Online News Popularity

Konstantinos Chronis BAFT1502

M.Sc. Business Analytics

Statistics for Business Analytics I

Professor: I. Ntzoufras

Data set number: 3

Contents

Abstract

Internet fame comes on like an earthquake, with little warning. In a matter of hours, a post can go viral and be shared up to a tremendous amount of times. Then it (usually) recedes into a very long, thin afterlife. But what are the reasons that can really make a post climb up the charts in popularity terms?

In this analysis this issue was tried to be addressed having acquired the necessary information about the elements that may have a effect on the target variable of the shares which are the keywords, rates of positive/negative words, number of links, images, videos, the closeness to LDA[1] topic, the polarity and the absolute levels of subjectivity and polarity, the day of the week and the content of the post.

The results that came out were really interesting and unexpected. One major reason that can influence the shares of a post is the day of the week, especially if it is posted during the weekend. Other important factor is the data channel of the article, specifically if it is related to the social media, entertainment and world news. More details regarding the results are to be found at the conclusion of the analysis.

[1] Models/unobserved groups that explain why some parts of the data are similar.

Introduction

For our analysis we collected the data referring to characteristics of the popular website of Mashable (www.mashable.com). A total number of 10000 observations from 61 different variables (58 predictive attributes, 2 non-predictive and 1 goal field) will be used for this analysis purposes. More specifically the variables[2] studied are:

0. url: URL of the article (non-explanatory)
1. timedelta: Days between the article publication and the dataset acquisition (non-explanatory)
2. n_tokens_title: Number of words in the title
3. n_tokens_content: Number of words in the content
4. n_unique_tokens: Rate of unique words in the content
5. n_non_stop_words: Rate of non-stop words in the content
6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content
7. num_hrefs: Number of links
8. num_self_hrefs: Number of links to other articles published by Mashable
9. num_imgs: Number of images
10. num_videos: Number of videos
11. average_token_length: Average length of the words in the content
12. num_keywords: Number of keywords in the metadata
13. data_channel_is_lifestyle: Is data channel 'Lifestyle'?
14. data_channel_is_entertainment: Is data channel 'Entertainment'?
15. data_channel_is_bus: Is data channel 'Business'?
16. data_channel_is_socmed: Is data channel 'Social Media'?
17. data_channel_is_tech: Is data channel 'Tech'?
18. data_channel_is_world: Is data channel 'World'?
19. kw_min_min: Worst keyword (min. shares)
20. kw_max_min: Worst keyword (max. shares)
21. kw_avg_min: Worst keyword (avg. shares)
22. kw_min_max: Best keyword (min. shares)
23. kw_max_max: Best keyword (max. shares)
24. kw_avg_max: Best keyword (avg. shares)

[2] See Reference 1 in the Appendix

25. kw_min_avg: Avg. keyword (min. shares)
26. kw_max_avg: Avg. keyword (max. shares)
27. kw_avg_avg: Avg. keyword (avg. shares)
28. self_reference_min_shares: Min. shares of referenced articles in Mashable
29. self_reference_max_shares: Max. shares of referenced articles in Mashable
30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable
31. weekday_is_monday: Was the article published on a Monday?
32. weekday_is_tuesday: Was the article published on a Tuesday?
33. weekday_is_wednesday: Was the article published on a Wednesday?
34. weekday_is_thursday: Was the article published on a Thursday?
35. weekday_is_friday: Was the article published on a Friday?
36. weekday_is_saturday: Was the article published on a Saturday?
37. weekday_is_sunday: Was the article published on a Sunday?
38. is_weekend: Was the article published on the weekend?
39. LDA_00: Closeness to LDA topic 0
40. LDA_01: Closeness to LDA topic 1
41. LDA_02: Closeness to LDA topic 2
42. LDA_03: Closeness to LDA topic 3
43. LDA_04: Closeness to LDA topic 4
44. global_subjectivity: Text subjectivity
45. global_sentiment_polarity: Text sentiment polarity
46. global_rate_positive_words: Rate of positive words in the content
47. global_rate_negative_words: Rate of negative words in the content
48. rate_positive_words: Rate of positive words among non-neutral tokens
49. rate_negative_words: Rate of negative words among non-neutral tokens
50. avg_positive_polarity: Avg. polarity of positive words 3
51. min_positive_polarity: Min. polarity of positive words
52. max_positive_polarity: Max. polarity of positive words
53. avg_negative_polarity: Avg. polarity of negative words
54. min_negative_polarity: Min. polarity of negative words
55. max_negative_polarity: Max. polarity of negative words

56. title_subjectivity: Title subjectivity
57. title_sentiment_polarity: Title polarity
58. abs_title_subjectivity: Absolute subjectivity level
59. abs_title_sentiment_polarity: Absolute polarity level
60. shares: Number of shares (target response)

The main and target variable of the study is the number of shares which measures the popularity of the site/post. We are interested to identify the ingredients of a successful post and what it takes to for it to become a viral. Is the day of the publishing of the article an important factor regarding the shares it receives? Does the channel it is in plays a role in the number it is reposted? Does it help if it contains images or videos? What about the positive or negative polarity of it? Is it contributing to its becoming a viral post? All these questions and hopefully more are to be answered after the analysis.

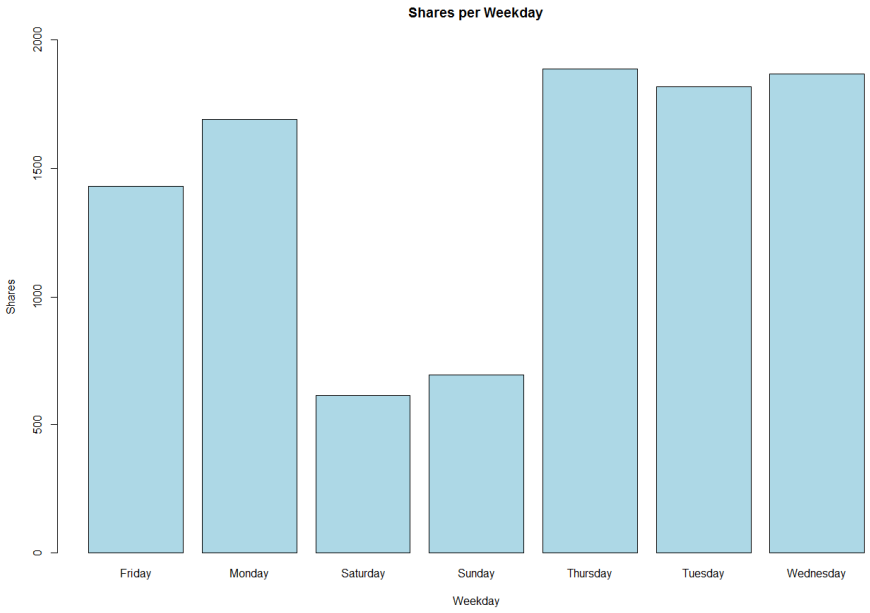
Descriptive Analysis

In order to start the analysis we will have to insert[3] the data in R where we will work on and check the structure[4] of it. In order to improve our analysis we consider essential to transform[5]-include the variables describing the content of the data channel (Lifestyle, Entertainment, etc.) in one categorical variable with the 6 possible categories of the content. Apart from that for similar purposes the variable referring to the days is to be transformed to one categorical variable which will include the 7 days. So after the drops of the 2 non predictive variables and the 2 new ones, we do have a number of 62 variables to examine. Now we can do some introductory descriptive measurements and check the normality of the target variable, as well. It would be wise to point out that the variables that refer to the number of shares, words, rates, links, images, videos, closeness to LDA topic, polarity and the absolute levels of subjectivity and polarity are all numerical variables. On the contrary, the new variables referring to the day of the week and the content of the post are categorical. Some indicative values that describe numerical variables in an acceptable manner are the mean, the median, the standard deviation, the quartiles, the skewness and the kurtosis.

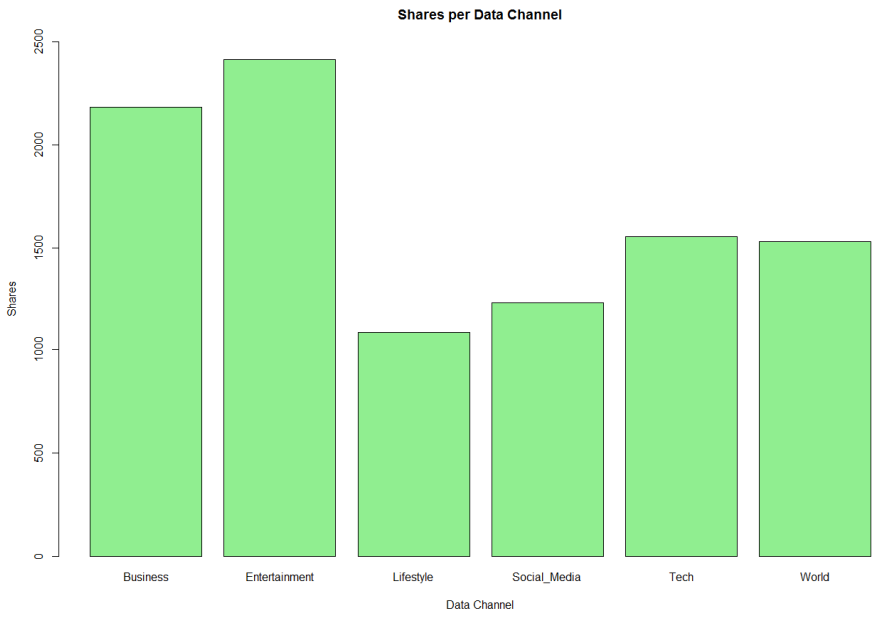
Now we will make some graphs to get a first glimpse of the data in the variables we consider important for our analysis and help us understand in a good way how each of the variables is distributed. The full description[6] of the data (using the sjPlot package) is shown in Table 1 in the Appendix. In that table we notice many interesting things about our data. For instance, the number of words in a title is from 3 to 19, or the average post contains one image in it (we select the median value for this conclusion, because the mean is influenced by the high outliers for this value). Another interesting point could be, for example, the minimum number of shares for a post is 5, while the post with the maximum value was shared 663600 times.

[3] See R Code in the Appendix, Command 1
[4] See R Code in the Appendix, Command 2
[5] See R Code in the Appendix, Command 3
[6] See R Code, Command 4 and Table 1 in the Appendix

Additionally, we would want to check the categorical variables and their connection to the target variable. To do that, we must compute[7] the plots shown below:



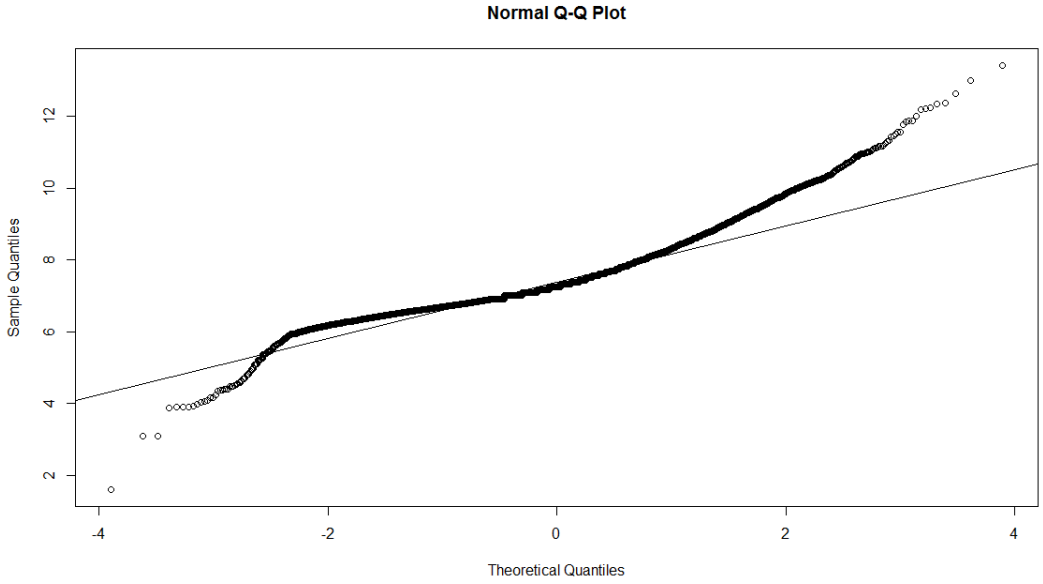
We observe that the days Saturday and Sunday seem to have a lower number of shares in comparison with the weekdays. So, it might be a good idea to use only the variable is_weekend in our models later.



Regarding the channels of the data, "Entertainment" seems to collect a higher numbers of shares than the others, while "Lifestyle" seems to have the lowest number.

[7] See R Code in the Appendix, Command 5

At this point, the target variable "Shares" should be examined more thoroughly. The first plot[8] with the quantiles is not allowing us to make conclusions so we should transform the data for the purpose of the plot using the log function. That plot is shown below:

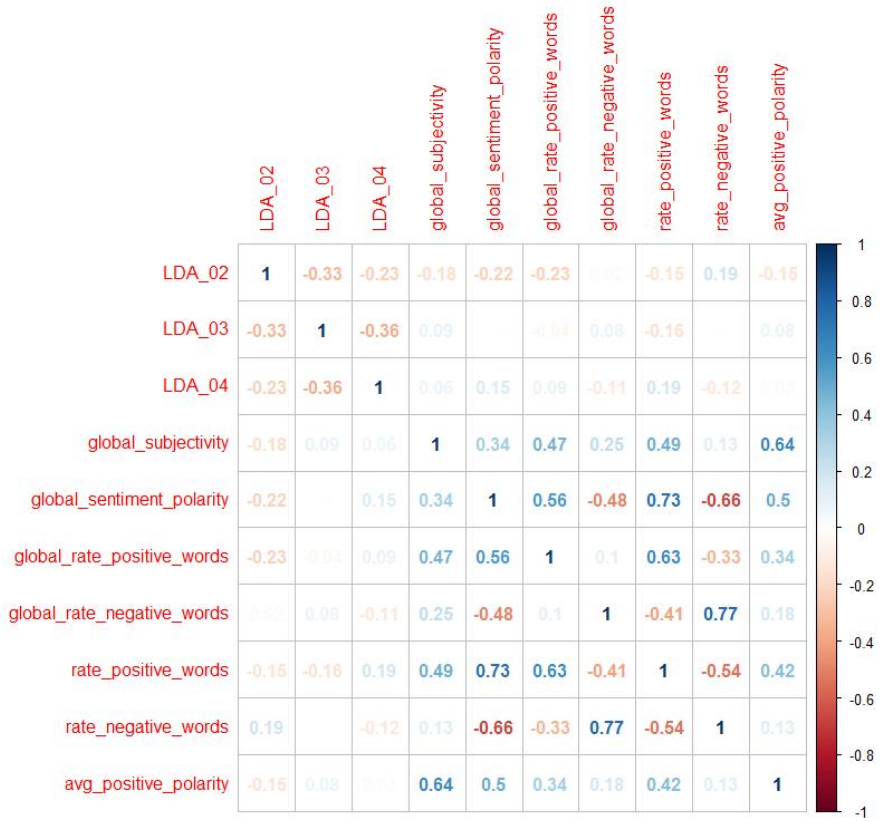


The normality is rejected clearly because the data are not aligning with the line except at same small fraction. We also run a normality test[9] and the p-value is lower than 2.2e-16 so we reject the null hypothesis that the SHARES variables follow a normal distribution.

[8] See Plot 1 in the Appendix
[9] See R Code in the Appendix, Command 6

Pairwise Comparisons

After reviewing the correlation R code[10] outputs the conclusions that arose in terms of high correlative variables are given in the Table 3 in the Appendix. By looking in the correlations between the variables and the p-values as well (the value under each correlation) we could select some of these variables to exclude from our model. Additionally we created the plots[11] of the variables in groups of 10 to have a visual perspective also. So it would be wise to choose the highlighted ones in Table 3 summarizing the high correlations for the linear regression along with the rest. Now have 47 variables to be used in the regression. Below is an indicative correlation plot between 10 variables:



We notice here that there is a high correlation between the variables global rate of negative words and the rate of negative words. The same applies for the positive ones.

[10] See R Code in the Appendix Command 8
[11] See R Code, Command 9 and Plot 2 in the Appendix

Predictive Models

After reviewing the relationships between the variables and extracting the high-correlated ones we are now at the point to attempt to create the first regression model.

First we could to do some assumption tests that have a meaningful insight, to have a clearer view in our data. We check, for instance, if the average post has 1400 shares (again here we choose the median value and not the mean because we do not want the effect of the extreme outliers). To accomplish that we do the Wilcoxon test[12] and with the p-value < 2.2e-16 we reject the Null hypothesis that the average post is best described with the value 1400 of posts. (We do the test in the Shares variables without the log because it is meaningless to run the Wilcoxon in a log function)

We should now compute our first model[13] but if we view the summary[14], the results are not satisfactory at all. Taking into account that the range of the SHARES variable is far greater than the others' we could try and log the variable. As it turns out the results[15] are much better now so for now we will proceed as mentioned above.

We will now use the Least Absolute Shrinkage and Selection Operator to proceed with the selection of the appropriate variables. The full procedure consists of the following steps:

1. Run LASSO for the dataset
2. Plot the regularization paths
3. Implement K-fold regularization
4. Estimate the coefficients

It should be mentioned that in the LASSO procedure[16] we will include all the variables of the dataset to make sure that we have excluded the right ones after the review of the correlation table and plots.

After that we will do the stepwise regression[17] in the model in an attempt to make it even simpler but with the same predictive capabilities. The stepwise procedure is adding and removing covariates step by step from a given model. (min AIC in our case).The procedure is

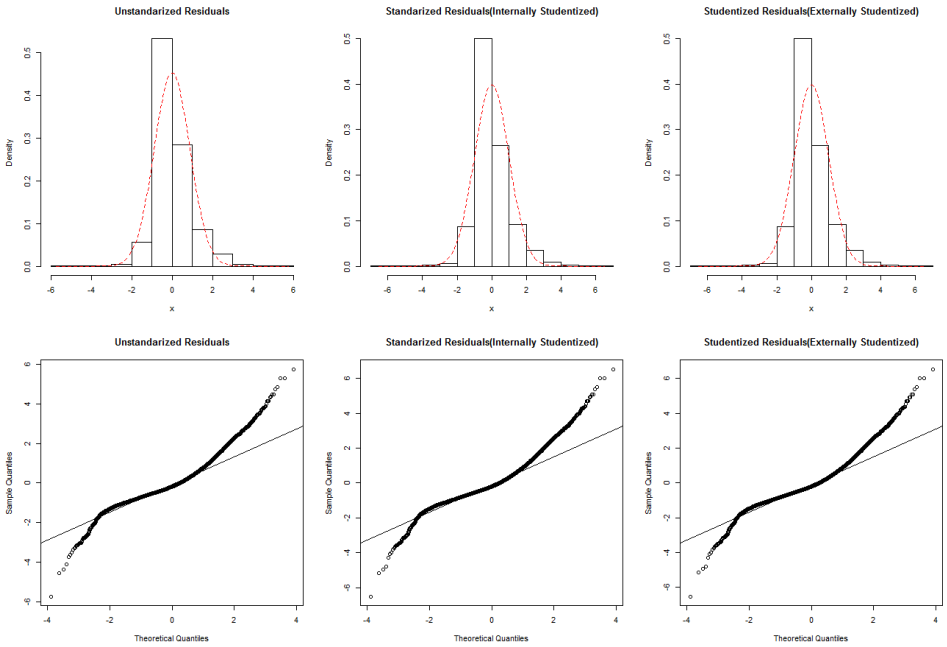
[12] See R Code in the Appendix, Command 10
[13] See R Code in the Appendix, Command 11
[14] See Summary 1 in the Appendix
[15] See Summary 2 in the Appendix

[16] See R Code, Commands 13,14,15, Plot 3 and Summaries 4,5 in the Appendix
[17] See R Code, Commands 16 and Summary 6 in the Appendix

stopped when no other improvement can be achieved. We will also include an analysis[18] of Variances for our model in comparison with the null model and the full one. The results are positive because with the p-values are below 2.2e-16 we reject the null hypotheses, so the model is better fitted than the model with only the constant or with all the variables.

We should now check for the multi-collinearity (which is the (statistically) high linear relationship between one explanatory with (some of) the rest of the explanatories) of the variables and to do that we will check the Variance Inflation Factors for each of the remaining variables. As we can see[19] none of them has a value bigger than 5 so we proceed with all of them. Now it is time to check the assumptions of the model.

1. Checking for normality (Comparison for different residuals using Qqplots[20])



It is clear the residuals do not pass the normality check. Too many observations stray out the line.

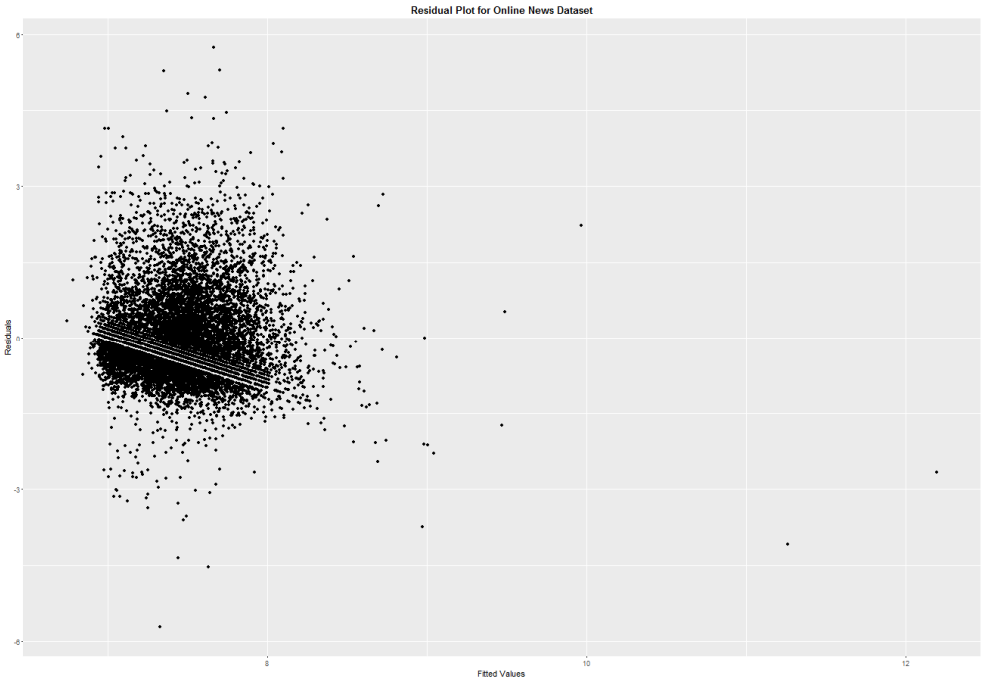
Additionally we did the normality test (Lillie KS)[21] to check the residuals. Again (with the p-value being 4.850351e-264) the normality is rejected.

[18] See R Code, Command 17 and Summary 7 in the Appendix
[19] See R Code, Command 18 and Summary 8 in the Appendix
[20] See R Code in the Appendix, Command 19
[21] See Plot 3 in the Appendix

2. Checking for independence of errors (using the randtests, lmtests packages)

Unfortunately, as we see[22] that the assumption of the randomness is not rejected as the p-value is 0.47. Furthermore, we run the Durbin-Watson test for serial correlation. The results as expected were that the null hypothesis that the true autocorrelation is equal or lower than 0 is not rejected. (p-value = 0.7758)

3. Homoscedasticity of errors (Plot[23] of the residuals versus fitted values)



We can see that although there does not seem to be a pattern involved the results are very difficult to explain.

4. Non-linearity check (using the car package residualPlots()[24] which plots[25] the residuals versus fitted values. Also computes a curvature test for the plot by adding a quadratic term and testing the quadratic to be zero. This is Tukey's test for nonadditivity when plotting against fitted values. This procedure tests the null hypothesis that the model is additive and no interaction is needed(Tukey,1949)

[22] See R Code, Command 20 and Summary 10 in the Appendix
[23] See R Code, Command 22 in the Appendix

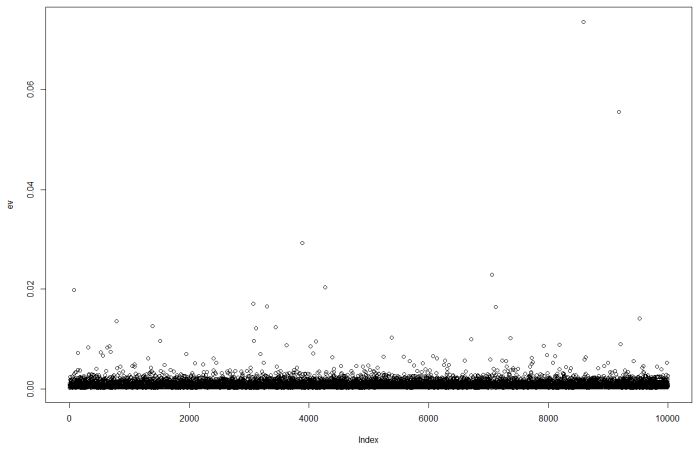
[24] See Reference 5
[25] See Plot 4 in the Appendix

Tukey's test results

	Test stat	Pr(>|t|)
num_hrefs	-0.695	0.487
num_keywords	-2.675	0.007
kw_avg_avg	-7.203	0.000
LDA_01	1.976	0.048
LDA_02	-2.265	0.024
is_weekend	-0.671	0.502
data_channel_is_entertainment	0.408	0.683
data_channel_is_socmed	-0.384	0.701
data_channel_is_world	-1.817	0.069
Tukey test	-6.831	0.000

Unfortunately, we can see at the results the Tukey test[26] is significant here (p-value = 0.00<0.05) so the null hypothesis that no interaction is needed is rejected. The other test included here are for the squared term of each variable. For example, the num_hrefs, is_weekend, data_channel_is_entertainment and data_channel_is_socmed are not significant so we can conclude that none of them has a quadratic relationship with the shares. We cannot say the same about the num_keywords,kw_avg_avg,LDA_01,LDA_02 and data_channel_is_world.

Now we will check for leverage points in our model. Most common measure of leverage is the hat value. In multiple regression, hat values measure the distance from the centroid point of all of the variables' (points of means).Below the plot[27] with the observations' hat values.



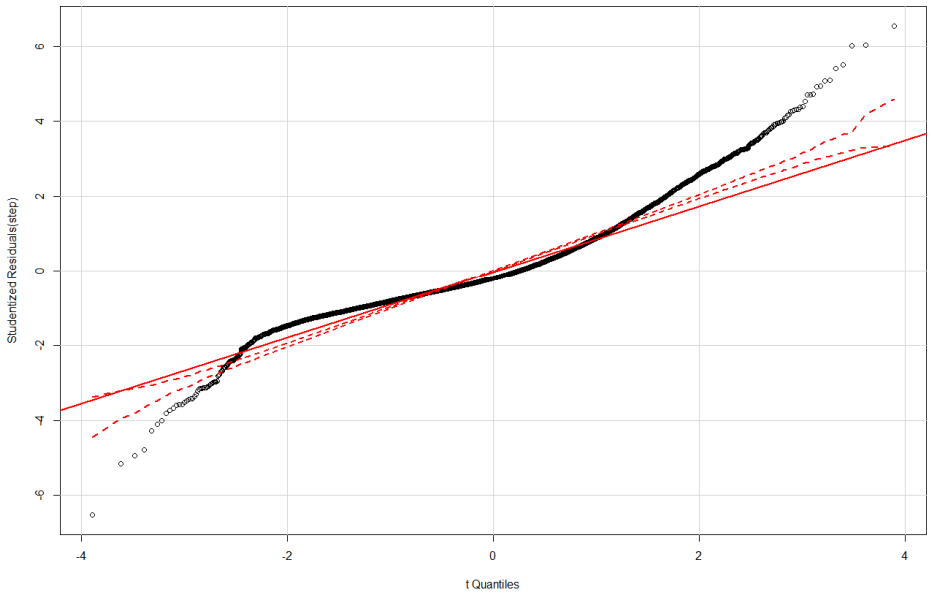
We notice that there are two observations with higher leverage than all the rest (~0.08 and ~0.06). These cases have high leverage, but not necessarily high influence.

[26] See R Code, Command 21 in the Appendix
[27] See R Code, Command 23 in the Appendix

Now is time to run some test[28] for outliers[29]. (The outlierTest()function in the car package gives Bonferroni p-value for the largest absolute studentized residuals. The Bonferroni p-value for the largest outlier is: p = 2np0 where p0 is the unadjusted p-value from a t-test with n - k - 2 degrees of freedom).The results which are shown below indicate that those observations are outliers but as of yet we have not assessed whether they influence the regression line.

	rstudent	unadjusted p-value	Bonferonni p
7875	6.535948	6.6293e-11	6.6293e-07
7319	-6.518618	7.4385e-11	7.4385e-07
2189	6.027661	1.7223e-09	1.7223e-05
6070	6.011859	1.8983e-09	1.8983e-05
222	5.497698	3.9425e-08	3.9425e-04
5630	5.399466	6.8372e-08	6.8372e-04
6625	-5.163550	2.4698e-07	2.4698e-03
265	5.090672	3.6330e-07	3.6330e-03
4548	5.073445	3.9770e-07	3.9770e-03
8482	-4.948690	7.5934e-07	7.5934e-03

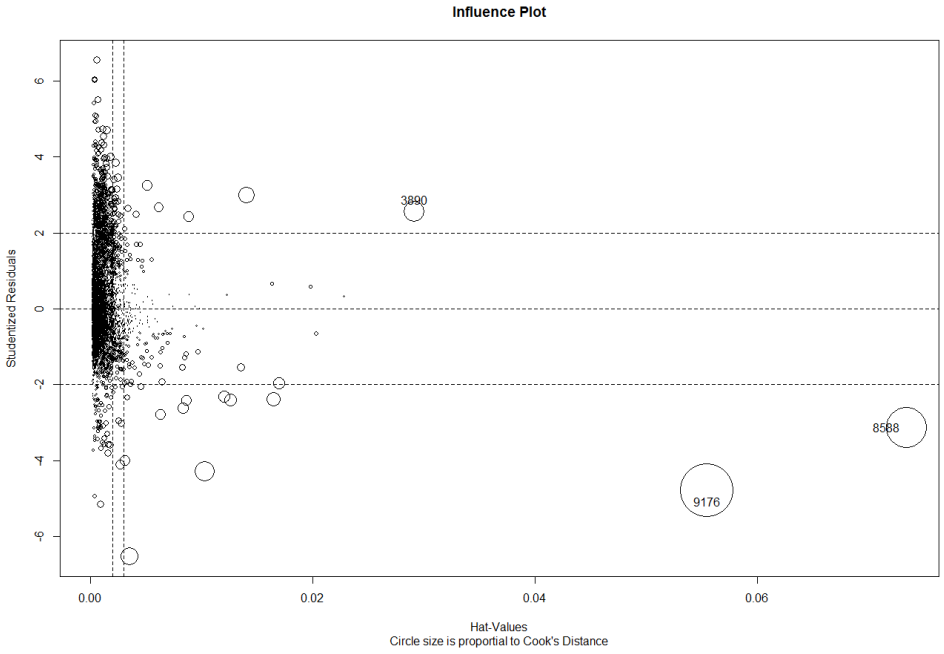
We can use a quantile comparison plots[30] to compare the distribution of a single variable to the t-distribution, assessing whether the distribution of the variable showed a departure from normality. Using the same technique, we can compare the distribution of the studentized residuals from our regression model to the t-distribution. Observations that stray outside of the 95% confidence envelope are statistically significant outliers.



We can see that many observations stray outside of the envelope thus being statistically significant outliers.

[28] See R Code, Command 23 in the Appendix
[29] See Reference 2
[30] See Reference 3

Checking[31] for influential points through the Influence Plot which displays studentized residuals, hat-values and Cook's D on a single plot. Cook's distance is the overall distance between the model coefficients with and without observations).In the plot the horizontal axis represents the hat-values; the vertical axis represents the studentized residuals; circles for each observation represent the relative size of the Cook's D. The radius is proportional to the square root of Cook's D, and thus the areas are proportional to the Cook's D.



We can see in this plot that two observations 9176 and 8588 have a high Cook's distance so they are influential points.

On to the next steps of the analysis, the Cross Validation[32] procedure[33] will provide us a better view if we selected the right model (in comparison for instance for a predictive model using only the stepwise regression without the LASSO). At the output we get the cross validation residual sums of squares (Overall MS), which is a corrected measure of prediction error averaged across all folds. For our model the Overall MS is 0.77 while in the other model (stepwise only) is 3.74.

[31] See R Code, Command 23 in the Appendix
[32] See Reference 4
[33] See R Code, Command 24,25 and Plots 5,6 in the Appendix

Thus, the **final model** in full form is:

Log(Shares) = 6196 + 0.005 x Number_of_links + 0.09 x Number_of_keywords + 0.0001 x Avg.keyword (avg shares) - 0.1875 x Closeness_to_LDA_topic_1 - 0.2831 x Closeness_to_LDA_topic_2 + 0.274 x Is_weekend – 0.2015 x Data_Channel_is_Entertainment + 0.2178 x Data_Channel_is_SocialMedia – 0.1461 x Data_Channel_is_World + ε

$\varepsilon \sim N(0,0.8805^2)$

Now proceeding to the final part of our analysis, we will partition[34] our data in order to be able to perform out-of-sample evaluation for our model. First, we test the performance of our model with the test set (a sampling vector of 15% of our dataset's shares values) we created. In order to do that we have to compute the Mean Square Error of our estimator model. The MSE of an estimator measures the average of the squares of the "errors", that is, the difference between the estimator and what is estimated.

The MSE computed[35] for our model fitted values and the actual values of our data set train set is 0.9367595, which given the fact that we have a total of 10000 observations is quite acceptable. Now we compute the MSE between the predicted values of our model and the actual ones in our test dataset. The result is 0.8157817 which is good given the total number of the shares.

Finally, we will test our model's predicted values' MSE in comparison to the actual values given to us in an external test set, we imported[36] for the purposes of this analysis. The output value of the MSE is 0.9236068 which is acceptable.

[34] See R Code, Command 26 in the Appendix
[35] See R Code, Command 27 in the Appendix
[36] See R Code, Commands 28 and 29 in the Appendix

Conclusion and Discussion

The main part of this analysis was to compute a good predictive model for understanding the parameters that make a post viral. The final model that is suggested has a rather bad fitted one (R^2adjusted = 0.10). The problems that we had to deal with were the large numbers of variables (=61) and observations (=10000), so the analysis was expected to suggest a bad fitted model.

Conclusively, the number of shares is affected in a positive manner by the day of publish, and more specifically whether it is on the weekend or not. This is not surprising as on the weekend people tend to spend more time in the social networks due to the fact that they have more spare time to themselves. Apart from that, the number of shares is affected in a positive way if its data channel is related to the Social Media. On the contrary, if the data channel is regarding the Entertainment or the World, it affects the number of shares in a negative manner. This could probably derives from the fact that we live in an era of the social networks and everything related to that is on the rise, while articles regarding entertainment and the world news are information obtained by the reader for his/her personal knowledge. Furthermore, the analysis led us to the fact that the shares is affected negatively if it closes to LDA topic models 0 or 1, and positively affected to the number of keywords in the metadata. Lastly, the number of links and the average keyword in a post with average shares seem to have a really low effect in the target variable of the shares value and thus, are included in the suggested model.

If possible, it would be more interesting and useful if there were logs containing the times of the posts because the time of the day that an article is published could play a huge role in the number of shares it gets. For instance, there seems to be a higher activity in the social networks at the later hours of the day (evening,night) so the number of shares could increase at the specific time periods. If this information is acquired or even without it, this analysis could be used by the media or other organizations to post the significant articles (related to important news, marketing opportunities, job openings, health care causes, charities, etc.) after taking into account important factors as the day or the time to further increase the promotion through the sharing of the articles they want.

References

1. K. Fernandes, P. Vinagre and P. Cortez. (2015*). A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News*. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal

2. Dave Armstrong. *Regression III Outliers and Influential Data*. University of Wisconsin Milwaukee Department of Political Science Lecture

3. Lehmann, E. L.; Casella, George (1998). *Theory of Point Estimation* (2nd ed.). New York: Springer.

4. Dr. Jon Starkweather, Research and Statistical Support consultant. *Cross Validation techniques in R: A brief overview of some methods, packages, and functions for assessing prediction models* article.

5. W. Holmes Finch, Jocelyn E. Bolin, Ken Kelley. *Multilevel Modeling Using* .Chapman and Hall Press, United States of America

APPENDIX

R Code:

Command 1: Importing the dataset and excluding the unnecessary variables.

```
>online_news <- read.csv2("alldata_onlinenews_03.csv")
>online_news$url<-NULL
> online_news$timedelta<-NULL
```

Command 2: Review of the structure of the dataset.

```
>str(online_news)
```

Command 3: Creating Factor variables from the dummy ones regarding the weekday and the data channel of the post.

```
> weekday<-data.frame(Monday = online_news$weekday_is_monday,Tuesday =
online_news$weekday_is_tuesday,Wednesday =
online_news$weekday_is_wednesday,Thursday =
online_news$weekday_is_thursday,Friday = online_news$weekday_is_friday,Saturday =
online_news$weekday_is_saturday,Sunday = online_news$weekday_is_sunday)
>weekday<-colnames(weekday)[max.col(weekday)]
>online_news<-cbind(online_news,weekday)

>data_channel<-data.frame(Lifestyle =
online_news$data_channel_is_lifestyle,Entertainment =
online_news$data_channel_is_entertainment,Business =
online_news$data_channel_is_bus,Social_Media =
online_news$data_channel_is_socmed,Tech = online_news$data_channel_is_tech,World =
online_news$data_channel_is_tech)
> data_channel<-colnames(data_channel)[max.col(data_channel)]
> online_news<-cbind(online_news,data_channel)
```

Command 4: HTML description of the dataset using the “sjPlot” package.

```
>install.packages("sjPlot")
>Library(sjPlot)
>sjt.df(online_news)
```

Command 5: Barplots of the factor variables with the target variable.

```
> barplot(table(online_news$weekday),ylim = c(0,2000),col = "lightblue",xlab =
"Weekday",ylab = "Shares",main = "Shares per Weekday")
> barplot(table(online_news$data_channel),ylim = c(0,2500),col = "lightgreen",xlab =
"Data Channel",ylab = "Shares",main = "Shares per Data Channel")
```

Command 6: Checking the normality of the target variable as shown to Plot 1.

```
> qqnorm(online_news$shares)
> qqline(online_news$shares)
> qqnorm(log(online_news$shares))
> qqline(log(online_news$shares))
>library(nortest)
>lillie.test(log(online_news$shares))
```

Command 7: Producing the frequency table of the target variable using the "sjPlot" package.

```
>sjt.frq(online_news$shares, variableLabels=list("Shares"), autoGroupAt=10)
```

Command 8: Producing the correlation matrix using the sjPlot package.

```
>sjt.corr(online_news[,1:60])
```

Command 9: Correlation comparisons between the variables in our dataset.

```
>library(corrPlot)
> newdatacor = cor(online_news[1:10])
> corrplot(newdatacor, method = "number")
> newdatacor = cor(online_news[11:20])
> corrplot(newdatacor, method = "number")
> newdatacor = cor(online_news[31:40])
```

```
> corrplot(newdatacor, method = "number")
> newdatacor = cor(online_news[41:50])
> corrplot(newdatacor, method = "number")
> newdatacor = cor(online_news[51:60])
> corrplot(newdatacor, method = "number")
```

Command 10: Assumption test for the median of the target variable SHARES.

```
>wilcox.test(online_news$shares, mu=1400)
```

Command 11: Creation of the first model excluding only the high-correlated variables.

```
>reg1 <- lm (shares ~ n_tokens_title + n_unique_tokens + num_hrefs +
num_self_hrefs + num_imgs + num_keywords + data_channel_is_lifestyle +
data_channel_is_entertainment + data_channel_is_bus + data_channel_is_socmed +
data_channel_is_world + kw_min_min + kw_min_max + kw_min_avg +
kw_avg_avg + self_reference_avg_sharess + is_weekend + LDA_00 +
LDA_01 + LDA_02 + LDA_03 + LDA_04 + global_subjectivity +
rate_positive_words + title_sentiment_polarity,data=online_news)
>summary(reg1)
```

Command 12: Second model transforming the target variable with log function.

```
>reg2 <- lm(log(shares) ~ n_tokens_title + n_unique_tokens + num_hrefs +
num_self_hrefs + num_imgs + num_keywords + data_channel_is_lifestyle +
data_channel_is_entertainment + data_channel_is_bus + data_channel_is_socmed +
data_channel_is_world + kw_min_min + kw_min_max + kw_min_avg +
kw_avg_avg + self_reference_avg_sharess + is_weekend + LDA_00 +
LDA_01 + LDA_02 + LDA_03 + LDA_04 + global_subjectivity +
rate_positive_words + title_sentiment_polarity,data=online_news)
>summary(reg2)
```

Command 13: Computing the model using the Lasso “lars” package.

```
> library(lars)
```

```
> X <- model.matrix(shares~.,data=online_news)
> Y <- log(online_news$shares)
>lasso1 <- lars(X,Y)
>plot(lasso1)
>res.cv<-cv.lars(X,Y)
>lamda<-res.cv$index
>cv<-res.cv$cv
>mincv.s<-lamda[cv==min(cv)]
>coef(lasso1,s = mincv.s,mode = 'fraction')
>mincv.s
  [1] 0.3333333
```

(This means that the data have shrinked by 66.6 %)

Command 14: Model with the “glmnet” package.

```
>library(glmnet)
>lasso2<-glmnet(X,Y)
>plot(lasso2,xvar = “lambda”, label = T)
>lasso3<-cv.glmnet(X,Y)
>blasso3<-coef(lasso3,s = "lambda.min")
>blasso3
```

Command 15: The model after the LASSO.

```
>reg3<-lm(log(shares)~num_hrefs +num_keywords + kw_avg_avg + LDA_01 +
LDA_02 + is_weekend +data_channel_is_entertainment + data_channel_is_socmed +
data_channel_is_world,data = online_news)
>summary(reg3)
```

Command 16: Checking the AIC procedure for our model.

```
>install.packages(“MASS”)
>library(MASS)
>step <- stepAIC(reg3, direction="both")
```

Command 17: Computing the Analysis of the Variance table to see our model in comparison to the null and the full models.

```
>online_news_null <- lm(log(shares)~1,data = online_news)
>online_news_full <- lm(log(shares)~.,data = online_news)
> anova(online_news_null, step)
> anova(online_news_full, step)
```

Command 18: Checking the Variance Inflation Factor for the remaining variables.

```
>library(car)
>vif(step)
```

Command 19 : Checking for normality(Comparisons of different residuals using Qqplots).

```
> par(mfcol = c(2,3))
> allres <- list();allres[[1]]<-step$residuals
> allres[[2]]<-rstandard(step)
> allres[[3]]<-rstudent(step)
> mt<-c();mt[1]<-'Unstandarized Residuals'
> mt[2]<-'Standarized Residuals(Internally Standarized)'
> mt[3]<-'Studentized Residuals(Externally Studentized)'
> for (i in 1:3){
        x<-allres[[i]]
        hist(x,probability = T,main = mt[i])
        x0<-seq(min(x),max(x),length.out=100)
        y0<-dnorm(x0,mean(x),sd(x))
        lines(x0,y0,col = 2,lty = 2)
        qqnorm(x,main = mt[i])
        qqline(x)
              }
> normality.pvalues <- matrix(nrow = 3,ncol = 1)
> row.names(normality.pvalues) <- c("Unstandarized","Standarized","Ext. Studentized")
```

```
> colnames(normality.pvalues) <-c("Lillie KS")
> library(nortest)
>  for (i in 1:3){
        res<-allres[[i]]
        normality.pvalues[i,1]<-lillie.test(res)$p.value
            }
> normality.pvalues
```

Command 20: Checking for independence.

```
> library(randtests)
>runs.test(step$residuals)
> library(lmtest)
>dwtest(step)
>library(car)
>durbinWatsonTest(step)
>dwt(step)
>dwt(step$residuals)
```

Command 21: Checking for non-linearity.

```
>residualPlots(step)
```

Command 22: Homoscedasity of errors(Checking the equality of variance in quartiles of fitted values).

```
> p1 <- qplot(step$fitted.values,step$residuals)
> p1 <- p1 + ggtitle("Residual Plot for Online News Data Set")
> p1 <- p1 + theme(plot.title = element_text(lineheight=.8, face="bold"))
> p1 <- p1 + xlab("Fitted Values")
> p1 <- p1 + ylab("Residuals")
> p1
>yhat<-fitted(step)
```

```
> qyhat<-cut ( yhat,breaks = 4)
> bartlett.test(step$residuals~qyhat)
```

Command 23: Checking for leverage points, outliers and influential points.

```
#Leverage points
> leveragePlots(step)

#Outliers
> outlierTest(step) # Bonferonni p-value for most extreme observations
>  qqPlot(step, simulate=T, labels=F)

#Influential points
> library(car)
> cutoff <- 4/((nrow(step)-length(step$coefficients)-2))
> plot(step, which=4, cook.levels=cutoff)
> influencePlot(step,   id.method="identify", main="Influence Plot", sub="Circle size is
proportial to Cook's Distance" )
```

Command 24: 10-fold Cross validation for our model(using the DAAG package).

```
>library(DAAG)

> cv.lm(data = online_news, step, m=10)
```

Command 25: 10-fold Cross validation for the model created with only stepwise regression.

```
>step2<-step(online_news_full, direction = “both”)
> cv.lm(data = online_news, step2, m=10)
```

Command 26: Creation of data partition to train and test for our model for out-of-sample evaluation.

```
 > library(caret)
> set.seed(4352345)
```

```
> online_sampling_vector <- createDataPartition(online_news$shares, p = 0.85, list =
FALSE)
> online_news_train <- online_news[online_sampling_vector,]
> online_news_train_labels <- online_news$shares[online_sampling_vector]
> online_news_test <- online_news[-online_sampling_vector,]
> online_news_test_labels <- online_news$shares[-online_sampling_vector]
```

Command 27: Computation of the MSE between the fitted values of our model and the values of the shares variable in our train set. Apart from that, the computation of the MSE between the predictions of our model and the test set we created.

```
>compute_mse <- function(predictions, actual) { mean((predictions-actual)^2) }

>online_news1_predictions <- predict(step, online_news_test)
>compute_mse(step$fitted.values, log(online_news_train$shares))
>compute_mse(online_news1_predictions, log(online_news_test$shares))
```

Command 28: Importing the test dataset to evaluate our model.

```
> online_news_popularity_test <- read.csv2("OnlineNewsPopularity_test.csv")
 >online_news_popularity_test$url<-NULL
> online_news_popularity_test$timedelta<-NULL
```

Command 29: Computation of the MSE between the predictions of our model and the external test set given to us to evaluate our model.

```
>compute_mse(online_news1_predictions, log(online_news_popularity_test$shares))
```

Tables

Table 1: Description of all the variables using the sjPlot package.

Variable	*vars*	*n*	*missings*	*missings (percentage)*	*mean*	*sd*	*median*	*trimmed*	*mad*	*min*	*max*	*range*	*skew*	*kurtosis*	*se*
X	1	10000	0	0	19651.72	11459.93	19613	19615.26	14644.38	3	39643	39640	0.02	-1.2	114.6
n_tokens_title	2	10000	0	0	10.36	2.11	10	10.33	1.48	3	19	16	0.17	-0.06	0.02
n_tokens_content	3	10000	0	0	544.08	464.21	412	471.37	302.45	0	6336	6336	2.65	13.92	4.64
n_unique_tokens	4	10000	0	0	0.6	7.01	0.54	0.54	0.1	0	701	701	99.91	9985.34	0.07
n_non_stop_words	5	10000	0	0	1.07	10.41	1	1	0	0	1042	1042	99.93	9987.76	0.1
n_non_stop_unique_tokens	6	10000	0	0	0.74	6.5	0.69	0.69	0.1	0	650	650	99.89	9981.79	0.06
num_hrefs	7	10000	0	0	11.1	12.02	8	8.97	5.93	0	187	187	4.02	29.54	0.12
num_self_hrefs	8	10000	0	0	3.31	4.05	2	2.69	1.48	0	116	116	6.46	94.53	0.04
num_imgs	9	10000	0	0	4.55	8.21	1	2.68	1.48	0	128	128	3.76	22.47	0.08
num_videos	10	10000	0	0	1.27	4.21	0	0.37	0	0	75	75	7.01	71.28	0.04
average_token_length	11	10000	0	0	4.55	0.84	4.66	4.66	0.28	0	6.51	6.51	-4.62	22.6	0.01
num_keywords	12	10000	0	0	7.24	1.9	7	7.3	1.48	1	10	9	-0.14	-0.83	0.02
data_channel_is_lifestyle	13	10000	0	0	0.05	0.22	0	0	0	0	1	1	4.09	14.74	0
data_channel_is_entertainment	14	10000	0	0	0.18	0.38	0	0.1	0	0	1	1	1.66	0.77	0
data_channel_is_bus	15	10000	0	0	0.16	0.36	0	0.07	0	0	1	1	1.88	1.52	0
data_channel_is_socmed	16	10000	0	0	0.06	0.23	0	0	0	0	1	1	3.77	12.18	0
data_channel_is_tech	17	10000	0	0	0.19	0.39	0	0.11	0	0	1	1	1.61	0.59	0
data_channel_is_world	18	10000	0	0	0.21	0.41	0	0.14	0	0	1	1	1.42	0	0
kw_min_min	19	10000	0	0	26.51	70.09	-1	6.12	0	-1	294	295	2.35	3.53	0.7
kw_max_min	20	10000	0	0	1148.26	3377.24	657	738.63	373.62	0	128500	128500	19.13	510.09	33.77
kw_avg_min	21	10000	0	0	311.21	540.01	237	250.02	154.44	-1	21516	21517	18.88	549.98	5.4
kw_min_max	22	10000	0	0	13602.34	57775.68	1450	4101.33	2149.77	0	843300	843300	10.34	121.93	577.76
kw_max_max	23	10000	0	0	750128.84	217795.47	843300	812861.32	0	0	843300	843300	-2.6	5.43	2177.95
kw_avg_max	24	10000	0	0	258383.13	136299.4	241781.1	251771.44	118115.57	0	843300	843300	0.63	0.78	1362.99
kw_min_avg	25	10000	0	0	1119.01	1138.68	1044.12	1000.12	1548.01	-1	3609.72	3610.72	0.46	-1.14	11.39
kw_max_avg	26	10000	0	0	5614.19	5740.95	4354.57	4782.24	1376.3	0	237966.67	237966.67	14.4	373.76	57.41
kw_avg_avg	27	10000	0	0	3132.66	1306.59	2867.96	2983.43	845.72	0	36717.23	36717.23	5.47	90.1	13.07
self_reference_min_shares	28	10000	0	0	4158.21	22424.55	1200	1651.03	1334.34	0	690400	690400	24.6	713.03	224.25
self_reference_max_shares	29	10000	0	0	10026.26	39604.68	2800	4424.96	4151.28	0	837700	837700	13.99	227.28	396.05
self_reference_avg_sharess	30	10000	0	0	6410.71	26133.88	2200	3018.35	2668.68	0	690400	690400	18.29	418.65	261.34
weekday_is_monday	31	10000	0	0	0.17	0.37	0	0.09	0	0	1	1	1.76	1.11	0
weekday_is_tuesday	32	10000	0	0	0.18	0.39	0	0.1	0	0	1	1	1.65	0.72	0
weekday_is_wednesday	33	10000	0	0	0.19	0.39	0	0.11	0	0	1	1	1.61	0.59	0
weekday_is_thursday	34	10000	0	0	0.19	0.39	0	0.11	0	0	1	1	1.59	0.53	0
weekday_is_friday	35	10000	0	0	0.14	0.35	0	0.05	0	0	1	1	2.04	2.15	0
weekday_is_saturday	36	10000	0	0	0.06	0.24	0	0	0	0	1	1	3.65	11.35	0
weekday_is_sunday	37	10000	0	0	0.07	0.25	0	0	0	0	1	1	3.39	9.52	0
is_weekend	38	10000	0	0	0.13	0.34	0	0.04	0	0	1	1	2.19	2.81	0
LDA_00	39	10000	0	0	0.18	0.26	0.03	0.12	0.02	0	0.93	0.93	1.59	1.15	0
LDA_01	40	10000	0	0	0.14	0.22	0.03	0.09	0.02	0	0.92	0.92	2.07	3.25	0
LDA_02	41	10000	0	0	0.22	0.28	0.04	0.16	0.03	0	0.92	0.92	1.33	0.3	0
LDA_03	42	10000	0	0	0.22	0.3	0.04	0.17	0.03	0	0.92	0.92	1.23	-0.05	0
LDA_04	43	10000	0	0	0.24	0.29	0.04	0.18	0.03	0	0.93	0.93	1.16	-0.13	0
global_subjectivity	44	10000	0	0	0.44	0.12	0.45	0.45	0.08	0	1	1	-1.36	4.71	0
global_sentiment_polarity	45	10000	0	0	0.12	0.1	0.12	0.12	0.09	-0.38	0.62	1	0.09	1.58	0
global_rate_positive_words	46	10000	0	0	0.04	0.02	0.04	0.04	0.02	0	0.16	0.16	0.33	1.04	0
global_rate_negative_words	47	10000	0	0	0.02	0.01	0.02	0.02	0.01	0	0.18	0.18	2.03	13.96	0
rate_positive_words	48	10000	0	0	0.68	0.19	0.71	0.7	0.15	0	1	1	-1.41	3.23	0
rate_negative_words	49	10000	0	0	0.29	0.16	0.28	0.28	0.15	0	1	1	0.42	0.55	0
avg_positive_polarity	50	10000	0	0	0.35	0.1	0.36	0.36	0.08	0	0.95	0.95	-0.74	3.37	0
min_positive_polarity	51	10000	0	0	0.1	0.07	0.1	0.09	0.06	0	0.9	0.9	2.77	13.41	0
max_positive_polarity	52	10000	0	0	0.76	0.25	0.8	0.79	0.3	0	1	1	-0.94	0.65	0
avg_negative_polarity	53	10000	0	0	-0.26	0.13	-0.25	-0.26	0.11	-1	0	1	-0.52	2.19	0
min_negative_polarity	54	10000	0	0	-0.52	0.29	-0.5	-0.52	0.3	-1	0	1	-0.09	-0.82	0
max_negative_polarity	55	10000	0	0	-0.11	0.09	-0.1	-0.09	0.07	-1	0	1	-3.42	18.93	0
title_subjectivity	56	10000	0	0	0.28	0.32	0.15	0.24	0.22	0	1	1	0.8	-0.56	0
title_sentiment_polarity	57	10000	0	0	0.07	0.26	0	0.06	0	-1	1	2	0.32	3.22	0
abs_title_subjectivity	58	10000	0	0	0.34	0.19	0.5	0.36	0	0	0.5	0.5	-0.6	-1.32	0

abs_title_sentiment_polarity	59	10000	0	0	0.16	0.22	0	0.11	0	0	1	1	1.68	2.63	0
shares	60	10000	0	0	3353.85	11760.1	1400	1857.44	889.56	5	663600	663595	29.2	1293.95	117.6
weekday*	61	10000	0	0	4.28	2.13	5	4.35	2.97	1	7	6	-0.26	-1.39	0.02
data_channel*	62	10000	0	0	3.21	1.78	3	3.14	2.97	1	6	5	0.26	-1.35	0.02

Plot 1: The qqplot of the shares variable without the transformation using the log.

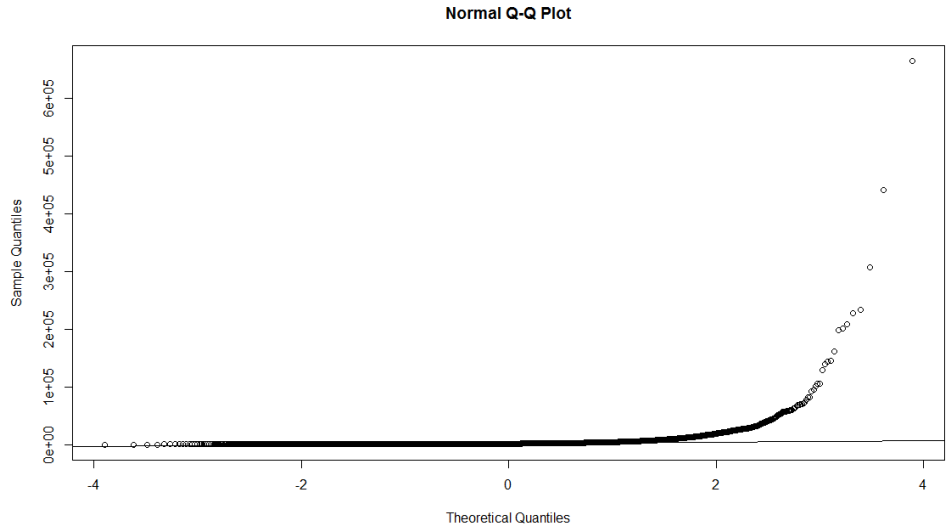


Table 2: Accumulative table with the frequency of the SHARES variable.

Shares				
value	*N*	*raw %*	*valid %*	*cumulative %*
5-66399	9972	99.72	99.72	99.72
66400-132999	16	0.16	0.16	99.88
133000-198999	5	0.05	0.05	99.93
199000-264999	4	0.04	0.04	99.97
265000-331999	1	0.01	0.01	99.98
398000-464999	1	0.01	0.01	99.99
597000-663999	1	0.01	0.01	100.00
missings	0	0.00		
total N=10000 · valid N=10000 · $\bar{x}$=3353.85 · σ=11760.10				

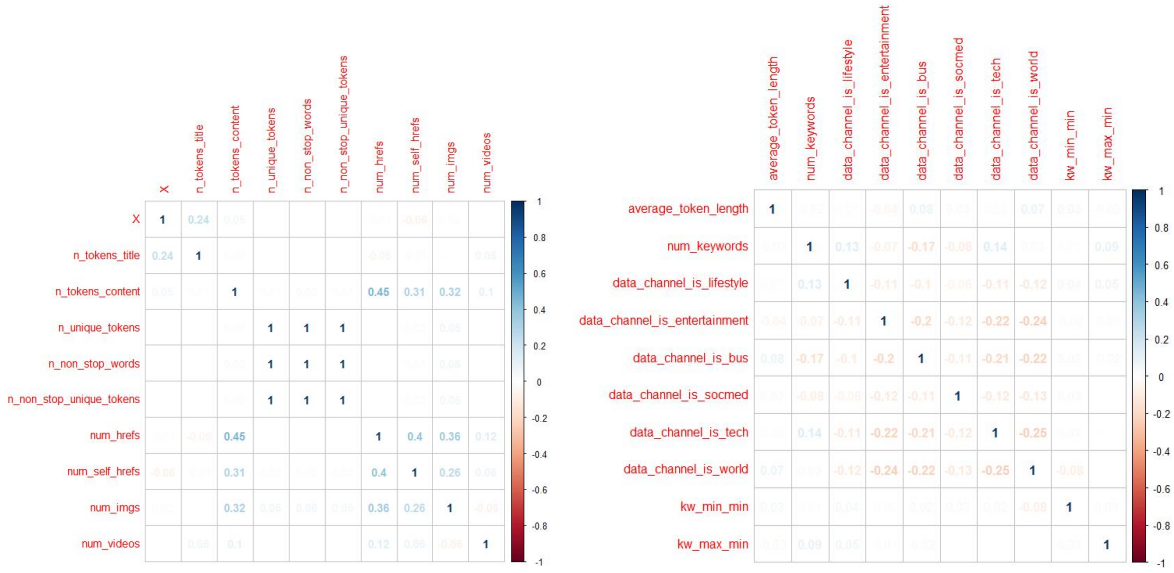
It is clear that almost all of the observations are below 66399 shares.

Table 3: Correlations between variables.

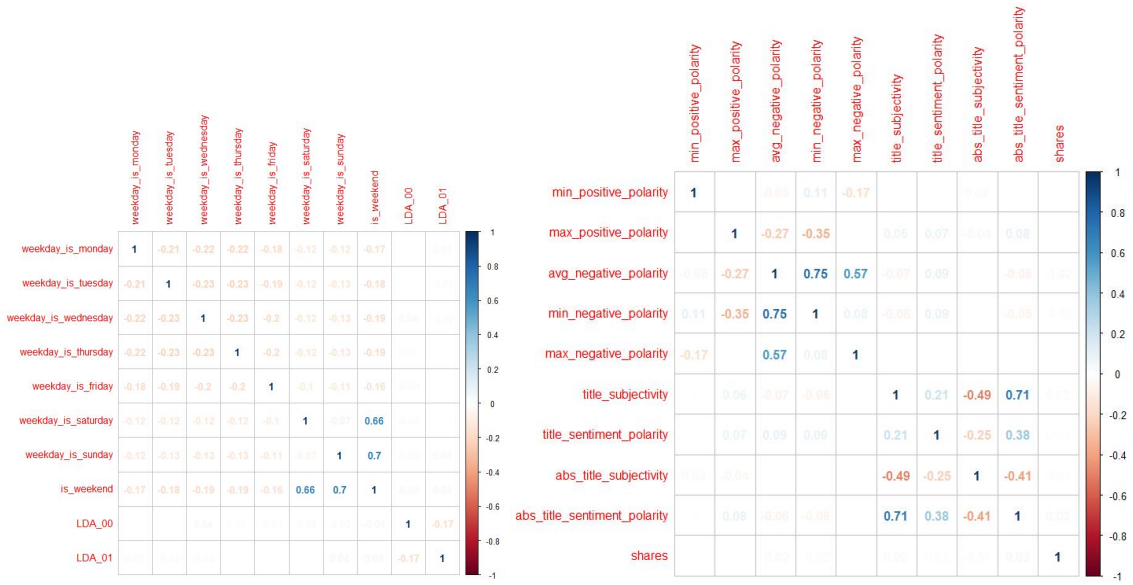
Variable 1	Correlation	Variable 2
n_tokens_content	*0.74*	*num_refs*
n_unique_tokens	*0.99*	*n_non_stop_words*
n_unique_tokens	*0.99*	*n_non_stop_unique_tokens*
average_token_length	*0.85*	*global_subjectivity*
kw_min_min	*0.99*	*kw_max_min*
kw_min_min	*-0.99*	*kw_max_max*
kw_max_min	*0.99*	*kw_avg_min*
kw_max_avg	*0.88*	*kw_avg_avg*
self_reference_min_shares	*0.97*	*self_reference_avg_shares*
self_reference_max_shares	*0.97*	*self_reference_avg_shares*
LDA_00	*0.94*	*data_channel_is_business*
global_subjectivity	*0.92*	*avg_positive_polarity*
global_sentiment_polarity	*0.91*	*rate_positive_words*
global_rate_positive_words	*0.85*	*rate_positive_words*
global_rate_negative_words	*0.91*	*rate_negative_words*
max_positive_polarity	*0.89*	*avg_positive_polarity*
avg_negative_polarity	*0.89*	*min_negative_polarity*
title_subjectivity	*0.94*	*abs_title_sentiment_polarity*

The highlighted variables are selected for the rest procedure. (The first pair although having a high correlation does not have a meaningful relation so we cannot exclude one of them).

Plot 2: Below are the correlation plots (in groups of 10):



We notice that in the first plot there is a high correlation between the variables n_unique_tokens, n_non_stop_words and n_non_stop_unique_tokens.



Here we notice the high correlation between the variables is_weekend, weekday_is_Saturday and weekday_is_Sunday, which is expected.

Summary 1: of the linear regression without the log (with only the exclusion of the variables from the correlation table.

```
Call:
lm(formula = shares ~ n_tokens_title + n_tokens_content + n_unique_tokens +
    num_hrefs + num_self_hrefs + num_imgs + num_videos + num_keywords +
    data_channel_is_lifestyle + data_channel_is_entertainment +
    data_channel_is_bus + data_channel_is_socmed + data_channel_is_tech +
    data_channel_is_world + kw_min_min + kw_min_max + kw_avg_max +
    kw_min_avg + kw_avg_avg + self_reference_avg_sharess + is_weekend +
    LDA_00 + LDA_01 + LDA_02 + LDA_03 + LDA_04 + global_subjectivity +
    rate_positive_words + rate_negative_words + min_positive_polarity +
    max_positive_polarity + avg_negative_polarity + max_negative_polarity +
    title_subjectivity + title_sentiment_polarity + abs_title_subjectivity +
    title_subjectivity, data = online_news)

Residuals:
   Min     1Q Median     3Q    Max
-28625  -2238  -1179    -57 657501

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -1.092e+06  1.311e+06  -0.833  0.40487
n_tokens_title                1.107e+02  5.751e+01   1.925  0.05432 .
n_tokens_content              8.713e-01  4.291e-01   2.030  0.04233 *
n_unique_tokens               1.561e+03  1.869e+03   0.835  0.40364
num_hrefs                     2.933e+01  1.225e+01   2.395  0.01666 *
num_self_hrefs               -7.700e+01  3.328e+01  -2.314  0.02069 *
num_imgs                      1.448e+01  1.713e+01   0.846  0.39784
num_videos                   -2.219e+01  3.025e+01  -0.733  0.46334
num_keywords                  9.182e+01  7.356e+01   1.248  0.21200
data_channel_is_lifestyle    -1.366e+03  7.935e+02  -1.721  0.08522 .
data_channel_is_entertainment -1.591e+03  4.915e+02  -3.237  0.00121 **
data_channel_is_bus          -1.252e+03  7.483e+02  -1.674  0.09419 .
data_channel_is_socmed       -9.034e+02  7.327e+02  -1.233  0.21759
data_channel_is_tech         -6.168e+02  7.316e+02  -0.843  0.39923
data_channel_is_world        -1.436e+03  7.399e+02  -1.941  0.05227 .
kw_min_min                    2.470e+00  2.162e+00   1.142  0.25328
kw_min_max                   -1.127e-03  2.338e-03  -0.482  0.62966
kw_avg_max                   -9.344e-04  1.456e-03  -0.642  0.52104
kw_min_avg                   -2.191e-01  1.275e-01  -1.719  0.08569 .
kw_avg_avg                    9.384e-01  1.206e-01   7.782 7.87e-15 ***
self_reference_avg_sharess    3.892e-03  4.537e-03   0.858  0.39108
is_weekend                    2.894e+02  3.500e+02   0.827  0.40829
LDA_00                        1.093e+06  1.311e+06   0.834  0.40426
LDA_01                        1.092e+06  1.311e+06   0.833  0.40480
LDA_02                        1.092e+06  1.311e+06   0.833  0.40485
LDA_03                        1.093e+06  1.311e+06   0.834  0.40413
LDA_04                        1.093e+06  1.311e+06   0.834  0.40446
global_subjectivity           6.191e+01  1.608e+03   0.039  0.96928
rate_positive_words          -2.842e+03  1.680e+03  -1.691  0.09079 .
rate_negative_words          -1.441e+03  1.677e+03  -0.859  0.39036
min_positive_polarity        -1.711e+03  1.910e+03  -0.896  0.37032
max_positive_polarity        -1.198e+01  6.608e+02  -0.018  0.98554
avg_negative_polarity        -1.028e+03  1.488e+03  -0.691  0.48968
max_negative_polarity        -7.398e+02  1.785e+03  -0.414  0.67856
title_subjectivity           -1.827e+02  4.297e+02  -0.425  0.67076
title_sentiment_polarity      3.556e+02  4.749e+02   0.749  0.45406
abs_title_subjectivity       -2.741e+02  7.281e+02  -0.376  0.70664
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11640 on 9963 degrees of freedom
```

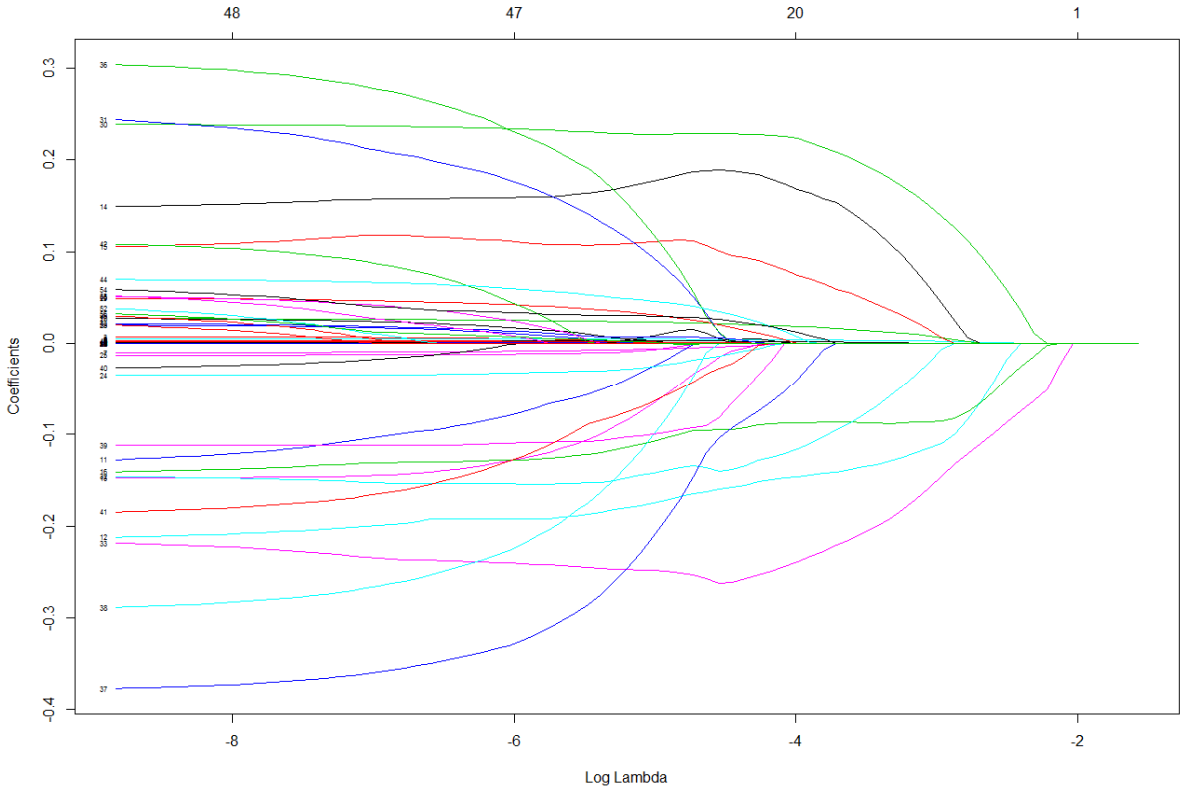
```
Multiple R-squared:  0.02409, Adjusted R-squared:  0.02056
F-statistic:  6.83 on 36 and 9963 DF,  p-value: < 2.2e-16
```

Summary 2: After the transformation of the SHARES variable using the log.

```
Call:
lm(formula = log(shares) ~ n_tokens_title + n_tokens_content +
n_unique_tokens + num_hrefs + num_self_hrefs + num_imgs +
num_videos + num_keywords + data_channel_is_lifestyle +
data_channel_is_entertainment +
data_channel_is_bus + data_channel_is_socmed + data_channel_is_tech +
data_channel_is_world + kw_min_min + kw_min_max + kw_avg_max +
kw_min_avg + kw_avg_avg + self_reference_avg_sharess + is_weekend +
LDA_00 + LDA_01 + LDA_02 + LDA_03 + LDA_04 + global_subjectivity +
rate_positive_words + rate_negative_words + min_positive_polarity +
max_positive_polarity + avg_negative_polarity + max_negative_polarity +
title_subjectivity + title_sentiment_polarity + abs_title_subjectivity +
title_subjectivity, data = online_news)

Residuals:
Min      1Q  Median      3Q     Max
-5.7050 -0.5436 -0.1697  0.3967  5.5085

Coefficients:
                               Estimate  Std. Error  t value Pr(>|t|)
(Intercept)                   1.504e+02  9.855e+01   1.526 0.126991
n_tokens_title                6.241e-03  4.324e-03   1.443 0.148930
n_tokens_content              1.026e-05  3.227e-05   0.318 0.750546
n_unique_tokens              -2.030e-01  1.405e-01  -1.444 0.148682
num_hrefs                     4.987e-03  9.211e-04   5.414 6.30e-08 ***
num_self_hrefs               -9.944e-03  2.502e-03  -3.974 7.11e-05 ***
num_imgs                      3.016e-03  1.288e-03   2.342 0.019186 *
num_videos                    2.718e-03  2.275e-03   1.195 0.232070
num_keywords                  2.668e-02  5.531e-03   4.824 1.43e-06 ***
data_channel_is_lifestyle    -1.212e-01  5.966e-02  -2.031 0.042327 *
data_channel_is_entertainment -2.187e-01  3.695e-02  -5.918 3.37e-09 ***
data_channel_is_bus          -1.908e-01  5.626e-02  -3.391 0.000699 ***
data_channel_is_socmed        1.626e-01  5.509e-02   2.952 0.003162 **
data_channel_is_tech          8.293e-02  5.501e-02   1.508 0.131688
data_channel_is_world        -1.616e-01  5.564e-02  -2.904 0.003694 **
kw_min_min                    6.386e-04  1.626e-04   3.928 8.61e-05 ***
kw_min_max                   -3.304e-07  1.758e-07  -1.879 0.060228 .
kw_avg_max                    4.891e-08  1.095e-07   0.447 0.655052
kw_min_avg                    1.838e-05  9.586e-06   1.918 0.055168 .
kw_avg_avg                    1.133e-04  9.068e-06  12.493  < 2e-16 ***
self_reference_avg_sharess    1.357e-06  3.412e-07   3.978 7.00e-05 ***
is_weekend                    2.681e-01  2.632e-02  10.188  < 2e-16 ***
LDA_00                       -1.432e+02  9.855e+01  -1.453 0.146218
LDA_01                       -1.436e+02  9.855e+01  -1.457 0.145112
LDA_02                       -1.437e+02  9.855e+01  -1.458 0.144923
LDA_03                       -1.435e+02  9.855e+01  -1.456 0.145521
LDA_04                       -1.434e+02  9.855e+01  -1.455 0.145615
global_subjectivity           3.185e-01  1.209e-01   2.635 0.008436 **
rate_positive_words          -2.405e-01  1.263e-01  -1.904 0.056965 .
rate_negative_words          -1.569e-01  1.261e-01  -1.244 0.213590
min_positive_polarity        -8.200e-02  1.436e-01  -0.571 0.568053
max_positive_polarity        -3.425e-02  4.969e-02  -0.689 0.490691
avg_negative_polarity        -1.745e-01  1.119e-01  -1.560 0.118791
max_negative_polarity         9.189e-02  1.342e-01   0.685 0.493579
title_subjectivity            2.039e-02  3.231e-02   0.631 0.527935
```

```
title_sentiment_polarity       6.843e-02  3.571e-02   1.916 0.055347 .
abs_title_subjectivity         4.720e-02  5.475e-02   0.862 0.388637
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8751 on 9963 degrees of freedom
Multiple R-squared:  0.1142,   Adjusted R-squared:  0.111
F-statistic: 35.68 on 36 and 9963 DF,  p-value: < 2.2e-16
```

Plot 3: Lasso Plot (using the glmnet package).



Summary 4: The summary of the coefficients in blasso3 in the LASSO procedure.

```
57 x 1 sparse Matrix of class "dgCMatrix"
1
(Intercept)                    7.096980748
X                              .
n_tokens_title                 .
n_tokens_content               .
n_unique_tokens                .
num_hrefs                      0.002642196
num_self_hrefs                 .
num_imgs                       .
num_videos                     .
num_keywords                   0.008085691
data_channel_is_lifestyle      .
```

```
data_channel_is_entertainment -0.114798076
data_channel_is_bus            .
data_channel_is_socmed         0.062315740
data_channel_is_tech           .
data_channel_is_world         -0.086631552
kw_min_min                     .
kw_min_max                     .
kw_avg_max                     .
kw_min_avg                     .
kw_avg_avg                     0.000103543
self_reference_avg_sharess     .
weekday_is_monday              .
weekday_is_tuesday             .
weekday_is_wednesday           .
weekday_is_thursday            .
weekday_is_friday              .
weekday_is_saturday            .
weekday_is_sunday              .
is_weekend                     0.150464916
LDA_00                         .
LDA_01                        -0.021103436
LDA_02                        -0.158338617
LDA_03                         .
LDA_04                         .
global_subjectivity            .
rate_positive_words            .
rate_negative_words            .
min_positive_polarity          .
max_positive_polarity          .
avg_negative_polarity          .
max_negative_polarity          .
title_subjectivity             .
title_sentiment_polarity       .
abs_title_subjectivity         .
```

Summary 5: The summary after LASSO.

```
Call:
lm(formula = log(shares) ~ num_hrefs + num_keywords + kw_avg_avg +
LDA_01 + LDA_02 + is_weekend + data_channel_is_entertainment +
data_channel_is_socmed + data_channel_is_world,
data = online_news)

Residuals:
Min      1Q  Median      3Q     Max
-5.7176 -0.5499 -0.1756  0.3989  5.7412

Coefficients:
                              Estimate  Std. Error t value  Pr(>|t|)
(Intercept)                   6.916e+00  4.456e-02 155.217  < 2e-16 ***
num_hrefs                     5.080e-03  7.499e-04   6.774 1.32e-11 ***
num_keywords                  2.900e-02  4.716e-03   6.150 8.05e-10 ***
kw_avg_avg                    1.223e-04  7.065e-06  17.318  < 2e-16 ***
LDA_01                       -1.875e-01  5.041e-02  -3.720 0.000200 ***
LDA_02                       -2.831e-01  5.893e-02  -4.804 1.58e-06 ***
is_weekend                    2.740e-01  2.632e-02  10.409  < 2e-16 ***
data_channel_is_entertainment -2.015e-01  2.909e-02  -6.927 4.58e-12 ***
data_channel_is_socmed         2.178e-01  3.915e-02   5.564 2.71e-08 ***
data_channel_is_world         -1.461e-01  4.111e-02  -3.553 0.000383 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8805 on 9990 degrees of freedom
Multiple R-squared:  0.1009,   Adjusted R-squared:    0.1
F-statistic: 124.5 on 9 and 9990 DF,  p-value: < 2.2e-16
```

Summary 6: The summary after AIC.

```
Call:
lm(formula = log(shares) ~ num_hrefs + num_keywords + kw_avg_avg +
LDA_01 + LDA_02 + is_weekend + data_channel_is_entertainment +
data_channel_is_socmed + data_channel_is_world ,
data = online_news)

Residuals:
Min      1Q  Median      3Q     Max
-5.7176 -0.5499 -0.1756  0.3989  5.7412

Coefficients:
                               Estimate  Std. Error t value  Pr(>|t|)
(Intercept)                   6.916e+00  4.456e-02 155.217  < 2e-16 ***
num_hrefs                     5.080e-03  7.499e-04   6.774 1.32e-11 ***
num_keywords                  2.900e-02  4.716e-03   6.150 8.05e-10 ***
kw_avg_avg                    1.223e-04  7.065e-06  17.318  < 2e-16 ***
LDA_01                       -1.875e-01  5.041e-02  -3.720 0.000200 ***
LDA_02                       -2.831e-01  5.893e-02  -4.804 1.58e-06 ***
is_weekend                    2.740e-01  2.632e-02  10.409  < 2e-16 ***
data_channel_is_entertainment -2.015e-01  2.909e-02  -6.927 4.58e-12 ***
data_channel_is_socmed         2.178e-01  3.915e-02   5.564 2.71e-08 ***
data_channel_is_world         -1.461e-01  4.111e-02  -3.553 0.000383 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8805 on 9990 degrees of freedom
Multiple R-squared:  0.1009,   Adjusted R-squared:    0.1
F-statistic: 124.5 on 9 and 9990 DF,  p-value: < 2.2e-16
```

Summary 7: Analysis of the Variance table.

```
Analysis of Variance Table

Model 1: log(shares) ~ 1
Model 2: log(shares) ~ num_hrefs + num_keywords + kw_avg_avg + LDA_01 +
LDA_02 + is_weekend + data_channel_is_entertainment + data_channel_is_socmed
+
data_channel_is_world
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1   9999 8613.8
2   9990 7745.0  9    868.74 124.5 < 2.2e-16 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Analysis of Variance Table

Model 1: log(shares) ~ X + n_tokens_title + n_tokens_content +
n_unique_tokens +
    n_non_stop_words + n_non_stop_unique_tokens + num_hrefs +
    num_self_hrefs + num_imgs + num_videos + average_token_length +
    num_keywords + data_channel_is_lifestyle + data_channel_is_entertainment
+
    data_channel_is_bus + data_channel_is_socmed + data_channel_is_tech +
    data_channel_is_world + kw_min_min + kw_max_min + kw_avg_min +
    kw_min_max + kw_max_max + kw_avg_max + kw_min_avg + kw_max_avg +
    kw_avg_avg + self_reference_min_shares + self_reference_max_shares +
    self_reference_avg_sharess + weekday_is_monday + weekday_is_tuesday +
    weekday_is_wednesday + weekday_is_thursday + weekday_is_friday +
    weekday_is_saturday + weekday_is_sunday + is_weekend + 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
Model 2: log(shares) ~ num_hrefs + num_keywords + kw_avg_avg + LDA_01 +
    LDA_02 + is_weekend + data_channel_is_entertainment +
data_channel_is_socmed +
    data_channel_is_world + data_channel_is_world
  Res.Df  RSS  Df Sum of Sq    F Pr(>F)
1   9942 7529
2   9990 7745 -48      -216 5.95 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

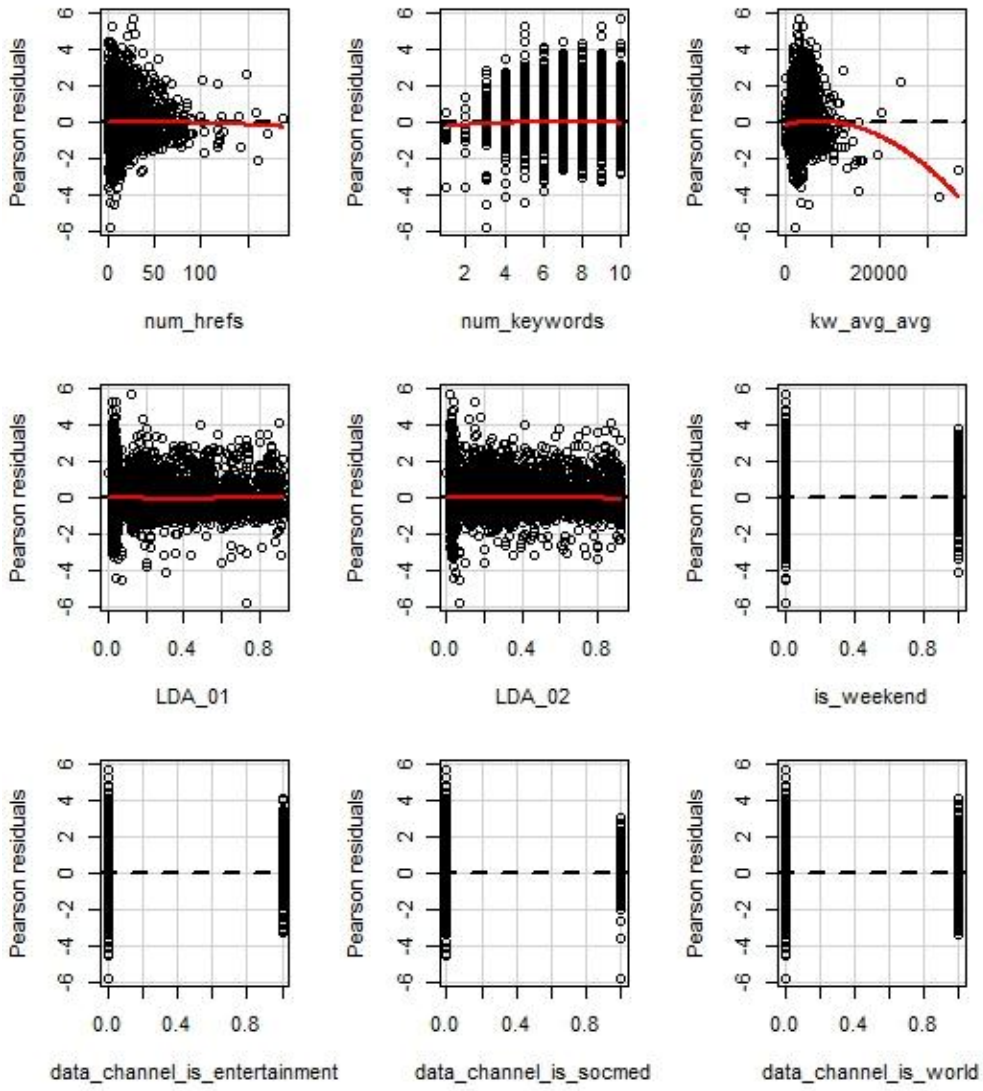
Summary 8 : Variance Inflation Factor for each variable (all below 4,so we can proceed).

num_hrefs	1.048309
num_keywords	1.038781
kw_avg_avg	1.098852
LDA_01	1.581267
LDA_02	3.592928
is_weekend	1.014564
data_channel_is_entertainment	1.612152
data_channel_is_socmed	1.086911
data_channel_is_world	3.630658

Plot 3: Normality tests for the residuals.

```
       normality.pvalues
                     Lillie KS
Unstandarized    4.850351e-264
Standarized      6.336815e-264
Ext. Studentized 4.453763e-264
```

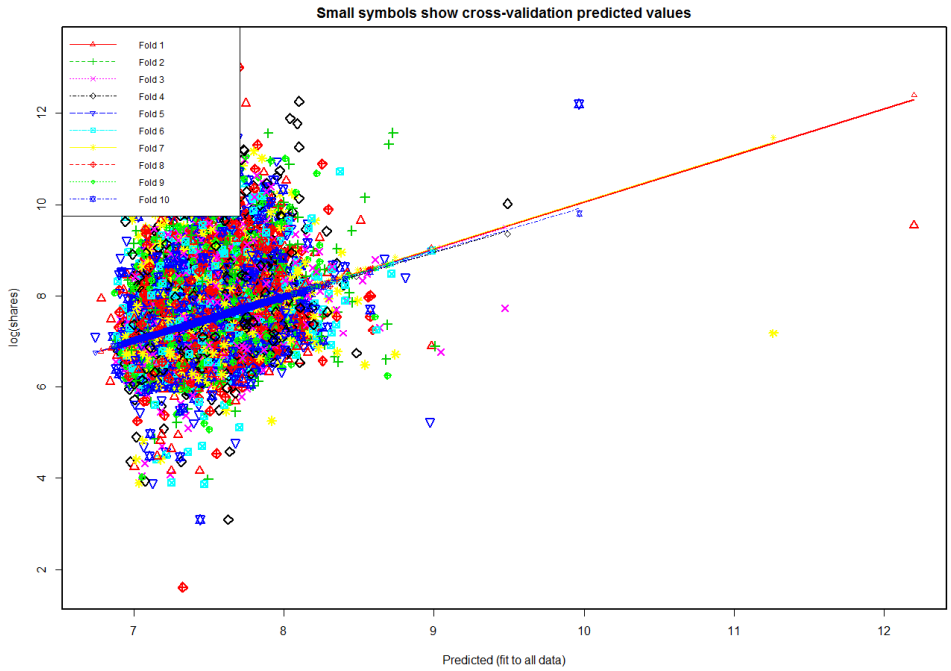
Plot 4: Residual Plots for each variable.



Summary 10: Randomness hypothesis for the residuals.

```
Runs Test

data:  step$residuals
statistic = 0.72004, runs = 5037, n1 = 5000, n2 = 5000, n = 10000, p-value = 0.4715
alternative hypothesis: nonrandomness
```

Plot 5: Cross Validation predicted values for our model (The function produces a plot (below) of each fold's predicted values against the actual outcome variable).



Plot 6: Cross Validation predicted values in the model created only by stepwise regression.