

# Lab 2 – MATH 240 – Computational Statistics

Caroline Devine  
Colgate University  
Math Department  
cdevine@colgate.edu

2/03/2025

## Abstract

Lab 2 focuses on how to automate data extraction for .WAV files through batch processing as well as extracting and analyzing musical features from JSON data.

**Keywords:** Data Collection, Lists, Batch Files, For Loops, JSON package

## 1 Introduction

This lab is based off a project by Professor Cipolli and his interest in the two bands: The Front Bottoms and Manchester Orchestra. His main question was to answer which band contributed most to the song they collaborated on - Allen Town. He analyzed 180 tracks excluding "Allentown" to analyze the data about what each band's sound like. These music files have metadata that we can analyze about these bands.

Our task was to learn how to process each track (.WAV file) in an example music folder provided through creating a batch file to ensure that we did not have to complete the same command or analysis multiple times depending on how many songs we had. Otherwise, our work would be inefficient and more vulnerable to errors or missing information.

The goal was to learn how to install, load, and learn how to use libraries; work with character objects; code for() loops; and access elements of vectors and lists.

Separated into Task 1 and Task 2, we built a batch file for data processing from a set of .WAV files using the four goals above as well as extracted elements from a previously provided .JSON file.

## 2 Methods

The dataset used was a directory called "Music" containing two artists: OfficeStuff and PeopleStuff. Each of the artists consisted of four albums with 2 .WAV files for each. These were representative of the two songs in each album.

### 2.1 Task 1: Building a Batch File for Data Processing

The first step in converting the provided .WAV files to .json files to build a batch file was installing the **stringr** package

which allows for string manipulation when extracting information out of the directories, subdirectories, and track files in the Music folder.

Steps to batch file:

1. Retrieve the albums from Music folder using `list.dirs()` function
2. Using **stringr** functions to isolate and count .WAV files for each of the albums
3. Extract track number, artist name, track title, album name from track file names and directory
4. Create command lines for each .WAV file to convert into the naming convention for .json files
5. Save the converted .json files to a text file called `batfile.txt`

Through this process, we used **stringr** package in R to do the majority of the manipulation of the .WAV files and extract meaningful data (Wickham, 2023). The `for()` loop function was used to automate the system for more efficient processing and future use.

### 2.2 Task 2

Now that we created a batch file, the next goal was to process a JSON file for key musical features to analyze. Instead of using the files from Task 1, we were provided with the .json file output for "The Front Bottoms - Talon Of The Hawk - Au Revoir (Adios)". The extracted data needed was the track's average loudness, spectral energy, danceability, temp in beats per minute, musical key and mode, and the duration in seconds.

Steps to extract the metadata:

1. Install **jsonlite** package used for reading and parsing JSON files in R (Ooms, 2014).
2. Use **stringr** package again to extract the artist, album, and track from the file name.
3. Use `fromJSON()` function to read the provided .json file
4. Extract the 7 necessary musical features

This process obtains key musical features of the provided track. The `for()` loop provides efficiency for automation for all files in future. This completes the goal stated in the introduction, allowing for comparison to start.

### 3 Results

For Task 1, we successfully converted .WAV files into .json files, exporting them into a batch file. For Task 2, we achieved the correct loudness, energy, danceability, tempo, key, mode and duration for the provided track's json file output.

### 4 Discussion

This process is now automated and can be replicated with other files as well. This creates an efficient way to compare larger datasets of multiple artists or albums.

### 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.