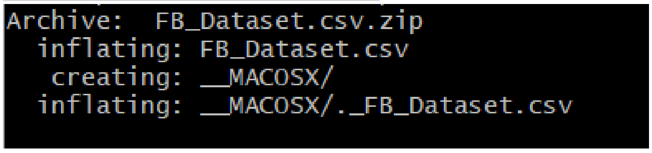
# FIT 1043 Assignment 3

Name: Vanessa Joyce Tan
Student ID: 30556864

## Task A

1) Decompress the zip file ‘FB_Dataset.csv.zip’.

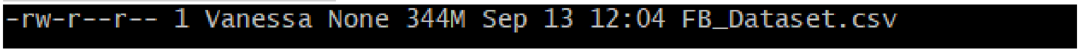
```
$ unzip FB_Dataset.csv.zip
```



```
Archive:  FB_Dataset.csv.zip
  inflating: FB_Dataset.csv
   creating: __MACOSX/
  inflating: __MACOSX/._FB_Dataset.csv
```

Checking the properties of the file so that we can look at the size of it.

```
$ ls -lh FB_Dataset.csv
```



```
-rw-r--r-- 1 Vanessa None 344M Sep 13 12:04 FB_Dataset.csv
```

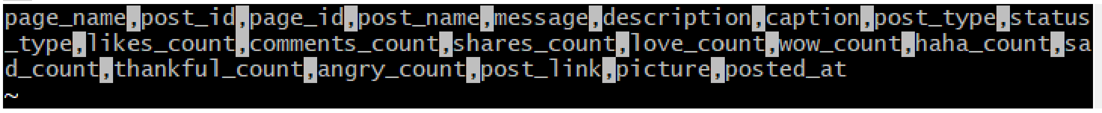
FB_Dataset.csv unzipped is 344 MB in size.

2) Use “less” to look at the first line.

```
$ head -1 FB_Dataset.csv | less
```

While reading the file, you can see what the delimiter is. The following code typed in “less” highlights the character ‘,’.

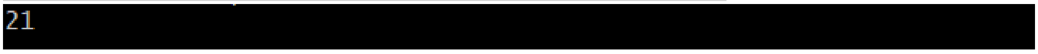
```
/,
```



```
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
~
```

Count how many fields (columns) are there in the first line. Using the “awk” utility, “ -F ‘,’ ” tells that the delimiter used is a comma and passing the command {print NF} is telling it to print the number of fields in the file ‘FB_Dataset.csv’. Pipe the results to the first row of the file to count the number of columns.

```
$ awk -F ',' '{print NF}' FB_Dataset.csv | head -1
```

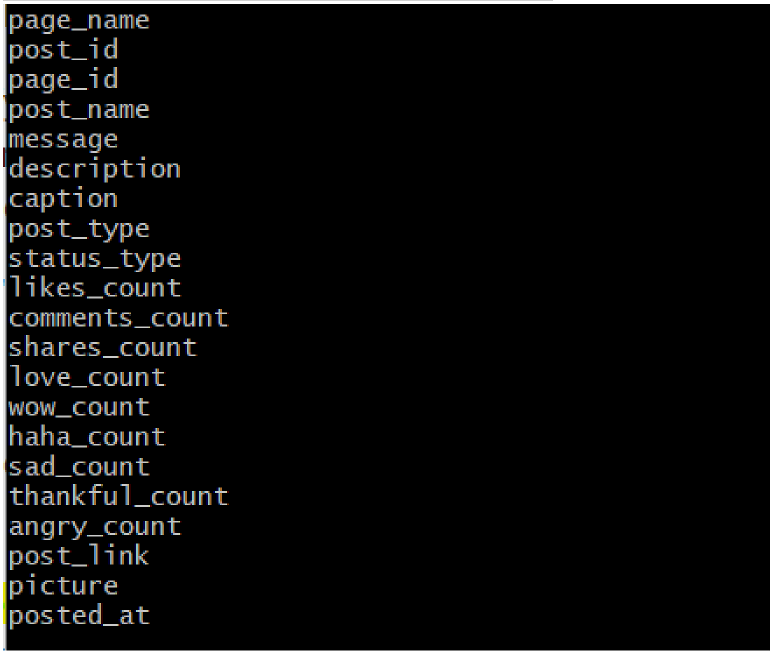


```
21
```

Therefore, we know that the delimiter used is a comma (,) and there are 21 columns.

3) Using the “tr” command to replace the commas on the first line with newline characters.

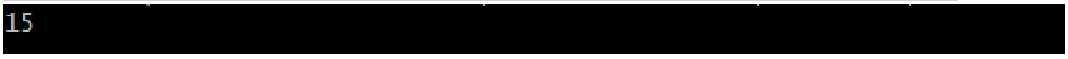
```
$ head -1 FB_Dataset.csv | tr ',' '\n'
```



```
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
sad_count
thankful_count
angry_count
post_link
picture
posted_at
```

4) Outputting the number of unique number of pages using ‘page_id’. Using the “cut” command, the “-d” flag indicates that the delimiter is a comma, “-f” flag is the column number. This is used to output the third column ‘page_id’ of FB_Dataset.csv. Pipe it to “tail –lines=+2” to exclude the header of the column. Sort it, then use the “-u” flag to only print out the unique lines. Pipe it to “wc” with the “-l” flag to count the number of lines.

```
$ cut -d',' -f3 FB_Dataset.csv | tail --lines=+2 | sort -u | wc -l
```

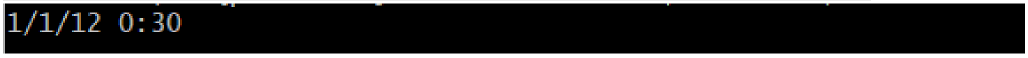


```
15
```

There are 15 unique pages.

5) Assuming the data is in order, the first row (excluding header) and the last row of the column ‘posted_at’ should give the date range.
Printing the first data for the column ‘posted_at’ using “awk”. Piping it to “head -2” gives the first two lines (including header) and then piping that to “tail -1” gives the data without the header.

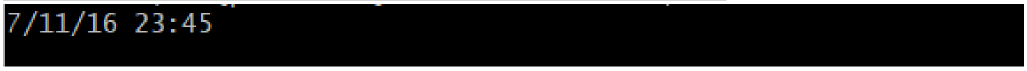
```
$ awk -F',' '{print $21}' FB_Dataset.csv | head -2 | tail -1
```



```
1/1/12 0:30
```

Printing the last data entry  in the column using the same technique above.

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

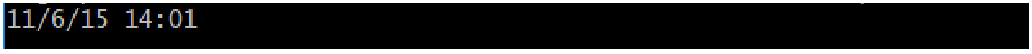


```
7/11/16 23:45
```

The date ranges from 1/1/12 to 7/11/16.

6) Finding the date and time of the first post regarding “Italian Dishes”. Using “grep”, the “-m 1” indicates that the file should stopped being read after the first match is found. The results were piped to awk to print the last field of that row.

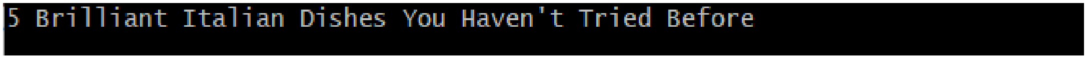
```
$ grep -m 1 "Italian Dishes" FB_Dataset.csv | awk -F',' '{print $21}'
```



```
11/6/15 14:01
```

Finding its post name. Same technique as above except it's printing the fourth column.

```
$ grep -m 1 "Italian Dishes" FB_Dataset.csv | awk -F',' '{print $4}'
```



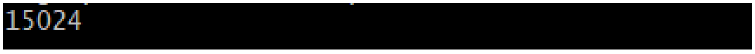
```
5 Brilliant Italian Dishes You Haven't Tried Before
```

The first mention of "Italian Dishes" in the file was a post with the name "Brilliant Italian Dishes You Haven't Tried Before" and was posted on 11/6/15 at 14:01 pm.

7) "Donald Trump" is mentioned 15024 times in the file.

Counting the number of times "Donald Trump" appears in the file. I used "grep" and the "-o" flag which outputs the matching string each time it is found in the file on separate lines. Then I used the "wc -l" command which counts the number of lines since they're all on separate lines.

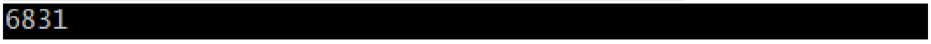
```
$ grep -o "Donald Trump" FB_Dataset.csv | wc -l
```



```
15024
```

8) "Barack Obama" is only mentioned 6831 times in the file. In terms of how many times their name was mentioned, Donald Trump is more popular on Facebook.
Counting the number of times "Barack Obama" appears in the file.

```
$ grep -o "Barack Obama" FB_Dataset.csv | wc -l
```



```
6831
```

9) Creating headers for the file "trump.txt". I used echo and the character ">" which writes the string before it into the file mentioned after the character.

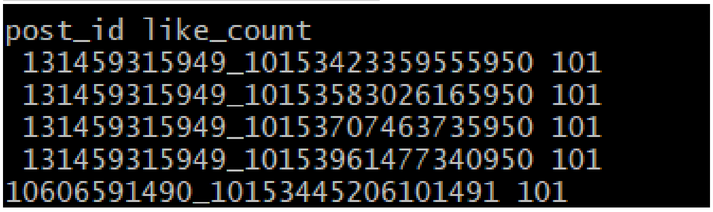
```
$ echo "post_id like_count" > trump.txt
```

Creating a new file "trump.txt" where it has the post_id of the posts containing the word "Trump" and it has more than 100 likes. Firstly, I used grep and the "-i" flag to find any row in FB_Dataset.csv containing the word "Trump" ignoring whether it was lower or uppercase. Then this was piped to awk to print only the second column (post_id) and the tenth column (likes_count) if the likes were more than 100. The output was sorted by the likes. All this was written to the file "trump.txt" using the character ">" which writes data to a file.

```
$ grep -i "Trump" FB_Dataset.csv | awk -F',' '$10>100{print $2, $10}'| sort
-k2 -n >> trump.txt
```

Reading the first 5 rows and header of resulting trump.txt file in less.

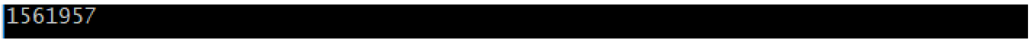
```
$ head -6 trump.txt | less
```



```
post_id like_count
 131459315949_10153423359555950 101
 131459315949_10153583026165950 101
 131459315949_10153707463735950 101
 131459315949_10153961477340950 101
10606591490_10153445206101491 101
```

10) Summing up number of love count for posts containing "Donald Trump". I used grep to look for the word "Donald Trump" in the FB_Dataset.csv file then piped it to awk to sum the values in the thirteenth column (love_count) of these rows.

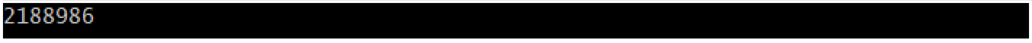
```
$ grep "Donald Trump" FB_Dataset.csv | awk -F',' '{sum+=$13}END{print sum}'
```



```
1561957
```

Summing up number of angry_count for posts containing "Donald Trump". Used same technique above except column is eighteenth column (angry_count).

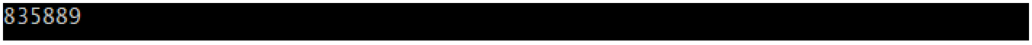
```
$ grep "Donald Trump" FB_Dataset.csv | awk -F',' '{sum+=$18}END{print sum}'
```



```
2188986
```

Summing up number of love_count for posts containing “Barack Obama”.

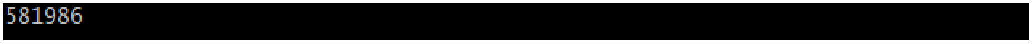
```
$ grep "Barack Obama" FB_Dataset.csv | awk -F',' '{sum+=$13}END{print sum}'
```



```
835889
```

Summing up number of angry_count for posts containing “Barack Obama”.

```
$ grep "Barack Obama" FB_Dataset.csv | awk -F',' '{sum+=$18}END{print sum}'
```



```
581986
```

From the statistics above, I gather that Barack Obama has more positive feeling among people. When looking at numbers alone, Donald Trump has more love_count than Obama. However, the angry_count for posts about Donald Trump is also extremely high and is very much higher than his love_count. Whereas Obama has more love_count than angry_count and we can hence conclude that Obama is more loved by the people. The high numbers of reaction on posts containing Donald Trump could be due to him being a person who says controversial things all the time which sparks a lot of posts and reaction to those posts. The counts also do not matter when we don’t know the context of the post. For all we know, these posts could be hate posts or posts supporting Donald Trump or Barack Obama.

## Task B

1) **A.**

Use grep to obtain rows containing the word “Trump” then output the posted_at column (column 21) containing the date and time of these posts and write them into a csv file called “time.csv”.

```
$ grep "Trump" FB_Dataset.csv | awk -F',' '{print $21}' > time.csv
```

Read the csv file into r.

```
t <- read.csv("C:/cygwin64/Assignment3/time.csv", header=FALSE)
```

Convert them from text values using the strptime() function.

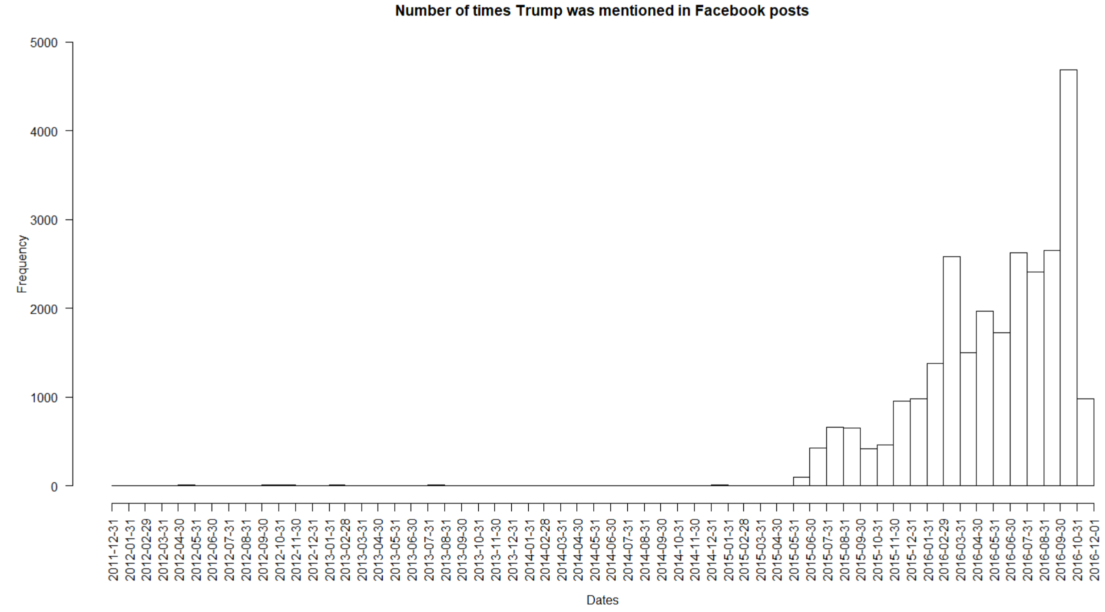
```
dates <- strptime(t$V1, format="%d/%m/%y %H:%M")
```

Plot the data into a histogram. Each bar represents a month of data.

```
hist(dates, breaks="months", freq=TRUE, main="Number of times Trump was mentioned in Facebook posts", ylim=c(0,5000), xlab="")
```

Move the x-axis label down so that it doesn’t overlap with the tick labels.

```
title(xlab="Dates", line=6)
```



**B.**

The histogram's distribution is skewed left (The number of posts about Trump is higher towards recent years). This could be due to the date of the U.S. elections held at the end of 2016. Starting from mid-year 2015, the posts about Trump started gradually increasing. This was probably around the time Trump announced that he was running for president and started campaigning. Some of these posts could be his campaign posts too. At the end of the year 2016, the U.S. held their elections and at this point Trump was a hot topic. Lots of people posted about the U.S. elections, some supporting Trump and others trying to convince people not to vote for him and so posts about him skyrocketed. After the U.S. elections, the buzz died down a bit and posts about him decreased.

2) **A.**

First, we need to create a text file with all the information on post_type (column 8) and comments_count (column 11) for fox-news.

Write headers into the text file "fox_comments.txt".

```
$ echo "post_type comments_count" > fox_comments.txt
```

Filter for "fox-news" and write the post_type and comment_count into the text file "fox_comments.txt".

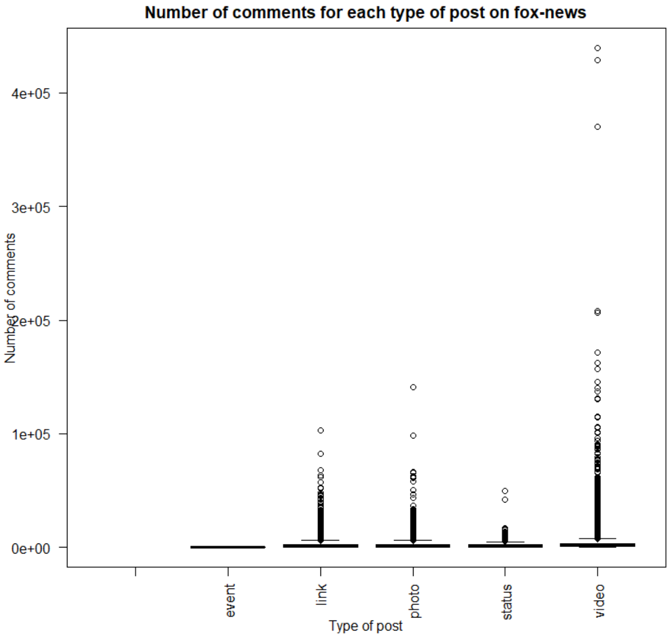
```
$ grep "fox-news" FB_Dataset.csv | awk -F',' '{print $8, $11}' >>
fox_comments.txt
```

Read text file as table in r.

```
table <- read.table(file="C:/cygwin64/Assignment3/fox_comments.txt", header=TRUE, sep=' ')
```

Plot boxplot for table.

```
boxplot(comments_count~post_type, data=table, main="Number of comments for each
type of post on fox-news", xlab="Type of post", ylab="Number of comments")
```



We can infer which post types of fox-news are the most engaging to their audience. Based on the number of comments, the most engaging post type is video because a few of their posts which were video types had a lot of comments. Their least engaging post type is event because the most comments they got on an event post is still much lower than the other post type's highest commented posts.

**B.**

Import library 'dplyr' for filtering.

```
library(dplyr)
```

Filter out the comments that are greater than 10, 000 in the table created earlier in part 2A. (The filter should keep those less than or equal to 10, 000).

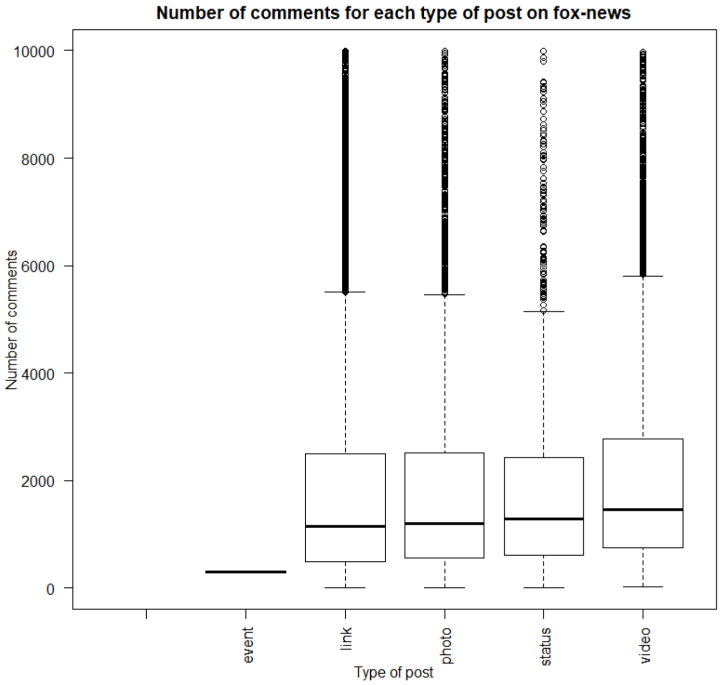
```
fltr <- filter(table, comments_count <= 10000)
```

Group the data by post_type and apply the filter.

```
grp <- group_by(fltr, post_type)
```

Redraw the boxplot for the filtered values.

```
boxplot(comments_count~post_type, data=grp, main="Number of comments for each type of post on fox-news", xlab="Type of post", ylab="Number of comments")
```



**C.**

As we can see from the boxplot above, the post type which has the highest median comment count for “fox-news” is video type.