ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

POST POPULARITY IDENTIFICATION AND ESTIMATION FROM MASHABLE DATA

DATASET NR : 25

MSc. BUSINESS ANALYTICS

STATISTICS FOR BUSINESS ANALYTICS I

BURCIN SARAC

2018-2019 / FULL-TIME

F2821825

Table of Contents

1. Introduction - 3 -

2. Exploratory Data Analysis and Pairwise Comparisons - 3 -

3. Feature Scaling and Model Selection - 5 -

4. Compare Prediction Performance in Test Data via Models - 15 -

5. Conclusion - 17 -

I. Reference pages of answers of the questions - 20 -

II. Related Figures about Distribution of Variables Individually and Relation Btw Response Variable- 22 -

III. R Codes of the project - 36 -

Table of Figures

Figure 1 - 4 -
Figure 2 - 5 -
Figure 3 - 8 -
Figure 4 - 8 -
Figure 5 - 11 -
Figure 6 - 11 -
Figure 7 - 13 -
Figure 8 - 15 -

Tables

Table 1 - 7 -
Table 2 - 8 -
Table 3 - 8 -
Table 4 - 10 -
Table 5 - 10 -
Table 6 - 13 -
Table 7 - 14 -

1. Introduction

Online news consumption, in terms of sharing in a social media account or a blog kind webpage, is getting popular day by day thanks to the increasing number of social media users and developing technologies through smartphones, tablet computers etc. And one possible way to measure the popularity of an online news is by its share numbers. This measure lead people from various sectors to predict future possible share numbers of a news beforehand. Predicting such popularity is valuable for nearly everyone no matter why the reason is. Because of popularity of this topic, there are many public datasets available publicly.

The data of this report is a public dataset which includes shares values of online news with their urls and some of its characteristics in terms of statistical variables of Mashable (www.mashable.com) website. And this report aims to identify popular news posts according to the given variables in a dataset and create models and select one of them as a final model to try to estimate a post's possible share numbers.

2. Exploratory Data Analysis and Pairwise Comparisons

The train dataset, which is going to help to fit a model for future predictions, includes 3000 observations of 62 variables at total. And the test dataset includes 10.000 observations of 62 variables and all of the variables and their order are matched with the train dataset. Additionally, none of them includes same data as the other.

One of the variables called "shares" is going to be the response variable of the model. This means, it is the target variable of the model with the help of other variables called predictors, in other words, all or some of other variables used for estimating target variable. In this case, it is planned to perform and fit a model with the train dataset with using true share results and estimations. Afterwards, the best fitted model is going to be used by estimating shares in test dataset. And it will have compared again with the given "shares" values at the test data, for evaluating model's performance.

After drilling down to data more, firstly it should be necessary to mention that, the datasets have not any NA(empty) rows, because of that it can be assumed there is not any missing data in the dataset.

It seems from dataset that, there are 14 factor variables, and they all are all dummy variables already converted from factor variables days and data channel, which is necessary for using them as predictors in multiple linear regression. Linear regression algorithm requires quantitative variables or factor variables only with binary factors, like Boolean or Dummy variable. Otherwise, it is needed to use another algorithm to predict from categorical predictors.

The response variable “shares” has 110.200 at maximum and the highest five variables are 102.200, 72600, 73.100, 57.400 respectively in train dataset. And it has 23 at minimum. The mean of “shares” data is 2955, which located higher than 3th quartile, this means the distribution of this data is skewed right. The distribution and the skewness can be seen from the “Figure 1” below as well. This means, it may needed to be normalized or transformed for fitting a valid model. There are several ways to handle this problem, like move all the share data between -1 to 1 with dividing all of 3000 shares rows to the max “shares” data (110200) or take logarithmic version of it for re-scaling this variable. This is also a necessary step for all predictors. For deciding which variables need this transformation, it is necessary to check if the variables distributed normally or not.

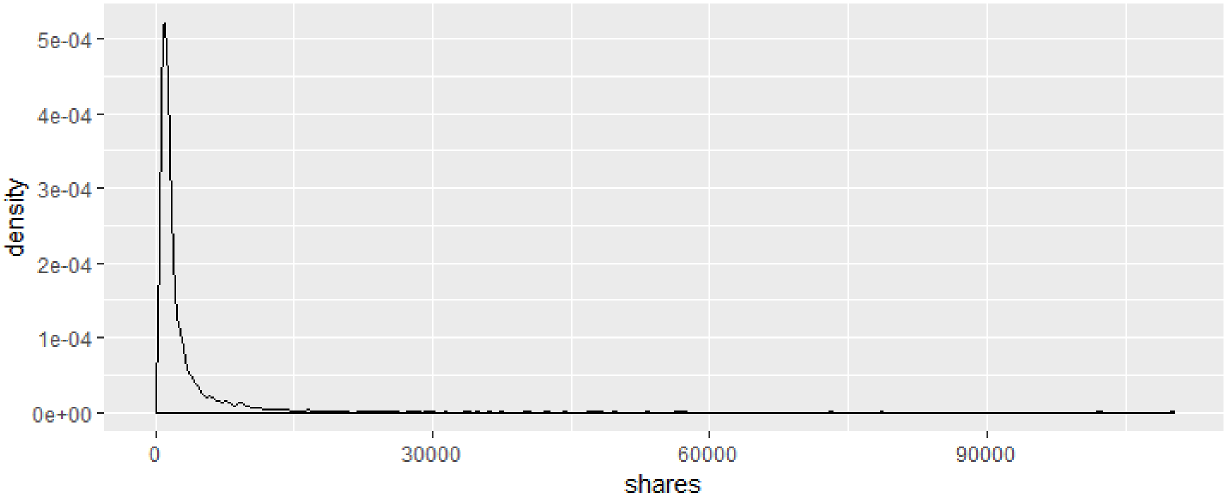


Figure 1

Moreover, the "data_channel_is_lifestyle", "data_channel_is_entertainment","data_channel_is_bus","data_channel_is_socmed","data_channel_is_tech", "data_channel_is_world","weekday_is_monday","weekday_is_tuesday","weekday_is_wednesday", "weekday_is_thursday","weekday_is_friday","weekday_is_saturday","weekday_is_sunday", "is_weekend" variables temporarily separated from the whole dataset and identified as logical variables, since R can calculate True

False data as 1 and 0, it will help the trained models to calculate these variables as factors. The distribution of these variables checked with a pie chart below (Figure 2)

On the other hand, the correlation between response variable “shares” and predictors was checked and according to the provided list “num_href”, “kw_max_avg” and “kw_avg_avg” predictors have highest positive correlation rates with 0.12, 0.12 and 0.17 respectively. Correlation refers to the effecting power of a predictor on the response variable.

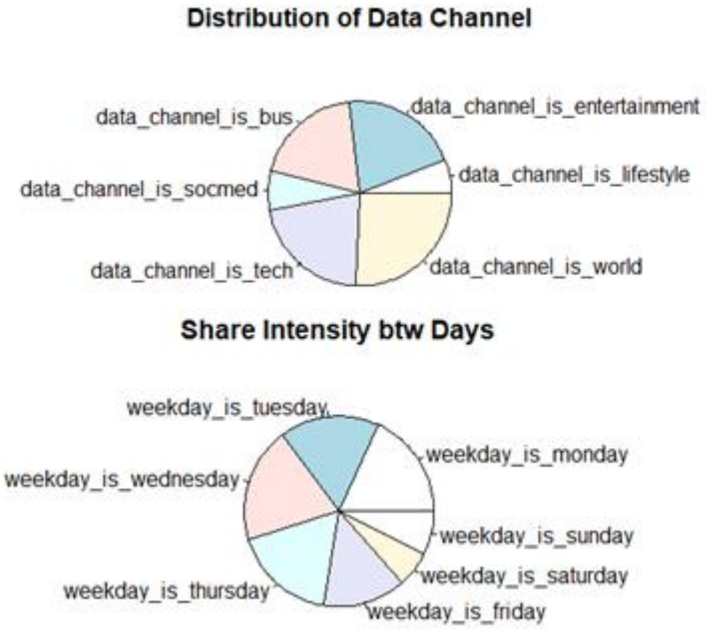


Figure 2

And the remaining variables defined as integers.All predictors checked via related statistical tests with the helps of R(tests : Lilliefors (Kolmogorov-Smirnov) and Shapiro-Wilk normality tests), but it determined that none of them distributed normally, like “shares” variable.

3. Feature Scaling and Model Selection

For this report, always the whole dataset combined with train and test data were transformed together, because after a model created, it needs to be used in test data in a same formula, because of this, all permanent changes/transformations made to the entire dataset. And after changing, the train and test dataset separated again.

First of all, the three non-explanatory variables directly omitted at the beginning stage to avoid misleading linear model. So first column (includes some random numbers, which may be the row numbers of data picked from the entire dataset), “URL” and “timedelta” variables dropped at the beginning.

After that, it is aimed that the multi-collinearity problem between variables determined and dropped from the dataset if there is any. Although multi-collinearity does not reduce the predictive power or reliability of the model as a whole, at least within the sample data set; but it affects calculations regarding individual predictors. So, it checked with the “alias()” formula with the response variable and all predictors, which gives that, all the day variables are dependent to each other as expected. To handle this multi-collinearity, firstly a “is_weekday” variable created as a dummy variable and all the other day variables dropped (weekday_is_monday, weekday_is_tuesday, weekday_is_wednesday, weekday_is_thursday, weekday_is_friday, weekday_is_saturday, weekday_is_sunday,is_weekend). And it checked with its correlation with response variable, however it shown that there were a predict ability loss occurred from deleting variables. Because of this, the variables brought back as the beginning and only the ones with higher correlation kept (weekday_is_tuesday, weekday_is_friday ,is_weekend) and all others dropped and tested again. This time it worked better and multi-collinearity problem solved as well.

After that, the highest correlation values filtered between response and the predictors. And then there is a raw model created manually by hand according to the correlation data. But the model’s predictors was not significant at the beginning. After some transforming and dropping a couple of variables, model became better according to its adjusted R^2 value. This value gives information to the tester about the accuracy of a model. Of course it should not taken into account as only by itself to determine performance of a model, because it may wrongly calculated by the software and mislead the owner, but it would be useful to quickly compare with other possible models. Normally residuals should be taken into account for assessing models’ performance as well. This value is a measure of the differences between values predicted by a model and actual observed values, in other words it estimates the standard deviation of the model, so, the smaller the residual standard errors, the better model fits.

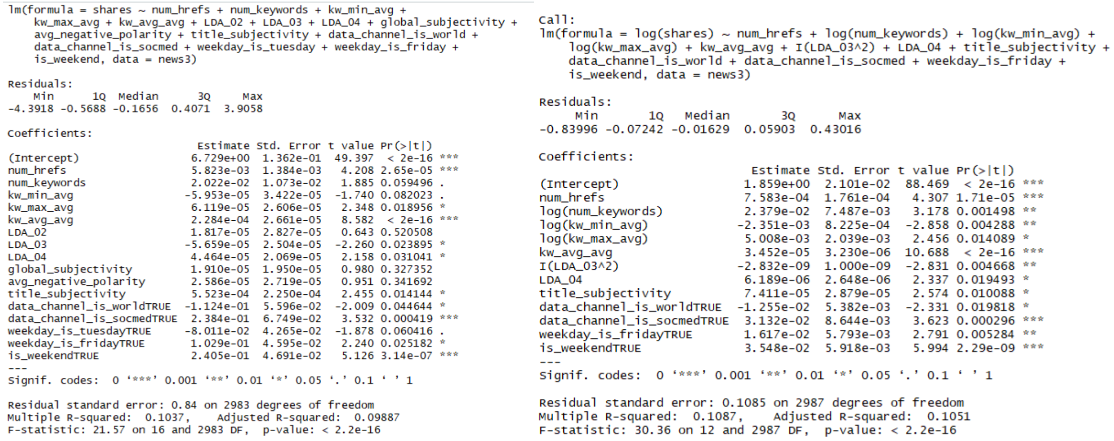
On the other hand, after a model selected, it also need to meet all the assumptions of a valid model. The assumptions checked from residuals and the main assumptions are;

- Normality – distribution of residuals expected to be normal
- Linearity - relationship between dependent and independent variables expected to be linear

- Homoscedasticity – variance of errors expected to be constant
- Independence of Errors- no correlation between errors is expected

Collinearity also would be a problem for selected model. Collinearity basically means two variables carry similar data for prediction. To prevent one of the variables should dropped from model.

If any of these assumptions is violated then the forecasts, confidence intervals, and scientific insights yielded by a regression model may be (at best) inefficient or (at worst) seriously biased or misleading.



```
lm(formula = shares ~ num_hrefs + num_keywords + kw_min_avg +
    kw_max_avg + kw_avg_avg + LDA_02 + LDA_03 + LDA_04 + global_subjectivity +
    avg_negative_polarity + title_subjectivity + data_channel_is_world +
    data_channel_is_socmed + weekday_is_tuesday + weekday_is_friday +
    is_weekend, data = news3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3918 -0.5688 -0.1656  0.4071  3.9058

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.729e+00  1.362e-01  49.397  < 2e-16 ***
num_hrefs                   5.823e-03  1.384e-03   4.208 2.65e-05 ***
num_keywords                2.022e-02  1.073e-02   1.885 0.059496 .
kw_min_avg                 -5.953e-05  3.422e-05  -1.740 0.082023 .
kw_max_avg                  6.119e-05  2.606e-05   2.348 0.018956 *
kw_avg_avg                  2.284e-04  2.661e-05   8.582  < 2e-16 ***
LDA_02                      1.817e-05  2.827e-05   0.643 0.520508
LDA_03                     -5.659e-05  2.504e-05  -2.260 0.023895 *
LDA_04                      4.464e-05  2.069e-05   2.158 0.031041 *
global_subjectivity         1.910e-05  1.950e-05   0.980 0.327352
avg_negative_polarity       2.586e-05  2.719e-05   0.951 0.341692
title_subjectivity          5.523e-04  2.250e-04   2.455 0.014144 *
data_channel_is_worldTRUE  -1.124e-01  5.596e-02  -2.009 0.044644 *
data_channel_is_socmedTRUE  2.384e-01  6.749e-02   3.532 0.000419 ***
weekday_is_tuesdayTRUE     -8.011e-02  4.265e-02  -1.878 0.060416 .
weekday_is_fridayTRUE       1.029e-01  4.595e-02   2.240 0.025182 *
is_weekendTRUE              2.405e-01  4.691e-02   5.126 3.14e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.84 on 2983 degrees of freedom
Multiple R-squared:  0.1037,    Adjusted R-squared:  0.09887
F-statistic: 21.57 on 16 and 2983 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = log(shares) ~ num_hrefs + log(num_keywords) + log(kw_min_avg) +
    log(kw_max_avg) + kw_avg_avg + I(LDA_03^2) + LDA_04 + title_subjectivity +
    data_channel_is_world + data_channel_is_socmed + weekday_is_friday +
    is_weekend, data = news3)

Residuals:
     Min       1Q   Median       3Q      Max
-0.83996 -0.07242 -0.01629  0.05903  0.43016

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.859e+00  2.101e-02  88.469  < 2e-16 ***
num_hrefs                   7.583e-04  1.761e-04   4.307 1.71e-05 ***
log(num_keywords)           2.379e-02  7.487e-03   3.178 0.001498 **
log(kw_min_avg)            -2.351e-03  8.225e-04  -2.858 0.004288 **
log(kw_max_avg)             5.008e-03  2.039e-03   2.456 0.014089 *
kw_avg_avg                  3.452e-05  3.230e-06  10.688  < 2e-16 ***
I(LDA_03^2)                -2.832e-09  1.000e-09  -2.831 0.004668 **
LDA_04                      6.189e-06  2.648e-06   2.337 0.019493 *
title_subjectivity          7.411e-05  2.879e-05   2.574 0.010088 *
data_channel_is_worldTRUE  -1.255e-02  5.382e-03  -2.331 0.019818 *
data_channel_is_socmedTRUE  3.132e-02  8.644e-03   3.623 0.000296 ***
weekday_is_fridayTRUE       1.617e-02  5.793e-03   2.791 0.005284 **
is_weekendTRUE              3.548e-02  5.918e-03   5.994 2.29e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1085 on 2987 degrees of freedom
Multiple R-squared:  0.1087,    Adjusted R-squared:  0.1051
F-statistic: 30.36 on 12 and 2987 DF,  p-value: < 2.2e-16
```

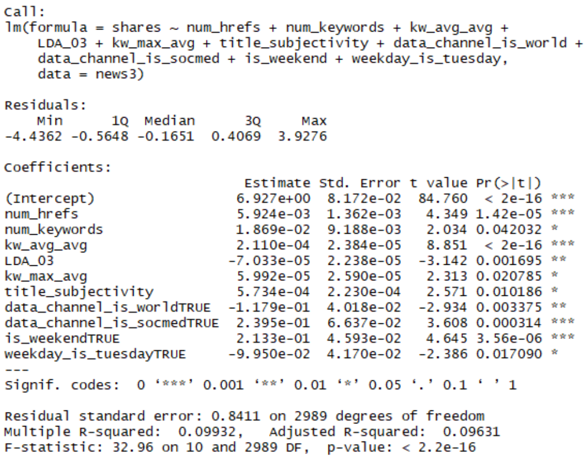
Table 1

With the aim of creating a valid model, some variables dropped and a raw model created by manually selecting the predictors. The model called “modelraw” designed(Table 1) and R^2 value was 9.88%, and residual standard error value was 0.84. The model can be written like this;

Log(Shares) = 6.927 + 0.0059*num_hrefs + 0.0186*num_keywords + 0.00021*kw_avg_avg -0.000073*LDA_03 + 0.0000599*kw_max_avg + 0.00057*title_subjectivity - 0.1179*data_channel_is_world + 0.2395*data_channel_is_socmed + 0.2133*is_weekend – 0.0995*weekday_is_Tuesday + Ɛ

Ɛ ~ N (0, 0.841^2)

But not all of the assumptions had met with this model. *Both the normality and the homoscedasticity assumptions are rejected but the linearity assumption is violated (Tukey’s* *p=0.001<0.05, Non-constant Variance Score p=~0<0.05, Levene's p=~0<0.05); see Table 2,3 for details & Figure 3 & 4 for visualizations of residuals.*



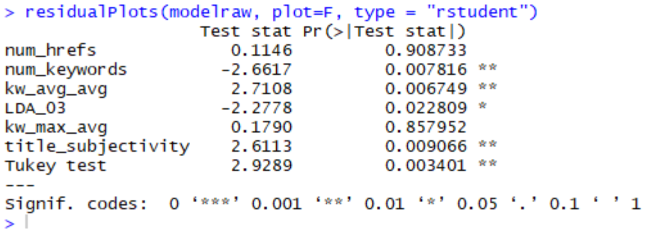
```
Call:
lm(formula = shares ~ num_hrefs + num_keywords + kw_avg_avg +
    LDA_03 + kw_max_avg + title_subjectivity + data_channel_is_world +
    data_channel_is_socmed + is_weekend + weekday_is_tuesday,
    data = news3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4362 -0.5648 -0.1651  0.4069  3.9276

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.927e+00  8.172e-02  84.760  < 2e-16 ***
num_hrefs                  5.924e-03  1.362e-03   4.349 1.42e-05 ***
num_keywords               1.869e-02  9.188e-03   2.034 0.042032 *
kw_avg_avg                 2.110e-04  2.384e-05   8.851  < 2e-16 ***
LDA_03                    -7.033e-05  2.238e-05  -3.142 0.001695 **
kw_max_avg                 5.992e-05  2.590e-05   2.313 0.020785 *
title_subjectivity         5.734e-04  2.230e-04   2.571 0.010186 *
data_channel_is_worldTRUE -1.179e-01  4.018e-02  -2.934 0.003375 **
data_channel_is_socmedTRUE 2.395e-01  6.637e-02   3.608 0.000314 ***
is_weekendTRUE             2.133e-01  4.593e-02   4.645 3.56e-06 ***
weekday_is_tuesdayTRUE    -9.950e-02  4.170e-02  -2.386 0.017090 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8411 on 2989 degrees of freedom
Multiple R-squared:  0.09932,   Adjusted R-squared:  0.09631
F-statistic: 32.96 on 10 and 2989 DF,  p-value: < 2.2e-16
```

Table 2



```
> residualPlots(modelraw, plot=F, type = "rstudent")
                   Test stat Pr(>|Test stat|)
num_hrefs             0.1146         0.908733
num_keywords         -2.6617         0.007816 **
kw_avg_avg            2.7108         0.006749 **
LDA_03               -2.2778         0.022809 *
kw_max_avg            0.1790         0.857952
title_subjectivity    2.6113         0.009066 **
Tukey test            2.9289         0.003401 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Table 3

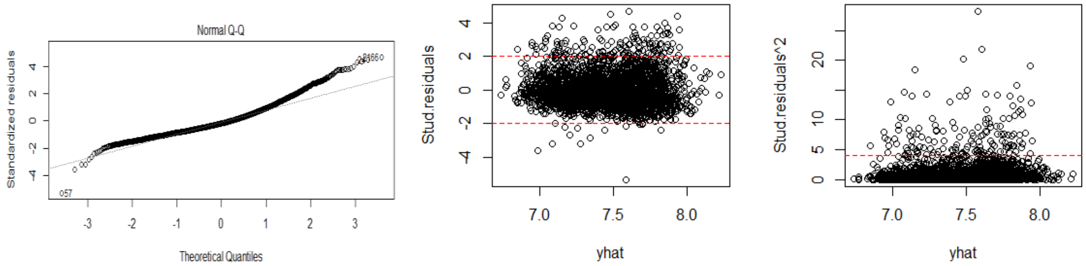


Figure 3

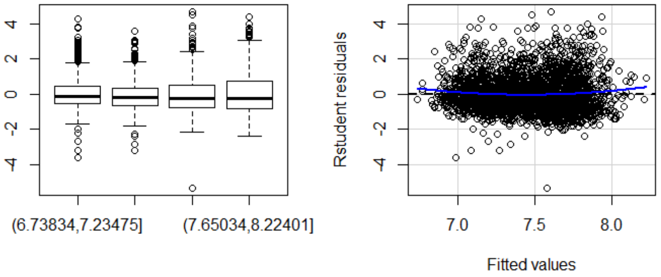


Figure 3

After a manual model created, two more valid models created with using stepwise and Lasso techniques. At the stepwise stage, a full model with all predictors and a null model which only includes intercept but not any predictor, created and all stepwise methods (“forward”, “backward”, “both”) tried and at last a model created with both method, selected as a base model and after selection of model, the variables tried to turned into significant predictors as an interpreter to the model. After several trials, decided a model called “**modelst2**” with 20 variables ;

log(shares) = 7.205+ 2.496e-08*(n_tokens_content^2), 0.00715*num_hrefs - 7.218e-05*average_token_length -0.228*data_channel_is_lifestyleTRUE - 0.332* data_channel_is_entertainmentTRUE -0.304*data_channel_is_busTRUE - 0.191*data_channel_is_world 0.001*kw_min_min - 4.194e-05*kw_avg_min - 9.175e-07*kw_min_max 7.035e-05*kw_max_avg + 2.135e-04*kw_avg_avg + 6.719e-07*self_reference_max_shares 0.116*weekday_is_friday + 0.265*is_weekend + 4.448e-15*(LDA_00^4) - 7.213e-05*LDA_03 -0.0059*max_positive_polarity + 8.950e-04*title_subjectivity + 8.811e-04*abs_title_subjectivity + Ɛ

Ɛ ~ N (0, 0.8233^2)

Model created based on Stepwise method: Both the normality and the homoscedasticity assumptions are not rejected and the linearity assumption met (Tukey’s p=0.12>0.05, Non-constant Variance Score p=~0<0.05, Levene’s p=~0<0.05); see Table 4 for model details & Figure 5 for visualizations of residuals.

After deciding this model, all assumptions were met and it resulted with 13.42% Adjusted R^2 , which means the model has 13% ability to explain the variability of the number of shares.

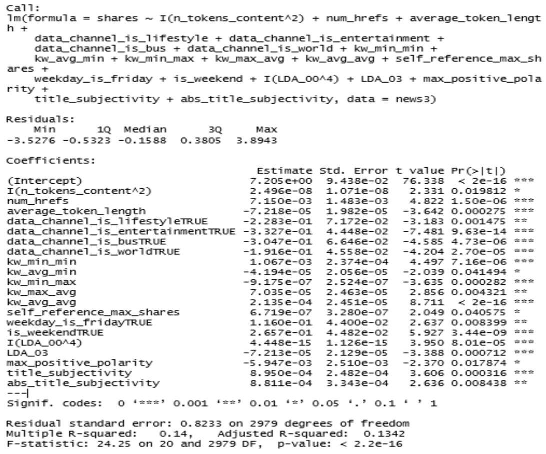
Addition to this, as mentioned above, another model created with using Lasso Regression technique for predictor selection. However same as stepwise method, in this method also some of selected predictors was not significant and the assumptions did not met. So again many variable transformations and selections were tried and a final model created, which also met with the assumptions for model validation. The created model called “**modellasso**” is;

Log(shares) = 7.057 + 7.334e-03*num_hrefs - 7.169e-05*average_token_length – 0.239*data_channel_is_entertainmentTRUE -+ 0.2518*data_channel_is_socmedTRUE + 16.96*data_channel_is_techTRUE + 8.666e-04*kw_min_min - 9.410e-07 *kw_min_max +

2.561e-08*(kw_max_avg^2 + 2.201e-04 *kw_avg_avg - 0.0909*weekday_is_tuesdayTRUE + 0.2288*is_weekendTRUE - 2.690e-07*(rate_negative_words^(2)) + 6.992e-13*(title_subjectivity^5) + Ɛ

Ɛ ~ N (0, 0.8275^2)

Model created based on Lasso method: Both the normality and the homoscedasticity assumptions are not rejected and the linearity assumption met (Tukey's p=0.19>0.05, Non-constant Variance Score p=~0<0.05, Levene's p=~0<0.05); see Table 5 for details & Figure 6 for visualizations of residuals.

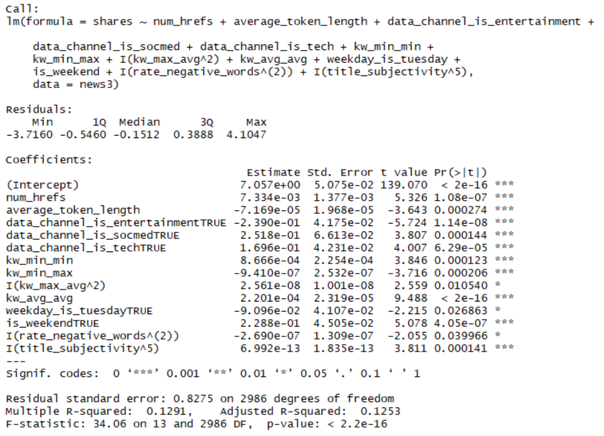


```
Call:
lm(formula = shares ~ I(n_tokens_content^2) + num_hrefs + average_token_lengt
h +
    data_channel_is_lifestyle + data_channel_is_entertainment +
    data_channel_is_bus + data_channel_is_world + kw_min_min +
    kw_avg_min + kw_min_max + kw_max_avg + kw_avg_avg + self_reference_max_sh
ares +
    weekday_is_friday + is_weekend + I(LDA_00^4) + LDA_03 + max_positive_pola
rity +
    title_subjectivity + abs_title_subjectivity, data = news3)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5276 -0.5323 -0.1588  0.3805  3.8943

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    7.205e+00  9.438e-02  76.338  < 2e-16 ***
I(n_tokens_content^2)          2.496e-08  1.071e-08   2.331 0.019812 *
num_hrefs                      7.150e-03  1.483e-03   4.822 1.50e-06 ***
average_token_length          -7.218e-05  1.982e-05  -3.642 0.000275 ***
data_channel_is_lifestyleTRUE -2.283e-01  7.172e-02  -3.183 0.001475 **
data_channel_is_entertainmentTRUE -3.327e-01  4.448e-02  -7.481 9.63e-14 ***
data_channel_is_busTRUE       -3.047e-01  6.646e-02  -4.585 4.73e-06 ***
data_channel_is_worldTRUE     -1.916e-01  4.558e-02  -4.204 2.70e-05 ***
kw_min_min                     1.067e-03  2.374e-04   4.497 7.16e-06 ***
kw_avg_min                    -4.194e-05  2.056e-05  -2.039 0.041494 *
kw_min_max                    -9.175e-07  2.524e-07  -3.635 0.000282 ***
kw_max_avg                     7.035e-05  2.463e-05   2.856 0.004321 **
kw_avg_avg                     2.135e-04  2.451e-05   8.711  < 2e-16 ***
self_reference_max_shares      6.719e-07  3.280e-07   2.049 0.040575 *
weekday_is_fridayTRUE          1.160e-01  4.400e-02   2.637 0.008399 **
is_weekendTRUE                 2.657e-01  4.482e-02   5.927 3.44e-09 ***
I(LDA_00^4)                    4.448e-15  1.126e-15   3.950 8.01e-05 ***
LDA_03                        -7.213e-05  2.129e-05  -3.388 0.000712 ***
max_positive_polarity         -5.947e-03  2.510e-03  -2.370 0.017874 *
title_subjectivity             8.950e-04  2.482e-04   3.606 0.000316 ***
abs_title_subjectivity         8.811e-04  3.343e-04   2.636 0.008438 **
---|
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8233 on 2979 degrees of freedom
Multiple R-squared:   0.14,	Adjusted R-squared:  0.1342
F-statistic: 24.25 on 20 and 2979 DF,  p-value: < 2.2e-16
```



```
Call:
lm(formula = shares ~ num_hrefs + average_token_length + data_channel_is_entertainment +

    data_channel_is_socmed + data_channel_is_tech + kw_min_min +
    kw_min_max + I(kw_max_avg^2) + kw_avg_avg + weekday_is_tuesday +
    is_weekend + I(rate_negative_words^(2)) + I(title_subjectivity^5),
    data = news3)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7160 -0.5460 -0.1512  0.3888  4.1047

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                       7.057e+00  5.075e-02 139.070  < 2e-16 ***
num_hrefs                         7.334e-03  1.377e-03   5.326 1.08e-07 ***
average_token_length             -7.169e-05  1.968e-05  -3.643 0.000274 ***
data_channel_is_entertainmentTRUE -2.390e-01  4.175e-02  -5.724 1.14e-08 ***
data_channel_is_socmedTRUE        2.518e-01  6.613e-02   3.807 0.000144 ***
data_channel_is_techTRUE          1.696e-01  4.231e-02   4.007 6.29e-05 ***
kw_min_min                        8.666e-04  2.254e-04   3.846 0.000123 ***
kw_min_max                       -9.410e-07  2.532e-07  -3.716 0.000206 ***
I(kw_max_avg^2)                   2.561e-08  1.001e-08   2.559 0.010540 *
kw_avg_avg                        2.201e-04  2.319e-05   9.488  < 2e-16 ***
weekday_is_tuesdayTRUE           -9.096e-02  4.107e-02  -2.215 0.026863 *
is_weekendTRUE                    2.288e-01  4.505e-02   5.078 4.05e-07 ***
I(rate_negative_words^(2))       -2.690e-07  1.309e-07  -2.055 0.039966 *
I(title_subjectivity^5)           6.992e-13  1.835e-13   3.811 0.000141 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8275 on 2986 degrees of freedom
Multiple R-squared:  0.1291,	Adjusted R-squared:  0.1253
F-statistic: 34.06 on 13 and 2986 DF,  p-value: < 2.2e-16
```

Table 4 & 5

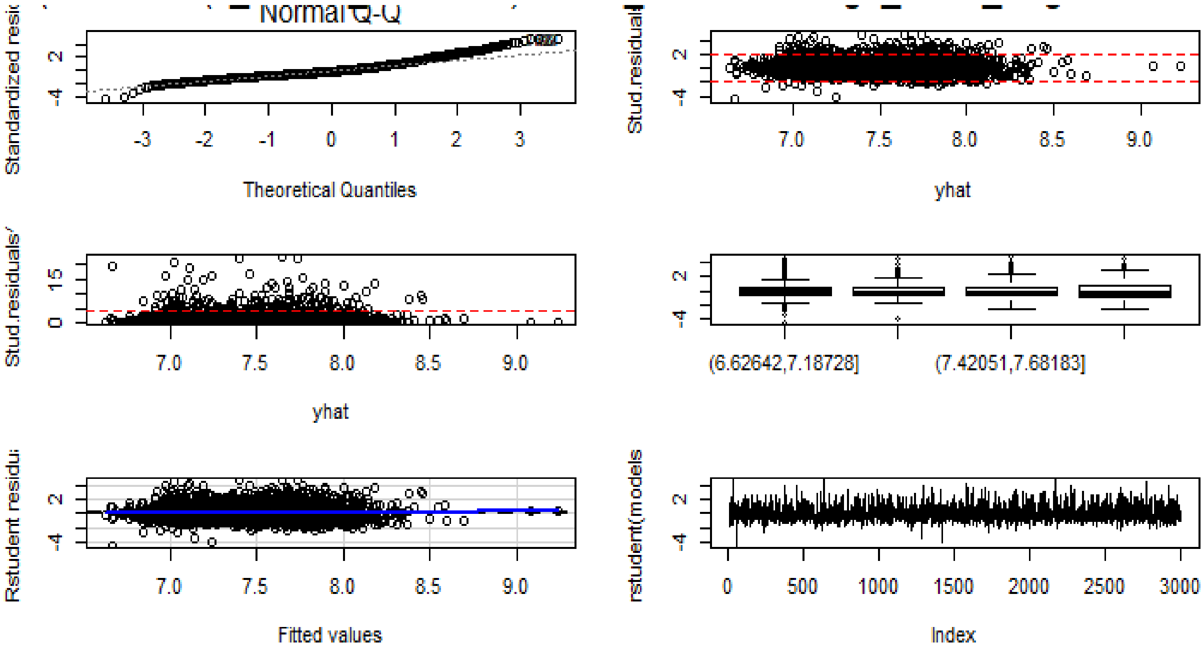


Figure 4

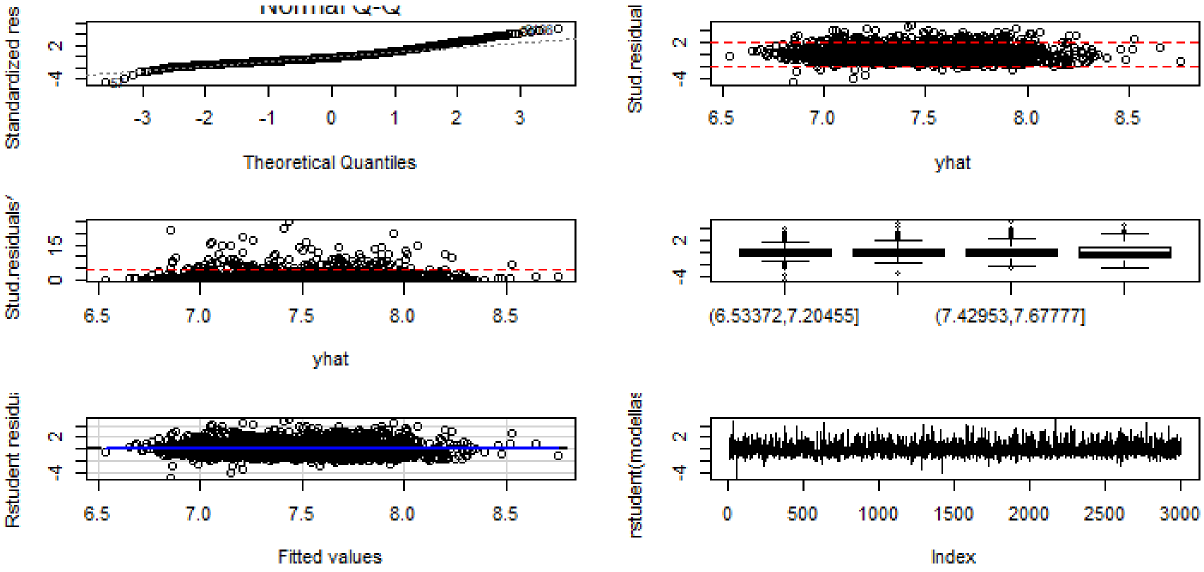


Figure 5

Before leaving the dataset with these valid models, it is tried with feature scaling if there is any room for improvement. With this aim two functions defined. One function called “scalevar()” aims to scale the whole applied column, between -1 to 1, with using (*x/max(x)*)formula. And the second function called “normalizevar()” aims to replace all x values in rows of applied column with (*(x-mean(x))/(max(x)-min(x))*) formula. With this formula, it is expected to make features have approximately zero mean(mean normalization).

Addition to these functions, also taking logarithm used as another method to rescale predictors and the response variable as well. However, instead of taking log(y), it is used as log(y+1), so such that zeros become 1s and can then be kept in the regression. This biases the model a bit and is somewhat frowned upon, but in practice its negative side effects are expected to be typically pretty minor.

After all, with manually checking, these techniques applied for some of predictors, which are mentioned below in detail;

1. Taking absolute of "kw_min_min", "kw_avg_min", "kw_min_avg" to avoid negative variables for taking log()
2. Taking log() and add 1 to the result data in "n_tokens_content", "kw_max_min", "kw_avg_min", "kw_min_min", "kw_min_avg", "kw_min_max", "kw_max_max",

"kw_avg_max", "kw_max_avg", "kw_avg_avg", "self_reference_min_shares", "self_reference_max_shares","self_reference_avg_sharess", "shares"

3. Try both scalevar() function and normalizevar() function and decide to apply scalevar() to both of "n_tokens_title","num_hrefs","num_self_hrefs","num_imgs", "num_videos", "average_token_length", "num_keywords"

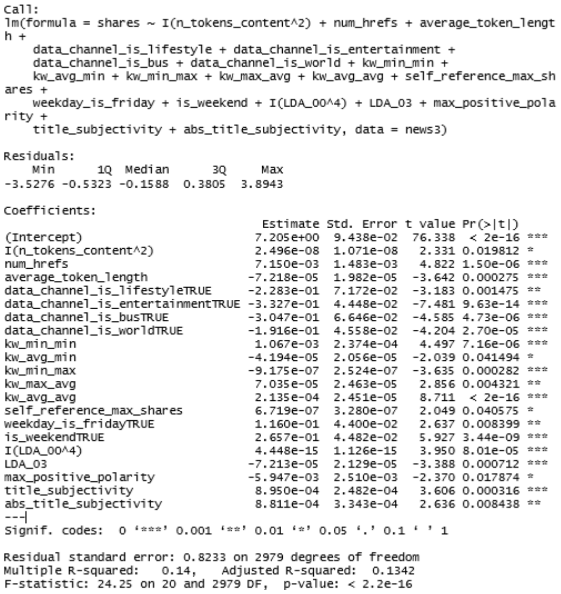
However, although after scaling predictors another model tried to created, the R^2 , significance and the predict performance were worse than old models mentioned above.

And the last try about feature scaling was only taking absolute values of “avg_negative_polarity”, “min_negative_polarity”, “max_negative_polarity”, “title_sentiment_polarity” and “kw_min_min”. After this step, “both” stepwise method used again with new dataset and another model created. Again there was some trials occurred to find a model with significant predictors and also a valid one in terms of assumptions. At last another model called “**modelst3**” created;

Log(shares) = 6.935 + 2.445e-08*(n_tokens_content^2) + 0.00709*num_hrefs - 6.588e-05* average_token_length – 0.239*data_channel_is_lifestyleTRUE – 0.3163*data_channel_is_entertainmentTRUE – 0.1258*data_channel_is_busTRUE – 0.2072*data_channel_is_worldTRUE + 0.001139*kw_min_min - 4.290e-05*kw_avg_min - 9.228e-07*kw_min_max + 0.06962*log(kw_max_avg + 1) + 8.077e-08*(kw_avg_avg^2) + 7.044e-07*self_reference_max_shares + 0.1184*weekday_is_fridayTRUE + 0.2685* is_weekendTRUE - 3.445e-08*(LDA_03^2) - 2.812e-07*(rate_negative_words^2) – 0.00508*max_positive_polarity + 8.917e-04*title_subjectivity + 9.065e-04*abs_title_subjectivity + Ɛ

$\text{Ɛ} \sim N(0, 0.8231^2)$

Both the normality and the homoscedasticity assumptions are not rejected and the linearity assumption met (Tukey’s p=0.72>0.05, Non-constant Variance Score p=~0<0.05, Levene’s p=~0<0.05); see Table 6 for details & Figure 7 for visualizations of residuals.

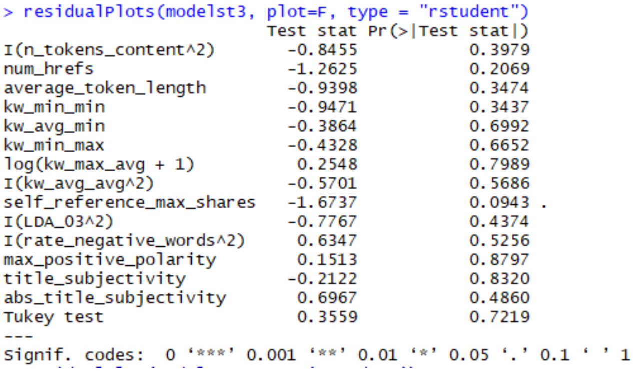


```
Call:
lm(formula = shares ~ I(n_tokens_content^2) + num_hrefs + average_token_lengt
h +
    data_channel_is_lifestyle + data_channel_is_entertainment +
    data_channel_is_bus + data_channel_is_world + kw_min_min +
    kw_avg_min + kw_min_max + kw_max_avg + kw_avg_avg + self_reference_max_sh
ares +
    weekday_is_friday + is_weekend + I(LDA_00^4) + LDA_03 + max_positive_pola
rity +
    title_subjectivity + abs_title_subjectivity, data = news3)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5276 -0.5323 -0.1588  0.3805  3.8943

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    7.205e+00  9.438e-02  76.338  < 2e-16 ***
I(n_tokens_content^2)          2.496e-08  1.071e-08   2.331 0.019812 *
num_hrefs                      7.150e-03  1.483e-03   4.822 1.50e-06 ***
average_token_length          -7.218e-05  1.982e-05  -3.642 0.000275 ***
data_channel_is_lifestyleTRUE -2.283e-01  7.172e-02  -3.183 0.001475 **
data_channel_is_entertainmentTRUE -3.327e-01  4.448e-02  -7.481 9.63e-14 ***
data_channel_is_busTRUE       -3.047e-01  6.646e-02  -4.585 4.73e-06 ***
data_channel_is_worldTRUE     -1.916e-01  4.558e-02  -4.204 2.70e-05 ***
kw_min_min                     1.067e-03  2.374e-04   4.497 7.16e-06 ***
kw_avg_min                    -4.194e-05  2.056e-05  -2.039 0.041494 *
kw_min_max                    -9.175e-07  2.524e-07  -3.635 0.000282 ***
kw_max_avg                     7.035e-05  2.463e-05   2.856 0.004321 **
kw_avg_avg                     2.135e-04  2.451e-05   8.711  < 2e-16 ***
self_reference_max_shares      6.719e-07  3.280e-07   2.049 0.040575 *
weekday_is_fridayTRUE          1.160e-01  4.400e-02   2.637 0.008399 **
is_weekendTRUE                 2.657e-01  4.482e-02   5.927 3.44e-09 ***
I(LDA_00^4)                    4.448e-15  1.126e-15   3.950 8.01e-05 ***
LDA_03                        -7.213e-05  2.129e-05  -3.388 0.000712 ***
max_positive_polarity         -5.947e-03  2.510e-03  -2.370 0.017874 *
title_subjectivity             8.950e-04  2.482e-04   3.606 0.000316 ***
abs_title_subjectivity         8.811e-04  3.343e-04   2.636 0.008438 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8233 on 2979 degrees of freedom
Multiple R-squared:   0.14,	Adjusted R-squared:  0.1342
F-statistic: 24.25 on 20 and 2979 DF,  p-value: < 2.2e-16
```



```
> residualPlots(modelst3, plot=F, type = "rstudent")
                         Test stat Pr(>|Test stat|)
I(n_tokens_content^2)      -0.8455           0.3979
num_hrefs                  -1.2625           0.2069
average_token_length       -0.9398           0.3474
kw_min_min                 -0.9471           0.3437
kw_avg_min                 -0.3864           0.6992
kw_min_max                 -0.4328           0.6652
log(kw_max_avg + 1)         0.2548           0.7989
I(kw_avg_avg^2)            -0.5701           0.5686
self_reference_max_shares  -1.6737           0.0943 .
I(LDA_03^2)                -0.7767           0.4374
I(rate_negative_words^2)    0.6347           0.5256
max_positive_polarity       0.1513           0.8797
title_subjectivity         -0.2122           0.8320
abs_title_subjectivity      0.6967           0.4860
Tukey test                  0.3559           0.7219
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 6

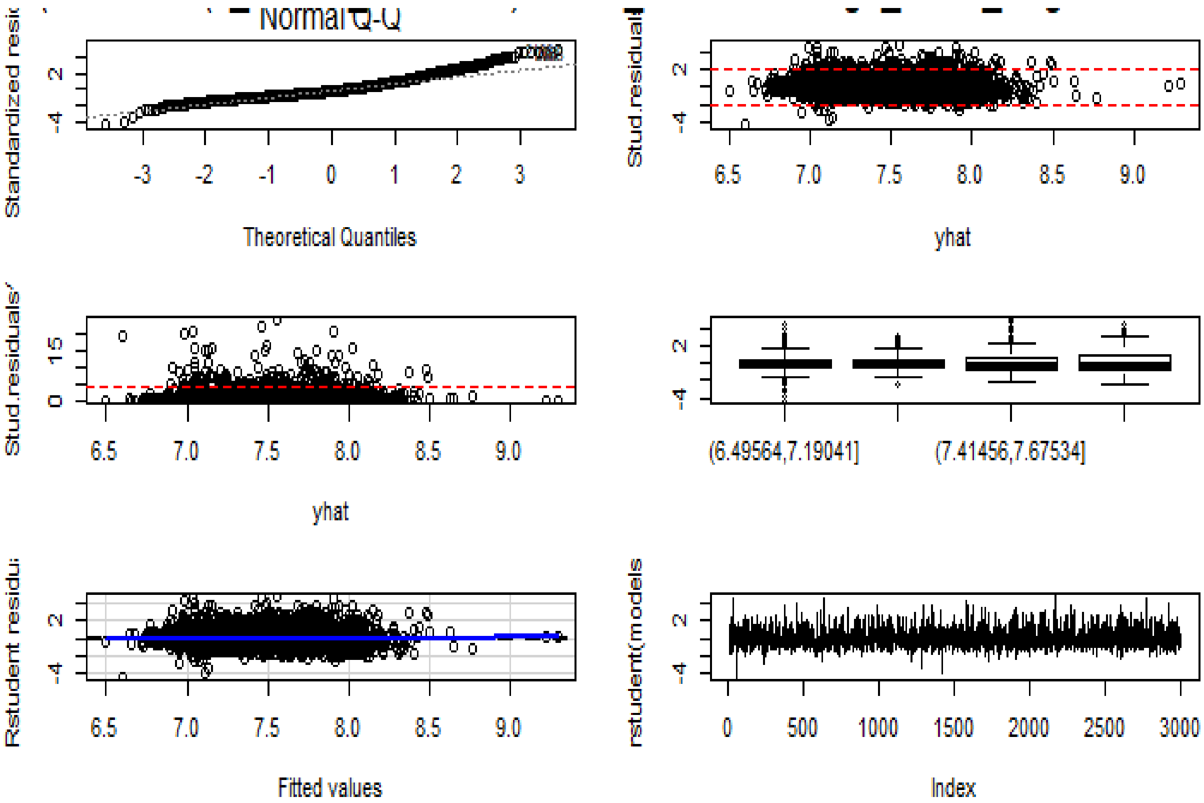
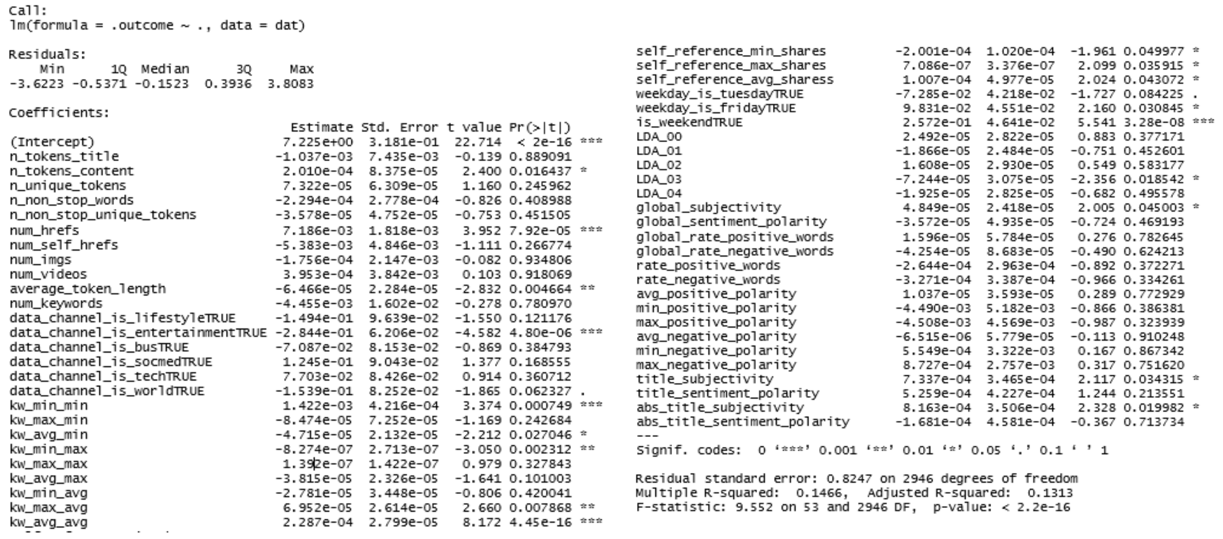


Figure 6

And lastly, a model selected called “**modelcv**” with the 10 fold cross validation method. This time again the cleaned dataset used and allow model to use all predictors as features. After checking summary of the cross validation model, it seems that R^2 resulted 13.13%, which means the model has 13.13% ability to explain the variability of the number of shares. However, this time some of predictors was not significant, which may cause to draw wrong conclusion between the variables. *Both the normality and the homoscedasticity assumptions are not rejected but the linearity assumption does not met* in terms of some of the predictors p-values *(Tukey's p=0.14>0.05, Non-constant Variance Score p=~0<0.05, Levene's p=~0<0.05); see Table 7 for details & Figure 8 for visualizations of residuals.*

Since the logarithm of shares data includes variables between 3.13 to 11.6, RMSE results which calculated around 0.8 in all models seems not good but not bad as well.

Moreover, none of the selected models' have collinearity problem except the last one created with cross validation method, which includes all variables as predictors.



```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.6223 -0.5371 -0.1523  0.3936  3.8083 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  7.225e+00  3.181e-01  22.714  < 2e-16 ***
n_tokens_title              -1.037e-03  7.435e-03  -0.139 0.889091    
n_tokens_content             2.010e-04  8.375e-05   2.400 0.016437 *  
n_unique_tokens              7.322e-05  6.309e-05   1.160 0.245962    
n_non_stop_words            -2.294e-04  2.778e-04  -0.826 0.408988    
n_non_stop_unique_tokens    -3.578e-05  4.752e-05  -0.753 0.451505    
num_hrefs                    7.186e-03  1.818e-03   3.952 7.92e-05 ***
num_self_hrefs              -5.383e-03  4.846e-03  -1.111 0.266774    
num_imgs                    -1.756e-04  2.147e-03  -0.082 0.934806    
num_videos                   3.953e-04  3.842e-03   0.103 0.918069    
average_token_length        -6.466e-05  2.284e-05  -2.832 0.004664 ** 
num_keywords                -4.455e-03  1.602e-02  -0.278 0.780970    
data_channel_is_lifestyleTRUE -1.494e-01  9.639e-02  -1.550 0.121176    
data_channel_is_entertainmentTRUE -2.844e-01  6.206e-02  -4.582 4.80e-06 ***
data_channel_is_busTRUE     -7.087e-02  8.153e-02  -0.869 0.384793    
data_channel_is_socmedTRUE   1.245e-01  9.043e-02   1.377 0.168555    
data_channel_is_techTRUE     7.703e-02  8.426e-02   0.914 0.360712    
data_channel_is_worldTRUE   -1.539e-01  8.252e-02  -1.865 0.062327 .  
kw_min_min                   1.422e-03  4.216e-04   3.374 0.000749 ***
kw_max_min                  -8.474e-05  7.252e-05  -1.169 0.242684    
kw_avg_min                  -4.715e-05  2.132e-05  -2.212 0.027046 *  
kw_min_max                  -8.274e-07  2.713e-07  -3.050 0.002312 ** 
kw_max_max                   1.392e-07  1.422e-07   0.979 0.327843    
kw_avg_max                  -3.815e-05  2.326e-05  -1.641 0.101003    
kw_min_avg                  -2.781e-05  3.448e-05  -0.806 0.420041    
kw_max_avg                   6.952e-05  2.614e-05   2.660 0.007868 ** 
kw_avg_avg                   2.287e-04  2.799e-05   8.172 4.45e-16 ***

self_reference_min_shares       -2.001e-04  1.020e-04  -1.961 0.049977 *  
self_reference_max_shares        7.086e-07  3.376e-07   2.099 0.035915 *  
self_reference_avg_sharess       1.007e-04  4.977e-05   2.024 0.043072 *  
weekday_is_tuesdayTRUE          -7.285e-02  4.218e-02  -1.727 0.084225 .  
weekday_is_fridayTRUE            9.831e-02  4.551e-02   2.160 0.030845 *  
is_weekendTRUE                   2.572e-01  4.641e-02   5.541 3.28e-08 ***
LDA_00                           2.492e-05  2.822e-05   0.883 0.377171    
LDA_01                          -1.866e-05  2.484e-05  -0.751 0.452601    
LDA_02                           1.608e-05  2.930e-05   0.549 0.583177    
LDA_03                          -7.244e-05  3.075e-05  -2.356 0.018542 *  
LDA_04                          -1.925e-05  2.825e-05  -0.682 0.495578    
global_subjectivity              4.849e-05  2.418e-05   2.005 0.045003 *  
global_sentiment_polarity       -3.572e-05  4.935e-05  -0.724 0.469193    
global_rate_positive_words       1.596e-05  5.784e-05   0.276 0.782645    
global_rate_negative_words      -4.254e-05  8.683e-05  -0.490 0.624213    
rate_positive_words             -2.644e-04  2.963e-04  -0.892 0.372271    
rate_negative_words             -3.271e-04  3.387e-04  -0.966 0.334261    
avg_positive_polarity            1.037e-05  3.593e-05   0.289 0.772929    
min_positive_polarity           -4.490e-03  5.182e-03  -0.866 0.386381    
max_positive_polarity           -4.508e-03  4.569e-03  -0.987 0.323939    
avg_negative_polarity           -6.515e-06  5.779e-05  -0.113 0.910248    
min_negative_polarity            5.549e-04  3.322e-03   0.167 0.867342    
max_negative_polarity            8.727e-04  2.757e-03   0.317 0.751620    
title_subjectivity               7.337e-04  3.465e-04   2.117 0.034315 *  
title_sentiment_polarity         5.259e-04  4.227e-04   1.244 0.213551    
abs_title_subjectivity           8.163e-04  3.506e-04   2.328 0.019982 *  
abs_title_sentiment_polarity    -1.681e-04  4.581e-04  -0.367 0.713734    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8247 on 2946 degrees of freedom
Multiple R-squared:  0.1466,	Adjusted R-squared:  0.1313 
F-statistic: 9.552 on 53 and 2946 DF,  p-value: < 2.2e-16
```

Table 7

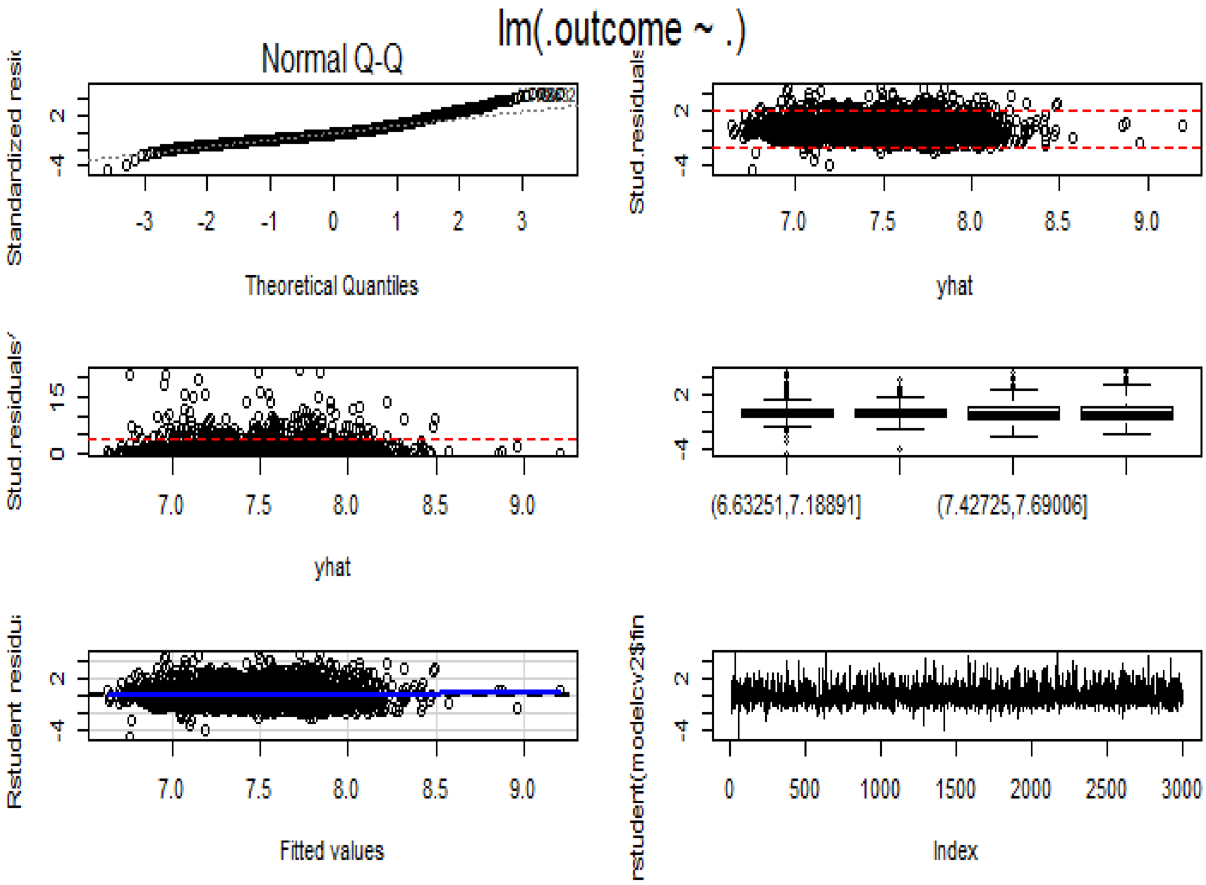


Figure 7

4. Compare Prediction Performance in Test Data via Models

From the previous topic as it mentioned, there are 5 models created at total. It is called to the first model "modelraw", the second model created via stepwise method called "modelst2", the third model called "modellasso", the fourth one called "modelst3" and the last model created via cross validation method called "modelcv". For reminding the modelraw and modelcv models did not capture all the assumptions, but the other three was totally valid models.

In this topic it is planned to compare models performance with the test data. As mentioned, none of the models saw this dataset before, in other words this dataset is an unseen data for the models.

For comparing the models in terms of their prediction performance, all predictors given to the model and expected that model predicts a share value, after this step, the predicted and the actual dataset compared and accuracy of model measured in terms of its R^2 and its root mean square errors. The meaning of R^2 is the model's capacity to explain response variable as it mentioned before, but the root mean square error calculated from differences between predicted values and actual values.

- The **modelraw** had;
  - 9.63% R^2 and 0.8411 RMSE values in **train dataset**,
  - 7.46% R^2 and 1.1558 RMSE values in **test dataset**.
- The **modelst2** had;
  - 13.42% R^2 and 0.8233 RMSE values in **train dataset**,
  - 0.095% R^2 and 14.7439 RMSE values in **test dataset**.
- The **modellasso** had;
  - 12.53% R^2 and 0.8275 RMSE values in **train dataset**,
  - 4.18% R^2 and 2.2728 RMSE values in **test dataset**.
- The **modelst3** had;
  - 13.46% R^2 and 0.8231 RMSE values in **train dataset**,
  - 6.77% R^2 and 2.3755 RMSE values in **test dataset.**
- The **modelcv** had;
  - 13.13% R^2 and 0.8247 RMSE values in **train dataset,**
  - 9.18% R^2 and 1.0843 RMSE values in **test dataset.**

RMSE shows how the model performs out-of-sample, rather than in-sample as it resulted in the training data. The RMSE for the training and the test sets should be very similar for a good model. If the RMSE for the test set is much higher than that of the training set, it is likely that the model was over fit to the data.

However, these results were expected because, all models selected only based on the train dataset and this causes model to maximize its performance on that dataset only, therefore the model captures the noise and the outliers in the data along with the underlying pattern. But it resulted overfitting problem at the model; overfitting in statistics means "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably"[1]. It can be best observed from modelst2 results, it

[1] Overfitting Explanation in Wikipedia (https://en.wikipedia.org/wiki/Overfitting)

resulted with far less than 1% accuracy in test dataset with 14.74 RMSE value, which was expected to give close results with the train data RMSE value of 0.8233. And because of this overfitting problem, all models except modelcv failed to accurately predict the response variable with a similar performance occurred in the training data.

But only the modelcv trained in a proper way with the train dataset. Because it splits data to 10 subsets and trains its model randomly some of these subsets and tests it with other subsets. This model selection process helps it to validate the model with unseen data and avoid overfitting. As it can be seen from prediction results, it got the best scores among trained models.

5. Conclusion

After carefully identified different model options and test their performance with unseen data, although highest accuracy obtained by modelcv(cross-validation model), the final model should be selected from significant and valid models, which encourage me to select **modelst3** as the final model of this paper. Fortunately, if one model has a low R-squared value but the independent variables are statistically significant and has valid assumptions, it can still draw important conclusions about the relationships between the variables. Statistically significant coefficients continue to represent the mean change in the dependent variable given a one-unit shift in the independent variable. Clearly, being able to draw conclusions like this is vital.

And based on the final model;

Log(shares) = 6.935 + 2.445e-08*(n_tokens_content^2) + 0.00709*num_hrefs - 6.588e-05* average_token_length – 0.239*data_channel_is_lifestyleTRUE – 0.3163*data_channel_is_entertainmentTRUE – 0.1258*data_channel_is_busTRUE – 0.2072*data_channel_is_worldTRUE + 0.001139*kw_min_min - 4.290e-05*kw_avg_min - 9.228e-07*kw_min_max + 0.06962*log(kw_max_avg + 1) + 8.077e-08*(kw_avg_avg^2) + 7.044e-07*self_reference_max_shares + 0.1184*weekday_is_fridayTRUE + 0.2685* is_weekendTRUE - 3.445e-08*(LDA_03^2) - 2.812e-07*(rate_negative_words^2) – 0.00508*max_positive_polarity + 8.917e-04*title_subjectivity + 9.065e-04*abs_title_subjectivity + Ɛ

Ɛ ~ N (0, 0.8231^2)

Number of shares predicted by number of words in the content, number of links, average length of the words in the content, data channel if it is 'Lifestyle', 'Entertainment', 'Business' or 'World', worst keyword (min. shares), worst keyword (avg. shares), best keyword (min. shares), avg. keyword (max. shares), avg. keyword (avg. shares), max. shares of referenced articles in Mashable, if weekday is Friday or weekend, rate of negative words among non-neutral tokens, max. polarity of positive words, title subjectivity and absolute subjectivity level.

Addition to this, according to the model it is assumed that, if a post's characteristics include 0 number of words in the content, 0 number of links, 0 average length of the words in the content, data channel is not one of 'Lifestyle', 'Entertainment', 'Business' or 'World', 0 number of worst keyword (min. shares), 0 number of worst keyword (avg. shares), 0 number of best keyword (min. shares), 0 number of avg. keyword (max. shares), 0 number of avg. keyword (avg. shares), 0 number of max. shares of referenced articles in Mashable, the weekday is not Friday or weekend, rate of negative words among non-neutral tokens is 0, max. polarity of positive words is 0, title subjectivity is 0 and absolute subjectivity level is 0 too, it resulted with logarithm of share numbers of 6.935, which can be interpreted to a real number after ejecting it from logarithm, 1027 number of shares. This seems unrealistic, but other characteristics, which did not include in the model as predictors, might be helpful to this result.

From the other perspective; it can be said that, this model assumes, for example; logarithmic version of number of shares increases by the square of number of X+1 words in the content multiple by 2.445e-08, if number of words in the content was X and it increases by 1 unit.

This model is a difficult structured model for interpreting because of transformed variables, but not impossible. It is still hard to schema a possible viral post due to its characteristics, but according to the final model, it can be said that;

- if a post's data channel is one of 'Lifestyle', 'Entertainment', 'Business' or 'World', it has a high negative effect to share numbers,
- if a post shared on Friday or weekend it has a high positive effect
- keywords have relatively lower effects to the share number but specifically worst keyword (avg. shares), best keyword (min. shares) have negative effect but the others have positive
- number of words in the content, number of links, average length of the words in the content, title subjectivity and absolute subjectivity level have positive effects as well

- and the last three variables called rate of negative words among non-neutral tokens, max. polarity of positive words have both negative effects on share numbers.

Last but not least, predictions based on human behaviors like the data used in this report includes subjectivity, so it is difficult to get higher accuracy in general.

I. Reference pages of answers of the questions

Q1 ;

The train dataset, which is going to help to fit a model for future predictions, includes 3000 observations of 62 variables at total. And the test dataset includes 10.000 observations of 62 variables and all of the variables and their order are matched with the train dataset. Additionally, none of them includes same data as the other. [1] "url”, "n_unique_tokens" "n_non_stop_words", "n_non_stop_unique_tokens", "average_token_length", "kw_max_min", "kw_avg_min", "kw_avg_max", "kw_min_avg”, "kw_max_avg", "kw_avg_avg" "self_reference_min_shares", "self_reference_avg_sharess", "LDA_00", "LDA_01", "LDA_02" "LDA_03", "LDA_04”, "global_subjectivity", "global_sentiment_polarity" "global_rate_positive_words", "global_rate_negative_words", "rate_positive_words" "rate_negative_words", "avg_positive_polarity", "min_positive_polarity" "max_positive_polarity", "avg_negative_polarity", "min_negative_polarity" "max_negative_polarity", "title_subjectivity","title_sentiment_polarity", "abs_title_subjectivity", "abs_title_sentiment_polarity" variables identified as factors, others are integers. *(See Report Page 3 – 5 for details, 21 – 34 for individual and pairwise graphs)*

Q2 :Best model selected according to its r^2 and standard error via Stepwise method and the selected model was;

log(shares) = 7.205+ 2.496e-08*(n_tokens_content^2), 0.00715*num_hrefs - 7.218e-05*average_token_length -0.228*data_channel_is_lifestyleTRUE - 0.332* data_channel_is_entertainmentTRUE -0.304*data_channel_is_busTRUE - 0.191*data_channel_is_world 0.001*kw_min_min - 4.194e-05*kw_avg_min - 9.175e-07*kw_min_max 7.035e-05*kw_max_avg + 2.135e-04*kw_avg_avg + 6.719e-07*self_reference_max_shares 0.116*weekday_is_friday + 0.265*is_weekend + 4.448e-15*(LDA_00^4) - 7.213e-05*LDA_03 -0.0059*max_positive_polarity + 8.950e-04*title_subjectivity + 8.811e-04*abs_title_subjectivity + Ɛ

Ɛ ~ N (0, 0.8233^2)

Q3: Assumptions checked, linearity and independence of errors violated, so the model predictors transformed and a valid model created.

Q4 : 10 fold cross validation created with caret package and its prediction power tested in test dataset, the results were better than the models created only via train data, but the assumptions did not met and more than half of predictors was not significant.

(See Report Page 5 – 15 for details of Q3, Q4, Q5)

Q5 – Q6: All models tested, at last the model created with 10 fold cross validation got the highest results from test data among other models.

Q7: The final model selected among the models created via stepwise method, because that was the best valid model due to its ability to met the assumptions. Although, the CV model got the highest results in terms of R^2 and residual error variance, it was not selected because in that model assumptions are violated. Because of this, itmay not work well if assumptions is violated then the forecasts, confidence intervals, and scientific insights yielded by a regression model may be (at best) inefficient or (at worst) seriously biased or misleading.

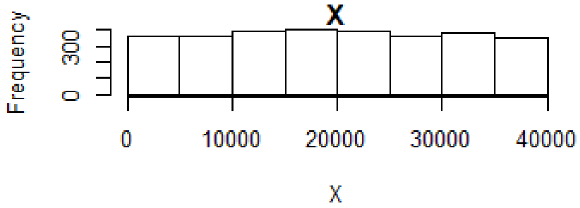
Q8 - Q9 : Number of shares predicted by number of words in the content, number of links, average length of the words in the content, data channel if it is 'Lifestyle', 'Entertainment', 'Business' or 'World', worst keyword (min. shares), worst keyword (avg. shares), best keyword (min. shares), avg. keyword (max. shares), avg. keyword (avg. shares), max. shares of referenced articles in Mashable, if weekday is Friday or weekend, rate of negative words among non-neutral tokens, max. polarity of positive words, title subjectivity and absolute subjectivity level.

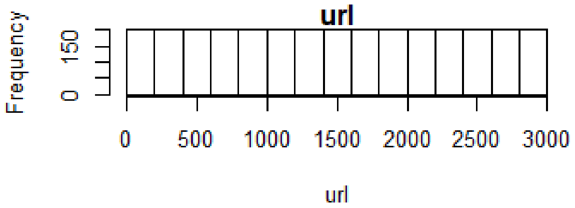
Addition to this, if a post's characteristics include 0 numbers of these variables mentioned above, it resulted with logarithm of share numbers of 6.935, which can be interpreted to a real number after ejecting it from logarithm, 1027 number of shares.

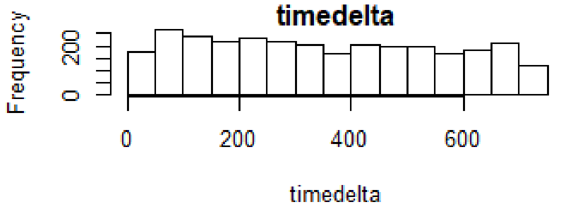
Additionally, this model assumes that, for example; logarithmic version of number of shares increases by the square number of X+1 words in the content multiple by 2.445e-08, if number of words in the content was X and it increases by 1 unit.

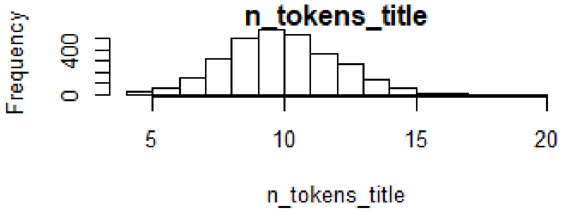
(See Report Page 15 – 19 for Q5, Q6, Q7, Q8, Q9)

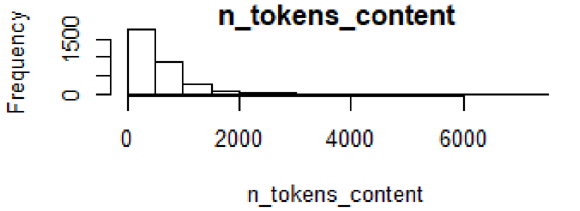
II. Related Figures About Distribution of Variables Individually and Relation Between Response Variable

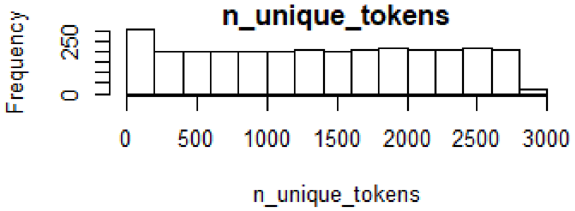


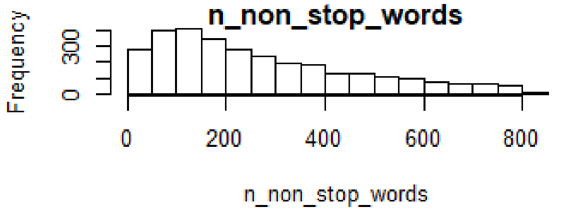


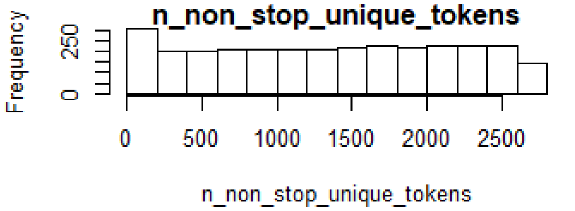


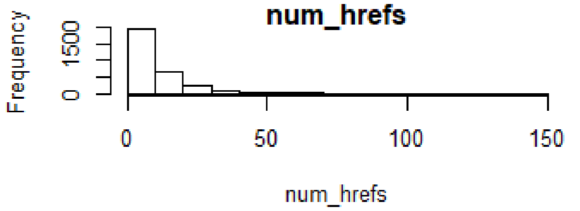


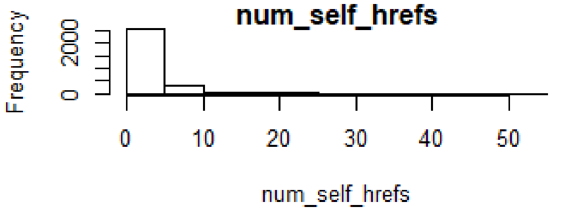


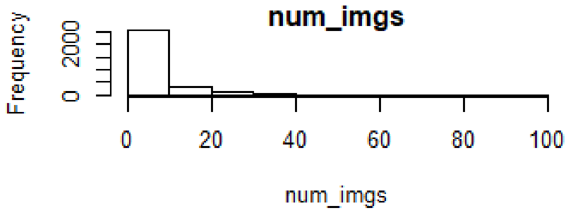


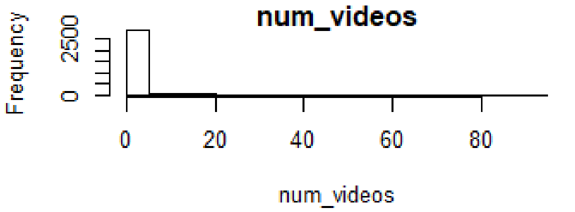


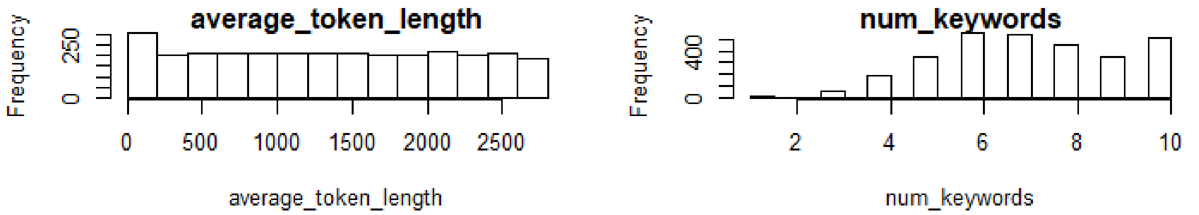


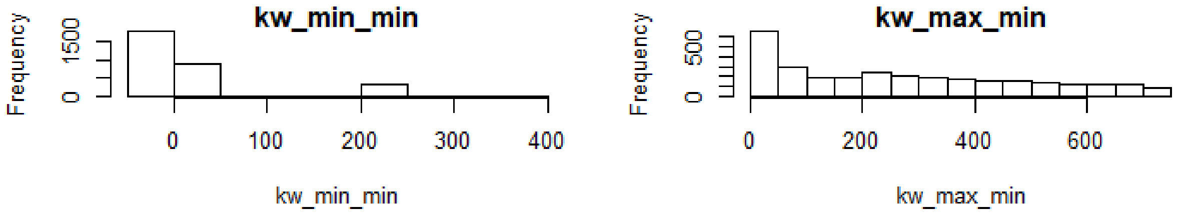


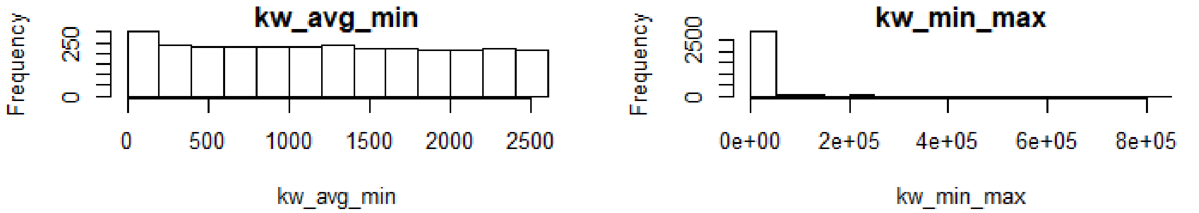


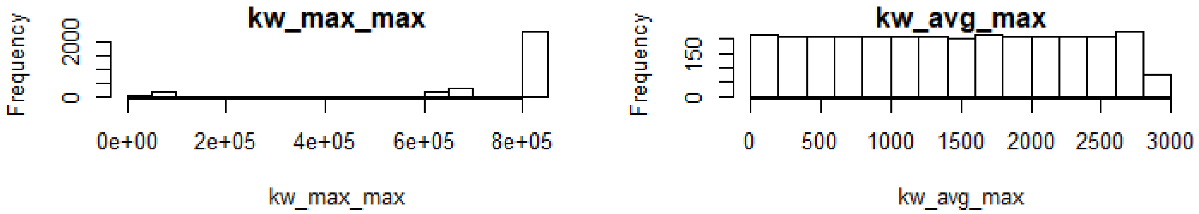


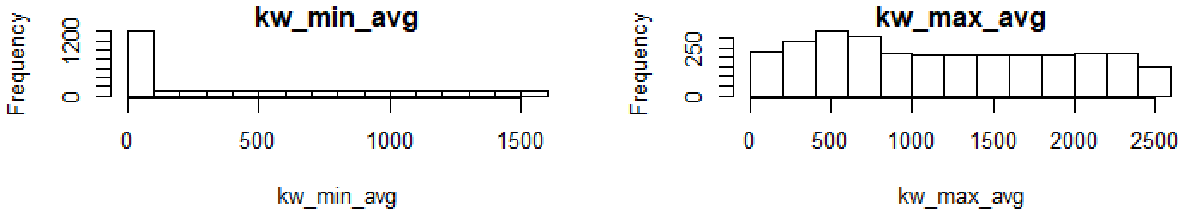


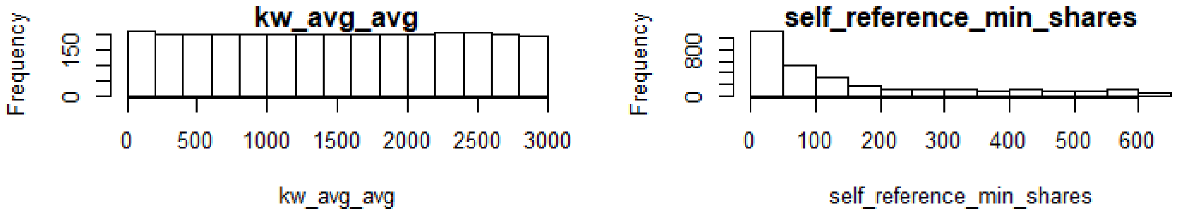

average_token_length
Frequency
0
250
0
500
1000
1500
2000
2500
average_token_length
num_keywords
Frequency
0
400
2
4
6
8
10
num_keywords

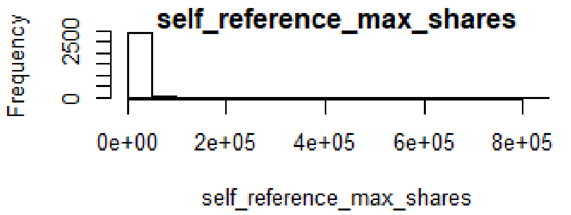

kw_min_min
Frequency
0
1500
0
100
200
300
400
kw_min_min
kw_max_min
Frequency
0
500
0
200
400
600
kw_max_min

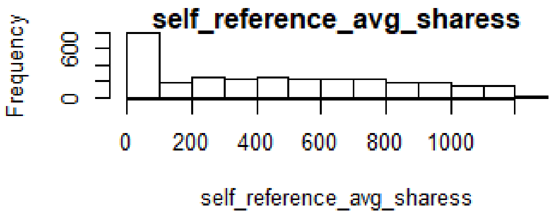

kw_avg_min
Frequency
0
250
0
500
1000
1500
2000
2500
kw_avg_min
kw_min_max
Frequency
0
2500
0e+00
2e+05
4e+05
6e+05
8e+05
kw_min_max

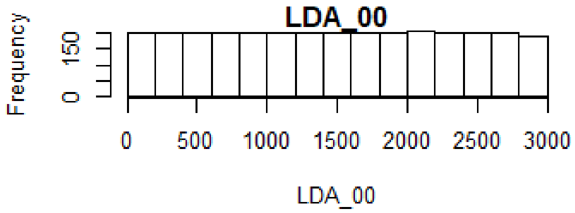

kw_max_max
Frequency
0
2000
0e+00
2e+05
4e+05
6e+05
8e+05
kw_max_max
kw_avg_max
Frequency
0
150
0
500
1000
1500
2000
2500
3000
kw_avg_max

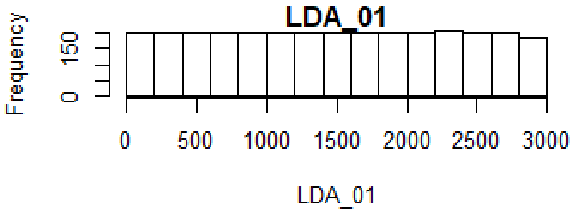

kw_min_avg
Frequency
0
1200
0
500
1000
1500
kw_min_avg
kw_max_avg
Frequency
0
250
0
500
1000
1500
2000
2500
kw_max_avg

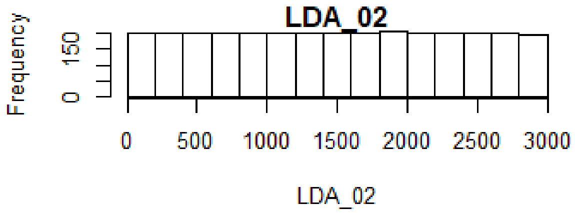

kw_avg_avg
Frequency
0
150
0
500
1000
1500
2000
2500
3000
kw_avg_avg
self_reference_min_shares
Frequency
0
800
0
100
200
300
400
500
600
self_reference_min_shares

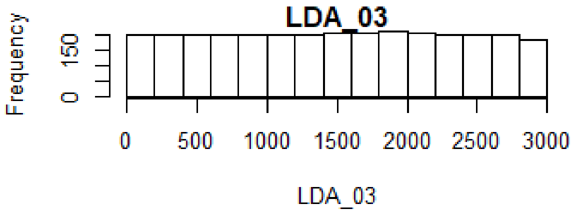

self_reference_max_shares
Frequency
0
2500
0e+00
2e+05
4e+05
6e+05
8e+05
self_reference_max_shares

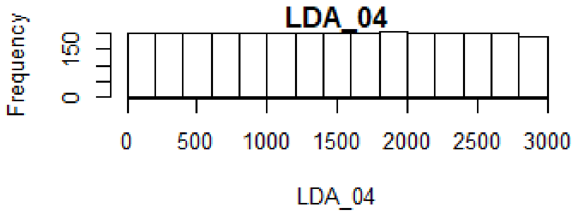

self_reference_avg_sharess
Frequency
0
600
0
200
400
600
800
1000
self_reference_avg_sharess

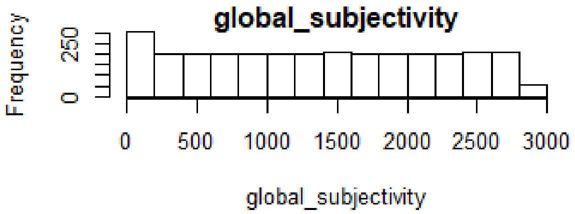

LDA_00
Frequency
0
150
0
500
1000
1500
2000
2500
3000
LDA_00

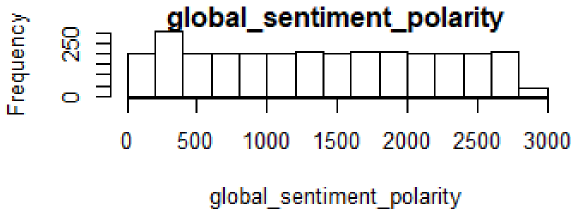

LDA_01
Frequency
0
150
0
500
1000
1500
2000
2500
3000
LDA_01

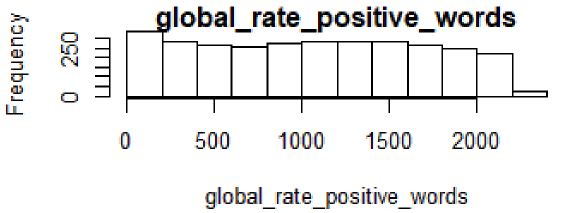

LDA_02
Frequency
0
150
0
500
1000
1500
2000
2500
3000
LDA_02

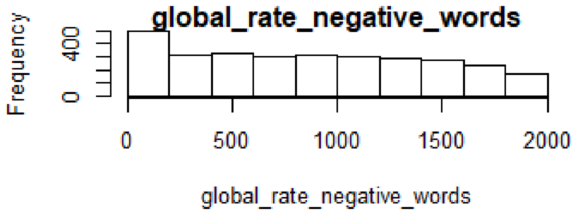

LDA_03
Frequency
0
150
0
500
1000
1500
2000
2500
3000
LDA_03

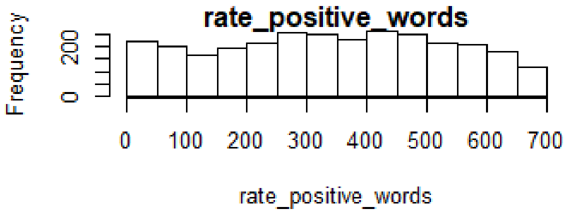

LDA_04
Frequency
0
150
0
500
1000
1500
2000
2500
3000
LDA_04

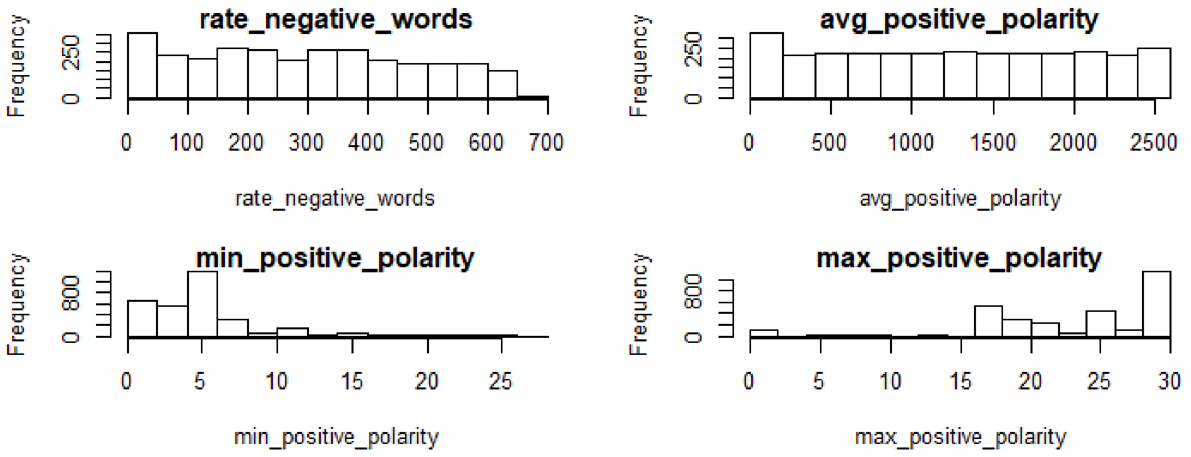

global_subjectivity
Frequency
0
250
0
500
1000
1500
2000
2500
3000
global_subjectivity

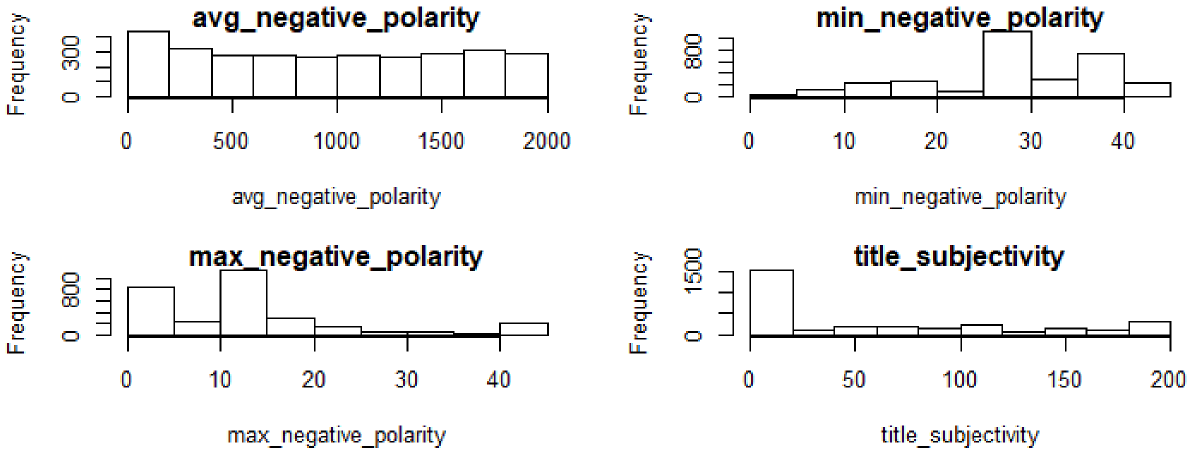

global_sentiment_polarity
Frequency
0
250
0
500
1000
1500
2000
2500
3000
global_sentiment_polarity

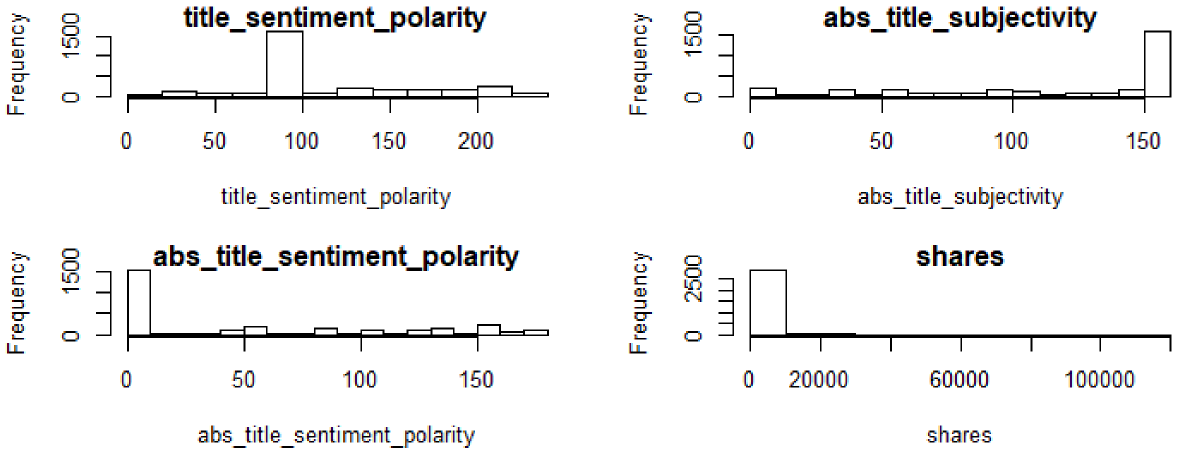

global_rate_positive_words
Frequency
0
250
0
500
1000
1500
2000
global_rate_positive_words

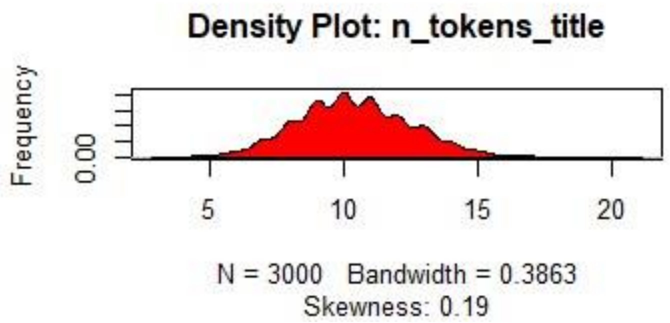

global_rate_negative_words
Frequency
0
400
0
500
1000
1500
2000
global_rate_negative_words

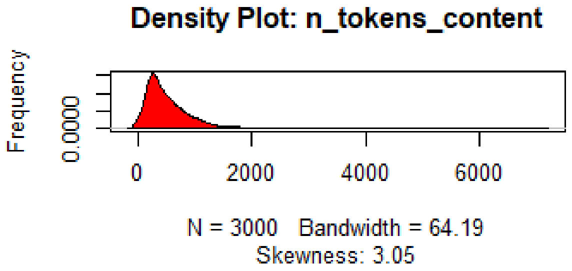

rate_positive_words
Frequency
0
200
0
100
200
300
400
500
600
700
rate_positive_words

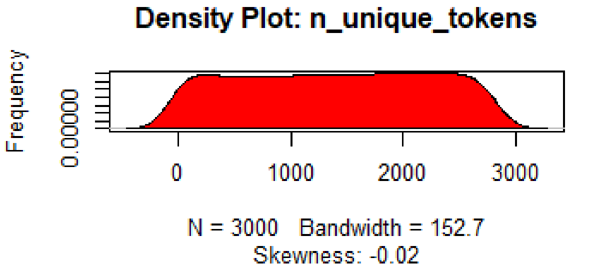

rate_negative_words
Frequency
0
250
0
100
200
300
400
500
600
700
rate_negative_words
avg_positive_polarity
Frequency
0
250
0
500
1000
1500
2000
2500
avg_positive_polarity
min_positive_polarity
Frequency
0
800
0
5
10
15
20
25
min_positive_polarity
max_positive_polarity
Frequency
0
800
0
5
10
15
20
25
30
max_positive_polarity

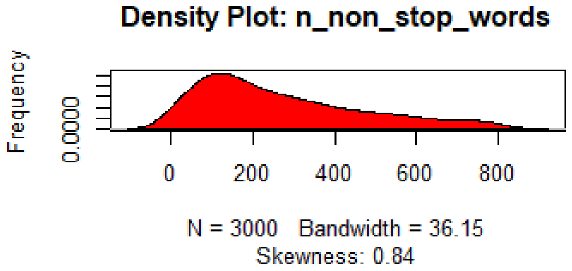

avg_negative_polarity
Frequency
0
300
0
500
1000
1500
2000
avg_negative_polarity
min_negative_polarity
Frequency
0
800
0
10
20
30
40
min_negative_polarity
max_negative_polarity
Frequency
0
800
0
10
20
30
40
max_negative_polarity
title_subjectivity
Frequency
0
1500
0
50
100
150
200
title_subjectivity

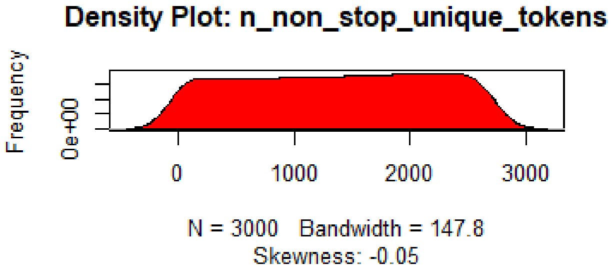

title_sentiment_polarity
Frequency
0
1500
0
50
100
150
200
title_sentiment_polarity
abs_title_subjectivity
Frequency
0
1500
0
50
100
150
abs_title_subjectivity
abs_title_sentiment_polarity
Frequency
0
1500
0
50
100
150
abs_title_sentiment_polarity
shares
Frequency
0
2500
0
20000
60000
100000
shares

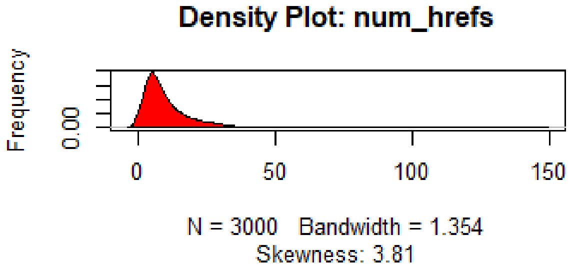

Density Plot: n_tokens_title
Frequency
0.00
5
10
15
20
N = 3000 Bandwidth = 0.3863
Skewness: 0.19

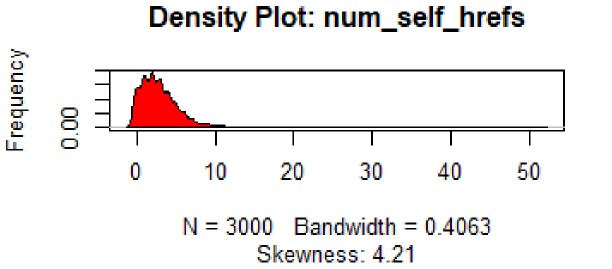

Density Plot: n_tokens_content
Frequency
0.0000
0
2000
4000
6000
N = 3000 Bandwidth = 64.19
Skewness: 3.05

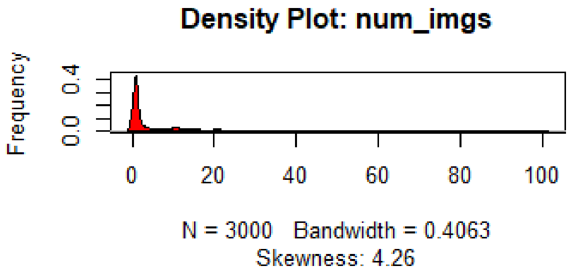

Density Plot: n_unique_tokens
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 152.7
Skewness: -0.02

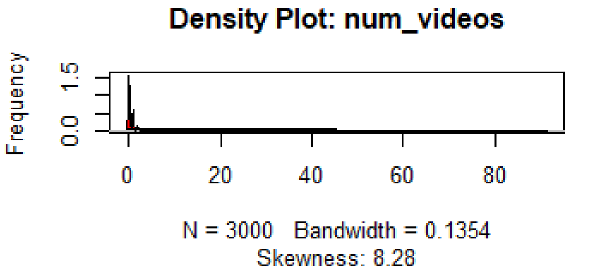

Density Plot: n_non_stop_words
Frequency
0.0000
0
200
400
600
800
N = 3000 Bandwidth = 36.15
Skewness: 0.84

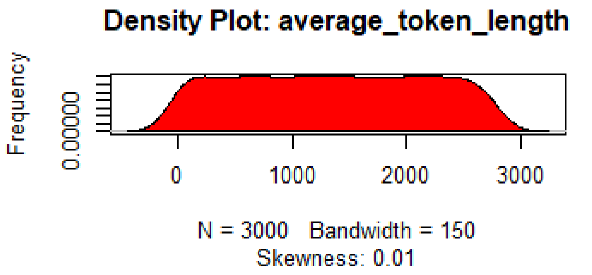

Density Plot: n_non_stop_unique_tokens
Frequency
0e+00
0
1000
2000
3000
N = 3000 Bandwidth = 147.8
Skewness: -0.05

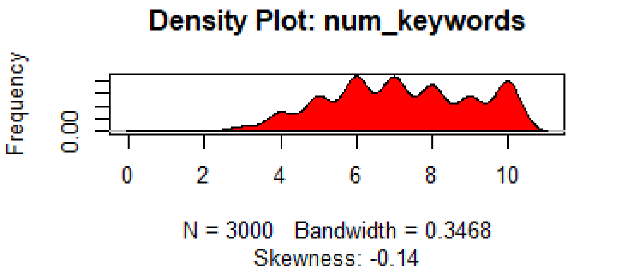

Density Plot: num_hrefs
Frequency
0.00
0
50
100
150
N = 3000 Bandwidth = 1.354
Skewness: 3.81

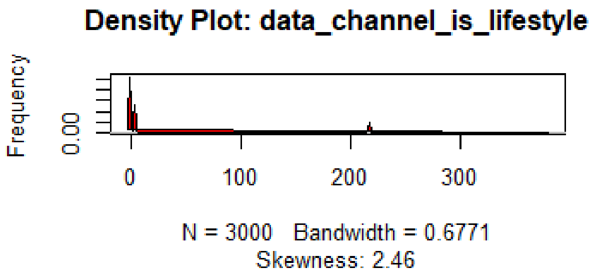

Density Plot: num_self_hrefs
Frequency
0.00
0
10
20
30
40
50
N = 3000 Bandwidth = 0.4063
Skewness: 4.21

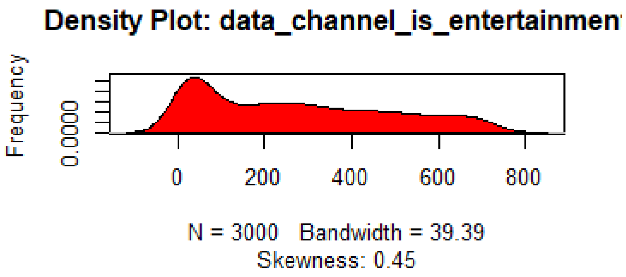

Density Plot: num_imgs
Frequency
0.0
0.4
0
20
40
60
80
100
N = 3000 Bandwidth = 0.4063
Skewness: 4.26

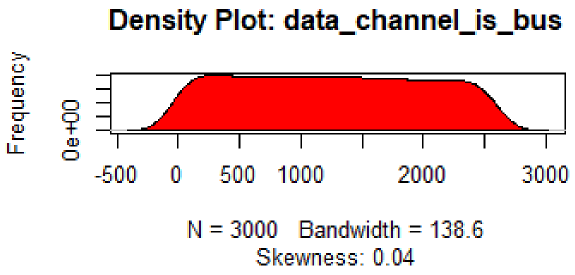

Density Plot: num_videos
Frequency
0.0
1.5
0
20
40
60
80
N = 3000 Bandwidth = 0.1354
Skewness: 8.28

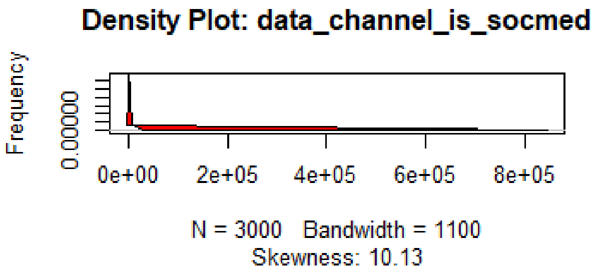

Density Plot: average_token_length
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 150
Skewness: 0.01

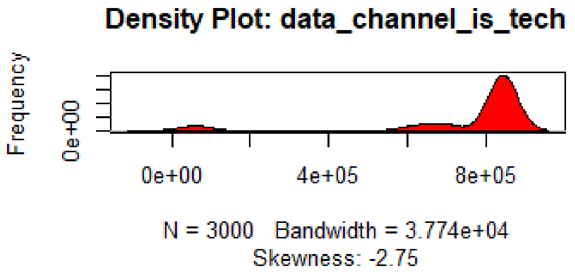

Density Plot: num_keywords
Frequency
0.00
0
2
4
6
8
10
N = 3000 Bandwidth = 0.3468
Skewness: -0.14

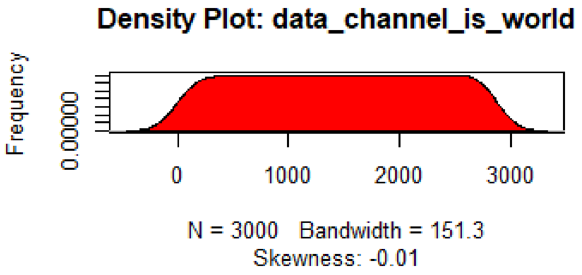

Density Plot: data_channel_is_lifestyle
Frequency
0.00
0
100
200
300
N = 3000 Bandwidth = 0.6771
Skewness: 2.46

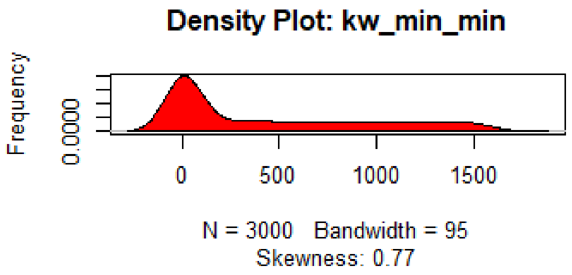

Density Plot: data_channel_is_entertainmen
Frequency
0.0000
0
200
400
600
800
N = 3000 Bandwidth = 39.39
Skewness: 0.45

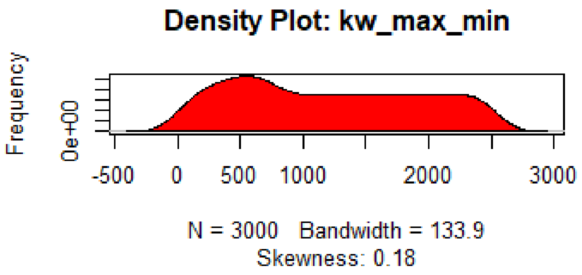

Density Plot: data_channel_is_bus
Frequency
0e+00
-500
0
500
1000
2000
3000
N = 3000 Bandwidth = 138.6
Skewness: 0.04

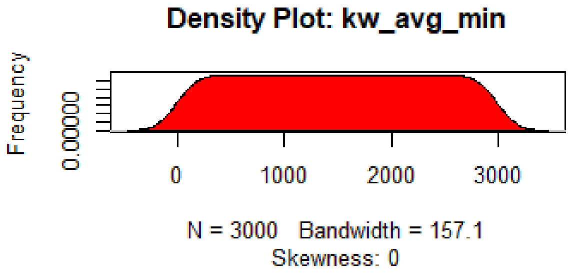

Density Plot: data_channel_is_socmed
Frequency
0.00000
0e+00
2e+05
4e+05
6e+05
8e+05
N = 3000 Bandwidth = 1100
Skewness: 10.13

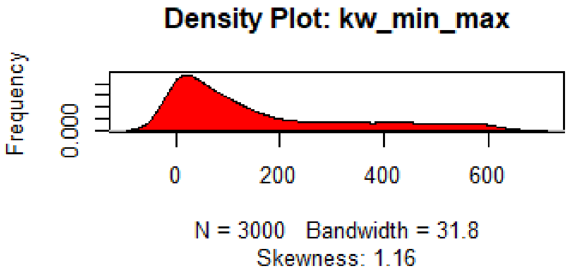

Density Plot: data_channel_is_tech
Frequency
0e+00
0e+00
4e+05
8e+05
N = 3000 Bandwidth = 3.774e+04
Skewness: -2.75

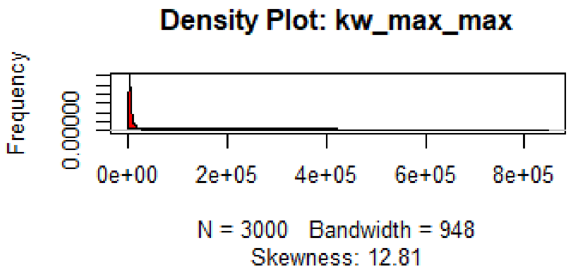

Density Plot: data_channel_is_world
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 151.3
Skewness: -0.01

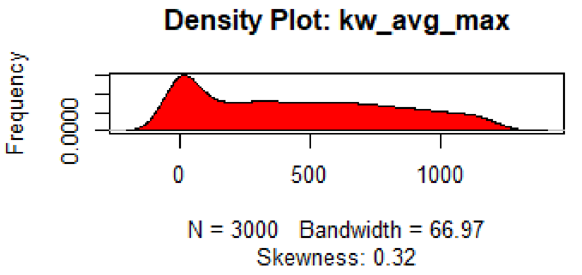

Density Plot: kw_min_min
Frequency
0.0000
0
500
1000
1500
N = 3000 Bandwidth = 95
Skewness: 0.77

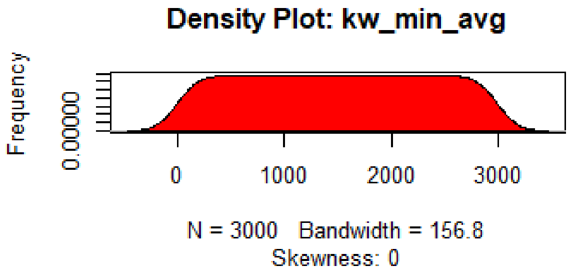

Density Plot: kw_max_min
Frequency
0e+00
-500
0
500
1000
2000
3000
N = 3000 Bandwidth = 133.9
Skewness: 0.18

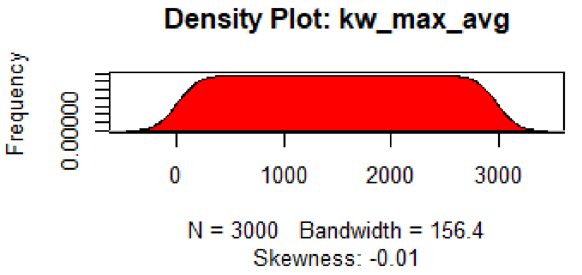

Density Plot: kw_avg_min
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 157.1
Skewness: 0

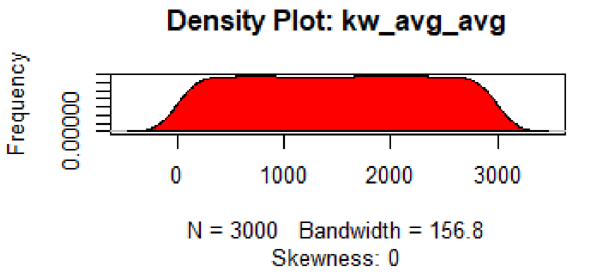

Density Plot: kw_min_max
Frequency
0.000
0
200
400
600
N = 3000 Bandwidth = 31.8
Skewness: 1.16

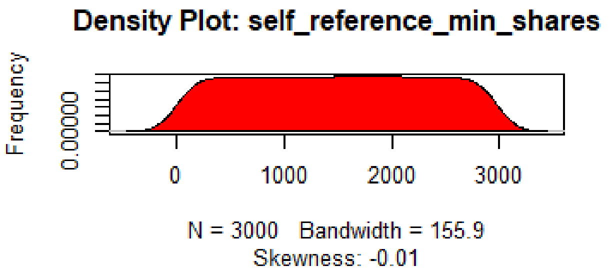

Density Plot: kw_max_max
Frequency
0.00000
0e+00
2e+05
4e+05
6e+05
8e+05
N = 3000 Bandwidth = 948
Skewness: 12.81

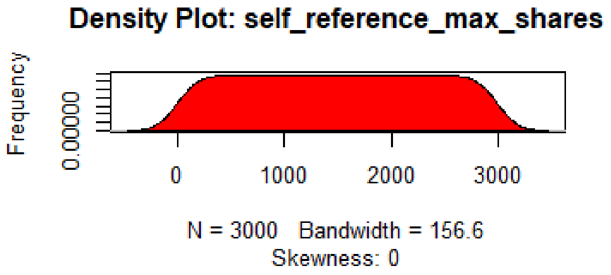

Density Plot: kw_avg_max
Frequency
0.0000
0
500
1000
N = 3000 Bandwidth = 66.97
Skewness: 0.32

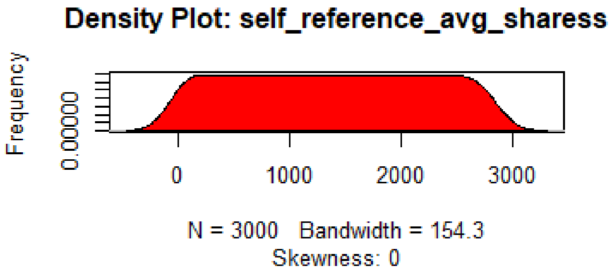

Density Plot: kw_min_avg
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 156.8
Skewness: 0

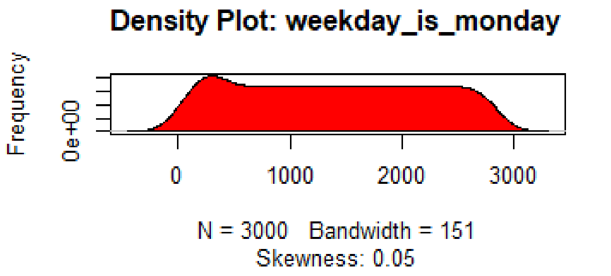

Density Plot: kw_max_avg
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 156.4
Skewness: -0.01

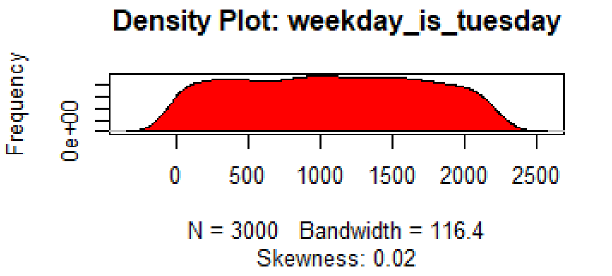

Density Plot: kw_avg_avg
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 156.8
Skewness: 0

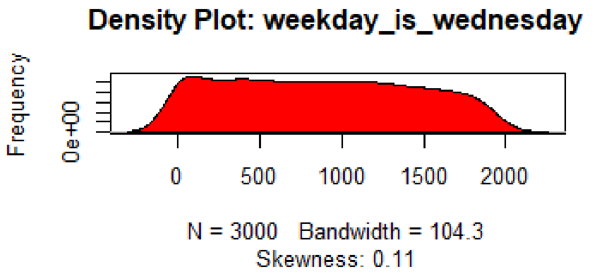

Density Plot: self_reference_min_shares
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 155.9
Skewness: -0.01

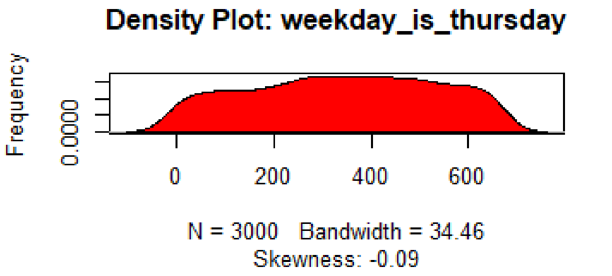

Density Plot: self_reference_max_shares
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 156.6
Skewness: 0

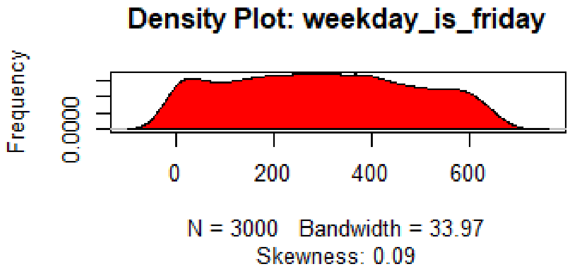

Density Plot: self_reference_avg_sharess
Frequency
0.00000
0
1000
2000
3000
N = 3000 Bandwidth = 154.3
Skewness: 0

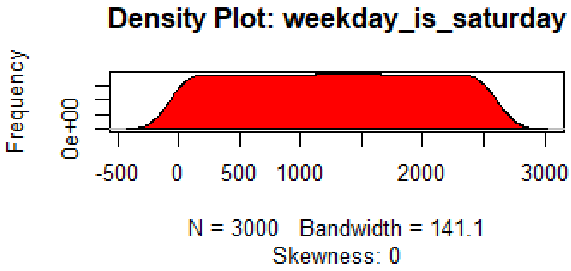

Density Plot: weekday_is_monday
Frequency
0e+00
0
1000
2000
3000
N = 3000 Bandwidth = 151
Skewness: 0.05

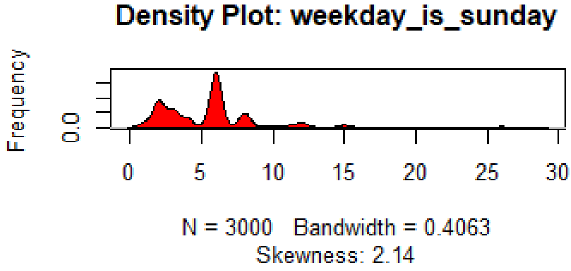

Density Plot: weekday_is_tuesday
Frequency
0e+00
0
500
1000
1500
2000
2500
N = 3000 Bandwidth = 116.4
Skewness: 0.02

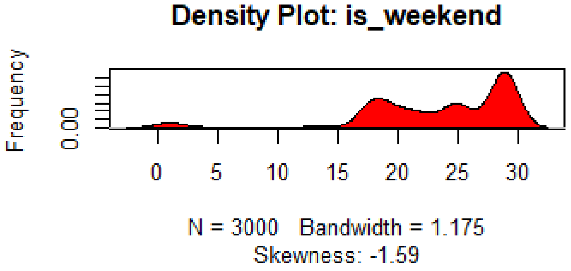

Density Plot: weekday_is_wednesday
Frequency
0e+00
0
500
1000
1500
2000
N = 3000 Bandwidth = 104.3
Skewness: 0.11

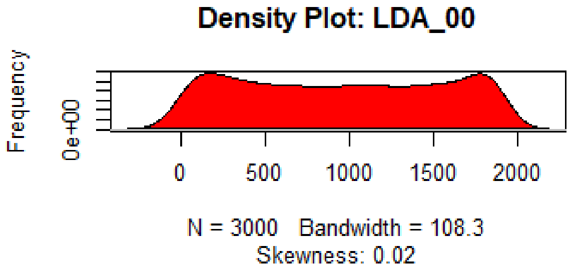

Density Plot: weekday_is_thursday
Frequency
0.0000
0
200
400
600
N = 3000 Bandwidth = 34.46
Skewness: -0.09

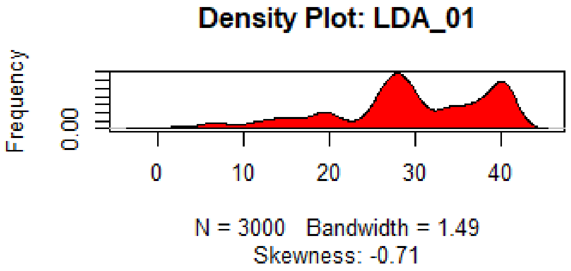

Density Plot: weekday_is_friday
Frequency
0.0000
0
200
400
600
N = 3000 Bandwidth = 33.97
Skewness: 0.09

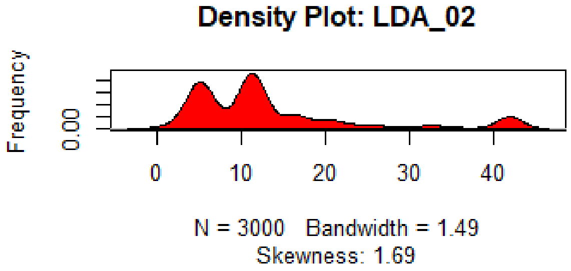

Density Plot: weekday_is_saturday
Frequency
0e+00
-500
0
500
1000
2000
3000
N = 3000 Bandwidth = 141.1
Skewness: 0

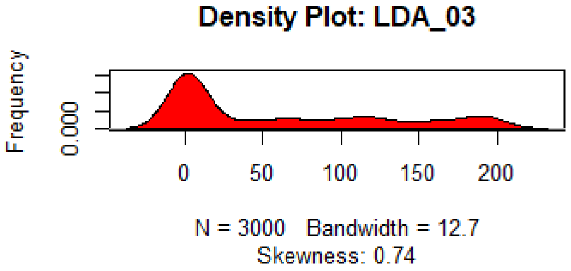

Density Plot: weekday_is_sunday
Frequency
0.0
0
5
10
15
20
25
30
N = 3000 Bandwidth = 0.4063
Skewness: 2.14

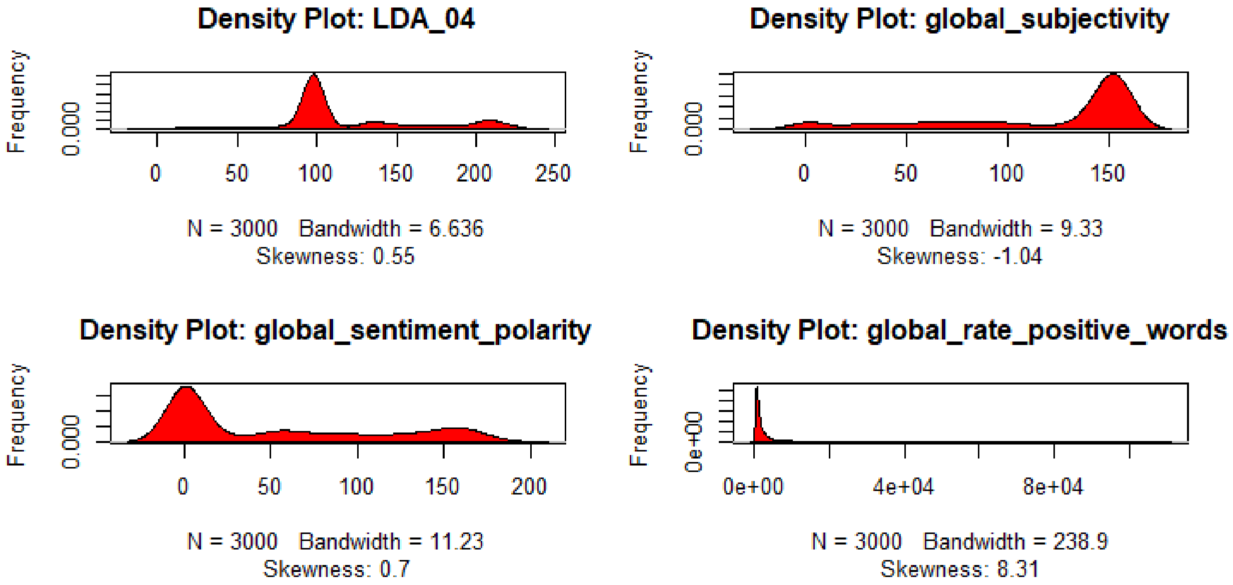

Density Plot: is_weekend
Frequency
0.00
0
5
10
15
20
25
30
N = 3000 Bandwidth = 1.175
Skewness: -1.59


Density Plot: LDA_00
Frequency
0e+00
0
500
1000
1500
2000
N = 3000 Bandwidth = 108.3
Skewness: 0.02

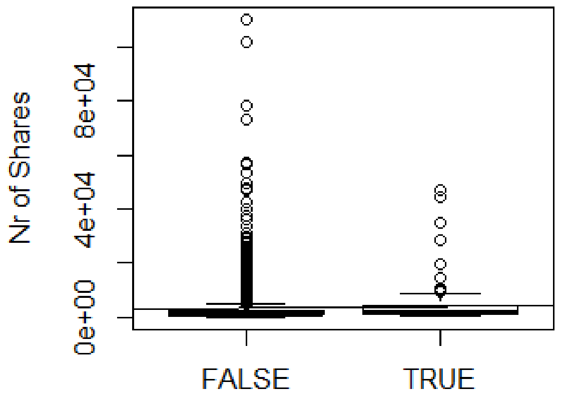

Density Plot: LDA_01
Frequency
0.00
0
10
20
30
40
N = 3000 Bandwidth = 1.49
Skewness: -0.71


Density Plot: LDA_02
Frequency
0.00
0
10
20
30
40
N = 3000 Bandwidth = 1.49
Skewness: 1.69


Density Plot: LDA_03
Frequency
0.000
0
50
100
150
200
N = 3000 Bandwidth = 12.7
Skewness: 0.74

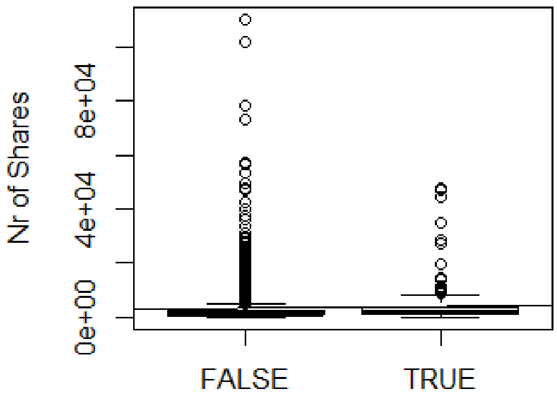

Density Plot: LDA_04
Frequency
0.000
0
50
100
150
200
250
N = 3000 Bandwidth = 6.636
Skewness: 0.55
Density Plot: global_subjectivity
Frequency
0.000
0
50
100
150
N = 3000 Bandwidth = 9.33
Skewness: -1.04
Density Plot: global_sentiment_polarity
Frequency
0.000
0
50
100
150
200
N = 3000 Bandwidth = 11.23
Skewness: 0.7
Density Plot: global_rate_positive_words
Frequency
0e+00
0e+00
4e+04
8e+04
N = 3000 Bandwidth = 238.9
Skewness: 8.31

weekday_is_sunday



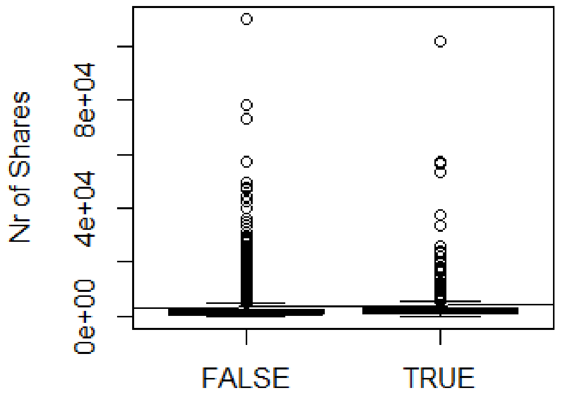
cor btw: 0.0288743718897188

is_weekend



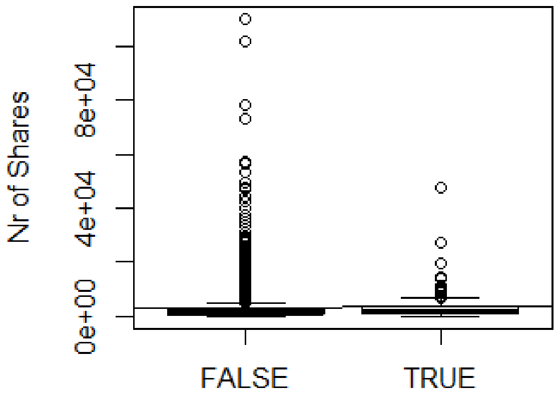
cor btw: 0.0326276911369044

weekday_is_friday

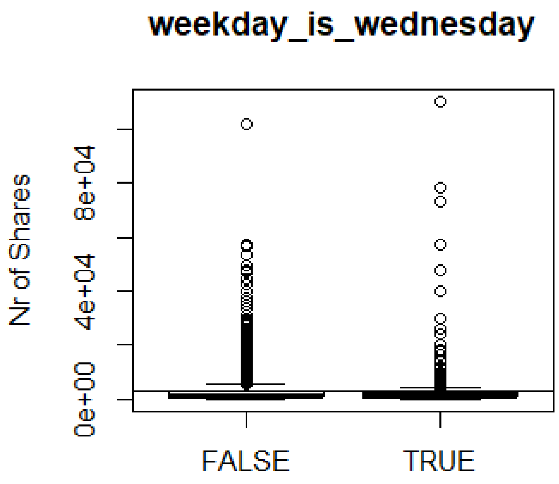


cor btw: 0.0388406009918291

weekday_is_saturday

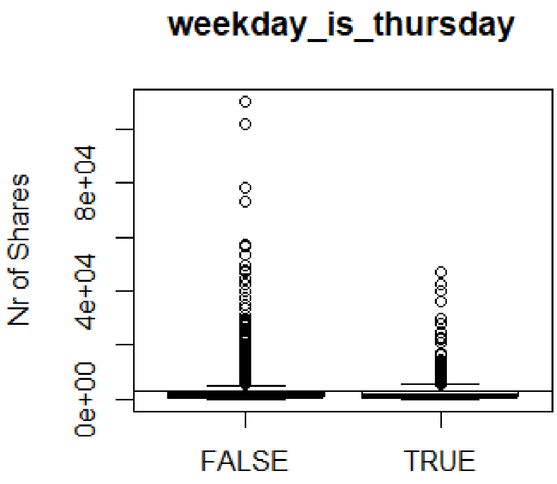


cor btw: 0.0150527521198238



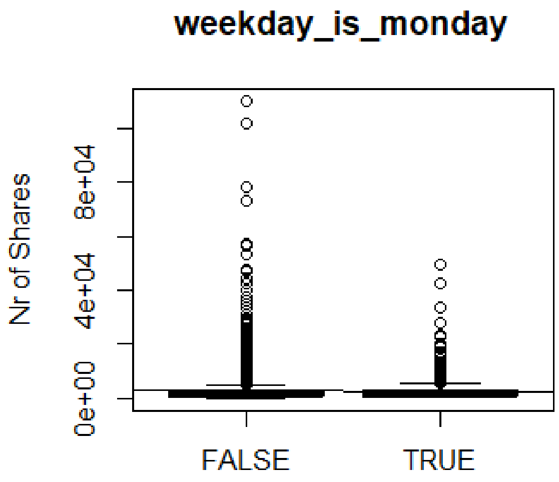
weekday_is_wednesday

cor btw: -0.0032172810610642



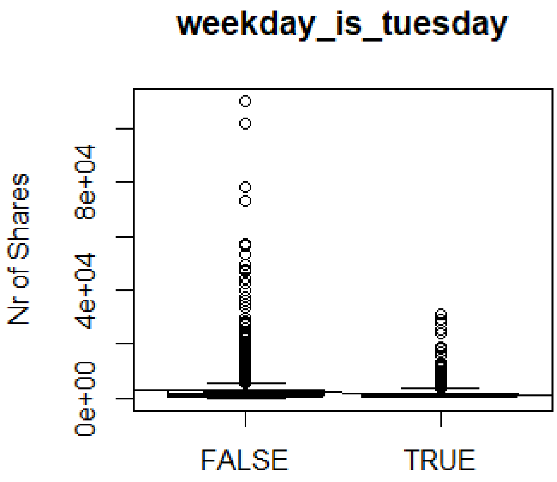
weekday_is_thursday

cor btw: -0.00151813046432575



weekday_is_monday

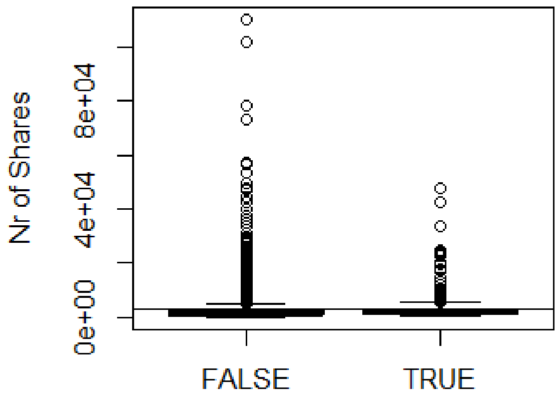
cor btw: -0.0112264463480694



weekday_is_tuesday

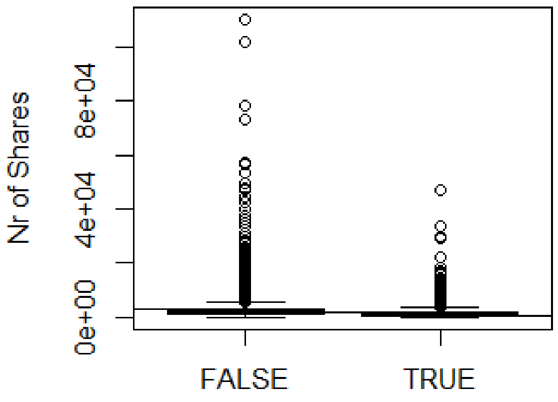
cor btw: -0.0492829543711614

data_channel_is_tech



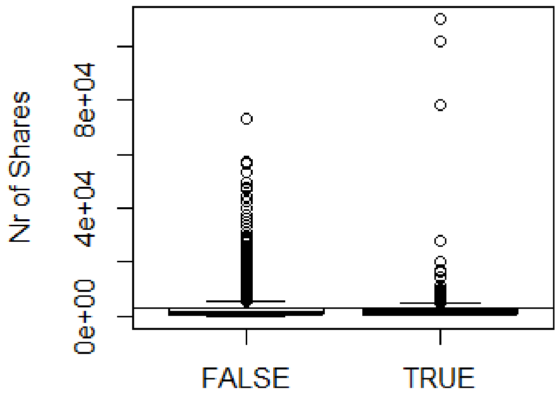
cor btw: -0.00209513019000524

data_channel_is_world



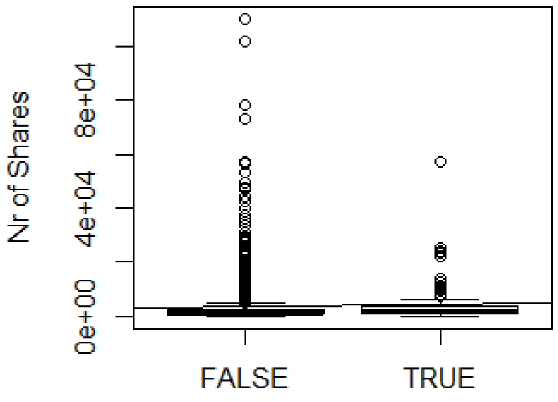
cor btw: -0.0694084565331446

data_channel_is_bus

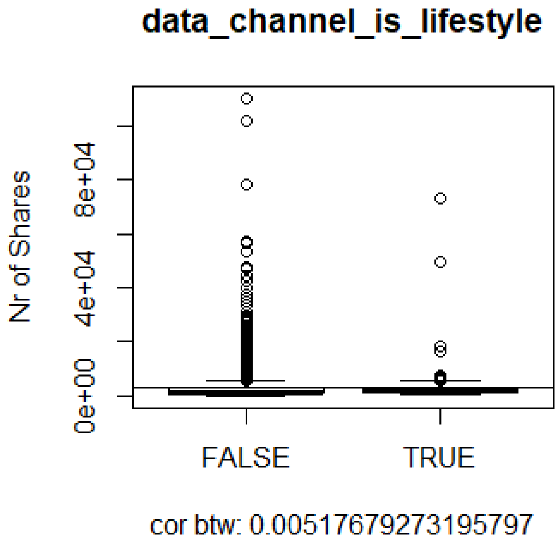


cor btw: -0.00645469705511627

data_channel_is_socmed

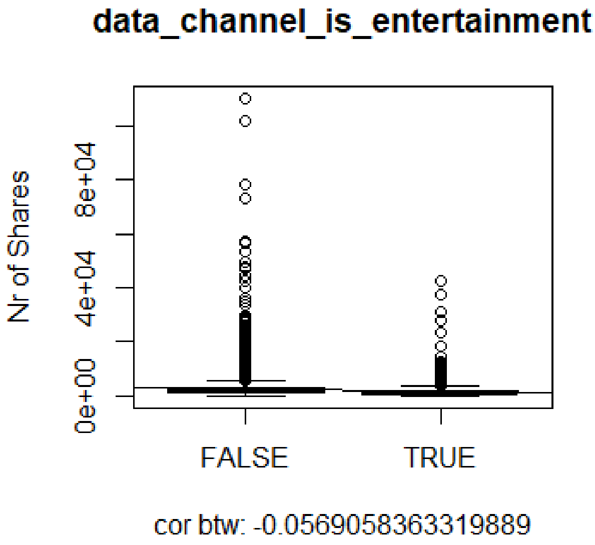


cor btw: 0.0309262366130622



data_channel_is_lifestyle

cor btw: 0.00517679273195797



data_channel_is_entertainment

cor btw: -0.0569058363319889

III. R Codes of the project;

```
setwd("E:/dersler/Statistics 1/main assignment")

library(tidyverse)

library(foreign)

library(nortest)

library(Hmisc)

library(arsenal)

library(psych)

library(corrplot)

library(car)

library(glmnet)

library(randtests)

library(lmtest)

library(caret)

library(e1071)


#Q1

news <- read.csv("alldata_onlinenews_25.csv", sep=";")

test <- read.csv("OnlineNewsPopularity_test.csv", sep=";")

names(test)

news <- subset(news, select = -c(X, url,timedelta))

names(news)

test <- subset(test, select = -c(X, url,timedelta))

names(test)

sum(news %in% test)


fulldata <- rbind(news, test)

str(fulldata)
```

```
which(is.na(fulldata))


str(news)
glimpse(news)
str(test)
summary(news)
head(news)


########################################
####change data types


sapply(news,class)
facnews <- news[,sapply(news, is.factor)]
intnews <- news[,sapply(news, is.integer)]


for(i in 1:62){
  if (count(unique(news[i]))<=2){
    print(unique(news[i]))
  }
}


a<- c("data_channel_is_lifestyle"
,"data_channel_is_entertainment"
,"data_channel_is_bus"
,"data_channel_is_socmed"
,"data_channel_is_tech"
,"data_channel_is_world"
,"weekday_is_monday"
,"weekday_is_tuesday"
,"weekday_is_wednesday"

,"weekday_is_thursday"

,"weekday_is_friday"

,"weekday_is_saturday"

,"weekday_is_sunday"

,"is_weekend")


facnews2 <- news[a]

facnews2[,a] <- apply(facnews2[,a], 2, function(x) as.logical(x))


intnews2 <- news[,!(names(news)%in%a)]

asNumeric <- function(x) as.numeric(as.character(x))

factorsNumeric <- function(d) modifyList(d, lapply(d[sapply(d,is.factor)], as.numeric))

intnews2 <- factorsNumeric(intnews2)

str(facnews2)

str(intnews2)

news2 <- cbind(intnews2, facnews2)

news2 <- news2[names(test)]

str(news2)


test2 <- test

test2[,a] <- apply(test2[,a], 2, function(x) as.logical(x))

test2[,!(names(test2)%in%a)] <-factorsNumeric(test2[,!(names(test2)%in%a)])

str(test2)


fulldata2 <- rbind(news2, test2)

################################################


#shares
```

```
head(sort(news2$shares, decreasing = T),5)

summary(news2$shares)

ggplot(news2, aes(shares))+geom_histogram()


par(mfrow=c(2,2))

for(i in 1:48){

  hist(intnews2[,i], main=names(intnews2)[i], xlab = names(intnews2)[i])

}


#####################################

#cheking normality of variables

normality <- function(x){

  lillie.test(x)

  shapiro.test(x)

  if((lillie.test(x)$p.value >0.05) &

     (shapiro.test(x)$p.value >0.05)){

    print("Data is normally distributed")

  } else {

    print("Not normally distributed, Reject null Hypothesis")

  }

}


for(i in 1:ncol(intnews2)){

  print(names(intnews2[i]))

  normality(intnews2[,i])

}

####################################

#### explore factor variables and probabilities
```

```
for(i in 1:ncol(facnews2)){
  tbl= table(facnews2[i])
  tbl = cbind(tbl,round(prop.table(tbl),2))
  colnames(tbl) <- c(names(facnews2)[i], "prob in column")
  print(tbl[2,])
}

par(mfrow=c(1,2))
for(i in 1:ncol(facnews2)){
  boxplot(intnews2$shares~facnews2[,i], xlab=paste("cor btw:",(cor(intnews2$shares, faknews2[i]))),
main=names(faknews2)[i], ylab='Nr of Shares')
  abline(lm(intnews2$shares~facnews2[,i]))
}

par(mfrow=c(1,1))
data_channel <- sapply(lapply(1:6, function(x) facnews2[,x]==1), sum)
pie(data_channel, labels = names(facnews2[1:7]), main = "Distribution of Data Channel")

days <- sapply(lapply(7:13, function(x) facnews2[,x]==1), sum)
pie(days, labels = names(faknews2[7:13]), main = "Share Intensity btw Days")

##################################
##### pairwise comparisons

par(mfrow=c(2,2))
for(i in 1:(ncol(news2)-1)){
  scatter.smooth(x=news2[,i], y=news2$shares,
        xlab = names(news2)[i],
        ylab = "Nr of shares",
        main=paste(names(news2)[i],"~shares"))
}


pairs(data=intnews2[1:6], intnews2$shares~.)
par(mfrow=c(1,1))
corrplot(cor(intnews2), method = "number",order='hclust', type="upper")
round(cor(intnews2, intnews2$shares),2)


par(mfrow=c(2, 2))
for(i in 1:ncol(intnews2)){
  plot(density(intnews2[,i]), main=paste("Density Plot:",names(news2)[i]), ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(intnews2[,i]), 2)))
  polygon(density(intnews2[,i]), col="red")
}



#####################################
#Q2


### checking multicollinearity
alias( lm(shares~.,news2) )
##Nonzero entries in the "complete" matrix show that those terms are linearly dependent on UseMonthly.
##This means they're highly correlated, so I need to solve this problem.



fulldata3 <- fulldata2
```

```
fulldata3$is_weekday <- fulldata3$weekday_is_friday +
  fulldata3$weekday_is_monday + fulldata3$weekday_is_thursday+
  fulldata3$weekday_is_tuesday+ fulldata3$weekday_is_wednesday
unique(fulldata3$is_weekday)
str(fulldata3)
sum(fulldata3$is_weekday , fulldata3$is_weekend)

#from now on is_weekday is has same data with is_weekend and other weekday datas
fulldata3 <- subset(fulldata3, select = -c(weekday_is_monday
                        ,weekday_is_tuesday
                        ,weekday_is_wednesday
                        ,weekday_is_thursday
                        ,weekday_is_friday
                        ,weekday_is_saturday
                        ,weekday_is_sunday
                        ,is_weekend))
news3 <- fulldata3[1:3000,]
test3 <- fulldata3[3001:13000,]
cor(news3$shares, news3$is_weekday)
modelday <- lm(log(shares)~., news3)
summary(modelday)             #12.9%

fulldata3 <- fulldata2
fulldata3 <- subset(fulldata3, select = -c(weekday_is_monday
                        ,weekday_is_wednesday
                        ,weekday_is_thursday
                        ,weekday_is_saturday
                        ,weekday_is_sunday
```

```
                                        ))

news3 <- fulldata3[1:3000,]
test3 <- fulldata3[3001:13000,]
modelday <- lm(log(shares)~., news3)
summary(modelday)                    #13.13%
round(vif(modelday),1)

# this works better

for(i in 1:(ncol(intnews2)-1)){
  if (abs(cor(intnews2$shares, intnews2[i]))>0.05){
    cn <- cor(intnews2$shares, intnews2[i])
    print(cor(intnews2$shares, intnews2[i]))
  }
}

# add log to shares

fulldata3$shares <- log(fulldata3$shares)
news3 <- fulldata3[1:3000,]
test3 <- fulldata3[3001:13000,]


##########################################
####create modelraw

modelraw <- lm(shares~num_hrefs+num_keywords+kw_min_avg+
             kw_max_avg+kw_avg_avg+LDA_02+LDA_03+LDA_04+
        global_subjectivity+avg_negative_polarity+
        title_subjectivity+data_channel_is_world+
        data_channel_is_socmed+weekday_is_tuesday+
        weekday_is_friday+is_weekend, news3)
summary(modelraw)    #9.88 %

modelraw <- lm(shares~num_hrefs+log(num_keywords)+log(kw_min_avg)+
          log(kw_max_avg)+kw_avg_avg+LDA_03+LDA_04+

          title_subjectivity+data_channel_is_world+
          data_channel_is_socmed+weekday_is_tuesday+
          weekday_is_friday+is_weekend, news3)
summary(modelraw)   #10.51%



#delete some of insignificant variables and try again, after
#several tries, I found one significant model
modelraw <- lm(shares~num_hrefs+num_keywords+
        kw_avg_avg+LDA_03+kw_max_avg+
        title_subjectivity+data_channel_is_world+
        data_channel_is_socmed+is_weekend+
        weekday_is_tuesday, news3)
summary(modelraw)   #9.63%
#####################################
#######create stepwise models raw data

#This time, I try to find a model using stepwise method without deleting days and taking log of shares;
modelfull <- lm(shares ~ .,news2)
modelnull <- lm(shares~1, news2)
```

```
summary(modelfull)
step(modelfull, direction='back')
modelsb <- lm(formula = shares ~ n_tokens_content + num_hrefs +
        average_token_length + data_channel_is_entertainment
       + data_channel_is_world + kw_min_min + kw_avg_min +
        kw_avg_max + kw_avg_avg + weekday_is_tuesday +
        LDA_04 + global_subjectivity + max_negative_polarity
       + title_subjectivity + abs_title_subjectivity, data = news2)
summary(modelsb)
step(modelfull, direction='both')
modelst <- lm(formula = shares ~ n_tokens_content + num_hrefs + average_token_length +
        data_channel_is_entertainment + data_channel_is_world + kw_min_min +
        kw_avg_min + kw_avg_max + kw_avg_avg + weekday_is_tuesday +
        LDA_04 + global_subjectivity + global_rate_positive_words +
        max_negative_polarity + title_subjectivity + abs_title_subjectivity +
        weekday_is_friday + max_positive_polarity, data = news2)
summary(modelst)
step(modelnull,
  scope = list(upper=modelfull),
  direction="forward",
  data=news2)
modelsf <- lm(formula = shares ~ kw_avg_avg + num_hrefs + data_channel_is_entertainment +
        average_token_length + LDA_04 + global_subjectivity + weekday_is_tuesday +
        max_negative_polarity + data_channel_is_tech + global_rate_positive_words +
        weekday_is_friday + title_subjectivity + abs_title_subjectivity +
        n_tokens_content + max_positive_polarity, data = news2)
summary(modelsf)
```

```
######################################
#######create stepwise models based on log(shares)


modelfull2 <- lm(shares ~ .,news3)
modelnull2 <- lm(shares~1, news3)
summary(modelfull2)             #13.13%
step(modelfull2, direction='back')
modelsb2 <- lm(formula = shares ~ n_tokens_content + n_non_stop_words + num_hrefs +
            average_token_length + data_channel_is_lifestyle + data_channel_is_entertainment +
            data_channel_is_bus + data_channel_is_world + kw_min_min +
            kw_avg_min + kw_min_max + kw_avg_max + kw_max_avg + kw_avg_avg +
            self_reference_min_shares + self_reference_max_shares + self_reference_avg_sharess +
            weekday_is_tuesday + weekday_is_friday + is_weekend + LDA_00 +
            LDA_03 + global_subjectivity + rate_negative_words + max_positive_polarity +
            title_subjectivity + abs_title_subjectivity, data = news3)
summary(modelsb2)        #13.46%
step(modelfull2, direction='both')
modelst2 <- lm(formula = shares ~ n_tokens_content + n_non_stop_words + num_hrefs +
            average_token_length + data_channel_is_lifestyle + data_channel_is_entertainment +
            data_channel_is_bus + data_channel_is_world + kw_min_min +
            kw_avg_min + kw_min_max + kw_avg_max + kw_max_avg + kw_avg_avg +
            self_reference_min_shares + self_reference_max_shares + self_reference_avg_sharess +
            weekday_is_tuesday + weekday_is_friday + is_weekend + LDA_00 +
            LDA_03 + global_subjectivity + rate_negative_words + max_positive_polarity +
            title_subjectivity + abs_title_subjectivity, data = news3)
summary(modelst2)        #13.46%
step(modelnull2,
   scope = list(upper=modelfull2),
    direction="forward",
    data=news3)
modelsf2 <-  lm(formula = shares ~ kw_avg_avg + data_channel_is_entertainment +
              is_weekend + num_hrefs + average_token_length + kw_min_min +
              kw_min_max + data_channel_is_tech + data_channel_is_socmed +
              kw_max_avg + title_subjectivity + weekday_is_friday + abs_title_subjectivity +
              rate_negative_words + self_reference_max_shares + kw_avg_min +
              LDA_03 + max_positive_polarity + global_subjectivity + n_tokens_content +
              n_non_stop_words + weekday_is_tuesday + LDA_04 + self_reference_min_shares +
              self_reference_avg_sharess + data_channel_is_world, data = news3)

summary(modelsf2)    # 13.4%



round(vif(modelfull2),1)


anova(modelsb, modelst, modelraw, modelsb2, modelsf2,modelst2)



######################################
#######create model with lasso


X <- model.matrix(modelfull2)[,-1]
lasso <- glmnet(X, news3$shares)
lasso1 <- cv.glmnet(X, news3$shares, alpha = 1)
lasso1$lambda
lasso1$lambda.min
lasso1$lambda.1se
plot(lasso1)
```

```
coef(lasso1, s = "lambda.min")
coef(lasso1, s = "lambda.1se")
modellassoraw <- lm(shares~num_hrefs+average_token_length+num_keywords
        +data_channel_is_entertainment+data_channel_is_world+data_channel_is_socmed+
         data_channel_is_tech+kw_min_min+kw_min_max+
         kw_max_avg+kw_avg_avg+self_reference_max_shares+weekday_is_tuesday+
         is_weekend+global_subjectivity+rate_negative_words+
         title_subjectivity,news3)

modellasso <- lm(shares~num_hrefs+average_token_length
        +data_channel_is_entertainment+data_channel_is_socmed+
         data_channel_is_tech+kw_min_min+kw_min_max+
         I(kw_max_avg^2)+kw_avg_avg+weekday_is_tuesday+
         is_weekend+I(rate_negative_words^(2))+
         I(title_subjectivity^5),news3)
summary(modellasso)   #%12.53
residualPlots(modellasso, plot=F)

anova(modelraw, modelsb,modelsf, modelst,modellasso)



################################################
###################################################
#feature scaling

normalizevar <- function(x) {
  ((x - mean(x))/(max(x)-min(x)))
}
```

```
scalevar <- function(x){
  x/max(x)
}
fd1 <- fulldata2



fd1[c("kw_min_min", "kw_avg_min", "kw_min_avg")] <- abs(fd1[c("kw_min_min", "kw_avg_min",
"kw_min_avg")]
)
fd1[c("n_tokens_content", "kw_max_min", "kw_avg_min", "kw_min_min",
      "kw_min_avg","kw_min_max","kw_max_max", "kw_avg_max",
      "kw_max_avg", "kw_avg_avg","self_reference_min_shares",
      "self_reference_max_shares","self_reference_avg_sharess", "shares")] <-
log(fd1[c("n_tokens_content", "kw_max_min", "kw_avg_min", "kw_min_min",
                                                            "kw_min_avg","kw_min_max","kw_max_max",
"kw_avg_max",
                                                            "kw_max_avg",
"kw_avg_avg","self_reference_min_shares",

"self_reference_max_shares","self_reference_avg_sharess", "shares")]+1)
fd1[c("n_tokens_title","num_hrefs","num_self_hrefs","num_imgs",
      "num_videos","average_token_length","num_keywords")] <-
scalevar(fd1[c("n_tokens_title","num_hrefs","num_self_hrefs","num_imgs",
                                                     "num_videos","average_token_length","num_keywords")])
n1 <- fd1[1:3000,]
t1 <- fd1[3001:13000,]
m1 <- lm(shares~.,n1)
summary(m1)             # 11.6%
step(m1, direction='both')


sb1 <- lm(formula = shares ~ n_tokens_content + n_non_stop_words + num_hrefs +
        num_self_hrefs + average_token_length + num_keywords + data_channel_is_lifestyle +
        data_channel_is_entertainment + data_channel_is_bus + data_channel_is_world +
        kw_min_min + kw_avg_min + kw_avg_max + kw_max_avg + kw_avg_avg +
        self_reference_min_shares + self_reference_max_shares + weekday_is_monday +
        weekday_is_tuesday + weekday_is_wednesday + weekday_is_thursday +
        weekday_is_friday + LDA_00 + global_subjectivity + title_subjectivity +
        abs_title_subjectivity, data = n1)
summary(sb1)        # 12%
round(vif(sb1),1)



fd1[c("n_tokens_content", "kw_max_min", "kw_avg_min", "kw_min_min",
   "kw_min_avg","kw_min_max","kw_max_max", "kw_avg_max",
   "kw_max_avg", "kw_avg_avg","self_reference_min_shares",
   "self_reference_max_shares","self_reference_avg_sharess",
   "shares","n_tokens_title","num_hrefs","num_self_hrefs","num_imgs",
   "num_videos","average_token_length","num_keywords")] <- log(fd1[c("n_tokens_content",
"kw_max_min", "kw_avg_min", "kw_min_min",
                                  "kw_min_avg","kw_min_max","kw_max_max", "kw_avg_max",
                                  "kw_max_avg", "kw_avg_avg","self_reference_min_shares",
                                  "self_reference_max_shares","self_reference_avg_sharess",

"shares","n_tokens_title","num_hrefs","num_self_hrefs","num_imgs",
                                  "num_videos","average_token_length","num_keywords")]+1)
n1 <- fd1[1:3000,]
t1 <- fd1[3001:13000,]
m1 <- lm(shares~.,n1)
summary(m1)         # 11.22%
step(m1, direction='both')
```

```
sb2 <- lm(formula = shares ~ n_tokens_content + n_non_stop_words + num_hrefs +
          num_self_hrefs + average_token_length + num_keywords + data_channel_is_lifestyle +
          data_channel_is_entertainment + data_channel_is_bus + data_channel_is_world +
          kw_min_min + kw_avg_min + kw_avg_max + kw_max_avg + kw_avg_avg +
          self_reference_min_shares + self_reference_max_shares + weekday_is_monday +
          weekday_is_tuesday + weekday_is_wednesday + weekday_is_thursday +
          weekday_is_friday + LDA_00 + global_subjectivity + rate_positive_words +
          max_positive_polarity + title_subjectivity + abs_title_subjectivity,
        data = n1)

sb2 <- lm(formula = shares ~ n_tokens_content  + num_hrefs +
          average_token_length + num_keywords + data_channel_is_lifestyle +
          data_channel_is_entertainment + data_channel_is_bus + data_channel_is_world +
          kw_min_min   + kw_max_avg + kw_avg_avg +
          self_reference_max_shares + weekday_is_monday +
          weekday_is_tuesday + weekday_is_wednesday + weekday_is_thursday +
          weekday_is_friday + LDA_00 + global_subjectivity  +
          title_subjectivity + abs_title_subjectivity,
        data = n1)

summary(sb2)            # 11.23%
residualPlots(sb2, plot=F)


#############################################
################################################
###############################################
##############################################
####cheking assumptions and transform predictors
```

```
#Q3

#######################################
#####modelraw assumptions

modelraw <- lm(shares~num_hrefs+num_keywords+
         kw_avg_avg+LDA_03+kw_max_avg+
         title_subjectivity+data_channel_is_world+
         data_channel_is_socmed+is_weekend+
         weekday_is_tuesday, news3)

summary(modelraw)     #9.63%
residualPlots(modelraw, plot=F, type = "rstudent")

par(mfrow=c(1,1))
plot(modelraw, which = 2)
Stud.residuals <- rstudent(modelraw)
yhat <- fitted(modelraw)
par(mfrow=c(1,2))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)
plot(yhat, Stud.residuals^2)
abline(h=4, col=2, lty=2)
ncvTest(modelraw)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)
leveneTest(rstudent(modelraw)~yhat.quantiles)
boxplot(rstudent(modelraw)~yhat.quantiles)
```

```
residualPlot(modelraw, type='rstudent')

residualPlots(modelraw, plot=F, type = "rstudent")

plot(rstudent(modelraw), type='l')


round(vif(modelraw),1)


######################################

#####modelst2 assumptions


modelst2 <- lm(formula = shares ~ I(n_tokens_content^2) + num_hrefs +

        average_token_length + data_channel_is_lifestyle + data_channel_is_entertainment +

        data_channel_is_bus + data_channel_is_world + kw_min_min +

        kw_avg_min + kw_min_max + kw_max_avg + kw_avg_avg +

        self_reference_max_shares +

        weekday_is_friday + is_weekend +

        I(LDA_00^4) +

        LDA_03 + max_positive_polarity +

        title_subjectivity + abs_title_subjectivity, data = news3)

summary(modelst2)     #13.42%


par(mfrow=c(1,1))

plot(modelst2, which = 2)

Stud.residuals <- rstudent(modelst2)

yhat <- fitted(modelst2)

par(mfrow=c(1,2))

plot(yhat, Stud.residuals)

abline(h=c(-2,2), col=2, lty=2)

plot(yhat, Stud.residuals^2)
```

```
abline(h=4, col=2, lty=2)

ncvTest(modelst2)

yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)

table(yhat.quantiles)

leveneTest(rstudent(modelst2)~yhat.quantiles)

boxplot(rstudent(modelst2)~yhat.quantiles)


residualPlot(modelst2, type='rstudent')

residualPlots(modelst2, plot=F, type = "rstudent")

plot(rstudent(modelst2), type='l')


round(vif(modelst2),1)


######################################

#####modellasso assumptions


summary(modellasso)

residualPlots(modellasso, plot=F)

par(mfrow=c(1,1))

plot(modellasso, which = 2)

Stud.residuals <- rstudent(modellasso)

yhat <- fitted(modellasso)

par(mfrow=c(1,2))

plot(yhat, Stud.residuals)

abline(h=c(-2,2), col=2, lty=2)

plot(yhat, Stud.residuals^2)

abline(h=4, col=2, lty=2)

ncvTest(modellasso)

yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
```

```
table(yhat.quantiles)

leveneTest(rstudent(modellasso)~yhat.quantiles)

boxplot(rstudent(modellasso)~yhat.quantiles)


residualPlot(modellasso, type='rstudent')

residualPlots(modellasso, plot=F, type = "rstudent")

plot(rstudent(modellasso), type='l')


round(vif(modellasso),1)



#####################################

#########################################

######################################

####transform predictors again and create modelst3


fulldata4 <- fulldata3

fulldata4$min_negative_polarity <- abs(fulldata4$min_negative_polarity)

fulldata4$max_negative_polarity <- abs(fulldata4$max_negative_polarity)

fulldata4$avg_negative_polarity <- abs(fulldata4$avg_negative_polarity)

fulldata4$title_sentiment_polarity <- abs(fulldata4$title_sentiment_polarity)

fulldata4$kw_min_min <- abs(fulldata4$kw_min_min)



summary(fulldata4[,names(intnews2)])


news4 <- fulldata4[1:3000,]

test4 <- fulldata4[3001:13000,]
```

```
modelfull3 <- lm(shares~.,news4)
step(modelfull3, direction='both')

modelst3 <- lm(formula = shares ~ n_tokens_content + n_non_stop_words + num_hrefs +
               average_token_length + data_channel_is_lifestyle + data_channel_is_entertainment +
               data_channel_is_bus + data_channel_is_world + kw_min_min +
               kw_avg_min + kw_min_max + kw_avg_max + kw_max_avg + kw_avg_avg +
               self_reference_min_shares + self_reference_max_shares + self_reference_avg_sharess +
               weekday_is_tuesday + weekday_is_friday + is_weekend + LDA_00 +
               LDA_03 + global_subjectivity + rate_negative_words + max_positive_polarity +
               title_subjectivity + abs_title_subjectivity, data = news4)

summary(modelst3)    #13.46%

modelst3 <- lm(formula = shares ~ I(n_tokens_content^2) + num_hrefs +
               average_token_length + data_channel_is_lifestyle + data_channel_is_entertainment +
               data_channel_is_bus + data_channel_is_world + kw_min_min +
               kw_avg_min + kw_min_max  + log(kw_max_avg+1) + I(kw_avg_avg^2) +
                self_reference_max_shares
                + weekday_is_friday + is_weekend +
               I(LDA_03^2)  + I(rate_negative_words^2) + max_positive_polarity +
               title_subjectivity + abs_title_subjectivity, data = news4)

summary(modelst3)    #13.46%


######################################
#####modelst3 assumptions
```

```
residualPlots(modelst3, plot=F)

par(mfrow=c(3,2))

plot(modelst3, which = 2)

Stud.residuals <- rstudent(modelst3)

yhat <- fitted(modelst3)

plot(yhat, Stud.residuals)

abline(h=c(-2,2), col=2, lty=2)

plot(yhat, Stud.residuals^2)

abline(h=4, col=2, lty=2)

ncvTest(modelst3)

yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)

table(yhat.quantiles)

leveneTest(rstudent(modelst3)~yhat.quantiles)

boxplot(rstudent(modelst3)~yhat.quantiles)


residualPlot(modelst3, type='rstudent')

residualPlots(modelst3, plot=F, type = "rstudent")

plot(rstudent(modelst3), type='l')


round(vif(modelst3),1)


#########################################

#Q4



######################################

#####create modelcv


modelcv <- train(
  shares ~ ., news3,
  method = "lm",
  trControl = trainControl(
    method = "cv", number = 10,
    verboseIter = TRUE
  )
)


summary(modelcv) # 13.13%


######################################
#####modelcv assumptions

residualPlots(modelcv$finalModel, plot=F, type="rstudent")
par(mfrow=c(1,1))
plot(modelcv$finalModel, which = 2)
Stud.residuals <- rstudent(modelcv$finalModel)
yhat <- fitted(modelcv$finalModel)
par(mfrow=c(1,2))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)
plot(yhat, Stud.residuals^2)
abline(h=4, col=2, lty=2)
ncvTest(modelcv2$finalModel)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)
leveneTest(rstudent(modelcv$finalModel)~yhat.quantiles)
```

```
boxplot(rstudent(modelcv$finalModel)~yhat.quantiles)


residualPlot(modelcv$finalModel, type='rstudent')
residualPlots(modelcv$finalModel, plot=F, type = "rstudent")
plot(rstudent(modelcv$finalModel), type='l')

round(vif(modelcv$finalModel),1)

########################################
######################################
#Q5

######################################
#####modelraw predictions



predraw <- predict(modelraw, test3)
actual_pred <- data.frame(cbind(actuals=test3$shares, predicteds=predraw))
(correlation_accuracy <- cor(actual_pred))
data.frame(
  R2 = R2(predraw, test3$shares),       #7.46%
  RMSE = RMSE(predraw, test3$shares),
)

######################################
#####modelst2 predictions


predst2 <- predict(modelst2, test3)
actual_pred <- data.frame(cbind(actuals=test3$shares, predicteds=predst2))
```

```
(correlation_accuracy <- cor(actual_pred))
data.frame(
  R2 = R2(predst2, test3$shares),       #0.95%
  RMSE = RMSE(predst2, test3$shares),
)

######################################
#####modellasso predictions


predlasso <- predict(modellasso, test3)
actual_pred <- data.frame(cbind(actuals=test3$shares, predicteds=predlasso))
(correlation_accuracy <- cor(actual_pred))
data.frame(
  R2 = R2(predlasso, test3$shares),     #4.19%
  RMSE = RMSE(predlasso, test3$shares),
)

######################################
#####modelst3 predictions


predst3 <- predict(modelst3, test4)
actual_pred <- data.frame(cbind(actuals=test4$shares, predicteds=predst3))
(correlation_accuracy <- cor(actual_pred))
data.frame(
  R2 = R2(predst3, test4$shares),       #6.77%
  RMSE = RMSE(predst3, test4$shares),
)
sqrt(mean((test4$shares - predst3)^2))
```

```
######################################
#####modelcv predictions
predcv <- predict(modelcv, test3)
actual_pred <- data.frame(cbind(actuals=test3$shares, predicteds=predcv))
(correlation_accuracy <- cor(actual_pred))
data.frame(
  R2 = R2(predcv, test3$shares),       #9.18%
  RMSE = RMSE(predcv, test3$shares),
)
```