4-28-2023

MathWorks Fitness Tracker

Tuan Quoc Le

```
clc
clear

%% Preliminary Calculations
% Gender Selection
gender = 0;
genderquestion = questdlg('Please select your gender:', 'Gender
Selection', 'Male', 'Female', 'Cancel', 'Cancel');

switch genderquestion
    case 'Male'
        gender = 1;
    case 'Female'
        gender = 2;
end

if gender ~= 0
    % Age Input
    age = {['Enter your age:']};
    agewindow = 'Age Input';
    age_answer = inputdlg(age, agewindow, 1);
    age = str2double(age_answer{1});

    % Height Input
    heightquestion = {['Enter your height (in):']};
    heightwindow = 'Height Input';
    heightinput = inputdlg(heightquestion, heightwindow, 1);
    height = str2double(heightinput{1});

    % Weight Input
    weightquestion = {['Enter your weight (lbs):']};
    weightwindow = 'Weight Input';
    weightinput = inputdlg(weightquestion, weightwindow, 1);
    weight = str2double(weightinput{1});

    % Calculate BMR
    if gender == 1 % Male
        BMR = 88.362 + (13.397 * weight / 2.205) + (4.799 * height *
2.54) - (5.677 * age);
    else % Female
        BMR = 447.593 + (9.247 * weight / 2.205) + (3.098 * height *
2.54) - (4.330 * age);
    end

    % Calculate BMI
    BMI = (weight / 2.205) / ((height * 2.54 / 100)^2);

    % Display results
    fprintf('Gender: %s\n', genderquestion);
    fprintf('Age: %d years\n', age);
    fprintf('Height: %.2f in\n', height);
    fprintf('Weight: %.2f lbs\n', weight);
    fprintf('BMR: %.2f Calories/day\n', BMR);
    fprintf('BMI: %.2f lb / ft^2\n', BMI * 23.73036);
end

%%Sensor Data (type data file with velocity and acceleration data
you'd like to analyze)
data = importdata('Fitness Tracker Data 3.mat');
position = data.Position;
acceleration = data.Acceleration;

%Preallocation
iRows = size(acceleration, 1);
accelerationMagnitude = zeros(iRows, 1);
jRows = size(position, 1);
positionmagnitude = zeros(jRows, 1);
changeinacceleration = zeros(jRows, 1);
velocity = zeros(jRows, 1);
deltapositionmagnitude = zeros(iRows , 1);
totalCalories = 0;
calories = 0;
totaldeltapositionmagnitude = 0;
for i = 1 : jRows
    positionCompx = position{i, 1}^2;
    positionCompy = position{i, 2}^2;
    positionCompz = position{i, 3}^2;
    positionComponents = positionCompx + positionCompy +
positionCompz;
    positionmagnitude(i) = sqrt(positionComponents);
    if i > 1
        prevPositionTimestamp = position.Timestamp(i-1);
        positionTimestamp = position.Timestamp(i);
        %Acceleration timestamps
        accelerationTimestamps =
acceleration.Timestamp(acceleration.Timestamp > prevPositionTimestamp
& acceleration.Timestamp <= positionTimestamp);
        %Corresponding acceleration indices
        accelerationIndices = find(ismember(acceleration.Timestamp,
accelerationTimestamps));
        %Acceleration magnitudes
        for j = 1 : length(accelerationIndices)
            accelerationCompx = acceleration{accelerationIndices(j),
1}^2;
            accelerationCompy = acceleration{accelerationIndices(j),
2}^2;
            accelerationCompz = acceleration{accelerationIndices(j),
3}^2;
            accelerationComponents = accelerationCompx +
accelerationCompy + accelerationCompz;
            accelerationMagnitude(accelerationIndices(j)) =
sqrt(accelerationComponents);
        end
        %%Change in distance
        deltaX = position{i, 1} - position{i-1, 1};
        deltaY = position{i, 2} - position{i-1, 2};
        deltaZ = position{i, 3} - position{i-1, 3};
        deltapositioncomponents = deltaX^2 + deltaY^2 + deltaZ^2;
        deltapositionmagnitude(i) = sqrt(deltapositioncomponents);
        totaldeltapositionmagnitude = totaldeltapositionmagnitude +
deltapositionmagnitude(i);

        %Change in acceleration
        if j > 1
            changeinacceleration(j) =
accelerationMagnitude(accelerationIndices(j)) -
accelerationMagnitude(accelerationIndices(j-1));
        else
            changeinacceleration(j) =
accelertaionMagnitude(accelerationIndices(j))
        end

        %Velocity
        deltaTime = seconds(positionTimestamp -
prevPositionTimestamp); %Seconds
        velocity(i) = changeinacceleration(i) * deltaTime;

        %Calorie Calculations
        if gender == 1 % Male
            if velocity(i) <= 1.667 || velocity(i) > 0.833
                calories = 5 * deltaTime / 60;
            elseif velocity(i) <= 0.833
                calories = 2.9 * deltaTime / 60;
            elseif velocity(i) <= 2.222 || velocity(i) > 1.667
                calories = 8.1 * deltaTime / 60;
            elseif velocity(i) > 2.778 || velocity(i) > 2.222
                calories = 10.4 * deltaTime / 60;
            end
        elseif gender == 2 % Female
            if velocity(i) <= 1.667 || velocity(i) > 0.833
                calories = 5.8 * deltaTime / 60;
            elseif velocity(i) <= 0.833
                calories = 3.2 * deltaTime / 60;
            elseif velocity(i) <= 2.222 || velocity(i) > 1.667
                calories = 8.3 * deltaTime / 60;
            elseif velocity(i) > 2.778 || velocity(i) > 2.222
                calories = 10.5 * deltaTime / 60;
            end
        end
    end
    totalCalories = totalCalories + calories;
    allowance = BMR + totalCalories;
end
elevationi = position{1 , 3};
elevationf = position{end , 3};
elevationChange = elevationf - elevationi;
```

```
fprintf('Total Distance Traveled = %.2f ft\n',
totaldeltapositionmagnitude * 3.281)
fprintf('Total Steps Taken = %.2f Steps\n',
round(totaldeltapositionmagnitude * 1.312335))
fprintf('Elevation Change = %.2f ft\n', elevationChange * 3.281)
fprintf('Total Calories Burned = %.2f Calories\n', totalCalories)
fprintf('Total Calorie Allowance = %.2f Calories\n', allowance)
```