test

May 21, 2022

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import spatial
plt.rcParams['figure.figsize'] = (10, 5)
```

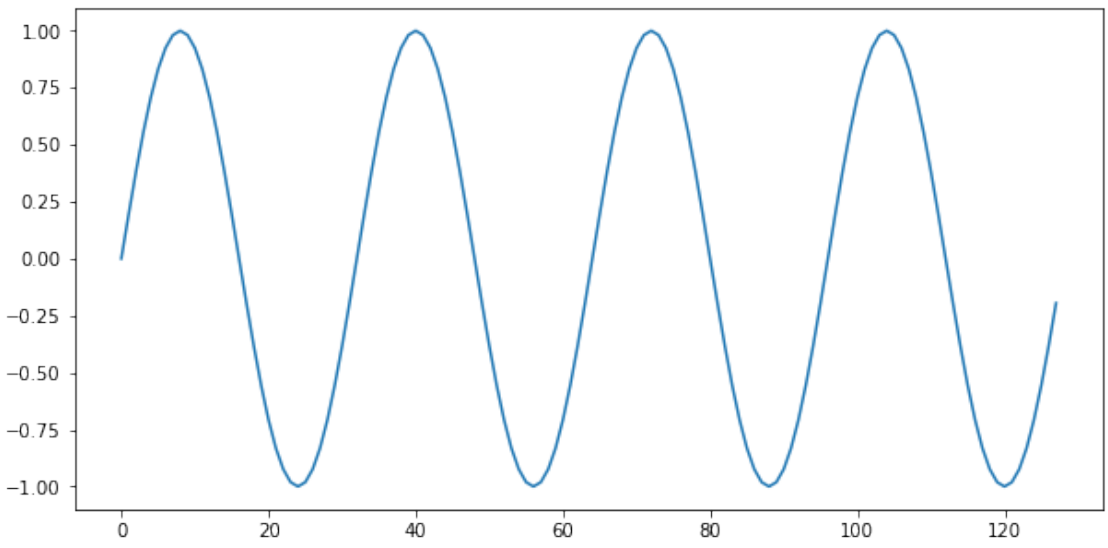
0.1 FFT tests

0.1.1 Sinewave

[ ]:
```
data = pd.read_csv('../results/tests/fft_sinewave.csv', header=None)
size = data.shape[0]
print(size)
freq = range(0, size, 1)
plt.plot(freq, data[0][:size])
```

```
128
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7f0c3755c790>]
```

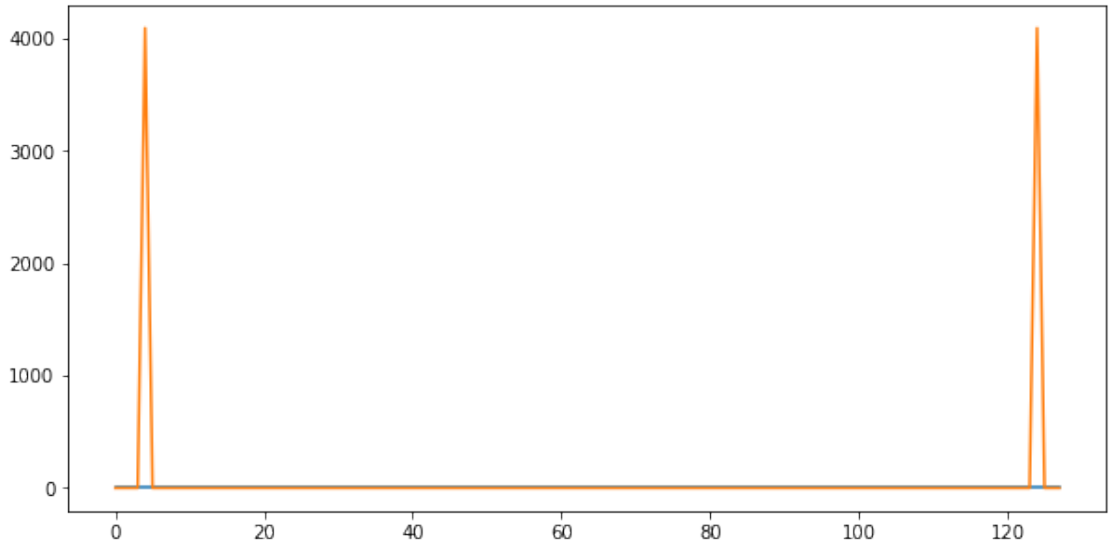


[ ]:
```
data = pd.read_csv('../results/tests/fft_fftsinewave.csv', header=None).
 ↪to_numpy()
size = data.shape[0]
print(data.shape)

freq = range(0, size, 1)
plt.plot(freq, np.abs(data[:size])**2)
```

```
(128, 2)
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7f0c375bc490>,
 <matplotlib.lines.Line2D at 0x7f0c375bcca0>]
```

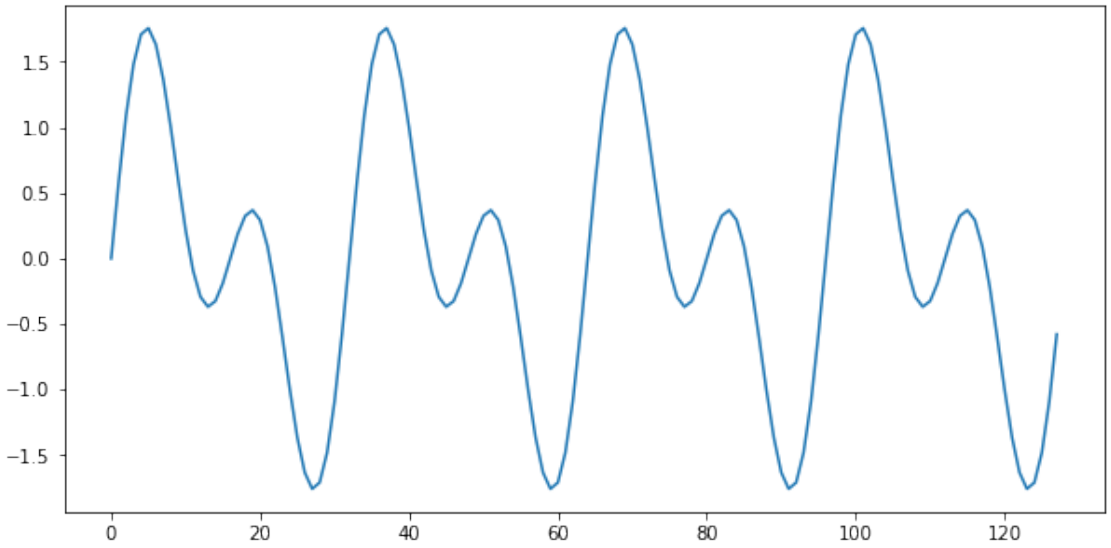


0.1.2 Wave sum

[ ]:
```
data = pd.read_csv('../results/tests/fft_wavesum.csv', header=None)
size = data.shape[0]
print(size)
freq = range(0, size, 1)
plt.plot(freq, data[0][:size])
```

```
128
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7f0c37809070>]
```

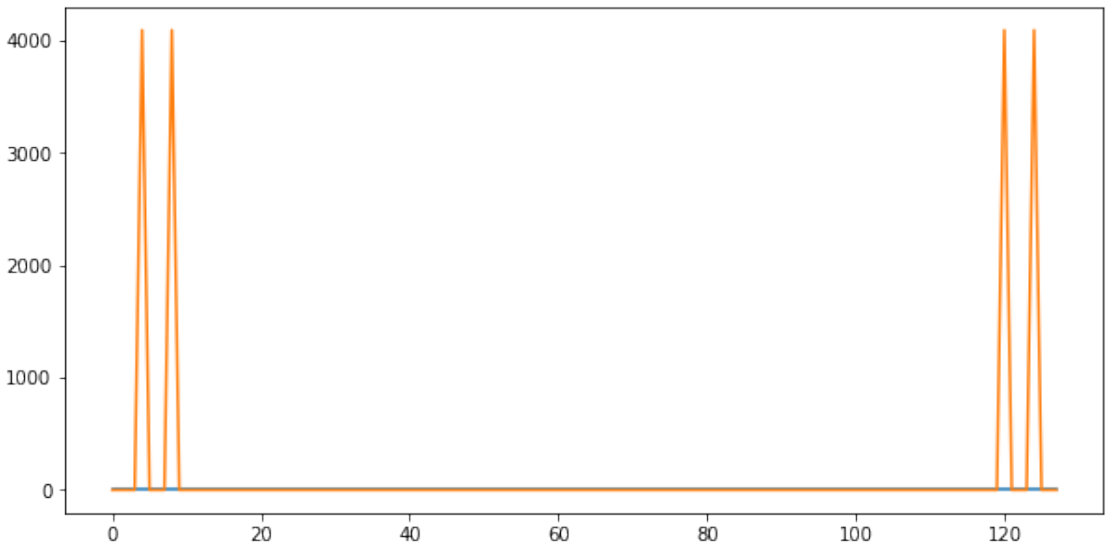


[ ]:
```
data = pd.read_csv('../results/tests/fft_fftwavesum.csv', header=None).
↪to_numpy()
size = data.shape[0]
print(size)

freq = range(0, size, 1)
plt.plot(freq, np.abs(data[:size])**2)
```

```
128
```

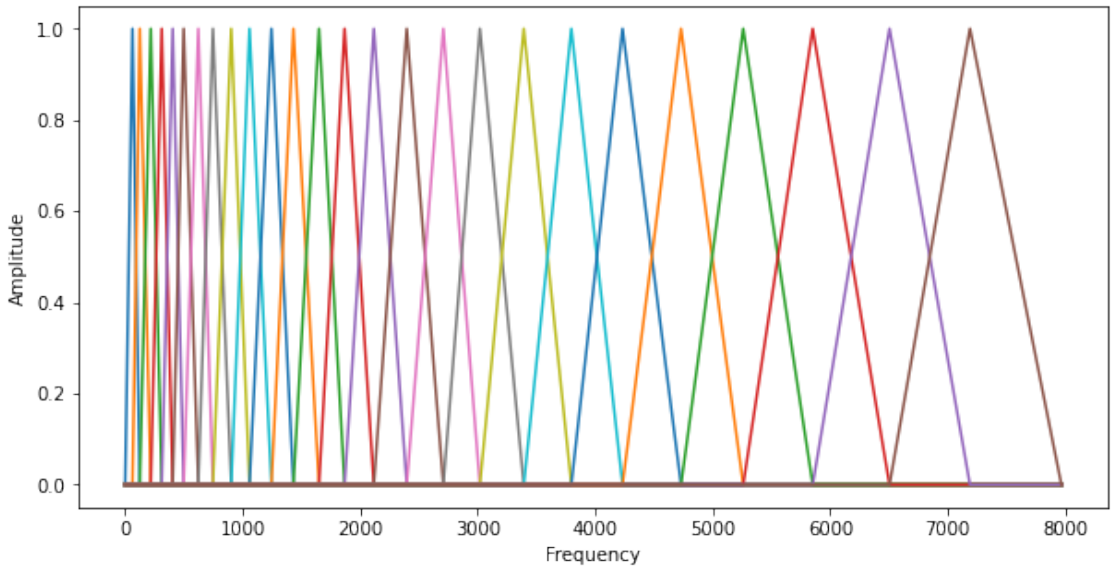
[ ]:
```
[<matplotlib.lines.Line2D at 0x7f0c37581d90>,
 <matplotlib.lines.Line2D at 0x7f0c375811f0>]
```



0.2 Mel coefficients

0.2.1 Filterbank

[ ]:
```
fbank = pd.read_csv('../results/tests/filterbank.csv', header=None).to_numpy()
plt.plot(fbank[0:257,:]);
plt.xticks(np.arange(0,258,32.125), np.arange(0,8001,1000));
plt.xlabel("Frequency");
plt.ylabel("Amplitude");
```



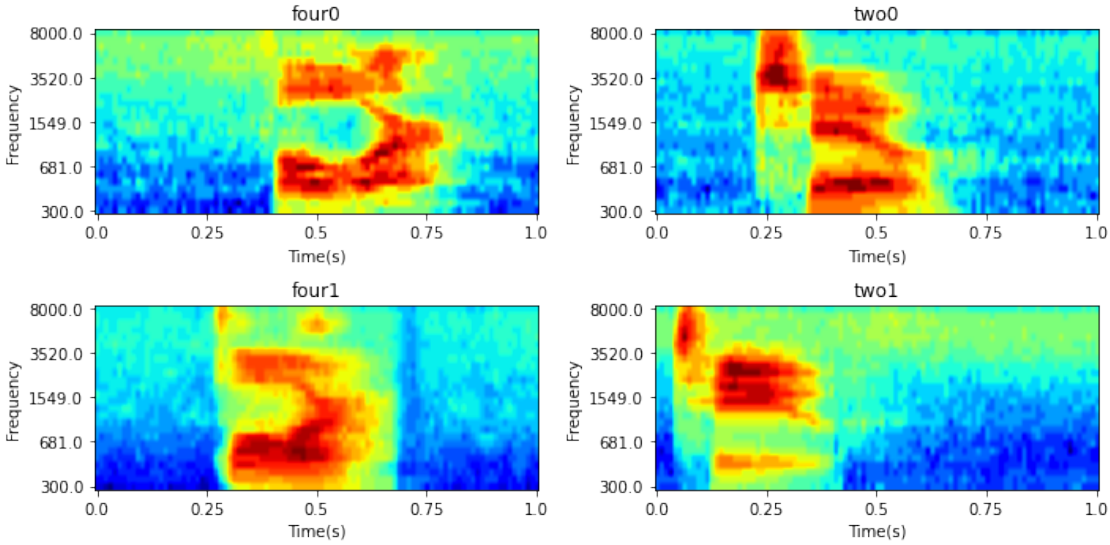
Lets compare the spectograms generated with mel scale for different audio recordings of the same word

[ ]:
```
sample_names = ['four0', 'two0', 'four1', 'two1', 'marvin', 'happy']
```

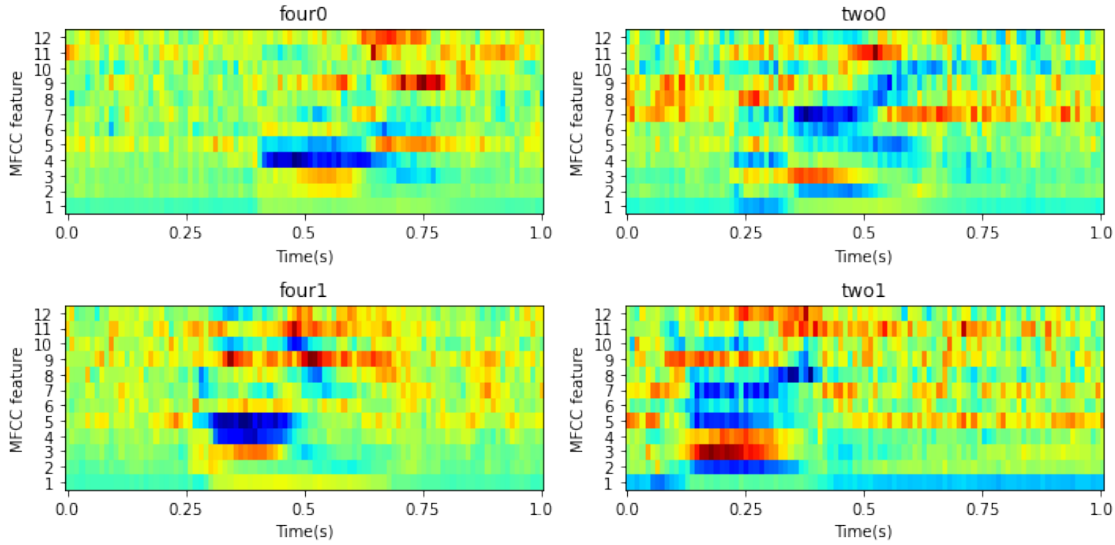
0.2.2 Mel spectograms

[ ]:
```
fig = plt.figure()
for i in range(0,4):
    fig.add_subplot(2,2,i+1)
    spectogram = pd.read_csv('../results/tests/'+ sample_names[i] +␣
↪'_spectogram.csv', header=None).to_numpy().transpose()
    plt.imshow(spectogram, origin='lower', cmap='jet', aspect='auto')
    plt.xticks(np.linspace(0, spectogram.shape[1]-1, num=5), np.linspace(0, 1,␣
↪num=5))
```

```
    plt.yticks(np.linspace(0, spectogram.shape[0]-1, num=5),np.trunc(np.
 ↪logspace(np.log2(300),np.log2(8000), base=2, num=5)))
    plt.title(sample_names[i])
    plt.xlabel("Time(s)");
    plt.ylabel("Frequency");
fig.tight_layout()
```



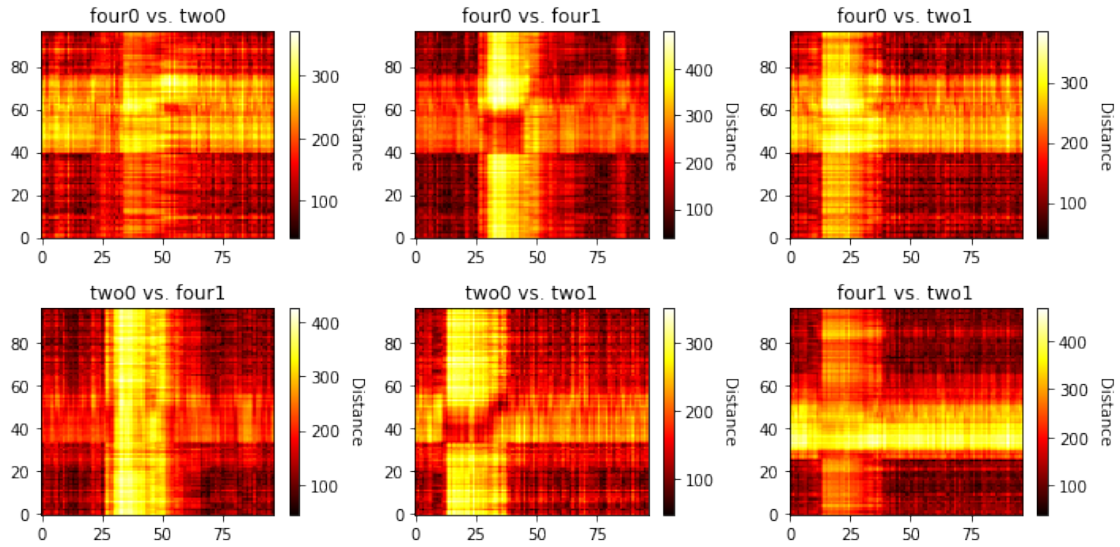
[ ]:
```
fig = plt.figure()
for i in range(0,4):
    fig.add_subplot(2,2,i+1)
    mfcc = pd.read_csv('../results/tests/'+ sample_names[i] + '_mfcc.csv',␣
 ↪header=None).to_numpy().transpose()
    plt.imshow(mfcc, origin='lower', aspect='auto', cmap='jet')
    plt.xticks(np.linspace(0, mfcc.shape[1]-1, num=5), np.linspace(0, 1, num=5))
    plt.yticks(np.arange(0,12,1), np.arange(1,13,1))
    plt.title(sample_names[i])
    plt.xlabel("Time(s)");
    plt.ylabel("MFCC feature");
fig.tight_layout()
```



0.3 Comparison

0.3.1 Distance matrix

[ ]:
```
from scipy import spatial
fig = plt.figure()
subplot_num = 1
for i in range(0,4):
    for j in range(i+1,4):
        fig.add_subplot(2,3,subplot_num)
        subplot_num = subplot_num + 1
        mfcc1 = pd.read_csv('../results/tests/'+ sample_names[i] + '_mfcc.csv',␣
↪header=None).to_numpy()
        mfcc2 = pd.read_csv('../results/tests/'+ sample_names[j] + '_mfcc.csv',␣
↪header=None).to_numpy()
        dmatrix = spatial.distance_matrix(mfcc1, mfcc2)
        im = plt.imshow(dmatrix, aspect='auto', interpolation='none',␣
↪cmap='hot', origin='lower')
        ax = fig.gca()
        plt.title(sample_names[i] + ' vs. ' + sample_names[j])
        cbar = ax.figure.colorbar(im, ax=ax)
        cbar.ax.set_ylabel('Distance', rotation=-90, va="bottom");
fig.tight_layout()
```



[ ]:
```
fig = plt.figure()
mfcc1 = pd.read_csv('../results/tests/'+ sample_names[0] + '_mfcc.csv',␣
 ↪header=None).to_numpy()
dmatrix = spatial.distance_matrix(mfcc1, mfcc1)
im = plt.imshow(dmatrix, aspect='auto', interpolation='none', cmap='hot',␣
 ↪origin='lower')
ax = fig.gca()
plt.title(sample_names[0] + ' vs. ' + sample_names[0])
cbar = ax.figure.colorbar(im, ax=ax)
cbar.ax.set_ylabel('Distance', rotation=-90, va="bottom");
fig.tight_layout()
```