Twitter Sentiment Analysis

Install Packages

also installing the dependencies ‘memoise’, ‘whisker’, ‘rstudioapi’, ‘git2r’, ‘withr’

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/memoise_1.1.0.tgz'
Content type 'application/x-gzip' length 27154 bytes (26 KB)
==================================================
downloaded 26 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/whisker_0.3-2.tgz'
Content type 'application/x-gzip' length 48077 bytes (46 KB)
==================================================
downloaded 46 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/rstudioapi_0.7.tgz'
Content type 'application/x-gzip' length 110343 bytes (107 KB)
==================================================
downloaded 107 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/git2r_0.23.0.tgz'
Content type 'application/x-gzip' length 194352 bytes (1.9 MB)
==================================================
downloaded 1.9 MB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/withr_2.1.2.tgz'
Content type 'application/x-gzip' length 122230 bytes (119 KB)
==================================================
downloaded 119 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/devtools_1.13.6.tgz'
Content type 'application/x-gzip' length 43876 bytes (428 KB)
==================================================
downloaded 428 KB

The downloaded binary packages are in
/var/folders/mh/gtsn8xhj12db0wxrnxnlm9f4000gn/T//RtmpMsDuGo/downloaded_packages
also installing the dependencies 'bit', 'bit64', 'rjson', 'DBI'

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/bit_1.1-14.tgz'
Content type 'application/x-gzip' length 248803 bytes (242 KB)
================================================================================
downloaded 242 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/bit64_0.9-7.tgz'
Content type 'application/x-gzip' length 534016 bytes (521 KB)
================================================================================
downloaded 521 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/rjson_0.2.20.tgz'
Content type 'application/x-gzip' length 176951 bytes (172 KB)
================================================================================
downloaded 172 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/DBI_1.0.0.tgz'
Content type 'application/x-gzip' length 829862 bytes (810 KB)
================================================================================
downloaded 810 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/twitteR_1.1.9.tgz'
Content type 'application/x-gzip' length 466439 bytes (455 KB)
================================================================================
downloaded 455 KB

The downloaded binary packages are in
/var/folders/mh/gtsn8xhjl2db0wxrnxq1m9f40000gn/T//RtmpMsDuGo/downloaded_packages

Load Packages

library(httr)

Attaching package: ‘httr’

The following object is masked from ‘package:NLP’:

content
library(devtools)

package ‘devtools’ was built under R version 3.4.4

library(twitteR)

Collect Tweets

# Get Tweet API credentials following https://www.slickremix.com/docs/how-to-get-api-keys-and-tokens-for-twitter/
consumer_key <- "puxmqW5Fb9zV3wddFzwGYmcCj"
consumer_secret <- "rtP4AYjMxvaKlzI2a9JNB5P26cnBHYJQb7H35cpcHuQC0tEzYV"
access_token <- "2607567668-XhrG0Ywzhh13FgcSgzh6hA1qz1KrG0w4xJNxA9Q3"
access_secret <- "xOhkVZQrIwJnxR865B51rgNq1tgLb7WDVRGRmoIiMFbP"

# Set Tweet API credentials
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

[1] "Using direct authentication"
Use a local file (".httr-oauth"), to cache OAuth access credentials between R sessions?

1: Yes
2: No

1

Adding .httr-oauth to .gitignore
# Collect Tweets

tw1 <- searchTwitter("artificial intelligence", n=100, lang='en', since = "2019-07-06", until = "2019-07-07")
tw2 <- searchTwitter("artificial intelligence", n=100, lang='en', since = "2019-07-07", until = "2019-07-08")
tw3 <- searchTwitter("artificial intelligence", n=100, lang='en', since = "2019-07-08", until = "2019-07-09")

# Convert the returned tweets in list to dataframe

df1 <- twListToDF(tw1)
df2 <- twListToDF(tw2)
df3 <- twListToDF(tw3)

# Combine tweets into one dataframe

df <- rbind(df1, df2, df3)

df

<table>
<thead>
<tr>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT @Datascience__: World Artificial Intelligence (AI) Market to Reach $71 Billion by 2024 - Majorly Driven by the Growing Adoption of Cloud...</td>
</tr>
<tr>
<td>World Artificial Intelligence (AI) Market to Reach $71 Billion by 2024 - Majorly Driven by the Growing Adoption of Cloud... <a href="https://t.co/BgEP8KpnJp">https://t.co/BgEP8KpnJp</a></td>
</tr>
<tr>
<td>Dentistry tech giant Oral-B originally announced its super smart toothbrush, the Genius X, at CES earlier this year... <a href="https://t.co/sfOJoO98eK">https://t.co/sfOJoO98eK</a></td>
</tr>
<tr>
<td>RT @MikeMacRaeMike: The last thing I see before I die, as I stare vacantly out my nursing home window, is an angry robot marching around ho...</td>
</tr>
<tr>
<td>RT @ChrisRabane: @dr_wardsam About the digital revolution, what you see now is just the tip of the iceberg. Deep learning is a phase. The g...</td>
</tr>
<tr>
<td>RT @RoemerAndres: Ethics in the Age of Artificial Intelligence - Scientific American Blog Network <a href="https://t.co/qV5iyDX1lc">https://t.co/qV5iyDX1lc</a></td>
</tr>
</tbody>
</table>
| RT @mitsmr: These 4 soft skills become more valuable as human-machine collaboration advances
1. Complexity
2. Creativity
3. Social a... |
# Subset only two columns with tweet text and created date

def <- subset(df, select=c(text, created))

<table>
<thead>
<tr>
<th>1-10 of 300 rows</th>
<th>1-1 of 2 columns</th>
<th>Previous</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>...</th>
<th>30</th>
<th>Next</th>
</tr>
</thead>
</table>

1
2
3
4
5
6
7
8
9
10

1-10 of 300 rows | 1-1 of 2 columns | Previous | 1 | 2 | 3 | 4 | 5 | 6 | ... | 30 | Next |
|------------------|------------------|----------|---|---|---|---|---|---|-----|----|------|

Redefining Recruitment Through Artificial Intelligence & Machine Learning - Experion Tec... https://t.co/EWHnz9oQoq

RT @MeterQubes: How we plan to use #AI and #deeplearning, to provide a better #decentralized environment for #traders. #Blo...

RT @SenJoniErnst: The #FY20NDAA invests in research & development for key technologies like artificial intelligence, directed energy, & hyp...
# Format tweet created date

df$created <- strftime(df$created, '%Y-%m-%d')

# save the returned tweets to local csv files
write.csv(df, file = "tweets_sample.csv", row.names=FALSE)

Generate Word Cloud

install.packages(c("tm", "SnowballC", "wordcloud", "RColorBrewer", "RCurl", "XML"))

also installing the dependencies 'NLP’, ‘slam’, ‘xml2’
There is a binary version available but the source version is later:

<table>
<thead>
<tr>
<th>binary</th>
<th>source needs_compilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML 3.98-1.19</td>
<td>3.98-1.20</td>
</tr>
</tbody>
</table>

Do you want to install from sources the package which needs compilation?

y
installing the source package ‘XML’

trying URL 'https://cran.rstudio.com/src/contrib/XML_3.98-1.20.tar.gz'
Content type 'application/x-gzip' length 1600829 bytes (1.5 MB)
==================================================
downloaded 1.5 MB

* installing *source* package ‘XML’ ...
** package ‘XML’ successfully unpacked and MD5 sums checked

checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for sed... /usr/bin/sed
checking for pkg-config... no
checking for xml2-config... /usr/bin/xml2-config
USE_XML2 = yes
SED_EXTENDED_ARG: -E
Minor 9, Patch 4 for 2.9.4
Located parser file -I/usr/include/libxml2/parser.h
Checking for 1.8: -I/usr/include/libxml2
Using libxml2.*
checking for gzopen in -lz... yes
checking for xmlParseFile in -lxml2... yes
checking for xmlHashSize in -lxml2... yes
Using built-in xmlHashSize
Checking DTD parsing (presence of externalSubset)...
checking for xmlHashSize in -lxml2... yes
Found xmlHashSize
checking for xmlDocDumpFormatMemoryEnc in -lxml2... yes
have xmlDocDumpFormatMemoryEnc()
checking for xmlOutputBufferCreateBuffer in -lxml2... yes
have xmlOutputBufferCreateBuffer()
checking for xmlDocDumpFormatMemoryEnc in -lxml2... yes
checking libxml/xmlversion.h usability... yes
checking libxml/xmlversion.h presence... yes
checking for libxml/xmlversion.h... yes
Expat: FALSE
Checking for return type of xmlHashScan element routine.
No return value for xmlHashScan
xmlNs has a context field
Checking for cetype_t enumeration
Using recent version of R with cetype_t enumeration type for encoding
checking for xmlsec1-config... no
nodegc default
xml-debug default
Version has XML_WITH_ZLIB
Version has xmlHasFeature()

Configuration information:

Libxml settings

libxml include directory: -I/usr/include/libxml2
libxml library directory: -lxm12 -lz -lpthread -licucore -lm -lz -lxm12
libxml 2:

Compilation flags: -DLIBXML2=1 -I/usr/include/libxml2 -DLIBXML -DUSE_EXTERNAL_SUBSET=1 -DROOT_HASDTD_NODE=1 -DDUMP_WITH_ENCODING=1 -DUSE_XML_VERSION_H=1 -DXML_ELEMENT_ETYPE=1 -DXML_ATTRIBUTE_ATYPE=1 -DNO_XML_HASH_SCANNER_RETURN=1 -DLIBXML_NAMESPACE_HAS_CONTEXT=1 -DHAVE_R_CETYPE_T=1 -DHAVE XML_WITH_ZLIB=1 -DHAVEXML_HAS_FEATURE=1 -DUSE_R=1 -D_R_=1 -DHAVE_VALIDITY=1 -DXML_REF_COUNT_NODES=1
Link flags: -lxm12 -lz -lpthread -licucore -lm -lz -lxm12

configure: creating ./config.status
config.status: creating src/Makevars
config.status: creating R/supports.R
config.status: creating inst/scripts/RSXML.csh
config.status: creating inst/scripts/RSXML.bsh

** libs

clang -I/Library/Frameworks/R.framework/Resources/include -DNDEBUG -DLIBXML -I/usr/include/libxml2 -DUSE_EXTERNAL_SUBSET=1 -DROOT_HASDTD_NODE=1 -DDUMP_WITH_ENCODING=1 -DUSE_XML_VERSION_H=1 -DXML_ELEMENT_ETYPE=1 -DXML_ATTRIBUTE_ATYPE=1 -DNO_XML_HASH_SCANNER_RETURN=1 -DLIBXML_NAMESPACE_HAS_CONTEXT=1 -DHAVE_R_CETYPE_T=1 -DHAVEXML_WITH_ZLIB=1 -DHAVEXML_HAS_FEATURE=1 -DUSE_R=1 -D_R_=1 -DHAVE_VALIDITY=1 -DXML_REF_COUNT_NODES=1
clang -I/Library/Frameworks/R.framework/Resources/include -DNDEBUG -DLIBXML -I/usr/include/libxml2 -DUSE_EXTERNAL_SUBSET=1 -DROOT_HASDTD_NODE=1 -DDUMP_WITH_ENCODING=1 -DUSE_XML_VERSION_H=1 -DXML_ELEMENT_ETYPE=1 -DXML_ATTRIBUTE_ATYPE=1 -DNO_XML_HASH_SCANNER_RETURN=1 -DLIBXML_NAMESPACE_HAS_CONTEXT=1 -DHAVE_R_CETYPE_T=1 -DHAVEXML_WITH_ZLIB=1 -DHAVEXML_HAS_FEATURE=1 -DUSE_R=1 -D_R_=1 -DHAVE_VALIDITY=1 -DXML_REF_COUNT_NODES=1
clang -I/Library/Frameworks/R.framework/Resources/include -DNDEBUG -DLIBXML -I/usr/include/libxml2 -DUSE_EXTERNAL_SUBSET=1 -DROOT_HASDTD_NODE=1 -DDUMP_WITH_ENCODING=1 -DUSE_XML_VERSION_H=1 -DXML_ELEMENT_ETYPE=1 -DXML_ATTRIBUTE_ATYPE=1 -DNO_XML_HASH_SCANNER_RETURN=1 -DLIBXML_NAMESPACE_HAS_CONTEXT=1 -DHAVE_R_CETYPE_T=1 -DHAVEXML_WITH_ZLIB=1 -DHAVEXML_HAS_FEATURE=1 -DUSE_R=1 -D_R_=1 -DHAVE_VALIDITY=1 -DXML_REF_COUNT_NODES=1
schema.c:207:9: warning: unused variable 'nprot' [-Wunused-variable]
   int nprot = 0;
^ 1 warning generated.
xmlsecurity.c:34:12: warning: unused variable 'foo' [-Wunused-variable]
static int foo;
    ^
1 warning generated.

installing to /Library/Frameworks/R.framework/Versions/3.4/Resources/library/XML/libs
** R
** inst
** preparing package for lazy loading
Creating a generic function for ‘source’ from package ‘base’ in package ‘XML’
in method for ‘xmlAttrsToDataFrame’ with signature ‘"AsIs"’: no definition for class
“AsIs”
in method for ‘readKeyValueDB’ with signature ‘"AsIs"’: no definition for class “AsIs”
in method for ‘readSolrDoc’ with signature ‘"AsIs"’: no definition for class “AsIs”
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (XML)

The downloaded source packages are in

‘/private/var/folders/mh/gtsn8xhjl2db0wxrnxqlm9f40000gn/T/RtmpmwwvRF1/downloaded_packages’

### rquery.wordcloud() : Word cloud generator
### - http://www.sthda.com

x : character string (plain text, web url, txt file path)
type : specify whether x is a plain text, a web page url or a file path
lang : the language of the text
excludeWords : a vector of words to exclude from the text
textStemming : reduces words to their root form
colorPalette : the name of color palette taken from RColorBrewer package,
    or a color name, or a color code
min.freq : words with frequency below min.freq will not be plotted
max.words : Maximum number of words to be plotted. least frequent terms dropped
value returned by the function : a list(tdm, freqTable)

rquery.wordcloud <- function(x, type=c("text", "url", "file"),
                           lang="english", excludeWords=NULL,
                           textStemming=FALSE, colorPalette="Dark2",
                           min.freq=3, max.words=200)
{
    library("tm")
    library("SnowballC")
    library("wordcloud")
    library("RColorBrewer")

    if(type[1]=="file") text <- readLines(x)
else if (type[1] == "url") text <- html_to_text(x)
else if (type[1] == "text") text <- x

# Load the text as a corpus
docs <- Corpus(VectorSource(text))
# Convert the text to lower case
docs <- tm_map(docs, content_transformer(tolower))
# Remove numbers
docs <- tm_map(docs, removeNumbers)
# Remove stopwords for the language
docs <- tm_map(docs, removeWords, stopwords(lang))
# Remove punctuations
docs <- tm_map(docs, removePunctuation)
# Eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)
# Remove your own stopwords
if (!is.null(excludeWords))
  docs <- tm_map(docs, removeWords, excludeWords)
# Text stemming
if (textStemming) docs <- tm_map(docs, stemDocument)
# Create term-document matrix
tdm <- TermDocumentMatrix(docs)
m <- as.matrix(tdm)
v <- sort(rowSums(m), decreasing=TRUE)
d <- data.frame(word = names(v), freq=v)
# check the color palette name
if (!colorPalette %in% rownames(brewer.pal.info)) colors = colorPalette
else colors = brewer.pal(8, colorPalette)
# Plot the word cloud
set.seed(1234)
wordcloud(d$word, d$freq, min.freq=min.freq, max.words=max.words,
         random.order=FALSE, rot.per=0, scale = c(3, 1),
         use.r.layout=FALSE, colors=colors)

invisible(list(tdm=tdm, freqTable = d))

#++++++++++++++++++++++
# Helper function
#++++++++++++++++++++++
# Download and parse webpage
# Download and parse webpage
html_to_text <- function(url){
  library(RCurl)
  library(XML)
  # download html
  html.doc <- getURL(url)
  # convert to plain text
  doc = htmlParse(html.doc, asText=TRUE)
  # "//text()" returns all text outside of HTML tags.
  # We also don't want text such as style and script codes
  text <- xpathSApply(doc, "//text() [not(ancestor::script)] [not(ancestor::style)] [not
# Format text vector into one character string
return(paste(text, collapse = " "))

# read tweets
def <- read.csv("tweets_sample.csv")

# generate word cloud
rquery.wordcloud(x=df$text, type="text", lang="english", excludeWords = NULL, textStemming = FALSE, colorPalette="Dark2", min.freq=10, max.words=200)
Sentiment Classification

#reference: https://github.com/okugami79/sentiment140
install_github('sentiment140', 'okugami79')

library(sentiment)
# read tweets
df <- read.csv("tweets_sample.csv")

# Perform sentiment classification on tweets
sentiment_df <- sentiment(df$text)

sentiment_df

<table>
<thead>
<tr>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT @Datascience__: World Artificial Intelligence (AI) Market to Reach $71 Billion by 2024 - Majorly Driven by Cloud<img src="https://t.co/BgEP8KpnJp" alt="" /></td>
</tr>
<tr>
<td>World Artificial Intelligence (AI) Market to Reach $71 Billion by 2024 - Majorly Driven by the Growing Adoption of Cloud<img src="https://t.co/BgEP8KpnJp" alt="" /></td>
</tr>
<tr>
<td>Dentistry tech giant Oral-B originally announced its super smart toothbrush, the Genius X, at CES earlier this year<img src="https://t.co/sfOJoO98eK" alt="" /></td>
</tr>
<tr>
<td>RT @MikeMacRaeMike: The last thing I see before I die, as I stare vacantly out my nursing home window, is a angry robot marching around<img src="https://t.co/sfOJoO98eK" alt="" /></td>
</tr>
<tr>
<td>RT @ChrisRabane: @dr_wardsam About the digital revolution, what you see now is just the tip of the iceberg phase. The g<img src="https://t.co/sfOJoO98eK" alt="" /></td>
</tr>
<tr>
<td>RT @RoemerAndres: Ethics in the Age of Artificial Intelligence - Scientific American Blog Network<img src="https://t.co/qV5iyDX1lc" alt="" /></td>
</tr>
<tr>
<td>RT @mitsmr: These 4 soft skills become more valuable as human-machine collaboration advances1. Complex reasoning2. Creativity3. Social a<img src="https://t.co/qV5iyDX1lc" alt="" /></td>
</tr>
</tbody>
</table>
# rename the column polarity to sentiment
df$sentiment <- sentiment_df$polarity
# change the data type to factor
df$created <- as.factor(df$created)
df$sentiment <- as.factor(df$sentiment)

# count frequency in each group
library(plyr)
group_count <- count(df, c('created', 'sentiment'))
group_count

<table>
<thead>
<tr>
<th>created</th>
<th>sentiment</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-07-06</td>
<td>negative</td>
<td>1</td>
</tr>
<tr>
<td>2019-07-06</td>
<td>neutral</td>
<td>79</td>
</tr>
<tr>
<td>2019-07-06</td>
<td>positive</td>
<td>20</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>neutral</td>
<td>91</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>positive</td>
<td>9</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>negative</td>
<td>2</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>neutral</td>
<td>87</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>positive</td>
<td>11</td>
</tr>
</tbody>
</table>

8 rows

tot_count <- count(df, c('created'))
tot_count

<table>
<thead>
<tr>
<th>created</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-07-06</td>
<td>300</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>300</td>
</tr>
</tbody>
</table>
# rename the column
group_count <- rename(group_count, c("freq"="sentiment_count"))

# rename the column
tot_count <- rename(tot_count, c("freq"="tot_count"))
```r
# merge the group count and total count
final_count <- merge(group_count, tot_count, by="created")

final_count

<table>
<thead>
<tr>
<th>created</th>
<th>sentiment</th>
<th>sentiment_count</th>
<th>tot_count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-07-06</td>
<td>negative</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-06</td>
<td>neutral</td>
<td>79</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-06</td>
<td>positive</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>neutral</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>positive</td>
<td>9</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>negative</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>neutral</td>
<td>87</td>
<td>100</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>positive</td>
<td>11</td>
<td>100</td>
</tr>
</tbody>
</table>

# calculate the ratio
final_count$sentiment_ratio <- final_count$sentiment_count / final_count$tot_count

final_count

<table>
<thead>
<tr>
<th>created</th>
<th>sentiment</th>
<th>sentiment_count</th>
<th>tot_count</th>
<th>sentiment_ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-07-06</td>
<td>negative</td>
<td>1</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-06</td>
<td>neutral</td>
<td>79</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-06</td>
<td>positive</td>
<td>20</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-07</td>
<td>neutral</td>
<td>91</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-07</td>
<td>positive</td>
<td>9</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-08</td>
<td>negative</td>
<td>2</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-08</td>
<td>neutral</td>
<td>87</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2019-07-08</td>
<td>positive</td>
<td>11</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
```
### Data Table

<table>
<thead>
<tr>
<th>Date</th>
<th>Sentiment</th>
<th>Frequency</th>
<th>Total Sentiments</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-07-06</td>
<td>negative</td>
<td>1</td>
<td>100</td>
<td>0.01</td>
</tr>
<tr>
<td>2019-07-06</td>
<td>neutral</td>
<td>79</td>
<td>100</td>
<td>0.79</td>
</tr>
<tr>
<td>2019-07-06</td>
<td>positive</td>
<td>20</td>
<td>100</td>
<td>0.20</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>neutral</td>
<td>91</td>
<td>100</td>
<td>0.91</td>
</tr>
<tr>
<td>2019-07-07</td>
<td>positive</td>
<td>9</td>
<td>100</td>
<td>0.09</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>negative</td>
<td>2</td>
<td>100</td>
<td>0.02</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>neutral</td>
<td>87</td>
<td>100</td>
<td>0.87</td>
</tr>
<tr>
<td>2019-07-08</td>
<td>positive</td>
<td>11</td>
<td>100</td>
<td>0.11</td>
</tr>
</tbody>
</table>

8 rows

---

```r
# draw the plot
library(ggplot2)

Attaching package: ‘ggplot2’

The following object is masked from ‘package:NLP’:

annotate

ggplot(final_count, aes(x=created, y=sentiment_ratio)) + geom_line(aes(group=sentiment, color=sentiment), size=2)
```
```r
# get only the positive results
final_count_positive <- subset(final_count, sentiment=='positive')
final_count_positive
```

```r
colnames(final_count_positive) <- c('created', 'sentiment', 'sentiment_count', 'tot_count', 'sentiment_ratio')
final_count_positive
```

```r
# draw the plot on the positive only
ggplot(final_count_positive, aes(x=created, y=sentiment_ratio, group = 1)) + geom_line(size=2, color='cornflowerblue')
```