

# 615hw4

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(usethis)
library(devtools)

devtools::install_github("Truenumbers/tnum/tnum")

## Skipping install of 'tnum' from a github remote, the SHA1 (dca65088) has not changed since last inst
## Use `force = TRUE` to force installation

library(tnum)
library(kableExtra)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:kableExtra':
##
## group_rows

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

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

library(gutenbergr)
library(tidytext)
library(textdata)
library(janeaustenr)
library(stringr)
library(tidyr)

gut<-gutenberg_works()
gd <- gutenberg_download(158)

## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
## Using mirror http://aleph.gutenberg.org

get_sentiments("afinn")

## # A tibble: 2,477 x 2
```

```
##      word      value
##      <chr>      <dbl>
## 1 abandon      -2
## 2 abandoned     -2
## 3 abandons      -2
## 4 abducted      -2
## 5 abduction     -2
## 6 abductions    -2
## 7 abhor         -3
## 8 abhorred      -3
## 9 abhorrent     -3
## 10 abhors       -3
## # ... with 2,467 more rows
```

```
get_sentiments("bing")
```

```
## # A tibble: 6,786 x 2
##      word      sentiment
##      <chr>      <chr>
## 1 2-faces      negative
## 2 abnormal     negative
## 3 abolish      negative
## 4 abominable   negative
## 5 abominably   negative
## 6 abominate    negative
## 7 abomination  negative
## 8 abort        negative
## 9 aborted      negative
## 10 aborts      negative
## # ... with 6,776 more rows
```

```
get_sentiments("nrc")
```

```
## # A tibble: 13,875 x 2
##      word      sentiment
##      <chr>      <chr>
## 1 abacus      trust
## 2 abandon     fear
## 3 abandon     negative
## 4 abandon     sadness
## 5 abandoned   anger
## 6 abandoned   fear
## 7 abandoned   negative
## 8 abandoned   sadness
## 9 abandonment anger
## 10 abandonment fear
## # ... with 13,865 more rows
```

```
tidy_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,
                                regex("^chapter [\\divxlc]",
                                      ignore_case = TRUE)))) %>%
  ungroup() %>%
```

```

unnest_tokens(word, text)

nrc_joy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)

## Joining, by = "word"

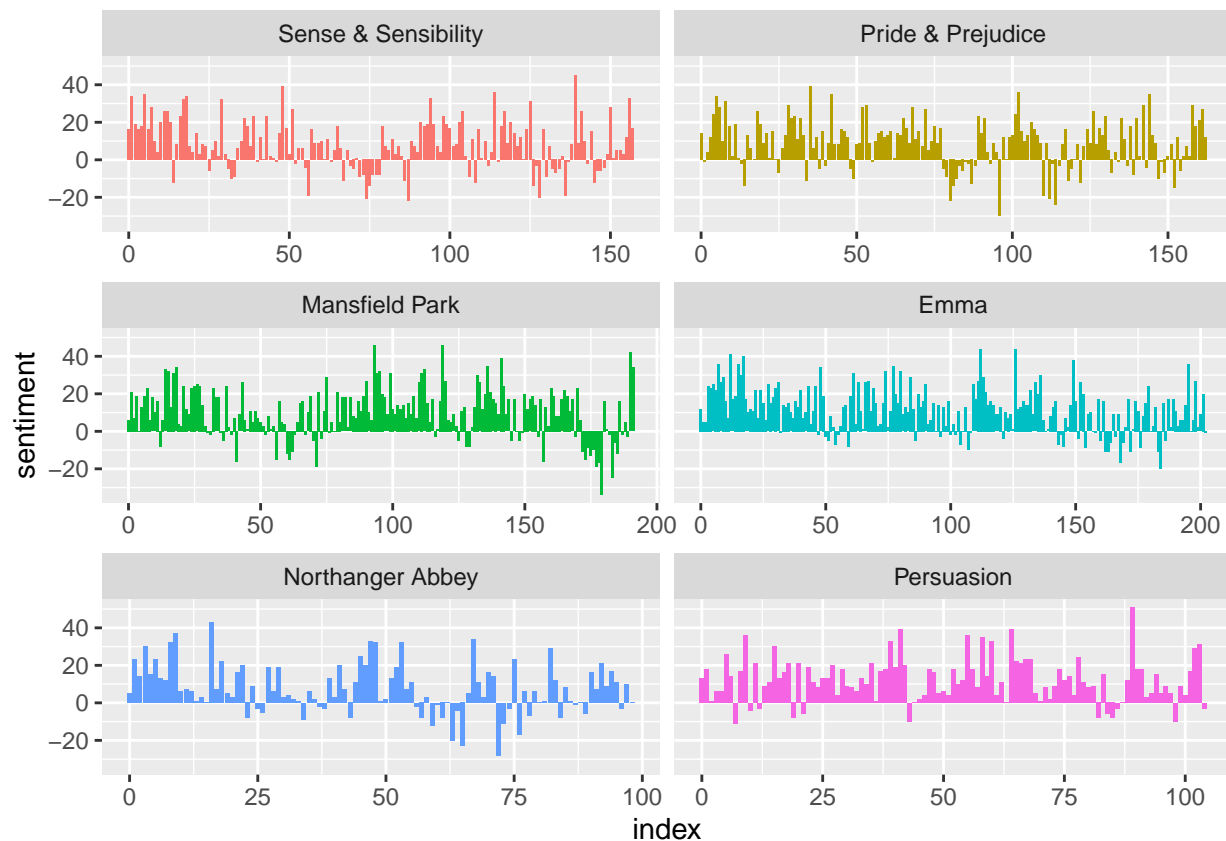
## # A tibble: 301 x 2
##   word      n
##   <chr>   <int>
## 1 good     359
## 2 friend   166
## 3 hope     143
## 4 happy    125
## 5 love     117
## 6 deal      92
## 7 found     92
## 8 present   89
## 9 kind      82
## 10 happiness 76
## # ... with 291 more rows

jane_austen_sentiment <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = linenumbers %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)

## Joining, by = "word"

library(ggplot2)
ggplot(jane_austen_sentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_x")

```



```
library(sentimentr)
sentiment(gd$text)
```

```
## Warning: Each time `sentiment` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory. It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.
```

```
##      element_id sentence_id word_count  sentiment
##  1:           1           1           1 0.00000000
##  2:           2           1          NA 0.00000000
##  3:           3           1           3 0.00000000
##  4:           4           1          NA 0.00000000
##  5:           5           1          NA 0.00000000
##  ---
## 21208:      16484           1          11 0.34673805
## 21209:      16485           1          12 -0.07216878
## 21210:      16486           1          11 0.75377836
## 21211:      16487           1          NA 0.00000000
## 21212:      16488           1           1 0.00000000
```

```
sentiment_by(gd$text, by = NULL)
```

```
## Warning: Each time `sentiment_by` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory. It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.
```

```
##      element_id word_count sd ave_sentiment
```

```
##      1:      1      1 NA    0.00000000
##      2:      2      0 NA    0.00000000
##      3:      3      3 NA    0.00000000
##      4:      4      0 NA    0.00000000
##      5:      5      0 NA    0.00000000
##    ---
## 16484:    16484    11 NA    0.34673805
## 16485:    16485    12 NA   -0.07216878
## 16486:    16486    11 NA    0.75377836
## 16487:    16487     0 NA    0.00000000
## 16488:    16488     1 NA    0.00000000
```

```
profanity(gd$text)
```

```
## Warning: Each time `profanity` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory. It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```
## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored
```

```

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

## Warning in gsub(pattern[i], replacement[i], text.var, fixed = fixed, ..., :
## argument 'perl = TRUE' will be ignored

##      element_id sentence_id word_count profanity_count profanity
##      1:          1           1           1              0          0
##      2:          2           1           0              0          0
##      3:          3           1           3              0          0
##      4:          4           1           0              0          0
##      5:          5           1           0              0          0
##      ---
## 21208:       16484           1          11              0          0
## 21209:       16485           1          12              0          0
## 21210:       16486           1          11              0          0
## 21211:       16487           1           0              0          0
## 21212:       16488           1           1              0          0

debates <- gd
debates_with_pol <- debates %>%
  get_sentences() %>%
  sentiment() %>%
  mutate(polarity_level = ifelse(sentiment < 0.2, "Negative",
                                ifelse(sentiment > 0.2, "Positive", "Neutral")))
debates_with_pol %>% filter(polarity_level == "Negative") %>% View()
debates_with_senti <- ggplot() + geom_boxplot(aes(y = person, x = sentiment))
debates %>%
  get_sentences() %>%
  sentiment_by(by = NULL) %>% #View()
  ggplot() + geom_density(aes(ave_sentiment))

```

