```
import warnings
warnings.filterwarnings('ignore')
```

```
import librosa
```

```
y , sr = librosa.load(librosa.util.example_audio_file())
```

```
import IPython.display as ipd
```

```
print(type(y), type(sr))
```

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

```
print(y.shape,sr)
```

```
(1355168,) 22050
```

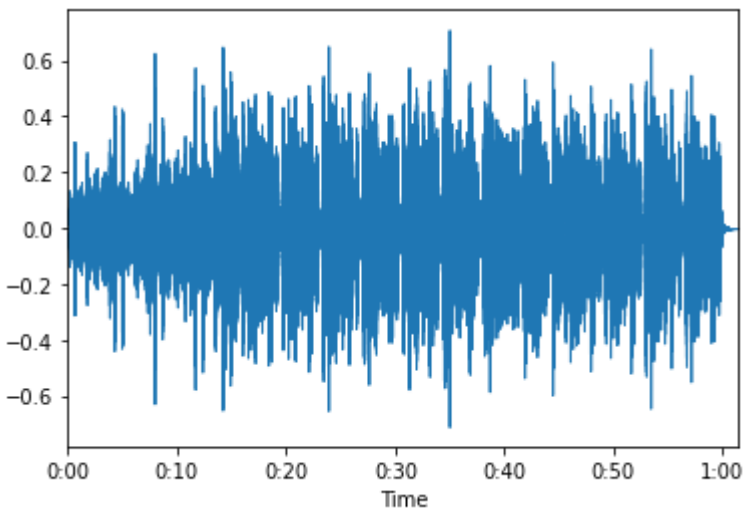
```
%matplotlib inline
import sklearn
import matplotlib.pyplot as plt
import librosa.display
```

```
plt.figure(figsize=(20, 5))
```

```
<Figure size 1440x360 with 0 Axes>
<Figure size 1440x360 with 0 Axes>
```

```
librosa.display.waveplot(y, sr=sr)
```

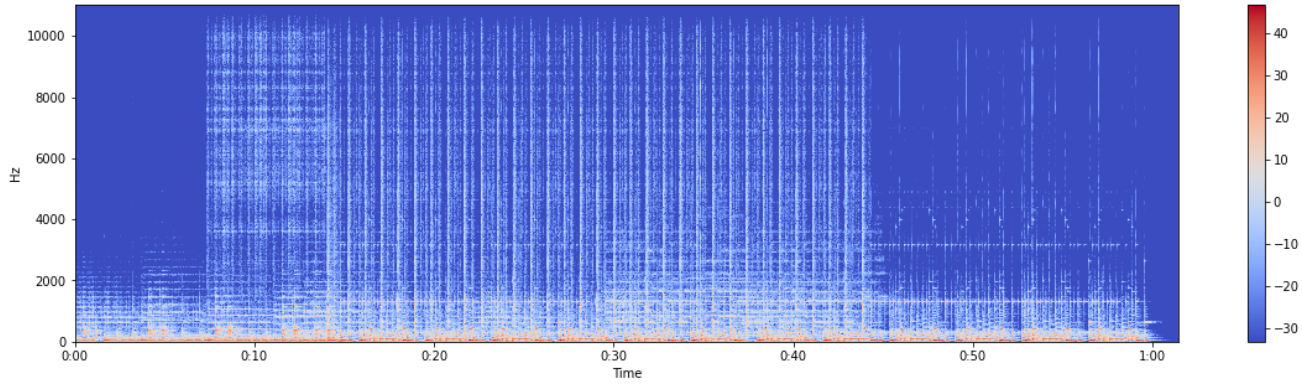
```
<matplotlib.collections.PolyCollection at 0x7fb904480350>
```




```
X = librosa.stft(y)
```

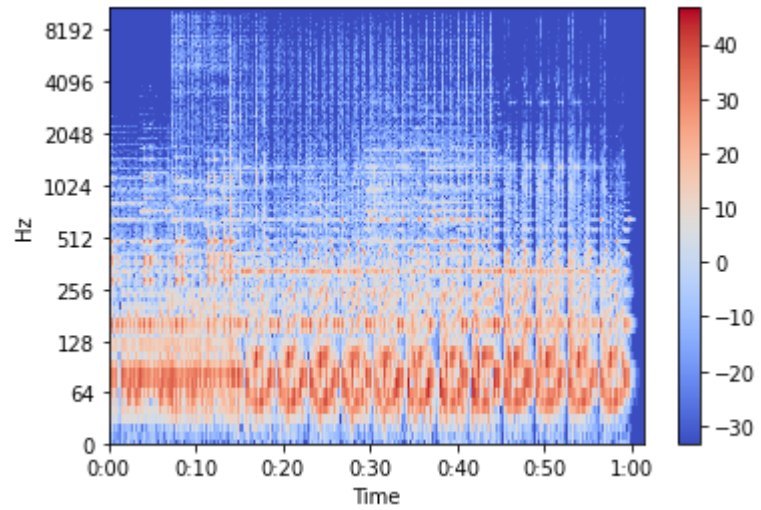
```
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(20, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7fb903f4cf50>
```




```
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7fb903630d90>
```

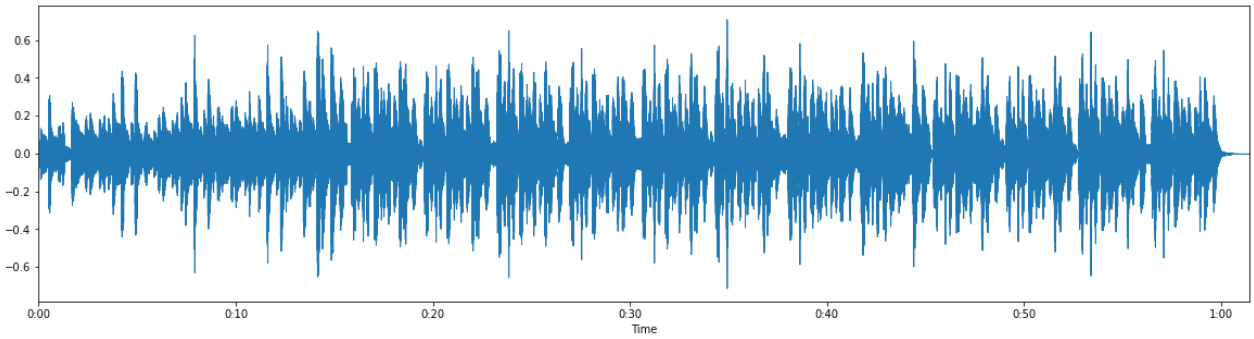



```
import numpy as np
sr = 22050 # sample rate
T = 5.0    # seconds
t = np.linspace(0, T, int(T*sr), endpoint=False) # time variable
x = 0.5*np.sin(2*np.pi*220*t)


ipd.Audio(x, rate=sr)
```

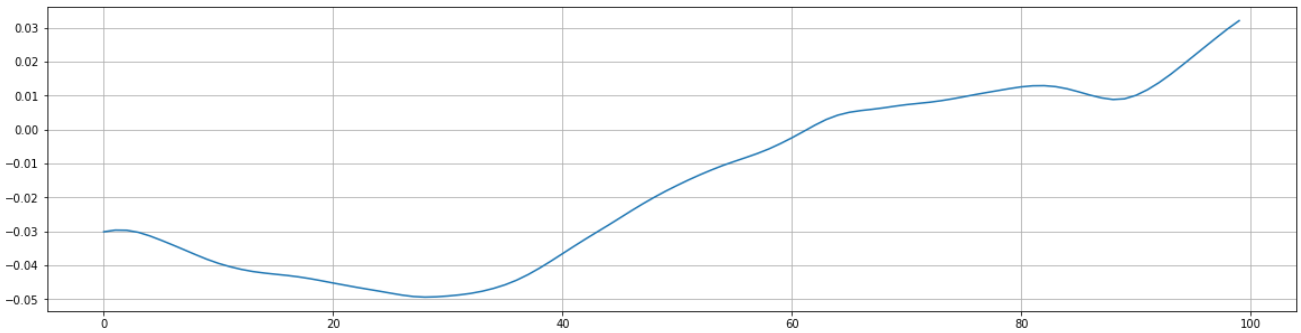
0:00 / 0:05

```
plt.figure(figsize=(20, 5))
librosa.display.waveplot(y, sr=sr)
```

```
<matplotlib.collections.PolyCollection at 0x7fb903299b50>
```




```
n0 = 9000
n1 = 9100
plt.figure(figsize=(20, 5))
plt.plot(y[n0:n1])
plt.grid()
```



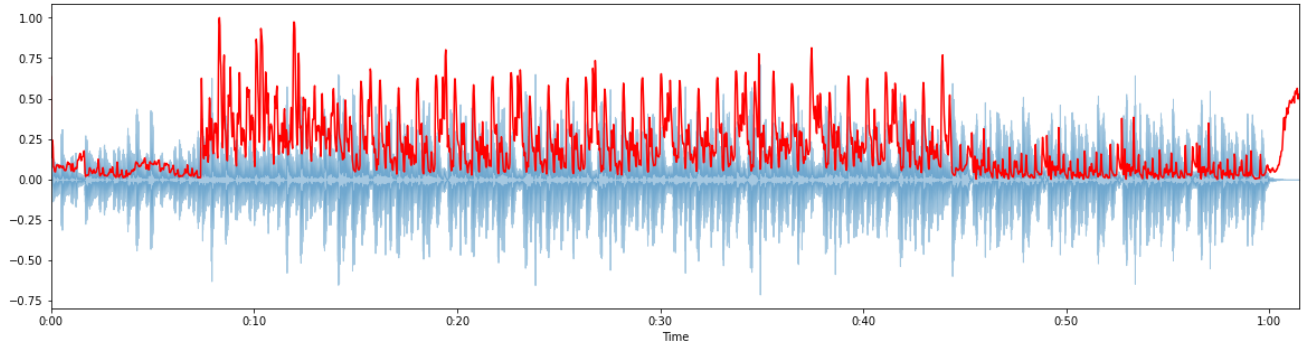

```
spectral_centroids = librosa.feature.spectral_centroid(y, sr=sr)[0]
spectral_centroids.shape
```

```
(2647,)
```

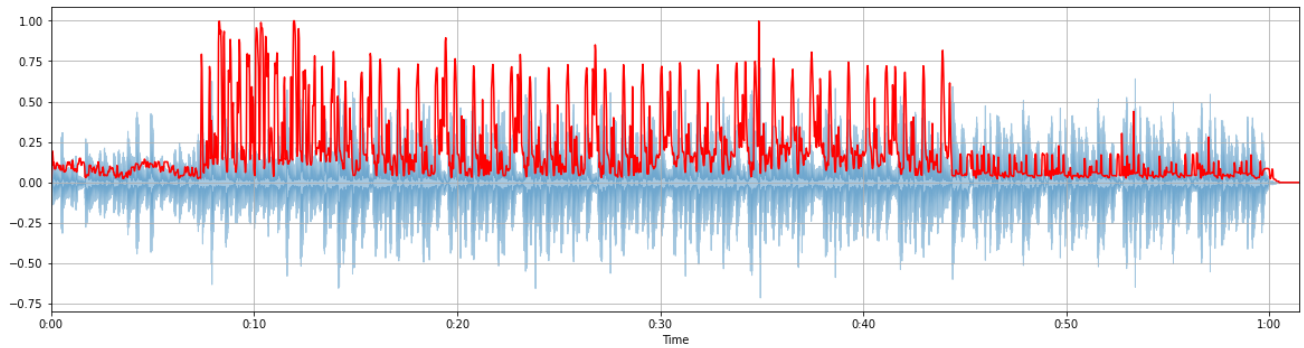
```
plt.figure(figsize=(20,5))
frames = range(len(spectral_centroids))
t = librosa.frames_to_time(frames)

# Normalising the spectral centroid for visualisation
def normalize(y, axis=0):
    return sklearn.preprocessing.minmax_scale(y, axis=axis)
```

```
#Plotting the Spectral Centroid along the waveform
librosa.display.waveplot(y, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_centroids), color='r')
```

```
[<matplotlib.lines.Line2D at 0x7fb904766c50>]
```

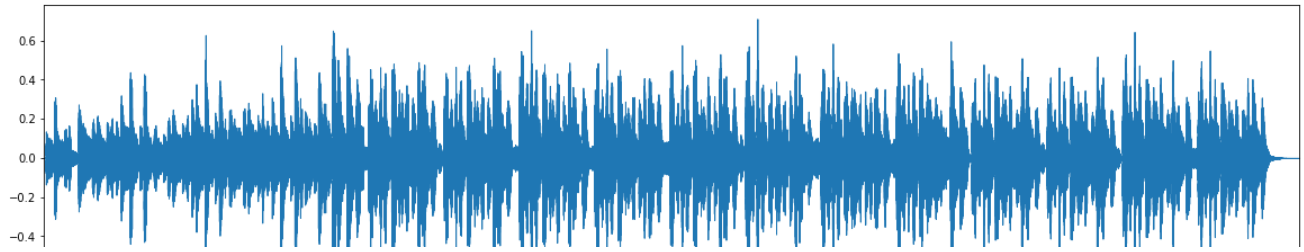



```
plt.figure(figsize=(20,5))
spectral_rolloff = librosa.feature.spectral_rolloff(y+0.01, sr=sr)[0]
librosa.display.waveplot(y, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_rolloff), color='r')
plt.grid()
```



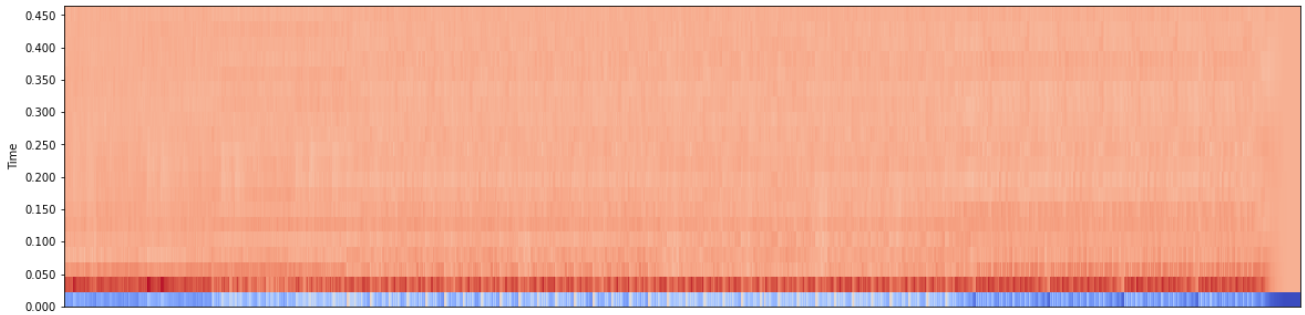

```
plt.figure(figsize=(20,5))
librosa.display.waveplot(y, sr=sr)
```

```
<matplotlib.collections.PolyCollection at 0x7fb904744110>
```




```
plt.figure(figsize=(20,5))
mfccs = librosa.feature.mfcc(y, sr=sr)
print(mfccs.shape)

librosa.display.specshow(mfccs, sr=sr, y_axis='time')
```

```
(20, 2647)
<matplotlib.collections.QuadMesh at 0x7fb904645b10>
```




```
mfccs = sklearn.preprocessing.scale(mfccs, axis=1)
print(mfccs.mean(axis=1))
print(mfccs.var(axis=1))
```

```
[ 5.7645595e-09  0.0000000e+00  5.7645595e-09  0.0000000e+00
  0.0000000e+00  5.7645595e-09  1.4411399e-08  0.0000000e+00
  2.8822797e-09 -8.6468397e-09 -1.1529119e-08 -2.8822797e-09
  0.0000000e+00  0.0000000e+00  0.0000000e+00  5.7645595e-09
  1.1529119e-08  0.0000000e+00  0.0000000e+00  0.0000000e+00]
[1.        1.0000002 1.        1.        1.        0.9999999 0.9999999
 1.        1.0000002 1.        1.        1.        1.        1.
 1.0000002 1.        1.        1.0000001 1.0000001 1.0000001]
```

```
plt.figure(figsize=(20,8))
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
```

```
<matplotlib.collections.QuadMesh at 0x7fb9046e4990>
```