Sentiment analysis on youtube comments

Get data from youtube api

```
library(tuber)
app_id <- "55618119590-d7rnr16n2dq4jqh3hh4amaclvgjgobo5.apps.googleusercontent.com"
api_key <- "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"
yt_oauth(app_id, api_key,token='')
comments1 <- get_all_comments(video_id = "wAZZ-UWGVHI")
comments2 <- get_all_comments(video_id = "FxosOM_Lg9o")
comments3 <- get_all_comments(video_id = "W0QuOku3LRo")
comments4 <- get_all_comments(video_id = "b3x28s61q3c")
comments5 <- get_all_comments(video_id = "4mgePWWCAmA")
comments6 <- get_all_comments(video_id = "kXiYSI7H2b0")
comments7 <- get_all_comments(video_id = "ErMwWXQxHp0")
comments8 <- get_all_comments(video_id = "18fwz9Itbvo")
comments <- rbind(comments1,comments2,comments3,comments4,comments5,comments6,comments7,
comments8)
nrow(comments)
```

```
## [1] 19636
```

```
write.csv(comments, file = "RawVideoComments.csv")
```

```
comments <- read.csv("RawVideoComments.csv", header=T, dec=".",sep=",")
# Load Libraries
library(tm)
```

```
## Loading required package: NLP
```

```
library(plyr)
library(class)
library(caret)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
## Loading required package: lattice
```

```
library(e1071)
library("syuzhet")
library(sentimentr)
```

```
##
## Attaching package: 'sentimentr'
```

```
## The following object is masked from 'package:syuzhet':
##
##     get_sentences
```

```
names(comments)[names(comments) == 'X'] <- 'element_id'
head(comments)
```

```
##   element_id     videoId
## 1          1 wAZZ-UWGVHI
## 2          2 wAZZ-UWGVHI
## 3          3 wAZZ-UWGVHI
## 4          4 wAZZ-UWGVHI
## 5          5 wAZZ-UWGVHI
## 6          6 wAZZ-UWGVHI
##
textDisplay
## 1 I enjoy the convenience of it but frankly I am pretty wary of cashless becoming the
norm. I don&#39;t want to live in a society where owning an expensive device made and co
ntrolled by a corporation is a prerequisite to participate in most aspects of that socie
ty.
## 2
Force humans to change through market power. Now that&#39;s what I call utopia! Great us
e of will power Apple!
## 3
I love apple pay, i never carry a wallet , i might carry my debit card and id and thats
it besides my phone and keys
## 4
Never used any mobile paying thing like google pay or this apple thing, prefer using a r
eal card, not some phone, the physical card feels more like paying.
## 5
I’ll always carry a physical wallet no matter what
## 6
India is far ahead in this matter.
##
textOriginal
## 1 I enjoy the convenience of it but frankly I am pretty wary of cashless becoming the
norm. I don't want to live in a society where owning an expensive device made and contro
lled by a corporation is a prerequisite to participate in most aspects of that society.
## 2
Force humans to change through market power. Now that's what I call utopia! Great use of
will power Apple!
## 3
I love apple pay, i never carry a wallet , i might carry my debit card and id and thats
it besides my phone and keys
## 4
Never used any mobile paying thing like google pay or this apple thing, prefer using a r
eal card, not some phone, the physical card feels more like paying.
## 5
I’ll always carry a physical wallet no matter what
## 6
India is far ahead in this matter.
##   authorDisplayName
## 1           litarea
## 2               L B
## 3          Uh swirv
## 4           Manuzki
## 5             Grant
## 6     Pranshu Anand
##
```

```
authorProfileImageUrl
## 1 https://yt3.ggpht.com/8Qu-vMHYuH6Nt_pmfGHmwCkX7UcUOssipFesiLeCJF6BED8nBg0pS7g0okyAR
x9E2uDYi5seOQ=s48-c-k-c0x00ffffff-no-rj
## 2   https://yt3.ggpht.com/ytc/AMLnZu9x8BA1dZan3i-vPr2L_Qg4phGoVP7fraSvdQA1--1ugVtB_nh
GEGeekJ3AJ0bJ=s48-c-k-c0x00ffffff-no-rj
## 3   https://yt3.ggpht.com/ytc/AMLnZu_Z8YLBHCPZrkR3KXahHoYAXXrsPF9321R5ZsMjVl9OtLheieI
lHs7xDoGa5xHZ=s48-c-k-c0x00ffffff-no-rj
## 4                         https://yt3.ggpht.com/ytc/AMLnZu8tDCbPAb8rWSxTcjcamu4gY6BkU
eEvpomq9zVuAQ=s48-c-k-c0x00ffffff-no-rj
## 5   https://yt3.ggpht.com/MJfnGSYx7A2W2pk2BcYe2qzSJB0weaVro6w7MQIb0yJIOX-fsv1pwZ5Pgdd
X6LVd6gf9ACnW=s48-c-k-c0x00ffffff-no-rj
## 6                         https://yt3.ggpht.com/ytc/AMLnZu9Ou2u97wGQOknuv3r9TUqO3kDBF
OG_FzKmbyDWvw=s48-c-k-c0x00ffffff-no-rj
##                                       authorChannelUrl
## 1 http://www.youtube.com/channel/UCBqSSi0Ms8N88XfIS_bcxkg
## 2 http://www.youtube.com/channel/UCL9PgwF0wSbIsWt0WpjyrBg
## 3 http://www.youtube.com/channel/UCaJqAPEUm92fkQBYxr2Ii2g
## 4 http://www.youtube.com/channel/UC_sbj9GXYaXdsOdvli0CFDg
## 5 http://www.youtube.com/channel/UCnKSWgLO2wZHR9OIIRoheHw
## 6 http://www.youtube.com/channel/UCOzCsf5QJOPQ9WRXwPxgzag
##      authorChannelId.value canRate viewerRating likeCount          publishedAt
## 1 UCBqSSi0Ms8N88XfIS_bcxkg    TRUE         none         0 2022-11-30T22:01:17Z
## 2 UCL9PgwF0wSbIsWt0WpjyrBg    TRUE         none         0 2022-11-30T16:34:25Z
## 3 UCaJqAPEUm92fkQBYxr2Ii2g    TRUE         none         0 2022-11-26T03:42:42Z
## 4 UC_sbj9GXYaXdsOdvli0CFDg    TRUE         none         0 2022-11-25T23:59:58Z
## 5 UCnKSWgLO2wZHR9OIIRoheHw    TRUE         none         0 2022-11-24T19:42:31Z
## 6 UCOzCsf5QJOPQ9WRXwPxgzag    TRUE         none         0 2022-11-16T18:14:46Z
##              updatedAt                            id parentId moderationStatus
## 1 2022-11-30T22:01:17Z Ugxujxvn86PXTJGjvQx4AaABAg     <NA>               NA
## 2 2022-11-30T16:34:25Z UgyA4Q157WmWqWH4qM54AaABAg     <NA>               NA
## 3 2022-11-26T03:42:42Z Ugx3yTghtggUvgzoALJ4AaABAg     <NA>               NA
## 4 2022-11-25T23:59:58Z Ugy9ShdKd9C8-gxgBvt4AaABAg     <NA>               NA
## 5 2022-11-24T19:42:31Z Ugw7ImnT8dlG-sRpsnN4AaABAg     <NA>               NA
## 6 2022-11-16T18:14:46Z UgwDHOTUMZQD4O_fnyx4AaABAg     <NA>               NA
```

Data cleaning using VCorpus

```
# Data cleaning

df.comments.corpus <- VCorpus(VectorSource(comments$textOriginal))
inspect(df.comments.corpus[1:2])
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 2
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 259
##
## [[2]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 106
```

```
# Character representation of a document
#lapply(df.comments.corpus[1:1], as.character)

# Convert to lowercase
df.comments.corpus.lc <- tm_map(df.comments.corpus, content_transformer(tolower))
#lapply(df.comments.corpus.lc[1:1], as.character)

# Remove stop-words
df.comments.corpus.sw <- tm_map(df.comments.corpus.lc, removeWords, stopwords("english"
))
#lapply(df.comments.corpus.sw[1:1], as.character)

# specify your custom stopwords as a character vector
df.comments.corpus.sw <- tm_map(df.comments.corpus.sw, removeWords, c("can", "india", "g
et","linus","just","will","use","one","like","even","video","thing","also","know","year"
))

#Strip whitespace
df.comments.corpus.ws <- tm_map(df.comments.corpus.sw, content_transformer(stripWhitespa
ce))
#lapply(df.comments.corpus.ws[1:1], as.character)

# Remove punctuation
df.comments.corpus.rp <- tm_map(df.comments.corpus.ws, content_transformer(removePunctua
tion))
#lapply(df.comments.corpus.rp[1:1], as.character)

# Text stemming - which reduces words to their root form
df.comments.corpus.ts <- tm_map(df.comments.corpus.rp, content_transformer(stemDocumen
t))
#lapply(df.comments.corpus.rp[1:1], as.character)

df.comments.corpus.clean <- df.comments.corpus.ts

# Convert to dataframe
clean.df <- data.frame(text=unlist(sapply(df.comments.corpus.clean, `[`, "content")),
    stringsAsFactors=F,element_id=comments$element_id, videoId=comments$videoId )
head(clean.df)
```

```
##
text
## 1.content enjoy conveni frank pretti wari cashless becom norm want live societi own e
xpens devic made control corpor prerequisit particip aspect societi
## 2.content
forc human chang market power now call utopia great power appl
## 3.content                                                                    love appl pa
y never carri wallet might carri debit card id that besid phone key
## 4.content                                                                 never use mobil
pay googl pay appl prefer use real card phone physic card feel pay
## 5.content
'll alway carri physic wallet matter
## 6.content
far ahead matter
##           element_id     videoId
## 1.content          1 wAZZ-UWGVHI
## 2.content          2 wAZZ-UWGVHI
## 3.content          3 wAZZ-UWGVHI
## 4.content          4 wAZZ-UWGVHI
## 5.content          5 wAZZ-UWGVHI
## 6.content          6 wAZZ-UWGVHI
```

```
# Build a term-document matrix
TextDoc_dtm <- TermDocumentMatrix(df.comments.corpus.clean)
dtm_m <- as.matrix(TextDoc_dtm)
# Sort by decreasing value of frequency
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
# Display the top 5 most frequent words
head(dtm_d, 10)
```

```
##            word freq
## peopl     peopl 1282
## need       need 1273
## student student 1075
## good       good 1054
## make       make 1023
## use         use 1018
## work       work  983
## love       love  861
## pay         pay  837
## tech       tech  837
```

Word cloud view of comments

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(tidytext)
library(textdata)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(reshape2)
dtm_d %>%
      inner_join(get_sentiments("bing")) %>%
      count(word,freq,sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "freq", fill = 0) %>%
  comparison.cloud(max.words=50)
```

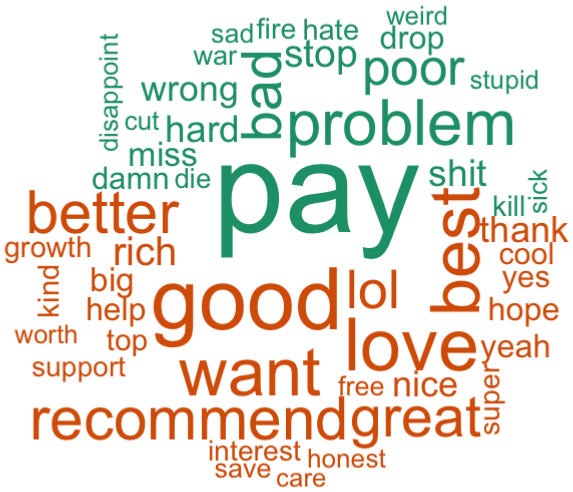
```
## Joining, by = "word"
```

```
## Warning in comparison.cloud(., max.words = 50): recommend could not be fit on
## page. It will not be plotted.
```


```
dtm_d %>%
      inner_join(get_sentiments("afinn")) %>% count(word,freq,value, sort = TRUE) %>%
  acast(word ~ ifelse(value > 0,"positive","negative"), value.var = "freq", fill = 0) %
>%
  comparison.cloud(max.words=50)
```

```
## Joining, by = "word"
```

negative



positive

Sentiment scores with Syuzhet

```
syuzhet_vector <- get_sentiment(clean.df$text, method="syuzhet")
# see the first row of the vector
head(syuzhet_vector)
```

```
## [1] 0.75 0.50 0.65 0.25 0.00 0.80
```

```
# see summary statistics of the vector
mean(syuzhet_vector)
```

```
## [1] 0.315492
```

```
bing_vector <- get_sentiment(clean.df$text, method="bing")
# see the first row of the vector
head(bing_vector)
```

```
## [1] 1 1 1 1 0 0
```

```
# see summary statistics of the vector
mean(bing_vector)
```

```
## [1] 0.2584539
```

```
afinn_vector <- get_sentiment(clean.df$text, method="afinn")
# see the first row of the vector
head(afinn_vector)
```

```
## [1]  3  3  2 -1  1  1
```

```
# see summary statistics of the vector
mean(afinn_vector)
```

```
## [1] 0.8227236
```

```
sentiment.scores.df<- data.frame(syuzhet_vector,bing_vector,afinn_vector)
sentiment.scores.df$element_id <- seq.int(nrow(sentiment.scores.df))
head(sentiment.scores.df)
```

```
##   syuzhet_vector bing_vector afinn_vector element_id
## 1           0.75           1            3          1
## 2           0.50           1            3          2
## 3           0.65           1            2          3
## 4           0.25           1           -1          4
## 5           0.00           0            1          5
## 6           0.80           0            1          6
```

Analyzing the comments as a whole sentence using sentimentr package

```
sentimentr.score <- sentiment(get_sentences(clean.df$text)) %>%
  group_by(element_id) %>%
  summarize(meanSentiment = mean(sentiment))

x <- merge(sentimentr.score, sentiment.scores.df, by = "element_id")
youtube_comments_data <- merge(x, comments, by = "element_id")
#youtube_comments_data$publishedAt <- as.Date(youtube_comments_data$publishedAt, format
="%Y-%m-%d")
youtube_comments_data$year <- format(as.Date(youtube_comments_data$publishedAt, format=
"%Y-%m-%d"),"%Y-%m")

head(youtube_comments_data)
```

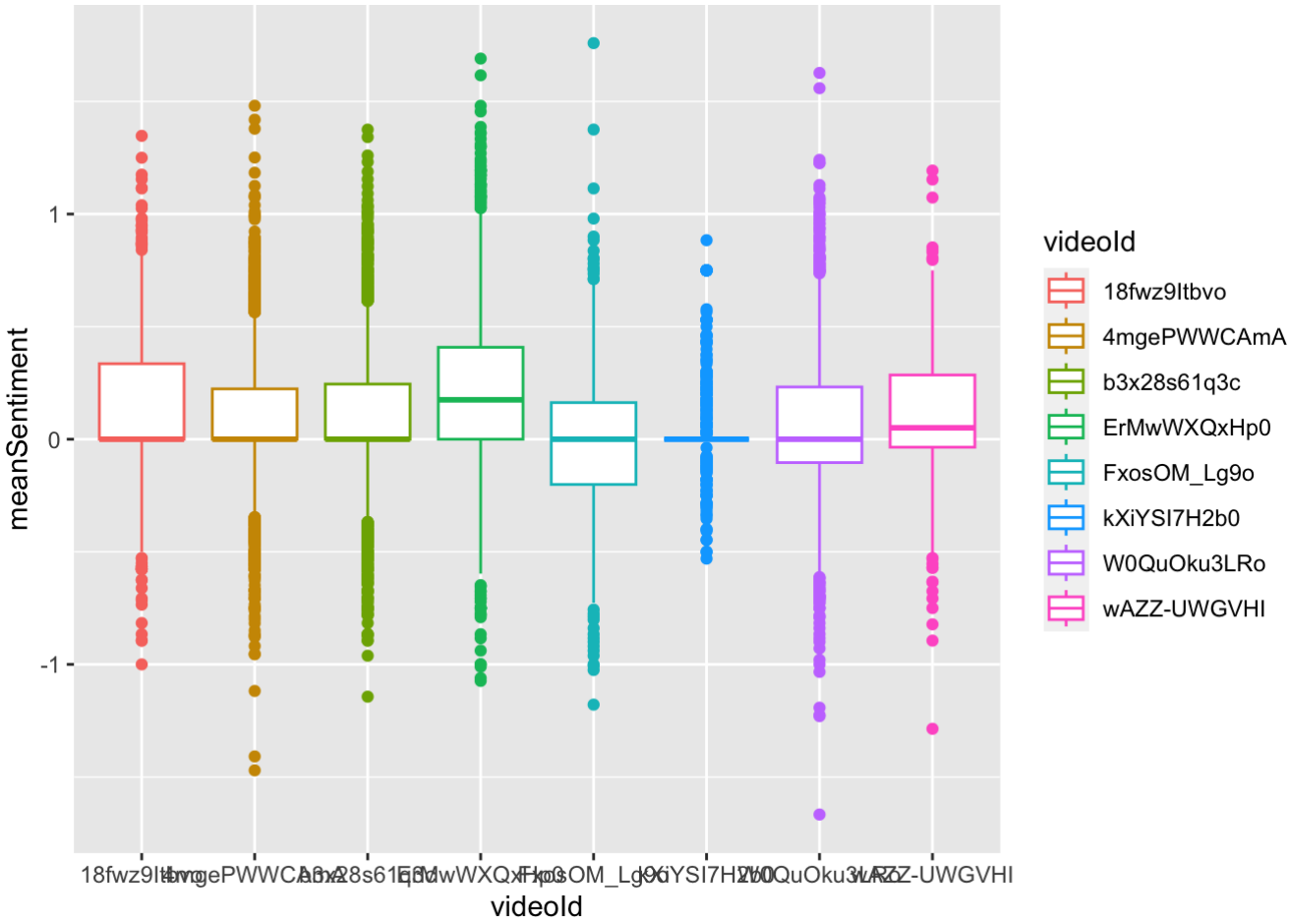
```
##   element_id meanSentiment syuzhet_vector bing_vector afinn_vector     videoId
## 1          1     0.1636634           0.75           1            3 wAZZ-UWGVHI
## 2          2     0.1507557           0.50           1            3 wAZZ-UWGVHI
## 3          3     0.2194691           0.65           1            2 wAZZ-UWGVHI
## 4          4     0.0700000           0.25           1           -1 wAZZ-UWGVHI
## 5          5     0.0000000           0.00           0            1 wAZZ-UWGVHI
## 6          6     0.4618802           0.80           0            1 wAZZ-UWGVHI
##
textDisplay
## 1 I enjoy the convenience of it but frankly I am pretty wary of cashless becoming the
norm. I don&#39;t want to live in a society where owning an expensive device made and co
ntrolled by a corporation is a prerequisite to participate in most aspects of that socie
ty.
## 2
Force humans to change through market power. Now that&#39;s what I call utopia! Great us
e of will power Apple!
## 3
I love apple pay, i never carry a wallet , i might carry my debit card and id and thats
it besides my phone and keys
## 4
Never used any mobile paying thing like google pay or this apple thing, prefer using a r
eal card, not some phone, the physical card feels more like paying.
## 5
I’ll always carry a physical wallet no matter what
## 6
India is far ahead in this matter.
##
textOriginal
## 1 I enjoy the convenience of it but frankly I am pretty wary of cashless becoming the
norm. I don't want to live in a society where owning an expensive device made and contro
lled by a corporation is a prerequisite to participate in most aspects of that society.
## 2
Force humans to change through market power. Now that's what I call utopia! Great use of
will power Apple!
## 3
I love apple pay, i never carry a wallet , i might carry my debit card and id and thats
it besides my phone and keys
## 4
Never used any mobile paying thing like google pay or this apple thing, prefer using a r
eal card, not some phone, the physical card feels more like paying.
## 5
I’ll always carry a physical wallet no matter what
## 6
India is far ahead in this matter.
##   authorDisplayName
## 1           litarea
## 2               L B
## 3          Uh swirv
## 4           Manuzki
## 5             Grant
## 6     Pranshu Anand
##
```

```
authorProfileImageUrl
## 1 https://yt3.ggpht.com/8Qu-vMHYuH6Nt_pmfGHmwCkX7UcUOssipFesiLeCJF6BED8nBg0pS7g0okyAR
x9E2uDYi5seOQ=s48-c-k-c0x00ffffff-no-rj
## 2   https://yt3.ggpht.com/ytc/AMLnZu9x8BA1dZan3i-vPr2L_Qg4phGoVP7fraSvdQA1--1ugVtB_nh
GEGeekJ3AJ0bJ=s48-c-k-c0x00ffffff-no-rj
## 3   https://yt3.ggpht.com/ytc/AMLnZu_Z8YLBHCPZrkR3KXahHoYAXXrsPF9321R5ZsMjVl9OtLheieI
lHs7xDoGa5xHZ=s48-c-k-c0x00ffffff-no-rj
## 4                         https://yt3.ggpht.com/ytc/AMLnZu8tDCbPAb8rWSxTcjcamu4gY6BkU
eEvpomq9zVuAQ=s48-c-k-c0x00ffffff-no-rj
## 5   https://yt3.ggpht.com/MJfnGSYx7A2W2pk2BcYe2qzSJB0weaVro6w7MQIb0yJIOX-fsv1pwZ5Pgdd
X6LVd6gf9ACnW=s48-c-k-c0x00ffffff-no-rj
## 6                         https://yt3.ggpht.com/ytc/AMLnZu9Ou2u97wGQOknuv3r9TUqO3kDBF
OG_FzKmbyDWvw=s48-c-k-c0x00ffffff-no-rj
##                                       authorChannelUrl
## 1 http://www.youtube.com/channel/UCBqSSi0Ms8N88XfIS_bcxkg
## 2 http://www.youtube.com/channel/UCL9PgwF0wSbIsWt0WpjyrBg
## 3 http://www.youtube.com/channel/UCaJqAPEUm92fkQBYxr2Ii2g
## 4 http://www.youtube.com/channel/UC_sbj9GXYaXdsOdvli0CFDg
## 5 http://www.youtube.com/channel/UCnKSWgLO2wZHR9OIIRoheHw
## 6 http://www.youtube.com/channel/UCOzCsf5QJOPQ9WRXwPxgzag
##      authorChannelId.value canRate viewerRating likeCount          publishedAt
## 1 UCBqSSi0Ms8N88XfIS_bcxkg    TRUE         none         0 2022-11-30T22:01:17Z
## 2 UCL9PgwF0wSbIsWt0WpjyrBg    TRUE         none         0 2022-11-30T16:34:25Z
## 3 UCaJqAPEUm92fkQBYxr2Ii2g    TRUE         none         0 2022-11-26T03:42:42Z
## 4 UC_sbj9GXYaXdsOdvli0CFDg    TRUE         none         0 2022-11-25T23:59:58Z
## 5 UCnKSWgLO2wZHR9OIIRoheHw    TRUE         none         0 2022-11-24T19:42:31Z
## 6 UCOzCsf5QJOPQ9WRXwPxgzag    TRUE         none         0 2022-11-16T18:14:46Z
##              updatedAt                            id parentId moderationStatus
## 1 2022-11-30T22:01:17Z Ugxujxvn86PXTJGjvQx4AaABAg     <NA>               NA
## 2 2022-11-30T16:34:25Z UgyA4Q157WmWqWH4qM54AaABAg     <NA>               NA
## 3 2022-11-26T03:42:42Z Ugx3yTghtggUvgzoALJ4AaABAg     <NA>               NA
## 4 2022-11-25T23:59:58Z Ugy9ShdKd9C8-gxgBvt4AaABAg     <NA>               NA
## 5 2022-11-24T19:42:31Z Ugw7ImnT8dlG-sRpsnN4AaABAg     <NA>               NA
## 6 2022-11-16T18:14:46Z UgwDHOTUMZQD4O_fnyx4AaABAg     <NA>               NA
##      year
## 1 2022-11
## 2 2022-11
## 3 2022-11
## 4 2022-11
## 5 2022-11
## 6 2022-11
```

```
# plot of sentiment over time & automatically choose a method to model the change
ggplot(youtube_comments_data, aes(x = as.Date(publishedAt), y = meanSentiment)) +
  geom_point(aes(color = videoId))+ # add points to our plot, color-coded by president
  geom_smooth(method = "auto") # pick a method & fit a model
```

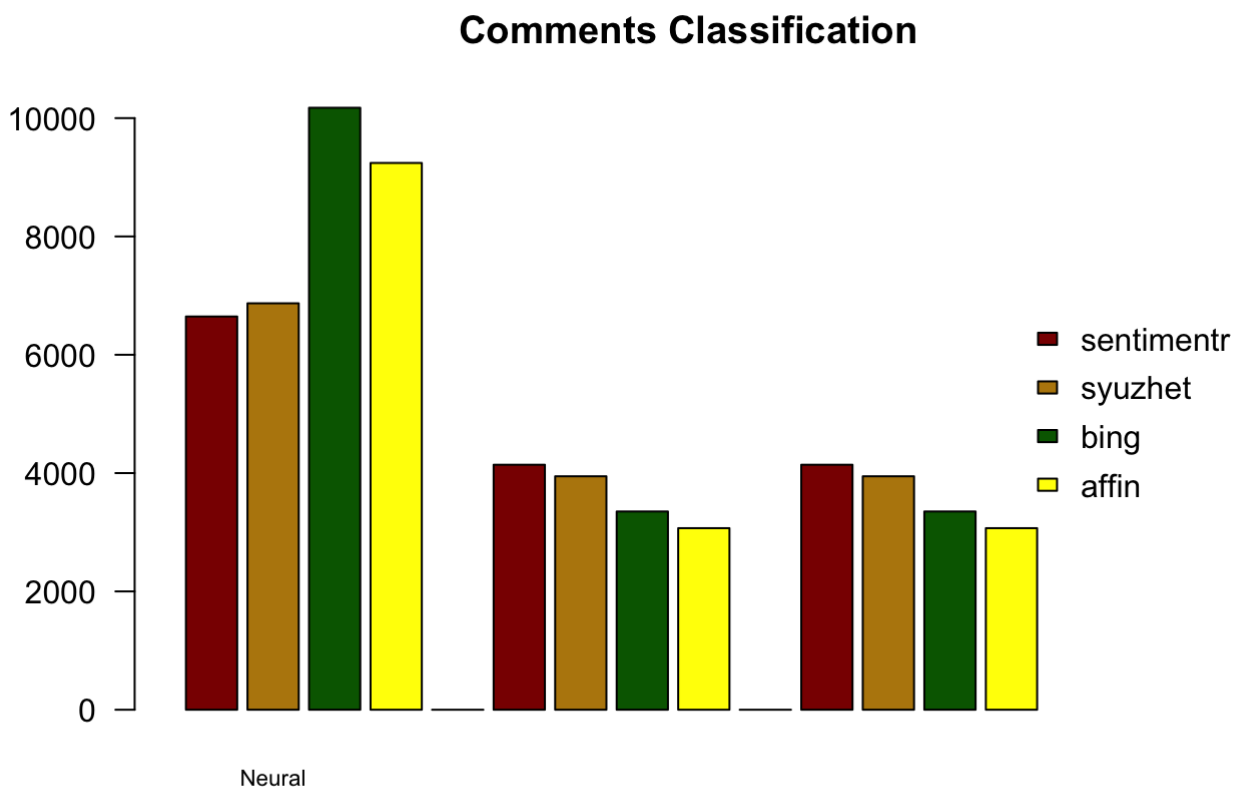
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```


```
# plot of sentiment by president
ggplot(youtube_comments_data, aes(x = videoId, y = meanSentiment, color = videoId)) +
  geom_boxplot() # draw a boxplot for each president
```



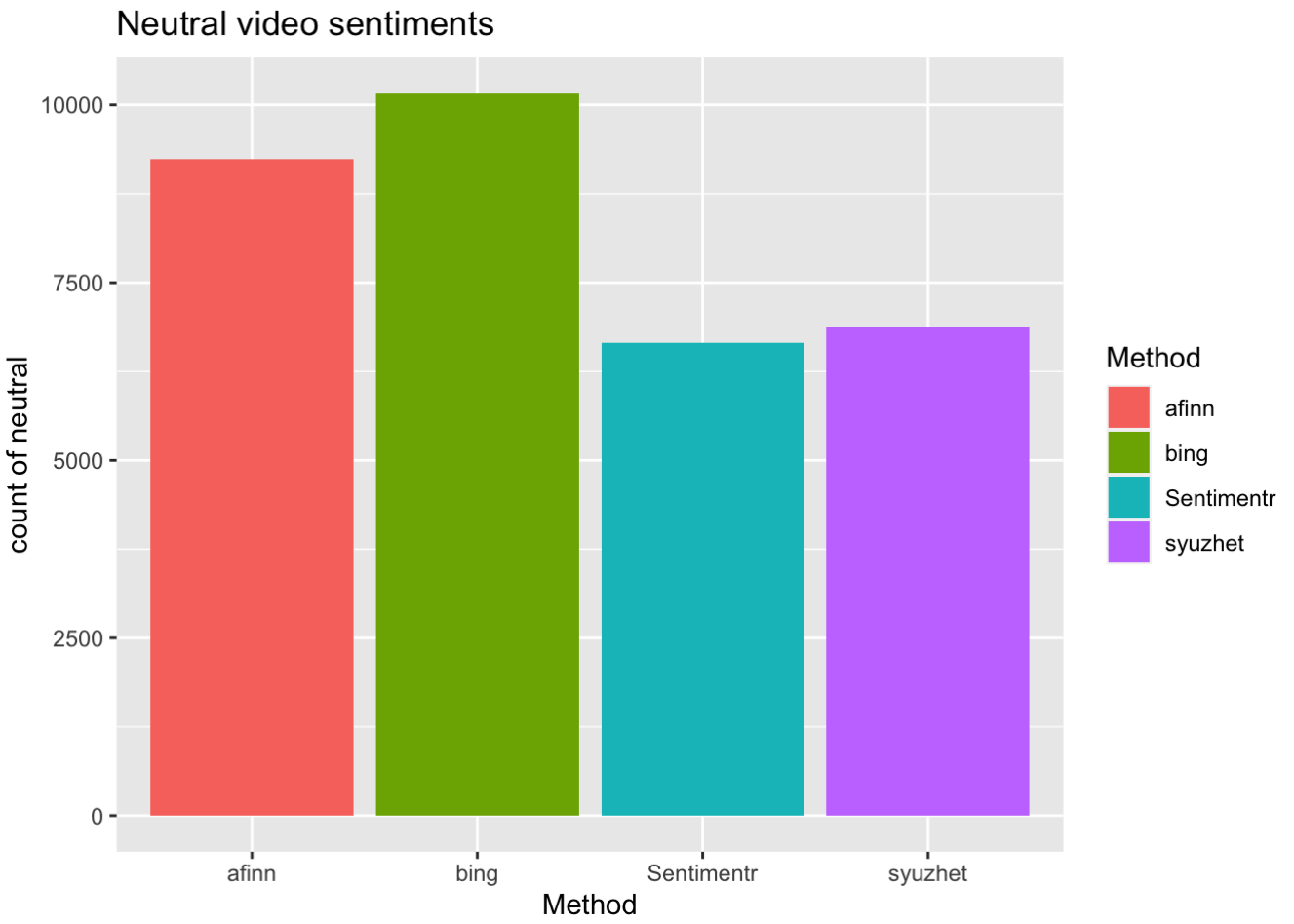
Comparision of sentiment scores of sentimentr package, syuzhet, bing and affin

```
sentimentr <- c("Sentimentr", min(youtube_comments_data$meanSentiment), max(youtube_comm
ents_data$meanSentiment),sum(youtube_comments_data$meanSentiment < 0),sum(youtube_commen
ts_data$meanSentiment == 0),sum(youtube_comments_data$meanSentiment > 0))
syuzhet <- c("syuzhet", min(youtube_comments_data$syuzhet_vector), max(youtube_comments_
data$syuzhet_vector),sum(youtube_comments_data$syuzhet_vector < 0),sum(youtube_comments_
data$syuzhet_vector == 0),sum(youtube_comments_data$syuzhet_vector > 0))
bing <- c("bing", min(youtube_comments_data$bing_vector),max(youtube_comments_data$bing_
vector),sum(youtube_comments_data$bing_vector < 0),sum(youtube_comments_data$bing_vector
== 0),sum(youtube_comments_data$bing_vector > 0))
afinn <- c("afinn",  min(youtube_comments_data$afinn_vector),max(youtube_comments_data$a
finn_vector),sum(youtube_comments_data$afinn_vector < 0),sum(youtube_comments_data$afinn
_vector == 0),sum(youtube_comments_data$afinn_vector > 0))

compare.scores <- rbind(sentimentr,syuzhet,bing,afinn)
colnames(compare.scores) <- c("Method", "Most negative score", "Most positive score","Nu
m of Negative","Num of Neutral","Num of Positive")
compare.scores <- data.frame(compare.scores)
compare.scores$Num.of.Neutral <- as.integer(compare.scores$Num.of.Neutral)
compare.scores$Num.of.Negative <- as.integer(compare.scores$Num.of.Negative)
compare.scores$Num.of.Positive <- as.integer(compare.scores$Num.of.Positive)
barplot(c(as.integer(compare.scores$Num.of.Neutral),0,as.integer(compare.scores$Num.of.N
egative),0,as.integer(compare.scores$Num.of.Negative))
,names.arg = c(" " ,"Neural" , " ", " "," ", "","Negative", " "," "," "," ","Positive ",
" "," "),col = c("darkred", "darkgoldenrod", "darkgreen","yellow", "darkred", "darkred",
"darkgoldenrod", "darkgreen","yellow", "darkred", "darkred","darkgoldenrod", "darkgreen"
,"yellow"),main = "Comments Classification",legend=TRUE,cex.names=0.7, las=1,beside = TR
UE,xlim = c(0, 20))
opar =par(oma = c(0,0,0,0), mar = c(0,0,0,0), new = TRUE)
legend(x = "right", legend = c("sentimentr","syuzhet","bing","affin"), fill = c("darkre
d", "darkgoldenrod", "darkgreen","yellow"), bty = "n",y.intersp = 2)
```

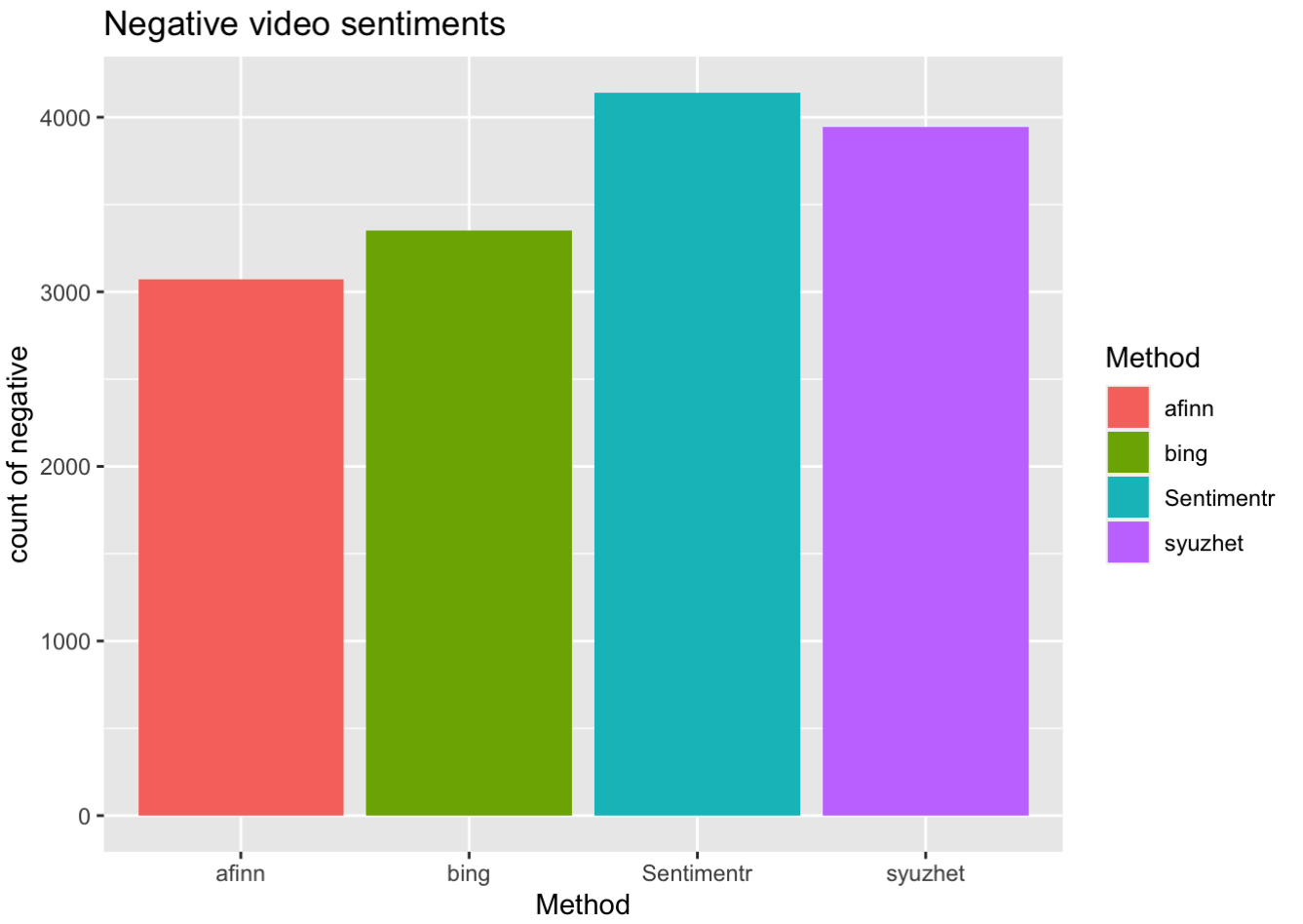



```
par(opar) # Reset par



quickplot(Method, data=compare.scores, weight=Num.of.Neutral, geom="bar", fill=Method, y
lab="count of neutral",)+ggtitle("Neutral video sentiments")
```

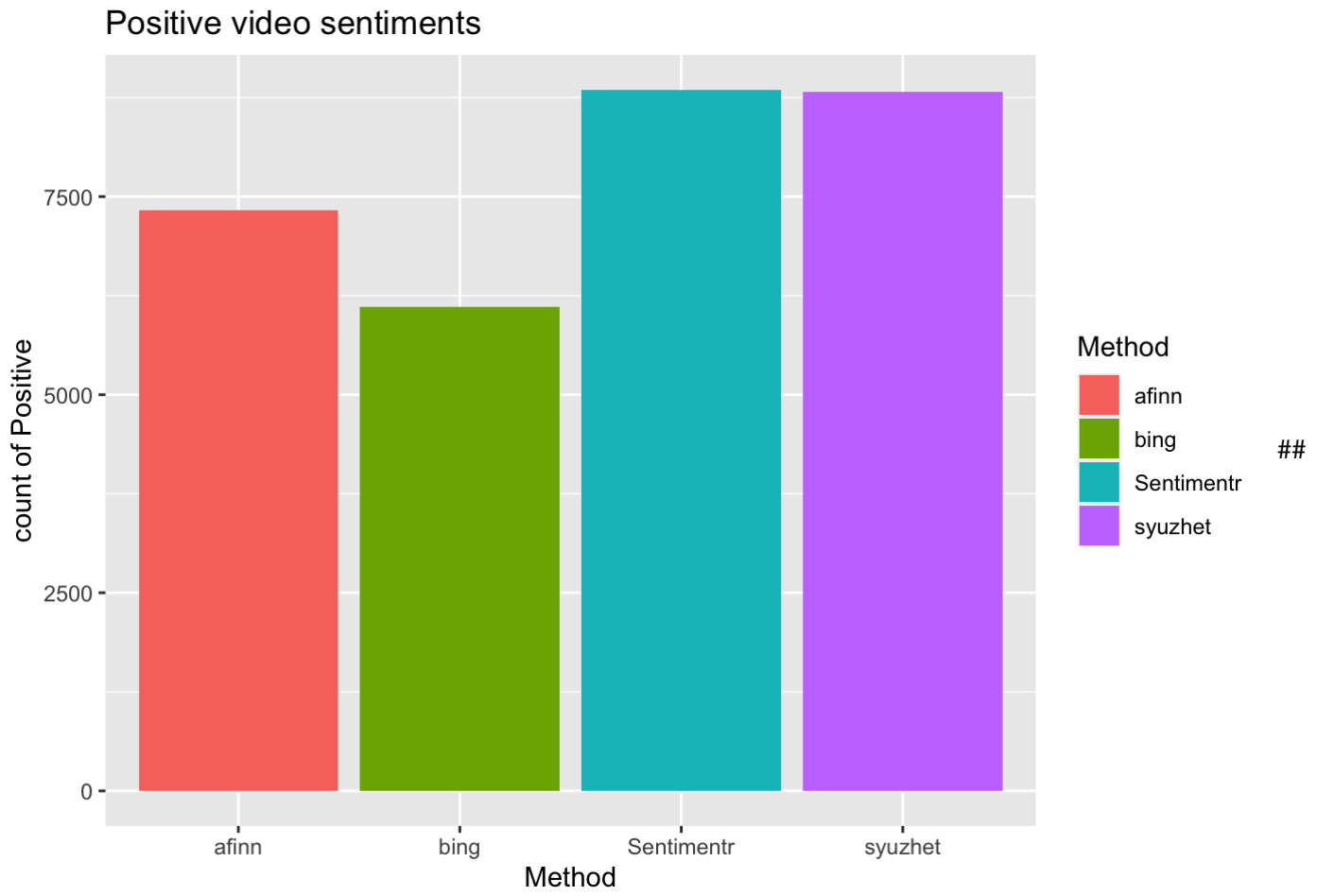
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```




```
quickplot(Method, data=compare.scores, weight=Num.of.Negative, geom="bar", fill=Method,
 ylab="count of negative")+ggtitle("Negative video sentiments")
```




```
quickplot(Method, data=compare.scores, weight=Num.of.Positive, geom="bar", fill=Method,
 ylab="count of Positive")+ggtitle("Positive video sentiments")
```



##

Analyzing the Sentiment for a video id

```
score.video <- youtube_comments_data %>%
  group_by(videoId) %>%
  summarize(videoSentimentr = mean(meanSentiment), videoSyuzhetSentiment = mean(syuzhet_
vector),videoBingSentiment = mean(bing_vector),videoAfinnSentiment = mean(afinn_vector))

score.video
```

```
## # A tibble: 8 × 5
##   videoId     videoSentimentr videoSyuzhetSentiment videoBingSentiment videoAf…¹
##   <chr>                 <dbl>                 <dbl>              <dbl>      <dbl>
## 1 18fwz9Itbvo          0.134                 0.355              0.265      1.06
## 2 4mgePWWCAmA          0.0695                0.202              0.205      0.565
## 3 b3x28s61q3c          0.0764                0.229              0.127      0.639
## 4 ErMwWXQxHp0          0.204                 0.631              0.747      1.88
## 5 FxosOM_Lg9o         -0.0105               -0.0376            -0.239     -0.244
## 6 kXiYSI7H2b0          0.0171                0.0343             0.0121     0.134
## 7 W0QuOku3LRo          0.0587                0.256             -0.0803     0.130
## 8 wAZZ-UWGVHI          0.104                 0.343              0.196      0.483
## # … with abbreviated variable name ¹videoAfinnSentiment
```

Emotion classification is done using NRC Word-Emotion Association Lexicon (aka EmoLex). The get_nrc_sentiments function returns a data frame with each row representing a sentence from the original file.

```
FxosOM_Lg9o<-get_nrc_sentiment(clean.df[clean.df$videoId=="FxosOM_Lg9o",]$text)
```

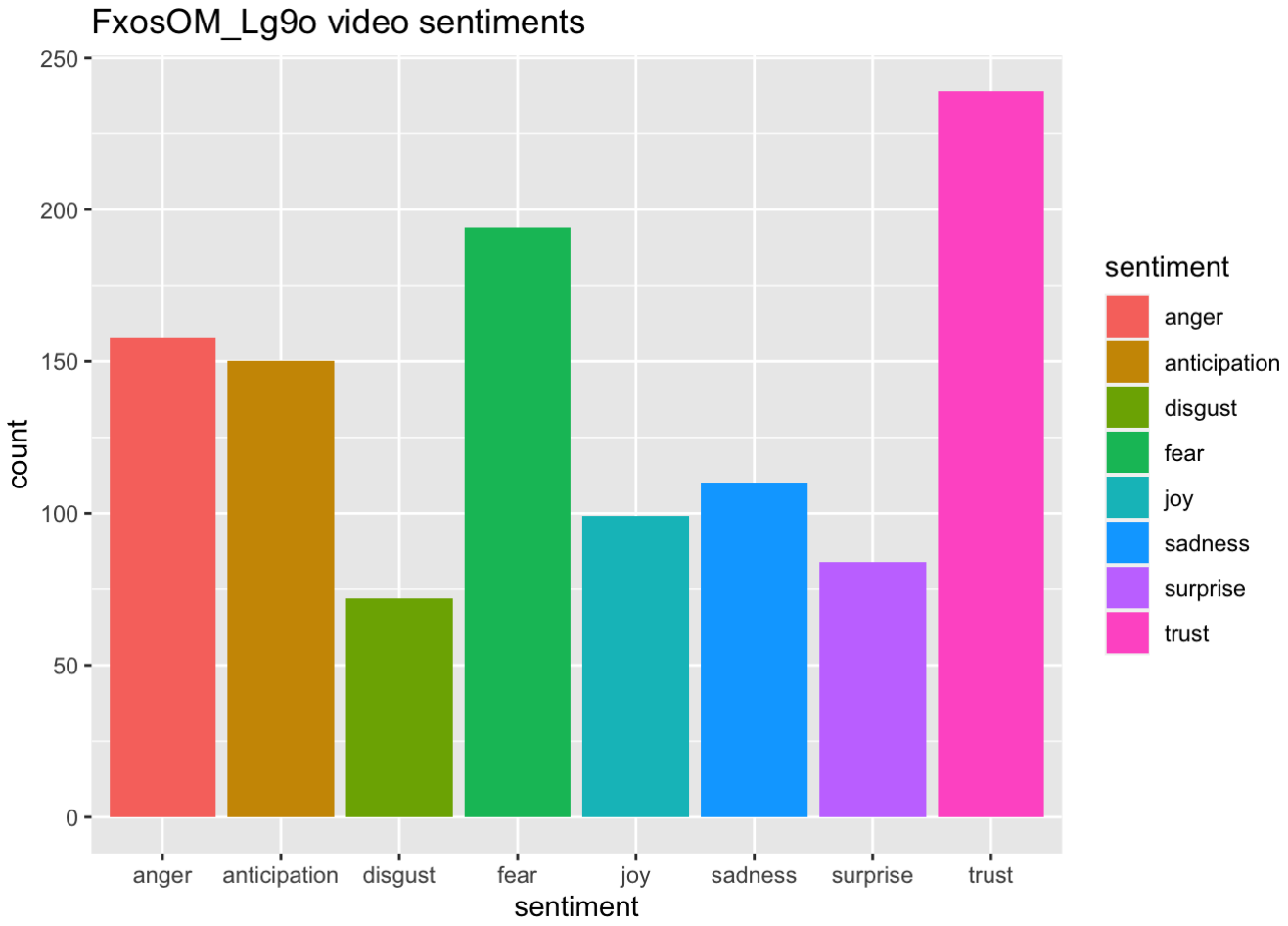
```
## Warning: `spread_()` was deprecated in tidyr 1.2.0.
## ℹ Please use `spread()` instead.
## ℹ The deprecated feature was likely used in the syuzhet package.
##   Please report the issue to the authors.
```

```
# head(d,10) - to see top 10 lines of the get_nrc_sentiment dataframe
head (FxosOM_Lg9o,10)
```

```
##    anger anticipation disgust fear joy sadness surprise trust negative positive
## 1      0            1       0    1   0       0        0     0        1        0
## 2      1            1       0    3   1       1        1     1        4        3
## 3      4            1       2    2   2       1        1     2        3        3
## 4      0            0       0    0   0       0        0     0        0        0
## 5      2            1       2    5   1       1        3     1        5        2
## 6      1            0       0    1   0       0        0     0        1        0
## 7      2            0       0    3   0       0        0     1        3        0
## 8      0            3       0    2   0       1        1     2        3        0
## 9      0            0       1    0   0       0        0     1        1        1
## 10     1            0       1    1   0       0        1     0        1        0
```

Visualize emotions for a video with negative sentiment scores.

```
#transpose
td<-data.frame(t(FxosOM_Lg9o))
#The function rowSums computes column sums across rows for each level of a grouping vari
able.
td_new <- data.frame(rowSums(td[2:253]))
#Transformation and cleaning
names(td_new)[1] <- "count"
td_new <- cbind("sentiment" = rownames(td_new), td_new)
rownames(td_new) <- NULL
td_new2<-td_new[1:8,]
#Plot One - count of words associated with each sentiment
quickplot(sentiment, data=td_new2, weight=count, geom="bar", fill=sentiment, ylab="coun
t")+ggtitle("FxosOM_Lg9o video sentiments")
```



Kaggle comments analysis with different sentiment analysis packages

```
kaggle.comments <- read.csv("Kaggle/comments.csv", header=T, dec=".",sep=",")
kaggle.comments$element_id <- seq.int(nrow(kaggle.comments))
names(kaggle.comments)[names(kaggle.comments) == 'Video.ID'] <- 'videoId'
head(kaggle.comments)
```

```
##   X     videoId
## 1 0 wAZZ-UWGVHI
## 2 1 wAZZ-UWGVHI
## 3 2 wAZZ-UWGVHI
## 4 3 wAZZ-UWGVHI
## 5 4 wAZZ-UWGVHI
## 6 5 wAZZ-UWGVHI
##
Comment
## 1
Let's not forget that Apple Pay in 2014 required a brand new iPhone in order to use it. A significant portion of Apple's user base wasn't able to use it even if they wanted to. As each successive iPhone incorporated the technology and older iPhones were replaced th e number of people who could use the technology increased.
## 2
Here in NZ 50% of retailers don’t even have contactless credit card machines like pay-wa ve which support Apple Pay. They don’t like the high fees that come with these.
## 3
I will forever acknowledge this channel with the help of your lessons and ideas explanat ions, Now It's quite helpful while you'll just sit at your comfort and monitor your acco unt Growth.
## 4 Whenever I go to a place that doesn’t take Apple Pay (doesn’t happen too often), i t’s such a drag. Between ‘contactless Covid’ habits and my getting the Apple Card, I’ve gotten so used to Apple Pay that I get seriously annoyed when a store doesn’t take it. I t feels like a shock, it’s crazy how quickly it took over my shopping routine! I’ve offi cially been brainwashed by Apple because now it feels so inconvenient to even carry a ph ysical card in my pocket.
## 5
Apple Pay is so convenient, secure, and easy to use. I used it while at the Korean and J apanese airports, no need for physical credit cards.
## 6
We’ve been hounding my bank to adopt Apple pay. I understand why they don’t want to do i t with the extra fees, but its just so easy and quick at the checkout.
##   Likes Sentiment element_id
## 1    95         1          1
## 2    19         0          2
## 3   161         2          3
## 4     8         0          4
## 5    34         2          5
## 6     8         1          6
```

```
# Data cleaning

kaggle.comments.corpus <- VCorpus(VectorSource(kaggle.comments$Comment))
inspect(kaggle.comments.corpus[1:2])
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 2
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 325
##
## [[2]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 167
```

```
# Convert to lowercase
kaggle.comments.corpus.lc <- tm_map(kaggle.comments.corpus, content_transformer(tolowe
r))

# Remove stop-words
kaggle.comments.corpus.sw <- tm_map(kaggle.comments.corpus.lc, removeWords, stopwords("e
nglish"))
#lapply(df.comments.corpus.sw[1:1], as.character)

# specify your custom stopwords as a character vector
kaggle.comments.corpus.sw <- tm_map(kaggle.comments.corpus.sw, removeWords, c("can", "in
dia", "get","linus","just","will","use","one","like","even","video","thing","also","kno
w","year"))

#Strip whitespace
kaggle.comments.corpus.ws <- tm_map(kaggle.comments.corpus.sw, content_transformer(strip
Whitespace))

# Remove punctuation
kaggle.comments.corpus.rp <- tm_map(kaggle.comments.corpus.ws, content_transformer(remov
ePunctuation))


# Text stemming - which reduces words to their root form
kaggle.comments.corpus.ts <- tm_map(kaggle.comments.corpus.rp, content_transformer(stemD
ocument))

kaggle.comments.corpus.clean <- kaggle.comments.corpus.ts

# Convert to dataframe
clean.df <- data.frame(text=unlist(sapply(kaggle.comments.corpus.clean, `[`, "content"
)),
    stringsAsFactors=F,element_id=kaggle.comments$element_id, videoId=kaggle.comments$vi
deoId )
head(clean.df)
```

```
##
text
## 1.content
forget appl pay 2014 requir brand new iphon order signific portion appl user base abl wa
nt success iphon incorpor technolog older iphon replac number peopl technolog increas
## 2.content
nz 50 retail don’t contactless credit card machin paywav support appl pay don’t high fee
come
## 3.content
forev acknowledg channel help lesson idea explan now quit help sit comfort monitor accou
nt growth
## 4.content whenev go place doesn’t take appl pay doesn’t happen often ’s drag ‘contact
less covid’ habit get appl card ’ve gotten use appl pay serious annoy store doesn’t take
feel shock ’s crazi quick took shop routin ’ve offici brainwash appl now feel inconveni
carri physic card pocket
## 5.content
appl pay conveni secur easi use korean japanes airport need physic credit card
## 6.content
’ve hound bank adopt appl pay understand don’t want extra fee easi quick checkout
##           element_id     videoId
## 1.content          1 wAZZ-UWGVHI
## 2.content          2 wAZZ-UWGVHI
## 3.content          3 wAZZ-UWGVHI
## 4.content          4 wAZZ-UWGVHI
## 5.content          5 wAZZ-UWGVHI
## 6.content          6 wAZZ-UWGVHI
```

Sentiment scores with Syuzhet

Convert all the vectors to same scale using sign (this converts values to between -1 and 1)

```
syuzhet_vector <- get_sentiment(clean.df$text, method="syuzhet")
syuzhet_vector <- sign(syuzhet_vector)
# see the first row of the vector
head(syuzhet_vector)
```

```
## [1]  1  1  1 -1  1  1
```

```
# see summary statistics of the vector
mean(syuzhet_vector)
```

```
## [1] 0.4891086
```

```
bing_vector <- get_sentiment(clean.df$text, method="bing")
bing_vector <- sign(bing_vector)
# see the first row of the vector
head(bing_vector)
```

```
## [1]  1  1  1 -1  0  0
```

```
# see summary statistics of the vector
mean(bing_vector)
```

```
## [1] 0.3524363
```

```
afinn_vector <- get_sentiment(clean.df$text, method="afinn")
afinn_vector <- sign(afinn_vector)
# see the first row of the vector
head(afinn_vector)
```

```
## [1]  1  1  1 -1 -1  1
```

```
# see summary statistics of the vector
mean(afinn_vector)
```

```
## [1] 0.4478244
```

```
sentiment.scores.df<- data.frame(syuzhet_vector,bing_vector,afinn_vector)
sentiment.scores.df$element_id <- seq.int(nrow(sentiment.scores.df))
head(sentiment.scores.df)
```

```
##   syuzhet_vector bing_vector afinn_vector element_id
## 1              1           1            1          1
## 2              1           1            1          2
## 3              1           1            1          3
## 4             -1          -1           -1          4
## 5              1           0           -1          5
## 6              1           0            1          6
```

```
sentimentr.score <- sentiment(get_sentences(clean.df$text)) %>%
  group_by(element_id) %>%
  summarize(meanSentiment = sign(mean(sentiment)))
```

```
x <- merge(sentimentr.score, sentiment.scores.df, by = "element_id")
kaggle.data <- merge(kaggle.comments,x, by = "element_id")
kaggle.data$meanSentiment <- with(kaggle.data, ifelse(meanSentiment == 0, 1, ifelse(mean
Sentiment > 0,2,0)))
kaggle.data$syuzhet_vector <- with(kaggle.data, ifelse(syuzhet_vector == 0, 1, ifelse(sy
uzhet_vector > 0,2,0)))
kaggle.data$bing_vector <- with(kaggle.data, ifelse(bing_vector == 0, 1, ifelse(bing_vec
tor > 0,2,0)))
kaggle.data$afinn_vector <- with(kaggle.data, ifelse(afinn_vector == 0, 1, ifelse(afinn_
vector > 0,2,0)))
head(kaggle.data)
```

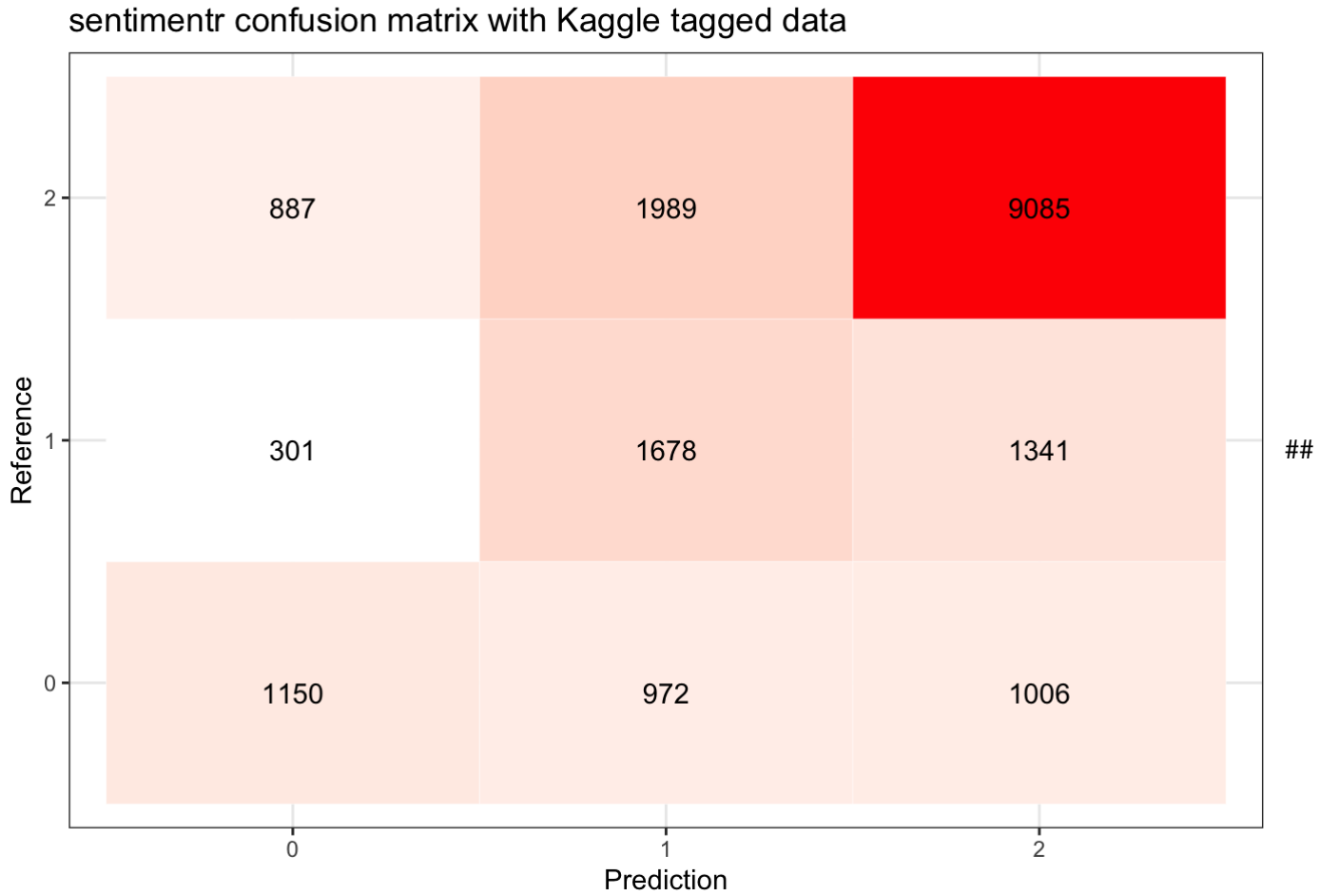
```
##   element_id X     videoId
## 1          1 0 wAZZ-UWGVHI
## 2          2 1 wAZZ-UWGVHI
## 3          3 2 wAZZ-UWGVHI
## 4          4 3 wAZZ-UWGVHI
## 5          5 4 wAZZ-UWGVHI
## 6          6 5 wAZZ-UWGVHI
##
Comment
## 1
Let's not forget that Apple Pay in 2014 required a brand new iPhone in order to use it. A significant portion of Apple's user base wasn't able to use it even if they wanted to. As each successive iPhone incorporated the technology and older iPhones were replaced th e number of people who could use the technology increased.
## 2
Here in NZ 50% of retailers don’t even have contactless credit card machines like pay-wa ve which support Apple Pay. They don’t like the high fees that come with these.
## 3
I will forever acknowledge this channel with the help of your lessons and ideas explanat ions, Now It's quite helpful while you'll just sit at your comfort and monitor your acco unt Growth.
## 4 Whenever I go to a place that doesn’t take Apple Pay (doesn’t happen too often), i t’s such a drag. Between ‘contactless Covid’ habits and my getting the Apple Card, I’ve gotten so used to Apple Pay that I get seriously annoyed when a store doesn’t take it. I t feels like a shock, it’s crazy how quickly it took over my shopping routine! I’ve offi cially been brainwashed by Apple because now it feels so inconvenient to even carry a ph ysical card in my pocket.
## 5
Apple Pay is so convenient, secure, and easy to use. I used it while at the Korean and J apanese airports, no need for physical credit cards.
## 6
We’ve been hounding my bank to adopt Apple pay. I understand why they don’t want to do i t with the extra fees, but its just so easy and quick at the checkout.
##   Likes Sentiment meanSentiment syuzhet_vector bing_vector afinn_vector
## 1    95         1             2              2           2             2
## 2    19         0             0              2           2             2
## 3   161         2             2              2           2             2
## 4     8         0             0              0           0             0
## 5    34         2             2              2           1             0
## 6     8         1             2              2           1             2
```

Confusion matrix for scores using sentimentr

```
cm <- confusionMatrix(data = as.factor(kaggle.data$Sentiment), reference = as.factor(kag gle.data$meanSentiment))
data.frame(cm$table)
```

```
##   Prediction Reference Freq
## 1          0         0 1150
## 2          1         0  972
## 3          2         0 1006
## 4          0         1  301
## 5          1         1 1678
## 6          2         1 1341
## 7          0         2  887
## 8          1         2 1989
## 9          2         2 9085
```

```
ggplot(data =  data.frame(cm$table), mapping = aes(x = Prediction, y = Reference)) +
  geom_tile(aes(fill = Freq), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red") +
  theme_bw() + theme(legend.position = "none")+
  ggtitle("sentimentr confusion matrix with Kaggle tagged data")
```

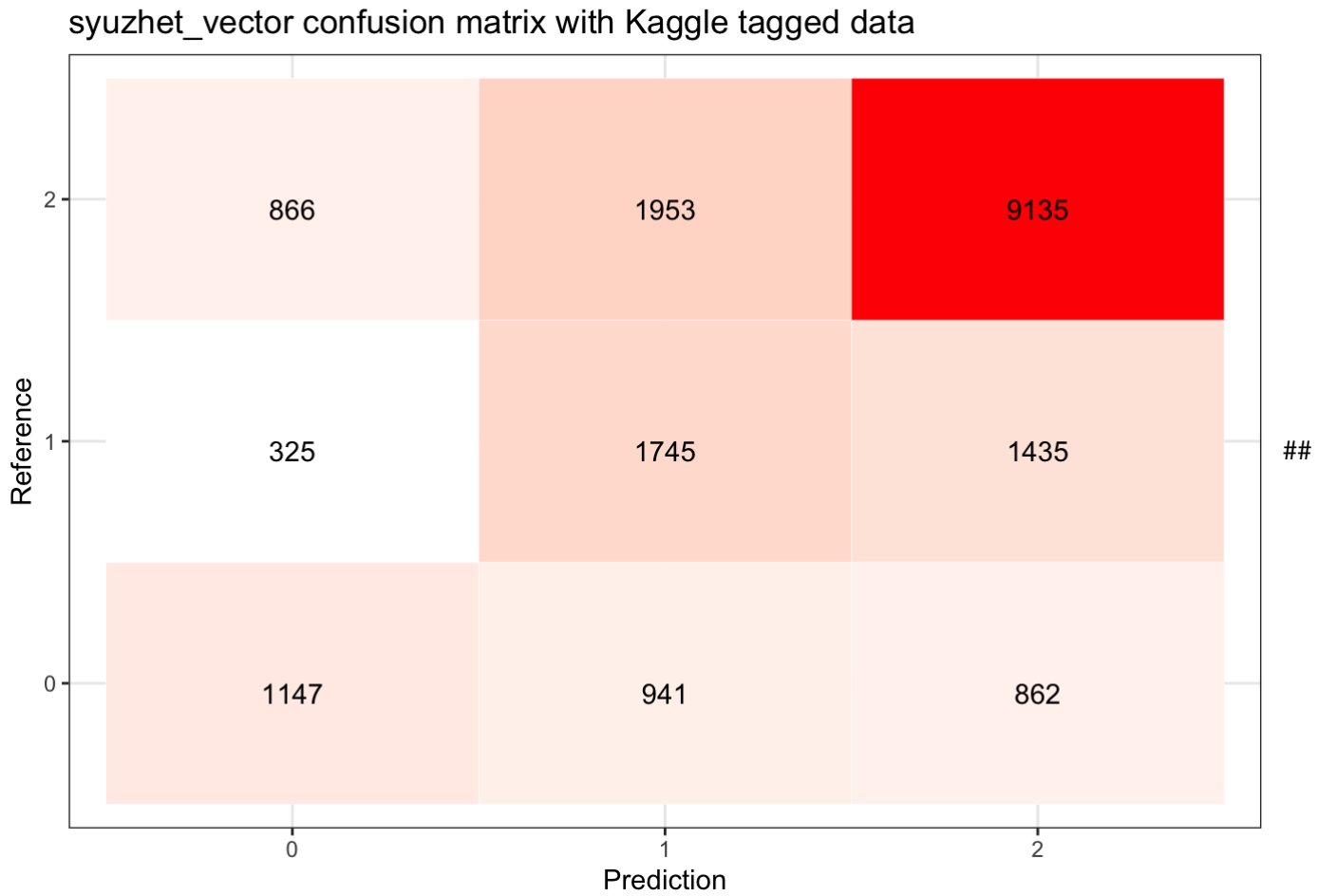


Confusion matrix for scores using syuzhet_vector

```
cm <-confusionMatrix(data = as.factor(kaggle.data$Sentiment), reference = as.factor(kagg
le.data$syuzhet_vector))
data.frame(cm$table)
```

```
##   Prediction Reference Freq
## 1          0         0 1147
## 2          1         0  941
## 3          2         0  862
## 4          0         1  325
## 5          1         1 1745
## 6          2         1 1435
## 7          0         2  866
## 8          1         2 1953
## 9          2         2 9135
```

```
ggplot(data =  data.frame(cm$table), mapping = aes(x = Prediction, y = Reference)) +
  geom_tile(aes(fill = Freq), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red") +
  theme_bw() + theme(legend.position = "none")+
  ggtitle("syuzhet_vector confusion matrix with Kaggle tagged data")
```

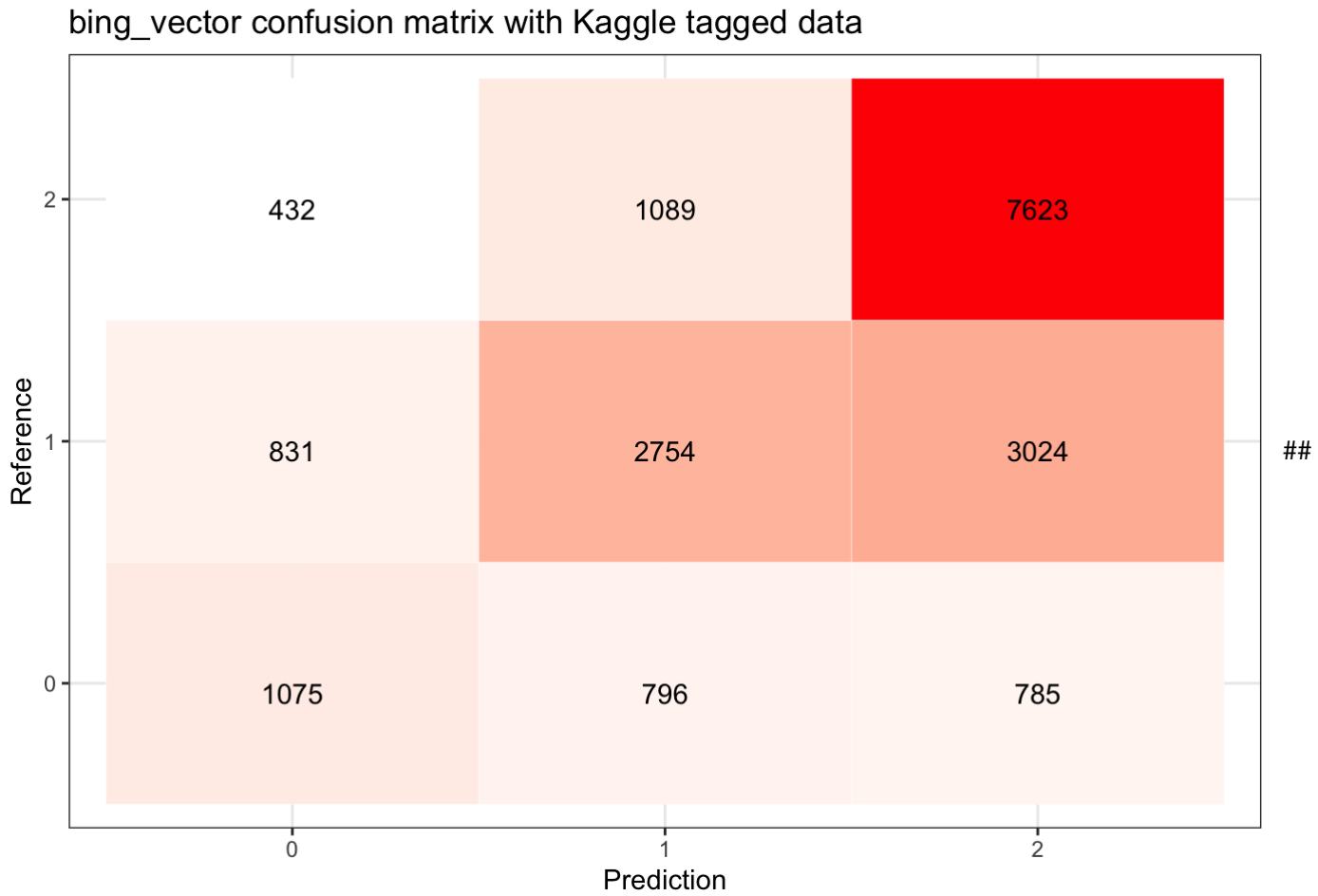


Confusion matrix for scores using bing_vector

```
cm <- confusionMatrix(data = as.factor(kaggle.data$Sentiment), reference = as.factor(kag
gle.data$bing_vector))
data.frame(cm$table)
```

```
##   Prediction Reference Freq
## 1          0         0 1075
## 2          1         0  796
## 3          2         0  785
## 4          0         1  831
## 5          1         1 2754
## 6          2         1 3024
## 7          0         2  432
## 8          1         2 1089
## 9          2         2 7623
```

```
ggplot(data =  data.frame(cm$table), mapping = aes(x = Prediction, y = Reference)) +
  geom_tile(aes(fill = Freq), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red") +
  theme_bw() + theme(legend.position = "none")+
  ggtitle("bing_vector confusion matrix with Kaggle tagged data")
```

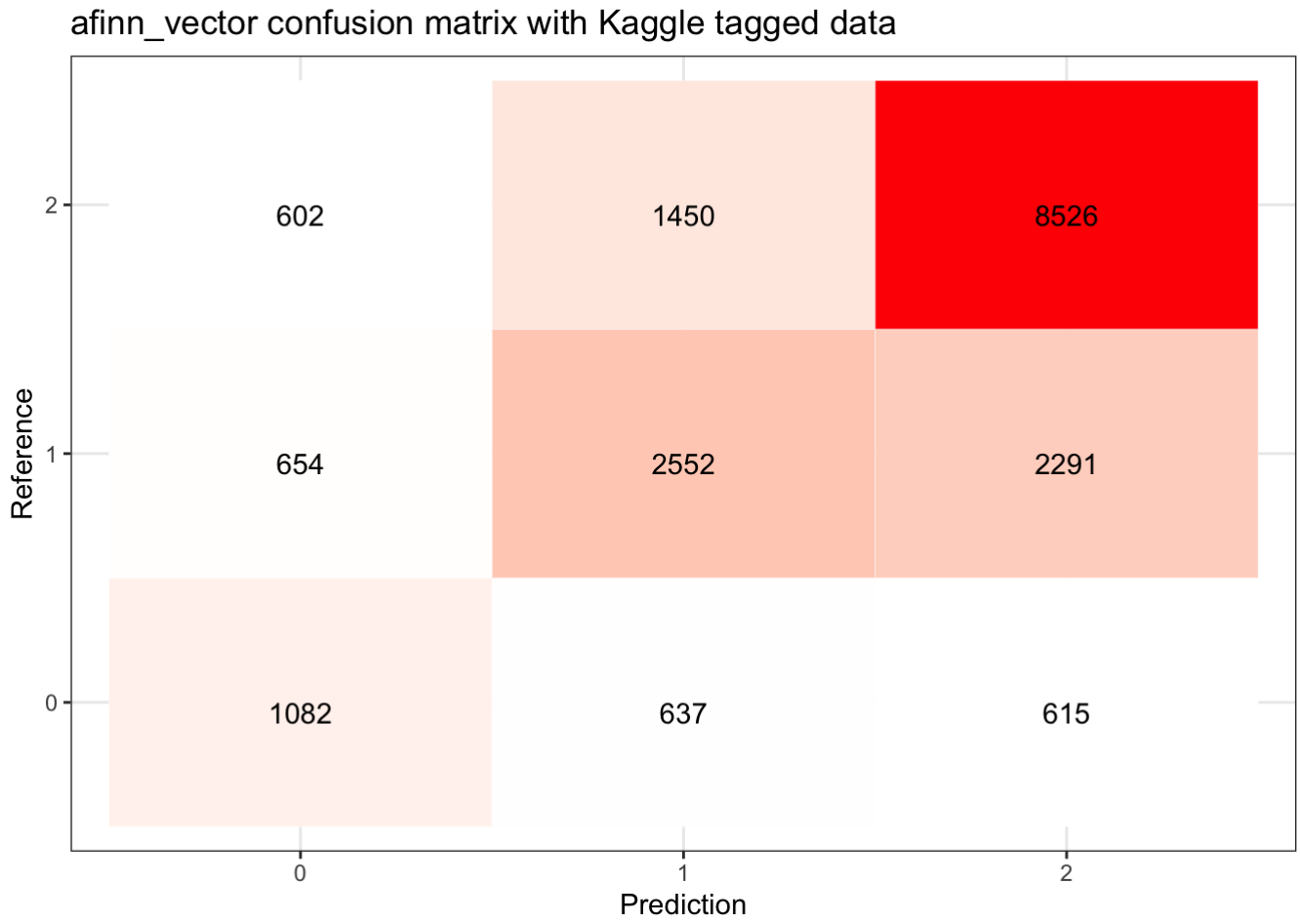


Confusion matrix for scores using afinn_vector

```
cm <- confusionMatrix(data = as.factor(kaggle.data$Sentiment), reference = as.factor(kag
gle.data$afinn_vector))
data.frame(cm$table)
```

```
##   Prediction Reference Freq
## 1          0         0 1082
## 2          1         0  637
## 3          2         0  615
## 4          0         1  654
## 5          1         1 2552
## 6          2         1 2291
## 7          0         2  602
## 8          1         2 1450
## 9          2         2 8526
```

```
ggplot(data =  data.frame(cm$table), mapping = aes(x = Prediction, y = Reference)) +
  geom_tile(aes(fill = Freq), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red") +
  theme_bw() + theme(legend.position = "none")+
  ggtitle("afinn_vector confusion matrix with Kaggle tagged data")
```




```
library(topicmodels)
library(quanteda)
```

```
## Package version: 3.2.3
## Unicode version: 14.0
## ICU version: 70.1
```

```
## Parallel computing: 8 of 8 threads used.
```

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

```
##
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:tm':
##
##     stopwords
```

```
## The following objects are masked from 'package:NLP':
##
##     meta, meta<-
```

```
fulltext <- corpus(kaggle.comments$Comment)
dtm <- dfm(fulltext, # input text
tolower = TRUE, stem = TRUE, # set lowercasing and stemming to TRUE
remove = stopwords("english")) # provide the stopwords for deletion
```

```
## Warning: 'dfm.corpus()' is deprecated. Use 'tokens()' first.
```

```
## Warning: 'remove' is deprecated; use dfm_remove() instead
```

```
## Warning: 'stem' is deprecated; use dfm_wordstem() instead
```

```
doc_freq <- docfreq(dtm) # document frequency per term (column)
dtm <- dtm[, doc_freq >= 2] # select terms with doc_freq >= 2
dtm <- dfm_weight(dtm, "prop") # weight the features using prop
docvars(dtm, "sentiment_class") <- kaggle.comments$Sentiment
train_dtm <- dfm_sample(dtm, size = 12000)
test_dtm <- dtm[setdiff(docnames(dtm), docnames(train_dtm)),]
```

textmodel_nb Naive Bayes classifier for texts. Fit a multinomial or Bernoulli Naive Bayes model, given a dfm and some training labels.

```
library("quanteda.textmodels")
# fit a Naive Bayes multinomial model and use it to predict the test data
nb_model <- textmodel_nb(train_dtm, y = docvars(train_dtm, "sentiment_class"),distributi
on = "Bernoulli", prior = "docfreq")
pred_nb <- predict(nb_model, newdata = test_dtm)

# compare prediction (rows) and actual is_prewar value (columns) in a table
table(prediction = pred_nb, sentiment_class = docvars(test_dtm, "sentiment_class"))
```

```
##           sentiment_class
## prediction    0    1    2
##          0  250  180  163
##          1   30  390  132
##          2  561 1032 3671
```

```
cm <- confusionMatrix(data = as.factor(docvars(test_dtm, "sentiment_class")), reference
 = as.factor(pred_nb))
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1    2
##          0  250   30  561
##          1  180  390 1032
##          2  163  132 3671
##
## Overall Statistics
##
##                Accuracy : 0.6726
##                  95% CI : (0.661, 0.6841)
##     No Information Rate : 0.8213
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2854
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2
## Sensitivity           0.42159  0.70652   0.6974
## Specificity           0.89838  0.79307   0.7424
## Pos Pred Value        0.29727  0.24345   0.9256
## Neg Pred Value        0.93840  0.96630   0.3479
## Prevalence            0.09253  0.08613   0.8213
## Detection Rate        0.03901  0.06085   0.5728
## Detection Prevalence  0.13122  0.24996   0.6188
## Balanced Accuracy     0.65998  0.74979   0.7199
```

```
data.frame(cm$table)
```

```
##   Prediction Reference Freq
## 1          0         0  250
## 2          1         0  180
## 3          2         0  163
## 4          0         1   30
## 5          1         1  390
## 6          2         1  132
## 7          0         2  561
## 8          1         2 1032
## 9          2         2 3671
```

```
ggplot(data =  data.frame(cm$table), mapping = aes(x = Prediction, y = Reference)) +
  geom_tile(aes(fill = Freq), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red") +
  theme_bw() + theme(legend.position = "none")+
  ggtitle("Naive Bayes model confusion matrix with Kaggle tagged data")
```