FIT5145 Introduction to Data Science Assignment 3:

31268102

Shihan Zhang

Task A: Investigating Facebook Data using shell commands

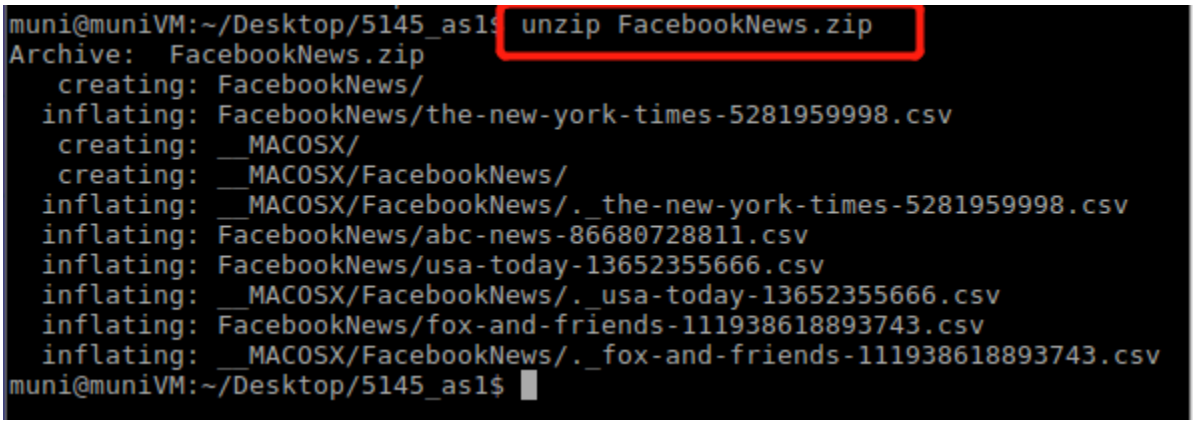
1) Decompress the file. How big is it (bytes)?

FacebookNews.zip **28384287 Bytes**

wc -c FacebookNews.zip

Decompress the file

unzip FacebookNews.zip



abc-news-86680728811.csv **26995316 Bytes**
fox-and-friends-111938618893743.csv **3390449 Bytes**
the-new-york-times-5281959998.csv **35190504 Bytes**
usa-today-13652355666.csv **28030251 Bytes**

wc -c *.csv | tail

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ wc -c  *.csv | tail
26995316 abc-news-86680728811.csv
 3390449 fox-and-friends-111938618893743.csv
35190504 the-new-york-times-5281959998.csv
28030251 usa-today-13652355666.csv
93606520 total
muni@muniVM:~/Desktop/5145_as1/FacebookNews$
```

2) What delimiter is used to separate the columns in the file and how many columns are there?
Comma is the delimiter.

abc-news-86680728811.csv **20 columns.**

head -1 abc-news-86680728811.csv | sed 's/[^,]//g' | wc -c

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ head -1 abc-news-86680728811.csv
"id","page_id","name","message","description","caption","post_type","status_type","likes_count","commen
ts_count","shares_count","love_count","wow_count","haha_count","sad_count","thankful_count","angry_coun
t","link","picture","posted_at"
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ head -1 abc-news-86680728811.csv | sed 's/[^,]//g'|wc -c
20
muni@muniVM:~/Desktop/5145_as1/FacebookNews$
```

fox-and-friends-111938618893743.csv **20 columns**

head -1 fox-and-friends-111938618893743.csv | sed 's/[^,]//g' | wc -c

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ head -1 fox-and-friends-111938618893743.csv | sed 's/[^,]/
/g'|wc -c
20
```

the-new-york-times-5281959998.csv **20 columns**

head -1 the-new-york-times-5281959998.csv | sed 's/[^,]//g' | wc -c

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ head -1 the-new-york-times-5281959998.csv | sed 's/[^,]//g
'|wc -c
20
```

usa-today-13652355666.csv **20 columns**

head -1 usa-today-13652355666.csv | sed 's/[^,]//g' | wc -c

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ head -1 usa-today-13652355666.csv | sed 's/[^,]//g'|wc -c
20
```

3) The first column is the unique identifier for each article. What are the other columns?
- The first column is the primarily for this data set.
- The other columns are the variables of the data set, but some variables can be treated as foreign key when joining operation happens (e.g. “page_id” or “post_type”)

4) How many articles are there in the file?
- 1 row contain 1 article with the first row of the file is the columns name.
- So total rows – 1 will be the number of articles is the file.
- Use *wc -l* display number of lines in the file

wc -l *.csv

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ wc -l *.csv
  43281 abc-news-86680728811.csv
   5959 fox-and-friends-111938618893743.csv
  47868 the-new-york-times-5281959998.csv
  38275 usa-today-13652355666.csv
 135383 total
muni@muniVM:~/Desktop/5145_as1/FacebookNews$
```

abc-news-86680728811.csv	**43280 articles**
fox-and-friends-111938618893743.csv	**5958 articles**
the-new-york-times-5281959998.csv	**47868 articles**
usa-today-13652355666.csv	**38275 articles**

5) What is the date range for the articles in this file? (Assume that the data is in order)
   To find the date range:
   - We use the date of the first article as the start date.
   - We use the date of the last article ad the ended date.

File Name	Start Date	End Date
abc-news-86680728811.csv	**2012-01-01**	**2016-11-07**
fox-and-friends-111938618893743.csv	**2012-01-25**	**2016-11-07**
the-new-york-times-5281959998.csv	**2012-09-08**	**2016-11-07**
usa-today-13652355666.csv	**2012-03-19**	**2016-11-07**
All 4 files together	**2012-01-01**	**2016-11-07**

Frist date of each file

awk -F'"',"' 'NR!=1{print $20}' abc-news-86680728811.csv | head -n 1

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'"',"' 'NR!=1{print $20}' abc-news-86680728811.csv | head -n 1
2012-01-01 00:30:26"
```

awk -F'"',"' 'NR!=1{print $20}' fox-and-friends-111938618893743.csv | head -n 1

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'"',"' 'NR!=1{print $20}' fox-and-friends-111938618893743.csv | head -n 1
2012-01-25 19:20:55"
```

awk -F'"',"' 'NR!=1{print $20}' the-new-york-times-5281959998.csv | head -n 1

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'"',"' 'NR!=1{print $20}' the-new-york-times-5281959998.csv | head -n 1
2012-09-08 15:16:55"
```

awk -F'"',"' 'NR!=1{print $20}' usa-today-13652355666.csv | head -n 1

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'"',"' 'NR!=1{print $20}' usa-today-13652355666.csv | head -n 1
2012-03-19 21:31:50"
```

For all 4 files together

awk -F'"',"' 'NR!=1{print $20}' *.csv | head -n 1

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'"',"' 'NR!=1{print $20}' *.csv | head -n 1
2012-01-01 00:30:26"
```

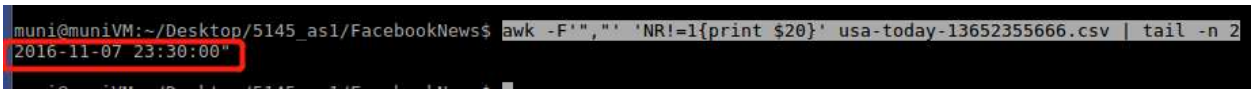
Last date of each file

awk -F'"',"' 'NR!=1{print $20}' abc-news-86680728811.csv | tail -n 1

cut -d',' -f20 fox-and-friends-111938618893743.csv | tail -n 1

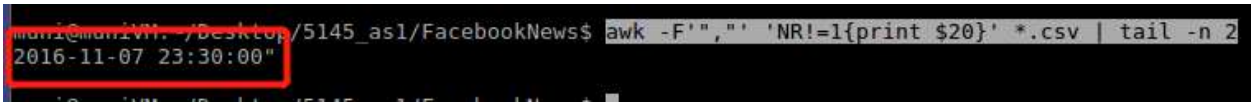
awk -F'"',"' 'NR!=1{print $20}' the-new-york-times-5281959998.csv | tail -n 1

awk -F'"',"' 'NR!=1{print $20}' usa-today-13652355666.csv | tail -n 2



For all 4 files together

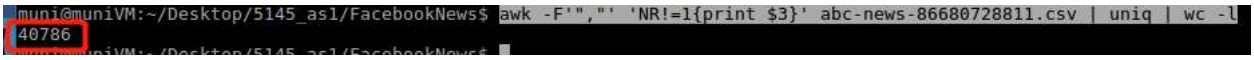
awk -F'"',"' 'NR!=1{print $20}' *.csv | tail -n 2



6) How many unique titles are there?
   - In these data file, the variable “name” represents the titles.
   - So, we need to count the total number of the unique value for column name.
   - Each line represents 1 unique title, so we use *wc -l*

abc-news-86680728811.csv **40786 unique Titles**

awk -F'"',"' 'NR!=1{print $3}' abc-news-86680728811.csv | uniq | wc -l

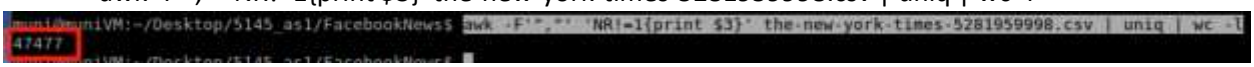


fox-and-friends-111938618893743.csv **2958 unique Titles**

awk -F'"',"' 'NR!=1{print $3}' fox-and-friends-111938618893743.csv | uniq | wc -l

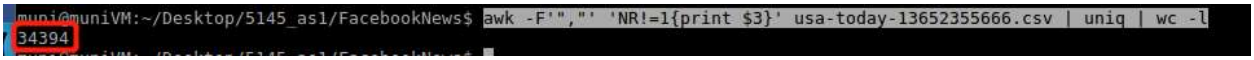
the-new-york-times-5281959998.csv **47464 unique Titles**

awk -F'"',"' 'NR!=1{print $3}' the-new-york-times-5281959998.csv | uniq | wc -l



usa-today-13652355666.csv **34297 unique Titles**

awk -F'"',"' 'NR!=1{print $3}' usa-today-13652355666.csv | uniq | wc -l



For all 4 files together **125619 unique Titles**

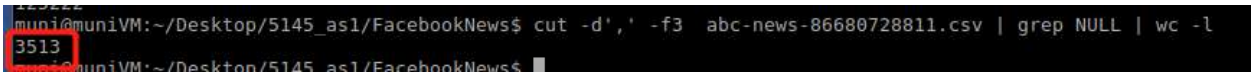
awk -F'"',"' 'NR!=1{print $3}' *.csv | uniq | wc -l

7) How many articles don't have a title?

- The article don't have a title was filled with value NULL.
- So, we need to count the total number of the article with name as "NULL".

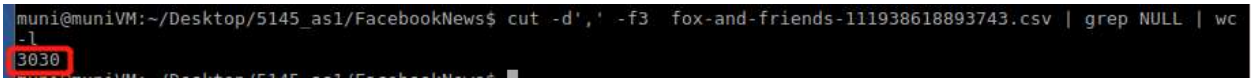
abc-news-86680728811.csv **3515 articles do not have title**

awk -F',' '$3 == "NULL" {print $3}' abc-news-86680728811.csv | wc -l



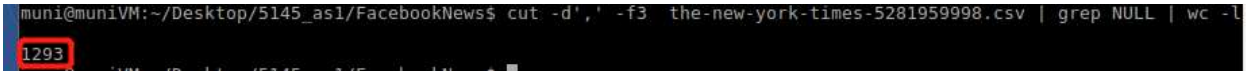
fox-and-friends-111938618893743.csv **3030 articles do not have title**

cut -d',' -f3 fox-and-friends-111938618893743.csv | grep NULL | wc -l



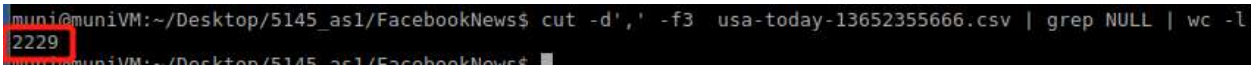
the-new-york-times-5281959998.csv **1293 articles do not have title**

cut -d',' -f3 the-new-york-times-5281959998.csv | grep NULL | wc -l



usa-today-13652355666.csv **2229 articles do not have title**

cut -d',' -f3 usa-today-13652355666.csv | grep NULL | wc -l



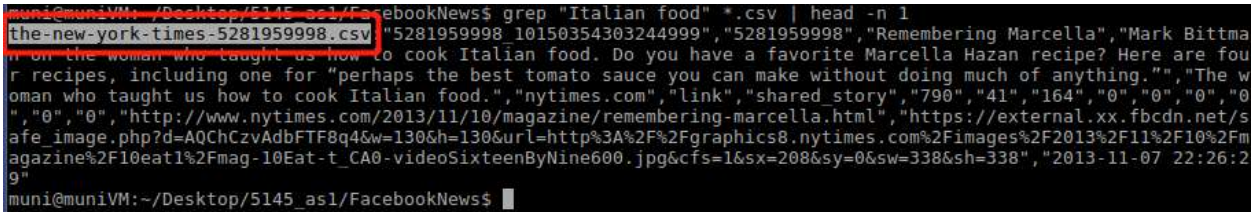
For all 4 files together **50325 articles do not have title**

cut -d',' -f3 *.csv | grep NULL | wc -l

8) When was the first mention in the files regarding "Italian food" and what was the title of the post?

The key word “Italian food” was first mention in file “the-new-york-times-5281959998.csv”. And in article id “5281959998_10150354303244999”

grep “Italian food” *.csv | head -n 1



The title of this post is “Remembering Marcella”

grep “Italian food” *.csv | cut -d’,’ -f3

9) How many times is “Hillary Clinton” mentioned in the articles? How did you find this? (Do not ignore the case)
   1. Use grep to find key word “Hillary Clinton”
   2. Use wc to count the result (-w display the number of words in the file)

For all 4 files together **162325 times**

grep "Hillary Clinton" *.csv | wc -w

10) What about “Donald Trump”? Who is the focus on more articles, Clinton or Trump? (Do not ignore the case)
   1. Find the total count of lines contain “Donald Trump”
   2. Find the total count of lines contain “Hillary Clinton”
   3. Compare the above results.

For all 4 files together **3239 articles contain “Donald Trump”**

grep "Donald Trump" *.csv | wc -l

For all 4 files together **3815 articles contain “Hillary Clinton”**

grep "Hillary Clinton" *.csv | wc -l

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ grep "Hillary Clinton" *.csv | wc -l
3815
muni@muniVM:~/Desktop/5145_as1/FacebookNews$
```

Comparing the results: 3815 > 3239, **Hillary Clinton** is focus on more articles.

11) Select the posts where “Trump” (Ignore the case) is mentioned in the post content and number of likes for those posts are greater than 100. Generate a new file with post_id and the sorted like_count and name it “trump.txt”. (In the output, you need to show the headers as well) [Hint: Find Trump in the message column, i.e., a specific column]. Then copy and paste the first 5 lines of trump.txt in your answer.
    1. Subset the column 1st ,4th & 9th column (id, message & like_count)
    2. Find all the posts include term “Trump” with like counts greater than 100
    3. Sort the post by like_count
    4. Output the result to “trump.txt”
    5. Display first 5 lines of “trump.txt “

For all 4 files together

awk -F'","' 'BEGIN{IGNORECASE = 1} $4 ~ /Trump/ && $9 > 100 {print $1,$9}' *.csv | sort -nk2 | uniq | head

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'","' 'BEGIN{IGNORECASE = 1} $4 ~ /Trump/ && $9 > 100 {print $1,$9}' *.csv | sort -nk2 | uniq | head
"86680728811_10154918482508812 101
"13652355666_10153405132230667 102
"13652355666_1705070773150777 102
"5281959998_10150850571004999 102
"86680728811_10154597155953812 102
"5281959998_10150797766644999 103
```

- Write header to “trump.txt”

  awk -F',' '{print $1,$9}' *.csv | head -n 1 > trump.txt

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F',' '{print $1,$9}' *.csv | head -n 1 > trump.txt
```

- Write post_id and like_count to “trump.txt”

awk -F'","' 'BEGIN{IGNORECASE = 1} $4 ~ /Trump/ && $9 > 100 {print $1,$9}' *.csv | sort -nk2 | uniq >> trump.txt

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ awk -F'","' 'BEGIN{IGNORECASE = 1} $4 ~ /Trump/ && $9 > 100 {print $1,$9}' *.csv | sort -nk2 | uniq >> trump.txt
```

- verify the file (display the first 5 lines)

"id" "likes_count"

"86680728811_10154918482508812 101

"13652355666_10153405132230667 102

"13652355666_1705070773150777 102

"5281959998_10150850571004999 102

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ cat trump.txt | head -n 5
"id" "likes_count"
"86680728811_10154918482508812 101
"13652355666_10153405132230667 102
"13652355666_1705070773150777 102
"5281959998_10150850571004999 102
```

12) Find the total number of love_count and angry_count for “Donald Trump” and “Hillary Clinton” separately. Who has more positive feeling among people? Justify your answer.

love_count for “Donald Trump” **288432**

grep -w "Donald Trump" *.csv | awk -F'","' '{sum += $12} END {print sum}'

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ grep -w "Donald Trump" *.csv | awk -F'","' '{sum += $12} END {print sum}'
288432
muni@muniVM:~/Desktop/5145_as1/FacebookNews$
```

love_count for “Hillary Clinton” **639772**

grep -w "Hillary Clinton" *.csv | awk -F'","' '{sum += $12} END {print sum}'

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ grep -w "Hillary Clinton" *.csv | awk -F'","' '{sum += $12} END {print sum}'
639772
```

angry_count for “Donald Trump” **1094288**

grep -w "Donald Trump" *.csv | awk -F'","' '{sum += $17} END {print sum}'

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ grep -w "Donald Trump" *.csv | awk -F'","' '{sum += $17} END {print sum}'
1094288
```

angry_count for “Hillary Clinton” **1557020**

awk -F'","' '$4 ~ /Hillary Clinton/ {sum += $17} END {print sum}' *.csv

```
muni@muniVM:~/Desktop/5145_as1/FacebookNews$ grep -w "Hillary Clinton" *.csv | awk -F'","' '{sum += $17} END {print sum}'
1557020
```

Who has more positive feeling among people?

To answer this question, we can set the “feeling_index = love_count / angry_count”

Larger index number means more positive feeling

For Trump: 288432 / 1094288 ≈ 0.26

For Clinton: 639772 / 1557020 ≈ 0.41

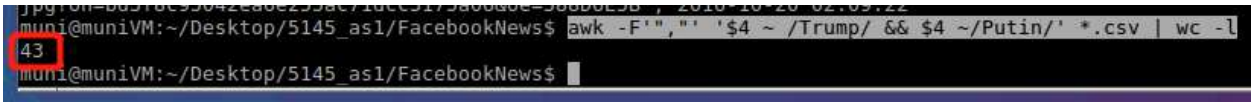
0.41 < 0.26

Therefore, Hillary Clinton has more positive feelings among people.

13) How many articles discussed Trump and Putin? How many discussed Trump but not Clinton?
    1. find article (column message includes term Trump and Putin)
    2. count the lines as 1 line represent 1 article
    3. find article (column message includes term Trump but not include Clinton)
    4. count the lines as 1 line represent 1 article

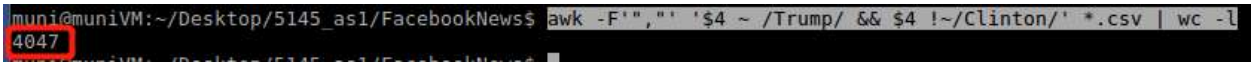
Discussed Trump and Putin **43 articles**

awk -F'","' '$4 ~ /Trump/ && $4 ~/Putin/' *.csv | wc -l



Discussed Trump but not Clinton **4047 articles**

awk -F'","' '$4 ~ /Trump/ && $4 !~/Clinton/' *.csv | wc -l



14） For each publication in trump.txt, find out which month had the most articles about Trump. Try to do this without using grep.

- Before we do this question, I need to reformate the trump.txt with delimiter "," and save to new file 't.txt'

  sed 's/ /","/g' trump.txt > t.txt

  cat t.txt | head

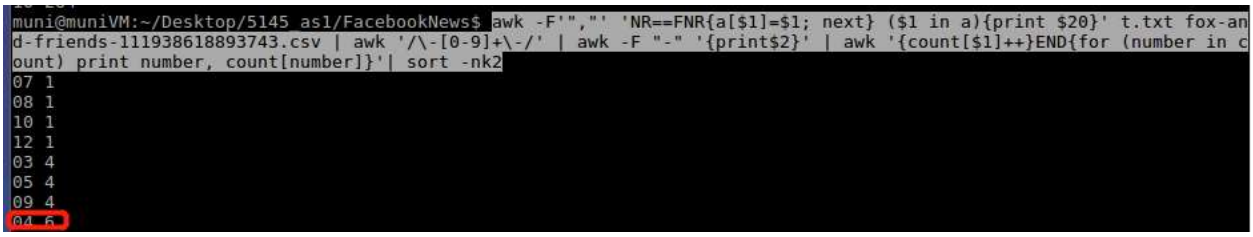
- The we use the following code to find the `id` records from 't.txt' pair with each publication and use regex to capture the month value. Finally, we use for loop to count the articles which publish in that month.

awk -F'"',"' 'NR==FNR{a[$1]=$1; next} ($1 in a){print $20}' t.txt abc-news-86680728811.csv | awk '/\-[0-9]+\-/' | awk -F "-" '{print$2}' | awk '{count[$1]++}END{for (number in count) print number, count[number]}'| sort -nk2

abc-news-86680728811.csv **largest = 264, Month = 10**

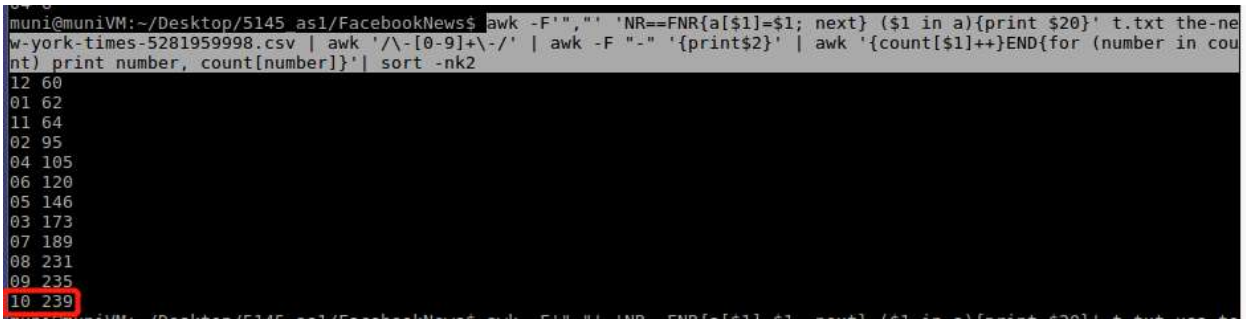
awk -F'"',"' 'NR==FNR{a[$1]=$1; next} ($1 in a){print $20}' t.txt fox-and-friends-111938618893743.csv | awk '/\-[0-9]+\-/' | awk -F "-" '{print$2}' | awk '{count[$1]++}END{for (number in count) print number, count[number]}'| sort -nk2

fox-and-friends-111938618893743.csv **largest = 6, Month = 04**



awk -F'"',"' 'NR==FNR{a[$1]=$1; next} ($1 in a){print $20}' t.txt the-new-york-times-5281959998.csv | awk '/\-[0-9]+\-/' | awk -F "-" '{print$2}' | awk '{count[$1]++}END{for (number in count) print number, count[number]}'| sort -nk2

the-new-york-times-5281959998.csv **largest = 239, Month = 10**



awk -F'","' 'NR==FNR{a[$1]=$1; next} ($1 in a){print $20}' t.txt usa-today-13652355666.csv | awk '/\-[0-9]+\-/' | awk -F "-" '{print$2}' | awk '{count[$1]++}END{for (number in count) print number, count[number]}'| sort -nk2

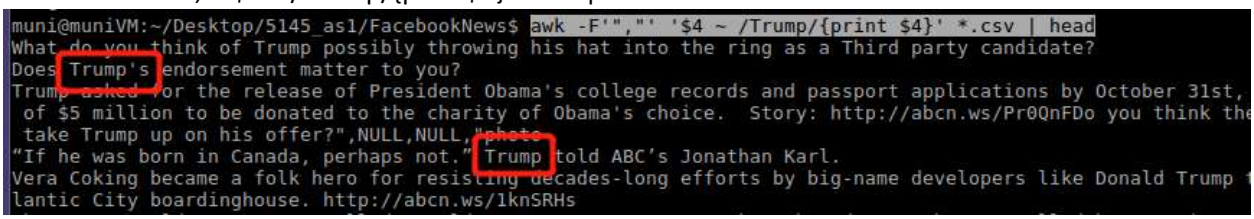
usa-today-13652355666.csv **largest = 185, Month = 10**

Task B: Graphing the Data in R

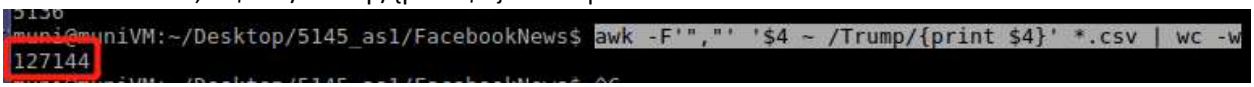
1) How many times does the term 'Trump' appear in the post message? (use Unix shell to answer to this question)
    1. subset to the target 4th column("messages")
    2. Find how many times term "Trump" appears (use grep to find term and count the result with wc -w)

For all 4 files together

- Subset to target column("message") and find term "Trump"
  awk -F'"',"' '$4 ~ /Trump/{print $4}' *.csv | head



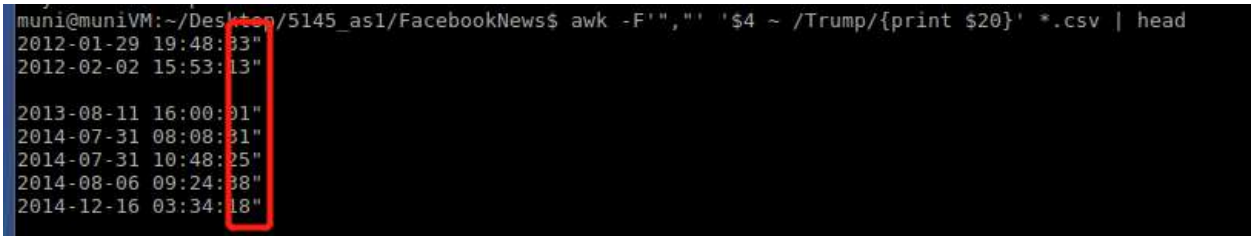
- count the result **"Trump" 127144 times**
  awk -F'"',"' '$4 ~ /Trump/{print $4}' *.csv | wc -wl



2) We want to consider how the amount of discussion regarding Donald Trump varies over the time period covered by the data file. To answer this question, you will need to extract the timestamps for all posts referring to Trump using shell.
    1. Find all posts (4th column) that has term "Trump" appears.
    2. Extract the data from 20th column from step 1.
    3. Save the timestamp data column as csv file.

For all 4 files together

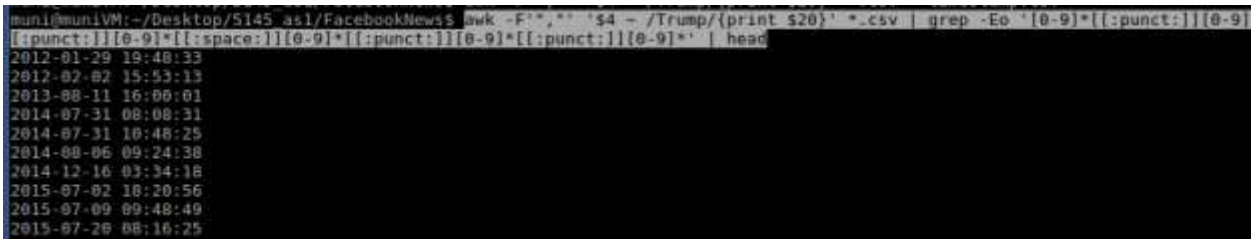
- **Find the date data by using regex (message contain term "Trump")**
  awk -F'"',"' '$4 ~ /Trump/{print $20}' *.csv | head



The returned result has a single quotation mark, which we don't want.

But we can use *grep()* with Regex to remove the quotation marks.

awk -F'"',"' '$4 ~ /Trump/{print $20}' *.csv | grep -Eo '[0-9]*[[:punct:]][0-9]*[[:punct:]][0-9]*[[:space:]][0-9]*[[:punct:]][0-9]*[[:punct:]][0-9]*' | head



- **Save/write the date data to output file(timestamp.csv)**

Write header first

- The header “posted_at” contain double quotation marks can not work really well with *read_csv()* in R.
- Therefore, I decided to use regex to capture only the text posted_at.

cut -d',' -f20 *.csv | grep -Eo '[a-z]{6}[[:punct:]][a-z]{2}' | head -n 1 > timestamp.csv

Write the timestamp data

awk -F'"',"' '$4 ~ /Trump/{print $20}' *.csv | grep -Eo '[0-9]*[[:punct:]][0-9]*[[:punct:]][0-9]*[[:space:]][0-9]*[[:punct:]][0-9]*[[:punct:]][0-9]*' >> timestamp.csv

- **Verify the output file**
  cat timestamp.csv | head -n 5

Read csv file in R studio (R code)

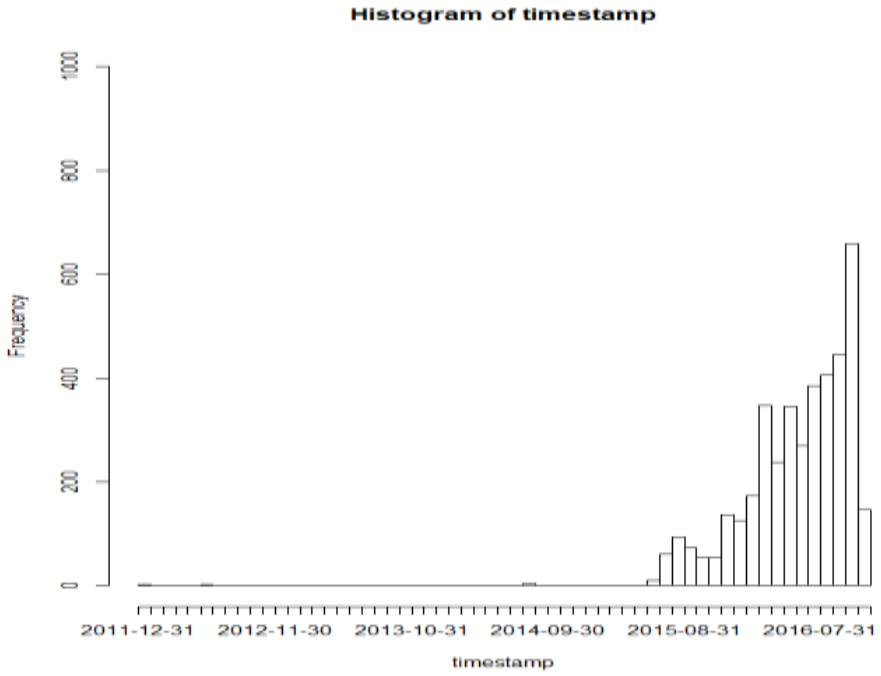
```
# load timestamp data

timestamp <- read.csv("C:/Users/SENMS/Desktop/tunnel/timestamp.txt")

timestamp <- strptime(timestamp$posted_at, format = "%Y-%m-%d %H:%M:%S", tz ="")
```

generate histogram

hist(x = timestamp, breaks = "month", freq = TRUE, ylim = range(0:1000))



Describe the pattern:

This is a left-skewed distribution. There was almost no frequency before 2015, and the frequency has increased explosively after 2015. As we know, this data is a record of Trump's media reports. From 2015 to 2016, he entered the public eye as a presidential candidate, which increased the number of reports. The presidential election at the end of 2016 brought the data to its peak. After the election, the number of media reports began to decline. Combined with real events, this histogram looks very reasonable.

3) In this question, we want to investigate the Facebook posts of a few top media sources. To answer this question, you will need to extract the Facebook posts made on the pages of "abc-news", "cnn" and "fox-news" from your original Facebook dataset.

    1. Use the Unix shell to first generate a file containing all the records belonging to "abc-news", "cnn" and "fox-news" only. Then read the resulting file in R.

awk -F'"',"' ' {print $0}' *.csv | head -n 1 > BQ3.txt

awk -F'"',"' '$18~/abcn/ || $18~/cnn/ || $18~/foxnews/ {print $0}' *.csv >> BQ3.txt

Save the output to 'BQ3.txt' file

```
muni@muniVM:~/Desktop/5145 as1/FacebookNews$ awk -F'","' '$18~/abcn/ || $18~/cnn
/ || $18~/foxnews/ {print $0}' *.csv > BQ3.txt
```

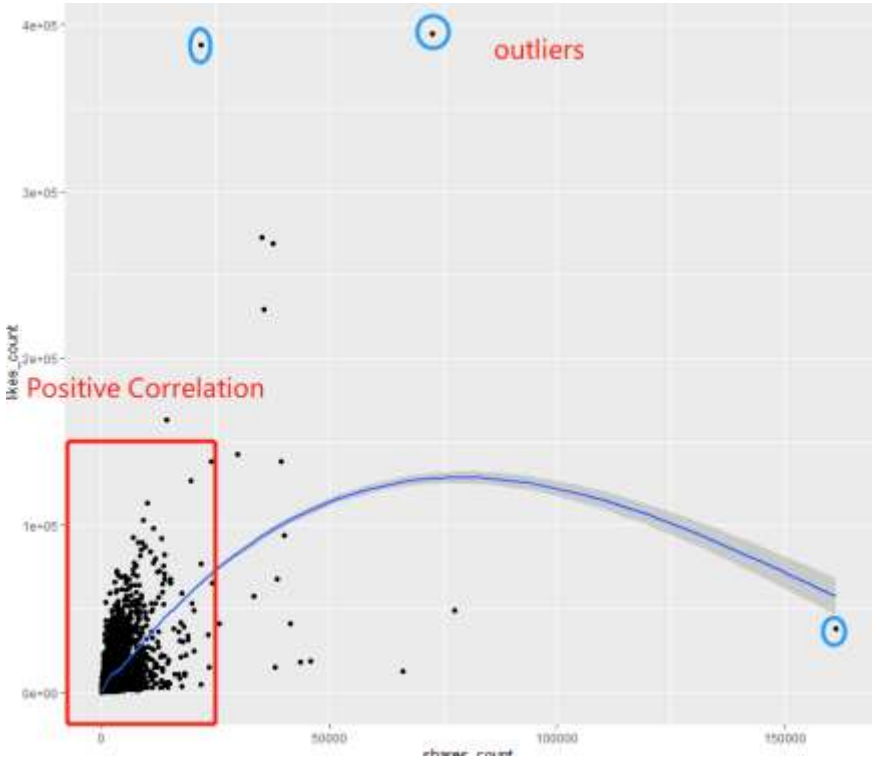
Read file in R

BQ3 <- read_csv("C:/Users/SENMS/Desktop/tunnel/BQ3.txt")

```
> str(BQ3)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':        25656 obs. of  20 variables:
 $ id              : chr  "<U+FEFF>\"86680728811_272953252761568\"" "<U+FEFF>\"86680728811_273859942672742\"" "<
811_10150499874478812\"" "<U+FEFF>\"86680728811_252342804833247\"" ...
 $ page_id         : num  8.67e+10 8.67e+10 8.67e+10 8.67e+10 8.67e+10 ...
 $ name            : chr  "Chief Justice Roberts Responds to Judicial Ethics Critics" "With Reservations, Obama
w Detention of Citizens" "Wishes For 2012 to Fall on Times Square" "NY Pharmacy Shootout Leaves Suspect, ATF A
 $ message         : chr  "Roberts took the unusual step of devoting the majority of  his annual  report to the
 ethics." "Do you agree with the new law?" "Some pretty cool confetti will rain down on New York City celebrat
cy was held up by a man seeking prescription medication." ...
```

2. Use appropriate R code to generate a plot showing the relationship between the number of shares and the number of likes in your dataset. Do you see any relationship?

   ggplot(BQ3, aes(shares_count, likes_count)) + geom_point() + geom_smooth()

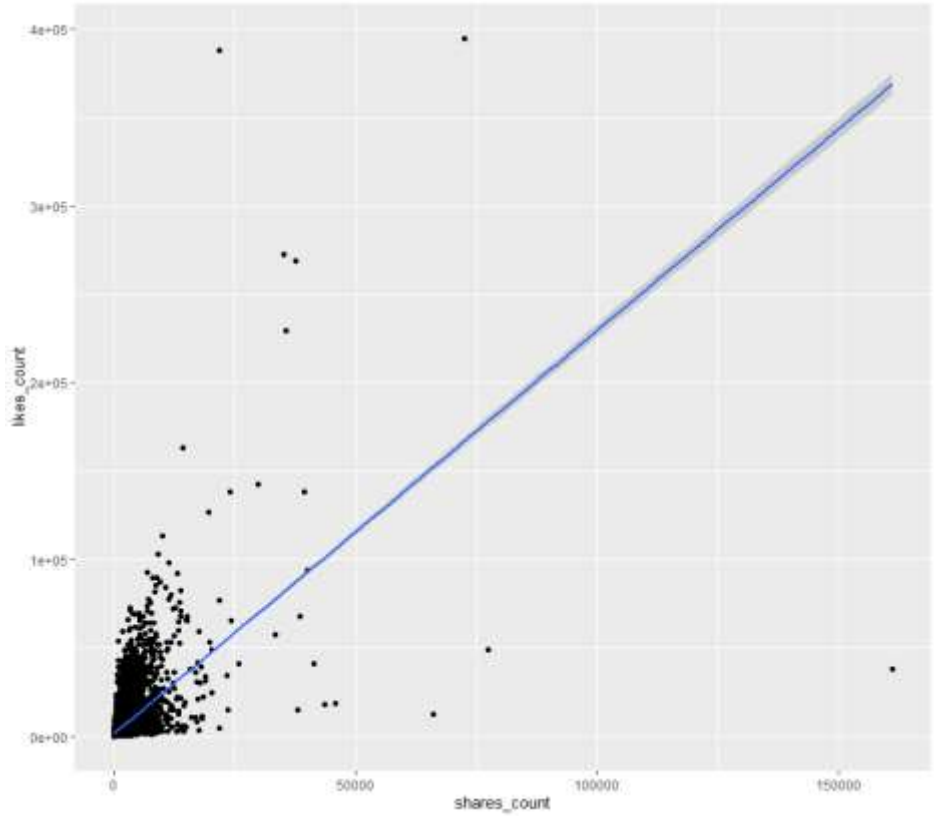


   Between 0-2500 shares_count, we see a positive correlation. Likes will increase when sharing increases. When share_count is greater than 2500, it is difficult for us to describe their relevance, which looks more like no correlation. We can also find some outliers which indicate with blue circles.

3. Fit a linear regression model using R to the above data (i.e., shares_count an likes_count) and plot the linear fit. Does it look like a good fit to you?

- Use ggplot2 linear regression model (lm)

ggplot(BQ3, aes(shares_count, likes_count)) + geom_point() + geom_smooth(method = lm)

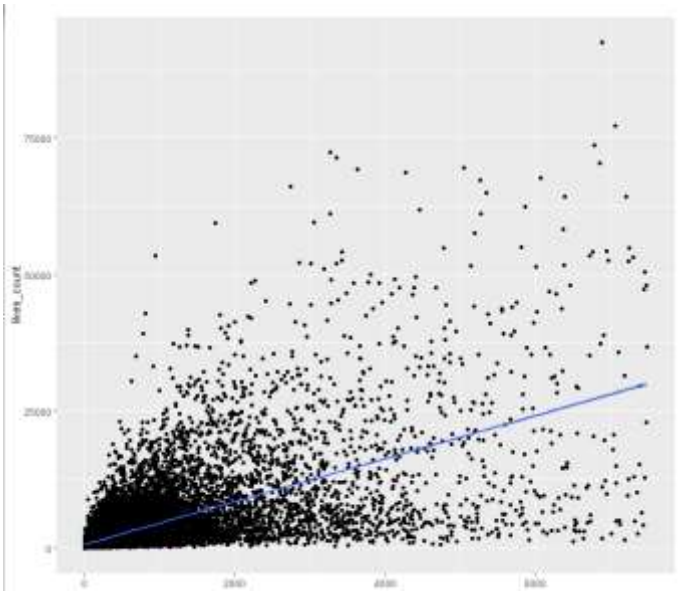


Our data span is very large and contains some outliers, which makes this linear regression model seem to be unable to fit well.

If we narrow the scope to shares_count 0-7500.

lean_BQ3 <- subset(BQ3, shares_count < 7500)

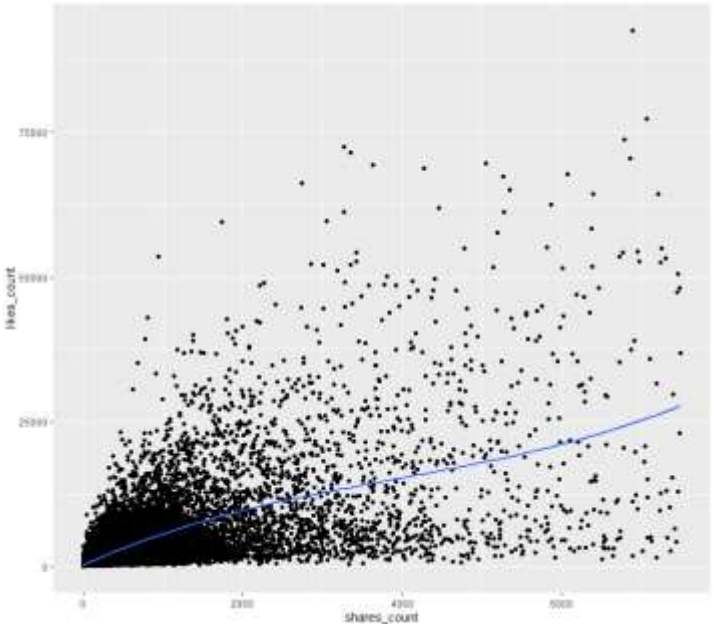
ggplot(lean_BQ3, aes(shares_count, likes_count)) + geom_point() + geom_smooth(method = lm)



Now, this model looks better than the previous one, but it is still not good enough.

What if we try polynomial linear regression?

```
ggplot(lean_BQ3, aes(shares_count, likes_count)) + geom_point()  +
stat_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, 3, raw=TRUE))
```



This fit has not improved significantly. It seems because of that the correlation between these two data is not strong enough.

4. Use the linear fit to predict the number of likes a post will generate if it is shared 0 times, 100 times, 1000 times, 10000 times and 100000 times on Facebook.

Create a linear regression model in R:
linear_model = lm(formula = BQ3$likes_count ~ BQ3$shares_count)

[1] **Create a new data frame for prediction**
Share_for_predict <- data.frame(c(0,100,1000,10000,100000))

Prediction:
predict(linear_model, newdata = Share_for_predict)

```
> predict(linear_model, newdata = Share_for_predict)
       1        2        3        4        5        6
1574.555 1936.858 1547.212 1624.685 2527.025 1547.212 1
```

Shares	Likes
0	1575
100	1937
1000	1547
10000	1625
100000	2527

Reference:

[1] R predict() function

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/predict.lm.html