Exploratory Data Analysis

Project Report
winter 2018
Instructor: Wayne Oldford

Presented by:
Rongrong Su
r2sue@uwaterloo.ca
20751802

Table of Contents

Online news popularity analysis

Introduction

With Internet expansion, the prediction of online news popularity is becoming a trendy research topic. In this project, I intend to analysis if an article will become popular or not by using features like type of content, number of images/videos , the day on which the paper was published, etc. The data set from UCI website summarizes a heterogeneous set of features about articles published by Mashable in a period of two years [1]. Dataset consists of 61 attributes: 58 predictive attributes, 2 non-predictive, 1 goal field.

Structure of data

Dataset

They extracted an extensive set (total of 47) features from the HTML code in order to turn this data suitable for learning models, as shown in Table 1. In the table, the attribute types were classified into: number – integer value; ratio – within [0, 1]; bool – $\in \{0, 1\}$; and nominal. Column Type shows within brackets (#) the number of variables related with the attribute. Similarly, to what is executed in [5,6], they performed a logarithmic transformation to scale the unbounded numeric features (e.g., number of words in article), while the nominal attributes were transformed with the common 1-of-C encoding.

Feature	Type (#)
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article links	number (1)
Minimum, average and maximum number of shares of Mashable links	number (3)
Digital Media	
Number of images	number (1)
Number of videos	number (1)
Time	
Day of the week	nominal (1)
Published on a weekend?	bool (1)

Feature	Type (#)
Keywords	
Number of keywords	number (1)
Worst keyword (min./avg./max. shares)	number (3)
Average keyword (min./avg./max. shares)	number (3)
Best keyword (min./avg./max. shares)	number (3)
Article category (Mashable data channel)	nominal (1)
Natural Language Processing	
Closeness to top 5 LDA topics	ratio (5)
Title subjectivity	ratio (1)
Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Title sentiment polarity	ratio (1)
Rate of positive and negative words	ratio (2)
Pos. words rate among non-neutral words	ratio (1)
Neg. words rate among non-neutral words	ratio (1)
Polarity of positive words (min./avg./max.)	ratio (3)
Polarity of negative words (min./avg./max.)	ratio (3)
Article text polarity score and its absolute difference to 0.5	ratio (2)

Target	Type (#)
Number of article Mashable shares	number (1)

Attributes

0. url: URL of the article (non-predictive)
1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)
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)
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
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)

Notes:
1. Stop Words usually refer to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on.

2. Min, Max, Avg related variables
Some of the features are dependent of particularities of the Mashable service: articles often reference other articles published in the same service; and articles have meta-data, such as keywords, data channel type and total number of shares (when considering Facebook, Twitter, Google+, LinkedIn, Stumble- Upon and Pinterest). Thus, we extracted the minimum, average and maximum number of shares (known before publication) of all Mashable links cited in the article. Similarly, we rank all article keyword average shares (known before publication), in order to get the worst, average and best keywords. For each of these keywords, we extract the minimum, average and maximum number of shares [2].

3. LDA
They also extracted several natural language processing features [2]. The Latent Dirichlet Allocation (LDA) [3] algorithm was applied to all Mashable texts (known before publication) in order to first identify the five top relevant topics and then measure the closeness of current article to such topics. To compute the subjectivity and polarity sentiment analysis, we adopted the Pattern web mining module (http://www.clips.ua.ac.be/pattern) [4], allowing the computation of sentiment polarity and subjectivity scores.

4. subjective
The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

5. polarity
Polarity, also known as orientation is he emotion expressed in the sentence. For example, the

English phrase “not a very great calculation” has a polarity of about -0.3, meaning it is slightly negative.

Recoding some variables

news_channel

I create a new categorical variable called news_channel valued with Lifestyle, Entertainment, Business, Social Media, Tech and World. These values will be derived from the following data channel indicator variables in the dataset:

data_channel_is_lifestyle
data_channel_is_entertainment
data_channel_is_bus
data_channel_is_socmed
data_channel_is_tech
data_channel_is_world

```
news_df$news_channel<-NA
news_df$news_channel[news_df$data_channel_is_lifestyle==1] <- "Lifestyle"
news_df$news_channel[news_df$data_channel_is_entertainment==1] <- "Entertainm
ent"
news_df$news_channel[news_df$data_channel_is_bus==1] <- "Business"
news_df$news_channel[news_df$data_channel_is_socmed==1] <- "Social Media"
news_df$news_channel[news_df$data_channel_is_tech==1] <- "Technology"
news_df$news_channel[news_df$data_channel_is_world==1] <- "World"
# Create the News Channel variable
news_df$news_channel <-  factor(news_df$news_channel,
                                       levels = c("Business",
                                                  "Entertainment",
                                                  "Lifestyle",
                                                  "Technology",
                                                  "World",
                                                  "Social Media"))
```

published_day

I create a new categorical variable called published_day will be created to indicate the day of the week the news article was published. The day of the week value will be derived from the following weekday indicator variables in the dataset:

weekday_is_monday
weekday_is_tuesday
weekday_is_wednesday
weekday_is_thursday

weekday_is_friday
weekday_is_saturday
weekday_is_sunday

```
news_df$published_day <- NA
news_df$published_day [news_df$weekday_is_monday==1] <- "Monday"
news_df$published_day [news_df$weekday_is_tuesday==1] <- "Tuesday"
news_df$published_day [news_df$weekday_is_wednesday==1] <- "Wednesday"
news_df$published_day [news_df$weekday_is_thursday==1] <- "Thursday"
news_df$published_day [news_df$weekday_is_friday==1] <- "Friday"
news_df$published_day [news_df$weekday_is_saturday==1] <- "Saturday"
news_df$published_day [news_df$weekday_is_sunday==1] <- "Sunday"

news_df$published_day <- factor(news_df$published_day,
                                levels = c( "Monday",
                                            "Tuesday",
                                            "Wednesday",
                                            "Thursday",
                                            "Friday",
                                            "Saturday",
                                            "Sunday"))
```

date and year of publication

I extract date, month and year of publication from URL.

```
news_df$published_date <- ymd(substr(news_df$url, 21, 30))
news_df$published_month<-as.factor(month(news_df$published_date))
news_df$published_year <- as.factor(substr(news_df$url, 21, 24))
```

removing ‘URL’

The useless variables like ‘URL’ are removed.

```
removevars <- c("url",
                "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",
                "timedelta")
```

So, the data set after transformed looks as follows.

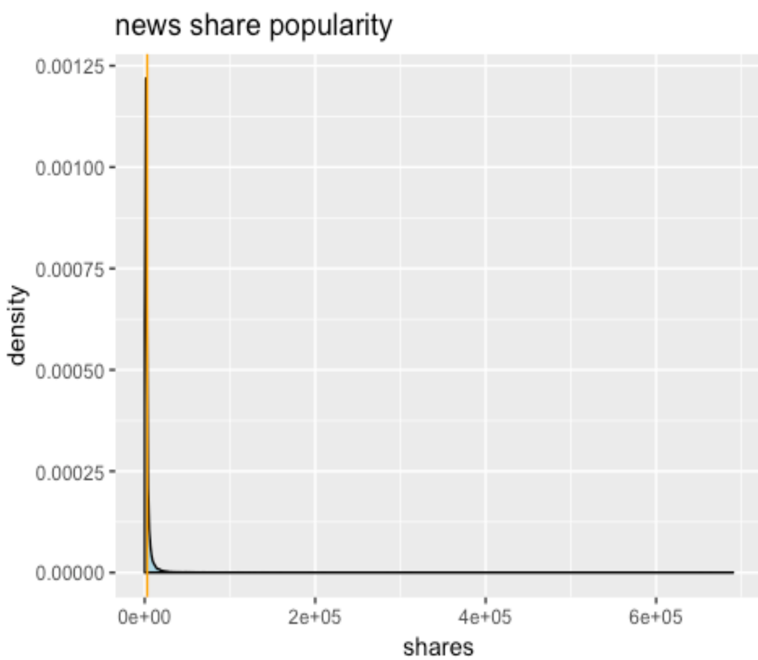
```
str(news_df)

## 'data.frame':    33510 obs. of  51 variables:
##  $ n_tokens_title              : num  12 9 9 9 13 10 8 12 11 10 ...
##  $ n_tokens_content            : num  219 255 211 531 1072 ...
##  $ n_unique_tokens             : num  0.664 0.605 0.575 0.504 0.416 ...
##  $ n_non_stop_words            : num  1 1 1 1 1 ...
##  $ n_non_stop_unique_tokens    : num  0.815 0.792 0.664 0.666 0.541 ...
##  $ num_hrefs                   : num  4 3 3 9 19 2 21 20 2 4 ...
##  $ num_self_hrefs              : num  2 1 1 0 19 2 20 20 0 1 ...
##  $ num_imgs                    : num  1 1 1 1 20 0 20 20 0 1 ...
##  $ num_videos                  : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ average_token_length        : num  4.68 4.91 4.39 4.4 4.68 ...
##  $ num_keywords                : num  5 4 6 7 7 9 10 9 7 5 ...
##  $ kw_min_min                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_max_min                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_avg_min                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_min_max                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_max_max                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_avg_max                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_min_avg                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_max_avg                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_avg_avg                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ self_reference_min_shares   : num  496 0 918 0 545 8500 545 545 0 0 ...
##  $ self_reference_max_shares   : num  496 0 918 0 16000 8500 16000 16000 0
 0 ...
##  $ self_reference_avg_sharess  : num  496 0 918 0 3151 ...
##  $ is_weekend                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LDA_00                      : num  0.5003 0.7998 0.2178 0.0286 0.0286 .
..
##  $ LDA_01                      : num  0.3783 0.05 0.0333 0.4193 0.0288 ...
##  $ LDA_02                      : num  0.04 0.0501 0.0334 0.4947 0.0286 ...
##  $ LDA_03                      : num  0.0413 0.0501 0.0333 0.0289 0.0286 .
..
##  $ LDA_04                      : num  0.0401 0.05 0.6822 0.0286 0.8854 ...
##  $ global_subjectivity         : num  0.522 0.341 0.702 0.43 0.514 ...
##  $ global_sentiment_polarity   : num  0.0926 0.1489 0.3233 0.1007 0.281 ..
.
##  $ global_rate_positive_words  : num  0.0457 0.0431 0.0569 0.0414 0.0746 .
..
##  $ global_rate_negative_words  : num  0.0137 0.01569 0.00948 0.02072 0.012
13 ...
##  $ rate_positive_words         : num  0.769 0.733 0.857 0.667 0.86 ...
##  $ rate_negative_words         : num  0.231 0.267 0.143 0.333 0.14 ...
##  $ avg_positive_polarity       : num  0.379 0.287 0.496 0.386 0.411 ...
##  $ min_positive_polarity       : num  0.1 0.0333 0.1 0.1364 0.0333 ...
##  $ max_positive_polarity       : num  0.7 0.7 1 0.8 1 0.6 1 1 0.8 0.5 ...
##  $ avg_negative_polarity       : num  -0.35 -0.119 -0.467 -0.37 -0.22 ...
##  $ min_negative_polarity       : num  -0.6 -0.125 -0.8 -0.6 -0.5 -0.4 -0.5
```

```
 -0.5 -0.125 -0.5 ...
##  $ max_negative_polarity       : num  -0.2 -0.1 -0.133 -0.167 -0.05 ...
##  $ title_subjectivity          : num  0.5 0 0 0 0.455 ...
##  $ title_sentiment_polarity    : num  -0.188 0 0 0 0.136 ...
##  $ abs_title_subjectivity      : num  0 0.5 0.5 0.5 0.0455 ...
##  $ abs_title_sentiment_polarity: num  0.188 0 0 0 0.136 ...
##  $ shares                      : int  593 711 1500 1200 505 855 556 891 36
00 710 ...
##  $ news_channel                : Factor w/ 6 levels "Business","Entertainm
ent",..: 2 1 1 2 4 4 3 4 4 5 ...
##  $ published_day               : Factor w/ 7 levels "Monday","Tuesday",..:
 1 1 1 1 1 1 1 1 1 1 ...
##  $ published_date              : Date, format: "2013-01-07" "2013-01-07" .
..
##  $ published_month             : Factor w/ 12 levels "1","2","3","4",..: 1
 1 1 1 1 1 1 1 1 1 ...
##  $ published_year              : Factor w/ 2 levels "2013","2014": 1 1 1 1
 1 1 1 1 1 1 ...
```

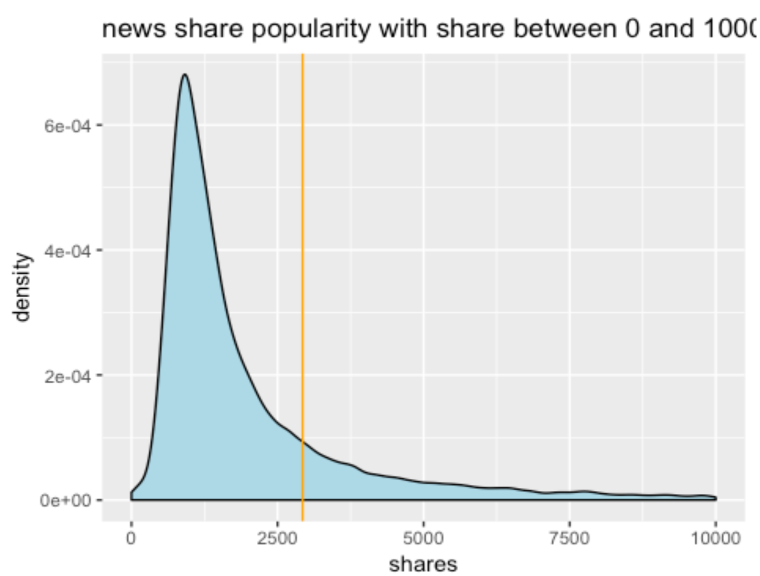
Exploratory data analysis

Density of news share

```
news_df %>% ggplot(mapping =aes(x=shares)) +
            geom_density(fill='lightblue') +
            geom_vline(xintercept = shares_boundry, col='orange') +
            ggtitle('news share popularity')
```



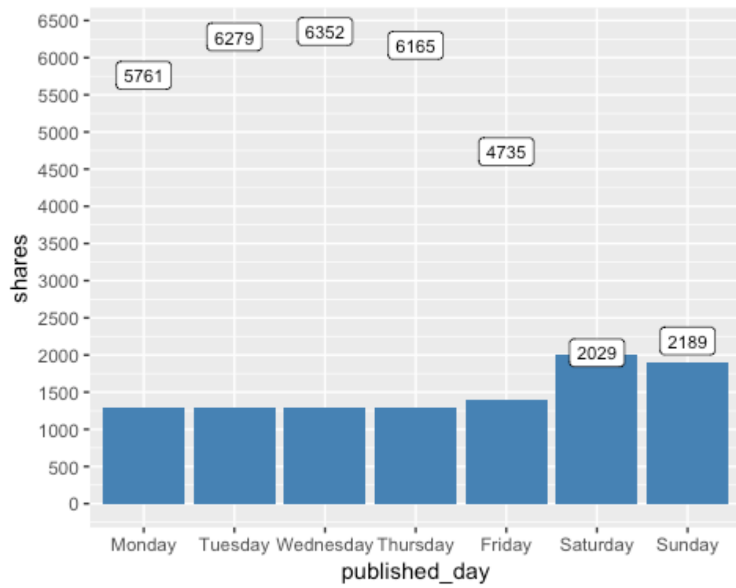
As can be seen shares of most articles are not so many and only a few successfully attract most people attention. And the below is a more careful look at it.



Time

day

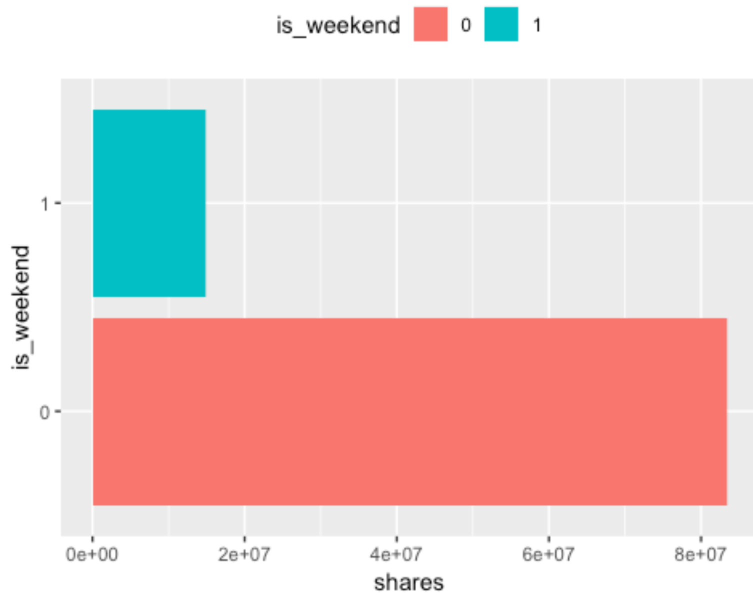
```
news_df %>% ggplot(aes(x=published_day, y=shares)) +
          geom_bar(stat='summary', fun.y='median', fill='steelblue') +
          scale_y_continuous(breaks=seq(0,15000,by=500)) +
          geom_label(stat='count', aes(label= ..count.., y= ..count..),size=
3)
```



It can be concluded that although most articles are published on week day the articles published on weekend are more likely popular. It kind of make sense. Since in weekend people have more time to browse webpages and share the interesting ones. Next I try to investigate the relationship between weekday and shares.

weekday

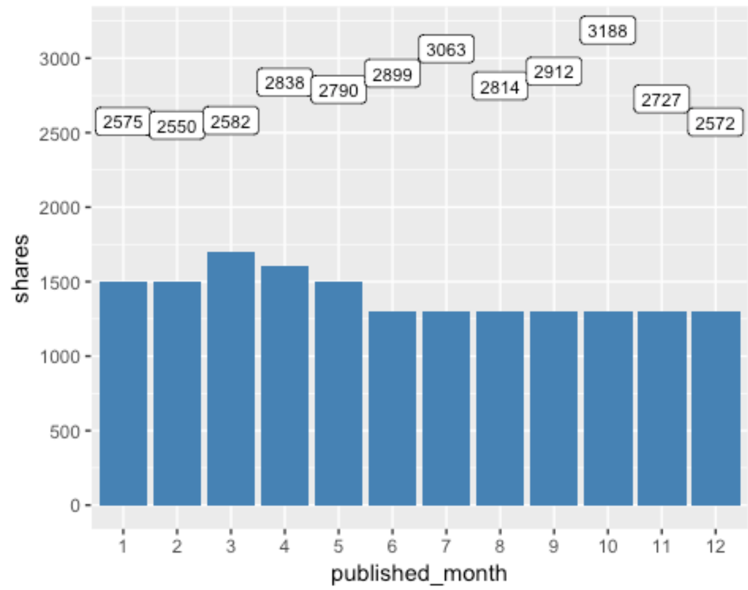
```
news_df <- news_df %>% mutate(is_weekend=as.factor(is_weekend))
news_df %>% ggplot(aes(x=is_weekend,y=shares,fill=is_weekend))+
            geom_bar(stat='identity',position =position_stack(reverse = TRU
E))+
            coord_flip() +
            theme(legend.position = "top")
```



In this case, it is obvious that the weekend will affect the number of shares of news.

month

```
news_df %>% ggplot(aes(x=published_month, y=shares)) +
          geom_bar(stat='summary', fun.y='median', fill='steelblue') +
          scale_y_continuous(breaks=seq(0,15000,by=500)) +
          geom_label(stat='count', aes(label= ..count.., y= ..count..),size=
3)
```



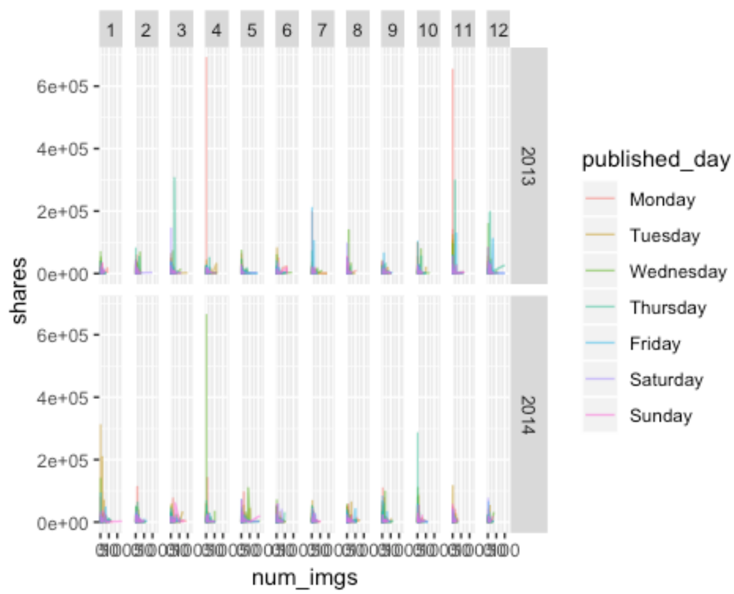
It seems like articles published in January, February, March, April and May have more shares. However, I do not find reasonable explanation for it. This is the further work. In my opinion, the effect of month is not so evident.

Digital Media

number of images

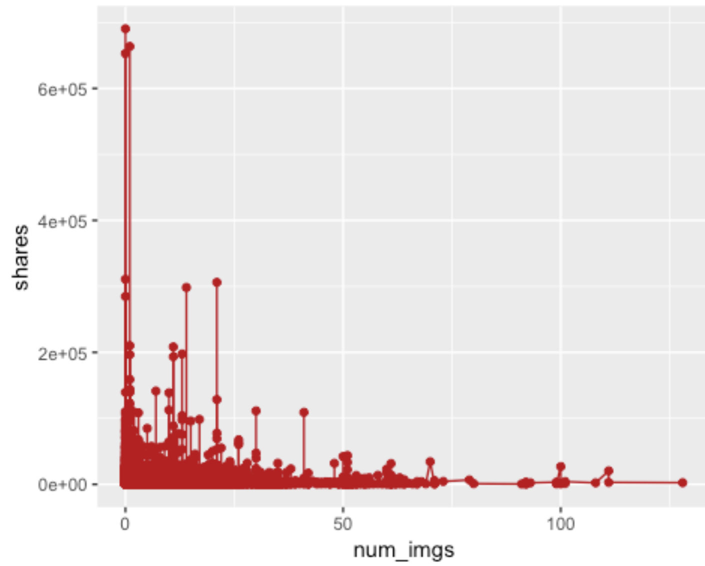
I draw a plot of relationship between num_imgs and shares in each day with respect to month and year.

```
news_df %>%
    ggplot(mapping = aes(x = num_imgs, y = shares, group = published_day, col
 = published_day)) +
    geom_line(alpha = 0.5) +
    facet_grid(published_year ~ published_month)
```

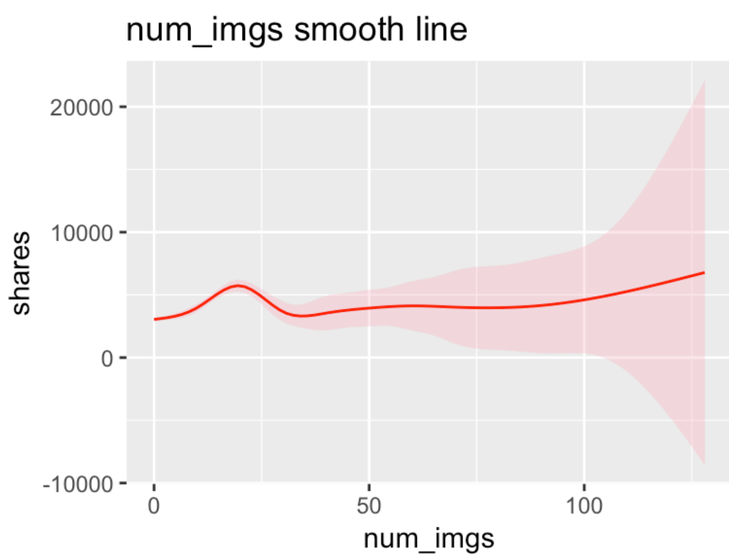


It seems like that the number of images has a little impact on number of shares.
Then I draw a plot of shares and num_imgs to see if there are some relationship in general.

```
news_df %>% ggplot(aes(x=num_imgs,y=shares))+
            geom_line(color='firebrick')+
            geom_point(color='firebrick')
```



Again, there is no obvious linear relationship between number of images and number of shares. More images do not mean that there would be more shares. And smooth line proves this conclusion.

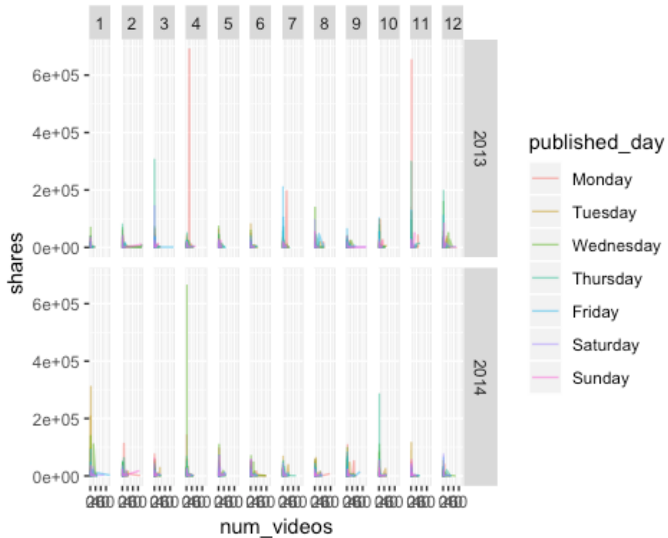


number of videos

I draw a plot of relationship between num_videos and shares in each day with respect to month and year.

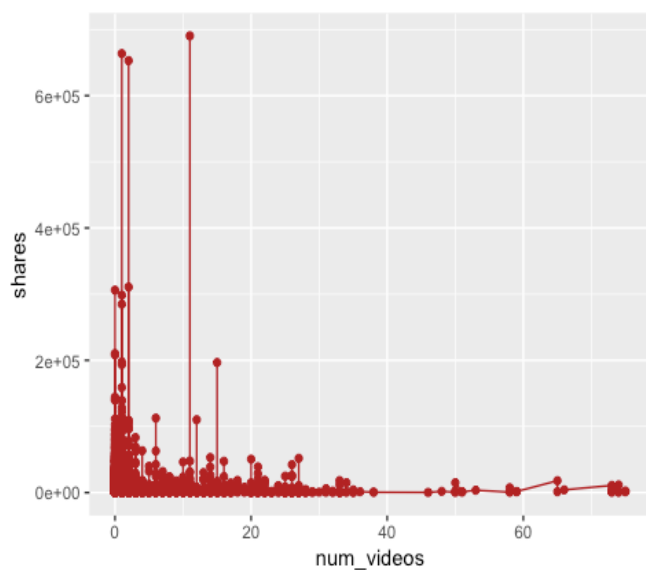
```
news_df %>%
    ggplot(mapping = aes(x = num_videos, y = shares, group = published_day, c
ol = published_day)) +
```

```r
    geom_line(alpha = 0.5) +
    facet_grid(published_year ~ published_month)
```

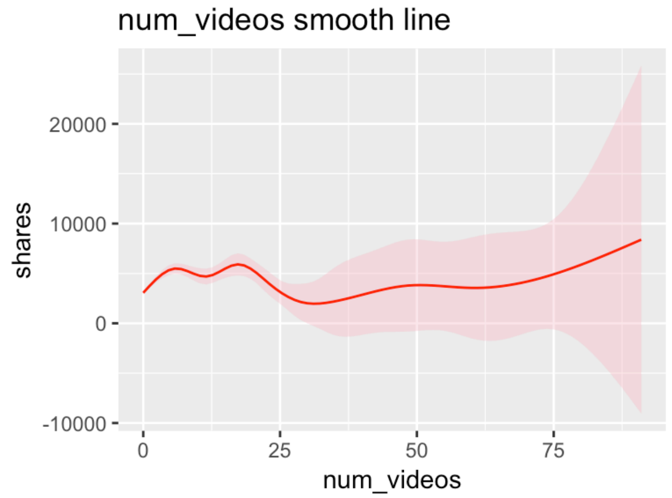


It seems like that the number of videos has a little impact on number of shares.
Then I draw a plot of shares and num_videos to see if there are some relationship in general.

```r
news_df %>% ggplot(aes(x=num_videos,y=shares))+
            geom_line(color='firebrick')+
            geom_point(color='firebrick')
```



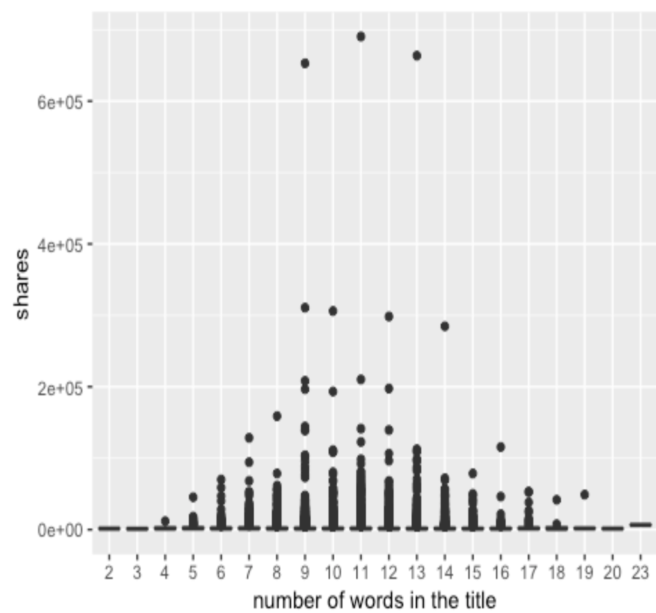
As can be seen there is no direct relationship between num_videos and shares. The smooth line also proves this conclusion.



Words

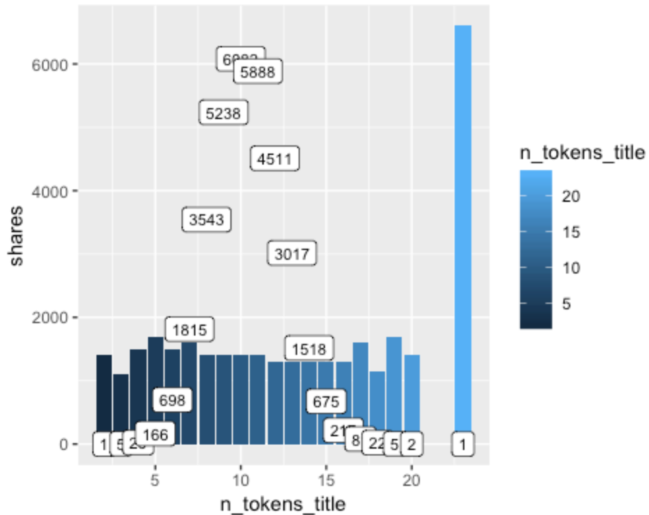
number of words in the title

```
ggplot(data=news_df, aes(x=factor(n_tokens_title), y=shares))+
        geom_boxplot() +
        labs(x='number of words in the title')
```



It seems like a not too long and too short title will be more attractive (9,10,11,12,13 words). And there are some extreme values of shares when number of tokens is equal to 9, 11, 13.

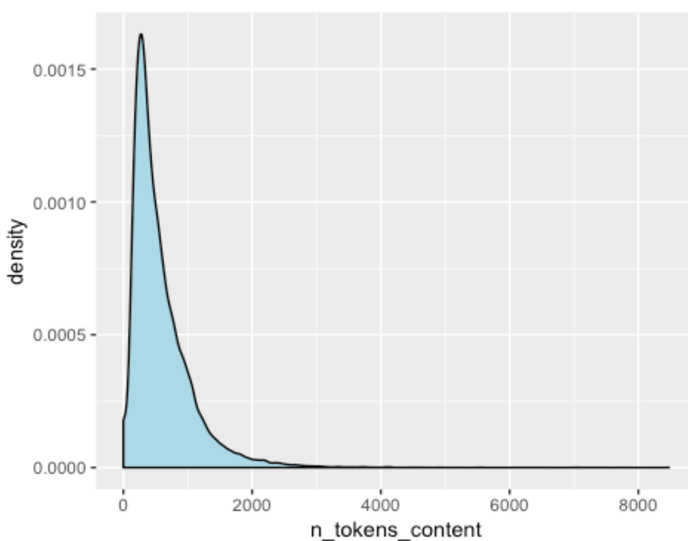
```
news_df %>% ggplot(aes(x=n_tokens_title,y=shares,fill=n_tokens_title))+
            geom_bar(stat='summary',fun.y='median')+
            geom_label(stat='count', aes(label= ..count.., y= ..count..),size
=3)
```



It seems like the number of words of title does not affect the popularity of articles. However, most authors choose a concise title with length with 8-12. Also, it shows that a successful article has a successful title. So, if an author has a good content, I suggest him/her write a tile with words number of 8-12.

number of words in the article

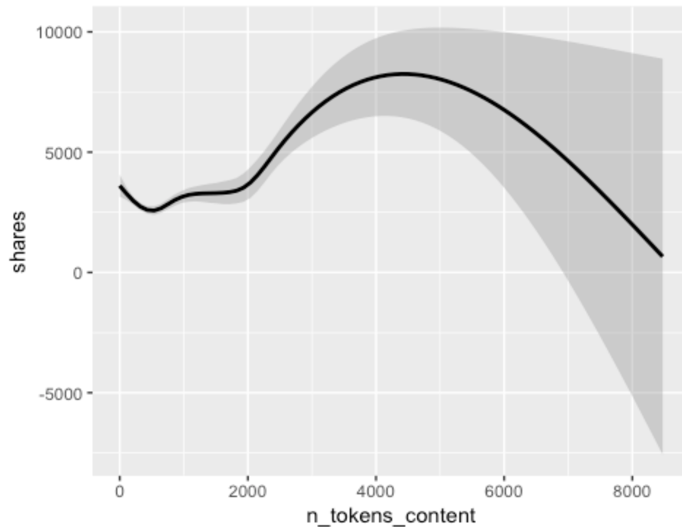
```
news_df %>% ggplot(mapping =aes(x=n_tokens_content)) +
            geom_density(fill='lightblue')
```



It shows that the number of words of most articles is between 0 and 2000. So an author can take this number as a reference.

```
news_df %>% ggplot(aes(x=n_tokens_content,y=shares))+
            #geom_line(color='firebrick')+
            #geom_point(color='firebrick')+
            geom_smooth(col = "black")
```

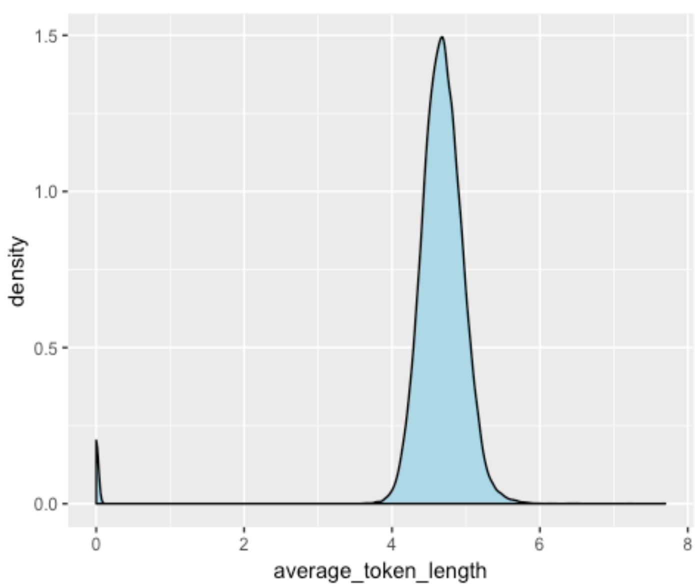
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



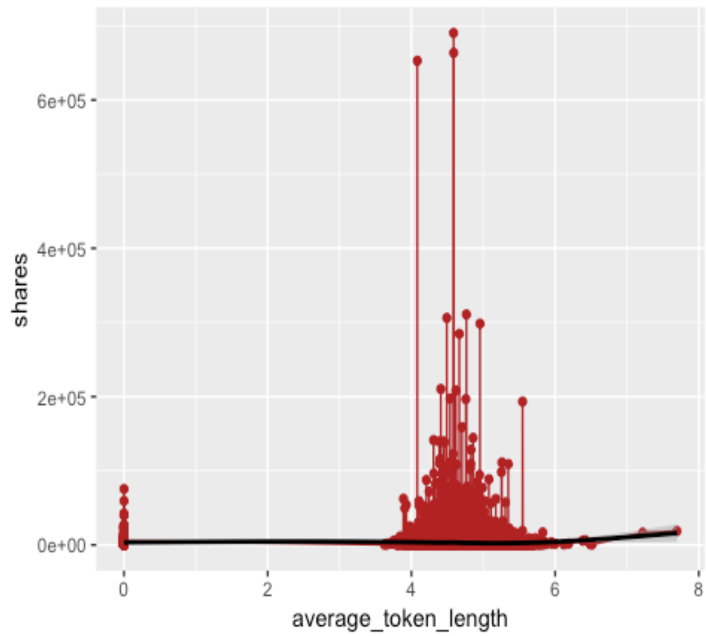
Based on the limited data, it can be concluded that when the number of words of articles in the range of (500,4000) more words indicates more shares.

average token length

```
news_df %>% ggplot(mapping =aes(x=average_token_length)) +
            geom_density(fill='lightblue')
```



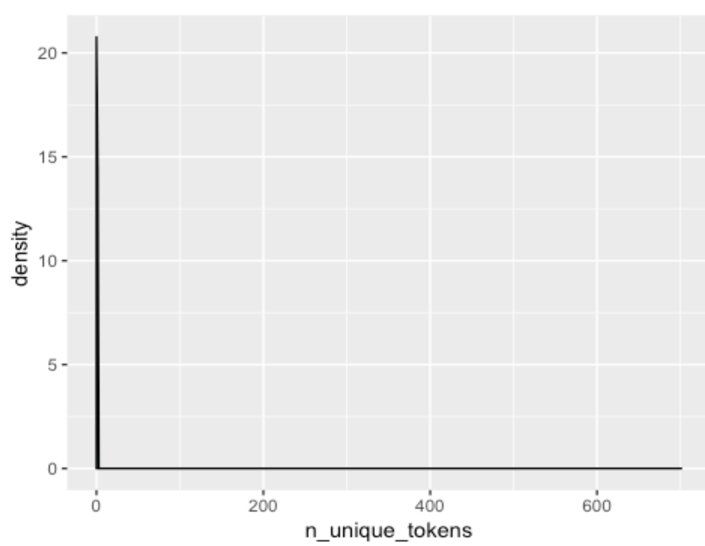

```
news_df %>% ggplot(aes(x=average_token_length,y=shares))+
            geom_line(color='firebrick')+
            geom_point(color='firebrick')+
            geom_smooth(col = "black")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



It can be concluded that most words have a length of 4-6 letters. Although I draw a plot to investigate the relationship between popularity and average token length I do not think it is reasonable to do that. In my opinion there should be no relationship between popularity and average token length.

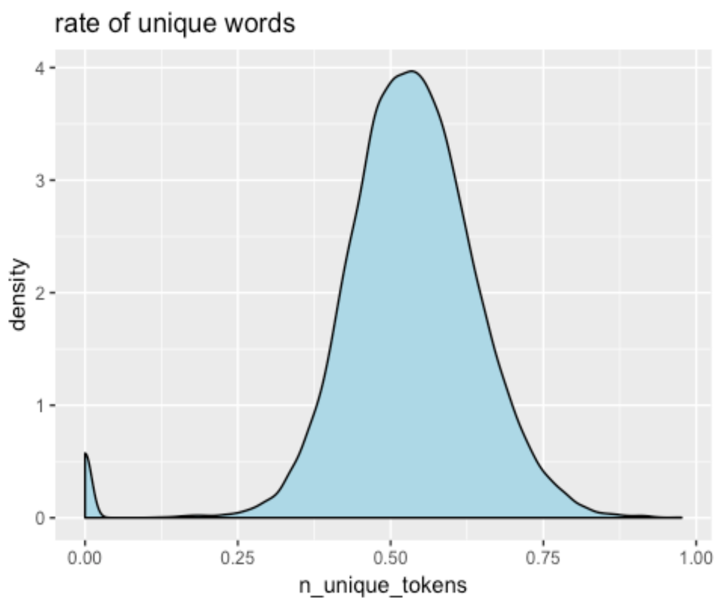
n_unique_tokens

```
news_df %>% ggplot(mapping =aes(x=n_unique_tokens)) +
            geom_density(fill='lightblue')
```



It shows that there are some extreme values. So, I take a careful look at it. And intuitively, the value of n_unique_tokens should between 0 and 1.

```
news_df %>% filter(n_unique_tokens<1) %>%
            ggplot(mapping =aes(x=n_unique_tokens, y=..density..)) +
            geom_density(fill='lightblue')+
            ggtitle('rate of unique words')
```



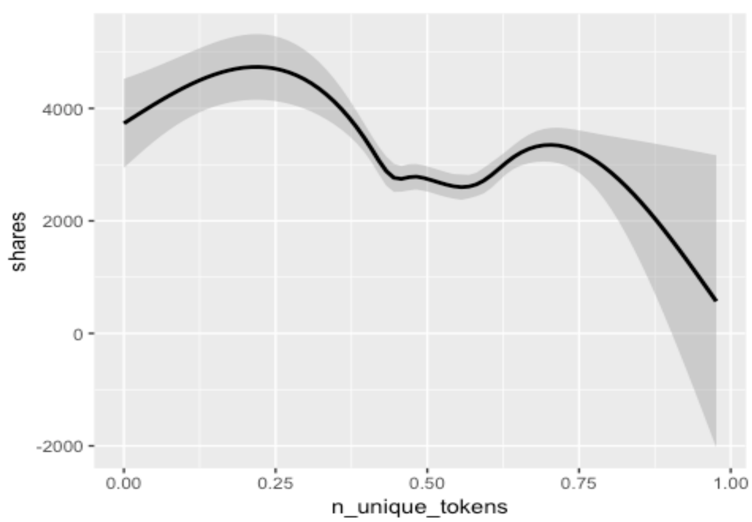
And I get the extreme values.

```
max(news_df$n_unique_tokens)

## [1] 701

sum(news_df$n_unique_tokens>1)

## [1] 1

news_df %>% filter(n_unique_tokens>1)

##   n_tokens_title n_tokens_content n_unique_tokens n_non_stop_words
## 1              9             1570             701             1042
##   n_non_stop_unique_tokens num_hrefs num_self_hrefs num_imgs num_videos
## 1                      650        11             10       51          0
##   average_token_length num_keywords kw_min_min kw_max_min kw_avg_min
## 1             4.696178            7         -1        778   143.7143
##   kw_min_max kw_max_max kw_avg_max kw_min_avg kw_max_avg kw_avg_avg
## 1      23100     843300   330442.9   2420.579   3490.599   2912.105
##   self_reference_min_shares self_reference_max_shares
## 1                       795                         0
##   self_reference_avg_sharess is_weekend LDA_00 LDA_01 LDA_02 LDA_03 LDA_04
## 1                   6924.375          0      0      0      0      0      0
##   global_subjectivity global_sentiment_polarity global_rate_positive_words
```

```
## 1                   0                          0                           0
##   global_rate_negative_words rate_positive_words rate_negative_words
## 1                          0                   0                   0
##   avg_positive_polarity min_positive_polarity max_positive_polarity
## 1                     0                     0                     0
##   avg_negative_polarity min_negative_polarity max_negative_polarity
## 1                     0                     0                     0
##   title_subjectivity title_sentiment_polarity abs_title_subjectivity
## 1                  0                        0                      0
##   abs_title_sentiment_polarity shares  news_channel published_day
## 1                            0   5900 Entertainment       Tuesday
##   published_date published_month published_year
## 1     2014-08-18               8           2014
```

Obviously, there are extrem values of number of shares. There is no sign indicates that there is evident relationship between rate of unique words and popularity. There is one weird thing: n_uniue_tokens mean the rate of unique words, but the maximum value of it is 701, which makes no sense. So, that is one point that needs more work. It seems that most common rate of unique words is between 0.375 and 0.625.

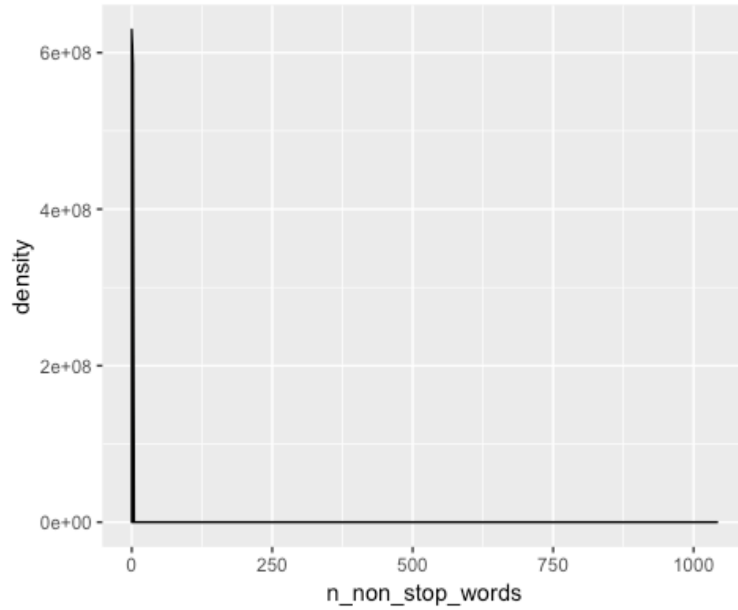
```
news_df %>% filter(n_unique_tokens<1) %>%
            ggplot(mapping=aes(x=n_unique_tokens,y=shares))+
            #geom_line(color='firebrick')+
            #geom_point(color='firebrick')+
            geom_smooth(col = "black")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Based on the limited data, the smooth function indicates that there is negative relationship between rate of unique tokens and shares. So, I would give a suggestion that an author should decrease the rate of unique tokens in contents.

n_non_stop_words:

```
news_df %>% ggplot(mapping =aes(x=n_non_stop_words)) +
            geom_density(fill='lightblue')
```




```
            ggtitle('rate of non-stop words')

## $title
## [1] "rate of non-stop words"
##
## attr(,"class")
## [1] "labels"

max(news_df$n_non_stop_words)

## [1] 1042

sum(news_df$n_non_stop_words>1)

## [1] 1

news_df %>% filter(n_non_stop_words>1)

##   n_tokens_title n_tokens_content n_unique_tokens n_non_stop_words
## 1              9             1570             701             1042
##   n_non_stop_unique_tokens num_hrefs num_self_hrefs num_imgs num_videos
## 1                      650        11             10       51          0
##   average_token_length num_keywords kw_min_min kw_max_min kw_avg_min
## 1             4.696178            7         -1        778   143.7143
##   kw_min_max kw_max_max kw_avg_max kw_min_avg kw_max_avg kw_avg_avg
## 1      23100     843300   330442.9   2420.579   3490.599   2912.105
##   self_reference_min_shares self_reference_max_shares
```

```
## 1                       795                            0
##   self_reference_avg_sharess is_weekend LDA_00 LDA_01 LDA_02 LDA_03 LDA_04
## 1                   6924.375          0      0      0      0      0      0
##   global_subjectivity global_sentiment_polarity global_rate_positive_words
## 1                   0                         0                          0
##   global_rate_negative_words rate_positive_words rate_negative_words
## 1                          0                   0                   0
##   avg_positive_polarity min_positive_polarity max_positive_polarity
## 1                     0                     0                     0
##   avg_negative_polarity min_negative_polarity max_negative_polarity
## 1                     0                     0                     0
##   title_subjectivity title_sentiment_polarity abs_title_subjectivity
## 1                  0                        0                      0
##   abs_title_sentiment_polarity shares  news_channel published_day
## 1                            0   5900 Entertainment       Tuesday
##   published_date published_month published_year
## 1     2014-08-18               8           2014
```

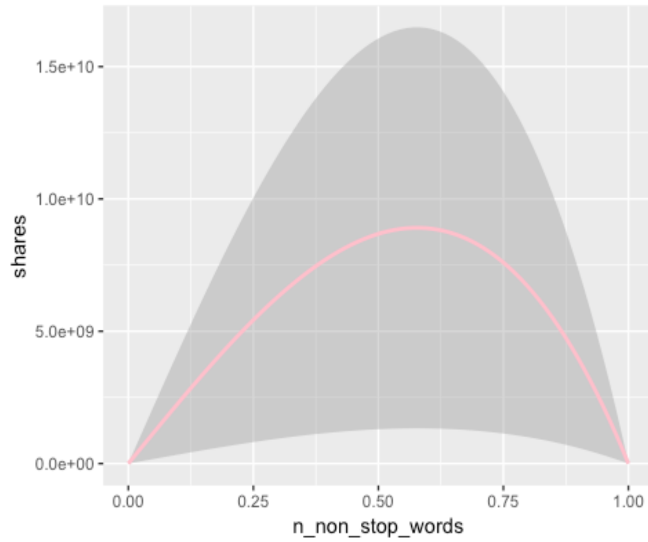
```
count(news_df$n_non_stop_words<1 &news_df$n_non_stop_words>=0.99)
```

```
##       x  freq
## 1 FALSE   539
## 2  TRUE 32971
```

Again, there is an outlier which is exactly the same outlier of n_non_stop_words. I delete this record. There is no sign indicates that there is evident relationship between rate of non-stop words and popularity. Rate of non-stop words of most articles are very close to 1 and only a few is 0.

```
news_df %>% filter(n_non_stop_words<2) %>%
            ggplot(mapping=aes(x=n_non_stop_words,y=shares))+
            #geom_line(color='firebrick')+
            #geom_point(color='firebrick')+
            geom_smooth(col = "pink")
```

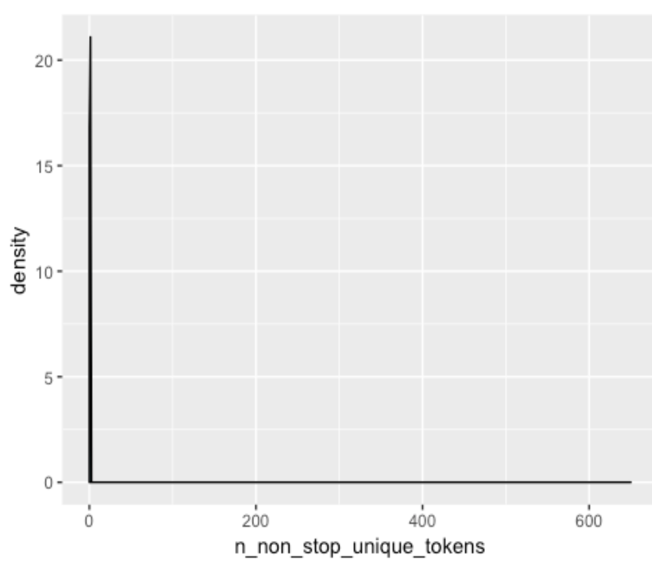
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Based on the limited data, the smooth function indicates that when rate of non-stop word is 0.625, the number of shares is maximum. So, if an author want to write a popular article he/she can try to control the rate of non-stop word around 0.65.

unique non-stop words

```
news_df %>% ggplot(mapping =aes(x=n_non_stop_unique_tokens)) +
            geom_density(fill='lightblue')
```




```
## $title
## [1] "rate of non-stop words"
##
## attr(,"class")
## [1] "labels"

max(news_df$n_non_stop_unique_tokens)
```

```
## [1] 650
```

```r
count(news_df$n_non_stop_unique_tokens>1)
```

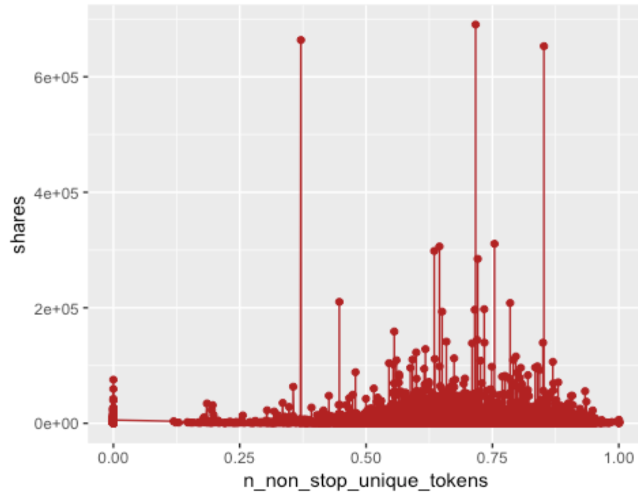
```
##       x  freq
## 1 FALSE 33509
## 2  TRUE     1
```

```r
news_df %>% filter(n_non_stop_unique_tokens>1)
```

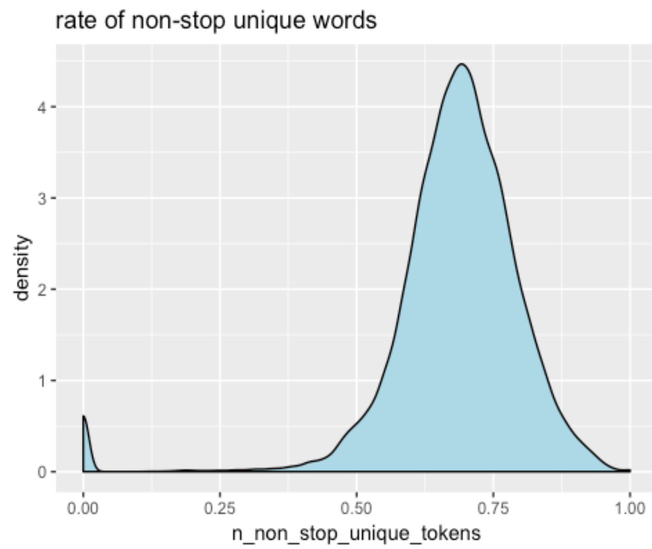
```
##   n_tokens_title n_tokens_content n_unique_tokens n_non_stop_words
## 1              9             1570             701             1042
##   n_non_stop_unique_tokens num_hrefs num_self_hrefs num_imgs num_videos
## 1                      650        11             10       51          0
##   average_token_length num_keywords kw_min_min kw_max_min kw_avg_min
## 1             4.696178            7         -1        778   143.7143
##   kw_min_max kw_max_max kw_avg_max kw_min_avg kw_max_avg kw_avg_avg
## 1      23100     843300   330442.9   2420.579   3490.599   2912.105
##   self_reference_min_shares self_reference_max_shares
## 1                       795                         0
##   self_reference_avg_sharess is_weekend LDA_00 LDA_01 LDA_02 LDA_03 LDA_04
## 1                   6924.375          0      0      0      0      0      0
##   global_subjectivity global_sentiment_polarity global_rate_positive_words
## 1                   0                         0                          0
##   global_rate_negative_words rate_positive_words rate_negative_words
## 1                          0                   0                   0
##   avg_positive_polarity min_positive_polarity max_positive_polarity
## 1                     0                     0                     0
##   avg_negative_polarity min_negative_polarity max_negative_polarity
## 1                     0                     0                     0
##   title_subjectivity title_sentiment_polarity abs_title_subjectivity
## 1                  0                        0                      0
##   abs_title_sentiment_polarity shares  news_channel published_day
## 1                            0   5900 Entertainment       Tuesday
##   published_date published_month published_year
## 1     2014-08-18               8           2014
```

Again, there is the same extreme value.

```r
news_df %>% filter(n_non_stop_unique_tokens<=1) %>%
            ggplot(mapping=aes(x=n_non_stop_unique_tokens,y=shares))+
            geom_line(color='firebrick')+
            geom_point(color='firebrick')
```

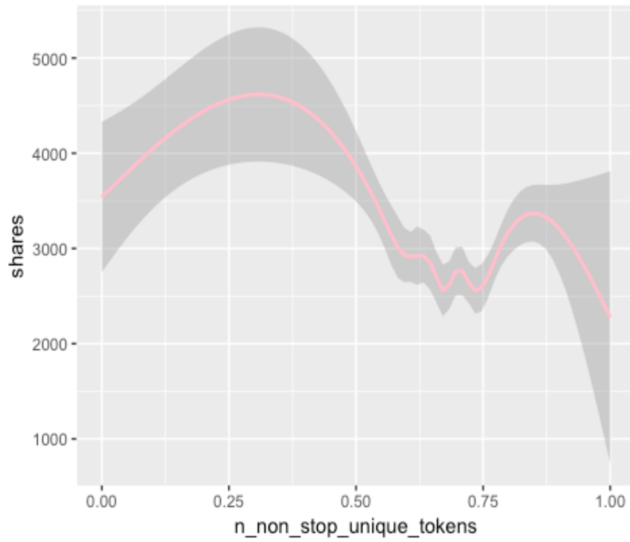



```
news_df %>% filter(n_non_stop_unique_tokens<=1) %>%
            ggplot(mapping =aes(x=n_non_stop_unique_tokens, y=..density..)) +
            geom_density(fill='lightblue')+
            ggtitle('rate of non-stop unique words')
```



Rates of unique non-stop words of most articles are between 0.5 and 1. Moreover articles with rate of unique non-stop words around 0.65 tend to be more popular.

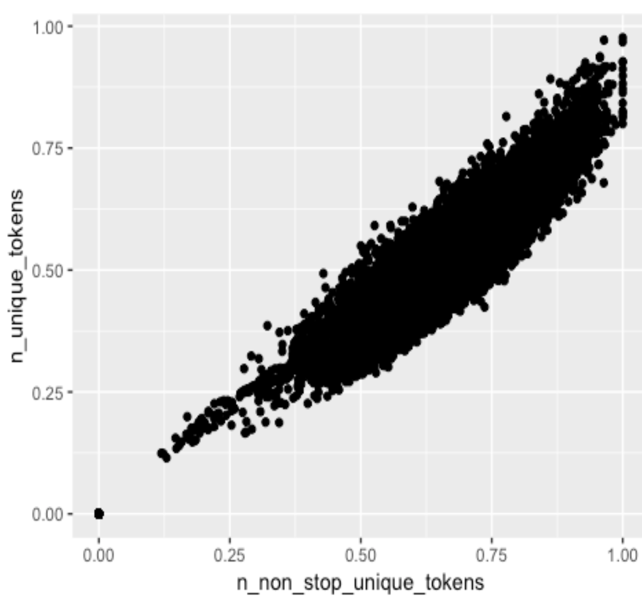
```
news_df %>% filter(n_non_stop_unique_tokens<=1) %>%
            ggplot(mapping=aes(x=n_non_stop_unique_tokens,y=shares))+
            geom_smooth(color='pink')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Based on the limited data, the smooth shows that if an author wants to right a popular article, he/she probably wants to control the rate of unique non-stop words in the content between 0 and 0.5.

I think there might be some relationship between rate of unique non-stop words and rate of unique words. So, I draw a plot.

```
news_df %>% subset(n_unique_tokens<=1&n_non_stop_unique_tokens<=1, select=c('
n_unique_tokens', 'n_non_stop_unique_tokens')) %>%
            ggplot(aes(x=n_non_stop_unique_tokens,y=n_unique_tokens))+
            geom_point()
```

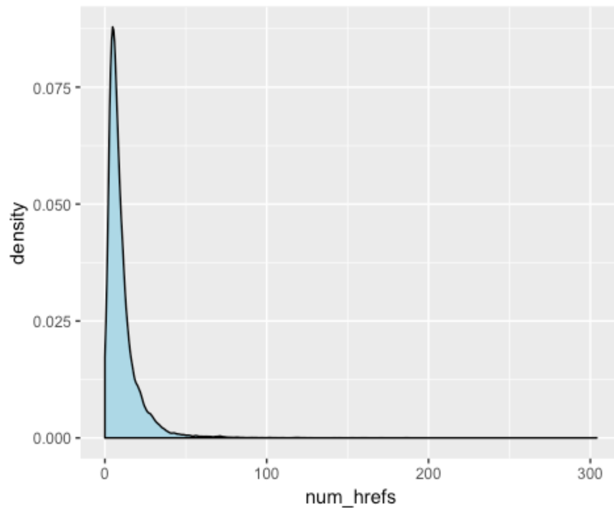


It shows that there is an apparent linear relationship between rate of unique non-stop words and rate of unique words. So, when I build a model I will only choose rate of unique non-stop words as one of variables in models.

Links

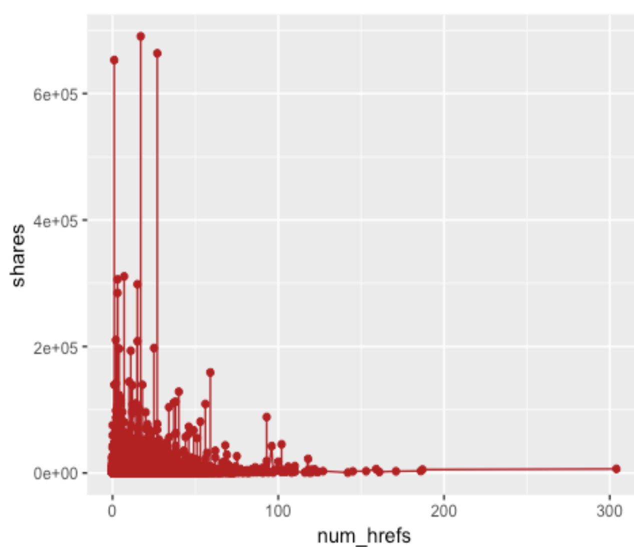
links number of links

```
news_df %>% ggplot(mapping =aes(x=num_hrefs, y=..density..)) +
            geom_density(fill='lightblue')
```

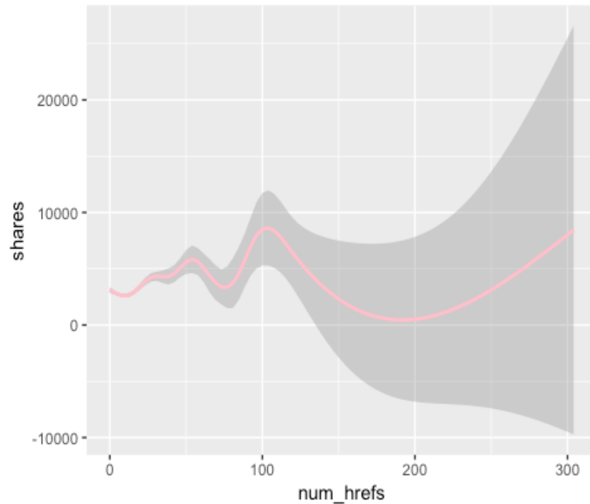


Number of links of most articles is between 0 and 50, which is reasonable. Then I look at the relationship between number of links and popularity.

```
news_df %>% ggplot(aes(x=num_hrefs,y=shares))+
            geom_line(color='firebrick')+
            geom_point(color='firebrick')
```



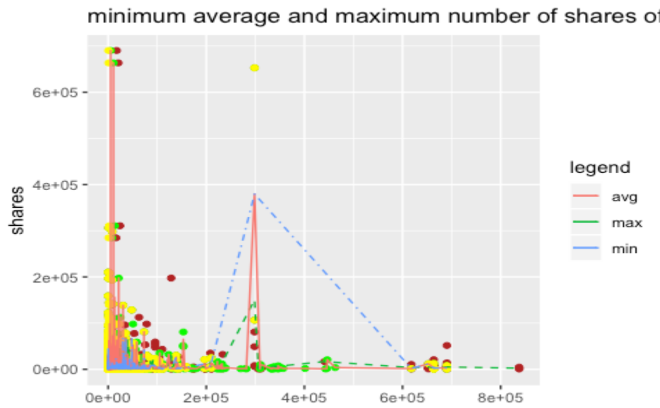
```r
news_df %>% ggplot(aes(x=num_hrefs,y=shares))+
            geom_smooth(color='pink')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



It can be concluded that an article with more links do not tend to be more popular. Conversely, an article with 0-125 links is more likely to be popular.

average number of Mashable links

```r
#cols<-c('firebrick', 'yellow', 'green')
news_df %>% ggplot(aes(x=self_reference_max_shares,y=shares))+
            geom_point(color='firebrick')+
            geom_point(aes(x=self_reference_avg_sharess,y=shares),color='gree
n')+
            geom_point(aes(x=self_reference_min_shares,y=shares),color='yello
w')+
            geom_line(aes(x=self_reference_max_shares,y=shares,color='max'),s
tat = 'summary',fun.y='mean',show.legend=TRUE,linetype = "dashed")+
            geom_line(aes(x=self_reference_avg_sharess,y=shares,color='avg'),
stat = 'summary',fun.y='mean',show.legend=TRUE)+
            geom_line(aes(x=self_reference_min_shares,y=shares,color='min'),s
tat = 'summary',fun.y='mean',show.legend=TRUE,linetype="dotdash")+
            scale_color_discrete('legend')+
            xlab('')+
            ggtitle('minimum average and maximum number of shares of Mashable
 links')
```

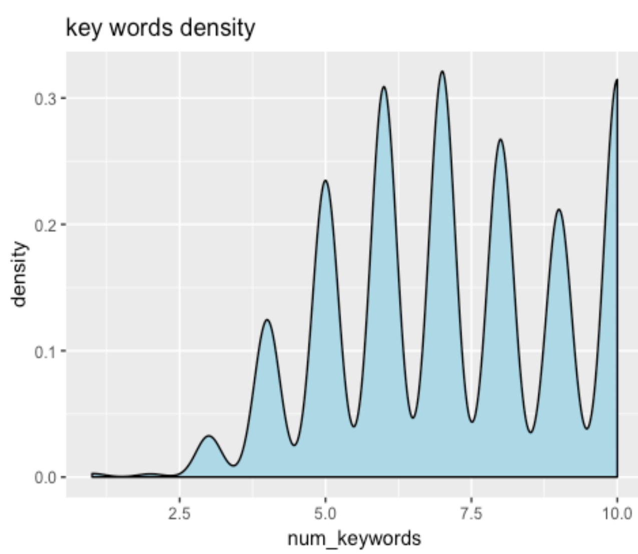


As can be seen in the above graph, the relationships between shares and minimum, average and maximum number of shares of Mashable links are similar. And the trend of minimum, average and maximum number of shares of Mashable links are similar. Also, an article with more referenced articles in Mashable is not definitely more popular.

Keywords

density of keywords

```
news_df %>% ggplot(mapping =aes(x=num_keywords)) +
            geom_density(fill='lightblue') +
            ggtitle('key words density')
```

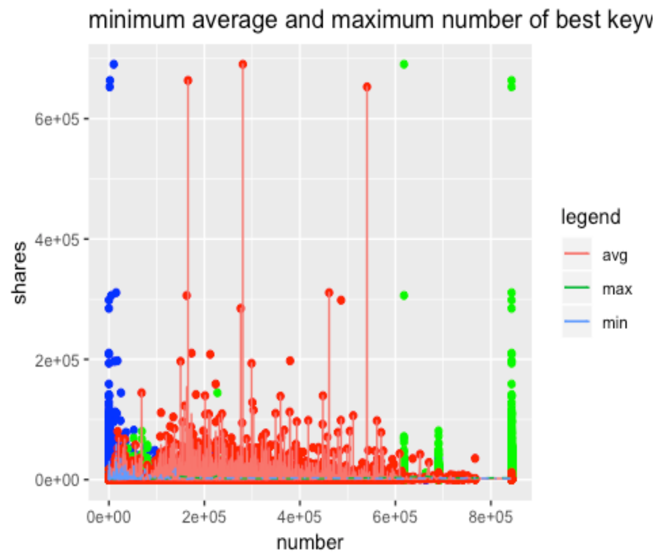


It shows that most articles have 4-10 key words.

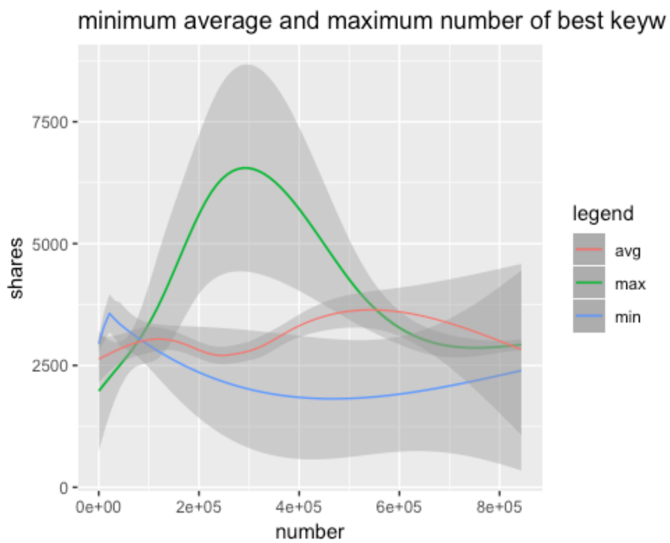
best Keyword

```
news_df %>% ggplot(aes(x=kw_min_max,y=shares))+
            geom_point(color='blue')+
            geom_point(aes(x=kw_max_max,y=shares),color='green')+
```

```
            geom_point(aes(x=kw_avg_max,y=shares),color='red')+
            geom_line(aes(x=kw_avg_max,y=shares,color='avg'),
                      stat = 'summary',fun.y='mean',show.legend=TRUE)+
            geom_line(aes(x=kw_max_max,y=shares,color='max'),
                      stat = 'summary',fun.y='mean',show.legend=TRUE,linetype
 = "dashed")+
            geom_line(aes(x=kw_min_max,y=shares,color='min'),
                      stat = 'summary',fun.y='mean',show.legend=TRUE,linetype
="dotdash")+
            scale_color_discrete('legend')+
            xlab('number')+
            ggtitle('minimum average and maximum number of best keyword')
```



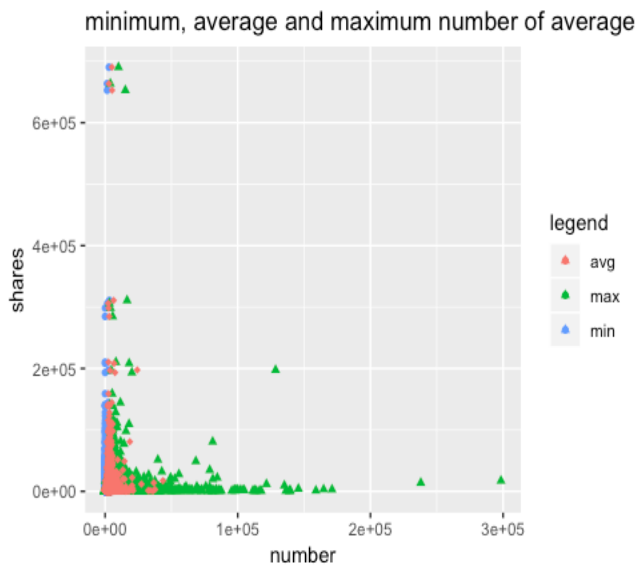

```
news_df %>% ggplot(aes(y=shares))+
            geom_smooth(mapping = aes(x=kw_max_max, y=shares,color='max'),lwd
=0.6)+
            geom_smooth(mapping = aes(x=kw_min_max, y=shares,color='min'),lwd
=0.5)+
             geom_smooth(mapping = aes(x=kw_avg_max, y=shares,color='avg'),lw
d=0.4)+
            scale_color_discrete('legend')+
            xlab('number')+
            ggtitle('minimum average and maximum number of best keyword')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



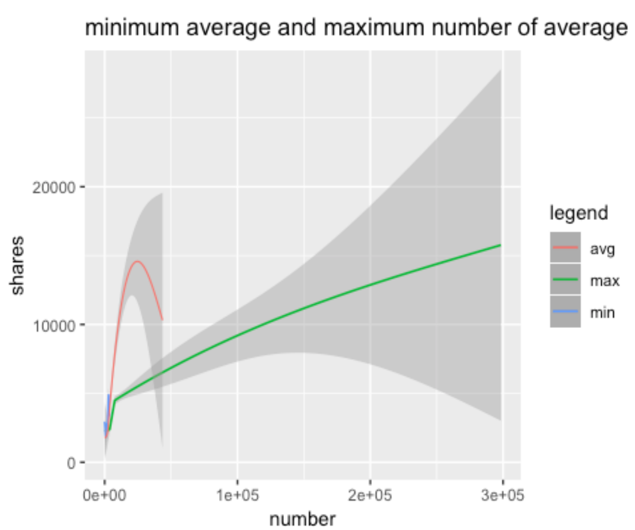
The relationship between shares and min, max and avg shares of best keywords are different. It seems that the best keyword of an article could affect the popularity of it. Since when the best keyword of an article is better, the number of shares tends to be bigger, which accords with common sense, i.e. popular keywords lead to popular articles.

average keyword

```
news_df %>% ggplot(aes(y=shares))+
            geom_point(aes(x=kw_min_avg,y=shares,color='min'),shape=16,show.l
egend = TRUE)+
            geom_point(aes(x=kw_max_avg,y=shares,color='max'),shape=17,show.l
egend = TRUE)+
            geom_point(aes(x=kw_avg_avg,y=shares,color='avg'),shape=18,show.l
egend = TRUE)+
            xlab('number')+
            scale_color_discrete('legend')+
            ggtitle('minimum, average and maximum number of average keyword')
```



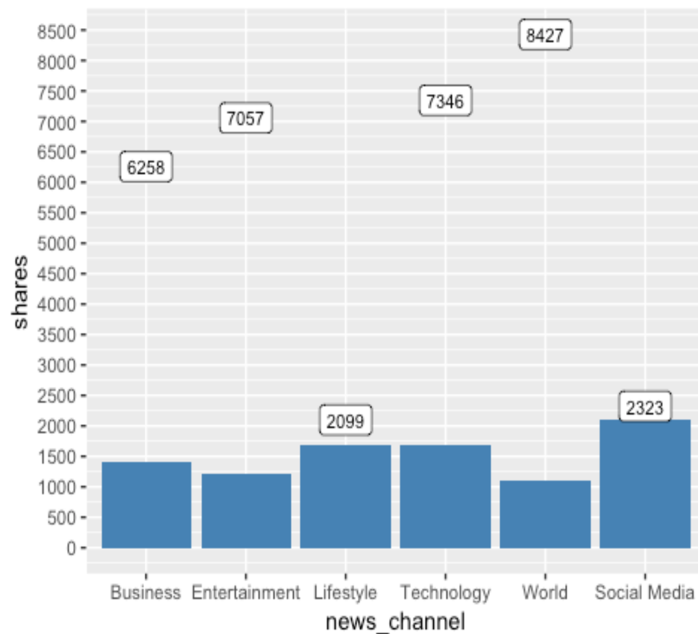

```
news_df %>% ggplot(aes(y=shares))+
         geom_smooth(mapping = aes(x=kw_max_avg, y=shares,color='max'),lwd
=0.6)+
         geom_smooth(mapping = aes(x=kw_min_avg, y=shares,color='min'),lwd
=0.5)+
          geom_smooth(mapping = aes(x=kw_avg_avg, y=shares,color='avg'),lw
d=0.4)+
         scale_color_discrete('legend')+
         xlab('number')+
         ggtitle('minimum average and maximum number of average keyword')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



The situation is kind of same as that of best keywords. Still popular keywords lead to popular articles.

channels

```
news_df %>% ggplot(aes(x=news_channel, y=shares)) +
          geom_bar(stat='summary', fun.y='median', fill='steelblue') +
          scale_y_continuous(breaks=seq(0,15000,by=500)) +
          geom_label(stat='count', aes(label= ..count.., y= ..count..),size=
3)
```



Obviously, the type of channels has a big impact on number of shares.

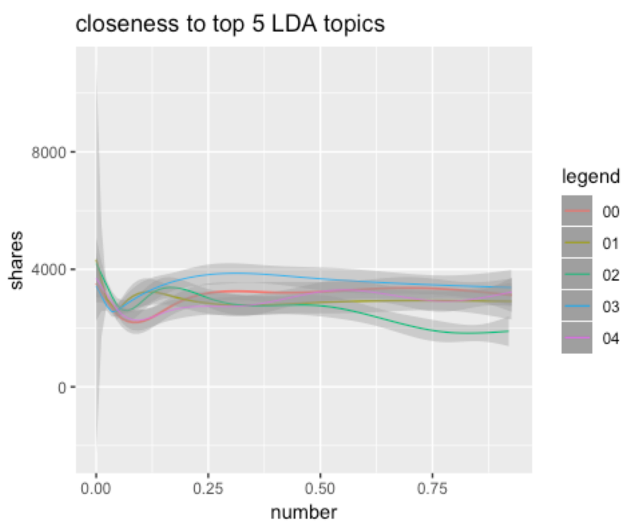
Natural Language Processing

For the variables in this section, I pick some typical variables to do analysis.

LDA

```
news_df %>% ggplot(aes(y=shares))+
           #geom_point(aes(x=LDA_00, y=shares),shape=16,color='pink',show.le
gend = TRUE)+
           geom_smooth(mapping = aes(x=LDA_00, y=shares,color='00'),lwd=0.6)
+
           #geom_point(aes(x=LDA_01, y=shares),shape=17,color='yellow',show.
legend = TRUE)+
           geom_smooth(mapping = aes(x=LDA_01, y=shares,color='01'),lwd=0.5)
+
```

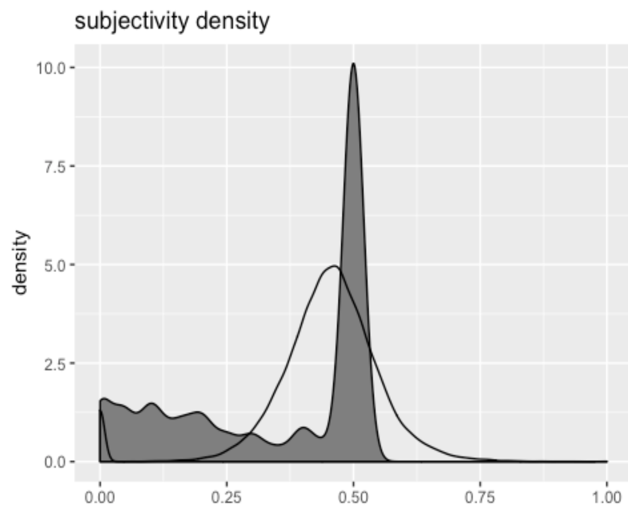
```
            #geom_point(aes(x=LDA_02, y=shares),shape=18,color='blue',show.le
gend = TRUE)+
            geom_smooth(mapping = aes(x=LDA_02, y=shares,color='02'),lwd=0.4)
+
            #geom_point(aes(x=LDA_03, y=shares),shape=19,color='green',show.l
egend = TRUE)+
            geom_smooth(mapping = aes(x=LDA_03, y=shares,color='03'),lwd=0.3)
+
            #geom_point(aes(x=LDA_04, y=shares),shape=20,color='orange',show.
legend = TRUE)+
            geom_smooth(mapping = aes(x=LDA_04, y=shares,color='04'),lwd=0.2)
+
            xlab('number')+
            scale_color_discrete('legend')+
            ggtitle('closeness to top 5 LDA topics')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



It can be concluded that an article which is more close to topic 3 is more likely popular. On the contrary, an article which is more close to topic 3 is less likely popular.

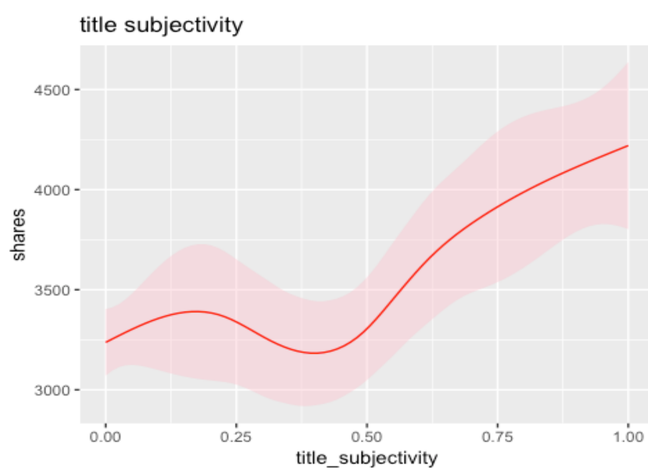
title and global subjectivity

```
news_df %>% ggplot(aes(x=abs_title_subjectivity,y=..density..))+
            geom_density(fill='grey50')+
            geom_density(aes(x=global_subjectivity,y=..density..))+
            xlab('')+
            ggtitle('subjectivity density')
```

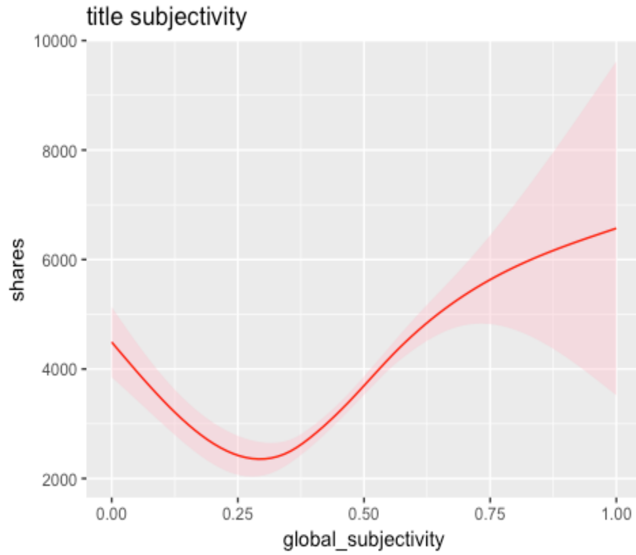


It shows that the distributions of abs_title_subjectivity and global_subjectivity are similar, i.e. subjectivities of most articles and titles are around 0.5.

```
news_df %>% ggplot(aes(x=title_subjectivity ,y=shares))+
            #geom_point(shape=16,color='orange')+
            geom_smooth(lwd = 0.5, col = "red", fill = "pink")+
            ggtitle('title subjectivity')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
news_df %>% ggplot(aes(x=global_subjectivity ,y=shares))+
            #geom_point(shape=16,color='orange')+
            geom_smooth(lwd = 0.5, col = "red", fill = "pink")+
            ggtitle('title subjectivity')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



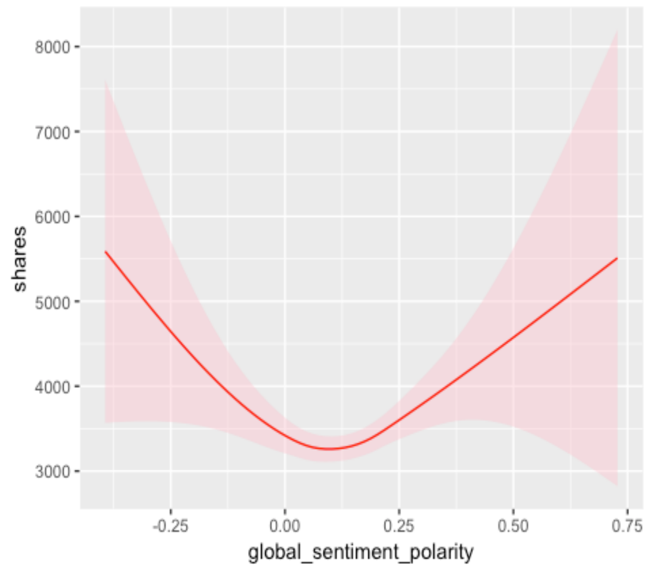
It seems like the relationships between shares and global subjectivity, title subjectivity are similar, which have decrease first and then increase after that. It can be concluded that an article which is more subjective tends to be more popular when its global subjectivity is greater than 0.33.

title and global polarity

```
news_df %>% ggplot(aes(x=title_sentiment_polarity ,y=shares))+
            #geom_point(shape=16,color='orange')+
            geom_smooth(lwd = 0.5, col = "red", fill = "pink")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


```
news_df %>% ggplot(aes(x=global_sentiment_polarity ,y=shares))+
            #geom_point(shape=16,color='orange')+
            geom_smooth(lwd = 0.5, col = "red", fill = "pink")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

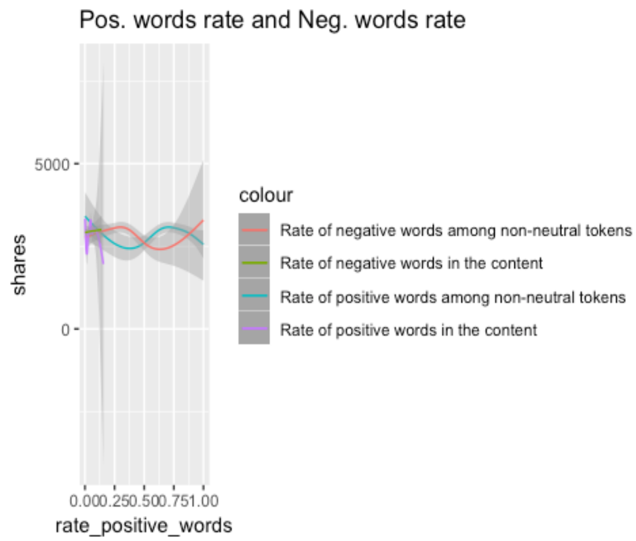


It seems like the relationships between shares and global polarity, title polarity are similar, which have decrease first and then increase after that. It can be concluded that an article whose abs global polarity is larger tends to be more popular.

pos. words rate and neg. words rate

```
news_df %>% ggplot(aes(y=shares))+
            #geom_point(shape=16,color='orange')+
            geom_smooth(mapping=aes(x=rate_positive_words,y=shares,
                                    color='Rate of positive words among non-n
eutral tokens'),
                        lwd = 0.5)+
            geom_smooth(mapping=aes(x=rate_negative_words,y=shares,
                                    color='Rate of negative words among non-n
eutral tokens'),
                        lwd = 0.5)+
            geom_smooth(mapping=aes(x=global_rate_positive_words,y=shares,
                                    color='Rate of positive words in the cont
ent'),
                        lwd = 0.5)+
            geom_smooth(mapping=aes(x=global_rate_negative_words,y=shares,
                                    color='Rate of negative words in the cont
ent'),
                        lwd = 0.5)+

            ggtitle('Pos. words rate and Neg. words rate')

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



In this case, it is hard to decide the best rate for rate of positive words among non-neutral tokens, rate of negative words among non-neutral tokens, rate of positive words in the content and rate of negative words in the content.

Conclusion

1. An author has better to publish news on weekend to get a high number of shares of articles.
2. Articles in the topic of lifestyle and social media are more likely to be shared.
3. A concise title helps article to be more popular.
4. Readers like subjective and polar articles.
5. An author is recommended to write an article with popular keywords.
6. An article with a low rate of unique tokens in contents tend to be popular.

Reference

[1] UCI. Online News Popularity Data Set. Available at: https://archive.ics.uci.edu/ml/datasets/online+news+popularity#
[2] K. Fernandes, P. Vinagre and P. Cortez. 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.
[3] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
[4] De Smedt, T., Nijs, L., Daelemans, W.: Creative web services with pattern. In: Proceedings of the Fifth International Conference on Computational Creativity (2014)

[5] Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commu- nications of the ACM 53(8), 80–88 (2010)
[6] Tatar, A., Antoniadis, P., De Amorim, M.D., Fdida, S.: From popularity prediction to ranking online news. Social Network Analysis and Mining 4(1), 1–12 (2014)