```
%correlationwindower.m
%A script which windows an activity vector by requiring the cross correlation of a sample data set to be above a certian threshold, and
%then counts the peaks which reside within that window
%Developed by www.activityhacker.com

%The following variables must be loaded into Matlab before begining
%t = Time (Seconds)
%x = X axis values (Gs)
%y = Y axis values (Gs)
%z = Z axis values (Gs)

% %Plot the input data
% plot(t,x,t,y,t,z)
% legend('X Axis','Y Axis','Z Axis');
% xlabel('Time (Seconds)');
% ylabel('Force (Gs)');

%Propt User for Analysis Parameter Inputs
prompt = {'Enter Start Time for Calibration Signal (s):','Enter Stop Time For Calibration Signal (s):','Enter Sample Frequency (Hz)','Enter Correlation Windowing Threshold (%)','Enter Minimum Peak Spacing (Samples)'};
dlg_title = 'Input';
num_lines = 1;
def = {'45','55','25','50','10'};
answer = inputdlg(prompt,dlg_title,num_lines,def);

%convert answer cell inputs to variables
ti = str2double(answer(1));
to = str2double(answer(2));
f = str2double(answer(3));
w = str2double(answer(4));
d = str2double(answer(5));

%convert windowing threshold to percent
w = w/100;

%extract the correlation signal
xs = x(ti*f:to*f);
ys = y(ti*f:to*f);
zs = z(ti*f:to*f);
ts = t(ti*f:to*f);

%generate the correlation vectors
xc = xcorr(x,xs);
yc = xcorr(y,ys);
zc = xcorr(z,zs);

%shorten the correlation vectors to remove 0s
xc = xc(length(x):length(xc));
yc = yc(length(y):length(yc));
```

```
zc = zc(length(z):length(zc));

%Normalize correltation vector amplitude to match axis amplitude
xc = (xc/max(xc))*max(x);
yc = (yc/max(yc))*max(y);
zc = (zc/max(zc))*max(z);

%create a windowing vector based on the windowing threshold input
i = 1;
xcc = [];
while i <= length(xc);
    if xc(i)<= w*max(xc);
        xcc = [xcc 0];
    elseif xc(i) > w*max(xc);
        xcc = [xcc 1];
    end
    i = i+1;
end

%create a windowed signal for the x axis by multiplying original signal by
%the windowing vector
xw = x.*xcc';

%calculate the average of all non-zero elements for the windowed signal
x_mean = sum(xw)/nnz(xw);

%caluclate peaks and peak locations for x axis
[x_pks, x_locs] = findpeaks(xw,'MINPEAKHEIGHT',x_mean,'MINPEAKDISTANCE',d);

%Loop turns peak locations from being relative samples to absolute in time
i=1;
x_peaks=[];
while i<=length(x_locs);
    temp =[t(x_locs(i)); x_pks(i)];
    x_peaks =[ x_peaks temp];
    i = i+1;
end;

hold on
meanx = ones(1,length(t))*x_mean;
plot(x_peaks(1,:),x_peaks(2,:),'k.');
plot(t,x,t,meanx), xlabel ('Time (s)'), ylabel('Force (Gs)');

%Print Number of "Steps" based on each Axes' data:
%avgsteps = round((length(x_locs)+length(y_locs)+length(z_locs))/3);
fprintf('X Axis Steps %d\n', length(x_locs));

%plot out the correlation vectors on top of the actual signal for each axis
% null = zeros(1,length(t));
% tmax = max(t);
% subplot(3,1,1), plot(t,xc,t,x,t,null,'k'), title('X Axis'), xlabel ('Time (s)'), ylabel↙
```

```
('Force (Gs)'), xlim([0 tmax]);
% subplot(3,1,2), plot(t,yc,t,y,t,null,'k'), title('Y Axis'), xlabel ('Time (s)'), ylabel
('Force (Gs)'), xlim([0 tmax]);
% subplot(3,1,3), plot(t,zc,t,z,t,null,'k'), title('Z Axis'), xlabel ('Time (s)'), ylabel
('Force (Gs)'), xlim([0 tmax]);
```