Lyric Analysis – King of Pop

ANALYISING MICHAEL JACKSON'S
DISCOGRAPHY USING NLP

Nabih Moukayed | Module: Introduction to Natural Language Processing | 08/05/2019

Introduction

This paper will provide insight into the data cleaning, exploration and analysis that I have conducted whilst dealing with Michael Jackson's lyrics. The program used for this mini-project is R due to the breadth of packages that allows swift cleaning and analysing of data. The objective of this project is to create a clean dataset containing Michael Jackson's lyrics that provides certain traits and attributes that would answer multiple questions. What makes a song a hit? What is the general feeling of a song? Do majority of the public enjoy happy or sad songs? And so on.

Web Scraping in R

The starting point in any data science project is having a dataset, it is the foundation that determines the complexity and outcome of given data. I decided to create my own dataset by scraping using the rvest package in R. I was able to web scrape a table from Wikipedia containing information on all the songs Michael Jackson has recorded along with corresponding release dates and albums. https://en.wikipedia.org/wiki/List_of_songs_recorded_by_Michael_Jackson.

This followed with modifying the dataset by dropping unnecessary columns (Ref,), patterns (") and adding a new column (chart_level) indicating whether a song reached the Billboard Top 10 or not. This was done by referring to the following: https://www.billboard.com/music/michael-jackson/chart-history/hot-100/3, those that did not make the Top 10 on the billboard chart were set to 'Uncharted'. The reason behind this will be explained further on in this report.

Once the songs have their chart levels assigned, the dataset is saved onto the local environment as 'MJ.csv'. Now that I have all of Michael's song data from Wikipedia, I still require his lyrics. Spending hours online, I was able to find a solution to scrape his lyrics all at once. This was done by scraping the following site http://www.songlyrics.com/michael-jackson-lyrics/ with a ten second pause between each song while scraping as I did not want my IP address to be banned from the website. The site had data on lyrics to 1165 of Michael Jackson's (it is worth to note that a lot of songs on that site are remixes, a Capella's and other variant versions of his original songs).

Saving the lyric dataset onto the local environment as 'lyrics.csv', I realise a formatting issue with the names of the songs (i.e. Man in the Mirror would be _man_in_the_mirror) and I could not merge both datasets to include lyrics from lyrics.csv to MJ.csv. After desperately finding ways to fix this issue, I decided to manually input the lyrics to match the corresponding song title in 'MJ.csv'. Once that was complete I named the new csv file 'final-MJ.csv'.

Reading & Modifying the data

Now that 'final-MJ.csv' is saved onto the system, Let's have a look and see what we can do to enhance the dataset in order to retrieve quality information.

```
> names(MJ_read)
[1] "X"           "Song"        "Year"        "Album"
[5] "Writer.s."   "Notes"       "Peak"        "chart_level"
[9] "lyrics"
```

This shows what columns are included in the dataset, looks like we don't need some of them so let us go ahead and drop "Writer.s.", "X" and "Notes".

```
> final_MJ <- MJ_read %>%
+   select(Song, lyrics,Year, Album, Peak, chart_level)
> names(final_MJ)
[1] "Song"        "lyrics"      "Year"        "Album"
[5] "Peak"        "chart_level"
```

Now that we have all the necessary columns let us go ahead and have a glimpse into the dataset and see how much information we have to deal with.

```
> glimpse(final_MJ[1,])
Observations: 1
Variables: 6
$ Song        <chr> "Off the Wall"
$ lyrics      <chr> "When the world is on your shoulder\nG...
$ Year        <int> 1979
$ Album       <chr> "Off the Wall"
$ Peak        <int> 10
$ chart_level <chr> "Uncharted"
```

```
> dim(final_MJ)
[1] 209   6
```

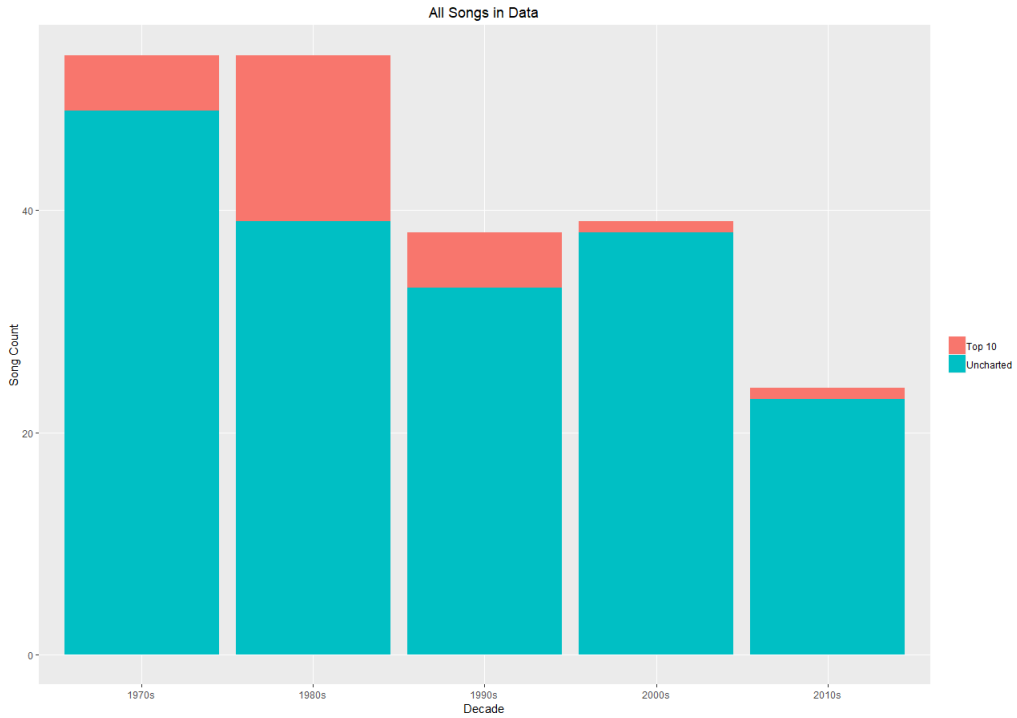
This shows us that the dimensions of the data set is 209 observations (rows) by 6 variables (columns). You might be wondering why the low number of observations especially that I was able to scrape 1165 songs. That is because a lot of songs that were scraped from https://www.azlyrics.com were alternative versions of the original song, furthermore I am only looking into songs sung primarily by Michael Jackson and not others.

Data Conditioning

```
> str(final_MJ[1,]$lyrics, nchar.max = 500)
 chr "When the world is on your shoulder\nGotta straighten up your
 act and boogie down\nIf you can't hang with the feeling\nThen the
re ain't no room for you, this part of town\n'Cause we're in the p
arty\nPeople, night and day\nLivin' crazy that's the only way\nSo
tonight gotta leave that nine to five upon the shelf\nAnd just enj
oy yourself\nGroove, let the madness in the music get to you\nLife
 ain't so bad at all if you live it off the wall\nLife ain't so ba
d at all\n (Live life off t"| __truncated__
```

The above command shows us that a lot of lyrics that were obtained include abbreviations or contractions. As this project includes lyrical analysis, such anomalies cannot exist in the corpus. Contractions such as ain't and shan't do not provide significant information. Splitting such contractions into two words provide a much clearer word count and can be associated with certain moods and sentiments. Furthermore, special characters such as numbers and punctuation can also create distortion when analysing the data therefore they also must be removed. Finally, in order to analyse the data efficiently, I have applied a lowercase argument to all lyrics in the dataset.

Applying the mutate function to create a decade column by clumping certain years together provides information on Jackson's musical activity and success.



Looks like the 80's spawned the highest number of chart-topping songs for the King of Pop. Now let us look at songs that reached number 1 on the billboard chart. A neat visual table (or should I say Kable) is created using the kableExtra package.

Michael Jackson's No. 1 Songs

Year	Song	Peak
1972	Ben	1
1979	Don't Stop 'Til You Get Enough	1
1979	Rock with You	1
1982	Beat It	1
1982	Billie Jean	1
1985	We Are the World	1
1987	Bad	1
1987	Dirty Diana	1
1987	I Just Can't Stop Loving You	1
1987	Man in the Mirror	1
1987	The Way You Make Me Feel	1
1991	Black or White	1
1995	You Are Not Alone	1

Text Mining/Text Analytics

With lyrics comes certain superfluous words that are not necessarily beneficial to our analysis. Removing such terms such as 'chorus', 'rap', 'michael' and so on cleans up the data in order for us to analyse correctly.

Moving onto analysing the lyrics, I start off by creating a tidytext version of the current dataset 'MJ_filter'. This includes tokenizing the lyrics using the function 'unnest_tokens', removing stop, duplicate, superfluous words as well as words that have less than four characters (as they could be quite repetitive and distort our data. After all, MJ is notoriously known for belting out onomatopoeic sounds. Hee!).

The following table shows which songs have the word 'world' included in them as well as the year they were released, peaks and chart level.

Tokenized Example - world

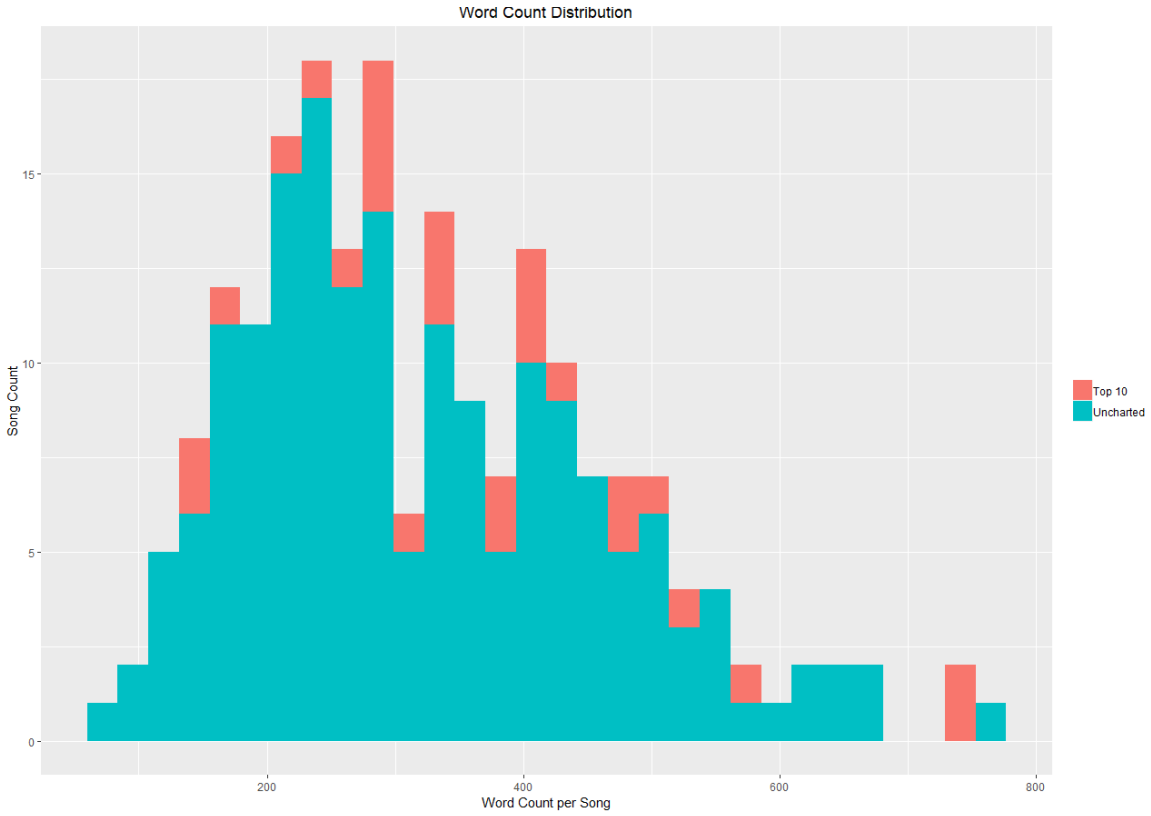
word	Song	Year	Peak	decade	chart_level
world	You Rock My World	2001	10	2000s	Top 10
world	Will You Be There	1991	7	1990s	Top 10
world	You Are Not Alone	1995	1	1990s	Top 10
world	Why You Wanna Trip on Me	1991	NA	1990s	Uncharted
world	Wings of My Love	1972	NA	1970s	Uncharted
world	With a Child's Heart	1973	NA	1970s	Uncharted
world	Xscape	2014	NA	2010s	Uncharted
world	You Are My Life	2001	NA	2000s	Uncharted
world	You Are There	1975	NA	1970s	Uncharted
world	You Can't Win	1978	NA	1970s	Uncharted

Diving deeper into analysing the corpus, I would like to know the top 10 songs with highest word frequency. This is done again by tokenizing but without filtering as we require the true count of the word frequency.

Songs With Highest Word Count

Song	chart_level	num_words
Blue Gangsta	Uncharted	762
Wanna Be Startin' Somethin'	Top 10	745
Man in the Mirror	Top 10	733
Who Is It	Uncharted	674
Chicago	Uncharted	672
Keep the Faith	Uncharted	652
Jam	Uncharted	644
Can't Let Her Get Away	Uncharted	632
What More Can I Give	Uncharted	626
Superfly Sister	Uncharted	606

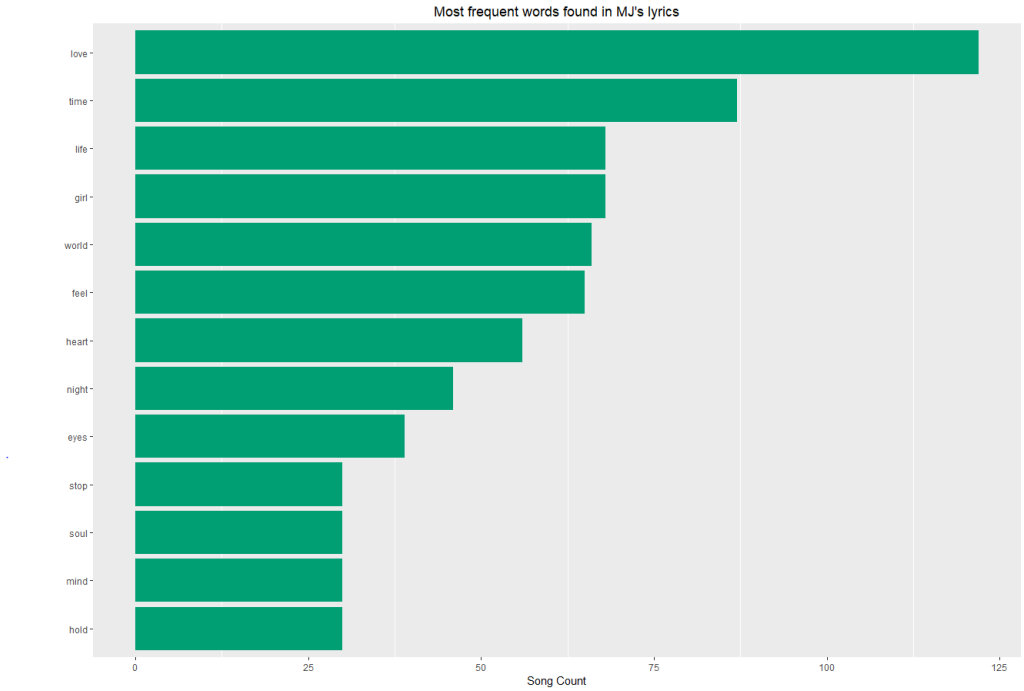
Delving even further into the data I wanted to see the word count distribution across all of Michael Jackson's songs. Was there a sweet spot? Are there any anomalies? In order to find out I managed to plot a histogram with word count per song on the x-axis and number of songs on the y-axis.



It is apparent that songs with around 200 to 400 words provide the best hits. This could be due to shorter track times making it easier for listeners to listen to and change to a different song or some other factor that we might not know about. It is also apparent from the graph that some songs with extreme large word counts can also hit the top 10 in the billboard charts. Looking into songs such as Wanna be startin' something and man in the mirror, it becomes quite clear that the songs are repetitive. A great can be found here discussing the topic of repetitiveness in songs.

Top Words

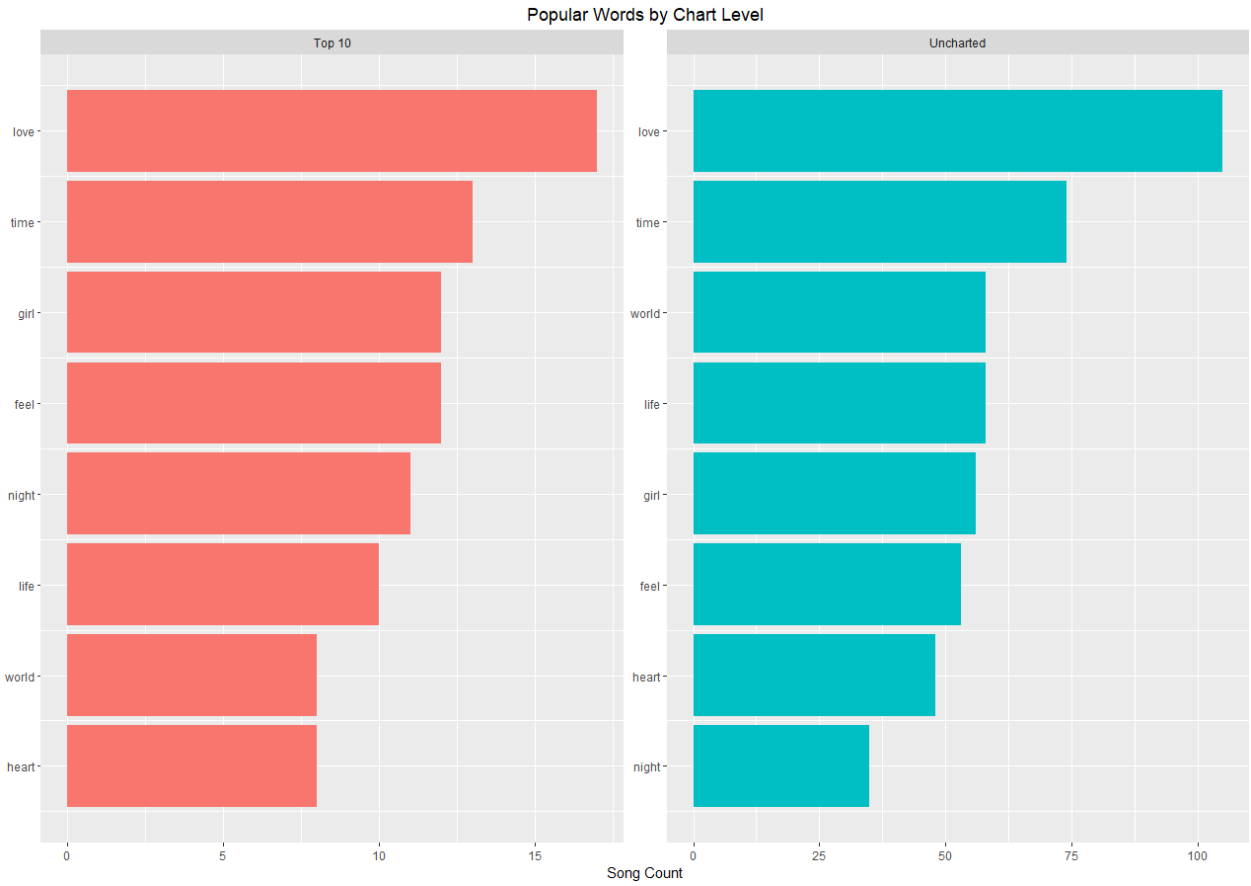
What are the King of Pop's most frequently used words found in his lyrics? A bit of playing around with the data I was able to find out his top 10 most frequent words.



Sometimes wordclouds, even though a lot of programmers bash them, can provide insight on the dataset. In this case it would be useful as we can see which words are most often used. Here we can see that love, time, heart, life and couple of other words are overused and trump the rest.

Popular Words

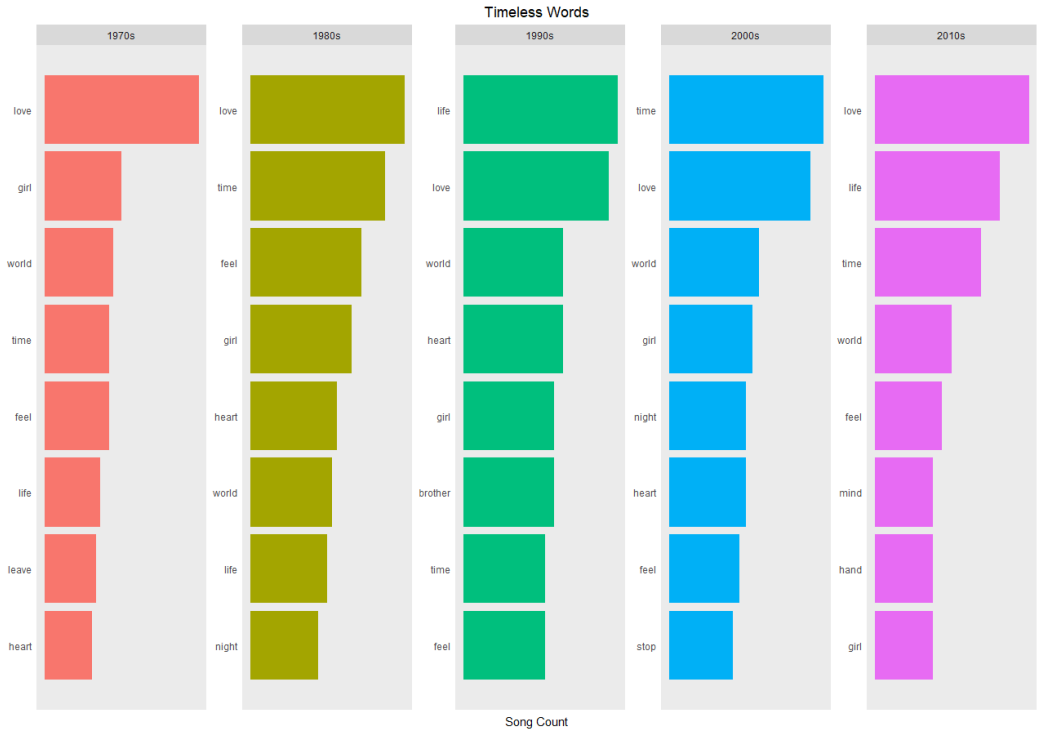
Looking into popular words by chart level shows us what the public gravitates to. What do the masses most likely want to hear? About love & heartbreak? Or about saving the world and coming together? Let's have a look.



Looks like people are more into their emotions than rescuing the planet. Personally, I think earth song peaking at 32 is an utter disgrace, rarely do we hear songs about things that do not involve ourselves or our love interests. I could go on and on about this, but I will leave that discussion for another time.

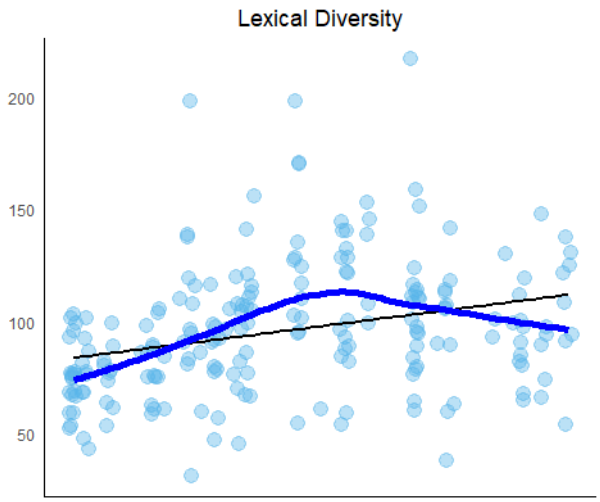
Timeless Words

Words that transcend time, where most never get tired of hearing in songs. Michael was active as a solo artist from the 70's all the way up to 2010's. Let's have a look and see which words have been most popular.

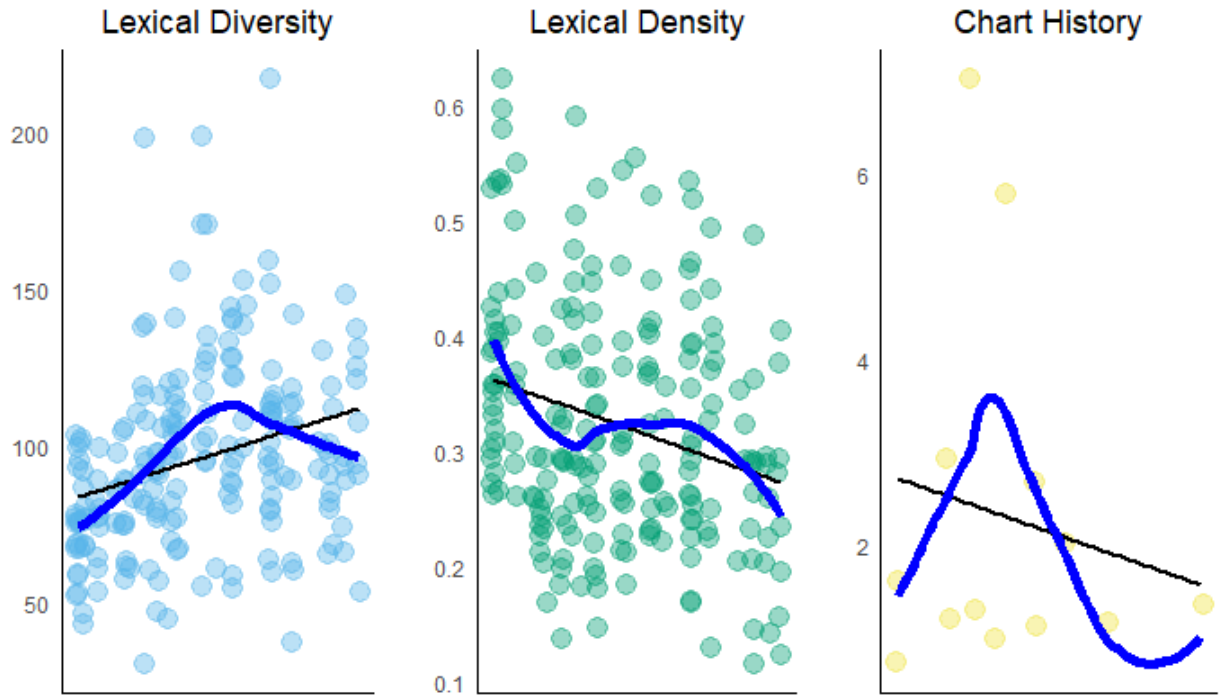


Lexical Diversity

How good is Michael's vocabulary? Do his songs end up being too repetitive or are they lyrical masterpieces?



This indicates that as the decades rolled by, the range of vocabulary found in Michael's songs increased. To find out how his lyrically complicated songs fared against his old tracks I plotted a lexical density plot i.e. number of unique words found in a song, and chart history that corresponds with both plots.



The following plot shows us that even though his vocabulary range increased, the number of unique words decreased (adding lots of repetition). The chart history plot shows that its heavily skewed to the right with a peak – this could signify when he was at his prime spewing out hits in the 80s.

Sentiment Lexicons

Exploring the meaning and intent behind the lyrics can be quite tricky. However, I have decided to apply certain lexicons from the tidytext package to conduct sentiment analysis on Michael Jackson's Lyrics. (AFINN, nrc and Bing).

Using nrc (captures the greatest number of lyrics) I wanted to see the moods provided by all of MJ's lyrics. The following was created using nrc sentiment lexicon.

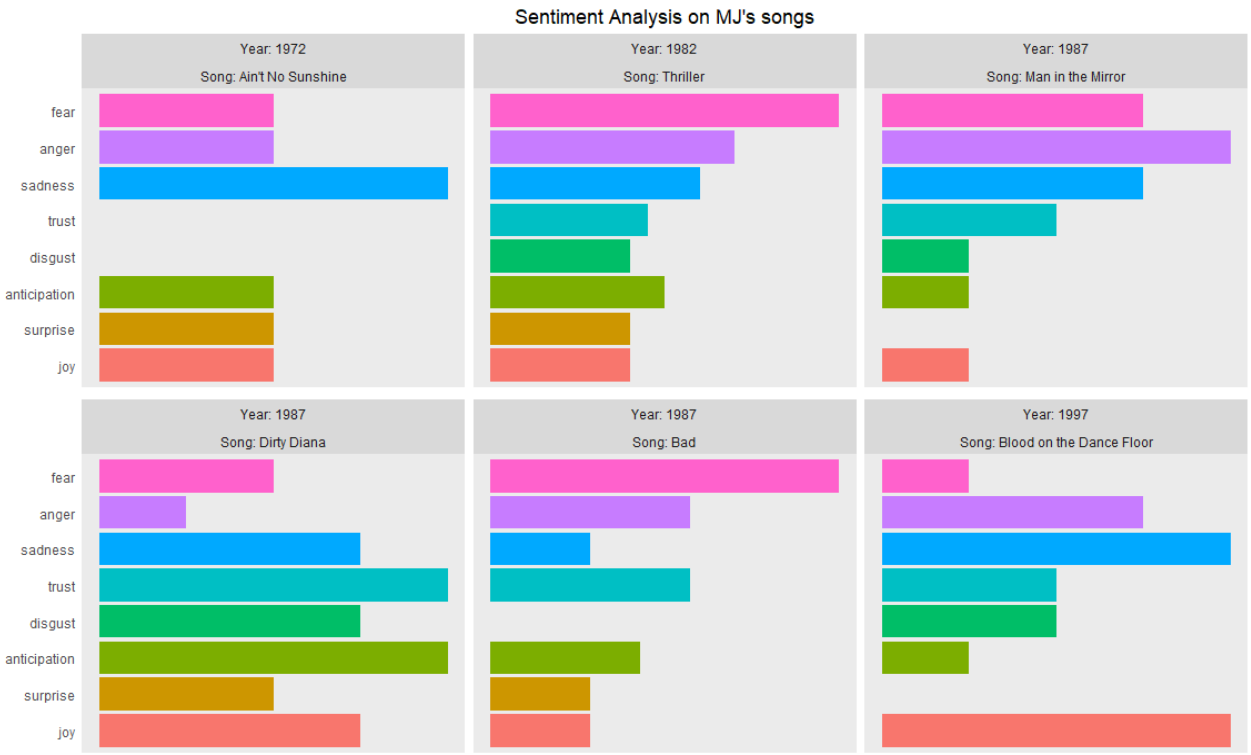
From what we can see, the lexicon captures and classifies over 1000 words with positive sentiment. I guess the King of Pop was a happy man after all. Let's do the same but using the bing lexicon which is more binary, classifying sentiment as either positive or negative.

Strange! Bing classified more negative lyrics than positive. There could be many reasons to this, it could be that bing does not capture or classify as much lyrics as nrc does. To make sure I have applied all three lexicons and tabled them together.

Lyrics Found In Lexicons

lexicon	lex_match_words	words_in_lyrics	match_ratio
AFINN	416	2966	0.1402562
bing	563	2966	0.1898179
nrc	768	2966	0.2589346

Taking a step deeper, I wanted to apply sentiment analysis on each song, what emotions do such songs evoke? Is the sentiment analysis classifying correctly? Let's have a look

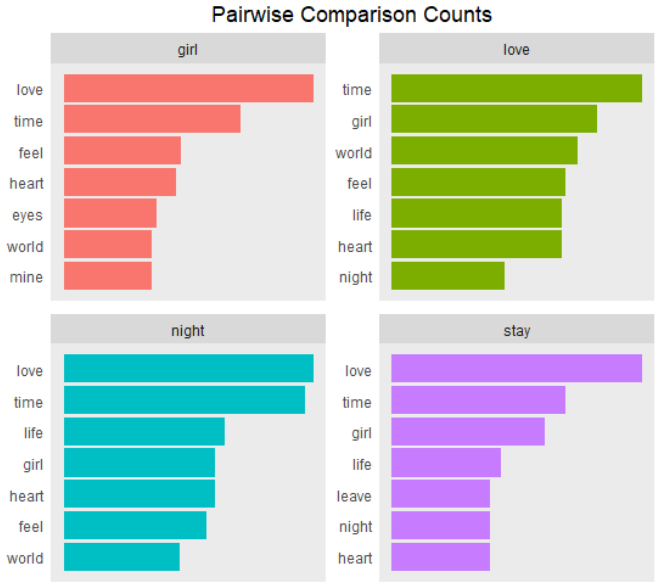


Ooh! Interesting. Let us start with the first song, Michael refers to a woman as his light, when she's gone then he is thrown into the darkness. A sad song indeed, and the analysis correctly classifies the song's sentiment. Thriller on the other hand, is about a beast (Michael in this case) that is going to strike at night and that she or he must be scared. Scared from the horror of the thriller. Again, correctly classified so are the rest. Finally, I will be moving on towards pairwise comparisons.

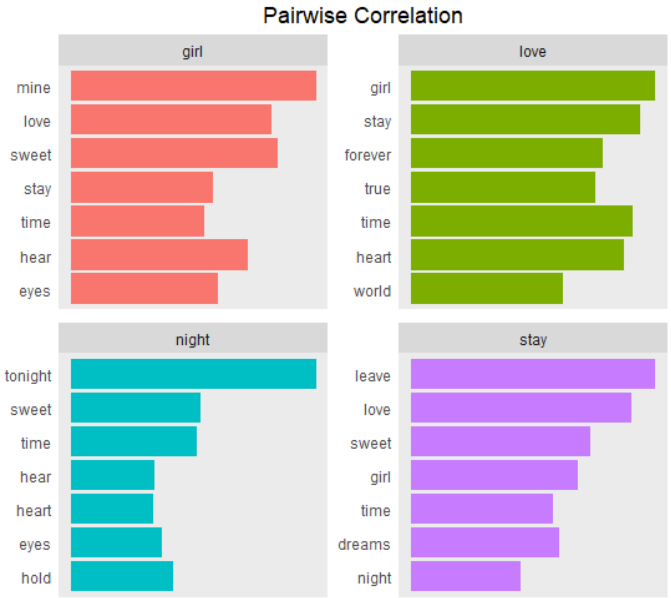
Pairwise comparisons

Ever wondered what a word could relate to? Or what word occurs mostly with what? That's where pairwise_cor() and pairwise_count() come to play.

Pairwise count tells us which words occur most often with given word. In this case I took a sample of four words: love, night, stay and girl and wanted to see which words most often occur with these words.



Pairwise cor tells us which words correlate mostly with given words.



Conclusion

Creating my own dataset in R was a first for me. It was tough, time consuming and nerve wrecking but I am happy that I was able to retrieve and clean the data to provide the results shown in the report and code. Using lexicons on the dataset provided a lot of insight on Michael's songs, what he probably felt and wanted us to feel. I hope this report provided fun and interesting ideas and knowledge. This is still a work in progress and I hope I will get back to it and improve it even further, perhaps by applying machine learning techniques etc.

Hope you enjoyed reading this report, I certainly enjoyed working on it.