Exploratory Data Analysis

Ada Lazuli

2022-07-09

Contents


```
library(tidyr)
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
library(ggplot2)
library(stringr)
library(tidytext)
library(forcats)
library(textdata)
```

Background

The data set was retrieved from COVID19 Fake News Dataset NLP and consists of tweets regarding COVID-19 news that are classified as *real* or *fake*. Consequently, this labeled data is ideal for developing a classification model to take in a tweet, pass its text through a function, and return a classification on whether the tweet contains real or fake news regarding covid-19. **NOTE**: There are two assumptions with the input data: 1. The tweet contains news, either real or fake 2. The tweet is concerning covid-19.

The source contained the following files:

1. **Constraint_Test.csv** - A comma separated file of 2140 tweets lacking classification.
2. **Constraint_Test.xlsx** - A MS-Excel formatted file of 2140 tweets lacking classification and appears to be identical to the test CSV in terms of text content.
3. **Constraint_Train.csv** - A comma separated file of 6420 tweets, each with a classification of real or fake.
4. **Constraint_Train.xlsx** - A MS-Excel formatted file of 6420 tweets, each with a classification of real or fake, identical to the train CSV.
5. **Constraint_Val.csv** - A comma separated file of 2140 tweets with classification. Initially it is not clear if the tweets in this file are duplicates of those found in the Test files.
6. **English_test_with_labels_.csv** - A comma separated file of 2140 tweets with classification that appear to be duplicates of those found in the Test files.
7. **test_ernie2.0_results.csv** - A comma separated file of 2140 rows that contain classification probabilities for whether the tweet in the Test files is real or fake. The file contains results after training ERNIE (Enhanced Representation Through Knowledge Integration) 2.0 on the training data (diptamath, 2021). Additional information on ERNIE 2,0 can be found here.

Initial Inspection of the data.

This analysis is performed on a system with the following specifications:

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] textdata_0.4.2 forcats_0.5.1  tidytext_0.3.3 stringr_1.4.0  ggplot2_3.3.6
## [6] dplyr_1.0.9    tidyr_1.2.0
##
## loaded via a namespace (and not attached):
```

```
##  [1] Rcpp_1.0.9        pillar_1.8.0      compiler_4.2.1    tokenizers_0.2.1
##  [5] tools_4.2.1       digest_0.6.29     evaluate_0.15     lifecycle_1.0.1
##  [9] tibble_3.1.8      gtable_0.3.0      lattice_0.20-45   pkgconfig_2.0.3
## [13] rlang_1.0.4       Matrix_1.4-1      cli_3.3.0         rstudioapi_0.13
## [17] yaml_2.3.5        xfun_0.31         fastmap_1.1.0     withr_2.5.0
## [21] janeaustenr_0.1.5 knitr_1.39        hms_1.1.1         fs_1.5.2
## [25] generics_0.1.3    vctrs_0.4.1       grid_4.2.1        tidyselect_1.1.2
## [29] glue_1.6.2        R6_2.5.1          fansi_1.0.3       rmarkdown_2.14
## [33] tzdb_0.3.0        readr_2.1.2       purrr_0.3.4       magrittr_2.0.3
## [37] ellipsis_0.3.2    SnowballC_0.7.0   scales_1.2.0      htmltools_0.5.3
## [41] colorspace_2.0-3  utf8_1.2.2        stringi_1.7.8     munsell_0.5.0
```

Due to the analysis being on a linux operating system and the content of the excel files seeming to be duplicated, **Constraint_Test.xlsx** and **Constraint_Train.xlsx** will be ignored. (Reading MS Excel files has difficult to resolve dependencies on linux)

Constraint Test [CSV]

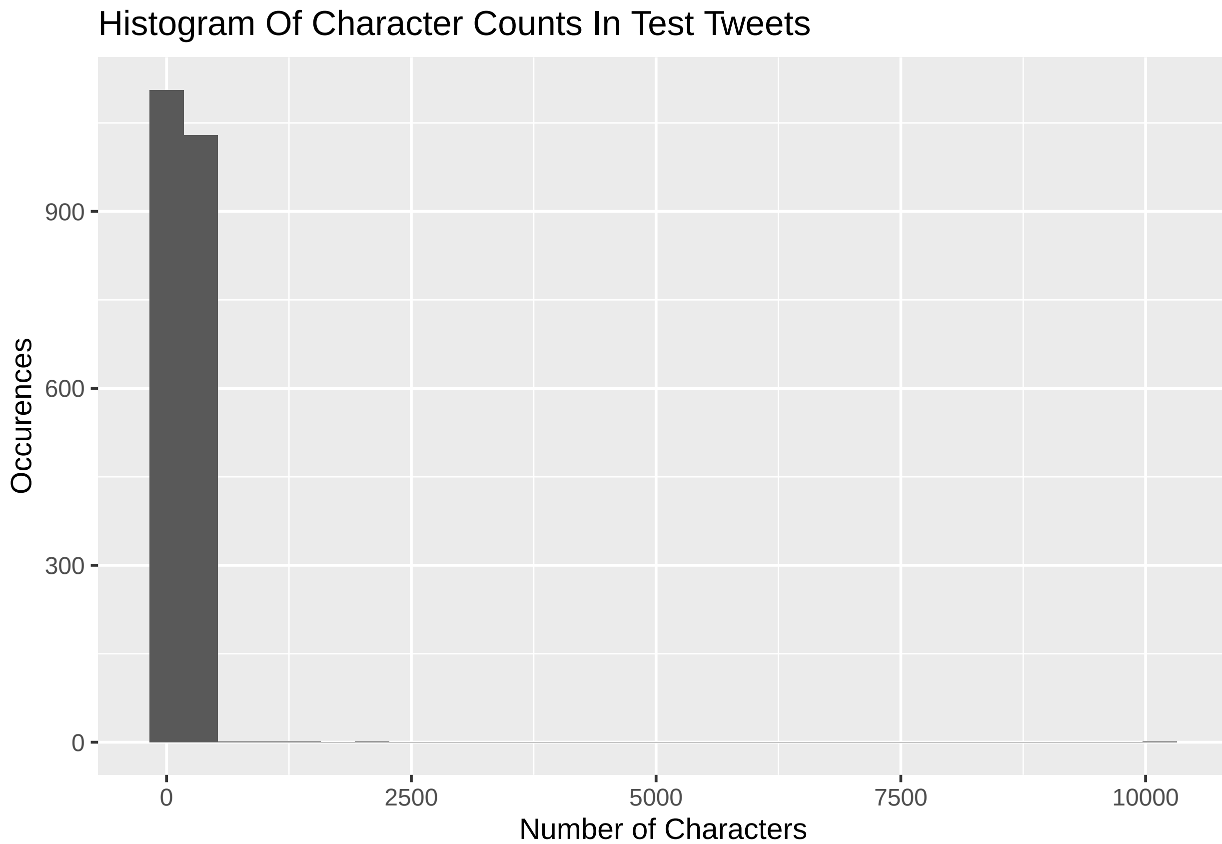
```
cnst_test <- read.csv("../data/Constraint_Test.csv")
glimpse(cnst_test)
```

```
## Rows: 2,140
## Columns: 2
## $ id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1~
## $ tweet <chr> "Our daily update is published. States reported 734k tests 39k n~
```

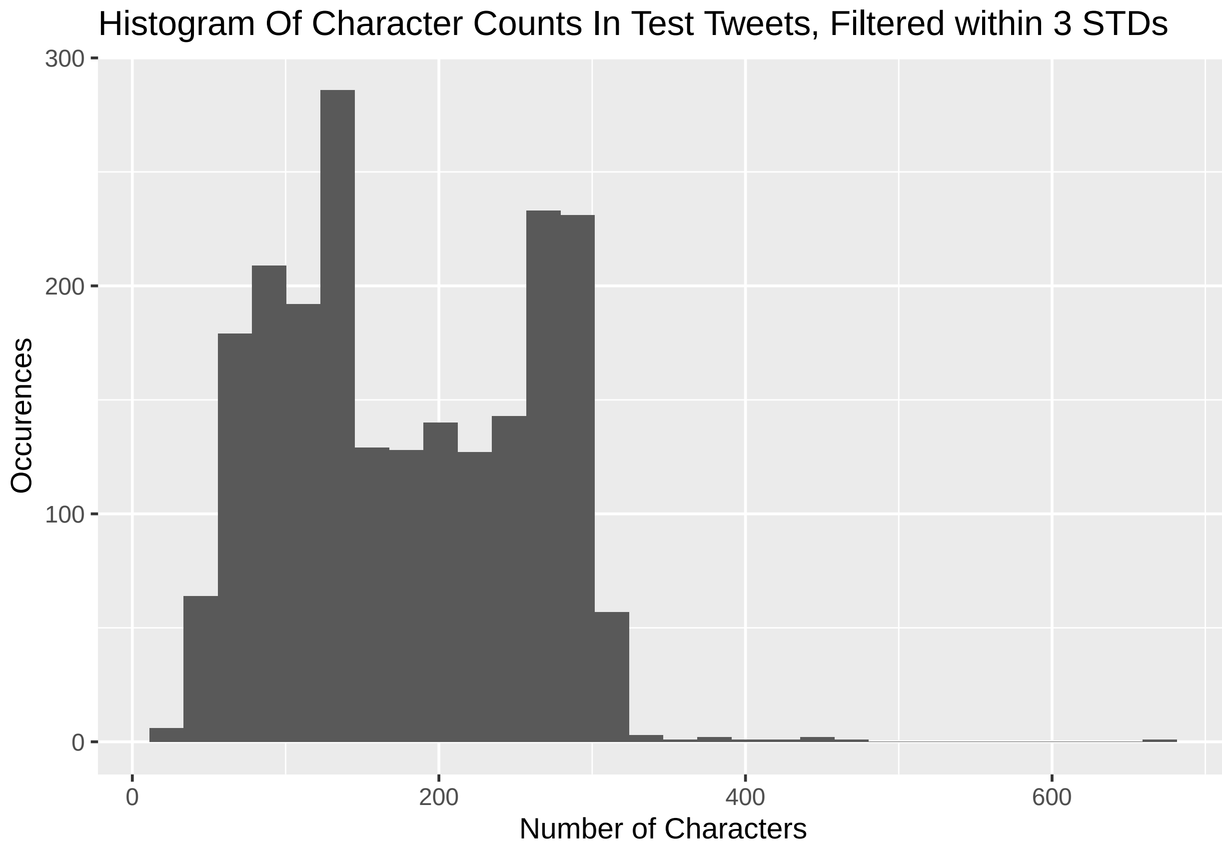
The file appears to have 2 columns and 2140 rows. The first column appears to be an ID that uniquely identifies each tweet, as it has 2140 unique values.

The tweet column appears to be just the text of various tweets. It is important to note that these tweets seem to be messy in that they have both non-alphanumeric characters present in the strings, as well as, pointers such as "@" and web addresses. To glean a sense of the variability in the length of the tweets, character counts will be uses initially:

```
count_df <- cnst_test %>% mutate(len = nchar(tweet))
ggplot(count_df, aes(x = len)) + geom_histogram() + labs(title = "Histogram Of Character Counts In Test
```




```
count_df <- cnst_test %>% mutate(len = nchar(tweet)) %>% filter(len < 3 * sd(len) + mean(len) ) %>% fil
ggplot(count_df, aes(x = len)) + geom_histogram() + labs(title = "Histogram Of Character Counts In Test
```



There appears to a few tweets that have character counts outside 3 standard deviations of the mean. The table below details the summary statistics for the character lengths of the test set.

```
summary((cnst_test %>% mutate(len = nchar(tweet)))$len)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    25.0   110.0   168.5   185.1   257.0 10171.0
```

Constraint Train [CSV]

```
cnst_train <- read.csv("../data/Constraint_Train.csv")
glimpse(cnst_train)
```

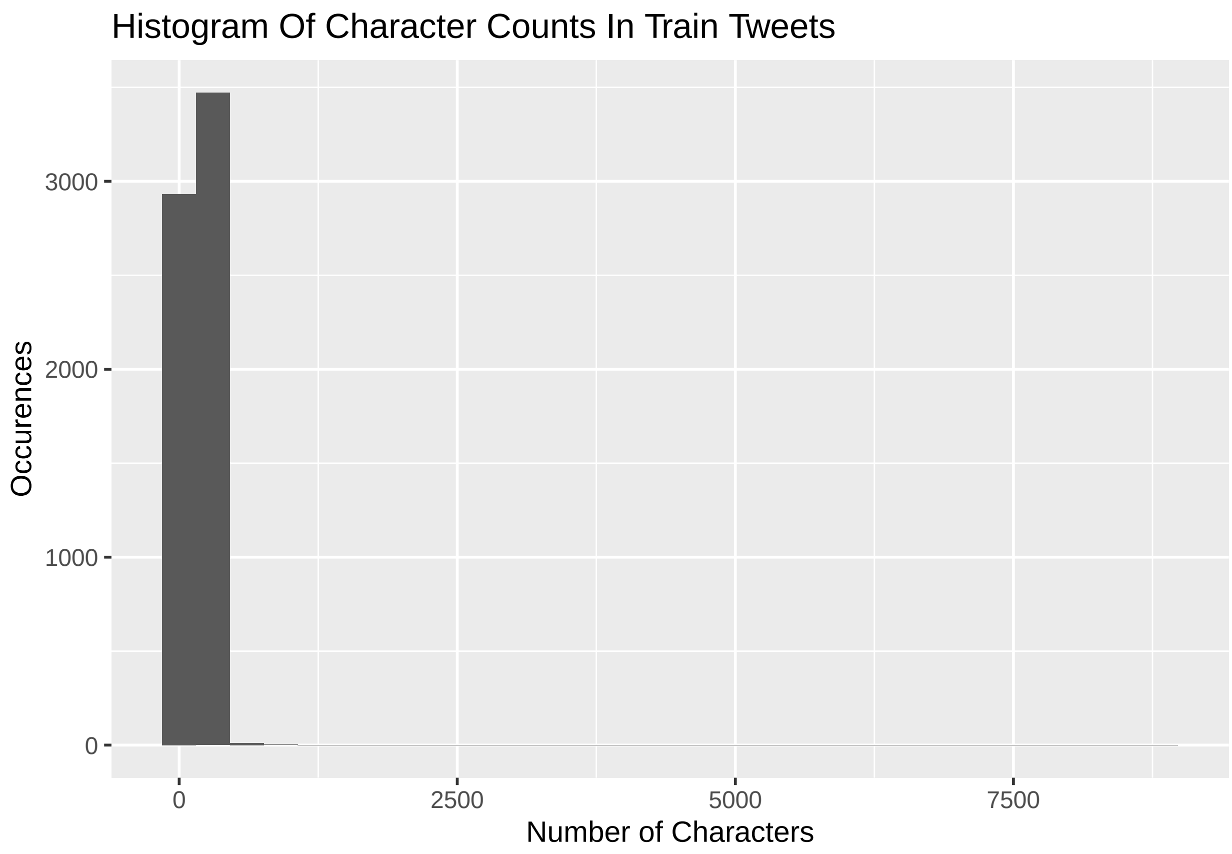
```
## Rows: 6,420
## Columns: 3
## $ id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1~
## $ tweet <chr> "The CDC currently reports 99031 deaths. In general the discrepa~
## $ label <chr> "real", "real", "fake", "real", "real", "real", "real", "fake", ~
```

The file appears to have 3 columns and 6420 rows. The first column appears to be an ID that uniquely identifies each tweet, as it has 6420 unique values.

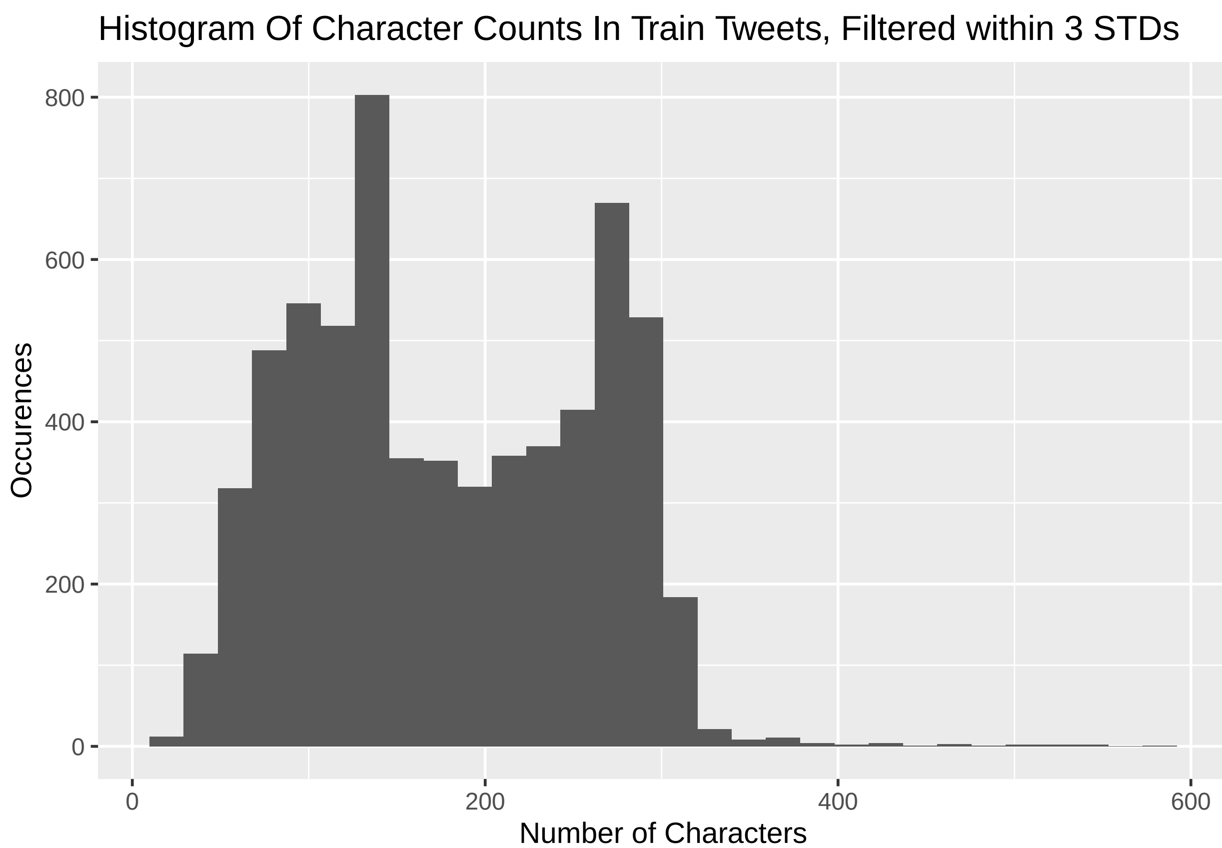
The tweet column appears to be just the text of tweets, just like the test file

Character Inspection

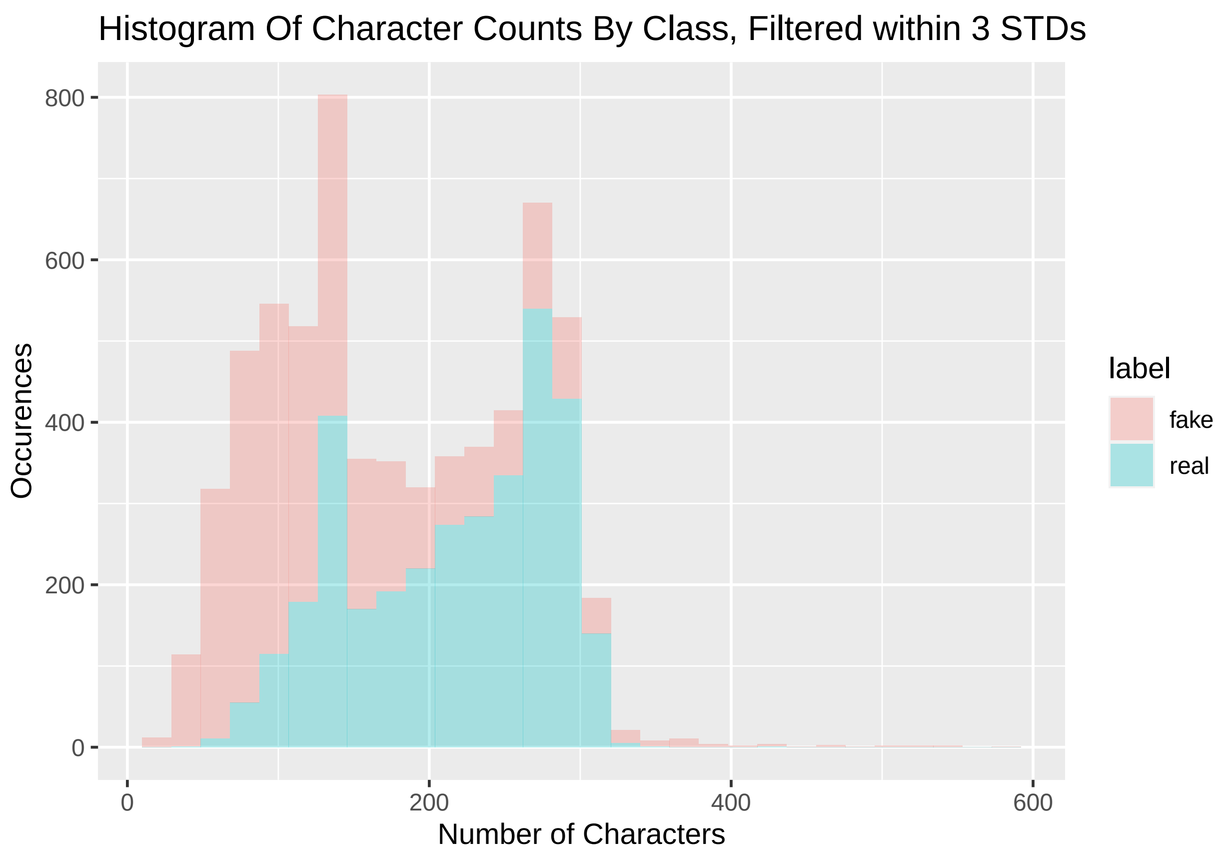
```
count_df <- cnst_train %>% mutate(len = nchar(tweet))
ggplot(count_df, aes(x = len)) + geom_histogram() + labs(title = "Histogram Of Character Counts In Trai
```




```
count_df <- cnst_train %>% mutate(len = nchar(tweet)) %>% filter(len < 3 * sd(len) + mean(len) ) %>% fil
ggplot(count_df, aes(x = len)) + geom_histogram() + labs(title = "Histogram Of Character Counts In Trai
```

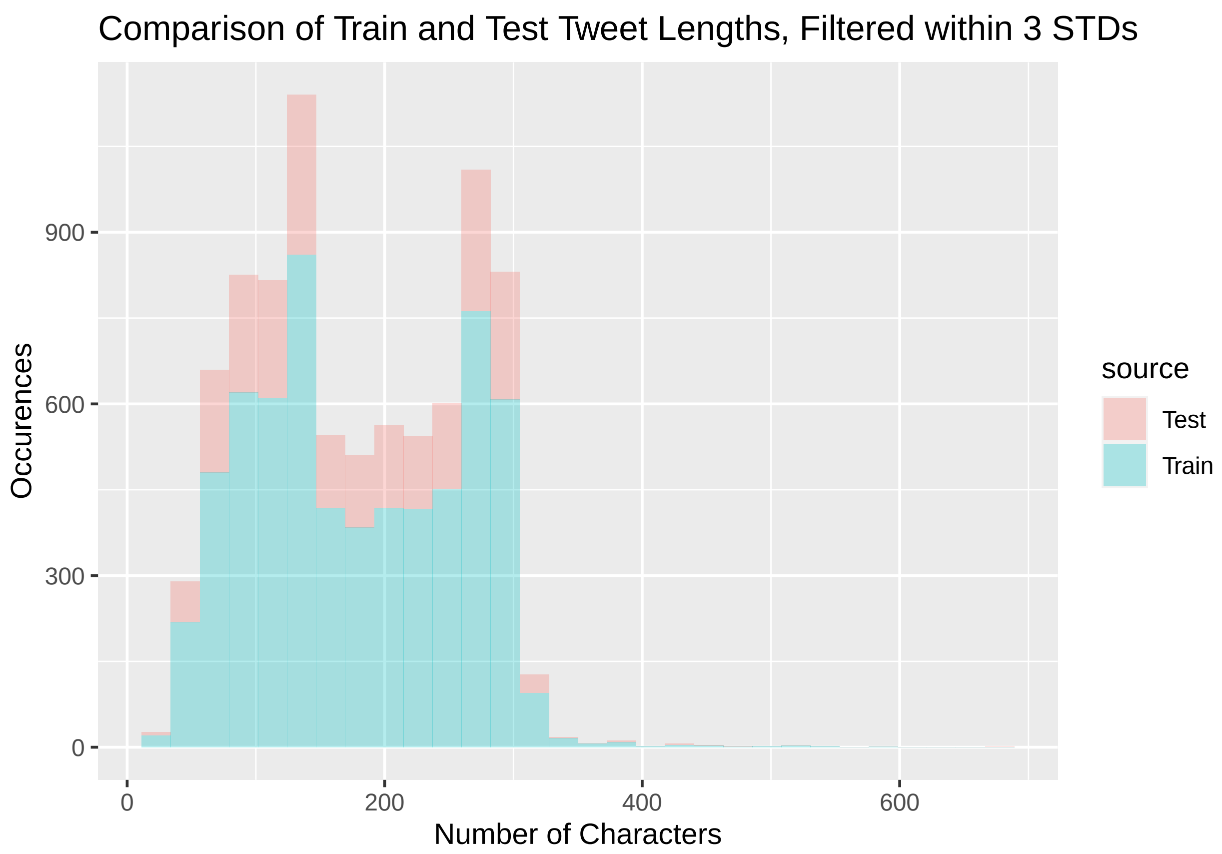



```
ggplot(count_df, aes(x = len, fill = label)) + geom_histogram(alpha = 0.3) + labs(title = "Histogram Of
```




```
# merge the cnst_train and cnst_test to look at their similarity

temp1 <- cnst_train %>% mutate(len = nchar(tweet)) %>% filter(len < 3 * sd(len) + mean(len) ) %>% filte
temp1$source <- "Train"
temp2 <- cnst_test %>% mutate(len = nchar(tweet)) %>% filter(len < 3 * sd(len) + mean(len) ) %>% filter
temp2$source <- "Test"
temp2$label <- "NA"
ggplot(rbind(temp1,temp2), aes(x = len, fill = source)) + geom_histogram( alpha = 0.3) + labs(title = "
```



There appears to a few tweets that have character counts outside 3 standard deviations of the mean and there is a difference in the distribution of the number of characters in *real* and *fake* tweets. Additionally, it appears that the training and test set have very similar distributions in terms of length.

The table below details the summary statistics for the character lengths of the training set.

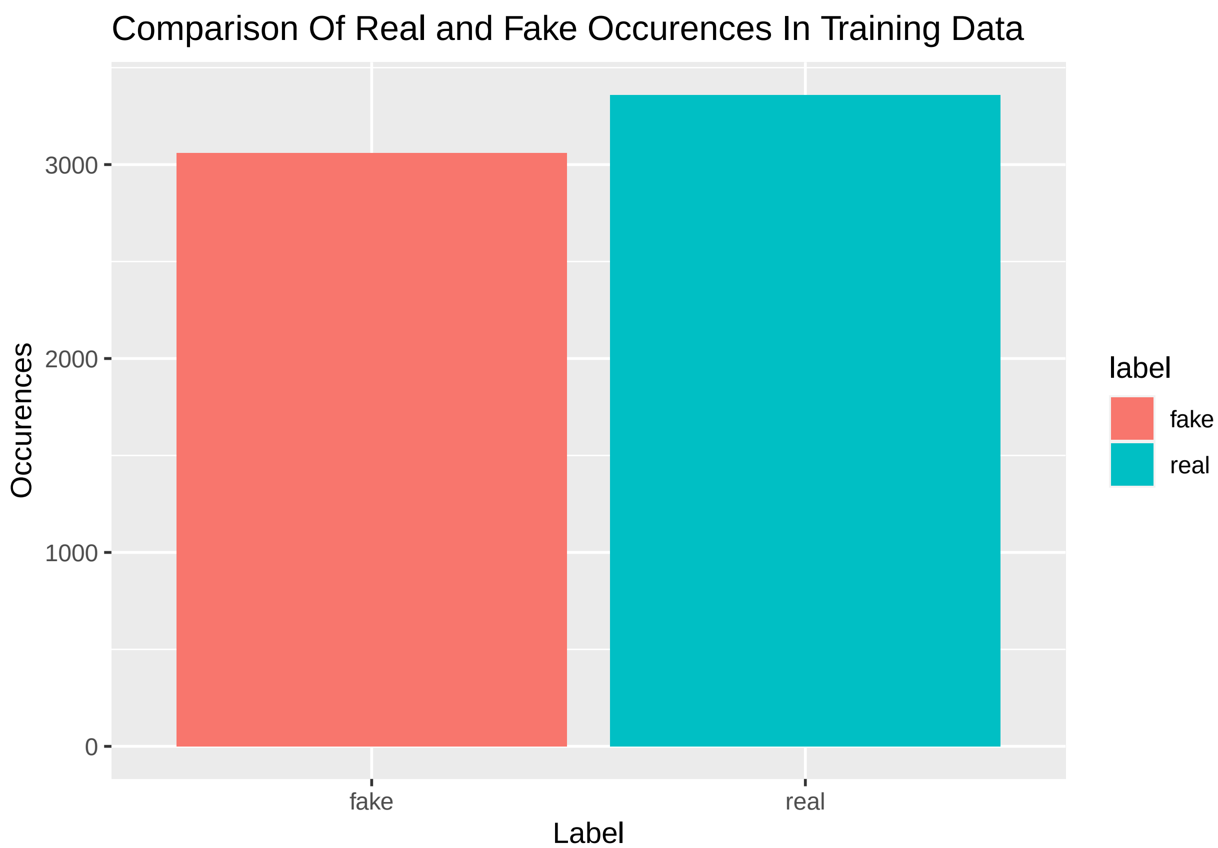
```
summary((cnst_train %>% mutate(len = nchar(tweet)))$len)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    18.0   111.0   169.0   181.7   255.2  8846.0
```

Class Balance

The training data contains a column title **label** that identifies whether the tweet contains real or fake news.

```
count_df <- cnst_train %>% count(label)
ggplot(count_df, aes(x = label, y = n, fill = label)) + geom_col() + labs(title = "Comparison Of Real an
```



It appears that class ratio for real and fake tweets is is balanced. 47.6635514 % of the data training tweets are *fake*, while the other 52.3364486 % are real.

Constraint Val [CSV]

```
cnst_val <- read.csv("../data/excess_files/Constraint_Val.csv")
glimpse(cnst_val)
```

```
## Rows: 2,140
## Columns: 3
## $ id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1~
## $ tweet <chr> "Chinese converting to Islam after realising that no muslim was ~
## $ label <chr> "fake", "fake", "fake", "fake", "real", "real", "real", "real", ~
```

The file appears to have 3 columns and 2140 rows. The first column appears to be an ID that uniquely identifies each tweet, as it has 2140 unique values.

Does Constraint Val match Constraint Test?

Given the similarity to **Constant_Test.csv**, a test is executed below to determine if the two files are duplicates:

```r
cnst_val %>% inner_join(cnst_test, by = "tweet")
```

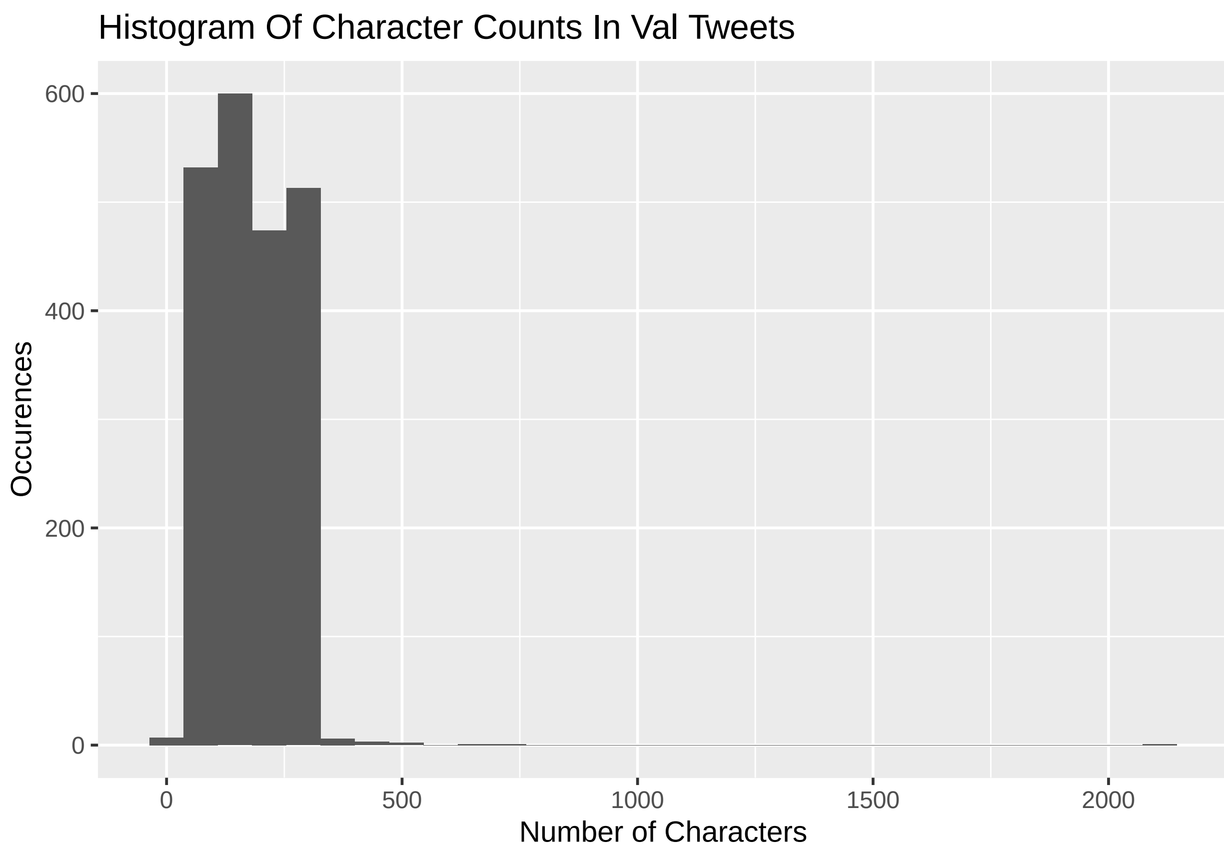
```
## [1] id.x  tweet label id.y
## <0 rows> (or 0-length row.names)
```

It appears that **Constraint_Test** and **Constraint_Val** do not have a single matching tweet.

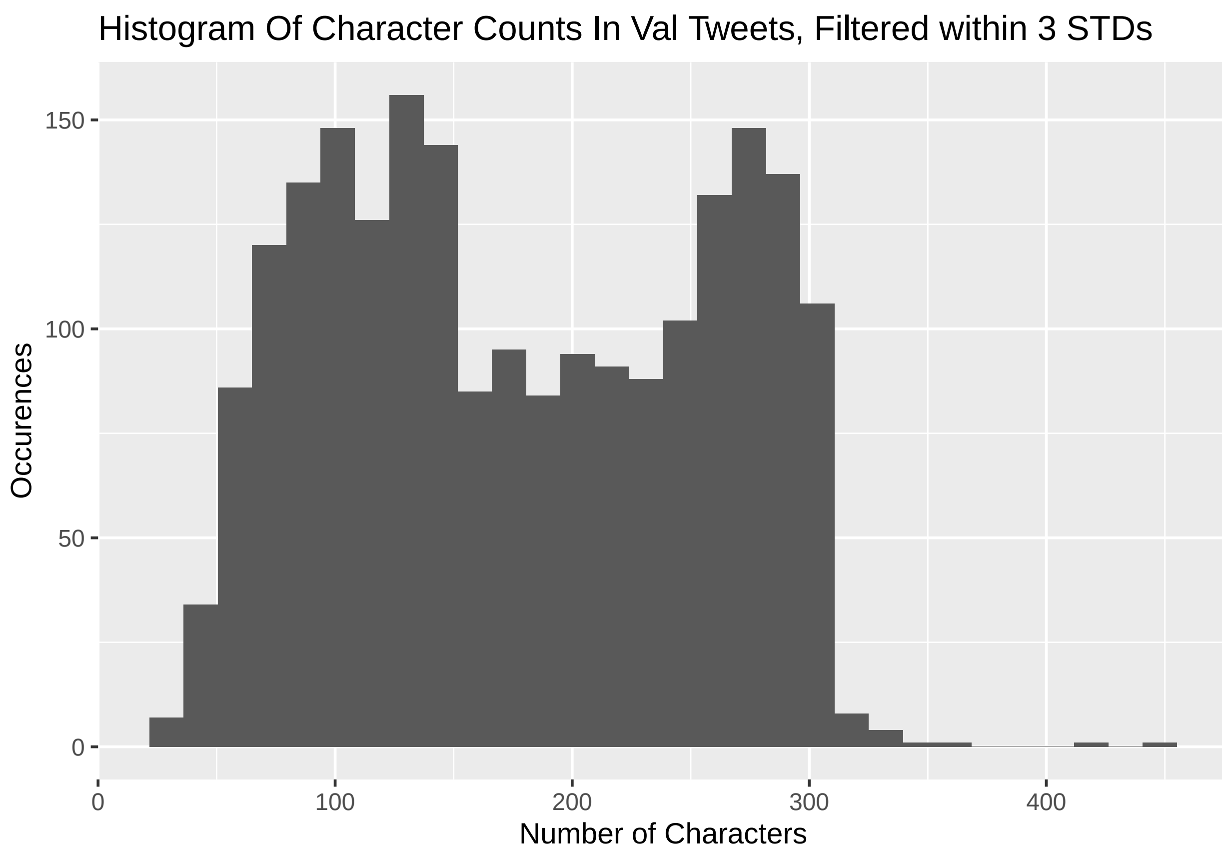
Character Length

This section studies the distribution of character lengths.

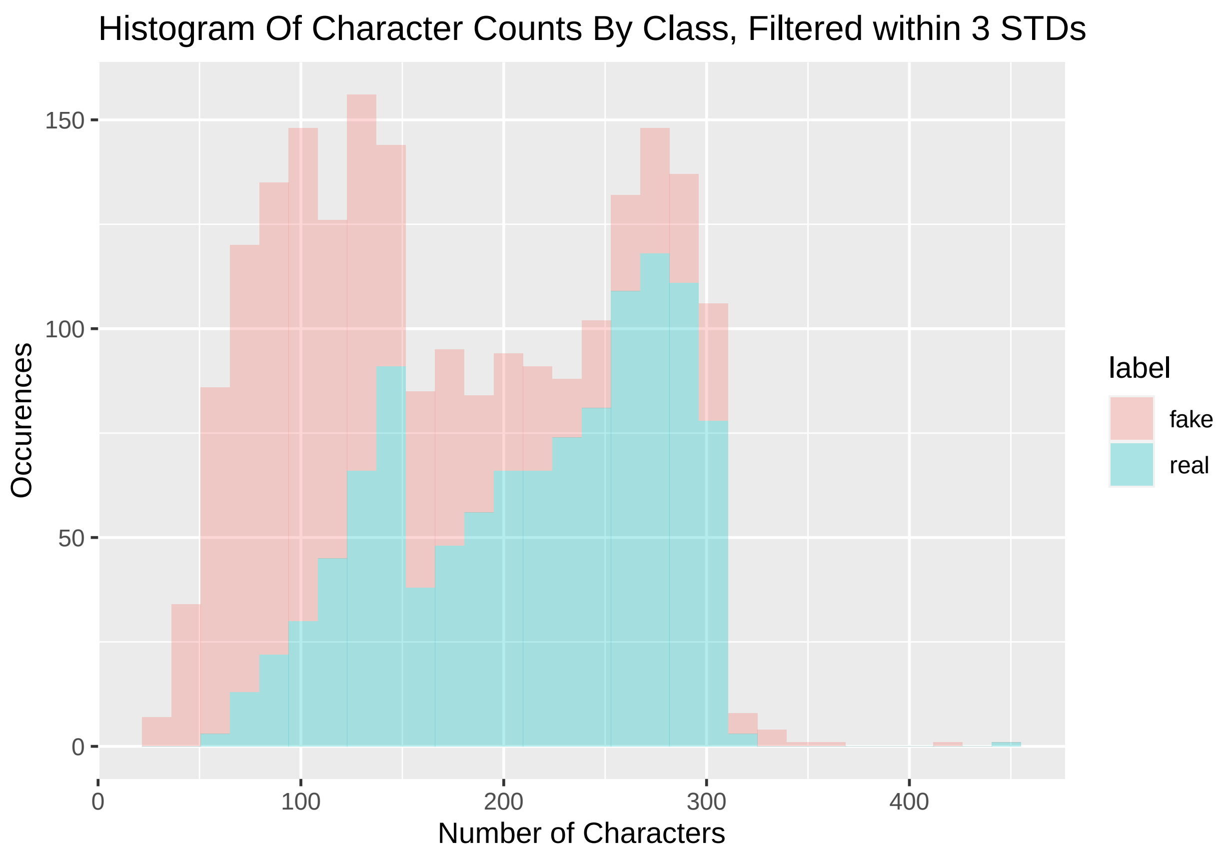
```r
count_df <- cnst_val %>% mutate(len = nchar(tweet))
ggplot(count_df, aes(x = len)) + geom_histogram() + labs(title = "Histogram Of Character Counts In Val 
```




```r
count_df <- cnst_val %>% mutate(len = nchar(tweet)) %>% filter(len < 3 * sd(len) + mean(len) ) %>% filt
ggplot(count_df, aes(x = len)) + geom_histogram() + labs(title = "Histogram Of Character Counts In Val 
```




```
ggplot(count_df, aes(x = len, fill = label)) + geom_histogram(alpha = 0.3) + labs(title = "Histogram Of
```

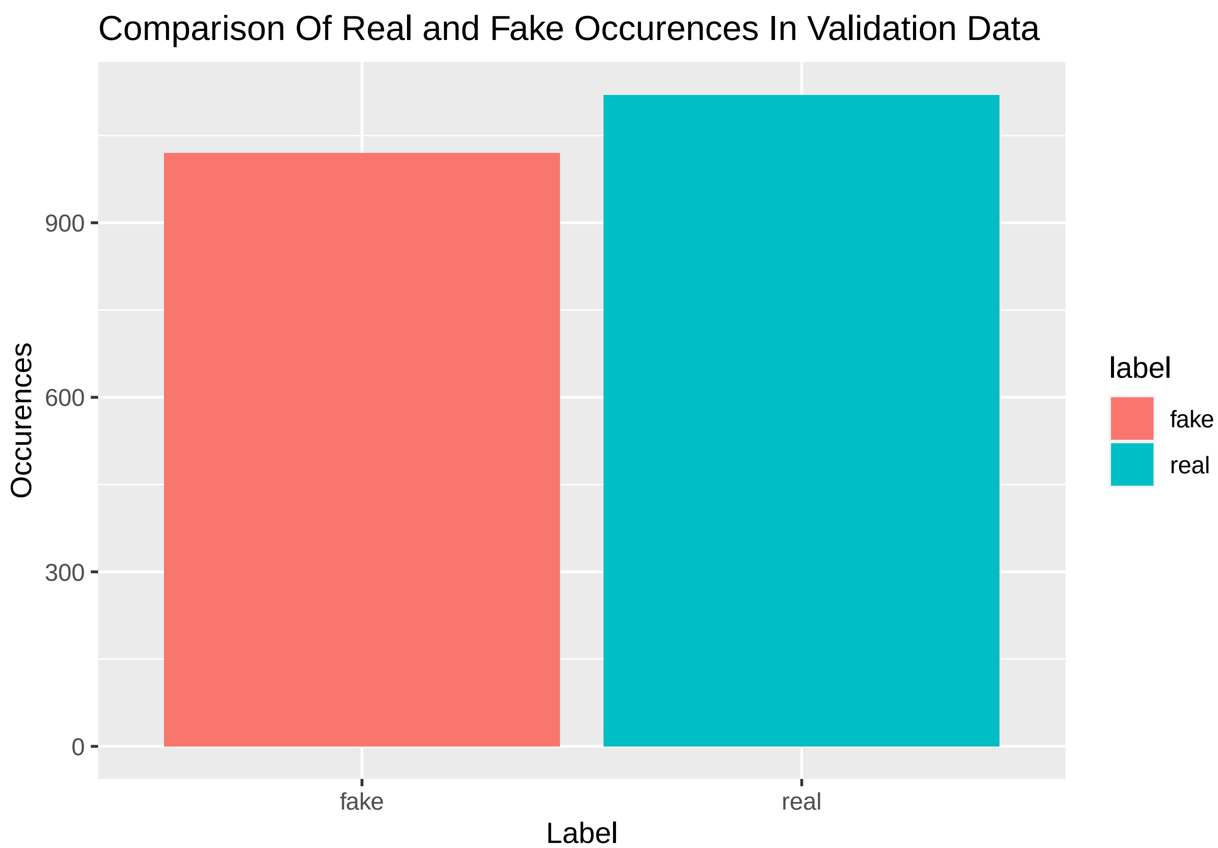


The distribution of character lengths appears to be similar to the the other files. It is also apparent that there seems to be a difference between the lengths of characters when split by *Real* or *Fake*

Class Balance

The validation data contains a column title **label** that identifies whether the tweet contains real or fake news.

```
count_df <- cnst_val %>% count(label)
ggplot(count_df, aes(x = label, y = n, fill = label)) + geom_col() + labs(title = "Comparison Of Real ar
```



It appears that class ratio for real and fake tweets is is balanced. 47.6635514 % of the data training tweets are *fake*, while the other 52.3364486 % are real.

The split of labels in the **validation** set seems to match the split of labels in the **training set**

English Test With Labels [CSV]

```
eng_test <- read.csv("../data/english_test_with_labels.csv")

glimpse(eng_test)
```

```
## Rows: 2,140
## Columns: 3
## $ id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1~
## $ tweet <chr> "Our daily update is published. States reported 734k tests 39k n~
## $ label <chr> "real", "fake", "fake", "real", "real", "real", "real", "real", ~
```

The file appears to have 3 columns and 2140 rows. The first column appears to be an ID that uniquely identifies each tweet, as it has 2140 unique values.

Does It Match Constraint Test?

Given the similarity to **Constant_Test.csv**, a test is executed below to determine if the two files are duplicates:

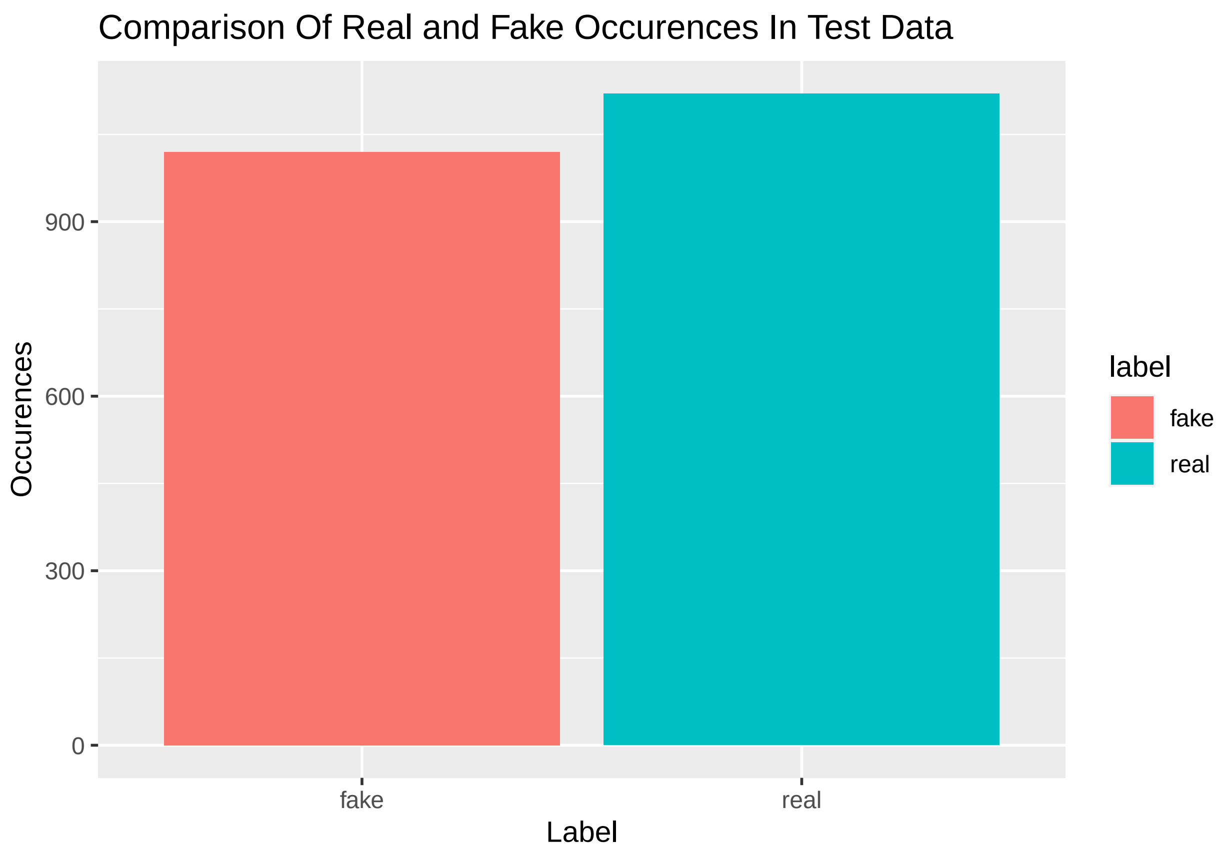
```
test <- eng_test %>% inner_join(cnst_test, by = "tweet")
```

Given that the inner join had 2140 and **Constraint_Test** had 2140. It appears that this file is a copy!

Class Balance

The test data contains a column title **label** that identifies whether the tweet contains real or fake news.

```
count_df <- eng_test %>% count(label)
ggplot(count_df, aes(x = label, y = n, fill = label)) + geom_col() + labs(title = "Comparison Of Real ar
```



It appears that class ratio for real and fake tweets is is balanced. 47.6635514 % of the data training tweets are *fake*, while the other 52.3364486 % are real.

Test ERNIE2.0 Results [CSV]

```
ernie <- read.csv("../data/excess_files/test_ernie2.0_results.csv")
glimpse(ernie)
```

```
## Rows: 2,140
## Columns: 3
## $ X             <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16~
```

```
## $ Model4_class0 <dbl> 0.9993706346, 0.0002565508, 0.0002471037, 0.9993759990, ~
## $ Model4_class1 <dbl> 0.0006293455, 0.9997434020, 0.9997529387, 0.0006240553, ~
```

The file appears to have 3 columns and 2140 rows. It seems that this file contains the results of passing **Constraint_Test** through ERNIE 2.0.

Confusion Matrix

	Test Real	Test Fake
Classified Real	1083	30
Classified Fake	37	990

ERNIE 2.0 performed very well and had an accuracy of **96.8691589**%.

Data Cleaning

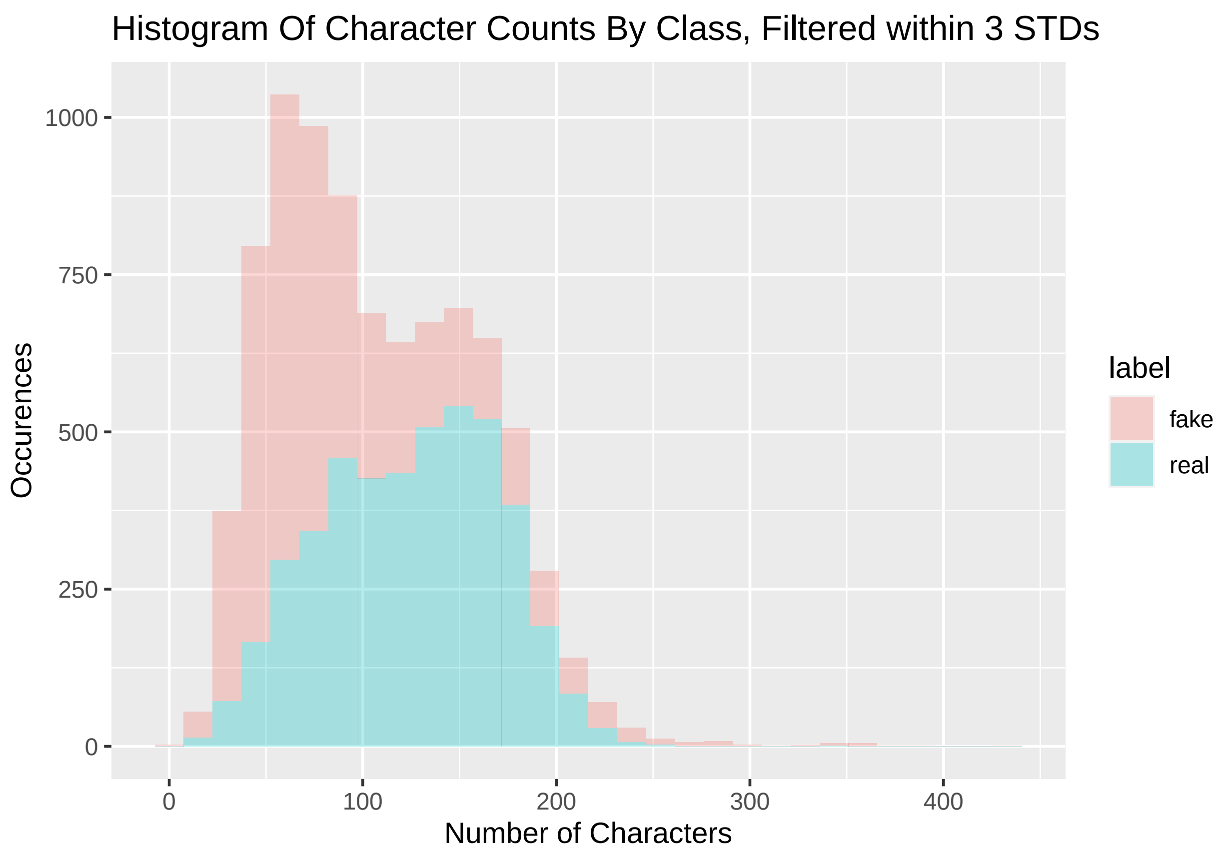
The data was cleaned and compiled into a single dataset provided by Winfrey "John" Johnson.

```
tweets <- read.csv("../data/tweets_prepped_no_http.csv")
```

Exploration

Revisit Character Lengths Post Cleaning

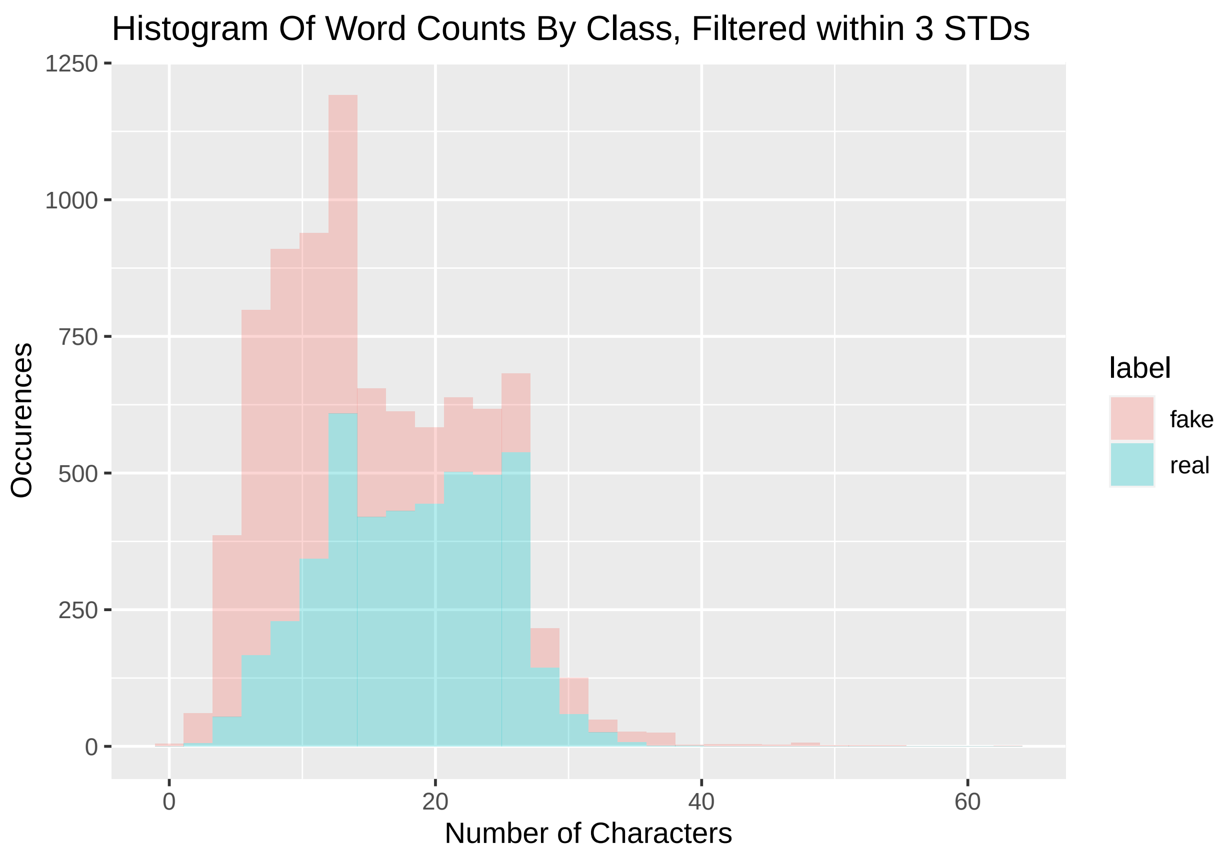
```
count_df <- tweets %>% mutate(len = nchar(tweet))  %>% filter(len < 3 * sd(len) + mean(len) ) %>% filte
ggplot(count_df, aes(x = len, fill = label)) + geom_histogram(alpha = 0.3) + labs(title = "Histogram Of
```



It seems that the difference between real and fake is less pronounced post cleaning, however, real tweets seem to be affected the most. This cold be an indicator that real tweets tend to use stopwords, symbols, or URLs more often.

Word Length Comparison

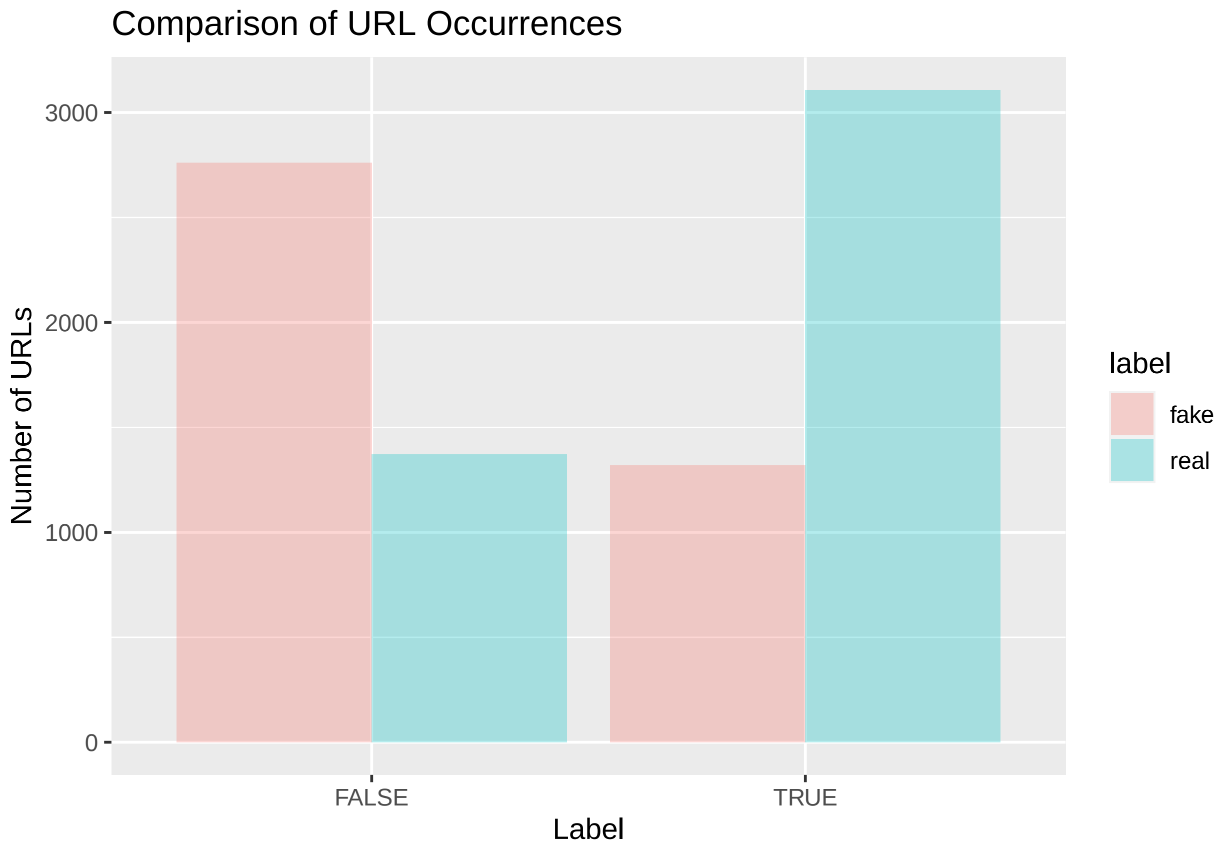
```
count_df <- tweets %>% mutate(words = str_count(tweet, " "))  %>% filter(words < 3 * sd(words) + mean(wo
ggplot(count_df, aes(x = words, fill = label)) + geom_histogram(alpha = 0.3) + labs(title = "Histogram O
```



It seems as though *fake* tweets have a tendency for shorter tweets.

Looking at URLs

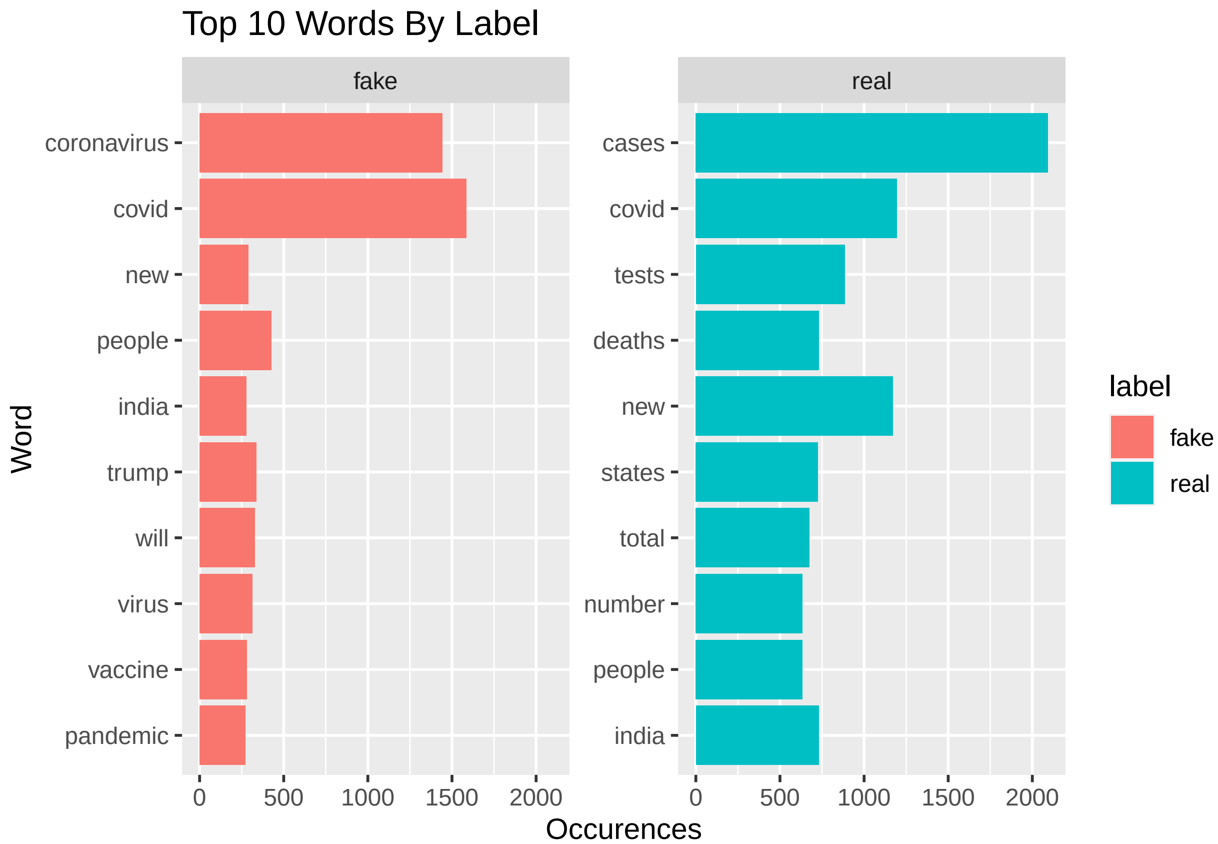
```
#count_df <- tweets %>% filter(grepl('http', tweet)) %>% count(label) %>% inner_join(tweets %>% count(l
ggplot(tweets, aes(x = link, fill = label)) + geom_bar(alpha = 0.3, position = "dodge") + labs(title =
```



It seems that URLs are significantly more common in *real* tweets compared to *fake* tweets,

Top 10 Words By Label

```
count_df <- tweets %>% unnest_tokens(term, tweet) %>% count(term, label) %>% group_by(label) %>% top_n(
ggplot(count_df, aes(x = sorted, y = n, fill = label)) + geom_col() + facet_wrap(~ label, scales = "fre
```



It is interesting to note that **Trump** is one of the top 10 words for fake news tweets. This could lend credence to political motivations for fake news tweets in the data. Again, **HTTPS** is very prominent for real tweets, but it is also very common in fake tweets.

Sentiment Analysis

For sentiment analysis, the individual words will need to be remain unnested. The next code block creates a new dataframe of unnested tweets for sentiment analysis.

A few different corpora are used for sentiment analysis, details on the corpora available in `tidytext` is well documented here.

```
unnested_tweets <- tweets %>% unnest_tokens(word, tweet) # note: using word instead of term to help wit
```

Using the bing Corpus

A glimpse of the `bing` corpus is provide below. It has 6783 words with 2 possible sentiment classifications.

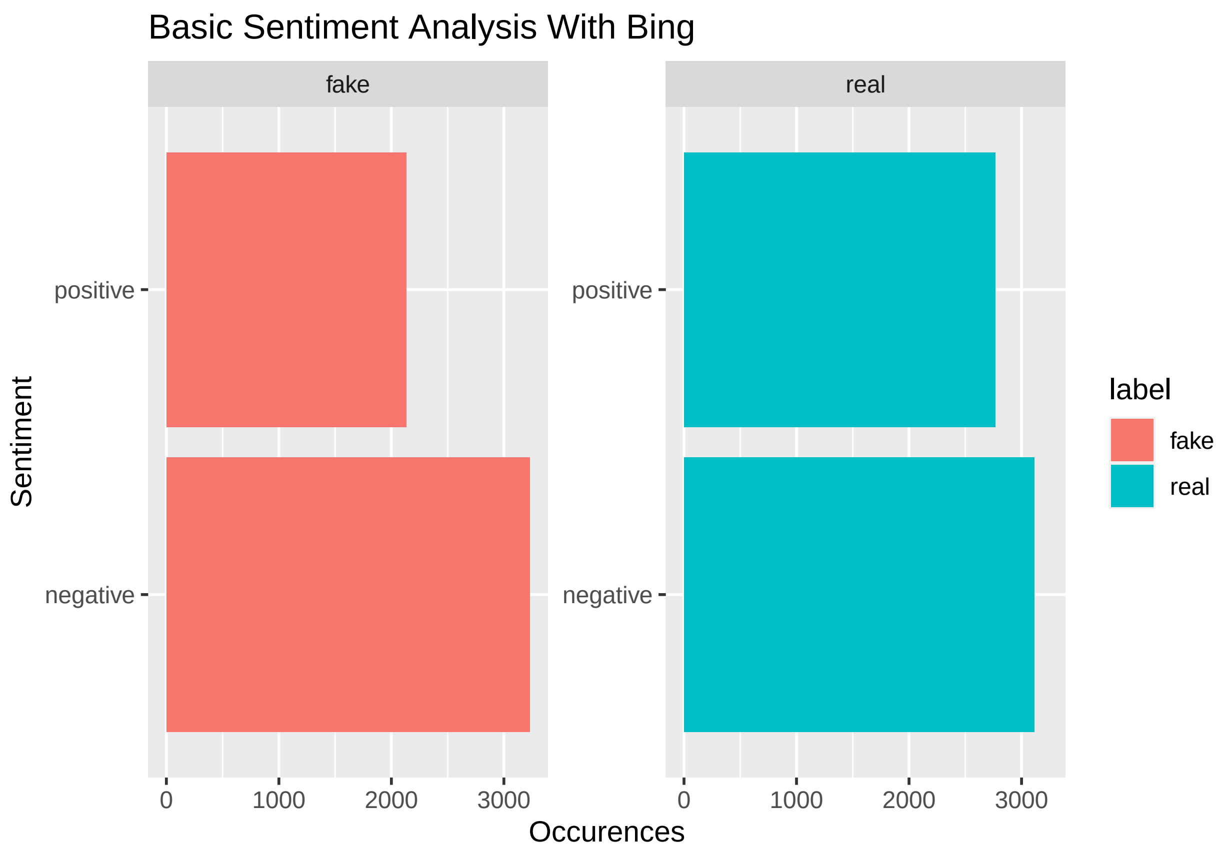
```
glimpse(get_sentiments("bing"))
```

```
## Rows: 6,786
## Columns: 2
## $ word      <chr> "2-faces", "abnormal", "abolish", "abominable", "abominably"~
## $ sentiment <chr> "negative", "negative", "negative", "negative", "negative", ~
```

```
table(get_sentiments("bing")$sentiment)
```

```
##
## negative positive
##     4781     2005
```

```
count_df <- unnested_tweets %>% inner_join(get_sentiments("bing")) %>% count(label, sentiment)
ggplot(count_df, aes(x = sentiment, y = n, fill = label)) + geom_col() + facet_wrap(~ label, scale = "f
```



It seems that fake tweets are more likely to be negative than positive, but the distinction seems weak.

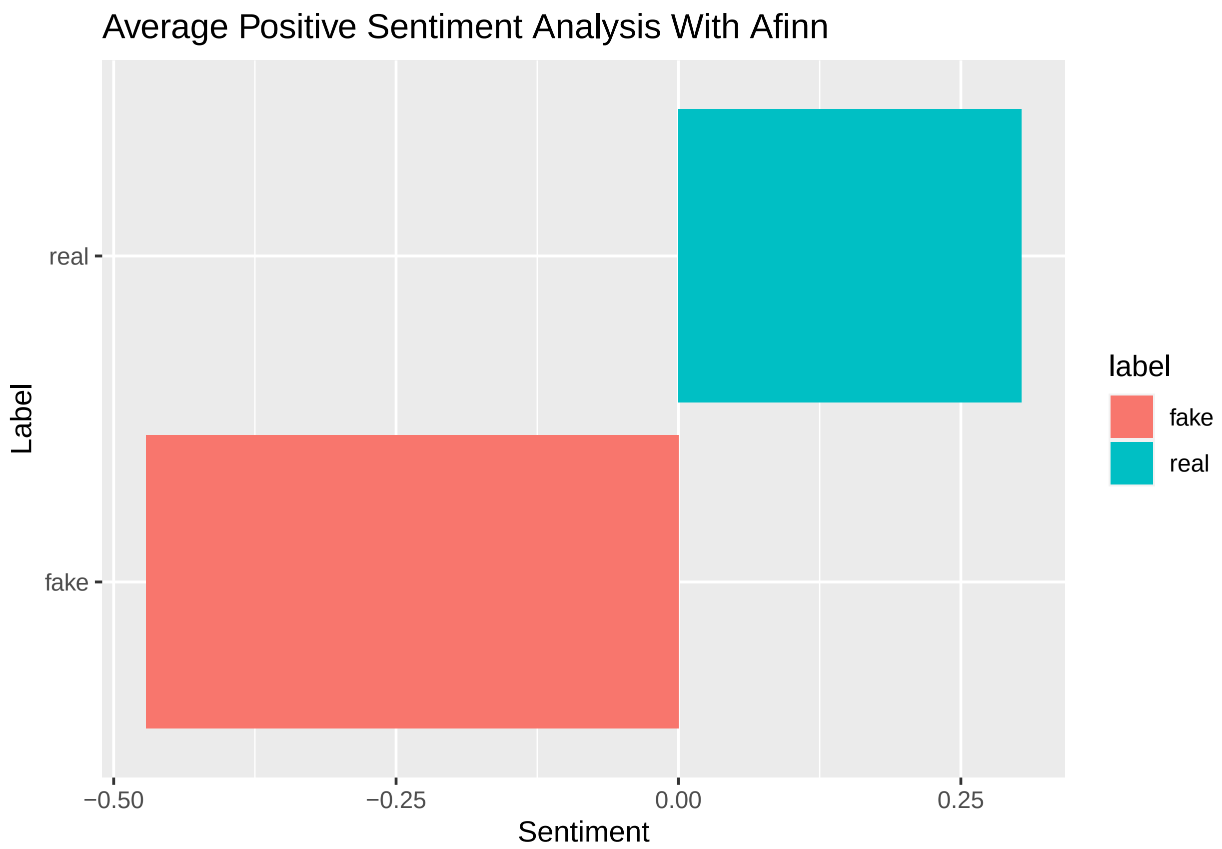
Using the afinn Corpus

A glimpse of the `afinn` corpus is provide below. It has 2477 words and provides an integer value of positive or negative. **NOTE**: If this line errors out, open an interactive R session, run `library(tidytext)` followed by `get_sentiments("afinn")`. The prompt to download the corpus is restricted to an interactive mode.

```
glimpse(get_sentiments("afinn"))
```

```
## Rows: 2,477
## Columns: 2
## $ word  <chr> "abandon", "abandoned", "abandons", "abducted", "abduction", "ab~
## $ value <dbl> -2, -2, -2, -2, -2, -2, -3, -3, -3, -3, 2, 2, 1, -1, -1, 2, 2, 2~
```

```
count_df <- unnested_tweets %>% inner_join(get_sentiments("afinn")) %>% group_by(label) %>% summarise(se
ggplot(count_df, aes(x = label, y = sentiment, fill = label)) + geom_col() + coord_flip() + labs(title =
```



Again, fake tweets seem to have a strong negative sentiment.

Using the loughran Corpus

A glimpse of the `loughran` corpus is provide below. It has 3917 words with 6 possible sentiment classifications.

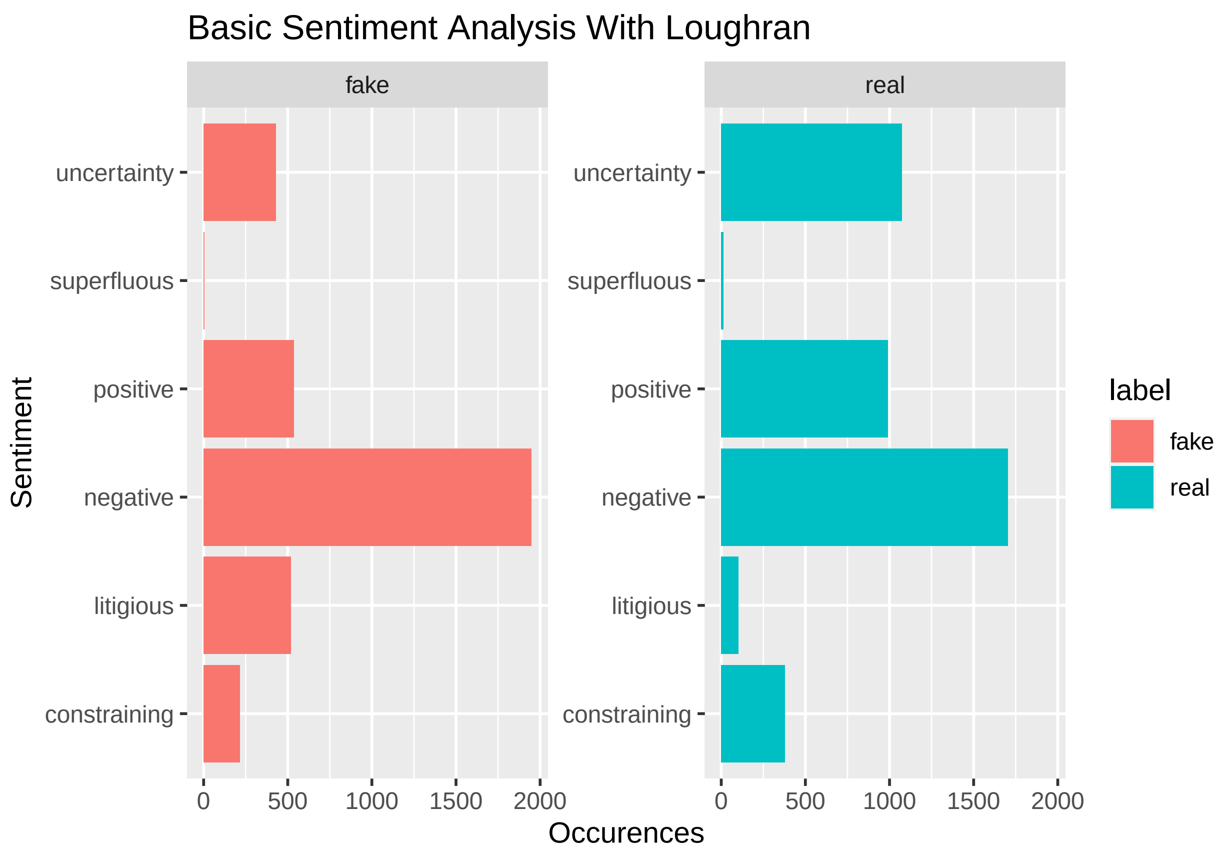
```
glimpse(get_sentiments("loughran"))
```

```
## Rows: 4,150
## Columns: 2
## $ word      <chr> "abandon", "abandoned", "abandoning", "abandonment", "abando~
## $ sentiment <chr> "negative", "negative", "negative", "negative", "negative", ~
```

```
table(get_sentiments("loughran")$sentiment)
```

```
##
## constraining    litigious     negative     positive  superfluous  uncertainty
##          184          904         2355          354           56          297
```

```r
count_df <- unnested_tweets %>% inner_join(get_sentiments("loughran")) %>% count(label, sentiment)
ggplot(count_df, aes(x = sentiment, y = n, fill = label)) + geom_col() + facet_wrap(~ label, scale = "f
```



The loughran corpus seems to show some differences between the types of tweets. Real tweets seem to have more `uncertainty` and are `litigious` while fake tweets are more `constraining`

Using the nrc Corpus

A glimpse of the `nrc` corpus is provide below. It has 6453 words with 10 possible sentiment classifications.

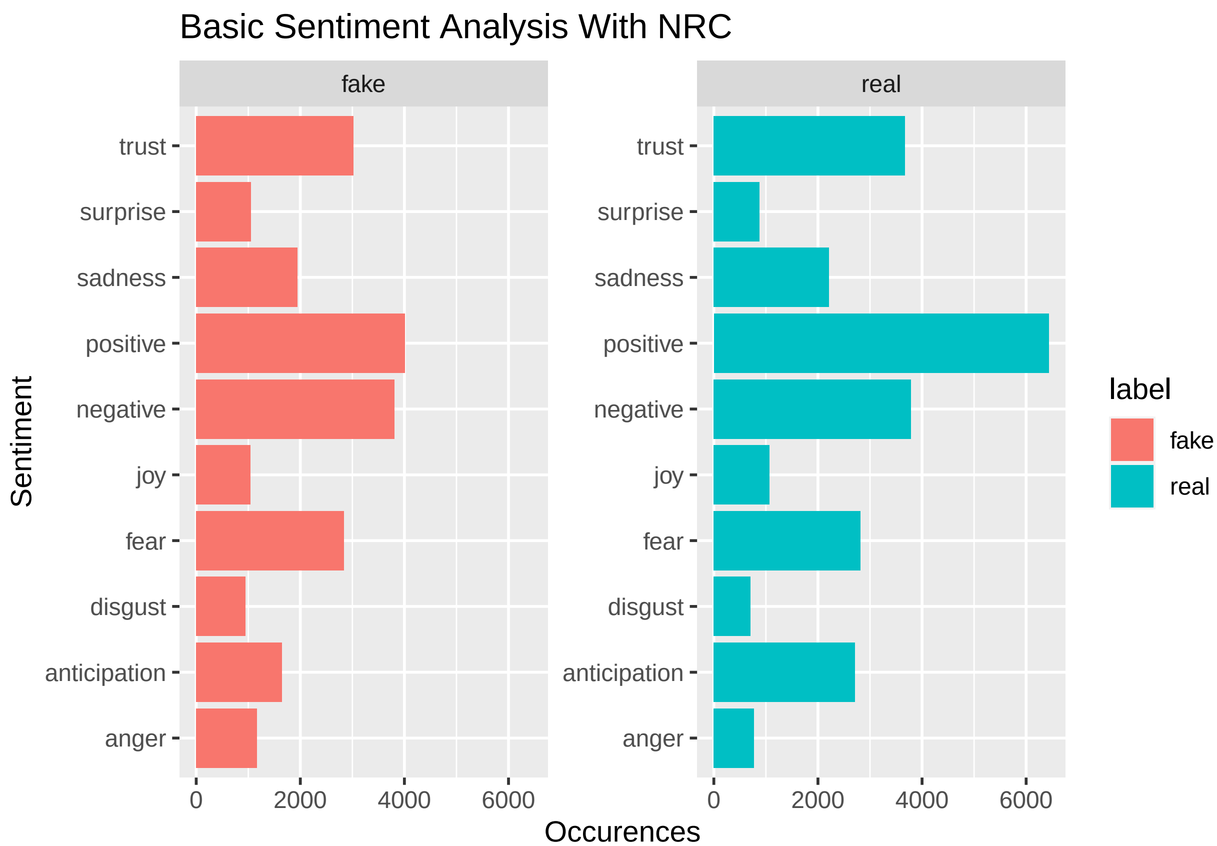
```r
glimpse(get_sentiments("nrc"))
```

```
## Rows: 13,872
## Columns: 2
## $ word      <chr> "abacus", "abandon", "abandon", "abandon", "abandoned", "aba~
## $ sentiment <chr> "trust", "fear", "negative", "sadness", "anger", "fear", "ne~
```

```r
table(get_sentiments("nrc")$sentiment)
```

```
##
##        anger anticipation      disgust         fear          joy     negative
##         1245          837         1056         1474          687         3316
##     positive      sadness     surprise        trust
##         2308         1187          532         1230
```

```
count_df <- unnested_tweets %>% inner_join(get_sentiments("nrc")) %>% count(label, sentiment)
ggplot(count_df, aes(x = sentiment, y = n, fill = label)) + geom_col() + facet_wrap(~ label, scale = "f
```



It is a little more difficult to visually see the differences with `nrc`, but it appears that the pronounced differences between real and fake is positiviy and anticipation.

References

Aghammadzada, E. (2021). *COVID19 Fake News Dataset NLP* . Retrieved from: https://www.kaggle.com/datasets/elvinagammed/covid19-fake-news-dataset-nlp

Baidu Research. (2018). *Baidu's Optimized ERNIE Achieves State-of-the-Art Results in NLP Tasks*. Retrieved from: http://research.baidu.com/Blog/index-view?id=121

diptamath. (2021). *COVID19 Fake News Detection in English*. Retrieved from: https://github.com/diptamath/covid_fake_news

Silge, J. & Robinson, D. (2022). *Sentiment analysis with tidy data* . Retrieved from: https://www.tidytextmining.com/sentiment.html#the-sentiments-datasets