Project 4: Document Classification

Heleine Fouda

2023-11-19

The goal of this project is to classify email documents as spam(unsolicited) or ham (solicited) based on classified training documents made available by spamassassin.

Getting Started:

Loading corpus

```
# Files directories
ham_directory <-"/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/easy_ham"
spam_directory <-"/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/spam_2"
```

```
# create vectora of document names
spam_files <- list.files(spam_directory)
ham_files <- list.files(ham_directory)
```

Let's take a peek at the files

```
glimpse(spam_files)
```

```
##  chr [1:1397] "00001.317e78fa8ee2f54cd4890fdc09ba8176" ...
```

```
length(spam_files)
```

```
## [1] 1397
```

```
glimpse(ham_files)
```

```
##  chr [1:2501] "00001.7c53336b37003a9286aba55d2945844c" ...
```

```
length(ham_files)
```

```
## [1] 2501
```

Email Content Extraction

1. Spam

```
# Define the spam directory
spam_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/spam_2"

# List of file names in the spam directory
spam_file_names <- list.files(spam_directory, full.names = FALSE)

# Choose one file to extract content
selected_file <- spam_file_names[1]  # Change the index as needed
```

```r
# Construct the full path to the file
file_path <- file.path(spam_directory, selected_file)

# Read the content of the file
content_spam <- readLines(file_path)

# Print or process the content as needed
cat("Content of", selected_file, ":\n")
```

```
## Content of 00001.317e78fa8ee2f54cd4890fdc09ba8176 :
```

```r
cat(content_spam , sep = "\n")
```

```
## From ilug-admin@linux.ie  Tue Aug  6 11:51:02 2002
## Return-Path: <ilug-admin@linux.ie>
## Delivered-To: yyyy@localhost.netnoteinc.com
## Received: from localhost (localhost [127.0.0.1])
##  by phobos.labs.netnoteinc.com (Postfix) with ESMTP id 9E1F5441DD
##  for <jm@localhost>; Tue,  6 Aug 2002 06:48:09 -0400 (EDT)
## Received: from phobos [127.0.0.1]
##  by localhost with IMAP (fetchmail-5.9.0)
##  for jm@localhost (single-drop); Tue, 06 Aug 2002 11:48:09 +0100 (IST)
## Received: from lugh.tuatha.org (root@lugh.tuatha.org [194.125.145.45]) by
##     dogma.slashnull.org (8.11.6/8.11.6) with ESMTP id g72LqWv13294 for
##     <jm-ilug@jmason.org>; Fri, 2 Aug 2002 22:52:32 +0100
## Received: from lugh (root@localhost [127.0.0.1]) by lugh.tuatha.org
##     (8.9.3/8.9.3) with ESMTP id WAA31224; Fri, 2 Aug 2002 22:50:17 +0100
## Received: from bettyjagessar.com (w142.z064000057.nyc-ny.dsl.cnc.net
##     [64.0.57.142]) by lugh.tuatha.org (8.9.3/8.9.3) with ESMTP id WAA31201 for
##     <ilug@linux.ie>; Fri, 2 Aug 2002 22:50:11 +0100
## X-Authentication-Warning: lugh.tuatha.org: Host w142.z064000057.nyc-ny.dsl.cnc.net
##     [64.0.57.142] claimed to be bettyjagessar.com
## Received: from 64.0.57.142 [202.63.165.34] by bettyjagessar.com
##     (SMTPD32-7.06 EVAL) id A42A7FC01F2; Fri, 02 Aug 2002 02:18:18 -0400
## Message-Id: <1028311679.886@0.57.142>
## Date: Fri, 02 Aug 2002 23:37:59 0530
## To: ilug@linux.ie
## From: "Start Now" <startnow2002@hotmail.com>
## MIME-Version: 1.0
## Content-Type: text/plain; charset="US-ASCII"; format=flowed
## Subject: [ILUG] STOP THE MLM INSANITY
## Sender: ilug-admin@linux.ie
## Errors-To: ilug-admin@linux.ie
## X-Mailman-Version: 1.1
## Precedence: bulk
## List-Id: Irish Linux Users' Group <ilug.linux.ie>
## X-Beenthere: ilug@linux.ie
##
## Greetings!
##
## You are receiving this letter because you have expressed an interest in
## receiving information about online business opportunities. If this is
## erroneous then please accept my most sincere apology. This is a one-time
## mailing, so no removal is necessary.
```

```
##
## If you've been burned, betrayed, and back-stabbed by multi-level marketing,
## MLM, then please read this letter. It could be the most important one that
## has ever landed in your Inbox.
##
## MULTI-LEVEL MARKETING IS A HUGE MISTAKE FOR MOST PEOPLE
##
## MLM has failed to deliver on its promises for the past 50 years. The pursuit
## of the "MLM Dream" has cost hundreds of thousands of people their friends,
## their fortunes and their sacred honor. The fact is that MLM is fatally
## flawed, meaning that it CANNOT work for most people.
##
## The companies and the few who earn the big money in MLM are NOT going to
## tell you the real story. FINALLY, there is someone who has the courage to
## cut through the hype and lies and tell the TRUTH about MLM.
##
## HERE'S GOOD NEWS
##
## There IS an alternative to MLM that WORKS, and works BIG! If you haven't yet
## abandoned your dreams, then you need to see this. Earning the kind of income
## you've dreamed about is easier than you think!
##
## With your permission, I'd like to send you a brief letter that will tell you
## WHY MLM doesn't work for most people and will then introduce you to
## something so new and refreshing that you'll wonder why you haven't heard of
## this before.
##
## I promise that there will be NO unwanted follow up, NO sales pitch, no one
## will call you, and your email address will only be used to send you the
## information. Period.
##
## To receive this free, life-changing information, simply click Reply, type
## "Send Info" in the Subject box and hit Send. I'll get the information to you
## within 24 hours. Just look for the words MLM WALL OF SHAME in your Inbox.
##
## Cordially,
##
## Siddhi
##
## P.S. Someone recently sent the letter to me and it has been the most
## eye-opening, financially beneficial information I have ever received. I
## honestly believe that you will feel the same way once you've read it. And
## it's FREE!
##
##
## ------------------------------------------------------------
## This email is NEVER sent unsolicited.  THIS IS NOT "SPAM". You are receiving
## this email because you EXPLICITLY signed yourself up to our list with our
## online signup form or through use of our FFA Links Page and E-MailDOM
## systems, which have EXPLICIT terms of use which state that through its use
## you agree to receive our emailings.  You may also be a member of a Altra
## Computer Systems list or one of many numerous FREE Marketing Services and as
## such you agreed when you signed up for such list that you would also be
## receiving this emailing.
## Due to the above, this email message cannot be considered unsolicitated, or
## spam.
## -----------------------------------------------------------
##
##
##
##
## --
## Irish Linux Users' Group: ilug@linux.ie
## http://www.linux.ie/mailman/listinfo/ilug for (un)subscription information.
## List maintainer: listmaster@linux.ie
```

```
cat("\n")
```

2. ham

```
# Define the ham directory
ham_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/easy_ham"

# List of file names in the ham directory
ham_file_names <- list.files(ham_directory, full.names = FALSE)

# Choose one file to extract content
selected_file <- ham_file_names[1]  # Change the index as needed

# Construct the full path to the file
file_path <- file.path(ham_directory, selected_file)

# Read the content of the file
content_ham <- readLines(file_path)

# Print or process the content as needed
cat("Content of", selected_file, ":\n")
```

```
## Content of 00001.7c53336b37003a9286aba55d2945844c :
```

```
cat(content_ham , sep = "\n")
```

```
## From exmh-workers-admin@redhat.com  Thu Aug 22 12:36:23 2002
## Return-Path: <exmh-workers-admin@spamassassin.taint.org>
## Delivered-To: zzzz@localhost.netnoteinc.com
## Received: from localhost (localhost [127.0.0.1])
##  by phobos.labs.netnoteinc.com (Postfix) with ESMTP id D03E543C36
##  for <zzzz@localhost>; Thu, 22 Aug 2002 07:36:16 -0400 (EDT)
## Received: from phobos [127.0.0.1]
##  by localhost with IMAP (fetchmail-5.9.0)
##  for zzzz@localhost (single-drop); Thu, 22 Aug 2002 12:36:16 +0100 (IST)
## Received: from listman.spamassassin.taint.org (listman.spamassassin.taint.org [66.187.233.211]) by
##     dogma.slashnull.org (8.11.6/8.11.6) with ESMTP id g7MBYrZ04811 for
##     <zzzz-exmh@spamassassin.taint.org>; Thu, 22 Aug 2002 12:34:53 +0100
## Received: from listman.spamassassin.taint.org (localhost.localdomain [127.0.0.1]) by
##     listman.redhat.com (Postfix) with ESMTP id 8386540858; Thu, 22 Aug 2002
##     07:35:02 -0400 (EDT)
## Delivered-To: exmh-workers@listman.spamassassin.taint.org
## Received: from int-mx1.corp.spamassassin.taint.org (int-mx1.corp.spamassassin.taint.org
##     [172.16.52.254]) by listman.redhat.com (Postfix) with ESMTP id 10CF8406D7
##     for <exmh-workers@listman.redhat.com>; Thu, 22 Aug 2002 07:34:10 -0400
##     (EDT)
## Received: (from mail@localhost) by int-mx1.corp.spamassassin.taint.org (8.11.6/8.11.6)
##     id g7MBY7g11259 for exmh-workers@listman.redhat.com; Thu, 22 Aug 2002
##     07:34:07 -0400
## Received: from mx1.spamassassin.taint.org (mx1.spamassassin.taint.org [172.16.48.31]) by
##     int-mx1.corp.redhat.com (8.11.6/8.11.6) with SMTP id g7MBY7Y11255 for
##     <exmh-workers@redhat.com>; Thu, 22 Aug 2002 07:34:07 -0400
## Received: from ratree.psu.ac.th ([202.28.97.6]) by mx1.spamassassin.taint.org
##     (8.11.6/8.11.6) with SMTP id g7MBIhl25223 for <exmh-workers@redhat.com>;
##     Thu, 22 Aug 2002 07:18:55 -0400
## Received: from delta.cs.mu.OZ.AU (delta.coe.psu.ac.th [172.30.0.98]) by
##     ratree.psu.ac.th (8.11.6/8.11.6) with ESMTP id g7MBWel29762;
##     Thu, 22 Aug 2002 18:32:40 +0700 (ICT)
## Received: from munnari.OZ.AU (localhost [127.0.0.1]) by delta.cs.mu.OZ.AU
##     (8.11.6/8.11.6) with ESMTP id g7MBQPW13260; Thu, 22 Aug 2002 18:26:25
##     +0700 (ICT)
## From: Robert Elz <kre@munnari.OZ.AU>
## To: Chris Garrigues <cwg-dated-1030377287.06fa6d@DeepEddy.Com>
## Cc: exmh-workers@spamassassin.taint.org
## Subject: Re: New Sequences Window
## In-Reply-To: <1029945287.4797.TMDA@deepeddy.vircio.com>
## References: <1029945287.4797.TMDA@deepeddy.vircio.com>
##     <1029882468.3116.TMDA@deepeddy.vircio.com> <9627.1029933001@munnari.OZ.AU>
##     <1029943066.26919.TMDA@deepeddy.vircio.com>
##     <1029944441.398.TMDA@deepeddy.vircio.com>
## MIME-Version: 1.0
## Content-Type: text/plain; charset=us-ascii
## Message-Id: <13258.1030015585@munnari.OZ.AU>
## X-Loop: exmh-workers@spamassassin.taint.org
## Sender: exmh-workers-admin@spamassassin.taint.org
## Errors-To: exmh-workers-admin@spamassassin.taint.org
## X-Beenthere: exmh-workers@spamassassin.taint.org
## X-Mailman-Version: 2.0.1
## Precedence: bulk
## List-Help: <mailto:exmh-workers-request@spamassassin.taint.org?subject=help>
## List-Post: <mailto:exmh-workers@spamassassin.taint.org>
## List-Subscribe: <https://listman.spamassassin.taint.org/mailman/listinfo/exmh-workers>,
##     <mailto:exmh-workers-request@redhat.com?subject=subscribe>
## List-Id: Discussion list for EXMH developers <exmh-workers.spamassassin.taint.org>
## List-Unsubscribe: <https://listman.spamassassin.taint.org/mailman/listinfo/exmh-workers>,
##     <mailto:exmh-workers-request@redhat.com?subject=unsubscribe>
## List-Archive: <https://listman.spamassassin.taint.org/mailman/private/exmh-workers/>
## Date: Thu, 22 Aug 2002 18:26:25 +0700
##
##     Date:        Wed, 21 Aug 2002 10:54:46 -0500
##     From:        Chris Garrigues <cwg-dated-1030377287.06fa6d@DeepEddy.Com>
##     Message-ID:  <1029945287.4797.TMDA@deepeddy.vircio.com>
##
##
##   | I can't reproduce this error.
##
## For me it is very repeatable... (like every time, without fail).
```

```
##
## This is the debug log of the pick happening ...
##
## 18:19:03 Pick_It {exec pick +inbox -list -lbrace -lbrace -subject ftp -rbrace -rbrace} {4852-4852 -se
## 18:19:03 exec pick +inbox -list -lbrace -lbrace -subject ftp -rbrace -rbrace 4852-4852 -sequence merc
## 18:19:04 Ftoc_PickMsgs {{1 hit}}
## 18:19:04 Marking 1 hits
## 18:19:04 tkerror: syntax error in expression "int ...
##
## Note, if I run the pick command by hand ...
##
## delta$ pick +inbox -list -lbrace -lbrace -subject ftp -rbrace -rbrace  4852-4852 -sequence mercury
## 1 hit
##
## That's where the "1 hit" comes from (obviously).  The version of nmh I'm
## using is ...
##
## delta$ pick -version
## pick -- nmh-1.0.4 [compiled on fuchsia.cs.mu.OZ.AU at Sun Mar 17 14:55:56 ICT 2002]
##
## And the relevant part of my .mh_profile ...
##
## delta$ mhparam pick
## -seq sel -list
##
##
## Since the pick command works, the sequence (actually, both of them, the
## one that's explicit on the command line, from the search popup, and the
## one that comes from .mh_profile) do get created.
##
## kre
##
## ps: this is still using the version of the code form a day ago, I haven't
## been able to reach the cvs repository today (local routing issue I think).
##
##
##
## _______________________________________________
## Exmh-workers mailing list
## Exmh-workers@redhat.com
## https://listman.redhat.com/mailman/listinfo/exmh-workers
```

```r
cat("\n")
```

```r
# Converting the data into Corpus and removing data without tm package
ham_2 <- Corpus(VectorSource(ham_file_names))

# Preprocessing function
preprocess_text <- function(text) {
  text <- tolower(text)
  text <- gsub("\\d+", "", text)  # removeNumbers
  text <- gsub("[[:punct:]]", "", text)  # removePunctuation
  text <- stripWhitespace(text)
  text <- removeWords(text, stopwords("english"))
  text <- removeWords(text, c("will", "the"))
  return(text)
}

# Apply preprocessing to each document in the corpus
ham_2 <- lapply(ham_2$content, preprocess_text)

# Now let's build a matrix and dataframe to show the number of words to make wordcloud
tdm_ham_2 <- TermDocumentMatrix(Corpus(VectorSource(ham_2)))
m_h <- as.matrix(tdm_ham_2)
v_h <- sort(rowSums(m_h), decreasing=TRUE)
d_h <- data.frame(ham2 = names(v_h), freq = v_h)
head(d_h, 40)
```

```
##                                    ham2 freq
## cbaabadc                       cbaabadc    1
## ceeefcdbeeffac           ceeefcdbeeffac    1
## ecceebeadccfef           ecceebeadccfef    1
## cbbccccaf                     cbbccccaf    1
## bfcdeafbcecdbdda       bfcdeafbcecdbdda    1
## eafaccfabbb                 eafaccfabbb    1
## aafcaaeaffed               aafcaaeaffed    1
## ddcfedcb                       ddcfedcb    1
## ecabcecbfdebed           ecabcecbfdebed    1
## dccacecb                       dccacecb    1
## fbcdebbdbaafcededde fbcdebbdbaafcededde    1
## abcdafbecea                 abcdafbecea    1
## cdbedcddeeeea             cdbedcddeeeea    1
## cbebbfcbaca                 cbebbfcbaca    1
## dbacdbcefd                   dbacdbcefd    1
## efceffebe                     efceffebe    1
## efdfcfeabeb                 efdfcfeabeb    1
## feebaddebafbc             feebaddebafbc    1
## cbdbdc                           cbdbdc    1
## defcbceccfe                 defcbceccfe    1
## ccbdebadc                     ccbdebadc    1
## fcdefdfdd                     fcdefdfdd    1
## eeeadfdeebcbefc         eeeadfdeebcbefc    1
## ccbcebeae                     ccbcebeae    1
## dbdcffaebb                   dbdcffaebb    1
## ffbceedbddbcff           ffbceedbddbcff    1
## dddceafdefdcebb         dddceafdefdcebb    1
## ddbaecbaecddb             ddbaecbaecddb    1
## dabbede                         dabbede    1
## ccecdffaeedef             ccecdffaeedef    1
## aceffbdcabb                 aceffbdcabb    1
## eabcaafbfcdaa             eabcaafbfcdaa    1
## cebdcccffebdce           cebdcccffebdce    1
## dabeafcdffddfebb       dabeafcdffddfebb    1
## eadddfcdaeee               eadddfcdaeee    1
## edcdbb                           edcdbb    1
## aaffbdafbcfadec         aaffbdafbcfadec    1
## cdafebdbcda                 cdafebdbcda    1
## beedcebdeed                 beedcebdeed    1
## eecdfbcebcdefafac     eecdfbcebcdefafac    1
```

```
str(content)
```

```
## function (x)
```

Now, let’s clean spam

```
# Define a function to clean the email content
cleanEmailContent <- function(content) {
  # Remove non-alphanumeric characters
  cleaned_content <- gsub("[^a-zA-Z0-9\\s]", "", content)

  # Remove extra whitespaces
  cleaned_content <- gsub("\\s+", " ", cleaned_content)

  return(cleaned_content)
}

# Define the spam directory
spam_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/spam_2"

# List of file names in the spam directory
spam_file_names <- list.files(spam_directory, full.names = FALSE)

# Choose one file to extract content
selected_file <- spam_file_names[1]  # Change the index as needed

# Construct the full path to the file
file_path <- file.path(spam_directory, selected_file)

# Read the content of the file
content <- readLines(file_path)

# Clean the content
cleaned_content <- cleanEmailContent(content)

# Print or process the cleaned content as needed
cat("Cleaned Content of", selected_file, ":\n")
```

```
## Cleaned Content of 00001.317e78fa8ee2f54cd4890fdc09ba8176 :
```

```
cat(cleaned_content, sep = "\n")
```

```
## FromilugadminlinuxieTueAug61151022002
## ReturnPathilugadminlinuxie
## DeliveredToyyyylocalhostnetnoteinccom
## Receivedfromlocalhostlocalhost127001
## byphoboslabsnetnoteinccomPostfixwithESMTPid9E1F5441DD
## forjmlocalhostTue6Aug20020648090400EDT
## Receivedfromphobos127001
## bylocalhostwithIMAPfetchmail590
## forjmlocalhostsingledropTue06Aug20021148090100IST
## Receivedfromlughtuathaorgrootlughtuathaorg19412514545by
## dogmaslashnullorg81168116withESMTPidg72LqWv13294for
## jmilugjmasonorgFri2Aug20022252320100
## Receivedfromlughrootlocalhost127001bylughtuathaorg
## 893893withESMTPidWAA31224Fri2Aug20022250170100
## Receivedfrombettyjagessarcomw142z064000057nycnydslcncnet
## 64057142bylughtuathaorg893893withESMTPidWAA31201for
## iluglinuxieFri2Aug20022250110100
## XAuthenticationWarninglughtuathaorgHostw142z064000057nycnydslcncnet
## 64057142claimedtobebettyjagessarcom
## Receivedfrom640571422026316534bybettyjagessarcom
## SMTPD32706EVALidA42A7FC01F2Fri02Aug20020218180400
## MessageId1028311679886057142
## DateFri02Aug20022337590530
## Toiluglinuxie
## FromStartNowstartnow2002hotmailcom
## MIMEVersion10
## ContentTypetextplaincharsetUSASCIIformatflowed
## SubjectILUGSTOPTHEMLMINSANITY
## Senderilugadminlinuxie
## ErrorsToilugadminlinuxie
## XMailmanVersion11
## Precedencebulk
## ListIdIrishLinuxUsersGroupiluglinuxie
## XBeenthereiluglinuxie
##
## Greetings
##
## Youarereceivingthisletterbecauseyouhaveexpressedaninterestin
## receivinginformationaboutonlinebusinessopportunitiesIfthisis
## erroneousthenpleaseacceptmymostsincereapologyThisisaonetime
## mailingsonoremovalisnecessary
##
## Ifyouvebeenburnedbetrayedandbackstabbedbymultilevelmarketing
## MLMthenpleasereadthisletterItcouldbethemostimportantonethat
## haseverlandedinyourInbox
##
## MULTILEVELMARKETINGISAHUGEMISTAKEFORMOSTPEOPLE
##
## MLMhasfailedtodeliveronitspromisesforthepast50yearsThepursuit
## oftheMLMDreamhascosthundredsofthousandsofpeopletheirfriends
## theirfortunesandtheirsacredhonorThefactisthatMLMisfatally
## flawedmeaningthatitCANNOTworkformostpeople
##
## ThecompaniesandthefewwhoearnthebigmoneyinMLMareNOTgoingto
## tellyoutherealstoryFINALLYthereissomeonewhohasthecourageto
## cutthroughthehypeandliesandtelltheTRUTHaboutMLM
##
## HERESGOODNEWS
##
## ThereISanalternativetoMLMthatWORKSandworksBIGIfyouhaventyet
## abandonedyourdreamsthenyouneedtoseethisEarningthekindofincome
## youvedreamedaboutiseasierthanyouthink
##
## WithyourpermissionIdliketosendyouabriefletterthatwilltellyou
## WHYMLMdoesntworkformostpeopleandwillthenintroduceyouto
## somethingsonewandrefreshingthatyoullwonderwhyyouhaventheardof
## thisbefore
```

```
##
## IpromisethattherewillbeNOunwantedfollowupNOsalespitchnoone
## willcallyouandyouremailaddresswillonlybeusedtosendyouthe
## informationPeriod
##
## ToreceivethisfreelifechanginginformationsimplyclickReplytype
## SendInfointheSubjectboxandhitSendIllgettheinformationtoyou
## within24hoursJustlookforthewordsMLMWALLOFSHAMEinyourInbox
##
## Cordially
##
## Siddhi
##
## PSSomeonerecentlysenttheletterttomeandithasbeenthemost
## eyeopeningfinanciallybeneficialinformationIhaveeverreceivedI
## honestlybelievethatyouwillfeelthesamewayonceyouvereaditAnd
## itsFREE
##
##
##
## ThisemailisNEVERsentunsolicitedTHISISNOTSPAMYouarereceiving
## thisemailbecauseyouEXPLICITLYsignedyourselfuptoourlistwithour
## onlinesignupformorthroughuseofourFFALinksPageandEMailDOM
## systemswhichhaveEXPLICITtermsofusewhichstatethatthroughitsuse
## youagreetoreceiveouremailingsYoumayalsobeamemberofaAltra
## ComputerSystemslistoroneofmanynumerousFREEMarketingServicesandas
## suchyouagreedwhenyousignedupforsuchlistthatyouwouldalsobe
## receivingthisemailing
## Duetotheabovethisemailmessagecannotbeconsideredunsolicitatedor
## spam
##
##
##
##
##
##
## IrishLinuxUsersGroupiluglinuxie
## httpwwwlinuxiemailmanlistinfoilugforunsubscriptioninformation
## Listmaintainerlistmasterlinuxie
```

```r
cat("\n")
```

Let’s clean ham

```r
cleanEmailContent <- function(content) {
  # Remove non-alphanumeric characters
  cleaned_content <- gsub("[^a-zA-Z0-9\\s]", "", content)

  # Remove extra whitespaces
  cleaned_content <- gsub("\\s+", " ", cleaned_content)

  return(cleaned_content)
}
```

```r
# Define the ham directory
ham_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/easy_ham"

# List of file names in the spam directory
ham_file_names <- list.files(ham_directory, full.names = FALSE)

# Choose one file to extract content
selected_file <- ham_file_names[1]

# Construct the full path to the file
file_path <- file.path(ham_directory, selected_file)

# Read the content of the file
content <- readLines(file_path)

# Clean the content
cleaned_content<-cleanEmailContent(content)

# Print or process the cleaned content as needed
cat("Cleaned Content of", selected_file, ":\n")
```

```
## Cleaned Content of 00001.7c53336b37003a9286aba55d2945844c :
```

```r
cat(cleaned_content, sep = "\n")
```

```
## FromexmhworkersadminredhatcomThuAug221236232002
## ReturnPathexmhworkersadminspamassassintaintorg
## DeliveredTozzzzlocalhostnetnoteinccom
## Receivedfromlocalhostlocalhost127001
## byphoboslabsnetnoteinccomPostfixwithESMTPidD03E543C36
## forzzzzlocalhostThu22Aug20020736160400EDT
## Receivedfromphobos127001
## bylocalhostwithIMAPfetchmail590
## forzzzzlocalhostsingledropThu22Aug20021236160100IST
## Receivedfromlistmanspamassassintaintorglistmanspamassassintaintorg66187233211by
## dogmaslashnullorg81168116withESMTPidg7MBYrZ04811for
## zzzzexmhspamassassintaintorgThu22Aug20021234530100
## Receivedfromlistmanspamassassintaintorglocalhostlocaldomain127001by
## listmanredhatcomPostfixwithESMTPid8386540858Thu22Aug2002
## 0735020400EDT
## DeliveredToexmhworkerslistmanspamassassintaintorg
## Receivedfromintmx1corpspamassassintaintorgintmx1corpspamassassintaintorg
## 1721652254bylistmanredhatcomPostfixwithESMTPid10CF8406D7
## forexmhworkerslistmanredhatcomThu22Aug20020734100400
## EDT
## Receivedfrommaillocalhostbyintmx1corpspamassassintaintorg81168116
## idg7MBY7g11259forexmhworkerslistmanredhatcomThu22Aug2002
## 0734070400
## Receivedfrommx1spamassassintaintorgmx1spamassassintaintorg172164831by
## intmx1corpredhatcom81168116withSMTPidg7MBY7Y11255for
## exmhworkersredhatcomThu22Aug20020734070400
## Receivedfromratreepsuacth20228976bymx1spamassassintaintorg
## 81168116withSMTPidg7MBIhl25223forexmhworkersredhatcom
## Thu22Aug20020718550400
## ReceivedfromdeltacsmuOZAUdeltacoepsuacth17230098by
## ratreepsuacth81168116withESMTPidg7MBWel29762
## Thu22Aug20021832400700ICT
## ReceivedfrommunnariOZAUlocalhost127001bydeltacsmuOZAU
## 81168116withESMTPidg7MBQPW13260Thu22Aug2002182625
## 0700ICT
## FromRobertElzkremunnariOZAU
## ToChrisGarriguescwgdated103037728706fa6dDeepEddyCom
## Ccexmhworkersspamassassintaintorg
## SubjectReNewSequencesWindow
## InReplyTo10299452874797TMDAdeepeddyvirciocom
## References10299452874797TMDAdeepeddyvirciocom
## 10298824683116TMDAdeepeddyvirciocom96271029933001munnariOZAU
## 102994306626919TMDAdeepeddyvirciocom
## 1029944441398TMDAdeepeddyvirciocom
## MIMEVersion10
## ContentTypetextplaincharsetusascii
## MessageId132581030015585munnariOZAU
## XLoopexmhworkersspamassassintaintorg
## Senderexmhworkersadminspamassassintaintorg
## ErrorsToexmhworkersadminspamassassintaintorg
## XBeenthereexmhworkersspamassassintaintorg
## XMailmanVersion201
## Precedencebulk
## ListHelpmailtoexmhworkersrequestspamassassintaintorgsubjecthelp
## ListPostmailtoexmhworkersspamassassintaintorg
## ListSubscribehttpslistmanspamassassintaintorgmailmanlistinfoexmhworkers
## mailtoexmhworkersrequestredhatcomsubjectsubscribe
## ListIdDiscussionlistforEXMHdevelopersexmhworkersspamassassintaintorg
## ListUnsubscribehttpslistmanspamassassintaintorgmailmanlistinfoexmhworkers
## mailtoexmhworkersrequestredhatcomsubjectunsubscribe
## ListArchivehttpslistmanspamassassintaintorgmailmanprivateexmhworkers
## DateThu22Aug20021826250700
##
## DateWed21Aug20021054460500
## FromChrisGarriguescwgdated103037728706fa6dDeepEddyCom
## MessageID10299452874797TMDAdeepeddyvirciocom
##
##
## Icantreproducethiserror
##
## Formeitisveryrepeatablelikeeverytimewithoutfail
##
## Thisisthedebuglogofthepickhappening
##
## 181903PickItexecpickinboxlistlbracelbracesubjectftprbracerbrace48524852sequencemercury
## 181903execpickinboxlistlbracelbracesubjectftprbracerbrace48524852sequencemercury
## 181904FtocPickMsgs1hit
## 181904Marking1hits
## 181904tkerrorsyntaxerrorinexpressionint
##
## NoteifIrunthepickcommandbyhand
##
## deltapickinboxlistlbracelbracesubjectftprbracerbrace48524852sequencemercury
## 1hit
##
## Thatswherethe1hitcomesfromobviouslyTheversionofnmhIm
## usingis
##
## deltapickversion
## picknmh104compiledonfuchsiacsmuOZAUatSunMar17145556ICT2002
##
## Andtherelevantpartofmymhprofile
##
## deltamhparampick
## seqsellist
##
##
## Sincethepickcommandworksthesequenceactuallybothofthemthe
## onethatsexplicitonthecommandlinefromthesearchpopupandthe
## onethatcomesfrommhprofiledogetcreated
##
## kre
##
## psthisisstillusingtheversionofthecodeformadayagoIhavent
## beenabletoreachthecvsrepositorytodaylocalroutingissueIthink
##
##
##
##
## Exmhworkersmailinglist
## Exmhworkersredhatcom
## httpslistmanredhatcommailmanlistinfoexmhworkers
```

```
cat("\n")
```

Let's convert spam into a matrix

```
# Function to preprocess and clean the text
preprocessText <- function(text) {
  # Convert to lowercase
  text <- tolower(text)
  # Remove numbers
  text <- removeNumbers(text)

  # Remove punctuation
  text <- removePunctuation(text)
  # Remove stopwords
  text <- removeWords(text, stopwords("en"))
  # Strip unnecessary whitespaces
  text <- stripWhitespace(text)

  return(text)
}

# Define the spam directory
spam_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/spam_2"
```

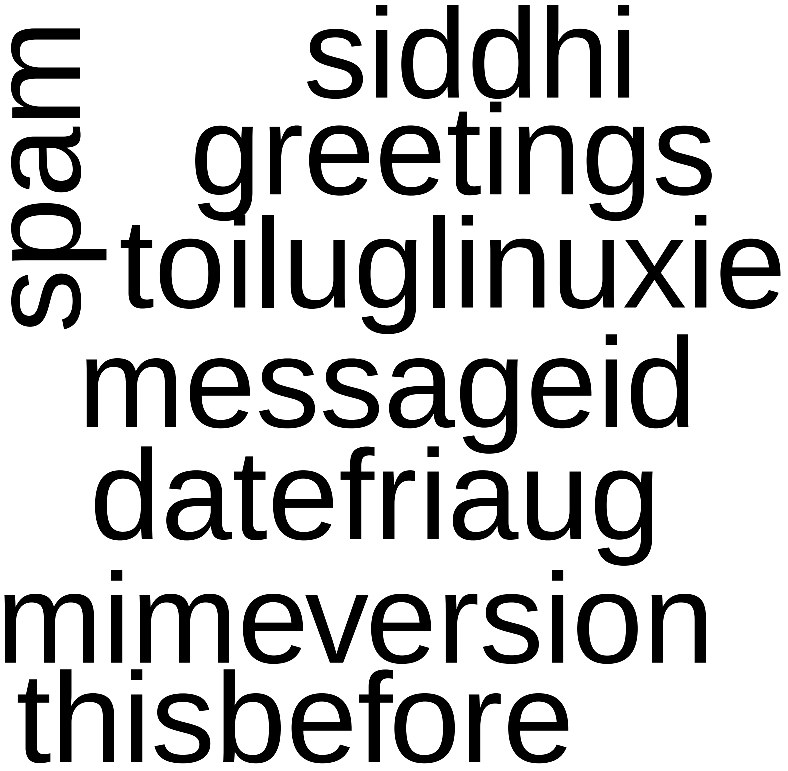
```r
# List of file names in the spam directory
spam_file_names <- list.files(spam_directory, full.names = FALSE)

# Choose one file to extract content
selected_file <- spam_file_names[1]  # Change the index as needed

# Construct the full path to the file
file_path <- file.path(spam_directory, selected_file)

# Read the content of the file
content <- readLines(file_path)

# Clean the content
cleaned_content <- cleanEmailContent(content)

# Preprocess the cleaned content
preprocessed_content <- preprocessText(cleaned_content)

# Create a Corpus
corpus1 <- Corpus(VectorSource(preprocessed_content))

# Create a Document-Term Matrix (DTM)
dtm1 <- DocumentTermMatrix(corpus1)

# Convert DTM to a matrix
matrix <- as.matrix(dtm1)
```

```r
spam_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/spam_2"
tdm_s <- TermDocumentMatrix((corpus1))
m_s <- as.matrix(tdm_s)
v_s <- sort(rowSums(m_s), decreasing=TRUE)
d_s <- data.frame(spam_2= names(v_s), freq=v_s)
head(d_s,5)
```

```
##                                                                                          spam_2
## fromilugadminlinuxietueaug                                           fromilugadminlinuxietueaug
## returnpathilugadminlinuxie                                           returnpathilugadminlinuxie
## deliveredtoyyyylocalhostnetnoteinccom                     deliveredtoyyyylocalhostnetnoteinccom
## receivedfromlocalhostlocalhost                                   receivedfromlocalhostlocalhost
## byphoboslabsnetnoteinccompostfixwithesmtpidefdd byphoboslabsnetnoteinccompostfixwithesmtpidefdd
##                                                 freq
## fromilugadminlinuxietueaug                         1
## returnpathilugadminlinuxie                         1
## deliveredtoyyyylocalhostnetnoteinccom              1
## receivedfromlocalhostlocalhost                     1
## byphoboslabsnetnoteinccompostfixwithesmtpidefdd    1
```

We can make use of wordcloud function to visualize the most common words in either the spam or the ham corpus. ### Let’s visualize spam corpus as a wordcloud

```r
# spam corpus-wordcloud
set.seed(131017)

wordcloud(corpus1, max.words = 500, random.order = FALSE)
```



Let's convert ham into a matrix

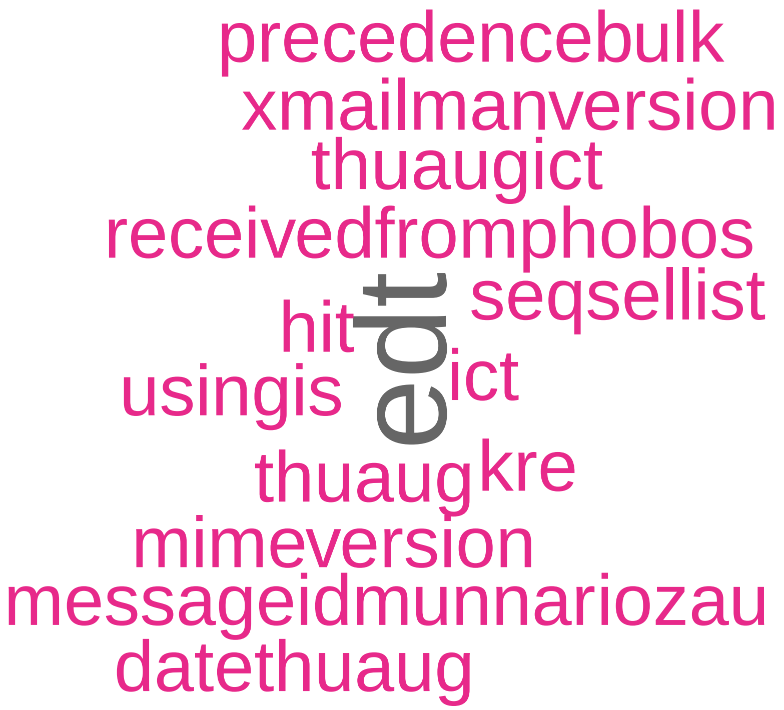
```r
# Function to preprocess and clean the text
preprocessText <- function(text) {
  # Convert to lowercase
  text <- tolower(text)
  # Remove numbers
  text <- removeNumbers(text)
  # Remove punctuation
  text <- removePunctuation(text)
  # Remove stopwords
  text <- removeWords(text, stopwords("en"))
  # Strip unnecessary whitespaces
  text <- stripWhitespace(text)

  return(text)
}

# Define the ham directory
ham_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/easy_ham"

# List of file names in the spam directory
ham_file_names <- list.files(ham_directory, full.names = FALSE)

# Choose one file to extract content
selected_file <- ham_file_names[1]  # Change the index as needed

# Construct the full path to the file
```

```r
file_path <- file.path(ham_directory, selected_file)

# Read the content of the file
content <- readLines(file_path)

# Clean the content
cleaned_content <- cleanEmailContent(content)

# Preprocess the cleaned content
preprocessed_content <- preprocessText(cleaned_content)
 # Split by non-word characters
 # Install and load the tm package if not already installed
if (!require("tm")) install.packages("tm", dependencies=TRUE)
library(tm)

# Function to tokenize text, handling specific cases
custom_tokenizer <- function(x) {
  # Split by non-word characters
  tokens <- unlist(strsplit(x, "\\W+"))

  # Handle specific cases
  tokens <- gsub("spamassassin", "spam assassin", tokens, ignore.case = TRUE)
  tokens <- gsub("received", "receive ed", tokens, ignore.case = TRUE)
  tokens <- gsub("delivered", "deliver ed", tokens, ignore.case = TRUE)
  tokens <- gsub("\\b(for|with)\\b", " \\1 ", tokens, ignore.case = TRUE)

  # Remove empty strings
  tokens <- tokens[tokens != ""]

  return(tokens)
}

# Create a Corpus
corpus <- Corpus(VectorSource(preprocessed_content))

# Create a Document-Term Matrix (DTM) with custom tokenizer
dtm2 <- DocumentTermMatrix(corpus, control = list(tokenize = custom_tokenizer))

# Convert DTM to a matrix
matrix <- as.matrix(dtm2)

# Convert the matrix to a data frame for better visualization
df <- as.data.frame(matrix)

# Create a Corpus
corpus2 <- Corpus(VectorSource(preprocessed_content))

# Create a Document-Term Matrix (DTM) with custom tokenizer
dtm2 <- DocumentTermMatrix(corpus2, control = list(tokenize = custom_tokenizer))

# Convert DTM to a matrix
matrix <- as.matrix(dtm2)
# Convert the matrix to a data frame for better visualization
```

```
df <- as.data.frame(matrix)
```

Let's visualize ham corpus

```
#Visualizing ham as a wordcloud
# Set Seed
set.seed(3300)

ham_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/easy_ham"

wordcloud(corpus2, max.words = 2000, random.order = FALSE, min.freq=60,colors=brewer.pal(8,"Dark2"))
```



```
#Reduce sparsity -ham
dtm2_filtered <- removeSparseTerms(dtm2, 0.99)
inspect(dtm2_filtered)
```

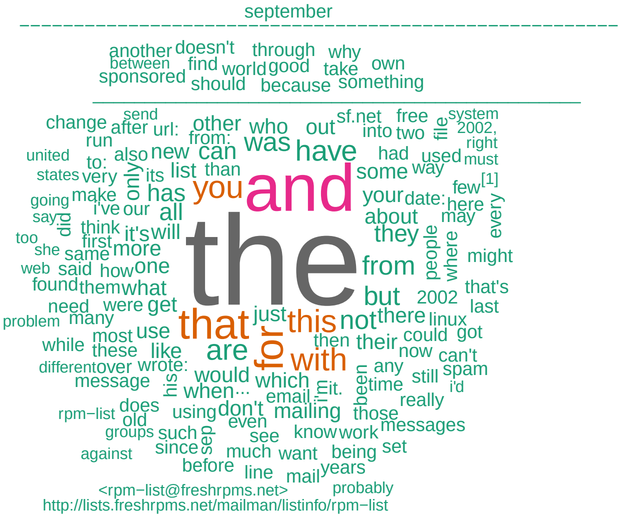
```
## <<DocumentTermMatrix (documents: 113, terms: 2)>>
## Non-/sparse entries: 4/222
## Sparsity           : 98%
## Maximal term length: 21
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs edt tmdadeepeddyvirciocom
##   1    0                     0
##   15   1                     0
##   2    0                     0
##   20   1                     0
##   3    0                     0
##   4    0                     0
##   43   0                     1
##   44   0                     1
##   5    0                     0
##   6    0                     0
```

```
#Reduce sparsity - spam
dtm1_filtered <- removeSparseTerms(dtm1, 0.99)
```

```
# Function to process each file
process_file <- function(file_path) {
  input_file <- readLines(file_path, warn = FALSE)

  # Find the index of the first blank row
  first_blank_row <- which(input_file == "")

  if (length(first_blank_row) == 0) {
    # If no blank row is found, use the length of the vector
    first_blank_row <- length(input_file)
  } else {
    # Use the first occurrence of a blank row
    first_blank_row <- first_blank_row[1] - 1
  }

  # Check if first_blank_row is valid
  if (first_blank_row > 0) {
    body <- input_file[-(1:first_blank_row)]
    body <- paste(body, collapse = " ")
    data.frame(content = body, doc_num = basename(file_path), stringsAsFactors = FALSE)
  } else {
    # If first_blank_row is not valid, return a data frame with NA
    data.frame(content = NA, doc_num = basename(file_path), stringsAsFactors = FALSE)
  }
}

# Directory containing ham files
ham_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/easy_ham"

# List of ham files
ham_files <- list.files(ham_directory, full.names = TRUE)

# Process each file and bind the results into a data frame
ham_df <- map_dfr(ham_files, process_file)

# Remove rows with NA content
ham_df <- ham_df %>% filter(!is.na(content))
```

Let’s visualize the ham dataframe

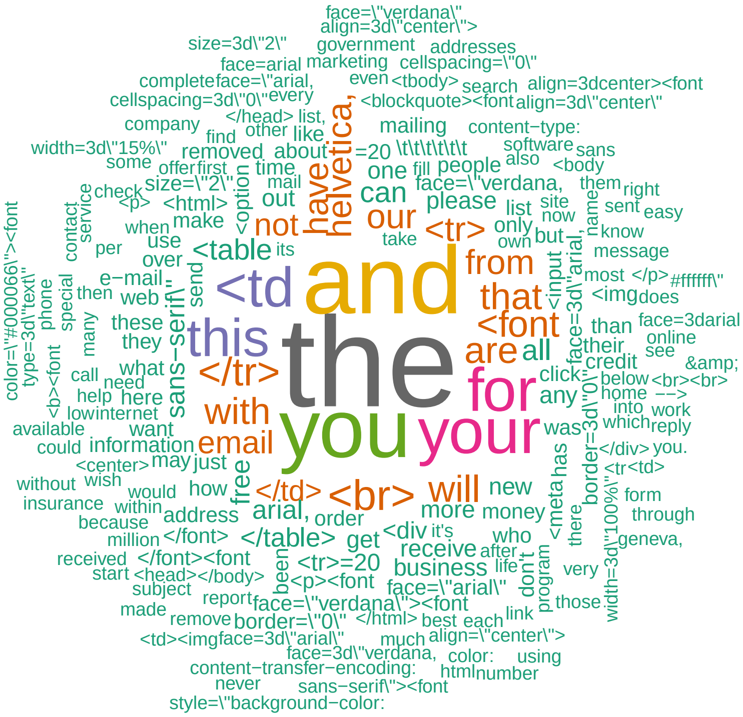
```
set.seed(131017)
wordcloud(ham_df, max.words = 1000, random.order = FALSE, min.freq=250,colors=brewer.pal(8,"Dark2"))
```



```r
# Function to process each file
process_file <- function(file_path) {
  input_file <- readLines(file_path, warn = FALSE)

  # Find the index of the first blank row
  first_blank_row <- which(input_file == "")

  if (length(first_blank_row) == 0) {
    # If no blank row is found, use the length of the vector
    first_blank_row <- length(input_file)
  } else {
    # Use the first occurrence of a blank row
    first_blank_row <- first_blank_row[1] - 1
  }

  # Check if first_blank_row is valid
  if (first_blank_row > 0) {
    body <- input_file[-(1:first_blank_row)]
    body <- paste(body, collapse = " ")
    data.frame(content = body, doc_num = basename(file_path), stringsAsFactors = FALSE)
  } else {
    # If first_blank_row is not valid, return a data frame with NA
    data.frame(content = NA, doc_num = basename(file_path), stringsAsFactors = FALSE)
  }
}

# Directory containing ham files
ham_directory <- "/Users/Heleine/Library/Mobile Documents/com~apple~CloudDocs/spamham/spam_2"

# List of ham files
spam_files <- list.files(spam_directory, full.names = TRUE)

# Process each file and bind the results into a data frame
spam_df <- map_dfr(spam_files, process_file)

# Remove rows with NA content
spam_df <- spam_df %>% filter(!is.na(content))
```

Let’s visualize the spam data frame

```
set.seed(131017)
wordcloud(spam_df, max.words = 1000, random.order = FALSE, min.freq=250,colors=brewer.pal(8,"Dark2"))
```



```
ham_corpus <- VCorpus(VectorSource(ham_df$content))
spam_corpus <- VCorpus(VectorSource(spam_df$content))
corpus <- c(ham_corpus,spam_corpus)
```

```
# leveraging library(text)
corpus <- corpus %>% tm_map(content_transformer(PlainTextDocument))
corpus <- corpus %>% tm_map(content_transformer(removePunctuation))

# Function to set encoding and perform text preprocessing
preprocess_text <- function(text) {
  text <- iconv(text, to = "UTF-8", sub = "byte")
  text <- tolower(text)
  text <- removeNumbers(text)
  text <- removeWords(text, stopwords("en"))
  return(text)
}

# Function to set encoding and perform text preprocessing
preprocess_text <- function(text) {
  # Ensure 'text' is a character vector
  if (!is.character(text)) {
    warning("Input is not a character vector. Returning unchanged.")
    return(text)
  }

  # Convert text to UTF-8 encoding
  text <- iconv(text, to = "UTF-8", sub = "byte")

  # Perform text preprocessing
  text <- tolower(text)
  text <- remove_numbers(text)
  text <- remove_words(text, stopwords = stopwords("en"))

  return(text)
}
```

Tidying up document-term matrix

```
# Function to preprocess text
preprocess_text <- function(text) {
  # Convert text to UTF-8 encoding
  text <- iconv(text, to = "UTF-8", sub = "byte")

  # Perform text preprocessing
  text <- tolower(text)
  text <- removeNumbers(text)

  # Split the text into a character vector of words
  words <- unlist(strsplit(text, "\\s+"))

  # Remove common English stopwords
  words <- setdiff(words, stopwords("en"))

  # Combine the words back into a preprocessed text
  preprocessed_text <- paste(words, collapse = " ")

  return(preprocessed_text)
}

# Apply text preprocessing to each document in the corpus
corpus <- lapply(corpus, function(doc) {
  doc$content <- preprocess_text(doc$content)
  return(doc)
})

# Create Document-Term Matrix (DTM) and remove sparse terms
dtm <- DocumentTermMatrix(corpus) %>% removeSparseTerms(0.95)

# Convert DTM to a tidy format
library(tidytext)

tidy_dtm <- tidy(dtm)


# Add a classification column based on the document index
tidy_dtm$classification <- ifelse(tidy_dtm$document <= length(ham_files), "ham", "spam")

# Display the head and structure of the tidy DTM
head(tidy_dtm)
```

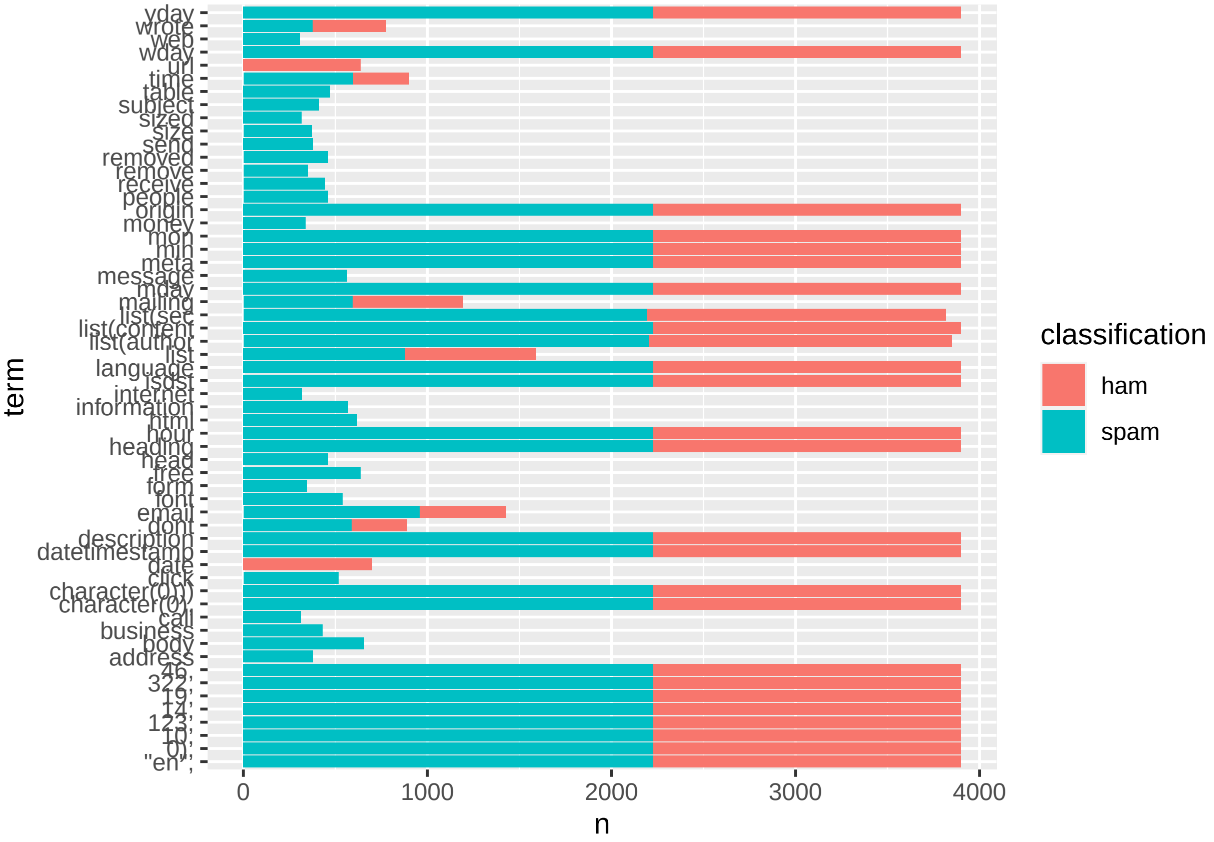
```
## # A tibble: 6 x 4
##   document term      count classification
##   <chr>    <chr>     <dbl> <chr>
## 1 1        "\"en\","     1 ham
## 2 1        "0),"         1 ham
## 3 1        "10,"         1 ham
## 4 1        "123,"        1 ham
## 5 1        "14,"         1 ham
## 6 1        "19,"         1 ham
```

```
glimpse(tidy_dtm)
```

```
## Rows: 208,557
## Columns: 4
## $ document       <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", ~
## $ term           <chr> "\"en\",", "0),", "10,", "123,", "14,", "19,", "322,", ~
## $ count          <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1~
## $ classification <chr> "ham", "ham", "ham", "ham", "ham", "ham", "ham", "ham",~
```

Let’s visualize both spam and ham with a barplot

```
data("stop_words")
tidy_dtm <- tidy_dtm %>%
  anti_join(stop_words, by= c("term"="word"))


# find common words
tidy_dtm %>%  group_by(term, classification) %>%
              count() %>% arrange(desc(n)) %>%
              filter(n>300) %>%
              ggplot(aes(x=term,y=n,fill=classification)) +
                  geom_bar(stat='identity') +
                  coord_flip()
```

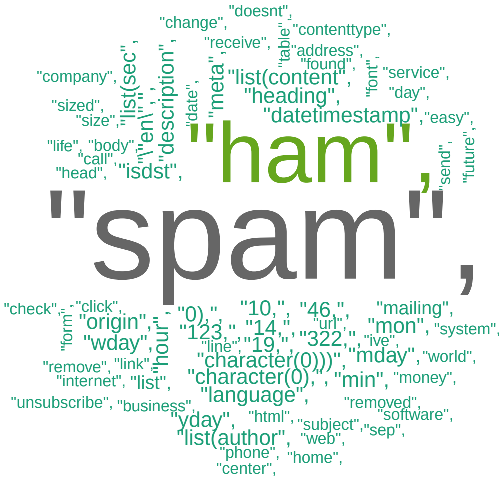


Adding more stop words

```
# I added more stop words based on the barplot. This was done in an iterative manner by finding words c
more_stops <- tibble(term = c("wrote","time", "people","email","free","sponsored", "people","message","
tidy_dtm <- tidy_dtm %>%
  anti_join(more_stops, by= "term")
```

Let’s now visualize both spam and ham using a wordcloud

```
# Set Seed
set.seed(3300)
wordcloud(tidy_dtm, max.words = 2000, random.order = FALSE, min.freq=300,colors=brewer.pal(8,"Dark2"))
```



Top ham words

```
#top ham words
tidy_dtm %>% filter(classification=='ham') %>%  group_by(term) %>% count() %>% arrange(desc(n))
```

```
## # A tibble: 159 x 2
## # Groups:   term [159]
##    term                 n
##    <chr>            <int>
##  1 "\"en\","         1671
##  2 "0),"             1671
##  3 "10,"             1671
##  4 "123,"            1671
##  5 "14,"             1671
##  6 "19,"             1671
##  7 "322,"            1671
##  8 "46,"             1671
##  9 "character(0)))"  1671
## 10 "character(0),"   1671
## # i 149 more rows
```

Top spam words

```
#top spam words
tidy_dtm %>% filter(classification=='spam') %>%  group_by(term) %>% count() %>% arrange(desc(n))
```

```
## # A tibble: 168 x 2
## # Groups:   term [168]
##    term                 n
##    <chr>            <int>
##  1 "\"en\","         2227
##  2 "0),"             2227
##  3 "10,"             2227
##  4 "123,"            2227
##  5 "14,"             2227
##  6 "19,"             2227
##  7 "322,"            2227
##  8 "46,"             2227
##  9 "character(0)))"  2227
## 10 "character(0),"   2227
## # i 158 more rows
```

Classification model

Now that the data is prepared a model can be run that predicts document classification. It is custom for the data to be split 75%/25% for training and test sets.

First, let's split into training and test sets

```
# Set Seed
set.seed(3300)

n <- nrow(tidy_dtm)
sample_size <- floor(0.10 * n)

# Generate random indices for the sample
sample_indices <- sample(1:n, size = sample_size, replace = FALSE)

# Create training and test sets
train <- tidy_dtm[sample_indices, ]
test <- tidy_dtm[-sample_indices, ]

# Convert to data frame and factor
train <- as.data.frame(train)
train$classification <- as.factor(train$classification)
```

Then, let's summarize classification

```
(summary(train$classification))
```

```
##  ham spam
## 5546 8993
```

```
(summary(train))
```

```
##    document             term               count       classification
##  Length:14539       Length:14539       Min.   :1.000   ham :5546
##  Class :character   Class :character   1st Qu.:1.000   spam:8993
##  Mode  :character   Mode  :character   Median :1.000
##                                        Mean   :1.057
##                                        3rd Qu.:1.000
##                                        Max.   :3.000
```