Saturday, June 22, 2013

Everything in Its Right Place: Visualization and Content Analysis of Radiohead Lyrics

Introduction
I am not a huge Radiohead fan.

To be honest, the Radiohead I know and love and remember is that which was a rock band without a lot of 'experimental' tracks - a band you discovered on Big Shiny Tunes 2, or because your friends told you about it, or because it was playing in the background of a bar you were at sometime in the 90's.

But I really do like their music, I've become familiar with more of it and overall it does possess a certain unique character in its entirety. Their range is so diverse and has changed so much over the years that it would be really hard not to find at least one track that someone will like. In this way they are very much like the Beatles, I suppose.

I was interested in doing some more content analysis type work and text mining in R, so I thought I'd try song lyrics and Radiohead immediately came to mind.

Background
In order to first do the analysis, we need all the data (shocking, I know). Somewhat surprisingly, putting '*radiohead data*' into Google comes up with little except for many, many links to the video and project for **House of Cards** which was made using LIDAR technology and had the data set publicly **released**.

So once again we are in this situation where we are responsible for not only analyzing all the data and communicating the findings, but also getting it as well. Such is the life of an analyst, everyday and otherwise (see my **previous musing** on this point).

The lyrics data was taken from the listing of Radiohead lyrics at **Green Plastic Radiohead**.

Normally it would be simply a matter of throwing something together in Python using **Beautiful Soup** as I have **done previously**. Unfortunately, due to the way these particular pages were coded, that proved to be a bit more difficult than expected.

As a result the extraction process ended up being a convoluted ad-hoc **data wrangling** exercise involving the use of **wget**, **sed** and **Beautiful Soup** - a process which was neither enjoyable nor something I would care to repeat.

In retrospect, two points:

Getting the data is not always easy.
Sometimes sitting down beforehand and looking at where you are getting it from, the format it is in and how to best go about getting it into the format you need will save you a lot of wasted time and frustration in the long run. Ask questions before you begin - what format is the data in now? What is the format I need/would like it to be in to do the analysis? What steps are required in order to get from one to the other (*i.e.* what is the data transformation or mapping process)?

That being said, my methods got me where I needed to be, however there were most likely easier, more straightforward approaches which would have saved a lot frustration on my part.

If you're going to code a website, use a sane page structure and give important page elements ids.
Make it easy on your other developers (and the rest of the world in general) by labeling your * * containers and other elements with ids (which are unique!!) or at least classes. Otherwise how are people going to scrape all your data and steal it for their own ends? I joke... kind of.

In this case my frustrations actually stemmed mainly from some questionable code for a **cache-buster**. But even once I got past that, the contents of the main page containers were somewhat inconsistent. Such is life, and the internet.

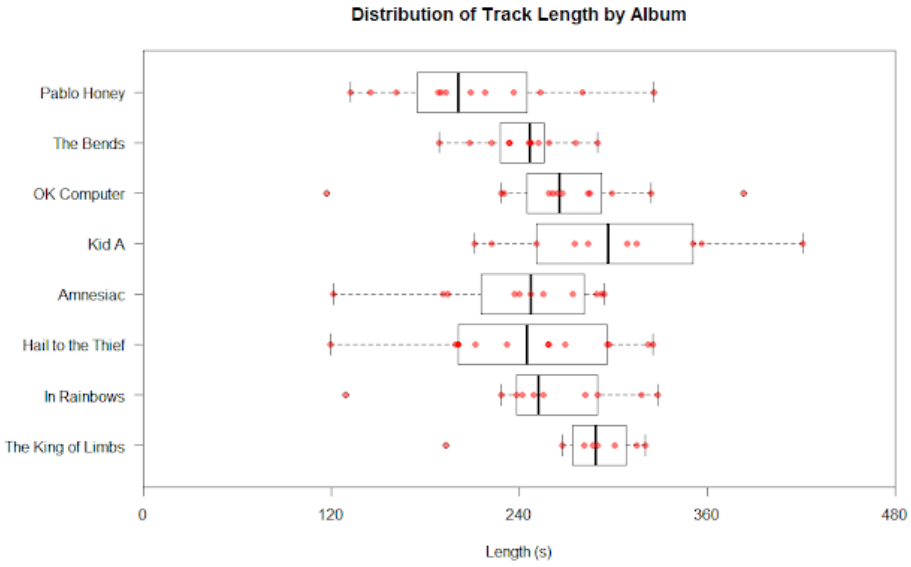
The remaining data, album and track length - were taken from the **Wikipedia pages** for each album and later merged with the calculations done with the text data in R.

Okay, enough whinging - we have the data - let's check it out.

Analysis
I stuck with what I consider to be the 'canonical' Radiohead albums - that is, the big releases you've probably heard about even if you're like me a not a hardcore Radiohead fan - 8 albums in total (*Pablo Honey, The Bends, OK Computer, Kid A, Amnesiac, Hail to the Thief, In Rainbows,* and *The King of Limbs*).

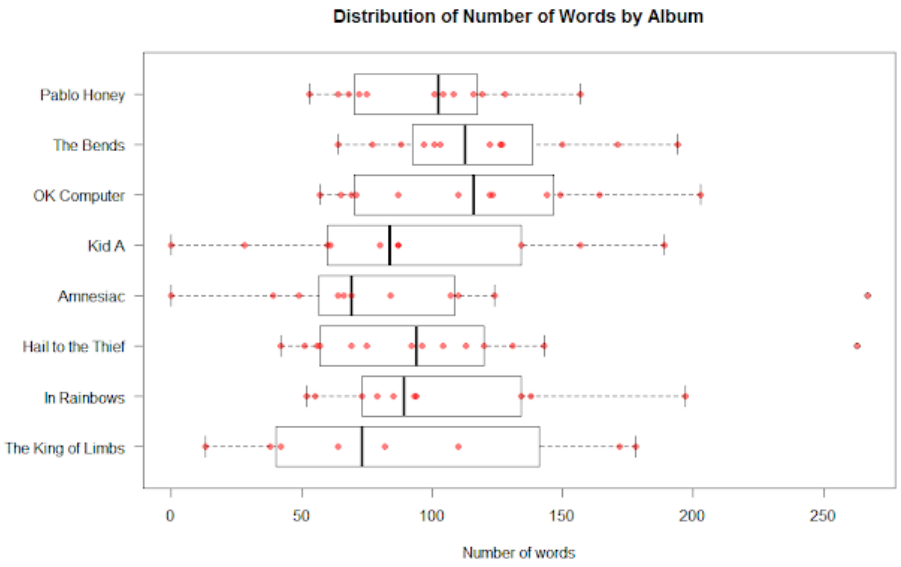
Unstructured (and non-quantitative) data always lends itself to more interesting analysis - with something like text, how do we analyze it? How do we quantify it? Let's start with the easily quantifiable parts and go from there.

Track Length
Below is a boxplot of the track lengths per album, with the points overplotted.



Interestingly *Pablo Honey* and *Kid A* have the largest ranges of track length (from 2:12 to 4:40 and 3:42 to 7:01 respectively) - if you ignore the single tracks around 2 minutes on *Amnesiac* and *Hail to the Thief* the variance of their track lengths is more in line with all the other albums. Ignoring the single outlier, *The King of Limbs* is appears to be special given its narrow range of track lengths.

Word Count
Next we look at the number of words (lyrics) per album:



There is a large range of word counts, from the two truly instrumental tracks (*Treefingers* on *Kid A* and *Hunting Bears* on *Amnesiac*) to the wordier tracks (*Dollars and Cents* and *A Wolf at the Door*). *Pablo Honey* almost looks like it has two categories of songs - with a split around the 80 word mark.

Okay, interesting and all, but again these are small amounts of data and only so much can be drawn out as such.

Going forward we examine two calculated quantities.

Calculated Quantities - Lexical Density and 'Lyrical Density'

In the realm of **content analysis** there is a measure known as **lexical density** which is a measure of the number of content words as a proportion of the total number of words - a value which ranges from 0 to 100. In general, the greater the lexical density of a text, the more content heavy it is and more 'unpacking' it takes to understand - texts with low lexical density are easier to understand.

According to Wikipedia the **formula** is as follows:

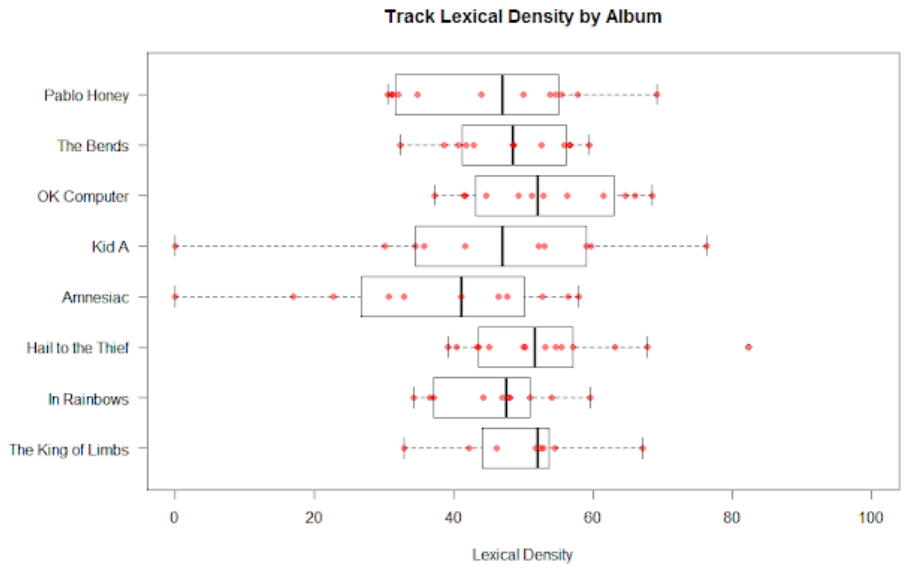
$$Ld = (Nlex/N) * 100$$

where *Ld* is the analysed text's lexical density, *NLex* is the number of lexical word tokens (nouns, adjectives, verbs, adverbs) in the analysed text, and *N* is the number of all tokens (total number of words) in the analysed text.

Now, I am not a linguist, however it sounds like this is just the ratio of words which are not **stopwords** to the total number - or could at least be approximated by it. That's what I went with in the calculations in R using the **tm package** (because I'm not going to write a package to calculate lexical density by myself).

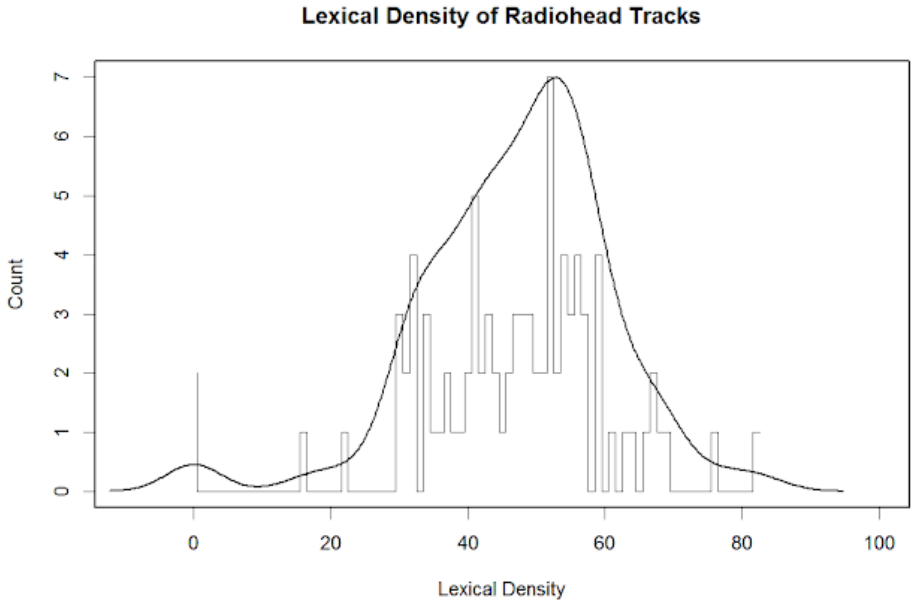
On a related note, I completely made up a quantity which I am calling *'lyrical density'* which is much easier to calculate and understand - this is just the number of lyrics per song over the track length, and is measured in words per second. An instrumental track would have lyrical density of zero, and a song with one word per second for the whole track would have a lyrical density of 1.

Lexical Density



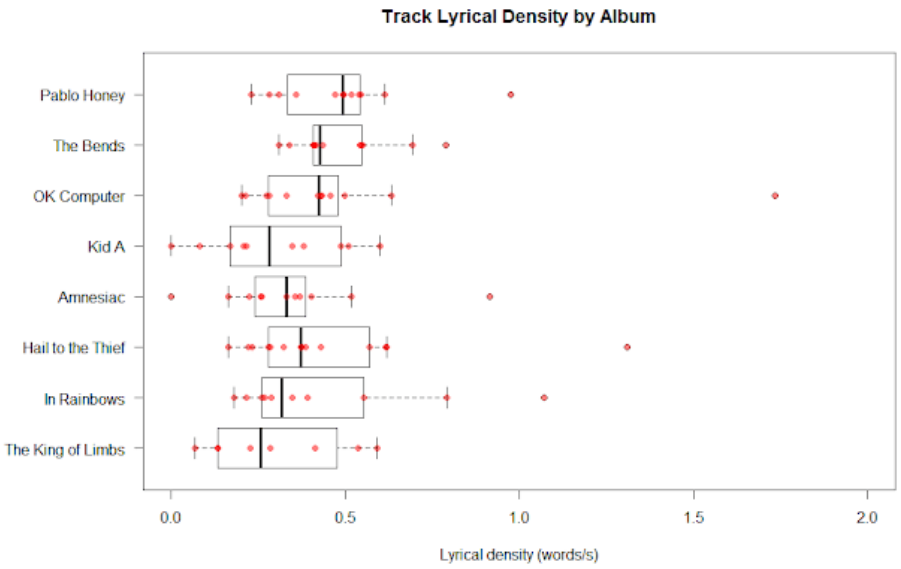
Looking at the calculated lexical density per album, we can see that the majority of songs have their lexical density between about 30 to 70. The two instrumental songs have a lexical density of 0 (as they have no words) and distribution appears most even on *OK Computer*. The most content-word heavy song is on *Hail to the Thief* and is *I Will (No Man's Land)*.

If you could imaging extending the number of songs Radiohead written to infinity, you might get a density function something like below, with the bulk of songs having density between 30 and 70 (which I imagine is a normal reasonable range for any text) and a little bump at 0 for their instrumental songs:



Lyrical Density

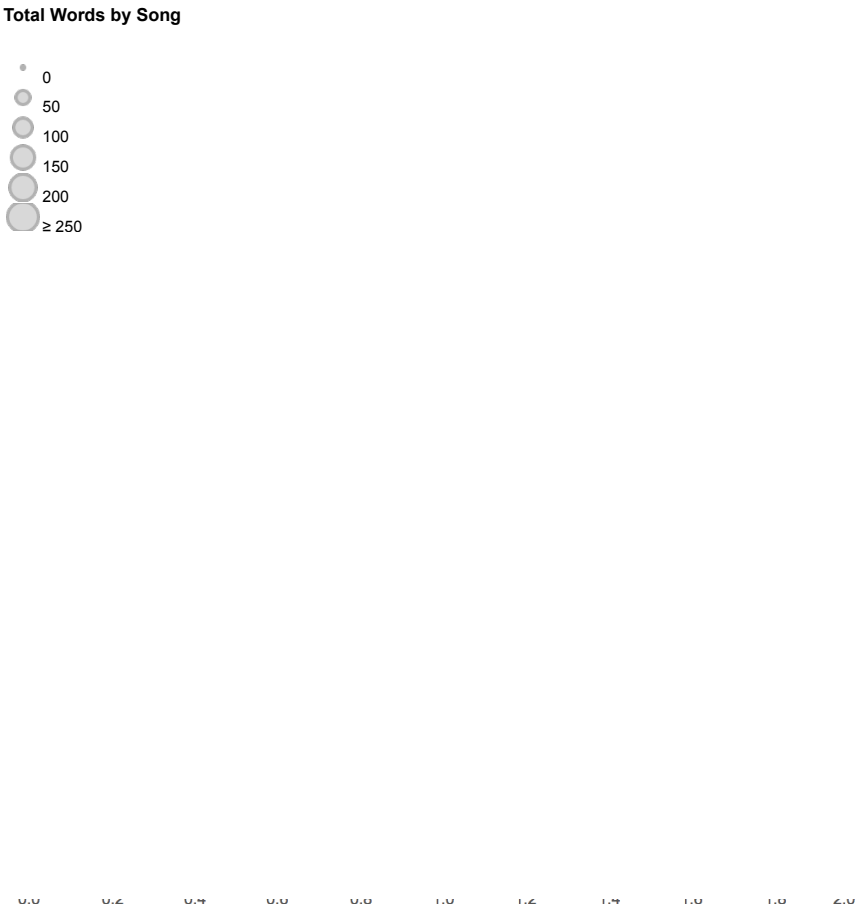
Next we come to my calculated quantity, lyrical density - or the number of words per second on each track.



Interestingly, there are outlying tracks near the high end where the proportion of words to the song length is greater than 1 (*Fitter Happier, A Wolf at the Door,* and *Faust Arp*). *Fitter Happier* shouldn't even really count, as it is really an instrumental track with a synthesized voice dubbed overtop. If you listen to *A Wolf at the Door* it is clear why the lyrical density is so high - Thom is practically rapping at points. Otherwise *Kid A* and *The King of Limbs* seem to have less quickly sung lyrics than the other albums on average.

Lexical Density + Lyrical Density

Putting it all together, we can examine the quantities for all of the Radiohead songs in one data visualization. You can examine different albums by clicking the color legend at the right, and compare multiple albums by holding CTRL and clicking more than one.



Learn About Tableau

The songs are colour-coded by album. The points are plotted by lexical density along y-axis against the lyrical density along the x-axis and sized by total number of words in the song. As such, the position of the point in the plot gives an idea of the rate of lyrical content in the track - a song like *I Might Be Wrong* is fitting a lot less content words into a song at a slower rate than a track like *A Wolf at the Door* which is packed much tighter with both lyrics and meaning.

Conclusion
This was an interesting project and it was fascinating to take something everyday like song lyrics and analyze them as data (though some Radiohead fans might argue that there is nothing 'everyday' about Radiohead lyrics).

All in all, I feel that a lot of the analysis has to be taken with a grain of salt (or a shaker or two), given the size of the data set (n = 89).

That being said, I still feel it is still proof positive that you can take something typically thought of as very artistic and qualitative like a song, and classify it in a meaningful way in quantitative fashion. I had never listened to the song *Fitter Happier*, yet it is a clear outlier in several measures - and listening to the song I discovered why - it is a track with a robot-like voice over and not containing sung lyrics at all.

A more interesting and ambitious project would be to take a much larger data set, where the measures examined here would be more reliable given the large *n,* and look at things such as trends in time (the evolution of American rock lyrics) or by genre / style of music. This sort of thing exists out there already to an extent, for example, in work done with **The Million Song Data Set** which I came across in some of my Google searches I made for this project.

But as I said, this would be a large and ambitious amount of work, which is perhaps more suited for something like a **research paper** or thesis - I am just one (everyday) analyst.

References & Resources
Radiohead Lyrics at Green Plastic Radiohead
http://www.greenplastic.com/radiohead-lyrics/

The Million Song Data Set
http://labrosa.ee.columbia.edu/millionsong/

Measuring the Evolution of Contemporary Western Popular Music [PDF]
http://www.nature.com/srep/2012/120726/srep00521/pdf/srep00521.pdf

Radiohead "House of Cards" by Aaron Koblin
http://www.aaronkoblin.com/work/rh/

code, data & plots on github
http://github.com/mylesmharrison/radiohead_lyrics

at **2:34 PM**

Labels: **dataviz**, **python**, **R**, **radiohead**, **tableau**, **text analysis**

5 comments:

Anonymous **September 25, 2015 at 12:05 PM**

Waaaah this is so coooollll!!!!! I'm a freshman majoring in statistical and this is so cool!!!! Goodluck Mr. Myles!!!!

Reply

Replies

Myles Harrison **September 28, 2015 at 2:58 PM**

Thanks! Glad you liked it!

Reply

jo October 14, 2016 at 1:23 PM

Thanks for the article, good piece of reading.
I am desperately looking for free softwares allowing content analysis.
Any recommendations?

Reply

Replies

Myles Harrison October 17, 2016 at 7:25 PM

Thank you, glad you enjoyed it!

You might be interested in Rapidminer or KNIME, both of which have text mining functionality.

jo October 20, 2016 at 6:20 AM

Hi Myles,
I will have a look at it.
Thank you

Reply

Enter your comment...

Comment as: alexing10 (Go

Sign out

Publish **Preview** Notify me

Newer Post **Home** **Older Post**

Subscribe to: **Post Comments (Atom)**

Previous

- ► **2017** (4)
- ► **2016** (3)
- ► **2015** (8)
- ► **2014** (17)
- ▼ **2013** (16)
  - ► **December** (1)
  - ► **November** (2)
  - ► **October** (1)
  - ► **September** (1)
  - ► **August** (1)
  - ▼ **June** (1)

    Everything in Its Right Place: Visualization and C...

- ► **May** (2)
- ► **April** (2)
- ► **March** (1)
- ► **February** (2)
- ► **January** (2)

► **2012** (26)

About Me