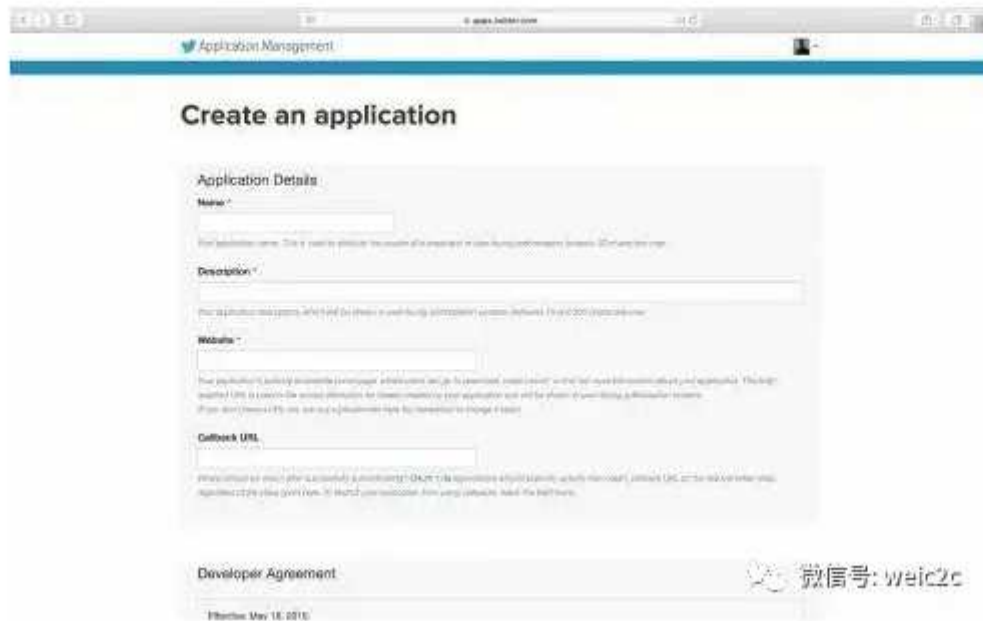


为了能够访问Twitter数据编程，我们需要创建一个与Twitter的API交互的应用程序。



**Create an application**

**Application Details**

**Name \***

Your application name. This is used to identify the account of an application in our developer systems. (20 characters max)

**Description \***

Your product description. We'll use this to show a brief description of your application. (140 characters max)

**Website \***

Your product's public website (homepage) address. We'll use this to verify your application. This URL is used to verify the application's domain. Your application will be shown in your listing, a developer listing. (If you don't have a website, you can use a placeholder like http://example.com)

**Callback URL**

This URL is used to receive the OAuth 1.0a response after a successful authentication. It must be a valid URL and registered with your provider. (If you don't have a callback URL, you can use a placeholder like http://example.com)

**Developer Agreement**

I agree with the terms of the Twitter API Developer Agreement.

Effective May 18, 2015

微信号: weic2c

注册后你将收到一个密钥和密码：

**Your Access Token**

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	229834223-DXyN4o2rGwaX2N4od3xwCoTIk6JLSoyxVPd6KuPDR8N
Access Token Secret	8Tsc8azA1yu1i08sBWXjw1Ybj6Sj1LSoyxVPd6KuPDR8N
Access Level	Read and write

**Application Settings**

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	DeOcQAW6Js0gkRSkWDqTNgv9F
Consumer Secret (API Secret)	QPwmj1U7NLpjdUqseFid1JUdyldU3J2XIUS5NCYxZ8cAtsw4
Access Level	Read, write, and direct messages (modify app permissions)

微信号: weic2c

获取密钥和密码后便可以在R里面授权我们的应用程序以代表我们访问Twitter：

```
options httr_oauth_cache=T
api_key = "4P0r42C2fG5nRKoe1ubJP1N7w"
api_secret = "SpHvrzMTUVVxUERjqFPzk090GA7oiDM8NlqOT8FNHY28Av9M9Q"
access_token = "229834223-DXyN4o2rGwaX2N4od3xwCoTIk6JLSoyxVPd6KuPDR8N"
access_token_secret = "8Tsc8azA1yu1i08sBWXjw1Ybj6Sj1LSoyxVPd6KuPDR8N"
```

微信号: weic2c

根据不同的搜索词，我们可以在几分钟之内收集到成千上万的tweet。这里我们测试一个关键词 littlecaesars的twitter结果：

## 抓取最新的1000条相关twitter

由于默认的抓取结果是json格式，因此使用twlisttodf函数将其转换成数据框

```
data=searchTwitter('littlecaesars',n=1000)
# Then I transform data to dataframe
d = twListToDF(data)
# I can view twitter data
```

然后我们做一些简单的文本清理

```
res=gsub(pattern="&"," ",res);
#Remove special words
res=gsub(pattern="#|@|*|%|!|>|<"," ",res);
res=gsub(pattern="[1|2|3|4|5|6|7|8|9|0]"," ",res);
```

从得到的数据里，我们可以看到有twitter发表时间，内容，经纬度等信息

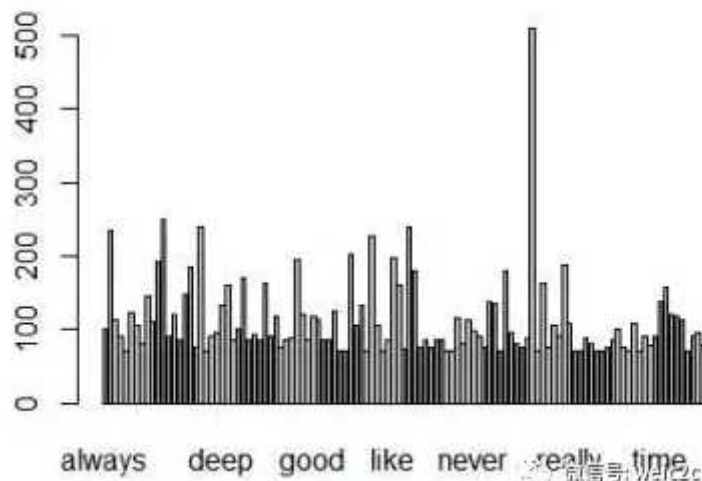
text	favorited	favoriteCount	replyToSN	created	retweeted	replyToSID	id
1 RT @Well_Regulated_ A member of our well regulated...	FALSE	0	N/A	2016-10-19 07:59:24	FALSE	N/A	78865074346
2 RT @Well_Regulated_ A member of our well regulated...	FALSE	0	N/A	2016-10-19 06:55:01	FALSE	N/A	78863451972
3 RT @richkenny17 @Jonathan_Just0 @littlecaesars ight...	FALSE	0	N/A	2016-10-19 05:53:24	FALSE	N/A	78861903568
4 @PapaJohns @littlecaesars @dominos @mesapizza wh...	FALSE	0	PapaJohns	2016-10-19 05:53:18	FALSE	N/A	78861900015
5 @Jonathan_Just0 @littlecaesars lghn dude	FALSE	0	Jonathan_Just0	2016-10-19 05:49:41	FALSE	788617891365154816	78861810189
6 @richkenny17 @littlecaesars lol dude my tweet was a...	FALSE	0	richkenny17	2016-10-19 05:48:51	FALSE	788617598141333504	78861789136
7 @Jonathan_Just0 @littlecaesars if they say 10 mins its...	FALSE	0	Jonathan_Just0	2016-10-19 05:47:41	FALSE	788616804352552961	78861759814
8 @Jonathan_Just0 @littlecaesars but you will learn. Ppl	FALSE	0	Jonathan_Just0	2016-10-19 05:47:11	FALSE	788616804352552961	78861747257
9 @richkenny17 @littlecaesars a lil too late for the advic...	FALSE	0	richkenny17	2016-10-19 05:44:32	FALSE	788616804352552961	78861680435
10 @Jonathan_Just0 @littlecaesars chill with that.	FALSE	0	Jonathan_Just0	2016-10-19 05:43:10	FALSE	788616804352552961	78861685660
11 I feel like @littlecaesars hurts themselves by playing c...	FALSE	0	N/A	2016-10-19 05:20:01	FALSE	N/A	78861061546
replyToSID	statusSource	screenName	retweetCount	isRetweet	retweeted	longitude	latitude
51420288	57824414	<a href="http://twitter.com/download/android" rel=...	0	FALSE	FALSE	N/A	N/A
17125824	N/A	<a href="http://foursquare.com" rel="nofollow">Fo...	0	FALSE	FALSE	-100.28612282	25.77062931
75889056	57824414	<a href="http://twitter.com/download/android" rel=...	0	FALSE	FALSE	N/A	N/A
29058562	2455171147	<a href="http://twitter.com/download/iphone" rel=...	0	FALSE	FALSE	N/A	N/A
24636290	N/A	<a href="http://twitter.com/download/android" rel=...	0	FALSE	FALSE	N/A	N/A
88692484	2455171147	<a href="http://twitter.com" rel="nofollow">Twitter	0	FALSE	FALSE	N/A	N/A
32125441	2455171147	<a href="http://twitter.com" rel="nofollow">Twitter	0	FALSE	FALSE	N/A	N/A
36224128	57824414	<a href="http://twitter.com/download/android" rel=...	0	FALSE	FALSE	N/A	N/A
30440192	2455171147	<a href="http://twitter.com/download/android" rel=...	0	FALSE	FALSE	N/A	N/A
26669056	N/A	<a href="http://twitter.com/download/android" rel=...	261	TRUE	FALSE	N/A	N/A
58089760	N/A	<a href="http://twitter.com/download/iphone" rel=...	261	TRUE	FALSE	N/A	N/A

在清理数据之后，我们对twitter内容进行分词，以便进行数据可视化

```
dtm <- DocumentTermMatrix(reuters,
                           control = list(weighting =
                                           function(x)
                                             weightTfIdf(x, normalize =
                                                         stopwords = TRUE))
```

分词之后可以得到相关twitter的高频词汇，然后将其可视化

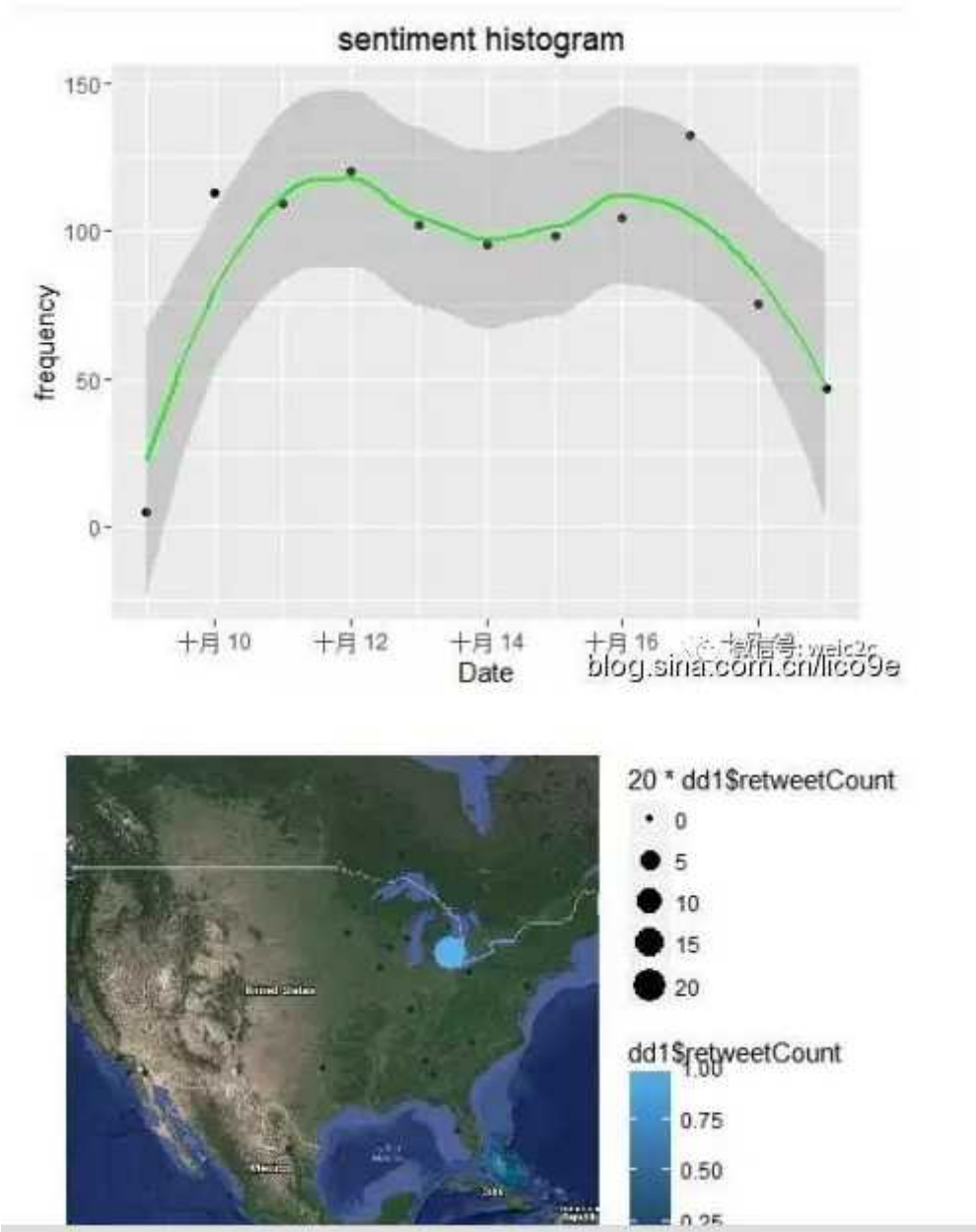
##	Terms
## Docs	'fine' 'll' 've' -days
## 1	0 0 0 0
## 2	0 0 0 0
## 3	0 0 0 0



bites  
hot amp;  
get breadlike  
pizza  
crazy just  
littlecaesars

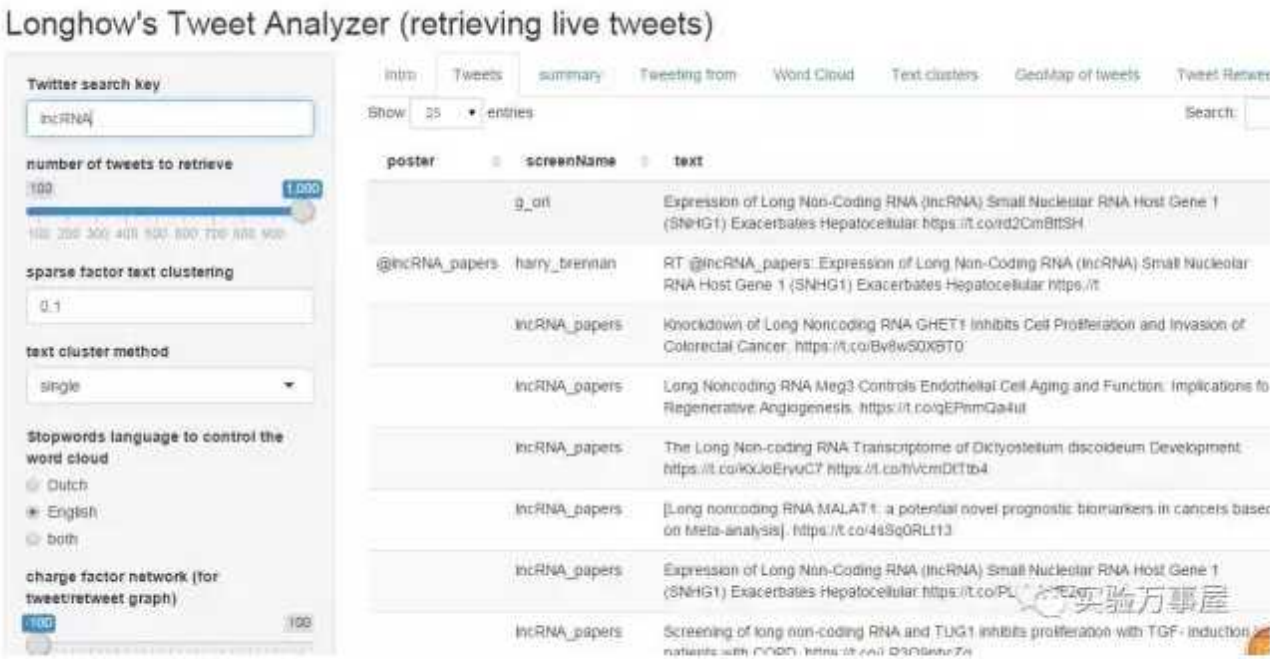
除此之外，还可以结合数据中的时间戳数据和地理数据进行可视化分析





推特和FB其实也是科研讨论的重镇。但是要怎么来分析推特上都讨论啥呢？光用Mendeley的话，只能有只言片语，这次又要带你打开新世界的大门了。

首先推荐用一款推特分析工具网站，叫做推特分析家，功能是实时分析推特上的动态。这是一款基于R语言Shiny的网页，由于这个是德国人做的，所以，会分析德语和英语两种语言。



所用到的数据分析的资源，其实就是推特上的人家的东西。会对这些文字，进行文本挖掘，然后来分析你要的东西。比如，我分析一下LncRNA哈。

左侧的是文本数据的来源，可以发现，这最近的推特还是前几天刚发的。也就是说这个网站分析的数据都是实时数据。

那这些推特具体讲的内容有些啥呢？主要是来自于LncRNA的论文和一些杂志的推送。反正推特就好比国外科研狗的票圈，转一下杂志上的牛逼文章的话，可能会显得更上等次，也是在跟老板说：看我很勤奋，有在看文献哦！

在WordCloud里，就会显示在推特上，讨论的最多的和lncRNA有关的词汇。比如：表达，变化，剪切，模式，肿瘤等等，说实话是没有什么特别大的用处哈。

接着是词频的分簇，可以看得到大概这个词在所有的句子中出现频率的分簇分析。但我  
不懂 “httpstcodadiagxfh” 有啥关系 “not so junk dna” ，要么是个网站啥的.....好吧，不管怎  
么样，好像推特上热议的应该是lncRNA的剪切。



接下去看一下network，这个是推送这些推特的账号之间的联系，可以看得，有蛮多是杂志，还有一些是谁？有可能就是痴迷于lncRNA研究的那些人了，有机会的话，可以考虑关注一下。

此外还有些附带功能，比如，我们会巧妙地发现，在国外研究lncRNA的安卓狗主要是上班前和下班后会推送lncRNA的内容。而苹果狗则集中在下午和午夜.....

网址：[longhowlam.shinyapps.io...](http://longhowlam.shinyapps.io/)

还顺便分析了一下别的关键词，比如：机器学习、深度学习。

这种网站基本都要翻墙才能进去的呢，毕竟要调用推特的数据。不过作为爱国少年的我，也想看看推特上都在讨论中国什么，于是我搜了一下“China”调整到推特内容1000，结果：

好吧，最近川普大爷赢了.....

近日，一直以“**推特治国**”闻名的川普正式宣誓就任了美国第 45 任总统。

川普这次在美国大选中胜出，他的推特也发挥了巨大的作用。相比大多数总统竞选人来说，他们都没时间自己发推。但推特玩的风生水起的川普却表示，他的推特都是自己发的.....

那么事实真的是这样吗？

有个美国网友发现**川普发推特有两个客户端**。一个安卓，另一个是 iPhone 。

而且这位细心的网友还发现，一些**言辞激烈**的推都**来自安卓**；而**画风比较正常**的推都**来自 iPhone**。

这一发现，也引起了数据分析师 David Robinson 的注意。David 注意到当川普发祝贺内容时，是通过 iPhone ；而当他抨击竞选对手时而是通过安卓。而且两个不同客户端通常发推的时间也不太相同。

本着科学严谨的态度，程序员小哥决定让数据说话，于是做了程序，抓取分析了川普发过的推，终于发现了一些模式。并且通过统计，图表，最终他基本确定，川普的推特并不是他一个人写的。

数据证明，安卓端和iPhone发的推分别是两个人所写的。而且发推时间，使用标签，加链接，转发的方式也截然不同。同时，安卓端发的内容更加激烈和消极。

如果就像川普采访中所说他使用的手机是三星 Galaxy ，我们可以确信用安卓发推的是川普本人，用 iPhone 发的大概是他的团队助理。

## 发推时间对比

首先用 twitterR 包中的 userTimeline 函数导入川普发推的时间数据：

◆ library ( dplyr )

◆ library ( purrr )

◆ library ( twitterR )

```
# You'd need to set global options with an authenticated  
appsetup_twitter_oauth(getOption("twitter_consumer_key"),
```

```
getOption("twitter consumer secret").
```

```
getOption("twitter_access_token"),

getOption("twitter_access_token_secret"))

# We can request only 3200 tweets at a time; it will return fewer

# depending on the APItrump_tweets <- userTimeline("realDonaldTrump", n =
3200)trump_tweets_df <- tbl_df(map_df(trump_tweets, as.data.frame))

# if you want to follow along without setting up Twitter authentication,

# just use my dataset:load(url("varianceexplained.org/f/..."))
```

稍微清理下数据,提取源文件。(在此只分析来自 iPhone 和 Android tweet 的数据, 除去很少一部分发自网页客户端和 iPad 的推文)。

```
library(tidyr)

tweets <- trump_tweets_df %>%

select(id, statusSource, text, created) %>%

extract(statusSource, "source", "Twitter for (.*)<") %>%

filter(source %in%c("iPhone", "Android"))
```

分析的数据包括**来自 iPhone 的 628 条推文**, **来自 Android 的 762 条推文**。

主要考虑推文是在一天内什么时间发布的, 在此我们可以发现区别:

```
◆ library(lubridate)

◆ library(scales)

tweets %>%

count(source, hour = hour(with_tz(created, "EST"))) %>%

mutate(percent = n /sum(n)) %>%

ggplot(aes(hour, percent, color = source)) +

geom_line() +
```

```
scale_y_continuous(labels = percent_format()) +  
  
labs(x = "Hour of day (EST)",  
  
y = "% of tweets",  
  
color = "")
```

川普一般习惯早上发推，而他的助理会集中在下午或晚上发推。

## 发文习惯对比

当川普的安卓手机转推时，习惯用双引号引用这整句话。



而 iPhone 转推时，一般不使用**双引号**。

安卓手机：500 多条推文没有双引号，200 多条有双引号

iPhone：几乎没有双引号

与此同时，在分享**链接和图片**时，安卓和 iPhone 也大不相同。

```
tweet_picture_counts <- tweets %>%
```

```
filter(!str_detect(text, '^\"')) %>%
```

```
count(source,
```

```
picture = ifelse(str_detect(text, "t.co"),
```

```
"Picture/link", "No picture/link"))

ggplot(tweet_picture_counts, aes(source, n, fill = picture)) +

geom_bar(stat = "identity", position = "dodge") +

labs(x = "", y = "Number of tweets", fill = "")
```

数据证明 iPhone 端 发的推文很多会附上图片，链接。内容也以宣传为主。

比如下面这条：

而川普安卓端发的推文没有图片、链接，更多是直接的文字，比如：

## 用词对比

在对比安卓和 iPhone 用词区别时，David 用到了他和 Julia Silge 一起编写的 tidytext 包。

用 unnest\_tokensfunction 把句子分解为单独的词:

```
library(tidytext)
```

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

```
filter(!str_detect(text, '^")) %>%
```

```
mutate(text = str_replace_all(text, "t.co/[A-Za-z\\d]+|&", "")) %>%
```

```
unnest_tokens(word, text, token = "regex", pattern = reg) %>%
```

```
filter(!word %in% stop_words$word)
```

```
intersect(word %in% stop_words$word,
```

```
str_detect(word, "[a-z]"))
```

```
tweet_words
```

```
## # A tibble: 8,753 x 4
```

```
## id source created word
```

```
## <chr> <chr> <time> <chr>
```

```
## 1 676494179216805888 iPhone 2015-12-14 20:09:15 record
```

```
## 2 676494179216805888 iPhone 2015-12-14 20:09:15 health
```

```
## 3 676494179216805888 iPhone 2015-12-14 20:09:15 #makeamericagreatagain
```

```
## 4 676494179216805888 iPhone 2015-12-14 20:09:15 #trump2016
```

```
## 5 676509769562251264 iPhone 2015-12-14 21:11:12 accolade
```

```
## 6 676509769562251264 iPhone 2015-12-14 21:11:12 @trumpgolf
```

```
## 7 676509769562251264 iPhone 2015-12-14 21:11:12 highly
```

```
## 8 676509769562251264 iPhone 2015-12-14 21:11:12 respected
```

```
## 9 676509769562251264 iPhone 2015-12-14 21:11:12 golf
```

```
## 10 676509769562251264 iPhone 2015-12-14 21:11:12 odyssey
```

```
## # ... with 8,743 more rows
```

总体来说川普推文中有哪些常用词呢？

在此基础上我们再来分别看安卓和 iPhone 常用词的区别。

```
android_iphone_ratios <- tweet_words %>%  
count(word, source) %>%  
filter(sum(n) >= 5) %>%  
spread(source, n, fill = 0) %>%  
ungroup() %>%  
mutate_each(funs((. + 1) / sum(. + 1)), -word) %>%  
mutate(logratio = log2(Android / iPhone)) %>%  
arrange(desc(logratio))
```

## 结论

- 带标签的推文基本来自 iPhone 。
- iPhone 推文中常用词有宣传性的词，比如：“参加”，“明天”，“晚上 7 点”。
- 安卓的推文常用有强烈情绪性的词汇，“差劲”，“疯了”，“软弱”，“傻瓜”等等。

## 情感分析

安卓和 iPhone 推文在情感上也有很大的差异，让我们来量化一下。用到 tidytext 当中的 NRC Word-Emotion Association 词典，主要把用词联系以下十种情绪分析：积极,消极,愤怒,期待,厌恶,恐惧,快乐,悲伤,惊讶,信任。

```
nrc <- sentiments %>%
```

```
filter(lexicon == "nrc") %>%
```

```
dplyr::select(word, sentiment)
```

```
nrc
```

```
## # A tibble: 13,901 x 2
```

```
## word sentiment
```

```
## <chr> <chr>
```

```
## 1 abacus trust
```

```
## 2 abandon fear
```

```
## 3 abandon negative
```



```
## 4 abandon sadness
```

```
## 5 abandoned anger
```

```
## 6 abandoned fear
```

```
## 7 abandoned negative
```

```
## 8 abandoned sadness
```

```
## 9 abandonment anger
```

```
## 10 abandonment fear
```

```
## # ... with 13,891 more rows
```

为了分别计算安卓和 iPhone 推文的情感，可以把不同用词分类。

```
sources <- tweet_words %>%
```

```
group_by(source) %>%
```

```
mutate(total_words = n()) %>%
```

```
ungroup() %>%
```

```
distinct(id, source, total_words)
```

```
by_source_sentiment <- tweet_words %>%
```

```
inner_join(nrc, by = "word") %>%
```

```
count(sentiment, id) %>%
```

```
ungroup() %>%
```

```
complete(sentiment, id, fill =list(n = 0)) %>%
```

```
inner_join(sources) %>%
```

```
group_by(source, sentiment, total_words) %>%
```

```
summarize(words =sum(n)) %>%
```

```
ungroup()

head(by_source_sentiment)

## # A tibble: 6 x 4

## source sentiment total_words words

## <chr> <chr> <int> <dbl>

## 1 Android anger 4901 321

## 2 Android anticipation 4901 256

## 3 Android disgust 4901 207

## 4 Android fear 4901 268

## 5 Android joy 4901 199

## 6 Android negative 4901 560
```

( 比如 , 我们可以看到安卓推文中 4901 个词中 321 个词与情感 “愤怒” 有关。 )

同时可以用 Poisson test 分析 , 比起 iPhone , 安卓推文更喜欢使用带强烈情绪的词。

```
library(broom)

sentiment_differences <- by_source_sentiment %>%

group_by(sentiment) %>%

do(tidy(poisson.test(.$words, .$total_words)))

sentiment_differences

## Source: local data frame [10 x 9]

## Groups: sentiment [10]

##

## sentiment estimate statistic p.value parameter conf.low
```

```
## (chr) (dbl) (dbl) (dbl) (dbl) (dbl)

## 1 anger 1.492863 321 2.193242e-05 274.3619 1.2353162

## 2 anticipation 1.169804 256 1.191668e-01 239.6467 0.9604950

## 3 disgust 1.677259 207 1.777434e-05 170.2164 1.3116238

## 4 fear 1.560280 268 1.886129e-05 225.6487 1.2640494

## 5 joy 1.002605 199 1.000000e+00 198.7724 0.8089357

## 6 negative 1.692841 560 7.094486e-13 459.1363 1.4586926

## 7 positive 1.058760 555 3.820571e-01 541.4449 0.9303732

## 8 sadness 1.620044 303 1.150493e-06 251.9650 1.3260252

## 9 surprise 1.167925 159 2.174483e-01 148.9393 0.9083517

## 10 trust 1.128482 369 1.471929e-01 350.5114 0.9597478

## Variables not shown: conf.high (dbl), method (fctr), alternative (fctr)
```

我们可以用 95% 的置信区间来明确二者的区别:

从而我们可知，川普安卓的推文比起 iPhone，使用“厌恶”“悲伤”“恐惧”“愤怒”等消极情绪词的比例高 40-80%

在数据挖掘下

川普推特背后的团队就这么被扒了个精光

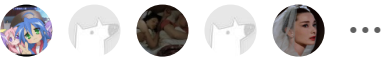
所以，看川普的推特，只要看安卓端的就好了。

但据报道，上任后的川普必须使用一部由美国特工处认证的安全加密手机，以替换他之前使用的安卓系统手机。据称前总统奥巴马就无法通过安全手机发推文，那使用安全手机后，川普还能继续愉快的“推特治国”吗？

**大家也可以加小编微信：tswenqu，进R语言中文社区 交流群，可以跟各位老师互相交流。**

☆ 收藏   分享   举报

👍 83



10 条评论

写下你的评论...



**Coka**  
谢谢作者分享，R其实还是很棒的，只是都被Python抢了风头...  
10 个月前



**氦氖氩氟氙氡**  
标题少个r  
10 个月前



**动心忍性**  
给数据帝跪了，虽然一知半解，但是觉得很厉害...  
10 个月前



**李晓文 (作者) 回复 氦氖氩氟氙氡**  
正常的  
10 个月前

查看对话



**李晓文 (作者) 回复 Coka**  
是呀，  
10 个月前

查看对话



**Bella**  
安卓的推文常用有强烈情绪性的词汇，“差劲”，“疯了”，“软弱”，“傻瓜”等等2333

10 个月前



倪妹倪妹的

好厉害！

10 个月前



喂鱼

还不错 能分享源代码吗 爬虫的

10 个月前



mockingbird

R也能爬虫啊，不知用python什么结果。棒棒的，数据分析还看R!

10 个月前



柳毅

我觉得转载别人文章的时候给个原文链接。比如David Robinson的。让人知道哪些是refered的哪些事original的。

10 个月前

1 赞

## 文章被以下专栏收录



R语言中文社区

R语言专业学习平台、视频、资讯、核心资源资源库、

[进入专栏](#)

## 推荐阅读



### 文本挖掘：手把手教你分析携程网评论数据

文本分析的应用越来越广泛，这不，我的工作也开始涉及了文本分析，今天就讲讲关于评论数据的... [查看全文](#) >

李晓文 · 10 个月前 · 发表于 R语言中文社区





使用R语言爬取川普人物进行情绪分析

人脸提供关于情绪的各种信息。微软于2015年12月推出免费服务，分析人脸，进行情绪检测。 检... 查看全文 >

李晓文 · 10 个月前 · 发表于 R语言中文社区



## 大数据分析美国大选——Twitter数据情感分析

在当今这个互联网时代，人们对于各种事情的舆论观点都散布在各种社交网络平台或新闻提要中。... 查看全文 >

Wayne Shi · 1 年前 · 发表于 程序员实验室



## 运用R和Tableau对美国总统候选人Donald Trump进行情绪分析

摘要： 文本挖掘、情绪分析，特别是对中文的文本挖掘，一直是学界及工业界比较难的课题，... 查看全文 >

李晓文 · 8 个月前 · 发表于 R语言中文社区