Assignment

2023-09-08

#dataset 'Global YouTube Statistics' the data has a mix of quantitative and qualitative (categorical) variables. # Source of the dataset : Kaggle # Link of the dataset : https://www.kaggle.com/code/zifanwei/global-youtube-statistics-2023-eda/input

```
#Importing dataset Global YouTube Statistics

Global.YouTube.Statistics <- read.csv("C:/Users/drpra/Downloads/Global YouTube Statistics.csv")

#Checking the top 5-6 rows of the dataset using head function
head(Global.YouTube.Statistics)
```

```
##   rank                  Youtuber subscribers  video.views         category
## 1    1                   T-Series   245000000 228000000000            Music
## 2    2              YouTube Movies   170000000            0 Film & Animation
## 3    3                   MrBeast   166000000  28368841870    Entertainment
## 4    4 Cocomelon - Nursery Rhymes   162000000 164000000000        Education
## 5    5                  SET India   159000000 148000000000            Shows
## 6    6                      Music   119000000            0              nan
##                       Title uploads       Country Abbreviation  channel_type
## 1                   T-Series   20082         India           IN         Music
## 2              youtubemovies       1 United States           US         Games
## 3                    MrBeast     741 United States           US Entertainment
## 4 Cocomelon - Nursery Rhymes     966 United States           US     Education
## 5                  SET India  116536         India           IN Entertainment
## 6                      Music       0           nan          nan         Music
##   video_views_rank country_rank channel_type_rank
## 1                1            1                 1
## 2          4055159         7670              7423
## 3               48            1                 1
## 4                2            2                 1
## 5                3            2                 2
## 6          4057944          NaN               NaN
##   video_views_for_the_last_30_days lowest_monthly_earnings
## 1                        2.258e+09                 564600
## 2                        1.200e+01                      0
## 3                        1.348e+09                 337000
## 4                        1.975e+09                 493800
## 5                        1.824e+09                 455900
## 6                              NaN                      0
##   highest_monthly_earnings lowest_yearly_earnings highest_yearly_earnings
## 1                  9.0e+06               6.8e+06              1.084e+08
## 2                  5.0e-02               4.0e-02              5.800e-01
## 3                  5.4e+06               4.0e+06              6.470e+07
## 4                  7.9e+06               5.9e+06              9.480e+07
```

```
## 5                  7.3e+06               5.5e+06              8.750e+07
## 6                  0.0e+00               0.0e+00              0.000e+00
##   subscribers_for_last_30_days created_year created_month created_date
## 1                        2e+06         2006           Mar           13
## 2                          NaN         2006           Mar            5
## 3                        8e+06         2012           Feb           20
## 4                        1e+06         2006           Sep            1
## 5                        1e+06         2006           Sep           20
## 6                          NaN         2013           Sep           24
##   Gross.tertiary.education.enrollment.... Population Unemployment.rate
## 1                                   28.1 1366417754              5.36
## 2                                   88.2  328239523             14.70
## 3                                   88.2  328239523             14.70
## 4                                   88.2  328239523             14.70
## 5                                   28.1 1366417754              5.36
## 6                                    NaN        NaN               NaN
##   Urban_population Latitude Longitude
## 1        471031528 20.59368  78.96288
## 2        270663028 37.09024 -95.71289
## 3        270663028 37.09024 -95.71289
## 4        270663028 37.09024 -95.71289
## 5        471031528 20.59368  78.96288
## 6              NaN      NaN       NaN
```

```r
#Performing the descriptive statistics for Quantitative and Qualitative variables.

#Descriptive statistics which means finding out mean, median, mode and
#standard deviation for Quantitative variables and for Qualitative variables we summarize the frequenci

#Quantitative variables are numperical contained variables and Qualitative
#is character based variables so the dataset Global YouTube Statistics
#contains both kinds of variables.

#Grouping all the quantitative variables.

# Selected some of the quantitative variables
quantitative_variables <- c("uploads", "subscribers", "rank")

#as the dataset is huge selected only few variables for permoning

# The descriptive statistics for some of the quantitative variables
summary(Global.YouTube.Statistics[quantitative_variables])
```

```
##     uploads          subscribers            rank
##  Min.   :     0.0   Min.   : 12300000   Min.   :  1.0
##  1st Qu.:   194.5   1st Qu.: 14500000   1st Qu.:249.5
##  Median :   729.0   Median : 17700000   Median :498.0
##  Mean   :  9187.1   Mean   : 22982412   Mean   :498.0
##  3rd Qu.:  2667.5   3rd Qu.: 24600000   3rd Qu.:746.5
##  Max.   :301308.0   Max.   :245000000   Max.   :995.0
```

```r
# The frequencies of the qualitative variables
# For selecting a particular variable we use '$' symbol.
table(Global.YouTube.Statistics$country)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
##   43   30   26   19   19   17   14   15   15   13   13    9   13   12    9    7
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
##    9    5    9    3    5    7    5    8    6    6    7    5    7    5    6    7
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
##    5    6    4    3    2    3    3    4    5    3    1    3    5    4    4    1
##   49   50   51   52   53   54   55   56   57   58   59   60   61   62   63   64
##    4    5    4    3    3    3    1    1    1    2    2    3    2    2    3    2
##   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79   80
##    1    2    5    2    3    3    2    2    1    4    3    1    3    3    4    2
##   81   82   83   84   85   86   87   88   89   90   91   92   93   94   95   96
##    4    2    3    2    3    2    4    2    2    3    5    2    2    4    3    3
##   97   98   99  100  101  102  103  104  105  106  107  108  109  110  111  112
##    4    4    3    8    4    3    4    3    2    4    4    7    4    3    3    5
##  113  114  115  116  117  118  119  120  121  122  123  124  125  127  129  130
##    6    4    3    4    3    4    1    5    1    2    8    6    7    1    2    1
##  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146
##    1    1    1    4    2    1    3    3    3    4    1    3    1    1    1    1
##  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  163
##    3    2    1    2    4    4    5    6    3    6    4    1    2    2    6    1
##  164  165  166  167  168  169  170  171  172  173  174  175  176  504  522  692
##    2    3    5    5    2    3    2    4    1    4    2    6    2    1    1    1
##  730  774  867  962 1203 1385 1549 1776 1795 1874 1920 1944 2063 2251 2460 2492
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 2573 2600 2753 2774 2889 2904 2973 3017 3021 3046 3305 3309 3450 3508 3554 3624
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 3671 3695 3726 3776 3810 3885 3926 4026 4044 4072 4093 4228 4248 4258 4651 4797
##    1    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 5048 5075 5208 5227 5417 5589 5803 5889 6095 6143 6266 6342 7574 7615 7627 7670
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 7683 7692 7700 7711 7736 7741
##    1    1    1    1    1    1
```

```
table(Global.YouTube.Statistics$channel_type)
```

```
##
##       Animals          Autos         Comedy      Education Entertainment
##             3              3             51             49            304
##          Film         Games          Howto          Music            nan
##            42             98             36            216             30
##          News     Nonprofit         People         Sports           Tech
##            30              2            101             13             17
```

```
table(Global.YouTube.Statistics$category)
```

```
##
##      Autos & Vehicles                Comedy             Education
##                     2                    69                    45
##         Entertainment      Film & Animation                Gaming
##                   241                    46                    94
##         Howto & Style                Movies                 Music
##                    40                     2                   202
```

```
##                   nan       News & Politics Nonprofits & Activism
##                    46                    26                     2
##        People & Blogs        Pets & Animals  Science & Technology
##                   132                     4                    17
##                 Shows                Sports              Trailers
##                    13                    11                     2
##       Travel & Events
##                     1
```

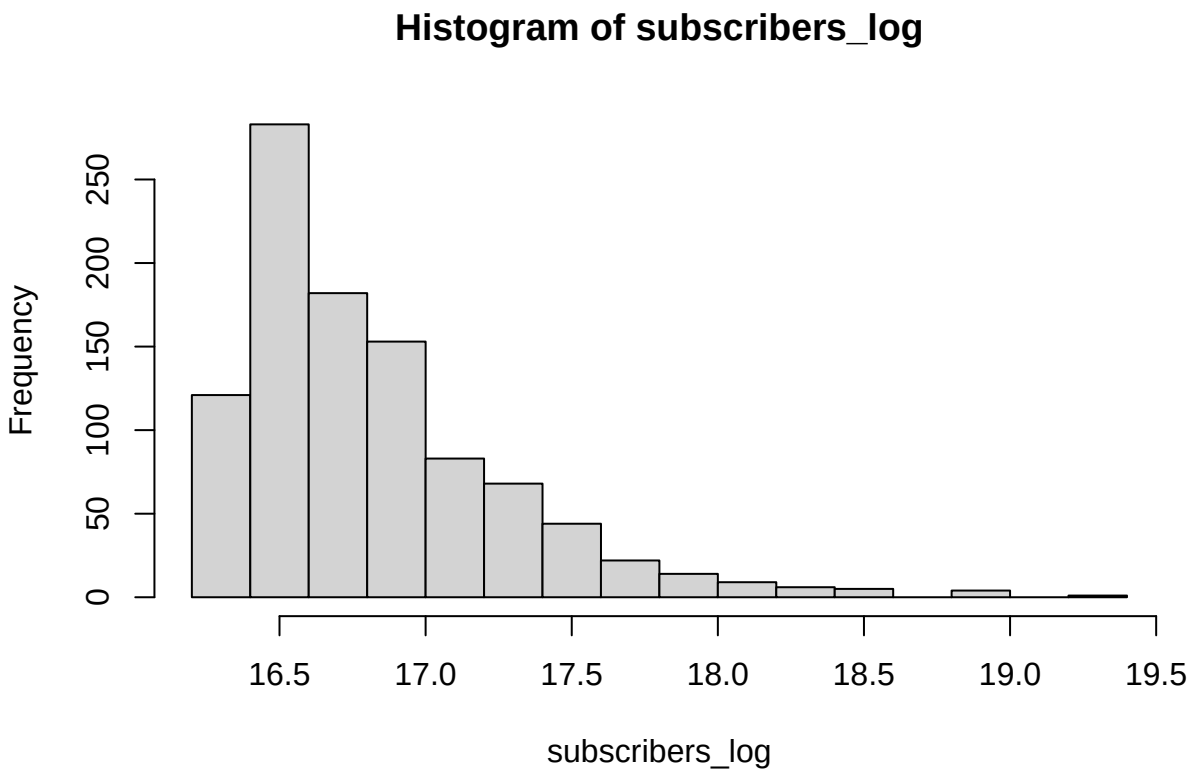
```r
# Transforming the subscribers variable
subscribers_log <- log(Global.YouTube.Statistics$subscribers)

#mentioning subscribers_log simply will give us the results of transformed values.
subscribers_log
```

```
##   [1] 19.31677 18.95131 18.92750 18.90311 18.88441 18.59463 18.53401 18.52504
##   [9] 18.47895 18.40962 18.38712 18.37986 18.35454 18.31310 18.28027 18.23435
##  [17] 18.19879 18.14097 18.13300 18.08661 18.08241 18.07112 18.03796 18.01271
##  [25] 18.00365 17.99909 17.98372 17.92638 17.90149 17.90149 17.89812 17.88283
##  [33] 17.87595 17.86903 17.86206 17.85681 17.84798 17.81554 17.80449 17.79519
##  [41] 17.79145 17.78391 17.78013 17.77059 17.68879 17.68463 17.67624 17.67413
##  [49] 17.66139 17.65711 17.65065 17.64632 17.64632 17.63322 17.62661 17.61548
##  [57] 17.61324 17.61100 17.60424 17.60424 17.60424 17.59286 17.59057 17.58827
##  [65] 17.58135 17.58135 17.56501 17.56266 17.56266 17.55080 17.53879 17.53637
##  [73] 17.52664 17.51928 17.51434 17.51186 17.49686 17.48928 17.48419 17.48419
##  [81] 17.48419 17.48163 17.47907 17.47650 17.46876 17.46357 17.46096 17.45835
##  [89] 17.45835 17.45835 17.45310 17.45046 17.44251 17.43985 17.43182 17.42643
##  [97] 17.42643 17.41829 17.41556 17.41556 17.41282 17.40733 17.40733 17.40457
## [105] 17.40180 17.39066 17.38504 17.38504 17.38222 17.38222 17.37656 17.37656
## [113] 17.36800 17.35936 17.35357 17.35066 17.34481 17.34187 17.34187 17.33597
## [121] 17.33597 17.33301 17.33301 17.33004 17.32706 17.32706 17.32706 17.32407
## [129] 17.32407 17.32107 17.30594 17.30289 17.30289 17.30289 17.29982 17.28748
## [137] 17.28437 17.28437 17.28437 17.28125 17.27812 17.27812 17.27498 17.27183
## [145] 17.27183 17.27183 17.27183 17.26867 17.26867 17.26232 17.26232 17.25593
## [153] 17.25593 17.25593 17.23977 17.23977 17.23977 17.23324 17.23324 17.22995
## [161] 17.22995 17.22995 17.22666 17.22335 17.22335 17.22335 17.22004 17.22004
## [169] 17.22004 17.22004 17.21671 17.21002 17.20328 17.19310 17.19310 17.18968
## [177] 17.18968 17.18968 17.18281 17.17935 17.16541 17.16190 17.16190 17.15837
## [185] 17.15837 17.15837 17.15483 17.15483 17.15128 17.14772 17.14055 17.14055
## [193] 17.13694 17.12970 17.12970 17.12605 17.12605 17.12605 17.12240 17.12240
## [201] 17.11504 17.11504 17.11504 17.11135 17.10764 17.10017 17.10017 17.10017
## [209] 17.09266 17.09266 17.08887 17.08887 17.08887 17.08508 17.08127 17.08127
## [217] 17.07745 17.07745 17.07361 17.07361 17.07361 17.06975 17.06589 17.06200
## [225] 17.06200 17.06200 17.05810 17.05810 17.05419 17.05419 17.05026 17.05026
## [233] 17.04631 17.04235 17.04235 17.04235 17.04235 17.04235 17.03838 17.03838
## [241] 17.03439 17.03439 17.02635 17.02635 17.02635 17.02635 17.02231 17.02231
## [249] 17.01826 17.01826 17.01418 17.01009 17.00599 17.00599 17.00186 17.00186
## [257] 16.99772 16.99772 16.99772 16.99772 16.99772 16.99772 16.99772 16.99356
## [265] 16.99356 16.99356 16.98939 16.98939 16.98939 16.98520 16.98520 16.98520
## [273] 16.98099 16.98099 16.98099 16.98099 16.97676 16.97676 16.97676 16.97676
## [281] 16.97676 16.97251 16.97251 16.96825 16.96825 16.96396 16.95966 16.95966
## [289] 16.95966 16.95534 16.95534 16.95534 16.95534 16.95534 16.95100 16.95100
## [297] 16.95100 16.95100 16.94665 16.94665 16.94665 16.94665 16.94227 16.93788
## [305] 16.93788 16.93346 16.93346 16.93346 16.93346 16.93346 16.93346 16.92903
```

```
## [313] 16.92903 16.92903 16.92457 16.92010 16.92010 16.91560 16.90655 16.90655
## [321] 16.90655 16.90200 16.90200 16.90200 16.90200 16.89742 16.89742 16.89742
## [329] 16.89742 16.89742 16.89742 16.89282 16.89282 16.88820 16.88820 16.88820
## [337] 16.88356 16.88356 16.88356 16.88356 16.87890 16.87890 16.87422 16.87422
## [345] 16.87422 16.87422 16.87422 16.87422 16.86951 16.86951 16.86478 16.86478
## [353] 16.86003 16.86003 16.86003 16.86003 16.85526 16.85526 16.85526 16.85526
## [361] 16.85526 16.85526 16.85046 16.85046 16.84564 16.84564 16.84564 16.84564
## [369] 16.84564 16.84080 16.84080 16.84080 16.83594 16.83594 16.83594 16.83105
## [377] 16.83105 16.83105 16.83105 16.83105 16.83105 16.82613 16.82613 16.82613
## [385] 16.82119 16.82119 16.82119 16.82119 16.82119 16.82119 16.82119 16.81623
## [393] 16.81623 16.81623 16.81623 16.81623 16.81623 16.81623 16.81623 16.81623
## [401] 16.81124 16.81124 16.81124 16.81124 16.81124 16.80623 16.80119 16.80119
## [409] 16.80119 16.79613 16.79613 16.79613 16.79613 16.79613 16.79104 16.79104
## [417] 16.79104 16.79104 16.79104 16.78593 16.78078 16.78078 16.78078 16.78078
## [425] 16.77562 16.77562 16.77562 16.77042 16.77042 16.76520 16.76520 16.76520
## [433] 16.76520 16.75995 16.75995 16.75995 16.75995 16.75995 16.75995 16.75467
## [441] 16.75467 16.75467 16.74937 16.74937 16.74937 16.74937 16.74937 16.74937
## [449] 16.74937 16.74403 16.74403 16.73867 16.73867 16.73867 16.73867 16.73328
## [457] 16.73328 16.73328 16.73328 16.73328 16.72786 16.72786 16.72786 16.72786
## [465] 16.72786 16.72241 16.72241 16.72241 16.72241 16.71693 16.71693 16.71142
## [473] 16.71142 16.71142 16.71142 16.71142 16.71142 16.70588 16.70588 16.70588
## [481] 16.70588 16.70588 16.70588 16.70588 16.70031 16.70031 16.70031 16.70031
## [489] 16.70031 16.70031 16.70031 16.70031 16.70031 16.69471 16.69471 16.68908
## [497] 16.68908 16.68908 16.68908 16.68908 16.68908 16.68908 16.68908 16.68908
## [505] 16.68908 16.68341 16.68341 16.68341 16.68341 16.68341 16.68341 16.67771
## [513] 16.67771 16.67771 16.67771 16.67198 16.67198 16.67198 16.67198 16.66622
## [521] 16.66622 16.66622 16.66622 16.66042 16.66042 16.66042 16.66042 16.66042
## [529] 16.66042 16.65459 16.65459 16.65459 16.64872 16.64872 16.64872 16.64872
## [537] 16.64282 16.64282 16.64282 16.64282 16.64282 16.64282 16.64282 16.63689
## [545] 16.63689 16.63689 16.63689 16.63689 16.63092 16.63092 16.63092 16.63092
## [553] 16.63092 16.62491 16.62491 16.62491 16.62491 16.62491 16.62491 16.62491
## [561] 16.62491 16.61887 16.61887 16.61887 16.61887 16.61279 16.61279 16.61279
## [569] 16.61279 16.61279 16.61279 16.61279 16.60668 16.60668 16.60668 16.60668
## [577] 16.60668 16.60668 16.60668 16.60668 16.60668 16.60052 16.60052 16.60052
## [585] 16.60052 16.60052 16.60052 16.60052 16.60052 16.60052 16.60052 16.59433
## [593] 16.59433 16.59433 16.59433 16.59433 16.59433 16.59433 16.59433 16.59433
## [601] 16.59433 16.58810 16.58810 16.58810 16.58810 16.58810 16.58183 16.58183
## [609] 16.58183 16.58183 16.58183 16.58183 16.58183 16.57552 16.57552 16.57552
## [617] 16.57552 16.56917 16.56917 16.56917 16.56917 16.56917 16.56917 16.56278
## [625] 16.56278 16.56278 16.56278 16.56278 16.55635 16.55635 16.55635 16.55635
## [633] 16.55635 16.55635 16.55635 16.55635 16.55635 16.55635 16.54988 16.54988
## [641] 16.54988 16.54988 16.54988 16.54988 16.54988 16.54988 16.54336 16.54336
## [649] 16.54336 16.54336 16.54336 16.53681 16.53681 16.53681 16.53681 16.53681
## [657] 16.53681 16.53681 16.53681 16.53681 16.53021 16.53021 16.53021 16.53021
## [665] 16.53021 16.53021 16.53021 16.53021 16.53021 16.53021 16.53021 16.53021
## [673] 16.53021 16.53021 16.53021 16.53021 16.52356 16.52356 16.52356 16.52356
## [681] 16.52356 16.52356 16.52356 16.52356 16.52356 16.52356 16.52356 16.52356
## [689] 16.52356 16.52356 16.52356 16.52356 16.52356 16.51687 16.51687 16.51687
## [697] 16.51687 16.51687 16.51687 16.51687 16.51687 16.51687 16.51687 16.51014
## [705] 16.51014 16.51014 16.51014 16.51014 16.51014 16.51014 16.51014 16.51014
## [713] 16.51014 16.51014 16.50336 16.50336 16.50336 16.50336 16.50336 16.50336
## [721] 16.50336 16.50336 16.50336 16.50336 16.50336 16.49653 16.49653 16.49653
## [729] 16.49653 16.49653 16.49653 16.49653 16.49653 16.49653 16.49653 16.48966
## [737] 16.48966 16.48966 16.48966 16.48966 16.48966 16.48966 16.48966 16.48966
```

```
## [745] 16.48966 16.48966 16.48966 16.48966 16.48966 16.48966 16.48966 16.48966
## [753] 16.48274 16.48274 16.48274 16.48274 16.48274 16.48274 16.48274 16.48274
## [761] 16.48274 16.48274 16.48274 16.48274 16.48274 16.48274 16.48274 16.48274
## [769] 16.47577 16.47577 16.46875 16.46875 16.46875 16.46875 16.46875 16.46875
## [777] 16.46875 16.46875 16.46875 16.46875 16.46875 16.46875 16.46169 16.46169
## [785] 16.46169 16.46169 16.46169 16.46169 16.46169 16.46169 16.46169 16.46169
## [793] 16.45457 16.45457 16.45457 16.45457 16.45457 16.45457 16.45457 16.45457
## [801] 16.45457 16.45457 16.44740 16.44740 16.44740 16.44740 16.44740 16.44740
## [809] 16.44740 16.44740 16.44740 16.44740 16.44740 16.44018 16.44018 16.44018
## [817] 16.44018 16.44018 16.44018 16.44018 16.44018 16.43291 16.43291 16.43291
## [825] 16.43291 16.43291 16.43291 16.43291 16.43291 16.43291 16.43291 16.42558
## [833] 16.42558 16.42558 16.42558 16.42558 16.42558 16.42558 16.41820 16.41820
## [841] 16.41820 16.41820 16.41820 16.41820 16.41820 16.41820 16.41820 16.41820
## [849] 16.41820 16.41077 16.41077 16.41077 16.41077 16.41077 16.41077 16.41077
## [857] 16.41077 16.41077 16.41077 16.41077 16.40327 16.40327 16.40327 16.40327
## [865] 16.40327 16.40327 16.40327 16.40327 16.40327 16.40327 16.40327 16.40327
## [873] 16.40327 16.40327 16.39573 16.39573 16.39573 16.39573 16.39573 16.39573
## [881] 16.39573 16.39573 16.39573 16.39573 16.39573 16.39573 16.38812 16.38812
## [889] 16.38812 16.38812 16.38812 16.38812 16.38812 16.38812 16.38812 16.38812
## [897] 16.38812 16.38812 16.38812 16.38812 16.38046 16.38046 16.38046 16.38046
## [905] 16.38046 16.38046 16.38046 16.38046 16.38046 16.38046 16.38046 16.37274
## [913] 16.37274 16.37274 16.37274 16.37274 16.37274 16.37274 16.37274 16.37274
## [921] 16.37274 16.37274 16.36496 16.36496 16.36496 16.36496 16.36496 16.36496
## [929] 16.36496 16.36496 16.35711 16.35711 16.35711 16.35711 16.35711 16.35711
## [937] 16.35711 16.35711 16.35711 16.35711 16.35711 16.35711 16.35711 16.34921
## [945] 16.34921 16.34921 16.34921 16.34921 16.34124 16.34124 16.34124 16.34124
## [953] 16.34124 16.34124 16.34124 16.34124 16.34124 16.34124 16.34124 16.34124
## [961] 16.34124 16.34124 16.34124 16.34124 16.34124 16.34124 16.34124 16.34124
## [969] 16.34124 16.34124 16.33321 16.33321 16.33321 16.33321 16.33321 16.33321
## [977] 16.33321 16.33321 16.33321 16.33321 16.33321 16.33321 16.33321 16.33321
## [985] 16.33321 16.33321 16.33321 16.33321 16.33321 16.33321 16.32511 16.32511
## [993] 16.32511 16.32511 16.32511
```

```r
# The distribution of the transformed variable using histogram.
hist(subscribers_log)
```




```
#The output of the code will be a histogram of the transformed variable.
#The histogram should be more normally distributed than the original variable.
```

```
#To knit as pdf and for scatter plotting installing packages and saving
#it to path.
install.packages("ggplot2", repos = "https://cran.rstudio.com")
```

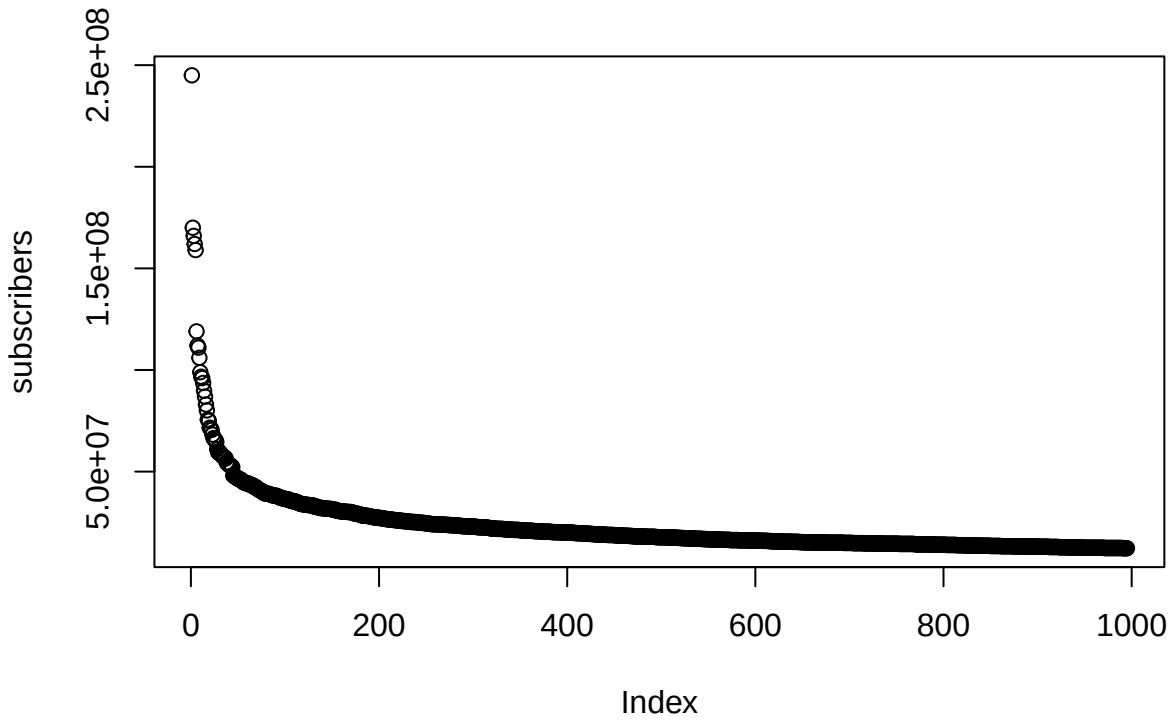
```
## Installing package into 'C:/Users/drpra/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\drpra\AppData\Local\Temp\Rtmp8epGPM\downloaded_packages
```

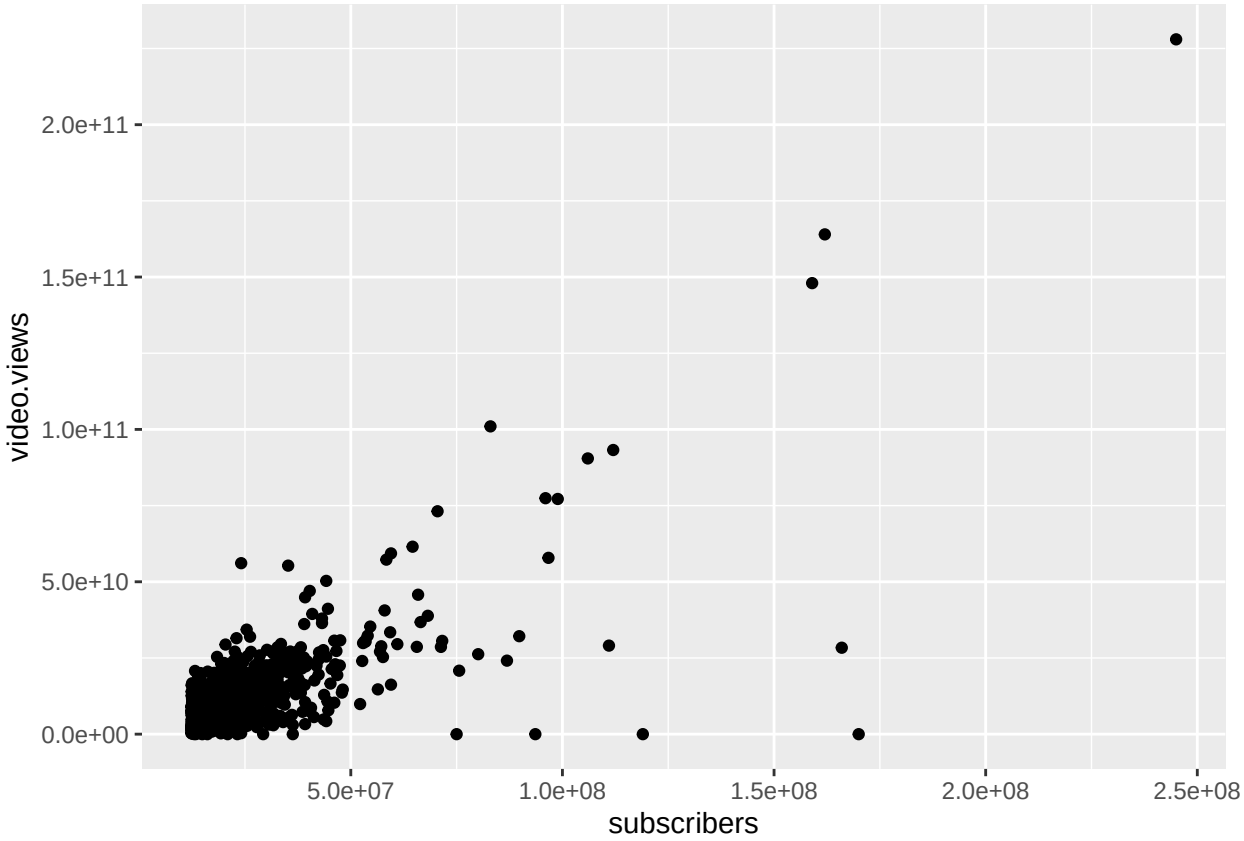
```
library(ggplot2)

writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', con = "~/.Renviron")
```

```
#Plotting technique using plot() function.

# Here i'm selecting the subscribers variable
subscribers <- Global.YouTube.Statistics$subscribers
```

```r
# Plotting the subscribers variable
plot(subscribers)
```




```r
# Plot the subscribers variable against the video.views variable
ggplot(Global.YouTube.Statistics, aes(x = subscribers, y = video.views)) +
  geom_point()
```




```
#A scatter plot is a type of plot that shows the relationship between two
#variables. The x-axis variable is the independent variable and the
#y-axis variable is the dependent variable. In this case, the x-axis
#variable is the subscribers variable and the y-axis variable is the video.views variable.

#The scatter plot shows that there is a positive correlation between
#the number of subscribers and the number of video views. This means
#that as the number of subscribers increases, the number of video views
#also tends to increase.
```