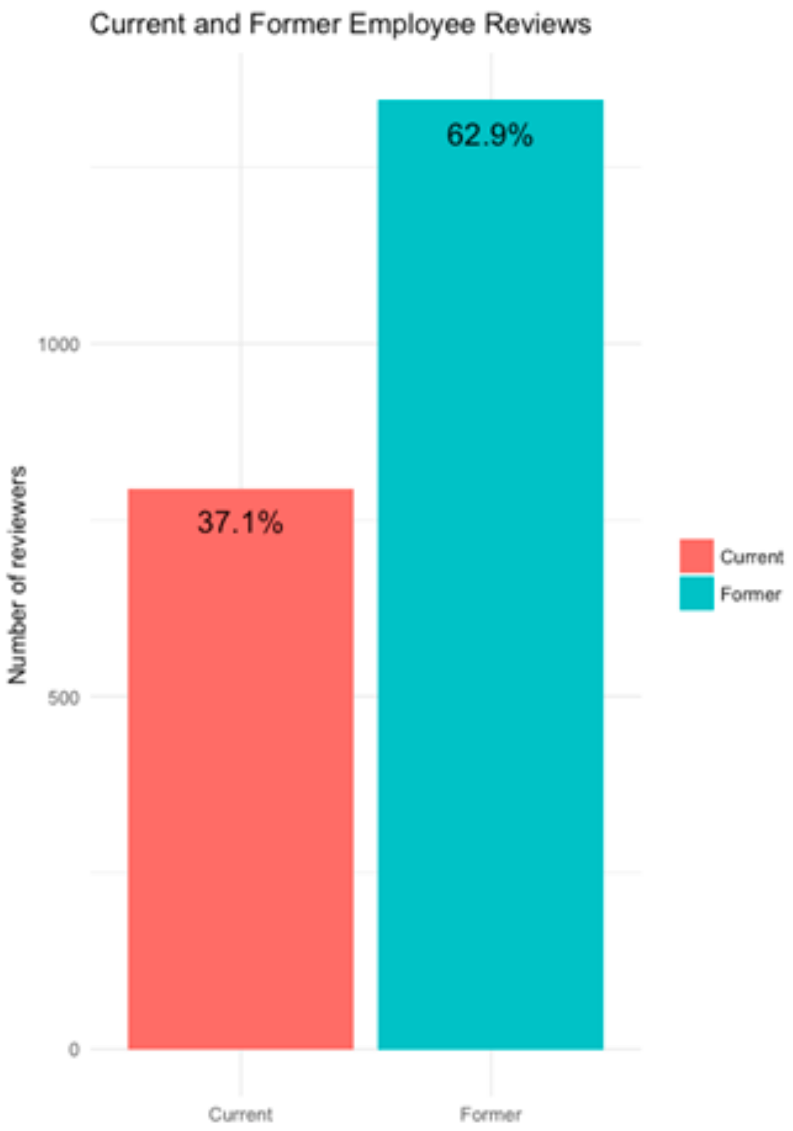
Sentiment Analysis of Fidelity Employee Reviews on Indeed.com

By: Aaron Stearns

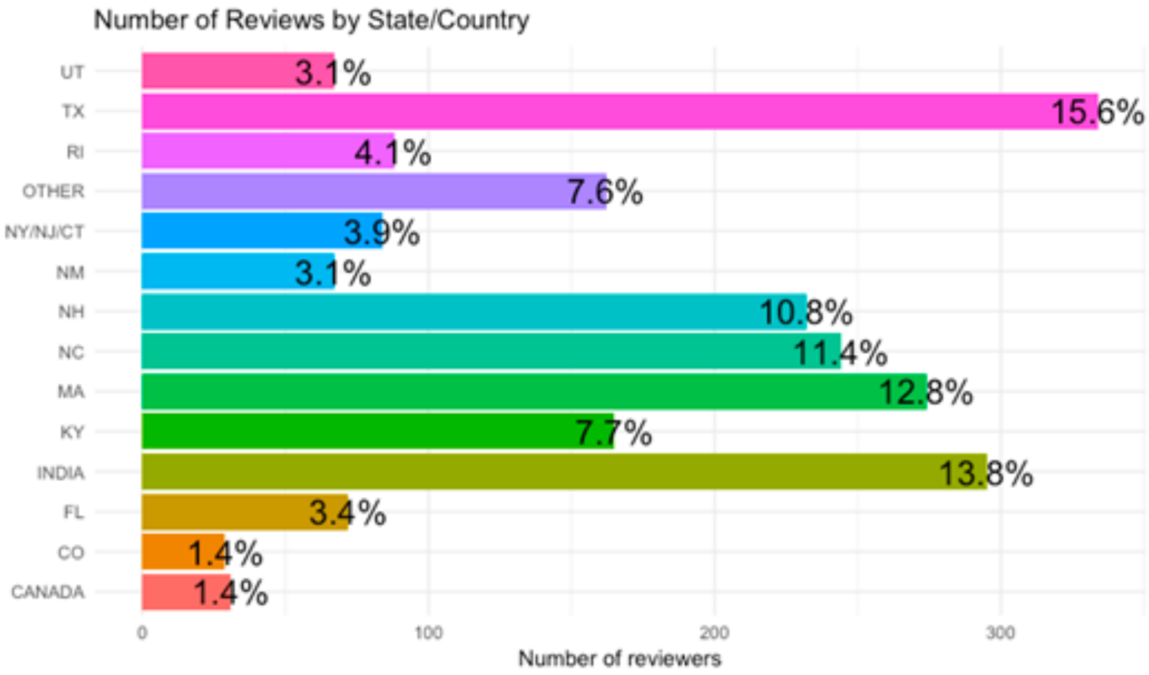
In order to better understand how current and former Fidelity Investments employees perceive the company, I used the R programming language to scrape 2,000+ Fidelity company reviews from 2012 to present on indeed.com, analyze the positive and negative sentiments expressed therein, and visualize the results. Part of the code is included on the last page of this document.

In this first plot, it can be observed that former employees leave nearly twice as many reviews as current employees:

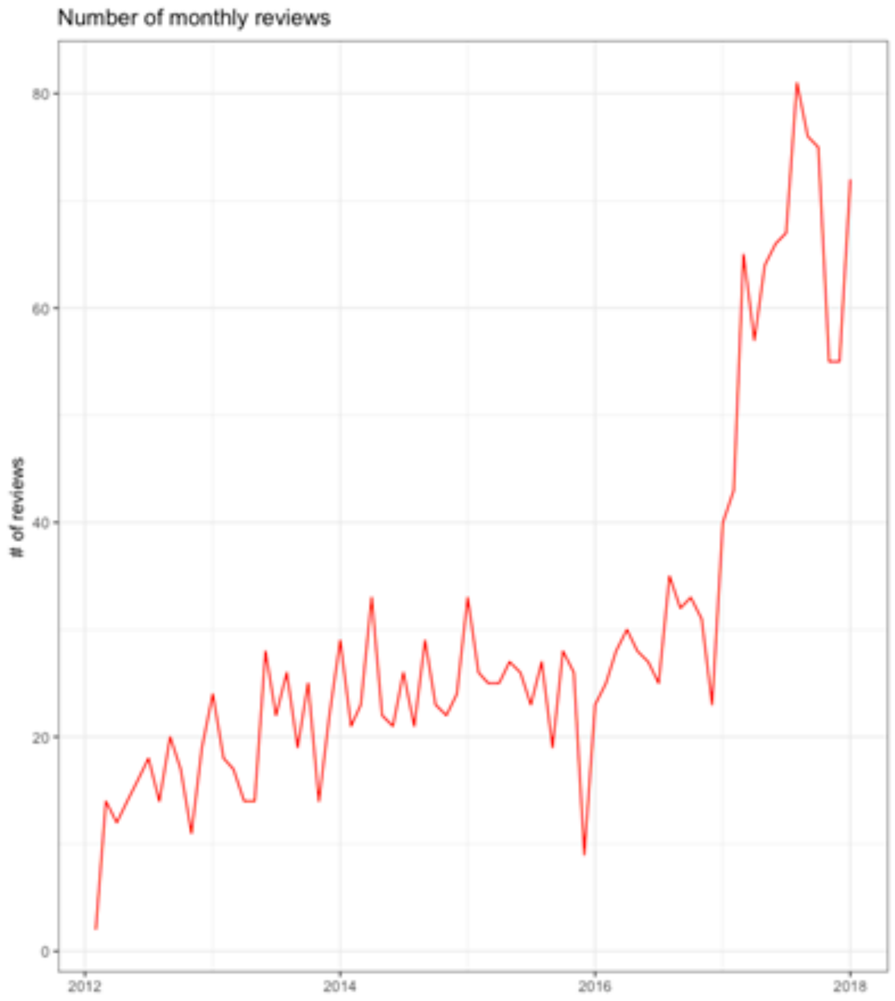


In order to visualize the locations of the submitted reviews, it was necessary to do some extensive formatting of the ‘job_location’ column of the data frame. This is because the reviewers were able to manually enter their locations, and so there are many irregularities and typos, (ex. ‘Westlake, TX’, ‘Westlak, Tx’, ‘WestlakeTX’, ‘Westlak, TEXAS’).

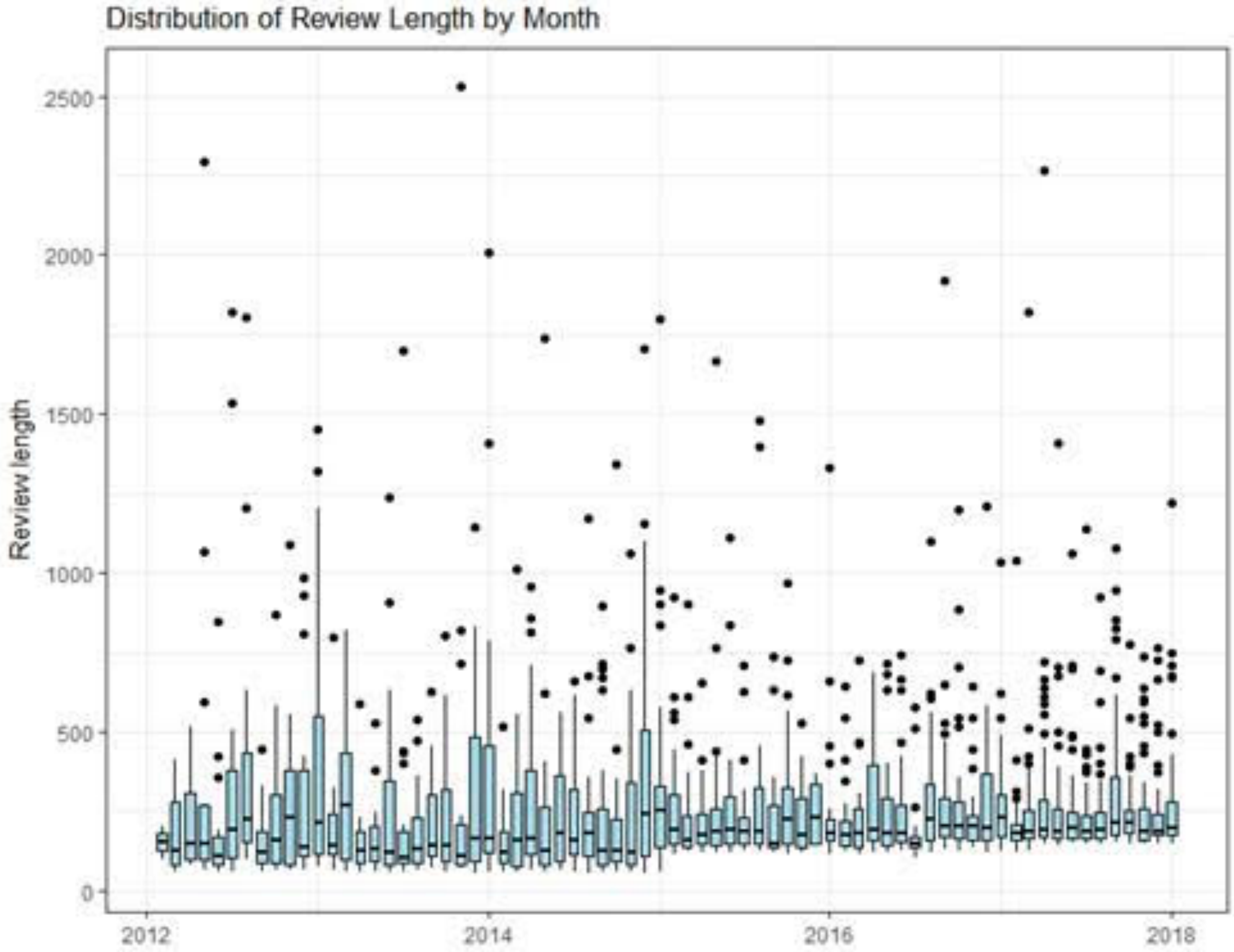
In order to clean up these locations, I used a series of SQL queries within R to create a new feature titled ‘state’. Due to the significant number of reviews from India and Canada, I also included these locations, as well as an ‘Other’ group that contains all of the employees who did not work at any of the major campuses.



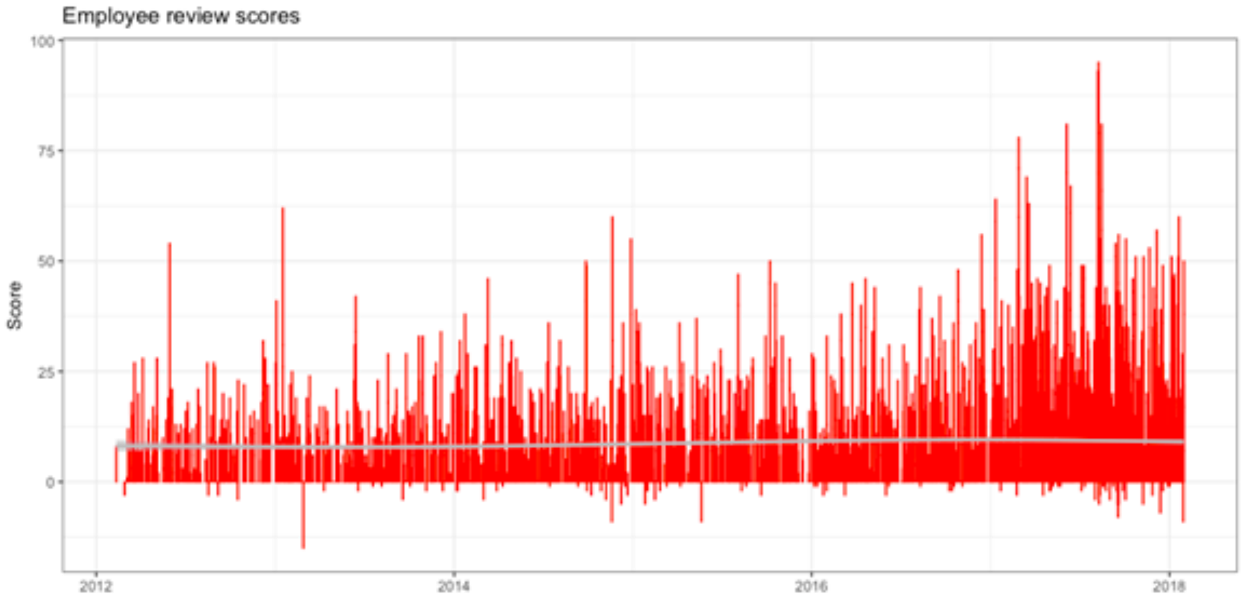
Next we look at the number of reviews submitted per month to Indeed:



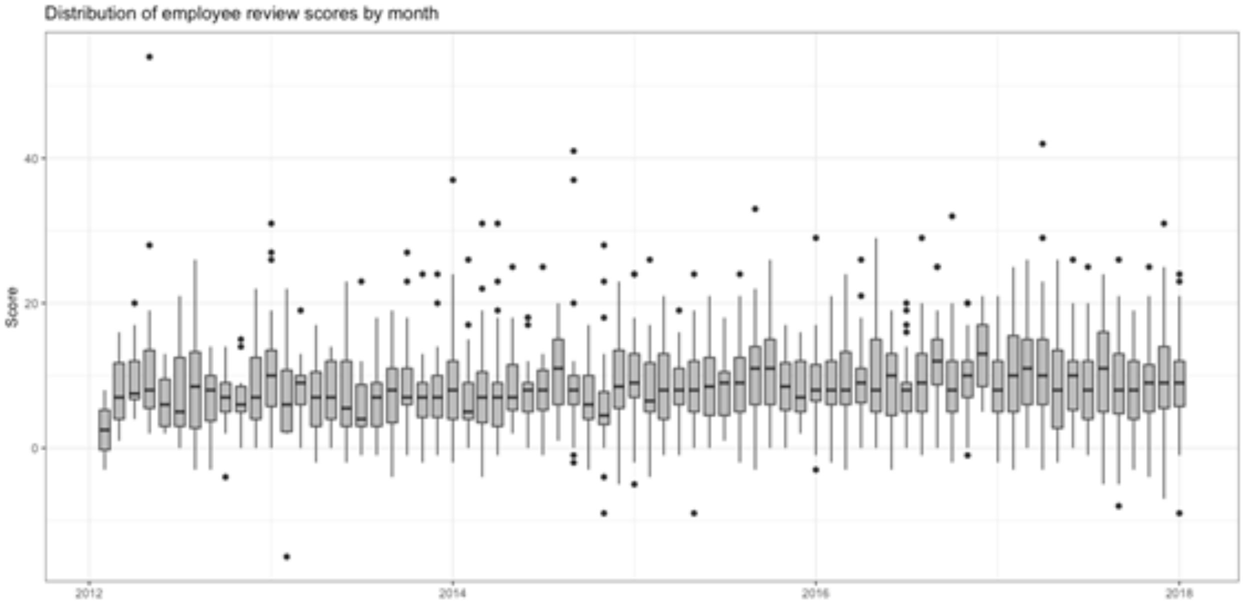
In this boxplot of review lengths by month, it can be observed that reviews have lengthened over time on average.



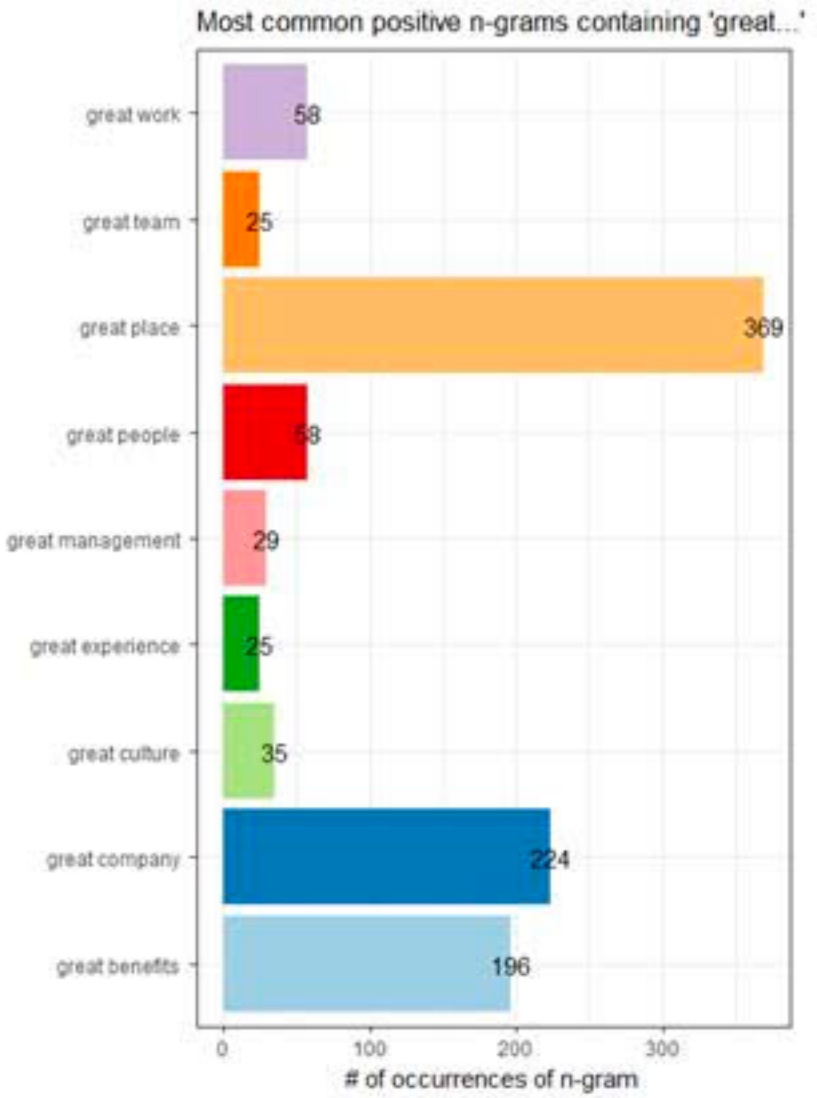
At this point I began the sentiment analysis on the review text by using the AFINN lexicon of positive and negative words which assigns a score from -5 to +5 to words based on the 'intensity' of the sentiment from negative to positive ('hate' is a -5 whereas 'amazing' is a +5). Then, I summed the positive and negative wordscores in each review and visualized all of the review scores in the graph below.
I included a trend line because although there is a significant increase in highly-positive employee reviews from 2016 to 2018, there are also many lower-scoring reviews in that period that balance out the more positive review scores.



Next, I visualized the distribution of employee review scores grouped by month:



In this final visualization, I split the review text into 2-word n-grams and focused on the most frequent positive n-grams containing the word "great" to see what people are saying about Fidelity:



<u>Conclusion:</u>

Overall, it can be seen that employee reviews tend to lean on the 'good' side. Very few reviews had negative sentiment scores, and a significant number of reviews had very high scores. As can be seen in the last graph, ‘great benefits’ is of significant importance to many reviewers. As an extension of this analysis, further insights might be made by looking into the job-specific and location-specific sentiments expressed by employees.

R script to scrape Fidelity reviews from Indeed.com

```
library(rvest)
library(purrr)
library(dplyr)
library(tidyr)
library(stringr)
library(lubridate)

# create sequence of numbers from 20 to 2140 that increments by 20, as
each indeed.com review page contains 20 reviews
nums <- seq(20, 2140, 20)

# use url_base and date_desc as strings that will be concatenated with
each number in the ‘nums’ sequence
url_base <- "https://www.indeed.com/cmp/Fidelity-Investments/reviews?
fcountry=ALL&start="
date_desc <- “&sort=date_desc&lang=en"

map_df(nums, function(i) {
  pg <- read_html(paste0(url_base, i, date_desc))
  data.frame(col = html_text(html_nodes(pg, “.cmp-review-date-
created”)), stringsAsFactors=FALSE)
}) -> dateCreated_indeed

map_df(nums, function(i) {
  pg <- read_html(paste0(url_base, i, date_desc))
  data.frame(col = html_text(html_nodes(pg, ".cmp-reviewer-job-location")),
             stringsAsFactors=FALSE)
}) -> jobLocation_indeed

map_df(nums, function(i) {
  pg <- read_html(paste0(url_base, i, date_desc))
  data.frame(col = html_text(html_nodes(pg, ".cmp-reviewer-job-title")),
             stringsAsFactors=FALSE)
}) -> jobTitle_indeed

map_df(nums, function(i) {
  pg <- read_html(paste0(url_base, i, date_desc))
  data.frame(col = html_text(html_nodes(pg, ".cmp-review-text")),
             stringsAsFactors=FALSE)
}) -> review_indeed

map_df(nums, function(i) {
  pg <- read_html(paste0(url_base, i, date_desc))
  data.frame(col = html_text(html_nodes(pg, ".cmp-review-description")),
             stringsAsFactors=FALSE)
}) -> reviews_indeed

map_df(nums, function(i) {
  pg <- read_html(paste0(url_base, i, date_desc))
  data.frame(col = html_text(html_nodes(pg, ".cmp-review-title span")),
             stringsAsFactors=FALSE)
}) -> titleSpan_indeed
```

```
## for some reason, every other row is blank in titleSpan_indeed
## remove blank rows
titleSpanInd <- seq(1, 4280, 2)
titleInd <- titleSpan_indeed[titleSpanInd, ]
titleInd <- data.frame(titleInd, stringsAsFactors = F)

df <- cbind(titleInd$titleInd,
            reviews_indeed$col,
            dateCreated_indeed$col,
            jobLocation_indeed$col,
            jobTitle_indeed$col)
df <- data.frame(df, stringsAsFactors = F)
colnames(df) <- c("review_title", "review", "date_created",
"job_location", "job_title")

df$current_former <- ifelse(grepl('Current', df$job_title), "Current",
"Former")
df$job_title <- str_replace_all(df$job_title, '(Current Employee)', "
")
df$job_title <- str_replace_all(df$job_title, '(Former Employee)', "
")
df$job_title <- str_replace_all(df$job_title, '[[:punct:]]', " ")

## format dates
df$date_created <- mdy(df$date_create)
```