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
library(tidyverse)
library(jsonlite)

#Code from Lab 3 for doing an example with The Front Bottoms-Talon of the Hawk-Au Revoir (Adios)
current.filename = "The Front Bottoms-Talon of the Hawk-Au Revoir (Adios).json"

str.current.filename= str_split(current.filename, "-")
artist= str.current.filename[[1]][1]
album= str.current.filename[[1]][2]
track= str_sub(str.current.filename[[1]][3], start=1, end=-6)

json.data=fromJSON(paste("EssentiaOutput/", current.filename, sep=""))

overall_loudness=json.data$lowlevel$loudness_ebu128$integrated #finds overall loudness from data table
spectral_energy= json.data$lowlevel$spectral_energy
dissonance = json.data$lowlevel$dissonance
pitch_salience = json.data$lowlevel$pitch_salience
bpm = json.data$rhythm$bpm
beat_loudness = json.data$rhythm$beats_loudness$mean
danceability = json.data$rhythm$danceability
tuning_frequency =json.data$tonal$tuning_frequency


full_list <- list.files("EssentiaOutput", pattern = "*.json", full.names = TRUE)

# Function to extract data from each JSON file
extract_data <- function(current_filename) {
  str_current_filename <- str_split(basename(current_filename), "-")[[1]]
  artist <- str_current_filename[1] #extracts artist name
  album <- str_current_filename[2] #extracts album name
  track <- str_sub(str_current_filename[3], start = 1, end = -6) #extracts track name

  json_data <- fromJSON(current_filename)

  tibble( #creates a tibble containing the desired data
    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$bpm,
    beat_loudness = json_data$rhythm$beats_loudness$mean,
    danceability = json_data$rhythm$danceability,
    tuning_frequency = json_data$tonal$tuning_frequency
  )
}

# Apply function to all files and combine results into a data frame
df <- map_dfr(full_list, extract_data)

data <- read_csv("EssentiaOutput/EssentiaModelOutput.csv")


data <- data %>%
  mutate( #uses mutate to make desired changes to the data frame called data
    valence = ave(deam_valence, emo_valence, muse_valence, FUN = mean),
    arousal = ave(deam_arousal, emo_arousal, muse_arousal, FUN = mean),
    aggressive = ave(nn_aggressive, eff_aggressive, FUN = mean),
    happy = ave(nn_happy, eff_happy, FUN = mean),
    party = ave(nn_party, eff_party, FUN = mean),
    relaxed = ave(nn_relax, eff_relax, FUN = mean),
    sad = ave(nn_sad, eff_sad, FUN = mean),
    acoustic = ave(nn_acoustic, eff_acoustic, FUN = mean),
    electronic = ave(nn_electronic, eff_electronic, FUN = mean),
    instrumental = ave(nn_instrumental, eff_instrumental, FUN = mean)
  )

# Rename timbreBright column using dplyr’s rename function
data <- data %>%
  rename(timbreBright = eff_timbre_bright)

# Select relevant columns for the new dataset
```

```r
new_data <- data %>%
  select(artist, album, track, valence, arousal, aggressive, happy,
         party, relaxed, sad, acoustic, electronic,
         instrumental, timbreBright)

# Read lyric data
lyric_data <- read_csv("LIWCOutput/LIWCOutput.csv")

# Merge all data together using left_join
final_merge_data <- df %>%
  left_join(new_data, by = c("artist", "album", "track")) |>
  left_join(lyric_data, by = c("artist", "album", "track"))|>
  rename("funct"="function")

#creates data set without Allentown
final_data_no_allentown <- final_merge_data %>%
  filter(track != "Allentown")
#writes the csv file for all tracks other than Allentown
write_csv(final_data_no_allentown, "trainingdata.csv")

#creates data set with just Allentown
final_data_allentown <- final_merge_data %>%
  filter(track == "Allentown")
#writes the csv file for just Allentown
write_csv(final_data_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.