Installing libraries

library(knitr)

> library(tidyr)

> library(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

> library(readr)

> library(ggplot2)

> library(tibble)

> library(stringr)

> library(gridExtra)

Attaching package: ‘gridExtra’

The following object is masked from ‘package:dplyr’:

combine

> library(scales)

Attaching package: ‘scales’

The following object is masked from ‘package:readr’:

col\_factor

> library(lubridate)

Attaching package: ‘lubridate’

The following objects are masked from ‘package:base’:

date, intersect, setdiff, union

> library(ggrepel)

> library(reshape2)

Attaching package: ‘reshape2’

The following object is masked from ‘package:tidyr’:

smiths

> library(kableExtra)

Attaching package: ‘kableExtra’

The following object is masked from ‘package:dplyr’:

group\_rows

> library(tm)

Loading required package: NLP

Attaching package: ‘NLP’

The following object is masked from ‘package:ggplot2’:

annotate

> library(wordcloud)

Loading required package: RColorBrewer

> library(tidytext)

> library(broom)

> library(topicmodels)

Loading and viewing data

> tweets <- read.csv("~/Chong IT-534/project/tweets.csv/tweets.csv", comment.char="#")

> view(tweets)

Size and structure of data

|  |
| --- |
| tweets$time <- ymd\_hms(tweets$time)  > glimpse(tweets)  Rows: 6,444  Columns: 28  $ id *<dbl>* 7.809256e+17, 7.809162e+17, 7.809116e+17, 7.809070e+17, 7.808…  $ handle *<chr>* "HillaryClinton", "HillaryClinton", "HillaryClinton", "Hillar…  $ text *<chr>* "The question in this election: Who can put the plans into ac…  $ is\_retweet *<chr>* "False", "True", "True", "False", "False", "False", "", "Fals…  $ original\_author *<chr>* "", "timkaine", "POTUS", "", "", "", "", "", "", "mcuban", "d…  $ time *<dttm>* 2016-09-28 00:22:34, 2016-09-27 23:45:00, 2016-09-27 23:26:4…  $ in\_reply\_to\_screen\_name *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ in\_reply\_to\_status\_id *<dbl>* NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  $ in\_reply\_to\_user\_id *<dbl>* NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  $ is\_quote\_status *<chr>* "False", "False", "False", "False", "False", "False", "", "Fa…  $ lang *<chr>* "en", "en", "en", "en", "en", "en", "", "en", "en", "en", "en…  $ retweet\_count *<int>* 218, 2445, 7834, 916, 859, 2181, NA, 1833, 4132, 1087, 3347, …  $ favorite\_count *<int>* 651, 5308, 27234, 2542, 2882, 6172, NA, 4954, 11239, 2824, 76…  $ longitude *<dbl>* NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  $ latitude *<dbl>* NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  $ place\_id *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_full\_name *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_name *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_type *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_country\_code *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_country *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_contained\_within *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_attributes *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ place\_bounding\_box *<chr>* "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "…  $ source\_url *<chr>* "https://studio.twitter.com", "http://twitter.com", "https://…  $ truncated *<chr>* "False", "False", "False", "False", "False", "True", "", "Fal…  $ entities *<chr>* "{'media': [{'display\_url': 'pic.twitter.com/XreEY9OicG', 'ex…  $ extended\_entities *<chr>* "{'media': [{'display\_url': 'pic.twitter.com/XreEY9OicG', 'si… |
|  |
| |  | | --- | |  | |

Exploratory Data Analysis

Checking language other than English

> kable(tweets %>% group\_by(lang) %>% count() %>% rename(Language = lang, 'Number of Tweets' = n))

<table>

<thead>

<tr>

<th style="text-align:left;"> Language </th>

<th style="text-align:right;"> Number of Tweets </th>

</tr>

</thead>

<tbody>

<tr>

<td style="text-align:left;"> </td>

<td style="text-align:right;"> 453 </td>

</tr>

<tr>

<td style="text-align:left;"> da </td>

<td style="text-align:right;"> 3 </td>

</tr>

<tr>

<td style="text-align:left;"> en </td>

<td style="text-align:right;"> 5846 </td>

</tr>

<tr>

<td style="text-align:left;"> es </td>

<td style="text-align:right;"> 95 </td>

</tr>

<tr>

<td style="text-align:left;"> et </td>

<td style="text-align:right;"> 1 </td>

</tr>

<tr>

<td style="text-align:left;"> fi </td>

<td style="text-align:right;"> 1 </td>

</tr>

<tr>

<td style="text-align:left;"> fr </td>

<td style="text-align:right;"> 2 </td>

</tr>

<tr>

<td style="text-align:left;"> tl </td>

<td style="text-align:right;"> 2 </td>

</tr>

<tr>

<td style="text-align:left;"> und </td>

<td style="text-align:right;"> 41 </td>

</tr>

</tbody>

</table>

|  |
| --- |
| kable(head(tweets %>% filter(lang=="es" & original\_author=="") %>% select(lang, is\_retweet, handle, text) %>% rename(Language = lang),5), format="html")%>%  + kable\_styling() %>%  + column\_spec(1, bold = T, width = "2cm", border\_right = T) %>%  + column\_spec(2, bold = T, width = "2cm", border\_right = T) %>%  + column\_spec(3, bold = T, width = "2cm", border\_right = T) %>%  + column\_spec(4, width = "19cm") |
|  |
| |  | | --- | | > | |

A screenshot of a computer

Description automatically generated

The table below shows the numbers of non-Spanish tweets that Clinton and Trump have written themselves

> tweets$handle <- sub("realDonaldTrump", "Trump", tweets$handle)

> tweets$handle <- sub("HillaryClinton", "Clinton", tweets$handle)

> tweets$is\_retweet <- as.logical(tweets$is\_retweet)

>

> kable(tweets %>% filter(is\_retweet==FALSE) %>% group\_by(handle) %>% count())

<table>

<thead>

<tr>

<th style="text-align:left;"> handle </th>

<th style="text-align:right;"> n </th>

</tr>

</thead>

<tbody>

<tr>

<td style="text-align:left;"> Clinton </td>

<td style="text-align:right;"> 2465 </td>

</tr>

<tr>

<td style="text-align:left;"> Trump </td>

<td style="text-align:right;"> 2846 </td>

</tr>

</tbody>

</table>

displayed people that were retweeted at least 5 times, as in total 274 people were retweeted

> p1 <- tweets %>% filter(original\_author != "") %>% group\_by(original\_author) %>% count() %>% filter(n>=5) %>% arrange(desc(n)) %>% ungroup()

> # Create a bubble chart

> ggplot(p1, aes(x = reorder(original\_author, n), y = n, size = n)) +

+ geom\_point(color = "darkgreen", alpha = 0.7) +

+ coord\_flip() +

+ labs(x = "", y = "Number of tweets retweeted by either Trump or Clinton") +

+ theme(legend.position = "none")

> # Create a bubble chart

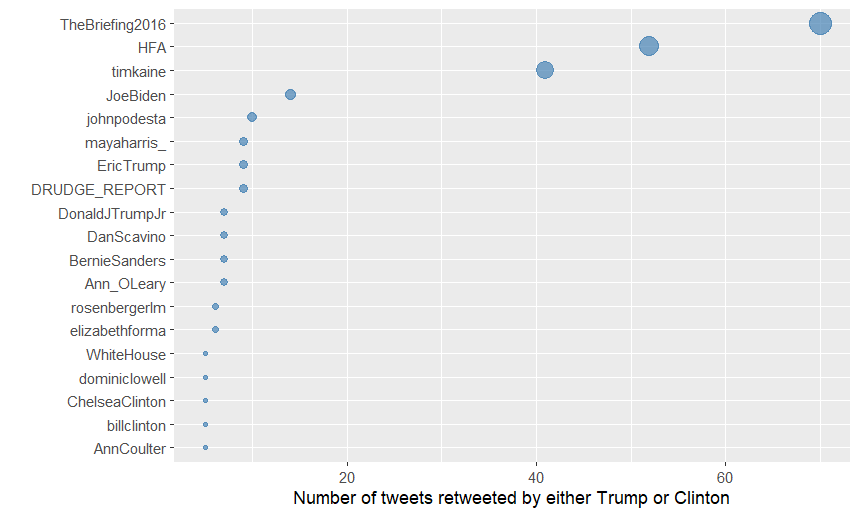
> ggplot(p1, aes(x = reorder(original\_author, n), y = n, size = n)) +

+ geom\_point(color = "steelblue", alpha = 0.7) +

+ coord\_flip() +

+ labs(x = "", y = "Number of tweets retweeted by either Trump or Clinton") +

+ theme(legend.position = "none")



finding the first 20 (non-Spanish) tweets.

> tweets$author <- ifelse(tweets$original\_author != "", tweets$original\_author, tweets$handle)

>

> kable(head(tweets %>% select(author, handle, text), 20), format = "html") %>%

+ kable\_styling() %>%

+ column\_spec(1, bold = T, width = "2cm", border\_right = T) %>%

+ column\_spec(2, bold = T, width = "2cm", border\_right = T) %>%

+ column\_spec(3, width = "19cm")

A screenshot of a computer

Description automatically generated

Removing unwanted characters

> tweets$text <- str\_replace\_all(tweets$text, "[\n]" , "") #remove new lines

> tweets$text <- str\_replace\_all(tweets$text, "&amp", "") # rm ampersand

>

> #URLs are always at the end and did not counts towards the 140 characters limit

> tweets$text <- str\_replace\_all(tweets$text, "http.\*" , "")

>

> tweets$text <- iconv(tweets$text, "latin1", "ASCII", sub="")

Creating a VCorpus object

separate corpusses for the Clinton and Trump Tweets. The first one that investigating is the “Trump corpus”.

> tweets <- tweets %>% rename (doc\_id = id)

> ClintonTweets <- tweets %>% filter(is\_retweet=="FALSE" & handle=="Clinton")

> TrumpTweets <- tweets %>% filter(is\_retweet=="FALSE" & handle=="Trump")

>

> TrumpCorpus <- DataframeSource(TrumpTweets)

> TrumpCorpus <- VCorpus(TrumpCorpus)

>

> ClintonCorpus <- DataframeSource(ClintonTweets)

> ClintonCorpus <- VCorpus(ClintonCorpus)

>

> TrumpCorpus

<<VCorpus>>

Metadata: corpus specific: 0, document level (indexed): 27

Content: documents: 2846

Cleaning with stopwords

> print(sort(stopwords("en")))

[1] "a" "about" "above" "after" "again" "against"

[7] "all" "am" "an" "and" "any" "are"

[13] "aren't" "as" "at" "be" "because" "been"

[19] "before" "being" "below" "between" "both" "but"

[25] "by" "can't" "cannot" "could" "couldn't" "did"

[31] "didn't" "do" "does" "doesn't" "doing" "don't"

[37] "down" "during" "each" "few" "for" "from"

[43] "further" "had" "hadn't" "has" "hasn't" "have"

[49] "haven't" "having" "he" "he'd" "he'll" "he's"

[55] "her" "here" "here's" "hers" "herself" "him"

[61] "himself" "his" "how" "how's" "i" "i'd"

[67] "i'll" "i'm" "i've" "if" "in" "into"

[73] "is" "isn't" "it" "it's" "its" "itself"

[79] "let's" "me" "more" "most" "mustn't" "my"

[85] "myself" "no" "nor" "not" "of" "off"

[91] "on" "once" "only" "or" "other" "ought"

[97] "our" "ours" "ourselves" "out" "over" "own"

[103] "same" "shan't" "she" "she'd" "she'll" "she's"

[109] "should" "shouldn't" "so" "some" "such" "than"

[115] "that" "that's" "the" "their" "theirs" "them"

[121] "themselves" "then" "there" "there's" "these" "they"

[127] "they'd" "they'll" "they're" "they've" "this" "those"

[133] "through" "to" "too" "under" "until" "up"

[139] "very" "was" "wasn't" "we" "we'd" "we'll"

[145] "we're" "we've" "were" "weren't" "what" "what's"

[151] "when" "when's" "where" "where's" "which" "while"

[157] "who" "who's" "whom" "why" "why's" "with"

[163] "won't" "would" "wouldn't" "you" "you'd" "you'll"

[169] "you're" "you've" "your" "yours" "yourself" "yourselves"

convert all characters into lower characters (no more capitals)

remove numbers

remove all English stopwords.

remove punctuation

strip whitespaces

> CleanCorpus <- function(x){

+ x <- tm\_map(x, content\_transformer(tolower))

+ x <- tm\_map(x, removeNumbers) #remove numbers before removing words. Otherwise "trump2016" leaves "trump"

+ x <- tm\_map(x, removeWords, tidytext::stop\_words$word)

+ x <- tm\_map(x, removePunctuation)

+ x <- tm\_map(x, stripWhitespace)

+ return(x)

+ }

>

> RemoveNames <- function(x) {

+ x <- tm\_map(x, removeWords, c("donald", "hillary", "clinton", "trump", "realdonaldtrump", "hillaryclinton"))

+ return(x)

+ }

>

> CreateTermsMatrix <- function(x) {

+ x <- TermDocumentMatrix(x)

+ x <- as.matrix(x)

+ y <- rowSums(x)

+ y <- sort(y, decreasing=TRUE)

+ return(y)

+ }

>

> TrumpCorpus <- CleanCorpus(TrumpCorpus)

> TermFreqTrump <- CreateTermsMatrix(TrumpCorpus)

>

> content(TrumpCorpus[[1]])

[1] "join pm rally tomorrow midamerica center council bluffs iowa tickets "

Most words trump used

> TrumpDF[1:20,] %>%

+ ggplot(aes(x = reorder(word, count), y = count, group = 1)) +

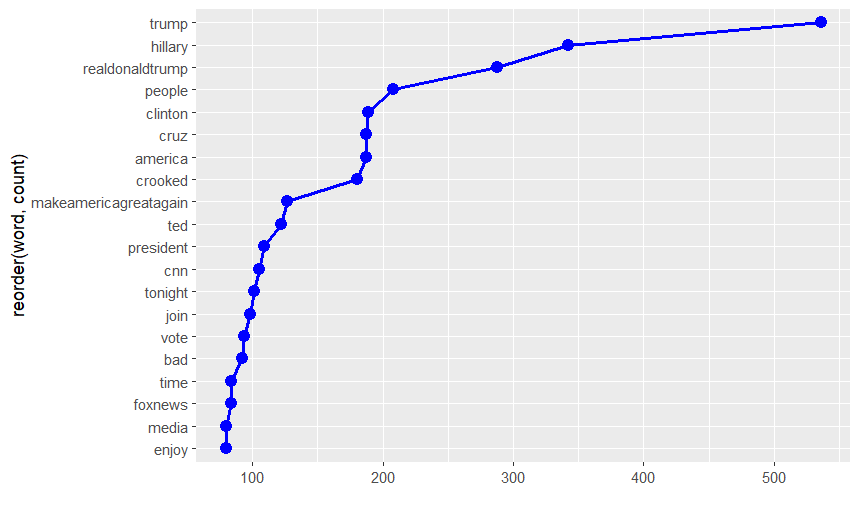
+ geom\_line(color = "blue", size = 1) +

+ geom\_point(color = "blue", size = 3) +

+ theme(legend.position = "none") +

+ labs(y = "") + # Label the y-axis

+ coord\_flip()



Most Words used by Clinton

> ClintonCorpus <- CleanCorpus(ClintonCorpus)

> TermFreqClinton <- CreateTermsMatrix(ClintonCorpus)

>

> ClintonDF <- data.frame(word=names(TermFreqClinton), count=TermFreqClinton)

>

> ClintonDF[1:20,] %>%

+ ggplot(aes(x = reorder(word, count), y = count, group = 1)) +

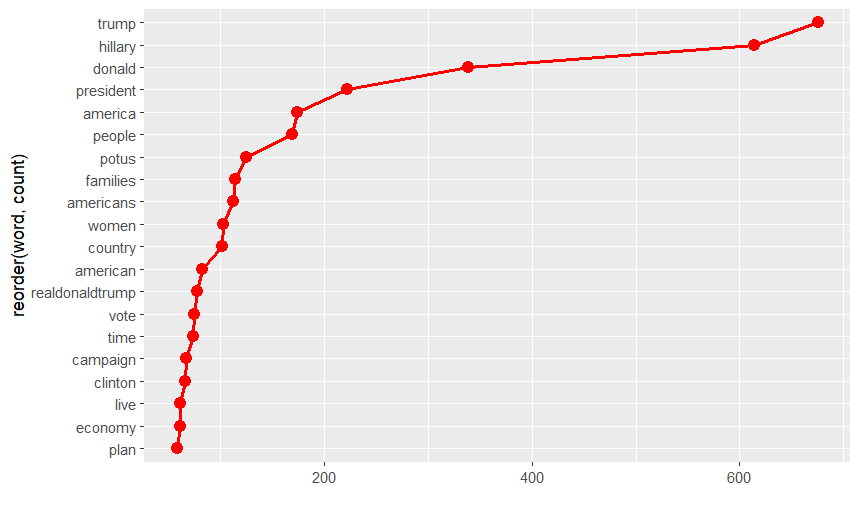
+ geom\_line(color = "red", size = 1) +

+ geom\_point(color = "red", size = 3) +

+ theme(legend.position = "none") +

+ labs(y = "") + # Label the y-axis

+ coord\_flip()



> set.seed(2018)

>

> TrumpCorpus1 <- RemoveNames(TrumpCorpus)

> TermFreqTrump <- CreateTermsMatrix(TrumpCorpus1)

> TrumpDF <- data.frame(word=names(TermFreqTrump), count=TermFreqTrump)

>

>

> wordcloud(TrumpDF$word, TrumpDF$count, max.words = 100, scale=c(2.5,.5), random.color = TRUE, colors=brewer.pal(9,"Set1"))

>

>

> ClintonCorpus1 <- RemoveNames(ClintonCorpus)

> TermFreqClinton <- CreateTermsMatrix(ClintonCorpus1)

> ClintonDF <- data.frame(word=names(TermFreqClinton), count=TermFreqClinton)

>

> wordcloud(ClintonDF$word, ClintonDF$count, max.words = 100, scale=c(2.5,.5), random.color = TRUE, colors=brewer.pal(9,"Set1"))

Most words comparison for both

> allClinton <- paste(ClintonTweets$text, collapse = " ")

> allTrump <- paste(TrumpTweets$text, collapse = " ")

> allClTr <- c(allClinton, allTrump)

>

> allClTr <- VectorSource(allClTr)

> allCorpus <- VCorpus(allClTr)

> allCorpus <- CleanCorpus(allCorpus)

> allCorpus <- RemoveNames(allCorpus)

>

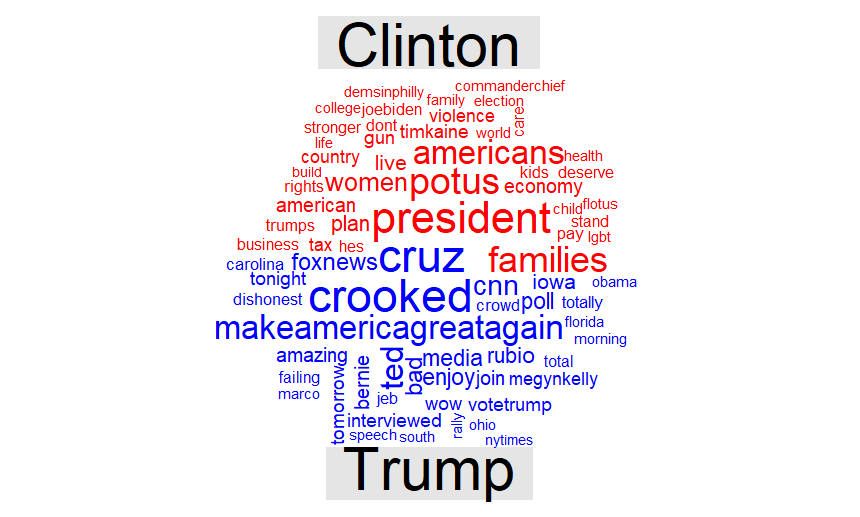
> TermsAll <- TermDocumentMatrix(allCorpus)

> colnames(TermsAll) <- c("Clinton", "Trump")

> MatrixAll <- as.matrix(TermsAll)

>

> comparison.cloud(MatrixAll, colors = c("red", "blue"), scale=c(2.3,.3), max.words = 75)



Sentiment analysis

The Bing lexicon (positive/negative, binary)

> get\_sentiments("bing")

# A tibble: 6,786 × 2

word sentiment

*<chr>* *<chr>*

1 2-faces negative

2 abnormal negative

3 abolish negative

4 abominable negative

5 abominably negative

6 abominate negative

7 abomination negative

8 abort negative

9 aborted negative

10 aborts negative

# ℹ 6,776 more rows

# ℹ Use `print(n = ...)` to see more rows

Positive and negative words used most frequently

|  |
| --- |
| #adding the date of the Tweets from the document level meta data  > DocMetaTrump1 <- meta(TrumpCorpus1)  > DocMetaTrump1$date <- date(DocMetaTrump1$time)  > TrumpTidy1$date <- DocMetaTrump1$date  >  > DocMetaClinton1 <- meta(ClintonCorpus1)  > DocMetaClinton1$date <- date(DocMetaClinton1$time)  > ClintonTidy1$date <- DocMetaClinton1$date  >  > NoNamesTidy <- bind\_rows(trump=TrumpTidy1, clinton=ClintonTidy1, .id="candidate")  > Words <- NoNamesTidy %>% unnest\_tokens(word, text)  Warning message:  Outer names are only allowed for unnamed scalar atomic inputs  >  > Bing <- Words %>% inner\_join(get\_sentiments("bing"), by="word") |
|  |
| |  | | --- | | > | |

## The nrc lexicon (2 sentiment categories, and 8 basic emotions)

> Nrc <- Words %>% inner\_join(get\_sentiments("nrc"), by="word")

Warning message:

In inner\_join(., get\_sentiments("nrc"), by = "word") :

Detected an unexpected many-to-many relationship between `x` and `y`.

ℹ Row 6 of `x` matches multiple rows in `y`.

ℹ Row 7277 of `y` matches multiple rows in `x`.

ℹ If a many-to-many relationship is expected, set `relationship = "many-to-many"` to silence

this warning.

> n1 <- Nrc %>% filter(candidate=="trump") %>% count(sentiment) %>%

+ ggplot(aes(x=sentiment, y=n, fill=sentiment)) +

+ geom\_bar(stat="identity") + coord\_polar() +

+ theme(legend.position = "none", axis.text.x = element\_blank()) +

+ geom\_text(aes(label=sentiment, y=2500)) +

+ labs(x="", y="", title="Trump")

> n2 <- Nrc %>% filter(candidate=="clinton") %>% count(sentiment) %>%

+ ggplot(aes(x=sentiment, y=n, fill=sentiment)) +

+ geom\_bar(stat="identity") + coord\_polar() +

+ theme(legend.position = "none", axis.text.x = element\_blank()) +

+ geom\_text(aes(label=sentiment, y=2500)) +

+ labs(x="", y="", title="Clinton")

> grid.arrange(n1, n2, nrow=1)

