TEXT MINING

WITH TIDY DATA PRINCIPLES

TIDYTEXT

HELLO

I'm Julia Silge

Data Scientist at Stack Overflow

@juliasilge

https://juliasilge.com/

TEXT DATA IS INCREASINGLY IMPORTANT

TEXT DATA IS INCREASINGLY IMPORTANT

NLP TRAINING IS SCARCE ON THE GROUND

TIDYTEXT

TIDY DATA PRINCIPLES
+
COUNT-BASED METHODS
=
🎉

tidytext: Text mining using dplyr, ggplot2, and other tidy tools

Authors: Julia Silge, David Robinson
License: MIT

build passing | build passing | CRAN 0.1.4 | coverage 82% | DOI 10.5281/zenodo.814233 | JOSS 10.21105/joss.00037

Using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use. Much of the infrastructure needed for text mining with tidy data frames already exists in packages like dplyr, broom, tidyr and ggplot2. In this package, we provide functions and supporting data sets to allow conversion of text to and from tidy formats, and to switch seamlessly between tidy tools and existing text mining packages.

https://github.com/juliasilge/tidytext

tidytext: Text mining using dplyr, ggplot2, and other tidy tools

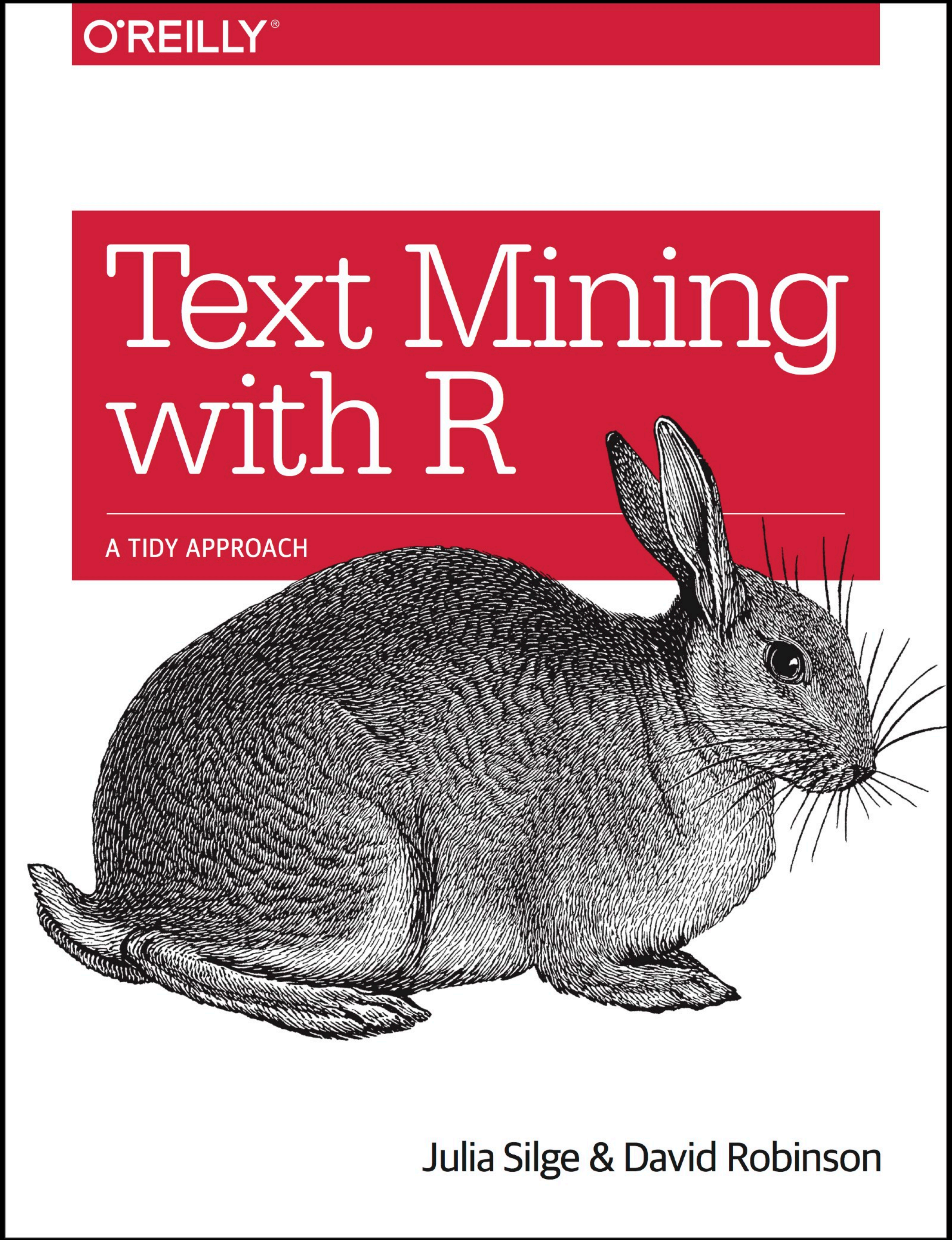
Authors: Julia Silge, David Robinson
License: MIT

build passing | build passing | CRAN 0.1.4 | coverage 82% | DOI 10.5281/zenodo.814233 | JOSS 10.21105/joss.00037

Using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use. Much of the infrastructure needed for text mining with tidy data frames already exists in packages like dplyr, broom, tidyr and ggplot2. In this package, we provide functions and supporting data sets to allow conversion of text to and from tidy formats, and to switch seamlessly between tidy tools and existing text mining packages.

tidyverse

https://github.com/juliasilge/tidytext



http://tidytextmining.com/

TIDYTEXT

WHAT DO WE MEAN BY TIDY TEXT?

WHAT DO WE MEAN BY TIDY TEXT?

```
> text <- c("Because I could not stop for Death -",
+           "He kindly stopped for me -",
+           "The Carriage held but just Ourselves -",
+           "and Immortality")
>
> text
[1] "Because I could not stop for Death -"
[2] "He kindly stopped for me -"
[3] "The Carriage held but just Ourselves -"
[4] "and Immortality"
```

WHAT DO WE MEAN BY TIDY TEXT?

```
> library(tidytext)
> text_df %>%
+     unnest_tokens(word, text)

# A tibble: 20 x 2
    line word
   <int> <chr>
 1     1 because
 2     1 i
 3     1 could
 4     1 not
 5     1 stop
 6     1 for
 7     1 death
 8     2 he
 9     2 kindly
10     2 stopped
11     2 for
12     2 me
13     3 the
```

WHAT DO WE MEAN BY TIDY TEXT?

```
> library(tidytext)
> text_df %>%
+     unnest_tokens(word, text)

# A tibble: 20 x 2
    line word
   <int> <chr>
 1     1 because
 2     1 i
 3     1 could
 4     1 not
 5     1 stop
 6     1 for
 7     1 death
 8     2 he
 9     2 kindly
10     2 stopped
11     2 for
12     2 me
```

- **Other columns have been retained**
- **Punctuation has been stripped**
- **Words have been converted to lowercase**

WHAT DO WE MEAN BY TIDY TEXT?

```
> tidy_books <- original_books %>%
+     unnest_tokens(word, text)
>
> tidy_books
# A tibble: 725,055 x 4
   book                linenumber chapter word
   <fct>                    <int>   <int> <chr>
 1 Sense & Sensibility          1       0 sense
 2 Sense & Sensibility          1       0 and
 3 Sense & Sensibility          1       0 sensibility
 4 Sense & Sensibility          3       0 by
 5 Sense & Sensibility          3       0 jane
 6 Sense & Sensibility          3       0 austen
 7 Sense & Sensibility          5       0 1811
 8 Sense & Sensibility         10       1 chapter
 9 Sense & Sensibility         10       1 1
10 Sense & Sensibility         13       1 the
# ... with 725,045 more rows
```

TIDYTEXT

OUR TEXT IS TIDY NOW

TIDYTEXT

OUR TEXT IS TIDY NOW

WHAT NEXT?

WORD FREQUENCIES

```
> read_csv("syndromicData_raw.csv") %>%
+   unnest_tokens(word, Chief.Complaint) %>%
+   count(word, sort = TRUE)
# A tibble: 3,088 x 2
   word        n
   <chr>   <int>
 1 pain    34047
 2 inj     11035
 3 back     8053
 4 injury   7950
 5 left     7937
 6 rt       7628
 7 r        7497
 8 l        7257
 9 right    7067
10 lac      7025
# ... with 3,078 more rows
```

TIDYTEXT

REMOVING STOP WORDS

```
> get_stopwords()
# A tibble: 175 x 2
   word      lexicon
   <chr>     <chr>
 1 i         snowball
 2 me        snowball
 3 my        snowball
 4 myself    snowball
 5 we        snowball
 6 our       snowball
 7 ours      snowball
 8 ourselves snowball
 9 you       snowball
10 your      snowball
# ... with 165 more rows
```

TIDYTEXT

REMOVING STOP WORDS

```
> get_stopwords(language = "es")
# A tibble: 308 x 2
   word  lexicon
   <chr> <chr>
 1 de    snowball
 2 la    snowball
 3 que   snowball
 4 el    snowball
 5 en    snowball
 6 y     snowball
 7 a     snowball
 8 los   snowball
 9 del   snowball
10 se    snowball
# ... with 298 more rows
```

TIDYTEXT

REMOVING STOP WORDS

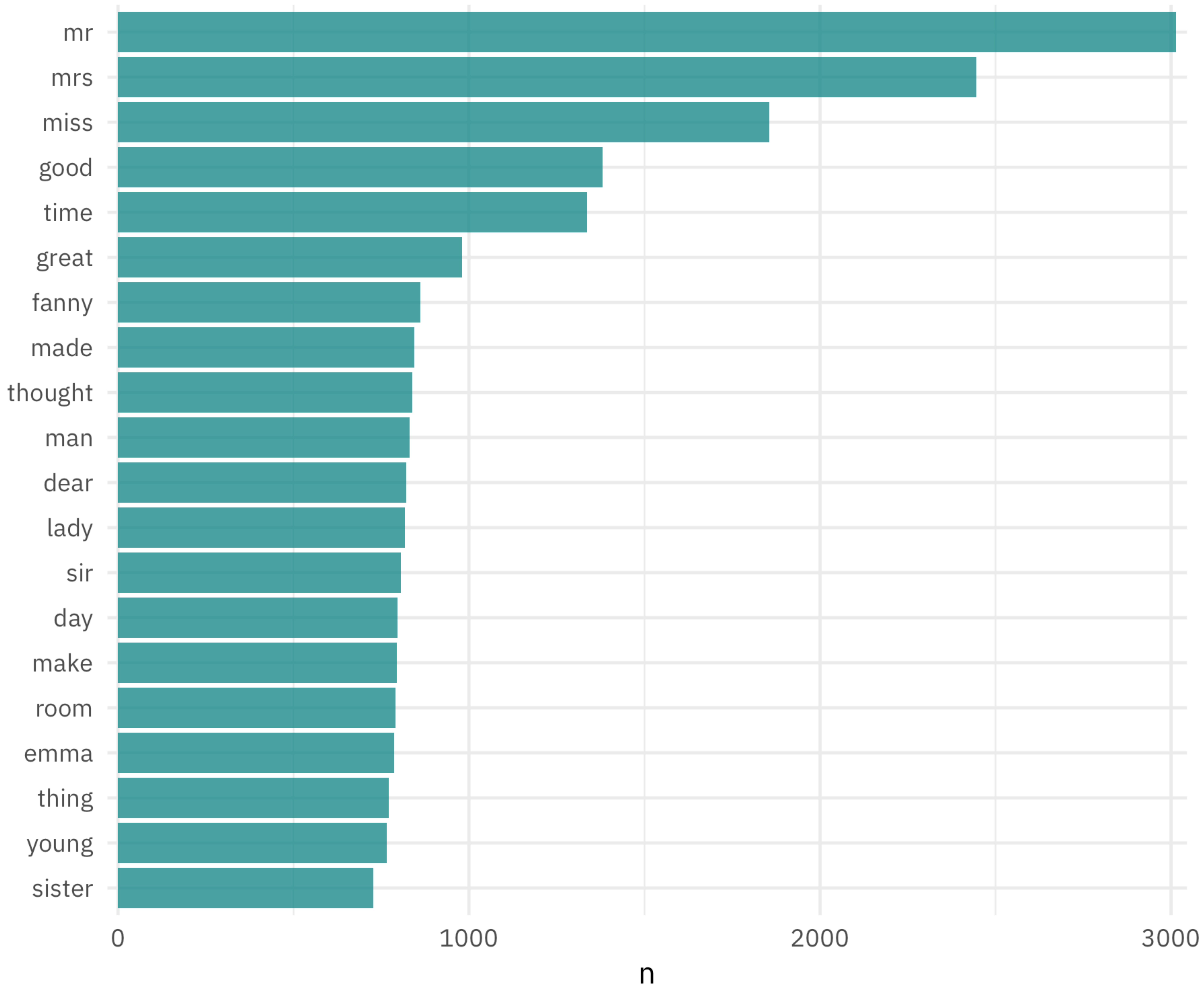
```
> get_stopwords(source = "smart")
# A tibble: 571 x 2
   word        lexicon
   <chr>       <chr>
 1 a           smart
 2 a's         smart
 3 able        smart
 4 about       smart
 5 above       smart
 6 according   smart
 7 accordingly smart
 8 across      smart
 9 actually    smart
10 after       smart
# ... with 561 more rows
```

TIDYTEXT

REMOVING STOP WORDS

```
tidy_books <- tidy_books %>%
    anti_join(get_stopwords(source = "smart"))

tidy_books %>%
    count(word, sort = TRUE)
```


mr
mrs
miss
good
time
great
fanny
made
thought
man
dear
lady
sir
day
make
room
emma
thing
young
sister
0
1000
2000
3000
n

TIDYTEXT

SENTIMENT ANALYSIS

```
> get_sentiments("afinn")
# A tibble: 2,476 x 2
   word       score
   <chr>      <int>
 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 2,466 more rows
```

TIDYTEXT

SENTIMENT ANALYSIS

```
> get_sentiments("bing")
# A tibble: 6,788 x 2
   word        sentiment
   <chr>       <chr>
 1 2-faced     negative
 2 2-faces     negative
 3 a+          positive
 4 abnormal    negative
 5 abolish     negative
 6 abominable  negative
 7 abominably  negative
 8 abominate   negative
 9 abomination negative
10 abort       negative
# ... with 6,778 more rows
```

TIDYTEXT

SENTIMENT ANALYSIS

```
> get_sentiments("nrc")
# A tibble: 13,901 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 13,891 more rows
```

TIDYTEXT

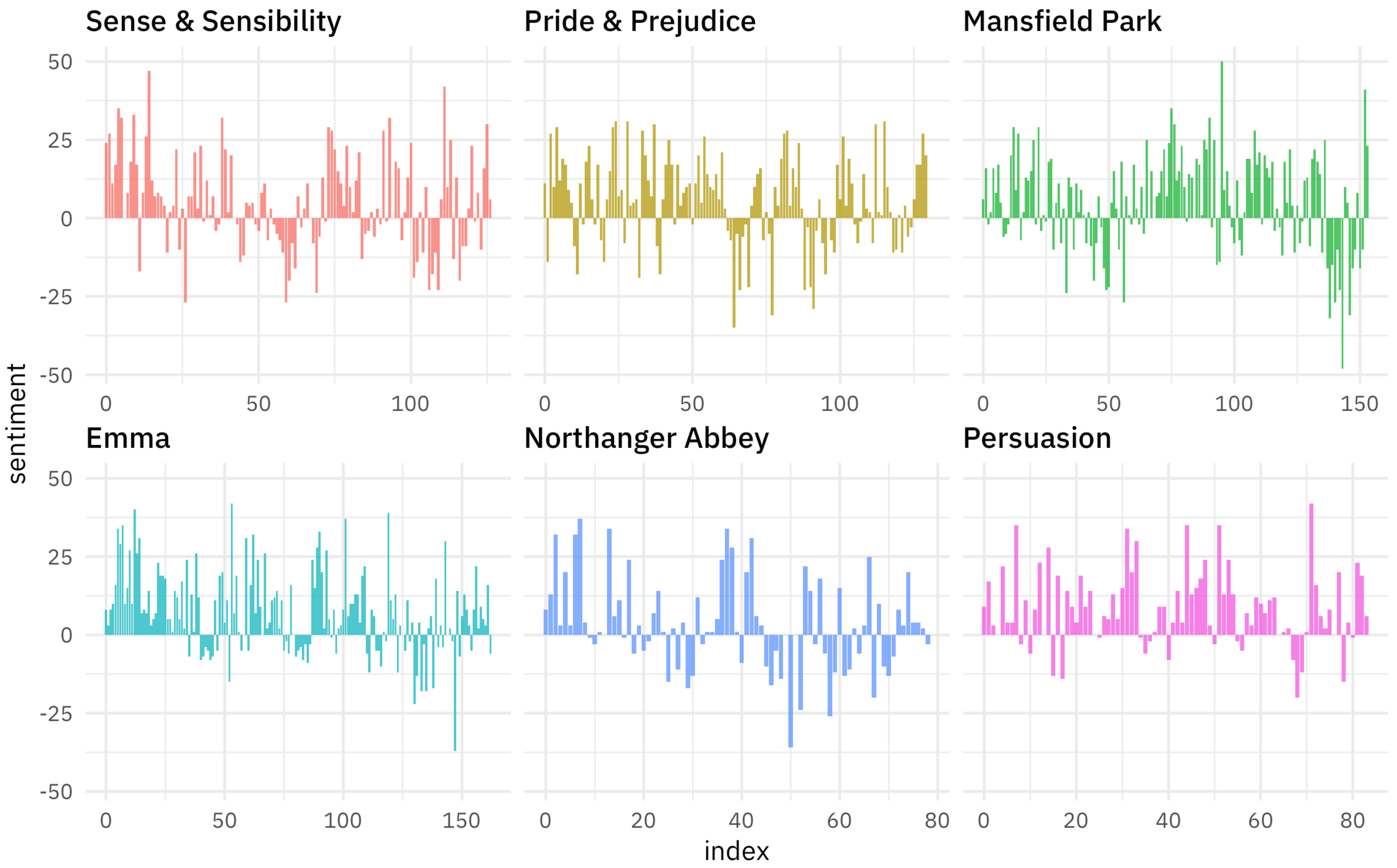
SENTIMENT ANALYSIS

```
> get_sentiments("loughran")
# A tibble: 4,149 x 2
   word         sentiment
   <chr>        <chr>
 1 abandon      negative
 2 abandoned    negative
 3 abandoning   negative
 4 abandonment  negative
 5 abandonments negative
 6 abandons     negative
 7 abdicated    negative
 8 abdicates    negative
 9 abdicating   negative
10 abdication   negative
# ... with 4,139 more rows
```

TIDYTEXT

SENTIMENT ANALYSIS

```
> janeaustensentiment <- tidy_books %>%
+     inner_join(get_sentiments("bing")) %>%
+     count(book, index = linenumber %/% 100, sentiment) %>%
+     spread(sentiment, n, fill = 0) %>%
+     mutate(sentiment = positive - negative)
```


Sense & Sensibility
Pride & Prejudice
Mansfield Park
Emma
Northanger Abbey
Persuasion
sentiment
index
50
25
0
-25
-50
0
50
100
150
20
40
60
80

TIDYTEXT

SENTIMENT ANALYSIS

Which words contribute to each sentiment?

```
> bing_word_counts <- austen_books() %>%
+     unnest_tokens(word, text) %>%
+     inner_join(get_sentiments("bing")) %>%
+     count(word, sentiment, sort = TRUE)
```

TIDYTEXT

SENTIMENT ANALYSIS

Which words contribute to each sentiment?

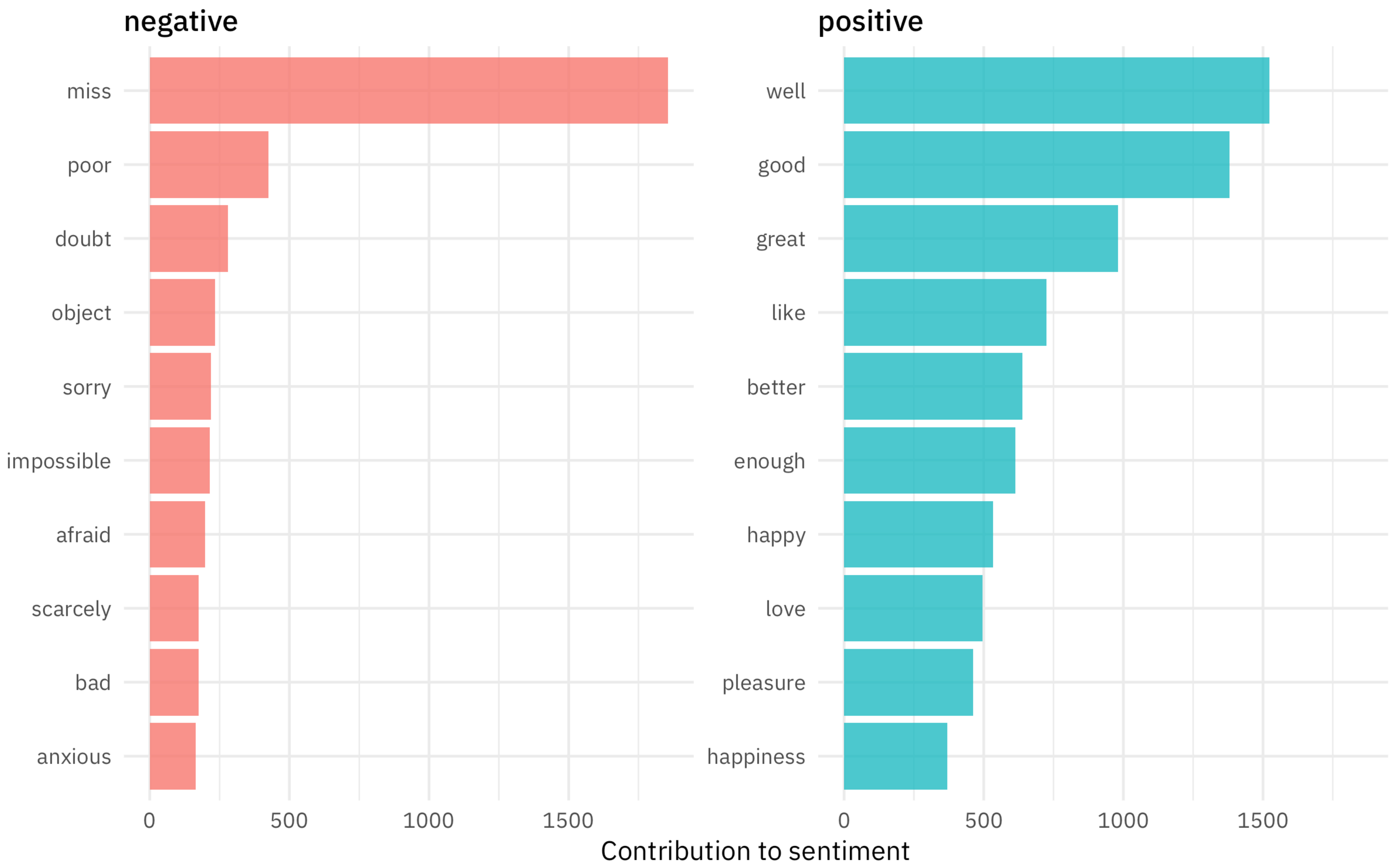
```
> bing_word_counts
# A tibble: 2,585 x 3
   word     sentiment     n
   <chr>    <chr>     <int>
 1 miss     negative   1855
 2 well     positive   1523
 3 good     positive   1380
 4 great    positive    981
 5 like     positive    725
 6 better   positive    639
 7 enough   positive    613
 8 happy    positive    534
 9 love     positive    495
10 pleasure positive    462
# ... with 2,575 more rows
```

TIDYTEXT

SENTIMENT ANALYSIS

Which words contribute to each sentiment?

```
> bing_word_counts
# A tibble: 2,585 x 3
   word      sentiment     n
   <chr>     <chr>     <int>
 1 miss      negative   1855
 2 well      positive   1523
 3 good      positive   1380
 4 great     positive    981
 5 like      positive    725
 6 better    positive    639
 7 enough    positive    613
 8 happy     positive    534
 9 love      positive    495
10 pleasure  positive    462
# ... with 2,575 more rows
```


negative
positive
miss
poor
doubt
object
sorry
impossible
afraid
scarcely
bad
anxious
well
good
great
like
better
enough
happy
love
pleasure
happiness
0
500
1000
1500
0
500
1000
1500
Contribution to sentiment

TIDYTEXT

WHAT IS A DOCUMENT ABOUT?

TERM FREQUENCY

INVERSE DOCUMENT FREQUENCY

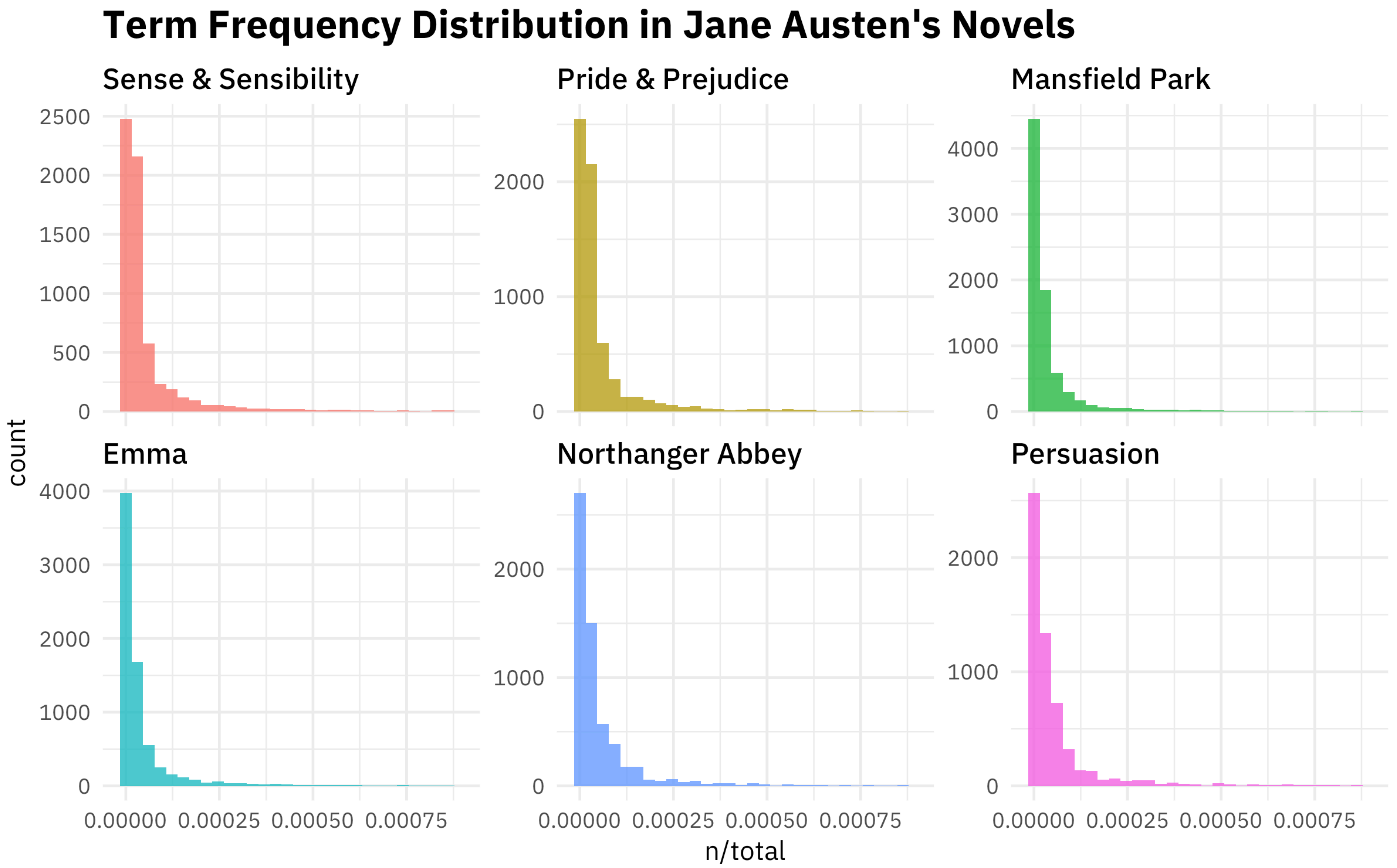
$$idf(\text{term}) = \ln\left(\frac{n_{\text{documents}}}{n_{\text{documents containing term}}}\right)$$

TF-IDF

```
> book_words <- austen_books() %>%
     unnest_tokens(word, text) %>%
     count(book, word, sort = TRUE)
>
> total_words <- book_words %>%
     group_by(book) %>%
     summarize(total = sum(n))
>
> book_words <- left_join(book_words, total_words)
```

TF-IDF

```
> book_words
# A tibble: 40,379 x 4
   book                word      n  total
   <fct>               <chr> <int>  <int>
 1 Mansfield Park      the    6206 160460
 2 Mansfield Park      to     5475 160460
 3 Mansfield Park      and    5438 160460
 4 Emma                to     5239 160996
 5 Emma                the    5201 160996
 6 Emma                and    4896 160996
 7 Mansfield Park      of     4778 160460
 8 Pride & Prejudice   the    4331 122204
 9 Emma                of     4291 160996
10 Pride & Prejudice   to     4162 122204
# ... with 40,369 more rows
```



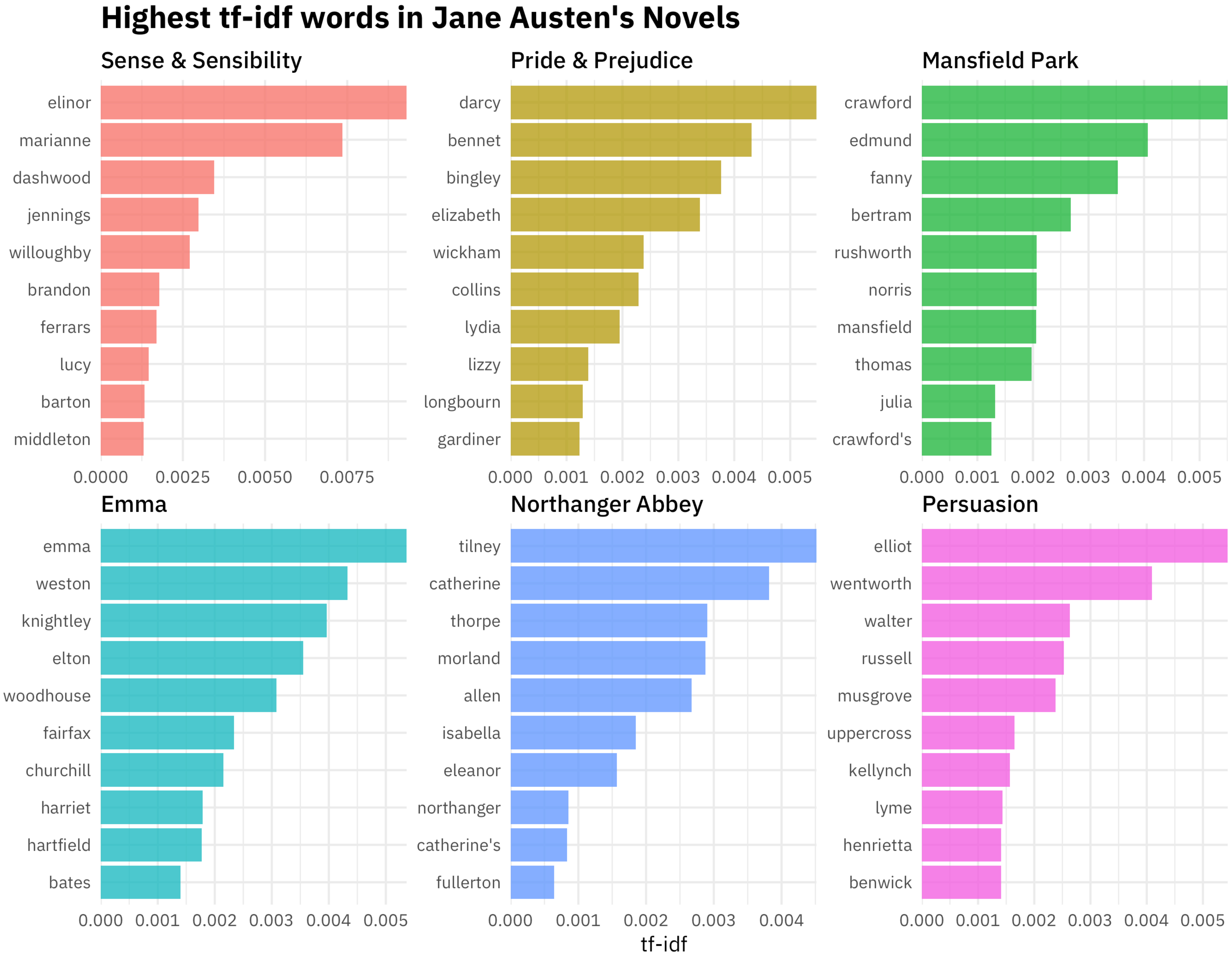
Term Frequency Distribution in Jane Austen's Novels
Sense & Sensibility
Pride & Prejudice
Mansfield Park
Emma
Northanger Abbey
Persuasion
count
n/total
0
500
1000
1500
2000
2500
3000
4000
0.00000
0.00025
0.00050
0.00075

TF-IDF

```
> book_words <- book_words %>%
+     bind_tf_idf(word, book, n)
> book_words
# A tibble: 40,379 x 7
   book                word      n  total     tf   idf tf_idf
   <fct>               <chr> <int>  <int>  <dbl> <dbl>  <dbl>
 1 Mansfield Park      the    6206 160460 0.0387     0      0
 2 Mansfield Park      to     5475 160460 0.0341     0      0
 3 Mansfield Park      and    5438 160460 0.0339     0      0
 4 Emma                to     5239 160996 0.0325     0      0
 5 Emma                the    5201 160996 0.0323     0      0
 6 Emma                and    4896 160996 0.0304     0      0
 7 Mansfield Park      of     4778 160460 0.0298     0      0
 8 Pride & Prejudice the    4331 122204 0.0354     0      0
 9 Emma                of     4291 160996 0.0267     0      0
10 Pride & Prejudice to     4162 122204 0.0341     0      0
# ... with 40,369 more rows
```

TF-IDF

```
> book_words %>%
+     arrange(desc(tf_idf))
# A tibble: 40,379 x 7
   book                word            n  total      tf   idf  tf_idf
   <fct>               <chr>       <int>  <int>   <dbl> <dbl>   <dbl>
 1 Sense & Sensibility elinor        623 119957 0.00519  1.79 0.00931
 2 Sense & Sensibility marianne      492 119957 0.00410  1.79 0.00735
 3 Mansfield Park      crawford      493 160460 0.00307  1.79 0.00551
 4 Pride & Prejudice   darcy         373 122204 0.00305  1.79 0.00547
 5 Persuasion          elliot        254  83658 0.00304  1.79 0.00544
 6 Emma                emma          786 160996 0.00488  1.10 0.00536
 7 Northanger Abbey    tilney        196  77780 0.00252  1.79 0.00452
 8 Emma                weston        389 160996 0.00242  1.79 0.00433
 9 Pride & Prejudice   bennet        294 122204 0.00241  1.79 0.00431
10 Persuasion          wentworth     191  83658 0.00228  1.79 0.00409
# ... with 40,369 more rows
```

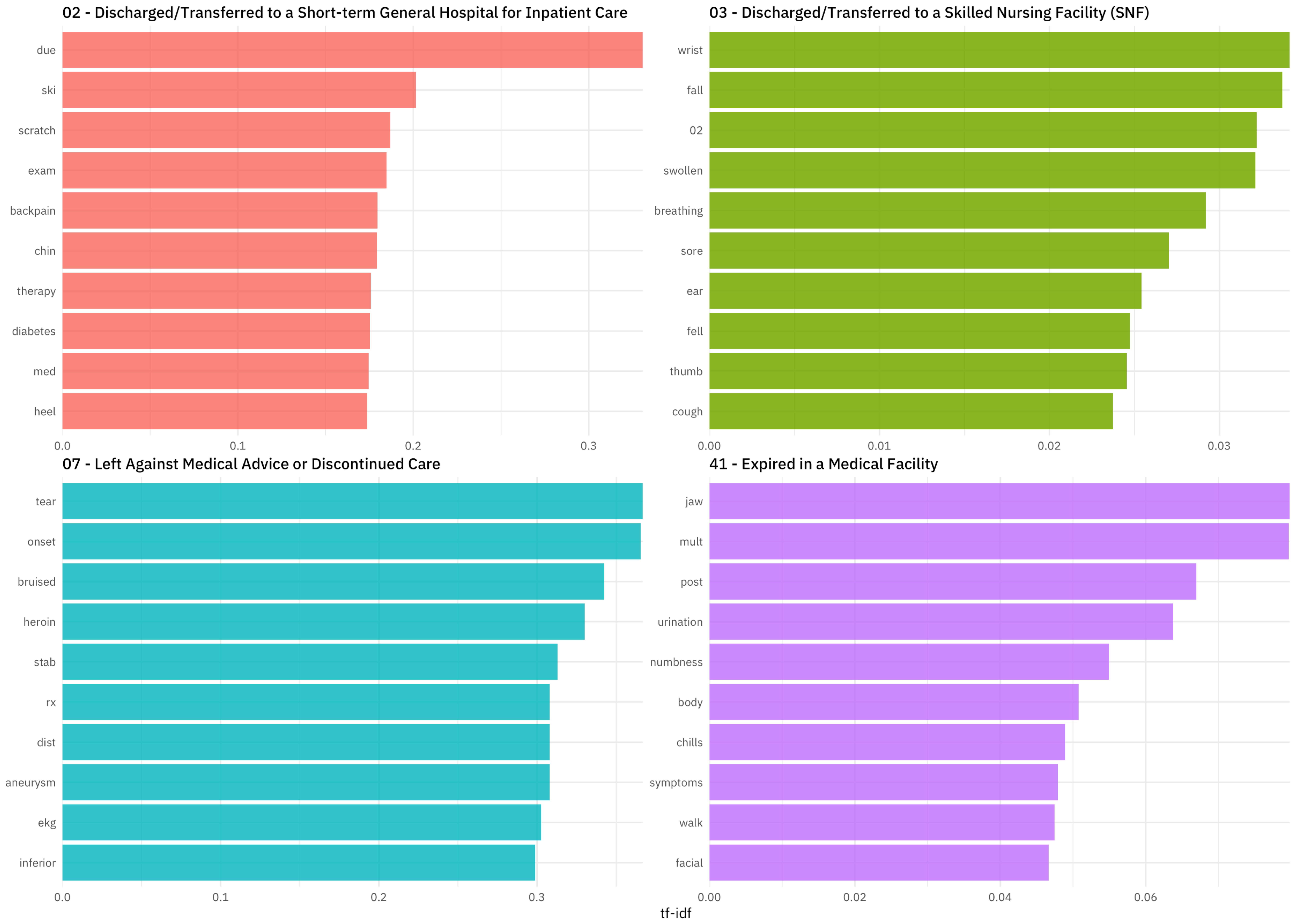


Highest tf-idf words in Jane Austen's Novels
Sense & Sensibility
elinor
marianne
dashwood
jennings
willoughby
brandon
ferrars
lucy
barton
middleton
0.0000 0.0025 0.0050 0.0075
Pride & Prejudice
darcy
bennet
bingley
elizabeth
wickham
collins
lydia
lizzy
longbourn
gardiner
0.000 0.001 0.002 0.003 0.004 0.005
Mansfield Park
crawford
edmund
fanny
bertram
rushworth
norris
mansfield
thomas
julia
crawford's
0.000 0.001 0.002 0.003 0.004 0.005
Emma
emma
weston
knightley
elton
woodhouse
fairfax
churchill
harriet
hartfield
bates
0.000 0.001 0.002 0.003 0.004 0.005
Northanger Abbey
tilney
catherine
thorpe
morland
allen
isabella
eleanor
northanger
catherine's
fullerton
0.000 0.001 0.002 0.003 0.004
Persuasion
elliot
wentworth
walter
russell
musgrove
uppercross
kellynch
lyme
henrietta
benwick
0.000 0.001 0.002 0.003 0.004 0.005
tf-idf

TIDYTEXT

TF-IDF

```
> complaint_words <- syndromic_cleaned %>%
+   unnest_tokens(word, Chief.Complaint) %>%
+   count(Discharge.Disposition, word, sort = TRUE) %>%
+   filter(n > 10,
+          !word %in% stop_words$word)
>
> complaint_tf_idf <- complaint_words %>%
+   bind_tf_idf(Discharge.Disposition, word, n)
>
> complaint_tf_idf
# A tibble: 3,416 x 6
   Discharge.Disposition                                word       n    tf   idf tf_idf
   <chr>                                                <chr>  <int> <dbl> <dbl>  <dbl>
 1 01- Discharge to Home or Self Care (Routine Discharge) pain   27196 0.799    0.     0.
 2 01- Discharge to Home or Self Care (Routine Discharge) inj     8789 0.798    0.     0.
 3 01- Discharge to Home or Self Care (Routine Discharge) injury  6347 0.800    0.     0.
 4 01- Discharge to Home or Self Care (Routine Discharge) left    6280 0.793    0.     0.
 5 01- Discharge to Home or Self Care (Routine Discharge) rt      6094 0.802    0.     0.
 6 01- Discharge to Home or Self Care (Routine Discharge) lac     5597 0.800    0.     0.
```



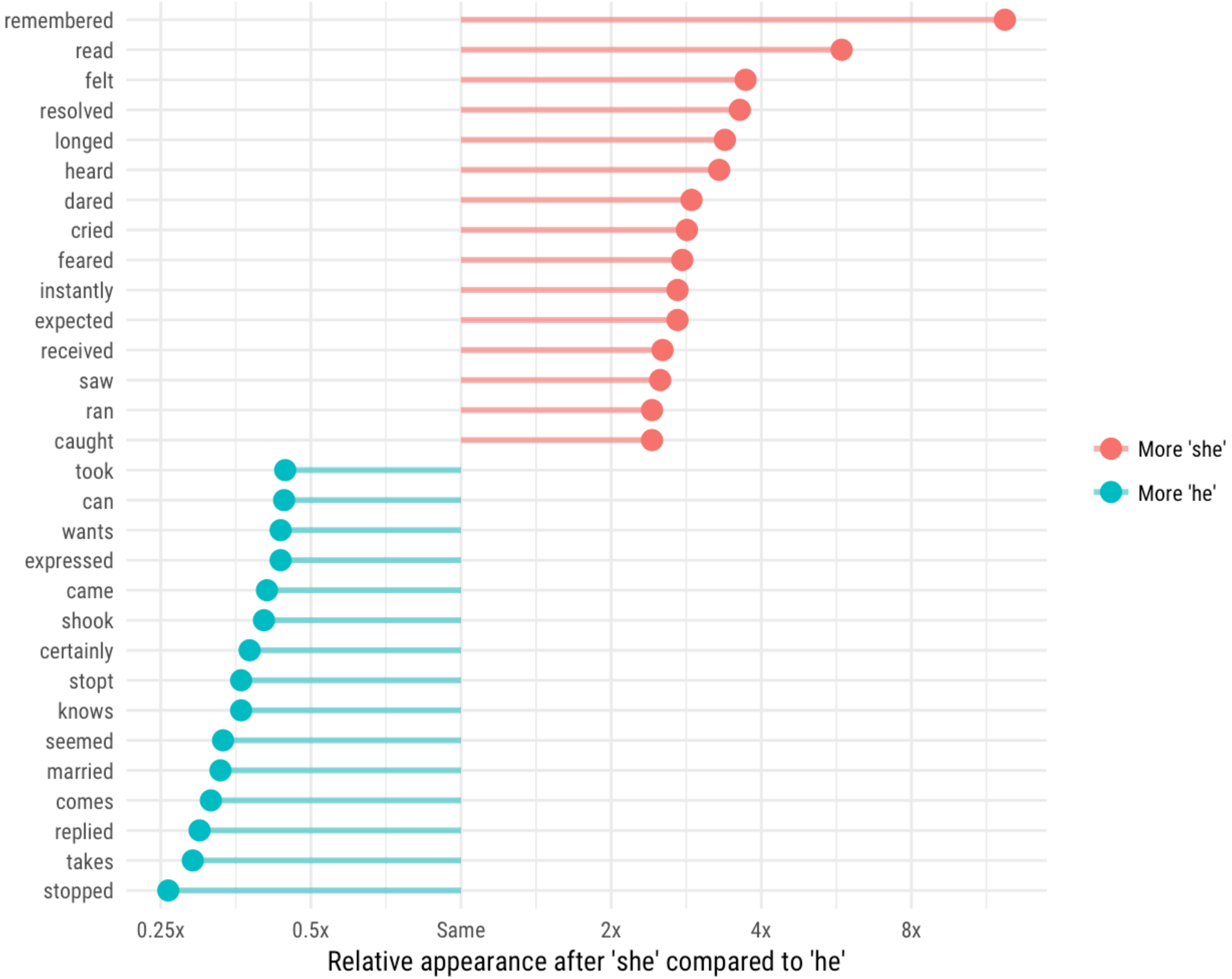
02 - Discharged/Transferred to a Short-term General Hospital for Inpatient Care
due
ski
scratch
exam
backpain
chin
therapy
diabetes
med
heel
0.0
0.1
0.2
0.3
03 - Discharged/Transferred to a Skilled Nursing Facility (SNF)
wrist
fall
02
swollen
breathing
sore
ear
fell
thumb
cough
0.00
0.01
0.02
0.03
07 - Left Against Medical Advice or Discontinued Care
tear
onset
bruised
heroin
stab
rx
dist
aneurysm
ekg
inferior
0.0
0.1
0.2
0.3
41 - Expired in a Medical Facility
jaw
mult
post
urination
numbness
body
chills
symptoms
walk
facial
0.00
0.02
0.04
0.06
tf-idf

TAKING TIDY TEXT TO THE NEXT LEVEL

N-GRAMS, NETWORKS, & NEGATION

Words paired with 'he' and 'she' in Jane Austen's novels

Women remember, read, and feel while men stop, take, and reply




ThePudding
She Giggles, He Gallops
Analyzing gender tropes in film with screen direction from 2,000 scripts.
By Julia Silge
+
Russell Goldenberg Amber Thomas Hanah Anderson

The top 800 words paired with "she" or "he"

Underlined words contain examples of their usage in screen direction.

EVEN

MORE "SHE" MORE "HE"

snuggles giggles squeals sobs weeps blushes clings rocks shrieks hugs shrinks gasps responds trembles pets flinches arches skips utters shudders startles buries swats murmurs resists

hovers caresses awakens shivers screams dances beats absently flees cleans stirs straddles cries moans bites realises mouths accepts wore smiles laughs wrote serves scoots liked

arranges scampers storms twirls softens ignores softly faints wonders fades sags hesitates casts applies hisses fiddles kisses sings awkwardly smokes stretches sips unbuttons stiffens

hurriedly hurries dries look ... early refuses tiptoes lingers beams pivots curls glides strokes meant abruptly retrieves bursts

She twirls, looking at herself in the mirror...

She twirls toward the door, grabbing her purse.

She twirls off. He chases her, beer and entries in hand.

She twirls a wooden katana as part of a floor routine.

She twirls a seductive finger around his tie. They kiss.

She twirls her finger around Milo's palm.

She twirls the string of her basketball shorts...

rakes relents reluctantly fr ... trails frowns retreats gonna licks touches reacts nearly sighs backs embraces squirms panics

yelps ends allows flashes ... rs shakes instinctively replies told freezes resumes creeps calms gives rushes sails tentatively

thrashes becomes types a ... s blinks dabs meets rests regards tilts attacks darts eyes brushes descends gently nervously

returns forces closes fixes ... s halts wakes bolts slaps fumbles quickly wears clasps faces feeds barely shrugs believes

floats left whacks blows i ... slices runs leans sounds washes swallows cranes observes accidentally marches rifles

squeezes begins kneels tu ... grimaces frees put calls glares tucks like plops scurries whispers tries remains actually

continues pinches tells lets yawns disappears heads looks chokes discovers plugs springs watches cautiously opens clutches studies dropped massages obeys suddenly loves scowls

crosses packs scribbles spits sneaks puts likes just lashes topples hangs lies angles starts claps adds flicks slowly cups angrily really stops pours jabs traces unzips crawls died

grasps slugs steadies breathes glances pushes directs inches hands reads comes unfolds winds attempts rubs snorts walks drifts sinks hides goes swivels feels keeps sprays sways

means ducks races repeats used steps sends finishes talking trips locks waits gets snatches chooses obviously takes decides plucks slides moves peers holds claws already stomps

swipes asks admires met said stands buttons owns says sets strips must wanted focuses fell unwraps edges unlocks remembers shines reaches jumps always places climbs stumbles

handles leafs needed passed surreptitiously concentrates helps interrupts reels scrambles steels blocks clenches floors gags splashes rips notices knocks enters finally rises listens

quietly jerks bumps wants lays kicks flies makes picks throws casually scans winces might dangles hefts flails waves dresses realizes passes catches plants thinks knows rings empties

hears sweeps may signs wrenches swims shares started recovers hops props tightens indicates hear dashes got peels will silently probably grows whips finds spins pulls switches

lights assumes ties digs hopes wheels lines performs settles tenses sniffs stabs wanders downs pounds notes slams twists shows weaves bends bounces curses expects hastily pokes

can sees acts scrolls brings arrives needs follows get zips manages proceeds deposits hardly strikes exhales still smells came tears yanks lifts knew shifts presses grabs excuses

straightens hustles speeds recognizes carries pays also falls stays tastes drags never speaks dials turned slows relaxes brought dumps sticks changes know come lowers flips bought

sleeps greets registers succeeds now even withdraws cracks writhes saw nudges scratches hobbles steals pauses collapses offers puffs grinds braces thought expected levels gave hits

drops spots lurches clambers ever eventually snaps writes searches gazes approaches selects eases dips cocks groans lives works winks swerves grips fills rummages loved joins

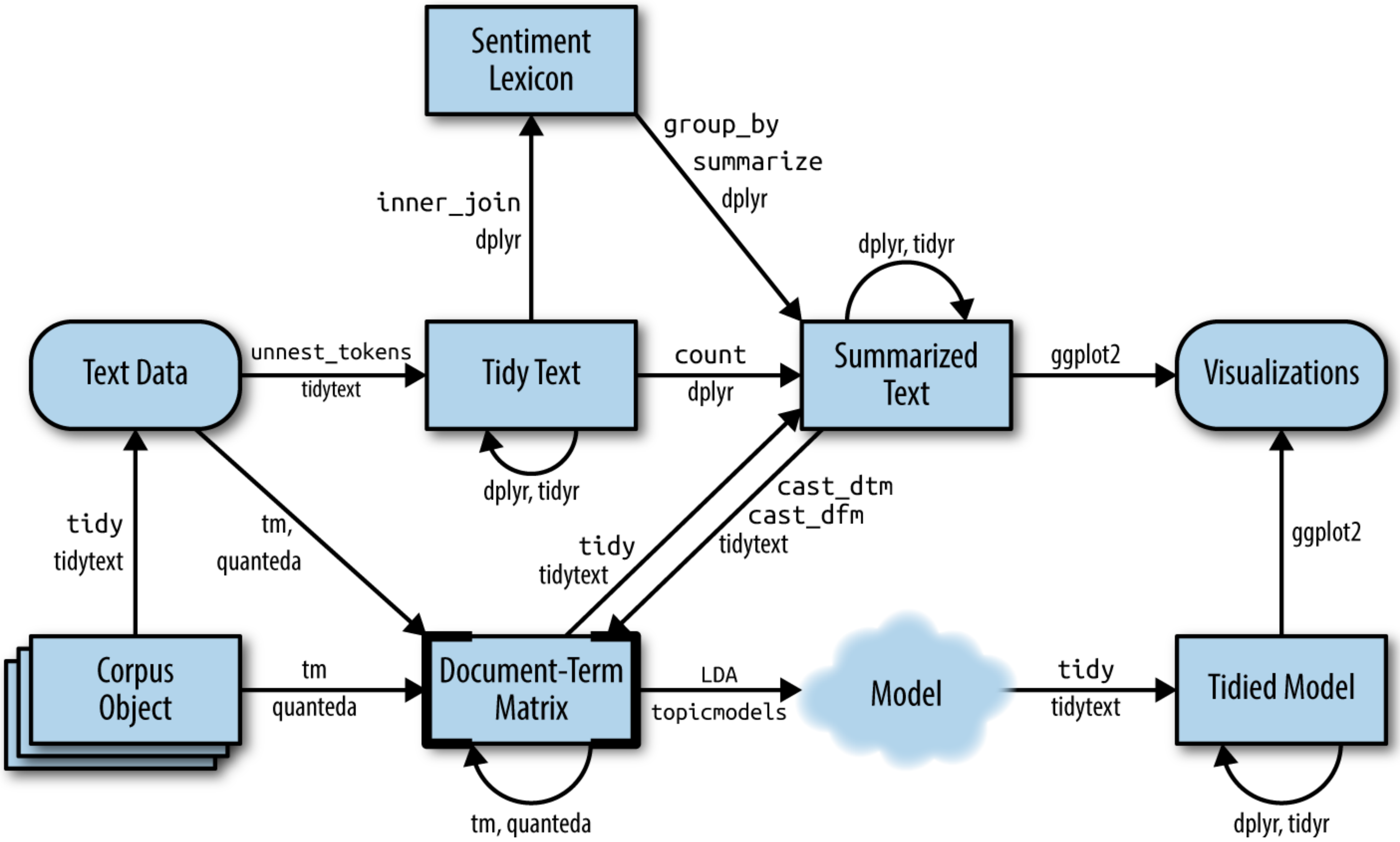
regains wiggles talks beckons gulps maneuvers zooms stalks seizes vanishes points thrusts hauls leaps hit adjusts heard shoots refers deals honks rams releases clears nears pries

TAKING TIDY TEXT TO THE NEXT LEVEL

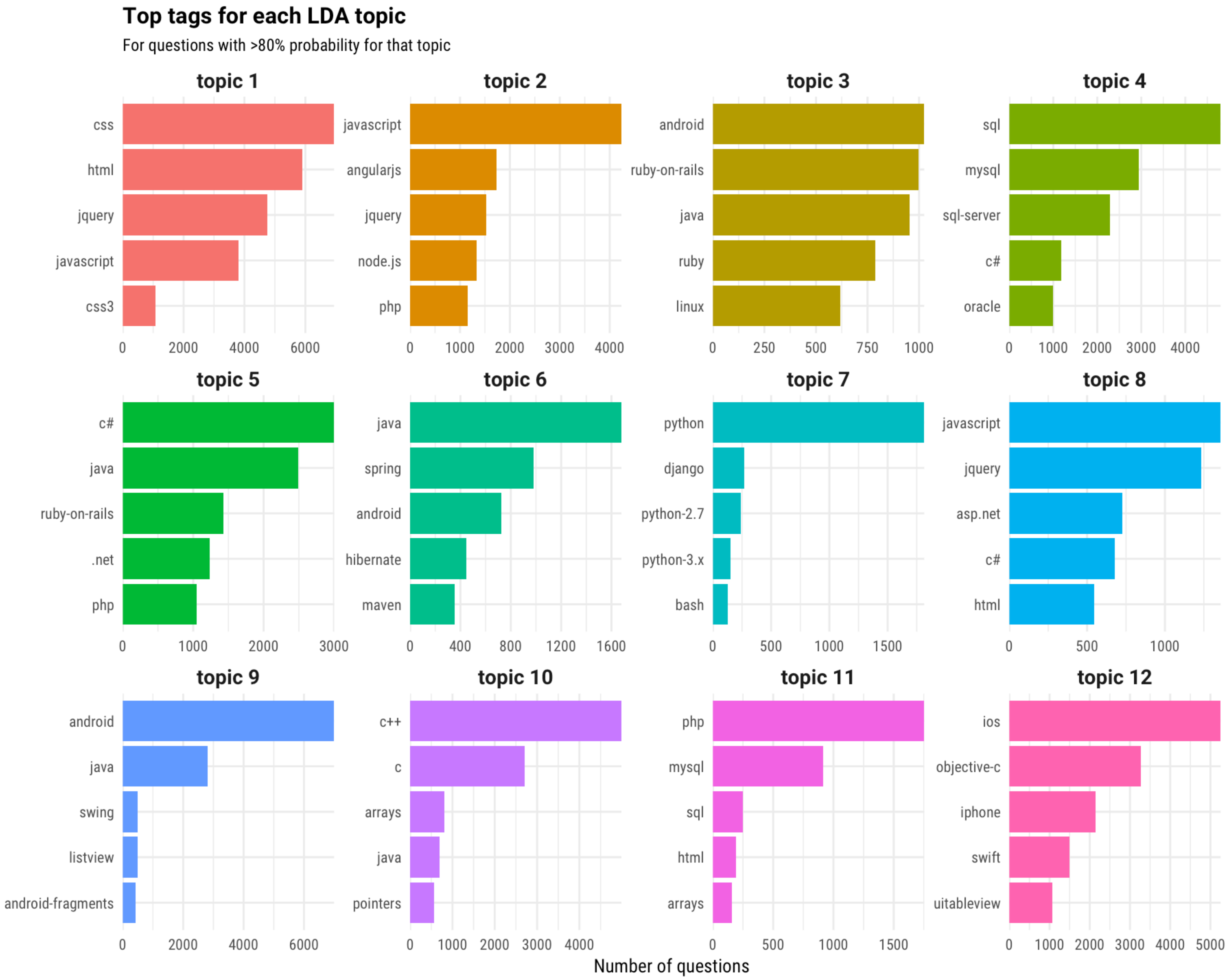
TIDYING

&

CASTING

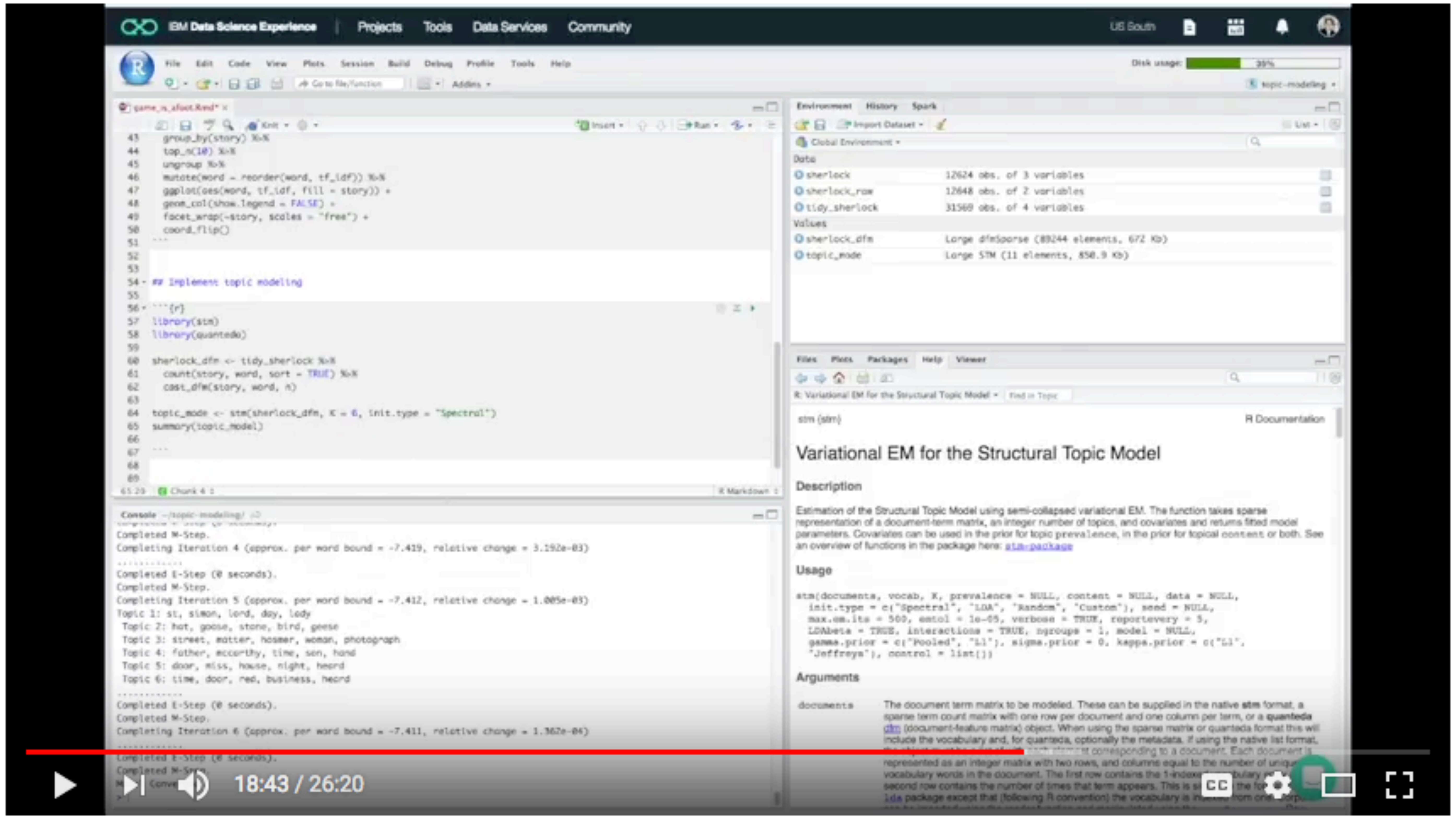


Sentiment Lexicon
group_by
summarize
dplyr
inner_join
dplyr
dplyr, tidyr
Text Data
unnest_tokens
tidytext
Tidy Text
count
dplyr
Summarized Text
ggplot2
Visualizations
dplyr, tidyr
cast_dtm
cast_dfm
tidytext
tidy
tidytext
tm,
quanteda
tidy
tidytext
ggplot2
Corpus Object
tm
quanteda
Document-Term Matrix
LDA
topicmodels
Model
tidy
tidytext
Tidied Model
tm, quanteda
dplyr, tidyr


Top tags for each LDA topic
For questions with >80% probability for that topic
topic 1
css
html
jquery
javascript
css3
0 2000 4000 6000
topic 2
javascript
angularjs
jquery
node.js
php
0 1000 2000 3000 4000
topic 3
android
ruby-on-rails
java
ruby
linux
0 250 500 750 1000
topic 4
sql
mysql
sql-server
c#
oracle
0 1000 2000 3000 4000
topic 5
c#
java
ruby-on-rails
.net
php
0 1000 2000 3000
topic 6
java
spring
android
hibernate
maven
0 400 800 1200 1600
topic 7
python
django
python-2.7
python-3.x
bash
0 500 1000 1500
topic 8
javascript
jquery
asp.net
c#
html
0 500 1000
topic 9
android
java
swing
listview
android-fragments
0 2000 4000 6000
topic 10
c++
c
arrays
java
pointers
0 1000 2000 3000 4000
topic 11
php
mysql
sql
html
arrays
0 500 1000 1500
topic 12
ios
objective-c
iphone
swift
uitableview
0 1000 2000 3000 4000 5000
Number of questions

YouTube

Search



Topic modeling with R and tidy data principles

1,372 views

42 0 SHARE

Julia Silge

Published on Dec 18, 2017

SUBSCRIBE 59

Watch along as I demonstrate how to train a topic model in R using the tidytext and stm packages on a collection of Sherlock Holmes stories. In this video, I'm working in IBM Cloud's Data Science Experience environment.

SHOW MORE

TAKING TIDY TEXT TO THE NEXT LEVEL

FINDING WORD VECTORS

TIDYTEXT

WORD VECTORS

```
> tidy_pmi <- hacker_news_text %>%
    unnest_tokens(word, text) %>%
    add_count(word) %>%
    filter(n >= 20) %>%
    select(-n) %>%
    slide_windows(quo(postID), 8) %>%
    pairwise_pmi(word, window_id)

> tidy_word_vectors <- tidy_pmi %>%
    widely_svd(item1, item2, pmi, nv = 256, maxit = 1000)
```

TIDYTEXT

WORD VECTORS

```
> tidy_word_vectors %>%
    nearest_synonyms("python")

## # A tibble: 27,267 x 2
##    item1       value
##    <chr>       <dbl>
##  1 python     0.0533
##  2 ruby       0.0309
##  3 java       0.0250
##  4 php        0.0241
##  5 c          0.0229
##  6 perl       0.0222
##  7 javascript 0.0203
##  8 django     0.0202
##  9 libraries  0.0184
## 10 languages  0.0180
## # ... with 27,257 more rows
```

TIDYTEXT

WORD VECTORS

```
> tidy_word_vectors %>%
    nearest_synonyms("bitcoin")

## # A tibble: 27,267 x 2
##    item1          value
##    <chr>          <dbl>
##  1 bitcoin       0.0626
##  2 currency      0.0328
##  3 btc           0.0320
##  4 coins         0.0300
##  5 blockchain    0.0285
##  6 bitcoins      0.0258
##  7 mining        0.0252
##  8 transactions 0.0241
##  9 transaction  0.0235
## 10 currencies   0.0228
## # ... with 27,257 more rows
```

TIDYTEXT

WORD VECTORS

```
> tidy_word_vectors %>%
    nearest_synonyms("women")

## # A tibble: 27,267 x 2
##    item1   value
##    <chr>   <dbl>
##  1 women  0.0648
##  2 men    0.0508
##  3 male   0.0345
##  4 female 0.0319
##  5 gender 0.0274
##  6 sex    0.0256
##  7 woman  0.0241
##  8 sexual 0.0226
##  9 males  0.0197
## 10 girls  0.0195
## # ... with 27,257 more rows
```

TIDYTEXT

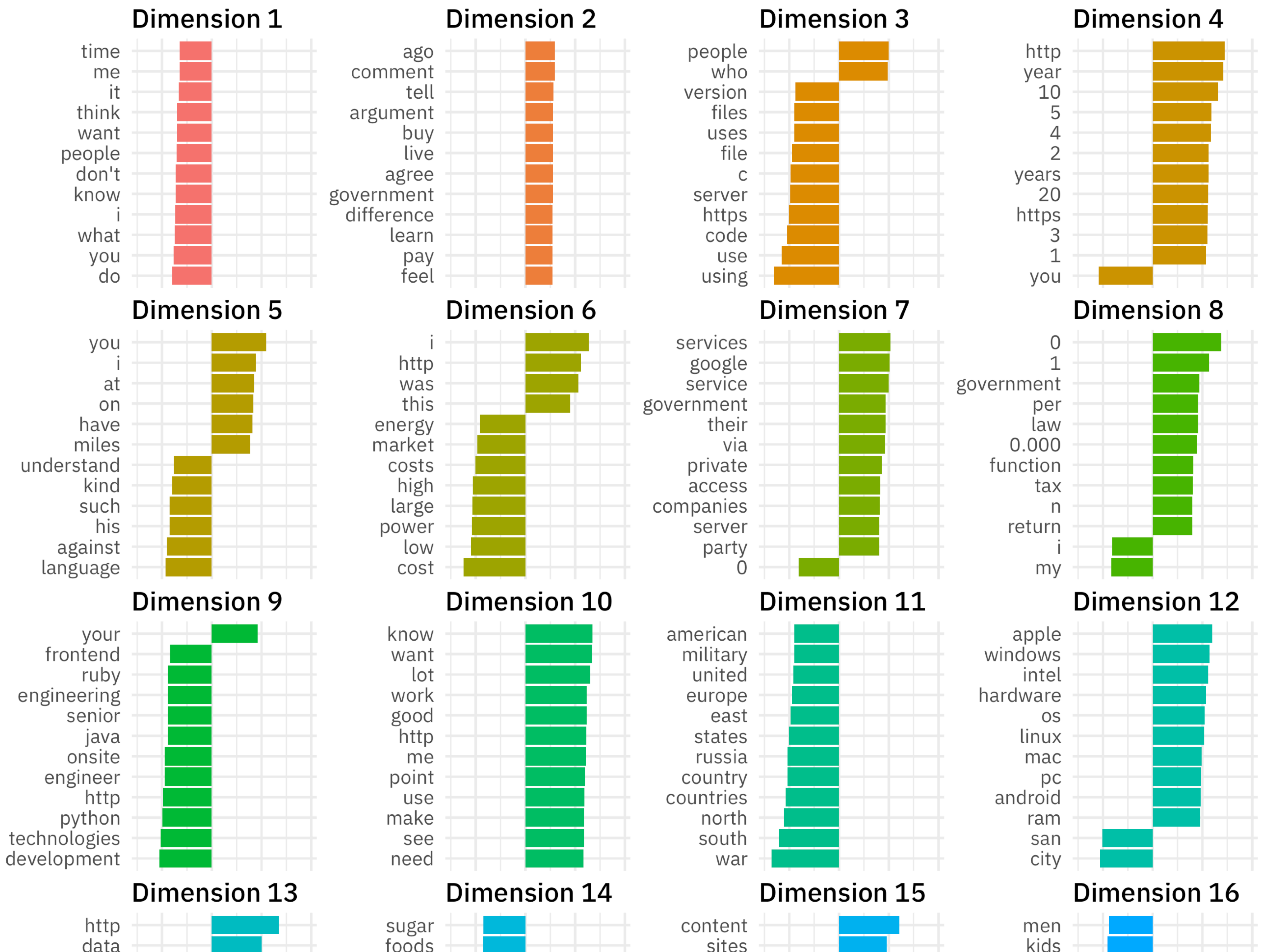
WORD VECTORS

```
> tidy_word_vectors %>%
    analogy("osx", "apple", "microsoft")

## # A tibble: 27,267 x 2
##    item1      value
##    <chr>      <dbl>
##  1 windows   0.0357
##  2 microsoft 0.0281
##  3 ms        0.0245
##  4 visual    0.0195
##  5 linux     0.0188
##  6 studio    0.0178
##  7 net       0.0171
##  8 desktop   0.0164
##  9 xp        0.0163
## 10 office    0.0147
## # ... with 27,257 more rows
```

First 24 principal components of the Hacker News corpus

Top words contributing to the components that explain the most variation



TIDYTEXT

THANK YOU

JULIA SILGE

@juliasilge

https://juliasilge.com

TIDYTEXT

THANK YOU

JULIA SILGE

@juliasilge

https://juliasilge.com

Author portraits from Wikimedia

Photos by Glen Noble and Kimberly Farmer on Unsplash