Music Insights

Reece Resendez

First, we need tidyverse to handle the data importing and wrangling Since we already have it installed on this system we can go ahead and implement it.

```r
suppressMessages(library(tidyverse))
suppressMessages(library(readr))

suppressMessages(survey <- read_csv("https://raw.githubusercontent.com/introdsci/MusicSurvey/master/mus

suppressMessages(preferences <- read_csv("https://raw.githubusercontent.com/introdsci/MusicSurvey/master
```

Now we want to go ahead and update all the column names for each of the dataframes to make referencing and appearance look better.

```r
colnames(survey)[colnames(survey) == "First, we are going to create a pseudonym for you to keep this su

colnames(survey)[colnames(survey) == "What is your pseudonym?"  ] <- "pseudonym"

colnames(survey)[colnames(survey) == "Sex"] <- "sex"

colnames(survey)[colnames(survey) == "Major"] <- "academic_major"

colnames(survey)[colnames(survey) == "Which musical instruments/talents do you play? (Select all that a

colnames(survey)[colnames(survey) == "Year you were born (YYYY)"  ] <- "year_born"

colnames(survey)[colnames(survey) == "Academic Year"] <- "academic_level"

colnames(survey)[colnames(survey) == "YOB"] <- "year_born"

colnames(survey)[colnames(survey) == "Artist"] <- "favorite_song_artist"

colnames(survey)[colnames(survey) == "Song"] <- "favorite_song"

colnames(survey)[colnames(survey) == "Link to song (on Youtube or Vimeo)"] <- "favorite_song_link"

colnames(survey)[colnames(survey) == "Timestamp"] <- "time_submitted"


colnames(survey)
```

```
##  [1] "time_submitted"      "pseudonym_generator"  "pseudonym"
##  [4] "sex"                 "academic_major"       "academic_level"
##  [7] "year_born"           "instrument_list"      "favorite_song_artist"
## [10] "favorite_song"       "favorite_song_link"
```

We want to make the code tidy so we are going to use two packages that come with tidyverse but need to be loaded in to work.

```r
suppressMessages(library(dplyr))
suppressMessages(library(tidyr))
```

Now we are going to move into creating seperate tables from these two dataframes we have. We will do this by using dplyr and tibbles. We are also going to manipulate the data for a few variables to standardize them and create proper levels for categorical variables.

```r
Person <- tibble(time_submitted = survey$time_submitted,
                 pseudonym_generator = survey$pseudonym_generator,
                 pseudonym = survey$pseudonym,
                 sex = survey$sex,
                 year_born = survey$year_born,
                 academic_level = survey$academic_level,
                 academic_major = survey$academic_major,
                 favorite_instrument = survey$instrument_list)

Person$academic_level <- as.factor(Person$academic_level)
Person$academic_major <- as.factor(Person$academic_major)

#Standardize spelling of variables for major
levels(Person$academic_major)[levels(Person$academic_major) == "Computer information systems"] <- "Compu

colnames(Person)
```

```
## [1] "time_submitted"      "pseudonym_generator" "pseudonym"
## [4] "sex"                 "year_born"           "academic_level"
## [7] "academic_major"      "favorite_instrument"
```

```r
#----------------------------------------------------------------

FavoriteSong <- tibble( pseudonym = survey$pseudonym,
                        song_artist = survey$favorite_song_artist,
                        favorite_song = survey$favorite_song,
                        song_link = survey$favorite_song_link)

colnames(FavoriteSong)
```

```
## [1] "pseudonym"     "song_artist"   "favorite_song" "song_link"
```

```r
#----------------------------------------------------------------
#This part was tricky. I manipulated the original dataframe preferences by using the gather function fr

preferences <- preferences %>% gather(song_to_rate, rating, 3:45)

#using this new manipulated preferences dataframe I can easily map the columns to my ratings tibble

Ratings <- tibble(pseudonym = preferences$`What was your pseudonym?`,
                  song_to_rate = preferences$song_to_rate,
                  ratings = preferences$rating)

Ratings$song_to_rate <- as.factor(Ratings$song_to_rate)
```

Time to manipulate the timestamps column in Person. We will be using a couple functions to format this column properly.

```
Person$time_submitted<-as.POSIXlt(parse_datetime(Person$time_submitted, format = "%m/%d/%y %H:%M"))
```