Exploring citizen-government interactions through analysis of twitter data

Mehdi Khan

December 2, 2017

Project Proposal:

Introduction: My interest in built environment was the reason I studied architecture. As an architect in my early career and now as a GIS/data professional in Planning departments in local government settings I regularly see the importance of data on the success or failure of design and planning decisions. Although the application of data science in business and finance industries etc. are huge, nevertheless, the wave of Bigdata and analytics have hit the field of urban planning too, which is conceptualized and defined by several terms, one of which is "Smart City". The concept of smart city could be explained as data driven city or urban development through the engagement of its four components - the government, the citizens, private businesses and academia. A data science project to measure the engagement and/or relationship between a local government entity and its citizens within the context of urban development or city operations is proposed here.

The problem statement: The project will examine if tweeter messages used by local governments and/or tweeter interactions between the governments and citizens can be used to track the level of involvement of citizens with their government (and vice versa) about urban planning or urban policy issues; and if these interactions can successfully be used to capture and visualize the frustrations or satisfactions of the citizens about various development/policy decisions.

Data source and scope of the project: Tweeter data that were sent by the governments and responses to those messages by the citizens (such as number of retweets, replies etc.) will be used as the primary data sources. Based on the availability of the data, the project will be limited to either one or more local governments or one or more agencies. Private or non-profit entities may be included based on the data availability, relevance and time.

Other consideration: Since urban developments and policies are tied to the use of land with specific boundaries, a spatial component or spatial analysis may be added to the project.

PROJECT DETAILS:

Area of interest and sample data: Howard County, Maryland a jurisdiction of around 300,000 people was selected as the area of interest for this project. Howard County government is active in social media and post messages about government events and news regularly. The diverse citizens with above average education and income were thought to be responsive and concerned about their governments' activities. Therefore, Howard County seemed to be a good candidate for the proposed study.

Although the proposal intended to only examine tweets related to urban planning and urban policy, because of the lack of enough data all tweets were considered.

Project restrictions: Twitter does not allow to access tweets that are more than two weeks old. In addition to that there are also restrictions on how many tweets will be returned by individual functions using twitter API.

Load libraries:

```
suppressWarnings(suppressMessages(library(twitteR)))
suppressWarnings(suppressMessages(library(RCurl)))
suppressWarnings(suppressMessages(library(RJSONIO)))
suppressWarnings(suppressMessages(library(stringr)))
suppressWarnings(suppressMessages(library(rtweet)))
suppressWarnings(suppressMessages(library(dismo)))
suppressWarnings(suppressMessages(library(maps)))
suppressWarnings(suppressMessages(library(ggplot2)))
suppressWarnings(suppressMessages(library(XML)))
suppressWarnings(suppressMessages(library(dplyr)))
suppressWarnings(suppressMessages(library(aws.s3)))
suppressWarnings(suppressMessages(library(aws.signature)))
suppressWarnings(suppressMessages(library(tm)))
suppressWarnings(suppressMessages(library(qdap)))
suppressWarnings(suppressMessages(library(SnowballC)))
suppressWarnings(suppressMessages(library(wordcloud)))
suppressWarnings(suppressMessages(library(topicmodels)))
suppressWarnings(suppressMessages(library(data.table)))
suppressWarnings(suppressMessages(library(tidytext)))
suppressWarnings(suppressMessages(library(RNewsflow)))
suppressWarnings(suppressMessages(library(portfolio)))
suppressWarnings(suppressMessages(library(jsonlite)))
suppressWarnings(suppressMessages(library(readr)))
```

Different libraries were used to access tweets that required authentication and access rights. The project also accessed to AWS to store and read data. All the API keys and tokens were saved as environmental variables that were retrieved when necessary.

Follwing codes were used in datacollection.Rmd but commented out here:

```
# api_key <- Sys.getenv('tweet_api_key') api_secret <-
# Sys.getenv('tweet_api_secret') token <-
# Sys.getenv('tweet_token') token_secret <-
# Sys.getenv('tweet_token_secret') #Create Twitter Connection
# setup_twitter_oauth(api_key, api_secret, token, token_secret)
# app <- Sys.getenv('tweet_app') consumer_key <-
# Sys.getenv('tweet_consumer_key') consumer_secret <-
# Sys.getenv('tweet_consumer_secret') twitter_token <-
# create_token( app = app, consumer_key = consumer_key,
# consumer_secret = consumer_secret)
```

Tweet Analysis of Howard County, Maryland

Using the function lookup_coords in the library 'rtweet' bounding box coordinates of Howard county was collected. The coordinates would be used to filter tweets to find county specific tweets only. Most frequently used twitter accounts by County government were collected from the Howard County website (https://www.howardcountymd.gov/)

Follwing codes were used in datacollection.Rmd but commented out here:

```
# HCcoord <- lookup_coords('Howard County, MD', 'country:US')
# HowardCounty_accounts <-
```

```
# c('HoCoGov','HoCoGovExec','HCPDNews','HCDFRS','HC_JonWeinstein','HoCoBOEMaryland','JenTerrasa')
```

Government twitter accounts were then used to find the associated twitter users and their followers (i.e.the citizens who have interests in government tweets)

The first four statements were used in datacollection.Rmd but commented out here:

```
# hcUsers <- lookupUsers(HowardCounty_accounts) HCfollowers <-
# lapply(hcUsers,function(x) { usr <- x; followersCount(usr) })
# HCfollowersDF <- as.data.frame(HCfollowers)
# write.csv(HCfollowersDF, file = 'HCfollowersDF.csv')

HCfollowersDF <- read.csv(file = "HCfollowersDF.csv", header = TRUE,
    sep = ",", stringsAsFactors = FALSE)

Gov_users <- colnames(HCfollowersDF)
Gov_users <- Gov_users[-1]
followers_count <- as.numeric(as.vector(HCfollowersDF[1, ]))
followers_count <- followers_count[-1]
HCfollowersDF <- data.frame(Gov_users = Gov_users, followers_count = followers_count)
Total_Follower <- sum(HCfollowersDF$followers_count)

HCfollowersDF
```

```
##         Gov_users followers_count
## 1         HoCoGov           12960
## 2     HoCoGovExec            3142
## 3        HCPDNews           99536
## 4          HCDFRS           14614
## 5 HC_JonWeinstein            1311
## 6 HoCoBOEMaryland             366
## 7      JenTerrasa            1351
```

```
ggplot(HCfollowersDF, aes(x = Gov_users, y = followers_count, fill = Gov_users)) +
    geom_bar(stat = "identity") + theme(axis.text.x = element_blank(),
    plot.title = element_text(size = 12, color = "blue", hjust = 0.5)) +
    ggtitle("Number of tweet followers by county accounts")
```

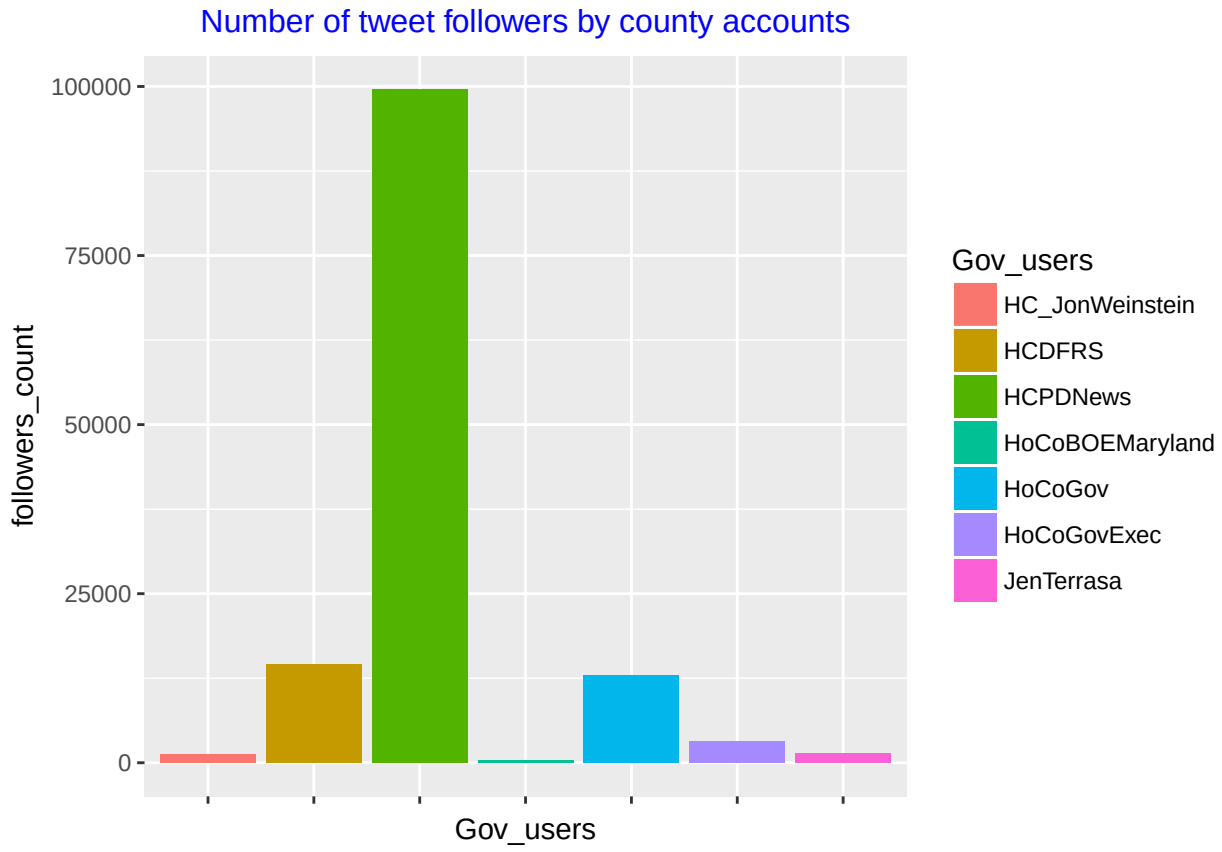

Figure 1.

While the total number (133,166) of Howard County followers are impressive compared to the County population (313,414), a closer look at the data shows that the Police Department (HCPDNews) is an outlier with 99,438 followers. So on the surface it might seem citizens pay close attention to their government while they are concerned about a specfic agency that deals with crime, safety and traffic control. Figure-1 shows the number of follwers following each county accounts.

Further Analysis

Functions were created to collect and evaluate citizens' tweets within the government. The first function "getGov_tweets" takes government accounts (government users) as its parameter and collect the recents tweets sent out by each of those government accounts. It returns all those tweets in a data frame.
the second function "FindHashtags" take the output of the "getGov_tweets" function as its parameter and check all the hashtags used by government accounts. It returns the most common hashtags used by the government. All the hashtags are stored in a character variable seperated by "OR" so that they can be used to search tweets as a query parameter.

Below two functions were used in datacollection.Rmd but also included here for reference:

```
getGov_tweets <- function(x) {
    gdf <- c()
    for (usr in x) {
        gvt <- userTimeline(x[1], n = 150)
        gvdf <- twListToDF(gvt)
        gdf <- rbind(gdf, gvdf)
```

```
    }
    return(gdf)

}


FindHashtags <- function(x) {
    all_hashtags <- str_extract_all(x$text, "#\\w+")
    DF <- as.data.frame(table(tolower(unlist(all_hashtags))))
    mostUsedHashTags <- as.character(DF[order(-DF$Freq)[1:4], 1])
    mostUsedHashTags <- mostUsedHashTags[!is.na(mostUsedHashTags)]
    mostUsed_HashTags <- paste(mostUsedHashTags, sep = "", collapse = " OR ")

    return(mostUsed_HashTags)
}
```

Tweets sent by Howard County, MD and its citizens:

search_tweets function of rtweet library was used to collect the citizen tweets. In order to select the tweets that were possibly generated as responds/reactions to government tweets, the most recent common hashtags used by the Howard County government and to control the citizen locations, the bounding box (coordinates of opposite corner points of the rectangle that contains the county polygon) of the County were used as query parameter. user_data function returned the users information of all the tweets. The citizens tweets were seperated from government tweets by comparing the users_id of the tweets.

Government tweets at a glance:

The first two statements were used in datacollection.Rmd but commented out here:

```
# HCgov_tweetDF <- getGov_tweets(hcUsers) write.csv(HCgov_tweetDF,
# file='HCgov_tweetDF.csv')

HCgov_tweetDF <- read.csv("HCgov_tweetDF.csv")
HC_retweet_count <- sum(HCgov_tweetDF$retweetCount)
HC_tweets_retwweted <- nrow(filter(HCgov_tweetDF, !HCgov_tweetDF$retweetCount ==
    0))

HC_favorite_count <- sum(HCgov_tweetDF$favoriteCount)
HC_tweets_favorited <- nrow(filter(HCgov_tweetDF, !HCgov_tweetDF$favoriteCount ==
    0))

total_count <- nrow(HCgov_tweetDF)

category <- c("total tweet", "retweet_count", "retweeted_tweet", "favorite_count",
    "favorited_tweet")
tweet_count <- c(total_count, HC_retweet_count, HC_tweets_retwweted,
    HC_favorite_count, HC_tweets_favorited)
id <- c(1:5)
likedTweetDF <- data.frame(id, category, tweet_count)

ggplot(likedTweetDF, aes(x = category, y = tweet_count, fill = category)) +
    geom_bar(stat = "identity") + geom_text(aes(label = tweet_count),
```

```
    vjust = 1.6, color = "white", position = position_dodge(0.9),
    size = 3.5) + scale_fill_brewer(palette = "Paired") + theme(axis.text.x = element_blank(),
    plot.title = element_text(size = 12, color = "blue", hjust = 0.5)) +
    ggtitle("Number of tweets that were retweeted or liked  \n and the counts of retweet and liked (fav
```

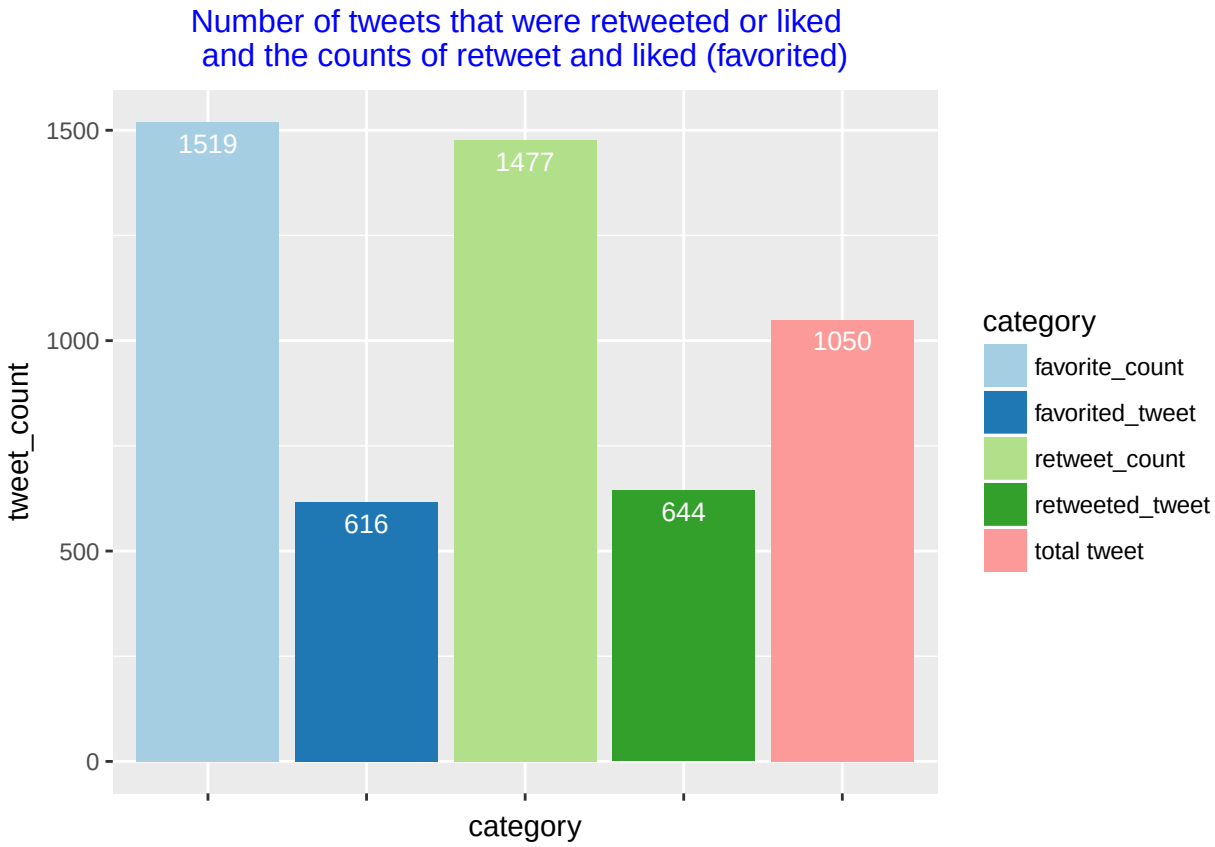

Figure 2

Above figure (Figure 2) shows a good number of government tweets were liked and retweeted. Out of 1050 original tweets, 644 and 616 were retweeted and favorited a total of 1477 and 1519 times respectively. The statisctics here suggest a good response to government tweets.

Frequency of government tweets:

```
HCgov_tweetDF$created <- as.Date.character(HCgov_tweetDF$created)
ggplot(HCgov_tweetDF, aes(x = created, fill = "red", col = "blue",
    alpha = 0.2)) + geom_histogram(position = "identity", bins = 20,
    show.legend = FALSE) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("Frequency of government tweets")
```

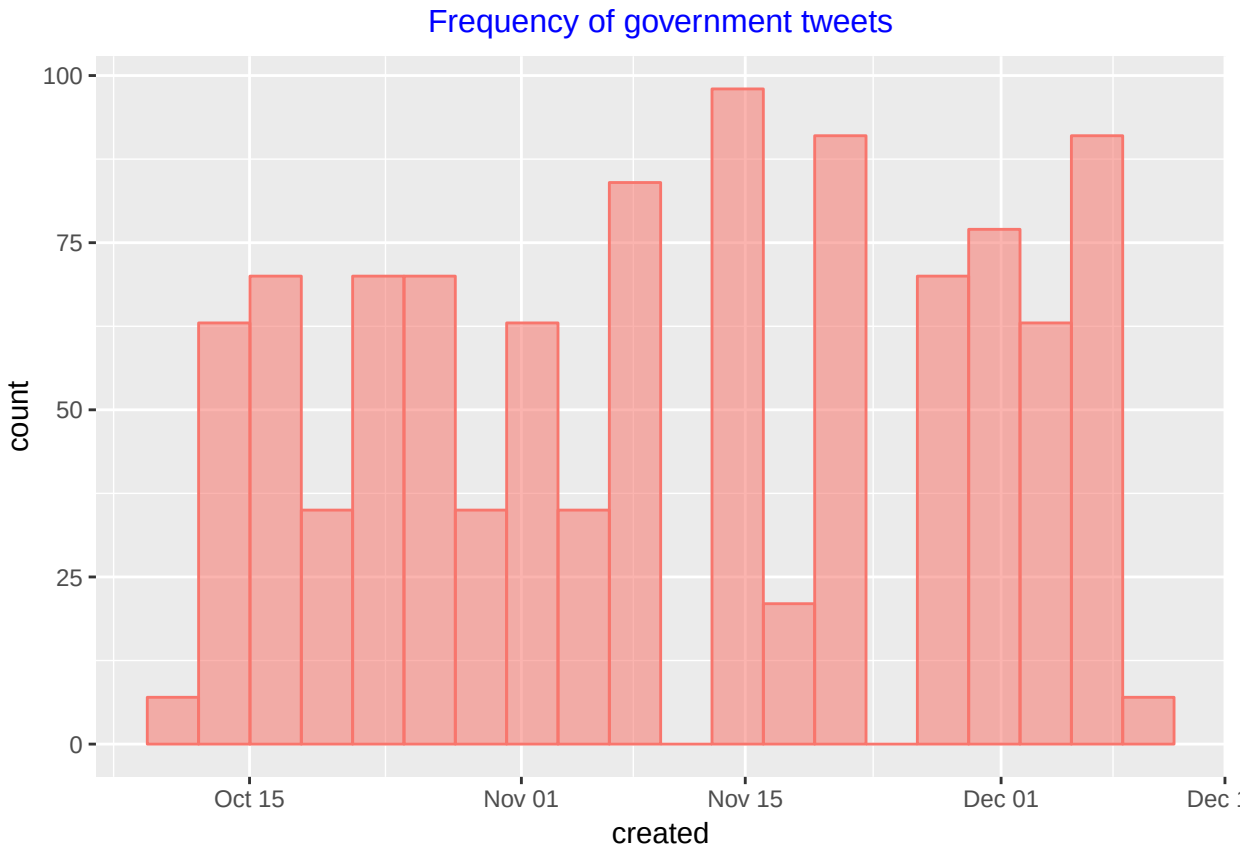

Figure 3.

Figure-3 shows that tweet frequencies i.e. tweets sent out by the county government every week are almost consistent.

collecting and evaluating citizen tweets:

Howrd county general tweets and citizen user-ids were collected using the below statements in datacollection.Rmd, which were commented out here:

```
hocogov_hashtags = FindHashtags(HCgov_tweetDF)
print(hocogov_hashtags)
```

```
## [1] "#hocomd OR #hocopolice OR #columbiamd OR #ellicottcitymd"
```

```
# HowardCounty_genTweets <- search_tweets( hocogov_hashtags,
# n=2000, token=twitter_token, type = 'mixed' )

HowardCounty_genTweets <- read.csv(file = "HCgen_tweetDF.csv", header = TRUE,
    sep = ",", stringsAsFactors = FALSE)


# HCgovUsersid <- sapply(hcUsers,function(x) x$id ) HCcitizens <-
# users_data(HowardCounty_genTweets) HCcitizens <-
# HCcitizens[!HCcitizens$location=='',] HCcitizens <-
# HCcitizens[!HCcitizens$user_id %in% HCgovUsersid,]
```

```
HCcitizens <- read.csv(file = "HCcitizens.csv", header = TRUE, sep = ",",
    stringsAsFactors = FALSE)
```

Connecting systems in real time:

The intention of the project was also to be able to share data with other systems, particularly with GIS so that various spatial analysis could be done with the tweet data. Two seperate cloud based systems were explored. Tweet data with location information were direcly stored to AWS (Amazon Web Service), which were consumed by ArcGIS online (an ESRI based cloud GIS) in order to analyze and visualuize data spatially in conjunction with other spatial data. Thus, all the changes could be updated and reflected across the systems real or near real time.

While it was possible to geocode data in ESRI platform, the geocode capability of 'dismo' library was experimented with 'geocode' function, which uses Google API. Note that the geocode operation here was limited due to the restrictions on free version of Google API.

The following geocode operations were done in datacollection.Rmd but commented out here:

```
# locations <- geocode(HCcitizens$location) locations <-
# na.omit(locations) locations <- filter(locations,
# !locations$longitude < -77.18711 & !locations$longitude >
# -76.69732) write.csv(locations, file='locate2.csv')
```

The mapping capabilities in R (ggplot2) was also experimented, which was found to be very limited (see the commented out code snippet that was found in 'https://gist.github.com/dsparks/4329876')

```
## MAPPING in R with(locations, plot(longitude, latitude)) worldMap
## <- map_data('county','maryland', wrap = TRUE)

# zp1 <- ggplot(worldMap) zp1 <- zp1 + geom_path(aes(x = long, y =
# lat, group = group), # Draw map colour = gray(2/3), lwd = 1/3)
# zp1 <- zp1 + geom_point(data = locations, # Add points
# indicating users aes(x = locations$longitude, y =
# locations$latitude), colour = 'RED', alpha = 1/2, size = 1) zp1
# <- zp1 + coord_equal() # Better projections are left for a
# future post zp1 <- zp1 + theme_minimal() # Drop background
# annotations print(zp1)
```

Exporting data into AWS (using 'aws.s3'library), the file can be accessed by the following link: https://s3.amazonaws.com/khdata/locate.csv The below HTML snippet can be used to view the map that was created based on locations data that was exported to AWS:

```
# <style>.embed-container {position: relative; padding-bottom:
# 75%; height: 0; max-width: 100%;} .embed-container iframe,
# .embed-container object, .embed-container iframe{position:
# absolute; top: 0; left: 0; width: 100%; height: 100%;}
# small{position: absolute; z-index: 40; bottom: 0; margin-bottom:
# -15px;}</style><div class='embed-container'><iframe width='400'
# height='300' frameborder='0' scrolling='no' marginheight='0'
# marginwidth='0' title='data607'
# src='//data607.maps.arcgis.com/apps/Embed/index.html?webmap=592b2fa442044589aacad05f7aafa313&amp;exte
```

the map can also be accessed by the below link: https://arcg.is/1ebujL

The following statements were used in datacollection.Rmd to stored data in AWS but commented out here:

```
b <- get_bucket("khdata")
# s3write_using(locations,FUN = write.csv, object = 'locate.csv',
# bucket = b )
```

geocoded data is also used to further filter the citizen users to make sure that the location of the users are in fact in and aroud Howard County and the tweets were originated by the Howard County residents and/or stake holders:

```
obj <- get_object(object = "locate.csv", bucket = b)  # getting data from AWS
locations <- read.csv(text = rawToChar(obj))

HCcitizens <- HCcitizens[HCcitizens$location %in% locations$originalPlace,
    ]


HowardCounty_citizensTweets <- HowardCounty_genTweets[HowardCounty_genTweets$user_id %in%
    HCcitizens$user_id, ]
```

Frequency of citizen tweets:

```
HowardCounty_citizensTweets$created_at <- as.Date(HowardCounty_citizensTweets$created_at)
ggplot(HowardCounty_citizensTweets, aes(x = created_at, fill = "green",
    col = "blue", alpha = 0.2)) + geom_histogram(position = "identity",
    bins = 20, show.legend = FALSE) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("Frequency of citizen tweets")
```

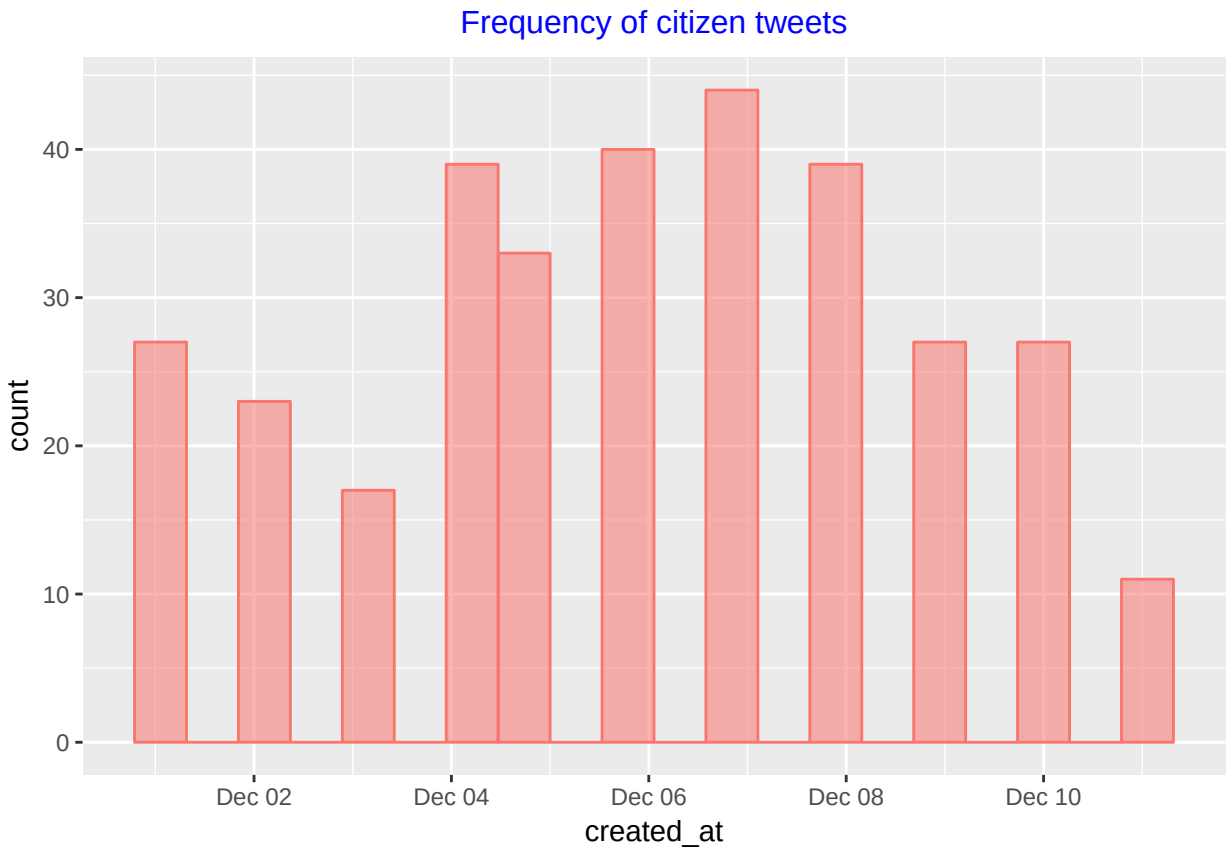

Figure 4.

Figure-4 shows that tweet frequencies of citizens varies a lot as oppose to the consistent nature government

tweet frequencies.

Tweet Text mining

Texts were analyzed to see if similar terms are common in both government and citizens tweets. in other word texts were explored to examine if the concerns and interests of citizens match with what government wanted to talk about, or how much of the concerns of the both groups overlapped.

A function was created clean a Corpus that would be created with tweet texts:

```
cleanCorp <- function(corp) {


    corp <- tm_map(corp, str_replace_all, "<[^>]+>", "")
    corp <- tm_map(corp, str_replace_all, "@\\w+", "")
    corp <- tm_map(corp, str_replace_all, "#\\w+", "")
    corp <- tm_map(corp, str_replace_all, "http\\w+", "")
    corp <- tm_map(corp, content_transformer(removePunctuation))

    # since in tweet people tend to abbrebriate and symbolize texts
    # the following three functions were used from qdab library
    corp <- tm_map(corp, content_transformer(replace_abbreviation))
    corp <- tm_map(corp, content_transformer(replace_contraction))
    corp <- tm_map(corp, content_transformer(replace_symbol))
    corp <- tm_map(corp, removeNumbers)

    corp <- tm_map(corp, content_transformer(tolower))
    corp <- tm_map(corp, PlainTextDocument)
    corp <- tm_map(corp, stripWhitespace)
    corp <- tm_map(corp, str_replace_all, "^ ", "")
    corp <- tm_map(corp, str_replace_all, " $", "")

    # corp <- tm_map(corp, content_transformer(stemDocument))
    corp <- tm_map(corp, removeWords, stopwords("english"))
    return(corp)
}




HCgov_corpus <- Corpus(VectorSource(HCgov_tweetDF$text))
HCgov_corpus <- cleanCorp(HCgov_corpus)

HCcitizen_corpus <- Corpus(VectorSource(HowardCounty_citizensTweets$text))
HCcitizen_corpus <- cleanCorp(HCcitizen_corpus)


HCgov_corpus <- tm_map(HCgov_corpus, str_replace_all, "md|pm|am",
    "")
HCgov_corpus <- tm_map(HCgov_corpus, removeWords, stopwords("english"))
frequent_terms <- freq_terms(HCgov_corpus$content, 30)
plot(frequent_terms)
```

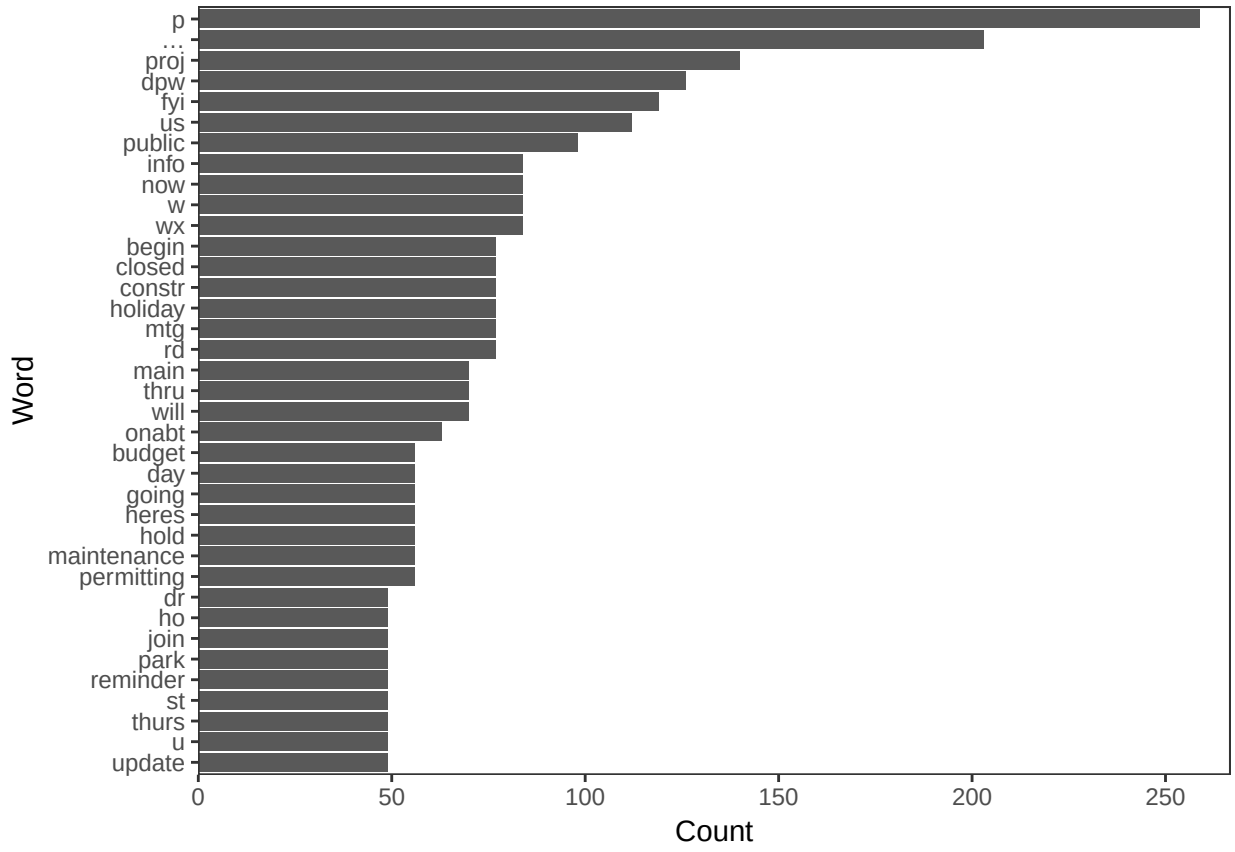

Figure 5.

```
HCcitizen_corpus <- tm_map(HCcitizen_corpus, str_replace_all, "md|pm|am",
    "")
HCcitizen_corpus <- tm_map(HCcitizen_corpus, removeWords, stopwords("english"))
citizen_frequent_terms <- freq_terms(HCcitizen_corpus$content, 30)
plot(citizen_frequent_terms)
```

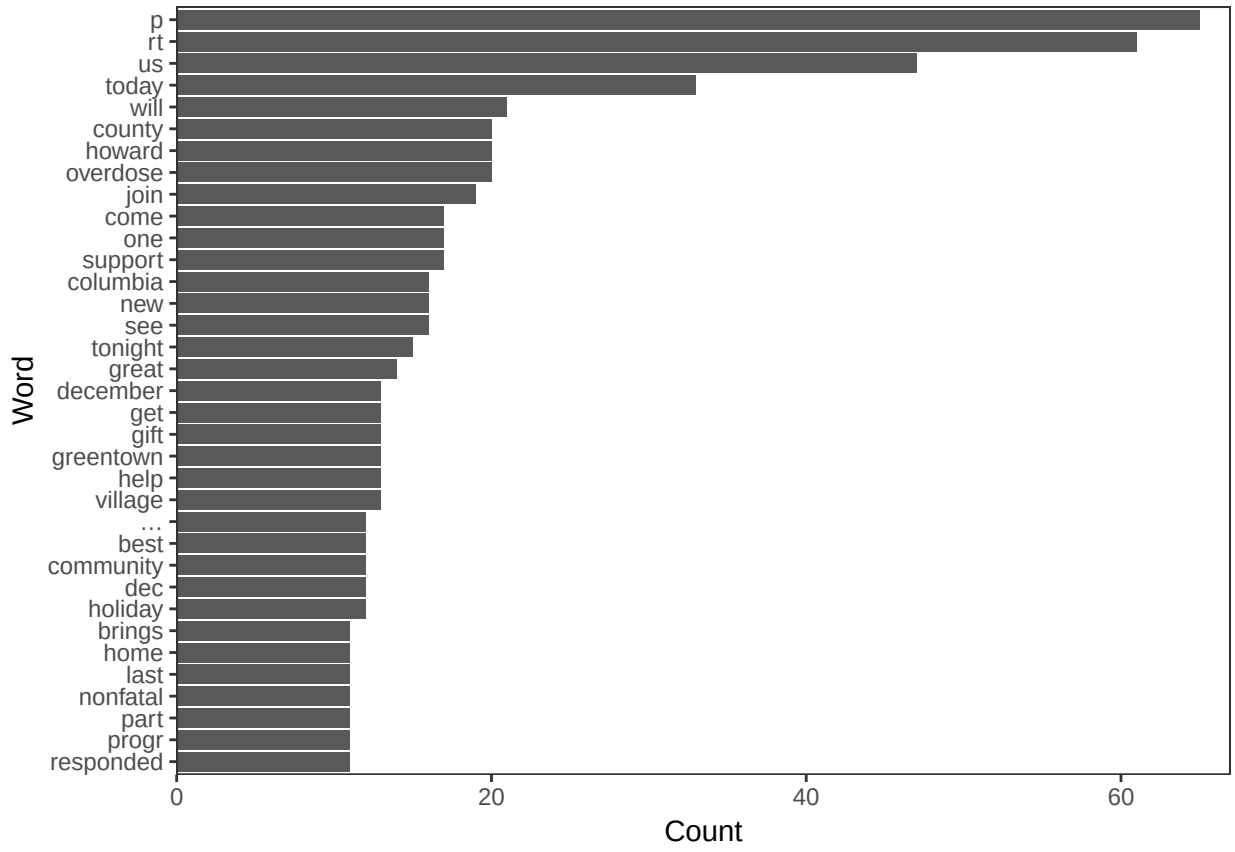

Figure 6.

Above two plots (Figure 5 and 6) show 30 most frequently used terms in tweets sent out by the government and the citizens. No significant match are seen between these two sets of words (terms), which suggest a very low overlapping of common discussions between citizens and governments.

Creating Term-document Matrix:

```
HCgov_tdm <- TermDocumentMatrix(HCgov_corpus)
HCcitizen_tdm <- TermDocumentMatrix(HCcitizen_corpus)



HCgov_tdm <- removeSparseTerms(HCgov_tdm, 0.99)
HCcitizen_tdm <- removeSparseTerms(HCcitizen_tdm, 0.99)

print(HCgov_tdm)
```

```
## <<TermDocumentMatrix (terms: 223, documents: 1050)>>
## Non-/sparse entries: 5600/228550
## Sparsity           : 98%
## Maximal term length: 12
## Weighting          : term frequency (tf)
```

```
print(HCcitizen_tdm)
```

```
## <<TermDocumentMatrix (terms: 191, documents: 327)>>
```

```
## Non-/sparse entries: 1291/61166
## Sparsity           : 98%
## Maximal term length: 14
## Weighting          : term frequency (tf)
```

Evaluation through Dendrograms:

Dendrograms were drawn for both government and citizens to see if they provide any interesting insights by creating clusteres based on word similarities.

```
drawDendogram <- function(x) {
    df <- as.data.frame(inspect(x))
    df_scale <- scale(df)
    d <- dist(df_scale, method = "euclidean")
    fit <- hclust(d, method = "ward.D2")
    return(fit)
}
```

Dendrogram for Howard County government tweets

```
HCgovDendo <- drawDendogram(HCgov_tdm)
```

```
## <<TermDocumentMatrix (terms: 223, documents: 1050)>>
## Non-/sparse entries: 5600/228550
## Sparsity           : 98%
## Maximal term length: 12
## Weighting          : term frequency (tf)
## Sample             :
##          Docs
## Terms     1015 112 115 265 415 565 715 76 865 89
##   begin      0   0   0   0   0   0   0  0   0  0
##   closed     0   0   0   0   0   0   0  0   0  0
##   constr     0   0   0   0   0   0   0  0   0  0
##   dpw        1   0   1   1   1   1   1  1   1  1
##   fyi        0   0   0   0   0   0   0  0   0  0
##   holiday    0   0   0   0   0   0   0  0   0  0
##   info       1   1   1   1   1   1   1  1   1  1
##   now        1   0   1   1   1   1   1  1   1  0
##   proj       0   0   0   0   0   0   0  0   0  0
##   public     0   1   0   0   0   0   0  0   0  0
```

```
plot(HCgovDendo)
rect.hclust(HCgovDendo, k = 4)
```

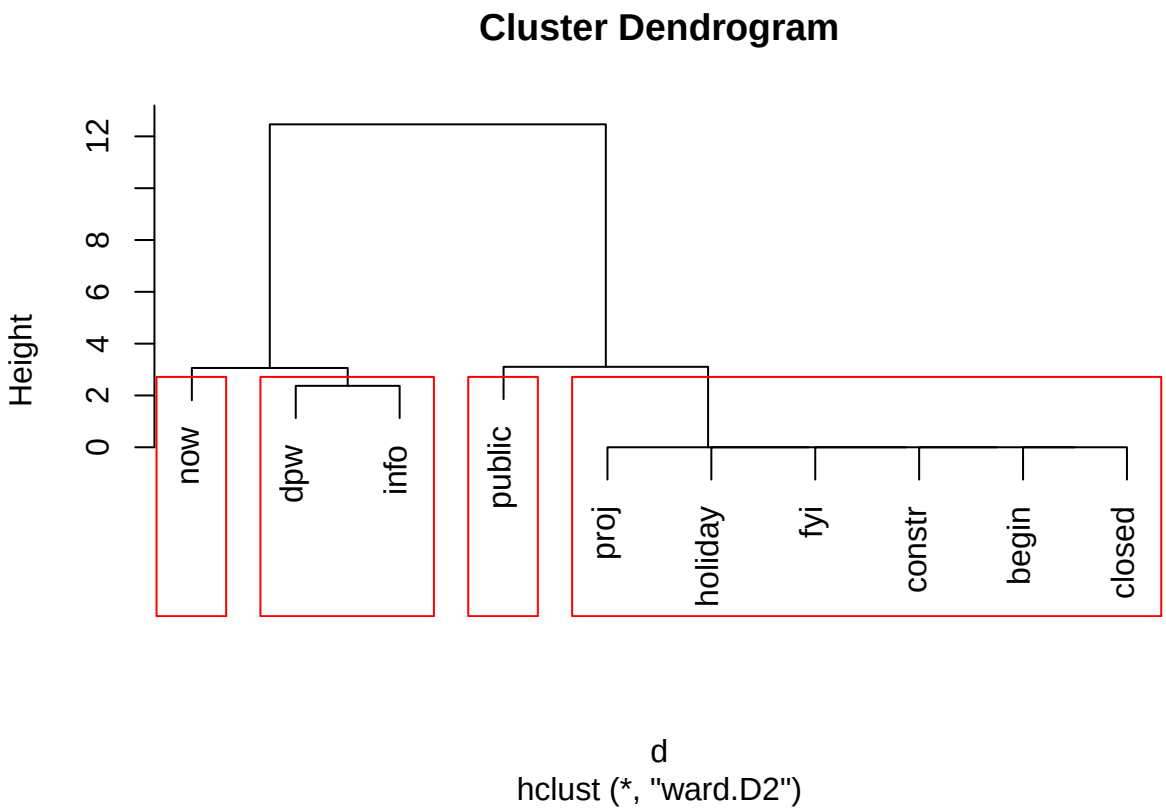

Figure 6.

Dendrogram for Howard County citizens tweets

```
HCcitizenDendo <- drawDendogram(HCcitizen_tdm)
```

```
## <<TermDocumentMatrix (terms: 191, documents: 327)>>
## Non-/sparse entries: 1291/61166
## Sparsity           : 98%
## Maximal term length: 14
## Weighting          : term frequency (tf)
## Sample             :
##           Docs
## Terms      171 196 198 25 26 269 270 29 30 57
##   columbia   0   0   0  0  0   0   0  0  0  0
##   come       0   0   0  0  0   0   0  1  1  0
##   county     1   0   0  0  0   0   0  0  0  0
##   howard     0   0   0  0  0   0   0  0  0  0
##   join       1   0   0  1  1   1   1  1  1  0
##   one        0   0   1  0  0   1   1  0  0  0
##   overdose   0   0   0  0  0   0   0  0  0  0
##   support    1   1   0  0  0   0   0  0  0  1
##   today      0   0   1  1  1   1   1  1  1  0
##   will       0   1   0  1  1   0   0  1  1  1
```

```
plot(HCcitizenDendo)
```

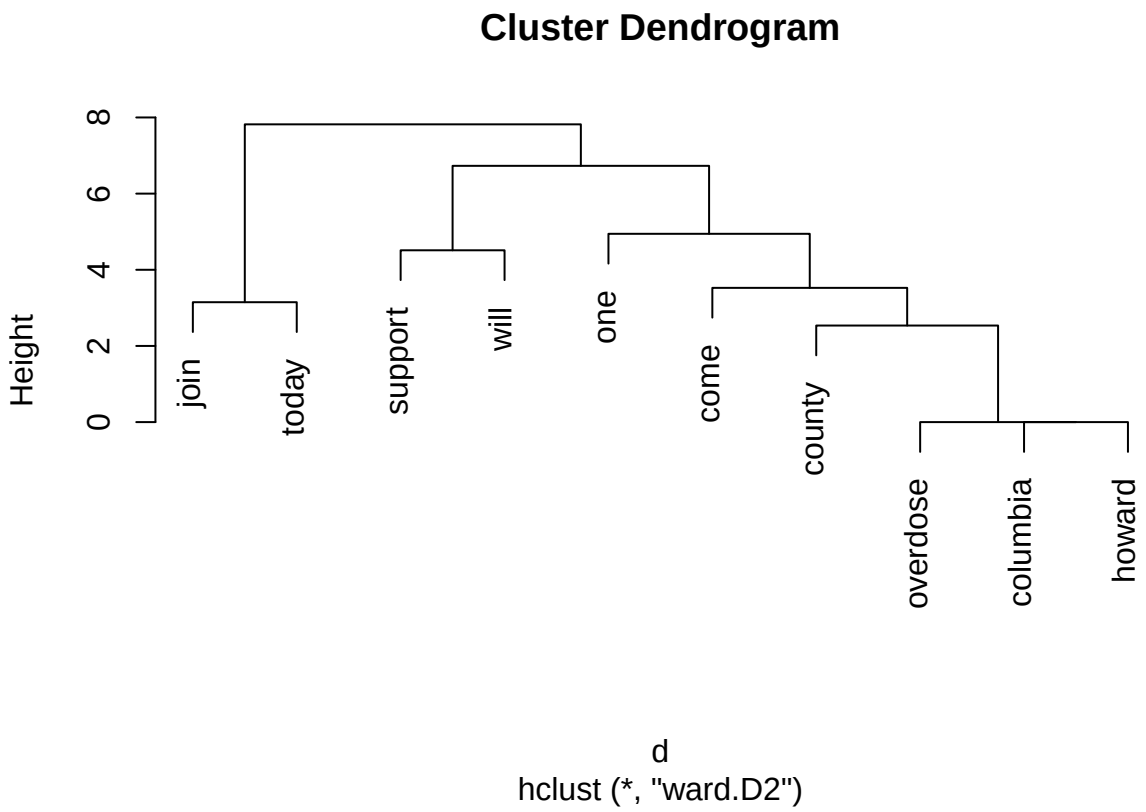

Figure 7.

The government dendogram (Figure 6) shows some association of the words that were used in their tweets. There were no association of words or no distinct clusters in the citizens tweets (Figure 7) suggesting no focused discussion on certain topics but many scattered interests.

Evaluation through Wordclouds:

In order to see the difference or commonality of interests or concerns of these two groups (government and citizens) two wordclouds were created. All the texts of each group were represented in two documents representing the government and the citizens in a common Corpus:

```
try.tolower = function(x) {
    y = NA
    try_error = tryCatch(tolower(x), error = function(e) e)
    if (!inherits(try_error, "error"))
        y = tolower(x)
    return(y)
}

HcgovText <- paste(unlist(HCgov_tweetDF$text), sep = " ", collapse = " ")
HccitizenText <- paste(unlist(HowardCounty_citizensTweets$text), sep = " ",
    collapse = " ")
HccitizenText <- sapply(HccitizenText, function(row) iconv(row, "latin1",
    "ASCII", sub = ""))

HcgovText <- sapply(HcgovText, try.tolower)
```

```
HccitizenText <- sapply(HccitizenText, try.tolower)

# HcgovText <- paste(HcgovText, collapse=' ') HccitizenText <-
# paste(HccitizenText, collapse=' ') HccitizenText <-
# as.character(HccitizenText)
HCtexts <- c(HcgovText, HccitizenText)

HC_Corpus <- Corpus(VectorSource(HCtexts))
HC_Corpus <- cleanCorp(HC_Corpus)
HC_Corpus <- tm_map(HC_Corpus, removePunctuation)
HC_Corpus <- tm_map(HC_Corpus, content_transformer(stemDocument))

HC_tdm <- TermDocumentMatrix(HC_Corpus)
HC_tdm <- as.matrix(HC_tdm)
colnames(HC_tdm) = c("Government Tweets", "Citizent Tweets")
```

Comarison cloud:

```
comparison.cloud(HC_tdm, colors = c("#00B2FF", "red"), title.size = 1,
    max.words = 200, scale = c(2.1, 0.49))
```

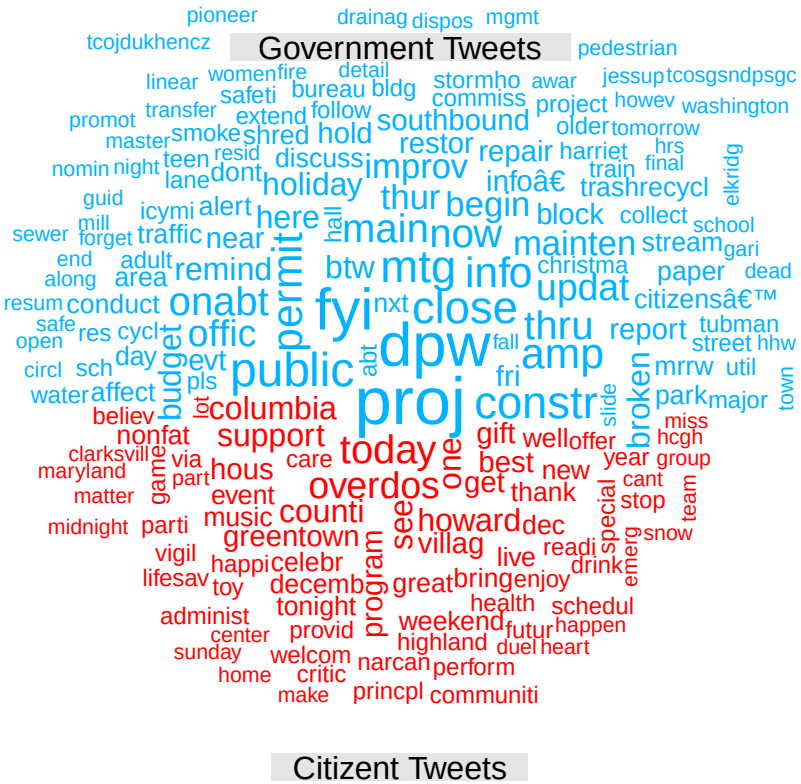

Figure 8.

The above cloud (Figure 8) suggests some relevant words concerning governmnet operations, such as project, DPW (public works), meeting, permit, construction, improvment, public, repair, maintenance etc. on the other hand citizen tweets seems very diverse and nothing really stands out i.e. no suggestion of any interaction between government and citizens,

Commonality Cloud:

```
commonality.cloud(HC_tdm, colors = brewer.pal(8, "Dark2"))
```

```
## Warning in wordcloud(rownames(term.matrix)[freq > 0], freq[freq > 0],
## min.freq = 0, : amp could not be fit on page. It will not be plotted.
```

Figure 9.

The commonality cloud (Figure 9) again suggest disconnect between the two groups. The common words found in the cloud such as amp, join, holiday, tonight, work etc. are very general and does not seem to suggest any interaction between the government and the citizens.

Topic model comparison:

Both government and citizens tweet texts were grouped under five topics each, and the 10 most frequent terms related to each topics were plotted to examine if there were any similarities between the topics and terms that would suggest any interaction:

Goverment topics:

```
HCgov_DTM <- as.DocumentTermMatrix(HCgov_tdm)

# HCgov_DTM_DS <- as.matrix(HCgov_DTM)
rowTotals <- apply(HCgov_DTM, 1, sum)  #Find the sum of words in each Document
HCgov_DTM <- HCgov_DTM[rowTotals > 0, ]
HCgov_DTM_DS <- as.matrix(HCgov_DTM)
ldamodel <- LDA(HCgov_DTM, k = 5, control = list(seed = 1500))
```

```
## topicwords <- terms(ldamodel,5) topicwords

gov_per_topic_per_word <- tidy(ldamodel, matrix = "beta")
head(gov_per_topic_per_word)
```

```
## # A tibble: 6 x 3
##   topic        term          beta
##   <int>       <chr>         <dbl>
## 1     1      budget 6.028790e-177
## 2     2      budget 1.049166e-177
## 3     3      budget  1.404210e-42
## 4     4      budget 1.501938e-172
## 5     5      budget  6.003521e-02
## 6     1 citizensâ<U+0080><U+0099> 1.195581e-176
```

```
gov_top_terms <- gov_per_topic_per_word %>% group_by(topic) %>% top_n(10,
    beta) %>% ungroup() %>% arrange(topic, -beta)

gov_top_terms %>% mutate(term = reorder(term, beta)) %>% ggplot(aes(term,
    beta, fill = factor(topic))) + geom_col(show.legend = FALSE) +
    facet_wrap(~topic, scales = "free") + coord_flip() + theme(plot.title = element_text(size = 11,
    color = "blue", hjust = 0.5)) + ggtitle("Most Frequent terms in government tweets \n catagorized un
```

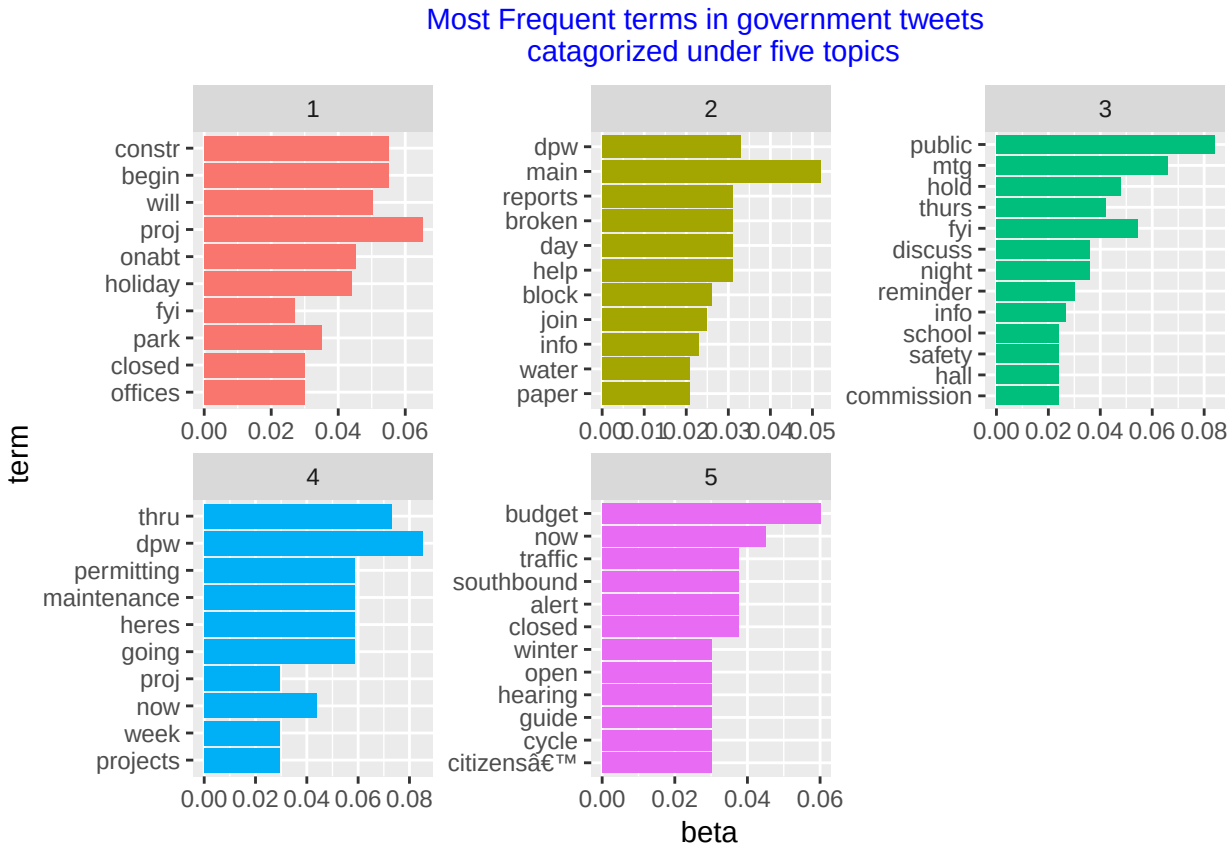

Figure 10.

Citizen topics:

```
HCcitizen_DTM <- as.DocumentTermMatrix(HCcitizen_tdm)

# HCgov_DTM_DS <- as.matrix(HCgov_DTM)
rowTotal <- apply(HCcitizen_DTM, 1, sum)  #Find the sum of words in each Document
HCcitizen_DTM <- HCcitizen_DTM[rowTotal > 0, ]

ctznldamodel <- LDA(HCcitizen_DTM, k = 5, control = list(seed = 1500))
## topicwords <- terms(ldamodel,5) topicwords

ctzn_per_topic_per_word <- tidy(ctznldamodel, matrix = "beta")
head(ctzn_per_topic_per_word)
```

```
## # A tibble: 6 x 3
##   topic     term         beta
##   <int>    <chr>        <dbl>
## 1     1 columbia 1.011136e-10
## 2     2 columbia 1.826983e-04
## 3     3 columbia 5.346606e-02
## 4     4 columbia 2.048270e-21
## 5     5 columbia 1.519486e-23
## 6     1     soon 1.336023e-02
```

```
ctzn_top_terms <- ctzn_per_topic_per_word %>% group_by(topic) %>%
    top_n(10, beta) %>% ungroup() %>% arrange(topic, -beta)

ctzn_top_terms %>% mutate(term = reorder(term, beta)) %>% ggplot(aes(term,
    beta, fill = factor(topic))) + geom_col(show.legend = FALSE) +
    facet_wrap(~topic, scales = "free") + coord_flip() + theme(plot.title = element_text(size = 11,
    color = "blue", hjust = 0.5)) + ggtitle("Most Frequent terms in citizen tweets \n catagorized under
```

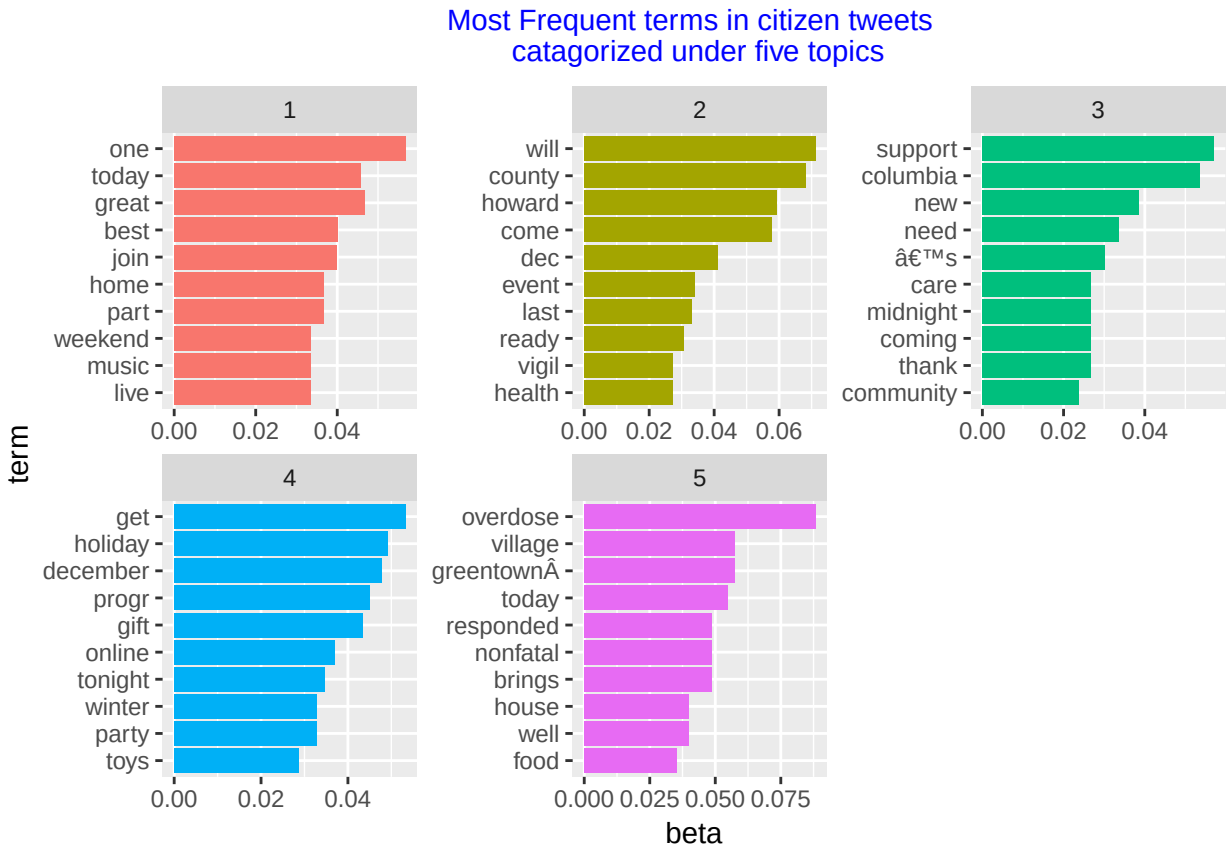

Figure 11.

All the above topics and related terms in government and citizen tweets (Figure 10 and 11) do not show any similarities re-affirm the suspicion that there is not enough citizens-government interactions through tweets.

Compare the document-term-matrix of government and citizen tweets:

```
govDTM <- DocumentTermMatrix(HCgov_corpus, control = list(weighting = weightTfIdf,
    stopwords = TRUE))
ctznDTM <- DocumentTermMatrix(HCcitizen_corpus, control = list(weighting = weightTfIdf,
    stopwords = TRUE))

doc_compare <- documents.compare(ctznDTM, govDTM, min.similarity = 0.45,
    n.topsim = NULL, return.zeros = FALSE)
```

```
## Warning in colnames(dtm) == colnames(dtm.y): longer object length is not a
## multiple of shorter object length
```

```
head(doc_compare)
```

```
##       x  y similarity
## 7   146  1  0.6628568
## 72   91  7  0.6625379
## 73   99  7  0.4957166
## 173 140 11  0.7022255
## 252 146 15  0.8417213
## 267 202 16  0.8663407
```

```
similar <- nrow(doc_compare)
gov_documents <- nrow(govDTM)
citizen_documents <- nrow(ctznDTM)

doc_count <- c(citizen_documents, gov_documents, similar)

CompareDF <- data.frame(documents = c("citizen", "government", "similar"),
    doc_count = doc_count)

ggplot(CompareDF, aes(x = documents, y = doc_count, fill = documents)) +
    geom_bar(stat = "identity", color = "black") + scale_fill_manual(values = c("#999999",
    "#E69F00", "#56B4E9")) + geom_text(aes(label = doc_count), vjust = 1.6,
    color = "white", position = position_dodge(0.9), size = 3.5) +
    scale_fill_brewer(palette = "Paired") + theme(axis.text.x = element_blank(),
    plot.title = element_text(size = 11, color = "blue", hjust = 0.5)) +
    ggtitle("Number of government and citizen documents (tweets) \n and the number of documents with at
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

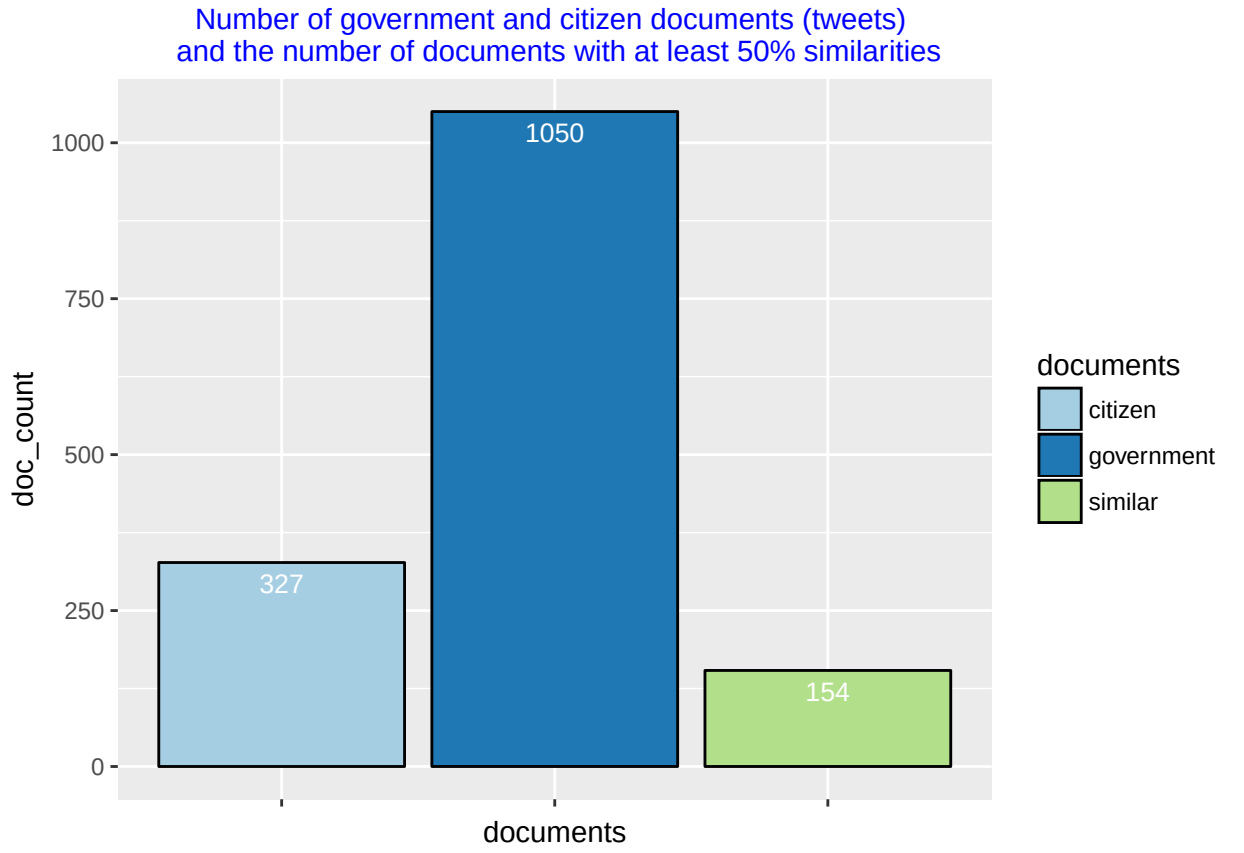

Figure 12.

While it seems significant to find 210 pair of documents have some significant similarities (see Figure 12) given the low number of citizen documents (only 351), the reason probably is that same citizen documents might have matched with multiple government documents. It could be the similarities among very general words as seen in the commonality cloud above. So the result of this document comparison process does not necessarily show any interaction between two the two groups.

Summary:

All the analysis above point to the fact that both the County government and the citizens need to take initiatives for effective communications. The number of tweets sent out the by the government and the number of totals followers they have are encouraging, which suggest both the willingness and environment are there to use social media such as tweeters for better communication between Howard County government and its citizens. The huge number of followers of Police Department means people, in general, are naturally drawn to stories or news that have quick and explicit impact on them such as a crime event or accidents. Therefore departments like Planning and zoning etc. that have significant influence on citizens' future livelihoods but are not immediately felt should be more proactive to connect to the citizens.

Reference:

1. http://rstudio-pubs-static.s3.amazonaws.com/256588_57b585da6c054349825cba46685d8464.html
2. http://tidytextmining.com/twitter.html#getting-the-data-and-distribution-of-tweets
3. https://heuristically.wordpress.com/2011/04/08/text-data-mining-twitter-r/
4. http://fredgibbs.net/tutorials/document-similarity-with-r.html
5. https://sites.google.com/site/miningtwitter/home
6. https://developer.twitter.com/en/docs/basics/getting-started
7. http://bogdanrau.com/blog/collecting-tweets-using-r-and-the-twitter-search-api/
8. https://davetang.org/muse/2013/04/06/using-the-r_twitter-package/
9. https://rstudio-pubs-static.s3.amazonaws.com/66739_c4422a1761bd4ee0b0bb8821d7780e12.html http://tidytextmining.com/topicmodeling.html