# IMT 573: Module 3 Lab

Advanced Visualization

Jenny Skytta

Due: April 18, 2022

**Collaborators:** *Independent work*

### Objectives

As we continue our data science journey, we are gaining skills in working with data. This might be reflected in more efficient ways to manipulate and summarize data, both of which can be useful for creating more advanced visualizations of that data. To accomplish many of the visualization tasks in these exercises you will need to make use of newly acquired data manipulation skills!

### Instructions

Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Cloud.

1. Open the `03_lab_advancedviz.Rmd` and save a copy to your local directory. Supply your solutions to the assignment by editing `03_lab_advancedviz.Rmd`.
2. First, replace the "YOUR NAME HERE" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do no need four different visualizations of the same pattern.
4. Collaboration on problem sets is fun and useful, and I encourage it, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit`. When the PDF report is generated rename the knitted PDF file to `lab3_YourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

### Setup

In this lab you will need, at minimum, the following R packages.

```r
# Load standard libraries
library(tidyverse)
```

```
install.packages("maps")
library(maps)
```

The data we will use in this lab comes from the Million Song Dataset. The Million Song Dataset is a collaboration between the Echo Nest and LabROSA, a laboratory working towards intelligent machine listening. The project was also funded in part by the National Science Foundation of America (NSF) to provide a large data set to evaluate research related to algorithms and information retrieval.

http://millionsongdataset.com/

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

We will use a subset of this data created by Ryan Whitcomb, rwhit94@vt.edu, which contains data on 10,000 songs. The data contains standard information about the songs such as artist name, title, and year released. Additionally, the data contains more advanced information; for example, the length of the song, how many musical bars long the song is, and how long the fade in to the song was.

```
# Load music data
music_data <- read_csv("data/music.csv")
```

### Problem 1: Inspection

First, inspect the data. You can use functions such as glimpse, head, tail, etc. to help you get a sense of what is contained in the data.

```
glimpse(music_data) #condensed view of dataframe
```

```
## Rows: 10,000
## Columns: 35
## $ artist.familiarity           <dbl> 0.58179377, 0.63063004, 0.48735679, 0.6~
## $ artist.hotttnesss            <dbl> 0.4019975, 0.4174996, 0.3434284, 0.4542~
## $ artist.id                    <chr> "ARD7TVE1187B99BFB1", "ARMJAGH1187FB546~
## $ artist.latitude              <dbl> 0.00000, 35.14968, 0.00000, 0.00000, 0.~
## $ artist.location              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ artist.longitude             <dbl> 0.00000, -90.04892, 0.00000, 0.00000, 0~
## $ artist.name                  <chr> "Casual", "The Box Tops", "Sonora Santa~
## $ artist.similar               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ artist.terms                 <chr> "hip hop", "blue-eyed soul", "salsa", "~
## $ artist.terms_freq            <dbl> 1.0000000, 1.0000000, 1.0000000, 0.9885~
## $ release.id                   <dbl> 300848, 300822, 514953, 287650, 611336,~
## $ release.name                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ song.artist_mbtags           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ song.artist_mbtags_count     <dbl> 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ song.bars_confidence         <dbl> 0.643, 0.007, 0.980, 0.017, 0.175, 0.12~
## $ song.bars_start              <dbl> 0.58521, 0.71054, 0.73152, 1.30621, 1.0~
## $ song.beats_confidence        <dbl> 0.834, 1.000, 0.980, 0.809, 0.883, 0.43~
## $ song.beats_start             <dbl> 0.58521, 0.20627, 0.73152, 0.81002, 0.1~
## $ song.duration                <dbl> 218.9318, 148.0355, 177.4755, 233.4036,~
## $ song.end_of_fade_in          <dbl> 0.247, 0.148, 0.282, 0.000, 0.066, 2.26~
## $ song.hotttnesss              <dbl> 0.6021200, -1.0000000, -1.0000000, -1.0~
## $ song.id                      <chr> "SOMZWCG12A8C13C480", "SOCIWDW12A8C13D4~
## $ song.key                       <dbl> 1, 6, 8, 0, 2, 5, 1, 4, 4, 7, 5, 7, 9, ~
## $ song.key_confidence            <dbl> 0.736, 0.169, 0.643, 0.751, 0.092, 0.63~
## $ song.loudness                  <dbl> -11.197, -9.843, -9.689, -9.013, -4.501~
## $ song.mode                      <dbl> 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, ~
## $ song.mode_confidence           <dbl> 0.636, 0.430, 0.565, 0.749, 0.371, 0.55~
## $ song.start_of_fade_out         <dbl> 218.932, 137.915, 172.304, 217.124, 198~
## $ song.tatums_confidence         <dbl> 0.779, 0.969, 0.482, 0.601, 1.000, 0.13~
## $ song.tatums_start              <dbl> 0.28519, 0.20627, 0.42132, 0.56254, 0.1~
## $ song.tempo                     <dbl> 92.198, 121.274, 100.070, 119.293, 129.~
## $ song.time_signature            <dbl> 4, 4, 1, 4, 4, 3, 1, 3, 4, 4, 1, 4, 4, ~
## $ song.time_signature_confidence <dbl> 0.778, 0.384, 0.000, 0.000, 0.562, 0.45~
## $ song.title                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ song.year                      <dbl> 0, 1969, 0, 1982, 2007, 0, 0, 0, 1984, ~
```

```
head(music_data) # first few observations of dataframe
```

```
## # A tibble: 6 x 35
##   artist.familiari~ artist.hotttnes~ artist.id   artist.latitude artist.location
##               <dbl>            <dbl> <chr>                 <dbl>           <dbl>
## 1             0.582            0.402 ARD7TVE118~             0                 0
## 2             0.631            0.417 ARMJAGH118~            35.1               0
## 3             0.487            0.343 ARKRRTF118~             0                 0
## 4             0.630            0.454 AR7G5I4118~             0                 0
## 5             0.651            0.402 ARXR32B118~             0                 0
## 6             0.535            0.385 ARKFYS9118~             0                 0
## # ... with 30 more variables: artist.longitude <dbl>, artist.name <chr>,
## #   artist.similar <dbl>, artist.terms <chr>, artist.terms_freq <dbl>,
## #   release.id <dbl>, release.name <dbl>, song.artist_mbtags <dbl>,
## #   song.artist_mbtags_count <dbl>, song.bars_confidence <dbl>,
## #   song.bars_start <dbl>, song.beats_confidence <dbl>, song.beats_start <dbl>,
## #   song.duration <dbl>, song.end_of_fade_in <dbl>, song.hotttnesss <dbl>,
## #   song.id <chr>, song.key <dbl>, song.key_confidence <dbl>,
## #   song.loudness <dbl>, song.mode <dbl>, song.mode_confidence <dbl>,
## #   song.start_of_fade_out <dbl>, song.tatums_confidence <dbl>,
## #   song.tatums_start <dbl>, song.tempo <dbl>, song.time_signature <dbl>,
## #   song.time_signature_confidence <dbl>, song.title <dbl>, song.year <dbl>
```

```
tail(music_data) #last few observations of dataframe
```

```
## # A tibble: 6 x 35
##   artist.familiari~ artist.hotttnes~ artist.id   artist.latitude artist.location
##               <dbl>            <dbl> <chr>                 <dbl>           <dbl>
## 1             0.607            0.401 ARDK055118~            31.3               0
## 2             0.723            0.500 AR4C6V0118~            39.6               0
## 3             0.512            0.410 AR9JLBU118~           -34.0               0
## 4             0.434            0.290 ARS1DCR118~             0                 0
## 5             0.334            0.217 ARAGMIV11F~             0                 0
## 6             0.609            0.509 ARYXOV8118~             0                 0
## # ... with 30 more variables: artist.longitude <dbl>, artist.name <chr>,
## #   artist.similar <dbl>, artist.terms <chr>, artist.terms_freq <dbl>,
## #   release.id <dbl>, release.name <dbl>, song.artist_mbtags <dbl>,
## #   song.artist_mbtags_count <dbl>, song.bars_confidence <dbl>,
## #   song.bars_start <dbl>, song.beats_confidence <dbl>, song.beats_start <dbl>,
## #   song.duration <dbl>, song.end_of_fade_in <dbl>, song.hotttnesss <dbl>,
## #   song.id <chr>, song.key <dbl>, song.key_confidence <dbl>,
## #   song.loudness <dbl>, song.mode <dbl>, song.mode_confidence <dbl>,
## #   song.start_of_fade_out <dbl>, song.tatums_confidence <dbl>,
## #   song.tatums_start <dbl>, song.tempo <dbl>, song.time_signature <dbl>,
## #   song.time_signature_confidence <dbl>, song.title <dbl>, song.year <dbl>
```

```r
dim(music_data) # dimensions of dataframe
```

```
## [1] 10000    35
```

### Problem 2: Pose a Question

Propose a question to guide your analysis. For example, you might ask if the average hotness scores of songs change over time? Or perhaps, what is the relationship between song duration and tempo? You can use one of these questions or develop your own. State which question you want to answer.

What is song hotttnesss you ask? According to the dataset description, it is a measure of the song's popularity, when downloaded (in December 2010). And measured on a scale of 0 to 1.

## Questions

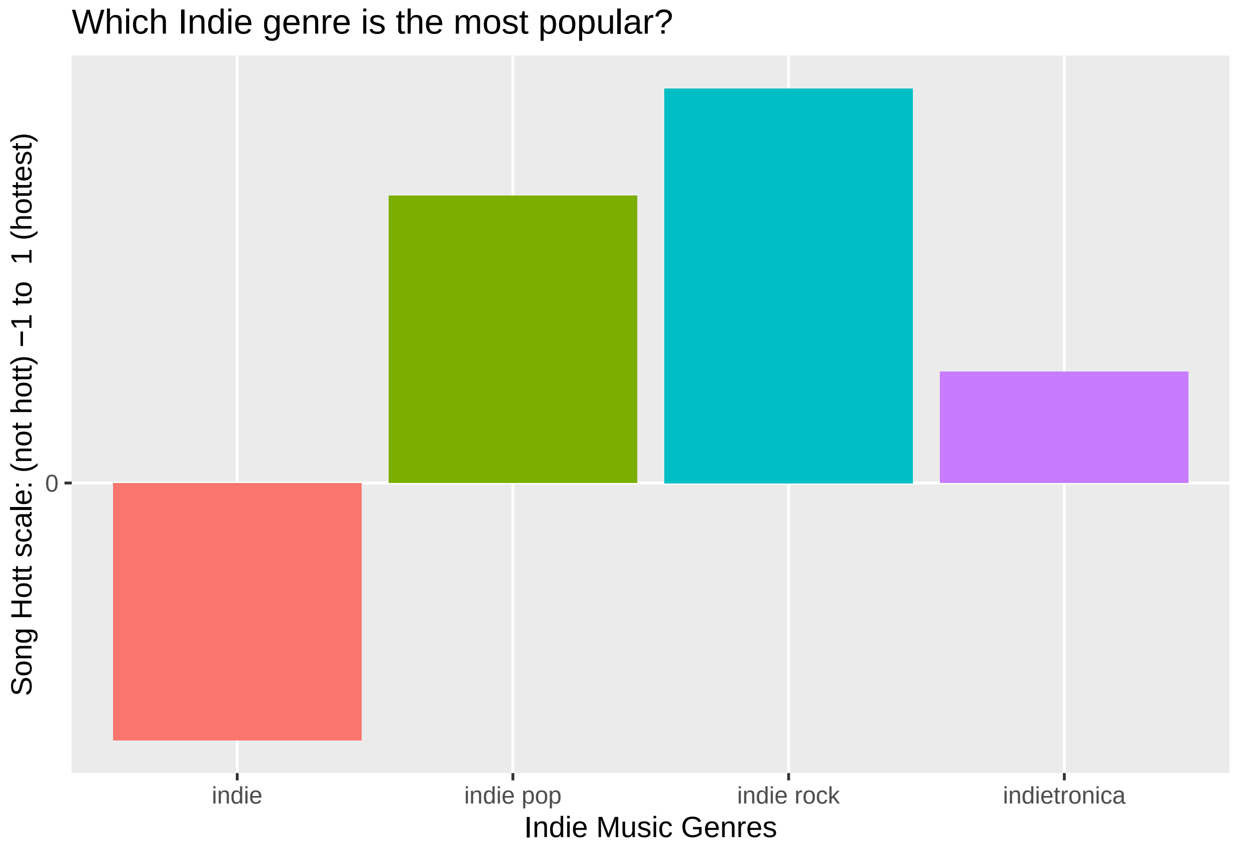
- *Which indie genres are most popular?*
- *Where was the location of the artist's with the oldest recordings?*

```r
#create a tibble
indie_pop_vs_rock <- music_data %>%
  filter(grepl('indie', artist.terms)) %>% # filter artist.terms column for "indie"
  group_by(artist.terms) %>% #group the data by genre
  mutate(arthot = mean(artist.hotttnesss)) %>% #average of artist hotness
  mutate(song = mean(song.hotttnesss)) %>% #average of song hotness
  mutate(hotness = mean(song+arthot)) %>% #average of avereages
  select(artist.terms, hotness) #select columns

indie_plot <- distinct(indie_pop_vs_rock) #pull only distinct columns

# create new tibble
oldest_song_location <- music_data %>%
  group_by(artist.name) %>% # group_by artist
  arrange(desc(-song.year)) %>%  # arrange by descending year
  filter(artist.latitude > 1 & artist.longitude > 1 & song.year > 1) %>% #filter to remove 0 values
  select(artist.name, song.year, artist.latitude, artist.longitude)
         # select artist.latitude, artist.longitude
# head(data, 5) to get the top 5
```

### Problem 3: Visualization

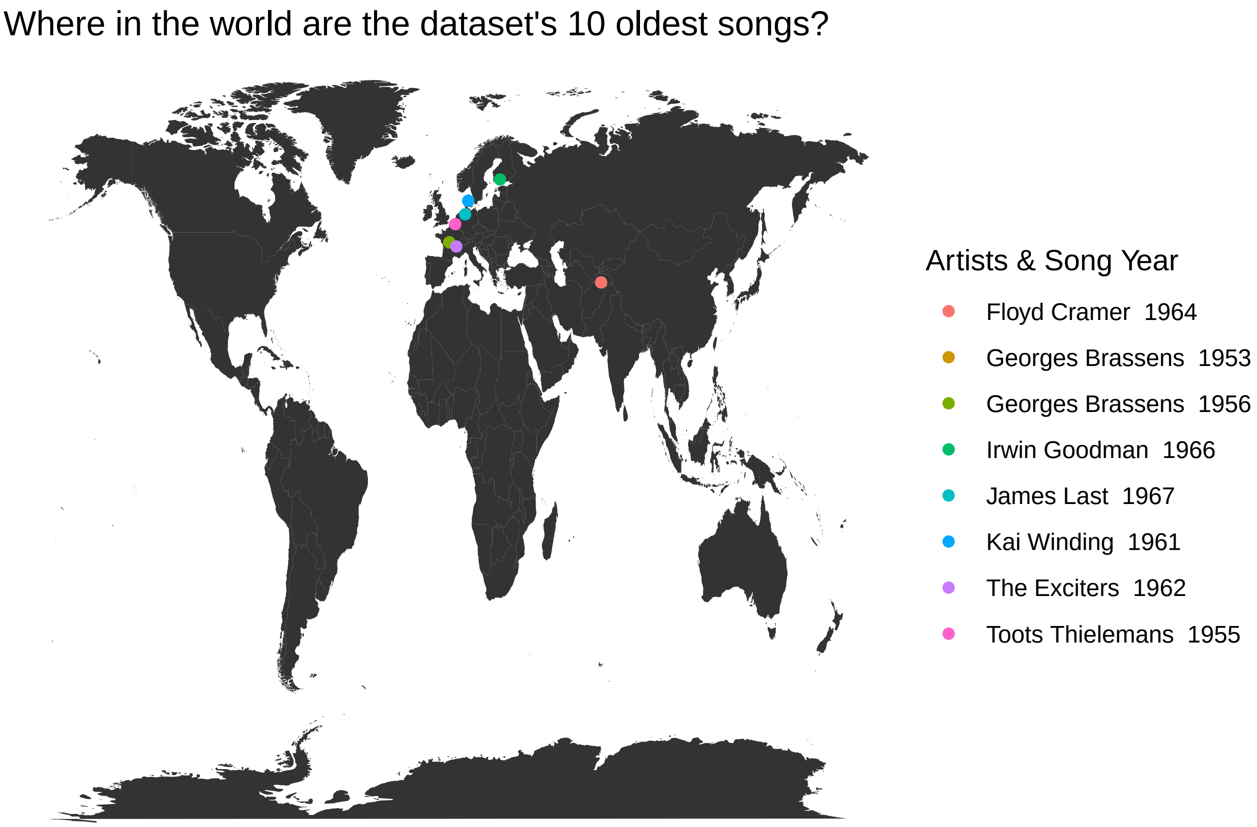
Create two visualizations to help gain insight into your question. Be sure to explain the visuals you create and what you take away from them.

```r
# create a barchart to show Indie genre's measured by
# recurrence of genre across hot measurement within the dataset.
ggplot(data = indie_plot) +
  geom_col(mapping = aes(x = artist.terms, y = hotness, fill = artist.terms),show.legend = FALSE) +
  labs( #add labels and remove legend
    title = "Which Indie genre is the most popular?",
    x = "Indie Music Genres",
    y = "Song Hott scale: (not hott) -1 to  1 (hottest)") +
    scale_y_continuous(breaks = seq(-1,0,1)) #set y scale range
```



```
oldest_song_loc_5 <- head(oldest_song_location, 10) %>%
  unite("Artist_Year", artist.name:song.year, sep= "  ")
#View(oldest_song_loc_5)
world <- map_data("world") #pull in world map for plot

# Visual display of where in the world the top 10
# oldest songs from the list originated.
  ggplot() + #plot on map of globe
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region)) +
  geom_point(data = oldest_song_loc_5,
    aes(artist.longitude, artist.latitude, color = Artist_Year),
    alpha = 2.7) +
  theme_void() +
    labs(
    title = "Where in the world are the dataset's 10 oldest songs?",
    color = "Artists & Song Year"
    )
```

```
## Warning: Ignoring unknown aesthetics: x, y
```



Where in the world are the dataset's 10 oldest songs?

## Visualization Reflection

- *Which indie genres are most popular?* The barchart visualization assesses which of the indie genres are the hottest within the dataset. In reviewing the dataset, I noticed the volume of genres within the dataset was enormous and varied. This is what initially piqued my curiosity about which genre might be the top "hot" genre. I originally looked at that data which could have directed me an entirely different direction. I then was curious about my favorite genre and how it was represented within the data. I didn't want to narrow too far so I left it vague using the term "indie". This worked well as there were only 61 observations. Overall, I'm not suprised that indie rock itself was the most frequent and its hottness scale measurement is also not terribly surprising. Indie itself as a genre ranked pretty low in the sub-hott ranking. Overall, a nice slice of 2010 perceptions.
- *Where was the location of the artist's with the oldest recordings?* This map visual was the best way I could capture the question about location pertaining to the oldest recordings. I was initially curious as to who and what were the oldest recordings in the dataset and after searching, was curious where they orginated. The geographical location for the artists shows that they all were originated from Europe. I had somewhat expected at least a few of these artists to be from the U.S. so that was surprising. This does leave a desire to narrow the map in scale to only show Europe but that could also suggest a focus on only Europe when we are looking at the entire world. Ultimately, though its sparse, I kept the scale to the entire world.

### Citations

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011. http://millionsongdataset.com/

Code written above is from the previous course IMT 511 which used the below text to support class scripts. https://www.google.com/books/edition/Programming_Skills_for_Data_Science/BnB6DwAAQBAJ?hl=en&gbpv=1&printsec=frontcover

GGplot of Maps https://datavizpyr.com/how-to-make-world-map-with-ggplot2-in-r/