```
%% FmapsChild.m
%% Takes all young child sound .wav files and finds the formant values
%% All older children, age 8 and above, were considered in the ADULT category
close all;
clear all;
clc;
%% absolute paths to the sound directories
Ball='C:\Users\bu\Documents\ELAT\VowelSpace\ChildrenVocalizationsTestModel\Ball - BEST';
Daddy='C:\Users\bu\Documents\ELAT\VowelSpace\ChildrenVocalizationsTestModel\Daddy';
Jeep='C:\Users\bu\Documents\ELAT\VowelSpace\ChildrenVocalizationsTestModel\Jeep';
No='C:\Users\bu\Documents\ELAT\VowelSpace\ChildrenVocalizationsTestModel\No';
%% make a cell - to "vectorize" directory path variables
sound_dirs={Ball, Daddy,Jeep, No};
num_sounds=length(sound_dirs);
words=cell(1,num_sounds);
%% peel off the directory name from the path
for i=1:num_sounds;
    path=sound_dirs{i};
[path, fname, ext] = fileparts(path);
    %This peels off the last folder from path.
    %The ext must be preserved in case the folder name has a dot it it.
opendir = strcat(fname, ext);
words{i}=opendir;
end
%% Each Directory is a cell. --- Within each cell you keep the struct array returned by the dir↙
function.
data=cell(length(num_sounds),3); %preallocate space given # sound types
disp(data);
    % Each row corresponds to a different sound
    % First column = directory name
    % Second column = "files struct" for .wav files
    % Third column = "formant data struct"
%% fill the data_structure cell columns 1 & 2, with directory names and struct of .wav files.
for i=1:num_sounds;
    wavPATH=fullfile(sound_dirs{i}, '*.wav'); % fullfile(Ball, '*.wav'); <-- gives specified path
    wavLIST=dir(wavPATH); %saves .wav files in a struct call wavs_in_dir
    data{i,2}=wavLIST;
    data{i,1}=words{i};
end %all raw data saved into data_structure
%% data_structure{i,3} holds the processed data <-- Formants
for i=1:num_sounds;
     NUMwavs=numel(data{i,2});
     ith_file=data{i,2};
     formants=struct('F1',[],'maxPxx1', [], 'F2',[],'maxPxx2',[], 'F3', [], 'maxPxx3', []);
     for k=1:NUMwavs;
     [y Fs]= audioread(fullfile(sound_dirs{i},data{i,2}(k).name));
     %need to define the window frame to match the size of the signal vector
        w=ones(1, length(y));
        %nfft points in discrete Fourier Transform (DFT)
        nfft=length(y);
        [pxx,f]= periodogram(y, w,nfft,Fs);
        %% now find the range of frequencies
        % L1= 200 - 800 Hz
        % L2= 800 - 1800 Hz
        % L3= 1800 - 3500 Hz
        %% find Level 1 indices for frequency values between 200 - 800 Hz
```

```
        fmin1=200; fmax1=800;
        L1_ind= find((f>=fmin1)&(f<=fmax1));%L1_ind are indicies that are within 200-800
        %% use L1_ind to find the relevant vectors
            chunkOpsdFreqs1= f(L1_ind); %chunk_O_psd_Frequencies_1 [=] chunkOpsdFreqs1 gives all
the freq values that are [200Hz, 800Hz]
            chunkOpsd1 = pxx(L1_ind); %chunkOpsd1 gives all the pxx values
            peak1=max(chunkOpsd1);
            %want to find the max intensity value this index will give the max intensity which
corresponds to formant 1
            %indices the same, because [pxx, f] makes pxx and f one to one element wise
            F_index1=find(chunkOpsd1==peak1);
            Formant1=chunkOpsdFreqs1(F_index1);
                    %% find Level 2 indices for frequency values between 800 - 1800 Hz
        Fmin2=800; Fmax2=1800;
        L2_ind=find((f>=Fmin2) & (f<=Fmax2));
        %% use L2_ind to find the relevant vectors
        chunkOpsd2= pxx(L2_ind);
        chunkOpsdFreqs2= f(L2_ind);
            peak2=max(chunkOpsd2);
            F_index2=find(chunkOpsd2==peak2);
            Formant2=chunkOpsdFreqs2(F_index2);
        %% find Level 3 indices for frequency values between 1800 - 3500 Hz
        Fmin3=1800; Fmax3=3500;
        L3_ind=find((f>=Fmin3) & (f<=Fmax3));
        %% find Level 3 indices for frequency values between 800 - 1800 Hz
        chunkOpsd3= pxx(L3_ind);
        chunkOpsdFreqs3= f(L3_ind);
            peak3=max(chunkOpsd3);
            F_index3=find(chunkOpsd3==peak3);
            Formant3=chunkOpsdFreqs3(F_index3);
        %% Store all values in formants struct
        % each row has its own formants struct
        formants(k).F1= Formant1; %Hz Values
        formants(k).F2= Formant2;
        formants(k).F3= Formant3;
        formants(k).maxPxx1=peak1; %power spectral density (y) values correlated to the formants,
which are Hz values.
        formants(k).maxPxx2=peak2;
        formants(k).maxPxx3=peak3;
     end
        data{i,3}=formants;
end

%% Get the F#s from the structure in the cell
Plot3D=figure(1);
Plot2D_12=figure(2);
Plot2D_23=figure(3);
Plot2D_13=figure(4);
colors= ['b', 'k', 'r', 'g', 'm'];
for i=1:num_sounds;
    num_pts=length(data{i,3});
           for n=1:num_pts;
            [F1pts]=zeros(1,num_pts);
            [F2pts]=zeros(1,num_pts);
            [F3pts]=zeros(1,num_pts);
           end
```

```
           for n=1:num_pts;
            F1pts(n)=data{i,3}(n).F1;
            F2pts(n)=data{i,3}(n).F2;
            F3pts(n)=data{i,3}(n).F3;
           end
    figure(Plot3D); scatter3(F1pts, F2pts, F3pts, colors(i)); hold on; xlim([200, 900]); ylim↙
([700, 1800]);
    figure(Plot2D_12); scatter(F1pts,F2pts, colors(i), 'o'); hold on; xlim([200,900]); ylim([700,↙
1800]);
    figure(Plot2D_13); scatter(F1pts,F3pts, colors(i), 'o'); hold on; xlim([200,900]); ylim↙
([1700,3600]);
    figure(Plot2D_23); scatter(F2pts,F3pts, colors(i), 'o'); hold on; xlim([700, 1800]); ylim↙
([1700, 3600]);
end

figure(Plot3D);
title('F1 vs F2 vs F3');
grid on; xlabel('F1 (Frequency) in Hz'); ylabel('F2 (Frequency) in Hz'); zlabel('F3 (Frequency)↙
in Hz');
legend(data{1,1}, data{2,1}, data{3,1}, data{4,1});

figure(Plot2D_13);
title ('F1 vs F3');
grid on; xlabel('F1 (Frequency) in Hz'); ylabel('F3 (Frequency) in Hz');
legend(data{1,1}, data{2,1}, data{3,1}, data{4,1});

figure(Plot2D_12);
title ('F1 vs F2');
grid on; xlabel('F1 (Frequency) in Hz'); ylabel('F2 (Frequency) in Hz');
legend(data{1,1}, data{2,1}, data{3,1}, data{4,1});

figure(Plot2D_23);
title('F2 vs F3');
grid on; xlabel('F2 (Frequency) in Hz'); ylabel('F3 (Frequency) in Hz');
legend(data{1,1}, data{2,1}, data{3,1}, data{4,1});
```