

# Lab 2 Report – Math240 – Computational Statistics

Jeremy Artiga  
No Affiliation  
No Department  
jartiga@colgate.edu

## Abstract

In this lab, we will examine a set of non-copyrighted .WAV files from a Music directory using Base R commands and packages. More specifically, we will analyze data within the .WAV files to build a batch file of commands that will allow data processing within the open-source program Essentia, which allows for music analysis, description, and synthesis. Our batch file will allow us to process *large* sets of .WAV files within Essentia within seconds w/o the need to process each file manually. We will also experiment with the `jsonlite` package, which allows us to extract musical analysis data from given .JSON files.

**Keywords:** This lab will enable us to practice (1) installing, loading, and learning about libraries/packages; (2) working with character objects; (3) writing `for()` loops; and (4) creating/accessing elements within vectors and lists

## 1 Introduction

For this lab, we will:

1. Install the `stringr` package for R (Wickham, 2023).
2. Figure out the commands needed to access a directories and .WAV files within a Music directory.
3. Create character/vector/list objects to store data within a Music directory.
4. Utilize `for()` loops to iterate through vector/lists to search/mutate data in a Music directory.
5. Utilize functions in the `stringr` package to subset and track .WAV file data.
6. Write to a .txt file using `writeln()`.
7. Install and load the `jsonlite` package for R.
8. Utilize the `jsonlite` package to load musical analysis data. (Ooms, 2014)
9. Utilize objects to save necessary musical data.

## 2 Methodology

In this section we will go through the process of loading the needed directory/files and packages; creating objects; and utilizing coding structures to accomplish our objectives.

We ultimately want to write command lines in the following format:

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

### 2.1 Package and Object Set Up

We will explore a file directory called MUSIC, which contains all the .WAV files we will examine, this directory will be downloaded and saved into the same directory as our R script.

We will then begin our R script by loading the `stringr` package using `library()`, this will allow us to perform string mutations. Now, we setup the objects which will store the subsets of our directory (i.e. albums, song lists, etc.) using descriptive variable naming. All the objects will correspond to data we need to store in order to write our Essentia command lines.

### 2.2 Code Structures

#### 2.2.1 Subsetting File Data

In order to write our batch file command lines, we will need to go through every .WAV file in the sub directories of the MUSIC directory, these sub directories are the albums of our .WAV files (i.e. song files).

We will begin by utilizing a `for()` loop to gather all the paths of the albums sub directories within the MUSIC directory. In other words, we are getting all the albums stored in the MUSIC directory.

We then have to go through each album sub directory to obtain all the .WAV files in the directory. This will require the use of nested `for()` loops, one for the album directories and one for the .WAV files. This will give us all of the .WAV files.

Next we will have to gather all the .WAV files and all our album directories and construct full directory paths with the albums and songs we obtained earlier. We will utilize another nested `for()` loop for this task as well. I also added in a structure that filters the .WAV files from the album directory.

## 2.2.2 Constructing Batch File Commands

Now we have to grab only the artists names and track names using the `stringr` package. By utilizing the naming convention for files, we can subset the track and artists names using `str_sub()` and `str_split()`, we'll perform this operation on all the `.WAV` files using a `for()` loop.

We will construct the `.json` output files by pasting the artists names, album name, and track name together with a `.json` extension at the end using the `paste()` function. The following format is as follows:

```
[artist name]-[album name]-[track name].json
```

Finally, we will paste together the string "streaming\_extractor\_music.exe" with our `.WAV` file name and our `.json` file name to produce the batch file command lines.

Using the `writeln()` we can write our batch file commands into a TXT file called `batfile.txt`, which will contain all the music processing command lines for Essentia to execute.

## 2.3 Extracting Data Using the jsonlite Package

Suppose we have a `.JSON` output song file we want to analyze. We can use the `jsonlite` package to obtain all sorts of musical data. From the key, tone, all the way to danceability! To illustrate this, we will use the `.JSON` output for "Au Revoir (Adios) on the Talon Of The Hawk" album by The Front Bottoms.

We will firstly load the `jsonlite` package using the `library()` function. Then we want to use `str_split()` to extract the artists, album, and track names from the file. This will be useful for a later project.

Now we want to load the `JSON` file into R using the `fromJSON()` function. This will load a list of all the musical data attributed to our file. Finally, we want to extract the data in the resulting list and store them into objects. We can do this by accessing lower-level elements in the list (much like a directory) using `$` syntax.

## 3 Discussion

By utilizing packages and base R coding structures such as `for()` loops, objects, and accessing elements within vectors/lists, we were able to create an Essentia command batch file from examination of `.WAV` files in the music directory. This not only enabled us to practice our abilities to utilize the said coding structures, but it also allowed us to build an R program which could analyze any sort of directory and pull the `.WAV` files for the sake of data processing into Essentia. We also figured out how to extract data from `.JSON` files using the `jsonlite` package to gather information on specific albums/songs, which we could potentially utilize to analyze songs and make general insights based on this analysis in another project.

## 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.