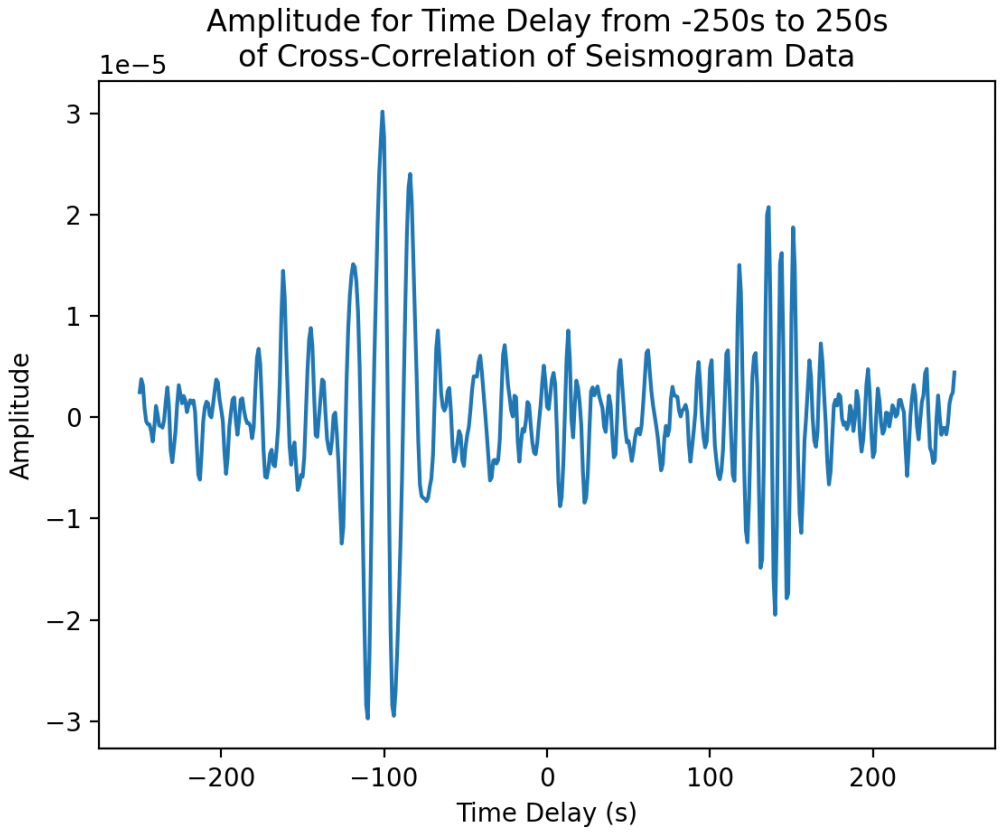
No Collaboration for any questions.

Exercise 1:

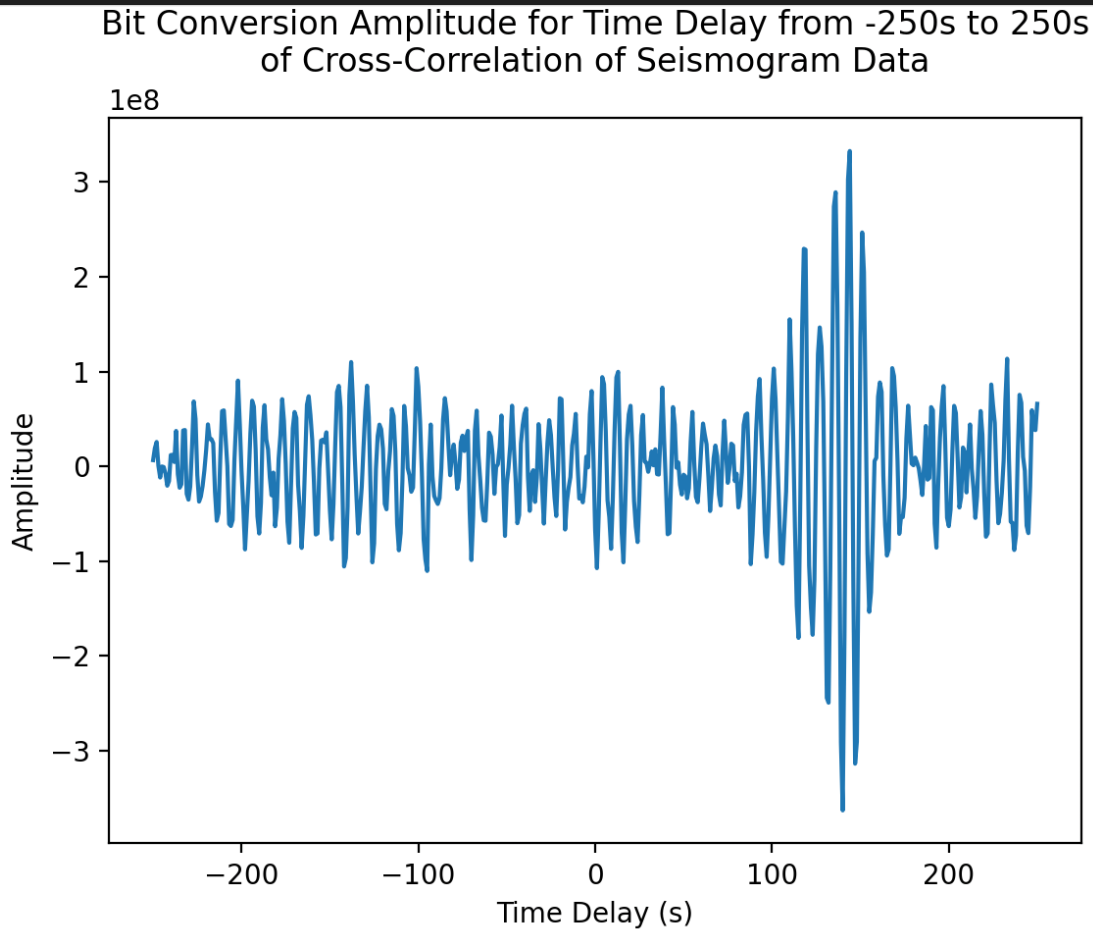
1.

```
import numpy as np
import matplotlib.pyplot as plt

tmp = np.genfromtxt("PHL_data.txt")
phl = tmp.flatten()

tmp = np.genfromtxt("MLAC_data.txt")
mlac = tmp.flatten()

zeros = np.zeros(250)

mlac = np.concatenate((zeros,mlac,zeros))

answer = [0]*501
for i in range(501):
    answer[i] = sum(np.conjugate(np.fft.fft(mlac[i:86400+i]))*np.fft.fft(phl))

print(answer)
plt.plot(np.arange(-250,251),answer)
plt.xlabel("Time Delay (s)")
plt.ylabel("Amplitude")
plt.title("Amplitude for Time Delay from -250s to 250s\nof Cross-Correlation of Seismogram Data")
plt.show()
```



2.

```
phl = np.sign(phl)
mlac = np.sign(mlac)

answer = [0]*501
for i in range(501):
    answer[i] = sum(np.conjugate(np.fft.fft(mlac[i:86400+i]))*np.fft.fft(phl))

plt.plot(np.arange(-250,251),answer)
plt.xlabel("Time Delay (s)")
plt.ylabel("Amplitude")
plt.title("Bit Conversion Amplitude for Time Delay from -250s to 250s\nof Cross-Correlation of Seismogram Data")
plt.show()
```

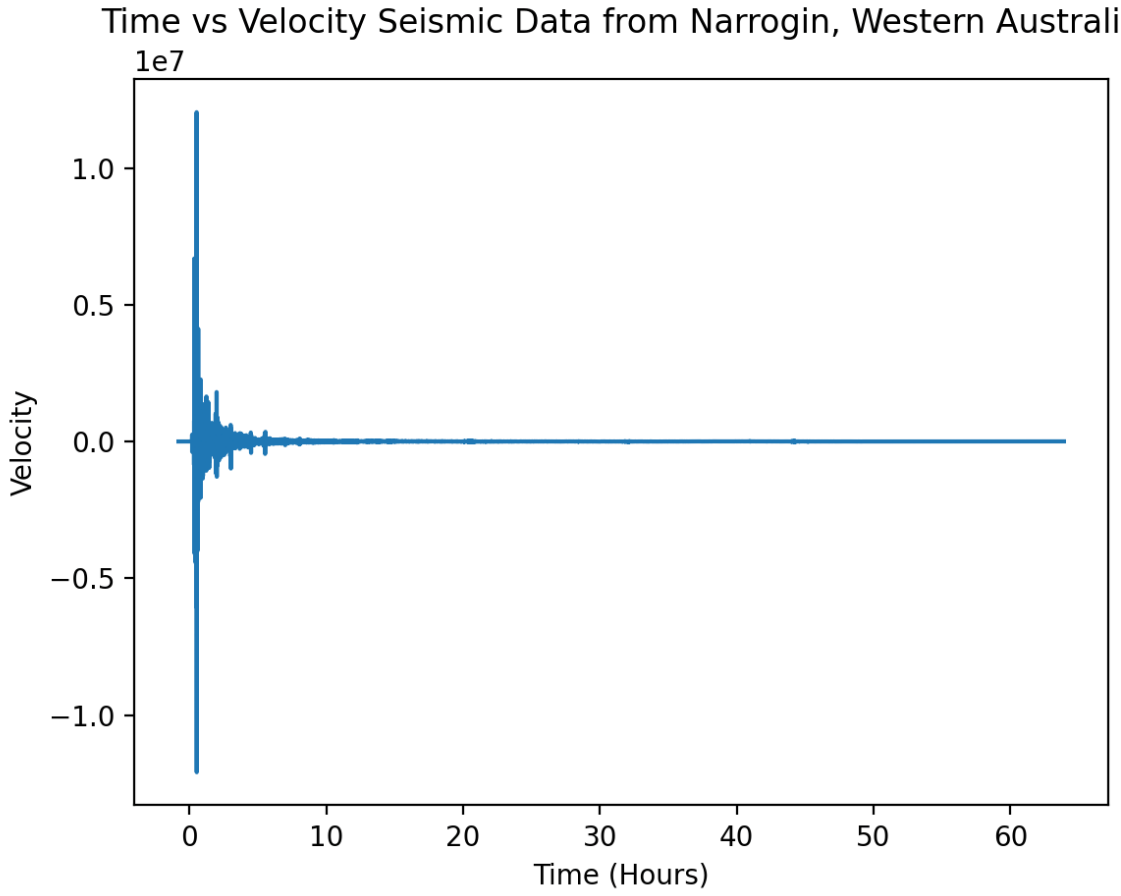


Despite a difference in amplitude, as the amplitude of the Bit-Conversion is unimportant, the phase data is relatively well preserved.

Exercise 2.

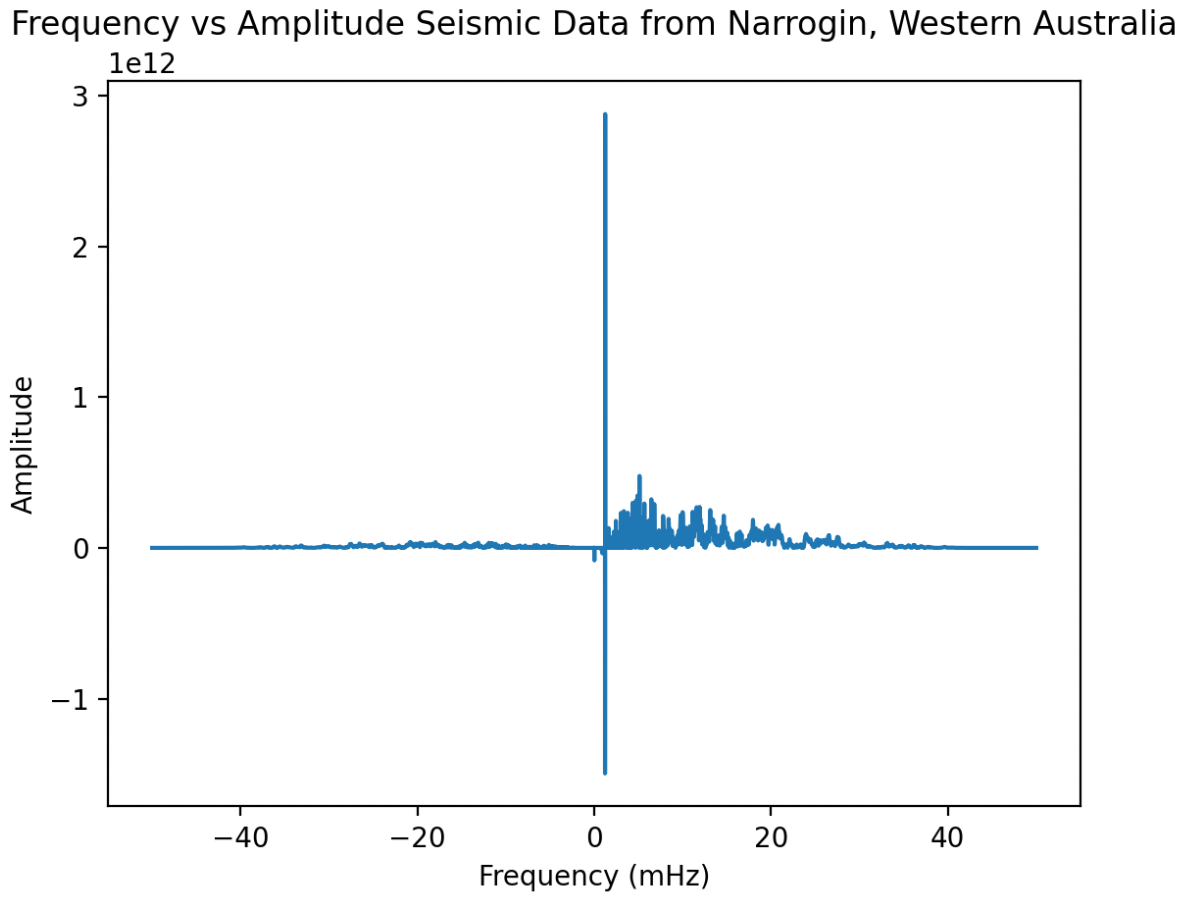
1.

```
import numpy as np
import matplotlib.pyplot as plt

tmp = np.genfromtxt("nwao.vh1")
nwao = tmp.flatten()

time = nwao[0:len(nwao):2]
data = nwao[1:len(nwao):2]

plt.plot(time/3600, data)
plt.xlabel("Time (Hours)")
plt.ylabel("Velocity")
plt.title("Time vs Velocity Seismic Data from Narrogin, Western Australia")
plt.show()
```



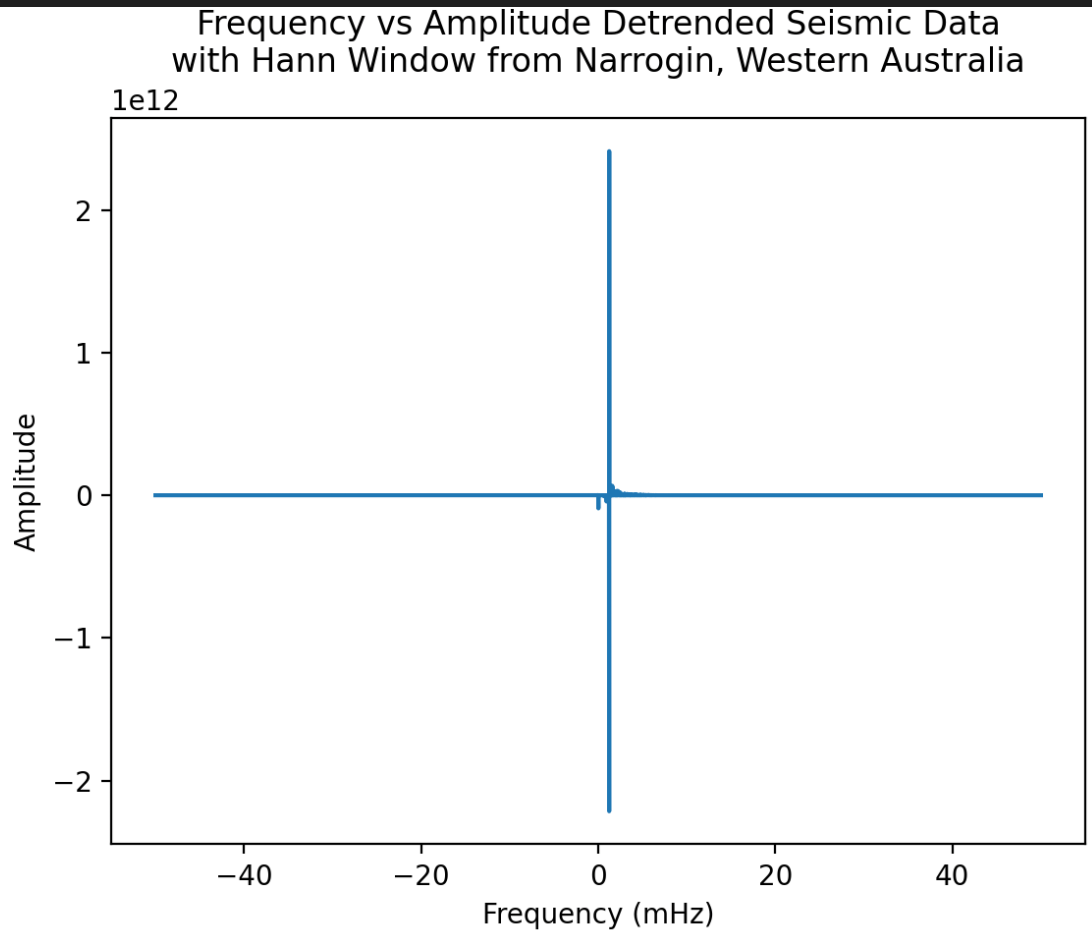
2.

```
p = (np.abs(np.fft.fft(data))**2)/time
w = 1000*np.fft.fftfreq(len(data),10)
arg = np.argsort(w)
p = p[arg]
w = w[arg]

w1 = w
p1 = p

plt.plot(w, p)
plt.xlabel("Frequency (mHz)")
plt.ylabel("Amplitude")
plt.title("Frequency vs Amplitude Seismic Data from Narrogin, Western Australia")
plt.show()
```



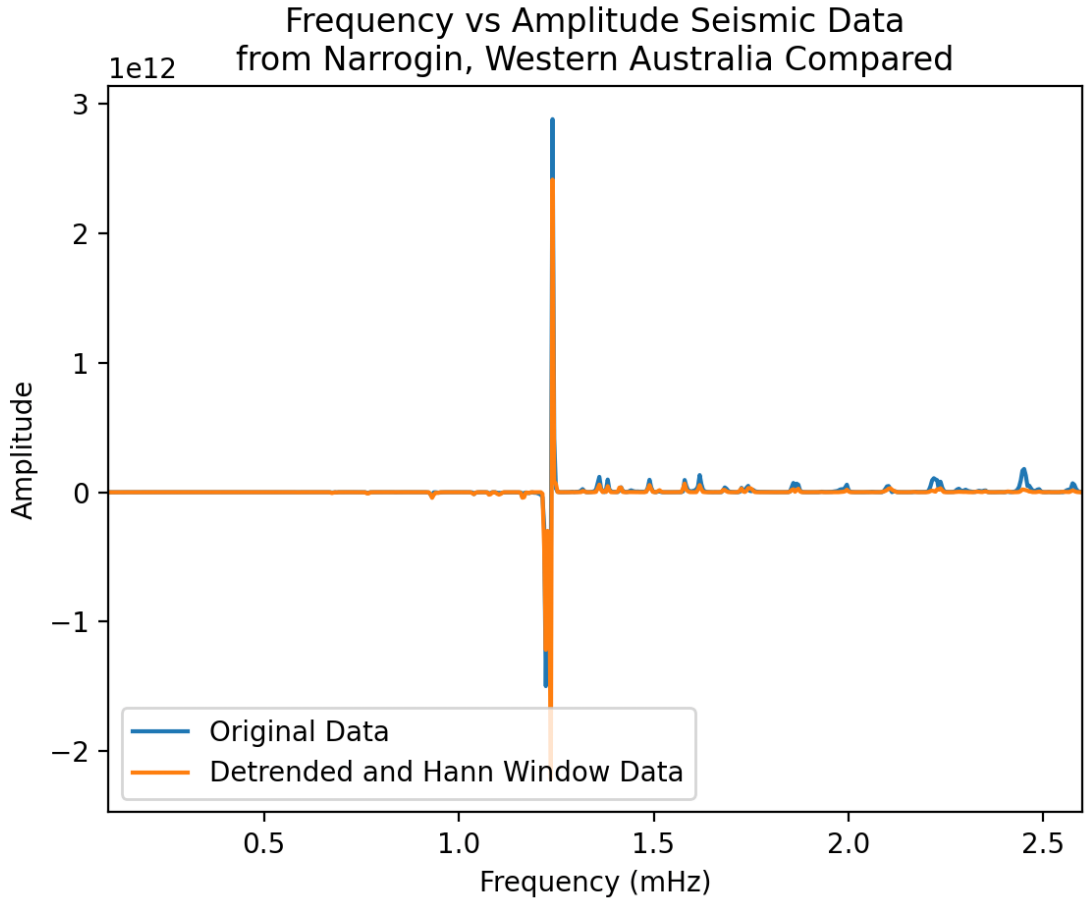
3.

```
trendfit = np.polyfit(time,data,1)
trend = []
for i in range(len(data)):
    trend.append(w[i]*trendfit[0]+trendfit[1])

data = data-trend
n = np.arange(0,len(data))

hann = 1-np.cos(2*np.pi*n/len(data))
data = data*hann

p = (np.abs(np.fft.fft(data))**2)/time
w = 1000*np.fft.fftfreq(len(data),10)
arg = np.argsort(w)
p = p[arg]
w = w[arg]

plt.plot(w, p)
plt.xlabel("Frequency (mHz)")
plt.ylabel("Amplitude")
plt.title("Frequency vs Amplitude Detrended Seismic Data\nwith Hann Window from Narrogin, Western Australia")
plt.show()
```



4.

```
plt.plot(w1,p1,label = "Original Data")
plt.plot(w,p,label = "Detrended and Hann Window Data")
plt.xlim(0.1,2.6)
plt.xlabel("Frequency (mHz)")
plt.ylabel("Amplitude")
plt.title("Frequency vs Amplitude Seismic Data\nfrom Narrogin, Western Australia Compared")
plt.legend()
plt.show()
```



The effect of detrending and applying a Hann window to the data is such that the amplitude of the main mode is mostly preserved but the amplitude of several smaller modes are not. The processed data still retains peaks at some of the additional minor modes, but they are smaller compared to the original data. Though it is also possible that these modes were the result of noise in the original data anyway and so the processing method was successful in eliminating them.

5.

```
plt.plot(w1,p1,label = "Original Data")
plt.plot(w,p,label = "Detrended and Hann Window Data")
plt.annotate("2S3",[1.244,2.877*10**12],[1.3,2.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("1S5",[1.365,1.1*10**11],[1.25,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("2S4",[1.385,1.1*10**11],[1.4,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("0T9",[1.495,1.1*10**11],[1.55,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("0S9",[1.584,1.1*10**11],[1.7,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("1S0",[1.625,1.1*10**11],[1.85,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("0S14",[2.224,1.1*10**11],[2,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.annotate("0S16",[2.456,1.1*10**11],[2.2,0.7*10**12], arrowprops=dict(arrowstyle = "->"))
plt.xlim(0.1,2.6)
plt.xlabel("Frequency (mHz)")
plt.ylabel("Amplitude")
plt.title("Frequency vs Amplitude Seismic Data\nfrom Narrogin, Western Australia Compared with Modes")
plt.legend()
plt.show()
```