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 0
# install.packages("stringr")
library(stringr)
# install.packages("jsonlite")
library(jsonlite)
# install.packages("tidyverse")
library(tidyverse)

# help(separate)
# help(str_sub)
# help(str_split)
# help(str_remove)
# help(tibble)
# help(list.files)
# Step 1

current.filename = tibble(file.name = "EssentiaOutput/The Front Bottoms-Talon Of The Hawk-Au Revoir (Adios).json")

file = current.filename %>%
  separate(file.name, into = c("path", "artist_album_track"), sep = "/", extra = "merge") %>%
  separate(artist_album_track, into = c("artist", "album", "track"), sep = "-", extra = "merge") %>%
  mutate(track = str_sub(track, 1, nchar(track) - 5)) %>%
  select(artist, album, track)

print(file)


## # A tibble: 1 x 3
##   artist            album             track
##   <chr>             <chr>             <chr>
## 1 The Front Bottoms Talon Of The Hawk Au Revoir (Adios)


file.data = fromJSON(current.filename$file.name[1])
data = tibble(
  overall_loudness = file.data$lowlevel$loudness_ebu128$integrated,
  spectral_energy = file.data$lowlevel$spectral_energy,
  dissonance = file.data$lowlevel$dissonance,
  pitch_salience = file.data$lowlevel$pitch_salience,
  bpm = file.data$rhythm$bpm,
  beats_loudness = file.data$rhythm$beats_loudness,
  danceability = file.data$rhythm$danceability,
  tuning_frequency = file.data$tonal$tuning_frequency
)
print(data)


## # A tibble: 1 x 8
##   overall_loudness spectral_energy dissonance   pitch_salience   bpm
##              <dbl> <named list>    <named list> <named list>   <dbl>
## 1            -10.7 <dbl [1]>       <dbl [1]>    <dbl [1]>       141.
## # i 3 more variables: beats_loudness <named list>, danceability <dbl>,
## #   tuning_frequency <dbl>


# Step 2

json.files = list.files("EssentiaOutput", pattern = ".json", full.names = TRUE)
frame2 = tibble()

for (song in json.files){
  remove.slash = str_split(song, "/")
  filename = remove.slash[length(remove.slash)]

  remove = str_split(filename, "-")
  artist = str_split(remove[length(remove) -2], "/")
  artist = as.character(artist[length(artist)])
  album = remove[length(remove) - 1]
  album = as.character(album[length(album)])
  track = str_sub(remove[length(remove)], 1, nchar(remove[length(remove)]) - 5)

  file.data2 = fromJSON(song)
  row = tibble(
    artist = artist,
    album = album,
    track = track,
```

```
    overall_loudness = file.data2$lowlevel$loudness_ebu128$integrated,
    spectral_energy = file.data2$lowlevel$spectral_energy$mean,
    dissonance = file.data2$lowlevel$dissonance$mean,
    pitch_salience = file.data2$lowlevel$pitch_salience$mean,
    bpm = file.data2$rhythm$bpm,
    beats_loudness = file.data2$rhythm$beats_loudness$mean,
    danceability = file.data2$rhythm$danceability,
    tuning_frequency = file.data2$tonal$tuning_frequency
  )
  frame2 = bind_rows(frame2, row)
}


# Step 3

csv = read.csv("EssentiaOutput/EssentiaModelOutput.csv") %>%
  mutate(
    v_sum = (deam_valence + emo_valence + muse_valence) / 3,
    a_sum = (deam_arousal + emo_arousal + muse_arousal) / 3,

    aggressive = (eff_aggressive + nn_aggressive) / 2,
    happy = (eff_happy + nn_happy) / 2,
    party = (eff_party + nn_party) / 2,
    relaxed = (eff_relax + nn_relax) / 2,
    sad = (eff_sad + nn_sad) / 2,

    acoustic = (eff_acoustic + nn_acoustic) / 2,
    electric = (eff_electronic + nn_electronic) / 2,

    instrumental = (eff_instrumental + nn_instrumental) / 2
  ) %>%
  rename(timbreBright = eff_timbre_bright) %>%
  select(artist, album, track, timbreBright, v_sum,
         a_sum, aggressive, happy, party, relaxed,
         sad, acoustic, electric, instrumental)


# Step 4

liw = read.csv("LIWCOutput/LIWCOutput.csv")
merge.data = csv %>%
  left_join(liw, by = c("artist", "album", "track")) %>%
  left_join(frame2, by = c("artist", "album", "track")) %>%
  rename("funct" = "function.")

trainingdata = merge.data %>% filter(track != "Allentown")
write_csv(trainingdata, "trainingdata.csv")

testingdata = merge.data %>% filter(track == "Allentown")
write_csv(trainingdata, "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.