Drugs.R

ericl

2020-04-26

```
# 1 prep environment
packages <- c("tidyverse","lubridate","tidytext","wordcloud",
              "RColorBrewer","SnowballC")
lapply(packages, library, character.only = TRUE)
```

```
## -- Attaching packages ---------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

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

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: RColorBrewer

## [[1]]
##  [1] "forcats"   "stringr"   "dplyr"     "purrr"     "readr"     "tidyr"
##  [7] "tibble"    "ggplot2"   "tidyverse" "stats"     "graphics"  "grDevices"
## [13] "utils"     "datasets"  "methods"   "base"
##
## [[2]]
##  [1] "lubridate" "forcats"   "stringr"   "dplyr"     "purrr"     "readr"
##  [7] "tidyr"     "tibble"    "ggplot2"   "tidyverse" "stats"     "graphics"
## [13] "grDevices" "utils"     "datasets"  "methods"   "base"
##
```

```
## [[3]]
##  [1] "tidytext"  "lubridate" "forcats"   "stringr"   "dplyr"     "purrr"
##  [7] "readr"     "tidyr"     "tibble"    "ggplot2"   "tidyverse" "stats"
## [13] "graphics"  "grDevices" "utils"     "datasets"  "methods"   "base"
##
## [[4]]
##  [1] "wordcloud"    "RColorBrewer" "tidytext"     "lubridate"    "forcats"
##  [6] "stringr"      "dplyr"        "purrr"        "readr"        "tidyr"
## [11] "tibble"       "ggplot2"      "tidyverse"    "stats"        "graphics"
## [16] "grDevices"    "utils"        "datasets"     "methods"      "base"
##
## [[5]]
##  [1] "wordcloud"    "RColorBrewer" "tidytext"     "lubridate"    "forcats"
##  [6] "stringr"      "dplyr"        "purrr"        "readr"        "tidyr"
## [11] "tibble"       "ggplot2"      "tidyverse"    "stats"        "graphics"
## [16] "grDevices"    "utils"        "datasets"     "methods"      "base"
##
## [[6]]
##  [1] "SnowballC"    "wordcloud"    "RColorBrewer" "tidytext"     "lubridate"
##  [6] "forcats"      "stringr"      "dplyr"        "purrr"        "readr"
## [11] "tidyr"        "tibble"       "ggplot2"      "tidyverse"    "stats"
## [16] "graphics"     "grDevices"    "utils"        "datasets"     "methods"
## [21] "base"
```

```
# 2 get data
data1 <- read_tsv("drugsComTrain_raw.tsv", na="NA")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   drugName = col_character(),
##   condition = col_character(),
##   review = col_character(),
##   rating = col_double(),
##   date = col_character(),
##   usefulCount = col_double()
## )
```

```
# 3 inspect, clean
summary(data1)
```

```
##        X1           drugName          condition            review
##  Min.   :     2   Length:161297      Length:161297      Length:161297
##  1st Qu.: 58063   Class :character   Class :character   Class :character
##  Median :115744   Mode  :character   Mode  :character   Mode  :character
##  Mean   :115924
##  3rd Qu.:173776
##  Max.   :232291
##      rating           date            usefulCount
##  Min.   : 1.000   Length:161297      Min.   :   0
```

```
##  1st Qu.: 5.000   Class :character   1st Qu.:   6
##  Median : 8.000   Mode  :character   Median :  16
##  Mean   : 6.994                      Mean   :  28
##  3rd Qu.:10.000                      3rd Qu.:  36
##  Max.   :10.000                      Max.   :1291
```

```
# condition: has some hmtl entries - replace with ""
index <- str_which(data1$condition, "users found this comment" )
data1 <- mutate(data1, condition = replace(condition, index, "" ) )
# date: change to proper format
data1$date <- as_date( mdy(data1$date) )
which(!complete.cases(data1))
```

```
## integer(0)
```

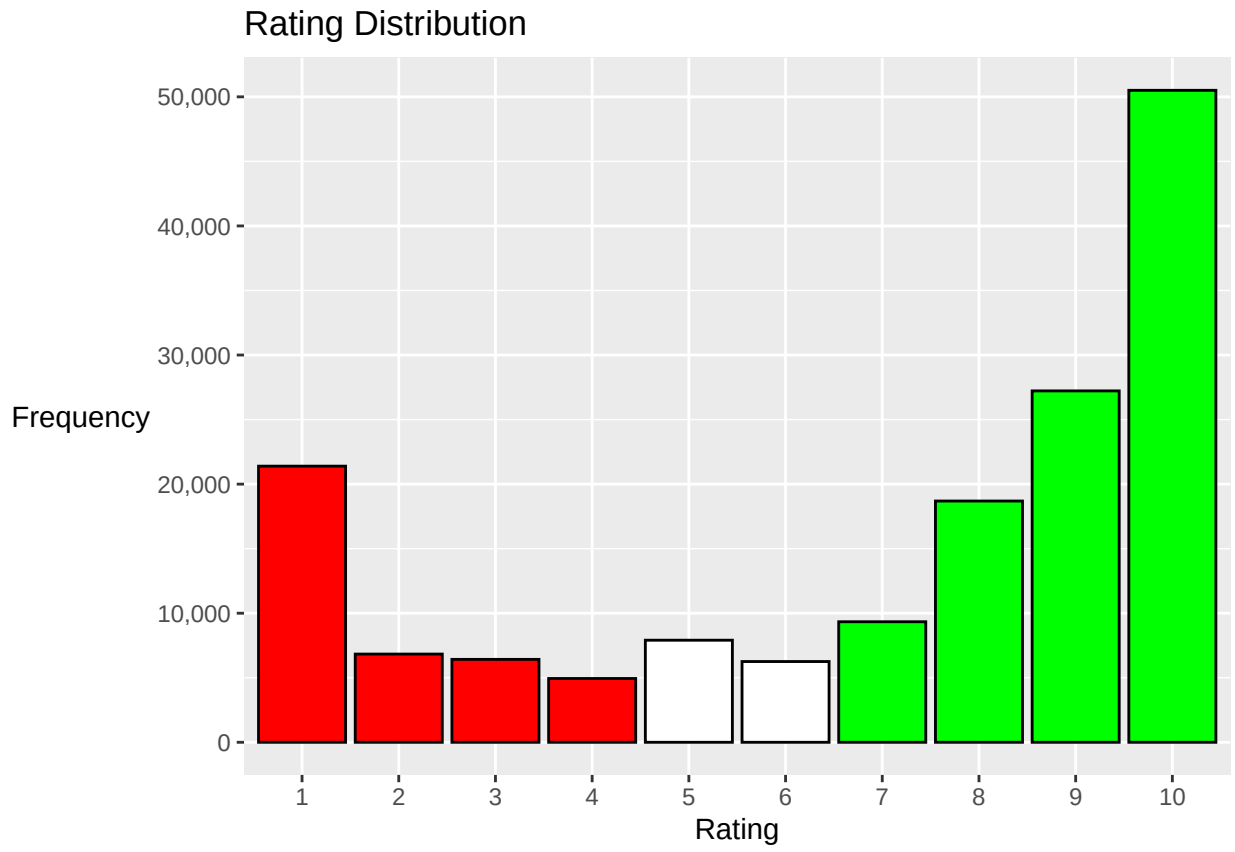
```
data1 <- data1 %>%
    filter(condition != "")
dataClean <- data1
rm(data1)

# 4 Look at the rating distribution
summary(dataClean$rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   8.000   6.997  10.000  10.000
```

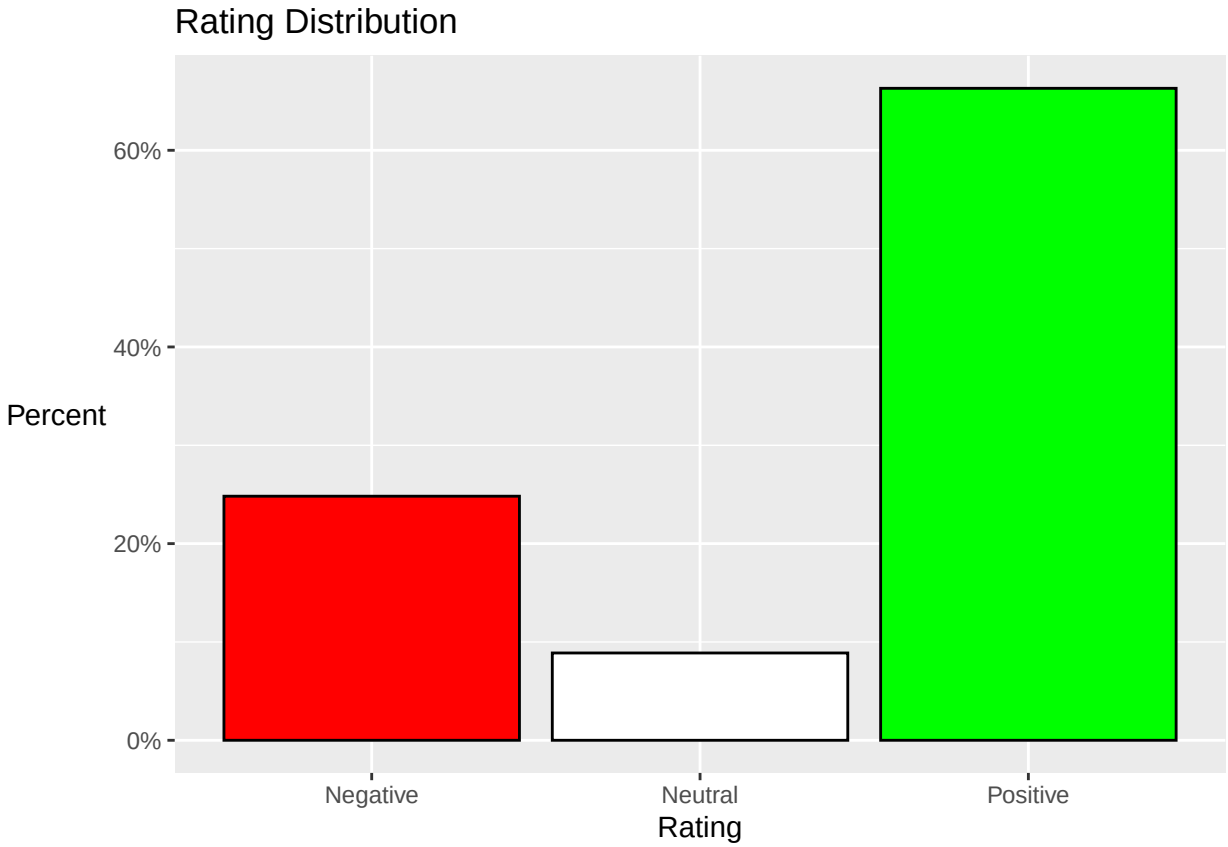
```
table(dataClean$rating)
```

```
##
##     1     2     3     4     5     6     7     8     9    10
## 21391  6833  6422  4942  7907  6254  9338 18688 27219 50504
```

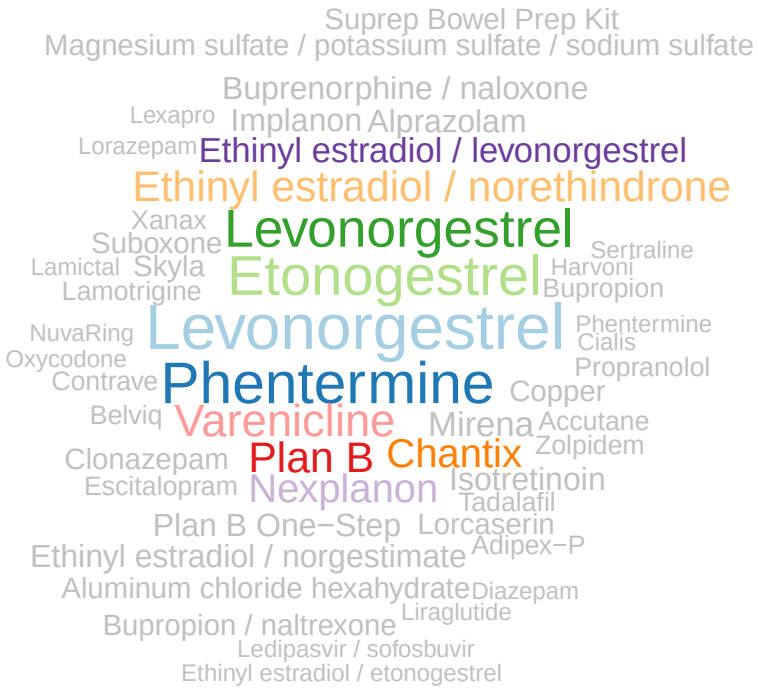
```
# look at numeric rating distribution
tbl <- tibble ( rating=1:10, frequency=table(dataClean$rating) )
ggplot(tbl, aes(x=as.factor(rating), y=frequency) ) +
    geom_col( fill=c( rep("red",4), rep("white",2), rep("green",4) ),
              col="black") +
    scale_y_continuous(labels=scales::comma) +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5)) +
    labs(title="Rating Distribution", x="Rating", y="Frequency")
```




```
# look at 3 tier sentiment
tblMod <- tibble( rating=c("Negative","Neutral","Positive"),
                  frequency=c( sum(tbl$frequency[1:4]),
                               sum(tbl$frequency[5:6]),
                               sum(tbl$frequency[7:10])
                               ) )
ggplot(tblMod, aes(x=rating, y=frequency/sum(frequency) ) ) +
    geom_col( fill=c("red", "white", "green"), col="black" ) +
    scale_y_continuous(labels=scales::percent) +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5)) +
    labs(title="Rating Distribution", x="Rating", y="Percent")
```




```
# 5 make wordclouds for Positive and Negative sentiment
#   for drug/condition combination
# positive words
wcPos <- dataClean %>%
    filter(rating>=7) %>%
    select(drugName,condition,rating) %>%
    group_by(drugName,condition,rating) %>%
    mutate(n=n() ) %>%
    distinct() %>%
    ungroup() %>%
    arrange(desc(rating), desc(n) ) %>%
    slice(1:50)
brewColPos <- c( brewer.pal(n = 10, name = "Paired"),
                 rep("#C0C0C0", (nrow(wcPos)-10) ) )
wcPos %>%
    with( wordcloud(drugName, n, scale=c(2,.5), random.order=F,
                          rot.per=0, colors=brewColPos, ordered.colors=T,
                          fixed.asp=T) )
```




```
wcPos %>%
    with( wordcloud(condition, n, scale=c(2,.5), random.order=F,
                          rot.per=0, colors=brewColPos, ordered.colors=T,
                          fixed.asp=T) )
```


```
# negative words
wcNeg <- dataClean %>%
    filter(rating<=4) %>%
    select(drugName,condition,rating) %>%
    group_by(drugName,condition,rating) %>%
    mutate(n=n() ) %>%
    distinct() %>%
    ungroup() %>%
    arrange( rating, desc(n) ) %>%
    slice(1:50)
brewColNeg <- c( brewer.pal(n = 10, name = "Paired"),
                 rep("#C0C0C0", (nrow(wcNeg)-10) ) )
wcNeg %>% with( wordcloud(drugName, n, scale=c(3,.5), random.order=F,
                          rot.per=0, colors=brewColNeg, ordered.colors=T,
                          fixed.asp=T) )
```

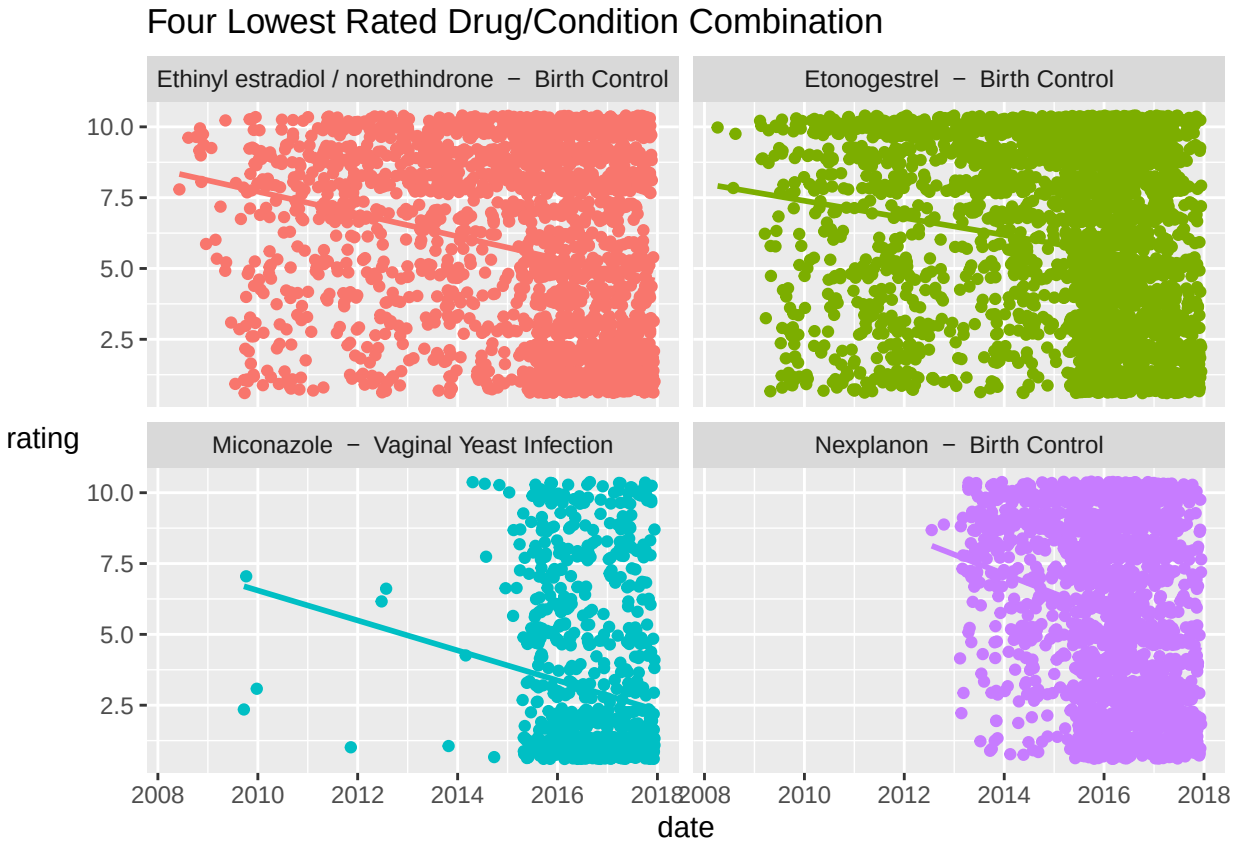
```
## Warning in wordcloud(drugName, n, scale = c(3, 0.5), random.order = F, rot.per =
## 0, : Ethinyl estradiol / norethindrone could not be fit on page. It will not be
## plotted.
```


```
wcNeg %>% with( wordcloud(condition, n, scale=c(3,.5), random.order=F,
                          rot.per=0, colors=brewColNeg, ordered.colors=T,
                          fixed.asp=T) )
```

```
## Warning in wordcloud(condition, n, scale = c(3, 0.5), random.order = F, :
## Vaginal Yeast Infection could not be fit on page. It will not be plotted.
```

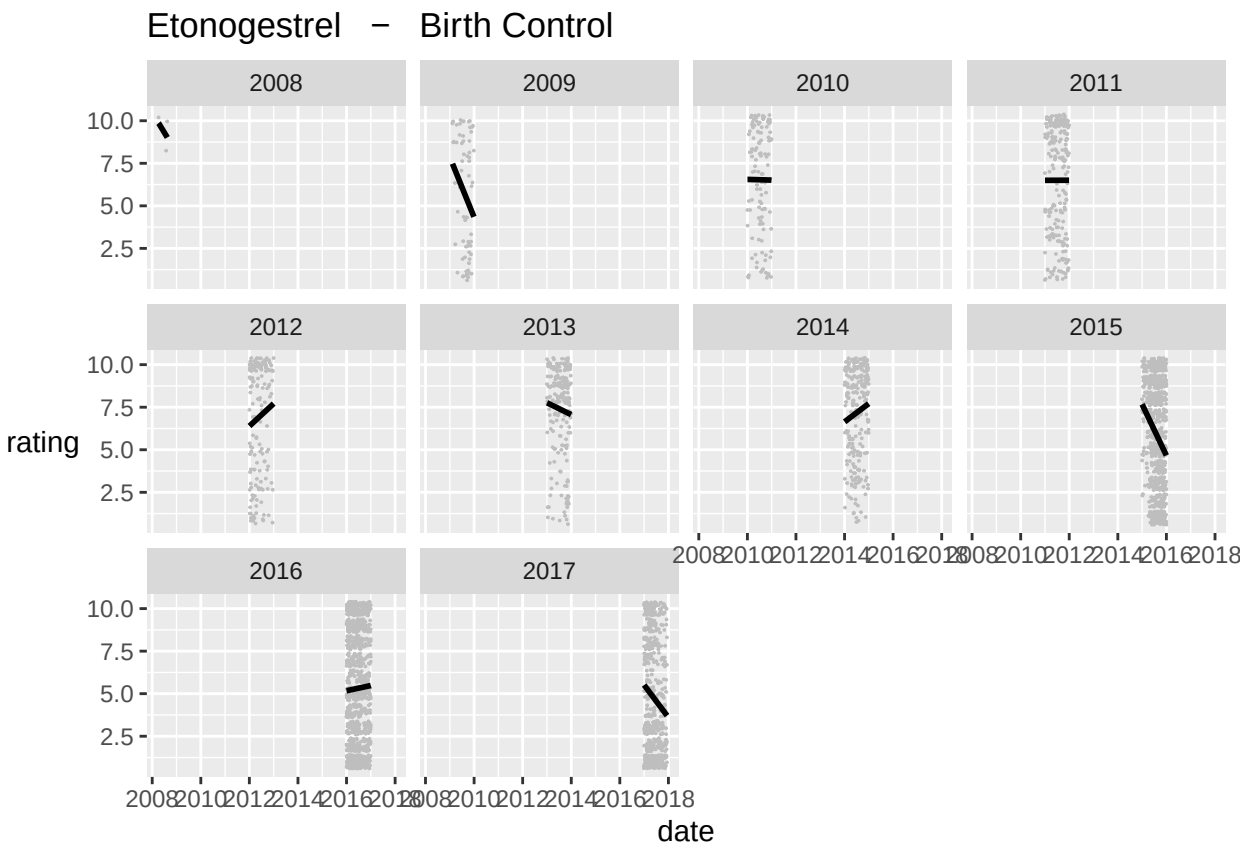

```
# 6 major trend over time for 4 drugs/conditions
majorTrnd1 <- dataClean %>%
    filter(rating<=4) %>%
    select(drugName,condition,rating) %>%
    group_by(drugName,condition,rating) %>%
    mutate(n=n() ) %>%
    distinct() %>%
    ungroup() %>%
    arrange( rating, desc(n) ) %>%
    slice(1:4)
majorTrnd2 <- dataClean %>%
    select(-X1, -review, -usefulCount) %>%
    semi_join(majorTrnd1, by=c("drugName","condition") ) %>%
    unite( "drug_cond", c("drugName","condition"), sep="  -  " )
ggplot(majorTrnd2, aes(date,rating,col=drug_cond) ) +
    geom_jitter() +
    geom_smooth(method = "lm", se=F) +
    facet_wrap( ~ drug_cond ) +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5),
          legend.position="none"
          ) +
    ggtitle("Four Lowest Rated Drug/Condition Combination")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# 7 minor trend over time for 1 drug/condition
minorTrnd1 <- dataClean %>%
    filter( drugName=="Etonogestrel", condition=="Birth Control" ) %>%
    select(-X1,-review,-usefulCount) %>%
    mutate(yr=year(date) )
ggplot(minorTrnd1, aes(date,rating) ) +
    geom_jitter(size=0,col="grey") +
    geom_smooth(method = "lm", se=F, col="black", size = 1) +
    facet_wrap( ~ yr ) +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5)) +
    ggtitle( paste(minorTrnd1$drugName, "  -  ", minorTrnd1$condition) )
```

```
## `geom_smooth()` using formula 'y ~ x'
```

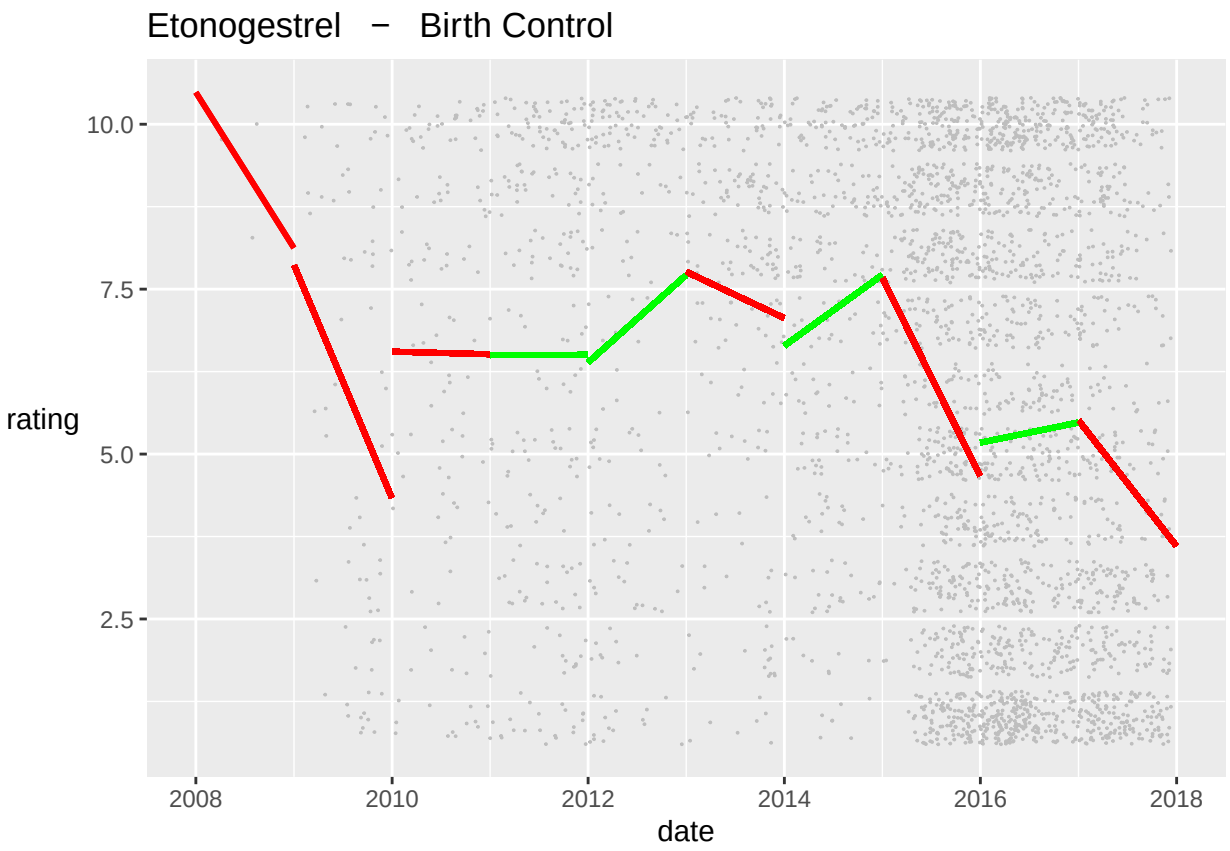



```
# build the models
# function to transform slope/intercept to y coordinates
transform_BM <- function( arg ){
    arg <- unlist(arg)
    dtBegin <- as.numeric ( arg[[1]] )
    dtEnd <- as.numeric ( arg[[2]] )
    b <-  as.numeric ( arg[[3]] )
    m <- as.numeric ( arg[[4]] )
    dt <- as.matrix ( c(dtBegin,dtEnd) )
    coef <- as.matrix ( c(b,m) )
    ycoors <- cbind(1,dt) %*% coef
    return( ycoors )
}
minorMdl1 <- minorTrnd1 %>%
    group_by(yr) %>%
    nest( -drugName, -condition, -yr ) %>%
    mutate(model = map( data, ~lm( rating~date, data=. ) ) ) %>%
    mutate( b = model[[1]][["coefficients"]][["(Intercept)"]],
            m = model[[1]][["coefficients"]][["date"]])
```

```
## Warning: All elements of `...` must be named.
## Did you want `data = c(rating, date)`?
```

```
minorMdl2 <- minorMdl1 %>%
    unnest(data) %>%
```

```
    select(yr,b,m) %>%
    distinct() %>%
    mutate( xFrom= as_date( as.numeric( as_date( paste(yr,"-1-1",sep="") ) ) ),
            xTo= as_date( as.numeric( as_date( paste(yr,"-12-31",sep="") ) ) ) ) )
minorMdl3 <- minorMdl2 %>%
    nest( data=c(xFrom,xTo,b,m) ) %>%
    mutate( coordinates = as.character( map( data , transform_BM ) ) ) %>%
    unnest(data)
minorMdl4 <- minorMdl3 %>%
    mutate( coordinates2 = str_remove_all(coordinates, "[c()]"  ) ) %>%
    separate( coordinates2, c("yFrom","yTo"), sep=",", convert=T )
minorMdlJoin <- minorMdl4 %>%
    select(yr,m,xFrom,xTo,yFrom,yTo)

# the data frame to finally plot
minorTrnd2 <- minorTrnd1 %>%
    left_join(minorMdlJoin, by="yr")

# plot minor trend
minorTrndPlot1 <- ggplot(minorTrnd2, aes(date,rating) ) +
    geom_jitter(size=0, col="grey") +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5)) +
    ggtitle( paste(minorTrnd2$drugName, "  -  ", minorTrnd2$condition) )
minorTrndPlot2 <- minorTrndPlot1 +
    geom_segment( aes(x=xFrom, xend=xTo, y=yFrom, yend=yTo),
                        size=1,
                        col=ifelse(minorTrnd2$m<0,"red","green") )
minorTrndPlot2
```

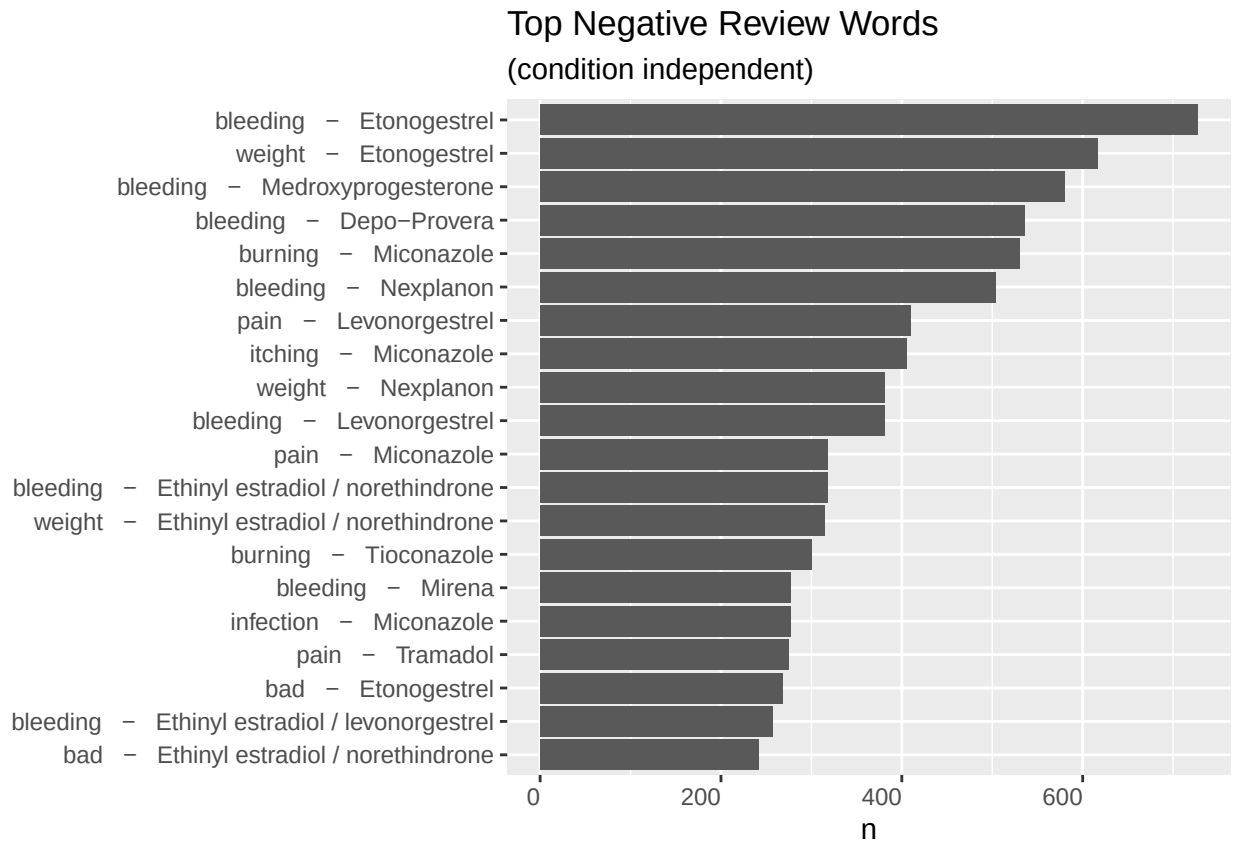



```
# 8 word extraction
# build negative word lexicon
negAfinn <- get_sentiments("afinn") %>%
    filter(value<0) %>%
    #mutate(value=1) %>%
    mutate(lexicon="afinn")
negBing <- get_sentiments("bing") %>%
    filter(sentiment=="negative") %>%
    rename(value=sentiment) %>%
    #mutate(value=1) %>%
    mutate(lexicon="bing") %>%
    anti_join(negAfinn,by="word") # no conflict w/ afinn
negNrc <- get_sentiments("nrc") %>%
    filter(sentiment=="negative") %>%
    rename(value=sentiment) %>%
    #mutate(value=1) %>%
    mutate(lexicon="nrc") %>%
    anti_join(negAfinn,by="word") %>% # no conflict w/ afinn
    anti_join(negBing,by="word") # no conflict w/ bing
negWords <- rbind(negAfinn,negBing,negNrc) %>%
    select(word,lexicon)

# review stats
revStat <-
    dataClean %>%
    select(X1,review) %>%
```

```
    unnest_tokens(word, review) %>%
    group_by(X1) %>%
    mutate(n=n())
summary(revStat$n)
```

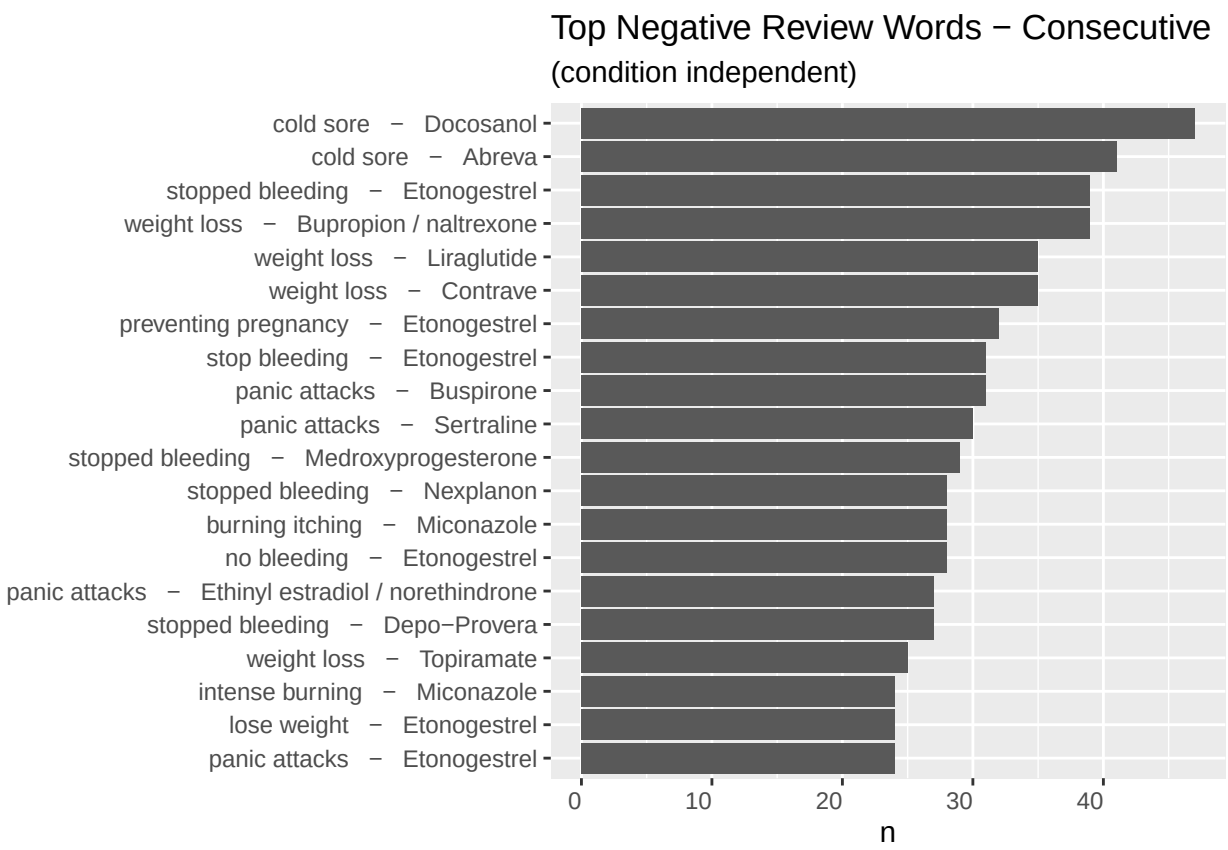
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    84.0   122.0   113.7   143.0  1988.0
```

```
# ngram = 1 analysis
# tokenize all reviews
rev1 <- dataClean %>%
    filter(rating<=4) %>%
    select(drugName,review) %>%
    unnest_tokens(word, review)
# build adverse drug reaction words
adrWords <- rev1 %>%
    inner_join( negWords, by="word" ) %>%
    select(drugName,word) %>%
    group_by(drugName,word) %>%
    mutate(n=n()) %>%
    arrange(desc(n))
# some context filtering
adrWords <- adrWords %>%
    filter( !word=="shot", !word=="no" ) %>%
    arrange(desc(n))
# plot negative words
adrPlot <- adrWords %>%
    select(drugName,word,n) %>%
    group_by(n,word) %>%
    distinct() %>%
    ungroup() %>%
    top_n(20, n) %>%
    arrange(n) %>%
    unite( "cond_drug", c("word","drugName"), sep="   -   " ) %>%
    mutate( plotName = factor(cond_drug, levels = cond_drug) )
ggplot(adrPlot, aes(x=plotName,y=n)  ) +
    geom_col() +
    coord_flip() +
    theme(axis.text.x = element_text(angle = 0, hjust=1),
          axis.title.y=element_blank()
          ) +
    ggtitle(label = "Top Negative Review Words",
            subtitle = "(condition independent)")
```




```
# ngram = 2 analysis
rev2 <- dataClean %>%
    filter(rating<=4) %>%
    select(drugName, review) %>%
    unnest_tokens(ngram, review, token="ngrams", n=2)
adrWords2 <- rev2 %>%
    separate(ngram, c("word1","word2"), sep=" ") %>%
    inner_join( negWords, by=c("word1"="word") ) %>%
    inner_join( negWords, by=c("word2"="word") ) %>%
    unite(words, word1, word2, sep=" ") %>%
    select(drugName,words) %>%
    group_by(drugName,words) %>%
    mutate(n=n()) %>%
    ungroup()
adrPlot2 <- adrWords2 %>%
    select(drugName,words,n) %>%
    group_by(n,words) %>%
    distinct() %>%
    ungroup() %>%
    top_n(20, n) %>%
    arrange(n) %>%
    unite( "cond_drug", c("words","drugName"), sep="   -   " ) %>%
    mutate( plotName = factor(cond_drug, levels = cond_drug) )
ggplot(adrPlot2, aes(x=plotName,y=n)  ) +
    geom_col() +
    coord_flip() +
```

```
    theme(axis.text.x = element_text(angle = 0, hjust=1),
          axis.title.y=element_blank()
    ) +
    ggtitle(label = "Top Negative Review Words - Consecutive",
            subtitle = "(condition independent)")
```




```
#
rev3 <- dataClean %>%
    filter(rating<=4) %>%
    select(drugName, review) %>%
    unnest_tokens(ngram, review, token="ngrams", n=3)
adrWords3 <- rev3 %>%
    separate(ngram, c("word1","word2","word3"), sep=" ") %>%
    inner_join( negWords, by=c("word1"="word") ) %>%
    inner_join( negWords, by=c("word2"="word") ) %>%
    inner_join( negWords, by=c("word3"="word") ) %>%
    unite(words, word1, word2, word3, sep=" ") %>%
    select(drugName,words) %>%
    group_by(drugName,words) %>%
    mutate(n=n()) %>%
    ungroup()
adrPlot3 <- adrWords3 %>%
    select(drugName,words,n) %>%
    group_by(n,words) %>%
    distinct() %>%
```

```
    ungroup() %>%
    top_n(20, n) %>%
    arrange(n) %>%
    unite( "cond_drug", c("words","drugName"), sep="   -   " ) %>%
    mutate( plotName = factor(cond_drug, levels = cond_drug) )
ggplot(adrPlot3, aes(x=plotName,y=n)  ) +
    geom_col() +
    coord_flip() +
    theme(axis.text.x = element_text(angle = 0, hjust=1),
          axis.title.y=element_blank()
    ) +
    ggtitle(label = "Top Negative Review Words - Consecutive",
            subtitle = "(condition independent)")
```