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.

```
#loading libraies
library(tidyverse)
library(jsonlite)

#load csv data files
essentia.csv = read_csv("EssentiaOutput/EssentiaModelOutput.csv")
liwc.csv = read_csv("LIWCOutput/LIWCOutput.csv")

#object setup
artist = c()
album = c()
track = c()
overall.loudness = c()
spectral.energy = c()
dissonance = c()
pitch.sailence = c()
tempo.bpm = c()
beat.loudness = c()
danceability = c()
tuning.freq = c()

#Getting list of .json songs and extracting data from each song
essentia.song.list = list.files(path = "EssentiaOutput")
for(song in essentia.song.list){
  if ((str_sub(song, start = -5) == ".json")){
    song.extract = str_split_1(str_sub(song, start = 1, end = -6), "-")
    artist = c(artist, song.extract[1])
    album = c(album, song.extract[2])
    track = c(track, song.extract[3])
    song.path = paste("EssentiaOutput", song, sep = "/")
    curr.json = fromJSON(song.path)
    overall.loudness = c(overall.loudness, curr.json$lowlevel$loudness_ebu128$integrated)
    spectral.energy = c(spectral.energy, curr.json$lowlevel$spectral_energy$mean)
    dissonance = c(dissonance, curr.json$lowlevel$dissonance$mean)
    pitch.sailence = c(pitch.sailence, curr.json$lowlevel$pitch_salience$mean)
    tempo.bpm = c(tempo.bpm, curr.json$rhythm$bpm)
    beat.loudness = c(beat.loudness, curr.json$rhythm$beats_loudness$mean)
    danceability = c(danceability, curr.json$rhythm$danceability)
    tuning.freq = c(tuning.freq, curr.json$tonal$tuning_frequency)
  }
}

#extracted data fused into a data frame
music.data = cbind(artist,
                   album,
                   track,
                   overall.loudness,
                   spectral.energy,
                   dissonance,
                   pitch.sailence,
                   tempo.bpm,
                   beat.loudness,
                   danceability,
                   tuning.freq)

#Creating the Master data set
final.dataframe <- essentia.csv |>
  rowwise() |>
  mutate(valence = mean(c(deam_valence,
                          emo_valence,
                          muse_valence)),
         arousal = mean(c(deam_arousal,
                          emo_arousal,
                          muse_arousal)),
         agressive = 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))) |>
  ungroup() |>
  rename(timbreBright = eff_timbre_bright) |>
  select("artist",
         "album",
         "track",
         "timbreBright",
         "valence",
         "arousal",
         "agressive",
         "happy",
         "party",
         "relaxed",
         "sad",
         "acoustic",
         "electric",
         "instrumental") %>%
  left_join(as_tibble(music.data), by = c("album", "track")) %>%
  left_join(liwc.csv, by = c("album", "track")) |>
  select(-artist, -artist.y) |>
  rename(artist = artist.x)

################################################################################
#Box Plot Analysis
################################################################################
train.data = read.csv("trainingdata.csv")

ggplot(aes(x = artist,
           y = as.numeric(spectral.energy)),
       data = train.data,) +
  geom_violin(fill = "grey90") +
  theme_bw() +
  coord_flip() +
  labs(x = "Artists",
       y = "Spectral Energy",)

ggplot(aes(x = artist,
           y = as.numeric(arousals)),
       data = train.data) +
  geom_violin(fill = "grey90") +
  theme_bw() +
  coord_flip() +
  labs(x = "Artists",
       y = "Arousal")

ggplot(aes(x = artist,
           y = as.numeric(Authentic)),
       data = train.data) +
  geom_violin(fill = "grey90") +
  theme_bw() +
  coord_flip() +
  labs(x = "Artists",
       y = "Authenticity")
```

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.