Table of Contents


```
% INSTRUCTIONS:
% 1) save this file into the directory that holds your work
% 2) make sure it runs without error
% 3) publish this file to PDF.  There are two ways to do this:
%    a) You can either use the 'Publish' tab, being sure to edit the
%    publishing options so the output format is 'pdf', or
%    b) run the commands below in the command window, one at a
%    time: (without the comment symbol!)
%    options.format = 'pdf';
%    publish('runAllHW4',options);
%
% 4) There should then be a PDF file in the subdirectory 'html'
%    that you can upload to TurnItIn.
%    PLEASE DOUBLE-CHECK it has everything you need!!

% prepare to run...
clear all  % clear out all variables
close all  % close any open figures
```

list ECG project files

```
dbtype('MainScript')
fprintf('\n --------------------------- \n')
dbtype('SetupTimeVector')
fprintf('\n --------------------------- \n')
dbtype('FindEcgPeaks')
fprintf('\n --------------------------- \n')
dbtype('CalcInstHR')
fprintf('\n --------------------------- \n')
dbtype('CalcStats')
```

```
1     %Alyssa Rose  HW4  2-19-18
2     %loads in data
3     load ecgClean.mat
4     %time vector generated
5     tvec = SetupTimeVector(length(ecgData),Fs);
6
7     %peaks in ecgData found
8     threshold = 1050;
9     iPeaks = FindEcgPeaks(ecgData,threshold);
10
```

```
    %heart rate calculated using previous 2 functions' values
    instBPM =  CalcInstHR(tvec,iPeaks);

    %output string made using heart rate data
    outStr = CalcStats(instBPM);

    %ecg plotted with peaks marked w/ red circles
    figure
    subplot(2,1,1)
    plot(tvec,ecgData)
    xlabel('Time,sec')
    ylabel('ECG signal, A/D units')
    title(outStr)
    hold on
    plot(tvec(iPeaks), ecgData(iPeaks),'ro')
    hold off

    %inst. beats per minute plotted
    subplot(2,1,2)
    plot(instBPM,'--s')
    xlabel('Beat #')
    ylabel('Instantaneous HR,BPM')
    %generated plots saved as jpeg file
    saveas(gcf, 'figure.jpg')
    %{
    1.) When threshold = 0, then every point will be labeled
    as a peak when MainScript is run, and the entire graph is
    covered in red dots.
       mean: 6048    median: 5400

    2.)When threshold = 1120, the top graph appears normal,
    whereas the instBPM graph returns values that seem impossible
    or unlikely for BPM values. This happens due to the limited
 data,
    and small difference in magnitude of peaks (since they are all
 high).
       mean: 64      median: 73
    the median value appears to be closer to expected values, and
 would
    be more accurate since it diminshes the effects of the drastic
    differences in instBPM values.

    3.) Thresholds between 1030 and 1130 appeared to give believable
 values
    for instBPM since the peaks were not too small or too high in
 value. If
    the values were smaller or larger than these thresholds, the #
 of peaks
    were too great or too small, resulting in drastic differences in
 peak
    values or not enough peaks (skewing data).

    4.)The large distortion causes the instBPM to be out of range of
 normal
```

```
57    values, with the ecg peaks occuring at large values and
 appearing to
58    cluster around 12 sec.
59        mean: 3821     median: 3600
60    neither value seems to be accurate, although median would be
 closer since
61    its value is less than the mean.
62
63    5.)Setting the threshold to around 1100 causes the data
64    to fit into an expected range since this is where most major
 peaks occured.
65        mean: 73      median: 83
66    the mean value is more accurate since it matches most closely
 with
67    previously calculated values
68    %}
```

```
1     function tvec = SetupTimeVector(npoints,Fs)
2     %{
3     returns a vector 'tvec' of times in seconds,
4     of length npoints, starting at t=0 and
5     spaced apart by 1/Fs seconds
6
7     Alyssa Rose  HW4  2-19-18
8     %}
9
10    %checks if values for npoints & Fs are allowed
11    if npoints <= 0 | Fs <= 0
12        tvec = [];
13        return
14    %preallocates tvec
15    %creates tvec by adding on a "step size" each loop
16    else
17        tvec = zeros(1,npoints);
18        step = (1/Fs);
19        for k = 2:npoints
20            tvec(k) = step+tvec(k-1);
21        end
22    end
23
24    return
```

```
1     function iPeaks  = FindEcgPeaks(ecg,threshold)
2     %{
3     find samples in input vector 'ecg' that are
4     peaks, and are above the value 'threshold'
5
6     Alyssa Rose  HW4  2-19-18
7     %}
8     n=length(ecg);
```

```
9     iPeaks = [];
10
11    %checks if first value is a peak
12    if (ecg(1) > threshold) && (ecg(1)> ecg(2))
13        iPeaks(1) = 1;
14    end
15    %checks if current value is local max and if
16    %the peak is greater than the threshold
17    for m = 2:(n-1)
18        if (ecg(m)>ecg(m-1))&& (ecg(m)>ecg(m+1))
19            if ecg(m)>threshold
20                iPeaks = [iPeaks, m];
21            end
22        end
23    end
24
25    %checks if last value is a peak
26    if (ecg(n)> ecg(n-1)) && (ecg(n) > threshold)
27        iPeaks(n) = n;
28    end
29
30    %gets rid of zeros, only stores values of peaks
31    delete = iPeaks ~= 0;
32    iPeaks = [iPeaks(delete)];
33    return
```

```
1     function instBPM = CalcInstHR(tvec,iPk)
2     %{
3     given a time vector tvec and list of samples for peaks,
4     returns a vector instBPM with the instantaneous beats per minute
5
6     Alyssa Rose  HW4  2-19-18
7     %}
8
9     %converts sec to minutes and creates empty vector
10    tvec = tvec./60;
11    instBPM = [];
12    if length(iPk)<2
13        instBPM = 0;
14        return
15    end
16    %calculates heart rate and stores it in instBPM
17    for k =1:length(iPk)-1
18        heartRate = 1/(tvec(iPk(k+1))- tvec(iPk(k)));
19        instBPM = [instBPM heartRate];
20    end
21    return
```

```
1     function outStr = CalcStats(instBPM)
2     %{
```

```
3     computes statistics (mean, median, std dev)
4     and defines output string
5
6     Alyssa Rose  HW4  2-19-18
7     %}
8
9     %calculates whole number mean value
10    X = mean(instBPM);
11    X=round(X);
12    %calculates whole number median value
13    Y = median(instBPM);
14    Y = round(Y);
15    %calculates standard deviation
16    Z = std(instBPM);
17    %prints out values w/ stdev to 1 decimal
18    outStr = sprintf('Heartrate, BPM: mean %d, median %d, stdev
 %.1f',X,Y,Z);
19    return
```

list and the vectorization problem

```
fprintf('\n\n --------------------------- \n\n')
type('HotTubVectors')
```

```
 ---------------------------


function [tempsHot, timesCold, nOK] = HotTubVectors(time,temp)
%{
Function takes in 2 equal sized vectors; temp(Fahrenheit) & time
 (hours);
and determines how well the hot tub stays within temp
 range( 102<=temp<=105).

Alyssa Rose  HW4  2-19-18
%}

%extracts temperatures > 105 degrees
tempsHot = temp(temp > 105);

%extracts times when temperature < 102 degrees
t = find(temp < 102);
timesCold = time(t);

%finds # hours when hot tub was in desired range
nOK= sum((temp >= 102) & (temp <= 105));
return
```

run ecg project

```
fprintf('\n\n --------------------------- \n\n')
```

```
MainScript


 ----------------------------
```

Published with MATLAB® R2017b