DATABASE PROJECT: SPOTIFY ANALYSIS

12/07/2018

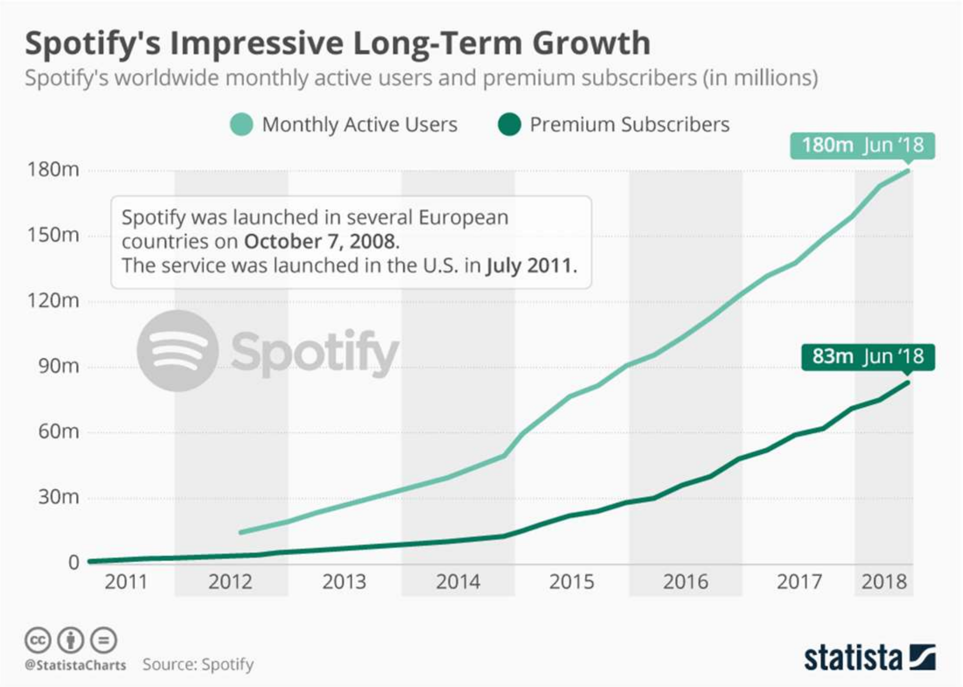
DATA ENGINEERING PLATFORMS
MScA 31012
FRIDAY6_9GROUP1

Team members:
Jorge Argueta | Anh Phan
Weihuang Xie | Jihun Lee

AGENDA

- Executive Summary
- Business Use Cases
- Data Processing
- Data Modeling
- Business Analysis
- Future Work
- Lesson Learned

- Companies today have a large marketing budget, unfortunately some are still trying to figure out how to get the best return on their investment. Data analysis can help us advise entertainment companies to achieve that goal. Spotify has been gathering data in the streaming music industry for the past 10 years across the world.
- Our goal is to analyze streaming data from different parts of the world, to advise artist, organizations or entertainment companies; in a way that they can make informed decisions in their next campaign or investment.
- With data visualization techniques we can simplify interpretation from 190 million subscribers, to better understand what customers are listening around the globe and perhaps understand their sentiment associates with those lyrics.



Spotify's Impressive Long-Term Growth

Spotify's worldwide monthly active users and premium subscribers (in millions)

Spotify was launched in several European countries on **October 7, 2008.**
The service was launched in the U.S. in **July 2011.**

@StatistaCharts Source: Spotify

Source: https://www.statista.com/chart/15697/spotify-user-growth/

Advertisement Pricing

- Rank the streaming of songs and define their popularity
- Spotify can increase price of advertisement when people click on top streaming songs

Musicians Branding

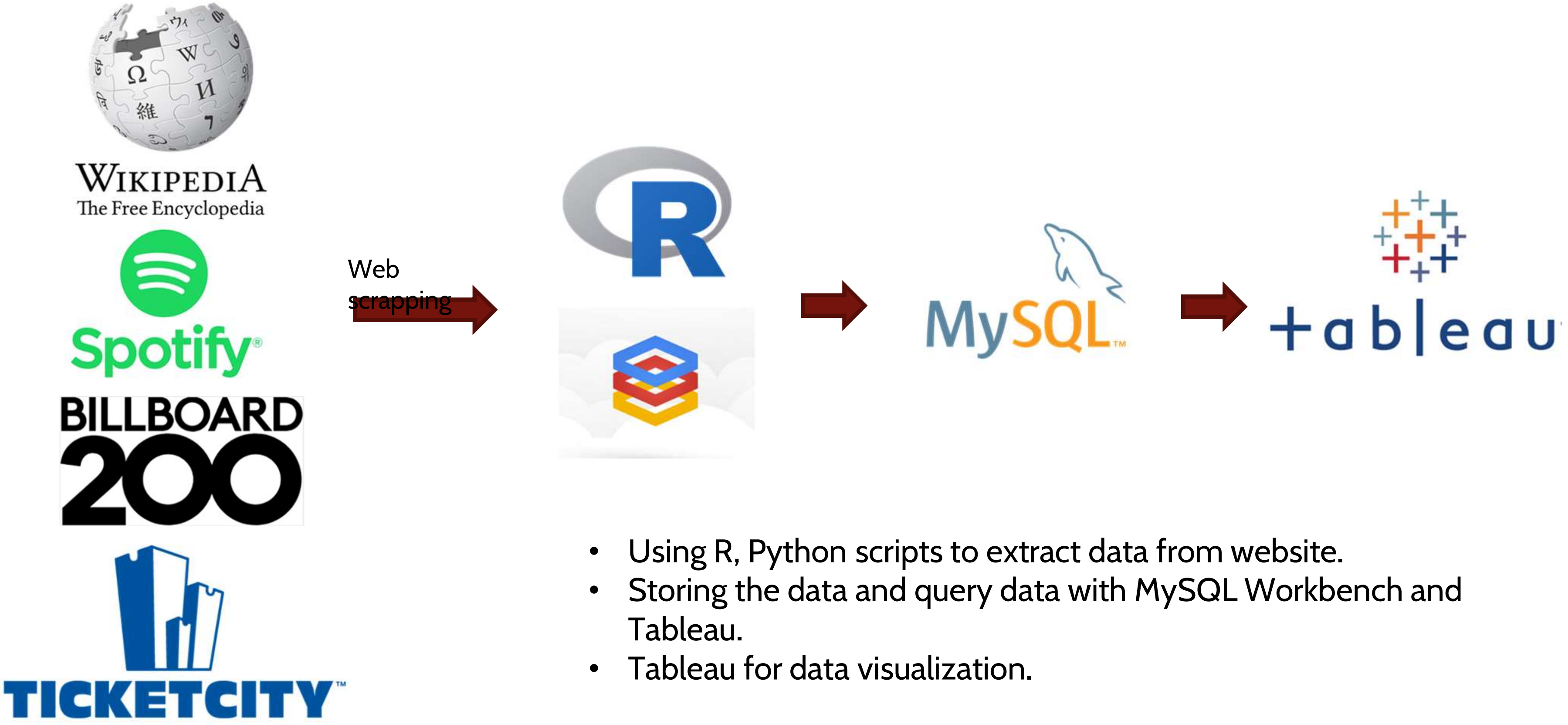
- Analyze popularity of songs and singers across different countries
- Musicians can improve their branding strategy and event schedule in different regions to increase their popularity

Concert Production

- Rank the popularity of singers and link this to their concerts schedule and locations.

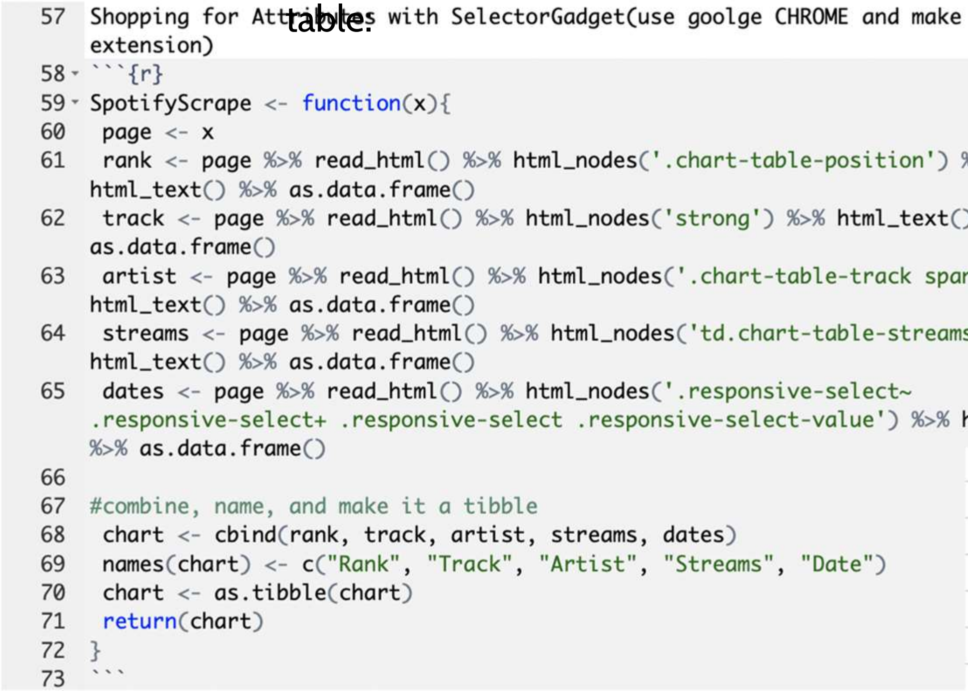
Music Production

- Rank the popularity of songs and analyze the lyrics of them
- Music producers can produce better music base on this analysis to catch market demand



- Using R, Python scripts to extract data from website.
- Storing the data and query data with MySQL Workbench and Tableau.
- Tableau for data visualization.

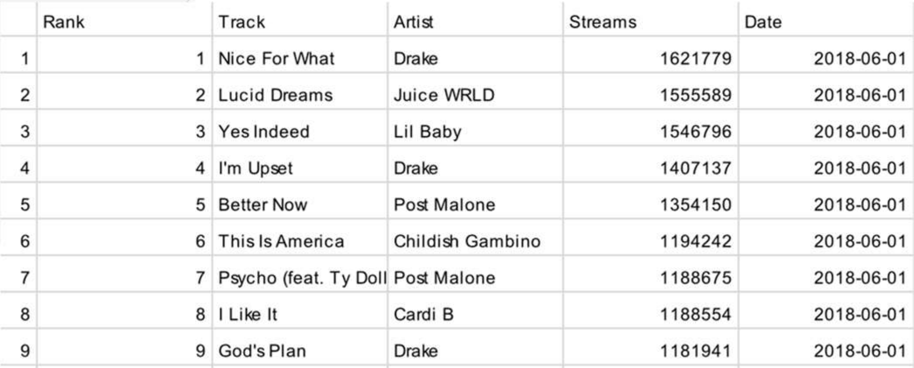
Code of Spotify table:



~~~
Shopping for Attributes with SelectorGadget(use goolge CHROME and make it an
extension)
```{r}
SpotifyScrape <- function(x){
 page <- x
 rank <- page %>% read_html() %>% html_nodes('.chart-table-position') %>%
html_text() %>% as.data.frame()
 track <- page %>% read_html() %>% html_nodes('strong') %>% html_text() %>%
as.data.frame()
 artist <- page %>% read_html() %>% html_nodes('.chart-table-track span') %>%
html_text() %>% as.data.frame()
 streams <- page %>% read_html() %>% html_nodes('td.chart-table-streams') %>%
html_text() %>% as.data.frame()
 dates <- page %>% read_html() %>% html_nodes('.responsive-select~
.responsive-select+ .responsive-select .responsive-select-value') %>% html_text()
%>% as.data.frame()

#combine, name, and make it a tibble
 chart <- cbind(rank, track, artist, streams, dates)
 names(chart) <- c("Rank", "Track", "Artist", "Streams", "Date")
 chart <- as.tibble(chart)
 return(chart)
}
```
~~~

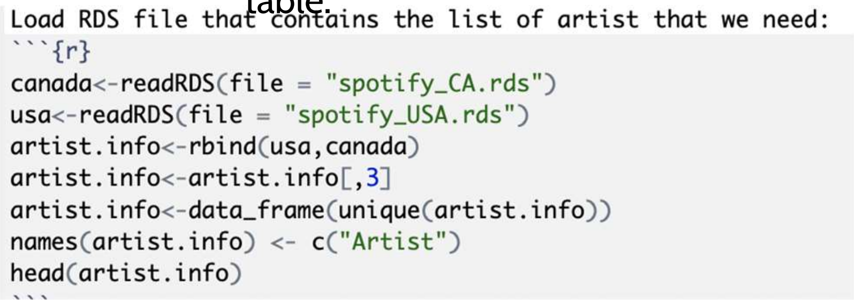
RStudio: the RVEST package and the Chrome Selector Gadget tool have allow us to collect HTML Nodes and download the data that we need to build our database. Out goal is to collect enough data to advise marketing campaigns before they decide to invest money in artist.

Outcome table

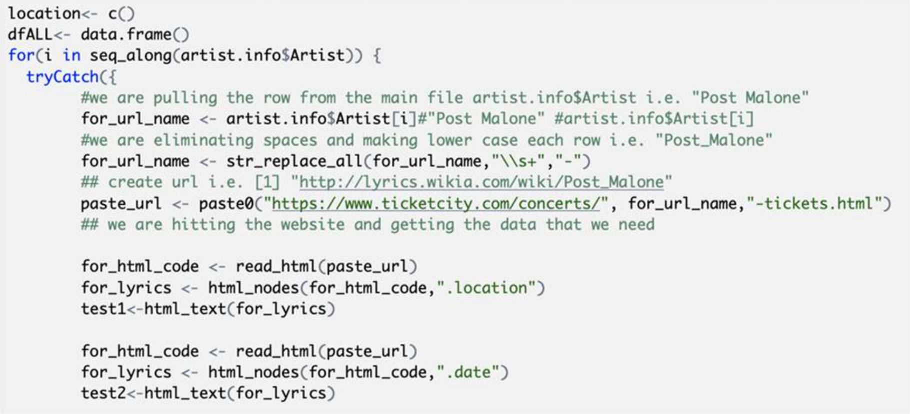


	Rank	Track	Artist	Streams	Date
1	1	Nice For What	Drake	1621779	2018-06-01
2	2	Lucid Dreams	Juice WRLD	1555589	2018-06-01
3	3	Yes Indeed	Lil Baby	1546796	2018-06-01
4	4	I'm Upset	Drake	1407137	2018-06-01
5	5	Better Now	Post Malone	1354150	2018-06-01
6	6	This Is America	Childish Gambino	1194242	2018-06-01
7	7	Psycho (feat. Ty Doll	Post Malone	1188675	2018-06-01
8	8	I Like It	Cardi B	1188554	2018-06-01
9	9	God's Plan	Drake	1181941	2018-06-01

Code of Concert table:

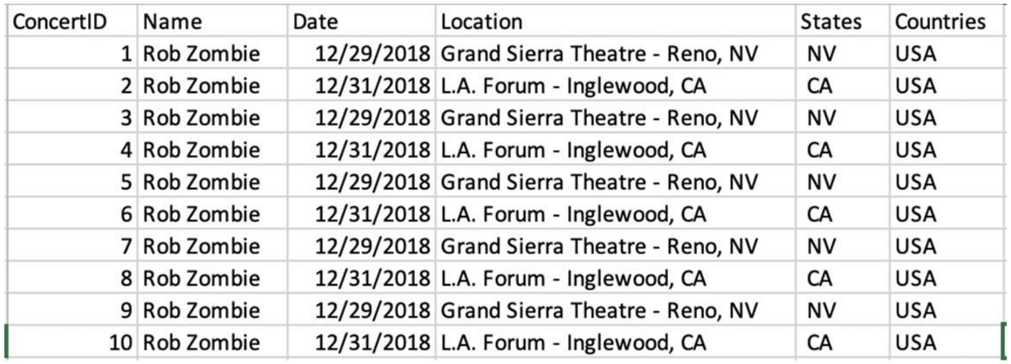


~~~~
Load RDS file that contains the list of artist that we need:
```{r}
canada<-readRDS(file = "spotify_CA.rds")
usa<-readRDS(file = "spotify_USA.rds")
artist.info<-rbind(usa,canada)
artist.info<-artist.info[,3]
artist.info<-data_frame(unique(artist.info))
names(artist.info) <- c("Artist")
head(artist.info)
```
~~~~



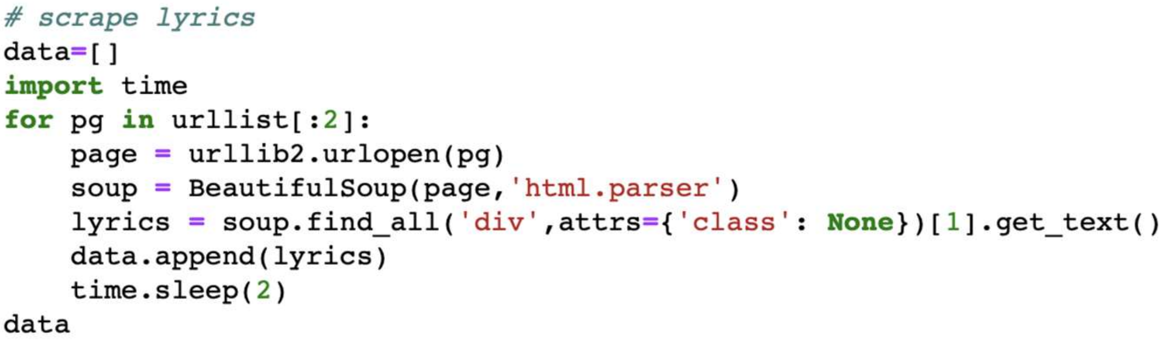
```
location<- c()
dfALL<- data.frame()
for(i in seq_along(artist.info$Artist)) {
  tryCatch({
        #we are pulling the row from the main file artist.info$Artist i.e. "Post Malone"
        for_url_name <- artist.info$Artist[i]#"Post Malone" #artist.info$Artist[i]
        #we are eliminating spaces and making lower case each row i.e. "Post_Malone"
        for_url_name <- str_replace_all(for_url_name,"\\s+","-")
        ## create url i.e. [1] "http://lyrics.wikia.com/wiki/Post_Malone"
        paste_url <- paste0("https://www.ticketcity.com/concerts/", for_url_name,"-tickets.html")
        ## we are hitting the website and getting the data that we need

        for_html_code <- read_html(paste_url)
        for_lyrics <- html_nodes(for_html_code,".location")
        test1<-html_text(for_lyrics)

        for_html_code <- read_html(paste_url)
        for_lyrics <- html_nodes(for_html_code,".date")
        test2<-html_text(for_lyrics)
```

Outcome table



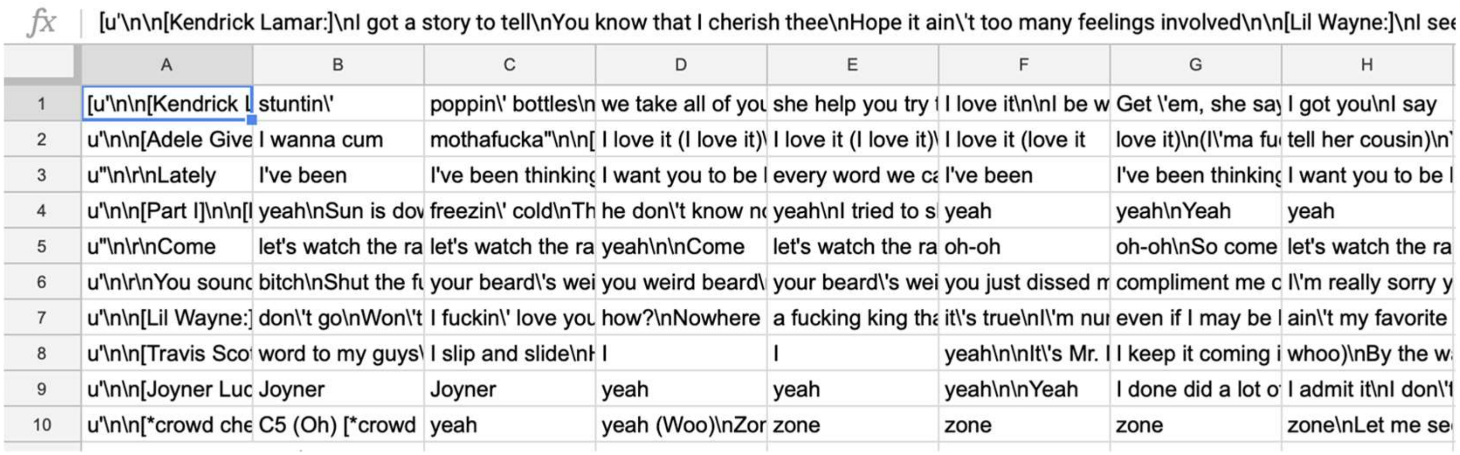
ConcertID	Name	Date	Location	States	Countries
1	Rob Zombie	12/29/2018	Grand Sierra Theatre - Reno, NV	NV	USA
2	Rob Zombie	12/31/2018	L.A. Forum - Inglewood, CA	CA	USA
3	Rob Zombie	12/29/2018	Grand Sierra Theatre - Reno, NV	NV	USA
4	Rob Zombie	12/31/2018	L.A. Forum - Inglewood, CA	CA	USA
5	Rob Zombie	12/29/2018	Grand Sierra Theatre - Reno, NV	NV	USA
6	Rob Zombie	12/31/2018	L.A. Forum - Inglewood, CA	CA	USA
7	Rob Zombie	12/29/2018	Grand Sierra Theatre - Reno, NV	NV	USA
8	Rob Zombie	12/31/2018	L.A. Forum - Inglewood, CA	CA	USA
9	Rob Zombie	12/29/2018	Grand Sierra Theatre - Reno, NV	NV	USA
10	Rob Zombie	12/31/2018	L.A. Forum - Inglewood, CA	CA	USA

Code of Lyrics table:



```
# scrape lyrics
data=[]
import time
for pg in urllist[:2]:
    page = urllib2.urlopen(pg)
    soup = BeautifulSoup(page,'html.parser')
    lyrics = soup.find_all('div',attrs={'class': None})[1].get_text()
    data.append(lyrics)
    time.sleep(2)
data
```

Outcome table



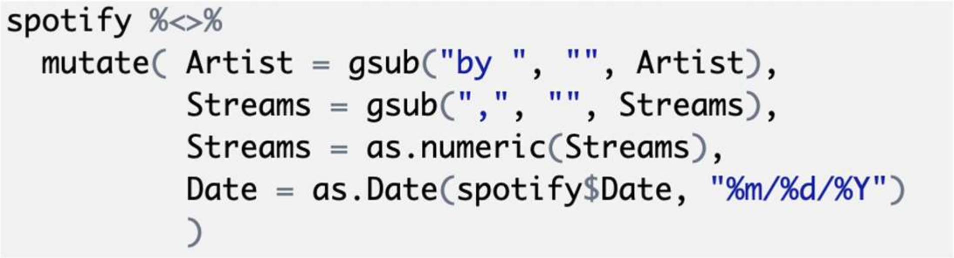
fx [u'\n\n[Kendrick Lamar:]\nI got a story to tell\nYou know that I cherish thee\nHope it ain\'t too many feelings involved\n\n[Lil Wayne:]\nI see

	A	B	C	D	E	F	G	H
1	[u'\n\n[Kendrick l	stuntin\'	poppin\' bottles\n	we take all of you	she help you try t	I love it\n\nI be w	Get \'em, she say	I got you\nI say
2	u'\n\n[Adele Give	I wanna cum	mothafucka"\n\n[	I love it (I love it)\	I love it (I love it)\	I love it (love it	love it)\n(I\'ma fu	tell her cousin)\n
3	u"\n\r\nLately	I've been	I've been thinking	I want you to be l	every word we ca	I've been	I've been thinking	I want you to be l
4	u'\n\n[Part I]\n\n[I	yeah\nSun is dov	freezin\' cold\nTh	he don\'t know nc	yeah\nI tried to s	yeah	yeah\nYeah	yeah
5	u"\n\r\nCome	let's watch the ra	let's watch the ra	yeah\n\nCome	let's watch the ra	oh-oh	oh-oh\nSo come	let's watch the ra
6	u'\n\r\nYou sounc	bitch\nShut the fu	your beard\'s wei	you weird beard\	your beard\'s wei	you just dissed m	compliment me c	I\'m really sorry y
7	u'\n\n[Lil Wayne:]	don\'t go\nWon\'t	I fuckin\' love you	how?\nNowhere	a fucking king tha	it\'s true\nI\'m nu	even if I may be l	ain\'t my favorite
8	u'\n\n[Travis Sco	word to my guys\	I slip and slide\nH	I	I	yeah\n\nIt\'s Mr. I	I keep it coming i	whoo)\nBy the wa
9	u'\n\n[Joyner Luc	Joyner	Joyner	yeah	yeah	yeah\n\nYeah	I done did a lot o	I admit it\nI don\'t
10	u'\n\n[*crowd che	C5 (Oh) [*crowd	yeah	yeah (Woo)\nZor	zone	zone	zone	zone\nLet me se

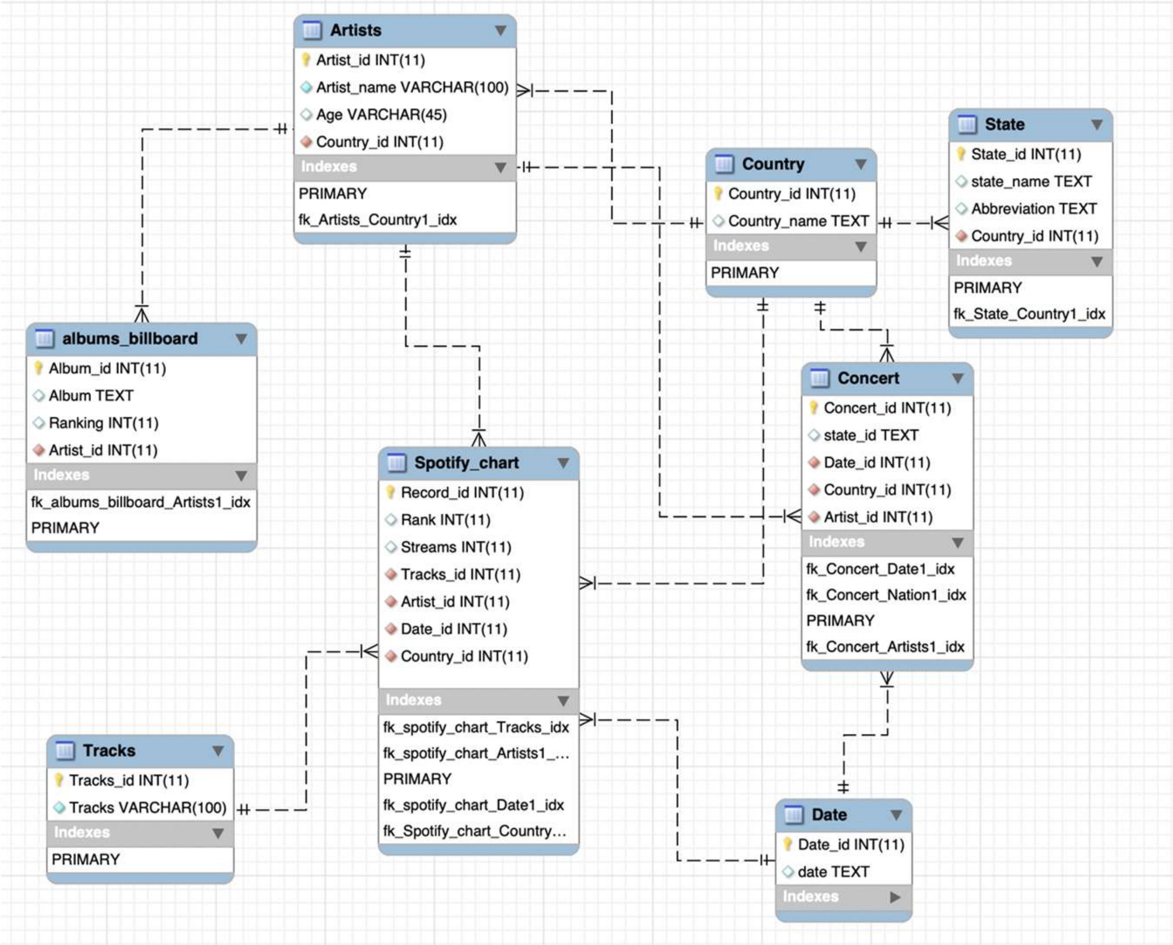
DATA CLEANING

RStudio

- **Lubridate**: date manipulation
- **Rvest**: extract pieces out of HTML documents using XPath and css selectors.
  - html_nodes ()
- **Dplyr** :Is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:
  - mutate() select() filter() summarise() arrange()



```
spotify %<>%
  mutate( Artist = gsub("by ", "", Artist),
          Streams = gsub(",", "", Streams),
          Streams = as.numeric(Streams),
          Date = as.Date(spotify$Date, "%m/%d/%Y")
          )
```



We have 5 main tables -- artist, albums, spotify, concert and songs

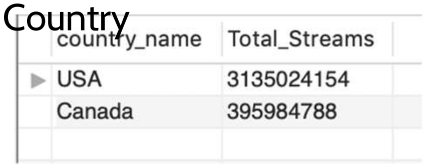
After normalization, we have extra tables – country, state and date, adding up to 8 tables

For attribute of ID, ranking, stream, age are INT type; country_name, state_name, abbreviation, album and Artist_name are either TEXT or VARCHAR type

The nature of our database is snowflake database and our fact-table is Spotify_chart

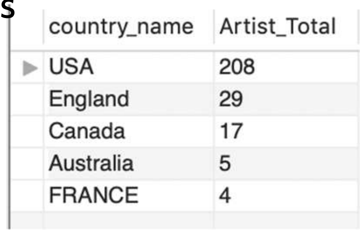
SQL QUERIES RESULT

Total of streams per Country



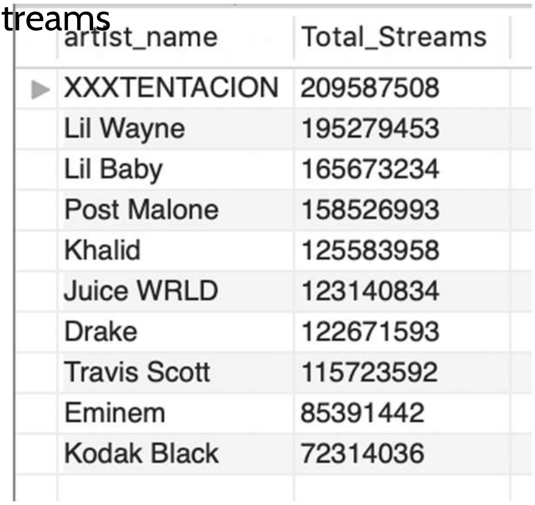
country_name	Total_Streams
USA	3135024154
Canada	395984788

Top 5 countries with the most Artists



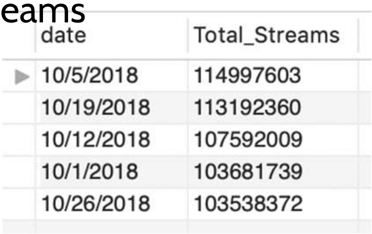
country_name	Artist_Total
USA	208
England	29
Canada	17
Australia	5
FRANCE	4

Top 10 artists with the most streams



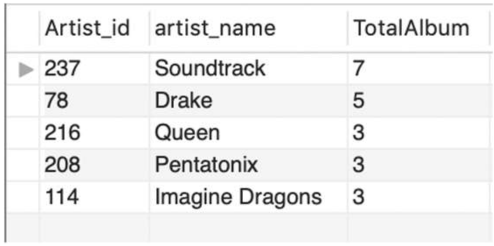
artist_name	Total_Streams
XXXTENTACION	209587508
Lil Wayne	195279453
Lil Baby	165673234
Post Malone	158526993
Khalid	125583958
Juice WRLD	123140834
Drake	122671593
Travis Scott	115723592
Eminem	85391442
Kodak Black	72314036

Top 5 days with the most streams



date	Total_Streams
10/5/2018	114997603
10/19/2018	113192360
10/12/2018	107592009
10/1/2018	103681739
10/26/2018	103538372

Top 5 artist who in the top 200 billboard

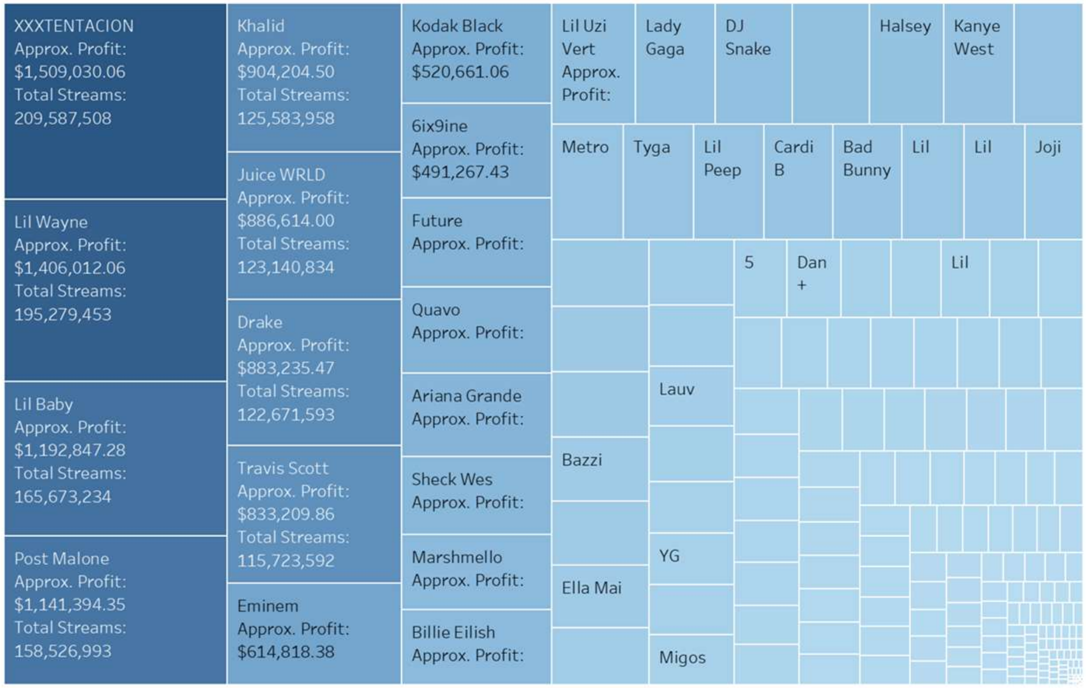


Artist_id	artist_name	TotalAlbum
237	Soundtrack	7
78	Drake	5
216	Queen	3
208	Pentatonix	3
114	Imagine Dragons	3

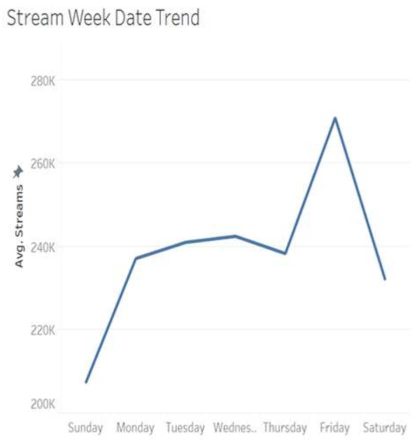
STREAMS

- Quantify the relationship between streams and dollars
- Dashboard to target potential artist for endorsements
- Relationship between content with the highest streaming frequency and sentiment
- Alignment between brand and artist values

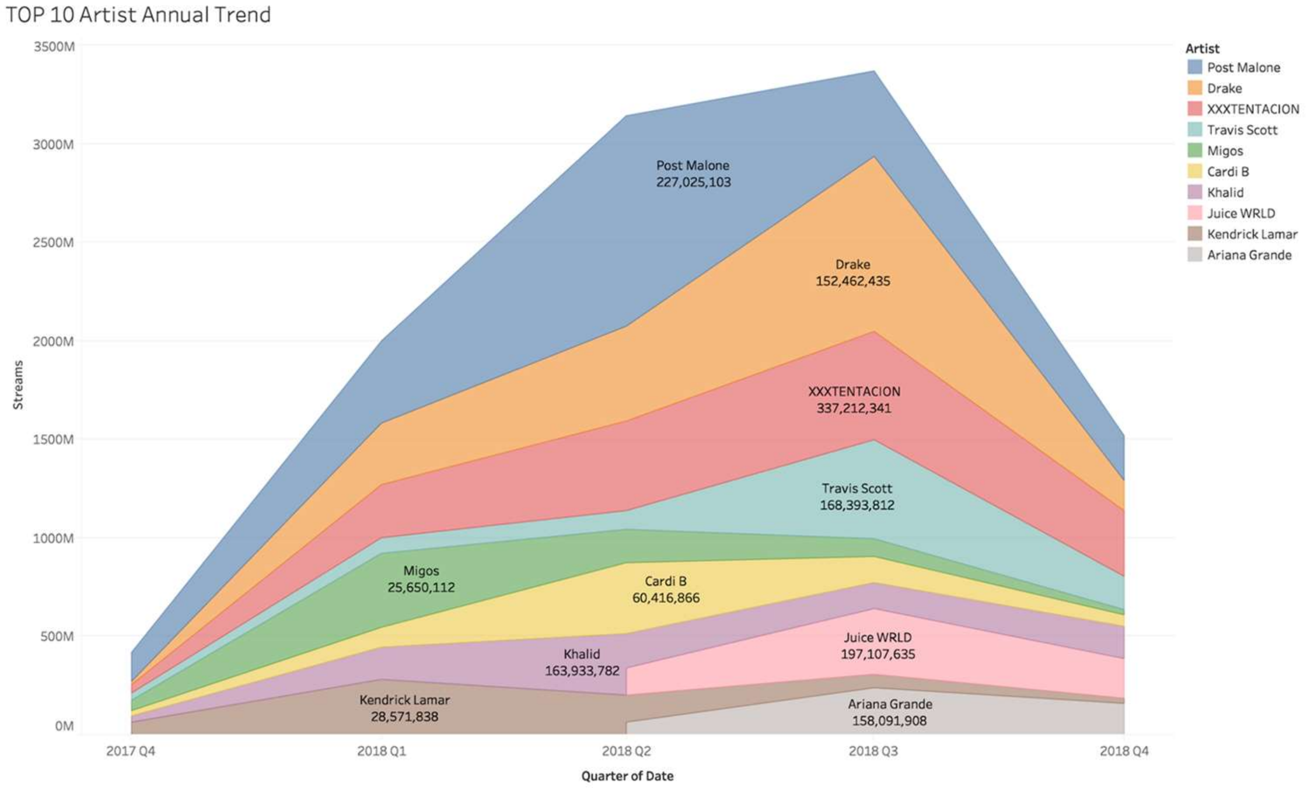
All Artists Streaming



MUSICIANS TRENDS

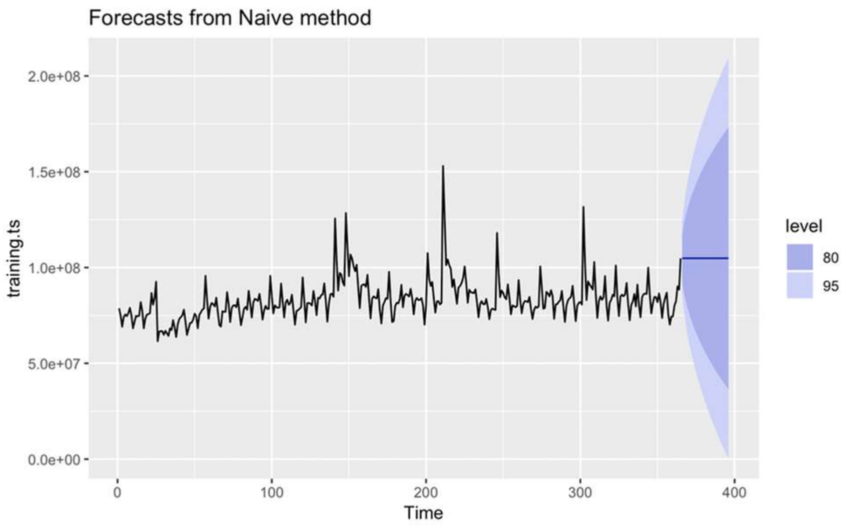


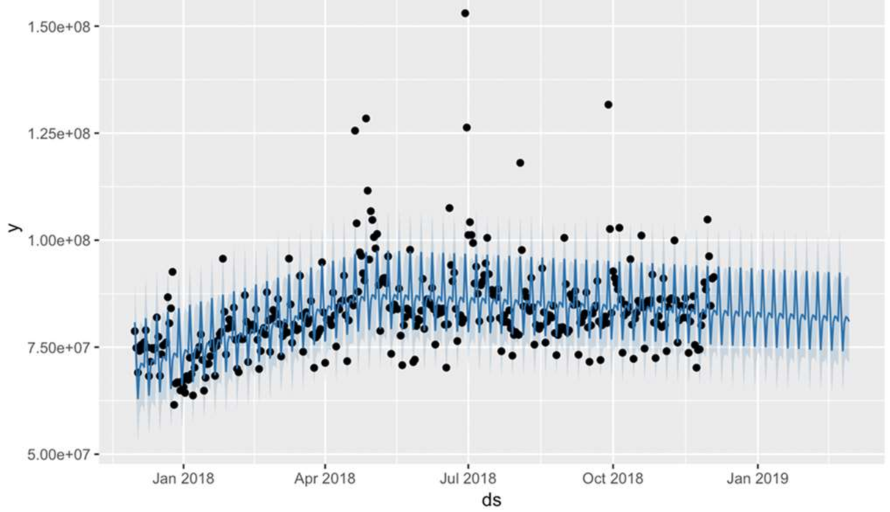
Friday should be a good time to buy advertising on Spotify



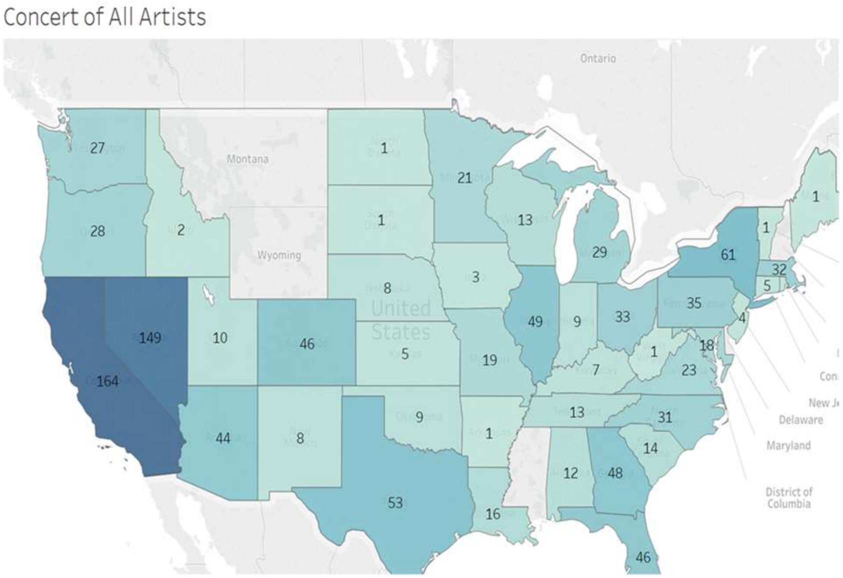
The plot of sum of Streams for Date Quarter. Color shows details about Artist. The marks are labeled by Artist and sum of Streams. The view is filtered on Artist, which keeps 10 of 472 members.

FORECASTING – NAIVE VS PROPHET

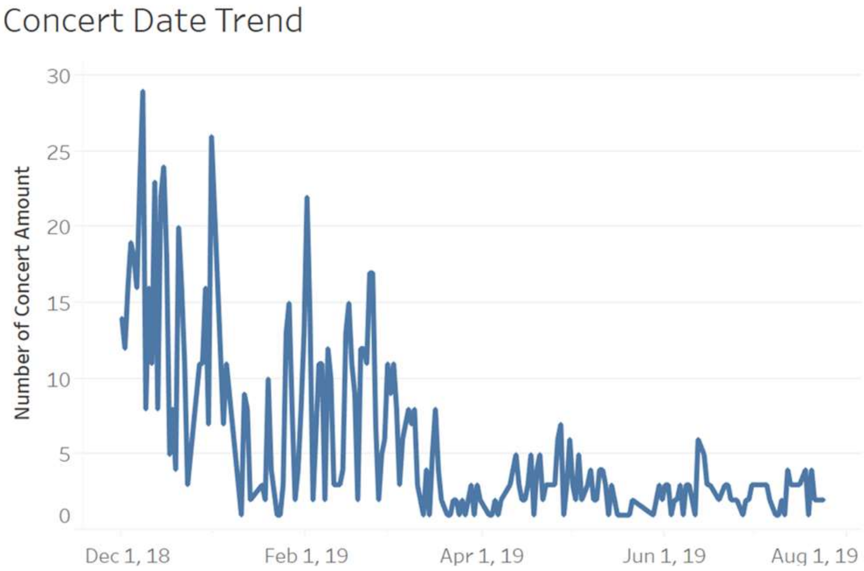




CONCERT PRODUCTION



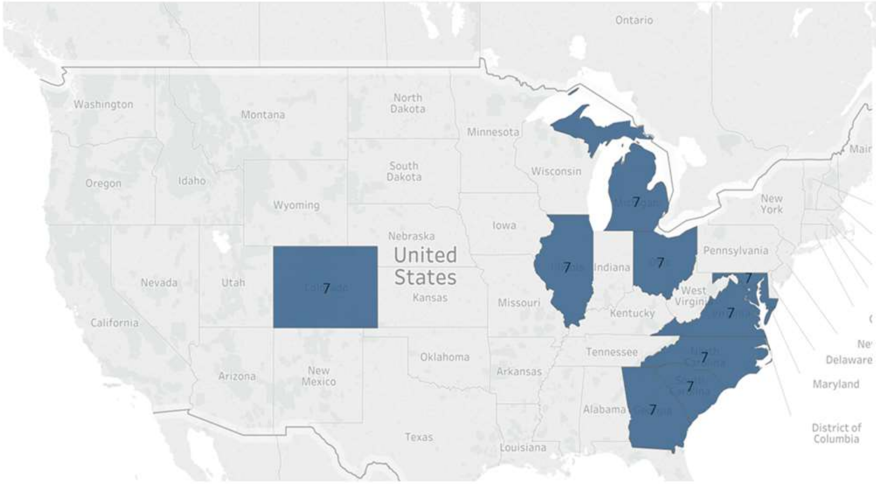
States with more concerts can come with advertising campaigns to specific customers group like music fans



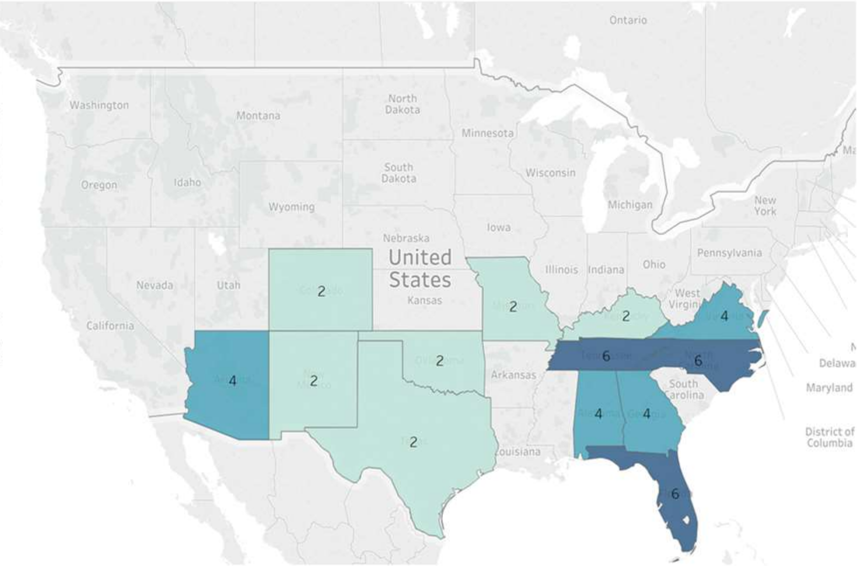
Most concerts are held on December and January.

MUSICIANS BRANDING

Concert of Mitchell Tenpenny



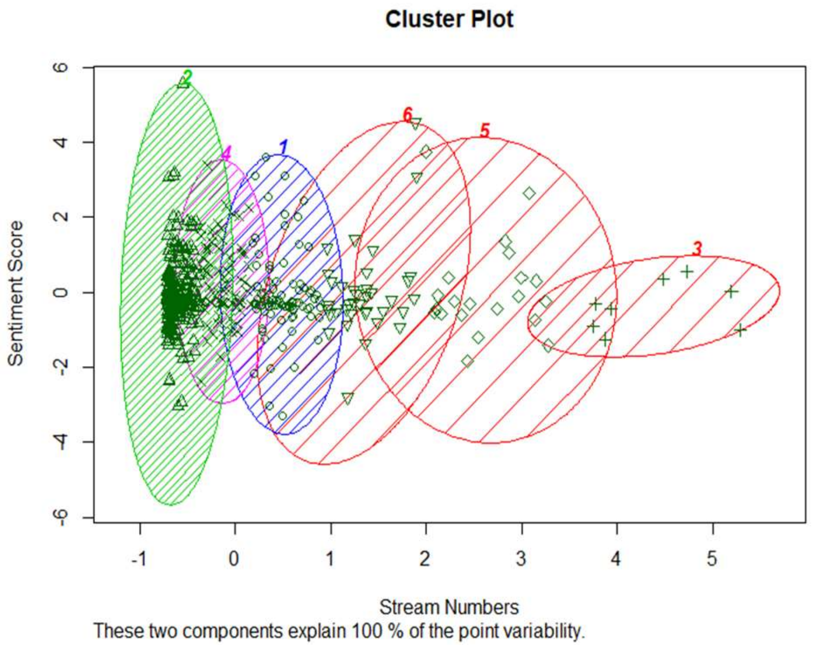
Concert of Lord Huron



Top artists have most concerts next year has different preference to hold their concert, they can exploit more locations based on top concert locations.

LYRICS ANALYSIS:

- Sentiment Analysis: negative words “” positive words “”
- Popular songs tend to be slightly negative
  (correlation: -0.12), mean value is also negative
- Most songs are centered around neutral sentiments
- Some songs are very positive while being still very popular
- Companies can identify artists that are overall positive or negative - drill down which artists resonate with their overall theme.
- Companies can narrow down to songs that evoke specific emotion that can resonates with message of advertisement.



Summary

- We want to use Spotify data to help artists, advertisement companies, entertainment, and talent agencies to make more informed decisions
- We collect artist, lyrics, stream, and concert ticket data, and evaluate metrics of popularity and sentiment
- Entertainment and talent agencies can get an estimate of concert ticket price based on region
- Advertisement companies can quickly narrow down songs and artists that better reflect the theme of advertisement

MORE DATA FOR DEEPER INSIGHT

- Go more international, collect data from countries other than North America
- Research and map the sentiments of successful advertisement songs to new songs in Spotify
- Forecasting each artist's total streams per day as time series data
  - Scrape more historical data rather than two-months frame
  - With greater abundance of longitudinal data, we can forecast with greater accuracy per artist, so go granular.

DATA ENGINEERING IS HARDER THAN IT LOOKS!

1. Use “sleep system” when scrapping to prevent your IP Address to be blocked
2. Use API’s available instead of HTML nodes in web page scraping
3. Forecast analysis requires at least 2 years of data in order to capture seasonality
4. Some tools are more convenient and useful in certain aspects of data processing
5. Only having a few variables is often not enough to produce a satisfying analysis even though data collection and processing is very painful

THANK YOU

APPENDIX

- https://www.statista.com/chart/15697/spotify-user-growth/
- https://www.cnbc.com/2018/01/26/how-spotify-apple-music-can-pay-musicians-more-commentary.html
- https://www.wikipedia.org
- https://www.spotify.com/us/
- https://www.spotify.com/ca-en/
- https://www.billboard.com/charts/billboard-200
- https://www.ticketcity.com
- http://www.azlyrics.com
- https://facebook.github.io/prophet/docs/quick_start.html#r-api