FIT 1043 Introduction to data science

Assignment 3

Full name: Lang Zolyn

Student ID: 30719704

*Note: Throughout the whole process, the data is **NOT** uncompressed and stored in my disk, it will be piped to gunzip for each line of command, this is to save time and memory, the **whole** CSV file don't need to be loaded in order to display the content*

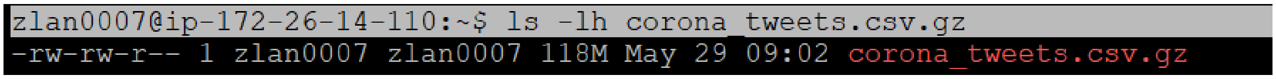
Commands are in one line (due to space limitations it is displayed as multiple lines) and blue.

Introduction

In this assignment, we will be using bash scripting to execute shell commands that will be needed for text processing commands. To analyze the provided data set which is the zipped corona_tweets.csv.gz, we will be using several basic commands such as "grep", "cut" and "awk" in the terminal and the "nano" shell script. Moreover, R programming language will also be needed for data visualization as it is designed mostly for statistical analysis purposes, it will play a good role in this case. Lastly, we will be extracting the necessary information needed to fully analyze this data set.

Part A: Inspecting the data

```
zlan0007@ip-172-26-14-110:~$ ls -lh corona_tweets.csv.gz
```

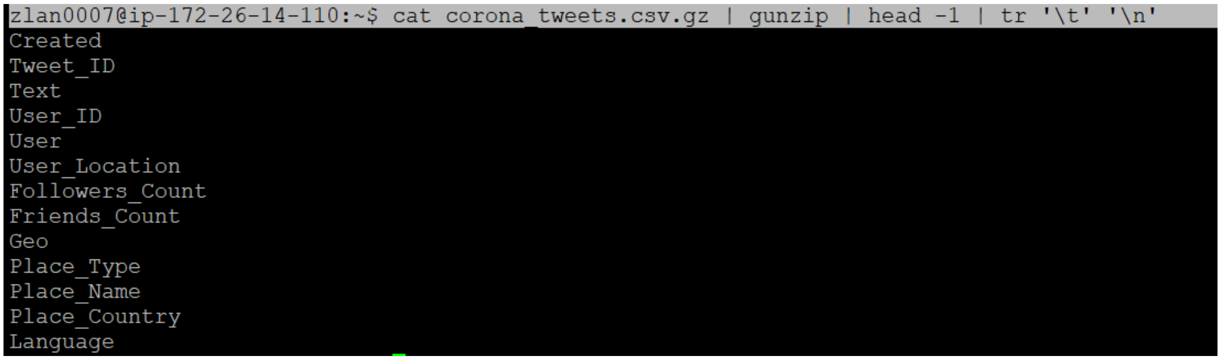


```
zlan0007@ip-172-26-14-110:~$ ls -lh corona_tweets.csv.gz
-rw-rw-r-- 1 zlan0007 zlan0007 118M May 29 09:02 corona_tweets.csv.gz
```

Answer: The size of corona_tweets.csv.gz is 118M .

```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip |
head -1 | tr '\t' '\n'
```

Explanation: In order to obtain the header names, the zipped csv file is piped to the gunzip application to unzip the corona_tweets.csv file, it is then piped to head -1 to display the first line of the corona_tweets.csv file, because it is harder to see the header in a single line, I decided to use the 'tr' command to replace the tabs on the firstline with the newline character.

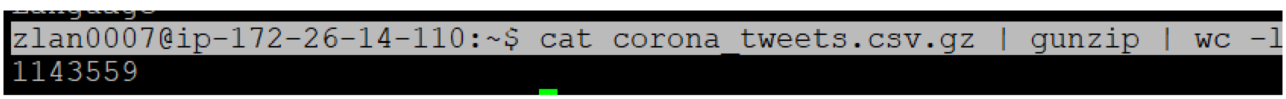


```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | head -1 | tr '\t' '\n'
Created
Tweet_ID
Text
User_ID
User
User_Location
Followers_Count
Friends_Count
Geo
Place_Type
Place_Name
Place_Country
Language
```

Answer: As shown as above, the header names in the corona_tweets.csv are "Created" , "Tweet_ID" , "Text" ,"User_ID", "User","User_Location","Followers_Count","Friends_Count","Geo","Place_Type", "Place_Name","Place_Country" and "Language".

zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | wc –l

Explanation: To get the number of lines in the dataset, I piped it to the wordcount "wc" command with the '-l' flag to count the total number of lines.



```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | wc -l
1143559
```

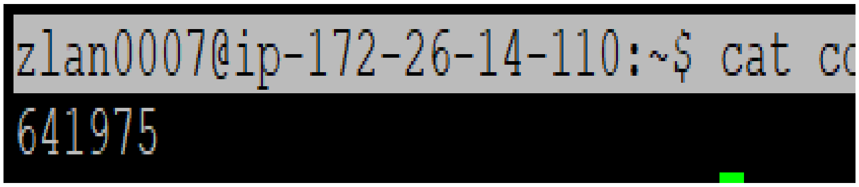
Answer: The corona_tweets.csv.gz file contains 1143559 lines.

Part B: Information from data

zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | cut -f4 | tail --lines=+2 | sort -u | wc –l

Explanation:

To obtain the number of unique twitter user, I decided to use the field User_ID, this is because it is a unique attribute. I used the “cut” command and the “-f” flag which indicates which column is used, in this case I used “User_ID” which is the column number 4. Then, I piped it to “tail – lines=+2” , because the header of the column should not be included, it was then sorted by using the “-u” flag to only output the unique values, I piped it to the “wc” command with the “-l” flag to count the total number of lines of the unique twitter user.

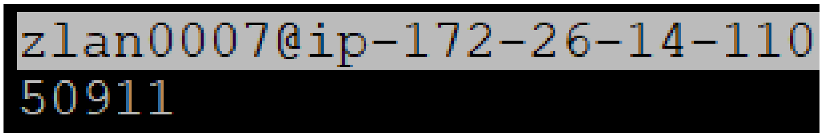


```
zlan0007@ip-172-26-14-110:~$ cat cc
641975
```

Answer: There are 641975 unique twitter users.

zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F '\t' '{ print $3}' | grep -i "death" | wc –l

Explanation: I used the “awk” command to only print the 3rd column which is the “Text” column and the “-F” flag to indicate the deliminater used was a tab character, then I piped it to “grep” and the “-i” flag which will include all the text that contains the word death ignoring whether it is spelled in uppercase or lowercase. It is then piped into the “wc” command and “-l” flag to count the number of lines.



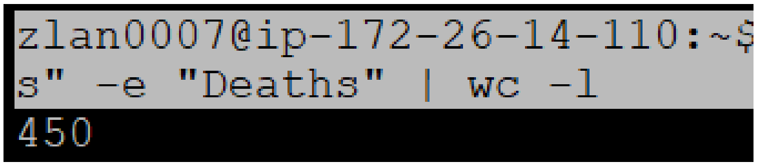
```
zlan0007@ip-172-26-14-110
50911
```

Answer: 50911 tweets mentioned the word “death” in any combination of uppercase or lowercase letters.

```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F '\t' '{print $3}' | grep -i "death\|deaths" | grep -v -e "death" -e "Death" -e "deaths" -e "Deaths" | wc –l
```

Explanation:

To get all the tweets that mentioned death and deaths but are not spelt properly, I used the "awk" and the "-F" flag to indicate the delimeter is a tab character, to only print column 3, which is the "Text" column. Next, I used "grep" and the "-i" flag to include all the tweets that contains the word death and deaths ignoring whether it is spelt in uppercase or lowercase letter. Afterthat, I piped it using "grep" and "-v " flag and "-e" flag which excludes all the word that are spelt in "death", "Death","deaths" and "Deaths" as requested in the question.



```
zlan0007@ip-172-26-14-110:~$
s" -e "Deaths" | wc -l
450
```

Answer: 450 tweets are not spelt exactly "death", "deaths", "Death" or "Deaths" but in other combination of uppercase and lowercase.

Outputting the specific lines from part 2(b) into a file called myText.txt

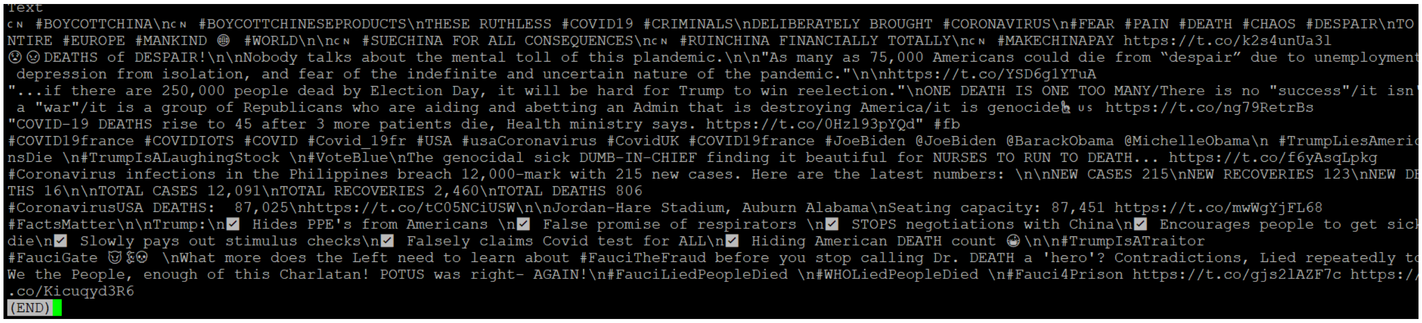
Explanation:

I used the "sort" command and "-u" flag which will only output the unique lines and then I used the ">" character which will write the string before it to the myText.text file as the file name is stated after the character

```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F '\t' '{print $3}' | grep -i "death\|deaths" | grep -v -e "death" -e "Death" -e "deaths" -e "Deaths" | sort -u > myText.txt
```

I used "less" to have a look at the output of the first 10 lines

```
zlan0007@ip-172-26-14-110:~$  head -10 myText.txt | less
```



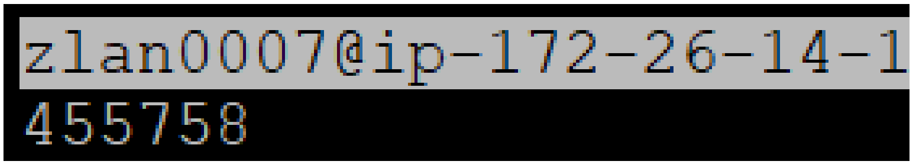
```
Text
🇨🇳 #BOYCOTTCHINA\n🇨🇳 #BOYCOTTCHINESEPRODUCTS\nTHESE RUTHLESS #COVID19 #CRIMINALS\nDELIBERATELY BROUGHT #CORONAVIRUS\n#FEAR #PAIN #DEATH #CHAOS #DESPAIR\nTO
NTIRE #EUROPE #MANKIND 🌐 #WORLD\n\n🇨🇳 #SUECHINA FOR ALL CONSEQUENCES\n🇨🇳 #RUINCHINA FINANCIALLY TOTALLY\n🇨🇳 #MAKECHINAPAY https://t.co/k2s4unUa3l
😧😠DEATHS of DESPAIR!\n\nNobody talks about the mental toll of this plandemic.\n\n"As many as 75,000 Americans could die from “despair” due to unemployment
 depression from isolation, and fear of the indefinite and uncertain nature of the pandemic."\n\nhttps://t.co/YSD6g1YTuA
"...if there are 250,000 people dead by Election Day, it will be hard for Trump to win reelection."\nONE DEATH IS ONE TOO MANY/There is no "success"/it isn'
 a "war"/it is a group of Republicans who are aiding and abetting an Admin that is destroying America/it is genocide🗽🇺🇸 https://t.co/ng79RetrBs
"COVID-19 DEATHS rise to 45 after 3 more patients die, Health ministry says. https://t.co/0Hzl93pYQd" #fb
#COVID19france #COVIDIOTS #COVID #Covid_19fr #USA #usaCoronavirus #CovidUK #COVID19france #JoeBiden @JoeBiden @BarackObama @MichelleObama\n #TrumpLiesAmeric
nsDie \n#TrumpIsALaughingStock \n#VoteBlue\nThe genocidal sick DUMB-IN-CHIEF finding it beautiful for NURSES TO RUN TO DEATH... https://t.co/f6yAsqLpkg
#Coronavirus infections in the Philippines breach 12,000-mark with 215 new cases. Here are the latest numbers: \n\nNEW CASES 215\nNEW RECOVERIES 123\nNEW DE
THS 16\n\nTOTAL CASES 12,091\nTOTAL RECOVERIES 2,460\nTOTAL DEATHS 806
#CoronavirusUSA DEATHS:  87,025\nhttps://t.co/tC05NCiUSW\n\nJordan-Hare Stadium, Auburn Alabama\nSeating capacity: 87,451 https://t.co/mwWgYjFL68
#FactsMatter\n\nTrump:\n☑ Hides PPE's from Americans \n☑ False promise of respirators \n☑ STOPS negotiations with China\n☑ Encourages people to get sick
die\n☑ Slowly pays out stimulus checks\n☑ Falsely claims Covid test for ALL\n☑ Hiding American DEATH count 😱\n\n#TrumpIsATraitor
#FauciGate 😈💱💀 \nWhat more does the Left need to learn about #FauciTheFraud before you stop calling Dr. DEATH a 'hero'? Contradictions, Lied repeatedly to
We the People, enough of this Charlatan! POTUS was right- AGAIN!\n#FauciLiedPeopleDied \n#WHOLiedPeopleDied \n#Fauci4Prison https://t.co/gjs2lAZF7c https://
.co/Kicuqyd3R6
(END)
```

Part C: Data aggregation

Grouping the twitter user (ID) by the number of followers that they have into different ranges.

zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7<=1000{print $4}' | sort -u | wc –l

Explanation: I used “awk” to only print column 4 which is the User_ID that have followers_count (column 7) with the amount less than or equal to 1000 followers, then I piped it to the “sort” command and “-u” flag to only print the unique User_ID, it is then piped into the “wc” command and “-l” flag to count the number of lines.

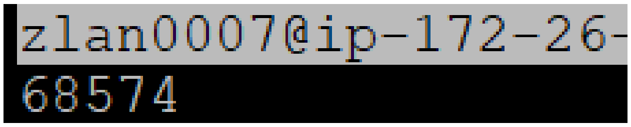


```
zlan0007@ip-172-26-14-1
455758
```

Answer: There are 455758 twitter user that have less than or equal to 1000 followers.

For each of the following code, I used the same method but different range.

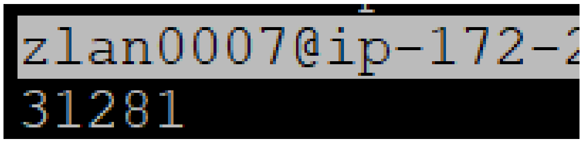
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=1001 && $7<=2000{print $4}' | sort -u | wc –l



```
zlan0007@ip-172-26-
68574
```

Answer: There are 68574 twitter user that have 1001 to 2000 followers.

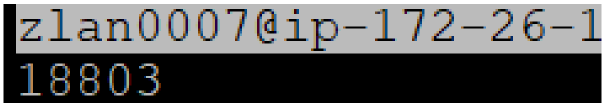
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=2001 && $7<=3000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-2
31281
```

Answer: There are 31281 twitter user that have 2001 to 3000 followers.

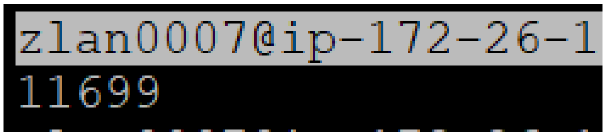
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=3001 && $7<=4000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-1
18803
```

Answer: There are 18803 twitter user that have 3001 to 4000 followers.

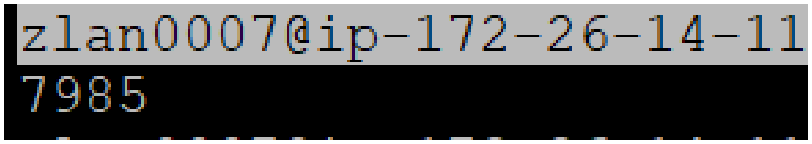
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=4001 && $7<=5000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-1
11699
```

Answer: There are 11699 twitter user that have 4001 to 5000 followers.

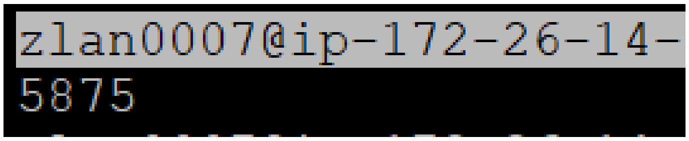
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=5001 && $7<=6000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-11
7985
```

Answer: There are 7985 twitter user that have 5001 to 6000 followers.

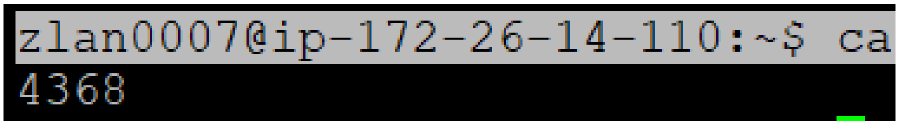
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t'
'$7>=6001 && $7<=7000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-
5875
```

Answer: There are 5875 twitter user that have 6001 to 7000 followers.

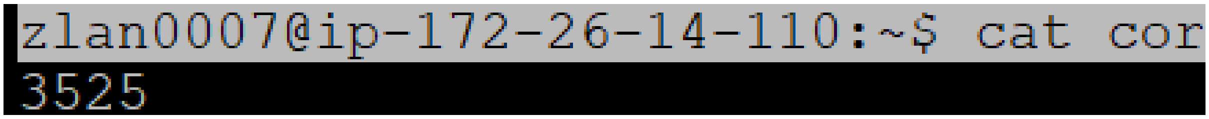
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t'
'$7>=7001 && $7<=8000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ ca
4368
```

Answer: There are 4368 twitter user that have 7001 to 8000 followers.

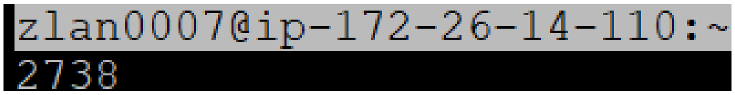
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t'
'$7>=8001 && $7<=9000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ cat cor
3525
```

Answer: There are 3525 twitter user that have 8001 to 9000 followers.

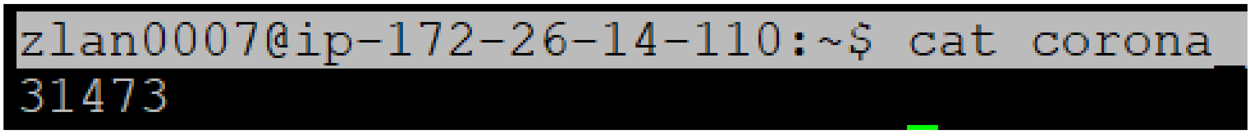
```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t'
'$7>=9001 && $7<=10000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~
2738
```

Answer: There are 2738 twitter user that have 9001 to 10000 followers.

```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | awk -F'\t'
'$7>10000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ cat corona
31473
```

Answer: There are 31473 twitter user that have more than 10000 followers.

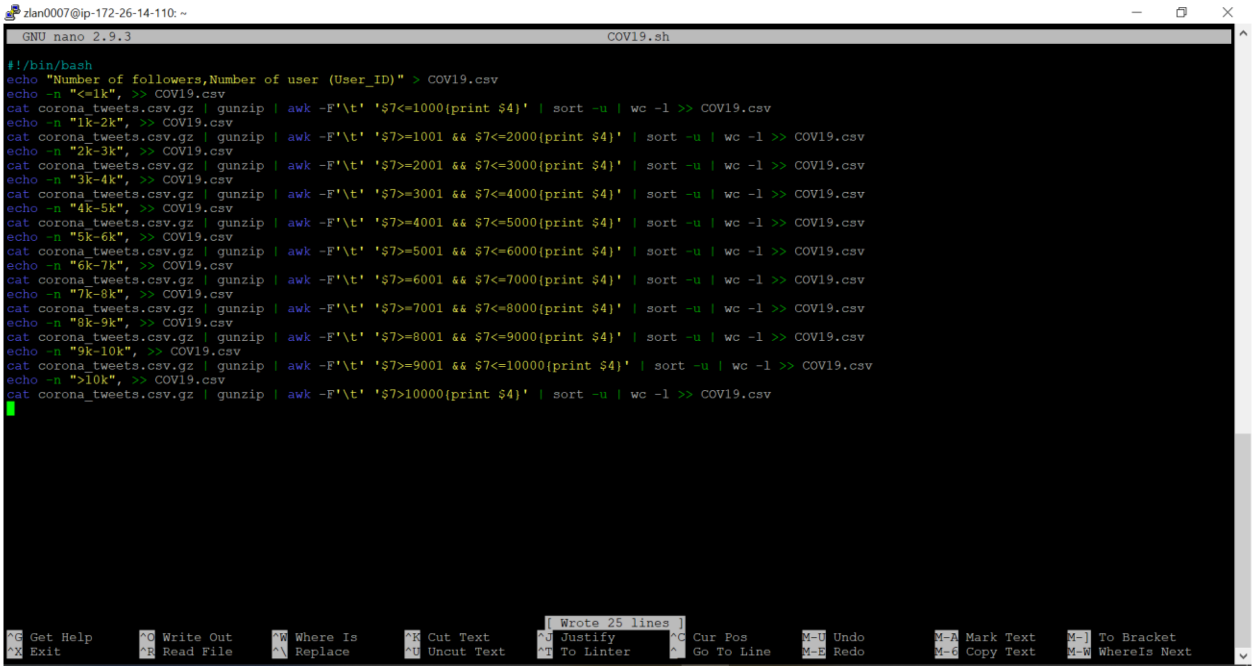
Explanation:

I used nano to write the following commands in a single shell script to output a csv file named COV19.csv. Inside the single shell script, I used echo to name my column as Number of followers and Number of user(User_ID), so that my CSV file contain two column with meaning names. I also used “echo –n” to indicate without a new line and save it as COV19.sh file then exit nano. Lastly, just execute the sh file using a single line of command which ic “bash COV19.sh”.

```
#!/bin/bash

echo "Number of followers,Number of user (User_ID)" > COV19.csv

echo -n "<=1k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7<=1000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "1k-2k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=1001 && $7<=2000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "2k-3k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=2001 && $7<=3000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "3k-4k," >> COV19.csv
```

```
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=3001 && $7<=4000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "4k-5k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=4001 && $7<=5000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "5k-6k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=5001 && $7<=6000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "6k-7k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=6001 && $7<=7000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "7k-8k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=7001 && $7<=8000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "8k-9k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=8001 && $7<=9000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n "9k-10k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=9001 && $7<=10000{print $4}' | sort -u | wc -l >> COV19.csv

echo -n ">10k," >> COV19.csv

cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>10000{print $4}' | sort -u | wc -l >> COV19.csv
```

Attached is the screenshot of the same code shown above.



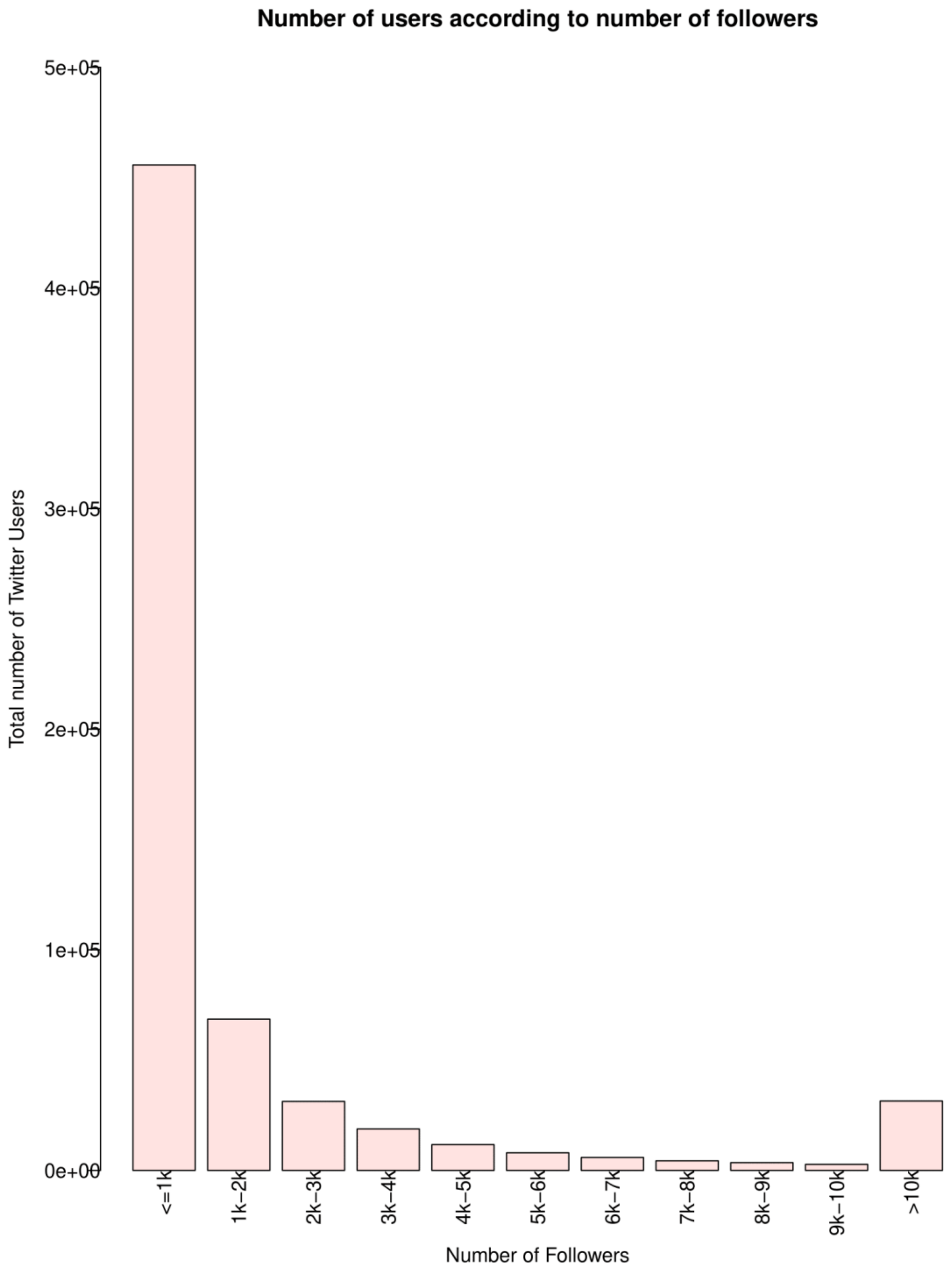
```
#!/bin/bash
echo "Number of followers,Number of user (User_ID)" > COV19.csv
echo -n "<=1k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7<=1000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "1k-2k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=1001 && $7<=2000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "2k-3k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=2001 && $7<=3000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "3k-4k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=3001 && $7<=4000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "4k-5k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=4001 && $7<=5000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "5k-6k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=5001 && $7<=6000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "6k-7k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=6001 && $7<=7000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "7k-8k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=7001 && $7<=8000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "8k-9k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=8001 && $7<=9000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n "9k-10k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>=9001 && $7<=10000{print $4}' | sort -u | wc -l >> COV19.csv
echo -n ">10k", >> COV19.csv
cat corona_tweets.csv.gz | gunzip | awk -F'\t' '$7>10000{print $4}' | sort -u | wc -l >> COV19.csv
```

[Wrote 25 lines]

Next, I read the COV19.csv file inside R and plotted a bar graph.

Provided below is the R code.

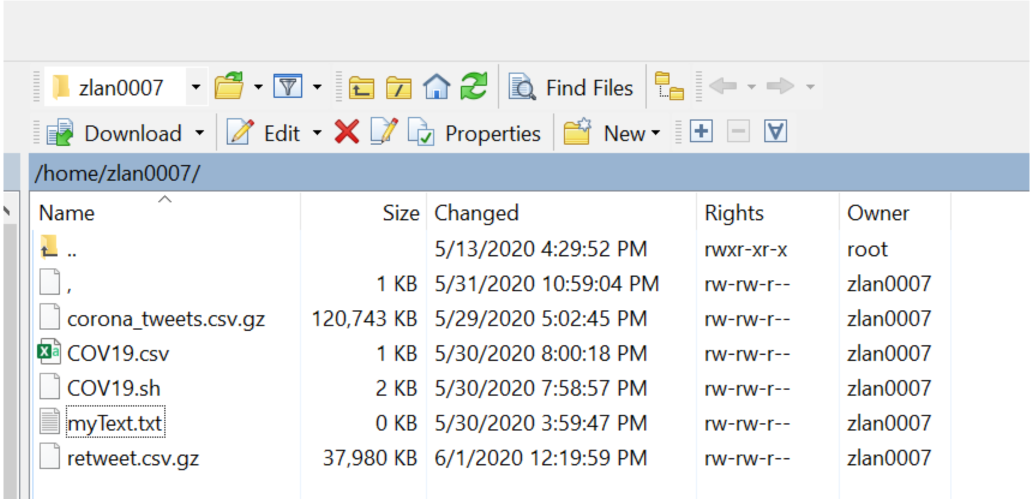
```
setwd("C:\Users\USER\Documents")

Total <- read.csv("COV19.csv")

barplot(Total$Number.of.user..User_ID.,names.arg = Total$Number.of.followers,las =2, mgp = c(3,0,0),ylim=c(0,500000),col =c('mistyrose'),main = 'Number of users according to number of followers',xlab='Number of Followers ',ylab='Total number of Twitter Users')
```


Number of users according to number of followers
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
Total number of Twitter Users
<=1k
1k−2k
2k−3k
3k−4k
4k−5k
5k−6k
6k−7k
7k−8k
8k−9k
9k−10k
>10k
Number of Followers

Part D : small challenge

```
zlan0007@ip-172-26-14-110:~$ cat corona_tweets.csv.gz | gunzip | grep -v "RT @" | sort -u | gzip >retweet.csv.gz
```

Explanation: I used “grep” and the flag “-v” so that it tells the grep to invert the match, that is only print out lines that don’t contain the string “RT @” , to zip it to gz, I used the opposite of gunzip which is gzip, followed by the character “>” so that it will be write into a zipped retweet.csv.gz file. I checked the directory to make sure the file is inside.

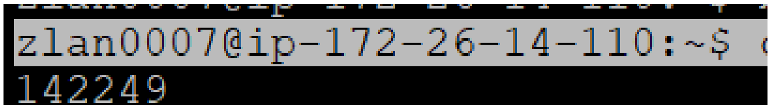


/home/zlan0007/

Name	Size	Changed	Rights	Owner
..		5/13/2020 4:29:52 PM	rwxr-xr-x	root
,	1 KB	5/31/2020 10:59:04 PM	rw-rw-r--	zlan0007
corona_tweets.csv.gz	120,743 KB	5/29/2020 5:02:45 PM	rw-rw-r--	zlan0007
COV19.csv	1 KB	5/30/2020 8:00:18 PM	rw-rw-r--	zlan0007
COV19.sh	2 KB	5/30/2020 7:58:57 PM	rw-rw-r--	zlan0007
myText.txt	0 KB	5/30/2020 3:59:47 PM	rw-rw-r--	zlan0007
retweet.csv.gz	37,980 KB	6/1/2020 12:19:59 PM	rw-rw-r--	zlan0007

Explanation: I used “awk” to only print column 4 which is the User_ID that have followers_count (column 7) with the amount less than or equal to 1000 followers, then I piped it to the “sort” command and “-u” flag to only print the unique User_ID, it is then piped into the “wc” command and “-l” flag to count the number of lines in the retweet.csv.gz file.

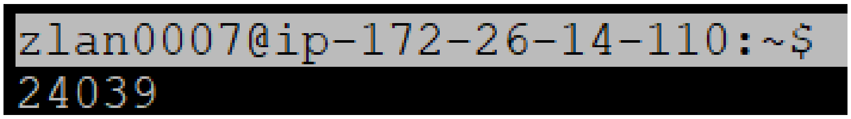
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t' '$7<=1000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ c
142249
```

Answer: There are 142249 non-retweet twitter user that have less than or equal to 1000 followers.

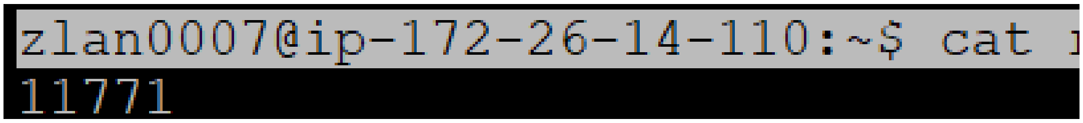
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=1001 && $7<=2000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$
24039
```

Answer: There are 24039 non-retweet twitter user that have 1001 to 2000 followers.

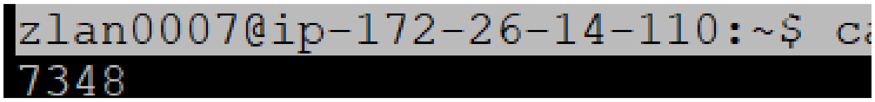
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=2001 && $7<=3000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ cat
11771
```

Answer: There are 11771 non-retweet twitter user that have 2001 to 3000 followers.

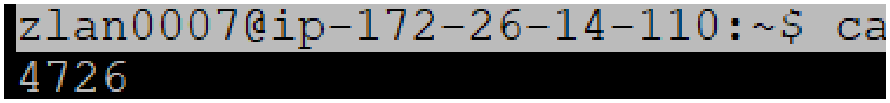
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=3001 && $7<=4000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ c
7348
```

Answer: There are 7348 non-retweet twitter user that have 3001 to 4000 followers.

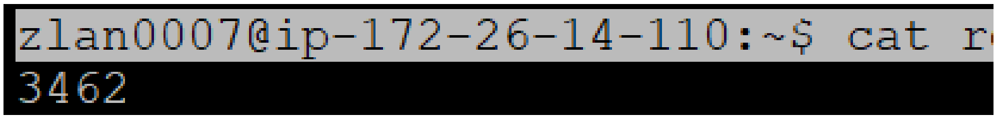
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=4001 && $7<=5000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ ca
4726
```

Answer: There are 4726 non-retweet twitter user that have 4001 to 5000 followers.

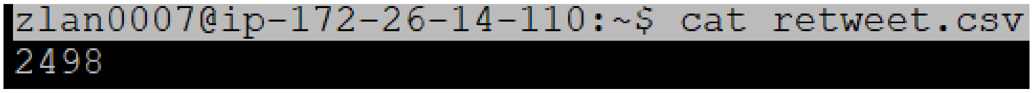
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=5001 && $7<=6000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ cat r
3462
```

Answer: There are 3462 non-retweet twitter user that have 5001 to 6000 followers.

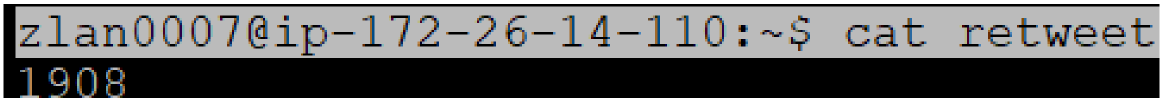
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=6001 && $7<=7000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv
2498
```

Answer: There are 2498 non-retweet twitter user that have 6001 to 7000 followers.

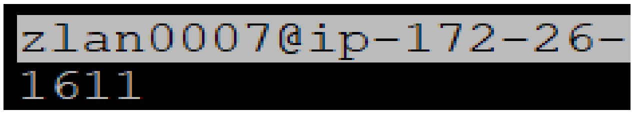
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=7001 && $7<=8000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:~$ cat retweet
1908
```

Answer: There are 1908 non-retweet twitter user that have 7001 to 8000 followers.

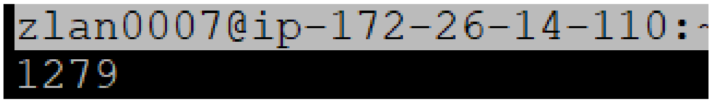
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=8001 && $7<=9000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-
1611
```

Answer: There are 1611 non-retweet twitter user that have 8001 to 9000 followers.

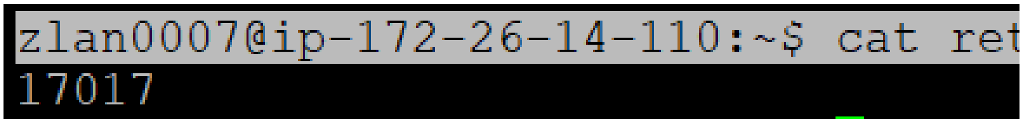
```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>=9001 && $7<=10000{print $4}' | sort -u | wc –l
```



```
zlan0007@ip-172-26-14-110:
1279
```

Answer: There are 1279 non-retweet twitter user that have 9001 to 10000 followers.

```
zlan0007@ip-172-26-14-110:~$ cat retweet.csv.gz | gunzip | awk -F'\t'
'$7>10000{print $4}' | sort -u | wc –l
```

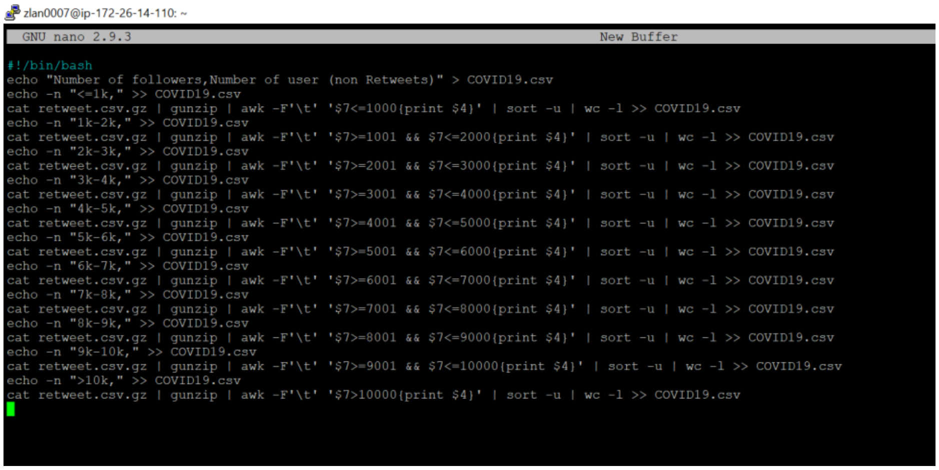


```
zlan0007@ip-172-26-14-110:~$ cat ret
17017
```

Answer: There are 17017 non-retweet twitter user that have more than 10000 followers.

Explanation:

Command using nano in a single shell script to output a CSV file named COVID19.csv and save it as COVID19.sh file, after exiting, just execute the sh file using bash COVID19.sh to output the CSV file.



```
#!/bin/bash
echo "Number of followers,Number of user (non Retweets)" > COVID19.csv
echo -n "<=1k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7<=1000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "1k-2k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=1001 && $7<=2000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "2k-3k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=2001 && $7<=3000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "3k-4k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=3001 && $7<=4000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "4k-5k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=4001 && $7<=5000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "5k-6k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=5001 && $7<=6000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "6k-7k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=6001 && $7<=7000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "7k-8k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=7001 && $7<=8000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "8k-9k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=8001 && $7<=9000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n "9k-10k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=9001 && $7<=10000{print $4}' | sort -u | wc -l >> COVID19.csv
echo -n ">10k," >> COVID19.csv
cat retweet.csv.gz | gunzip | awk -F'\t' '$7>10000{print $4}' | sort -u | wc -l >> COVID19.csv
```

```
#!/bin/bash

echo "Number of followers,Number of user (non Retweets)" > COVID19.csv

echo -n "<=1k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7<=1000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "1k-2k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=1001 && $7<=2000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "2k-3k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=2001 && $7<=3000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "3k-4k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=3001 && $7<=4000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "4k-5k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=4001 && $7<=5000{print $4}' | sort -u | wc -l >> COVID19.csv
```

```
echo -n "5k-6k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=5001 && $7<=6000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "6k-7k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=6001 && $7<=7000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "7k-8k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=7001 && $7<=8000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "8k-9k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=8001 && $7<=9000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n "9k-10k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>=9001 && $7<=10000{print $4}' | sort -u | wc -l >> COVID19.csv

echo -n ">10k," >> COVID19.csv

cat retweet.csv.gz | gunzip | awk -F'\t' '$7>10000{print $4}' | sort -u | wc -l >> COVID19.csv

zlan0007@ip-172-26-14-110:~$ bash COVID19.sh
```

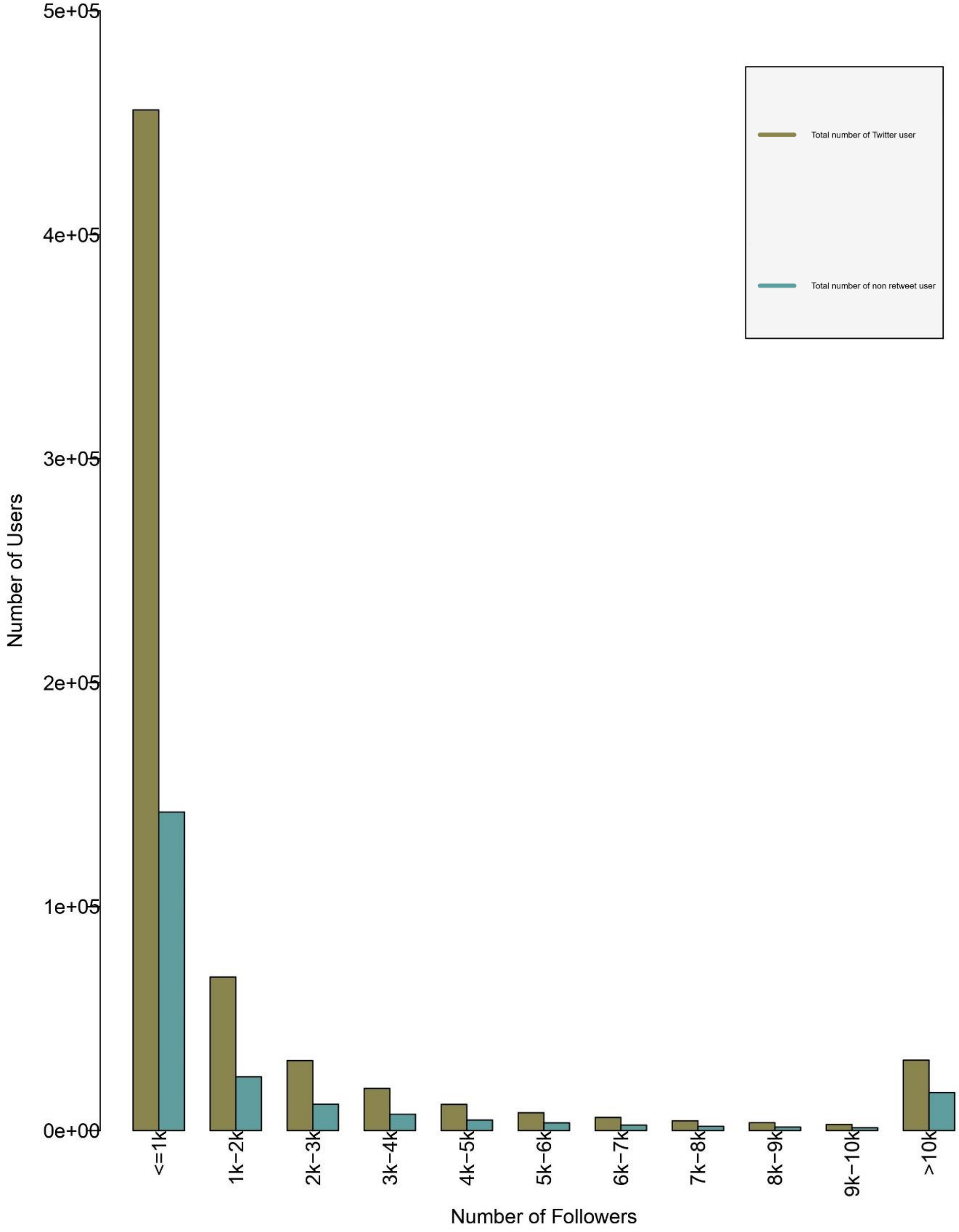
Reading the CSV file and plotting the side by side bar graph (R code) :

My legend is blocking the graph, hence I decided to use legend.outside and replot my legend.

```
setwd("C:\Users\USER\Documents")

Non_retweets <- read.csv("COVID19.csv")


result <- rbind(Total$Number.of.user..User_ID.,Non_retweets$Number.of.user..non.Retweets.)

barplot(result,beside=TRUE, names.arg = Total$Number.of.followers,las = 2, mgp = c(3,0,0),ylim=c(0,500000),col =c('khaki4','cadetblue'), legend.outside = c(side=1,'Total','Non_retweets'), main = 'Side by side bar graph on total number of user and non retweets user',xlab='Number of Followers',ylab='Number of Users')

legend("topright", inset=.05, cex = 0.25, c("Total number of Twitter user","Total number of non retweet user"), horiz=FALSE, lty=c(1,1), lwd=c(2,2), col=c('khaki4','cadetblue'), bg="grey96")
```

Explanation:

From the graph below, most of the user have followers less than or equal to 1k, the least number of user have many followers from 9k to 10k, the majority of users wouldn’t have such a high number of followers because maybe they are not a social influencer, but they are very active in tweeting since from the tweets extracted we have 142299 non-retweets user and total 455758 users, meaning that the majority of users are from the retweet users.



Side by side bar graph on total number of user and non retweets user

Conclusion

COVID-19 is a coronavirus disease that recently emerged in the late 2019 and it is now a hot topic on twitter. This means that it has a global impact that affects all kinds of people and different sectors such as the private, public, and social sectors. Hence, the number of tweets for this data set is extremely high, and it will continue to grow, assuming that a vaccine or cure has not been found. In this dataset, the word "death" has been mentioned 50911 times. I believe that this is because this virus has led many people to meet their end including doctors and nurses from all over the world who have been risking their lives to treat the patients. Majority of the twitter users are from the younger generation, this means that the young generation is highly aware of how deadly this pandemic is. By raising awareness over social media, hopefully, everyone can play their part to maintain social distancing and take care of themselves.