Table of Contents

Neural Time-Series Analysis Lab

By: Elizabeth Nemeti Date:Oct 28 22

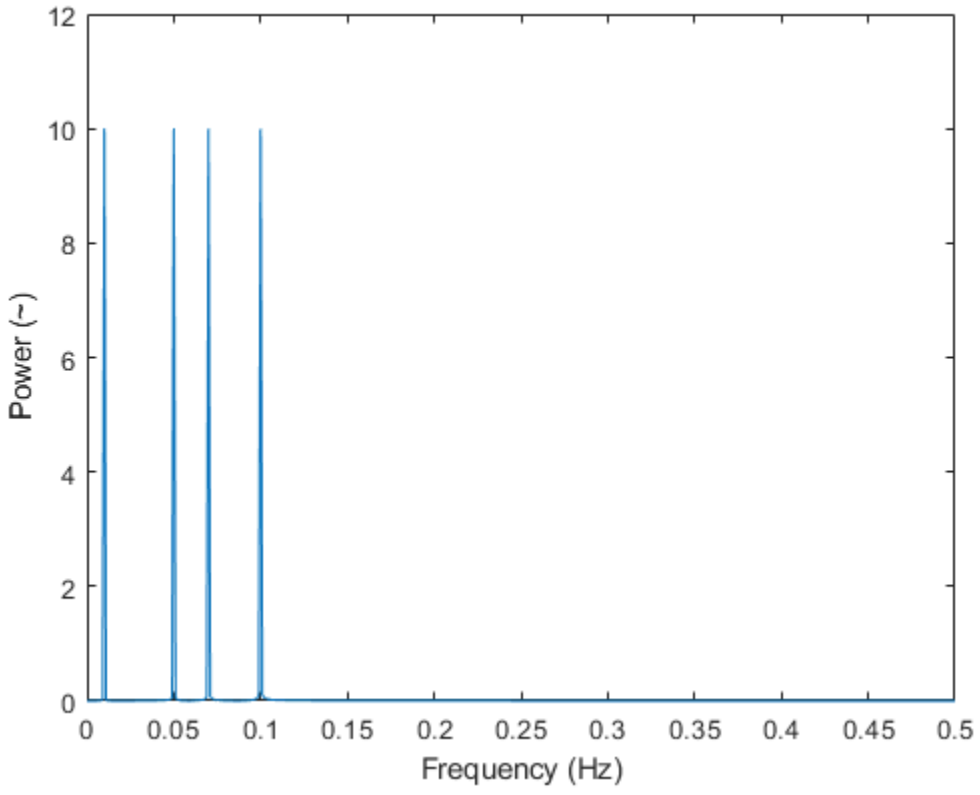
```
clc
clear
close all
```

Question A

Generate a time series by creating and summing sine waves, as in figure 1B. Use four sine waves, so that the individual sine waves are still somewhat visible in the sum. Perform a Fourier analysis on the resulting time series and plot the power structure. Confirm that your code is correct by comparing the frequencies with nonzero power to the frequencies of the sine waves that you generated.

```
% recreating figure A
res = 10000;
time = linspace(0, 1000, res);

sin_1 = sin(time* 2*pi*0.1);
sin_2 = sin(time* 2*pi*0.07);
sin_3 = sin(time* 2*pi*0.05);
sin_4 = sin(time* 2*pi*0.01);

t = tiledlayout(4,1);
ylabel(t, "Amplitude")
xlabel(t, "Time (ms)")

nexttile
plot(time,sin_1)

nexttile
plot(time,sin_2)

nexttile
plot(time,sin_3)

nexttile
plot(time,sin_4)

saveas(t, "QA_plot1.png")
```

```
% creating plot b
combine_sin = sin_1 + sin_2 + sin_3 + sin_4; % add the sine curves

figure(2)
plot(time, combine_sin)
ylabel("Amplitude")
xlabel("Time (ms)")
saveas(figure(2), "QA_sum_sin.png")

% performing Fourier analyis
L=1000; % 1000ms for length
fs = 1;
[f, P1] = make_FFT(L, combine_sin, fs);

% plotting the power structure
figure(3)
plot(f,P1)
xlabel("Frequency (Hz)")
ylabel("Power (~)")
saveas(figure(3), "QA_fft.png")
```

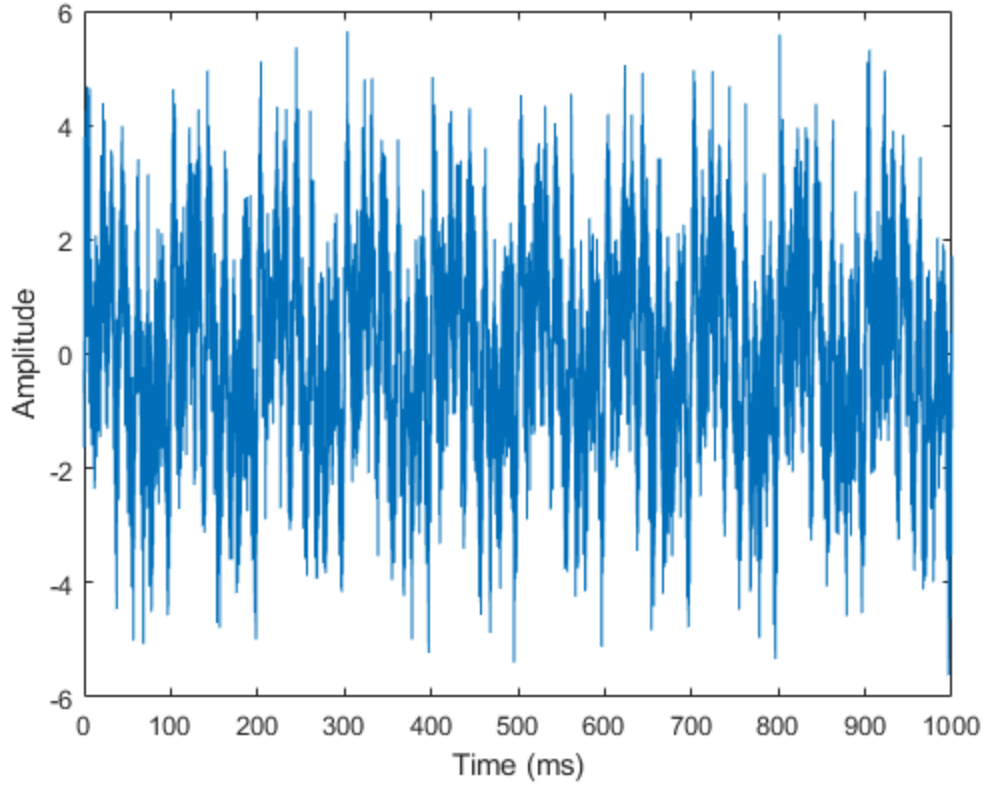


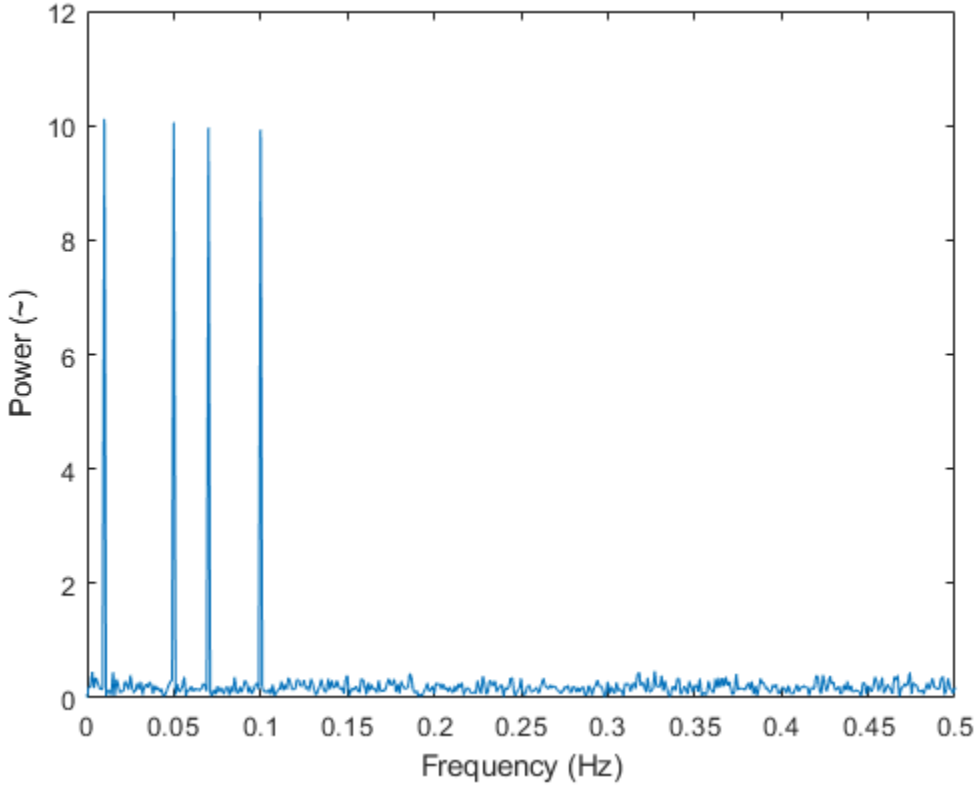
Question B

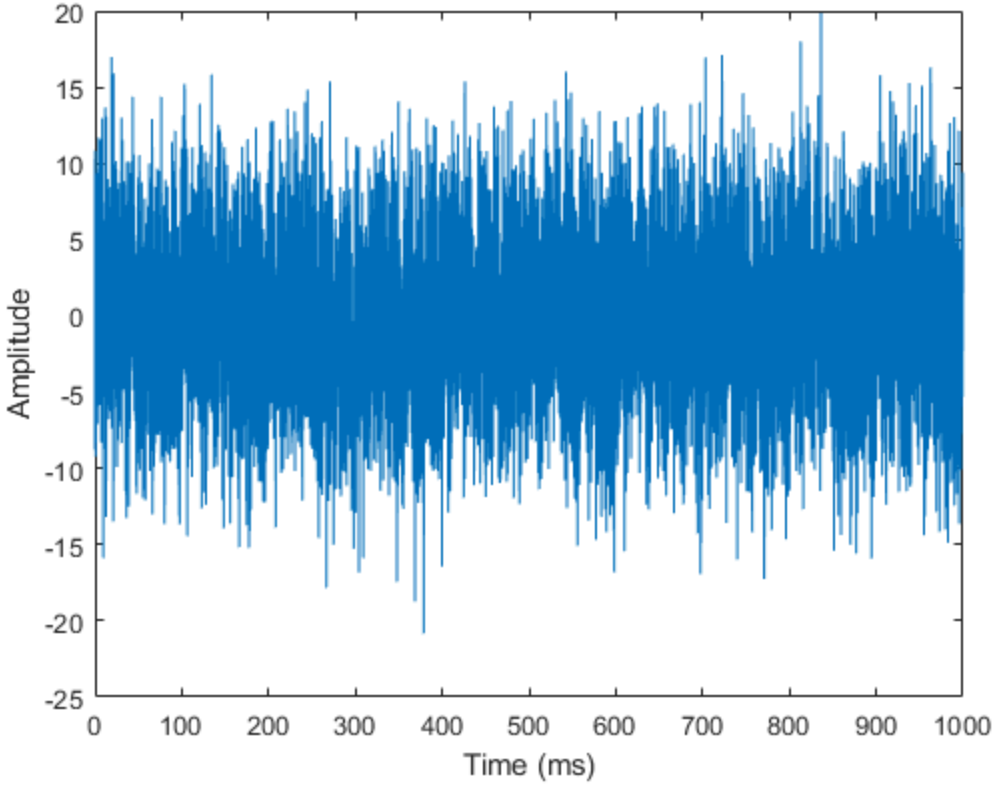
Try adding random noise to the signal before computing the Fourier transform. First, add a small amount of noise so that the sine waves are still visually recognizable. Next, add a large amount of noise so that the sine waves are no

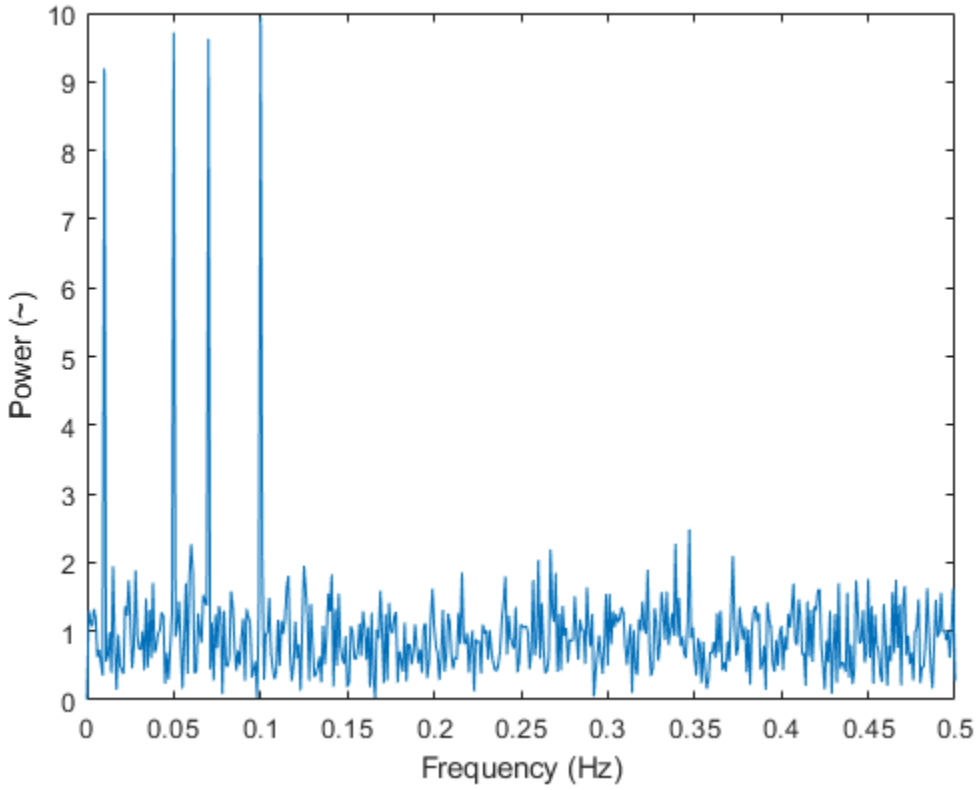
longer visually recognizable in the time domain data. Perform a Fourier analysis on the two noisy signals and plot the results. What is the effect of a small and a large amount of noise in the power spectrum? Are the sine waves with noise easier to detect in the time domain or in the frequency domain, or is it equally easy/difficult to detect a sine wave in the presence of noise?

```
X = combine_sin + 1*randn(size(time)); % adding noise corruption (small)

figure(4)
plot(time, X)
ylabel("Amplitude")
xlabel("Time (ms)")
saveas(figure(4), "QB_sins_small.png")


[f, P1] = make_FFT(L, X, fs); % fourier transform on small noise corruption

figure(5)
plot(f,P1)
xlabel("Frequency (Hz)")
ylabel("Power (~)")
saveas(figure(5), "QB_fft_small.png")


X = combine_sin + 5*randn(size(time)); % noise corruption (larger)

figure(6)
plot(time, X)
ylabel("Amplitude")
xlabel("Time (ms)")
saveas(figure(6), "QB_sins_big.png")


[f, P1] = make_FFT(L, X, fs); % fourier transform on larger noise corruption

figure(7)
plot(f,P1)
xlabel("Frequency (Hz)")
ylabel("Power (~)")
saveas(figure(7), "QB_fft_big.png")
```


Amplitude
Time (ms)
6
4
2
0
-2
-4
-6
0
100
200
300
400
500
600
700
800
900
1000


Power (~)
Frequency (Hz)
12
10
8
6
4
2
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

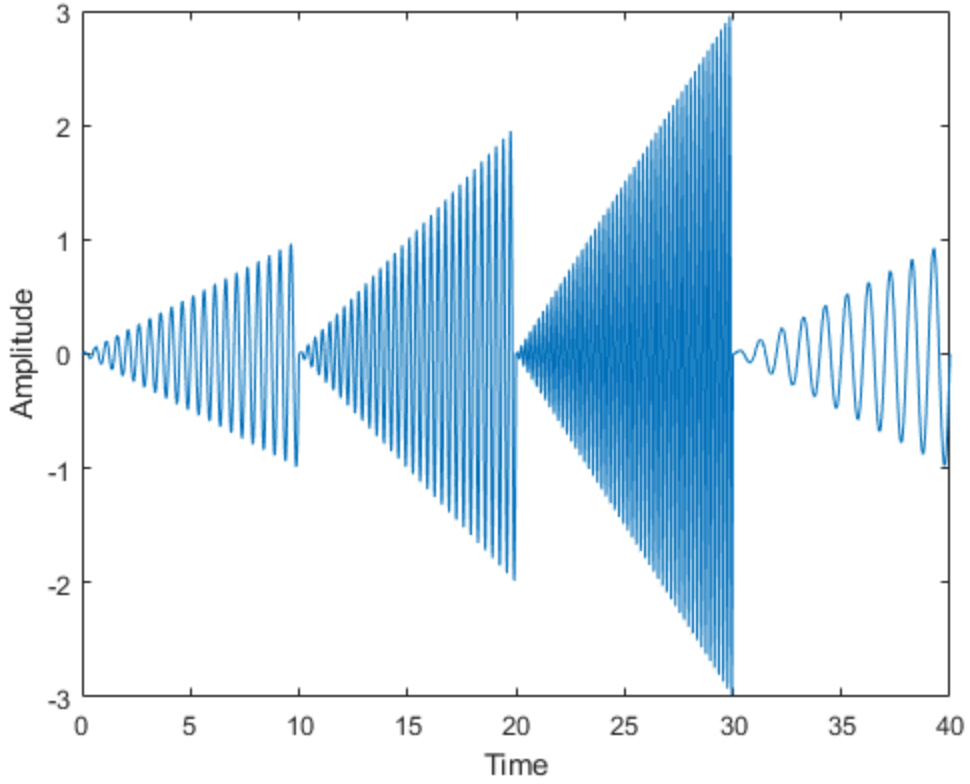

Amplitude
Time (ms)
20
15
10
5
0
-5
-10
-15
-20
-25
0
100
200
300
400
500
600
700
800
900
1000

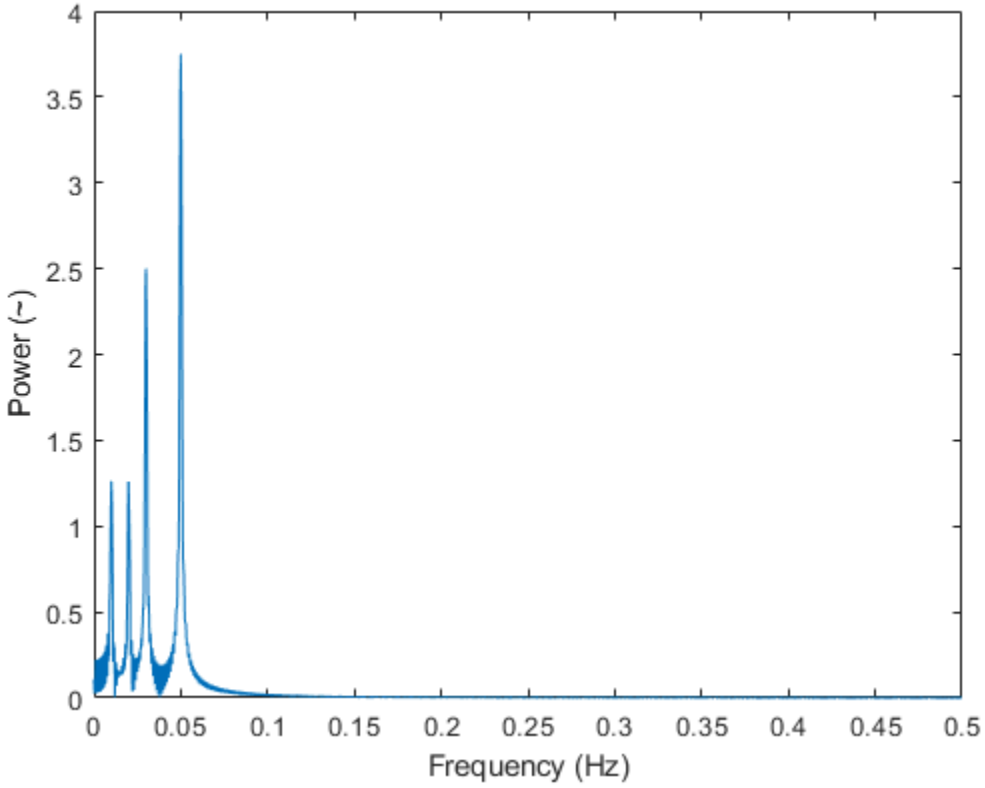

Power (~)
Frequency (Hz)
10
9
8
7
6
5
4
3
2
1
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Question C

create a nonstationary time series (similar to the waveform in Figure 2) using the sine waves from section A, perform the Fourier analysis on the entire signal and plot the results.

```
amplitude_1 = 0:0.0001:1;
amplitude_2 = 0:0.0002:2;
amplitude_3 = 0:0.0003:3;
time = 0:0.001:40;

series_1 = amplitude_1 .* sin(2*pi*2*time(1:10001));
series_2 = amplitude_2(1:end-1) .* sin(2*pi*3*time(10002:20001));
series_3 = amplitude_3(1:end-1) .* sin(2*pi*5*time(20002:30001));
series_4 = amplitude_1(1:end-1) .* sin(2*pi*1*time(30002:40001));

complete_series = [series_1, series_2, series_3, series_4]; % creating entire
 signal

figure(8)
plot(time, complete_series)
xlabel("Time")
ylabel("Amplitude")
saveas(figure(8), "QC_combine_sin.png")

L=4000;
fs = 1;
[f, P1] = make_FFT(L, complete_series, fs); % performing fourier transform
figure(9)
plot(f(1:end),P1(1:end))
xlabel("Frequency (Hz)")
ylabel("Power (~)")
saveas(figure(9), "QC_fft.png")
```

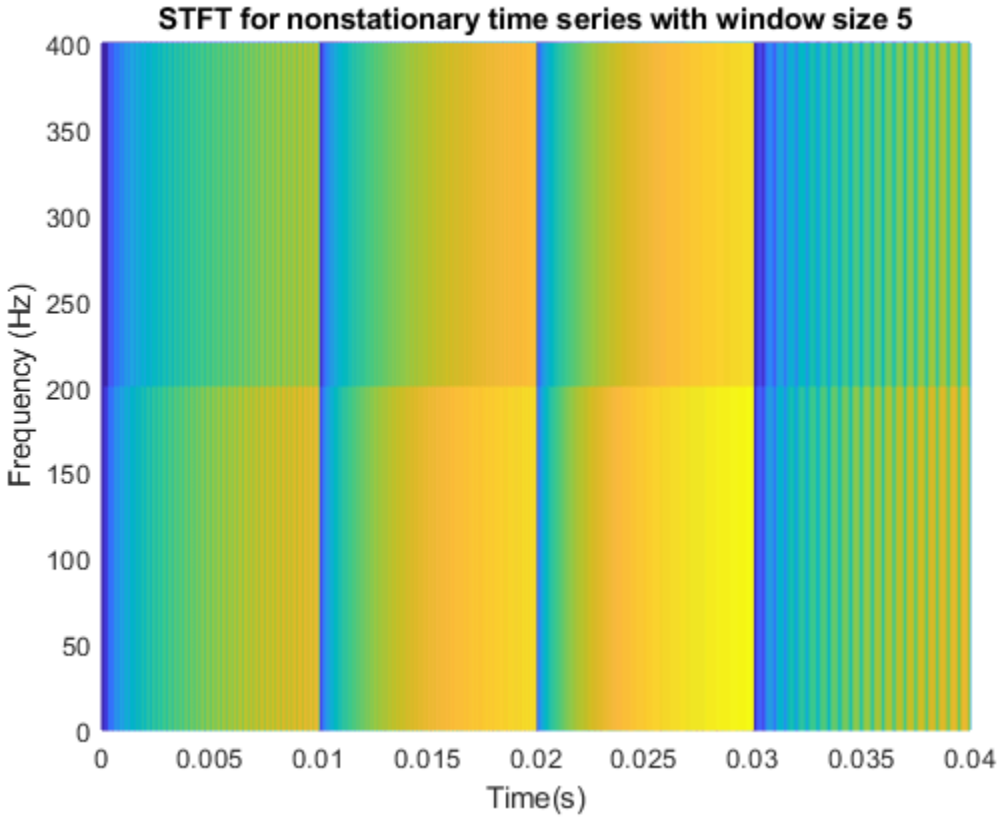

Amplitude
Time
3
2
1
0
-1
-2
-3
0
5
10
15
20
25
30
35
40

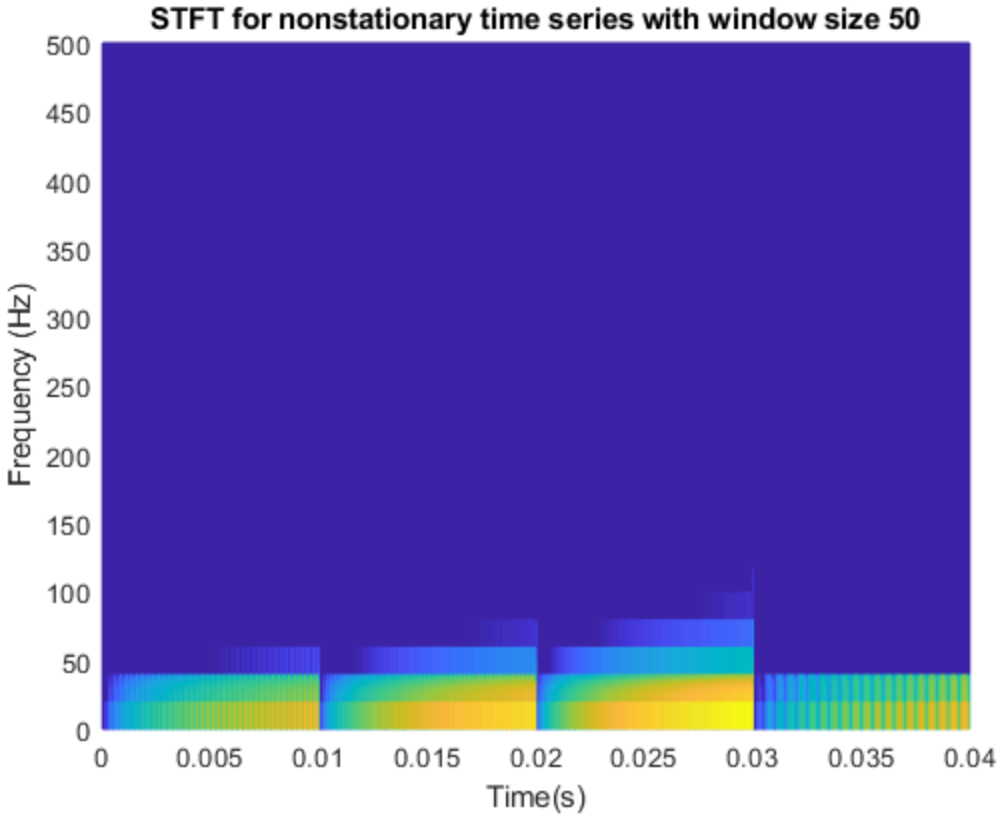

Power (~)
Frequency (Hz)
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

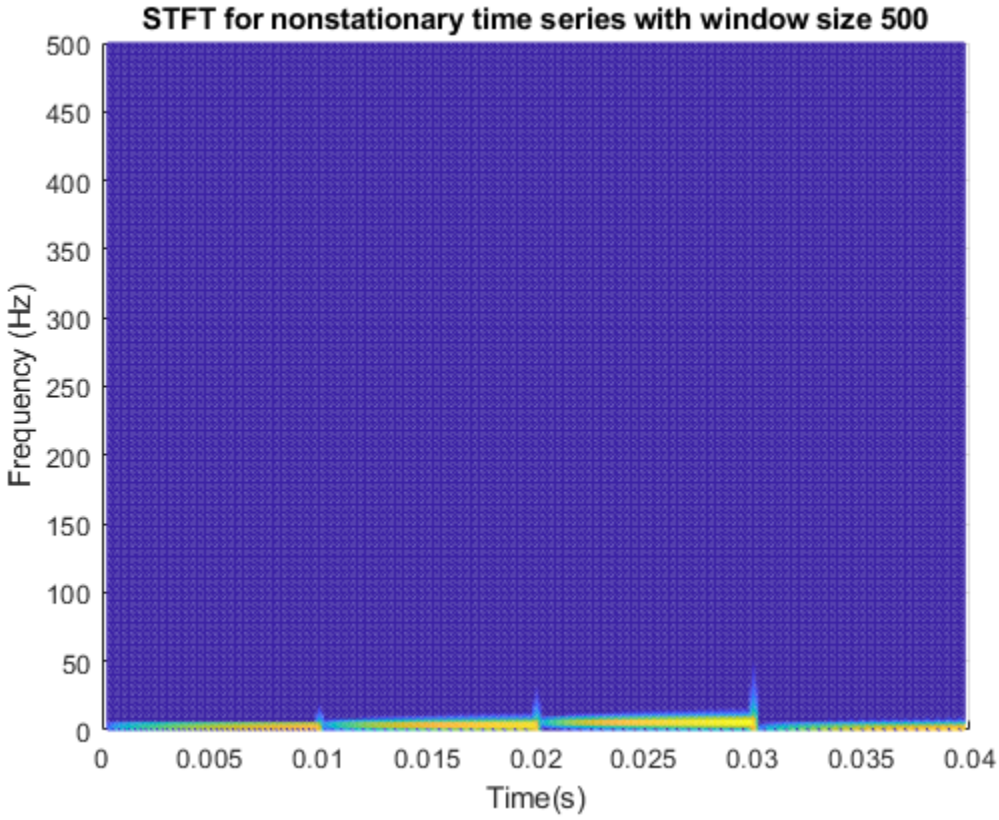
Question D

Perform Short-Term Fourier Transform on the entire signal from section C and plot the results. Experiment with different window sizes and compare the results.

```
stft_D(10e5, complete_series, 5, 50)
saveas(figure(10), "D_5.png")
stft_D(10e5, complete_series, 50, 50)
saveas(figure(11), "D_50.png")
stft_D(10e5, complete_series, 500, 50)
saveas(figure(12), "D_500.png")
```




STFT for nonstationary time series with window size 50
Frequency (Hz)
500
450
400
350
300
250
200
150
100
50
0
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
Time(s)


STFT for nonstationary time series with window size 500
Frequency (Hz)
500
450
400
350
300
250
200
150
100
50
0
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
Time(s)

Question E

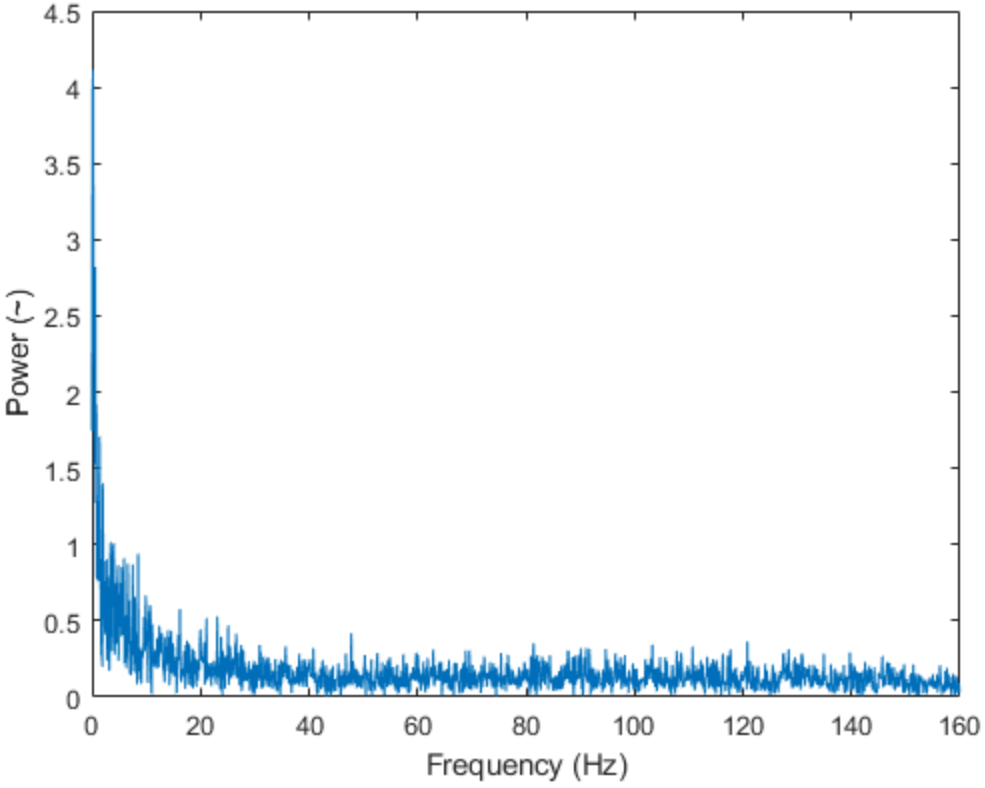
‘dataset.mat’ contains data from real EEG recordings. This dataset is a concatenation of multiple recording segments (trials). Repeat the analysis in sections B and D using the EEG dataset. How many trials are included in this dataset?

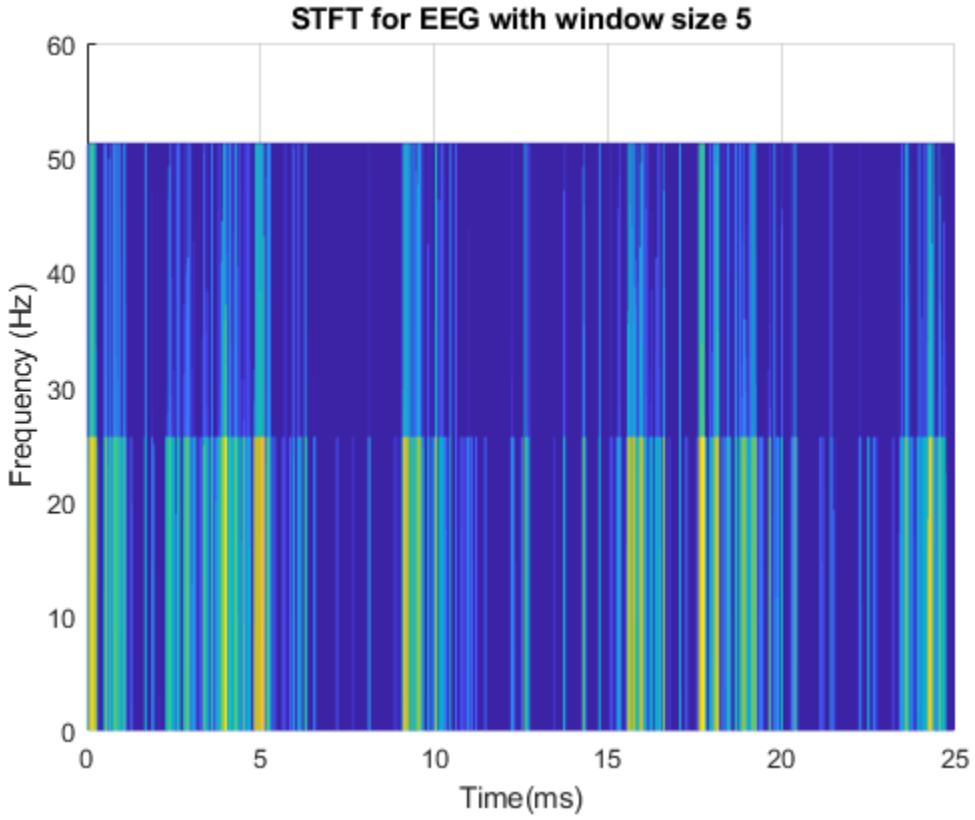
```
load dataset.mat

% figure(69)
% plot(1:3200, eeg)

L=3200;
fs = 320;
[f, P1] = make_FFT(L, eeg, fs);
figure(13)
plot(f,P1)
xlabel("Frequency (Hz)")
ylabel("Power (~)")
saveas(figure(13), "E_FFT.png")
snapnow


stft_eeg(eeg, 128, 5, 10)
saveas(figure(14), "E_5.png")
stft_eeg(eeg, 128, 50, 10)
saveas(figure(15), "E_50.png")
stft_eeg(eeg, 128, 500, 10)
saveas(figure(16), "E_500.png")



% saveas(figure(14), "QC_fft.png")
```

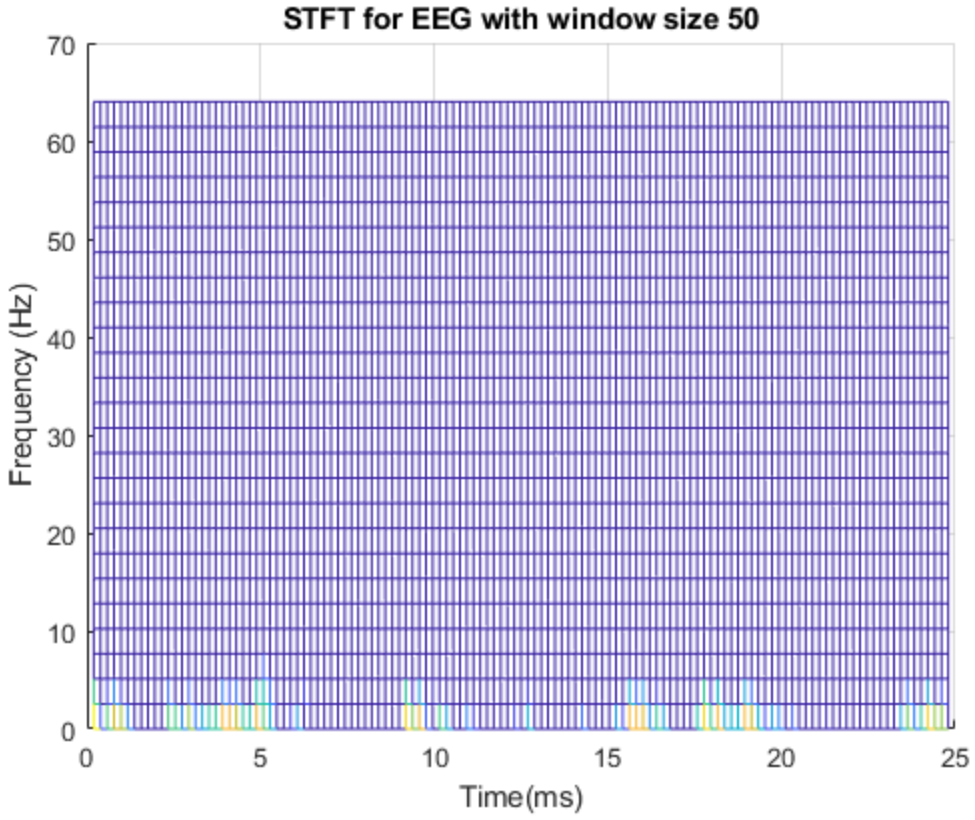
Functions

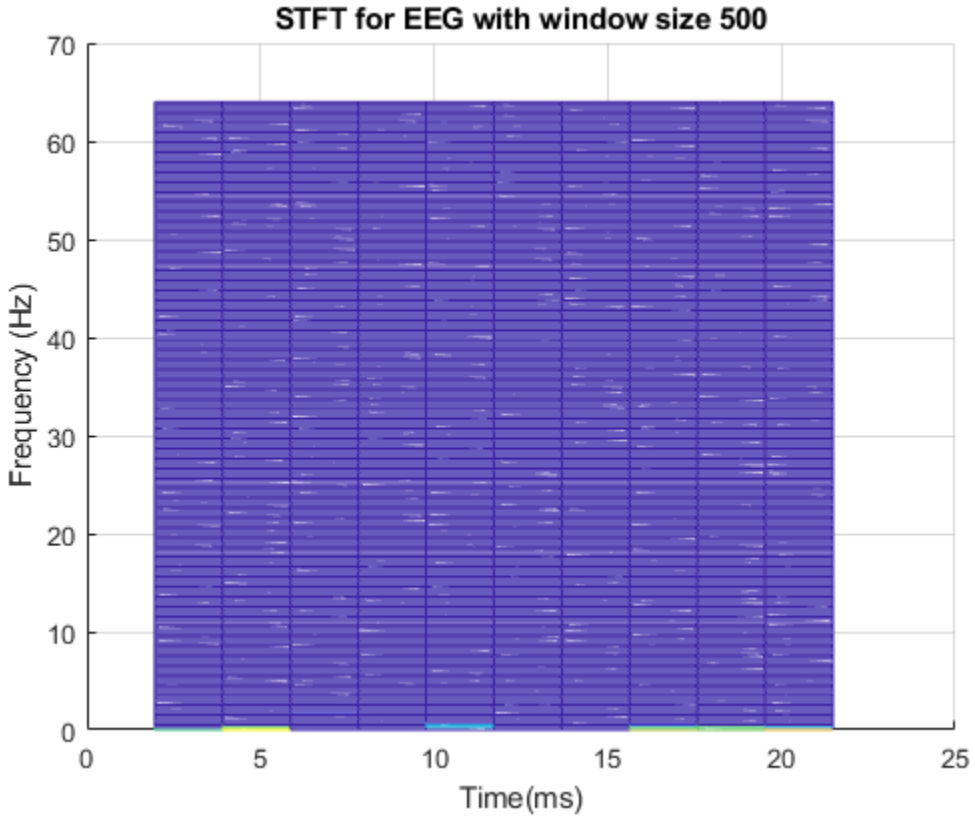
```
function [x, y]  = make_FFT(L, signal, fs)


fourier_sum_sin = fft(signal);
P2 = abs(fourier_sum_sin/L);
y = P2(1:L/2+1);
y(2:end-1) = 2*y(2:end-1);
x = fs*(0:(L/2))/L;

end

function stft_D(fs, series, window_size, color_range)

[s,f,t] = stft(series, fs, Window=hann(window_size),
 FrequencyRange="onesided");
sdb = mag2db(abs(s));

figure
mesh(t,f/1000,sdb);
```

```
cc = max(sdb(:))+[-color_range 0];
ax = gca;
ax.CLim = cc;
title(['STFT for nonstationary time series with window size '
 num2str(window_size)])
xlabel("Time(s)")
ylabel("Frequency (Hz)")
view(2)

end

function stft_eeg(series,fs, window_size, color_range)

[s,f,t] = stft(series, fs, Window=hann(window_size),
 OverlapLength=int64(window_size/2), FrequencyRange="onesided");
sdb = mag2db(abs(s));

figure
mesh(t,f,sdb);

cc = max(sdb(:))+[-color_range 0];
ax = gca;
ax.CLim = cc;
xlabel("Time(ms)")
ylabel("Frequency (Hz)")
title(['STFT for EEG with window size ' num2str(window_size)])
view(2)


end
```


Power (~)
Frequency (Hz)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
0
20
40
60
80
100
120
140
160


STFT for EEG with window size 5
Frequency (Hz)
Time(ms)
0
10
20
30
40
50
60
0
5
10
15
20
25


STFT for EEG with window size 50
Frequency (Hz)
Time(ms)
0
10
20
30
40
50
60
70
0
5
10
15
20
25


STFT for EEG with window size 500
Frequency (Hz)
Time(ms)
0
10
20
30
40
50
60
70
0
5
10
15
20
25

Published with MATLAB® R2022b