Twitter zzh-gj-jz association

26 May, 2022

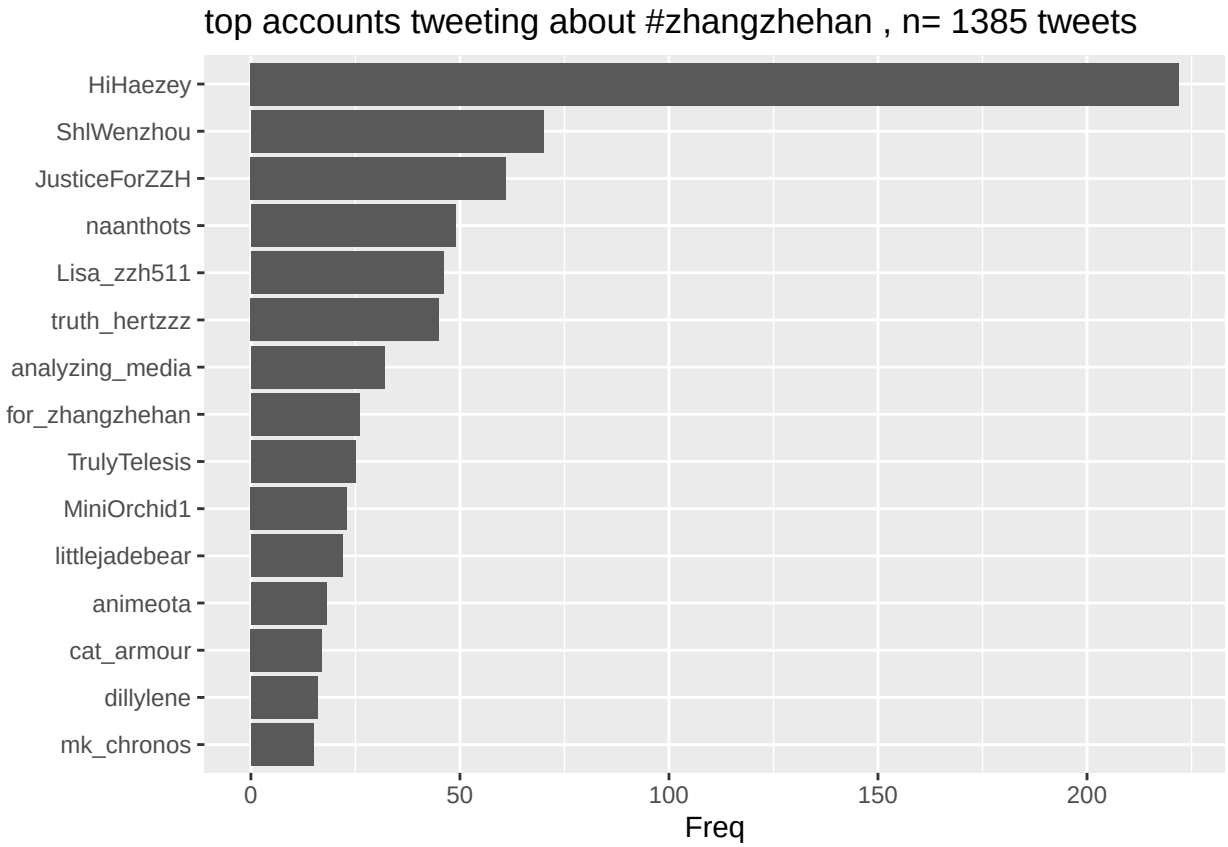
In this analysis I will first extract tweets mentioning "#zhangzhehan", "#gongjun" or "#junzhe" and then check the overlap between accounts tweeting about these topics. After playing around a bit I decided to use only hashtags instead of full names to focus on dedicated tweets instead of casual mentions.

First let's get the tweets via Twitter API. Please note that there is a rate limit of up to 18000 tweets per day. Nonetheless, I never manage to get the maximum of 6000 tweets per query for some reason. Therefore this search is limited to the last couple of days.

I will then plot the accounts with the most tweets about these topics.

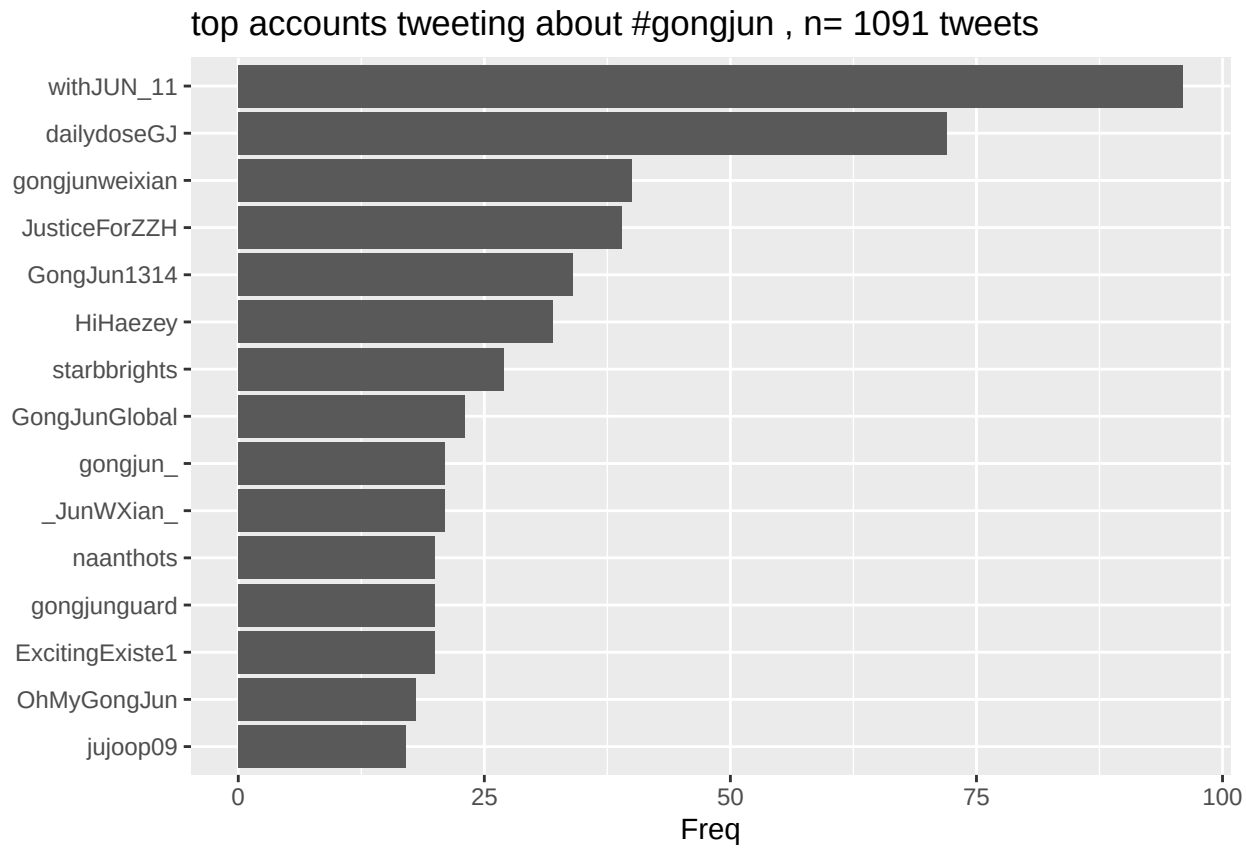
```
###extract tweets

tweets_extract_plot<-function(words,title){
  tweets<-search_tweets2(words,n = 6000,lang = "en",include_rts = FALSE)
  n_tweets<-nrow(tweets)
  stats<-paste(title,collapse=" ")
  stats<-paste(stats,", n=",n_tweets,"tweets")

  plot<-table(tweets$screen_name)%>%
    as.data.frame() %>%
    arrange(desc(Freq)) %>%
    top_n(15) %>%
    mutate(Var1 = reorder(Var1, Freq)) %>%
    ggplot(aes(x = Var1, y = Freq)) +
    geom_col() +
    xlab(NULL) +
    coord_flip() +
    ggtitle(paste("top accounts tweeting about",stats))
  print(plot)
  return(tweets)
}
zzh_tweets<-tweets_extract_plot(c("#zhangzhehan"),"#zhangzhehan")
```

```
## Selecting by Freq
```



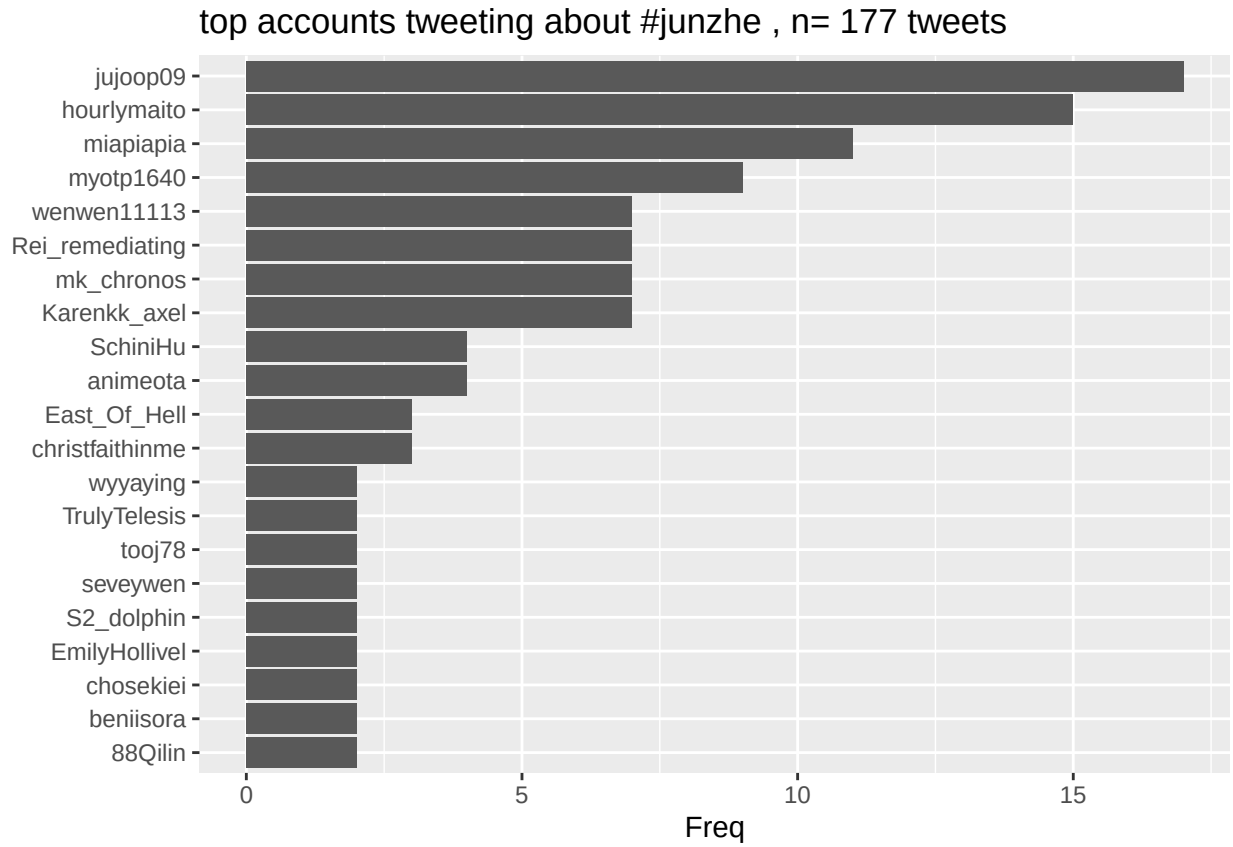

```
gj_tweets<-tweets_extract_plot(c("#gongjun"),"#gongjun")
```

```
## Selecting by Freq
```




```
jz_tweets<-tweets_extract_plot(c("#junzhe"),"#junzhe")
```

```
## Selecting by Freq
```



Quick info on which timespan the datasets cover and how many tweets per day

```
#calculate rate of tweets per hour
tweet_stats<-function(data,title){
  print(paste("dataset:",title))
  print(paste("number of tweets:",nrow(data)))
  start<-data[nrow(data),]$created_at
  end<-data[1,]$created_at
  print(paste("start:",start))
  print(paste("end:",end))
  diff = end - start
  print(diff)
  #tweets per day
  tw_p_day<-round(nrow(data)/as.numeric(diff),2)
  print(paste("tweets per day:",tw_p_day))
  return(tw_p_day)
}

st_zzh<-tweet_stats(zzh_tweets,"#zhangzhehan")
```

```
## [1] "dataset: #zhangzhehan"
## [1] "number of tweets: 1385"
## [1] "start: 2022-05-17 23:46:18"
## [1] "end: 2022-05-26 10:41:55"
## Time difference of 8.455289 days
## [1] "tweets per day: 163.8"
```

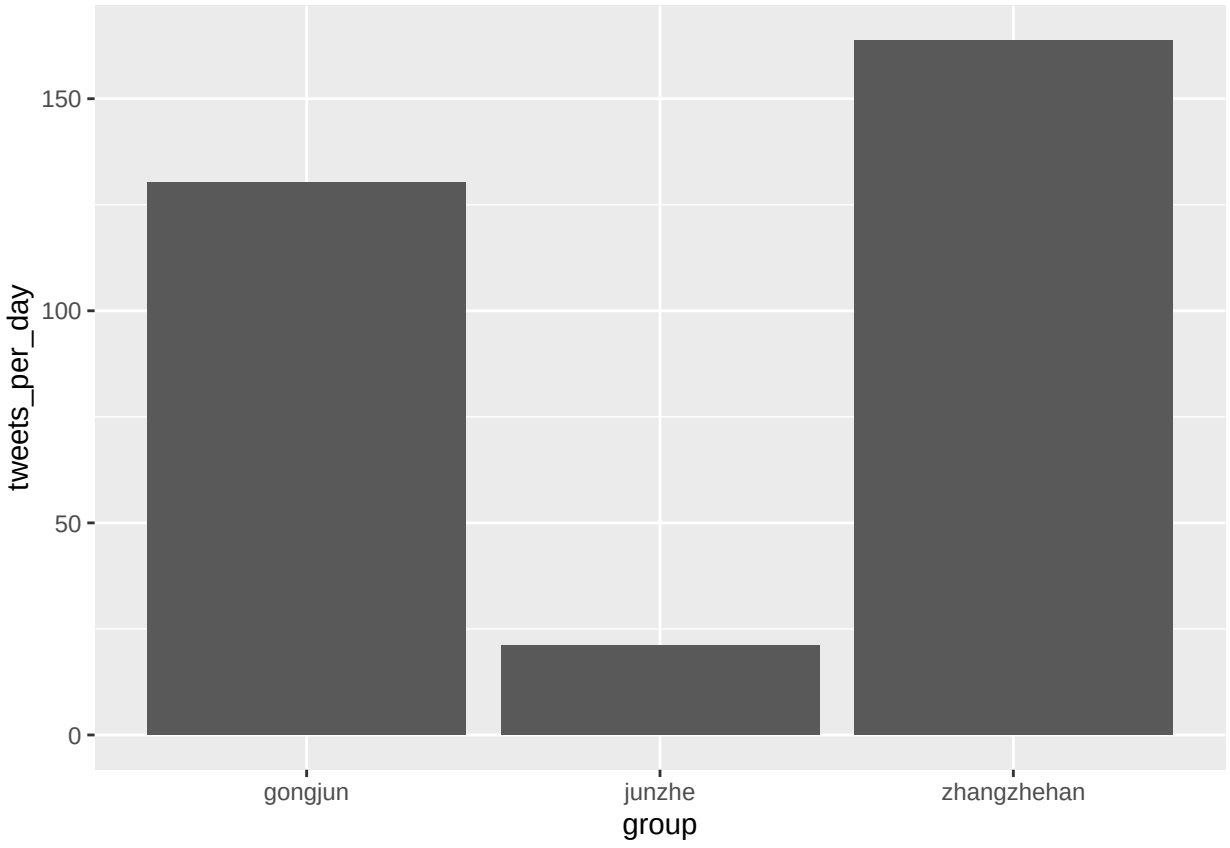
```
st_gj<-tweet_stats(gj_tweets,"#gongjun")
```

```
## [1] "dataset: #gongjun"
## [1] "number of tweets: 1091"
## [1] "start: 2022-05-18 01:39:45"
## [1] "end: 2022-05-26 10:41:55"
## Time difference of 8.376505 days
## [1] "tweets per day: 130.25"
```

```
st_jz<-tweet_stats(jz_tweets,"#junzhe")
```

```
## [1] "dataset: #junzhe"
## [1] "number of tweets: 177"
## [1] "start: 2022-05-18 01:32:49"
## [1] "end: 2022-05-26 10:05:23"
## Time difference of 8.355949 days
## [1] "tweets per day: 21.18"
```

```
bargraph<-data.frame(group=c("zhangzhehan","gongjun","junzhe"),tweets_per_day=c(st_zzh,st_gj,st_jz))

ggplot(bargraph,aes(x=group,y=tweets_per_day))+
  geom_col()
```



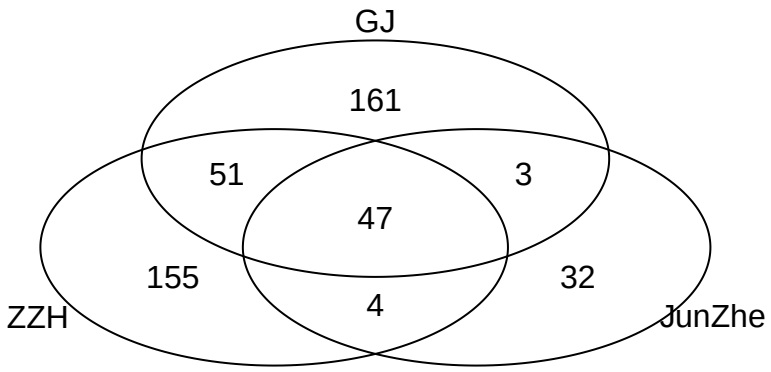
Next let's make a Venn diagram to see the overlap between accounts tweeting about #junzhe, #zhangzhehan or #gongjun. I have to use 2 Venn diagram packages because gplots gives me the numbers+intersections but only venneuler lets me make a beautiful diagram with proportional circle areas.

```
###Venn diagram
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
x <- list("ZZH" = unique(zzh_tweets$screen_name),
          "JunZhe" = unique(jz_tweets$screen_name),
          "GJ" = unique(gj_tweets$screen_name))
v.table <- venn(x)
```




```
length(unique(zzh_tweets$screen_name))
```

```
## [1] 257
```

```
length(unique(jz_tweets$screen_name))
```

```
## [1] 86
```

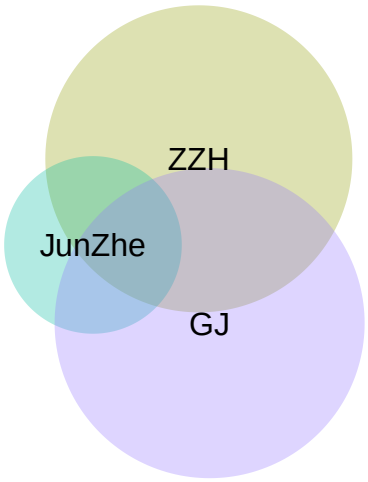
```
length(unique(gj_tweets$screen_name))
```

```
## [1] 262
```

```
x <- attr(v.table,"intersections")
names_venn<-sapply(seq_along(x), function(i) paste(names(x)[[i]]))
numbers_venn<-lapply(seq_along(x), function(i) length(x[[i]]))
names_venn<-str_replace(names_venn, ":", "&")
names_venn<-str_replace(names_venn, ":", "&")
numbers_venn<-as.numeric(numbers_venn)
names(numbers_venn)<-names_venn


library(venneuler)
```

```
## Loading required package: rJava
```

```
v <- venneuler(numbers_venn)
plot(v)
```

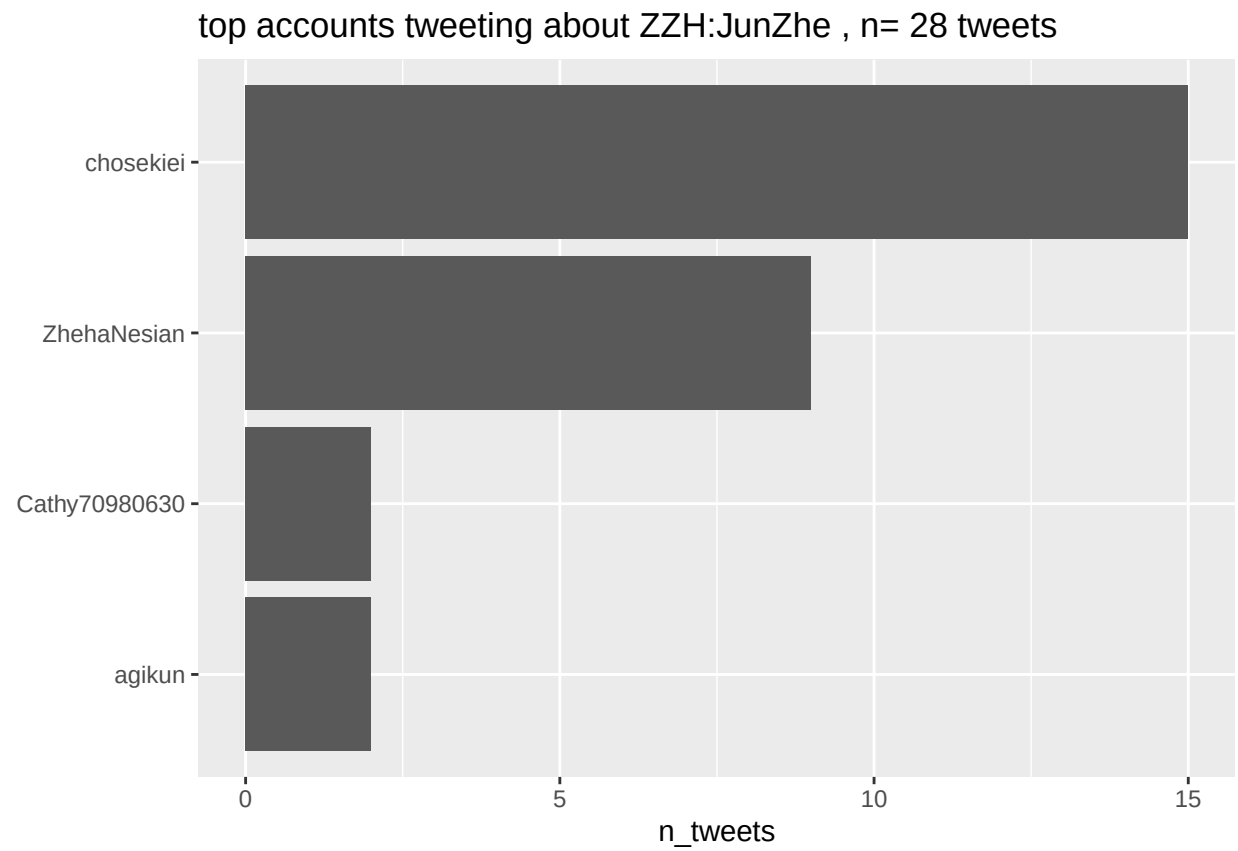


This Venn diagram only shows numbers of accounts tweeting about the three hashtags However I also want to see how many tweets are coming from each section of the Venn diagram.

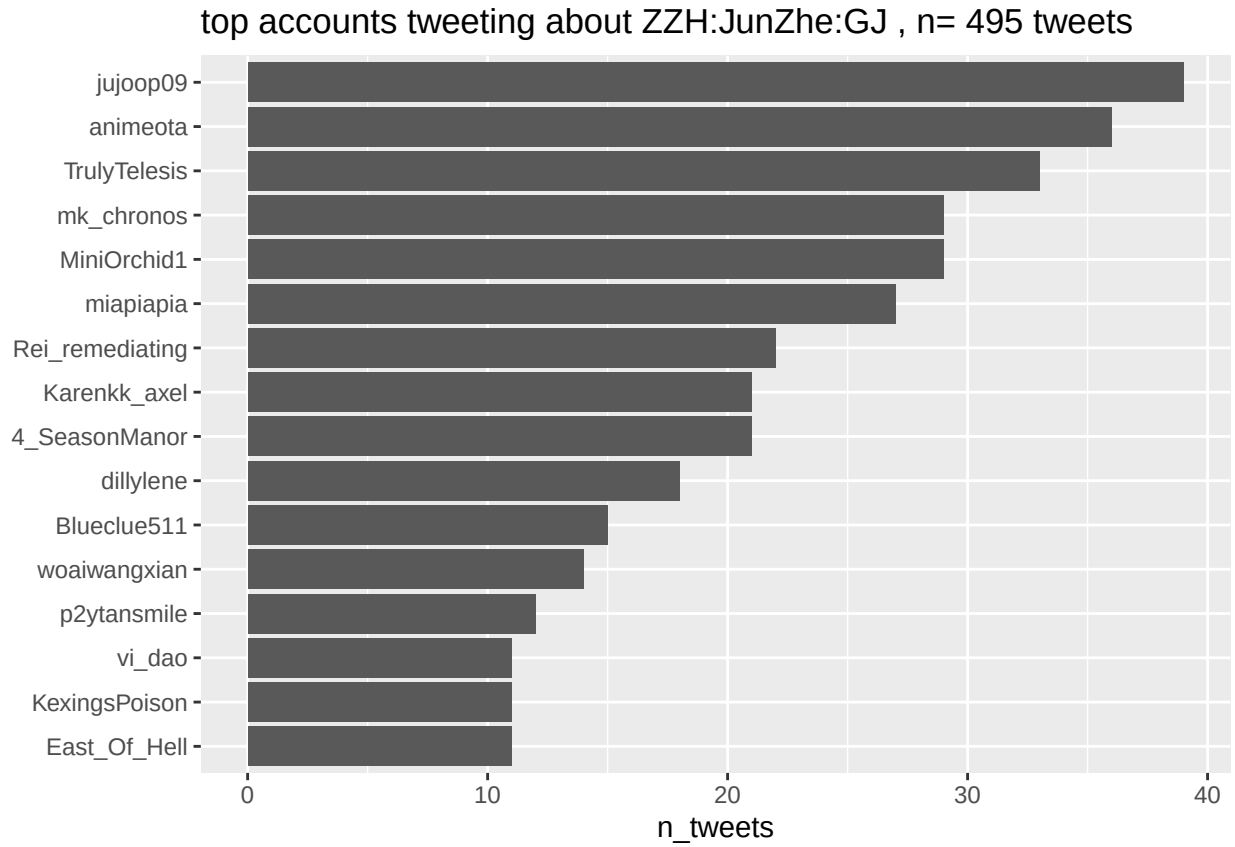
```r
#merge lists of tweets
zzh_tweets$group<-c("#zhangzhehan")
gj_tweets$group<-c("#gongjun")
jz_tweets$group<-c("#junzhe")

tweets_all<-rbind(zzh_tweets,gj_tweets,jz_tweets)
#get tweet counts per user
tw_counts_user<-table(tweets_all$screen_name)%>%
    as.data.frame() %>%
    arrange(desc(Freq))
names(tw_counts_user)<-c("user","n_tweets")

#get account names from venn slices
venn_slices <- attr(v.table,"intersections")


#tweet counts for accounts in each venn slice
get_tweet_counts<-function(index,data,tweets_count){
  data<-as.data.frame(data)
  names(data)<-c("user")
  data<-merge(data,tweets_count,by=c("user"))
  tweets_sum<-sum(data$n_tweets)
  data %>%
    arrange(desc(n_tweets)) %>%
    top_n(15) %>%
    mutate(user = reorder(user, n_tweets)) %>%
    ggplot(aes(x = user, y = n_tweets)) +
    geom_col() +
    xlab(NULL) +
    coord_flip() +
    ggtitle(paste("top accounts tweeting about",index,", n=",tweets_sum,"tweets"))

}

lapply(seq_along(venn_slices), function(i) get_tweet_counts(names(x)[[i]], x[[i]],tw_counts_user))
```

```
## Selecting by n_tweets
## Selecting by n_tweets
## Selecting by n_tweets
## Selecting by n_tweets
## Selecting by n_tweets
## Selecting by n_tweets
## Selecting by n_tweets
```

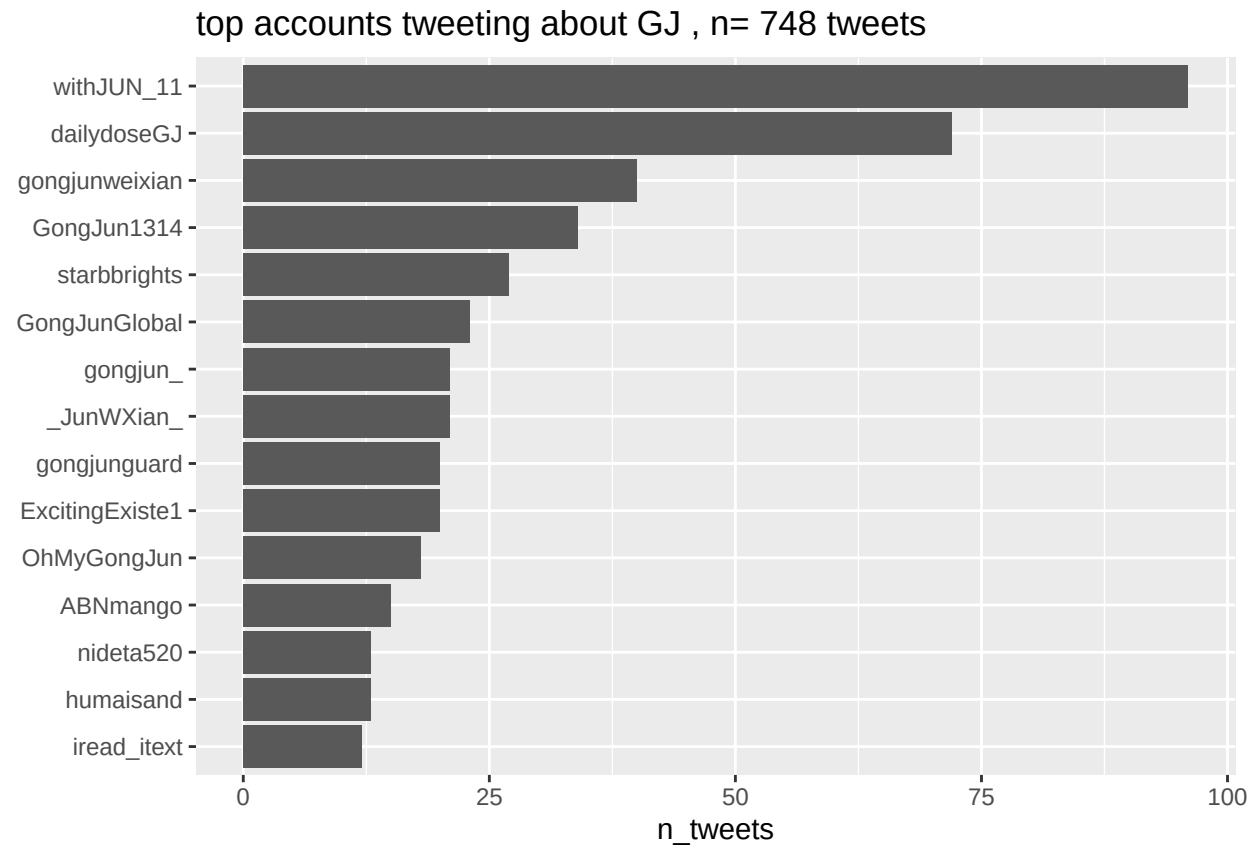
```
## [[1]]
```


top accounts tweeting about ZZH:JunZhe , n= 28 tweets
chosekiei
ZhehaNesian
Cathy70980630
agikun
0
5
10
15
n_tweets

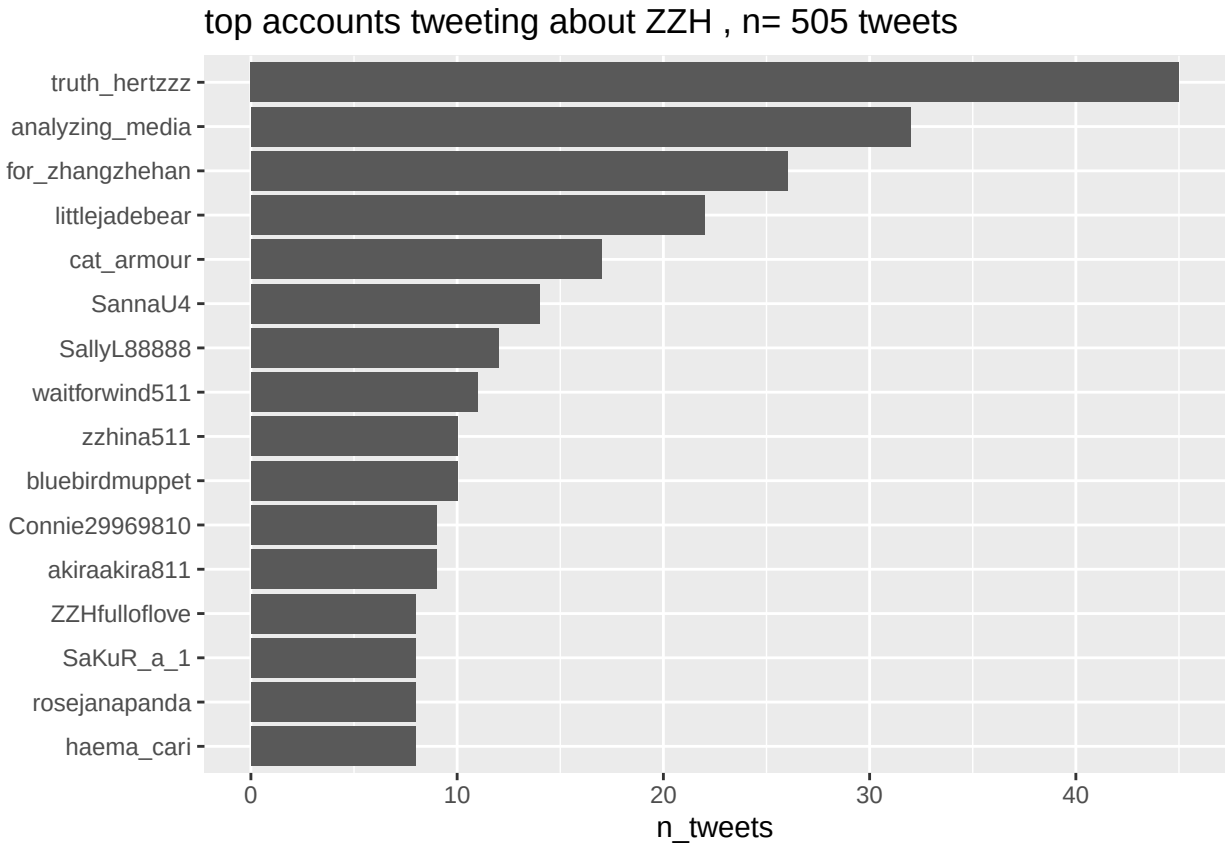
```
##
## [[2]]
```


top accounts tweeting about ZZH:JunZhe:GJ , n= 495 tweets
jujoop09
animeota
TrulyTelesis
mk_chronos
MiniOrchid1
miapiapia
Rei_remediating
Karenkk_axel
4_SeasonManor
dillylene
Blueclue511
woaiwangxian
p2ytansmile
vi_dao
KexingsPoison
East_Of_Hell
0
10
20
30
40
n_tweets

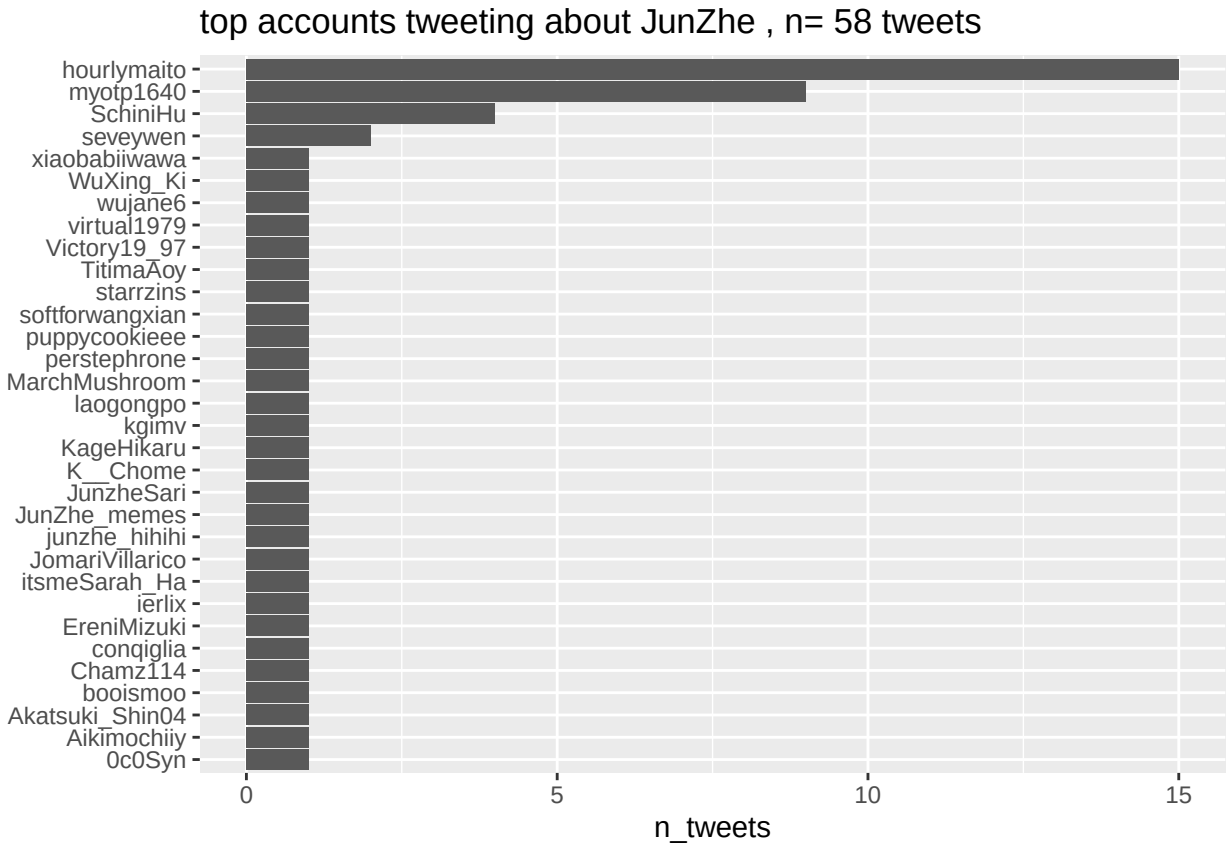
```
##
## [[3]]
```


top accounts tweeting about GJ , n= 748 tweets
withJUN_11
dailydoseGJ
gongjunweixian
GongJun1314
starbbrights
GongJunGlobal
gongjun_
JunWXian
gongjunguard
ExcitingExiste1
OhMyGongJun
ABNmango
nideta520
humaisand
iread_itext
0
25
50
75
100
n_tweets

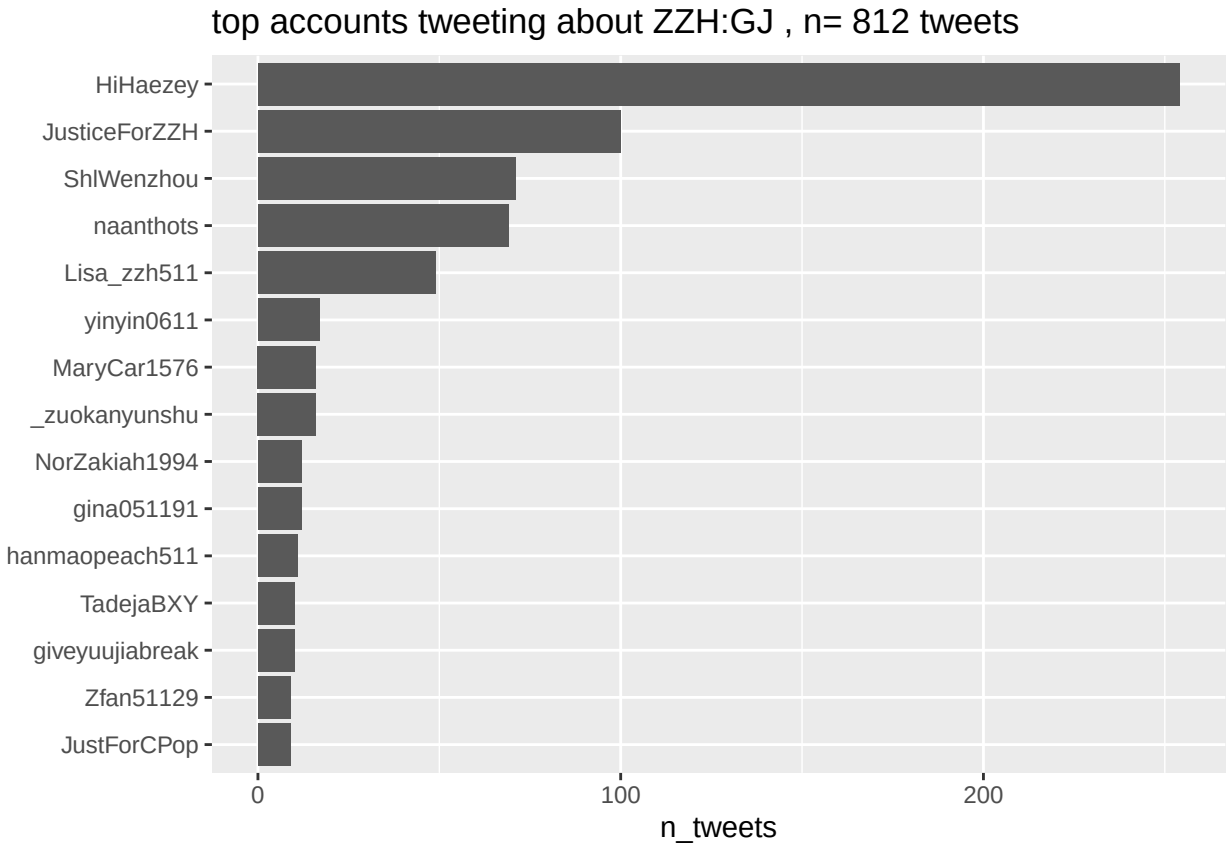
```
##
## [[4]]
```


top accounts tweeting about ZZH , n= 505 tweets
truth_hertzzz
analyzing_media
for_zhangzhehan
littlejadebear
cat_armour
SannaU4
SallyL88888
waitforwind511
zzhina511
bluebirdmuppet
Connie29969810
akiraakira811
ZZHfulloflove
SaKuR_a_1
rosejanapanda
haema_cari
0
10
20
30
40
n_tweets

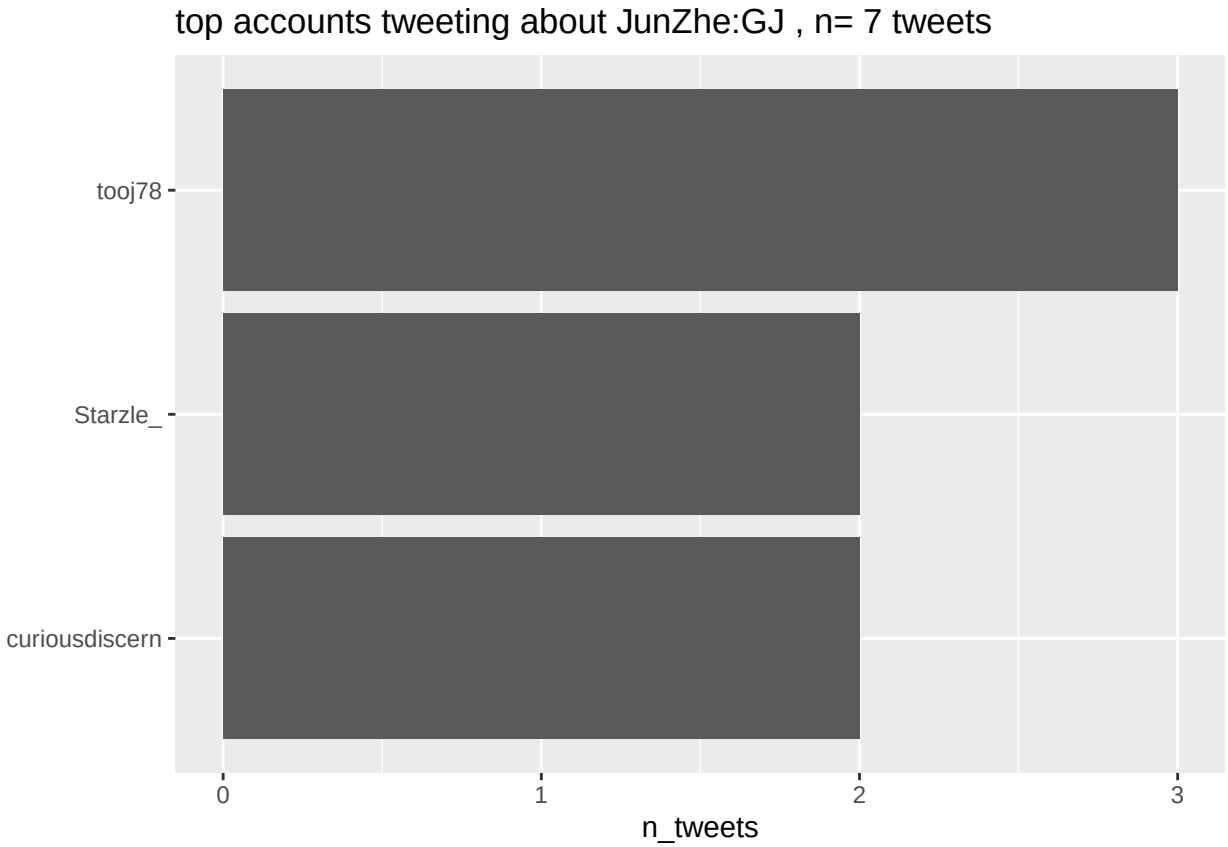
```
##
## [[5]]
```


top accounts tweeting about JunZhe , n= 58 tweets
hourlymaito
myotp1640
SchiniHu
seveywen
xiaobabiiwawa
WuXing_Ki
wujane6
virtual1979
Victory19_97
TitimaĀoy
starrzins
softforwangxian
puppycookieee
perstephrone
MarchMushroom
laogongpo
kgimv
KageHikaru
K__Chome
JunzheSari
JunZhe_memes
junzhe_hihihi
JomariVillarico
itsmeSarah_Ha
ierlix
EreniMizuki
conqiglia
Chamz114
booismoo
Akatsuki_Shin04
Aikimochiiy
0c0Syn
0
5
10
15
n_tweets

```
##
## [[6]]
```


top accounts tweeting about ZZH:GJ , n= 812 tweets
HiHaezey
JusticeForZZH
ShlWenzhou
naanthots
Lisa_zzh511
yinyin0611
MaryCar1576
_zuokanyunshu
NorZakiah1994
gina051191
hanmaopeach511
TadejaBXY
giveyuujiabreak
Zfan51129
JustForCPop
0
100
200
n_tweets

```
##
## [[7]]
```



Now I have information on which accounts are also tweeting about the other two hashtags I looked at. Will now add this information to initial bar plot showing tweet counts per hashtag

```
#extract user names for each slice of venn diagram
#transfer user names and slice association to new data frame
add_venn_slice_info<-function(index,data){
  df_out<-data.frame(data,index)
  names(df_out)<-c("screen_name","venn_slice")
  return(df_out)
}

venn_slice_users<-lapply(seq_along(venn_slices), function(i) add_venn_slice_info(names(x)[[i]], x[[i]]))
tweets_all<-(merge(tweets_all,venn_slice_users,by="screen_name"))

unique(venn_slice_users$venn_slice)
```

```
## [1] "ZZH:JunZhe"    "ZZH:JunZhe:GJ" "GJ"            "ZZH"
## [5] "JunZhe"        "ZZH:GJ"        "JunZhe:GJ"
```

```
ggplot(tweets_all,aes(x=group,fill=venn_slice))+
  geom_bar()+
  scale_fill_manual(values = c("ZZH:JunZhe"="blueviolet",
            "ZZH:JunZhe:GJ" ="purple",
            "GJ" = "red",
            "ZZH" = "blue",
            "JunZhe" =  "violet",
```

```
            "ZZH:GJ" = "yellow",
            "JunZhe:GJ"= "magenta"))
```