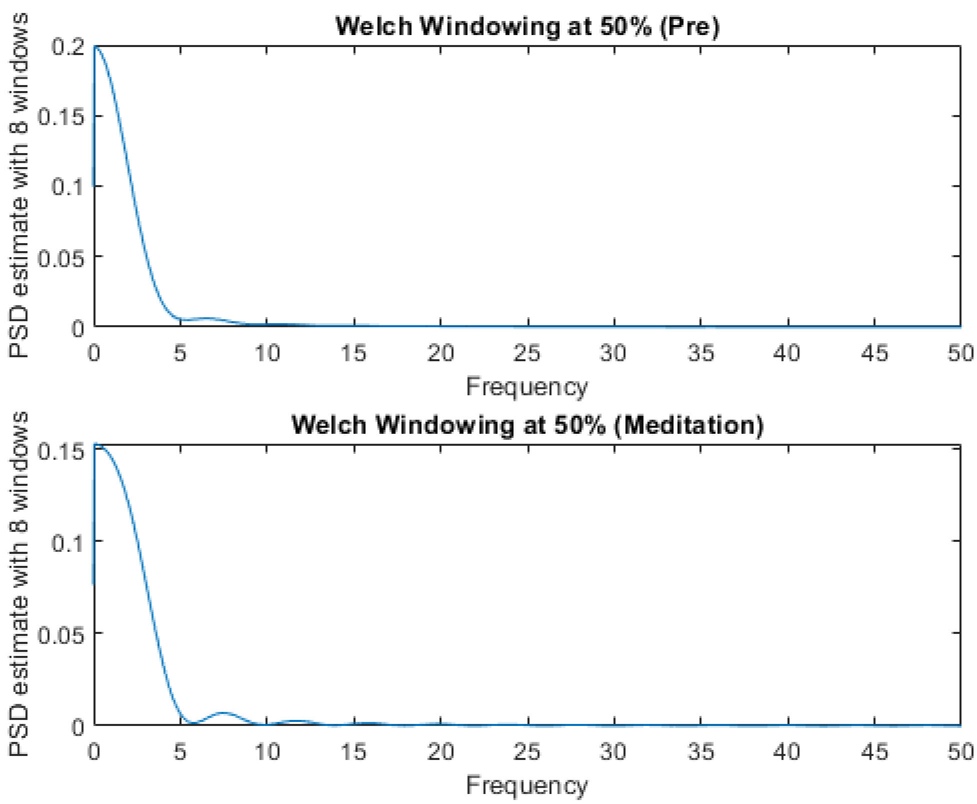
```
% Christopher Morales
% EE 4820   3/22/22
```

Problem 1

```
% Sample Frequency
fs = 100;

% Sample Interval
ts = 1/fs;

% Loading data
load Hr_pre.mat;
load Hr_med.mat;

% Evenly spaced time vector
tpre = t_pre(1) : ts : t_pre(end);
tmed = t_med(1) : ts : t_med(end);

% Creating the length and rounding it
L = round(2*fs);

% How much resolution we need
Nfft = 2^11;

% Creating how many windows on a vector of ones
win = ones(round(L/8), 1);

% Getting the length of the window
Lwin = round(L/10);

% What percentage of overlap do we want for the window (in this case 50%)
Noverlap = round(Lwin*0.5);

% Getting the frequency of the sinsoidal
fsigpre = 1 / ( t_pre(2) - t_pre(1) );
fsigmed = 1 / ( t_med(2) - t_med(1) );

% Sinsoidal
ypre = sin(2 * pi * fsigpre * tpre);
ymed = sin(2 * pi * fsigmed * tmed);

% Welch with 8 windows and 50% overlap
[Pxxpre, fpre] = pwelch(ypre, win, Noverlap, Nfft, fs);
[Pxxmed, fmed] = pwelch(ymed, win, Noverlap, Nfft, fs);


% Creating a figure
figure(1);

% Using the definition of power density, in frequency, parseval theorem, we
```

```
% can get the magnitude of Pxx
subplot(2,1,1);
plot(fpre, Pxxpre);
title("Welch Windowing at 50% (Pre)");
xlabel('Frequency');
ylabel('PSD estimate with 8 windows');


subplot(2,1,2);
plot(fmed, Pxxmed);
title("Welch Windowing at 50% (Meditation)");
xlabel('Frequency');
ylabel('PSD estimate with 8 windows');
```



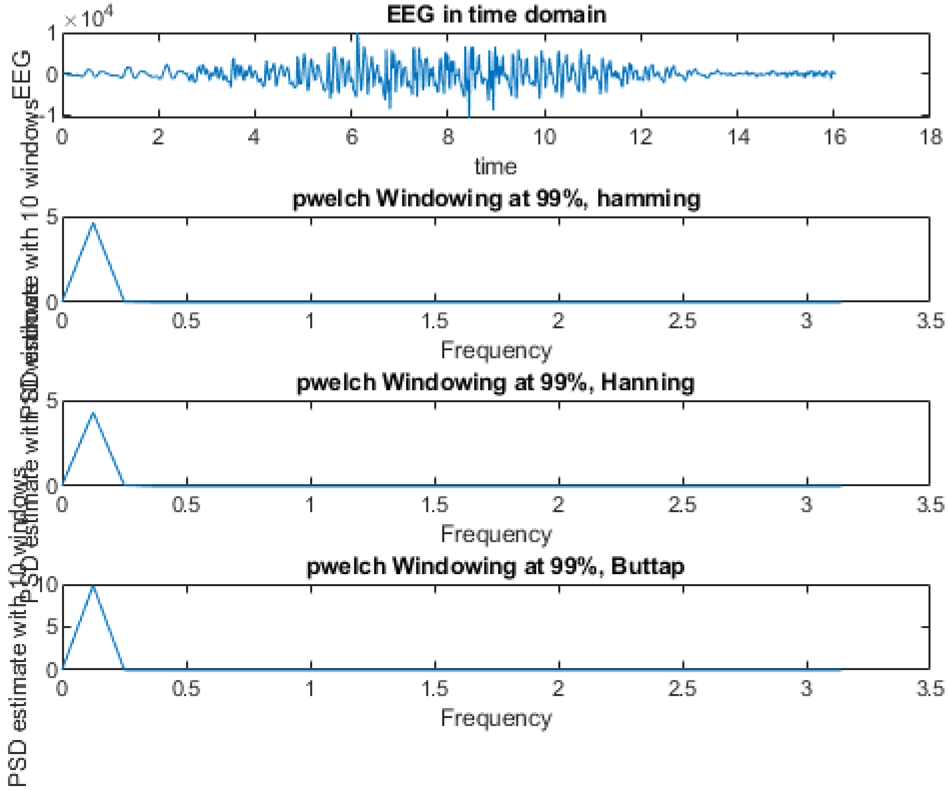
Problem 2

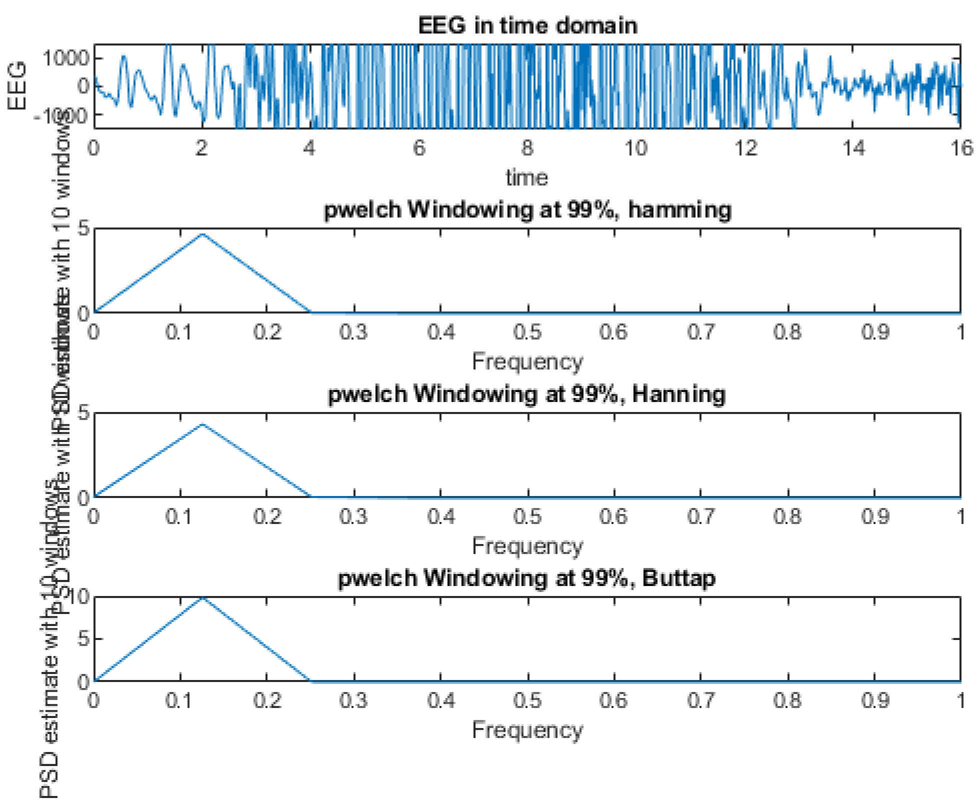
```
% load data
load eeg_data.mat;

% Sample Frequency
fs = 50;

% How much resolution we need
Nfft = 2^11;
```

```
% Creating a length
L = length(eeg);

% Creating the time vector
t = [1:L]/fs;

% Creating the length of the window
Lwin = round(L/10);

% Creating the window
win = window(@hamming, Lwin);
win1 = window(@hanning, Lwin);
win2 = window(@buttap, Lwin);

% What percentage of overlap do we want for the window (in this case 99%)
Noverlap = round(Lwin * 0.99);

% Creating the sinsoidal
y = sin(2*pi*t);

% Welch with 10 windows and 99% overlap
[Pxx, f] = pwelch(y, win, Noverlap, fs);
[Pxx1, f1] = pwelch(y, win1, Noverlap, fs);
[Pxx2, f2] = pwelch(y, win2, Noverlap, fs);

% Creating the figure
figure(2);

% Plotting the egg in time
subplot(4, 1, 1);
plot(t, eeg);
title('EEG in time domain');
xlabel('time');
ylabel('EEG');

subplot(4, 1, 2);
plot(f, Pxx);
title('pwelch Windowing at 99%, hamming');
xlabel('Frequency');
ylabel('PSD estimate with 10 windows');

subplot(4, 1, 3);
plot(f1, Pxx1);
title('pwelch Windowing at 99%, Hanning');
xlabel('Frequency');
ylabel('PSD estimate with 10 windows');

subplot(4, 1, 4);
plot(f2, Pxx2);
title('pwelch Windowing at 99%, Buttap');
xlabel('Frequency');
ylabel('PSD estimate with 10 windows');
```

```
fprintf('All 3 methods seemed to look the same but upon zooming \n');
fprintf('we can see the x axis being chopped off or slightly shifted!!!\n');
fprintf('I also noticed that for Buttap the height double compared to the others\n');
fprintf('Another figure will displayed the same graphs but zoomed in \n');

% Creating the figure
figure(3);

% Plotting the egg in time
subplot(4, 1, 1);
plot(t, eeg);
title('EEG in time domain');
xlabel('time');
ylabel('EEG');
axis([0 16 -1500 1500]);

subplot(4, 1, 2);
plot(f, Pxx);
title('pwelch Windowing at 99%, hamming');
xlabel('Frequency');
ylabel('PSD estimate with 10 windows');
axis([0 1 0 5]);

subplot(4, 1, 3);
plot(f1, Pxx1);
title('pwelch Windowing at 99%, Hanning');
xlabel('Frequency');
ylabel('PSD estimate with 10 windows');
axis([0 1 0 5]);

subplot(4, 1, 4);
plot(f2, Pxx2);
title('pwelch Windowing at 99%, Buttap');
xlabel('Frequency');
ylabel('PSD estimate with 10 windows');
axis([0 1 0 10]);
```

```
All 3 methods seemed to look the same but upon zooming
we can see the x axis being chopped off or slightly shifted!!!
I also noticed that for Buttap the height double compared to the others
Another figure will displayed the same graphs but zoomed in
```


EEG in time domain
×10^4
EEG
time
pwelch Windowing at 99%, hamming
pwelch Windowing at 99%, Hanning
pwelch Windowing at 99%, Buttap
PSD estimate with 10 windows
Frequency

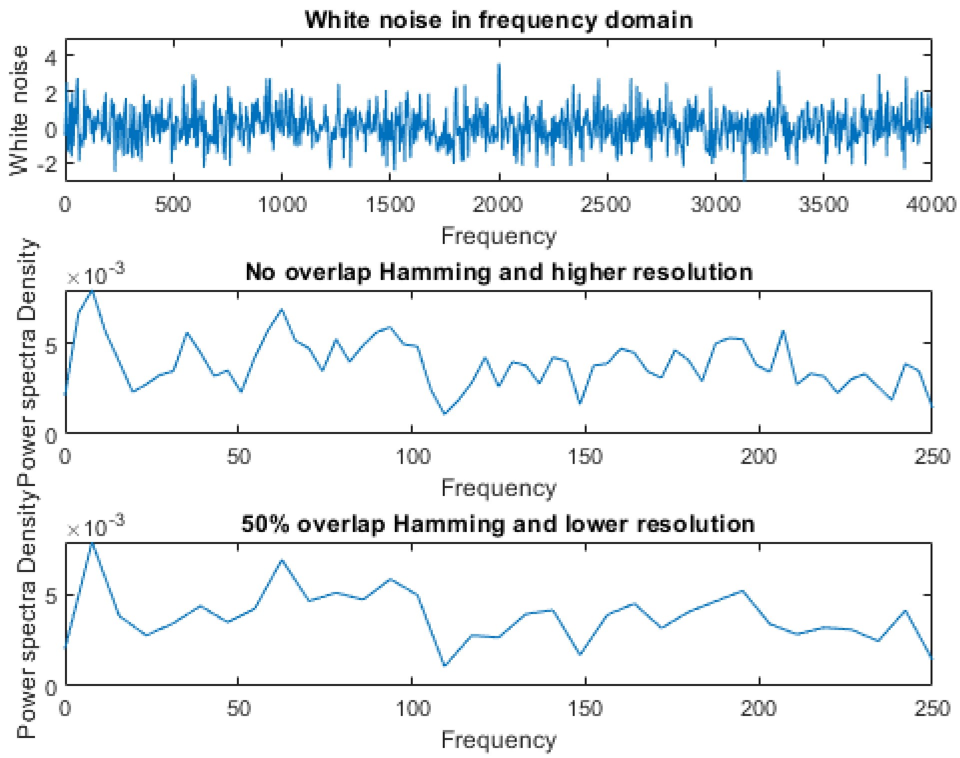

EEG in time domain
EEG
time
pwelch Windowing at 99%, hamming
pwelch Windowing at 99%, Hanning
pwelch Windowing at 99%, Buttap
PSD estimate with 10 windows
Frequency

Problem 3

```
% Samples
N = 1024;

% Sample Frequency
fs = 500;

% How much resolution we want
Nfft = 2^7;
Nfft2 = 2^6;

% Creating the time vector
t = [0:(N-1)]/fs;

% Creating the frequency vector
f1 = [0:(N-1)]*fs/Nfft;

% A white noise vector
noise = randn(1, N);

% The power spectrum direct appoach (equation 3.34)
y = abs(fft(noise)).^2;

% Creating the window
win = window(@hamming, Nfft);

% I dont think this is not needed due to the resolution itself
% Will be enough to determine the difference and rather than the iwndows
%win2 = window(@hamming, Nfft2);


% Creating the length of the window
L = length(t);
Lwin = round(t*0.50);

% Creating overlap but in this case we want 0%
Noverlap = 0;
Noverlap2 = 0.5;

% pwelch for different overlap and resolution
[Pxx, f] = pwelch(noise, win, Noverlap, Nfft, fs );
[Pxx2, f2] = pwelch(noise, win, Noverlap2, Nfft2, fs );


% Creating a figure
figure(4);

% Plotting the white noise in frequency
subplot(3,1,1);
plot(f1, noise);
title('White noise in frequency domain');
xlabel('Frequency');
```

```
ylabel('White noise');

% Applying the noise to fft (essientally creating the window effect somewhat)
%subplot(3,1,2);
%plot(f, y);
%title('Appling FFT to the noise');
%xlabel('Frequency');
%ylabel('noise after applying Nfft');

% ploting the wpleh with the resolution of 128 samples and no overlap
subplot(3,1,2);
plot(f,Pxx);
title('No overlap Hamming and higher resolution');
xlabel('Frequency');
ylabel('Power spectra Density');

% ploting the wpleh with the resolution of 128 samples and no overlap
subplot(3,1,3);
plot(f2,Pxx2);
title('50% overlap Hamming and lower resolution');
xlabel('Frequency');
ylabel('Power spectra Density');
```

```
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
```



Problem 4

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART A %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART B %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load the data
load SleepEEGdataPS4-1.mat;

%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART C %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Creating the list size
N1 = length(Epoch(16).Data(:,4));
N2 = length(Epoch(73).Data(:,4));


% Getting the frequency sample rate
% The frequenmcy is the same for all channels
fs = Epoch(16).Fs(4);

% Creating the frequency domain
t1 = [0:(N1-1)]*fs;
t2 = [0:(N2-1)]*fs;

% Creating the frequency domain
f1 = [0:(N1-1)]/fs;
f2 = [0:(N2-1)]/fs;


% Gathering the data from the specific Epoch Data
% Transposing the vector
dat16 = Epoch(16).Data(:,4);
dat39 = Epoch(39).Data(:,4);
dat73 = Epoch(73).Data(:,4);
dat147 = Epoch(147).Data(:,4);

% Creating an figure
figure(5);
plot(dat16, dat39, dat73, dat147);
title('Epoch Data all four channels ');
xlabel('time');
ylabel('EEG Signal');

% Creating another figure and plot each EEG
figure(6);

% Plotting the four data sets
subplot(4,1,1);
plot(dat16);
title('Epoch Data 16 ');
xlabel('time');
ylabel('EEG Signal');

subplot(4,1,2);
plot(dat39);
title('Epoch Data 39 ');
xlabel('time');
ylabel('EEG Signal');

subplot(4,1,3);
plot(dat73);
title('Epoch Data 73 ');
xlabel('time');
ylabel('EEG Signal');
```

```
subplot(4,1,4);
plot(dat147);
title('Epoch Data 147 ');
xlabel('time');
ylabel('EEG Signal');

%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART D %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('By looking at the plot its just a blob in figure 5\n');
fprintf('however, when plotting each chanel its more clear, figure 6 \n');
fprintf('So no, I cant determine the peaks by looking at the graph \n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART E %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Creating the resolution
Nfft = 2^11;

% Getting the length
L = 2^10;

% Creating how many windows on a vector of ones
win = ones(round(L/10), 1);

% Getting the length of the window
Lwin = round(L/10);

% Creating the window
win1 = window(@hamming, Lwin);

% Creating the overlap in this case its 0%
Noverlap = 0;

% pwelch for different overlap and resolution
[Pxx, f] = pwelch(dat16, win, Noverlap, Nfft, fs );
[Pxx2, f2] = pwelch(dat16, win1, Noverlap, Nfft, fs );

% Creating the figure
figure(7);

subplot(2,1,1);
plot(f,Pxx);
title('Rectangular windows power spectra density');
xlabel('Frequency');
```

```
ylabel('Power spectra density');

subplot(2,1,2);
plot(f2,Pxx2);
title('Hamming windows power spectra density');
xlabel('Frequency');
ylabel('Power spectra density');


%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART F %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('When reviewing the data set, hamming seems to be the most effective  \n');
fprintf(' Due to the smoothness of the graph \n');
fprintf('I saw 4 peaks for retangular and 2 peaks for hamming');

%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART G %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Creating how many windows on a vector of ones
win = ones(round(L/25), 1);

% Getting the length of the window
Lwin = round(L/25);

% Creating the window
win1 = window(@hamming, Lwin);

% Creating the overlap in this case its 0%
Noverlap = 0;

% pwelch for different overlap and resolution
% Increasing the window from 10 to 15
[Pxx3, f3] = pwelch(dat16, win, Noverlap, Nfft, fs );
[Pxx4, f4] = pwelch(dat16, win1, Noverlap, Nfft, fs );

% Creating the figure
figure(8);

subplot(2,1,1);
plot(f3,Pxx3);
title('Rectangular windows power spectra density with 25 windows');
xlabel('Frequency');
ylabel('Power spectra density');
```

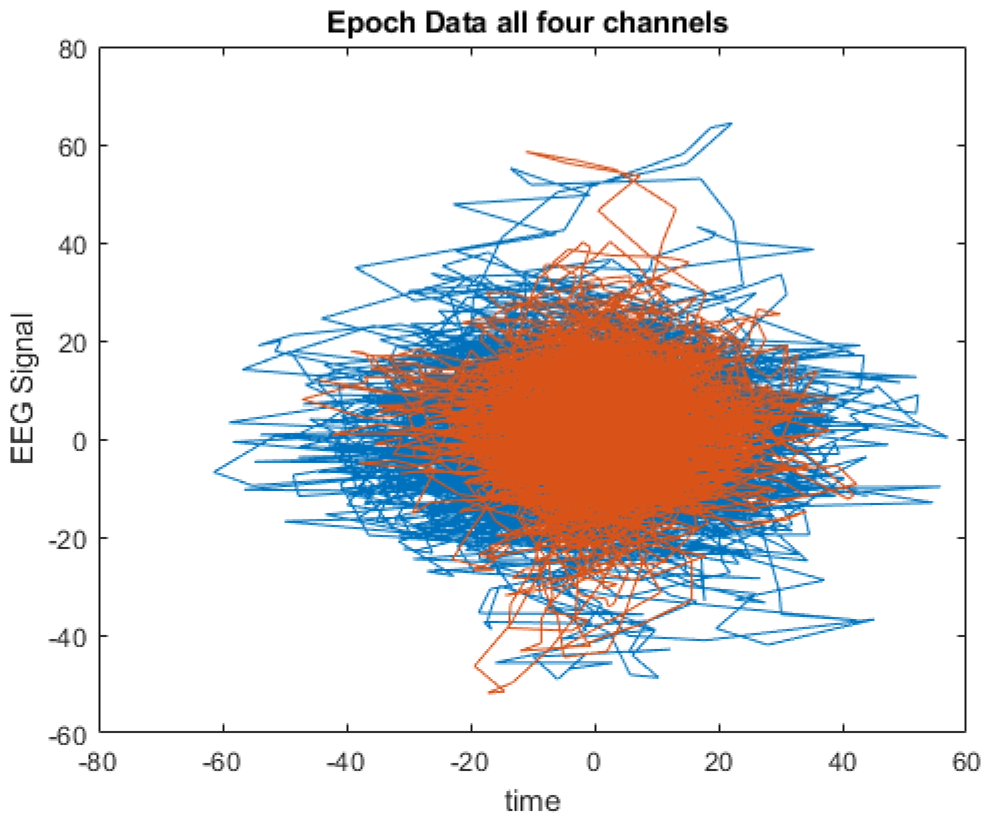
```
subplot(2,1,2);
plot(f4,Pxx4);
title('Hamming windows power spectra density with 25 windows');
xlabel('Frequency');
ylabel('Power spectra density');

fprintf('By increasing the window, the peaks reduced and the smoothes or resolution increased
\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

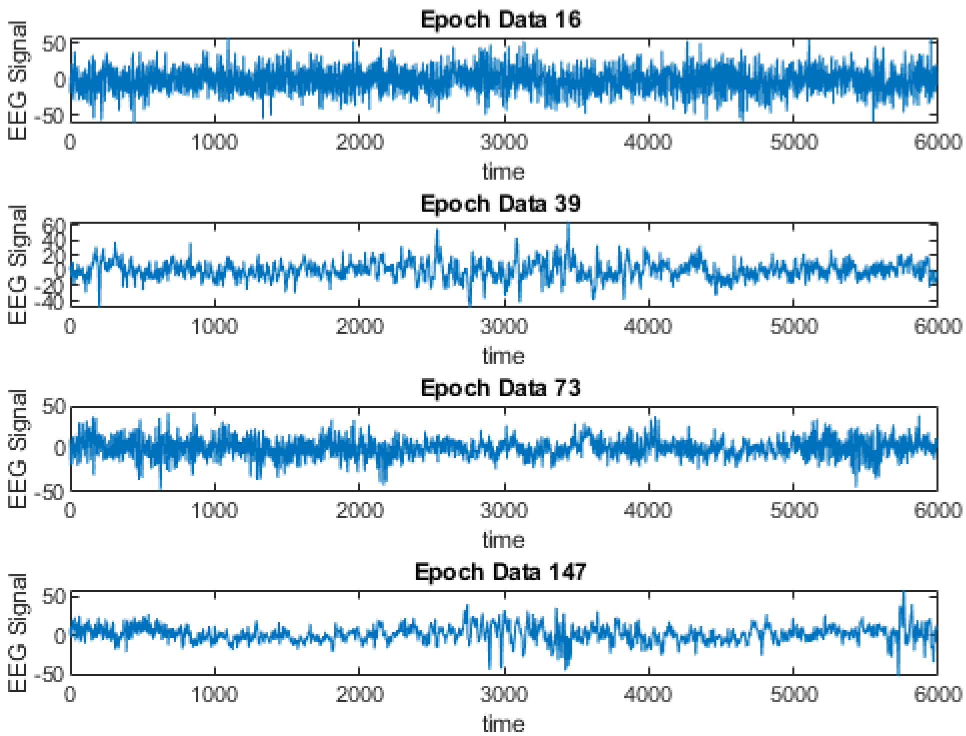
PART H %%%

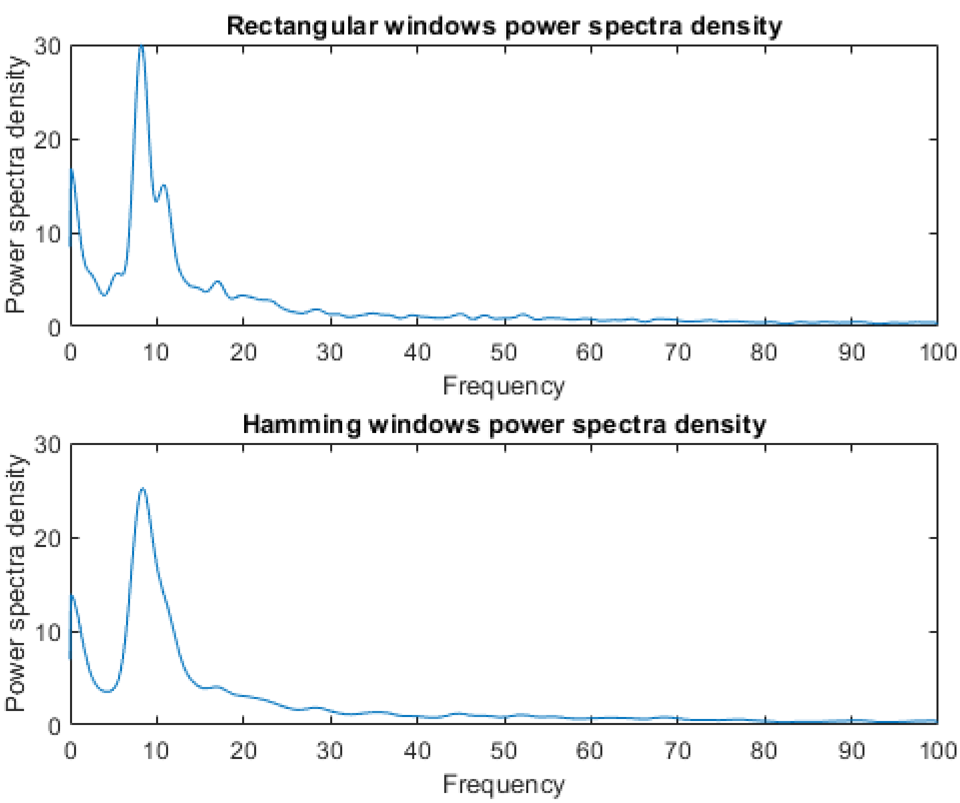
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

dat16 = Epoch(16).Data(:,4);
dat18 = Epoch(18).Data(:,4);
dat19 = Epoch(19).Data(:,4);
dat33 = Epoch(33).Data(:,4);
dat39 = Epoch(39).Data(:,4);
dat41 = Epoch(41).Data(:,4);
dat43 = Epoch(43).Data(:,4);
dat44 = Epoch(44).Data(:,4);
dat73 = Epoch(73).Data(:,4);
dat147 = Epoch(147).Data(:,4);
dat148 = Epoch(148).Data(:,4);
dat634= Epoch(634).Data(:,4);

% Creating how many windows on a vector of ones
win = ones(round(L/25), 1);

% Getting the length of the window
Lwin = round(L/25);

% Creating the window
win1 = window(@hamming, Lwin);

% Creating the overlap in this case its 0%
Noverlap = 0;

% pwelch for different overlap and resolution
% Increasing the window from 10 to 15
[Pxx5, f5] = pwelch(dat16, win, Noverlap, Nfft, fs );
[Pxx6, f6] = pwelch(dat18, win1, Noverlap, Nfft, fs );
[Pxx7, f7] = pwelch(dat19, win, Noverlap, Nfft, fs );
[Pxx8, f8] = pwelch(dat33, win1, Noverlap, Nfft, fs );
[Pxx9, f9] = pwelch(dat39, win, Noverlap, Nfft, fs );
[Pxx10, f10] = pwelch(dat41, win1, Noverlap, Nfft, fs );
[Pxx11, f11] = pwelch(dat43, win, Noverlap, Nfft, fs );
[Pxx12, f12] = pwelch(dat44, win1, Noverlap, Nfft, fs );
[Pxx13, f13] = pwelch(dat73, win, Noverlap, Nfft, fs );
[Pxx14, f14] = pwelch(dat147, win1, Noverlap, Nfft, fs );
```

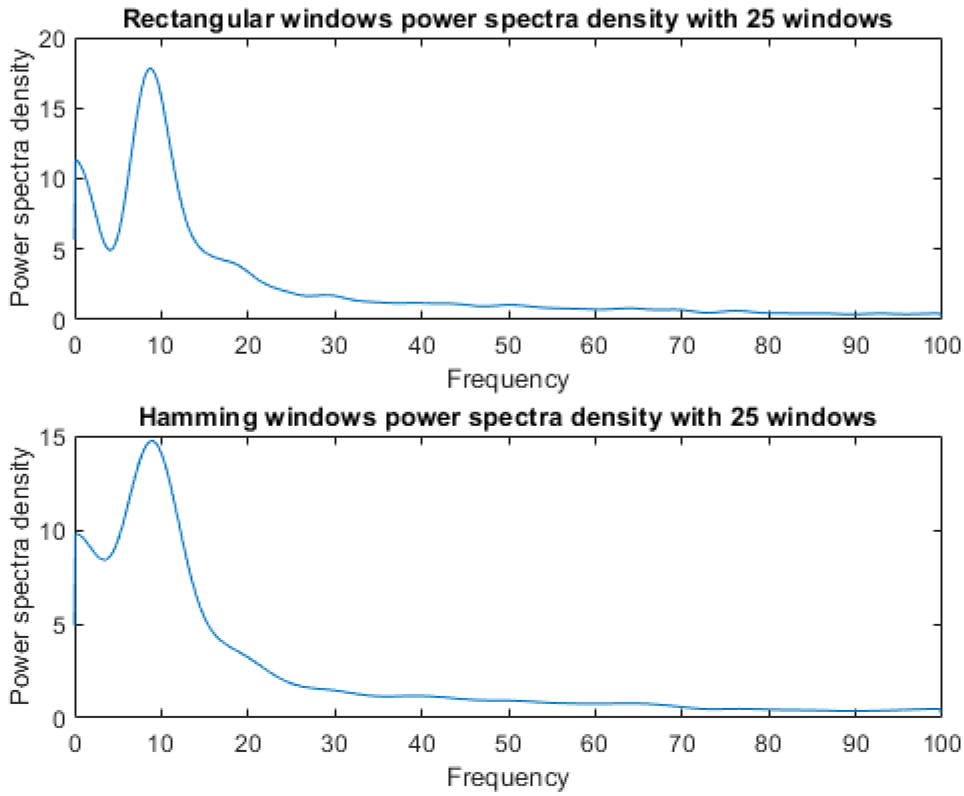
```
[Pxx15, f15] = pwelch(dat148, win, Noverlap, Nfft, fs );
[Pxx16, f16] = pwelch(dat634, win1, Noverlap, Nfft, fs );
```

```
By looking at the plot its just a blob in figure 5
however, when plotting each chanel its more clear, figure 6
So no, I cant determine the peaks by looking at the graph
When reviewing the data set, hamming seems to be the most effective
 Due to the smoothness of the graph
I saw 4 peaks for retangular and 2 peaks for hammingBy increasing the window, the peaks reduced
and the smoothes or resolution increased
```




Epoch Data 16
EEG Signal
time
Epoch Data 39
EEG Signal
time
Epoch Data 73
EEG Signal
time
Epoch Data 147
EEG Signal
time


Rectangular windows power spectra density
Power spectra density
Frequency
Hamming windows power spectra density
Power spectra density
Frequency



Published with MATLAB® R2021b