

# Lab 02 – MATH 240 – Computational Statistics

Harrison Wolfe  
Colgate University  
Math  
hwolfe@colgate.edu

## Abstract

This goal of this lab was to create a batch file containing a large number of codes to run to turn .wav files into .json files. Another goal of this lab was to start using large sets of data to analyze json files.

**Keywords:** Loops; Batch files; Organizing data; Subsetting

## 1 Introduction

This paper is intended to describe the process in which we can extract data from music and use that data to analyze trends. There are several important components to music like Key Signature, Time Signature, Mode, Tempo, Loudness (**Dynamics**) and many more. Music can be comprised of vectors of information which is what the `jsonlite` package provides to us. (Ooms, 2014)

### 1.1 Tasks

In this lab we are dealing with 14 "songs" which are wav files recorded from various things laying around an office or something that a person could make. We want to use these songs to analyze musical trends about them and in doing so we need to convert their file type from a .wav to a .json. To do this we need a batch file to repeat this process several times without having to manually input each one. The other task as a part of this lab is to gather data about a song, "**Au Revoir**", which is already a json file.

## 2 Methods

### 2.1 Task 1 Methods

In this task there were many steps required to organize the batch file. First it was important that we pulled the directories from the folder of music using the `list.dirs` command and then using the `list.files` we could list the files from each directory. Then, using the `stringr` package we were able to set it up in a way to be processed. (Wickham, 2023) With this package we were able to create both the input and output file command within the batch file. We first had to list the exact file by using the `list.files` function for the input and setting that as a vector with an ending of .wav to signify that this is what we want to turn into a json file. Then, for the output as we wanted in the format of album-artist-song.json we needed to separate the name using the `stringr` package saving the artist, album, and song separately with .json at

the end to show the output should be a json file. Then again using the `paste` tool we could add that all together making one command in the batch file. Using for loops for each of these processes we could replicate this for all 14 songs to make a complete batch file.

### 2.2 Task 2 Methods

In this task we used the `jsonlite` package to give us various information about a given song. By installing the package and using the `fromJSON` command we were able to see all of this data about the json file. It gave us information like the Key, Tempo, and many others. After saving this information to an object we could use the subsetting tool to figure out this various information using help from the guide to find the exact path of the attributes needed from the song.

## 3 Results

While there were not many results from this lab we had two products from each task. The first product was a complete batch file which will later allow us to convert the .wav files into .json files. Doing this will allow us to use the `jsonlite` package to extract the information that we want from the songs. This information will give a statistical background for the style of a given artist and also numerically define differences between music. The other product from this lab was a list of information about the song **Au Revoir** from the album **Talon Of The Hawk** by **The Front Bottoms**.

## 4 Discussion

There is very little subjectivity from this lab as it was a methodical assignment that explored two tasks which could either be done correctly or incorrectly. As there was no real product it is hard to evaluate something and present conclusions from data when we are just producing something rather than analyzing. While there was the list of data given from the song **Au Revoir**, there were no hypothesis testing, confidence intervals, or any other informative processes to draw from the data.

## 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.