Twitter Analysis Using R 2.0

Darren Upton

Comparing Sentiments Between #LGBT and #MAGA

LGBT Twitter and MAGA Twitter have countless differences that can be explored using packages to access Twitter's API to download dataframes of tweets. With these, the tweets can be processed using sentiment analysis to determine which group is the most "positive" and "negative" as well as other characteristics.

Get DF Of Useful Variables

The data frame downloaded from Twitter has many unnecessary columns (at least for this analysis), thus shrinking the size of the data frame in use makes the analysis faster and less confusing.

```
#Load Data Frames
load("#LGBT.RData")
load("#MAGA.RData")
load("#LGBT2.RData")
load("#MAGA2.RData")

LGBT<-(rbind(LGBT,LGBT2))
MAGA<-(rbind(MAGA,MAGA2))

#Only one Tweet per account every 15 minutes
sub.LGBT<-LGBT[!duplicated(cbind(LGBT$user_id,date(LGBT$created_at),
                                 round(minute(LGBT$created_at)/15)*15,hour(LGBT$created_at))),]
sub.MAGA<-MAGA[!duplicated(cbind(MAGA$user_id,date(MAGA$created_at),
                                 round(minute(MAGA$created_at)/15)*15,hour(MAGA$created_at))),]
#Combine into Single Data Frame
LGBT.Text<-data.frame(account=sub.LGBT$screen_name ,text=sub.LGBT$text, time=sub.LGBT$created_at,
                      type="LGBT",id=sub.LGBT$status_id)
MAGA.Text<-data.frame(account=sub.MAGA$screen_name ,text=sub.MAGA$text, time=sub.MAGA$created_at,
                      type="MAGA", id=sub.MAGA$status_id)
Combined<-unique(rbind(LGBT.Text,MAGA.Text))

#Total Number of Tweets
summary(Combined)
```

```
##    account             text               time
##  Length:717188      Length:717188      Min.   :2020-06-10 10:21:33
##  Class :character   Class :character   1st Qu.:2020-07-15 17:14:57
##  Mode  :character   Mode  :character   Median :2020-09-06 13:57:55
##                                        Mean   :2020-12-31 07:55:01
##                                        3rd Qu.:2021-06-11 14:04:58
##                                        Max.   :2022-02-07 21:34:00
##      type               id
##  Length:717188      Length:717188
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
#summary(as.factor(MAGA$screen_name))
#summary(as.factor(LGBT$screen_name))

#format(object.size(sub.LGBT)/8, units = "MB")
rm(LGBT,MAGA,sub.LGBT,sub.MAGA,LGBT.Text,MAGA.Text)

head(Combined)
```

	account <chr>
1	mjwwww_
2	toocool4skool69
3	toocool4skool69
4	toocool4skool69
5	PhilOllenberg
6	Emilyvail

6 rows | 1-2 of 6 columns

Split Character Vectors into Individual Words & Tidy Data Frame of Words

This removes “filler” words from the data frame, so that the sentiment analysis is more fruitful and the word cloud is useful.

```
Tidy.Words<- Combined %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words,by="word") %>%
  filter(is.na(word) != TRUE)
head(Tidy.Words)
```

	account <chr>	time <dttm>	type <chr>	id <chr>	word <chr>
1	mjwwww_	2020-09-27 23:42:41	LGBT	1310364267657527299	it’s
2	mjwwww_	2020-09-27 23:42:41	LGBT	1310364267657527299	24th

	account	time	type	id	word
	<chr>	<dttm>	<chr>	<chr>	<chr>
3	mjwwww_	2020-09-27 23:42:41	LGBT	1310364267657527299	birthday
4	mjwwww_	2020-09-27 23:42:41	LGBT	1310364267657527299	twitter
5	mjwwww_	2020-09-27 23:42:41	LGBT	1310364267657527299	peeps
6	mjwwww_	2020-09-27 23:42:41	LGBT	1310364267657527299	lgbt

6 rows

Word Clouds

This chunk gets counts for the words and places the top 0.1 % in a word cloud based on the type (LGBT or MAGA).

```
#Character vectors that need to be removed from the mix
remove.words<-c("https","t.co",1:10,"2a",2020,2021,"it's","i'm","lgbt",
                "maga","tcot")

#LGBT Counts
LGBT.Count<-Tidy.Words %>%
  filter(type=="LGBT") %>%
  dplyr::count(word, sort = TRUE) %>%
  filter(n > quantile(n, 0.999),
         !word %in% remove.words)

#MAGA Counts
MAGA.Count<-Tidy.Words %>%
  filter(type=="MAGA") %>%
  dplyr::count(word, sort = TRUE) %>%
  filter(n > quantile(n, 0.999),
         !word %in% remove.words)

#Word Clouds
wordcloud2(LGBT.Count[1:75,])
```


```
wordcloud2(MAGA.Count[1:75,])
```

```
#Save Word Clouds
#saveWidget(LGBT.cloud,"lgbt.html",selfcontained=F)
#webshot("lgbt.html","LGBT.cloud.png",delay=5,vwidth=480,vheight=480)
#saveWidget(MAGA.cloud,"maga.html",selfcontained=F)
#webshot("maga.html","MAGA.cloud.png",delay=5,vwidth=480,vheight=480)

#Base Word Cloud
#wordcloud(words = LGBT.Count$word, freq = LGBT.Count$n,
#min.freq = 1,scale=c(4.5,1),max.words=200, random.order=FALSE, rot.per=0.15,
#colors=brewer.pal(8, "Dark2"))
```

Get Sentiments

This chunk gets the sentiment for each word using four different methods that will be compared in later chunks.

```
#First Sentiment Method (Values between -3 and 3)
Sent1<-Tidy.Words %>%
  inner_join(get_sentiments("afinn"),by="word") %>%
  ddply(c(.(id),.(type),.(time)),summarize,Sentiment=sum(value)) %>%
  filter(Sentiment < quantile(Sentiment, 0.999,na.rm=T),
         Sentiment > quantile(Sentiment, 0.001,na.rm=T))

#Second Sentiment Method (Values between -1 and 1)
Sent2 <- Tidy.Words %>%
  inner_join(get_sentiments("bing"),by="word") %>%
  #Count has issues with all of the other packages
  dplyr::count(id, type, time, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative) %>%
  filter(sentiment < quantile(sentiment, 0.9995,na.rm=T),
         sentiment > quantile(sentiment, 0.0005,na.rm=T))

#Third Sentiment Method (Multiple categories)
Sent3<-Tidy.Words %>%
  inner_join(get_sentiments("loughran"),by="word") %>%
  #Very few "superfluous" words in df
  filter(sentiment!="superfluous")

#Fourth Sentiment Method (Multiple categories)
Sent4<-Tidy.Words %>%
  inner_join(get_sentiments("nrc"),by="word")
```

Statatistical Analysis

Student's two-sample t-tests are performed on the first and second sentiment method to compare the means. The null hypothesis states that there is no difference and the alternative states that the LGBT mean sentiment is greater (more positive) than the MAGA mean sentiment.

```
#Sent1 compare means
t.test(Sentiment~type,data=Sent1,alternative="greater",conf.level=0.99)
```

```
##
##  Welch Two Sample t-test
##
## data:  Sentiment by type
## t = 170.32, df = 287402, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 99 percent confidence interval:
##  2.121901      Inf
## sample estimates:
## mean in group LGBT mean in group MAGA
##           1.003222          -1.148062
```

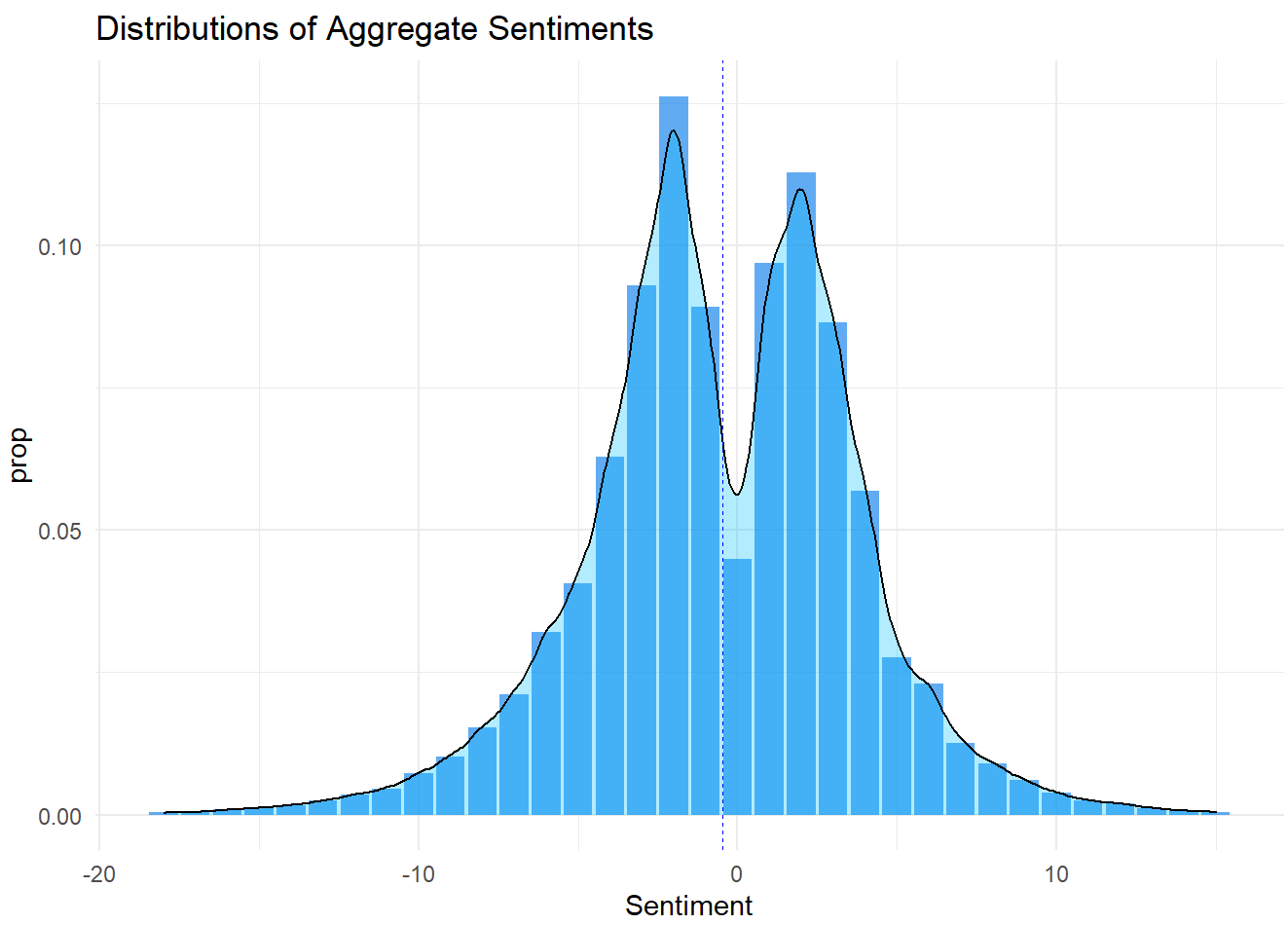
```
#Sent2 compare means
t.test(sentiment~type,data=Sent2,alternative="greater",conf.level=0.99)
```

```
##
##  Welch Two Sample t-test
##
## data:  sentiment by type
## t = 114.1, df = 305076, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 99 percent confidence interval:
##  0.6206478       Inf
## sample estimates:
## mean in group LGBT mean in group MAGA
##          0.2881626         -0.3454030
```

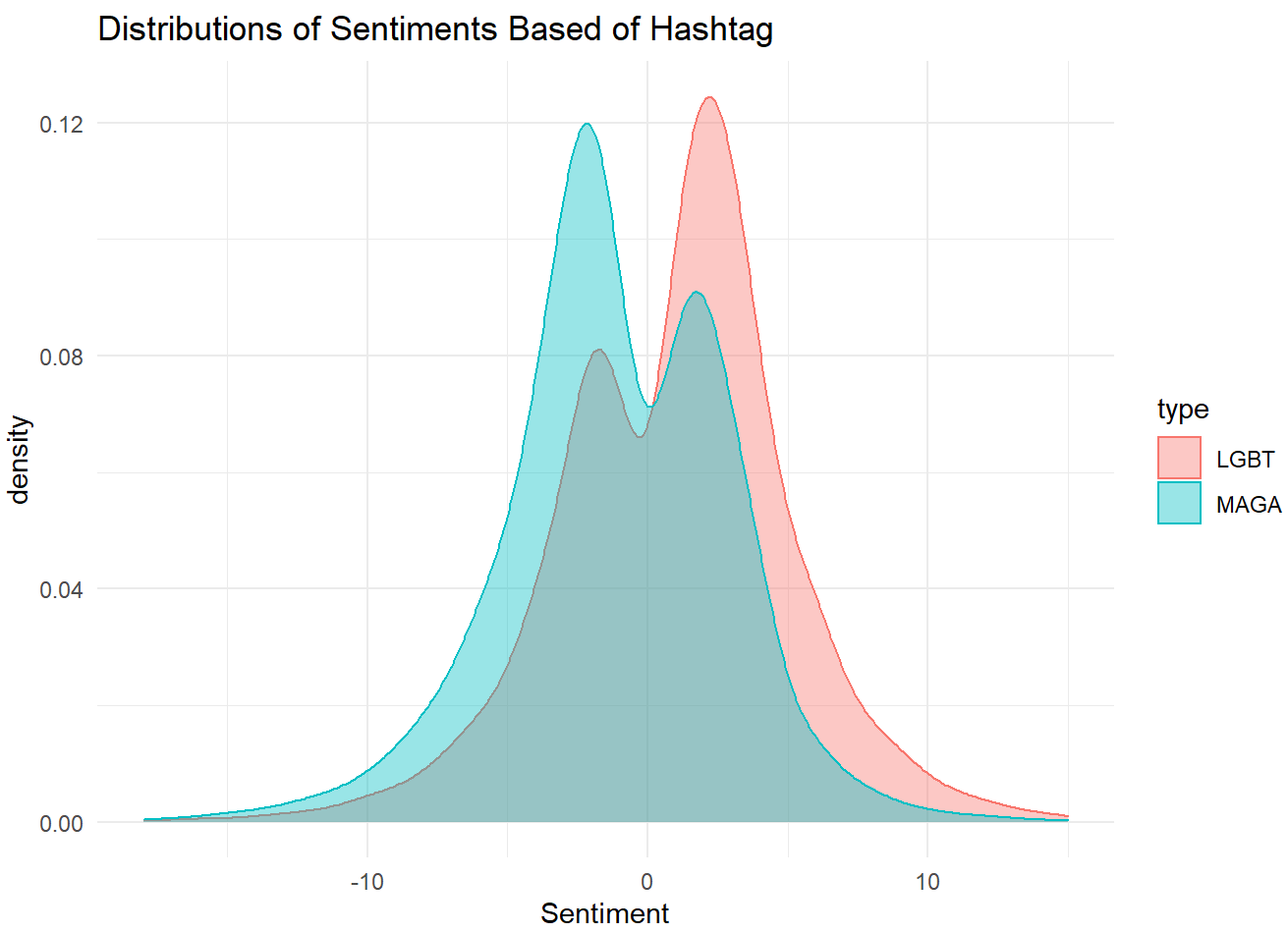
Plots

This chunk plots the distributions of sentiments for the four methods.

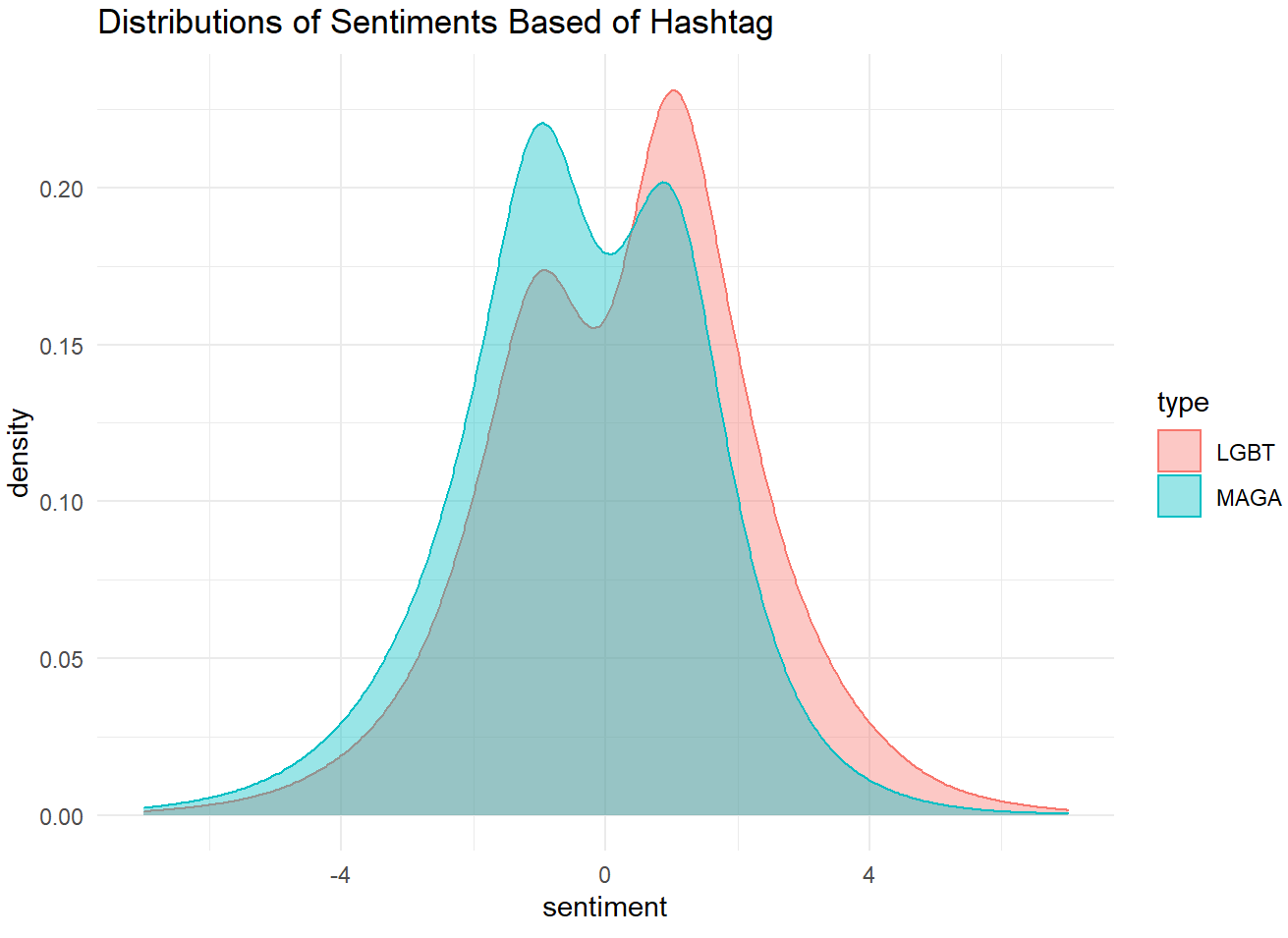
```
#General
ggplot(Sent1,aes(x=Sentiment))+geom_bar(aes(y=..prop..),fill="dodgerblue2",alpha=0.7)+
  geom_density(alpha=0.3,bw=0.5,fill="deepskyblue")+
  geom_vline(aes(xintercept=mean(Sentiment)),color="blue", linetype="dashed", size=0.12)+
  ggtitle("Distributions of Aggregate Sentiments")+theme_minimal()
```



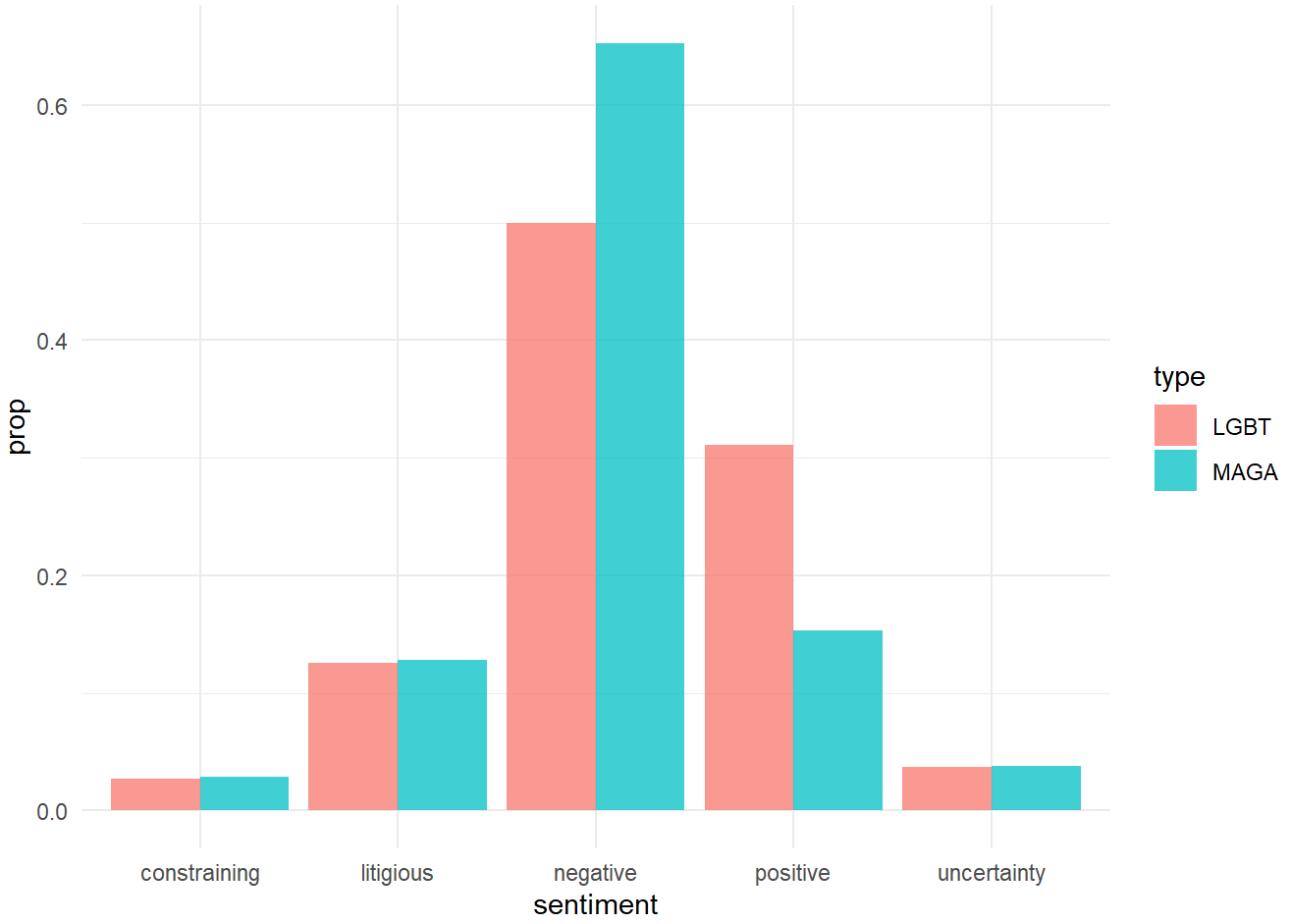

```
#Densities
ggplot(Sent1,aes(x=Sentiment,fill=type,color=type))+geom_density(alpha=0.4,bw=0.8)+
  ggtitle("Distributions of Sentiments Based of Hashtag")+theme_minimal()
```



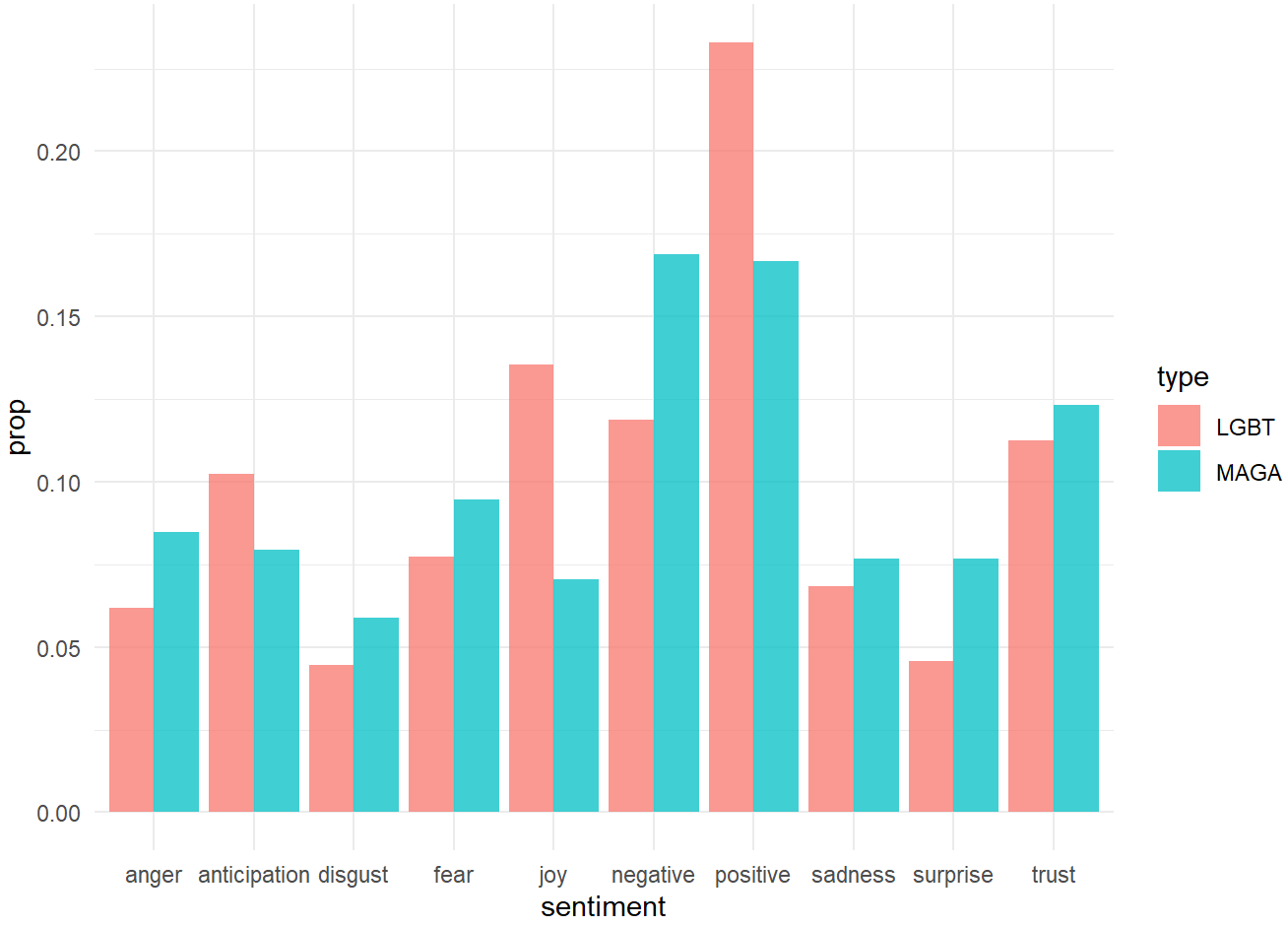

```
ggplot(Sent2,aes(x=sentiment,fill=type,color=type))+geom_density(alpha=0.4,bw=0.6)+
  ggtitle("Distributions of Sentiments Based of Hashtag")+theme_minimal()
```



```
#Other Sentiments
ggplot(Sent3,aes(x=sentiment,group=type,fill=type))+
  geom_bar(aes(y=..prop..,), position=position_dodge(),alpha=0.75)+theme_minimal()
```

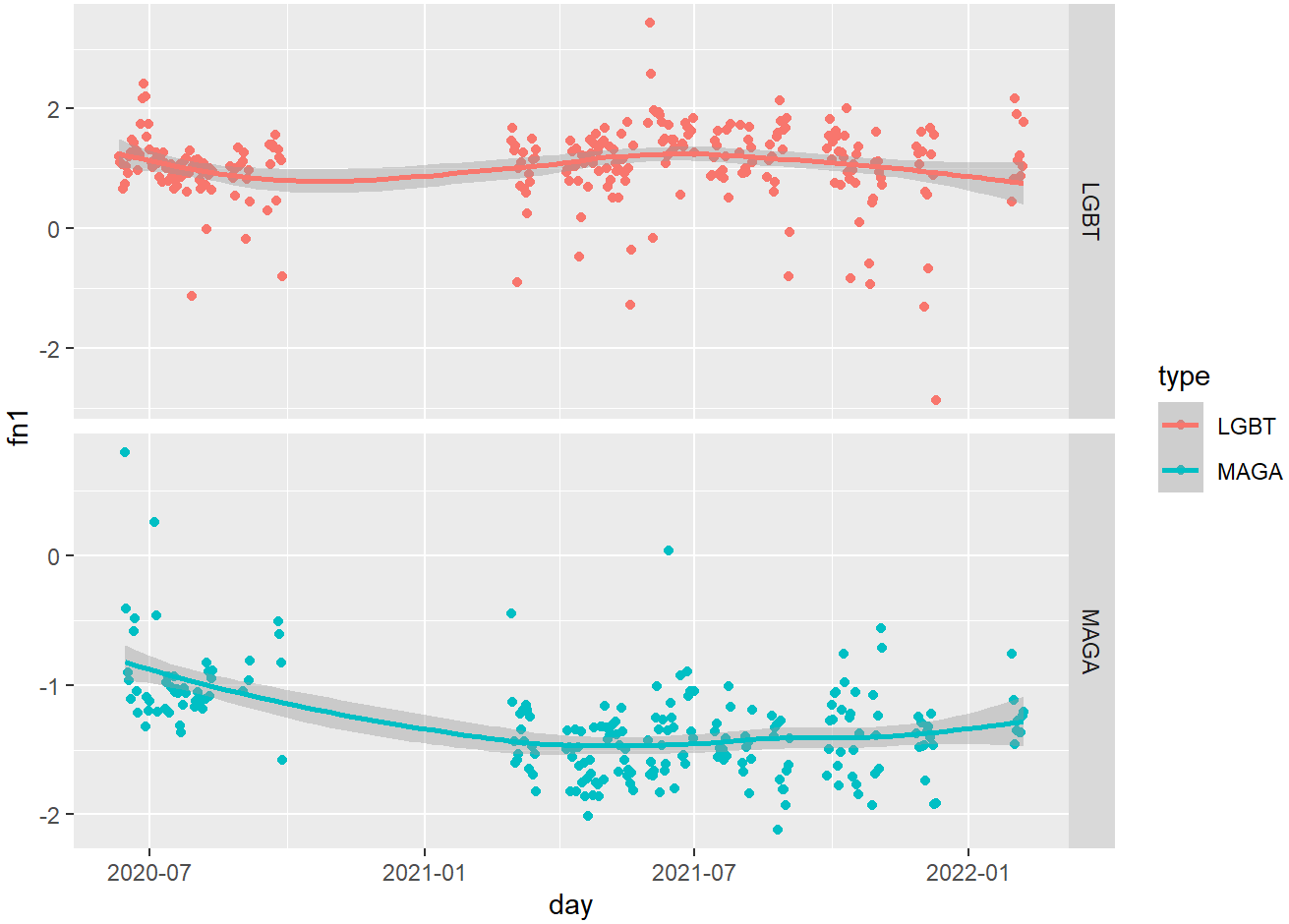



```
ggplot(Sent4,aes(x=sentiment,group=type,fill=type))+
  geom_bar(aes(y=..prop..,), position=position_dodge(),alpha=0.75)+theme_minimal()
```



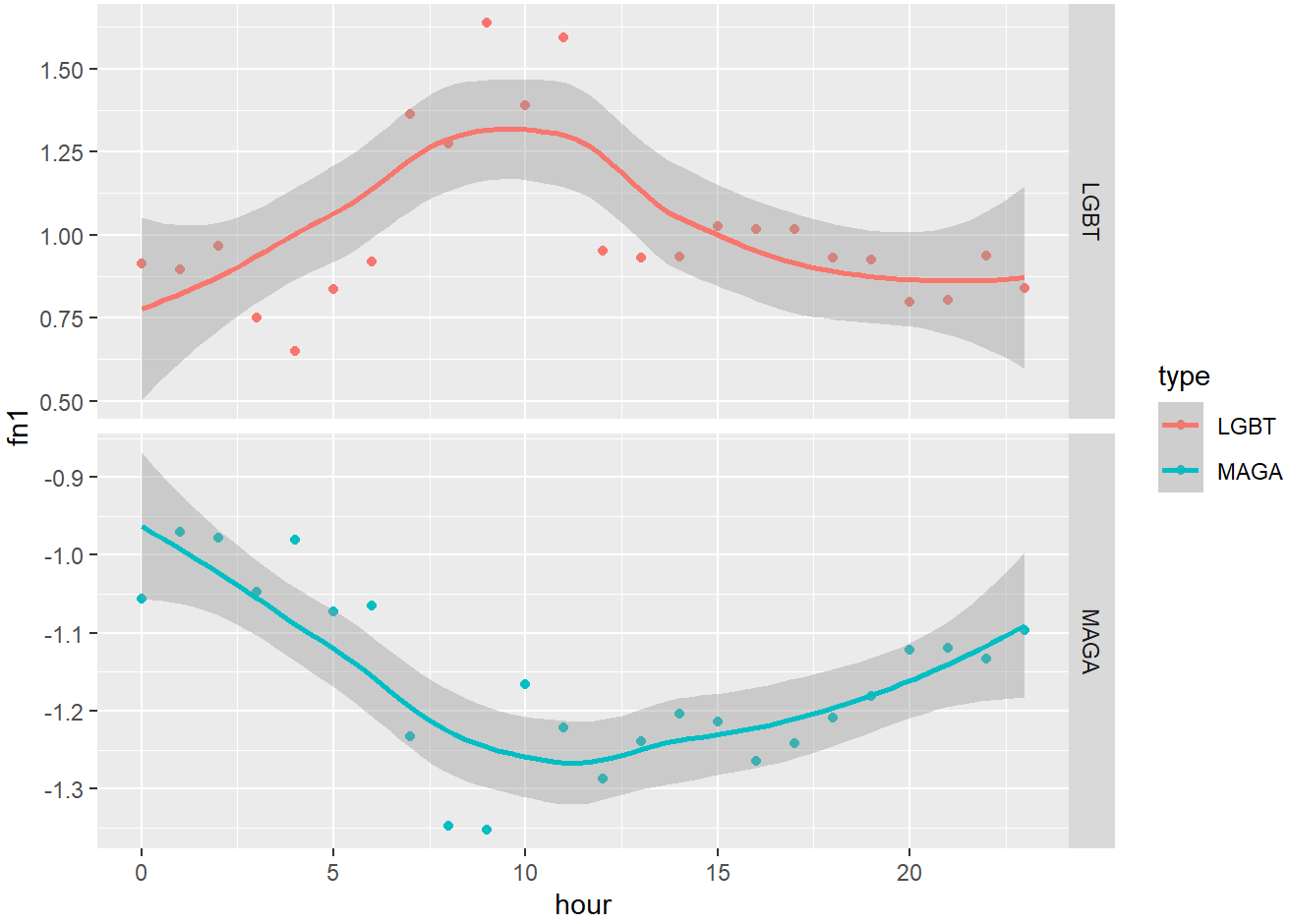

```
Sent1 %>% mutate(day=date(time)) %>% select(c(day,type,Sentiment)) %>% group_by(day,type) %>% summarise_all
(list(mean,sd,length)) %>% ggplot(aes(day,fn1,color=type))+geom_point()+geom_smooth()+
  facet_grid(vars(type),scales = "free")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



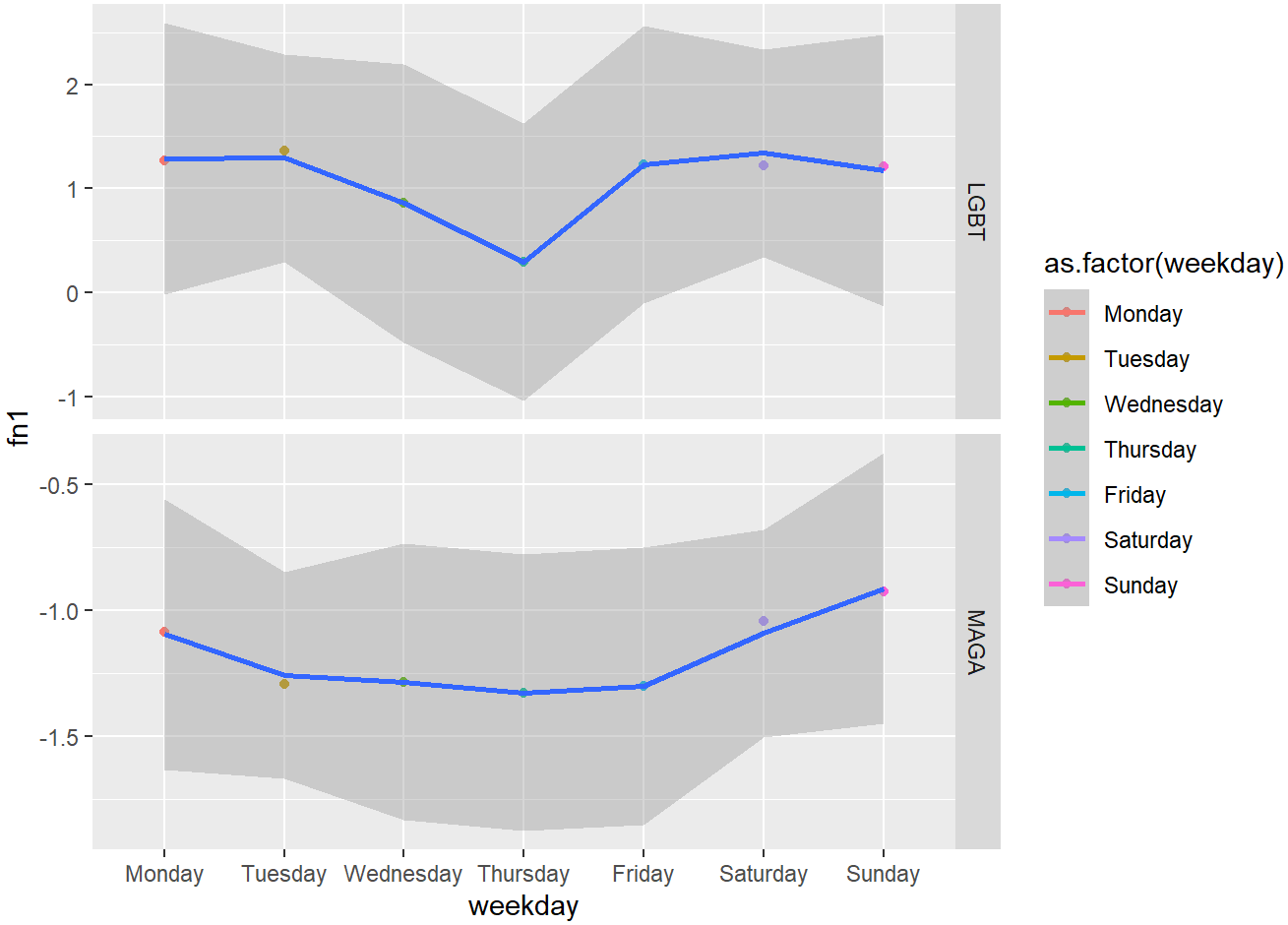

```
Sent1 %>% mutate(hour=hour(time)) %>% select(c(hour,type,Sentiment)) %>% group_by(hour,type) %>% summarise_
all(list(mean,sd,length)) %>% ggplot(aes(hour,fn1,color=type))+geom_point()+geom_smooth()+
  facet_grid(vars(type),scales = "free")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



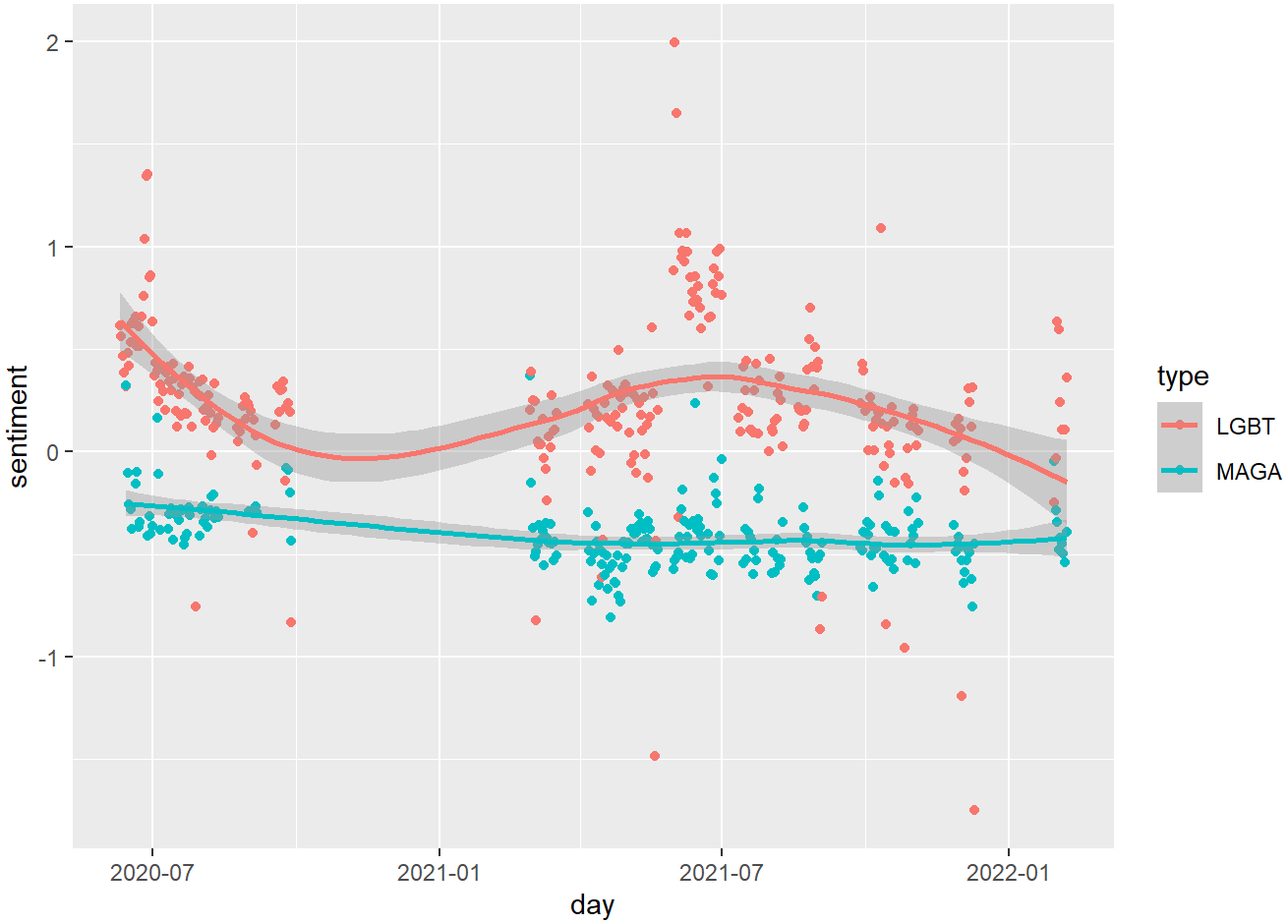

```
Sent1 %>% mutate(weekday=factor(weekdays(time),
             levels=c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"))) %>% select
(c(weekday,type,Sentiment)) %>% group_by(weekday,type) %>% summarise_all(list(mean,sd,length)) %>%  ggplot
(aes(weekday,fn1,color=as.factor(weekday),group=type))+geom_point()+geom_smooth()+
  facet_grid(vars(type),scales = "free")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
Sent2 %>% mutate(day=date(time))  %>% select(c(day,type,sentiment)) %>% group_by(day,type) %>% summarise_al
l(mean) %>% ggplot(aes(day,sentiment,color=type))+geom_point()+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
after.june<- Sent1 %>% filter(time>as.Date("06/30/2020","%m/%d/%y"),type=="LGBT")
june<- Sent1 %>% filter(time<as.Date("07/01/2020","%m/%d/%y"),type=="LGBT")

t.test(june$Sentiment,after.june$Sentiment)
```

```
##
##  Welch Two Sample t-test
##
## data:  june$Sentiment and after.june$Sentiment
## t = 14.143, df = 28308, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3730158 0.4930393
## sample estimates:
## mean of x mean of y
## 1.3820623 0.9490347
```

```
pre.election<- Sent1 %>% filter(time<as.Date("11/01/2020","%m/%d/%y"),type=="MAGA")
after.election<- Sent1 %>% filter(time>as.Date("11/1/2020","%m/%d/%y"),type=="MAGA")

t.test(pre.election$Sentiment,after.election$Sentiment)
```

```
##
##  Welch Two Sample t-test
##
## data:  pre.election$Sentiment and after.election$Sentiment
## t = 34.859, df = 303607, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.456739 0.511159
## sample estimates:
##  mean of x  mean of y
## -0.9508041 -1.4347531
```

```
Sent1 %>% filter(date(time)>as.Date("2/20/21","%m/%d/%y"),
                 date(time)<as.Date("03/15/21","%m/%d/%y"),type=="MAGA") %>% summary
```

```
##       id                type               time
##  Length:15394       Length:15394       Min.   :2021-02-28 16:54:25
##  Class :character   Class :character   1st Qu.:2021-03-03 01:46:21
##  Mode  :character   Mode  :character   Median :2021-03-06 18:04:11
##                                        Mean   :2021-03-07 02:30:47
##                                        3rd Qu.:2021-03-11 00:10:55
##                                        Max.   :2021-03-14 23:56:42
##    Sentiment
##  Min.   :-18.000
##  1st Qu.: -3.000
##  Median : -2.000
##  Mean   : -1.301
##  3rd Qu.:  1.000
##  Max.   : 14.000
```

```
head(date((Sent1$time)))
```

```
## [1] "2020-09-21" "2020-09-26" "2020-09-27" "2020-09-27" "2020-09-25"
## [6] "2020-09-21"
```

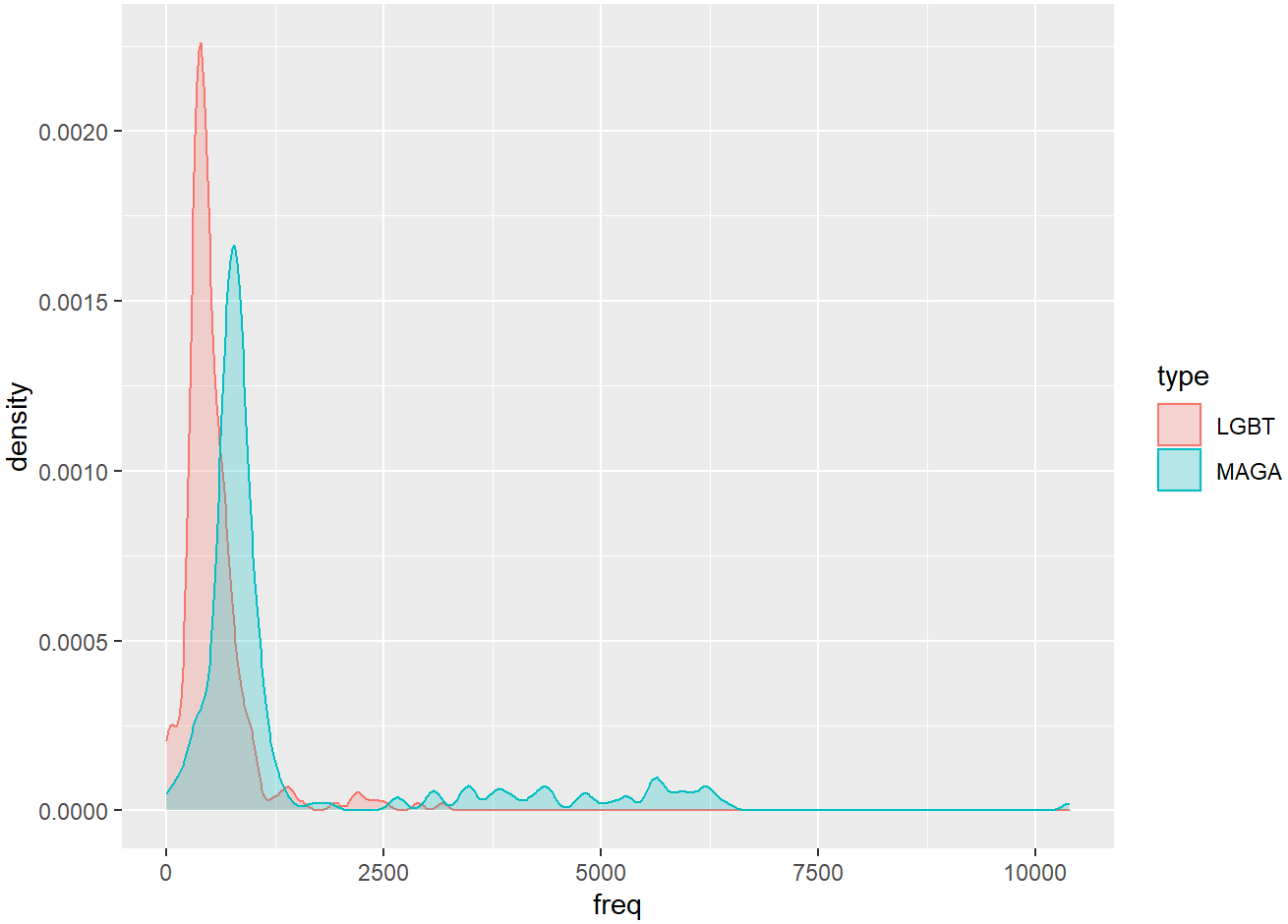
```
Sent2 %>% filter(date(time)==as.Date("2/28/21","%m/%d/%y"),type=="MAGA") %>% select(sentiment) %>% summaris
e_all(mean)
```

sentiment
<dbl>
0.3692771

1 row

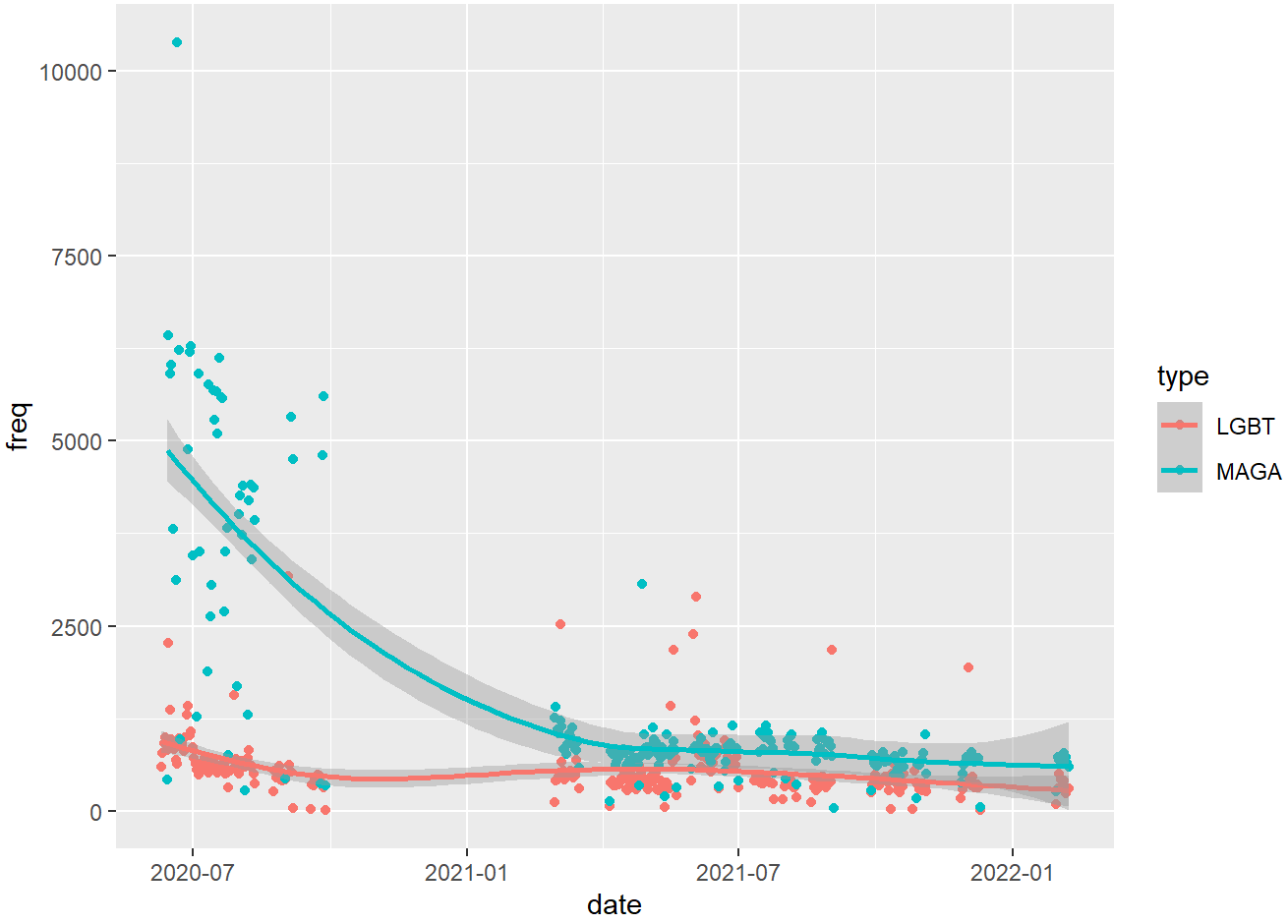
```
date_counts <- Sent1 %>% mutate(date=date(time)) %>% select(type,date) %>%
                         group_by(date,type) %>% count() #%>% filter(freq<2500)

ggplot(date_counts,aes(freq,fill=type,color=type)) + geom_density(alpha=0.25) #+ geom_smooth()
```




```
ggplot(date_counts,aes(date,freq,color=type))+geom_point()+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
date_counts %>% select(type,freq) %>% group_by(type) %>% summarize_all(median)
```

type <chr>	freq <dbl>
LGBT	452.5
MAGA	830.0

2 rows

```
#date_counts %>% filter(date<as.Date("8/1/22","%m/%d/%y"))
date_counts %>% select(type,freq) %>% group_by(type) %>% summarize_all(sum)
```

type <chr>	freq <int>
LGBT	153305
MAGA	336609

2 rows

```
Combined %>% select(type) %>% group_by(type) %>% count
```

type <chr>	freq <int>
LGBT	220925