# Data Mining and Analysis using rtweet package

#Censuskenya2019 tweets

*Innocenter Amima*

```
library(rtweet) #twitter mining. All you need is a Twitter account (user name and password)
library(ggplot2) #plotting
library(dplyr) #pipes  tidyverse
library(tidytext) # text mining
library(stopwords)

theme_set(theme_classic()) #setting theme to classic()
```

## Census 2019 in Kenya

The 8th 2019 Population and Housing Census started from the night of 24/25th August 2019 and ended on 31st August 2019.

Census involved counting of people within the border of Kenya at a specific time. Census is an important process for the Govenrment as it provides evidence for proper planning and resource allocation, policy formulation and targeting of development plans. You can read more about the census here and here

### Objectives

Here we shall

1. Perform data mining using `rtweet` package.
2. Determine unique words in #Censuskenya2019 tweets.
3. Identify top user accounts in #Censuskenya2019 tweets.
4. Plot time series of tweets including #Censuskenya2019.

### Data Mining

> I decided to use `rtweet` given that it has more functionality compared to other twitter APIS like `twitteR`, `streamR`.

Kindly note the tweets harvested are based on who I follow on Twitter - it is a sample of what people are tweeting about #Censuskenya2019.

```
censusTweets <- search_tweets(q="#Censuskenya2019",n=10000, include_rts = FALSE, lang='en')

censusKE <- censusTweets #creating a copy

#glimpse(censusKE)
```

The function `search_tweet()` returns tweets for the past 6-9 days. Unfortunately, I do not have a premium account - if you do try using `search_30day()` and the function requires env_name.

```
head(censusKE$text) #Top Tweet Unique Words
```

```
## [1] "Does &lt;ENUMERATOR&gt; have difficulty  waiting for census money ?\n1. Yes some difficulty \n2
## [2] "@KTNKenya @Ashleymazuri Wauh... So #Kibra voters Went home for #Censuskenya2019 ?"
## [3] "#Censuskenya2019 #mombasavoices #voicesinspaces #voicesinspacesmombasa\nCultural diversity makes
## [4] "#Universalhealthcoverage #CancerTrearmentke#Lifestyle #\"Ahealthynationisawealthynation\" so the
## [5] "How much did we spend on this again!!!??its funny till now I haven't seen the enumerators in my
## [6] "@wakanyago @ODPP_KE @StandardKenya @citizentvkenya @NTVnewsroom @KTNKenya @NationBreaking @TheS
```

## Data Cleaning, Analysis and Visualization

When tweeting people use connectors and other wordss. `tinytex` package has a function known as `stop_words()` that has three lexicons for English stop words. Below are some stop words

```
head(stop_words)
```

```
## # A tibble: 6 x 2
##   word      lexicon
##   <chr>     <chr>
## 1 a         SMART
## 2 a's       SMART
## 3 able      SMART
## 4 about     SMART
## 5 above     SMART
## 6 according SMART
```

We use `unnest_tokens()` from `tidytxt` package to convert any text from upper to lower case, remove punctuation, add unique ID. We clean the data, convert all the text to lower case and remove stop words

```
censusKE %>%
  dplyr::select(text) %>%
  unnest_tokens(Words, text) %>%
  filter(!Words %in% stop_words$word) %>%
  count(Words, sort=TRUE)
```

```
## # A tibble: 2,149 x 2
##    Words                n
##    <chr>            <int>
##  1 censuskenya2019    304
##  2 https              209
##  3 t.co               209
##  4 census              76
##  5 gainwithtreavor     64
##  6 gainwithxtiandela   64
##  7 thecountynews       63
##  8 trapadrive          63
##  9 kibradecides        61
## 10 knbstats            44
## # ... with 2,139 more rows
```

https appear as the 2nd highest word in #Censuskenya2019 - these represents links shared and we shall remove the https links from the text. Find and replace functions in base R include:

1. `sub(pattern, replacement, text)` replaces ONLY the first match in each element of a text vector.
2. `gsub(pattern, replacement, text)` replacess ALL the matching patterns of a text vector.

```
censusKE$strpWords <- gsub("https.*", "", censusKE$text) #removing https.* links
```

Base R has various functions that are used for regular expression and they achieve different outcomes. A very gentle introduction to regular expression has been done by Jon Calder as a course on Swirl(). Installation can be done by either using

```
library(swirl)
install_course("Regular Expressions")
swirl()
```

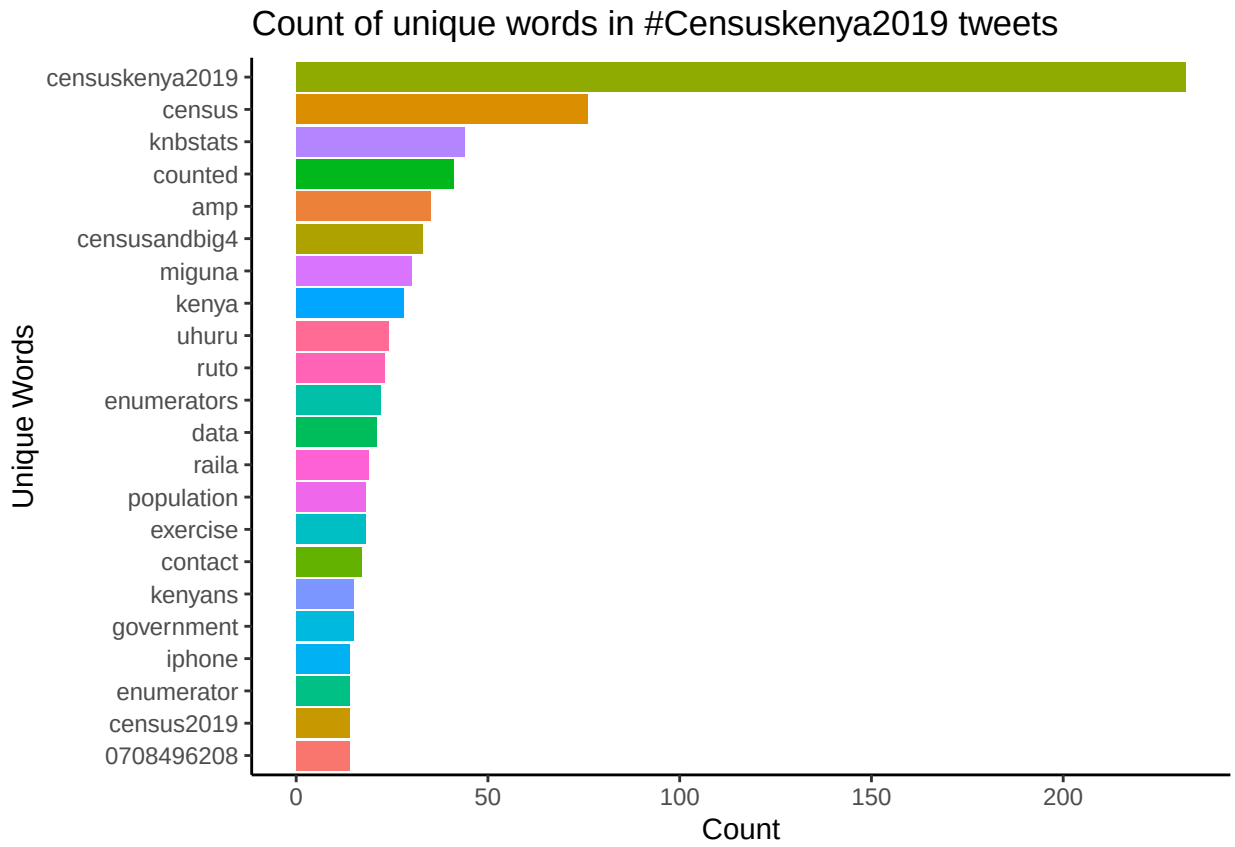
or alternatively downloading the latest version directly from Github

```
install_course_github("jonmcalder", "Regular_Expressions")
```

### Unique words in #Censuskenya2019 tweets

```
censusKE %>%
  dplyr::select(strpWords) %>%
  unnest_tokens(word, strpWords) %>%
  filter(!word %in% stop_words$word) %>%
  count(word, sort = TRUE) %>%
  top_n(20) %>%
  ggplot(censusKE, mapping =  aes(reorder(word, n), n)) +
    geom_bar(stat = 'identity', aes(fill=word), show.legend = FALSE) +
    coord_flip()+
    labs(title = "Count of unique words in #Censuskenya2019 tweets ", x="Unique Words", y="Count")
```

```
## Selecting by n
```



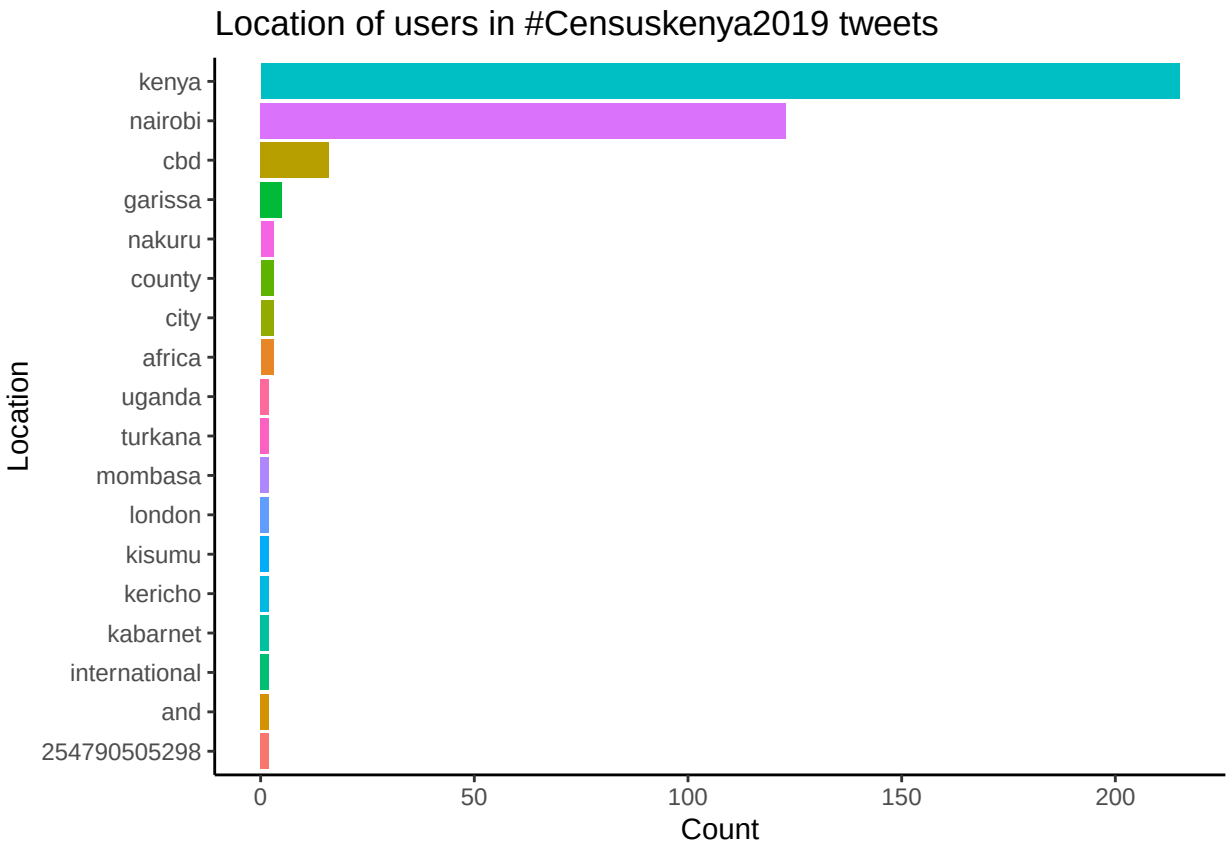

I was expecting to see *KNBS* : it is ranked number 3 on the list. *Censusandbig4* agenda ranks at the top 5. Politician names such as *Uhuru, Raila, Ruto* are among the top 20 unique words being tweeted under the hashtag. *Enumerators* were hired to perform this *excercise* and hence these two words are among the top 20 unique words. However, a *mobile number* is among the top 20 words - am not sure if it is a hot line for census?

### Top users in #Censuskenya2019 tweets

`user_data()` returns information of the users including screen names, location, creation time, description...

```
users <- users_data(censusKE)
users %>%
  dplyr::select(location) %>%
  unnest_tokens(Location, location) %>%
  count(Location, sort =TRUE) %>%
  top_n(10) %>%
  ggplot(users, mapping = aes(reorder(Location, n), n)) +
    geom_bar(stat = 'identity', aes(fill=Location), show.legend = FALSE) +
    coord_flip() +
    labs(title = "Location of users in #Censuskenya2019 tweets", x="Location", y="Count")
```

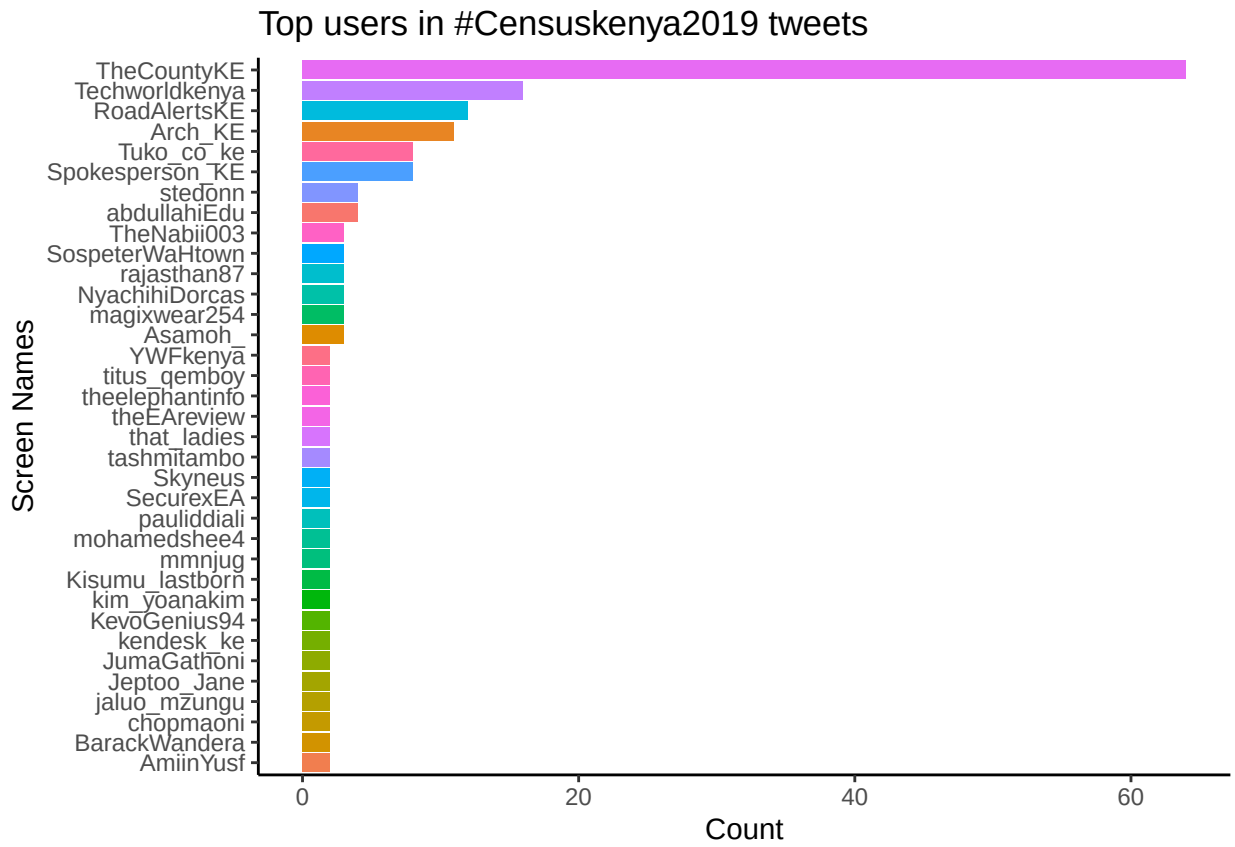
```
## Selecting by n
```




Most users were tweeting #Censususkenya2019 while in Kenya -especially Nairobi.

```
users %>%
  dplyr::select(screen_name) %>%
  count(screen_name, sort = TRUE) %>%
  top_n(15) %>%
  ggplot(users, mapping = aes(reorder(screen_name, n), n)) +
    geom_bar(stat = 'identity', aes(fill=screen_name), show.legend = FALSE)  +
    labs(title="Top users in #Censuskenya2019 tweets", x="Screen Names", y="Count")+
    coord_flip()
```

```
## Selecting by n
```




I was expecting KNBS account to be among the top looks like they've not been active in #Censuskenya2019 tweets.

**Users with verified account in #Censuskenya2019 tweets.**

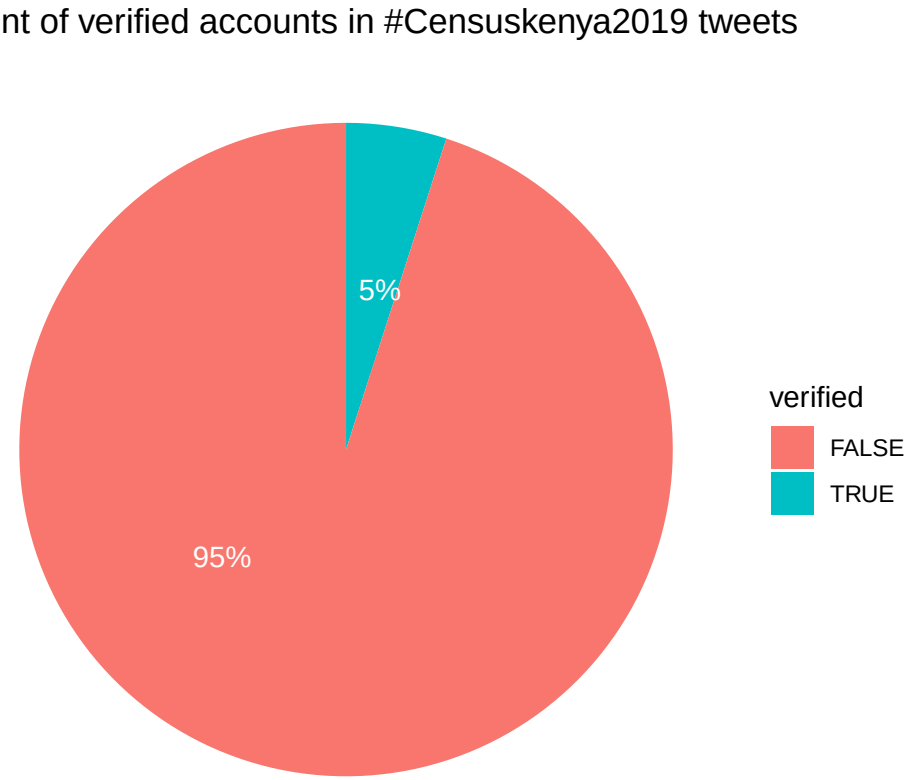
```
table(users$verified)
```

```
##
## FALSE  TRUE
##   287    15
```

Only 15 user accounts are veified in the `users` data object.

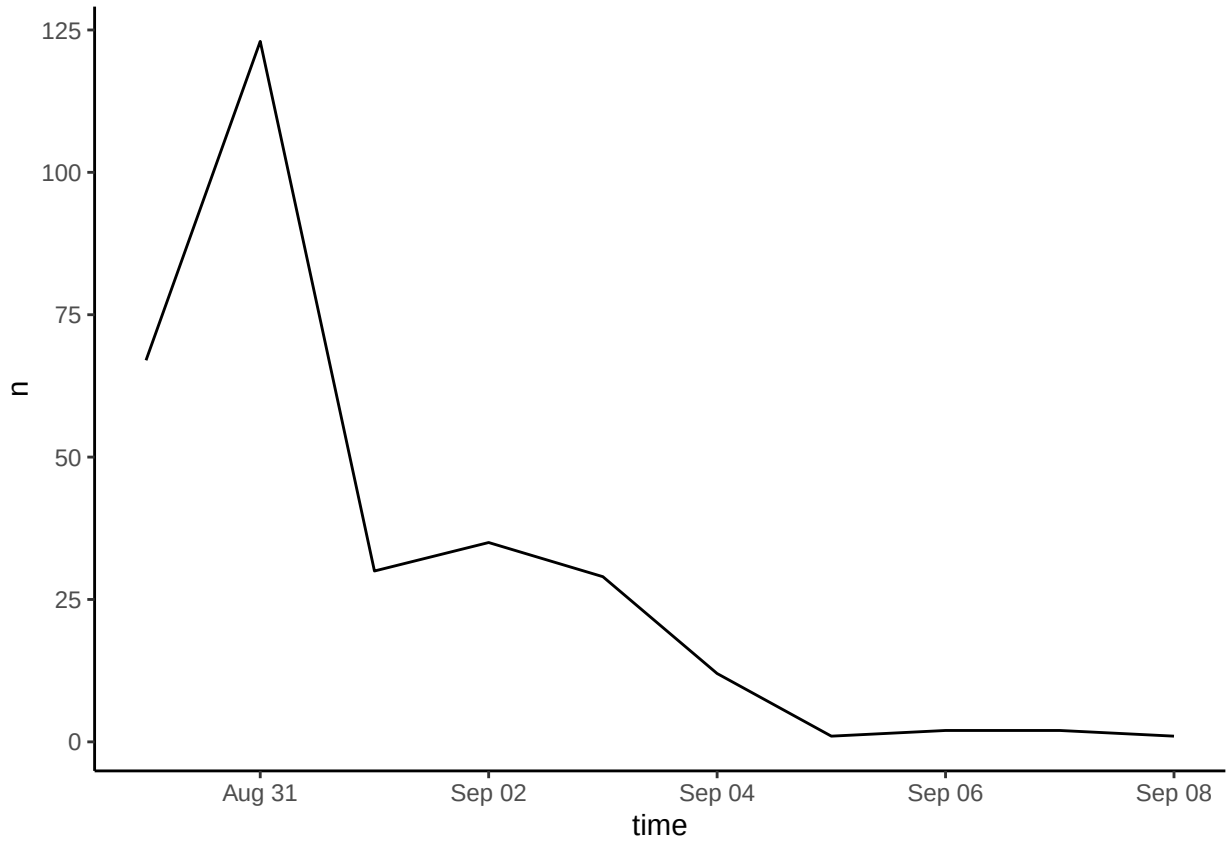
```
users %>%
  count(verified, sort=TRUE) %>%
  mutate(perc = n * 100/nrow(users)) ->verified.users


ggplot(verified.users, aes(x="", y=perc, fill=verified))+
  geom_bar(width =1, stat = 'identity') +
  coord_polar("y", start=0)+
  labs(title = "Count of verified accounts in #Censuskenya2019 tweets")+
  geom_text(aes(y=(0.67*perc), label=sprintf("%0.0f%%", round(perc,2))), color="white")+
  theme_void()
```




**Time series of #Censuskenya2019 tweets**

```
ts_plot(censusKE, by="days")
```




From the time series plot a lot of activity is seen on the last day of Census 2019 in Kenya i.e between 30th and 31st of August 2019.