AD_1_final

June 5, 2024

1 Activity Detection

Part 1

Data source: https://www.kaggle.com/datasets/luisomoreau/activity-detection

Our data consists of 12 folders, where each folder represents one activity. In each folder (except one), there are 12 CSV files with data. Each CSV file corresponds to one sensor that recorded the data. A description of the files with their values is provided below.

Acceleration (Accelerometer) - Accelerometer_z: Acceleration along the Z-axis. - Accelerometer_y: Acceleration along the Y-axis. - Accelerometer_x: Acceleration along the X-axis.

Annotation - empty

Gravity - Gravity_z: Gravity vector component along the Z-axis. - Gravity_y: Gravity vector component along the Y-axis. - Gravity_x: Gravity vector component along the X-axis.

Gyroscope - Gyroscope_z: Angular velocity around the Z-axis. - Gyroscope_y: Angular velocity around the Y-axis. - Gyroscope_x: Angular velocity around the X-axis.

Location - Location_bearingAccuracy: Bearing (azimuth) accuracy in location. - Location_speedAccuracy: Speed accuracy in location. - Location_verticalAccuracy: Altitude accuracy in location. - Location_horizontalAccuracy: Horizontal accuracy in location. - Location_speed: Speed in location. - Location_bearing: Bearing (azimuth) in location. - Location_altitude: Altitude in location. - Location_longitude: Longitude in location. - Location_latitude: Latitude in location.

Metadata - additional data

GPS (LocationGps) - LocationGps_bearingAccuracy: Bearing (azimuth) accuracy obtained from GPS. - LocationGps_speedAccuracy: Speed accuracy obtained from GPS. - LocationGps_verticalAccuracy: Altitude accuracy obtained from GPS. - LocationGps_horizontalAccuracy: Horizontal accuracy obtained from GPS. - LocationGps_speed: Speed obtained from GPS. - LocationGps_bearing: Bearing (azimuth) obtained from GPS. - LocationGps_altitude: Altitude obtained from GPS. - LocationGps_longitude: Longitude obtained from GPS. - LocationGps_latitude: Latitude obtained from GPS.

Network Location (LocationNetwork) - LocationNetwork_bearingAccuracy: Bearing (azimuth) accuracy obtained from the network. - LocationNetwork_speedAccuracy: Speed accuracy obtained from the network. - LocationNetwork_verticalAccuracy: Altitude accuracy obtained from the network. - LocationNetwork_horizontalAccuracy: Horizontal accuracy obtained from the network. - LocationNetwork_speed: Speed obtained from the network. - LocationNetwork_bearing: Bearing (azimuth) obtained from the network. - LocationNetwork_altitude: Altitude obtained from the network. - LocationNetwork_longitude: Longitude obtained from the network. - LocationNetwork_latitude: Latitude obtained from the network.

Magnetometer - Magnetometer_z: Magnetic field strength along the Z-axis. - Magnetometer_y: Magnetic field strength along the Y-axis. - Magnetometer_x: Magnetic field strength along

the X-axis.

Orientation - Orientation_qz: Z component of the quaternion representing orientation. - Orientation_qy: Y component of the quaternion representing orientation. - Orientation_qx: X component of the quaternion representing orientation. - Orientation_qw: W component of the quaternion representing orientation. - Orientation_roll: Roll angle of the orientation. - Orientation_pitch: Pitch angle of the orientation. - Orientation_yaw: Yaw angle of the orientation.

Pedometer - Pedometer_steps: Number of steps recorded by the pedometer.

Total Acceleration - TotalAcceleration_z: Total acceleration along the Z-axis. - TotalAcceleration_y: Total acceleration along the Y-axis. - TotalAcceleration_x: Total acceleration along the X-axis.

1.1 BUSINESS GOAL

We work for a company that makes devices for athletes (like sports watches) that track physical activities. Using sensors, they collect data such as speed and location from each activity separately. The user doesn't select the type of activity - the smart system just knows when they start doing something. This way, we get a bunch of activities with different data points. We want to cluster these activities to figure out what kinds of activities our users prefer and when they do them. This can be used for more personalized ads or for classification problems.

1.2 EDA

1.2.1 Imports

[ ]:
```
import pandas as pd
import numpy as np
import datetime
import os
import seaborn as sns
import matplotlib.pyplot as plt
import math
import random
```

1.2.2 Reading csv

Considering only one activity now because there are the same csv files everywhere. Now we want to get a preliminary overview of the dataset and its features. Annotation is empty csv (in all activities) so leaving it.

[ ]:
```
file_paths = [
    ('..//data//Cycling-2023-09-14_06-22-31//Accelerometer.csv',␣
↪'Accelerometer'),
    ('..//data//Cycling-2023-09-14_06-22-31//Gravity.csv', 'Gravity'),
    ('..//data//Cycling-2023-09-14_06-22-31//Gyroscope.csv', 'Gyroscope'),
    ('..//data//Cycling-2023-09-14_06-22-31//Location.csv', 'Location'),
    ('..//data//Cycling-2023-09-14_06-22-31//LocationGps.csv', 'LocationGps'),
    ('..//data//Cycling-2023-09-14_06-22-31//LocationNetwork.csv',␣
↪'LocationNetwork'),
```

```
    ('..//data//Cycling-2023-09-14_06-22-31//Magnetometer.csv', 'Magnetometer'),
    ('..//data//Cycling-2023-09-14_06-22-31//Metadata.csv', 'Metadata'),
    ('..//data//Cycling-2023-09-14_06-22-31//Orientation.csv', 'Orientation'),
    ('..//data//Cycling-2023-09-14_06-22-31//Pedometer.csv', 'Pedometer'),
    ('..//data//Cycling-2023-09-14_06-22-31//TotalAcceleration.csv',␣
 ↪'TotalAcceleration')
]
```

1.2.3 Preliminary overview

[ ]:
```
dataframes = {name: pd.read_csv(path) for path, name in file_paths}
```

[ ]:
```
for name,df in dataframes.items():
    print(name, df.shape)
```

```
Accelerometer (71461, 5)
Gravity (71461, 5)
Gyroscope (71847, 5)
Location (230, 11)
LocationGps (181, 11)
LocationNetwork (17, 11)
Magnetometer (9037, 5)
Metadata (1, 10)
Orientation (71461, 9)
Pedometer (32, 3)
TotalAcceleration (71456, 5)
```

[ ]:
```
def print_info(df,name):
        ␣
 ↪print("==========================================================================================
        print(f"{name} - below\n")
        ␣
 ↪print("==========================================================================================
        print(df.head())
        print("\n")
        print(df.info())
```

[ ]:
```
for name,df in dataframes.items():
    print_info(df,name)
```

```
==========================================================================================
====================


Accelerometer - below

==========================================================================================
```

```
====================


                  time  seconds_elapsed         z         y         x
0  1694672551573238300         0.121238  0.112874 -0.020792  0.156903
1  1694672551575757300         0.123757  0.235006 -0.003417  0.169683
2  1694672551578276400         0.126276  0.223140 -0.081017  0.135991
3  1694672551580795600         0.128796  0.191143 -0.067305  0.127650
4  1694672551583314400         0.131314  0.076245 -0.029743  0.083652


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71461 entries, 0 to 71460
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             71461 non-null  int64
 1   seconds_elapsed  71461 non-null  float64
 2   z                71461 non-null  float64
 3   y                71461 non-null  float64
 4   x                71461 non-null  float64
dtypes: float64(4), int64(1)
memory usage: 2.7 MB
None
================================================================================
====================


Gravity - below

================================================================================
====================


                  time  seconds_elapsed         z         y         x
0  1694672551573238300         0.121238  8.257127  5.262842  0.541047
1  1694672551575757300         0.123757  8.254994  5.266468  0.538318
2  1694672551578276400         0.126276  8.252911  5.269967  0.536009
3  1694672551580795600         0.128796  8.250857  5.273355  0.534300
4  1694672551583314400         0.131314  8.248755  5.276743  0.533298


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71461 entries, 0 to 71460
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             71461 non-null  int64
```

```
 1   seconds_elapsed  71461 non-null  float64
 2   z                71461 non-null  float64
 3   y                71461 non-null  float64
 4   x                71461 non-null  float64
dtypes: float64(4), int64(1)
memory usage: 2.7 MB
None
================================================================================
====================


Gyroscope - below

================================================================================
====================


                  time  seconds_elapsed         z         y         x
0  1694672551540493300         0.088493 -0.153038  0.011825  0.268263
1  1694672551570763500         0.118763 -0.122100  0.067237  0.173525
2  1694672551573238300         0.121238 -0.113575  0.065037  0.172425
3  1694672551575757300         0.123757 -0.108350  0.049087  0.170363
4  1694672551578276400         0.126276 -0.107250  0.024612  0.168162


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71847 entries, 0 to 71846
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             71847 non-null  int64
 1   seconds_elapsed  71847 non-null  float64
 2   z                71847 non-null  float64
 3   y                71847 non-null  float64
 4   x                71847 non-null  float64
dtypes: float64(4), int64(1)
memory usage: 2.7 MB
None
================================================================================
====================


Location - below

================================================================================
====================
```

```
                  time  seconds_elapsed  bearingAccuracy  speedAccuracy  \
0  1694672552897000000            1.445         0.000000            0.0
1  1694672553232000000            1.780         0.000000            0.0
2  1694672553838000000            2.386         0.000000            0.0
3  1694672554338000000            2.886        47.900002            2.6
4  1694672554838000000            3.386        50.700001            2.1

   verticalAccuracy  horizontalAccuracy     speed    bearing   altitude  \
0           1.00000              12.588  0.000000   0.000000  85.400002
1           1.00335              12.042  2.598672  26.936354  85.400002
2           1.00941              11.802  0.240886   8.161302  85.400002
3           1.01441              10.377  1.540603  38.479248  85.400002
4           1.01941               9.221  1.634604  41.726456  85.400002

   longitude   latitude
0   3.138636  50.682395
1   3.138626  50.682377
2   3.138632  50.682386
3   3.138629  50.682404
4   3.138631  50.682416


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 230 entries, 0 to 229
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   time                230 non-null    int64
 1   seconds_elapsed     230 non-null    float64
 2   bearingAccuracy     230 non-null    float64
 3   speedAccuracy       230 non-null    float64
 4   verticalAccuracy    230 non-null    float64
 5   horizontalAccuracy  230 non-null    float64
 6   speed               230 non-null    float64
 7   bearing             230 non-null    float64
 8   altitude            230 non-null    float64
 9   longitude           230 non-null    float64
 10  latitude            230 non-null    float64
dtypes: float64(10), int64(1)
memory usage: 19.9 KB
None
================================================================================
====================


LocationGps - below

================================================================================
```

```
====================


                  time  seconds_elapsed  bearingAccuracy  speedAccuracy  \
0  1694672553232000000            1.780        44.400002            3.8
1  1694672553838000000            2.386        47.500000            2.8
2  1694672554338000000            2.886        47.900002            2.6
3  1694672554838000000            3.386        50.700001            2.1
4  1694672555838000000            4.386        91.099998            3.1

   verticalAccuracy  horizontalAccuracy     speed    bearing  altitude  \
0        134.699997                 8.5  2.873694  28.480000      74.5
1        132.500000                 8.1  0.000000  28.480000      76.4
2        113.599998                 6.1  2.101611  40.630001      84.2
3        104.400002                 4.6  1.649170  43.939999      84.6
4        101.300003                 3.4  1.316145  37.709999      81.3

   longitude   latitude
0   3.138585  50.682308
1   3.138620  50.682360
2   3.138627  50.682412
3   3.138628  50.682420
4   3.138612  50.682423


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181 entries, 0 to 180
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   time                181 non-null    int64
 1   seconds_elapsed     181 non-null    float64
 2   bearingAccuracy     181 non-null    float64
 3   speedAccuracy       181 non-null    float64
 4   verticalAccuracy    181 non-null    float64
 5   horizontalAccuracy  181 non-null    float64
 6   speed               181 non-null    float64
 7   bearing             181 non-null    float64
 8   altitude            181 non-null    float64
 9   longitude           181 non-null    float64
 10  latitude            181 non-null    float64
dtypes: float64(10), int64(1)
memory usage: 15.7 KB
None
================================================================================
====================
```

LocationNetwork - below

```
================================================================================
====================


                  time  seconds_elapsed  bearingAccuracy  speedAccuracy  \
0  1694672552897000000            1.445                0              0
1  1694672555397000000            3.945                0              0
2  1694672560377000000            8.925                0              0
3  1694672563350000000           11.898                0              0
4  1694672584378000000           32.926                0              0

   verticalAccuracy  horizontalAccuracy  speed  bearing   altitude  longitude  \
0          1.000000           12.588000      0        0  85.400002   3.138636
1          1.000000           18.080000      0        0  85.400002   3.138624
2          1.000000           12.088000      0        0  85.300003   3.138647
3          1.000000           22.297001      0        0  85.099998   3.138437
4          1.147216           15.861000      0        0  84.200005   3.137801

    latitude
0  50.682395
1  50.682402
2  50.682497
3  50.682568
4  50.683183


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17 entries, 0 to 16
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   time                17 non-null     int64
 1   seconds_elapsed     17 non-null     float64
 2   bearingAccuracy     17 non-null     int64
 3   speedAccuracy       17 non-null     int64
 4   verticalAccuracy    17 non-null     float64
 5   horizontalAccuracy  17 non-null     float64
 6   speed               17 non-null     int64
 7   bearing             17 non-null     int64
 8   altitude            17 non-null     float64
 9   longitude           17 non-null     float64
 10  latitude            17 non-null     float64
dtypes: float64(6), int64(5)
memory usage: 1.6 KB
None
================================================================================
```

```
====================


Magnetometer - below

================================================================================
====================


                  time  seconds_elapsed          z        y          x
0  1694672551553178000         0.101178 -40.968750 -2.47500  25.275002
1  1694672551573182200         0.121182 -42.000000 -2.49375  25.143751
2  1694672551593195800         0.141196 -41.681252 -2.40000  25.068750
3  1694672551613221600         0.161222 -42.225002 -2.32500  24.843752
4  1694672551633251600         0.181251 -43.368752 -2.68125  24.825001


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9037 entries, 0 to 9036
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             9037 non-null   int64
 1   seconds_elapsed  9037 non-null   float64
 2   z                9037 non-null   float64
 3   y                9037 non-null   float64
 4   x                9037 non-null   float64
dtypes: float64(4), int64(1)
memory usage: 353.1 KB
None
================================================================================
====================


Metadata - below

================================================================================
====================


   version device name  recording epoch time       recording time  \
0        3     CPH2399         1694672551452  2023-09-14_06-22-31

  recording timezone platform appVersion  \
0       Europe/Paris  android     1.20.0

                             device id  \
0  c8dd5094-d4bd-47e1-b2c0-cc4205c1707a
```

```
                                                sensors           sampleRateMs
0  Accelerometer|Gravity|Gyroscope|Orientation|Ma...  0|0|0|0|0|0|10||0|0|0


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   version               1 non-null      int64
 1   device name           1 non-null      object
 2   recording epoch time  1 non-null      int64
 3   recording time        1 non-null      object
 4   recording timezone    1 non-null      object
 5   platform              1 non-null      object
 6   appVersion            1 non-null      object
 7   device id             1 non-null      object
 8   sensors               1 non-null      object
 9   sampleRateMs          1 non-null      object
dtypes: int64(2), object(8)
memory usage: 208.0+ bytes
None
================================================================================
====================


Orientation - below

================================================================================
====================


                  time  seconds_elapsed        qz        qy        qx  \
0  1694672551573238300         0.121238  0.440996  0.102955  0.261542
1  1694672551575757300         0.123757  0.440878  0.103150  0.261673
2  1694672551578276400         0.126276  0.440760  0.103322  0.261808
3  1694672551580795600         0.128796  0.440637  0.103461  0.261953
4  1694672551583314400         0.131314  0.440506  0.103565  0.262116

         qw      roll     pitch       yaw
0  0.852361 -0.065428 -0.566475 -0.973963
1  0.852358 -0.065115 -0.566913 -0.973670
2  0.852357 -0.064853 -0.567336 -0.973392
3  0.852359 -0.064663 -0.567746 -0.973121
4  0.852364 -0.064558 -0.568155 -0.972857
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71461 entries, 0 to 71460
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             71461 non-null  int64
 1   seconds_elapsed  71461 non-null  float64
 2   qz               71461 non-null  float64
 3   qy               71461 non-null  float64
 4   qx               71461 non-null  float64
 5   qw               71461 non-null  float64
 6   roll             71461 non-null  float64
 7   pitch            71461 non-null  float64
 8   yaw              71461 non-null  float64
dtypes: float64(8), int64(1)
memory usage: 4.9 MB
None
================================================================================
====================
```

Pedometer - below

```
================================================================================
====================
```

	time	seconds_elapsed	steps
0	1694672543371911400	-8.080089	0
1	1694672563032940000	11.580940	10
2	1694672564995320600	13.543321	12
3	1694672566959853600	15.507854	14
4	1694672568924632800	17.472633	18

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             32 non-null     int64
 1   seconds_elapsed  32 non-null     float64
 2   steps            32 non-null     int64
dtypes: float64(1), int64(2)
memory usage: 896.0 bytes
None
================================================================================
====================
```

```
TotalAcceleration - below

================================================================================
====================


                   time  seconds_elapsed         z        y        x
0  1694672551555605000         0.103605  8.287050  5.43105  0.45600
1  1694672551570763500         0.118763  8.311050  5.34000  0.62895
2  1694672551573238300         0.121238  8.370001  5.24205  0.69795
3  1694672551575757300         0.123757  8.490001  5.26305  0.70800
4  1694672551578276400         0.126276  8.476050  5.18895  0.67200


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71456 entries, 0 to 71455
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             71456 non-null  int64
 1   seconds_elapsed  71456 non-null  float64
 2   z                71456 non-null  float64
 3   y                71456 non-null  float64
 4   x                71456 non-null  float64
dtypes: float64(4), int64(1)
memory usage: 2.7 MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71456 entries, 0 to 71455
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             71456 non-null  int64
 1   seconds_elapsed  71456 non-null  float64
 2   z                71456 non-null  float64
 3   y                71456 non-null  float64
 4   x                71456 non-null  float64
dtypes: float64(4), int64(1)
memory usage: 2.7 MB
None
```

We decided to not consider Metadata in further analysis due to lack of the importance.

1.2.4 Merging CSV files within one activity

Now we can merge all csv files into one dataframe. We will use the time as the index (so we don't need seconds_elapsed column - we won't include it). We're performing a full join, simultaneously filling in missing values with the closest data available. As we said we will not include Metadata and Annotation.

[ ]:
```
# Path to the main directory
path = '..//data//'
os.listdir(path)
```

[ ]:
```
['Cycling-2023-09-14_06-22-31',
 'Cycling-2023-09-14_06-33-47',
 'Cycling-2023-09-14_06-47-00',
 'Cycling-2023-09-16_07-43-07',
 'Cycling-2023-09-16_09-25-09',
 'Cycling-2023-10-18_06-36-17',
 'Cycling-2023-10-18_06-51-26',
 'Sitting-2023-09-14_08-37-45',
 'Sitting-2023-09-14_09-11-15',
 'Sitting-2023-10-18_09-05-37',
 'Walking-2023-09-14_21-51-59',
 'Walking-2023-09-16_18-14-40']
```

1.2.5 Deleting unwanted csv files

[ ]:
```
# Listing all folders in the main directory
contents = [folder for folder in os.listdir(path) if os.path.isdir(os.path.
 ↪join(path, folder))]

for folder in contents:

    # Paths to the new folder where we'll save the cleaned data
    new_folder_path = '..//cleared_data//'

    full_new_folder_path = os.path.join(new_folder_path, folder)
    os.makedirs(full_new_folder_path, exist_ok=True)

    # Full path to the folder
    full_folder_path = os.path.join(path, folder)

    # Listing all files in the folder
    files = os.listdir(full_folder_path)

    for file in files:
        full_file_path = os.path.join(full_folder_path, file)

        # Checking if the file is not empty before loading
```

```
        if os.path.getsize(full_file_path) > 0:
            try:
                # Attempt to read the CSV file
                df = pd.read_csv(full_file_path)
                if 'time' in df.columns:
                    df['time'] = pd.to_datetime(df['time'])
                else:
                    continue
                print(f'File loaded: {full_file_path}')
                if 'seconds_elapsed' in df.columns:
                    df = df.drop(columns=['seconds_elapsed'])
                    df.to_csv(full_new_folder_path+'//'+file, index=False)
            except pd.errors.EmptyDataError:
                print(f'The file {full_file_path} is empty or has no columns to␣
↪parse.')
        else:
            print(f'The file {full_file_path} is empty.')
```

```
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Accelerometer.csv
The file ..//data//Cycling-2023-09-14_06-22-31\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Gravity.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Gyroscope.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Location.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\LocationGps.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\LocationNetwork.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Magnetometer.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Orientation.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\Pedometer.csv
File loaded: ..//data//Cycling-2023-09-14_06-22-31\TotalAcceleration.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Accelerometer.csv
The file ..//data//Cycling-2023-09-14_06-33-47\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Gravity.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Gyroscope.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Location.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\LocationGps.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\LocationNetwork.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Magnetometer.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Orientation.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\Pedometer.csv
File loaded: ..//data//Cycling-2023-09-14_06-33-47\TotalAcceleration.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Accelerometer.csv
The file ..//data//Cycling-2023-09-14_06-47-00\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Gravity.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Gyroscope.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Location.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\LocationGps.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\LocationNetwork.csv
```

```
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Magnetometer.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Orientation.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\Pedometer.csv
File loaded: ..//data//Cycling-2023-09-14_06-47-00\TotalAcceleration.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Accelerometer.csv
The file ..//data//Cycling-2023-09-16_07-43-07\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Gravity.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Gyroscope.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Location.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\LocationGps.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\LocationNetwork.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Magnetometer.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Orientation.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\Pedometer.csv
File loaded: ..//data//Cycling-2023-09-16_07-43-07\TotalAcceleration.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Accelerometer.csv
The file ..//data//Cycling-2023-09-16_09-25-09\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Gravity.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Gyroscope.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Location.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\LocationGps.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\LocationNetwork.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Magnetometer.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Orientation.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\Pedometer.csv
File loaded: ..//data//Cycling-2023-09-16_09-25-09\TotalAcceleration.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Accelerometer.csv
The file ..//data//Cycling-2023-10-18_06-36-17\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Gravity.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Gyroscope.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Location.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\LocationGps.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\LocationNetwork.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Magnetometer.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Orientation.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\Pedometer.csv
File loaded: ..//data//Cycling-2023-10-18_06-36-17\TotalAcceleration.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Accelerometer.csv
The file ..//data//Cycling-2023-10-18_06-51-26\Annotation.csv is empty.
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Gravity.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Gyroscope.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Location.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\LocationGps.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\LocationNetwork.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Magnetometer.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Orientation.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\Pedometer.csv
File loaded: ..//data//Cycling-2023-10-18_06-51-26\TotalAcceleration.csv
```

```
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Accelerometer.csv
The file ..//data//Sitting-2023-09-14_08-37-45\Annotation.csv is empty.
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Gravity.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Gyroscope.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Location.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\LocationGps.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\LocationNetwork.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Magnetometer.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Orientation.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\Pedometer.csv
File loaded: ..//data//Sitting-2023-09-14_08-37-45\TotalAcceleration.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Accelerometer.csv
The file ..//data//Sitting-2023-09-14_09-11-15\Annotation.csv is empty.
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Gravity.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Gyroscope.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Location.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\LocationGps.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\LocationNetwork.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Magnetometer.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Orientation.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\Pedometer.csv
File loaded: ..//data//Sitting-2023-09-14_09-11-15\TotalAcceleration.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Accelerometer.csv
The file ..//data//Sitting-2023-10-18_09-05-37\Annotation.csv is empty.
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Gravity.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Gyroscope.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Location.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Magnetometer.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Orientation.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\Pedometer.csv
File loaded: ..//data//Sitting-2023-10-18_09-05-37\TotalAcceleration.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\Accelerometer.csv
The file ..//data//Walking-2023-09-14_21-51-59\Annotation.csv is empty.
File loaded: ..//data//Walking-2023-09-14_21-51-59\Gravity.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\Gyroscope.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\Location.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\LocationGps.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\LocationNetwork.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\Magnetometer.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\Orientation.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\Pedometer.csv
File loaded: ..//data//Walking-2023-09-14_21-51-59\TotalAcceleration.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\Accelerometer.csv
The file ..//data//Walking-2023-09-16_18-14-40\Annotation.csv is empty.
File loaded: ..//data//Walking-2023-09-16_18-14-40\Gravity.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\Gyroscope.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\Location.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\LocationGps.csv
```

```
File loaded: ..//data//Walking-2023-09-16_18-14-40\LocationNetwork.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\Magnetometer.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\Orientation.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\Pedometer.csv
File loaded: ..//data//Walking-2023-09-16_18-14-40\TotalAcceleration.csv
```

1.2.6 Merging

[ ]:
```
folder_path = '../cleared_data'
folders = os.listdir(folder_path)
folders.sort()

for folder in folders:
    one_activity_path = os.path.join(folder_path, folder)
    print("Files in", folder + ":")
    print(os.listdir(one_activity_path))
    print()

i = 1
for dir in folders:
    activity = dir.split("-")[0]

    folder = os.path.join(folder_path, dir)

    file_list = [f for f in os.listdir(folder) if f.endswith('.csv')]

    result = pd.DataFrame({'time': []})



    for file_path in file_list:
        file = os.path.splitext(os.path.basename(file_path))[0]
        if file == "Annotation" or file == "Metadata":
            continue
        file_name = file + "_" + str(i)

        file_data = pd.read_csv(os.path.join(folder, file_path))

        file_data['time'] = pd.to_datetime(file_data['time'])

        # Rename columns except 'time'
        file_data = file_data.rename(columns={col: file + "_" + col if col !=␣
 ↪'time' else col for col in file_data.columns})

        if result.empty:
            result = file_data
        else:
```

```
            # Perform fuzzy join on 'time' column
            result = pd.merge_asof(result.sort_values('time'), file_data.
↪sort_values('time'), on='time', direction='nearest')


    result['time'] = pd.to_datetime(result['time'])
    result['time'] = result['time'].apply(lambda x: x.timestamp()*1000000)
    result.to_csv(f'../merged_data2/{activity}_{i}.csv', index=False)
    i += 1
```

```
Files in Cycling-2023-09-14_06-22-31:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Cycling-2023-09-14_06-33-47:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Cycling-2023-09-14_06-47-00:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Cycling-2023-09-16_07-43-07:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Cycling-2023-09-16_09-25-09:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Cycling-2023-10-18_06-36-17:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Cycling-2023-10-18_06-51-26:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Sitting-2023-09-14_08-37-45:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
```

```
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Sitting-2023-09-14_09-11-15:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Sitting-2023-10-18_09-05-37:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'Magnetometer.csv', 'Orientation.csv', 'Pedometer.csv', 'TotalAcceleration.csv']

Files in Walking-2023-09-14_21-51-59:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']

Files in Walking-2023-09-16_18-14-40:
['Accelerometer.csv', 'Gravity.csv', 'Gyroscope.csv', 'Location.csv',
'LocationGps.csv', 'LocationNetwork.csv', 'Magnetometer.csv', 'Orientation.csv',
'Pedometer.csv', 'TotalAcceleration.csv']
```

1.2.7 Reading files from the merged_data2 folder

[ ]:
```
cycling_1 = pd.read_csv('..//merged_data2//Cycling_1.csv')
cycling_1['id'] = 'cycling_1'
cycling_1['act_type'] = 0

cycling_2 = pd.read_csv('..//merged_data2//Cycling_2.csv')
cycling_2['id'] = 'cycling_2'
cycling_2['act_type'] = 0

cycling_3 = pd.read_csv('..//merged_data2//Cycling_3.csv')
cycling_3['id'] = 'cycling_3'
cycling_3['act_type'] = 0

cycling_4 = pd.read_csv('..//merged_data2//Cycling_4.csv')
cycling_4['id'] = 'cycling_4'
cycling_4['act_type'] = 0

cycling_5 = pd.read_csv('..//merged_data2//Cycling_5.csv')
cycling_5['id'] = 'cycling_5'
cycling_5['act_type'] = 0

cycling_6 = pd.read_csv('..//merged_data2//Cycling_6.csv')
cycling_6['id'] = 'cycling_6'
```

```
cycling_6['act_type'] = 0

cycling_7 = pd.read_csv('..//merged_data2//Cycling_7.csv')
cycling_7['id'] = 'cycling_7'
cycling_7['act_type'] = 0

sitting_8 = pd.read_csv('..//merged_data2//Sitting_8.csv')
sitting_8['id'] = 'sitting_8'
sitting_8['act_type'] = 1

sitting_9 = pd.read_csv('..//merged_data2//Sitting_9.csv')
sitting_9['id'] = 'sitting_9'
sitting_9['act_type'] = 1

sitting_10 = pd.read_csv('..//merged_data2//Sitting_10.csv')
sitting_10['id'] = 'sitting_10'
sitting_10['act_type'] = 1

walking_11 = pd.read_csv('..//merged_data2//Walking_11.csv')
walking_11['id'] = 'walking_11'
walking_11['act_type'] = 2

walking_12 = pd.read_csv('..//merged_data2//Walking_12.csv')
walking_12['id'] = 'walking_12'
walking_12['act_type'] = 2
```

We added actitivty_id, where 0 - cycling, 1 - sitting, 2 - walking. It is only needed to better visualisations by the end, we wont use this feature in models.

[ ]:
```
dataframes0 = [cycling_1.copy(), cycling_2.copy(), cycling_3.copy(), cycling_4.
↪copy(), cycling_5.copy(), cycling_6.copy(), cycling_7.copy(), sitting_8.
↪copy(), sitting_9.copy(), sitting_10.copy(), walking_11.copy(), walking_12.
↪copy()]
```

1.2.8 Splitting activities to groups

We split data to have more activities.

[ ]:
```
def split_and_add_unique_number(dataframes, number_of_rows):
    processed_dataframes = []
    k = 0
    for df in dataframes:
        df['activity_id'] = 0

        num_rows = df.shape[0]
        num_groups = num_rows // number_of_rows

        for i in range(num_groups):
            start_idx = i * number_of_rows
```

```
            end_idx = (i + 1) * number_of_rows

            df.iloc[start_idx:end_idx, df.columns.get_loc('activity_id')] = k
            k += 1
        df.iloc[num_groups * number_of_rows:, df.columns.
↪get_loc('activity_id')] = k
        k += 1

        processed_dataframes.append(df)

    return processed_dataframes
```

[ ]:
```
number_of_rows = 1000
dataframes = split_and_add_unique_number(dataframes0, number_of_rows)
```

[ ]:
```
dataframes0[0]
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
0      1.694673e+15         0.112874        -0.020792         0.156903
1      1.694673e+15         0.235006        -0.003417         0.169683
2      1.694673e+15         0.223140        -0.081017         0.135991
3      1.694673e+15         0.191143        -0.067305         0.127650
4      1.694673e+15         0.076245        -0.029743         0.083652
...             ...              ...              ...              ...
71456  1.694673e+15        -0.044467         0.185158        -0.116663
71457  1.694673e+15         0.009013         0.174825        -0.092028
71458  1.694673e+15         0.026442         0.186087        -0.049940
71459  1.694673e+15        -0.004457         0.151791        -0.026528
71460  1.694673e+15         0.009332         0.157430         0.006750

       Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  Gyroscope_x  \
0       8.257127   5.262842   0.541047    -0.113575     0.065037     0.172425
1       8.254994   5.266468   0.538318    -0.108350     0.049087     0.170363
2       8.252911   5.269967   0.536009    -0.107250     0.024612     0.168162
3       8.250857   5.273355   0.534300    -0.105050    -0.006187     0.165000
4       8.248755   5.276743   0.533298    -0.107250    -0.031762     0.162800
...          ...        ...        ...          ...          ...          ...
71456   8.723468   4.464842   0.369713    -0.076312    -0.037400    -0.179850
71457   8.722938   4.466176   0.366078    -0.074112    -0.075625    -0.196900
71458   8.722609   4.466913   0.364940    -0.072050    -0.116050    -0.213813
71459   8.722457   4.467160   0.365528    -0.065587    -0.152213    -0.229762
71460   8.722619   4.466620   0.368250    -0.063525    -0.186312    -0.240488

       ...  Orientation_roll  Orientation_pitch  Orientation_yaw  \
0      ...         -0.065428          -0.566475        -0.973963
1      ...         -0.065115          -0.566913        -0.973670
2      ...         -0.064853          -0.567336        -0.973392
3      ...         -0.064663          -0.567746        -0.973121
```

```
4      ...         -0.064558          -0.568155        -0.972857
...    ...               ...                ...              ...
71456  ...         -0.042769          -0.472684        -1.312718
71457  ...         -0.042348          -0.472837        -1.312315
71458  ...         -0.042214          -0.472923        -1.317852
71459  ...         -0.042274          -0.472951        -1.317581
71460  ...         -0.042579          -0.472889        -1.317389

       Pedometer_steps  TotalAcceleration_z  TotalAcceleration_y  \
0                    0             8.370001              5.24205
1                    0             8.490001              5.26305
2                    0             8.476050              5.18895
3                    0             8.442000              5.20605
4                    0             8.325001              5.24700
...                ...                  ...                  ...
71456              149             8.799001              4.60695
71457              149             8.799001              4.60695
71458              149             8.799001              4.60695
71459              149             8.799001              4.60695
71460              149             8.799001              4.60695

       TotalAcceleration_x         id  act_type  activity_id
0                  0.69795  cycling_1         0            0
1                  0.70800  cycling_1         0            0
2                  0.67200  cycling_1         0            0
3                  0.66195  cycling_1         0            0
4                  0.61695  cycling_1         0            0
...                    ...        ...       ...          ...
71456              0.21900  cycling_1         0           71
71457              0.21900  cycling_1         0           71
71458              0.21900  cycling_1         0           71
71459              0.21900  cycling_1         0           71
71460              0.21900  cycling_1         0           71

[71461 rows x 54 columns]
```

[ ]:
```
dataframes[1]
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
0       1.694673e+15         0.015474        -0.226825        -0.091155
1       1.694673e+15         0.180070        -0.271099        -0.089934
2       1.694673e+15         0.365452        -0.311201        -0.138382
3       1.694673e+15         0.483836        -0.410178        -0.127567
4       1.694673e+15         0.559147        -0.474245        -0.156521
...              ...              ...              ...              ...
170711  1.694674e+15         0.180080         0.083863         0.135739
170712  1.694674e+15         0.178696         0.075105         0.146896
170713  1.694674e+15         0.114075         0.109703         0.140638
```

```
170714  1.694674e+15         0.140795         0.121037         0.136420
170715  1.694674e+15         0.145888         0.082986         0.113767

        Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  \
0        8.256577   5.056825   1.558155    -0.321200    -0.138737
1        8.252931   5.063149   1.556934    -0.345675    -0.140937
2        8.249498   5.069201   1.555432    -0.369050    -0.140937
3        8.246164   5.075178   1.553617    -0.388300    -0.139838
4        8.242853   5.081195   1.551521    -0.403150    -0.131313
...           ...        ...        ...          ...          ...
170711   8.939920   3.740088   1.503311     0.034925     0.067100
170712   8.939355   3.741946   1.502054     0.037125     0.080850
170713   8.938876   3.743347   1.501412     0.036025     0.089375
170714   8.938255   3.745964   1.498580     0.039187     0.099000
170715   8.938112   3.746064   1.499183     0.039187     0.099000

        Gyroscope_x  ...  Orientation_roll  Orientation_pitch  \
0          0.244475  ...         -0.186520          -0.541770
1          0.227425  ...         -0.186458          -0.542523
2          0.221100  ...         -0.186358          -0.543244
3          0.221100  ...         -0.186219          -0.543956
4          0.226325  ...         -0.186047          -0.544673
...             ...  ...               ...                ...
170711     0.047575  ...         -0.167055          -0.391919
170712     0.052938  ...         -0.166932          -0.392128
170713     0.056100  ...         -0.166871          -0.392282
170714     0.057200  ...         -0.166774          -0.392444
170715     0.057200  ...         -0.166642          -0.392581

        Orientation_yaw  Pedometer_steps  TotalAcceleration_z  \
0             -1.393478                1             8.272051
1             -1.392684                1             8.433001
2             -1.391809                1             8.614950
3             -1.390862                1             8.730000
4             -1.389858                1             8.802000
...                 ...              ...                  ...
170711         1.028005              482             9.120001
170712         1.027724              482             9.118051
170713         1.027557              482             9.052951
170714         1.027396              482             9.079050
170715         1.027232              482             9.079050

        TotalAcceleration_y  TotalAcceleration_x         id  act_type  \
0                   4.83000              1.46700  cycling_2         0
1                   4.79205              1.46700  cycling_2         0
2                   4.75800              1.41705  cycling_2         0
3                   4.66500              1.42605  cycling_2         0
```

```
4                   4.60695              1.39500  cycling_2         0
...                     ...                  ...        ...       ...
170711              3.82395              1.63905  cycling_2         0
170712              3.81705              1.64895  cycling_2         0
170713              3.85305              1.64205  cycling_2         0
170714              3.86700              1.63500  cycling_2         0
170715              3.86700              1.63500  cycling_2         0

        activity_id
0                72
1                72
2                72
3                72
4                72
...             ...
170711          242
170712          242
170713          242
170714          242
170715          242

[170716 rows x 54 columns]
```

Now let's merge it all into one big csv file to perform EDA.

[ ]:
```
result0 = pd.concat([
    (dataframes[0]),
    (dataframes[1]),
    (dataframes[2]),
    (dataframes[3]),
    (dataframes[4]),
    (dataframes[5]),
    (dataframes[6]),
    (dataframes[7]),
    (dataframes[8]),
    (dataframes[9]),
    (dataframes[10]),
    (dataframes[11])
], ignore_index=True)
```

[ ]:
```
result0
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
0       1.694673e+15         0.112874        -0.020792         0.156903
1       1.694673e+15         0.235006        -0.003417         0.169683
2       1.694673e+15         0.223140        -0.081017         0.135991
3       1.694673e+15         0.191143        -0.067305         0.127650
4       1.694673e+15         0.076245        -0.029743         0.083652
...              ...              ...              ...              ...
```

```
2775506  1.694889e+15        -0.857878        -0.025720        -0.367761
2775507  1.694889e+15        -0.818873         0.034961        -0.285863
2775508  1.694889e+15        -0.737333         0.055158        -0.267903
2775509  1.694889e+15        -0.712534         0.047186        -0.249589
2775510  1.694889e+15        -0.635440         0.094296        -0.177881

         Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  \
0         8.257127   5.262842   0.541047    -0.113575     0.065037
1         8.254994   5.266468   0.538318    -0.108350     0.049087
2         8.252911   5.269967   0.536009    -0.107250     0.024612
3         8.250857   5.273355   0.534300    -0.105050    -0.006187
4         8.248755   5.276743   0.533298    -0.107250    -0.031762
...            ...        ...        ...          ...          ...
2775506   7.549828   6.081670   1.477761     0.015812    -0.002887
2775507   7.546823   6.086089   1.474913     0.015812    -0.002887
2775508   7.544333   6.089892   1.471953     0.015812    -0.002887
2775509   7.542484   6.092764   1.469539     0.015812    -0.002887
2775510   7.540390   6.095755   1.467881     0.015812    -0.002887

         Gyroscope_x  ...  Orientation_roll  Orientation_pitch  \
0           0.172425  ...         -0.065428          -0.566475
1           0.170363  ...         -0.065115          -0.566913
2           0.168162  ...         -0.064853          -0.567336
3           0.165000  ...         -0.064663          -0.567746
4           0.162800  ...         -0.064558          -0.568155
...              ...  ...               ...                ...
2775506     0.314463  ...         -0.195289          -0.669651
2775507     0.314463  ...         -0.195010          -0.670229
2775508     0.314463  ...         -0.194698          -0.670728
2775509     0.314463  ...         -0.194441          -0.671103
2775510     0.314463  ...         -0.194281          -0.671491

         Orientation_yaw  Pedometer_steps  TotalAcceleration_z  \
0              -0.973963                0             8.370001
1              -0.973670                0             8.490001
2              -0.973392                0             8.476050
3              -0.973121                0             8.442000
4              -0.972857                0             8.325001
...                  ...              ...                  ...
2775506         1.989292             1489             6.691950
2775507         1.988996             1489             6.727950
2775508         1.985080             1489             6.807000
2775509         1.984742             1489             6.829950
2775510         1.984317             1489             6.829950

         TotalAcceleration_y  TotalAcceleration_x          id  act_type  \
0                   5.242050              0.69795   cycling_1         0
```

```
1                   5.263050              0.70800   cycling_1         0
2                   5.188950              0.67200   cycling_1         0
3                   5.206050              0.66195   cycling_1         0
4                   5.247000              0.61695   cycling_1         0
...                      ...                   ...         ...       ...
2775506             6.055950              1.11000  walking_12         2
2775507             6.121050              1.18905  walking_12         2
2775508             6.145051              1.20405  walking_12         2
2775509             6.139950              1.21995  walking_12         2
2775510             6.139950              1.21995  walking_12         2

         activity_id
0                  0
1                  0
2                  0
3                  0
4                  0
...              ...
2775506         2780
2775507         2780
2775508         2780
2775509         2780
2775510         2780

[2775511 rows x 54 columns]
```

Saving it to csv file.

[ ]:
```
result0.to_csv("..//final_data//result0_data.csv", index=False)
```

1.3 Splitting into train, test and valid

[ ]:
```
number_of_act = result0['activity_id'].nunique()
number_of_act
```

[ ]:
```
2781
```

[ ]:
```
num_samples = int(0.7 * number_of_act)
random_values = random.sample(range(number_of_act), num_samples)

train_valid = result0[result0['activity_id'].isin(random_values)]
test = result0[~result0['activity_id'].isin(random_values)]
```

[ ]:
```
test
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
1000     1.694673e+15        -6.628151         8.400816        12.724605
1001     1.694673e+15        -7.099690         7.518416        11.707123
1002     1.694673e+15        -7.400462         6.935641        10.715312
1003     1.694673e+15        -7.182696         6.686847         9.732042
```

```
1004     1.694673e+15        -6.843181         6.644148         8.381004
...               ...              ...              ...              ...
2771164  1.694889e+15         0.098736         0.027143         0.039114
2771165  1.694889e+15         0.102740         0.042348         0.030427
2771166  1.694889e+15         0.156071         0.004308         0.007788
2771167  1.694889e+15         0.120071         0.018258         0.019788
2771168  1.694889e+15         0.100666         0.015337        -0.011892

         Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  \
1000      0.162101  -9.645816   1.761346     1.974362     0.122512
1001      0.150639  -9.642416   1.780878     1.839200    -0.188238
1002      0.130411  -9.646591   1.759739     1.718887    -0.504350
1003      0.101646  -9.648897   1.748958     1.613562    -0.804375
1004      0.042181  -9.654198   1.721946     1.526250    -1.057650
...            ...        ...        ...          ...          ...
2771164  -8.892786   2.481907   3.305886     0.014712    -0.005087
2771165  -8.891690   2.483652   3.307523     0.016912    -0.002887
2771166  -8.890121   2.485692   3.310212     0.014712    -0.003987
2771167  -8.890121   2.485692   3.310212     0.012650    -0.003987
2771168  -8.889616   2.486663   3.310843     0.013750     0.000275

         Gyroscope_x  ...  Orientation_roll  Orientation_pitch  \
1000       -0.169125  ...         -1.475532           1.385582
1001       -0.211750  ...         -1.482804           1.388674
1002       -0.267025  ...         -1.493189           1.390990
1003       -0.318175  ...         -1.509104           1.392257
1004       -0.340450  ...         -1.535874           1.391369
...              ...  ...               ...                ...
2771164    -0.006875  ...         -2.787151          -0.255470
2771165    -0.006875  ...         -2.786953          -0.255655
2771166    -0.006875  ...         -2.786595          -0.255876
2771167    -0.005912  ...         -2.786595          -0.255876
2771168    -0.003712  ...         -2.786561          -0.255976

         Orientation_yaw  Pedometer_steps  TotalAcceleration_z  \
1000           -0.924110               10            -6.466050
1001           -0.908072               10            -6.949050
1002           -0.898299               10            -7.270051
1003           -0.883821               10            -7.081050
1004           -0.859356               10            -6.801000
...                  ...              ...                  ...
2771164         2.805723             1489            -8.794050
2771165         2.805817             1489            -8.788950
2771166         2.806245             1489            -8.734051
2771167         2.806245             1489            -8.770050
2771168         2.806337             1489            -8.788950
```

```
         TotalAcceleration_y  TotalAcceleration_x          id  act_type  \
1000                -1.24500            14.485950   cycling_1         0
1001                -2.12400            13.488001   cycling_1         0
1002                -2.71095            12.475051   cycling_1         0
1003                -2.96205            11.481001   cycling_1         0
1004                -3.01005            10.102950   cycling_1         0
...                      ...                  ...         ...       ...
2771164              2.50905             3.345000  walking_12         2
2771165              2.52600             3.337950  walking_12         2
2771166              2.49000             3.318000  walking_12         2
2771167              2.50395             3.330000  walking_12         2
2771168              2.50200             3.298950  walking_12         2

         activity_id
1000               1
1001               1
1002               1
1003               1
1004               1
...              ...
2771164         2775
2771165         2775
2771166         2775
2771167         2775
2771168         2775

[833535 rows x 54 columns]
```

1.3.1 Saving test csv

[ ]:
```
X_test = test.drop(columns=['act_type'])
Y_test = test[['act_type', 'activity_id']]
Y_test = Y_test.groupby('activity_id')['act_type'].apply(lambda x: x.iloc[0]).
 ↪reset_index()



X_test.to_csv("..//final_data//X_test.csv", index=False)
Y_test.to_csv("..//final_data//Y_test.csv", index=False)
```

[ ]:
```
Y_test
```

[ ]:
```
     activity_id  act_type
0              1         0
1              2         0
2              5         0
3              6         0
```

```
4             14         0
..           ...       ...
830         2769         2
831         2770         2
832         2771         2
833         2773         2
834         2775         2

[835 rows x 2 columns]
```

[ ]:
```
X_test
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
1000     1.694673e+15        -6.628151         8.400816        12.724605
1001     1.694673e+15        -7.099690         7.518416        11.707123
1002     1.694673e+15        -7.400462         6.935641        10.715312
1003     1.694673e+15        -7.182696         6.686847         9.732042
1004     1.694673e+15        -6.843181         6.644148         8.381004
...               ...              ...              ...              ...
2771164  1.694889e+15         0.098736         0.027143         0.039114
2771165  1.694889e+15         0.102740         0.042348         0.030427
2771166  1.694889e+15         0.156071         0.004308         0.007788
2771167  1.694889e+15         0.120071         0.018258         0.019788
2771168  1.694889e+15         0.100666         0.015337        -0.011892

         Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  \
1000      0.162101  -9.645816   1.761346     1.974362     0.122512
1001      0.150639  -9.642416   1.780878     1.839200    -0.188238
1002      0.130411  -9.646591   1.759739     1.718887    -0.504350
1003      0.101646  -9.648897   1.748958     1.613562    -0.804375
1004      0.042181  -9.654198   1.721946     1.526250    -1.057650
...            ...        ...        ...          ...          ...
2771164  -8.892786   2.481907   3.305886     0.014712    -0.005087
2771165  -8.891690   2.483652   3.307523     0.016912    -0.002887
2771166  -8.890121   2.485692   3.310212     0.014712    -0.003987
2771167  -8.890121   2.485692   3.310212     0.012650    -0.003987
2771168  -8.889616   2.486663   3.310843     0.013750     0.000275

         Gyroscope_x  ...  Orientation_qw  Orientation_roll  \
1000       -0.169125  ...        0.318166         -1.475532
1001       -0.211750  ...        0.319820         -1.482804
1002       -0.267025  ...        0.318680         -1.493189
1003       -0.318175  ...        0.317510         -1.509104
1004       -0.340450  ...        0.316523         -1.535874
...              ...  ...             ...               ...
2771164    -0.006875  ...        0.094402         -2.787151
2771165    -0.006875  ...        0.094482         -2.786953
2771166    -0.006875  ...        0.094597         -2.786595
```

```
2771167    -0.005912  ...        0.094597         -2.786595
2771168    -0.003712  ...        0.094651         -2.786561

         Orientation_pitch  Orientation_yaw  Pedometer_steps  \
1000              1.385582        -0.924110               10
1001              1.388674        -0.908072               10
1002              1.390990        -0.898299               10
1003              1.392257        -0.883821               10
1004              1.391369        -0.859356               10
...                    ...              ...              ...
2771164          -0.255470         2.805723             1489
2771165          -0.255655         2.805817             1489
2771166          -0.255876         2.806245             1489
2771167          -0.255876         2.806245             1489
2771168          -0.255976         2.806337             1489

         TotalAcceleration_z  TotalAcceleration_y  TotalAcceleration_x  \
1000               -6.466050             -1.24500            14.485950
1001               -6.949050             -2.12400            13.488001
1002               -7.270051             -2.71095            12.475051
1003               -7.081050             -2.96205            11.481001
1004               -6.801000             -3.01005            10.102950
...                      ...                  ...                  ...
2771164            -8.794050              2.50905             3.345000
2771165            -8.788950              2.52600             3.337950
2771166            -8.734051              2.49000             3.318000
2771167            -8.770050              2.50395             3.330000
2771168            -8.788950              2.50200             3.298950

                id  activity_id
1000      cycling_1            1
1001      cycling_1            1
1002      cycling_1            1
1003      cycling_1            1
1004      cycling_1            1
...             ...          ...
2771164  walking_12         2775
2771165  walking_12         2775
2771166  walking_12         2775
2771167  walking_12         2775
2771168  walking_12         2775

[833535 rows x 53 columns]
```

1.3.2 Making train and valid csv

[ ]:
```
number_of_act_train_valid = train_valid['activity_id'].nunique()
number_of_act_train_valid
```

[ ]:
```
1946
```

[ ]:
```
num_samples = int(0.7 * number_of_act_train_valid)
random_values_train_valid = random.sample(random_values, num_samples)

train = train_valid[train_valid['activity_id'].isin(random_values_train_valid)]
valid = train_valid[~train_valid['activity_id'].isin(random_values_train_valid)]
```

1.3.3 Saving train and valid csv

[ ]:
```
X_train = train.drop(columns=['act_type'])
Y_train = train[['act_type', 'activity_id']]
Y_train = Y_train.groupby('activity_id')['act_type'].apply(lambda x: x.iloc[0]).
 ↪reset_index()

X_valid = valid.drop(columns=['act_type'])
Y_valid = valid[['act_type', 'activity_id']]
Y_valid = Y_valid.groupby('activity_id')['act_type'].apply(lambda x: x.iloc[0]).
 ↪reset_index()

X_train.to_csv("..//final_data//X_train.csv", index=False)
Y_train.to_csv("..//final_data//Y_train.csv", index=False)
X_valid.to_csv("..//final_data//X_valid.csv", index=False)
Y_valid.to_csv("..//final_data//Y_valid.csv", index=False)
```

[ ]:
```
X_train
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
7000     1.694673e+15         1.291255         1.599357        -1.142604
7001     1.694673e+15         1.234206         1.576475        -1.107837
7002     1.694673e+15         1.241680         1.503248        -1.117860
7003     1.694673e+15         1.231439         1.371741        -0.999492
7004     1.694673e+15         1.269879         1.208365        -0.961690
...               ...              ...              ...              ...
2775506  1.694889e+15        -0.857878        -0.025720        -0.367761
2775507  1.694889e+15        -0.818873         0.034961        -0.285863
2775508  1.694889e+15        -0.737333         0.055158        -0.267903
2775509  1.694889e+15        -0.712534         0.047186        -0.249589
2775510  1.694889e+15        -0.635440         0.094296        -0.177881

         Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  \
7000     -6.871256  -6.104307   3.419604    -0.468050    -0.020075
7001     -6.893256  -6.076475   3.424887    -0.447700    -0.012650
7002     -6.914680  -6.049298   3.429810    -0.422262    -0.010450
```

```
7003     -6.935489  -6.022791   3.434442    -0.402050    -0.009488
7004     -6.973929  -5.972365   3.444640    -0.381837     0.000137
...            ...        ...        ...          ...          ...
2775506   7.549828   6.081670   1.477761     0.015812    -0.002887
2775507   7.546823   6.086089   1.474913     0.015812    -0.002887
2775508   7.544333   6.089892   1.471953     0.015812    -0.002887
2775509   7.542484   6.092764   1.469539     0.015812    -0.002887
2775510   7.540390   6.095755   1.467881     0.015812    -0.002887

         Gyroscope_x  ...  Orientation_qw  Orientation_roll  \
7000       -1.317387  ...        0.315518         -2.679851
7001       -1.295112  ...        0.314863         -2.680510
7002       -1.269537  ...        0.314218         -2.681173
7003       -1.249325  ...        0.313663         -2.681832
7004       -1.230212  ...        0.313060         -2.682458
...              ...  ...             ...               ...
2775506     0.314463  ...        0.485231         -0.195289
2775507     0.314463  ...        0.485322         -0.195010
2775508     0.314463  ...        0.486885         -0.194698
2775509     0.314463  ...        0.487014         -0.194441
2775510     0.314463  ...        0.487162         -0.194281

         Orientation_pitch  Orientation_yaw  Pedometer_steps  \
7000              0.671913         0.722495               18
7001              0.668292         0.723219               18
7002              0.664765         0.723970               18
7003              0.661336         0.725430               18
7004              0.658039         0.726194               18
...                    ...              ...              ...
2775506          -0.669651         1.989292             1489
2775507          -0.670229         1.988996             1489
2775508          -0.670728         1.985080             1489
2775509          -0.671103         1.984742             1489
2775510          -0.671491         1.984317             1489

         TotalAcceleration_z  TotalAcceleration_y  TotalAcceleration_x  \
7000                -5.58000            -4.504950              2.27700
7001                -5.65905            -4.500000              2.31705
7002                -5.67300            -4.546050              2.31195
7003                -5.70405            -4.651050              2.43495
7004                -5.70405            -4.764000              2.48295
...                      ...                  ...                  ...
2775506              6.69195             6.055950              1.11000
2775507              6.72795             6.121050              1.18905
2775508              6.80700             6.145051              1.20405
2775509              6.82995             6.139950              1.21995
2775510              6.82995             6.139950              1.21995
```

```
                id  activity_id
7000      cycling_1            7
7001      cycling_1            7
7002      cycling_1            7
7003      cycling_1            7
7004      cycling_1            7
...             ...          ...
2775506  walking_12         2780
2775507  walking_12         2780
2775508  walking_12         2780
2775509  walking_12         2780
2775510  walking_12         2780

[1358801 rows x 53 columns]
```

[ ]:
```
Y_train
```

[ ]:
```
      activity_id  act_type
0               7         0
1               8         0
2               9         0
3              12         0
4              13         0
...           ...       ...
1357         2774         2
1358         2776         2
1359         2777         2
1360         2778         2
1361         2780         2

[1362 rows x 2 columns]
```

[ ]:
```
X_valid
```

[ ]:
```
                time  Accelerometer_z  Accelerometer_y  Accelerometer_x  \
0       1.694673e+15         0.112874        -0.020792         0.156903
1       1.694673e+15         0.235006        -0.003417         0.169683
2       1.694673e+15         0.223140        -0.081017         0.135991
3       1.694673e+15         0.191143        -0.067305         0.127650
4       1.694673e+15         0.076245        -0.029743         0.083652
...              ...              ...              ...              ...
2775164 1.694889e+15         3.666044        -0.579963        -1.099331
2775165 1.694889e+15         3.765838        -0.516886        -1.075790
2775166 1.694889e+15         3.853888        -0.355936        -0.946790
2775167 1.694889e+15         3.904089        -0.158427        -0.783069
2775168 1.694889e+15         3.954189        -0.019377        -0.675069

         Gravity_z  Gravity_y  Gravity_x  Gyroscope_z  Gyroscope_y  \
```

```
0         8.257127   5.262842   0.541047    -0.113575     0.065037
1         8.254994   5.266468   0.538318    -0.108350     0.049087
2         8.252911   5.269967   0.536009    -0.107250     0.024612
3         8.250857   5.273355   0.534300    -0.105050    -0.006187
4         8.248755   5.276743   0.533298    -0.107250    -0.031762
...            ...        ...        ...          ...          ...
2775164   9.448906   2.588913   0.431381     0.039187    -2.297350
2775165   9.421113   2.656936   0.594740     0.038225    -2.398412
2775166   9.421113   2.656936   0.594740     0.032862    -2.482425
2775167   9.423862   2.648427   0.589119     0.024338    -2.552687
2775168   9.423862   2.648427   0.589119     0.006325    -2.611263

         Gyroscope_x  ...  Orientation_qw  Orientation_roll  \
0           0.172425  ...        0.852361         -0.065428
1           0.170363  ...        0.852358         -0.065115
2           0.168162  ...        0.852357         -0.064853
3           0.165000  ...        0.852359         -0.064663
4           0.162800  ...        0.852364         -0.064558
...              ...  ...             ...               ...
2775164     0.997700  ...        0.354196         -0.047979
2775165     1.099862  ...        0.353809         -0.052973
2775166     1.197762  ...        0.353809         -0.052973
2775167     1.293600  ...        0.350840         -0.064849
2775168     1.384075  ...        0.350840         -0.064849

         Orientation_pitch  Orientation_yaw  Pedometer_steps  \
0                -0.566475        -0.973963                0
1                -0.566913        -0.973670                0
2                -0.567336        -0.973392                0
3                -0.567746        -0.973121                0
4                -0.568155        -0.972857                0
...                    ...              ...              ...
2775164          -0.269527         2.403827             1489
2775165          -0.270938         2.403826             1489
2775166          -0.270938         2.403826             1489
2775167          -0.275802         2.408078             1489
2775168          -0.275802         2.408078             1489

         TotalAcceleration_z  TotalAcceleration_y  TotalAcceleration_x  \
0                   8.370001              5.24205              0.69795
1                   8.490001              5.26305              0.70800
2                   8.476050              5.18895              0.67200
3                   8.442000              5.20605              0.66195
4                   8.325001              5.24700              0.61695
...                      ...                  ...                  ...
2775164            13.114950              2.00895             -0.66795
2775165            13.186951              2.14005             -0.48105
```

```
2775166            13.275001             2.30100            -0.35205
2775167            13.327950             2.49000            -0.19395
2775168            13.378051             2.62905            -0.08595

                id  activity_id
0         cycling_1            0
1         cycling_1            0
2         cycling_1            0
3         cycling_1            0
4         cycling_1            0
...             ...          ...
2775164  walking_12         2779
2775165  walking_12         2779
2775166  walking_12         2779
2775167  walking_12         2779
2775168  walking_12         2779

[583175 rows x 53 columns]
```

[ ]:
```
Y_valid
```

[ ]:
```
     activity_id  act_type
0              0         0
1              3         0
2              4         0
3             10         0
4             11         0
..           ...       ...
579         2740         2
580         2748         2
581         2751         2
582         2767         2
583         2779         2

[584 rows x 2 columns]
```

1.4 Advanced EDA

1.4.1 Reading data

[ ]:
```
X_train = pd.read_csv("..//final_data//X_train.csv")
Y_train = pd.read_csv("..//final_data//Y_train.csv")

# X_train = pd.read_csv("..//final_data//X_test.csv") # for validation
# Y_train = pd.read_csv("..//final_data//Y_test.csv") # for validation

# X_train = pd.read_csv("..//final_data//X_valid.csv") # for validation
# Y_train = pd.read_csv("..//final_data//Y_valid.csv") # for validation
```

1.4.2 Looking for unnecessary features

[ ]:
```
X_train.columns
```

[ ]:
```
Index(['time', 'Accelerometer_z', 'Accelerometer_y', 'Accelerometer_x',
       'Gravity_z', 'Gravity_y', 'Gravity_x', 'Gyroscope_z', 'Gyroscope_y',
       'Gyroscope_x', 'Location_bearingAccuracy', 'Location_speedAccuracy',
       'Location_verticalAccuracy', 'Location_horizontalAccuracy',
       'Location_speed', 'Location_bearing', 'Location_altitude',
       'Location_longitude', 'Location_latitude',
       'LocationGps_bearingAccuracy', 'LocationGps_speedAccuracy',
       'LocationGps_verticalAccuracy', 'LocationGps_horizontalAccuracy',
       'LocationGps_speed', 'LocationGps_bearing', 'LocationGps_altitude',
       'LocationGps_longitude', 'LocationGps_latitude',
       'LocationNetwork_bearingAccuracy', 'LocationNetwork_speedAccuracy',
       'LocationNetwork_verticalAccuracy',
       'LocationNetwork_horizontalAccuracy', 'LocationNetwork_speed',
       'LocationNetwork_bearing', 'LocationNetwork_altitude',
       'LocationNetwork_longitude', 'LocationNetwork_latitude',
       'Magnetometer_z', 'Magnetometer_y', 'Magnetometer_x', 'Orientation_qz',
       'Orientation_qy', 'Orientation_qx', 'Orientation_qw',
       'Orientation_roll', 'Orientation_pitch', 'Orientation_yaw',
       'Pedometer_steps', 'TotalAcceleration_z', 'TotalAcceleration_y',
       'TotalAcceleration_x', 'id', 'activity_id'],
      dtype='object')
```

Accelerometer - can be important in movement analysis
Annotation - deleted
Gravity - skipped - unnsecessary (same value about 9.81)
Gyroscope - can be important same as accelerometer
Location, LocationGPS, LocationNetwork - at this moment it seems to be important
Magnetometer - can be important
Metadata - deleted
Orientation - we drop columns Orientation_qx, Orientation_qy, Orientation_qz, Orientation_qw, because they contain the same information as Orientation_roll, Orientation_pitch, Orientation_yaw
Pedometer - number of steps, can be important
TotalAcceleration - (Accelerometer + Gravity =?= TotalAcceleration) (HYPOTHESIS) - can be important

[ ]:
```
df = X_train.copy()
tmp = pd.DataFrame({
    'x': abs(df['Accelerometer_x'] + df['Gravity_x'] -␣
 ↪df['TotalAcceleration_x']),
    'y': abs(df['Accelerometer_y'] + df['Gravity_y'] -␣
 ↪df['TotalAcceleration_y']),
    'z': abs(df['Accelerometer_z'] + df['Gravity_z'] -␣
 ↪df['TotalAcceleration_z'])
}).dropna()
```

```
print(tmp.describe())
```

```
                  x             y             z
count  1.358801e+06  1.358801e+06  1.358801e+06
mean   5.002818e-07  4.184985e-07  1.404236e-06
std    2.101887e-04  1.378169e-04  5.639351e-04
min    0.000000e+00  0.000000e+00  0.000000e+00
25%    0.000000e+00  0.000000e+00  0.000000e+00
50%    0.000000e+00  0.000000e+00  0.000000e+00
75%    8.881784e-16  1.110223e-16  2.220446e-16
max    1.560000e-01  9.495020e-02  4.570503e-01
```

Values close to 0, so we think we can leave only TotalAcceleration in our data frame.

1.4.3 Histograms

[ ]:
```
df.hist(figsize = (20, 20))
```

[ ]:
```
array([[<Axes: title={'center': 'time'}>,
        <Axes: title={'center': 'Accelerometer_z'}>,
        <Axes: title={'center': 'Accelerometer_y'}>,
        <Axes: title={'center': 'Accelerometer_x'}>,
        <Axes: title={'center': 'Gravity_z'}>,
        <Axes: title={'center': 'Gravity_y'}>,
        <Axes: title={'center': 'Gravity_x'}>],
       [<Axes: title={'center': 'Gyroscope_z'}>,
        <Axes: title={'center': 'Gyroscope_y'}>,
        <Axes: title={'center': 'Gyroscope_x'}>,
        <Axes: title={'center': 'Location_bearingAccuracy'}>,
        <Axes: title={'center': 'Location_speedAccuracy'}>,
        <Axes: title={'center': 'Location_verticalAccuracy'}>,
        <Axes: title={'center': 'Location_horizontalAccuracy'}>],
       [<Axes: title={'center': 'Location_speed'}>,
        <Axes: title={'center': 'Location_bearing'}>,
        <Axes: title={'center': 'Location_altitude'}>,
        <Axes: title={'center': 'Location_longitude'}>,
        <Axes: title={'center': 'Location_latitude'}>,
        <Axes: title={'center': 'LocationGps_bearingAccuracy'}>,
        <Axes: title={'center': 'LocationGps_speedAccuracy'}>],
       [<Axes: title={'center': 'LocationGps_verticalAccuracy'}>,
        <Axes: title={'center': 'LocationGps_horizontalAccuracy'}>,
        <Axes: title={'center': 'LocationGps_speed'}>,
        <Axes: title={'center': 'LocationGps_bearing'}>,
        <Axes: title={'center': 'LocationGps_altitude'}>,
        <Axes: title={'center': 'LocationGps_longitude'}>,
        <Axes: title={'center': 'LocationGps_latitude'}>],
       [<Axes: title={'center': 'LocationNetwork_bearingAccuracy'}>,
        <Axes: title={'center': 'LocationNetwork_speedAccuracy'}>,
```

```
 <Axes: title={'center': 'LocationNetwork_verticalAccuracy'}>,
 <Axes: title={'center': 'LocationNetwork_horizontalAccuracy'}>,
 <Axes: title={'center': 'LocationNetwork_speed'}>,
 <Axes: title={'center': 'LocationNetwork_bearing'}>,
 <Axes: title={'center': 'LocationNetwork_altitude'}>],
[<Axes: title={'center': 'LocationNetwork_longitude'}>,
 <Axes: title={'center': 'LocationNetwork_latitude'}>,
 <Axes: title={'center': 'Magnetometer_z'}>,
 <Axes: title={'center': 'Magnetometer_y'}>,
 <Axes: title={'center': 'Magnetometer_x'}>,
 <Axes: title={'center': 'Orientation_qz'}>,
 <Axes: title={'center': 'Orientation_qy'}>],
[<Axes: title={'center': 'Orientation_qx'}>,
 <Axes: title={'center': 'Orientation_qw'}>,
 <Axes: title={'center': 'Orientation_roll'}>,
 <Axes: title={'center': 'Orientation_pitch'}>,
 <Axes: title={'center': 'Orientation_yaw'}>,
 <Axes: title={'center': 'Pedometer_steps'}>,
 <Axes: title={'center': 'TotalAcceleration_z'}>],
[<Axes: title={'center': 'TotalAcceleration_y'}>,
 <Axes: title={'center': 'TotalAcceleration_x'}>,
 <Axes: title={'center': 'activity_id'}>, <Axes: >, <Axes: >,
 <Axes: >, <Axes: >]], dtype=object)
```

Columns LocationNetwork_bearingAccuracy, LocationNetwork_speedAccuracy, LocationNetwork_horizontalAccuracy, LocationNetwork_speed, LocationNetwork_bearing, Location_horizontalAccuracy, LocationGps_horizontalAccuracy to check.

[ ]:
```
if 'LocationNetwork_bearingAccuracy' in df.columns:
    print(df['LocationNetwork_bearingAccuracy'].describe())
    print()
if 'LocationNetwork_speedAccuracy' in df.columns:
    print(df['LocationNetwork_speedAccuracy'].describe())
    print()
if 'LocationNetwork_horizontalAccuracy' in df.columns:
    print(df['LocationNetwork_horizontalAccuracy'].describe())
    print()
if 'LocationNetwork_speed' in df.columns:
    print(df['LocationNetwork_speed'].describe())
```

```
    print()
if 'LocationNetwork_bearing' in df.columns:
    print(df['LocationNetwork_bearing'].describe())
    print()
if 'Location_horizontalAccuracy' in df.columns:
    print(df['Location_horizontalAccuracy'].describe())
    print()
if 'LocationGPS_horizontalAccuracy' in df.columns:
    print(df['LocationGPS_horizontalAccuracy'].describe())
    print()
```

```
count    1301793.0
mean           0.0
std            0.0
min            0.0
25%            0.0
50%            0.0
75%            0.0
max            0.0
Name: LocationNetwork_bearingAccuracy, dtype: float64

count    1301793.0
mean           0.0
std            0.0
min            0.0
25%            0.0
50%            0.0
75%            0.0
max            0.0
Name: LocationNetwork_speedAccuracy, dtype: float64

count    1.301793e+06
mean     2.097725e+02
std      3.129762e+02
min      1.150500e+01
25%      1.650200e+01
50%      5.610000e+01
75%      1.496000e+02
max      1.899999e+03
Name: LocationNetwork_horizontalAccuracy, dtype: float64

count    1301793.0
mean           0.0
std            0.0
min            0.0
25%            0.0
50%            0.0
```

```
75%            0.0
max            0.0
Name: LocationNetwork_speed, dtype: float64

count    1301793.0
mean           0.0
std            0.0
min            0.0
25%            0.0
50%            0.0
75%            0.0
max            0.0
Name: LocationNetwork_bearing, dtype: float64

count    1.358801e+06
mean     5.580394e+00
std      2.874915e+00
min      3.389000e+00
25%      3.900000e+00
50%      4.374000e+00
75%      6.814000e+00
max      1.161000e+02
Name: Location_horizontalAccuracy, dtype: float64
```

Almost all columns from above are 0 - delete them.

1.4.4 Reduction of unnecessary features

[ ]:
```
def modify_data(df):
    # Calculate the magnitude of Gyroscope
    df['Gyroscope'] = np.sqrt(df['Gyroscope_x']**2 + df['Gyroscope_y']**2 +␣
↪df['Gyroscope_z']**2)

    # Calculate the magnitude of Magnetometer
    df['Magnetometer'] = np.sqrt(df['Magnetometer_x']**2 +␣
↪df['Magnetometer_y']**2 + df['Magnetometer_z']**2)

    # Calculate the magnitude of Total Acceleration
    df['TotalAcceleration'] = np.sqrt(df['TotalAcceleration_x']**2 +␣
↪df['TotalAcceleration_y']**2 + df['TotalAcceleration_z']**2)

    # Select the required columns
    df = df[['time', 'Gyroscope', 'Location_speed', 'Orientation_roll',␣
↪'Orientation_pitch',
             'Orientation_yaw', 'Pedometer_steps', 'Magnetometer',␣
↪'TotalAcceleration',
```

```
            'Location_longitude', 'Location_latitude', 'Location_altitude',␣
↪'Location_bearing', 'activity_id']]

    return df
```

[ ]:
```
df = modify_data(df)
df
```

[ ]:
```
                time  Gyroscope  Location_speed  Orientation_roll  \
0       1.694673e+15   1.398207        7.007311         -2.679851
1       1.694673e+15   1.370369        7.007311         -2.680510
2       1.694673e+15   1.337961        7.007311         -2.681173
3       1.694673e+15   1.312458        7.007311         -2.681832
4       1.694673e+15   1.288108        7.007311         -2.682458
...              ...        ...             ...               ...
1358796 1.694889e+15   0.314873        0.000000         -0.195289
1358797 1.694889e+15   0.314873        0.000000         -0.195010
1358798 1.694889e+15   0.314873        0.000000         -0.194698
1358799 1.694889e+15   0.314873        0.000000         -0.194441
1358800 1.694889e+15   0.314873        0.000000         -0.194281

         Orientation_pitch  Orientation_yaw  Pedometer_steps  Magnetometer  \
0                 0.671913         0.722495               18     53.467965
1                 0.668292         0.723219               18     53.467965
2                 0.664765         0.723970               18     53.467965
3                 0.661336         0.725430               18     53.467965
4                 0.658039         0.726194               18     53.467965
...                    ...              ...              ...           ...
1358796          -0.669651         1.989292             1489     47.586604
1358797          -0.670229         1.988996             1489     47.586604
1358798          -0.670728         1.985080             1489     47.586604
1358799          -0.671103         1.984742             1489     47.586604
1358800          -0.671491         1.984317             1489     47.586604

         TotalAcceleration  Location_longitude  Location_latitude  \
0                 7.524341            3.138052          50.682839
1                 7.592337            3.138052          50.682839
2                 7.628540            3.138052          50.682839
3                 7.752254            3.138052          50.682839
4                 7.835619            3.138052          50.682839
...                    ...                 ...                ...
1358796           9.093340            3.152370          50.681517
1358797           9.173135            3.152370          50.681517
1358798           9.249142            3.152370          50.681517
1358799           9.264744            3.152370          50.681517
1358800           9.264744            3.152370          50.681517

         Location_altitude  Location_bearing  activity_id
```

```
0                85.099998        315.167419            7
1                85.099998        315.167419            7
2                85.099998        315.167419            7
3                85.099998        315.167419            7
4                85.099998        315.167419            7
...                    ...               ...          ...
1358796          89.900002          0.000000         2780
1358797          89.900002          0.000000         2780
1358798          89.900002          0.000000         2780
1358799          89.900002          0.000000         2780
1358800          89.900002          0.000000         2780

[1358801 rows x 14 columns]
```

1.4.5 Checking for null values

[ ]:
```
def check_null_values(df):
    print("Checking for null values in the dataset")
    for column in df.columns:
        print(column)
        print(df[column].isnull().sum())
    print()
    print()
```

[ ]:
```
check_null_values(df)
```

```
Checking for null values in the dataset
time
0
Gyroscope
0
Location_speed
0
Orientation_roll
0
Orientation_pitch
1
Orientation_yaw
0
Pedometer_steps
0
Magnetometer
0
TotalAcceleration
0
Location_longitude
0
```

```
Location_latitude
0
Location_altitude
0
Location_bearing
0
activity_id
0
```

[ ]:
```
df = df.dropna(subset=['Orientation_pitch'])
```

[ ]:
```
check_null_values(df)
```

```
Checking for null values in the dataset
time
0
Gyroscope
0
Location_speed
0
Orientation_roll
0
Orientation_pitch
0
Orientation_yaw
0
Pedometer_steps
0
Magnetometer
0
TotalAcceleration
0
Location_longitude
0
Location_latitude
0
Location_altitude
0
Location_bearing
0
activity_id
0
```

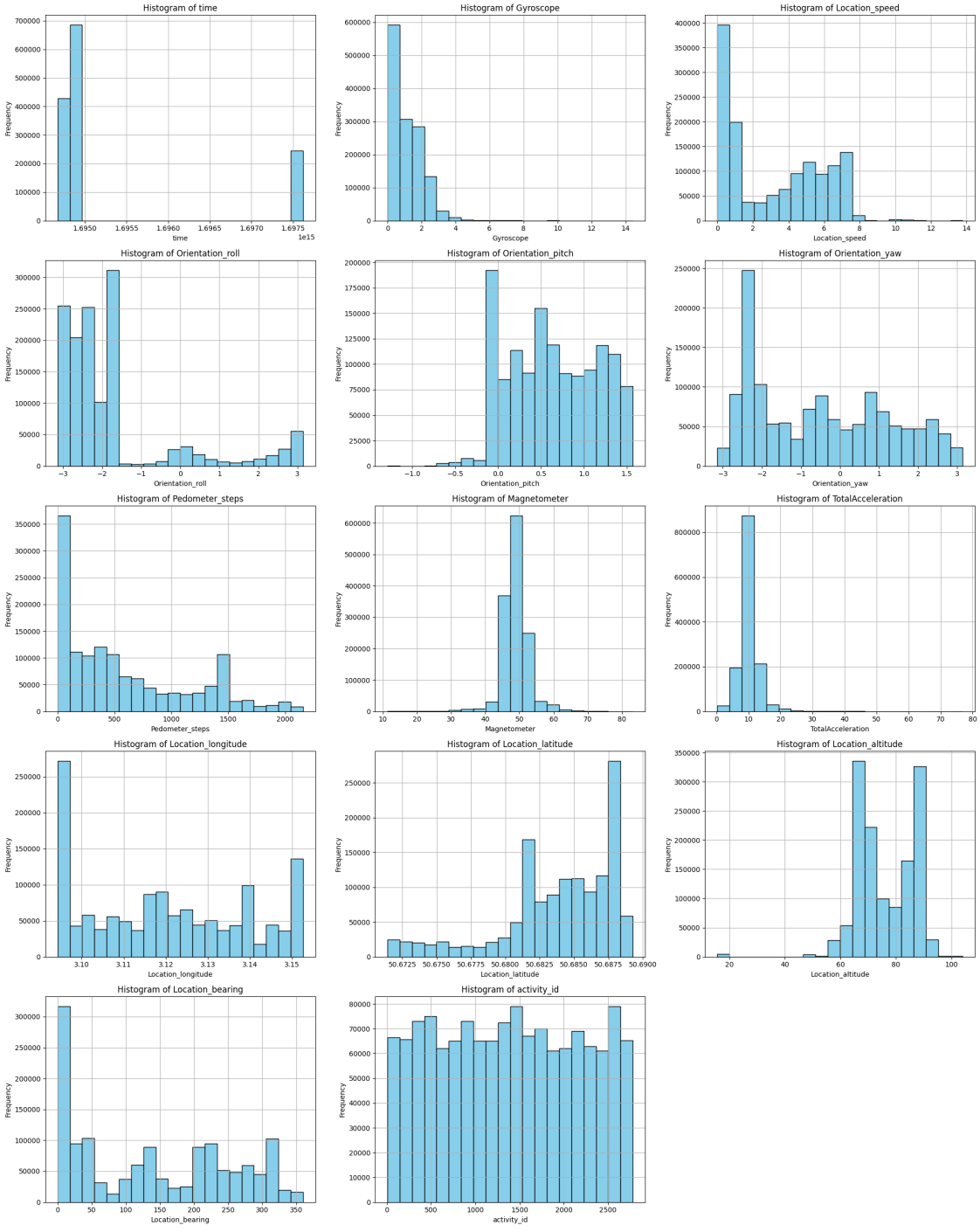
There are no missing values now.

1.4.6 Histograms for every column

[ ]:
```
def plot_histograms(df):
    print("Histograms")

    num_columns = len(df.columns)
    num_rows = (num_columns + 2) // 3

    fig, axes = plt.subplots(num_rows, 3, figsize=(20, 5 * num_rows))
    axes = axes.flatten()

    for i, column in enumerate(df.columns):
        axes[i].hist(df[column], bins=20, color='skyblue', edgecolor='black')
        axes[i].set_title(f'Histogram of {column}')
        axes[i].set_xlabel(column)
        axes[i].set_ylabel('Frequency')
        axes[i].grid(True)

    # Ukrywanie pustych subplotów, jeli istniej
    for j in range(i + 1, len(axes)):
        fig.delaxes(axes[j])

    plt.tight_layout()
    plt.show()

    print()
    print()
```

[ ]:
```
plot_histograms(df)
```

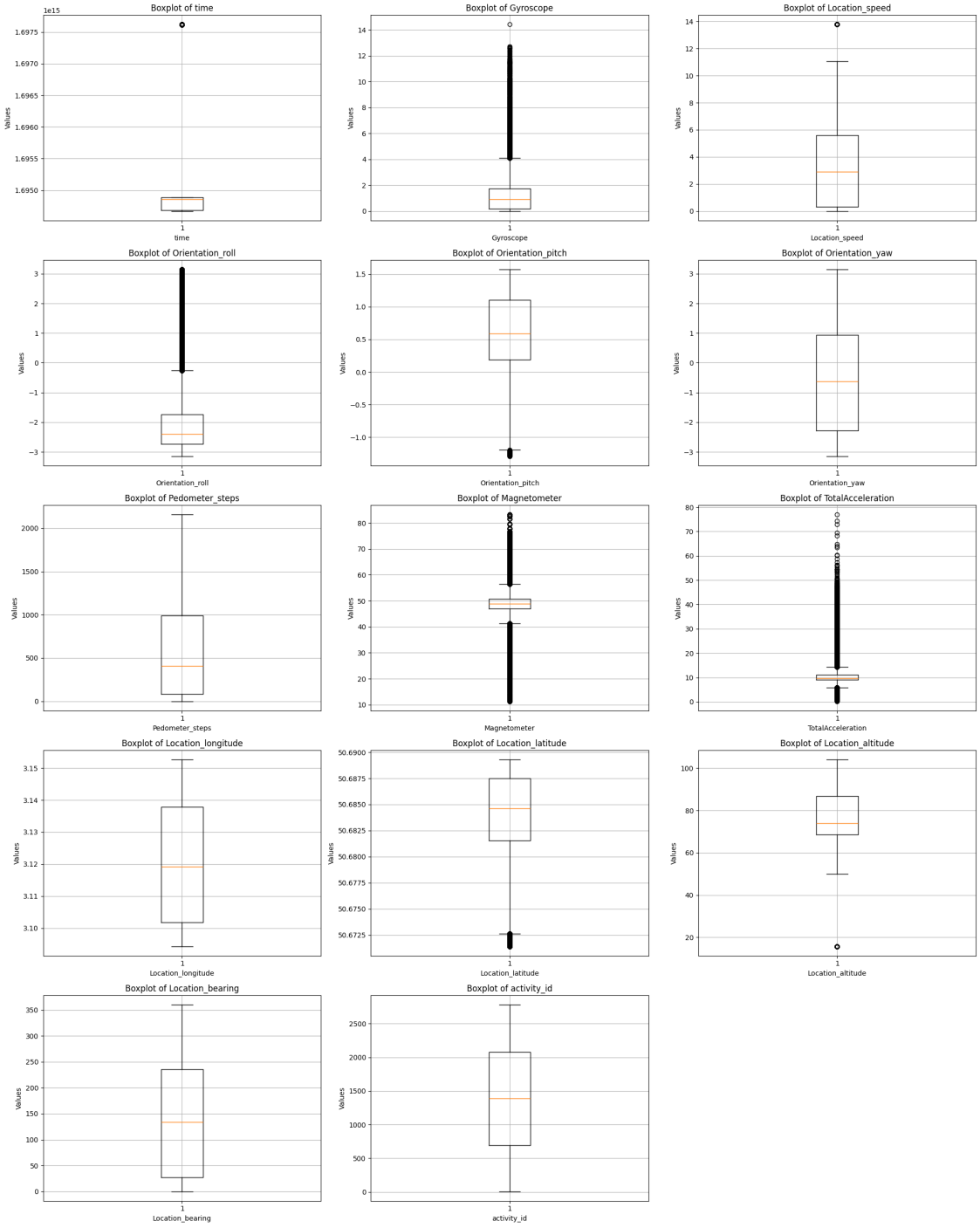
```
Histograms
```


Histogram of time
Frequency
time
1e15
Histogram of Gyroscope
Gyroscope
Histogram of Location_speed
Location_speed
Histogram of Orientation_roll
Orientation_roll
Histogram of Orientation_pitch
Orientation_pitch
Histogram of Orientation_yaw
Orientation_yaw
Histogram of Pedometer_steps
Pedometer_steps
Histogram of Magnetometer
Magnetometer
Histogram of TotalAcceleration
TotalAcceleration
Histogram of Location_longitude
Location_longitude
Histogram of Location_latitude
Location_latitude
Histogram of Location_altitude
Location_altitude
Histogram of Location_bearing
Location_bearing
Histogram of activity_id
activity_id

1.4.7 Boxplots for every column

[ ]:
```
def plot_boxplots(df):
    num_columns = len(df.columns) - 1
    num_rows = (num_columns + 2) // 3
    fig, axes = plt.subplots(num_rows, 3, figsize=(20, 5 * num_rows))
    axes = axes.flatten()

    for i, column in enumerate(df.columns):
        if column == 'id':
            continue
        axes[i].boxplot(df[column])
        axes[i].set_title(f'Boxplot of {column}')
        axes[i].set_xlabel(column)
        axes[i].set_ylabel('Values')
        axes[i].grid(True)

    # Ukrywanie pustych subplotów, jeli istniej
    for j in range(i + 1, len(axes)):
        fig.delaxes(axes[j])

    plt.tight_layout()
    plt.show()
```

[ ]:
```
plot_boxplots(df)
```



1.4.8 Replacing outliers in Location_speed

We think that we should replace outliers in Location_speed column.

[ ]:
```
def replace_outliers_with_quantile(df):
    speed_quantile = df['Location_speed'].quantile(0.95)

    df.loc[df['Location_speed'] > speed_quantile, 'Location_speed'] =␣
 ↪speed_quantile

    return df
```

[ ]:
```
replace_outliers_with_quantile(df)
```

[ ]:
```
                time  Gyroscope  Location_speed  Orientation_roll  \
0       1.694673e+15   1.398207        7.007311         -2.679851
1       1.694673e+15   1.370369        7.007311         -2.680510
2       1.694673e+15   1.337961        7.007311         -2.681173
3       1.694673e+15   1.312458        7.007311         -2.681832
4       1.694673e+15   1.288108        7.007311         -2.682458
...              ...        ...             ...               ...
1358796 1.694889e+15   0.314873        0.000000         -0.195289
1358797 1.694889e+15   0.314873        0.000000         -0.195010
1358798 1.694889e+15   0.314873        0.000000         -0.194698
1358799 1.694889e+15   0.314873        0.000000         -0.194441
1358800 1.694889e+15   0.314873        0.000000         -0.194281

         Orientation_pitch  Orientation_yaw  Pedometer_steps  Magnetometer  \
0                 0.671913         0.722495               18     53.467965
1                 0.668292         0.723219               18     53.467965
2                 0.664765         0.723970               18     53.467965
3                 0.661336         0.725430               18     53.467965
4                 0.658039         0.726194               18     53.467965
...                    ...              ...              ...           ...
1358796          -0.669651         1.989292             1489     47.586604
1358797          -0.670229         1.988996             1489     47.586604
1358798          -0.670728         1.985080             1489     47.586604
1358799          -0.671103         1.984742             1489     47.586604
1358800          -0.671491         1.984317             1489     47.586604

         TotalAcceleration  Location_longitude  Location_latitude  \
0                 7.524341            3.138052          50.682839
1                 7.592337            3.138052          50.682839
2                 7.628540            3.138052          50.682839
3                 7.752254            3.138052          50.682839
4                 7.835619            3.138052          50.682839
...                    ...                 ...                ...
1358796           9.093340            3.152370          50.681517
1358797           9.173135            3.152370          50.681517
1358798           9.249142            3.152370          50.681517
1358799           9.264744            3.152370          50.681517
1358800           9.264744            3.152370          50.681517
```

```
         Location_altitude  Location_bearing  activity_id
0                85.099998        315.167419            7
1                85.099998        315.167419            7
2                85.099998        315.167419            7
3                85.099998        315.167419            7
4                85.099998        315.167419            7
...                    ...               ...          ...
1358796          89.900002          0.000000         2780
1358797          89.900002          0.000000         2780
1358798          89.900002          0.000000         2780
1358799          89.900002          0.000000         2780
1358800          89.900002          0.000000         2780

[1358800 rows x 14 columns]
```

1.5 Feature engineering

We will stary by defining function to calculate total distance.

[ ]:
```python
def haversine_distance(lat1, lon1, lat2, lon2):
    """
    Calculate the great-circle distance between two points
    on the Earth's surface specified in decimal degrees using the Haversine␣
↪formula.
    """
    # Przeliczanie na radiany
    lat1 = math.radians(lat1)
    lon1 = math.radians(lon1)
    lat2 = math.radians(lat2)
    lon2 = math.radians(lon2)

    # Rónica midzy szerokociami i dugociami geograficznymi
    dlat = lat2 - lat1
    dlon = lon2 - lon1

    # Obliczanie odlegoci za pomoc formuy haversine
    a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon␣
↪/ 2)**2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    R = 6371 # rednica Ziemi w kilometrach
    distance = R * c

    return distance # Odlego w kilometrach
```

1.5.1 Function to summarise activity

A function that aggregates available data about one activity.

[ ]:
```
def summarise_activity(df):
    def summarise_group(group):
        total_distance = 0
        for i in range(len(group) - 1):
            distance = haversine_distance(
                group['Location_latitude'].iloc[i], group['Location_longitude'].
↪iloc[i],
                group['Location_latitude'].iloc[i + 1],␣
↪group['Location_longitude'].iloc[i + 1]
            )
            total_distance += distance

        total_steps = group['Pedometer_steps'].max() - group['Pedometer_steps'].
↪min()  # Uycie maksymalnej liczby kroków
        time_duration_seconds = (group['time'].max() - group['time'].min()) /␣
↪10**6  # Czas trwania w sekundach
        time_duration_minutes = time_duration_seconds / 60  # Czas trwania w␣
↪minutach

        if time_duration_minutes == 0:
            steps_per_minute = 0
        else:
            steps_per_minute = total_steps / time_duration_minutes  # Kroki na␣
↪minut

        return pd.Series({
            'id': group['activity_id'].iloc[0],  # Unique id
            'total_time': (group['time'].max() - group['time'].min()) / 10 **␣
↪6,  # Duration in seconds

            'mean_speed': group['Location_speed'].mean() * 3.6,  # Average␣
↪speed in km/h
            'max_speed': group['Location_speed'].max() * 3.6,  # Maximum speed␣
↪in km/h
            'min_speed': group['Location_speed'].min() * 3.6,  # Minimum speed␣
↪in km/h

            'total_distance': total_distance,  # Total distance

            'mean_acceleration': group['TotalAcceleration'].mean(),  # Average␣
↪acceleration
            'max_acceleration': group['TotalAcceleration'].max(),  # Maximum␣
↪acceleration
```

```
            'min_acceleration': group['TotalAcceleration'].min(),  # Minimum␣
↪acceleration
            'sd_acceleration': group['TotalAcceleration'].std(),  # Standard␣
↪deviation of acceleration

            'mean_gyroscope': group['Gyroscope'].mean(),  # Average gyroscope
            'mean_magnetometer': group['Magnetometer'].mean(),  # Average␣
↪magnetometer

            'steps_per_minute': steps_per_minute,  # Steps per minute
            'total_steps': group['Pedometer_steps'].max(),  # Total steps

            'average_roll': group['Orientation_roll'].mean(),  # Average roll
            'median_roll': group['Orientation_roll'].median(),  # Median roll
            'min_roll': group['Orientation_roll'].min(),  # Minimum roll
            'max_roll': group['Orientation_roll'].max(),  # Maximum roll
            'sd_roll': group['Orientation_roll'].std(),  # Standard deviation␣
↪of roll

            'average_pitch': group['Orientation_pitch'].mean(),  # Average␣
↪pitch
            'median_pitch': group['Orientation_pitch'].median(),  # Median␣
↪pitch
            'min_pitch': group['Orientation_pitch'].min(),  # Minimum pitch
            'max_pitch': group['Orientation_pitch'].max(),  # Maximum pitch
            'sd_pitch': group['Orientation_pitch'].std(),  # Standard deviation␣
↪of pitch

            'average_yaw': group['Orientation_yaw'].mean(),  # Average yaw
            'median_yaw': group['Orientation_yaw'].median(),  # Median yaw
            'min_yaw': group['Orientation_yaw'].min(),  # Minimum yaw
            'max_yaw': group['Orientation_yaw'].max(),  # Maximum yaw
            'sd_yaw': group['Orientation_yaw'].std()  # Standard deviation of␣
↪yaw
        })

    grouped = df.groupby('activity_id')
    summary_df = grouped.apply(summarise_group).reset_index(drop=True)

    return summary_df
```

[ ]:
```
result = summarise_activity(df)
```

[ ]:
```
result
```

[ ]:
```
          id  total_time    mean_speed     max_speed     min_speed  \
0        7.0    2.516518  2.408345e+01  2.522632e+01  2.356563e+01
1        8.0    2.516498  2.172662e+01  2.387931e+01  1.837205e+01
```

```
2        9.0    2.516481  1.501819e+01  1.837205e+01  1.257880e+01
3       12.0    2.516464  2.259201e+01  2.355715e+01  1.923772e+01
4       13.0    2.516464  2.374844e+01  2.398566e+01  2.299106e+01
...      ...         ...           ...           ...           ...
1357  2774.0    2.516899  9.276274e-31  5.110263e-30  1.783098e-33
1358  2776.0    2.516883  2.389216e-36  1.065754e-35  2.291695e-39
1359  2777.0    2.516877  8.063703e-40  2.291695e-39  3.944935e-42
1360  2778.0    2.516874  9.794135e-43  3.944935e-42  0.000000e+00
1361  2780.0    0.859112  0.000000e+00  0.000000e+00  0.000000e+00

      total_distance  mean_acceleration  max_acceleration  min_acceleration  \
0           0.013338           9.919805         33.370849          2.437895
1           0.016061           9.841829         19.279124          4.524240
2           0.007480           9.842018         33.347219          3.645055
3           0.020190          10.071431         14.881125          3.962931
4           0.021156           9.505357         13.644567          4.475255
...              ...                ...               ...               ...
1357        0.000000           9.717226         10.134856          9.111182
1358        0.000000           9.713042         10.172372          9.214315
1359        0.000000           9.713457          9.876455          9.330433
1360        0.000000           9.747030         12.981812          7.583295
1361        0.000000          10.352763         13.589607          9.024704

      sd_acceleration  ...  average_pitch  median_pitch  min_pitch  max_pitch  \
0            4.656513  ...       0.983790      1.084632   0.555272   1.180261
1            2.392057  ...       1.074759      1.156837   0.672223   1.221615
2            3.281112  ...       0.920063      0.955256   0.625777   1.166046
3            2.263844  ...       0.889693      0.945513   0.531719   1.186114
4            2.545282  ...       0.843380      0.854464   0.531811   1.167627
...               ...  ...            ...           ...        ...        ...
1357         0.096787  ...      -0.265641     -0.260294  -0.298774  -0.245418
1358         0.111261  ...      -0.260877     -0.256462  -0.307705  -0.236793
1359         0.043896  ...      -0.250566     -0.250087  -0.258994  -0.245895
1360         0.482260  ...      -0.246482     -0.245037  -0.314204  -0.229960
1361         0.747762  ...      -0.604976     -0.651176  -0.708414  -0.281792

      sd_pitch  average_yaw  median_yaw   min_yaw   max_yaw    sd_yaw
0     0.206061     0.926043    0.940593  0.722495  1.136638  0.098998
1     0.162367     1.097486    1.082139  0.880115  1.298065  0.131101
2     0.184394     1.109633    1.101204  0.762258  1.389442  0.170913
3     0.203140     1.555590    1.521060  1.384156  1.799936  0.115415
4     0.201057     1.613284    1.568194  1.453768  1.920251  0.126165
...        ...          ...         ...       ...       ...       ...
1357  0.013399     2.783523    2.790204  2.725673  2.810498  0.017086
1358  0.013606     2.770419    2.778999  2.704930  2.806852  0.033515
1359  0.002698     2.737599    2.740781  2.725769  2.745026  0.006032
1360  0.008723     2.727584    2.732468  2.577463  2.743785  0.023602
```

```
1361  0.103262     2.227272    2.289268  1.984317  2.425556  0.132824

[1362 rows x 29 columns]
```

1.6 Saving to csv file

[ ]:
```
result.to_csv("..//final_data//result_data.csv", index=False)

# result.to_csv("..//final_data//result_data_valid.csv", index=False) # for␣
 ↪validation
# result.to_csv("..//final_data//result_data_test.csv", index=False) # for␣
 ↪validation
```

We have almost ready DataFrame for clustering. It still needs one EDA and one feature engineering, but we'll do that in the next file.