```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Example 3: data processing

@author Austin Downey
"""

import IPython as IP
IP.get_ipython().run_line_magic('reset', '-sf')

import numpy as np
import scipy as sp
from scipy import fftpack, signal # have to add
import matplotlib as mpl
import matplotlib.pyplot as plt

plt.close('all')

#%% Load and plot data
D = np.loadtxt('vibration_data/Vibration_measurement.txt',skiprows=23)

tt = D[:,0]
dd = D[:,1]

plt.figure('beam data',figsize=(6.5,3))
plt.plot(tt,dd,'-',label='data 1')
plt.grid(True)
plt.xlabel('time (s)')
plt.ylabel('acceleration (ms$^2$)')
plt.title('beam data')
plt.xlim([-0.1,45])
plt.legend(framealpha=1,loc=0)
plt.tight_layout()
plt.savefig('plot.pdf')
plt.savefig('plot_1.png')
plt.savefig('plot_2.png',dpi=300)



#%% Plot an FFT of the data

# Number of sample points
N = np.shape(dd)[0] # or dd.shape[0]
# sample spacing
T = (tt[-1]-tt[0])/tt.shape[0]
yf = sp.fftpack.fft(dd)
yyf = 2.0/N * np.abs(yf[0:N//2])
xf = np.linspace(0.0, 1.0/(2.0*T), N//2)

plt.figure('FFt plot',figsize=(6.5,3))
plt.plot(xf,yyf)
plt.grid()
plt.xlim([0,150])
plt.xlabel('frequency (Hz)')
plt.ylabel('power')
plt.tight_layout()
plt.savefig('FFT',dpi=300)

#%% Plot a spectrogram of the data

fs=1/T
x=dd

plt.figure('Spectrogram',figsize=(6.5,3))
f, t, Sxx = sp.signal.spectrogram(x, fs,window=('tukey', 5000), nperseg=100000,
                                  noverlap=50000,)
```

```
plt.pcolormesh(t, f, Sxx,vmax=2)
plt.ylabel('Frequency (Hz)')
plt.xlabel('Time (s)')
plt.ylim([0,150])
plt.tight_layout()
plt.savefig('Spectrogram',dpi=300)
```