Enron Email Text Cluster Analysis

Sean O'Malley

```
## Loading required package: tm
```

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

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
## Loading required package: cluster
```

```
## Loading required package: ggplot2
```

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

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

Create Corpus

```
enron_corp <- Corpus(VectorSource(enron$message))
```

Preprocess Corpus

```
### Lowercase

enron_corp <- tm_map(enron_corp, content_transformer(tolower))

### Replace Symbol with Space Function
  # source https://eight2late.wordpress.com/2015/07/22/a-gentle-introduction-to-cluster-analysis-us
ing-r/

gsubSpace <- content_transformer(function(x,y){ return (gsub(y, " ", x))})

  enron_corp <- tm_map(enron_corp, gsubSpace, "-")
  enron_corp <- tm_map(enron_corp, gsubSpace, ":")
  enron_corp <- tm_map(enron_corp, gsubSpace, "'")
  enron_corp <- tm_map(enron_corp, gsubSpace, '"')

### Remove Punctuation / Numbers

enron_corp <- tm_map(enron_corp, removePunctuation)
enron_corp <- tm_map(enron_corp, removeNumbers)
enron_corp <- tm_map(enron_corp, stripWhitespace)

### Remove Stopwords

enron_corp <- tm_map(enron_corp, removeWords, stopwords("english"))

### Stem to root words

enron_corp <- tm_map(enron_corp,stemDocument)
```

Convert to Document Term Metrix

```
enron_dtm <- DocumentTermMatrix(enron_corp)

# Gives us an overview of what is going on in the dtm

inspect(enron_dtm)
```

```
## <<DocumentTermMatrix (documents: 100000, terms: 321131)>>
## Non-/sparse entries: 12790165/32100309835
## Sparsity           : 100%
## Maximal term length: 474
## Weighting          : term frequency (tf)
## Sample             :
##        Terms
## Docs    bcc content date enron folder messag origin subject type will
##   1516    1       2    4     0      1      7      2       3    1  100
##   24664   1       3    6     1      1      2      1       5    2  114
##   42025   1       4    7   565      1      4      6       4    3   68
##   44576   1       4   33     5      1      9      2       6    3  194
##   48117   1       2    2     0      1      2      1       2    2  117
##   76918   1       3    6     1      1      2      1       5    2  114
##   83464   1       2   13   628      1      1      3       4    1   83
##   84499   1       2   45     0      1      6      2       4    2  211
##   92111   1       4   33  1342      1      8      8      10    7  137
##   98452   1       2    2     0      1      2      1       2    2  117
```

```
## [8160] "clynescorpenronenron"
## [8161] "enl"
## [8162] "newscast"
## [8163] "fontfont"
## [8164] "ibj"
## [8165] "vpp"
## [8166] "nigeria"
## [8167] "fenner"
## [8168] "sheraton"
## [8169] "laden"
## [8170] "tibco"
## [8171] "shuttl"
## [8172] "quicklink"
## [8173] "daphnecobigplanetcom"
## [8174] "lwbthemarinebigplanetcom"
## [8175] "joani"
## [8176] "gss"
## [8177] "cohen"
## [8178] "icap"
## [8179] "prearrang"
## [8180] "fbi"
## [8181] "accentur"
## [8182] "nancysellersrobertmondavicom"
## [8183] "tale"
## [8184] "embassi"
## [8185] "sonat"
## [8186] "vjwcleanpowerorg"
## [8187] "elektro"
## [8188] "hes"
## [8189] "chevron"
## [8190] "dwp"
## [8191] "lineback"
## [8192] "methanol"
## [8193] "erichardsonsarofimcom"
## [8194] "cgarciaenroncom"
## [8195] "despeyenroncom"
## [8196] "mmitcheenroncom"
## [8197] "vversenenroncom"
## [8198] "dominicdimarecalchambercom"
## [8199] "klawcom"
## [8200] "gisbaolcom"
## [8201] "dasr"
## [8202] "buycom"
## [8203] "enewsbuycomcgi"
## [8204] "ptsize"
## [8205] "wwwmoneynetimagescleargif"
## [8206] "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
## [8207] "pinnacl"
## [8208] "alignright"
## [8209] "alignrighttdtd"
## [8210] "footballfantasysportslinecommpelinkscriptplay"
## [8211] "footballfantasysportslinecommpelinkscriptplayersleaguebigeownerrandomkey"
## [8212] "cuiaba"
## [8213] "rrga"
## [8214] "sherlyn"
## [8215] "mvc"
## [8216] "fonttdtr"
## [8217] "wwwfoolcommaspi"
## [8218] "wwwlnksrvcommaspi"
## [8219] "mso"
```

```
## [8220] "erichardsonsarofimcomenron"
## [8221] "cand"
## [8222] "wwwrigzonecomimagesspacergif"
## [8223] "insync"
## [8224] "borland"
## [8225] "footballfantasysportslinecommpelinkscriptplayersleagueeeeownerrandomkey"
## [8226] "fonttdtd"
## [8227] "classmsonorm"
## [8228] "spanp"
## [8229] "alignd"
## [8230] "footballfantasysportslinecommpelinkscriptmpplayersleaguebigeownerrandomkeyweek"
## [8231] "wwwdeltacomimagesemailprogramsspacergif"
```

```
# the huge size is killing my rstudio, this helps reduce size that out
enron_dtm <- removeSparseTerms(enron_dtm, sparse = 0.8)

inspect(enron_dtm[])
```

```
## <<DocumentTermMatrix (documents: 100000, terms: 45)>>
## Non-/sparse entries: 2543920/1956080
## Sparsity           : 43%
## Maximal term length: 17
## Weighting          : term frequency (tf)
## Sample             :
##        Terms
## Docs    bcc content date enron folder messag origin subject type will
##   28666   1       9   17     1      1     27     25      92    1   42
##   42025   1       4    7   565      1      4      6       4    3   68
##   44576   1       4   33     5      1      9      2       6    3  194
##   55187   1       4   23   201      1     31     36      53   15  142
##   59950   1       7    4   479      1      3      9       3    1   85
##   65702   1       4   23   201      1     31     36      53   15  142
##   6860    1       3    5   434      1      7      3       5    1   39
##   83464   1       2   13   628      1      1      3       4    1   83
##   84499   1       2   45     0      1      6      2       4    2  211
##   92111   1       4   33  1342      1      8      8      10    7  137
```

Weight DTM by TFIDF

```
enron_dtm_tfidf <- weightTfIdf(enron_dtm)

# Convert to input to matrix to best fit kmeans, make rownames nrow

enron_m <- as.matrix(enron_dtm_tfidf)
rownames(enron_m) <- 1:nrow(enron_m)
```

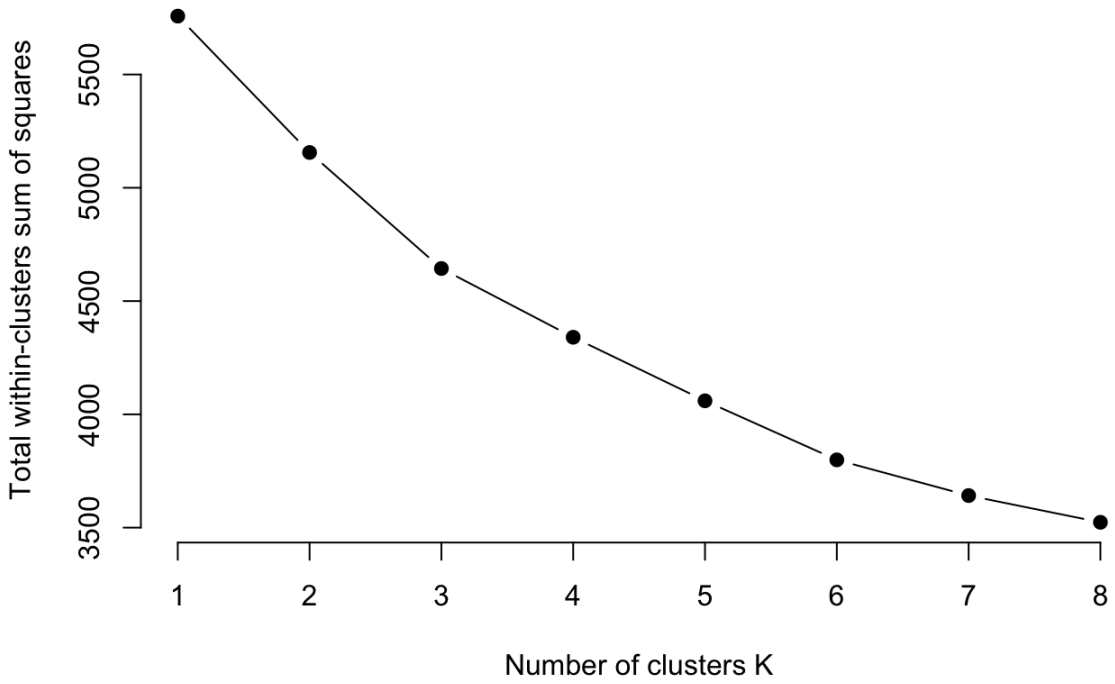
Find Optimal K for Kmeans using Elbow method

```
k_max <- 8

# Parallel Process KMeans Elbow Chart
library(parallel)
no_cores <- detectCores() - 1
cl <- makeCluster(no_cores)

clusterExport(cl, "enron_m")
wss <- parSapply(cl, 1:k_max, FUN = function(k){kmeans(enron_m, k, nstart=100, iter.max = 8 )$tot.w
ithinss})


plot(1:k_max, wss,
      type="b", pch = 19, frame = FALSE,
      xlab="Number of clusters K",
      ylab="Total within-clusters sum of squares")
```



K-Means Analysis (set to 3)

```
enron_k <- kmeans(enron_m, 3, nstart = 100)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5000000)
```

```
# What do our centers look like

enron_k$centers
```

```
##         ascii bcc          bit   charsetus content date   document encod
## 1 0.001449883   0 0.0008499491 0.001450669       0    0 0.01208953     0
## 2 0.003444894   0 0.0019878256 0.003448474       0    0 0.01950268     0
## 3 0.002139363   0 0.0012918772 0.002140857       0    0 0.01755261     0
##        enron filenam folder   foldersal javamailevansthym messag mime
## 1 0.01015778       0      0 0.006213446                  0      0    0
## 2 0.01840502       0      0 0.016598615                  0      0    0
## 3 0.28013386       0      0 0.010790447                  0      0    0
##           mon origin         pst subject textplain transfer type version
## 1 0.006961574      0 0.009214534       0         0        0    0       0
## 2 0.015694098      0 0.018492612       0         0        0    0       0
## 3 0.010766589      0 0.013897826       0         0        0    0       0
##          can    forward        know        let       mail        may
## 1 0.01290466 0.01578176 0.009581387 0.00667164 1.02177664 0.01688953
## 2 0.02198514 0.02029258 0.018791177 0.01608176 0.02223241 0.02354321
## 3 0.03016062 0.02070291 0.016725686 0.01036419 0.01587717 0.03134343
##          non         pdt       sent       time       will       call
## 1 0.01619171 0.007741262 0.02250390 0.01558264 0.02865659 0.01681825
## 2 0.02575362 0.016280490 0.02028606 0.02105951 0.03238860 0.01943175
## 3 0.02626663 0.011725425 0.01252595 0.04602898 0.08384966 0.02800458
##          get      pleas privilegedpst     attach       thank         wed
## 1 0.01484758 0.01202248   0.008731443 0.01027022 0.007945565 0.007115383
## 2 0.02066327 0.02292720   0.014332937 0.01878569 0.020590454 0.016182118
## 3 0.01540430 0.02733310   0.013260019 0.01481573 0.014417924 0.012125234
##          item       mark       need         tue
## 1 0.008063836 0.02218634 0.01188369 0.007292413
## 2 0.015012615 0.02642339 0.01956066 0.016037263
## 3 0.013166129 0.02854876 0.02566744 0.010338654
```

```
length(enron_k$cluster)
```

```
## [1] 100000
```

Find Distinguishing terms of every cluster

```
require(tidyr)
```

```
## Loading required package: tidyr
```

```
enron_m_final <- cbind(enron_m, CLUSTER = enron_k$cluster)

head(enron_m_final)
```

```
##         ascii bcc         bit   charsetus content date   document encod
## 1 0.004553373   0 0.002520941 0.004560506       0    0 0.07238759     0
## 2 0.000000000   0 0.000000000 0.000000000       0    0 0.03277929     0
## 3 0.004751345   0 0.002630547 0.004758789       0    0 0.00000000     0
## 4 0.003525192   0 0.001951696 0.003530714       0    0 0.00000000     0
## 5 0.004047442   0 0.002240837 0.004053783       0    0 0.00000000     0
## 6 0.004047442   0 0.002240837 0.004053783       0    0 0.00000000     0
##        enron filenam folder  foldersal javamailevansthym messag mime
## 1 0.07037923       0      0 0.08415647                 0      0    0
## 2 0.28682856       0      0 0.00000000                 0      0    0
## 3 0.00000000       0      0 0.00000000                 0      0    0
## 4 0.00000000       0      0 0.00000000                 0      0    0
## 5 0.06255932       0      0 0.00000000                 0      0    0
## 6 0.00000000       0      0 0.00000000                 0      0    0
##          mon origin        pst subject textplain transfer type version
## 1 0.09561264      0 0.04195832       0         0        0    0       0
## 2 0.00000000      0 0.00000000       0         0        0    0       0
## 3 0.00000000      0 0.04378259       0         0        0    0       0
## 4 0.07402269      0 0.03248386       0         0        0    0       0
## 5 0.00000000      0 0.03729628       0         0        0    0       0
## 6 0.08498901      0 0.07459256       0         0        0    0       0
##          can    forward       know        let       mail        may
## 1 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 2 0.06639079 0.03451617 0.10827373 0.08405365 0.04057723 0.03521281
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.13874794 0.00000000
## 5 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 6 0.00000000 0.00000000 0.07084577 0.08249710 0.00000000 0.00000000
##          non       pdt       sent      time       will       call
## 1 0.00000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
## 2 0.03762827 0.0176807 0.03416058 0.1556653 0.17924679 0.00000000
## 3 0.00000000 0.0000000 0.00000000 0.0000000 0.00000000 0.09302683
## 4 0.25732880 0.0000000 0.05840357 0.0000000 0.00000000 0.00000000
## 5 0.00000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
## 6 0.00000000 0.0000000 0.00000000 0.0000000 0.04398185 0.00000000
##          get      pleas privilegedpst     attach      thank        wed
## 1 0.00000000 0.00000000     0.0000000 0.00000000 0.00000000 0.00000000
## 2 0.00000000 0.00000000     0.0000000 0.00000000 0.00000000 0.00000000
## 3 0.09386996 0.05071751     0.0000000 0.00000000 0.00000000 0.00000000
## 4 0.06964546 0.00000000     0.0700756 0.00000000 0.00000000 0.00000000
## 5 0.00000000 0.04320380     0.0000000 0.08330642 0.05313903 0.08210072
## 6 0.00000000 0.04320380     0.0000000 0.00000000 0.00000000 0.00000000
##         item mark need tue CLUSTER
## 1 0.00000000    0    0   0       2
## 2 0.00000000    0    0   0       3
## 3 0.00000000    0    0   0       2
## 4 0.00000000    0    0   0       2
## 5 0.00000000    0    0   0       2
## 6 0.08575211    0    0   0       2
```

```
enron_df_final <- as.data.frame(enron_m_final)

enron_df_final <- enron_df_final %>%
                        group_by(CLUSTER) %>%
                        summarise_all(funs(sum))

# Top Words by cluster

enron_df_final
```

```
## # A tibble: 3 × 46
##   CLUSTER       ascii   bcc         bit   charsetus content  date
##     <dbl>       <dbl> <dbl>       <dbl>       <dbl>   <dbl> <dbl>
## 1       1   0.8757296     0   0.5133693   0.8762043       0     0
## 2       2 316.0105060     0 182.3492023 316.3388386       0     0
## 3       3  16.3939355     0   9.8996547  16.4053880       0     0
## # ... with 39 more variables: document <dbl>, encod <dbl>, enron <dbl>,
## #   filenam <dbl>, folder <dbl>, foldersal <dbl>, javamailevansthym <dbl>,
## #   messag <dbl>, mime <dbl>, mon <dbl>, origin <dbl>, pst <dbl>,
## #   subject <dbl>, textplain <dbl>, transfer <dbl>, type <dbl>,
## #   version <dbl>, can <dbl>, forward <dbl>, know <dbl>, let <dbl>,
## #   mail <dbl>, may <dbl>, non <dbl>, pdt <dbl>, sent <dbl>, time <dbl>,
## #   will <dbl>, call <dbl>, get <dbl>, pleas <dbl>, privilegedpst <dbl>,
## #   attach <dbl>, thank <dbl>, wed <dbl>, item <dbl>, mark <dbl>,
## #   need <dbl>, tue <dbl>
```

```
# Top 10 words in cluster 1

Cluster_1 <- enron_df_final %>%
                  filter(CLUSTER == 1) %>%
                  select(-CLUSTER) %>%
                  t() %>%
                  as.data.frame()


Cluster_1 <- Cluster_1 %>%
                  `colnames<-`(c("Top_Freq")) %>%
                  mutate(Words = rownames(Cluster_1)) %>%
                  arrange(desc(Top_Freq)) %>%
                  head(10)

Cluster_1
```

```
##      Top_Freq   Words
## 1  617.153091    mail
## 2   17.308581    will
## 3   13.592359    sent
## 4   13.400547    mark
## 5   10.201275     may
## 6   10.158223    call
## 7    9.779795     non
## 8    9.532182 forward
## 9    9.411914    time
## 10   8.967940     get
```

```
# Top 10 words in cluster 2

Cluster_2 <- enron_df_final %>%
                  filter(CLUSTER == 2) %>%
                  select(-CLUSTER) %>%
                  t() %>%
                  as.data.frame()

Cluster_2 <-  Cluster_2 %>%
                  `colnames<-`(c("Top_Freq")) %>%
                  mutate(Words = rownames(Cluster_2)) %>%
                  arrange(desc(Top_Freq)) %>%
                  head(10)

Cluster_2
```

```
##    Top_Freq Words
## 1  2971.104  will
## 2  2423.897  mark
## 3  2362.457   non
## 4  2159.690   may
## 5  2103.180 pleas
## 6  2039.446  mail
## 7  2016.763   can
## 8  1931.852  time
## 9  1895.504   get
## 10 1888.824 thank
```

```
# Top 10 words in cluster 3

Cluster_3 <- enron_df_final %>%
                  filter(CLUSTER == 2) %>%
                  select(-CLUSTER) %>%
                  t() %>%
                  as.data.frame()

Cluster_3 <-  Cluster_3 %>%
                  `colnames<-`(c("Top_Freq")) %>%
                  mutate(Words = rownames(Cluster_3)) %>%
                  arrange(desc(Top_Freq)) %>%
                  head(10)

Cluster_3
```

```
##    Top_Freq Words
## 1  2971.104  will
## 2  2423.897  mark
## 3  2362.457   non
## 4  2159.690   may
## 5  2103.180 pleas
## 6  2039.446  mail
## 7  2016.763   can
## 8  1931.852  time
## 9  1895.504   get
## 10 1888.824 thank
```

All in all the analysis was frusterating in that my local memory, even when using paralleliztion, could not chew on a large majority of the data. Given the assignment, I accomplished what was needed, but in regard to significant insight I was largely held back due to computational expense. Nevertheless, learning how those work arounds were actually super helpful in getting me to grasp the data, the goal and the tools I had available to accomplish this task. Also, text analytics is night and day easier to acocmplish in R vs. Python.