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.

```r
#loading libraries
library(tidyverse)
library(jsonlite)

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

#Loading the List of Songs
essentia.song.list = list.files(path = "EssentiaOutput")
#Setting up the music.data dataframe(tibble)
music.data <- tibble(
  artist = character(),
  album = character(),
  track = character(),
  overall.loudness = numeric(),
  spectral.energy = numeric(),
  dissonance = numeric(),
  pitch.sailence = numeric(),
  tempo.bpm = numeric(),
  beat.loudness = numeric(),
  danceability = numeric(),
  tuning.freq = numeric()
)

for(song in essentia.song.list) { #For each song in the song list
  if ((str_sub(song, start = -5) == ".json")){ #Pick only JSON files
    song.path = paste("EssentiaOutput", song, sep = "/")
    song.extract = str_split_1(str_sub(song, start = 1, end = -6), "-") #Get artist, album, and track info
    song.data = fromJSON(song.path)

    music.data <- bind_rows(music.data, tibble( #Bind all the extracted data to the music.data tibble
      artist = song.extract[1],                 #for every song in the song list
      album = song.extract[2],
      track = song.extract[3],
      overall.loudness = song.data$lowlevel$loudness_ebu128$integrated,
      spectral.energy = song.data$lowlevel$spectral_energy$mean,
      dissonance = song.data$lowlevel$dissonance$mean,
      pitch.sailence = song.data$lowlevel$pitch_salience$mean,
      tempo.bpm = song.data$rhythm$bpm,
      beat.loudness = song.data$rhythm$beats_loudness$mean,
      danceability = song.data$rhythm$danceability,
      tuning.freq = song.data$tonal$tuning_frequency
    ))
  }
}

#Creating the Master data set
final.dataframe <- essentia.csv |>
  rowwise() |>
  mutate(valence = mean(c(deam_valence, #Computes Valence values
                          emo_valence,
                          muse_valence)),
         arousal = mean(c(deam_arousal, #Computes Arousal values
                          emo_arousal,
                          muse_arousal)),
         agressive = mean(c(eff_aggressive, #Computes Agressive values
                            nn_aggressive)),
         happy = mean(c(eff_happy, #Computes Happy values
                        nn_happy)),
         party = mean(c(eff_party, #Computes Party values
                        nn_party)),
         relaxed = mean(c(eff_relax, #Computes Relaxed values
                          nn_relax)),
         sad = mean(c(eff_sad, #Computes Sad values
                      nn_sad)),
         acoustic = mean(c(eff_acoustic, #Computes Acoustic values
                           nn_acoustic)),
         electric = mean(c(eff_electronic, #Computes Electric values
                           nn_electronic)),
         instrumental = mean(c(eff_instrumental, #Computes Instrumental values
                               nn_instrumental))) |>
  ungroup() |>
  rename(timbreBright = eff_timbre_bright) |> #rename a column to timbreBright
  select("artist", #  Select only columns with data we want
         "album",
         "track",
         "timbreBright",
         "valence",
         "arousal",
         "agressive",
         "happy",
         "party",
         "relaxed",
         "sad",
         "acoustic",
         "electric",
         "instrumental") %>%
  left_join(as_tibble(music.data), by = c("album", "track")) %>% #Join our essentia data with music.data
  left_join(liwc.csv, by = c("album", "track")) |> #join essentia data with liwc data
  select(-artist, -artist.y) |> #remove duplicated columns
  rename(artist = artist.x) #rename column to artist

###############################################################################
#Box Plot Analysis
###############################################################################
train.data = read.csv("trainingdata.csv") #Get our training data (W/O Allentown)

ggplot(aes(x = artist, #Plots distribution of artists on spectral energy
           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, #Plots distribution of artists on arousal
           y = as.numeric(arousals)),
       data = train.data) +
  geom_violin(fill = "grey90") +
  theme_bw() +
  coord_flip() +
  labs(x = "Artists",
       y = "Arousal")

ggplot(aes(x = artist, #Plots distribution of artists on Authenticity
           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.