Airlines Text Analysis

Ashwin Malshé

8/22/2020

Introduction

Text analysis is a commonly used marketing research tool. Using text analysis on product reviews, discussion forums, and social media posts and comments, we can obtain important insights about brands. In this example, we will analyze tweets pertaining to six major US airlines. We will carry out two types of analyses:

1. Wordcloud: A wordcloud is a visualization of words in a textual data set such that the size (and sometimes color) of a word is mapped to the frequency of the word in the text. Thus, the words that occur more often in the text show up larger in the wordcloud. This is a simple way of showing which words people are using while talking about a brand.
2. Sentiment analysis: This analysis captures the sentiment underlying the text. Most commonly, text is categorized simply as positive or negative. That might be sufficient in some cases such as knowing how many product reviews are positive. However, more sophisticated sentiment analysis can categorize text into specific sentiments such as sadness, joy, anger, etc.

Background

For the past several years, US airlines have shifted a large part of their customer service to Twitter. Therefore, increasingly, fliers tweet to the airlines directly rather than emailing or calling them. This presents a great opportunity to us to do text analysis on the tweets sent to the airlines!

We collected around 100,000 tweets sent to the following six major US airlines in June and July 2020:

1. Alaska
2. American
3. Delta
4. JetBlue
5. Southwest
6. United

In this lesson, we will use R, a popular open-source statistical programming language to carry out text analysis on these tweets.

Setup

R is a powerful software on top of which people have built many modules to carry out specific tasks. For text analysis and a little bit of data manipulation, we will use four modules or packages. Let's first load them in our current working session.

Load libraries

Click on the green triangle in the top right corner of the following gray box.

If you did not get any error messages, the packages are now loaded. This means that we are ready to carry out the next step of loading the tweets in the current session.

Load the data

R can read many different types of data files. Most of you probably use Excel to store your data. There are packages in R that can read Excel files directly into R. I had previously saved the tweets into a data format that R understand natively. It has `.rds` extension and the data file can be read using a function `readRDS()`. In the next code chunk (that is the gray box), we are simply reading the tweets saved in a file titled `airlines-data.rds` and then storing in an object titled `airlines_data`.

Once again, click on the green triangle on right to run this code.

```
airlines_data <- readRDS("airlines-data.rds")
```

If you were successful in running this code, you should now see the data object `airlines_data` appearing in the top right pane titled "Data". Now we are ready to perform text analysis!

Wordcloud

I have written a function titled `make_wordcloud()` that will create a wordcloud by taking the name of an airline as an input from you. Let's create a wordcloud for American Air. Note that depending on the size of the data set, it may take a few seconds to create the wordcloud. So, please be patient!

```
make_wordcloud(airline = "American")
```

Interestingly, "mask" shows up as the largest, and therefore more frequent, word in the tweets addressed to American! Recall that many people wanted to travel in summer and airlines were not uniformly applying social distancing and mask wearing rules. Therefore, it is understandable that people were mentioning masks in their tweets. Perhaps they wanted to know whether American has a mandatory mask policy on its flights.

It is difficult to say whether appearance of certain words in the tweets is a good thing or a bad thing. For instance, people using curse words in their tweets is clearly a bad thing. However, using the word "mask" doesn't tell us anything about the valence of the tweet. This is a limitation of wordclouds.

Exercise 1

The wordcloud above looks a bit cluttered. This is because, by default `make_wordcloud` includes all the words that have appeared at least 200 times in all the tweets sent to an airline. I used 200 as a reasonable cutoff but it is not the most optimum threshold. You can tweak that by setting the argument `min_freq` in the function. If you increase this number, the wordcloud will be become sparse. Reducing this number will make the wordcloud dense.

Give it a spin! Start with something large like 500 and see what happens.

```
make_wordcloud(airline = "American", min_freq = )
```

Exercise 2

There are six airline tweets in this data. How about creating a wordcloud for Southwest or any other airline?

For this, you will need to provide the name of the airline in the function. This name has to match EXACTLY to any of the names below. *Also you have to provide the name WITH the quotes.* You may use single or double quotes.

"Alaska" "American" "Delta" "Jetblue"
"Southwest" "United"

If you make any changes to the names, the function won't run and instead throw an error. Often these errors are non informative and scary! So please pay attention to what you are inputting.

```
# Make sure the airline name is wrapped in quotes!

make_wordcloud(airline = )
```

Now you can compare your wordcloud with the American Airlines wordcloud above. Here are the things you should focus on:

1. Are the most frequent words in two wordclouds different?

a. If they are not different, what can you infer from that?

b. If they are different, what can you infer from that?

2. Can you piece together a story about each wordcloud based on the frequently used words?
3. Did anything surprise you?
4. What will you tell the marketing managers of these airlines that will give them new insights about their fliers?

Sentiment analysis

Imagine that you are a tourist visiting USA but you don't speak English well. You know a few words but that's about it. While waiting for a cab, someone points to your shoes and says "Nice shoes, where did you buy them?" Now you don't know so many words in English but you know what "nice" means. Would you consider the valence of the question positive or negative? Well, based on that one word, it is likely that you will think it is a positive question. This way of doing the sentiment analysis is knows as a "lexicon-based" sentiment analysis. This is because you have a lexicon or a dictionary of English words in your head and you are simply matching the word you just came across with the lexicon to determine its valence. Because the word "nice" is positive, you infer that the whole sentence is positive.

Of course, this method completely disregards many nonverbal cues such as the gestures and facial expressions of the person who asked the question. You also disregard the tone in which the question was asked. Additionally, this method may lead to incorrect inferences when more complicated language structure exists. For instance, if the person said "Nice shoes, my shoes are terrible!" Is this sentence positive for you or not? With lexicon-based sentiment analysis, you will cancel out nice with terrible and conclude that this sentence is neutral, where in fact it is still positive for you!

Although there are numerous different methods to do a sentiment analysis, lexicon-based method still remains one of the most popular ways to do sentiment analysis. Commonly, researchers create lexicons that document word lists pertaining to specific emotion or valence. For instance, following words are negative: sad, bad, unhappy, and terrible while the following words are positive: happy, great, fantastic, and awesome. One such lexicon is NRC Word-Emotion Association Lexicon created by the researcher Saif Mohammed. Read more about it by visiting NRC website.

JetBlue sentiment analysis

NRC categorizes each tweet into eight emotions:

Anger Anticipation Disgust Fear Joy Sadness Surprise Trust

Additionally, it also output Positive and Negative valence of the tweets.

I have written a function titled `get_sentiment()` which creates a table of various emotions.

```
jetblue_sent <- get_sentiment(airline = "Jetblue")
```

You can take a look at the first few rows of this data set by running the following function. It will print out first 10 observations. Each row corresponds to a tweet.

```
head(jetblue_sent, 10)
```

```
##    airline          created_at anger anticipation disgust fear  joy sadness
## 1  Jetblue 2020-07-29 13:46:23  0.00         5.88       0 0.00 5.88    0.00
## 2  Jetblue 2020-06-21 16:05:50  0.00         4.65       0 0.00 2.33    0.00
## 3  Jetblue 2020-06-02 23:02:34  0.00         0.00       0 0.00 0.00    0.00
## 4  Jetblue 2020-06-18 14:28:22  0.00         2.63       0 0.00 2.63    0.00
## 5  Jetblue 2020-06-18 17:20:34  0.00         7.69       0 0.00 7.69    0.00
## 6  Jetblue 2020-07-20 13:40:52  0.00         0.00       0 4.00 0.00    0.00
## 7  Jetblue 2020-07-10 13:35:41  2.08         0.00       0 2.08 0.00    4.17
## 8  Jetblue 2020-06-01 17:21:38  0.00         1.89       0 1.89 1.89    1.89
## 9  Jetblue 2020-06-17 23:33:14  2.70         5.41       0 2.70 5.41    0.00
## 10 Jetblue 2020-07-16 13:50:53  0.00         0.00       0 2.04 0.00    0.00
##    surprise trust positive negative       WPS WC           status_id
## 1      0.00  5.88    11.76     0.00 17.000000 17 1288470933632454656
## 2      2.33  0.00     6.98     0.00 14.333333 43 1274735288573726726
## 3      0.00  1.82     3.64     0.00 13.750000 55 1267954791608614912
## 4      2.63  0.00     5.26     0.00  7.600000 38 1273623594409623552
## 5      7.69  0.00    15.38     0.00  4.333333 13 1273666931070042113
## 6      0.00  4.00     8.00     0.00 12.500000 25 1285208053458100225
## 7      0.00  2.08     0.00     8.33 12.000000 48 1281582869048066048
## 8      1.89  3.77     5.66     0.00 13.250000 53 1267506603835654148
## 9      0.00  5.41    10.81     2.70 12.333333 37 1273398326747168775
## 10     2.04  2.04     2.04     0.00  9.800000 49 1283761021048455169
```

The numbers in the columns pertaining to various emotions are percentage of words in the tweet. The total number of words in a tweet is in the column `WC`, which is short for word count.

Let's print out the summary sentiment for JetBlue. Don't worry about the code. Just note that we are using `jetblue_sent`. Later when you want to analyze another airline, you can just replace this object with another pertaining to that airline.

```
jetblue_sent %>%
  select(anger, anticipation,  disgust, fear, joy, sadness, surprise, trust,
         positive, negative) %>%
  summary()
```

```
##      anger          anticipation       disgust             fear
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000
```

```
##  1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.000
##  Median : 0.0000   Median : 0.000   Median : 0.0000   Median : 0.000
##  Mean   : 0.8137   Mean   : 1.916   Mean   : 0.5585   Mean   : 1.052
##  3rd Qu.: 0.0000   3rd Qu.: 3.120   3rd Qu.: 0.0000   3rd Qu.: 0.000
##  Max.   :33.3300   Max.   :50.000   Max.   :33.3300   Max.   :33.330
##       joy            sadness          surprise          trust
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.00
##  1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.00
##  Median : 0.000   Median : 0.000   Median : 0.0000   Median : 0.00
##  Mean   : 1.562   Mean   : 1.139   Mean   : 0.8292   Mean   : 2.35
##  3rd Qu.: 2.000   3rd Qu.: 0.000   3rd Qu.: 0.0000   3rd Qu.: 4.00
##  Max.   :50.000   Max.   :33.330   Max.   :50.0000   Max.   :33.33
##     positive        negative
##  Min.   : 0.00   Min.   : 0.000
##  1st Qu.: 0.00   1st Qu.: 0.000
##  Median : 2.63   Median : 0.000
##  Mean   : 3.76   Mean   : 1.883
##  3rd Qu.: 6.06   3rd Qu.: 3.120
##  Max.   :50.00   Max.   :33.330
```

Although we have multiple summary statistics available, let's focus on the mean or average. First, we see that on average there are 3.76% positive words in a tweet compared to only 1.883% negative words. That's a large difference.

Exercise 3

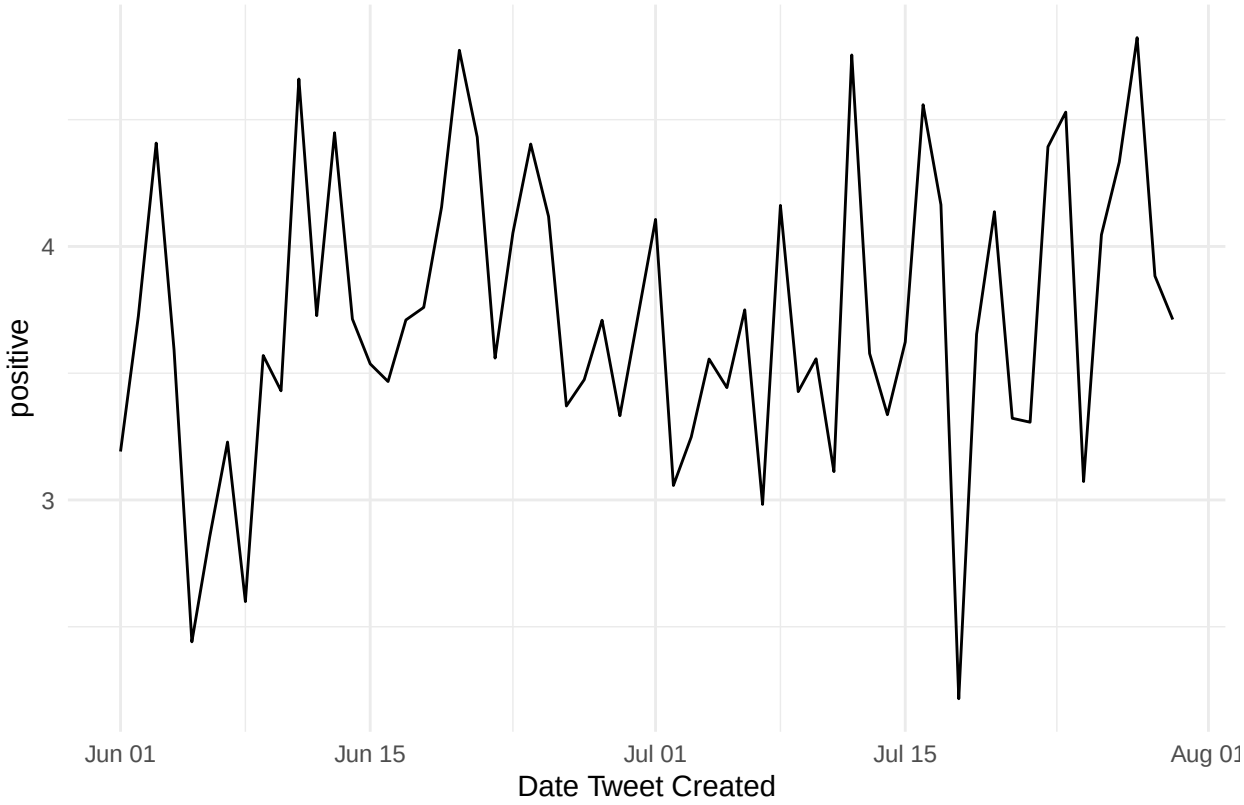
Study the average scores for the other sentiments. Is there anything interesting you find there?

Time series graph

Note that we have the dates when the tweets were sent. Let's make a line graph to plot the average daily sentiment. I have created a function which needs two inputs. First, we have to supply it the data with sentiment. Second, we supply the emotion we want to plot.

Let's make a line graph with positive sentiment.

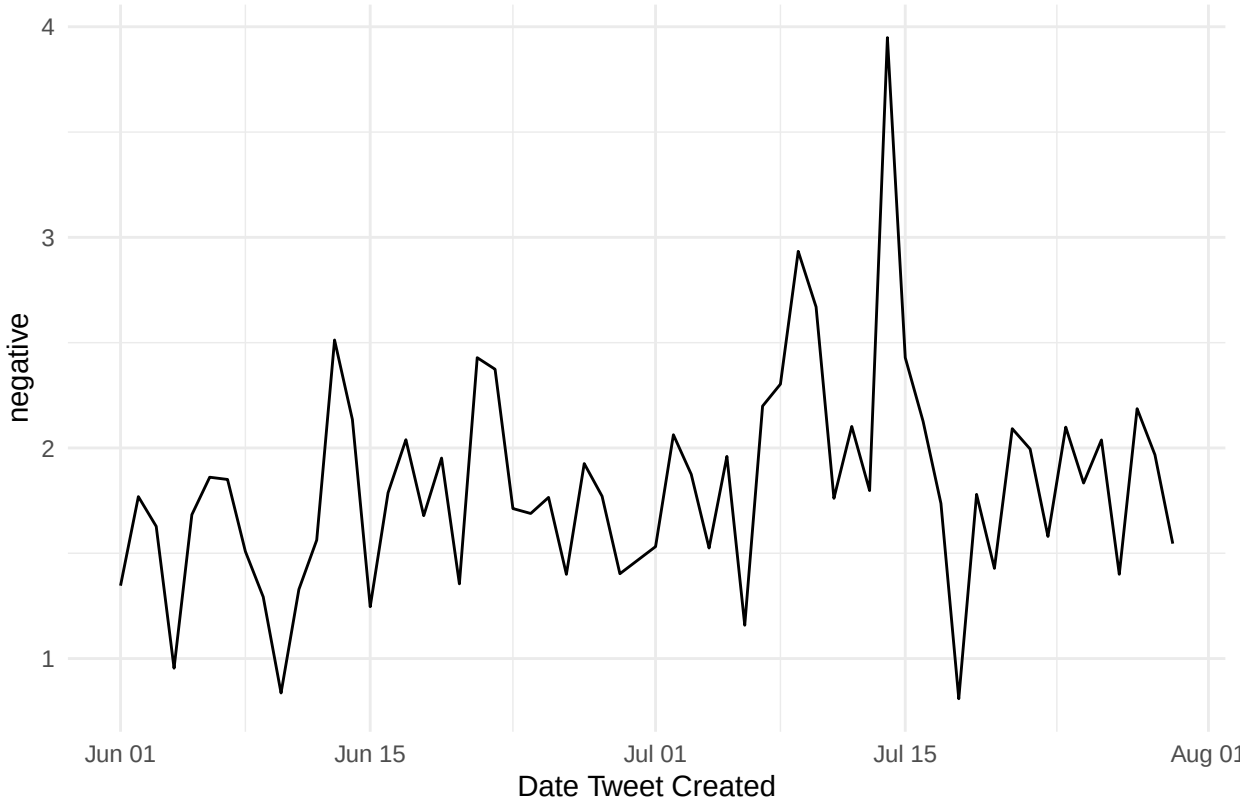
```
make_lineplot(data = jetblue_sent, emotion = positive)
```



Graph created using Ashwin Malshe's GenAI lesson.

And let's compare it with a line graph with negative sentiment.

```
make_lineplot(data = jetblue_sent, emotion = negative)
```



Graph created using Ashwin Malshe's GenAI lesson.

Interestingly, we get quite different trends for positive and negative sentiment for JetBlue. For instance, on 14th July 2020, the negativity in the tweets shot up considerably. If you compare that to the positive sentiment, we don't observe anything odd on that day. This suggests that JetBlue managers should look into this event to understand the cause of this negativity.

Of course, looking into this today is probably meaningless. However, marketing managers can monitor Twitter sentiment in real time too. This can help them avoid crisis-like situations.

Exercise 4

Repeat the above exercise for JetBlue with other emotions. Note that you have to use correct name for the emotion as it shows up in the data set. For instance angry tweets will be shown with this code:

```
make_lineplot(data = jetblue_sent, emotion = anger)
```

The emotion is `anger` because that's the name of column in our data set.

Exercise 5

Create a sentiment data set for another airline of your choice. Compare the sentiment trends for your chosen airline and JetBlue.

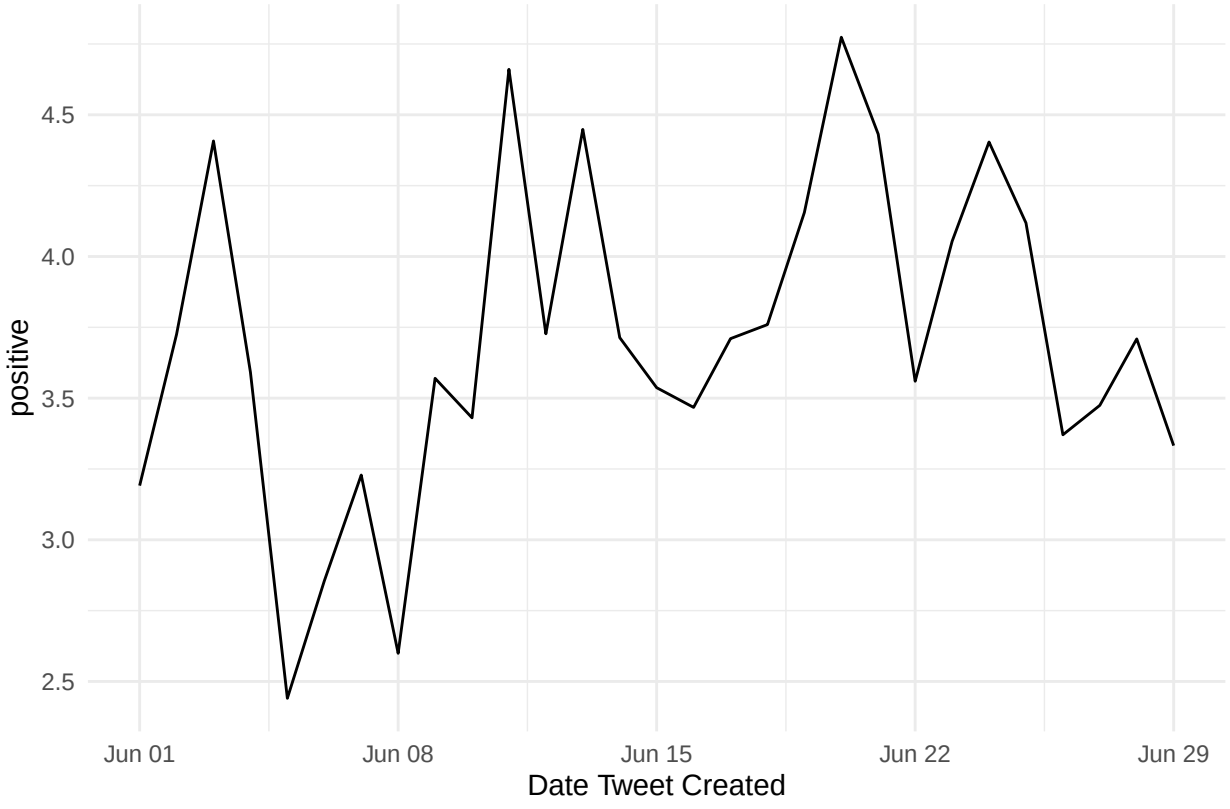
Exercise 6

The function `make_lineplot()` can take two additional inputs for the start and end date. Our data set spans 1st of June 2020 to 31st July 2020. However, if you want to focus on a smaller date range, you can do that by specifying a start date and an end date.

For instance, for JetBlue, let's get trend only for June 2020.

Note that the date has to be specified in yyyy-mm-dd format and wrapped in quotes. If you don't follow this, your will get an error.

```
make_lineplot(data = jetblue_sent, emotion = positive,
              start_date = "2020-06-01", end_date = "2020-06-30")
```



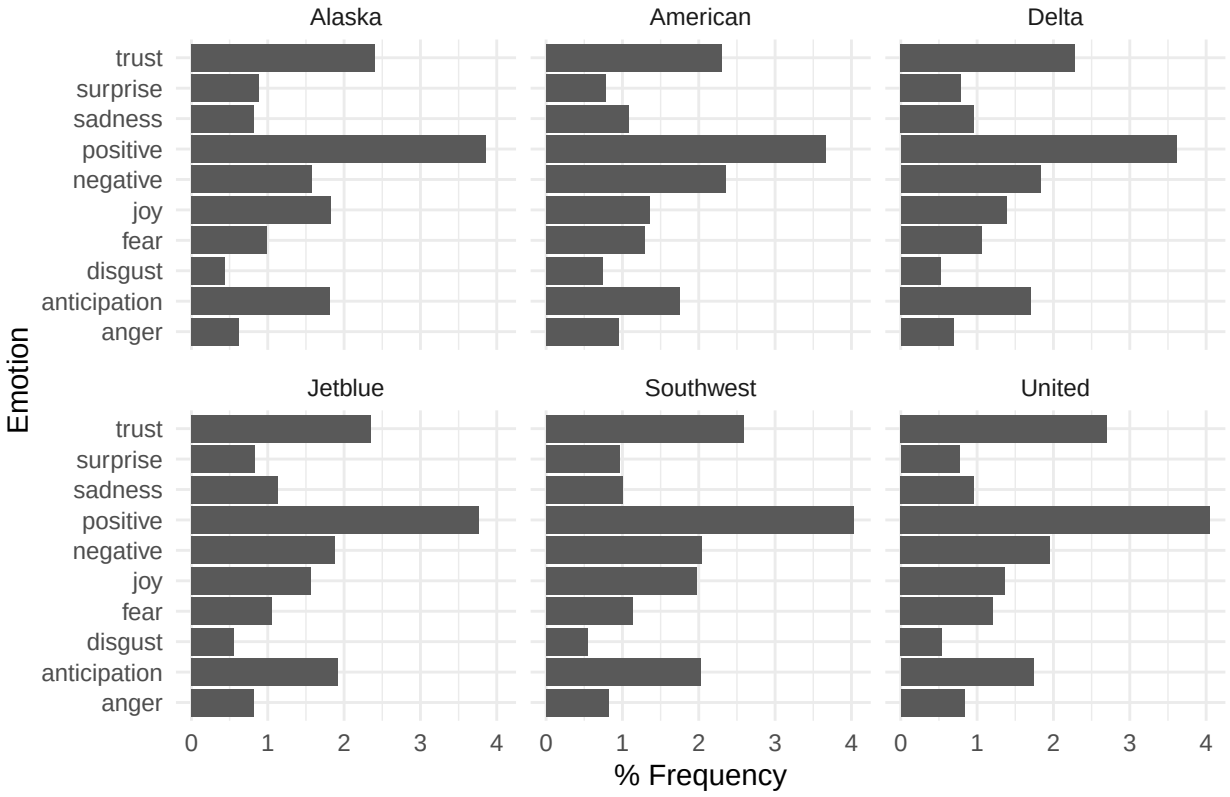
Graph created using Ashwin Malshe's GenAI lesson.

Change these dates and see how your line plot changes.

Comparing all the airlines

Finally, we will create a bar graph that compares the sentiment from all the airlines in one plot. I have created a function to make this plot, which doesn't require any input! Just run it below and see the output.

```
make_barplot()
```



Graph created using Ashwin Malshe's GenAI lesson.

Conclusion

In this lesson, you learned how to use Twitter to create wordclouds and perform sentiment analysis. This lesson uses R programming language. If you want to learn more about text analysis using R, I recommend this free book: https://www.tidytextmining.com.

If this lesson made you curious to learn more about data science, please write to me at ashwin.malshe@utsa.edu. UTSA has a one-year masters program in data analytics. Learn more about it here: https://business.utsa.edu/programs/ms-data-analytics/