Project Title:

YouTube Channel Data

Submitted by:

Smit Shiroya

Faculty Mentor:

Dr. Shilpa Balan

Submitted to:

California State University, Los Angeles

A. Data Set Description

There is a case to be made for information to be the most valuable commodity in the society we live in today. The prevalence of high-speed communication channels that allow anyone to interact with anyone else in the world instantly eliminates barriers to the free exchange of information. Naturally, this requires the development of innovative technologies that support these interactions and allow for them to be persistent. With the number of people alive and the variety of media for them to communicate through, billions of petabytes of data are bound to be generated daily.

As network speeds improve, the ability to consume information in rich-media formats continues to increase. When combined with the rigidity of orthodox mediums like the television and movies compared to the attention-optimal mechanics of social media, it is inevitable that video formats optimize towards data consumption and generation.

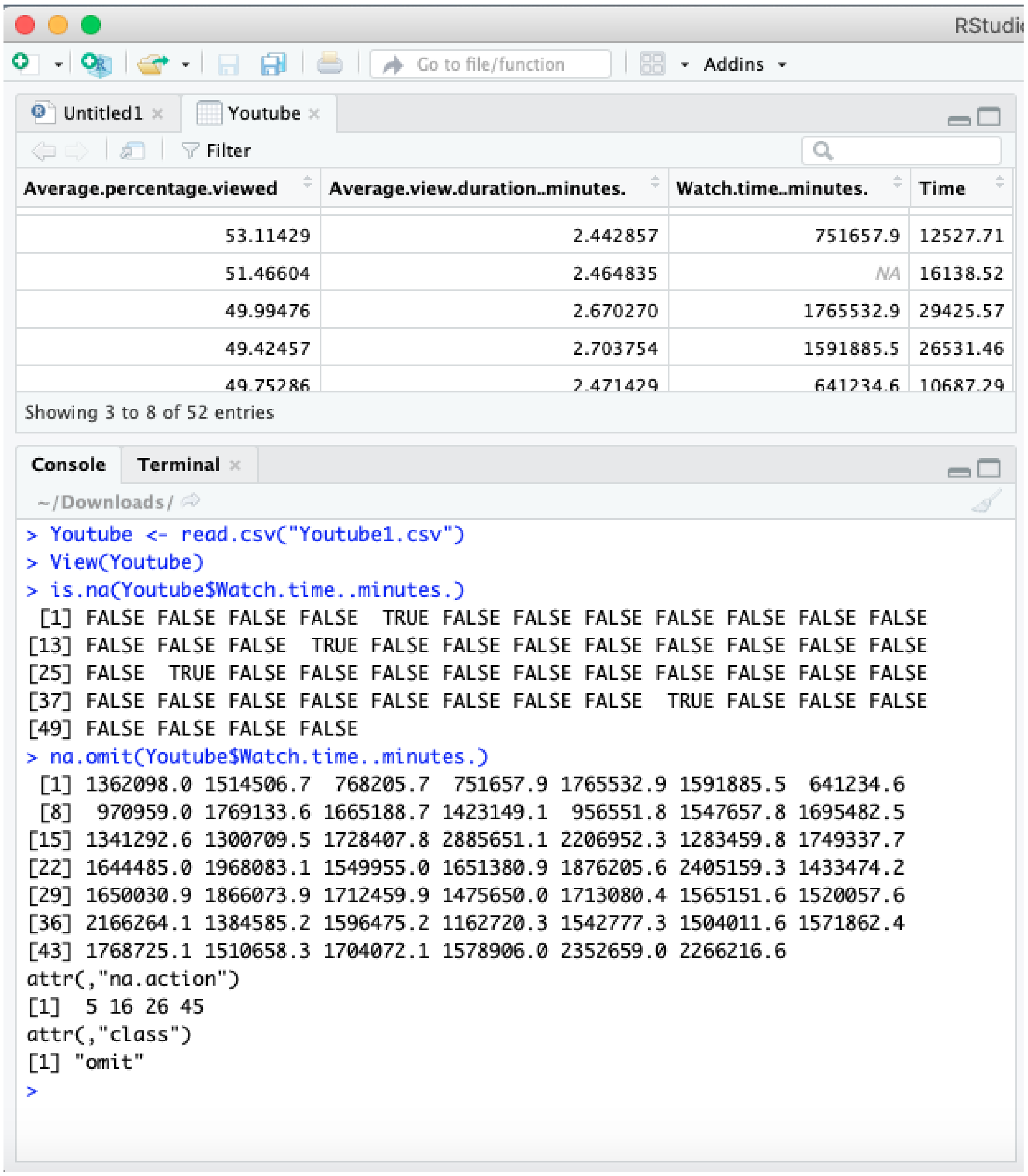
YouTube is the most popular social network for video, and in this project, we'll look at the data generated by a content company that primarily creates and distributes content with a digital-first approach.

B. Data Cleaning

Data cleaning is done so that we can visualize the data in a better way, and to reduce the errors in visualization. Most algorithms will not process records with missing values. For the same, the default is to- delete or remove the irrelevant columns, splitting columns if necessary, combining columns, deleting columns or rows with empty values, etc. In RStudio, data cleaning can be done using libraries. In this dataset we use the library 'dplyr' for the data cleaning process.

1. Omission of certain data

Here we are cleaning the column named 'Watch.time.minute' and removing the blank cells. Since not many records are missing values, we can omit them so that it will help us to visualize the data more accurately. However, omission is not a practical idea and cannot be done if many records have missing values.



Average.percentage.viewed	Average.view.duration..minutes.	Watch.time..minutes.	Time
53.11429	2.442857	751657.9	12527.71
51.46604	2.464835	NA	16138.52
49.99476	2.670270	1765532.9	29425.57
49.42457	2.703754	1591885.5	26531.46
49.75286	2.471429	641234.6	10687.29

Showing 3 to 8 of 52 entries

```
> Youtube <- read.csv("Youtube1.csv")
> View(Youtube)
> is.na(Youtube$Watch.time..minutes.)
 [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[13] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
[49] FALSE FALSE FALSE FALSE
> na.omit(Youtube$Watch.time..minutes.)
 [1] 1362098.0 1514506.7  768205.7  751657.9 1765532.9 1591885.5  641234.6
 [8]  970959.0 1769133.6 1665188.7 1423149.1  956551.8 1547657.8 1695482.5
[15] 1341292.6 1300709.5 1728407.8 2885651.1 2206952.3 1283459.8 1749337.7
[22] 1644485.0 1968083.1 1549955.0 1651380.9 1876205.6 2405159.3 1433474.2
[29] 1650030.9 1866073.9 1712459.9 1475650.0 1713080.4 1565151.6 1520057.6
[36] 2166264.1 1384585.2 1596475.2 1162720.3 1542777.3 1504011.6 1571862.4
[43] 1768725.1 1510658.3 1704072.1 1578906.0 2352659.0 2266216.6
attr(,"na.action")
[1]  5 16 26 45
attr(,"class")
[1] "omit"
>
```

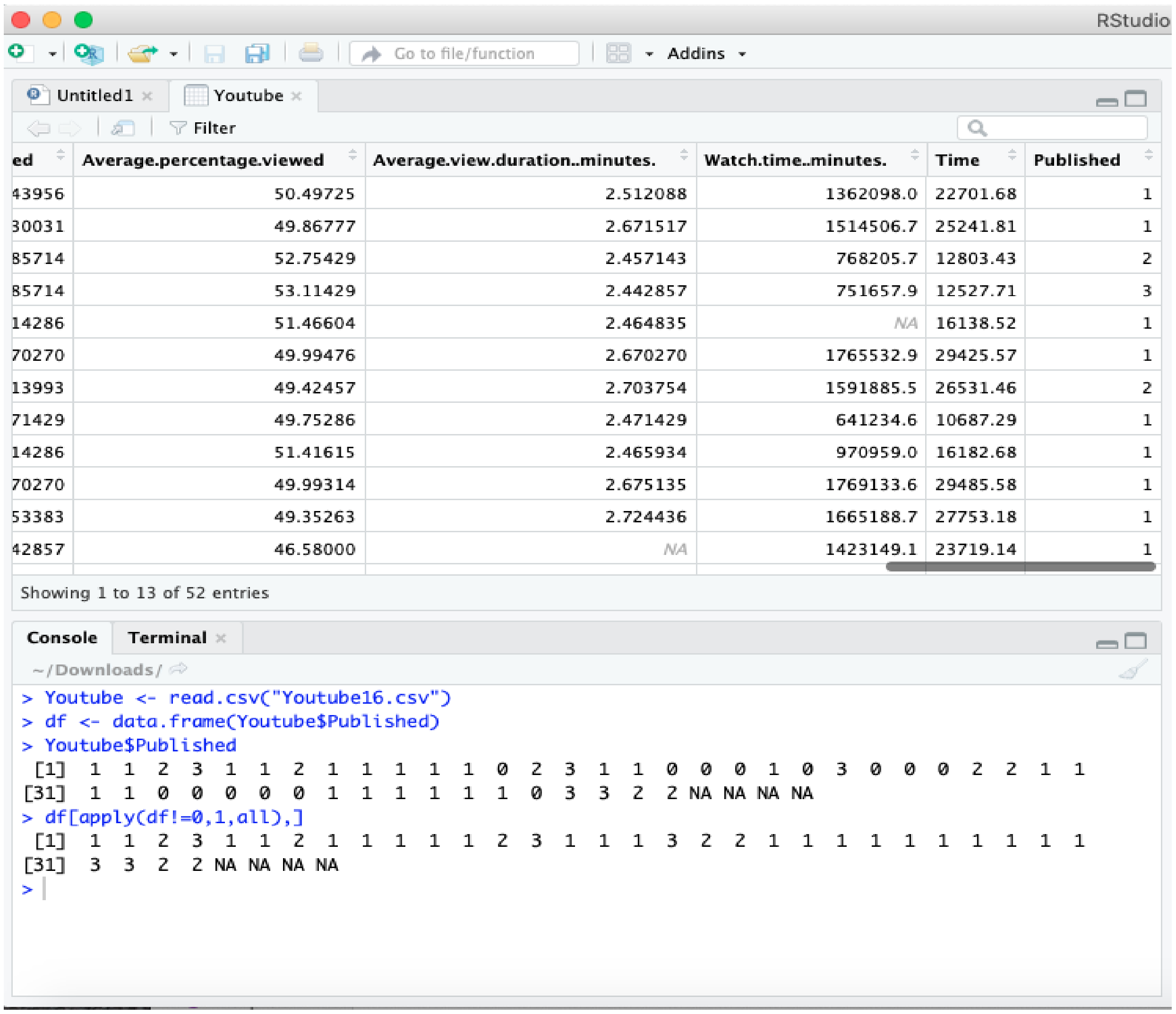
Code:

```
> is.na(Youtube$Watch.time..minutes.)
 [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE
```

```
[16]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
FALSE FALSE FALSE FALSE
[31] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE  TRUE
[46] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> na.omit(Youtube$Watch.time..minutes.)
 [1]  1362098.0  1514506.7   768205.7   751657.9  1765532.9  1591885.5   641234.6   970959.0
1769133.6
[10]  1665188.7  1423149.1   956551.8  1547657.8  1695482.5  1341292.6  1300709.5  1728407.8
2885651.1
[19]  2206952.3  1283459.8  1749337.7  1644485.0  1968083.1  1549955.0  1651380.9  1876205.6
2405159.3
[28]  1433474.2  1650030.9  1866073.9  1712459.9  1475650.0  1713080.4  1565151.6  1520057.6
2166264.1
[37]  1384585.2  1596475.2  1162720.3  1542777.3  1504011.6  1571862.4  1768725.1  1510658.3
1704072.1
[46] 1578906.0 2352659.0 2266216.6
attr(,"na.action")
[1]  5 16 26 45
attr(,"class")
[1] "omit"
```

2. Setting rule by applying a condition for data cleaning

In this method, we remove the value “0” from the data for particular columns and for that we have firstly created a data frame. Next, we transfer the particular data into that data frame and apply the set of rule that will help us to remove undesired values from the data set.



ed	Average.percentage.viewed	Average.view.duration..minutes.	Watch.time..minutes.	Time	Published
43956	50.49725	2.512088	1362098.0	22701.68	1
30031	49.86777	2.671517	1514506.7	25241.81	1
85714	52.75429	2.457143	768205.7	12803.43	2
85714	53.11429	2.442857	751657.9	12527.71	3
14286	51.46604	2.464835	NA	16138.52	1
70270	49.99476	2.670270	1765532.9	29425.57	1
13993	49.42457	2.703754	1591885.5	26531.46	2
71429	49.75286	2.471429	641234.6	10687.29	1
14286	51.41615	2.465934	970959.0	16182.68	1
70270	49.99314	2.675135	1769133.6	29485.58	1
53383	49.35263	2.724436	1665188.7	27753.18	1
42857	46.58000	NA	1423149.1	23719.14	1

Showing 1 to 13 of 52 entries

```
> Youtube <- read.csv("Youtube16.csv")
> df <- data.frame(Youtube$Published)
> Youtube$Published
 [1]  1  1  2  3  1  1  2  1  1  1  1  1  0  2  3  1  1  0  0  0  1  0  3  0  0  0  2  2  1  1
[31]  1  1  0  0  0  0  0  1  1  1  1  1  1  0  3  3  2  2 NA NA NA NA
> df[apply(df!=0,1,all),]
 [1]  1  1  2  3  1  1  2  1  1  1  1  1  2  3  1  1  1  3  2  2  1  1  1  1  1  1  1  1  1  1
[31]  3  3  2  2 NA NA NA NA
> 
```

Code:

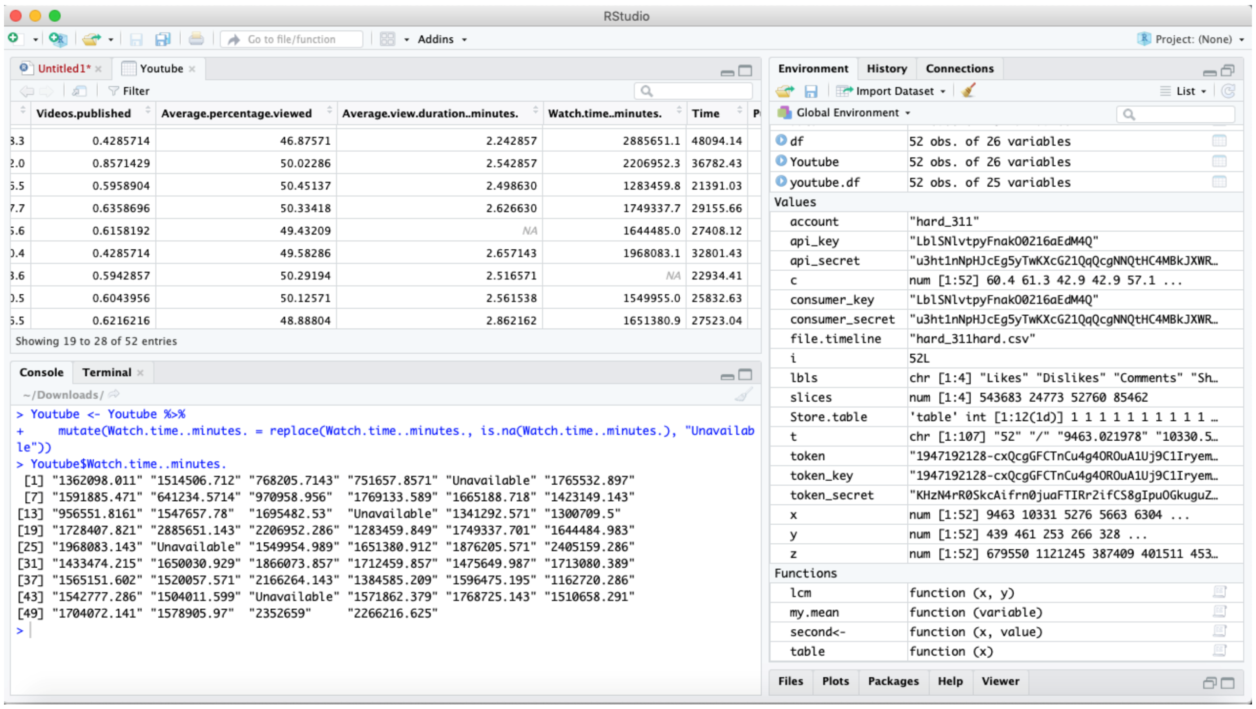
> Youtube <- read.csv("Youtube16.csv")

> df <- data.frame(Youtube$Published)

> Youtube$Published

```
 [1]  1  1  2  3  1  1  2  1  1  1  1  1  0  2  3  1  1  0  0  0  1  0  3  0  0  0  2  2  1  1

[31]  1  1  0  0  0  0  0  1  1  1  1  1  1  0  3  3  2  2 NA NA NA NA

> df[apply(df!=0,1,all),]

 [1]  1  1  2  3  1  1  2  1  1  1  1  1  2  3  1  1  1  3  2  2  1  1  1  1  1  1  1  1  1  1

[31]  3  3  2  2 NA NA NA NA
```

3. Method- 3: Replace “NA” keyword with “Unavailable” by using replace function.



	Videos.published	Average.percentage.viewed	Average.view.duration..minutes.	Watch.time..minutes.	Time
3.3	0.4285714	46.87571	2.242857	2885651.1	48094.14
2.0	0.8571429	50.02286	2.542857	2206952.3	36782.43
5.5	0.5958904	50.45137	2.498630	1283459.8	21391.03
7.7	0.6358696	50.33418	2.626630	1749337.7	29155.66
5.6	0.6158192	49.43209	NA	1644485.0	27408.12
0.4	0.4285714	49.58286	2.657143	1968083.1	32801.43
3.6	0.5942857	50.29194	2.516571	NA	22934.41
0.5	0.6043956	50.12571	2.561538	1549955.0	25832.63
5.5	0.6216216	48.88804	2.862162	1651380.9	27523.04

Showing 19 to 28 of 52 entries

```
> Youtube <- Youtube %>%
+     mutate(Watch.time..minutes. = replace(Watch.time..minutes., is.na(Watch.time..minutes.), "Unavailab
le"))
> Youtube$Watch.time..minutes.
 [1] "1362098.011" "1514506.712" "768205.7143" "751657.8571" "Unavailable" "1765532.897"
 [7] "1591885.471" "641234.5714" "970958.956"  "1769133.589" "1665188.718" "1423149.143"
[13] "956551.8161" "1547657.78"  "1695482.53"  "Unavailable" "1341292.571" "1300709.5"
[19] "1728407.821" "2885651.143" "2206952.286" "1283459.849" "1749337.701" "1644484.983"
[25] "1968083.143" "Unavailable" "1549954.989" "1651380.912" "1876205.571" "2405159.286"
[31] "1433474.215" "1650030.929" "1866073.857" "1712459.857" "1475649.987" "1713080.389"
[37] "1565151.602" "1520057.571" "2166264.143" "1384585.209" "1596475.195" "1162720.286"
[43] "1542777.286" "1504011.599" "Unavailable" "1571862.379" "1768725.143" "1510658.291"
[49] "1704072.141" "1578905.97"  "2352659"     "2266216.625"
> 
```

df	52 obs. of 26 variables
Youtube	52 obs. of 26 variables
youtube.df	52 obs. of 25 variables
Values	
account	"hard_311"
api_key	"LblSNlvtpyFnakO0216aEdM4Q"
api_secret	"u3ht1nNpHJcEg5yTwKXcG21QqQcgNNQtHC4MBkJXWR…
c	num [1:52] 60.4 61.3 42.9 42.9 57.1 ...
consumer_key	"LblSNlvtpyFnakO0216aEdM4Q"
consumer_secret	"u3ht1nNpHJcEg5yTwKXcG21QqQcgNNQtHC4MBkJXWR…
file.timeline	"hard_311hard.csv"
i	52L
lbls	chr [1:4] "Likes" "Dislikes" "Comments" "Sh…
slices	num [1:4] 543683 24773 52760 85462
Store.table	'table' int [1:12(1d)] 1 1 1 1 1 1 1 1 1 1 …
t	chr [1:107] "52" "/" "9463.021978" "10330.5…
token	"1947192128-cxQcgGFCTnCu4g4OROuA1Uj9C1Iryem…
token_key	"1947192128-cxQcgGFCTnCu4g4OROuA1Uj9C1Iryem…
token_secret	"KHzN4rR0SkcAifrn0juaFTIRr2ifCS8gIpuOGkuguZ…
x	num [1:52] 9463 10331 5276 5663 6304 ...
y	num [1:52] 439 461 253 266 328 ...
z	num [1:52] 679550 1121245 387409 401511 453…
Functions	
lcm	function (x, y)
my.mean	function (variable)
second<-	function (x, value)
table	function (x)

Code :

```
Youtube <- Youtube %>%

   mutate (Watch.time..minutes. = replace (Watch.time..minutes., is.na(Watch.time..minutes.),
"Unavailable"))
```

C. Data Analysis and Visualizations

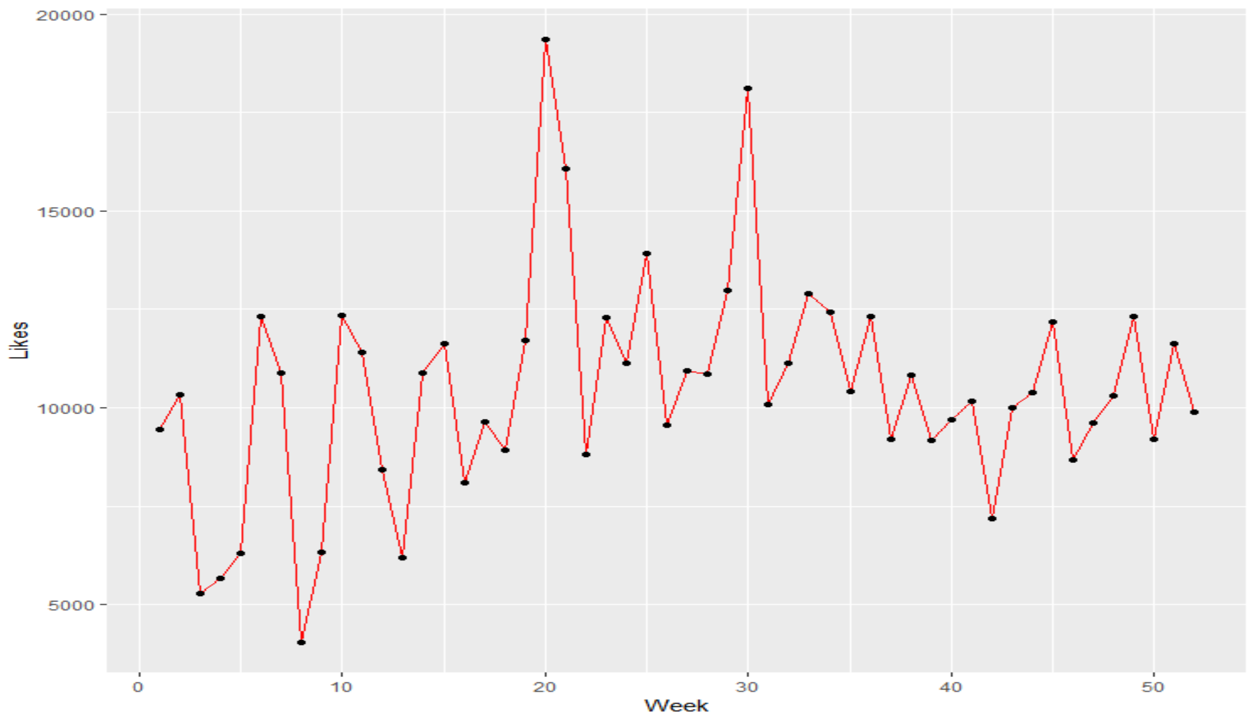
1. Plot-1: Line Plot- Week vs. Likes

1.1 Code:

```
> library(ggplot2)

> library(dplyr)

> library(scales)

> ggplot(data = head(youtube,n=52), aes(x=Week, y=Likes, group=1)) +

+  geom_line(color="red") + geom_point()
```

1.2 Question:

Do the number of likes for the YouTube videos change as time (in weeks) passes by?



(Plot 1)

1.3 Interpretation:

The above graph shows the changes in the number of likes on the videos uploaded on YouTube as weeks pass by for 2018. It can be seen that the number of likes in the entire year roughly lie between 4,500 to 20,000. The lowest number of likes are encountered in the 8th week, and the highest number of likes in week 20. Overall, the data sees a great deal of variation when it comes to the number of likes throughout the year. There are certain weeks that see extreme variations. There could be several reasons for the that, like- upload of videos regarding trending issues or important events; holiday season, when people spend their leisure time on YouTube; at times the reason for such spikes could also be massive data reporting errors.

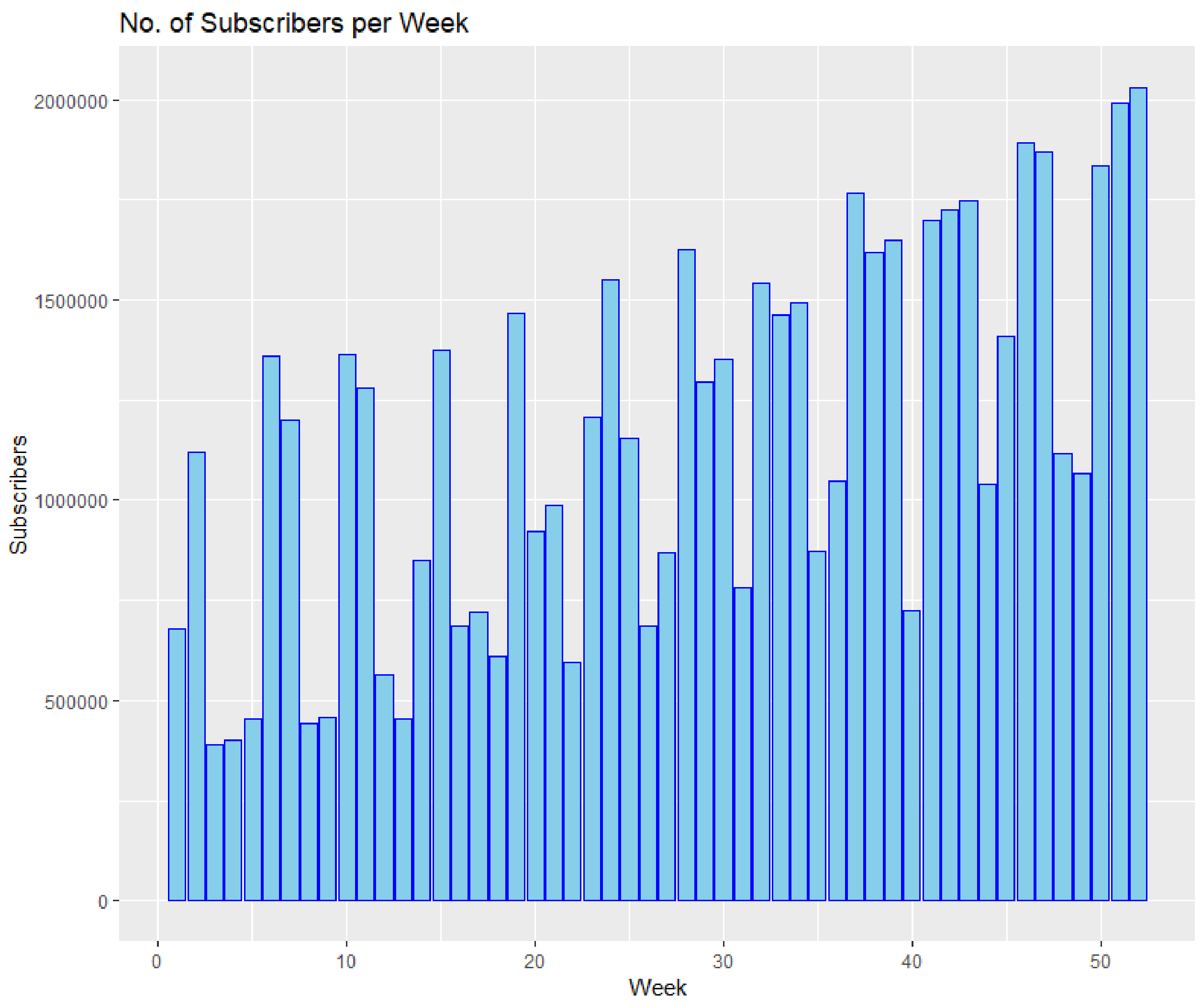
2. **Plot-2: Bar Graph- Week vs. Subscribers**

2.1 Code:

```
> library(ggplot2)

> library(dplyr)

> library(scales)

> ggplot(Youtube)+ geom_bar(aes(x=Week, y=Subscribers),  stat = 'summary', fun.y= 'mean',
fill='skyblue', color='blue')+labs(title="No. of Subscribers per
Week",x="Week",y="Subscribers")
```

2.2 Question:

Do the number of subscribers keep on increasing over time (in weeks)?



(Plot 2)

2.3 Interpretation:

The bar graph shows the number of YouTube subscribers per week for 2018. The question is based on the assumption that, as time passes by and a greater number of people have access to smartphones, internet, and get educated about platforms such as YouTube, the number of subscribers would keep on increasing. However, the same result cannot be seen from the graph. There is a constant increase and decrease in the number of subscribers on a weekly basis. It can be seen that the least number of subscribers are there in week 3. When the year comes to an end, the number of subscribers keep on increasing, the highest number being in the last week, i.e., week 52.

The possible reasons for the same could be- the viewers keep on subscribing and unsubscribing based on their interest in the YouTube channel; YouTube removing spam subscribers; increase or decrease in the social networks of the people who upload the content; the uploaded content is not what the subscribers had signed up for, etc.

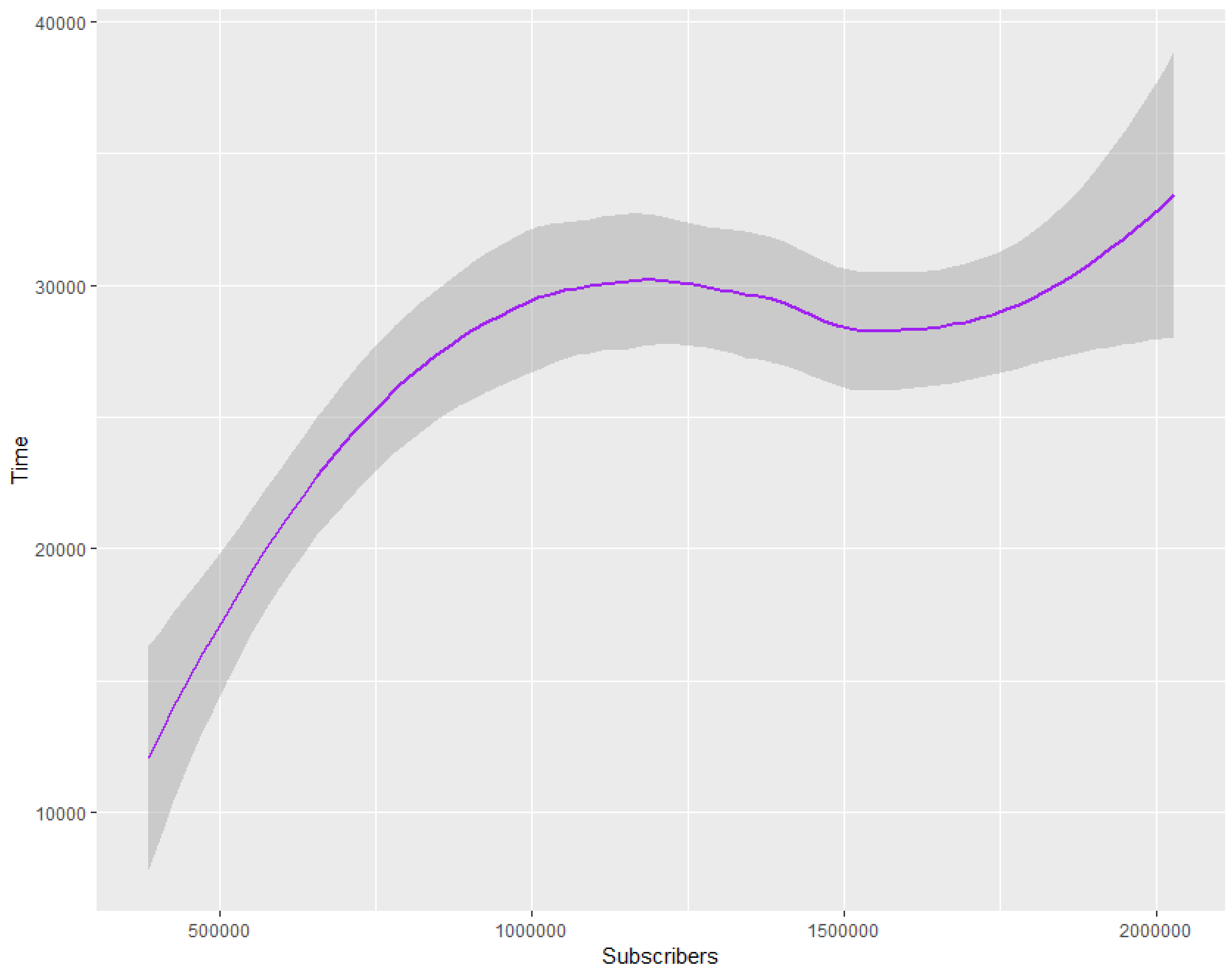
3. <u>Plot-3: Smooth Line Plot- Subscribers vs. Time (in hours)</u>

3.1 Code:

```
> library(ggplot2)
> library(dplyr)
> library(scales)
> ggplot(youtube.df)
+geom_smooth (aes (x = Subscribers, y = Time), color = 'purple', method = 'loess')
```

3.2 Question:

Does the increase in the number of subscribers increase the watch time (in hours) of viewers?



(Plot 3)

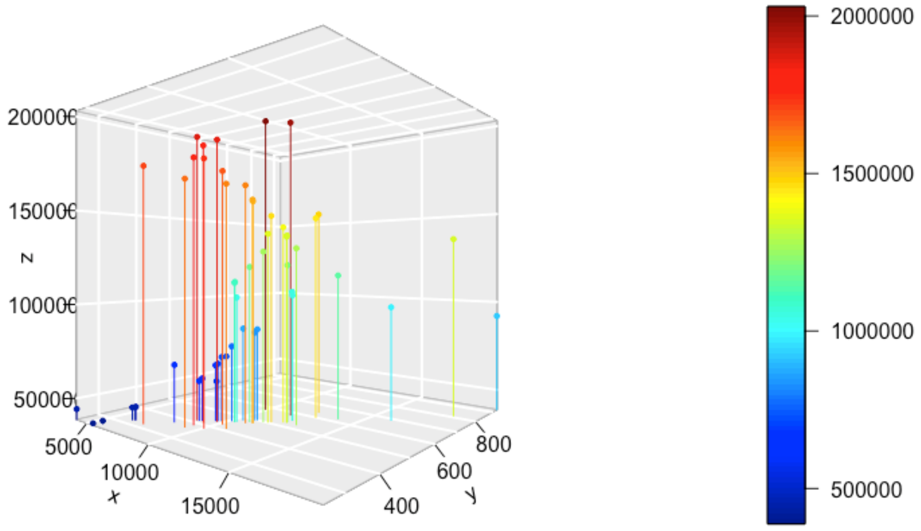
3.3 Interpretation:

The above graph shows the relationship between the number of subscribers and the watch time of viewers in hours, in the form of a smooth line plot. The number of subscribers for the year 2018 lies roughly between 380,000 to a little more than 2,000,000, and the range of the watch time in hours in between 10,000 hours to almost 50,000 hours. It can be seen from the graph that the watch time keeps on increasing till the number of subscribers reach about 1,200,000. There is a small drop in the watch time until the number of consumers reach to 1,750,000, and after that, there is once again a steady increase in the watch time. The reasons for the drop could be- no interesting uploads, busy schedules, etc. Other than that, there is a constant increase in the view time (in hrs).

4. **Plot-4:3D Visualization of Likes Vs Dislikes Vs Subscribers (Extra Graph)**

4.1 Code:

```
> x <- Youtube$Likes

> y <- Youtube$Dislikes

> z <- Youtube$Subscribers

> library (Plotrix)

> scatter3D(x, y, z, phi = 0, bty = "g",  type = "h",   ticktype = "detailed", pch = 19, cex = 0.5)
```

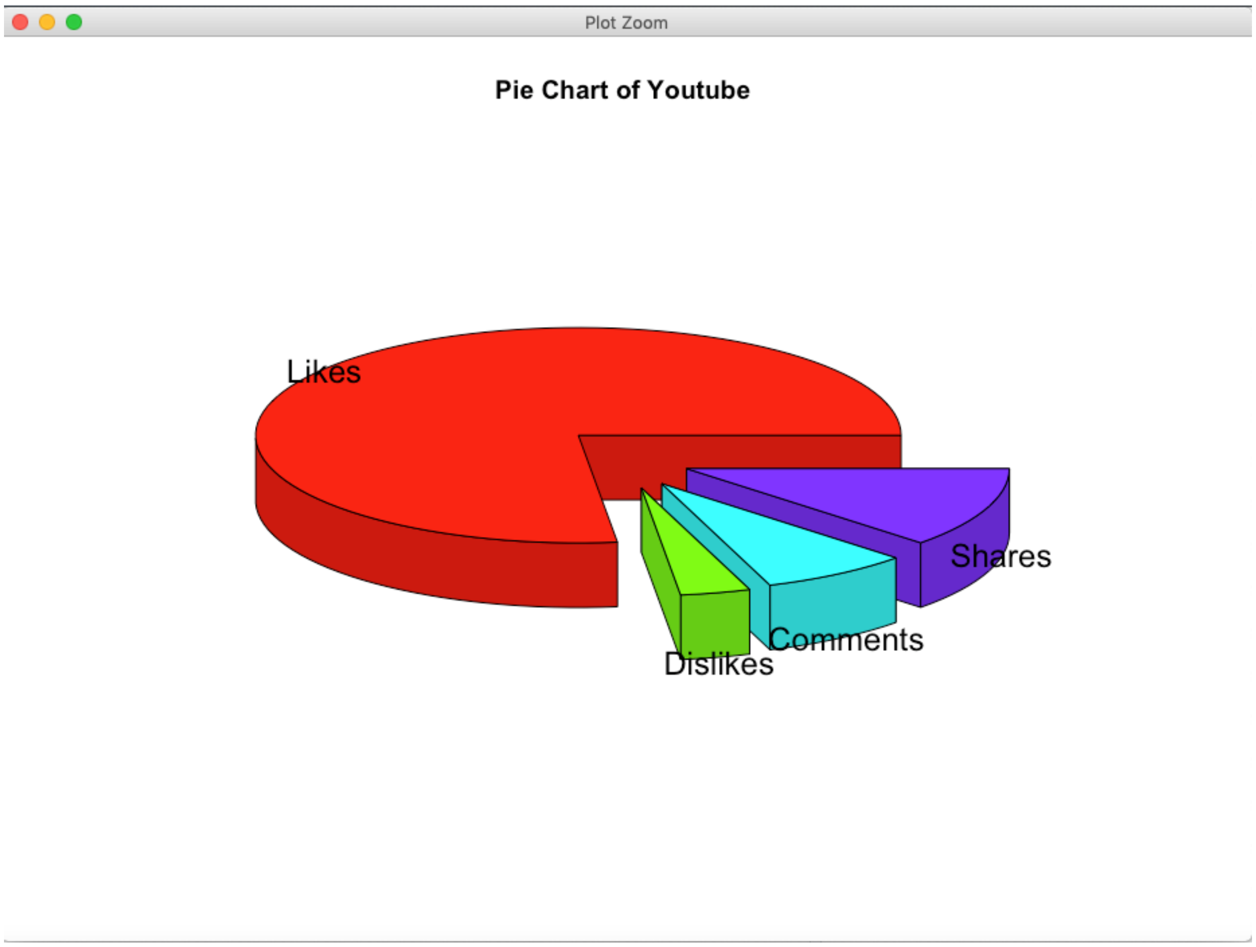


(Plot 4)

5. Plot-5: 3D Pie Chart of Likes, Dislikes, Comments, Shares

5.1 Code:

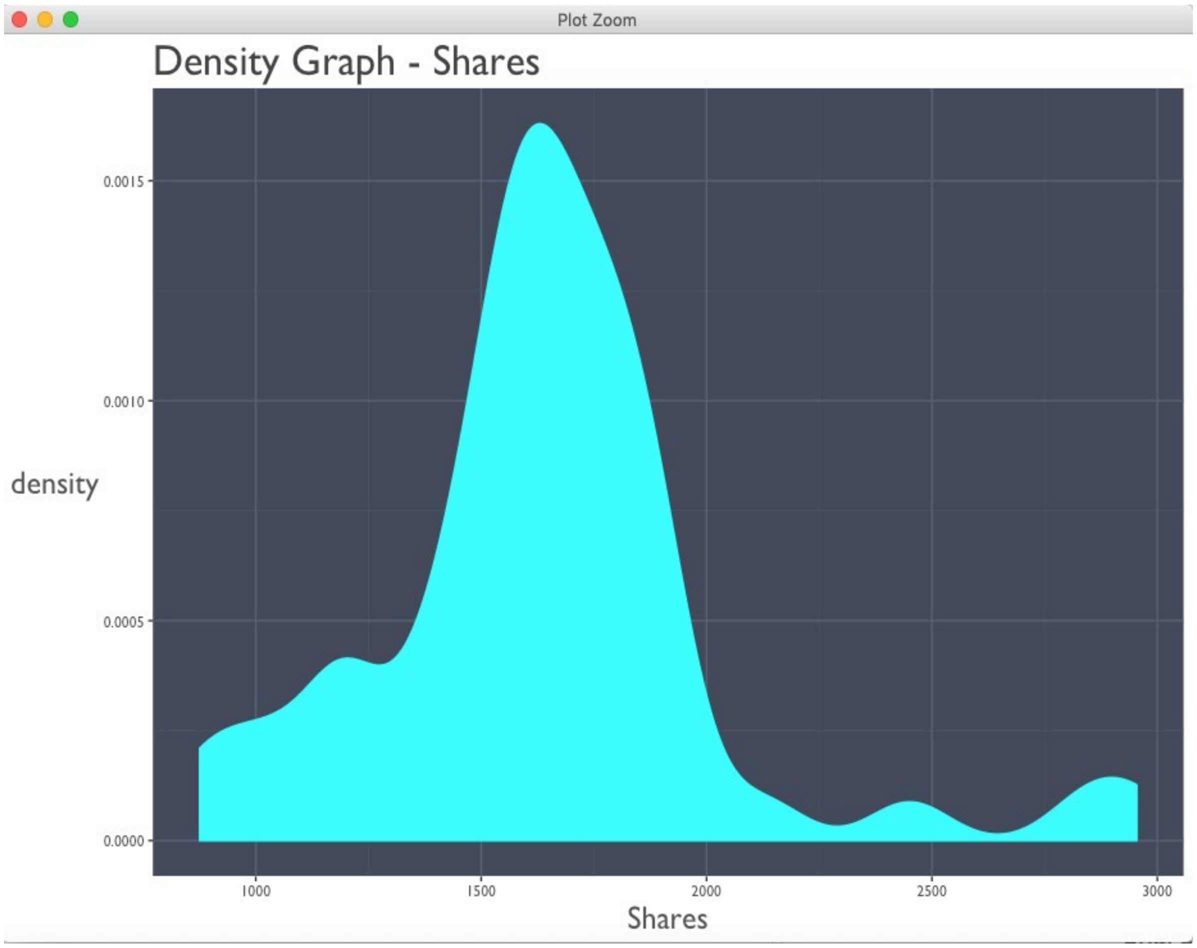
```
> slices <-
c(sum(youtube.df$Likes),sum(youtube.df$Dislikes),sum(youtube.df$Comments),sum(youtu
be.df$Shares))
>
> lbls <- c("Likes", "Dislikes","Comments","Shares")
> pie3D(slices,labels=lbls,explode=0.2,main="Pie Chart of Youtube ")
```



6 <u>Plot-6: Density Graph of Number of Video Shares</u>

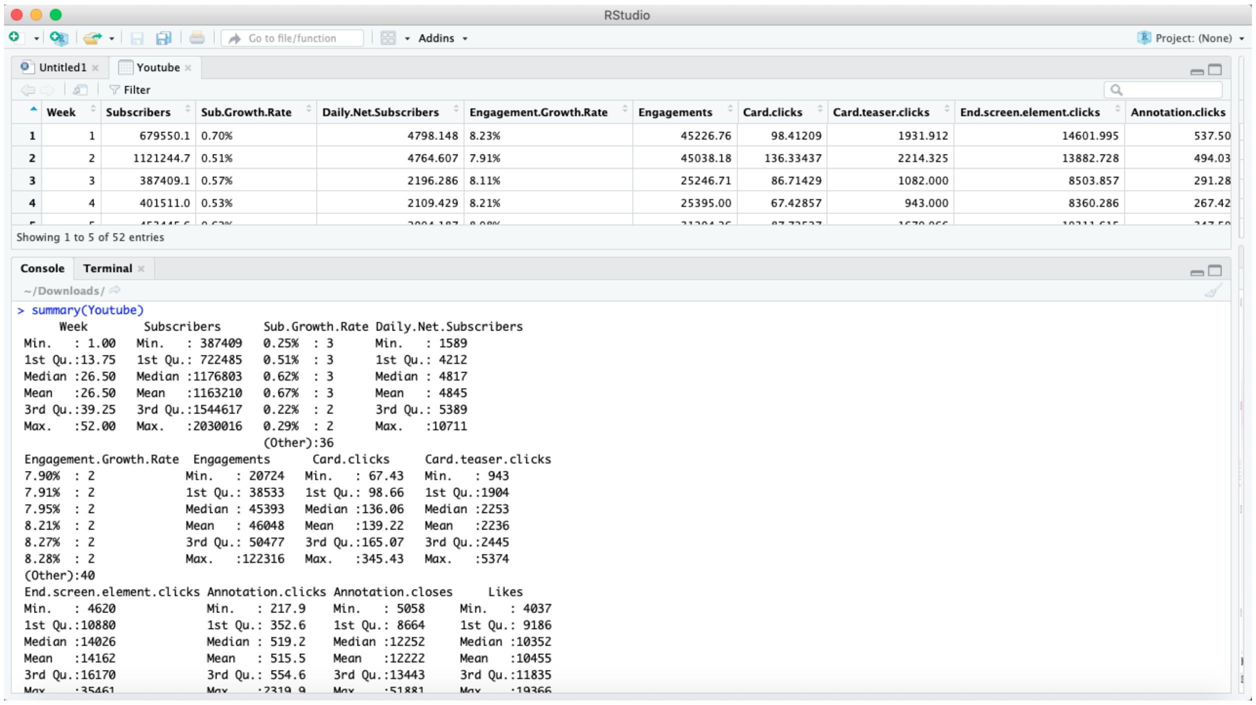
6.1 Code:

```
> ggplot(data = Youtube, aes(x = Shares)) +
+     geom_density(fill = 'cyan', color = 'cyan') +
+     labs(title = 'Density Graph - Shares') +
+     theme(text = element_text(family = 'Gill Sans', color = "#444444")
+           ,panel.background = element_rect(fill = '#444B5A')
+           ,panel.grid.minor = element_line(color = '#4d5566')
+           ,panel.grid.major = element_line(color = '#586174')
+           ,plot.title = element_text(size = 24)
+           ,axis.title = element_text(size = 18, color = '#555555')
+           ,axis.title.y = element_text(vjust = .5, angle = 0)
+           ,axis.title.x = element_text(hjust = .5)
+     )
```


Plot Zoom
Density Graph - Shares
density
0.0015
0.0010
0.0005
0.0000
1000
1500
2000
2500
3000
Shares

(Plot 6)

7 Statistical Summary and Functions



	Week	Subscribers	Sub.Growth.Rate	Daily.Net.Subscribers	Engagement.Growth.Rate	Engagements	Card.clicks	Card.teaser.clicks	End.screen.element.clicks	Annotation.clicks
1	1	679550.1	0.70%	4798.148	8.23%	45226.76	98.41209	1931.912	14601.995	537.50
2	2	1121244.7	0.51%	4764.607	7.91%	45038.18	136.33437	2214.325	13882.728	494.03
3	3	387409.1	0.57%	2196.286	8.11%	25246.71	86.71429	1082.000	8503.857	291.28
4	4	401511.0	0.53%	2109.429	8.21%	25395.00	67.42857	943.000	8360.286	267.42

Showing 1 to 5 of 52 entries

Console Terminal

~/Downloads/

```
> summary(Youtube)
      Week        Subscribers      Sub.Growth.Rate Daily.Net.Subscribers
 Min.   : 1.00   Min.   : 387409   0.25%  : 3      Min.   : 1589
 1st Qu.:13.75   1st Qu.: 722485   0.51%  : 3      1st Qu.: 4212
 Median :26.50   Median :1176803   0.62%  : 3      Median : 4817
 Mean   :26.50   Mean   :1163210   0.67%  : 3      Mean   : 4845
 3rd Qu.:39.25   3rd Qu.:1544617   0.22%  : 2      3rd Qu.: 5389
 Max.   :52.00   Max.   :2030016   0.29%  : 2      Max.   :10711
                                   (Other):36
 Engagement.Growth.Rate  Engagements      Card.clicks     Card.teaser.clicks
 7.90%  : 2             Min.   : 20724   Min.   : 67.43   Min.   : 943
 7.91%  : 2             1st Qu.: 38533   1st Qu.: 98.66   1st Qu.:1904
 7.95%  : 2             Median : 45393   Median :136.06   Median :2253
 8.21%  : 2             Mean   : 46048   Mean   :139.22   Mean   :2236
 8.27%  : 2             3rd Qu.: 50477   3rd Qu.:165.07   3rd Qu.:2445
 8.28%  : 2             Max.   :122316   Max.   :345.43   Max.   :5374
 (Other):40
 End.screen.element.clicks Annotation.clicks Annotation.closes     Likes
 Min.   : 4620             Min.   : 217.9    Min.   : 5058     Min.   : 4037
 1st Qu.:10880             1st Qu.: 352.6    1st Qu.: 8664     1st Qu.: 9186
 Median :14026             Median : 519.2    Median :12252     Median :10352
 Mean   :14162             Mean   : 515.5    Mean   :12222     Mean   :10455
 3rd Qu.:16170             3rd Qu.: 554.6    3rd Qu.:13443     3rd Qu.:11835
 Max.   :35461             Max.   :2319.9    Max.   :51881     Max.   :19366
```

Code:

```
> summary(Youtube)
      Week        Subscribers      Sub.Growth.Rate Daily.Net.Subscribers
 Min.   : 1.00   Min.   : 387409   0.25%  : 3      Min.   : 1589
 1st Qu.:13.75   1st Qu.: 722485   0.51%  : 3      1st Qu.: 4212
 Median :26.50   Median :1176803   0.62%  : 3      Median : 4817
 Mean   :26.50   Mean   :1163210   0.67%  : 3      Mean   : 4845
 3rd Qu.:39.25   3rd Qu.:1544617   0.22%  : 2      3rd Qu.: 5389
 Max.   :52.00   Max.   :2030016   0.29%  : 2      Max.   :10711
                                   (Other):36
```

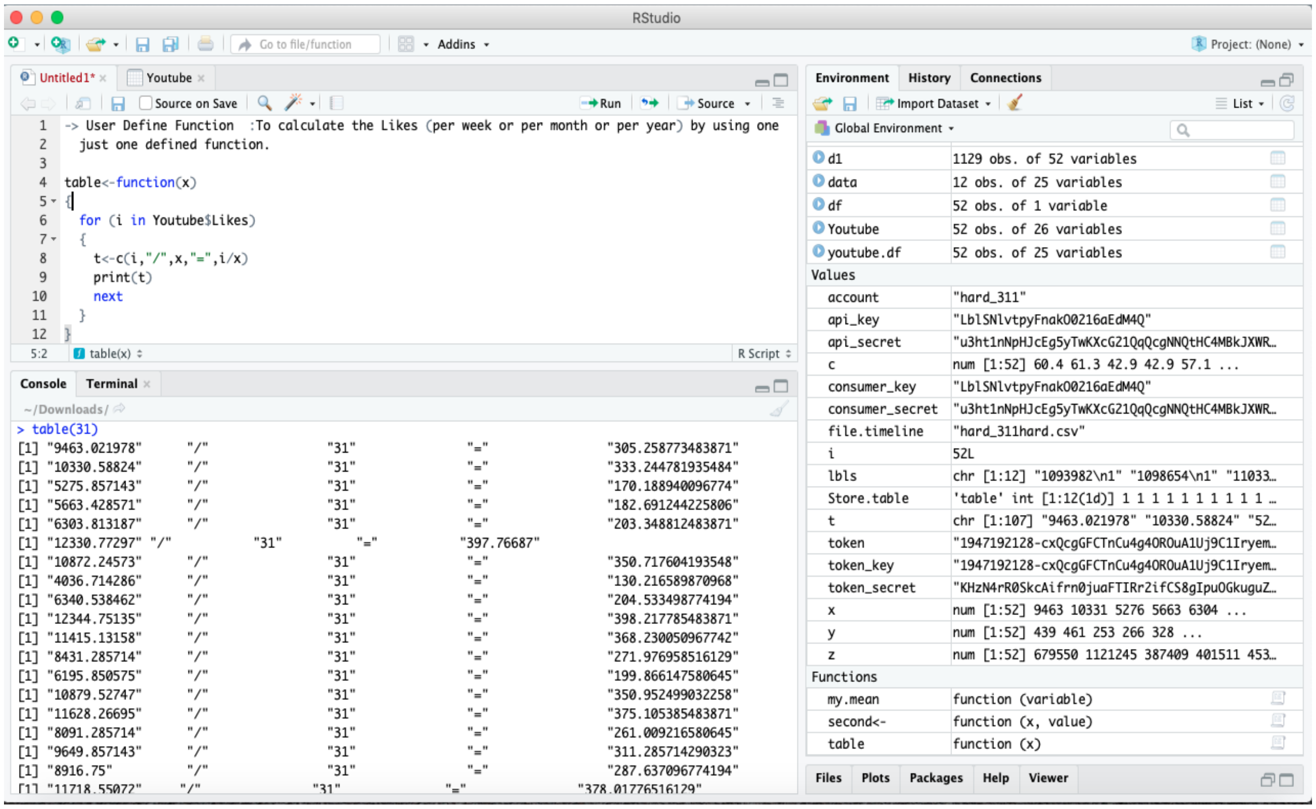
```
Engagement.Growth.Rate  Engagements     Card.clicks     Card.teaser.clicks
7.90%  : 2           Min.   : 20724   Min.   : 67.43   Min.   : 943
7.91%  : 2           1st Qu.: 38533   1st Qu.: 98.66   1st Qu.:1904
7.95%  : 2           Median : 45393   Median :136.06   Median :2253
8.21%  : 2           Mean   : 46048   Mean   :139.22   Mean   :2236
8.27%  : 2           3rd Qu.: 50477   3rd Qu.:165.07   3rd Qu.:2445
8.28%  : 2           Max.   :122316   Max.   :345.43   Max.   :5374
(Other):40

End.screen.element.clicks Annotation.clicks Annotation.closes     Likes
Min.   : 4620            Min.   : 217.9    Min.   : 5058     Min.   : 4037
1st Qu.:10880            1st Qu.: 352.6    1st Qu.: 8664     1st Qu.: 9186
Median :14026            Median : 519.2    Median :12252     Median :10352
Mean   :14162            Mean   : 515.5    Mean   :12222     Mean   :10455
3rd Qu.:16170            3rd Qu.: 554.6    3rd Qu.:13443     3rd Qu.:11835
Max.   :35461            Max.   :2319.9    Max.   :51881     Max.   :19366

   Dislikes      Sentiment     Shares        Comments      Convesation.Rate
Min.   :246.0   95.59% : 3   Min.   : 875.6   Min.   : 433.3   0.17%  :17
1st Qu.:415.8   95.78% : 3   1st Qu.:1504.9   1st Qu.: 889.2   0.18%  :10
Median :466.9   94.95% : 2   Median :1626.2   Median :1048.4   0.16%  : 4
Mean   :476.4   95.31% : 2   Mean   :1643.5   Mean   :1014.6   0.19%  : 4
3rd Qu.:529.3   95.44% : 2   3rd Qu.:1792.0   3rd Qu.:1097.3   0.20%  : 4
```

```
Max.   :930.4   95.50% : 2   Max.   :2954.4   Max.   :2021.3   0.14%  : 3
           (Other):38                       (Other):10

Videos.in.playlists  Shares.View     Views        Videos.published
Min.   :1372       0.29%  :14   Min.   : 260392   Min.   :0.0000
1st Qu.:2726       0.28%  :10   1st Qu.: 519685   1st Qu.:0.5284
Median :3132       0.30%  : 8   Median : 578115   Median :0.6087
Mean   :3183       0.31%  : 7   Mean   : 590177   Mean   :0.5907
3rd Qu.:3519       0.32%  : 4   3rd Qu.: 661417   3rd Qu.:0.6344
Max.   :5465       0.16%  : 1   Max.   :1269318   Max.   :1.0000
                   (Other): 8

Average.percentage.viewed Average.view.duration..minutes. Watch.time..minutes.
Min.   :39.92            Min.   :2.243                  Min.   : 641235
1st Qu.:48.11            1st Qu.:2.539                  1st Qu.:1430893
Median :49.67            Median :2.657                  Median :1585396
Mean   :49.19            Mean   :2.689                  Mean   :1605422
3rd Qu.:50.35            3rd Qu.:2.761                  3rd Qu.:1753386
Max.   :65.14            Max.   :3.471                  Max.   :2885651
                         NA's   :5                      NA's   :4

    Time        Published
Min.   :10687   Min.   :0.000
```

```
1st Qu.:23041   1st Qu.:0.000

Median :26256   Median :1.000

Mean   :26395   Mean   :1.062

3rd Qu.:29206   3rd Qu.:1.250

Max.   :48094   Max.   :3.000

                NA's   :4
```

User Defined Function:

To calculate the Total Likes (*per week or per month or per year*) by using just one defined function.



```r
-> User Define Function  :To calculate the Likes (per week or per month or per year) by using one
  just one defined function.

table<-function(x)
{
  for (i in Youtube$Likes)
  {
    t<-c(i,"/",x,"=",i/x)
    print(t)
    next
  }
}
```

```
> table(31)
[1] "9463.021978"    "/"              "31"             "="              "305.258773483871"
[1] "10330.58824"    "/"              "31"             "="              "333.244781935484"
[1] "5275.857143"    "/"              "31"             "="              "170.188940096774"
[1] "5663.428571"    "/"              "31"             "="              "182.691244225806"
[1] "6303.813187"    "/"              "31"             "="              "203.348812483871"
[1] "12330.77297" "/"           "31"          "="           "397.76687"
[1] "10872.24573"    "/"              "31"             "="              "350.717604193548"
[1] "4036.714286"    "/"              "31"             "="              "130.216589870968"
[1] "6340.538462"    "/"              "31"             "="              "204.533498774194"
[1] "12344.75135"    "/"              "31"             "="              "398.217785483871"
[1] "11415.13158"    "/"              "31"             "="              "368.230050967742"
[1] "8431.285714"    "/"              "31"             "="              "271.976958516129"
[1] "6195.850575"    "/"              "31"             "="              "199.866147580645"
[1] "10879.52747"    "/"              "31"             "="              "350.952499032258"
[1] "11628.26695"    "/"              "31"             "="              "375.105385483871"
[1] "8091.285714"    "/"              "31"             "="              "261.009216580645"
[1] "9649.857143"    "/"              "31"             "="              "311.285714290323"
[1] "8916.75"        "/"              "31"             "="              "287.637096774194"
[1] "11718.55072"    "/"              "31"             "="              "378.01776516129"
```

```
d1                1129 obs. of 52 variables
data              12 obs. of 25 variables
df                52 obs. of 1 variable
Youtube           52 obs. of 26 variables
youtube.df        52 obs. of 25 variables
Values
  account         "hard_311"
  api_key         "LblSNlvtpyFnak00216aEdM4Q"
  api_secret      "u3ht1nNpHJcEg5yTwKXcG21QqQcgNNQtHC4MBkJXWR…
  c               num [1:52] 60.4 61.3 42.9 42.9 57.1 ...
  consumer_key    "LblSNlvtpyFnak00216aEdM4Q"
  consumer_secret "u3ht1nNpHJcEg5yTwKXcG21QqQcgNNQtHC4MBkJXWR…
  file.timeline   "hard_311hard.csv"
  i               52L
  lbls            chr [1:12] "1093982\n1" "1098654\n1" "11033…
  Store.table     'table' int [1:12(1d)] 1 1 1 1 1 1 1 1 1 1 …
  t               chr [1:107] "9463.021978" "10330.58824" "52…
  token           "1947192128-cxQcgGFCTnCu4g40ROuA1Uj9C1Iryem…
  token_key       "1947192128-cxQcgGFCTnCu4g40ROuA1Uj9C1Iryem…
  token_secret    "KHzN4rR0SkcAifrn0juaFTIRr2ifCS8gIpu0GkuguZ…
  x               num [1:52] 9463 10331 5276 5663 6304 ...
  y               num [1:52] 439 461 253 266 328 ...
  z               num [1:52] 679550 1121245 387409 401511 453…
Functions
  my.mean         function (variable)
  second<-        function (x, value)
  table           function (x)
```

Code:

```
table<-function(x)

{

  for (i in Youtube$Likes)

  {

    t<-c(i,"/",x,"=",i/x)

    print(t)

    next

  }

}
```

8 Full Code

- is.na (Youtube$Watch.time..minutes.)
- na.omit(Youtube$Watch.time..minutes.)
- Youtube <- read.csv("Youtube16.csv")
- df <- data.frame(Youtube$Published)
- Youtube$Published
- df[apply(df!=0,1,all),]
- Youtube <- Youtube %>%

  mutate (Watch.time..minutes. = replace (Watch.time..minutes., is.na(Watch.time..minutes.), "Unavailable"))
- library(ggplot2)
- library(dplyr)
- library(scales)
- ggplot(data = head(youtube,n=52), aes(x=Week, y=Likes, group=1)) +

  geom_line(color="red") + geom_point()
- library(ggplot2)
- library(dplyr)
- library(scales)
- ggplot(Youtube)+ geom_bar(aes(x=Week, y=Subscribers), stat = 'summary', fun.y= 'mean', fill='skyblue', color='blue')+labs(title="No. of Subscribers per Week",x="Week",y="Subscribers")

- library(ggplot2)
- library(dplyr)
- library(scales)
- ggplot(youtube.df)

  + geom_smooth (aes (x = Subscribers, y = Time), color = 'purple', method = 'loess')
- x <- Youtube$Likes
- y <- Youtube$Dislikes
- z <- Youtube$Subscribers
- library (Plotrix)
- scatter3D(x, y, z, phi = 0, bty = "g", type = "h",   ticktype = "detailed", pch = 19, cex = 0.5)
- ggplot(data = Youtube, aes(x = Shares)) +

```
 geom_density(fill = 'cyan', color = 'cyan') +
 labs(title = 'Density Graph - Shares') +
 theme(text = element_text(family = 'Gill Sans', color = "#444444")
 ,panel.background = element_rect(fill = '#444B5A')
  ,panel.grid.minor = element_line(color = '#4d5566')
  ,panel.grid.major = element_line(color = '#586174')
  ,plot.title = element_text(size = 24)
  ,axis.title = element_text(size = 18, color = '#555555')
  ,axis.title.y = element_text(vjust = .5, angle = 0)
  ,axis.title.x = element_text(hjust = .5)
    )
```

- summary (Youtube)
- table<-function(x)

```
{
 for (i in Youtube$Likes)
 {
   t<-c(i,"/",x,"=",i/x)
   print(t)
   next
 }
}
```