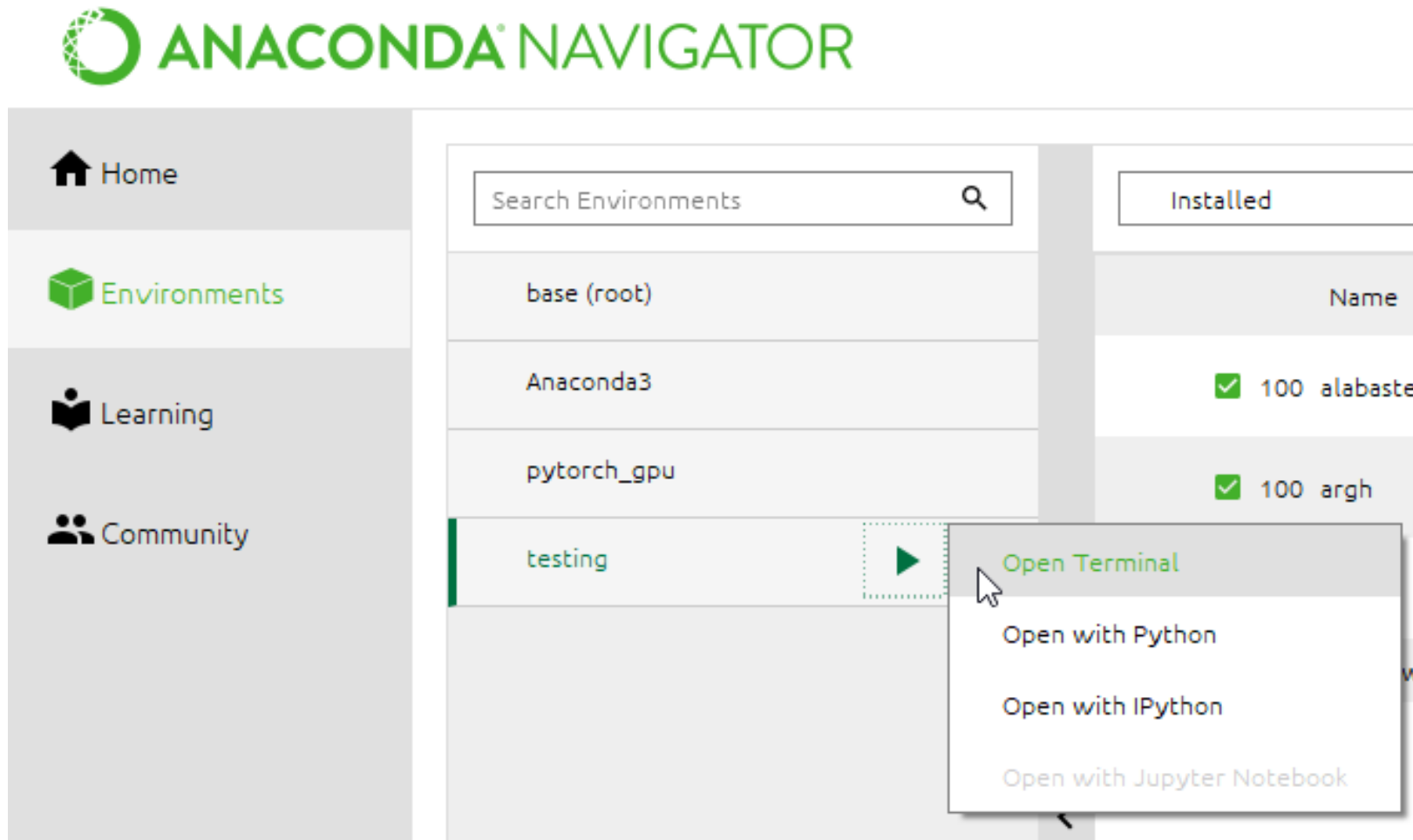# Audio Signal Processing

With Python

Require these module

```
import sounddevice as sd
import soundfile as sf
from scipy.io.wavfile import write
import matplotlib.pyplot as plt
import numpy as np
```

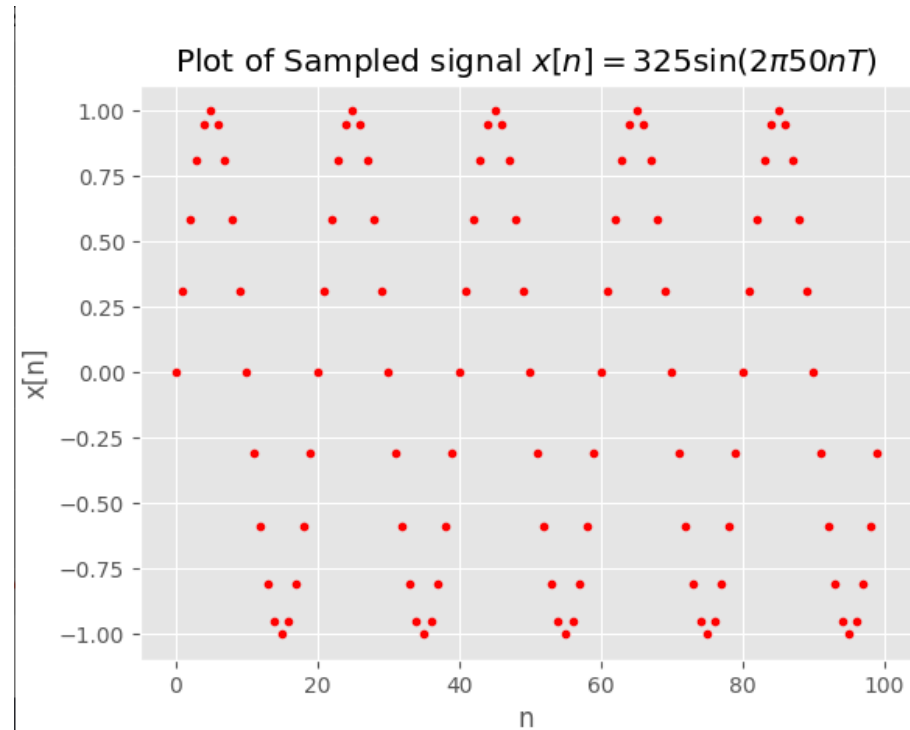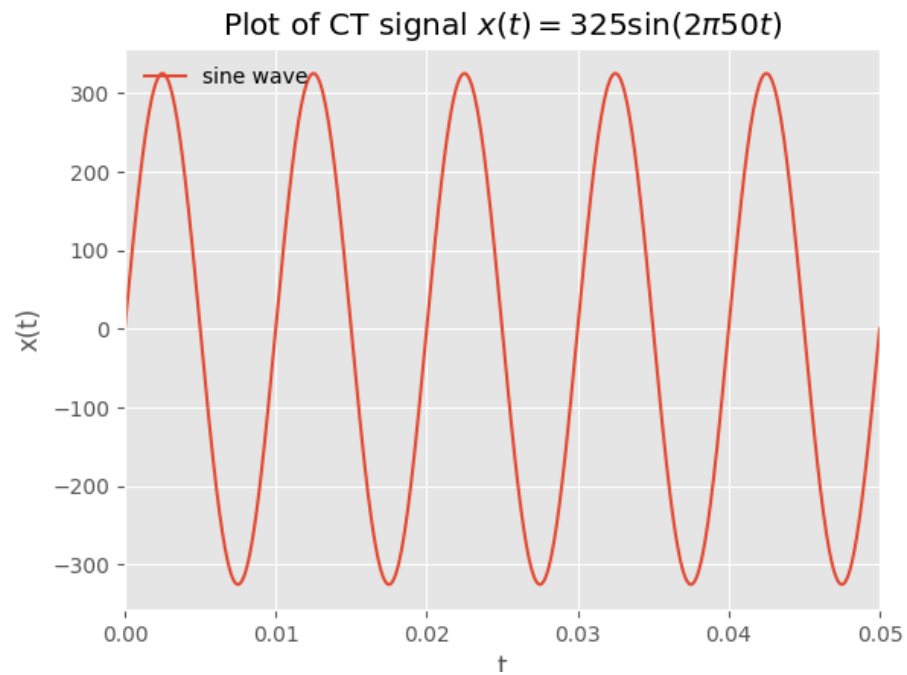# You can use the base environment too

Do pip install

```
(testing) C:\Users\user>pip install soundfile

 (testing) C:\Users\user>pip install sounddevice

(testing) C:\Users\user>pip install scipy
```

Plotting signal in time domain

- script_01_plot_signal.py

Plot of CT signal $x(t) = 325\sin(2\pi 50t)$

Plot of Sampled signal $x[n] = 325\sin(2\pi 50nT)$

Sound analysis

- Purpose is to experiment with simple technique to detect the presence of voice in a quiet enviornment
- **script_02_run_sound_analysis_in_time_domain.py**

Do audio recording

Record your own voice saying 4-5 words

```
#%% Do audio recording
# Parameter
filename = 'output.wav'
fs = 44100  # Sample rate
seconds = 3  # Duration of recording


myrecording = sd.rec(int(seconds * fs), samplerate=fs,
channels=2)
sd.wait()  # Wait until recording is finished
write(filename, fs, myrecording)  # Save as WAV file
```

Sound playback

```
#%% Playback the recorded sound

filename = 'output.wav'
# Extract data and sampling rate from file
data, fs = sf.read(filename, dtype='float32')
sd.play(data, fs)
status = sd.wait()  # Wait until file is done playing
```

```python
#%% Display the recorded sound waveform (plot signal in time domain)
plt.figure(1)
sig1 = myrecording[:,1] # pick channel 1 of stereo audio stream
plt.plot(sig1)

print(" Total samples for {} secs recorded sound = {}
".format(seconds,len(sig1)))

# Show time in secs for x axis
T = 1/fs # sampling period
tvec = T * np.arange(len(sig1))
plt.figure(2)
plt.plot(tvec,sig1)
stringt = "Recorded sound of {} secs".format(seconds)
plt.title(stringt)
plt.show()
```
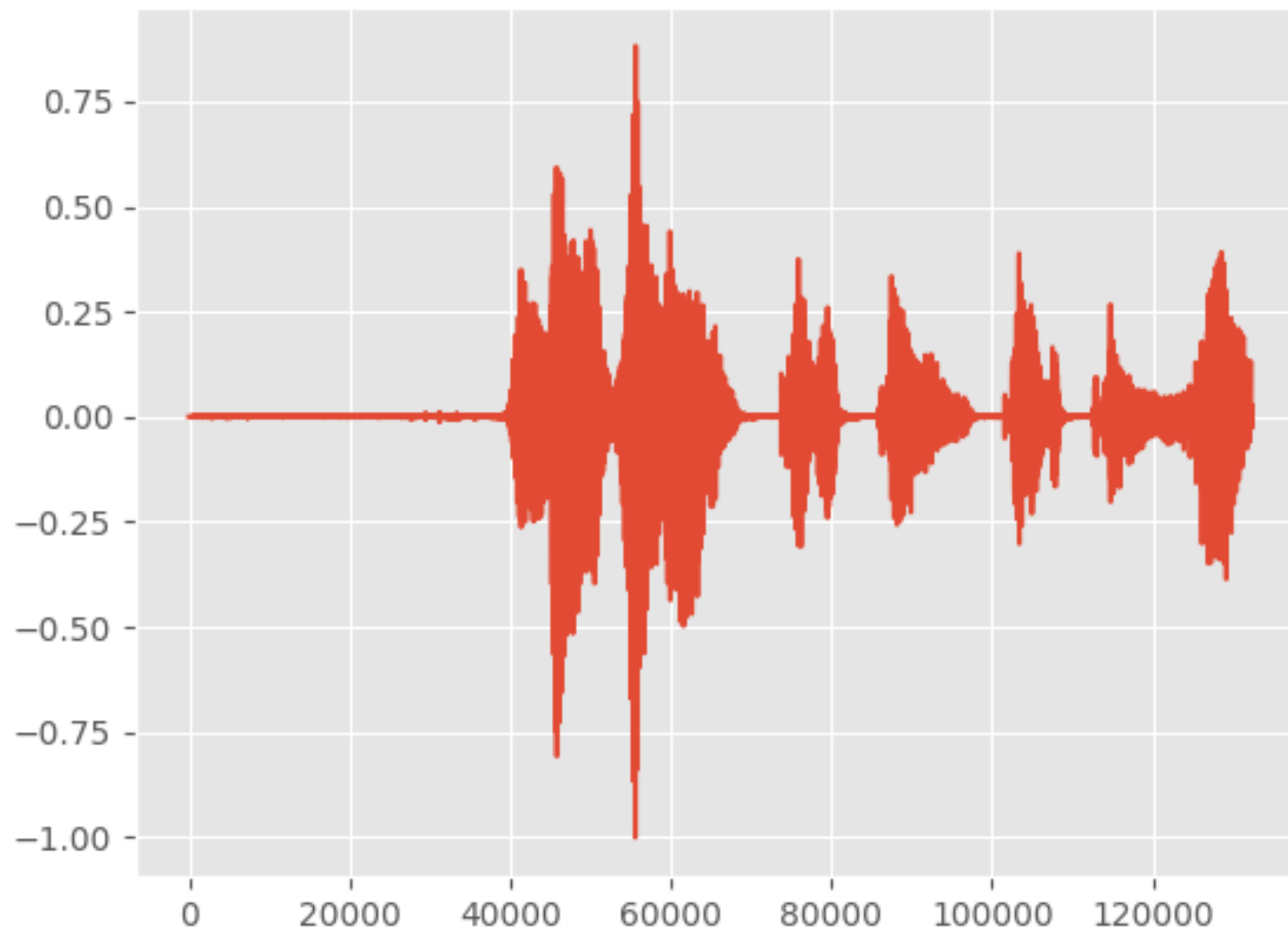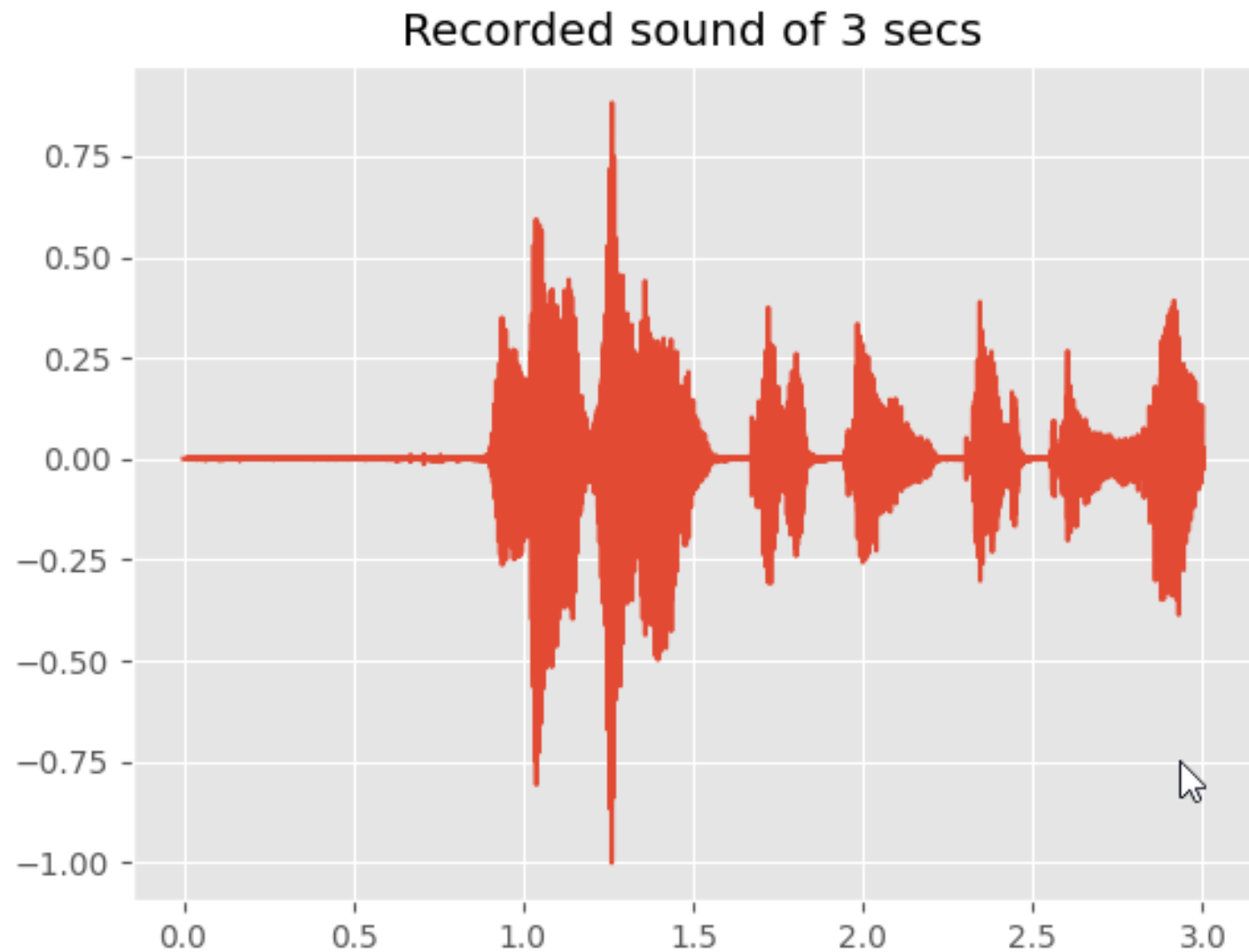
A waveform with with integer indexs (sample number)

Hear my recorded sound hello, hello, testing testing

- X axis in time

Recorded sound of 3 secs

Play selected segment

```python
#%% Plot only selected sound segment
t_start = 1.0
t_end = 3 # sec

n1 = int( np.floor(t_start/T) )
n2 = int( np.floor(t_end/T) )

sig1_sel = sig1[n1:n2+1] # include n2
numPt = len(sig1_sel)
tvec = T * np.arange(n1,n1+numPt)

plt.figure(3)
plt.plot(tvec,sig1_sel)
stringt = "Recorded sound of {} secs to {} secs".format(t_start, t_end)
plt.title(stringt)
plt.show()

# Play only selected segment
sd.play(sig1_sel, fs)
status = sd.wait()  # Wait until file is done playing
```
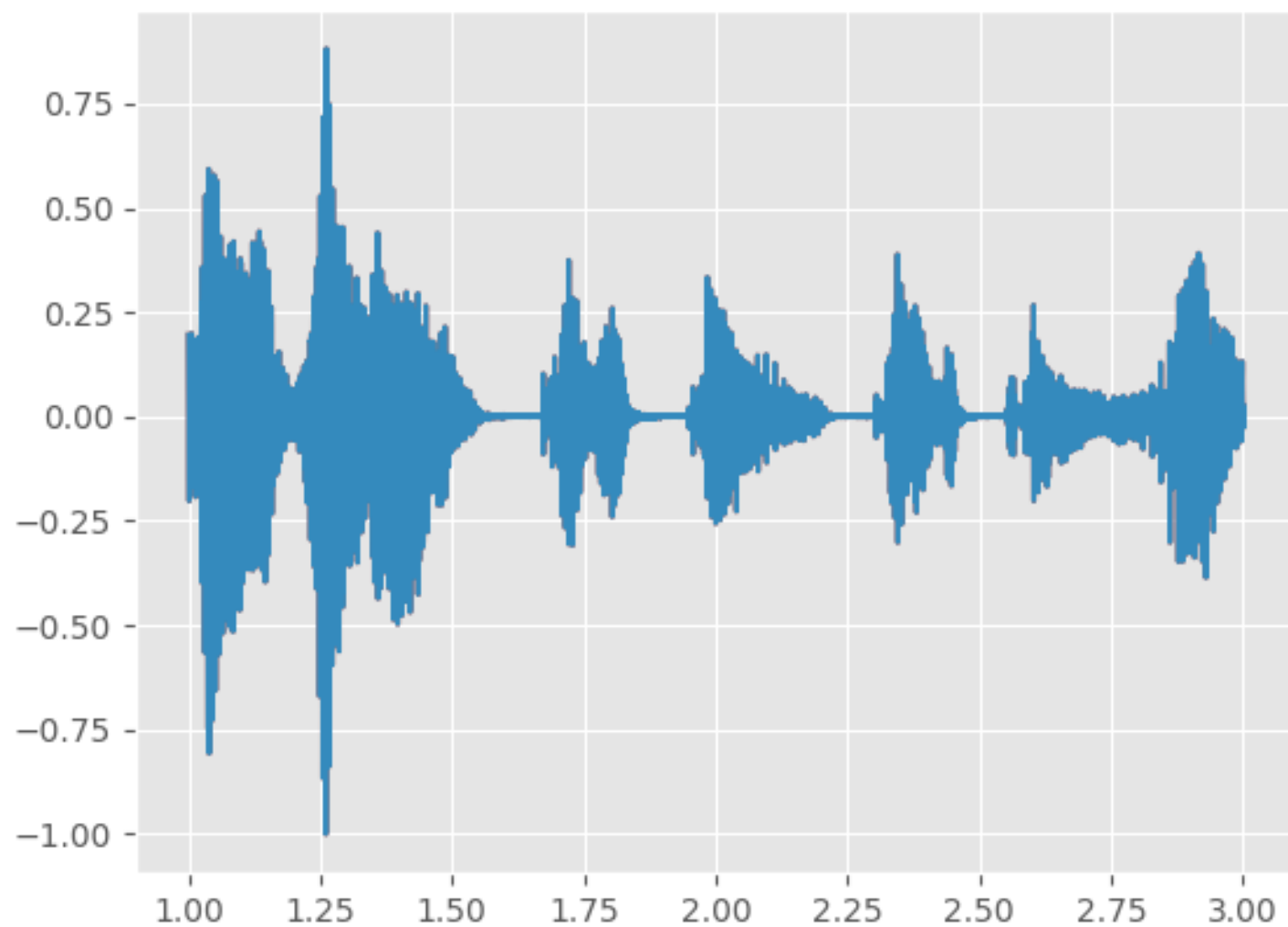
Recorded sound of 1.0 secs to 3 secs

Challenge exercise for students

- Design a voice activity detector

- Write a program to monitor the sound of a quite room during an online test

- Print out an alert when a sound of somebody speaking is detected