Homework Lesson 7

Leon Woltermann

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
d1862 <- read.delim("dispatch_1862.tsv", encoding="UTF-8", header=TRUE, quote="")
sw1 <- scan("sw1.md", what="character", sep="\n")

# General ones
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library("RColorBrewer")

# Text Analysis Specific
library(stringr)
library(tidytext)
library(wordcloud)
library(quanteda)
```

```
## Package version: 3.2.1
## Unicode version: 13.0
## ICU version: 67.1

## Parallel computing: 8 of 8 threads used.

## See https://quanteda.io for tutorials and examples.
```

```
library(readtext)
library(textstem)
```

```
## Loading required package: koRpus.lang.en
```

```
## Loading required package: koRpus
```

```
## Loading required package: sylly
```

```
## For information on available language packages for 'koRpus', run
##
##   available.koRpus.lang()
##
## and see ?install.koRpus.lang()
```

```
##
## Attaching package: 'koRpus'
```

```
## The following objects are masked from 'package:quanteda':
##
##     tokens, types
```

```
## The following object is masked from 'package:readr':
##
##     tokenize
```

```
library(tidyr)
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
```

```
## The following object is masked from 'package:tidyr':
##
##     crossing
```

```
## The following object is masked from 'package:tibble':
##
##     as_data_frame
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(ggraph)
```

```
d1862 %>%
  count(type, sort=T)
```

```
##        type     n
## 1   article 14680
## 2  ad-blank  7029
## 3    orders  3987
## 4    advert  2037
## 5     death   233
## 6   married   186
## 7      died    70
## 8      poem    45
## 9     order    30
## 10   letter    27
## 11  ordered    15
## 12    entry     7
## 13  acticle     2
## 14  adverts     2
## 15   notice     2
## 16     role     2
## 17  runaway     2
## 18  article     1
## 19       25     1
## 20  aritcle     1
## 21   artcle     1
## 22   articl     1
## 23 articler     1
## 24   aticle     1
## 25   death,     1
## 26 married,     1
## 27    marry     1
## 28    oders     1
## 29     oped     1
## 30  ordinal     1
## 31 printrun     1
## 32  ranaway     1
## 33   simple     1
## 34   Wanted     1
```

```
#death subset
death_d1862 <- d1862 %>%
  filter(type=="death" | type == "died")

head(death_d1862)
```

```
##                     id       date  type header
```

```
## 1 1862-02-20_death_122 1862-02-20 death   Died
## 2  1862-12-01_death_59 1862-12-01 death  Died.
## 3  1862-10-28_death_73 1862-10-28 death  Died.
## 4   1862-11-22_died_66 1862-11-22  died  Died.
## 5  1862-02-17_death_94 1862-02-17 death  Died,
## 6  1862-10-25_death_30 1862-10-25 death  Died,
##
## 1
## 2
## 3
## 4
## 5
## 6 Died, Was killed at the battle of Sharpsburg, 17th September, Wm Myrtland Taylor in the 17th year
```

```
#marriage subset
marriage_d1862 <- d1862 %>%
  filter(type=="married")

head(marriage_d1862)
```

```
##                       id       date    type   header
## 1 1862-02-20_married_121 1862-02-20 married  Married
## 2  1862-12-01_married_58 1862-12-01 married Married.
## 3  1862-11-22_married_65 1862-11-22 married Married.
## 4  1862-02-17_married_93 1862-02-17 married Married,
## 5  1862-03-26_married_84 1862-03-26 married Married.
## 6  1862-12-22_married_54 1862-12-22 married Married.
##
## 1
## 2
## 3
## 4
## 5
## 6 Married. At Bellevue, in the county of King William, on the 18th inst, by the Rev. John O. Turpin,
```

```
#poem subset
poem_d1862 <- d1862 %>%
  filter(type=="poem")

head(poem_d1862)
```

```
##                    id       date type
## 1 1862-02-20_poem_179 1862-02-20 poem
## 2 1862-04-26_poem_174 1862-04-26 poem
## 3 1862-04-16_poem_218 1862-04-16 poem
## 4   1862-10-18_poem_1 1862-10-18 poem
## 5 1862-03-25_poem_216 1862-03-25 poem
## 6 1862-04-15_poem_260 1862-04-15 poem
##                                                                    header
## 1                                    [write for the Richmond Dispatch.]lines
## 2                                                                War Song
## 3                                 [written for the Richmond Dispatch.]Shiloh.
## 4                                                       A mother's Prayer.
```

```
## 5 The old grenadier's story.told on a Bench outside the Invalids.by G. W. Yeorebury.
## 6          [from the Petersburg Express.]the death of Gen. A. S. Johnston.by Lillie.
##
## 1
## 2
## 3
## 4
## 5 The old grenadier's story. told on a Bench outside the Invalids. by G. W. Yeorebury. T'was the day
## 6
```

#Describe problems with the data set and how they can be fixed.

```
test_set <- death_d1862

test_set_tidy <- test_set %>%
  mutate(item_number = cumsum(str_detect(text, regex("^", ignore_case = TRUE)))) %>%
  select(-type) %>%
  unnest_tokens(word, text) %>%
  mutate(word_number = row_number())

head(test_set_tidy)
```

```
##                     id       date header item_number     word word_number
## 1 1862-02-20_death_122 1862-02-20   Died           1   diedin           1
## 2 1862-02-20_death_122 1862-02-20   Died           1 richmond           2
## 3 1862-02-20_death_122 1862-02-20   Died           1       on           3
## 4 1862-02-20_death_122 1862-02-20   Died           1      the           4
## 5 1862-02-20_death_122 1862-02-20   Died           1     19th           5
## 6 1862-02-20_death_122 1862-02-20   Died           1     inst           6
```

```
data("stop_words")

test_set_tidy_clean <- test_set_tidy %>%
  anti_join(stop_words, by="word")

head(test_set_tidy_clean)
```

```
##                     id       date header item_number     word word_number
## 1 1862-02-20_death_122 1862-02-20   Died           1   diedin           1
## 2 1862-02-20_death_122 1862-02-20   Died           1 richmond           2
## 3 1862-02-20_death_122 1862-02-20   Died           1     19th           5
## 4 1862-02-20_death_122 1862-02-20   Died           1     inst           6
## 5 1862-02-20_death_122 1862-02-20   Died           1 berenice           8
## 6 1862-02-20_death_122 1862-02-20   Died           1 adelaide           9
```

```
#create a stop word list

frequency_list <- test_set_tidy %>%
  count(word, sort=T)

most_frequen_words <- head(frequency_list, 2000) %>%
  add_column(exclude = "")
```

```
test_set_tidy %>%
  count(word, sort = TRUE) %>%
  head(15)
```

```
##       word    n
## 1      the 9352
## 2       of 8148
## 3      and 5875
## 4      his 3390
## 5       to 3355
## 6       in 3272
## 7        a 2626
## 8       at 2388
## 9       on 2331
## 10      he 1595
## 11     her 1533
## 12     was 1211
## 13 o'clock 1102
## 14    from 1084
## 15    this 1084
```

###Wordclouds

```
#wordcloud including stop words

test_set_stopWords <- test_set_tidy %>%
  count(word, sort=T)

set.seed(1234)
wordcloud(words=test_set_stopWords$word, freq=test_set_stopWords$n,
          min.freq = 1, rot.per = .25, random.order=FALSE, #scale=c(5,.5),
          max.words=150, colors=brewer.pal(8, "Dark2"))
```

#What can we glean out form this wordcloud? Create a wordcloud for obituaries.

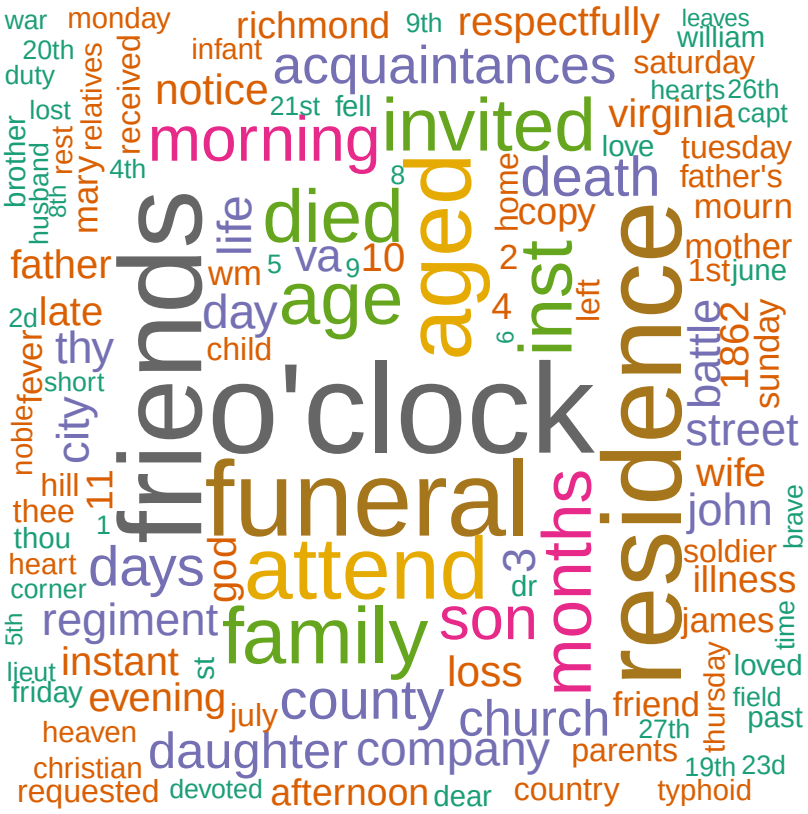
The wordcloud including stop words offers limited insights into the obituaries. The main words are prepositions or articles. However, there is one thing I recognized. "his" is more prominent than "her" and "she"

doesn't even occur in the center of the cloud. This could be an indication that more obituaries concerning men. Against the background of male soldiers fighting in the civil war this assumption seems quite probable.

```
#wordcloud without stop words
test_set_tidy_clean <- test_set_tidy %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort=T)

set.seed(1234)
wordcloud(words=test_set_tidy_clean$word, freq=test_set_tidy_clean$n,
          min.freq = 1, rot.per = .25, random.order=FALSE, #scale=c(5,.5),
          max.words=150, colors=brewer.pal(8, "Dark2"))
```

```
## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : papers could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : wednesday could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : streets could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : children could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : hope could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : affectionate could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : deceased could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : 17th could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : a.m could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : 18th could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : beloved could not be fit on page. It will not be
## plotted.
```

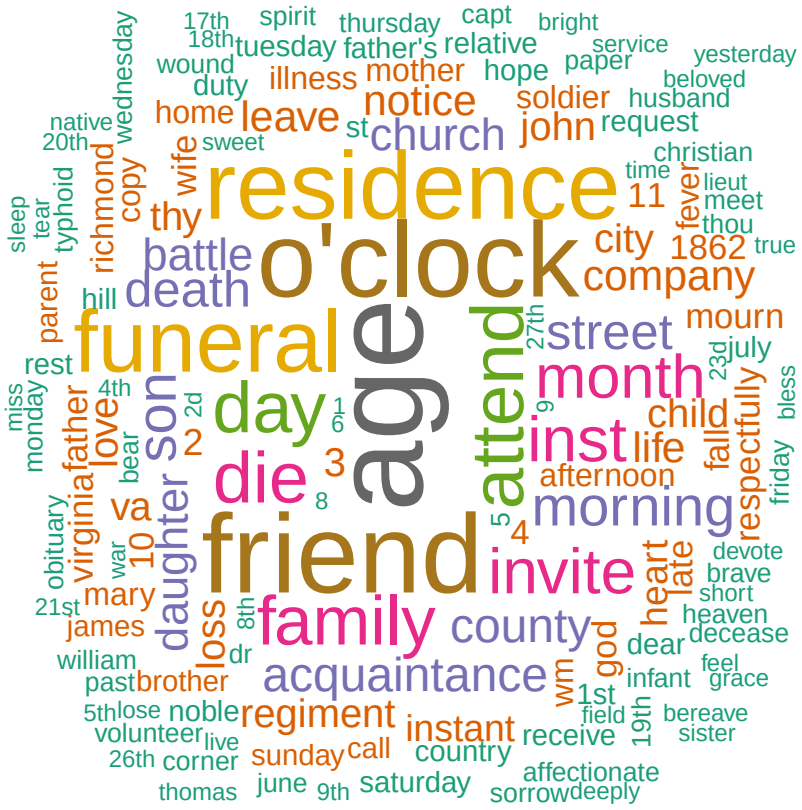
```
## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : spirit could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : volunteers could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : deeply could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : thomas could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : meet could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : native could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : called could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : true could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : yesterday could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : august could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : feel could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : miss could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : night could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = test_set_tidy_clean$word, freq =
## test_set_tidy_clean$n, : half could not be fit on page. It will not be plotted.
```



#Create a wordcloud for obituaries, but without stop words.

The wordcloud without stop words offers a more detailed picture concerning the obituaries. Compared to the previous wordcloud there are almost no green words on the edge of the cloud and a lot of bigger words in the center. The cloud is much more diverse making it also more complicated to interpret. “o’clock” is the most prominent word suggesting that the time of the death is important.

```
test_set_tidy_lemmatized <- test_set
#test_set_tidy_lemmatized$text <- lemmatize_strings(test_set$text)

test_set_tidy_lemmatized <- test_set_tidy_lemmatized %>%
  mutate(item_number = cumsum(str_detect(text, regex("^", ignore_case = TRUE)))) %>%
  select(-type) %>%
  mutate(lemmatizedText = lemmatize_strings(text)) %>%
  unnest_tokens(word, lemmatizedText) %>%
  mutate(word_number = row_number())

test_set_tidy_lemmatized_clean <- test_set_tidy_lemmatized %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort=T)

set.seed(1234)
wordcloud(words=test_set_tidy_lemmatized_clean$word, freq=test_set_tidy_lemmatized_clean$n,
          min.freq = 1, rot.per = .25, random.order=FALSE, #scale=c(5,.5),
          max.words=150, colors=brewer.pal(8, "Dark2"))
```



#Create a wordcloud for obituaries, but on lemmatized texts and without stop words.

In this wordcloud (similar to the first one) there is a clearer distribution of words. There are not as many bigger words than in the cloud before. The center and the edge are better distinguishable.

#Summarize your observations below.

My conclusion is that in the first cloud the center and the edge are very clearly distinguishable. The frequency of prepositions, articles and pronouns and other words is very different. This makes it easy to analyze but is rather a superficial insight into the data. In the second cloud the center and the edge are not very distinguishable. The frequencies of all the words are quite similar which makes it harder to find patterns. The last cloud is somewhat in between the two other clouds. The center and the edge are better distinguishable than in the second cloud. Different than in the first cloud there are not semantically meaningless words as prepositions, pronouns, and articles.

I think I would prefer the last one because it gives some semantic insights and the frequency of the words is still different. However, when analyzing grammar than the first one would be more convenient.

#Word Distribution Plots

```
test_set1 <- d1862
test_set1$date <- as.Date(test_set1$date, format="%Y-%m-%d")

test_set1_tidy <- test_set1 %>%
  mutate(item_number = cumsum(str_detect(text, regex("^", ignore_case = TRUE)))) %>%
  select(-type) %>%
  unnest_tokens(word, text) %>%
  mutate(word_number = row_number())

head(test_set1_tidy)
```

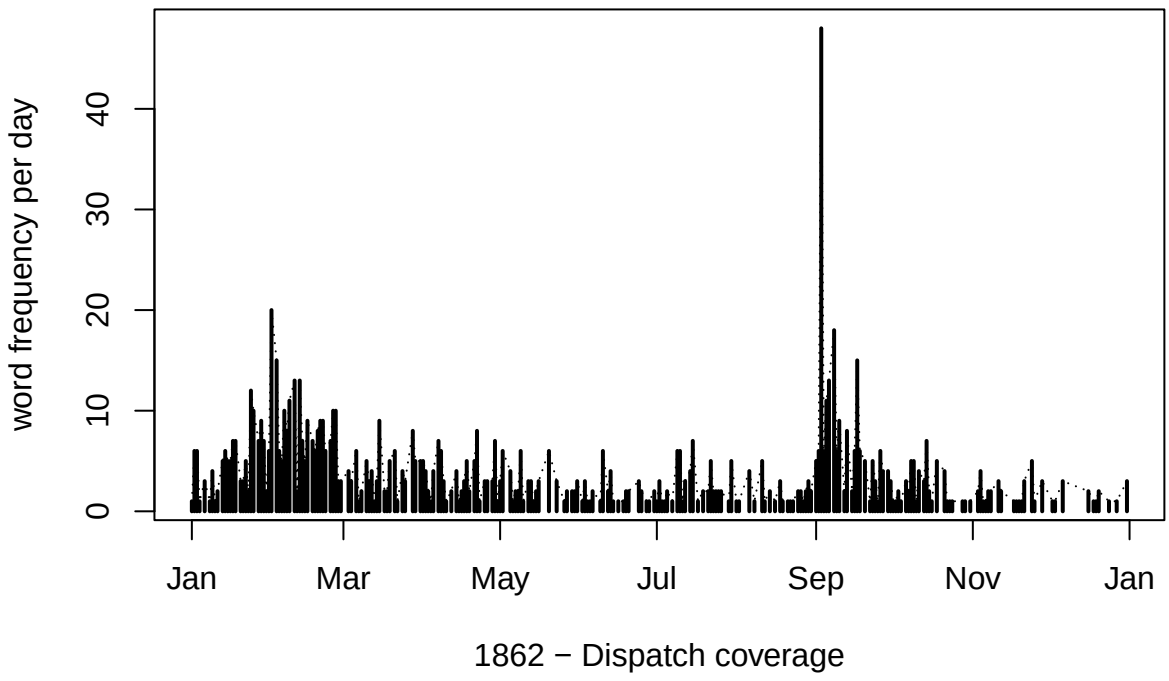
```
##                       id       date             header item_number     word
```

```
## 1 1862-02-20_ad-blank_20 1862-02-20 Richmond Dispatch.           1 richmond
## 2 1862-02-20_ad-blank_20 1862-02-20 Richmond Dispatch.           1 dispatch
## 3 1862-02-20_ad-blank_20 1862-02-20 Richmond Dispatch.           1 thursday
## 4 1862-02-20_ad-blank_20 1862-02-20 Richmond Dispatch.           1  morning
## 5 1862-02-20_ad-blank_20 1862-02-20 Richmond Dispatch.           1      feb
## 6 1862-02-20_ad-blank_20 1862-02-20 Richmond Dispatch.           1       20
##   word_number
## 1           1
## 2           2
## 3           3
## 4           4
## 5           5
## 6           6
```

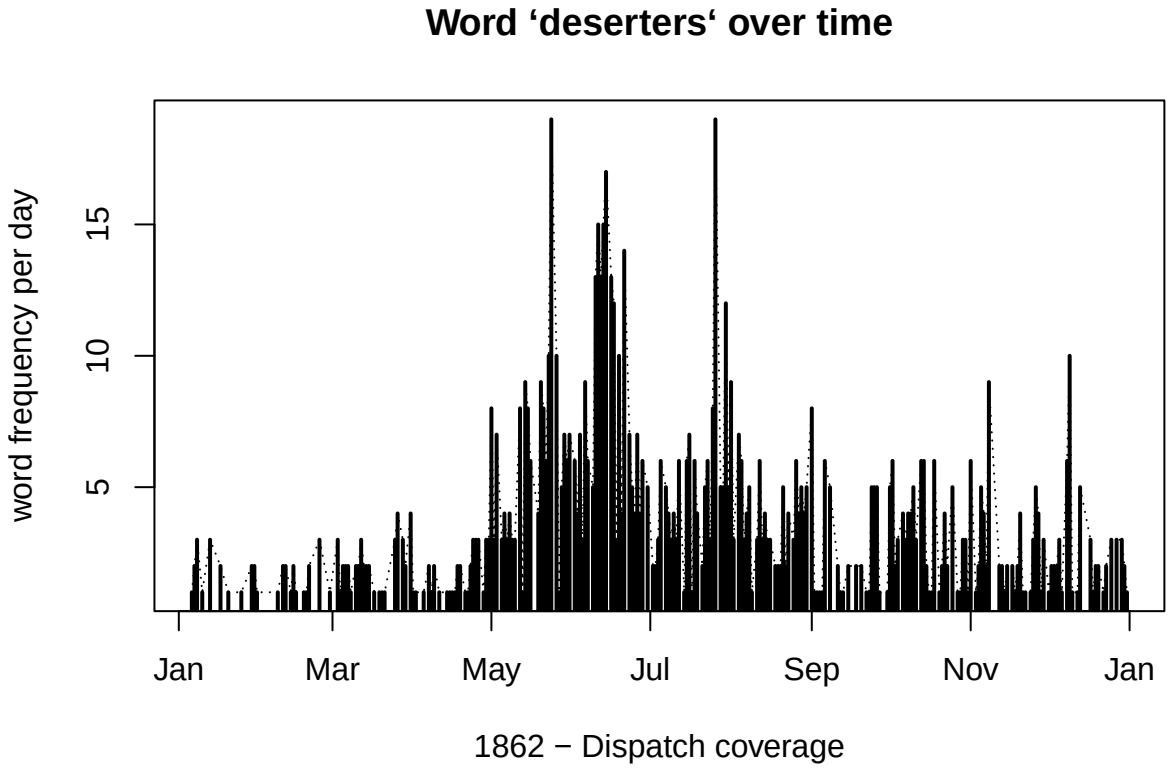
```
test_set1_tidy_freqDay <- test_set1_tidy %>%
  anti_join(stop_words, by="word") %>%
  group_by(date) %>%
  count(word)

head(test_set1_tidy_freqDay)
```

```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       word      n
##   <date>     <chr> <int>
## 1 1862-01-01 000      13
## 2 1862-01-01 007       1
## 3 1862-01-01 014       1
## 4 1862-01-01 1        11
## 5 1862-01-01 10        5
## 6 1862-01-01 100       2
```

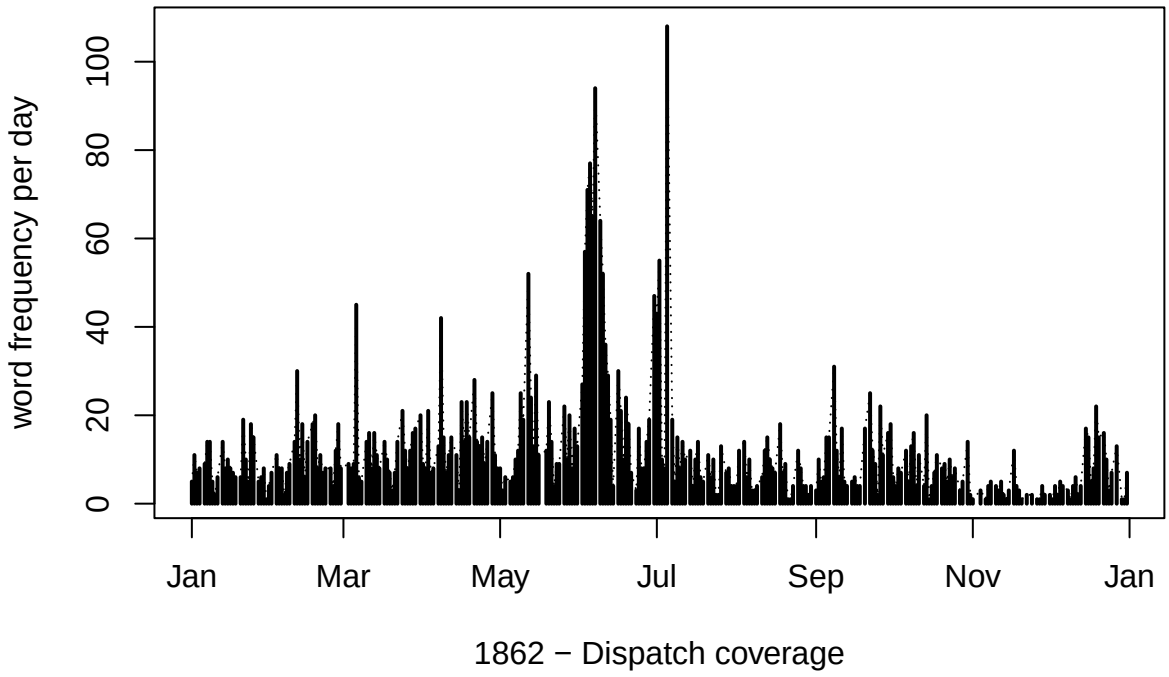
```
# interesting examples:
# deserters, killed,
# donelson (The Battle of Fort Donelson took place in early February of 1862),
# manassas (place of the Second Bull Run, fought in August 28-30, 1862),
# shiloh (Battle of Shiloh took place in April of 1862)

ourWord = "manassas"

test_set1_tidy_word <- test_set1_tidy_freqDay %>%
  filter(word==ourWord)

plot(x=test_set1_tidy_word$date, y=test_set1_tidy_word$n, type="l", lty=3, lwd=1,
     main=paste0("Word `", ourWord, "` over time"),
     xlab = "1862 - Dispatch coverage", ylab = "word frequency per day")
segments(x0=test_set1_tidy_word$date, x1=test_set1_tidy_word$date, y0=0, y1=test_set1_tidy_word$n, lty=
```




```
# deserters

ourWord = "deserters"

test_set1_tidy_word <- test_set1_tidy_freqDay %>%
  filter(word==ourWord)

plot(x=test_set1_tidy_word$date, y=test_set1_tidy_word$n, type="l", lty=3, lwd=1,
     main=paste0("Word `", ourWord, "` over time"),
     xlab = "1862 - Dispatch coverage", ylab = "word frequency per day")
segments(x0=test_set1_tidy_word$date, x1=test_set1_tidy_word$date, y0=0, y1=test_set1_tidy_word$n, lty=
```




```
#killed

ourWord = "killed"

test_set1_tidy_word <- test_set1_tidy_freqDay %>%
  filter(word==ourWord)

plot(x=test_set1_tidy_word$date, y=test_set1_tidy_word$n, type="l", lty=3, lwd=1,
     main=paste0("Word `", ourWord, "` over time"),
     xlab = "1862 - Dispatch coverage", ylab = "word frequency per day")
segments(x0=test_set1_tidy_word$date, x1=test_set1_tidy_word$date, y0=0, y1=test_set1_tidy_word$n, lty=
```



###The graph like this can be used in a different way. Try words killed and deserters. When do these words spike? Can you interpret these graphs?

The graph based on the word “deserters” spikes around May and August. This period coheres with the Peninsular Campaign. It was a “large-scale but unsuccessful Union effort to capture the Confederate capital at Richmond” (https://www.britannica.com/event/Peninsular-Campaign). Part of this campaign were the “The Battle of Seven Pines” and “The Seven Days Battles” on June and July. These two battles cohere with the spikes of the graph based on the word “killed”. The “Battle of Seven Pines” had a high number of casualties (11000) which also coheres with the second graph (https://en.wikipedia.org/wiki/Battle_of_Seven_Pines). Also the “Seven Days Battles” had a high number of casualties (https://en.wikipedia.org/wiki/Seven_Days_Battles).

###Bigrams

```
#creating a bigram
bigram_test <- d1862 %>%
  select(type, text) %>%
  unnest_tokens(bigram, text, token="ngrams", n=2) %>%
  separate(bigram, c("word1", "word2"), sep = " ")

head(bigram_test)
```

```
##       type    word1    word2
## 1 ad-blank richmond dispatch
## 2 ad-blank dispatch thursday
## 3 ad-blank thursday  morning
## 4 ad-blank  morning      feb
## 5 ad-blank      feb       20
## 6 ad-blank       20     1862
```

```
#filtering bigram
bigrams_filtered <- bigram_test %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

head(bigrams_filtered)
```

```
##       type    word1    word2
## 1 ad-blank richmond dispatch
## 2 ad-blank dispatch thursday
## 3 ad-blank thursday  morning
## 4 ad-blank  morning      feb
## 5 ad-blank      feb       20
## 6 ad-blank       20     1862
```

```
#uniting bigram
bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")

head(bigrams_united)
```

```
##       type            bigram
## 1 ad-blank richmond dispatch
## 2 ad-blank dispatch thursday
## 3 ad-blank  thursday morning
## 4 ad-blank       morning feb
## 5 ad-blank            feb 20
## 6 ad-blank           20 1862
```

```
#one-bigram-per-row
bigram_campaign <- bigrams_filtered %>%
  filter(word2 == "campaign") %>%
  count(type, word1, sort = TRUE)

head(bigram_campaign )
```

```
##      type    word1  n
## 1 article   winter 18
## 2 article   spring 17
## 3 article   summer 12
## 4 article  western  8
## 5 article virginia  7
## 6 article missouri  6
```

```
#tf_idf of bigrams
bigram_tf_idf <- bigrams_united %>%
  count(type, bigram) %>%
  bind_tf_idf(bigram, type, n) %>%
  arrange(desc(n))

head(bigram_tf_idf)
```

```
##       type          bigram    n          tf      idf      tf_idf
## 1 ad-blank           NA NA 1813 0.130319149 1.916923 0.249811723
## 2  article          5 feet 1220 0.001894336 1.580450 0.002993905
## 3  article  north carolina 1085 0.001684717 1.734601 0.002922312
## 4 ad-blank   auction sales  823 0.059157562 1.916923 0.113400468
## 5  article  dollars reward  814 0.001263926 2.427748 0.003068494
## 6   orders light artillery  670 0.007209573 1.734601 0.012505732
```

```
#bigram networks
bigram_count <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2))

bigram_graph <- bigram_count %>%
  filter(n > 300) %>%
  graph_from_data_frame()
```

```
set.seed(4321)

ggraph(bigram_graph, layout = "fr") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```