Amazon Reviews Sentiment Analysis

Mirza Naseh Ahmad

3/8/2021

#setting the working directory  
setwd('E:\\Ryerson\\CIND820\\Code')  
#Loading the Data  
  
#downloading the data from the source  
#fn <- "amazon\_reviews\_us\_Automotive\_v1\_00.tsv.gz"  
  
#if ( !file.exists(fn) ) {  
# download.file("https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon\_reviews\_us\_Automotive\_v1\_00.tsv.gz",  
# fn)  
# untar(fn)  
#}

datacomplete<-as.data.frame(fread('amazon\_reviews\_us\_Automotive\_v1\_00.tsv'),stringsAsFactors = FALSE, quote ="")

## Warning in fread("amazon\_reviews\_us\_Automotive\_v1\_00.tsv"): Found and  
## resolved improper quoting out-of-sample. First healed line 4209: <<US 35675398  
## R5YPUKJBX85J0 B005G82C8E 826586699 "THE E Z SLIDE SHADE" for Retractable  
## Sunshades Available for All Car Models. Please send the model and car year.  
## Automotive 5 5 6 N Y Best choice on the market. This is so convenient!!! I live  
## in phoenix AZ where the internal temperature of cars can bake cookies. literally  
## look it up on YouTube. I hate fussing with all the sun shades putting them  
## behind the seat where half the time you just don't use them. I looked at other  
## styles of permanent sun >>. If the fields are not quoted (e.g. field separator  
## does not appear within any field), try quote="" to avoid this warning.

sampledata<-summary(datacomplete)  
  
skimmed <- skim\_to\_wide(datacomplete)

## Warning: 'skim\_to\_wide' is deprecated.  
## Use 'skim()' instead.  
## See help("Deprecated")

skimmed[, c(1:5, 9:11, 13, 15:16)]

Data summary

|  |  |
| --- | --- |
| Name | .data |
| Number of rows | 3514942 |
| Number of columns | 15 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 9 |
| Date | 1 |
| numeric | 5 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | whitespace |
| marketplace | 0 | 1 | 2 | 0 |
| review\_id | 0 | 1 | 9 | 0 |
| product\_id | 0 | 1 | 10 | 0 |
| product\_title | 2 | 1 | 1 | 0 |
| product\_category | 0 | 1 | 10 | 0 |
| vine | 0 | 1 | 1 | 0 |
| verified\_purchase | 0 | 1 | 1 | 0 |
| review\_headline | 2 | 1 | 0 | 0 |
| review\_body | 0 | 1 | 0 | 0 |

**Variable type: Date**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | n\_unique |
| review\_date | 0 | 1 | 1999-10-24 | 2015-08-31 | 4156 |

**Variable type: numeric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | sd | p0 |
| customer\_id | 0 | 1 | 15529237.55 | 10004 |
| product\_parent | 0 | 1 | 289226012.24 | 771 |
| star\_rating | 0 | 1 | 1.26 | 1 |
| helpful\_votes | 0 | 1 | 16.97 | 0 |
| total\_votes | 0 | 1 | 17.46 | 0 |

vine<-as.factor(datacomplete$vine)  
summary(vine)

## N Y   
## 3509017 5925

#Marketplace where the review was written.  
marketplace<- as.factor(datacomplete$marketplace)  
summary(marketplace)

## US   
## 3514942

#total Unique Customers  
uniquecustomers <- unique(datacomplete$customer\_id)  
length(uniquecustomers)

## [1] 1907652

#total Unique products reviewed  
uniqueproducts <- unique(datacomplete$product\_id)  
length(uniqueproducts)

## [1] 762982

#Textlength   
datacomplete$reviewlength <- nchar(datacomplete$review\_body)  
  
summary(datacomplete$reviewlength)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 51.0 128.0 215.6 250.0 27588.0

#Highest rated products.  
top\_rated\_products <- datacomplete %>%  
 group\_by(product\_id) %>%   
 summarize(count\_votes = n()) %>%   
 arrange(desc(count\_votes))  
  
top\_rated\_products1 <- top\_rated\_products[top\_rated\_products$count\_votes > 10 ,]  
summary(top\_rated\_products1)

## product\_id count\_votes   
## Length:53852 Min. : 11.00   
## Class :character 1st Qu.: 14.00   
## Mode :character Median : 20.00   
## Mean : 38.69   
## 3rd Qu.: 37.00   
## Max. :4894.00

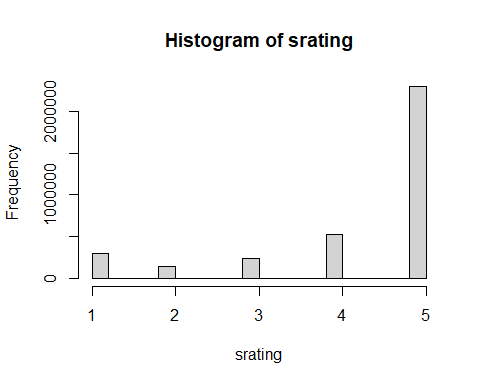
head(top\_rated\_products1)

## # A tibble: 6 x 2  
## product\_id count\_votes  
## <chr> <int>  
## 1 B005NLQAHS 4894  
## 2 B000CITK8S 4422  
## 3 B001LHVOVK 3694  
## 4 B00068XCQU 2688  
## 5 B001AIZ5HY 2483  
## 6 B00080QHMM 2069

data<- datacomplete[datacomplete$product\_id %in% top\_rated\_products1$product\_id ,]  
  
str(data)

## 'data.frame': 2083289 obs. of 16 variables:  
## $ marketplace : chr "US" "US" "US" "US" ...  
## $ customer\_id : int 42462164 52570308 38200102 184627 40946484 35335277 2583392 10127903 17937309 32813839 ...  
## $ review\_id : chr "R3NORADVJO6IE6" "R2DA9DOT03UW6I" "R2OGCH681EQHU6" "R1DB5DA7CWWTI8" ...  
## $ product\_id : chr "B000C7S0TO" "B000GKD5NI" "B009SDA7TE" "B0002JMAKW" ...  
## $ product\_parent : int 907684644 105401756 728471129 267002949 389524802 816815445 732072782 533179941 126169507 897802619 ...  
## $ product\_title : chr "Spectra Premium CU1909 Complete Radiator for Toyota Camry" "Suncutters Rear Window Shade" "Lug Nuts Landcruiser Tundra OEM Mag 14x1.5 Thread Set of 20 Pcs" "Castrol 12614 Dot 4 Synthetic Brake Fluid" ...  
## $ product\_category : chr "Automotive" "Automotive" "Automotive" "Automotive" ...  
## $ star\_rating : int 5 5 5 5 5 4 5 5 5 3 ...  
## $ helpful\_votes : int 0 2 0 0 0 0 0 0 0 0 ...  
## $ total\_votes : int 0 3 0 0 0 0 0 0 0 0 ...  
## $ vine : chr "N" "N" "N" "N" ...  
## $ verified\_purchase: chr "Y" "Y" "Y" "Y" ...  
## $ review\_headline : chr "Five Stars" "Good for the price. Fits fairly good on 2010 Toyota Camry" "Five Stars" "convenient, fast delivery," ...  
## $ review\_body : chr "Put it in fine, no problems. Shipping was decent 5 days." "Good for the price! So far I have put it up in the back window of my 2010 Camry and after several times using "| \_\_truncated\_\_ "Fit perfectly on my 2012 Tundra with stock alu rims." "convenient ,fast delivery, and fair price" ...  
## $ review\_date : IDate, format: "2015-08-31" "2015-08-31" ...  
## $ reviewlength : int 56 691 52 41 161 431 96 132 156 35 ...

#Star Rating  
srating<-datacomplete$star\_rating  
hist(srating)



prop.table(table(srating))

## srating  
## 1 2 3 4 5   
## 0.08531208 0.04203967 0.06824750 0.14983604 0.65456471

#Data Prep  
datacomplete$reviewlength <- nchar(datacomplete$review\_body)  
  
summary(datacomplete$reviewlength)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 51.0 128.0 215.6 250.0 27588.0

str(datacomplete)

## 'data.frame': 3514942 obs. of 16 variables:  
## $ marketplace : chr "US" "US" "US" "US" ...  
## $ customer\_id : int 36075342 42462164 21241933 52570308 38200102 34866169 184627 2975964 40946484 12554469 ...  
## $ review\_id : chr "RAB23OVFNCXZQ" "R3NORADVJO6IE6" "R299F4SO98S5OO" "R2DA9DOT03UW6I" ...  
## $ product\_id : chr "B00LPRXQ4Y" "B000C7S0TO" "B000CO9WE4" "B000GKD5NI" ...  
## $ product\_parent : int 339193102 907684644 752246352 105401756 728471129 962286893 267002949 570789093 389524802 21425394 ...  
## $ product\_title : chr "17\" 2003-2006 Ford EXPEDITION Ford F150 2004-2008 OEM Chrome Center Cap Hubcap Wheel Cover 3518 5L34-1A096-GA" "Spectra Premium CU1909 Complete Radiator for Toyota Camry" "K&N E-4665 High Performance Replacement Industrial Air Filter" "Suncutters Rear Window Shade" ...  
## $ product\_category : chr "Automotive" "Automotive" "Automotive" "Automotive" ...  
## $ star\_rating : int 1 5 5 5 5 5 5 5 5 1 ...  
## $ helpful\_votes : int 0 0 1 2 0 2 0 0 0 0 ...  
## $ total\_votes : int 0 0 1 3 0 2 0 0 0 0 ...  
## $ vine : chr "N" "N" "N" "N" ...  
## $ verified\_purchase: chr "Y" "Y" "Y" "Y" ...  
## $ review\_headline : chr "As it was used," "Five Stars" "Great fit and performance on the surface drive motor." "Good for the price. Fits fairly good on 2010 Toyota Camry" ...  
## $ review\_body : chr "As it was used, the method that Ford used to attach it to the wheel was worn and thus the cap flew off into the"| \_\_truncated\_\_ "Put it in fine, no problems. Shipping was decent 5 days." "Fit wonderfully on my 36HP Pro-Drive motor." "Good for the price! So far I have put it up in the back window of my 2010 Camry and after several times using "| \_\_truncated\_\_ ...  
## $ review\_date : IDate, format: "2015-08-31" "2015-08-31" ...  
## $ reviewlength : int 134 56 43 691 52 108 41 7 161 103 ...

#removing "NA" values  
  
datacomplete<-na.omit(datacomplete)  
  
skimmed <- skim\_to\_wide(datacomplete)

## Warning: 'skim\_to\_wide' is deprecated.  
## Use 'skim()' instead.  
## See help("Deprecated")

skimmed[, c(1:5, 9:11, 13, 15:16)]

Data summary

|  |  |
| --- | --- |
| Name | .data |
| Number of rows | 3514938 |
| Number of columns | 16 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 9 |
| Date | 1 |
| numeric | 6 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | whitespace |
| marketplace | 0 | 1 | 2 | 0 |
| review\_id | 0 | 1 | 9 | 0 |
| product\_id | 0 | 1 | 10 | 0 |
| product\_title | 0 | 1 | 1 | 0 |
| product\_category | 0 | 1 | 10 | 0 |
| vine | 0 | 1 | 1 | 0 |
| verified\_purchase | 0 | 1 | 1 | 0 |
| review\_headline | 0 | 1 | 0 | 0 |
| review\_body | 0 | 1 | 0 | 0 |

**Variable type: Date**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | n\_unique |
| review\_date | 0 | 1 | 1999-10-24 | 2015-08-31 | 4156 |

**Variable type: numeric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | sd | p0 |
| customer\_id | 0 | 1 | 15529241.21 | 10004 |
| product\_parent | 0 | 1 | 289226013.63 | 771 |
| star\_rating | 0 | 1 | 1.26 | 1 |
| helpful\_votes | 0 | 1 | 16.97 | 0 |
| total\_votes | 0 | 1 | 17.46 | 0 |
| reviewlength | 0 | 1 | 328.98 | 0 |

#removing zero text values  
zerotext<-datacomplete[datacomplete$reviewlength == 0 ,]  
  
  
datacomplete<-datacomplete[datacomplete$reviewlength != 0,]  
  
  
  
#filtering out non verified purchases  
vpcount = table(datacomplete$verified\_purchase)  
vpcount = as.data.frame(vpcount)  
names(vpcount)[1] = 'Verified purchase'  
vpcount

## Verified purchase Freq  
## 1 N 286461  
## 2 Y 3227966

datavp<-datacomplete[datacomplete$verified\_purchase != 'N' & datacomplete$product\_id %in% top\_rated\_products1$product\_id,]  
  
table(datavp$verified\_purchase)

##   
## Y   
## 1917385

summary(datavp$reviewlength)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 54.0 130.0 211.7 249.0 27177.0

glimpse(datavp)

## Rows: 1,917,385  
## Columns: 16  
## $ marketplace <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US"~  
## $ customer\_id <int> 42462164, 52570308, 38200102, 184627, 40946484, 3533~  
## $ review\_id <chr> "R3NORADVJO6IE6", "R2DA9DOT03UW6I", "R2OGCH681EQHU6"~  
## $ product\_id <chr> "B000C7S0TO", "B000GKD5NI", "B009SDA7TE", "B0002JMAK~  
## $ product\_parent <int> 907684644, 105401756, 728471129, 267002949, 38952480~  
## $ product\_title <chr> "Spectra Premium CU1909 Complete Radiator for Toyota~  
## $ product\_category <chr> "Automotive", "Automotive", "Automotive", "Automotiv~  
## $ star\_rating <int> 5, 5, 5, 5, 5, 4, 5, 5, 5, 3, 1, 5, 1, 5, 5, 5, 5, 5~  
## $ helpful\_votes <int> 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  
## $ total\_votes <int> 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0~  
## $ vine <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N~  
## $ verified\_purchase <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y~  
## $ review\_headline <chr> "Five Stars", "Good for the price. Fits fairly good ~  
## $ review\_body <chr> "Put it in fine, no problems. Shipping was decent 5 ~  
## $ review\_date <date> 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 201~  
## $ reviewlength <int> 56, 691, 52, 41, 161, 431, 96, 132, 156, 35, 294, 47~

#Stratified Sampling  
set.seed(1000)  
options(stringsASFacgtors = FALSE)  
sr1<- filter(datavp, star\_rating == 1)  
sr2<- filter(datavp, star\_rating == 2)  
sr3<- filter(datavp, star\_rating == 3)  
sr4<- filter(datavp, star\_rating == 4)  
sr5<- filter(datavp, star\_rating == 5)  
  
sampledata1<- sample\_n(sr1,74829 , replace = FALSE)  
sampledata2<- sample\_n(sr2,74829 , replace = FALSE)  
sampledata3<- sample\_n(sr3,74829 , replace = FALSE)  
sampledata4<- sample\_n(sr4,74829 , replace = FALSE)  
sampledata5<- sample\_n(sr5,74829 , replace = FALSE)  
  
sampledata <- rbind(sampledata1, sampledata2, sampledata3, sampledata4, sampledata5)  
sampledata <- data.table(rating = sampledata$star\_rating ,review = sampledata$review\_body , reviewlength = sampledata$reviewlength)  
  
data <- data.table(rating = datavp$star\_rating ,review = datavp$review\_body , reviewlength = datavp$reviewlength)  
  
set.seed(10)  
datareduction <- sample(1:nrow(data), 0.01 \* nrow(data))  
data<- data[datareduction, ]  
  
sampledatareduction <- sample(1:nrow(sampledata), 0.05 \* nrow(sampledata))  
sampledata<-sampledata[sampledatareduction, ]  
  
glimpse(data)

## Rows: 19,173  
## Columns: 3  
## $ rating <int> 5, 5, 5, 4, 5, 4, 4, 5, 5, 2, 3, 5, 5, 5, 5, 5, 5, 4, 5, ~  
## $ review <chr> "Perfect fit, looks great.", "Excellent products from thi~  
## $ reviewlength <int> 25, 129, 518, 593, 527, 135, 4, 192, 171, 254, 300, 21, 1~

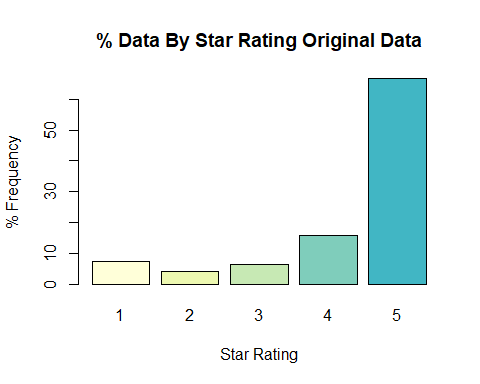
glimpse(sampledata)

## Rows: 18,707  
## Columns: 3  
## $ rating <int> 4, 1, 4, 2, 2, 4, 4, 2, 1, 1, 3, 5, 4, 5, 4, 5, 3, 5, 2, ~  
## $ review <chr> "alls good", "It worked twice and stopped working. Cant c~  
## $ reviewlength <int> 9, 175, 218, 524, 862, 157, 41, 269, 113, 116, 166, 247, ~

round(100\*prop.table(table(data$rating)), digits = 2)

##   
## 1 2 3 4 5   
## 7.27 3.94 6.40 15.66 66.72

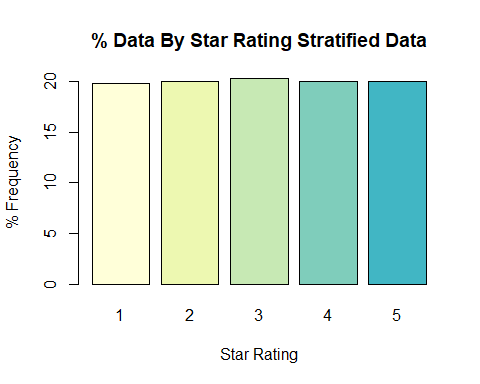
SRFactor <- as.factor(data$rating)  
barplot(round(100\*prop.table(table(data$rating)), digits = 2), xlab = "Star Rating", ylab = "% Frequency", main = "% Data By Star Rating Original Data", col = c(brewer.pal(9,"YlGnBu")))



round(100\*prop.table(table(sampledata$rating)), digits = 2)

##   
## 1 2 3 4 5   
## 19.83 19.94 20.26 20.01 19.96

SRFactor <- as.factor(sampledata$rating)  
barplot(round(100\*prop.table(table(sampledata$rating)), digits = 2), xlab = "Star Rating", ylab = "% Frequency", main = "% Data By Star Rating Stratified Data", col = c(brewer.pal(9,"YlGnBu")))



glimpse(data)

## Rows: 19,173  
## Columns: 3  
## $ rating <int> 5, 5, 5, 4, 5, 4, 4, 5, 5, 2, 3, 5, 5, 5, 5, 5, 5, 4, 5, ~  
## $ review <chr> "Perfect fit, looks great.", "Excellent products from thi~  
## $ reviewlength <int> 25, 129, 518, 593, 527, 135, 4, 192, 171, 254, 300, 21, 1~

glimpse(sampledata)

## Rows: 18,707  
## Columns: 3  
## $ rating <int> 4, 1, 4, 2, 2, 4, 4, 2, 1, 1, 3, 5, 4, 5, 4, 5, 3, 5, 2, ~  
## $ review <chr> "alls good", "It worked twice and stopped working. Cant c~  
## $ reviewlength <int> 9, 175, 218, 524, 862, 157, 41, 269, 113, 116, 166, 247, ~

#Splitting the data into training and test set (70/30 split)  
  
set.seed(1002)  
indexes<- createDataPartition(data$rating, times = 1 ,p = 0.7, list = FALSE)  
train<-data[indexes,]  
test <- data[-indexes,]  
train <- data.table(rating = train$rating ,review = train$review , reviewlength = train$reviewlength)  
test <- data.table(rating = test$rating ,review = test$review , reviewlength = test$reviewlength)  
  
glimpse(train)

## Rows: 13,422  
## Columns: 3  
## $ rating <int> 5, 5, 5, 4, 5, 5, 2, 5, 5, 5, 4, 4, 4, 1, 4, 4, 5, 3, 2, ~  
## $ review <chr> "Excellent products from this company. Have purchased thr~  
## $ reviewlength <int> 129, 518, 527, 4, 192, 171, 254, 127, 337, 340, 8, 222, 1~

glimpse(test)

## Rows: 5,751  
## Columns: 3  
## $ rating <int> 5, 4, 4, 3, 5, 5, 5, 5, 4, 4, 5, 5, 1, 5, 5, 4, 5, 5, 4, ~  
## $ review <chr> "Perfect fit, looks great.", "About what I expected, size~  
## $ reviewlength <int> 25, 593, 135, 300, 21, 98, 277, 956, 116, 111, 318, 33, 1~

set.seed(1001)  
indexes1<- createDataPartition(sampledata$rating, times = 1 ,p = 0.7, list = FALSE)  
train1 <- sampledata[indexes1,]  
test1 <- sampledata[-indexes1,]  
  
glimpse(train1)

## Rows: 13,096  
## Columns: 3  
## $ rating <int> 4, 1, 4, 2, 2, 4, 4, 2, 1, 1, 5, 5, 4, 5, 3, 5, 2, 2, 3, ~  
## $ review <chr> "alls good", "It worked twice and stopped working. Cant c~  
## $ reviewlength <int> 9, 175, 218, 524, 862, 157, 41, 269, 113, 116, 247, 212, ~

glimpse(test1)

## Rows: 5,611  
## Columns: 3  
## $ rating <int> 3, 4, 1, 4, 2, 5, 4, 4, 4, 4, 1, 4, 1, 4, 2, 2, 2, 2, 3, ~  
## $ review <chr> "I wish people would publish which vehicle it worked with~  
## $ reviewlength <int> 166, 205, 218, 104, 533, 100, 520, 32, 93, 395, 254, 347,~

#Splitting Data into Positive, Negative and neutral reviews, by star rating  
  
sampledatay3 <-train[train$rating == 3, ]  
sampledatay5 <-train[train$rating > 3 ,]  
sampledatay1 <-train[train$rating <3 ,]  
glimpse(sampledatay3)

## Rows: 863  
## Columns: 3  
## $ rating <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~  
## $ review <chr> "Works but I wish the scent was better.", "Based off the ~  
## $ reviewlength <int> 38, 288, 105, 380, 69, 9, 130, 241, 190, 416, 63, 7, 121,~

#stratified sample  
sampledatay31 <-train1[train1$rating == 3 ,]  
sampledatay51<-train1[train1$rating > 3 ,]  
sampledatay11 <-train1[train1$rating <3 ,]  
glimpse(sampledatay31)

## Rows: 2,653  
## Columns: 3  
## $ rating <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~  
## $ review <chr> "It'll do in a pinch. Easy installation, but I added 3 m~  
## $ reviewlength <int> 215, 17, 187, 89, 116, 326, 109, 111, 591, 548, 748, 18, ~

#cleaning up the text and removing special characters  
  
fix.contractions <- function(doc) {  
 doc <- gsub("won't", "will not", doc)  
 doc <- gsub("can't", "can not", doc)  
 doc <- gsub("n't", " not", doc)  
 doc <- gsub("'ll", " will", doc)  
 doc <- gsub("'re", " are", doc)  
 doc <- gsub("'ve", " have", doc)  
 doc <- gsub("'m", " am", doc)  
 doc <- gsub("'d", " would", doc)  
 doc <- gsub("<br />", "", doc)  
 doc <- gsub("\n", "", doc)  
 # 's could be 'is' or could be possessive: it has no expansion  
 doc <- gsub("'s", "", doc)  
 doc<- gsub("miss.","",doc)  
 doc<- gsub("mr.","",doc)  
 return(doc)  
}  
removeSpecialChars <- function(x) gsub("[^a-zA-Z0-9 ]", " ", x)  
  
reduced\_data <- data.table(sampledatay3)  
reduced\_data5 <- data.table(sampledatay5)  
reduced\_data1 <- data.table(sampledatay1)  
  
  
reduced\_data$review <- tolower(reduced\_data$review)  
reduced\_data$review <- sapply(sampledatay3$review, removeSpecialChars)  
reduced\_data$review <- sapply(reduced\_data$review, fix.contractions)  
  
reduced\_data5$review <- tolower(reduced\_data5$review)  
reduced\_data5$review <- sapply(sampledatay5$review, removeSpecialChars)  
reduced\_data5$review <- sapply(reduced\_data5$review, fix.contractions)  
  
  
  
reduced\_data1$review <- sapply(sampledatay1$review, removeSpecialChars)  
reduced\_data1$review <- sapply(reduced\_data1$review, fix.contractions)  
reduced\_data1$review <- tolower(reduced\_data1$review)  
  
reviewtext<-reduced\_data$review  
reviewtext5<-reduced\_data5$review  
reviewtext1<-reduced\_data1$review  
  
glimpse(reviewtext)

## chr [1:863] "Works but I wish the scent was better " ...

revtext <- data.table(words = c(reviewtext))  
revtext5 <- data.table(words = c(reviewtext5))  
revtext1 <- data.table(words = c(reviewtext1))  
  
glimpse(revtext)

## Rows: 863  
## Columns: 1  
## $ words <chr> "Works but I wish the scent was better ", "Based off the picture~

summary(revtext)

## words   
## Length:863   
## Class :character   
## Mode :character

set.seed(100)  
alltext<-c(revtext,revtext5, revtext1)  
corpus <- VCorpus(VectorSource((alltext)))  
corpus <- tm\_map(corpus, removePunctuation)  
corpus <- tm\_map(corpus, content\_transformer(tolower))  
corpus <- tm\_map(corpus, removeNumbers)  
corpus <- tm\_map(corpus, stripWhitespace)  
corpus <- tm\_map(corpus, removeWords, stopwords('english'))  
corpus <- tm\_map(corpus, stemDocument)  
corpus <- tm\_map(corpus, PlainTextDocument)  
tdm <- TermDocumentMatrix(corpus)  
m <- as.matrix(tdm)  
v <- sort(rowSums(m),decreasing=TRUE)  
d <- data.frame(word = names(v),freq=v)  
  
wordcloud(d$word,d$freq,c(3,.51),2,100,FALSE,.1,colors = brewer.pal(8,"Dark2"))



#AFINN Lexicon   
  
rta <- data.table(review = c(revtext$words))  
review\_sentiment <- rta %>%  
 unnest\_tokens(word, review, token = "words") %>%  
 inner\_join(get\_sentiments("afinn")) %>%  
 filter(!nchar(word) < 3) %>%   
 anti\_join(stop\_words)

## Joining, by = "word"  
## Joining, by = "word"

glimpse(review\_sentiment)

## Rows: 1,580  
## Columns: 2  
## $ word <chr> "easy", "bright", "pretty", "comfortable", "protect", "true", "b~  
## $ value <dbl> 1, 1, 1, 2, 1, 2, 1, 2, 3, 1, 1, 2, -3, -3, 4, 2, 3, -1, 3, -1, ~

rta5 <- data.table(review = c(revtext5$words))  
review\_sentiment5 <- rta5 %>%  
 unnest\_tokens(word, review, token = "words") %>%  
 inner\_join(get\_sentiments("afinn")) %>%  
 filter(!nchar(word) < 3) %>%   
 anti\_join(stop\_words)

## Joining, by = "word"  
## Joining, by = "word"

rta1 <- data.table(review = c(revtext1$words))  
review\_sentiment1 <- rta1 %>%  
 unnest\_tokens(word, review, token = "words") %>%  
 inner\_join(get\_sentiments("afinn")) %>%  
 filter(!nchar(word) < 3) %>%   
 anti\_join(stop\_words)

## Joining, by = "word"  
## Joining, by = "word"

count<-table(review\_sentiment5$value + review\_sentiment$value + review\_sentiment1$value)

## Warning in review\_sentiment5$value + review\_sentiment$value: longer object  
## length is not a multiple of shorter object length

## Warning in review\_sentiment5$value + review\_sentiment$value +  
## review\_sentiment1$value: longer object length is not a multiple of shorter  
## object length

nchar(review\_sentiment1)

## word value   
## 29930 11113

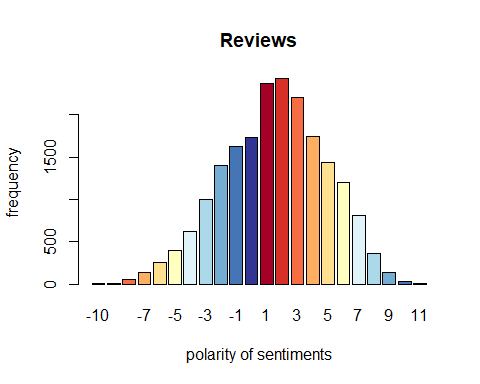
nchar(review\_sentiment5)

## word value   
## 193222 64665

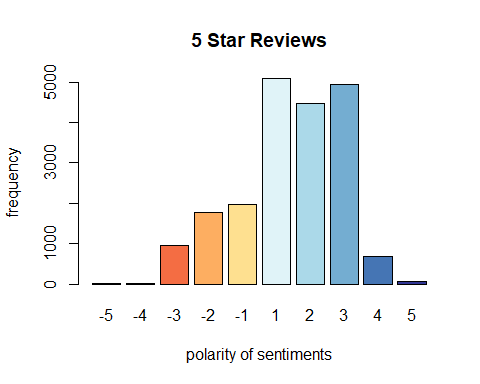
nchar(review\_sentiment)

## word value   
## 14989 5356

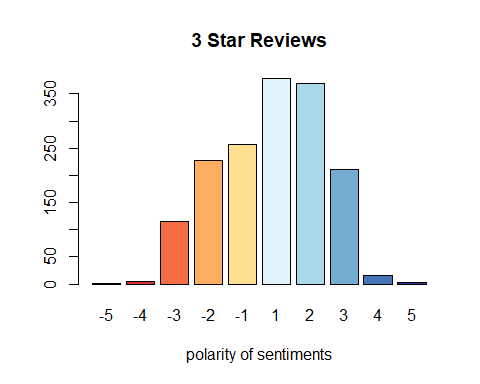
barplot(count, main="Reviews", xlab="polarity of sentiments",ylab = "frequency", col = brewer.pal(11,"RdYlBu"))



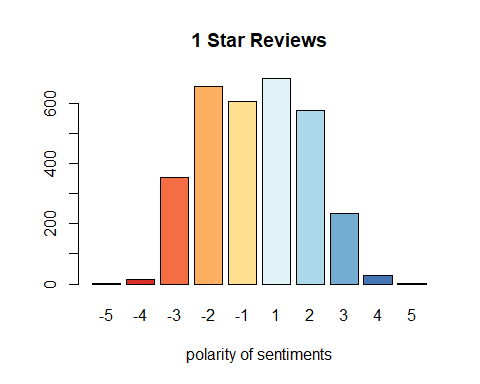
count<-table(review\_sentiment5$value)  
  
barplot(count, main="5 Star Reviews",  
 xlab="polarity of sentiments",ylab = "frequency", col = brewer.pal(10,"RdYlBu"))



count<-table(review\_sentiment$value)  
  
barplot(count, main="3 Star Reviews",  
 xlab="polarity of sentiments", col = brewer.pal(10,"RdYlBu"))



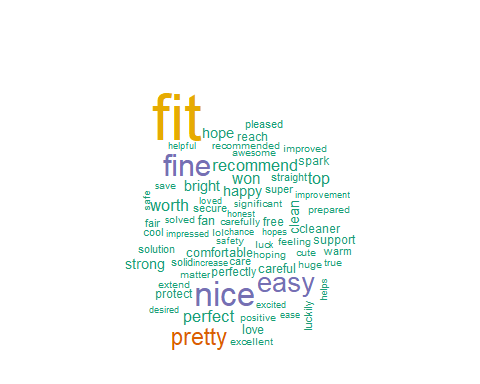
count<-table(review\_sentiment1$value)  
  
barplot(count, main="1 Star Reviews",  
 xlab="polarity of sentiments", col = brewer.pal(10,"RdYlBu"))



p\_sent<-review\_sentiment[review\_sentiment$value>= 0,]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
wordcloud(p\_sent$word,max.words = 100,colors=brewer.pal(6,"Dark2"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents



n\_sent<-review\_sentiment[review\_sentiment$value< 0,]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
wordcloud(n\_sent$word,max.words = 100,colors=brewer.pal(6,"Dark2"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents  
  
## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in wordcloud(n\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## broke could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## difficult could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## wrong could not be fit on page. It will not be plotted.



p\_sent5<-review\_sentiment5[review\_sentiment5$value>= 0,]  
  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
wordcloud(p\_sent5$word,max.words = 100,colors=brewer.pal(8,"Dark2"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents

## Warning in wordcloud(p\_sent5$word, max.words = 100, colors = brewer.pal(8, :  
## recommend could not be fit on page. It will not be plotted.

## Warning in wordcloud(p\_sent5$word, max.words = 100, colors = brewer.pal(8, : fit  
## could not be fit on page. It will not be plotted.



p\_sent1<-review\_sentiment1[review\_sentiment1$value>= 0,]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
wordcloud(p\_sent1$word,max.words = 100,colors=brewer.pal(8,"Dark2"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents



n\_sent5<-review\_sentiment5[review\_sentiment5$value< 0,]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
wordcloud(n\_sent5$word,min.freq=25,colors=brewer.pal(8,"Dark2"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents  
  
## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in wordcloud(n\_sent5$word, min.freq = 25, colors = brewer.pal(8, : wrong  
## could not be fit on page. It will not be plotted.



n\_sent1<-review\_sentiment1[review\_sentiment1$value< 0,]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
wordcloud(n\_sent1$word,min.freq=100,colors=brewer.pal(6,"Paired"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## waste could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## wrong could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## embarrassing could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : worn  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## stressed could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## nasty could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## charged could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## alarm could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## complain could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : pay  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## cutting could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## forget could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : hurt  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## warning could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## excuse could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## beaten could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## stopped could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## screwed could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## empty could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## isolated could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## tired could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## defect could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## missing could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## terrible could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## harmful could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## mistake could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : hard  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## struggle could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## avoid could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## useless could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## sluggish could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## prevents could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## errors could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## falling could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## disappointed could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## worst could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## charges could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : fake  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## apologized could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## feeble could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## crazy could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : stop  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## refused could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## destroying could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## costly could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : sad  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## silly could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : suck  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## unacceptable could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## blocking could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## pressure could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : bad  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : risk  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## bother could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## regret could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## disappointment could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## obnoxious could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## discarded could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## awful could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## panic could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## insane could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## broken could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : leak  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## broke could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## worthless could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : dump  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## disaster could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## shocks could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : rant  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## avoided could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## annoys could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## awkward could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## preventing could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## suspected could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## failing could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## failure could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## warnings could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## unclear could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## uncomfortable could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## prevent could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## strike could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## terribly could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : bore  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : mad  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : dead  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## lowest could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## firing could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## shame could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## skeptical could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## tension could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : lost  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## misleading could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : torn  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## nonsense could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## demonstration could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## defeated could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## rigged could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## upset could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## failed could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## inconvenient could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## accidentally could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : scam  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : cuts  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## stuck could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## stops could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## misunderstanding could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## worry could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : bomb  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## disgrace could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, : anti  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(n\_sent1$word, min.freq = 100, colors = brewer.pal(6, :  
## shock could not be fit on page. It will not be plotted.



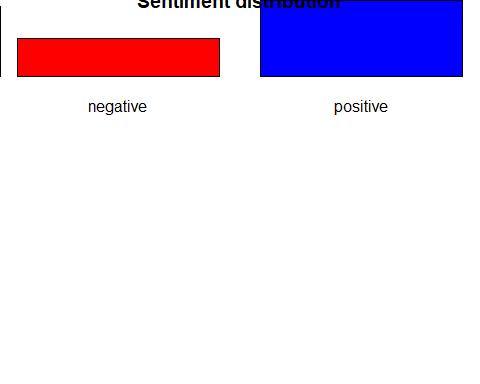
#BING Lexicon  
  
rtb <- data.table(review = c(revtext$words))  
review\_sentimentb <- rta %>%  
 unnest\_tokens(word, review, token = "words") %>%  
 inner\_join(get\_sentiments("bing"))

## Joining, by = "word"

str(review\_sentimentb)

## Classes 'data.table' and 'data.frame': 3265 obs. of 2 variables:  
## $ word : chr "works" "better" "realistic" "like" ...  
## $ sentiment: chr "positive" "positive" "positive" "positive" ...  
## - attr(\*, ".internal.selfref")=<externalptr>

countb<-table(review\_sentimentb$sentiment)  
  
barplot(countb, main="Sentiment distribution",  
 ylab="Number of sentiments", col = c("red","blue"))  
  
  
pos\_sent<-review\_sentimentb[review\_sentimentb$sentiment == 'positive',]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))



par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
  
wordcloud(pos\_sent$word,max.words = 100,colors=brewer.pal(8,"Dark2"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents

## Warning in wordcloud(pos\_sent$word, max.words = 100, colors = brewer.pal(8, :  
## good could not be fit on page. It will not be plotted.



neg\_sent<-review\_sentimentb[review\_sentimentb$sentiment == 'negative',]  
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
par(mar=rep(0, 4))  
plot.new()  
set.seed(100)  
  
  
wordcloud(neg\_sent$word,max.words = 100,colors=brewer.pal(6,"Paired"),scale=c(4,.5))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## problem could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## disappointed could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## expensive could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## broke could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## frustrating could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## complaints could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## smell could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## difficult could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## trouble could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## impossible could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## complaint could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## fault could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## slack could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## wrong could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## useless could not be fit on page. It will not be plotted.

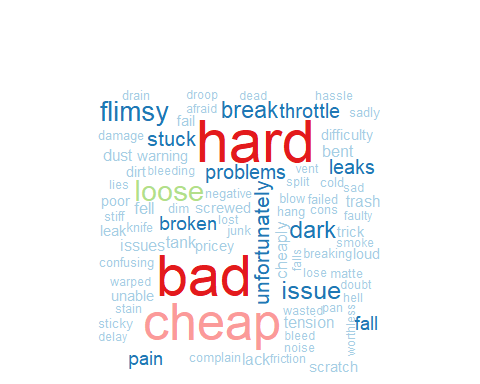
## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## bump could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## mess could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## hangs could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## scratches could not be fit on page. It will not be plotted.

## Warning in wordcloud(neg\_sent$word, max.words = 100, colors = brewer.pal(6, :  
## crack could not be fit on page. It will not be plotted.



#Syuzhet’s package  
  
  
algotext<-gsub("http[^[:blank:]]+","",alltext)  
algotext<-gsub("@\\w+","",algotext)  
algotext<-gsub("[[:punct:]]"," ",algotext)  
algotext<-gsub("[^[:alnum:]]"," ",algotext)  
algotext<-gsub("Miss."," ",algotext)  
algotext<-gsub("Mr."," ",algotext)  
  
algosent<-get\_nrc\_sentiment((algotext))  
  
  
algosent.positive =sum(algosent$positive)  
algosent.anger =sum(algosent$anger)  
algosent.anticipation =sum(algosent$anticipation)  
algosent.disgust =sum(algosent$disgust)  
algosent.fear =sum(algosent$fear)  
algosent.joy =sum(algosent$joy)  
algosent.sadness =sum(algosent$sadness)  
algosent.surprise =sum(algosent$surprise)  
algosent.trust =sum(algosent$trust)  
algosent.negative =sum(algosent$negative)  
  
colSums(algosent)

## anger anticipation disgust fear joy sadness   
## 592 603 470 692 461 618   
## surprise trust negative positive   
## 361 875 1629 1585

head(algosent)

## anger anticipation disgust fear joy sadness surprise trust negative positive  
## 1 121 131 82 129 84 125 71 193 330 343  
## 2 311 320 257 382 261 314 191 447 861 838  
## 3 160 152 131 181 116 179 99 235 438 404

yAxis <- c(algosent.positive,  
 + algosent.anger,  
 + algosent.anticipation,  
 + algosent.disgust,  
 + algosent.fear,  
 + algosent.joy,  
 + algosent.sadness,  
 + algosent.surprise,  
 + algosent.trust,  
 + algosent.negative)  
xAxis <- c("Negative","Anger","Anticipation","Disgust","Fear","Joy","Sadness","Surprise","Trust","Positive")  
yRange <- range(0,yAxis) + 500  
barplot(yAxis, names.arg = xAxis,  
 xlab = "Emotional valence", ylab = "Score", main = "Emotional Valence", col = brewer.pal(10,"RdYlBu"))

