In [1]:
```
import librosa
audio_path = 'Black_Footed_Albatross_0016_796067_9.mp3'
x , sr = librosa.load(audio_path)
print(type(x), type(sr))
```

```
<class 'numpy.ndarray'> <class 'int'>
```

In [2]:
```
import IPython.display as ipd
ipd.Audio(audio_path)
```

Out[2]:

In [29]:
```
#display waveform
%matplotlib inline
import matplotlib.pyplot as plt
import librosa.display
plt.figure(figsize=(14, 5))
librosa.display.waveshow(x, sr=sr)
```

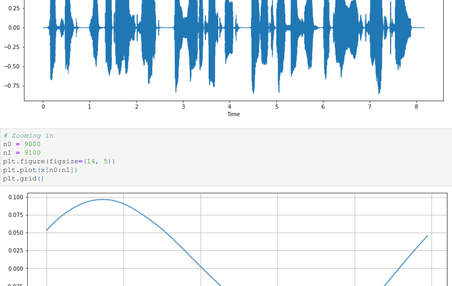
Out[29]:

```
<librosa.display.AdaptiveWaveplot at 0x2922f7d6850>
```

In [19]:
```
#display Spectrogram
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
#If to pring log of frequencies
#librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
plt.colorbar()
```

Out[19]:

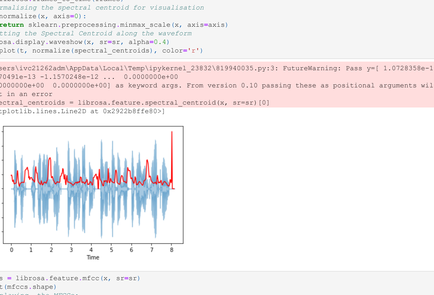
```
<matplotlib.colorbar.Colorbar at 0x2922aa55f40>
```



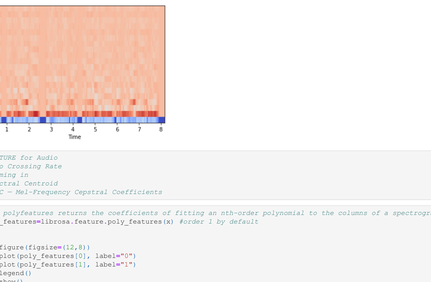
In [21]:
```
x, sr = librosa.load(audio_path)
#Plot the signal:
plt.figure(figsize=(14, 5))
librosa.display.waveshow(x, sr=sr)
```

Out[21]:

```
<librosa.display.AdaptiveWaveplot at 0x2922ab99130>
```



In [22]:
```
# Zooming in
n0 = 9000
n1 = 9100
plt.figure(figsize=(14, 5))
plt.plot(x[n0:n1])
plt.grid()
```

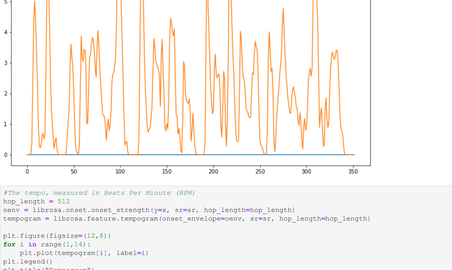


In [24]:
```
#spectral centroid -- centre of mass -- weighted mean of the frequencies present in the sound
import sklearn
spectral_centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]
spectral_centroids.shape
# Computing the time variable for visualization
frames = range(len(spectral_centroids))
t = librosa.frames_to_time(frames)
# Normalising the spectral centroid for visualisation
def normalize(x, axis=0):
    return sklearn.preprocessing.minmax_scale(x, axis=axis)
#Plotting the Spectral Centroid along the waveform
librosa.display.waveshow(x, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_centroids), color='r')
```

```
C:\Users\irc21262adm\AppData\Local\Temp\ipykernel_23832\819940035.py:3: FutureWarning: Pass y=[ 1.0728358e-13 -
6.2670491e-13 -1.1570248e-12 ...  0.0000000e+00
  0.0000000e+00  0.0000000e+00] as keyword args. From version 0.10 passing these as positional arguments will r
esult in an error
  spectral_centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]
```

Out[24]:

```
[<matplotlib.lines.Line2D at 0x2922b8ffe80>]
```



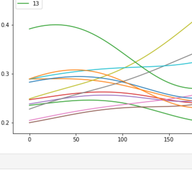
In [25]:
```
mfccs = librosa.feature.mfcc(x, sr=sr)
print(mfccs.shape)
#Displaying  the MFCCs:
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
```

```
(20, 352)
```

```
C:\Users\irc21262adm\AppData\Local\Temp\ipykernel_23832\1745963525.py:1: FutureWarning: Pass y=[ 1.0728358e-13
-6.2670491e-13 -1.1570248e-12 ...  0.0000000e+00
  0.0000000e+00  0.0000000e+00] as keyword args. From version 0.10 passing these as positional arguments will r
esult in an error
  mfccs = librosa.feature.mfcc(x, sr=sr)
```

Out[25]:

```
<matplotlib.collections.QuadMesh at 0x2922b958d00>
```



In [ ]:
```
#FEATURE for Audio
#Zero Crossing Rate
#Zooming in
#Spectral Centroid
#MFCC — Mel-Frequency Cepstral Coefficients
```

In [26]:
```
#The polyfeatures returns the coefficients of fitting an nth-order polynomial to the columns of a spectrogram
poly_features=librosa.feature.poly_features(x)  #order 1 by default


plt.figure(figsize=(12,8))
plt.plot(poly_features[0], label="0")
plt.plot(poly_features[1], label="1")
plt.legend()
plt.show()
```

```
C:\Users\irc21262adm\AppData\Local\Temp\ipykernel_23832\1064129351.py:1: FutureWarning: Pass y=[ 1.0728358e-13
-6.2670491e-13 -1.1570248e-12 ...  0.0000000e+00
  0.0000000e+00  0.0000000e+00] as keyword args. From version 0.10 passing these as positional arguments will r
esult in an error
  poly_features=librosa.feature.poly_features(x)  #order 1 by default
```


In [28]:
```
#The tempo, measured in Beats Per Minute (BPM)
hop_length = 512
oenv = librosa.onset.onset_strength(y=x, sr=sr, hop_length=hop_length)
tempogram = librosa.feature.tempogram(onset_envelope=oenv, sr=sr, hop_length=hop_length)

plt.figure(figsize=(12,8))
for i in range(1,14):
    plt.plot(tempogram[i], label=i)
plt.legend()
plt.title("Tempogram")
plt.show()
```


In [ ]: