```
%----------------------------------------------------------
%Merging Files for
%MultiModalBioSignalAnalysis MMBSA
%Bachelor Thesis Guillermo Hidalgo Gadea
%Fatigue detection based on multimodal biosignal analysis
%----------------------------------------------------------

    %% load Metadata for StartDriving and Microsleep
    clc;
    FILENAME = 'F:\Recordings\MetaData\MetaData.xlsx';
    DELIMITER = '\t';
    [num, txt, raw] = xlsread(FILENAME);
    METADATA = raw;
    SUBJECTS = raw(:,1);

    %% match SmartEye & Movisen matrix
    fprintf('Merging SmartEye and Movisens Files...\n');
    PATH1 = 'F:\Processed\SmartEye\';
    PATH2 = 'F:\Processed\Movisens\';
    PATH3 = 'F:\Processed\MovisensAnalyzer\';

    %search SmartEye directory for subjects
    files = dir('F:\Processed\SmartEye\*.csv');
    files = strvcat(files.name);
    x = size(files);
    for i = 1:x(1)
        %start timer
        tic

        %read SmartEye .csv file
        FILE = [PATH1 files(i,:)];
        [a,name,b] = fileparts(FILE);
        fprintf('loading SmartEye File ...');
        fprintf('\n');
        fprintf('filename: %s', name);  % name = vpxxx_pupil_eyelid.csv
        fprintf('\n');
        E = dlmread(FILE);
        TIME1 = E(:,1);

        %get driving time from MetaData
        SUBJECT = strcmp(name(1:5), SUBJECTS);  %search subject name in MetaData
        SUBJECTROW = find(SUBJECT == 1);
        STARTDRIVINGTIME = METADATA{SUBJECTROW,3};
        ENDDRIVINGTIME  = METADATA{SUBJECTROW,4};
        DRIVINGTIME = ENDDRIVINGTIME - STARTDRIVINGTIME;

        %read Movisens .csv file
        fprintf('loading Movisens File ...\n');
        FILE = strcat(PATH2, name(1:5), '_ecg.csv');  %difference between srtcat and []?
        fprintf('filename: %s', FILE(end-12:end-4));
        fprintf('\n');
        F = dlmread(FILE);
        TIME2 = F(:,1);

        %read MovisensAnalyzer .csv file
        fprintf('loading Hrv File ...\n');
        FILE = strcat(PATH3, name(1:5), '_hrv.csv');
        fprintf('filename: %s', FILE(end-12:end-4));
        fprintf('\n');
        HRV = dlmread(FILE);
        TIME3 = HRV(:,1);

        %create common Timeline
        Hz = 100; %set samplerate
        x = DRIVINGTIME * Hz; %leght of TIMELINE with given samplerate
        TIMELINE = TIME2(1:x); %create new timeline recycling TIME2, array to large to be created?
        TIMELINE(1) = STARTDRIVINGTIME; %defining Start time = STARTDRIVINGTIME, same as TIME1 and TIME2

        for i = 2:x  %#ok<FXSET>
            TIMELINE(i) = TIMELINE(i-1) + 1/Hz; %create TIMELINE with given samplerate
        end

        %create placeholder arrays with predefined TIMELINE
        TIMESTAMP = TIMELINE;
        FRAMENUMBER = TIMELINE;
        FRAMERATE = TIMELINE;
        EYELIDOPENING = TIMELINE;
        EYELIDOPENINGQ = TIMELINE;
        PUPILDIAMETER = TIMELINE;
        PUPILDIAMETERQ  = TIMELINE;
        FILTEREDPUPILDIAMETER = TIMELINE;
        FILTEREDPUPILDIAMETERQ = TIMELINE;
        ECG = TIMELINE;
        HR = TIMELINE;
        HrvHf = TIMELINE;
        HrvLf = TIMELINE;
        HrvLfHf = TIMELINE;
        HrvPnn50 = TIMELINE;
        HrvRmssd = TIMELINE;
        HrvSd1 = TIMELINE;
        HrvSd2 = TIMELINE;
        HrvSd2Sd1 = TIMELINE;
        HrvSdnn = TIMELINE;
        HrvSdsd = TIMELINE;

        %replace firts entry in arrays with STARTDRIVING values
        TIMESTAMP(1) = E(1,2);
        FRAMENUMBER(1) = E(1,3);
        FRAMERATE(1) = E(1,4);
        EYELIDOPENING(1) = E(1,5);
        EYELIDOPENINGQ(1) = E(1,6);
        PUPILDIAMETER(1) = E(1,7);
        PUPILDIAMETERQ(1)  = E(1,8);
        FILTEREDPUPILDIAMETER(1) = E(1,9);
        FILTEREDPUPILDIAMETERQ(1) = E(1,10);
        ECG(1) = F(1,2);
        HR(1) = HRV(1,2);
        HrvHf(1) = HRV(1,3);
        HrvLf(1) = HRV(1,4);
        HrvLfHf(1) = HRV(1,5);
        HrvPnn50(1) = HRV(1,6);
        HrvRmssd(1) = HRV(1,7);
        HrvSd1(1) = HRV(1,8);
        HrvSd2(1) = HRV(1,9);
        HrvSd2Sd1(1) = HRV(1,10);
        HrvSdnn(1) = HRV(1,11);
```

```
    HrvSdsd(1) = HRV(1,12);

    %transform SmartEye placeholder INTERPOLATION
    fprintf('merging SmartEye File ...\n');
    for j = 2:x
        START = min(find(TIME1 > TIMELINE(j-1)));  %find smallest element in TIME1 larger TIMELINE(j-1)
        END = max(find(TIME1 <= TIMELINE(j)));  %find largest element in TIME1 smaller than TIMELINE(j)
        if END >= START
            TIMESTAMP(j) = mean(E(START:END,2));
            FRAMENUMBER(j) = mean(E(START:END,3));
            FRAMERATE(j) = mean(E(START:END,4));
            EYELIDOPENING(j) = mean(E(START:END,5));
            EYELIDOPENINGQ(j) = mean(E(START:END,6));
            PUPILDIAMETER(j) = mean(E(START:END,7));
            PUPILDIAMETERQ(j)  = mean(E(START:END,8));
            FILTEREDPUPILDIAMETER(j) = mean(E(START:END,9));
            FILTEREDPUPILDIAMETERQ(j) = mean(E(START:END,10));
        else %if END < START and no values to be averaged, fill up with prior value
            TIMESTAMP(j) = TIMESTAMP(j-1);
            FRAMENUMBER(j) = FRAMENUMBER(j-1);
            FRAMERATE(j) = FRAMERATE(j-1);
            EYELIDOPENING(j) = EYELIDOPENING(j-1);
            EYELIDOPENINGQ(j) = EYELIDOPENINGQ(j-1);
            PUPILDIAMETER(j) = PUPILDIAMETER(j-1);
            PUPILDIAMETERQ(j) = PUPILDIAMETERQ(j-1);
            FILTEREDPUPILDIAMETER(j) = FILTEREDPUPILDIAMETER(j-1);
            FILTEREDPUPILDIAMETERQ(j) = FILTEREDPUPILDIAMETERQ(j-1);
        end
    end

   %transform MovisensAnalyzer placeholder INTERPOLATION
    fprintf('merging Hrv File ...\n');
    for l = 2:x
        START = min(find(TIME3 > TIMELINE(l-1)));  %find smallest element in TIME1 larger TIMELINE(j-1)
        END = max(find(TIME3 <= TIMELINE(l)));  %find largest element in TIME1 smaller than TIMELINE(j)
        if END >= START
            HR(l) = mean(HRV(START:END,2));
            HrvHf(l) = mean(HRV(START:END,3));
            HrvLf(l) = mean(HRV(START:END,4));
            HrvLfHf(l) = mean(HRV(START:END,5));
            HrvPnn50(l) = mean(HRV(START:END,6));
            HrvRmssd(l) = mean(HRV(START:END,7));
            HrvSd1(l) = mean(HRV(START:END,8));
            HrvSd2(l) = mean(HRV(START:END,9));
            HrvSd2Sd1(l) = mean(HRV(START:END,10));
            HrvSdnn(l) = mean(HRV(START:END,11));
            HrvSdsd(l) = mean(HRV(START:END,12));
        else %if END < START and no values to be averaged, fill up with prior value
            HR(l) = HR(l-1);
            HrvHf(l) = HrvHf(l-1);
            HrvLf(l) = HrvLf(l-1);
            HrvLfHf(l) = HrvLfHf(l-1);
            HrvPnn50(l) = HrvPnn50(l-1);
            HrvRmssd(l) = HrvRmssd(l-1);
```

```
                HrvSd1(l) = HrvSd1(l-1);
                HrvSd2(l) = HrvSd2(l-1);
                HrvSd2Sd1(l) = HrvSd2Sd1(l-1);
                HrvSdnn(l) = HrvSdnn(l-1);
                HrvSdsd(l) = HrvSdsd(l-1);
            end
        end

        %transform Movisens placeholder DECIMATION, DOWNSAMPLING
        fprintf('merging Movisens File ...\n')
        START = 1;
        for k = 2:x
            if START+100 > size(TIME2)
                TIME = TIME2(START:end); %TIME2 too long, trim to search in first 20 elements
                vec = (TIME <= TIMELINE(k)); %find elements in TIME smaller/equal TIMELINE(k)in binary
                index = transpose((1:length(vec)));
                c = vec.*index; %translate binary to array index
                END = max(c); %find largest element/index
                END = END + START; %shift index from interval TIME to array TIME2
                ECG(k) = mean(F(START:END,2)); %average all elements between START and END in row 2 (ECG)
                START = END; %START(k+1) = END(k) for next iteration
            else
                TIME = TIME2(START:START+100); %TIME2 too long, trim to search in first 20 elements
                vec = (TIME <= TIMELINE(k)); %find elements in TIME smaller/equal TIMELINE(k)in binary
                index = transpose((1:length(vec)));
                c = vec.*index; %translate binary to array index
                END = max(c); %find largest element/index
                END = END + START; %shift index from interval TIME to array TIME2
                ECG(k) = mean(F(START:END,2)); %average all elements between START and END in row 2 (ECG)
                START = END; %START(k+1) = END(k) for next iteration
            end
        end

        %merge Movisens and SmartEye
        I = horzcat(TIMELINE, ECG, HR, HrvHf, HrvLf, HrvLfHf, HrvPnn50, HrvRmssd, HrvSd1, HrvSd2, HrvSd2Sd1, HrvSdnn, HrvSdsd, TIMESTAMP, FRAMENUMBER, FRAMERATE, EYELIDOPENING, EYELIDOPENINGQ, PUPILDIAMETER, PUPILDIAMETERQ, FILTEREDPUPILDIAMETER, FILTEREDPUPILDIAMETERQ);

        %write to csv
        fprintf('Writing file...\n');
        CSVfile = strcat('F:\Processed\Merged\',name(1:5),'_merged','.csv'); %where to store the outputs...
        dlmwrite(CSVfile, I,'precision','%10.5f'); %missing header
        fprintf('Done!\n');

        %elapsed time in loop
        toc
    end
    fprintf('Data Merging completed!');
    %clear variables after block

```