In [23]:
```
import os
print(os.getcwd())
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.io import wavfile
```

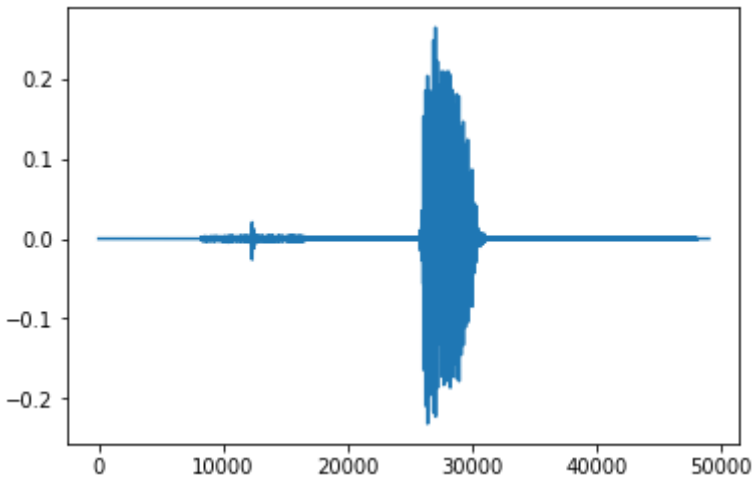
```
/Users/heosangbeom/Desktop/data_final/Audio
```

In [24]:
```
os.chdir("/Users/heosangbeom/Desktop/data_final/Audio")
```

In [25]:
```
import librosa
audio_path = ("/Users/heosangbeom/Desktop/data_final/Audio/허상범_a.mp4")
```

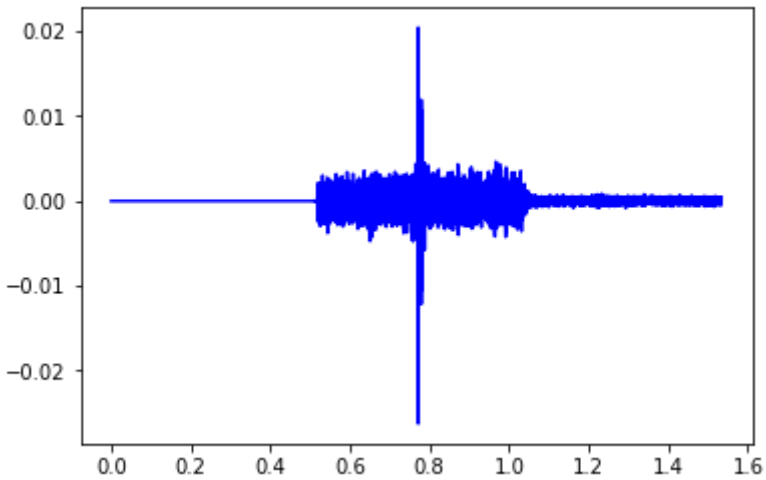
In [157]:
```
y, sr = librosa.load(audio_path, sr=16000) # y=frequency, sr=sampling rate
time = np.linspace(0, len(y)/sr, len(y))
ax1.set_ylabel("Amplitude")
ax1.set_xlabel("Time [s]")
fig = plt.plot(y) # plot
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/librosa/core/audio.py:165: UserWar
ning: PySoundFile failed. Trying audioread instead.
  warnings.warn("PySoundFile failed. Trying audioread instead.")
```



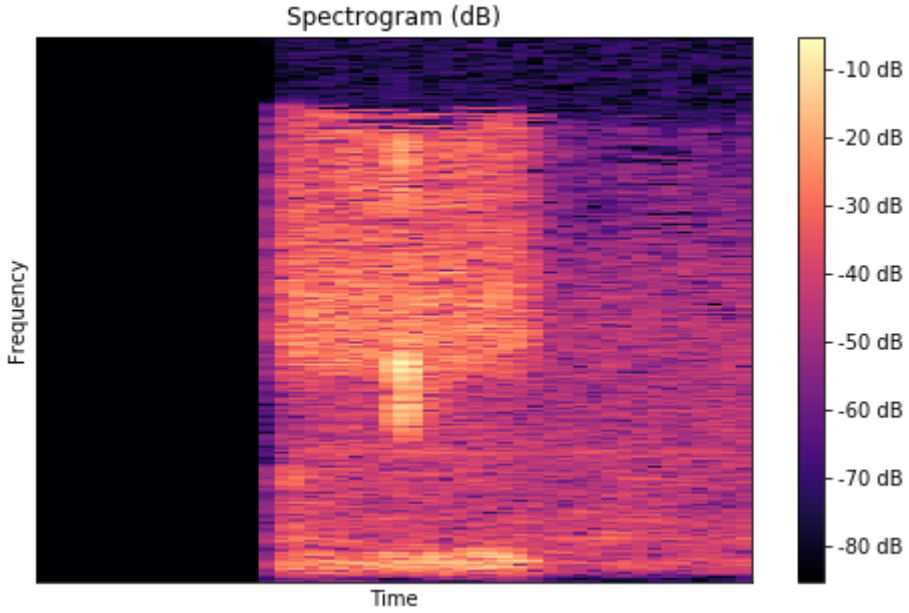
In [156]:
```
# 명확한 이진 분류를 위해 음성의 절반만을 사용하여 좀 더 깨끗한 음성 데이터 만들기
half = len(y)/2
y2 = y[:round(half)]
time2 = np.linspace(0, len(y2)/sr, len(y2))
fig2, ax2 = plt.subplots()
ax2.plot(time2, y2, color = 'b', label='speech waveform')
ax1.set_ylabel("Amplitude") # y 축
ax1.set_xlabel("Time [s]") # x 축
```

Out[156]:
```
Text(0.5, 17.200000000000003, 'Time [s]')
```



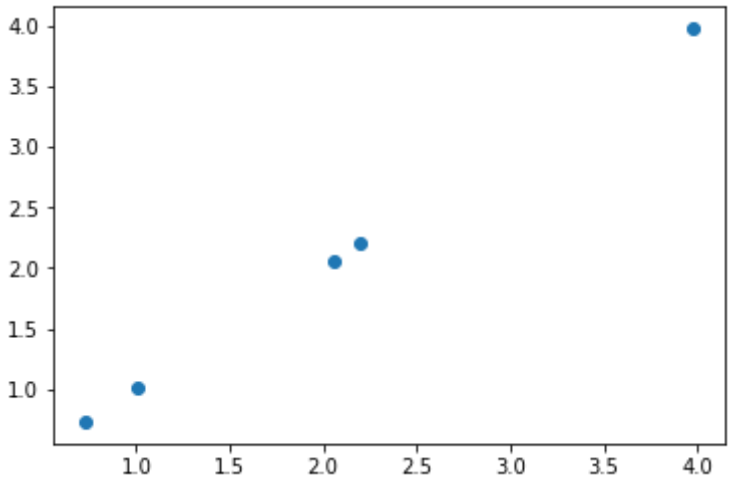
In [164]:
```
# 원본 노이즈 감소 및 주파수에 의존하는 음성 데이터를 인식하기 위해 Short Term Fourier Tra
# STFT된 음성 데이터를 spectrogram으로 표현하여 raw data와 어떻게 다른지 시각화

hop_length = 512  # 전체 frame 수
n_fft = 2048  # frame 하나당 sample 수
hop_length_duration = float(hop_length)/sr
n_fft_duration = float(n_fft)/sr

# STFT 변환
stft = librosa.stft(y2, n_fft=n_fft, hop_length=hop_length)
magnitude = np.abs(stft)

# display spectrogram
plt.figure(figsize=(8,5))
librosa.display.specshow(log_spectrogram, sr=sr, hop_length=hop_length)
plt.xlabel("Time")
plt.ylabel("Frequency")
plt.colorbar(format="%+2.0f dB")
plt.title("Spectrogram (dB)")
```

Out[164]:
```
Text(0.5, 1.0, 'Spectrogram (dB)')
```

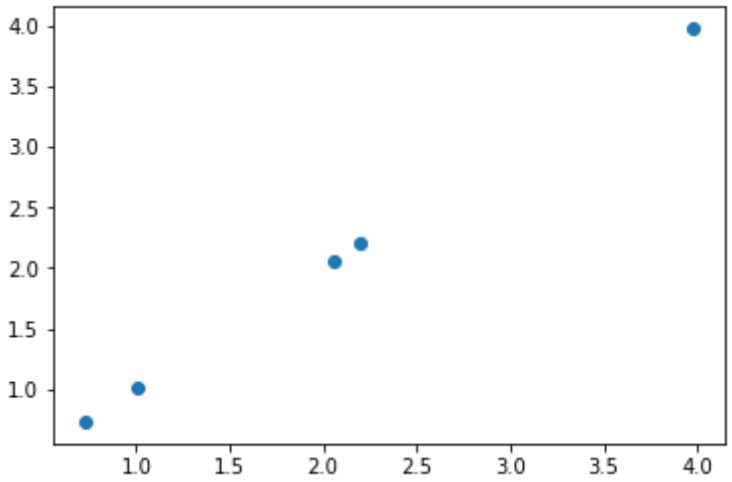


In [226]:
```
# 로지스틱 회귀 모형에 데이터 학습시키기 위한 음성데이터 이진 분류 모델 만들기
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# 데이터셋 생성
sample_audio = np.random.uniform(-1,1,16000) # 샘플 오디오 데이터 난수 생성, 주파수
scaled = np.int16(sample_audio/np.max(np.abs(sample_audio)) * 16000)

x_train = np.random.normal(1, 2, 5) # sample audio data
y_train = y2 # 전처리 데이터 (label)
x_test = scaled
y_test = y2
```

In [230]:
```
plot_x = x_train[:]
plot_y = x_train[:]
plot_color = y_train.reshape(24544,)
plt.scatter(plot_x, plot_y)
plt.show()
```



In [228]:
```
sample_audio2 = np.random.uniform(-1,1,44100)
scaled2 = np.int16(sample_audio2/np.max(np.abs(sample_audio2)) * 32767)
sample_audio2 = np.random.uniform(-1,1,44100) # 샘플 오디오 데이터, 주파수 44100
scaled = np.int16(sample_audio2/np.max(np.abs(sample_audio2)) * 44100)
x_train = np.random.normal(1, 2, 5) # sample audio data
y_train = y2 # 전처리 데이터 (label)
x_test = scaled2
y_test = y2

plot_x = x_train[:]
plot_y = x_train[:]
plot_color = y_train.reshape(24544,)
plt.scatter(plot_x, plot_y)
plt.show()
```



In [ ]:
```
```