Cs 360/560

Lyrical Notes Final Project Process Book

Allison Wong, Ien Whang, Kelly Heaney

Project code: https://github.com/azywong/lyricalnotes
Project website: https://azywong.github.io/lyricalnotes/

Overview and Motivation

Overview

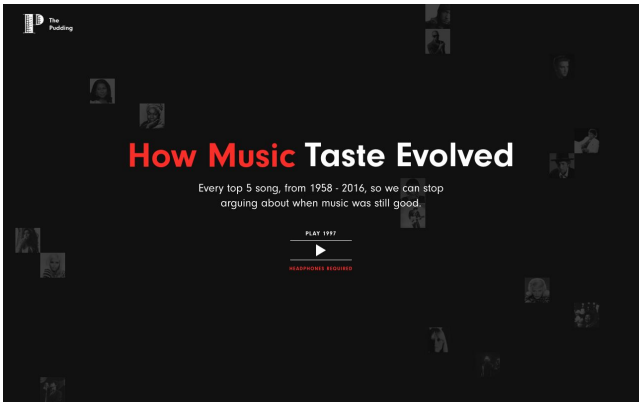
The Billboard Hot 100 uses radio airplay, sales, and streaming metrics to determine the popularity of a song on a weekly basis. Using some APIs, we scraped 25 years of weekly Billboard Top 100 charts and the lyrics of over 8,000 unique songs. Our goal is to visualize how music and music taste evolved between 1990 and 2015.

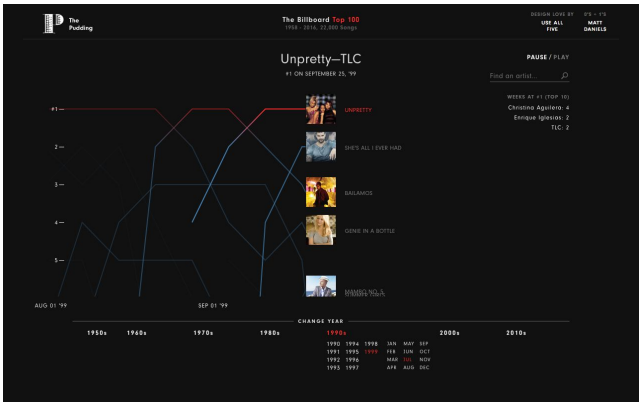
Motivation

Music is a wonderful thing that almost everyone enjoys, and it is something the all of us appreciate very much. One thing about music that is apparent to us is that people have different tastes in music — it is different from one individual to the next, and has probably even changed over time for each individual as well. These tastes are reflected in the songs produced by the music industry and we are curious to see how these tastes have influenced the music that has been produced. We suspect that within the music, there are trends that reflect the climate of society, and how it has shifted through time. Music is much more than just sound; It is powerful and has the incredible power to mirror the deepest human emotions, desires, and experiences. Through the data, we hope to uncover the meaning behind the music and gain a better understanding of its influence on us, and how we've influenced it.

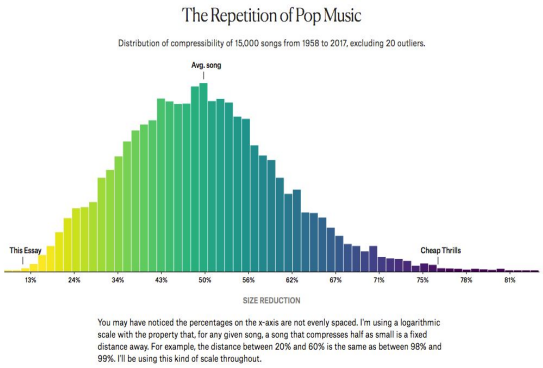
Related Work

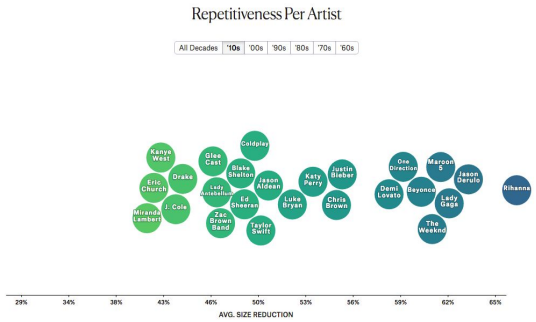
How Music Taste Evolved on Pudding.cool



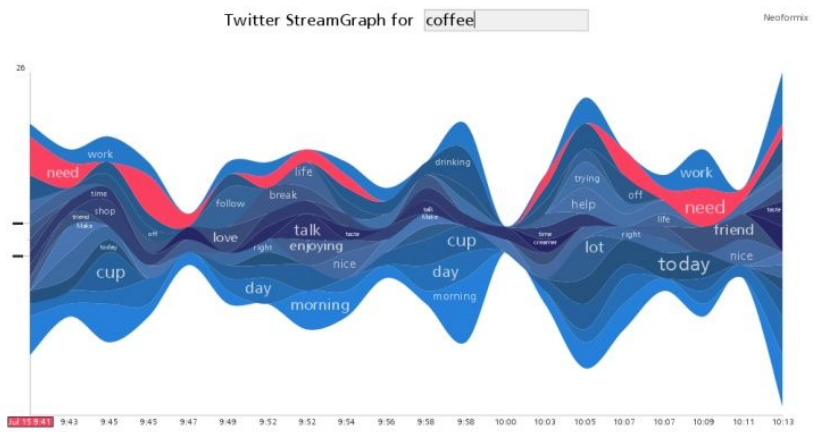


Are Pop Lyrics Getting More Repetitive on Pudding.cool

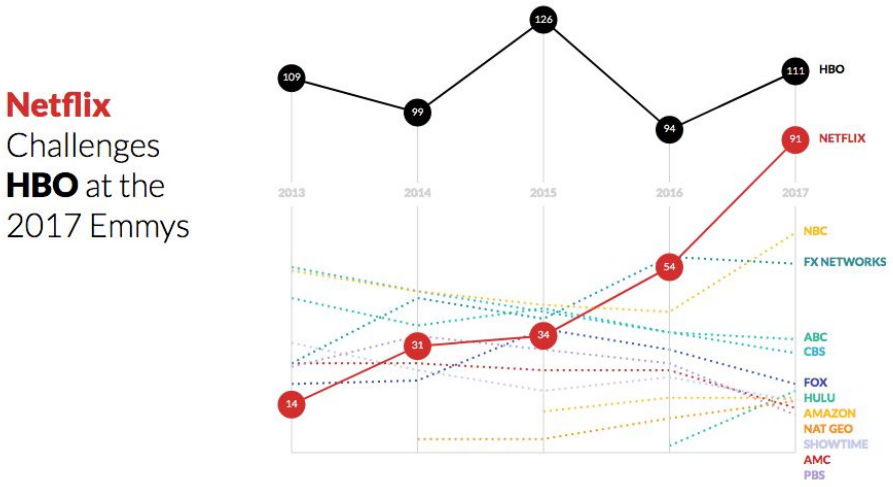




Stream Graph on Keyword Twitter Data on Neoformix



Line Chart by Susie Lu on Bl.ocks.org



Questions

- Explore trends in popular music across more than two decades to see if anything stands out
- Find patterns in lyrics such as word frequency and word choice
- Determine if current events or social climate influence popularity of certain words or the music heard
- Do word choices change or get more repetitive over time?
- What does the vocabulary of a typical pop song look like? How about over time?
- How much do certain keywords show up through pop songs?

We saw a lot of repetition in songs on the charts, so during the course of our project, we focused on the top artists and songs in terms of weeks on the chart, then had more exploratory visualizations in terms of lyrics search. The lyrics data was so large, that it was easier to present them as an explorable data set as in visualization 4, with a separate visualization (as in visualization 3) for emphasis on interesting ones.

Data

Scraping

We wrote a nodejs app to hit the billboard API and save each week's chart data as a json file. Then, after parsing each week within our range (1990-2015), we ran the song and artist data through two different lyrics finding methods in order to get as many song lyrics as possible.

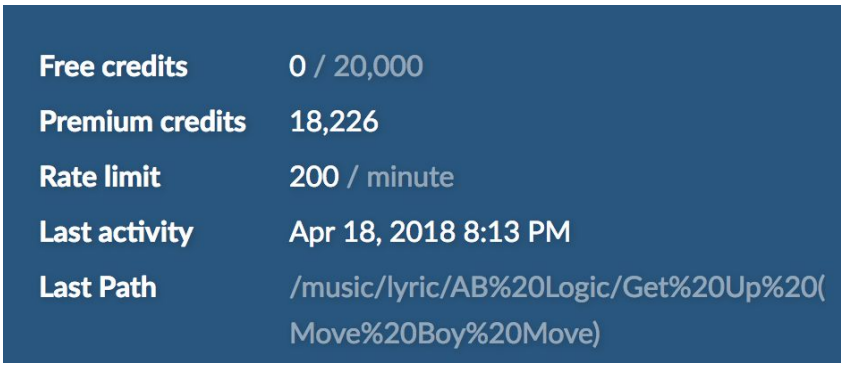
Billboard API

http://billboard.modulo.site/

We scraped each week of billboard rankings from 1958 to 2015 using a node.js app More specifically, the app calculates all the Saturdays in the date range (the API only takes Saturdays as input) and hits the url endpoint, then saves the resulting json. This API had no rate limiting and was pretty painless.

Song lyrics

API seeds: https://orion.apiseeds.com/



We initially went with this API which rate limits us to up to 200 requests a minute, and limits the total number of requests. Timing the requests using node.js took a bit, and also accounting for differences in artist names and song titles between the APIs (especially songs with featuring artists, parentheses, or special characters in the title). We got the bulk of the lyrics from this API.

Lyric-get node js package: https://github.com/rhnvrm/lyric-api

This nodejs package does web crawling work from a lyrics website. It doesn't rate limit, but there are also no descriptive error messages, so we only did two passes with the remaining input, once without correcting for featuring artists and parentheses, and once correcting for those factors. We got approximately two thousand remaining lyrics from this method.

Clean up/aggregation

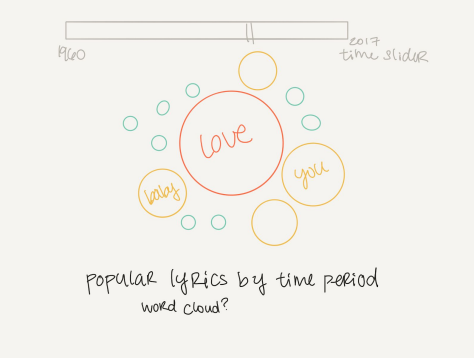
The cleanup was performed in R. The job entailed:

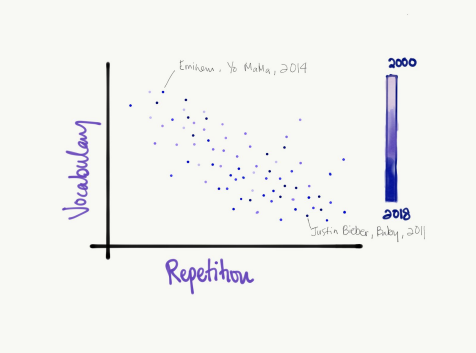
- Parsing JSON files
- Extracting word and artist frequencies by aggregating unique entries
- Splitting and cleaning lyric strings for manipulation and analysis
- Retrieving IDs of missing entries to keep track of files needed
- Piecing together intermediate and final CSV files for individual visualisations

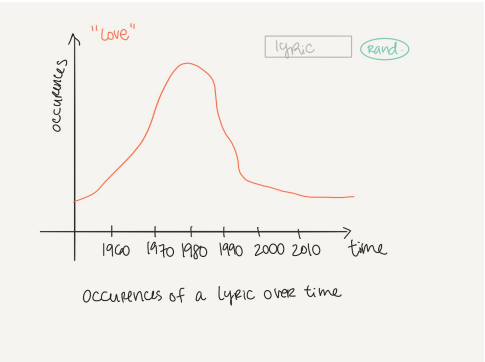
The process was time consuming mainly because of the quantity of data we had scraped, but then again the algorithms used could have been designed better. All in all, it was an integral process that was completed without too much hassle.

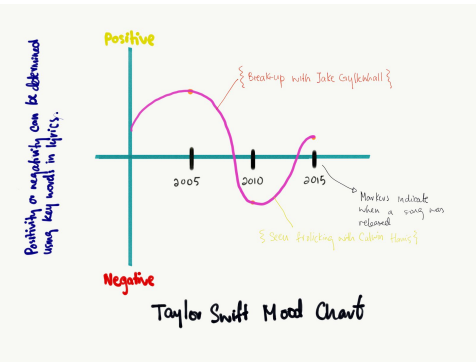
Exploratory Data

We came up with some preliminary designs for our initial proposal while the data was processing. We knew what format the data would take (weekly billboard data that we would condense down to each month) and json lyrics files. What we didn't know was what the end data would look like in terms of word frequency and patterns.





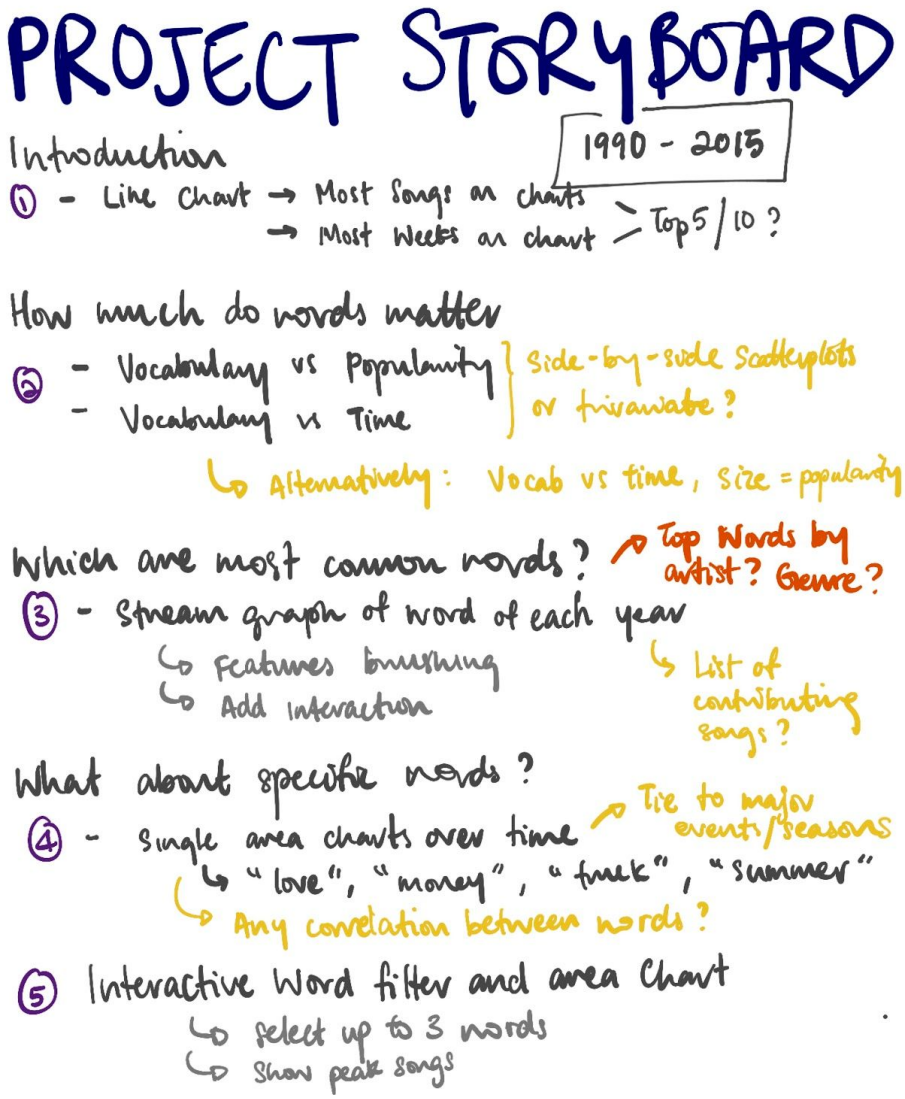




Analysis

After some review, our project didn't have a common thread across visualizations to tie the ideas together. At this point we had roughly 20-30% of our data, so we didn't have a full idea of what the trends or patterns in the data would look like yet.

Based on what we did have and our initial ideas, we put together a revised storyboard.

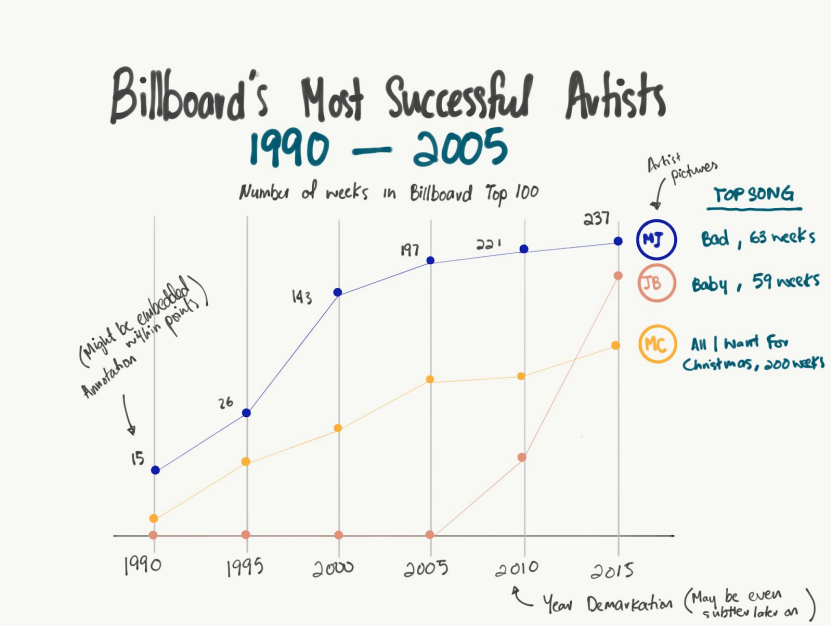


Based on what information we did have at the time, and throughout our project's life, we found that some of our initial visualizations weren't so viable or interesting (such as an early heat map of artists by state). In this storyboard, we narrowed down our large list of general ideas to four visualizations that made sense in terms of progression and how they fit together.

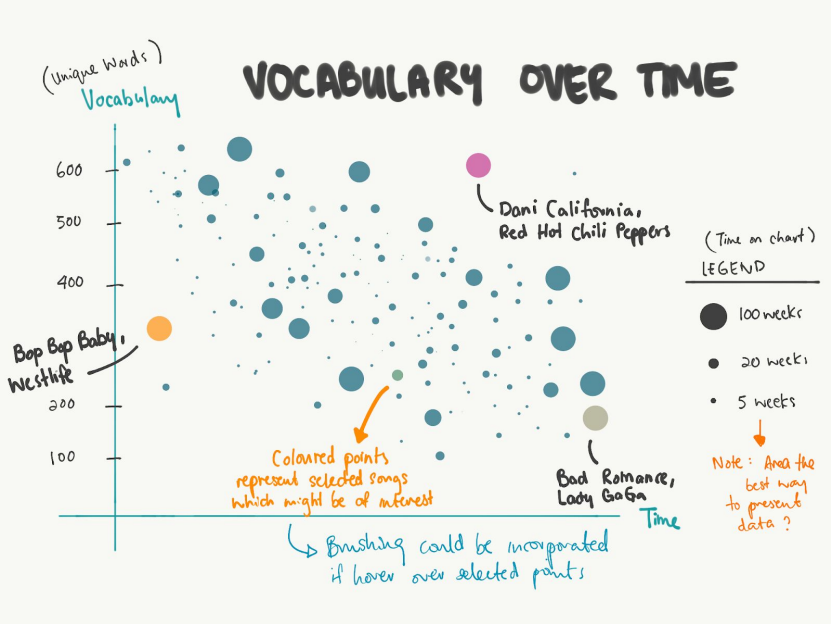
In terms of visualizations, the data set was too large to completely load into Tableau, so we had to narrow down the dataset specifically for each visualization. Then we directly put the data into preliminary visualizations in d3 to see how they would look.

Design Evolution

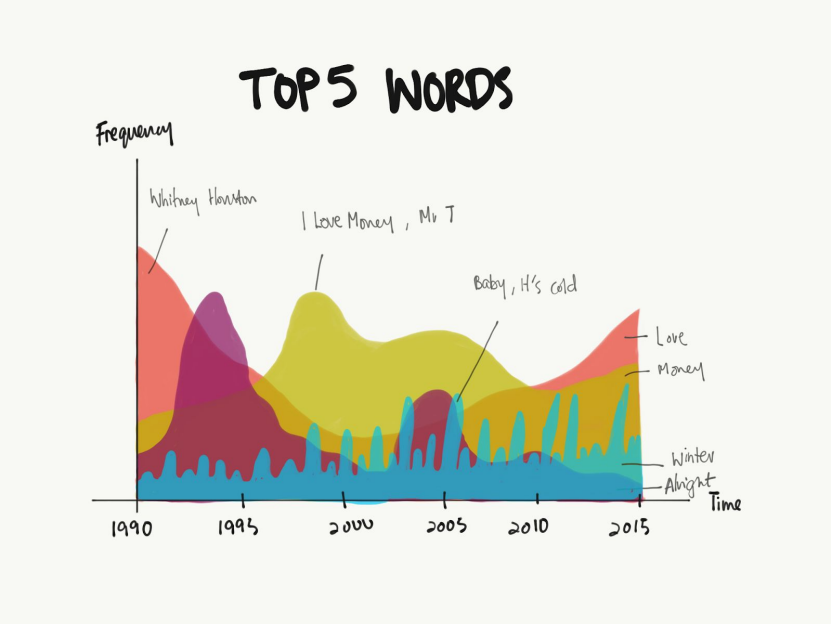
Our revised story board was based on feedback after the initial proposal, but was still before we had the entire data set. These are the initial sketches from our revised proposal.



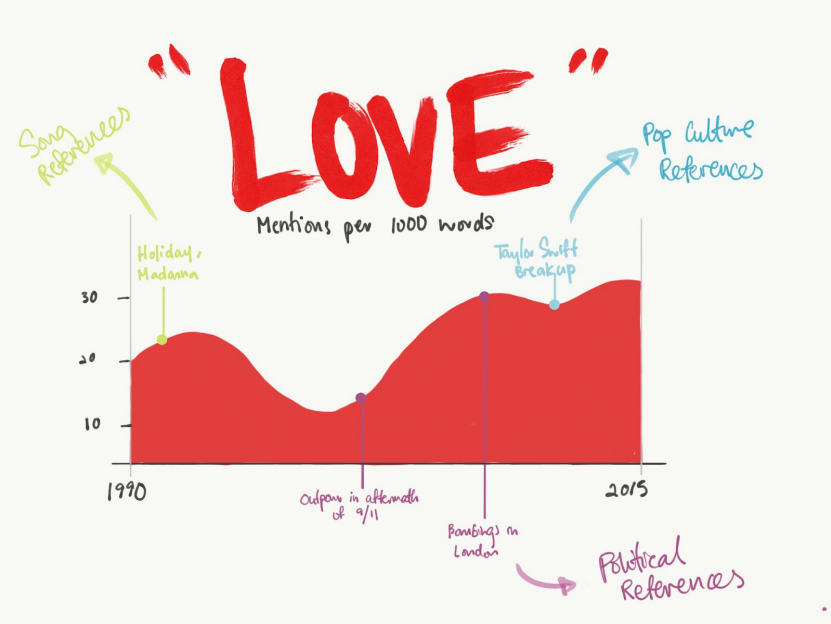
This visualization came after noticing the amount of repetition on the charts, and after some observation, how Mariah Carey had a seasonal spike yearly during Christmas. It changed over time to include annotations relative to interesting moments on the chart, rather than which song was charting at the time (since there could be multiples).



This visualization was designed to show vocabulary of top songs over time. Originally we thought to have colors to show select songs or some kind of different factor, but it turned out to be too distracting. Keeping it to just one color was better, and having size show weeks on chart, and turning on opacity to show overlap and selections.



This visualization in terms of the key idea stayed the most similar over time. In the original implementation, we added too many colors and a zoomable effect, which got ugly fast.



Instead of a word cloud with a scroller for time, we went with a searchable view that showed the progression of a word over time. Rather than seeing an arbitrary number of words per year, we decided it would be more fun for it to be searchable.

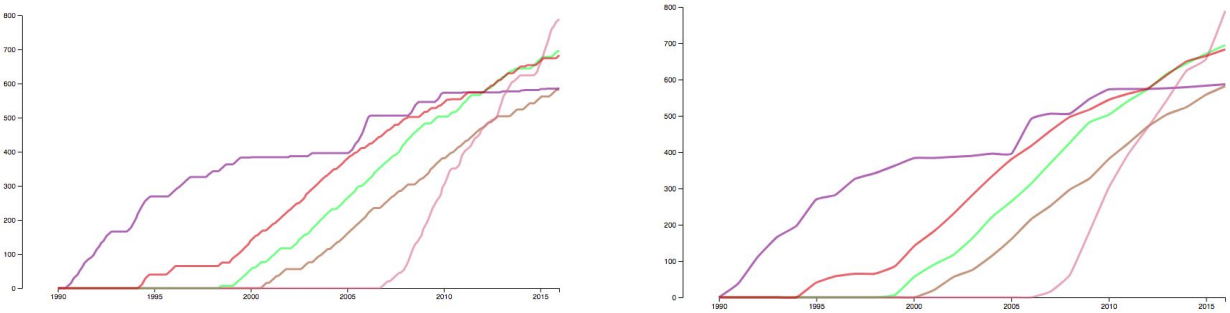
We combined the annotation idea to the visualization above and had this data visualization be searchable with an input field.

Implementation

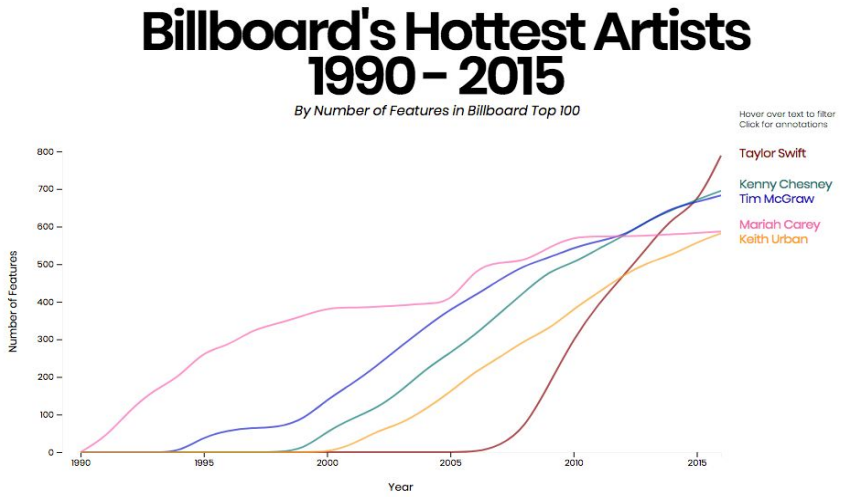
The visualizations were all implemented with d3.js

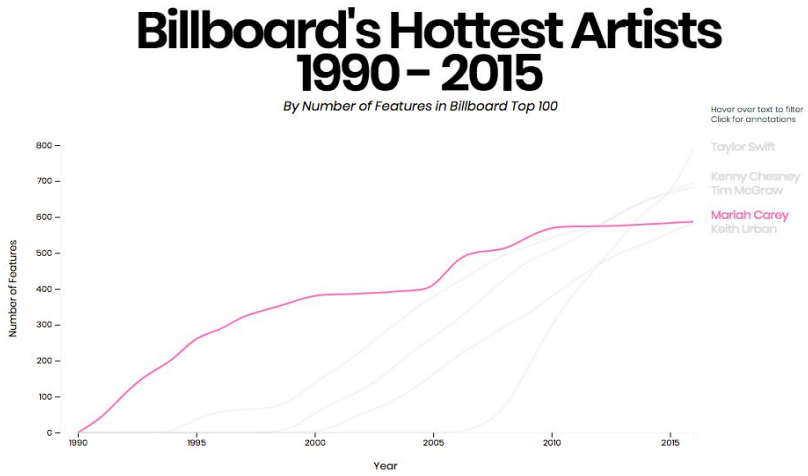
Visualization 1

Our introductory visualization offers a general overview of the five most prolific artists between 1990 to 2015 in terms of their number of appearances on Billboard's Hot 100.



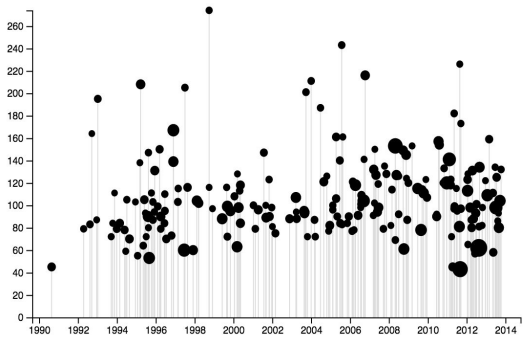
The improved visualization incorporates smoother lines, brushing, and on-demand annotations to give the viewer a brief overview each artist's music career.

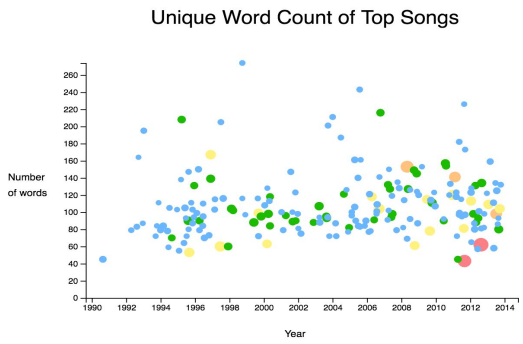


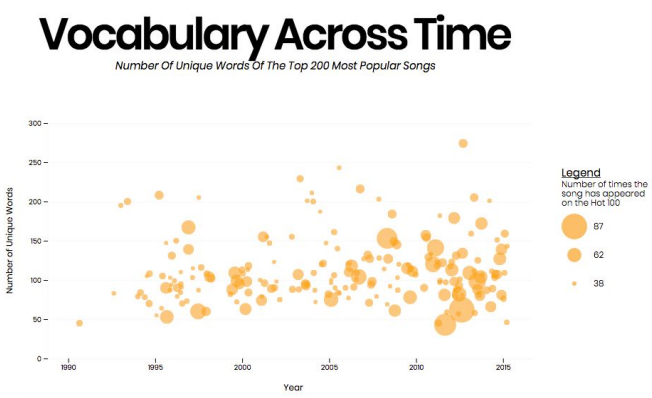


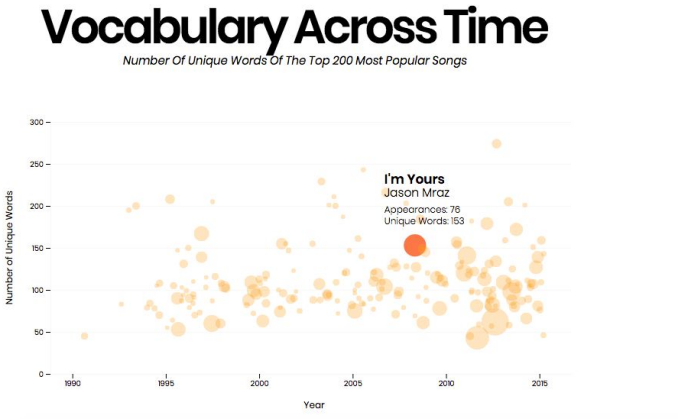
Visualization 2

The early visualization gave us an idea of how our data looked and provided us direction with how to design later versions.





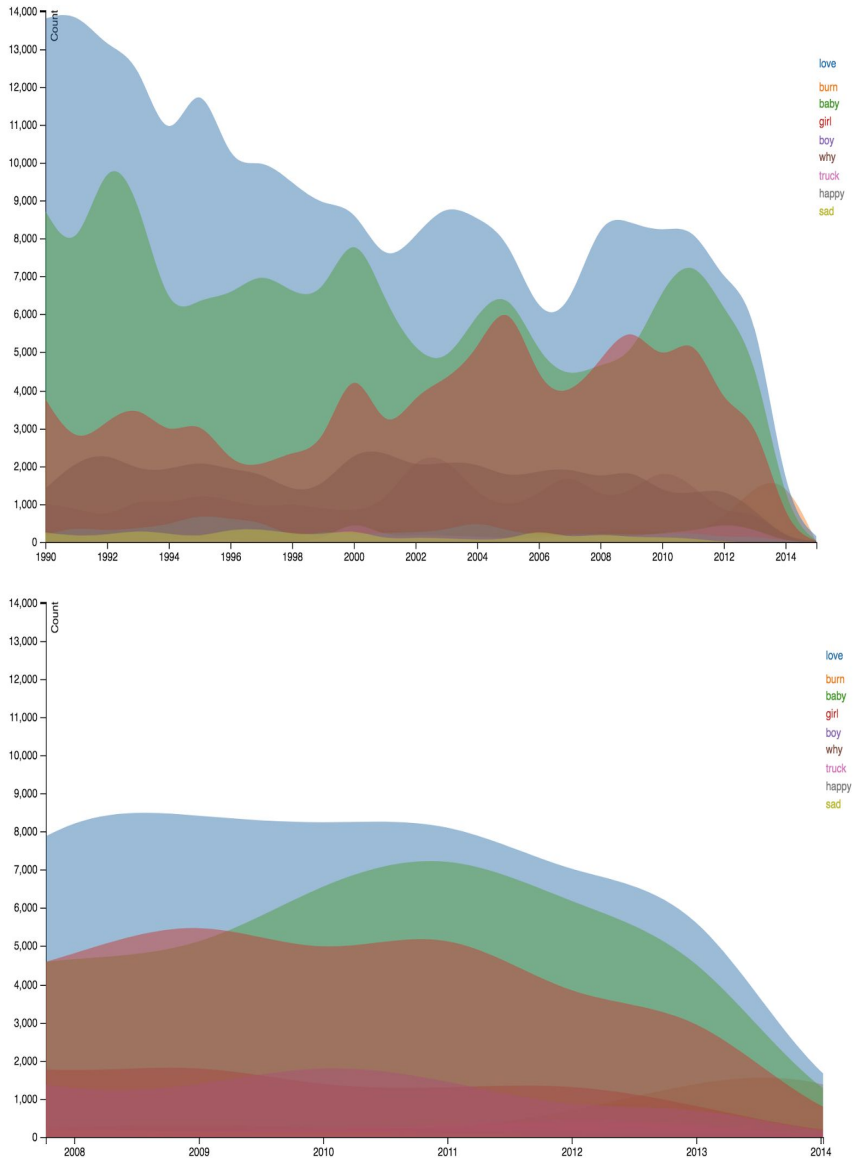




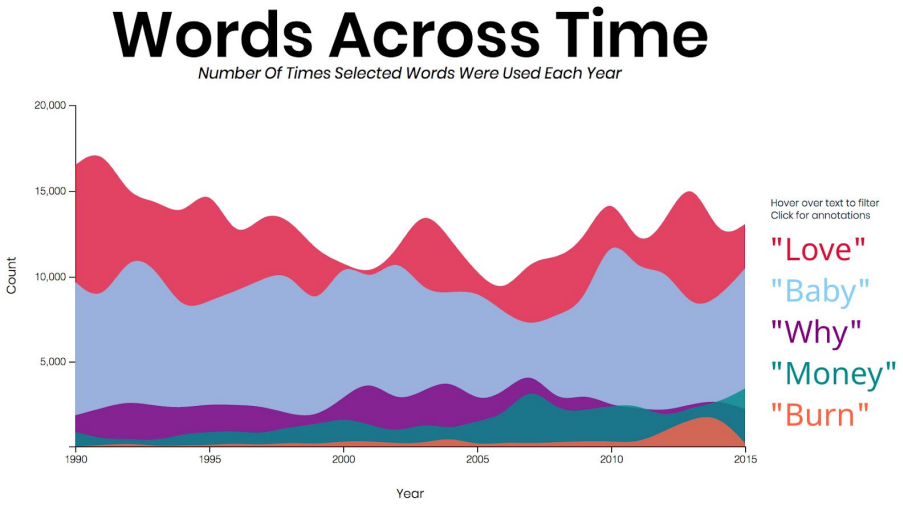
The improved visualization shows songs as circles and layers each song through partial opacity. The visualization features a re-worked tooltip and a legend that disappears when songs are being hovered over to reduce distraction.

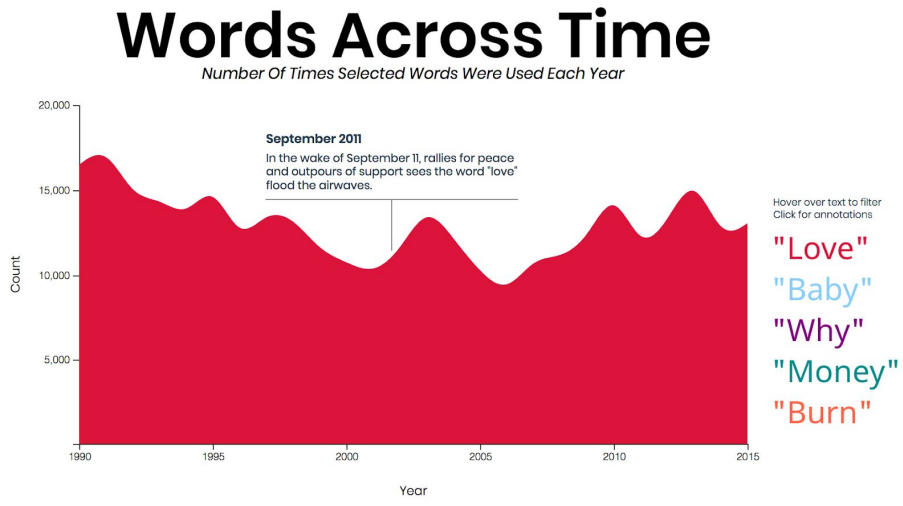
Visualization 3

Our initial visualization featured 10 selected works and their usage over the 25 years of data. We would whittle down the number of words visualized in our improved version. It also featured zooming (which was distracting and not immediately useful) and a lot of very ugly colors and transparencies.



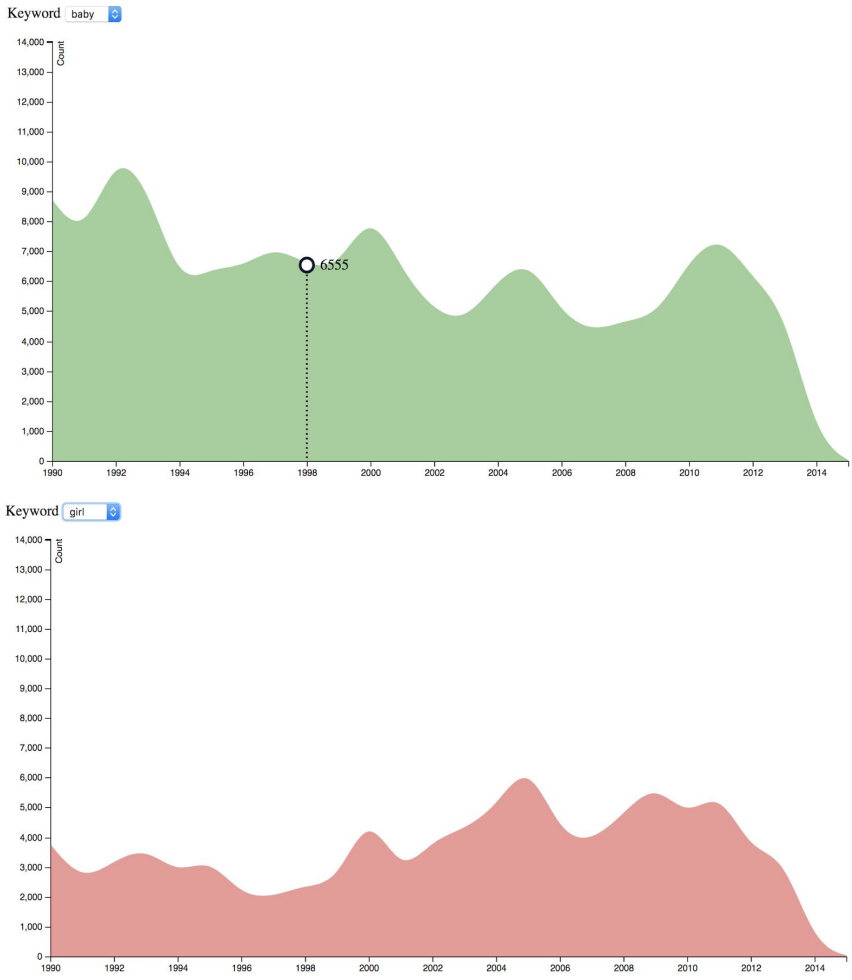
Here, we focused simplifying the visualization and making it more intuitive. We also added annotations to the graph to highlight points in time we thought to be more interesting.





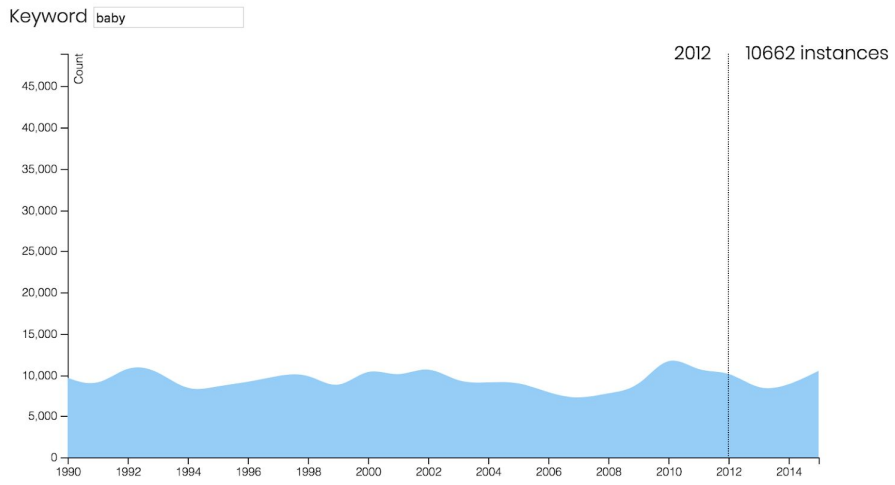
Visualization 4

This visualization featured a close up view of different words and their occurrences over time with a dropdown to select different words. It was originally constructed using the same data as in viz 3, which featured only 10 selected words.



We also added an input search field that allows for a word search that has expanded beyond the original 10 words to the entire dataset. This visualization loads multiple smaller csv files in the background since our original full dataset was too large. The hover over with emphasis on a point was changed to be a vertical line.

The axes also changed to scale with each searched word, because we don't need to compare the result across all words which had a large range of values For example, "and" had several tens of thousands of occurrences, and using the same fixed scale for "burn" would make burn look flat.



Evaluation

The visualizations work as intended, but given more time we would have loved to add sentiment analysis on the songs and see if the sentiments change over time or by artist. Also the data didn't provide any information on the genre data, which could be a different way to segment and look at the data.

Also from the data, top artists don't always have a linear or steady rise in the charts to the top. The number of unique words hasn't changed much, but more recent songs have appeared on the chart more frequently.

We also didn't answer some of our original questions from our proposal, regarding seasonality or seasonal trends of songs on the chart, for example, maybe some songs or lyrics are more prevalent in the summer.

Task List

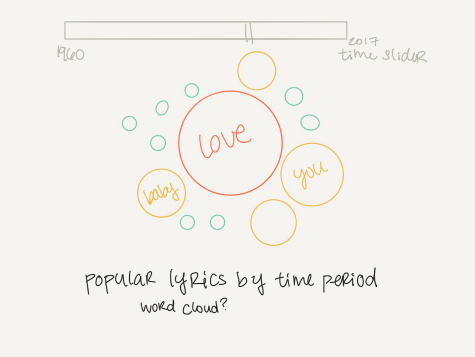
https://trello.com/b/weD3jvfl/final-project-site

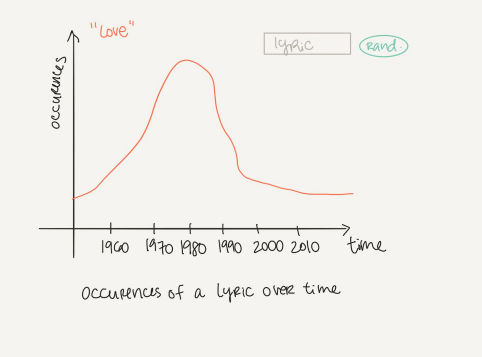
Informal Progress Journalling

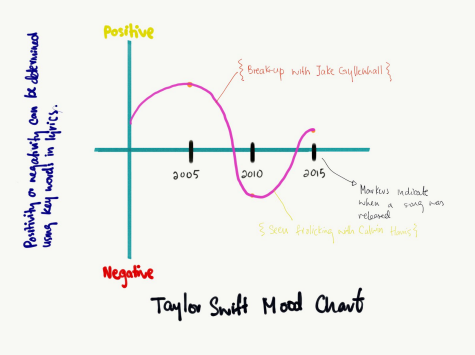
2018.04.06

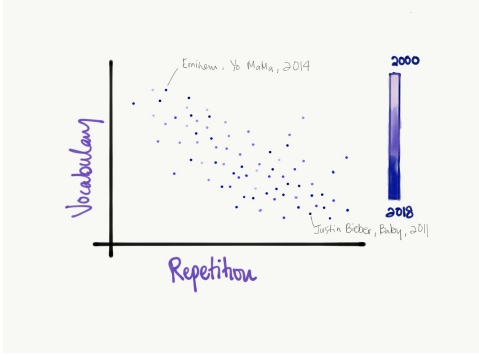
Formation of Storyboard

- Selected topic: How music lyrics has changed over time
    - Other topics considered: Authors and their language characteristics, kpop trends, fountain pen usage and calligraphy hobbies
- Determined where to retrieve data
- Came up with preliminary visualisation drawings and deadlines for work





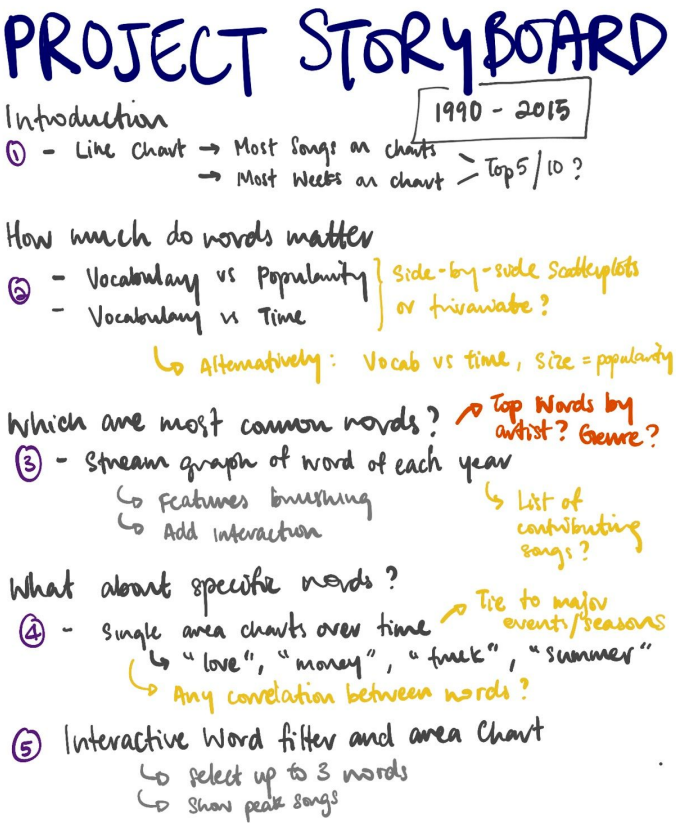




2018.04.09

Storyboard Revision

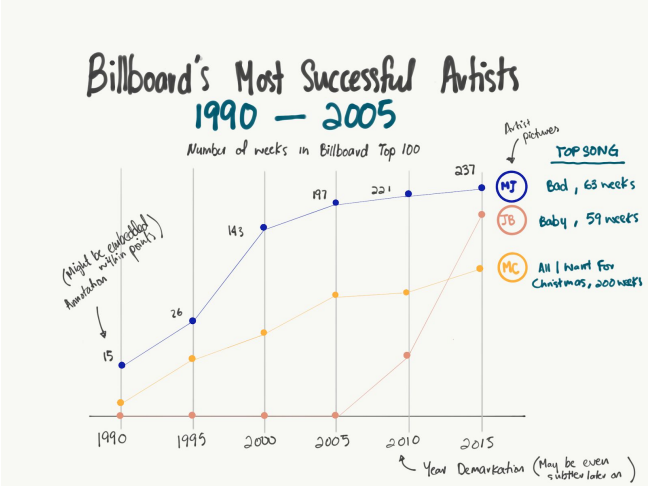
- Narrowed scope of project and focused on key goals
- Developed general ideas for how we wanted visualizations to look and improved on sketches from the project proposal
- Further solidified project timelines

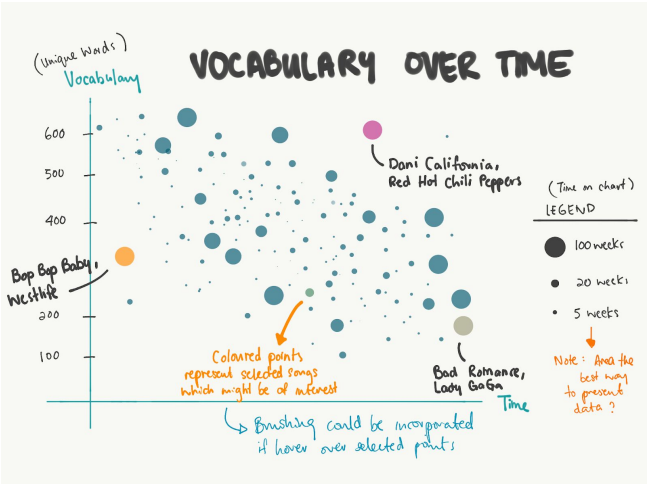


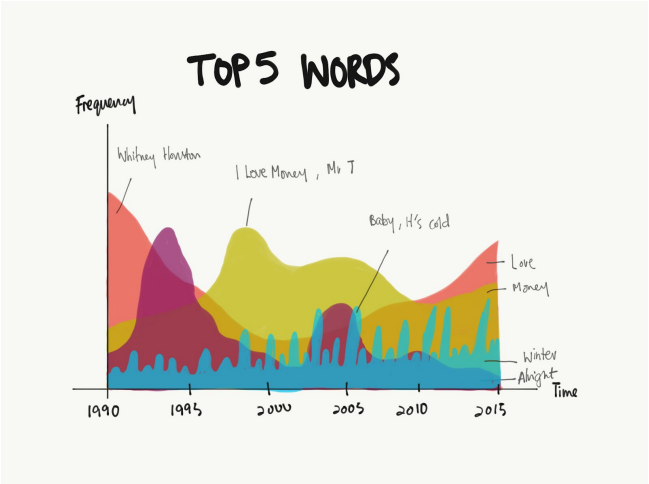
2018.04.11

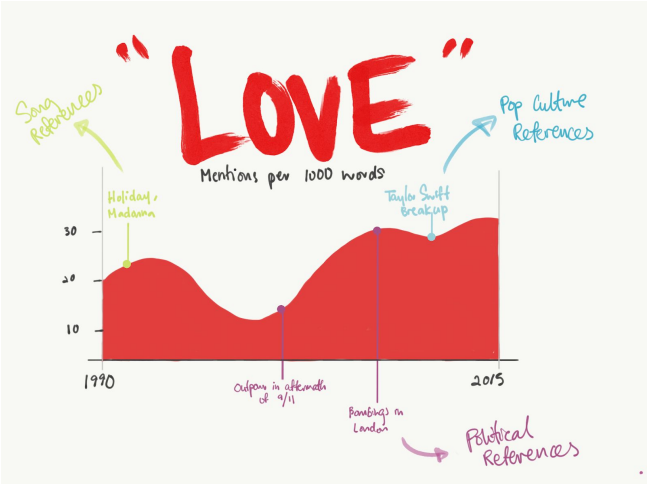
Improved Sketches

- Greater focus in storytelling allowed us to have a clearer idea of what kind of information we wanted to convey with our visualisations
- Visualisations themselves are more focused and have general interactivity and storytelling elements included in design





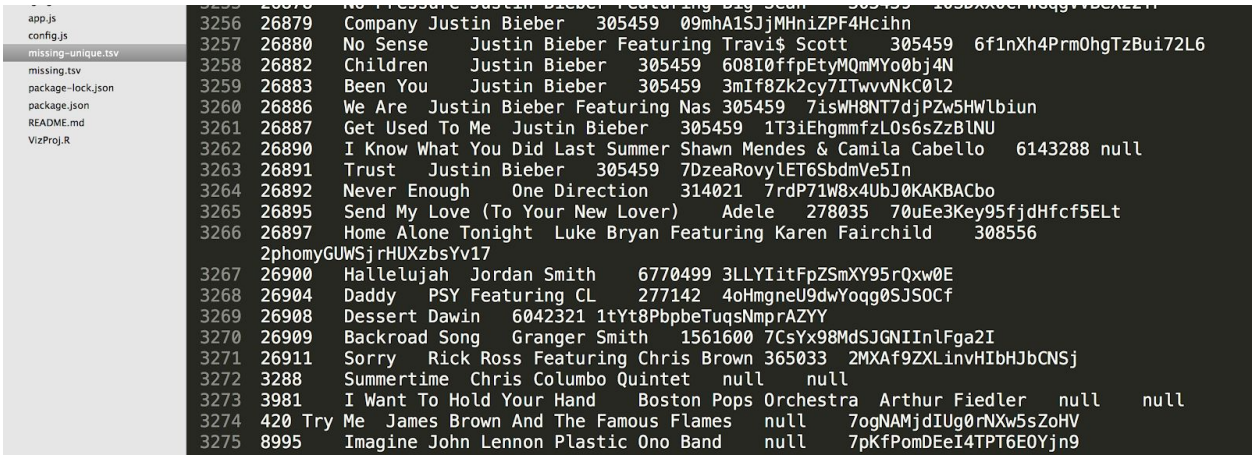




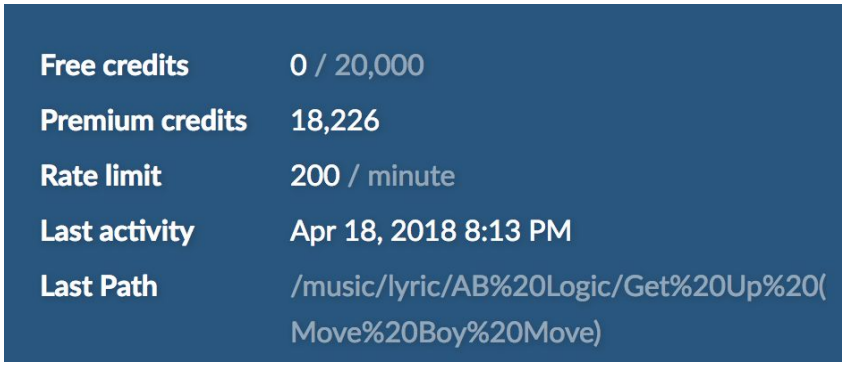
2018.04.18

Data Scraping

- Last pass on api crawling on songs with missing lyrics. If they're not found here, then they're not in that specific lyrics API



- Has 3000+ "missing records" - hoping to put a small dent in this and leave the node js app running for a while until it gets through all the API calls
- Might as well make good use of our api credits



Data Cleaning

- Cleaning the data to create dataset optimised for each visualisation
    - Each visualisation requires inputs and ranges
    - Visualisation #1 requires both feature sums and weekly data, however, visualisation #2 requires different cumulative sums of unique words by songs

```
61
62  setwd("Billboard 1990 - 2015/") # set wd
63  files <- Sys.glob("*.json") # get files with same extension
64
65  df <- as.character(data.frame()) # data frame for all longs
66  for (f in 1:length(files)) { # loop through each file
67    obj <- fromJSON(file = files[f])
68    date <- str_replace_all(files[f], ".json", "") # remove extension and get date
69    print(date) # check for progress
70    songs <- obj$songs
71    if (length(songs) == 0) {
72      print(paste("ERROR: ", date))
73    }
74    else {
75      for (i in 1:length(songs)) { # loop through songs in each list
76        if (length(songs) != 100) { # check which length not 100
77          # print(paste("NOT 100 SONGS: ", date))
78        }
79        song <- songs[[i]]
80        song.vect <- c(date, song$rank, song$song_id, song$song_name, song$display_artist)
81        df <- rbind(df, song.vect)
82      }
83    }
84  }
85  colnames(df) <- c("Date", "Rank", "Song_id", "Song_Name", "Artist") # set column names
86  rownames(df) <- NULL # get rid of row names
87  df <- as.data.frame(df) # coerce to dataframe
88
89  df$Date <- as.Date(df$Date, "%Y-%m-%d")
90  df <- df[order(as.Date(df$Date, format = "%Y-%m-%d")),] # order by date
91  write.csv(df, "masterBillboard.csv") # write data frame to csv
92
93  # song.name.table <- as.data.frame(table(df$`Song Name`))
94  artist.table <- as.data.frame(table(df$Artist))
95  colnames(artist.table) <- c("Artist", "Freq")
96  artist.table <- artist.table[order(-artist.table$Freq),] # order by frequency
97  top5.artists <- as.vector(artist.table$Artist[1:5]) # top 5 artists
98  top5.artists.df <- subset(df, df$Artist == top5.artists)
```

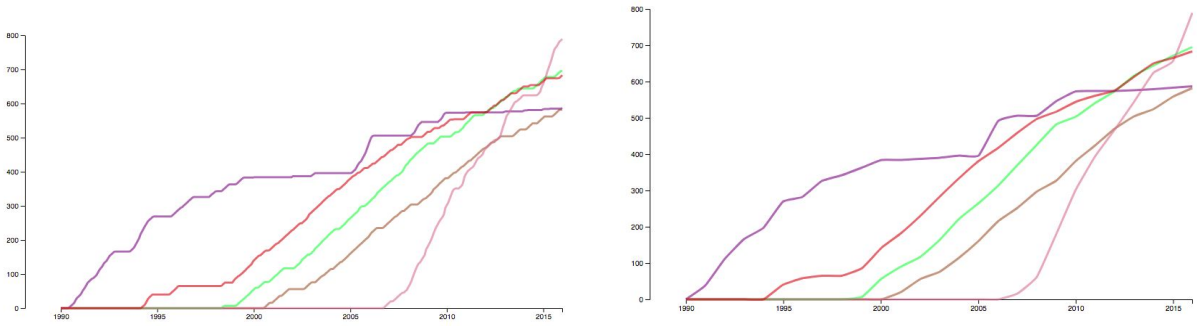
song_id	song_name	artist	lyrics	num_apps	vocab	first_app
25606	Radioactive	Imagine Dragons	I'm waking up to ash and dust I wipe my brow and I sw...	87	62	2012-08-18
25214	Sail	AWOLNATION	Sail! This is how I show my love I made it in my mind b...	79	43	2011-09-03
23693	I'm Yours	Jason Mraz	Well you dawned on me and you bet I felt it I tried to b...	76	153	2008-05-03
24922	Party Rock Anthem	LMFAO Featuring Lauren Bennett & GoonRock	Party rock! Yeah! Whoo! Let's go! Party rock is in the h...	68	141	2011-02-12
25911	Counting Stars	OneRepublic	Lately, I've been, I've been losing sleep Dreaming abou...	68	98	2013-07-06
19966	How Do I Live	LeAnn Rimes	How do I get through a night without you If I had to liv...	65	60	1997-06-21
24887	Rolling In The Deep	Adele	There's a fire starting in my heart Reaching a fever pitc...	65	120	2010-12-25
23092	Before He Cheats	Carrie Underwood	Right now, he's probably slow dancing With a bleached...	64	104	2006-09-16
19755	Foolish Games/You Were Meant For Me	Jewel	I hear the clock, it's six a.m. I feel so far from where I'v...	61	167	1996-11-30
25762	Demons	Imagine Dragons	When the days are cold And the cards all fold And the ...	61	109	2013-01-26
19364	Macarena (Bayside Boys Mix)	Los Del Rio	Dale a tu cuerpo alegria Macarena Que tu cuerpo es pa...	60	53	1995-09-02
24251	Need You Now	Lady Antebellum	Picture perfect memories, Scattered all around the floo...	60	78	2009-08-29
25390	Somebody That I Used To Know	Gotye Featuring Kimbra	Now and then I think of when we were together Like w...	59	113	2012-01-21
26001	All Of Me	John Legend	What would I do without your smart mouth Drawing m...	59	104	2013-09-21
22931	How To Save A Life	The Fray	Step one you say we need to talk He walks you say sit ...	58	118	2006-04-15
20813	Higher	Creed	When dreaming I'm guided through another world Tim...	57	99	1999-09-11
23882	Use Somebody	Kings Of Leon	I've been roaming around Always looking down at all I ...	57	61	2008-10-11
25194	Lights	Ellie Goulding	I had a way then losing it all on my own I had a heart t...	57	81	2011-08-20
20984	The Way You Love Me	Faith Hill	If I could grant, you one wish I wish you could see the ...	56	63	2000-03-11
24182	I Gotta Feeling	The Black Eyed Peas	[Hook] I got a feeling that tonight's gonna be a good ni...	56	115	2009-06-27
19343	Missing	Everything But The Girl	I step off the train I'm walking down your street again ...	55	90	1995-08-12
20728	Amazed	Lonestar	Everytime our eyes meet This feeling inside me Is almo...	55	88	1999-06-05
21927	Unwell	matchbox twenty	All day staring at the ceiling Making friends with shado...	54	107	2003-03-22
24310	Hey, Soul Sister	Train	Hey, hey, hey Your lipstick stains On the front lobe of ...	54	111	2009-10-17
25618	Cruise	Florida Georgia Line	Baby you a song You make me wanna roll my windows...	54	134	2012-09-01
19760	Barely Breathing	Duncan Sheik	[Verse 1] I know what you're doing, I see it all too clear...	53	139	1996-11-30
20202	Too Close	Next	I wonder if she could tell I'm hard right now, hmmm Y...	53	104	1998-02-14

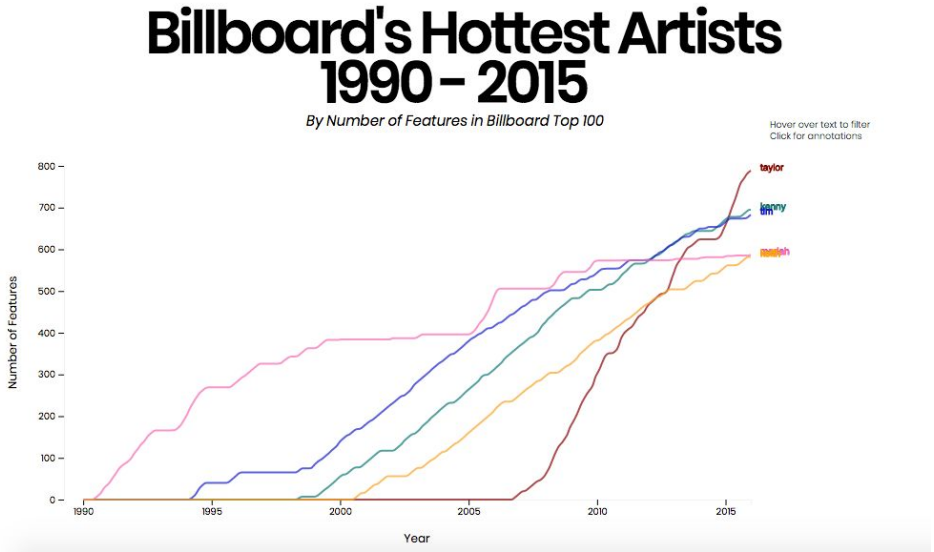
2018.04.19

Visualisation #1 Drawing and Testing

- Transplanted preliminary design of visualisation #1 into D3
- Realised that there were some design choices that had to be made
    - Using weekly and monthly data would mean lines would not be smooth even after implementation of d3 curves
    - 2015 is not a decisive end date, since the start of 2015 is not the same as the end of 2015 on an axis, so the axes would need to be redesigned
    - Labels overlap when drawn in the same function, so they will have to be drawn individually

Below, the difference between points denoted by month (left) and year (right), and the overlapping of labels. This is drawn pre-interpolation.





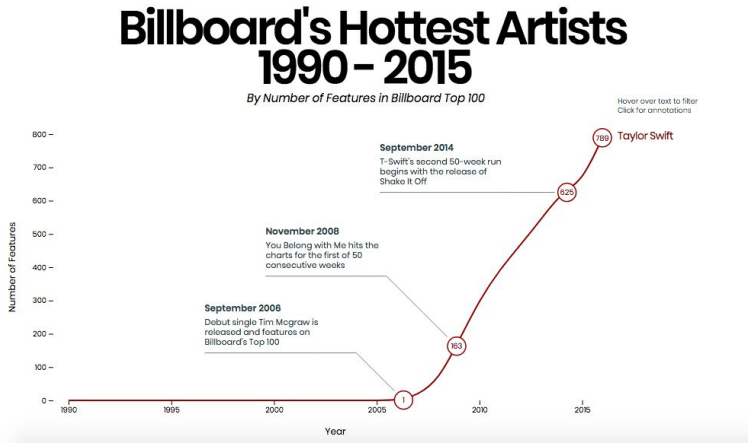
2018.04.20

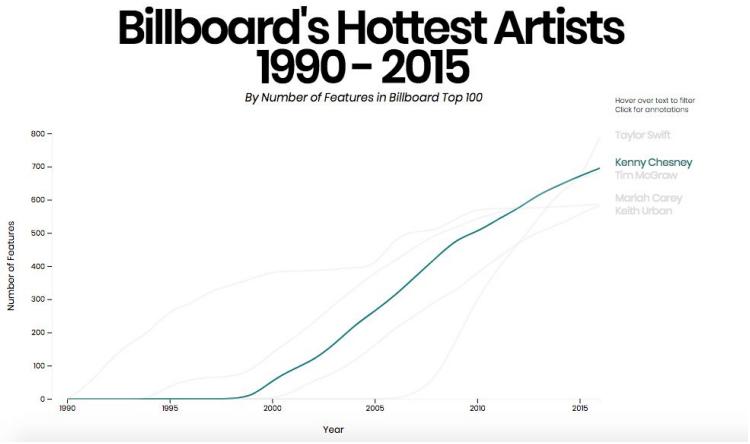
Visualisation #1 Drawings, Annotations, and Interaction

- Completed the rest of Visualisation #1
    - Implemented manual labeling of each line
    - Fixed fonts and added curvature to lines
    - Added annotation with d3-annotations API found here
    - Added line markers and numbering with simplicity in mind
    - Added hover and clicking interaction for brushing and annotation

Designs may be tweaked after further testing.

Below are screenshots of the visualisation. Annotations on click (top) and brushing on hover of the labels (bottom).

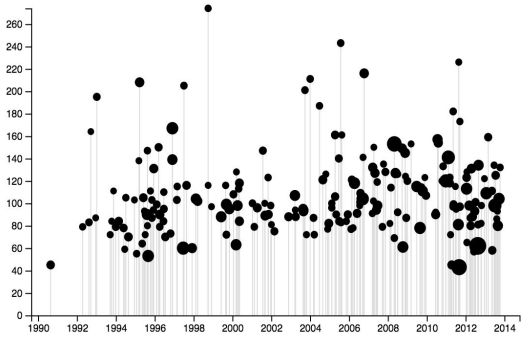


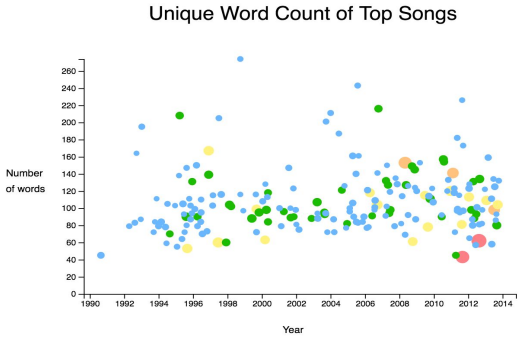


2018.04.26

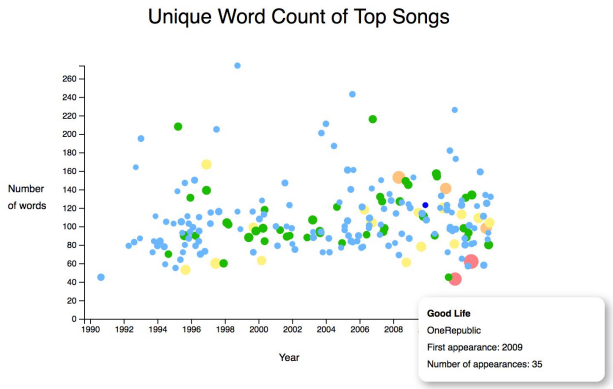
Visualization #2 Initial Drawings

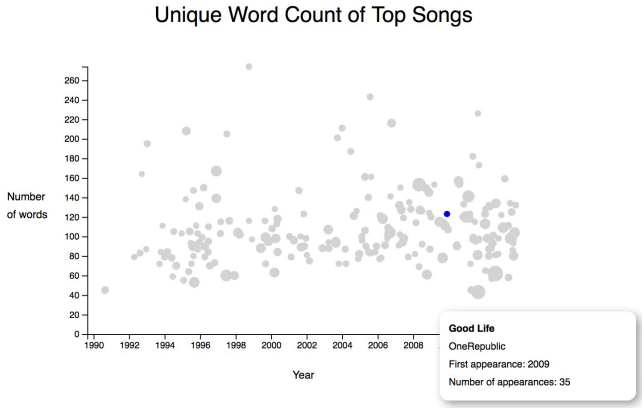
- Initial scatterplot from draft, based on an example from the textbook



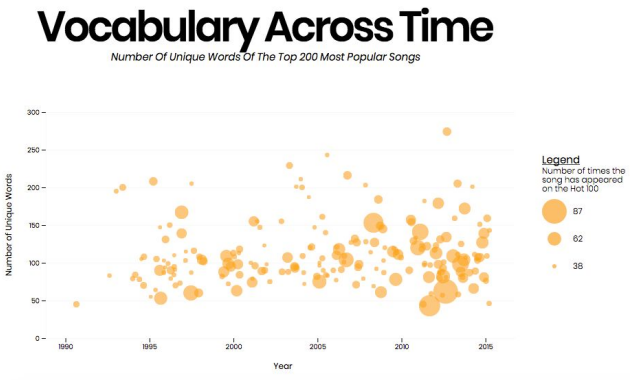


- Removed lines and added color, axes labels, and chart title for the next iteration





- Started with svg tooltips for testing, then switched to div tooltips; also implemented brushing
- Since color wasn't as effective as size for denoting the frequency of appearance, we decided to use only one color for all songs
- As a result, we made the circles partially transparent to make it easier to differentiate each circle in lieu of color



2018.04.29

Putting everything from beta together and getting missing lyrics

- Used lyric-api to look for missing lyrics from our list
    - Took two passes, one with no “correction” for titles and artist names
    - Second pass omitting things in parentheses in titles, and featuring artists in search
- Using scrollama for scrollytelling
    - Added each vis from beta into the main index, got each one to load on demand when scrolled

Code for new endpoint to get only missings songs. Songs are saved to data/missing/..

```
app.get('/lgget', function(req, res) {
    fs.readFile('missing-unique-2.tsv', 'utf8', function(err, data) {
        if (err) throw err;
        var data = tsv.parse(data);
        for (var i = 0; i < data.length - 1; i++) {
            var songName = String(data[i].song_name);
            songName = songName.replace(/\(.+\)/, "");
            var artistName = String(data[i].display_artist);
            artistName = artistName.replace(/ Featuring .+/,"");
            var songfilename = "data/missing/" + data[i].song_id + "-lyrics.json";
            var songObject = {
                "song_id": data[i].song_id,
                "song_name": data[i].song_name,
                "artist_id": data[i].artist_id,
                "display_artist": data[i].display_artist,
                "spotify_id": data[i].spotify_id
            }
            lgget(songName, artistName, songfilename, songObject, i);
        };

    });
});
```

After running the script twice, the end result is 2138 new song lyrics

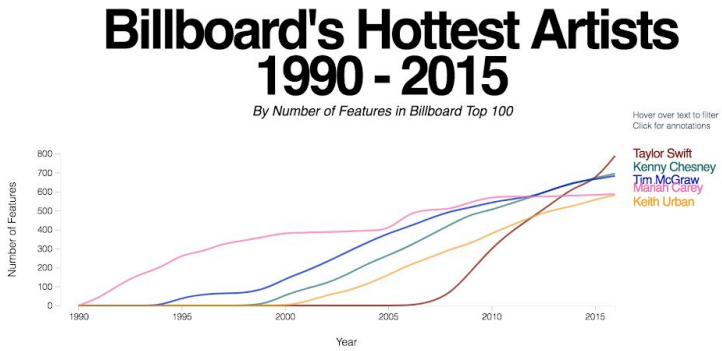
```
[Bobot:missing allisonw$ ls | wc -l
     1586
[Bobot:missing allisonw$ ls | wc -l
     1588
[Bobot:missing allisonw$ ls | wc -l
     1806
[Bobot:missing allisonw$ ls | wc -l
     2138
[Bobot:missing allisonw$ ls | wc -l
     2138
 Bobot:missing allisonw$
```

The screenshot below shows the end of a file with data of songs with lyrics that are still missing from our dataset. For a total of ~800 something

```
791  26678   This Summer's Gonna Hurt... Maroon 5    309663  73h73Cm8SZRyyoyEsnd9SP
792  26683   A Thousand Years    Sawyer Fredericks   6489732 1xbtFabmTCcxN0gozYj2AZ
793  26685   Please  Sawyer Fredericks   6489732 2pTnrrRF6creKP8lVDpgqw
794  26687   Old Man Sawyer Fredericks   6489732 2Yh8cnpddujSLPOVJKlSDt
795  26703   Ulay Oh How I Became The Bomb   6591067 2bxBdzCkGpWw0oXWEKMXJV
796  26706   Hood Go Crazy   Tech N9Ne Featuring 2 Chainz & B.o.B    371716  0kIHfcHGFGm5xb0dSlKEdp
797  26715   B**** I'm Madonna   Madonna Featuring Nicki Minaj   308786  3Ik6ZVvOgr2Pu5cnI8Gsu0
798  26723   Yoga    Janelle Monae & Jidenna 304187  null
799  26825   Runnin' (Lose It All)   Naughty Boy Featuring Beyonce & Arrow Benjamin  312065
     0H2BuFu2MuJfnF1yHKjBcp
800  26911   Sorry   Rick Ross Featuring Chris Brown 365033  2MXAf9ZXLinvHIbHJbCNSj
801  3288    Summertime  Chris Columbo Quintet   null    null
802  3981    I Want To Hold Your Hand    Boston Pops Orchestra  Arthur Fiedler   null    null
803  420 Try Me  James Brown And The Famous Flames   null    7ogNAMjdIUg0rNXw5sZoHV
804  8995    Imagine John Lennon Plastic Ono Band    null    7pKfPomDEeI4TPT6EOYjn9
```

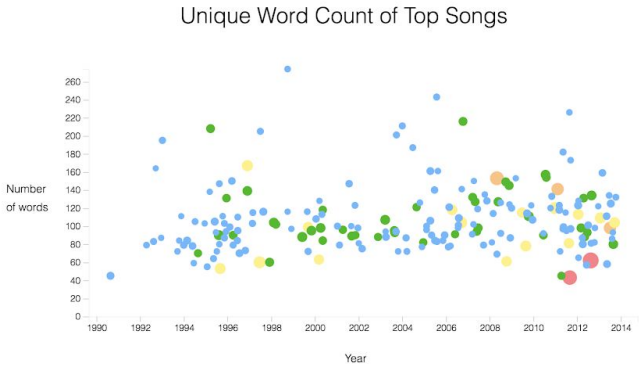
What the current website looks like with placeholders:

STEP 1



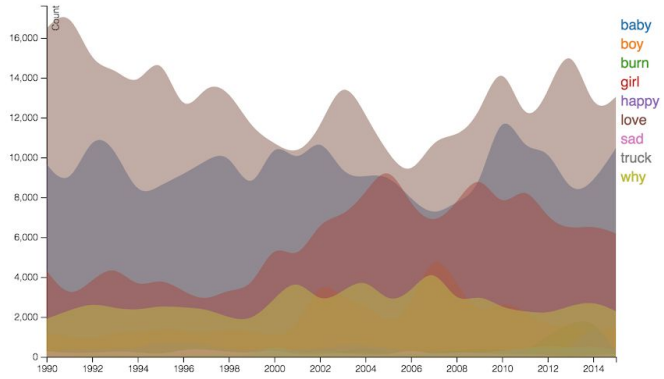
STEP 2

STEP 2



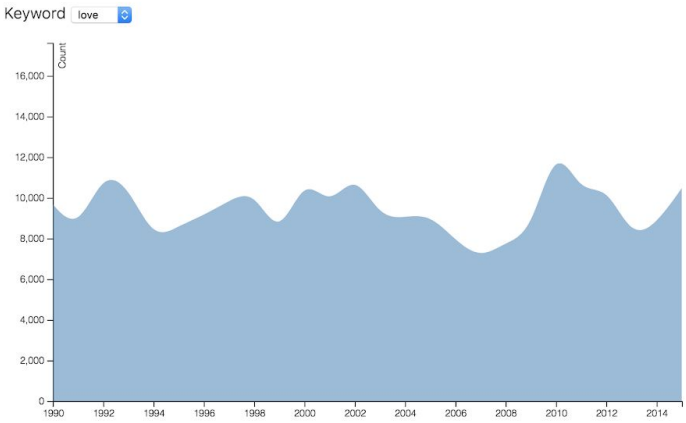
STEP 3

STEP 3



STEP 4

STEP 4



2018.04.30

Completed story draft 1

- Drafted story detailing our motivations, thought process, and explanations of visualizations

Completed dataset for Visualisation #4

- Completed and compiled dataset of 46198 unique words and their frequencies across all years

year	num.unique.words		a	after	and	any	chorus	dont	holla	i	like	nasty	now	pretending	thats	uhhuh	yo	you	$$$	$1	$100	$1000	$10k	$20	$200	$20000	$4000	$450	$50	$7500	$hort	$ign	$money	+
1990	4349	0	0	0	0	0	4	16	0	0	4	0	0	20	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0
1991	8251	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0
1992	4833	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
1993	5077	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	30	0	0	0	0	0	0	4	0	0	0	0	0	0	0	253
1994	5468	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	60	0	3	0	0	0	0	0	0	0	0	0	3	0	0	44
1995	6014	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	19
1996	7531	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75
1997	4808	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0
1998	4175	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	13	0	0	12
1999	3527	5	0	0	0	0	0	0	0	0	0	15	20	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	50
2000	4670	1	0	0	0	0	2	0	0	2	0	0	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
2001	4603	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	119
2002	6683	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74
2003	6161	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	291
2004	5614	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	0	163
2005	6634	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	20	129
2006	6719	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86
2007	6265	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36
2008	6008	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	206
2009	4900	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77
2010	4854	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	38
2011	3246	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	9	0	0	63
2012	1770	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0
2013	1113	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2014	992	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	72
2015	1441	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	0	0	26	0	0	0	7	12	0	0	66	0	0

2018.05.01

Amended story draft

- Decided to amend the tone of the story to make it more lighthearted and fun

Title: THE HOTTEST OF THE HOT 100

Subtitle: A closer look at Billboard's Hot 100 over the years

The Billboard Hot 100 is the music industry standard record chart in the United States for singles, where chart rankings are based on sales, radio plays, and online streaming. It has been around since the August 4, 1958, and the first song to be in the coveted number one spot was Ricky Nelson's "Poor Little Fool". The current number one song is "Nice for What" by Drake.

Cue Dramatic Music

Narrator with deep radio voice:

Like all things that have stood the test of time, the Billboard Hot 100 has stories to tell. It has seen the coming of generational talents, the rise and fall of one-hit wonders, the soundtrack of pivotal socio-political movements, and the timeless legacy of icons. The Hot 100 tells the story of the human experience, because it is less about how we found the songs we love, and more about how the songs we love found us. It tells the story of change, and it tells the story of constants. So with the data that we've compiled of Billboard's weekly Hot 100 charts from 1990 to 2015, and the lyrics of all songs that featured on the chart in between that time period, we hope to tell that very story.

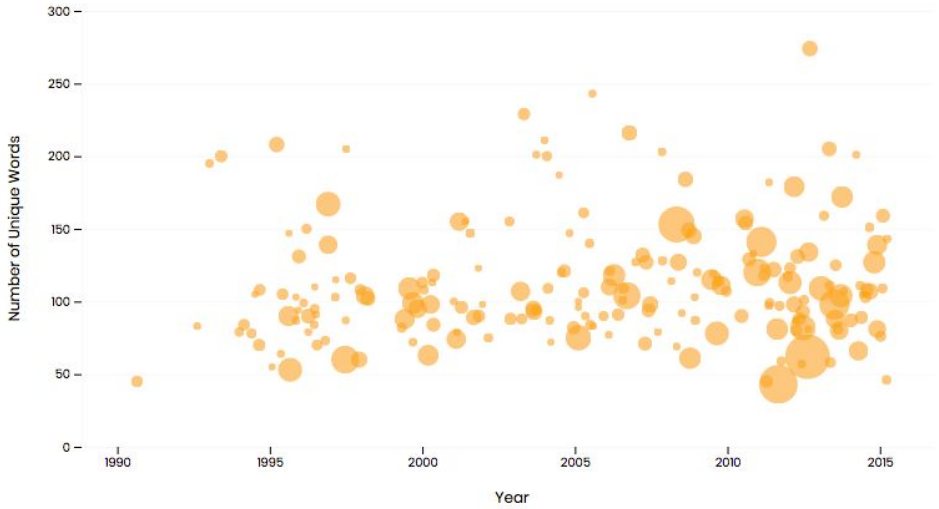
Music Fades Out

2018.05.03

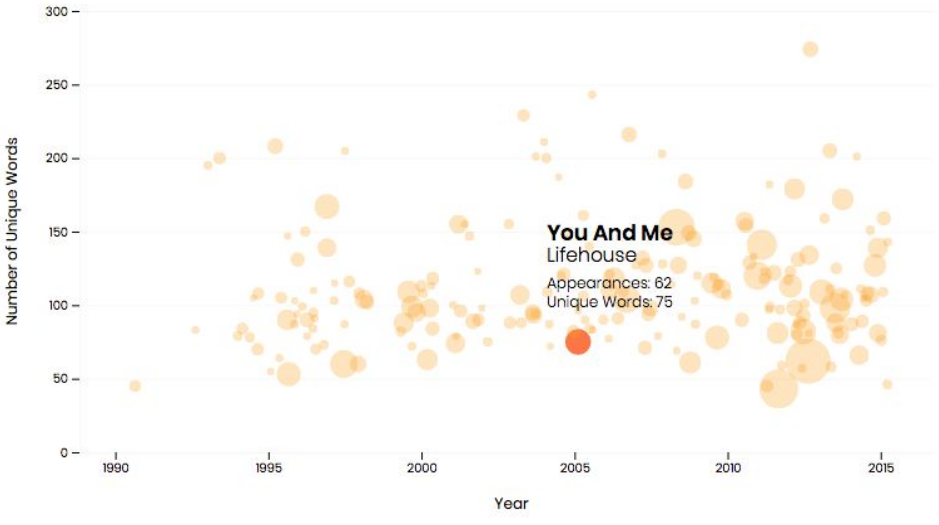
Visualization #2 Revision

- Updated the data to take out redundant columns and speed up loading
- Adjusted the tooltips so they didn't obstruct the view of the other points
- Adjusted the axes and added grid-lines
- Used radius of circles to denote frequency instead of using both colour and size
- Updated fonts to standardize with other visualization
- Updated mouseover and mouseout interaction

Unique Word Count of Top Songs



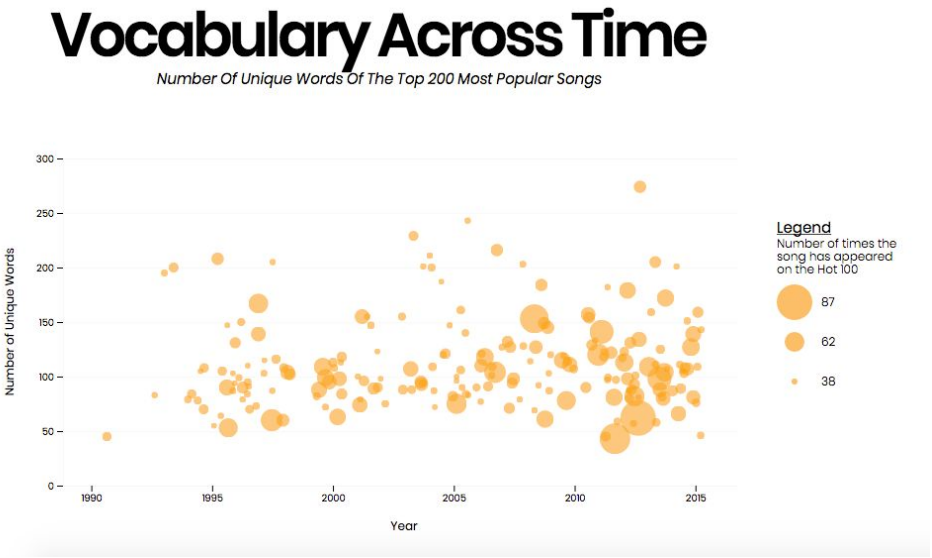
Unique Word Count of Top Songs

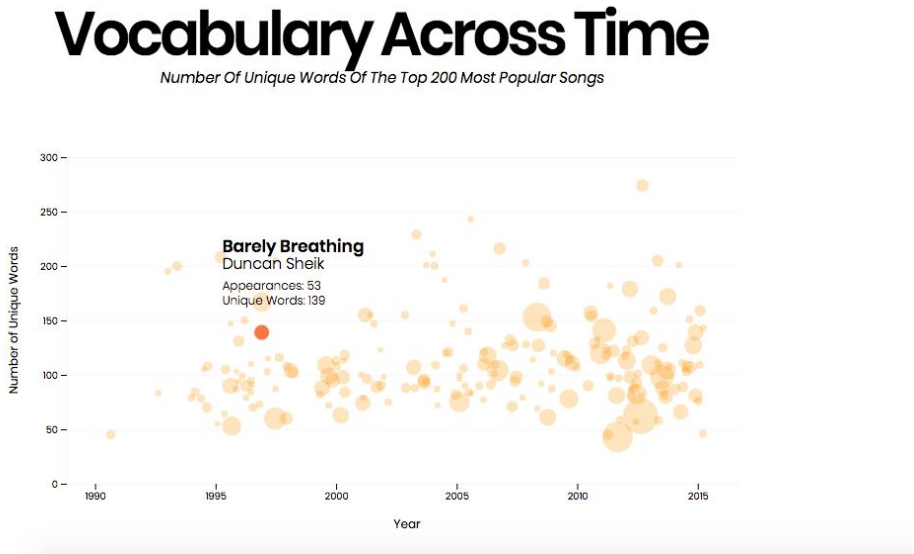


2018.05.05

Visualization #2 Final Version

- Titles, legends, interaction and instructions are finalised
- Fonts are standardized
- Tested on website

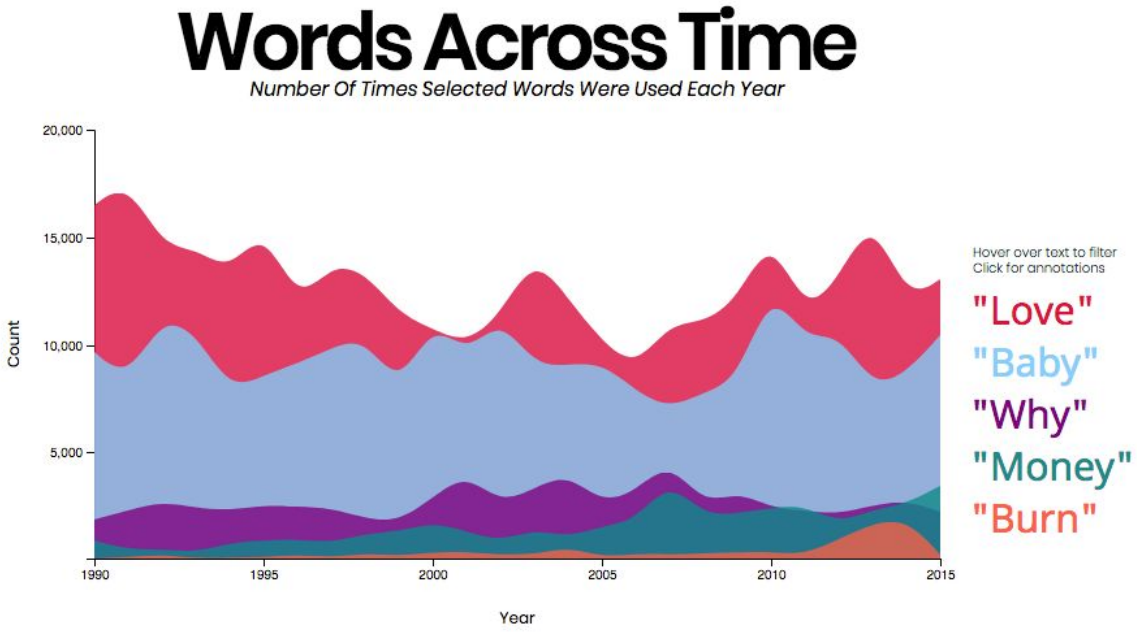


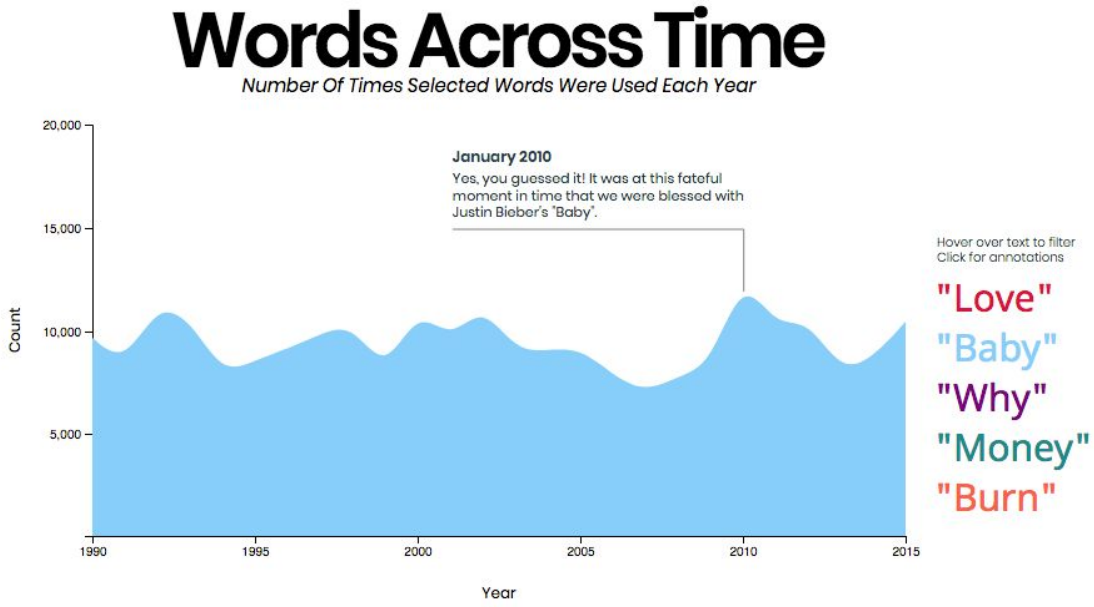


2018.05.06

Visualization #3 Final Version

- Number of selected words narrowed down for focus
- Colour palette has been amended
- Fonts standardized





2018.05.07

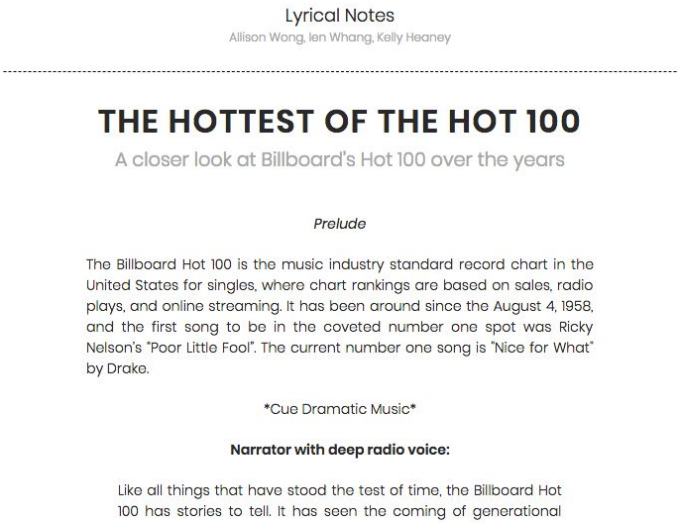
Visualization #4 Final Version

- Final Version of visualization optimised by splitting up csv file into 6 parts which are alphabetically sorted
- Hover-over labels are adjusted
- Axis amended to be interactive
    - Thought process was that since we were not comparing the usage of words between each other, an updating scale would be helpful in providing readability and will not be misrepresenting the data

2018.05.10

Final tweaks to website

- Tweaked minor aesthetics (font sizes and spacings) for final website
- Adjusted dimensions of visualizations



What we did find interesting, however, was that the songs on the higher end of thevocabulary spectrum are songs that have lengthy rap sections like Thrift Shop (in the upper right hand corner), Gold Digger, and Get Low. Hover over each circle on the chart to see that song's title, artist, vocabulary count, and number of appearances on the Hot 100.

Following that train of thought, we wanted to visualise the usage of specific words over time with the idea that the words used might give us an insight into the pop culture. So we modified our data, selected some of the most frequently used words, charted them, and was rewarded with some interesting results.