```
import os, time, pygame, sys
import matplotlib.pyplot as plt
import numpy as np
from scipy.io import wavfile
from scipy.signal import spectrogram
from matplotlib import cm

WD = os.path.dirname(sys.argv[0])

# 1: Load a song and play it back.
fn = os.path.join(WD,'paperback_writer.wav') # <-- path to a wav file here
pygame.mixer.init()
song = pygame.mixer.Sound(fn) # there are a number of different packages
to play audio; pygame is one example.

song.play()
time.sleep(8)
song.stop()

# 2. Load the song as a wavfile.
samplerate, data = wavfile.read(fn)

tStart = 0.5 # in seconds
tEnd = 7 # in seconds

nStart = int(tStart*samplerate)
nEnd = tEnd*samplerate
t = np.linspace(tStart,tEnd,nEnd - nStart)
sample = np.mean(data[nStart:nEnd],axis=1) # get the sample

# Plot the waveform
fig,ax = plt.subplots()
ax.plot(t - t[0],sample/np.max(sample),linewidth=0.3);
ax.set_xlabel('Time (s)')
ax.set_ylabel('Amplitude')
ax.set_xlim((0,ts[-1]))
ax.set_title('Waveform')
ax.set_yticks([])
plt.tight_layout()
plt.show()

# 3. Calculate and plot the spectrogram.
nBits = 10
fs,ts,spect = spectrogram(sample,samplerate,nperseg=2**nBits)

fig,ax = plt.subplots()
ax.contourf(ts,fs/1e3,np.log10(abs(spect)+0.1),20,cmap=cm.YlGnBu)
ax.set_ylim((0.0,12)) #kHz
ax.set_xlim((0.0,ts[-1])) #kHz
ax.set_xlabel('Time (s)');     ax.set_ylabel('Frequency (kHz)')
ax.set_title('Spectrogram')
plt.tight_layout()
plt.show()
```