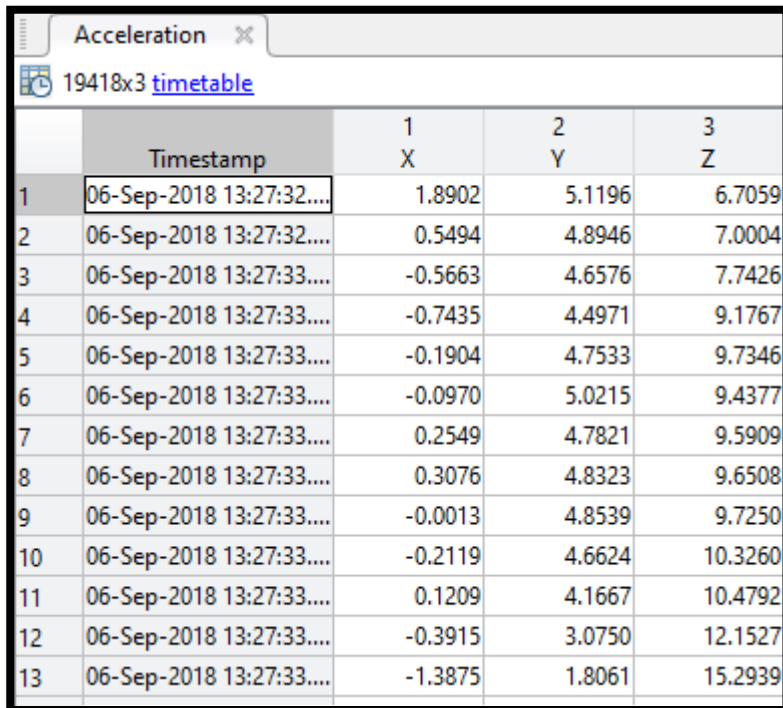


MATLAB Hackathon

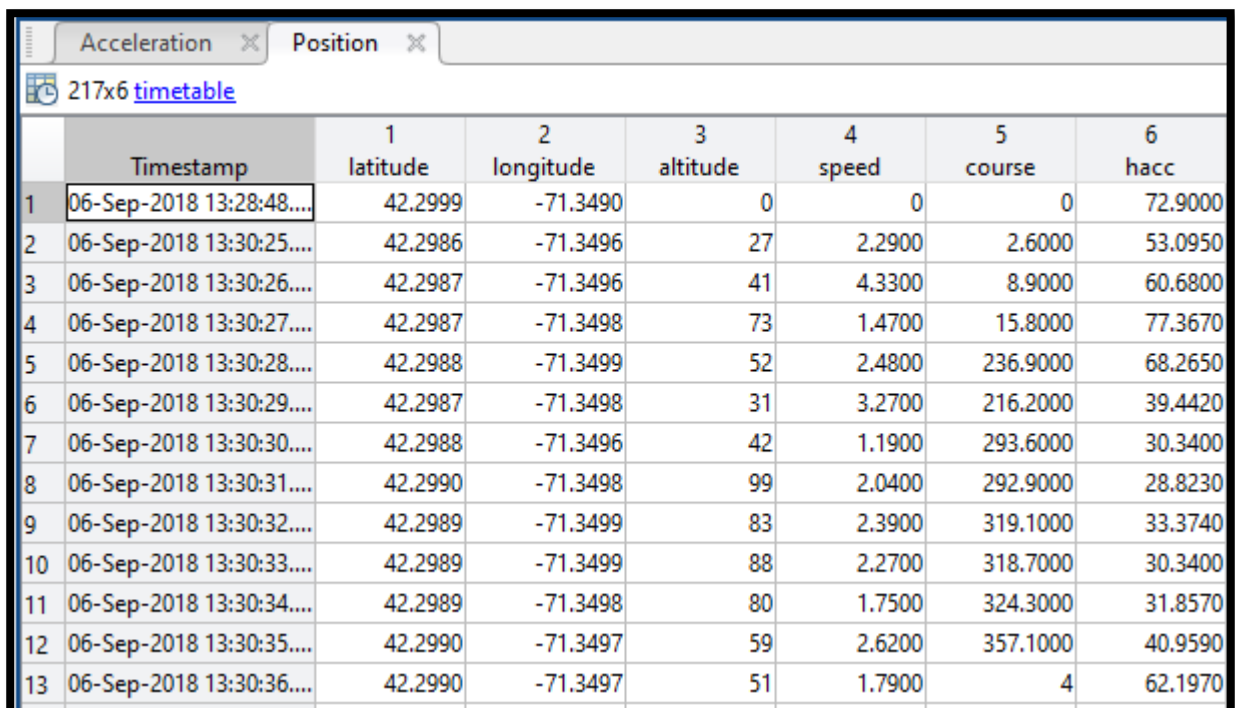
Fitness tracker

As a first step, data were collected from a cell phone on which the MATLAB Mobile application was installed and were transferred to a computer to be uploaded to MATLAB r2022b. The data collected are related to the position obtained by GPS according to latitude and longitude and to the accelerometers; after that once imported, they were extracted to be saved to variables within the workspace.



A screenshot of a MATLAB table titled 'Acceleration' with 19418 rows and 3 columns. The columns are labeled 'Timestamp', '1 X', '2 Y', and '3 Z'. The data shows acceleration values over time from 06-Sep-2018 13:27:32 to 13:27:33.

	Timestamp	1 X	2 Y	3 Z
1	06-Sep-2018 13:27:32....	1.8902	5.1196	6.7059
2	06-Sep-2018 13:27:32....	0.5494	4.8946	7.0004
3	06-Sep-2018 13:27:33....	-0.5663	4.6576	7.7426
4	06-Sep-2018 13:27:33....	-0.7435	4.4971	9.1767
5	06-Sep-2018 13:27:33....	-0.1904	4.7533	9.7346
6	06-Sep-2018 13:27:33....	-0.0970	5.0215	9.4377
7	06-Sep-2018 13:27:33....	0.2549	4.7821	9.5909
8	06-Sep-2018 13:27:33....	0.3076	4.8323	9.6508
9	06-Sep-2018 13:27:33....	-0.0013	4.8539	9.7250
10	06-Sep-2018 13:27:33....	-0.2119	4.6624	10.3260
11	06-Sep-2018 13:27:33....	0.1209	4.1667	10.4792
12	06-Sep-2018 13:27:33....	-0.3915	3.0750	12.1527
13	06-Sep-2018 13:27:33....	-1.3875	1.8061	15.2939



A screenshot of a MATLAB table titled 'Position' with 217 rows and 6 columns. The columns are labeled 'Timestamp', '1 latitude', '2 longitude', '3 altitude', '4 speed', '5 course', and '6 hacc'. The data shows position and acceleration values over time from 06-Sep-2018 13:28:48 to 13:30:36.

	Timestamp	1 latitude	2 longitude	3 altitude	4 speed	5 course	6 hacc
1	06-Sep-2018 13:28:48....	42.2999	-71.3490	0	0	0	72.9000
2	06-Sep-2018 13:30:25....	42.2986	-71.3496	27	2.2900	2.6000	53.0950
3	06-Sep-2018 13:30:26....	42.2987	-71.3496	41	4.3300	8.9000	60.6800
4	06-Sep-2018 13:30:27....	42.2987	-71.3498	73	1.4700	15.8000	77.3670
5	06-Sep-2018 13:30:28....	42.2988	-71.3499	52	2.4800	236.9000	68.2650
6	06-Sep-2018 13:30:29....	42.2987	-71.3498	31	3.2700	216.2000	39.4420
7	06-Sep-2018 13:30:30....	42.2988	-71.3496	42	1.1900	293.6000	30.3400
8	06-Sep-2018 13:30:31....	42.2990	-71.3498	99	2.0400	292.9000	28.8230
9	06-Sep-2018 13:30:32....	42.2989	-71.3499	83	2.3900	319.1000	33.3740
10	06-Sep-2018 13:30:33....	42.2989	-71.3499	88	2.2700	318.7000	30.3400
11	06-Sep-2018 13:30:34....	42.2989	-71.3498	80	1.7500	324.3000	31.8570
12	06-Sep-2018 13:30:35....	42.2990	-71.3497	59	2.6200	357.1000	40.9590
13	06-Sep-2018 13:30:36....	42.2990	-71.3497	51	1.7900	4	62.1970

This is the code that performs the above:

```

load('SensorLog#3.mat');
LAT = Position.latitude;
LON = Position.longitude;
TIMEposition=Position.Timestamp;
Xacc = Acceleration.X;
Yacc = Acceleration.Y;
Zacc = Acceleration.Z;
TIMEacc=Acceleration.Timestamp;

```

Next, standard data for an average person were decided, specifically his weight and the distance traveled by taking a single step; furthermore the variable for total distance traveled was generated:

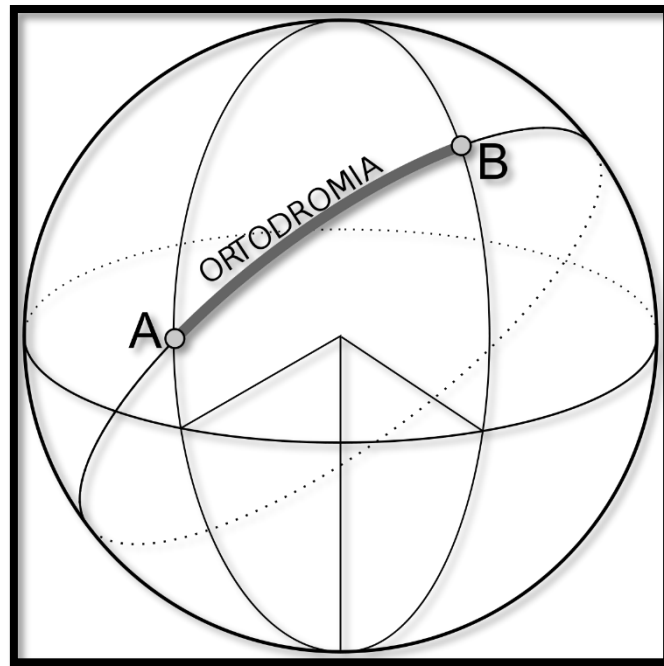
```

%% VARIABLES
step = 0.6; %m
totaldis = 0;
weight = 80;

```

To obtain the value of distance traveled, a Orthodromy formula was used to report the change in angles given by GPS coordinates in distance through the function `d = distance(site1,site2)` and this was done for the entire length of the position vector (divided into latitude and longitude).

The screenshot shows the MATLAB Help Center interface. The top navigation bar includes 'Help', 'Documentation', 'Examples', 'Functions', and 'Apps'. A search bar is located on the right. The left sidebar shows a 'CONTENTS' menu with links to 'Documentation Home', 'RF and Mixed Signal', 'Antenna Toolbox', and 'RF Propagation'. The main content area is titled 'distance' and 'Distance between sites'. It includes a 'Syntax' section with three function signatures: `d = distance(site1,site2)`, `d = distance(site1,site2,path)`, and `d = distance(__,Name,Value)`. The 'Description' section explains that `d = distance(site1,site2)` returns the distance in meters between `site1` and `site2`, `d = distance(site1,site2,path)` returns the distance using a specified path type, and `d = distance(__,Name,Value)` returns the distance with additional options. The 'Examples' section is titled 'Distance Between Transmitter and Receiver Site' and includes a code block for creating transmitter and receiver sites and calculating the Euclidean distance in km. A button labeled 'Open Live Script' is visible.



The code for this process is given here:

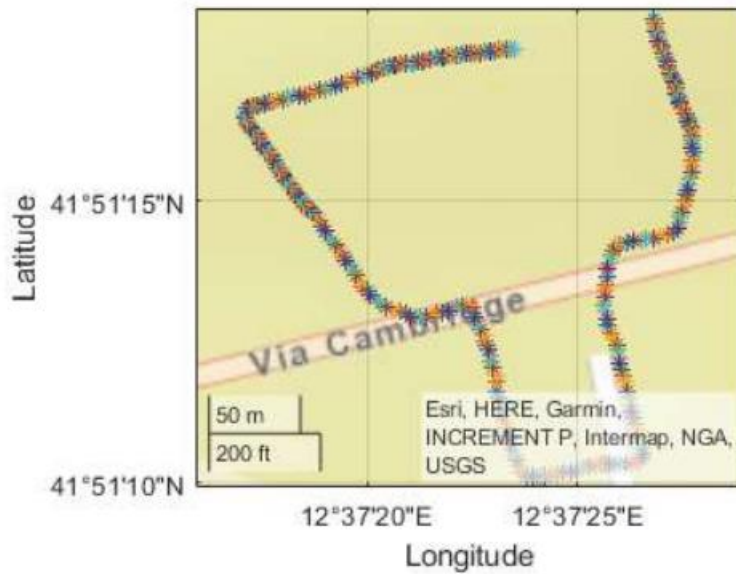
```
for i = 1:(length(LAT)-1)      % Loop through every data sample
    LAT1 = LAT(i);             % Latitude of the iLth sample
    LON1 = LON(i);             % Longitude of the i0th sample
    LAT2 = LAT(i+1);           % Latitude of the (i+1)Ath sample
    LON2 = LON(i+1);           % Latitude of the (i+1)Lth sample
    tx = txsite('Name','MathWorks','Latitude',LAT1,'Longitude',LON1);
    rx = rxsite('Name','Fenway Park','Latitude',LAT2,'Longitude',LON2);

    DIFF = distance(tx,rx,'greatcircle');
    dis = DIFF;
    totaldis = totaldis + dis;
end
totaldis

totaldis = 787.5137
```

The path traveled by the user and its code is shown here:

```
for i = 1:(length(LAT)-1)      % Loop for live plot
    geoplot(LAT(i),LON(i),'-*')
    geobasemap streets          % Rappresentation based on geographic grid
    pause(0.1);
    hold on;
end
```



Finally, the values for kilocalories burned during the run and the total number of steps taken were calculated; the code for this part is shown here:

```
steps = round(totaldis/step)           % Steps taken

steps = 1313

kcal = round(steps * 0.0005 * weight)  % Burned calories

kcal = 53
```

Furthermore, through the vector of positions and the corresponding time vector, we calculated the average speed held by the person; the code is shown here:

```
i = 0;
av_speed = 0;
inst_speed = Position(TIMEposition, "speed" );
inst_speed = table2array(inst_speed);
for i = 1:1:271
    av_speed = av_speed + inst_speed(i);
end
av_speed = (270 / av_speed) * (1000/60) % Average speed in min/Km
```