LangCon 2019

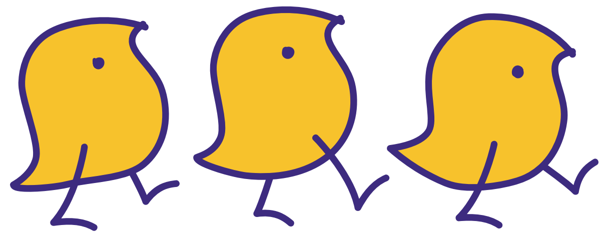
Korean Text Analysis in R

연세치대 치의학교육연구센터 김준혁

자기소개

- 치과의사
- 작가, 칼럼니스트, 번역가
- 의료윤리, 의료인문학 연구자
- + 디지털 인문학(digital humanities) 연구자? 개발자?

목차



R에서 텍스트 분석을 한다고?

Why R?

- “텍스트 분석은 당연히 Python에서 하는 거 아니야?”
- “그렇긴 한데요… R도 장점이 있긴 한데….”

- R의 장점?

R vs. Python

- R
  - Visualization with *ggplot2*
  - For statisticians
  - Slow
  - Steep learning curve
- Python
  - Sharing workflow with *iPython Notebook*
  - Multipurpose
  - For work
  - Lack of statistical packages/functions

Willems, K. (May 12, 2015). Choosing R or Python for Data Analysis? An Infographic. *DataCamp*. <https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>

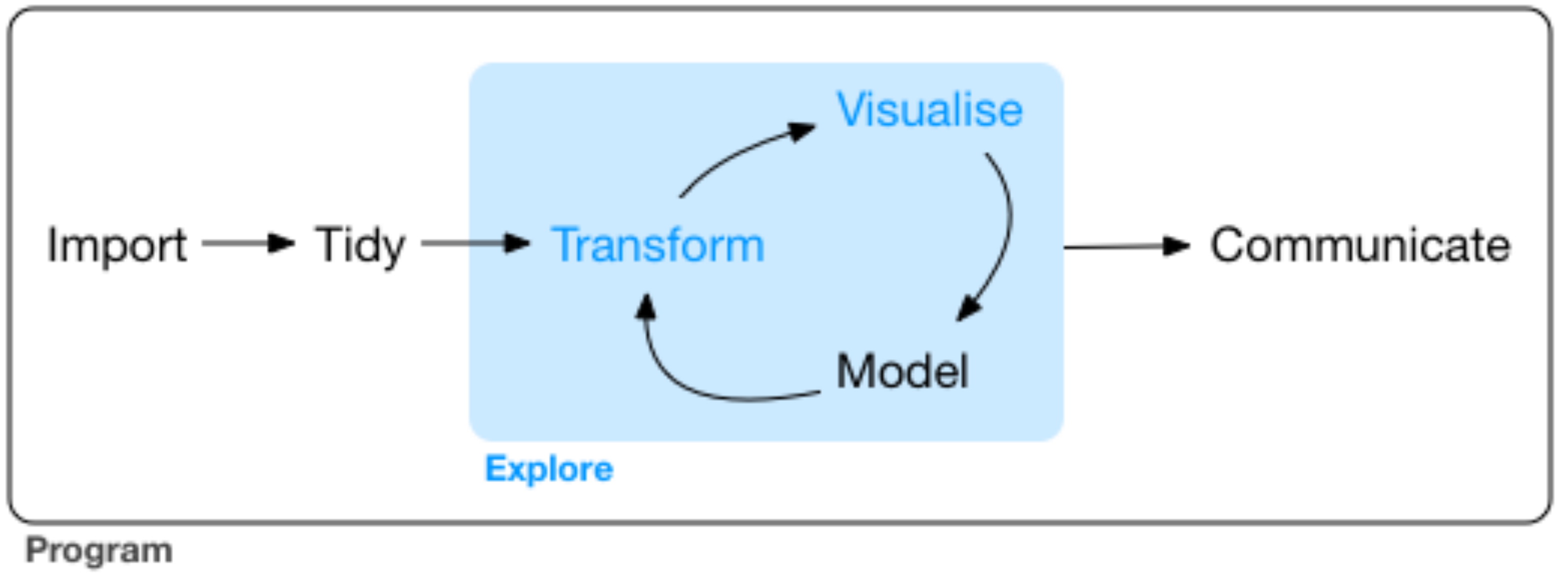
Then, still R?

- "여전히 R일 필요가 있나요?"
- "없습니다. 하지만…."

- GPL License, forcefully *copyleft*
- What's your purpose of text analysis? If you want to deploy a service, choose Python. If you want to do a research, how about using R?

Tidy Principle

- Tidy data
  - “The data obtained as a result of a process called data tidying”

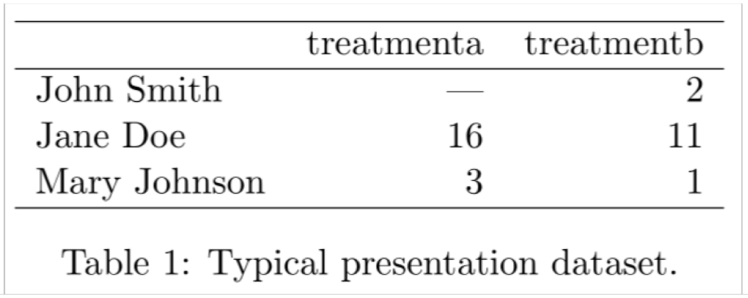


Wikipedia contributors. Tidy data. *Wikipedia, The Free Encyclopedia*. Oct 23, 2018. https://en.wikipedia.org/wiki/Tidy_data
Wickham H., Grolemund G. (2017). “2. Introduction”. *R for Data Science*. O’Reilly

Tidy Principle

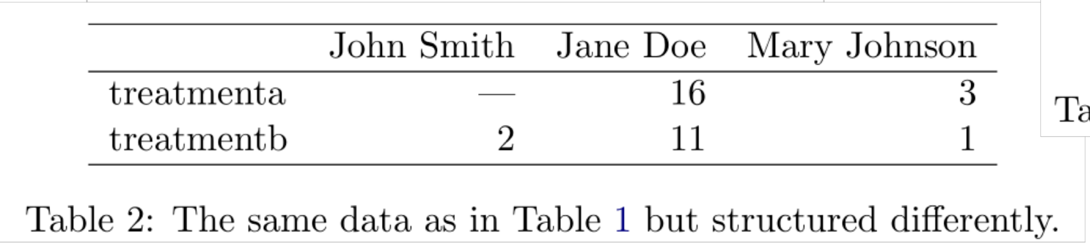
Language Conference 2019

- Characteristics of tidy data
  1. Each variable you measure should be in one column.
  2. Each different observation of that variable should be in a different row.
  3. There should be one table for each “kind” of variable.
  4. If you have multiple tables, they should include a column in the table that allows them to be linked.



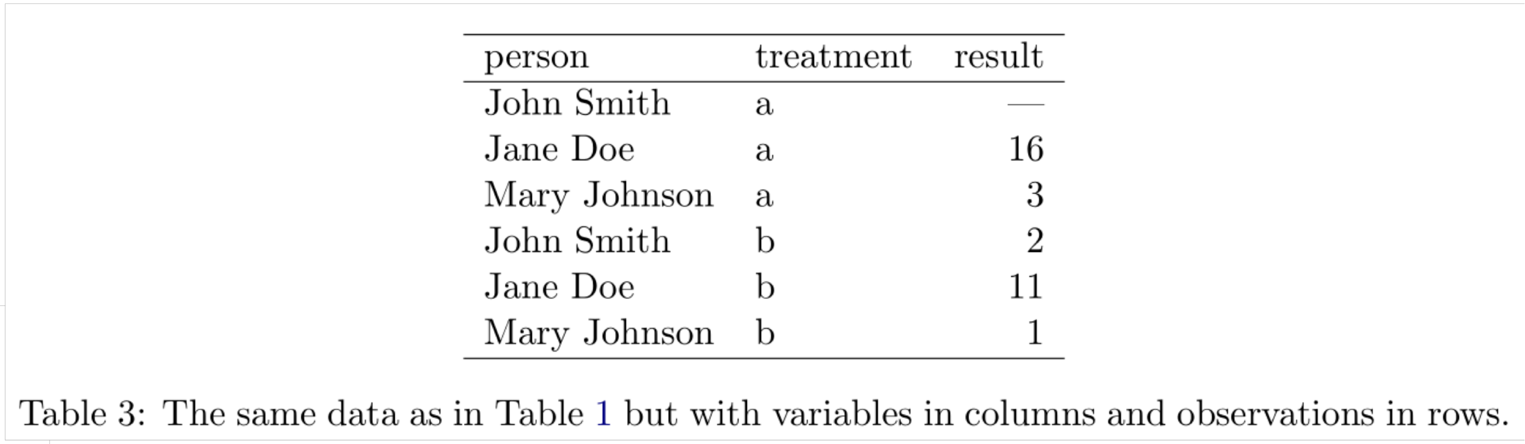
	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

Table 1: Typical presentation dataset.



	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Table 2: The same data as in Table 1 but structured differently.



person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Table 3: The same data as in Table 1 but with variables in columns and observations in rows.

Wikipedia contributors. Tidy Data. *Wikipedia, The Free Encyclopedia*. Oct 23, 2018. https://en.wikipedia.org/wiki/Tidy_data
Wickham, H. (2013). Tidy Data. *Journal of Statistics*. 59(10).

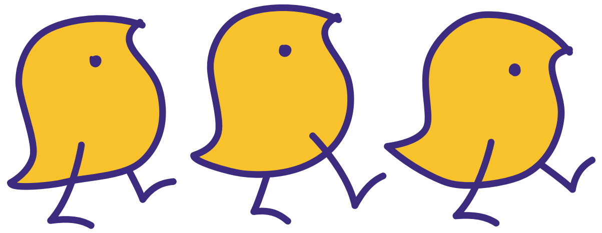
Packages for Korean NLP in R

- Preprocessing
  - Morpheme analyzer: KoNLP, Rmecabko/RcppMeCab
  - Word spacing: KoSpacing
  - Tokenizing/Freq/TDM/…: tm, tidytext
- Transform/Model
  - Topic modeling: topicmodels, stm
  - Word embedding: wordVectors, fastTextR, …
  - Modeling packages like caret, glm, xgboost, …
  - Deep Learning: Tensorflow, Keras, …
- Visualize
  - ggplot2
  - wordcloud

tidytext?

- *Text Mining with R: a Tidy Approach* by Julia Silge and David Robinson[1]
- Hadley Wickham's **tidy data**
  - Each variable is a column
  - Each observation is a row
  - Each type of observational unit is a table
- For text: "a table with one-token-per-row"
- Token: single word, n-gram, sentence, paragraph, …

[1] *Text Mining with R*. <https://www.tidytextmining.com/>



Introducing RcppMeCab

LangCon 2019

NLP Research

- Research with English text
- Research with Korean text

- Word frequency based models works fine, but …
- NLP 연구자가 아닌 인문학, 사회학 연구자라면 한글 텍스트를 가지고 어떤 결론을 내는 것은 시기상조일 수 있음

Addressing Korean Texts

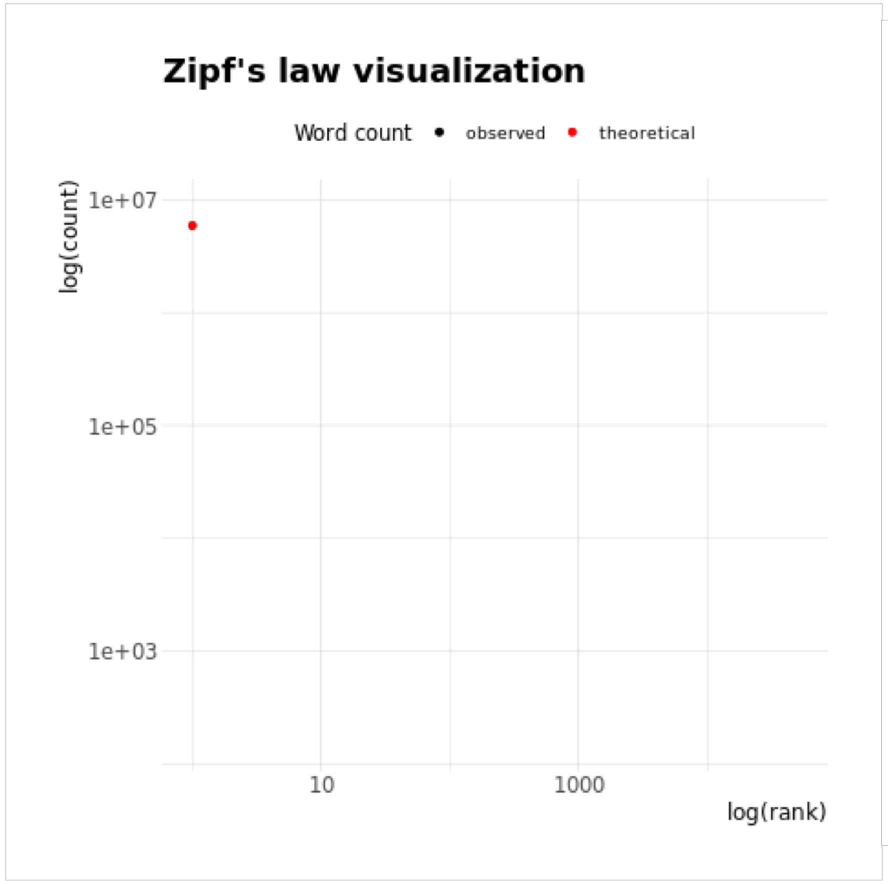
- 연구 정확도가 떨어진다. (예: "… 머신 러닝 기법으로 베이즈 분류기 모형을 설정했고, 3000개의 댓글로 테스트 했을 때 대략 80%의 정확도를 확인할 수 있었기에 해당 모형으로 나머지 분류를 진행했다")
- 왜?
  - 한글 텍스트에 관한 연구자의 이해 부족
  - 한글 텍스트와 frequency-based model이 딱 맞아 떨어지지 않음

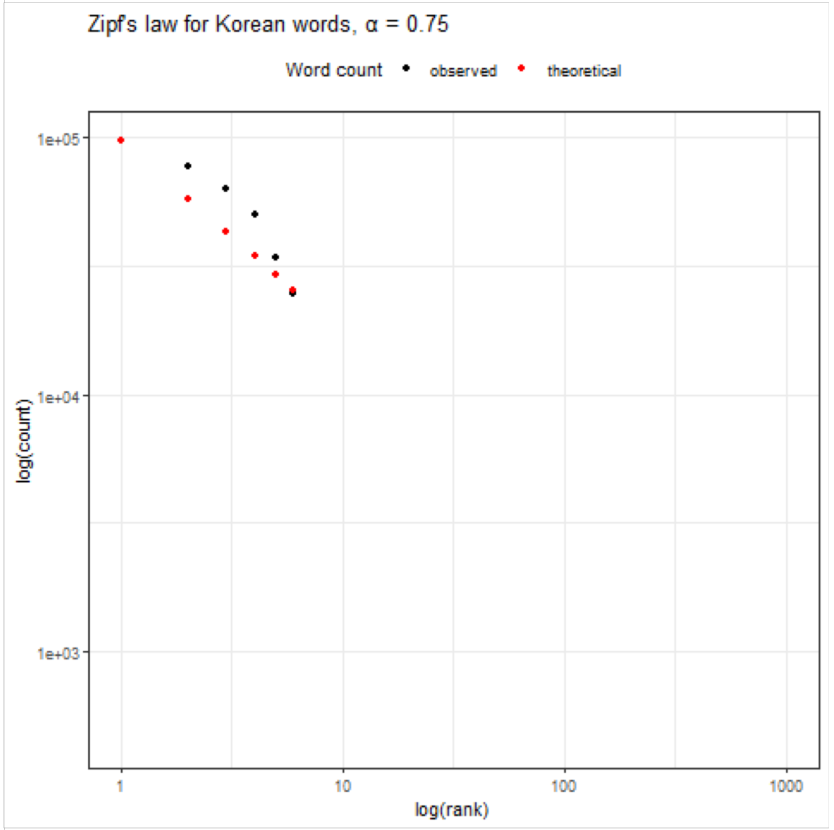
박명석, 권영진, 이상용. (2018). 댓글이 음원 판매량에 미치는 차별적 영향에 관한 텍스트마이닝 분석. 지식경영연구. 19(2). 91-108쪽.

Zipf's Law and Korean

Language Conference 2019

- Zipf's Law: The frequency of any word is inversely proportional to its rank in the frequency table





"The curve slopes more gently than that of English text or of French text. In other words, the Mandelbrot's '*information temperature*', 1/B, should be larger than English or French."

Choi, S. W. (2000). Some Statistical Properties and Zipf's Law in Korean Text Corpus. *J Quantitative Linguistics*. 7(1). 19-30.

Based on Maj M. (Feb 7, 2019). "Investigating words distribution with R—Zipf's law". *Appsilon Data Science*. Korean word frequency is from 김한샘. (2005). 현대 국어 사용 빈도 조사 2. 국립국어원

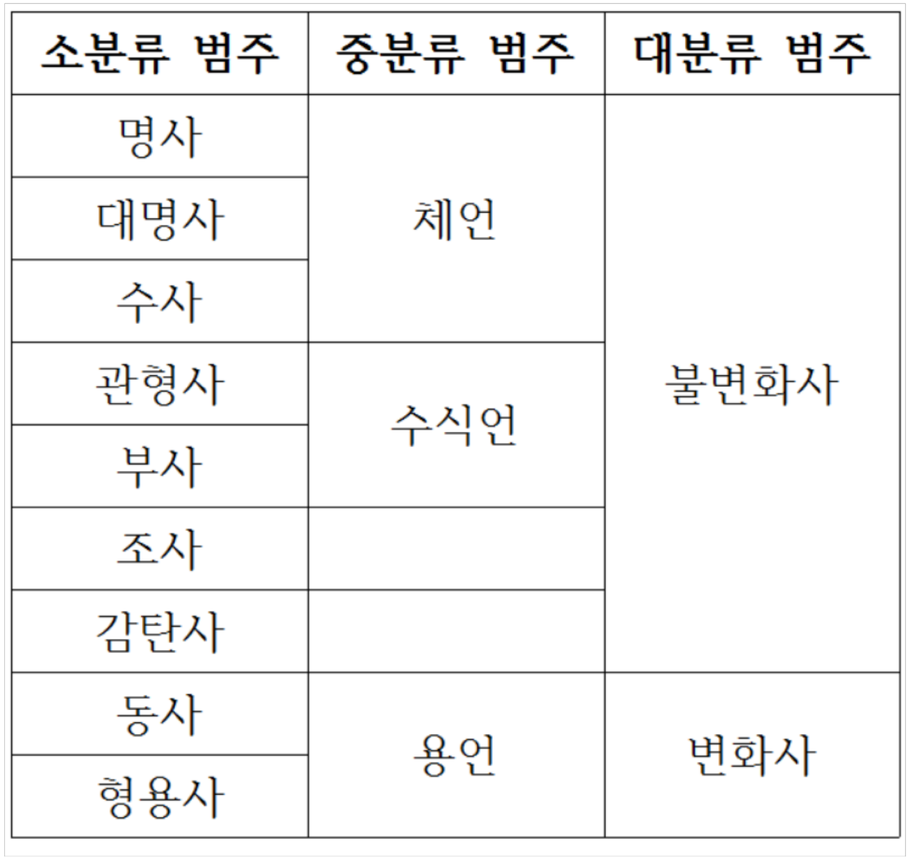
한글 토큰

- 띄어쓰기 기준?
- 형태소 기준?
- 형태소+품사 기준?

Morpheme Analysis

Language Conference 2019

- “어떤 대상 어절을 최소의 의미 단위인 ‘형태소’로 분석하는 것”
- 정보 검색 엔진
- 난점
  - 미등록어, 오탈자, 띄어쓰기 오류
  - 중의성, 신조처 처리
  - 복합 명사 분해



소분류 범주	중분류 범주	대분류 범주
명사	체언	불변화사
대명사		
수사		
관형사	수식언	
부사		
조사		
감탄사		
동사	용언	변화사
형용사		

자연어 처리. (2018년 9월 12일). 위키백과.
<https://ko.wikipedia.org/wiki/%EC%9E%90%EC%97%B0%EC%96%B4_%EC%B2%98%EB%A6%AC>
한국어 품사 분류와 분포(distribution). (Apr 21, 2017). *ratsgo’s blog*.
<https://ratsgo.github.io/korean%20linguistics/2017/04/21/wordclass/>

KoNLP? KoNLPy? Khaiii?

- KoNLP (https://github.com/haven-jeon/KoNLP)
  - A R package for Korean NLP
  - Hannanum (한나눔, 카이스트) analyzer
  - Based on Java
- KoNLPy (http://konlpy.org/en/latest/)
  - A Python package for NLP of the Korean language
  - Hannanum, Kkma (꼬꼬마, 서울대), Komoran (샤인웨어), Mecab, Okt (Open Korean Text)
- Khaiii (https://github.com/kakao/khaiii)
  - Kakao Hangul Analyzer III
  - 21세기 세종계획 최종 성과물 보완, 85만 문장
  - CNN-based

MeCab

- “Open-source text segmentation library for use with text written in the Japanese language.”[1]
- Taku Kudou (工藤拓) maintains the project.
- Bi-gram Markov model with CRF (identification model)[2]
- MeCab ~~[미캡]~~ [메카브]

Viterbi algorithm[3]
A dynamic programming algorithm for finding the most likely sequence of hidden states
Conditional Random Fields (CRFs)[4]
An undirected graphical model whose nodes can be divided into exactly two disjoint sets X and Y
The conditional distribution p(Y|X) is modeled

[1] MeCab. *Wikipedia, the free encyclopedia*. <https://en.wikipedia.org/wiki/MeCab>
[2] Taku Kudou. *MeCab*. <http://taku910.github.io/mecab/>
[3] Wikipedia contributors. (Feb 1, 2019). Viterbi algorithm. *Wikipedia, The Free Encyclopedia*.
[4] Wikipedia contributors. (Jan 30, 2019). Conditional random field. *Wikipedia, The Free Encyclopedia*.

은전한닢 프로젝트

- "검색에서 쓸만한 오픈소스 한국어 형태소 분석기를 만들자!"[1]
- mecab-ko, MeCab fork project for Korean
- mecab-ko-dic, MeCab 용 한국어 형태소 사전
  - 21세기 세종계획 모든 현대 말뭉치에서 50문장씩 추출해서 학습[2]

[1] 은전한닢 프로젝트를 소개합니다. 2013년 2월 12일. <http://eunjeon.blogspot.com/2013/02/blog-post.html>
[2] mecab-ko-dic 소개. <https://bitbucket.org/eunjeon/mecab-ko-dic>

개인적으로 생각하던 MeCab의 장점

- 빠르다.
- 띄어쓰기를 신경 쓰지 않아도 된다.
  - 아직 전희원님의 KoSpacing이 나오기 전이었기에…[1]

[1] *KoSpacing*. <https://github.com/haven-jeon/KoSpacing>

하지만...

Language Conference 2019

- 텍스트 분석을 막 익히던 2013-14년에는 Python의 KoNLPy와 R의 KoNLP가 대표적인 형태소 분석 패키지
- "한글 분석은 당연히 Python에서 해야 하는 것 아냐?"
- R의 단점
  - Encoding

그렇다면...

- 문제를 해결해보자. 어떻게? 패키지 제작
- RmecabKo (First release in CRAN: 2017-10-3)
  - 개인 repo에 일본어용 RMeCab이 있음을 확인[1]
  - "그렇다면 한글을 위한 (정확히는 mecab-ko를 위한) R 패키지도 개발할 수 있지 않을까?"
  - 문제는 C, C++ programming 을 몰랐다는 것. 지금도 잘 몰라요.
  - RMeCab code를 보다가 이게 MeCab 예제에 있는 C, C++ API 코드와 구조가 같다는 것을 알게 되었으나…

[1] *RMeCab*. <http://rmecab.jp/wiki/index.php?RMeCab>

RmecabKo

Language Conference 2019

- RCpp 패키지 설명을 읽고 뜯고 헤매다가 당시 컴퓨터에서 어떻게 했는지도 모르는채 MeCab를 R 상에서 실행에 성공
- "좋아! 성공이다!" … 그럴리가요.
- 온갖 시행착오(그야말로 trial-and-error)를 거쳐 RmecabKo 초기 버전 완성하고 Github에 업로드
- 목표: "tidytext를 한글 분석에 쉽게 적용할 수 있는 한글 품사 분석 패키지 개발"

RcppMeCab

Language Conference 2019

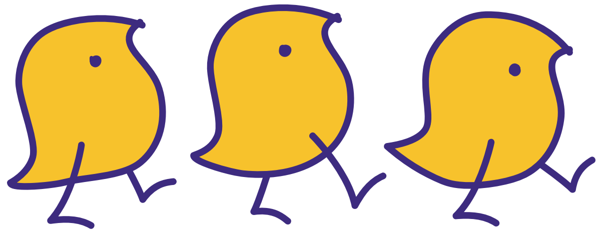
- 전희원 님: “RmecabKo를 R 아시아권 언어 개발자들 포럼에서 소개하고 싶어요.”
- 나: “RmecabKo는 한글 용인데요…”
- 전희원 님: “UTF-8 표준으로 CJK에 다 적용할 수 있지 않아요?”
- 나: “아하하… 물론 그렇긴 한데…”

RcppMeCab release

- First release in Github: 2018-5-18 (0.0.0.1)
- First release in CRAN: 2018-10-7 (0.0.1.1)
- 목적: “MeCab engine/dictionary 만 바꾸면 CJK 다 분석할 수 있는 범용 형태소 분석기 for R”
- Input encoding 을 UTF-8 으로 한정

향후 계획

- RcppMeCab 개선
  - Output 양식 추가
- RmecabKo 개선
  - RcppMeCab을 import한 후 추가 기능을 넣는 방식으로 바꿀 예정
  - Input encoding 강제 (with enc2utf8)
  - 추가기능: 사용자 사전(individual & system), mecab-ko & mecab-ko-dic 설치 함수 (in Windows), 형태소별 추출, n-gram 기능 확대, sentiment 사전 정리
  - 추가하지 않을 것: 불용어 사전



사례: 한글 텍스트 분류하기

예제: Text Classification with Tidy Data Principles [1]

Language Conference 2019

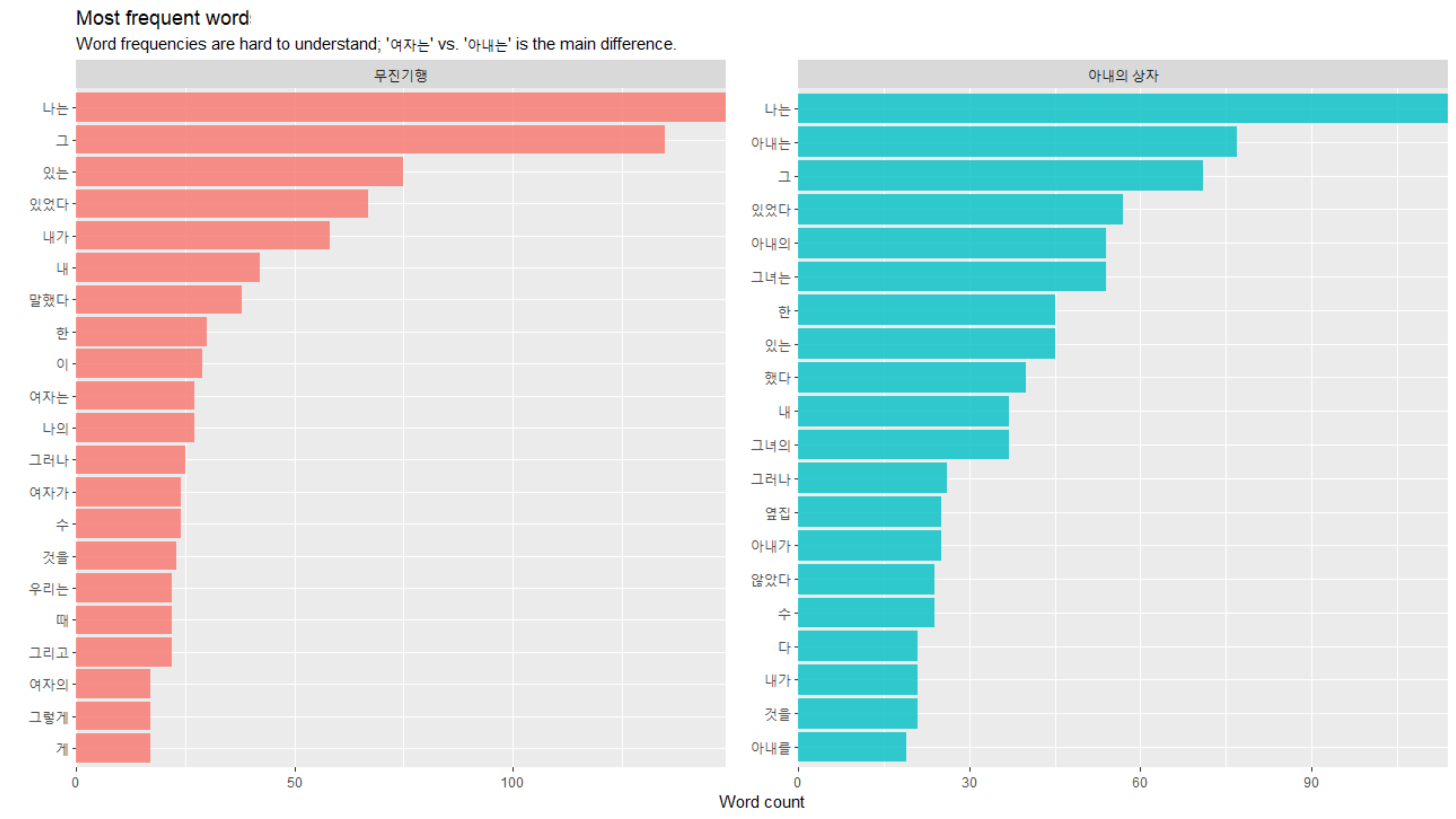
```
library(tidyverse)
library(tidytext)
library(RcppMeCab)
library(rsample)
library(glmnet)
library(doParallel) # Parallelization in Windows
library(broom)
library(yardstick)
```

[1] Silge J. Text Classification with Tidy Data Principles. <https://juliasilge.com/blog/tidy-text-classification/>

```
# 무진기행(김승옥), 아내의 상자(은희경) 비교

con <- file("김승옥_무진기행.txt")
mujin <- readLines(con)
close(con)
mujin <- iconv(mujin, "CP949", "UTF-8")

con <- file("아내의 상자 - 은희경.txt", encoding = "UTF-8")
box <- readLines(con)
close(con)

books <- data_frame(text = mujin, title = enc2utf8("무진기행")) %>%
  rbind(data_frame(text = box, title = enc2utf8("아내의 상자"))) %>%
  mutate(document = row_number())

books
```

```
                                text                                   title      document
무진기행                                                                무진기행          1
김승옥                                                                  무진기행          2
                                                                        무진기행          3
버스가 산모퉁이를 돌아갈 때 나는 …                                       무진기행          4
\앞으로 십킬로 남았군요.\""                                              무진기행          5
\예, 한 삼십분 후에 도착할 겁니다.\""                                    무진기행          6
그들은 농사 관계의 시찰원들인 듯했다. 아니 그렇지 않은 ….                 무진기행          7
\무진엔 명산물이…… 뭐 별로 없지요?\""                                    무진기행          8
```

```
# 띄어쓰기 기준 tokenization

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

tidy_books %>%
  count(title, word, sort = TRUE) %>%
  anti_join(get_stopwords()) %>%
  group_by(title) %>%
  top_n(20) %>%
  ungroup() %>%
  ggplot(aes(reorder_within(word, n, title), n, fill = title)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  scale_x_reordered() + coord_flip() + facet_wrap(~ title, scales = "free") +
  scale_y_continuous(expand = c(0, 0)) +
  labs(x = NULL, y = "Word count",
    title = "Most frequent words after removing stop words",
    subtitle = "Word frequencies are hard to understand; '여자는' vs. '아내는' is the main difference.")
```

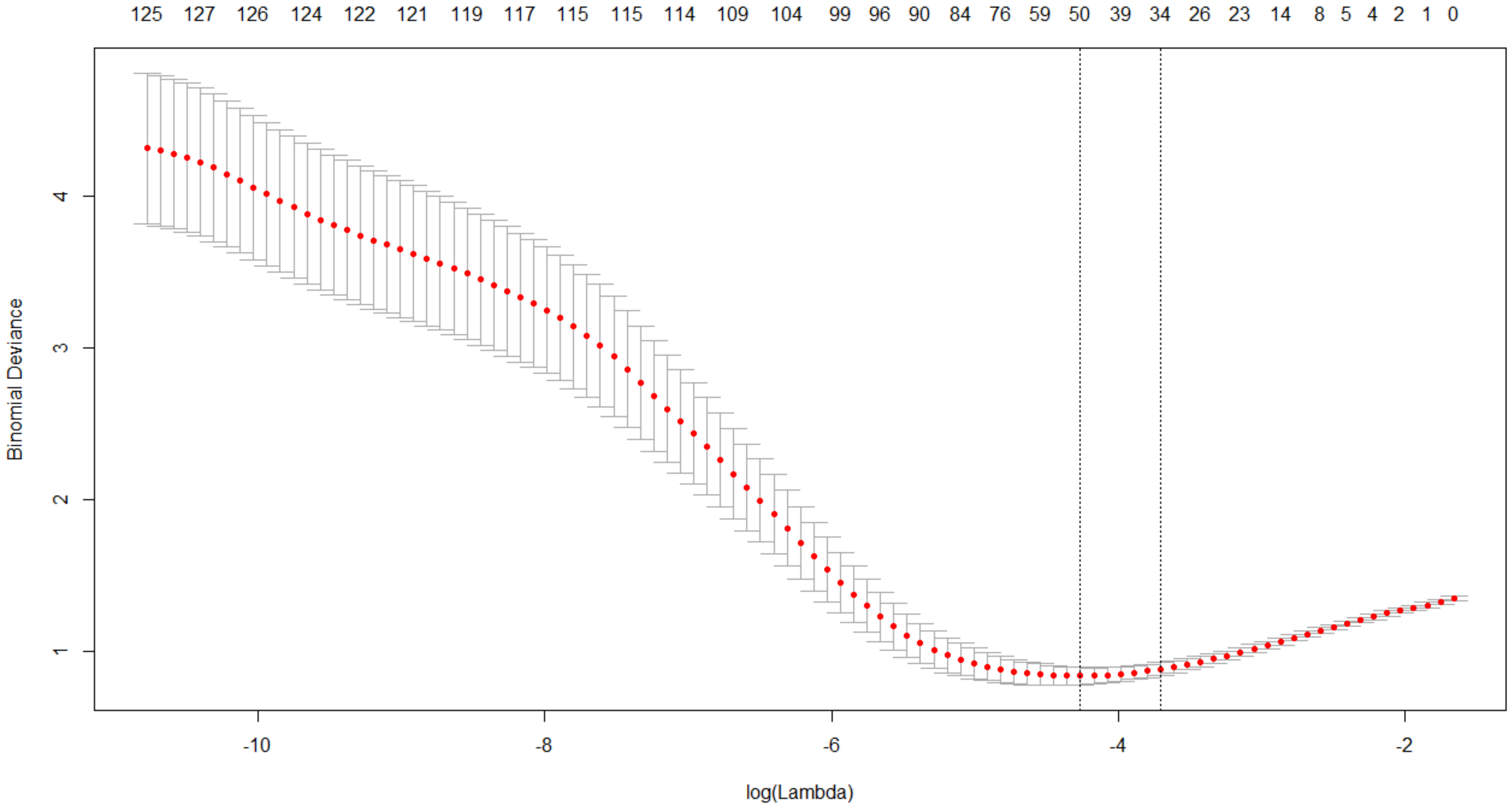

Most frequent word
Word frequencies are hard to understand; '여자는' vs. '아내는' is the main difference.
무진기행
나는
그
있는
있었다
내가
내
말했다
한
이
여자는
나의
그러나
여자가
수
것을
우리는
때
그리고
여자의
그렇게
게
0
50
100
아내의 상자
나는
아내는
그
있었다
아내의
그녀는
한
있는
했다
내
그녀의
그러나
옆집
아내가
않았다
수
다
내가
것을
아내를
0
30
60
90
Word count

```
# glmnet training을 위한 training/test set separation

books_split <- books %>%
  select(document) %>%
  initial_split()
train_data <- training(books_split)
test_data <- testing(books_split)

sparse_words <- tidy_books %>%
  count(document, word) %>%
  inner_join(train_data) %>%
  cast_sparse(document, word, n)

class(sparse_words) # [1] “dgCMatrix”

dim(sparse_words) # [1] 342 147

word_rownames <- as.integer(rownames(sparse_words))

books_joined <- data_frame(document = word_rownames) %>%
  left_join(books %>% select(document, title))
```

```
# glmnet training

registerDoParallel(4)

is_box <- books_joined$title == "아내의 상자"
model <- cv.glmnet(sparse_words, is_box, family = "binomial", parallel = TRUE, keep = TRUE)

plot(model)

plot(model$glmnet.fit)
```

"**Glmnet** is a package that fits a generalized linear model via penalized maximum likelihood."
Automated lambda selection by mean-squared error (Gaussian) or binomial deviance (binomial)

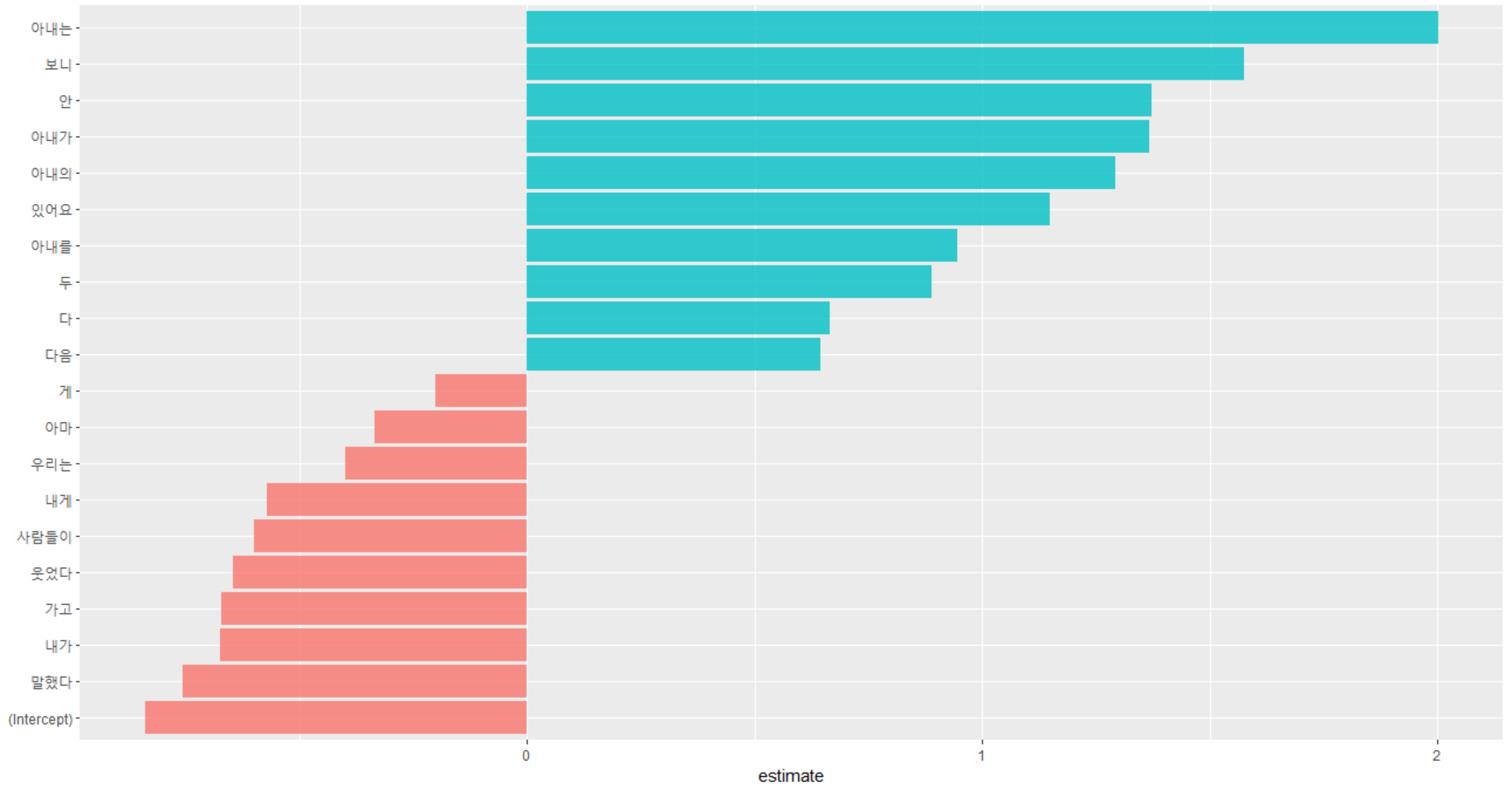
Hastie T., Qian J. (Jun 26, 2014). Glmnet Vignette. <https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html>


125 127 126 124 122 121 119 117 115 115 114 109 104 99 96 90 84 76 59 50 39 34 26 23 14 8 5 4 2 1 0
Binomial Deviance
1
2
3
4
-10
-8
-6
-4
-2
log(Lambda)

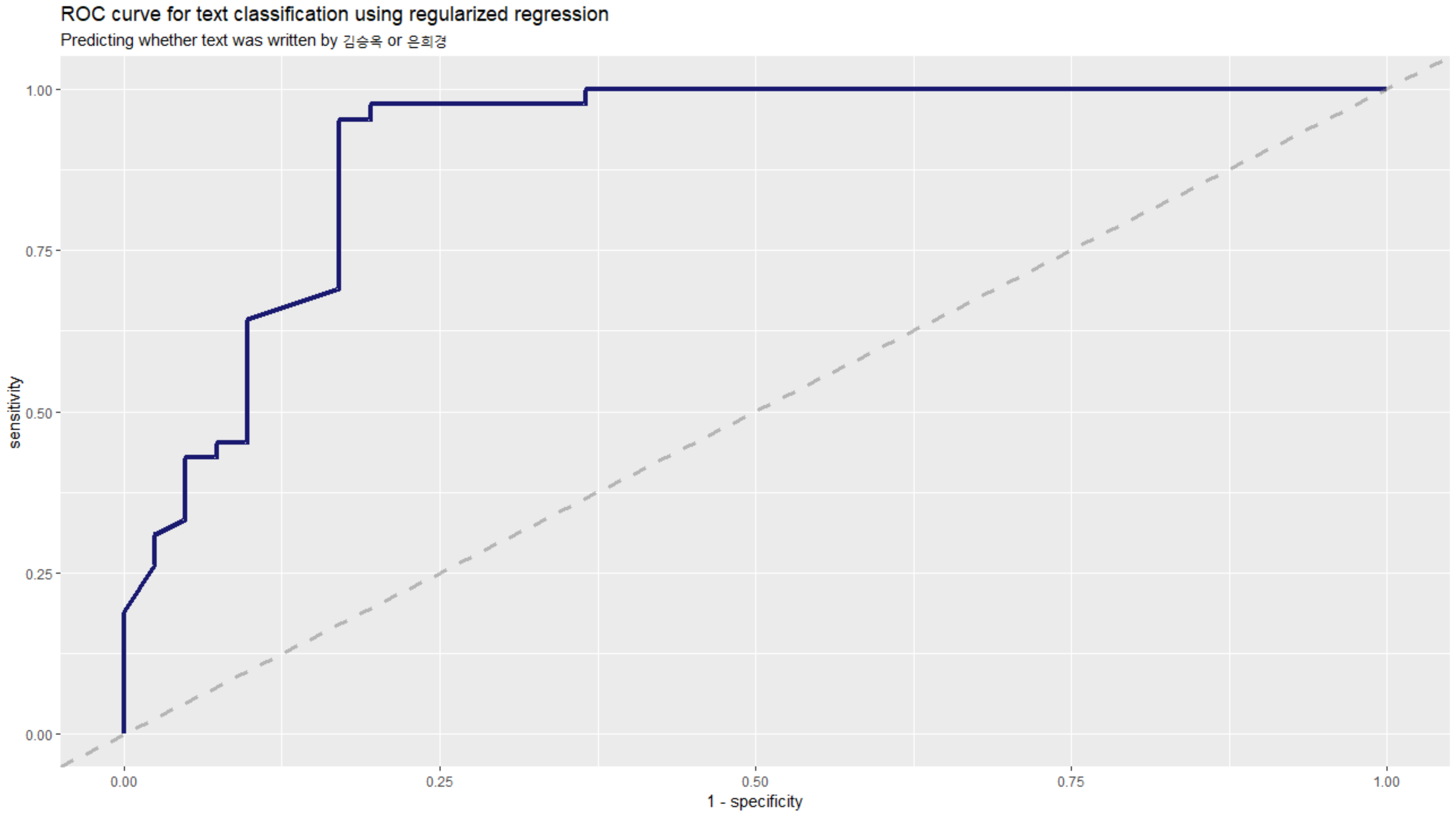
```r
# word probabilities

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

coefs %>%
  group_by(estimate > 0) %>%
  top_n(10, abs(estimate)) %>%
  ungroup() %>%
  ggplot(aes(fct_reorder(term, estimate), estimate, fill = estimate > 0)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  coord_flip() +
  labs(
    x = NULL,
    title = "Coefficients that increase/decrease probability the most",
    subtitle = "A document mentioning 말했다 is unlikely to be written by 은희경"
  )
```

Coefficients that increase/decrease probability the most

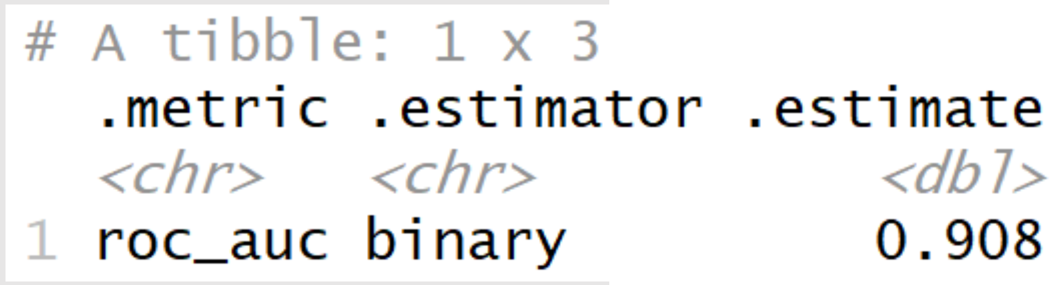
A document mentioning 말했다 is unlikely to be written by 은희경



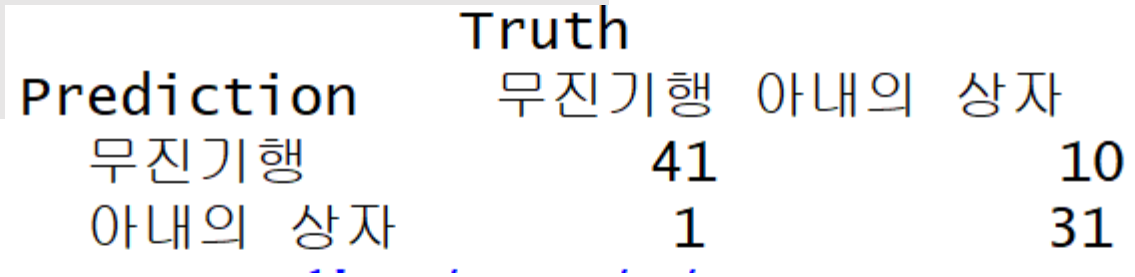
```r
# ROC curve

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

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

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

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) +
  labs(title = "ROC curve for text classification using regularized regression",
    subtitle = "Predicting whether text was written by 김승옥 or 은희경")
```


ROC curve for text classification using regularized regression
Predicting whether text was written by 김승옥 or 은희경
sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 - specificity

```
# AUC & confusion matrix

comment_classes %>%
  roc_auc(title, probability)

comment_classes %>%
  mutate(
    prediction = case_when(probability > 0.5 ~ "아내의 상자",
                           TRUE ~ "무진기행"),
    prediction = as.factor(prediction)
  ) %>%
  conf_mat(title, prediction)
```



```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.908
```



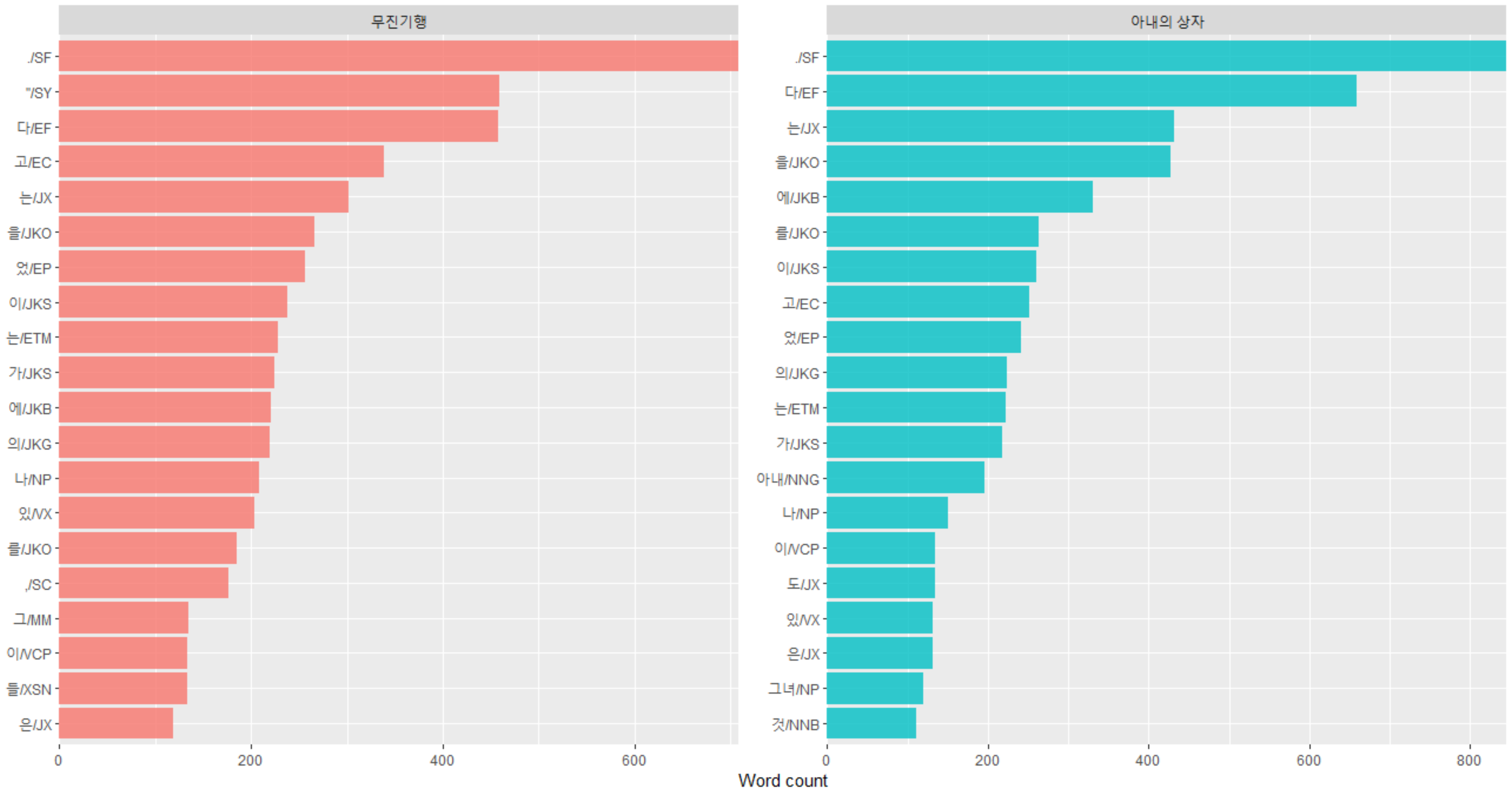
```
                Truth
Prediction      무진기행 아내의 상자
  무진기행            41          10
  아내의 상자          1          31
```

yardstick::roc_auc(data, truth, estimate)
yardstick::conf_mat(data, truth, estimate)

```
# 형태소 기준 tokenization

tidy_books <- books %>%
  unnest_tokens(word, text, token = pos, to_lower = FALSE) %>%
  group_by(word) %>%
  filter(n() > 10) %>%
  ungroup()

tidy_books %>%
  count(title, word, sort = TRUE) %>%
  anti_join(get_stopwords()) %>%
  group_by(title) %>%
  top_n(20) %>%
  ungroup() %>%
  ggplot(aes(reorder_within(word, n, title), n, fill = title)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  scale_x_reordered() + coord_flip() + facet_wrap(~ title, scales = "free") +
  scale_y_continuous(expand = c(0, 0)) +
  labs(x = NULL, y = "Word count",
    title = "Most frequent words after removing stop words",
    subtitle = "Word frequencies are hard to understand; only '아내/NNG' for 아내의 상자 is noticeable
difference.")
```

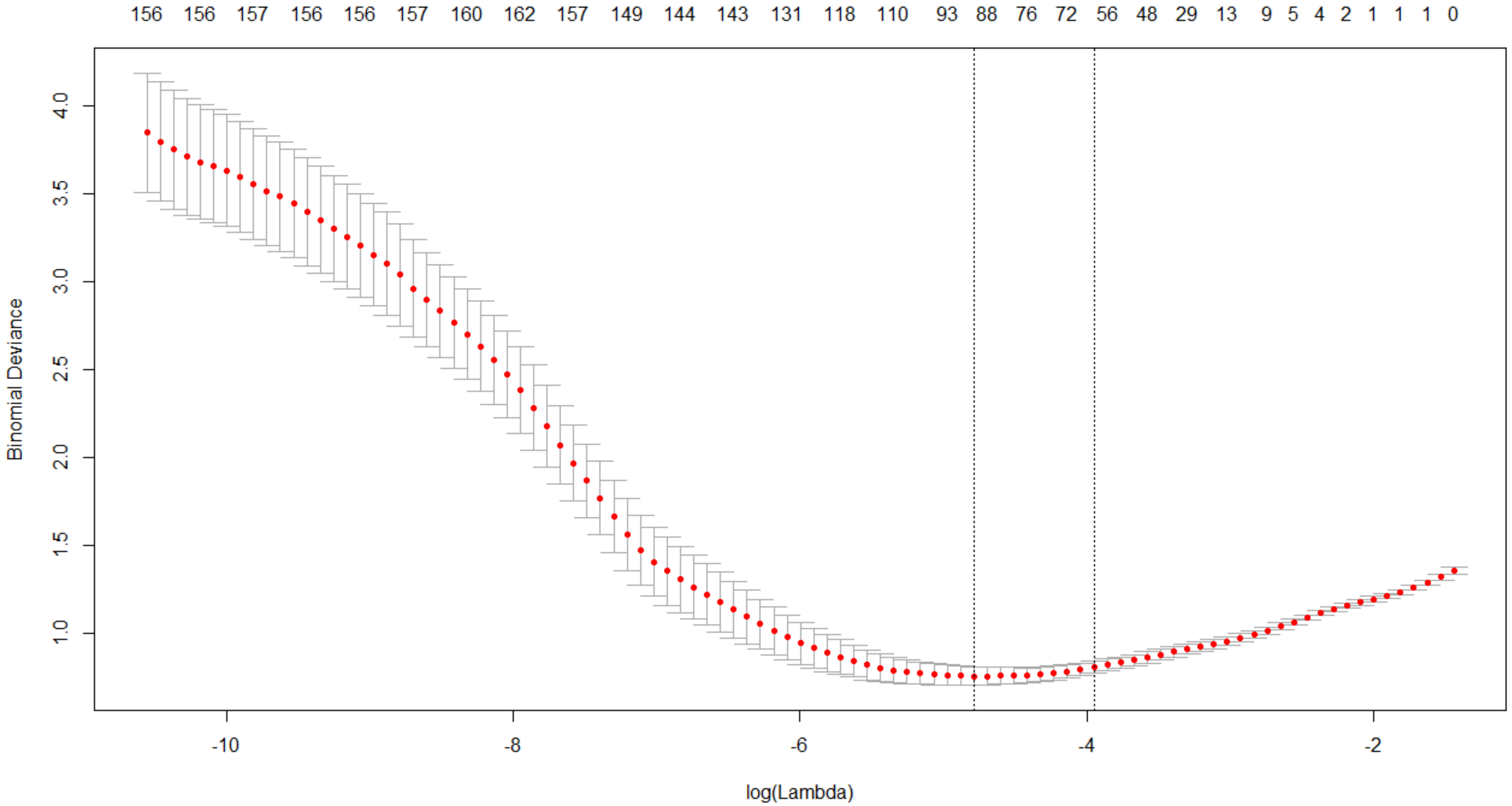
Most frequent words

Word frequencies are hard to understand; only '아내/NNG' for 아내의 상자 is noticeable difference.

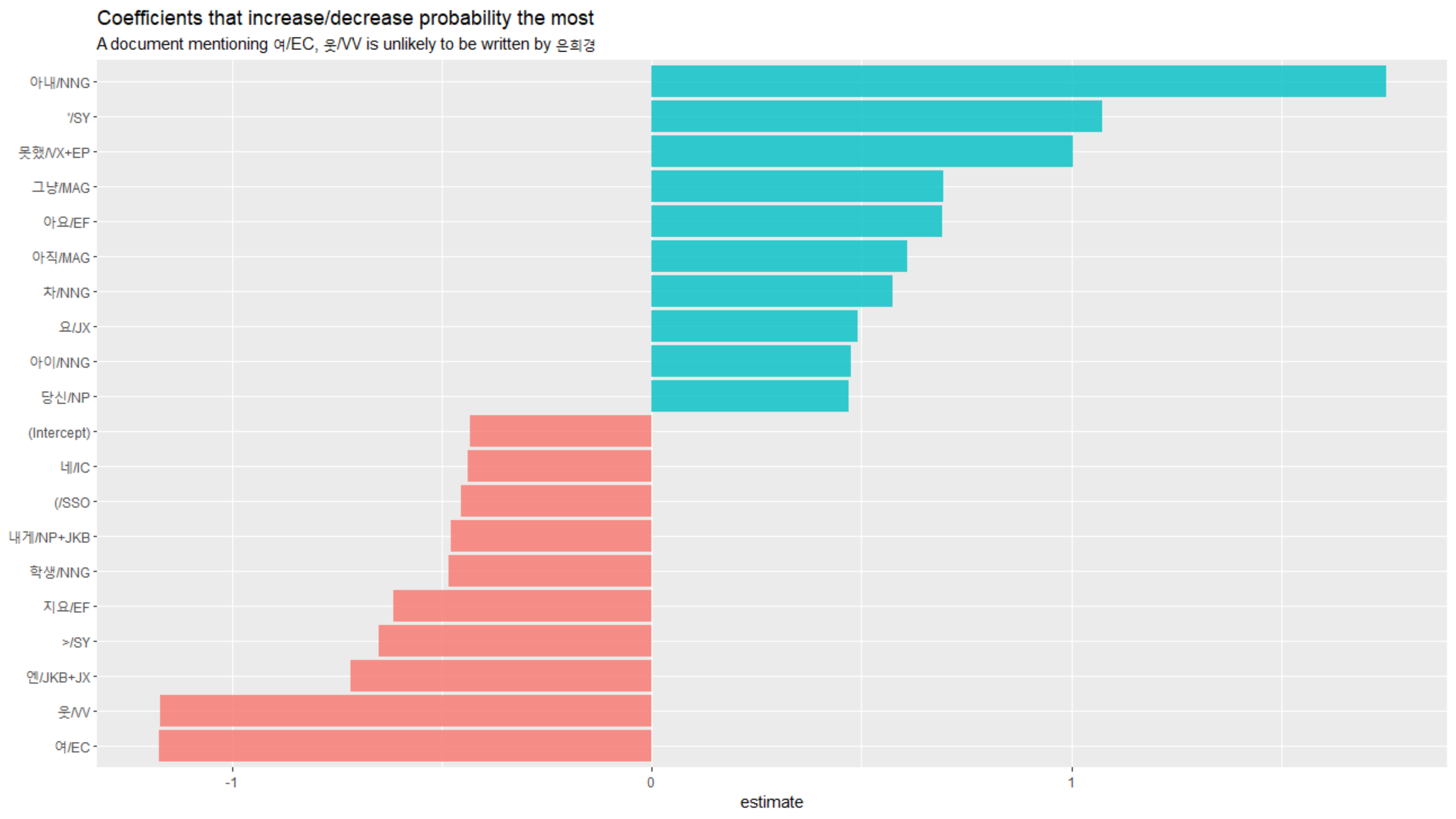


```
# glmnet training을 위한 training/test set separation

books_split <- books %>%
  select(document) %>%
  initial_split()
train_data <- training(books_split)
test_data <- testing(books_split)

sparse_words <- tidy_books %>%
  count(document, word) %>%
  inner_join(train_data) %>%
  cast_sparse(document, word, n)

class(sparse_words) # [1] "dgCMatrix"

dim(sparse_words) # [1] 342 147

word_rownames <- as.integer(rownames(sparse_words))

books_joined <- data_frame(document = word_rownames) %>%
  left_join(books %>% select(document, title))
```

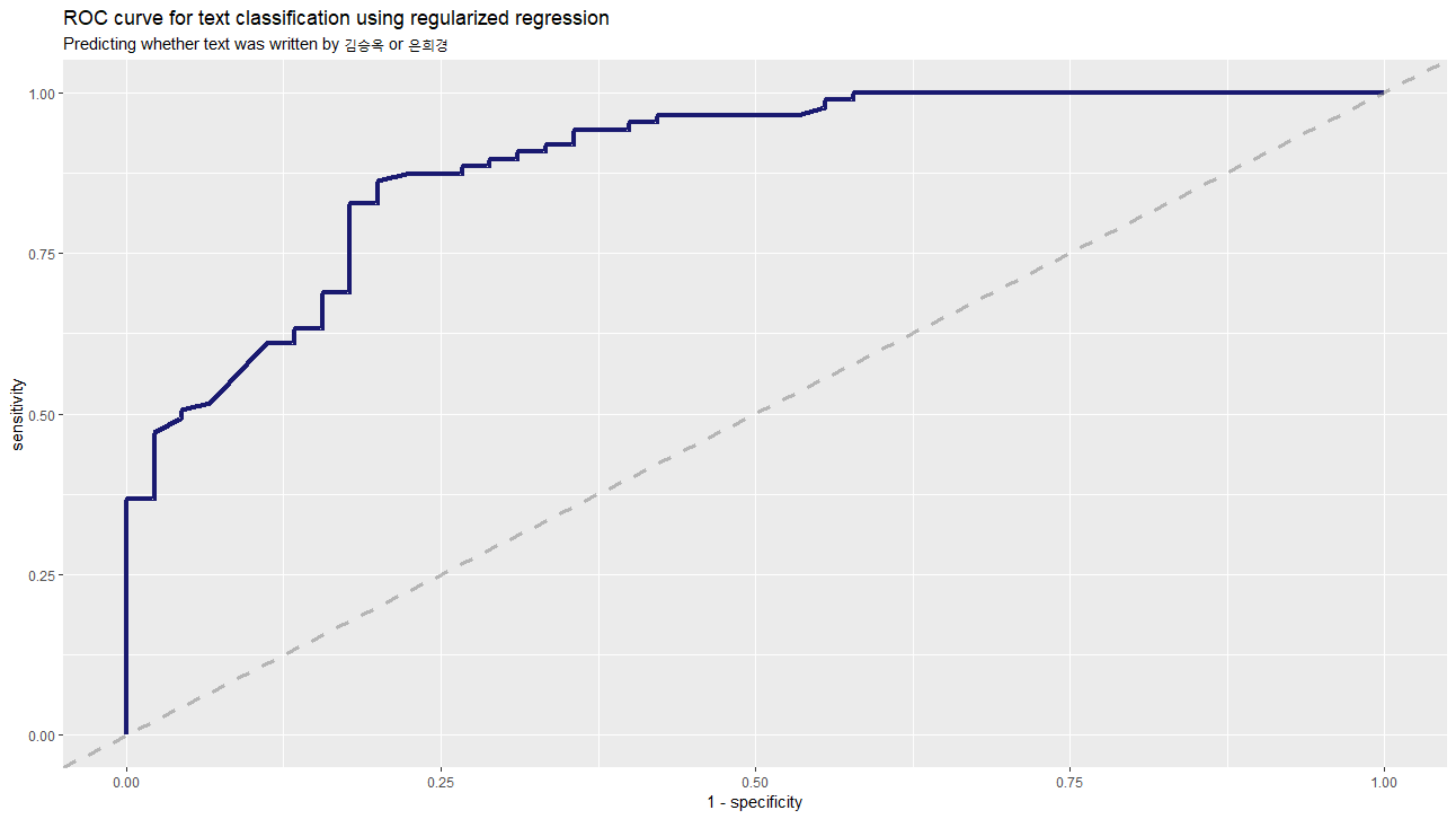
```
# glmnet training

registerDoParallel(4)

is_box <- books_joined$title == "아내의 상자"
model <- cv.glmnet(sparse_words, is_box, family = "binomial", parallel = TRUE, keep = TRUE)

plot(model)

plot(model$glmnet.fit)
```


156 156 157 156 156 157 160 162 157 149 144 143 131 118 110 93 88 76 72 56 48 29 13 9 5 4 2 1 1 1 0
Binomial Deviance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
-10
-8
-6
-4
-2
log(Lambda)

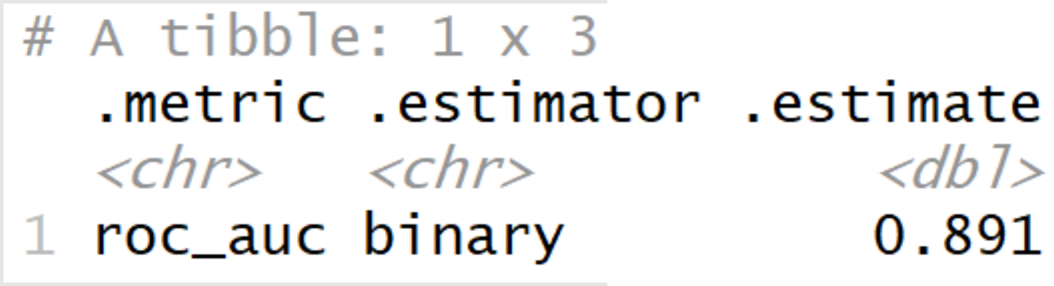
```r
# word probabilities

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

coefs %>%
  group_by(estimate > 0) %>%
  top_n(10, abs(estimate)) %>%
  ungroup() %>%
  ggplot(aes(fct_reorder(term, estimate), estimate, fill = estimate > 0)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  coord_flip() +
  labs(
    x = NULL,
    title = "Coefficients that increase/decrease probability the most",
    subtitle = "A document mentioning 말했다 is unlikely to be written by 은희경"
  )
```


Coefficients that increase/decrease probability the most
A document mentioning 여/EC, 웃/VV is unlikely to be written by 은희경
아내/NNG
'/SY
못했/VX+EP
그냥/MAG
아요/EF
아직/MAG
차/NNG
요/JX
아이/NNG
당신/NP
(Intercept)
네/IC
(/SSO
내게/NP+JKB
학생/NNG
지요/EF
>/SY
엔/JKB+JX
웃/VV
여/EC
-1
0
1
estimate

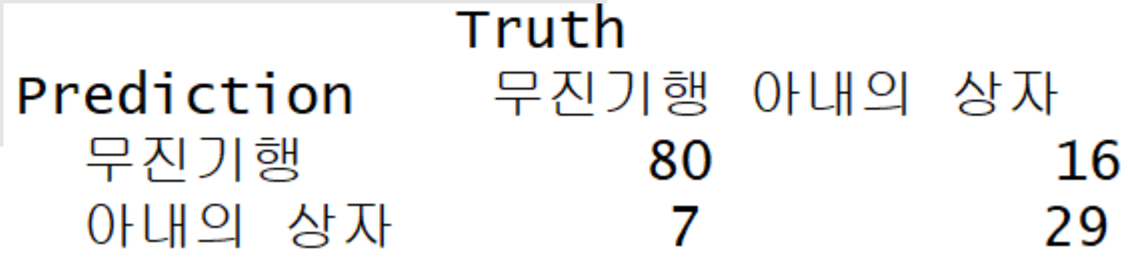
```
# ROC curve

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

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

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

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) +
  labs(title = "ROC curve for text classification using regularized regression",
    subtitle = "Predicting whether text was written by 김승옥 or 은희경")
```


ROC curve for text classification using regularized regression
Predicting whether text was written by 김승옥 or 은희경
sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 - specificity

```
# AUC & confusion matrix

comment_classes %>%
  roc_auc(title, probability)

comment_classes %>%
  mutate(
    prediction = case_when(probability > 0.5 ~ "아내의 상자",
                           TRUE ~ "무진기행"),
    prediction = as.factor(prediction)
  ) %>%
  conf_mat(title, prediction)
```



```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.891
```

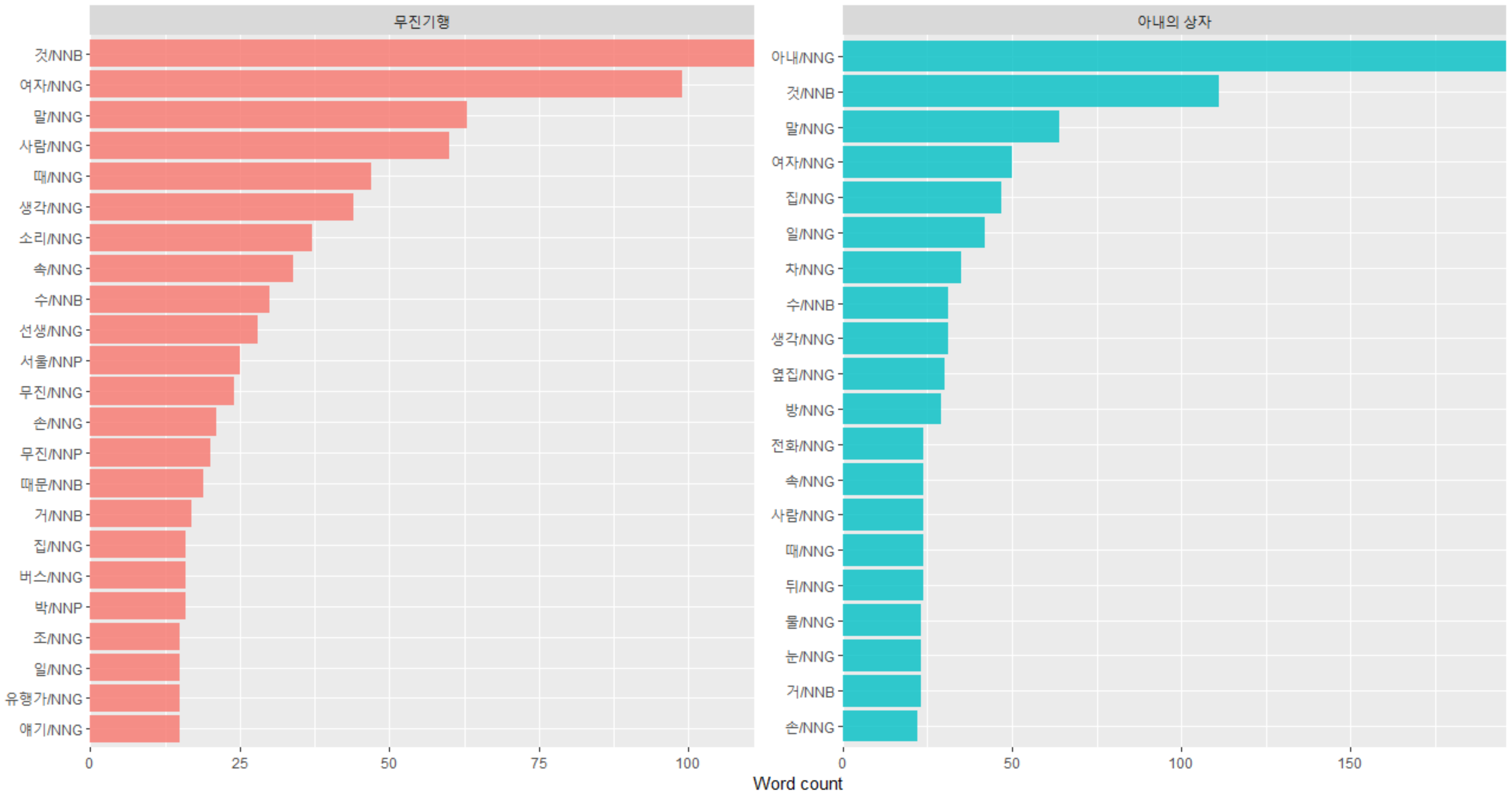


```
             Truth
Prediction    무진기행 아내의 상자
  무진기행          80          16
  아내의 상자        7          29
```

```r
# 형태소 기준 tokenization, 명사만 추출

tidy_books <- books %>%
  unnest_tokens(word, text, token = pos, to_lower = FALSE) %>%
  group_by(word) %>%
  filter(n() > 10) %>%
  ungroup()

tidy_books %>%
  count(title, word, sort = TRUE) %>%
  anti_join(get_stopwords()) %>%
  group_by(title) %>%
  top_n(20) %>%
  ungroup() %>%
  ggplot(aes(reorder_within(word, n, title), n, fill = title)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  scale_x_reordered() + coord_flip() + facet_wrap(~ title, scales = "free") +
  scale_y_continuous(expand = c(0, 0)) +
  labs(x = NULL, y = "Word count",
    title = "Most frequent words after removing stop words",
    subtitle = "Word frequencies are hard to understand; only '아내/NNG' for 아내의 상자 is noticeable
difference.")
```

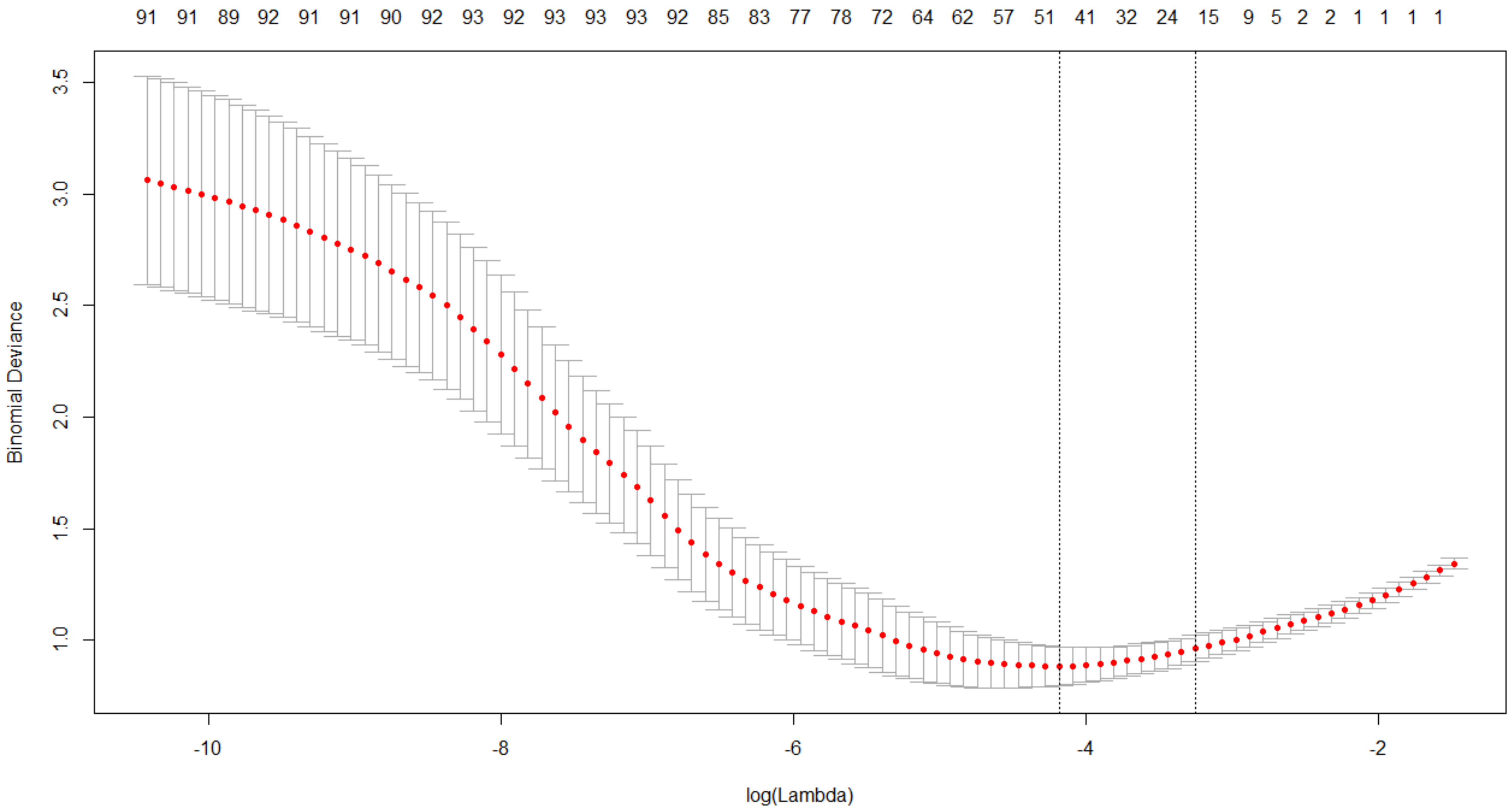
Most frequent words

Words like 것/NNB, 말/NNG occupy similar rank but others are quite different.



```
# glmnet training을 위한 training/test set separation

books_split <- books %>%
  select(document) %>%
  initial_split()
train_data <- training(books_split)
test_data <- testing(books_split)

sparse_words <- tidy_books %>%
  count(document, word) %>%
  inner_join(train_data) %>%
  cast_sparse(document, word, n)

class(sparse_words) # [1] “dgCMatrix”

dim(sparse_words) # [1] 342 147

word_rownames <- as.integer(rownames(sparse_words))

books_joined <- data_frame(document = word_rownames) %>%
  left_join(books %>% select(document, title))
```

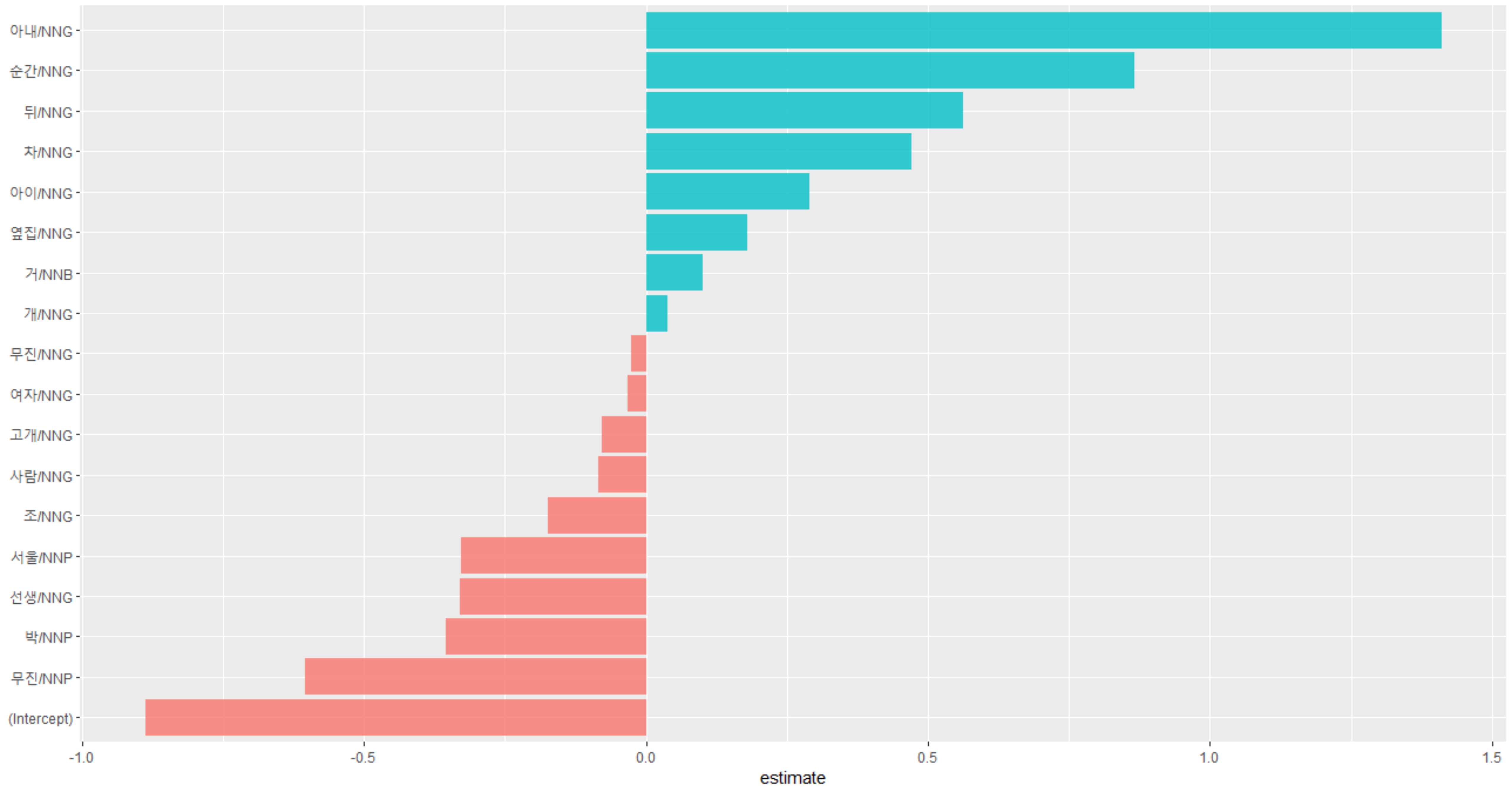
```r
# glmnet training

registerDoParallel(4)

is_box <- books_joined$title == "아내의 상자"
model <- cv.glmnet(sparse_words, is_box, family = "binomial", parallel = TRUE, keep = TRUE)

plot(model)

plot(model$glmnet.fit)
```


91 91 89 92 91 91 90 92 93 92 93 93 93 92 85 83 77 78 72 64 62 57 51 41 32 24 15 9 5 2 2 1 1 1 1
Binomial Deviance
3.5
3.0
2.5
2.0
1.5
1.0
-10
-8
-6
-4
-2
log(Lambda)

```
# word probabilities

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

coefs %>%
  group_by(estimate > 0) %>%
  top_n(10, abs(estimate)) %>%
  ungroup() %>%
  ggplot(aes(fct_reorder(term, estimate), estimate, fill = estimate > 0)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  coord_flip() +
  labs(
    x = NULL,
    title = "Coefficients that increase/decrease probability the most",
    subtitle = "A document mentioning 말했다 is unlikely to be written by 은희경"
  )
```

Coefficients that increase/decrease probability the most

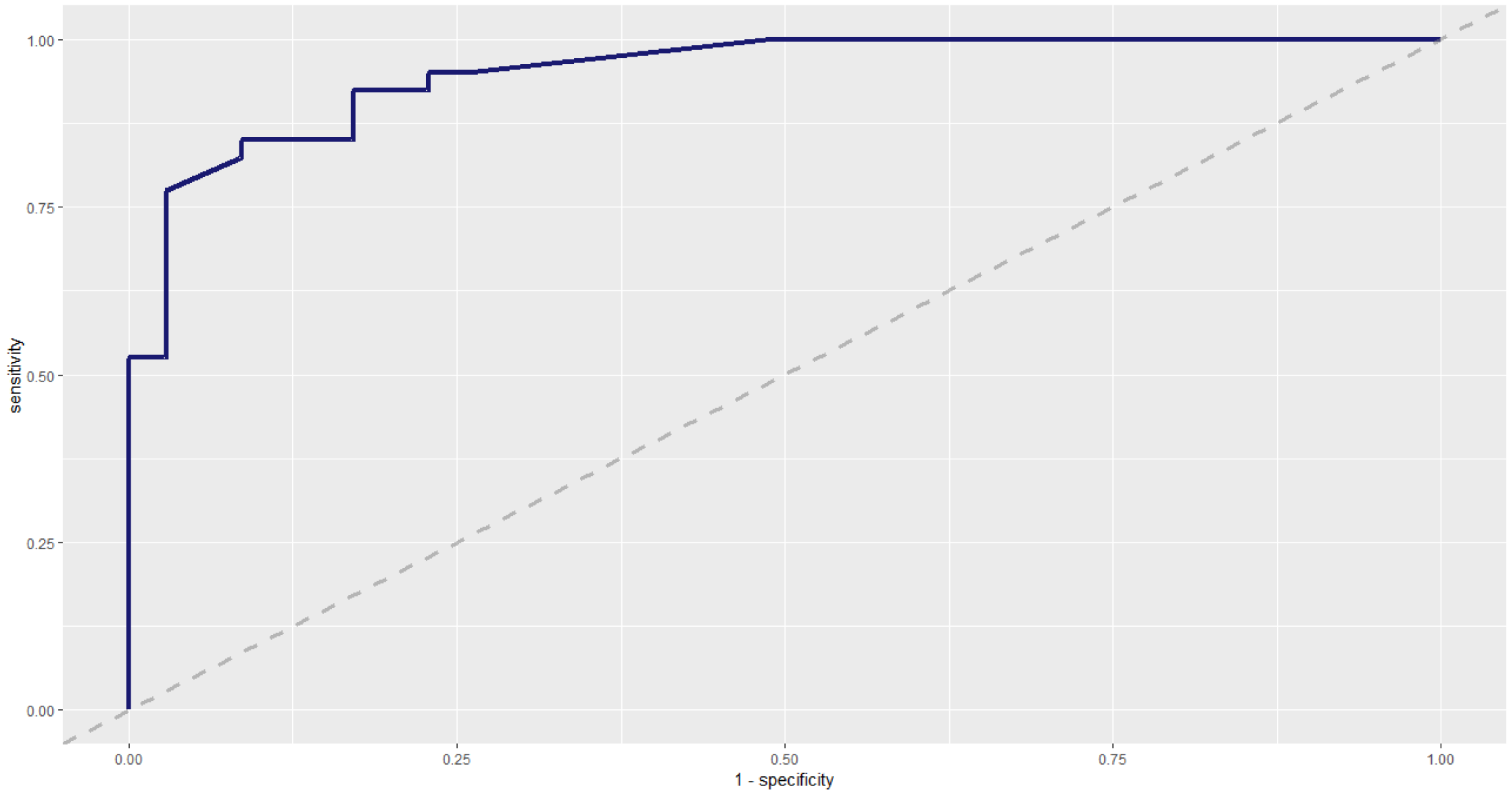
A document mentioning 무진/NNP, 박/NNP is unlikely to be written by 은희경



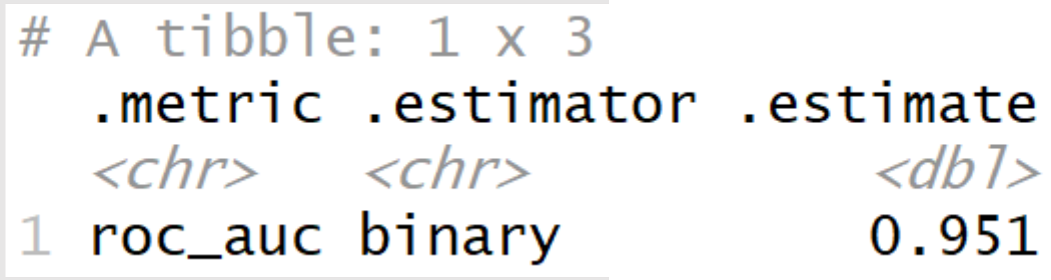
```
# ROC curve

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

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

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

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) +
  labs(title = "ROC curve for text classification using regularized regression",
    subtitle = "Predicting whether text was written by 김승옥 or 은희경")
```

ROC curve for text classification using regularized regression

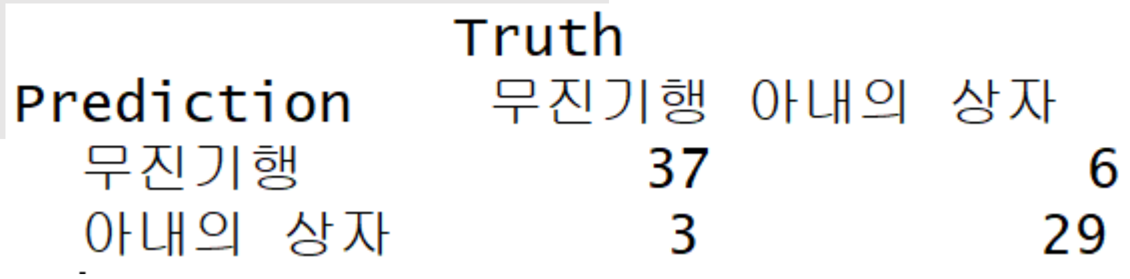
Predicting whether text was written by 김승옥 or 은희경



```r
# AUC & confusion matrix

comment_classes %>%
  roc_auc(title, probability)

comment_classes %>%
  mutate(
    prediction = case_when(probability > 0.5 ~ "아내의 상자",
                           TRUE ~ "무진기행"),
    prediction = as.factor(prediction)
  ) %>%
  conf_mat(title, prediction)
```

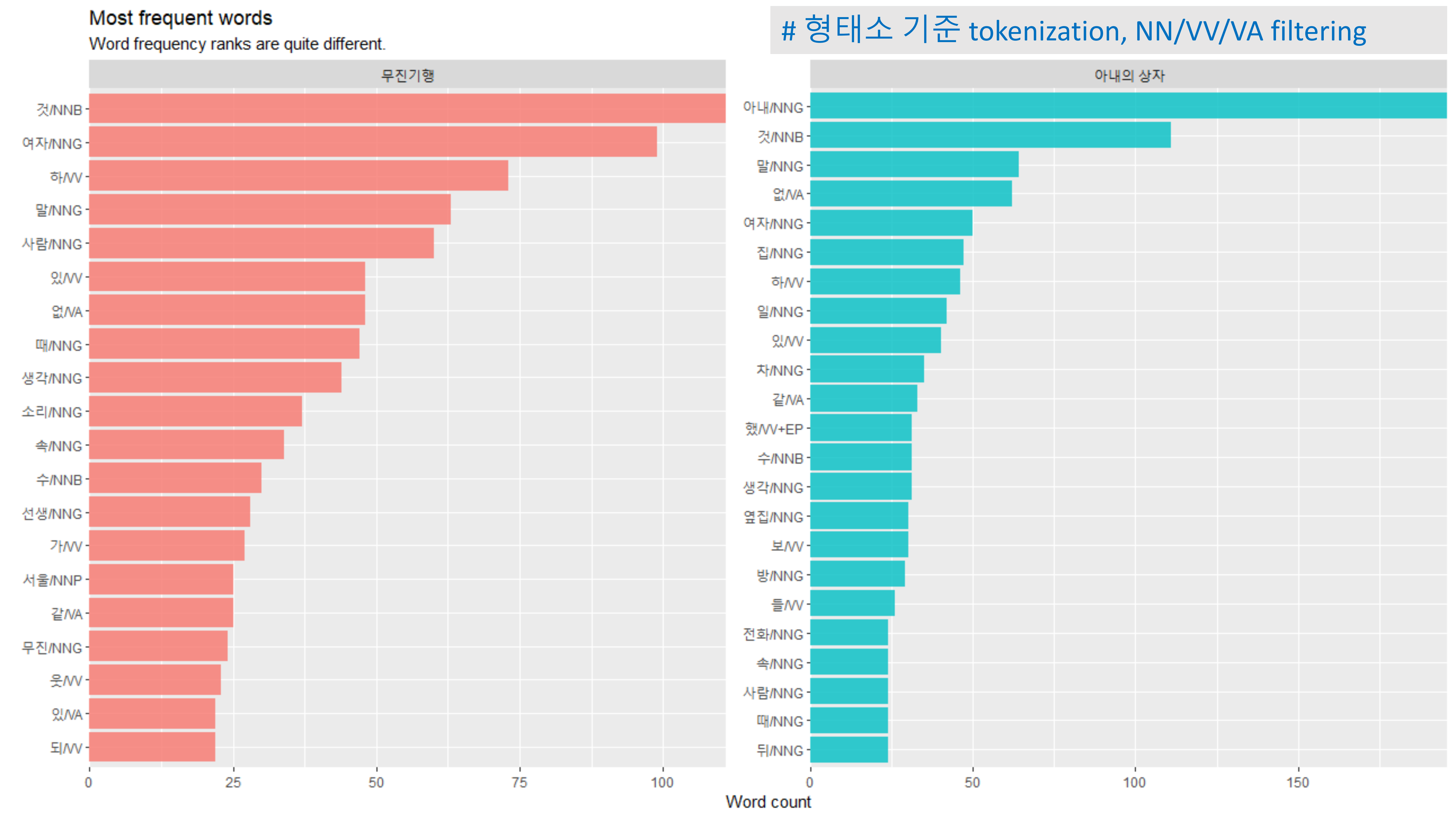


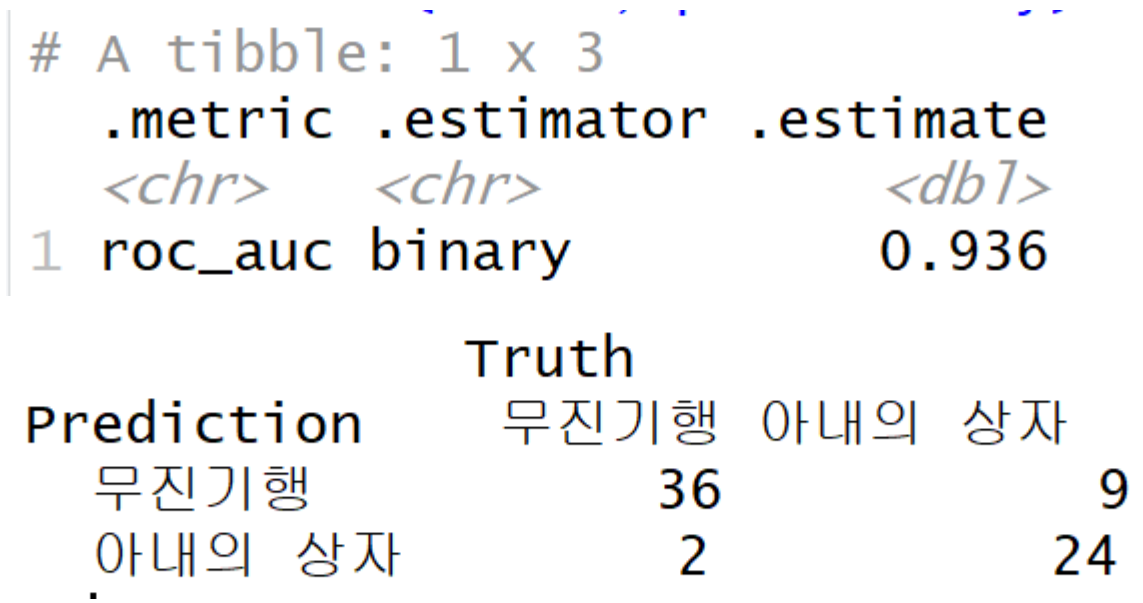
```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.951
```



```
             Truth
Prediction    무진기행 아내의 상자
  무진기행          37           6
  아내의 상자        3          29
```

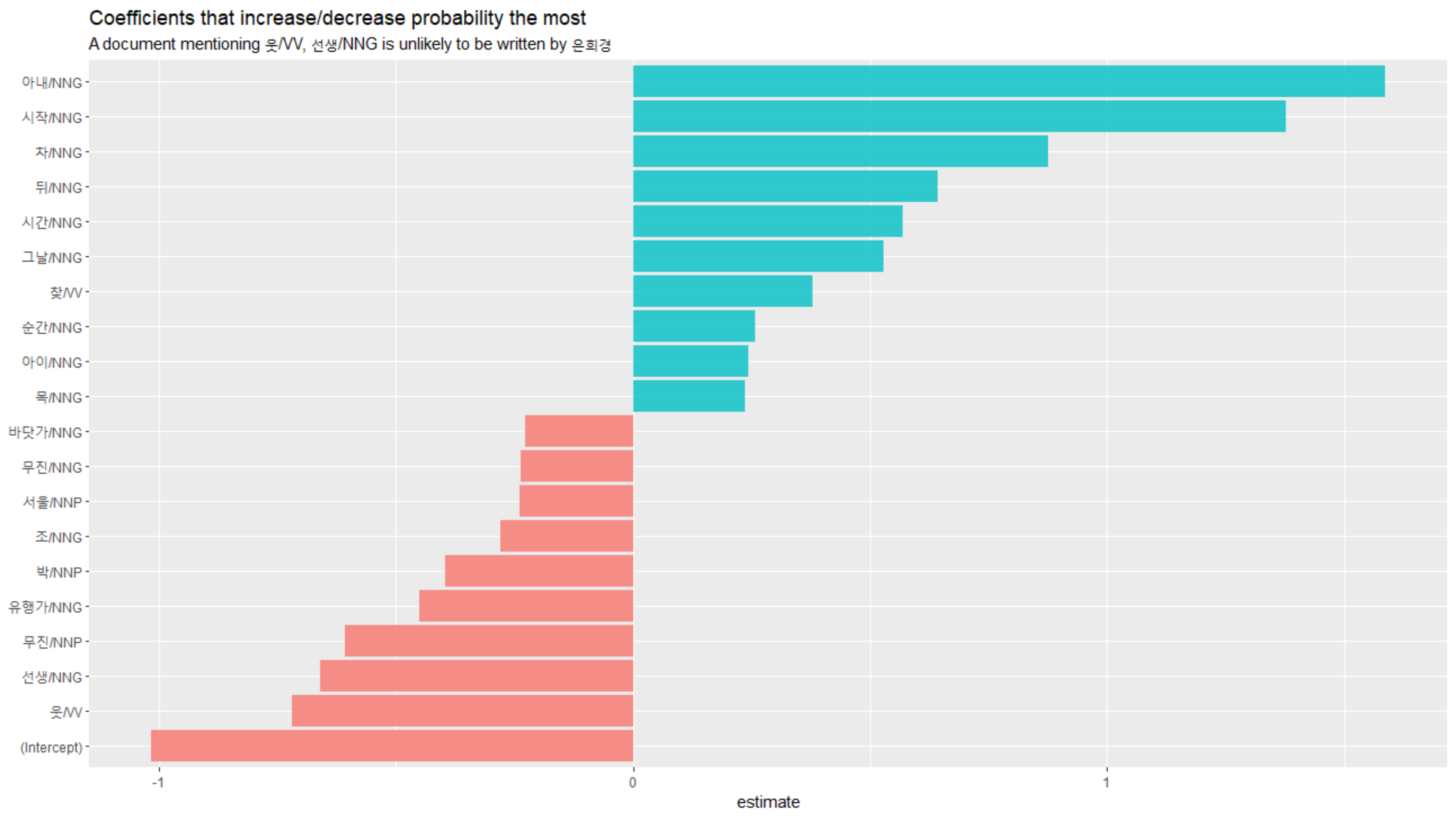
# 형태소 기준 tokenization, NN/VV/VA filtering





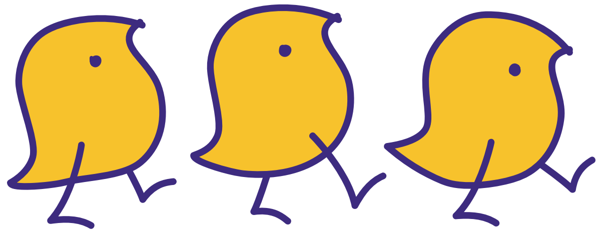
```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.936

             Truth
Prediction    무진기행 아내의 상자
  무진기행          36           9
  아내의 상자        2          24
```

# VV/VA (동사, 형용사) 추가가 예측력을 높이지는 않음


Coefficients that increase/decrease probability the most
A document mentioning 웃/VV, 선생/NNG is unlikely to be written by 은희경
아내/NNG
시작/NNG
차/NNG
뒤/NNG
시간/NNG
그날/NNG
찾/VV
순간/NNG
아이/NNG
목/NNG
바닷가/NNG
무진/NNG
서울/NNP
조/NNG
박/NNP
유행가/NNG
무진/NNP
선생/NNG
웃/VV
(Intercept)
-1
0
1
estimate

Gist

- 빈도 모형 기반으로 분석할 경우 특정 형태소만 남기는 게 효율적(많은 한글 텍스트에선 명사만 남기는 것이 결과가 좋지만, 동사, 형용사 등을 다양하게 확인할 필요)
- 따라서 불용어 사전 사용은 권하지 않음
- 그러나 딥 러닝(CNN/RNN)에 넣을 때는 형태소를 선별하면 오히려 예측력이나 분류 결과가 나빠지므로 주의



Q&A