Name: Bryan Cheng Hengze

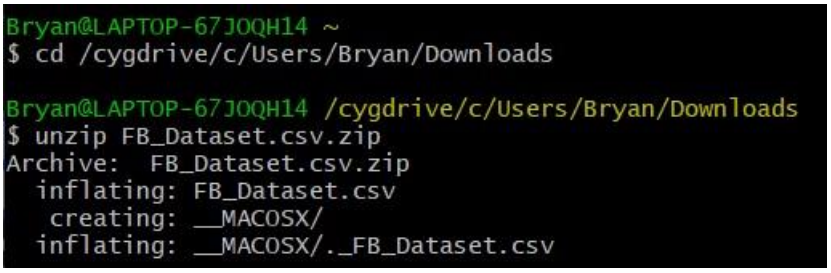
Task A: Investigating Facebook Data using shell commands

1)

$ cd /cygdrive/c/Users/Bryan/Downloads

$ unzip FB_Dataset.csv.zip

Output:



Firstly, I change my directory to my Downloads folder so that I can access the FB_Dataset.csv.zip that I just downloaded.

Then, I decompressed the file and extract the files from FB_Dataset.csv.zip

$ ls -lh FB_Dataset.csv

Output:

ls –lh provides many important file attributes such as permissions, user who owns the file, group which the file belong to, the size of the file in bytes and the date of the file that was last changed. The h in ls –lh translates the size to a more human-friendly notation.

Therefore, we can see that the file is 344MB big.

2)

$ head -1 FB_Dataset.csv Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ head -1 FB_Dataset.csv
page_name,post_id,page_id,post_name,message,description,caption,post_type,status_t
ype,likes_count,comments_count,shares_count,love_count,wow_count,haha_count,sad_co
unt,thankful_count,angry_count,post_link,picture,posted_at
```

This code outputs the first line of the file.

As we can see from the first line of the CSV file which is the header, the delimiter used to separate the columns in the file are commas.

$ awk –F ‘,’ ‘{print NF; exit}’ FB_Dataset.csv Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ awk -F ',' '{print NF; exit}' FB_Dataset.csv
21
```

awk processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed it the instruction {print NF; exit}, which tells it to print the number of fields in the current record (NF) and exit so that it does not read the whole file.

Therefore, we can see that there are 21 columns.

3)

$ awk –F ‘,’ ‘{$2 = “”; print $0}’ FB_Dataset.csv | head -1 Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ awk -F ',' '{$2 ="";print $0}' FB_Dataset.csv | head -1
page_name  page_id post_name message description caption post_type status_type l
ikes_count comments_count shares_count love_count wow_count haha_count sad_count
 thankful_count angry_count post_link picture posted_at
```

awk processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed it the instruction {$2 = “” ; print $0}, which tells it to print out the name other than the 2nd column as I set $2 = “” and printed the rest of the column using print $0. Then I pipe the results to head -1 which is the first line that is the header.

Therefore, we can see that the name of the columns other than the 2nd column is page_name, page_id, post_name, message, description, caption, post_type, status_type, likes_count, comments_count, shares_count, love_count, wow_count, haha_count, sad_count, thankful_count, angry_count, post_link, picture and posted_at.

4)

$ awk –F ‘,’ ‘{if(NR!=1){print$3}}’ FB_Dataset.csv | sort | uniq | wc -l Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ awk -F ',' '{if(NR!=1){print$3}}' FB_Dataset.csv | sort | uniq | wc -l
15
```

awk processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed it the instruction {if(NR != 1){print$3}}, which tells it to print out the column 3 IF the number of records(NR) is not 1. This is because if the NR == 1, it will

be taking the header “page_id” as one of the pages as well. I then pipe the results to sort the input because I am going to pipe the results into uniq filters adjacent matching lines from input writing to output. The output now show the unique pages there is. Lastly, I pipe all of those into wc –l where wc does a word count and the –l flag reports the line, to get the number of unique pages.

Therefore, we can see that there are 15 unique pages.

5)

$ cat FB_Dataset.csv | awk –F ‘,’ ‘{if(NR==2){print $21}}’

$ cat FB_Dataset.csv | tail –n 1 | awk –F ‘,’ ‘{print $21}’ Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ cat FB_Dataset.csv | awk -F ',' '{if(NR == 2){print $21}}'
1/1/12 0:30
```

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ cat FB_Dataset.csv | tail -n 1 | awk -F ',' '{print $21}'
7/11/16 23:45
```

cat loads the FB_Dataset.csv and output it to the terminal. Then I pipe the results to awk which processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed the instruction {if(NR == 2){print$21}}, which tells it to print out the column 21(posted_at) IF the number of records(NR) is 2. This is to get the date of first post in this file.

After that, to find the last post in this file, I use cat to load the file again and then pipe the result to tail –n 1 which takes the final line of the row. Then I pipe it again to awk which processes a file one line at a time and use flag –F ‘,’ to tell awk that the columns are separated by commas. I then pass the instruction to print out only the column 21. This will then get the date of last post in this file.

Therefore, we can see that the date range for Facebook post in this file is 1/1/12 - 7/11/16 6)

$ awk –F ‘,’ ‘$0 ~ /Italian Dishes/{print $4;exit}’ FB_Dataset.csv Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ awk -F ',' '$0 ~ /Italian Dishes/{print $4;exit}' FB_Dataset.csv
5 Brilliant Italian Dishes You Haven't Tried Before
```

awk processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. Then $0 ~ /Italian Dishes/ gets the line which has the word “Italian Dishes” and {print $4;exit} prints the post name of that line and exits as they only want the first mention in the file regarding “Italian Dishes”.

7)

$ grep –o “Donald Trump” FB_Dataset.csv | wc –l Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ grep -o "Donald Trump" FB_Dataset.csv | wc -l
15024
```

We can see that the number of times “Donald Trump” is mention in the file is 15024. I found this by using grep –o to search only the part of a line matching “Donald Trump” in FB_Dataset.csv. Then, I pipe the results to wc –l which prints the newline counts.

8)

$ grep –o “Barrack Obama” FB_Dataset.csv | wc –l Output:

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ grep -o "Barack Obama" FB_Dataset.csv | wc -l
6831
```

We can see that the number of times “Barack Obama” is mention in the file is 6831. I found this by using grep –o to search only the part of a line matching “Barack Obama” in FB_Dataset.csv. Then, I pipe the results to wc –l which prints the newline counts.

Therefore, we can see that Donald Trump is more popular than Barack Obama on Facebook because more people mention Donald Trump compared to Barack Obama.

9)

$ grep –i “Trump” FB_Dataset.csv | awk –F ‘,’ ‘$10 > 100 {print $2,$10}’ > trump_unsorted .txt

$ sort –nk2 trump_unsorted.txt > trump.txt

$ echo –e “post_id likes_count\n$(cat trump.txt)” > trump.txt $

head -6 trump.txt

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ head -6 trump.txt
post_id likes_count
131459315949_10153423359555950 101
131459315949_10153583026165950 101
131459315949_10153707463735950 101
131459315949_10153961477340950 101
10606591490_10153445206101491 101
```

$ grep –i which searches for any post where “Trump” is mention ignoring case distinctions from the file FB_Dataset.csv. Then I pipe the results awk which processes a file one line at a

time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed a condition where if column 10(like_count) is more than 100 then print the column 2(post_id) and column 10(like_count) and redirect it to save the data to a file called trump_unsorted.txt.

After doing that, I sorted the data in trump_unsorted.txt by the like_count using $sort –nk2 and redirected it to save the data to a file called trump.txt.

Then, I used the echo command to display the line of text using echo –e where –e enables the interpretation of backslash escapes. I used cat trump.txt so that I could concatenate the string “post_id likes_count to the start of the file of trump.txt. I then redirect it to save the data to the file trump.txt again.

Therefore, by using head -6, we can see the first 5 rows and the column headers in the trump.txt file.

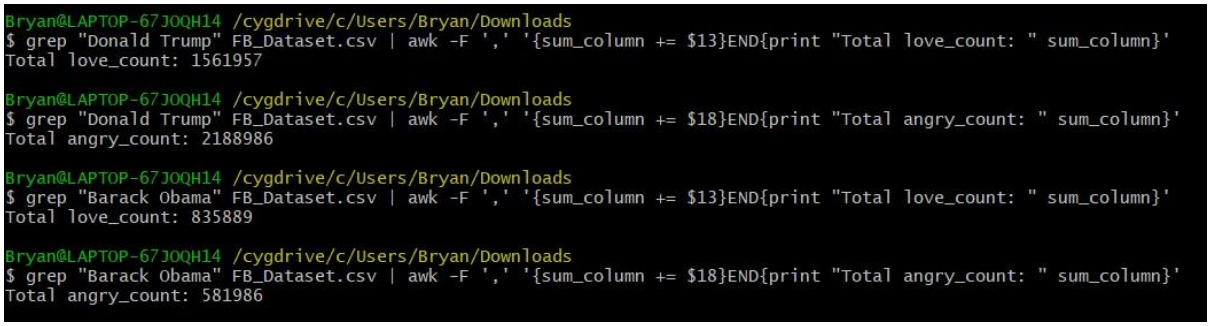
10)

$ grep “Donald Trump” FB_Dataset.csv | awk –F ‘,’ ‘{sum_column += $13}END{print “Total love_count: “ sum_column}’

$ grep “Donald Trump” FB_Dataset.csv | awk –F ‘,’ ‘{sum_column += $18}END{print “Total angry_count: “ sum_column}’

$ grep “Barack Obama” FB_Dataset.csv | awk –F ‘,’ ‘{sum_column += $13}END{print “Total love_count: “ sum_column}’

$ grep “Barack Obama” FB_Dataset.csv | awk –F ‘,’ ‘{sum_column += $18}END{print “Total angry_count: “ sum_column}’



I grep “Donald Trump” FB_Dataset.csv to search for line that contains the word “Donald Trump”. Then I pipe the results to awk which processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed the instruction to get the sum of the column 13(love_count) and then print the total love count.

Then, I did the same line of code again but instead of getting the sum of column 13(love_count), I took the sum of column 18(angry_count) and then print the total angry count.

I then did the same for “Barrack Obama”.

Therefore, we can see that Barack Obama has a more positive feeling among people compared to Donald Trump.

This is because the percent love_count is of the total feeling is

Donald Trump: (1561957/3750943)*100 = 41.64% Barack

Obama: (835889/1417875)*100 = 58.95%

Task B: Graphing the Data in R

1)

Part A

```
Bryan@LAPTOP-67JOQH14 /cygdrive/c/Users/Bryan/Downloads
$ grep -i "Trump" FB_Dataset.csv | awk -F ',' $21'{print $21}' > timestamp_trump.txt
```

$ grep –i which searches for any post where “Trump” is mention ignoring case distinctions from the file FB_Dataset.csv. Then I pipe the results awk which processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. And I have passed the instruction to print the column 21(posted_at) and redirect it to save the data to a file called timestamp_trump.txt.

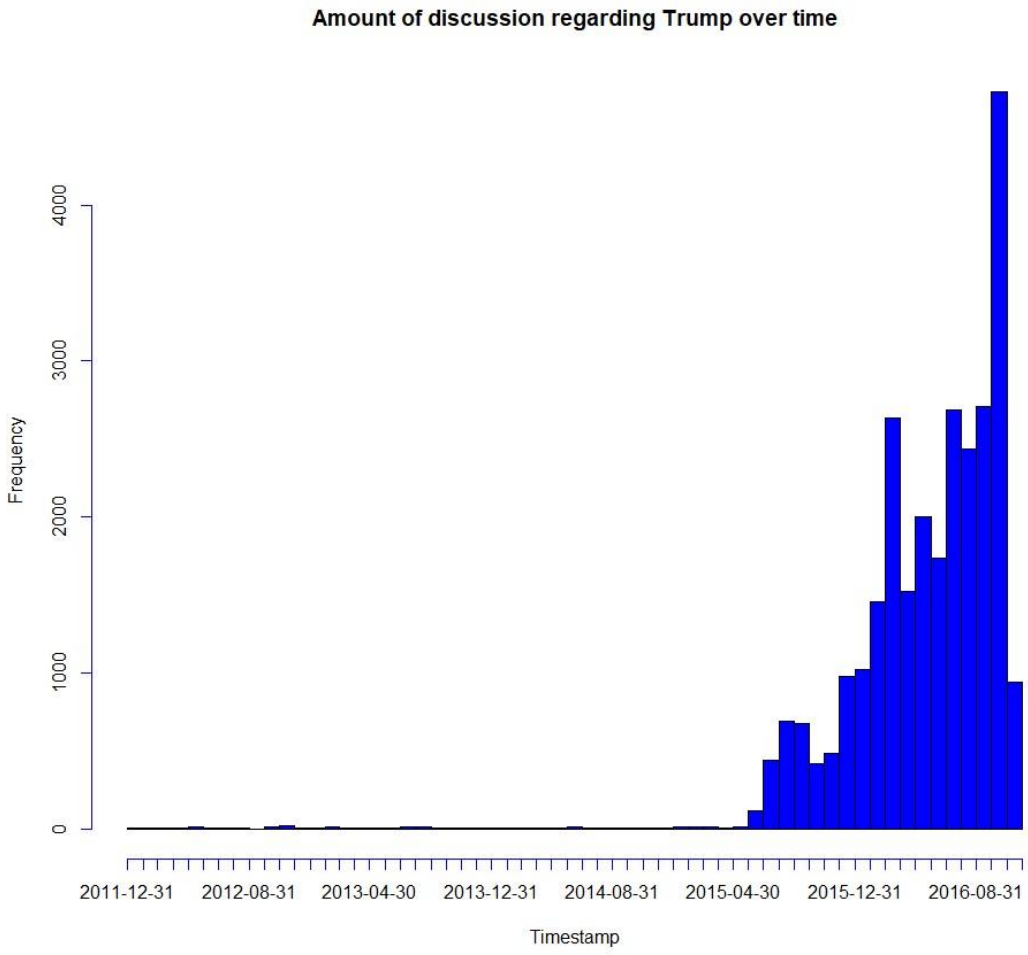
```
File <- read.csv("timestamp_trump.txt", header = FALSE) File$V1

<- strptime(File$V1, format("%d/%m/%y"))

hist(File$V1, breaks = "months", freq = TRUE,main = "Amount of discussion regarding Trump over time",xlab = "Timestamp", col = "blue")
```

First, I read the timestamp_trump.txt and assigned it to the variable called File with specifying header = FALSE which states that there are no headers in the file.

Then, I used strptime which converts strings as timestamps to the first column of the file with the format %d/%m/%y.

Finally, I used the hist() function to plot the data that I have in File$V1 with the breaks which are the cutoff points for the bins set to months, showing the frequency, title of histogram being “Amount of discussion regarding Trump over time”, x axis being “Timestamp” and the colour of the graph being blue.

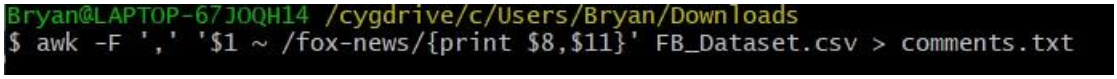
Output:



Part B

The histogram is skewed to the left as most of the data are on the right, with a few smaller values showing up on the left side of the histogram. This means that the amount of discussion regarding Trump only came up late in the timestamp which was only around after 2015 as seen in the graph where the frequency started to overall increase. The highest amount of discussion regarding trump was around 2016.

2) Part A



$ awk -F ',' '$1 ~ /fox-news/{print $8,$11}' FB_Dataset.csv > comments.txt

awk processes a file one line at a time. The flag –F ‘,’ tells awk that the columns are separated by commas. Then $1 ~ /fox-news/ gets the line which has the word “fox-news” in column 1(page_name) and {print $8,$11} prints column 8 (post type) and column 11(comment count) of that line. I then redirect it to save the data to a file called comments.txt.

install.packages("dplyr") library(dplyr)

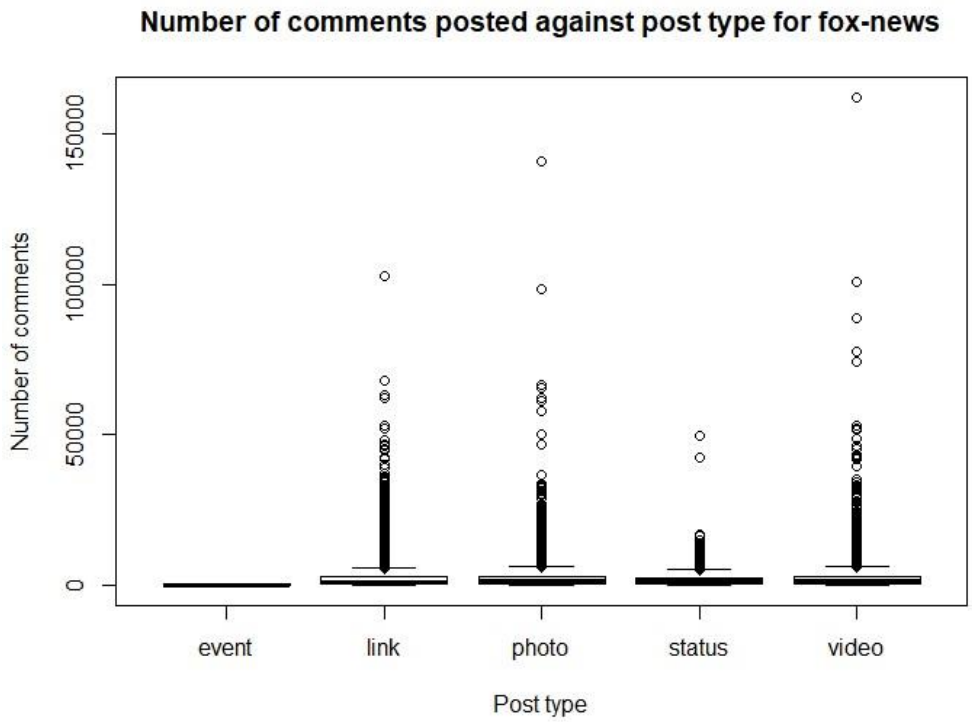
Firstly, I installed the dplyr package and imported it in order to use the function called filter.

fb <- read.csv('comments.txt',header = FALSE,sep = ' ')

boxplot(fb$V2~ fb$V1, data = fb , main ="Number of comments posted against post type for fox-news",xlab = 'Post type', ylab = 'Number of comments')

I read the comments.txt and assigned it to the variable called fb with specifying header = FALSE which states that there are no headers in the file where the field separator character is a space. Now, I am able to plot the boxplot using the boxplot() function.

Output:



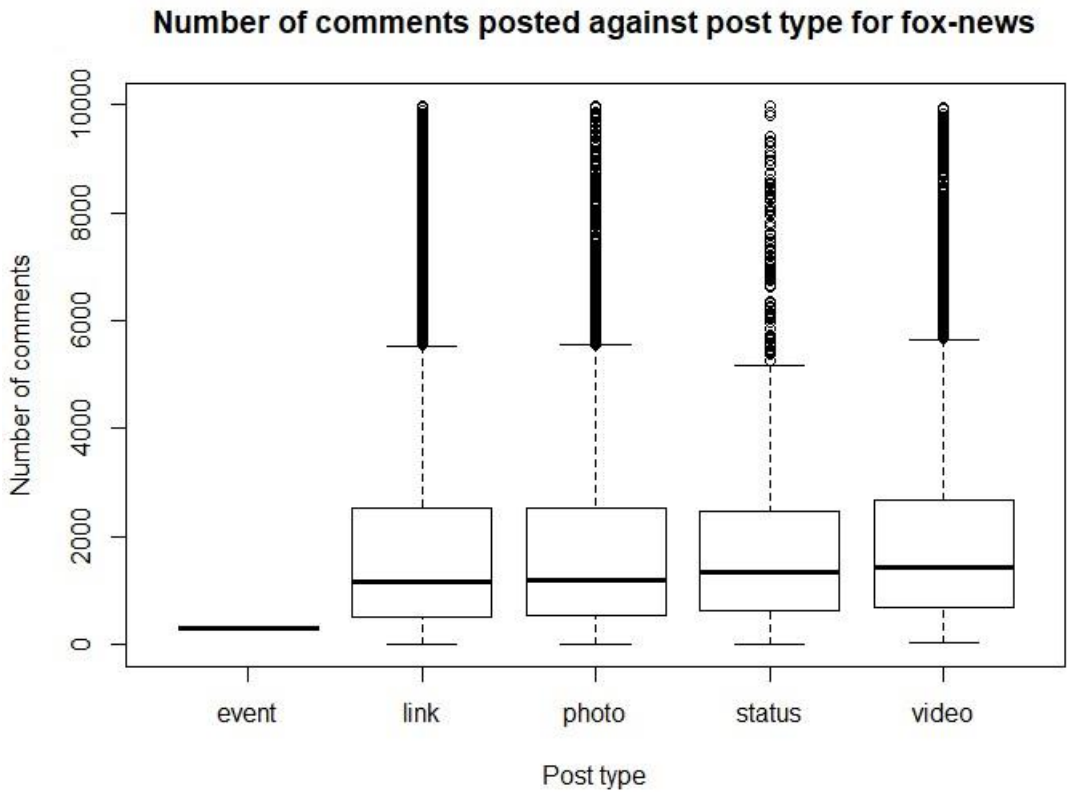
I could not infer anything from this plot as there are too many outliers present in the plot. However, we could see that video has the highest outlier and also a few high number outliers. Therefore, we may conclude from this plot that video was the most engaging post type. Part B

```
filt <- filter(fb, fb$V2 <= 10000)

boxplot(filt$V2~filt$V1, data = filt, main = "Number of comments posted against post type for fox-news",xlab = 'Post type', ylab = 'Number of comments')
```

To make it more readable, I redrew the boxplot filtering out comments_count that are greater than 10000 using the filter() function provided by the dplyr library and assigned it to the variable filt.

Then I created the box plot again using the boxplot() function.

Output:



Part C

From the boxplot above we can see that the type of post that has on average been most effective for fox-news is video.