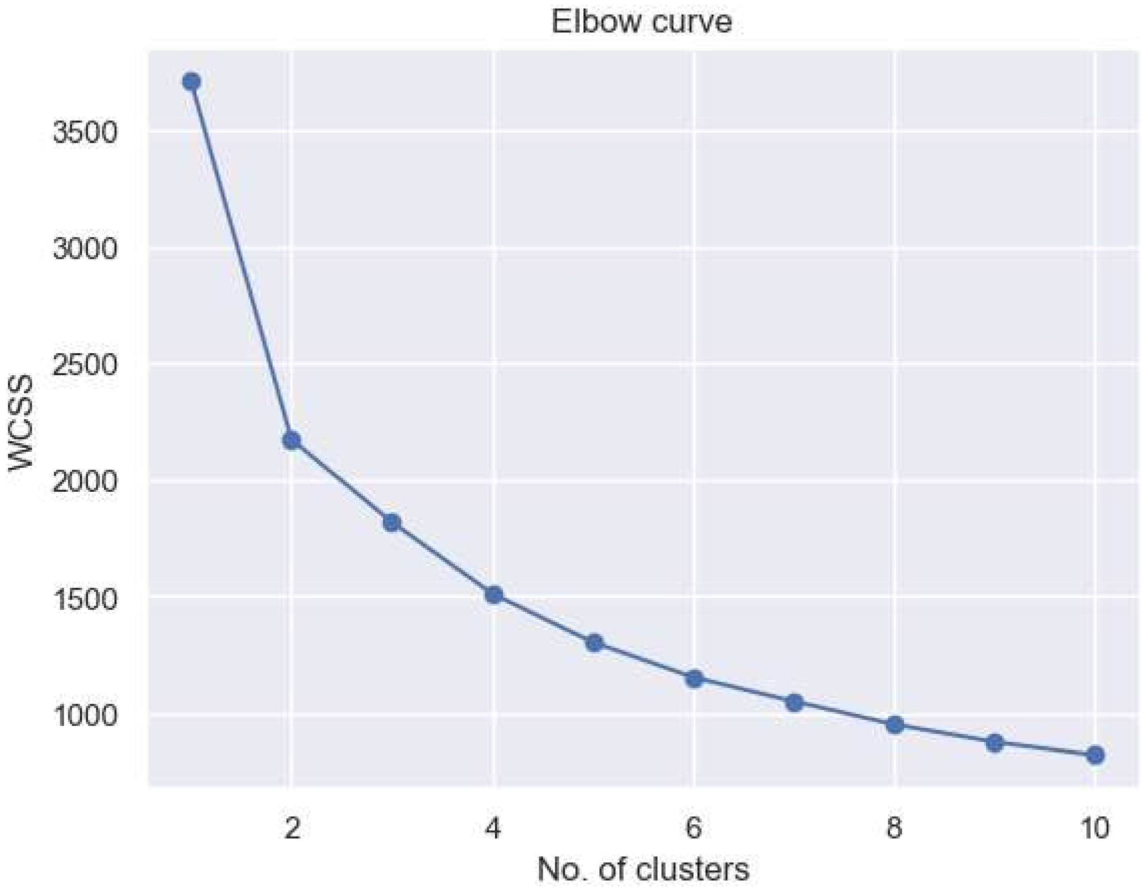
In [29]:
```
wcss = []
for i in range(1,11):
 kmeans=KMeans(n_clusters=i,init='k-means++',random_state=25)
 kmeans.fit(features_scaled)

 wcss.append(kmeans.inertia_)
```

In [30]:
```
wcss
```

Out[30]:
```
[3708.0000000000005,
 2175.841220851975,
 1819.0640059045031,
 1512.9264786000306,
 1304.0651072495848,
 1153.469883721601,
 1050.0921930875998,
 950.9928347871928,
 876.165439812513,
 820.0044652888889]
```

In [31]:
```
sns.set()
plt.plot(range(1,11),wcss, marker="o")
plt.title('Elbow curve')
plt.xlabel('No. of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [32]:

```
kmeans=KMeans(n_clusters=4,init='k-means++',random_state=0)
y=kmeans.fit_predict(features_scaled)
```

In [33]:

```
print(y)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 0 2 2 2
 0 2 2 2 2 0 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 2 0 0 0 0 2 2 2 0 2
 2 0 2 0 2 3 0 1 0 2 2 0 0 0 0 0 2 0 2 0 1 2 2 2 2 2 0 2 2 2 2 2 2 0 2 1 1
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 1 1 1 1 2
 2 2 2 1 2 1 0 1 1 2 1 1 1 1 1 1 2 1 0 0 1 1 0 2 0 0 0 0 2 2 2 2 2 2 0 2 2
 2 2 2 2 2 2 2 2 2 2 0 2 2 2 0 2 2 2 2 2 0 0 0 0 1 0 1 0 0 0 3 0 0 1 0 1 1
 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 2 0 0 0 1 1 2 0 0 0 1 0
 1 0 1 0 2 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0
 0 0 0 0 0 0 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 3 1 1 1 1 1 2 1 1
 1 1 0 0 3 1 1 2 1 3 3 2 1 1 1 3 3 3 3 1 1 1 3 3 0 3 3 1 1 3 3 3 1 0 3 3 3
 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 2 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 0 0 1 1 3 0 1 0 1 0 0 0
 1 0 1 1 1 1 0 1 1 1 0 1 0 1 3 3 3 3 3 3 3 3 3 1 2 3 3 3 1 3 3 3 3 3 3 3 3
 3 3 3 3 3 3 1 3 1 3 3 1 1 1 3 1 3 3 1 1 3 3 3 3 3 1 1 3 3 1 3 3 1 1 3 3 1
 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 3 3 1
 1 3 3 3 3 3 3 1 3 3 1 3 1 3 1 1 1 3 1 3 3 1 1 3 3 3]
```

In [34]:

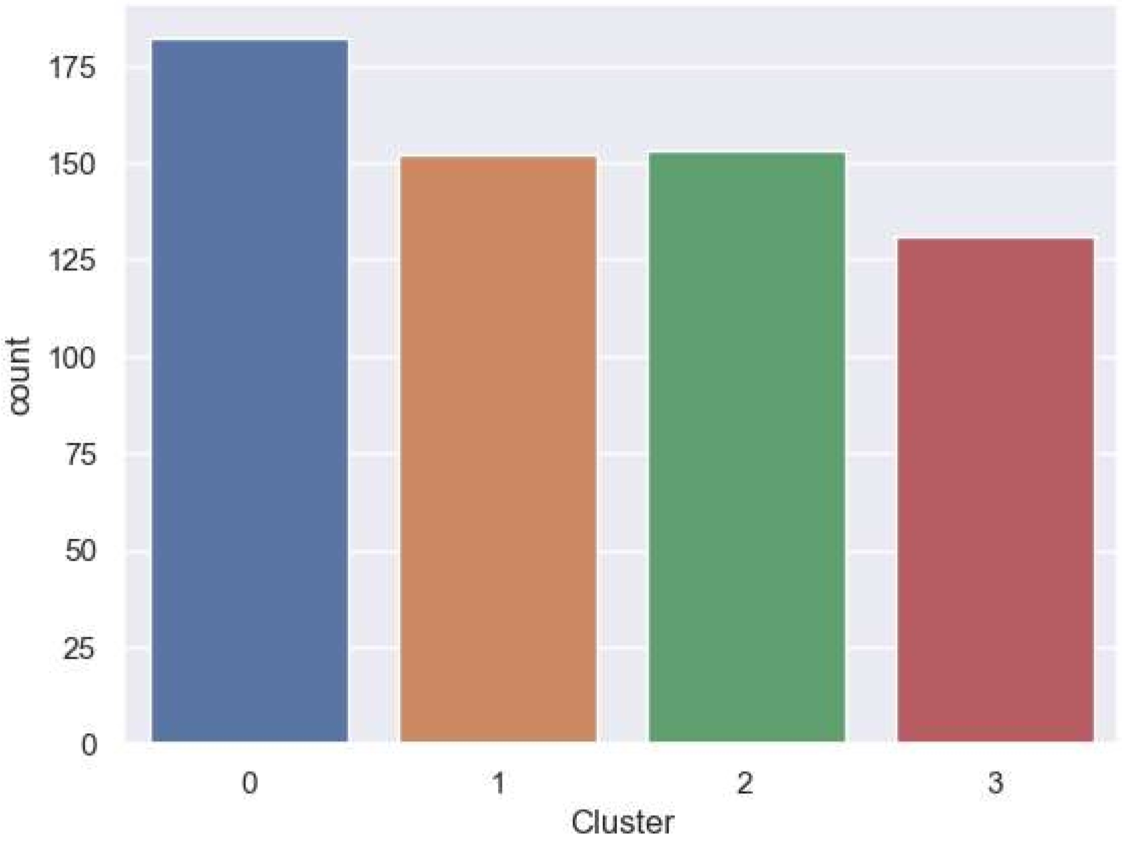
```
data_output = features_scaled.copy(deep = True)
data_output['Cluster'] = kmeans.labels_
data_output.head()
```

Out[34]:

	Id	TotalSteps	TotalDistance	TrackerDistance	SedentaryMinutes	Calories	Cluster
0	-1.304116	1.452212	1.175632	1.175632	-0.769412	-0.264036	0
1	-1.304116	0.862949	0.654844	0.654844	-0.609900	-0.549606	0
2	-1.304116	0.796180	0.576555	0.576555	0.858944	-0.581504	0
3	-1.304116	0.626709	0.419978	0.419978	-0.776059	-0.628593	0
4	-1.304116	1.332514	1.059901	1.059901	-0.619869	-0.449352	0

```
In [35]: sns.countplot(x='Cluster',data=data_output)
```

```
Out[35]: <Axes: xlabel='Cluster', ylabel='count'>
```



```
In [36]: np.unique(kmeans.labels_, return_counts=True)
```

```
Out[36]: (array([0, 1, 2, 3]), array([182, 152, 153, 131], dtype=int64))
```

```
In [37]: from sklearn.metrics import silhouette_score, calinski_harabasz_score,davies_b
         silhouette_avg = silhouette_score(features_scaled, y)
         print(f"Silhouette Score: {silhouette_avg}")
```

```
Silhouette Score: 0.27505378864386226
```

```
In [38]: calinski_harabasz_index = calinski_harabasz_score(features_scaled, y)
         print(f"Calinski-Harabasz Index: {calinski_harabasz_index}")
```

```
Calinski-Harabasz Index: 296.946604522984
```

```
In [39]: davies_bouldin_index = davies_bouldin_score(features_scaled, y)
         print(f"Davies-Bouldin Index: {davies_bouldin_index}")
```

```
Davies-Bouldin Index: 1.2957328426411534
```

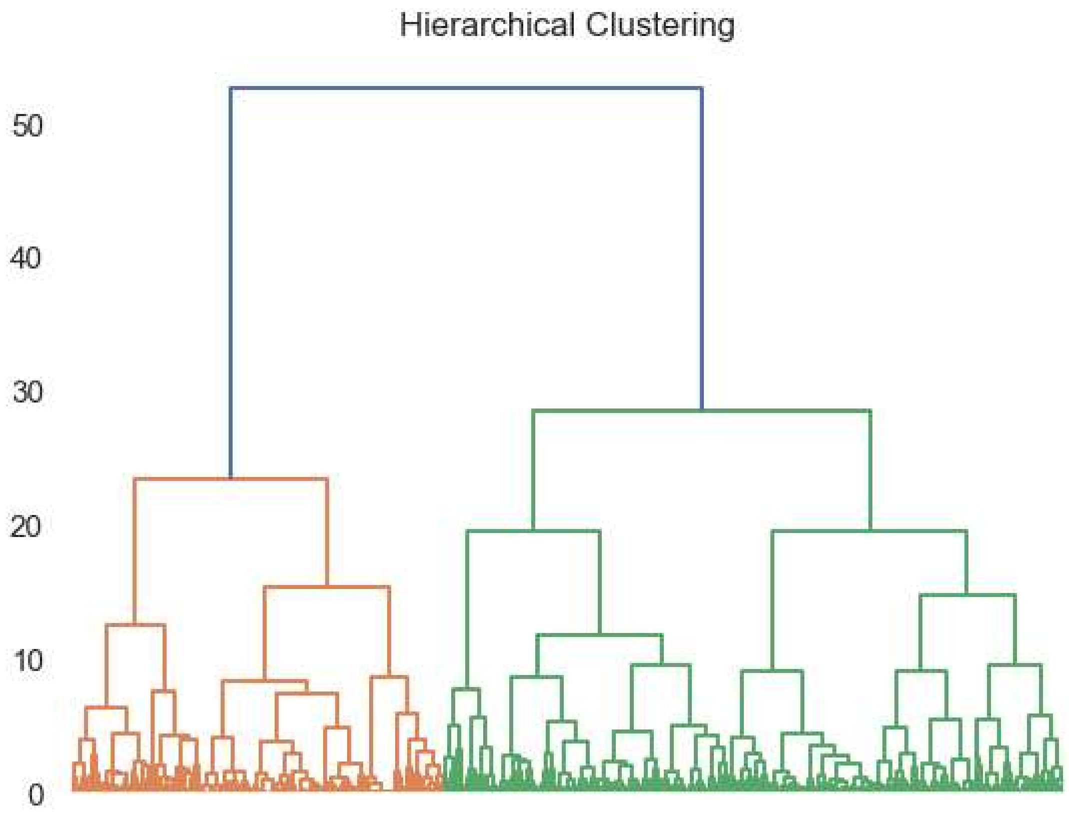
In [40]:
```
import scipy.cluster.hierarchy as sch
from sklearn.preprocessing import scale as s
from scipy.cluster.hierarchy import dendrogram, linkage
```

In [41]:
```
Z = sch.linkage(features_scaled,method='ward')
Z
```

Out[41]:
```
array([[ 144.        ,  145.        ,    0.        ,    2.        ],
       [ 146.        ,  618.        ,    0.        ,    3.        ],
       [ 147.        ,  619.        ,    0.        ,    4.        ],
       ...,
       [1227.        , 1229.        ,   23.40079289,  232.        ],
       [1230.        , 1231.        ,   28.55974612,  386.        ],
       [1232.        , 1233.        ,   52.70231   ,  618.        ]])
```

In [42]:
```
den = sch.dendrogram(Z)
plt.tick_params(
 axis='x',
 which='both',
 bottom=False,
 top=False,
 labelbottom=False)
plt.title('Hierarchical Clustering')
```

Out[42]:
```
Text(0.5, 1.0, 'Hierarchical Clustering')
```



In [43]:
```
from sklearn.cluster import AgglomerativeClustering
```

In [44]:
```
hc_model = AgglomerativeClustering(n_clusters = 2, affinity = 'euclidean', lin
```

In [45]:
```
y_cluster = hc_model.fit_predict(features_scaled)
```

In [46]:
```
y_cluster
```

Out[46]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0,
       0, 0], dtype=int64)
```

In [47]:
```
data_out = features_scaled.copy(deep = True)
data_out['Cluster'] = hc_model.labels_
data_out.head()
```

Out[47]:

	Id	TotalSteps	TotalDistance	TrackerDistance	SedentaryMinutes	Calories	Cluster
0	-1.304116	1.452212	1.175632	1.175632	-0.769412	-0.264036	0
1	-1.304116	0.862949	0.654844	0.654844	-0.609900	-0.549606	0
2	-1.304116	0.796180	0.576555	0.576555	0.858944	-0.581504	0
3	-1.304116	0.626709	0.419978	0.419978	-0.776059	-0.628593	0
4	-1.304116	1.332514	1.059901	1.059901	-0.619869	-0.449352	0

In [48]:

```
np.unique(hc_model.labels_, return_counts=True)
```

Out[48]:

```
(array([0, 1], dtype=int64), array([386, 232], dtype=int64))
```

In [49]:

```
silhouette_avg = silhouette_score(features_scaled, y_cluster)
print(f"Silhouette Score: {silhouette_avg}")
```

```
Silhouette Score: 0.31831904855358084
```

In [50]:

```
calinski_harabasz_index = calinski_harabasz_score(features_scaled, y_cluster)
print(f"Calinski-Harabasz Index: {calinski_harabasz_index}")
```

```
Calinski-Harabasz Index: 368.8634196782999
```

In [51]:

```
davies_bouldin_index = davies_bouldin_score(features_scaled, y_cluster)
print(f"Davies-Bouldin Index: {davies_bouldin_index}")
```

```
Davies-Bouldin Index: 1.1652064872757217
```

In [ ]:

```

```