A. Amplitude envelope and loudness

```
import librosa
import librosa.display
import IPython.display as ipd
import matplotlib.pyplot as plt
```

```
!pip install reportlab
```

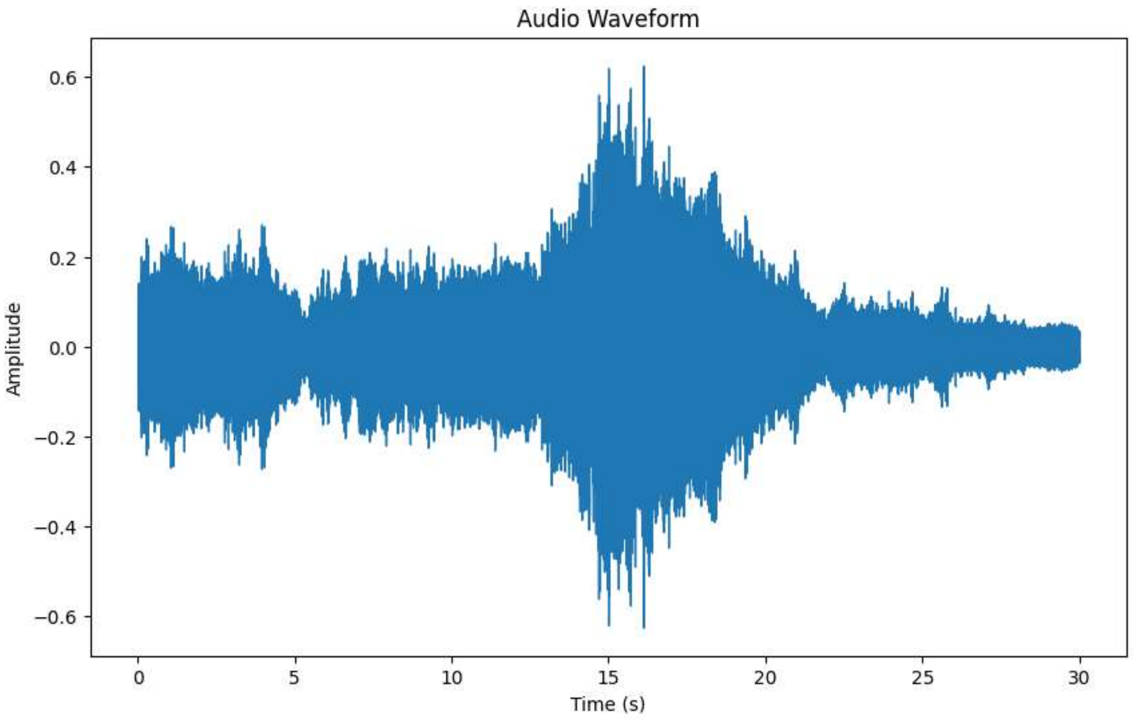
```
    Collecting reportlab
      Downloading reportlab-4.0.6-py3-none-any.whl (1.9 MB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 18.8 MB/s eta 0:00:00
    Requirement already satisfied: pillow>=9.0.0 in /usr/local/lib/python3.10/dist-packages (from reportlab) (9.4.0)
    Installing collected packages: reportlab
    Successfully installed reportlab-4.0.6
```

```
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
```

```
# Step 2: Load and Play Audio
audio_file = "Audio_file.wav"
audio, sr = librosa.load(audio_file)
ipd.Audio(audio, rate=sr)
```

0:00 / 0:30

```
plt.figure(figsize=(10, 6))
librosa.display.waveshow(audio, sr=sr)
plt.title("Audio Waveform")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()
```

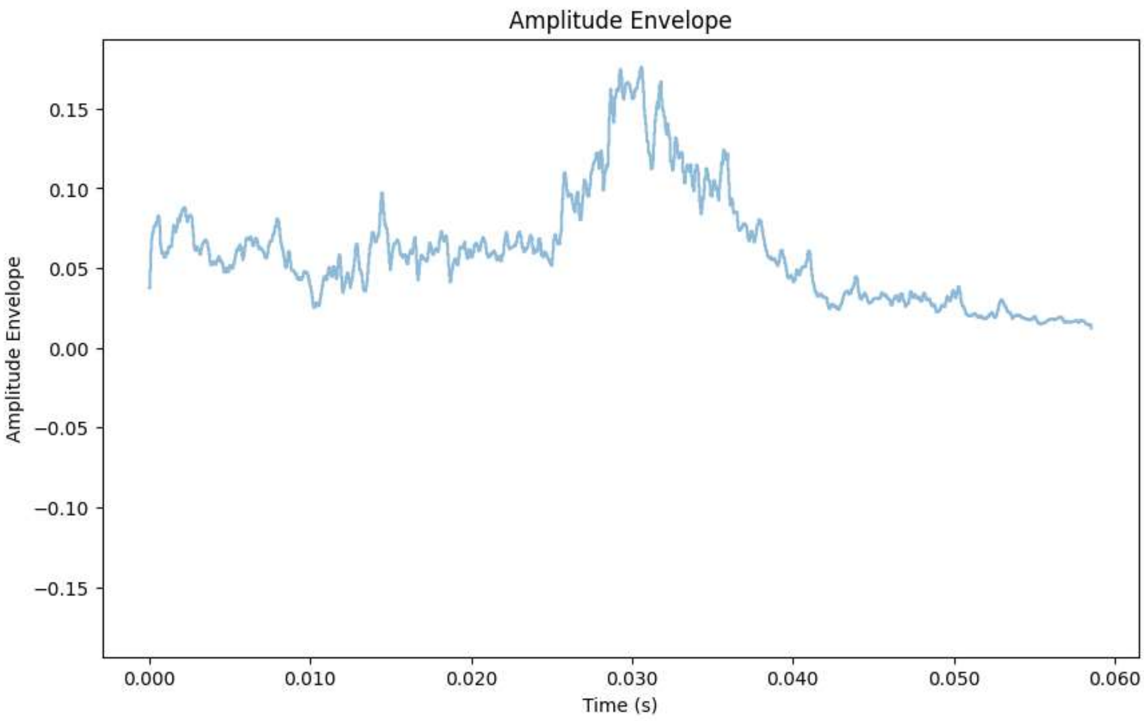


```
amplitude_envelope = librosa.feature.rms(y=audio)
num_frames = amplitude_envelope.shape[1]
```

```
num_frames = amplitude_envelope.shape[1]
```

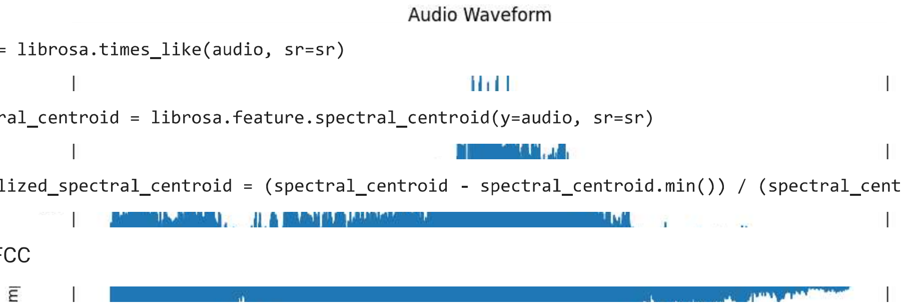
```
amplitude_envelope = librosa.feature.rms(y=audio, frame_length=2048, hop_length=512)
```

```
plt.figure(figsize=(10, 6))
librosa.display.waveshow(amplitude_envelope, sr=sr, alpha=0.5)
plt.title("Amplitude Envelope")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude Envelope")
plt.show()
```



B. Spectral centroid

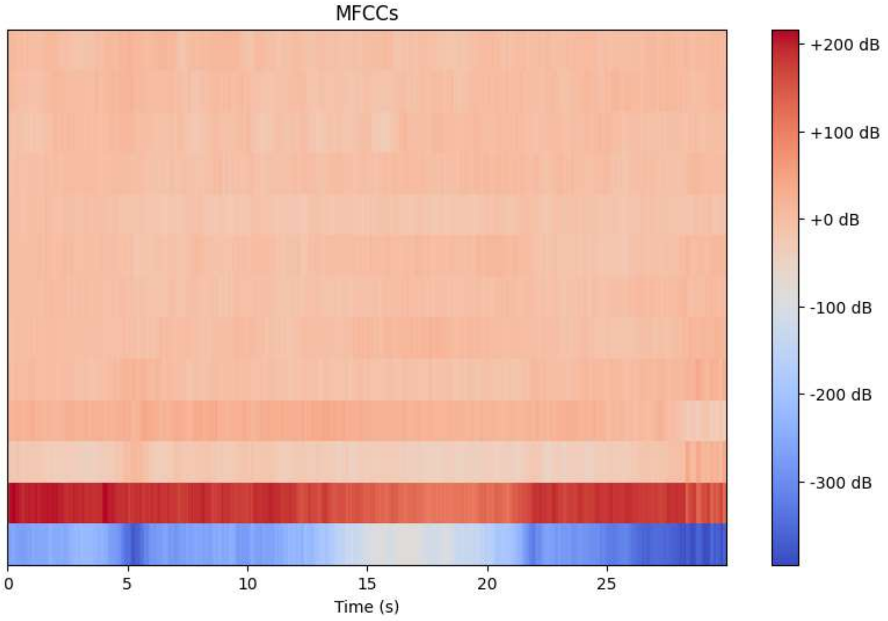
```
plt.figure(figsize=(10, 6))
librosa.display.waveshow(audio, sr=sr)
plt.title("Audio Waveform")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()
```



```
time = librosa.times_like(audio, sr=sr)

spectral_centroid = librosa.feature.spectral_centroid(y=audio, sr=sr)

normalized_spectral_centroid = (spectral_centroid - spectral_centroid.min()) / (spectral_centroid.max() - spectral_centroid.min())
```

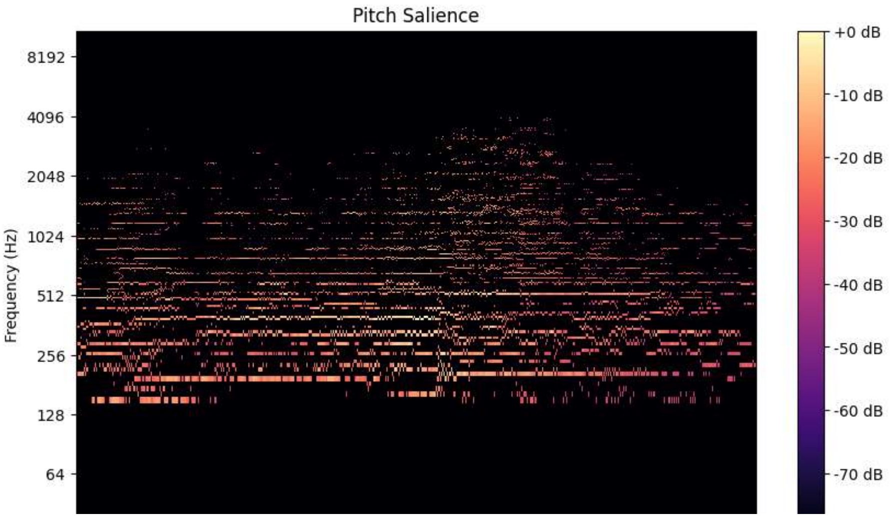
C. MFCC

```
mfccs = librosa.feature.mfcc(y=audio, sr=sr, n_mfcc=13)

plt.figure(figsize=(10, 6))
librosa.display.specshow(mfccs, x_axis='time', sr=sr)
plt.colorbar(format='%+2.0f dB')
plt.title("MFCCs")
plt.xlabel("Time (s)")
plt.show()
```



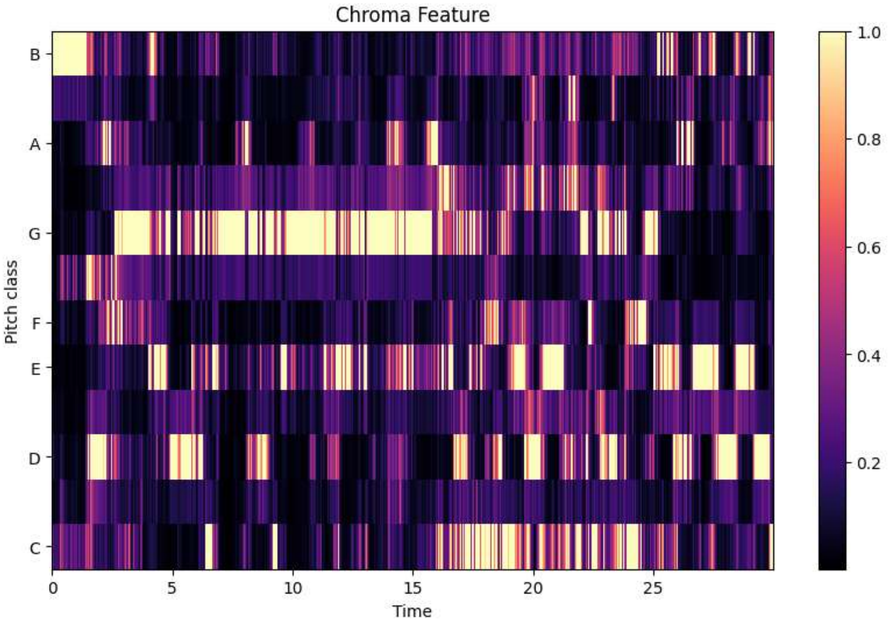
D. Pitch salience

```
import numpy as np
pitches, magnitudes = librosa.piptrack(y=audio, sr=sr)


plt.figure(figsize=(10, 6))
librosa.display.specshow(librosa.amplitude_to_db(magnitudes, ref=np.max), y_axis='log', x_axis='time')
plt.colorbar(format='%+2.0f dB')
plt.title("Pitch Salience")
plt.xlabel("Time (s)")
plt.ylabel("Frequency (Hz)")
plt.show()
```



E. Chroma

```
import librosa
import librosa.display
import matplotlib.pyplot as plt

# Step 2: Load audio file with librosa library
audio_file = "Audio_file.wav"
y, sr = librosa.load(audio_file)

# Step 3: Perform Chroma feature extraction using librosa library
chroma = librosa.feature.chroma_stft(y=y, sr=sr)

# Step 4: Display Output graph
plt.figure(figsize=(10, 6))
librosa.display.specshow(chroma, y_axis='chroma', x_axis='time')
plt.colorbar()
plt.title('Chroma Feature')
plt.show()
```


Chroma Feature
Pitch class
B
A
G
F
E
D
C
Time