Extracting from FB Dataset (Cygwin) and Plotting using Rstudio

Name: Ryan Li Jian Tang

Contents

Extraction of Data using Cygwin .....- 1 -

Visualisation using Rstudio .....- 5 -

Extraction of Data using Cygwin

Task A:

1.

Code: unzip FB_Dataset.csv.zip

Explanation:

Uncompressing the file to see the full size of the file.

Output:

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ unzip FB_Dataset.csv.zip
Archive:  FB_Dataset.csv.zip
  inflating: FB_Dataset.csv
   creating: __MACOSX/
  inflating: __MACOSX/._FB_Dataset.csv
```

Code: ls -lh FB_Dataset.csv

Explanation:

Code to find the size of the uncompressed file. The size of the file is 344 MB.

Output:

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ ls -lh FB_Dataset.csv
-rw-r--r--+ 1 Ryan Tang None 344M Sep 13  2019 FB_Dataset.csv
```

2.

Code: head -1 FB_Dataset.csv | less

Explanation:

Filter out the first row of the file to get the column headers. Then by looking at the output by using “less”, the delimiter is possibly “,”. To confirm this, in this output screen the input “/ <,>” is entered to see if it really is the delimiter. In this case, yes, it is.

Output:

```
page_name,post_id,page_id,post_name,message,description,caption,post_type,status
_type,likes_count,comments_count,shares_count,love_count,wow_count,haha_count,sa
d_count,thankful_count,angry_count,post_link,picture,posted_at
~
```

Code: head -1 FB_Data.set.csv | tr '\,' '\n' | cat -b

Explanation:

The above code is used to extract the first row from the file, containing the column headers. Then proceed to change the delimiter to "newline" to make each column header more visible. Final pipe command is used to just output the lines and pair it with the line number.

Output:

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ head -1 FB_Dataset.csv | tr '\,' '\n' | cat -b
     1  page_name
     2  post_id
     3  page_id
     4  post_name
     5  message
     6  description
     7  caption
     8  post_type
     9  status_type
    10  likes_count
    11  comments_count
    12  shares_count
    13  love_count
    14  wow_count
    15  haha_count
    16  sad_count
    17  thankful_count
    18  angry_count
    19  post_link
    20  picture
    21  posted_at
```

3.

Code: cat FB_Dataset.csv | awk -F',' '{print $3}' | grep -v -i "page_id" | uniq |wc -l

Explanation:

Cat is to print out the output from the file FB_Dataset.csv. Then the 2nd pipe command is to select the 3rd column, "page_id", by telling it that the delimiter between columns are ','. The 3rd pipe command is to select everything else but not the column header. The 4th pipe command is used to find the unique values and print out the unique lines. Finally, the final pipe command is used to count the number of unique lines. Thus there are a total of 15 unique pages.

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ cat FB_Dataset.csv | awk -F',' '{print $3}' | grep -v -i "page_id"| uniq | wc -l
15
```

4.

Code:

head -2 FB_Dataset.csv | awk -F',' '{print $21}'

tail -1 FB_Dataset.csv | awk -F',' '{print $21}'

Explanation:

Since the data is sorted, thus we just need to extract the first available data and the last data in the file. By using head and tail, we can easily find the range of dates. Furthermore, we can just use awk to select the "posted_at" column to reduce the amount of unnecessary information in the output. From the output we can see that the first post was made on 1st of January of 2012 at 12:30 A.M. and the last post was made on the 7th of November of 2016 at 11:45 P.M.

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ head -2 FB_Dataset.csv | awk -F',' '{print $21}'
posted_at
1/1/12 0:30

Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ tail -1 FB_Dataset.csv | awk -F',' '{print $21}'
7/11/16 23:45
```

5.

Code: cat FB_Dataset.csv | awk -F',' '{print $4}' | grep -i -o -c "donald trump"

Explanation:

Select the column "post_name" from the datafile by using the 2nd pipe command. The "post_name" is the 4th column in the datafile found from question 2. Then using the 3rd pipe command to proceed to find the all instances of Donald Trump being mentioned, "-i" is used to tell the code to ignore upper or lower cases. -o is to separate all instances of "donald trump" on any lines. Finally -c is to output the number of appearances of "donald trump" instead of outputting lines containing it. In this case "donald trump" has appeared 7592 times.

Output:

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ cat FB_Dataset.csv | awk -F',' '{print $4}' | grep -i -o -c "donald trump"
7592
```

Code: cat_FB_Dataset.csv | awk -F',' '{print $4, $21}' | grep -I "donald trump" | head -1

Explanation:

Select both columns of "post_name" and "posted_at" to find the timing at which the post was posted. Then using pipe commands to filter out post names that contains "Donald Trump" and its respective date. I can then use the command head -1 to find the first ever post of Donald Trump. This can be done as the data in the file was previously sorted in order according to question 4. Thus, when the filtering command iterates down the file it outputs the data in order. Title of Post is: "Donald Trump Staff Reaching Out to Financers .. Campaign Managers to Explore Third Party Bid" posted on 30/1/2012 21:07.

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ cat FB_Dataset.csv | awk -F',' '{print $4, $21}' | grep -i "donald trump" | head -1
Donald Trump Staff Reaching Out to Financers .. Campaign Managers to Explore Third Party Bid 30/1/12 21:07
```

Code: cat FB_Dataset.csv | awk -F',' '{print $4,$10,$11,$12,$13,$14,$15,$16,$17,$18,$21}' | grep -i "donald trump" | head -1

Explanation:

Filter out the post name, dates and all columns that contains "_counts" as they are all reactions towards to the post. Then afterwards only take the lines that contains "Donald Trump" by using the 3rd pipe command. Afterwards, since the output is in order I can then use "head -1" to output the earliest instance with its reactions. By looking at what each column means from question 2, there are no negative reactions towards the post and there is a considerable amount of likes for the post, at 174. The total number of reactions for this post is the sum of all number delimited by a <space>, thus 174 + 223 + 68 = 465 reactions.

Output:

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ cat FB_Dataset.csv | awk -F',' '{print $4, $10,$11,$12,$13,$14,$15,$16,$17,$18,$21}' | grep -i "donald trump" | head -1
Donald Trump Staff Reaching Out to Financers .. Campaign Managers to Explore Third Party Bid 174 223 68 0 0 0 0 0 0 30/1/12 21:07
```

6.

Code: cat FB_Dataset.csv | (head -n 1; grep -I "trump") | awk -F',' 'NR ==1 || $10 > 100 {print $2, $10}' | (sed -u 1q; sort -r -g -k2) > trump.txt

Explanation:

The 2nd pipe command is to keep the column headers and select the rows where "trump" is mentioned in the post content by using grep. The 3rd pipe command is to filter posts that have more than 100 likes and only keep the "post_id" and "likes_count" columns. 4th pipe command is to keep the first row (which is the column header) by skipping over it and sorting the rest of the posts based on "likes_count" column in descending order. Then outputting all of these into a text file.

Outputs:

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ cat FB_Dataset.csv |(head -n 1; grep -i "trump")|awk -F',' 'NR == 1 || $10 > 100 {print $2,$10}'| (sed -u 1q; sort -r -g -k2) > trump.txt
```

trump.txt:

Columns are delimited by a <space>

```
post_id likes_count
_22228735667216_1015396016795221722 368179
5550296508_10154298504746509 248012
18468761129_10153524839811130 229187
15704546335_10154108339971336 222119
15704546335_10154646163096336 218748
```

Task B1:

Code: cat FB_Dataset.csv | awk -F',' 'NR == 1 || $1 == "the-wall-street-journal" {OFS = ','; print $11, $8}' > the-wall-street-journal.csv

Explanation:

Keep the column header and take those that have “the-wall-street-journal” as their page name. Then proceed to print out the columns “comment_count” and “post_type” with the delimiter being ‘,’, as the output is going to be a comma separated value file.

```
Ryan Tang@Tang /cygdrive/e/Monash Files/Semester 2/FIT 1043/Assignment 3
$ cat FB_Dataset.csv |awk -F','  'NR == 1 || $1 == "the-wall-street-journal" {OFS = ","; print $11,$8}'> the-wall-street-journal.csv
```

7.

Visualisation using Rstudio

Code *using Rstudio*:

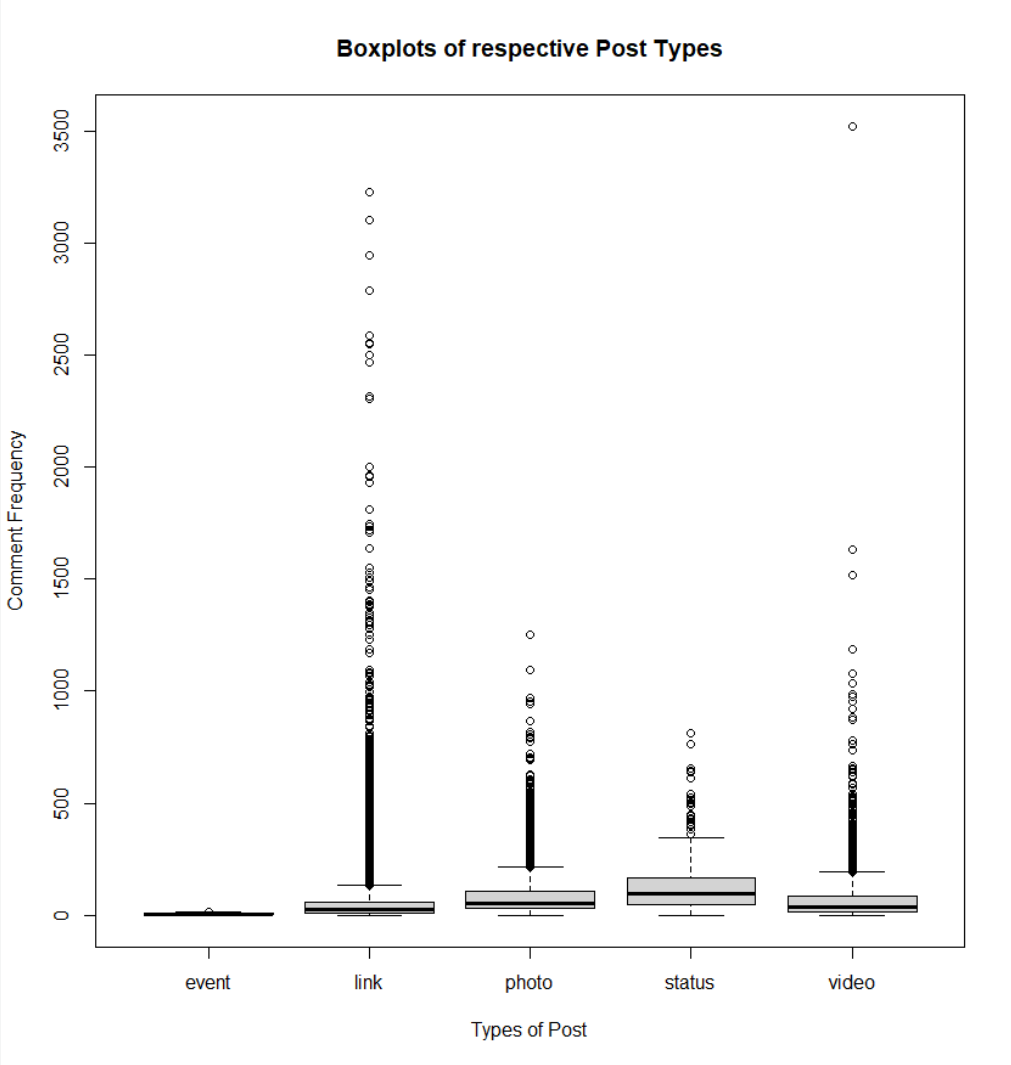
```
data =  read.csv(“the-wall-street-journal.csv”)

data = data[data$comments_count < 4000,]

boxplot(data$comments_count ~ data$post_type, main = “Boxplots of respective Post Types”, ylab = “Comment Frequency, xlab =”Types of Post”)
```

Explanation:

Read in the data and filter out any post that have more than 4000 comments. Then plot a boxplot based on the data frame.

Output:

On the next page



Looking at the boxplot for "link", it shows that a majority and bulk of posts contain only a few comments. However, by looking at the anomalous plots it can be seen that it occasionally produces some of the most engaging posts. This is further amplified by the fact that most of its anomalous data are higher than most other plots. On the other hand, the most successful and reliable type of post to provide an engaging experience is "status". This can be seen by its physical boxplot averaging higher than the rest followed by its median being higher than the rest as well.

8.

Code *using Rstudio*:

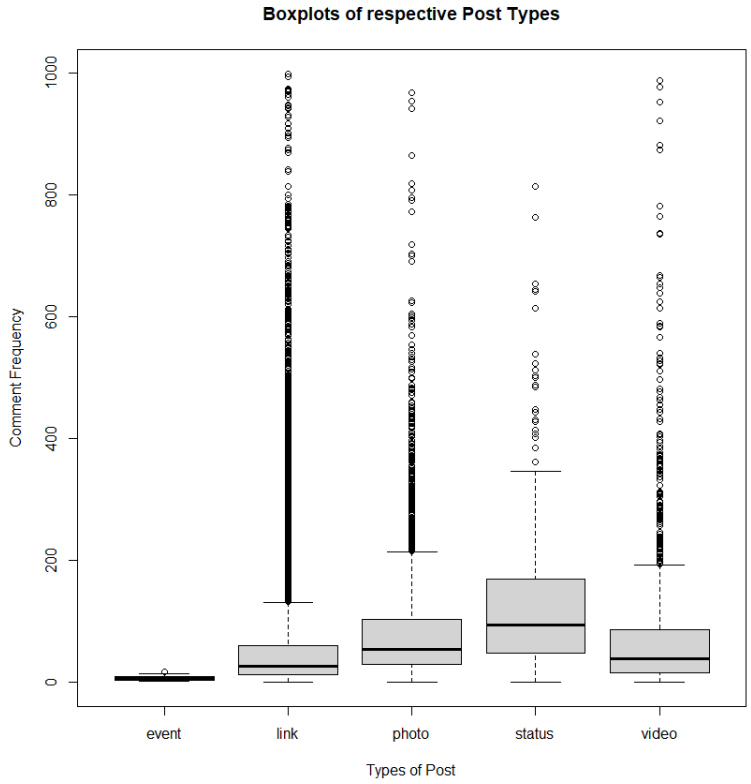
```
data = data[data$comments_count <= 1000,]
```

```
boxplot(data$comments_count ~ data$post_type, main = "Boxplots of respective Post Types", ylab = "Comment Frequency, xlab ="Types of Post")
```

Explanation:

Remove posts that have more than 1000 comments. Then proceed to plot a boxplot with the new modified data.

Output:



9.

According to the graph given by question 8, the type of post with the highest median is “status”. This can be seen with its 2nd quartile line being the highest out of all the types of post available.

Task B2:

10.

Code:

Using Cygwin to filter out the correct posts:

```
cat FB_Dataset.csv|awk -F',' 'BEGIN{OFS=",";} $1 == “abc-news” {print $5, $10, $11, $12, $13, $14, $15, $16, $17, $18, $21}' | grep -i "donald Trump" > Donald_Reactions.csv
```

Using Rstudio to do the processing of data:

```
data = read.csv ("Donald_Reactions.csv", header = FALSE)

data = data [, 2:11]

newDates = gsub (" .*", "",data$V11)

newDays = weekdays (as.Date(newDates , “%d/%m/%y”))

data$V11 = newDays

wanted = data.frame ("Reactions" = rowSums(data [,1:9]), "Day" = data$V11)

library(dplyr)
```

```
grp = group_by (wanted, Day)

result = summarise (grp, sum=sum (Reactions))

barplot (result$sum, main = "Total Reactions Daily", xlab = "Day", ylab = "Number of Reactions", names.arg = result$Day, col = "lightblue")
```

Explanation:

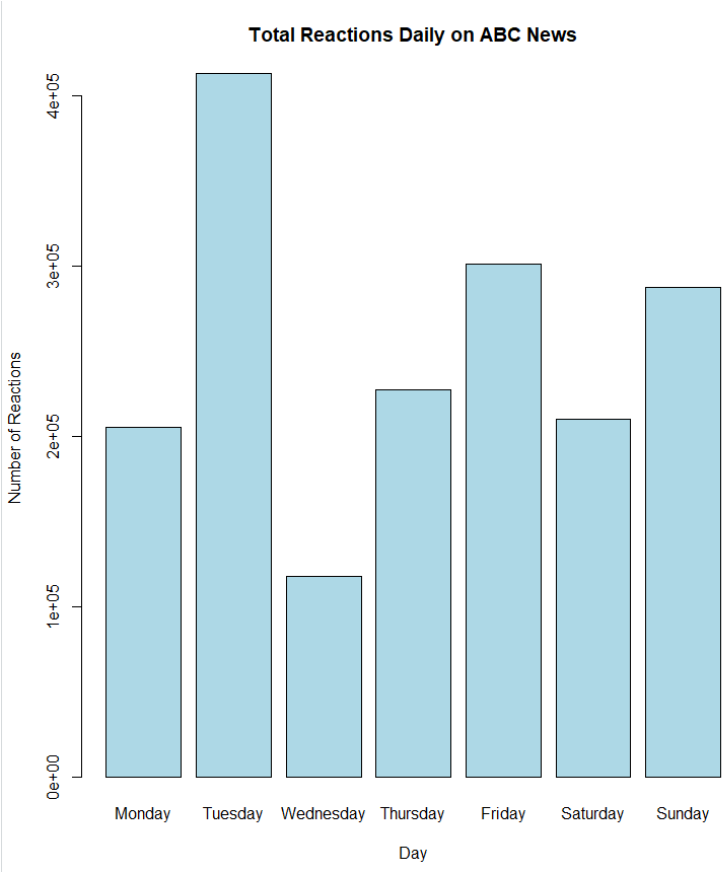
Cygwin:

First, only take posts that are published by “abc-news” using awk. Then take posts that mentions “Donald Trump”. Then posts that are selected gets outputted with all columns from 10 to 18 and the date column 21 to a file. The columns from 10-18 can all be considered as reactions. This is evident as when the counts are increased in these columns, a user somewhere must physically and actively comment or select an emotion to generate these “reactions”. The rest of the columns are all features of the post and does not provide any information on reactions from viewers.

Rstudio:

Using Rstudio to read in the file, I first remove the messages from the posts since they aren’t needed anymore. Afterwards since the “posted_at” column from the file has a combination of both date and time, I had to remove the time. Examining the data, the date and time were delimited by a <space>, thus, using gsub to extract the part before the <space> I could extract the dates and convert it to a weekday using R’s base command. Afterwards replacing it in the data frame. I proceeded to sum up all those reactions and create a new Data Frame with the newly summed ‘Reactions’ and Days. Finally proceeding to group the data based on the days, summarising the data and plotting it out.

Output:



11.

From the bar chart from question 10 the two most popular days are Tuesday and Friday. Furthermore, comparing weekdays to weekends there are no differences that can be seen. Through the graph there are no patterns that can be easily seen, with only Wednesday showing the least amount of reactions out of all 7 days.

12.

Code:

```
library(dplyr)

data = read.csv (“Donald_Reactions.csv”, header = FALSE)

data = data [,2:11]

date = gsub (“ .*“,””,data$V11)

day = weekdays(as.Date(date, "%d/%m/%y"))

hour = gsub (“.* “,””,data$V11)

hour = gsub (“:.*“,””, hour)

newData = data.frame ("Reactions" = rowSums(data[,1:9]), "day" =day, "hour" = hour)
```

Code to Plot for “Tuesday”:

```
filter = newData[(newData$day == "Tuesday"),]

filter = group_by(filter, hour)

filter = summarise(filter, sum = sum(Reactions))

barplot(filter$sum, main = "Total Reactions on Tuesdays", xlab = "Hour", ylab = "Number of Reactions",
names.arg = filter$hour, col = "lightblue")
```

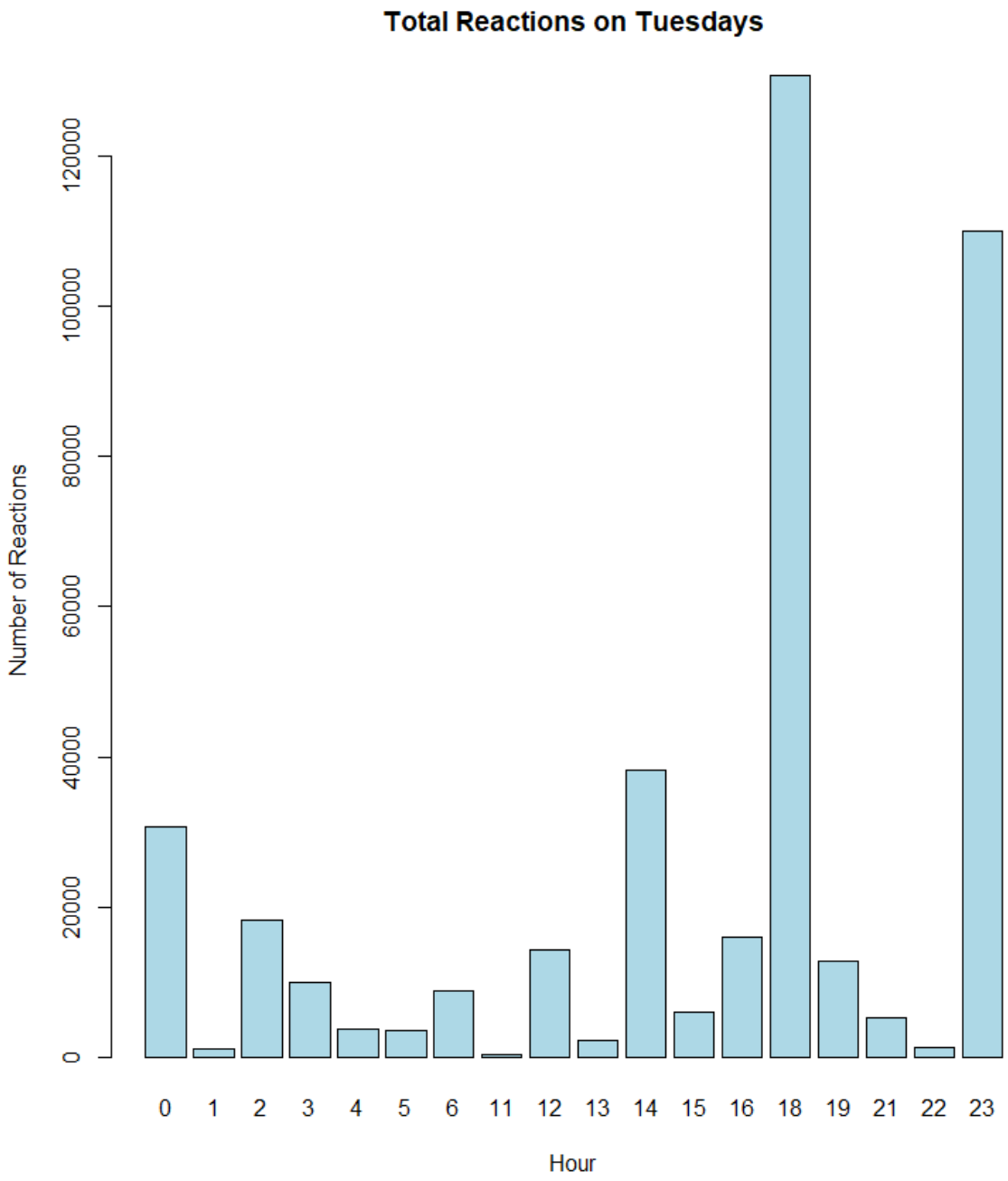
Code to plot for “Friday”:

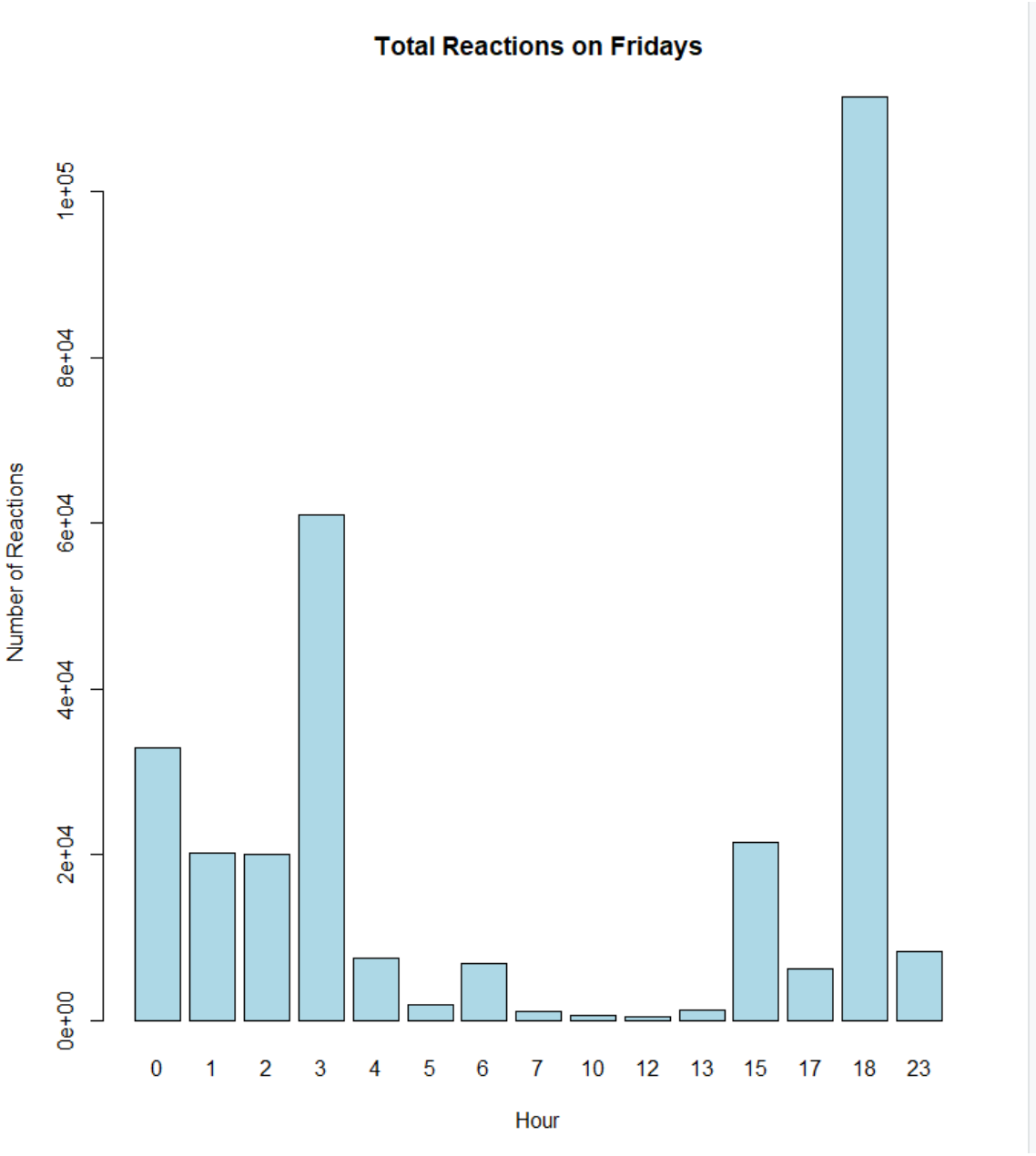
```
filter = newData[(newData$day == "Friday"),]

filter = group_by (filter, hour)

filter = summarise (filter, sum=sum (Reactions))

barplot(filter$sum, main = "Total Reactions on Fridays", xlab = "Hour", ylab = "Number of Reactions",
names.arg = filter$hour, col = "lightblue")
```

Explanation of Code:

First, extract the hours and date from the “posted_at” column. Then convert date into days as done before, then for hours remove the minutes and only keep the hour indicator for sorting into groups later on. Create a new data frame with the summed-up Reactions, Day and Hour. Filter out the day wanted, then sort them by the hours and summarise the reactions. Then for each respective day plot the bar plot.

Outputs:


Total Reactions on Tuesdays
Number of Reactions
0
20000
40000
60000
80000
100000
120000
0
1
2
3
4
5
6
11
12
13
14
15
16
18
19
21
22
23
Hour



Explanation of graph:

For Tuesdays, the most reactions that happened in the day occurred at 18:00 – 19:00. For Fridays, the highest number of reactions occurred from 18:00-19:00. Looking at both graphs, they seem to show that most viewers tend to react later in the day. This is probably since most viewers will only be free in the later parts of the day due to work, school or etc. Thus, showing little number of reactions during the day and a lot more during the night.

13.

a) Tuesday at 18:00-19:00 had the highest number of reactions at over 12000 reactions.

b) If the objective is to allow the most amount of people to react to the post, then it is a good idea to publish the post on the days and at the peak hours found. However, it is better to post a slightly bit earlier than the peak hours, possibly 1 hour before, to allow the post to circulate and be ready to be

viewed when the peak hours arrive. This allows for the most optimum number of views. On the other hand, for whatever reason, if it is to let the least amount of people view the post, then it will be a good idea to publish the post in the morning around 07:00 on a Wednesday. This is as most people will be on the way to work or school and will not be capable of reading the news at that time. Furthermore, during the day the published post will then be covered by other published news resulting in the least number of views.