Data preperation and Modeling

CHINDU

```r
library(tidytext)
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
```

```r
tweet <- read.csv('Tweets.csv')
prop.table(table(tweet$airline_sentiment))
```

```
##
##  negative   neutral  positive
## 0.6269126 0.2116803 0.1614071
```

```r
tweet$text <- gsub("^@\\w+ *", "", tweet$text)  # remove @airline
head(tweet)
```

```
##       tweet_id airline_sentiment airline_sentiment_confidence negativereason
## 1 5.703061e+17           neutral                       1.0000
## 2 5.703011e+17          positive                       0.3486
## 3 5.703011e+17           neutral                       0.6837
## 4 5.703010e+17          negative                       1.0000    Bad Flight
## 5 5.703008e+17          negative                       1.0000    Can't Tell
## 6 5.703008e+17          negative                       1.0000    Can't Tell
##   negativereason_confidence        airline airline_sentiment_gold       name
## 1                        NA Virgin America                           cairdin
## 2                    0.0000 Virgin America                          jnardino
## 3                        NA Virgin America                        yvonnalynn
## 4                    0.7033 Virgin America                          jnardino
## 5                    1.0000 Virgin America                          jnardino
## 6                    0.6842 Virgin America                          jnardino
##   negativereason_gold retweet_count
## 1                                 0
## 2                                 0
```

```
## 3                                 0
## 4                                 0
## 5                                 0
## 6                                 0
##
## 1
## 2                                                                  plus you've added commercials to th
## 3                                                                   I didn't today... Must mean I need
## 4           it's really aggressive to blast obnoxious "entertainment" in your guests' faces &amp; the
## 5                                                                                   and it's a really
## 6 seriously would pay $30 a flight for seats that didn't have this playing.\nit's really the only bad
##   tweet_coord             tweet_created tweet_location
## 1             2015-02-24 11:35:52 -0800
## 2             2015-02-24 11:15:59 -0800
## 3             2015-02-24 11:15:48 -0800      Lets Play
## 4             2015-02-24 11:15:36 -0800
## 5             2015-02-24 11:14:45 -0800
## 6             2015-02-24 11:14:33 -0800
##                user_timezone
## 1 Eastern Time (US & Canada)
## 2 Pacific Time (US & Canada)
## 3 Central Time (US & Canada)
## 4 Pacific Time (US & Canada)
## 5 Pacific Time (US & Canada)
## 6 Pacific Time (US & Canada)
```

```
# since unnest_tokens deal with punctuations and lowercase, we just need to worry about the other prepr
library(tm)
```

```
## Loading required package: NLP
```

```
tweet_data <- subset(tweet, airline_sentiment != 'neutral')
tweet_data <- subset(tweet_data, select=c('tweet_id','airline_sentiment', 'text','airline'))
tweet_data$text<-  gsub("\\W|\\d|http\\w?", " ", tweet_data$text, perl = T)
# Change special characters to english letters
library(stringi)
tweet_data$text<-stringi::stri_trans_general(tweet_data$text, "latin-ascii")
```

Lets understand how many tweets are there for each airline

```
tweet_data %>% group_by(airline) %>%
  summarise(Total_tweets=n_distinct(tweet_id))
```

```
## # A tibble: 6 x 2
##   airline        Total_tweets
##   <fct>                 <int>
## 1 American               2180
## 2 Delta                  1499
## 3 Southwest              1756
## 4 United                 3125
## 5 US Airways             2532
## 6 Virgin America          333
```

Unnest token for further analysis

```
tweet_data_token<- tweet_data%>%
  unnest_tokens(word,text)%>%
  anti_join(stop_words)
```

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

Checking the common words

```
library(ggplot2)
```

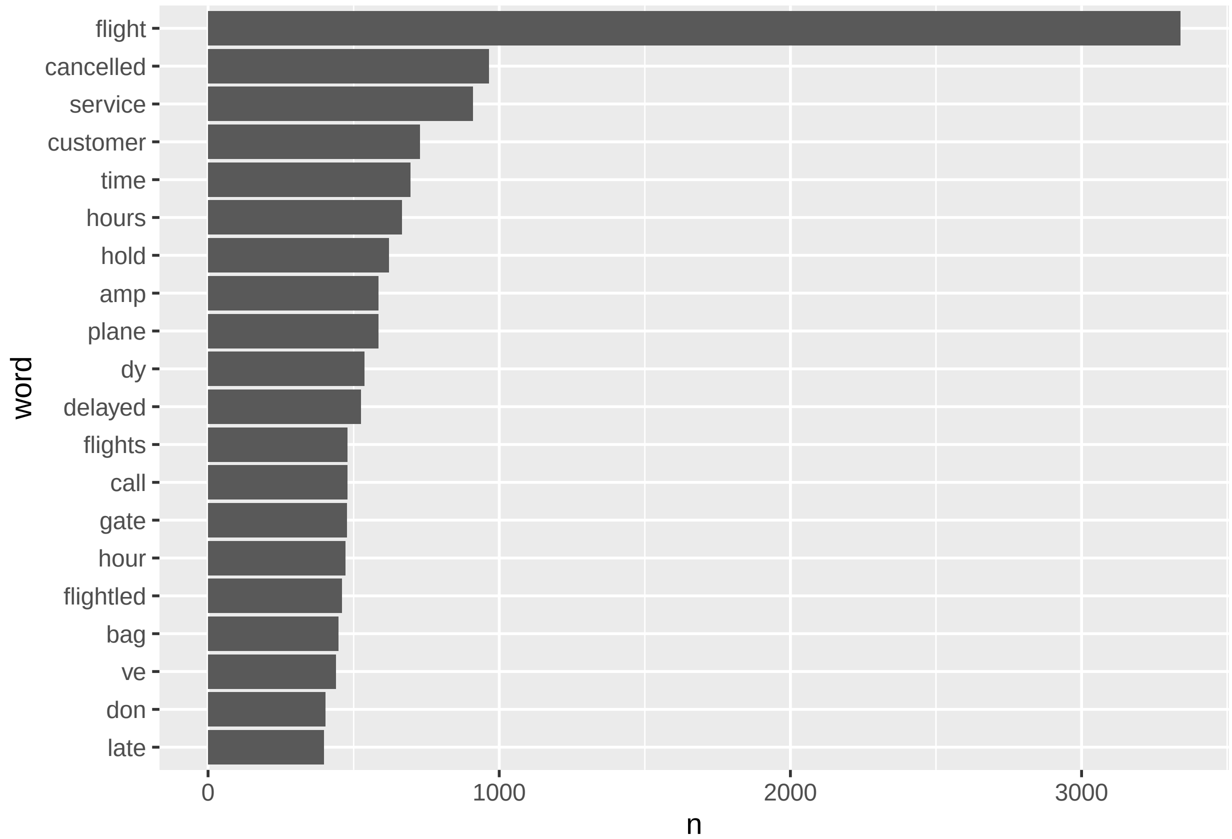
```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

```
word_count<-tweet_data_token %>%
  count(word,sort=TRUE)
word_count
```

```
## # A tibble: 10,974 x 2
##    word          n
##    <chr>     <int>
##  1 flight     3339
##  2 cancelled   964
##  3 service     910
##  4 customer    727
##  5 time        695
##  6 hours       666
##  7 hold        621
##  8 amp         585
##  9 plane       584
## 10 dy          536
## # ... with 10,964 more rows
```

```
word_count %>%
  top_n(20) %>%
  mutate(word = reorder(word, n)) %>%
  # Use aes() to put words on the x-axis and frequency on the y-axis
  ggplot(aes(word, n)) +
  # Make a bar chart with geom_col()
  geom_col() +
  coord_flip()
```

```
## Selecting by n
```



The common words are flight, cancelled,service,customer,time,hours,hold etc.

```
word_totals <- tweet_data_token %>%
  group_by (tweet_id) %>%
  count ()
```

```
new<-tweet_data_token %>%
  inner_join(get_sentiments("bing")) %>%
  group_by (tweet_id)
```

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

```
table(new$airline_sentiment,new$sentiment)
```

```
##
##             negative positive
##   negative      6998     2078
##   neutral          0        0
##   positive       394     1465
```

```
word_counts <- tweet_data_token %>%
  # Implement sentiment analysis using the "bing" lexicon
  inner_join(get_sentiments("bing")) %>%
  # Count by word and sentiment
  count(word, sentiment)
```

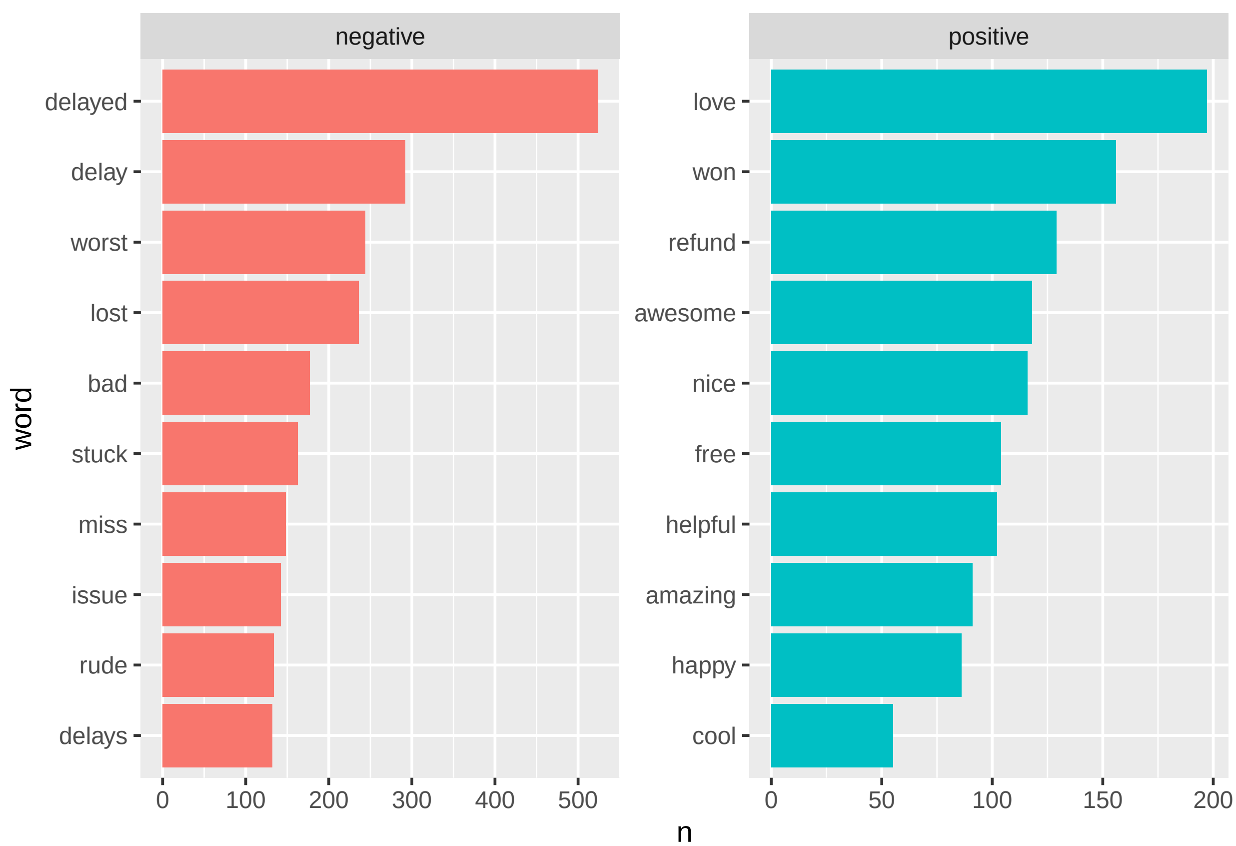
```
## Joining, by = "word"
```

Understanding the words that are influencing the sentiment score

```r
top_words <- word_counts %>%
  # Group by sentiment
  group_by(sentiment) %>%
  # Take the top 10 for each sentiment
  top_n(10) %>%
  ungroup() %>%
  # Make word a factor in order of n
  mutate(word = reorder(word, n))
```

```
## Selecting by n
```

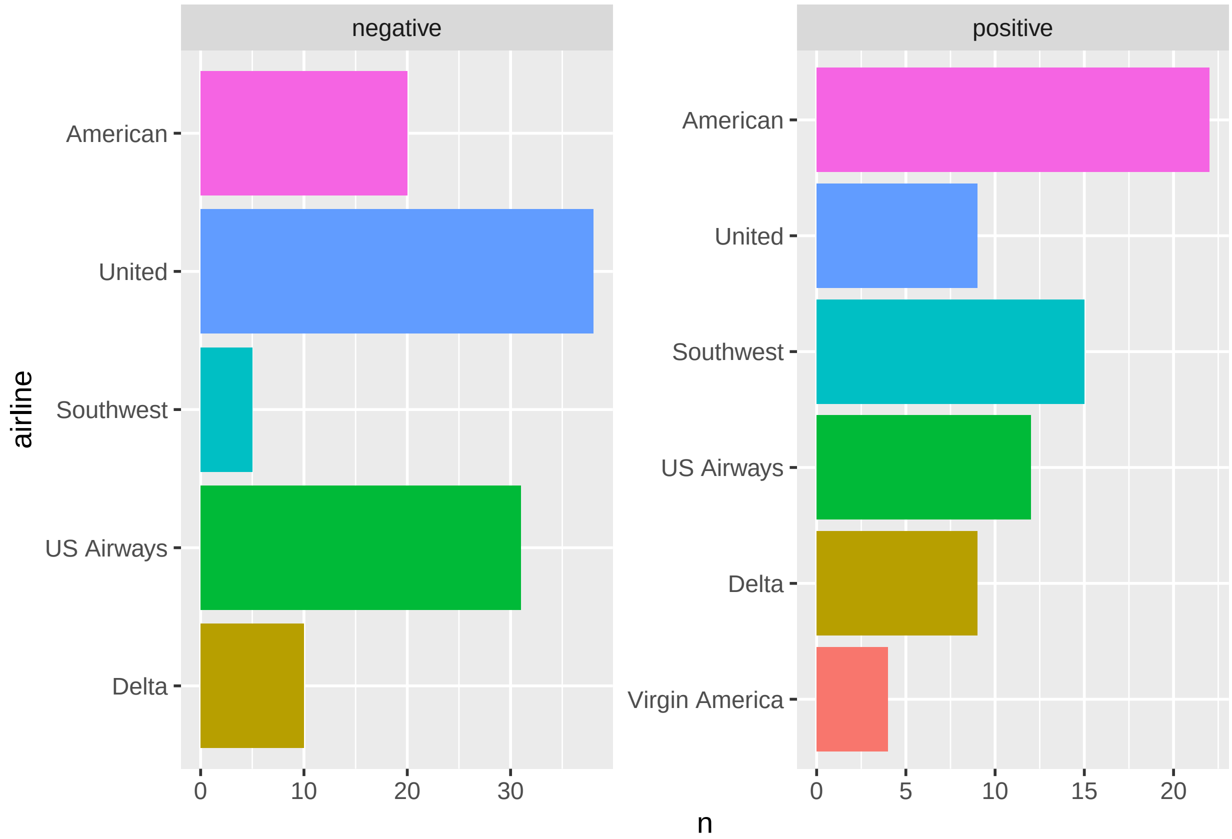
```r
# Use aes() to put words on the x-axis and n on the y-axis
ggplot(top_words, aes(word, n, fill = sentiment)) +
  # Make a bar chart with geom_col()
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free") +
  coord_flip()
```



We should always check which words are contributing to the sentiment scores. Depending on the dataset it may not be what you want.

comparison of positive and negarive reation by airline

```
new %>%
    count(airline, sentiment) %>%
    group_by(sentiment) %>%
    top_n(10, n) %>%
    ungroup() %>%
    mutate(airline = reorder(airline, n)) %>%
    ggplot(aes(airline, n, fill = airline)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~ sentiment, scales = "free") +
    coord_flip()
```

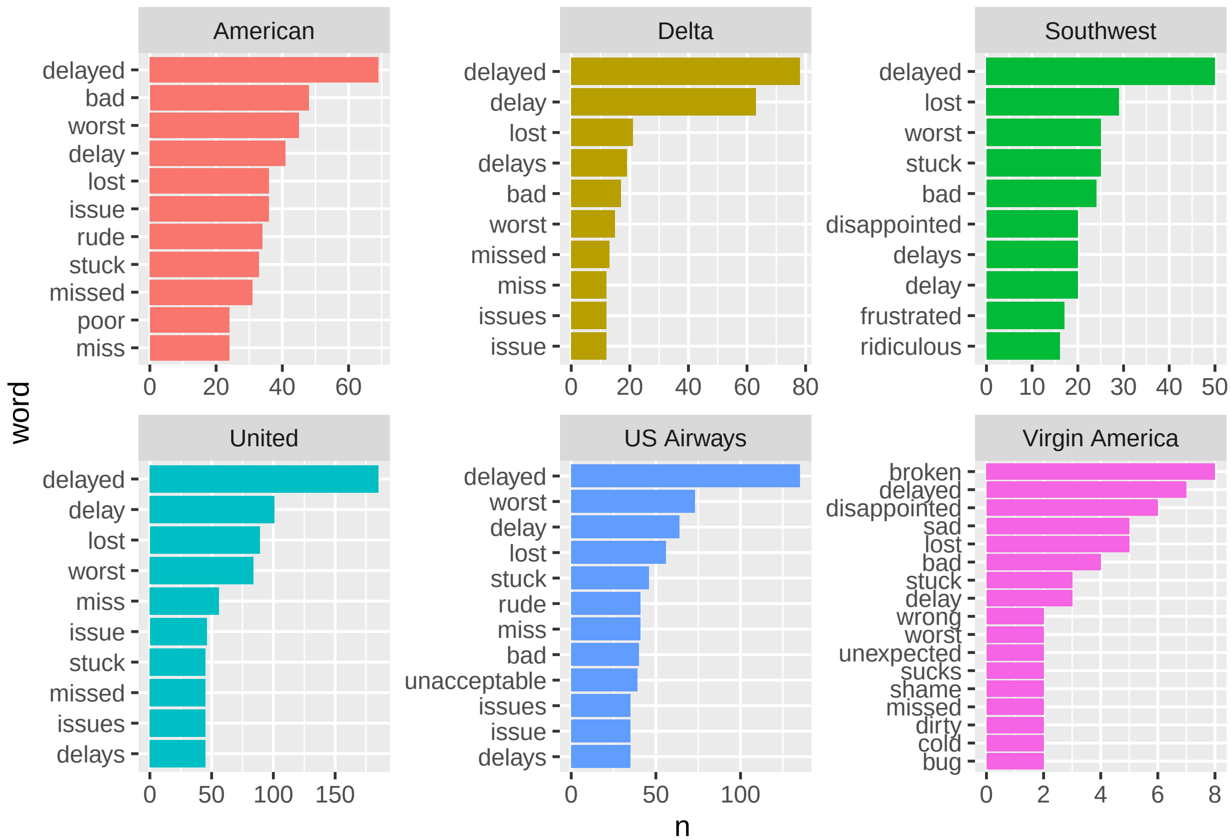



```
new2 <- subset(new, select=c('airline','sentiment','word'))
```

Analyse top negatice words for each airline

```
new2 %>%
    filter(sentiment == "negative") %>%
    count(word, airline) %>%
    group_by(airline) %>%
    top_n(10, n) %>%
    ungroup() %>%
    mutate(word = reorder(paste(word, airline, sep = "__"), n)) %>%
```

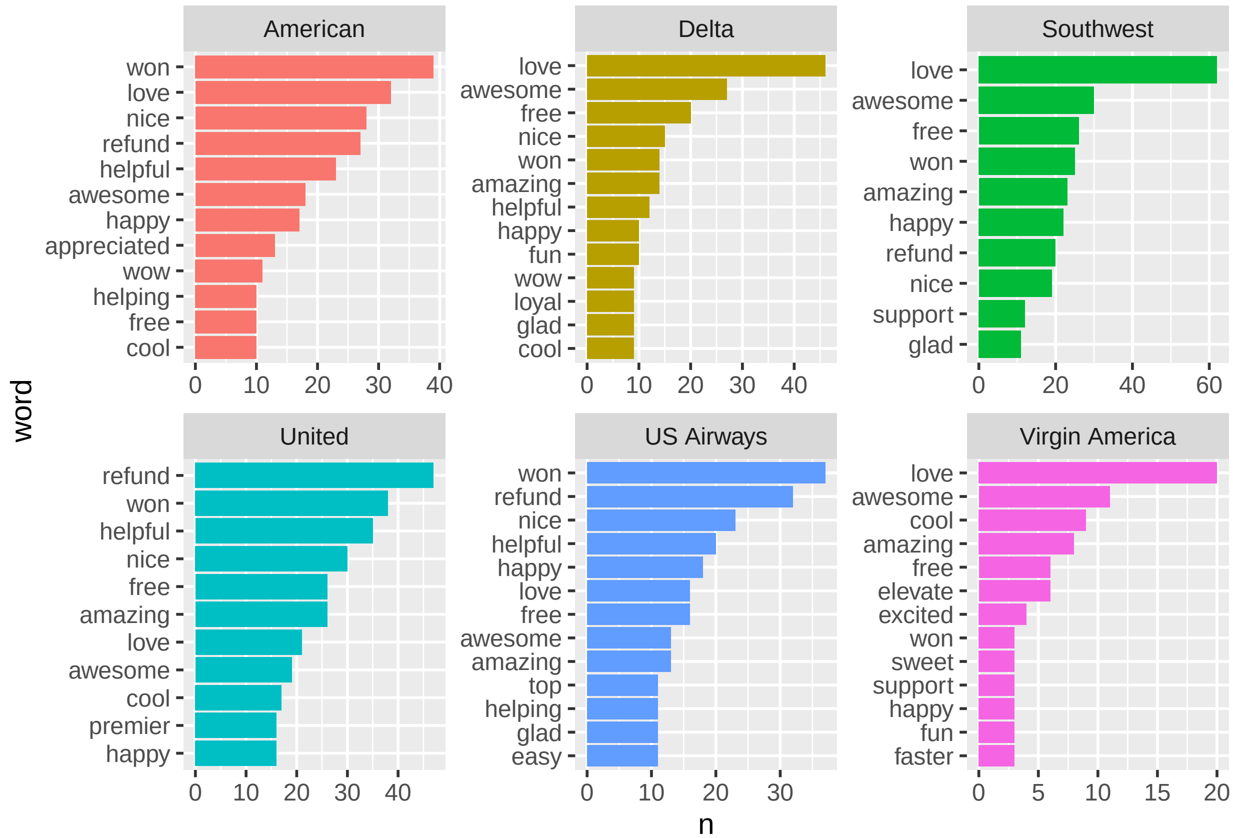
```
    ggplot(aes(word, n, fill = airline)) +
    geom_col(show.legend = FALSE) +
    scale_x_discrete(labels = function(x) gsub("__.+$", "", x)) +
    facet_wrap(~ airline, nrow = 2, scales = "free") +
    coord_flip()
```



here we can see some issue that we need to work on. Words like delayed and delays and delay means the same thing. We should group these together to get more meaningful analysis.

Lets look at the positive words for each airlines

```
new2 %>%
    filter(sentiment == "positive") %>%
    count(word, airline) %>%
    group_by(airline) %>%
    top_n(10, n) %>%
    ungroup() %>%
    mutate(word = reorder(paste(word, airline, sep = "__"), n)) %>%
    ggplot(aes(word, n, fill = airline)) +
    geom_col(show.legend = FALSE) +
    scale_x_discrete(labels = function(x) gsub("__.+$", "", x)) +
    facet_wrap(~ airline, nrow = 2, scales = "free") +
    coord_flip()
```




```
new2 %>%
    filter(sentiment == "negative") %>%
    count(word, airline) %>%
    group_by(airline) %>%
    top_n(10, n)
```

```
## # A tibble: 70 x 3
## # Groups:   airline [6]
##    word   airline            n
##    <chr>  <fct>          <int>
##  1 bad    American          48
##  2 bad    Delta             17
##  3 bad    Southwest         24
##  4 bad    US Airways        40
##  5 bad    Virgin America     4
##  6 broken Virgin America     8
##  7 bug    Virgin America     2
##  8 cold   Virgin America     2
##  9 delay  American          41
## 10 delay  Delta             63
## # ... with 60 more rows
```

```
new<-tweet_data_token %>%
  inner_join(get_sentiments("afinn"))
```

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

```r
new2<-subset(new, select=c('tweet_id','value'))

new2 %>%
  group_by (tweet_id) %>%
  summarise(sentiment=sum(value))
```

```
## # A tibble: 7,147 x 2
##    tweet_id sentiment
##       <dbl>     <dbl>
##  1  5.68e17        -1
##  2  5.68e17         1
##  3  5.68e17         2
##  4  5.68e17         3
##  5  5.68e17         2
##  6  5.68e17         3
##  7  5.68e17        -5
##  8  5.68e17        -5
##  9  5.68e17         3
## 10  5.68e17        -2
## # ... with 7,137 more rows
```

```r
new$sentiment2<-ifelse(new2$value>0,"Positive","Negative")

table(new$airline_sentiment,new$sentiment2)
```

```
##
##            Negative Positive
##   negative     6859     2605
##   neutral         0        0
##   positive      324     1521
```