pset6_ParthDesai

Parth Desai

2023-04-10

Question 1

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
emails <- read.csv('Emails.csv', stringsAsFactors = FALSE)
```

Part 1.1

```
colnames(emails)
```

```
##  [1] "Id"                           "DocNumber"
##  [3] "MetadataSubject"              "MetadataTo"
##  [5] "MetadataFrom"                 "SenderPersonId"
##  [7] "MetadataDateSent"             "MetadataDateReleased"
##  [9] "MetadataPdfLink"              "MetadataCaseNumber"
## [11] "MetadataDocumentClass"        "ExtractedSubject"
## [13] "ExtractedTo"                  "ExtractedFrom"
## [15] "ExtractedCc"                  "ExtractedDateSent"
## [17] "ExtractedCaseNumber"          "ExtractedDocNumber"
## [19] "ExtractedDateReleased"        "ExtractedReleaseInPartOrFull"
## [21] "ExtractedBodyText"            "RawText"
```

Column 22 has the raw text

Part 1.2

```
email_1 <- str_replace_all(emails[1,22], '[^[:alnum:]]+', ' ')
email_1 <- str_replace_all(email_1, '\\s+', ' ')
```

Part 1.3

```
email_vector <- strsplit(email_1, ' ')
```

Part 1.4

```
length(email_vector[[1]])
```

```
## [1] 143
```

Question 2

Part 2.1

```
benghazi_mention <- c()
benghzi_count <- str_which(emails[,22], fixed('benghazi', ignore_case = TRUE))
for (i in 1:nrow(emails)) {
  trial <- as.vector(strsplit(emails[benghzi_count[i],22], ' '))
  benghazi_mention[i] <- length(trial[[1]][str_which(trial[[1]], fixed('benghazi', ignore_case = TRUE))]
}

head(benghazi_mention, n = 5)
```

```
## [1]  2  8  2  4 10
```

```
tail(benghzi_count, n = 5)
```

```
## [1] 292 293 294 295 296
```

Part 2.2

```
benghazi_cleaned <- c()
benghazi_output <- c()
for (i in 1:length(benghzi_count)) {
  benghazi_cleaned[i] <- str_replace_all(emails[benghzi_count[i], 22], '[^[:alnum:]\\s]+', '')
  benghazi_cleaned[i] <- str_replace_all(benghazi_cleaned[i], '\\s+', ' ')
  benghazi_cleaned[i] <- tolower(benghazi_cleaned[i])
}

benghazi_regex <- "\\b(\\w+\\s+\\w+\\s+)?benghazi(\\s+\\w+\\s+\\w+)?\\b"
benghazi_matches <- regmatches(benghazi_cleaned, gregexpr(benghazi_regex, benghazi_cleaned))
benghazi_output <- lapply(benghazi_matches, function(matches) unlist(matches))


benghazi_output[[2]]
```

```
## [1] "house select benghazi comm subject"
## [2] "gathering around benghazi qaddafi is"
## [3] "officers to benghazi to assist"
## [4] "house select benghazi comm subject"
## [5] "house select benghazi comm subject"
## [6] "house select benghazi comm subject"
## [7] "house select benghazi comm subject"
## [8] "house select benghazi comm subject"
```

```
benghazi_output[length(benghzi_count)]
```

```
## [[1]]
## [1] "house select benghazi comm subject" "house select benghazi comm subject"
```

Part 2.3

Benghazi is mentioned when discussing a course of action the House of Representatives will take.

Question 3

```
pos_words <- read.delim("positive-words.txt", header = F, stringsAsFactors = F)[,1]
neg_words <- read.delim("negative-words.txt", header = F, stringsAsFactors = F)[,1]
```

Part 3.1

```
email_clean <- c()
clean_split <- c()
pos_count <- c()
neg_count <- c()
for (i in 1:nrow(emails)) {
  email_clean[i] <- str_replace_all(emails[i, 22], '[[:punct:]]', ' ')
  email_clean[i] <- str_replace_all(email_clean[i], '\\s+', ' ')
  email_clean[i] <- tolower(email_clean[i])
  clean_split <- strsplit(email_clean[i], ' ')[[1]]
  pos_count[i] <- sum(clean_split %in% pos_words)
  neg_count[i] <- sum(clean_split %in% neg_words)
}

head(pos_count, n = 5)
```

```
## [1]  7 32  2  4 33
```

```
tail(neg_count, n = 5)
```

```
## [1]  8  0 45  0  0
```

Part 3.2

```
sent_frame <- data.frame('Benghazi' = benghazi_mention, 'Positive' = pos_count, 'Negative' = neg_count)
ratio <- c()

for (i in 1:nrow(sent_frame)) {
  if((pos_count[i] == 0) && (neg_count[i] == 0)){
    ratio[i] <- 0.5
  }
  else{
    ratio[i] <- ((pos_count[i])/(pos_count[i] + neg_count[i]))
  }
}

regress <- lm(ratio ~ Benghazi, data = sent_frame)
summary(regress)
```

```
##
## Call:
## lm(formula = ratio ~ Benghazi, data = sent_frame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61400 -0.11400 -0.08871  0.24314  0.38600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6139990  0.0031914 192.394   <2e-16 ***
## Benghazi    0.0002937  0.0013784   0.213    0.831
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.282 on 7943 degrees of freedom
## Multiple R-squared:  5.717e-06,  Adjusted R-squared:  -0.0001202
## F-statistic: 0.04541 on 1 and 7943 DF,  p-value: 0.8313
```

Question 4

Part 4.1

```
load('DTM.RData')
row_lengths = apply(dtm_use, 1, function(z) sqrt(sum(z^2)))
dtm_norm = dtm_use/row_lengths
```

Part 4.2

```
K <- 3
set.seed(12345)
K3_norm <- kmeans(dtm_norm, centers = K)
prop.table(table(K3_norm$cluster))
```

```
##
##         1         2         3
## 0.7603640 0.1021234 0.1375126
```

Part 4.3

```
K2 <- 6
set.seed(12345)
K6_og <- kmeans(dtm_use, centers = K2)
set.seed(12345)
K6_og_norm <- kmeans(dtm_norm, centers = K2, nstart = 3)
prop.table(table(K6_og$cluster))
```

```
##
##           1           2           3           4           5           6
## 0.066734075 0.040444894 0.027300303 0.739130435 0.123356926 0.003033367
```

```
prop.table(table(K6_og_norm$cluster))
```

```
##
##          1          2          3          4          5          6
## 0.60869565 0.10313448 0.06774520 0.03437816 0.05561173 0.13043478
```

```
top_words_unnorm = lapply(1:6, function(i) {
  cluster <- K6_og$cluster == i
  words <- colnames(dtm_use)[cluster]
  freq <- rowSums(dtm_use[,cluster])
  mean_freq <- mean(freq)
  top_freq_words <- head(sort(freq, decreasing = TRUE), 10)
  top_diff_words <- head(sort(freq - mean_freq, decreasing = TRUE), 10)
  list(unorm_top_freq_words = top_freq_words, unorm_top_diff_words = top_diff_words)
})

top_words_norm = lapply(1:6, function(i) {
  cluster <- K6_og_norm$cluster == i
  words <- colnames(dtm_use)[cluster]
  freq <- rowSums(dtm_use[,cluster])
  mean_freq <- mean(freq)
  top_freq_words <- head(sort(freq, decreasing = TRUE), 10)
  top_diff_words <- head(sort(freq - mean_freq, decreasing = TRUE), 10)
  list(norm_top_freq_words = top_freq_words, norm_top_diff_words = top_diff_words)
})

top_words <- data.frame(top_words_unnorm, top_words_norm)
top_words
```

```
##    unorm_top_freq_words unorm_top_diff_words unorm_top_freq_words.1
## 1                     4             3.419616                      3
## 2                     4             3.419616                      2
## 3                     4             3.419616                      2
## 4                     4             3.419616                      2
## 5                     4             3.419616                      2
## 6                     4             3.419616                      2
## 7                     4             3.419616                      2
## 8                     4             3.419616                      2
## 9                     3             2.419616                      2
## 10                    3             2.419616                      2
##    unorm_top_diff_words.1 unorm_top_freq_words.2 unorm_top_diff_words.2
## 1                 2.77452                      3               2.789687
## 2                 1.77452                      2               1.789687
## 3                 1.77452                      2               1.789687
## 4                 1.77452                      2               1.789687
## 5                 1.77452                      2               1.789687
## 6                 1.77452                      2               1.789687
## 7                 1.77452                      2               1.789687
## 8                 1.77452                      2               1.789687
## 9                 1.77452                      2               1.789687
## 10                1.77452                      2               1.789687
##    unorm_top_freq_words.3 unorm_top_diff_words.3 unorm_top_freq_words.4
## 1                      14                8.39636                      5
## 2                      14                8.39636                      5
## 3                      14                8.39636                      4
## 4                      14                8.39636                      4
## 5                      13                7.39636                      4
## 6                      13                7.39636                      4
## 7                      13                7.39636                      4
## 8                      13                7.39636                      4
## 9                      13                7.39636                      4
## 10                     12                6.39636                      3
##    unorm_top_diff_words.4 unorm_top_freq_words.5 unorm_top_diff_words.5
## 1                4.152679                      1              0.9888777
## 2                4.152679                      1              0.9888777
## 3                3.152679                      1              0.9888777
## 4                3.152679                      1              0.9888777
## 5                3.152679                      1              0.9888777
## 6                3.152679                      1              0.9888777
## 7                3.152679                      1              0.9888777
## 8                3.152679                      1              0.9888777
## 9                3.152679                      1              0.9888777
## 10               2.152679                      1              0.9888777
##    norm_top_freq_words norm_top_diff_words norm_top_freq_words.1
## 1                   13            8.320526                     4
## 2                   13            8.320526                     4
## 3                   12            7.320526                     4
## 4                   12            7.320526                     4
## 5                   11            6.320526                     4
## 6                   11            6.320526                     3
## 7                   11            6.320526                     3
## 8                   11            6.320526                     3
## 9                   11            6.320526                     3
## 10                  11           6.320526                   3
##    norm_top_diff_words.1 norm_top_freq_words.2 norm_top_diff_words.2
## 1               3.331648                     5              4.518706
## 2               3.331648                     4              3.518706
## 3               3.331648                     4              3.518706
## 4               3.331648                     4              3.518706
## 5               3.331648                     3              2.518706
## 6               2.331648                     3              2.518706
## 7               2.331648                     3              2.518706
## 8               2.331648                     3              2.518706
## 9               2.331648                     3              2.518706
## 10              2.331648                     3              2.518706
##    norm_top_freq_words.3 norm_top_diff_words.3 norm_top_freq_words.4
## 1                      3              2.787664                     4
## 2                      3              2.787664                     3
## 3                      2              1.787664                     3
## 4                      2              1.787664                     3
## 5                      2              1.787664                     3
## 6                      2              1.787664                     3
## 7                      2              1.787664                     3
## 8                      2              1.787664                     3
## 9                      2              1.787664                     3
## 10                     2              1.787664                     2
##    norm_top_diff_words.4 norm_top_freq_words.5 norm_top_diff_words.5
## 1               3.534884                     5              4.028311
## 2               2.534884                     5              4.028311
## 3               2.534884                     4              3.028311
## 4               2.534884                     4              3.028311
## 5               2.534884                     4              3.028311
## 6               2.534884                     4              3.028311
## 7               2.534884                     4              3.028311
## 8               2.534884                     4              3.028311
## 9               2.534884                     4              3.028311
## 10              1.534884                     4              3.028311
```

Part 4.4

Each cluster captures the most frequent and unique word choice of a given email compared to its 3 or 6 closest neighbors. I think the normalized document term matrix is more meaningful as the document length is held more constantly. This gives an equal length which to compare all documents that is not provided by the original document term matrix.