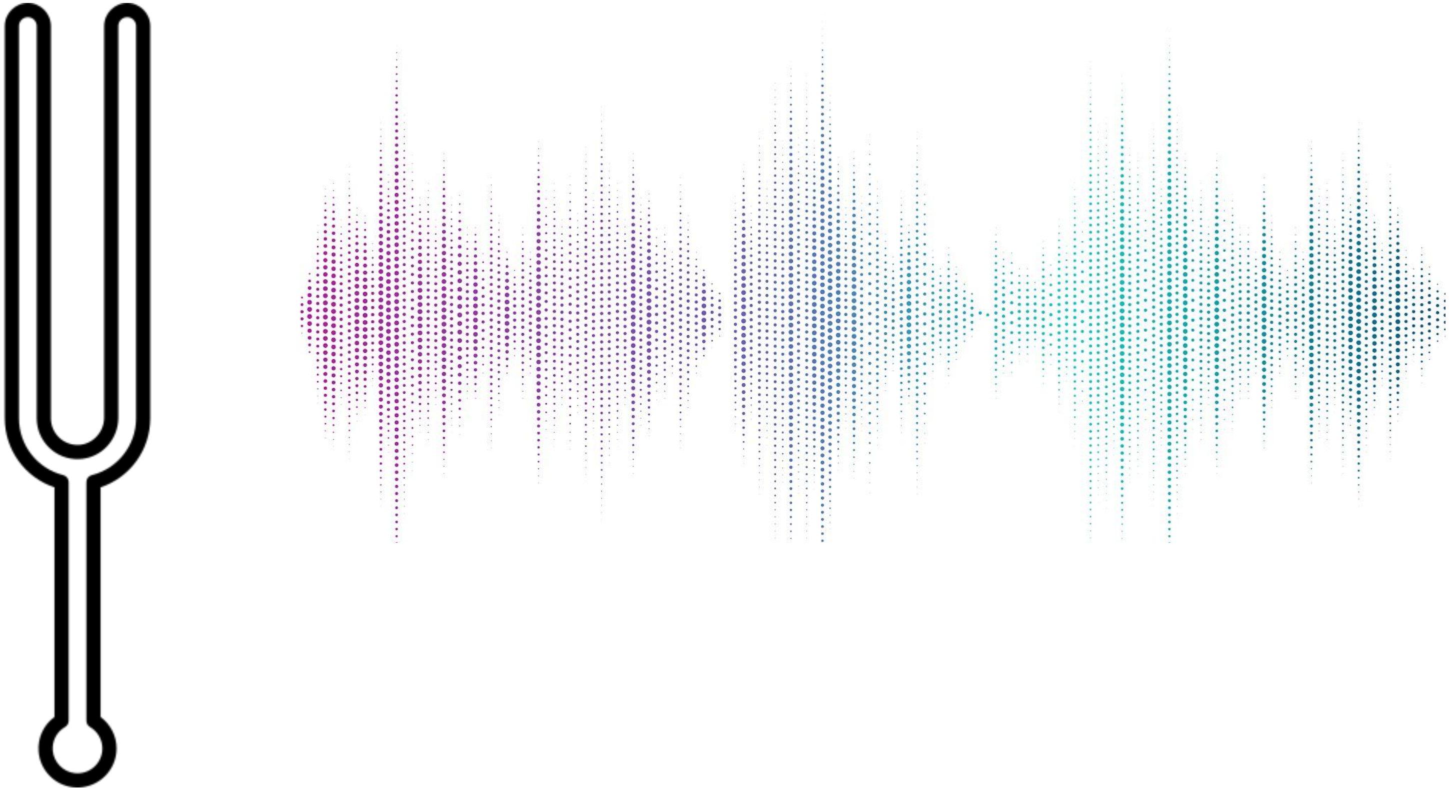
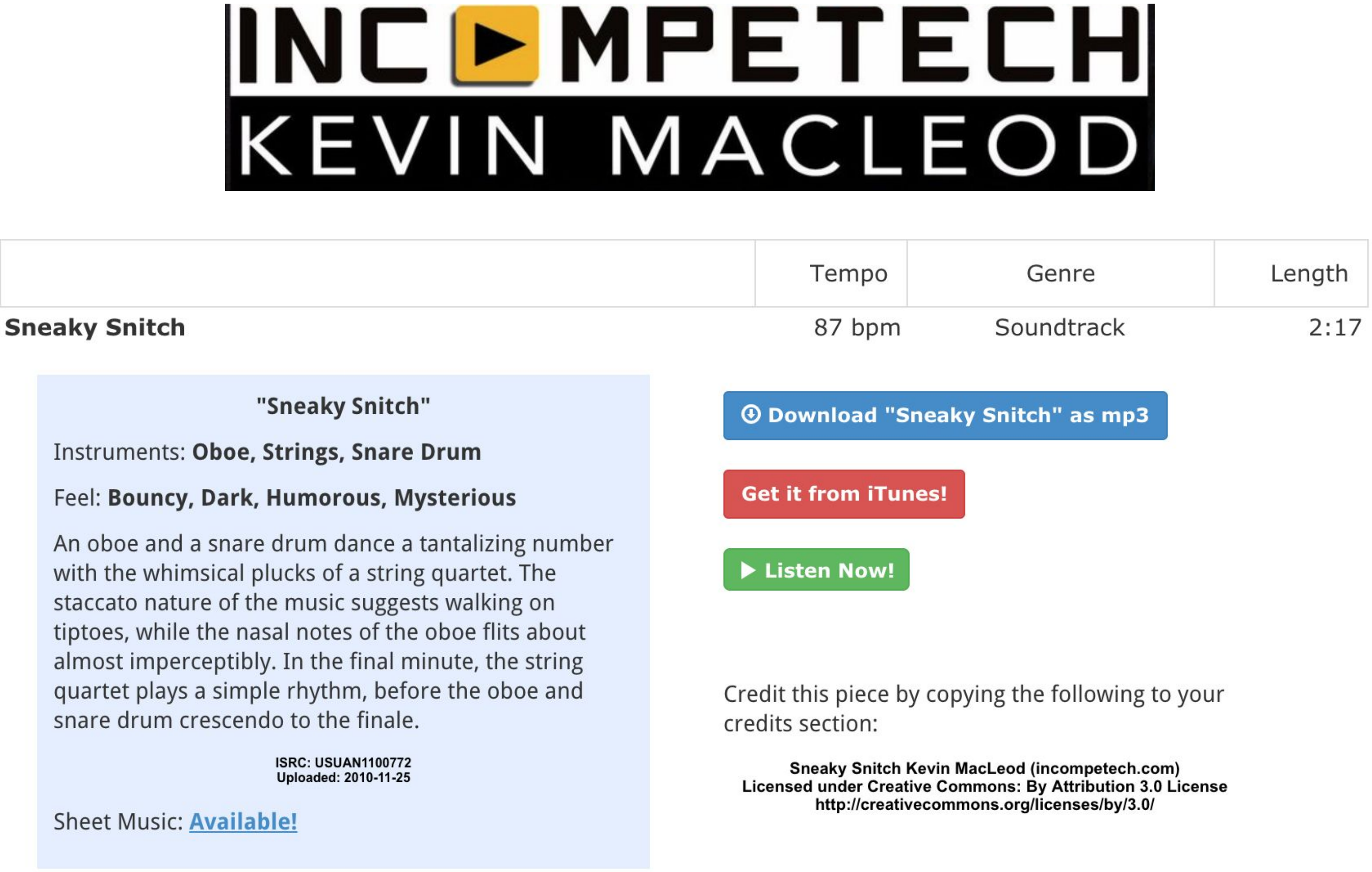
Audio Analysis

Getting and Cleaning Data



Icon made by FreePIk from www.flaticon.com, vector created by rawpixel.com on www.freepik.com



	Tempo	Genre	Length
Sneaky Snitch	87 bpm	Soundtrack	2:17

"Sneaky Snitch"

Instruments: **Oboe, Strings, Snare Drum**

Feel: **Bouncy, Dark, Humorous, Mysterious**

An oboe and a snare drum dance a tantalizing number with the whimsical plucks of a string quartet. The staccato nature of the music suggests walking on tiptoes, while the nasal notes of the oboe flits about almost imperceptibly. In the final minute, the string quartet plays a simple rhythm, before the oboe and snare drum crescendo to the finale.

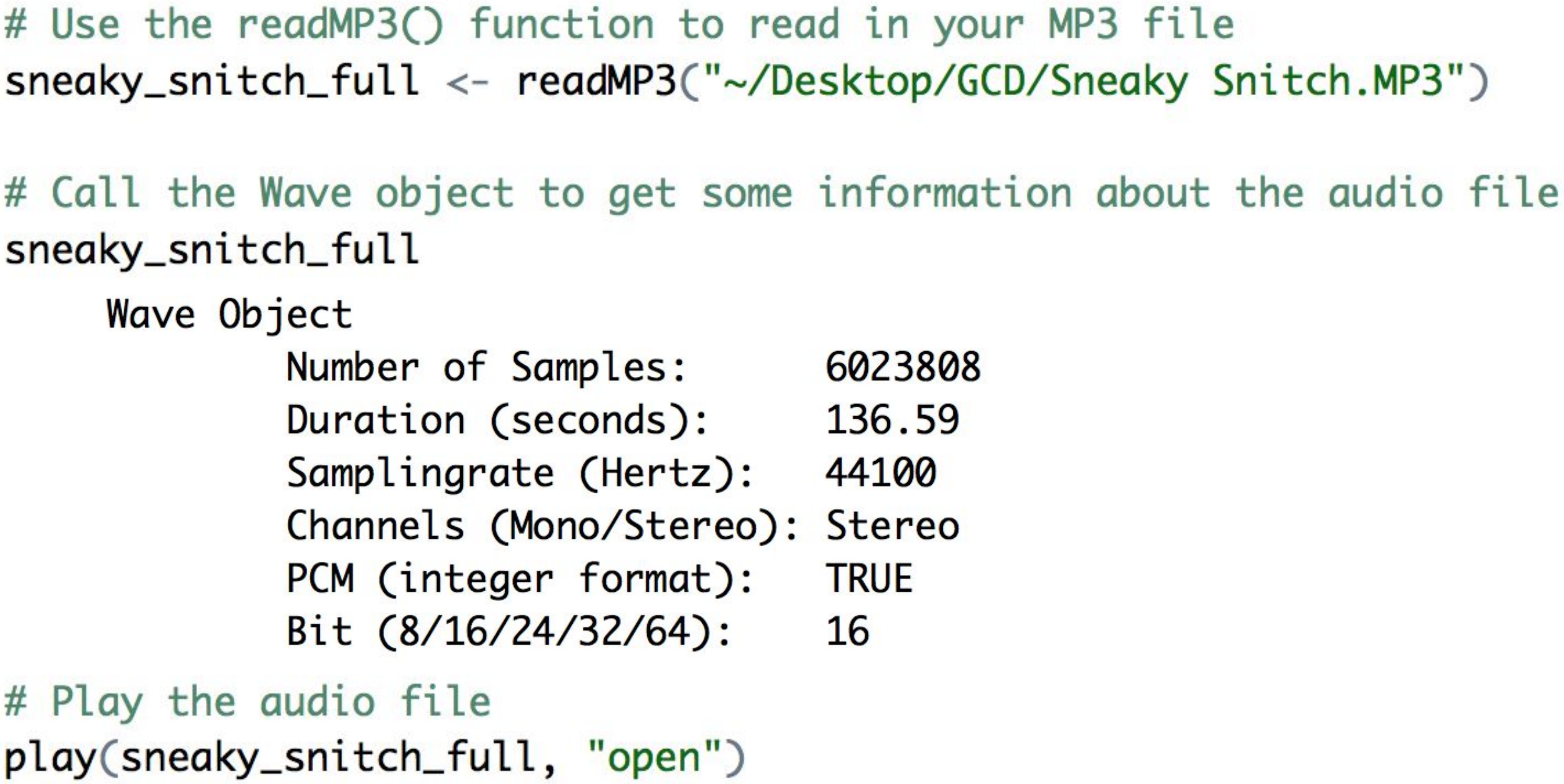
ISRC: USUAN1100772
Uploaded: 2010-11-25

Sheet Music: **Available!**

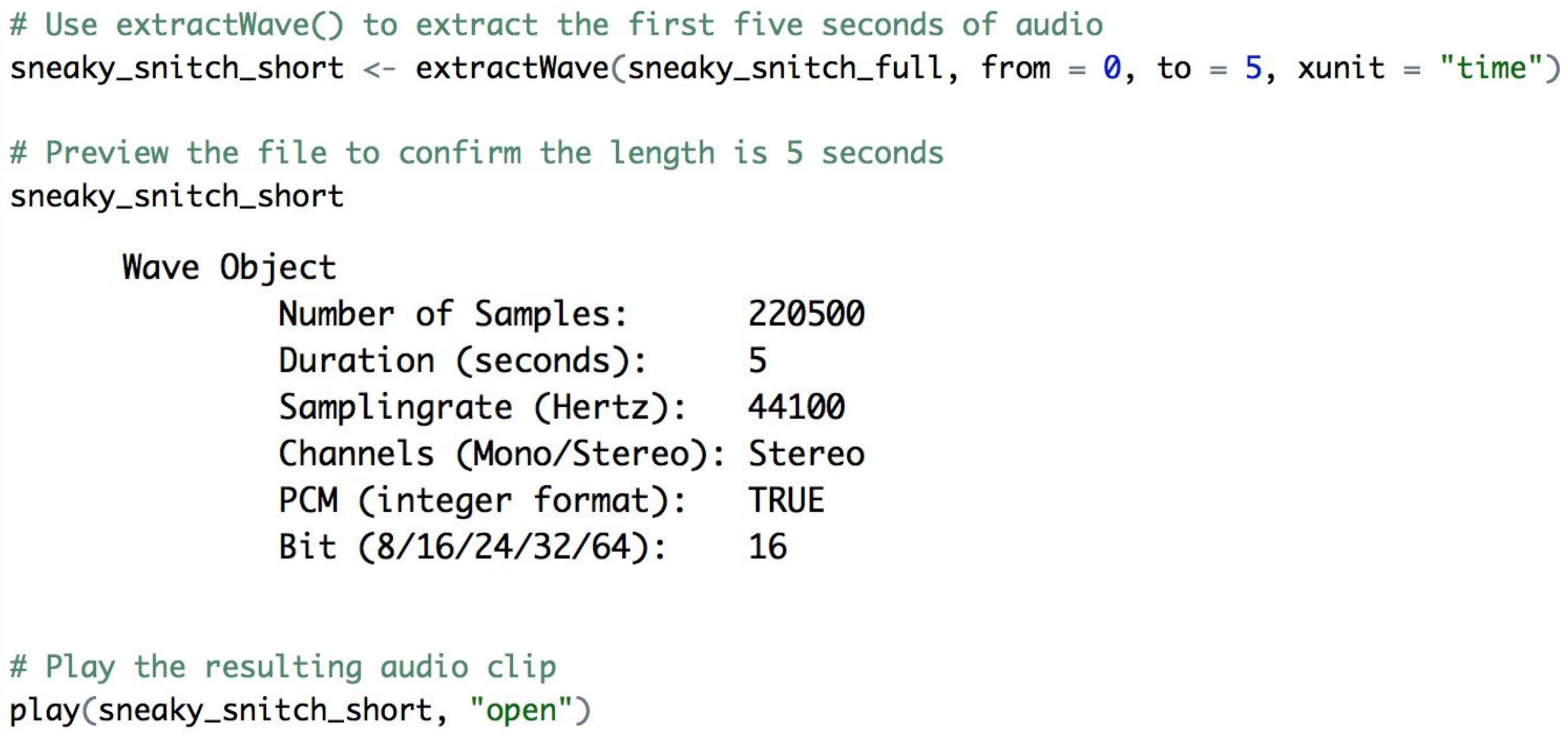
Credit this piece by copying the following to your credits section:

Sneaky Snitch Kevin MacLeod (incompetech.com)
Licensed under Creative Commons: By Attribution 3.0 License
http://creativecommons.org/licenses/by/3.0/

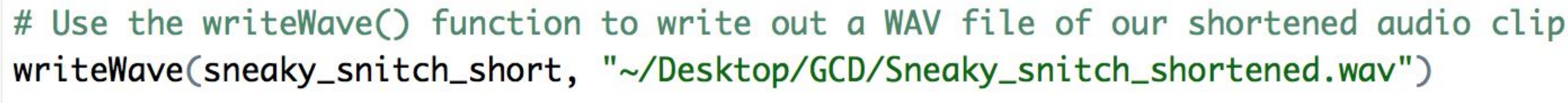
https://twitter.com/kmacleod; https://www.incompetech.com/music/royalty-free/index.html?isrc=USUAN1100772



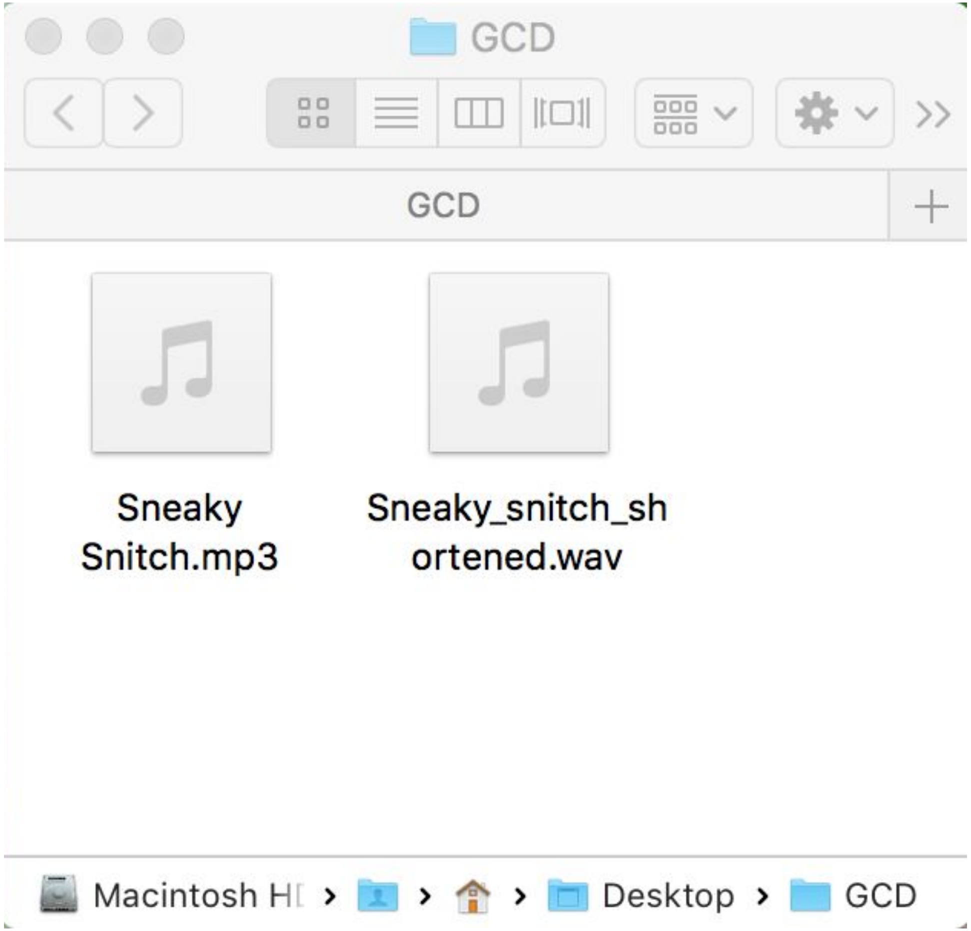
```
# Use the readMP3() function to read in your MP3 file
sneaky_snitch_full <- readMP3("~/Desktop/GCD/Sneaky Snitch.MP3")

# Call the Wave object to get some information about the audio file
sneaky_snitch_full
    Wave Object
            Number of Samples:      6023808
            Duration (seconds):     136.59
            Samplingrate (Hertz):   44100
            Channels (Mono/Stereo): Stereo
            PCM (integer format):   TRUE
            Bit (8/16/24/32/64):    16
# Play the audio file
play(sneaky_snitch_full, "open")
```



```
# Use extractWave() to extract the first five seconds of audio
sneaky_snitch_short <- extractWave(sneaky_snitch_full, from = 0, to = 5, xunit = "time")

# Preview the file to confirm the length is 5 seconds
sneaky_snitch_short

      Wave Object
              Number of Samples:      220500
              Duration (seconds):     5
              Samplingrate (Hertz):   44100
              Channels (Mono/Stereo): Stereo
              PCM (integer format):   TRUE
              Bit (8/16/24/32/64):    16


# Play the resulting audio clip
play(sneaky_snitch_short, "open")
```

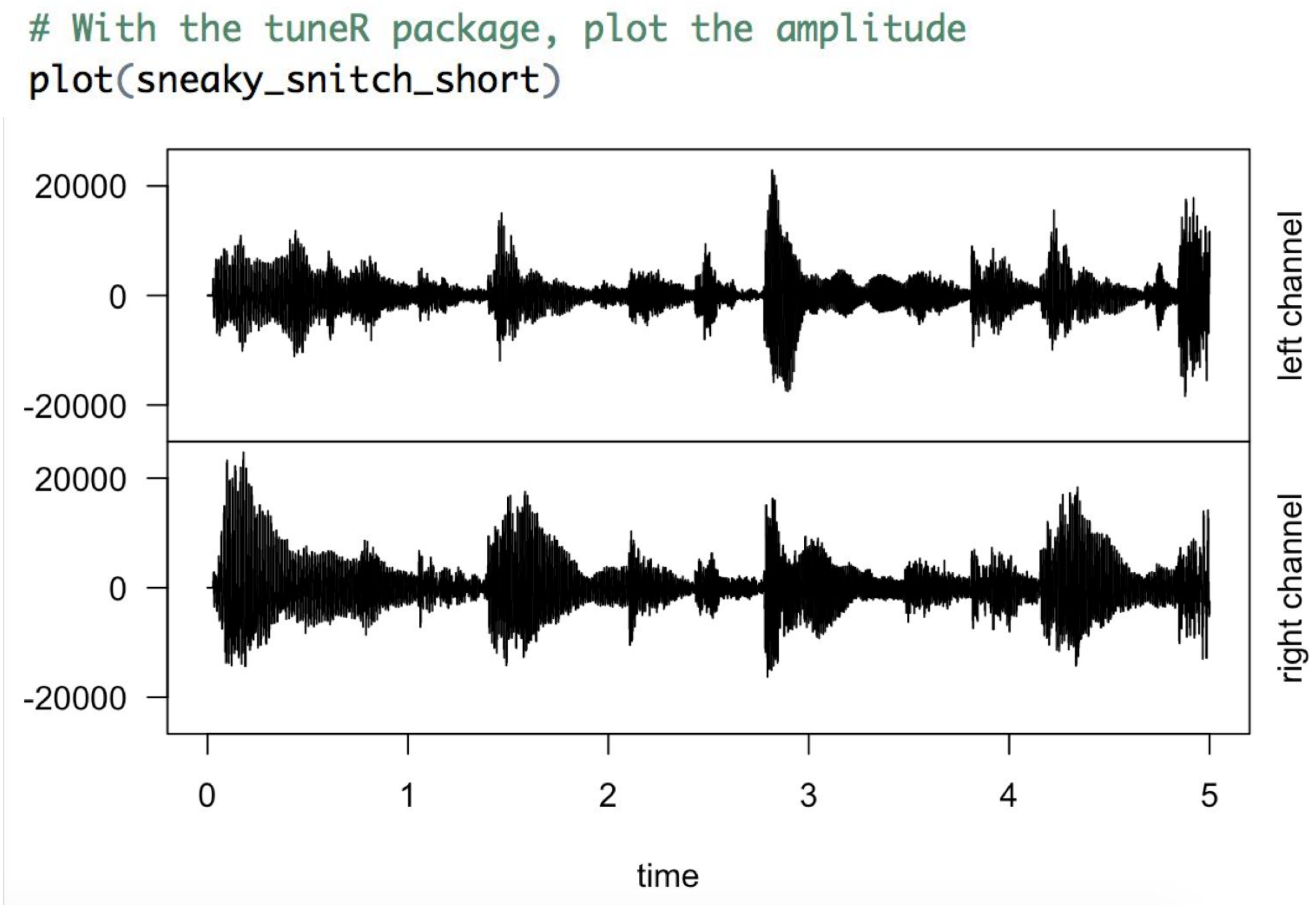


```
# Use the writeWave() function to write out a WAV file of our shortened audio clip
writeWave(sneaky_snitch_short, "~/Desktop/GCD/Sneaky_snitch_shortened.wav")
```



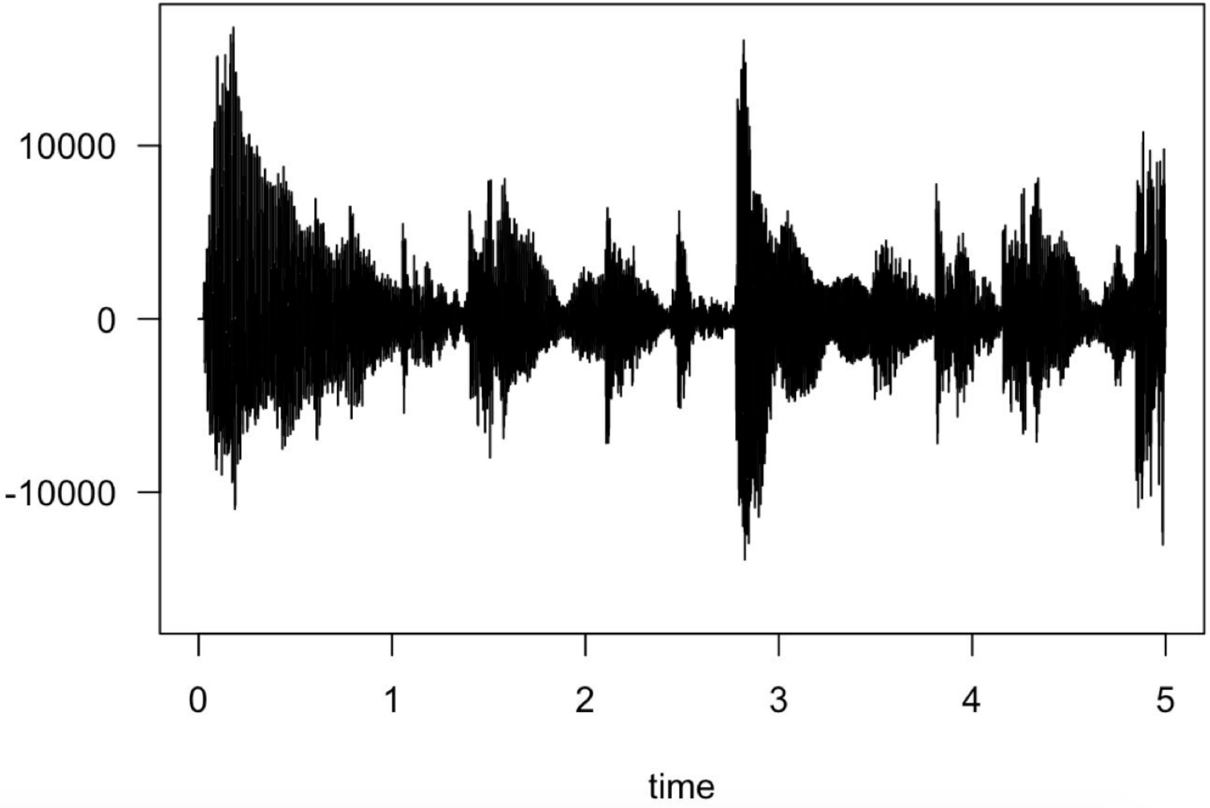
Sneaky Snitch.mp3

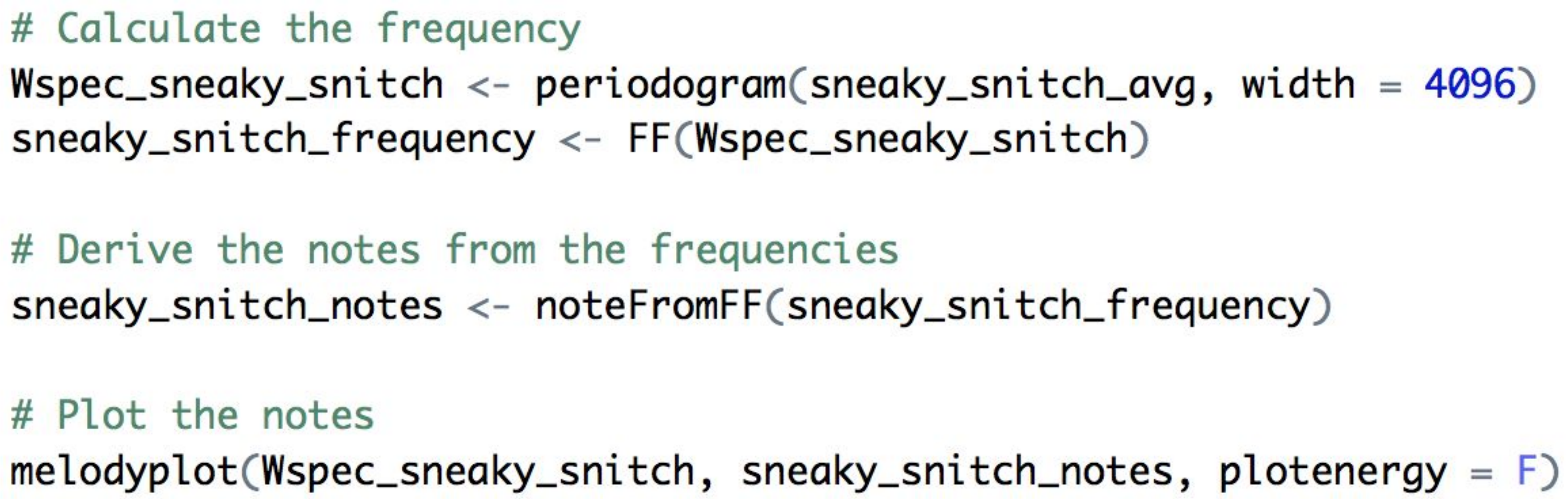
Sneaky_snitch_shortened.wav



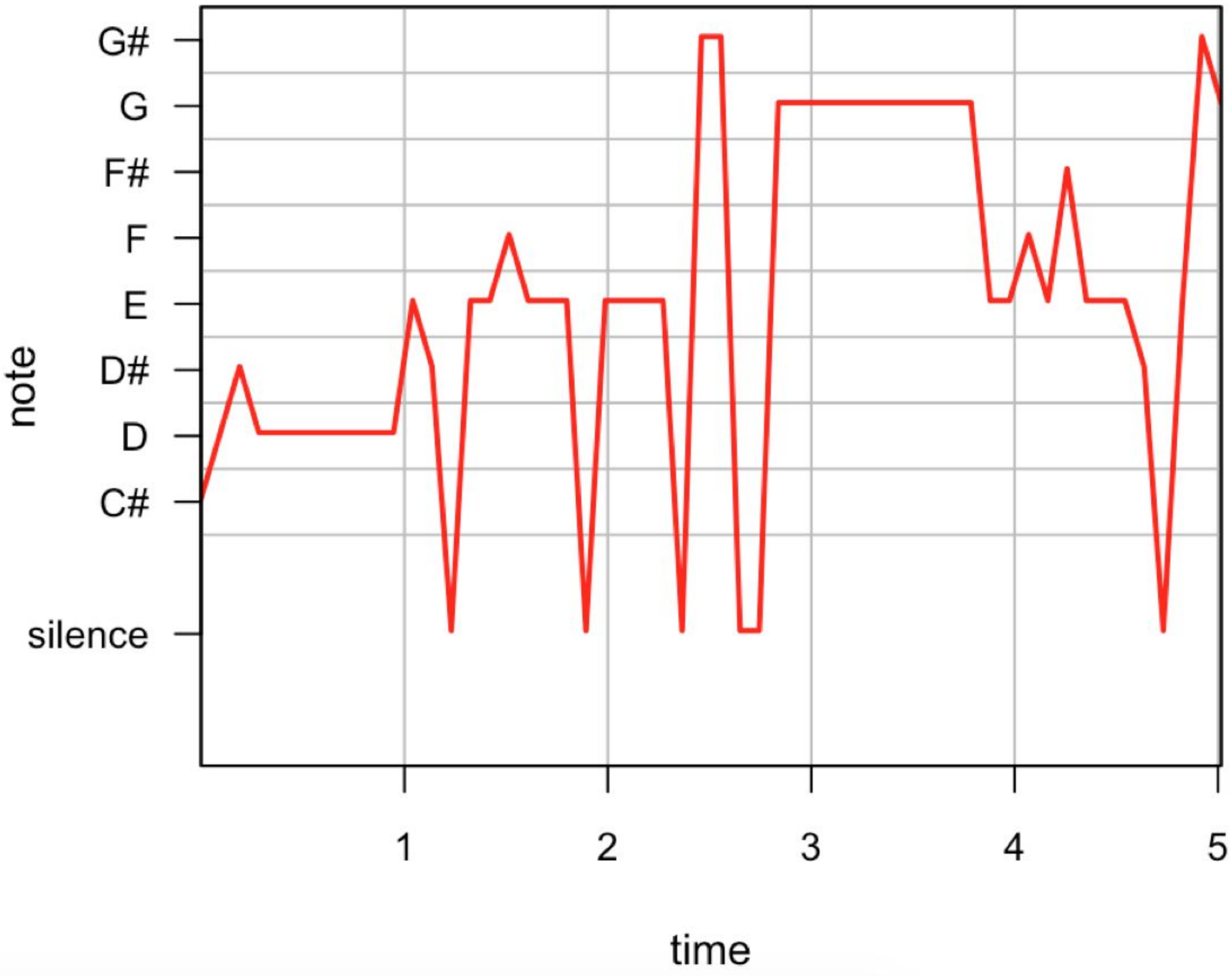
```r
# With the tuneR package, plot the amplitude
plot(sneaky_snitch_short)
```

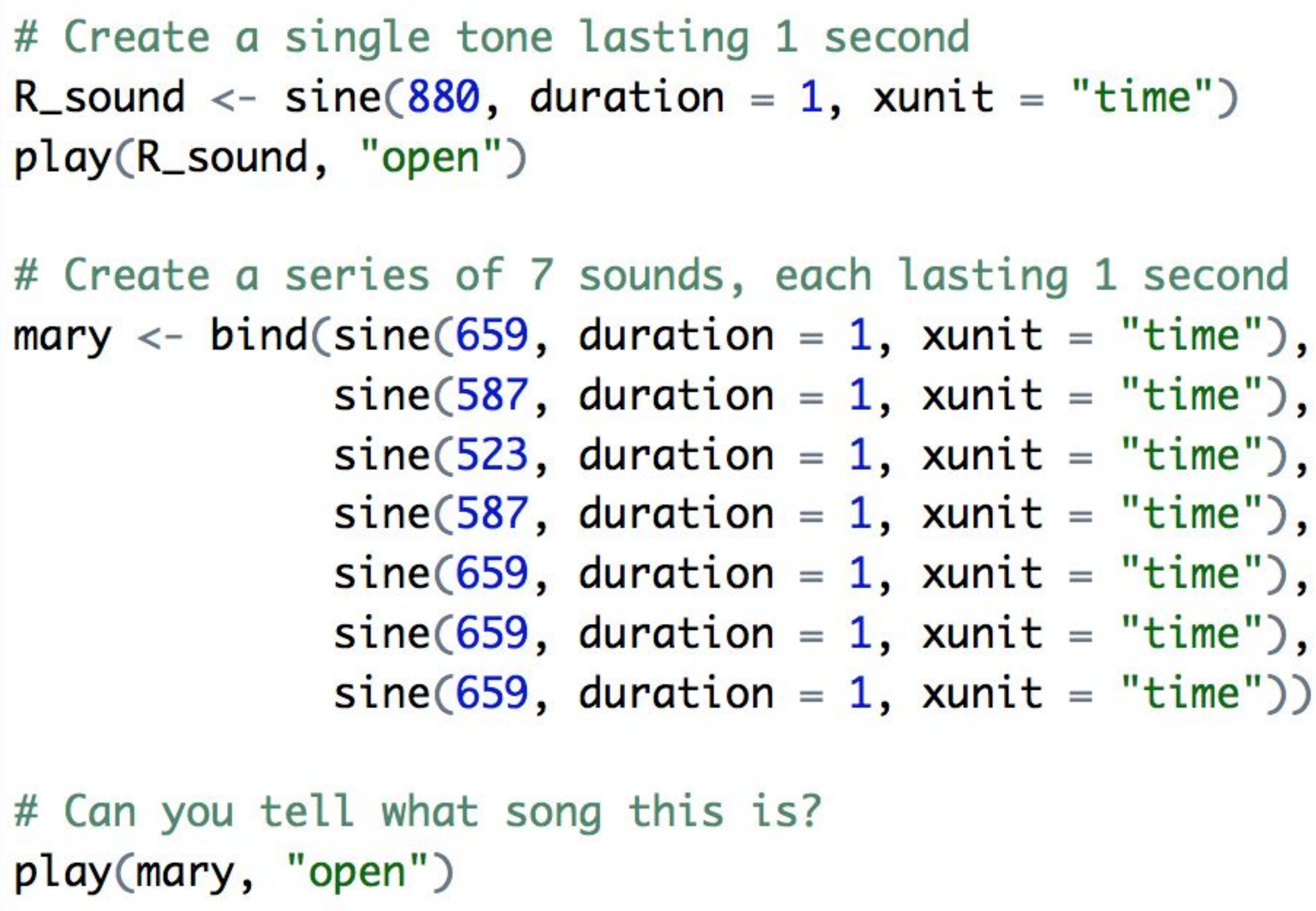
```r
# Average the two channels using mono()
sneaky_snitch_avg <- mono(sneaky_snitch_short, "both")

# Plot the average amplitude
plot(sneaky_snitch_avg)
```



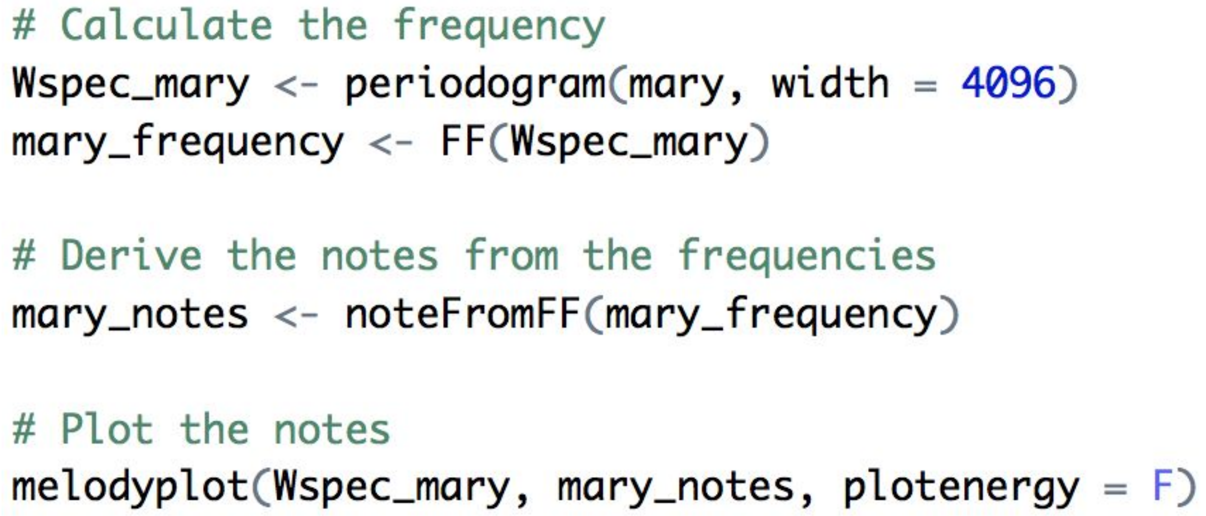


```r
# Calculate the frequency
Wspec_sneaky_snitch <- periodogram(sneaky_snitch_avg, width = 4096)
sneaky_snitch_frequency <- FF(Wspec_sneaky_snitch)

# Derive the notes from the frequencies
sneaky_snitch_notes <- noteFromFF(sneaky_snitch_frequency)

# Plot the notes
melodyplot(Wspec_sneaky_snitch, sneaky_snitch_notes, plotenergy = F)
```


G#
G
F#
F
E
D#
D
C#
silence
note
1
2
3
4
5
time

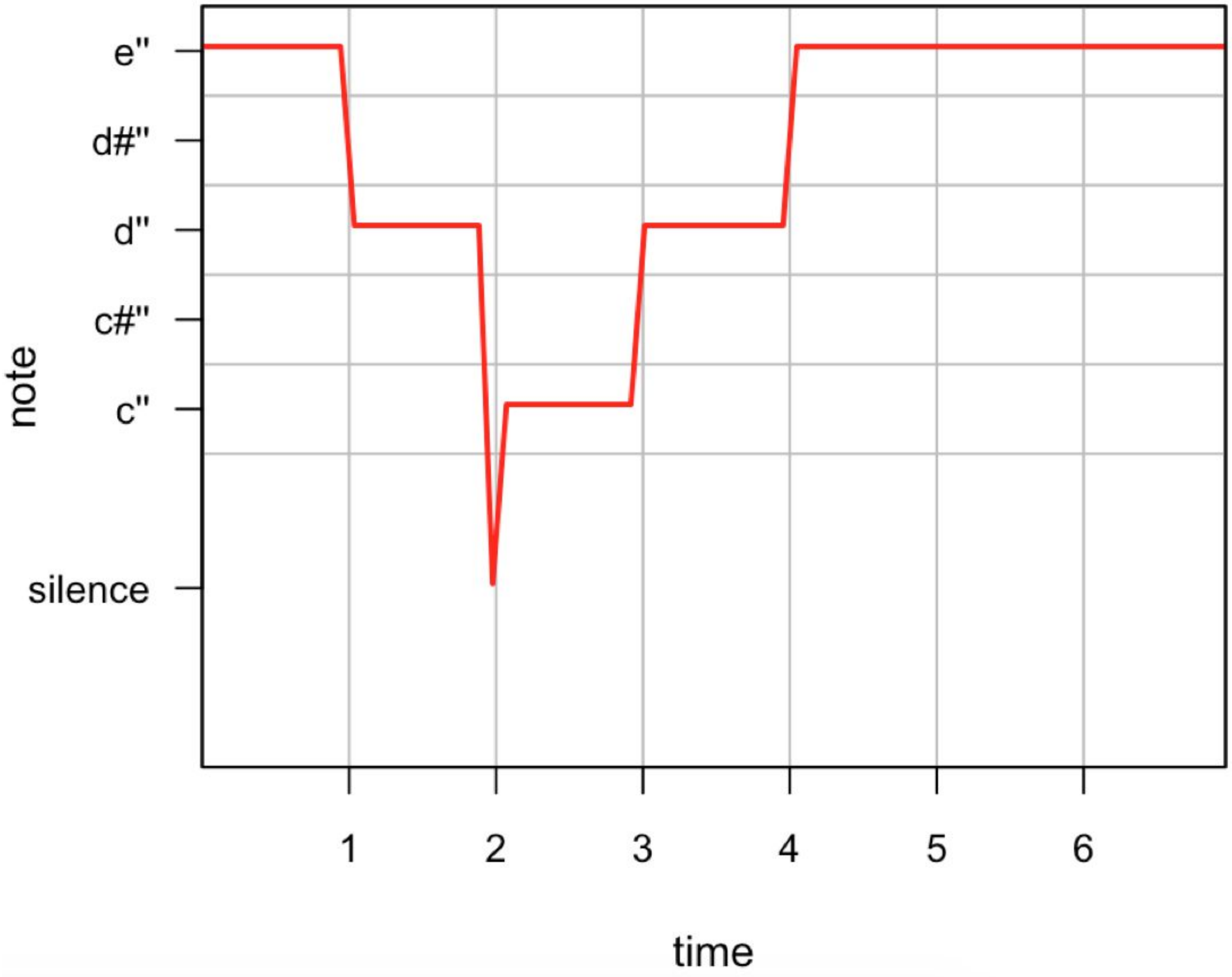


```r
# Create a single tone lasting 1 second
R_sound <- sine(880, duration = 1, xunit = "time")
play(R_sound, "open")

# Create a series of 7 sounds, each lasting 1 second
mary <- bind(sine(659, duration = 1, xunit = "time"),
             sine(587, duration = 1, xunit = "time"),
             sine(523, duration = 1, xunit = "time"),
             sine(587, duration = 1, xunit = "time"),
             sine(659, duration = 1, xunit = "time"),
             sine(659, duration = 1, xunit = "time"),
             sine(659, duration = 1, xunit = "time"))

# Can you tell what song this is?
play(mary, "open")
```



```r
# Calculate the frequency
Wspec_mary <- periodogram(mary, width = 4096)
mary_frequency <- FF(Wspec_mary)

# Derive the notes from the frequencies
mary_notes <- noteFromFF(mary_frequency)

# Plot the notes
melodyplot(Wspec_mary, mary_notes, plotenergy = F)
```

Background vector created by macrovector at www.freepik.com


e''
d#''
d''
c#''
c''
silence
note
1
2
3
4
5
6
time

Summarizing: Audio Analysis

Getting and Cleaning Data