Investigating Facebook data using shell commands and graphing the data in R

Task A: Investigating Facebook data using shell commands

1. The file FB_Dataset.csv.zip can be decompressed using the unzip command.
   `unzip FB_Dataset.csv.zip`

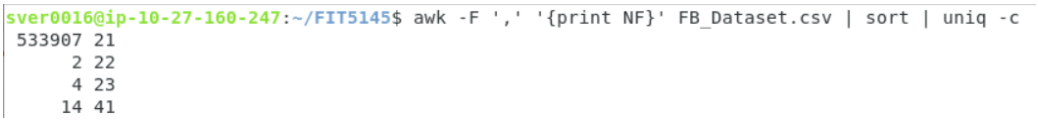
   The size of the file can be determined using below command. The file size is 344 MB.
   `ls -sh FB_Dataset.csv`

2. By checking the first line in the CSV file, we see that each field name is separated by a comma. Hence, we can say that the delimiter used here is comma. Below is the screenshot for the command:

   `head -1 FB_Dataset.csv`

   To see the number of columns, we can use below command. In this 'awk' is used for pattern scanning, '-F' is used to denote the field separator (in this case, it is comma) and 'NF' denotes number of fields in current record. Using pipe functionality, we can supply output of one command as input to other command. By doing so, we sort the data and then pick unique entries by pre-fixing the frequency.

   `awk -F ',' '{print NF}' FB_Dataset.csv | sort | uniq -c`



   From the above screenshot, we can deduce that there are 533907 rows with 21 columns, 2 rows with 22 columns, 4 rows with 23 columns and 14 rows with 41 columns. In all the cases, we see that there are only 21 column headings but rest of the column names are not present.

3. As stated in the previous answer, there are 21 columns with proper column heading whereas rest of the columns have no heading. Below code gives the column names which have heading. The columns are namely 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 and posted_at.

```
head -1 FB_Dataset.csv
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ head -1 FB_Dataset.csv
page_name,post_id,page_id,post_name,message,description,caption,post_type,status_type,likes_count,comments_count,shares_count,love_cou
nt,wow_count,haha_count,sad_count,thankful_count,angry_count,post_link,picture,posted_at
```

4. The number of facebook posts can be determined with the help of post_id column. In order to find out exact number of posts, we need to filter out NULLs as they don't give any proper information about a post and the heading. The below command displays the total number of posts.

```
awk -F ',' '{if($2!="")}{print $2}}' FB_Dataset.csv | tail -n+2 | uniq
| wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($2!="")}{print $2}}' FB_Dataset.csv | tail -n+2 | uniq | wc -l
533924
```

As mentioned earlier, awk is used for pattern scanning, '-F' for field separator (comma in this case), if condition to filter out NULLs, once we select whole column 2 data (post_id), we take a subset of the data (data except header), take unique count of the posts and then count the number of lines. It is interesting to note that post_id '5.55E+35' has two entries and there's a NULL in this column. Therefore, 3 records (header, NULL and repeated post_id) are excluded from over all count/field entries of 533927.

If we are to consider the number of facebook posts without considering one repeated value earlier, below code serves the purpose.

```
awk -F ',' '{if($2!="")}{print $2}}' FB_Dataset.csv | tail -n+2 | wc -
l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($2!="")}{print $2}}' FB_Dataset.csv | tail -n+2 | wc -l
533925
```

If we are to consider the number of facebook posts without considering the empty string and one repeated value earlier, below code serves the purpose.

```
awk -F ',' '{print $2}' FB_Dataset.csv | tail -n+2 | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{print $2}' FB_Dataset.csv | tail -n+2 | wc -l
533926
```

5. The only date field is posted_at which is the 21st column. Using awk command, we look for pattern matching of 'posted_at' so that we can filter out the header and then display 1st record and last record using head -1 and tail -1 respectively. We use '&&' as conditional statement execution. Here, the second statement will be executed only if the first statement is executed successfully. The below command displays the minimum and maximum dates in that particular column.

```
awk -F ',' '{if(tolower($21)!~"posted_at"){print $21}}'
FB_Dataset.csv | head -1 && awk -F ','
'{if(tolower($21)!~"posted_at"){print $21}}' FB_Dataset.csv | tail -1
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if(tolower($21)!~"posted_at"){print $21}}' FB_Dataset.csv | head -1 && awk -F ','
 '{if(tolower($21)!~"posted_at"){print $21}}' FB_Dataset.csv | tail -1
1/1/12 0:30
7/11/16 23:45
```

6. The number of pages can be obtained by counting unique number of rows in page_id column. Using awk command, we look for the pattern 'page_id' so that we filter out the header, pick the unique subset among all the rows selected and then count the number of lines. This gives us the result 15 implying that there are 15 unique pages. Below code is for the same :

```
awk -F ',' '{if(tolower($3)!~"page_id"){print $3}}' FB_Dataset.csv |
uniq | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if(tolower($3)!~"page_id"){print $3}}' FB_Dataset.csv | uniq | wc -l
15
```

7. There are a total of 512809 posts. As mentioned by Mahasa in forum discussions, post_id consists of page_id and message_id. Here, we need to find unique number of message_id. This is can be done using awk command with field separator as comma. Firstly, we print the post_id and then using pipe command, we split the data based on '_' and then print the message_id. The resultant record contains the header which isn't required. So, we extract the subset of the whole data such that we exclude header (we do this using tail command) and finally we pick the unique entries using uniq command and display the number of lines using wc -l command. Below is the command used.

```
awk -F ',' '{print $2}' FB_Dataset.csv | awk -F '_' '{print $2}' |
tail -n+2 | uniq | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{print $2}' FB_Dataset.csv | awk -F '_' '{print $2}' | tail -n+2 | uniq | wc -l
512809
```

If we are to exclude empty string in the column, below code serves the purpose.

```
awk -F ',' '{if($2!="")}{print $2}}' FB_Dataset.csv | awk -F '_'
'{print $2}' | tail -n+2 | uniq | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($2!="")}{print $2}}' FB_Dataset.csv | awk -F '_' '{print $2}' | tail -n+2 | uniq | wc -l
512808
```

There's another way we can look at it - there are a total of 510122 posts. One page can have multiple posts – this implies that for a particular page id, we have multiple post names and using both columns, we need to figure out the count. We can do so by using awk command, filtering out the header, filtering out empty fields and 'NULL', taking a unique subset and then counting the number of records. Below is the command for the same.

```
awk -F ',' '{if($4!~"post_name" && ($4!="" || $4!="NULL")){print $3,
$4}}' FB_Dataset.csv | uniq | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($4!~"post_name" && ($4!="" || $4!="NULL")){print $3,$4}}' FB_Dataset.csv | uniq | wc -l
510122
```

Alternatively, if we are to consider only the post_name column, the changed code also results into same answer as above.

```
awk -F ‘,’ ‘{if($4!~”post_name” && ($4!=“” || $4!=“NULL”)){print $4}}’ FB_Dataset.csv | uniq | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($4!~"post_name" && ($4!="" || $4!="NULL")){print $4}}' FB_Dataset.csv | uniq | wc -l
510122
```

8. The first mention of ‘Italian Dishes’ was made in the row number 308739. It was mentioned in the page name ‘the-huffington-post’ with the post name as ‘5 Brilliant Italian Dishes You Haven't Tried Before’. We can use grep command for this. ‘-n’ gives the line number and ‘-m 1’ gives the first occurrence of the text supplied. Below is the code:

```
grep -n -m 1 ‘Italian Dishes’ FB_Dataset.csv
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ grep -n -m 1 'Italian Dishes' FB_Dataset.csv
308739:the-huffington-post,18468761129_10153133124136130,18468761129,5 Brilliant Italian Dishes You Haven't Tried Before,Move over  fettuccine alfredo.,No fettuc
cine alfredo or penne alla vodka here. Each of these recipes -- for Italian 'sushi ' roast chicken with a serious kick and more -- offers a fresh approach to the
 beloved cuisine.,huff.to,link,shared_story,397,35,277,0,0,0,0,0,0,http://huff.to/1f4MMOo,https://external.xx.fbcdn.net/safe_image.php?d=AQAJulr3tN0ACu5d&w=130&h
=130&url=http%3A%2F%2Fi.huffpost.com%2Fgen%2F3029028%2Fimages%2Fo-AUTHENTIC-ITALIAN-RECIPES-facebook.jpg&cfs=1,11/6/15 14:01
```

The above code displays the row number while below code displays the date and time when the phrase ‘Italian Dishes’ was mentioned i.e. 11/6/15 14:01.

```
awk -F ‘,’ ‘{if($0~”Italian Dishes”){print $21}}’ FB_Dataset.csv
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($0~"Italian Dishes"){print $21}}' FB_Dataset.csv
11/6/15 14:01
```

9. As per forum discussions, we are to count the number of occurrence of ‘Barack Obama’ per line in a file. The name ‘Barack Obama’ occurred 5639 times in the dataset. We can find it by using grep command. The option of grep command ‘-o’ gives us the exact match/only words of the pattern provided (in this case, ‘Barack Obama’) from a particular line. ‘\b’ is used to define the starting and the ending boundary. “[^’]” is used to exclude apostrophe (words such as “Barack Obama’s”) because grep by default captures such occurrences too. Since we do not want such occurrences, we use below command.

```
grep -o “\bBarack Obama\b[^’]” FB_Dataset.csv | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o "\bBarack Obama\b[^']" FB_Dataset.csv | wc -l
5639
```

Alternatively, if we are to check occurrence of ‘Barack Obama’ as a pattern anywhere in the file instead of as a whole phrase, the count is 6831. Below is the code for the same.

```
grep -o ‘Barack Obama’ FB_Dataset.csv | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o 'Barack Obama' FB_Dataset.csv | wc -l
6831
```

10. Similar to the above question, we can find out the number of occurrences of the name ‘Donald Trump’. We see that there are a total of 12268 occurrences of ‘Donald Trump’ as a whole phrase in the dataset and 15024 overall number of occurrences of ‘Donald Trump’ in the dataset. Compared to the occurrences of name ‘Barack Obama’, we see that occurrences of the name ‘Donald Trump’ is more. Therefore, Donald Trump is more famous compared to Barack Obama. Below are the commands used.

```
grep -o ‘Barack Obama’ FB_Dataset.csv | wc -l
grep -o “\bBarack Obama\b[^’]” FB_Dataset.csv | wc -l
grep -o ‘Donald Trump’ FB_Dataset.csv | wc -l
grep -o “\bDonald Trump\b[^’]” FB_Dataset.csv | wc -l
```

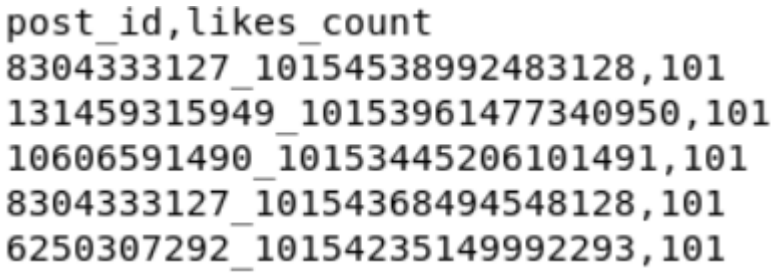
```
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o 'Barack Obama' FB_Dataset.csv | wc -l
6831
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o "\bBarack Obama\b[^']" FB_Dataset.csv | wc -l
5639
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o 'Donald Trump' FB_Dataset.csv | wc -l
15024
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o "\bDonald Trump\b[^']" FB_Dataset.csv | wc -l
12268
```

11. We first extract the column names or headers (‘post_id’ and ‘likes_count’) using cut command. We mention the delimiter of the dataset using ‘-d’ option of the cut command, select column ids (usually starts from 1) using ‘-f’ option of the cut command and extract out only the first row (header) and dump into ‘trump.txt’ file. To select the posts where ‘Trump’ (ignoring the case) is mentioned and where likes count is greater than 100, we use awk command. We mention the field separator, set up a condition on likes_count column (should be greater than 100) and print post_id, likes_count and message column details. Then using pipe functionality, we sort the data extracted in previous step based on the second column(likes_count) where ‘-V’ option is used to sort numeric values in between text. This is followed by searching occurrences of ‘Trump’ using grep -i command (ignoring the case) and then with the help of awk command, we simply append the post_id and likes_count columns into ‘trump.txt’ file. Below is the code for the same.

```
cut -d ‘,’ -f 2,10 FB_Dataset.csv | head -1 > trump.txt
awk -F ‘,’ ‘{if($10>100){print $2, $10, $5}}’ FB_Dataset.csv | sort -
V -k2 | grep -i “trump” | awk -F ‘ ‘ ‘OFS=”,” {print $1, $2}’ >>
trump.txt
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ cut -d ',' -f 2,10 FB_Dataset.csv | head -1 > trump.txt
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($10>100){print $2, $10, $5}}' FB_Dataset.csv | sort -V -k2 | grep -i "trump" | awk -F ' ' 'OFS="," {print
$1, $2}' >> trump.txt
```

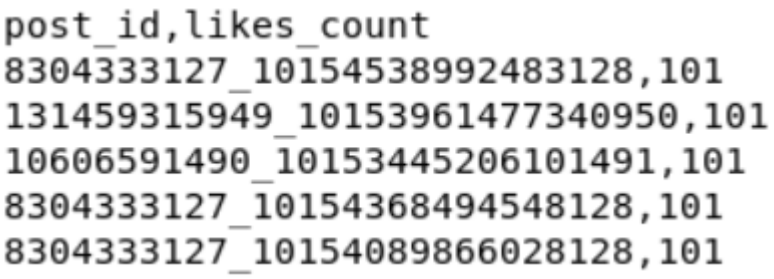
Below is the snapshot of first few rows of trump.txt file.



If we are to consider only the term 'Trump' ignoring the case, the code changes as below.

```
cut -d ',' -f 2,10 FB_Dataset.csv | head -1 > trump1.txt
awk -F ',' '{if($10>100){print $2, $10, $5}}' FB_Dataset.csv | sort -
V -k2 | grep -i "\btrump\b[^']" | awk -F ' ' 'OFS=",," {print $1, $2}'
>> trump1.txt
```

Therefore, results changes as below. Screenshot of few rows from trump1.txt file.



12. The total number of love_count for Barack Obama and Donald Trump is 835889 and 1561957 respectively where as the total number of angry_count for Barack Obama and Donald Trump is 581989 and 2188986 respectively. If we subtract the angry count from love count, we see that the result for Obama is 253900 where as the result for Trump is -627029. The negative result for Trump implies that he has more angry count compared to love count whereas Obama has more love count compare to angry count. Hence, we can conclude that Barack Obama has more positive feeling compared to Donald Trump.

```
tail -n+2 FB_Dataset.csv | grep "Barack Obama" | awk -F ',' '{sum +=
$13} END {print sum}'

tail -n+2 FB_Dataset.csv | grep "Donald Trump" | awk -F ',' '{sum +=
$13} END {print sum}'

tail -n+2 FB_Dataset.csv | grep "Barack Obama" | awk -F ',' '{sum +=
$18} END {print sum}'

tail -n+2 FB_Dataset.csv | grep "Donald Trump" | awk -F ',' '{sum +=
$18} END {print sum}'
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | grep "Barack Obama" | awk -F ',' '{sum += $13} END {print sum}'
835889
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | grep "Donald Trump" | awk -F ',' '{sum += $13} END {print sum}'
1561957
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | grep "Barack Obama" | awk -F ',' '{sum += $18} END {print sum}'
581989
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | grep "Donald Trump" | awk -F ',' '{sum += $18} END {print sum}'
2188986
```

On the other hand, if we are to consider only message column, the code changes as below.

```
tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}'
| grep "Barack Obama" | awk -F ',' '{sum += $2} END {print sum}'

tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}'
| grep "Barack Obama" | awk -F ',' '{sum += $3} END {print sum}'

tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}'
| grep "Donald Trump" | awk -F ',' '{sum += $2} END {print sum}'

tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}'
| grep "Donald Trump" | awk -F ',' '{sum += $3} END {print sum}'
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}' | grep "Barack Obama" | awk -F ',' '{sum+=$2} END
{print sum}'
786896
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}' | grep "Barack Obama" | awk -F ',' '{sum+=$3} END
{print sum}'
539859
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}' | grep "Donald Trump" | awk -F ',' '{sum+=$2} END
{print sum}'
435717
sver0016@ip-10-27-160-247:~/FIT5145$ tail -n+2 FB_Dataset.csv | awk -F ',' 'OFS="," {print $5, $13, $18}' | grep "Donald Trump" | awk -F ',' '{sum+=$3} END
{print sum}'
396766
```

The total number of love_count for Barack Obama and Donald Trump in message column is 786896 and 435717 respectively where as the total number of angry_count for Barack Obama and Donald Trump is 539859 and 396766 respectively. If we subtract love and angry count for Barack Obama, we get the result 247037 whereas for Donald Trump, we get the result 38951. This implies that there are more counts of love for Barack Obama compared to that of Donald Trump. Hence, Barack Obama has more positive feeling.

Task B: Graphing the data in R

1. Following the forum discussions, there are multiple ways to answer this question. One way could be the overall occurrences of Trump in dataset. The count for it is 52673. Below is the code for the same.

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

```
sver0016@ip-10-27-160-247:~/FIT5145$ grep -o "Trump" FB_Dataset.csv | wc -l
52673
```

Alternatively, we can also look for the pattern in each of the columns namely, post_name, message and description. The count for “Trump” in post_name, message and description is 19581, 22338 and 9683 respectively. Below are the codes for the same.

```
awk -F ‘,’ ‘{print $4}’ FB_Dataset.csv | grep -o “Trump” | wc -l
awk -F ‘,’ ‘{print $5}’ FB_Dataset.csv | grep -o “Trump” | wc -l
awk -F ‘,’ ‘{print $6}’ FB_Dataset.csv | grep -o “Trump” | wc -l
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{print $4}' FB_Dataset.csv | grep -o "Trump" | wc -l
19581
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{print $5}' FB_Dataset.csv | grep -o "Trump" | wc -l
22338
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{print $6}' FB_Dataset.csv | grep -o "Trump" | wc -l
9683
```

2. We first extract the ‘posted_at’ column for all the trump related posts from the dataset and save in the ‘trumpPosts.csv’ file. Below is the code for the same. Here, empty strings are excluded.

```
grep “Trump” FB_Dataset.csv | awk -F ‘,’ ‘{if($21!=””){print $21}}’ >
trumpPosts.csv
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ grep "Trump" FB_Dataset.csv | awk -F ',' '{if($21!=""){print $21}}' > trumpPosts.csv
```

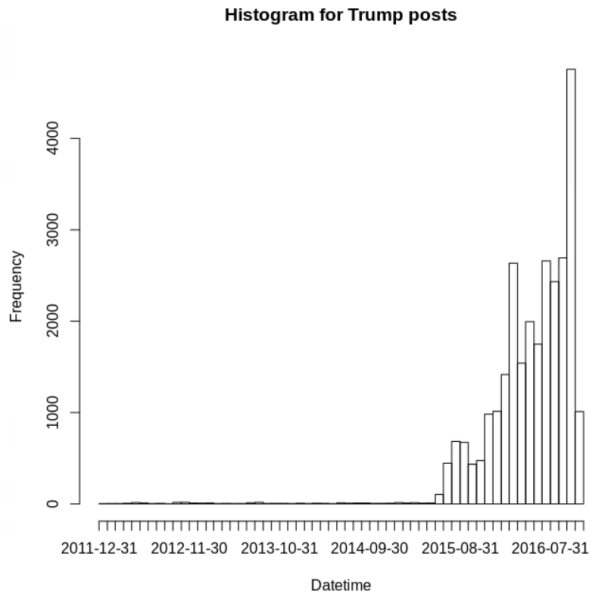
As it can be seen in the snapshot below, the datetime format of the column ‘posted_at’ is in the form of dd/mm/yy hh:mm. Hence, the format that we are supposed to use is %d/%m/%y %H:%M.

```
sver0016@ip-10-27-160-247:~/FIT5145$ grep "Trump" FB_Dataset.csv | awk -F ',' '{if($21!=""){print $21}}' | head -5
29/1/12 19:48
30/1/12 21:07
2/2/12 15:53
3/4/12 0:49
5/10/12 2:00
```

2.1. The data we have in the ‘trumpPosts.csv’ file is in the string format. This needs to be converted into datetime format which can be done using strptime function. Below code reads the data from the file in form of a table in R using read.table command and then we convert that data into datetime format using strptime function. After we convert the data into desired format, we can plot a histogram for the same.

```
dateString <- read.table(“trumpPosts.csv”, header=FALSE, sep=”\n”)
dates <- strptime(dateString[,1], “%d/%m/%y %H:%M”)
hist(dates, breaks = “months”, xlab=”Datetime”, ylab=”Frequency”,
main = “Histogram for Trump posts”, freq=TRUE)
```

```
> dateString <- read.table("trumpPosts.csv", header=FALSE, sep="\n")
> dates <- strptime(dateString[,1], "%d/%m/%y %H:%M")
> hist(dates, breaks="months", xlab="Datetime", ylab="Frequency", main="Histogram for Trump posts", freq=TRUE)
```



2.2. We see that from 2011 to 2014 end, there was barely any buzz for "Trump" but from the year 2015 to the end of the period, we see gradual increase in the amount of discussion regarding "Trump". The sudden increment can be accounted to Trump's decision to run for USA's presidential elections in the year 2015. That is when the campaigning began and a media image was build for him. Also, with the advancement in the technology, people are more exposed to the news easily. As Facebook is one of the trending social media, it is no surprise that there were good amount of discussions regarding Donald Trump.

3. We can extract the desired data from actual dataset using below commands. The first command will fetch only the column names while the second command will fetch relevant rows.

```
head -1 FB_Dataset.csv > mediaPosts.csv
awk -F ‘,’ ‘{if($1~”abc-news” || $1~”cnn” || $1~”fox-news”){print}}’
FB_Dataset.csv >> mediaPosts.csv
```

```
sver0016@ip-10-27-160-247:~/FIT5145$ head -1 FB_Dataset.csv > mediaPosts.csv
sver0016@ip-10-27-160-247:~/FIT5145$ awk -F ',' '{if($1~"abc-news" || $1~"cnn" || $1~"fox-news"){print}}' FB_Dataset.csv >> mediaPosts.csv
```

3.1. Once we have the required data in a csv file (mediaPosts.csv), we can read the data in R using below command.

```
mediaPosts <- read.csv(“mediaPosts.csv”, header=TRUE, sep=”,”)
```

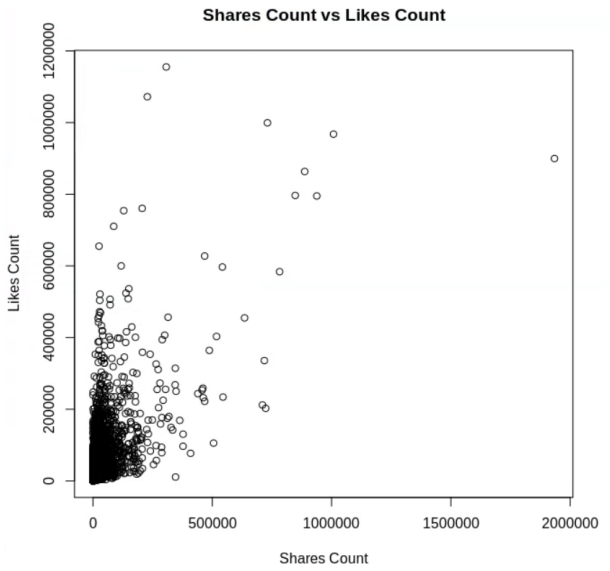
```
> mediaPosts <- read.csv("mediaPosts.csv", header=TRUE, sep=",")
```

3.2. We can display the graph using plot function in R. By default, this function generates scatter plot which we would be using to determine any correlation between shares count and likes count. The below command plots the graph.

```
plot(mediaPosts$shares_count, mediaPosts$likes_count, xlab=”Shares
Count”, ylab=”Likes Count”, main=”Shares count vs Likes Count”)
```

```
> plot(mediaPosts$shares_count, mediaPosts$likes_count, xlab="Shares Count", ylab="Likes Count", main="Shares Count vs Likes Count")
```

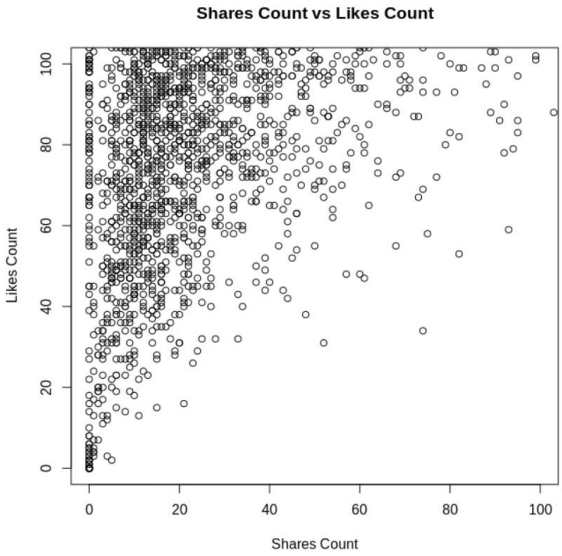
The scatter plot appears somewhat as below with shares count on x-axis and likes count on y-axis. To get better understanding about the graph, I've set appropriate x label, y label and title. It can be noticed that most of the data points are clustered between 0 and 30000 counts of shares and likes. We see no clear correlation between shares and likes count.



However, if we are to reduce the scale and see the trend, it can be noticed that in general, people tend to 'like' a post than to 'share' the same post (screenshot below). Hence, overall, we see that there more number of likes on a particular post compared to the number of shares. We also see that there are very few cases where there are huge number of likes and shares (points that are away from the cluster in the above snapshot).

```
plot(mediaPosts$shares_count, mediaPosts$likes_count, xlab="Shares Count", ylab="Likes Count", main="Shares count vs Likes Count", xlim=c(0,100), ylim=c(0,100))
```

```
> plot(mediaPosts$shares_count, mediaPosts$likes_count, xlab="Shares Count", ylab="Likes Count", main="Shares Count vs Likes Count", xlim=c(0,
100), ylim=c(0,100))
```

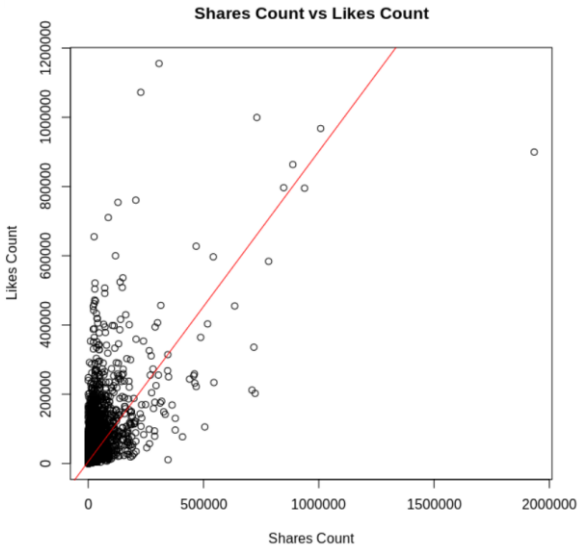


3.3. We can fit a linear regression model using the R command 'lm'. Below is the code for the same followed by screenshot of the plot.

```
mediaPosts <- read.csv("mediaPosts.csv", header=TRUE, sep=",")

lrModel <- lm(likes_count ~ shares_count, data=mediaPosts)

plot(mediaPosts$shares_count, mediaPosts$likes_count, xlab="Shares
Count", ylab="Likes Count", main="Shares count vs Likes Count")

abline(lrModel, col="red")
```

```
> mediaPosts <- read.csv("mediaPosts.csv", header=TRUE, sep=",")
> lrModel <- lm(likes_count ~ shares_count, data=mediaPosts)
> plot(mediaPosts$shares_count, mediaPosts$likes_count, xlab="Shares Count", ylab="Likes Count", main="Shares Count vs Likes Count")
> abline(lrModel, col="red")
```

In the first line, we read the data into a vector variable named as mediaPosts. In the second step, we fit the data (likes and shares count) to linear regression model. In the consecutive steps, we plot the graph between shares and likes count and the linear fit shown as a red line in the graph.



The linear model isn't the best fit for the data points we have as we see there is a high variance for most of the data points.

```
summary(lrModel)
```

```
> summary(lrModel)

Call:
lm(formula = likes_count ~ shares_count, data = mediaPosts)

Residuals:
    Min      1Q  Median      3Q     Max
-839327   -4841   -3781    -636  874541

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.409e+03  5.140e+01   105.2   <2e-16 ***
shares_count 8.963e-01  3.449e-03   259.9   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16410 on 104801 degrees of freedom
Multiple R-squared:  0.3919,    Adjusted R-squared:  0.3919
F-statistic: 6.754e+04 on 1 and 104801 DF,  p-value: < 2.2e-16
```

The above statement gives us the summary of the linear model. From the above result, we can see that R-squared value is very low. The higher the R-squared value, better the model fits. As we can see that the R-squared value is very less, this model doesn't fits well for the given data set.

3.4. Here, we need to predict the likes count given the shares count using the model we created earlier with the dataset. Below is the code for the same.

```
predict(lrModel, data.frame(shares_count=0))

predict(lrModel, data.frame(shares_count=1000))

predict(lrModel, data.frame(shares_count=10000))

predict(lrModel, data.frame(shares_count=100000))
```

```
> predict(lrModel, data.frame(shares_count=0))
       1
5408.544
> predict(lrModel, data.frame(shares_count=1000))
       1
6304.821
> predict(lrModel, data.frame(shares_count=10000))
       1
14371.32
> predict(lrModel, data.frame(shares_count=100000))
       1
95036.29
```

References

- Monash tutorial files
- https://stat.ethz.ch/R-manual/R-devel/library/base/html/strptime.html
- https://www.stat.berkeley.edu/~s133/dates.html
- https://www.rdocumentation.org/packages/EnvStats/versions/2.3.1/topics/predict.lm#targetText=Predict%20Method%20for%20Linear%20Model,a%20component%20called%20n.coefs%20.