Text-as-Data Exercise Solutions

Alicia R. Chen

5/28/2021

```
packages <- c("tidyverse","data.table", "lubridate", 'ggplot2', "quanteda", "RColorBrewer")
lapply(packages, library, character.only = TRUE)
```

```
## [[1]]
##  [1] "forcats"   "stringr"   "dplyr"     "purrr"     "readr"     "tidyr"
##  [7] "tibble"    "ggplot2"   "tidyverse" "stats"     "graphics"  "grDevices"
## [13] "utils"     "datasets"  "methods"   "base"
##
## [[2]]
##  [1] "data.table" "forcats"    "stringr"    "dplyr"      "purrr"
##  [6] "readr"      "tidyr"      "tibble"     "ggplot2"    "tidyverse"
## [11] "stats"      "graphics"   "grDevices"  "utils"      "datasets"
## [16] "methods"    "base"
##
## [[3]]
##  [1] "lubridate"  "data.table" "forcats"    "stringr"    "dplyr"
##  [6] "purrr"      "readr"      "tidyr"      "tibble"     "ggplot2"
## [11] "tidyverse"  "stats"      "graphics"   "grDevices"  "utils"
## [16] "datasets"   "methods"    "base"
##
## [[4]]
##  [1] "lubridate"  "data.table" "forcats"    "stringr"    "dplyr"
##  [6] "purrr"      "readr"      "tidyr"      "tibble"     "ggplot2"
## [11] "tidyverse"  "stats"      "graphics"   "grDevices"  "utils"
## [16] "datasets"   "methods"    "base"
##
## [[5]]
##  [1] "quanteda"   "lubridate"  "data.table" "forcats"    "stringr"
##  [6] "dplyr"      "purrr"      "readr"      "tidyr"      "tibble"
## [11] "ggplot2"    "tidyverse"  "stats"      "graphics"   "grDevices"
## [16] "utils"      "datasets"   "methods"    "base"
##
## [[6]]
##  [1] "RColorBrewer" "quanteda"     "lubridate"    "data.table"   "forcats"
##  [6] "stringr"      "dplyr"        "purrr"        "readr"        "tidyr"
## [11] "tibble"       "ggplot2"      "tidyverse"    "stats"        "graphics"
## [16] "grDevices"    "utils"        "datasets"     "methods"      "base"
```

Q1

```
df <- fread("./ira_tweets_csv_hashed.csv", fill=TRUE)
```

```r
cat("Total tweets:", nrow(df))
```

```
## Total tweets: 1826345
```

```r
cat("English tweets:", nrow(df[df$tweet_language=="en",]), "Non-English tweets:", nrow(df[df$tweet_langu
```

```
## English tweets: 596227 Non-English tweets: 1230118
```

```r
cat("Self-reported locations:", nrow(df[df$user_reported_location != ""]))
```

```
## Self-reported locations: 1503154
```

```r
cat("Contains BLM keywords:", sum(str_detect(df$tweet_text, regex('Black Lives Matter|BLM', ignore_case
```

```
## Contains BLM keywords: 1578
```

```r
cat("Mentions Sputnik or RT:", sum(str_detect(df$tweet_text, "@SputnikInt|@RT_com")))
```

```
## Mentions Sputnik or RT: 942
```

Q2 Creating DTM

One thing to consider when working with Twitter data is removing emojis, links, mentions etc. (depending on your use case). Also removing the RT: text which just shows that it's a retweet and is not useful to include

```r
# only English tweets
df <- df[df$tweet_language=="en",]
df$Date <- as.Date(df$tweet_time)

# function to remove capitalization, punctuation, and special characters and numbers & apply Porter Ste
tweetclean <- function(i){
  text <- i
  text <- gsub("RT", "", text)
  # cleaning
  text <- tolower(text) #convert to lowercase

  text <- gsub("https\\S*|http\\S*", "", text) #remove URLs
  text <- gsub("pic.twitter\\S*", "", text) #remove picture links
  text <- gsub("[^\x01-\x7F]", " ", text) #remove emojis
  #text <- gsub("\\b\\d+\\b", "", text) #remove standalone numbers
  text <- gsub("[[:digit:]]+", "", text) #remove all numbers
  text <- gsub("@\\w+ *", "", text) #remove mentions
  text <- gsub ("#\\w+ *", "", text) # remove hashtags
  text <- gsub("&amp", "", text) # remove & from html
  text <- gsub("[[:punct:] ]+", " ", text) # remove all punctuation

  # Porter stemmer
  text <- SnowballC::wordStem(text)

  text
}
df$tweet_text_cleaned <- tweetclean(df$tweet_text)

# remove stopwords
stop_words <- readLines("http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/a11-smart-stop-lis
corpus <- corpus(df$tweet_text_clean, docnames = df$tweetid)
```

```
dtm <- dfm(corpus, remove=stop_words,verbose=TRUE)
```

```
head(dtm)
```

```
## Document-feature matrix of: 6 documents, 95,099 features (>99.99% sparse) and 0 docvars.
##                     features
## docs                 sun cloud give moon shadow rhythm world makes apparent
##   567357519547207680   1     1    1    1      1      1     1     1        1
##   493894187079974912   0     0    0    0      0      0     0     0        0
##   493688319902220288   0     0    0    0      0      0     0     0        0
##   497543470211678209   0     0    0    0      0      0     0     0        0
##   500956712657223680   0     0    0    0      0      0     0     0        0
##   548763776267218944   0     0    0    0      0      0     0     0        0
##                     features
## docs                 faith
##   567357519547207680     1
##   493894187079974912     0
##   493688319902220288     0
##   497543470211678209     0
##   500956712657223680     0
##   548763776267218944     0
## [ reached max_nfeat ... 95,089 more features ]
```

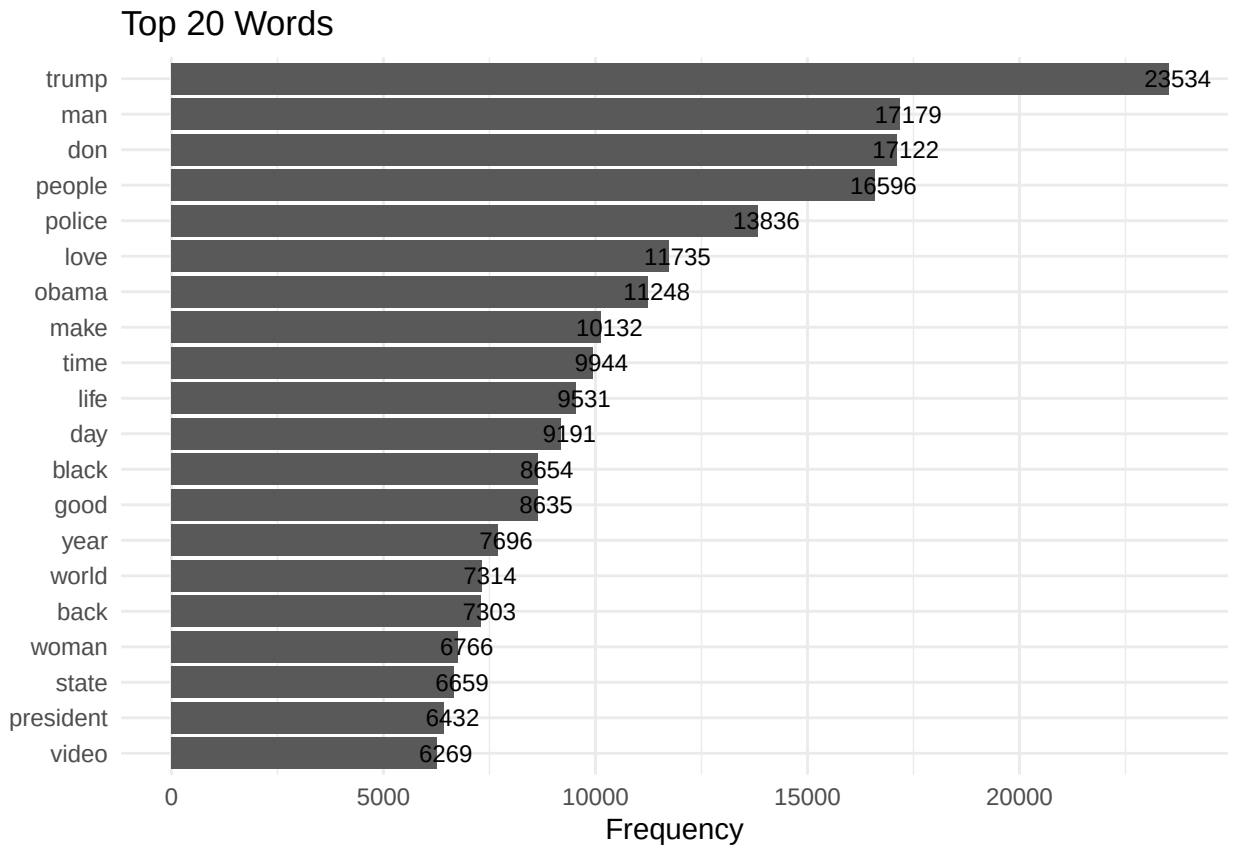
```
cat("Number of words:", ncol(dtm))
```

```
## Number of words: 95099
```

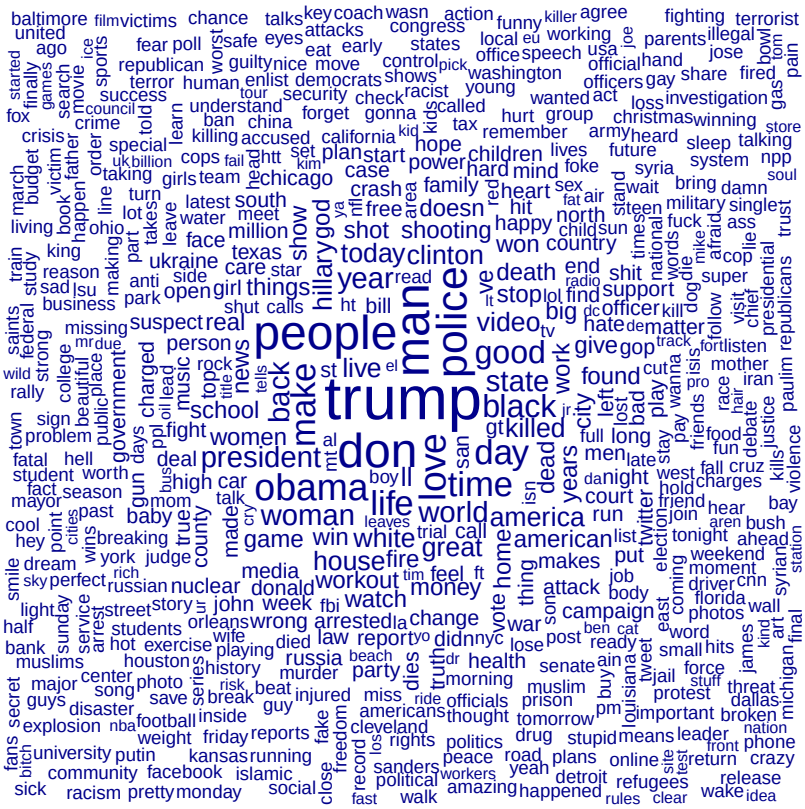
```
# Top words
colSums(dtm)[order(colSums(dtm),decreasing=TRUE)[1:20]]
```

```
##     trump       man       don    people    police      love     obama      make
##     23534     17179     17122     16596     13836     11735     11248     10132
##      time      life       day     black      good      year     world      back
##      9944      9531      9191      8654      8635      7696      7314      7303
##     woman     state president     video
##      6766      6659      6432      6269
```

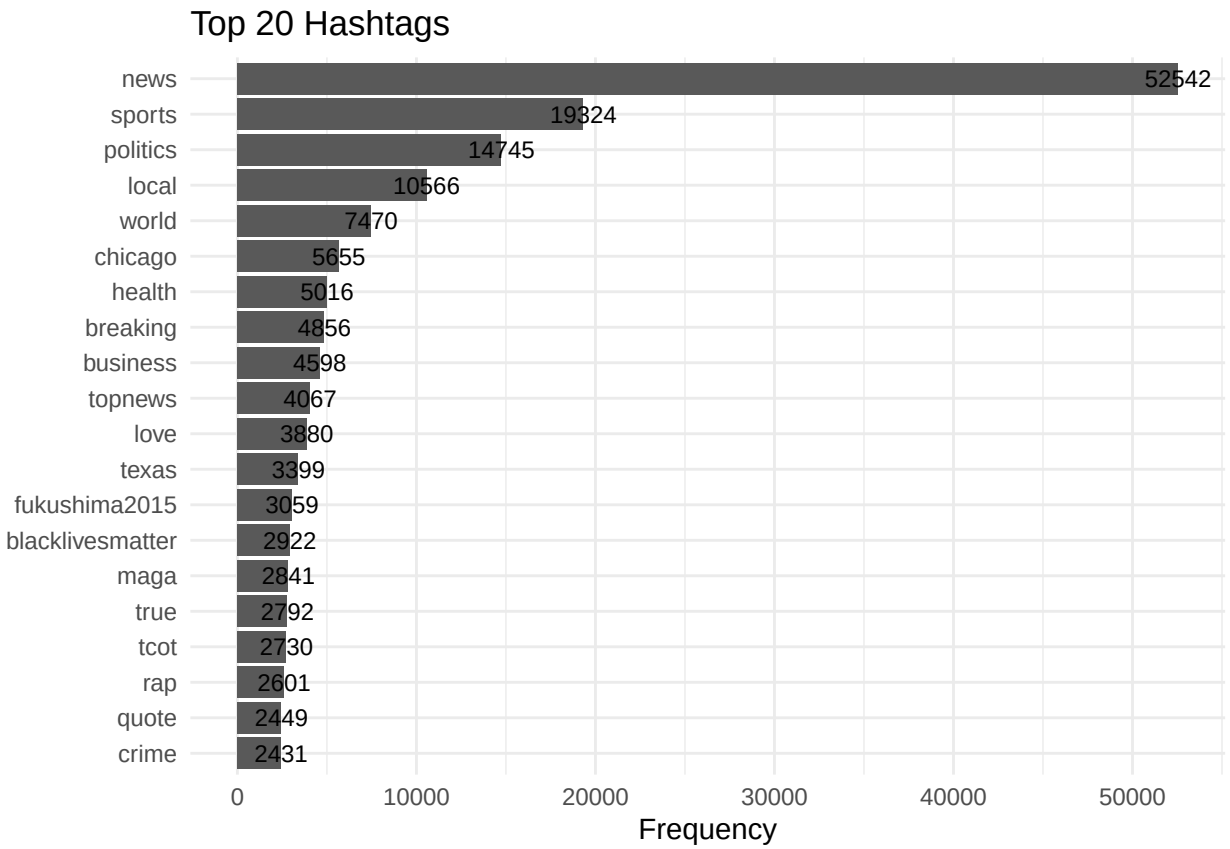
```
topwords <- quanteda.textstats::textstat_frequency(dtm)
topwords[1:20] %>%
  ggplot(aes(x=reorder(feature,frequency), y=frequency, label=frequency)) +
  labs(title = "Top 20 Words") +
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Frequency") +
  theme_minimal() +
  geom_text(nudge_y=200, size = 3)
```



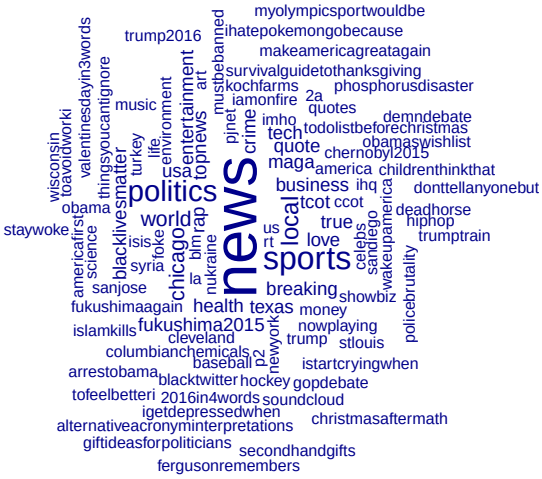

```
set.seed(pi)
quanteda.textplots::textplot_wordcloud(dtm, rotation=0.25, min_size=.75, max_size=3,max_words=1000,)
```




```
#create dfm
tags_corpus <- corpus(df$hashtags)
tags_dtm <- dfm(tags_corpus, remove = c("[", "]", ",", "'", "#", " "))

#plot hashtag frequency
toptags<- quanteda.textstats::textstat_frequency(tags_dtm)
toptags[1:20] %>%
  ggplot(aes(x=reorder(feature,frequency), y=frequency, label=frequency)) +
  labs(title = "Top 20 Hashtags") +
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Frequency") +
  theme_minimal() +
  geom_text(nudge_y=5, size = 3)
```




```
#word cloud
quanteda.textplots::textplot_wordcloud(tags_dtm, rotation = 0.25, min_size=.75, max_size=3, max_words=1
```



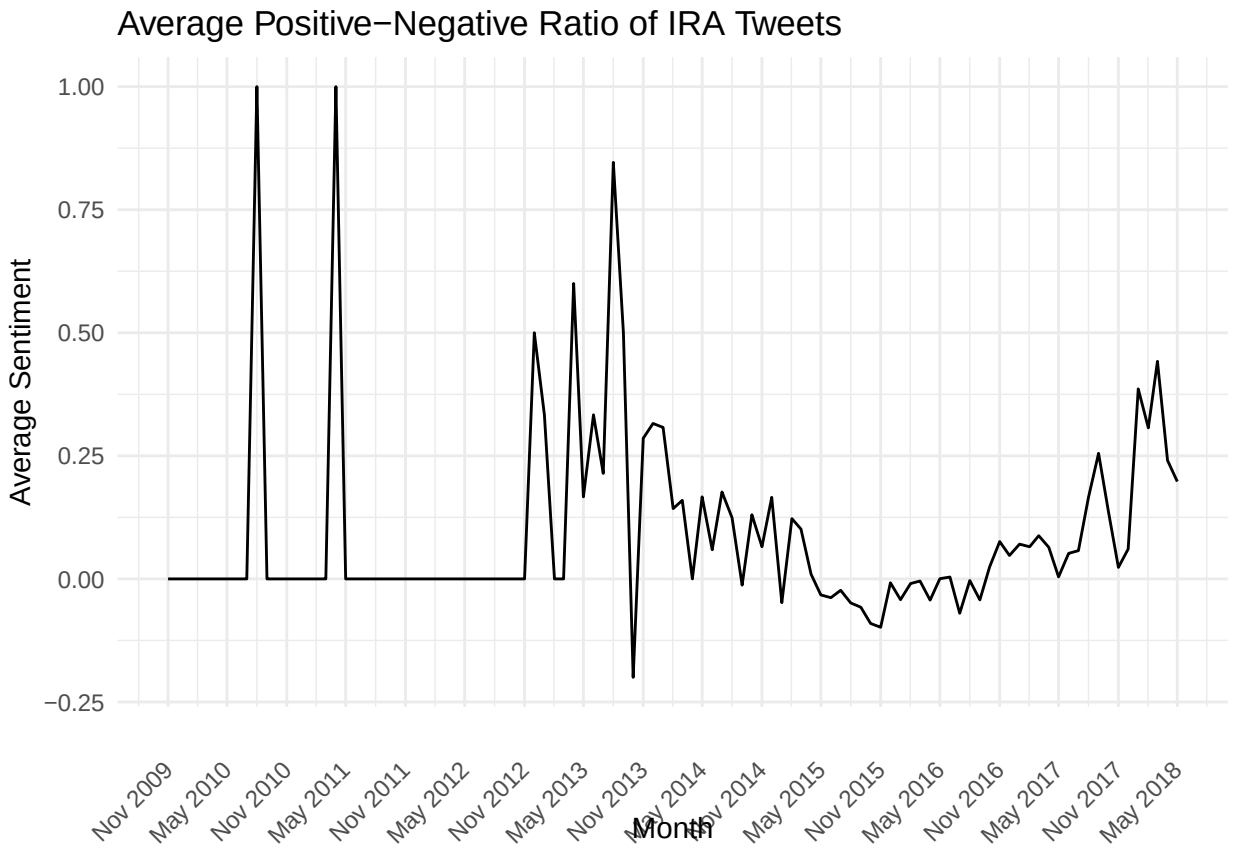
Dictionary Methods

```
pos_words = readLines('https://raw.githubusercontent.com/nealcaren/quant-text-fall-2014/master/positive
neg_words = readLines('https://raw.githubusercontent.com/nealcaren/quant-text-fall-2014/master/negative

df$positive <- rowSums(dtm[,which(featnames(dtm) %in% pos_words)])
df$negative <- rowSums(dtm[,which(featnames(dtm) %in% neg_words)])
df$pos_neg_ratio <- df$positive - df$negative
```

```
summary(df[,positive:pos_neg_ratio])
```

```
##     positive          negative       pos_neg_ratio
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :-14.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: -1.0000
##  Median : 0.0000   Median : 0.0000   Median :  0.0000
##  Mean   : 0.4963   Mean   : 0.4698   Mean   :  0.0265
##  3rd Qu.: 1.0000   3rd Qu.: 1.0000   3rd Qu.:  1.0000
##  Max.   :11.0000   Max.   :15.0000   Max.   : 11.0000
```

```
byMonth <- df %>%
  group_by(month = as.Date(cut(Date, "month"))) %>%
  summarise(sentiment = mean(pos_neg_ratio))
# Balance!!!
all_months <- data.frame(month = seq(as.Date(cut(min(df$Date),"month")),as.Date(cut(max(df$Date),"month
byMonth <- right_join(byMonth,all_months)
byMonth[is.na(byMonth)] <- 0
byMonth <- byMonth[order(byMonth$month),]

ggplot(byMonth, aes(x=month,y=sentiment,group=1)) +
  geom_line(stat='identity') +
  scale_x_date(date_breaks = "6 month", date_labels = "%b %Y") +
  theme_minimal() +
  xlab("Month") +
  ylab("Average Sentiment") +
  ggtitle("Average Positive-Negative Ratio of IRA Tweets") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



Topic Modeling

```
# Save only top 500 words
dtm <- dtm[,which(featnames(dtm) %in% topwords$feature[1:300])]

# Are there any tweets with no words?
dtm <- dtm[rowSums(dtm)>0,]
```

```
# number of topics
K <- 10

# Run LDA
topicModel <- topicmodels::LDA(dtm, K, control=list(seed = pi))

# Top 10 words
topicmodels::terms(topicModel, 10)
```

```
##       Topic 1  Topic 2     Topic 3   Topic 4   Topic 5  Topic 6  Topic 7
##  [1,] "trump"  "trump"     "don"     "obama"   "police" "black"  "ll"
##  [2,] "man"    "man"       "people"  "black"   "man"    "game"   "time"
##  [3,] "people" "live"      "big"     "clinton" "day"    "love"   "great"
##  [4,] "make"   "president" "mind"    "trump"   "gt"     "people" "state"
##  [5,] "home"   "bill"      "trump"   "workout" "city"   "man"    "make"
##  [6,] "police" "woman"     "things"  "white"   "people" "day"    "life"
##  [7,] "plan"   "great"     "school"  "dies"    "open"   "make"   "things"
##  [8,] "life"   "shooting"  "shot"    "video"   "love"   "win"    "good"
##  [9,] "video"  "women"     "playing" "good"    "time"   "don"    "obama"
## [10,] "good"   "arrested"  "san"     "hillary" "world"  "world"  "man"
##       Topic 8   Topic 9   Topic 10
##  [1,] "trump"   "back"    "don"
##  [2,] "state"   "people"  "police"
##  [3,] "donald"  "killed"  "make"
##  [4,] "love"    "don"     "life"
##  [5,] "man"     "woman"   "love"
##  [6,] "clinton" "law"     "call"
##  [7,] "good"    "won"     "time"
##  [8,] "hillary" "country" "leave"
##  [9,] "people"  "trump"   "woman"
## [10,] "ve"      "love"    "give"
```