Prediction\_ta

18/04/2021

# Loading necessary packages   
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
library(tidytext)  
library(SnowballC)  
library(syuzhet)  
library(tidyr)  
library(qdap)

## Loading required package: qdapDictionaries

## Loading required package: qdapRegex

##   
## Attaching package: 'qdapRegex'

## The following object is masked from 'package:ggplot2':  
##   
## %+%

## The following object is masked from 'package:dplyr':  
##   
## explain

## Loading required package: qdapTools

##   
## Attaching package: 'qdapTools'

## The following object is masked from 'package:dplyr':  
##   
## id

## Loading required package: RColorBrewer

##   
## Attaching package: 'qdap'

## The following object is masked from 'package:tidyr':  
##   
## %>%

## The following object is masked from 'package:dplyr':  
##   
## %>%

## The following objects are masked from 'package:base':  
##   
## Filter, proportions

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ tibble 3.1.0 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1  
## ✓ purrr 0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x qdapRegex::%+%() masks ggplot2::%+%()  
## x forcats::%>%() masks stringr::%>%(), purrr::%>%(), tibble::%>%(), qdap::%>%(), tidyr::%>%(), dplyr::%>%()  
## x qdapRegex::explain() masks dplyr::explain()  
## x dplyr::filter() masks stats::filter()  
## x qdapTools::id() masks dplyr::id()  
## x dplyr::lag() masks stats::lag()

library(tokenizers)  
library(tm)

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:qdap':  
##   
## ngrams

## The following object is masked from 'package:ggplot2':  
##   
## annotate

##   
## Attaching package: 'tm'

## The following objects are masked from 'package:qdap':  
##   
## as.DocumentTermMatrix, as.TermDocumentMatrix

library(stringi)  
library(ggrepel)  
library(wordcloud)  
library(readr)  
library(smacof)

## Loading required package: plotrix

##   
## Attaching package: 'plotrix'

## The following object is masked from 'package:syuzhet':  
##   
## rescale

## Loading required package: colorspace

## Loading required package: e1071

##   
## Attaching package: 'smacof'

## The following object is masked from 'package:base':  
##   
## transform

library(ggfortify)  
library(ggthemes)  
library(stats)  
library(quanteda)

## Package version: 3.0.0  
## Unicode version: 10.0  
## ICU version: 61.1

## Parallel computing: 2 of 4 threads used.

## See https://quanteda.io for tutorials and examples.

##   
## Attaching package: 'quanteda'

## The following object is masked from 'package:tm':  
##   
## stopwords

## The following objects are masked from 'package:NLP':  
##   
## meta, meta<-

## The following object is masked from 'package:qdap':  
##   
## %>%

library(magrittr)

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:qdap':  
##   
## %>%

## The following object is masked from 'package:tidyr':  
##   
## extract

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(syuzhet)  
library(RDRPOSTagger)

## Loading required package: rJava

library(data.table)

##   
## Attaching package: 'data.table'

## The following object is masked from 'package:purrr':  
##   
## transpose

## The following object is masked from 'package:qdapTools':  
##   
## shift

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

library(gtools)

##   
## Attaching package: 'gtools'

## The following object is masked from 'package:e1071':  
##   
## permutations

library(combinat)

##   
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':  
##   
## combn

library(NMF)

## Loading required package: pkgmaker

## Loading required package: registry

## Registered S3 methods overwritten by 'registry':  
## method from   
## print.registry\_field proxy  
## print.registry\_entry proxy

## Loading required package: rngtools

## Loading required package: cluster

## NMF - BioConductor layer [OK] | Shared memory capabilities [NO: bigmemory] | Cores 3/4

## To enable shared memory capabilities, try: install.extras('  
## NMF  
## ')

##   
## Attaching package: 'NMF'

## The following object is masked from 'package:plotrix':  
##   
## dispersion

library(tidyr)

# Import and clean the data set   
wd <- "~/Documents/R"  
reviews\_df <- read\_csv("~/Desktop/Text-Analytics-\_files/Womens Clothing E-Commerce Reviews.csv")

## Warning: Missing column names filled in: 'X1' [1]

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## X1 = col\_double(),  
## `Clothing ID` = col\_double(),  
## Age = col\_double(),  
## Title = col\_character(),  
## `Review Text` = col\_character(),  
## Rating = col\_double(),  
## `Recommended IND` = col\_double(),  
## `Positive Feedback Count` = col\_double(),  
## `Division Name` = col\_character(),  
## `Department Name` = col\_character(),  
## `Class Name` = col\_character()  
## )

names(reviews\_df)[5] <- "reviewtext"  
reviews\_df$Title[is.na(reviews\_df$Title)] <- " "  
reviews\_df$reviewtext[is.na(reviews\_df$reviewtext)] <- " "  
reviews\_df$text\_full <- paste(reviews\_df$Title, reviews\_df$reviewtext, sep = " ")  
reviews\_df <- reviews\_df[!(reviews\_df$reviewtext == " "), ]  
  
reviews\_df$Rating\_b <- "happy"  
reviews\_df[reviews\_df$Rating < 4,]$Rating\_b <- "unhappy"  
  
# Remove punctuation  
reviews\_df$Review <- as.character(reviews\_df$text\_full) %>%   
 tolower() %>%   
 {gsub(":( |-|o)\*\\("," SADSMILE ", .)} %>% # Find :( or :-( or : ( or :o(  
 {gsub(":( |-|o)\*\\)"," HAPPYSMILE ", .)} %>% # Find :) or :-) or : ) or :o)  
 {gsub("(\"| |\\$)-+\\.-+"," NUMBER", .)} %>% # Find numbers  
 {gsub("([0-9]+:)\*[0-9]+ \*am"," TIME\_AM", .)} %>% # Find time AM  
 {gsub("([0-9]+:)\*[0-9]+ \*pm"," TIME\_PM", .)} %>% # Find time PM  
 {gsub("-+:-+","TIME", .)} %>% # Find general time  
 {gsub("\\$ ?[0-9]\*[\\.,]\*[0-9]+"," DOLLARVALUE ", .)} %>% # Find Dollar values  
 {gsub("[0-9]\*[\\.,]\*[0-9]+"," NUMBER ", .)} %>% # Find remaining numbers  
 {gsub("-"," ", .)} %>% # Remove all -  
 {gsub("&"," and ", .)} %>% # Find general time  
 {gsub("\"+"," ", .)} %>% # Remove all "  
 {gsub("\\|+"," ", .)} %>% # Remove all |  
 {gsub("\_+"," ", .)} %>% # Remove all \_  
 {gsub(";+"," ", .)} %>% # Remove excess ;  
 {gsub(" +"," ", .)} %>% # Remove excess spaces  
 {gsub("\\.+","\\.", .)} %>% # Remove excess .  
 {gsub("\'\"+"," ", .)} # Remove all '  
  
reviews\_df\_raw <- reviews\_df  
  
reviews\_df <- reviews\_df\_raw[,c(1,5,6,13)]  
names(reviews\_df)[1] <- "User\_ID"  
reviews\_df[,1] <- reviews\_df[,1]+1  
names(reviews\_df)[3] <- "rating"  
  
# Remove stop words   
ignorelist = stop\_words %>% filter(!word %in% c("no", "not", "never"))  
  
for (j in 1:nrow(reviews\_df)) {  
   
 words <- reviews\_df[j,] %>%   
 unnest\_tokens(word, reviewtext) %>%   
 anti\_join(ignorelist, by="word")  
   
 stemmed <- wordStem(words$word, language = "porter")  
 reviews\_df[j, "stemmed\_reviewtext\_with\_no"] <- paste(stemmed, collapse = " ")  
   
 # Again, but with ignoring all stopwords  
 nostopwords <- reviews\_df[j,] %>% unnest\_tokens(word, reviewtext) %>%  
 anti\_join( stop\_words, by = "word")  
 stemmed <- wordStem(nostopwords$word, language = "porter")  
   
 # Add variables to data  
 reviews\_df[j, "stemmed\_reviewtext"] <- paste(stemmed, collapse = " ")  
 reviews\_df[j, "reviewtext"] <- paste((nostopwords$word), collapse = " ")  
 reviews\_df[j, "Nr\_of\_words"]<- nrow(nostopwords)  
}  
print("done")

## [1] "done"

# Bigram set ----  
all\_bigrams <- reviews\_df[,c("User\_ID", "stemmed\_reviewtext\_with\_no")] %>%   
 unnest\_tokens(bigram, stemmed\_reviewtext\_with\_no, token = "ngrams", n = 2 )  
#This ignores sentences within a review.. could be improved.  
  
head(all\_bigrams)

## # A tibble: 6 x 2  
## User\_ID bigram   
## <dbl> <chr>   
## 1 1 absolut wonder  
## 2 1 wonder silki   
## 3 1 silki sexi   
## 4 1 sexi comfort   
## 5 2 love dress   
## 6 2 dress sooo

all\_bigrams <- all\_bigrams %>% dplyr::count(bigram, sort = TRUE)  
all\_bigrams[1:20,]

## # A tibble: 20 x 2  
## bigram n  
## <chr> <int>  
## 1 true size 1352  
## 2 fit perfectli 1104  
## 3 love dress 985  
## 4 love top 694  
## 5 fit true 538  
## 6 love color 527  
## 7 usual size 490  
## 8 size fit 479  
## 9 fit perfect 478  
## 10 dress fit 439  
## 11 wear size 414  
## 12 size 4 413  
## 13 size 6 407  
## 14 love love 394  
## 15 highli recommend 385  
## 16 fabric soft 368  
## 17 super cute 363  
## 18 medium fit 360  
## 19 absolut love 356  
## 20 skinni jean 350

sel\_bigrams <- all\_bigrams %>% filter(n>200)  
sel\_bigrams

## # A tibble: 71 x 2  
## bigram n  
## <chr> <int>  
## 1 true size 1352  
## 2 fit perfectli 1104  
## 3 love dress 985  
## 4 love top 694  
## 5 fit true 538  
## 6 love color 527  
## 7 usual size 490  
## 8 size fit 479  
## 9 fit perfect 478  
## 10 dress fit 439  
## # … with 61 more rows

# Analyze bi-grams  
bigrams\_sep <- separate(all\_bigrams, bigram, c("word1", "word2"), sep = " ")  
bigrams\_sep[1:20,]

## # A tibble: 20 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 true size 1352  
## 2 fit perfectli 1104  
## 3 love dress 985  
## 4 love top 694  
## 5 fit true 538  
## 6 love color 527  
## 7 usual size 490  
## 8 size fit 479  
## 9 fit perfect 478  
## 10 dress fit 439  
## 11 wear size 414  
## 12 size 4 413  
## 13 size 6 407  
## 14 love love 394  
## 15 highli recommend 385  
## 16 fabric soft 368  
## 17 super cute 363  
## 18 medium fit 360  
## 19 absolut love 356  
## 20 skinni jean 350

# Look at bigrams where first word = ...  
bigrams\_sep %>% filter(word1 == "no") %>% top\_n(10, n)

## # A tibble: 10 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 no shape 69  
## 2 no except 51  
## 3 no issu 50  
## 4 no stretch 49  
## 5 no matter 30  
## 6 no petit 29  
## 7 no idea 27  
## 8 no line 27  
## 9 no wear 23  
## 10 no zipper 22

bigrams\_sep %>% filter(word1 == "never") %>% top\_n(10, n)

## # A tibble: 12 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 never wear 32  
## 2 never bui 19  
## 3 never purchas 15  
## 4 never size 13  
## 5 never disappoint 12  
## 6 never worn 11  
## 7 never bought 10  
## 8 never write 10  
## 9 never own 7  
## 10 never fit 6  
## 11 never look 6  
## 12 never receiv 6

bigrams\_sep %>% filter(word1 == "not") %>% top\_n(10, n)

## # A tibble: 10 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not flatter 345  
## 2 not fit 274  
## 3 not tight 210  
## 4 not short 175  
## 5 not wear 169  
## 6 not worth 131  
## 7 not disappoint 108  
## 8 not dress 103  
## 9 not sheer 102  
## 10 not size 102

# Select infrequent and very frequent words to remove from review text ----   
  
# Get word frequency after stemming  
frequency <- reviews\_df %>% unnest\_tokens(word, stemmed\_reviewtext) %>% dplyr::count(word, sort=TRUE)  
  
# Select very frequent or infrequent words  
infrequent <- frequency %>% filter(n < 0.01\*nrow(reviews\_df))  
frequent <- frequency %>% filter(word %in% c("dress")) # you can extend this list with word you want to remove  
toremove <- full\_join(frequent, infrequent, by = "word") # combining these word lists  
frequency

## # A tibble: 10,073 x 2  
## word n  
## <chr> <int>  
## 1 dress 12179  
## 2 fit 11509  
## 3 love 11391  
## 4 size 10716  
## 5 top 8364  
## 6 wear 8075  
## 7 color 7299  
## 8 fabric 4890  
## 9 perfect 3852  
## 10 nice 3819  
## # … with 10,063 more rows

toremove

## # A tibble: 9,722 x 3  
## word n.x n.y  
## <chr> <int> <int>  
## 1 dress 12179 NA  
## 2 peopl NA 226  
## 3 incredibli NA 225  
## 4 structur NA 225  
## 5 complaint NA 224  
## 6 match NA 224  
## 7 odd NA 224  
## 8 sexi NA 224  
## 9 awesom NA 223  
## 10 think NA 223  
## # … with 9,712 more rows

# Remove common words from stemmed reviewtext  
  
for (j in 1:nrow(reviews\_df))   
{  
 tmp <- anti\_join( (reviews\_df[j,] %>% unnest\_tokens(word, stemmed\_reviewtext) ), toremove, by = "word")   
   
 reviews\_df$stemmed\_reviewtext[j] <- paste(tmp$word[-1], collapse = " ")  
}  
  
head(reviews\_df)

## # A tibble: 6 x 7  
## User\_ID reviewtext rating Rating\_b stemmed\_reviewtext… stemmed\_reviewtext   
## <dbl> <chr> <dbl> <chr> <chr> <chr>   
## 1 1 absolutely wo… 4 happy absolut wonder sil… wonder comfort   
## 2 2 love dress so… 5 happy love dress sooo pr… pretti store glad …  
## 3 3 hopes dress i… 3 unhappy hope dress initi p… petit usual size f…  
## 4 4 love love lov… 5 happy love love love jum… love love jumpsuit…  
## 5 5 shirt flatter… 5 happy shirt flatter due … flatter due front …  
## 6 6 love tracy re… 2 unhappy love traci rees dr… petit 5 tall wear …  
## # … with 1 more variable: Nr\_of\_words <int>

#reviews\_df$stemmed\_reviewtext   
#save(reviews\_df, file="Saved\_reviews\_df.Rda")

# Get document term matrix ----   
  
reviews\_df$User\_ID <- as.character(reviews\_df$User\_ID) %>% as.factor()   
  
review\_dtm <- reviews\_df %>%   
 unnest\_tokens(word, stemmed\_reviewtext) %>%   
 dplyr::count(User\_ID, word, sort=TRUE) %>%   
 ungroup() %>%  
 cast\_dtm(User\_ID,word,n)  
  
#Get document term matrix for bi-grams   
  
review\_dtm\_bi <- reviews\_df %>%   
 unnest\_tokens(bigram, stemmed\_reviewtext\_with\_no, token = "ngrams", n = 2) %>%   
 filter(bigram %in% sel\_bigrams$bigram) %>%  
 dplyr::count(User\_ID, bigram, sort=TRUE)  
review\_dtm\_bi$User\_ID = as.character(review\_dtm\_bi$User\_ID)  
  
review\_dtm\_bi <- review\_dtm\_bi %>%   
 ungroup() %>%  
 cast\_dtm(User\_ID, bigram, n)

# Principal Component Analysis   
N\_factors <- 20  
pca\_results <- prcomp(review\_dtm, scale = FALSE, rank. = N\_factors) #get the 20 most important factors  
rawLoadings <- pca\_results$rotation[, 1:N\_factors] %\*% diag(pca\_results$sdev, N\_factors, N\_factors)  
rotated <- varimax(rawLoadings)  
  
pca\_results$rotation <- rotated$loadings  
pca\_results$x <- scale(pca\_results$x[,1:N\_factors]) %\*% rotated$rotmat

# Add the factors to the data frame  
lastcol <- ncol(reviews\_df)  
reviews\_df <- data.frame(reviews\_df, factor = pca\_results$x)  
colnames(reviews\_df)[(lastcol+1):(lastcol+N\_factors)] <- paste0("factor", 1:N\_factors)  
  
# Figure out which words load high on each factor  
factor\_labels <- NULL   
for (j in 1:N\_factors) {  
 aa<-abs(pca\_results$rotation[,j]) %>% sort(decreasing = TRUE)   
 factor\_labels <- rbind(factor\_labels, paste0(names(aa[1:8])))  
}  
factor\_labels

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]   
## [1,] "size" "fit" "run" "medium" "true" "waist" "6"   
## [2,] "top" "pant" "love" "bit" "sleev" "arm" "comfort"  
## [3,] "fit" "perfectli" "size" "loos" "materi" "perfect" "medium"   
## [4,] "love" "top" "nice" "fabric" "feel" "size" "store"   
## [5,] "wear" "perfect" "size" "shirt" "sweater" "bra" "length"   
## [6,] "color" "love" "blue" "fit" "nice" "beauti" "green"   
## [7,] "fabric" "medium" "skirt" "soft" "flatter" "qualiti" "love"   
## [8,] "petit" "x" "size" "regular" "fit" "length" "store"   
## [9,] "shirt" "top" "white" "cute" "sleev" "jean" "pant"   
## [10,] "jean" "pair" "leg" "size" "pant" "fit" "wear"   
## [11,] "sweater" "sleev" "soft" "top" "wear" "pant" "medium"   
## [12,] "perfect" "wear" "length" "comfort" "summer" "short" "medium"   
## [13,] "waist" "skirt" "size" "top" "hip" "medium" "short"   
## [14,] "nice" "perfect" "love" "medium" "pant" "materi" "fabric"   
## [15,] "soft" "comfort" "materi" "skirt" "nice" "super" "pant"   
## [16,] "flatter" "cut" "fabric" "fit" "run" "shape" "waist"   
## [17,] "short" "length" "pant" "waist" "sleev" "perfect" "petit"   
## [18,] "pant" "top" "sweater" "skirt" "retail" "black" "fabric"   
## [19,] "bit" "top" "run" "size" "sweater" "pant" "cute"   
## [20,] "fabric" "materi" "medium" "size" "skirt" "top" "soft"   
## [,8]   
## [1,] "4"   
## [2,] "skirt"   
## [3,] "nice"   
## [4,] "cute"   
## [5,] "underneath"  
## [6,] "sweater"   
## [7,] "feel"   
## [8,] "lb"   
## [9,] "perfect"   
## [10,] "skinni"   
## [11,] "bit"   
## [12,] "waist"   
## [13,] "line"   
## [14,] "top"   
## [15,] "top"   
## [16,] "skirt"   
## [17,] "skirt"   
## [18,] "soft"   
## [19,] "skirt"   
## [20,] "waist"

# Add indicators for 50 most common words ----  
  
counts <- colSums(as.matrix(review\_dtm)) %>% sort(decreasing=TRUE)  
  
lastcol <- ncol(reviews\_df)  
N\_words\_stored <- 50  
word\_labels <- (names(counts)[1:N\_words\_stored])  
reviews\_df <- data.frame(reviews\_df, words = as.matrix(review\_dtm[,word\_labels]))  
names(reviews\_df)[(lastcol+1):(lastcol+N\_words\_stored)] <- word\_labels  
  
# Add inferred emotions ----   
  
nrc\_emotions <- get\_nrc\_sentiment(reviews\_df$reviewtext)  
  
reviews\_df <- data.frame(reviews\_df, nrc\_emotions)

# Adding bigrams ----  
  
review\_dtm\_bi <- as.matrix(review\_dtm\_bi)  
reviews\_df <- cbind(reviews\_df, review\_dtm\_bi[match(rownames(reviews\_df), rownames(review\_dtm\_bi)),])  
reviews\_df[is.na(reviews\_df)] <- 0  
  
names(reviews\_df) <- gsub(" ", "\_", names(reviews\_df), fixed = TRUE)

# Preparation of the predictive features ----   
N\_factors <- 20  
N\_emotions <- 10   
N\_words\_stored <- 50  
N\_bigrams\_stored <- 71  
  
index <- 7  
factornames <- colnames(reviews\_df)[index:(index+N\_factors-1)]  
index <- index + N\_factors  
wordnames <- colnames(reviews\_df)[index:(index+N\_words\_stored-1)]  
index <- index + N\_words\_stored  
emotionnames <- colnames(reviews\_df)[index:(index+N\_emotions-1)]  
index <- index + N\_emotions  
bigramnames <- colnames(reviews\_df)[index:(index+N\_bigrams\_stored-1)]  
index <- index + N\_bigrams\_stored  
  
set.seed(1234)   
estimation\_sample <- sort(sample(1:nrow(reviews\_df), size = round(0.7\*nrow(reviews\_df))))  
test\_sample <- (1:nrow(reviews\_df))[-estimation\_sample]

# Prepare strings   
  
allFactors <- paste("(", paste(factornames,collapse=" + "), ")")  
allEmotions <- paste("(", paste(emotionnames,collapse=" + "), ")")  
allWords <- paste("(", paste(wordnames,collapse=" + "), ")")  
allBigrams <- paste("(", paste(bigramnames,collapse=" + "), ")")  
allWordsAndBigrams <- paste("(", paste(c(wordnames, bigramnames),collapse=" + "), ")")  
  
allFactors

## [1] "( Nr\_of\_words + factor1 + factor2 + factor3 + factor4 + factor5 + factor6 + factor7 + factor8 + factor9 + factor10 + factor11 + factor12 + factor13 + factor14 + factor15 + factor16 + factor17 + factor18 + factor19 )"

# Basic linear model (without interactions and variable selection), different features and combinations ----   
  
f <- paste("rating ~ Nr\_of\_words + ", allFactors, " + ", allEmotions, " + ", allWords , " + ", allBigrams)  
lm.all <- lm(f, data=reviews\_df[estimation\_sample,] )  
summary(lm.all)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.1952 -0.5102 0.3329 0.7722 2.2146   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.671545 0.392167 11.912 < 2e-16 \*\*\*  
## Nr\_of\_words -0.003040 0.001117 -2.722 0.006488 \*\*   
## factor1 0.755119 0.277636 2.720 0.006539 \*\*   
## factor2 0.101534 0.278395 0.365 0.715332   
## factor3 0.286698 0.301941 0.950 0.342373   
## factor4 0.014090 0.372903 0.038 0.969860   
## factor5 0.042731 0.208769 0.205 0.837824   
## factor6 -0.191465 0.143866 -1.331 0.183255   
## factor7 -0.295238 0.152021 -1.942 0.052145 .   
## factor8 0.088068 0.072530 1.214 0.224678   
## factor9 -0.077403 0.147950 -0.523 0.600864   
## factor10 -0.180130 0.086777 -2.076 0.037931 \*   
## factor11 0.063316 0.085149 0.744 0.457135   
## factor12 0.130214 0.100504 1.296 0.195128   
## factor13 0.168518 0.109394 1.540 0.123467   
## factor14 -0.197777 0.150814 -1.311 0.189743   
## factor15 0.040641 0.124765 0.326 0.744624   
## factor16 -0.087175 0.092048 -0.947 0.343625   
## factor17 0.007433 0.075824 0.098 0.921911   
## factor18 -0.018604 0.111334 -0.167 0.867289   
## factor19 0.129929 0.145377 0.894 0.371476   
## fall 0.089296 0.042532 2.099 0.035789 \*   
## anger 0.017364 0.017318 1.003 0.316055   
## anticipation -0.043510 0.010910 -3.988 6.69e-05 \*\*\*  
## disgust -0.489147 0.019154 -25.537 < 2e-16 \*\*\*  
## fear -0.012520 0.018848 -0.664 0.506529   
## joy 0.255436 0.010546 24.222 < 2e-16 \*\*\*  
## sadness -0.002341 0.013208 -0.177 0.859303   
## surprise 0.011082 0.015118 0.733 0.463561   
## trust 0.051570 0.010064 5.124 3.02e-07 \*\*\*  
## negative -0.005479 0.010776 -0.508 0.611175   
## factor20 -0.382653 0.185351 -2.064 0.038989 \*   
## fit -0.482010 0.444369 -1.085 0.278068   
## size -0.997060 0.393647 -2.533 0.011323 \*   
## wear -0.138949 0.336810 -0.413 0.679945   
## love -0.094591 0.631262 -0.150 0.880890   
## top 0.192370 0.464136 0.414 0.678537   
## color -0.268230 0.253401 -1.059 0.289836   
## fabric 0.803942 0.416220 1.932 0.053435 .   
## perfect 0.359861 0.292912 1.229 0.219254   
## nice -0.453861 0.320815 -1.415 0.157173   
## flatter -0.243817 0.230799 -1.056 0.290800   
## soft -0.040057 0.197951 -0.202 0.839641   
## comfort -0.031850 0.142058 -0.224 0.822602   
## bit 0.158166 0.267936 0.590 0.554990   
## materi -0.259867 0.232306 -1.119 0.263311   
## length 0.077841 0.130704 0.596 0.551480   
## shirt 0.066154 0.309937 0.213 0.830983   
## waist 0.072401 0.150001 0.483 0.629337   
## run -0.010162 0.104970 -0.097 0.922878   
## jean -0.081692 0.186669 -0.438 0.661663   
## cute 0.044801 0.063413 0.706 0.479891   
## petit 0.058399 0.100334 0.582 0.560541   
## short -0.028559 0.114660 -0.249 0.803307   
## beauti 0.045694 0.055027 0.830 0.406327   
## sweater -0.104522 0.198520 -0.527 0.598544   
## medium 0.023384 0.145144 0.161 0.872012   
## feel 0.044930 0.071071 0.632 0.527273   
## qualiti -0.010777 0.051935 -0.208 0.835621   
## x 0.076104 0.092661 0.821 0.411483   
## sleev -0.026225 0.072979 -0.359 0.719338   
## bought -0.024052 0.050430 -0.477 0.633421   
## retail 0.016836 0.057293 0.294 0.768877   
## skirt 0.082026 0.146092 0.561 0.574485   
## pretti 0.009014 0.038800 0.232 0.816290   
## style 0.090768 0.038585 2.352 0.018664 \*   
## review 0.013013 0.046604 0.279 0.780083   
## pant -0.085146 0.167248 -0.509 0.610688   
## cut 0.015537 0.059574 0.261 0.794244   
## store 0.034225 0.052588 0.651 0.515171   
## purchas -0.043627 0.039418 -1.107 0.268409   
## design 0.026115 0.036606 0.713 0.475599   
## tight -0.040132 0.039740 -1.010 0.312566   
## black -0.013443 0.042904 -0.313 0.754026   
## return 0.023236 0.041175 0.564 0.572546   
## true -0.063206 0.091589 -0.690 0.490140   
## pictur 0.034099 0.039533 0.863 0.388402   
## summer 0.031134 0.054128 0.575 0.565164   
## super -0.010891 0.060665 -0.180 0.857532   
## line -0.038197 0.041339 -0.924 0.355505   
## detail 0.002263 0.044809 0.050 0.959729   
## positive -0.015766 0.008545 -1.845 0.065045 .   
## dress\_dress -0.303303 0.085561 -3.545 0.000394 \*\*\*  
## bodi\_type 0.079221 0.081206 0.976 0.329301   
## size\_x 0.060499 0.091461 0.661 0.508323   
## size\_8 0.022834 0.076961 0.297 0.766707   
## size\_size -0.012488 0.071556 -0.175 0.861460   
## dress\_wear -0.187729 0.086121 -2.180 0.029286 \*   
## normal\_size -0.101033 0.077360 -1.306 0.191569   
## love\_top -0.044716 0.051344 -0.871 0.383822   
## petit\_size -0.071979 0.074934 -0.961 0.336789   
## size\_2 -0.021073 0.070120 -0.301 0.763779   
## love\_love -0.018378 0.053946 -0.341 0.733357   
## arm\_hole 0.132476 0.080566 1.644 0.100132   
## sale\_price 0.061080 0.086709 0.704 0.481181   
## dress\_not -0.026115 0.071640 -0.365 0.715465   
## not\_fit -0.065704 0.077619 -0.846 0.397289   
## size\_6 0.076235 0.066935 1.139 0.254748   
## regular\_size 0.078748 0.074211 1.061 0.288640   
## fit\_perfectli -0.007129 0.061501 -0.116 0.907721   
## dress\_fit 0.051597 0.060797 0.849 0.396077   
## beauti\_dress 0.046393 0.085943 0.540 0.589337   
## dress\_love 0.038428 0.077920 0.493 0.621895   
## dress\_beauti 0.142275 0.081394 1.748 0.080485 .   
## fit\_flatter -0.028634 0.090628 -0.316 0.752043   
## wear\_size -0.112307 0.067483 -1.664 0.096087 .   
## super\_cute -0.122784 0.077197 -1.591 0.111734   
## highli\_recommend -0.027226 0.067696 -0.402 0.687561   
## love\_fit -0.053871 0.076452 -0.705 0.481046   
## fabric\_soft -0.017620 0.073970 -0.238 0.811726   
## fit\_perfect -0.043825 0.063946 -0.685 0.493132   
## fit\_nice 0.084390 0.082804 1.019 0.308141   
## wear\_dress -0.088518 0.080567 -1.099 0.271920   
## length\_perfect 0.069600 0.083518 0.833 0.404661   
## size\_medium 0.007315 0.079687 0.092 0.926859   
## read\_review -0.026912 0.089969 -0.299 0.764847   
## fabric\_nice 0.008512 0.086777 0.098 0.921858   
## fit\_true 0.048455 0.075431 0.642 0.520639   
## true\_size -0.130121 0.080073 -1.625 0.104176   
## love\_dress -0.069215 0.041811 -1.655 0.097856 .   
## light\_weight 0.104760 0.093501 1.120 0.262553   
## size\_4 -0.064985 0.067506 -0.963 0.335734   
## fit\_not -0.180200 0.078550 -2.294 0.021800 \*   
## love\_sweater -0.128395 0.086201 -1.489 0.136381   
## super\_soft 0.001360 0.082183 0.017 0.986802   
## size\_fit 0.083205 0.061760 1.347 0.177923   
## not\_tight 0.118204 0.094102 1.256 0.209088   
## love\_color -0.034906 0.057607 -0.606 0.544566   
## size\_10 -0.045205 0.095406 -0.474 0.635640   
## top\_fit 0.130186 0.078101 1.667 0.095554 .   
## skinni\_jean -0.127389 0.081102 -1.571 0.116265   
## usual\_size -0.017846 0.062948 -0.284 0.776797   
## x\_fit 0.067243 0.080823 0.832 0.405437   
## absolut\_love 0.077118 0.071408 1.080 0.280179   
## medium\_fit 0.056724 0.074586 0.761 0.446956   
## dress\_run -0.018943 0.087783 -0.216 0.829150   
## top\_not -0.098794 0.093676 -1.055 0.291606   
## plan\_wear 0.044311 0.088095 0.503 0.614975   
## not\_flatter -0.075210 0.072867 -1.032 0.302015   
## materi\_soft 0.044426 0.081976 0.542 0.587870   
## color\_love 0.048307 0.084589 0.571 0.567956   
## fit\_tt 0.026299 0.082884 0.317 0.751025   
## soft\_comfort -0.065968 0.075407 -0.875 0.381682   
## color\_beauti -0.074851 0.087034 -0.860 0.389789   
## fell\_love -0.002350 0.087100 -0.027 0.978476   
## run\_true 0.013271 0.098303 0.135 0.892615   
## previou\_review -0.056607 0.093214 -0.607 0.543672   
## wait\_wear 0.049873 0.080465 0.620 0.535389   
## love\_shirt 0.196830 0.089789 2.192 0.028385 \*   
## run\_size 0.087376 0.080052 1.091 0.275076   
## receiv\_compliment -0.039538 0.083991 -0.471 0.637829   
## agre\_review -0.136919 0.095219 -1.438 0.150471   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.057 on 15697 degrees of freedom  
## Multiple R-squared: 0.1222, Adjusted R-squared: 0.1137   
## F-statistic: 14.47 on 151 and 15697 DF, p-value: < 2.2e-16

f <- paste("rating ~ Nr\_of\_words + ", allFactors, " + ", allWords , " + ", allBigrams)  
lm.nodict <- lm(f, data=reviews\_df[estimation\_sample,] )  
summary(lm.nodict)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8266 -0.4484 0.5010 0.8228 1.5726   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.9193994 0.4075893 12.070 < 2e-16 \*\*\*  
## Nr\_of\_words -0.0104166 0.0010513 -9.909 < 2e-16 \*\*\*  
## factor1 0.8540924 0.2903704 2.941 0.003272 \*\*   
## factor2 -0.0172084 0.2862290 -0.060 0.952060   
## factor3 0.2447513 0.3156749 0.775 0.438158   
## factor4 0.0105351 0.3882885 0.027 0.978355   
## factor5 0.1079186 0.2182509 0.494 0.620981   
## factor6 -0.2404397 0.1491445 -1.612 0.106955   
## factor7 -0.3188575 0.1588820 -2.007 0.044779 \*   
## factor8 0.1030532 0.0756757 1.362 0.173289   
## factor9 -0.0598522 0.1545726 -0.387 0.698605   
## factor10 -0.1886189 0.0907559 -2.078 0.037697 \*   
## factor11 0.0845814 0.0888103 0.952 0.340917   
## factor12 0.0395796 0.0944545 0.419 0.675198   
## factor13 0.1449245 0.1132436 1.280 0.200649   
## factor14 -0.2344701 0.1577149 -1.487 0.137122   
## factor15 0.0467784 0.1281510 0.365 0.715097   
## factor16 -0.0277792 0.0951123 -0.292 0.770239   
## factor17 -0.0351750 0.0762951 -0.461 0.644778   
## factor18 0.0029630 0.1163934 0.025 0.979691   
## factor19 0.2505625 0.1446414 1.732 0.083240 .   
## factor20 -0.3849572 0.1937246 -1.987 0.046925 \*   
## fit -0.4163801 0.4646667 -0.896 0.370222   
## size -1.1150387 0.4116978 -2.708 0.006768 \*\*   
## wear -0.2914219 0.3520364 -0.828 0.407786   
## love -0.1016087 0.6580856 -0.154 0.877296   
## top -0.0017515 0.4769149 -0.004 0.997070   
## color -0.3381461 0.2631373 -1.285 0.198792   
## fabric 0.8432528 0.4351419 1.938 0.052656 .   
## perfect 0.1943928 0.2859394 0.680 0.496616   
## nice -0.5761816 0.3309498 -1.741 0.081704 .   
## flatter -0.1384515 0.2390199 -0.579 0.562431   
## soft -0.0781425 0.2065886 -0.378 0.705249   
## comfort -0.0451626 0.1461775 -0.309 0.757358   
## bit 0.3965101 0.2686350 1.476 0.139959   
## materi -0.2778403 0.2428607 -1.144 0.252628   
## length 0.0120622 0.1273545 0.095 0.924544   
## shirt 0.0082646 0.3235355 0.026 0.979621   
## waist 0.0550069 0.1568247 0.351 0.725778   
## run 0.0068295 0.1079676 0.063 0.949564   
## jean -0.0848510 0.1944272 -0.436 0.662541   
## cute 0.0804863 0.0641150 1.255 0.209373   
## petit 0.0607607 0.1048113 0.580 0.562115   
## short -0.0911234 0.1183305 -0.770 0.441267   
## beauti 0.0391221 0.0563127 0.695 0.487235   
## sweater -0.1568560 0.2043367 -0.768 0.442715   
## medium -0.0686921 0.1499635 -0.458 0.646916   
## feel 0.0475049 0.0738489 0.643 0.520057   
## qualiti -0.0104843 0.0538845 -0.195 0.845732   
## x 0.0905975 0.0964909 0.939 0.347785   
## sleev -0.0399324 0.0742440 -0.538 0.590686   
## bought -0.0441808 0.0526746 -0.839 0.401622   
## retail -0.0083057 0.0597769 -0.139 0.889496   
## skirt 0.0381549 0.1502712 0.254 0.799571   
## pretti 0.0009054 0.0405669 0.022 0.982194   
## style 0.1092764 0.0402761 2.713 0.006671 \*\*   
## review 0.0101555 0.0487324 0.208 0.834925   
## pant -0.0556745 0.1733342 -0.321 0.748065   
## cut 0.0337878 0.0620574 0.544 0.586132   
## store 0.0277784 0.0546357 0.508 0.611159   
## purchas -0.0749550 0.0406188 -1.845 0.065009 .   
## design 0.0216282 0.0382602 0.565 0.571883   
## tight -0.0093894 0.0414552 -0.226 0.820820   
## black -0.0199752 0.0448516 -0.445 0.656064   
## return 0.0329538 0.0427918 0.770 0.441254   
## true -0.0755288 0.0957528 -0.789 0.430247   
## pictur 0.0369291 0.0412688 0.895 0.370885   
## summer -0.0026272 0.0547465 -0.048 0.961727   
## super 0.0065177 0.0624831 0.104 0.916923   
## line -0.0299261 0.0429585 -0.697 0.486046   
## detail -0.0155353 0.0466437 -0.333 0.739091   
## positive 0.1283031 0.0058542 21.916 < 2e-16 \*\*\*  
## dress\_dress -0.3002126 0.0894784 -3.355 0.000795 \*\*\*  
## bodi\_type 0.0802760 0.0848649 0.946 0.344200   
## size\_x -0.0038938 0.0956423 -0.041 0.967526   
## size\_8 -0.0335030 0.0803126 -0.417 0.676569   
## size\_size 0.0062058 0.0748313 0.083 0.933908   
## dress\_wear -0.1866071 0.0900619 -2.072 0.038283 \*   
## normal\_size -0.1119887 0.0808906 -1.384 0.166241   
## love\_top -0.0391206 0.0536895 -0.729 0.466229   
## petit\_size -0.0855617 0.0783619 -1.092 0.274903   
## size\_2 -0.0123561 0.0732908 -0.169 0.866121   
## love\_love -0.0407409 0.0564049 -0.722 0.470125   
## arm\_hole 0.1330908 0.0841685 1.581 0.113843   
## sale\_price 0.1184261 0.0905740 1.308 0.191060   
## dress\_not -0.0263455 0.0749090 -0.352 0.725068   
## not\_fit -0.0454275 0.0811780 -0.560 0.575758   
## size\_6 0.0761527 0.0699671 1.088 0.276432   
## regular\_size 0.0575009 0.0774328 0.743 0.457740   
## fit\_perfectli -0.0206650 0.0643075 -0.321 0.747951   
## dress\_fit 0.0963689 0.0635636 1.516 0.129514   
## beauti\_dress 0.0128535 0.0898677 0.143 0.886271   
## dress\_love 0.0671622 0.0814684 0.824 0.409727   
## dress\_beauti 0.1545210 0.0851115 1.816 0.069464 .   
## fit\_flatter 0.0236041 0.0947747 0.249 0.803322   
## wear\_size -0.0822408 0.0705742 -1.165 0.243912   
## super\_cute -0.1416616 0.0807439 -1.754 0.079372 .   
## highli\_recommend -0.0246073 0.0707529 -0.348 0.728001   
## love\_fit -0.0649832 0.0799510 -0.813 0.416352   
## fabric\_soft 0.0120475 0.0773663 0.156 0.876255   
## fit\_perfect -0.0156206 0.0668555 -0.234 0.815262   
## fit\_nice 0.1050954 0.0865929 1.214 0.224891   
## wear\_dress -0.0512447 0.0842585 -0.608 0.543074   
## length\_perfect 0.1021119 0.0873423 1.169 0.242381   
## size\_medium 0.0137780 0.0833421 0.165 0.868695   
## read\_review -0.0594669 0.0940897 -0.632 0.527380   
## fabric\_nice 0.0176573 0.0907436 0.195 0.845720   
## fit\_true 0.0308858 0.0788863 0.392 0.695416   
## true\_size -0.1317244 0.0837447 -1.573 0.115755   
## love\_dress -0.0530942 0.0437206 -1.214 0.224614   
## light\_weight 0.1114938 0.0967572 1.152 0.249213   
## size\_4 -0.0767675 0.0705885 -1.088 0.276817   
## fit\_not -0.1541657 0.0821570 -1.876 0.060608 .   
## love\_sweater -0.1552770 0.0901482 -1.722 0.085005 .   
## super\_soft -0.0085051 0.0859359 -0.099 0.921163   
## size\_fit 0.1000612 0.0645854 1.549 0.121334   
## not\_tight 0.1287415 0.0984057 1.308 0.190800   
## love\_color -0.0351538 0.0602311 -0.584 0.559465   
## size\_10 -0.0497024 0.0997251 -0.498 0.618213   
## top\_fit 0.1341338 0.0816771 1.642 0.100559   
## skinni\_jean -0.1067745 0.0847729 -1.260 0.207855   
## usual\_size -0.0147822 0.0658111 -0.225 0.822281   
## x\_fit 0.1148820 0.0845264 1.359 0.174126   
## absolut\_love 0.0797676 0.0746868 1.068 0.285524   
## medium\_fit 0.0735747 0.0780157 0.943 0.345657   
## dress\_run 0.0029044 0.0918196 0.032 0.974766   
## top\_not -0.1035764 0.0979678 -1.057 0.290414   
## plan\_wear 0.0693968 0.0921006 0.753 0.451167   
## not\_flatter -0.0894934 0.0762058 -1.174 0.240267   
## materi\_soft 0.0363648 0.0857223 0.424 0.671414   
## color\_love 0.0549897 0.0884489 0.622 0.534140   
## fit\_tt 0.0051284 0.0866133 0.059 0.952786   
## soft\_comfort -0.0671427 0.0788383 -0.852 0.394420   
## color\_beauti -0.0890115 0.0910247 -0.978 0.328147   
## fell\_love 0.0194154 0.0910749 0.213 0.831189   
## run\_true 0.0362186 0.1028058 0.352 0.724617   
## previou\_review -0.0529243 0.0974915 -0.543 0.587233   
## wait\_wear 0.0425761 0.0841688 0.506 0.612975   
## love\_shirt 0.1943565 0.0938922 2.070 0.038469 \*   
## run\_size 0.0908339 0.0837174 1.085 0.277936   
## receiv\_compliment -0.0522685 0.0878179 -0.595 0.551724   
## agre\_review -0.1347799 0.0995953 -1.353 0.175987   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.105 on 15707 degrees of freedom  
## Multiple R-squared: 0.03878, Adjusted R-squared: 0.03015   
## F-statistic: 4.494 on 141 and 15707 DF, p-value: < 2.2e-16

f <- paste("rating ~ Nr\_of\_words + ", allFactors)  
lm.onlyfactors <- lm(f, data = reviews\_df[estimation\_sample, ])  
summary(lm.onlyfactors)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.3260 -0.2262 0.7752 0.8228 1.0057   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.1409009 0.0207495 199.566 <2e-16 \*\*\*  
## Nr\_of\_words 0.0019187 0.0008993 2.133 0.0329 \*   
## factor1 0.0100389 0.0089008 1.128 0.2594   
## factor2 -0.0031037 0.0090080 -0.345 0.7304   
## factor3 0.0042736 0.0088954 0.480 0.6309   
## factor4 -0.0088150 0.0089363 -0.986 0.3239   
## factor5 -0.0116757 0.0090077 -1.296 0.1949   
## factor6 0.0087846 0.0089395 0.983 0.3258   
## factor7 0.0052889 0.0088377 0.598 0.5495   
## factor8 0.0055039 0.0088479 0.622 0.5339   
## factor9 -0.0096136 0.0089333 -1.076 0.2819   
## factor10 -0.0032757 0.0087807 -0.373 0.7091   
## factor11 0.0044460 0.0087989 0.505 0.6134   
## factor12 -0.0028422 0.0088375 -0.322 0.7478   
## factor13 -0.0044088 0.0090075 -0.489 0.6245   
## factor14 0.0023152 0.0088993 0.260 0.7947   
## factor15 -0.0117796 0.0089409 -1.317 0.1877   
## factor16 -0.0033779 0.0089551 -0.377 0.7060   
## factor17 0.0068586 0.0088762 0.773 0.4397   
## factor18 0.0040630 0.0089326 0.455 0.6492   
## factor19 -0.0010369 0.0088379 -0.117 0.9066   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.123 on 15828 degrees of freedom  
## Multiple R-squared: 0.0009668, Adjusted R-squared: -0.0002955   
## F-statistic: 0.7659 on 20 and 15828 DF, p-value: 0.7579

f <- paste("rating ~ Nr\_of\_words + ", allEmotions)  
lm.onlyemotions <- lm(f, data = reviews\_df[estimation\_sample,])  
summary(lm.onlyemotions)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.1920 -0.4742 0.3413 0.7981 2.1887   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.978866 0.020320 195.806 < 2e-16 \*\*\*  
## Nr\_of\_words -0.003809 0.001044 -3.650 0.000263 \*\*\*  
## fall 0.078494 0.031945 2.457 0.014016 \*   
## anger 0.013950 0.017211 0.811 0.417648   
## anticipation -0.047142 0.010764 -4.380 1.20e-05 \*\*\*  
## disgust -0.485797 0.019065 -25.481 < 2e-16 \*\*\*  
## fear -0.011809 0.018771 -0.629 0.529299   
## joy 0.245024 0.008830 27.749 < 2e-16 \*\*\*  
## sadness -0.003370 0.013150 -0.256 0.797765   
## surprise 0.009898 0.015069 0.657 0.511266   
## trust 0.046857 0.009616 4.873 1.11e-06 \*\*\*  
## negative -0.001660 0.010631 -0.156 0.875893   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.057 on 15837 degrees of freedom  
## Multiple R-squared: 0.1135, Adjusted R-squared: 0.1129   
## F-statistic: 184.3 on 11 and 15837 DF, p-value: < 2.2e-16

f <- paste("rating ~ Nr\_of\_words + ", allWords)  
lm.onlywords <- lm(f, data = reviews\_df[estimation\_sample,])  
summary(lm.onlywords)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.3885 -0.2542 0.7442 0.8243 1.1485   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.1507166 0.0268243 154.737 < 2e-16 \*\*\*  
## Nr\_of\_words 0.0018945 0.0009002 2.105 0.03534 \*   
## factor20 0.0673898 0.0918349 0.734 0.46307   
## fit -0.0011373 0.0140769 -0.081 0.93561   
## size 0.0248595 0.0295628 0.841 0.40041   
## wear -0.0117270 0.0175826 -0.667 0.50480   
## love -0.0161501 0.0154442 -1.046 0.29571   
## top 0.0034688 0.0142854 0.243 0.80815   
## color -0.0241891 0.0166136 -1.456 0.14542   
## fabric -0.0535662 0.1044613 -0.513 0.60811   
## perfect 0.0210005 0.0241272 0.870 0.38409   
## nice 0.0097081 0.0250307 0.388 0.69813   
## flatter -0.0087417 0.0235487 -0.371 0.71048   
## soft -0.0077044 0.0384695 -0.200 0.84127   
## comfort -0.0111514 0.0253563 -0.440 0.66010   
## bit 0.0264104 0.0341963 0.772 0.43994   
## materi 0.0888692 0.1292046 0.688 0.49158   
## length -0.0115382 0.0363252 -0.318 0.75076   
## shirt -0.0262915 0.0207391 -1.268 0.20492   
## waist -0.0108882 0.0423955 -0.257 0.79732   
## run -0.0386671 0.0361877 -1.069 0.28531   
## jean -0.0030884 0.0249952 -0.124 0.90167   
## cute -0.0096570 0.0277836 -0.348 0.72816   
## petit 0.0033943 0.0257030 0.132 0.89494   
## short -0.0275956 0.0295606 -0.934 0.35056   
## beauti 0.0188564 0.0297643 0.634 0.52640   
## sweater 0.0139392 0.0270864 0.515 0.60683   
## medium -0.0675559 0.1194912 -0.565 0.57183   
## feel 0.0394652 0.0356537 1.107 0.26835   
## qualiti -0.0147159 0.0362982 -0.405 0.68518   
## x 0.0200857 0.0280494 0.716 0.47395   
## sleev -0.0172998 0.0285666 -0.606 0.54479   
## bought -0.0326764 0.0370578 -0.882 0.37791   
## retail 0.0011855 0.0361566 0.033 0.97384   
## skirt 0.0534045 0.0633183 0.843 0.39900   
## pretti 0.0132488 0.0306521 0.432 0.66558   
## style 0.0787118 0.0299344 2.629 0.00856 \*\*   
## review -0.0308439 0.0352458 -0.875 0.38153   
## pant -0.0095647 0.0250075 -0.382 0.70211   
## cut 0.0522917 0.0313580 1.668 0.09542 .   
## store 0.0095403 0.0336041 0.284 0.77649   
## purchas -0.0191977 0.0317651 -0.604 0.54561   
## design 0.0112943 0.0319400 0.354 0.72364   
## tight 0.0197269 0.0315619 0.625 0.53197   
## black -0.0085978 0.0291809 -0.295 0.76827   
## return -0.0213402 0.0337561 -0.632 0.52727   
## true 0.0189004 0.0387544 0.488 0.62577   
## pictur 0.0285313 0.0332068 0.859 0.39024   
## summer -0.0205734 0.0335366 -0.613 0.53958   
## super -0.0376130 0.0344054 -1.093 0.27431   
## line -0.0542916 0.0345444 -1.572 0.11605   
## detail 0.0111256 0.0349118 0.319 0.74998   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.123 on 15797 degrees of freedom  
## Multiple R-squared: 0.002717, Adjusted R-squared: -0.0005024   
## F-statistic: 0.844 on 51 and 15797 DF, p-value: 0.7779

f <- paste("rating ~ Nr\_of\_words + ", allBigrams)  
lm.bigrams <- lm(f, data = reviews\_df[estimation\_sample,])  
summary(lm.bigrams)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8687 -0.4346 0.5109 0.8263 1.5041   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.040118 0.022418 180.219 < 2e-16 \*\*\*  
## Nr\_of\_words -0.010378 0.001049 -9.896 < 2e-16 \*\*\*  
## positive 0.128196 0.005838 21.958 < 2e-16 \*\*\*  
## dress\_dress -0.306899 0.089209 -3.440 0.000583 \*\*\*  
## bodi\_type 0.089264 0.080668 1.107 0.268502   
## size\_x 0.015573 0.089680 0.174 0.862143   
## size\_8 0.028307 0.075734 0.374 0.708578   
## size\_size 0.006194 0.069285 0.089 0.928769   
## dress\_wear -0.193591 0.088473 -2.188 0.028675 \*   
## normal\_size -0.092076 0.078275 -1.176 0.239488   
## love\_top -0.043808 0.050446 -0.868 0.385179   
## petit\_size -0.091086 0.070102 -1.299 0.193848   
## size\_2 0.022684 0.067620 0.335 0.737279   
## love\_love -0.062743 0.052197 -1.202 0.229366   
## arm\_hole 0.084524 0.079348 1.065 0.286787   
## sale\_price 0.091847 0.085848 1.070 0.284691   
## dress\_not -0.035468 0.074580 -0.476 0.634390   
## not\_fit -0.056119 0.079207 -0.709 0.478638   
## size\_6 0.134153 0.064509 2.080 0.037577 \*   
## regular\_size 0.067450 0.068078 0.991 0.321807   
## fit\_perfectli 0.020947 0.041149 0.509 0.610731   
## dress\_fit 0.090468 0.062488 1.448 0.147704   
## beauti\_dress 0.022706 0.088713 0.256 0.797995   
## dress\_love 0.057656 0.080405 0.717 0.473340   
## dress\_beauti 0.160627 0.083448 1.925 0.054262 .   
## fit\_flatter 0.003147 0.090750 0.035 0.972339   
## wear\_size -0.094189 0.067245 -1.401 0.161326   
## super\_cute -0.143365 0.069037 -2.077 0.037852 \*   
## highli\_recommend 0.034405 0.067025 0.513 0.607739   
## love\_fit -0.088510 0.077732 -1.139 0.254866   
## fabric\_soft 0.001432 0.070152 0.020 0.983717   
## fit\_perfect -0.008367 0.060877 -0.137 0.890684   
## fit\_nice 0.076080 0.081379 0.935 0.349859   
## wear\_dress -0.064752 0.082630 -0.784 0.433265   
## length\_perfect 0.088392 0.079875 1.107 0.268474   
## size\_medium 0.014725 0.077160 0.191 0.848660   
## read\_review -0.052078 0.084673 -0.615 0.538527   
## fabric\_nice 0.023695 0.085666 0.277 0.782091   
## fit\_true 0.034386 0.077125 0.446 0.655710   
## true\_size -0.048427 0.052945 -0.915 0.360386   
## love\_dress -0.057676 0.043085 -1.339 0.180699   
## light\_weight 0.117711 0.090805 1.296 0.194886   
## size\_4 -0.043106 0.064374 -0.670 0.503111   
## fit\_not -0.158963 0.080960 -1.963 0.049609 \*   
## love\_sweater -0.105699 0.084263 -1.254 0.209718   
## super\_soft -0.050270 0.072750 -0.691 0.489582   
## size\_fit 0.087426 0.062075 1.408 0.159038   
## not\_tight 0.135504 0.091942 1.474 0.140553   
## love\_color -0.058111 0.056810 -1.023 0.306374   
## size\_10 0.001683 0.095259 0.018 0.985906   
## top\_fit 0.123367 0.078885 1.564 0.117866   
## skinni\_jean -0.067971 0.072956 -0.932 0.351518   
## usual\_size 0.023934 0.061818 0.387 0.698639   
## x\_fit 0.118591 0.078191 1.517 0.129366   
## absolut\_love 0.065546 0.072099 0.909 0.363305   
## medium\_fit 0.062657 0.070429 0.890 0.373664   
## dress\_run -0.021779 0.088975 -0.245 0.806633   
## top\_not -0.103681 0.095657 -1.084 0.278436   
## plan\_wear 0.076470 0.089714 0.852 0.394016   
## not\_flatter -0.090481 0.072376 -1.250 0.211261   
## materi\_soft 0.001699 0.077812 0.022 0.982576   
## color\_love 0.028829 0.085240 0.338 0.735206   
## fit\_tt 0.008172 0.084357 0.097 0.922827   
## soft\_comfort -0.092470 0.072558 -1.274 0.202530   
## color\_beauti -0.103510 0.085512 -1.210 0.226114   
## fell\_love 0.006367 0.087813 0.073 0.942203   
## run\_true 0.037319 0.098292 0.380 0.704194   
## previou\_review -0.055196 0.089420 -0.617 0.537069   
## wait\_wear 0.049674 0.081594 0.609 0.542671   
## love\_shirt 0.117646 0.087805 1.340 0.180316   
## run\_size 0.064734 0.078521 0.824 0.409715   
## receiv\_compliment -0.029503 0.083587 -0.353 0.724123   
## agre\_review -0.141689 0.091857 -1.542 0.122973   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.105 on 15776 degrees of freedom  
## Multiple R-squared: 0.03517, Adjusted R-squared: 0.03076   
## F-statistic: 7.986 on 72 and 15776 DF, p-value: < 2.2e-16

f <- paste("rating ~ Nr\_of\_words + ", allWords , " + ",allBigrams)  
lm.words\_bigrams <- lm(f, data = reviews\_df[estimation\_sample,])  
summary(lm.words\_bigrams)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8112 -0.4437 0.5045 0.8258 1.5551   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.0480011 0.0271061 149.339 < 2e-16 \*\*\*  
## Nr\_of\_words -0.0104103 0.0010509 -9.906 < 2e-16 \*\*\*  
## factor20 0.0692364 0.0954239 0.726 0.468115   
## fit -0.0126497 0.0164458 -0.769 0.441802   
## size 0.0249711 0.0319297 0.782 0.434188   
## wear -0.0014301 0.0187272 -0.076 0.939129   
## love -0.0186621 0.0180470 -1.034 0.301112   
## top 0.0002395 0.0154274 0.016 0.987612   
## color -0.0128516 0.0173317 -0.742 0.458394   
## fabric -0.0578089 0.1082847 -0.534 0.593446   
## perfect 0.0135023 0.0264230 0.511 0.609355   
## nice -0.0053087 0.0265859 -0.200 0.841732   
## flatter -0.0068331 0.0250969 -0.272 0.785420   
## soft -0.0109849 0.0427681 -0.257 0.797299   
## comfort 0.0026081 0.0263704 0.099 0.921218   
## bit 0.0254571 0.0345752 0.736 0.461571   
## materi 0.0948176 0.1342906 0.706 0.480160   
## length -0.0149376 0.0373657 -0.400 0.689333   
## shirt -0.0410134 0.0218174 -1.880 0.060148 .   
## waist -0.0169862 0.0433707 -0.392 0.695320   
## run -0.0492395 0.0389809 -1.263 0.206547   
## jean 0.0044077 0.0261939 0.168 0.866371   
## cute -0.0009325 0.0292065 -0.032 0.974529   
## petit 0.0086641 0.0275578 0.314 0.753223   
## short -0.0293706 0.0295891 -0.993 0.320914   
## beauti 0.0144367 0.0313797 0.460 0.645476   
## sweater 0.0327526 0.0283670 1.155 0.248271   
## medium -0.0882653 0.1245468 -0.709 0.478526   
## feel 0.0419731 0.0358635 1.170 0.241875   
## qualiti -0.0058243 0.0364344 -0.160 0.872997   
## x 0.0106097 0.0311565 0.341 0.733462   
## sleev -0.0136718 0.0284963 -0.480 0.631395   
## bought -0.0320830 0.0373925 -0.858 0.390902   
## retail -0.0086806 0.0363775 -0.239 0.811400   
## skirt 0.0536066 0.0652315 0.822 0.411209   
## pretti 0.0112504 0.0303102 0.371 0.710512   
## style 0.0800898 0.0295686 2.709 0.006764 \*\*   
## review -0.0089655 0.0417959 -0.215 0.830156   
## pant -0.0023118 0.0247678 -0.093 0.925635   
## cut 0.0427400 0.0311397 1.373 0.169920   
## store 0.0078464 0.0336447 0.233 0.815599   
## purchas -0.0322880 0.0315993 -1.022 0.306893   
## design -0.0004336 0.0315877 -0.014 0.989049   
## tight 0.0017679 0.0333119 0.053 0.957676   
## black -0.0149576 0.0288384 -0.519 0.603998   
## return -0.0121657 0.0333969 -0.364 0.715656   
## true 0.1103761 0.0725947 1.520 0.128420   
## pictur 0.0361069 0.0329224 1.097 0.272777   
## summer -0.0181864 0.0332395 -0.547 0.584294   
## super -0.0104999 0.0402835 -0.261 0.794365   
## line -0.0505466 0.0343854 -1.470 0.141582   
## detail 0.0181721 0.0346389 0.525 0.599858   
## positive 0.1280900 0.0058523 21.887 < 2e-16 \*\*\*  
## dress\_dress -0.3040690 0.0894085 -3.401 0.000673 \*\*\*  
## bodi\_type 0.0880073 0.0810900 1.085 0.277803   
## size\_x -0.0082287 0.0955546 -0.086 0.931376   
## size\_8 0.0230515 0.0777193 0.297 0.766776   
## size\_size -0.0008641 0.0746612 -0.012 0.990766   
## dress\_wear -0.1867979 0.0899402 -2.077 0.037826 \*   
## normal\_size -0.0876805 0.0801771 -1.094 0.274153   
## love\_top -0.0398478 0.0535973 -0.743 0.457210   
## petit\_size -0.0984988 0.0781828 -1.260 0.207741   
## size\_2 0.0269641 0.0707350 0.381 0.703061   
## love\_love -0.0401778 0.0563482 -0.713 0.475839   
## arm\_hole 0.0861758 0.0797267 1.081 0.279762   
## sale\_price 0.1041789 0.0864854 1.205 0.228382   
## dress\_not -0.0318038 0.0748249 -0.425 0.670811   
## not\_fit -0.0518381 0.0810773 -0.639 0.522594   
## size\_6 0.1346172 0.0660500 2.038 0.041556 \*   
## regular\_size 0.0746388 0.0715656 1.043 0.296991   
## fit\_perfectli 0.0278051 0.0428556 0.649 0.516471   
## dress\_fit 0.0930756 0.0634249 1.467 0.142262   
## beauti\_dress 0.0151688 0.0897997 0.169 0.865863   
## dress\_love 0.0661845 0.0813955 0.813 0.416160   
## dress\_beauti 0.1531462 0.0850198 1.801 0.071675 .   
## fit\_flatter 0.0172280 0.0946358 0.182 0.855549   
## wear\_size -0.0893886 0.0703689 -1.270 0.204002   
## super\_cute -0.1415410 0.0806680 -1.755 0.079345 .   
## highli\_recommend 0.0370575 0.0674086 0.550 0.582503   
## love\_fit -0.0734177 0.0798031 -0.920 0.357595   
## fabric\_soft 0.0142725 0.0772826 0.185 0.853483   
## fit\_perfect -0.0197522 0.0667274 -0.296 0.767224   
## fit\_nice 0.1022839 0.0864701 1.183 0.236874   
## wear\_dress -0.0587446 0.0841659 -0.698 0.485212   
## length\_perfect 0.0998278 0.0872363 1.144 0.252501   
## size\_medium 0.0096144 0.0832586 0.115 0.908069   
## read\_review -0.0593745 0.0939861 -0.632 0.527568   
## fabric\_nice 0.0203404 0.0906410 0.224 0.822444   
## fit\_true 0.0275150 0.0788053 0.349 0.726980   
## true\_size -0.1375629 0.0836264 -1.645 0.099996 .   
## love\_dress -0.0555777 0.0435905 -1.275 0.202330   
## light\_weight 0.1168555 0.0912404 1.281 0.200303   
## size\_4 -0.0403265 0.0667970 -0.604 0.546040   
## fit\_not -0.1584068 0.0821056 -1.929 0.053711 .   
## love\_sweater -0.1562283 0.0900306 -1.735 0.082711 .   
## super\_soft -0.0069067 0.0857834 -0.081 0.935830   
## size\_fit 0.0904296 0.0644810 1.402 0.160808   
## not\_tight 0.1365868 0.0982440 1.390 0.164463   
## love\_color -0.0452962 0.0599933 -0.755 0.450247   
## size\_10 -0.0083364 0.0974870 -0.086 0.931855   
## top\_fit 0.1259655 0.0815809 1.544 0.122595   
## skinni\_jean -0.0806204 0.0783770 -1.029 0.303673   
## usual\_size 0.0141160 0.0637233 0.222 0.824690   
## x\_fit 0.1085841 0.0844223 1.286 0.198392   
## absolut\_love 0.0821844 0.0740398 1.110 0.267015   
## medium\_fit 0.0714986 0.0779588 0.917 0.359087   
## dress\_run 0.0048007 0.0916930 0.052 0.958246   
## top\_not -0.1102327 0.0978851 -1.126 0.260122   
## plan\_wear 0.0859106 0.0913110 0.941 0.346793   
## not\_flatter -0.0941275 0.0760554 -1.238 0.215876   
## materi\_soft 0.0358347 0.0855647 0.419 0.675366   
## color\_love 0.0539946 0.0883690 0.611 0.541200   
## fit\_tt 0.0154006 0.0857887 0.180 0.857533   
## soft\_comfort -0.0693907 0.0787791 -0.881 0.378425   
## color\_beauti -0.0969922 0.0908953 -1.067 0.285954   
## fell\_love 0.0140966 0.0897909 0.157 0.875252   
## run\_true 0.0401846 0.1027410 0.391 0.695710   
## previou\_review -0.0592782 0.0973604 -0.609 0.542631   
## wait\_wear 0.0567240 0.0832958 0.681 0.495885   
## love\_shirt 0.1870287 0.0937570 1.995 0.046079 \*   
## run\_size 0.0859429 0.0835340 1.029 0.303572   
## receiv\_compliment -0.0305604 0.0839594 -0.364 0.715871   
## agre\_review -0.1451545 0.0994969 -1.459 0.144617   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.105 on 15726 degrees of freedom  
## Multiple R-squared: 0.03743, Adjusted R-squared: 0.02996   
## F-statistic: 5.013 on 122 and 15726 DF, p-value: < 2.2e-16

f <- paste("rating ~ Nr\_of\_words + positive + negative")  
lm.posneg <- lm(f, data = reviews\_df[estimation\_sample,])  
summary(lm.posneg)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8528 -0.4351 0.5119 0.8317 1.4661   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.045391 0.020964 192.965 < 2e-16 \*\*\*  
## Nr\_of\_words -0.005802 0.001152 -5.035 4.84e-07 \*\*\*  
## positive 0.125683 0.005819 21.597 < 2e-16 \*\*\*  
## negative -0.076213 0.008069 -9.445 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.102 on 15845 degrees of freedom  
## Multiple R-squared: 0.03536, Adjusted R-squared: 0.03518   
## F-statistic: 193.6 on 3 and 15845 DF, p-value: < 2.2e-16

# Optimal model through backward selection  
f <- "rating ~ Nr\_of\_words + factor1 + factor3 + factor6 +   
 factor7 + factor8 + factor9 + factor10 + factor13 + factor16 +   
 factor19 + fall + anticipation + disgust + joy + trust +   
 factor20 + fit + size + wear + love + color + fabric + nice +   
 flatter + jean + feel + style + purchas + line + positive +   
 dress\_dress + dress\_wear + arm\_hole + size\_6 + dress\_beauti +   
 wear\_size + super\_cute + true\_size + love\_dress + fit\_not +   
 love\_sweater + top\_fit + skinni\_jean + love\_shirt + agre\_review +   
 bought"  
lm.optimal <- lm(f, data = reviews\_df[estimation\_sample,])  
summary(lm.optimal)

##   
## Call:  
## lm(formula = f, data = reviews\_df[estimation\_sample, ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.2117 -0.5068 0.3363 0.7778 2.1969   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.402727 0.086047 51.167 < 2e-16 \*\*\*  
## Nr\_of\_words -0.003223 0.001033 -3.121 0.001806 \*\*   
## factor1 0.289985 0.084488 3.432 0.000600 \*\*\*  
## factor3 0.189781 0.077899 2.436 0.014852 \*   
## factor6 -0.174437 0.073688 -2.367 0.017934 \*   
## factor7 -0.138071 0.041415 -3.334 0.000859 \*\*\*  
## factor8 0.063791 0.016770 3.804 0.000143 \*\*\*  
## factor9 -0.027963 0.009795 -2.855 0.004313 \*\*   
## factor10 -0.103398 0.029671 -3.485 0.000494 \*\*\*  
## factor13 0.042565 0.016613 2.562 0.010412 \*   
## factor16 -0.078922 0.027261 -2.895 0.003795 \*\*   
## factor19 0.048972 0.013518 3.623 0.000292 \*\*\*  
## fall 0.078312 0.032265 2.427 0.015230 \*   
## anticipation -0.042773 0.010660 -4.012 6.04e-05 \*\*\*  
## disgust -0.490222 0.016325 -30.029 < 2e-16 \*\*\*  
## joy 0.256521 0.010360 24.762 < 2e-16 \*\*\*  
## trust 0.050840 0.009815 5.180 2.25e-07 \*\*\*  
## factor20 -0.106357 0.036667 -2.901 0.003729 \*\*   
## fit -0.308837 0.114619 -2.694 0.007058 \*\*   
## size -0.337057 0.116006 -2.906 0.003672 \*\*   
## wear -0.038117 0.018304 -2.083 0.037312 \*   
## love -0.032398 0.016001 -2.025 0.042916 \*   
## color -0.281230 0.128505 -2.188 0.028650 \*   
## fabric 0.317939 0.100586 3.161 0.001576 \*\*   
## nice -0.051137 0.022821 -2.241 0.025057 \*   
## flatter -0.199782 0.071649 -2.788 0.005304 \*\*   
## jean -0.114202 0.058514 -1.952 0.050991 .   
## feel 0.058324 0.028030 2.081 0.037469 \*   
## style 0.065500 0.028449 2.302 0.021327 \*   
## purchas -0.047144 0.029619 -1.592 0.111481   
## line -0.062047 0.030960 -2.004 0.045075 \*   
## positive -0.015206 0.008419 -1.806 0.070901 .   
## dress\_dress -0.308480 0.084907 -3.633 0.000281 \*\*\*  
## dress\_wear -0.182521 0.085390 -2.137 0.032574 \*   
## arm\_hole 0.121332 0.076053 1.595 0.110654   
## size\_6 0.094153 0.063388 1.485 0.137474   
## dress\_beauti 0.143009 0.079847 1.791 0.073308 .   
## wear\_size -0.113760 0.065159 -1.746 0.080850 .   
## super\_cute -0.136480 0.066711 -2.046 0.040789 \*   
## true\_size -0.076276 0.039006 -1.956 0.050539 .   
## love\_dress -0.068200 0.040892 -1.668 0.095375 .   
## fit\_not -0.187257 0.076392 -2.451 0.014247 \*   
## love\_sweater -0.126344 0.081886 -1.543 0.122870   
## top\_fit 0.128039 0.076261 1.679 0.093181 .   
## skinni\_jean -0.126940 0.075080 -1.691 0.090910 .   
## love\_shirt 0.202984 0.088883 2.284 0.022401 \*   
## agre\_review -0.147297 0.087847 -1.677 0.093612 .   
## bought -0.044642 0.029594 -1.508 0.131449   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.055 on 15801 degrees of freedom  
## Multiple R-squared: 0.1195, Adjusted R-squared: 0.1168   
## F-statistic: 45.61 on 47 and 15801 DF, p-value: < 2.2e-16

#predictive performance of model with all features  
lm.all\_predict <- predict(lm.all,data = reviews\_df[test\_sample,])  
t.test(reviews\_df[test\_sample,"rating"], lm.all\_predict)

##   
## Welch Two Sample t-test  
##   
## data: reviews\_df[test\_sample, "rating"] and lm.all\_predict  
## t = 0.55624, df = 7541.5, p-value = 0.5781  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.01924804 0.03449891  
## sample estimates:  
## mean of x mean of y   
## 4.188899 4.181273

#Predictive performance of model with subset of features based on backward selection (optimal model)  
lm.optimal\_predict <- predict(lm.optimal,data = reviews\_df[test\_sample,])  
t.test(reviews\_df[test\_sample,"rating"], lm.optimal\_predict)

##   
## Welch Two Sample t-test  
##   
## data: reviews\_df[test\_sample, "rating"] and lm.optimal\_predict  
## t = 0.55655, df = 7524.7, p-value = 0.5778  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.01923271 0.03448359  
## sample estimates:  
## mean of x mean of y   
## 4.188899 4.181273

# AIC comparison for all linear models   
  
AIC(lm.all, lm.nodict, lm.onlyfactors, lm.onlyemotions, lm.onlywords, lm.bigrams, lm.words\_bigrams, lm.posneg, lm.optimal)

## df AIC  
## lm.all 153 46877.06  
## lm.nodict 143 48295.20  
## lm.onlyfactors 22 48664.68  
## lm.onlyemotions 13 46752.45  
## lm.onlywords 53 48698.88  
## lm.bigrams 74 48216.63  
## lm.words\_bigrams 124 48279.35  
## lm.posneg 5 48075.38  
## lm.optimal 49 46717.69

# Test some nested models  
  
anova(lm.all, lm.optimal)

## Analysis of Variance Table  
##   
## Model 1: rating ~ Nr\_of\_words + (Nr\_of\_words + factor1 + factor2 + factor3 +   
## factor4 + factor5 + factor6 + factor7 + factor8 + factor9 +   
## factor10 + factor11 + factor12 + factor13 + factor14 + factor15 +   
## factor16 + factor17 + factor18 + factor19) + (fall + anger +   
## anticipation + disgust + fear + joy + sadness + surprise +   
## trust + negative) + (factor20 + fit + size + wear + love +   
## top + color + fabric + perfect + nice + flatter + soft +   
## comfort + bit + materi + length + shirt + waist + run + jean +   
## cute + petit + short + beauti + sweater + medium + feel +   
## qualiti + x + sleev + bought + retail + skirt + pretti +   
## style + review + pant + cut + store + purchas + design +   
## tight + black + return + true + pictur + summer + super +   
## line + detail) + (positive + dress\_dress + bodi\_type + size\_x +   
## size\_8 + size\_size + dress\_wear + normal\_size + love\_top +   
## petit\_size + size\_2 + love\_love + arm\_hole + sale\_price +   
## dress\_not + not\_fit + size\_6 + regular\_size + fit\_perfectli +   
## dress\_fit + beauti\_dress + dress\_love + dress\_beauti + fit\_flatter +   
## wear\_size + super\_cute + highli\_recommend + love\_fit + fabric\_soft +   
## fit\_perfect + fit\_nice + wear\_dress + length\_perfect + size\_medium +   
## read\_review + fabric\_nice + fit\_true + true\_size + love\_dress +   
## light\_weight + size\_4 + fit\_not + love\_sweater + super\_soft +   
## size\_fit + not\_tight + love\_color + size\_10 + top\_fit + skinni\_jean +   
## usual\_size + x\_fit + absolut\_love + medium\_fit + dress\_run +   
## top\_not + plan\_wear + not\_flatter + materi\_soft + color\_love +   
## fit\_tt + soft\_comfort + color\_beauti + fell\_love + run\_true +   
## previou\_review + wait\_wear + love\_shirt + run\_size + receiv\_compliment +   
## agre\_review)  
## Model 2: rating ~ Nr\_of\_words + factor1 + factor3 + factor6 + factor7 +   
## factor8 + factor9 + factor10 + factor13 + factor16 + factor19 +   
## fall + anticipation + disgust + joy + trust + factor20 +   
## fit + size + wear + love + color + fabric + nice + flatter +   
## jean + feel + style + purchas + line + positive + dress\_dress +   
## dress\_wear + arm\_hole + size\_6 + dress\_beauti + wear\_size +   
## super\_cute + true\_size + love\_dress + fit\_not + love\_sweater +   
## top\_fit + skinni\_jean + love\_shirt + agre\_review + bought  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 15697 17525   
## 2 15801 17579 -104 -53.857 0.4638 1

anova(lm.posneg, lm.onlyemotions)

## Analysis of Variance Table  
##   
## Model 1: rating ~ Nr\_of\_words + positive + negative  
## Model 2: rating ~ Nr\_of\_words + (fall + anger + anticipation + disgust +   
## fear + joy + sadness + surprise + trust + negative)  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 15845 19258   
## 2 15837 17698 8 1560.1 174.51 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(lm.words\_bigrams, lm.all)

## Analysis of Variance Table  
##   
## Model 1: rating ~ Nr\_of\_words + (factor20 + fit + size + wear + love +   
## top + color + fabric + perfect + nice + flatter + soft +   
## comfort + bit + materi + length + shirt + waist + run + jean +   
## cute + petit + short + beauti + sweater + medium + feel +   
## qualiti + x + sleev + bought + retail + skirt + pretti +   
## style + review + pant + cut + store + purchas + design +   
## tight + black + return + true + pictur + summer + super +   
## line + detail) + (positive + dress\_dress + bodi\_type + size\_x +   
## size\_8 + size\_size + dress\_wear + normal\_size + love\_top +   
## petit\_size + size\_2 + love\_love + arm\_hole + sale\_price +   
## dress\_not + not\_fit + size\_6 + regular\_size + fit\_perfectli +   
## dress\_fit + beauti\_dress + dress\_love + dress\_beauti + fit\_flatter +   
## wear\_size + super\_cute + highli\_recommend + love\_fit + fabric\_soft +   
## fit\_perfect + fit\_nice + wear\_dress + length\_perfect + size\_medium +   
## read\_review + fabric\_nice + fit\_true + true\_size + love\_dress +   
## light\_weight + size\_4 + fit\_not + love\_sweater + super\_soft +   
## size\_fit + not\_tight + love\_color + size\_10 + top\_fit + skinni\_jean +   
## usual\_size + x\_fit + absolut\_love + medium\_fit + dress\_run +   
## top\_not + plan\_wear + not\_flatter + materi\_soft + color\_love +   
## fit\_tt + soft\_comfort + color\_beauti + fell\_love + run\_true +   
## previou\_review + wait\_wear + love\_shirt + run\_size + receiv\_compliment +   
## agre\_review)  
## Model 2: rating ~ Nr\_of\_words + (Nr\_of\_words + factor1 + factor2 + factor3 +   
## factor4 + factor5 + factor6 + factor7 + factor8 + factor9 +   
## factor10 + factor11 + factor12 + factor13 + factor14 + factor15 +   
## factor16 + factor17 + factor18 + factor19) + (fall + anger +   
## anticipation + disgust + fear + joy + sadness + surprise +   
## trust + negative) + (factor20 + fit + size + wear + love +   
## top + color + fabric + perfect + nice + flatter + soft +   
## comfort + bit + materi + length + shirt + waist + run + jean +   
## cute + petit + short + beauti + sweater + medium + feel +   
## qualiti + x + sleev + bought + retail + skirt + pretti +   
## style + review + pant + cut + store + purchas + design +   
## tight + black + return + true + pictur + summer + super +   
## line + detail) + (positive + dress\_dress + bodi\_type + size\_x +   
## size\_8 + size\_size + dress\_wear + normal\_size + love\_top +   
## petit\_size + size\_2 + love\_love + arm\_hole + sale\_price +   
## dress\_not + not\_fit + size\_6 + regular\_size + fit\_perfectli +   
## dress\_fit + beauti\_dress + dress\_love + dress\_beauti + fit\_flatter +   
## wear\_size + super\_cute + highli\_recommend + love\_fit + fabric\_soft +   
## fit\_perfect + fit\_nice + wear\_dress + length\_perfect + size\_medium +   
## read\_review + fabric\_nice + fit\_true + true\_size + love\_dress +   
## light\_weight + size\_4 + fit\_not + love\_sweater + super\_soft +   
## size\_fit + not\_tight + love\_color + size\_10 + top\_fit + skinni\_jean +   
## usual\_size + x\_fit + absolut\_love + medium\_fit + dress\_run +   
## top\_not + plan\_wear + not\_flatter + materi\_soft + color\_love +   
## fit\_tt + soft\_comfort + color\_beauti + fell\_love + run\_true +   
## previou\_review + wait\_wear + love\_shirt + run\_size + receiv\_compliment +   
## agre\_review)  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 15726 19217   
## 2 15697 17525 29 1691.5 52.242 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(lm.all, lm.onlyemotions)

## Analysis of Variance Table  
##   
## Model 1: rating ~ Nr\_of\_words + (Nr\_of\_words + factor1 + factor2 + factor3 +   
## factor4 + factor5 + factor6 + factor7 + factor8 + factor9 +   
## factor10 + factor11 + factor12 + factor13 + factor14 + factor15 +   
## factor16 + factor17 + factor18 + factor19) + (fall + anger +   
## anticipation + disgust + fear + joy + sadness + surprise +   
## trust + negative) + (factor20 + fit + size + wear + love +   
## top + color + fabric + perfect + nice + flatter + soft +   
## comfort + bit + materi + length + shirt + waist + run + jean +   
## cute + petit + short + beauti + sweater + medium + feel +   
## qualiti + x + sleev + bought + retail + skirt + pretti +   
## style + review + pant + cut + store + purchas + design +   
## tight + black + return + true + pictur + summer + super +   
## line + detail) + (positive + dress\_dress + bodi\_type + size\_x +   
## size\_8 + size\_size + dress\_wear + normal\_size + love\_top +   
## petit\_size + size\_2 + love\_love + arm\_hole + sale\_price +   
## dress\_not + not\_fit + size\_6 + regular\_size + fit\_perfectli +   
## dress\_fit + beauti\_dress + dress\_love + dress\_beauti + fit\_flatter +   
## wear\_size + super\_cute + highli\_recommend + love\_fit + fabric\_soft +   
## fit\_perfect + fit\_nice + wear\_dress + length\_perfect + size\_medium +   
## read\_review + fabric\_nice + fit\_true + true\_size + love\_dress +   
## light\_weight + size\_4 + fit\_not + love\_sweater + super\_soft +   
## size\_fit + not\_tight + love\_color + size\_10 + top\_fit + skinni\_jean +   
## usual\_size + x\_fit + absolut\_love + medium\_fit + dress\_run +   
## top\_not + plan\_wear + not\_flatter + materi\_soft + color\_love +   
## fit\_tt + soft\_comfort + color\_beauti + fell\_love + run\_true +   
## previou\_review + wait\_wear + love\_shirt + run\_size + receiv\_compliment +   
## agre\_review)  
## Model 2: rating ~ Nr\_of\_words + (fall + anger + anticipation + disgust +   
## fear + joy + sadness + surprise + trust + negative)  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 15697 17525   
## 2 15837 17698 -140 -172.67 1.1047 0.1889

# take happy or unhappy as the binary dependent variable, did not pursue.   
  
#reviews\_df$Rating\_b<- ifelse(reviews\_df$Rating\_b == "happy",1,0)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allFactors, " + ", allEmotions, " + ", allWords , " + ", allBigrams)  
#lm.all <- lm(f, data=reviews\_df[estimation\_sample,] )  
#summary(lm.all)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allFactors, " + ", allWords , " + ", allBigrams)  
#lm.nodict <- lm(f, data=reviews\_df[estimation\_sample,] )  
#summary(lm.nodict)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allFactors)  
#lm.onlyfactors <- lm(f, data = reviews\_df[estimation\_sample, ])  
#summary(lm.onlyfactors)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allEmotions)  
#lm.onlyemotions <- lm(f, data = reviews\_df[estimation\_sample,])  
#summary(lm.onlyemotions)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allWords)  
#lm.onlywords <- lm(f, data = reviews\_df[estimation\_sample,])  
#summary(lm.onlywords)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allBigrams)  
#lm.bigrams <- lm(f, data = reviews\_df[estimation\_sample,])  
#summary(lm.bigrams)  
  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + ", allWords , " + ",allBigrams)  
#lm.words\_bigrams <- lm(f, data = reviews\_df[estimation\_sample,])  
#summary(lm.words\_bigrams)  
  
#f <- paste("Rating\_b ~ Nr\_of\_words + positive + negative")  
#lm.posneg <- lm(f, data = reviews\_df[estimation\_sample,])  
#summary(lm.posneg)  
  
#f <- Rating\_b ~ Nr\_of\_words + factor1 + factor3 + factor6 + factor7 + factor8 + factor9 + factor10 + factor13 + factor16 + factor19 + fall + anticipation + disgust + joy + trust + factor20 + fit + size + wear + love + color + fabric + nice + flatter + jean + feel + style + purchas + line + positive + dress\_dress + dress\_wear + arm\_hole + size\_6 + dress\_beauti + wear\_size + super\_cute + true\_size + love\_dress + fit\_not + love\_sweater + top\_fit + skinni\_jean + love\_shirt + agre\_review + bought  
#lm.optimal <- lm(f, data = reviews\_df[estimation\_sample,])  
#summary(lm.optimal)  
  
# AIC comparison  
#AIC(lm.all, lm.nodict, lm.onlyfactors, lm.onlyemotions, lm.onlywords, lm.bigrams, lm.words\_bigrams, lm.posneg, lm.optimal)

# Plot prediction histograms  
#library(ggplot2)  
#dat <- data.frame(Prediction=predict(lm.optimal), rating=reviews\_df[estimation\_sample, "rating"])  
#ggplot(dat, aes(x=Prediction)) +   
 #geom\_histogram(data=subset(dat,rating == 1),fill = "red", alpha = 0.2) +  
 #geom\_histogram(data=subset(dat,rating == 2),fill = "blue", alpha = 0.2) +

#Predictive performance of all linear models and combinations -   
  
for (m in list(lm.all, lm.nodict, lm.onlyfactors, lm.onlyemotions, lm.onlywords, lm.bigrams, lm.words\_bigrams, lm.posneg, lm.optimal))  
{  
 predicted <- predict.lm(m, reviews\_df[estimation\_sample,])  
 mse <- mean((as.numeric(reviews\_df[estimation\_sample,"rating"])-predicted)^2)  
   
 predicted\_test <- predict.lm(m, reviews\_df[test\_sample,])  
 mse\_test<- mean((as.numeric(reviews\_df[test\_sample,"rating"])-predicted\_test)^2)  
 print(c(sqrt(mse), sqrt(mse\_test)))  
}

## [1] 1.051558 1.042271  
## [1] 1.100366 1.091730  
## [1] 1.121799 1.099943  
## [1] 1.056725 1.036441  
## [1] 1.120816 1.101626  
## [1] 1.102431 1.088703  
## [1] 1.101135 1.090724  
## [1] 1.102318 1.081484  
## [1] 1.053172 1.041545

print("lm.all, lm.nodict, lm.onlyfactors, lm.onlyemotions, lm.onlywords, lm.bigrams, lm.words\_bigrams, lm.posneg, lm.optimal")

## [1] "lm.all, lm.nodict, lm.onlyfactors, lm.onlyemotions, lm.onlywords, lm.bigrams, lm.words\_bigrams, lm.posneg, lm.optimal"

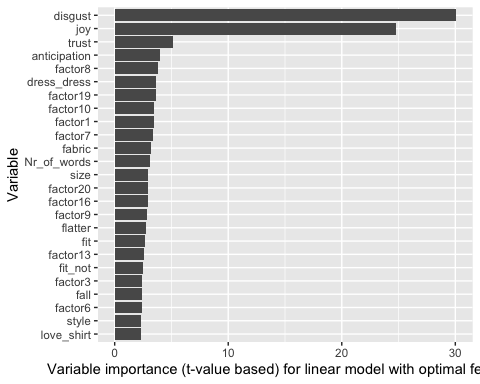
#Variable importance using t-values of the linear model with all features  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

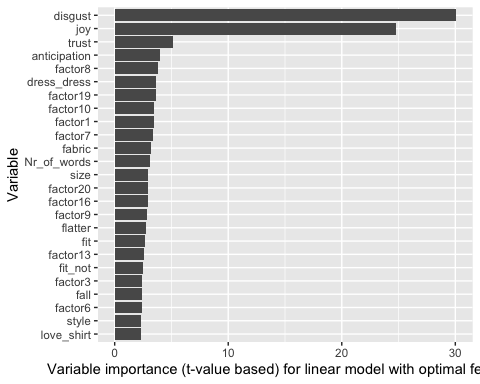
## The following object is masked from 'package:purrr':  
##   
## lift

vi <- varImp(lm.optimal)   
vi$Variable <- rownames(vi)  
vi <- vi[order(-vi$Overall),]  
vi$Variable <- factor(vi$Variable, levels = rev(vi$Variable))  
  
ggplot(vi[1:25, ], aes(Variable,Overall)) + geom\_bar(stat = "identity") + coord\_flip() + ylab("Variable importance (t-value based) for linear model with optimal features")



#ggplot(vi[(nrow(vi)-24):nrow(vi), ], aes(Variable,Overall)) + geom\_bar(stat = "identity") + coord\_flip() + ylab("Variable importance (t-value) for linear model with all features")  
  
# Variable importance using standardized coefficients  
#library("matrixStats")  
#vi <- coef(lm.optimal)/sqrt(colVars(model.matrix(formula(lm.optimal), data=reviews\_df[estimation\_sample,])))  
#StdCoef <- data.frame(StdCoef=vi[2:length(vi)])  
#StdCoef$Variable <- rownames(StdCoef)  
#StdCoef <- StdCoef[order(- StdCoef$StdCoef),]  
#StdCoef$Variable <- factor( StdCoef$Variable, levels = rev( StdCoef$Variable))  
  
#StdCoef <- StdCoef[1:25,]  
#ggplot( StdCoef, aes(Variable,StdCoef)) + geom\_bar(stat = "identity") + coord\_flip() + ylab("Variable importance (standardized coefficients)")

#Variable importance using t-values of the optimal linear model  
library(caret)  
vi <- varImp(lm.optimal) #vs. lm. all  
vi$Variable <- rownames(vi)  
vi <- vi[order(-vi$Overall),]  
vi$Variable <- factor(vi$Variable, levels = rev(vi$Variable))  
  
ggplot(vi[1:25, ], aes(Variable,Overall)) + geom\_bar(stat = "identity") + coord\_flip() + ylab("Variable importance (t-value based) for linear model with optimal features")



#ggplot(vi[(nrow(vi)-24):nrow(vi), ], aes(Variable,Overall)) + geom\_bar(stat = "identity") + coord\_flip() + ylab("Variable importance (t-value) for linear model with all features")

# Backward selection  
#f <- paste("rating ~ Nr\_of\_words + ", allFactors, " + ", allEmotions, " + ", allWords , " + ", allBigrams)  
#lm.fe\_step <- step(lm(f, data = reviews\_df[estimation\_sample,]), direction = "both")  
#summary(lm.fe\_step)

# Random Forest Model   
library("randomForest")

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:Biobase':  
##   
## combine

## The following object is masked from 'package:BiocGenerics':  
##   
## combine

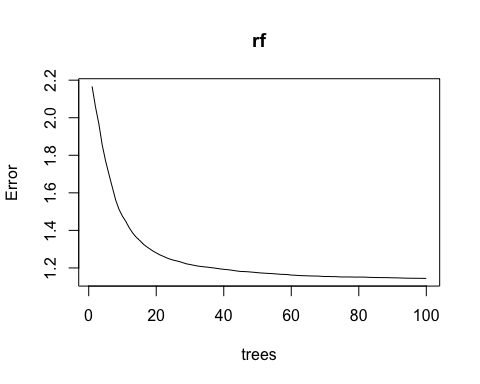
## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:dplyr':  
##   
## combine

f <- paste("rating ~ Nr\_of\_words + ", allFactors, " + ", allEmotions, " + ", allWords , " + ", allBigrams)  
rf = randomForest(formula(f),   
 ntree = 100,  
 data = reviews\_df[estimation\_sample,],  
 importance = TRUE)

## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

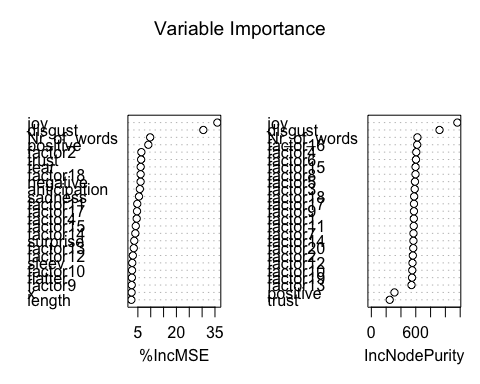
plot(rf)



print(rf) #linear model works better -overfitting, works only for the estimation sample. LINEAR MODEL WINS YAY

##   
## Call:  
## randomForest(formula = formula(f), data = reviews\_df[estimation\_sample, ], ntree = 100, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 100  
## No. of variables tried at each split: 50  
##   
## Mean of squared residuals: 1.144295  
## % Var explained: 9.16

# Variable Importance based on Random Forest Model   
varImpPlot(rf,   
 sort = T,  
 n.var = 25,  
 main = "Variable Importance")



# Predictive Performance of Random Forest Model   
library(caret)  
pred.all <- predict(rf) #mse is better for linear model, accuracy is better. Random forest takes longer  
print("overall\_rmse")

## [1] "overall\_rmse"

mse\_all <- mean((as.numeric(reviews\_df[,"rating"])-pred.all)^2)

## Warning in as.numeric(reviews\_df[, "rating"]) - pred.all: longer object length  
## is not a multiple of shorter object length

print(sqrt(mse\_all))

## [1] 1.201835

pred.est <- predict(rf, reviews\_df[estimation\_sample,])  
print("estimation\_rmse")

## [1] "estimation\_rmse"

mse\_est <- mean((as.numeric(reviews\_df[estimation\_sample,"rating"])-pred.est)^2)  
print(sqrt(mse\_est))

## [1] 0.4469495

pred.test <- predict(rf, reviews\_df[test\_sample,])  
print("test\_mse")

## [1] "test\_mse"

mse\_test <- mean((as.numeric(reviews\_df[test\_sample,"rating"])-pred.test)^2)  
print(sqrt(mse\_test))

## [1] 1.042505

# Lasso regression for all features ----  
  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:qdap':  
##   
## %&%

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1-1

##   
## Attaching package: 'glmnet'

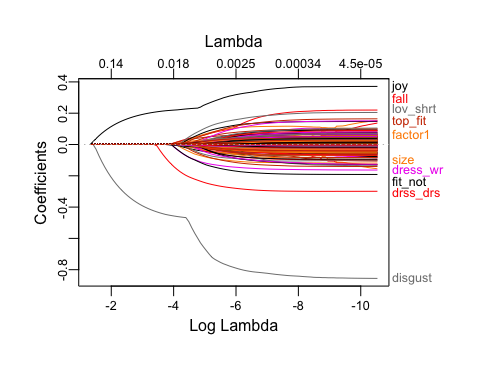
## The following object is masked from 'package:gtools':  
##   
## na.replace

library(plotmo) # for plot\_glmnet

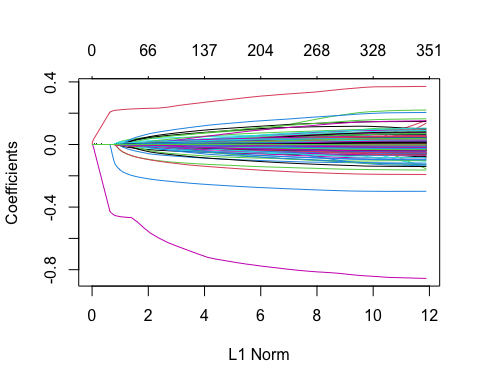
## Loading required package: Formula

## Loading required package: TeachingDemos

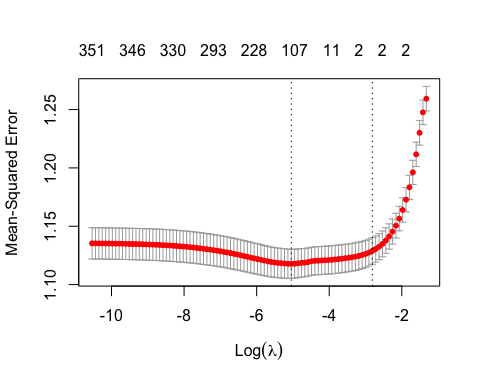
f = paste("~ 0 + Nr\_of\_words + ", allFactors, " \* ", allEmotions, " + ", allWords, " + ", allBigrams)  
  
# Collect explanatory variables in a (large) matrix  
LargeX <- model.matrix(formula(f), data=reviews\_df)  
y <- as.matrix(reviews\_df[estimation\_sample, "rating"])  
  
lasso.mod <- glmnet(LargeX[estimation\_sample,], y, alpha = 1)  
  
plot\_glmnet(lasso.mod)



plot(lasso.mod)



cvfit <- cv.glmnet(LargeX[estimation\_sample,], y, alpha = 1)  
plot(cvfit)



coef(lasso.mod, cvfit$lambda.1se)

## 352 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 3.9879564  
## Nr\_of\_words .   
## factor1 .   
## factor2 .   
## factor3 .   
## factor4 .   
## factor5 .   
## factor6 .   
## factor7 .   
## factor8 .   
## factor9 .   
## factor10 .   
## factor11 .   
## factor12 .   
## factor13 .   
## factor14 .   
## factor15 .   
## factor16 .   
## factor17 .   
## factor18 .   
## factor19 .   
## fall .   
## anger .   
## anticipation .   
## disgust -0.3700904  
## fear .   
## joy 0.1829833  
## sadness .   
## surprise .   
## trust .   
## negative .   
## factor20 .   
## fit .   
## size .   
## wear .   
## love .   
## top .   
## color .   
## fabric .   
## perfect .   
## nice .   
## flatter .   
## soft .   
## comfort .   
## bit .   
## materi .   
## length .   
## shirt .   
## waist .   
## run .   
## jean .   
## cute .   
## petit .   
## short .   
## beauti .   
## sweater .   
## medium .   
## feel .   
## qualiti .   
## x .   
## sleev .   
## bought .   
## retail .   
## skirt .   
## pretti .   
## style .   
## review .   
## pant .   
## cut .   
## store .   
## purchas .   
## design .   
## tight .   
## black .   
## return .   
## true .   
## pictur .   
## summer .   
## super .   
## line .   
## detail .   
## positive .   
## dress\_dress .   
## bodi\_type .   
## size\_x .   
## size\_8 .   
## size\_size .   
## dress\_wear .   
## normal\_size .   
## love\_top .   
## petit\_size .   
## size\_2 .   
## love\_love .   
## arm\_hole .   
## sale\_price .   
## dress\_not .   
## not\_fit .   
## size\_6 .   
## regular\_size .   
## fit\_perfectli .   
## dress\_fit .   
## beauti\_dress .   
## dress\_love .   
## dress\_beauti .   
## fit\_flatter .   
## wear\_size .   
## super\_cute .   
## highli\_recommend .   
## love\_fit .   
## fabric\_soft .   
## fit\_perfect .   
## fit\_nice .   
## wear\_dress .   
## length\_perfect .   
## size\_medium .   
## read\_review .   
## fabric\_nice .   
## fit\_true .   
## true\_size .   
## love\_dress .   
## light\_weight .   
## size\_4 .   
## fit\_not .   
## love\_sweater .   
## super\_soft .   
## size\_fit .   
## not\_tight .   
## love\_color .   
## size\_10 .   
## top\_fit .   
## skinni\_jean .   
## usual\_size .   
## x\_fit .   
## absolut\_love .   
## medium\_fit .   
## dress\_run .   
## top\_not .   
## plan\_wear .   
## not\_flatter .   
## materi\_soft .   
## color\_love .   
## fit\_tt .   
## soft\_comfort .   
## color\_beauti .   
## fell\_love .   
## run\_true .   
## previou\_review .   
## wait\_wear .   
## love\_shirt .   
## run\_size .   
## receiv\_compliment .   
## agre\_review .   
## Nr\_of\_words:fall .   
## Nr\_of\_words:anger .   
## Nr\_of\_words:anticipation .   
## Nr\_of\_words:disgust .   
## Nr\_of\_words:fear .   
## Nr\_of\_words:joy .   
## Nr\_of\_words:sadness .   
## Nr\_of\_words:surprise .   
## Nr\_of\_words:trust .   
## Nr\_of\_words:negative .   
## factor1:fall .   
## factor1:anger .   
## factor1:anticipation .   
## factor1:disgust .   
## factor1:fear .   
## factor1:joy .   
## factor1:sadness .   
## factor1:surprise .   
## factor1:trust .   
## factor1:negative .   
## factor2:fall .   
## factor2:anger .   
## factor2:anticipation .   
## factor2:disgust .   
## factor2:fear .   
## factor2:joy .   
## factor2:sadness .   
## factor2:surprise .   
## factor2:trust .   
## factor2:negative .   
## factor3:fall .   
## factor3:anger .   
## factor3:anticipation .   
## factor3:disgust .   
## factor3:fear .   
## factor3:joy .   
## factor3:sadness .   
## factor3:surprise .   
## factor3:trust .   
## factor3:negative .   
## factor4:fall .   
## factor4:anger .   
## factor4:anticipation .   
## factor4:disgust .   
## factor4:fear .   
## factor4:joy .   
## factor4:sadness .   
## factor4:surprise .   
## factor4:trust .   
## factor4:negative .   
## factor5:fall .   
## factor5:anger .   
## factor5:anticipation .   
## factor5:disgust .   
## factor5:fear .   
## factor5:joy .   
## factor5:sadness .   
## factor5:surprise .   
## factor5:trust .   
## factor5:negative .   
## factor6:fall .   
## factor6:anger .   
## factor6:anticipation .   
## factor6:disgust .   
## factor6:fear .   
## factor6:joy .   
## factor6:sadness .   
## factor6:surprise .   
## factor6:trust .   
## factor6:negative .   
## factor7:fall .   
## factor7:anger .   
## factor7:anticipation .   
## factor7:disgust .   
## factor7:fear .   
## factor7:joy .   
## factor7:sadness .   
## factor7:surprise .   
## factor7:trust .   
## factor7:negative .   
## factor8:fall .   
## factor8:anger .   
## factor8:anticipation .   
## factor8:disgust .   
## factor8:fear .   
## factor8:joy .   
## factor8:sadness .   
## factor8:surprise .   
## factor8:trust .   
## factor8:negative .   
## factor9:fall .   
## factor9:anger .   
## factor9:anticipation .   
## factor9:disgust .   
## factor9:fear .   
## factor9:joy .   
## factor9:sadness .   
## factor9:surprise .   
## factor9:trust .   
## factor9:negative .   
## factor10:fall .   
## factor10:anger .   
## factor10:anticipation .   
## factor10:disgust .   
## factor10:fear .   
## factor10:joy .   
## factor10:sadness .   
## factor10:surprise .   
## factor10:trust .   
## factor10:negative .   
## factor11:fall .   
## factor11:anger .   
## factor11:anticipation .   
## factor11:disgust .   
## factor11:fear .   
## factor11:joy .   
## factor11:sadness .   
## factor11:surprise .   
## factor11:trust .   
## factor11:negative .   
## factor12:fall .   
## factor12:anger .   
## factor12:anticipation .   
## factor12:disgust .   
## factor12:fear .   
## factor12:joy .   
## factor12:sadness .   
## factor12:surprise .   
## factor12:trust .   
## factor12:negative .   
## factor13:fall .   
## factor13:anger .   
## factor13:anticipation .   
## factor13:disgust .   
## factor13:fear .   
## factor13:joy .   
## factor13:sadness .   
## factor13:surprise .   
## factor13:trust .   
## factor13:negative .   
## factor14:fall .   
## factor14:anger .   
## factor14:anticipation .   
## factor14:disgust .   
## factor14:fear .   
## factor14:joy .   
## factor14:sadness .   
## factor14:surprise .   
## factor14:trust .   
## factor14:negative .   
## factor15:fall .   
## factor15:anger .   
## factor15:anticipation .   
## factor15:disgust .   
## factor15:fear .   
## factor15:joy .   
## factor15:sadness .   
## factor15:surprise .   
## factor15:trust .   
## factor15:negative .   
## factor16:fall .   
## factor16:anger .   
## factor16:anticipation .   
## factor16:disgust .   
## factor16:fear .   
## factor16:joy .   
## factor16:sadness .   
## factor16:surprise .   
## factor16:trust .   
## factor16:negative .   
## factor17:fall .   
## factor17:anger .   
## factor17:anticipation .   
## factor17:disgust .   
## factor17:fear .   
## factor17:joy .   
## factor17:sadness .   
## factor17:surprise .   
## factor17:trust .   
## factor17:negative .   
## factor18:fall .   
## factor18:anger .   
## factor18:anticipation .   
## factor18:disgust .   
## factor18:fear .   
## factor18:joy .   
## factor18:sadness .   
## factor18:surprise .   
## factor18:trust .   
## factor18:negative .   
## factor19:fall .   
## factor19:anger .   
## factor19:anticipation .   
## factor19:disgust .   
## factor19:fear .   
## factor19:joy .   
## factor19:sadness .   
## factor19:surprise .   
## factor19:trust .   
## factor19:negative .

par <- predict(lasso.mod, s = cvfit$lambda.min, type='coefficients')  
nnzero(par)

## [1] 118

par <- predict(lasso.mod, s = cvfit$lambda.1se, type='coefficients')  
nnzero(par)

## [1] 3

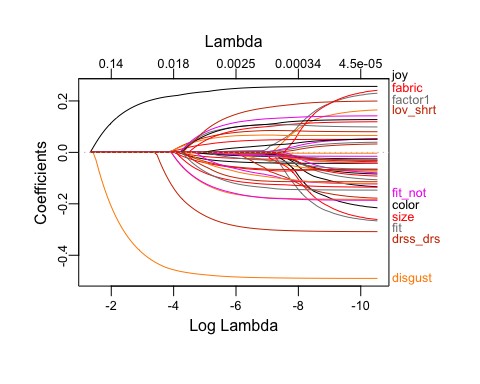
lasso.pred <- predict(lasso.mod, s = cvfit$lambda.1se, newx = LargeX[estimation\_sample,])  
lasso.pred.test <- predict(lasso.mod, s = cvfit$lambda.1se, newx = LargeX[test\_sample,])  
  
#RMSE Estimation  
MSE\_lasso\_all\_est <- mean((reviews\_df[estimation\_sample, "rating"]-lasso.pred)^2)  
print(sqrt(MSE\_lasso\_all\_est))

## [1] 1.062202

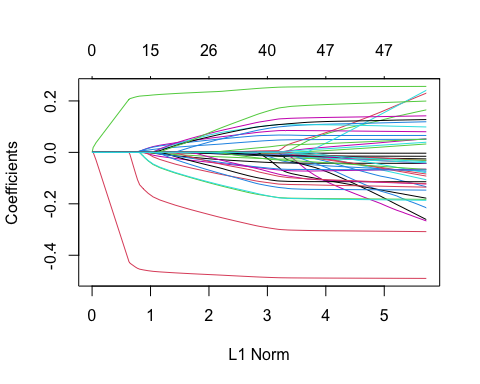
MSE\_lasso\_all <- mean((reviews\_df[test\_sample, "rating"]-lasso.pred.test)^2)  
print(sqrt(MSE\_lasso\_all))

## [1] 1.040398

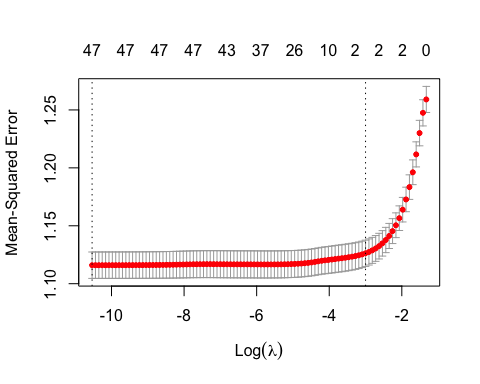
# Lasso regression for optimal features ----  
  
library(glmnet)  
library(plotmo) # for plot\_glmnet  
  
f\_optimal <- "~ 0 + Nr\_of\_words + factor1 + factor3 + factor6 +  
 factor7 + factor8 + factor9 + factor10 + factor13 + factor16 +   
 factor19 + fall + anticipation + disgust + joy + trust +   
 factor20 + fit + size + wear + love + color + fabric + nice +   
 flatter + jean + feel + style + purchas + line + positive +   
 dress\_dress + dress\_wear + arm\_hole + size\_6 + dress\_beauti +   
 wear\_size + super\_cute + true\_size + love\_dress + fit\_not +   
 love\_sweater + top\_fit + skinni\_jean + love\_shirt + agre\_review +   
 bought"  
  
# Collect explanatory variables in a (large) matrix  
LargeX <- model.matrix(formula(f\_optimal), data=reviews\_df)  
y <- as.matrix(reviews\_df[estimation\_sample, "rating"])  
  
lasso.mod <- glmnet(LargeX[estimation\_sample,], y, alpha = 1)  
  
plot\_glmnet(lasso.mod)



plot(lasso.mod)



cvfit <- cv.glmnet(LargeX[estimation\_sample,], y, alpha = 1)  
plot(cvfit)



coef(lasso.mod, cvfit$lambda.1se)

## 48 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 3.9789759  
## Nr\_of\_words .   
## factor1 .   
## factor3 .   
## factor6 .   
## factor7 .   
## factor8 .   
## factor9 .   
## factor10 .   
## factor13 .   
## factor16 .   
## factor19 .   
## fall .   
## anticipation .   
## disgust -0.3916495  
## joy 0.1921913  
## trust .   
## factor20 .   
## fit .   
## size .   
## wear .   
## love .   
## color .   
## fabric .   
## nice .   
## flatter .   
## jean .   
## feel .   
## style .   
## purchas .   
## line .   
## positive .   
## dress\_dress .   
## dress\_wear .   
## arm\_hole .   
## size\_6 .   
## dress\_beauti .   
## wear\_size .   
## super\_cute .   
## true\_size .   
## love\_dress .   
## fit\_not .   
## love\_sweater .   
## top\_fit .   
## skinni\_jean .   
## love\_shirt .   
## agre\_review .   
## bought .

par <- predict(lasso.mod, s = cvfit$lambda.min, type='coefficients')  
nnzero(par)

## [1] 48

par <- predict(lasso.mod, s = cvfit$lambda.1se, type='coefficients')  
nnzero(par)

## [1] 3

lasso.pred <- predict(lasso.mod, s = cvfit$lambda.1se, newx = LargeX[estimation\_sample,])  
lasso.pred.test <- predict(lasso.mod, s = cvfit$lambda.1se, newx = LargeX[test\_sample,])  
  
MSE\_lasso\_optimal\_est <- mean((reviews\_df[estimation\_sample, "rating"]-lasso.pred)^2)  
print(sqrt(MSE\_lasso\_optimal\_est))

## [1] 1.061018

MSE\_lasso\_optimal <-mean((reviews\_df[test\_sample, "rating"]-lasso.pred.test)^2)   
print(sqrt(MSE\_lasso\_optimal))

## [1] 1.039305