Sentiment analysis is a method of natural language processing that involves classifying words in a document based on whether a word is positive or negative, or whether it is related to a set of basic human emotions; the exact results differ based on the sentiment analysis method selected. The [tidytext](https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html) R package has 4 different sentiment analysis methods:

* “AFINN” for [Finn Årup Nielsen](http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010) – which classifies words from -5 to +5 in terms of negative or positive valence
* “bing” for [Bing Liu and colleagues](https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html) – which classifies words as either positive or negative
* “loughran” for [Loughran-McDonald](https://sraf.nd.edu/textual-analysis/resources/) – mostly for financial and nonfiction works, which classifies as positive or negative, as well as topics of uncertainty, litigious, modal, and constraining
* “nrc” for the [NRC lexicon](http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm) – which classifies words into eight [basic emotions](http://www.deeplytrivial.com/2016/05/on-feelings-woah-oh-oh-feelings.html) (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) as well as positive or negative sentiment

Sentiment analysis works on unigrams – single words – but you can aggregate across multiple words to look at sentiment across a text.

To demonstrate sentiment analysis, I’ll use one of my favorite songs: “Hotel California” by the Eagles.

I know, I know.

[](https://i1.wp.com/3.bp.blogspot.com/-E8RqtLBZX1U/Wv46lGHnMBI/AAAAAAAALkQ/Ewq0bsYAUSw5ea-o4sFKKw0Nd9aGJQKmACLcBGAs/s1600/eagles.gif?ssl=1)

Using similar code as last week, let’s pull in the lyrics of the song.

library(geniusR)  
library(tidyverse)

hotel\_calif <- genius\_lyrics(artist = "Eagles", song = "Hotel California") %>%  
 mutate(line = row\_number())

First, we’ll chop up these 43 lines into individual words, using the tidytext package and unnest\_tokens function.

library(tidytext)  
tidy\_hc <- hotel\_calif %>%  
 unnest\_tokens(word,lyric)

This is also probably the point I would remove stop words with anti\_join. But these common words are very unlikely to have a sentiment attached to them, so I’ll leave them in, knowing they’ll be filtered out anyway by this analysis. We have 4 lexicons to choose from. Loughran is more financial and textual, but we’ll still see how well it can classify the words anyway. First, let’s create a data frame of our 4 sentiment lexicons.

new\_sentiments <- sentiments %>%  
 mutate( sentiment = ifelse(lexicon == "AFINN" & score >= 0, "positive",  
 ifelse(lexicon == "AFINN" & score < 0,  
 "negative", sentiment))) %>%  
 group\_by(lexicon) %>%  
 mutate(words\_in\_lexicon = n\_distinct(word)) %>%  
 ungroup()

Now, we’ll see how well the 4 lexicons match up with the words in the lyrics. Big thanks to [Debbie Liske at Data Camp](https://www.datacamp.com/community/tutorials/sentiment-analysis-R) for this piece of code (and several other pieces used in this post):

my\_kable\_styling <- function(dat, caption) {  
 kable(dat, "html", escape = FALSE, caption = caption) %>%  
 kable\_styling(bootstrap\_options = c("striped", "condensed", "bordered"),  
 full\_width = FALSE)  
}  
  
  
library(kableExtra)  
library(formattable)  
library(yarrr)

tidy\_hc %>%  
 mutate(words\_in\_lyrics = n\_distinct(word)) %>%  
 inner\_join(new\_sentiments) %>%  
 group\_by(lexicon, words\_in\_lyrics, words\_in\_lexicon) %>%  
 summarise(lex\_match\_words = n\_distinct(word)) %>%  
 ungroup() %>%  
 mutate(total\_match\_words = sum(lex\_match\_words),  
 match\_ratio = lex\_match\_words/words\_in\_lyrics) %>%  
 select(lexicon, lex\_match\_words, words\_in\_lyrics, match\_ratio) %>%  
 mutate(lex\_match\_words = color\_bar("lightblue")(lex\_match\_words),  
 lexicon = color\_tile("lightgreen","lightgreen")(lexicon)) %>%  
 my\_kable\_styling(caption = "Lyrics Found In Lexicons")

## Joining, by = "word"

| Lyrics Found In Lexicons | | | |
| --- | --- | --- | --- |
| **lexicon** | **lex\_match\_words** | **words\_in\_lyrics** | **match\_ratio** |
| AFINN | 18 | 175 | 0.1028571 |
| bing | 18 | 175 | 0.1028571 |
| loughran | 1 | 175 | 0.0057143 |
| nrc | 23 | 175 | 0.1314286 |

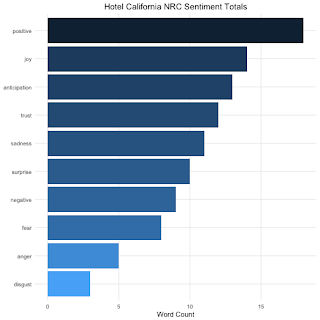
NRC offers the best match, classifying about 13% of the words in the lyrics. (It’s not unusual to have such a low percentage. Not all words have a sentiment.)

hcsentiment <- tidy\_hc %>%  
 inner\_join(get\_sentiments("nrc"), by = "word")  
  
hcsentiment

## # A tibble: 103 x 4  
## track\_title line word sentiment  
##   
## 1 Hotel California 1 dark sadness   
## 2 Hotel California 1 desert anger   
## 3 Hotel California 1 desert disgust   
## 4 Hotel California 1 desert fear   
## 5 Hotel California 1 desert negative   
## 6 Hotel California 1 desert sadness   
## 7 Hotel California 1 cool positive   
## 8 Hotel California 2 smell anger   
## 9 Hotel California 2 smell disgust   
## 10 Hotel California 2 smell negative   
## # ... with 93 more rows

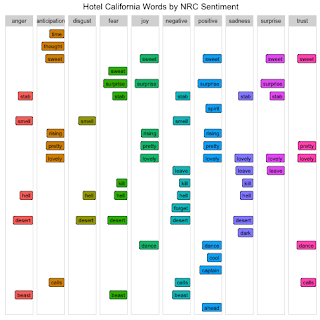
Let’s visualize the counts of different emotions and sentiments in the NRC lexicon.

theme\_lyrics <- function(aticks = element\_blank(),  
 pgminor = element\_blank(),  
 lt = element\_blank(),  
 lp = "none")  
{  
 theme(plot.title = element\_text(hjust = 0.5), #Center the title  
 axis.ticks = aticks, #Set axis ticks to on or off  
 panel.grid.minor = pgminor, #Turn the minor grid lines on or off  
 legend.title = lt, #Turn the legend title on or off  
 legend.position = lp) #Turn the legend on or off  
}  
  
hcsentiment %>%  
 group\_by(sentiment) %>%  
 summarise(word\_count = n()) %>%  
 ungroup() %>%  
 mutate(sentiment = reorder(sentiment, word\_count)) %>%  
 ggplot(aes(sentiment, word\_count, fill = -word\_count)) +  
 geom\_col() +  
 guides(fill = FALSE) +  
 theme\_minimal() + theme\_lyrics() +  
 labs(x = NULL, y = "Word Count") +  
 ggtitle("Hotel California NRC Sentiment Totals") +  
 coord\_flip()

[](https://i1.wp.com/1.bp.blogspot.com/-S79qwh3n2_U/Wv46zQIK0rI/AAAAAAAALkU/td8fTe2DtfcYeSeF4GGfyTQhXMhrp7ljACLcBGAs/s1600/unnamed-chunk-6-1.png?ssl=1)

Most of the words appear to be positively-valenced. How do the individual words match up?

library(ggrepel)  
  
plot\_words <- hcsentiment %>%  
 group\_by(sentiment) %>%  
 count(word, sort = TRUE) %>%  
 arrange(desc(n)) %>%  
 ungroup()  
  
plot\_words %>%  
 ggplot(aes(word, 1, label = word, fill = sentiment)) +  
 geom\_point(color = "white") +  
 geom\_label\_repel(force = 1, nudge\_y = 0.5,  
 direction = "y",  
 box.padding = 0.04,  
 segment.color = "white",  
 size = 3) +  
 facet\_grid(~sentiment) +  
 theme\_lyrics() +  
 theme(axis.text.y = element\_blank(), axis.line.x = element\_blank(),  
 axis.title.x = element\_blank(), axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 panel.grid = element\_blank(), panel.background = element\_blank(),  
 panel.border = element\_rect("lightgray", fill = NA),  
 strip.text.x = element\_text(size = 9)) +  
 xlab(NULL) + ylab(NULL) +  
 ggtitle("Hotel California Words by NRC Sentiment") +  
 coord\_flip()

[](https://i1.wp.com/1.bp.blogspot.com/-EtATflHXVSs/Wv464JMmxQI/AAAAAAAALkc/bCLO-HkdCr8Oau2phcTVN9F89riDTLubQCLcBGAs/s1600/unnamed-chunk-7-1.png?ssl=1)

It looks like some words are being misclassified. For instance, “smell” as in “warm smell of colitas” is being classified as anger, disgust, and negative. But that doesn’t explain the overall positive bent being applied to the song. If you listen to the song, you know it’s not really a happy song. It starts off somewhat negative – or at least, ambiguous – as the narrator is driving on a dark desert highway. He’s tired and having trouble seeing, and notices the Hotel California, a shimmering oasis on the horizon. He stops in and is greated by a “lovely face” in a “lovely place.” At the hotel, everyone seems happy: they dance and drink, they have fancy cars, they have pretty “friends.”

But the song is in a minor key. Though not always a sign that a song is sad, it is, at the very least, a hint of something ominous, lurking below the surface. Soon, things turn bad for the narrator. The lovely-faced woman tells him they are “just prisoners here of our own device.” He tries to run away, but the night man tells him, “You can check out anytime you like, but you can never leave.”

The song seems to be a metaphor for something, perhaps fame and excess, which was also the subject of another song on the same album, “Life in the Fast Lane.” To someone seeking fame, life is dreary, dark, and deserted. Fame is like an oasis – beautiful and shimmering, an escape. But it isn’t all it appears to be. You may be surrounded by beautiful people, but you can only call them “friends.” You trust no one. And once you join that lifestyle, you might be able to check out, perhaps through farewell tour(s), but you can never leave that life – people know who you are (or were) and there’s no disappearing. And it could be about something even darker that it’s hard to escape from, like substance abuse. Whatever meaning you ascribe to the song, the overall message seems to be that things are not as wonderful as they appear on the surface.

So if we follow our own understanding of the song’s trajectory, we’d say it starts off somewhat negatively, becomes positive in the middle, then dips back into the negative at the end, when the narrator tries to escape and finds he cannot.

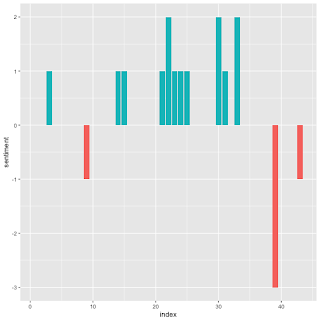
We can chart this, using the line number, which coincides with the location of the word in the song. We’ll stick with NRC since it offered the best match, but for simplicity, we’ll only pay attention to the positive and negative sentiment codes.

hcsentiment\_index <- tidy\_hc %>%  
 inner\_join(get\_sentiments("nrc")%>%  
 filter(sentiment %in% c("positive",  
 "negative"))) %>%  
 count(index = line, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)

## Joining, by = "word"

This gives us a data frame that aggregates sentiment by line. If a line contains more positive than negative words, its overall sentiment is positive, and vice versa. Because not every word in the lyrics has a sentiment, not every line has an associated aggregate sentiment. But it gives us a sort of trajectory over the course of the song. We can visualize this trajectory like this:

hcsentiment\_index %>%  
 ggplot(aes(index, sentiment, fill = sentiment > 0)) +  
 geom\_col(show.legend = FALSE)

[](https://i2.wp.com/4.bp.blogspot.com/-GDjSB5zx3eY/Wv468cfK7kI/AAAAAAAALkg/foyTNo0VmCEk8C5EM69vFDL6YQ5dwlVWQCLcBGAs/s1600/unnamed-chunk-9-1.png?ssl=1)

As the chart shows, the song starts somewhat positive, with a dip soon after into the negative. The middle of the song is positive, as the narrator describes the decadence of the Hotel California. But it turns dark at the end, and stays that way as the guitar solo soars in.