SentiMap

HU WEI

1 March 2018

Import Data

First of all, we can import the data.

```r
rm(list=ls())
t2<-read.csv("twitter_file_with_text.csv",fill=T, sep=",", stringsAsFactors = FALSE)
names(t2)
```

```
##  [1] "follow_request_sent"
##  [2] "contributors"
##  [3] "truncated"
##  [4] "profile_use_background_image"
##  [5] "profile_sidebar_fill_color"
##  [6] "time_zone"
##  [7] "in_reply_to_status_id"
##  [8] "id"
##  [9] "favorite_count"
## [10] "verified"
## [11] "sentiment"
## [12] "profile_text_color"
## [13] "profile_image_url_https"
## [14] "retweeted"
## [15] "is_translator"
## [16] "source"
## [17] "followers_count"
## [18] "protected"
## [19] "in_reply_to_screen_name"
## [20] "in_reply_to_user_id"
## [21] "default_profile_image"
## [22] "retweet_count"
## [23] "id_str"
## [24] "favorited"
## [25] "utc_offset"
## [26] "statuses_count"
## [27] "profile_background_color"
## [28] "friends_count"
## [29] "profile_background_image_url_https"
## [30] "profile_link_color"
## [31] "profile_image_url"
## [32] "notifications"
## [33] "geo_enabled"
## [34] "profile_banner_url"
## [35] "in_reply_to_user_id_str"
## [36] "profile_background_image_url"
## [37] "lang"
## [38] "profile_background_tile"
## [39] "favourites_count"
## [40] "screen_name"
## [41] "url"
## [42] "created_at"
## [43] "contributors_enabled"
## [44] "location"
## [45] "filter_level"
## [46] "in_reply_to_status_id_str"
## [47] "profile_sidebar_border_color"
## [48] "place"
## [49] "default_profile"
## [50] "following"
## [51] "listed_count"
```

We can see that there are **51** variables in this dataset.

```
attach(t2)
Size<-dim(t2)
Size
```

```
## [1] 2491   51
```

There are **2491** observations.

Classes of variables

Get the class of each variable in dataset.

```
lapply(t2, class)
```

```
## $follow_request_sent
## [1] "logical"
##
## $contributors
## [1] "logical"
##
## $truncated
## [1] "character"
##
## $profile_use_background_image
## [1] "character"
##
## $profile_sidebar_fill_color
## [1] "character"
##
## $time_zone
## [1] "character"
##
## $in_reply_to_status_id
## [1] "numeric"
##
## $id
## [1] "integer"
##
## $favorite_count
## [1] "integer"
##
## $verified
## [1] "character"
##
## $sentiment
## [1] "integer"
##
## $profile_text_color
## [1] "character"
##
## $profile_image_url_https
## [1] "character"
##
## $retweeted
## [1] "character"
##
## $is_translator
## [1] "character"
##
## $source
## [1] "character"
##
## $followers_count
## [1] "integer"
##
## $protected
## [1] "character"
##
## $in_reply_to_screen_name
## [1] "character"
##
## $in_reply_to_user_id
## [1] "integer"
##
## $default_profile_image
## [1] "character"
##
## $retweet_count
## [1] "integer"
##
## $id_str
## [1] "integer"
##
## $favorited
## [1] "character"
##
## $utc_offset
## [1] "integer"
##
## $statuses_count
## [1] "integer"
##
## $profile_background_color
## [1] "character"
##
## $friends_count
## [1] "integer"
##
## $profile_background_image_url_https
## [1] "character"
##
## $profile_link_color
## [1] "character"
##
## $profile_image_url
## [1] "character"
##
## $notifications
## [1] "logical"
##
## $geo_enabled
## [1] "character"
##
## $profile_banner_url
## [1] "character"
##
## $in_reply_to_user_id_str
## [1] "integer"
##
## $profile_background_image_url
## [1] "character"
##
## $lang
## [1] "character"
##
## $profile_background_tile
## [1] "character"
##
## $favourites_count
## [1] "integer"
##
## $screen_name
## [1] "character"
##
## $url
## [1] "character"
##
## $created_at
## [1] "character"
##
## $contributors_enabled
## [1] "character"
##
## $location
## [1] "character"
##
## $filter_level
## [1] "character"
##
## $in_reply_to_status_id_str
## [1] "numeric"
##
## $profile_sidebar_border_color
## [1] "character"
##
## $place
## [1] "character"
##
## $default_profile
## [1] "character"
##
## $following
## [1] "logical"
##
## $listed_count
## [1] "integer"
```

We are interested in numeric variables as follow:

- sentiment
- followers_count
- statuses_count
- friends_count
- favourites_count
- listed_count

Descriptive statistics

Sentiment

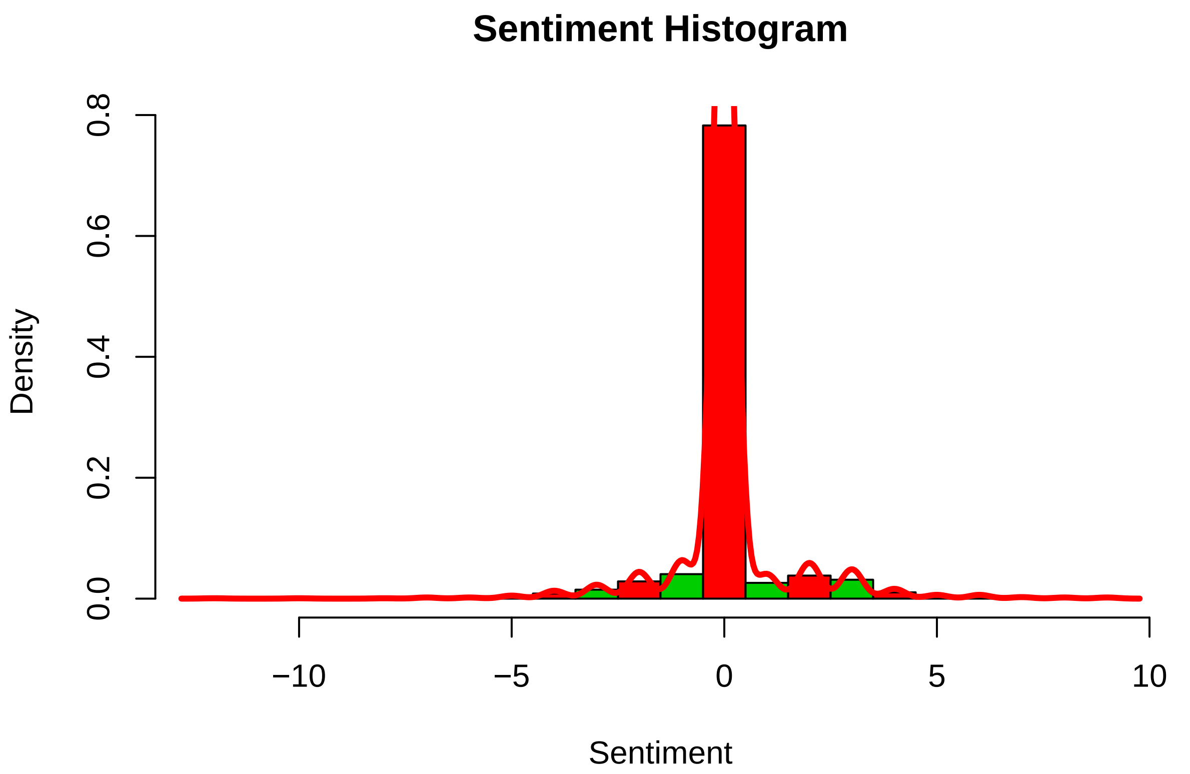
```
summary(sentiment)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -12.00000   0.00000   0.00000   0.09434   0.00000   9.00000
```

```
table(sentiment)
```

```
## sentiment
##  -12  -10   -8   -7   -6   -5   -4   -3   -2   -1    0    1    2    3    4
##    1    1    1    3    3    8   21   37   71  101 1950   65   95   78   26
##    5    6    7    8    9
##   10   10    4    3    3
```

```
hist(sentiment, freq=F, main="Sentiment Histogram", breaks=seq(from=-12.5,to=9.5,by=1), col=c(2,3), xlab
# Add the line of density, "col" for color, "lwd" for line width
lines(density(sentiment),col=2,lwd=3)
```



```
sum(sentiment==0)/Size[1]
```

```
## [1] 0.7828181
```

There are **20** level of sentiment and most of them are **neutral** (78.28%).

Followers count

```
summary(followers_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     104     252    2512     610 1379617
```

```
quantile(followers_count)
```

```
##      0%     25%     50%     75%    100%
##       0     104     252     610 1379617
```

We can find that the range of follower_count is really large. Thus we can analyze the logrithmic value of follower_count.

```
followers_count2<-log(followers_count[followers_count!=0])
summary(followers_count2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.663   5.545   5.536   6.422  14.137
```

```
quantile(followers_count2)
```

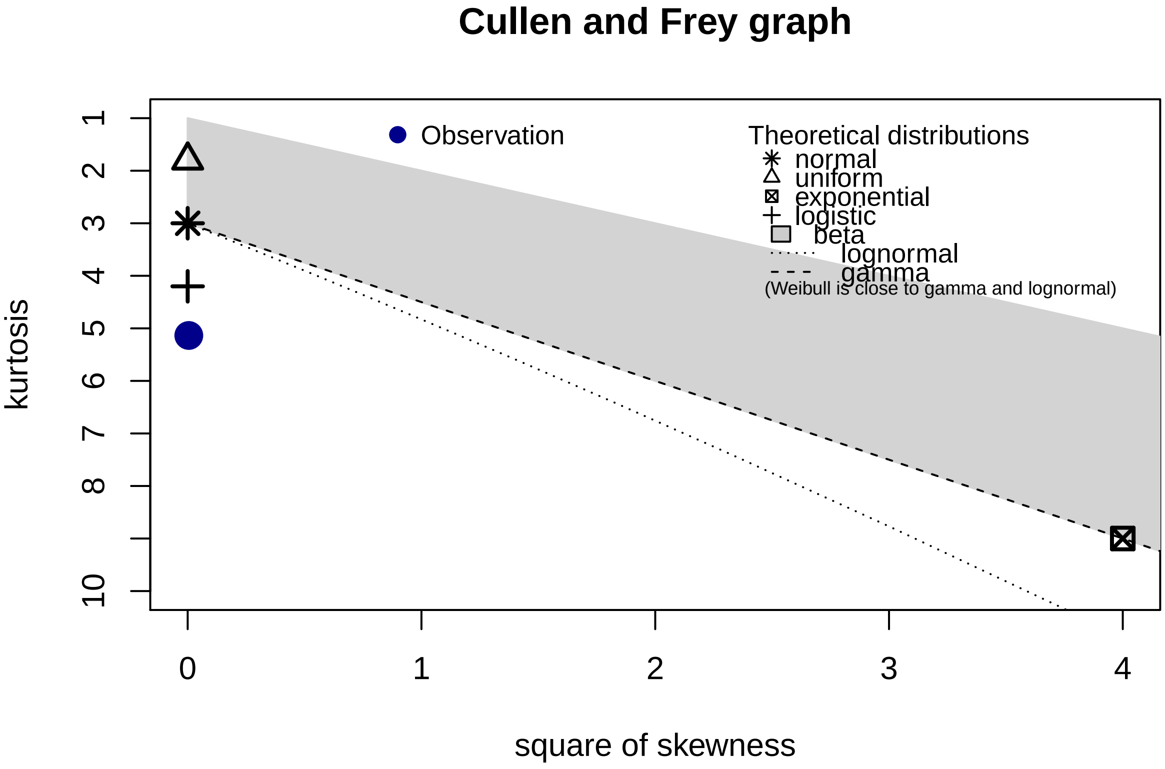
```
##        0%       25%       50%       75%      100%
##  0.000000  4.663439  5.545177  6.421622 14.137316
```

The result is better. Then we can use the excellent fitdistrplus package which offers some nice functions for distribution fitting. We will use the functiondescdist to gain some ideas about possible candidate distributions.

```
#install.packages("fitdistrplus")
library(fitdistrplus)
```

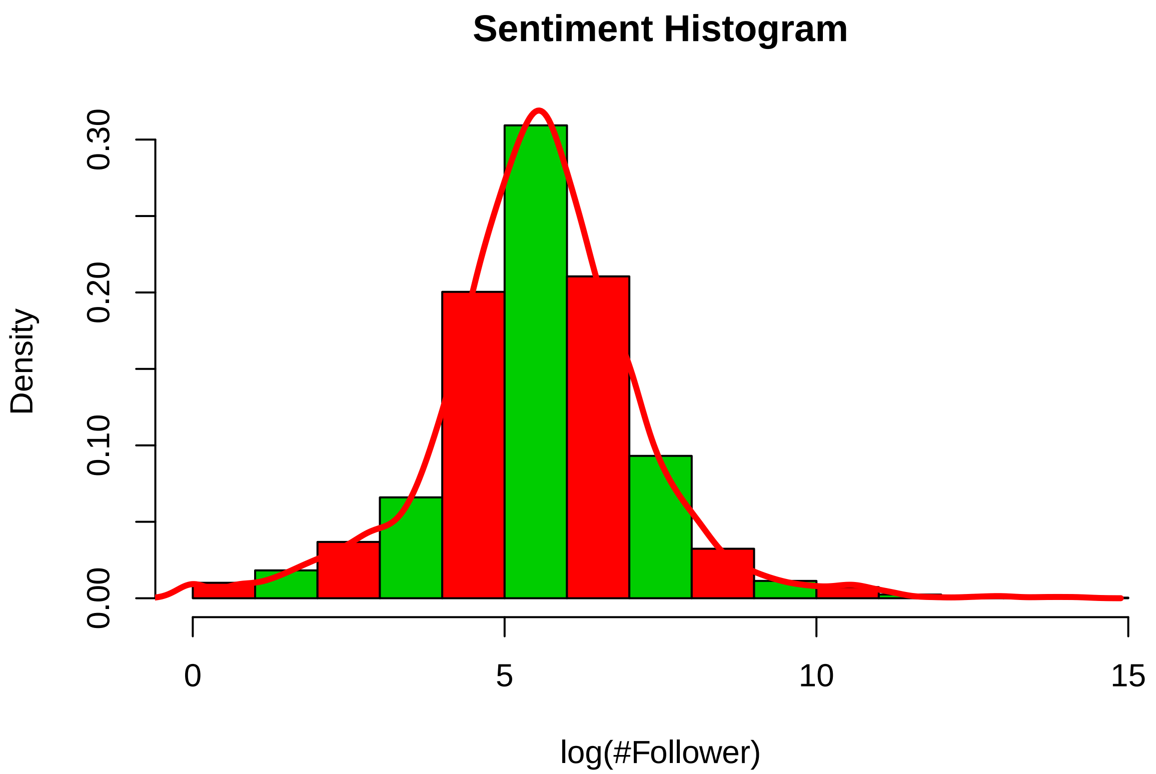
```
## Loading required package: MASS

## Loading required package: survival
```

```
#install.packages("logspline")
library(logspline)
descdist(followers_count2, discrete = FALSE)
```

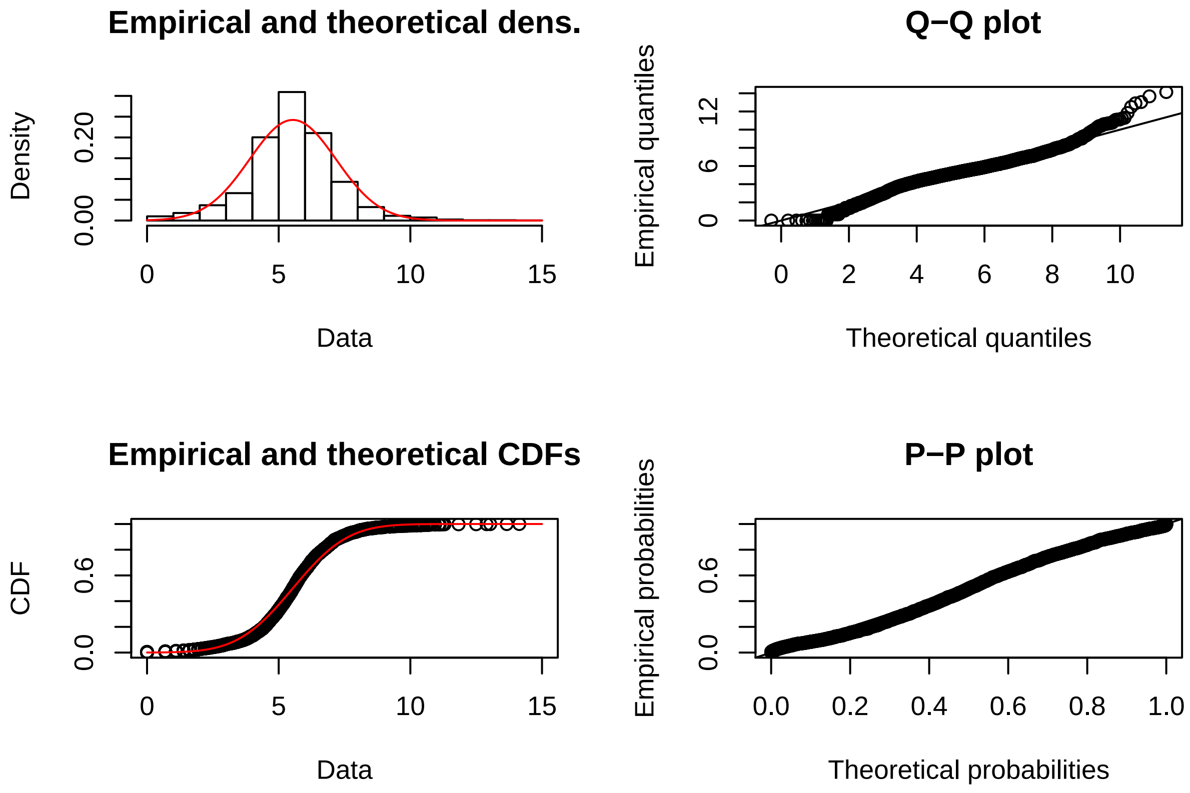


```
## summary statistics
## ------
## min:  0   max:  14.13732
## median:  5.545177
## mean:  5.536197
## estimated sd:  1.647834
## estimated skewness:  0.06936065
## estimated kurtosis:  5.135501
```

```
hist(followers_count2, freq=F, main="Sentiment Histogram", breaks=seq(from=0,to=15,by=1), col=c(2,3), xl
# Add the line of density, "col" for color, "lwd" for line width
lines(density(followers_count2),col=2,lwd=3)
```



```
fit.norm <- fitdist(followers_count2, "norm")
plot(fit.norm)
```



Percentage of followers number that exceeds 1000

```
sum(followers_count>1000)/Size[1]
```

```
## [1] 0.1633882
```

Percentage of followers number that exceeds 5000

```
sum(followers_count>5000)/Size[1]
```

```
## [1] 0.0337214
```

Statuses Count

```
summary(statuses_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0    1239    4387   11729   13182  295091 
```

```
summary(friends_count)
```

```
##     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##      0.0     120.0     258.0     921.0     557.5 354695.0 
```

```
summary(favourites_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.0    10.0    83.0   831.8   445.0 94135.0 
```

```
summary(listed_count)
```

```
##     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##     0.00      0.00      0.00     17.72      1.00 12319.00 
```

```
sum(lang=="en")/Size[1]
```

```
## [1] 0.505821
```

```
sum(geo_enabled == "True")/Size[1]
```

```
## [1] 0.3753513
```

```
sum(location[lang=="en"]=="")/Size[1]
```

```
## [1] 0.190285
```

Time Zone

Get geographical data

```
# Check version of R, becasue ggmap require R version higher than 3.4.3
#R.Version()
#install.packages("ggmap")
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
#install.packages("tidyverse")
#library(tidyverse)
```

```
# Check the version info of ggmap
#sessionInfo()
```

Read the georaphical data from `geocoded.csv`.

```
geocoded<-read.csv("geocoded.csv",fill=T, sep=",", stringsAsFactors = FALSE)
```

Plot Map

```
#install.packages("rworldmap")
library(rworldmap)
```

```
## Loading required package: sp
```

```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type :   vignette('rworldmap')
```

```
newmap <- getMap(resolution = "low")
plot(newmap, asp = 1)
points(geocoded$lon, geocoded$lat, col = "red", cex = .6)
```