```
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/00228/smsspamcollection.zip"

if (!file.exists("smsspamcollection.zip"))
  {
  download.file(url=url, destfile="smsspamcollection.zip", method="curl")
  }
unzip("smsspamcollection.zip")

data_text <- read.delim("SMSSpamCollection", sep="\t", header=F, colClasses="character", quote="")
```

```
str(data_text)
```

```
## 'data.frame':    5574 obs. of  2 variables:
##  $ V1: chr  "ham" "ham" "spam" "ham" ...
##  $ V2: chr  "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got a
more wat..." "Ok lar... Joking wif u oni..." "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Te
xt FA to 87121 to receive entry question("| __truncated__ "U dun say so early hor... U c already then say..." ...
```

```
head(data_text)
```

```
##     V1
## 1  ham
## 2  ham
## 3 spam
## 4  ham
## 5  ham
## 6 spam
##
V2
## 1                                                 Go until jurong point, crazy.. Available only in bugis n great worl
d la e buffet... Cine there got amore wat...
## 2
Ok lar... Joking wif u oni...
## 3 Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question
(std txt rate)T&C's apply 08452810075over18's
## 4                                                                                                          U dun
say so early hor... U c already then say...
## 5                                                                                                Nah I don't think
he goes to usf, he lives around here though
## 6        FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it stil
l? Tb ok! XxX std chgs to send, Â£1.50 to rcv
```

```
colnames(data_text) <- c("Class","Text")
colnames(data_text)
```

```
## [1] "Class" "Text"
```

```
data_text$Class <- factor(data_text$Class)
prop.table(table(data_text$Class))
```

```
##
##       ham      spam
## 0.8659849 0.1340151
```

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.1.2
```

```
## Loading required package: NLP
```

```
library(SnowballC)
```

```
corpus = VCorpus(VectorSource(data_text$Text))
as.character(corpus[[1]])
```

```
## [1] "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wa
t..."
```

```
corpus = tm_map(corpus, content_transformer(tolower))
corpus = tm_map(corpus, removeNumbers)
corpus = tm_map(corpus, removePunctuation)
corpus = tm_map(corpus, removeWords, stopwords("english"))
corpus = tm_map(corpus, stemDocument)
corpus = tm_map(corpus, stripWhitespace)
as.character(corpus[[1]])
```

```
## [1] "go jurong point crazi avail bugi n great world la e buffet cine got amor wat"
```

```
dtm = DocumentTermMatrix(corpus)
```

```
dtm
```

```
## <<DocumentTermMatrix (documents: 5574, terms: 6981)>>
## Non-/sparse entries: 43801/38868293
## Sparsity           : 100%
## Maximal term length: 40
## Weighting          : term frequency (tf)
```

```
dtm = removeSparseTerms(dtm, 0.999)

dim(dtm)
```

```
## [1] 5574 1209
```

```
inspect(dtm[40:50, 10:15])
```

```
## <<DocumentTermMatrix (documents: 11, terms: 6)>>
## Non-/sparse entries: 0/66
## Sparsity           : 100%
## Maximal term length: 7
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs activ actual add address admir adult
##   40     0      0   0       0     0     0
##   41     0      0   0       0     0     0
##   42     0      0   0       0     0     0
##   43     0      0   0       0     0     0
##   44     0      0   0       0     0     0
##   45     0      0   0       0     0     0
##   46     0      0   0       0     0     0
##   47     0      0   0       0     0     0
##   48     0      0   0       0     0     0
##   49     0      0   0       0     0     0
##   50     0      0   0       0     0     0
```

```
convert_count <- function(x) {
  y <- ifelse(x > 0, 1,0)
  y <- factor(y, levels=c(0,1), labels=c("No", "Yes"))
  y
}

# Apply the convert_count function to get final training and testing DTMs
datasetNB <- apply(dtm, 2, convert_count)

dataset = as.data.frame(as.matrix(datasetNB))
```

```
freq<- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
tail(freq, 10)
```

```
##   vikki vodafon    vote     vri  wherev     wnt     wwq     yay    yiju     zed
##       6       6       6       6       6       6       6       6       6       6
```

```
findFreqTerms(dtm, lowfreq=60)
```

```
##   [1] "alreadi"  "also"     "amp"      "anyth"    "around"   "ask"
##   [7] "award"    "babe"     "back"     "buy"      "call"     "can"
##  [13] "cant"     "care"     "cash"     "chat"     "claim"    "come"
##  [19] "contact"  "cos"      "custom"   "day"      "dear"     "didnt"
##  [25] "dont"     "end"      "even"     "everi"    "feel"     "find"
##  [31] "finish"   "first"    "free"     "friend"   "get"      "give"
##  [37] "good"     "got"      "great"    "gud"      "guy"      "happi"
##  [43] "help"     "hey"      "home"     "hope"     "ill"      "ive"
##  [49] "just"     "keep"     "know"     "last"     "later"    "leav"
##  [55] "let"      "life"     "like"     "lol"      "look"     "lor"
##  [61] "love"     "ltgt"     "make"     "meet"     "messag"   "min"
##  [67] "miss"     "mobil"    "morn"     "msg"      "much"     "need"
##  [73] "new"      "next"     "night"    "nokia"    "now"      "number"
##  [79] "one"      "person"   "phone"    "pick"     "place"    "pleas"
##  [85] "pls"      "prize"    "realli"   "repli"    "right"    "said"
##  [91] "say"      "see"      "send"     "sent"     "servic"   "show"
##  [97] "sleep"    "smile"    "someon"   "someth"   "sorri"    "start"
## [103] "still"    "stop"     "sure"     "take"     "talk"     "tell"
## [109] "text"     "thank"    "that"     "thing"    "think"    "time"
## [115] "today"    "tomorrow" "tone"     "tonight"  "tri"      "txt"
## [121] "urgent"   "use"      "wait"     "want"     "wat"      "watch"
## [127] "way"      "week"     "well"     "went"     "will"     "win"
## [133] "wish"     "won"      "work"     "yeah"     "year"     "yes"
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

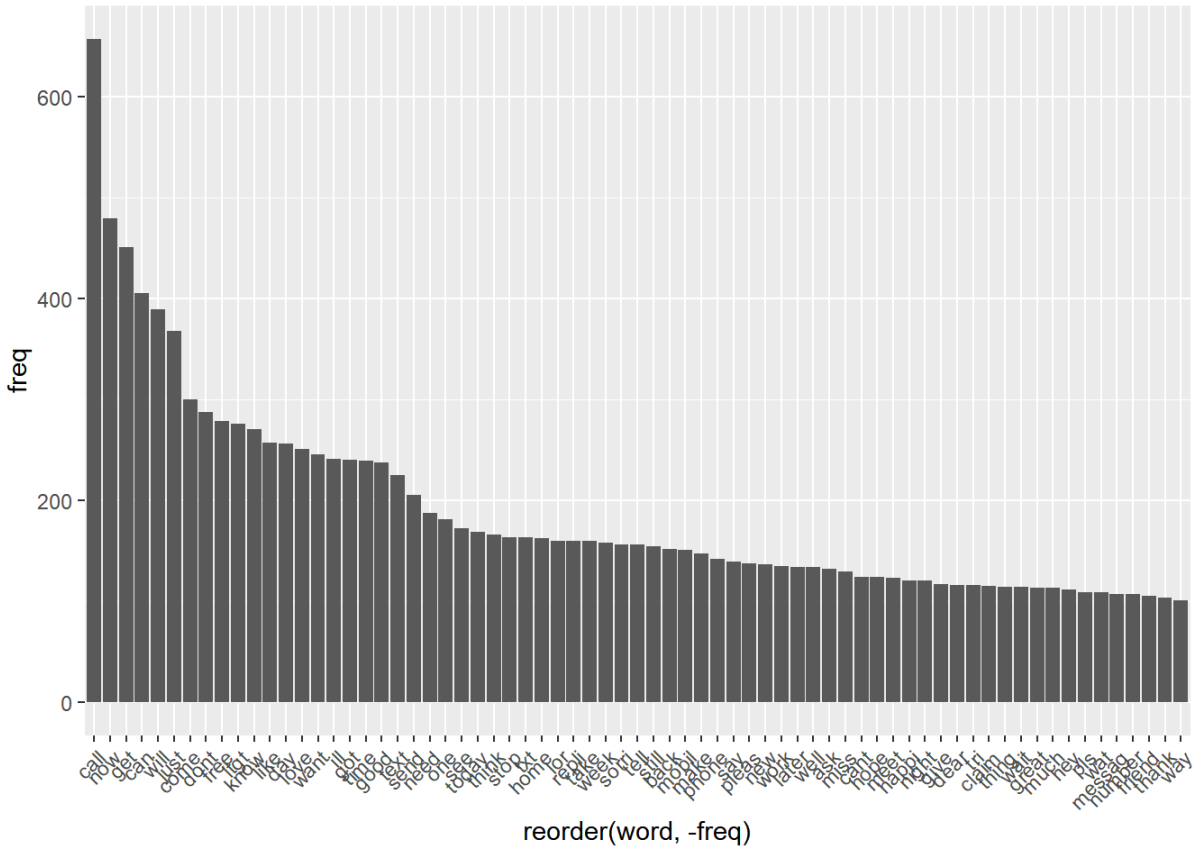
```
##
## Attaching package: 'ggplot2'
```

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

```
wf<- data.frame(word=names(freq), freq=freq)
head(wf)
```

```
##      word freq
## call call  657
## now   now  479
## get   get  451
## can   can  405
## will will  389
## just just  368
```

```
pp <- ggplot(subset(wf, freq>100), aes(x=reorder(word, -freq), y =freq)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x=element_text(angle=45, hjust=1))
pp
```



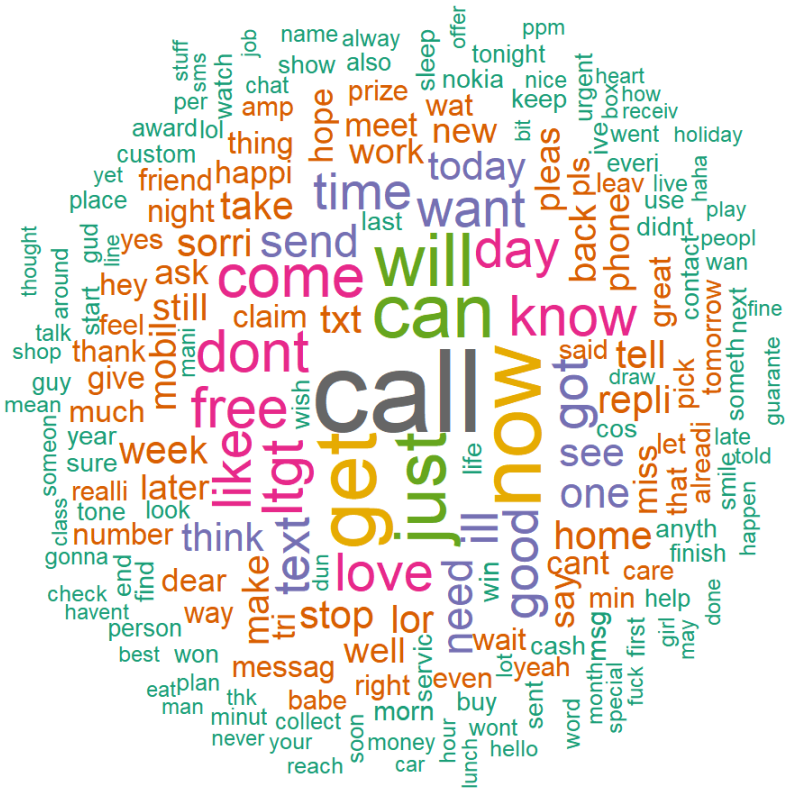

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.1.2
```

```
## Loading required package: RColorBrewer
```

```
library(RColorBrewer)
```

```
set.seed(1234)
wordcloud(words = wf$word, freq = wf$freq, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```




```
dataset$Class = data_text$Class
str(dataset$Class)
```

```
##  Factor w/ 2 levels "ham","spam": 1 1 2 1 1 2 1 1 2 2 ...
```

```
set.seed(222)
split = sample(2,nrow(dataset),prob = c(0.75,0.25),replace = TRUE)
train_set = dataset[split == 1,]
test_set = dataset[split == 2,]

prop.table(table(train_set$Class))
```

```
##
##       ham      spam
## 0.8670327 0.1329673
```

```
prop.table(table(test_set$Class))
```

```
##
##       ham      spam
## 0.8628159 0.1371841
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

```
## Loading required package: lattice
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.2
```

```
control <- trainControl(method="repeatedcv", number=10, repeats=3)
system.time( classifier_nb <- naiveBayes(train_set, train_set$Class, laplace = 1,
                                         trControl = control,tuneLength = 7) )
```

```
##    user  system elapsed
##    0.78    0.08    0.86
```

```
nb_pred = predict(classifier_nb, type = 'class', newdata = test_set)

confusionMatrix(nb_pred,test_set$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  ham spam
##       ham  1193    1
##       spam    2  189
##
##                Accuracy : 0.9978
##                  95% CI : (0.9937, 0.9996)
##     No Information Rate : 0.8628
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9909
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9983
##             Specificity : 0.9947
##          Pos Pred Value : 0.9992
##          Neg Pred Value : 0.9895
##              Prevalence : 0.8628
##          Detection Rate : 0.8614
##    Detection Prevalence : 0.8621
##       Balanced Accuracy : 0.9965
##
##        'Positive' Class : ham
##
```