Hurricane Harvey in America

xueting tao

introduction

Hurricane Harvey is an Atlantic hurricane that has attack America for several times in the past year. The most recent attack happened on August 13 when the National Hurricane center detect the hurricane on the western east of Africa. It hit United States on Texas and Louisiana on August 23rd. In this project, I am going to use the information from twitter to identify possible the most serious hurt area and its possible moving route.

Method:

1. Basic assumption:

1.1. The severity of the hurricane affects the times people tweets.

1.2. The location information in the location column the tweet original display is accurate and represent the people's real location.

1.3. The missing and useless data is random.

2. Data extraction:

2.1 The data comes from twitter from Sep 4th to Sep 8th. For each day, I pooled the 10,000 tweets data from the API(except for Sep 4th) and use the location information as the basic data source to identify the path of hurricane and use the number of the same location occur as the severity of the hit.

2.2 Follow the instruction, first I need to create a Twitter Aplication:

I create a new application with the name of xtao8 and get the API key, API secreat, Access token and Access secret.

2.3 Install twitterR. Since I would like to access Harvey and it mainly start on August 17th at night, I would like to start my search from Sep 4th and follow until Sep 8th. If the data set is too big, it would be difficult to analysis. So I would like to pull the first 10,000 sample several times and get useful information from them.

The code I used:

```
#getting the data: install twitterR since install.pacakge("twitterR")is not avialble
#let me try another one
install.packages(c("devtools", "rjson", "bit64", "httr"))
#follow the instruction and restart R
library(devtools)
install_github("geoffjentry/twitteR")

library(twitteR)
setup_twitter_oauth("API key",
                    "API secret",
                    "Access token",
                    "Access secret")
#selection:
1
```

```
#load useful package
library(dplyr)
library(stringr)
library(rebus)
library(tidyr)
library(ggplot2)
library(ggmap)
library(mapproj)
library(twitteR)
```

3. getting and cleaning data:

3.1. First, I would like to try to pull 10000 sample from the twitter randomly and see how much data can be used. The day I can go back as early as possible is 9/5, so start from 9/5, I will get 5 days data and change it into data frame. The code used list below, using 9-4 and 9-29 as an example

```
#data at 9-4
sample_9_4<-searchTwitter('#HurricaneHarvey until:2017-9-5', n=10000)
HH9_4<-twListToDF(sample_9_4)
#data at 9-29
HH9_29<-searchTwitter('a until:2017-9-30 ',n=10000)
HH9_29<-twListToDF(HH9_29)
```

3.2. Delete lines without location information in the original dataset and create new dataset called cHHs.(9-4as an example)

```
#select data that have location information

for(i in c(4:8,29)){
 cHHs<-get(paste("HH9",i,sep="_"))
 cHHs$location[cHHs$location==""]<-NA
 cHHs$location<-as.factor(cHHs$location)
 cHHs<-filter(cHHs,!is.na(location))
 write.csv(cHHs,paste("c9-",i,".csv",sep=""))
 assign(paste("cHH9",i,sep="_"),cHHs)

}
```

3.3 In order to continue analysis, I need first clean the format of the data, seperate the location information by city and state, showing all the charachers in upper form and cleaning the spaces. After that, I seperate the dataset into two part, the one with useful state information and the one without useful state information.

```
#seperate the location information into state and city,export those data with state information
for(i in c(4:8,29)){
  temp<-get(paste("cHH9",i,sep="_"))
  temp<-separate(temp,location,c("city","state"),sep=",")
  temp$state<-toupper(temp$state)
  temp$city<-toupper(temp$city)
  temp$state<-as.factor(str_replace_all(temp$state,pattern=" ",replacement=""))
  temp$city<-as.factor(str_replace_all(temp$city,pattern=" ",replacement=""))
  assign(paste("cHH9",i,sep="_"),temp)
}
```

3.4 I calculated the time each city shows in TX and LA, both in the data that containing a state information and without state information. Add state information to the final dataframe

```
for(i in c(4:8,29)){

  temp<-get(paste("cHH9",i,sep="_"))
  y_state<-filter(temp,state!="USA"
                  &!is.na(state))
  y_state_1<-filter(y_state,state=="TX"|state=="TEXAS")
  y_state_2<-filter(y_state,state=="LA"|state=="LOUISIANA")
  y_state_3<-filter(y_state,state=="MA"|state=="MASSACHUSETTS")
  y_state_4<-filter(y_state,state=="NV"|state=="NEVADA")

  for(a in 1:4){
    assign(paste("city_c",a,sep="_"),
           summary(get(paste("y_state",a,sep="_"))$city,
                   maxsum = 1000))
    assign(paste("name_city",a,sep="_"),
           names(get(paste("city_c",a,sep="_"))))
    assign(paste("city_ystate",a,sep="_"),
           data.frame(get(paste("city_c",a,sep="_"))))
    temp2<-get(paste("city_ystate",a,sep="_"))
    temp2$city<-as.factor(get(paste("name_city",a,sep="_")))

    #add colnames
    colnames(temp2)<-c("number","city")
    assign(paste("city_ystate",a,sep="_"),
           temp2)
  }
  city_ystate_1$state<-as.factor("TX")
  city_ystate_2$state<-as.factor("LA")
  city_ystate_3$state<-as.factor("MA")
  city_ystate_4$state<-as.factor("NV")
  city_ystate<-rbind(city_ystate_1,city_ystate_2,city_ystate_3,city_ystate_4)

#clear nonsense data$rename
  city_ystate<-filter(city_ystate,number>0)
#calculate frequency
  city_ystate$freq<-city_ystate[ ,"number"]/sum(city_ystate[ ,"number"])
#add date value
  city_ystate$date<-paste("9",i,sep="-")
#reorder
  city_ystate<-city_ystate[order(-city_ystate$number), ]
#save the data
  write.csv(city_ystate,paste("citys_9_",i,".csv",sep=""))

}
```

4. Graphing: The graphing process containing several different steps. 4.1 Basic row data exploration. Graphing the first 20 cities that have the most frequency of showing for different dates using ggplot (geom_bar) and for the reference dates, too. Also, the multiplot function from the internet helped to merge all the graph together.

```
#read in the dataset
for(i in c(4:8,29)){

  temp<-read.csv(paste("citys_9_",i,".csv",sep=""))
```

```
  temp<-temp[ ,-1]
  assign(paste("citys_9",i,sep="_"),temp)

}
a<-0
for (i in c(4:8,29)){
  a<-a+1
  city_ystate<-get(paste("citys_9",i,sep="_"))
  city_ystate$city<-factor(city_ystate$city, levels=unique(city_ystate$city))
  assign(paste("p",a,sep=""),ggplot(head(city_ystate,20),aes(x=city,y=freq,fill=state))+
         geom_bar(stat = "identity")+
         labs(title=paste("Sep/",i,"th",sep=""))+
         coord_flip())
}
```

In order to show all the graphs in a big graph, I searched for multigraph function from the internet

```
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
```

```
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

PLOT THE GRAPH:

```
multiplot(p1, p2, p3, p4,p5,p6, cols=2)
```

4.2 Exclude other confounders using the data 9/29 as reference. In order to exclude other confoundings that may have effect on twitter number, I extract the twitter date at Sep 29th. Since September 29th is the date almost a month away from the hit of hurricane Harvey, we can suppose that the twitter number in that date can represent usual twitter number. I get the frequency of each city that shows up in the twitter for the top 40 cities each day from Sep 4th to Sep 8th and Sep 29th. And then, using the frequency in the hurricane days to subtract the frequency of none hurricane days and get the absolute differences. The reason that I didn't use relative risk as indicator is that some cities are in hurricane hit dates are not present in non-hurricane hit dates. These cities have the frequency of 0 and we cannot use them as dominator. After we get the absolute difference, that is the one indicator I would use for the following step, including showing them on the map. The code I use is listed below:

```
for (i in c(4:8,29)){

  head_city<-head(get(paste("citys_9",i,sep="_")),40)
  head_city$freq<-head_city[ ,"number"]/sum(head_city[ ,"number"])
  assign(paste("head",i,sep=""),head_city)
}
for (i in 4:8){

  head<-get(paste("head",i,sep=""))
  head_re<-merge(x=head,y=head29,by="city",all.x=TRUE)
  head_cl<-head_re %>%
    select(city,freq.x,freq.y,state.x) %>%
    arrange(desc(freq.x))
  head_cl[is.na(head_cl)]<-0
  head_cl<-head_cl %>%
    mutate(freq_ad=freq.x-freq.y) %>%
    arrange(desc(freq_ad))
  write.csv(head_cl,paste("adcity_9_",i,".csv",sep=""))
}
```

4.3 using ggmap package and ggplot to graph the area on the map as to visualize the data.

4.3.1 first, use the ggmap get the geo information from the internet, and save the data into a new dataset

```
#big one
library(ggmap)
library(mapproj)

#read in the dataset:
for (i in 4:8){
  assign(paste("adcity_9_",i,sep=""),read.csv(paste("adcity_9_",i,".csv",sep=""))[,-1])
}
```

```
#fit geo information:
for (i in 4:8){
  location<-get(paste("adcity_9",4,sep="_"))

  location$longi<-NA
  location$latti<-NA

  for(a in 1:nrow(location)){
    location[a, 6:7] <- geocode(as.character(location[a,"city"])) %>% as.numeric
  }

  #for those not fitted in the first loop
  for(b in 1:nrow(location)){
    if(is.na(location[b, 7])){
      location[b, 6:7] <- geocode(as.character(location[i,"city"])) %>% as.numeric
    }
    else{}
  }
  location<-filter(location,!is.na(longi))
  location<-filter(location,longi<=-60&longi>=-140&latti>=20&latti<=50)
  assign(paste("adcity_9",i,sep="_"),location)
  write.csv(location,paste("loc_9_",i,".csv",sep=""))
}
```

4.3.2 Using the ggmap package, showing the result. The code I used is as below

```
#read in the datasete
  for(i in 4:8) {

    loc<-read.csv(paste("loc_9_",i,".csv",sep=""))
    loc<-loc[ ,-1]
    assign(paste("loc_9",i,sep="_"),loc)
  }
statemap<-get_map(location='Texas',zoom=6,maptype="toner")
statemap<-ggmap(statemap)

b<-0
for (i in 4:8){
  b<-b+1
  location<-get(paste("loc_9",i,sep="_"))
  texas<-filter(location,state=="TX"|state=="LA")
  assign(paste("gb",b,sep=""),
         statemap +  geom_point(aes(x=longi, y = latti), data = texas,
                                alpha=0.4, col="yellow",size = texas$freq_ad*0.1)+
                     geom_point(aes(x = longi, y = latti), data = texas,
                                alpha = 0.6, col = "red", size = 0.5) +
                     theme(axis.title.x = element_text(size = 5),
                           axis.text.x=element_text(size = 5),
                           axis.title.y = element_text(size = 5),
                           axis.text.y=element_text(size = 5),
                           text = element_text(size = 10))+
                     labs(title=paste("Sep/",i,"th",sep=""))+
                     scale_size_continuous(range = range(texas$number)))
```

```
}
multiplot(gb1, gb2, gb3, gb4,gb5, cols=2)
```

5. Statically comparing and assumption checking

5.1 comparing the frequency of Maryland cities showed up and TX/LA cities showed up. Test the hypothesis of whether most hit area got the most tweets number.

5.2 Comparing the frequency of hurricane dates and non-hurricane dates. Both 5.1 and 5.2 are done by the adjusting. If after adjusting, there are more TX/LA cities shows up than before adjusting, it would support the assumption

5.3 Using the number difference between reference date and hurricane date as the new number to identify the most several hurt areas.

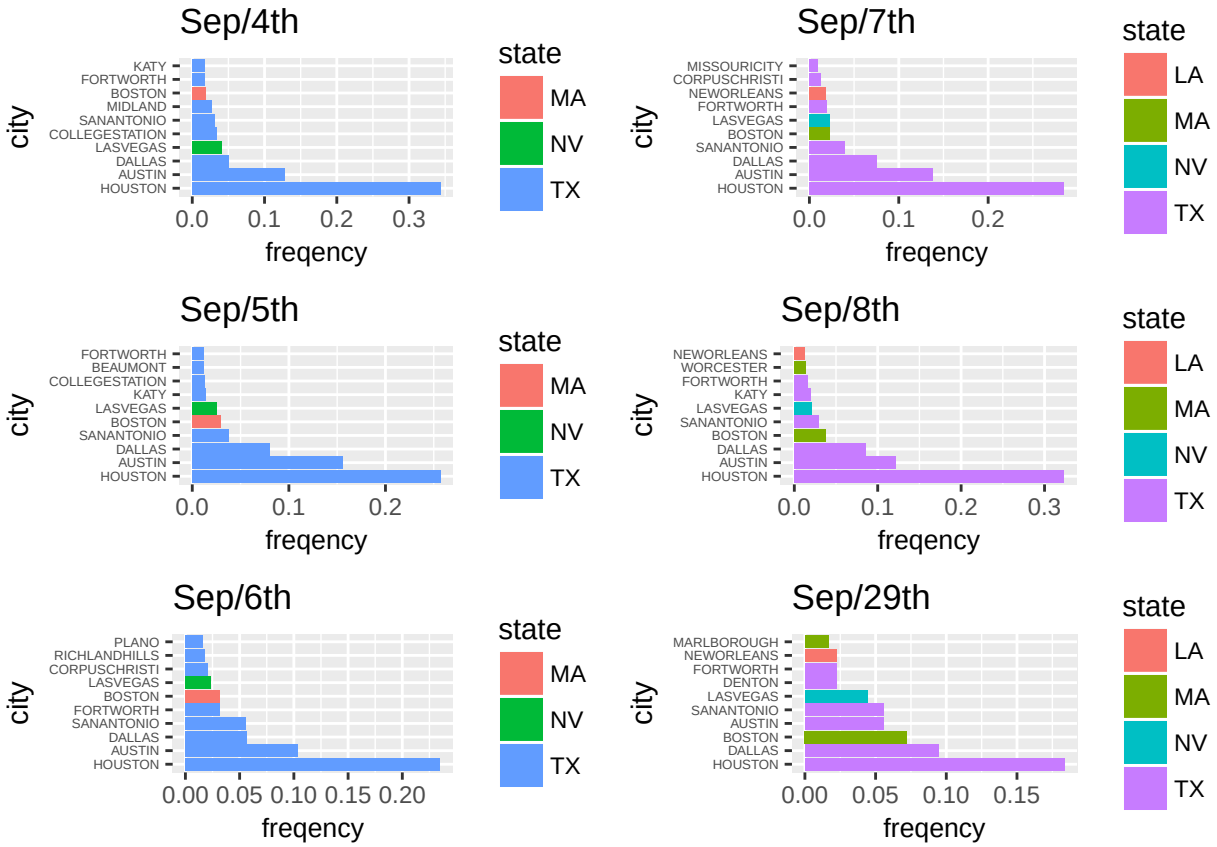
result:

1. Raw data:

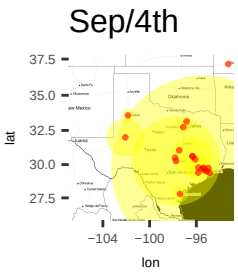
After the step before, I got the row data for 5 days listed in the table below.

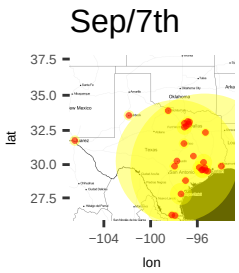
```
##   dates observation_num variable_num
## 1   9-4            3256           20
## 2   9-5            7521           20
## 3   9-6            7359           20
## 4   9-7            7503           20
## 5   9-8            7411           20
## 6  9-29            6798           20
```

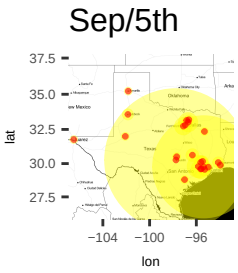
The Bar plot before adjusting among different cities is as below:

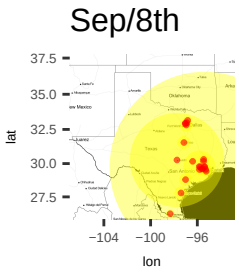


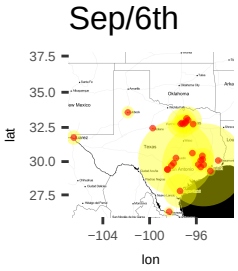
2. Basic map after adjusting comparing different dates:





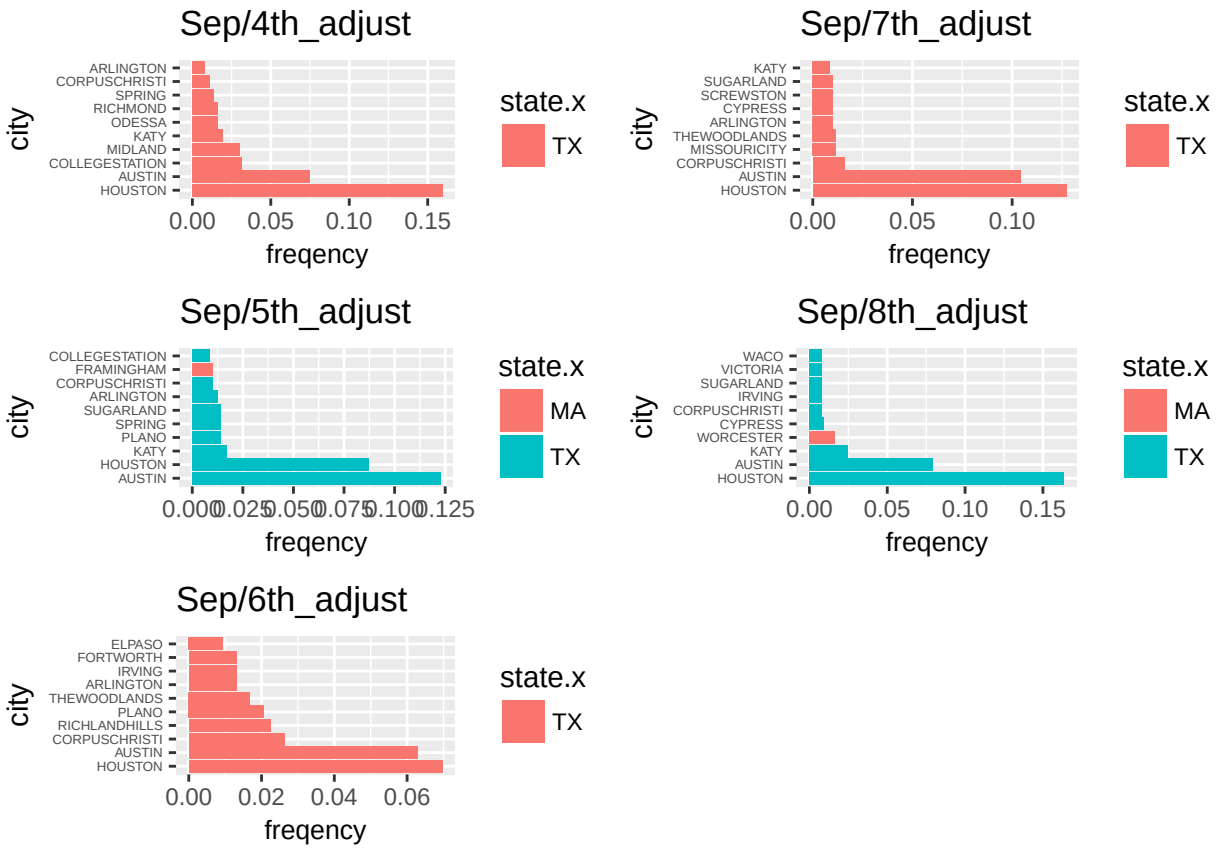






3. Assumption checking:

3.1 I am going to use bar plot to show each date, the top 10 cities before adjusting and after adjusting to test the hypothesis 1 and 2.The following plot is the result after adjusting



From the bar plot, we could see clear difference between adjustment and non-adjustment data. Taking September 4th as an example, before adjustment, there are 3 cities in non-hurricane area showing up in the first 20 cities, but after adjustment, there is no city. It shows that, there is a difference between the tweet pattern before and after the hurricane hurt.

3.2 Identify the most serious hurt area: From the result below, the city that have the most tweets after adjustment is Huston, Austin, other cities changed a lot base on different dates.

```
##       9-4                 9-5                 9-6                 9-7
##  [1,] "HOUSTON"           "AUSTIN"            "HOUSTON"           "HOUSTON"
##  [2,] "AUSTIN"            "HOUSTON"           "AUSTIN"            "AUSTIN"
##  [3,] "COLLEGESTATION"    "KATY"              "CORPUSCHRISTI"     "CORPUSCHRISTI"
##  [4,] "MIDLAND"           "PLANO"             "RICHLANDHILLS"     "MISSOURICITY"
##  [5,] "KATY"              "SPRING"            "PLANO"             "THEWOODLANDS"
##  [6,] "ODESSA"            "SUGARLAND"         "THEWOODLANDS"      "ARLINGTON"
##  [7,] "RICHMOND"          "ARLINGTON"         "ARLINGTON"         "CYPRESS"
##  [8,] "SPRING"            "CORPUSCHRISTI"     "IRVING"            "SCREWSTON"
##  [9,] "CORPUSCHRISTI"     "FRAMINGHAM"        "FORTWORTH"         "SUGARLAND"
## [10,] "ARLINGTON"         "COLLEGESTATION"    "ELPASO"            "KATY"
## [11,] "CYPRESS"           "PEARLAND"          "TOMBALL"           "RICHMOND"
## [12,] "LUBBOCK"           "TOMBALL"           "CYPRESS"           "CAMBRIDGE"
## [13,] "QUINCY"            "BEAUMONT"          "FARMERSBRANCH"     "ELPASO"
## [14,] "BAYCITY"           "LUBBOCK"           "KATY"              "FRISCO"
## [15,] "BAYTOWN"           "THEWOODLANDS"      "LUBBOCK"           "IRVING"
## [16,] "CAMBRIDGE"         "IRVING"            "SANMARCOS"         "LUBBOCK"
## [17,] "ELGIN"             "MCKINNEY"          "COLLEGESTATION"    "PEARLAND"
## [18,] "FRISCO"            "SCREWSTON"         "FRISCO"            "SANMARCOS"
## [19,] "GALVESTONCOUNTY"   "AMARILLO"          "GALVESTON"         "WACO"
```

```
## [20,] "NAVASOTA"        "AMHERST"        "GEORGETOWN"     "CLUTCHCITY"
##       9-8
##  [1,] "HOUSTON"
##  [2,] "AUSTIN"
##  [3,] "KATY"
##  [4,] "WORCESTER"
##  [5,] "CYPRESS"
##  [6,] "CORPUSCHRISTI"
##  [7,] "IRVING"
##  [8,] "SUGARLAND"
##  [9,] "VICTORIA"
## [10,] "WACO"
## [11,] "ARLINGTON"
## [12,] "PEARLAND"
## [13,] "ALVIN"
## [14,] "CONROE"
## [15,] "DUXBURY"
## [16,] "HOUSTONAREA"
## [17,] "RICHMOND"
## [18,] "SPRING"
## [19,] "THEWOODLANDS"
## [20,] "CARROLLTON"
```

Discussion.

In this project, I predict the most hurt area by counting the number of tweets sending from each city. From the result, Huston and Austin get the most frequency of showing up and this is the same as what I found from the actual damage information from the internet. However, there are few limitations about this analysis. As the result, although that the results are most similar to the actual serious hurt area, there is an exception. Dallas, which is also serious hurt, not shown itself in the map. In order to make sure why this happen, I check again the original dataset before the adjustment, and there it is, always in the best 10 of the city list. So, what happened is, it shows up in high frequency on date 9/29, too. And after the subtraction, it disappear. This might happen by on the specific day of 9/29, people in Dallas accidently tweet a lot by some reason. However, by using the twitter, it can give a basic map of what is happening across the country, but not that precisely.

Data source:

- twitter
- twitterR package
- multigraph