

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.

```
#####
# Load libraries
#####
library("tidyverse")
library("jsonlite")
library("stringr")

#####
# Step 1 - Working with Au Revoir (Adios)
#####
#create a tibble to store object properties
au.revoir.features <- tibble()

#Part1 - creating an object
au.revoir.filename <- "The Front Bottoms-Talon Of The Hawk-Au Revoir (Adios).json"

#Part2 - split the string
split.au.revoir.filename <- au.revoir.filename |>
  str_sub(start = 1L, end = -6L)|> #remove trailing .json
  str_split(pattern = '-', simplify =T) #split string

#extract artist, album, and song - input them into tibble
au.revoir.features <- au.revoir.features|>
  mutate(artist = split.au.revoir.filename[1]) |> #extract the artist
  mutate(album = split.au.revoir.filename[2]) |> #extract the album
  mutate(track = split.au.revoir.filename[3]) |> #extract the track
#create a new row with all values
add_row(artist = split.au.revoir.filename[1], album = split.au.revoir.filename[2],
        track = split.au.revoir.filename[3])

#Part3 - load JSON file
essentia.output.filename <- paste("EssentiaOutput", au.revoir.filename, sep = '/')
essentia.output <- fromJSON(essentia.output.filename)

#Part4 - extract song characteristic
au.revoir.features <- au.revoir.features|>
  #extract overall loudness
  mutate(overall_loudness = essentia.output$lowlevel$loudness_ebu128$integrated) |>
  #extract spectral energy
  mutate(spectral_energy = essentia.output$lowlevel$spectral_energy$mean)|>
  #extract dissonance
  mutate(dissonance = essentia.output$lowlevel$dissonance$mean) |>
  #extract pitch salience
  mutate(pitch_salience = essentia.output$lowlevel$pitch_salience$mean)|>
  #extract tempo in beats per minute
  mutate(bpm = essentia.output$rhythm$bpm)|>
  #extract beats loudness
  mutate(beats_loudness = essentia.output$rhythm$beats_loudness$mean)|>
  #extract danceability
  mutate(danceability = essentia.output$rhythm$danceability)|>
  #extract tuning frequency
  mutate(tuning_frequency = essentia.output$tonal$tuning_frequency)

#####
# Step 2 - Load and clean data from Essentia models for each .JSON file
#####
#create a tibble to store audio properties
three.bands.features <- tibble()

#load all files from EssentiaOutput
all.files <- list.files("EssentiaOutput")
#subset all .json files
json.count <- str_count(all.files, pattern = ".json")
json.files <- subset(all.files, json.count>0)
times.to.repeat = length(json.files) #variable for the loop count

#complete Step1 for all .json files
for (i in 1:times.to.repeat){
  #current file
  curr.file = json.files[i]

  #process current file name
  split.curr.file <- curr.file |>
```

```

    str_sub(start = 1L, end = -6L)|> #remove trailing .json
    str_split(pattern = '-', simplify =T) #split string

#load JSON file
current.essentia.output.filename <- paste("EssentiaOutput", curr.file, sep = '/')
current.essentia.output <- fromJSON(current.essentia.output.filename)

#extract song characteristic into its own tibble
new.row <- tibble(
  artist = split.curr.file[1],
  album = split.curr.file[2],
  track = split.curr.file[3],
  overall_loudness = current.essentia.output$lowlevel$loudness_ebui28$integrated,
  spectral_energy = current.essentia.output$lowlevel$spectral_energy$mean,
  dissonance = current.essentia.output$lowlevel$dissonance$mean,
  pitch_salience = current.essentia.output$lowlevel$pitch_salience$mean,
  bpm = current.essentia.output$rhythm$bpm,
  beats_loudness = current.essentia.output$rhythm$beats_loudness$mean,
  danceability = current.essentia.output$rhythm$danceability,
  tuning_frequency = current.essentia.output$tonal$tuning_frequency)

#add new row for current song to the tibble
three.bands.features <- three.bands.features %>%
  bind_rows(., new.row)
}

#####
# Step 3 - Load and clean EssentiaModelOutput.csv
#####
#Part1 - load csv file
essentia.model.output <- read_csv("EssentiaOutput/EssentiaModelOutput.csv")

essentia.model.output <- essentia.model.output %>%
  #Part2 - add valance and arousal columns
  #create column for valence by taking a mean of three columns
  mutate(valence = rowMeans(select(., deam_valence, emo_valence, muse_valence))) %>%
  #create column for arousal by taking a mean of three columns
  mutate(arousal = rowMeans(select(., deam_arousal, emo_arousal, muse_arousal))) %>%
  #Part3 - new mood columns with features
  #create column for aggressive by taking a mean of two columns
  mutate(aggressive = rowMeans(select(., eff_aggressive, nn_aggressive))) %>%
  #create column for happy by taking a mean of two columns
  mutate(happy = rowMeans(select(., eff_happy, nn_happy))) %>%
  #create column for party by taking a mean of two columns
  mutate(party = rowMeans(select(., eff_party, nn_party))) %>%
  #create column for relaxed by taking a mean of two columns
  mutate(relaxed = rowMeans(select(., eff_relax, nn_relax))) %>%
  #create column for sad by taking a mean of two columns
  mutate(sad = rowMeans(select(., eff_sad, nn_sad))) %>%
  #Part4 - acoustic and electric averaging
  #create column for acoustic by taking a mean of two columns
  mutate(acoustic = rowMeans(select(., eff_acoustic, nn_acoustic))) %>%
  #create column for electric sound by taking a mean of two columns
  mutate(electric = rowMeans(select(., eff_electronic, nn_electronic))) %>%
  #Part5 - compute instrumental
  #create column for instrumental by taking a mean of two columns
  mutate(instrumental = rowMeans(select(., eff_instrumental, nn_instrumental))) %>%
  #Part6 - rename eff_timbre_bright column
  rename(timbreBright = eff_timbre_bright) %>%
  #Part7 - retained created features and columns for artists, album, and track
  select(artist, album, track, valence, arousal, aggressive,
    happy, party, relaxed, sad, acoustic, electric,
    instrumental, timbreBright)

#####
# Step 4 - Load LIWC data and compile full data set
#####
#Part 1 - load csv file
lyrics.analysis <- read_csv("LIWCOutput/LIWCOutput.csv")

#Part 2 - merge the data
common.columns <- c("artist", "album", "track")
final.three.band.data <- three.bands.features |>
  inner_join(essentia.model.output, by = common.columns) |>
  inner_join(lyrics.analysis, by =common.columns)|>
  rename(funct = "function") #Part 3 - rename function column

```

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.