```
# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt

plt.close('all')

d1 = np.genfromtxt('noise1_trial.csv',delimiter=',',skip_header=1)
d2 = np.genfromtxt('noise2_trial.csv',delimiter=',',skip_header=1)
d3 = np.genfromtxt('noise3_trial.csv',delimiter=',',skip_header=1)


#Sampling rate and Nyquist frequency
Fs = abs(1/(d1[1,0]-d1[0,0]))
nyq = Fs/2
freq = np.arange(0,nyq+Fs/len(d1[:,0]),Fs/len(d1[:,0]))
freq = [a/nyq for a in freq]

#Initial Amplitude Plots
plt.figure()
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('N1')
plt.plot(d1[:,0],d1[:,1],linewidth=0.1)
plt.savefig('N1_amp.pdf')
plt.figure()
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('N2')
plt.plot(d2[:,0],d2[:,1],linewidth=0.1)
plt.savefig('N2_amp.pdf')
plt.figure()
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('N3')
plt.plot(d3[:,0],d3[:,1],linewidth=0.1)
plt.savefig('N3_amp.pdf')

f1 = np.fft.rfft(d1[:,1] - np.average(d1[:,1]))
f2 = np.fft.rfft(d2[:,1] - np.average(d2[:,1]))
f3 = np.fft.rfft(d3[:,1] - np.average(d3[:,1]))

f1 = [a*2/len(d1[:,0]) for a in f1]
f1[0] = f1[0]/2
f1[-1] = f1[-1]/2

f2 = [a*2/len(d2[:,0]) for a in f2]
f2[0] = f2[0]/2
f2[-1] = f2[-1]/2

f3 = [a*2/len(d3[:,0]) for a in f3]
f3[0] = f3[0]/2
f3[-1] = f3[-1]/2

##Nyquist cutoff
#f1 = f1[int(len(f1)/2):len(f1)]
#f2 = f2[int(len(f2)/2):len(f2)]
```

```
#f3 = f3[int(len(f3)/2):len(f3)]



p1 = [x*np.conj(x) for x in f1]
p2 = [x*np.conj(x) for x in f2]
p3 = [x*np.conj(x) for x in f3]

plt.figure()
plt.plot(freq,p1,linewidth = 0.5)
plt.xlabel('$f$ / $f_{Nyq}$')
plt.ylabel('Power Spectrum')
plt.title('N1')
plt.savefig('N1_fft.pdf')

plt.figure()
plt.plot(freq,p2,linewidth = 0.5)
plt.xlabel('$f$ / $f_{Nyq}$')
plt.ylabel('Power Spectrum')
plt.title('N2')
plt.savefig('N2_fft.pdf')

plt.figure()
plt.plot(freq,p3,linewidth = 0.5)
plt.xlabel('$f$ / $f_{Nyq}$')
plt.ylabel('Power Spectrum')
plt.title('N3')
plt.savefig('N3_fft.pdf')
```