Team 16

Christian, Simon und Cuca

23 5 2021

Daten einlesen

```
library(tidyverse)
```

```
## Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame'
## when loading 'dplyr'
```

```
library(stringi)

pm_csv <- str_c("data/2020-12-", 1:26, "_presseportal.csv")
pm_csv <- c(pm_csv, str_c("data/2021-1-", 1:31, "_presseportal.csv"))
pm_csv <- c(pm_csv, str_c("data/2021-2-", 1:28, "_presseportal.csv"))
pm_csv <- c(pm_csv, str_c("data/2021-3-", 1:31, "_presseportal.csv"))
pm_csv <- c(pm_csv, str_c("data/2021-4-", 1:30, "_presseportal.csv"))
pm_csv <- c(pm_csv, str_c("data/2021-5-", 1:21, "_presseportal.csv"))
pm_list <- lapply(pm_csv, read_csv)
pm <- do.call(rbind, pm_list)

tweets <- read_csv("data/copbird_table_tweet.csv")
tweets <- tweets[tweets$created_at >= "2021-04-01", 1:4]
usersX <- read_csv("data/copbird_table_user_ext.csv")
tweetXstate <- read_csv("data/copbird_table_tweet_ext_state.csv")
blaulicht <- read_csv("data/2020-12_2021-05_presseportal.csv")

pm_demo <- read_csv("data/copbird_table_pm_topiced_demonstr.csv")
```

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

```
tw_demo <- read_csv("data/copbird_table_tweet_topiced_demonstr.csv")
```

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

```
pm_drogen <- read_csv("data/copbird_table_pm_topiced_drogen.csv")
```

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

```r
tw_drogen <- read_csv("data/copbird_table_tweet_topiced_drogen.csv")
```

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

```r
pm_rass <- read_csv("data/copbird_table_pm_topiced_rassis.csv")
```

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

```r
tw_rass <- read_csv("data/copbird_table_tweet_topiced_rassis.csv")
```

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

Scrapen der Pressemeldungen (seit Dezember 2020)

Zuordnung von Orten der Pressemeldungen und Tweets

```r
head(usersX)
```

```
## # A tibble: 6 x 4
##   user_id name                           handle          bundesland
##     <dbl> <chr>                          <chr>           <chr>
## 1 1.03e18 Polizei Wittlich               PolizeiWittlich Rheinland-Pfalz
## 2 1.14e18 Bayerisches Landeskriminalamt LKA_Bayern      Bayern
## 3 1.17e18 Polizei Stendal                Polizei_SDL     Sachsen-Anhalt
## 4 1.18e18 Polizei Ravensburg             PolizeiRV       Baden-Württemberg
## 5 1.23e18 Polizei Bad Nenndorf           Polizei_BadN    Niedersachsen
## 6 1.30e18 Polizei ZPD NI                 Polizei_ZPD_NI  Niedersachsen
```

```r
head(tweetXstate[, 5:8])
```

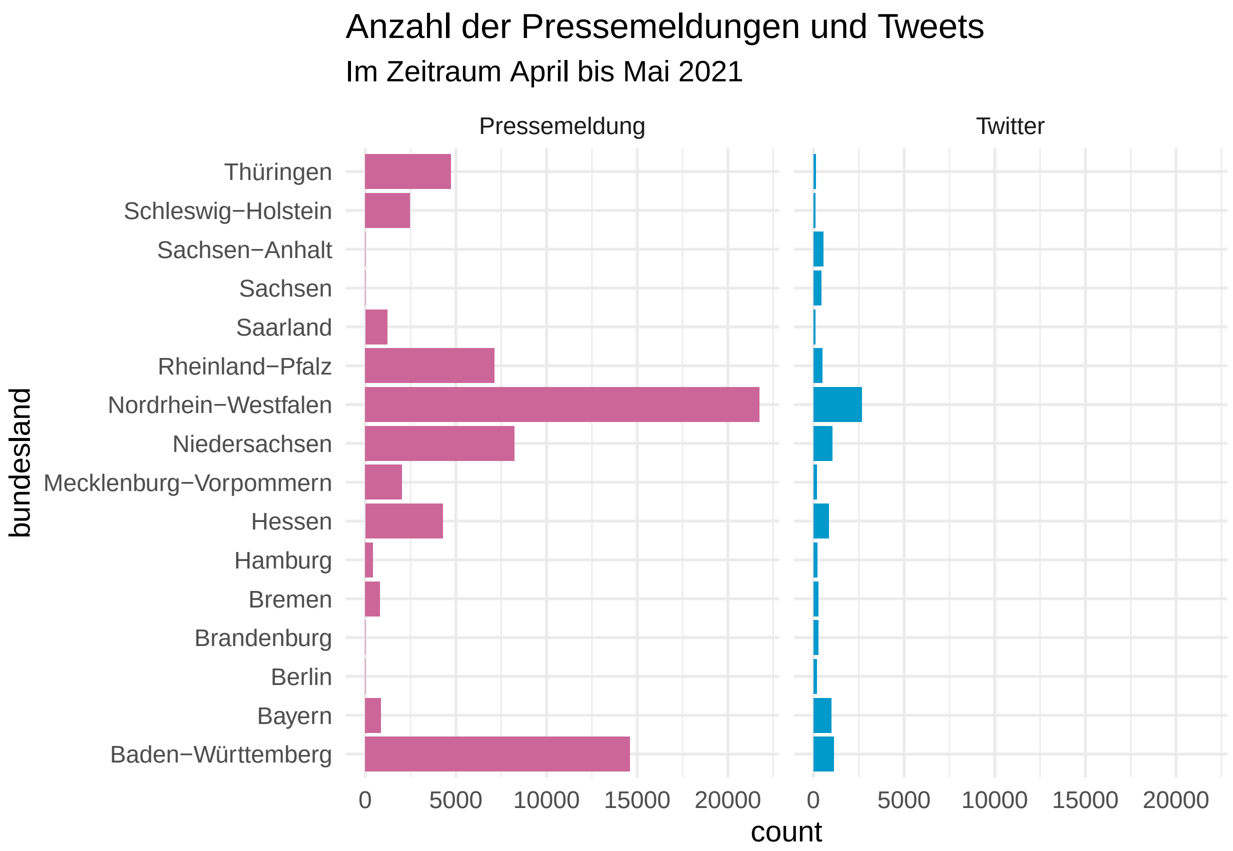
```
## # A tibble: 6 x 4
##   user_name                       handle          stadt   bundesland
##   <chr>                           <chr>           <chr>   <chr>
## 1 Polizei Oldenburg-Stadt/Ammerl Polizei_OL      <NA>    <NA>
## 2 Polizei Berlin                  polizeiberlin   Berlin  Berlin
## 3 Polizei Berlin                  polizeiberlin   Berlin  Berlin
## 4 Polizei München                 PolizeiMuenchen München Bayern
## 5 Polizei Sachsen                 PolizeiSachsen  Dresden Sachsen
## 6 Polizei Berlin                  polizeiberlin   Berlin  Berlin
```

```r
blaulicht$tw_user_id <- as.character(blaulicht$tw_user_id)
head(blaulicht[, -c(2, 5)])
```

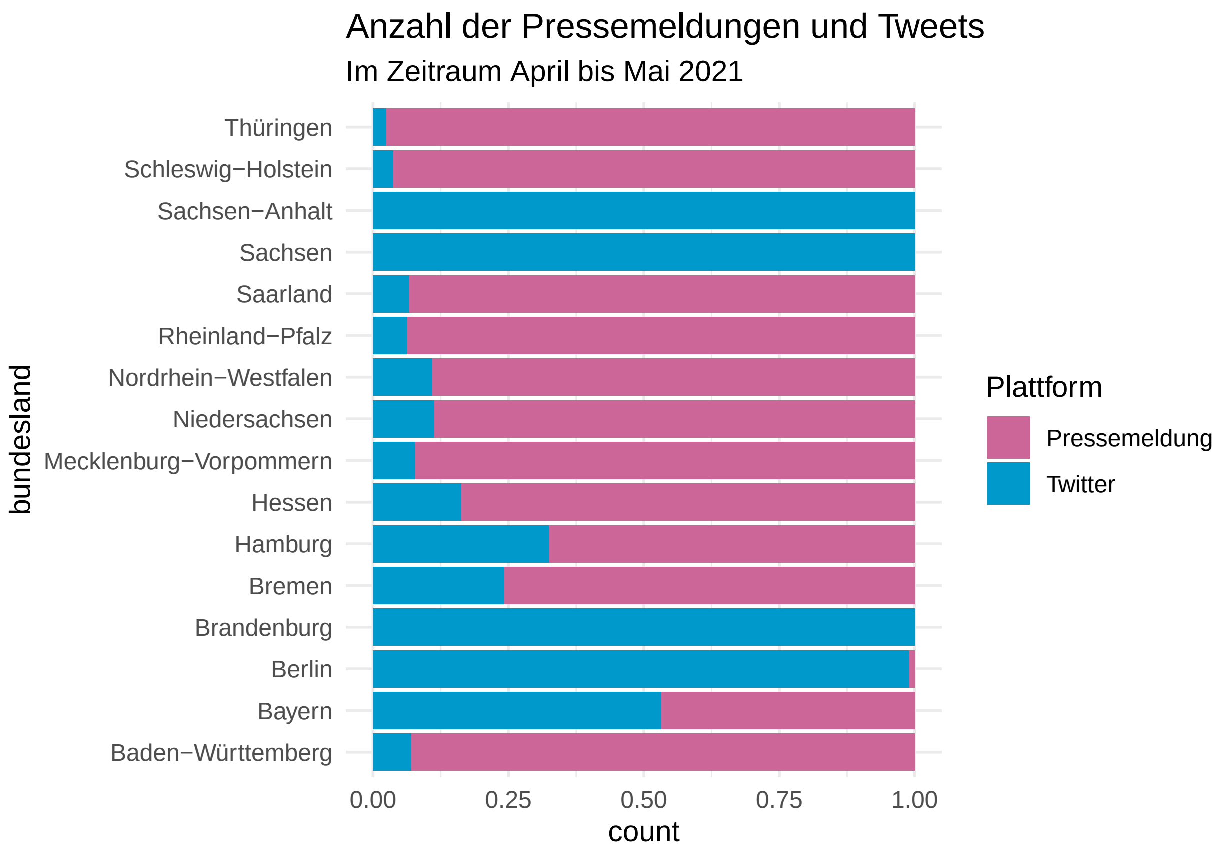
```
## # A tibble: 6 x 4
##   article_id     location           bundesland         tw_user_id
##   <chr>          <chr>              <chr>              <chr>
## 1 137462-4788051 Mühlacker          baden-wuerttemberg <NA>
## 2 110972-4788043 Karlsruhe-Oststadt baden-wuerttemberg 3029998264
## 3 110975-4788034 Achern             baden-wuerttemberg <NA>
## 4 110976-4788018 Reutlingen         baden-wuerttemberg 823465058650955776
## 5 138081-4788002 Friedrichshafen    baden-wuerttemberg <NA>
## 6 138081-4787991 Bodenseekreis      baden-wuerttemberg <NA>
```

Anzahl Pressemeldungen vs. Tweets

```r
land_tw <- full_join(tweets, usersX[c(1, 4)], by = "user_id")
land_tw$bundesland[land_tw$bundesland == "-"] <- NA_character_
land_tw <- land_tw %>% group_by(bundesland) %>% count()
land_tw$bundesland <- as.factor(land_tw$bundesland)

land_pm <- pm %>% group_by(bundesland) %>% count()
land_pm$bundesland[land_pm$bundesland == "berlin-brandenburg"] <- "berlin"
land_pm$bundesland <- stri_trans_totitle(land_pm$bundesland)
land_pm$bundesland <- gsub("ue", "ü", land_pm$bundesland)
land_pm$bundesland <- factor(land_pm$bundesland, levels = levels(land_tw$bundesland))

land_pm_tw <- full_join(land_pm, land_tw, by = "bundesland")
names(land_pm_tw)[2:3] <- c("Pressemeldung", "Twitter")
land_pm_tw <- land_pm_tw[-which(is.na(land_pm_tw$bundesland)), ]
land_pm_tw$Pressemeldung[which(is.na(land_pm_tw$Pressemeldung))] <- 0
land_pm_tw <- gather(land_pm_tw, key = "Plattform", value = "count", -bundesland)

ggplot(land_pm_tw) +
  geom_col(aes(x = bundesland, y = count, fill = Plattform)) +
  scale_fill_manual(values = c("#CC6699", "#0099CC")) +
  facet_wrap(~Plattform) +
  coord_flip() +
  guides(fill = FALSE) +
  labs(title = "Anzahl der Pressemeldungen und Tweets",
       subtitle = "Im Zeitraum April bis Mai 2021") +
  theme_minimal()
```



```r
ggplot(land_pm_tw) +
  geom_col(aes(x = bundesland, y = count, fill = Plattform), position = "fill") +
  scale_fill_manual(values = c("#CC6699", "#0099CC")) +
  coord_flip() +
  labs(title = "Anzahl der Pressemeldungen und Tweets",
       subtitle = "Im Zeitraum April bis Mai 2021") +
  theme_minimal()
```



Topic modelling

```
# library(quanteda)
# library(tidyverse)
# library(topicmodels)
# library(ldatuning)
# library(stm)
# library(wordcloud)
#
# pm <- pm[!is.na(pm$content), ]
# tok <- tokens(pm$content_ber_satzzeichen)
# mydfm <- dfm(tok, remove_numbers = TRUE, remove_punct = TRUE, remove_symbols = TRUE, remove = stopwor
# mydfm.trim <-  dfm_trim(mydfm, min_docfreq = 3, max_docfreq = 65)
# # mydfm.trim
#
# anzahl.themen <- 10
# anzahl.woerter <- 10
# dfm2topicmodels <- convert(mydfm.trim, to = "topicmodels")
# lda.modell <- LDA(dfm2topicmodels, anzahl.themen)
# lda.modell
# topmod <- as.data.frame(terms(lda.modell, anzahl.woerter))
# topmod
#
```

```
# write_csv(topmod, "data/topicmodel.csv")
```

Auswahl der Keywords

topic_1 = ['demonstr', 'kundgeb']

topic_2 = ['drogen', 'weed', 'graas', 'lsd', 'cannabis', 'ecstasy', 'kokain', 'meth', 'crystal']

topic_3 = ['rassis', 'diskriminier', 'ausländerfeindlich', 'fremdenfeindlich', 'fremdenhass']

topic_4 = ['antisem', 'juden', 'synagoge', 'judenhass', 'judenfeindlich', 'holocaust']

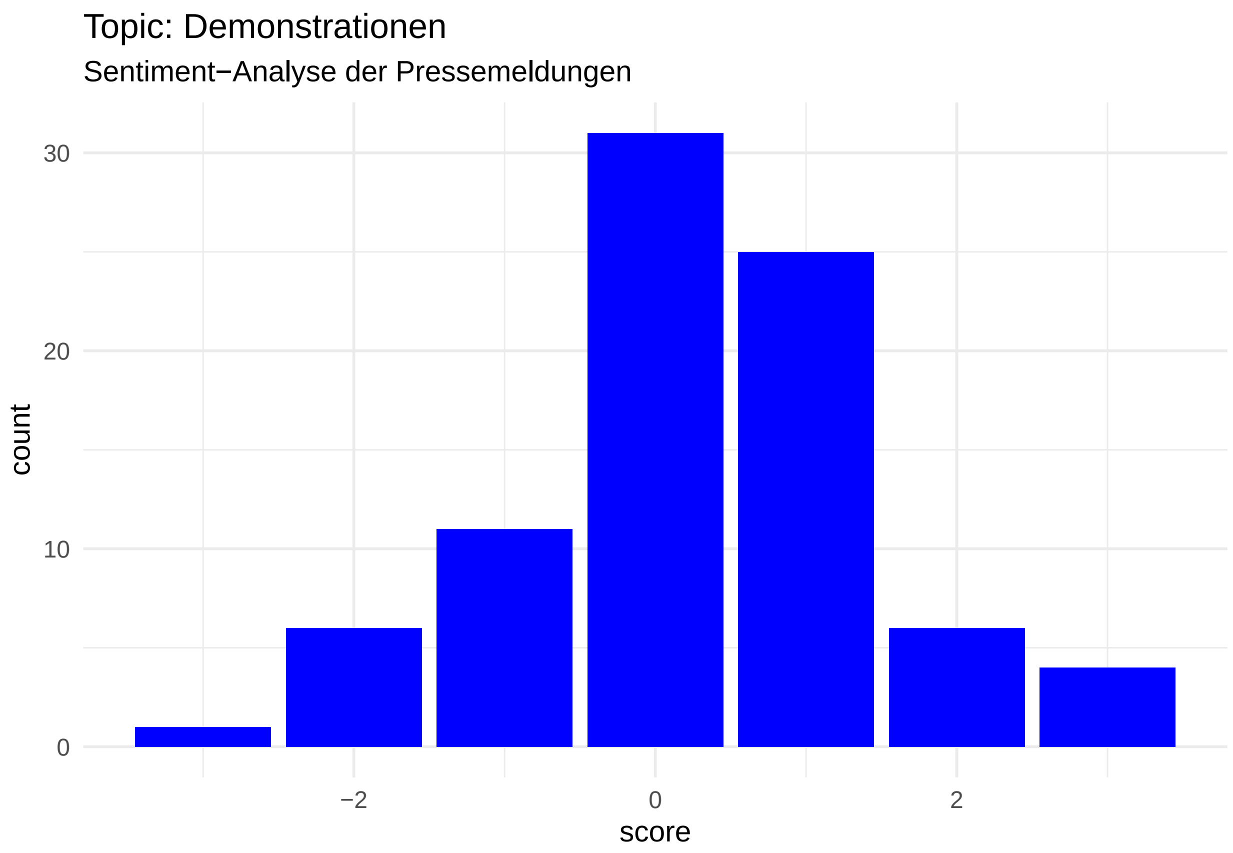
Sentiment Analyse

```
readAndflattenSentiWS <- function(filename) {
  words = readLines(filename, encoding="UTF-8")
  words <- sub("\\|[A-Z]+\t[0-9.-]+\t?", ",", words)
  words <- unlist(strsplit(words, ","))
  words <- tolower(words)
  return(words)
}

pos.words <- c(scan("SentiWS/positive-words.txt",what=’character’, comment.char=’;’, quiet=T),
               readAndflattenSentiWS("SentiWS/positive-words.txt"))
neg.words <- c(scan("SentiWS/negative-words.txt",what=’character’, comment.char=’;’, quiet=T),
              readAndflattenSentiWS("SentiWS/negative-words.txt"))

score.sentiment = function(sentences, pos.words, neg.words, .progress=’none’) {
  require(plyr)
  require(stringr)
  scores = laply(sentences, function(sentence, pos.words, neg.words)
  {
    # clean up sentences with R’s regex-driven global substitute, gsub():
    sentence = gsub(’[[:punct:]]’, ’’, sentence)
    sentence = gsub(’[[:cntrl:]]’, ’’, sentence)
    sentence = gsub(’\\d+’, ’’, sentence)
    # and convert to lower case:
    sentence = tolower(sentence)
    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, ’\\s+’)
    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)
    # compare our words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)
    # match() returns the position of the matched term or NA
    # I don’t just want a TRUE/FALSE! How can I do this?
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)
    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
```

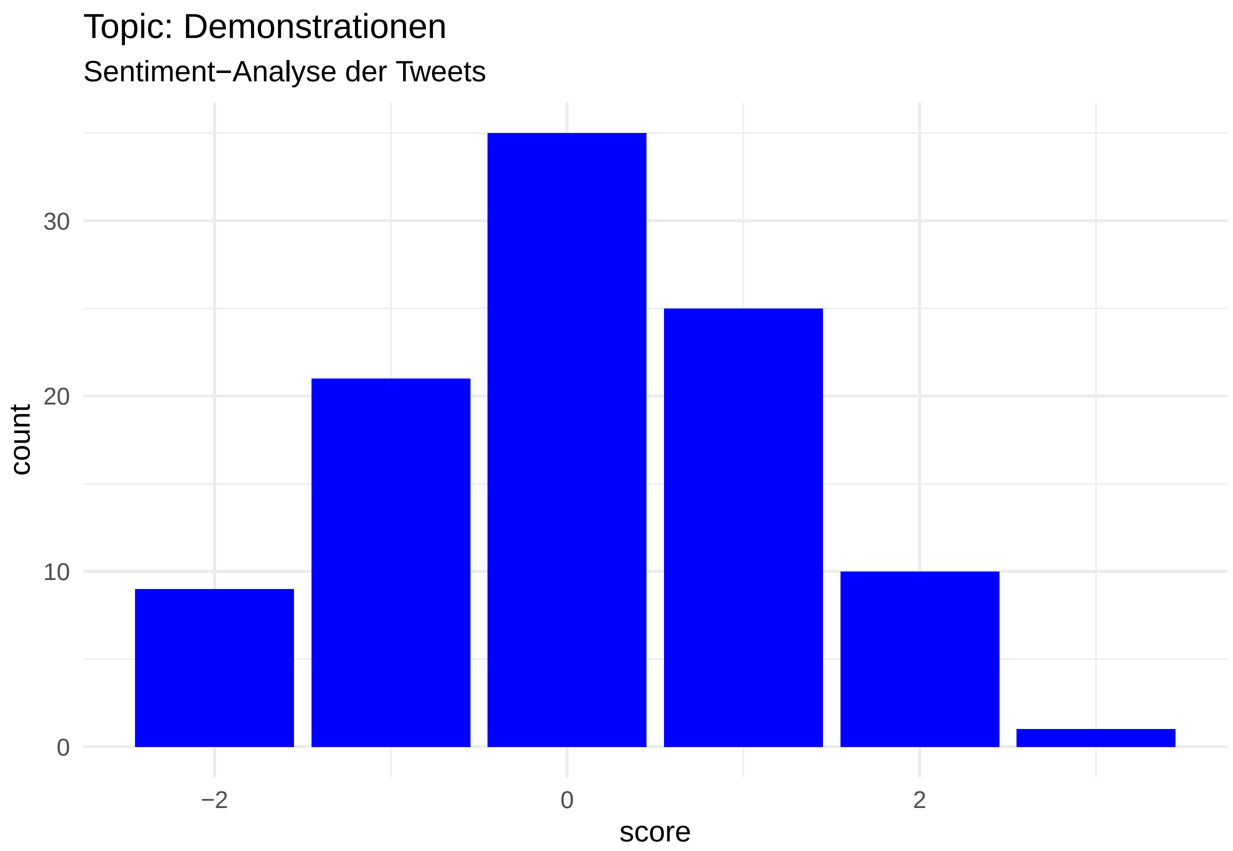
```r
    score = sum(pos.matches) - sum(neg.matches)
    return(score)
  },
  pos.words, neg.words, .progress=.progress )
  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}

score_pm_demo <- score.sentiment(pm_demo$content, pos.words, neg.words)
```

```
## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

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

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

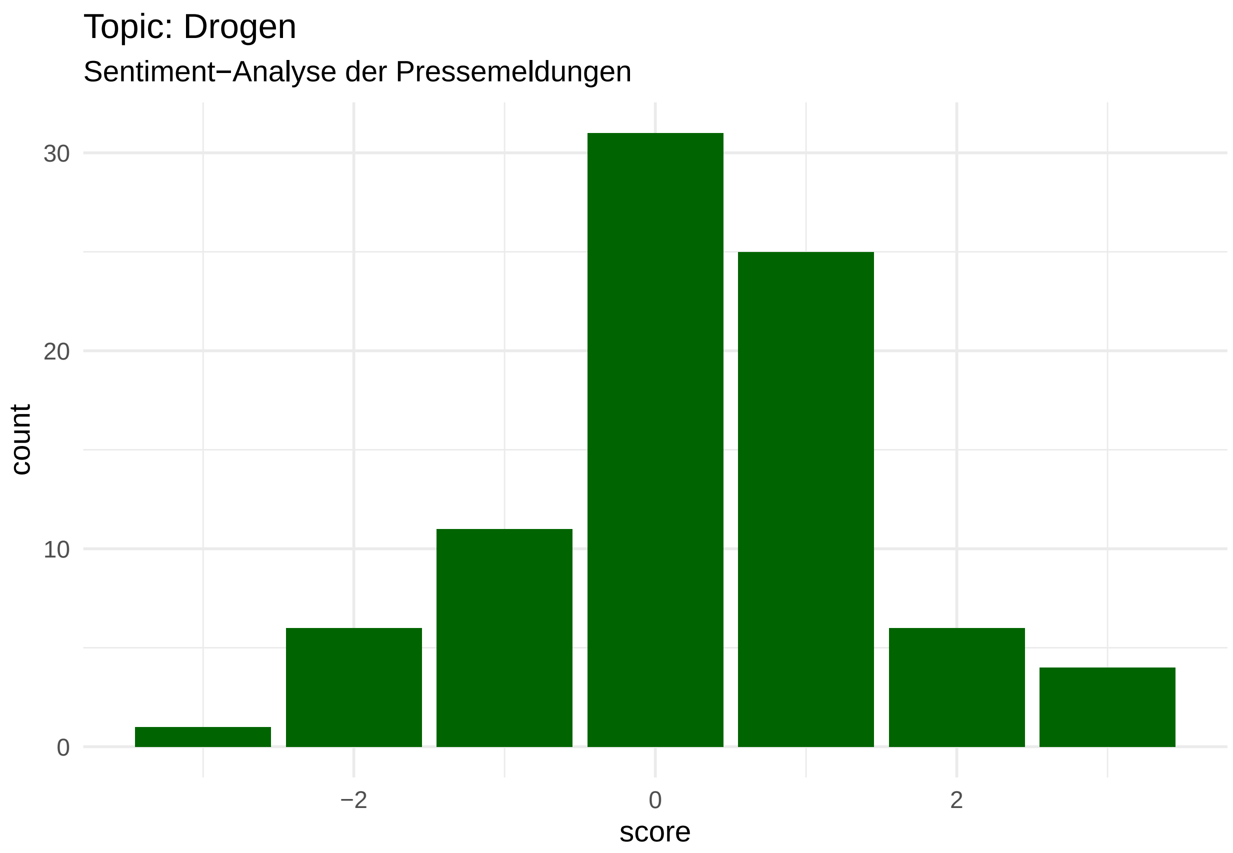
```r
score_tw_demo <- score.sentiment(tw_demo$tweet_text, pos.words, neg.words)

ggplot(score_pm_demo) +
  geom_bar(aes(x = score), fill = "blue") +
  labs(title = "Topic: Demonstrationen", subtitle = "Sentiment-Analyse der Pressemeldungen") +
  theme_minimal()
```



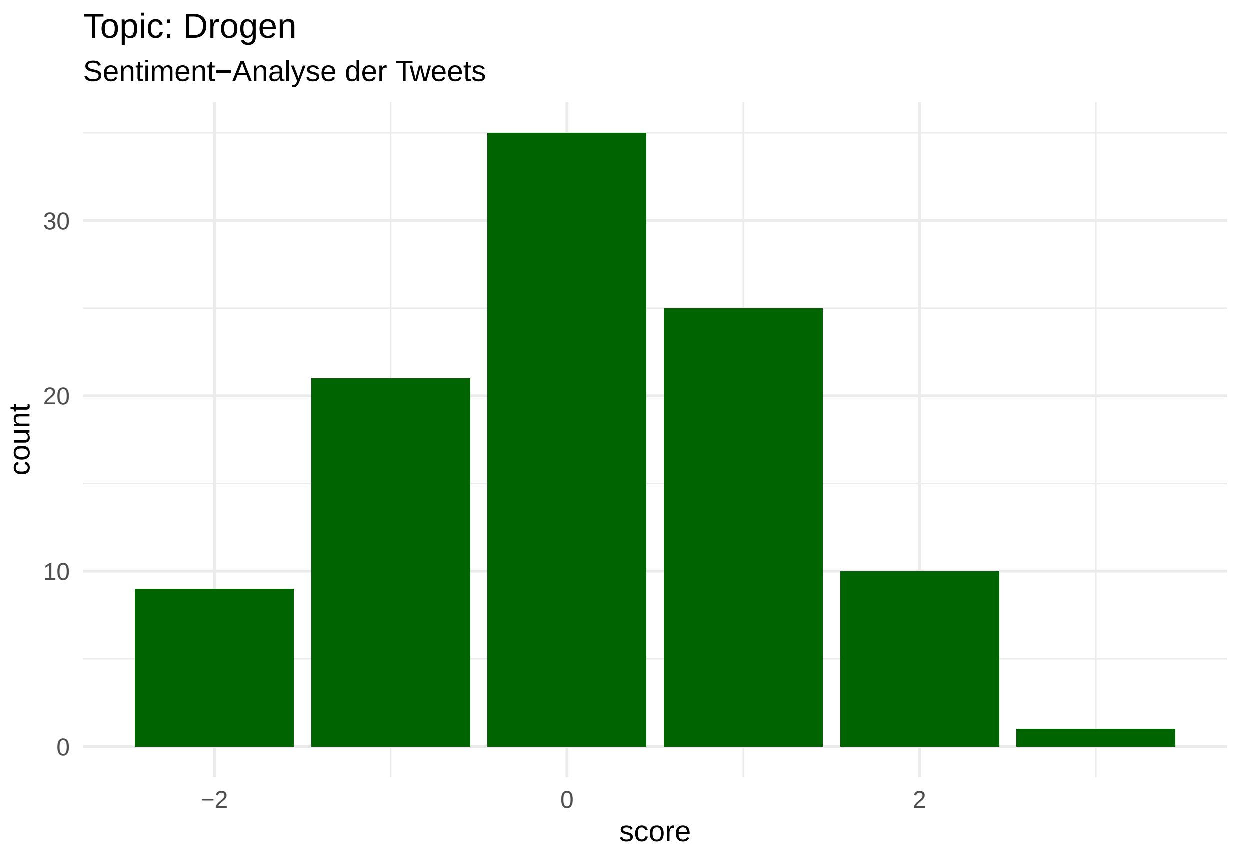
```r
ggplot(score_tw_demo) +
  geom_bar(aes(x = score), fill = "blue") +
  labs(title = "Topic: Demonstrationen", subtitle = "Sentiment-Analyse der Tweets") +
  theme_minimal()
```



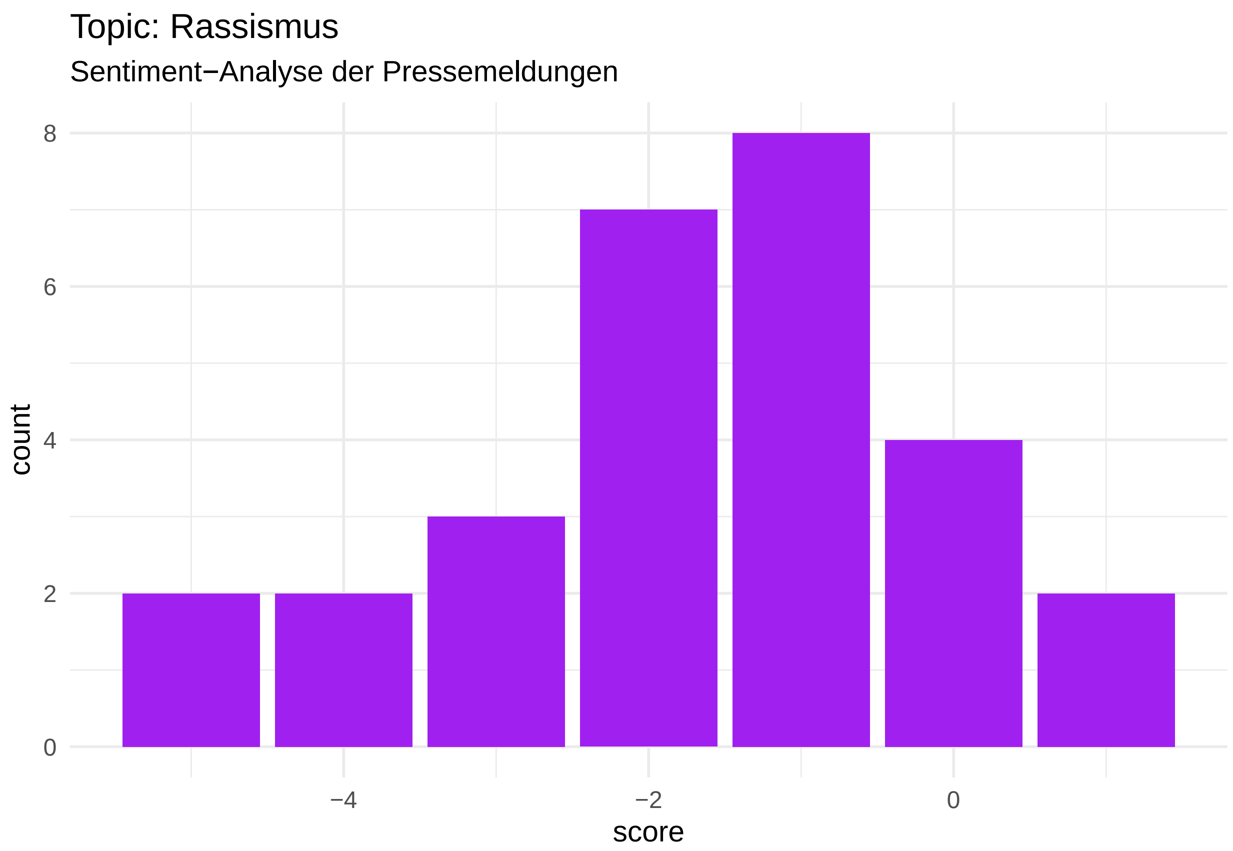
```r
score_pm_drogen <- score.sentiment(pm_demo$content, pos.words, neg.words)
score_tw_drogen <- score.sentiment(tw_demo$tweet_text, pos.words, neg.words)

ggplot(score_pm_drogen) +
  geom_bar(aes(x = score), fill = "darkgreen") +
  labs(title = "Topic: Drogen", subtitle = "Sentiment-Analyse der Pressemeldungen") +
  theme_minimal()
```



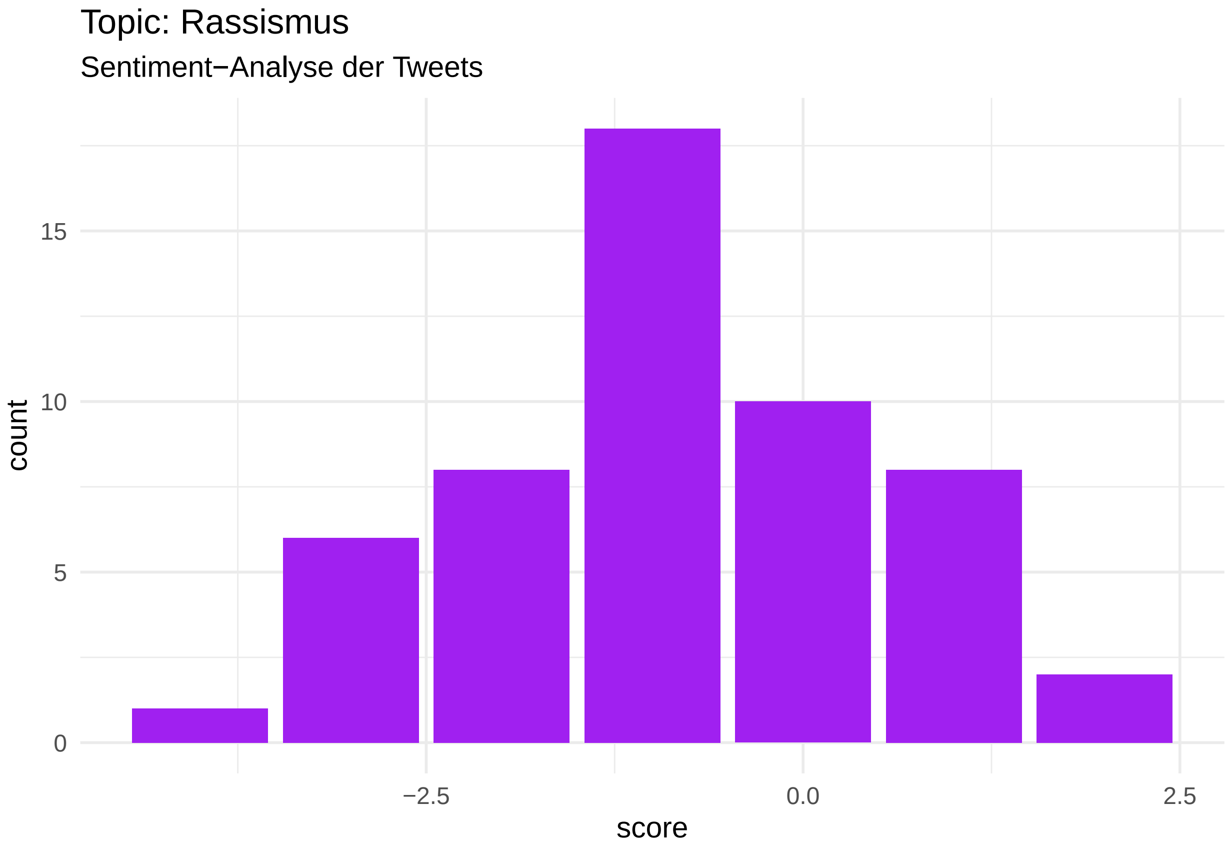
```r
ggplot(score_tw_drogen) +
  geom_bar(aes(x = score), fill = "darkgreen") +
  labs(title = "Topic: Drogen", subtitle = "Sentiment-Analyse der Tweets") +
  theme_minimal()
```



```r
score_pm_rass <- score.sentiment(pm_rass$content, pos.words, neg.words)
score_tw_rass <- score.sentiment(tw_rass$tweet_text, pos.words, neg.words)

ggplot(score_pm_rass) +
  geom_bar(aes(x = score), fill = "purple") +
  labs(title = "Topic: Rassismus", subtitle = "Sentiment-Analyse der Pressemeldungen") +
  theme_minimal()
```



```
ggplot(score_tw_rass) +
  geom_bar(aes(x = score), fill = "purple") +
  labs(title = "Topic: Rassismus", subtitle = "Sentiment-Analyse der Tweets") +
  theme_minimal()
```



```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Germany.1252  LC_CTYPE=German_Germany.1252
## [3] LC_MONETARY=German_Germany.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] plyr_1.8.6      stringi_1.5.3   forcats_0.5.0   stringr_1.4.0
##  [5] dplyr_1.0.0     purrr_0.3.4     readr_1.3.1     tidyr_1.1.0
##  [9] tibble_3.1.0    ggplot2_3.3.3   tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.0  xfun_0.22         haven_2.3.1       lattice_0.20-41
##  [5] colorspace_2.0-0  vctrs_0.3.7       generics_0.0.2    htmltools_0.5.1.1
##  [9] yaml_2.2.1        utf8_1.2.1        blob_1.2.1        rlang_0.4.10
## [13] pillar_1.6.0      withr_2.4.1       glue_1.4.2        DBI_1.1.0
## [17] dbplyr_1.4.4      modelr_0.1.8      readxl_1.3.1      lifecycle_1.0.0
## [21] munsell_0.5.0     gtable_0.3.0      cellranger_1.1.0  rvest_0.3.5
## [25] evaluate_0.14     labeling_0.4.2    knitr_1.31        fansi_0.4.2
## [29] highr_0.8         broom_0.5.6       Rcpp_1.0.6        backports_1.2.1
## [33] scales_1.1.1      jsonlite_1.7.2    farver_2.1.0      fs_1.4.1
## [37] hms_0.5.3         digest_0.6.27     grid_4.0.5        cli_2.4.0
## [41] tools_4.0.5       magrittr_2.0.1    crayon_1.4.1      pkgconfig_2.0.3
## [45] ellipsis_0.3.1    xml2_1.3.2        reprex_0.3.0      lubridate_1.7.9
## [49] rstudioapi_0.13   assertthat_0.2.1  rmarkdown_2.7     httr_1.4.2
## [53] R6_2.5.0          nlme_3.1-152      compiler_4.0.5
```