script_wo_cred

```
library(jsonlite)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.2     ✓ dplyr   1.0.6
## ✓ tidyr   1.1.3     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
```

```
library(rtweet)
```

```
## 
## Attaching package: 'rtweet'
```

```
## The following object is masked from 'package:purrr':
## 
##     flatten
```

```
## The following object is masked from 'package:jsonlite':
## 
##     flatten
```

```
library(magrittr)
```

```
## 
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
## 
##     set_names
```

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

```
library(maps)
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(dplyr)
library(tidytext)
library(ggplot2)
library(wordcloud)
```

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(topicmodels)
```

```
tweet_annot <- read_csv("TweetAnnotations")
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Corpus = col_character(),
##   tweet_id = col_character(),
##   annotator = col_character(),
##   ann1 = col_character(),
##   ann2 = col_character(),
##   ann3 = col_character(),
##   ann4 = col_character(),
##   ann5 = col_character(),
##   ann6 = col_character()
## )
```

```
tweets <- read_csv("AllTweets.csv")
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_character(),
##   created_at = col_datetime(format = ""),
##   display_text_width = col_double(),
##   is_quote = col_logical(),
##   is_retweet = col_logical(),
##   favorite_count = col_double(),
##   retweet_count = col_double(),
##   quote_count = col_logical(),
##   reply_count = col_logical(),
##   symbols = col_logical(),
##   ext_media_type = col_logical(),
##   quoted_created_at = col_datetime(format = ""),
##   quoted_favorite_count = col_double(),
##   quoted_retweet_count = col_double(),
##   quoted_followers_count = col_double(),
##   quoted_friends_count = col_double(),
##   quoted_statuses_count = col_double(),
##   quoted_verified = col_logical(),
##   retweet_status_id = col_logical(),
##   retweet_text = col_logical(),
##   retweet_created_at = col_logical()
##   # ... with 21 more columns
## )
## ℹ Use `spec()` for the full column specifications.
```

```
## Warning: 1 parsing failure.
##  row     col           expected actual             file
## 3775 symbols 1/0/T/F/TRUE/FALSE      h 'AllTweets.csv'
```

```
tweets_with_corpus <- left_join(tweets, select(tweet_annot, Corpus, tweet_id), by = c("status_id" = "tweet_id"))
```

```
tweets_clean <- tweets_with_corpus %>%
  select(user_id:text, Corpus, reply_to_status_id, is_quote:symbols, quoted_status_id:quoted_statuses_count, retweet_status_id:retweet_statuses_count, place_name:bbox_coords, followers_count:favourites_count)

# remove URLs
tweets_clean$text <- gsub("https\\S*","", tweets_clean$text)
# remove "@username" tags
tweets_clean$text <- gsub("@\\w+", "", tweets_clean$text)
#remove all URLs with t.co.
tweets_clean$text <- gsub("http://t+","", tweets_clean$text)
```

```
tweets_tokens <- tweets_clean %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_remove_all(text, "&amp;|&lt;|&gt;")) %>%
  unnest_tokens(word, text, token = "tweets") %>%
  filter(!str_detect(word, "^[0-9]*$")) %>%
  anti_join(stop_words)
```

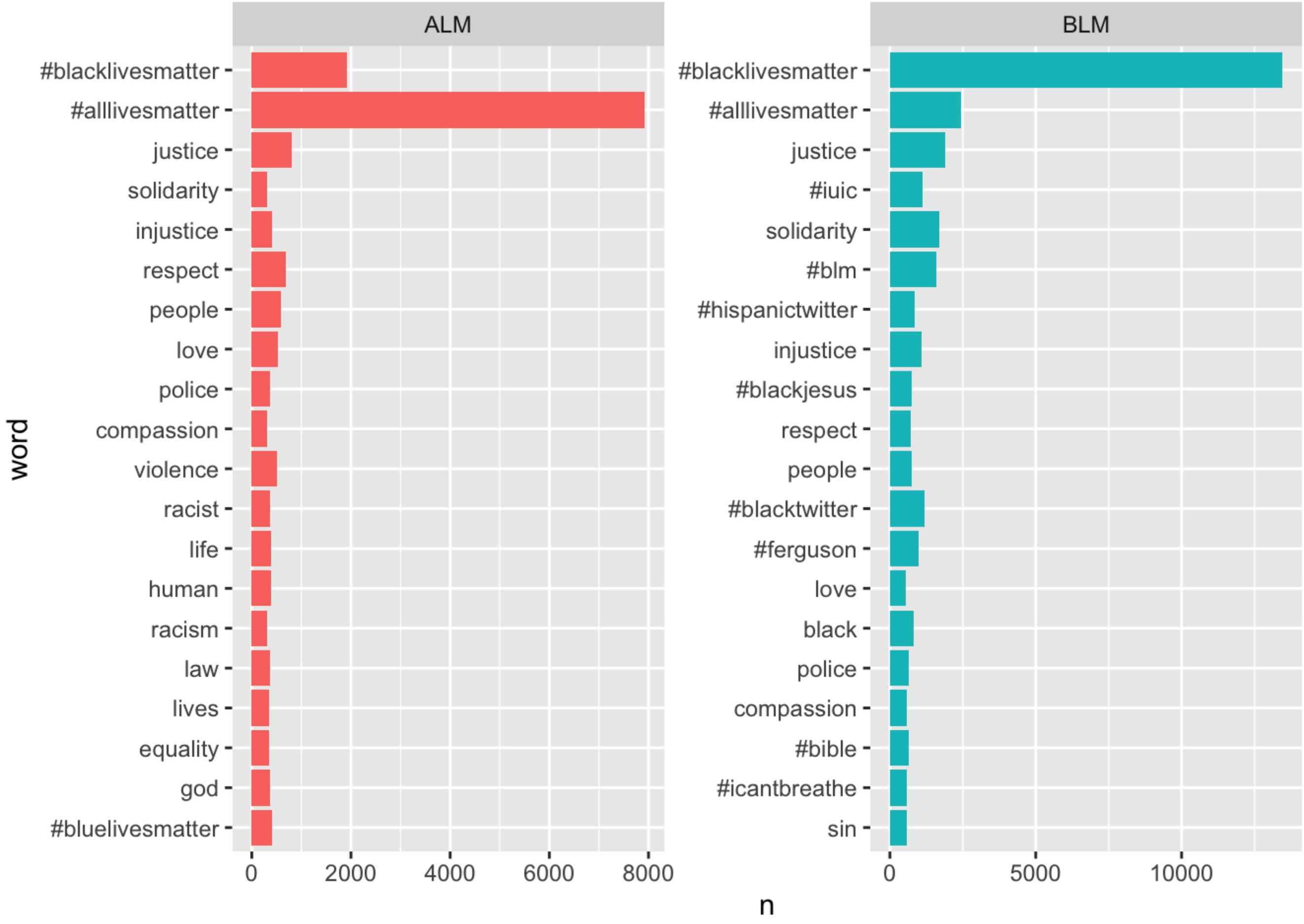
```
## Using `to_lower = TRUE` with `token = 'tweets'` may not preserve URLs.
```

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

```
#Frequently Occuring words in ALM, BLM Corpus.
t2 <- tweets_tokens %>%
  filter(!is.na(Corpus)) %>%
  mutate(Corpus = factor(Corpus)) %>%
  group_by(Corpus) %>%
  count(word) %>%
  arrange(n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  group_by(Corpus) %>%
  top_n(20)
```

```
## Selecting by n
```

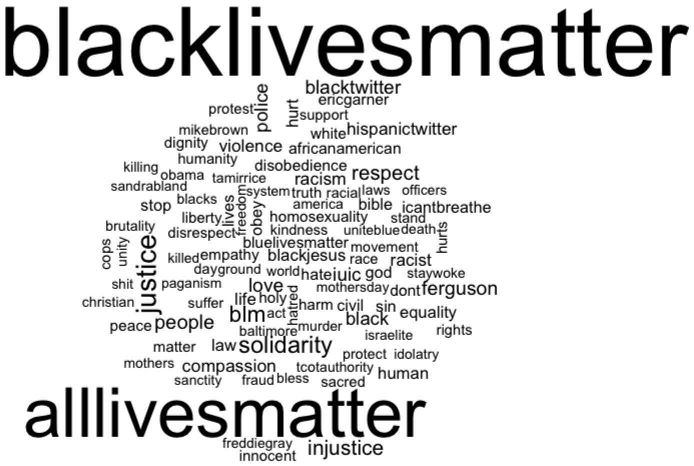
```
ggplot(t2) +
  geom_col(mapping = aes(word, n, fill = Corpus), show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~Corpus, scales = "free")
```



```
wordcloud(tweets_tokens$word, min.freq = 200, scale = c(3, 0.4))
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
```

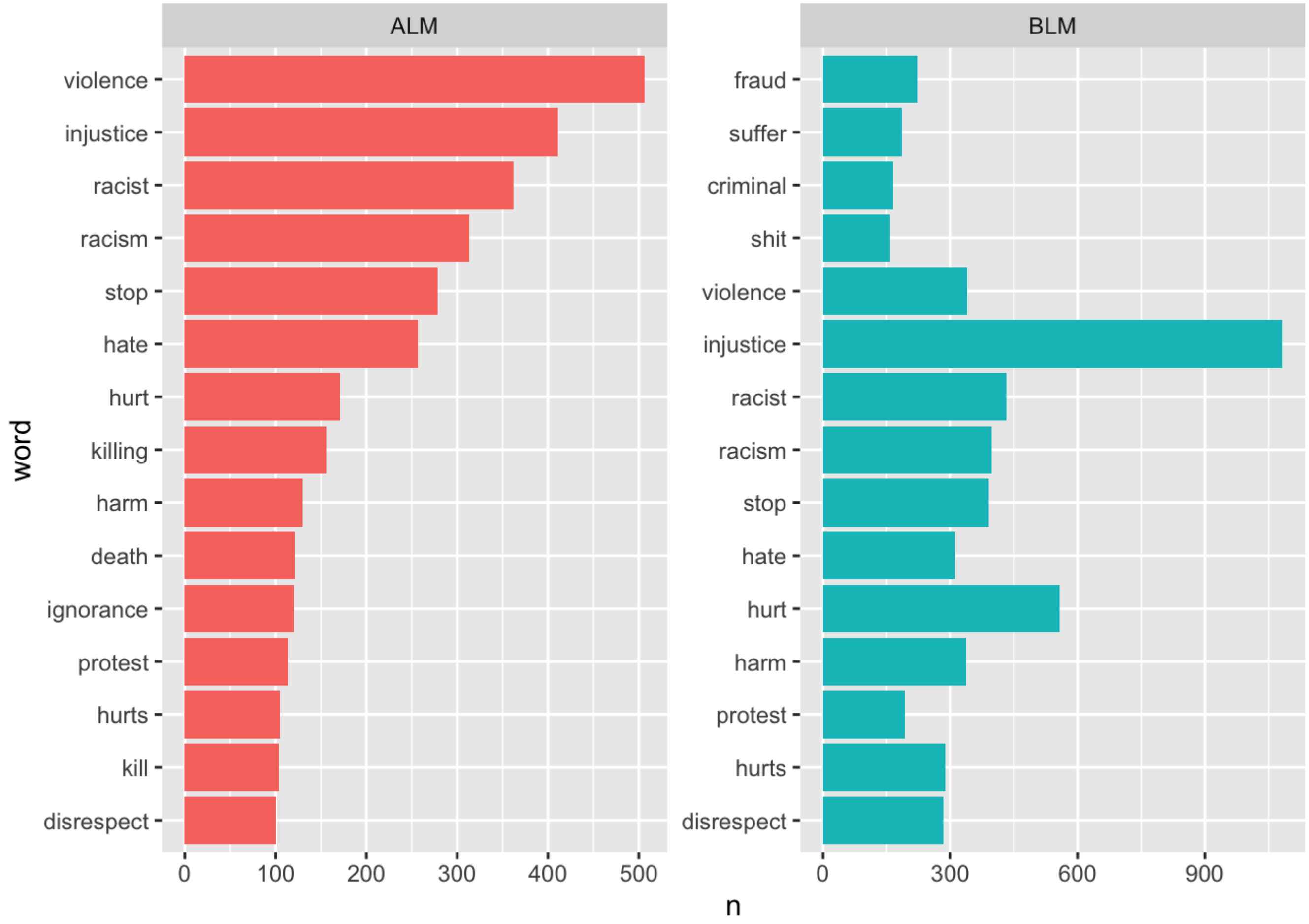


```
#positive and negative sentiments
tweets_sentiments <- tweets_tokens %>% left_join(get_sentiments('afinn'))
```

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

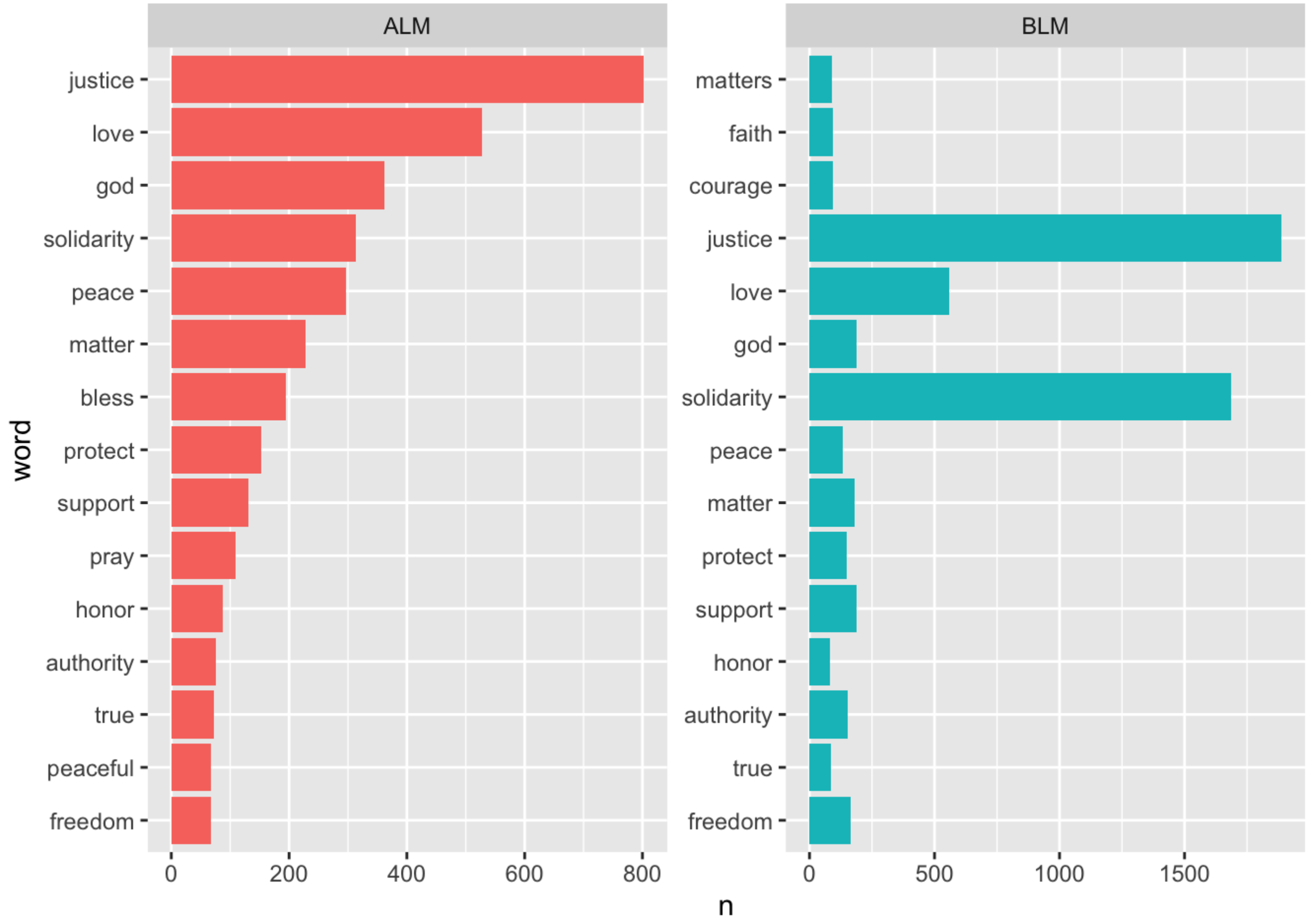
```
#most negative words used by ALM and BLM posts.
tweets_sentiments %>%
  filter(value < 0) %>%
  group_by(Corpus) %>%
  count(word) %>%
  top_n(15) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot() +
  geom_col(mapping = aes(word, n, fill = Corpus), show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~Corpus, scales = "free")
```

```
## Selecting by n
```



```
#most positive words used by ALM and BLM posts.
tweets_sentiments %>%
  filter(value > 0) %>%
  group_by(Corpus) %>%
  count(word) %>%
  top_n(15) %>%
  mutate(word = factor(word)) %>%
  mutate(word = fct_reorder(word, n)) %>%
  ggplot() +
  geom_col(mapping = aes(word, n, fill = Corpus), show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~Corpus, scales = "free")
```

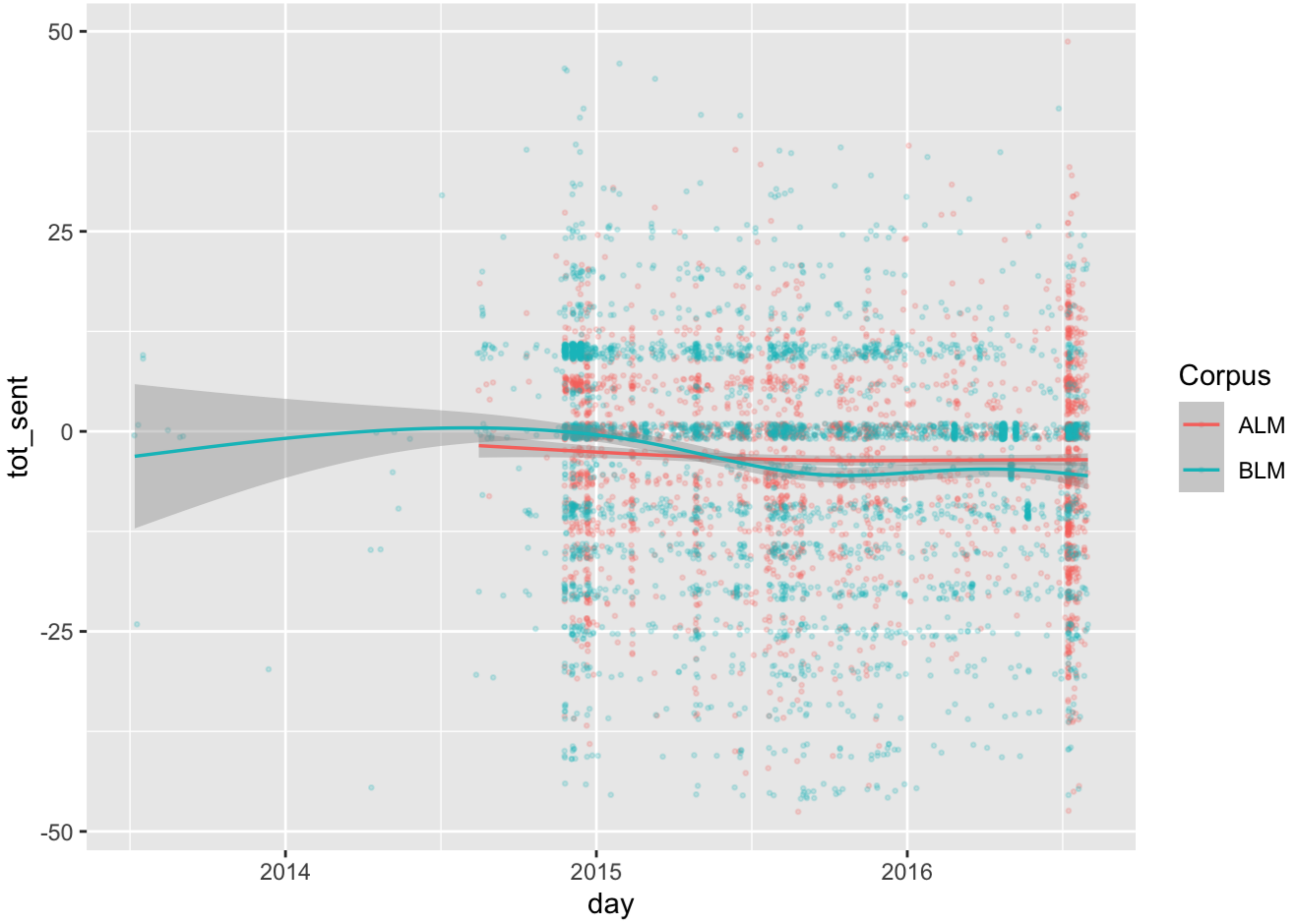
```
## Selecting by n
```



```
#Timeline of ALM vs BLM Tweets. (filtered out the extreme cases)
tweets_sentiments %>%
  mutate(day = floor_date(ymd_hms(created_at), unit = "day")) %>%
  group_by(Corpus, day, status_id) %>%
  summarise(tot_sent = sum(value, na.rm = TRUE)) %>%
  filter(abs(tot_sent) < 50) %>%
  ggplot(aes(day, tot_sent, color = Corpus)) +
  geom_jitter(size = 0.5, height = 1, alpha = 0.2) +
  geom_smooth(size = 0.6)
```

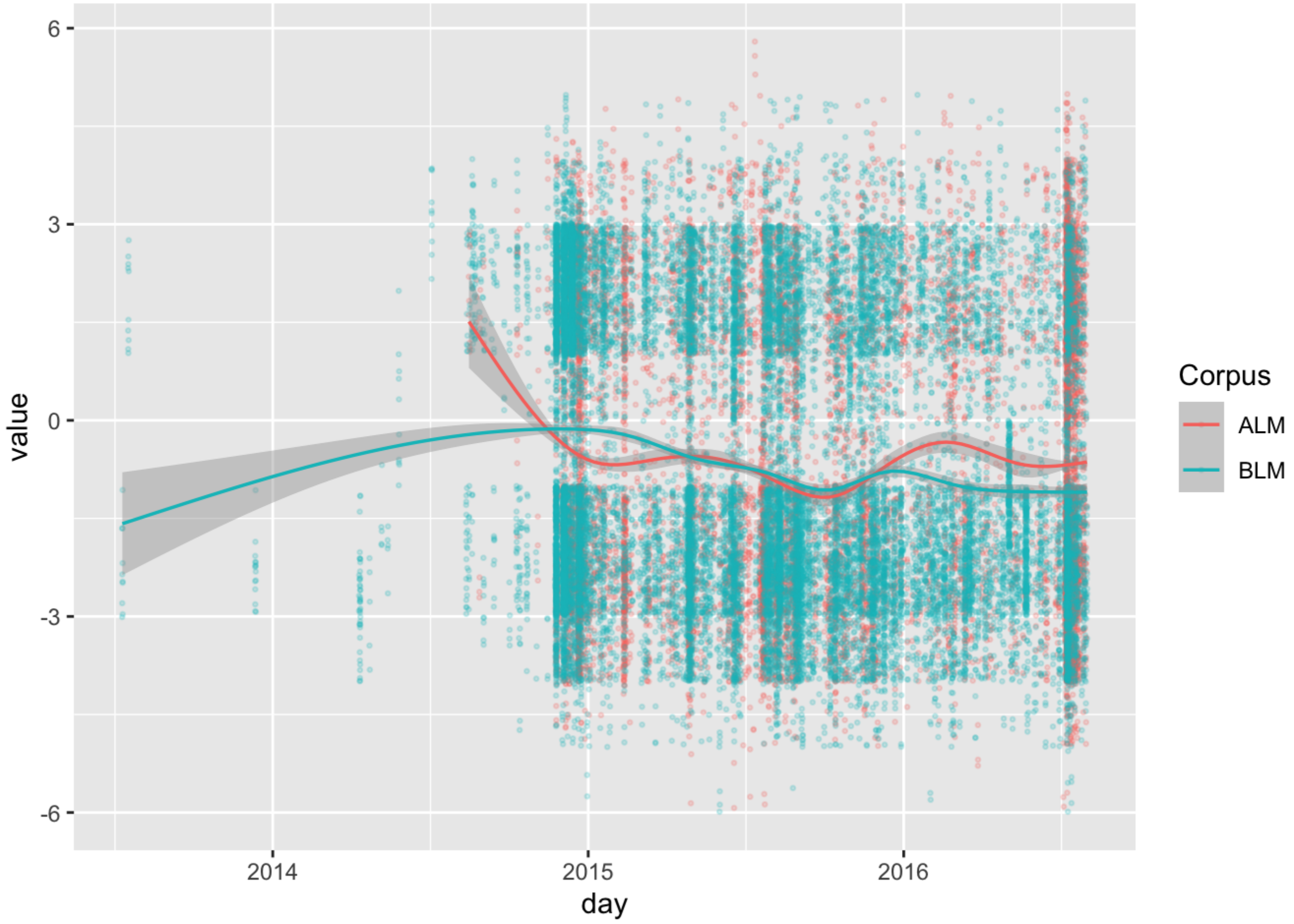
```
## `summarise()` has grouped output by 'Corpus', 'day'. You can override using the `.
groups` argument.
```

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



```
#Timeline of ALM vs BLM Tweets. (filtered out the extreme cases)
tweets_sentiments %>%
  mutate(day = floor_date(ymd_hms(created_at), unit = "day")) %>%
  group_by(Corpus, day) %>%
  filter(!is.na(value)) %>%
  ggplot(aes(day, value, color = Corpus)) +
  geom_jitter(size = 0.5, height = 1, alpha = 0.2) +
  geom_smooth(size = 0.6)
```

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



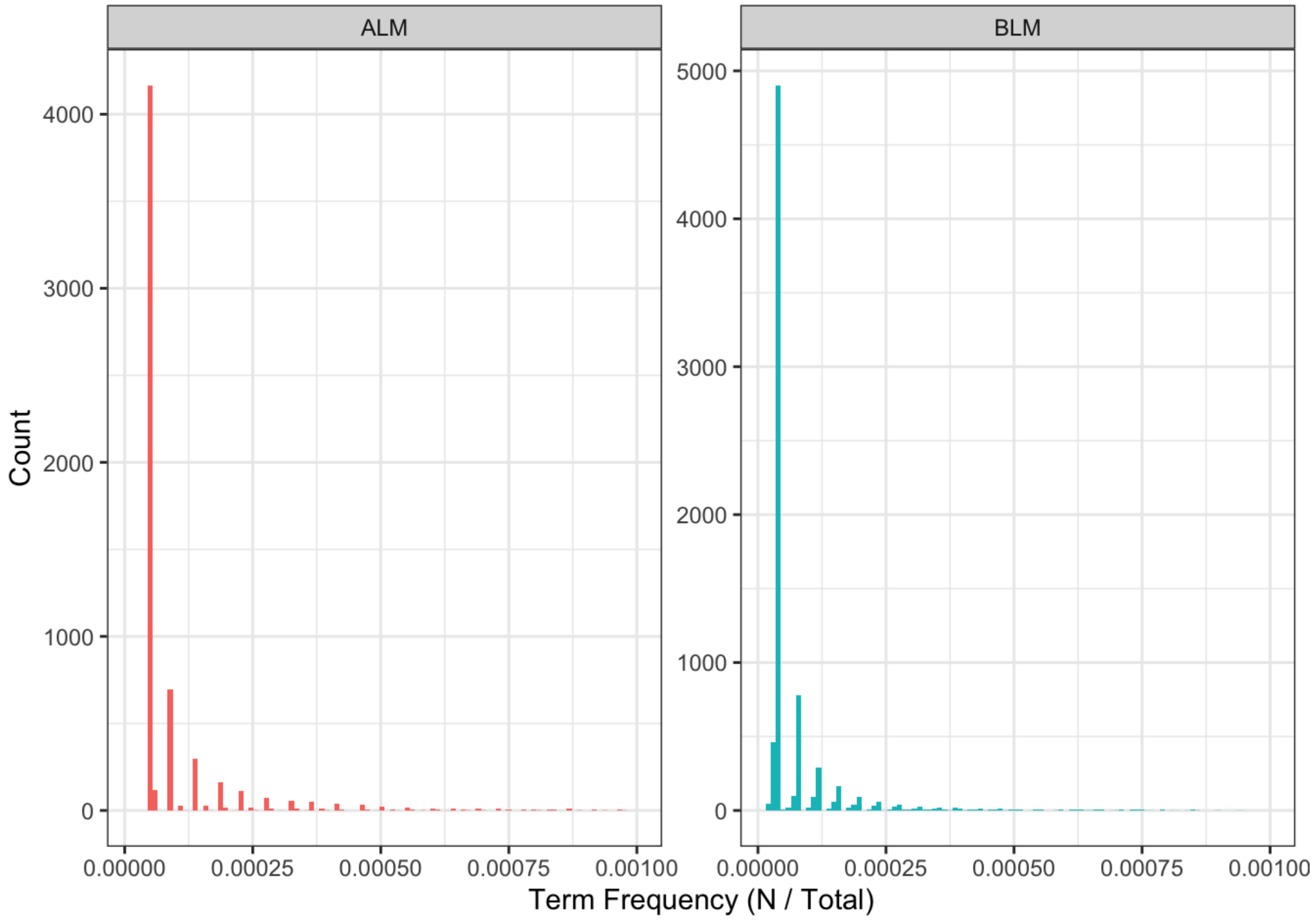
```
tweets_clean <- tweets_tokens %>%
  select(status_id, Corpus, created_at, word) %>%
  count(Corpus, word, sort = TRUE)

total_words <- tweets_clean %>%
  group_by(Corpus) %>%
  summarise(total = sum(n))

tweets_clean <- left_join(tweets_clean, total_words)
```

```
## Joining, by = "Corpus"
```

```
ggplot(tweets_clean, aes(n/total, fill = Corpus)) +
  geom_histogram(show.legend = FALSE, bins = 100) +
  xlim(NA, 0.001) +
  facet_wrap(~Corpus, scales = "free_y") +
  labs(x = "Term Frequency (N / Total)", y = "Count") +
  theme_bw()
```

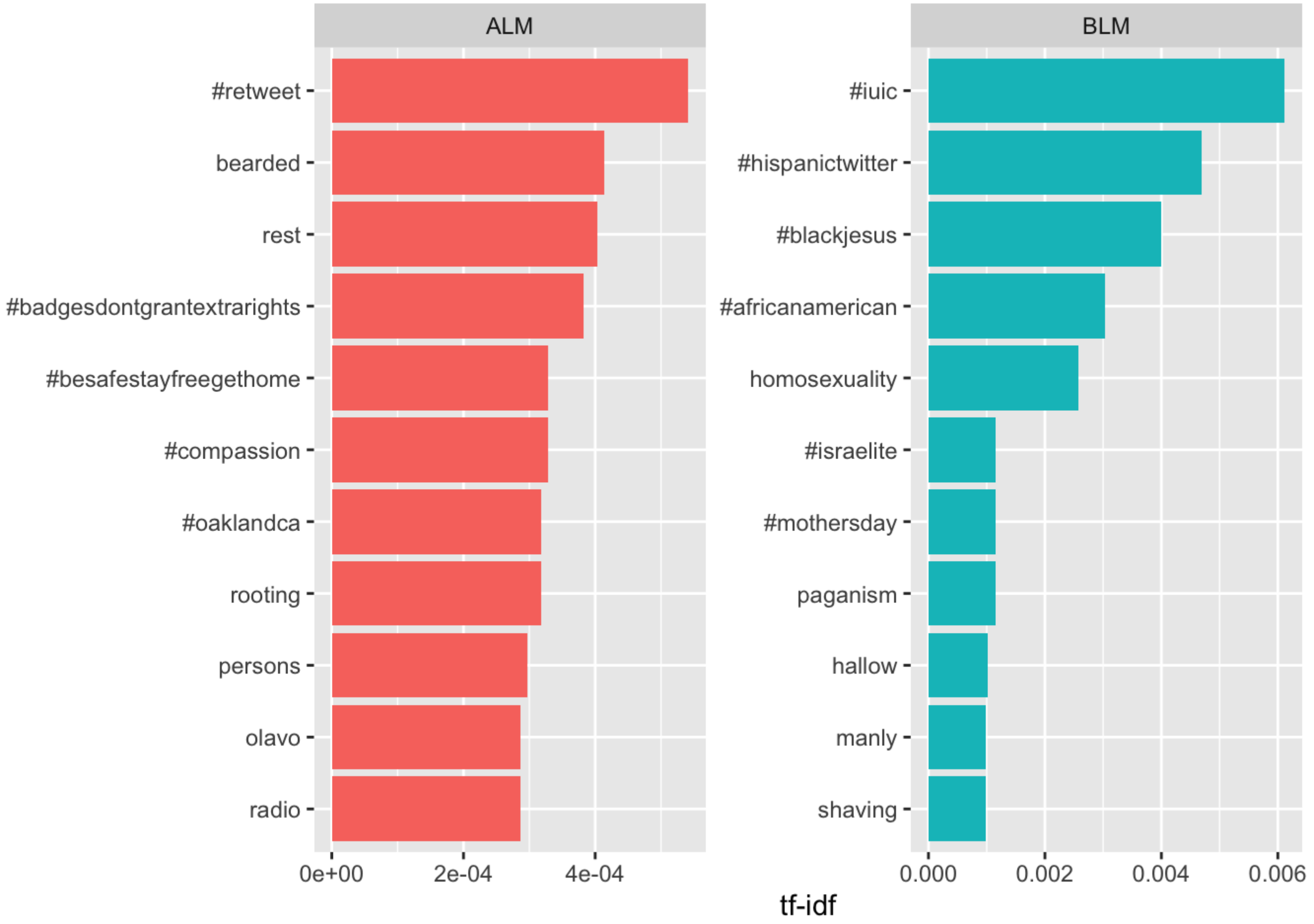
```
## Warning: Removed 224 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```
tweets_clean %>%
  bind_tf_idf(word, Corpus, n) %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(Corpus) %>%
  filter(word != "day2") %>%
  top_n(10) %>%
  ungroup() %>%
  ggplot(aes(word, tf_idf, fill = Corpus)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~Corpus, scales = "free") +
  coord_flip()
```

```
## Selecting by tf_idf
```

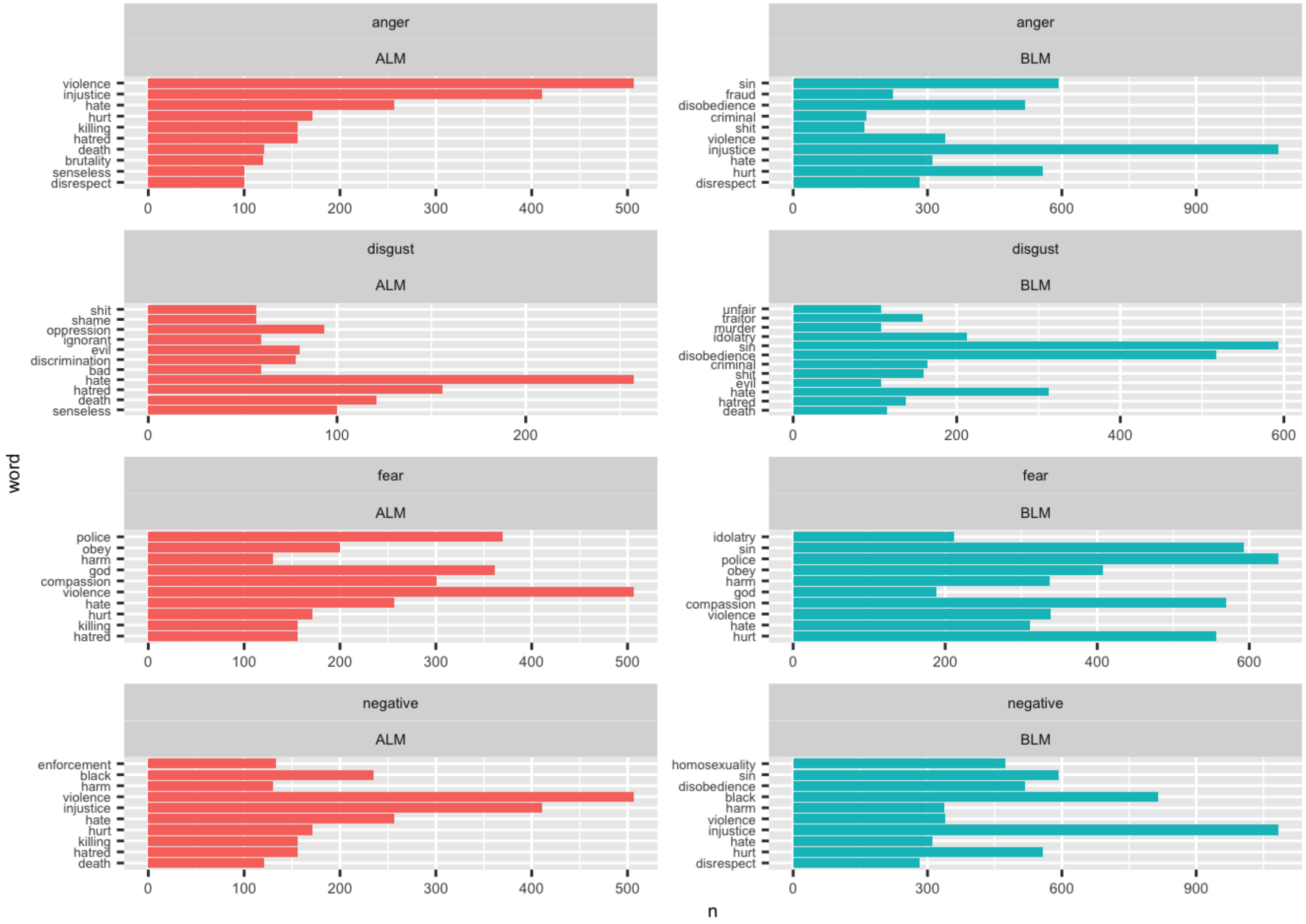


```
tweets_sentiments <- tweets_tokens %>% left_join(get_sentiments('nrc'))
```

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

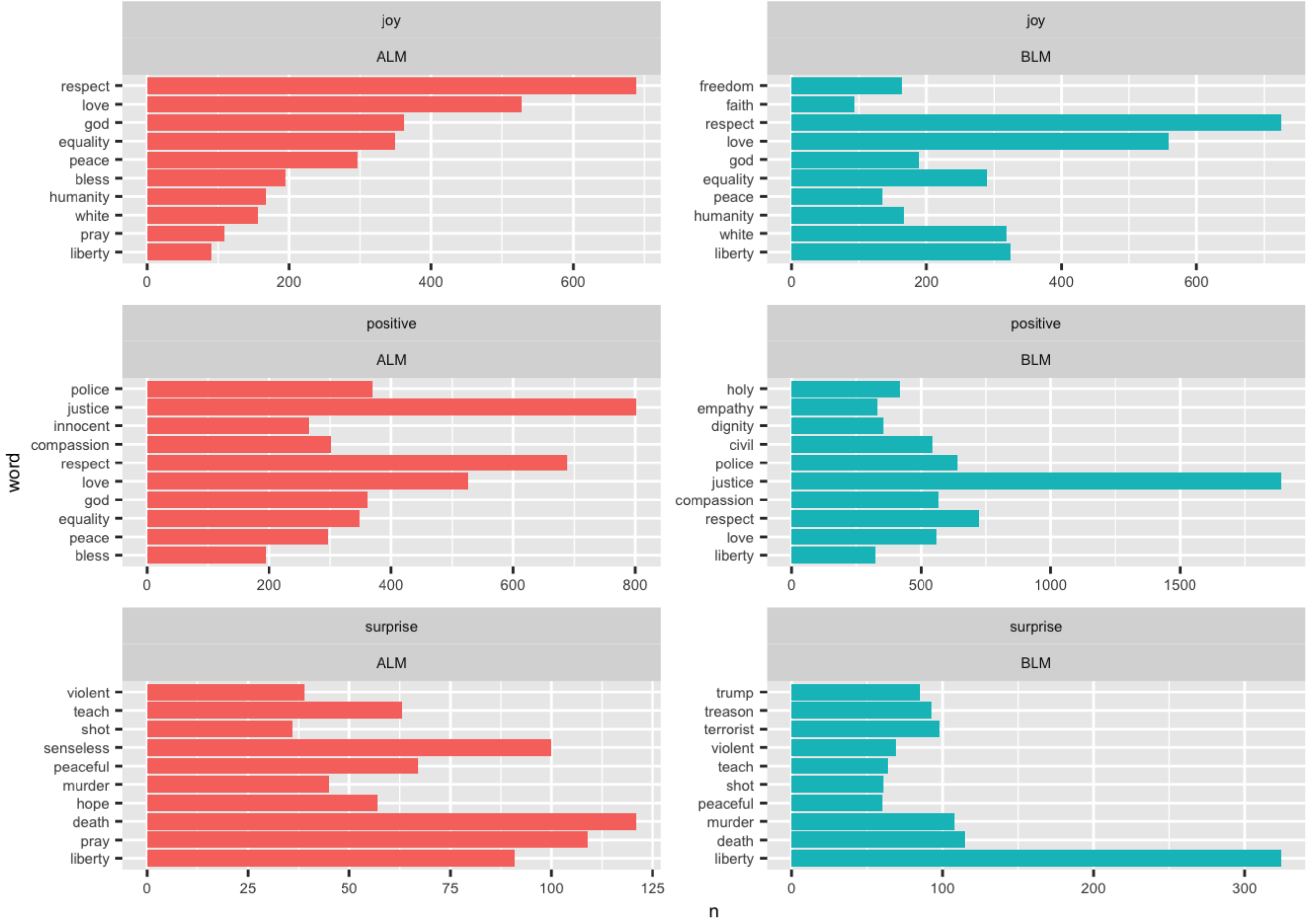
```
tweets_sentiments %>%
  filter(sentiment %in% c("fear", "negative", "anger", "disgust")) %>%
  group_by(Corpus, sentiment) %>%
  count(word) %>%
  arrange(desc(n)) %>%
  top_n(10) %>%
  mutate(word = factor(word)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot() +
  geom_col(mapping = aes(word, n, fill = Corpus), show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~sentiment + Corpus, scales = "free", ncol = 2) +
  theme(axis.text.y = element_text(size=5), text = element_text(size=7))
```

```
## Selecting by n
```

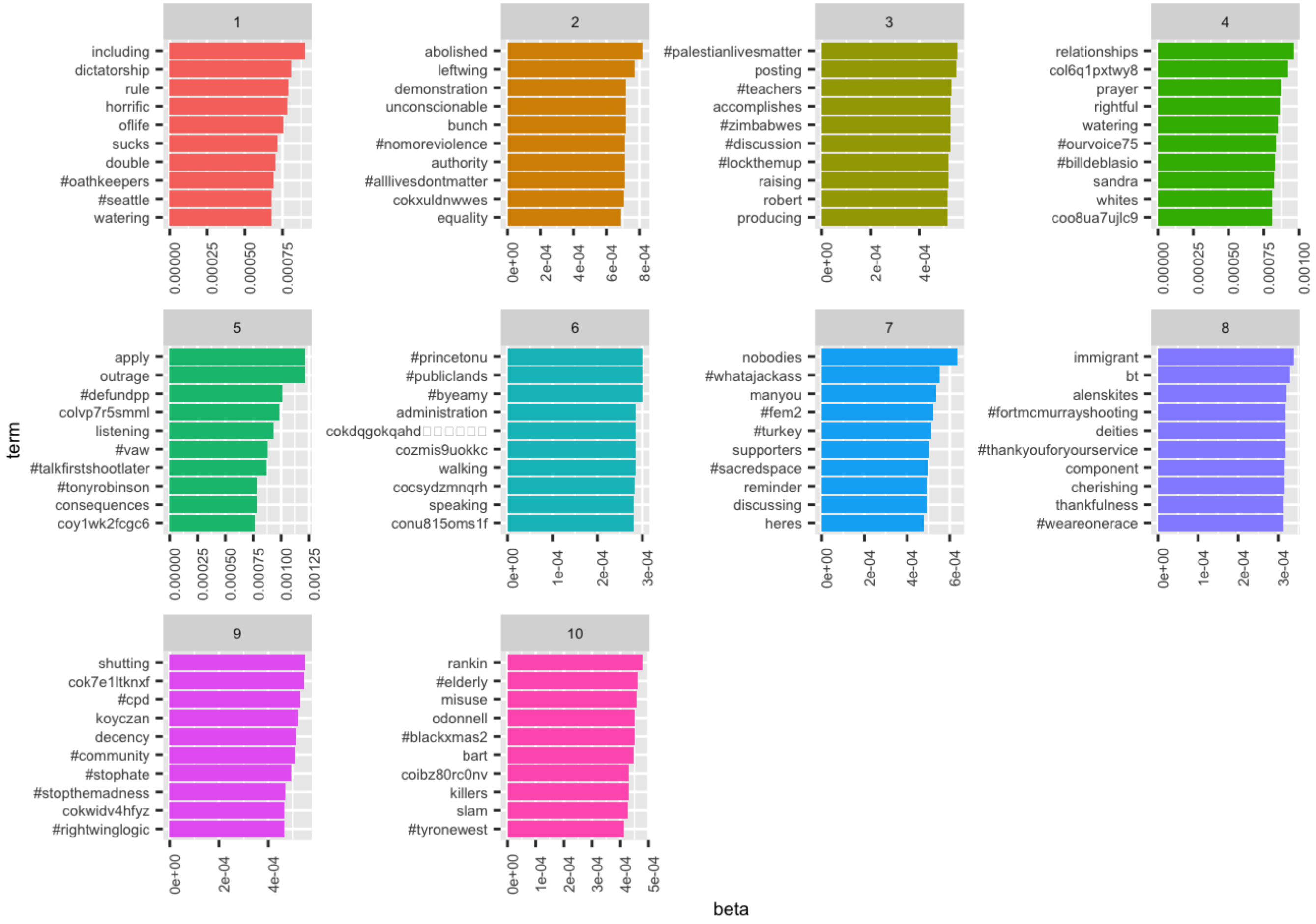


```
#positive affects related to the posts
tweets_sentiments %>%
  filter(sentiment %in% c("surprise", "positive", "joy")) %>%
  group_by(Corpus, sentiment) %>%
  count(word) %>%
  arrange(desc(n)) %>%
  top_n(10) %>%
  mutate(word = factor(word)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot() +
  geom_col(mapping = aes(word, n, fill = Corpus), show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~sentiment + Corpus, scales = "free", ncol = 2) +
  theme(text = element_text(size = 7))
```

```
## Selecting by n
```



```
data_dtm <- tweets_clean %>%
  count(Corpus, word) %>%
  cast_dtm(document = Corpus, term = word, value = n, weighting = tm::weightTf)
tweets_lda <- LDA(data_dtm, k = 10, control = list(seed = 1234))
tweet_topics <- tidy(tweets_lda, matrix = "beta")

# make dataframe showcasing the 10 words with highest beta per topic
tweet_top_terms <- tweet_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

# plot top words for each topic
tweet_top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  scale_x_reordered() +
  theme(axis.text.x = element_text(angle = 90), text = element_text(size = 7))
```



```
tweet_annot_clean <- tweet_annot %>%
  select(Corpus:ann1)

tweets_morality <- left_join(tweets, select(tweet_annot, Corpus, tweet_id, annotator,
ann1), by = c("status_id" = "tweet_id")) %>%
  select(status_id, created_at, text, Corpus, ann1, annotator, favorite_count, retwee
t_count, reply_count)
```

```
tweets_morality %>%
  distinct(ann1)
```

```
## # A tibble: 11 x 1
##    ann1
##    <chr>
##  1 harm
##  2 authority
##  3 degradation
##  4 care
##  5 fairness
##  6 loyalty
##  7 non-moral
##  8 cheating
##  9 betrayal
## 10 subversion
## 11 purity
```

```
tweets_morality %>%
  select(Corpus, ann1, status_id, text)
```

```
## # A tibble: 25,403 x 4
##    Corpus ann1      status_id      text
##    <chr>  <chr>     <chr>          <chr>
##  1 ALM    harm      x665629403842… #AllLivesMatter  https://t.co/dk1saq84DN
##  2 ALM    harm      x665629403842… #AllLivesMatter  https://t.co/dk1saq84DN
##  3 ALM    harm      x665629403842… #AllLivesMatter  https://t.co/dk1saq84DN
##  4 ALM    harm      x665629403842… #AllLivesMatter  https://t.co/dk1saq84DN
##  5 ALM    harm      x547086823043… How is de Blasio's encouragement of peaceful…
##  6 ALM    authority x547086823043… How is de Blasio's encouragement of peaceful…
##  7 ALM    degradat… x547086823043… How is de Blasio's encouragement of peaceful…
##  8 ALM    harm      x547086823043… How is de Blasio's encouragement of peaceful…
##  9 ALM    care      x594300751960… Let'sfind love and peace #AllLivesMatter htt…
## 10 ALM    care      x594300751960… Let'sfind love and peace #AllLivesMatter htt…
## # … with 25,393 more rows
```

```
tweets_morality %>%
  group_by(Corpus, status_id, ann1) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  group_by(Corpus, status_id) %>%
  mutate(maj = n/sum(n)) %>%
  arrange(status_id) %>%
  filter(maj == 0.5) %>%
  distinct(status_id)
```

```
## `summarise()` has grouped output by 'Corpus', 'status_id'. You can override using
the `.groups` argument.
```

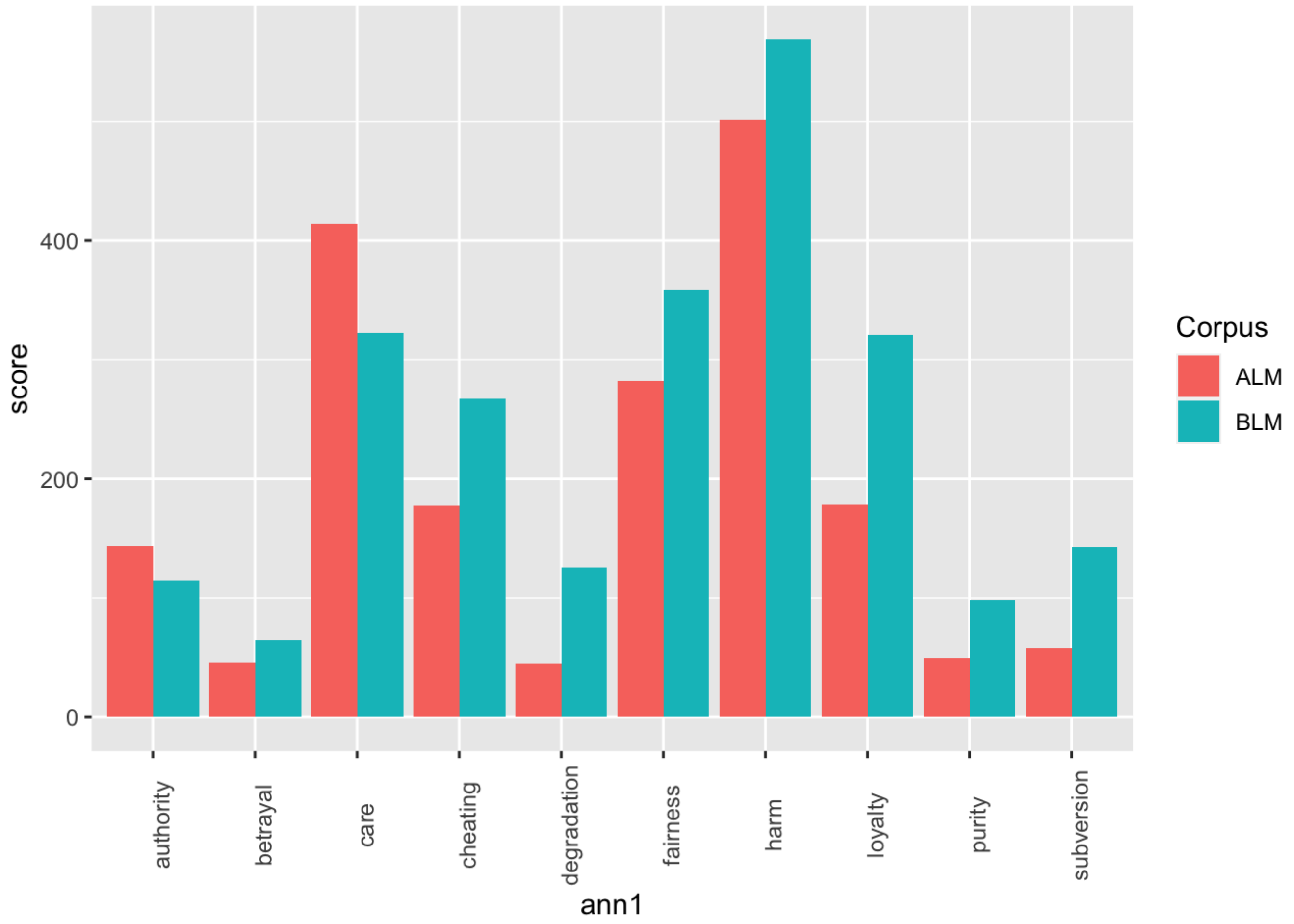
```
## # A tibble: 126 x 2
## # Groups:   Corpus, status_id [126]
##    Corpus status_id
##    <chr>  <chr>
##  1 BLM    x373464973147516928
##  2 BLM    x537256345427140608
##  3 ALM    x537431692021997568
##  4 ALM    x537681598989475841
##  5 BLM    x539967017051521024
##  6 ALM    x540400187458351105
##  7 BLM    x541065077684985856
##  8 BLM    x541308308695822336
##  9 BLM    x541317736601632768
## 10 BLM    x541638586496344067
## # … with 116 more rows
```

```
morality_score <- tweets_morality %>%
  group_by(Corpus, status_id, text, ann1) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  group_by(Corpus, status_id) %>%
  mutate(maj = n/sum(n))
```

```
## `summarise()` has grouped output by 'Corpus', 'status_id', 'text'. You can overrid
e using the `.groups` argument.
```

```
morality_score %>%
  select(Corpus, status_id, ann1, maj) %>%
  filter(ann1 != "non-moral") %>%
  group_by(Corpus, ann1) %>%
  summarise(score = sum(maj)) %>%
  ggplot() +
  geom_col(aes(ann1, score, fill = Corpus), position ="dodge") +
  theme(axis.text.x = element_text(angle = 90))
```

```
## `summarise()` has grouped output by 'Corpus'. You can override using the `.groups`
argument.
```


score
authority
betrayal
care
cheating
degradation
fairness
harm
loyalty
purity
subversion
ann1
Corpus
ALM
BLM