

Lab 2 – MATH 240 – Computational Statistics

Jake Schneider
Colgate University
Department of Mathematics
jdschneider@colgate.edu

2/4/2025

Abstract

In this lab we explored file directories and subdirectories creating vectors and loops that allowed us to sort and process .WAV files and .JSON files. These loops helped us to create a list of song names and to extract sound metrics effectively. In the first task we created a batch file that displayed our desired output file name for each song and the second task we extracted different sound metrics from one .JSON file.

Keywords: Vectors; Objects; Loops; Libraries

1 Introduction

The purpose of this lab is to process a directory of audio files into an executable file and to extract different musical metrics from a song. Processing each file individual is extremely tedious and time consuming. To solve this problem we need to come up with a solution that automates this process making it as efficient as possible, especially with very large files.

In the first task we create a loop that enables us to pull the different elements from our music directory so that we can create our executable file to process each track. Creating this output.

```
streaming_extractor_music.exe "EXAMPLE.wav"  
"EXAMPLE.json"
```

In the second task we download the .JSON output for the song Au Revoir on the Talon Of The Hawk album by The Front Bottoms. We extract various musical metrics and create a data table to display the metrics that we have collected.

2 Methods

The data we worked with was different for each part of the lab. The data that we are working with in the first task is a directory of .WAV files that were created by our teacher and we need to create a loop in order to sort through the subdirectories in order to find out different .WAV files. The data that we work with in the second task is a .JSON file for a song and we need to extrapolate different musical measurement from this file.

2.1 Batch File Processing

For the first task we needed to automate the extraction of the audio file attributes. To process the .WAV files efficiently we created a list of subdirectories within our MUSIC/ file. Using the number of slashes(/) allowed us to filter subdirectories. By using loops we were able to extract the .WAV files from each subdirectory efficiently and then use the stringr package (Wickham, 2023) to help format them into the batch file using the `str_split()` and `str_sub()` functions. We can then use our paste command to produce our desired outputs to the batch file.

2.2 .JSON Processing

In the second task process the data in the .JSON file required us to to install the jsonlite package (Ooms, 2014) and use the stringr package to extract the artist, album and track from the file name. To extract our musical data we need to use `fromJSON()` function giving us a large list and from there we can then extract average loudness, spectral energy, danceability, bpm, msusical key, msucial mode and length. We then used this data to create a data frame.

3 Results

The automated process that we created for the first task successfully allowed us to gather and display the information that we were seeking in a batch file which we can now use later. From our second task we were able to successfully pull out each of the desired musical elements that we were seeking and put them into a data frame for analysis in the future.

4 Discussion

This lab showcased how beneficial it is to use loops in order to help process very large data files quickly. The batch file creation enable us to breakdown a large file of data quickly and effectively so that we have the ability to analyze it further in the future. The second part of the lab was similar to the first except instead of creating a loop we manually extracted the data which is still beneficial as we can go back thorough this code and put loops in it to automate it.

Overall, this lab displayed the importance of using loops and creating vectors to help automate data extraction

and sortment from large files.

References

- Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between json data and r objects. *arXiv:1403.2805 [stat.CO]*.
- Wickham, H. (2023). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.5.1.