FIT1043 Introduction to Data Science

Assignment 3

Rivaldi Purnama 32633025

Part A

1. A1

```
cd downloads
```

To change the working directory to downloads, because the file is there.

```
gunzip fb_dataset.gz
```

Unzipping the file with the gunzip command to expand the file.

```
ls -lh fb_dataset
```

Checking the file size with the lh command and with ls to display.

```
rival@DESKTOP-AEP9GMG MINGW64 ~/downloads
$ ls -lh fb_dataset
-rw-r--r-- 1 rival 197609 344M May 20 11:01 fb_dataset
```

The size is 344 MB

2. A2

```
head -2 fb_dataset
```

```
$ head -2 fb_dataset
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
abc-news,86680728811_272953252761568,86680728811,Chief Justice Roberts Responds
to Judicial Ethics Critics,Roberts took the unusual step of devoting the majorit
y of  his annual  report to the issue of judicial ethics.,PAUL J. RICHARDS/AFP/G
etty Images Chief Justice John Roberts issued a ringing endorsement Saturday nig
ht of his colleagues_ ability to determine when they should step down from a cas
e because of a conflict of interest. _I have complete confidence in the capabili
ty of my colleagues to determine when ...,abcnews.go.com,link,shared_story,61,27
,12,0,0,0,0,0,0,http://abcnews.go.com/blogs/headlines/2011/12/chief-justice-robe
rts-responds-to-judicial-ethics-critics/,https://external.xx.fbcdn.net/safe_imag
e.php?d=AQAPXteeHLT2K7Rb&w=130&h=130&url=http%3A%2F%2Fabcnews.go.com%2Fimages%2F
Politics%2Fgty_chief_justice_john_roberts_jt_111231_wblog.jpg&cfs=1&sx=108&sy=0&
sw=269&sh=269,1/1/12 0:30
```

This is to get the header and first 1 row through all columns. In the output, the columns seem to be separated by a coma.

So the delimiter is a “,”.

```
wc -l fb_dataset
```

```
rival@DESKTOP-AEP9GMG MINGW64 ~/downloads
$ wc -l fb_dataset
533940 fb_dataset
```

This is to get the number of lines (in this case, lines = rows).

So, the number of rows is 533940

3. A3

```
head -1 fb_dataset
```

```
rival@DESKTOP-AEP9GMG MINGW64 ~/downloads
$ head -1 fb_dataset
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
```

The header is the column and we can get it with the head command and put -1 to indicate the first row, the header. **There are 21 columns:**

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. A4

The unique pages is the page_id from the page_id column. Different page_name will have different page_id.

```
awk -F ',' '{print $3}' fb_dataset | uniq
```

```
rival@DESKTOP-AEP9GMG MINGW64 ~/downloads
$ awk -F ',' '{print $3}' fb_dataset | uniq
page_id
86680728811
2.29E+11
1.31E+11
5550296508
1.12E+14
15704546335
1.56E+14
10643211755
18468761129
5863113009
5281959998
8304333127
6250307292
10606591490
13652355666
```

The [awk -F ',' '{print $3}' fb_dataset] command allows us to go to the third column (page_id). It is then piped into the next part where we use a [uniq] command to get the unique values.

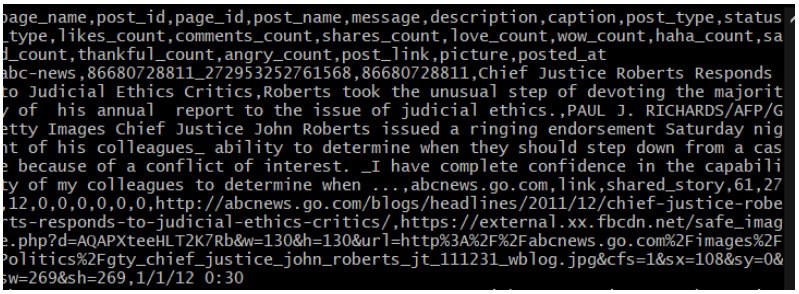
So, there are 15 unique pages.

5. **A5**

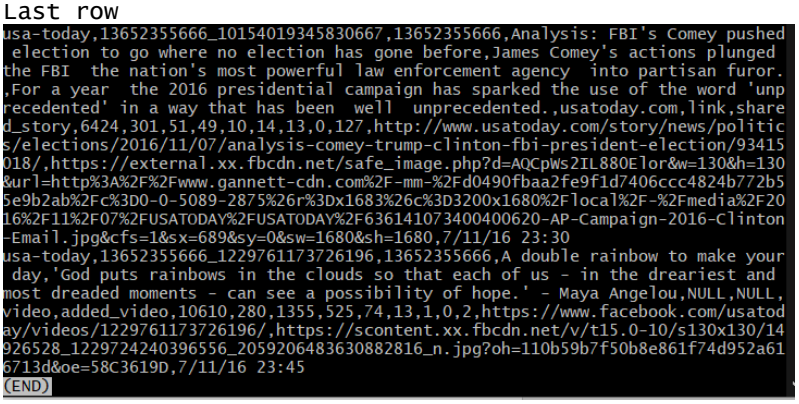
```
cat fb_dataset | less
```

The cat command is to load the file and output it to the terminal, so we output the fb_dataset. The less command is to display the content one screen at a time. Then, from the beginning look through to the end with shift+G.

First row



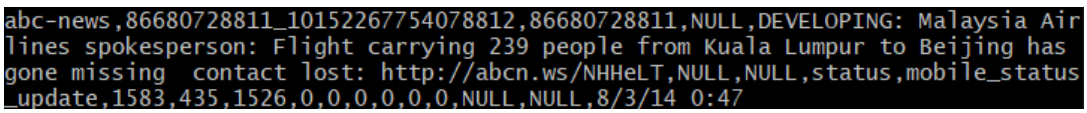
Last row



So, the date ranges from 1/1/12 to 7/11/16.

6. **A6**

```
grep "Malaysia Airlines" fb_dataset | less
```



The grep command is to search for a string. The piping to less means that we are to view the contents that contains “Malaysia Airlines” one page at a time.

So,

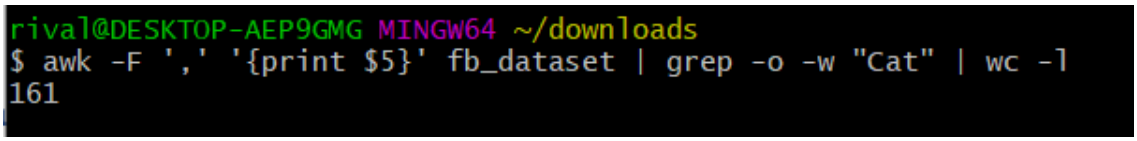
First occurrence: 8/3/14

Message: DEVELOPING: Malaysia Airlines spokesperson: Flight carrying 239 people from Kuala Lumpur to Beijing has gone missing contact lost: http://abcn.ws/NHHeLT

First media: abc-news

7. A7

```
awk -F ',' '{print $5}' fb_dataset | grep -o –w "Cat" | wc –l
```



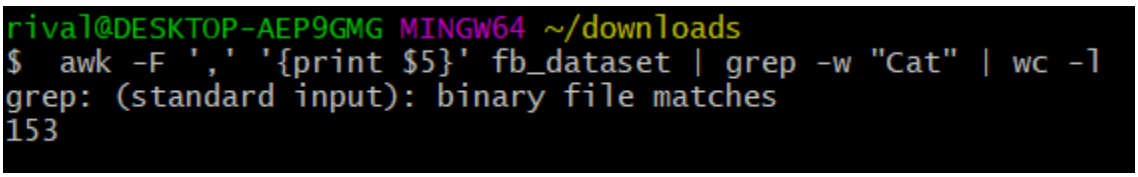
The [awk –F ‘,’ ‘{print $5}’ fb_dataset] is to get the message column in the fb_dataset. It is then piped to The [grep -o –w “Cat”] command which is to get the lines that has the string “Cat” and setting them into different individual lines. The [wc –l] command then counts the number of lines, meaning the number of “Cat” that is piped.

There are 161 occurences of “Cat”

8. A8

I define popularity by how many entries of message contains “Cat” and how many contains “Dog”, not how many times each of them appear overall.

```
awk -F ',' '{print $5}' fb_dataset | grep -w "Cat" | wc –l
```



The [awk –F ‘,’ ‘{print $5}’ fb_dataset] is to get the message column in the fb_dataset. It is then piped to The [grep –w “Cat”] command to get the rows that contains “Cat”, and it will be piped to [wc –l] function to count how many message that contains “Cat”.

There are 153

```
awk -F ',' '{print $5}' fb_dataset | grep -w "Dog" | wc -l
```

```
rival@DESKTOP-AEP9GMG MINGW64 ~/downloads
$ awk -F ',' '{print $5}' fb_dataset | grep -w "Dog" | wc -l
303
```

The second part is to change the “Cat” into “Dog” to search for the messages containing “Dog”.

There are 303.

There are mode “Dog” than “Cat”, which means “Dog” is more popular.

So, dog is more popular on Facebook.

9. A9

```
cat fb_dataset | cut -d ',' -f 2,5,10 | grep --ignore-case -w "Cat" | sort -k 3
-t ',' -n | awk -F ',' '$3>=1000 {print $1,$3}' > cat.txt
```

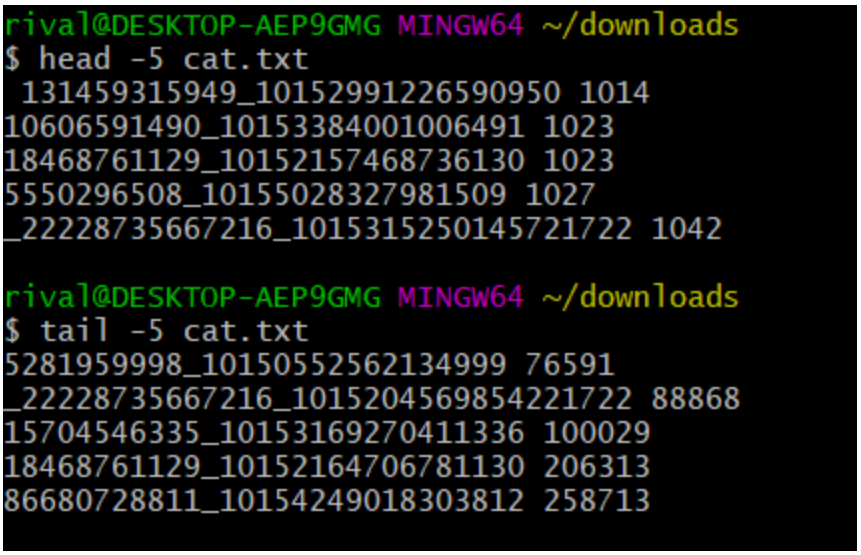
We first read fb_dataset with [cat fb_dataset] command. Then is is piped to [cut -d ',' -f 2,5,10] command to cut the dataframe into 3 columns. It is then piped to [grep --ignore-case -w "Cat"] command to get the rows “cat” and/or “Cat”. It is then piped [sort -k 3 -t ',' -n] command to sort according to like_count which is the third column. Then it is piped to [awk -F ',' '$3>=1000 {print $1,$3}' > cat.txt] command to only print the ones that have like_count of 1000 or larger, it will be printed as post_id,like_count and it will then be extracted to a file called cat.txt.

```
head -5 cat.txt
```

This is to get the first 5 rows.

```
tail -5 cat.txt
```

This is to get the last 5 rows.



10. A10

We first get the love_count and the hate_count from the cat posts into cat_count.txt file. Then with the new file, we count them.

We do the same for the dog one.

I score the likeness by subtracting each of the love_count total by love_count + hate_count (which means total love-count – total reactions) total of each animal and then comparing them.

```
cat fb_dataset | grep --ignore-case -w "Cat" | cut -d ',' -f 13,18 | awk -F ',' '{print $1,$2}' > cat_count.txt
```

The [cat fb_dataset] command is to load the dataset. It is then piped to [grep --ignore-case -w "Cat"] command which is to get online the rows that contains “Cat” and/or “cat”. It is then piped to [cut -d ',' -f 13,18 | awk -F ',' '{print $1,$2}' > cat_count.txt] command which is to get the love_count and the hate_count for cats and extract it to a new file called cat_count.txt.

```
cat cat_count.txt | awk '{sum+=$1} END {print sum}'
```

The [cat cat_count.txt] command is to load the cat_count dataset, which is then piped to the [awk '{sum+=$1} END {print sum}'] that functions to add the numbers in the love_count, the first column.

There are 214346 love_count for cats

```
cat cat_count.txt | awk '{sum+=$2} END {print sum}'
```

The [cat cat_count.txt] command is to load the cat_count dataset, which is then piped to the [awk '{sum+=21} END {print sum}'] that functions to add the numbers in the hate_count, the second column.

There are 23855 hate_count for cats

Positive feeling = 214346 / 214346 + 23855 = 0.989

The commands for the dog_count and counting the total love_count and hate_count is the same as cat, just that we are replacing the "Cat" with "Dog".

```
cat fb_dataset | grep --ignore-case -w "Dog" | cut -d ',' -f 13,18 | awk -F ','
'{print $1,$2}' > dog_count.txt
```

```
cat dog_count.txt | awk '{sum+=$1} END {print sum}'
```

There are 1339606 love_count for dogs

```
cat dog_count.txt | awk '{sum+=$2} END {print sum}'
```

There are 195954 hate_count for dogs

Positive feeling = 1339606 / 1339606 + 195954 = 0.872

Cat positive feeling = 98.9%

Dog positive feeling = 87.2%

So cats receive more positive feeling from people that reacted on Facebook.

Part B

1. ## B1

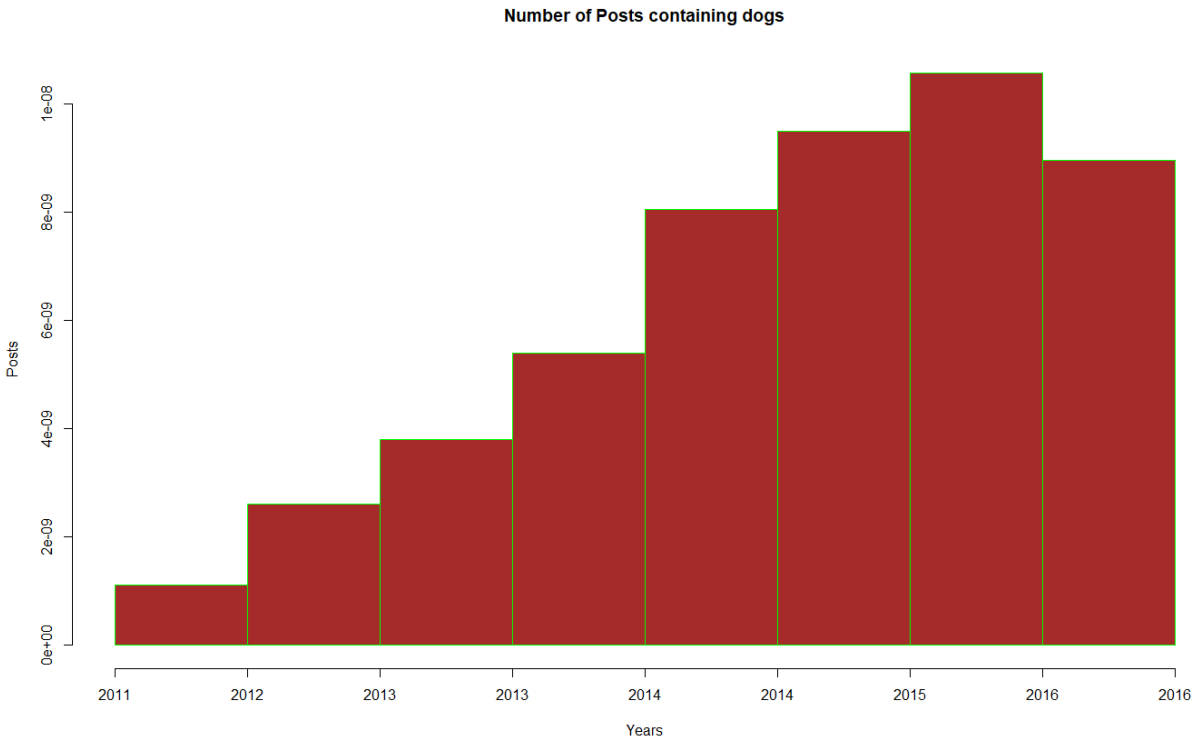
We first extract the posted_at column

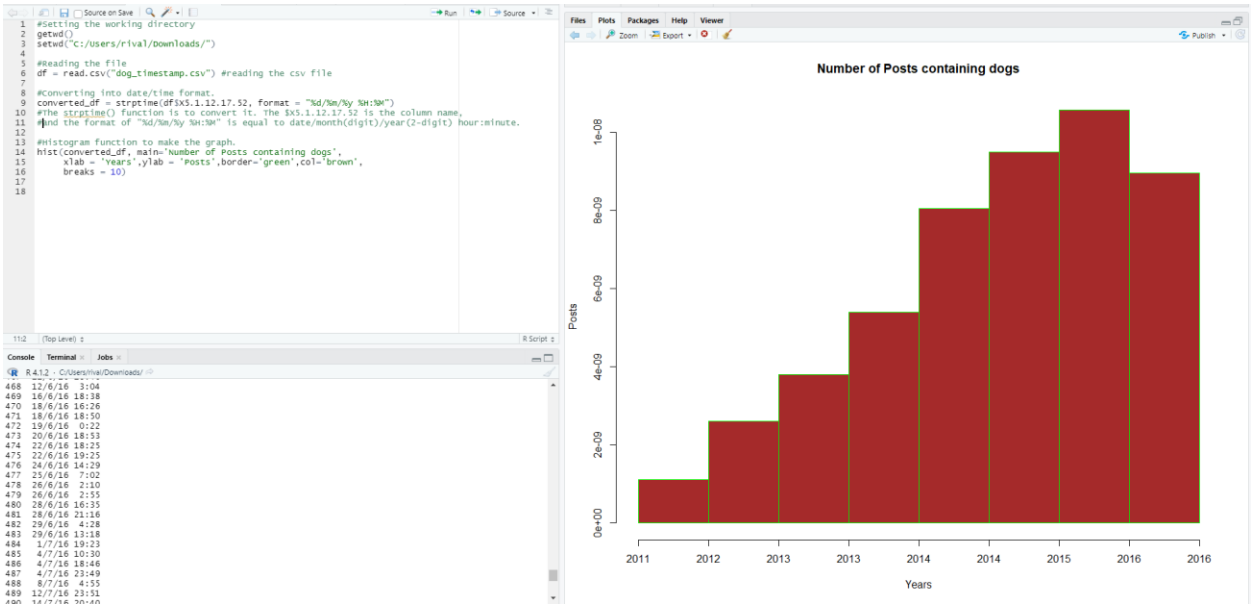
```
cat fb_dataset | grep --ignore-case -w "Dog" | awk -F ',' '{print $21}' >
dog_timestamp.csv
```

The [cat fb_dataset] command is to load the data, then piped to [grep –ignore-case –w “Dog”] command to get the rows that contain “Dog” and/or “dog”. It is then piped to [awk -F ',' '{print $21}' > dog_timestamp.csv] command to save the posted_at column into a new csv file which is dog_timestamp.csv.

From RStudio:

```
#Setting the working directory

getwd()

setwd("C:/Users/rival/Downloads/")


#Reading the file

df = read.csv("dog_timestamp.csv") #reading the csv file

#Converting into date/time format.

converted_df = strptime(df$X5.1.12.17.52, format = "%d/%m/%y %H:%M")

#The strptime() function is to convert it. The $X5.1.12.17.52 is the column name, and the format of "%d/%m/%y %H:%M" is equal to date/month(digit)/year(2-digit) hour:minute.


#Histogram function to make the graph.

hist(converted_df, main='Number of Posts containing dogs',

    xlab = 'Years',ylab = 'Posts',border='green',col='brown',

    breaks = 10)
```





2. B2

We can first download the required data onto a new csv file.

```
cat fb_dataset | cut -d ',' -f 1,8,11 | grep -w "abc-news" | awk -F ','
'$2=="event" || $2=="link" || $2=="photo" || $2=="status" || $2=="video" {print
$2,$3}' > post_engagement.txt
```

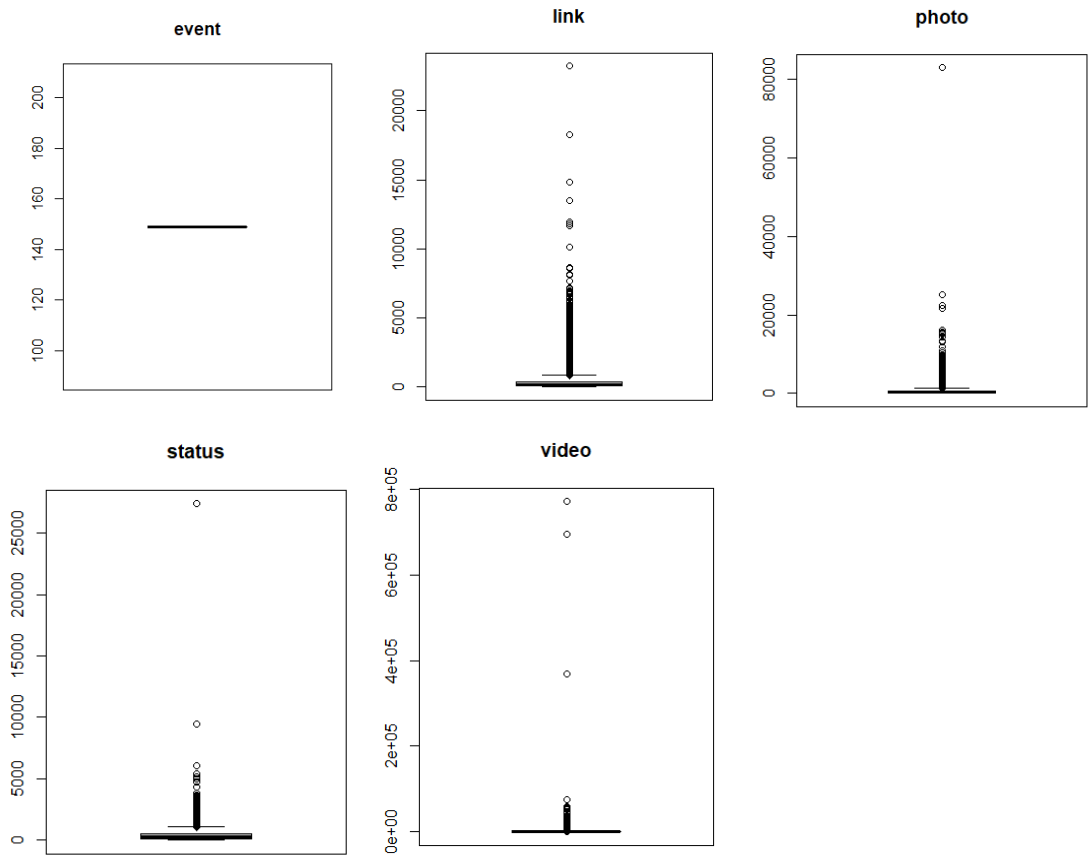
The [cat fb_dataset] command is to load the original dataset, it is then piped to [cut -d ',' -f 1,8,11] to be cut into 3 columns, page_name, post_type, and comments_count.
It is then piped to [awk -F ',' '$2=="event" || $2=="link" || $2=="photo" || $2=="status" || $2=="video" {print $2,$3}' > post_engagement.csv] command to extract only the rows that the post_type is event/link/photo/status/video to post_engagement.txt.

i.

From RStudio:

```
df2 = read.table("post_engagement.txt")

for (v1 in c("event","link","photo","status","video")){
  df2_new = df2

  df2_new = df2_new %>% filter(V1 == v1)

  df2_new[,2]%>%boxplot(main = v1)
}
```

Using a for-loop to loop through all the post_type which is V1. Then using a boxplot, making a graph for each of the types.



It seems that “video” is the most engaging type.

ii.

From RStudio:

```
for (v1 in c("link","photo","status","video")){
  df2_new = df2

  df2_new = df2_new %>% filter(V1 == v1)%>%filter(V2 > 1000)

  df2_new[,2]%>%boxplot(main = v1)

}
```

We only exclude “event” because we know that there is only 1 entry and the comments is even less than 1000 which means there is 0. So, we take the other 4 types and then filter the number of commands with filter() from dplyr library.