MDR Text Analysis

Nathan

April 23, 2018

## Text Analysis

library(XML)  
library(httr)  
library(rvest)

library(tidytext)  
library(rlist)  
library(tidytext)  
library(tidyverse)

library(lubridate)

library(scales)

library(curl)

#establish stop words  
mdr\_stop\_words <- bind\_rows(data\_frame(word = c("b", "6", "2018", "2017", "2014", "2015", "essure",  
 "reported", "report", "inserted", "information",  
 "female", "lot", "pt", "2008", "2009",  
 "30", "day", "micro", "inserts", "insert",  
 "2011", "2010", "4", "99.74", "fallopian"),  
 lexicon = c("custom")),  
 stop\_words)

#1 Read in the MDR data and create a tibble with date as data and text as character  
MDRresultsCSV <- read.csv("MDRresults.csv")

#Note: I know we should not use our local wd, but I could not get the markdown without obtaining an “error” regarding the csv rows. The online file can be obtained with the following code:

# MDRresultsCSV <- read.csv(curl("https://drive.google.com/open?id=1DryePp8gAl\_j9hX6nfMM4-LddCBBJ6Qw"))

MDRresultsTBL <- as\_tibble(MDRresultsCSV)  
MDRresults <- MDRresultsTBL %>%  
 mutate(text = as.character(text)) %>%  
 mutate(date = date(date))

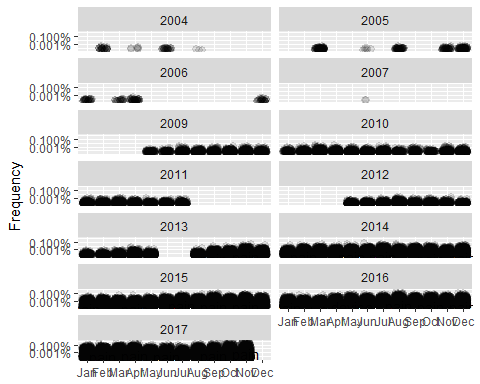
#2 total overall word count: No surprise that Pain would show up as the most common.  
MDRtotalwc <- MDRresults %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>%  
 anti\_join(mdr\_stop\_words) %>%  
 count(word, sort = TRUE) %>%  
 na.omit()

## Joining, by = "word"

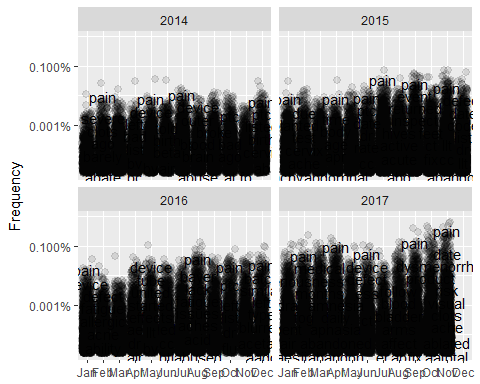
#3 Save frequencies of words  
MDRfreq <- MDRresults %>%  
 dplyr::select(-text\_type\_code, -patient\_sequence\_number) %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>%  
 anti\_join(mdr\_stop\_words) %>%  
 na.omit() %>%  
 mutate(month = month.abb[month(date)]) %>%  
 mutate(year = year(date)) %>%  
 count(month, year, word, sort = TRUE) %>%  
 mutate(freq = (n/sum(n))) %>%  
 arrange(desc(freq))

## Joining, by = "word"

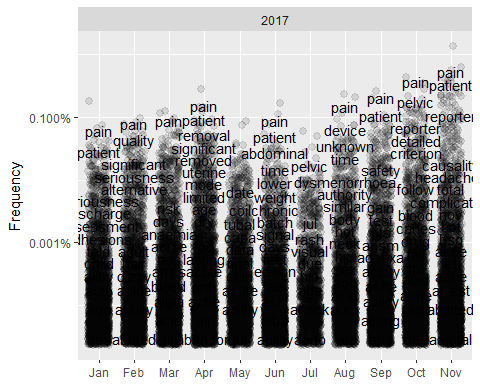
ggplot(MDRfreq, aes(x = factor(month, levels = month.abb), y = freq)) +  
 geom\_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +  
 geom\_text(aes(label = word), check\_overlap = TRUE, vjust = 1.5) +  
 scale\_y\_log10(labels = percent\_format()) +  
 scale\_color\_gradient(limits = c(0, 0.1), low ="darkslategray4", high = "gray75") +  
 facet\_wrap(~year, ncol = 2) +  
 theme(legend.position = "none") +  
 labs(y = "Frequency", x = NULL)



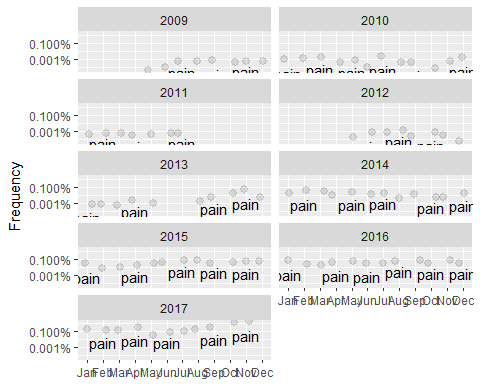
#4 Plotting word frequencies  
MDRfreq14 <- MDRfreq %>%  
 filter(year >= 2014)  
#Narrowing years  
ggplot(MDRfreq14, aes(x = factor(month, levels = month.abb), y = freq)) +  
 geom\_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +  
 geom\_text(aes(label = word), check\_overlap = TRUE, vjust = 1.5) +  
 scale\_y\_log10(labels = percent\_format()) +  
 scale\_color\_gradient(limits = c(0, 0.1), low ="darkslategray4", high = "gray75") +  
 facet\_wrap(~year, ncol = 2) +  
 theme(legend.position = "none") +  
 labs(y = "Frequency", x = NULL)



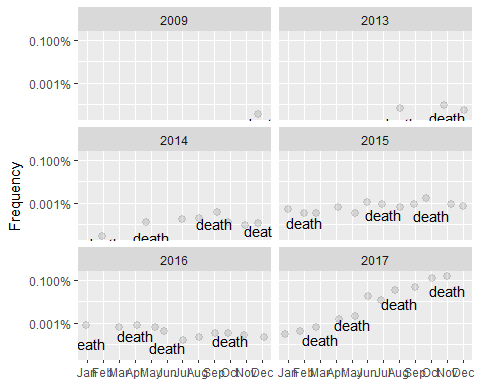
MDRfreq17 <- MDRfreq %>%  
 filter(year == 2017)  
#Or choose by year  
ggplot(MDRfreq17, aes(x = factor(month, levels = month.abb), y = freq)) +  
 geom\_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +  
 geom\_text(aes(label = word), check\_overlap = TRUE, vjust = 1.5) +  
 scale\_y\_log10(labels = percent\_format()) +  
 scale\_color\_gradient(limits = c(0, 0.1), low ="darkslategray4", high = "gray75") +  
 facet\_wrap(~year, ncol = 2) +  
 theme(legend.position = "none") +  
 labs(y = "Frequency", x = NULL)



#5 plot the rate of pain over the years  
MDRpain <- MDRfreq %>%  
 filter(word == "pain",  
 year >= 2009)  
ggplot(MDRpain, aes(x = factor(month, levels = month.abb), y = freq)) +  
 geom\_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +  
 geom\_text(aes(label = word), check\_overlap = TRUE, vjust = 1.5) +  
 scale\_y\_log10(labels = percent\_format()) +  
 scale\_color\_gradient(limits = c(0, 0.1), low ="darkslategray4", high = "gray75") +  
 facet\_wrap(~year, ncol = 2) +  
 theme(legend.position = "none") +  
 labs(y = "Frequency", x = NULL)



#did anyone die?  
MDRdeath <- MDRfreq %>%  
 filter(word == "death")  
  
ggplot(MDRdeath, aes(x = factor(month, levels = month.abb), y = freq)) +  
 geom\_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +  
 geom\_text(aes(label = word), check\_overlap = TRUE, vjust = 1.5) +  
 scale\_y\_log10(labels = percent\_format()) +  
 scale\_color\_gradient(limits = c(0, 0.1), low ="darkslategray4", high = "gray75") +  
 facet\_wrap(~year, ncol = 2) +  
 theme(legend.position = "none") +  
 labs(y = "Frequency", x = NULL)



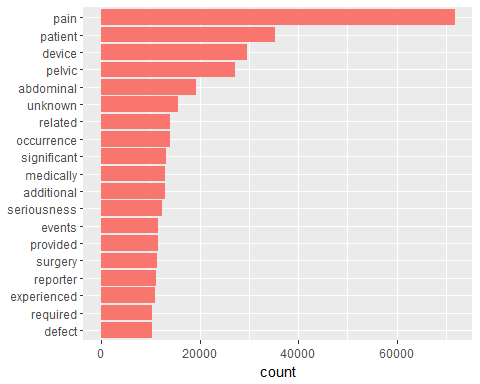
#6 obtain total word count:  
MDRcount <- MDRresults %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>%  
 anti\_join(mdr\_stop\_words) %>%  
 count(word, sort = TRUE) %>%  
 na.omit()

## Joining, by = "word"

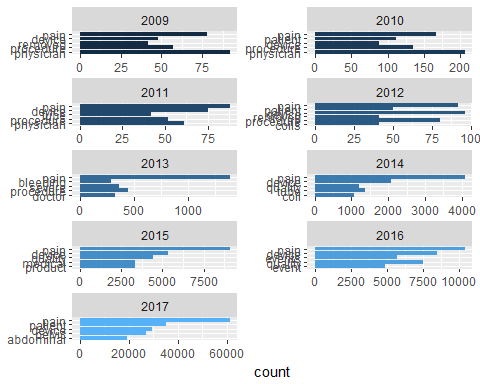
#word count over years:  
MDRyearcount <- MDRresults %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>%  
 mutate(year = year(date)) %>%  
 anti\_join(mdr\_stop\_words) %>%  
 count(year, word, sort = TRUE) %>%  
 na.omit()

## Joining, by = "word"

#plot wordcount   
plot\_MDRyearcount <- MDRyearcount %>%  
 count(word, n) %>%  
 arrange(desc(n)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%  
 top\_n(20, n) %>%  
 ungroup()  
  
ggplot(plot\_MDRyearcount, aes(word, n, fill = "red")) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "count") +  
 coord\_flip()



#plot wordcount over years from 2009  
plot\_MDRyearcount <- MDRyearcount %>%  
 count(word, year, n) %>%  
 arrange(desc(n)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%  
 group\_by(year) %>%  
 top\_n(5, n) %>%  
 filter(year >= 2009) %>%  
 ungroup()  
  
ggplot(plot\_MDRyearcount, aes(word, n, fill = year)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "count") +  
 facet\_wrap(~year, ncol = 2, scales = "free") +  
 coord\_flip()

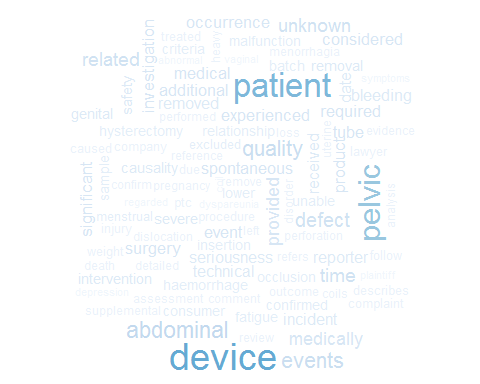


#word cloud  
library(wordcloud)

## Loading required package: RColorBrewer

MDRtotalwc %>%  
 with(wordcloud(word, n, max.words = 100, colors=colorRampPalette(brewer.pal(9,"Blues"))(32)))

## Warning in wordcloud(word, n, max.words = 100, colors =  
## colorRampPalette(brewer.pal(9, : pain could not be fit on page. It will not  
## be plotted.



#7 using bigrams  
MDRbigrams <- MDRresults %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
#n = 2 sets pairs of two consecutive words, also known as bigrams  
MDRbigrams

## # A tibble: 4,713,447 x 6  
## X date mdr\_text\_key text\_type\_code patient\_sequenc~ bigram   
## <int> <date> <int> <fct> <int> <chr>   
## 1 1 2009-06-29 1270211 Description of ~ 1 pt had   
## 2 1 2009-06-29 1270211 Description of ~ 1 had es~  
## 3 1 2009-06-29 1270211 Description of ~ 1 essure~  
## 4 1 2009-06-29 1270211 Description of ~ 1 proced~  
## 5 1 2009-06-29 1270211 Description of ~ 1 perfor~  
## 6 1 2009-06-29 1270211 Description of ~ 1 on b   
## 7 1 2009-06-29 1270211 Description of ~ 1 b 6   
## 8 1 2009-06-29 1270211 Description of ~ 1 6 2008   
## 9 1 2009-06-29 1270211 Description of ~ 1 2008 w~  
## 10 1 2009-06-29 1270211 Description of ~ 1 with b~  
## # ... with 4,713,437 more rows

#check the most common bigrams:  
MDRbigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 309,839 x 2  
## bigram n  
## <chr> <int>  
## 1 b 6 37044  
## 2 pelvic pain 30022  
## 3 the patient 27886  
## 4 of the 24199  
## 5 had essure 22236  
## 6 on b 20125  
## 7 abdominal pain 17675  
## 8 in the 14465  
## 9 occurrence of 14278  
## 10 the occurrence 14262  
## # ... with 309,829 more rows

#to remove these common stopwords, start by separating the individual words in each column using separate:  
  
bigrams\_separated <- MDRbigrams %>%  
 separate(bigram, c("word1", "word2", sep = " "))

## Warning: Expected 3 pieces. Additional pieces discarded in 161 rows [35760,  
## 92137, 92138, 105372, 105373, 138268, 138269, 149449, 257376, 257377,  
## 309221, 309406, 313841, 373098, 432221, 432222, 438217, 438218, 479475,  
## 479476, ...].

## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 4687744  
## rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  
## 20, ...].

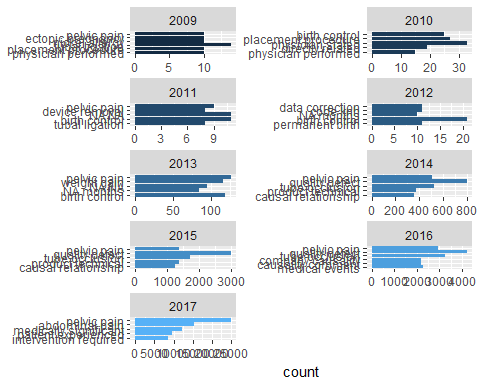
bigrams\_filtered <- bigrams\_separated %>%  
 filter(!word1 %in% mdr\_stop\_words$word) %>%  
 filter(!word2 %in% mdr\_stop\_words$word) %>%  
 mutate(word1 = str\_extract(word1, "[a-z']+")) %>%  
 mutate(word2 = str\_extract(word2, "[a-z']+"))   
  
bigram\_counts <- bigrams\_filtered %>%  
 count(word1, word2, sort = TRUE)

#8 highest bigram counts  
bigram\_counts

## # A tibble: 111,672 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 pelvic pain 30025  
## 2 abdominal pain 17676  
## 3 medically significant 12475  
## 4 quality defect 11892  
## 5 patient experienced 9949  
## 6 intervention required 8793  
## 7 genital haemorrhage 8724  
## 8 unknown date 8589  
## 9 seriousness criteria 8515  
## 10 criteria medically 8467  
## # ... with 111,662 more rows

#it is now time to recombine- when dealing with bigrams, keep in mind: separate->filter->count->unite  
bigrams\_united <- bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ") %>%  
 select(-` `)

#9 Analyzing bigrams  
#bigram counts per year  
bigrams\_united\_count <- bigrams\_united %>%  
 mutate(year = year(date)) %>%  
 count(year, bigram, sort = TRUE)  
  
plot\_bigrams\_united <- bigrams\_united\_count %>%  
 count(bigram, year, n) %>%  
 arrange(desc(n)) %>%  
 mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%  
 group\_by(year) %>%  
 top\_n(5, n) %>%  
 filter(year >= 2009) %>%  
 ungroup()  
  
ggplot(plot\_bigrams\_united, aes(bigram, n, fill = year)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "count") +  
 facet\_wrap(~year, ncol = 2, scales = "free") +  
 coord\_flip()



#10 let's check what kind of pain we are seeing  
bigram\_counts %>%  
 filter(word2 == "pain") %>%  
 print(n = 10)

## # A tibble: 1,013 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 pelvic pain 30025  
## 2 abdominal pain 17676  
## 3 pain pain 2614  
## 4 joint pain 1627  
## 5 severe pain 1351  
## 6 procedural pain 1303  
## 7 vulvovaginal pain 1189  
## 8 menstrual pain 922  
## 9 uncharacterized pain 568  
## 10 dyspareunia pain 554  
## # ... with 1,003 more rows

#what about loss?  
bigram\_counts %>%  
 filter(word2 == "loss") %>%  
 print(n = 10)

## # A tibble: 355 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 hair loss 2687  
## 2 memory loss 594  
## 3 weight loss 275  
## 4 tooth loss 233  
## 5 blood loss 190  
## 6 fluid loss 92  
## 7 libido loss 80  
## 8 sweats loss 73  
## 9 pain loss 69  
## 10 fatigue loss 58  
## # ... with 345 more rows

#allergy?  
bigram\_counts %>%  
 filter(word2 == "allergy") %>%  
 print(n = 10)

## # A tibble: 275 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 nickel allergy 773  
## 2 experienced allergy 145  
## 3 pain allergy 139  
## 4 metals allergy 120  
## 5 considered allergy 99  
## 6 lower allergy 68  
## 7 term allergy 47  
## 8 device allergy 40  
## 9 metal allergy 40  
## 10 required allergy 29  
## # ... with 265 more rows

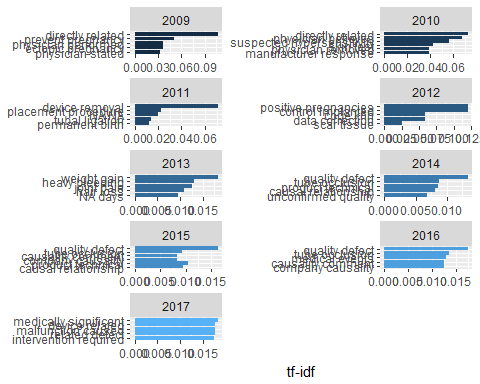
#did the device work?  
bigram\_counts %>%  
 filter(word2 == "birth") %>%  
 print(n = 50)

## # A tibble: 85 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 permanent birth 410  
## 2 live birth 109  
## 3 giving birth 50  
## 4 premature birth 14  
## 5 hormonal birth 13  
## 6 child birth 11  
## 7 oral birth 9  
## 8 low birth 8  
## 9 prescribed birth 8  
## 10 effective birth 7  
## 11 preterm birth 7  
## 12 taking birth 7  
## 13 discuss birth 5  
## 14 vbac birth 5  
## 15 alternate birth 3  
## 16 alternative birth 3  
## 17 backup birth 3  
## 18 continuous birth 3  
## 19 hormone birth 3  
## 20 baby birth 2  
## 21 current birth 2  
## 22 delivery birth 2  
## 23 dose birth 2  
## 24 horrible birth 2  
## 25 illnesses birth 2  
## 26 normal birth 2  
## 27 prescribing birth 2  
## 28 vaginal birth 2  
## 29 add birth 1  
## 30 additional birth 1  
## 31 b birth 1  
## 32 coils birth 1  
## 33 control birth 1  
## 34 cyclen birth 1  
## 35 daughter birth 1  
## 36 denied birth 1  
## 37 elective birth 1  
## 38 finalize birth 1  
## 39 foreign birth 1  
## 40 free birth 1  
## 41 healthy birth 1  
## 42 home birth 1  
## 43 hormones birth 1  
## 44 hypertension birth 1  
## 45 implant birth 1  
## 46 implantable birth 1  
## 47 including birth 1  
## 48 insertion birth 1  
## 49 iv birth 1  
## 50 killers birth 1  
## # ... with 35 more rows

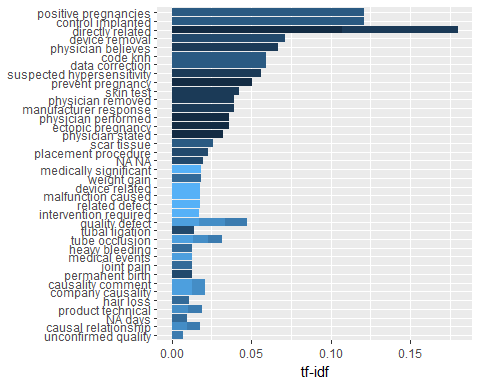
#11 we can also look at the tf-idf of bigrams to analyze importance   
bigram\_tf\_idf <- bigrams\_united %>%  
 mutate(year = year(date)) %>%  
 count(year, bigram) %>%  
 bind\_tf\_idf(bigram, year, n) %>%  
 filter(n > 5) %>%  
 arrange(desc(tf\_idf))  
  
bigram\_tf\_idf

## # A tibble: 12,794 x 6  
## year bigram n tf idf tf\_idf  
## <dbl> <chr> <int> <dbl> <dbl> <dbl>  
## 1 2017. medically significant 12191 0.0200 1.10 0.0220  
## 2 2016. quality defect 4244 0.0231 0.875 0.0203  
## 3 2017. intervention required 8554 0.0141 1.39 0.0195  
## 4 2017. criteria medically 8302 0.0136 1.39 0.0189  
## 5 2015. quality defect 3013 0.0193 0.875 0.0169  
## 6 2017. genital haemorrhage 8430 0.0138 1.10 0.0152  
## 7 2017. seriousness criteria 8426 0.0138 1.10 0.0152  
## 8 2017. unknown date 8408 0.0138 1.10 0.0152  
## 9 2017. device related 6356 0.0104 1.39 0.0145  
## 10 2017. malfunction caused 6356 0.0104 1.39 0.0145  
## # ... with 12,784 more rows

#and plot   
#by year  
plot\_bigrams <- bigram\_tf\_idf %>%  
 bind\_tf\_idf(bigram, year, n) %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%  
 group\_by(year) %>%  
 top\_n(5, tf\_idf) %>%  
 filter(year >= 2009) %>%  
 ungroup()  
  
ggplot(plot\_bigrams, aes(bigram, tf\_idf, fill = year)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~year, ncol = 2, scales = "free") +  
 coord\_flip()



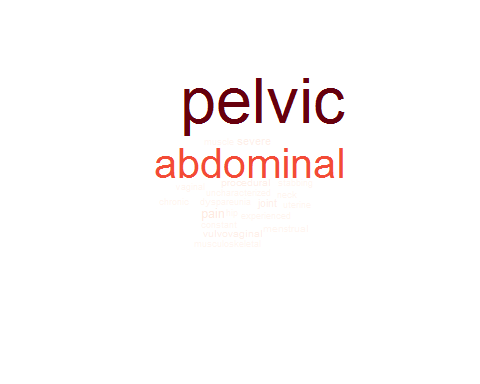
#all together  
plot\_bigrams <- bigram\_tf\_idf %>%  
 bind\_tf\_idf(bigram, year, n) %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%  
 group\_by(year) %>%  
 top\_n(5, tf\_idf) %>%  
 ungroup()  
  
ggplot(plot\_bigrams, aes(bigram, tf\_idf, fill = year)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 coord\_flip()



#12 Try some different word clouds:  
bigrams\_united %>%  
 count(bigram, sort = TRUE) %>%  
 with(wordcloud(bigram, n, max.words = 50, colors=colorRampPalette(brewer.pal(9,"Blues"))(25)))



#by type of pain  
bigram\_counts %>%  
 filter(word2 == "pain",  
 n >= 100) %>%  
 with(wordcloud(word1, n, max.words = 20, colors=colorRampPalette(brewer.pal(9,"Reds"))(32)))



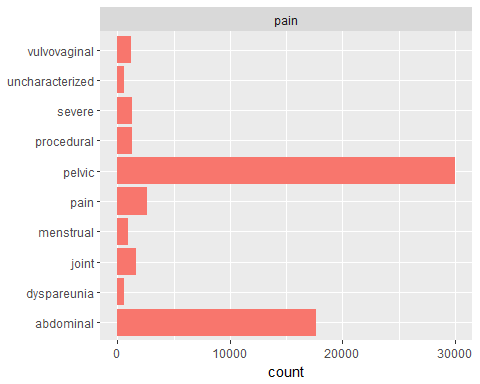
#by type of loss  
bigram\_counts %>%  
 filter(word2 == "loss",  
 n >= 100) %>%  
 with(wordcloud(word1, n, max.words = 50, colors=colorRampPalette(brewer.pal(9,"Greens"))(32)))



#13 filtering out words of interest:  
#Plot of Pain:  
MDRpainwords <- bigrams\_filtered %>%  
 filter(word2 == "pain") %>%  
 count(word1, word2, sort = TRUE) %>%  
 ungroup()  
  
MDRpainwords

## # A tibble: 1,013 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 pelvic pain 30025  
## 2 abdominal pain 17676  
## 3 pain pain 2614  
## 4 joint pain 1627  
## 5 severe pain 1351  
## 6 procedural pain 1303  
## 7 vulvovaginal pain 1189  
## 8 menstrual pain 922  
## 9 uncharacterized pain 568  
## 10 dyspareunia pain 554  
## # ... with 1,003 more rows

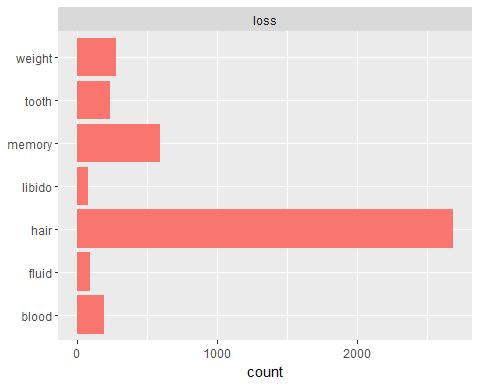
plot\_of\_pain <- MDRpainwords %>%  
 group\_by(word1) %>%  
 arrange(desc(abs(n))) %>%  
 filter(n > 500) %>%  
 ungroup()  
  
ggplot(plot\_of\_pain, aes(word1, n, fill = word2)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "count") +  
 facet\_wrap(~word2, ncol = 2, scales = "free") +  
 coord\_flip()



#Plot of Loss:  
MDRlosswords <- bigrams\_filtered %>%  
 filter(word2 == "loss") %>%  
 count(word1, word2, sort = TRUE) %>%  
 ungroup()  
  
MDRlosswords

## # A tibble: 355 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 hair loss 2687  
## 2 memory loss 594  
## 3 weight loss 275  
## 4 tooth loss 233  
## 5 blood loss 190  
## 6 fluid loss 92  
## 7 libido loss 80  
## 8 sweats loss 73  
## 9 pain loss 69  
## 10 fatigue loss 58  
## # ... with 345 more rows

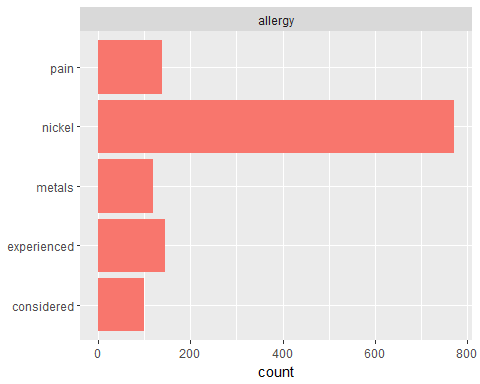
plot\_of\_loss <- MDRlosswords %>%  
 group\_by(word1) %>%  
 arrange(desc(abs(n))) %>%  
 filter(n > 75) %>%  
 ungroup()  
  
ggplot(plot\_of\_loss, aes(word1, n, fill = word2)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "count") +  
 facet\_wrap(~word2, ncol = 2, scales = "free") +  
 coord\_flip()



#Plot of allergies:  
MDRallergy <- bigrams\_filtered %>%  
 filter(word2 == "allergy") %>%  
 count(word1, word2, sort = TRUE) %>%  
 ungroup()  
  
MDRallergy

## # A tibble: 275 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 nickel allergy 773  
## 2 experienced allergy 145  
## 3 pain allergy 139  
## 4 metals allergy 120  
## 5 considered allergy 99  
## 6 lower allergy 68  
## 7 term allergy 47  
## 8 device allergy 40  
## 9 metal allergy 40  
## 10 required allergy 29  
## # ... with 265 more rows

plot\_of\_allergy <- MDRallergy %>%  
 group\_by(word1) %>%  
 arrange(desc(abs(n))) %>%  
 filter(n > 75) %>%  
 ungroup()  
  
ggplot(plot\_of\_allergy, aes(word1, n, fill = word2)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "count") +  
 facet\_wrap(~word2, ncol = 2, scales = "free") +  
 coord\_flip()



#14 We could try some Trigrams  
MDRtrigrams <- MDRresults %>%  
 unnest\_tokens(trigram, text, token = "ngrams", n = 3)  
#n = 3 sets pairs of three consecutive words, also known as trigrams  
MDRtrigrams

## # A tibble: 4,691,024 x 6  
## X date mdr\_text\_key text\_type\_code patient\_sequence~ trigram  
## <int> <date> <int> <fct> <int> <chr>   
## 1 1 2009-06-29 1270211 Description of~ 1 pt had~  
## 2 1 2009-06-29 1270211 Description of~ 1 had es~  
## 3 1 2009-06-29 1270211 Description of~ 1 essure~  
## 4 1 2009-06-29 1270211 Description of~ 1 proced~  
## 5 1 2009-06-29 1270211 Description of~ 1 perfor~  
## 6 1 2009-06-29 1270211 Description of~ 1 on b 6   
## 7 1 2009-06-29 1270211 Description of~ 1 b 6 20~  
## 8 1 2009-06-29 1270211 Description of~ 1 6 2008~  
## 9 1 2009-06-29 1270211 Description of~ 1 2008 w~  
## 10 1 2009-06-29 1270211 Description of~ 1 with b~  
## # ... with 4,691,014 more rows

#check the most common trigrams:  
MDRtrigrams %>%  
 count(trigram, sort = TRUE)

## # A tibble: 854,701 x 2  
## trigram n  
## <chr> <int>  
## 1 on b 6 19839  
## 2 the occurrence of 14262  
## 3 who had essure 12998  
## 4 had essure inserted 12451  
## 5 the patient experienced 9890  
## 6 female patient who 9204  
## 7 medically significant and 9047  
## 8 there is no 9034  
## 9 at the time 8745  
## 10 the time of 8642  
## # ... with 854,691 more rows

#to remove these common stopwords, start by separating the individual words in each column using separate:  
  
trigrams\_separated <- MDRtrigrams %>%  
 separate(trigram, c("word1", "word2", "word3", sep = " "))

## Warning: Expected 4 pieces. Additional pieces discarded in 428 rows [187,  
## 1351, 5482, 29339, 35400, 35401, 38082, 38084, 40316, 40958, 56624, 88866,  
## 90803, 91307, 91308, 91309, 97885, 98472, 104415, 104416, ...].

## Warning: Expected 4 pieces. Missing pieces filled with `NA` in 4652905  
## rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  
## 20, ...].

trigrams\_filtered <- trigrams\_separated %>%  
 filter(!word1 %in% mdr\_stop\_words$word) %>%  
 filter(!word2 %in% mdr\_stop\_words$word) %>%  
 filter(!word3 %in% mdr\_stop\_words$word)   
  
trigram\_counts <- trigrams\_filtered %>%  
 count(word1, word2, word3, sort = TRUE)  
  
#highest trigram counts  
trigram\_counts

## # A tibble: 145,688 x 4  
## word1 word2 word3 n  
## <chr> <chr> <chr> <int>  
## 1 criteria medically significant 8467  
## 2 seriousness criteria medically 8245  
## 3 company causality comment 6785  
## 4 device related defect 6363  
## 5 abdominal pain lower 5466  
## 6 pain seriousness criteria 4624  
## 7 experienced pelvic pain 4336  
## 8 patient experienced pelvic 4183  
## 9 product quality defect 4142  
## 10 pelvic pain seriousness 4042  
## # ... with 145,678 more rows

#it is now time to recombine  
trigrams\_united <- trigrams\_filtered %>%  
 unite(trigram, word1, word2, word3, sep = " ") %>%  
 select(-` `)  
  
#Analyzing trigrams  
#bigram counts per year  
trigrams\_united\_count <- trigrams\_united %>%  
 mutate(year = year(date)) %>%  
 count(year, trigram, sort = TRUE)  
  
#trigram tf idf  
trigram\_tf\_idf <- trigrams\_united %>%  
 mutate(year = year(date)) %>%  
 count(year, trigram) %>%  
 bind\_tf\_idf(trigram, year, n) %>%  
 filter(n > 5) %>%  
 arrange(desc(tf\_idf))  
  
trigram\_tf\_idf

## # A tibble: 7,902 x 6  
## year trigram n tf idf tf\_idf  
## <dbl> <chr> <int> <dbl> <dbl> <dbl>  
## 1 2017. criteria medically significant 8302 0.0217 1.39 0.0301  
## 2 2017. seriousness criteria medically 8188 0.0214 1.39 0.0297  
## 3 2016. company causality comment 2199 0.0271 0.875 0.0237  
## 4 2017. device related defect 6352 0.0166 1.39 0.0231  
## 5 2016. quality safety evaluation 1277 0.0157 1.39 0.0218  
## 6 2016. product quality defect 2006 0.0247 0.875 0.0216  
## 7 2016. technical failure mode 1222 0.0151 1.39 0.0209  
## 8 2016. manufacturing batch record 1279 0.0158 1.10 0.0173  
## 9 2017. pain seriousness criteria 4614 0.0121 1.39 0.0167  
## 10 2017. abdominal pain lower 5402 0.0141 1.10 0.0155  
## # ... with 7,892 more rows

plot\_trigrams <- trigram\_tf\_idf %>%  
 bind\_tf\_idf(trigram, year, n) %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(trigram = factor(trigram, levels = rev(unique(trigram)))) %>%  
 group\_by(year) %>%  
 top\_n(5, tf\_idf) %>%  
 ungroup()  
  
ggplot(plot\_trigrams, aes(trigram, tf\_idf, fill = year)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 coord\_flip()

