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.

```
############################################################
###                       Step 2                         ####
#Completing step 1 for every JSON File
############################################################
library(stringr)
library(jsonlite)
library(tidyverse)


#Making our empty dataframe
empty.dataframe = data.frame(
  artist = character(),
  album = character(),
  track = character(),
  overall_loudness = numeric(),
  spectral_energy = numeric(),
  dissonance = numeric(),
  pitch_salience = numeric(),
  bpm = numeric(),
  beats_loudness = numeric(),
  danceability = numeric(),
  tuning_frequency = numeric()
)

song.dataframe = empty.dataframe

#Pulling list of all tracks to be iterated
folder.name = 'EssentiaOutput'
all.tracks = list.files(folder.name)

#Making sure only valid json files are being used
json.count = str_count(all.tracks, ".json")
track.indices = which(json.count == 1)


df.index = 1 #Initializing where to input in dataframe
for (i in track.indices) {
  #Running for loop over every track

  ############################################################
  ###                     Step 1                          ####
  # Pulling file and extracting necessary info from JSON
  ############################################################
  current.filename = all.tracks[i]
  #current.filename = "The Front Bottoms-Talon Of The Hawk-Au Revoir (Adios).json"
  current.songname = str_sub(current.filename, end = -6)

  #Making a vector containing artist, album, and song
  song.vector = str_split_1(current.songname, "-")

  ###Pulling data from JSON file
  json.pathname = paste(folder.name, current.filename, sep = '/')
  json.track = fromJSON(json.pathname)


  song.dataframe[df.index, ] <- tibble( #Adding the information directly through a tibble
    artist = song.vector[1],
    album = song.vector[2],
    track = song.vector[3],
    overall_loudness = json.track$lowlevel$average_loudness,
    spectral_energy = json.track$lowlevel$spectral_energy$mean,
    dissonance = json.track$lowlevel$dissonance$mean,
    pitch_salience = json.track$lowlevel$pitch_salience$mean,
    bpm = json.track$rhythm$bpm,
    beats_loudness = json.track$rhythm$beats_loudness$mean,
    danceability = json.track$rhythm$danceability,
    tuning_frequency = json.track$tonal$tuning_frequency
  )


  df.index=df.index+1 #Moving to next dataframe index
}
```

```
############################################################
###                      Step 3                          ####
# Making new dataframe using essentia models
############################################################
#Loading our dataframe file
output.model = read.csv("EssentiaOutput/EssentiaModelOutput.csv")

###Writing the matrix utilizing tinyverse
###First focusing on transitioning to using means, then will select only desired columns
output.model = output.model %>%
  #I'm a little confused how rowwise works, but when doing some research this seemed
  #the best way to accomplish my task, is this saying that from anything past that pipe
  #I want to have my functions apply to the entirety of the row?
  rowwise() %>%
  mutate(
    arousal = mean(c(emo_arousal, deam_arousal, muse_arousal)),
    valence = mean(c(emo_valence, deam_valence, muse_valence)),
    aggressive = mean(c(eff_aggressive, nn_aggressive)),
    happy = mean(c(eff_happy, nn_happy)),
    party = mean(c(eff_party, nn_party)),
    relaxed = mean(c(eff_relax, nn_relax)),
    sad = mean(c(eff_sad, nn_sad)),
    acoustic = mean(c(eff_acoustic, nn_acoustic)),
    electric = mean(c(eff_electronic, nn_electronic)),
    instrumental = mean(c(eff_instrumental, nn_instrumental)),
    timbreBright = eff_timbre_bright
  )
###Now that all desired columns are made, we retain only desired
essentia.dataframe = output.model %>%
  select(artist, album, track, arousal, valence, aggressive, happy,
         party, relaxed, sad, acoustic, electric, instrumental, timbreBright)


############################################################
###                      Step 4                          ####
# Making a final dataframe that includes all information
############################################################
lyric.output = read.csv("LIWCOutput/LIWCOutput.csv")
#Loading dataframe


###Merging together our .wav data dataframe and our essentia dataframe
final.dataframe = merge(song.dataframe, essentia.dataframe,
                        by = c("artist", "album", "track")
)


###Adding our lyric dataframe
final.dataframe = merge(final.dataframe, lyric.output,
                        by = c("artist", "album", "track")
)

# Rename function column
final.dataframe = final.dataframe %>%
  rename(func = function.)

#Making every column that is numerical actually have numeric class
final.dataframe[, 4:ncol(final.dataframe)] <- lapply(final.dataframe[, 4:ncol(final.dataframe)], as.numeric)


############################################################
###                      Step 5                          ####
# Separating out Allentown
############################################################
#Making a file that contains everything except Allentown
write.csv(final.dataframe[final.dataframe$track!="Allentown", ], "trainingdata.csv")

#Making a file that ONLY contains Allentown
write.csv(final.dataframe[final.dataframe$track=="Allentown", ], "testingdata.csv")
```

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.