Project Mid-Term

MATH6640
Topics in Statistics

Analyzing Members' Participation Trend in the #edchat Online Community

Submitted by:
Safia Rahmat

PROBLEM STATEMENT

Analyzing Members' Participation Trend in the #edchat Online Community

Introduction

#edchat is an influential micro blogging communities for educators' professional development. It started in 2009 and has been consistently identified as one of the most popular hash tags in education by a number of websites. One of its most important activities is the weekly synchronous chat where members from all over the world participate and have discussion on a selected topic. More specifically, #edchat hosts two synchronous chats every Tuesday at 12pm NYC (5 pm UK), and at 7 pm NYC (12 am UK). It starts with the facilitator posting the chat topic, and then all the participants joined the discussion about the topic.

This project aims at extending our understanding of microblogging-based learning communities by examining members' participation in the weekly online synchronous chats from 2009 to 2015.

Here are the major research questions:

- How many participants contributed to in the synchronous online chats hosted by the #edchat community? Is the number of participants increasing or decreasing over the past five years?
- How does the level of participation vary among the contributing members based on the number of tweets generated? Do some members contribute much more than others during a certain period of time?
- How often do new members join the community's online chatting events? Does that trend stay the same over the past five years?

Database Structure

A Database named TwitterDB was created to store all the synchronous chats generated by the #edchat community from 2009 to 2015. It consists of two primary tables: (a) Twitter Table and (b) Members_Tweets Table.

Members_Tweet Table:
This table consists of ten columns.

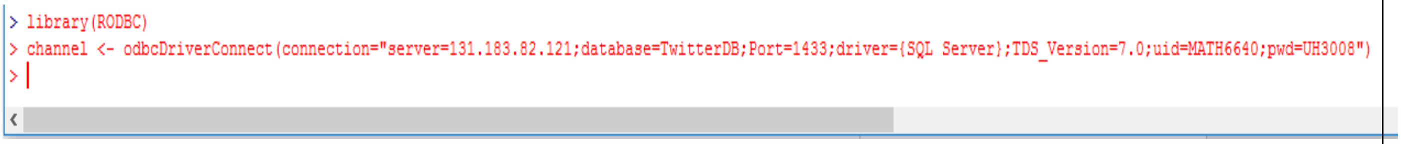
(a) ID : Each tweet is given a unique Id to identify it uniquely from other tweets.
(b) CHAT_ID: Each chat consists of a number of tweets. Each tweet is assigned a unique ID which is similar to chat id to identify a tweet from a particular chat. It is created as a bigint data type.
(c) Username: The username of the participant who posted the tweet. It is created as a nvarchar data type of size 255.
(d) Date: The date of a tweet. It is created as a Date data type.
(e) Year: The year when the tweet was posted
(f) Month: The month when the tweet was posted
(g) Status: The content of tweets.

Connection to SQL Server:
ip: 131.183.82.121
login: MATH6640
password: UH3008

Example by using R:

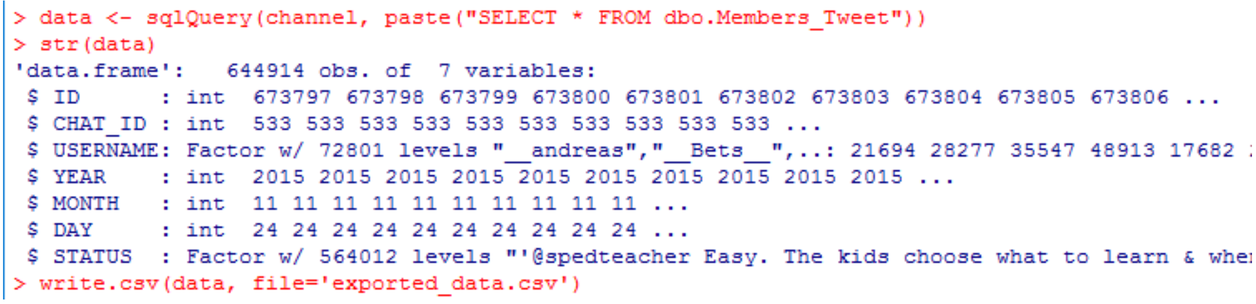
library(RODBC)
channel <-
odbcDriverConnect(connection="server=131.183.82.121;database=TwitterDB;Port=1433;driver={SQL Server};TDS_Version=7.0;uid=MATH6640;pwd=UH3008")
data <- sqlQuery(channel, paste("SELECT * FROM dbo.Members_Tweet"))

Connect to the server using R:



```
> library(RODBC)
> channel <- odbcDriverConnect(connection="server=131.183.82.121;database=TwitterDB;Port=1433;driver={SQL Server};TDS_Version=7.0;uid=MATH6640;pwd=UH3008")
> 
```

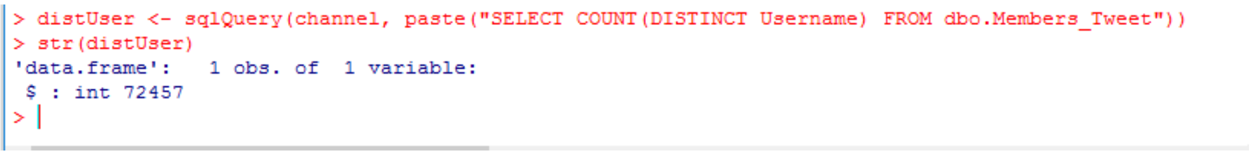
Storing the values of database in CSV file:



```
> data <- sqlQuery(channel, paste("SELECT * FROM dbo.Members_Tweet"))
> str(data)
'data.frame':	644914 obs. of  7 variables:
 $ ID      : int  673797 673798 673799 673800 673801 673802 673803 673804 673805 673806 ...
 $ CHAT_ID : int  533 533 533 533 533 533 533 533 533 533 ...
 $ USERNAME: Factor w/ 72801 levels "__andreas","__Bets__",..: 21694 28277 35547 48913 17682 :
 $ YEAR    : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ MONTH   : int  11 11 11 11 11 11 11 11 11 11 ...
 $ DAY     : int  24 24 24 24 24 24 24 24 24 24 ...
 $ STATUS  : Factor w/ 564012 levels "'@spedteacher Easy. The kids choose what to learn & whe
> write.csv(data, file='exported_data.csv')
```

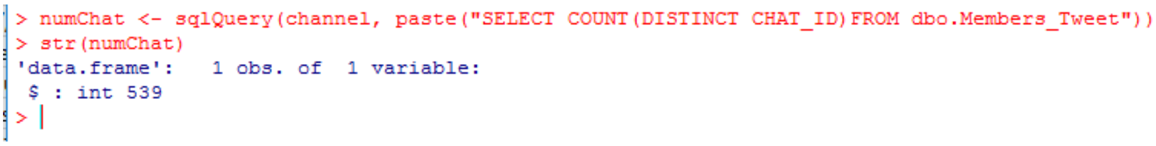
Queries

1. Total number of distinct users



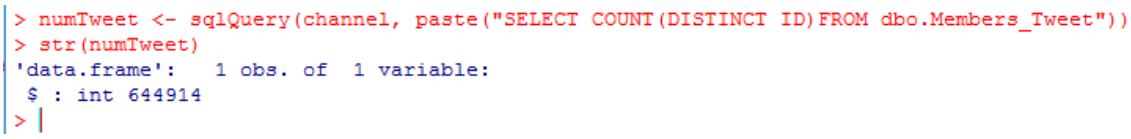
```
> distUser <- sqlQuery(channel, paste("SELECT COUNT(DISTINCT Username) FROM dbo.Members_Tweet"))
> str(distUser)
'data.frame':	1 obs. of  1 variable:
 $ : int 72457
> 
```

2. Total number of chats



```
> numChat <- sqlQuery(channel, paste("SELECT COUNT(DISTINCT CHAT_ID)FROM dbo.Members_Tweet"))
> str(numChat)
'data.frame':	1 obs. of  1 variable:
 $ : int 539
> 
```

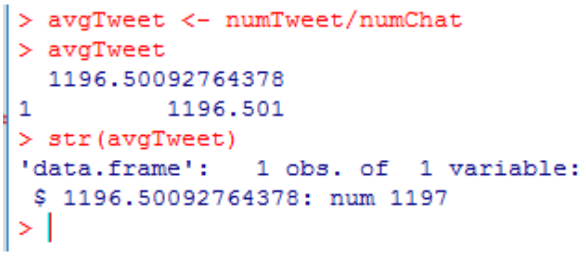
3. Total number of tweets



```
> numTweet <- sqlQuery(channel, paste("SELECT COUNT(DISTINCT ID)FROM dbo.Members_Tweet"))
> str(numTweet)
'data.frame':	1 obs. of  1 variable:
 $ : int 644914
> 
```

4. Average number of tweets in a chat (mean, SD, Median, Mode, Maximum/Minimum)

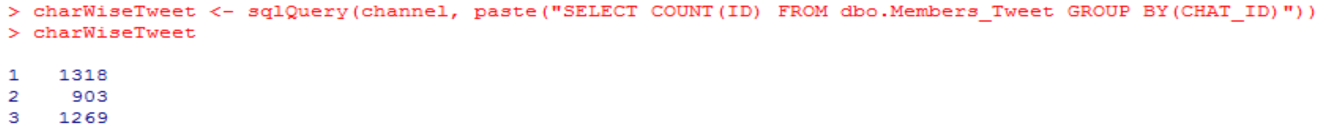
Average/Mean:



```
> avgTweet <- numTweet/numChat
> avgTweet
  1196.50092764378
1         1196.501
> str(avgTweet)
'data.frame':   1 obs. of  1 variable:
 $ 1196.50092764378: num 1197
> 
```

Standard Deviation:

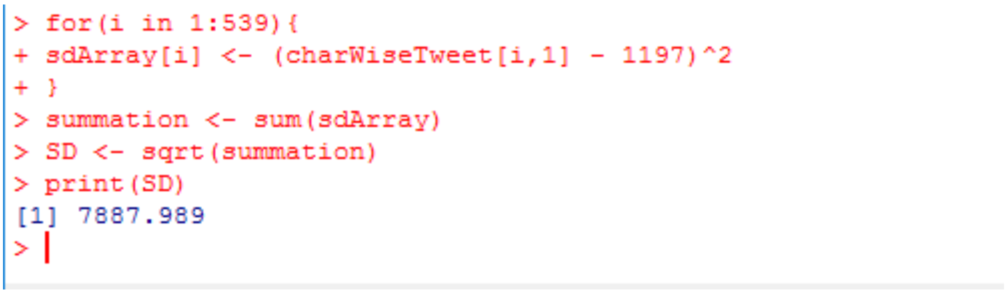
Sorting the tweets chat-wise:



```
> charWiseTweet <- sqlQuery(channel, paste("SELECT COUNT(ID) FROM dbo.Members_Tweet GROUP BY(CHAT_ID)"))
> charWiseTweet

1   1318
2    903
3   1269
```

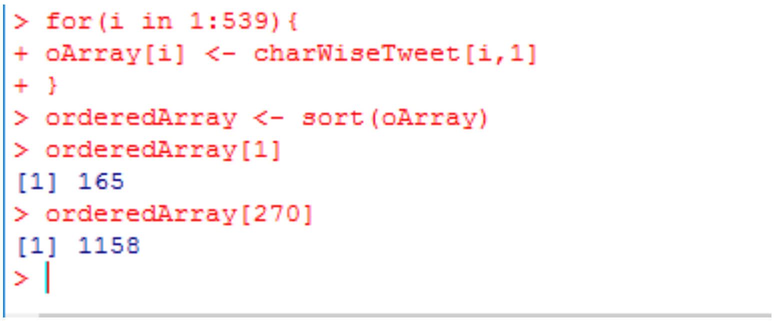
finding standard deviation:



```
> for(i in 1:539){
+ sdArray[i] <- (charWiseTweet[i,1] - 1197)^2
+ }
> summation <- sum(sdArray)
> SD <- sqrt(summation)
> print(SD)
[1] 7887.989
> 
```

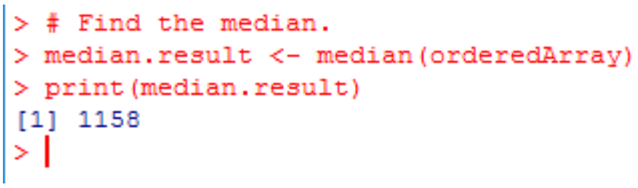
Median:

Total number of distinct chats: 539
Median of tweets i.e the number of tweets for CHAT_ID (539/2) : 1158
orderedArray contains the elements in ascending order.



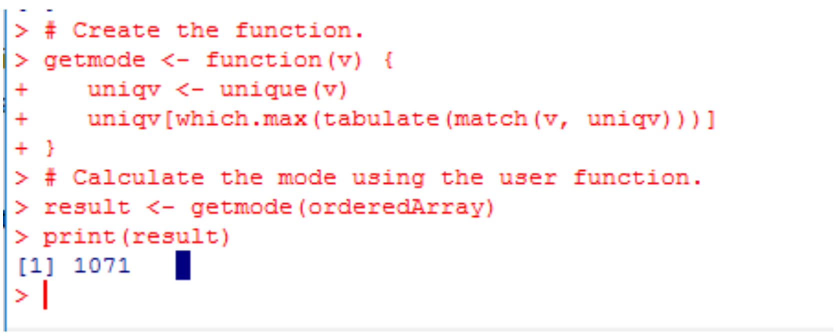
```
> for(i in 1:539){
+ oArray[i] <- charWiseTweet[i,1]
+ }
> orderedArray <- sort(oArray)
> orderedArray[1]
[1] 165
> orderedArray[270]
[1] 1158
> 
```

Another method for median:



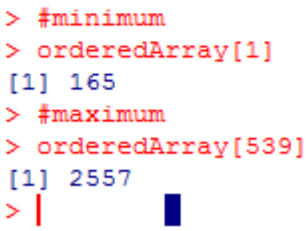
```
> # Find the median.
> median.result <- median(orderedArray)
> print(median.result)
[1] 1158
> 
```

Mode:



```
> # Create the function.
> getmode <- function(v) {
+    uniqv <- unique(v)
+    uniqv[which.max(tabulate(match(v, uniqv)))]
+ }
> # Calculate the mode using the user function.
> result <- getmode(orderedArray)
> print(result)
[1] 1071
> 
```

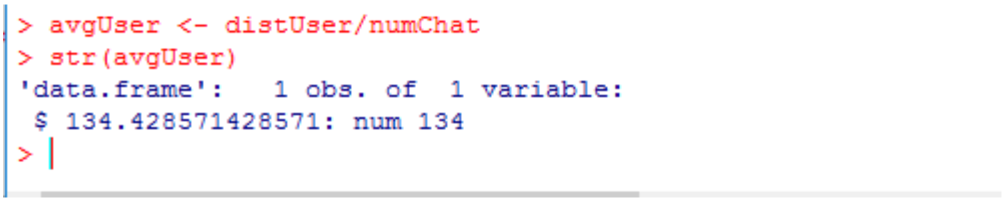
Minimum/Maximum:



```
> #minimum
> orderedArray[1]
[1] 165
> #maximum
> orderedArray[539]
[1] 2557
> 
```

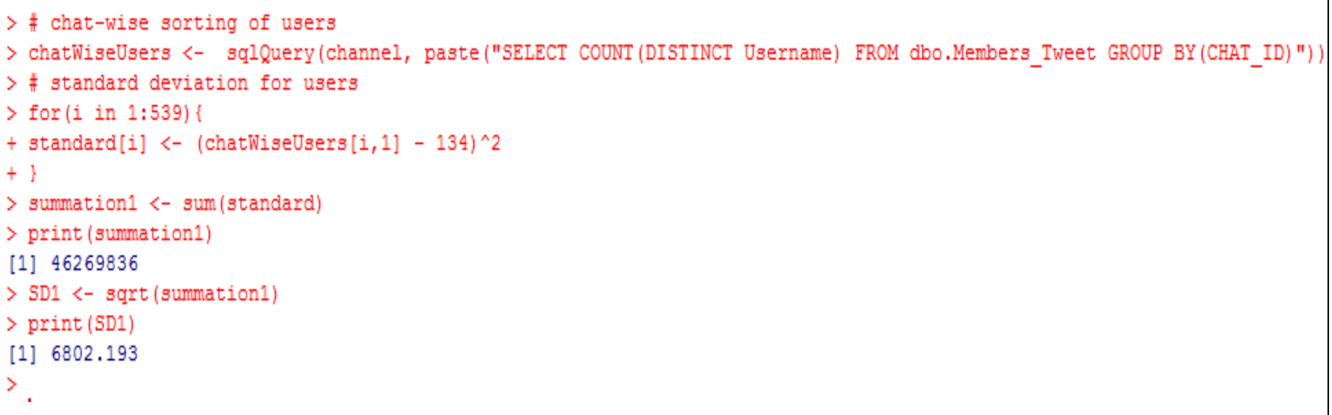
5. Average number of users in a chat (mean, SD, Median, Mode, Maximum/Minimum)

Average/Mean:



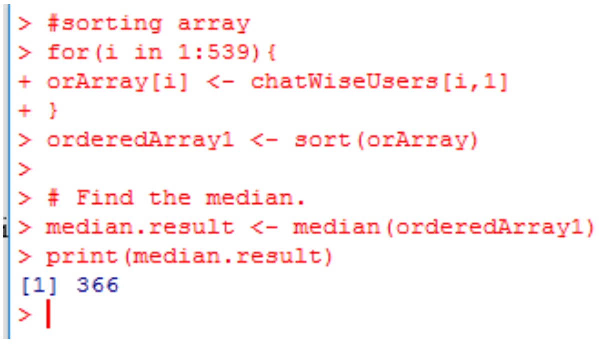
```
> avgUser <- distUser/numChat
> str(avgUser)
'data.frame':   1 obs. of  1 variable:
 $ 134.428571428571: num 134
> 
```

Standard Deviation:



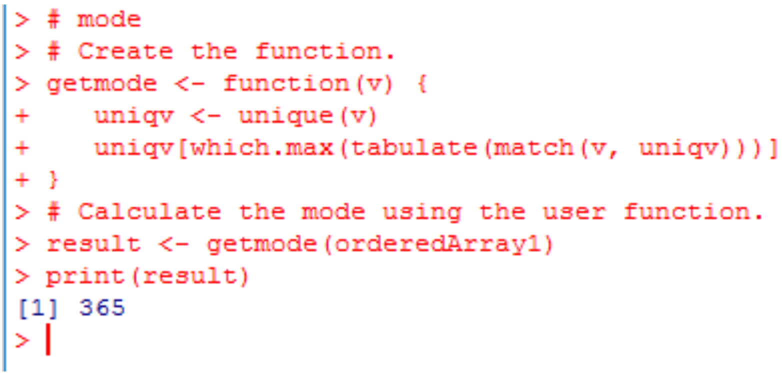
```
> # chat-wise sorting of users
> chatWiseUsers <-  sqlQuery(channel, paste("SELECT COUNT(DISTINCT Username) FROM dbo.Members_Tweet GROUP BY(CHAT_ID)"))
> # standard deviation for users
> for(i in 1:539){
+ standard[i] <- (chatWiseUsers[i,1] - 134)^2
+ }
> summation1 <- sum(standard)
> print(summation1)
[1] 46269836
> SD1 <- sqrt(summation1)
> print(SD1)
[1] 6802.193
> 
```

Median:



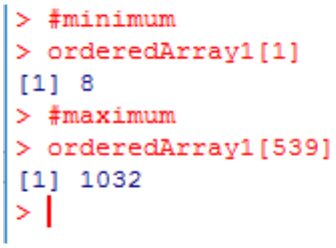
```
> #sorting array
> for(i in 1:539){
+ orArray[i] <- chatWiseUsers[i,1]
+ }
> orderedArray1 <- sort(orArray)
>
> # Find the median.
> median.result <- median(orderedArray1)
> print(median.result)
[1] 366
>
```

Mode:



```
> # mode
> # Create the function.
> getmode <- function(v) {
+    uniqv <- unique(v)
+    uniqv[which.max(tabulate(match(v, uniqv)))]
+ }
> # Calculate the mode using the user function.
> result <- getmode(orderedArray1)
> print(result)
[1] 365
>
```

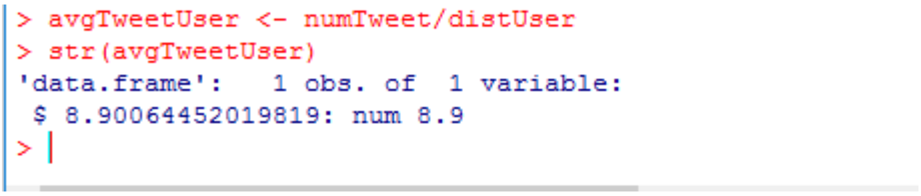
Minimum/Maximum:



```
> #minimum
> orderedArray1[1]
[1] 8
> #maximum
> orderedArray1[539]
[1] 1032
>
```

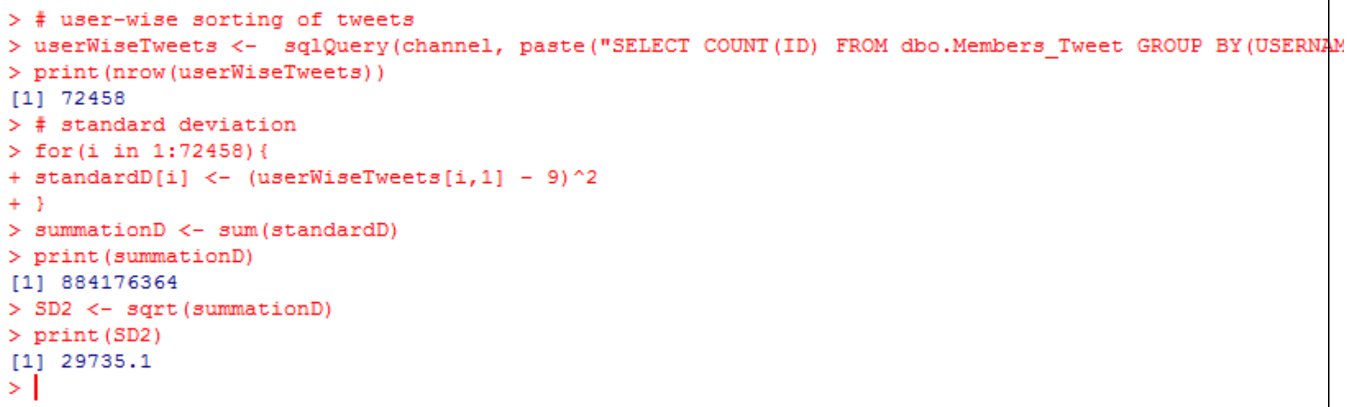
6. Average number of tweets contributed by a user (mean, SD, Median, Mode, Maximum/Minimum)

Average/Mean:



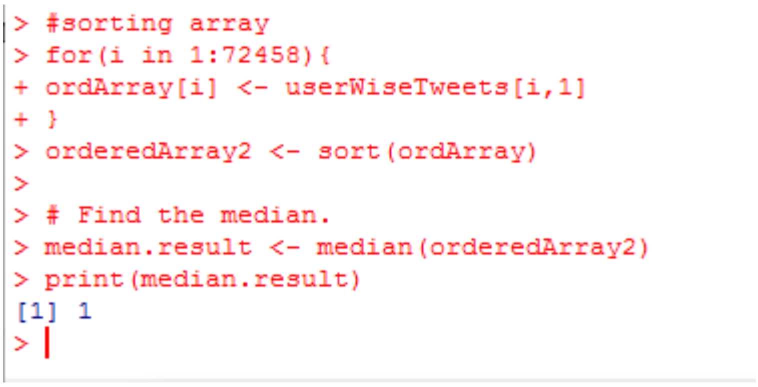
```
> avgTweetUser <- numTweet/distUser
> str(avgTweetUser)
'data.frame':   1 obs. of  1 variable:
 $ 8.90064452019819: num 8.9
>
```

Standard Deviation:



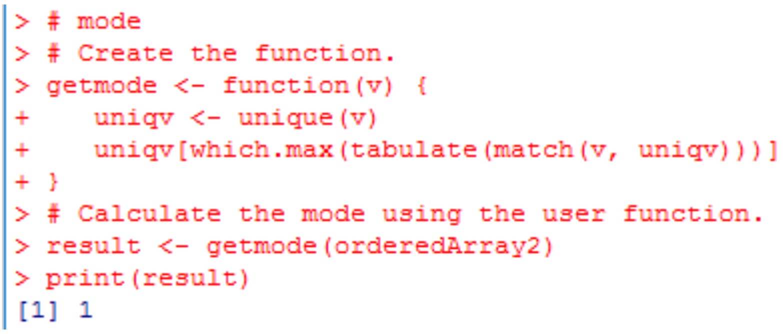
```
> # user-wise sorting of tweets
> userWiseTweets <-  sqlQuery(channel, paste("SELECT COUNT(ID) FROM dbo.Members_Tweet GROUP BY(USERNAM
> print(nrow(userWiseTweets))
[1] 72458
> # standard deviation
> for(i in 1:72458){
+ standardD[i] <- (userWiseTweets[i,1] - 9)^2
+ }
> summationD <- sum(standardD)
> print(summationD)
[1] 884176364
> SD2 <- sqrt(summationD)
> print(SD2)
[1] 29735.1
> 
```

Median:



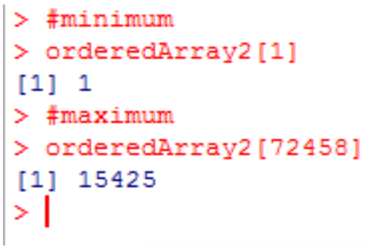
```
> #sorting array
> for(i in 1:72458){
+ ordArray[i] <- userWiseTweets[i,1]
+ }
> orderedArray2 <- sort(ordArray)
>
> # Find the median.
> median.result <- median(orderedArray2)
> print(median.result)
[1] 1
> 
```

Mode:



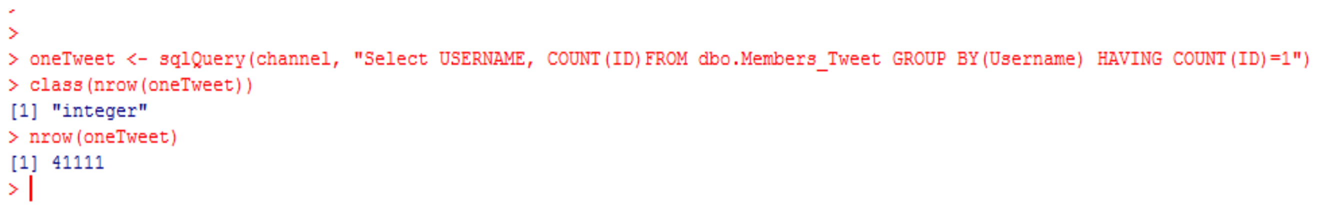
```
> # mode
> # Create the function.
> getmode <- function(v) {
+    uniqv <- unique(v)
+    uniqv[which.max(tabulate(match(v, uniqv)))]
+ }
> # Calculate the mode using the user function.
> result <- getmode(orderedArray2)
> print(result)
[1] 1
```

Minimum/Maximum:



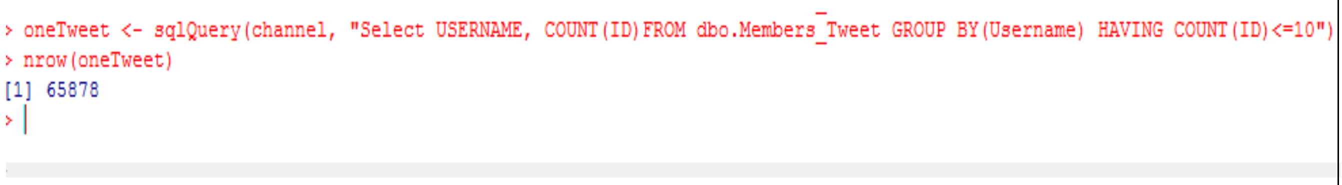
```
> #minimum
> orderedArray2[1]
[1] 1
> #maximum
> orderedArray2[72458]
[1] 15425
> 
```

7. Number/Percentage of users that contributed only 1 tweet



```
>
> oneTweet <- sqlQuery(channel, "Select USERNAME, COUNT(ID)FROM dbo.Members_Tweet GROUP BY(Username) HAVING COUNT(ID)=1")
> class(nrow(oneTweet))
[1] "integer"
> nrow(oneTweet)
[1] 41111
>
```

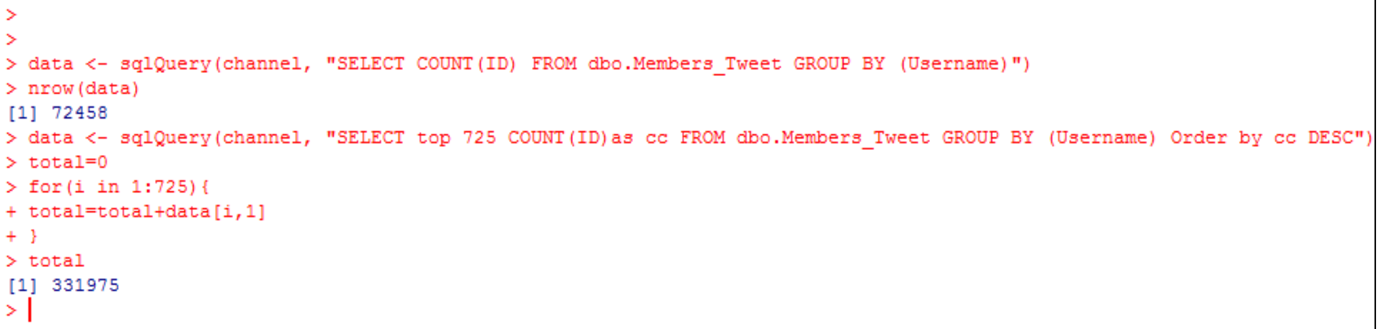
8. Number/Percentage of users that contributed less than or equal to 10 tweets



```
> oneTweet <- sqlQuery(channel, "Select USERNAME, COUNT(ID)FROM dbo.Members_Tweet GROUP BY(Username) HAVING COUNT(ID)<=10")
> nrow(oneTweet)
[1] 65878
>
```

9. Number/Percentage of tweets contributed by the top 1% users who contributed the highest number of tweets

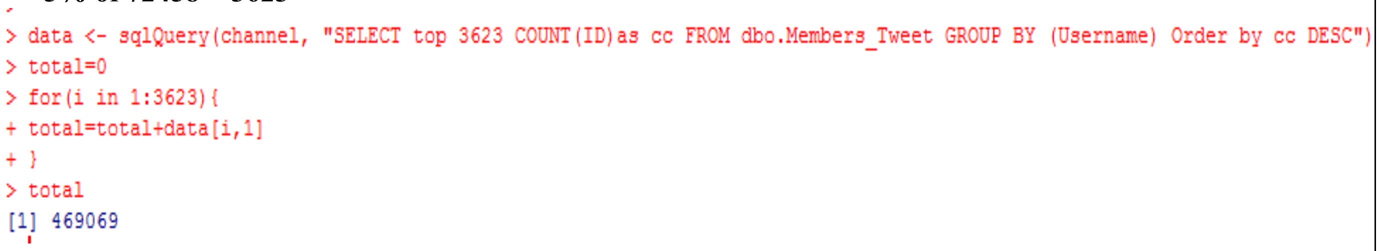
1% of 72458 = 725



```
>
>
> data <- sqlQuery(channel, "SELECT COUNT(ID) FROM dbo.Members_Tweet GROUP BY (Username)")
> nrow(data)
[1] 72458
> data <- sqlQuery(channel, "SELECT top 725 COUNT(ID)as cc FROM dbo.Members_Tweet GROUP BY (Username) Order by cc DESC")
> total=0
> for(i in 1:725){
+ total=total+data[i,1]
+ }
> total
[1] 331975
>
```

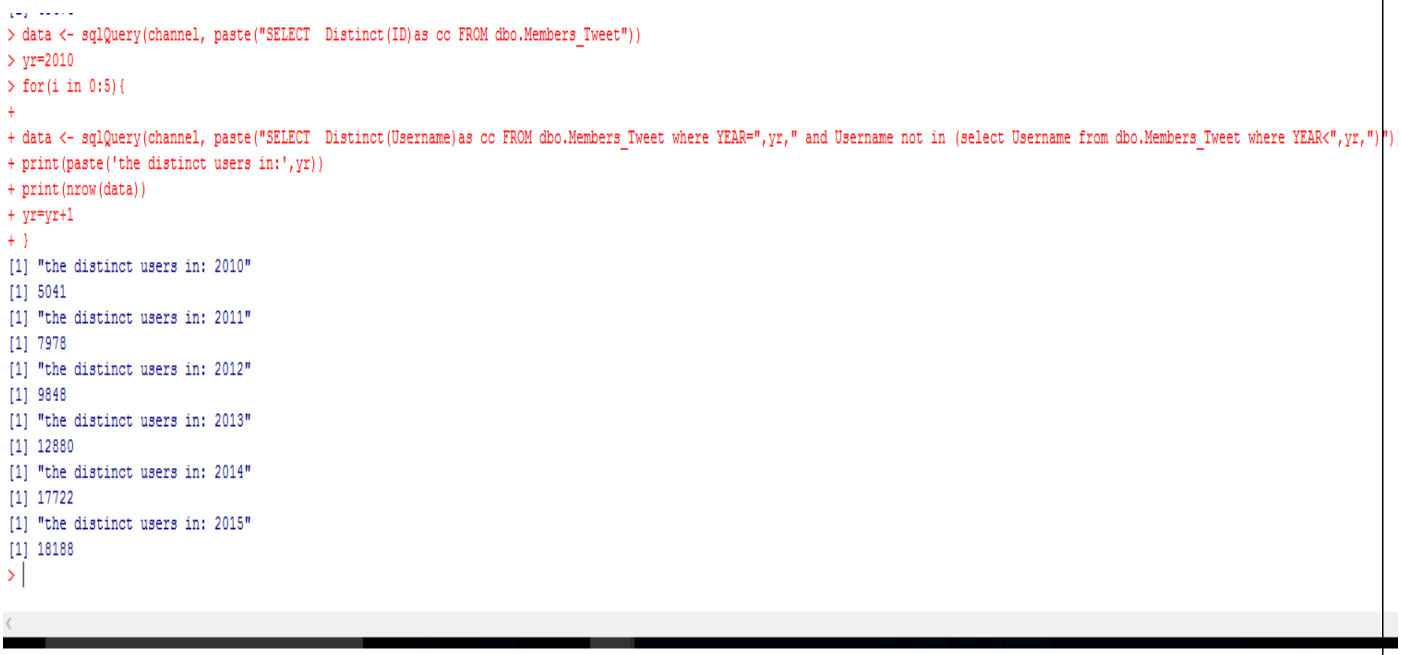
10. Number/Percentage of tweets contributed by the top 5% users

5% of 72458 = 3623

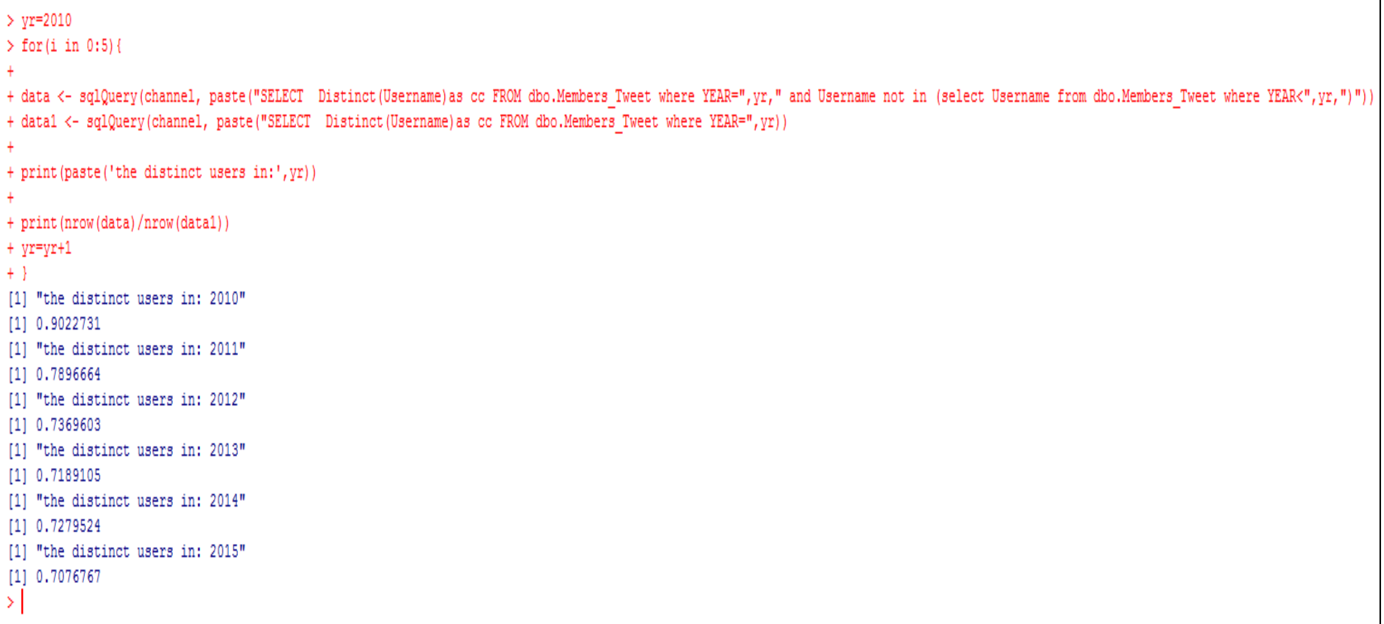


```
> data <- sqlQuery(channel, "SELECT top 3623 COUNT(ID)as cc FROM dbo.Members_Tweet GROUP BY (Username) Order by cc DESC")
> total=0
> for(i in 1:3623){
+ total=total+data[i,1]
+ }
> total
[1] 469069
```

11. Number of new users who joined the chats each year (New users are defined as those who has never participated in any chat before.) Percentages of new users among all those who participated in a given year (2010, 2011, 2012, 2013, 2014, 2015).



```
> data <- sqlQuery(channel, paste("SELECT  Distinct(ID)as cc FROM dbo.Members_Tweet"))
> yr=2010
> for(i in 0:5){
+
+ data <- sqlQuery(channel, paste("SELECT  Distinct(Username)as cc FROM dbo.Members_Tweet where YEAR=",yr," and Username not in (select Username from dbo.Members_Tweet where YEAR<",yr,")"))
+ print(paste('the distinct users in:',yr))
+ print(nrow(data))
+ yr=yr+1
+ }
[1] "the distinct users in: 2010"
[1] 5041
[1] "the distinct users in: 2011"
[1] 7978
[1] "the distinct users in: 2012"
[1] 9848
[1] "the distinct users in: 2013"
[1] 12880
[1] "the distinct users in: 2014"
[1] 17722
[1] "the distinct users in: 2015"
[1] 18188
>
```



```
> yr=2010
> for(i in 0:5){
+
+ data <- sqlQuery(channel, paste("SELECT  Distinct(Username)as cc FROM dbo.Members_Tweet where YEAR=",yr," and Username not in (select Username from dbo.Members_Tweet where YEAR<",yr,")"))
+ data1 <- sqlQuery(channel, paste("SELECT  Distinct(Username)as cc FROM dbo.Members_Tweet where YEAR=",yr))
+
+ print(paste('the distinct users in:',yr))
+
+ print(nrow(data)/nrow(data1))
+ yr=yr+1
+ }
[1] "the distinct users in: 2010"
[1] 0.9022731
[1] "the distinct users in: 2011"
[1] 0.7896664
[1] "the distinct users in: 2012"
[1] 0.7369603
[1] "the distinct users in: 2013"
[1] 0.7189105
[1] "the distinct users in: 2014"
[1] 0.7279524
[1] "the distinct users in: 2015"
[1] 0.7076767
>
```