

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.

```
# Code to run here.
#####
#Homework Number 5
#Fixed Issue where code would not run by replacing Json_file with Json_file_Adios
#####

##### Step 2
#changes#
#Added in the tidyverse library
#Removed library("stringr") since it is included in "tidyverse"
library("jsonlite")
library("tidyverse")

##### Part 1
#changes#
#No change necessary, this step is just assigning a variable
Json_file_Adios <- "The Front Bottoms-Talon Of The Hawk-Au Revoir (Adios).json"

##### Part 2
#changes#
#Took out Pastes since they were unnecessary
#Since this is using Stringr already, which is part of tidyverse, this step is already being done with tidyverse
split_Json_file_Adios <- str_split(Json_file_Adios, "-", simplify = TRUE)
song_name_with_json_Adios = (split_Json_file_Adios[3])
track_Adios <- str_sub(song_name_with_json_Adios, 1, str_length(song_name_with_json_Adios) - 5)
artist_Adios = (split_Json_file_Adios[1])
album_Adios = (split_Json_file_Adios[2])

##### Part 3
#changes#
#No change necessary, this step is just assigning a variable
JSON_Data <- fromJSON(paste("EssentiaOutput", "/", Json_file_Adios, sep = ""))

##### Part 4
#changes#
#Used the pluck() function and introduced pipes
overall_loudness <- JSON_Data |>
  pluck("lowlevel", "loudness_ebu128", "integrated")
spectral_energy <- JSON_Data |>
  pluck("lowlevel", "spectral_energy")
dissonance <- JSON_Data |>
  pluck("lowlevel", "dissonance")
danceability <- JSON_Data |>
  pluck("rhythm", "danceability")
pitch_salience <- JSON_Data |>
  pluck("lowlevel", "pitch_salience")
bpm <- JSON_Data |>
  pluck("rhythm", "bpm")
beats_loudness <- JSON_Data |>
  pluck("rhythm", "beats_loudness")
tuning_frequency <- JSON_Data |>
  pluck("tonal", "tuning_frequency")

##### Step 2
# artist album track
#changes#
#Changed my data frame to a tibble
df_notplot <- tibble(
  artist = character(), album = character(), track = character(),
  overall_loudness = numeric(), spectral_energy = numeric(),
  dissonance = numeric(), danceability = numeric(),
  pitch_salience = numeric(), bpm = numeric(),
  beats_loudness = numeric(), tuning_frequency = numeric()
)
# Create a data frame to save the results
essentia_files <- list.files(path = "EssentiaOutput/")
json_files_new <- essentia_files[str_count(essentia_files, ".json")==1]
#changes#
#Took out pastes that were not necessary
#This step is already using tidyverse since it is using stringr which is a part of tidyverse
for (i in 1:length(json_files_new)) {
  current_filename = json_files_new[i]
  split_Json_file <- str_split(current_filename, "-", simplify = TRUE)
  song_name_with_json = (split_Json_file[3])
}
```

```

track <- str_sub(song_name_with.json, 1, str_length(song_name_with.json) - 5)
artist = (split_json_file[1])
album = (split_json_file[2])
# remove .json from the song
# save artist/album/track

# Read in the file as JSON_DataVie JSON_Data_New <- fromJSON(paste("json_files_new", "/", i, sep = ""))
JSON_Data_New <- fromJSON(paste0("EssentiaOutput/", current_filename))
#changes#
#Used the pluck() function and introduced pipes
overall_loudness <- JSON_Data_New |>
  pluck("lowlevel", "loudness_ebul28", "integrated")
spectral_energy <- JSON_Data_New |>
  pluck("lowlevel", "spectral_energy")
dissonance <- JSON_Data_New |>
  pluck("lowlevel", "dissonance")
danceability <- JSON_Data_New |>
  pluck("rhythm", "danceability")
pitch_salience <- JSON_Data_New |>
  pluck("lowlevel", "pitch_salience")
bpm <- JSON_Data_New |>
  pluck("rhythm", "bpm")
beats_loudness <- JSON_Data_New |>
  pluck("rhythm", "beats_loudness")
tuning_frequency <- JSON_Data_New |>
  pluck("tonal", "tuning_frequency")

# Save the results to the ith row of the data frame
#changes#
#Changed data frame to a tibble and used bind_rows() instead of rbind()
#Removed column renaming since doing the dataframe this way solved the
#issue which caused me to have to rename some columns in the first place
df_notplot <- df_notplot %>%
  bind_rows(tibble(
    artist = artist,
    album = album,
    track = track,
    overall_loudness = as.numeric(overall_loudness),
    spectral_energy = as.numeric(spectral_energy),
    dissonance = as.numeric(dissonance),
    danceability = as.numeric(danceability),
    pitch_salience = as.numeric(pitch_salience),
    bpm = as.numeric(bpm),
    beats_loudness = as.numeric(beats_loudness),
    tuning_frequency = as.numeric(tuning_frequency)
  ))
}

##### Step 3 Part 1
#changes#
#changed read.csv() to read_csv()
Essentia_Output_Original <- read_csv("EssentiaOutput/EssentiaModelOutput.csv")

##### Step 3 Part 2
#changes#
#used mutate to create new columns
Essentia_Output_Original |>
  mutate(arousal = (emo_arousal + deam_arousal + muse_arousal) / 3) |>
  mutate(valence = (emo_valence + deam_valence + muse_valence) / 3) |>

##### Step 3 Part 3
#changes#
#used mutate to create new columns
mutate(aggressive = (nn_aggressive + eff_aggressive) / 2) |>
mutate(happy = (nn_happy + eff_happy) / 2) |>
mutate(party = (nn_party + eff_party) / 2) |>
mutate(relaxed = (nn_relax + eff_relax) / 2) |>
mutate(sad = (nn_sad + eff_sad) / 2) |>

##### Step 3 Part 4
#changes#
#used mutate to create new columns
mutate(acoustic = (nn_acoustic + eff_acoustic) / 2) |>
mutate(electric = (nn_electronic + eff_electronic) / 2) |>

##### Step 3 Part 5
#changes#

```

```

#used mutate to create new columns
mutate(instrumental =(nn_instrumental + eff_instrumental)/ 2) |>

##### Step 3 Part 6
#changes#
#used rename in the pipe to rename the column
rename(timbreBright = eff_timbre_bright )

##### Step 3 Part 7
#changes#
#Used select() function and pipe
Essentia_Output_New <- Essentia_Output_Original |>
  select(artist, album, track, arousal, valence, aggressive, happy, party, relaxed,
    sad, acoustic, electric, instrumental, timbreBright)

##### Step 4 Part 1
#changes#
#changed read.csv() to read_csv()
LIWC_Output <- read_csv("LIWCOutput/LIWCOutput.csv")

##### Step 4 Part 2
#changes#
#Used full_join() instead of merge() and introduced pipes
Merged_df_1 <- df_notplot |>
  full_join(Essentia_Output_New, by = c("artist", "album", "track"))
Final_Merged_df <- Merged_df_1 |>
  full_join(LIWC_Output, by = c("artist", "album", "track")) |>
##### Step 4 Part 3
#changes#
#Used rename() function and ran it through a pipe
rename(funct = 'function')

```

References

- Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between json data and r objects. *arXiv:1403.2805 [stat.CO]*.
- 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.