IMT 573: Module 3 Lab

Advanced Visualization

Tommy Huynh

Due: July 8, 2021

Collaborators: List collaborators here.

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.

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

```
## Warning: package 'tidyr' was built under R version 4.1.1
```

```
## Warning: package 'readr' was built under R version 4.1.1
```

```
library(ggplot2)
```

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("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.

```
head(music_data)
```

```
## # A tibble: 6 x 35
##   artist.familiarity artist.hotttnesss artist.id artist.latitude artist.location
##                <dbl>             <dbl> <chr>               <dbl>           <dbl>
## 1              0.582             0.402 ARD7TVE1~             0                0
## 2              0.631             0.417 ARMJAGH1~            35.1              0
## 3              0.487             0.343 ARKRRTF1~             0                0
## 4              0.630             0.454 AR7G5I41~             0                0
## 5              0.651             0.402 ARXR32B1~             0                0
## 6              0.535             0.385 ARKFYS91~             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>, ...
```

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.

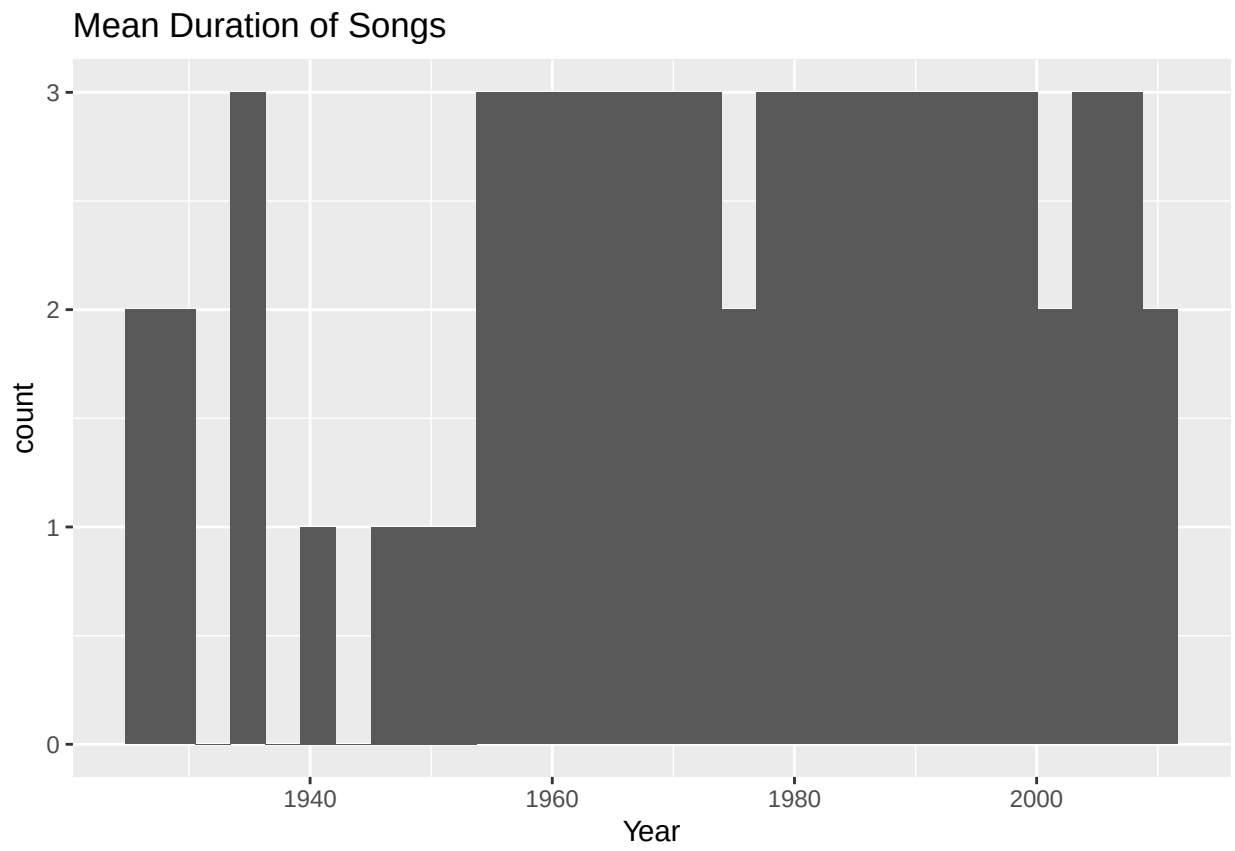
My question is: Have the lengths of songs increased, decreased, or remained constant over time?

Problem 3: Visualization

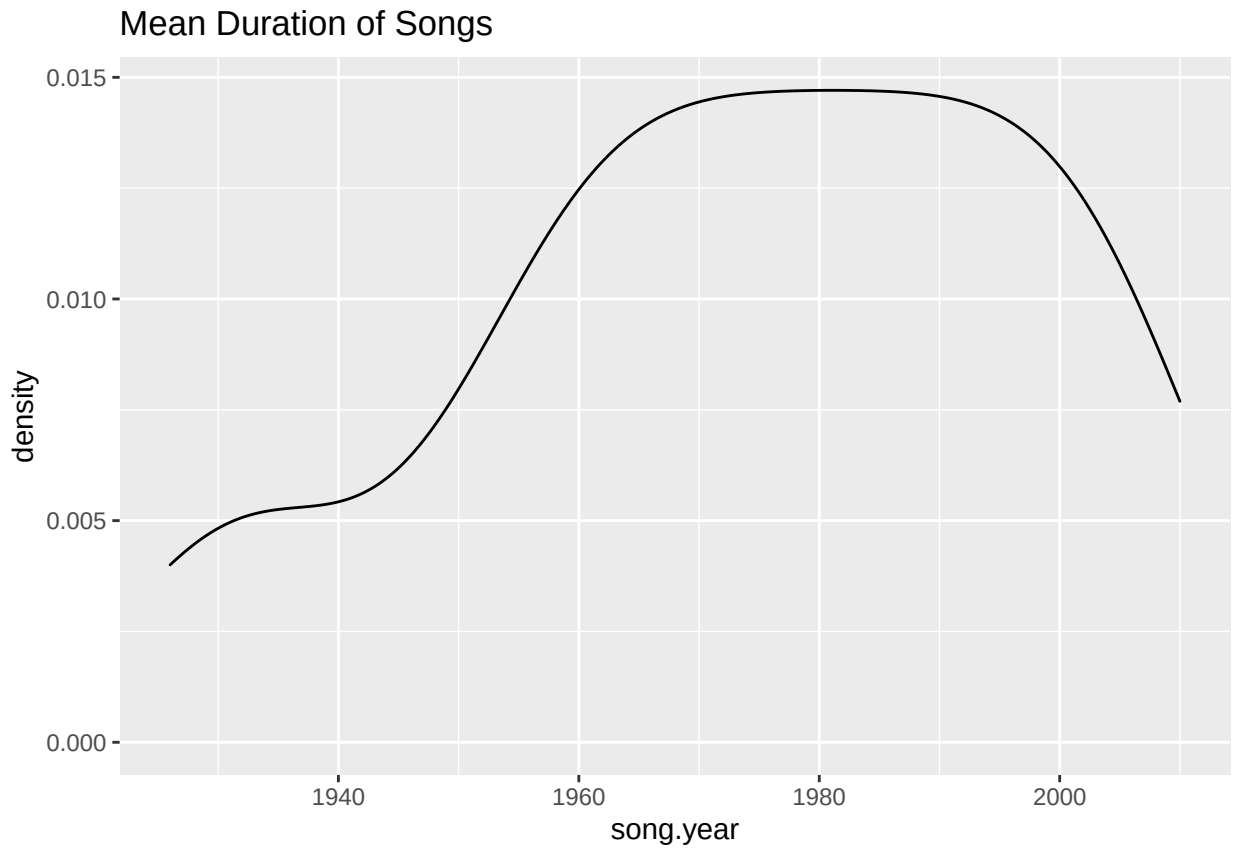
Create two visualizations to help gain insight into your question. Be sure to explain the visuals you create and what you take away from them.

```
#This code subsets the music_data dataframe into a data frame that only has the song year and duration
#It then proceeds to aggregate it into a mean for the year, and from there we can create the visuals
year_length <- select(music_data, song.year, song.duration)
year_length <- year_length[apply(year_length, 1, function(row) all(row !=0 )), ]

year_length <- aggregate(year_length, by = list(year_length$song.year), FUN = mean)
year_length$Group.1 <- NULL

ggplot(year_length, aes(x = song.year)) +
  geom_histogram() +
  labs(title = "Mean Duration of Songs",
       x = "Year")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(year_length, aes(x = song.year)) +
  geom_density() +
  labs(title = "Mean Duration of Songs")
```



My takeaway from the visuals is that the durations of songs stayed relatively the same, but on a microscopic scale they are trending towads decreasing lengths.