FIT1043 DATA SCIENCE ASSIGNMENT 3

Name: NASHMIA SHAKEEL

STUDENT ID:34091904

A1):

Q1:

Code:
```
cp /cygdrive/c/Users/AB/Downloads/corona_tweets.csv.gz .
```

Explanation: I have copied the file "corona_tweets.csv.gz" into my unix terminal by using cp command which is used to copy files into Unix terminal

Code:
```
ls -lh corona_tweets.csv.gz
-rwx------ 1 AB None 118M May 24 10:30 corona_tweets.csv.gz
```

Explanation: The size of the file is 118 megabytes
The commands used here are ls and lh. ls command is used to list files in a directory and lh is used with ls in order to show file size in a readable format like megabytes instead of bytes.

Q2:

Code:
```
gunzip -c corona_tweets.csv.gz | head -1

Created Tweet_ID        Text    User_ID User    User_Location
Followers_CountFriends_Count    Geo     Place_Type      Place_Name
Place_Country   Language
```

Explanation: There are a total of 13 columns in this file

1. Created
2. Tweet_ID
3. Text
4. User_TD
5. User
6. User_Location
7. Followers_Count
8. Friends_Count

9. Geo
10. Place_Type
11. Place_Name
12. Place_Country
13. Language

The commands used here are gunzip -c, head -1 and | (pipe)

gunzip command is used to decompress a file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

head command is used to show first few lines of a file. While setting head with an option of -1 we restricted it to only show the first line of the file.

Q3):

Code:

```
gunzip -c corona_tweets.csv.gz | wc -l
1143559
```

Explanation: There are 1143559 lines in the dataset

The commands used are gunzip -c , | (pipe) and wc-l

gunzip command is used to decompress a file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

wc command is used to count words, lines and characters of a file. However, by setting – l we have restricted wc to only count lines of the file.

A2):

Q1):

Code:

```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '{print $4}' | sort | uniq
| wc -l
641976
```

Explanation: There are 641975 twitter users in the file since the first line is the header of the file where the names of the columns are present hence, we will exclude that line from our answer hence leaving us with answer 641975.

The commands used in this code are gunzip -c, awk -F'\t' '{print $4}', sort, uniq,
wc -l

gunzip command is used to decompress corona_tweets.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

wc command is used to count words, lines and characters of a file. However, by setting – l we have restricted wc to only count lines of the file.

awk -F'\t' '{print $4}' here command awk is used to extract data from the fourth column that is User ID and these columns are separated by tabs (\t)

sort command sorts the fourth column that is extracted alphabetically.

Uniq command removes duplicate adjacent lines from the sorted column.

The reason why uniq is not sufficient to answer this question is because uniq only removes the duplicate lines that are adjacent to each other and not the ones that are not adjacent hence this way we first use sort to sort the column so that duplicate lines can become adjacent to each other and then we apply uniq on them to remove the duplicate lines. This way we get the right number of unique twitter users.

Q2a):

Code:

```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '{print $3}' | grep -iwc "vaccine"
16569
```

Explanation: The number of tweets that mentioned “vaccine” in any combination of uppercase or lowercase letters are 16569

The commands used here are gunzip -c, |(pipe), awk -F'\t' '{print $3}', grep -iwc

gunzip command is used to decompress corona_tweets.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

awk -F'\t' '{print $3}' here command awk is used to extract data from the third column that is Text and these columns are separated by tabs (\t)

grep -iwc this command is used here to search for the exact word "vaccine" in the third column regardless if it lowercases or uppercase this is done through the -i option and number of tweets that mention the word "vaccine" are counted through the option -c option. The -w option makes sure that grep only looks for the whole word "vaccine" and does not match with parts of the word like "vaccines". Through this we make sure that only the word "vaccine" is counted.

Q2b)

Code:

```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '{print $3}' | grep -iw 'vaccine' | grep -v 'vaccine' | grep -v 'Vaccine' | wc -l
270
```

Explanation: The number of tweets that don't mention "vaccine" or "Vaccine" but do mention them in other combination of lowercase and uppercase are 270

The commands being used here are gunzip -c, awk -F'\t' '{print $3}', grep -iw,
grep -v, wc -l and | (pipe)

gunzip command is used to decompress corona_tweets.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

awk -F'\t' '{print $3}' here command awk is used to extract data from the third column that is Text and these columns are separated by tabs (\t)

grep -iw 'vaccine' searches for 'vaccine' regardless of the fact that whether it is uppercase or lowercase as we set the option -i. It also only searches for whole vaccine words and does not match with parts of it like it does not count vaccinated.

grep -v 'vaccine' excludes the lines that mention vaccine specifically as -v inverts the match so that grep only looks for lines from the output of grep -iw that do not contain 'vaccine'

grep -v 'Vaccine' excludes the lines that mention Vaccine specifically as -v inverts the match so that grep only looks for lines from the output of grep -iw that do not contain 'Vaccine'

wc command is used to count words, lines and characters of a file. However, by setting – l we have restricted wc to only count lines of the file.

Q2c)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '{print $3}' | grep -iw 'vaccine' | grep -v 'vaccine' | grep -v 'Vaccine' > Result.txt
```

Explanation:

The commands being used here are gunzip -c, awk -F'\t' '{print $3}', grep -iw,
grep -v and | (pipe)

gunzip command is used to decompress corona_tweets.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

awk -F'\t' '{print $3}' here command awk is used to extract data from the third column that is Text and these columns are separated by tabs (\t)

grep -iw 'vaccine' searches for 'vaccine' regardless of the fact that whether it is uppercase or lowercase as we set the option -i. It also only searches for whole vaccine words and does not match with parts of it like it does not count vaccinated.

grep -v 'vaccine' excludes the lines that mention vaccine specifically as -v inverts the match so that grep only looks for lines from the output of grep -iw that do not contain 'vaccine'

grep -v 'Vaccine' excludes the lines that mention Vaccine specifically as -v inverts the match so that grep only looks for lines from the output of grep -iw that do not contain 'Vaccine'

> Result.txt by doing this I am saving the data in file Result.txt

A3)

Q1a)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 <= 1500' | cut -f4 | sort | uniq | wc -l
498480
```

The number of twitter users that have followers less than equal to 1500 are 498480

Q1b)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 1501 && $7 <= 2500'
| cut -f4 | sort | uniq | wc -l
43891
```

The number of twitter users that have followers between and including 1501 and 2500 are 43891

Q1c)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 2501 && $7 <= 3500'
| cut -f4 | sort | uniq | wc -l
23620
```

The number of twitter users that have followers between and including 2501 and 3500 are 23620

Q1d)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 3501 && $7 <= 4500'
| cut -f4 | sort | uniq | wc -l
15165
```

The number of twitter users that have followers between and including 3501 and 4500 are 15165

Q1e)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 4501 && $7 <= 5500'
| cut -f4 | sort | uniq | wc -l
9297
```

The number of twitter users that have followers between and including 4501 and 5500 are 9297

Q1f)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 5501 && $7 <= 6500'
| cut -f4 | sort | uniq | wc -l
6848
```

The number of twitter users that have followers between and including 5501 and 6500 are 6848

Q1g)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 6501 && $7 <= 7500'
| cut -f4 | sort | uniq | wc -l
5076
```

The number of twitter users that have followers between and including 6501 and 7500 are 5076

Q1h)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 7501 && $7 <= 8500'
| cut -f4 | sort | uniq | wc -l
3855
```

The number of twitter users that have followers between and including 7501 and 8500 are 3855

Q1i)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 >= 8501 && $7 <= 9500'
| cut -f4 | sort | uniq | wc -l
3072
```

The number of twitter users that have followers between and including 8501 and 9500 are 3072

Q1j)

Code:
```
gunzip -c corona_tweets.csv.gz | awk -F'\t' '$7 > 9500' | cut -f4 |
sort | uniq | wc -l
32772
```

The number of twitter users that have followers more than 9500 are 32772

Explanation for this whole question:

Commands used are gunzip-c, awk -F'\t', cut-f4, sort, uniq , wc- l, | (pipe)

gunzip command is used to decompress corona_tweets.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

awk -F'\t' '$7 >= given range && $7 <= given range' here command awk is used to extract data from the third column that is followers count and these columns are separated by tabs (\t). It also only includes lines from the 7th column that match the given range and filters out the rest

cut -f4 command is used here to only extract specific field from the data and in this case since we need User ID that is present in 4th column we have specified -f4 so that cut only select data from the fourth column

sort command sorts the fourth column that is extracted alphabetically so that duplicate user Ids become adjacent.

Uniq command removes duplicate adjacent user Ids from the sorted column.

wc command is used to count words, lines and characters of a file. However, by setting – l we have restricted wc to only count lines of the file.

Q2)

Down below I have created a csv file named "datascience6 (1)". It includes the result from the previous question and has two columns one is Range and the other is number of twitter users

Range	number of twitter users
<= 1500	498480
1501 -2500	43891
2501 - 3500	23620
3501 - 4500	15165
4501 - 5500	9297
5501 - 6500	6848
6501 - 7500	5076
7501 - 8500	3855
8501 - 9500	3072
>9500	32772

Q3)

For the R code I have made use of Rstudio where in order to set the working directory I first created a folder over there named datascience then I went to session then selected set working directory and then chose datascience folder and set my working directory.

Code:

```
getwd()
data=read.table("datascience6.csv",header=TRUE,sep=",")
```

Explanation: I used getwd() to confirm that my working directory has been rightly set
Then I have made use of read.table () in order to read the data from the csv file and create a data frame which is then stored in the variable name data. I have set header to TRUE so that the read.table() can know that the first line of the data are columns. I have set sep=”,” in order to indicate that the csv file has comma separated values.

Q4)

Code:

```
getwd()
data=read.table("datascience6 (1).csv",header=TRUE,sep=",")
par(mar = c(6, 6, 4, 2) + 0.1)
barplot(data$number.of.twitter.users, names.arg = data$Range, xlab =
"Range", ylab = "number of twitter users", col = "yellow", main =
"Number of Twitter Users according to their number of
followers",cex.names = 0.8)
```

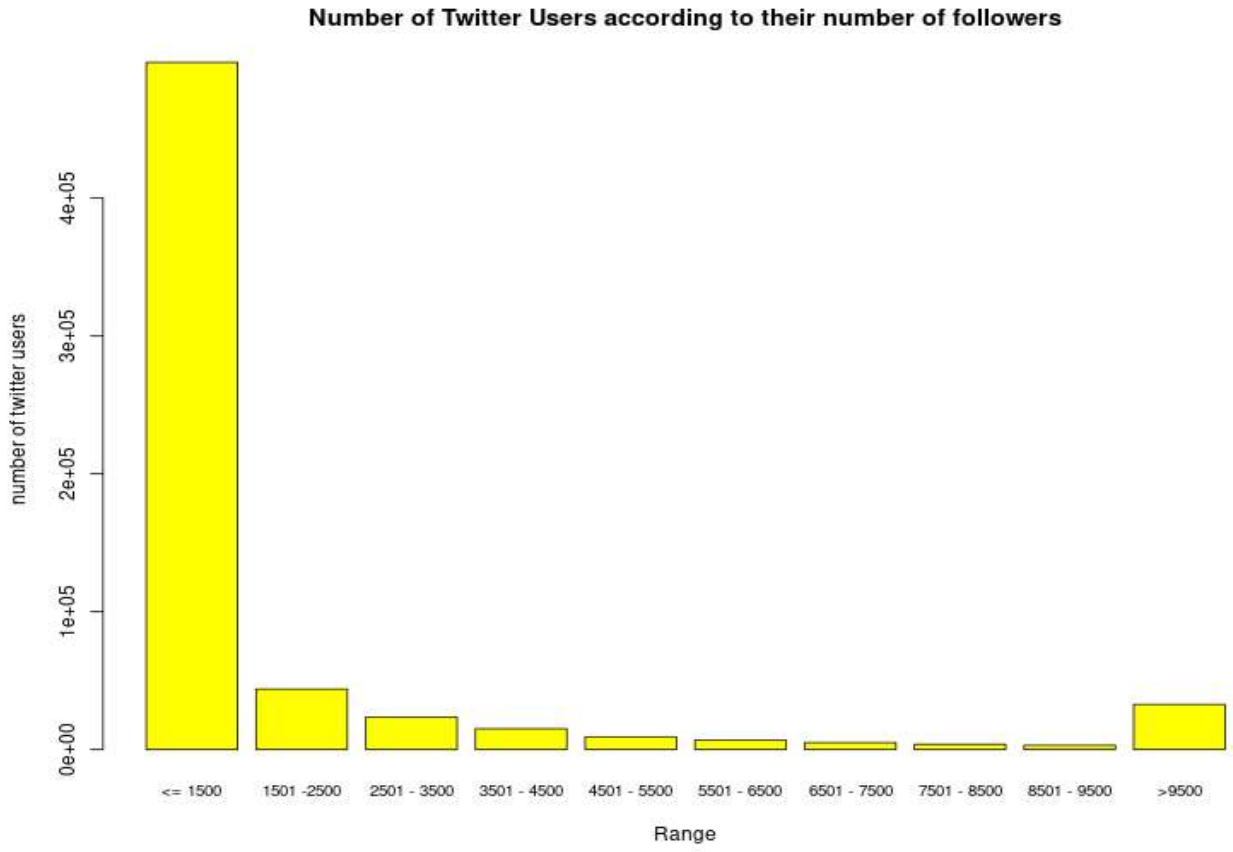
The above code creates a bar chart

```
png("Twitter_users_according_to_their_followers_barchart1.png", width
= 800, height = 600)
par(mar = c(6, 6, 4, 2) + 0.1)
barplot(data$number.of.twitter.users, names.arg = data$Range, xlab =
"Range", ylab = "number of twitter users", col = "yellow", main =
"Number of Twitter Users according to their number of
followers",cex.names = 0.8)
dev.off()
```

Explanation:
png() is used to open a graphic device in order to save the plot as a png image
par() is used to set the parameters
mar is used to set the margins
barplot() is used to create a bar chart
dev.off() is used to close the graphical device opened by png ()

This code will result in a png image of the bar chart that has been created through this code.

The image down below is the bar chart created by the above code:



A4)

Q1)

Code:

```
gunzip -c corona_tweets.csv.gz | awk -F '\t' '$3 !~ /^RT @/' | gzip >
No_retweets_file2.csv.gz
```

Explanation: The commands used here are gunzip -c, awk -F '\t' '$3 !~ /^RT @/', gzip, and |(pipe)

gunzip command is used to decompress corona_tweets.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

awk -F '\t' '$3 !~ /^RT @/' the awk command here is used to extract data from the third column which is Text and then $3 !~ /^RT @/' is used to exclude lines that start with 'RT @' hence this way we get lines that do not include any retweets.

gzip is used to compress the data then this compressed data is copied into the file No_retweets_file2.csv.gz

Q2)

Q2a)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 <= 1500' | cut -f4 | sort | uniq | wc -l
157068
```

The number of twitter users that have followers less than equal to 1500 are 157068

Q2b)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 1501 && $7 <= 2500' | cut -f4 | sort | uniq | wc -l
16073
```

The number of twitter users that have followers between and including 1501 and 2500 are 16073

Q2c)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 2501 && $7 <= 3500' | cut -f4 | sort | uniq | wc -l
9016
```

The number of twitter users that have followers between and including 2501 and 3500 are 9016

Q2d)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 3501 && $7 <= 4500' | cut -f4 | sort | uniq | wc -l
```

6071

The number of twitter users that have followers between and including 3501 and 4500 are 6071

Q2e)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 4501 && $7 <= 5500' | cut -f4 | sort | uniq | wc -l
```
3872

The number of twitter users that have followers between and including 4501 and 5500 are 3872

Q2f)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 5501 && $7 <= 6500' | cut -f4 | sort | uniq | wc -l
```
2967

The number of twitter users that have followers between and including 5501 and 6500 are 2967

Q2g)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 6501 && $7 <= 7500' | cut -f4 | sort | uniq | wc -l
```
2187

The number of twitter users that have followers between and including 6501 and 7500 are 2187

Q2h)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 7501 && $7 <= 8500' | cut -f4 | sort | uniq | wc -l
```
1726

The number of twitter users that have followers between and including 7501 and 8500 are 1726

Q2i)

Code:
```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 >= 8501 && $7 <= 9500' | cut -f4 | sort | uniq | wc -l
```
1428

The number of twitter users that have followers between and including 8501 and 9500 are 1428

Q2j)

Code:

```
gunzip -c No_retweets_file2.csv.gz | awk -F'\t' '$7 > 9500' | cut -f4 | sort | uniq | wc -l
17645
```

The number of twitter users that have followers more than 9500 are 17645

Explanation for this whole question:

Commands used are gunzip-c, awk -F'\t', cut-f4, sort, uniq , wc- l, | (pipe)

gunzip command is used to decompress No_retweets_file2.csv.gz file and -c enables gunzip to write the decompressed output into standard output (stdout) rather than writing it into a file. Hence gunzip -c helps us to show data of a compressed file.

| (pipe) takes output from the command present on the left side and passes that as an input for the command on the right side.

awk -F'\t' '$7 >= given range && $7 <= given range' here command awk is used to extract data from the third column that is followers count and these columns are separated by tabs (\t). It also only includes lines from the 7th column that match the given range and filters out the rest

cut -f4 command is used here to only extract specific field from the data and in this case since we need User ID that is present in 4th column we have specified -f4 so that cut only selects data from the fourth column

sort command sorts the fourth column that is extracted alphabetically so that duplicate user Ids become adjacent.

Uniq command removes duplicate adjacent user Ids from the sorted column.

wc command is used to count words, lines and characters of a file. However, by setting – l we have restricted wc to only count lines of the file.

Down below I have included the screenshot of my csv file named “datascience8” which has 2 columns Range and number of twitter users

Range	number of twitter users
<= 1500	157068
1501 -2500	16073
2501 - 3500	9016
3501 - 4500	6071
4501 - 5500	3872
5501 - 6500	2967
6501 - 7500	2187
7501 - 8500	1726
8501 - 9500	1428
>9500	17645

Q3):

For the R code I have made use of Rstudio where in order to set the working directory I first created a folder over there named datascience then I went to session then selected set working directory and then chose datascience folder and set my working directory.

Code:

```
getwd()

data1=read.table("datascience8.csv",header=TRUE,sep=",")
```

Explanation: I used getwd() to confirm that my working directory has been rightly set
Then I have made use of read.table () in order to read the data from the csv file and create a data frame which is then stored in the variable name data1. I have set header to TRUE so that the read.table() can know that the first line of the data are columns. I have set sep=”,” in order to indicate that the csv file has somma separated values.

Q4)

Code:

```
com_data=merge(data1,data,by="Range",all=TRUE,sort=FALSE)
barplot(height=rbind(data1$number.of.twitter.users, data$number.of.twitter.users),
      beside=TRUE,
      names.arg=com_data$Range,
      legend.text=c("no_retweet","retweet"),
      main="Number of twitter users according to the number of followers",
      xlab="Range",
      ylab="Number of twitter users",
      cex.names=0.8,
      col=c("yellow","green"))
```

The above code creates a bar chart

Code:

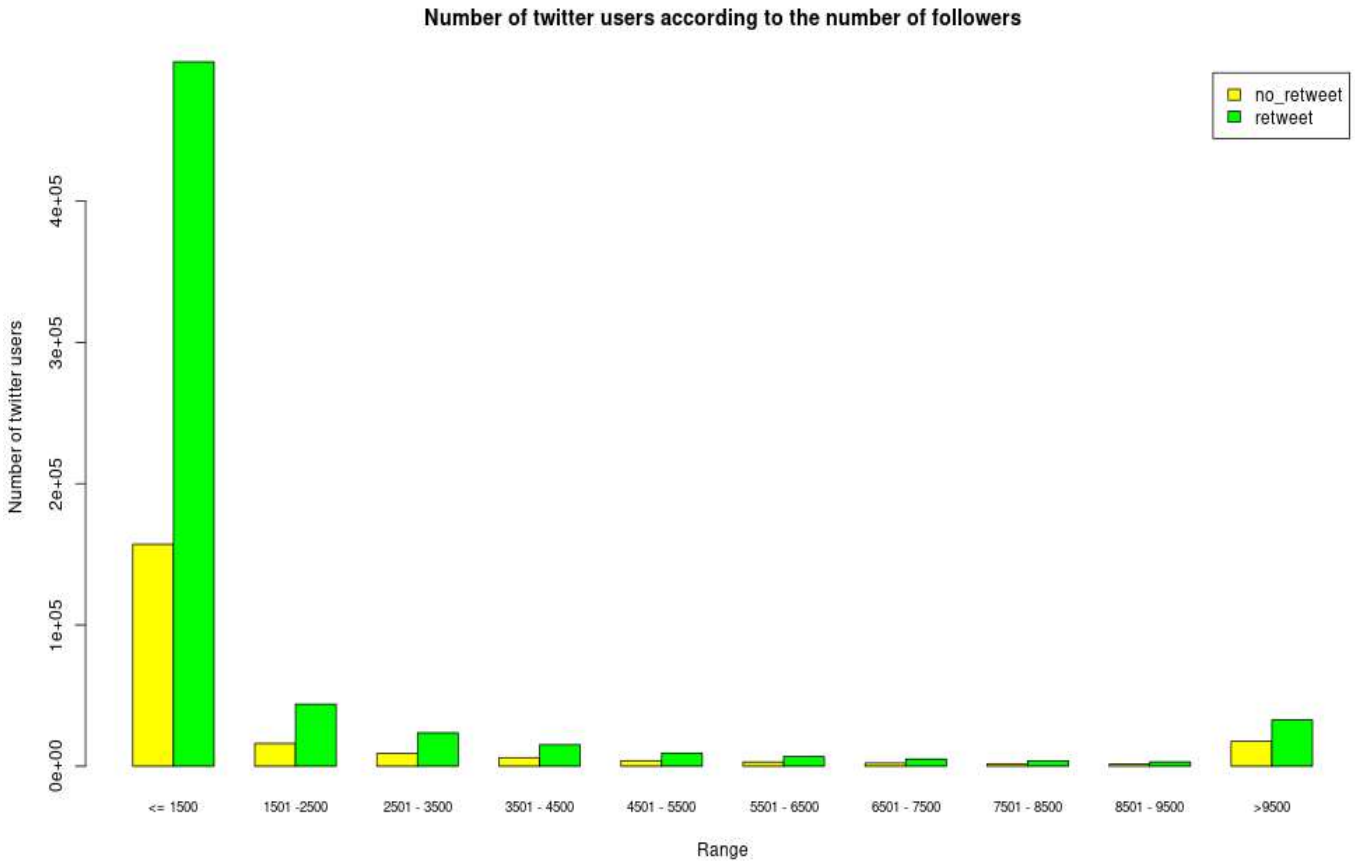
```
png("Retweet_comparsion_Noretweet2.png", width = 1000, height = 600)
com_data=merge(data1,data,by="Range",all=TRUE,sort=FALSE)
barplot(height=rbind(data1$number.of.twitter.users, data$number.of.twitter.users),
      beside=TRUE,
      names.arg=com_data$Range,
      legend.text=c("no_retweet","retweet"),
      main="Number of twitter users according to the number of followers",
      xlab="Range",
      ylab="Number of twitter users",
```

```
    cex.names=0.8,
    col=c("yellow","green"))
dev.off()
```

```
Explanation:
png() is used to open a graphic device in order to save the plot as a png
image
par() is used to set the parameters
merge() is used to merge the two dataframes together by a common value in
this case Range this helps us create a side by side bar chart
mar is used to set the margins
barplot() is used to create a bar chart
dev.off() is used to close the graphical device opened by png ()
```

This code will result in a png image of the bar chart that has been created through this code.

The image down below is the bar chart created by the above code:



Q5)

Most Twitter users have 1500 or less followers and in this range more users have retweeted (green bar) as compared to those who have not retweeted (yellow bar)

As the follower count increases the number of users that have those number of followers decreases

In all ranges users who have retweeted (green bar) are more than those who haven’t (yellow bar)

Hence, Twitter users that have less followers on twitter show more engagement on the platform through their retweets. This bar chart shows that users who retweet are more active on twitter hence the number of users that retweet (green bar) are more than the number of users who don’t (yellow bar).