ProjectPreAnalysis

June 25, 2019

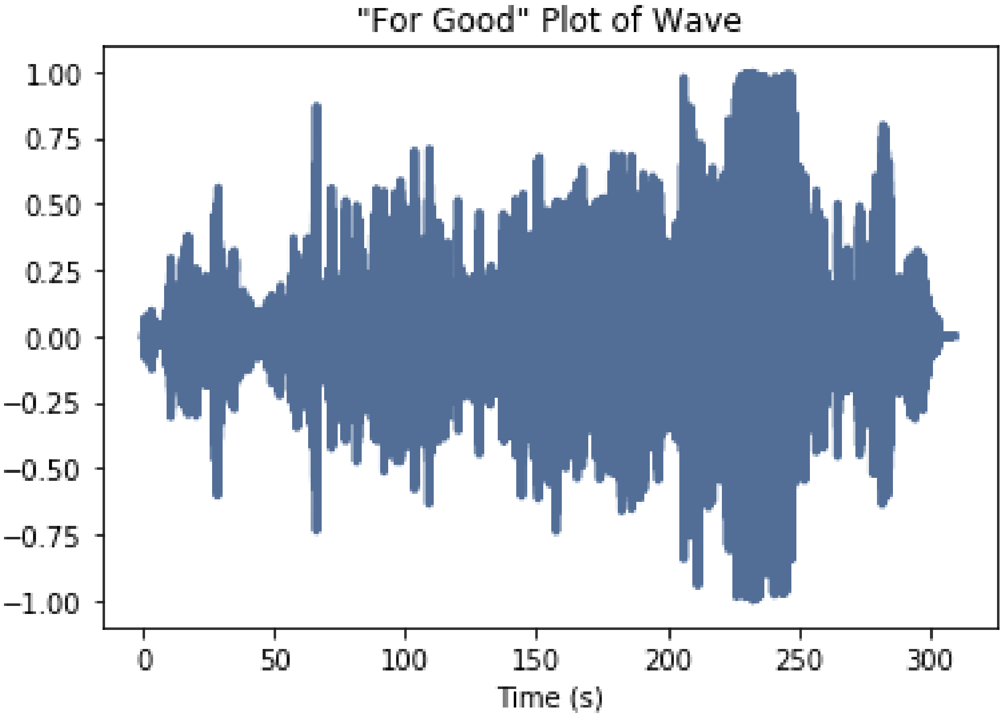
1 Project Pre-Anlysis

This script is to allow pre analysis in order to determine if the songs in question are viable uses for our purpose.

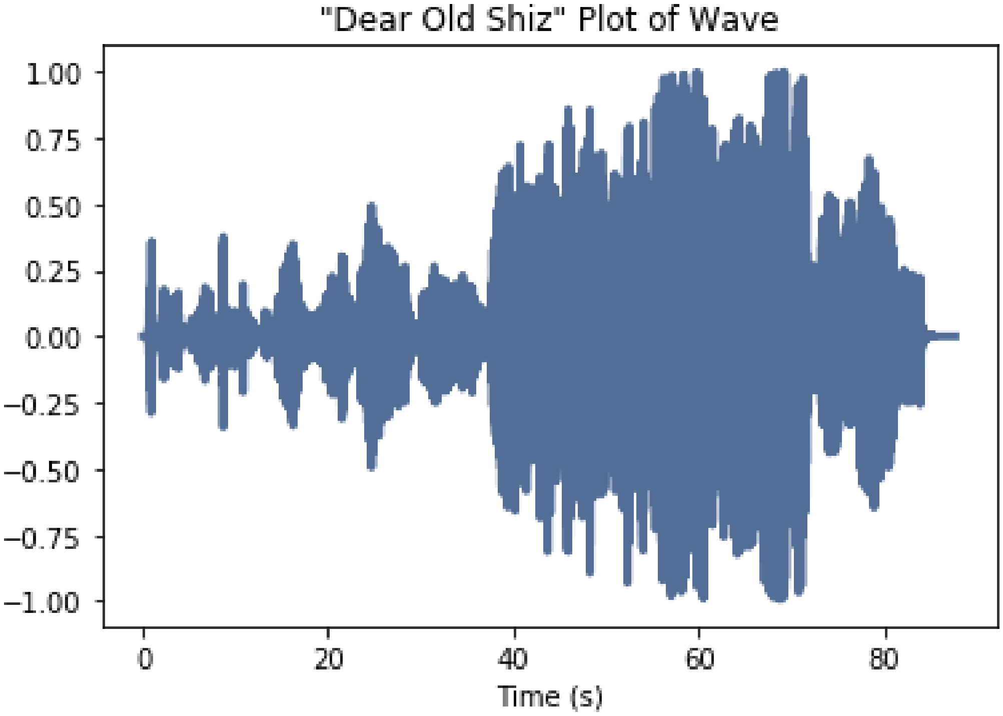
```
In [1]: from __future__ import print_function, division

        #Importing the DSP library
        import sys
        sys.path.insert(0, '../../ThinkDSP/code')

        import thinkdsp
        import thinkplot
        import thinkstats2

        import numpy as np

        import warnings
        warnings.filterwarnings('ignore')

        from IPython.html.widgets import interact, fixed
        from IPython.html import widgets
```

```
In [3]: # Importing the music as a wave

        # First we import the song For Good which will be our 'reference' wave throughout the 
        # We are looking for the main introduction that is a repeated theme throughout the ent
        good = thinkdsp.read_wave('WickedAlbum/ForGood.wav')
        good.normalize()

        # A known, and simple, example of this pattern is the introduction of Dear Old Shiz.
        shiz = thinkdsp.read_wave('WickedAlbum/DearOldShiz.wav')
        shiz.normalize()
```

```
In [4]: #Let us plot the For Good Wave just for kicks to see what it looks like.
        good.plot()
        thinkplot.config(xlabel='Time (s)',title='"For Good" Plot of Wave')
```



```
In [5]: # Do the same thing for the Dear Old Shiz wave
        shiz.plot()
        thinkplot.config(xlabel='Time (s)',title='"Dear Old Shiz" Plot of Wave')
```

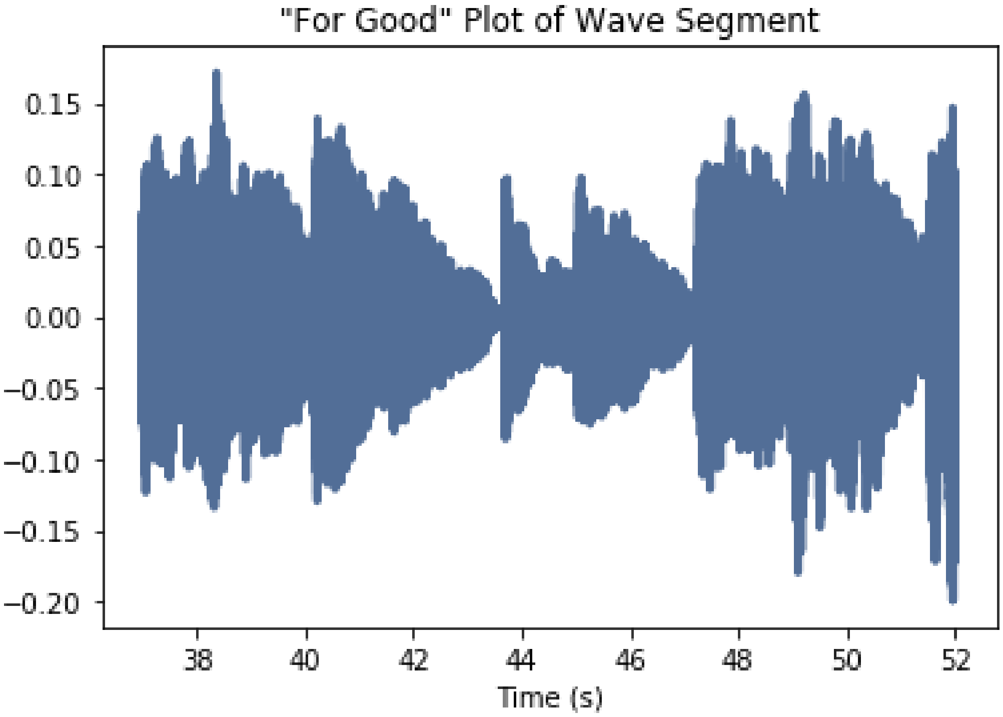


```
In [6]: # We are curious about specific segments of the waves which we know contain the patter
        # throughout the entire music soundtrack.
        # We are manually going through and comparing the waves/ spectrums/etc. in order to de
        # before running this through the Matrix Profile.

        #Taking a short segment
        good_short = good.segment(start=37,duration=15)
        shiz_short = shiz.segment(start=14,duration=23.5)
        #Writing out the segment to a new file to check that we picked the correct part.
        good_shortSeg = good_short.write(filename='good_segment.wav')
        shiz_shortSeg = shiz_short.write(filename='shiz_segment.wav')
```

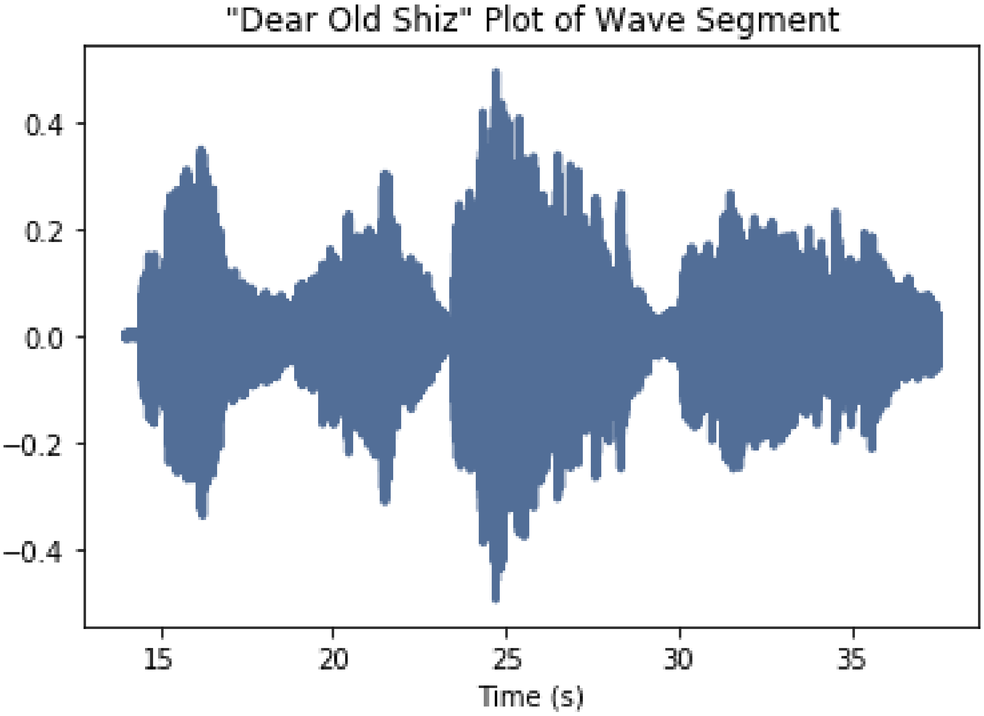
```
Writing good_segment.wav
Writing shiz_segment.wav
```

Let us plot those short segments to understand the wave structure of each.

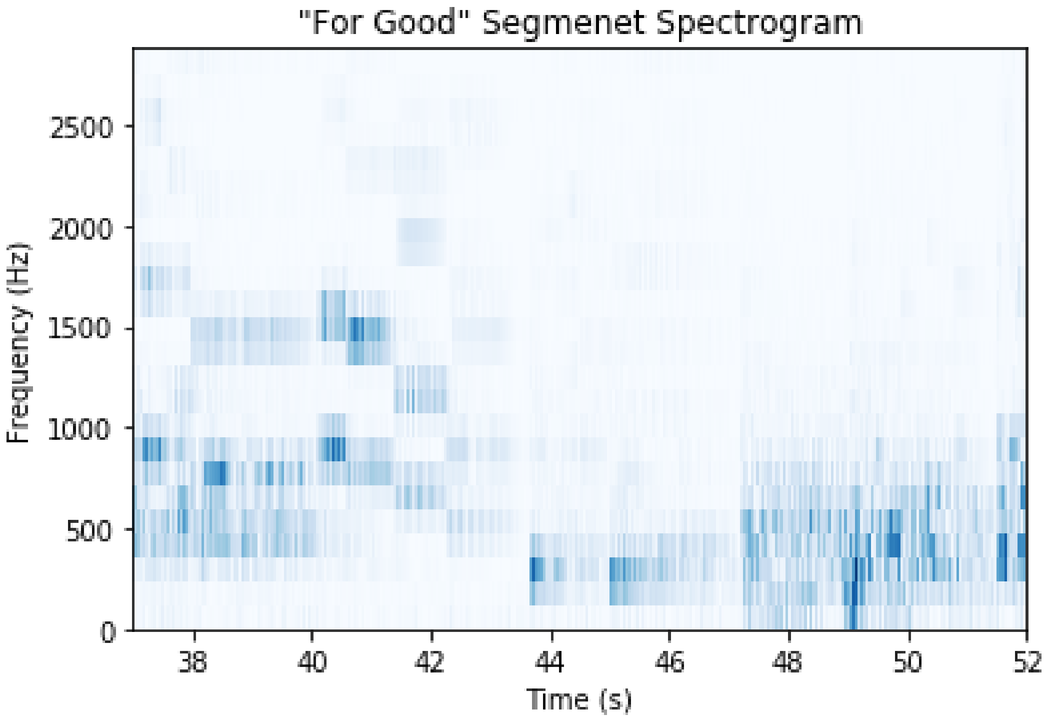
```
In [7]: #First 'For Good' Wave segmenet
        good_short.plot()
        thinkplot.config(xlabel='Time (s)',title='"For Good" Plot of Wave Segment')
```



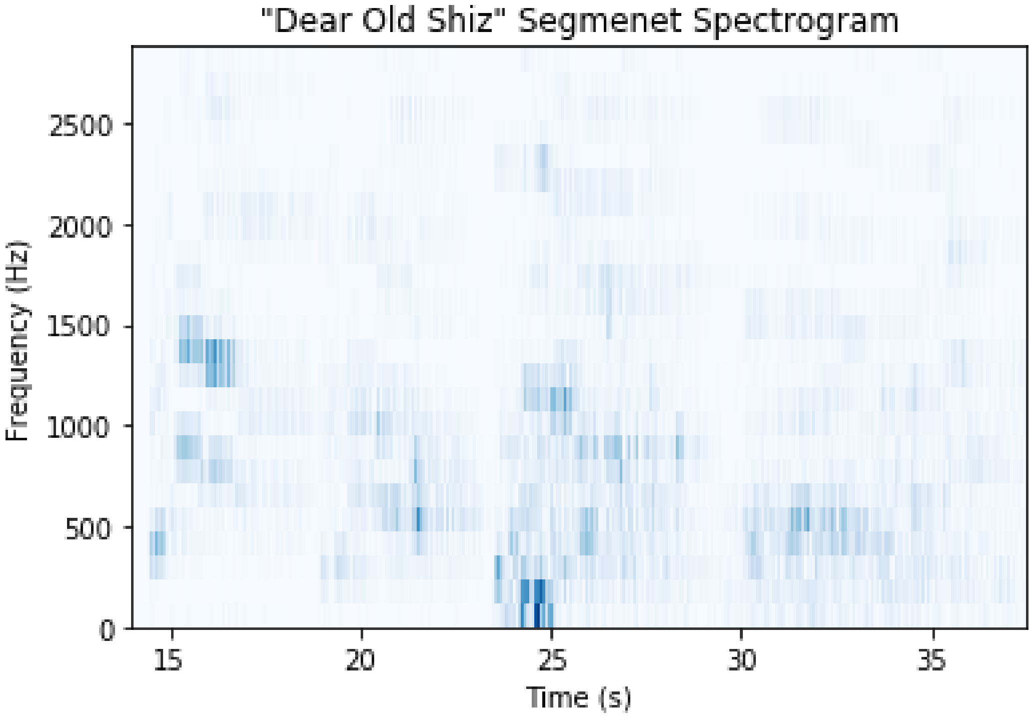
```
In [8]: #Next 'Dear Old Shiz' Wave segmenet
        shiz_short.plot()
        thinkplot.config(xlabel='Time (s)',title='"Dear Old Shiz" Plot of Wave Segment')
```



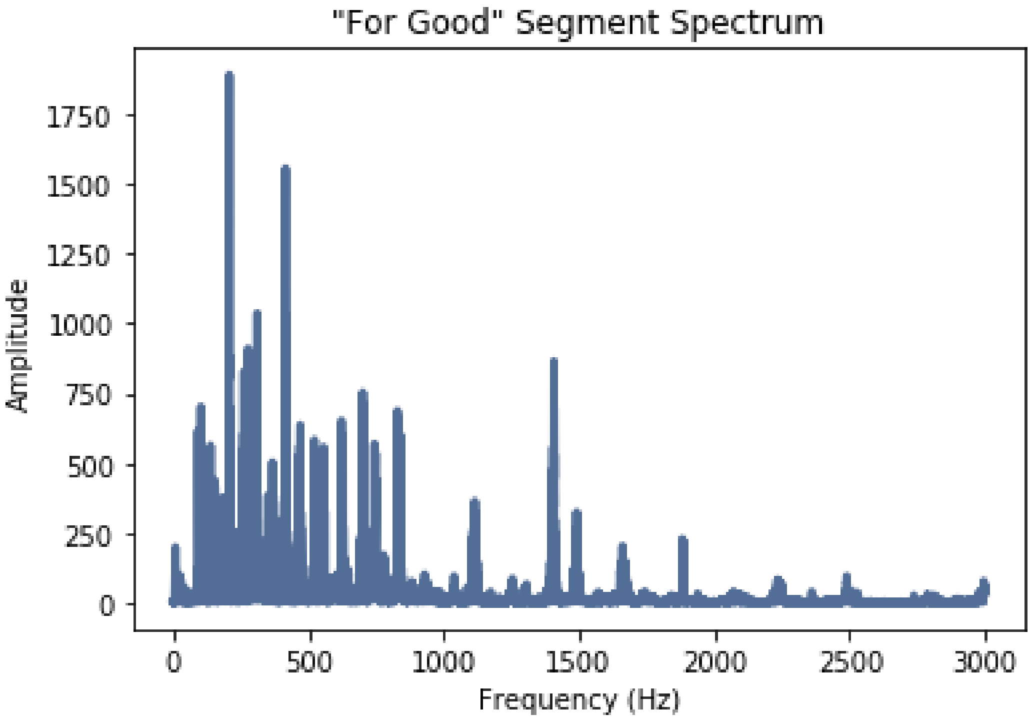
```
In [9]: # Converting into Spectrogram and plotting the "For Good" Segment

        # The spectrogram shows the harmonic structure over time.
        good_gram = good_short.make_spectrogram(seg_length=400)
        good_gram.plot(high=3000)
        thinkplot.config(xlabel='Time (s)', ylabel='Frequency (Hz)', title='"For Good" Segmene
```



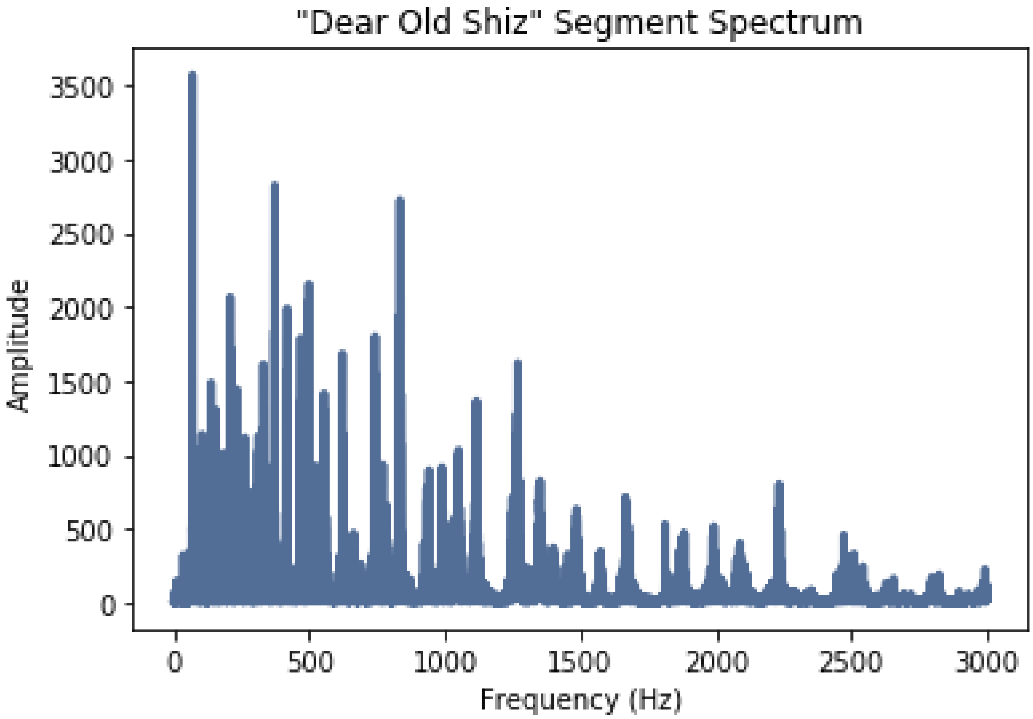
```
In [10]: # Converting into Spectrogram and plotting the "Dear Old Shiz" Segment

         # The spectrogram shows the harmonic structure over time.
         shiz_gram = shiz_short.make_spectrogram(seg_length=400)
         shiz_gram.plot(high=3000)
         thinkplot.config(xlabel='Time (s)', ylabel='Frequency (Hz)', title='"Dear Old Shiz" Se
```



```
In [11]: # Plotting the Spectrum for the segment of "For Good"

         spectrum = good_short.make_spectrum()
         spectrum.plot(high=3000)
         thinkplot.config(xlabel='Frequency (Hz)', ylabel='Amplitude', title='"For Good" Segmen
```



```
In [12]: # Plotting the Spectrum for the segment of "Dear Old Shiz"

         spectrum = shiz_short.make_spectrum()
         spectrum.plot(high=3000)
         thinkplot.config(xlabel='Frequency (Hz)', ylabel='Amplitude', title='"Dear Old Shiz" S
```


"Dear Old Shiz" Segment Spectrum
Amplitude
3500
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000
Frequency (Hz)

```
In [ ]:
```