RcppMeCab?

김준혁

자기소개

치과의사

작가

번역가

의료윤리, 의료인문학 연구자

+ 개발자?

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]

[1] MeCab. *Wikipedia, the free encyclopedia*. <https://en.wikipedia.org/wiki/MeCab>
[2] Taku Kudou. *MeCab*. <http://taku910.github.io/mecab/>

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...

텍스트 분석을 막 익히던 2013-14년에는 Python의 KoNLPy와 R의 KoNLP가 대표적인 형태소 분석 패키지

"한글 분석은 당연히 Python에서 해야 하는 것 아냐?"

R의 장점

- 자료 구조가 직관적이다.

R의 단점

- Encoding

그렇다면…

문제를 해결해보자. 어떻게? 패키지 제작

RmecabKo (First release in Github & 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

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”

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/>

한글 토큰

띄어쓰기 기준?

형태소 기준?

형태소+품사 기준?

RcppMeCab

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
  - Input encoding 강제여부 (with enc2utf8)?
  - Output 양식 추가?
- RmecabKo 개선
  - RcppMeCab을 import한 후 추가 기능을 넣는 방식으로 바꿀 예정
  - 추가기능: 사용자 사전, mecab-ko & mecab-ko-dic 설치 함수, stopwords (?), 형태소별 추출, n-gram, sentiment

예제: Text Classification with Tidy Data Principles [1]

```
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/>

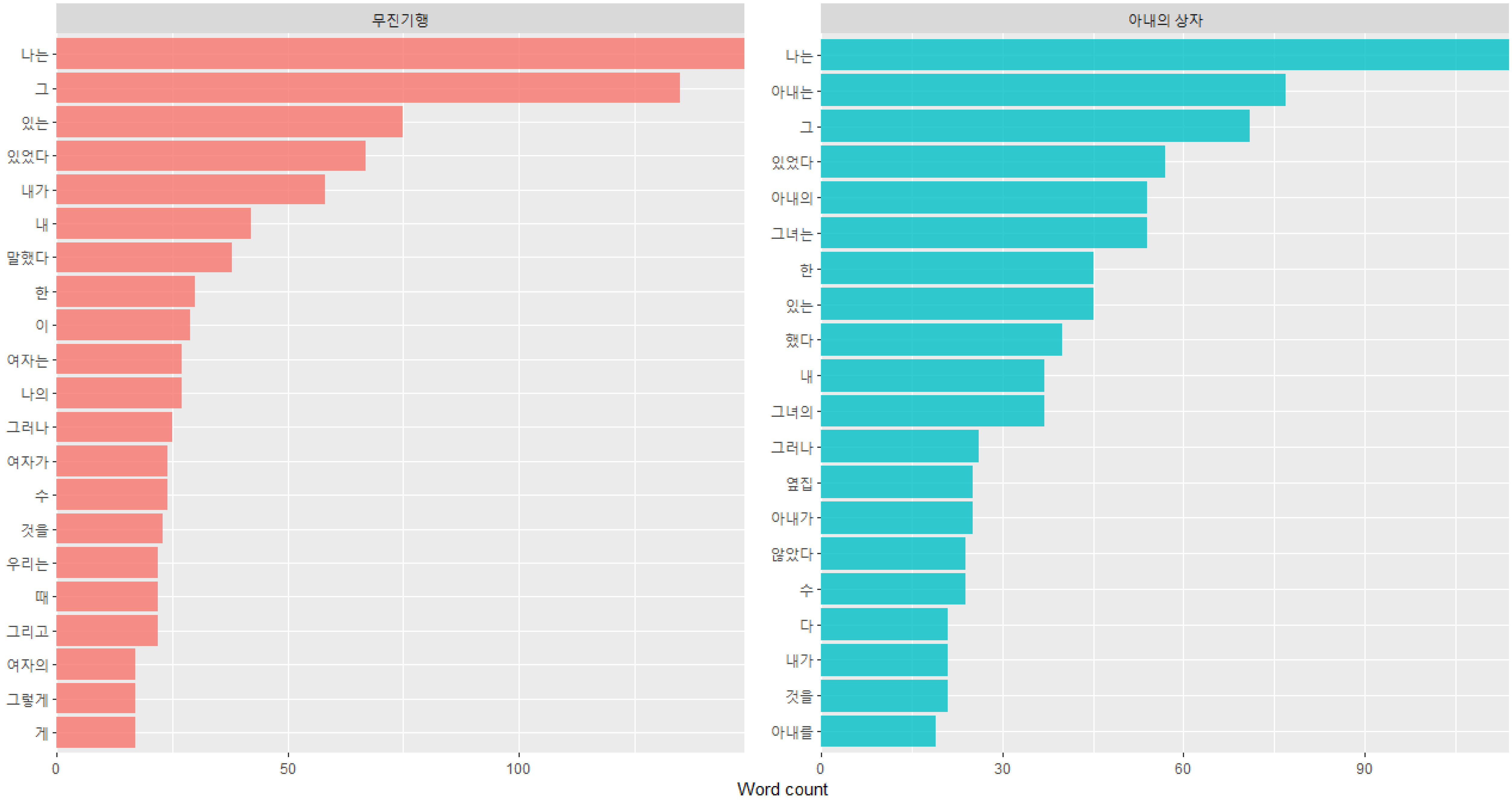
```
# 무진기행(김승옥), 아내의 상자(은희경) 비교

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

```r
# 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.")
```

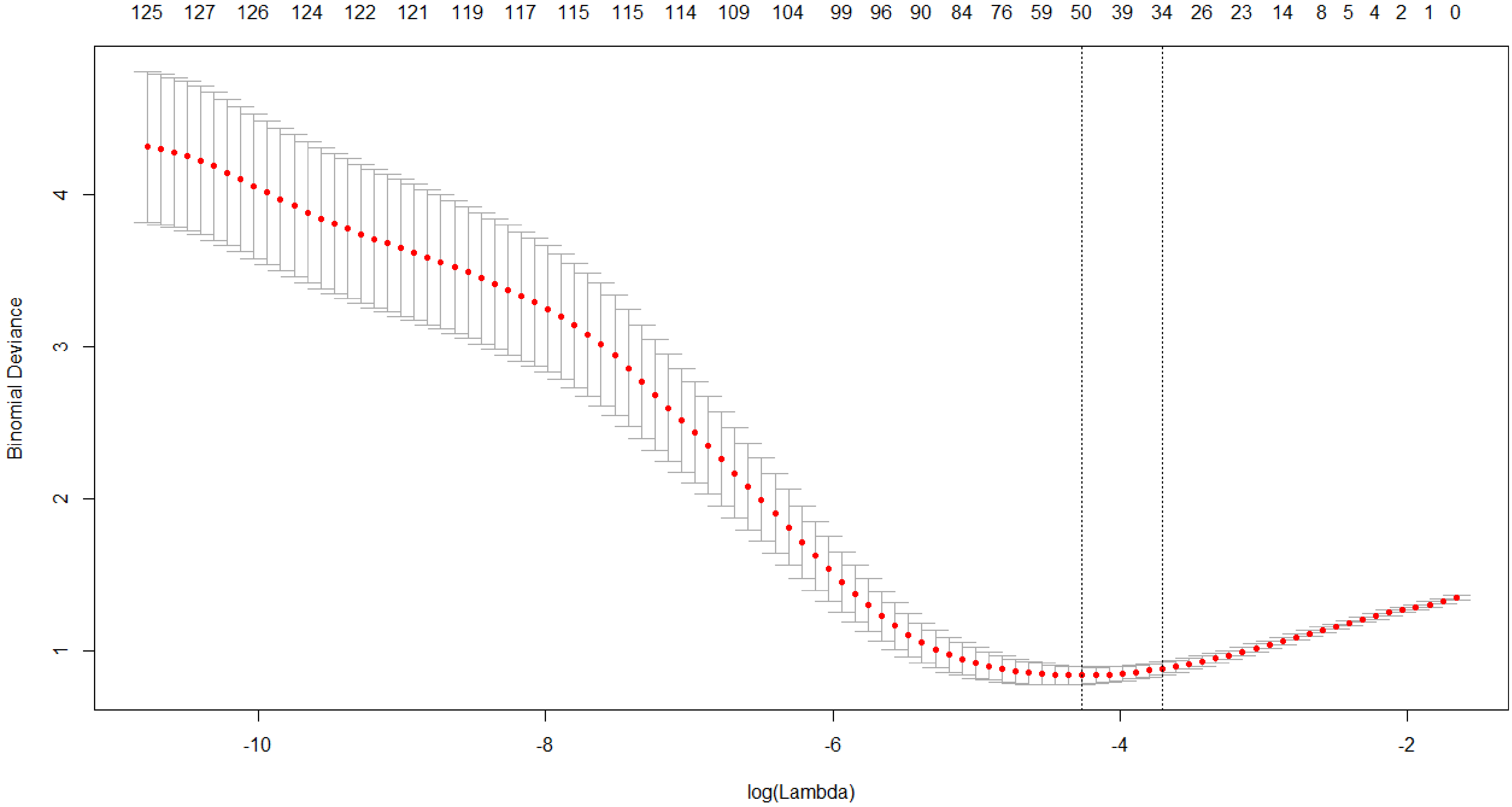
Most frequent word

Word frequencies are hard to understand; '여자는' vs. '아내는' is the main difference.



```
# 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))
```

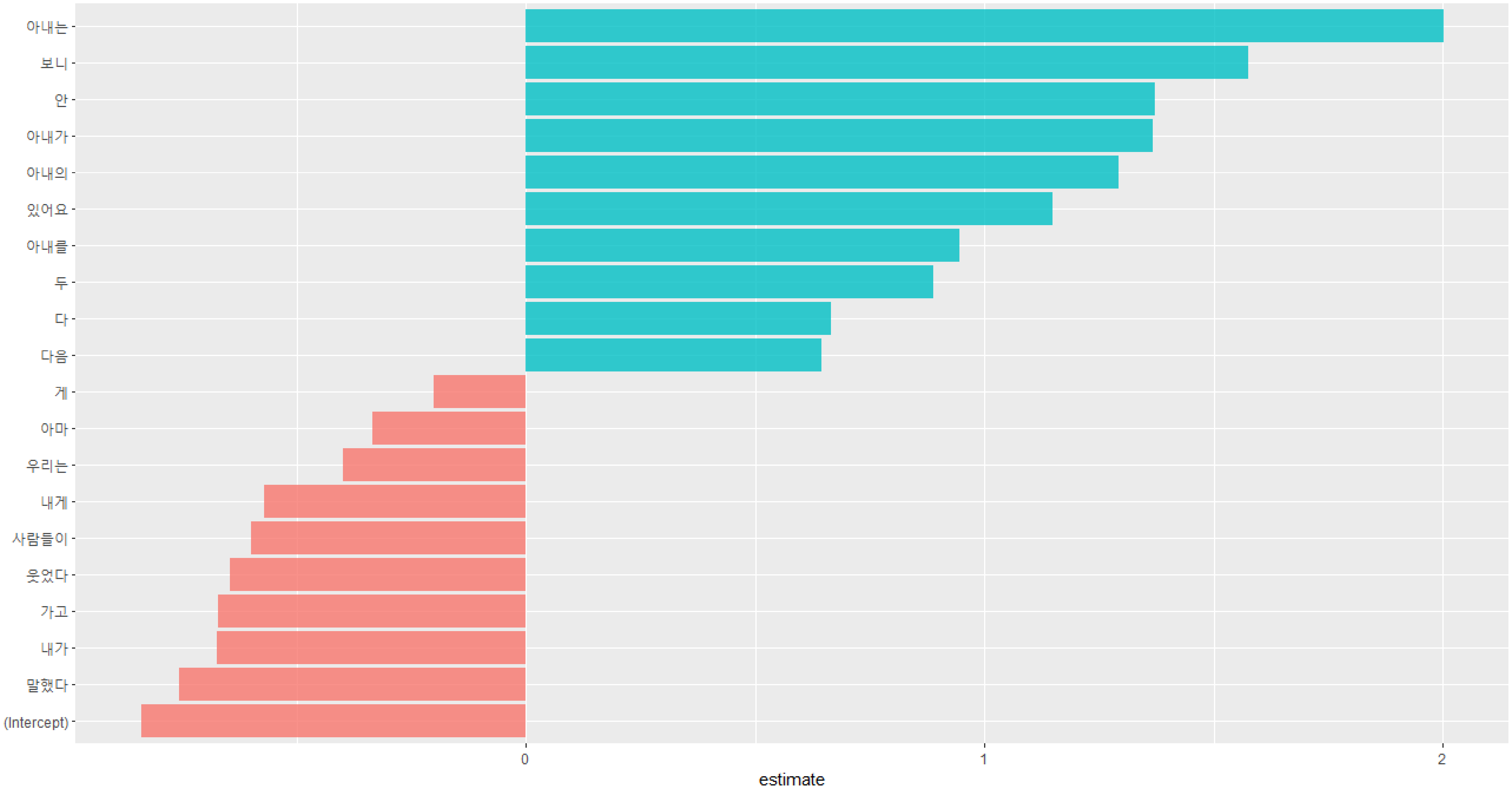
```
# glmnet training

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)
```


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)

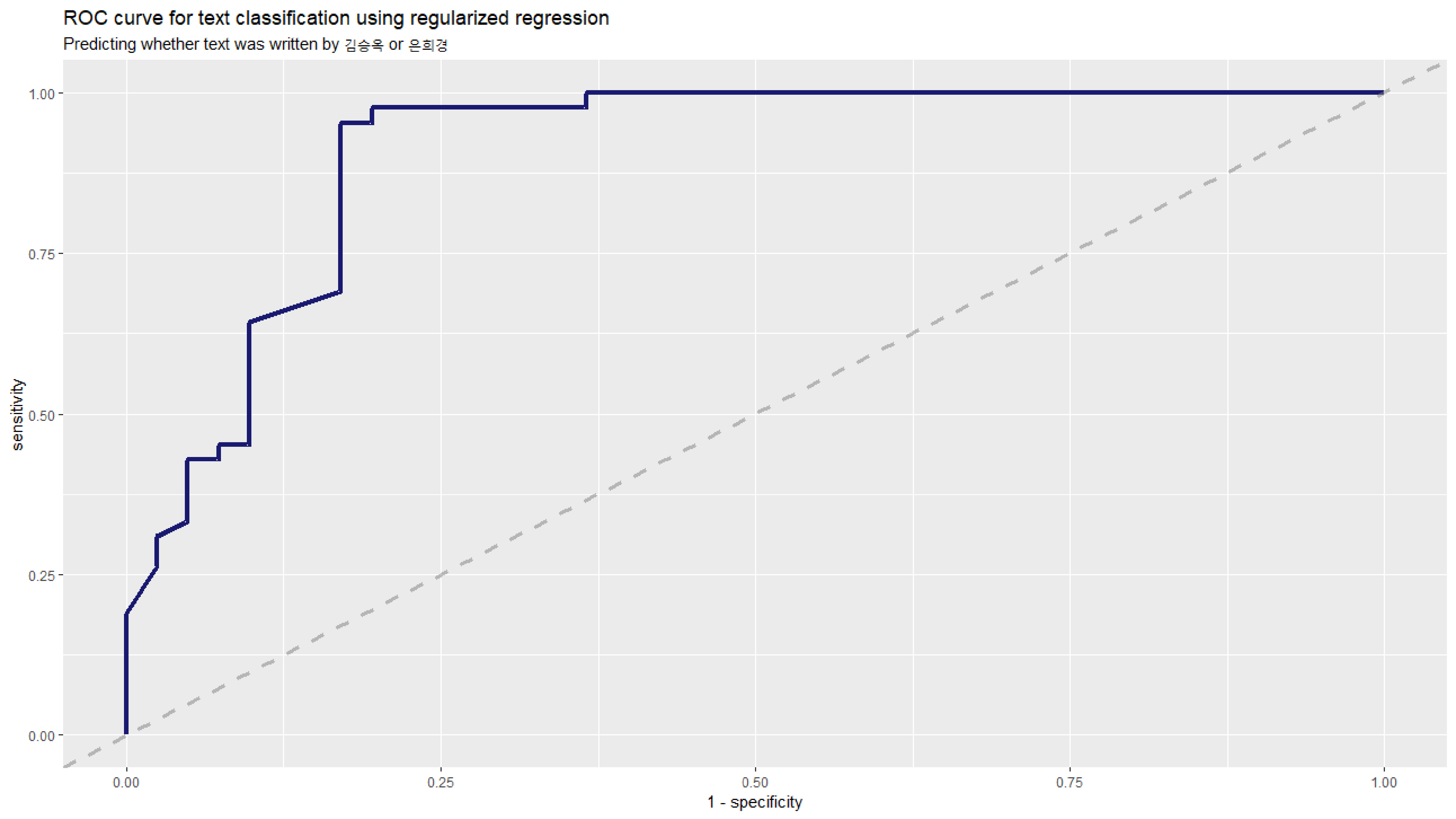
```
# word probabilities

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

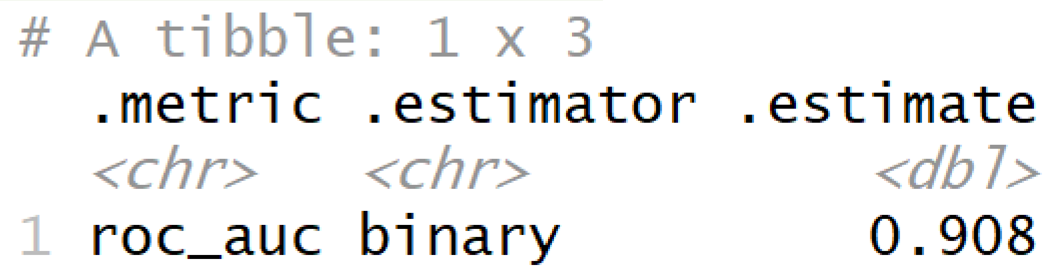
A document mentioning 말했다 is unlikely to be written by 은희경



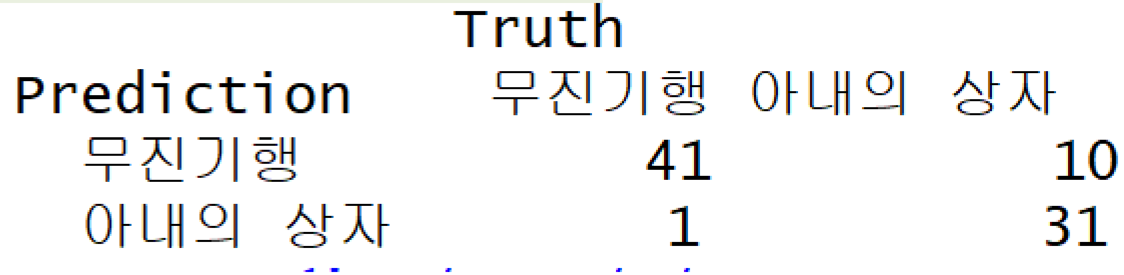
```r
# ROC curve

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
```

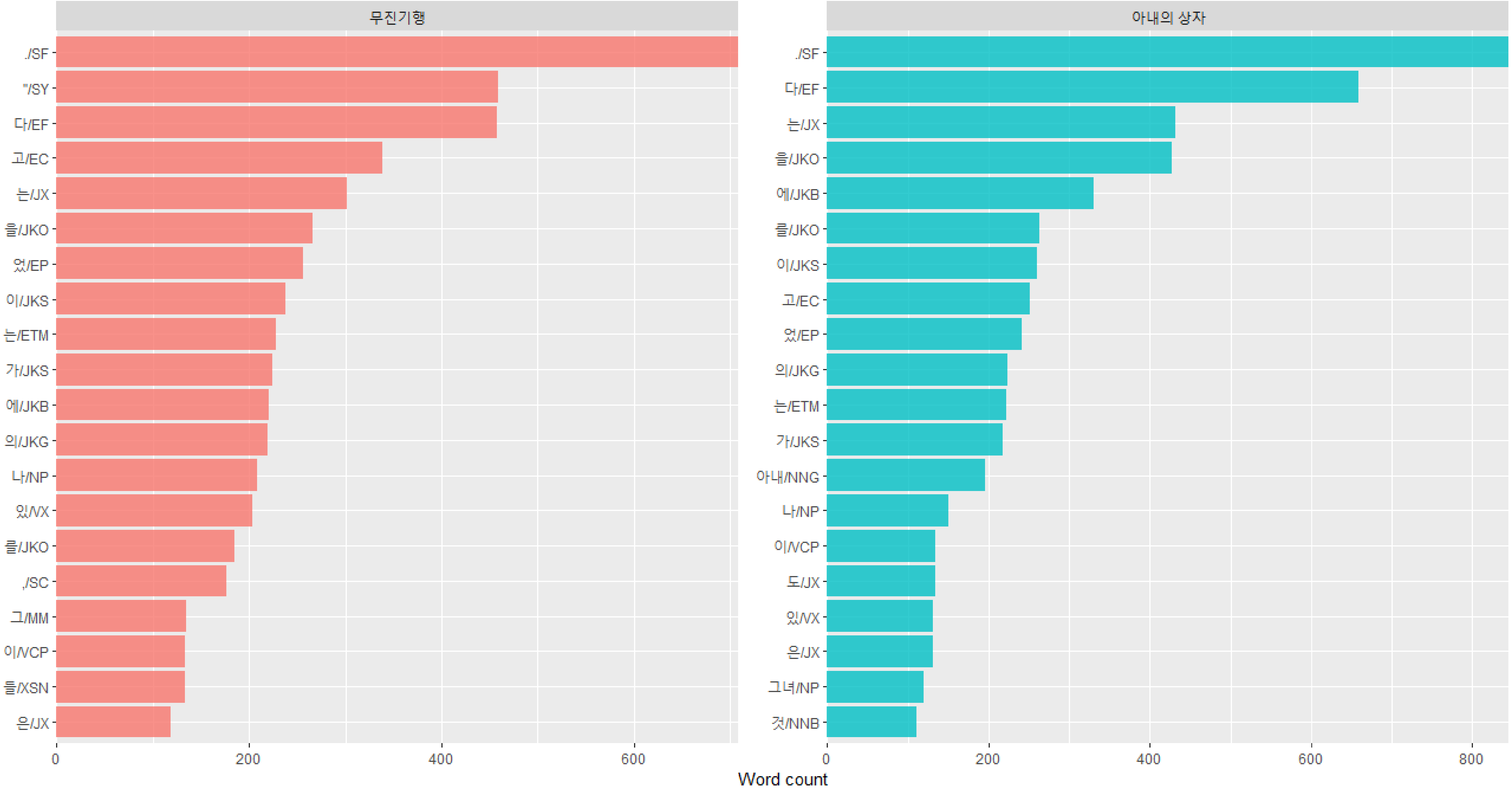


```
             Truth
Prediction    무진기행 아내의 상자
  무진기행          41          10
  아내의 상자        1          31
```

```
# 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.")
```

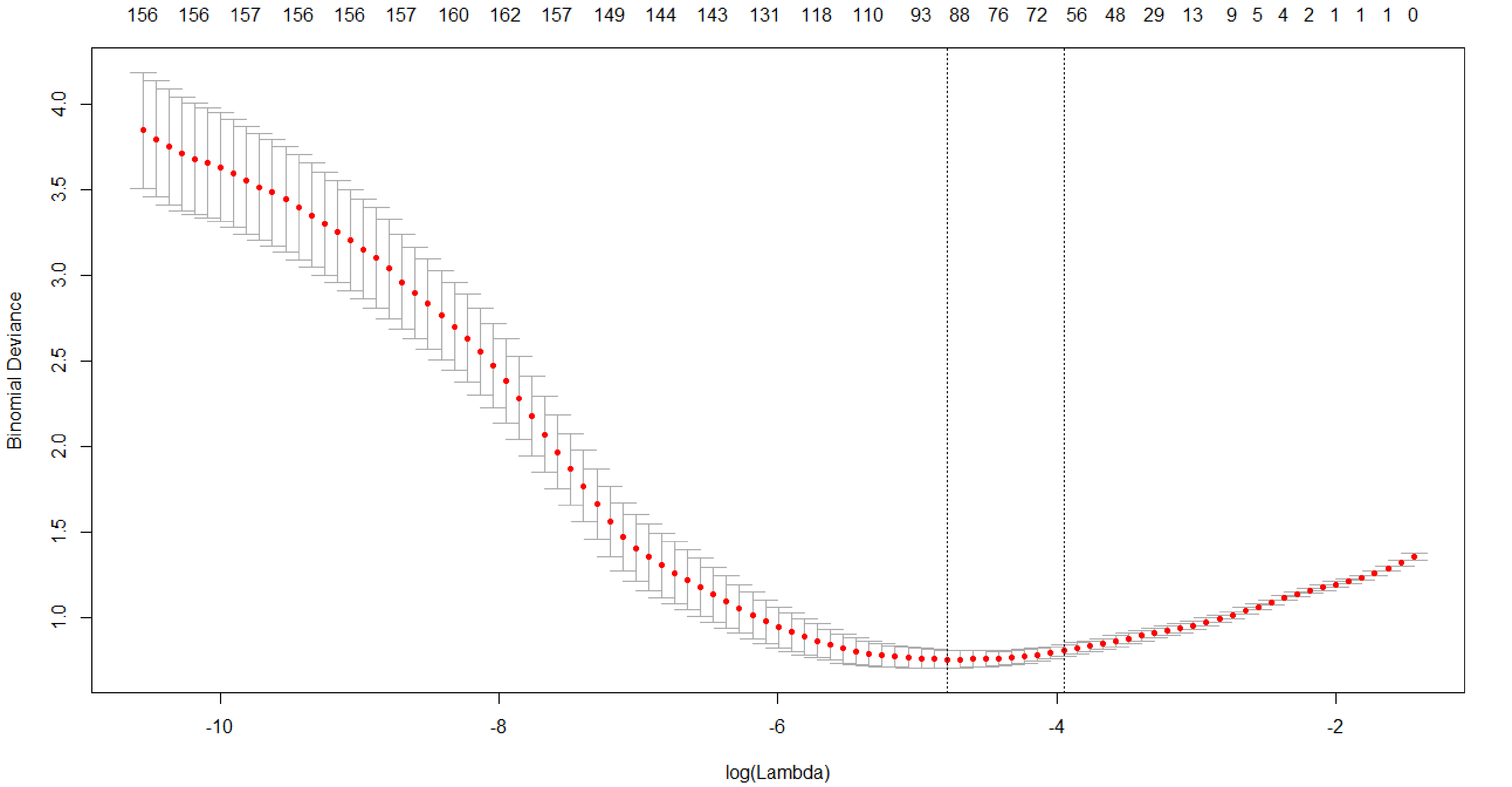
Most frequent words

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))
```

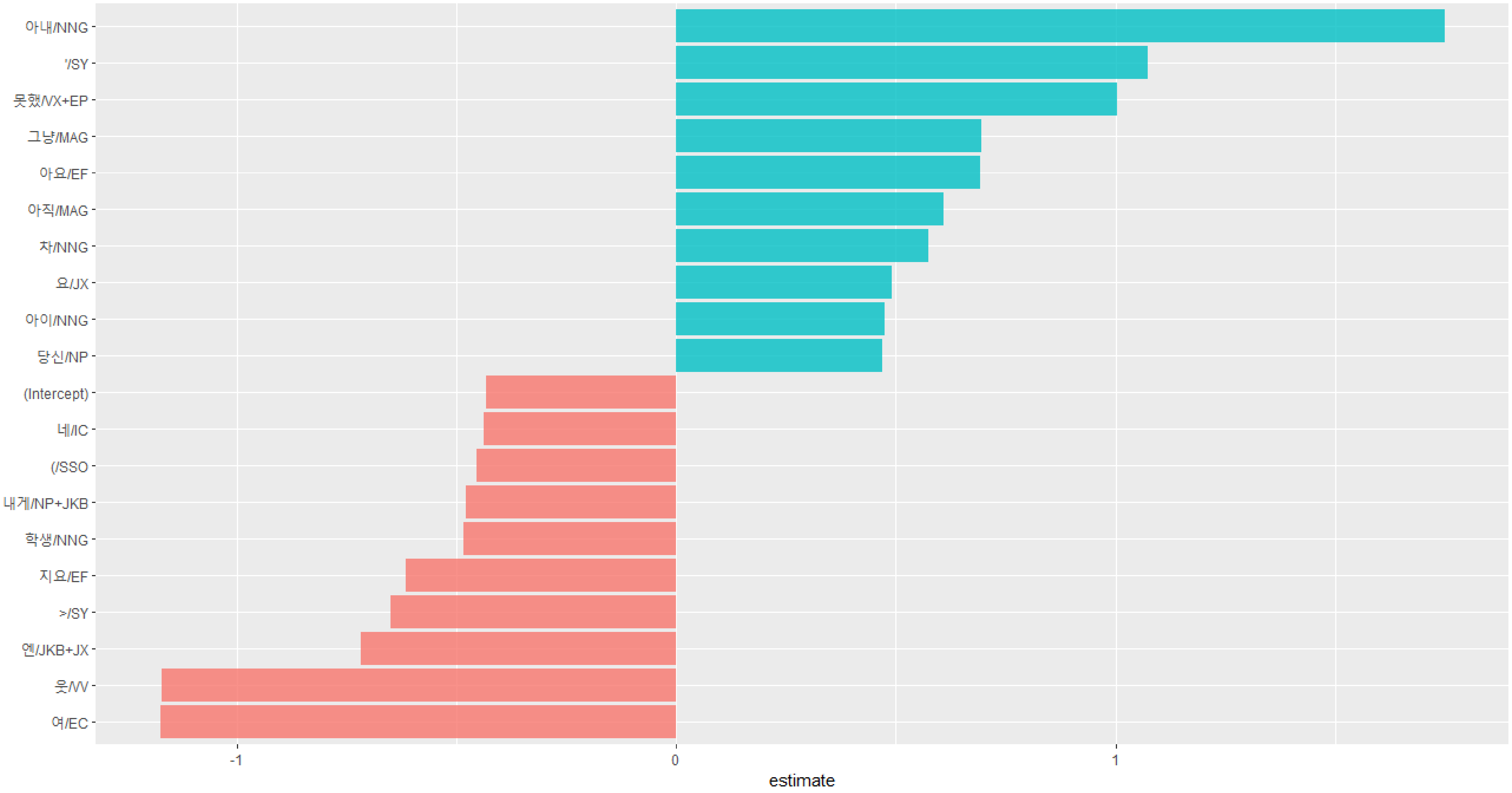
```
# glmnet training

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
4.0
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

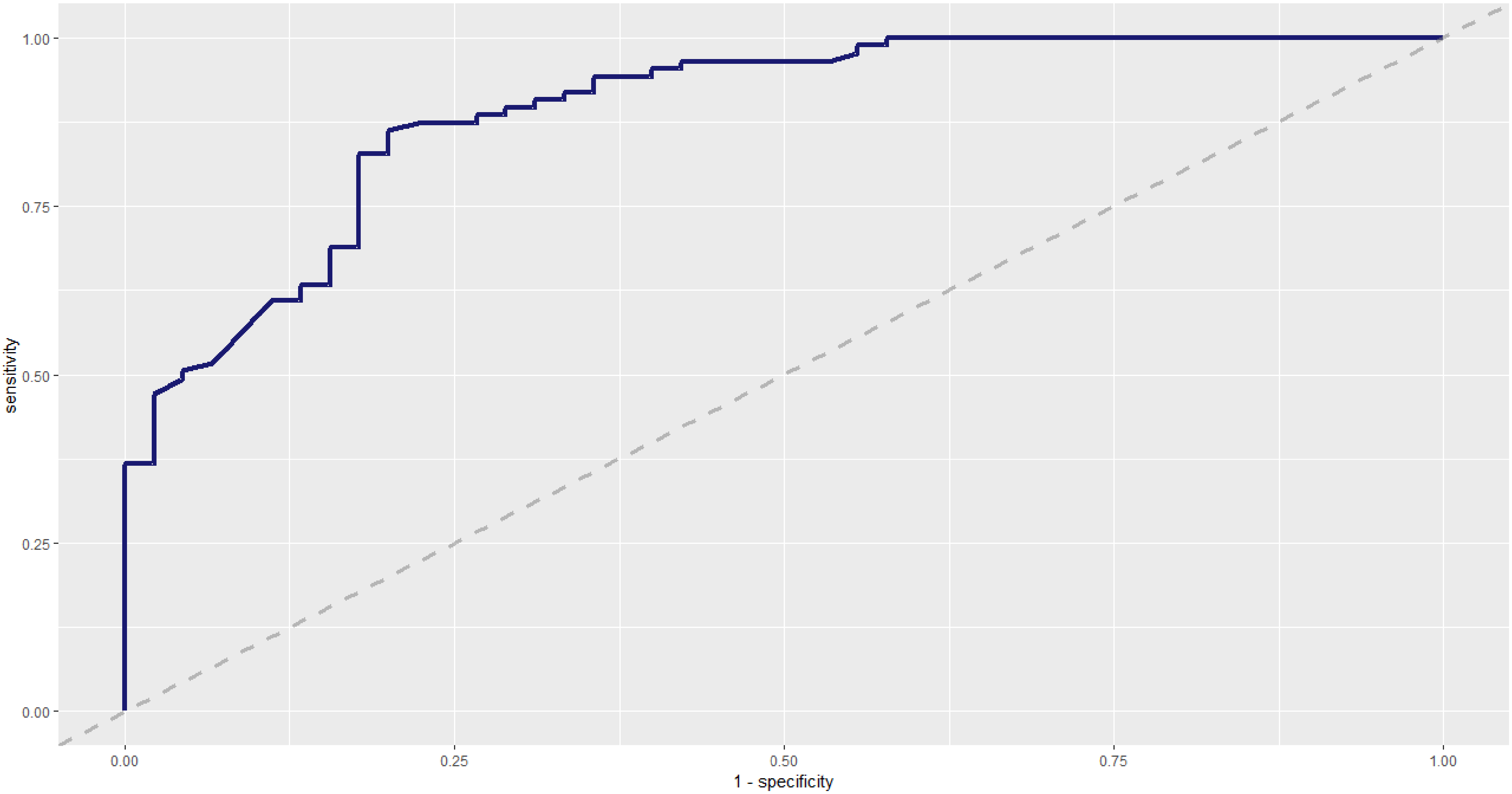
A document mentioning 여/EC, 웃/VV is unlikely to be written by 은희경



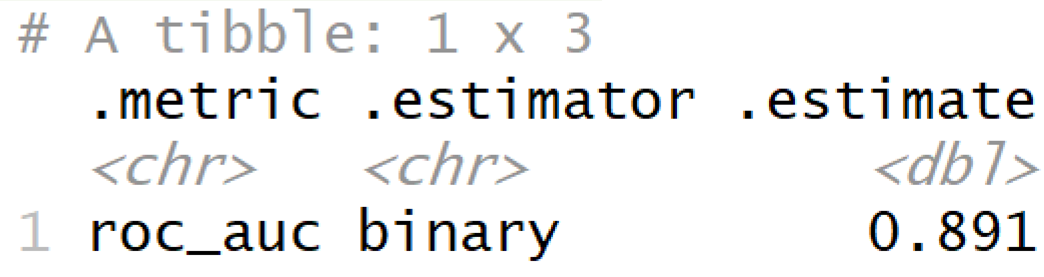
```
# ROC curve

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

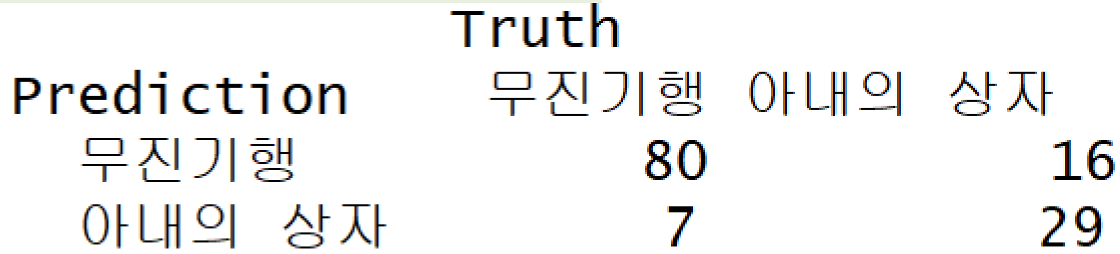
Predicting whether text was written by 김승옥 or 은희경



```
# 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
```

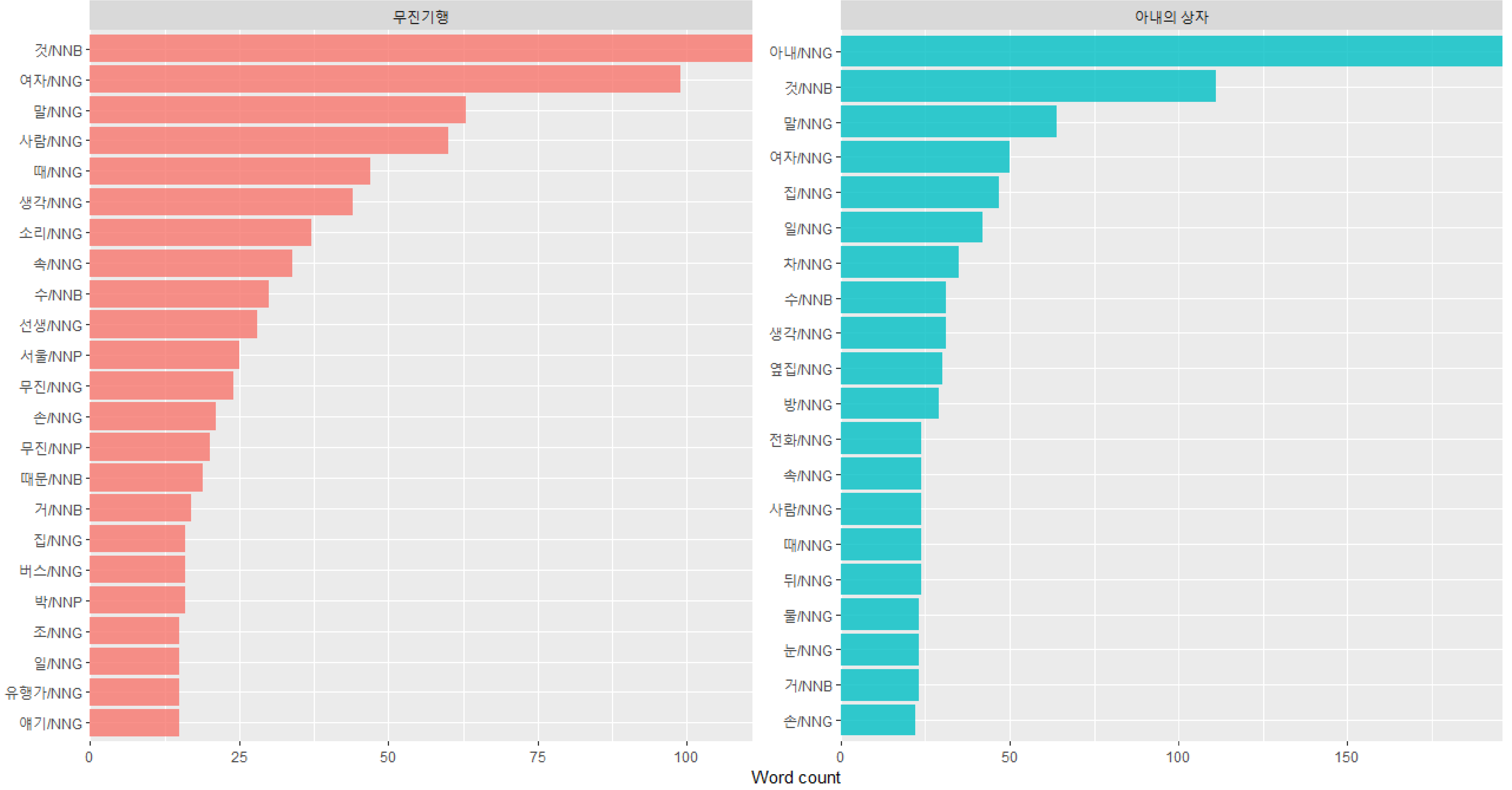


```
             Truth
Prediction    무진기행 아내의 상자
  무진기행          80          16
  아내의 상자        7          29
```

```
# 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.")
```

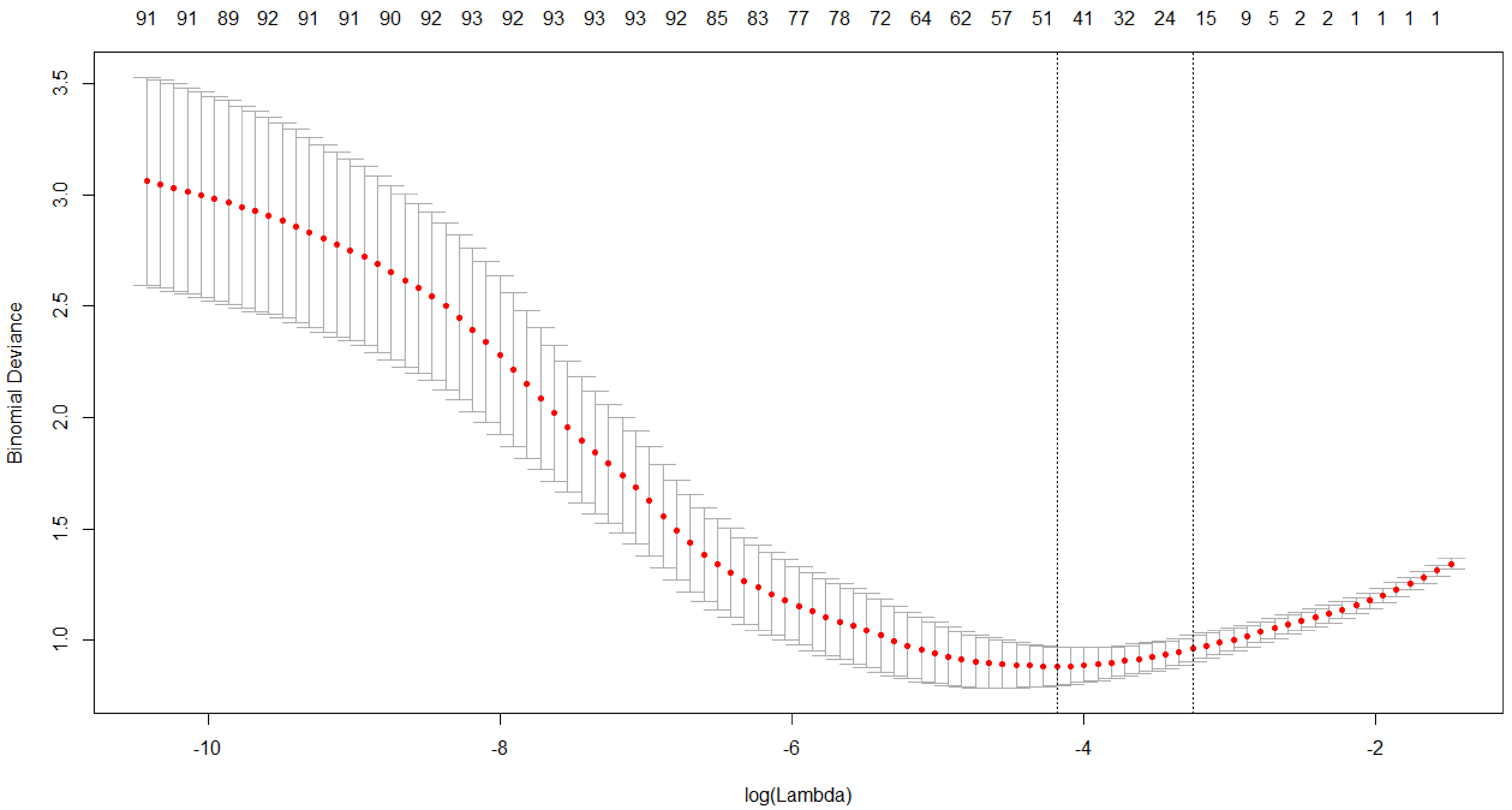
Most frequent words

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

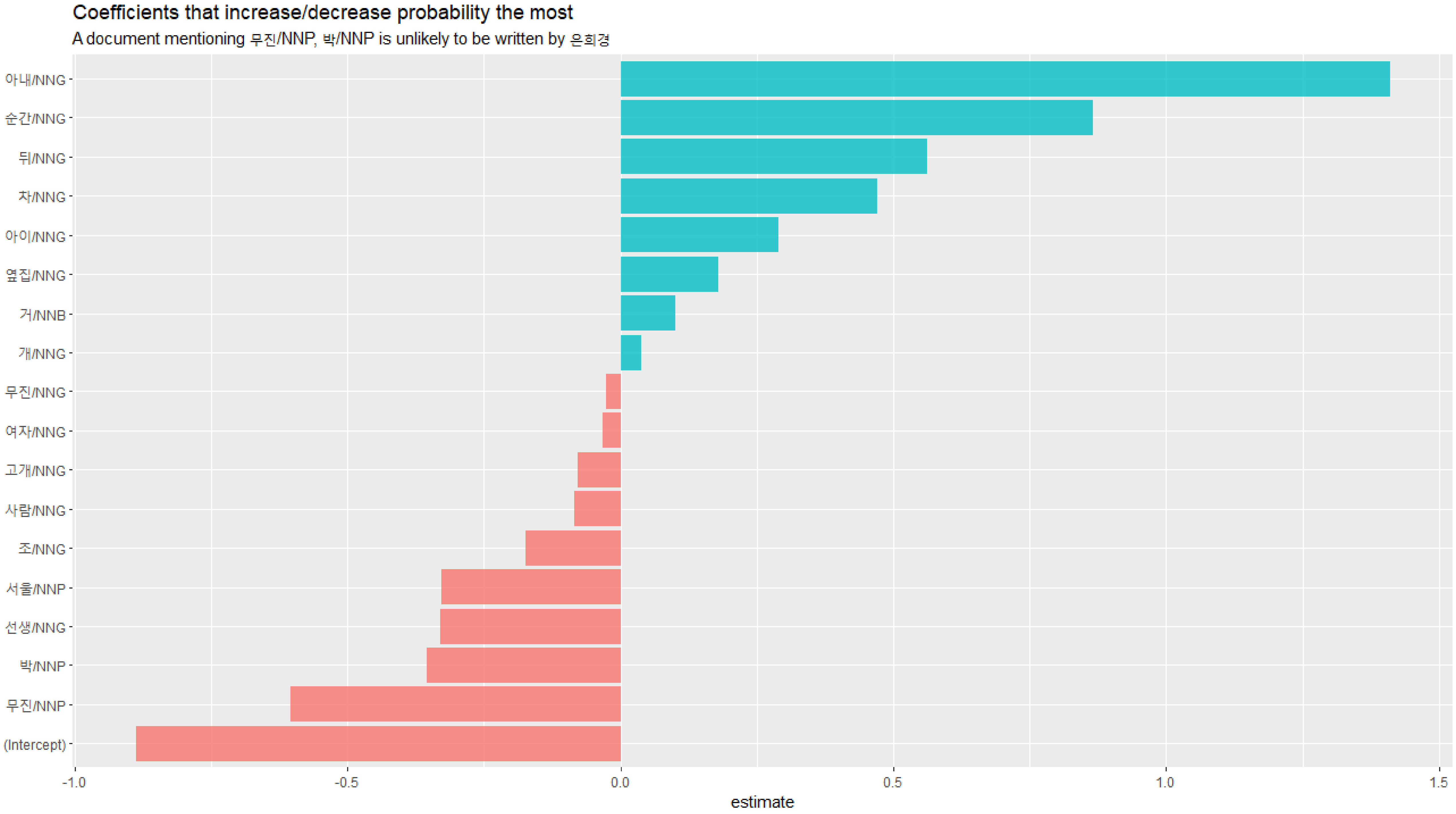


```
# glmnet training을 위한 training/test set separation

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)

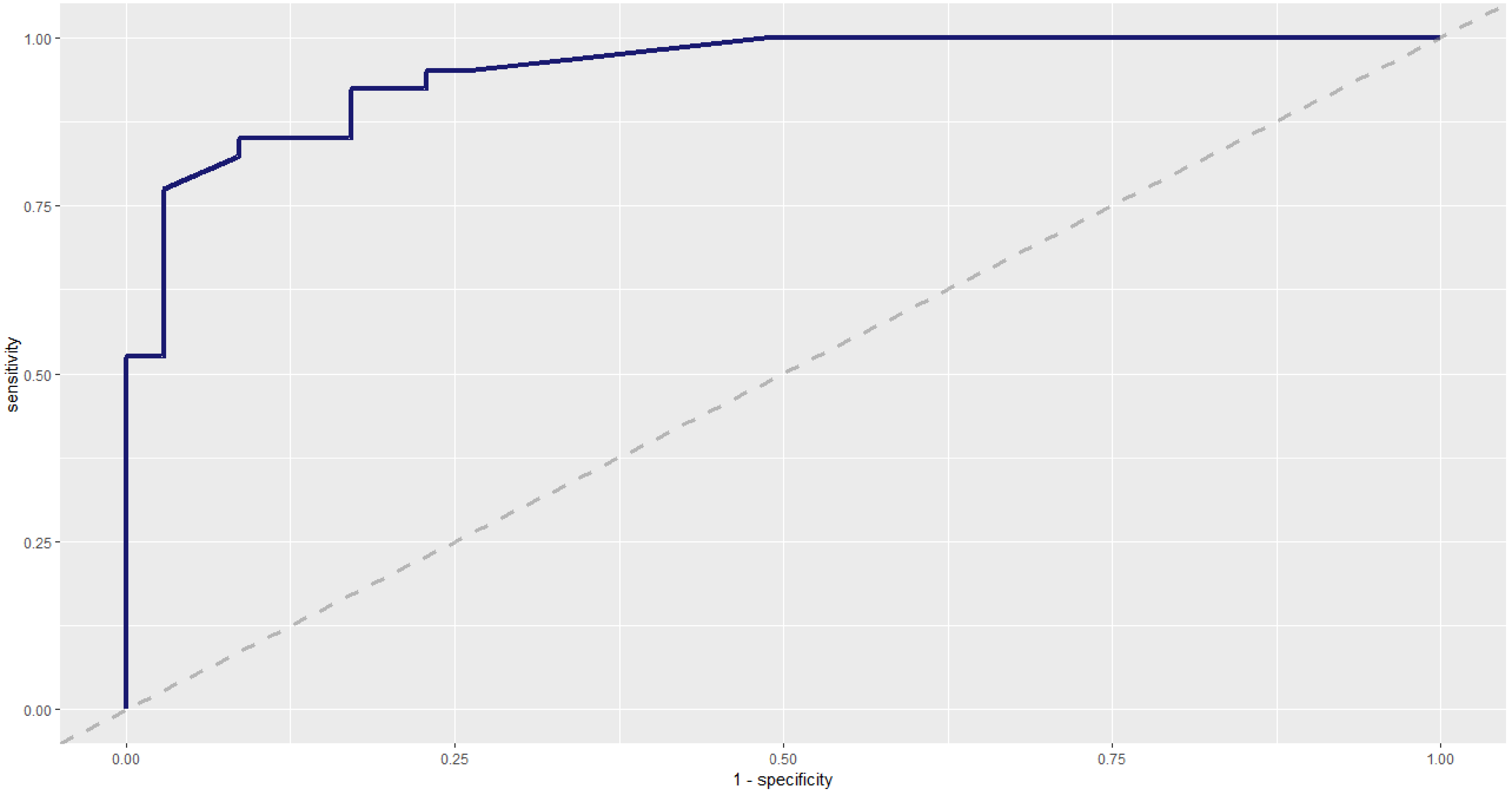
```
# word probabilities

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

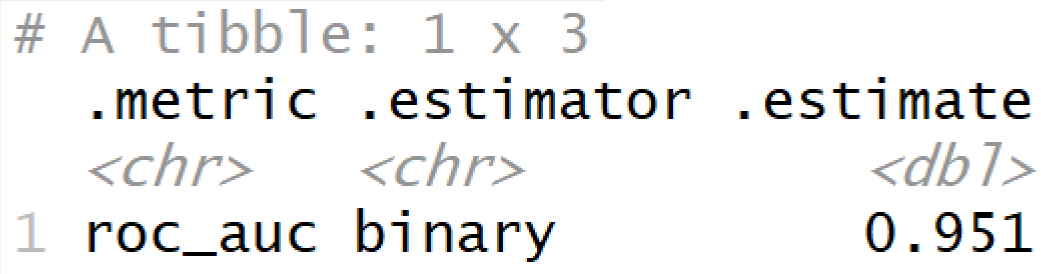
```
# ROC curve

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")
```



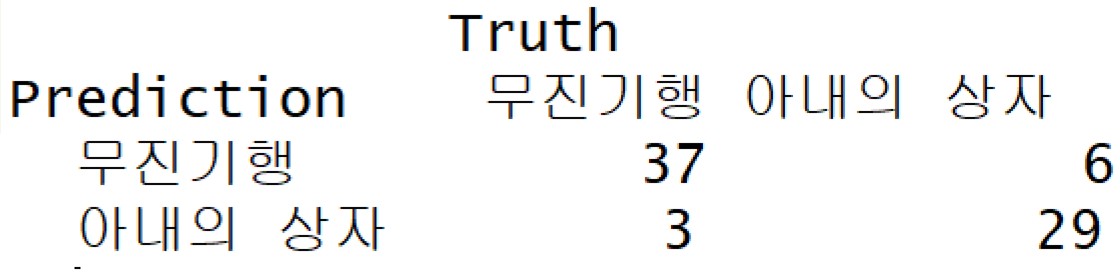
ROC curve for text classification using regularized regression

Predicting whether text was written by 김승옥 or 은희경

```
# 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)
```

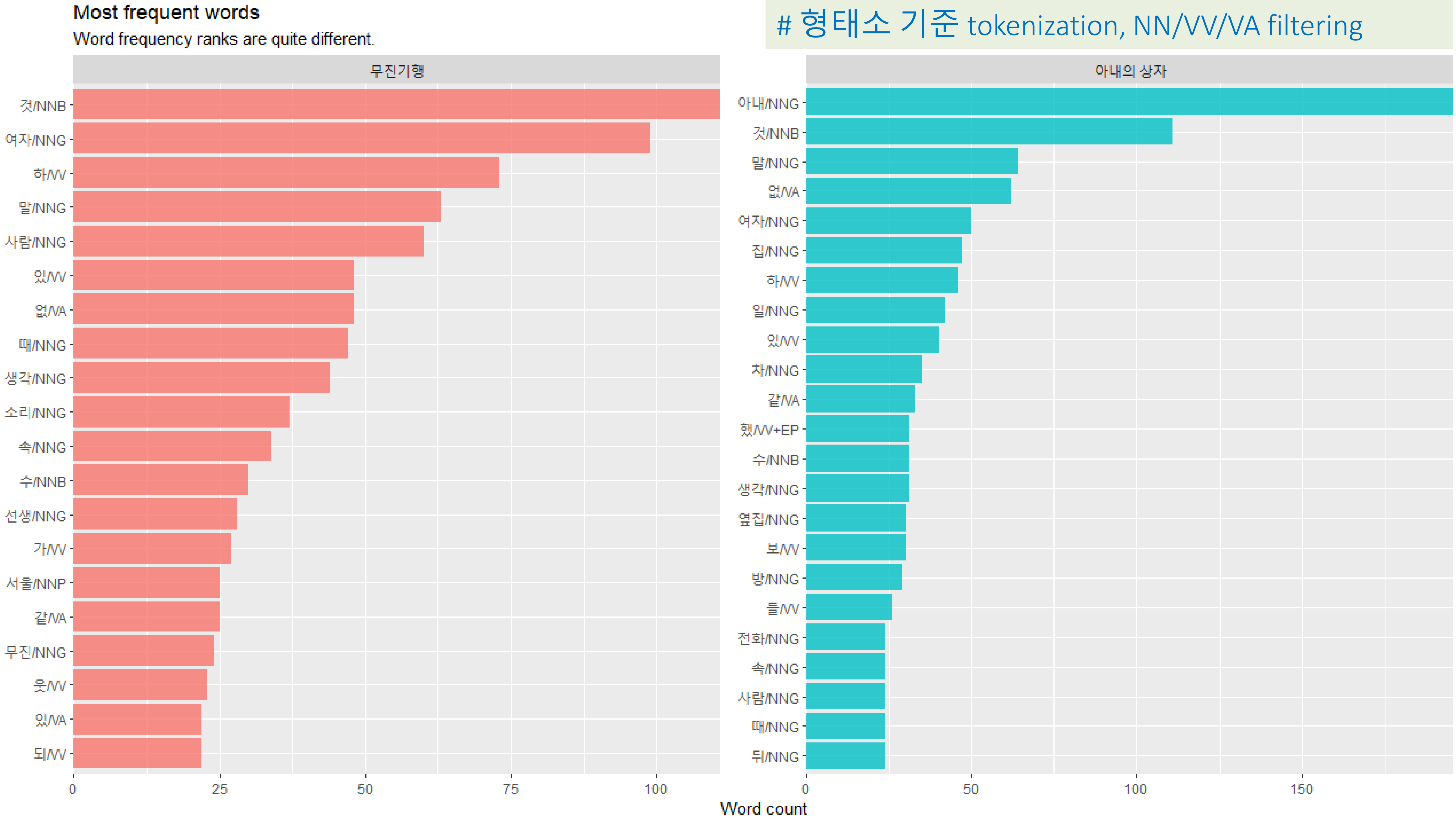


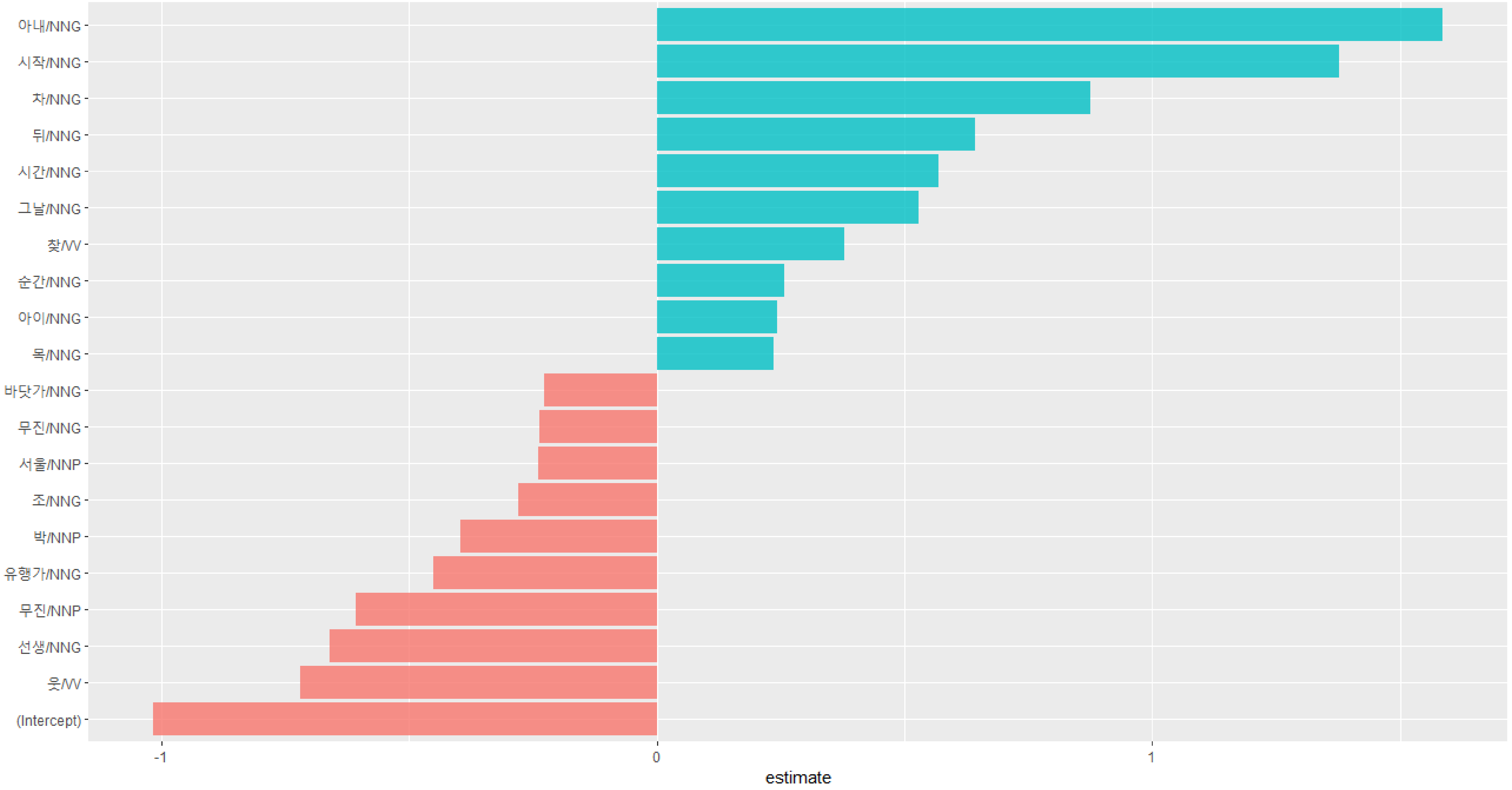
```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.951
```



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

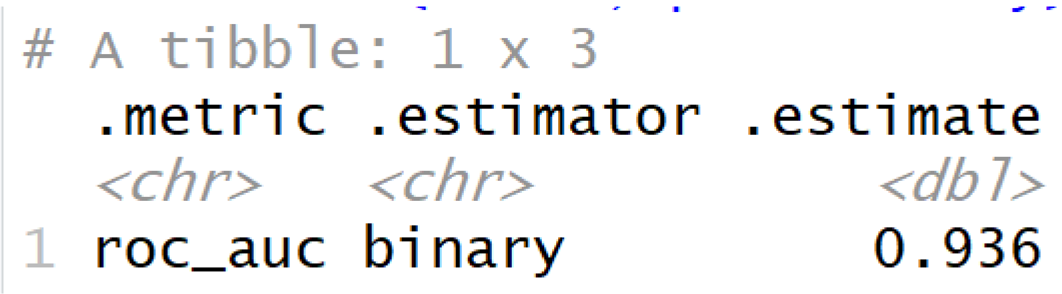
# 형태소 기준 tokenization, NN/VV/VA filtering



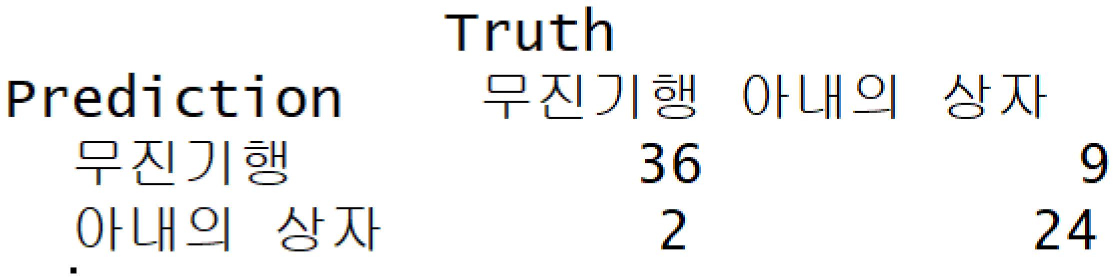


Coefficients that increase/decrease probability the most

A document mentioning 웃/VV, 선생/NNG is unlikely to be written by 은희경



```
# 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

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)

그러나 딥 러닝(CNN/RNN)에 넣을 때는 형태소를 선별하면 오히려 예측력이나 분류 결과가 나빠지므로 주의

감사합니다