# EDA Visualizations and Audio Exploration

## Import Libraries

In [1]:
```
import ast
import librosa
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
import plotly.graph_objs as go
from collections import Counter
import matplotlib.pyplot as plt
from IPython.display import Audio
```

## EDA

In [2]:
```
trainmeta_df = pd.read_csv("/input/birdclef-2023/train_metadata.csv")
trainmeta_df.head()
```

Out[2]:

| | primary_label | secondary_labels | type | latitude | longitude | scientific_name | common_name | author | license | rating | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | abethr1 | [] | ['song'] | 4.3906 | 38.2788 | Turdus tephronotus | African Bare-eyed Thrush | Rolf A. de By | Creative Commons Attribution-NonCommercial-Sha... | 4.0 | https://www.canto.org/1 |
| **1** | abethr1 | [] | ['call'] | -2.9524 | 38.2921 | Turdus tephronotus | African Bare-eyed Thrush | James Bradley | Creative Commons Attribution-NonCommercial-Sha... | 3.5 | https://www.canto.org/3 |
| **2** | abethr1 | [] | ['song'] | -2.9524 | 38.2921 | Turdus tephronotus | African Bare-eyed Thrush | James Bradley | Creative Commons Attribution-NonCommercial-Sha... | 3.5 | https://www.canto.org/3 |
| **3** | abethr1 | [] | ['song'] | -2.9524 | 38.2921 | Turdus tephronotus | African Bare-eyed Thrush | James Bradley | Creative Commons Attribution-NonCommercial-Sha... | 5.0 | https://www.canto.org/3 |
| **4** | abethr1 | [] | ['call', 'song'] | -2.9524 | 38.2921 | Turdus tephronotus | African Bare-eyed Thrush | James Bradley | Creative Commons Attribution-NonCommercial-Sha... | 4.5 | https://www.canto.org/3 |

In [3]:
```
x = trainmeta_df[trainmeta_df["primary_label"] == "afgfly1"]
x
```

Out[3]:

| | primary_label | secondary_labels | type | latitude | longitude | scientific_name | common_name | author | license | rating | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **356** | afgfly1 | [] | ['song'] | -7.5253 | 34.8521 | Bradornis microrhynchus | African Gray Flycatcher | Martin St-Michel | Creative Commons Attribution-NonCommercial-Sha... | 3.0 | http car |
| **357** | afgfly1 | [] | ['call'] | 8.9200 | 40.0430 | Bradornis microrhynchus | African Gray Flycatcher | Andrew Spencer | Creative Commons Attribution-NonCommercial-Sha... | 5.0 | http car |
| **358** | afgfly1 | [] | ['call'] | 8.9200 | 40.0430 | Bradornis microrhynchus | African Gray Flycatcher | Andrew Spencer | Creative Commons Attribution-NonCommercial-Sha... | 5.0 | http car |
| **359** | afgfly1 | [] | ['call'] | -1.5765 | 36.6316 | Bradornis microrhynchus | African Gray Flycatcher | James Bradley | Creative Commons Attribution-NonCommercial-Sha... | 5.0 | http car |
| **360** | afgfly1 | ['combul2', 'kerspa2', 'ratcis1'] | ['call,begging call,juvenile'] | -2.8145 | 37.4113 | Bradornis microrhynchus | African Gray Flycatcher | Rory Nefdt | Creative Commons Attribution-NonCommercial-Sha... | 3.5 | http car |
| **361** | afgfly1 | [] | ['adult', 'call', 'sex uncertain'] | -3.1481 | 36.6951 | Bradornis microrhynchus | African Gray Flycatcher | isaac kilusu | Creative Commons Attribution-NonCommercial-Sha... | 4.0 | http car |
| **362** | afgfly1 | [] | ['call', 'juvenile', 'sex uncertain'] | -3.1481 | 36.6951 | Bradornis microrhynchus | African Gray Flycatcher | isaac kilusu | Creative Commons Attribution-NonCommercial-Sha... | 5.0 | http car |
| **363** | afgfly1 | [] | ['adult', 'call', 'sex uncertain'] | -3.1481 | 36.6951 | Bradornis microrhynchus | African Gray Flycatcher | isaac kilusu | Creative Commons Attribution-NonCommercial-Sha... | 5.0 | http car |

## Train Meta Data

In [4]:
```
print(trainmeta_df.info())
```

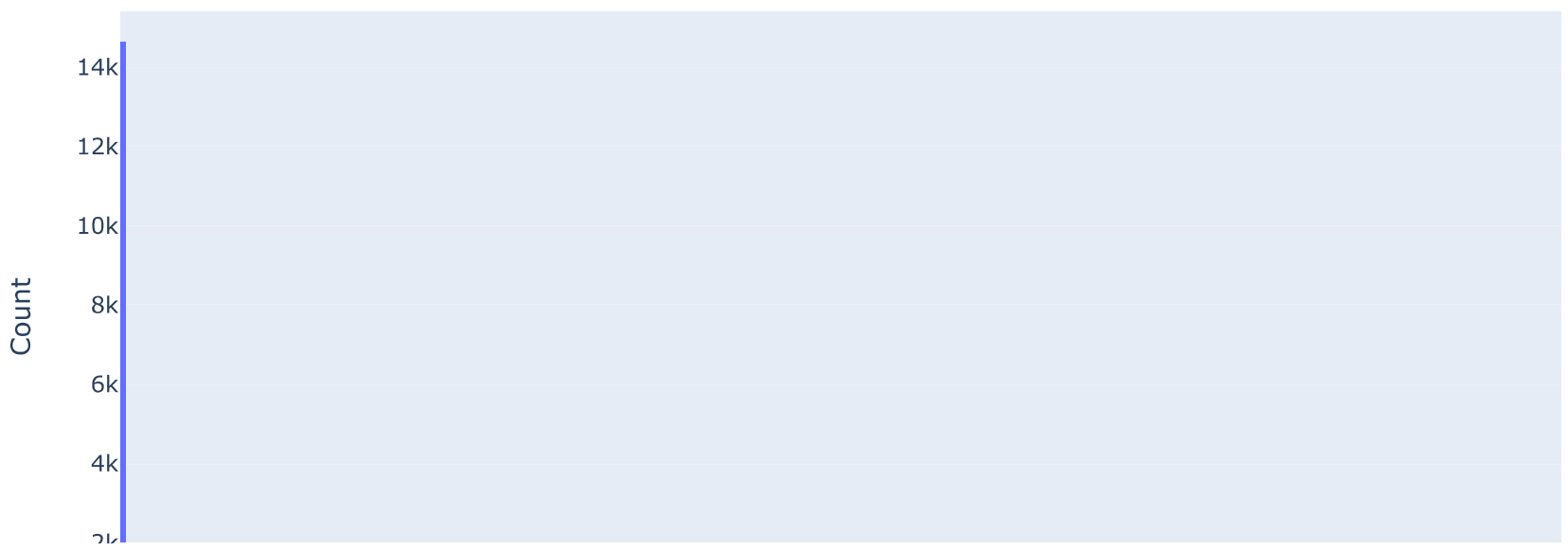
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16941 entries, 0 to 16940
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   primary_label     16941 non-null  object 
 1   secondary_labels  16941 non-null  object 
 2   type              16941 non-null  object 
 3   latitude          16714 non-null  float64
 4   longitude         16714 non-null  float64
 5   scientific_name   16941 non-null  object 
 6   common_name       16941 non-null  object 
 7   author            16941 non-null  object 
 8   license           16941 non-null  object 
 9   rating            16941 non-null  float64
 10  url               16941 non-null  object 
 11  filename          16941 non-null  object 
dtypes: float64(3), object(9)
memory usage: 1.6+ MB
None
```

## Plots

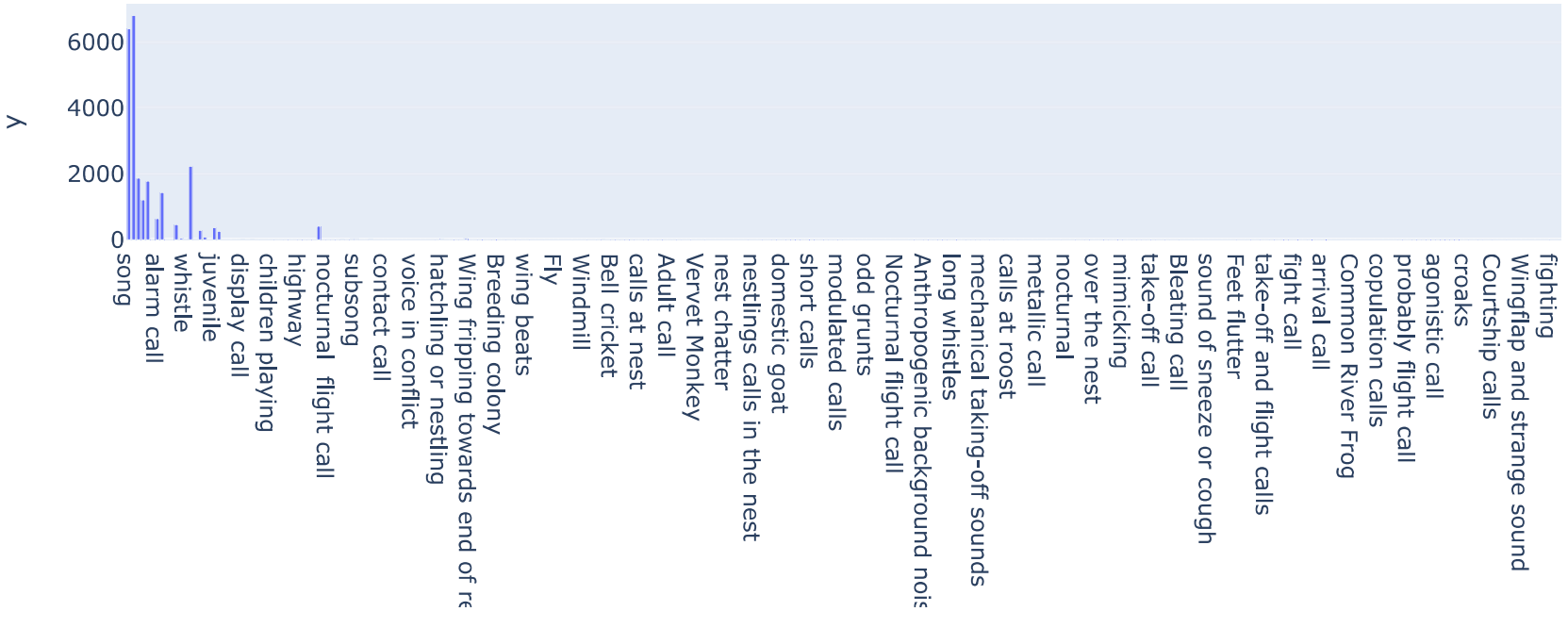
In [5]:
```
fig = px.histogram(trainmeta_df, x="primary_label", nbins=len(trainmeta_df["primary_label"].unique()))
fig.update_layout(title_text="Distribution of Primary Labels")
fig.show()
```


In [6]:
```
secondary_df = trainmeta_df['secondary_labels'].str.replace('[','').str.replace(']','').str.replace('\'','').str.split(
secondary_df = pd.merge(trainmeta_df.drop(columns=['secondary_labels']), secondary_df, left_index=True, right_index=Tru

fig = px.histogram(secondary_df, x="secondary_label", title="Distribution of Secondary Labels")
fig.update_layout(xaxis_title="Secondary Label", yaxis_title="Count")
fig.show()
```

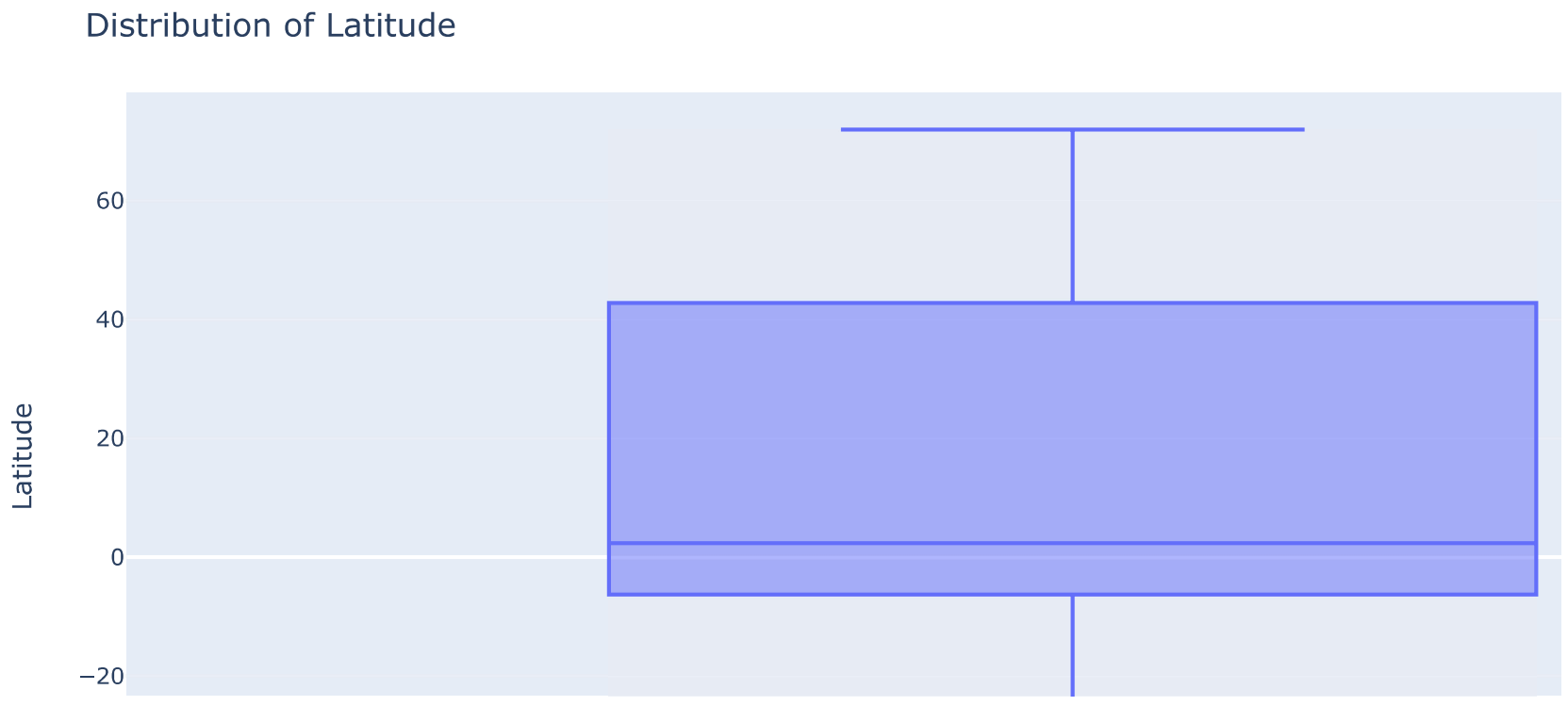
```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:

The default value of regex will change from True to False in a future version. In addition, single character regular ex
pressions will *not* be treated as literal strings when regex=True.
```

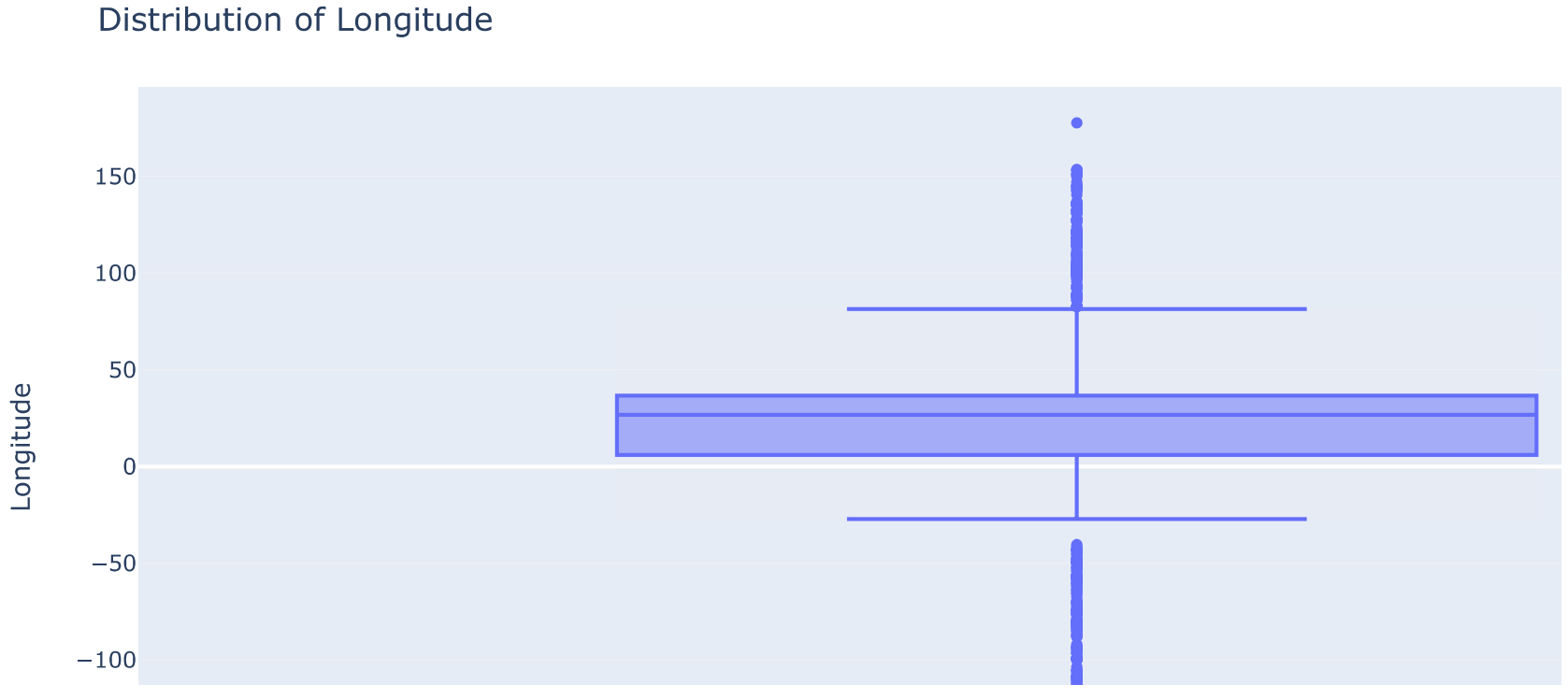


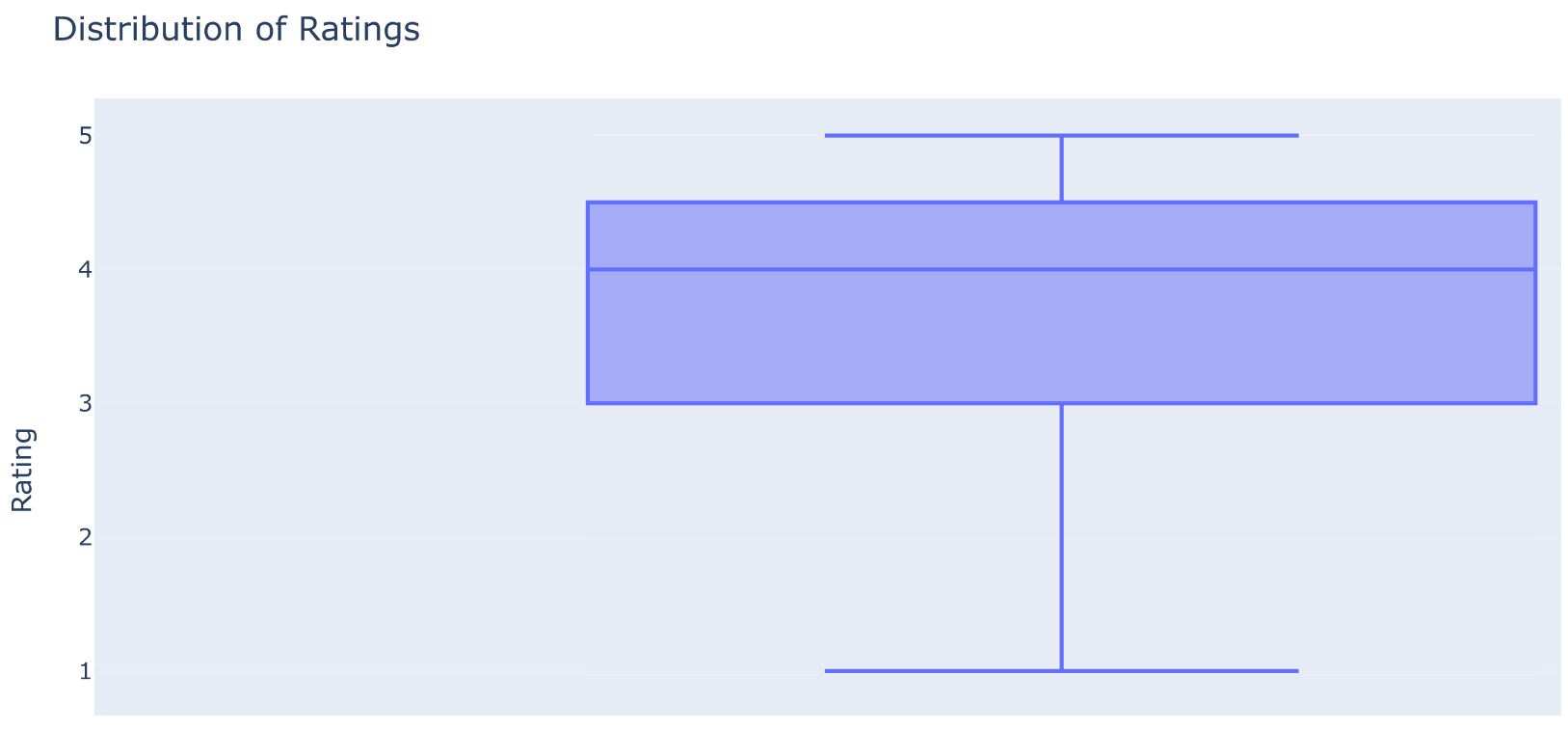

In [7]:
```
# Flatten the list of labels in the "type" column
labels = [label.strip("[]'") for sublist in trainmeta_df['type'].apply(ast.literal_eval) for label in sublist]

# Count the occurrence of each label
label_counts = Counter(labels)

# Create a bar plot of the label counts
fig = px.bar(x=list(label_counts.keys()), y=list(label_counts.values()))
fig.update_layout(title_text="Distribution of Types")
fig.show()
```



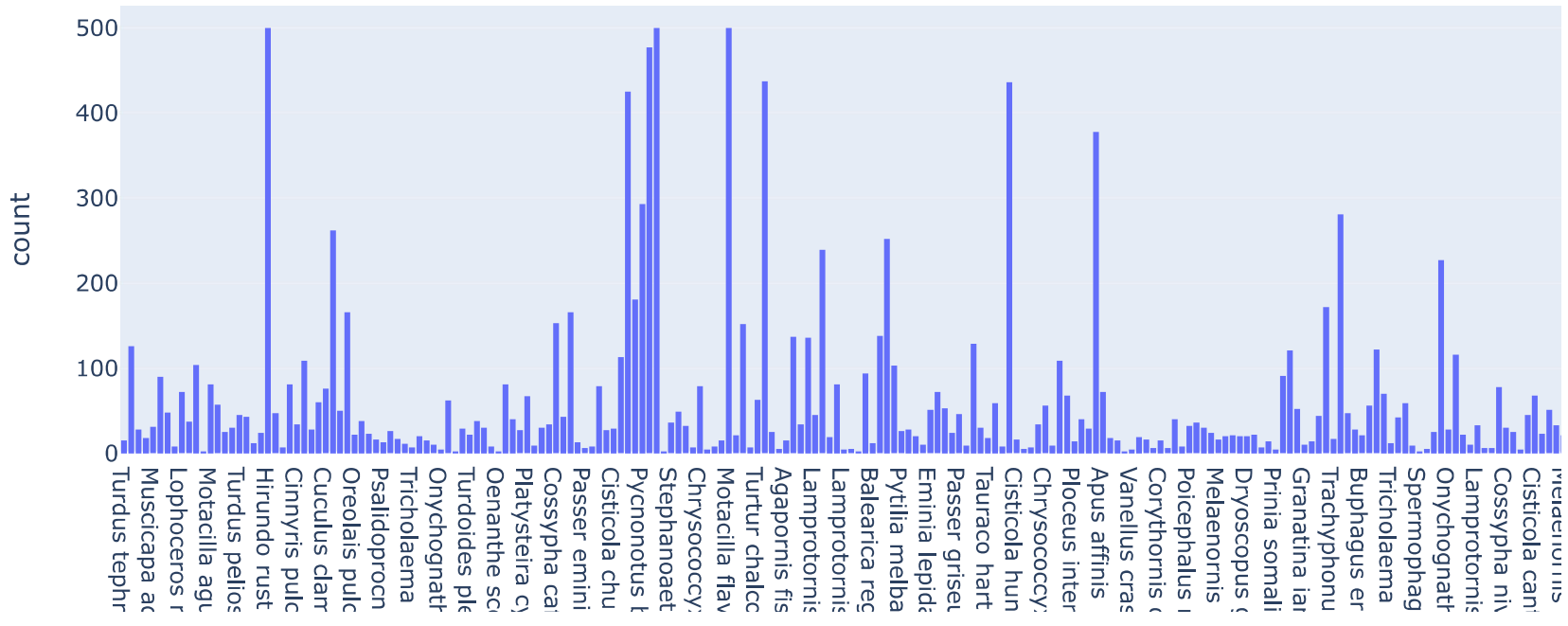

In [8]:
```
fig1 = px.box(trainmeta_df, y="latitude")
fig1.update_layout(title_text="Distribution of Latitude",
                   yaxis=dict(title="Latitude"))

# create a box plot for longitude
fig2 = px.box(trainmeta_df, y="longitude")
fig2.update_layout(title_text="Distribution of Longitude",
                   yaxis=dict(title="Longitude"))

# create a box plot for rating
fig3 = px.box(trainmeta_df, y="rating")
fig3.update_layout(title_text="Distribution of Ratings",
                   yaxis=dict(title="Rating"))

# show the figures
fig1.show()
fig2.show()
fig3.show()
```



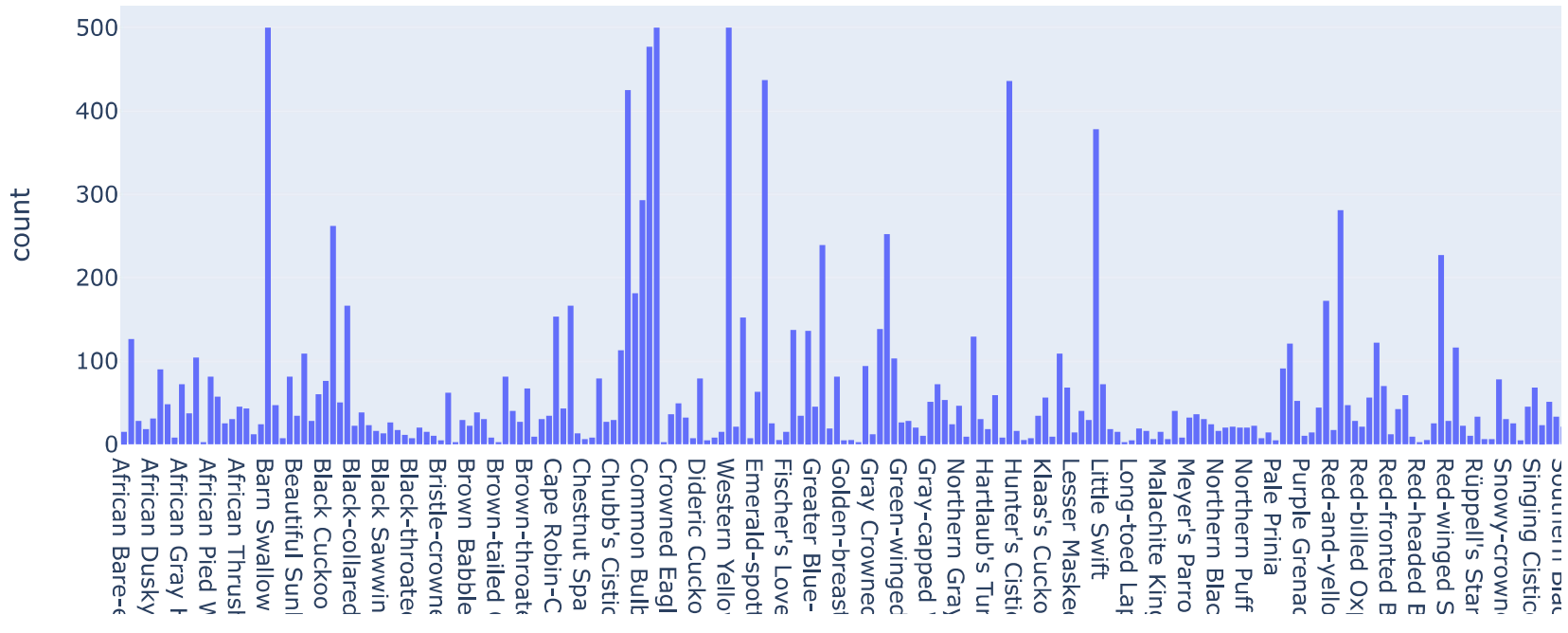


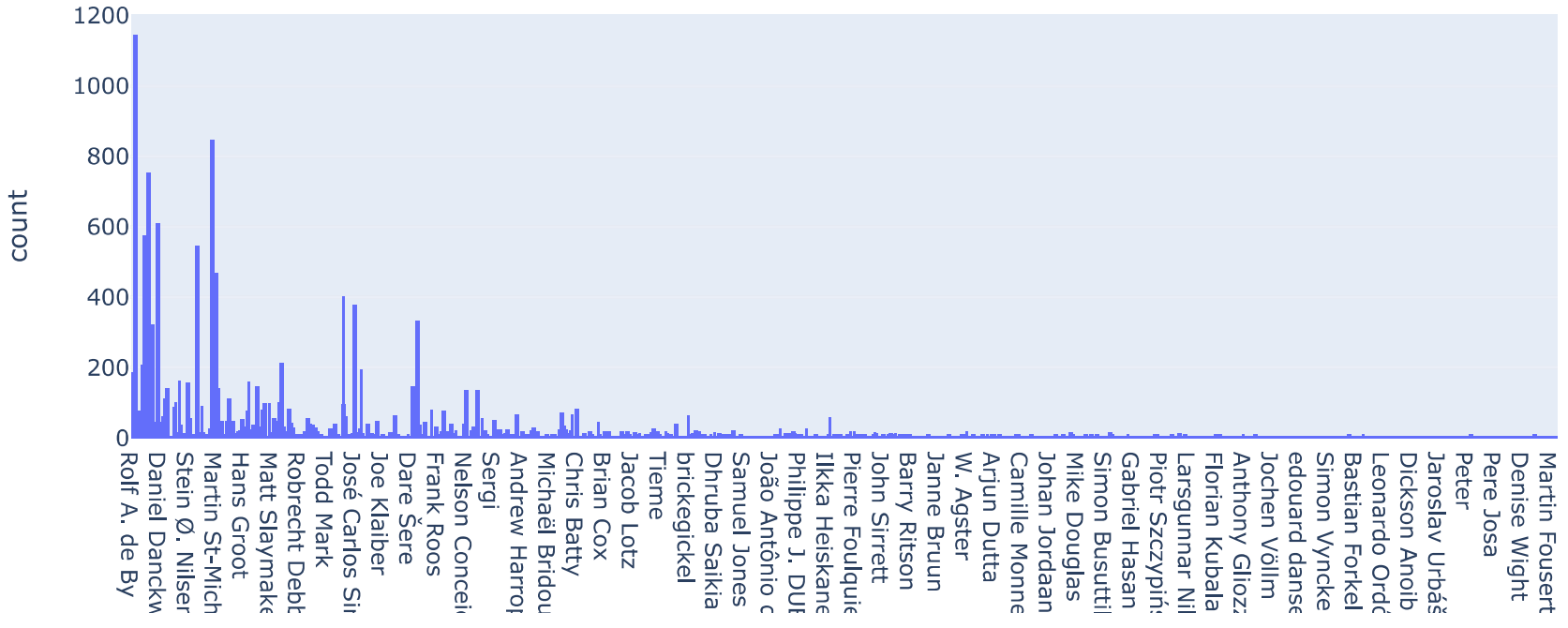

In [9]:
```
fig = px.histogram(trainmeta_df, x="scientific_name", nbins=len(trainmeta_df["scientific_name"].unique()))
fig.update_layout(title_text="Distribution of Scientific Names")
fig.show()
```



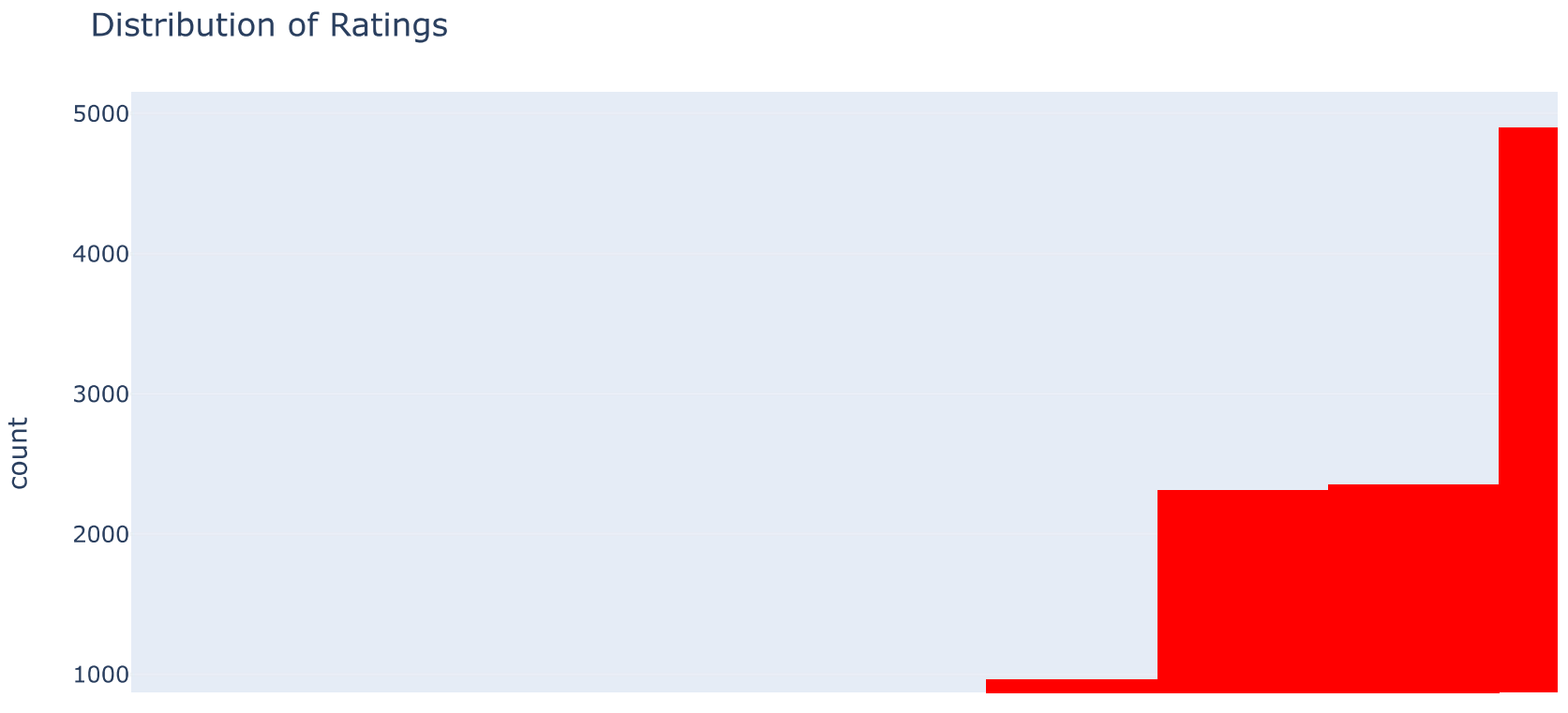

In [10]:
```
fig = px.histogram(trainmeta_df, x="common_name", nbins=len(trainmeta_df["common_name"].unique()))
fig.update_layout(title_text="Distribution of Common Names")
fig.show()
```




In [11]:
```
fig = px.histogram(trainmeta_df, x="author", nbins=len(trainmeta_df["author"].unique()))
fig.update_layout(title_text="Distribution of Authors")
fig.show()
```



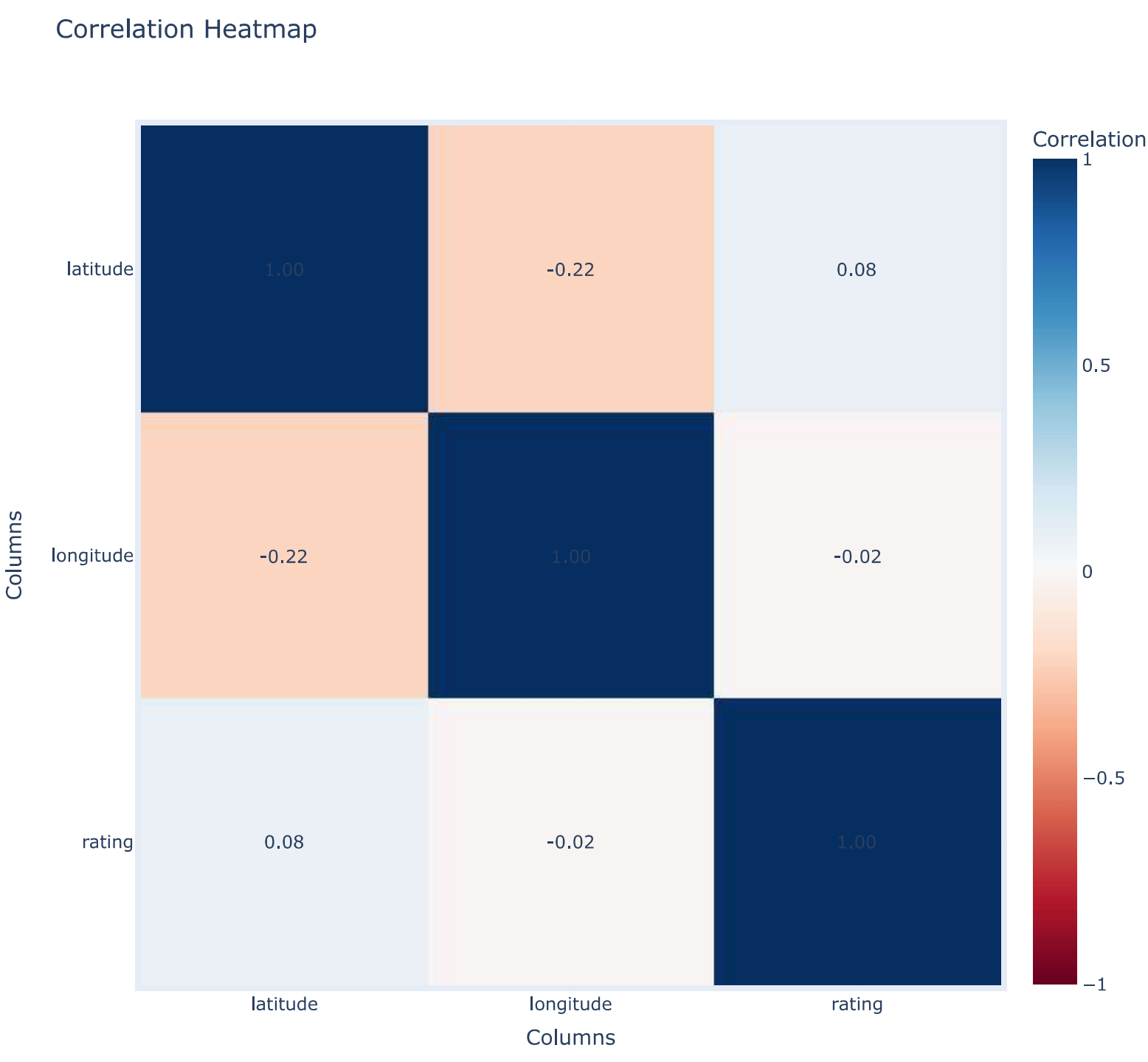

In [12]:
```
fig = px.histogram(trainmeta_df, x="rating", nbins=len(trainmeta_df["rating"].unique()) , color_discrete_sequence=['red
fig.update_layout(title_text="Distribution of Ratings")
fig.show()
```




## Correlation Plots

In [13]:
```
# drop columns from correlation matrix
corr = trainmeta_df.corr()

# create correlation heatmap
fig = px.imshow(corr,
                labels=dict(x="Columns", y="Columns", color="Correlation"),
                x=corr.columns,
                y=corr.columns,
                color_continuous_scale='RdBU',
                zmin=-1,
                zmax=1,
                title="Correlation Heatmap")

# add text annotations
annotations = []
for i, row in enumerate(corr.values):
    for j, value in enumerate(row):
        text = '{:.2f}'.format(value)
        annotations.append(dict(x=corr.columns[j], y=corr.columns[i], text=text, showarrow=False))

fig.update_layout(width=800, height=800)
```

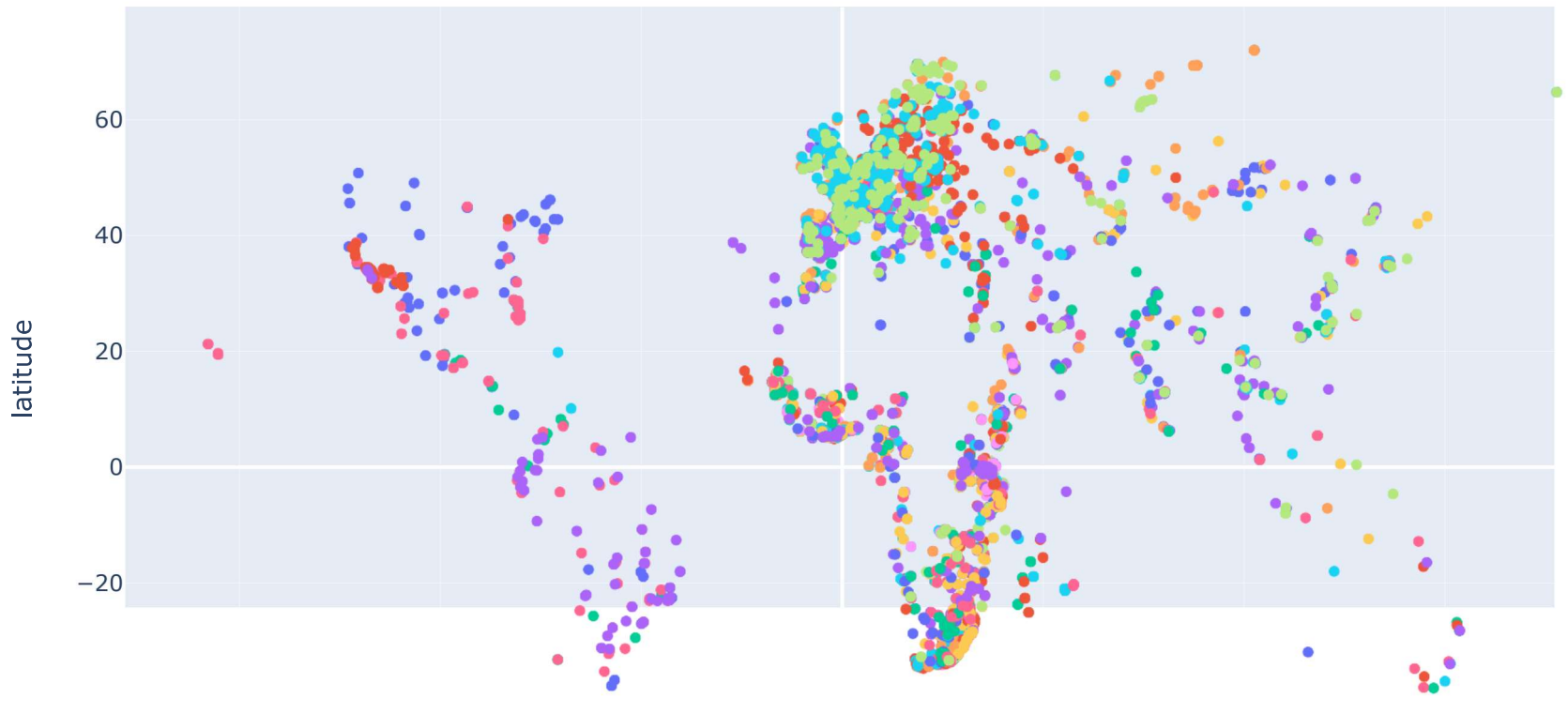
```
fig.update_traces(showscale=True, colorbar_thickness=25, colorbar_len=0.75)
fig.update_layout(margin=dict(l=50, r=50, b=100, t=100, pad=4))
fig.update_layout(annotations=annotations)
fig.show()
```




## Interactive Map Plots

### Scatter Plot

In [14]:
```
fig = px.scatter(trainmeta_df, x="longitude", y="latitude", color="common_name")
fig.update_layout(title="Distribution of Recordings by Location")
fig.show()
```




Map Box (Open Street Map)

In [15]:
```
fig = px.scatter_mapbox(trainmeta_df, lat="latitude", lon="longitude", color="common_name",
                        hover_name="filename", hover_data=["common_name", "author", "rating"],
                        zoom=3, height=500)
fig.update_layout(mapbox_style="open-street-map")
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

EBird Taxonomy

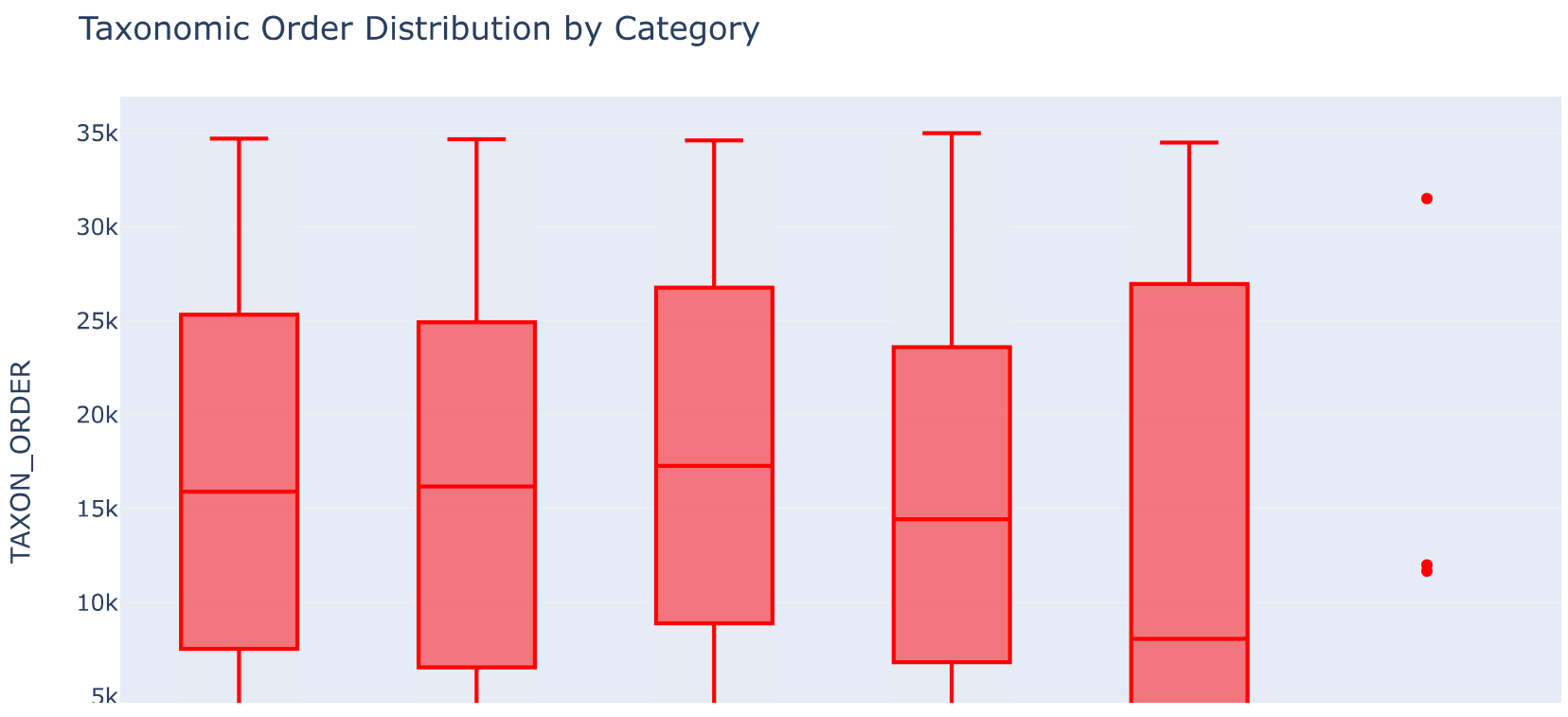
In [17]:
```
train_species_df = pd.read_csv("/input/birdclef-2023/eBird_Taxonomy_v2021.csv")
train_species_df.head()
```

Out[17]:

| | TAXON_ORDER | CATEGORY | SPECIES_CODE | PRIMARY_COM_NAME | SCI_NAME | ORDER1 | FAMILY | SPECIES_GROUP |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | species | ostric2 | Common Ostrich | Struthio camelus | Struthioniformes | Struthionidae (Ostriches) | Ostriches |
| **1** | 6 | species | ostric3 | Somali Ostrich | Struthio molybdophanes | Struthioniformes | Struthionidae (Ostriches) | NaN |
| **2** | 7 | slash | y00934 | Common/Somali Ostrich | Struthio camelus/molybdophanes | Struthioniformes | Struthionidae (Ostriches) | NaN |
| **3** | 8 | species | grerhe1 | Greater Rhea | Rhea americana | Rheiformes | Rheidae (Rheas) | Rheas |
| **4** | 14 | species | lesrhe2 | Lesser Rhea | Rhea pennata | Rheiformes | Rheidae (Rheas) | NaN |

In [18]:

```
# Histogram of the taxonomic order counts
fig1 = px.histogram(train_species_df, x="TAXON_ORDER", color_discrete_sequence=['aquamarine'])
fig1.update_layout(title_text="Distribution of Taxonomic Orders")

# Bar plot of the species group counts
# Box plot of the taxonomic order counts by category
fig3 = px.box(train_species_df, x="CATEGORY", y="TAXON_ORDER", color_discrete_sequence=['red'])
fig3.update_layout(title_text="Taxonomic Order Distribution by Category")

# Scatter plot of the taxonomic order counts by family
fig4 = px.scatter(train_species_df, x="FAMILY", y="TAXON_ORDER")
fig4.update_layout(title_text="Taxonomic Order Distribution by Family")

# Show the plots
fig1.show()
fig3.show()
fig4.show()
```







## AUDIO EXPLORATION

**Audio Files :** An audio file is a type of digital file format that stores recorded sound or music. It can be played back through speakers or headphones and is commonly used in a variety of applications, such as music, film, television, radio, and other forms of media. Audio files come in many different formats, including **MP3, WAV, OGG, AAC, and FLAC.**

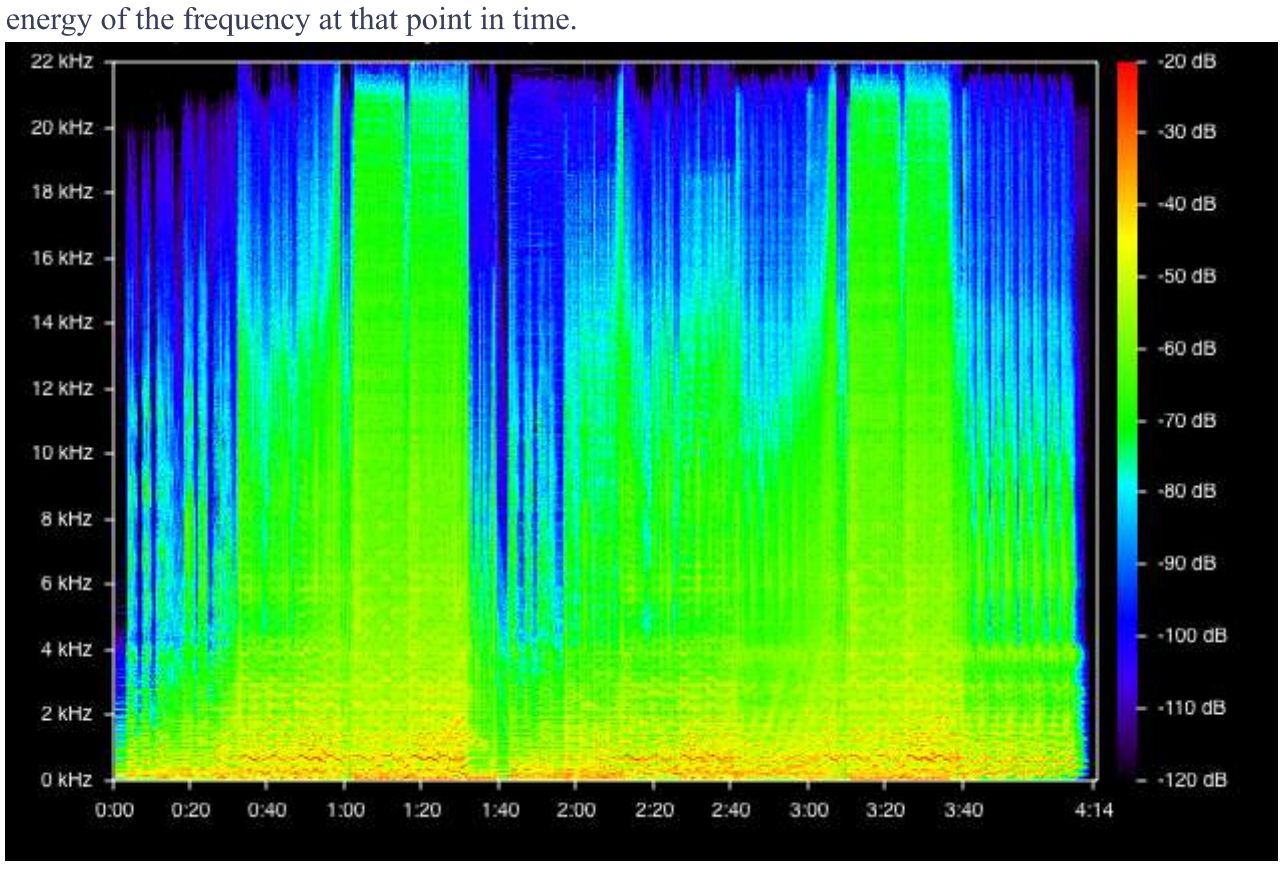
### How to Visualize Audio Files ?

### There are many ways in which we can view audio in 2D like :

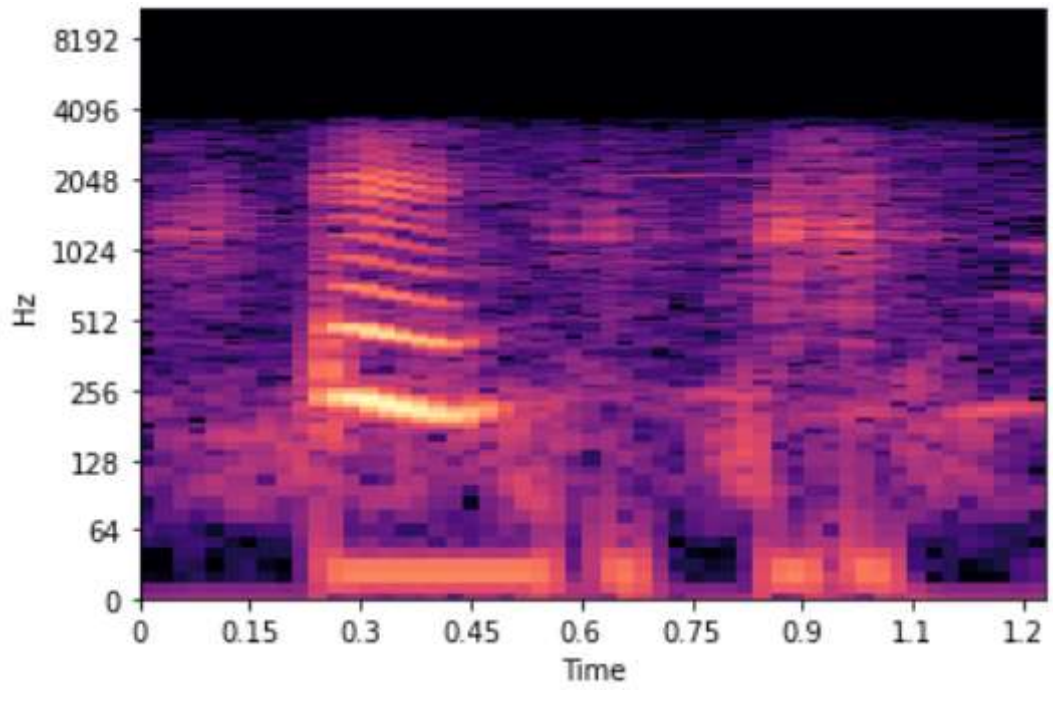
1. **Waveforms :** In audio processing, a waveform is a graphical representation of a sound signal that shows how the signal varies over time. It is a plot of the amplitude of the sound wave on the y-axis versus time on the x-axis. Waveforms can be

used to visualize and analyze the properties of audio signals, such as frequency, amplitude, phase, and duration.

2. **Spectograms :** A spectrogram is a visual representation of the spectrum of frequencies of a sound or other signal as it varies with time. It is a way of analyzing the sound signal to understand how the signal changes over time and what frequencies are present in the signal at any given point in time. The x-axis of a spectrogram represents time, while the y-axis represents frequency. The intensity of the color or brightness of each point in the spectrogram represents the amplitude or energy of the frequency at that point in time.

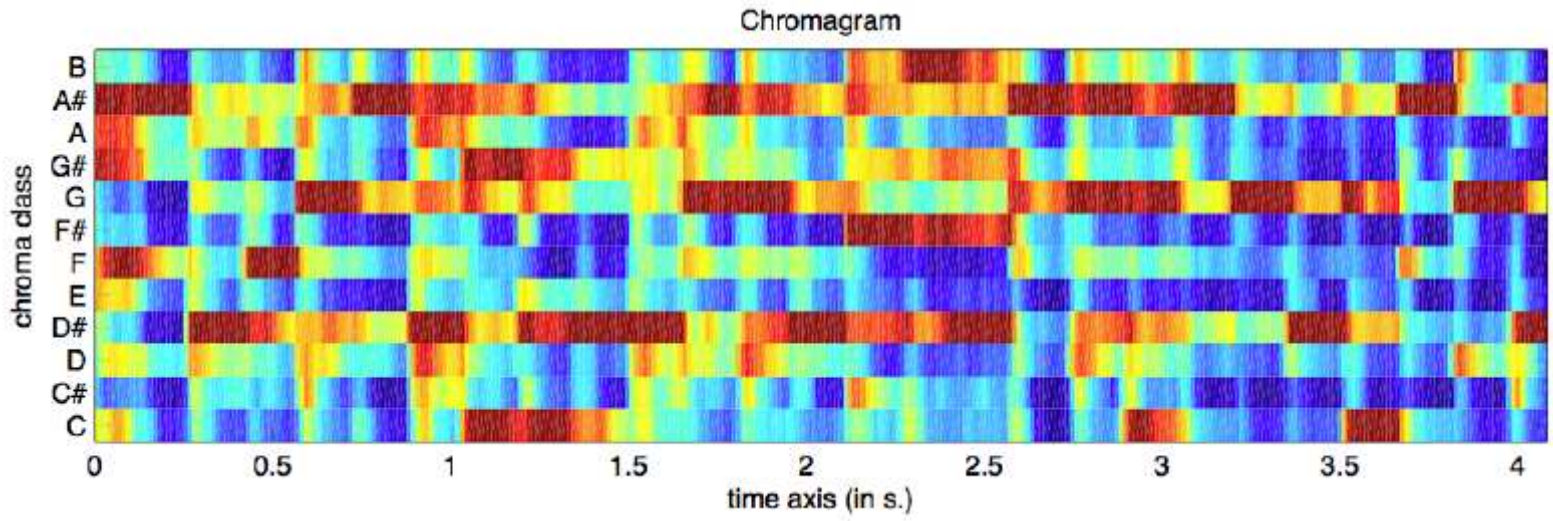



3. **Mel Spectograms :** Mel spectrograms are a type of spectrogram that use the Mel scale to determine the spacing of frequency bins. The Mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. By using the Mel scale, the frequency bins are spaced closer together at lower frequencies and farther apart at higher frequencies, which better matches the way humans perceive sound.

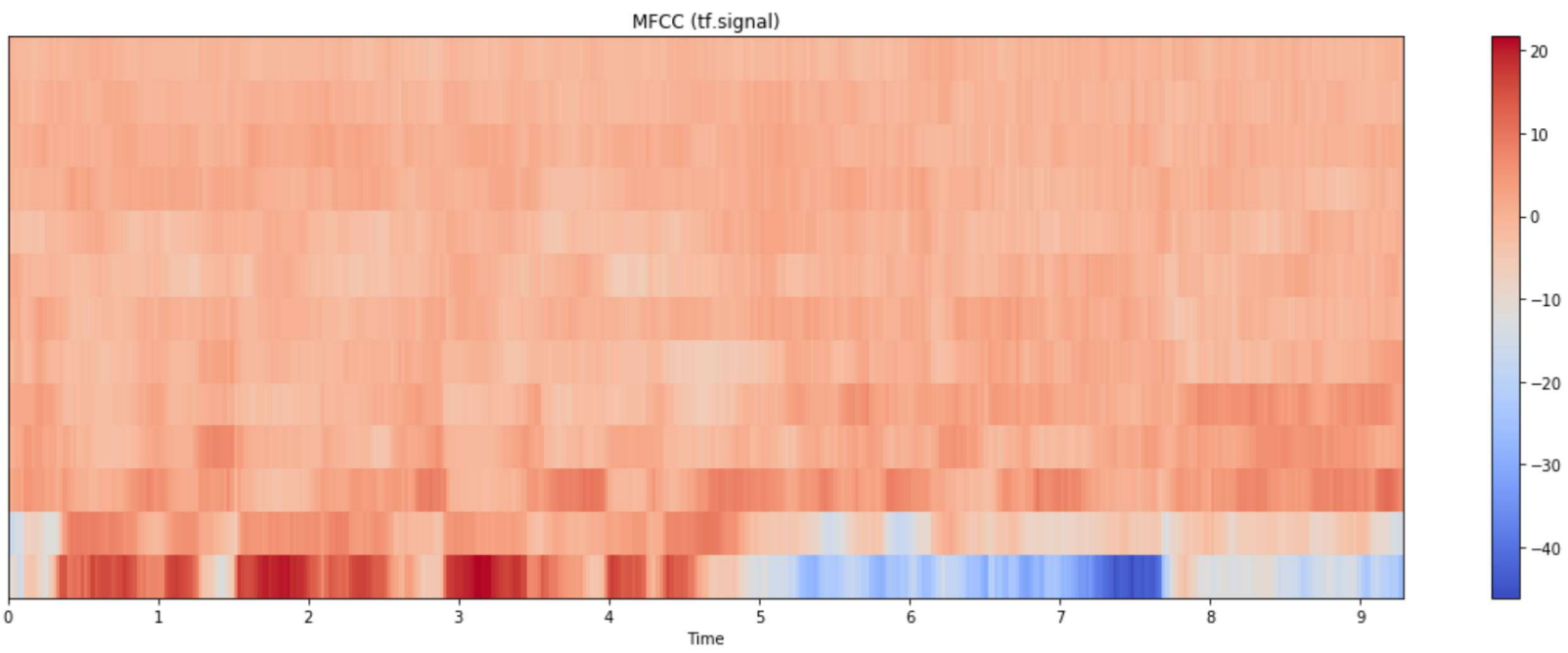



4. **Chromagram :** A chromagram is a visual representation of the pitch content of an audio signal, where each column corresponds to a particular frequency band or pitch class. It is a 2D representation of the distribution of energy in a sound, plotted as a function of time and pitch class.The chromagram is calculated by taking the short-time Fourier transform (STFT) of the audio signal and projecting the resulting spectrogram onto a pitch-class basis, typically using a bank of 12 log-spaced

filters to represent the 12 pitch classes of the Western musical scale




5. **MFCC :** MFCC stands for Mel Frequency Cepstral Coefficients, which is a feature extraction technique commonly used in speech processing, music information retrieval, and other related fields.MFCCs are derived from a short-time Fourier transform (STFT) of the audio signal, which involves breaking the signal down into small, overlapping segments and performing a Fourier transform on each segment to obtain its frequency spectrum. The Mel scale is then applied to the frequency spectrum to transform it into a logarithmic scale that approximates the human auditory system's perception of sound.




Helper Function

In [19]:
```python
def audio_eda(audio_path):

    # Load an audio file
    samples, sample_rate = librosa.load(audio_path)

    # Visualize the waveform
    plt.figure(figsize=(14, 5))
    librosa.display.waveshow(samples, sr=sample_rate)
    plt.title('Waveform')

    # Compute the spectrogram
    spectrogram = librosa.stft(samples)
    spectrogram_db = librosa.amplitude_to_db(abs(spectrogram))

    # Visualize the spectrogram
    plt.figure(figsize=(14, 5))
    librosa.display.specshow(spectrogram_db, sr=sample_rate, x_axis='time', y_axis='log')
    plt.colorbar(format='%+2.0f dB')
    plt.title('Spectrogram (dB)')

    # Compute the mel spectrogram


    # Visualize the mel spectrogram
    S = librosa.feature.melspectrogram(y=samples, sr=sample_rate)

    # Visualize mel spectrogram
    plt.figure(figsize=(10, 4))
    librosa.display.specshow(librosa.power_to_db(S, ref=np.max), y_axis='mel', fmax=8000, x_axis='time')
    plt.colorbar(format='%+2.0f dB')
    plt.title('Mel spectrogram')
    plt.tight_layout()


    # Compute the chromagram
    chromagram = librosa.feature.chroma_stft( y = samples , sr = sample_rate)

    # Visualize the chromagram
    plt.figure(figsize=(14, 5))
    librosa.display.specshow(chromagram, sr=sample_rate, x_axis='time', y_axis='chroma')
    plt.colorbar()
    plt.title('Chromagram')
```

```
    # Compute the MFCCs
    mfccs = librosa.feature.mfcc(y=samples, sr=sample_rate, n_mfcc=13)

    # Visualize the MFCCs
    plt.figure(figsize=(14, 5))
    librosa.display.specshow(mfccs, sr=sample_rate, x_axis='time')
    plt.colorbar()
    plt.title('MFCCs')

    # Show the plots
    display(Audio(samples, rate=sample_rate))
    plt.show()
```

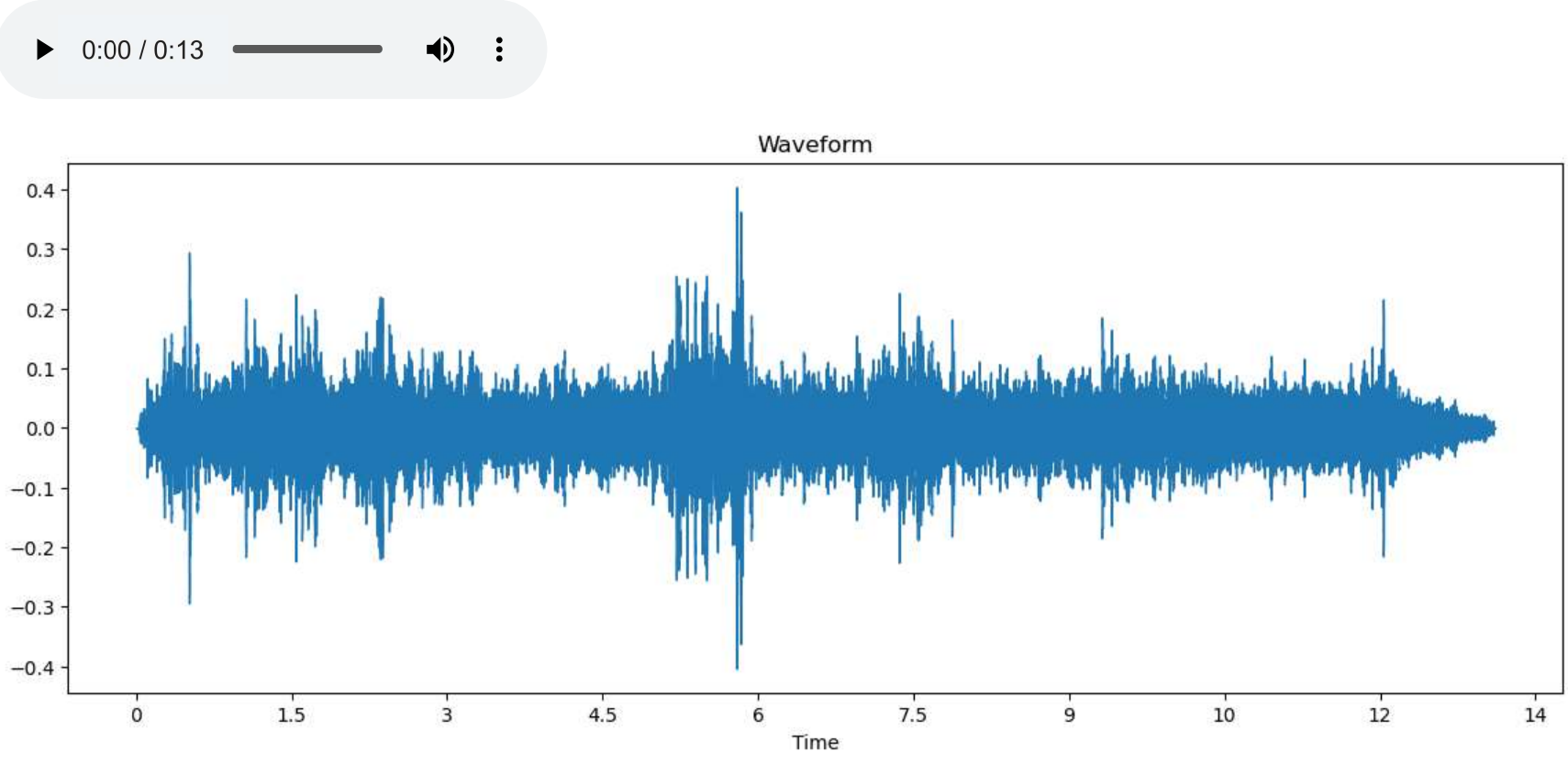
Audio Exploration

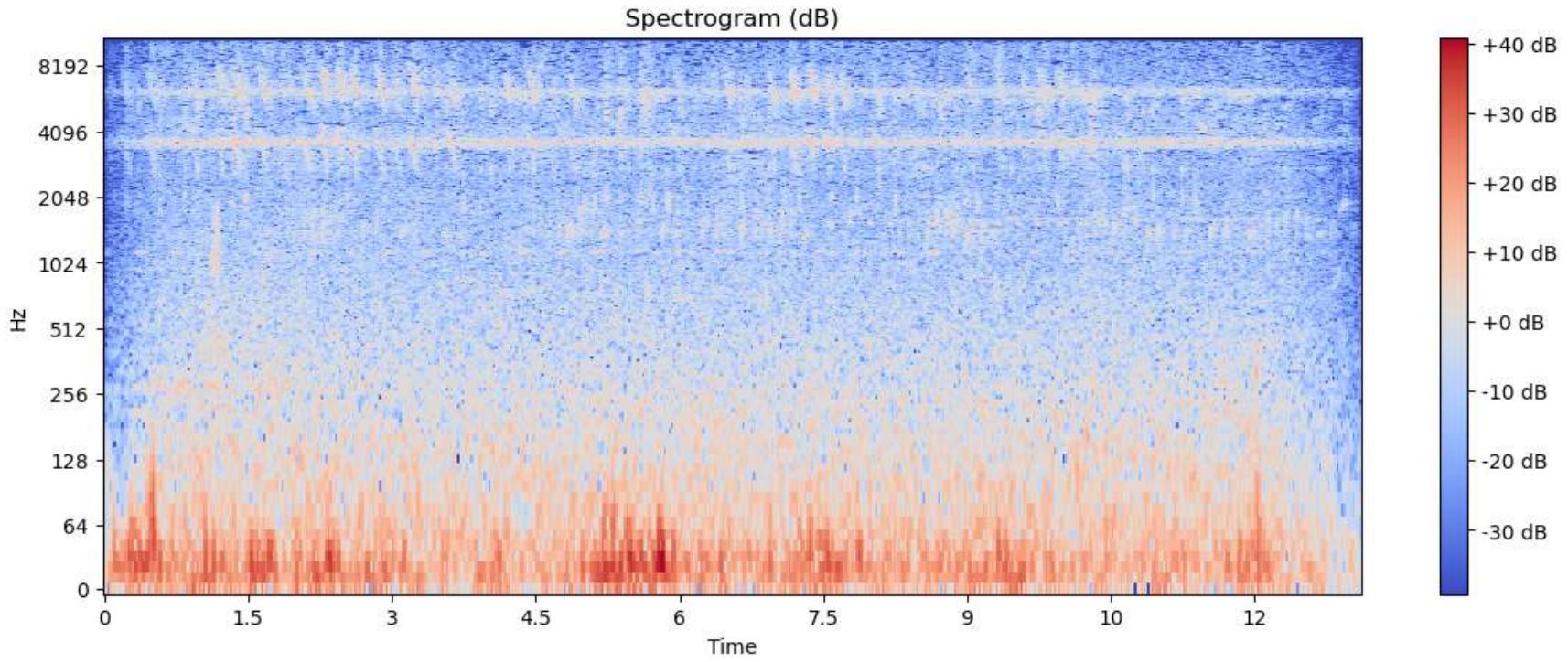
## Black-and-white Mannikin

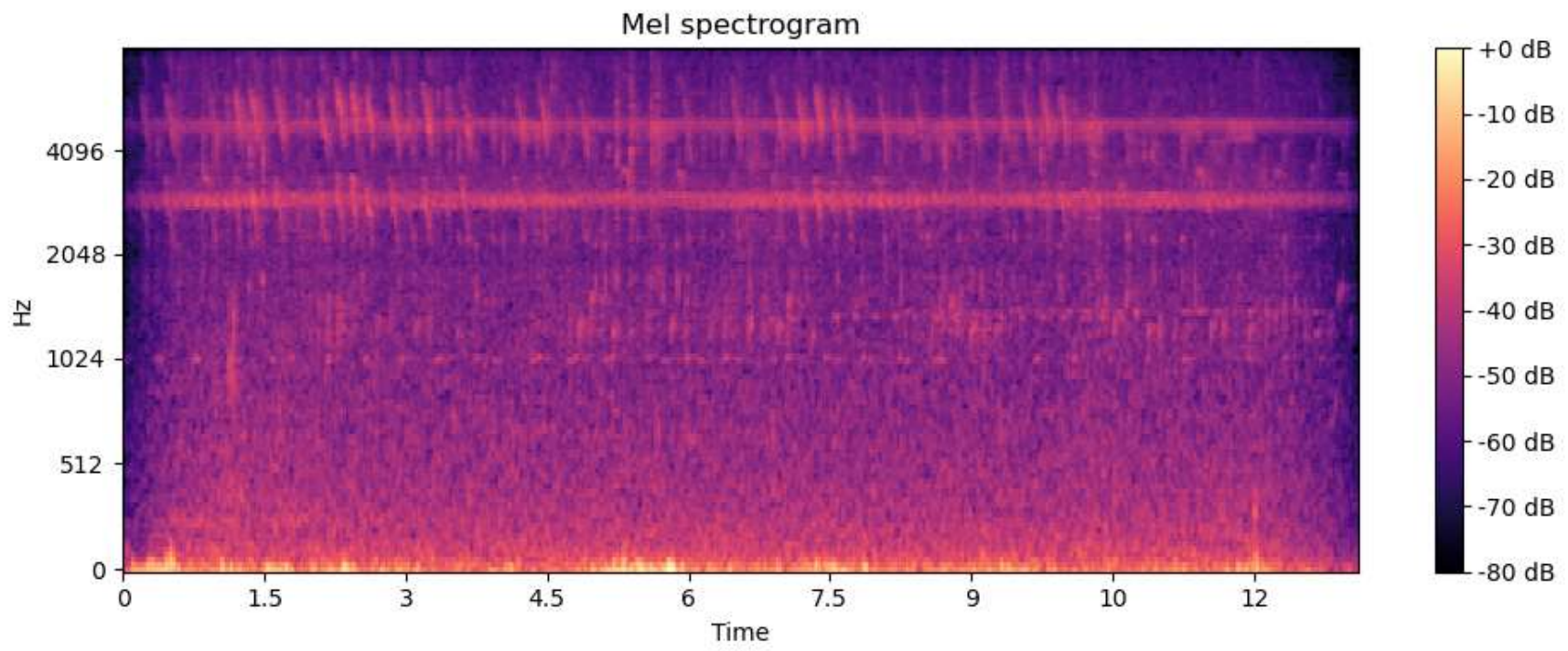
**Primary Label :** bawman1

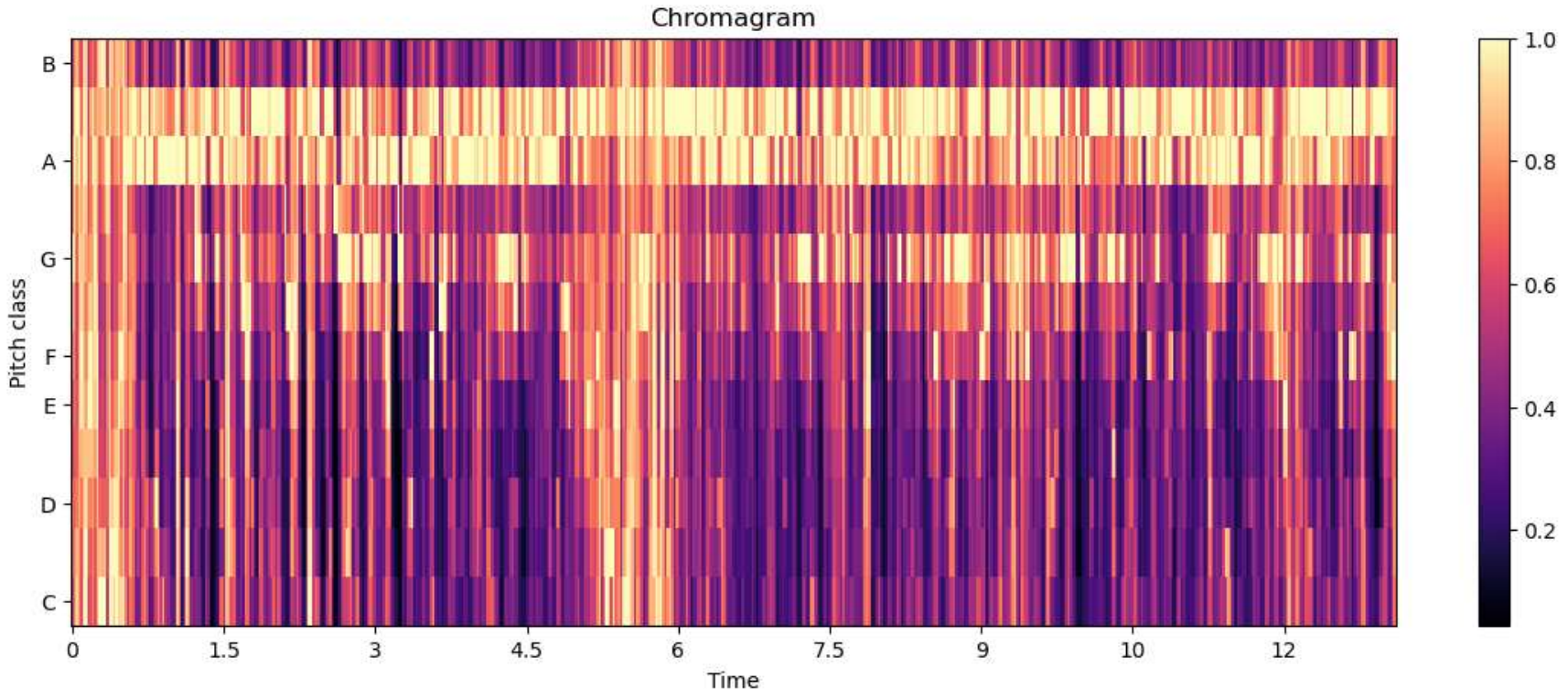
**Scientific Name :** Spermestes bicolor

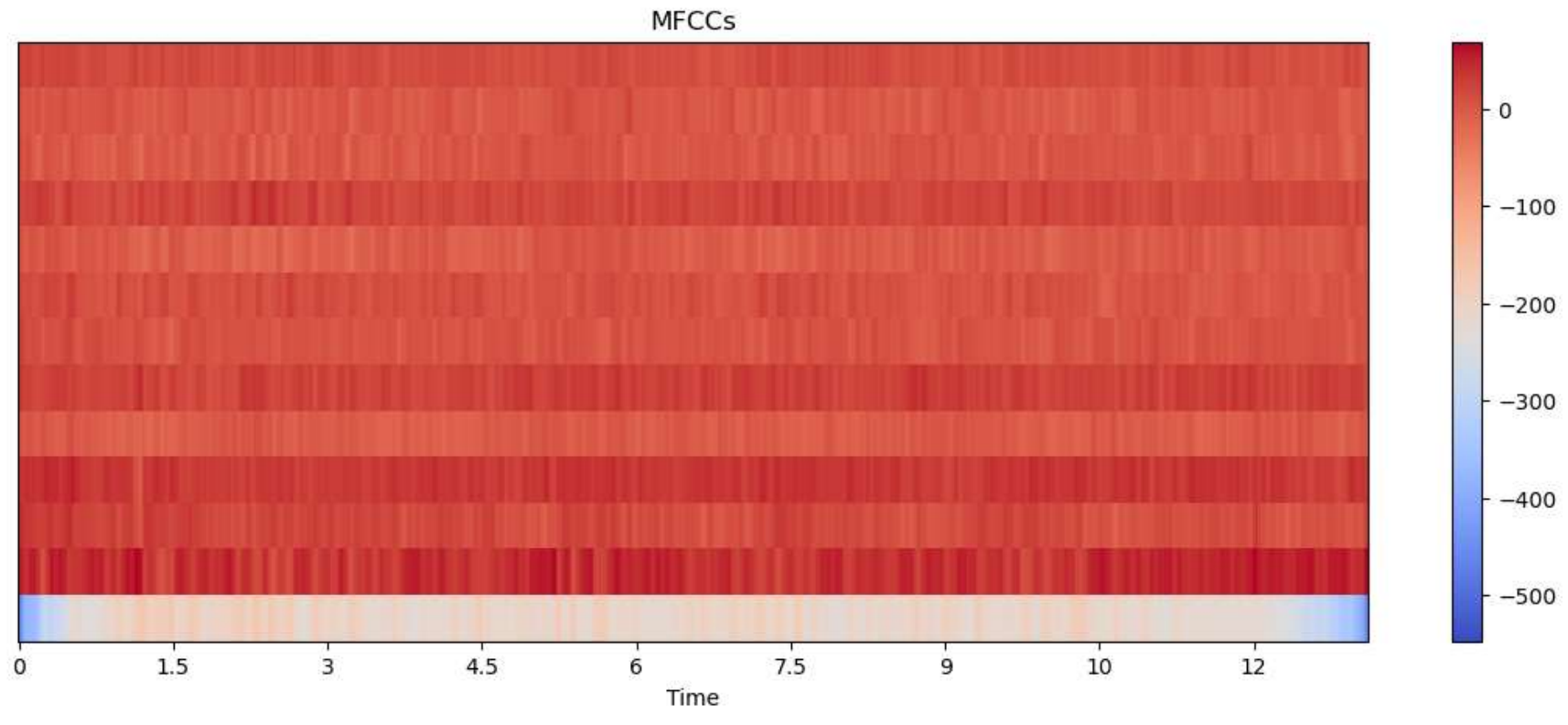
In [20]:
```
audio_eda("/input/birdclef-2023/train_audio/bawman1/XC115075.ogg")
```




Spectrogram (dB)
8192
4096
2048
1024
512
256
128
64
0
Hz
0
1.5
3
4.5
6
7.5
9
10
12
Time
+40 dB
+30 dB
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB


Mel spectrogram
4096
2048
1024
512
0
Hz
0
1.5
3
4.5
6
7.5
9
10
12
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Chromagram
B
A
G
F
E
D
C
Pitch class
0
1.5
3
4.5
6
7.5
9
10
12
Time
1.0
0.8
0.6
0.4
0.2


MFCCs
0
1.5
3
4.5
6
7.5
9
10
12
Time
0
−100
−200
−300
−400
−500

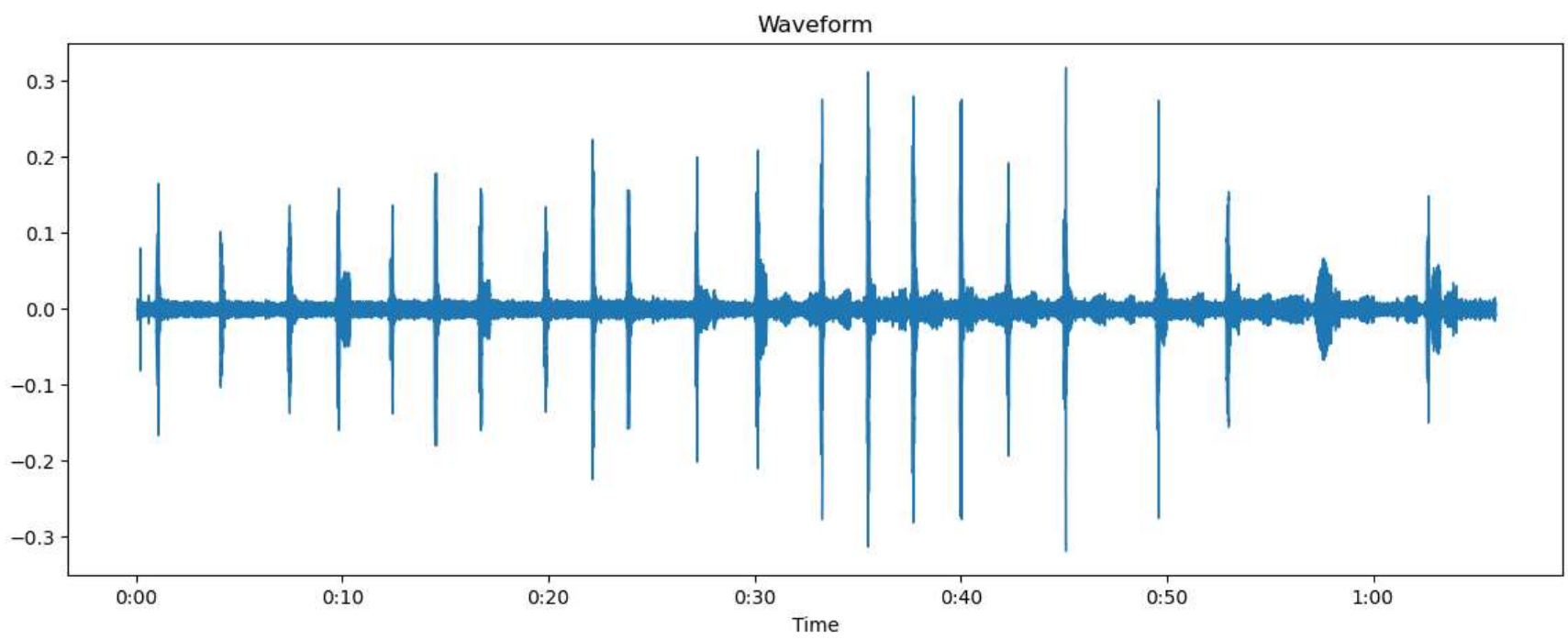
African Black-headed Oriole

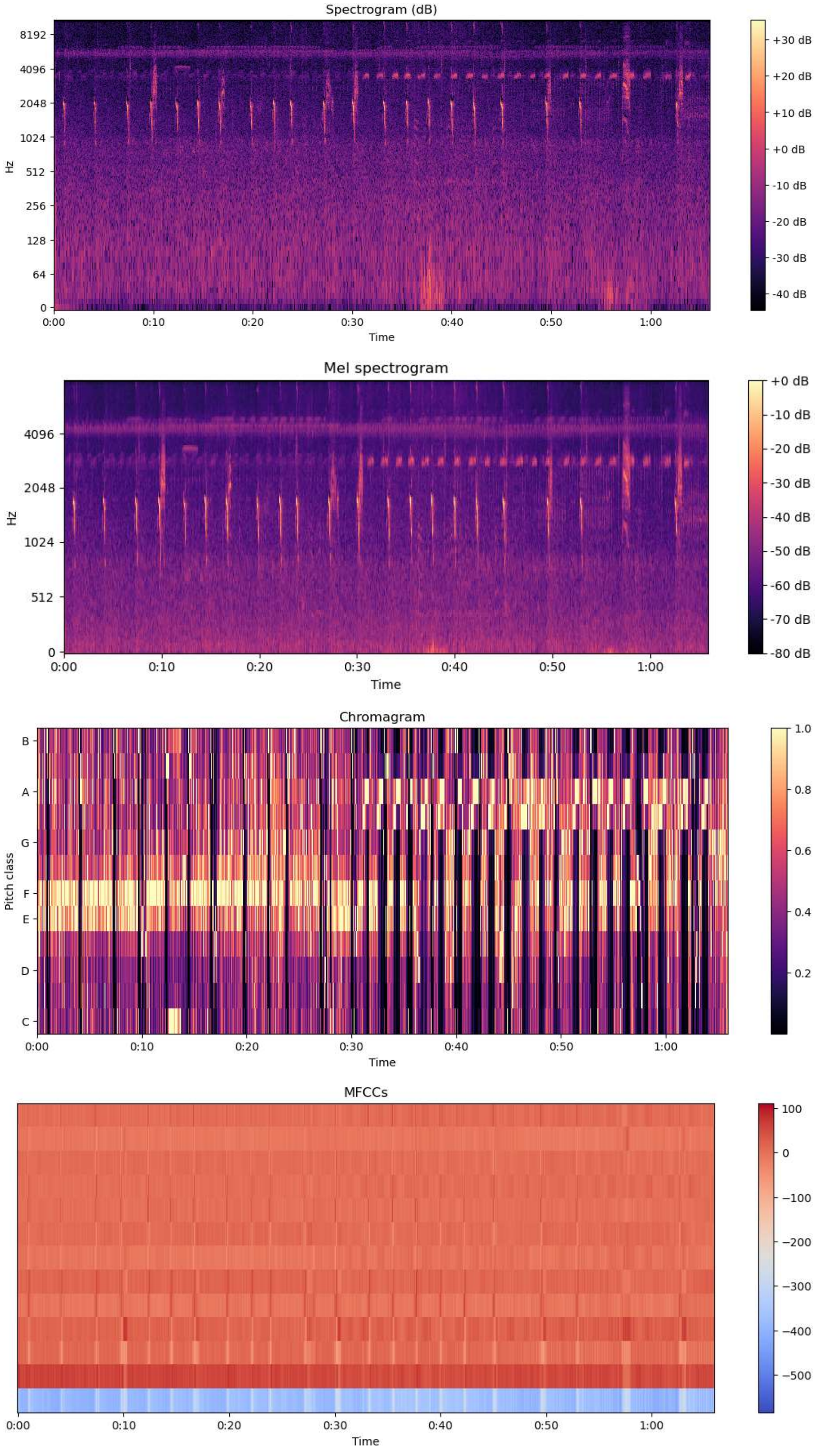
**Primary Label :** abhori1

**Scientific Name :** Oriolus larvatus

In [21]:
```
audio_eda("/input/birdclef-2023/train_audio/abhori1/XC120251.ogg")
```

0:00 / 1:05




Spectrogram (dB)
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:10
0:20
0:30
0:40
0:50
1:00
Time
+30 dB
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
Mel spectrogram
4096
2048
1024
512
0
Hz
0:00
0:10
0:20
0:30
0:40
0:50
1:00
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB
Chromagram
B
A
G
F
E
D
C
Pitch class
0:00
0:10
0:20
0:30
0:40
0:50
1:00
Time
1.0
0.8
0.6
0.4
0.2
MFCCs
0:00
0:10
0:20
0:30
0:40
0:50
1:00
Time
100
0
−100
−200
−300
−400
−500

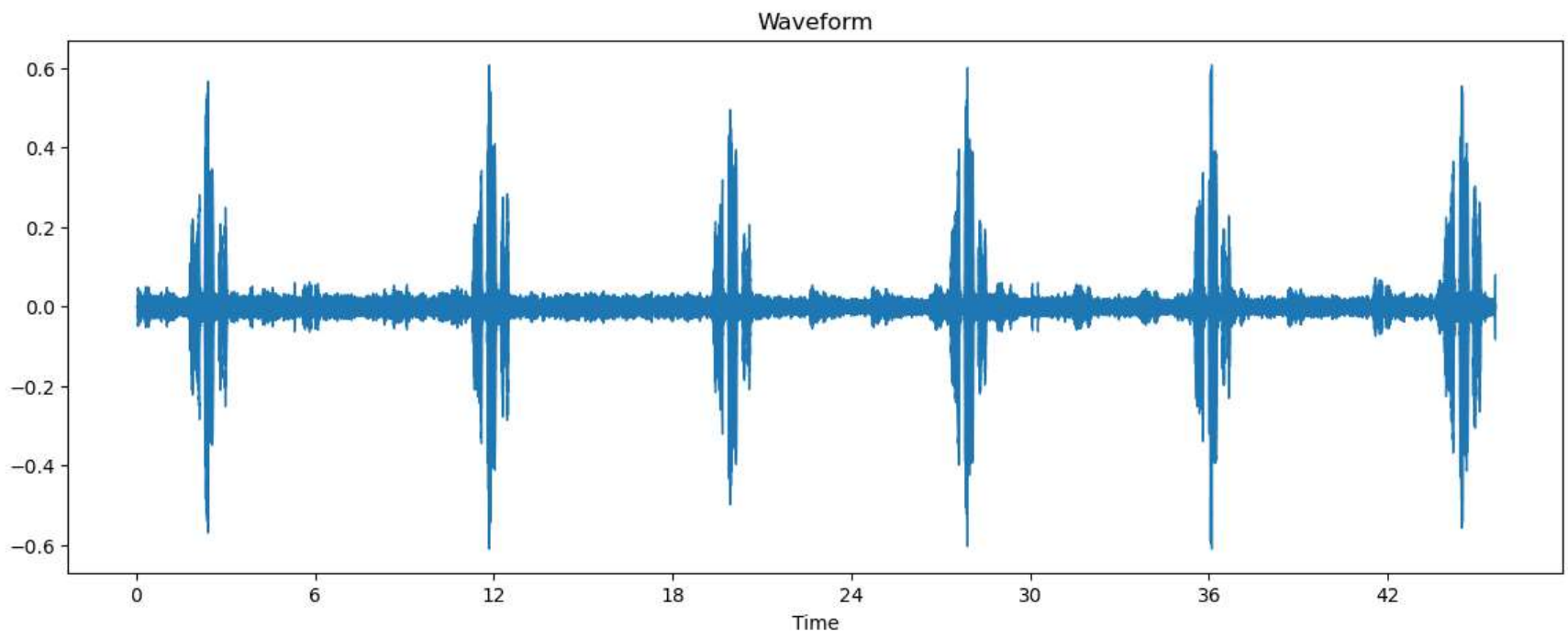
African Bare-eyed Thrush

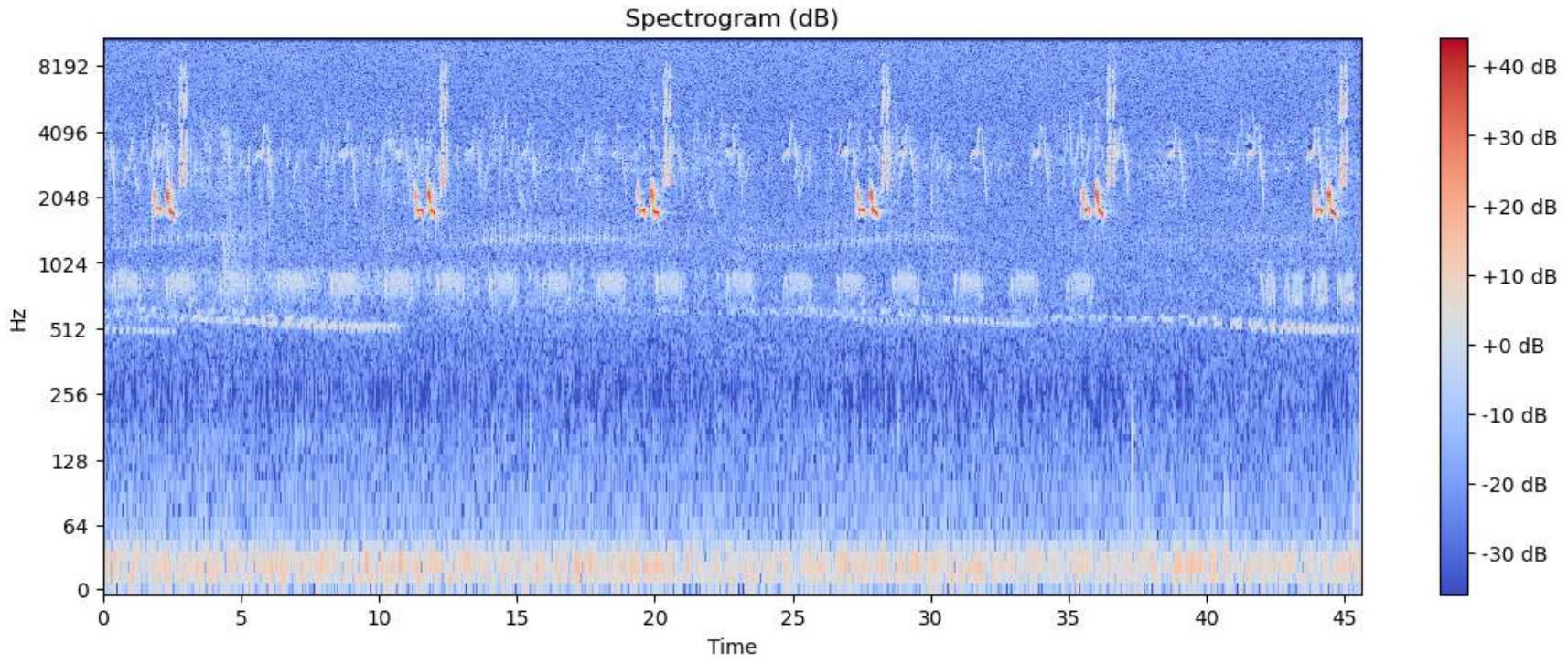
**Primary Label :** abethr1

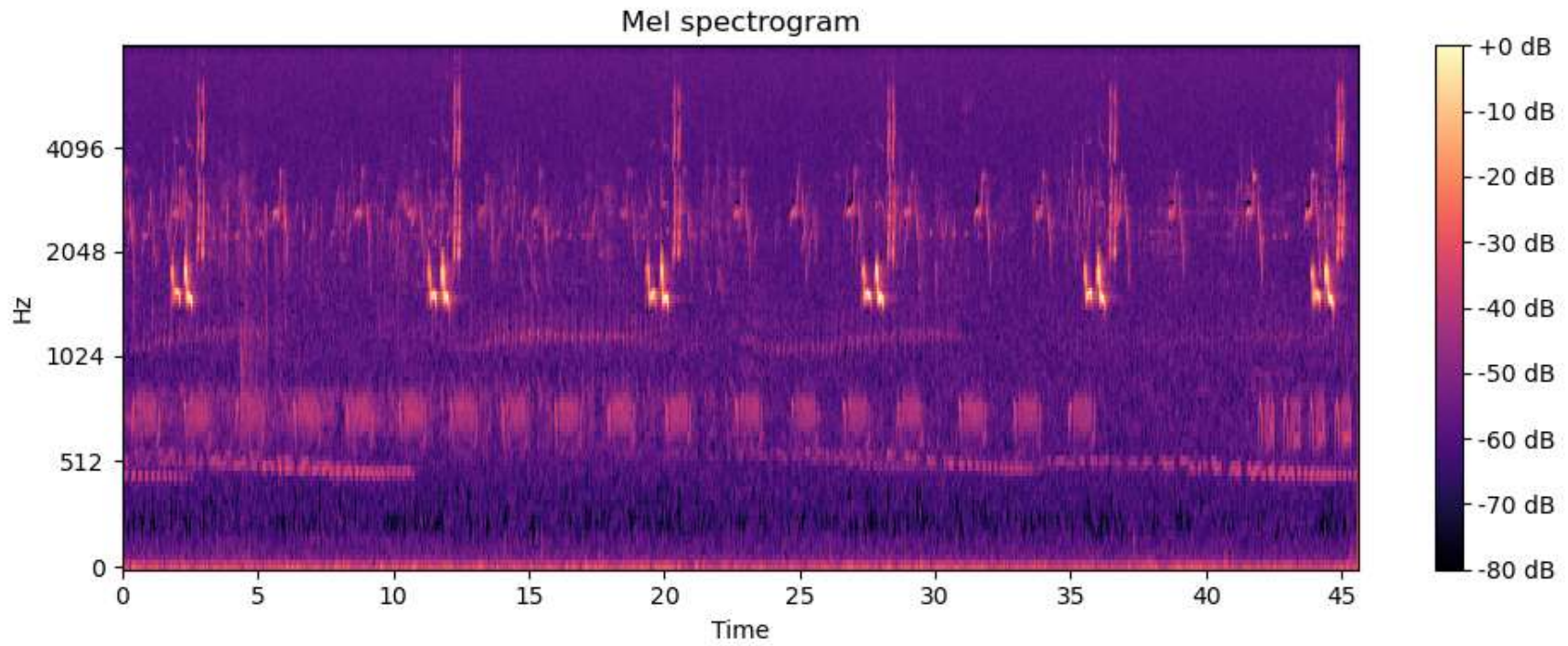
**Scientific Name :** Turdus tephronotus

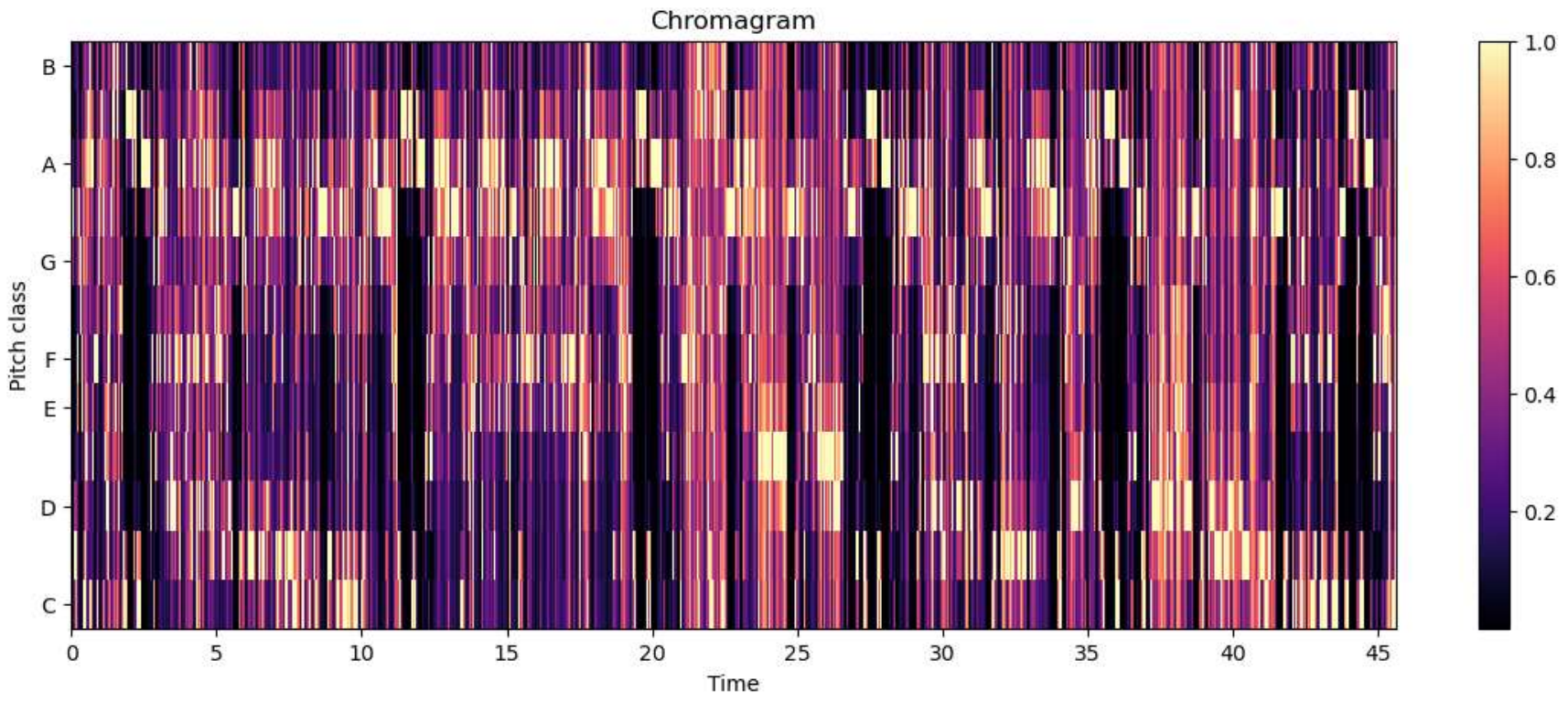
In [22]:
```
audio_eda("/input/birdclef-2023/train_audio/abethr1/XC128013.ogg")
```

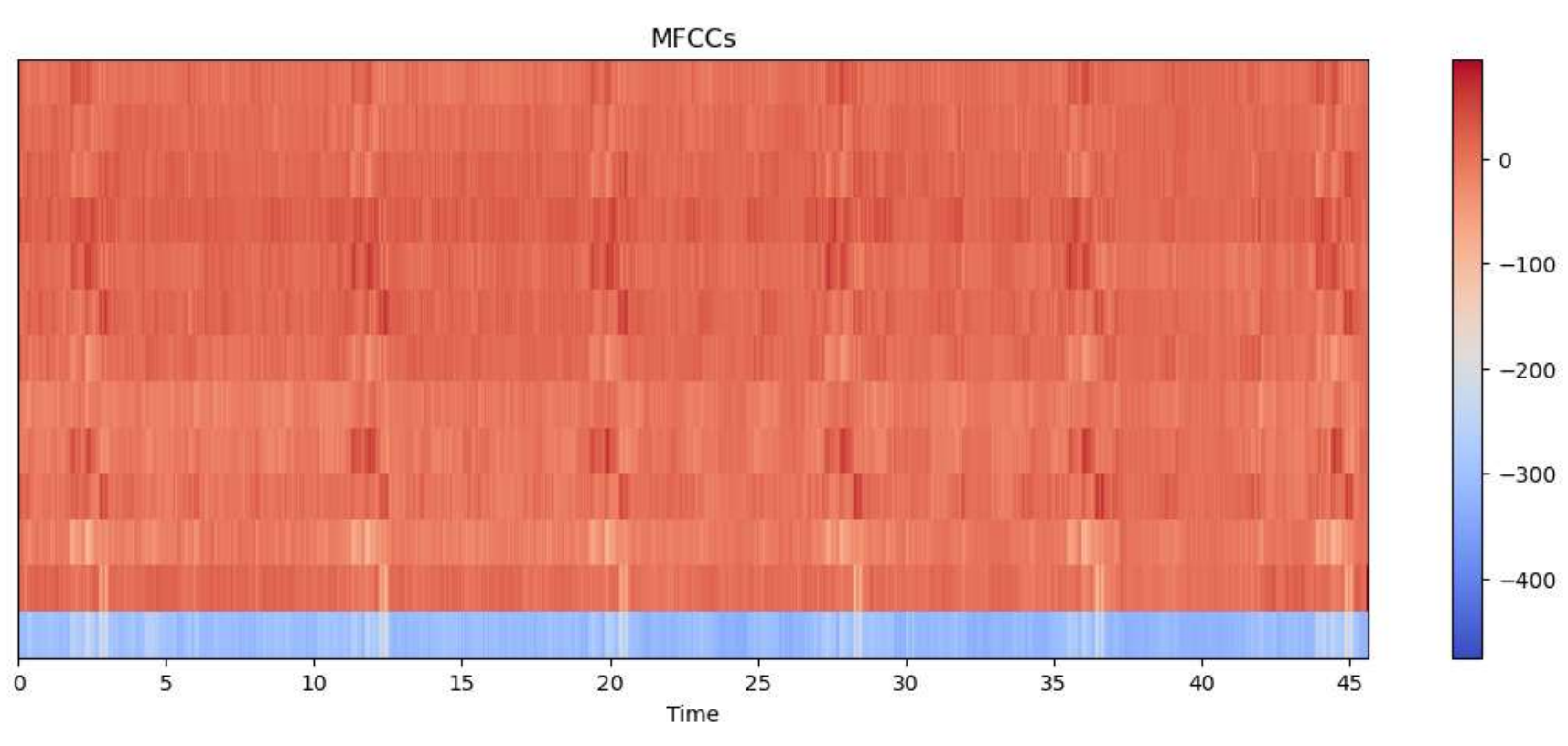
0:00 / 0:45




Spectrogram (dB)
Hz
8192
4096
2048
1024
512
256
128
64
0
0
5
10
15
20
25
30
35
40
45
Time
+40 dB
+30 dB
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB


Mel spectrogram
Hz
4096
2048
1024
512
0
0
5
10
15
20
25
30
35
40
45
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Chromagram
Pitch class
B
A
G
F
E
D
C
0
5
10
15
20
25
30
35
40
45
Time
1.0
0.8
0.6
0.4
0.2


MFCCs
0
5
10
15
20
25
30
35
40
45
Time
0
−100
−200
−300
−400

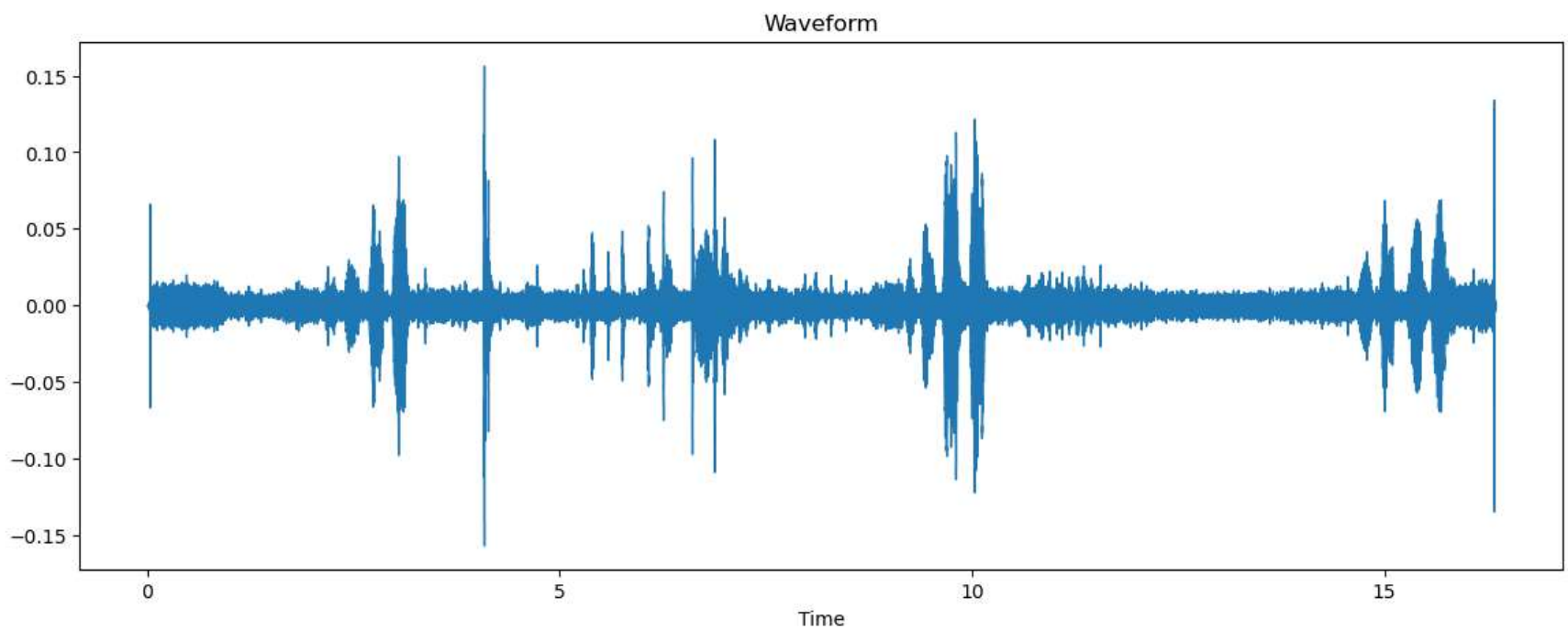
African Gray Flycatcher

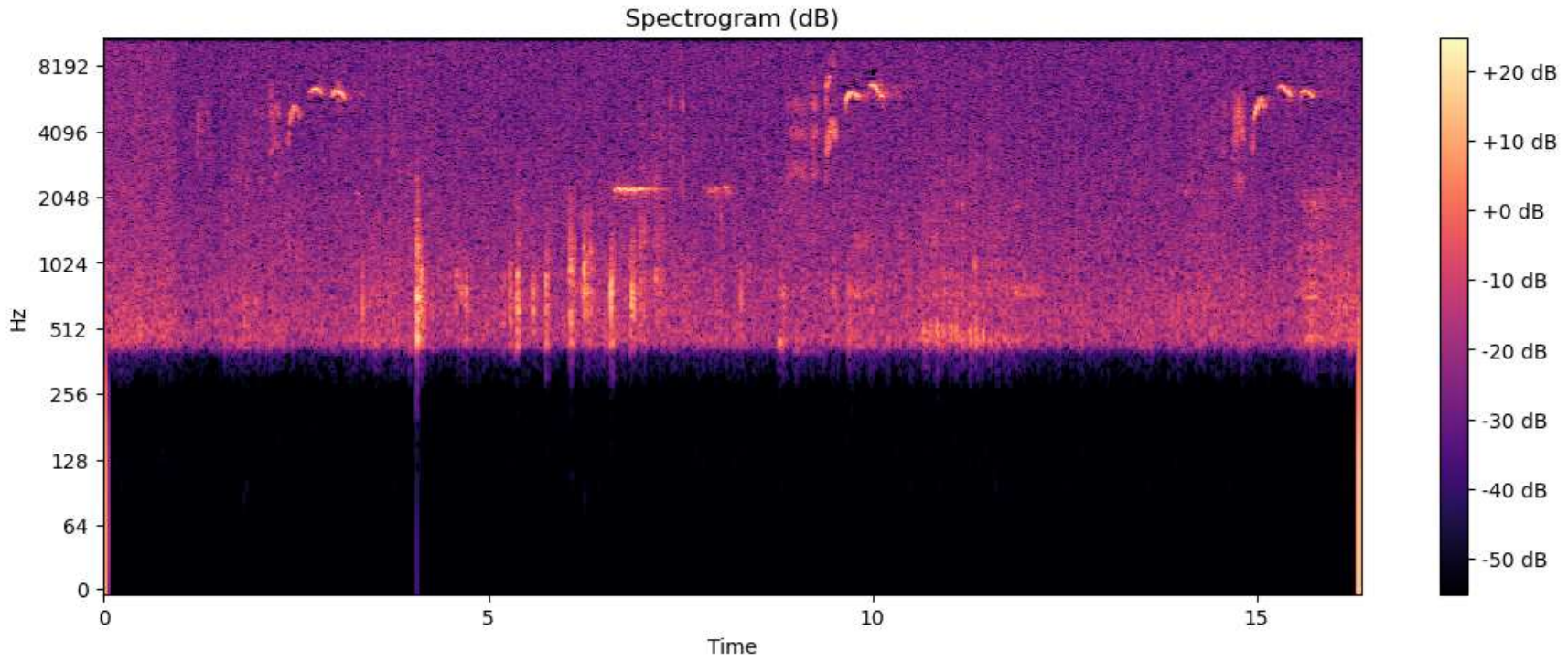
**Primary Label :** afgfly1

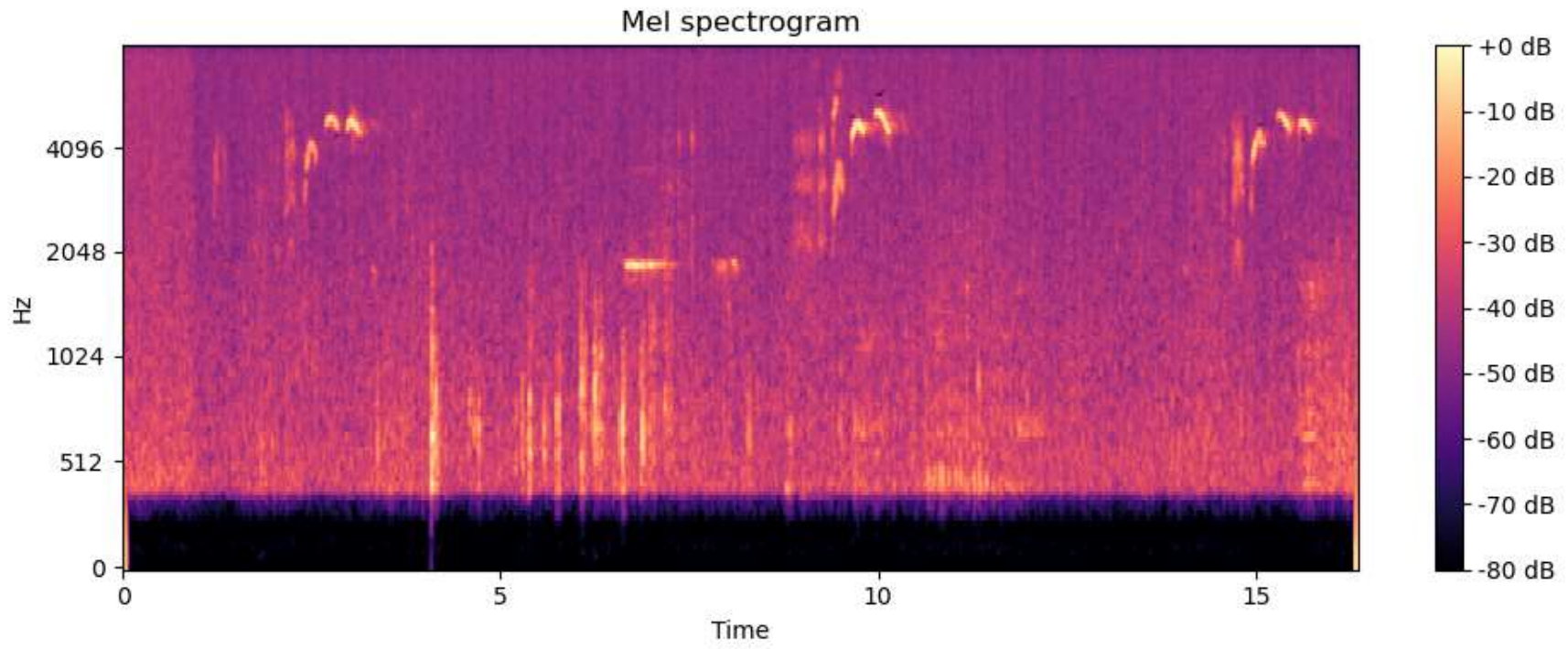
**Scientific Name :** Bradornis microrhynchus

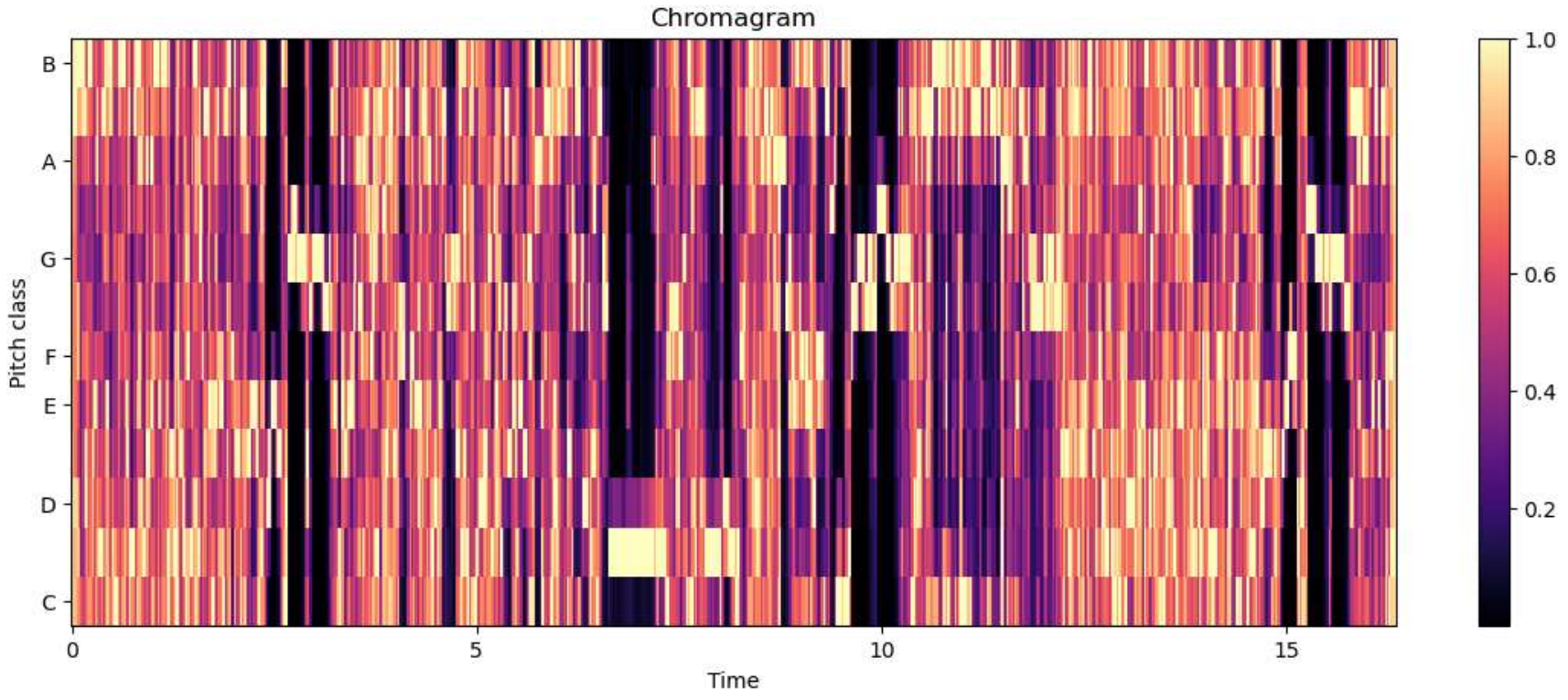
In [23]:
```
audio_eda("/input/birdclef-2023/train_audio/afgfly1/XC134487.ogg")
```

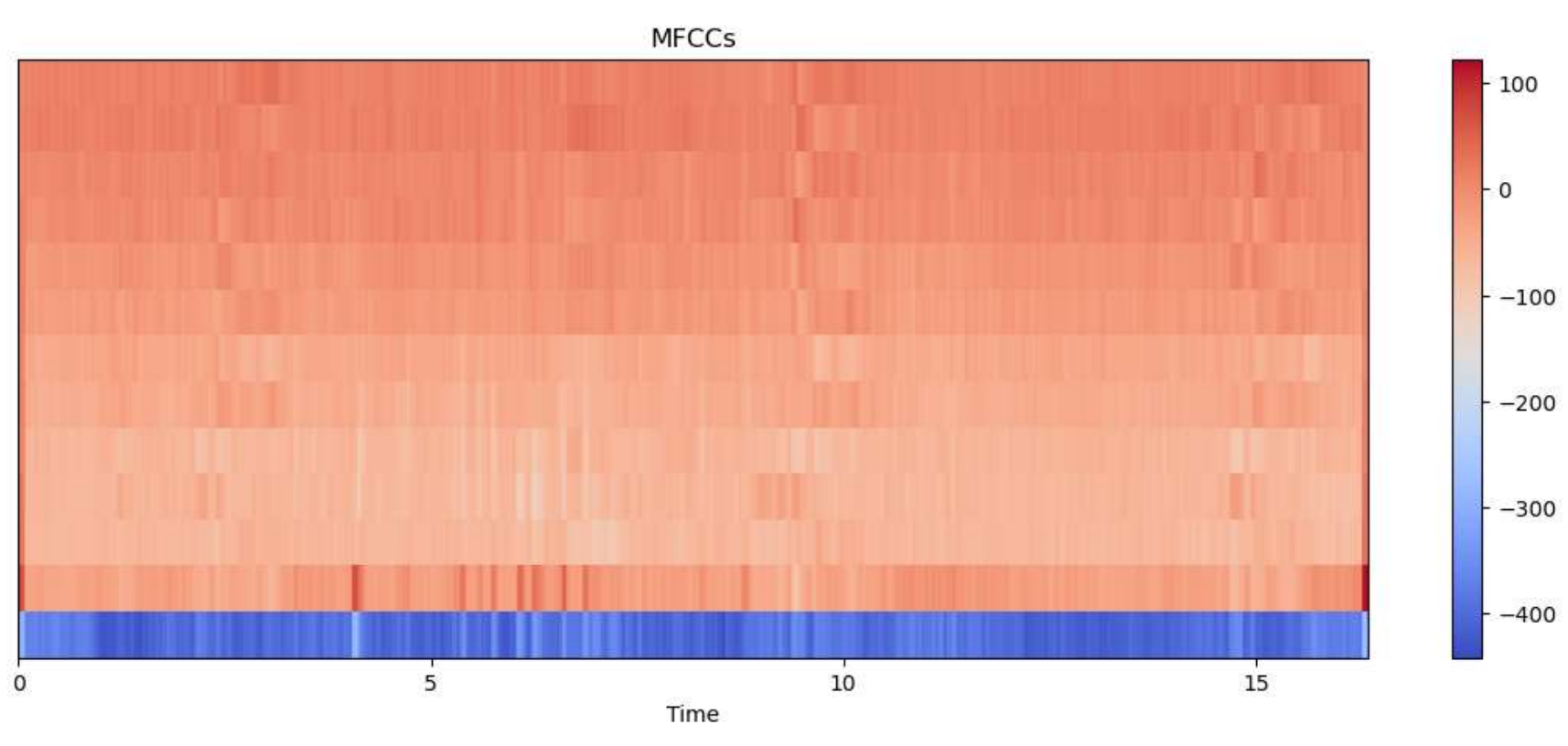
0:00 / 0:16




Spectrogram (dB)
8192
4096
2048
1024
512
256
128
64
0
Hz
0
5
10
15
Time
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB


Mel spectrogram
4096
2048
1024
512
0
Hz
0
5
10
15
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Chromagram
B
A
G
F
E
D
C
Pitch class
0
5
10
15
Time
1.0
0.8
0.6
0.4
0.2


MFCCs
0
5
10
15
Time
100
0
−100
−200
−300
−400

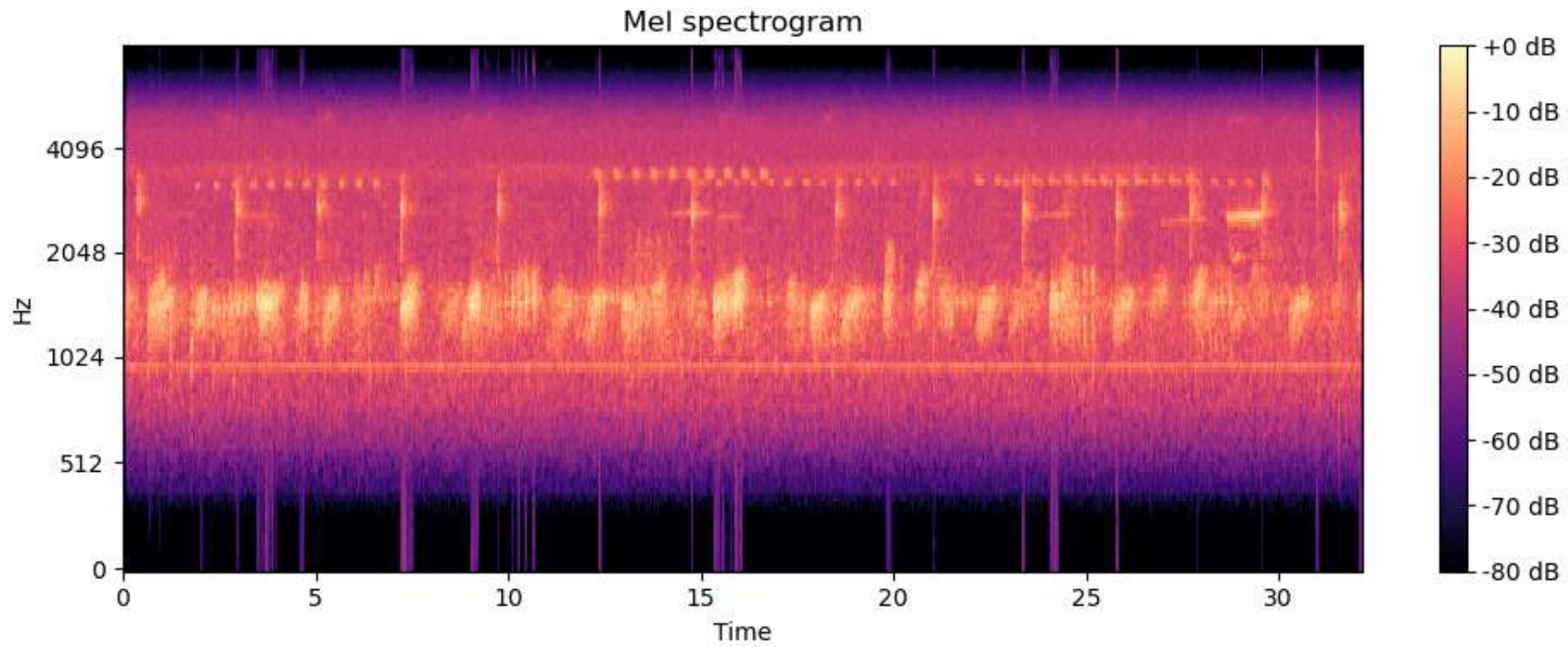
African Goshawk

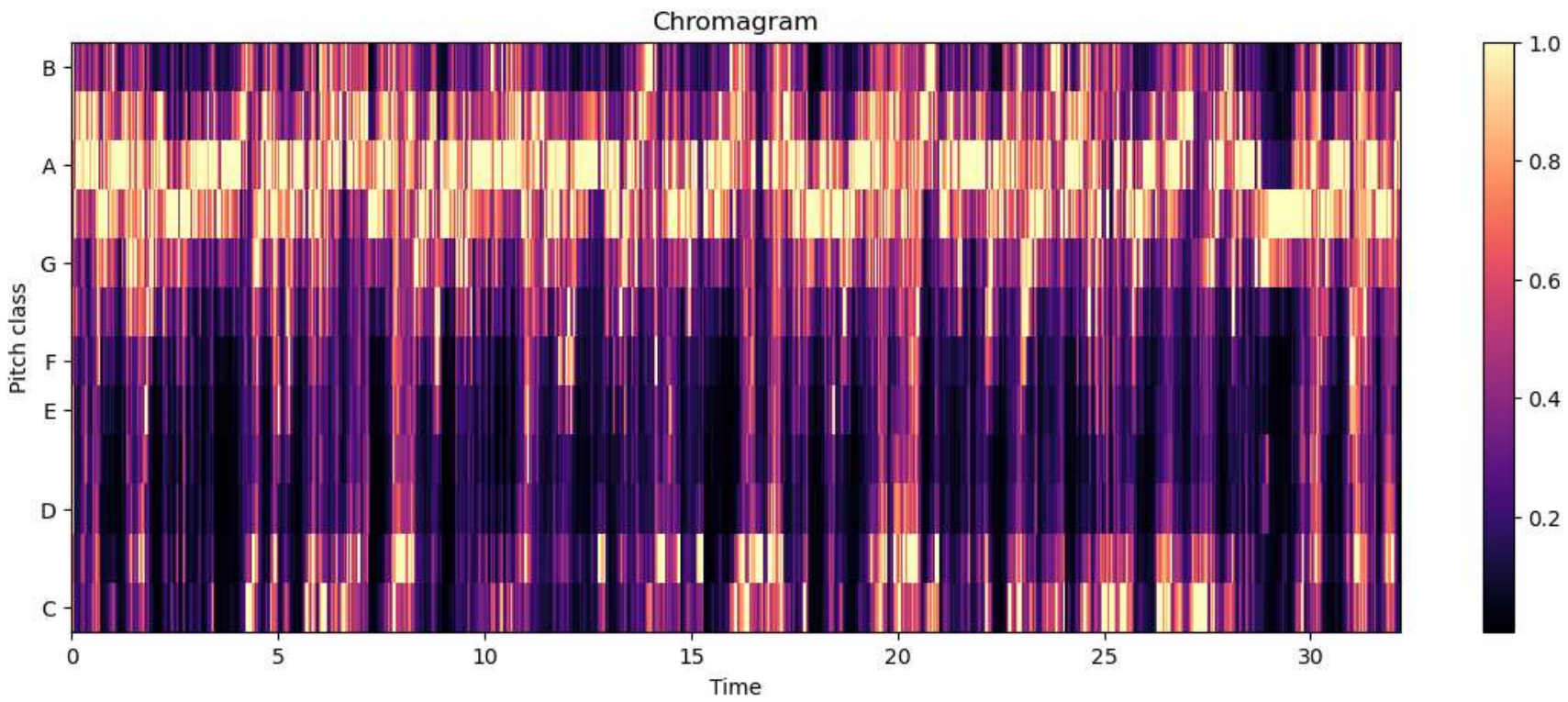

**Primary Label :** afrgos1

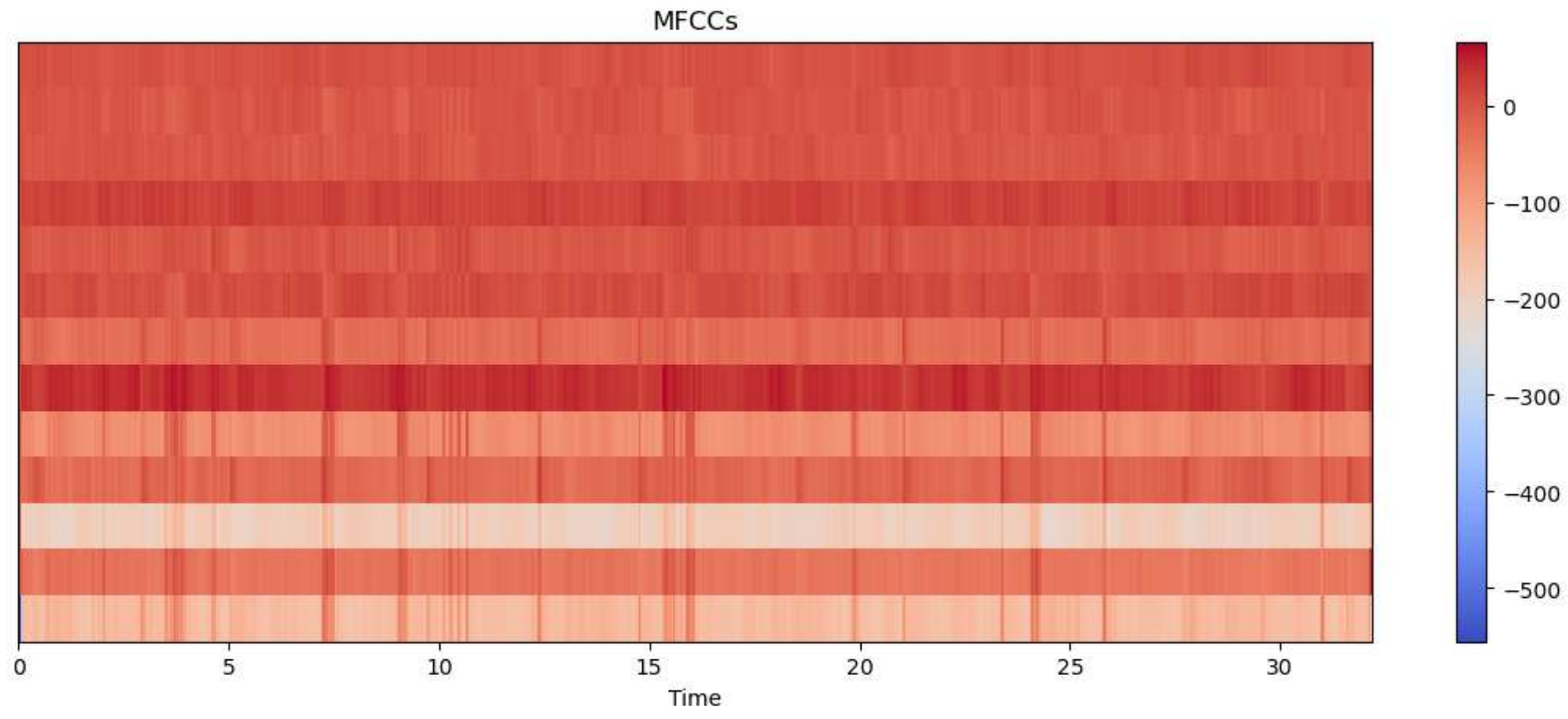
**Scientific Name :** Accipiter tachiro

In [24]:
```
audio_eda("/input/birdclef-2023/train_audio/afrgos1/XC115984.ogg")
```

0:00 / 0:32


Mel spectrogram
Hz
4096
2048
1024
512
0
0
5
10
15
20
25
30
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Chromagram
Pitch class
B
A
G
F
E
D
C
0
5
10
15
20
25
30
Time
1.0
0.8
0.6
0.4
0.2


MFCCs
0
5
10
15
20
25
30
Time
0
−100
−200
−300
−400
−500