

1. In Lab 3, you wrangled data from Essentia, Essentia models and LIWC. Rework your solution to Lab 3 using **tidyverse** (Wickham et al., 2019) instead of base R. Specifically, rewrite your code for steps 1-4 of task 2 using **tidyverse** (Wickham et al., 2019). Make sure to address any issues I noted in your code file, and ensure that your code runs in the directory as it is set up.

```
#####
# HW 5
# Avery Johnson
#####

#####
# Code Task: Compile Data from Essentia
#####

#####
# Step 0: install the stringr and jsonlite packages for R
#####

library("jsonlite")
library("tidyverse")

#####
# Step 1: Work with the song Au Revoir on the Talon of the Hawk album
#####

#Substep 1
current.filename <- "The Front Bottoms-Talon Of The Hawk-Au Revoir (Adios).json"

#Substep 2
file_parts <- str_split(current.filename, "-", simplify=TRUE)
artist <- file_parts[1]
album <- file_parts[2]
track <- file_parts[3]
track <- file_parts[3] |>
  str_remove(".json$")

#Substep 3
json_data <- fromJSON(file.path("EssentiaOutput", current.filename))

#Substep 4
overall_loudness <- json_data$loudness_ebu128$integrated
spectral_energy <- json_data$lowlevel$spectral_energy
dissonance <- json_data$lowlevel$dissonance
pitch_salience <- json_data$lowlevel$pitch_salience
bpm <- json_data$rhythm$bpm
beats_loudness <- json_data$rhythm$beats_loudness
danceability <- json_data$rhythm$danceability
tuning_frequency <- json_data$tonal$tuning_frequency

#####
# Step 2: complete step 1 for all .JSON files in the EssentiaOutput Folder
#####

json_files <- list.files("EssentiaOutput", pattern="\\.json$", full.names=TRUE) # Find all JSON files

# function to extract data in each JSON file
df_results <- json_files %>%
  #map_df applies a function to each element in the json_files list
  # .x is a placeholder that represents each element in the list
  map_df(~{
    json_data <- fromJSON(.x)

    file_parts <- str_split(basename(.x), "-", simplify=TRUE)
    artist <- file_parts[1]
    album <- file_parts[2]
    track <- file_parts[3]
    track <- file_parts[3] |>
      str_remove(".json$")

    #extract the features
    tibble(
      artist = artist,
      album = album,
      track = track,
      overall_loudness = json_data$lowlevel$loudness_ebu128$integrated,
      spectral_energy = json_data$lowlevel$spectral_energy,
      dissonance = json_data$lowlevel$dissonance,
      pitch_salience = json_data$lowlevel$pitch_salience,
      bpm = json_data$rhythm$bpm,
```

```

    beats_loudness = json_data$rhythm$beats_loudness,
    danceability = json_data$rhythm$danceability,
    tuning_frequency = json_data$tonal$tuning_frequency
  )
})

#####
# Step 3: Load and clean the data from the Essentia models by completing the
# following steps
#####

#substep 1
essentia_model <- read_csv("EssentiaOutput/EssentiaModelOutput.csv")

cleaned_essentia <- essentia_model |>
  mutate(
    valence = rowMeans(essentia_model[,c("deam_valence", "emo_valence", "muse_valence")]),
    arousal = rowMeans(essentia_model[,c("deam_arousal", "emo_arousal", "muse_arousal")]),
    aggressive = rowMeans(essentia_model[,c("eff_aggressive", "nn_aggressive")]),
    happy = rowMeans(essentia_model[,c("eff_happy", "nn_happy")]),
    party = rowMeans(essentia_model[,c("eff_party", "nn_party")]),
    relax = rowMeans(essentia_model[,c("eff_relax", "nn_relax")]),
    sad = rowMeans(essentia_model[,c("eff_sad", "nn_sad")]),
    acoustic = rowMeans(essentia_model[,c("eff_acoustic", "nn_acoustic")]),
    electronic = rowMeans(essentia_model[,c("eff_electronic", "nn_electronic")]),
    instrumental = rowMeans(essentia_model[,c("eff_instrumental", "nn_instrumental")])
  ) |>
  rename(timbreBright = eff_timbre_bright) |>
  select(artist, album, track, valence, arousal, aggressive, happy, party,
         relax, sad, acoustic, electronic, instrumental, timbreBright)

#####
# Step 4: Load the data from LIWC and compile the full dataset
#####

#substep 1
liwc_output <- read_csv("LIWCOutput/LIWCOutput.csv")

# Merge df_results and cleaned_essentia
merged_df <- df_results |>
  left_join(cleaned_essentia) |>
  left_join(liwc_output)

#substep 3
merged_df <- merged_df |>
  rename(funcnt = 'function')

```

In this assignment, I rewrote my lab 3 code using the **tidyverse** (Wickham et al., 2019) package instead of base R to improve readability and efficiency. To extract data, I used `str_split()` and `str_remove()` to obtain the artist, album, and track names from filenames. Instead of a `for` loop, I used `map_df()` to iterate through all JSON files, applying a function to extract and store features in a tibble for cleaner and more efficient processing. To clean the model data, I used `mutate()` to add new columns, along with `rename()` and `select()` to keep only relevant columns and improve clarity. For merging, **tidyverse** provides a more concise approach by allowing the use of `left_join()` multiple times within a pipeline. Overall, switching to **tidyverse** made my code more readable, efficient, and consistent while correctly performing all required tasks.

References

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.