Project work - dati cross section

Kevin Capano 844018, Sara Licaj 846892, Susanna Maugeri 839365

Esame di Statistica Computazionale del 25 novembre 2020

Importazione dei dati

```
file <- read.csv("dataset_finito.csv", sep=",", dec = ".",
            stringsAsFactors=TRUE, na.strings=c("NA","NaN", ""))
```

Presentazione dataset e statistiche descrittive

Il nostro dataset si compone di 39644 osservazioni per 17 variabili.
Ogni osservazione si riferisce ad un articolo di un giornale web, Mashable.
La variabile url è quella identificativa di ogni osservazione.

Le altre variabili sono:

- n_tokens_title: numero di parole nel titolo
- n_tokens_content: numero di parole nell'articolo
- n_unique_tokens: percentuale di parole uniche nell'articolo
- n_non_stop_words: percentuale di non-stop-words nell'articolo
- n_non_stop_unique_tokens: percentuale di parole uniche non-stop-words nell'articolo
- num_hrefs: numero di link
- num_imgs: numero di immagini
- num_videos: numero di video
- average_token_length: lunghezza media delle parole nell'articolo
- num_keywords: numero di keywords nei metadata
- argomento: argomento trattato, è una variabile fattorea 6 livelli
- day: giorno di pubblicazione dell'articolo, è una variabile fattore a 7 livelli
- is_weekend: variabile binaria con 0 se l'articolo è stato pubblicato durante la settimana e 1 se è stato pubblicato durante il weekend
- rate_positive_words: percentuale di parole positive tra i tokens non neutri
- rate_negative_words: percentuale di parole negative tra i tokens non neutri

Queste variabili sono usate per predire la variabile target Shares, che indica il numero di volte che l'articolo che è stato condiviso.

```
summary(file[,-c(1, 16, 17)])
```

```
##  n_tokens_title n_tokens_content n_unique_tokens    n_non_stop_words
##  Min.   : 2.0   Min.   :   0.0   Min.   :  0.0000   Min.   :   0.0000
##  1st Qu.: 9.0   1st Qu.: 246.0   1st Qu.:  0.4709   1st Qu.:   1.0000
##  Median :10.0   Median : 409.0   Median :  0.5392   Median :   1.0000
##  Mean   :10.4   Mean   : 546.5   Mean   :  0.5482   Mean   :   0.9965
##  3rd Qu.:12.0   3rd Qu.: 716.0   3rd Qu.:  0.6087   3rd Qu.:   1.0000
##  Max.   :23.0   Max.   :8474.0   Max.   :701.0000   Max.   :1042.0000
##  n_non_stop_unique_tokens   num_hrefs         num_imgs         num_videos
##  Min.   :  0.0000         Min.   :  0.00   Min.   :  0.000   Min.   : 0.00
```

```
##  1st Qu.:  0.6257         1st Qu.:  4.00   1st Qu.:  1.000   1st Qu.: 0.00
##  Median :  0.6905         Median :  8.00   Median :  1.000   Median : 0.00
##  Mean   :  0.6892         Mean   : 10.88   Mean   :  4.544   Mean   : 1.25
##  3rd Qu.:  0.7546         3rd Qu.: 14.00   3rd Qu.:  4.000   3rd Qu.: 1.00
##  Max.   :650.0000         Max.   :304.00   Max.   :128.000   Max.   :91.00
##  average_token_length  num_keywords      is_weekend     rate_positive_words
##  Min.   :0.000        Min.   : 1.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:4.478        1st Qu.: 6.000   1st Qu.:0.0000   1st Qu.:0.6000
##  Median :4.664        Median : 7.000   Median :0.0000   Median :0.7105
##  Mean   :4.548        Mean   : 7.224   Mean   :0.1309   Mean   :0.6822
##  3rd Qu.:4.855        3rd Qu.: 9.000   3rd Qu.:0.0000   3rd Qu.:0.8000
##  Max.   :8.042        Max.   :10.000   Max.   :1.0000   Max.   :1.0000
##  rate_negative_words     shares
##  Min.   :0.0000      Min.   :     1
##  1st Qu.:0.1852      1st Qu.:   946
##  Median :0.2800      Median :  1400
##  Mean   :0.2879      Mean   :  3395
##  3rd Qu.:0.3846      3rd Qu.:  2800
##  Max.   :1.0000      Max.   :843300

table(file[, 16])

##
##  business entertain lifestyle social me technolog     world
##      6258      7057      2099      2323      7346      8427

table(file[, 17])

##
## friday monday saturd sunday thursd tuesda wednes
##   5701   6661   2453   2737   7267   7390   7435
```

Si nota che nessuna delle variabili presenta valori negativi. Per quanto riguarda la variabile target "shares", si nota che i valori sono compresi tra 1 e 843300 e che in media ogni articolo viene condiviso 3395 volte. Si tratta di una distribuzione molto asimmetrica con coda a destra. L'argomento e il giorno di pubblicazione più frequenti per gli articoli sono World e il mercoledì.

Missing data

Conteggio

```
sapply(file, function(x)(sum(is.na(x))))

##                       url           n_tokens_title         n_tokens_content
##                         0                        0                        0
##          n_unique_tokens         n_non_stop_words n_non_stop_unique_tokens
##                         0                        0                        0
##                 num_hrefs                 num_imgs               num_videos
##                         0                        0                        0
##     average_token_length             num_keywords               is_weekend
##                         0                        0                        0
##      rate_positive_words      rate_negative_words                   shares
```

```
##                         0                         0                         0
##                 argomento                       day
##                      6134                         0
```

Il primo controllo da effettuare sul nostro modello è la presenza o meno di dati mancanti: l'unica variabile che presenta missing values è "argomento", per un totale di 6134 unità.

Per scrivere più agevolmente le variabili esplicative del modello:

```
formula <- paste(colnames(file), collapse="+")
formula
```

```
## [1] "url+n_tokens_title+n_tokens_content+n_unique_tokens+n_non_stop_words+n_non
_stop_unique_tokens+num_hrefs+num_imgs+num_videos+average_token_length+num_keyword
s+is_weekend+rate_positive_words+rate_negative_words+shares+argomento+day"
```

```
modello_base <- lm(shares ~ n_tokens_title + n_tokens_content + n_unique_tokens +
n_non_stop_words + n_non_stop_unique_tokens + num_hrefs + num_imgs +num_videos +
average_token_length + num_keywords + is_weekend +rate_positive_words +
rate_negative_words + argomento + day, data=file)
summary(modello_base)
```

```
##
## Call:
## lm(formula = shares ~ n_tokens_title + n_tokens_content + n_unique_tokens +
##     n_non_stop_words + n_non_stop_unique_tokens + num_hrefs +
##     num_imgs + num_videos + average_token_length + num_keywords +
##     is_weekend + rate_positive_words + rate_negative_words +
##     argomento + day, data = file)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -7737  -1935  -1279   -295 686263
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2547.7244   556.9215   4.575 4.79e-06 ***
## n_tokens_title               35.3162    24.8009   1.424 0.154460
## n_tokens_content              0.5361     0.1905   2.814 0.004893 **
## n_unique_tokens            4219.7117  1645.1534   2.565 0.010324 *
## n_non_stop_words          -2142.8229   566.0717  -3.785 0.000154 ***
## n_non_stop_unique_tokens -1106.4410  1450.4146  -0.763 0.445561
## num_hrefs                    30.6826     5.8481   5.247 1.56e-07 ***
## num_imgs                     19.9146     8.3407   2.388 0.016962 *
## num_videos                   45.5286    14.9446   3.046 0.002317 **
## average_token_length      -1131.1129   215.2787  -5.254 1.50e-07 ***
## num_keywords                106.2591    27.9296   3.805 0.000142 ***
## is_weekend                  649.3626   244.9057   2.651 0.008018 **
## rate_positive_words        4501.2954  1180.2137   3.814 0.000137 ***
## rate_negative_words        4946.3048  1214.0914   4.074 4.63e-05 ***
## argomentoentertain         -622.5038   174.6292  -3.565 0.000365 ***
## argomentolifestyle          156.0376   245.5851   0.635 0.525191
```

```
## argomentosocial me         314.4945   232.0783   1.355 0.175388
## argomentotechnolog        -277.7557   169.4597  -1.639 0.101209
## argomentoworld            -837.4761   166.0352  -5.044 4.58e-07 ***
## daymonday                  349.3676   185.3691   1.885 0.059477 .
## daysaturd                  -72.8297   291.4559  -0.250 0.802680
## daysunday                       NA         NA      NA       NA
## daythursd                   22.1525   182.5479   0.121 0.903413
## daytuesda                   26.9721   181.8676   0.148 0.882102
## daywednes                   11.3309   181.4125   0.062 0.950197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9442 on 33486 degrees of freedom
##   (6134 observations deleted due to missingness)
## Multiple R-squared:  0.006669,   Adjusted R-squared:  0.005987
## F-statistic: 9.775 on 23 and 33486 DF,  p-value: < 2.2e-16
```

Il modello appena creato è il punto di partenza del percorso di costruzione di un modello robusto, si tratta di una relazione lineare dove "shares" rappresenta la variabile target e la variabile "url" è stata esclusa dall'insieme delle esplicative, in quanto è l'identificativo delle osservazioni. Nell'output del summary si può notare che per la modalità "Sunday" della variabile "Day" non è stato calcolato nessun parametro, inoltre i residui non sono simmetrici intorno al valore nullo e l'indice R^2 è prossimo allo zero. Più della metà delle variabili risultano significative.

Imputazione

Utilizziamo la seguente lista per scrivere rapidamente l'insieme di covariate inserite nel modello, ad eccezione di "url" e "shares":

```
lista <- paste(colnames(file), collapse=",")
lista

## [1] "url,n_tokens_title,n_tokens_content,n_unique_tokens,n_non_stop_words,n_non
_stop_unique_tokens,num_hrefs,num_imgs,num_videos,average_token_length,num_keyword
s,is_weekend,rate_positive_words,rate_negative_words,shares,argomento,day"

covariate <- file[,c("n_tokens_title", "n_tokens_content", "n_unique_tokens", "n_n
on_stop_words","n_non_stop_unique_tokens", "num_hrefs", "num_imgs", "num_videos",
"average_token_length", "num_keywords", "is_weekend", "rate_positive_words","rate_
negative_words", "argomento", "day")]
```

Il modello di partenza include l'unica variabile che presenta dei missing values ("argomento"), di conseguenza è necessaria una procedura di imputazione per sostituire i dati mancanti, al fine di evitare che il modello venga eseguito solamente su un sottoinsieme di osservazioni. In particolare, utilizziamo il pacchetto mice per eseguire una multiple imputation e, poichè la variabile che presenta dati mancanti è categoriale, i missing values verranno imputati tramite un modello logistico.

```
library(mice)

tempData <- mice(covariate, m=1, maxit=20, meth='pmm', seed=500)
```

```
data_imputed <- complete(tempData,1)
names(data_imputed)
```

```
##  [1] "n_tokens_title"           "n_tokens_content"
##  [3] "n_unique_tokens"          "n_non_stop_words"
##  [5] "n_non_stop_unique_tokens" "num_hrefs"
##  [7] "num_imgs"                 "num_videos"
##  [9] "average_token_length"     "num_keywords"
## [11] "is_weekend"               "rate_positive_words"
## [13] "rate_negative_words"      "argomento"
## [15] "day"
```

```
sapply(data_imputed, function(x)(sum(is.na(x))))
```

```
##           n_tokens_title         n_tokens_content          n_unique_tokens
##                        0                        0                        0
##         n_non_stop_words n_non_stop_unique_tokens                num_hrefs
##                        0                        0                        0
##                 num_imgs               num_videos     average_token_length
##                        0                        0                        0
##             num_keywords               is_weekend      rate_positive_words
##                        0                        0                        0
##      rate_negative_words                argomento                      day
##                        0                        0                        0
```

La procedura di imputazione è stata efficace e tutti i dati mancanti sono stati imputati correttamente.

```
dati_completi=cbind(data_imputed, file$shares)
names(dati_completi)
```

```
##  [1] "n_tokens_title"           "n_tokens_content"
##  [3] "n_unique_tokens"          "n_non_stop_words"
##  [5] "n_non_stop_unique_tokens" "num_hrefs"
##  [7] "num_imgs"                 "num_videos"
##  [9] "average_token_length"     "num_keywords"
## [11] "is_weekend"               "rate_positive_words"
## [13] "rate_negative_words"      "argomento"
## [15] "day"                      "file$shares"
```

```
names(dati_completi)[16] <- "shares"

covariate_giuste <- dati_completi[,c("n_tokens_title", "n_tokens_content", "n_uniq
ue_tokens","n_non_stop_words", "n_non_stop_unique_tokens", "num_hrefs","num_imgs"
, "num_videos", "average_token_length", "num_keywords", "is_weekend", "rate_positi
ve_words", "rate_negative_words","argomento", "day")]
```

Modello completo su dati completi

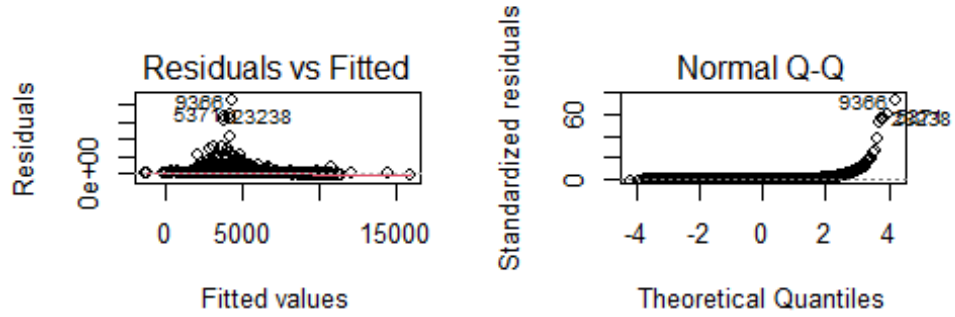
```
modello_base_completo <- lm(shares ~ n_tokens_title + n_tokens_content + n_unique_
tokens + n_non_stop_words + n_non_stop_unique_tokens + num_hrefs + num_imgs + num_v
ideos + average_token_length + num_keywords + is_weekend + rate_positive_words + ra
te_negative_words + argomento + day, data=dati_completi)
summary(modello_base_completo)
```

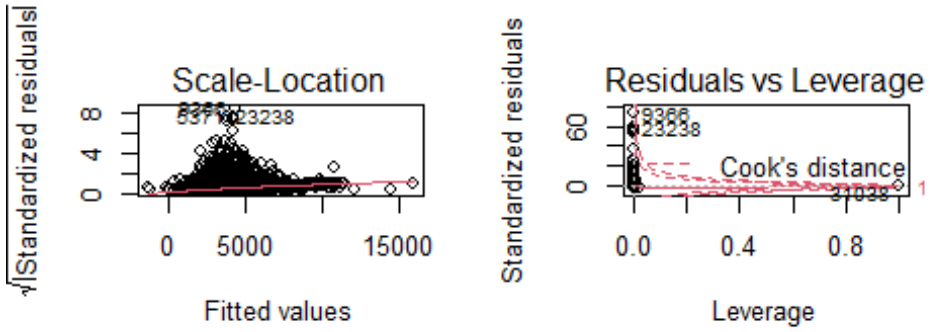
```
##
## Call:
## lm(formula = shares ~ n_tokens_title + n_tokens_content + n_unique_tokens +
##     n_non_stop_words + n_non_stop_unique_tokens + num_hrefs +
##     num_imgs + num_videos + average_token_length + num_keywords +
##     is_weekend + rate_positive_words + rate_negative_words +
##     argomento + day, data = dati_completi)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -10785  -2382  -1569   -446 838993
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3372.1416   539.9447   6.245 4.27e-10 ***
## n_tokens_title               65.6533    28.1255   2.334 0.019585 *
## n_tokens_content              0.1664     0.2146   0.775 0.438088
## n_unique_tokens            9555.7752  1717.7743   5.563 2.67e-08 ***
## n_non_stop_words          -2966.1351   602.2693  -4.925 8.47e-07 ***
## n_non_stop_unique_tokens  -5544.5218  1529.5128  -3.625 0.000289 ***
## num_hrefs                    46.9844     6.2040   7.573 3.72e-14 ***
## num_imgs                     40.5051     8.7044   4.653 3.28e-06 ***
## num_videos                   53.1003    15.0345   3.532 0.000413 ***
## average_token_length      -1029.8220   233.7678  -4.405 1.06e-05 ***
## num_keywords                 93.3572    31.9700   2.920 0.003501 **
## is_weekend                  324.8777   270.3966   1.201 0.229570
## rate_positive_words        4145.9389  1246.3158   3.327 0.000880 ***
## rate_negative_words        4752.8693  1280.3607   3.712 0.000206 ***
## argomentoentertain         -585.1255   191.8878  -3.049 0.002295 **
## argomentolifestyle          131.8514   276.0499   0.478 0.632912
## argomentosocial me          147.8167   259.4514   0.570 0.568866
## argomentotechnolog          -69.4462   188.7730  -0.368 0.712963
## argomentoworld             -609.5682   186.5685  -3.267 0.001087 **
## daymonday                   377.3670   209.2721   1.803 0.071359 .
## daysaturd                   346.9001   322.7628   1.075 0.282478
## daysunday                         NA         NA      NA       NA
## daythursd                   -78.6664   205.1429  -0.383 0.701373
## daytuesda                   -78.5809   204.4425  -0.384 0.700709
## daywednes                    70.7257   204.1938   0.346 0.729070
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11590 on 39620 degrees of freedom
## Multiple R-squared:  0.007107,   Adjusted R-squared:  0.00653
## F-statistic: 12.33 on 23 and 39620 DF,  p-value: < 2.2e-16
```

Per il modello adattato su dati non missing valgono le stesse considerazioni fatta in precedenza.

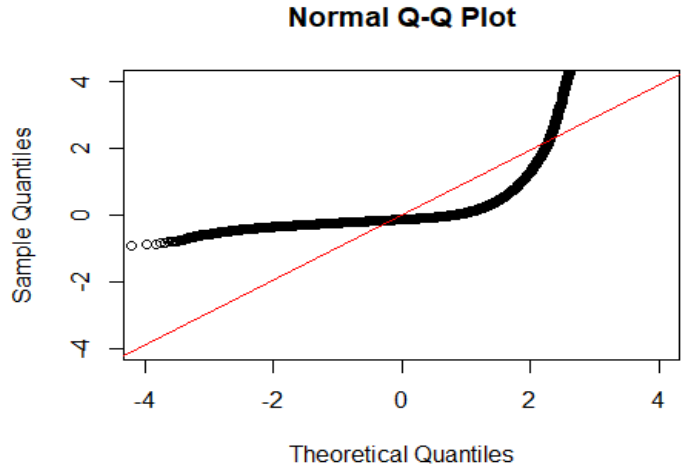
Diagnostiche dei residui del modello completo

```
par(mfrow=c(2,2))
plot(modello_base_completo)
```






```
par(mfrow=c(1,1))
resstand <- rstandard(modello_base_completo)
qqnorm(resstand, xlim=c(-4, 4), ylim=c(-4, 4))
x <- rnorm(1000)
qqline(x, col='red')
```



Dalle diagnostiche dei residui è possibile ipotizzare che lo standard error non sia robusto e che quindi il problema NaN verrà risolto applicando gli Standard Error robusti di White.

Il grafico "Residual vs Fitted" suggerisce che non vi siano pattern non lineari. A prima vista dal "Normal Q-Q" si può pensare che i residui abbiamo un andamento normale ad eccezione della coda di destra, in realtà con un plot più preciso si nota che l'andamento non è assolutamente normale.

Il grafico "Scale Location" mostra che i residui non sono posizionati in modo casuale rispetto ai fitted values, ma che quelli per i valori dei fiddet values tra 2500 e 5000 hanno valori standardizzati più elevati.

Dal grafico "Residual vs Leverage" si nota che vi sono alcune osservazioni che si discostano dal gruppo delle altre per influenza, per esempio la 31038.

Multicollinearità

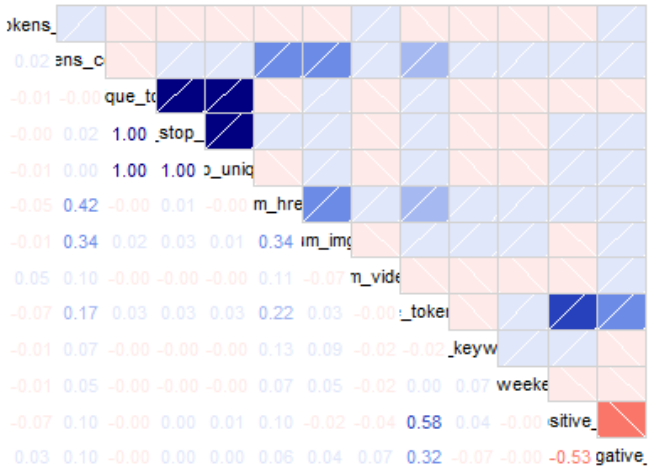
Covariate numeriche:

```
library(plyr)
library(dplyr)

file_numeric <- covariate_giuste %>% dplyr::select_if(is.numeric)
colnames(file_numeric)
```

```
##  [1] "n_tokens_title"           "n_tokens_content"
##  [3] "n_unique_tokens"          "n_non_stop_words"
##  [5] "n_non_stop_unique_tokens" "num_hrefs"
##  [7] "num_imgs"                 "num_videos"
##  [9] "average_token_length"     "num_keywords"
## [11] "is_weekend"               "rate_positive_words"
## [13] "rate_negative_words"
```

Matrice di correlazione tra le covariate:

```
require(corrgram)

corrgram(file_numeric, lower.panel = panel.cor, cex=1, cex.labels = 1)
```



Le variabili "n_unique_tokens", "n_non_stop_words" e "n_non_stop_unique_tokens" sono perfettamente collineari tra loro, pertanto sarà necessario eliminare almeno una di queste esplicative. La strategia ottimale è quella di rimuovere una covariata per volta, in base al valore degli indici Tol e VIF calcolati di seguito.

```
library(mctest)
imcdiag(modello_base)

##
## Call:
## imcdiag(mod = modello_base)
##
## All Individual Multicollinearity Diagnostics Result
##
##                                 VIF    TOL
## n_tokens_title               1.0534 0.9494
## n_tokens_content             3.1930 0.3132
## n_unique_tokens          14909.3130 0.0001
## n_non_stop_words          3896.8794 0.0003
## n_non_stop_unique_tokens  9961.6330 0.0001
## num_hrefs                    1.4443 0.6924
## num_imgs                     1.5607 0.6407
## num_videos                   1.1195 0.8932
## average_token_length         7.2851 0.1373
## num_keywords                 1.1176 0.8948
## is_weekend                      Inf 0.0000
## rate_positive_words         15.3201 0.0653
## rate_negative_words         12.5660 0.0796
## argomentoentertain           1.9055 0.5248
## argomentolifestyle           1.3310 0.7513
## argomentosocial me           1.3061 0.7656
## argomentotechnolog           1.8475 0.5413
## argomentoworld               1.9505 0.5127
## daymonday                    1.8387 0.5439
## daysaturd                       Inf 0.0000
## daysunday                       Inf 0.0000
## daythursd                    1.8804 0.5318
## daytuesda                    1.8930 0.5283
## daywednes                    1.9003 0.5262
##
## R-square of y on all x: 0.0067
## ================================
```

Le soglie di riferimento suggeriscono di rimuovere dal modello le variabili con indice Tol minore di 0.3 e VIF maggiore di 5. Il criterio ideale è quello di eliminare inizialmente la variabile con indice Tol inferiore: in questo caso si tratta di "is_weekend", una dummy con modalità 0=weekday-1=weekend, collineare alla variabile "Day".

```
modello_base1 <- lm(shares ~ n_tokens_title + n_tokens_content + n_unique_tokens +
n_non_stop_words + n_non_stop_unique_tokens + num_hrefs +num_imgs + num_videos +
average_token_length + num_keywords +rate_positive_words + rate_negative_words +
argomento + day,data=dati_completi)
imcdiag(modello_base1)

##
## Call:
## imcdiag(mod = modello_base1)
```

```
##
## All Individual Multicollinearity Diagnostics Result
##
##                                VIF    TOL
## n_tokens_title               1.0435 0.9583
## n_tokens_content             3.0173 0.3314
## n_unique_tokens          10796.2399 0.0001
## n_non_stop_words          2930.0129 0.0003
## n_non_stop_unique_tokens  7360.4436 0.0001
## num_hrefs                    1.4590 0.6854
## num_imgs                     1.5442 0.6476
## num_videos                   1.1259 0.8882
## average_token_length        11.5014 0.0869
## num_keywords                 1.0996 0.9094
## rate_positive_words         16.5877 0.0603
## rate_negative_words         11.7994 0.0847
## argomentoentertain           1.8232 0.5485
## argomentolifestyle           1.3037 0.7670
## argomentosocial me           1.2996 0.7695
## argomentotechnolog           1.7969 0.5565
## argomentoworld               1.9183 0.5213
## daymonday                    1.8071 0.5534
## daysaturd                    1.3486 0.7415
## daysunday                    1.3871 0.7209
## daythursd                    1.8597 0.5377
## daytuesda                    1.8711 0.5344
## daywednes                    1.8753 0.5332
##
## R-square of y on all x: 0.0071
## ================================
```

La seconda esplicativa che deve essere eliminata è "n_unique_tokens", che presenta un Tol pari a 0.0001, il minore in assoluto.

```
modello_base2 <- lm(shares ~ n_tokens_title + n_tokens_content + n_non_stop_words
+ n_non_stop_unique_tokens + num_hrefs + num_imgs + num_videos + average_token_leng
th + num_keywords + rate_positive_words +rate_negative_words + argomento + day,
data=dati_completi)
imcdiag(modello_base2)
```

```
##
## Call:
## imcdiag(mod = modello_base2)
##
## All Individual Multicollinearity Diagnostics Result
##
##                                VIF    TOL
## n_tokens_title               1.0435 0.9583
## n_tokens_content             1.8410 0.5432
## n_non_stop_words          1993.1693 0.0005
## n_non_stop_unique_tokens  1991.3902 0.0005
## num_hrefs                    1.4432 0.6929
```

```
## num_imgs                    1.4209 0.7038
## num_videos                  1.0983 0.9105
## average_token_length       10.8275 0.0924
## num_keywords                1.0995 0.9095
## rate_positive_words        13.8932 0.0720
## rate_negative_words         9.8807 0.1012
## argomentoentertain          1.8113 0.5521
## argomentolifestyle          1.3030 0.7675
## argomentosocial me          1.2982 0.7703
## argomentotechnolog          1.7963 0.5567
## argomentoworld              1.9149 0.5222
## daymonday                   1.8071 0.5534
## daysaturd                   1.3485 0.7416
## daysunday                   1.3865 0.7212
## daythursd                   1.8597 0.5377
## daytuesda                   1.8711 0.5344
## daywednes                   1.8753 0.5333
##
## R-square of y on all x: 0.0063
## ================================
```

A questo punto le variabili "n_non_stop_words" e "n_non_stop_unique_tokens" presentano uguali valori di Tol, perciò si decide quale eliminare in base al VIF superiore: la scelta, quindi, ricade su "n_non_stop_words".

```
modello_base3 <- lm(shares ~ n_tokens_title + n_tokens_content + n_non_stop_unique
_tokens +num_hrefs + num_imgs + num_videos + average_token_length +num_keywords +
rate_positive_words + rate_negative_words +argomento + day, data=dati_completi)
imcdiag(modello_base3)
```

```
##
## Call:
## imcdiag(mod = modello_base3)
##
## All Individual Multicollinearity Diagnostics Result
##
##                              VIF    TOL
## n_tokens_title            1.0427 0.9591
## n_tokens_content          1.3913 0.7188
## n_non_stop_unique_tokens  1.0089 0.9912
## num_hrefs                 1.4339 0.6974
## num_imgs                  1.2822 0.7799
## num_videos                1.0975 0.9112
## average_token_length     10.7320 0.0932
## num_keywords              1.0991 0.9098
## rate_positive_words      13.4380 0.0744
## rate_negative_words       9.5654 0.1045
## argomentoentertain        1.8104 0.5524
## argomentolifestyle        1.3028 0.7676
## argomentosocial me        1.2979 0.7705
## argomentotechnolog        1.7932 0.5576
## argomentoworld            1.9141 0.5225
```

```
## daymonday                 1.8068 0.5535
## daysaturd                 1.3483 0.7417
## daysunday                 1.3865 0.7212
## daythursd                 1.8594 0.5378
## daytuesda                 1.8708 0.5345
## daywednes                 1.8750 0.5333
##
##
## R-square of y on all x: 0.0062
##
## ================================
```

Osservando ulteriormente i dati è possibile notare che "rate_positive_words" e "rate_negative_words" sono complemetari a 1, perciò eliminiamo quella con Tol minore. Poichè "rate_positive_words" è correlata positivamente con "average_token_length", ci si aspetta che anche il valore di VIF di questa variabile si normalizzi.

```
modello_base4 <- lm(shares ~ n_tokens_title + n_tokens_content + n_non_stop_unique
_tokens + num_hrefs + num_imgs + num_videos + average_token_length + num_keywords
+  rate_negative_words + argomento + day, data=dati_completi)
imcdiag(modello_base4)

##
## Call:
## imcdiag(mod = modello_base4)
##
## All Individual Multicollinearity Diagnostics Result
##
##                             VIF    TOL
## n_tokens_title           1.0379 0.9635
## n_tokens_content         1.3198 0.7577
## n_non_stop_unique_tokens 1.0021 0.9979
## num_hrefs                1.3849 0.7221
## num_imgs                 1.2688 0.7882
## num_videos               1.0971 0.9115
## average_token_length     1.2067 0.8287
## num_keywords             1.0991 0.9099
## rate_negative_words      1.2005 0.8330
## argomentoentertain       1.7993 0.5558
## argomentolifestyle       1.2991 0.7698
## argomentosocial me       1.2937 0.7730
## argomentotechnolog       1.7714 0.5645
## argomentoworld           1.8541 0.5393
## daymonday                1.8068 0.5535
## daysaturd                1.3482 0.7417
## daysunday                1.3860 0.7215
## daythursd                1.8592 0.5379
## daytuesda                1.8708 0.5345
## daywednes                1.8749 0.5334
##
## R-square of y on all x: 0.0062
```

```
##
## ================================
```

Il modello ottenuto finora sembra sufficientemente accettabile, pertanto è possibile proseguire con gli step successivi.

Covariate fattore:

Verifichiamo l'eventuale presenza di collinearità per le covariate di tipo factor:

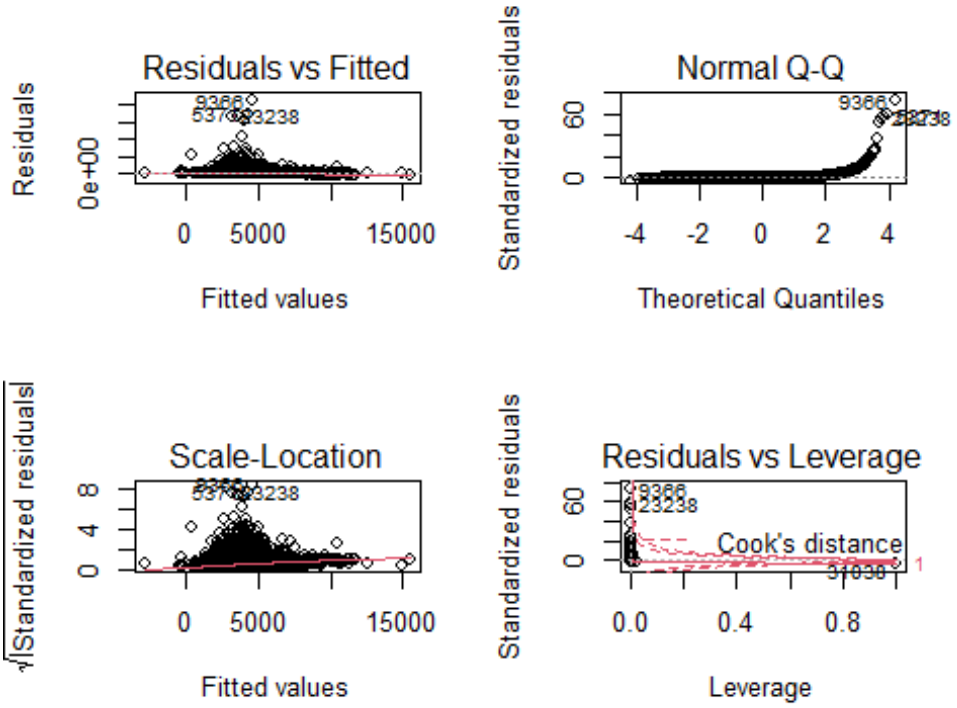
```
file_fac <- covariate_giuste %>% dplyr::select_if(is.factor)
combos <- combn(ncol(file_fac),2)
adply(combos, 2, function(x) {
  test <- chisq.test(file_fac[, x[1]], file_fac[, x[2]])
  tab  <- table(file_fac[, x[1]], file_fac[, x[2]])
  out <- data.frame("Row" = colnames(file_fac)[x[1]]
                    , "Column" = colnames(file_fac[x[2]])
                    , "Chi.Square" = round(test$statistic,3)
                    , "df"= test$parameter
                    , "p.value" = round(test$p.value, 3)
                    , "n" = sum(table(file_fac[,x[1]], file_fac[,x[2]]))
                    , "Chi.Square norm"  =test$statistic/(sum(table(file_fac[,x[1]
], file_fac[,x[2]]))*
                      min(length(unique(file_fac[,x[1]]))-1 , length(unique(file_f
ac[,x[2]]))-1))
  )


  return(out)

})
```

```
##   X1       Row Column Chi.Square df p.value     n Chi.Square.norm
## 1  1 argomento    day    271.051 30       0 39644  0.001367426
```

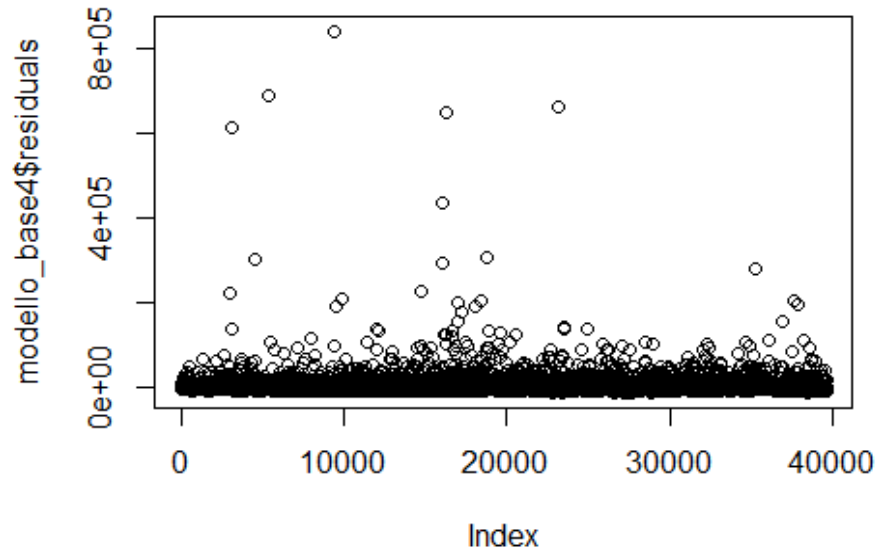
Il valore del Chi-quadrato normalizzato è 0.00137, essendo molto basso e non superiore a 0.9, si tratta di una quantità che non suggerisce la necessità di eliminare ulteriori covariate dal modello.

Diagnostiche dei residui del modello senza collinearità

```
par(mfrow=c(2,2))
plot(modello_base4)
```

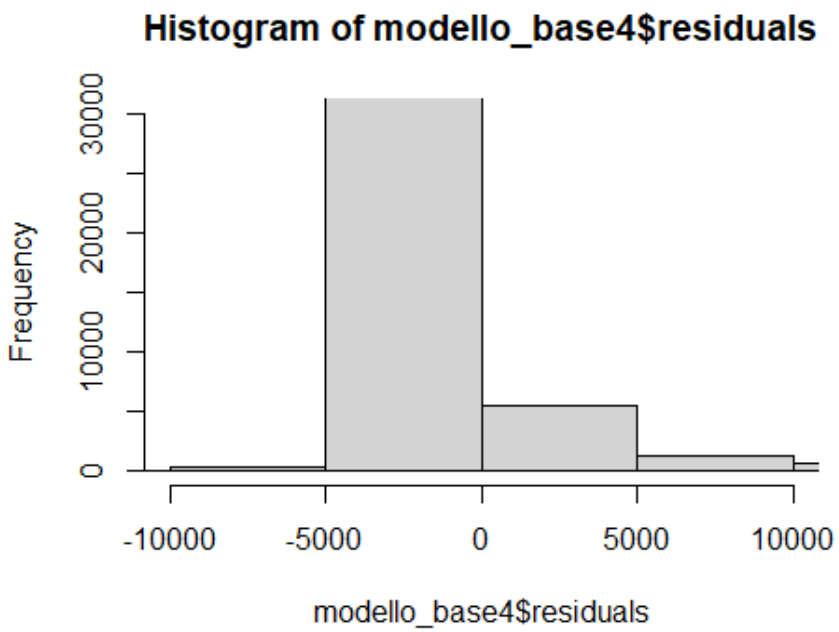



```
par(mfrow=c(1,1))
plot(modello_base4$residuals)
```




```
max(modello_base4$residuals)
## [1] 838720.7
min(modello_base4$residuals)
## [1] -10998.03
```

```r
hist(modello_base4$residuals, breaks=200, xlim=c(-10000, 10000), ylim=c(0, 30000))
```



Dalle diagnostiche si osserva che i residui sono quasi tutti compresi tra -5000 e 0, inoltre vi sono chiari segni di eteroschedasticità, ovvero la varianza non è costante per tutte le osservazioni. Eliminando la collinearità tra le variabili, i plot di diagnostiche non sembrano subire sostanziali miglioramenti.

Test per l'eteroschedasticità del modello senza collinearità

Per confermare l'ipotesi di eteroschedasticità emersa dalle diagnostiche dei residui, applichiamo il test di Breusch-Pagan, ricordando che il rifiuto dell'ipotesi nulla mostra una situazione di non omoschedasticità.

```r
library(lmtest)

bptest(modello_base4)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modello_base4
## BP = 18.837, df = 20, p-value = 0.5325
```

Il risultato del test non fornisce evidenza di eteroschedasticità, tuttavia potrebbero esserci dei punti influenti da eliminare e questo aspetto verrà valutato successivamente. Un altro test utile, oltre che più restrittivo rispetto a Breusch-Pagan, per verificare la presenza di eteroschedasticità è il test di White:

```r
library(car)

ncvTest(modello_base4)
```

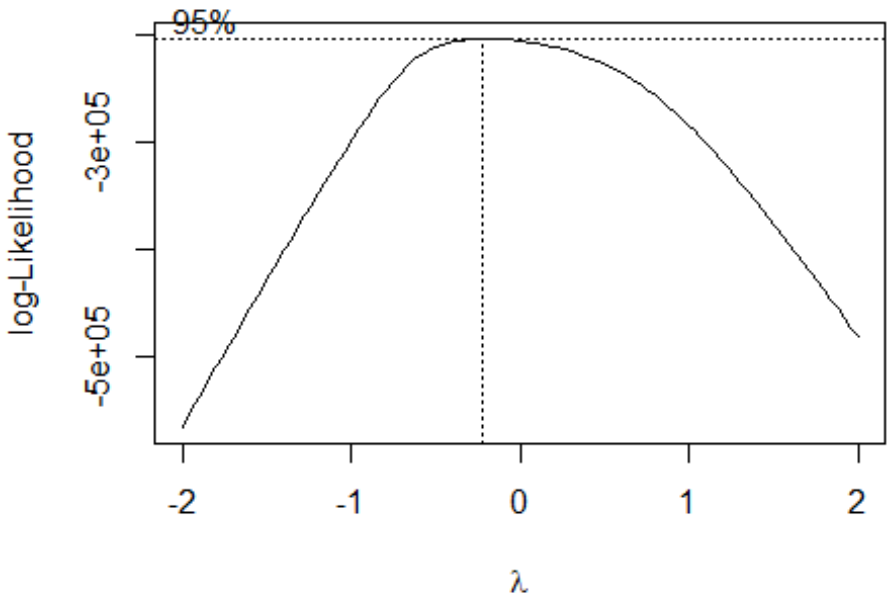
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 5814.333, Df = 1, p = < 2.22e-16
```

Questa volta l'ipotesi nulla di omoschedasticità viene rifiutata, quindi significa che c'è almeno una variabile che è responsabile di eteroschedasticità. In concordanza con quanto osservato dalle diagnostiche dei residui, si considera maggiormente valido il risultato del test di White e andiamo ad applicare diverse strategie per porre rimedio alla violazione di questo assunto molto importante.

Linearità

Trasformazione ottimale del target con Box-Cox

```
library(MASS)

boxcoxreg1<-boxcox(modello_base4)
```




```
lambda=boxcoxreg1$x[which.max(boxcoxreg1$y)]
lambda

## [1] -0.2222222
```

Il valore di lambda ottenuto è -0.2222222: l'approssimazione migliore è per lambda=0 e ciò corrisponde ad una trasformazione logaritmica della variabile target.

```
modello_base5 <- lm(log(shares+1) ~ n_tokens_title + n_tokens_content + n_non_stop
_unique_tokens +num_hrefs + num_imgs + num_videos + average_token_length + num_key
```

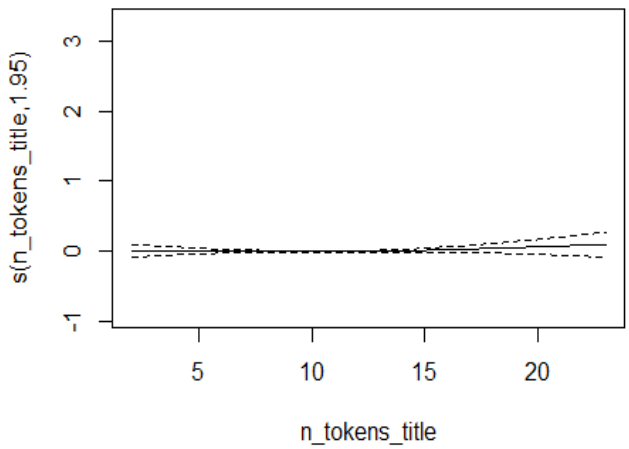
```
words + rate_negative_words + argomento + day, data=dati_completi)
summary(modello_base5)

##
## Call:
## lm(formula = log(shares + 1) ~ n_tokens_title + n_tokens_content +
##     n_non_stop_unique_tokens + num_hrefs + num_imgs + num_videos +
##     average_token_length + num_keywords + rate_negative_words +
##     argomento + day, data = dati_completi)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.117 -0.581 -0.198  0.415  5.943
##
## Coefficients:
##                              Estimate Std. Error t value             Pr(>|t|)
## (Intercept)                 7.6036324  0.0413838  183.73 < 0.0000000000000002 ***
## n_tokens_title              0.0011653  0.0021829    0.53               0.5935
## n_tokens_content           -0.0000721  0.0000110   -6.53   0.0000000000663252 ***
## n_non_stop_unique_tokens    0.0017690  0.0013889    1.27               0.2028
## num_hrefs                   0.0073218  0.0004704   15.56 < 0.0000000000000002 ***
## num_imgs                    0.0075977  0.0006141   12.37 < 0.0000000000000002 ***
## num_videos                  0.0089171  0.0011550    7.72   0.0000000000000119 ***
## average_token_length       -0.0678033  0.0058929  -11.51 < 0.0000000000000002 ***
## num_keywords                0.0198486  0.0024875    7.98   0.0000000000000015 ***
## rate_negative_words        -0.0172096  0.0317843   -0.54               0.5882
## argomentoentertain         -0.1502126  0.0148358  -10.13 < 0.0000000000000002 ***
## argomentolifestyle          0.0620470  0.0214454    2.89               0.0038 **
## argomentosocial me          0.2551699  0.0201462   12.67 < 0.0000000000000002 ***
## argomentotechnolog          0.1016504  0.0145866    6.97   0.0000000000032479 ***
## argomentoworld             -0.2115346  0.0142749  -14.82 < 0.0000000000000002 ***
## daymonday                  -0.0285792  0.0162853   -1.75               0.0793 .
## daysaturd                   0.2336324  0.0218307   10.70 < 0.0000000000000002 ***
## daysunday                   0.2320524  0.0210352   11.03 < 0.0000000000000002 ***
## daythursd                  -0.0686589  0.0159635   -4.30   0.0000170441843328 ***
## daytuesda                  -0.0791891  0.0159095   -4.98   0.0000006468496512 ***
## daywednes                  -0.0800651  0.0158896   -5.04   0.0000004704462842 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.902 on 39623 degrees of freedom
## Multiple R-squared:  0.0593, Adjusted R-squared:  0.0588
## F-statistic:  125 on 20 and 39623 DF,  p-value: <0.0000000000000002
```

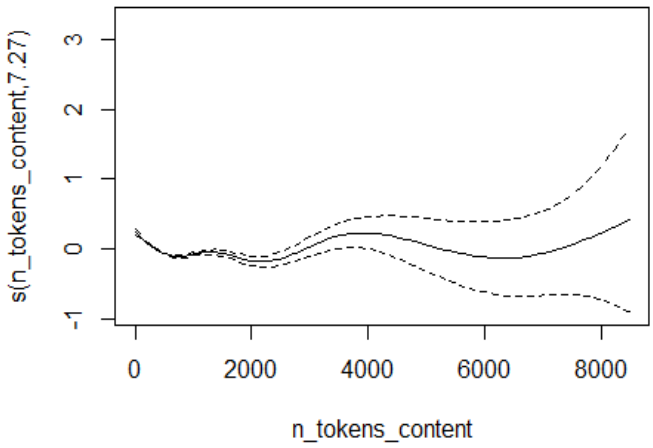
Si nota un aumento dell'R^2 rispetto al valore iniziale, anche se rimane ancora molto basso, identificando uno scarso adattamento del modello ai dati. Vi è anche un lieve miglioramento della distribuzione dei residui, che appaiono più simmetrici rispetto a prima.

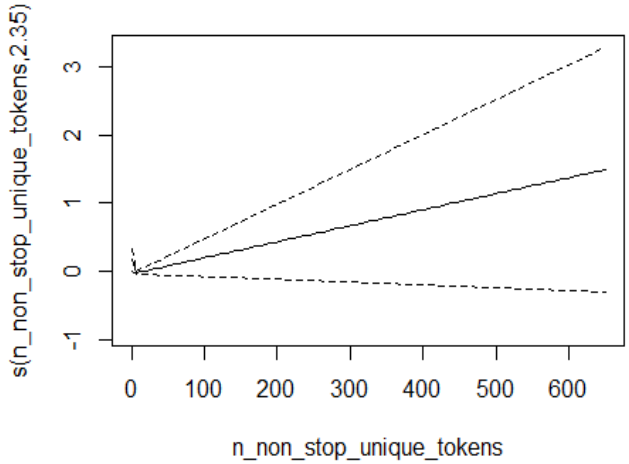
Trasformazione ottimale delle covariate

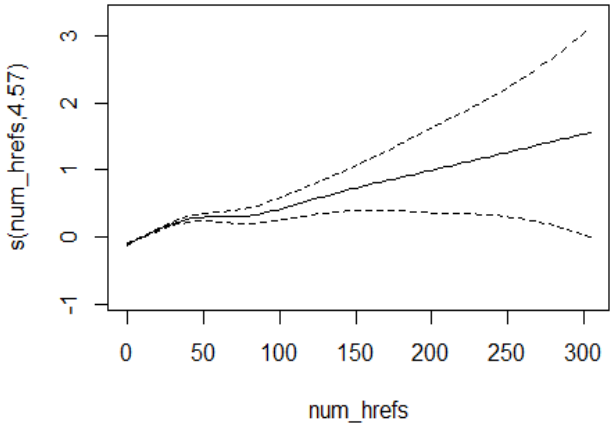
```
library(mgcv)
```

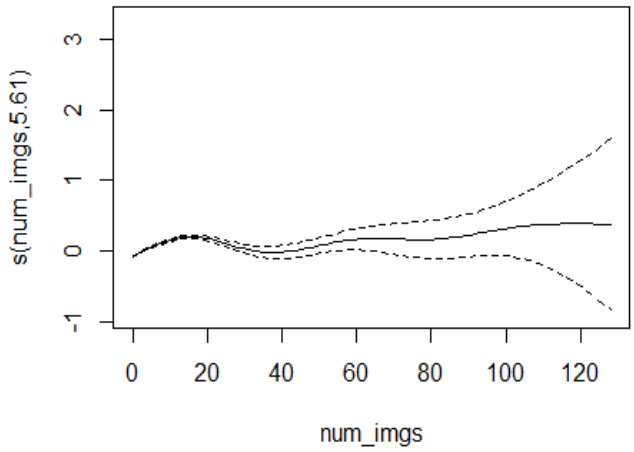
```r
gam1 <- gam(log(shares + 1) ~ s(n_tokens_title) + s(n_tokens_content) +
    s(n_non_stop_unique_tokens) + s(num_hrefs) + s(num_imgs) + s(num_videos) +
    s(average_token_length) + s(num_keywords) + s(rate_negative_words) +
    argomento + day, data = dati_completi)
summary(gam1)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(shares + 1) ~ s(n_tokens_title) + s(n_tokens_content) + s(n_non_stop_unique
_tokens) +
##     s(num_hrefs) + s(num_imgs) + s(num_videos) + s(average_token_length) +
##     s(num_keywords) + s(rate_negative_words) + argomento + day
##
## Parametric coefficients:
##                     Estimate Std. Error t value             Pr(>|t|)
## (Intercept)           7.5406     0.0154  490.80 < 0.0000000000000002 ***
## argomentoentertain   -0.1793     0.0149  -12.00 < 0.0000000000000002 ***
## argomentolifestyle    0.0454     0.0214    2.12                0.034 *
## argomentosocial me    0.2504     0.0202   12.38 < 0.0000000000000002 ***
## argomentotechnolog    0.0800     0.0147    5.46           0.000000049 ***
## argomentoworld       -0.1943     0.0146  -13.34 < 0.0000000000000002 ***
## daymonday            -0.0273     0.0162   -1.69                0.091 .
## daysaturd             0.2357     0.0217   10.87 < 0.0000000000000002 ***
## daysunday             0.2273     0.0209   10.88 < 0.0000000000000002 ***
## daythursd            -0.0670     0.0158   -4.23           0.000023445 ***
## daytuesda            -0.0804     0.0158   -5.10           0.000000346 ***
## daywednes            -0.0788     0.0158   -5.00           0.000000573 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                               edf Ref.df     F             p-value
## s(n_tokens_title)            1.95   2.48  0.35               0.643
## s(n_tokens_content)          7.27   8.21 25.44 < 0.0000000000000002 ***
## s(n_non_stop_unique_tokens) 2.35   2.87  3.92                0.021 *
## s(num_hrefs)                 4.57   5.48 49.08 < 0.0000000000000002 ***
## s(num_imgs)                  5.61   6.51 43.34 < 0.0000000000000002 ***
## s(num_videos)                8.64   8.95 28.06 < 0.0000000000000002 ***
## s(average_token_length)      4.15   5.14  8.83           0.000000016 ***
## s(num_keywords)              7.14   7.88 12.10 < 0.0000000000000002 ***
## s(rate_negative_words)       5.82   6.99  6.87           0.000000035 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0757   Deviance explained =  7.7%
## GCV = 0.80011  Scale est. = 0.79891    n = 39644

plot(gam1)
```

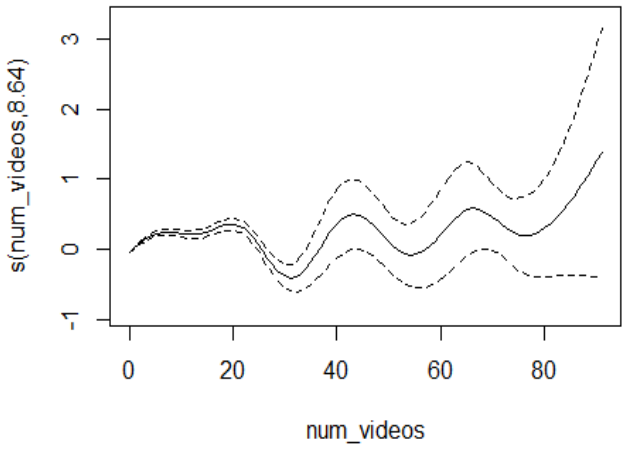

s(n_tokens_title,1.95)
-1
0
1
2
3
5
10
15
20
n_tokens_title

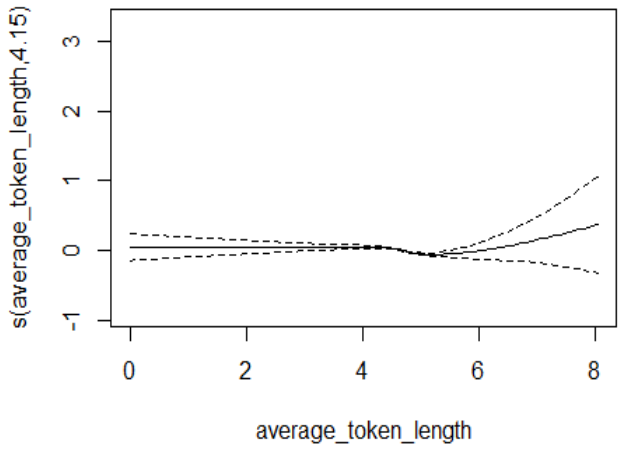

s(n_tokens_content,7.27)
-1
0
1
2
3
0
2000
4000
6000
8000
n_tokens_content

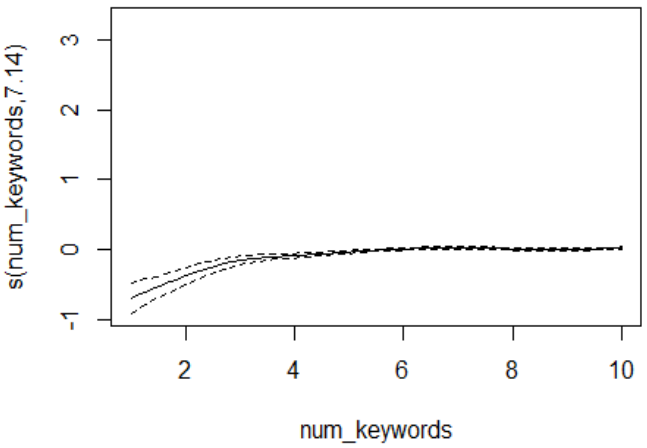

s(n_non_stop_unique_tokens,2.35)
-1
0
1
2
3
0
100
200
300
400
500
600
n_non_stop_unique_tokens

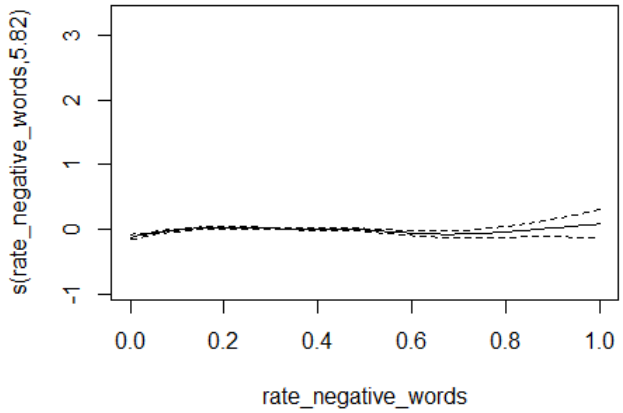

s(num_hrefs,4.57)
-1
0
1
2
3
0
50
100
150
200
250
300
num_hrefs


s(num_imgs,5.61)
-1
0
1
2
3
0
20
40
60
80
100
120
num_imgs


s(num_videos,8.64)
-1
0
1
2
3
0
20
40
60
80
num_videos







Le variabili "n_tokens_title", "num_keywords" e "num_hrefs" mostrano un andamento lineare, perciò non verranno sostituite da alcuna trasfrormazione.

Le variabili "n_tokens_content", "num_imgs" e "num_videos" mostrano un andamento che ricorda quello sinusoidale nella parte inziale, perciò per semplicità non subiranno alcuna trasformazione.

Per le variabili "n_non_stop_unique_tokens", "average_token_lenght" e "rate_negative_words" viene invece richiesta la trasformazione ottimale.

```
anova.gam(gam1, modello_base5, test="LRT")
```

```
## Warning in anova.gam(gam1, modello_base5, test = "LRT"): test argument ignored
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(shares + 1) ~ s(n_tokens_title) + s(n_tokens_content) + s(n_non_stop_unique
_tokens) +
##     s(num_hrefs) + s(num_imgs) + s(num_videos) + s(average_token_length) +
##     s(num_keywords) + s(rate_negative_words) + argomento + day
##
```

```
## Parametric Terms:
##           df     F             p-value
## argomento  5 171.2 <0.0000000000000002
## day        6  81.5 <0.0000000000000002
##
## Approximate significance of smooth terms:
##                               edf Ref.df     F              p-value
## s(n_tokens_title)            1.95   2.48  0.35                0.643
## s(n_tokens_content)          7.27   8.21 25.44 < 0.0000000000000002
## s(n_non_stop_unique_tokens) 2.35   2.87  3.92                0.021
## s(num_hrefs)                4.57   5.48 49.08 < 0.0000000000000002
## s(num_imgs)                 5.61   6.51 43.34 < 0.0000000000000002
## s(num_videos)               8.64   8.95 28.06 < 0.0000000000000002
## s(average_token_length)     4.15   5.14  8.83          0.000000016
## s(num_keywords)             7.14   7.88 12.10 < 0.0000000000000002
## s(rate_negative_words)      5.82   6.99  6.87          0.000000035
```

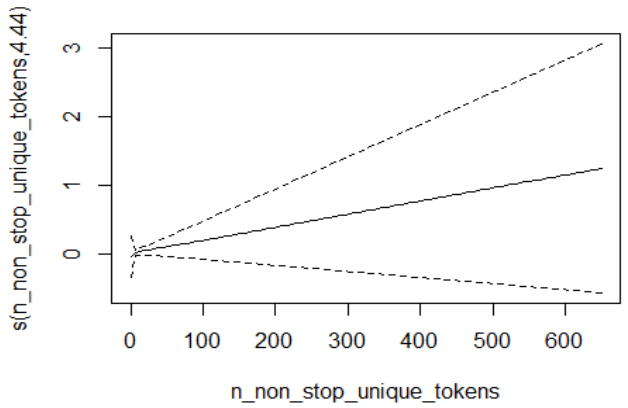
Tutte le trasformazioni appaiono significativamente utili, tranne quella per la variabile "n_tokens_title", il cui grafico appare perfettamente lineare.

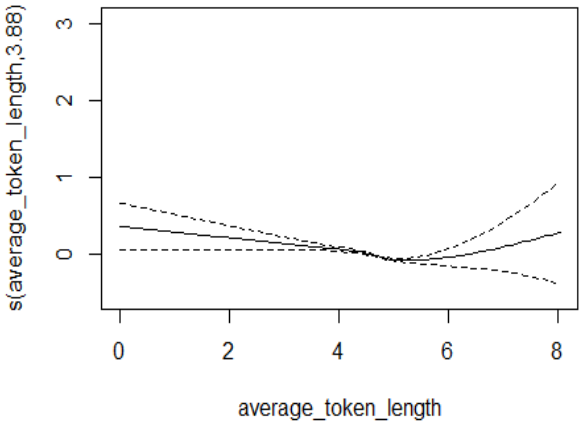
```
gam2 <- gam(log(shares + 1) ~ n_tokens_title + n_tokens_content +
    s(n_non_stop_unique_tokens) + num_hrefs + num_imgs + num_videos +
    s(average_token_length) + num_keywords + s(rate_negative_words) +
    argomento + day, data = dati_completi)
summary(gam2)
```

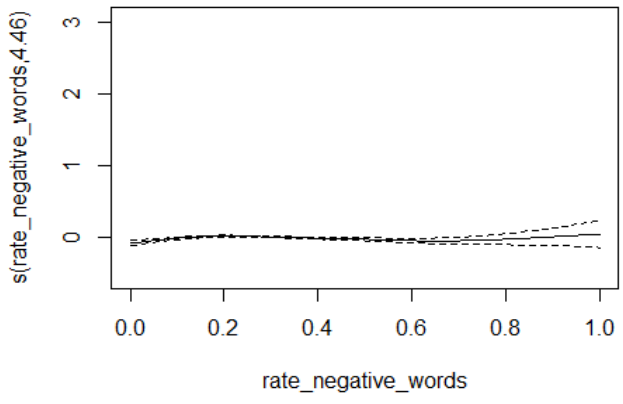
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(shares + 1) ~ n_tokens_title + n_tokens_content + s(n_non_stop_unique_token
s) +
##     num_hrefs + num_imgs + num_videos + s(average_token_length) +
##     num_keywords + s(rate_negative_words) + argomento + day
##
## Parametric coefficients:
##                      Estimate Std. Error t value             Pr(>|t|)
## (Intercept)         7.3125820  0.0324939  225.04 < 0.0000000000000002 ***
## n_tokens_title      0.0006727  0.0021869    0.31               0.7584
## n_tokens_content   -0.0000989  0.0000134   -7.37    0.000000000000177 ***
## num_hrefs           0.0076082  0.0004854   15.67 < 0.0000000000000002 ***
## num_imgs            0.0072707  0.0006818   10.66 < 0.0000000000000002 ***
## num_videos          0.0088972  0.0011593    7.67    0.000000000000017 ***
## num_keywords        0.0192809  0.0024914    7.74    0.000000000000010 ***
## argomentoentertain -0.1552647  0.0148882  -10.43 < 0.0000000000000002 ***
## argomentolifestyle  0.0556540  0.0214679    2.59               0.0095 **
## argomentosocial me  0.2481259  0.0201783   12.30 < 0.0000000000000002 ***
## argomentotechnolog  0.0922183  0.0146863    6.28    0.000000000343869 ***
## argomentoworld     -0.1950702  0.0145677  -13.39 < 0.0000000000000002 ***
## daymonday          -0.0270531  0.0162765   -1.66               0.0965 .
## daysaturd           0.2351137  0.0218215   10.77 < 0.0000000000000002 ***
```

```
## daysunday            0.2331687  0.0210306   11.09 < 0.0000000000000002 ***
## daythursd           -0.0670341  0.0159566   -4.20    0.000026627678567 ***
## daytuesda           -0.0780131  0.0159014   -4.91    0.000000932902443 ***
## daywednes           -0.0792502  0.0158807   -4.99    0.000000605213121 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                edf Ref.df     F           p-value
## s(n_non_stop_unique_tokens) 4.44   5.36  2.80           0.01578 *
## s(average_token_length)     3.88   4.86 14.69 0.000000000000069 ***
## s(rate_negative_words)      4.46   5.53  4.66           0.00015 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0604   Deviance explained = 6.11%
## GCV = 0.81269  Scale est. = 0.81206   n = 39644
```

```
plot(gam2)
```








```
modello_base6 <- lm(log(shares+1) ~ n_tokens_title + n_tokens_content +
    n_non_stop_unique_tokens + log(num_hrefs+1) + num_imgs + num_videos +
    average_token_length + num_keywords + rate_negative_words + argomento + day +
I(num_imgs^2) + I(num_videos^2)+ I(num_videos^3)+ I(num_imgs^3) , data = dati_comp
```

```
leti)
summary(modello_base6)

##
## Call:
## lm(formula = log(shares + 1) ~ n_tokens_title + n_tokens_content +
##     n_non_stop_unique_tokens + log(num_hrefs + 1) + num_imgs +
##     num_videos + average_token_length + num_keywords + rate_negative_words +
##     argomento + day + I(num_imgs^2) + I(num_videos^2) + I(num_videos^3) +
##     I(num_imgs^3), data = dati_completi)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.041 -0.579 -0.194  0.414  5.957
##
## Coefficients:
##                               Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)                7.575064684  0.041291203  183.45 < 0.0000000000000002 ***
## n_tokens_title             0.000297453  0.002176213    0.14                0.891
## n_tokens_content          -0.000071359  0.000011195   -6.37        0.00000000019 ***
## n_non_stop_unique_tokens   0.002211478  0.001385048    1.60                0.110
## log(num_hrefs + 1)         0.096522111  0.007551341   12.78 < 0.0000000000000002 ***
## num_imgs                   0.027573077  0.001740816   15.84 < 0.0000000000000002 ***
## num_videos                 0.054839100  0.003827354   14.33 < 0.0000000000000002 ***
## average_token_length      -0.092476257  0.006637438  -13.93 < 0.0000000000000002 ***
## num_keywords               0.014965029  0.002503594    5.98        0.00000000229 ***
## rate_negative_words       -0.012251787  0.031715465   -0.39                0.699
## argomentoentertain        -0.166789336  0.014886704  -11.20 < 0.0000000000000002 ***
## argomentolifestyle         0.048441296  0.021412243    2.26                0.024 *
## argomentosocial me         0.247750036  0.020090848   12.33 < 0.0000000000000002 ***
## argomentotechnolog         0.087135743  0.014569575    5.98        0.00000000224 ***
## argomentoworld            -0.215520031  0.014242140  -15.13 < 0.0000000000000002 ***
## daymonday                 -0.028239362  0.016232752   -1.74                0.082 .
## daysaturd                  0.228457535  0.021773723   10.49 < 0.0000000000000002 ***
## daysunday                  0.229284648  0.020966879   10.94 < 0.0000000000000002 ***
## daythursd                 -0.069806252  0.015912969   -4.39        0.00001153523 ***
## daytuesda                 -0.081004817  0.015858271   -5.11        0.00000032701 ***
## daywednes                 -0.080984601  0.015838750   -5.11        0.00000031844 ***
## I(num_imgs^2)             -0.000763415  0.000071595  -10.66 < 0.0000000000000002 ***
## I(num_videos^2)           -0.002604340  0.000223421  -11.66 < 0.0000000000000002 ***
## I(num_videos^3)            0.000026672  0.000002528   10.55 < 0.0000000000000002 ***
## I(num_imgs^3)              0.000005547  0.000000634    8.75 < 0.0000000000000002 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.899 on 39619 degrees of freedom
## Multiple R-squared:  0.0656, Adjusted R-squared:  0.065
## F-statistic:  116 on 24 and 39619 DF,  p-value: <0.0000000000000002
```

Per le variabili “num_imgs” e “num_videos” sono stati inseriti i termini quadratici e cubici, per riprodurre almeno in parte l’andamento curvilineo che presentano. Alla variabile “num_hrefs”, invece, è stata applicata la trasformazione logaritmica.

Model selection

Si applica la procedura di Best Subset per la miglior metrica AIC e poi si eliminano le variabili che non risultano significative.

```
library(MASS)
step <- stepAIC(modello_base6, direction="both")
```

```
## Start:  AIC=-8423
## log(shares + 1) ~ n_tokens_title + n_tokens_content + n_non_stop_unique_tokens
+
##     log(num_hrefs + 1) + num_imgs + num_videos + average_token_length +
##     num_keywords + rate_negative_words + argomento + day + I(num_imgs^2) +
##     I(num_videos^2) + I(num_videos^3) + I(num_imgs^3)
##
##                              Df Sum of Sq   RSS   AIC
## - n_tokens_title              1         0 32016 -8425
## - rate_negative_words         1         0 32016 -8425
## <none>                                    32015 -8423
## - n_non_stop_unique_tokens    1         2 32018 -8422
## - num_keywords                1        29 32044 -8389
## - n_tokens_content            1        33 32048 -8384
## - I(num_imgs^3)               1        62 32077 -8348
## - I(num_videos^3)             1        90 32105 -8313
## - I(num_imgs^2)               1        92 32107 -8311
## - I(num_videos^2)             1       110 32125 -8289
## - log(num_hrefs + 1)          1       132 32148 -8262
## - average_token_length        1       157 32172 -8231
## - num_videos                  1       166 32181 -8220
## - num_imgs                    1       203 32218 -8174
## - day                         6       392 32407 -7952
## - argomento                   5       784 32800 -7473
##
## Step:  AIC=-8425
## log(shares + 1) ~ n_tokens_content + n_non_stop_unique_tokens +
##     log(num_hrefs + 1) + num_imgs + num_videos + average_token_length +
##     num_keywords + rate_negative_words + argomento + day + I(num_imgs^2) +
##     I(num_videos^2) + I(num_videos^3) + I(num_imgs^3)
##
##                              Df Sum of Sq   RSS   AIC
## - rate_negative_words         1         0 32016 -8427
## <none>                                    32016 -8425
## - n_non_stop_unique_tokens    1         2 32018 -8424
## + n_tokens_title              1         0 32015 -8423
## - num_keywords                1        29 32044 -8391
## - n_tokens_content            1        33 32048 -8386
## - I(num_imgs^3)               1        62 32077 -8350
## - I(num_videos^3)             1        90 32106 -8315
## - I(num_imgs^2)               1        92 32107 -8313
## - I(num_videos^2)             1       110 32125 -8291
## - log(num_hrefs + 1)          1       132 32148 -8263
```

```
## - average_token_length      1       157 32173 -8232
## - num_videos                1       166 32182 -8221
## - num_imgs                  1       203 32218 -8176
## - day                       6       392 32407 -7954
## - argomento                 5       795 32811 -7462
##
## Step:  AIC=-8427
## log(shares + 1) ~ n_tokens_content + n_non_stop_unique_tokens +
##     log(num_hrefs + 1) + num_imgs + num_videos + average_token_length +
##     num_keywords + argomento + day + I(num_imgs^2) + I(num_videos^2) +
##     I(num_videos^3) + I(num_imgs^3)
##
##                             Df Sum of Sq   RSS   AIC
## <none>                                   32016 -8427
## - n_non_stop_unique_tokens  1         2 32018 -8426
## + rate_negative_words       1         0 32016 -8425
## + n_tokens_title            1         0 32016 -8425
## - num_keywords              1        29 32045 -8392
## - n_tokens_content          1        33 32049 -8388
## - I(num_imgs^3)             1        62 32078 -8352
## - I(num_videos^3)           1        90 32106 -8317
## - I(num_imgs^2)             1        92 32108 -8315
## - I(num_videos^2)           1       110 32126 -8292
## - log(num_hrefs + 1)        1       133 32148 -8265
## - num_videos                1       166 32182 -8223
## - average_token_length      1       175 32191 -8212
## - num_imgs                  1       203 32218 -8178
## - day                       6       392 32407 -7956
## - argomento                 5       833 32849 -7418
```

Il modello migliore risulta essere:

```
modello_mod_sel <- lm(log(shares+1) ~ n_tokens_content + n_non_stop_unique_tokens
+ log(num_hrefs+1) + num_imgs + num_videos + average_token_length + num_keywords +
argomento + day + I(num_imgs^2) + I(num_videos^2)+ I(num_videos^3)+ I(num_imgs^3)
, data = dati_completi)
summary(modello_mod_sel)
```

```
##
## Call:
## lm(formula = log(shares + 1) ~ n_tokens_content + n_non_stop_unique_tokens +
##     log(num_hrefs + 1) + num_imgs + num_videos + average_token_length +
##     num_keywords + argomento + day + I(num_imgs^2) + I(num_videos^2) +
##     I(num_videos^3) + I(num_imgs^3), data = dati_completi)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.040 -0.579 -0.194  0.414  5.956
##
## Coefficients:
##                              Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)              7.577984520  0.033794346  224.24 < 0.0000000000000002 ***
## n_tokens_content        -0.000071479  0.000011175   -6.40        0.00000000016 ***
```

```
## n_non_stop_unique_tokens  0.002216551  0.001384917    1.60                0.109
## log(num_hrefs + 1)        0.096584280  0.007542516   12.81 < 0.0000000000000002 ***
## num_imgs                  0.027568087  0.001740390   15.84 < 0.0000000000000002 ***
## num_videos                0.054842947  0.003824186   14.34 < 0.0000000000000002 ***
## average_token_length     -0.093269492  0.006332734  -14.73 < 0.0000000000000002 ***
## num_keywords              0.015041445  0.002496112    6.03        0.00000000170 ***
## argomentoentertain       -0.167163887  0.014779328  -11.31 < 0.0000000000000002 ***
## argomentolifestyle        0.047982398  0.021371450    2.25                0.025 *
## argomentosocial me        0.247646144  0.020057924   12.35 < 0.0000000000000002 ***
## argomentotechnolog        0.087023620  0.014566333    5.97        0.00000000233 ***
## argomentoworld           -0.216490150  0.013976078  -15.49 < 0.0000000000000002 ***
## daymonday                -0.028195196  0.016232011   -1.74                0.082 .
## daysaturd                 0.228522159  0.021771698   10.50 < 0.0000000000000002 ***
## daysunday                 0.229365914  0.020965350   10.94 < 0.0000000000000002 ***
## daythursd                -0.069742249  0.015910816   -4.38        0.00001171829 ***
## daytuesda                -0.080871292  0.015854485   -5.10        0.00000033969 ***
## daywednes                -0.080863653  0.015835575   -5.11        0.00000032978 ***
## I(num_imgs^2)            -0.000763732  0.000071570  -10.67 < 0.0000000000000002 ***
## I(num_videos^2)          -0.002607218  0.000223260  -11.68 < 0.0000000000000002 ***
## I(num_videos^3)           0.000026712  0.000002526   10.58 < 0.0000000000000002 ***
## I(num_imgs^3)             0.000005551  0.000000634    8.76 < 0.0000000000000002 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.899 on 39621 degrees of freedom
## Multiple R-squared:  0.0656, Adjusted R-squared:  0.0651
## F-statistic:  126 on 22 and 39621 DF,  p-value: <0.0000000000000002
```

Nell'insieme di covariate conservate dal best subset ce n'è una non significativa, che deve essere eliminata: si tratta di "n_non_stop_unique_tokens".

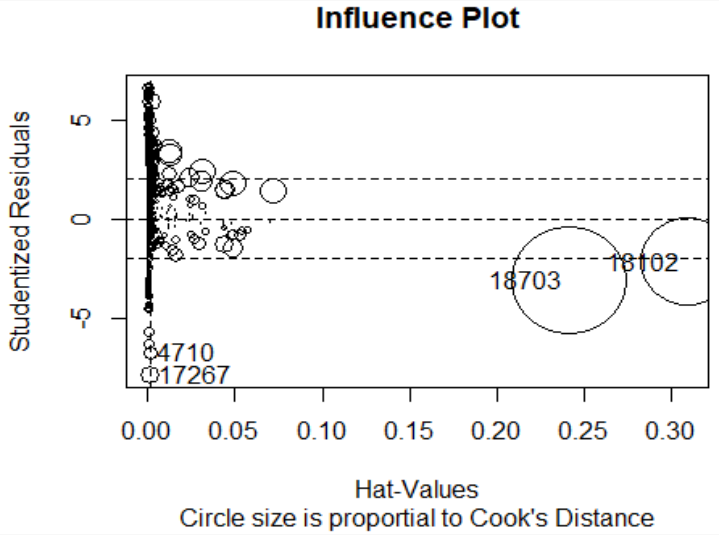
```
modello_mod_sel <- lm(log(shares+1) ~ n_tokens_content + log(num_hrefs+1) + num_im
gs + num_videos + average_token_length + num_keywords + argomento + day + I(num_im
gs^2) + I(num_videos^2)+ I(num_videos^3)+ I(num_imgs^3) , data = dati_completi)
summary(modello_mod_sel)

##
## Call:
## lm(formula = log(shares + 1) ~ n_tokens_content + log(num_hrefs +
##     1) + num_imgs + num_videos + average_token_length + num_keywords +
##     argomento + day + I(num_imgs^2) + I(num_videos^2) + I(num_videos^3) +
##     I(num_imgs^3), data = dati_completi)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.040 -0.579 -0.194  0.414  5.956
##
## Coefficients:
##                           Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)           7.578176888  0.033794799  224.24 < 0.0000000000000002 ***
## n_tokens_content     -0.000071575  0.000011175   -6.41        0.00000000015 ***
## log(num_hrefs + 1)    0.096504855  0.007542501   12.79 < 0.0000000000000002 ***
## num_imgs              0.027495601  0.001739835   15.80 < 0.0000000000000002 ***
```

```
## num_videos             0.054832932  0.003824256   14.34 < 0.0000000000000002 ***
## average_token_length -0.092920130  0.006329095  -14.68 < 0.0000000000000002 ***
## num_keywords           0.015043326  0.002496161    6.03        0.00000000169 ***
## argomentoentertain    -0.167016709  0.014779333  -11.30 < 0.0000000000000002 ***
## argomentolifestyle     0.048048795  0.021371831    2.25                0.025 *
## argomentosocial me     0.247627520  0.020058315   12.35 < 0.0000000000000002 ***
## argomentotechnolog     0.087034213  0.014566619    5.97        0.00000000232 ***
## argomentoworld        -0.216549690  0.013976304  -15.49 < 0.0000000000000002 ***
## daymonday             -0.028206235  0.016232329   -1.74                0.082 .
## daysaturd              0.228528357  0.021772127   10.50 < 0.0000000000000002 ***
## daysunday              0.229329513  0.020965751   10.94 < 0.0000000000000002 ***
## daythursd             -0.069748704  0.015911129   -4.38        0.00001170112 ***
## daytuesda             -0.080683495  0.015854364   -5.09        0.00000036152 ***
## daywednes             -0.080865122  0.015835887   -5.11        0.00000032979 ***
## I(num_imgs^2)         -0.000759880  0.000071531  -10.62 < 0.0000000000000002 ***
## I(num_videos^2)       -0.002606885  0.000223264  -11.68 < 0.0000000000000002 ***
## I(num_videos^3)        0.000026708  0.000002526   10.57 < 0.0000000000000002 ***
## I(num_imgs^3)          0.000005520  0.000000633    8.72 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.899 on 39622 degrees of freedom
## Multiple R-squared:  0.0655, Adjusted R-squared:  0.065
## F-statistic:  132 on 21 and 39622 DF,  p-value: <0.0000000000000002
```

A questo punto tutte le covariate selezionate sono effettivamente significative.

Eliminazione dei punti influenti

```
library(car)
influencePlot(modello_mod_sel, main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance")
```




```
##       StudRes     Hat    CookD
## 4710    -6.68 0.00109 0.00221
```

```
## 17267   -7.84 0.00139 0.00389
## 18102   -2.15 0.30888 0.09413
## 18703   -3.12 0.24178 0.14115
```

```
cooksd <- cooks.distance(modello_mod_sel)
cooksda=data.frame(cooksd)
summary(cooksd)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000016 0.0000065 0.0000326 0.0000199 0.1411537
```

```
n_used=length(modello_mod_sel$residuals)
n_used
```

```
## [1] 39644
```

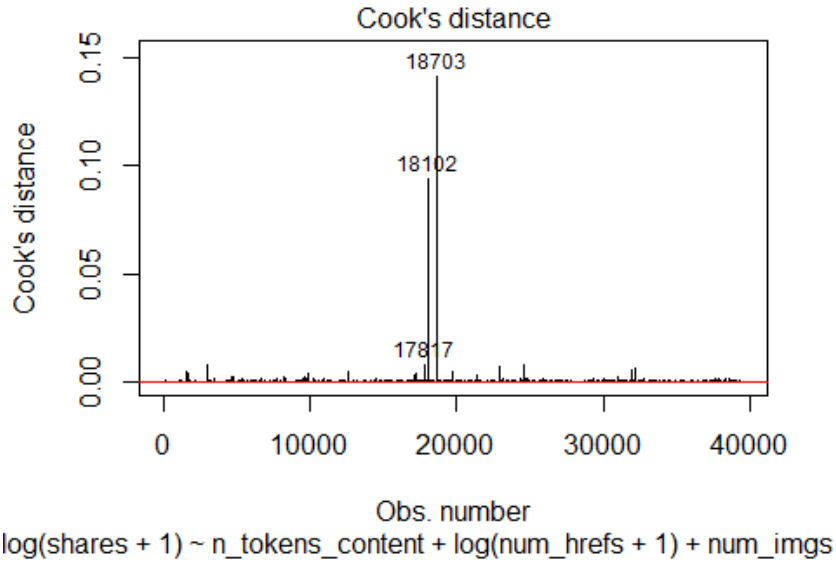
```
nrow(dati_completi)
```

```
## [1] 39644
```

```
# usa tutto
```

```
cutoff <- 4/(n_used)
```

```
plot(modello_mod_sel, which=4, cook.levels=cutoff)
abline(h=cutoff, col='red')
```



Eliminiamo i punti influenti:

```
NOinflu=data.frame(dati_completi[cooksd < cutoff, ])
```

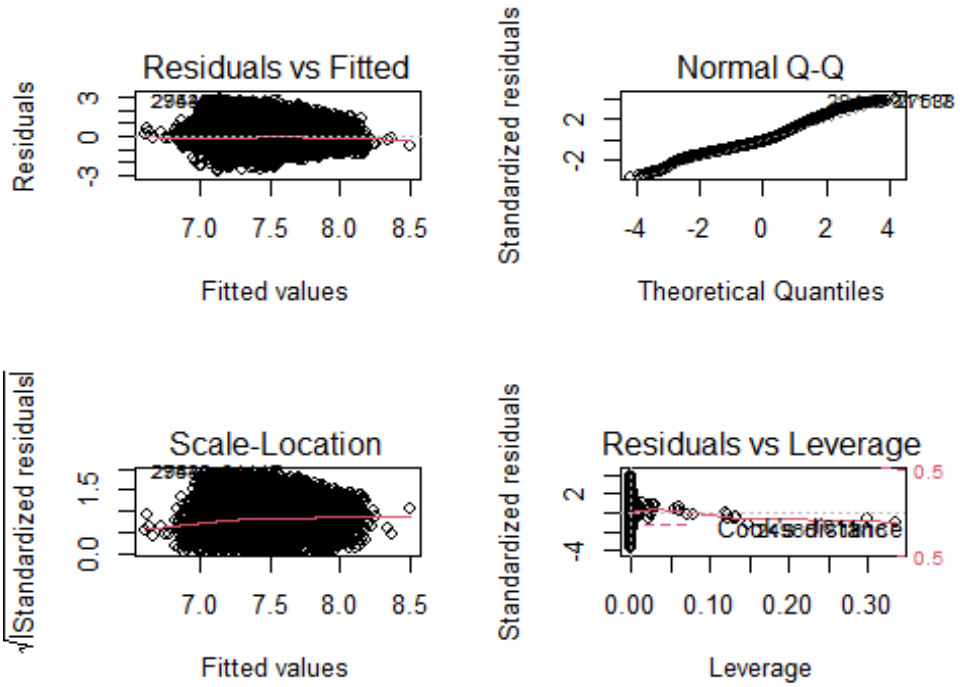
NOinflu corrisponde a dati_completi senza i punti influenti.

Fit del modello migliore sui punti non influenti:

```
modello_mod_sel2 <- lm(log(shares+1) ~ n_tokens_content +log(num_hrefs+1) + num_im
gs + num_videos + average_token_length + num_keywords + argomento + day + I(num_im
gs^2) + I(num_videos^2)+ I(num_videos^3)+ I(num_imgs^3) , data = NOinflu)
summary(modello_mod_sel)

##
## Call:
## lm(formula = log(shares + 1) ~ n_tokens_content + log(num_hrefs +
##     1) + num_imgs + num_videos + average_token_length + num_keywords +
##     argomento + day + I(num_imgs^2) + I(num_videos^2) + I(num_videos^3) +
##     I(num_imgs^3), data = dati_completi)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.040 -0.579 -0.194  0.414  5.956
##
## Coefficients:
##                            Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)           7.578176888  0.033794799  224.24 < 0.0000000000000002 ***
## n_tokens_content     -0.000071575  0.000011175   -6.41        0.00000000015 ***
## log(num_hrefs + 1)    0.096504855  0.007542501   12.79 < 0.0000000000000002 ***
## num_imgs              0.027495601  0.001739835   15.80 < 0.0000000000000002 ***
## num_videos            0.054832932  0.003824256   14.34 < 0.0000000000000002 ***
## average_token_length -0.092920130  0.006329095  -14.68 < 0.0000000000000002 ***
## num_keywords          0.015043326  0.002496161    6.03        0.00000000169 ***
## argomentoentertain   -0.167016709  0.014779333  -11.30 < 0.0000000000000002 ***
## argomentolifestyle    0.048048795  0.021371831    2.25                0.025 *
## argomentosocial me    0.247627520  0.020058315   12.35 < 0.0000000000000002 ***
## argomentotechnolog    0.087034213  0.014566619    5.97        0.00000000232 ***
## argomentoworld       -0.216549690  0.013976304  -15.49 < 0.0000000000000002 ***
## daymonday            -0.028206235  0.016232329   -1.74                0.082 .
## daysaturd             0.228528357  0.021772127   10.50 < 0.0000000000000002 ***
## daysunday             0.229329513  0.020965751   10.94 < 0.0000000000000002 ***
## daythursd            -0.069748704  0.015911129   -4.38        0.00001170112 ***
## daytuesda            -0.080683495  0.015854364   -5.09        0.00000036152 ***
## daywednes            -0.080865122  0.015835887   -5.11        0.00000032979 ***
## I(num_imgs^2)        -0.000759880  0.000071531  -10.62 < 0.0000000000000002 ***
## I(num_videos^2)      -0.002606885  0.000223264  -11.68 < 0.0000000000000002 ***
## I(num_videos^3)       0.000026708  0.000002526   10.57 < 0.0000000000000002 ***
## I(num_imgs^3)         0.000005520  0.000000633    8.72 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.899 on 39622 degrees of freedom
## Multiple R-squared:  0.0655, Adjusted R-squared:  0.065
## F-statistic:  132 on 21 and 39622 DF,  p-value: <0.0000000000000002
```

Diagnostiche dei residui del modello dopo le best trasformation per le variabili e l'esclusione dei punti influenti

```
par(mfrow=c(2,2))
plot(modello_mod_sel2)
```




```
par(mfrow=c(1,1))
```

Osserviamo che la LOESS stimata nel grafico tra residui vs fitted values è sostanzialmente orizzontale, a indicare che l'eliminazione dei punti influenti migliora la relazione lineare del modello; sempre nello stesso grafico si osserva, tuttavia, che la variabilità dei residui diminuisce all'aumentare dei valori fittati dal modello: probabilmente non resta che applicare gli standard error di White per tutelare l'inferenza. Infine nel qq-plot si nota un sensibile miglioramento della normalità dei residui a seguito del cut-off dei punti influenti.

Eteroschedasticità

Test per l'eteroschedasticità dopo l'esclusione dei punti influenti

```
bptest(modello_mod_sel2)

##
##  studentized Breusch-Pagan test
##
## data:  modello_mod_sel2
## BP = 538.4, df = 21, p-value <0.0000000000000002
```

Nonostante l'eliminazione dei valori influenti, il test di Breusch-Pagan mostra la persistenza di eteroschedasticità. L'unico rimedio rimasto è applicare la correzione di White per gli Standard Error.

Standard errors robusti di White

Osserviamo che il problema dell'eteroschedasticità non si risolve con l'eliminazione dei punti influenti, perciò procediamo con la correzione di White per gli standard errors.

```
library(sandwich)
library(lmtest)
coeftest(modello_mod_sel2, vcov = vcovHC(modello_mod_sel2))
```

```
##
## t test of coefficients:
##
##                           Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)           7.4180569211  0.0270660059 274.073 < 0.0000000000000002 ***
## n_tokens_content     -0.0000327500  0.0000088073  -3.719             0.000201 ***
## log(num_hrefs + 1)    0.0766929906  0.0061263066  12.519 < 0.0000000000000002 ***
## num_imgs              0.0259340355  0.0015882987  16.328 < 0.0000000000000002 ***
## num_videos            0.0424892187  0.0034789310  12.213 < 0.0000000000000002 ***
## average_token_length -0.0597726146  0.0050841625 -11.757 < 0.0000000000000002 ***
## num_keywords          0.0115398770  0.0020159358   5.724       0.000000010463 ***
## argomentoentertain   -0.1850736177  0.0122887203 -15.060 < 0.0000000000000002 ***
## argomentolifestyle   -0.0030064282  0.0173807352  -0.173             0.862672
## argomentosocial me    0.2718599701  0.0161316702  16.853 < 0.0000000000000002 ***
## argomentotechnolog    0.1193930847  0.0120984408   9.868 < 0.0000000000000002 ***
## argomentoworld       -0.2147879092  0.0112017784 -19.174 < 0.0000000000000002 ***
## daymonday            -0.0338292531  0.0134263612  -2.520             0.011752 *
## daysaturd             0.2271085309  0.0162639101  13.964 < 0.0000000000000002 ***
## daysunday             0.2267643779  0.0161192384  14.068 < 0.0000000000000002 ***
## daythursd            -0.0680736103  0.0129320935  -5.264       0.000000141782 ***
## daytuesda            -0.0791203892  0.0129195592  -6.124       0.000000000921 ***
## daywednes            -0.0824328702  0.0129007172  -6.390       0.000000000168 ***
## I(num_imgs^2)        -0.0009870608  0.0000682814 -14.456 < 0.0000000000000002 ***
## I(num_videos^2)      -0.0025691726  0.0002161694 -11.885 < 0.0000000000000002 ***
## I(num_videos^3)       0.0000293935  0.0000030010   9.795 < 0.0000000000000002 ***
## I(num_imgs^3)         0.0000083775  0.0000006805  12.311 < 0.0000000000000002 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(lmSupport)
```

```
modelCorrectSE(modello_mod_sel2)
```

```
## Uncorrected Tests of Coefficients
##
##                            Estimate  Std. Error t value
## (Intercept)            7.41805692 0.028668200  258.76
## n_tokens_content      -0.00003275 0.000009558   -3.43
## log(num_hrefs + 1)     0.07669299 0.006293258   12.19
## num_imgs               0.02593404 0.001642389   15.79
## num_videos             0.04248922 0.003655542   11.62
## average_token_length -0.05977261 0.005452229  -10.96
## num_keywords           0.01153988 0.002049390    5.63
## argomentoentertain    -0.18507362 0.012132079  -15.25
```

```
## argomentolifestyle   -0.00300643 0.017734606   -0.17
## argomentosocial me    0.27185997 0.016561602   16.42
## argomentotechnolog    0.11939308 0.011855547   10.07
## argomentoworld       -0.21478791 0.011384439  -18.87
## daymonday            -0.03382925 0.013302935   -2.54
## daysaturd             0.22710853 0.017997844   12.62
## daysunday             0.22676438 0.017264436   13.13
## daythursd            -0.06807361 0.013021740   -5.23
## daytuesda            -0.07912039 0.012970687   -6.10
## daywednes            -0.08243287 0.012953753   -6.36
## I(num_imgs^2)        -0.00098706 0.000078872  -12.51
## I(num_videos^2)      -0.00256917 0.000239998  -10.70
## I(num_videos^3)       0.00002939 0.000003262    9.01
## I(num_imgs^3)         0.00000838 0.000000838    9.99
##
## White (1980) Heteroscedasticity-corrected SEs and Tests
##
##                         Estimate  Std. Error t value
## (Intercept)           7.41805692 0.027066006 274.073
## n_tokens_content     -0.00003275 0.000008807  -3.719
## log(num_hrefs + 1)    0.07669299 0.006126307  12.519
## num_imgs              0.02593404 0.001588299  16.328
## num_videos            0.04248922 0.003478931  12.213
## average_token_length -0.05977261 0.005084162 -11.757
## num_keywords          0.01153988 0.002015936   5.724
## argomentoentertain   -0.18507362 0.012288720 -15.060
## argomentolifestyle   -0.00300643 0.017380735  -0.173
## argomentosocial me    0.27185997 0.016131670  16.853
## argomentotechnolog    0.11939308 0.012098441   9.868
## argomentoworld       -0.21478791 0.011201778 -19.174
## daymonday            -0.03382925 0.013426361  -2.520
## daysaturd             0.22710853 0.016263910  13.964
## daysunday             0.22676438 0.016119238  14.068
## daythursd            -0.06807361 0.012932093  -5.264
## daytuesda            -0.07912039 0.012919559  -6.124
## daywednes            -0.08243287 0.012900717  -6.390
## I(num_imgs^2)        -0.00098706 0.000068281 -14.456
## I(num_videos^2)      -0.00256917 0.000216169 -11.885
## I(num_videos^3)       0.00002939 0.000003001   9.795
## I(num_imgs^3)         0.00000838 0.000000681  12.311
##

bptest(modello_mod_sel2)

##
##  studentized Breusch-Pagan test
##
## data:  modello_mod_sel2
## BP = 538.4, df = 21, p-value <0.0000000000000002
```

L'esito del test di Breusch-Pagan rimane invariato, in quanto gli Standard Error di White non eliminano l'eteroschedasticità, ma ne tengono conto per correggere le stime degli Standard Error dei coefficienti OLS.

Bootstrap

```
library(car)
boot_model <- Boot(modello_mod_sel2, R=1999)
```

Osservando la distanza tra il parametro stimato con MLE e il parametro stimato con BootStrap (bootBias), si nota che questa quantità è molto piccola per ogni coefficiente stimato.

```
summary(boot_model)
```

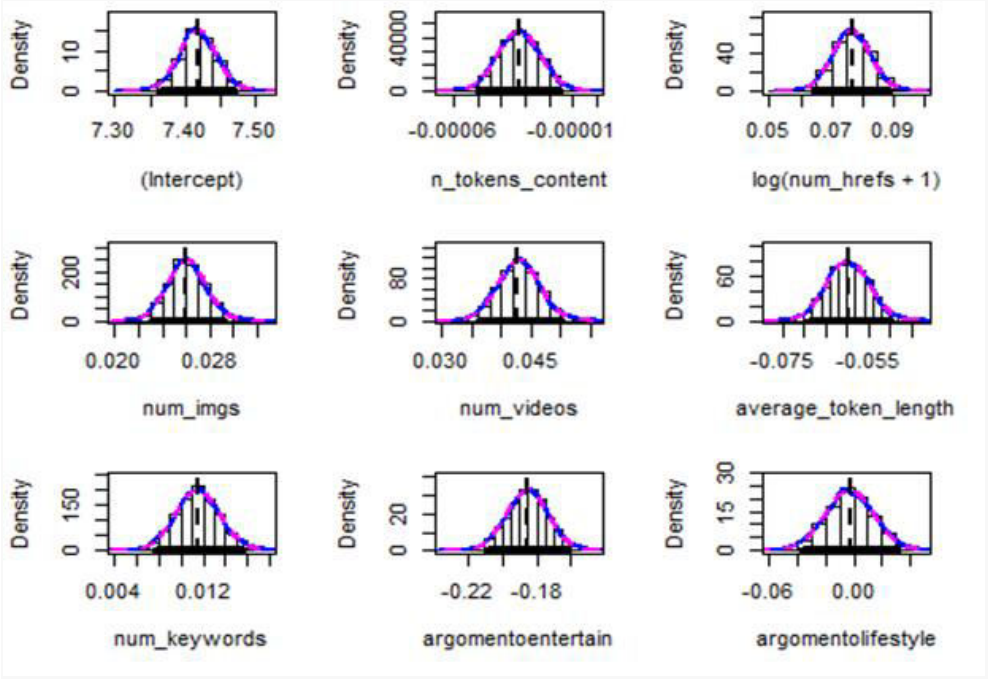
```
##
## Number of bootstrap replications R = 1999
##                            original      bootBias       bootSE      bootMed
## (Intercept)           7.418056921 -0.0000695281 0.0267629594  7.417855162
## n_tokens_content     -0.000032750  0.0000001656 0.0000088841 -0.000032646
## log(num_hrefs + 1)    0.076692991 -0.0001091763 0.0061668363  0.076423086
## num_imgs              0.025934036  0.0001138265 0.0016110894  0.025993015
## num_videos            0.042489219  0.0004261900 0.0034911540  0.042920841
## average_token_length -0.059772615  0.0000484419 0.0049878858 -0.059827386
## num_keywords          0.011539877 -0.0000265706 0.0020044985  0.011514773
## argomentoentertain   -0.185073618 -0.0001884666 0.0122306906 -0.185050671
## argomentolifestyle   -0.003006428 -0.0000036694 0.0173318592 -0.003362676
## argomentosocial me    0.271859970 -0.0000358991 0.0159672023  0.272219222
## argomentotechnolog    0.119393085 -0.0002178824 0.0120320511  0.119388332
## argomentoworld       -0.214787909  0.0000629729 0.0110340952 -0.214642304
## daymonday            -0.033829253 -0.0000993231 0.0136768254 -0.033825412
## daysaturd             0.227108531 -0.0005092141 0.0161542273  0.226469173
## daysunday             0.226764378 -0.0003515151 0.0160884345  0.226423880
## daythursd            -0.068073610 -0.0002603714 0.0128537906 -0.068566352
## daytuesda            -0.079120389  0.0001109377 0.0131107433 -0.079342620
## daywednes            -0.082432870 -0.0001810667 0.0130671333 -0.082838646
## I(num_imgs^2)        -0.000987061 -0.0000085753 0.0000701814 -0.000993484
## I(num_videos^2)      -0.002569173 -0.0000415592 0.0002248152 -0.002598049
## I(num_videos^3)       0.000029393  0.0000008617 0.0000033572  0.000029850
## I(num_imgs^3)         0.000008377  0.0000001211 0.0000007191  0.000008442
```

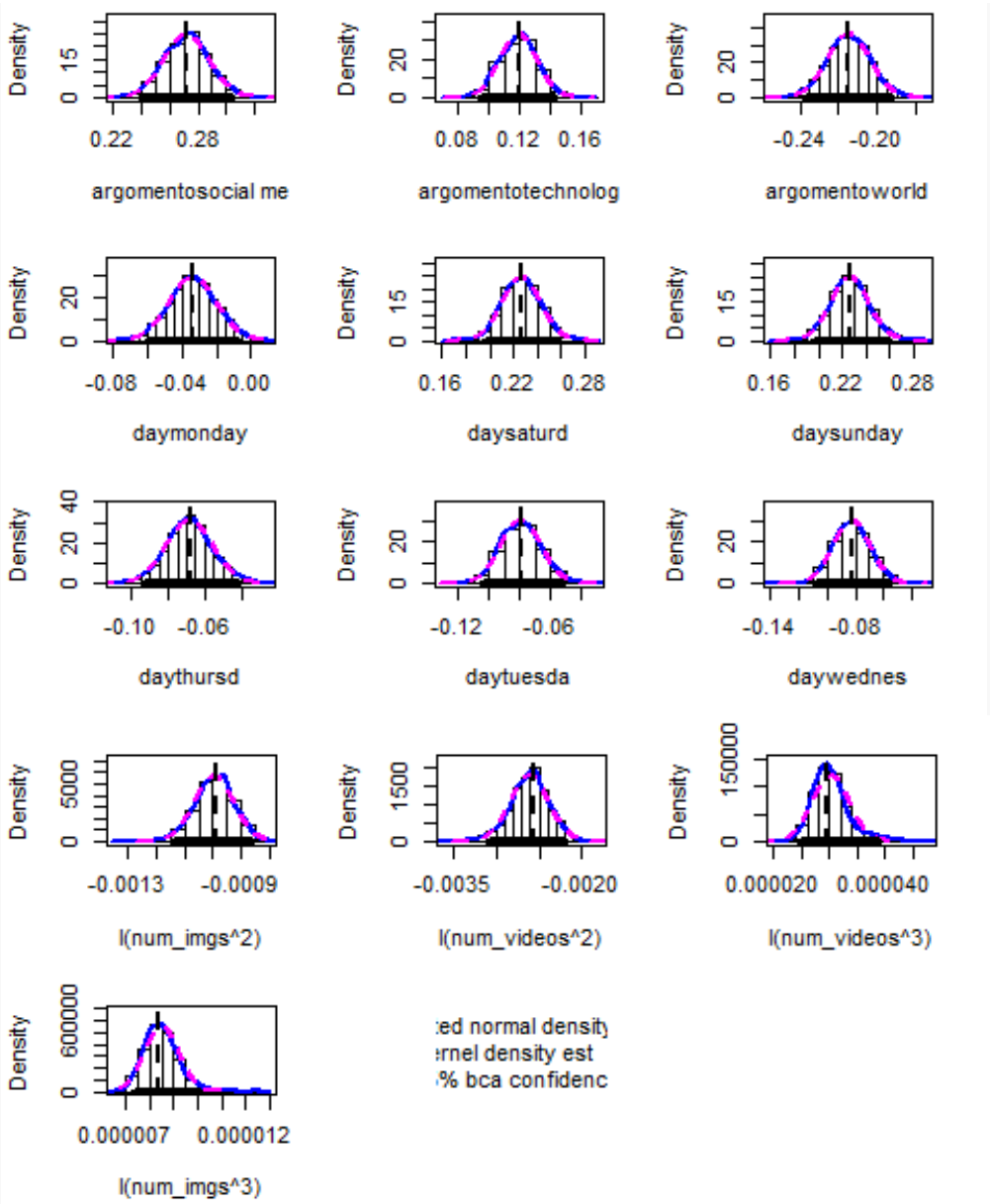
```
Confint(boot_model, level=c(0.95), type="perc")
```

```
## Bootstrap percent confidence intervals
##
##                             Estimate          2.5 %         97.5 %
## (Intercept)           7.41805692105  7.36379826204  7.4703072782
## n_tokens_content     -0.00003274998 -0.00004960862 -0.0000157538
## log(num_hrefs + 1)    0.07669299060  0.06432613820  0.0887402772
## num_imgs              0.02593403553  0.02301523508  0.0292968371
## num_videos            0.04248921867  0.03600654313  0.0500820854
```

```
## average_token_length -0.05977261459 -0.06927801698 -0.0495074902
## num_keywords          0.01153987700  0.00764640903  0.0155457978
## argomentoentertain   -0.18507361773 -0.20945801854 -0.1613238857
## argomentolifestyle   -0.00300642815 -0.03797659752  0.0308253467
## argomentosocial me    0.27185997012  0.24038635606  0.3037498337
## argomentotechnolog    0.11939308465  0.09514708707  0.1421403473
## argomentoworld       -0.21478790921 -0.23749312179 -0.1927411878
## daymonday            -0.03382925307 -0.06073398105 -0.0072533638
## daysaturd             0.22710853089  0.19567684491  0.2581199153
## daysunday             0.22676437788  0.19524574996  0.2596779299
## daythursd            -0.06807361032 -0.09379888311 -0.0416121847
## daytuesda            -0.07912038920 -0.10419656482 -0.0528326245
## daywednes            -0.08243287019 -0.10795913699 -0.0567457538
## I(num_imgs^2)        -0.00098706079 -0.00114199108 -0.0008675040
## I(num_videos^2)      -0.00256917258 -0.00309636975 -0.0021979584
## I(num_videos^3)       0.00002939348  0.00002483609  0.0000387354
## I(num_imgs^3)         0.00000837746  0.00000727948  0.0000100528
```

```
hist(boot_model, ask=T, legend="separate")
```





Confronto con le stime OLS:

```
summary(modello_mod_sel2)
```

```
##
## Call:
## lm(formula = log(shares + 1) ~ n_tokens_content + log(num_hrefs +
##     1) + num_imgs + num_videos + average_token_length + num_keywords +
##     argomento + day + I(num_imgs^2) + I(num_videos^2) + I(num_videos^3) +
##     I(num_imgs^3), data = NOinflu)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.625 -0.500 -0.141  0.398  2.766
##
```

```
## Coefficients:
##                            Estimate   Std. Error t value             Pr(>|t|)    
## (Intercept)           7.418056921  0.028668200  258.76 < 0.0000000000000002 ***
## n_tokens_content     -0.000032750  0.000009558   -3.43              0.00061 ***
## log(num_hrefs + 1)    0.076692991  0.006293258   12.19 < 0.0000000000000002 ***
## num_imgs              0.025934036  0.001642389   15.79 < 0.0000000000000002 ***
## num_videos            0.042489219  0.003655542   11.62 < 0.0000000000000002 ***
## average_token_length -0.059772615  0.005452229  -10.96 < 0.0000000000000002 ***
## num_keywords          0.011539877  0.002049390    5.63          0.0000000181 ***
## argomentoentertain   -0.185073618  0.012132079  -15.25 < 0.0000000000000002 ***
## argomentolifestyle   -0.003006428  0.017734606   -0.17              0.86539    
## argomentosocial me    0.271859970  0.016561602   16.42 < 0.0000000000000002 ***
## argomentotechnolog    0.119393085  0.011855547   10.07 < 0.0000000000000002 ***
## argomentoworld       -0.214787909  0.011384439  -18.87 < 0.0000000000000002 ***
## daymonday            -0.033829253  0.013302935   -2.54              0.01099 *  
## daysaturd             0.227108531  0.017997844   12.62 < 0.0000000000000002 ***
## daysunday             0.226764378  0.017264436   13.13 < 0.0000000000000002 ***
## daythursd            -0.068073610  0.013021740   -5.23          0.0000001726 ***
## daytuesda            -0.079120389  0.012970687   -6.10          0.0000000011 ***
## daywednes            -0.082432870  0.012953753   -6.36          0.0000000002 ***
## I(num_imgs^2)        -0.000987061  0.000078872  -12.51 < 0.0000000000000002 ***
## I(num_videos^2)      -0.002569173  0.000239998  -10.70 < 0.0000000000000002 ***
## I(num_videos^3)       0.000029393  0.000003262    9.01 < 0.0000000000000002 ***
## I(num_imgs^3)         0.000008377  0.000000838    9.99 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.717 on 37619 degrees of freedom
## Multiple R-squared:  0.0856, Adjusted R-squared:  0.0851 
## F-statistic:  168 on 21 and 37619 DF,  p-value: <0.0000000000000002
```

La linea tratteggiata rappresenta la stima del parametro MLE. Poichè le linee tratteggiate di tutte le variabili cadono all'interno dell'intervallo di confidenza Bootstrap, risulta che tutte le stime MSE siano robuste.

Se l'intervallo di confidenza Bootstrap (corrispondente alla riga nera marcata in basso) comprende lo zero, il parametro risulta non significativo, mentre se non lo comprende risulta significativo.

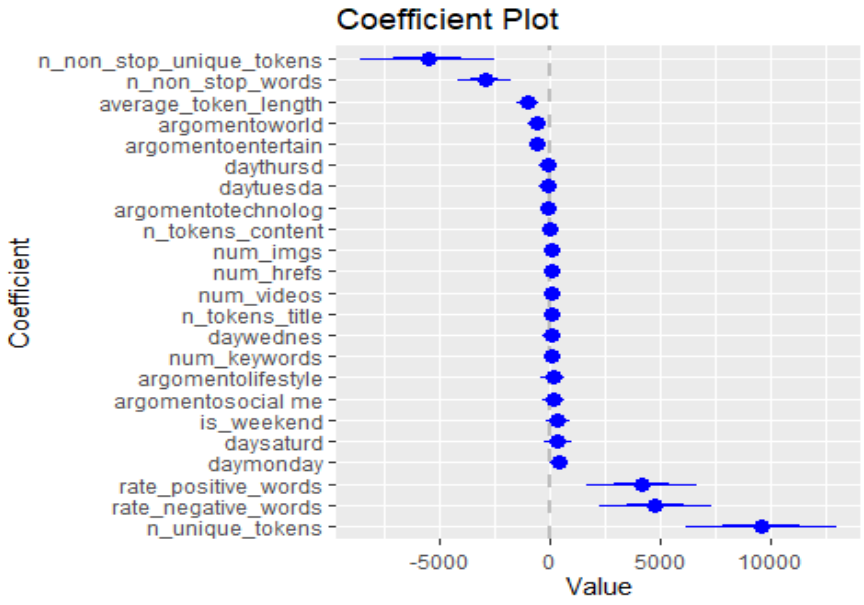
Le stime Bootstrap e i loro intervalli di confidenza confermano quanto osservato per le stime OLS del modello: i parametri sono tutti significativi ad eccezione di "argomento_lifestyle" e "n_non_stop_unique_tokens", poichè i loro intervalli di confidenza comprendono il valore nullo. Per "day_monday" e "n_tokens_content", per le quali la signficatività era bassa, si osserva che l'intervallo di confidenza Bootstrap non comprende lo zero, tuttavia ci si avvicina molto.

Per le variabili "num_imgs^2", "num_imgs^3", "num_videos^2" e "num_videos^3" invece si nota che, nonostante la stima MLE risulti molto significativa, l'intervallo di confidenza si avvicina molto al valore nullo (ma bisogna considerare che i loro parametri sono molto bassi e vicini a zero e che l'intervallo di confidenza si definisce intorno a quei valori).

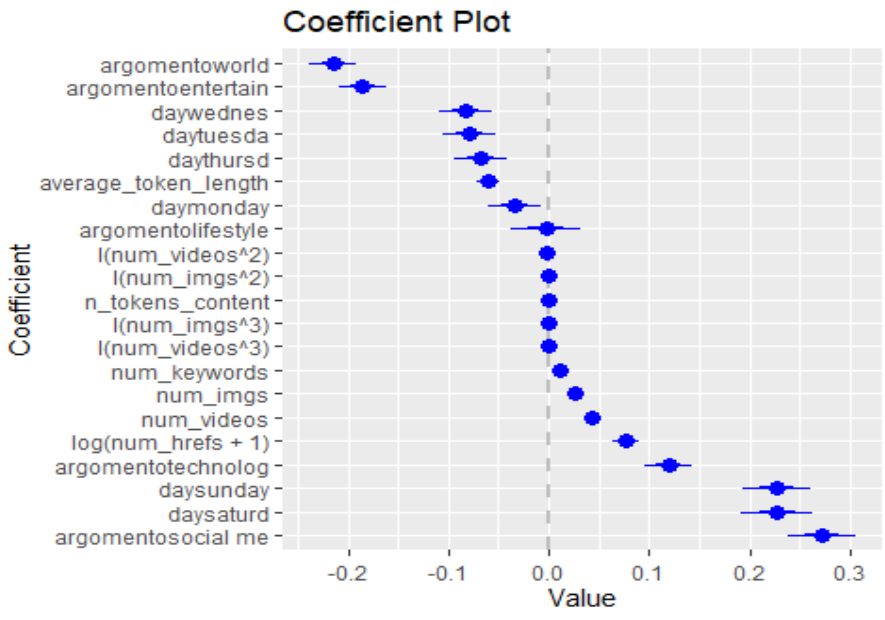
Confronto finale

Confronto dei coefficienti del modello iniziale e di quello robusto

```
library(coefplot)

## Warning: package 'coefplot' was built under R version 4.0.3

## Loading required package: ggplot2

coefplot(modello_base_completo, intercept=FALSE, decreasing = TRUE, sort = "magnit
ude")
```

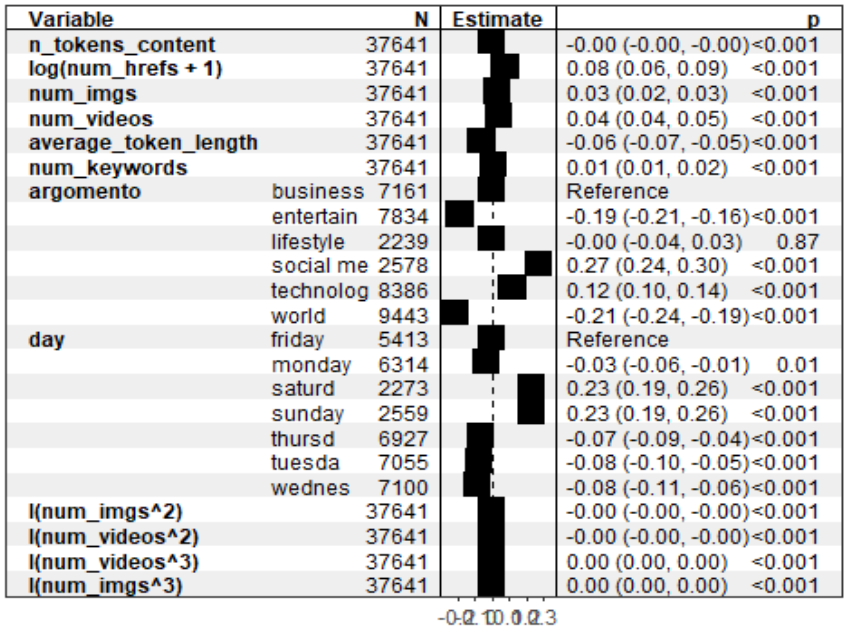



```
coefplot(modello_mod_sel2, intercept=FALSE, decreasing = TRUE, sort = "magnitude")
```



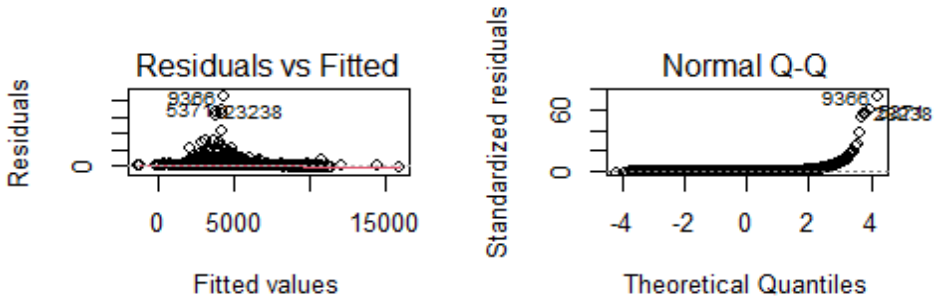
I coefficienti del modello robusto finale appaiono migliori di quelli del modello completo iniziale, in quanto una maggior quota di essi è significativa.

```
library(forestmodel)

## Warning: package 'forestmodel' was built under R version 4.0.3

print(forest_model(modello_mod_sel2))
```

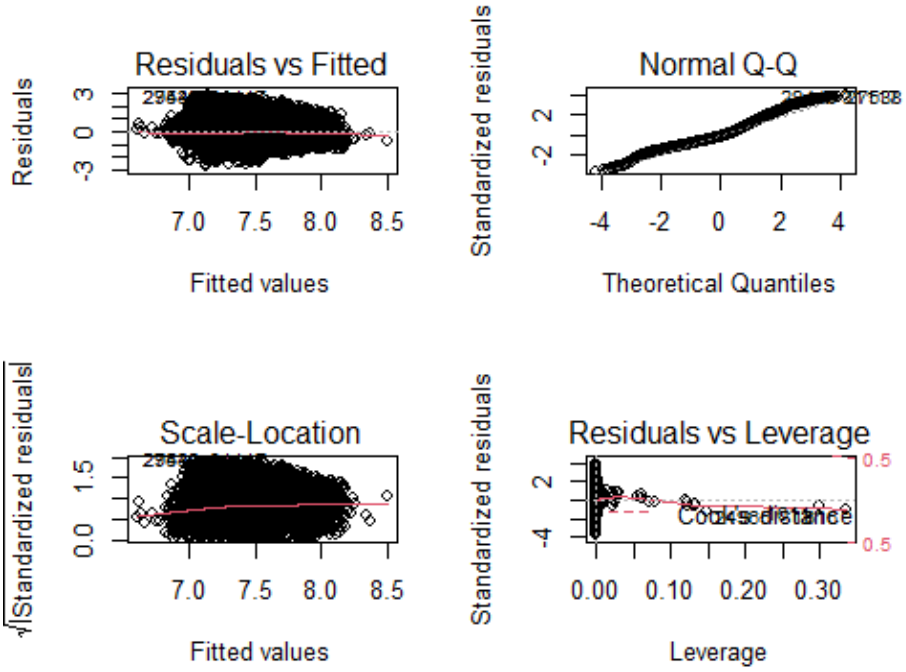


Confronto delle diagnostiche iniziali e finali

```
par(mfrow=c(2,2))
plot(modello_base_completo)
```



```r
plot(modello_mod_sel2)
```

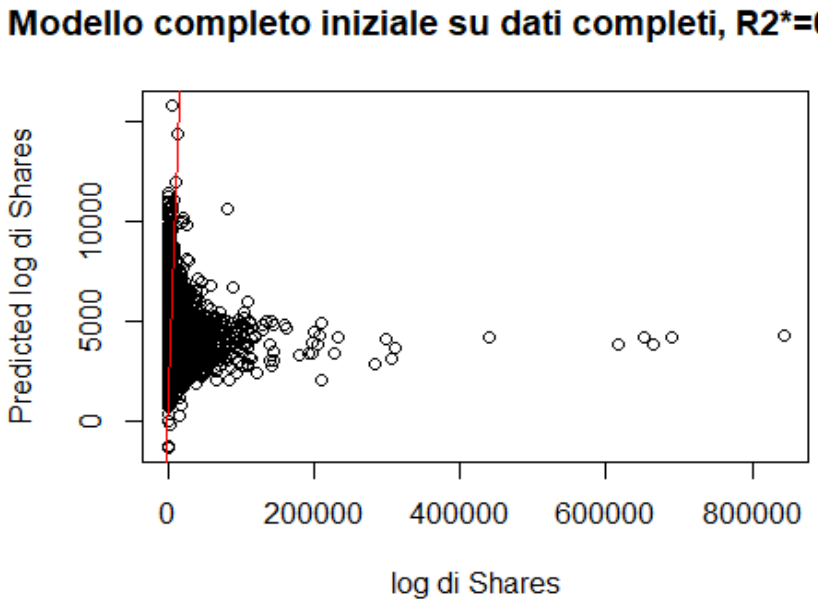



```r
par(mfrow=c(1,1))
```

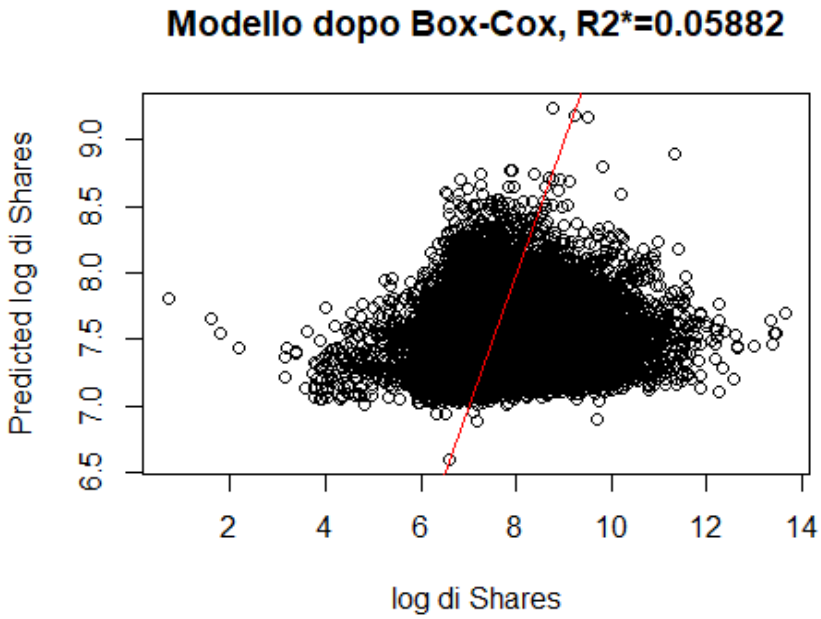
Si osserva un miglioramento delle diagnostiche finale rispetto a quelle di partenza. In particolare, nel plot "Residual vs Fitted" i residui appaiono più simmetrici e casuali intorno alla loro media 0; inoltre, la LOESS appare orizzontale. Il Q-Q plot mostra che i residui si distribuiscono maggiormente in maniera Normale. Dal plot "Scale-Location" si potrebbe intuire la persistenza dell'eteroschedasticità, in quanto i residui mostrano la tipica forma ad imbuto. Nel plot "Residual vs Leverage", invece, si nota che non vi sono più osservazioni oltre i limiti di influenza.

Confronto dei plot y osservati vs y stimati per i modelli ottenuti in ciascuno step

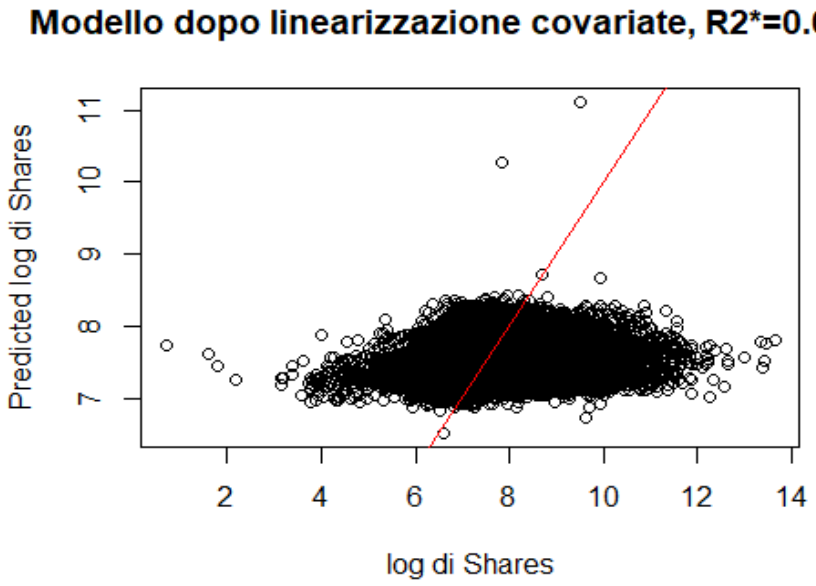
```r
plot(dati_completi$shares, modello_base_completo$fitted.values,
     main="Modello completo iniziale su dati completi, R2*=0.00653",
     xlab="log di Shares", ylab="Predicted log di Shares")
abline(0,1, col='red')
```



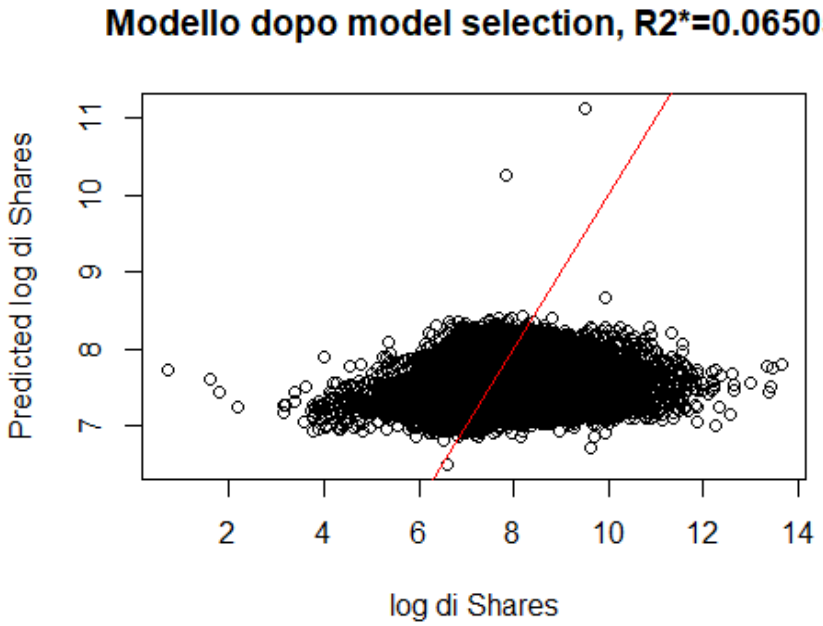

```
plot(log((dati_completi$shares)+1), modello_base5$fitted.values,
     main="Modello dopo Box-Cox, R2*=0.05882",
     xlab="log di Shares", ylab="Predicted log di Shares")
abline(0,1, col='red')
```




```
plot(log((dati_completi$shares)+1), modello_base6$fitted.values,
     main="Modello dopo linearizzazione covariate, R2*=0.06501",
     xlab="log di Shares", ylab="Predicted log di Shares")
abline(0,1, col='red')
```

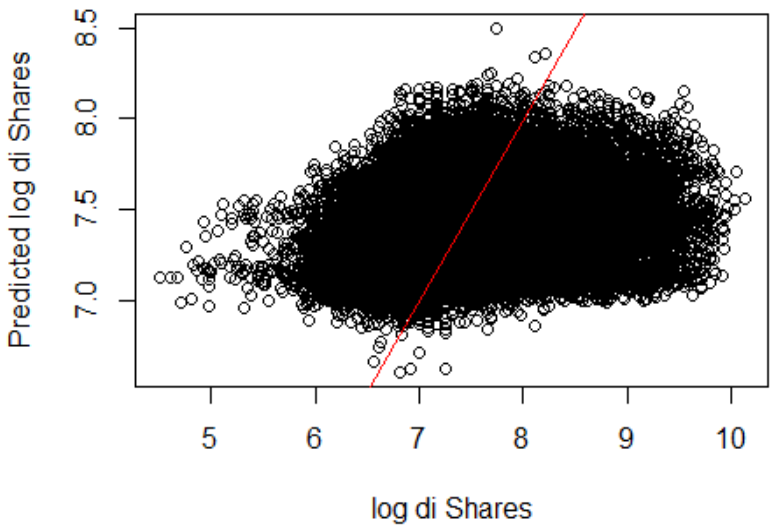



```r
plot(log((dati_completi$shares)+1), modello_mod_sel$fitted.values,
     main="Modello dopo model selection, R2*=0.06505",
     xlab="log di Shares", ylab="Predicted log di Shares")
abline(0,1, col='red')
```




```r
plot(log((NOinflu$shares)+1), modello_mod_sel2$fitted.values,
     main="Modello finale dopo eliminazione dei punti influenti, R2*=0.08445",
     xlab="log di Shares", ylab="Predicted log di Shares")
abline(0,1, col='red')
```

lello finale dopo eliminazione dei punti influenti, R2*=



Il modello di partenza descrive molto male i dati, come conferma anche l'indice R^2, e si nota la presenza di diversi punti che si allontanano dalla nuvola in cui si concentrano i dati.

Dopo la trasformazione della variabile target attraverso la procedura di Box e Cox, la distribuzione dei punti migliora notevolmente e appare casuale intorno alla bisettrice del quadrante.

Dopo il processo di linearizzazione delle covariate e la model selection, la nuvola di osservazioni si compatta ulteriormente anche se non sembra distribuirsi lungo la bisettrice.

Il modello finale, dopo l'eliminazione dei punti influenti, mostra il miglioramento maggiore rispetto a tutti gli altri passaggi. Con l'esclusione di questi punti problematici, i parametri del modello non subiscono più la loro influenza e quindi si specializzano sui dati rimanenti. Questo grafico non mostra un adattamento perfetto, tuttavia è un risultato coerente col fatto che il modello analizzato soffra ancora di eteroschedasticità. Nonostante complessivamente l'adattamento del modello sia aumentato di oltre 10 volte, rimane scarso.

Project work - serie storica

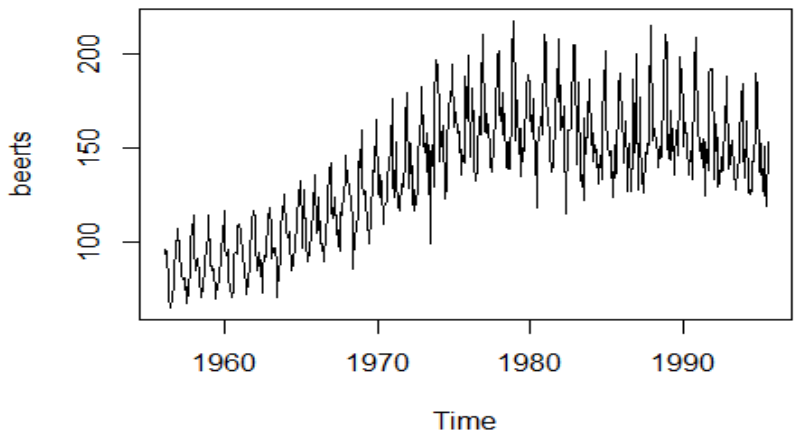
Kevin Capano 844018, Sara Licaj 846892, Susanna Maugeri 839365

Esame di Statistica Computazionale del 25 novembre 2020

La serie storica analizzata fa riferimento alla produzione mensile di birra in Austria, di cui sono stati raccolti i dati per il periodo che va da gennaio del 1956 ad agosto del 1995.

Importazione dei dati e plot

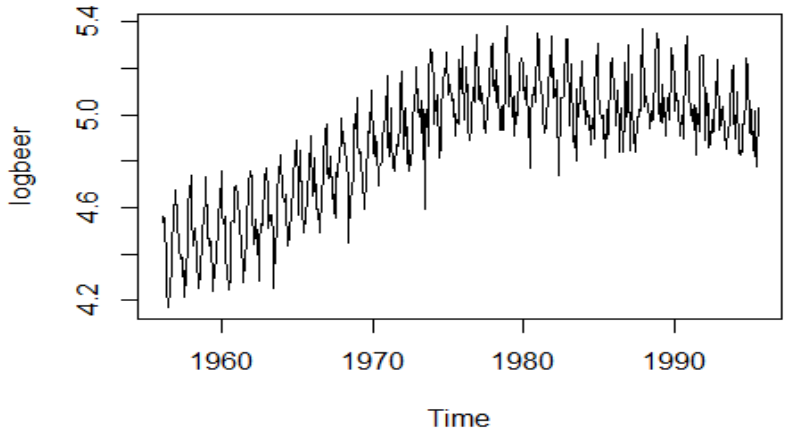
```
beer <- read.csv("beer.csv", header = TRUE, as.is = TRUE)
beerts <- ts(beer$Monthly.beer.production, start=(1956), frequency=12)
plot.ts(beerts)
```



La serie ha un andamento chiaramente stagionale e occorre stabilizzare la varianza che aumenta nel tempo. Inoltre, si osserva la presenza di un leggero trend crescente.

Stabilizzazione della varianza

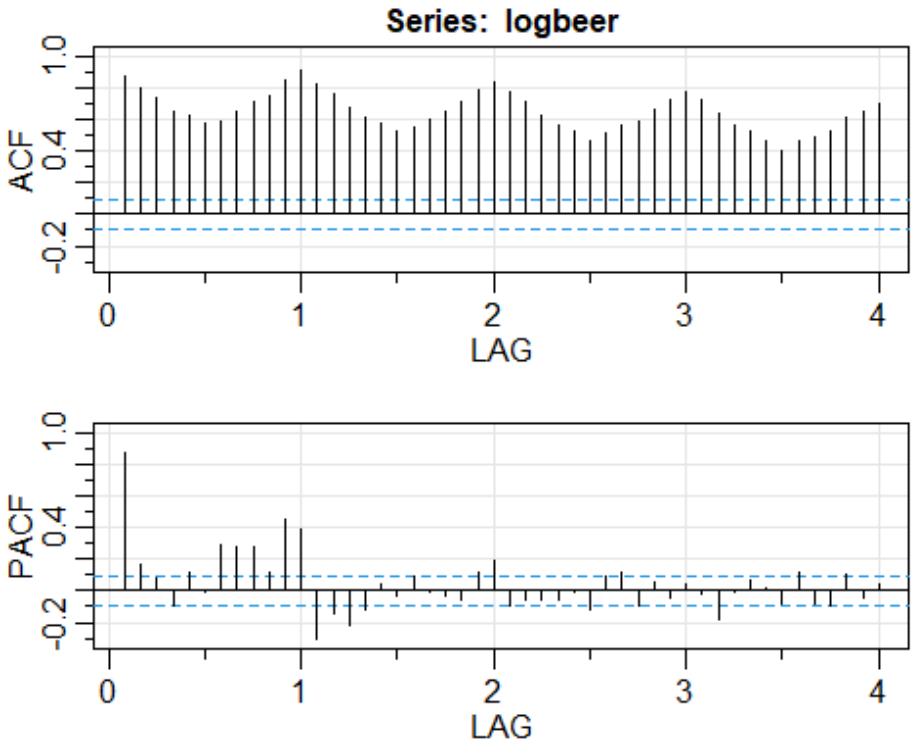
```
logbeer <- log(beerts)
plot.ts(logbeer)
```



Applicando una trasformazione logaritmica alla serie la varianza si stabilizza e non aumenta nel tempo.

Acf e pacf

```
library(astsa)

acf2(logbeer,48)
```



Dall'ACF rileviamo la stagionalità di ordine 12. I lag sui numeri interi sono annuali, si riferiscono alla correlazione della serie con la serie stessa 12, 24, 36 mesi prima, cioè per la sua componente stagionale. Gli spike presenti prima del primo valore unitario, invece, sono la struttura di autocorrelazione della serie nella sua componente non stagionale.

ACF:

- Seasonal: potrebbe essere MA(1) o MA(2) o MA(3) o superiore, poichè si osserva una correlazione molto forte anche dopo diversi anni;
- Non seasonal: potrebbe essere MA(1) o MA(2) o MA(3) o superiore, anche in questo caso vi è una correlazione molto forte.

Il grafico suggerisce che la stagionalità è molto forte. Potrebbe essere un modello arima stagionale.

PACF:

- Seasonal: potrebbe essere AR(1) o AR(2);

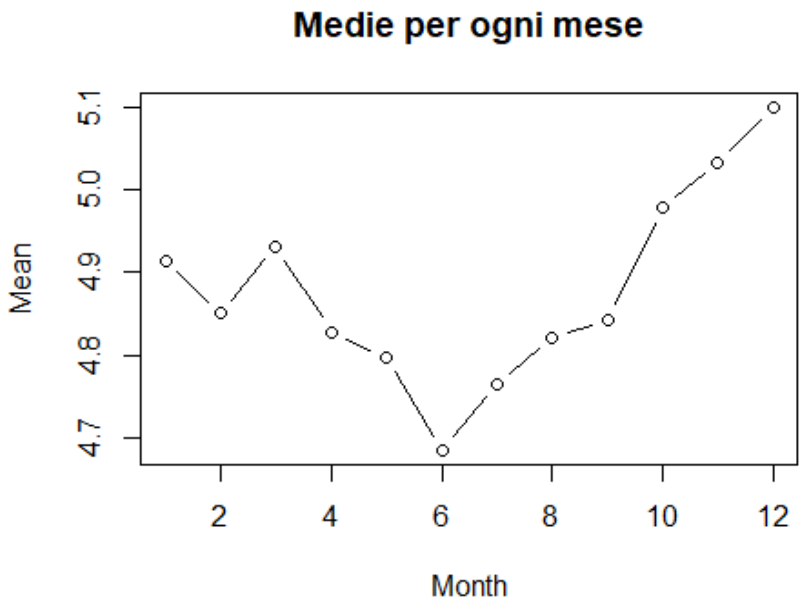
- Non seasonal: anche in questo caso potrebbe essere AR(1) o AR(2).

Si osservano correlazioni significative per la serie con 1 e 2 mesi prima. Questo suggerisce una struttura ARMA o ARIMA nella componente non stagionale.

Il modello potrebbe essere: ARIMA(2,0,3)(2,1,3)[12] o inferiore a causa dell'over-specification.

Intuizione del trend, grafico delle medie mensili negli anni, trend medio

```
flowm = matrix(logbeer, ncol=12, byrow=TRUE)

## Warning in matrix(logbeer, ncol = 12, byrow = TRUE): data length [476] is not a
## sub-multiple or multiple of the number of rows [40]

col.means=apply(flowm,2,mean)
plot(col.means, type="b", main="Medie per ogni mese", xlab="Month", ylab="Mean")
```



Nonostante non sia un'analisi molto fine, si può osservare che mediamente si produce più birra nei mesi invernali. In particolare, si nota un incremento nella produzione di birra in corrispondenza dell'Oktoberfest.

Rimozione della stagionalità

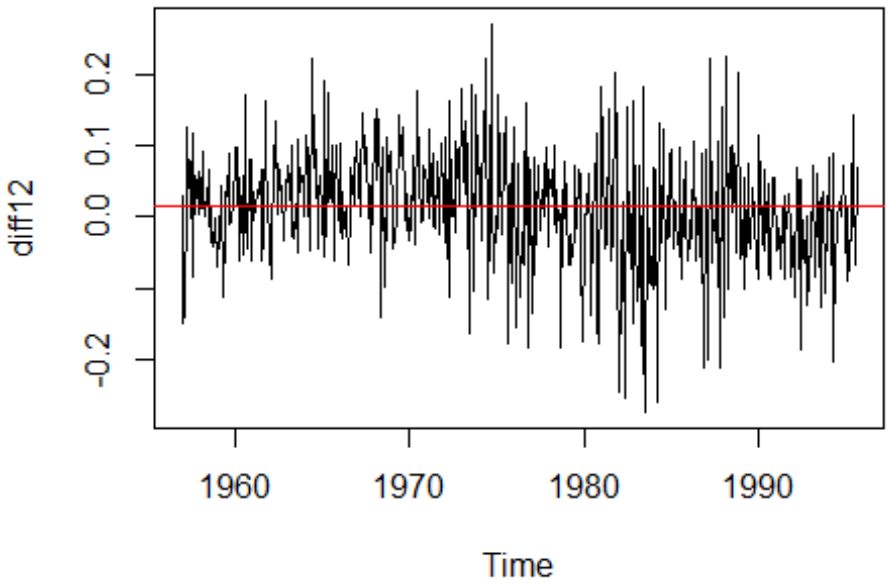
Quante differenze di ordine 12 occorre effettuare per ottenere una serie libera dalla componente stagionale?

```
library(forecast)

nsdiffs(logbeer)
```

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

Una sola.

```
diff12 = diff(logbeer, 12)
plot.ts(diff12)
abline(h=mean(diff12), col='red')
```



Dopo aver differenziato la serie appare più stazionaria, è opportuno chiedersi se sia seasonal stationary.

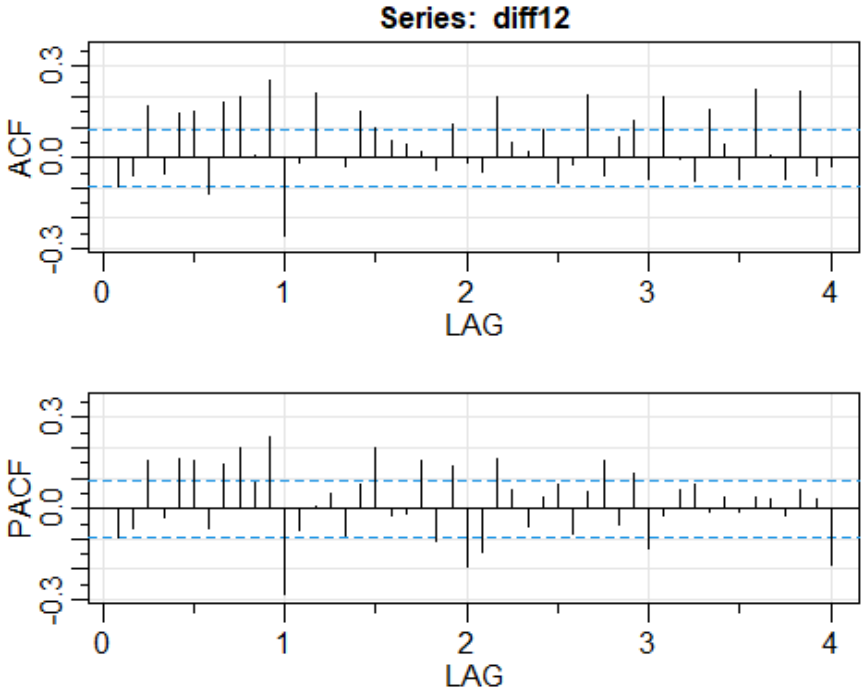
```
nsdiffs(diff12)

## [1] 0
```

Abbiamo rimosso la non stazionarietà dovuta alla componente stagionale, ovvero è seasonal stationary, tuttavia vi potrebbe essere un'altra fonte di non stazionarietà da rilevare; perciò verifichiamo che la serie risulti stazionaria.

```
ndiffs(diff12)

## [1] 1
```

La serie non è stazionaria, quindi vi è un'altra fonte di non stazionarietà che potrebbe essere un trend deterministico o una radice unitaria.

```
acf2(diff12,48)
```



ACF:

- Seasonal: potrebbe essere MA(1), poichè si osserva uno spike al primo lag;
- Non seasonal: si osserva un'alternanza di spike significativi e non, perciò consideriamo nonseasoal MA(5) o inferiore.

PACF:

- Seasonal: si osservano 4 spikes, uno per ogni valore unitario. Potrebbe essere un AR(4) o inferiore per overspecification;
- Non seasonal: si osserva un'alternanza di spike significativi e non per i primi valori: consideriamo nonseasoal AR(5) o inferiore.

I residui della componente stagionale potrebbero avere una struttura ARMA(4,1) oppure inferiore. La componente non stagionale,invece, potrebbe avere una struttura ARIMA(5,1,5) o inferiore.

Il modello verosimile per la serie originale potrebbe essere (5,1,5)(4,1,1)[12] o una struttura con ordini inferiori, a causa dell'over specification.

```
library(urca)

auto.arima(diff12)

## Series: diff12
## ARIMA(3,1,3)(2,0,1)[12] with drift
##
## Coefficients:
##          ar1     ar2     ar3      ma1      ma2     ma3    sar1     sar2
```

```
##       0.0678  0.3574  0.1622  -1.1566  -0.2818  0.4955  0.0970  -0.0980
## s.e.  0.1555  0.1150  0.0653   0.1502   0.2320  0.1097  0.0587   0.0542
##          sma1   drift
##       -0.8511  -1e-04
## s.e.   0.0380   1e-04
##
## sigma^2 estimated as 0.004437:  log likelihood=593.67
## AIC=-1165.34   AICc=-1164.75   BIC=-1119.82
```

La funzione auto.arima consiglia una struttura ARIMA(3,1,3)(2,0,1)[12] with drift per la serie differenziata per la stagionalità, dunque una struttura ARIMA(3,1,3)(2,1,1)[12] col drift per la serie originale.

Controllo della stazionarietà

Vengono applicati i test ADF e KPSS per capire, con il primo, se c'è una radice unitaria e con il secondo se c'è un trend stazionario o stocastico.

```
mean(diff12)
```

```
## [1] 0.01419873
```

Anche se la media è prossima a zero, la si considera diversa da zero per il test di stazionarietà.

Per il test ADF le ipotesi sono le seguenti:

H0: presenza di una radice unitaria

H1: assenza di radici unitarie

```
summary(ur.df(diff12, "trend", lags=12))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.289395 -0.042292  0.001521  0.048034  0.244227
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.914e-02  9.779e-03   2.980 0.003043 **
## z.lag.1      -6.952e-01  1.473e-01  -4.719 3.20e-06 ***
## tt           -8.265e-05  3.256e-05  -2.539 0.011472 *
```

```
## z.diff.lag1  -4.402e-01  1.451e-01  -3.033 0.002566 **
## z.diff.lag2  -5.486e-01  1.459e-01  -3.761 0.000192 ***
## z.diff.lag3  -4.403e-01  1.461e-01  -3.013 0.002735 **
## z.diff.lag4  -4.215e-01  1.449e-01  -2.909 0.003811 **
## z.diff.lag5  -3.391e-01  1.409e-01  -2.407 0.016520 *
## z.diff.lag6  -2.275e-01  1.347e-01  -1.689 0.091928 .
## z.diff.lag7  -2.329e-01  1.276e-01  -1.825 0.068661 .
## z.diff.lag8  -1.001e-01  1.182e-01  -0.846 0.397795
## z.diff.lag9   1.401e-01  1.051e-01   1.332 0.183509
## z.diff.lag10  2.224e-01  8.984e-02   2.476 0.013679 *
## z.diff.lag11  3.956e-01  6.879e-02   5.752 1.67e-08 ***
## z.diff.lag12  9.289e-02  4.723e-02   1.967 0.049863 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07323 on 436 degrees of freedom
## Multiple R-squared:  0.6787, Adjusted R-squared:  0.6683
## F-statistic: 65.77 on 14 and 436 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -4.7186 7.452 11.1644
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
```

Il risultato del test non mostra la presenza di radici unitarie, nonostante la serie non sia stazionaria. Data questa evidenza si tratta dunque di una serie stazionaria attorno ad un trend deterministico. Anche l'intercetta ed il trend sono significativamente e congiuntamente diversi da 0.

Per il test KPSS le ipotesi sono le seguenti:

H0: variabile stazionaria

H1: variabile trend stazionaria (per type = "tau")

```
ndiffs(diff12)
```

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

```
kpss.test=ur.kpss(diff12, type = "tau")
summary(kpss.test)
```

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: tau with 5 lags.
##
```

```
## Value of test-statistic is: 0.2419
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

Poichè l'ipotesi nulla viene rifiutata, si ha la conferma che la serie sia trend stazionaria e che il trend sia deterministico.

Che tipo di trend deterministico è?

Verifichiamo se il trend è lineare, quadratico o cubico:

```
trend = seq(1:length(diff12))
trend2 <- trend*trend
trend3 <- trend*trend*trend
ttt <- cbind(trend, trend2, trend3)
auto.arima(diff12, xreg=ttt)
```

```
## Series: diff12
## Regression with ARIMA(5,0,0)(2,0,0)[12] errors
##
## Coefficients:
##           ar1      ar2     ar3      ar4     ar5     sar1     sar2  trend
##       -0.1093  -0.0832  0.2274  -0.0695  0.1259  -0.4677  -0.2538  7e-04
## s.e.   0.0462   0.0471  0.0484   0.0464  0.0463   0.0480   0.0464  0e+00
##       trend2  trend3
##            0       0
## s.e.       0       0
##
## sigma^2 estimated as 0.005428:  log likelihood=554.95
## AIC=-1087.91   AICc=-1087.32   BIC=-1042.37
```

Il trend è lineare, poichè il termine quadratico e quello cubico non vengono rilevati.

Auto.arima con l'aggiunta del trend lineare

Auto.arima sul modello originale:

```
library(forecast)
trend = seq(1:length(logbeer))
modello <- auto.arima(logbeer, xreg=trend, seasonal = TRUE)
summary(modello)
```

```
## Series: logbeer
## Regression with ARIMA(1,0,1)(0,0,2)[12] errors
##
## Coefficients:
##         ar1      ma1    sma1    sma2  intercept    xreg
##       0.777  -0.4080  0.5867  0.4114     4.5482  0.0014
```

```
## s.e.  0.058   0.0894  0.0476  0.0397     0.0442  0.0002
##
## sigma^2 estimated as 0.009334:  log likelihood=436.49
## AIC=-858.97   AICc=-858.73   BIC=-829.81
##
## Training set error measures:
##                        ME       RMSE        MAE         MPE   MAPE     MASE
## Training set 0.0006041835 0.09599935 0.07582833 -0.03650969 1.5506 1.125887
##                     ACF1
## Training set 0.01180701
```

L'auto.arima sul modello originale non rileva che la stagionalità sia fonte di non stazionarietà. Consiglia la struttura ARIMA(1,0,1)(0,0,2)[12] per i residui attorno al trend deterministico.

Questo è il primo dei due modelli competitivi.

Auto.arima sul modello differenziato:

```
trend = seq(1:length(diff12))
modello <- auto.arima(diff12, xreg=trend)
summary(modello)
```

```
## Series: diff12
## Regression with ARIMA(2,0,2)(0,0,1)[12] errors
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sma1  intercept    xreg
##       1.6725  -0.6777  -1.7871  0.8372  -0.8636     0.0404  -1e-04
## s.e.  0.0739   0.0738   0.0535  0.0500   0.0302     0.0100   1e-04
##
## sigma^2 estimated as 0.004566:  log likelihood=587.81
## AIC=-1159.62   AICc=-1159.3   BIC=-1126.5
##
## Training set error measures:
##                        ME       RMSE       MAE MPE MAPE      MASE         ACF1
## Training set 8.689793e-05 0.06706199 0.0522866 NaN  Inf 0.4993306 -0.006825379
```

L'auto.arima su diff12 consiglia una struttura ARIMA(2,0,2)(0,0,1)[12] per i residui attorno al trend deterministico, questo, secondo la nostra ipotesi, è il modello per la serie destagionalizzata e detrendizzata.

Questo è il secondo dei due modelli competitivi e il suo trend è 0.0404-0.0001t.

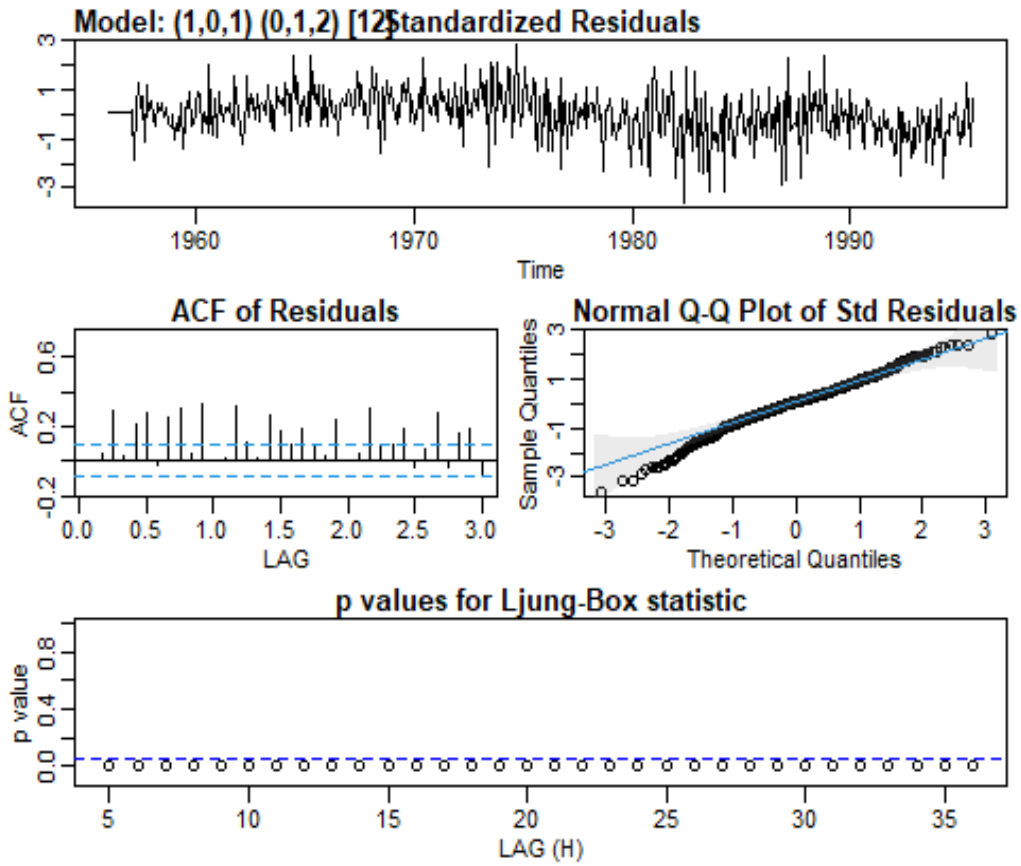
I due modelli competitivi sono:

Auto.arima: Regression with ARIMA(1,0,1)(0,1,2)[12] errors

Procedura standard: Regression with ARIMA(2,0,2)(0,1,1)[12] errors

```
library(astsa)
sarima(logbeer, 1,0,1,0,1,2,12)
```

```
## converged
```

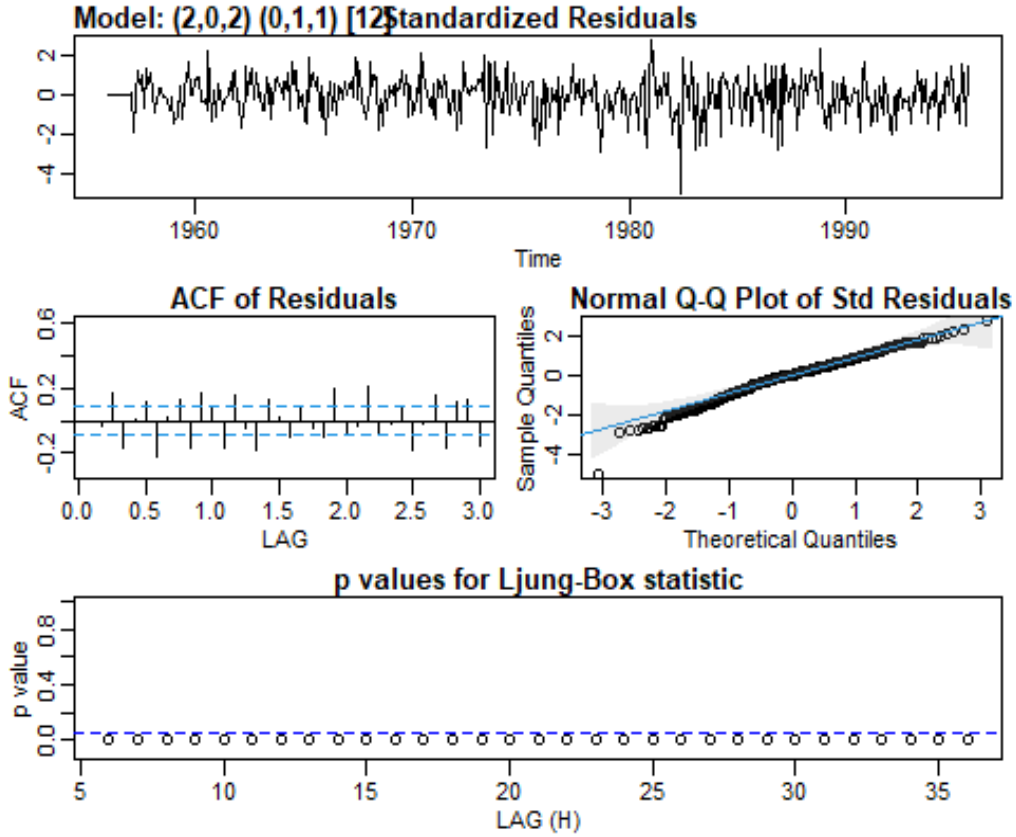



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): Si è prodotto un NaN

##           ar1     ma1     sma1     sma2  constant
##       -0.0018  0.0116  -0.3053  -0.0447    0.0012
## s.e.      NaN     NaN   0.0484   0.0412    0.0002
##
## sigma^2 estimated as 0.006706:  log likelihood = 502.04,  aic = -992.08
##
## $degrees_of_freedom
## [1] 459
##
## $ttable
##          Estimate     SE t.value p.value
## ar1       -0.0018    NaN     NaN     NaN
## ma1        0.0116    NaN     NaN     NaN
## sma1      -0.3053 0.0484 -6.3024  0.0000
## sma2      -0.0447 0.0412 -1.0854  0.2783
```

```
## constant   0.0012 0.0002  5.5738  0.0000
##
## $AIC
## [1] -2.088582
##
## $AICc
## [1] -2.088312
##
## $BIC
## [1] -2.036288
```

```
sarima(logbeer, 2,0,2,0,1,1,12)
```

```
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sma1  constant
##       1.6696  -0.6712  -1.7784  0.8293  -0.8544     0.001
## s.e.  0.0751   0.0749   0.0540  0.0501   0.0305     0.001
##
```

```
## sigma^2 estimated as 0.00454:  log likelihood = 585.8,  aic = -1157.59
##
## $degrees_of_freedom
## [1] 458
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        1.6696 0.0751  22.2450  0.0000
## ar2       -0.6712 0.0749  -8.9591  0.0000
## ma1       -1.7784 0.0540 -32.9531  0.0000
## ma2        0.8293 0.0501  16.5430  0.0000
## sma1      -0.8544 0.0305 -28.0007  0.0000
## constant   0.0010 0.0010   0.9993  0.3182
##
## $AIC
## [1] -2.437035
##
## $AICc
## [1] -2.436657
##
## $BIC
## [1] -2.376026
```

Il secondo modello sembra migliore per il BIC, ma i p-value di Ljung-Box sono tutti significativi. Questo significa che gli errori sono tra loro correlati e probabilmente vi è qualche errore di specificazione del modello.

Detrendizzazione della serie

logbeer: serie originale

diff12: serie destagionalizzata

trend deterministico di diff12: 0.0404 - 0.0001t

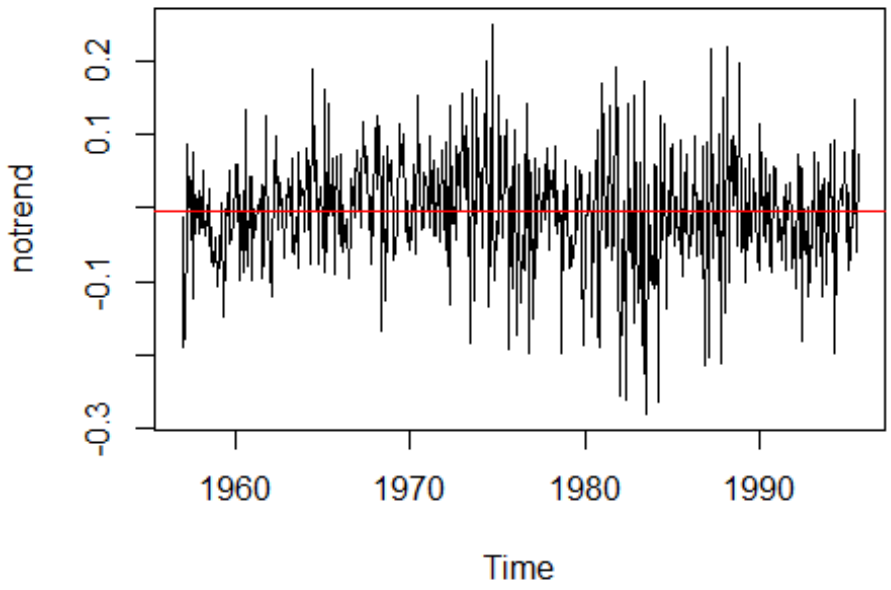
diff12 = 0.0404 - 0.0001t + vt

diff12 - 0.0404 + 0.0001t = vt, serie detrendizzata

Occorre capire la struttura ARMA di vt.

Si definisce la serie detrendizzata e se ne chiede la struttura ARMA:

```
trend = seq(1:length(diff12))
notrend <- diff12 - 0.0404 + 0.0001*trend
plot.ts(notrend)
abline(h=mean(notrend), col='red')
```



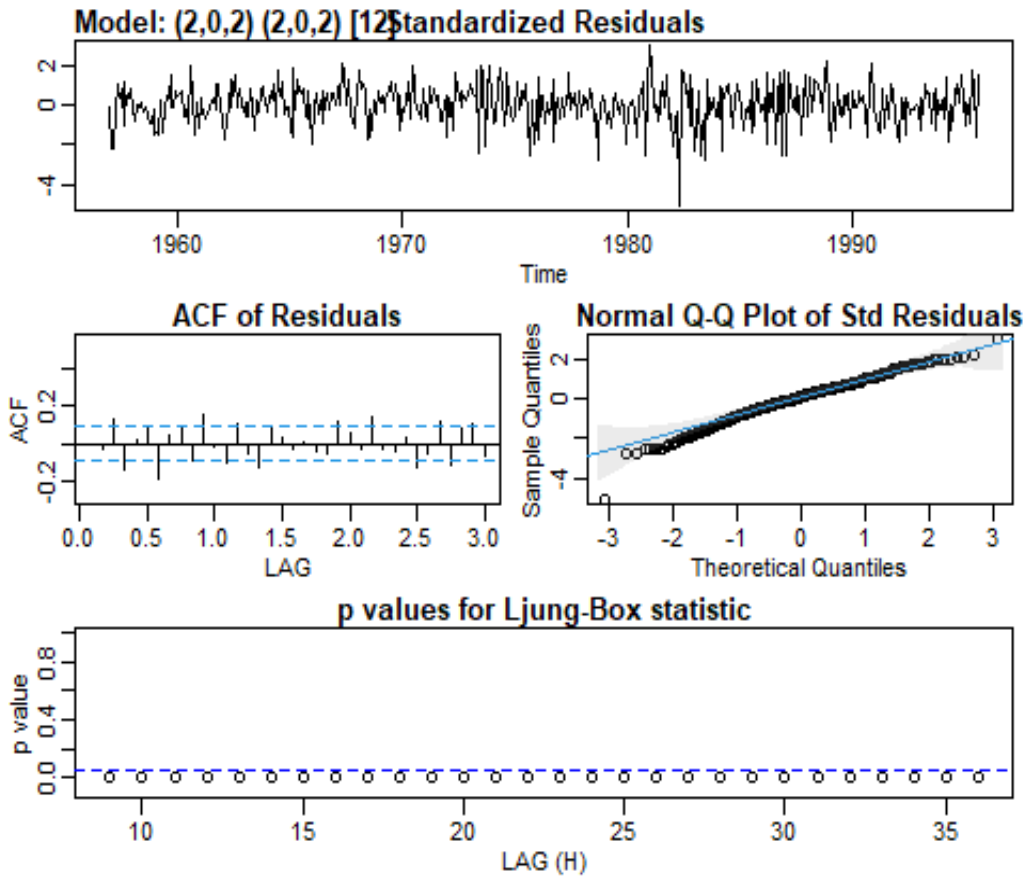

```
ndiffs(notrend)

## [1] 0
```

Notrend è la serie destagionalizzata e detrendizzata, come atteso è stazionaria intorno al valore 0.

```
fit2 <- auto.arima(notrend)
summary(fit2)

## Series: notrend
## ARIMA(2,0,2)(2,0,2)[12] with zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1     sar2     sma1    sma2
##       1.6629  -0.6665  -1.7538  0.8021  0.6079  -0.2492  -1.3603  0.4813
## s.e.  0.0903   0.0898   0.0674  0.0608  0.1246   0.0618   0.1206  0.1095
##
## sigma^2 estimated as 0.004421:  log likelihood=595.84
## AIC=-1173.67   AICc=-1173.28   BIC=-1136.41
##
## Training set error measures:
##                       ME       RMSE        MAE      MPE     MAPE      MASE
## Training set -0.001172563 0.06591756 0.05078319 77.18312 187.7213 0.4849967
##                      ACF1
## Training set -0.001280227
```

L'auto.arima su notrend consiglia una struttura ARIMA(2,0,2)(2,0,2)[12] with zero mean.

```
sarima(notrend, 2,0,2,2,0,2,12)

## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, transform.pars =
trans,
##     fixed = fixed, optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1     sar2     sma1    sma2
##       1.6604  -0.6642  -1.7525  0.8009  0.6077  -0.2498  -1.3600  0.4816
## s.e.  0.0905   0.0899   0.0676  0.0611  0.1242   0.0618   0.1201  0.1091
##         xmean
##       -0.0044
## s.e.   0.0070
##
## sigma^2 estimated as 0.004341:  log likelihood = 596.05,  aic = -1172.1
##
## $degrees_of_freedom
## [1] 455
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1     1.6604 0.0905  18.3498  0.0000
## ar2    -0.6642 0.0899  -7.3850  0.0000
```

```
## ma1    -1.7525 0.0676 -25.9169  0.0000
## ma2     0.8009 0.0611  13.1091  0.0000
## sar1    0.6077 0.1242   4.8948  0.0000
## sar2   -0.2498 0.0618  -4.0394  0.0001
## sma1   -1.3600 0.1201 -11.3209  0.0000
## sma2    0.4816 0.1091   4.4160  0.0000
## xmean  -0.0044 0.0070  -0.6347  0.5259
##
## $AIC
## [1] -2.526075
##
## $AICc
## [1] -2.52522
##
## $BIC
## [1] -2.436853
```

Nonostante la struttura (2,0,2)(2,0,2)[12] per la serie detrendizzata sia consigliata da auto.arima, la funzione sarima mostra che i p-value rimangono significativi.

Purtroppo i test non coincidono e non si può fare niente di più.

"None of the models considered here pass all of the residual tests. In practise, we would normally use the best model we could find, even if it did not pass all of the tests." (otexts.com/fpp3/seasonal-arima.html)

Confronto tra auto.arima su serie destagionalizzata e auto.arima su serie anche detrendizzata

Il BIC per il modello trovato con auto.arima sulla serie solo destagionalizzata è -1126.5, mentre quello per il modello trovato dall'auto.arima per la serie anche detrendizzata è -1136.41.

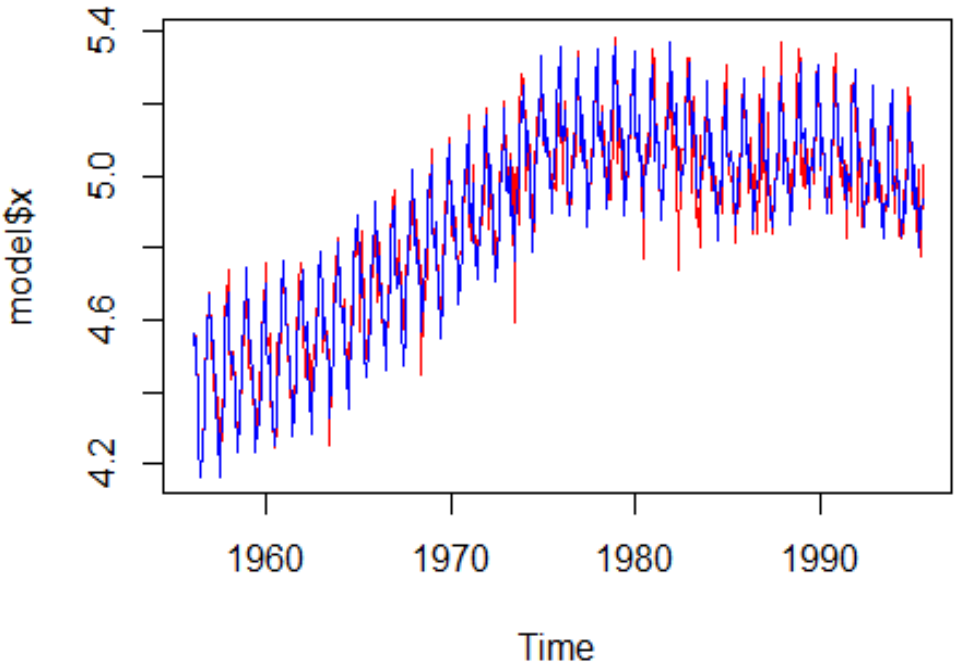
Dunque il secondo modello risulta migliore, avendo un BIC inferiore.

Plot di osservati e stimati

```
model <- Arima(logbeer, order=c(2,0,2),
            seasonal=list(order=c(2, 1, 2), period=12), include.drift = TRUE)
model
```

```
## Series: logbeer
## ARIMA(2,0,2)(2,1,2)[12] with drift
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1     sar2     sma1    sma2
##       1.6559  -0.6576  -1.7455  0.7951  0.6291  -0.2607  -1.3781  0.5067
## s.e.  0.0914   0.0911   0.0682  0.0615  0.1216   0.0600   0.1179  0.1042
##        drift
##       0.0009
## s.e.  0.0012
```

```
##
## sigma^2 estimated as 0.004449:  log likelihood=594.86
## AIC=-1169.72   AICc=-1169.24   BIC=-1128.32

plot(model$x, col='red')
lines(fitted(model), col='blue')
```

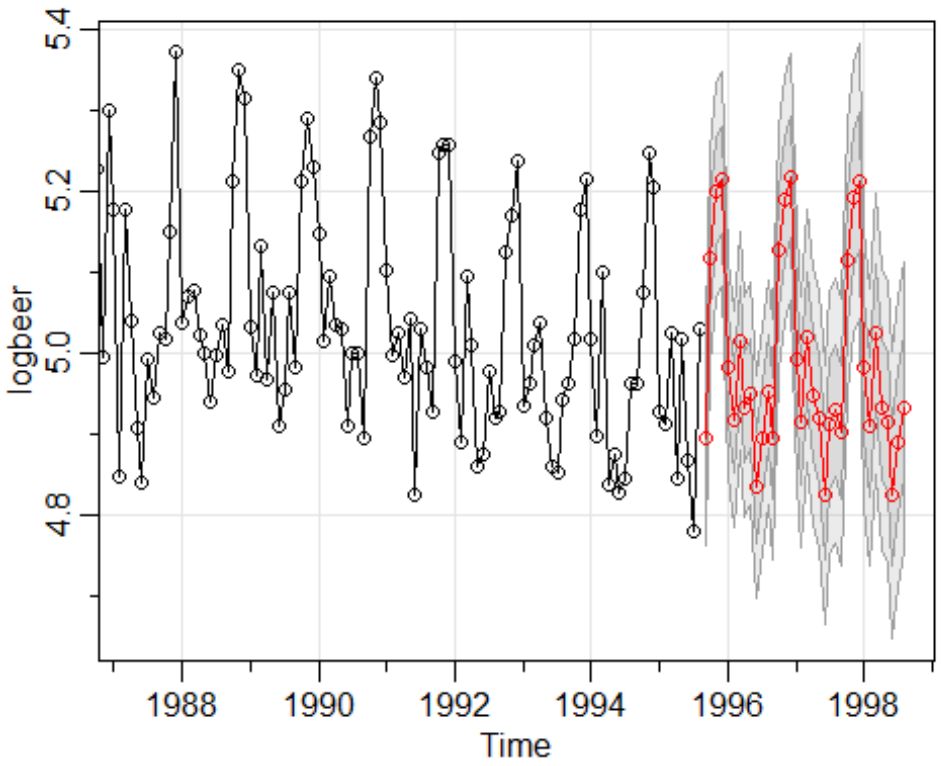


Nonostante il test per l'autocorrelazione dei residui di Ljung-Box non dia risultati confortanti, il modello sembra fittare molto bene i dati.

Forecast

Si considerano i tre anni successivi all'ultima osservazione.

```
sarima.for(logbeer, 36, 2,0,2,2,1,2,12, no.constant=FALSE)
```

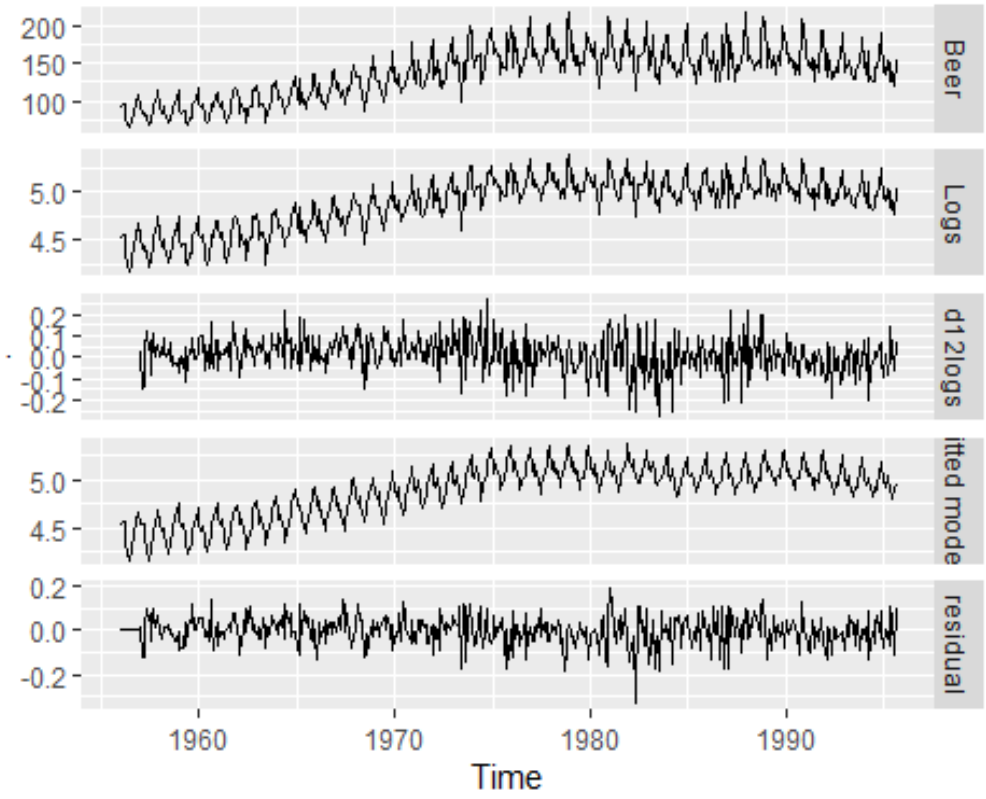



```
## $pred
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1995
## 1996 4.982193 4.918335 5.014965 4.933347 4.949871 4.834711 4.894135 4.951278
## 1997 4.993387 4.915509 5.020713 4.948039 4.919245 4.824853 4.911973 4.928963
## 1998 4.982270 4.908957 5.023734 4.931383 4.915084 4.824881 4.890993 4.933426
##           Sep      Oct      Nov      Dec
## 1995 4.894521 5.118246 5.199582 5.214248
## 1996 4.894422 5.128028 5.191026 5.217469
## 1997 4.901902 5.115349 5.192183 5.212186
## 1998
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 1995
## 1996 0.06655272 0.06686000 0.06726346 0.06773408 0.06824819 0.06878837
## 1997 0.07652185 0.07722413 0.07794728 0.07867931 0.07941234 0.08014125
## 1998 0.08520692 0.08587087 0.08652649 0.08717286 0.08780948 0.08843614
##             Jul        Aug        Sep        Oct        Nov        Dec
## 1995                       0.06604586 0.06631080 0.06631477 0.06636942
## 1996 0.06934246 0.06990226 0.07438900 0.07475517 0.07525805 0.07585812
## 1997 0.08086283 0.08157511 0.08251677 0.08318600 0.08386121 0.08453619
## 1998 0.08905279 0.08965947
```

La parte in rosso mostra le previsioni per il triennio succesivo all'ultima osservazione, con i relativi intervalli di confidenza che diventano più ampi al passare del tempo.

Riassunto degli step fondamentali

```
library(dplyr)

cbind("Beer" = beerts, "Logs" = logbeer, "d12logs" = diff12, "fitted
model"=fitted(model), "residual"=logbeer-fitted(model))  %>%
  autoplot(facets=TRUE)
```



Beer è la serie osservata originale, Logs è la sua trasformazione logaritmica e d12logs è la serie differenziata di ordine 12; si nota che con questo tipo di differenziazione si perde il primo anno di osservazioni. Fitted model è l'ipotesi di modello a cui si giunge tramite le varie analisi, mentre residual è la differenza tra il modello originale e quello fittato.

Nel complesso, nei vari passaggi si osserva un graduale miglioramento.