



# A Text Analysis of Trump's Tweets

## DataCamp Project

**Kelompok 2**

Pembimbing: **Budi Sujatmiko, dr., M. Epid.**

# MEET THE TEAM



Pembimbing :  
**Budi Sujatmiko, dr., M.Epid.**



**Anindya Nadira R.**  
NPM. 130110200062



**Farsya Azka K.**  
NPM. 130110200066



**Raihan Al Biruni**  
NPM. 130110210138

# TABLE OF CONTENTS



01

## Introduction

Project description.

03

## Results

Project task's results.

02

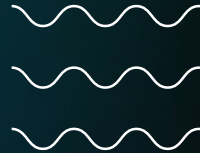
## Methods

Project requirements, methods, and tasks.

04

## Discussion

Result discussion & our innovative idea.





01

# Introduction

---

Project Description



# Project Description



**Todd Vaziri**

@tvaziri


Every non-hyperbolic tweet is from iPhone (his staff).

Every hyperbolic tweet is from Android (from him).

2:20 AM · Aug 7, 2016 · Tweetbot for iOS

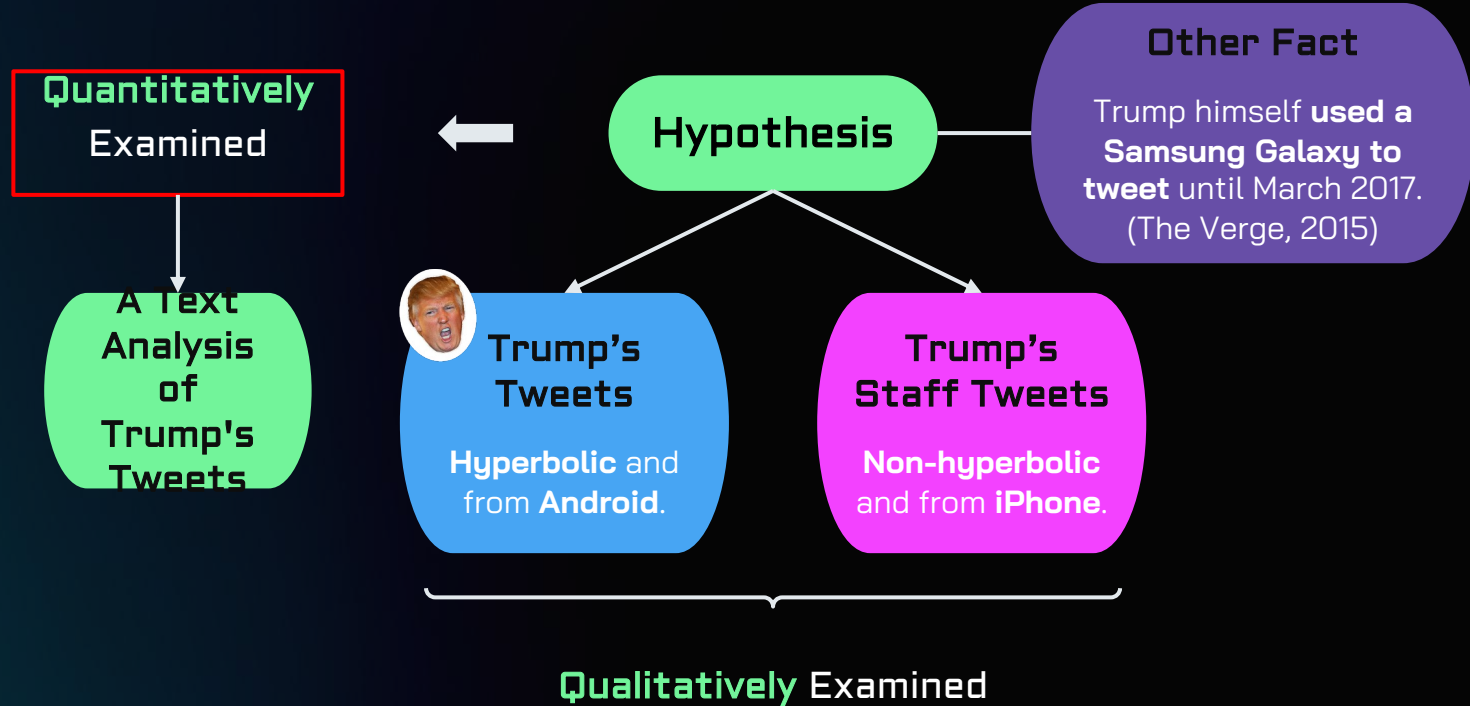
	<b>Donald J. Trump</b> @realDonaldTrump Good luck #TeamUSA #OpeningCeremony #Rio2016 <a href="https://pic.twitter.com/mS8qsQpJPh">pic.twitter.com/mS8qsQpJPh</a>
27,391 Likes	8,392 Retweets
Aug 5, 2016 at 8:59 PM	via <b>Twitter for iPhone</b>

	<b>Donald J. Trump</b> @realDonaldTrump Heading to New Hampshire - will be talking about Hillary saying her brain SHORT CIRCUITED, and other things!
4,451 Likes	1,480 Retweets
Aug 6, 2016 at 11:11 AM	via <b>Twitter for Android</b>

Trump's tweets differences. ([Twitter.com](https://twitter.com))

# Project Description





02





# Methods



















---

Project Requirements and Methods



# Prerequisites



<div></div> <div><h3>Sentiment Analysis in R</h3></div> <div><p>Learn sentiment analysis by identifying positive and negative language, specific emotional intent...</p></div> <div></div> <div><div> Share</div><div>Replay Course</div></div>	<div></div> <div><h3>String Manipulation with stringr in R</h3></div> <div><p>Learn how to pull character strings apart, put them back together and use the stringr package.</p></div> <div></div> <div><div> Share</div><div>Replay Course</div></div>
<div></div> <div><h3>Introduction to the Tidyverse</h3></div> <div><p>Get started on the path to exploring and visualizing your own data with the tidyverse, a powerful...</p></div> <div></div> <div><div> Share</div><div>Replay Course</div></div>	<div></div> <div><h3>Intermediate Data Visualization with ggplot2</h3></div> <div><p>Learn to use facets, coordinate systems and statistics in ggplot2 to create meaningful explainer...</p></div> <div></div> <div><div> Share</div><div>Replay Course</div></div>



# Prerequisite: Sentiment Analysis in R

## 1. Fast & dirty: Polarity scoring

- Learn how to apply qdap's sentiment function called `polarity()`

## 1. Sentiment analysis the tidytext way

- Explore 3 subjectivity lexicons from tidytext
- Do an inner join to score some text

## 1. Visualizing sentiment

- Make compelling visuals with your sentiment output

## 1. Case study: Airbnb reviews



# Prerequisite: String Manipulation with **stringr** in R

## 1. **String basics**

- How to enter strings in R
- How to control how numbers are transformed to strings
- How to combine strings together to produce output that combines text and nicely formatted numbers.

## 1. **Introduction to stringr**

- How to detect specific patterns in strings
- How to split strings apart
- How to find and replace parts of strings.

## 1. **Pattern matching with regular expressions**

- Regular expressions, a language for describing patterns in strings → manipulate strings

## 1. **More advanced matching and manipulation**

- Capturing, back-referencing, é



# Prerequisite: Introduction to the Tidyverse

## Data Wrangling

Filter & arrange observation; mutate to add or change a column.

```
filter()  
arrange()  
mutate()
```

## Coordinates

Create data visualizations after filtering, arranging, and mutating data.

```
library(ggplot2)  
ggplot()  
%>%
```

## Grouping & summarizing

Collapsing large datasets into manageable summaries.

```
group_by()  
summarise()
```

## Type of Visualization

Create and interpret diagram: line-, bar-, boxplot, and histogram.

```
geom_line()  
geom_bar()  
geom_boxplot()  
geom_histogram()
```

# Prerequisite: Intermediate Data Visualization with ggplot2

## Statistic

Simple data plotting and applying variety of statistical methods.

```
geom_point()  
geom_jitter()  
mean()  
median()  
sd()
```

## Data Visualization

Create and interpret coordinates layers.

```
aes()
```

## Facets

Splitting plots into multiple panes based on subsets of the dataset.


```
facet_wrap()
```

## Best Practices

Create and interpret heat maps, pie charts, and dynamite plots.



```
geom_tile()  
geom_bar()  
coord_polar()  
stat_summary()
```

# Method: Text Mining & Sentiment Analysis



Learned about this mental map demonstrating how **text mining is the process of going from an unorganized state to an organized state** ⇒ techniques aligning to feature extraction.

6 defined steps:

- 
- 
1. Define the problem & specific goals
  2. Identify the text
  3. Organize the text
  4. Extract features
  5. Analyze
  6. Draw a conclusion/reach an insight

# Tasks



## Clean those tweets

Clean the data: select, filter, extract, etc.



## The quote tweet is dead

Data manipulation and visualization.

## Comparison of words

Data manipulation.

1

2

3

4

5

6

## The tweets

Load the data.

## Is "time" the giveaway?

Data manipulation and visualization.

## Links and pictures

Data manipulation and visualization.



# Tasks



**Common words:  
Android vs. iPhone (i)**

Data manipulation.



**Adding  
sentiments**

Data manipulation.

**Conclusion**

Project conclusion.

7

**Most common  
words**

Data manipulation and  
visualization.

8

**Common words:  
Android vs. iPhone (ii)**

Data manipulation and  
visualization.

9

10

**Android vs. iPhone  
sentiments**

Data manipulation and  
visualization.

11

12





# 03

# Results

---

Project Task and Results

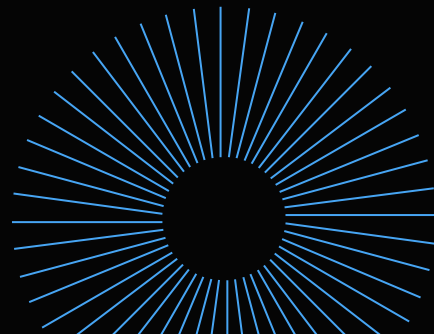




# Task 1 | The Tweets

**Goals**

Load the data.



# 1. The Tweets

## STEP 01: *Load the libraries*

```
library(dplyr)
library(readr)
library(tidyr)
library(lubridate)
```

## STEP 02: *Read in the data*

```
tweets <- read_csv("datasets/trump_tweets.csv", guess_max = 36000) %>%
  filter(created_at >= "2015-06-01" , created_at <= "2016-11-08")
```

## STEP 03: *Inspect the first six rows*

head(tweets)

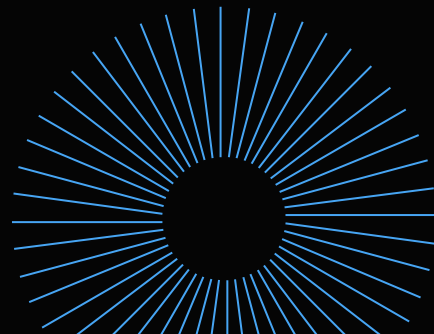
source	id_str	text	created_at	retweet_count	in_reply_to_user_id_str	favorite_count	is_retweet
<chr>	<dbl>	<chr>	<dtm>	<dbl>	<dbl>	<dbl>	<lgl>
Twitter for Android	6.827032e+17	I would like to wish everyone A HAPPY AND HEALTHY NEW YEAR. WE MUST ALL WORK TOGETHER TO, FINALLY, MAKE AMERICA SAFE AGAIN AND GREAT AGAIN!	2015-12-31 23:21:49	6776	NA	16495	FALSE
Twitter for Android	6.827007e+17	Do you believe that The State Department, on NEW YEAR'S EVE, just released more of Hillary's e-mails. They just want it all to end. BAD!	2015-12-31 23:11:35	2755	NA	6824	FALSE
Twitter for iPhone	6.826351e+17	THANK YOU ILLINOIS! Let's not forget to get family & friends- out to VOTE IN 2016! <a href="https://t.co/fg5kMbNLYK">https://t.co/fg5kMbNLYK</a> <a href="https://t.co/dtMASlq4cf">https://t.co/dtMASlq4cf</a>	2015-12-31 18:51:12	2468	NA	6047	FALSE



## Task 2 | Clean those tweets

### Goals

To clean the data by extracting iPhone and Android tweets only.



## 2. Clean Those Tweets

### Step 01

Count the number of tweets by device!

```
tweets %>% count(source)
```

Output [01]:

source	n
<chr>	<int>
Twitter for Android	4240
Twitter for iPhone	2275
Twitter Web Client	1301
Twitter Ads	63
Instagram	47
Twitter for BlackBerry	45

### Step 02

Select certain variables!

```
cleaned_tweets <- tweets %>%  
  select(id_str, source, text, created_at)
```

Output [02]:

id_str	source	text	created_at
<dbl>	<chr>	<chr>	<dtm>
6.827032e+17	Twitter for Android	I would like to wish everyone A HAPPY AND HEALTHY NEW YEAR. WE MUST ALL WORK TOGETHER TO, FINALLY, MAKE AMERICA SAFE AGAIN AND GREAT AGAIN!	2015-12-31 23:21:49
6.827007e+17	Twitter for Android	Do you believe that The State Department, on NEW YEAR'S EVE, just released more of Hillary's e-mails. They just want it all to end. BAD!	2015-12-31 23:11:35
6.826351e+17	Twitter for iPhone	THANK YOU ILLINOIS! Let's not forget to get family & friends- out to VOTE IN 2016! <a href="https://t.co/Ig5kMbNLYK">https://t.co/Ig5kMbNLYK</a> <a href="https://t.co/dtMAslq4cf">https://t.co/dtMAslq4cf</a>	2015-12-31 18:51:12
6.826053e+17	Twitter for iPhone	HAPPY BIRTHDAY to my son, @DonaldJTrumpJr! Very proud of you! #TBT <a href="https://t.co/ULerCEOCGX">https://t.co/ULerCEOCGX</a> <a href="https://t.co/nbxPVdarJM">https://t.co/nbxPVdarJM</a>	2015-12-31 16:52:38
6.825788e+17	Twitter for Android	I would feel sorry for @JebBush and how badly he is doing with his campaign other than for the fact he took millions of \$'s of hit ads on me	2015-12-31 15:07:18
6.825446e+17	Twitter for iPhone	#MakeAmericaGreatAgain #Trump2016 <a href="https://t.co/ElXos0wh9">https://t.co/ElXos0wh9</a>	2015-12-31 12:51:35

## 2. Clean Those Tweets

### Step 03

#### Filter with Twitter for Android and iPhone only!

```
cleaned_tweets <- tweets %>%  
  select(id_str, source, text, created_at) %>%  
  filter(source %in% c("Twitter for  
    iPhone", "Twitter for Android"))
```

### Step 04

#### Extract "Android" and "iPhone" only!

```
cleaned_tweets <- tweets %>%  
  select(id_str, source, text, created_at) %>%  
  filter(source %in% c("Twitter for  
    iPhone", "Twitter for Android")) %>%  
  extract(source, "source", "(\\w+)$")
```

### Output [03]:

id_str	source	text	created_at
<dbl>	<chr>	<chr>	<dtm>
6.827032e+17	Twitter for Android	I would like to wish everyone A HAPPY AND HEALTHY NEW YEAR. WE MUST ALL WORK TOGETHER TO, FINALLY, MAKE AMERICA SAFE AGAIN AND GREAT AGAIN!	2015-12-31 23:21:49
6.827007e+17	Twitter for Android	Do you believe that The State Department, on NEW YEAR'S EVE, just released more of Hillary's e-mails. They just want it all to end. BAD!	2015-12-31 23:11:35
6.826351e+17	Twitter for iPhone	THANK YOU ILLINOIS! Let's not forget to get family & friends- out to VOTE IN 2016! <a href="https://t.co/Ig5kMbNLYK">https://t.co/Ig5kMbNLYK</a> <a href="https://t.co/dtMASlq4cf">https://t.co/dtMASlq4cf</a>	2015-12-31 18:51:12
6.826053e+17	Twitter for iPhone	HAPPY BIRTHDAY to my son, @DonaldJTrumpJr! Very proud of you! #TBT <a href="https://t.co/ULerCEOCGX">https://t.co/ULerCEOCGX</a> <a href="https://t.co/nbxPVdarJM">https://t.co/nbxPVdarJM</a>	2015-12-31 16:52:38
6.825788e+17	Twitter for Android	I would feel sorry for @JebBush and how badly he is doing with his campaign other than for the fact he took millions of \$'s of hit ads on me	2015-12-31 15:07:18
6.825446e+17	Twitter for iPhone	#MakeAmericaGreatAgain #Trump2016 <a href="https://t.co/IEIXos0wh9">https://t.co/IEIXos0wh9</a>	2015-12-31 12:51:35

## 2. Clean Those Tweets

### Step 03

#### Filter with Twitter for Android and iPhone only!

```
cleaned_tweets <- tweets %>%  
  select(id_str, source, text, created_at) %>%  
  filter(source %in% c("Twitter for  
    iPhone", "Twitter for Android"))
```

### Step 04

#### Extract "Android" and "iPhone" only!

```
cleaned_tweets <- tweets %>%  
  select(id_str, source, text, created_at) %>%  
  filter(source %in% c("Twitter for  
    iPhone", "Twitter for Android")) %>%  
  extract(source, "source", "(\\w+)$")
```

### Output [04]:

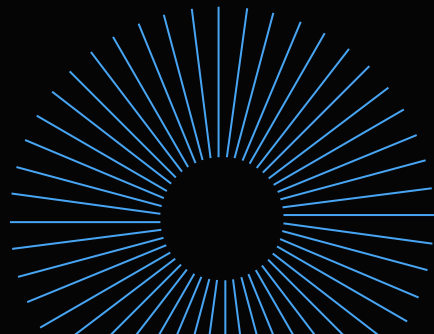
id_str	source	text	created_at
<dbl>	<chr>	<chr>	<dtm>
6.827032e+17	Android	I would like to wish everyone A HAPPY AND HEALTHY NEW YEAR. WE MUST ALL WORK TOGETHER TO, FINALLY, MAKE AMERICA SAFE AGAIN AND GREAT AGAIN!	2015-12-31 23:21:49
6.827007e+17	Android	Do you believe that The State Department, on NEW YEAR'S EVE, just released more of Hillary's e-mails. They just want it all to end. BAD!	2015-12-31 23:11:35
6.826351e+17	iPhone	THANK YOU ILLINOIS! Let's not forget to get family & friends-out to VOTE IN 2016! <a href="https://t.co/Ig5kMbNLYK">https://t.co/Ig5kMbNLYK</a> <a href="https://t.co/dtMaslq4cf">https://t.co/dtMaslq4cf</a>	2015-12-31 18:51:12
6.826053e+17	iPhone	HAPPY BIRTHDAY to my son, @DonaldJTrumpJr! Very proud of you! #TBT <a href="https://t.co/ULerCEOCGX">https://t.co/ULerCEOCGX</a> <a href="https://t.co/nbxPVdarJM">https://t.co/nbxPVdarJM</a>	2015-12-31 16:52:38
6.825788e+17	Android	I would feel sorry for @JebBush and how badly he is doing with his campaign other than for the fact he took millions of \$'s of hit ads on me	2015-12-31 15:07:18
6.825446e+17	iPhone	#MakeAmericaGreatAgain #Trump2016 <a href="https://t.co/IEIXos0wh9">https://t.co/IEIXos0wh9</a>	2015-12-31 12:51:35



## Task 3 | Is “time” the giveaway?

### Goals

To spot the difference of time between Android tweets and iPhone tweets.



### 3. Is “Time” The Giveaway?

#### Step 01

Load the packages!

```
library(ggplot2)
```

#### Step 02

Count the tweets by hour and device!

```
cleaned_tweets %>%  
  count(source, hour = hour(with_tz(created_at,  
    "EST")))
```

#### Step 03

Add a new column!

```
cleaned_tweets %>%  
  count(source, hour = hour(with_tz(created_at,  
    "EST"))) %>%  
  mutate(percent = n / sum(n))
```

Output [02]:

source	hour	n
<chr>	<int>	<int>
Android	0	104
Android	1	43
Android	2	40
Android	3	52
Android	4	104
Android	5	169



### 3. Is “Time” The Giveaway?

#### Step 01

##### Load the packages!

```
library(ggplot2)
```

#### Step 02

##### Count the tweets by hour and device!

```
cleaned_tweets %>%  
  count(source, hour = hour(with_tz(created_at,  
    "EST")))
```

#### Step 03

##### Add a new column!

```
cleaned_tweets %>%  
  count(source, hour = hour(with_tz(created_at,  
    "EST"))) %>%  
  mutate(percent = n / sum(n))
```

#### Output [03]:

source	hour	n	percent
<chr>	<int>	<int>	<dbl>
Android	0	104	0.015963162
Android	1	43	0.006600153
Android	2	40	0.006139678
Android	3	52	0.007981581
Android	4	104	0.015963162
Android	5	169	0.025940138

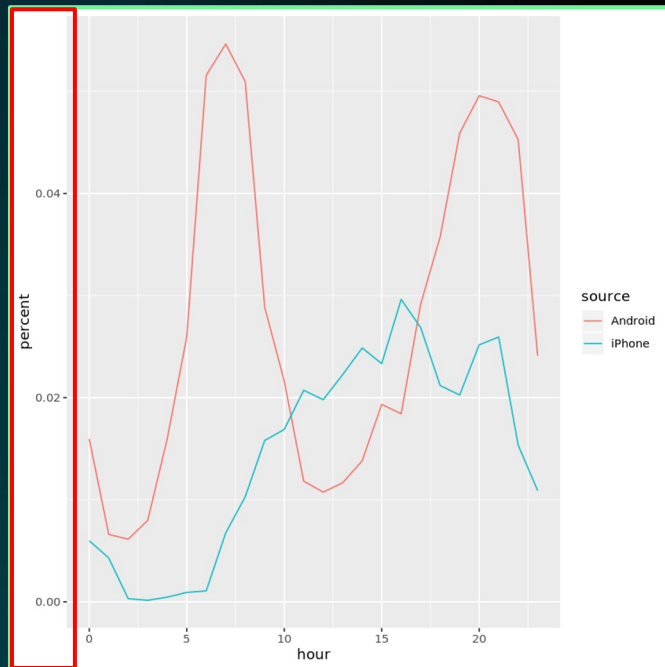
# 3. Is “Time” The Giveaway?

## Step 04

### Plot the percent of tweets by hour!

```
cleaned_tweets %>%  
  count(source, hour = hour(with_tz(created_at,  
    "EST"))) %>%  
  mutate(percent = n / sum(n)) %>%  
  ggplot(aes(hour, percent, color = source)) +  
  geom_line()
```

## Output [04]:



# 3. Is “Time” The Giveaway?

## Step 05

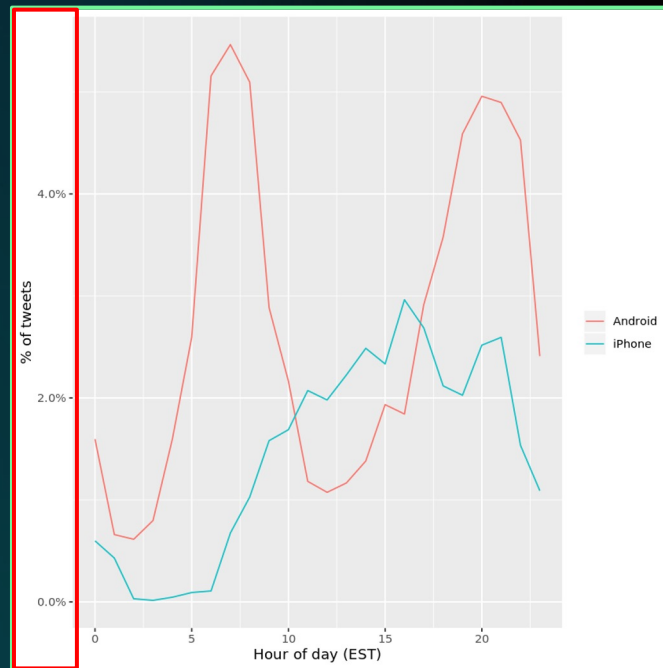
### Add some labels!

```
cleaned_tweets <-  
  count(source, hour = hour(with_tz(created_at,  
    "EST"))) %>%  
  mutate(percent = n / sum(n)) %>%  
  ggplot(aes(hour, percent, color = source)) +  
  geom_line() +  
  scale_y_continuous(labels =  
    scales::label_percent()) +  
  labs(x = "Hour of day (EST)",  
    y = "% of tweets", color = "")
```

### Interpretation

**Android tweets** are more often in the **early morning** or **later in the evening**. **iPhone tweets** are more often in the **afternoon**.

## Output [05]:

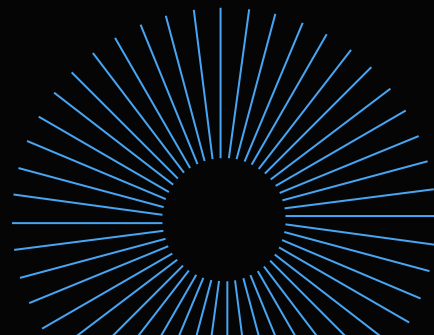




# Task 4 | The quote tweet is dead

## Goals

Create a bar plot of the number of tweets that are quoted and not quoted from each device.



# 4. The quote tweet is dead

## Step 01

### Load the stringr package

```
library(stringr)
```

## Step 02

### Count the tweets to determine the number of tweets quoted or not by each device

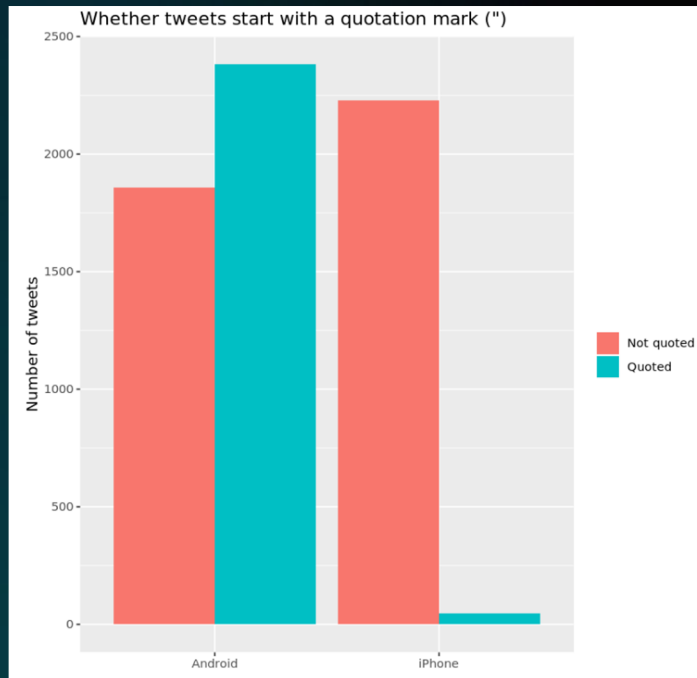
```
cleaned_tweets %>%  
  count(source,  
         quoted = ifelse(str_detect(text, '^\"'),  
                           "Quoted", "Not quoted")) %>%
```

## Step 03

### Plot the tweet by quoted or not by each device

```
ggplot(aes(source, n, fill = quoted)) +  
  geom_bar(stat = "identity", position = "dodge") +  
  labs(x = "", y = "Number of tweets", fill = "") +  
  ggtitle('Whether tweets start with a quotation mark ("')'
```

Output [04]:

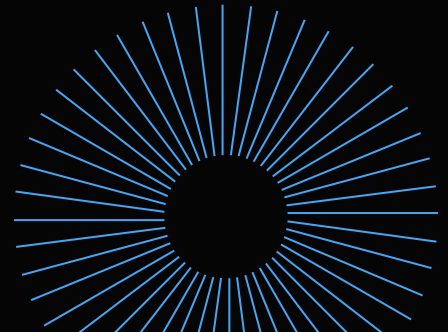




## Task 5 | Links and Pictures

### Goals

To create a bar plot of the number of tweets that do and do not have a picture/link from each device.



## 5. Links and Pictures

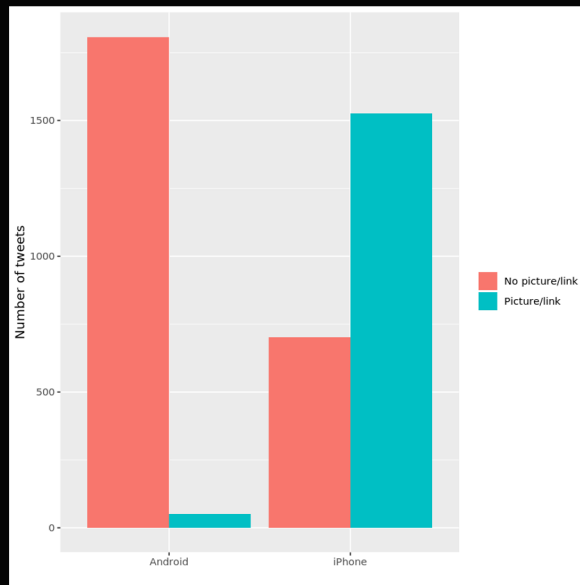
*# Count the number of tweets with and without picture/links by device*

```
tweet_picture_counts <- cleaned_tweets %>%  
  filter(!str_detect(text, '^')) %>%  
  count(source,  
         picture = ifelse(str_detect(text, "t.co"),  
                           "Picture/link", "No  
picture/link"))
```

*# Make a bar plot*

```
ggplot(tweet_picture_counts, aes(source, n, fill =  
  picture)) +  
  geom_bar(stat = "identity", position = "dodge")  
+  
  labs(x = "", y = "Number of tweets", fill = "")
```

Output [06]:

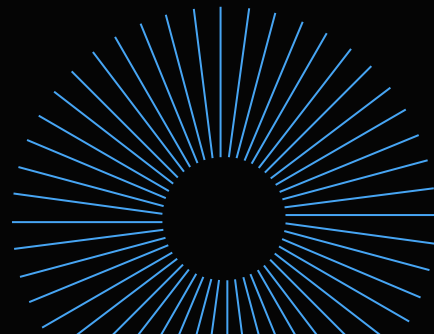




# Task 6 | Comparison of words

## Goals

Create a new data frame of words from all the tweets





# 6. Comparison of words

## Step 01

### Load the tidytext package

```
library(tidytext)
```

## Step 02

### Create a regex pattern to remove quote tweets

```
reg <- "([A-Za-z\\d#@']|'?![A-Za-z\\d#@'])"
```

## Output [06]:

A tibble: 6 x 4

id_str	source	created_at	word
<dbl>	<chr>	<dtm>	<chr>
6.053187e+17	Android	2015-06-01 10:23:13	@foxandfriends
6.053187e+17	Android	2015-06-01 10:23:13	enjoy
6.066705e+17	Android	2015-06-05 03:55:04	worst
6.066705e+17	Android	2015-06-05 03:55:04	boring
6.066705e+17	Android	2015-06-05 03:55:04	political
6.066705e+17	Android	2015-06-05 03:55:04	pundits

## Step 03

### Transform lines of text into words

```
tweet_words <- cleaned_tweets %>%  
  filter(!str_detect(text, '^\"')) %>%  
  mutate(text = str_replace_all(text,  
    "https://t.co/[A-Za-z\\d]+|&";, "")) %>%  
  unnest_tokens(word, text, token = "regex",  
    pattern = reg) %>%
```

## Step 04

### Remove any stopwords

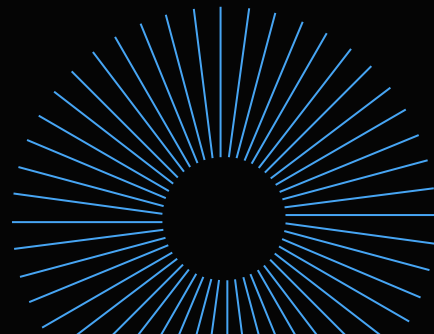
```
filter(!word %in% stop_words$word,  
  str_detect(word, "[a-z]"))
```



# Task 7 | Most common words

## Goals

Plot the most common words



# 7. Most common words

## Step 01

Count the most common words and sort them

```
tweet_words %>%  
  count(word, sort = TRUE) %>%
```

## Step 02

Take first 20 words and reorder according to number of occurrences

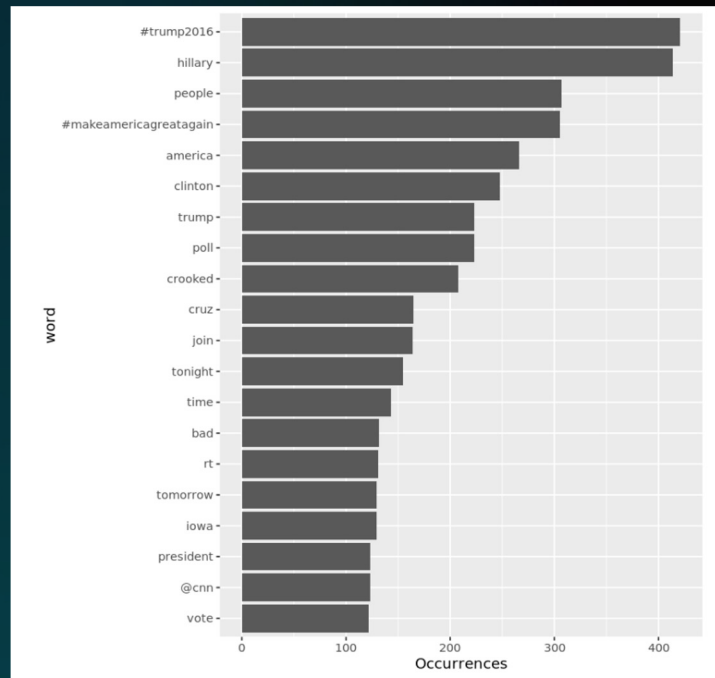
```
head(20) %>%  
mutate(word = reorder(word, n)) %>%
```

## Step 03

Plot the most common words

```
ggplot(aes(word, n)) +  
  geom_bar(stat = "identity") +  
  ylab("Occurrences") +  
  coord_flip()
```

Output [07]:

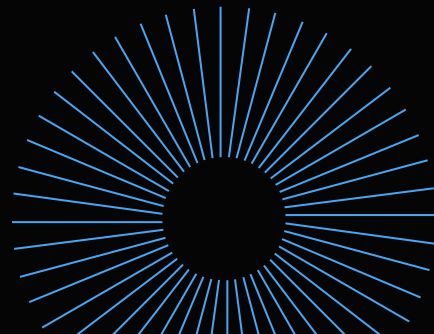




## Task 8 | Common words: Android v. iPhone (i)

### Goals

To find which words are most common from the Android relative to the iPhone, and vice versa.



## 8. Common Words: Android v. iPhone (i)

### Step 01

**Count the occurrences of each word!**

```
android_iphone_ratios <- tweet_words %>%  
  count(word, source)
```

### Step 02

**Filter for words that occurs at least 5 times!**

```
android_iphone_ratios <- tweet_words %>%  
  count(word, source) %>%  
  group_by(word) %>%  
  filter(sum(n) >= 5)
```

### Output [01]:

word	source	n
<chr>	<chr>	<int>
#'s	iPhone	4
#1for	Android	1
#2a	iPhone	5
#6days	iPhone	1
#abc2020	iPhone	2
#alconvention2016	iPhone	1

## 8. Common Words: Android v. iPhone (i)

### Step 01

**Count the occurrences of each word!**

```
android_iphone_ratios <- tweet_words %>%  
  count(word, source)
```

### Step 02

**Filter for words that occurs at least 5 times!**

```
android_iphone_ratios <- tweet_words %>%  
  count(word, source) %>%  
  group_by(word) %>%  
  filter(sum(n) >= 5)
```

### Output [02]:

word	source	n
<chr>	<chr>	<int>
#2a	iPhone	5
#americafirst	iPhone	71
#bigleaguetruth	iPhone	24
#caucusfortrump	iPhone	10
#crookedhillary	iPhone	30
#debate	iPhone	7

## 8. Common Words: Android v. iPhone (i)

### Step 03

**Turn source values into column!**

```
android_iphone_ratios <- tweet_words %>%  
  count(word, source) %>%  
  group_by(word) %>%  
  filter(sum(n) >= 5) %>%  
  spread(source, n, fill = 0) %>%  
  ungroup()
```

### Step 04

**Create odds ratio for Android and iPhone!**

```
android_iphone_ratios <- tweet_words %>%  
  (...) %>%  
  spread(source, n, fill = 0) %>%  
  ungroup() %>%  
  mutate_if(is.numeric, ~((. + 1) / sum(. + 1)))
```

### Output [03]:

word	Android	iPhone
<chr>	<dbl>	<dbl>
#2a	0	5
#americafirst	0	71
#bigleaguetruth	0	24
#caucusfortrump	0	10
#crookedhillary	0	30
#debate	0	7

## 8. Common Words: Android v. iPhone (i)

### Step 03

Turn source values into column!

```
android_iphone_ratios <- tweet_words %>%  
  count(word, source) %>%  
  group_by(word) %>%  
  filter(sum(n) >= 5) %>%  
  spread(source, n, fill = 0) %>%  
  ungroup()
```

### Step 04

Create odds ratio for Android and iPhone!

```
android_iphone_ratios <- tweet_words %>%  
  (...) %>%  
  spread(source, n, fill = 0) %>%  
  ungroup() %>%  
  mutate_if(is.numeric, ~((. + 1) / sum(. + 1)))
```

### Output [04]:

word	Android	iPhone
<chr>	<dbl>	<dbl>
#2a	7.259001e-05	0.0004790802
#americafirst	7.259001e-05	0.0057489620
#bigleaguetruth	7.259001e-05	0.0019961674
#caucusfortrump	7.259001e-05	0.0008783136
#crookedhillary	7.259001e-05	0.0024752475
#debate	7.259001e-05	0.0006387736



## 8. Common Words: Android v. iPhone (i)

### Step 05

Create log ratio: Android divided by iPhone!

```
android_iphone_ratios <- tweet_words %>%  
  (...) %>%  
  ungroup() %>%  
  mutate_if(is.numeric,  
            ~((. + 1)/sum(. + 1))) %>%  
  mutate(logratio = log2(Android / iPhone))
```

### Step 06

Arrange the data!

```
android_iphone_ratios <- tweet_words %>%  
  (...) %>%  
  mutate_if(is.numeric,  
            ~((. + 1)/sum(. + 1))) %>%  
  mutate(logratio = log2(Android / iPhone)) %>%  
  arrange(desc(logratio))
```

### Output [06]:

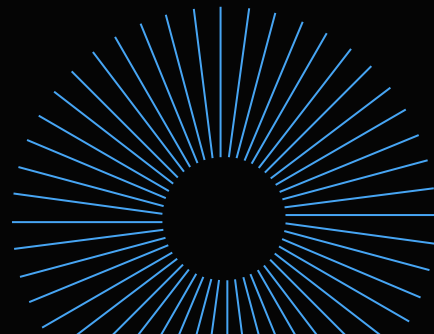
word	Android	iPhone	logratio
<chr>	<dbl>	<dbl>	<dbl>
mails	0.0016695703	7.984669e-05	4.386100
poor	0.0010162602	7.984669e-05	3.669893
poorly	0.0009436702	7.984669e-05	3.562978
bosses	0.0008710801	7.984669e-05	3.447501
turnberry	0.0008710801	7.984669e-05	3.447501
angry	0.0007984901	7.984669e-05	3.321970



## Task 9 | Common words: Android v. iPhone (ii)

### Goals

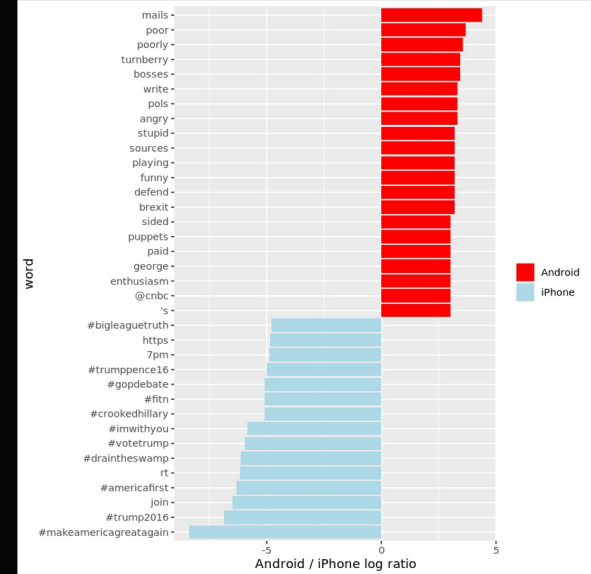
To find which words are most common from the Android relative to the iPhone, and vice versa.



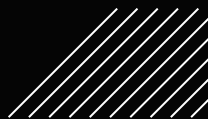
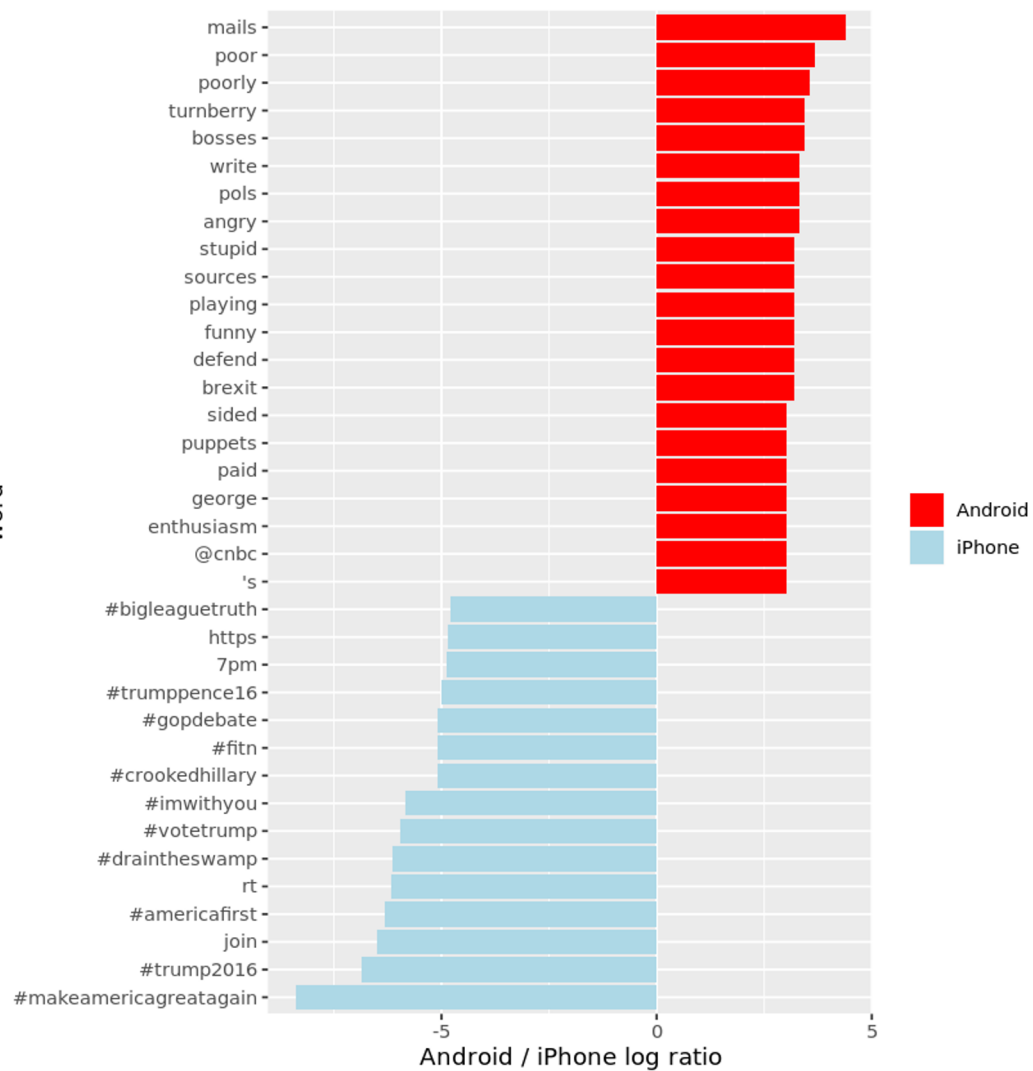
## 9. Common Words: Android v. iPhone (ii)

*# Plot the log odds ratio for each word by device*

```
android_iphone_ratios %>%  
  group_by(logratio > 0) %>%  
  top_n(15, abs(logratio)) %>%  
  ungroup() %>%  
  mutate(word = reorder(word, logratio)) %>%  
  ggplot(aes(word, logratio, fill = logratio > 0)) +  
  geom_bar(stat = "identity") +  
  coord_flip() +  
  ylab("Android / iPhone log ratio") +  
  scale_fill_manual(name = "", labels = c("Android",  
    "iPhone"),  
    values = c("red", "lightblue"))
```



word

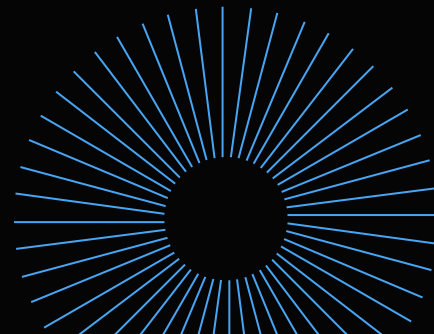




# Task 10 | Adding sentiments

## Goals

To spot the difference in sentiment of each word.



# 10. Adding Sentiments

## Step 01

Create a sentiment data frame!

```
nrc <- read_rds("datasets/nrc.rds")
```

## Step 02

Join nrc with android\_iphone\_ratios!

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  inner_join(nrc, by = "word")
```

## Step 03

Remove "positive" and "negative" sentiment!

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  inner_join(nrc, by = "word")  
  filter(!sentiment %in% c("positive", "negative"))
```

Output [01]:

word	sentiment
<chr>	<chr>
abacus	trust
abandon	fear
abandon	negative
abandon	sadness
abandoned	anger
abandoned	fear

# 10. Adding Sentiments

## Step 01

Create a sentiment data frame!

```
nrc <- read_rds("datasets/nrc.rds")
```

## Step 02

Join nrc with android\_iphone\_ratios!

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  inner_join(nrc, by = "word")
```

## Step 03

Remove "positive" and "negative" sentiment!

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  inner_join(nrc, by = "word")  
  filter(!sentiment %in% c("positive", "negative"))
```

## Output [02]:

word	Android	iPhone	logratio	sentiment
<chr>	<dbl>	<dbl>	<dbl>	<chr>
poorly	0.0009436702	7.984669e-05	3.562978	negative
angry	0.0007984901	7.984669e-05	3.321970	anger
angry	0.0007984901	7.984669e-05	3.321970	disgust
angry	0.0007984901	7.984669e-05	3.321970	negative
defend	0.0007259001	7.984669e-05	3.184466	fear
defend	0.0007259001	7.984669e-05	3.184466	positive

# 10. Adding Sentiments

## Step 01

Create a sentiment data frame!

```
nrc <- read_rds("datasets/nrc.rds")
```

## Step 02

Join nrc with android\_iphone\_ratios!

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  inner_join(nrc, by = "word")
```

## Step 03

Remove "positive" and "negative" sentiment!

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  inner_join(nrc, by = "word")  
  filter(!sentiment %in% c("positive", "negative"))
```

Output [03]:

word	Android	iPhone	logratio	sentiment
<chr>	<dbl>	<dbl>	<dbl>	<chr>
angry	0.0007984901	7.984669e-05	3.321970	anger
angry	0.0007984901	7.984669e-05	3.321970	disgust
defend	0.0007259001	7.984669e-05	3.184466	fear
enthusiasm	0.0006533101	7.984669e-05	3.032463	anticipation
enthusiasm	0.0006533101	7.984669e-05	3.032463	joy
enthusiasm	0.0006533101	7.984669e-05	3.032463	surprise



# 10. Adding Sentiments

## Step 04

**Reorder the values of sentiment and word!**

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  (...) %>%  
  mutate(sentiment = reorder(sentiment, -logratio),  
         word = reorder(word, -logratio))
```

## Step 05

**Take the top 10 values of each sentiment group!**

```
android_iphone_sentiment <- android_iphone_ratios %>%  
  (...) %>%  
  group_by(sentiment) %>%  
  top_n(10, abs(logratio)) %>%  
  ungroup()
```

## Output [04]:

word	Android	iPhone	logratio	sentiment
<fct>	<dbl>	<dbl>	<dbl>	<fct>
angry	0.0007984901	7.984669e-05	3.321970	anger
angry	0.0007984901	7.984669e-05	3.321970	disgust
defend	0.0007259001	7.984669e-05	3.184466	fear
enthusiasm	0.0006533101	7.984669e-05	3.032463	anticipation
enthusiasm	0.0006533101	7.984669e-05	3.032463	joy
enthusiasm	0.0006533101	7.984669e-05	3.032463	surprise

# 10. Adding Sentiments

## Step 04

**Reorder the values of sentiment and word!**

```
android_iphone_sentiment <- android_iphone_ratios
%>%
  (...) %>%
  mutate(sentiment = reorder(sentiment, -logratio),
         word = reorder(word, -logratio))
```

## Step 05

**Take the top 10 values of each sentiment group!**

```
android_iphone_sentiment <- android_iphone_ratios %>%
  (...) %>%
  group_by(sentiment) %>%
  top_n(10, abs(logratio)) %>%
  ungroup()
```

## Output [05]:

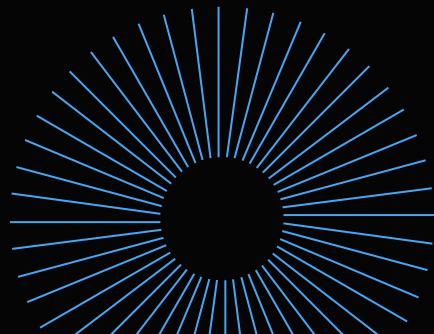
word	Android	iPhone	logratio	sentiment
<fct>	<dbl>	<dbl>	<dbl>	<fct>
angry	0.0007984901	7.984669e-05	3.321970	anger
angry	0.0007984901	7.984669e-05	3.321970	disgust
defend	0.0007259001	7.984669e-05	3.184466	fear
enthusiasm	0.0006533101	7.984669e-05	3.032463	anticipation
enthusiasm	0.0006533101	7.984669e-05	3.032463	joy
enthusiasm	0.0006533101	7.984669e-05	3.032463	surprise
badly	0.0025406504	3.193868e-04	2.991821	sadness
crazy	0.0016695703	2.395401e-04	2.801138	anger
crazy	0.0016695703	2.395401e-04	2.801138	fear
crazy	0.0016695703	2.395401e-04	2.801138	sadness



# Task 11 | Android vs. iPhone sentiments

## Goals

Plot the log odds ratio of each word from both devices by sentiment



# 11. Android vs. iPhone sentiments

## Step 01

### Plot the data frame

```
ggplot(android_iphone_sentiment, aes(word,
logratio, fill = logratio < 0)) +
```

## Step 02

### Facet the data by sentiment and create two rows

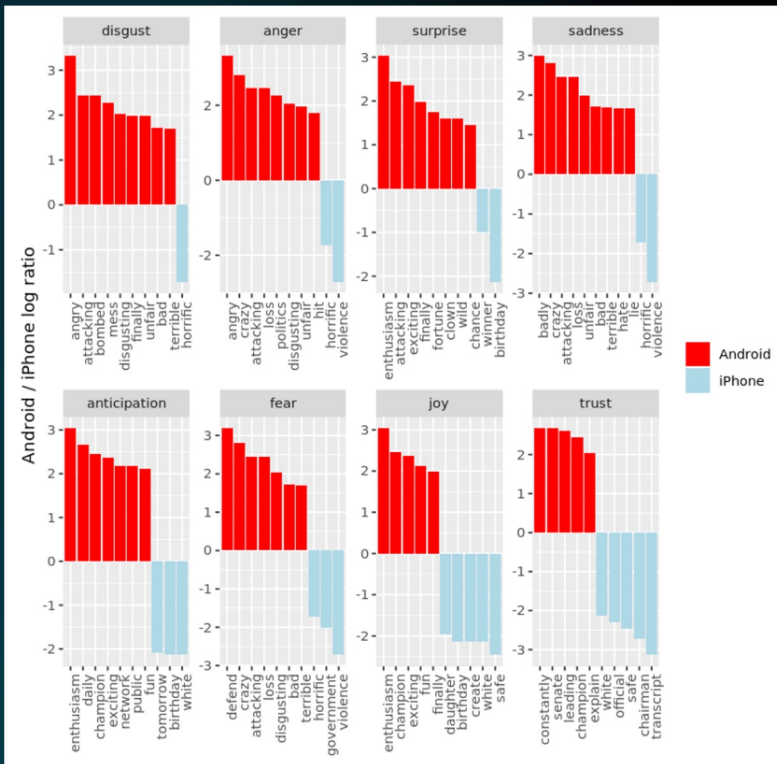
```
facet_wrap(~ sentiment, scales = "free", nrow =
2) +
```

## Step 03

### Map the data frame using geom\_bar

```
geom_bar(stat = "identity") + theme(axis.text.x =
element_text(angle = 90, hjust = 1)) + labs(x =
"", y = "Android / iPhone log ratio") +
scale_fill_manual(name = "", labels =
c("Android", "iPhone"), values = c("red",
"lightblue"))
```

## Output [11]:





# 04



## Discussion

---

Result Discussion and Our Innovative Idea



# Conclusion: The ghost in the political machine

There's a difference in style and sentiment between Trump's tweets from the Android and the iPhone. We know Trump used the Android until March 2017, but who was tweeting from the iPhone on Trump's behalf? I was fascinated by a New Yorker article about Tony Schwartz, Trump's ghostwriter for The Art of the Deal. Of particular interest was how Schwartz imitated Trump's voice and philosophy:

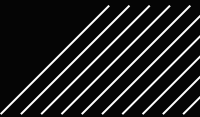
In his journal, Schwartz describes the process of trying to make Trump's voice palatable in the book. It was kind of "a trick," he writes, to mimic Trump's blunt, staccato, no-apologies delivery while making him seem almost boyishly appealing.... Looking back at the text now, Schwartz says, "I created a character far more winning than Trump is."

A lot has been written about Trump's mental state. But I'd rather get inside the head of the anonymous staffer whose job is to imitate Trump's unique cadence ("Very sad!") or put a positive spin on it, to millions of his followers. Are they a true believer, or just a cog in a political machine, mixing whatever mainstream appeal they can into the @realDonaldTrump concoction? Like Tony Schwartz, will they one day regret their involvement?

# WHAT DO YOU THINK?

TRUE BELIEVER

COG



# Group Idea: Misinformation Surrounding Monkeypox

Monkeypox travelled to Canada. The unvaccinated can't travel to Canada. 🤔

WHAT IF monkeypox is the cover for symptoms of vaccine induced autoimmune conditions....?

**FALSE** May 21, 2022 · Twitter for Android

Who is surprised that after millions of people have been injected with genetically modified chimp virus, there is now an outbreak of monkeypox?

12:33 PM · May 20, 2022 · Twitter Web App

**MISLEADING** 6,352 Likes



Shingles is the monkey pox because it's all a hoax, it's from the side effects from the Pfizer vaccines

11:49 PM · May 22, 2022 · Twitter for iPhone



# Group Idea: Misinformation Surrounding Monkeypox

## Basic Concept

Analyze misinformation (hoax) regarding Monkeypox in Twitter using R.

## Methods

Using text mining to compare every single words in hoax tweets with real information regarding Monkeypox.

# THANK YOU!

Do you have any questions?

