1 Assignment 2. Linear regression.

1.1 Selection of features

First of all, let's take a look at all continuous features' scatterplot to indentify which of them are linear dependent. Some features have nearly log-normal distributions, so, for more accurate and reliable linear regression we will logarithm these features.

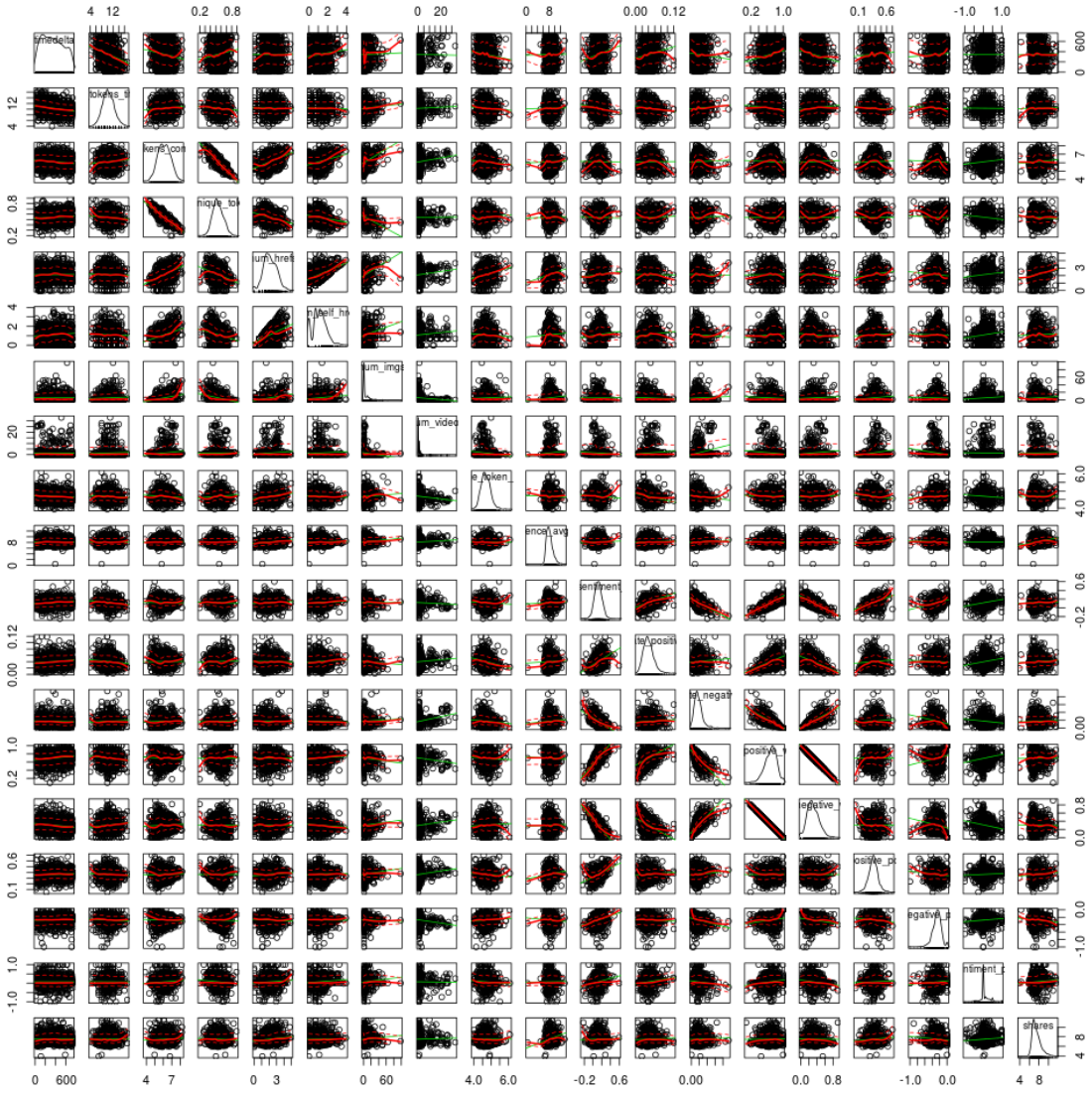
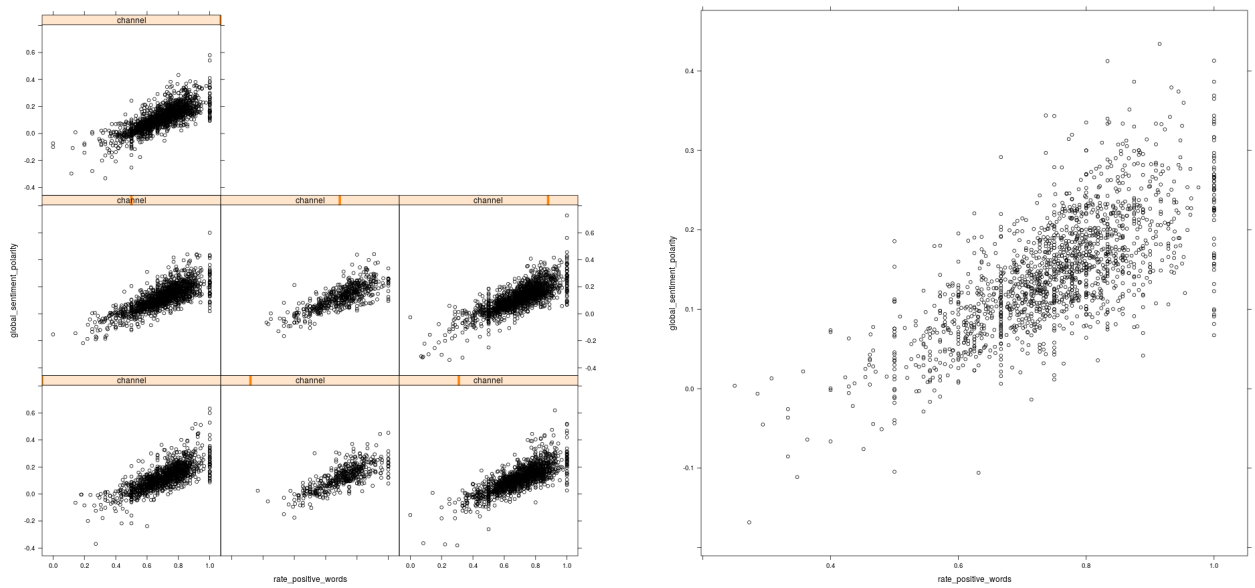


Figure 1: Scatterplot matrix of all considered continuous features

As we can see from the scatterplot, the majority of pairs are not linear dependent. Fortunately, `global_sentiment_pola` and `rate_positive_words` are linear dependent and we can easily understand why: they measure practicaly the same characteristic. The first feature is normalized from 0 to 1 (in out data from 0 to 0.7), the second one takes values from 0 to 1.

We will predict sentiment polarity over positive words rate. As we have rather heterogeneous data, let's make sure we won't be able to do our regression better with the help of groupping by `Channel` (Figure **??**, a).

All chanells looks very similar, and we dicided to consider only technical channel (just to reduce the sample size). After all these actions our scatterplot looks like at (Figure **??**, b). Further at this section we will call the



a) b)

Figure 2: Groupped by channel (a) and only technical channel (b) dependece between `global_sentiment_polarity` and `rate_positive_words`.

predicted and the prediction features just `global_sentiment_polarity` and `rate_positive_words`, implying we work with only one channel.

1.2 Model of linear regression

Using basic functions in R, we have built a linear regression with slope equals 0.4341 and intercept equals -0.1788. The results of the regression you can see in Figure **??**.

The slope is significantly positive (p-value equals 0) and it's not surpising: the more positive words are in the article, the more text is of positive polarity.

1.3 Correlation and determinacy coefficients

The correlation equals 0.7170736 and the coefficient of determination equals 0.5142 (ajusted is 0.5139). As we know from the definition of the coefficient of determination, R^2 measures of how well the regression line approximates the real data points and equals the ratio of explained variance. It's believed in practice that $R^2 > 0.5$ is acceptable, but not enough accurate.

Particularly the value of 0.5139 means that about 51% of variablility between the two `global_sentiment_polarity` and `rate_positive_words` is captured by the linear model built with linear regression and the remaining 49% of variability still remains unaccounted for.

In another words the value of determination coefficient R^2 shows the rate of decrease of the variance of `global_sentiment_polarity` after its linear relation to `rate_positive_words` has been taken into account by the regression.

1.4 Bootstrap

We have conducted 5000 boostrap trials to estimate 95% confidence intervals of slope, intercept and correlation coefficient. The results are summarized by the histograms shown in the figures **??**-**??**.

It can be easily seen that histograms are pretty similar to the normal type of distribution. But let us prove that.

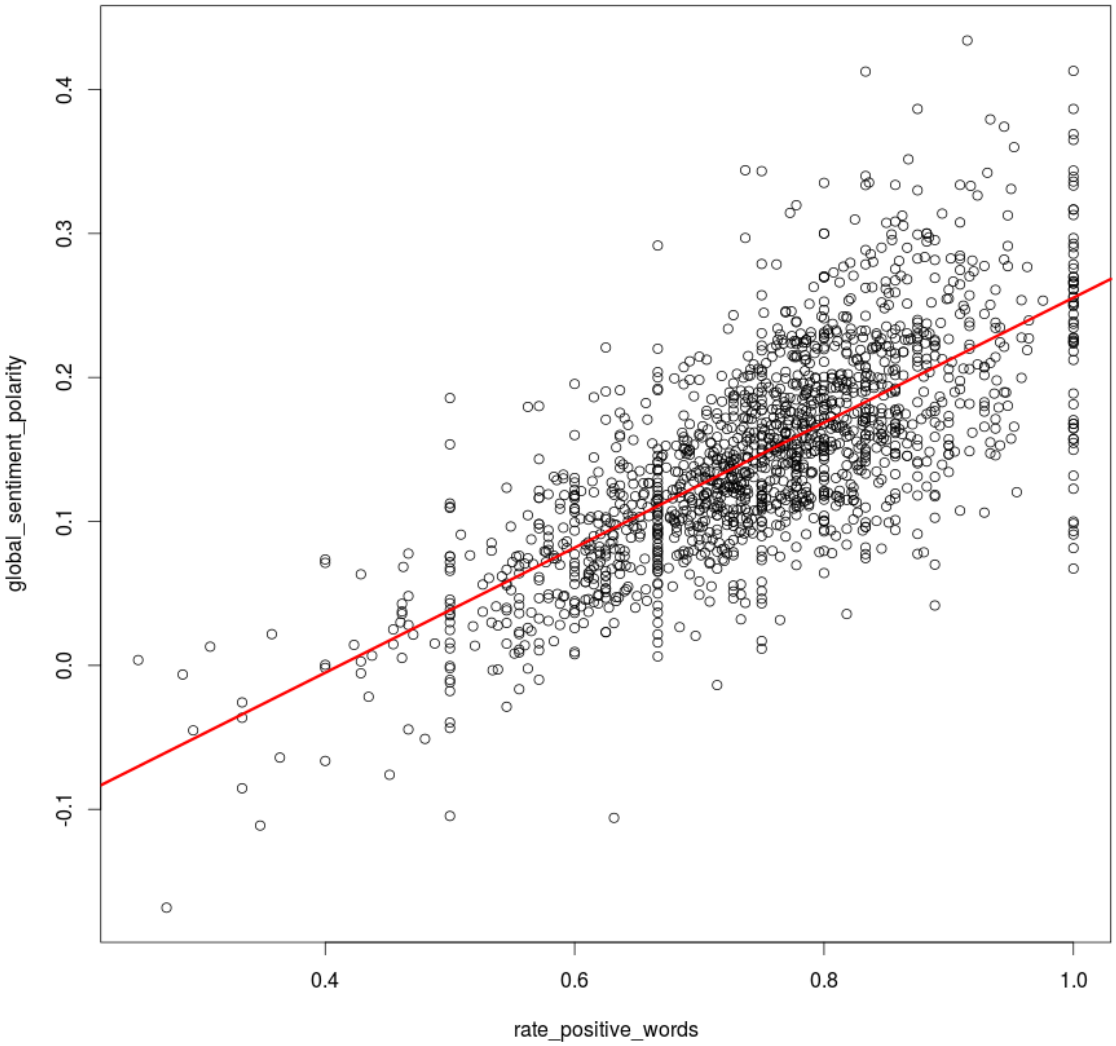


Figure 3

We have performed the Shapiro-Wilk normality test with intercept, slope and correlation coefficient bootstrap distributions. The results shown in the Table **??** are a little bit striking: with good p-value it is trustworthy that intercept and slope boostrap samples are obtained from the normal distribution, but we could not say so about correlation coefficient samples data.

So for the correlation coefficient we compute CI using ranked quntiles. The aforesaid results we obtained are shown in Table **??**. It is worth to mention that the difference in estimating correlation coefficient CI using the assumption of normallity and without such is not very huge, but we think it is a good idea not only test normality of the data by it's visualization but also with statistical tests.

Table 1: Shapiro-Wilk normallity test p-value

	95% p-value
Intercept	0.2279
Slope	0.244
Correlation	5.621e-08

Table 2: 95% confidence intervals (CI's) for intercept, slope, correlation coefficient based on bootrap techinque

	95% CI
Intercept	(-0.209; -0.148)
Slope	(0.392; 0.476)
Correlation (normal)	(0.668; 0.765)
Correlation (percentile)	(0.666, 0.763)

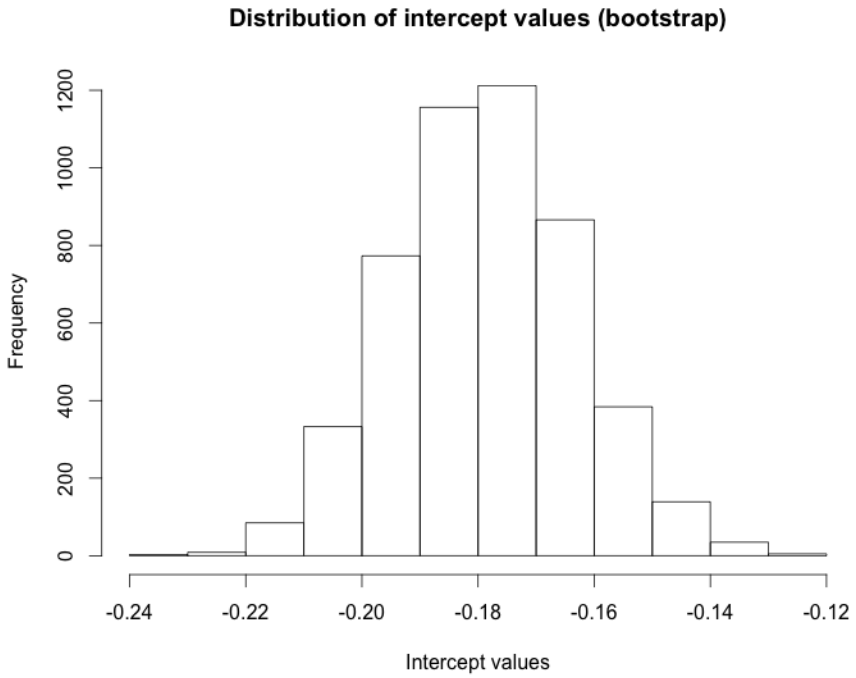


Figure 4: Distribution of intercept value of linear regression models built with each pair of sampled (`global_sentiment_polarity`) and (`rate_positive_words`) using bootrsap

1.5 Average relative error

Recall average relative error (ARE) and coefficient of determination (R^2) definitions:

$$\text{ARE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|,$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

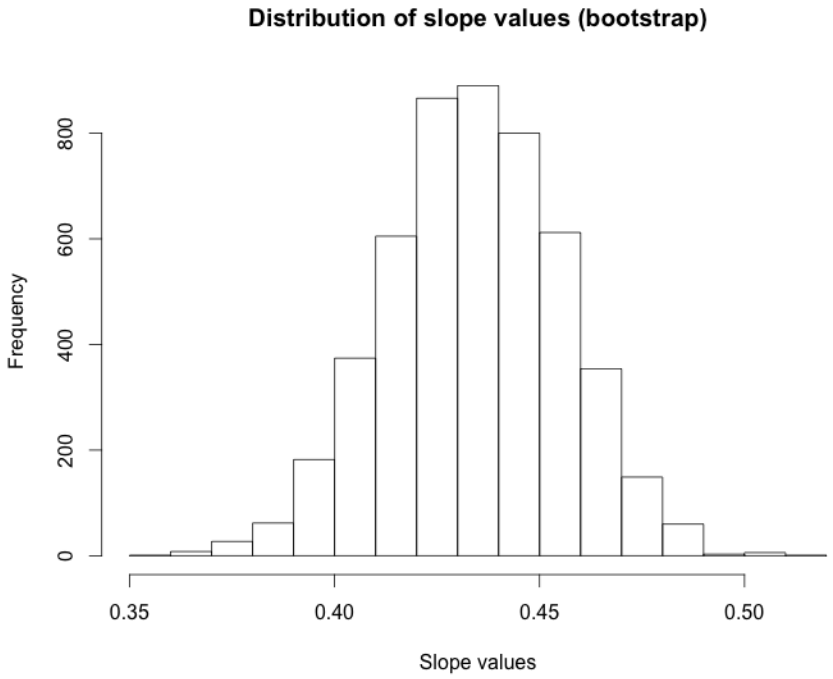


Figure 5: Distribution of slope value of linear regression models built with each pair of sampled (`global_sentiment_polarity`) and (`rate_positive_words`) using bootrsap

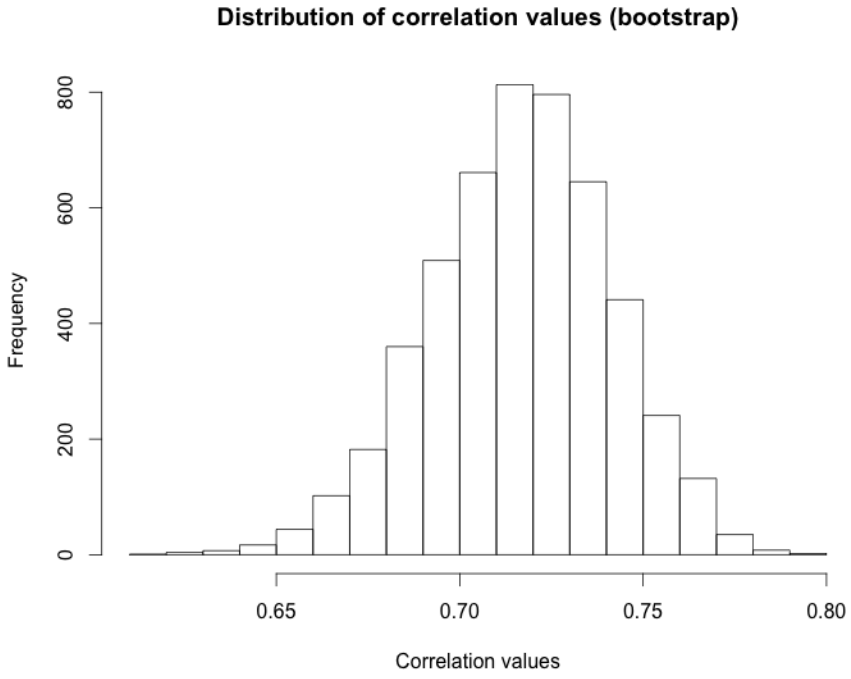


Figure 6: Distribution of correlation values between dependent variable (`global_sentiment_polarity`) and regressor (`rate_positive_words`)

```
mean( abs((y.feature - model$fitted.values)/y.feature) * 100) # in %

## [1] 56.3725

summary_regression$r.squared * 100 # in %

## [1] 51.41945
```

As we can see, considered values are reasonably close. But we should note that ARE is sensitive to the addition of a constant to all of the y_i while R^2 is not. That is, we could obtain any arbitrary value of ARE keeping the coefficient of determination constant by adding some constants to y_i.

It was suprise to us that ARE can be greater than 1 (which is the case if y_i is much less than $y_i - \hat{y}_i$).

According to all these facts one could conclude that comparing ARE and R^2 is meaningless without additional assumptions.

1.6 Nature-inspired algorithm

We will use nature-inspired algorithm to compute the parameters of linear regression that minimize the absolute relative error. If the target values of regression are y_i and predicted by linear regression values are $\hat{y}_i$, the absolute relative error is:

$$\frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{1}$$

We have implemented algorithm similar to the one, which is described in [**?**], but we will minimize the function `delta(coefficients, x, y)`, which looks like

```
delta <- function(coefficients, x,y){
  a = coefficients[1]; b = coefficients[2]
  yp <- a*x + b
  esq <- mean( abs((y - yp)/y) )
}
```

We also need the function that compute permissible limits for coefficients. Let look at two values of target $y_i = a \cdot x_i + b$ and $y_j = a \cdot x_j + b$ $(i \neq j)$ and express a, b in terms of x and y:

$$a_{ij} = \frac{y_j - y_i}{x_j - x_i},\ b_{ij} = \frac{y_i x_j - y_j x_i}{x_j - x_i}.$$

And then we calculate max and min of coefficients a and b among all pairs (x_i, y_i) and (x_j, y_j).

Using the nature-inspired approach (we have implemented it in R-function nlr) we obtained the following values of slope (a) and intercept (b) and value of relative error:

```
# The regression model: y.feature = a * x.feature + b
# Coefficients of slope and intercept respectively:
model.nlr <- nlr(x.feature, y.feature)
model.nlr

## [1]  0.3610875 -0.1570541

# Value of relative error:
eps.nlr <- y.feature - model.nlr[1]*x.feature - model.nlr[2]
mean( abs(eps.nlr / y.feature) ) * 100

## [1] 50.04851
```

Comparing the values of two relative errors, we can see that nature-inspired approach reduce its value by a few percent.

In R-language there is a package **genalg** with function **rbga** that implement this approach. The results obtained with function **rbga** are very close to the results described above:

```
# The regression model: y.feature = a * x.feature + b
# Calculate the permissible limits for a and b
bound <- ddr(x.feature, y.feature)
bounds.min <- c(bound[[1]][1],bound[[2]][1]) # (a.min, b.min)
bounds.max <- c(bound[[1]][2],bound[[2]][2]) # (a.max, b.max)

rbga.res <- rbga(bounds.min, bounds.max, popSize = 30, iters = 5000,
   evalFunc = function(coefs) delta(coefs, x.feature, y.feature))

# Results (we need to take a look at "Best Solution")
cat(summary(rbga.res))

## GA Settings
##   Type                  = floats chromosome
##   Population size       = 30
##   Number of Generations = 5000
##   Elitism               = 6
##   Mutation Chance       = 0.333333333333333
##
## Search Domain
##   Var 1 = [-6.674999999946,6.47188552188199]
##   Var 2 = [-5.27885519411197,5.97499999995278]
##
## GA Results
##   Best Solution : 0.348751377798375 -0.14645601290741

# Value of relative error:
delta(c(0.3477, -0.1463), x.feature, y.feature) * 100

## [1] 49.92909
```