Text Mining Kanye West

Zach Silvey

November 12, 2017

Motivation

I became interested in text mining after attending a talk by Julia Silge at the Portland R User meetup group. She was demoing some of the concepts from her new book Text Mining With R: A Tidy Approach, coauthored by David Robinson. The book covers basics of text mining with practical exapmles in the context of their recently released R package `tidytext`. Text mining is interesting to me from both a personal and professional perspective, as a market research I often am dealing with open-ended survey responses and text mining seemed like a great way to get more value out of those data. After finishing the book I figured what better way to demos some of my new skills than to do a text analysis project. This is that projects, my aim is to include an analysis from each of the chapters in the book.

All code and data for this project are available on GitHub.

The data

Survey data is my bread and butter but even I can admit that it isn't generally the most exciting. So after some thought I figured I'd go with a data source that is more pop-culturally relavent by using song lyrics. And who would make a better subject than Kanye West. More specifically, I collected the lyrics from all of the songs on Kanye's core discography. This includes his 7 studio releases, no deluxe editions or b-sides. I initally wanted to use this as a chance to work on my web scraping skills but after messing with it for a couple days I went ahead and collected the data manually. My source was genius, from there I pulled: album, song title, year released, length, featured artist(s), writer(s), and whether the track was a skit or not. I organized the data in Excel and then exported it as a CSV for easy ingestion into R.

Packages

The required packages used in this analysis are

```
library(tidyverse)
library(tidytext)
```

Importing and cleaning the data

The dataset was manually and therefore was kept fairly simple and relatively clean so necessary cleaning is minimal. The first thing I did was create a new `runtime` variable to store song lenght in seconds, then I removed text between [and] in the `lyrics` column. Genius uses text surrounded by square brakets to indicate parts of songs (e.g., verse, chorus, bridge, etc.), while these may be useful for more advanced analyses I chose to remove them for simplicity here.

```
lyrics <- read_csv("Data/Kanye Lyrics (beta1).csv",
                   col_types = cols(`Run-time` = col_character())) %>%
  separate("Run-time", c("minutes", "seconds"), sep = ":", convert = TRUE) %>%
  mutate("run_time" = (minutes * 60) + seconds) %>%
  mutate(Lyrics = str_replace_all(Lyrics, "\\[[^]]*]", "")) %>%
  select(-minutes, -seconds)
```

```
glimpse(lyrics)
```

```
## Observations: 121
## Variables: 8
## $ Album                <chr> "The College Dropout", "The College Dropo...
## $ Song                 <chr> "Intro (Skit)", "We Don't Care", "Graduat...
## $ Year                 <int> 2004, 2004, 2004, 2004, 2004, 2004, 2004,...
## $ `Featured artist(s)` <chr> NA, NA, NA, "Syleena Johnson", NA, "GLC", ...
## $ `Writer(s)`          <chr> "Kanye West", "Kanye West, Miri Ben-Ari", ...
## $ Skit                 <chr> "Y", "N", "N", "N", "N", "N", "N", "N", "...
## $ Lyrics               <chr> "Kanye, can I talk to you for a minute? M...
## $ run_time             <dbl> 19, 239, 82, 223, 69, 324, 193, 324, 289,...
```

Word frequencies

A common place to start a text mining analysis is to look at term frequencies. This gives us an idea of the type of words that are being used in our documents of interest. We can examine term frequencies with a few simple commands. The first step is to tokenize the text, in this case into single words. To tokenize we use the `unnest_tokens()` function.

```
lyrics %>%
  unnest_tokens(word, Lyrics) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 5,799 x 2
##     word     n
##    <chr> <int>
##  1     i  2375
##  2   the  2192
##  3   you  1702
##  4    to  1229
##  5   and  1132
##  6     a  1103
##  7    my   843
##  8    it   802
##  9    me   760
## 10    in   746
## # ... with 5,789 more rows
```

Upon examination we see that the most common words found across Kanye West lyrics are not that interesting. This is true in almost any corpora of natural language. These words fall into a category of text called stop words, and `tidytext` provides a quick way to remove them. But before we remove the stop words we should take not that the most common word in Kanye's lyrics is "I". To those familiar with Kanye's antics this will come as no surprise.

```
lyrics %>%
  unnest_tokens(word, Lyrics) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
## # A tibble: 5,296 x 2
##      word     n
##     <chr> <int>
```

```
##  1   love   191
##  2     la   187
##  3   shit   187
##  4   yeah   169
##  5   life   139
##  6 niggas   138
##  7   feel   133
##  8  nigga   131
##  9     uh   130
## 10   baby   125
## # ... with 5,286 more rows
```

With the addition of a single `anti_join` we removed the stop words and are left with words containing more information. Interestingly love comes to the top, a topic Kanye often addresses in his work. Let look at the ocurrance of "love" across his albums.

```
album_totals <- lyrics %>%
  unnest_tokens(word, Lyrics) %>%
  anti_join(stop_words) %>%
  group_by(Album) %>%
  count(word) %>%
  summarise(total = sum(n))
```

```
## Joining, by = "word"
```

```
lyrics %>%
  unnest_tokens(word, Lyrics) %>%
  anti_join(stop_words) %>%
  filter(word == "love") %>%
  count(word, Album) %>%
  inner_join(album_totals) %>%
  mutate(freq = n/total) %>%
  arrange(-freq)
```

```
## Joining, by = "word"
```

```
## Joining, by = "Album"
```

```
## # A tibble: 8 x 5
##    word                              Album     n total        freq
##   <chr>                              <chr> <int> <int>       <dbl>
## 1  love                 808s & Heartbreak    42  1130 0.037168142
## 2  love                  Watch the Throne    34  2366 0.014370245
## 3  love                            Yeezus    21  1628 0.012899263
## 4  love My Beautiful Dark Twisted Fantasy    30  2854 0.010511563
## 5  love                The Life of Pablo    33  3342 0.009874327
## 6  love                       Graduation     7  2269 0.003085059
## 7  love               The College Dropout    14  4672 0.002996575
## 8  love                 Late Registration    10  3470 0.002881844
```

When we rank Kanye's albums by frequency of occurance of the word love, "808s & Heartbreak" rises to the top. If you've read Kanye's Wikipedia page you'll know that he made 808s soon after his mother passed away and his then fiance ended their engament. Another trend we see when looking at "love" is that it appears there is a relationship between occurance of the term and tenure in his career, with love appearing more commonly on his newer works.

```
year_totals <- lyrics %>%
  unnest_tokens(word, Lyrics) %>%
```

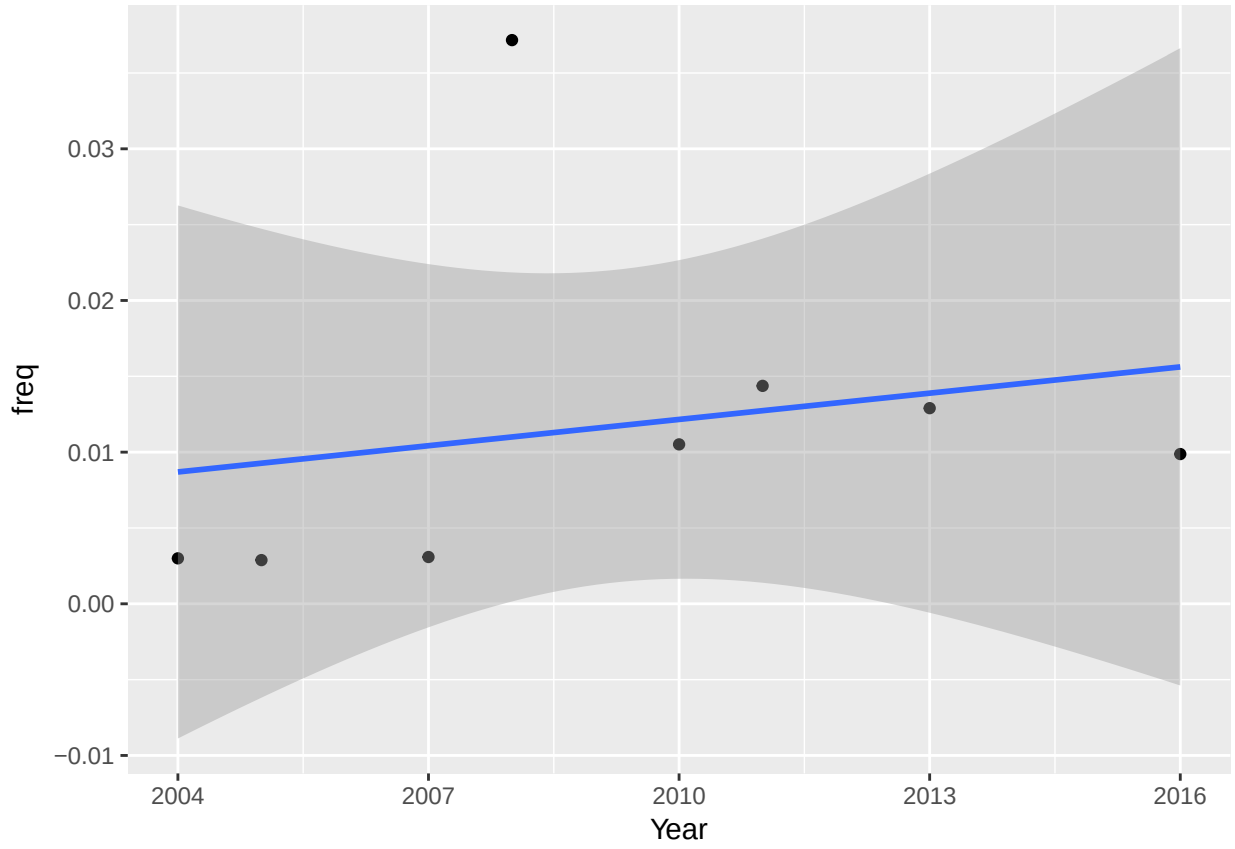
```
  anti_join(stop_words) %>%
  group_by(Year) %>%
  count(word) %>%
  summarise(total = sum(n))
```

```
## Joining, by = "word"
```

```
lyrics %>%
  unnest_tokens(word, Lyrics) %>%
  anti_join(stop_words) %>%
  filter(word == "love") %>%
  count(word, Year) %>%
  inner_join(year_totals) %>%
  mutate(freq = n/total) %>%
  ggplot(aes(Year, freq)) +
  geom_point(show.legend = FALSE) +
  geom_smooth(method = "lm", se = TRUE)
```

```
## Joining, by = "word"
```

```
## Joining, by = "Year"
```



A relationship made even more obvious when we remove the outlier of 808s.

```
lyrics %>%
  unnest_tokens(word, Lyrics) %>%
  anti_join(stop_words) %>%
  filter(word == "love" & Year != 2008) %>%
```

```
  count(word, Year) %>%
  inner_join(year_totals) %>%
  mutate(freq = n/total) %>%
  ggplot(aes(Year, freq)) +
  geom_point(show.legend = FALSE) +
  geom_smooth(method = "lm", se = TRUE)
```

```
## Joining, by = "word"
```

```
## Joining, by = "Year"
```