Text Analytics and Natural Language Processing (NLP)

Hult International Business School

Prof. Thomas Kurnicki

A3 Business Insight Report: Airlines Industry during the Coronavirus Pandemic

Carolina Játiva

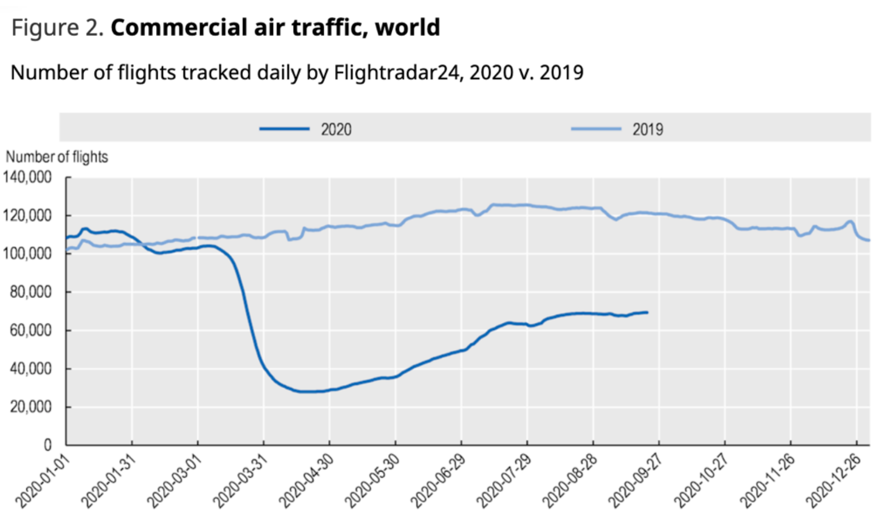
10th February 2020

Wordcount: 1310

Introduction

The coronavirus pandemic has negatively impacted many industries around the world. One of the sectors that was affected strongly in the US is the airline industry. Given the increased fear of contracting the virus and the government regulations around the world that required airports to close to stop the pandemic, there was a "dramatic drop in demand for passenger air transport" (OECD, 2020). Due to the pandemic, airlines today face two uncertainties. The first one is the "cost of health-related measures" such as operating costs for health and safety requirements (e.g., disinfection, PPE, temperature checks, or viral tests) (OECD, 2020). Secondly, "travel restrictions and lockdowns are likely to change transport behavior by cautious consumers" (OECD, 2020). This decrease in the demand in the airline industry is threatening many airlines in the US. Therefore, they need to tackle this situation by taking the appropriate strategies that will allow them to stay profitable and at the same time take care of the safety and health of the passengers.

The following report will analyze recently posted tweets that contain the following hashtags: #jetblue, #american airlines, #united airlines. The objective of looking for these hashtags in the most recent Twitter posts is to see what Twitter users are posting about these three airlines. Also, the analysis will look if there is a trend of the pandemic topic in the posts.



OECD. (2020).

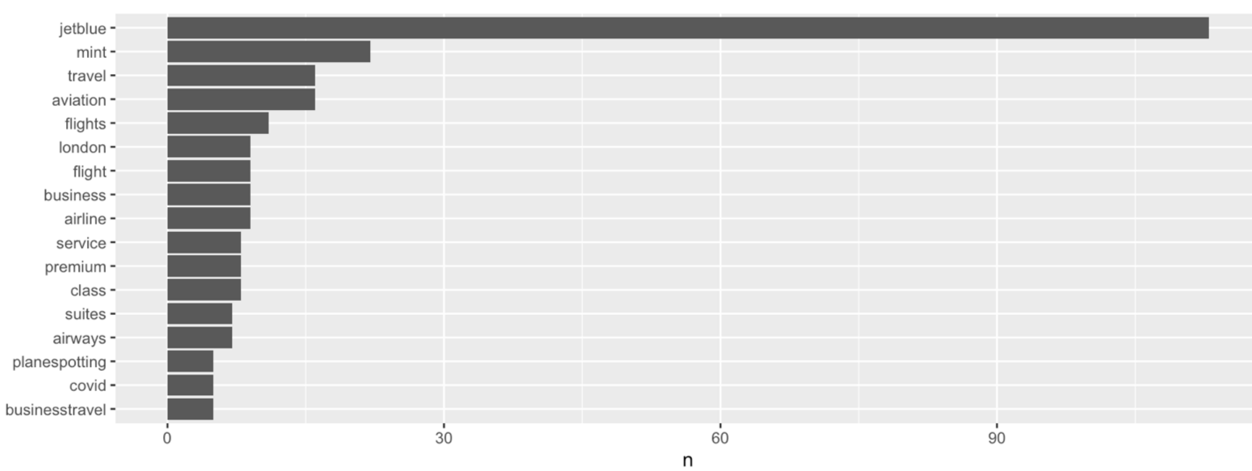
Analysis & Business Insights

Using the library and package "rtweet," it was possible to pull data from Twitter that contains hashtags of the three airlines: JetBlue, American Airlines, and United Airlines. Then, the stop words were removed to create three different data frames in a tidy format for each airline by tokenizing the datasets' words. Therefore, it was possible to get a count of the most repeated words. As it was suspected, we can see that in the top 20 words of the three airlines, we can see terms such as covid. This suggests that it is a trending topic when talking about the airline industry.

	JetBlue		American Airlines		United Airlines	
	word	n	word	n	word	n
1	jetblue	129	americanairlines	291	unitedairlines	122
2	mint	28	american	89	united	53
3	travel	18	airlines	80	airlines	23
4	aviation	17	americanair	75	flight	22
5	business	13	de	47	mask	14
6	flights	13	travel	34	travel	13
7	london	12	flight	33	aviation	12
8	class	11	aviation	31	flying	11
9	flight	11	la	29	covid19	10
10	premium	10	covid	27	fly	10
11	airline	9	13,000	26	avgeek	9
12	service	9	employees	26	boeing	9
13	suites	8	en	20	i'm	9
14	airways	7	avgeek	19	pilot	9
15	de	7	covid19	19	unitedtogether	9
16	businesstravel	6	flights	18	flights	8
17	covid	6	pandemic	18	plane	8
18	airbus	5	miami	17	1976	7
19	airlines	5	airline	16	airbus	7
20	blue	5	furlough	16	airline	7

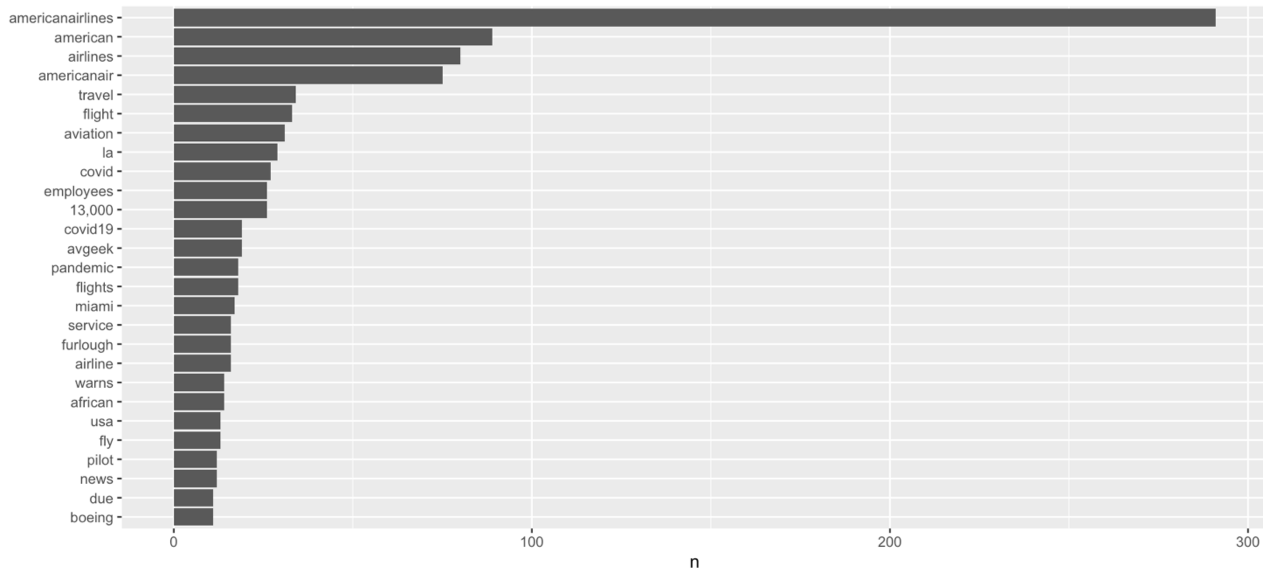
As we can see in the tidy data frame of Jet Blue, covid is one of the most frequent words in the frequency plot. Another interesting insight for Jet Blue is that one of the most frequent words is mint, which is the recently launched cabin by JetBlue "for its new transatlantic services. Every passenger in the Mint cabin will be welcomed with more privacy, more space, and lie flat comfort at every seat" (Bailey, 2021).

Fig. 1 – Frequency histogram of most used words in JetBlue



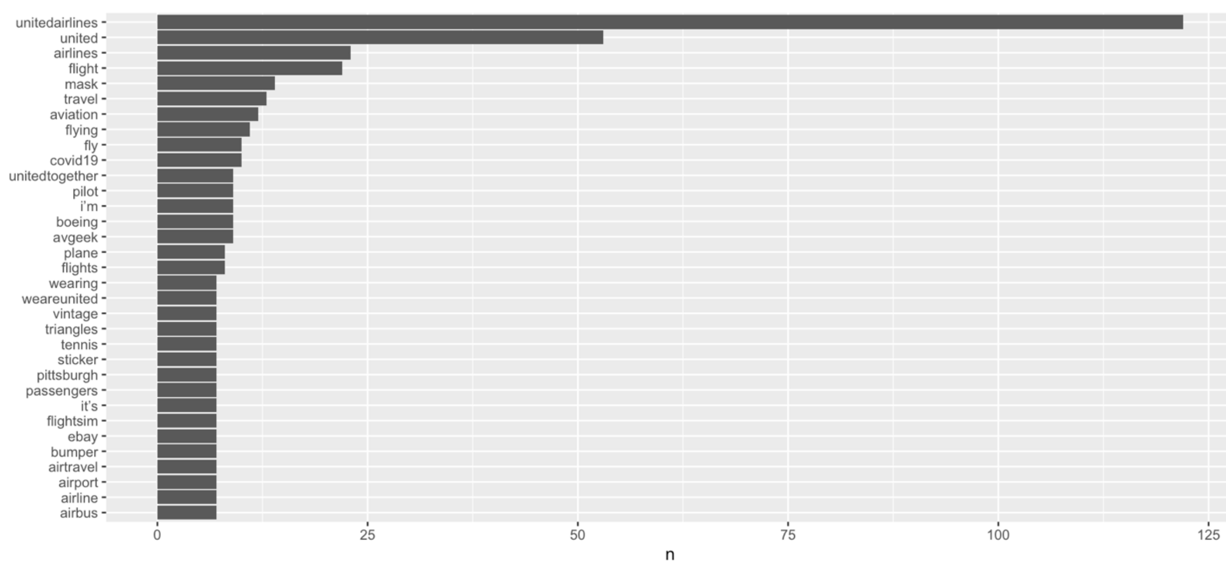
In the tidy data frame of American Airlines, we can see that covid, covid19, and pandemic are the most frequent words. Also, we can see words like 13.000, employees, and furlough given that "American Airlines said it will send furlough notice to about 13,000 employees as a second round of federal payroll aid is set to expire next month, and travel demand remains in tatters" (Joseph, 2021). Moreover, we can see the word Miami, given that "American Airlines is strengthening its commitment to its Miami hub with the announcement of two new international routes to Tel Aviv (TLV) and Paramaribo, Suriname (PBM), beginning this summer" (American Airlines Newsroom, 2021).

Fig. 2 – Frequency histogram of most used words in American Airlines



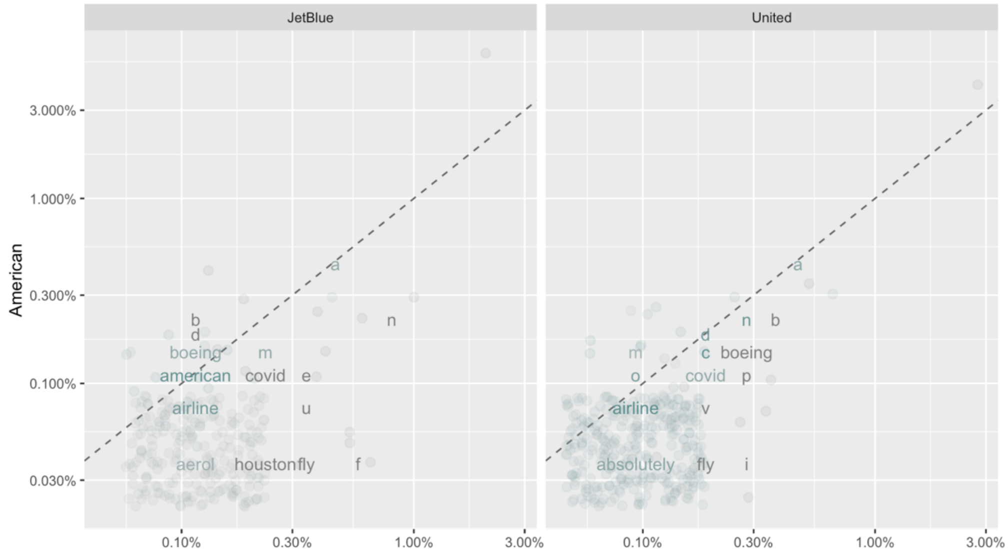
Finally, as we can see in the tidy data frame of United Airlines, we can see as well that mask and covid19 are the most frequent words. Once again, we see that the pandemic topic is currently trendy to discuss in the airline industry.

Fig. 3 – Frequency histogram of most used words in United Airlines



Next, a correlogram was created, taking as benchmark American Airlines given that the airline has the most significant market share in the US (Statista, 2020). We can see that in both comparisons, the word covid is close to the diagonal line, which means that this word has a similar frequency in the three data sets.

Fig.4 – Correlogram between American Airlines (axis Y) and JetBlue and United Airlines (axis X)



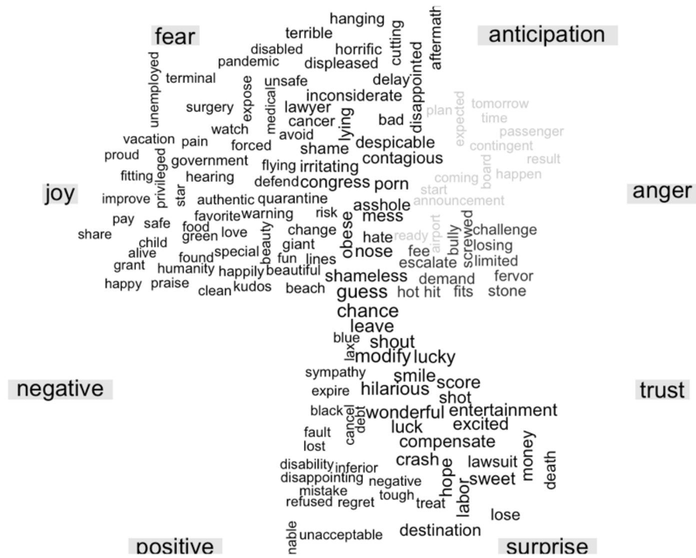
The correlation test for the most frequent words for the three airlines, shows that American Airlines and JetBlue have a correlation of 0.92. At the same time, American Airlines and United Airlines have a correlation of 0.97. We obtain high correlations given that there is a trend in Twitter about talking of covid, as has been discussed before.

In the following word cloud for JetBlue, we can see a cluster of words more towards the positive and joy sentiment even though the coronavirus pandemic is an important event occurring in today's world. It seems that the launch of JetBlue of the Mint cabin is creating a positive reaction in public. Therefore, we can say that JetBlue's strategy has successfully attracted customers in the middle of the pandemic, as we can see words such as happy, luxury, feature, traveling in the joy, and positive sentiments.

Fig.5 – Word cloud NRC Sentiment for JetBlue

In contrast, when we look at the word cloud for American Airlines, we can see a cluster of words more towards fear, anticipation, anger and surprise sentiments. We can say that this is mostly due to the coronavirus pandemic which is an important event in today's world. In addition, the laid off employees of American Airlines could be a reason why we see more a tendency towards negative sentiments.

Fig6 – Word cloud NRC Sentiment for American Airlines



Finally, in the word cloud of the NRC Sentiment for United, we can see that the sentiments that have a higher frequency are positive, joy, anticipation, surprise, and fear. In United's case, it seems as the coronavirus topic is still heavily mentioned on Twitter as we can see quarantine and pandemic near the sentiment of fear. However, other positive sentiments have high frequencies, such as positive and joy. This suggests that United Airlines could have good customer satisfaction. According to Statista, in 2020, United Airlines was in the top 5 airlines index score of customer satisfaction in the United States (Statista, 2020).

Fig7 – Word cloud NRC Sentiment for United

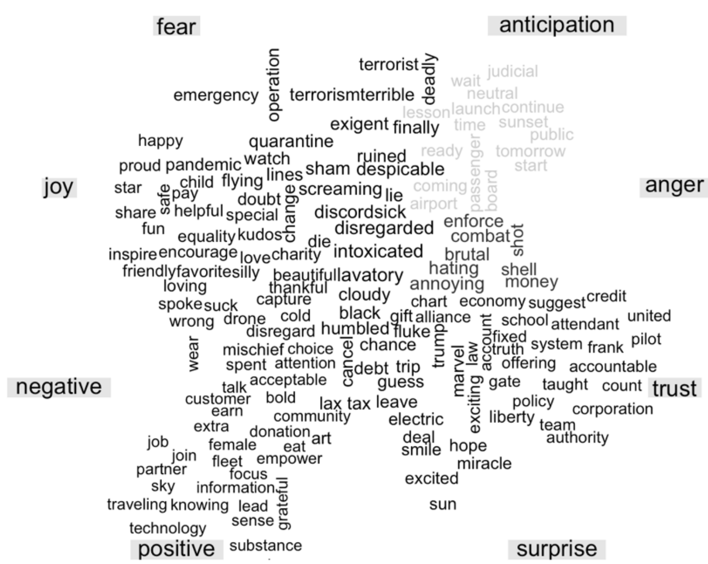
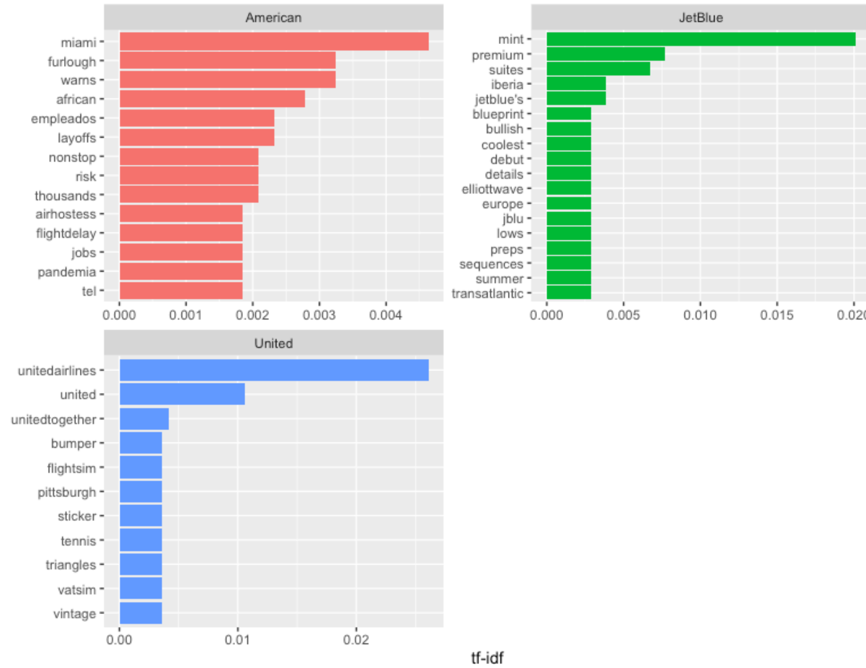


Fig8 – Frequency histograms highest tf-idf words in JetBlue, American Airlines, and United Airlines



Looking at the graphical output of the tf-idf, we can see some critical words for the airlines that can give us some business value. In American Airlines' case we see that one of the most frequent words are Miami and furlough. Despite the bad news mentioned before of the employees laid off, there was also some good news such as the new destinations routes from Miami to Tel Aviv and Paramaribo. For Jet Blue, we see once again that the words with a high frequency are mint and premium word, which come from the launch of the Mint Premium Cabin of Jet Blue. When looking at United Airlines, we can't see any word that would bring some business value.

Recommendations & Conclusion

As we could see, the coronavirus is a trendy topic for the airline industry. It is clear that it affects the demand for flight tickets and is causing the laid-offs of many employees, such as the case of American and United Airlines. It will be key for the airline's success during the pandemic to "waive change and cancellation fees, inform travelers of specific cleaning/sanitization actions" (J.D. Power, 2020).

Other strategies that can be applied to improve the situation, are what JetBlue and American Airlines have done by launching premium cabins and new destinations routes. These strategies could be an excellent example to follow for United Airlines. Given that Twitter is a viral social network and has many active users, what is being said in the tweets will significantly impact the reputation and how customers see these brands. Therefore, the more positive news of these airlines will be better the effect of the pandemic's adverse effects and eventually this could lead to an increase in the demand of tickets.

References

American Airlines. (2021*). American Airlines Aligns Existing Mask Requirements with US Government Mandate.* Retrieved 9th February 2020 from: http://news.aa.com/news/news-details/2021/American-Airlines-Aligns-Existing-Mask-Requirements-with-US-Government-Mandate-OPS-DIS-02/default.aspx

American Airlines Newsroom. (2021). *American Airlines Becomes the Only US Carrier with Nonstop Service from Miami to Tel Aviv and Paramaribo, Suriname.* Retrieved 9th February 2020 from: http://news.aa.com/news/news-details/2021/American-Airlines-Becomes-the-Only-US-Carrier-with-Nonstop-Service-from-Miami-to-Tel-Aviv-and-Paramaribo-Suriname-NET-RTS-02/

Bailey, J. (2021). *Game changing JetBlue Reveals Stunning Transatlantic Mint Suites.* Retrieved 9th February 2020 from: https://simpleflying.com/jetblue-mint-suite/

BBC. (2021). *Covid map: Coronavirus cases, deaths, vaccinations by country.* Retrieved 10th February 2020 from: https://www.bbc.com/news/world-51235105

J.D.Power. (2020). *Importance of Trust, Transparency to Airline Satisfaction Grows as Industry Confronts Pandemic Fears, J.D. Power Finds.* Retrieved 10th February 2020 from: https://www.jdpower.com/business/press-releases/2020-north-america-airline-satisfaction-study

Joseph, L. (2021). *American warns 13,000 employees of furloughs as airlines prepare to lose federal aid next month.* . Retrieved 10th February 2020 from: https://www.cnbc.com/2021/02/03/american-airlines-employees-furlough-notices-covid-travel-stays-low.html

OECD. (2020). *COVID-19 and the aviation industry: Impact and policy responses.* Retrieved 9th February 2020 from: http://www.oecd.org/coronavirus/policy-responses/covid-19-and-the-aviation-industry-impact-and-policy-responses-26d521c1/

Statista. (2020). *Domestic market share of leading US airlines from November 2019 to October 2020.* Retrieved 9th February 2020 from: https://www.statista.com/statistics/250577/domestic-market-share-of-leading-us-airlines/

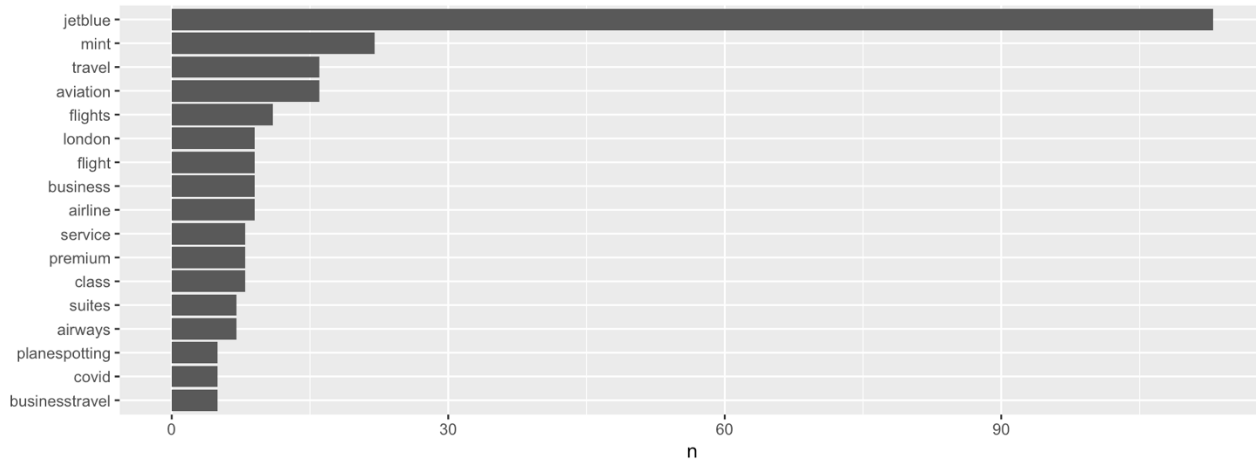
Statista. (2020). *American customer satisfaction index scores for airlines in the United States from 1995 to 2020.* Retrieved 10th February 2020 from: https://www.statista.com/statistics/194941/customer-satisfaction-with-us-airlines-since-1995/

Thiruvengadam, Meena. (2021). *Woman Who Refused to Comply With Airline Mask Rules Arrested After Landing at Washington, DC Airport.* Retrieved 9th February 2020 from: https://www.travelandleisure.com/airlines-airports/american-airlines/american-airlines-passenger-refuses-mask-vaccine

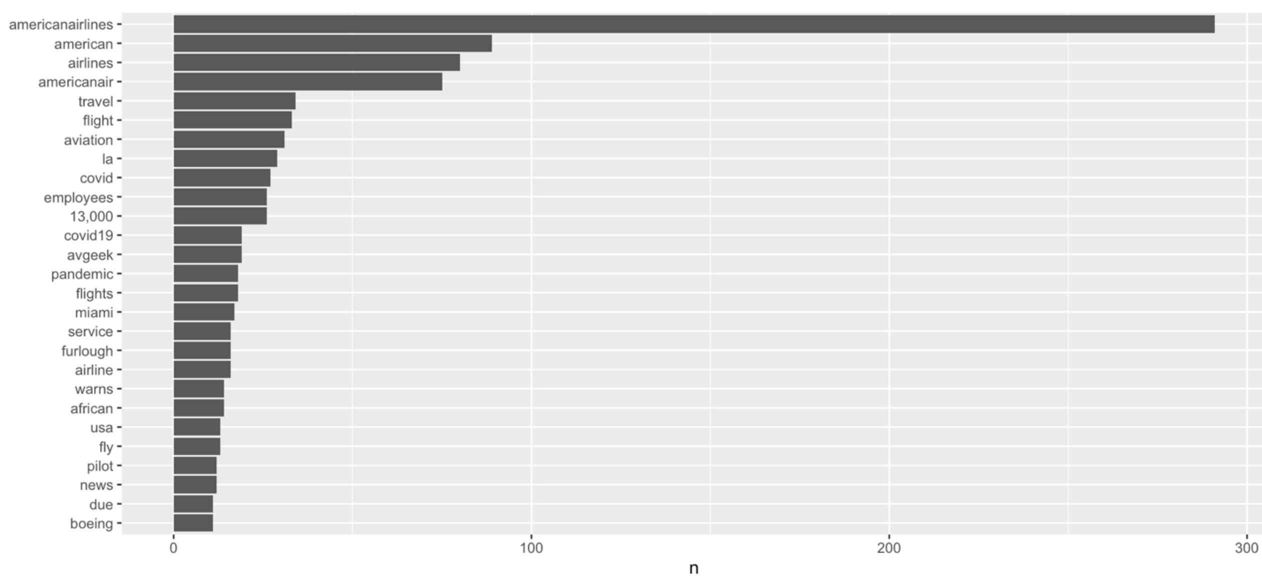
Appendix: R Code & Outputs

```r
#Loading libraries
library(tidyverse)
library(tidytext)
library(textdata)
library(dplyr)
library(widyr)
library(tidyr)
library(stringr)
library(scales)
library(twitteR)
library(rtweet)
library(tm)
library(ggplot2)
library(igraph)
library(ggraph)
library(reshape2)
library(wordcloud)


##########################################
#####Downloading data from twitter#######
##########################################
jetblue<- search_tweets(
  "#jetblue", n = 18000, include_rts = FALSE
)

american<- search_tweets(
  "#americanairlines ", n = 18000, include_rts = FALSE
)

united<- search_tweets(
  "#unitedairlines ", n = 18000, include_rts = FALSE
)

#calling the stop words
data(stop_words)

#creating my own stop_words
custom_stop_words <- tribble(
  ~word, ~lexicon,
  "http", "CUSTOM",
  "https", "CUSTOM",
  "rt", "CUSTOM",
  "t.co", "CUSTOM",
  "amp", "CUSTOM",
  "1", "CUSTOM",
  "2", "CUSTOM",
  "3", "CUSTOM",
  "19", "CUSTOM",
  "15", "CUSTOM",
  "en", "CUSTOM",
  "de", "CUSTOM",
  "i'm", "CUSTOM",
  "it's", "CUSTOM",
  "bfim44pcr8", "CUSTOM",
  "1976", "CUSTOM",
```

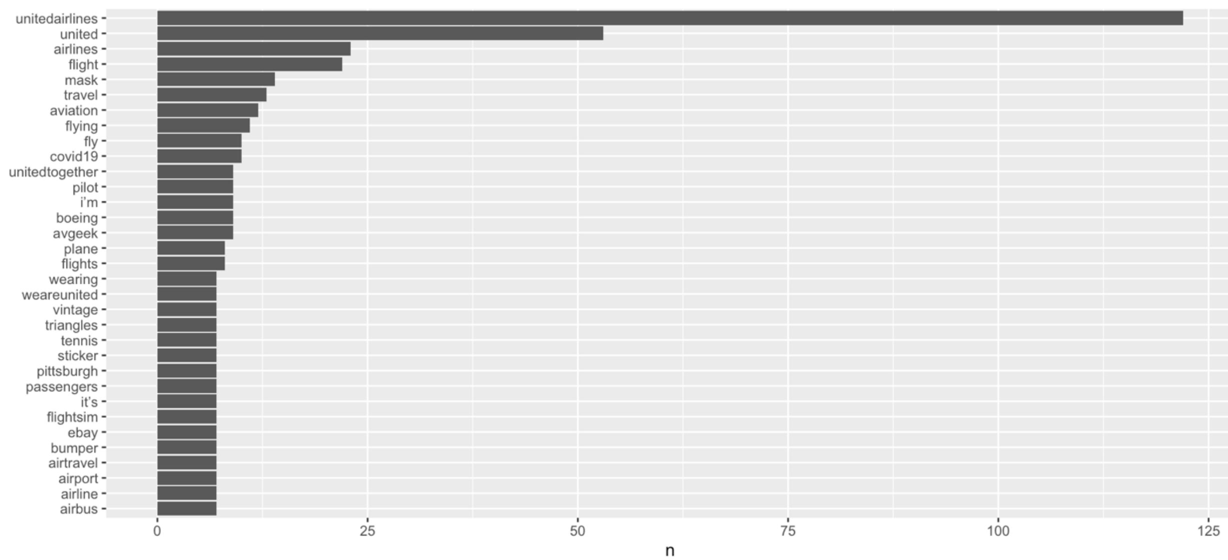
```
  "aa", "CUSTOM",
  "c31rrxg8ix", "CUSTOM",
  "13.000", "CUSTOM",
  "aal", "CUSTOM",
  "del", "CUSTOM",
  "a321nx", "CUSTOM",
  "n2105j", "CUSTOM",
  "vk8rbogqqb", "CUSTOM",
  "ur", "CUSTOM",
)

#joining the custom stop words to the stop words
stop_words2 <- stop_words  %>%
  bind_rows(custom_stop_words)

##########################################
################Tokenization##############
##########################################

tidy_jetblue <- jetblue %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words2) %>%
  count(word, sort = T)

tidy_american <- american %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words2) %>%
  count(word, sort = T)

tidy_united <- united %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words2) %>%
  count(word, sort = T)

############################################
#########Token frequency histograms#########
############################################

freq_jetblue <-tidy_jetblue %>%
  filter(n > 4) %>%
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_jetblue)
```




```
freq_american <-tidy_american %>%
  filter(n > 10) %>%
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_american)
```

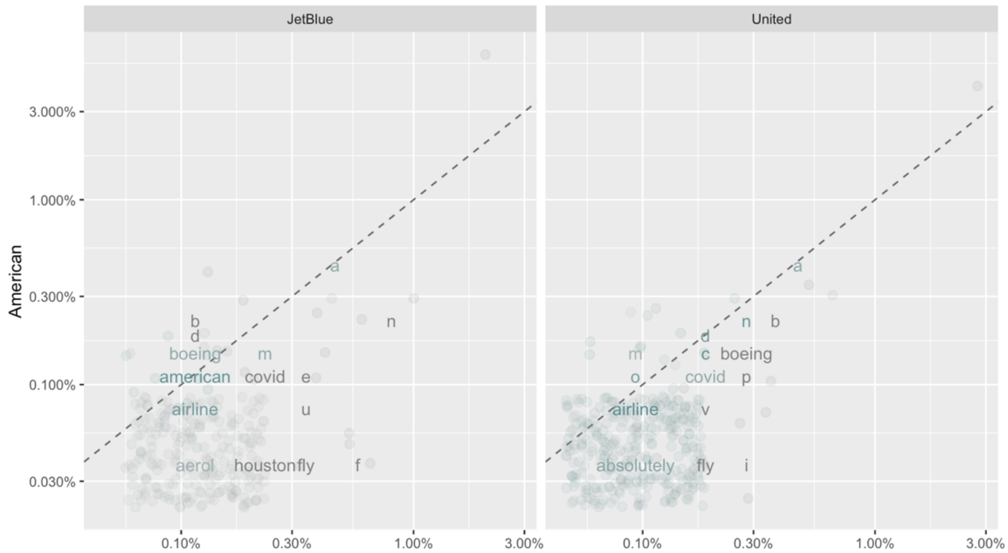



```
freq_united <-tidy_united %>%
  filter(n > 6) %>%
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_united)
```




```
#Correlation between the different airlines
frequency_airlines <- bind_rows(mutate(tidy_jetblue, author = "JetBlue"),
                                mutate(tidy_american, author = "American"),
                                mutate(tidy_united, author = "United")
) %>% #closing bind rows

  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n))%>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion, `JetBlue`, `United`)

print(frequency_airlines)

# A tibble: 6,634 x 4
   word          American author  proportion
   <chr>            <dbl> <chr>        <dbl>
 1 a             0.00514  JetBlue    0.00280
 2 aa            0.000429 JetBlue   NA
 3 aaae          0.000429 JetBlue   NA
 4 aaaedelivers  0.000429 JetBlue   NA
 5 aabird        0.000429 JetBlue   NA
 6 aadvantage    0.000429 JetBlue   NA
 7 aairlinesfail 0.000429 JetBlue   NA
 8 aasafetyvideo 0.000429 JetBlue   NA
 9 aastews       0.000429 JetBlue   NA
10 aateam        0.000429 JetBlue   NA
# … with 6,624 more rows

##########################################
############Correlogram################
##########################################
ggplot(frequency_airlines, aes(x=proportion, y= `American`,
                               color = abs(`American`- proportion)))+
  geom_abline(color="grey40", lty=2)+
  geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
  geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
```

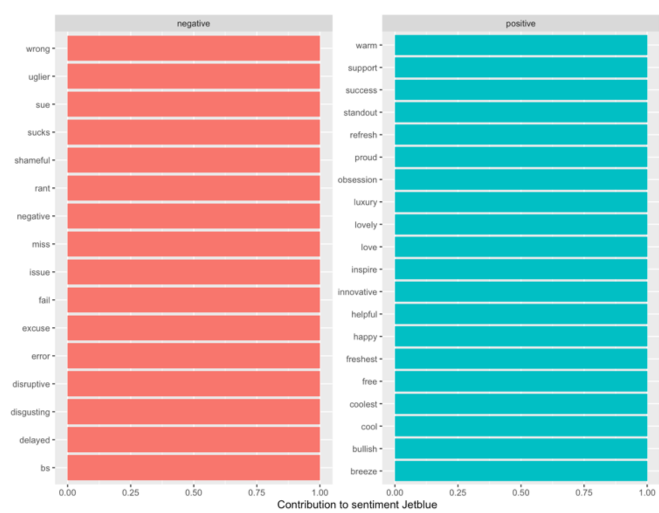
```
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels= percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
  facet_wrap(~author, ncol=2)+
  theme(legend.position = "none")+
  labs(y= "American", x=NULL)
```




```
##########################################
#######doing the cor.test() #############
##########################################

cor.test(data=frequency_airlines[frequency_airlines$author == "JetBlue",],
         ~proportion + `American`)


cor.test(data=frequency_airlines[frequency_airlines$author == "United",],
         ~proportion + `American`)

> cor.test(data=frequency_airlines[frequency_airlines$author == "JetBlue",],
+          ~proportion + `American`)

	Pearson's product-moment correlation

data:  proportion and American
t = 37.199, df = 227, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9060949 0.9431708
sample estimates:
      cor
0.9268604

>
>
> cor.test(data=frequency_airlines[frequency_airlines$author == "United",],
+          ~proportion + `American`)

	Pearson's product-moment correlation
```

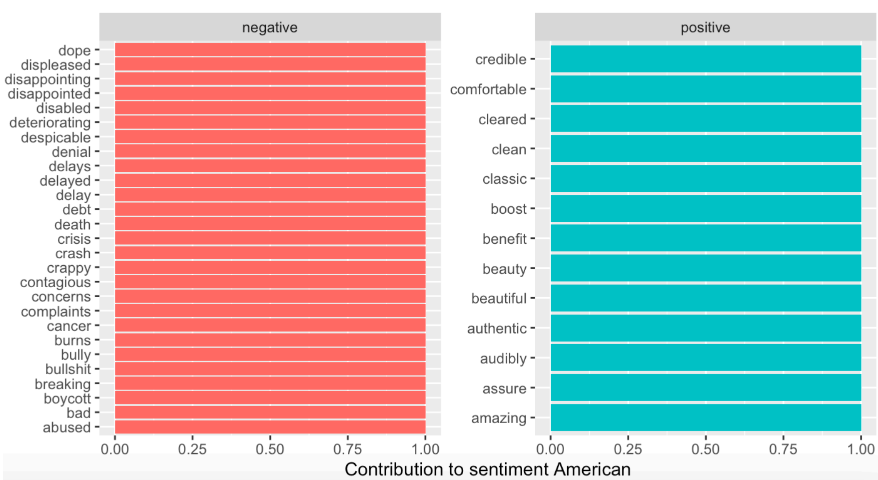
```
data:  proportion and American
t = 80.94, df = 396, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9648873 0.9761964
sample estimates:
      cor
0.9710816
```

```
##########################################
#######Bing sentiment analysis###########
##########################################

###############JetBlue##################
bing_tidy_jetblue <- tidy_jetblue %>%
  inner_join(get_sentiments("bing"))  %>%
  count (word, sentiment, sort = T)  %>%
  arrange(desc(n))

top_bing_tidy_jetblue <- bing_tidy_jetblue[1:80,]

top_bing_tidy_jetblue %>%
  group_by(sentiment)%>%
  top_n(20,n)%>%
  ungroup%>%
  mutate(word=reorder(word,n))%>%
  ggplot(aes(x=word, y=n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales="free_y")+
  labs(y = "Contribution to sentiment Jetblue",
       x= NULL)+
  coord_flip()
```



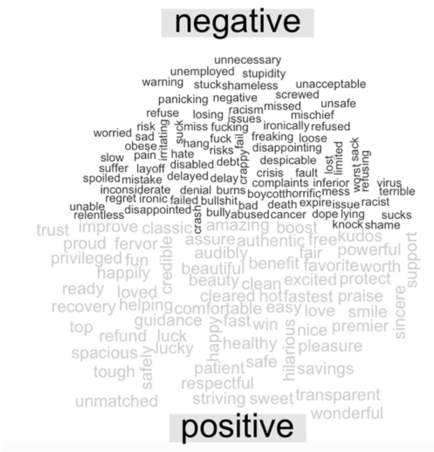

```
cloud_jetblue_bing <- tidy_jetblue %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=TRUE) %>% #token per sentiment
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey80"),
                  max.words=500, scale=c(1, 0.8),
                  title.size=1.5)
```


```
###########American Airlines#############
bing_tidy_american <- tidy_american  %>%
  inner_join(get_sentiments("bing"))  %>%
  count (word, sentiment, sort = T)  %>%
  ungroup()

top_bing_tidy_american <- bing_tidy_american[1:40,]

top_bing_tidy_american %>%
  group_by(sentiment)%>%
  top_n(20,n)%>%
  ungroup%>%
  mutate(word=reorder(word,n))%>%
  ggplot(aes(x=word, y=n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales="free_y")+
  labs(y = "Contribution to sentiment American",
       x= NULL)+
  coord_flip()
```

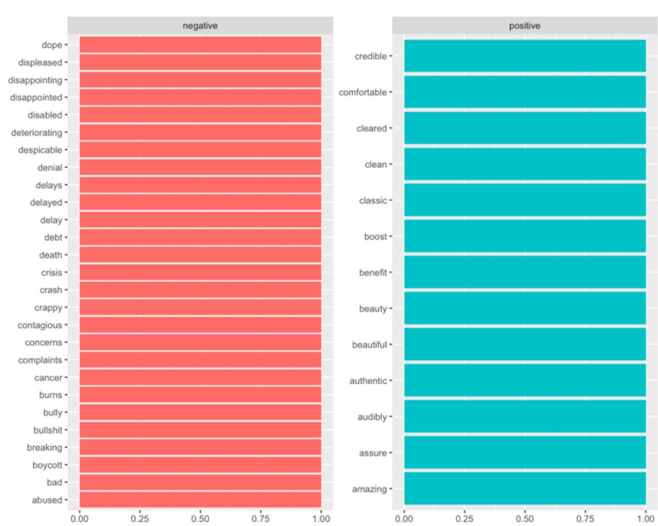



```
cloud_american_bing <- tidy_american %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=TRUE) %>% #token per sentiment
  acast(word ~sentiment, value.var="n", fill=0) %>%
```

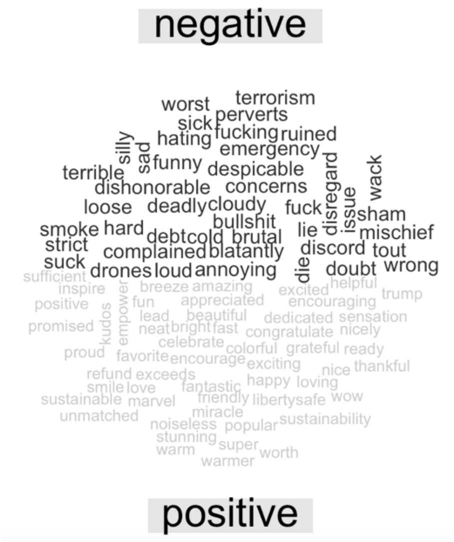
```
comparison.cloud(colors = c("grey20", "grey80"),
                 max.words=150, scale=c(1, 0.1),
                 title.size=2)
```



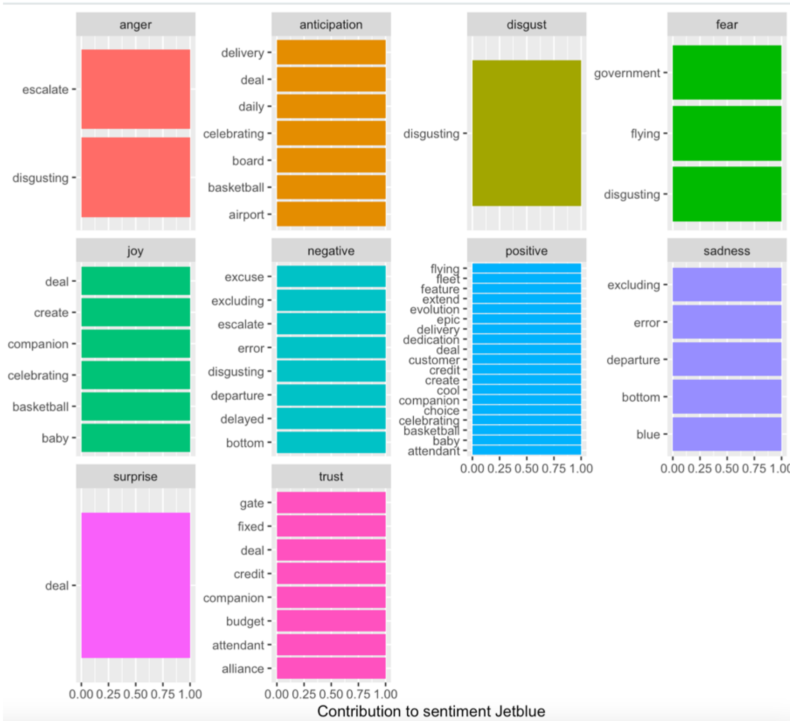

```
###########United Airlines#############
bing_tidy_united <- tidy_united  %>%
  inner_join(get_sentiments("bing"))  %>%
  count (word, sentiment, sort = T)  %>%
  ungroup()

top_bing_tidy_united <- bing_tidy_american[1:40,]

top_bing_tidy_united %>%
  group_by(sentiment)%>%
  top_n(20,n)%>%
  ungroup%>%
  mutate(word=reorder(word,n))%>%
  ggplot(aes(x=word, y=n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales="free_y")+
  labs(y = "Contribution to sentiment United",
       x= NULL)+
  coord_flip()
```



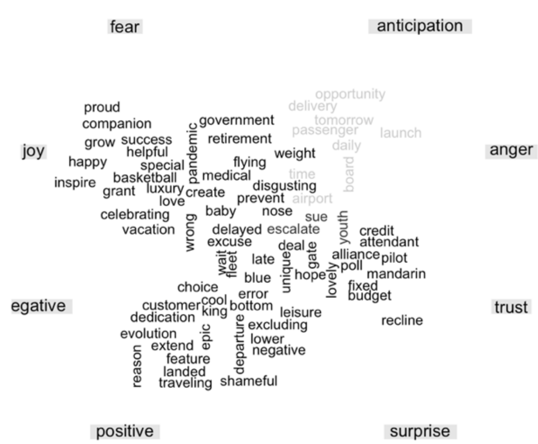
```
cloud_united_bing <- tidy_united %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=TRUE) %>% #token per sentiment
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey80"),
                   max.words=150, scale=c(1, 0.1),
                   title.size=2)
```




```
##########################################
#######NRC sentiment analysis############

###############JetBlue##################
nrc_tidy_jetblue <- tidy_jetblue %>%
  inner_join(get_sentiments("nrc"))  %>%
  count (word, sentiment, sort = T)  %>%
  arrange(desc(n))

top_nrc_tidy_jetblue <- nrc_tidy_jetblue[1:60,]

top_nrc_tidy_jetblue %>%
  group_by(sentiment)%>%
  top_n(15,n)%>%
  ungroup%>%
  mutate(word=reorder(word,n))%>%
  ggplot(aes(x=word, y=n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales="free_y")+
  labs(y = "Contribution to sentiment Jetblue",
       x= NULL)+
  coord_flip()
```

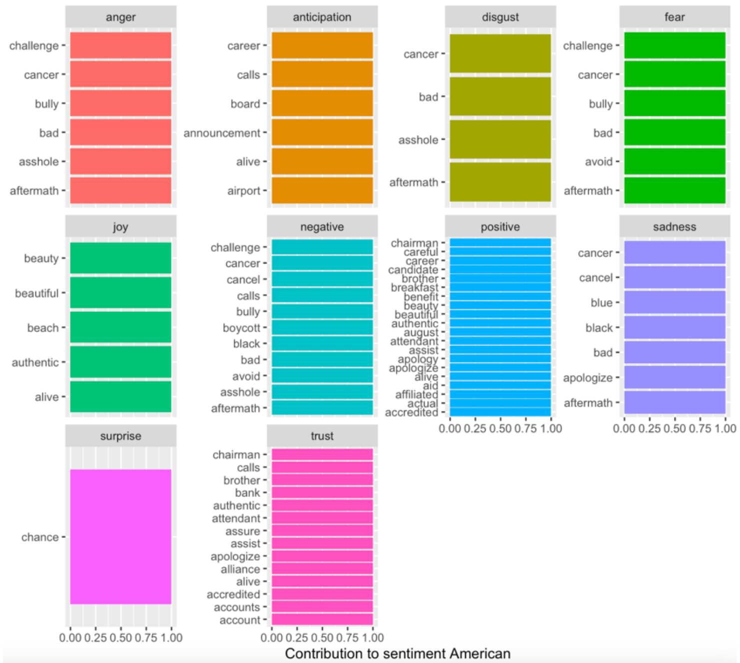



```
cloud_jetblue_nrc <- tidy_jetblue %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort=TRUE) %>% #token per sentiment
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey80"),
                   max.words=500, scale=c(1, 1),
                   title.size=1.2)
```

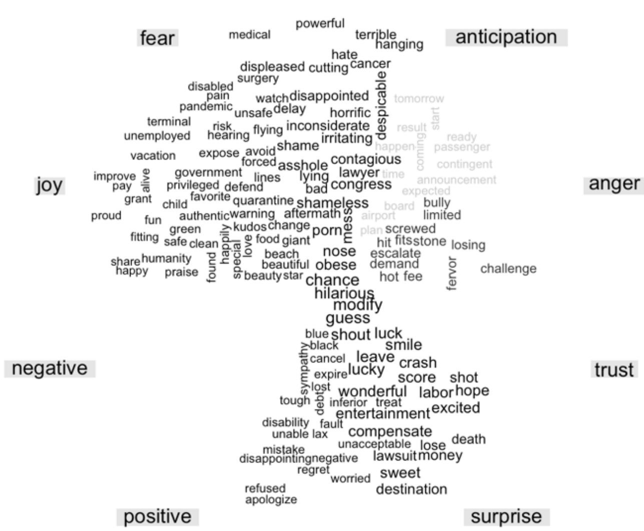



```
###########American Airlines#############
nrc_tidy_american <- tidy_american  %>%
  inner_join(get_sentiments("nrc"))  %>%
  count (word, sentiment, sort = T)  %>%
  ungroup()

top_nrc_tidy_american <- nrc_tidy_american[1:80,]

top_nrc_tidy_american %>%
  group_by(sentiment)%>%
```

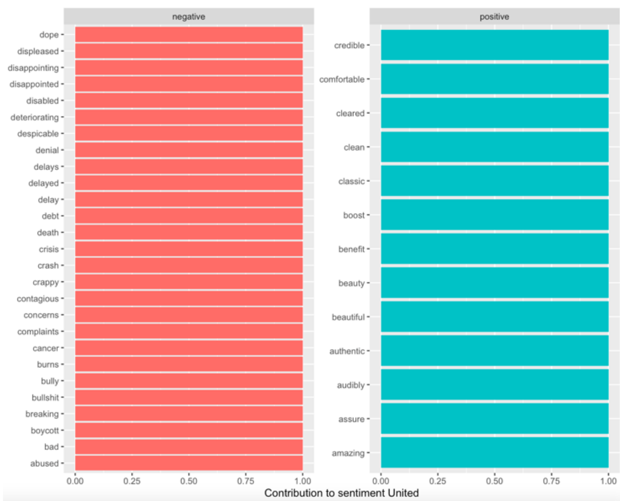
```
  top_n(30,n)%>%
  ungroup%>%
  mutate(word=reorder(word,n))%>%
  ggplot(aes(x=word, y=n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales="free_y")+
  labs(y = "Contribution to sentiment American",
       x= NULL)+
  coord_flip()
```



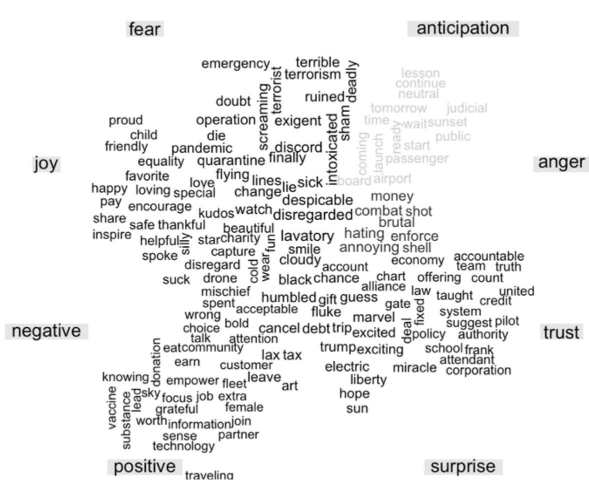

```
cloud_american_nrc <- tidy_american %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort=TRUE) %>% #token per sentiment
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey80"),
                   max.words=150, scale=c(1, 0.5),
                   title.size=1.2)
```



```
###########United Airlines#############
nrc_tidy_united <- tidy_united  %>%
  inner_join(get_sentiments("nrc"))  %>%
  count (word, sentiment, sort = T)  %>%
  ungroup()

top_nrc_tidy_united <- bing_tidy_american[1:40,]
top_nrc_tidy_united %>%
  group_by(sentiment)%>%
  top_n(20,n)%>%
  ungroup%>%
  mutate(word=reorder(word,n))%>%
  ggplot(aes(x=word, y=n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales="free_y")+
  labs(y = "Contribution to sentiment American",
       x= NULL)+
  coord_flip()
```

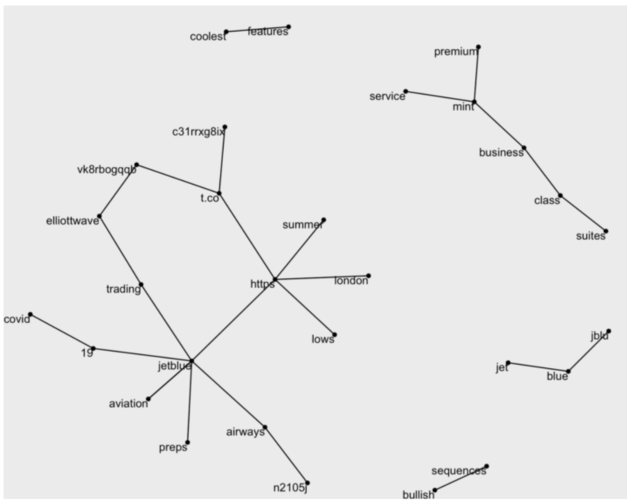



```
cloud_united_nrc <- tidy_united %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort=TRUE) %>% #token per sentiment
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey80"),
                   max.words=500, scale=c(1, 0.8),
                   title.size=1.2)
```



```
#############################################
############### JETBLUE #####################
#############################################

#############################################
###### N-grams and tokenizing ###############
#############################################
jetblue_bigrams <- jetblue %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

jetblue_bigrams #We want to see the bigrams (words that appear together, "pairs")

jetblue_bigrams %>%

#to remove stop words from the bigram data, we need to use the separate function:
jetblue_separated <- jetblue_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

jetblue_filtered <- jetblue_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
jetblue_counts <- jetblue_filtered %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
jetblue_counts

# A tibble: 748 x 3
   word1    word2          n
   <chr>    <chr>      <int>
 1 https    t.co          71
 2 jetblue  https         11
 3 business class          6
 4 jetblue  jetblue        6
 5 jetblue  airways        5
 6 mint     business       5
 7 mint     service        4
 8 premium  mint           4
 9 t.co     c31rrxg8ix     4
10 19       jetblue        3

###########################################################
###### What if we are interested in the most common #######
################ 4 consecutive words - quadro-gram ########
###########################################################
jetblue_quadrogram <- jetblue %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word)

jetblue_quadrogram

######################################################
####### VISUALISING A BIGRAM NETWORK ################
```

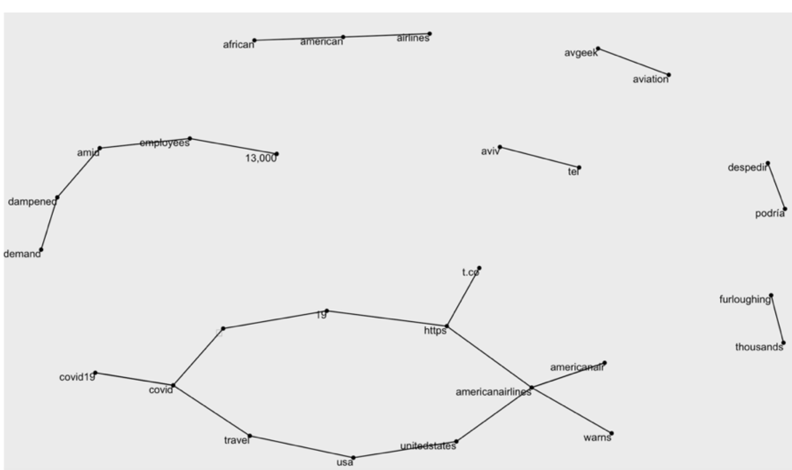
```
#####################################################

jetblue_bigram_graph <- jetblue_counts %>%
  filter(n>2) %>%
  graph_from_data_frame()

jetblue_bigram_graph

ggraph(jetblue_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```




```
############################################################
####################AMERICAN ##############################
############################################################

#############################################
###### N-grams and tokenizing ###############
#############################################

american_bigrams <- american %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

american_bigrams #We want to see the bigrams (words that appear together, "pairs")

american_bigrams %>%
  count(bigram, sort = TRUE)

american_separated <- american_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

american_filtered <- american_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
american_counts <- american_filtered %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
american_counts
```

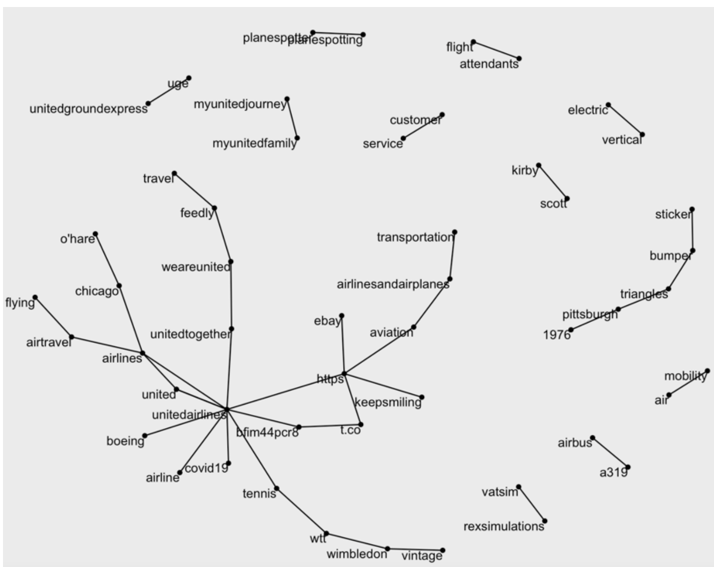
```
# A tibble: 2,787 x 3
   word1            word2                n
   <chr>            <chr>            <int>
 1 https            t.co               248
 2 american         airlines            59
 3 americanairlines https               37
 4 avgeek           aviation             9
 5 13,000           employees            8
 6 americanair      americanairlines     8
 7 americanairlines warns                8
 8 african          american             7
 9 podría           despedir             7
10 tel              aviv                 7
# … with 2,777 more rows
```

```
############################################################
###### What if we are interested in the most common #######
################ 4 consecutive words - quadro-gram ########
############################################################
american_quadrogram <- american %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word)
american_quadrogram

#######################################################
####### VISUALISING A BIGRAM NETWORK #################
#######################################################

american_bigram_graph <- american_counts %>%
  filter(n>5) %>%
  graph_from_data_frame()
american_bigram_graph

ggraph(american_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```



```
##############################################################
###################UNITED# ###############################
##############################################################

#############################################
###### N-grams and tokenizing ###############
#############################################

united_bigrams <- united %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

united_bigrams #We want to see the bigrams (words that appear together, "pairs")

united_bigrams %>%
  count(bigram, sort = TRUE) #this has many stop words, need to remove them

#to remove stop words from the bigram data, we need to use the separate function:
united_separated <- united_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

united_filtered <- united_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
united_counts <- united_filtered %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
united_counts

# A tibble: 1,395 x 3
   word1          word2              n
   <chr>          <chr>          <int>
 1 https          t.co             131
 2 united         airlines          17
 3 unitedairlines https             17
 4 united         unitedairlines     8
 5 1976           pittsburgh         7
 6 bumper         sticker            7
 7 ebay           https              7
 8 pittsburgh     triangles          7
 9 t.co           bfim44pcr8         7
10 bfim44pcr8     unitedairlines     6
# … with 1,385 more rows


############################################################
###### What if we are interested in the most common #######
################ 4 consecutive words - quadro-gram ########
############################################################
united_quadrogram <- united %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word)
```

```
united_quadrogram

######################################################
####### VISUALISING A BIGRAM NETWORK #################
######################################################

united_bigram_graph <- united_counts %>%
  filter(n>2) %>%
  graph_from_data_frame()

united_bigram_graph

ggraph(united_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```




```
################TFIDF Airlines####################
airlines <- bind_rows(mutate(jetblue, author = "JetBlue"),
                      mutate(american, author = "American"),
                      mutate(united, author = "United"))
airlines_tokens <- airlines %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words2) %>%
  count(word, sort=T)

#############################################
###### N-grams and tokenizing ###############
#############################################

airlines_bigrams <- airlines %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

airlines_bigrams #We want to see the bigrams (words that appear together, "pairs")

airlines_bigrams %>%
  count(bigram, sort = TRUE) #this has many stop words, need to remove them
```

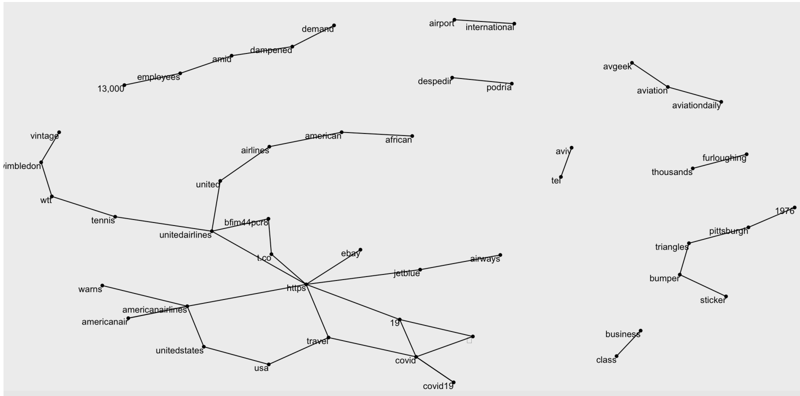
```
#to remove stop words from the bigram data, we need to use the separate function:
airlines_separated <- airlines_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

airlines_filtered <- airlines_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
airlines_counts <- airlines_filtered %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
airlines_counts

# A tibble: 4,699 x 3
   word1            word2                n
   <chr>            <chr>            <int>
 1 https            t.co               450
 2 american         airlines            60
 3 americanairlines https               38
 4 united           airlines            19
 5 unitedairlines   https               17
 6 avgeek           aviation            11
 7 jetblue          https               11
 8 business         class               10
 9 13,000           employees            8
10 americanair      americanairlines     8
# … with 4,689 more rows

############################################################
###### What if we are interested in the most common #######
################ 4 consecutive words - quadro-gram ########
############################################################

airlines_quadrogram <- airlines %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word)

airlines_quadrogram

############################################################
###### We can also apply the tf_idf framework  ############
########### on our bigram and quadro-gram #################
############################################################

airlines_united <- airlines_filtered %>%
  unite(bigram, word1, word2, sep=" ") #we need to unite what we split in the previous section

airlines_bigram_tf_idf <- airlines_united %>%
  count(author, bigram) %>%
  bind_tf_idf(bigram, author, n) %>%
  arrange(desc(tf_idf))

airlines_bigram_tf_idf
```

```
# A tibble: 4,930 x 6
   author   bigram                    n      tf   idf  tf_idf
   <chr>    <chr>                 <int>   <dbl> <dbl>   <dbl>
 1 JetBlue  jetblue https            11 0.0120  1.10  0.0132
 2 United   unitedairlines https     17 0.00966 1.10  0.0106
 3 JetBlue  jetblue jetblue           6 0.00654 1.10  0.00718
 4 American american airlines        59 0.0156  0.405 0.00632
 5 JetBlue  jetblue airways           5 0.00545 1.10  0.00598
 6 JetBlue  mint business             5 0.00545 1.10  0.00598
 7 United   united unitedairlines     8 0.00455 1.10  0.00500
 8 JetBlue  mint service              4 0.00436 1.10  0.00479
 9 JetBlue  premium mint              4 0.00436 1.10  0.00479
10 JetBlue  t.co c31rrxg8ix           4 0.00436 1.10  0.00479
# … with 4,920 more rows
```

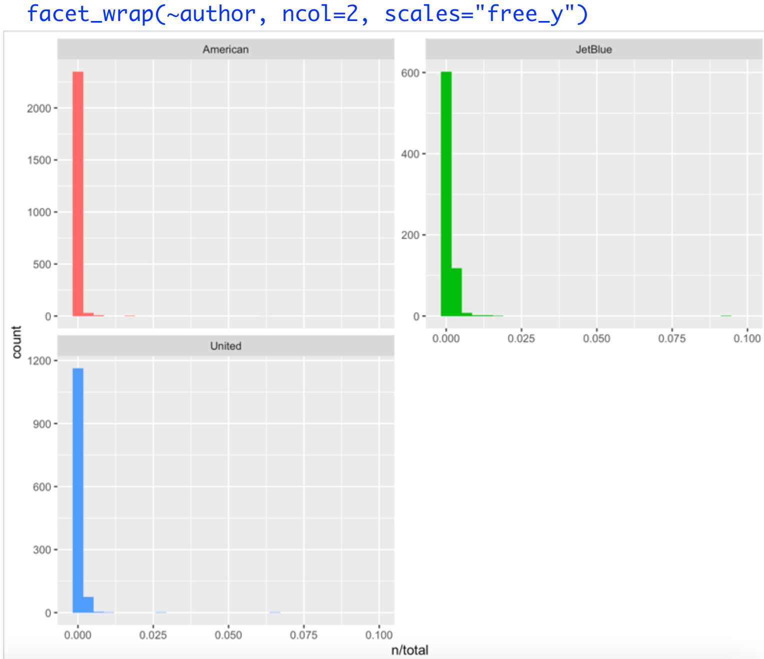
```
##### lets do the same for a quadrogram

airlines_quadrogram_united <- airlines_quadrogram %>%
  unite(quadrogram, word1, word2, word3, word4, sep=" ") #we need to unite what we split in the previous section

airlines_quadrogram_tf_idf <- airlines_quadrogram_united %>%
  count(author, quadrogram) %>%
  bind_tf_idf(quadrogram, author, n) %>%
  arrange(desc(tf_idf))

airlines_quadrogram_tf_idf
```

```
# A tibble: 3,721 x 6
   author  quadrogram                              n      tf   idf  tf_idf
   <chr>   <chr>                               <int>   <dbl> <dbl>   <dbl>
 1 United  ebay https t.co bfim44pcr8              7 0.00621  1.10 0.00682
 2 JetBlue london https t.co c31rrxg8ix            3 0.00565  1.10 0.00621
 3 JetBlue mint business class suites              3 0.00565  1.10 0.00621
 4 United  1976 pittsburgh triangles bumper        6 0.00532  1.10 0.00584
 5 United  bfim44pcr8 unitedairlines tennis…       6 0.00532  1.10 0.00584
 6 United  https t.co bfim44pcr8 unitedairl…       6 0.00532  1.10 0.00584
 7 United  pittsburgh triangles bumper stic…       6 0.00532  1.10 0.00584
 8 United  t.co bfim44pcr8 unitedairlines t…       6 0.00532  1.10 0.00584
 9 United  tennis wtt wimbledon vintage            6 0.00532  1.10 0.00584
10 United  unitedairlines tennis wtt wimble…       6 0.00532  1.10 0.00584
# … with 3,711 more rows
```

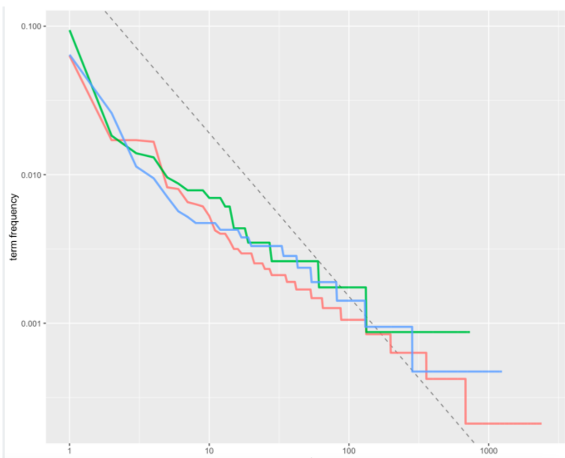
```
######################################################
####### VISUALISING A BIGRAM NETWORK #################
######################################################

airlines_bigram_graph <- airlines_counts %>%
  filter(n>5) %>%
  graph_from_data_frame()

airlines_bigram_graph

ggraph(airlines_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```




```
##########################################################
################ tf_idf Analysis########################
##########################################################

tf_idf_airlines <- bind_rows(mutate(jetblue, author = "JetBlue"),
                             mutate(american, author = "American"),
                             mutate(united, author = "United")) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words2) %>%
  count(author, word, sort=TRUE) %>%
  ungroup()

total_words <- tf_idf_airlines %>%
  group_by(author) %>%
  summarize(total=sum(n))

airlines_words <- left_join(tf_idf_airlines, total_words)

print(airlines_words)

ggplot(airlines_words, aes(n/total, fill = author))+
  geom_histogram(show.legend=FALSE)+
  xlim(NA, 0.1) +
  facet_wrap(~author, ncol=2, scales="free_y")
```



```
#####################################
########## ZIPF's law ###############
#####################################

freq_by_rank <- airlines_words %>%
  group_by(author) %>%
  mutate(rank = row_number(),
         `term frequency` = n/total)
freq_by_rank

# plot ZIPF's Law
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color=author))+
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()
```




```
##################################################
################ TF_IDF ##########################
##################################################
airlines_words_idf <-airlines_words %>%
  bind_tf_idf(word, author, n)

#reorganize the table
airlines_words_idf %>%
  arrange(desc(tf_idf))
```

```
# A tibble: 4,311 x 7
   author   word               n total      tf   idf  tf_idf
   <chr>    <chr>          <int> <int>   <dbl> <dbl>   <dbl>
 1 United   unitedairlines   142  2160 0.0657  0.405 0.0267
 2 JetBlue  mint              21  1097 0.0191  1.10  0.0210
 3 United   united            57  2160 0.0264  0.405 0.0107
 4 JetBlue  premium            8  1097 0.00729 1.10  0.00801
 5 JetBlue  suites             7  1097 0.00638 1.10  0.00701
 6 JetBlue  airways            6  1097 0.00547 1.10  0.00601
 7 American 13,000            25  4589 0.00545 1.10  0.00599
 8 American miami             20  4589 0.00436 1.10  0.00479
 9 United   unitedtogether     8  2160 0.00370 1.10  0.00407
10 JetBlue  iberia             4  1097 0.00365 1.10  0.00401
```

```
# … with 4,301 more rows

#graphical approach
airlines_words_idf %>%
  anti_join(stop_words2) %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(author) %>%
  top_n(10) %>% #top highest tfidf tokens
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=author))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~author, ncol=2, scales="free")+
  coord_flip()
```