Simple Spectrogram

- sample bird call audio

```
[channels, sample_rate] = audioread('../datasets/XC403881.wav');
whos;
```

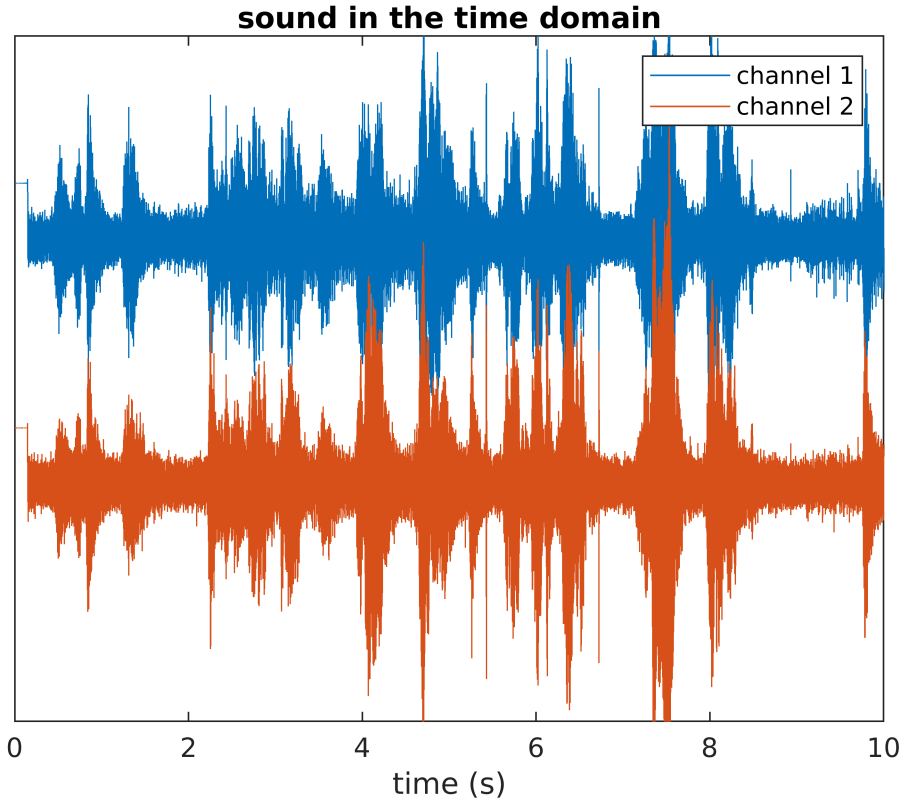
```
  Name                Size                 Bytes  Class                                    Attributes

  N                   1x1                      8  double
  ax                  1x1                      0  matlab.graphics.axis.Axes
  c                   1x1                      0  matlab.graphics.illustration.ColorBar
  channels         450879x2                7214064  double
  colorz              1x2                      4  char
  f1                  1x1                      8  matlab.ui.Figure
  f2                  1x1                      8  matlab.ui.Figure
  f3                  1x1                      8  matlab.ui.Figure
  f4                  1x1                      0  matlab.ui.Figure
  freq_               1x2                    256  cell
  frequency         513x1                   4104  double
  hz_vals             1x225440             1803520  double
  k                   1x1                      8  double
  powers           450879x1                3607032  double
  sample_rate         1x1                      8  double
  spectrum          513x500                4104000  double                                   complex
  t                   1x450879             3607032  double
  time                1x500                   4000  double
```

- note there are 2 channels
- play the audio

```
soundsc(channels, sample_rate);
N = length(channels);
t = (0:N-1)/sample_rate;
```

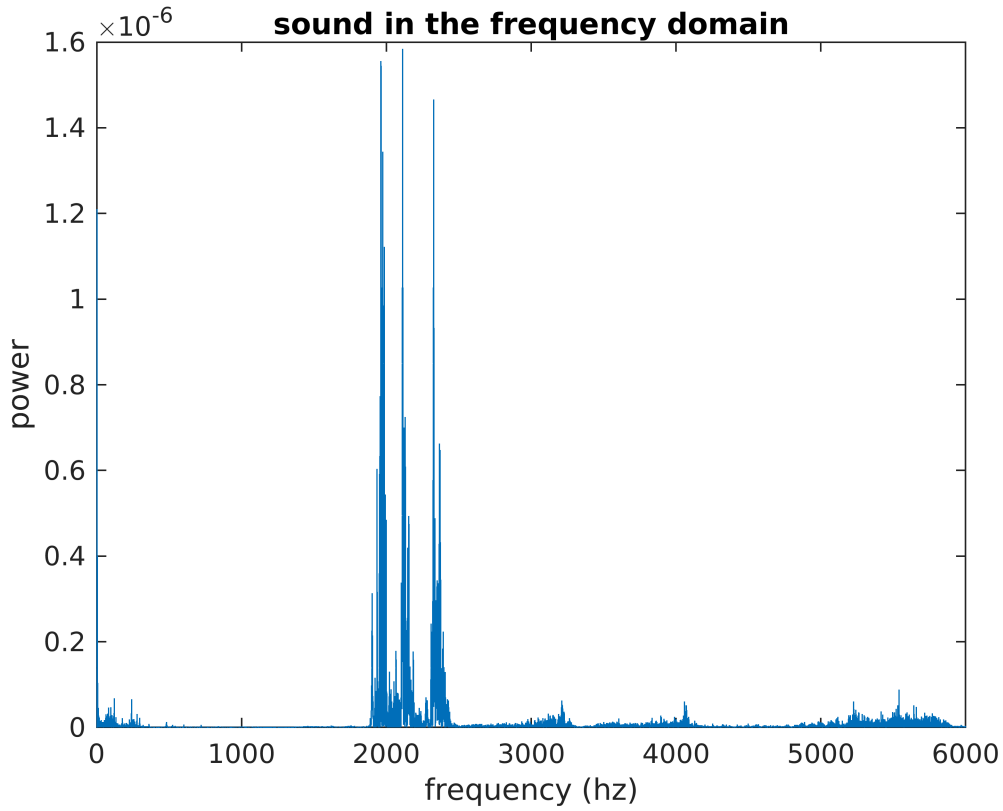
- plot the channels

```
f1 = figure(1); clf;
f1.Position = [0 0 600 300];
plot(t, bsxfun(@plus, channels, [.25 0])); %offset the channels
xlabel("time (s)");
title("sound in the time domain");
xlim([0 10]);
ylim([-.3 .4]);
yticks([]);
legend(["channel 1", "channel 2"]);
```



- compute and plot the power spectrum

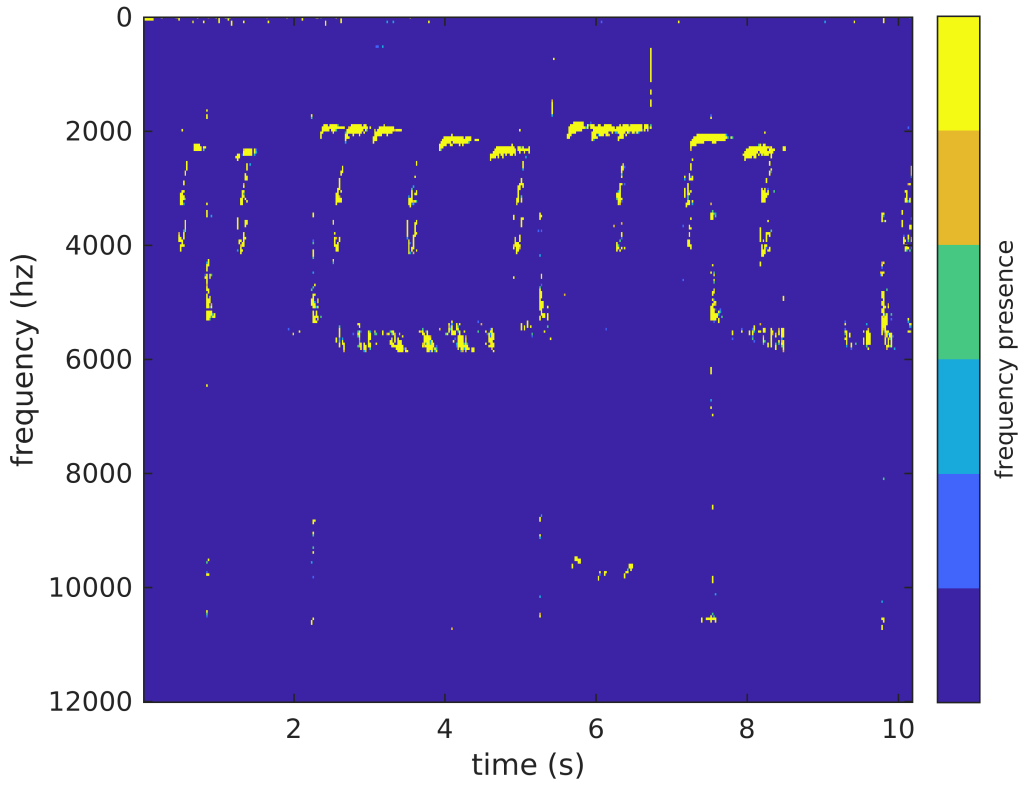
```
hz_vals = linspace(0, sample_rate/2, floor(N/2)+1);
powers = abs(fft(detrend(channels(:,1)))/N).^2;

f2 = figure(2); clf;
f2.Position = [0 0 500 300];
plot(hz_vals, powers(1:length(hz_vals)));
xlabel("frequency (hz)");
ylabel("power");
title("sound in the frequency domain");
xlim([0 6000]);
```



Now Apply spectrogram

- use function in signal processing tool box

```
[spectrum, frequency, time] = spectrogram(detrend(channels(:,1)), hann(1000), 100, [],

f3 = figure(3); clf;
f3.Position = [0 0 600 300];
imagesc(time, frequency, abs(spectrum).^2);
ax = gca;
ax.CLim = [3 4];
ax.YLim = frequency([1 dsearchn(frequency, 12000)]);
ax.XLim = time([1 end]);
colormap(parula(6));
c = colorbar();
c.Label.String = "frequency presence";
set(c, 'YTick', []);
xlabel("time (s)");
ylabel("frequency (hz)");
```



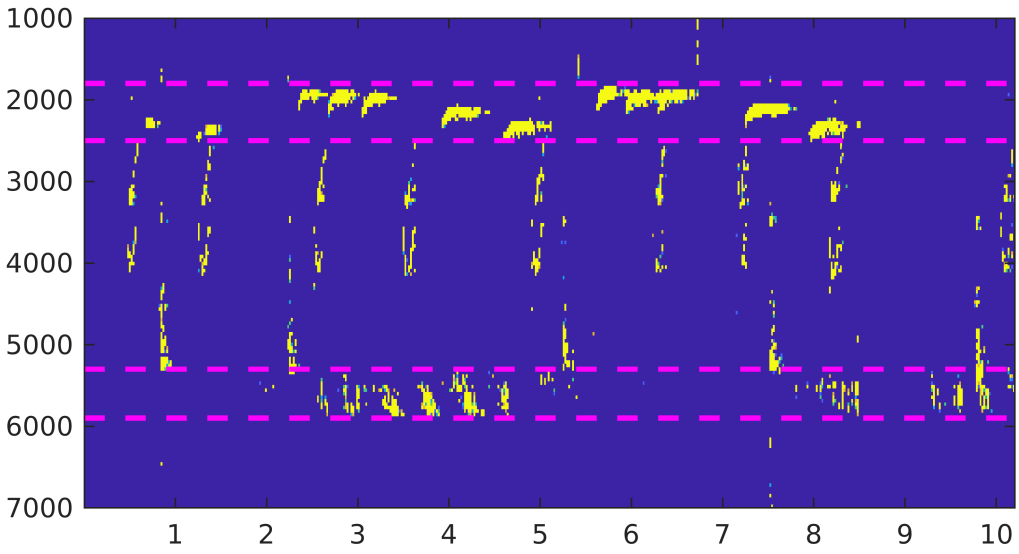
Separate the sources

- visually inspect the spectrogram to identify frequencies of interest

```
freq_{1} = [1800 2500];
freq_{2} = [5300 5900];
```

Plot

```
colorz = 'kk';
f4 = figure(4); clf;
f4.Position = [0 0 600 300];
imagesc(time, frequency, abs(spectrum).^2);
ylim([1000 7000]);
ax = gca;
ax.CLim = [3 4];
hold on;
for k = 1:length(freq_)
    plot(get(gca,'xlim'), [1 1]*freq_{k}(1), 'm--', 'linew', 2);
    plot(get(gca,'xlim'), [1 1]*freq_{k}(2), 'm--', 'linew', 2);
end
hold off;
```


1000
2000
3000
4000
5000
6000
7000
1
2
3
4
5
6
7
8
9
10