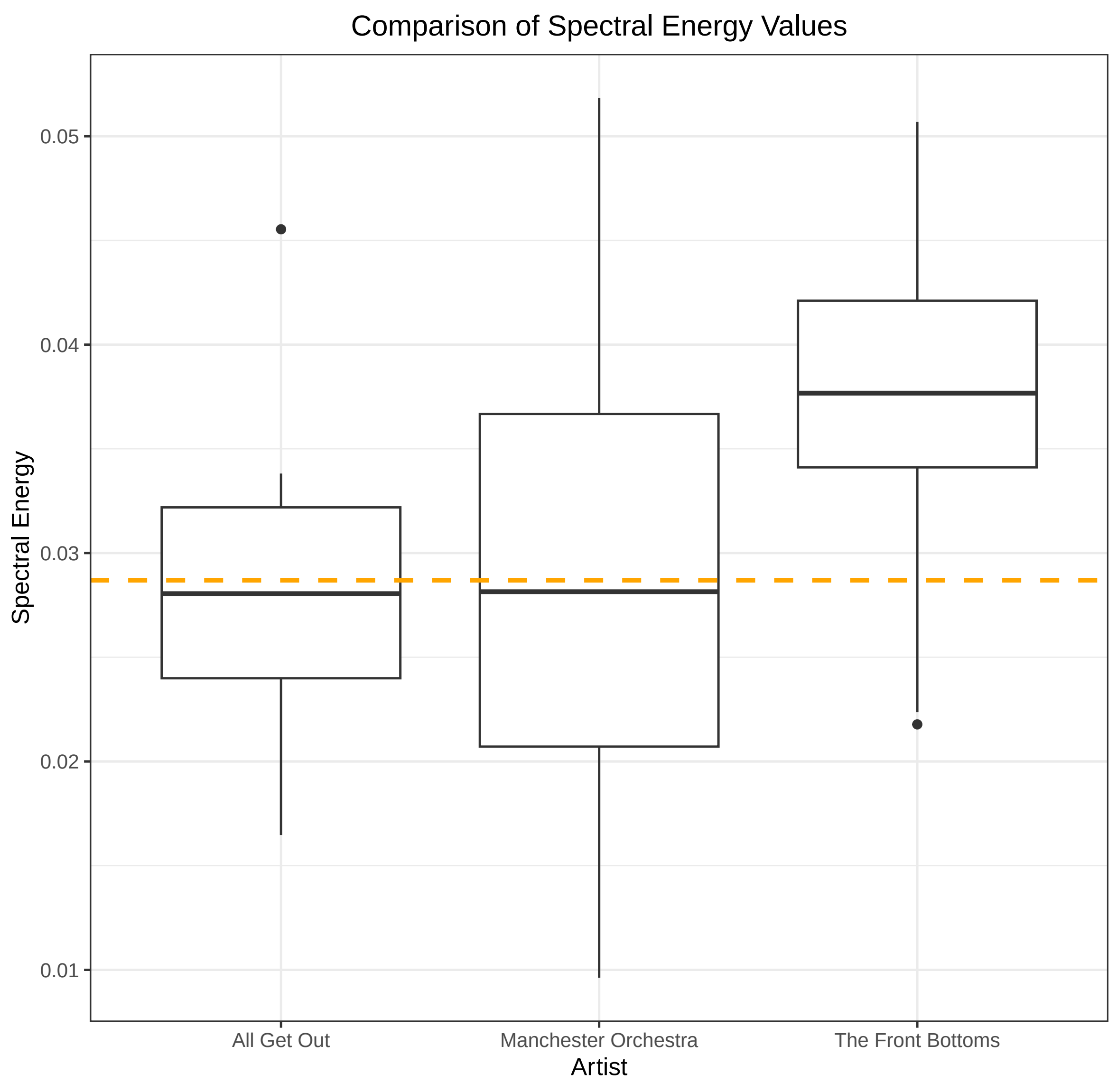
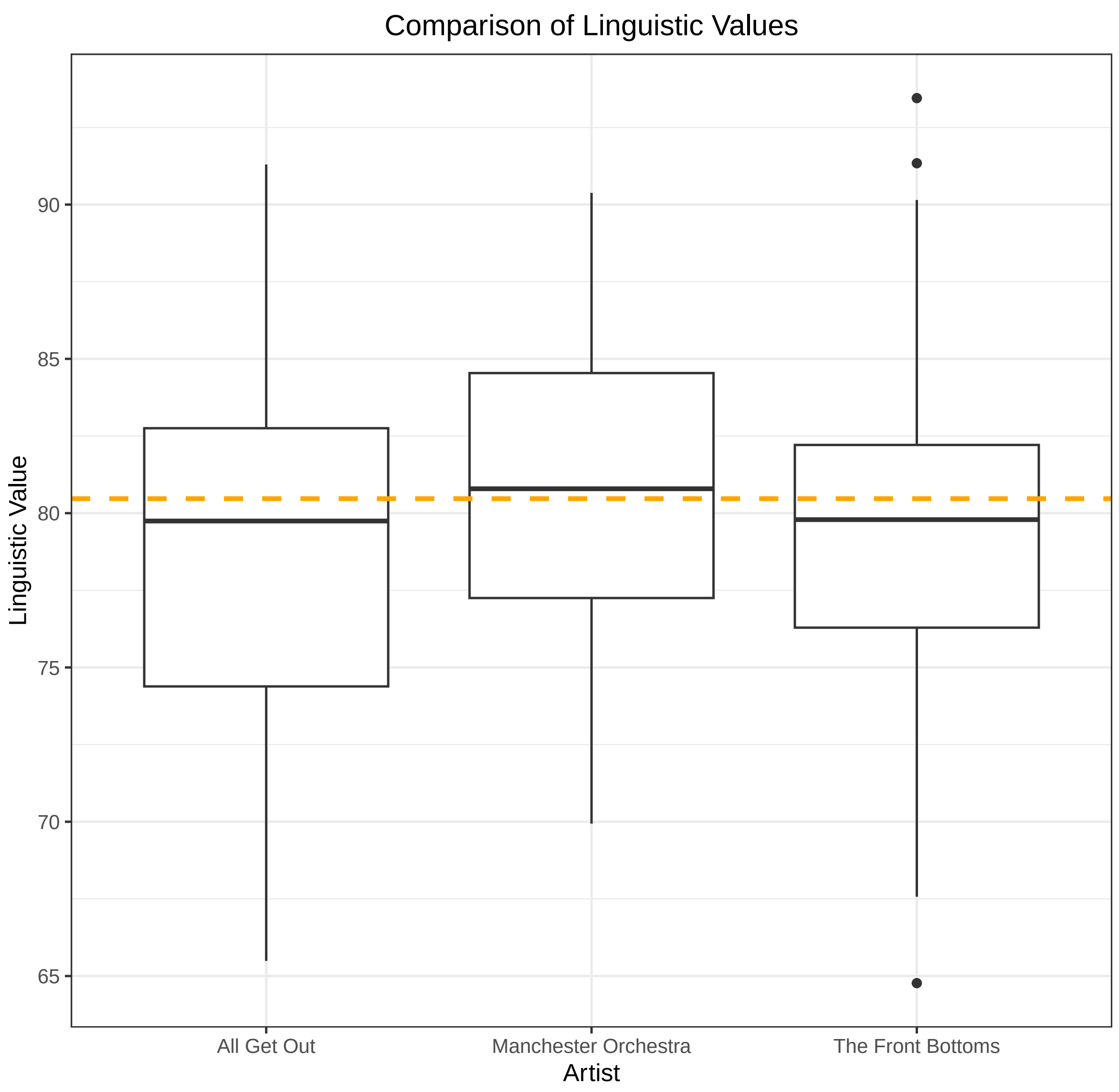
1. In Lab 3, you wrangled data from Essentia, Essentia models and LIWC. Rework your solution to Lab 3 using `tidyverse` (Wickham et al., 2019) instead of base `R`. Specifically, rewrite your code for steps 1-4 of task 2 using `tidyverse` (Wickham et al., 2019). Make sure to address any issues I noted in your code file, and ensure that your code runs in the directory as it is set up.

```r
library(stringr)
library(jsonlite)
library(tidyverse)

filenames <- list.files("EssentiaOutput")
json.indices <- which(str_count(filenames, pattern = ".json")==1)
json.files <- filenames[json.indices]
len <- length(json.files)

df <- tibble(
  artist = character(len),
  album = character(len),
  track = character(len),
  `overall loudness` = numeric(len),
  `spectral energy` = numeric(len),
  dissonance = numeric(len),
  `pitch salience` = numeric(len),
  tempo = numeric(len),
  `beat loudness` = numeric(len),
  danceability = numeric(len),
  `tuning frequency` = numeric(len))

for(i in 1:length(json.files)){
  current.filename <- json.files[i]
  track.info <- str_split(str_sub(current.filename, end = -6), "-", simplify = T)
  track.path <- paste("EssentiaOutput/",current.filename, sep = "") # track file path
  loaded.json <- fromJSON(track.path)

  df[i, ] <- tibble(
    artist = track.info[1],
    album = track.info[2],
    track = track.info[3],
    overall_loudness = loaded.json$lowlevel$loudness_ebu128$integrated,
    spectral_energy = unlist(loaded.json$lowlevel$spectral_energy),
    dissonance = unlist(loaded.json$lowlevel$dissonance),
    pitch_salience = unlist(loaded.json$lowlevel$pitch_salience),
    tempo = loaded.json$rhythm$bpm,
    beat_loudness = unlist(loaded.json$rhythm$beats_loudness),
    danceability = unlist(loaded.json$rhythm$danceability),
    tuning_frequency = unlist(loaded.json$tonal$tuning_frequency))
}
# view(df)

essentia.model <- read_csv("EssentiaOutput/EssentiaModelOutput.csv") %>%
  mutate(
    valence = rowMeans(select(., deam_valence, emo_valence, muse_valence), na.rm = T),
    arousal = rowMeans(select(., deam_arousal, emo_arousal, muse_arousal), na.rm = T),
    aggressive = rowMeans(select(., eff_aggressive, nn_aggressive), na.rm = T),
    happy = rowMeans(select(., eff_happy, nn_happy), na.rm = T),
    party = rowMeans(select(., eff_party, nn_party), na.rm = T),
    relaxed = rowMeans(select(., eff_relax, nn_relax), na.rm = T),
    sad = rowMeans(select(., eff_sad, nn_sad), na.rm = T),
    acoustic = rowMeans(select(., eff_acoustic, nn_acoustic), na.rm = T),
    electric = rowMeans(select(., eff_electronic, nn_electronic), na.rm = T),
    instrumental = rowMeans(select(., eff_instrumental, nn_instrumental), na.rm = T),
    timbreBright = eff_timbre_bright) |>
  select(artist, album, track, valence, arousal, aggressive, happy, party,
         relaxed, sad, acoustic, electric, instrumental, timbreBright)
# view(essentia.model)

liwc.output <- read_csv("LIWCOutput/LIWCOutput.csv")

merged.output <- df |>
  left_join(essentia.model, by = c("artist","album","track")) |>
  left_join(liwc.output, by = c("artist","album","track")) |>
  rename(funct = "function")
# view(merged.output)

training.data <- filter(merged.output, track != "Allentown")
write_csv(training.data, "trainingdata.csv")
testing.data <- filter(merged.output, track == "Allentown")
write_csv(testing.data, "testingdata.csv")
# view(training.data)
# view(testing.data)
# Spectral energy plot
ggplot(data = training.data) +
  geom_boxplot(aes(x = artist, y = `spectral energy`)) +
  geom_hline(yintercept = testing.data$`spectral energy`,
             color = "orange", linetype = "dashed", size = 1) +
  theme_bw() +
  xlab("Artist") +
  ylab("Spectral Energy") +
  labs(title = "Comparison of Spectral Energy Values",
       caption = "Box plot of spectral energy between artists. Spectral energy of 'Allentown' in dashed orange.") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5))
```



Box plot of spectral energy between artists. Spectral energy of 'Allentown' in dashed orange.

```r
# Linguistic plot
ggplot(data = training.data) +
  geom_boxplot(aes(x = artist, y = Linguistic)) +
  geom_hline(yintercept = testing.data$Linguistic,
             color = "orange", linetype = "dashed", size = 1) +
  theme_bw() +
  xlab("Artist") +
  ylab("Linguistic Value") +
  labs(title = "Comparison of Linguistic Values",
       caption = "Box plot of linguistic value between artists. Linguistic value of 'Allentown' in dashed orange.") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5))
```



Box plot of linguistic value between artists. Linguistic value of 'Allentown' in dashed orange.

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.