Text Modeling

USING TIDY PRINCIPLES

Julia Silge | SDSS | 29 May 2019

Let's install some packages

```
install.packages(c("tidyverse",
                   "tidytext",
                   "gutenbergr",
                   "stm",
                   "glmnet",
                   "yardstick"))
```

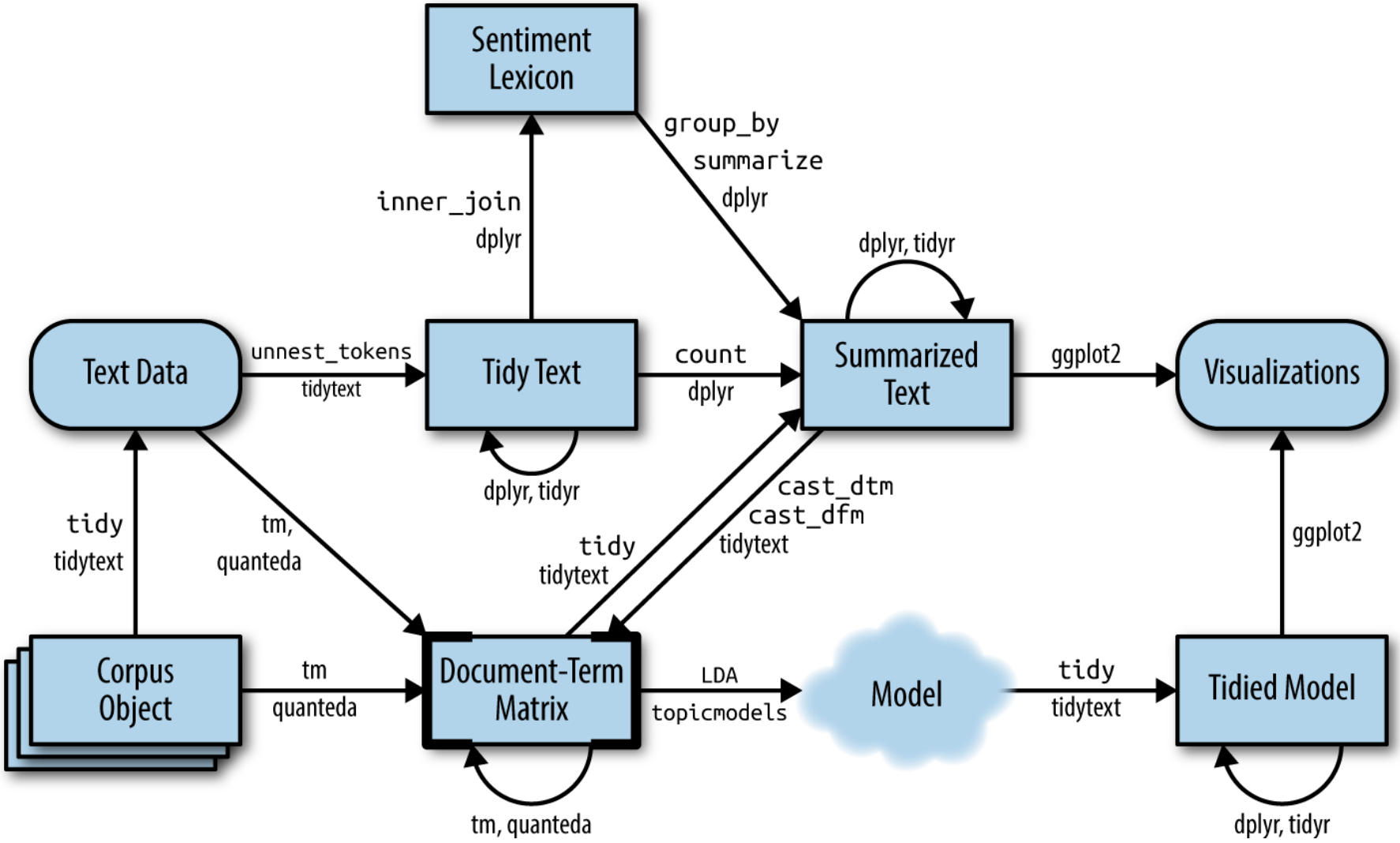
Find us at...

@juliasilge
@juliasilge
juliasilge.com

Find us at...

@dataandme
@batpigandme
maraaverick.rbind.io

TIDYING AND CASTING

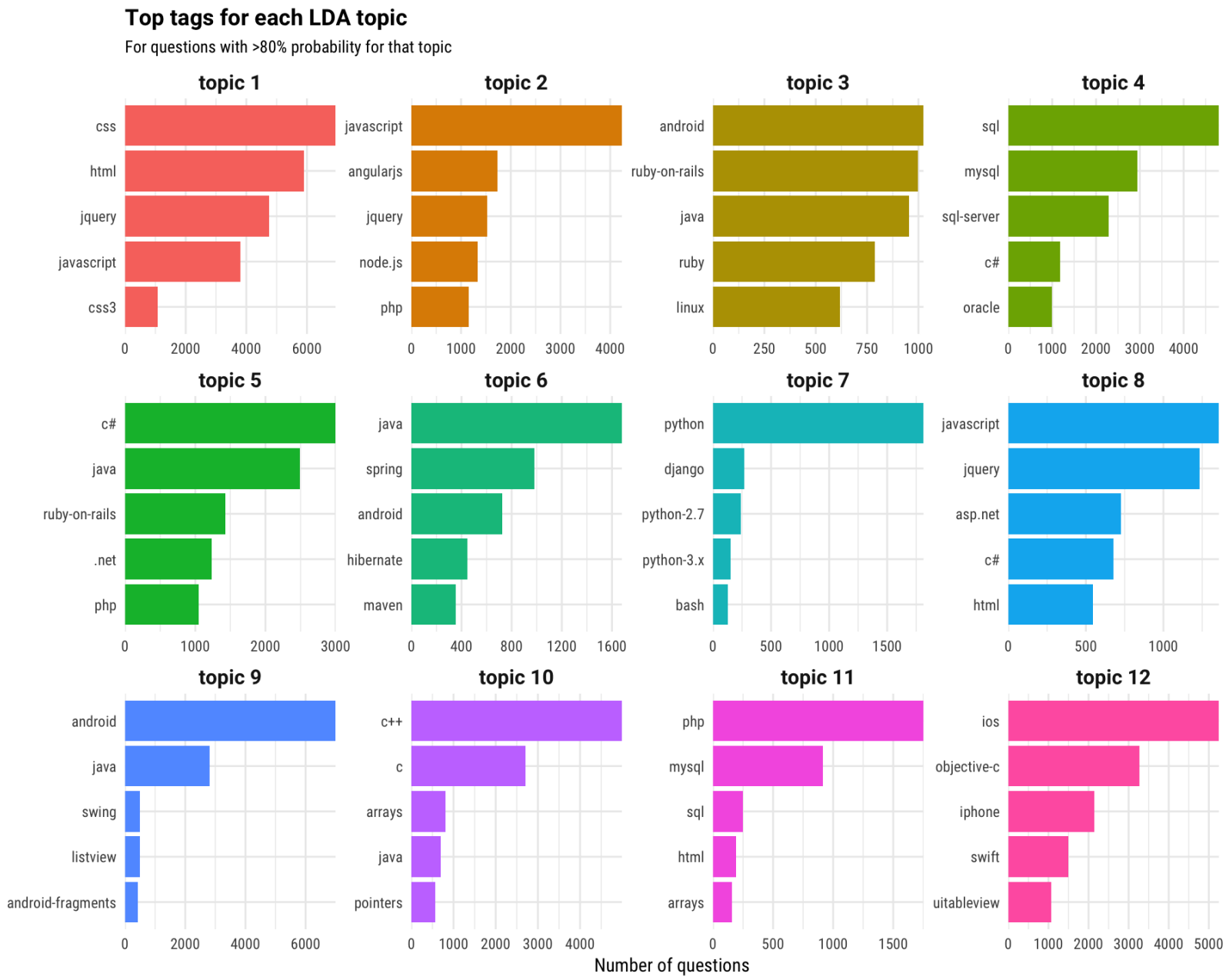

Sentiment Lexicon
group_by
summarize
dplyr
inner_join
dplyr
dplyr, tidyr
Text Data
unnest_tokens
tidytext
Tidy Text
count
dplyr
Summarized Text
ggplot2
Visualizations
dplyr, tidyr
tidy
tidytext
tm,
quanteda
tidy
tidytext
cast_dtm
cast_dfm
tidytext
ggplot2
Corpus Object
tm
quanteda
Document-Term Matrix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

Two powerful NLP techniques

- Topic modeling
- Text classification

Topic modeling

- Each DOCUMENT = mixture of topics
- Each TOPIC = mixture of words


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

GREAT LIBRARY HEIST 🕵️‍♀️

Downloading your text data

```
library(tidyverse)
library(gutenbergr)

titles <- c("Twenty Thousand Leagues under the Sea",
            "The War of the Worlds",
            "Pride and Prejudice",
            "Great Expectations")

books <- gutenberg_works(title %in% titles) %>%
  gutenberg_download(meta_fields = "title")

books
```

```
## # A tibble: 51,663 x 3
##    gutenberg_id text                                                title
##           <int> <chr>                                               <chr>
##  1           36 The War of the Worlds                               The War of the…
##  2           36 ""                                                  The War of the…
##  3           36 by H. G. Wells [1898]                               The War of the…
##  4           36 ""                                                  The War of the…
##  5           36 ""                                                  The War of the…
##  6           36 "     But who shall dwell in these worlds …         The War of the…
##  7           36 "     inhabited? .  .  .  Are we or they L…         The War of the…
##  8           36 "     World? .  .  .  And how are all thin…         The War of the…
##  9           36 "          KEPLER (quoted in The Anatomy o…         The War of the…
## 10           36 ""                                                  The War of the…
## # … with 51,653 more rows
```

Someone has torn your books apart! 😭

```
by_chapter <- books %>%
  group_by(title) %>%
  mutate(chapter = cumsum(str_detect(text,
                                     regex("^chapter ",
                                           ignore_case = TRUE)))) %>%
  ungroup() %>%
  filter(chapter > 0) %>%
  unite(document, title, chapter)

by_chapter
```

```
## # A tibble: 51,602 x 3
##    gutenberg_id text                                                document
##           <int> <chr>                                               <chr>
##  1           36 CHAPTER ONE                                         The War of the …
##  2           36 ""                                                  The War of the …
##  3           36 THE EVE OF THE WAR                                  The War of the …
##  4           36 ""                                                  The War of the …
##  5           36 ""                                                  The War of the …
##  6           36 No one would have believed in the last ye…          The War of the …
##  7           36 century that this world was being watched…          The War of the …
##  8           36 intelligences greater than man's and yet …          The War of the …
##  9           36 men busied themselves about their various…          The War of the …
## 10           36 scrutinised and studied, perhaps almost a…          The War of the …
## # … with 51,592 more rows
```

Can we put them back together?

```
library(tidytext)

word_counts <- by_chapter %>%
  unnest_tokens(word, text) %>%
  anti_join(get_stopwords(source = "smart")) %>%
  count(document, word, sort = TRUE)

word_counts
```

```
## # A tibble: 111,650 x 3
##    document               word        n
##    <chr>                  <chr>   <int>
##  1 Great Expectations_57  joe        88
##  2 Great Expectations_7   joe        70
##  3 Pride and Prejudice_18 mr         66
##  4 Great Expectations_17  biddy      63
##  5 Great Expectations_27  joe        58
##  6 Great Expectations_38  estella    58
##  7 Great Expectations_2   joe        56
##  8 Great Expectations_23  pocket     53
##  9 Great Expectations_15  joe        50
## 10 Great Expectations_18  joe        50
## # … with 111,640 more rows
```

Can we put them back together?

```
words_sparse <- word_counts %>%
  cast_sparse(document, word, n)

class(words_sparse)
```

```
## [1] "dgCMatrix"
## attr(,"package")
## [1] "Matrix"
```

Train a topic model

Use a sparse matrix or a `quanteda::dfm` object as input

```
library(stm)

topic_model <- stm(words_sparse, K = 4,
                   verbose = FALSE, init.type = "Spectral")

summary(topic_model)
```

```
## A topic model with 4 topics, 193 documents and a 18360 word dictionary.

## Topic 1 Top Words:
##       Highest Prob: mr, elizabeth, mrs, darcy, bennet, miss, jane
##       FREX: elizabeth, darcy, bennet, bingley, wickham, collins, lydia
##       Lift: wickham, nephew, phillips, brighton, meryton, bourgh, mend
##       Score: elizabeth, darcy, bennet, bingley, wickham, jane, lydia
## Topic 2 Top Words:
##       Highest Prob: captain, nautilus, sea, nemo, ned, conseil, land
##       FREX: nautilus, nemo, ned, conseil, canadian, ocean, seas
##       Lift: vanikoro, indian, d'urville, reefs, scotia, shark's, solidification
##       Score: nautilus, nemo, ned, conseil, canadian, ocean, captain
## Topic 3 Top Words:
##       Highest Prob: mr, joe, miss, time, pip, looked, herbert
##       FREX: joe, pip, herbert, wemmick, havisham, estella, biddy
##       Lift: towel, giv, whimple, meantersay, jew, rot, barnard's
##       Score: joe, wemmick, pip, jaggers, havisham, estella, herbert
## Topic 4 Top Words:
##       Highest Prob: people, martians, man, time, black, men, night
##       FREX: martians, martian, woking, mars, curate, pine, ulla
##       Lift: martians, mars, curate, shepperton, henderson, hood, ripley
##       Score: martians, martian, woking, cylinder, curate, ulla, pine
```

Exploring the output of topic modeling

Time for tidying!

```
chapter_topics <- tidy(topic_model, matrix = "beta")

chapter_topics
```

```
## # A tibble: 73,440 x 3
##    topic term        beta
##    <int> <chr>      <dbl>
##  1     1 joe   8.69e-104
##  2     2 joe   3.03e-139
##  3     3 joe   1.21e-  2
##  4     4 joe   3.28e- 19
##  5     1 mr    1.90e-  2
##  6     2 mr    1.91e-  4
##  7     3 mr    1.22e-  2
##  8     4 mr    1.15e- 45
##  9     1 biddy 3.21e- 80
## 10     2 biddy 3.84e-149
## # … with 73,430 more rows
```

Exploring the output of topic modeling

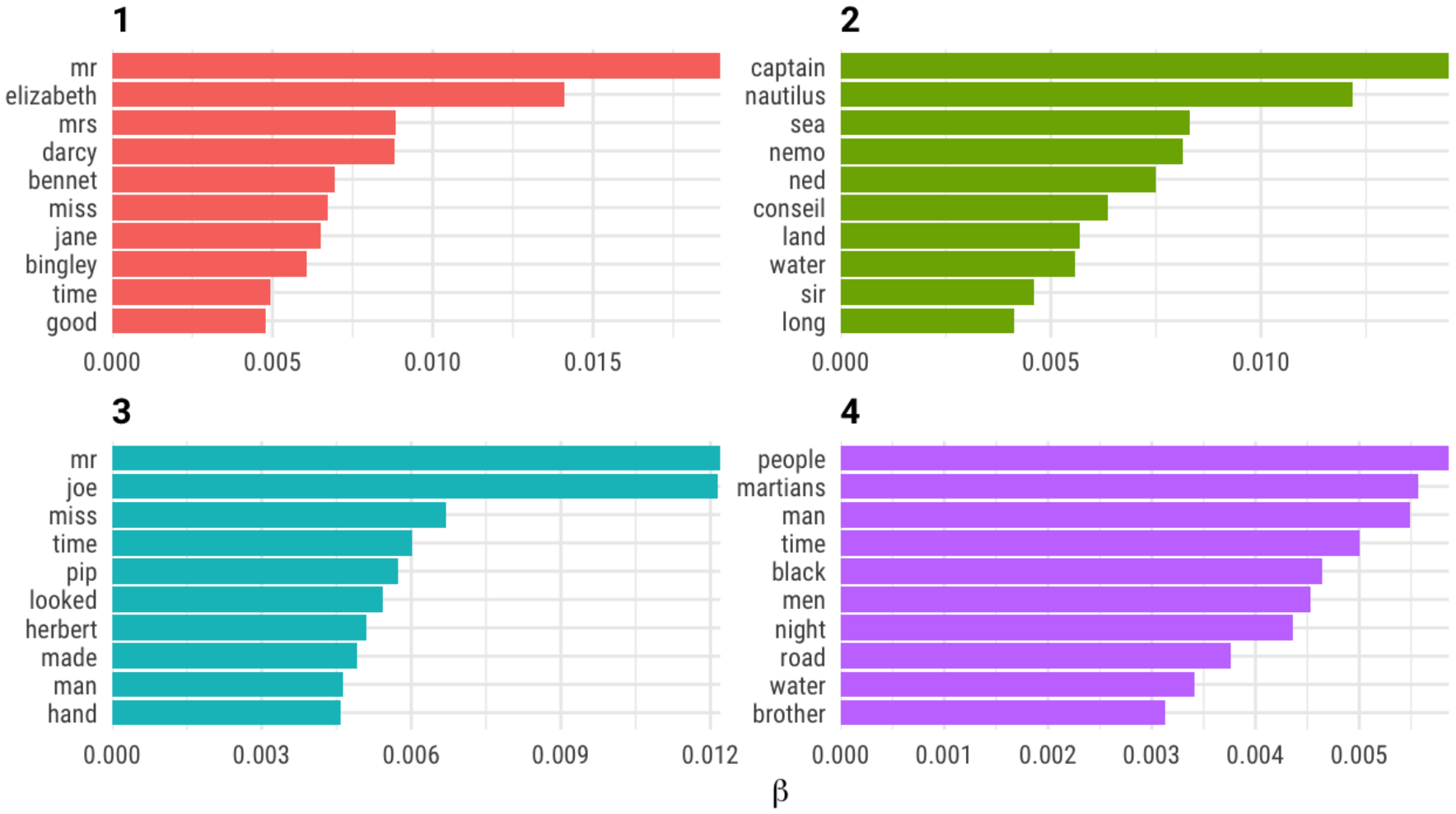
```
top_terms <- chapter_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms
```

```
## # A tibble: 40 x 3
##    topic term         beta
##    <int> <chr>       <dbl>
##  1     1 mr         0.0190
##  2     1 elizabeth  0.0141
##  3     1 mrs        0.00886
##  4     1 darcy      0.00881
##  5     1 bennet     0.00694
##  6     1 miss       0.00674
##  7     1 jane       0.00652
##  8     1 bingley    0.00607
##  9     1 time       0.00493
## 10     1 good       0.00480
## # … with 30 more rows
```

Exploring the output of topic modeling

```
top_terms %>%
  mutate(term = fct_reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```


1
mr
elizabeth
mrs
darcy
bennet
miss
jane
bingley
time
good
0.000
0.005
0.010
0.015
2
captain
nautilus
sea
nemo
ned
conseil
land
water
sir
long
0.000
0.005
0.010
3
mr
joe
miss
time
pip
looked
herbert
made
man
hand
0.000
0.003
0.006
0.009
0.012
4
people
martians
man
time
black
men
night
road
water
brother
0.000
0.001
0.002
0.003
0.004
0.005
β

How are documents classified?

```
chapters_gamma <- tidy(topic_model, matrix = "gamma",
                       document_names = rownames(words_sparse))

chapters_gamma
```

```
## # A tibble: 772 x 3
##    document               topic    gamma
##    <chr>                  <int>    <dbl>
##  1 Great Expectations_57      1 0.000792
##  2 Great Expectations_7       1 0.00340
##  3 Pride and Prejudice_18     1 1.000
##  4 Great Expectations_17      1 0.0480
##  5 Great Expectations_27      1 0.000367
##  6 Great Expectations_38      1 0.00110
##  7 Great Expectations_2       1 0.000531
##  8 Great Expectations_23      1 0.432
##  9 Great Expectations_15      1 0.000565
## 10 Great Expectations_18      1 0.000277
## # … with 762 more rows
```

How are documents classified?

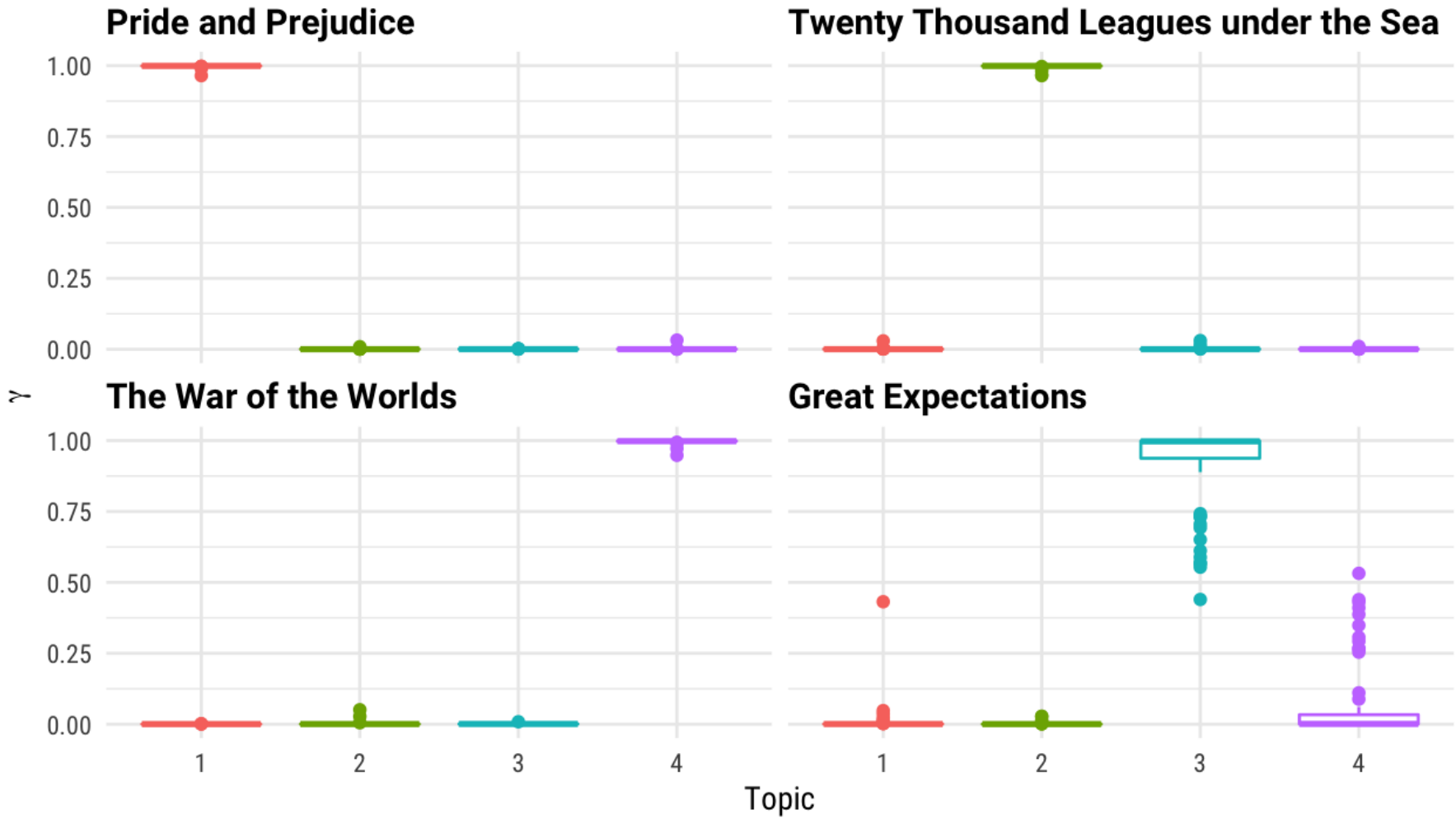
```
chapters_parsed <- chapters_gamma %>%
  separate(document, c("title", "chapter"),
           sep = "_", convert = TRUE)

chapters_parsed
```

```
## # A tibble: 772 x 4
##    title                chapter topic    gamma
##    <chr>                  <int> <int>    <dbl>
##  1 Great Expectations        57     1 0.000792
##  2 Great Expectations         7     1 0.00340
##  3 Pride and Prejudice       18     1 1.000
##  4 Great Expectations        17     1 0.0480
##  5 Great Expectations        27     1 0.000367
##  6 Great Expectations        38     1 0.00110
##  7 Great Expectations         2     1 0.000531
##  8 Great Expectations        23     1 0.432
##  9 Great Expectations        15     1 0.000565
## 10 Great Expectations        18     1 0.000277
## # … with 762 more rows
```

How are documents classified?

```
chapters_parsed %>%
  mutate(title = fct_reorder(title, gamma * topic)) %>%
  ggplot(aes(factor(topic), gamma)) +
  geom_boxplot() +
  facet_wrap(~ title)
```


Pride and Prejudice
Twenty Thousand Leagues under the Sea
The War of the Worlds
Great Expectations
γ
1.00
0.75
0.50
0.25
0.00
1
2
3
4
Topic

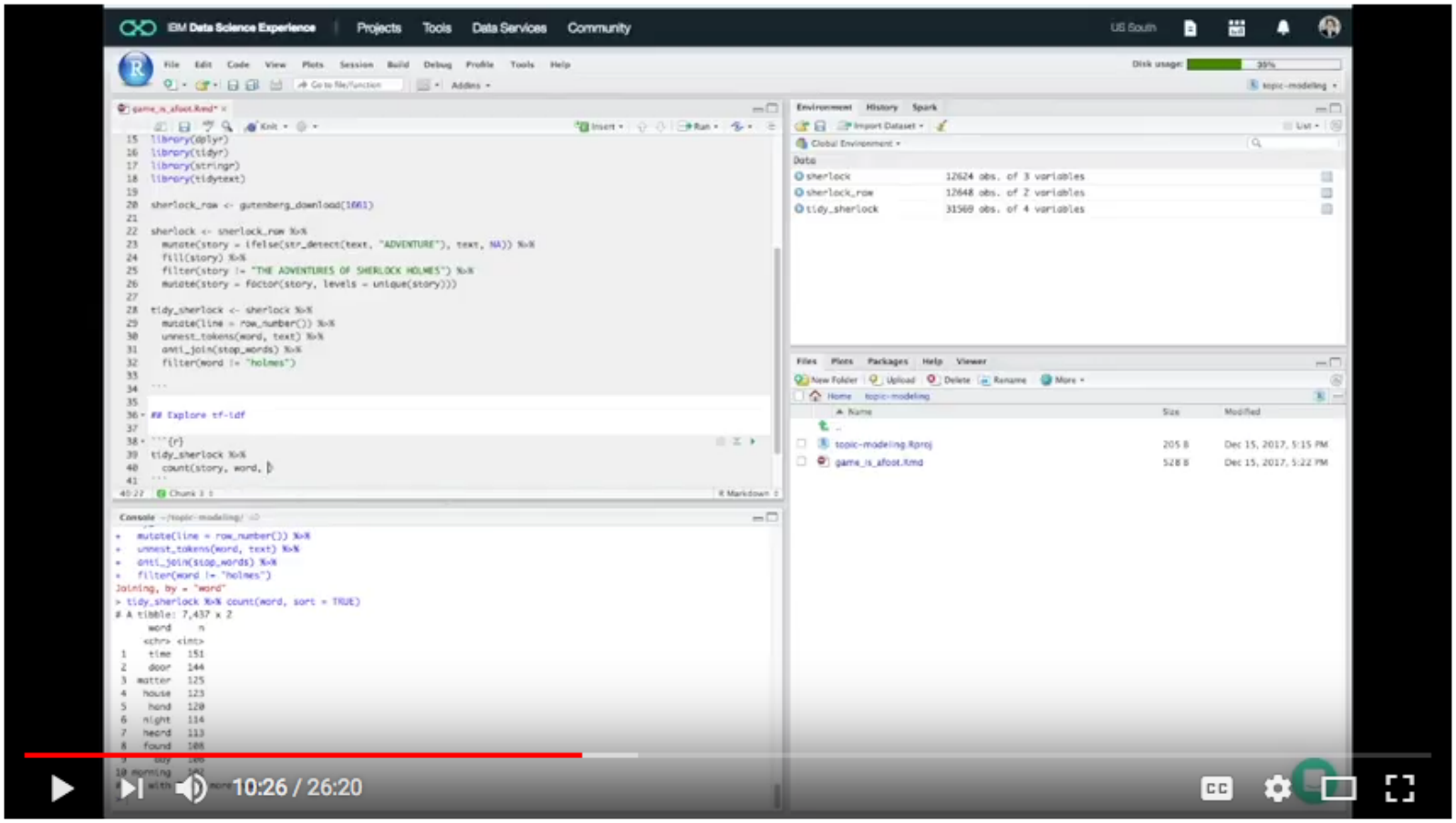
GOING FARTHER 🚀

Tidying model output

Which words in each document are assigned to which topics?

- `augment()`
- Add information to each observation in the original data

YouTube Search



Topic modeling with R and tidy data principles

1,971 views

51 0 SHARE

Julia Silge
Published on Dec 18, 2017

SUBSCRIBE 74

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

Using stm

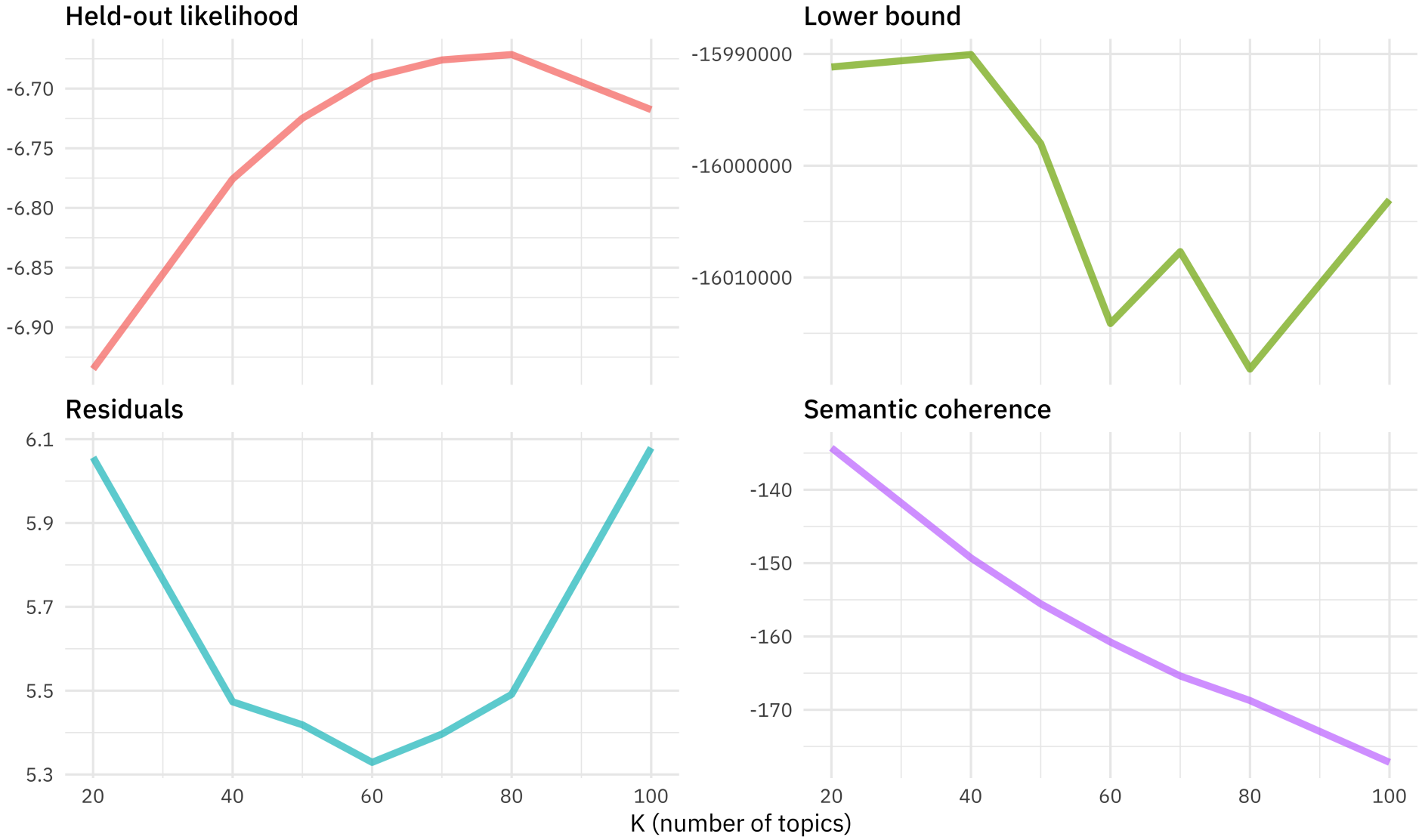
- Document-level covariates

```
topic_model <- stm(words_sparse, K = 0, init.type = "Spectral",
                   prevalence = ~s(Year),
                   data = covariates,
                   verbose = FALSE)
```

- Use functions for `semanticCoherence()`, `checkResiduals()`, `exclusivity()`, and more!
- Check out http://www.structuraltopicmodel.com/
- See my blog post for how to choose `K`, the number of topics

Model diagnostics by number of topics

These diagnostics indicate that a good number of topics would be around 60



Stemming?

Advice from Schofield & Mimno

"Comparing Apples to Apple: The Effects of Stemmers on Topic Models"

> **Despite their frequent use in topic modeling, we find that stemmers produce no meaningful improvement in likelihood and coherence and in fact can degrade topic stability.**

TEXT CLASSIFICATION

Downloading your text data

```
library(tidyverse)
library(gutenbergr)

titles <- c("The War of the Worlds",
            "Pride and Prejudice")

books <- gutenberg_works(title %in% titles) %>%
  gutenberg_download(meta_fields = "title") %>%
  mutate(document = row_number())

books
```

```
## # A tibble: 19,504 x 4
##    gutenberg_id text                                    title          document
##           <int> <chr>                                   <chr>             <int>
##  1           36 The War of the Worlds                   The War of t…         1
##  2           36 ""                                      The War of t…         2
##  3           36 by H. G. Wells [1898]                   The War of t…         3
##  4           36 ""                                      The War of t…         4
##  5           36 ""                                      The War of t…         5
##  6           36 "     But who shall dwell in these …    The War of t…         6
##  7           36 "     inhabited? .  .  .  Are we or…    The War of t…         7
##  8           36 "     World? .  .  .  And how are a…    The War of t…         8
##  9           36 "          KEPLER (quoted in The An…    The War of t…         9
## 10           36 ""                                      The War of t…        10
## # … with 19,494 more rows
```

Making a tidy dataset

Use this kind of data structure for EDA! 💅

```
library(tidytext)

tidy_books <- books %>%
  unnest_tokens(word, text) %>%
  group_by(word) %>%
  filter(n() > 10) %>%
  ungroup

tidy_books
```

```
## # A tibble: 159,707 x 4
##    gutenberg_id title                 document word
##           <int> <chr>                    <int> <chr>
##  1           36 The War of the Worlds        1 the
##  2           36 The War of the Worlds        1 war
##  3           36 The War of the Worlds        1 of
##  4           36 The War of the Worlds        1 the
##  5           36 The War of the Worlds        3 by
##  6           36 The War of the Worlds        6 but
##  7           36 The War of the Worlds        6 who
##  8           36 The War of the Worlds        6 shall
##  9           36 The War of the Worlds        6 in
## 10           36 The War of the Worlds        6 these
## # … with 159,697 more rows
```

Cast to a sparse matrix

And build a dataframe with a response variable

```
sparse_words <- tidy_books %>%
  count(document, word, sort = TRUE) %>%
  cast_sparse(document, word, n)

books_joined <- tibble(document = as.integer(rownames(sparse_words))) %>%
  left_join(books %>%
              select(document, title))
```

Train a glmnet model

```
library(glmnet)
library(doMC)
registerDoMC(cores = 8)

is_jane <- books_joined$title == "Pride and Prejudice"

model <- cv.glmnet(sparse_words, is_jane, family = "binomial",
                   parallel = TRUE, keep = TRUE)
```

Tidying our model

Tidy, then filter to choose some lambda from glmnet output

```
library(broom)

coefs <- model$glmnet.fit %>%
  tidy() %>%
  filter(lambda == model$lambda.1se)

Intercept <- coefs %>%
  filter(term == "(Intercept)") %>%
  pull(estimate)
```

Tidying our model

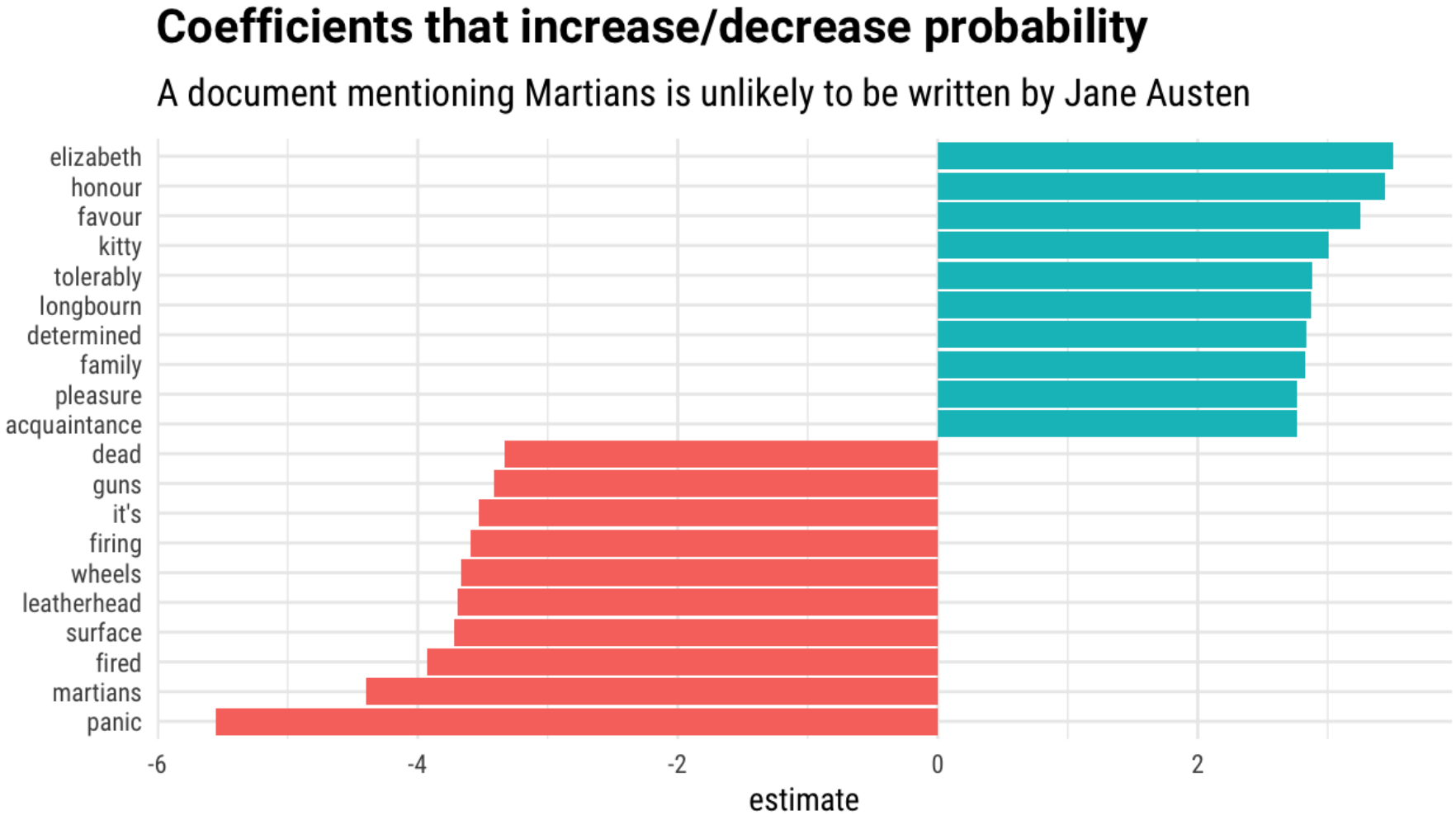
```
classifications <- tidy_books %>%
  inner_join(coefs, by = c("word" = "term")) %>%
  group_by(document) %>%
  summarize(score = sum(estimate)) %>%
  mutate(probability = plogis(Intercept + score))

classifications
```

```
## # A tibble: 16,001 x 3
##    document  score probability
##       <int>  <dbl>       <dbl>
##  1        1 -2.34      0.110
##  2        3  0.205     0.611
##  3        6  1.85      0.890
##  4        7 -1.02      0.315
##  5        8 -1.25      0.268
##  6        9 -0.526     0.430
##  7       13 -0.238     0.502
##  8       15 -5.47      0.00533
##  9       19  0.373     0.650
## 10       21 -2.34      0.110
## # … with 15,991 more rows
```

Understanding our model

```
coefs %>%
  group_by(estimate > 0) %>%
  top_n(10, abs(estimate)) %>%
  ungroup %>%
  ggplot(aes(fct_reorder(term, estimate),
             estimate,
             fill = estimate > 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip()
```


Coefficients that increase/decrease probability
A document mentioning Martians is unlikely to be written by Jane Austen
elizabeth
honour
favour
kitty
tolerably
longbourn
determined
family
pleasure
acquaintance
dead
guns
it's
firing
wheels
leatherhead
surface
fired
martians
panic
-6
-4
-2
0
2
estimate

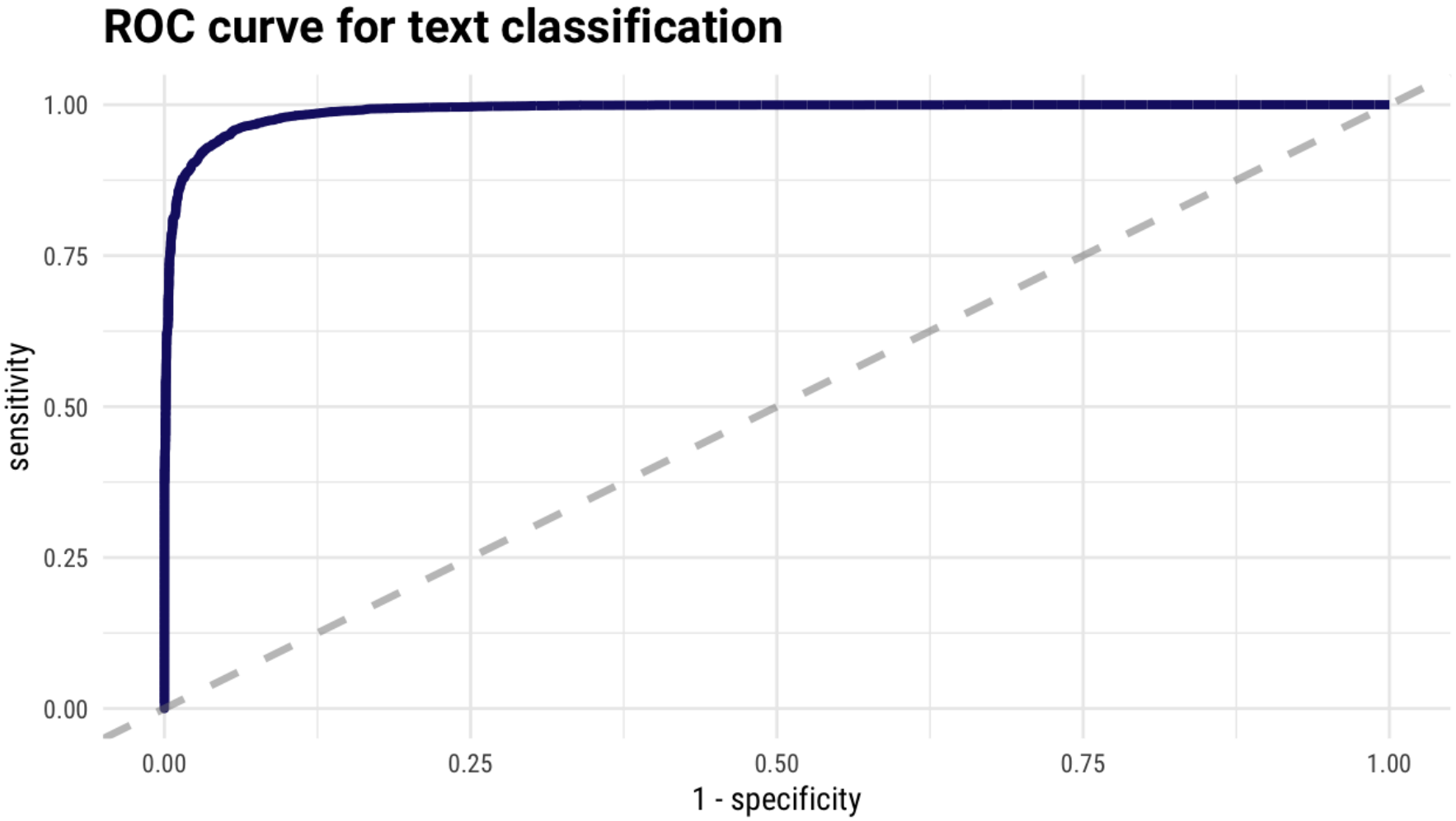
ROC

```
library(yardstick)

comment_classes <- classifications %>%
  left_join(books %>%
    select(title, document), by = "document") %>%
  mutate(title = as.factor(title))
```

ROC

```
comment_classes %>%
  roc_curve(title, probability) %>%
  ggplot(aes(x = 1 - specificity, y = sensitivity)) +
  geom_line(
    color = "midnightblue",
    size = 1.5
  ) +
  geom_abline(
    lty = 2, alpha = 0.5,
    color = "gray50",
    size = 1.2
  )
```


ROC curve for text classification
1.00
0.75
0.50
0.25
0.00
sensitivity
0.00
0.25
0.50
0.75
1.00
1 - specificity

AUC for model

```
comment_classes %>%
  roc_auc(title, probability)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.990
```

Confusion matrix

```
comment_classes %>%
  mutate(
    prediction = case_when(
      probability > 0.5 ~ "Pride and Prejudice",
      TRUE ~ "The War of the Worlds"
    ),
    prediction = as.factor(prediction)
  ) %>%
  conf_mat(title, prediction)
```

```
##                          Truth
## Prediction                Pride and Prejudice The War of the Worlds
##   Pride and Prejudice                   10351                   484
##   The War of the Worlds                   264                  4902
```

Misclassifications

Let's talk about misclassifications. Which documents here were incorrectly predicted to be written by Jane Austen?

```
comment_classes %>%
  filter(
    probability > .8,
    title == "The War of the Worlds"
  ) %>%
  sample_n(10) %>%
  inner_join(books %>%
    select(document, text)) %>%
  select(probability, text)
```

```
## # A tibble: 10 x 2
##    probability text
##          <dbl> <chr>
##  1       0.858 ladies there being by no means the least active.
##  2       0.851 is wrong as well as any, but not what is possible to tortur…
##  3       0.972 She put her hand to her throat--swayed.  I made a step forw…
##  4       0.962 "\"Be a man!\" said I.  \"You are scared out of your wits! …
##  5       0.832 "\"Take this!\" said the slender lady, and she gave my brot…
##  6       0.827 decorum were necessarily different from ours; and not only …
##  7       0.906 "\"Half a mile, you say?\" said he."
##  8       0.910 breed.  I tell you, I'm grim set on living.  And if I'm not…
##  9       0.854 would be advisable to kill him, lest his actions attracted …
## 10       0.919 winter.  Its air is much more attenuated than ours, its oce…
```

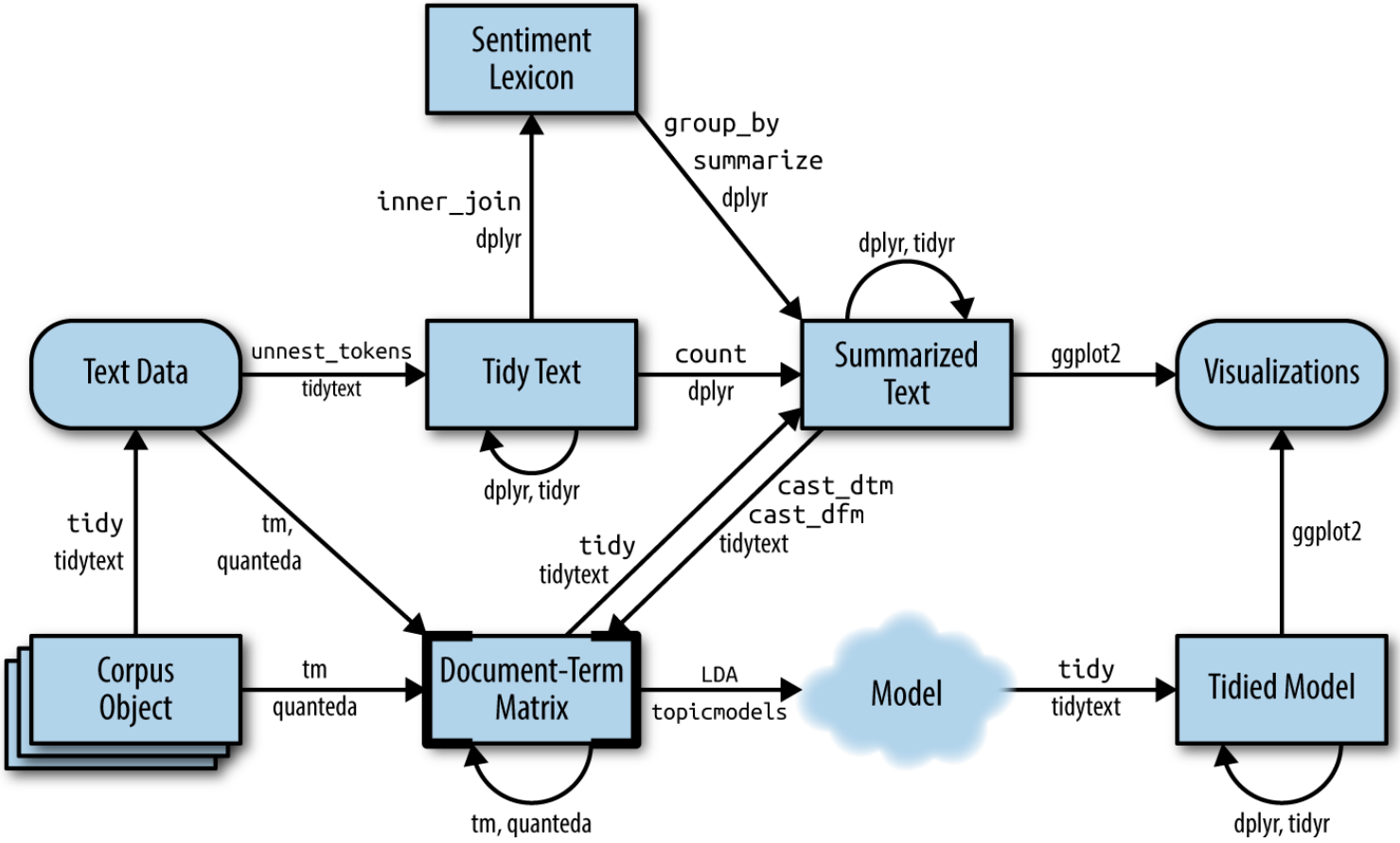
Misclassifications

Let's talk about misclassifications. Which documents here were incorrectly predicted to *not* be written by Jane Austen?

```
comment_classes %>%
  filter(
    probability < .3,
    title == "Pride and Prejudice"
  ) %>%
  sample_n(10) %>%
  inner_join(books %>%
    select(document, text)) %>%
  select(probability, text)
```

```
## # A tibble: 10 x 2
##    probability text
##          <dbl> <chr>
##  1       0.188 is so violent, that it would be the death of half the good …
##  2       0.208 blush. He absolutely started, and for a moment seemed immov…
##  3       0.269 of contradictions and varieties, sighed at the perverseness…
##  4       0.220 suddenly came forward from the road, which led behind it to…
##  5       0.199 "\"A little sea-bathing would set me up forever.\""
##  6       0.286 it had just transpired that he had left gaming debts behind…
##  7       0.266 of the gates into the ground.
##  8       0.120 the happiest of men.
##  9       0.218 They travelled as expeditiously as possible, and, sleeping …
## 10       0.279 the improvements it was receiving, he was happily employed …
```

Workflow for text mining/modeling



Go explore real-world text!

Thanks!

- tidytextmining.com
- @juliasilge
- @juliasilge
- juliasilge.com
- @dataandme
- @batpigandme
- maraaverick.rbind.io

Slides created with **remark.js** and the R package **xaringan**