```matlab
% file SIOC 221A HW 5
%
% author Julia Dohner
%
% due date November 2, 2017

clear all; close all;

%% Compute two spectra - attempt via Lecture 7 Notes

% compute two spectra for white noise and autoregressive datasets by
% breaking the data up into segments

N = 10000; % length of each chunk of data
M = 20; % number of segments splitting data into
p = N/M; % datapoints per segment

% generating 10,000 element dataset with Gaussian white noise
a=randn(N,1);
b(1) = a(1);

% generating a second dataset using autoregressive process
for i=2:length(a)
    b(i)=.5*b(i-1)+a(i);
end

a_reshape = reshape(a,N/M,M);
b_reshape = reshape(b,N/M,M);
a_f = fft(a_reshape);
b_f = fft(b_reshape);

a_amp = abs(a_f(1:p/2+1,:)).^2; % aplitude of a = abs(f_t(from 1 to 500/2+1, all columns)^2
a_amp(2:p/2,:) = 2*a_amp(2:p/2,:)/p; % normalizing values in amp_a (except for 0)
a_amp = a_amp(2:251,:); % dumping the mean
a_amp = a_amp(1:249,:); % dumping last value
b_amp=abs(b_f(1:p/2+1,:)).^2;
b_amp(2:p/2,:)=2*b_amp(2:p/2,:)/p;
b_amp = b_amp(2:251,:); % dumping the mean
b_amp = b_amp(1:249,:); % dumping last value

frequency=(0:p/2-2)/p; % for N even
a_amp_mean = mean(a_amp,2);
b_amp_mean = mean(b_amp,2);

figure
semilogy(frequency,a_amp_mean, '-r', frequency, b_amp_mean, '-b')
xlabel('\fontsize{14}frequency')
ylabel('\fontsize{14}units')
title('\fontsize{16}Fourier Transform of Gaussian White Noise and Autoregressive Datasets')
legend('\fontsize{12}Gaussian white noise','Autoregressive data');


%% add error bars to your spectra

nu = 2*floor(N/p);
err_high = nu/chi2inv(0.05/2,nu);
err_low = nu/chi2inv(1-0.05/2,nu);
ratio_chi2 = err_high/err_low;
```

```matlab
figure(1)
semilogy(1:p/2-1, a_amp_mean, '-b', [p/4 p/4],[err_low err_high]*a_amp_mean(p/4), '-r')
xlabel('\fontsize{14}frequency')
ylabel('\fontsize{14}units')
title('\fontsize{16}Fourier Transform of Gaussian White Noise')
legend('\fontsize{12}Gaussian white noise','\chi^{2}-computed uncertainty');

figure(2)
semilogy(1:p/2-1, b_amp_mean, '-r', [p/4 p/4],[err_low err_high]*b_amp_mean(p/4), '-b')
xlabel('\fontsize{14}frequency')
ylabel('\fontsize{14}units')
title('\fontsize{16}Fourier Transform of Autoregressive Data')
legend('\fontsize{12}autoregressive data','\chi^{2}-computed uncertainty');

%% monte carlo simulation - Gaussian

% bigMatrix is a matrix (500x4,000) containing 200 500x20 matrices
bigMatrix_0 = randn(2000000,1);
bigMatrix = reshape(bigMatrix_0,500,4000);
%bigMatrix = randn(500,4000);
bigMatrix = fft(bigMatrix); % compute fft of each matrix

bigMatrix_amp=(abs(bigMatrix(1:p/2+1,:)).^2)/p; %needed to move the /p to this line from line↙
below
bigMatrix_amp(2:p/2,:)=2*bigMatrix_amp(2:p/2,:);
bigMatrix_amp = bigMatrix_amp(2:251,:); % dumping the mean


for i=1:20:4000
    bigMatrix_mean_amp(:,floor(i/20)+1) = mean(bigMatrix_amp(:,i:i+19),2);
end

%turn mean amplitude result into a column vector
bigMatrix_pdf = bigMatrix_mean_amp(:);
bigMatrix_pdf = sort(bigMatrix_pdf,'descend');
err_low_data = prctile(bigMatrix_pdf,0.025);
err_high_data = prctile(bigMatrix_pdf,0.975);
ratio_monteCarlo_Gaussian = err_high_data/err_low_data;

bigMatrix_pdf_hist = histogram(bigMatrix_pdf); %histogram just a way of visualizing, just need↙
info from raw values (have it once sorted)


%% repeat above steps for Gaussian white noise, but using a Hanning window

% Hanning window treatment of bigMatrix in Question #4
bigMatrix_4 = randn(500,4000);
dataHan = fft(detrend(bigMatrix_4).*(hann(500)*ones(1,4000)));

dataHan_amp=(abs(dataHan(1:p/2+1,:)).^2)/p;
dataHan_amp(2:p/2,:)=2*dataHan_amp(2:p/2,:);
dataHan_amp = dataHan_amp(2:251,:); % dumping the mean

for i=1:20:4000 %4000-19
    dataHan_mean_amp(:,floor(i/20)+1) = mean(dataHan_amp(:,i:i+19),2);
end

%turn mean amplitude result into a column vector
dataHan_pdf = dataHan_mean_amp(:);
```

```matlab
dataHan_pdf = sort(dataHan_pdf,'descend');
err_low_data = prctile(dataHan_pdf,0.025);
err_high_data = prctile(dataHan_pdf,0.975);
ratio_Hann = err_high_data/err_low_data;

figure(3)
dataHan_pdf_hist = histogram(dataHan_pdf);
xlabel('\fontsize{14}units')
ylabel('\fontsize{14}probability')
title('\fontsize{16}PDF of Gaussian white noise data with Hanning window applied')
```