```
% get a section of the sound file
filename = 'X:/Evan/MATLAB stuff/percent flight/unofficial
recordings/samplefly.wav';
[x, Fs] = audioread('X:/Evan/MATLAB stuff/Final Scripts/allsoundz.wav');   % load an
audio file
x = x(:, 1);                        % get the first channel
L = length(x);                      % signal length
t = (0:L-1)/Fs;                     % time vector
totalt = max(size(x))/Fs; %finding video length

% perform "fast fourrier transform" fft on the sound file
% This converts the signal from the time domain to a breakdown of its
% component frequencies
Y= fft(x);


if rem(L,2)~=0   %ensures that L/2 is an integer (needed for indexing for P1 and P2)
    L = L+1;
end

% find single-sided amplitude spectrum P1
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1); %https://www.mathworks.com/help/matlab/ref/fft.html
f = Fs*(0:(L/2))/L; % creates vector of frequency domain on which to plot P1

% identify very prominent peaks
[pk, loc] = findpeaks(P1,f,'MinPeakProminence',3*mean(P1), 'MinPeakDistance',100);
pkorder = sort(pk, 'descend');

%%
% this section ensures that only the three desired peaks for tracking are
% counted
i=1;
while i<=length(loc)
    if loc(i)<150 && pk(i)<pkorder(2)
         pk(i) = [];
         loc(i) = [];
    elseif loc(i)<100
         pk(i) = [];
         loc(i) = [];
    elseif loc(i)>200 && loc(i)<400 && pk(i)<pkorder(2)
         pk(i) = [];
         loc(i)= [];
    elseif pk(i)<(2/3)*pkorder(3)
         pk(i) = [];
         loc(i)= [];
    else
        i = i+1;
    end
```

```
   pkorder = sort(pk, 'descend');
end
loc = loc';


%% This section performs a weighted mean (center of gravity) to the top of each peak
to ensure that the middle of the peak is recorded, not just the highest point, which
may be off
%with a larger data set, however, this section may become unnecessary
[mpk, mloc] = findpeaks(P1,f, 'MinPeakDistance', 5);


i = 1;
zpk=pk;
while i<= length(loc)
    j = 1;
    counter=1;

    while j<= length(mloc)
        if abs(mloc(j)-loc(i))<=100 && abs(mpk(j)-pk(i))<=0.4*pk(i)
            counter = counter+1;

            zloc(i,1) = loc(i)*pk(i);
            zloc(i,j+1) = mpk(j)*mloc(j);
            zpk(i) = zpk(i)+mpk(j);

        end
        j=j+1;
    end
    zloc(i)= sum(zloc(i,:))./zpk(i);
    zloc = zloc(:,1);
    zpk(i) = zpk(i)/counter;
    i=i+1;
end
% pk = zpk;
loc(1:3) = zloc(1:3);
%% This section displays the peak frequencies.
% It plots P1 with the three tracked values appearing as circles
% and lists the detected frequencies in the command window

fprintf('peak %.0f was identified at %.2f Hz\n',[1:length(loc);loc'])
%.0f


hold on
figure(ii)
one = ones(1,length(f));
plot(f,P1)
title('Single-Sided Amplitude Spectrum of X(t)');
xlabel('Frequency [kHz]');
```

```
ylabel('|P1(f)|');
hold on
plot(loc,pk,'o')
hold off

% hold on
% plot(mloc,mpk,'x')

axis([0 1000 0 1.25*max(pk)])

% axis([0 1000 0 15e-2]);
% f = get(0,'Format')

%%
% figure(1)
% plot(t,x)
%
% newY =ifft(abs(Y(f==f(120<=f<=250))));
%
% x==abs(ifft(Y))
% % figure(2)
% % plot(t,ifft(Y(f==f(120<=f<=250))))
```