Web & Social Media Assignment

Group 7 - BABI Bangalore - Section B

17 June 2018

For the assignment we have have identified brand as ‘World Environment Day’, which is celebrated on June 5 every year. This year UN ran a campaign on reduction of plastic waste, the campaign was advertised as #BeatPlasticPollution campaign on twitter.

In this assignment, we aim at understanding the sentiments of the people from around the world, on saving the planet and try understand the top trending topics when it comes to saving planet

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

#Step 1: Load Packages  
library(bit64)

## Loading required package: bit

## Attaching package bit

## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)

## creators: bit bitwhich

## coercion: as.logical as.integer as.bit as.bitwhich which

## operator: ! & | xor != ==

## querying: print length any all min max range sum summary

## bit access: length<- [ [<- [[ [[<-

## for more help type ?bit

##   
## Attaching package: 'bit'

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

## Attaching package bit64

## package:bit64 (c) 2011-2012 Jens Oehlschlaegel

## creators: integer64 seq :

## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bin

## logical operator: ! & | xor != == < <= >= >

## arithmetic operator: + - \* / %/% %% ^

## math: sign abs sqrt log log2 log10

## math: floor ceiling trunc round

## querying: is.integer64 is.vector [is.atomic} [length] format print str

## values: is.na is.nan is.finite is.infinite

## aggregation: any all min max range sum prod

## cumulation: diff cummin cummax cumsum cumprod

## access: length<- [ [<- [[ [[<-

## combine: c rep cbind rbind as.data.frame

## WARNING don't use as subscripts

## WARNING semantics differ from integer

## for more help type ?bit64

##   
## Attaching package: 'bit64'

## The following object is masked from 'package:bit':  
##   
## still.identical

## The following objects are masked from 'package:base':  
##   
## %in%, :, is.double, match, order, rank

library(twitteR)  
library(ROAuth)  
library(SnowballC)  
library(tm)

## Loading required package: NLP

library(ggplot2)

##   
## Attaching package: 'ggplot2'

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

library(RColorBrewer)  
library(wordcloud)  
library(topicmodels)  
library(data.table)

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

## The following object is masked from 'package:bit':  
##   
## setattr

library(stringi)  
library(syuzhet)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

## The following objects are masked from 'package:twitteR':  
##   
## id, location

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

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

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

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

library(grid)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

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

#Step 2: Set the working directory  
setwd("F:/BABI/Group Assignments/Web and Social media Analytics")  
getwd()

## [1] "F:/BABI/Group Assignments/Web and Social media Analytics"

#Step 3: Download Tweets from Twitter

#Step 4: Load the csv file  
tweets.df <- read.csv("tweetsWorldEnvironmentDay.csv")   
  
#Step 5: Convert date to correct date format  
tweets.df$created <- as.Date(tweets.df$created, format= "%d-%m-%Y")  
tweets.df$text <- as.character(tweets.df$text)  
str(tweets.df)

## 'data.frame': 1000 obs. of 17 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ text : chr "RT @RedTRaccoon: #WorldEnvironmentDay\n\nMy 7 year-old son asked me if it could get that bad and I had to expla"| \_\_truncated\_\_ "RT @Abhishe66526400: #WorldEnvironmentDay \n#<U+092A><U+0942><U+0930><U+094D><U+0923><U+092C><U+094D><U+0930><U"| \_\_truncated\_\_ "RT @SadhguruJV: This is a promise. You are wearing the mask to express that we will not make this little girl s"| \_\_truncated\_\_ "RT @MahindraRise: On World Earth Day we kickstarted a plastic collection drive across our facilities to #BeatPl"| \_\_truncated\_\_ ...  
## $ favorited : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ favoriteCount: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ replyToSN : Factor w/ 18 levels "aliaa08","DTurn12",..: NA NA NA NA NA NA NA NA NA NA ...  
## $ created : Date, format: "2018-05-06" "2018-05-06" ...  
## $ truncated : logi FALSE FALSE FALSE FALSE TRUE FALSE ...  
## $ replyToSID : num NA NA NA NA NA NA NA NA NA NA ...  
## $ id : num 1e+18 1e+18 1e+18 1e+18 1e+18 ...  
## $ replyToUID : num NA NA NA NA NA NA NA NA NA NA ...  
## $ statusSource : Factor w/ 29 levels "<a href=\"http://allafrica.com/\" rel=\"nofollow\">allAfrica.com</a>",..: 7 7 7 7 7 5 5 7 8 5 ...  
## $ screenName : Factor w/ 867 levels "\_HollyLake","\_karinaluiza",..: 253 739 51 698 188 225 78 860 633 290 ...  
## $ retweetCount : int 141 32 210 38 0 23 11 456 2 6 ...  
## $ isRetweet : logi TRUE TRUE TRUE TRUE FALSE TRUE ...  
## $ retweeted : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ longitude : logi NA NA NA NA NA NA ...  
## $ latitude : logi NA NA NA NA NA NA ...

#Step 6: Data Cleaning  
#Clean the text data by removing links, tags and delimiters.  
#Build a Corpus, and specify the location to be the character Vectors  
  
 #Creating document corpus with tweet text  
 myCorpus<- Corpus(VectorSource(tweets.df$text))   
  
 #Converting text to Lower Case  
 myCorpus <- tm\_map(myCorpus, content\_transformer(stri\_trans\_tolower))  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## rt @adgpi: 05 june 2018, to mark the #worldenvironmentday,  
## tree plantation drive was organised at the college of  
## defence management campus

#Removing the links (URLs)  
 removeURL <- function(x) gsub("http[^[:space:]]\*", "", x)   
 myCorpus <- tm\_map(myCorpus, content\_transformer(removeURL))  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## rt @adgpi: 05 june 2018, to mark the #worldenvironmentday,  
## tree plantation drive was organised at the college of  
## defence management campus

#Removing the Usernames  
 removeUsername <- function(x) gsub("@[^[:space:]]\*", "", x)   
 myCorpus <- tm\_map(myCorpus, content\_transformer(removeUsername))  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## rt 05 june 2018, to mark the #worldenvironmentday, tree  
## plantation drive was organised at the college of defence  
## management campus

#Removing anything except the english language and space  
 removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]\*", "", x)   
 myCorpus <- tm\_map(myCorpus, content\_transformer(removeNumPunct))  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## rt june to mark the worldenvironmentday tree plantation  
## drive was organised at the college of defence management  
## campus

#Removing anything except the english language and space  
 removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]\*", "", x)   
 myCorpus <- tm\_map(myCorpus, content\_transformer(removeNumPunct))  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## rt june to mark the worldenvironmentday tree plantation  
## drive was organised at the college of defence management  
## campus

#Removing Stopwords  
 myStopWords<- c((stopwords('english')),c("rt", "uu", "uuu", "uuuu", "us", "de", "dÃ", "worldenvironmentday", "environment", "uub", "uue", "amp", "la", "india", "dÃ", "day", "years", "year", "uauuuduucuduuudueuucuu","uufuf", "ufu", "uuc", "th", "pm", "en", "ueuu", "uuf", "ufuuaefufuub", "uauuuc", "uuuuu", "uuauu", "uau", "ucuuu", "s", "el", "fufuuaefufuub", "un", "uauudufueuuu", "fufucuf", "uudu", "uueuu"))  
 myCorpus<- tm\_map(myCorpus,removeWords , myStopWords)   
 writeLines(strwrap(myCorpus[[750]]$content,60))

## june mark tree plantation drive organised college defence  
## management campus

#Removing Single letter words  
 removeSingle <- function(x) gsub(" . ", " ", x)   
 myCorpus <- tm\_map(myCorpus, content\_transformer(removeSingle))  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## june mark tree plantation drive organised college defence  
## management campus

#Remove Extra Whitespaces  
 myCorpus<- tm\_map(myCorpus, stripWhitespace)  
 writeLines(strwrap(myCorpus[[750]]$content,60))

## june mark tree plantation drive organised college defence  
## management campus

#Step 7: Let us keep a copy of "myCorpus" for stem completion later  
myCorpusCopy<- myCorpus  
#myCorpus <- myCorpusCopy

#Stem words in the corpus  
#myCorpus<-tm\_map(myCorpus, stemDocument)  
#writeLines(strwrap(myCorpus[[250]]$content,60))  
  
#Function to correct/complete the text after stemming  
#stemCompletion2 <- function(x,dictionary) {  
# x <- unlist(strsplit(as.character(x)," "))  
# x <- x[x !=""]  
# x <- stemCompletion(x, dictionary = dictionary)  
# x <- paste(x, sep="", collapse=" ")  
# PlainTextDocument(stripWhitespace(x))#}  
  
#Stem Complete and Display the same tweet above with the completed and corrected text.  
#myCorpus <- lapply(myCorpus, stemCompletion2, dictionary=myCorpusCopy)  
#myCorpus <- Corpus(VectorSource(myCorpus))  
#writeLines(strwrap(myCorpus[[250]]$content, 60))  
  
#Correcting mis-splet words  
#wordFreq <- function(corpus,word)  
#{results<- lapply(corpus,function(x){ grep(as.character(x),pattern = paste0("\\<", #word))})sum(unlist(results))}  
  
#n.plastic <- wordFreq(myCorpus, "plastic")  
#n.plastics <- wordFreq(myCorpus, "plastics")  
#cat(n.plstic, n.plstics)  
  
#Replace words with the proper ones  
replaceWord <- function(corpus, plastics, plastic)  
{  
tm\_map(corpus, content\_transformer(gsub), pattern=plastics, replacement=plastic)  
 }  
myCorpus<- replaceWord(myCorpus, "plastics", "plastic")

#Step 8: Creating a term document matrix  
#myCorpus <- Corpus(VectorSource(myCorpus))  
tdm<- TermDocumentMatrix(myCorpus, control= list(wordLengths= c(1, Inf)))  
tdm

## <<TermDocumentMatrix (terms: 3561, documents: 1000)>>  
## Non-/sparse entries: 8962/3552038  
## Sparsity : 100%  
## Maximal term length: 111  
## Weighting : term frequency (tf)

#Step 9: Find the terms used most frequently  
(freq.terms <- findFreqTerms(tdm, lowfreq = 25))

## [1] "future" "make" "will"   
## [4] "beatplasticpollution" "earth" "plastic"   
## [7] "world" "planet" "every"   
## [10] "oceans" "singleuse" "need"   
## [13] "can" "help" "save"   
## [16] "lets" "pollution" "one"   
## [19] "happy" "waste" "nature"   
## [22] "protect" "use" "together"   
## [25] "time" "today" "think"   
## [28] "dÃ" "let" "live"

term.freq <- rowSums(as.matrix(tdm))  
term.freq <- subset(term.freq, term.freq > 25)  
df <- data.frame(term = names(term.freq), freq= term.freq)

#Step 10: Frequency Analysis  
(freq.terms <- findFreqTerms(tdm, lowfreq = 10))

## [1] "bad" "get" "u"   
## [4] "future" "make" "will"   
## [7] "beatplasticpollution" "earth" "plastic"   
## [10] "world" "people" "care"   
## [13] "children" "planet" "every"   
## [16] "oceans" "removing" "seas"   
## [19] "trash" "air" "water"   
## [22] "bottles" "cant" "refuse"   
## [25] "reuse" "singleuse" "environmental"   
## [28] "bags" "change" "need"   
## [31] "using" "challenge" "great"   
## [34] "stop" "create" "dont"   
## [37] "just" "love" "take"   
## [40] "taking" "can" "help"   
## [43] "save" "want" "lets"   
## [46] "never" "remember" "beat"   
## [49] "pollution" "pledge" "tree"   
## [52] "million" "one" "happy"   
## [55] "first" "food" "greener"   
## [58] "sustainable" "waste" "tag"   
## [61] "win" "machine" "animals"   
## [64] "made" "important" "less"   
## [67] "nature" "protect" "use"   
## [70] "way" "celebrate" "clean"   
## [73] "together" "impact" "time"   
## [76] "youre" "look" "new"   
## [79] "today" "may" "responsibility"   
## [82] "say" "well" "must"   
## [85] "part" "solution" "beautiful"   
## [88] "green" "life" "towards"   
## [91] "trees" "ocean" "think"   
## [94] "proud" "dÃ" "generations"   
## [97] "let" "live" "planting"   
## [100] "now" "something" "wed"   
## [103] "cover" "healthy" "home"   
## [106] "eating" "elephants" "everyone"   
## [109] "billion" "old" "go"   
## [112] "beatplastic" "breaking" "commits"   
## [115] "eliminate" "awareness" "spread"   
## [118] "reduce" "heres" "global"   
## [121] "work" "join" "free"   
## [124] "find" "improve" "know"   
## [127] "give" "set" "climate"

term.freq <- rowSums(as.matrix(tdm))  
term.freq <- subset(term.freq, term.freq > 10)  
df1 <- data.frame(term = names(term.freq), freq= term.freq)  
  
(freq.terms <- findFreqTerms(tdm, lowfreq = 40))

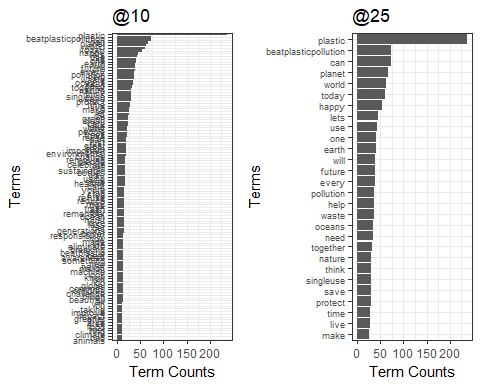
## [1] "beatplasticpollution" "earth" "plastic"   
## [4] "world" "planet" "can"   
## [7] "lets" "one" "happy"   
## [10] "use" "today"

term.freq <- rowSums(as.matrix(tdm))  
term.freq <- subset(term.freq, term.freq > 40)  
df2 <- data.frame(term = names(term.freq), freq= term.freq)  
  
(freq.terms <- findFreqTerms(tdm, lowfreq = 65))

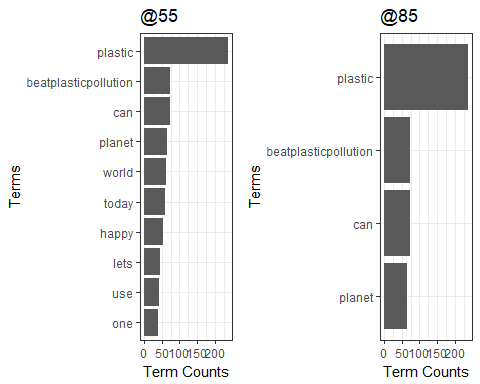
## [1] "beatplasticpollution" "plastic" "planet"   
## [4] "can"

term.freq <- rowSums(as.matrix(tdm))  
term.freq <- subset(term.freq, term.freq > 65)  
df3 <- data.frame(term = names(term.freq), freq= term.freq)

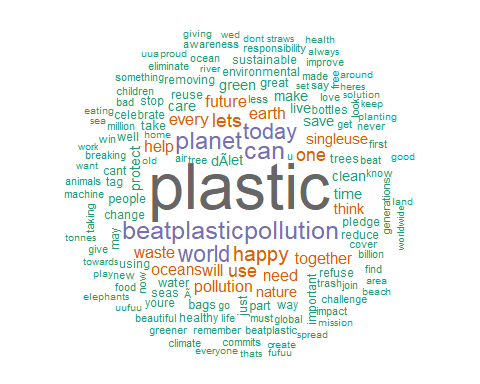
#Step 11: Plotting the graph of frequent terms  
p1=ggplot(df1, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@10", x="Terms", y="Term Counts")) + theme(axis.text.y = element\_text(size=7))  
  
  
p2=ggplot(df, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@25", x="Terms", y="Term Counts"))+  
 theme(axis.text.y = element\_text(size=7))  
  
  
p3=ggplot(df2, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@55", x="Terms", y="Term Counts"))  
  
p4=ggplot(df3, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@85", x="Terms", y="Term Counts"))   
  
grid.arrange(p1,p2,ncol=2)



grid.arrange(p3,p4,ncol=2)



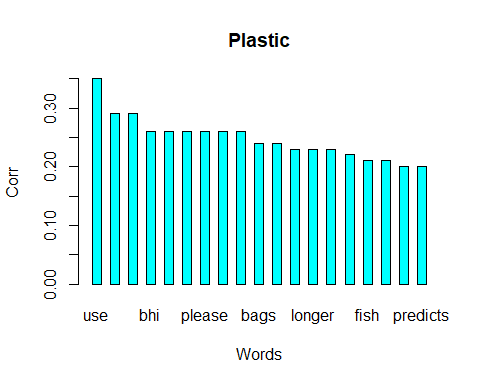
#Step 12: Calculate the frequency of words and sort it by frequency and setting up the Wordcloud  
word.freq <-sort(rowSums(as.matrix(tdm)), decreasing= F)  
pal<- brewer.pal(8, "Dark2")  
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 2, random.order = F, colors = pal, max.words = 150)



#Step 13: Find association with a specific keyword in the tweets - Plastic, Pollution, Planet, Oceans  
list1<- findAssocs(tdm, "plastic", 0.2)  
corrdf1 <- t(data.frame(t(sapply(list1,c))))  
corrdf1

## plastic  
## use 0.35  
## singleuse 0.29  
## pollution 0.29  
## bhi 0.26  
## glasses 0.26  
## kulhad 0.26  
## please 0.26  
## restaurant 0.26  
## tas 0.26  
## bags 0.24  
## beat 0.24  
## refuse 0.23  
## longer 0.23  
## next. 0.23  
## straws 0.22  
## fish 0.21  
## give 0.21  
## waste 0.20  
## predicts 0.20

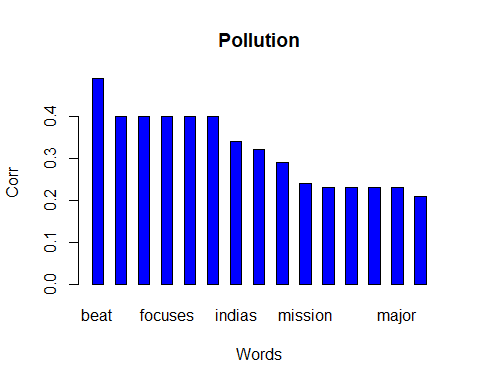
barplot(t(as.matrix(corrdf1)), beside=TRUE,xlab = "Words",ylab = "Corr",col = "Cyan",main = "Plastic",border = "black")



list1<- findAssocs(tdm, "pollution", 0.2)  
corrdf1 <- t(data.frame(t(sapply(list1,c))))  
corrdf1

## pollution  
## beat 0.49  
## beating 0.40  
## bharat 0.40  
## focuses 0.40  
## rightly 0.40  
## swachh 0.40  
## indias 0.34  
## specia 0.32  
## plastic 0.29  
## mission 0.24  
## entirely 0.23  
## crisis 0.23  
## threatening 0.23  
## major 0.23  
## solve 0.21

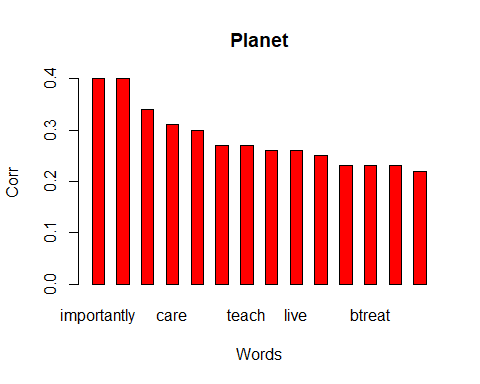
barplot(t(as.matrix(corrdf1)), beside=TRUE,xlab = "Words",ylab = "Corr",col = "blue",main = "Pollution",border = "black")



list1<- findAssocs(tdm, "planet", 0.2)  
corrdf1 <- t(data.frame(t(sapply(list1,c))))  
corrdf1

## planet  
## importantly 0.40  
## respect 0.40  
## belong 0.34  
## care 0.31  
## ensure 0.30  
## storytelling 0.27  
## teach 0.27  
## greetings 0.26  
## live 0.26  
## itfufuu 0.25  
## generations 0.23  
## btreat 0.23  
## utmost 0.23  
## children 0.22

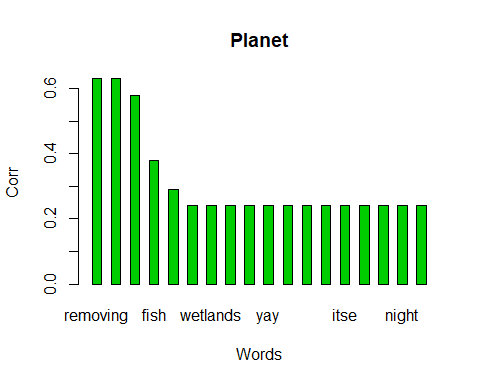
barplot(t(as.matrix(corrdf1)), beside=TRUE,xlab = "Words",ylab = "Corr",col = "red",main = "Planet",border = "black")



list1<- findAssocs(tdm, "oceans", 0.2)  
corrdf1 <- t(data.frame(t(sapply(list1,c))))  
corrdf1

## oceans  
## removing 0.63  
## trash 0.63  
## seas 0.58  
## fish 0.38  
## predicts 0.29  
## hats 0.24  
## wetlands 0.24  
## outweigh 0.24  
## urg 0.24  
## yay 0.24  
## filling 0.24  
## leak 0.24  
## emerging 0.24  
## itse 0.24  
## keeps 0.24  
## luckily 0.24  
## night 0.24  
## pieces 0.24

barplot(t(as.matrix(corrdf1)), beside=TRUE,xlab = "Words",ylab = "Corr",col = "green3",main = "Planet",border = "black")

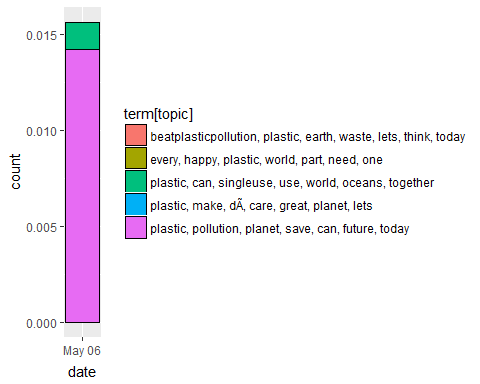


#Step 14: Topic Modelling to identify latent/hidden topics using LDA technique  
dtm <- as.DocumentTermMatrix(tdm)  
  
rowTotals <- apply(dtm , 1, sum)  
  
NullDocs <- dtm[rowTotals==0, ]  
dtm <- dtm[rowTotals> 0, ]  
  
if (length(NullDocs$dimnames$Docs) > 0) {  
tweets.df <- tweets.df[-as.numeric(NullDocs$dimnames$Docs),]  
}  
  
lda <- LDA(dtm, k = 5) # find 5 topic  
term <- terms(lda, 7) # first 7 terms of every topic  
(term <- apply(term, MARGIN = 2, paste, collapse = ", "))

## Topic 1   
## "plastic, make, dÃ, care, great, planet, lets"   
## Topic 2   
## "beatplasticpollution, plastic, earth, waste, lets, think, today"   
## Topic 3   
## "every, happy, plastic, world, part, need, one"   
## Topic 4   
## "plastic, can, singleuse, use, world, oceans, together"   
## Topic 5   
## "plastic, pollution, planet, save, can, future, today"

topics<- topics(lda)  
topics<- data.frame(date=(tweets.df$created), topic = topics)  
qplot (date, ..count.., data=topics, geom ="density", fill= term[topic], position="stack")

## Warning: `position` is deprecated



#Step 15: Conducting Sentiment Analysis  
mysentiment<-get\_nrc\_sentiment((tweets.df$text))  
  
# Get the sentiment score for each emotion  
mysentiment.positive =sum(mysentiment$positive)  
mysentiment.anger =sum(mysentiment$anger)  
mysentiment.anticipation =sum(mysentiment$anticipation)  
mysentiment.disgust =sum(mysentiment$disgust)  
mysentiment.fear =sum(mysentiment$fear)  
mysentiment.joy =sum(mysentiment$joy)  
mysentiment.sadness =sum(mysentiment$sadness)  
mysentiment.surprise =sum(mysentiment$surprise)  
mysentiment.trust =sum(mysentiment$trust)  
mysentiment.negative =sum(mysentiment$negative)  
  
# Create the bar chart  
yAxis <- c(mysentiment.positive,  
 + mysentiment.anger,  
 + mysentiment.anticipation,  
 + mysentiment.disgust,  
 + mysentiment.fear,  
 + mysentiment.joy,  
 + mysentiment.sadness,  
 + mysentiment.surprise,  
 + mysentiment.trust,  
 + mysentiment.negative)  
  
xAxis <- c("Positive","Anger","Anticipation","Disgust","Fear","Joy","Sadness",  
 "Surprise","Trust","Negative")  
colors <- c("green","red","blue","orange","red","green","orange","blue","green","red")  
yRange <- range(0,yAxis)  
barplot(yAxis, names.arg = xAxis,   
 xlab = "Emotional valence", ylab = "Score", main = "Twitter sentiment",   
 sub = "World Environment Day", col = colors, border = "black", xpd = F, ylim = yRange,  
 axisnames = T, cex.axis = 0.8, cex.sub = 0.8, col.sub = "blue")

