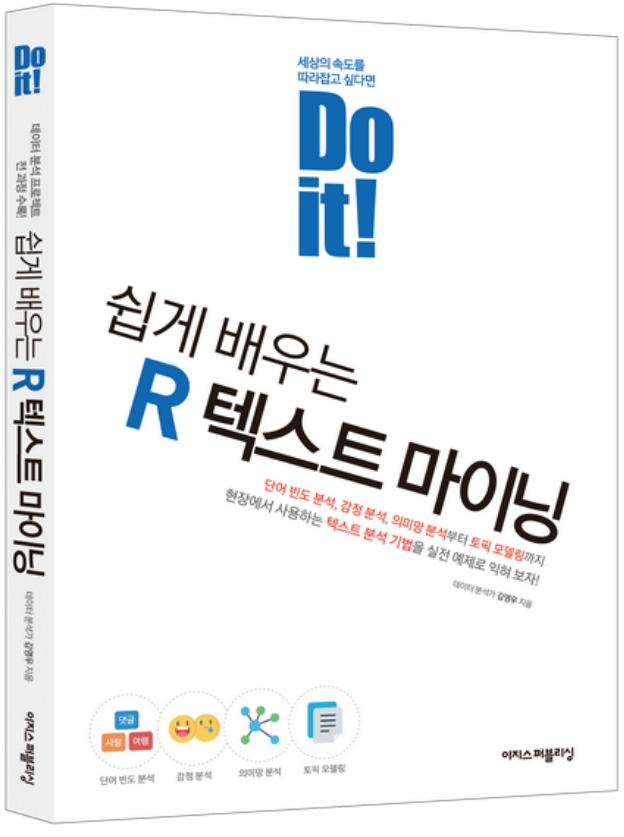
텍스트를 쉽게 분석하는 방법



github.com/youngwoos/Doit_textmining

facebook.com/groups/datacommunity

- 네이버책
  - yes24
  - 알라딘
  - 교보문고

Part 01 단어 빈도 분석: 무엇을 강조했을까?

01-1 토큰화하기(link)

01-2 단어 빈도 분석하기(link)

Part 02 비교 분석: 무엇이 다를까?

02-1 단어 빈도 비교하기(link)

02-2 오즈비 - 상대적으로 중요한 단어 비교하기(link)

02-3 TF-IDF - 여러 텍스트의 단어 비교하기(link)

Part 03 의미망 분석: 어떤 맥락에서 단어를 썼을까?

03-1 동시 출현 단어 분석: Co-occurrence analysis(link)

03-2 동시 출현 네트워크: Co-occurrence network(link)

03-3 단어 간 상관 분석: Phi coefficient(link)

03-4 연이어 사용된 단어쌍 분석: n-gram(link)

This slide

youngwoos.github.io/rmeetup_tidy_textmining/tidy_textmining.html

Full Version

github.com/youngwoos/Doit_textmining/tree/main/docs

01 단어 빈도 분석:
무엇을 강조했을까?

01-1 토큰화하기

토큰화(tokenization)

- 토큰(token): 텍스트의 기본 단위(ex: 단락, 문장, 단어, 형태소)
- 토큰화: 텍스트를 토큰으로 나누는 작업

`tidytext` 패키지

- 텍스트를 정돈된 데이터(Tidy Data) 형태를 유지하며 분석
- `dplyr`, `ggplot2` 패키지와 함께 활용
- 토큰화하기: `unnest_tokens()`

샘플 텍스트로 작동 원리 알아보기

```
library(dplyr)
text <- tibble(value = "대한민국은 민주공화국이다. 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.")

text
```

```
## # A tibble: 1 x 1
##   value
##   <chr>
## 1 대한민국은 민주공화국이다. 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.…
```

```r
install.packages("tidytext")
library(tidytext)

# 문장 기준 토큰화
text %>%
  unnest_tokens(input = value,         # 토큰화할 텍스트
                output = word,         # 토큰을 담을 변수명
                token = "sentences")  # 문장 기준
```

```
## # A tibble: 2 x 1
##   word
##   <chr>
## 1 대한민국은 민주공화국이다.
## 2 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
```

```r
# 띄어쓰기 기준 토큰화
text %>%
  unnest_tokens(input = value,
                output = word,
                token = "words")      # 띄어쓰기 기준
```

```
## # A tibble: 10 x 1
##    word
##    <chr>
##  1 대한민국은
##  2 민주공화국이다
##  3 대한민국의
##  4 주권은
##  5 국민에게
##  6 있고
##  7 모든
##  8 권력은
##  9 국민으로부터
## 10 나온다
```

```r
# 문자 기준 토큰화
text %>%
  unnest_tokens(input = value,
                output = word,
                token = "characters")  # 문자 기준
```

```
## # A tibble: 40 x 1
##    word
##    <chr>
##  1 대
##  2 한
##  3 민
##  4 국
##  5 은
##  6 민
##  7 주
##  8 공
##  9 화
## 10 국
## 11 이
## 12 다
## 13 대
```

형태소 추출

```
library(KoNLP)
extractNoun(text$value)
```

```
## [1] "대한민국"   "민주공화국" "대한민국"   "주권"       "국민"
## [6] "권력"       "국민"
```

```
text <- text %>%
  unnest_tokens(input = value,
                output = word,
                token = "sentences")

text
```

```
## # A tibble: 2 x 1
##   word
##   <chr>
## 1 대한민국은 민주공화국이다.
## 2 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
```

```r
# 형태소 추출
text %>%
  unnest_tokens(input = word,
                output = word,
                token = extractNoun) # 명사 기준
```

```
## # A tibble: 7 x 1
##   word
##   <chr>
## 1 대한민국
## 2 민주공화국
## 3 대한민국
## 4 주권
## 5 국민
## 6 권력
## 7 국민
```

01-2 단어 빈도 분석하기

연설문에서 명사 추출하기

문재인 대통령 연설문 불러오기

```
raw_moon <- readLines("speech_moon.txt", encoding = "UTF-8")
```

기본적인 전처리

```
library(stringr)

moon <- raw_moon %>%
  str_replace_all("[^가-힣]", " ") %>%  # 한글만 남기기
  str_squish() %>%                      # 중복 공백 제거
  as_tibble()                           # tibble로 변환

moon
```

```
## # A tibble: 117 x 1
##    value
##    <chr>
##  1 "정권교체 하겠습니다"
##  2 "정치교체 하겠습니다"
##  3 "시대교체 하겠습니다"
##  4 ""
##  5 "불비불명 이라는 고사가 있습니다 남쪽 언덕 나뭇가지에 앉아 년 동안 날지도 울지도 않는
새 그러나 그 새는 한번 날면…
##  6 ""
##  7 "그 동안 정치와 거리를 둬 왔습니다 그러나 암울한 시대가 저를 정치로 불러냈습니다 더
이상 남쪽 나뭇가지에 머무를 수…
##  8 ""
##  9 ""
## 10 "우리나라 대통령 이 되겠습니다"
## # … with 107 more rows
```

명사 기준 토큰화

```
word_noun <- moon %>%
  unnest_tokens(input = value,
                output = word,
                token = extractNoun)

word_noun
```

```
## # A tibble: 1,757 x 1
##    word
##    <chr>
##  1 "정권교체"
##  2 "하겠습니"
##  3 "정치"
##  4 "교체"
##  5 "하겠습니"
##  6 "시대"
##  7 "교체"
##  8 "하겠습니"
##  9 ""
## 10 "불비불명"
## # … with 1,747 more rows
```

단어 빈도 구하기

- 빈도가 높은 명사를 보면 글쓴이가 무엇을 강조했는지 알 수 있음
- `# A tibble: 704 x 2`: 연설문이 704개의 명사로 구성됨

```
word_noun <- word_noun %>%
  count(word, sort = T) %>%    # 단어 빈도 구해 내림차순 정렬
  filter(str_count(word) > 1)  # 두 글자 이상만 남기기

word_noun
```

```
## # A tibble: 704 x 2
##    word       n
##    <chr>  <int>
##  1 국민      21
##  2 일자리    21
##  3 나라      19
##  4 우리      17
##  5 경제      15
##  6 사회      14
##  7 성장      13
##  8 대통령    12
##  9 정치      12
## 10 하게      12
## # … with 694 more rows
```

띄어쓰기 기준 추출

```
moon %>%
  unnest_tokens(input = value,
                output = word,
                token = "words") %>%
  count(word, sort = T) %>%
  filter(str_count(word) > 1)
```

```
## # A tibble: 1,384 x 2
##    word              n
##    <chr>         <int>
##  1 합니다           27
##  2 있습니다         13
##  3 저는             13
##  4 있는             12
##  5 함께             12
##  6 만들겠습니다     11
##  7 일자리           10
##  8 국민의            9
##  9 우리              9
## 10 우리나라          9
## # … with 1,374 more rows
```

명사 추출

```
moon %>%
  unnest_tokens(input = value,
                output = word,
                token = extractNoun) %>%
  count(word, sort = T) %>%
  filter(str_count(word) > 1)
```

```
## # A tibble: 704 x 2
##    word       n
##    <chr>  <int>
##  1 국민      21
##  2 일자리    21
##  3 나라      19
##  4 우리      17
##  5 경제      15
##  6 사회      14
##  7 성장      13
##  8 대통령    12
##  9 정치      12
## 10 하게      12
## # … with 694 more rows
```

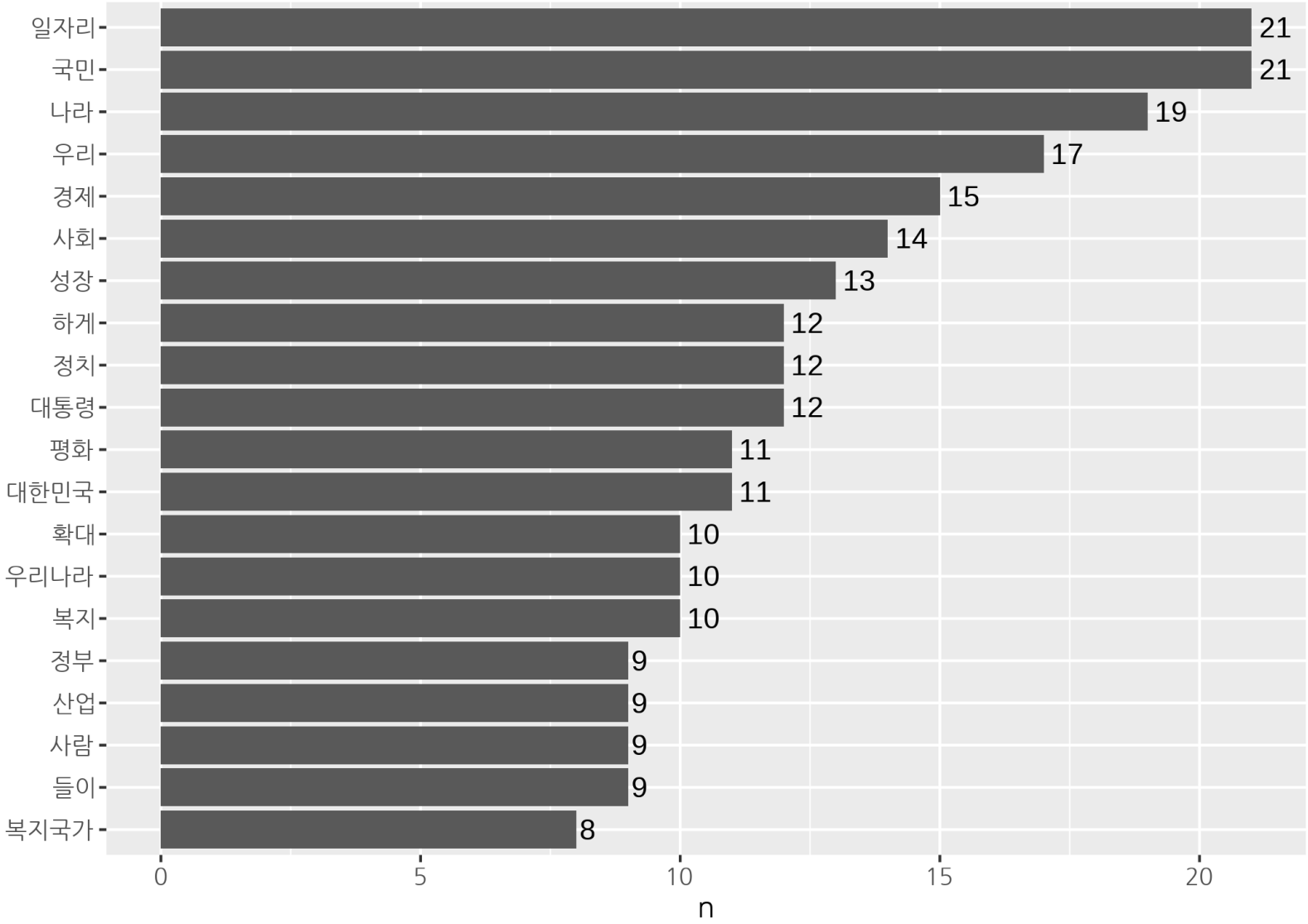
막대 그래프 만들기

```
# 상위 20개 단어 추출
top20 <- word_noun %>%
  head(20)

top20
```

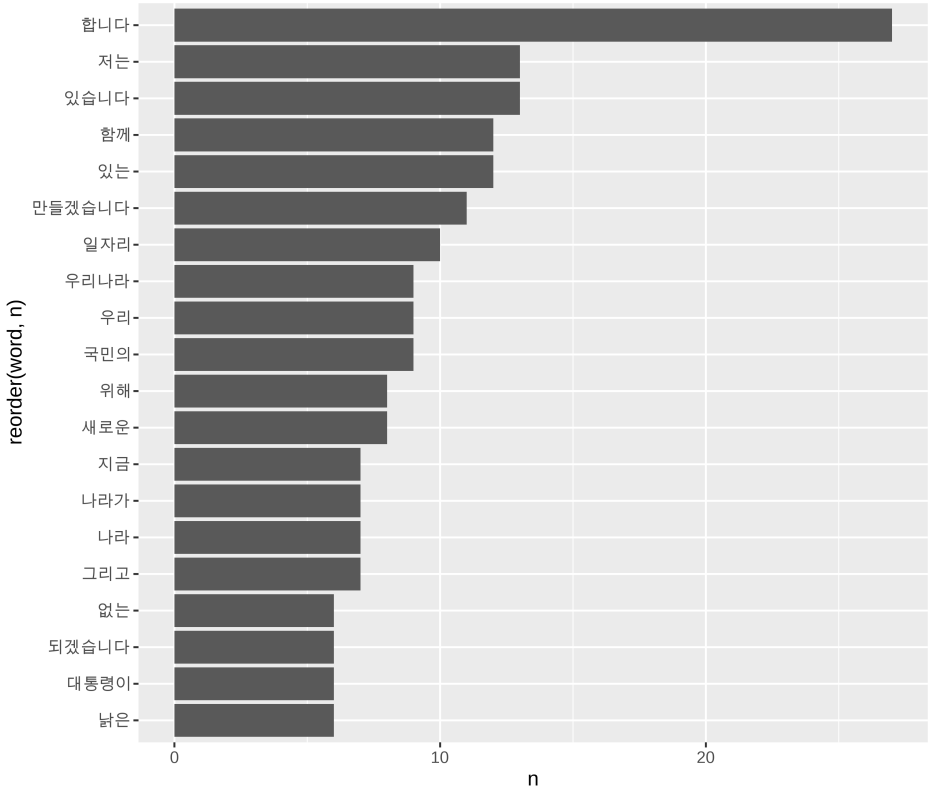
```
## # A tibble: 20 x 2
##    word          n
##    <chr>     <int>
##  1 국민         21
##  2 일자리       21
##  3 나라         19
##  4 우리         17
##  5 경제         15
##  6 사회         14
##  7 성장         13
##  8 대통령       12
##  9 정치         12
## 10 하게         12
## 11 대한민국     11
## 12 평화         11
## 13 복지         10
```

```r
# 막대 그래프 만들기
library(ggplot2)

ggplot(top20, aes(x = reorder(word, n), y = n)) +
  geom_col() +
  coord_flip() +
  geom_text(aes(label = n), hjust = -0.3) +
  labs(x = NULL) +
  theme(text = element_text(family = "nanumgothic"))
```

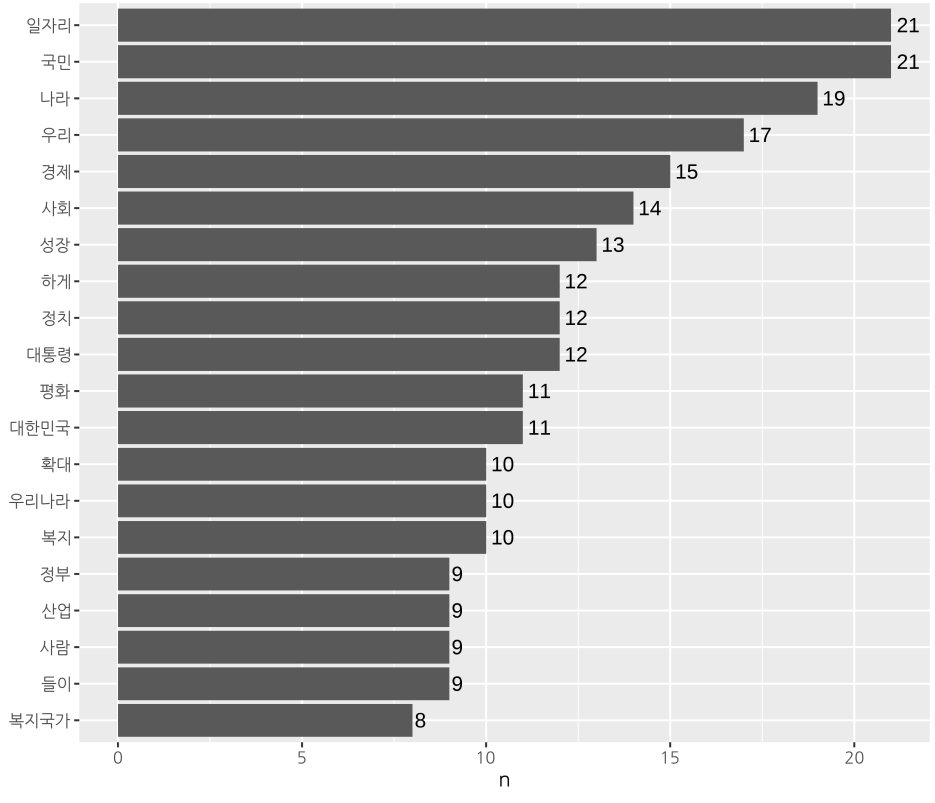
- 명사로 되어있기 때문에 연설문의 내용을 이해하기 쉬움



띄어쓰기 기준 추출



명사 추출



워드 클라우드 만들기

```
# 폰트 설정
library(showtext)
font_add_google(name = "Black Han Sans", family = "blackhansans")
showtext_auto()

library(ggwordcloud)
ggplot(word_noun, aes(label = word, size = n, col = n)) +
  geom_text_wordcloud(seed = 1234, family = "blackhansans") +
  scale_radius(limits = c(3, NA),
               range = c(3, 15)) +
  scale_color_gradient(low = "#66aaf2", high = "#004EA1") +
  theme_minimal()
```


지배
요구
진정
책임
질서
교육혁신
모델
기업
나가겠습니
변화
혁명
집단
원전
희망
선언
정의
우리나라
결과
중소기업
정책
환경
서민
관계
투자
남북
국가
상생
강화
양극화
세계
거리
공정
들이
행복
협력
일자리
대한민국
시민
부담
혁신
노인
사람
성장
과도
재벌
이명박
강자
군대
어르신
아이들
하게
사회
사람들
민주화
여성
국민
불안
약자
에너지
평생
나라
확대
시대
가지
지방
성장전략
사업
아래
공평
복지
추진
우리
특권
가치
평화
세상
한반도
비중
기회
개선
경제
교육
건강
대통령
지원
되겠습니
채택
경쟁
정치
비정규직
복지국가
참여
하기
신산업
소수
창출
대통령의
산업
때문
정부
고용
그것
담쟁이
구조
원칙
보통
고통
역사
출마
권력
시장
비전
하겠습니
전환
강력
주인
문재인
활용
세금
적극

띄어쓰기 기준 추출

명사 추출

02 비교 분석: 무엇이 다를까?

목차

02-1 단어 빈도 비교하기

비교 분석

- 여러 텍스트를 비교해 차이를 알아보는 분석 방법
- 단어 빈도 분석을 응용해 자주 사용된 단어의 차이를 살펴봄

텍스트 합치기

- 텍스트를 비교하기 위해 여러 개의 텍스트를 하나의 데이터셋으로 합치는 작업

데이터 불러오기

- 문재인 대통령과 박근혜 전 대통령의 대선 출마 선언문 불러오기
- tibble 구조로 변환하고 연설문 구분 위해 대통령 이름 부여

```
library(dplyr)

# 문재인 대통령 연설문 불러오기
raw_moon <- readLines("speech_moon.txt", encoding = "UTF-8")
moon <- raw_moon %>%
  as_tibble() %>%
  mutate(president = "moon")

# 박근혜 대통령 연설문 불러오기
raw_park <- readLines("speech_park.txt", encoding = "UTF-8")
park <- raw_park %>%
  as_tibble() %>%
  mutate(president = "park")
```

데이터 합치기

```
bind_speeches <- bind_rows(moon, park) %>%
  select(president, value)
```

집단별 단어 빈도 구하기

기본적인 전처리 및 토큰화

- 한글 이외의 문자, 연속된 공백 제거
- `bind_speeches`는 tibble 구조이므로 `mutate()` 활용

```
# 기본적인 전처리
library(stringr)
speeches <- bind_speeches %>%
  mutate(value = str_replace_all(value, "[^가-힣]", " "),
         value = str_squish(value))

speeches
```

```
## # A tibble: 213 x 2
##    president value
##    <chr>     <chr>
##  1 moon      "정권교체 하겠습니다"
##  2 moon      "정치교체 하겠습니다"
##  3 moon      "시대교체 하겠습니다"
##  4 moon      ""
##  5 moon      "불비불명 이라는 고사가 있습니다 남쪽 언덕 나뭇가지에 앉아 년 …
##  6 moon      ""
##  7 moon      "그 동안 정치와 거리를 둬 왔습니다 그러나 암울한 시대가 저를 …
##  8 moon      ""
##  9 moon      ""
## 10 moon      "우리나라 대통령 이 되겠습니다"
## # … with 203 more rows
```

- 형태소 분석기를 이용해 명사 기준 토큰화

```
# 토큰화
library(tidytext)
library(KoNLP)

speeches <- speeches %>%
  unnest_tokens(input = value,
                output = word,
                token = extractNoun)
```

두 연설문의 단어 빈도 구하기

```
frequency <- speeches %>%
  count(president, word) %>%   # 연설문 및 단어별 빈도
  filter(str_count(word) > 1)  # 두 글자 이상 추출

head(frequency)
```

```
## # A tibble: 6 x 3
##   president word         n
##   <chr>     <chr>    <int>
## 1 moon      가동         1
## 2 moon      가사         1
## 3 moon      가슴         2
## 4 moon      가족         1
## 5 moon      가족구조     1
## 6 moon      가지         4
```

`count()`는 입력한 변수의 알파벳, 가나다순으로 행을 정렬함

연설문에 가장 많이 사용된 단어 추출하기

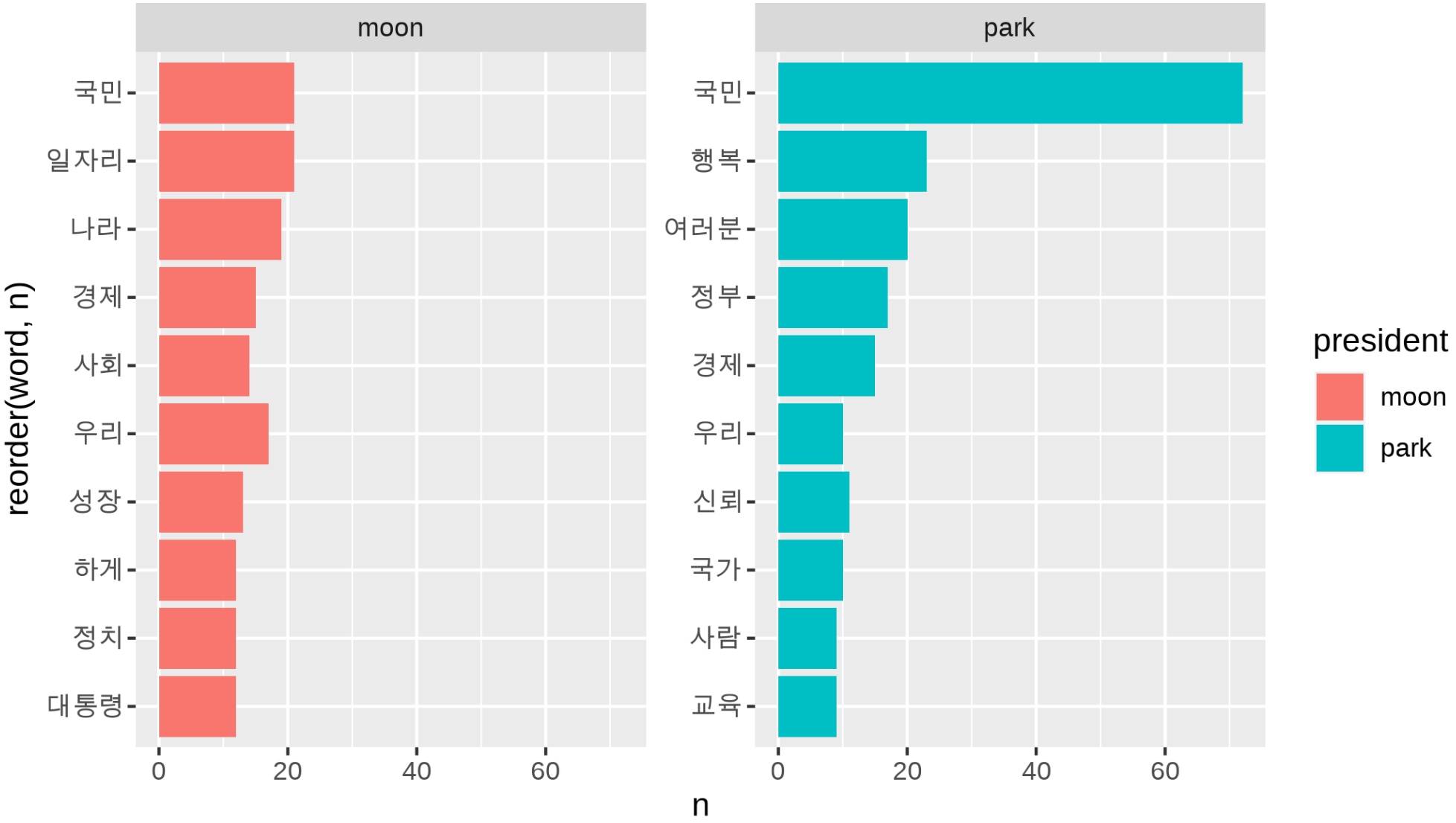
- `president`별 고빈도 단어 상위 10개 추출

```
top10 <- frequency %>%
  group_by(president) %>%
  slice_max(n, n = 10, with_ties = F)

top10
```

```
## # A tibble: 20 x 3
## # Groups:   president [2]
##    president word       n
##    <chr>     <chr>  <int>
##  1 moon      국민      21
##  2 moon      일자리    21
##  3 moon      나라      19
##  4 moon      우리      17
##  5 moon      경제      15
##  6 moon      사회      14
##  7 moon      성장      13
##  8 moon      대통령    12
##  9 moon      정치      12
## 10 moon      하게      12
## 11 park      국민      72
## 12 park      행복      23
## 13 park      여러분    20
## 14 park      정부      17
## 15 park      경제      15
## 16 park      신뢰      11
## 17 park      국가      10
## 18 park      우리      10
## 19 park      교육       9
## 20 park      사람       9
```

막대 그래프 만들기

```
ggplot(top10, aes(x = reorder(word, n),
                  y = n,
                  fill = president)) +
  geom_col() +
  coord_flip() +
  facet_wrap(~ president,          # president별 그래프 생성
             scales = "free_y")  # y축 통일하지 않음
```

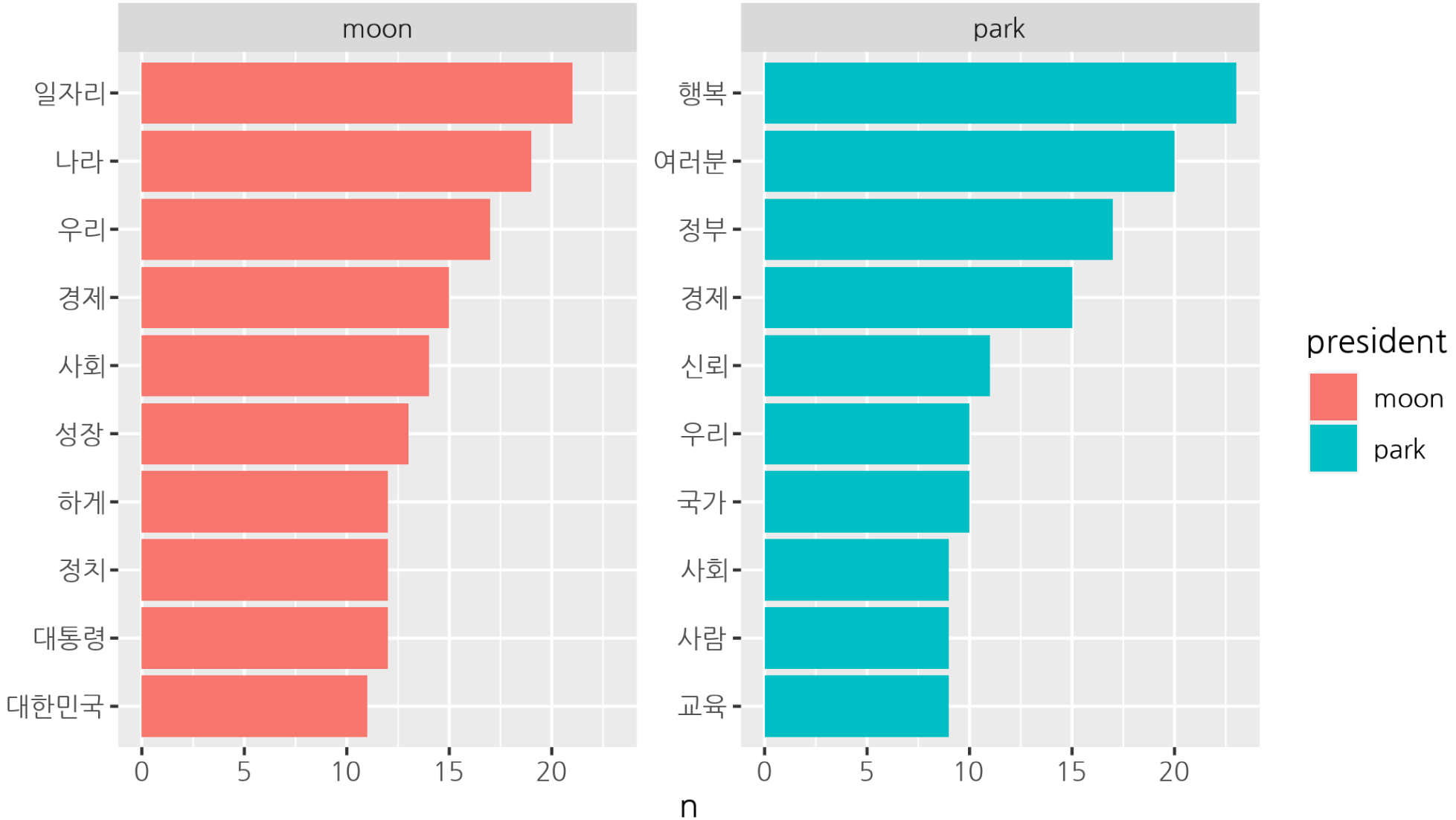

moon
park
국민
일자리
나라
경제
사회
우리
성장
하게
정치
대통령
행복
여러분
정부
신뢰
국가
사람
교육
reorder(word, n)
0
20
40
60
n
president
moon
park

특정 단어 제외하고 막대 그래프 만들기

- 박근혜 전 대통령 `"국민"` 빈도 너무 높아 다른 단어들 차이 드러나지 않음
- 전반적인 단어 빈도가 잘 드러나도록 제거

```
top10 <- frequency %>%
  filter(word != "국민") %>%
  group_by(president) %>%
  slice_max(n, n = 10, with_ties = F)
```

```
ggplot(top10, aes(x = reorder_within(word, n, president),  # 변수 항목별 축 순서 구하기
                  y = n,
                  fill = president)) +
  geom_col() +
  coord_flip() +
  facet_wrap(~ president, scales = "free_y") +
  scale_x_reordered() +
  labs(x = NULL) +                                         # x축 삭제
  theme(text = element_text(family = "nanumgothic"))       # 폰트
```


moon
park
일자리
나라
우리
경제
사회
성장
하게
정치
대통령
대한민국
행복
여러분
정부
경제
신뢰
우리
국가
사회
사람
교육
0
5
10
15
20
n
president
moon
park

02-2 오즈비:
상대적으로 중요한 단어 비교하기

- 빈도 높은 단어를 비교하면
  - 어떤 텍스트든 일반적인 단어 빈도 높아 텍스트 차이 잘 드러나지 않음
    - ex) 연설문: `"우리"`, `"사회"`, `"경제"`, `"일자리"`
- 텍스트의 차이를 알아보려면
  - 특정 텍스트에는 많이 사용되었지만 다른 텍스트에는 적게 사용된 단어를 살펴봐야함

오즈비(odds ratio)

- 어떤 사건이 A 조건에서 발생할 확률이 B 조건에서 발생할 확률에 비해 얼마나 더 큰지를 나타냄
- 단어가 두 텍스트 중 어디에 등장할 확률이 높은지, 상대적인 중요도를 알 수 있음

$$\text{odds ratio} = \frac{\left(\frac{n+1}{\text{total}+1}\right)_{\text{Text A}}}{\left(\frac{n+1}{\text{total}+1}\right)_{\text{Text B}}}$$

- n: 각 단어의 빈도
- total: 전체 단어 빈도

연설문 단어 빈도를 Wide form으로 변환하기

```
library(tidyr)
frequency_wide <- frequency %>%
  pivot_wider(names_from = president,
              values_from = n,
              values_fill = list(n = 0))

frequency_wide
```

```
## # A tibble: 955 x 3
##    word      moon  park
##    <chr>    <int> <int>
##  1 가동         1     0
##  2 가사         1     0
##  3 가슴         2     0
##  4 가족         1     1
##  5 가족구조     1     0
##  6 가지         4     0
##  7 가치         3     1
##  8 각종         1     0
##  9 감당         1     0
## 10 강력         3     0
## # … with 945 more rows
```

오즈비 구하기

```
frequency_wide <- frequency_wide %>%
  mutate(odds_ratio = ((moon + 1)/(sum(moon + 1)))/
                      ((park + 1)/(sum(park + 1))))
```

- 오즈비를 보면 단어가 어떤 텍스트에서 상대적으로 더 많이 사용됐는지 알 수 있음
  - `"moon"`에서 상대적인 비중 클수록 1보다 큰 값
  - `"park"`에서 상대적인 비중 클수록 1보다 작은 값
  - 두 연설문에서 단어 비중 같으면 1

```
frequency_wide %>%
  arrange(-odds_ratio)
```

```
## # A tibble: 955 x 4
##    word      moon  park odds_ratio
##    <chr>    <int> <int>      <dbl>
##  1 복지국가     8     0       7.12
##  2 세상         6     0       5.54
##  3 여성         6     0       5.54
##  4 정의         6     0       5.54
##  5 강자         5     0       4.75
##  6 공평         5     0       4.75
##  7 대통령의     5     0       4.75
##  8 보통         5     0       4.75
##  9 상생         5     0       4.75
## 10 지방         5     0       4.75
## # … with 945 more rows
```

상대적으로 중요한 단어 추출하기

오즈비가 가장 높거나 가장 낮은 단어 추출하기

```
top10 <- frequency_wide %>%
  filter(rank(odds_ratio) <= 10 | rank(-odds_ratio) <= 10)
```

- 상위 10개: `"moon"`에서 더 자주 사용되어 `odds_ratio`가 높은 단어

```
## # A tibble: 20 x 4
##    word      moon  park odds_ratio
##    <chr>    <int> <int>      <dbl>
##  1 복지국가     8     0     7.12
##  2 세상         6     0     5.54
##  3 여성         6     0     5.54
##  4 정의         6     0     5.54
##  5 강자         5     0     4.75
##  6 공평         5     0     4.75
##  7 대통령의     5     0     4.75
##  8 보통         5     0     4.75
##  9 상생         5     0     4.75
## 10 지방         5     0     4.75
## 11 과제         0     4     0.158
## 12 국정운영     0     4     0.158
## 13 시작         0     4     0.158
## 14 지식         0     4     0.158
## 15 행복         3    23     0.132
## 16 실천         0     5     0.132
## 17 정보         0     5     0.132
## 18 투명         0     5     0.132
## 19 여러분       2    20     0.113
## 20 박근혜       0     8     0.0879
```

- 하위 10개: `"park"`에서 더 자주 사용되어 `odds_ratio`가 낮은 단어

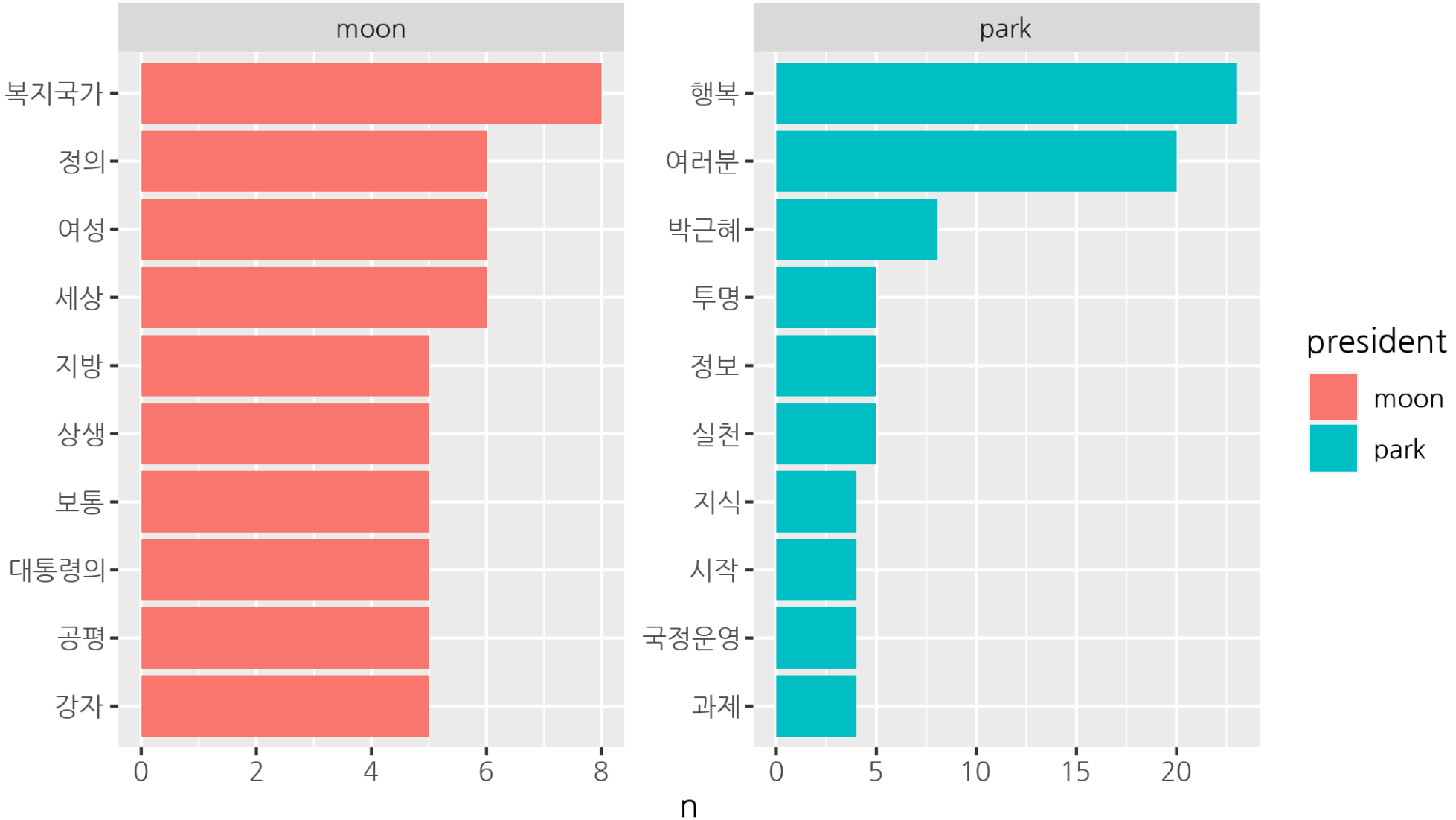
```
## # A tibble: 20 x 4
##    word      moon  park odds_ratio
##    <chr>    <int> <int>      <dbl>
##  1 복지국가     8     0     7.12
##  2 세상         6     0     5.54
##  3 여성         6     0     5.54
##  4 정의         6     0     5.54
##  5 강자         5     0     4.75
##  6 공평         5     0     4.75
##  7 대통령의     5     0     4.75
##  8 보통         5     0     4.75
##  9 상생         5     0     4.75
## 10 지방         5     0     4.75
## 11 과제         0     4     0.158
## 12 국정운영     0     4     0.158
## 13 시작         0     4     0.158
## 14 지식         0     4     0.158
## 15 행복         3    23     0.132
## 16 실천         0     5     0.132
## 17 정보         0     5     0.132
## 18 투명         0     5     0.132
## 19 여러분       2    20     0.113
## 20 박근혜       0     8     0.0879
```

막대 그래프 만들기

```
# 비중이 큰 연설문을 나타낸 변수 추가하기
top10 <- top10 %>%
  mutate(president = ifelse(odds_ratio > 1, "moon", "park"),
         n = ifelse(odds_ratio > 1, moon, park))

top10
```

```
## # A tibble: 20 x 6
##    word      moon  park odds_ratio president     n
##    <chr>    <int> <int>      <dbl> <chr>     <int>
##  1 강자         5     0      4.75  moon          5
##  2 공평         5     0      4.75  moon          5
##  3 대통령의     5     0      4.75  moon          5
##  4 보통         5     0      4.75  moon          5
##  5 복지국가     8     0      7.12  moon          8
##  6 상생         5     0      4.75  moon          5
##  7 세상         6     0      5.54  moon          6
##  8 여러분       2    20      0.113 park         20
##  9 여성         6     0      5.54  moon          6
## 10 정의         6     0      5.54  moon          6
## # … with 10 more rows
```

```r
ggplot(top10, aes(x = reorder_within(word, n, president),
                  y = n,
                  fill = president)) +
  geom_col() +
  coord_flip() +
  facet_wrap(~ president, scales = "free") +
  scale_x_reordered() +
  labs(x = NULL) +                                    # x축 삭제
  theme(text = element_text(family = "nanumgothic"))  # 폰트
```


moon
park
복지국가
정의
여성
세상
지방
상생
보통
대통령의
공평
강자
행복
여러분
박근혜
투명
정보
실천
지식
시작
국정운영
과제
0
2
4
6
8
0
5
10
15
20
n
president
moon
park

로그 오즈비로 단어 비교하기

로그 오즈비(log odds ratio)

- 오즈비에 로그를 취한 값
- 단어의 오즈비가 1보다 크면 `+`, 1보다 작으면 `-`가 됨
- 단어가 두 텍스트 중 어디에서 비중이 큰지에 따라 서로 다른 부호
  - `"moon"`에서 비중이 커서 `odds_ratio`가 1보다 큰 단어 `+`
  - `"park"`에서 비중이 커서 `odds_ratio`가 1보다 작은 단어 `-`

주요 텍스트	단어	오즈비	로그 오즈비
moon	복지국가	7.12	1.96
moon	세상	5.54	1.71
park	박근혜	0.09	-2.43
park	여러분	0.11	-2.18

로그 오즈비(log odds ratio)

- 오즈비에 로그를 취한 값
- 단어의 오즈비가 1보다 크면 `+`, 1보다 작으면 `-`가 됨
- 단어가 두 텍스트 중 어디에서 비중이 큰지에 따라 서로 다른 부호
  - `"moon"`에서 비중이 커서 `odds_ratio`가 1보다 큰 단어 `+`
  - `"park"`에서 비중이 커서 `odds_ratio`가 1보다 작은 단어 `-`

$$\text{odds ratio} = \frac{\left(\frac{n+1}{\text{total}+1}\right)_{\text{Text A}}}{\left(\frac{n+1}{\text{total}+1}\right)_{\text{Text B}}}$$

로그 오즈비(log odds ratio)

- 오즈비에 로그를 취한 값
- 단어의 오즈비가 1보다 크면 `+`, 1보다 작으면 `-`가 됨
- 단어가 두 텍스트 중 어디에서 비중이 큰지에 따라 서로 다른 부호
  - `"moon"`에서 비중이 커서 `odds_ratio`가 1보다 큰 단어 `+`
  - `"park"`에서 비중이 커서 `odds_ratio`가 1보다 작은 단어 `-`

$$\log \text{odds ratio} = \log \left(\frac{\left(\frac{n+1}{\text{total}+1} \right)_{\text{Text A}}}{\left(\frac{n+1}{\text{total}+1} \right)_{\text{Text B}}} \right)$$

로그 오즈비 구하기

```
frequency_wide <- frequency_wide %>%
  mutate(log_odds_ratio = log(((moon + 1) / (sum(moon + 1))) /
                              ((park + 1) / (sum(park + 1)))))
```

로그 오즈비를 이용해 중요한 단어 비교하기

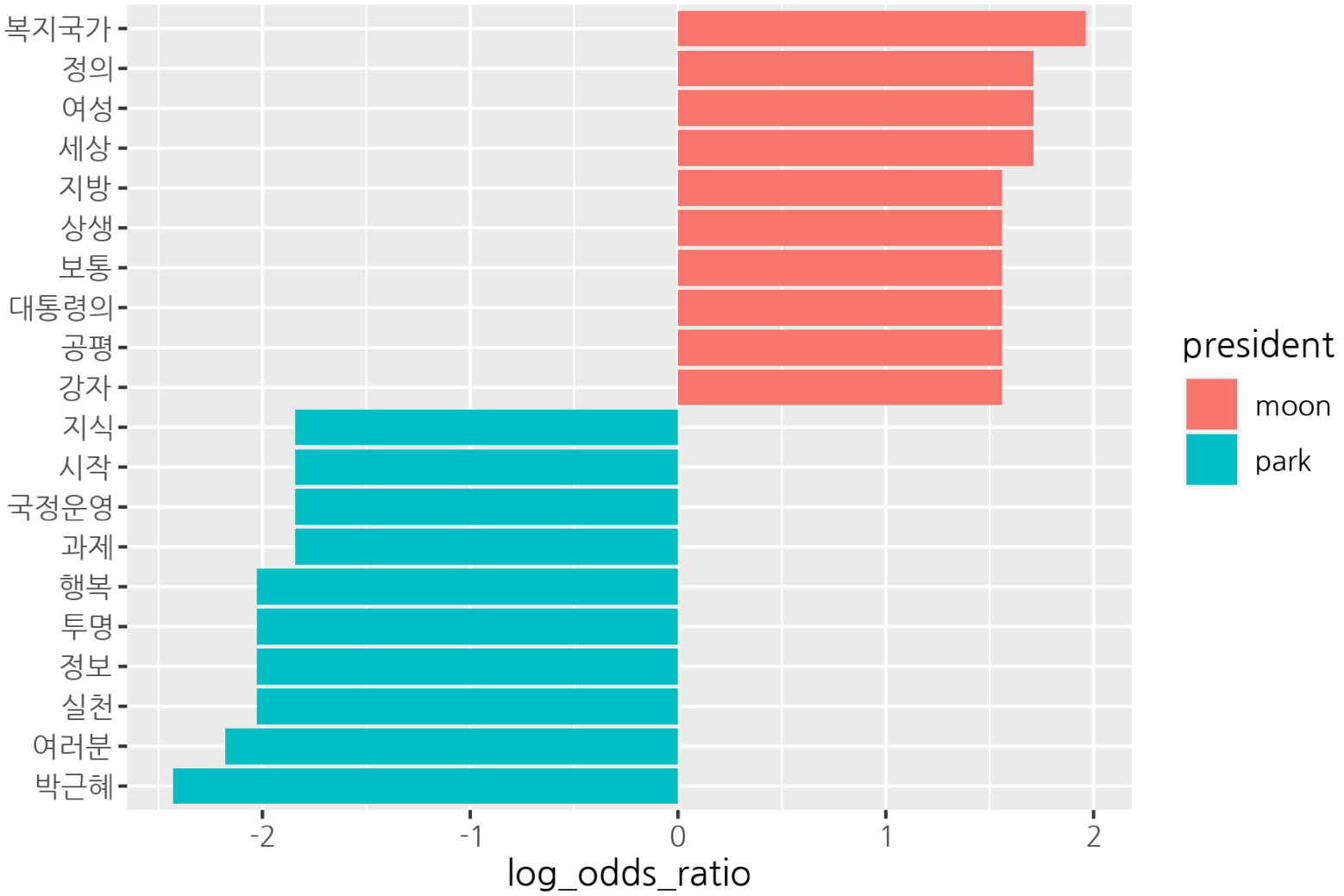
- 두 연설문 각각 `log_odds_ratio` Top 10 추출

```
top10 <- frequency_wide %>%
  group_by(president = ifelse(log_odds_ratio > 0, "moon", "park")) %>%
  slice_max(abs(log_odds_ratio), n = 10, with_ties = F)
```

막대 그래프 만들기

- 단어가 어느 연설문에서 중요한지에 따라 서로 다른 축 방향으로 표현됨

```
ggplot(top10, aes(x = reorder(word, log_odds_ratio),
                  y = log_odds_ratio,
                  fill = president)) +
  geom_col() +
  coord_flip() +
  labs(x = NULL) +
  theme(text = element_text(family = "nanumgothic"))
```


복지국가
정의
여성
세상
지방
상생
보통
대통령의
공평
강자
지식
시작
국정운영
과제
행복
투명
정보
실천
여러분
박근혜
-2
-1
0
1
2
log_odds_ratio
president
moon
park

02-3 TF-IDF:
여러 텍스트의 단어 비교하기

오즈비의 한계

- 두 조건의 확률을 이용해 계산
- 여러 텍스트 비교하기 불편
- 두 개 이상의 텍스트 비교할 때는 **TF-IDF** 활용

TF-IDF(Term Frequency - Inverse Document Frequency)

- 어떤 단어가 흔하지 않으면서도 특정 텍스트에서는 자주 사용된 정도를 나타낸 지표
- 텍스트의 개성을 드러내는 주요 단어를 찾는데 활용
- TF(단어 빈도)와 IDF(역 문서 빈도)를 곱한 값
  - TF: 단어가 분석 대상이 되는 텍스트 내에서 많이 사용될수록 커짐
  - IDF: 단어가 사용된 텍스트가 드물수록 커짐

$$\text{TF-IDF} = TF \times \log \frac{\text{N}}{\text{DF}}$$

TF-IDF

- **흔하지 않은 단어**인데 **특정 텍스트에서 자주 사용**될수록 큰 값
  - TF-IDF가 큰 단어를 보면 다른 텍스트와 구별되는 특징을 알 수 있음

단어	자기소개서 A	자기소개서 B	자기소개서 C
저는	$15\times \log\frac{3}{3}=0$	$10\times \log\frac{3}{3}=0$	$10\times \log\frac{3}{3}=0$
스카이다이빙	$3\times \log\frac{3}{1}=3.3$	$0\times \log\frac{3}{1}=0$	$0\times \log\frac{3}{1}=0$
자기주도적	$3\times \log\frac{3}{3}=0$	$5\times \log\frac{3}{3}=0$	$3\times \log\frac{3}{3}=0$
데이터	$0\times \log\frac{3}{2}=0$	$5\times \log\frac{3}{2}=2$	$1\times \log\frac{3}{2}=0.4$
배낭여행	$2\times \log\frac{3}{3}=0$	$3\times \log\frac{3}{3}=0$	$5\times \log\frac{3}{3}=0$

💡 소수점 둘째 자리에서 반올림하여 표기

TF-IDF 구하기

1. 단어 빈도 구하기

```
# 데이터 불러오기
install.packages("readr")
library(readr)

raw_speeches <- read_csv("speeches_presidents.csv")
raw_speeches
```

```
## # A tibble: 4 x 2
##   president value
##   <chr>     <chr>
## 1 문재인    "정권교체 하겠습니다!   정치교체 하겠습니다!   시대교체 하겠습…
## 2 박근혜    "존경하는 국민 여러분! 저는 오늘, 국민 한 분 한 분의 꿈이 이…
## 3 이명박    "존경하는 국민 여러분, 사랑하는 한나라당 당원 동지 여러분! 저는…
## 4 노무현    "어느때인가 부터 제가 대통령이 되겠다고 말을 하기 시작했습니다. …
```

```r
# 기본적인 전처리
speeches <- raw_speeches %>%
  mutate(value = str_replace_all(value, "[^가-힣]", " "),
         value = str_squish(value))

# 토큰화
speeches <- speeches %>%
  unnest_tokens(input = value,
                output = word,
                token = extractNoun)

# 단어 빈도 구하기
frequecy <- speeches %>%
  count(president, word) %>%
  filter(str_count(word) > 1)

frequecy
```

```
## # A tibble: 1,513 x 3
##    president word      n
##    <chr>     <chr> <int>
##  1 노무현    가슴      2
##  2 노무현    가훈      2
##  3 노무현    갈등      1
##  4 노무현    감옥      1
##  5 노무현    강자      1
##  6 노무현    개편      4
##  7 노무현    개혁      4
##  8 노무현    건국      1
##  9 노무현    경선      1
## 10 노무현    경쟁      1
## # … with 1,503 more rows
```

3.4.2 TF-IDF 구하기

- `tidytext::bind_tf_idf()`
  - `term` : 단어
  - `document` : 텍스트 구분 기준
  - `n` : 단어 빈도

```
frequecy <- frequecy %>%
  bind_tf_idf(term = word,            # 단어
              document = president,   # 텍스트 구분 기준
              n = n) %>%              # 단어 빈도
  arrange(-tf_idf)

frequecy
```

```
## # A tibble: 1,513 x 6
##    president word         n      tf   idf tf_idf
##    <chr>     <chr>    <int>   <dbl> <dbl>  <dbl>
##  1 노무현    공식         6 0.0163  1.39  0.0227
##  2 노무현    비젼         6 0.0163  1.39  0.0227
##  3 노무현    정계         6 0.0163  1.39  0.0227
##  4 이명박    리더십       6 0.0158  1.39  0.0219
##  5 노무현    권력         9 0.0245  0.693 0.0170
##  6 노무현    개편         4 0.0109  1.39  0.0151
##  7 이명박    당원         4 0.0105  1.39  0.0146
##  8 이명박    동지         4 0.0105  1.39  0.0146
##  9 이명박    일류국가     4 0.0105  1.39  0.0146
## 10 박근혜    박근혜       8 0.00962 1.39  0.0133
## # … with 1,503 more rows
```

`tf`: 대상 텍스트의 전체 단어 수에서 해당 단어의 수가 차지하는 '비중'. 텍스트에 사용된 전체 단어 수가 많을수록 작아짐

TF-IDF가 높은 단어 살펴보기

- 텍스트의 특징을 드러내는 중요한 단어
- 각 대통령이 다른 대통령들과 달리 무엇을 강조했는지 알 수 있음

```
frequecy %>% filter(president == "문재인")
```

```
## # A tibble: 688 x 6
##    president word          n      tf   idf  tf_idf
##    <chr>     <chr>     <int>   <dbl> <dbl>   <dbl>
##  1 문재인    복지국가      8 0.00608 1.39  0.00843
##  2 문재인    여성          6 0.00456 1.39  0.00633
##  3 문재인    공평          5 0.00380 1.39  0.00527
##  4 문재인    담쟁이        5 0.00380 1.39  0.00527
##  5 문재인    대통령의      5 0.00380 1.39  0.00527
##  6 문재인    보통          5 0.00380 1.39  0.00527
##  7 문재인    상생          5 0.00380 1.39  0.00527
##  8 문재인    우리나라     10 0.00760 0.693 0.00527
##  9 문재인    지방          5 0.00380 1.39  0.00527
## 10 문재인    확대         10 0.00760 0.693 0.00527
## # … with 678 more rows
```

TF-IDF가 높은 단어 살펴보기

- 텍스트의 특징을 드러내는 중요한 단어
- 각 대통령이 다른 대통령들과 달리 무엇을 강조했는지 알 수 있음

```
frequecy %>% filter(president == "박근혜")
```

```
## # A tibble: 407 x 6
##    president word         n      tf   idf  tf_idf
##    <chr>     <chr>    <int>   <dbl> <dbl>   <dbl>
##  1 박근혜    박근혜       8 0.00962 1.39  0.0133
##  2 박근혜    정보         5 0.00601 1.39  0.00833
##  3 박근혜    투명         5 0.00601 1.39  0.00833
##  4 박근혜    행복        23 0.0276  0.288 0.00795
##  5 박근혜    교육         9 0.0108  0.693 0.00750
##  6 박근혜    국정운영     4 0.00481 1.39  0.00666
##  7 박근혜    정부        17 0.0204  0.288 0.00588
##  8 박근혜    개개인       3 0.00361 1.39  0.00500
##  9 박근혜    개인         3 0.00361 1.39  0.00500
## 10 박근혜    공개         3 0.00361 1.39  0.00500
## # … with 397 more rows
```

TF-IDF가 높은 단어 살펴보기

- 텍스트의 특징을 드러내는 중요한 단어
- 각 대통령이 다른 대통령들과 달리 무엇을 강조했는지 알 수 있음

```
frequecy %>% filter(president == "이명박")
```

```
## # A tibble: 202 x 6
##    president word         n      tf   idf  tf_idf
##    <chr>     <chr>    <int>   <dbl> <dbl>   <dbl>
##  1 이명박    리더십       6 0.0158  1.39  0.0219
##  2 이명박    당원         4 0.0105  1.39  0.0146
##  3 이명박    동지         4 0.0105  1.39  0.0146
##  4 이명박    일류국가     4 0.0105  1.39  0.0146
##  5 이명박    한나라       7 0.0184  0.693 0.0128
##  6 이명박    나라        15 0.0395  0.288 0.0114
##  7 이명박    도약         3 0.00789 1.39  0.0109
##  8 이명박    일하         3 0.00789 1.39  0.0109
##  9 이명박    사랑         5 0.0132  0.693 0.00912
## 10 이명박    인생         5 0.0132  0.693 0.00912
## # … with 192 more rows
```

TF-IDF가 높은 단어 살펴보기

- 텍스트의 특징을 드러내는 중요한 단어
- 각 대통령이 다른 대통령들과 달리 무엇을 강조했는지 알 수 있음

```
frequecy %>% filter(president == "노무현")
```

```
## # A tibble: 216 x 6
##    president word         n      tf   idf  tf_idf
##    <chr>     <chr>    <int>   <dbl> <dbl>   <dbl>
##  1 노무현    공식         6 0.0163  1.39  0.0227
##  2 노무현    비젼         6 0.0163  1.39  0.0227
##  3 노무현    정계         6 0.0163  1.39  0.0227
##  4 노무현    권력         9 0.0245  0.693 0.0170
##  5 노무현    개편         4 0.0109  1.39  0.0151
##  6 노무현    국회의원     3 0.00817 1.39  0.0113
##  7 노무현    남북대화     3 0.00817 1.39  0.0113
##  8 노무현    총리         3 0.00817 1.39  0.0113
##  9 노무현    가훈         2 0.00545 1.39  0.00755
## 10 노무현    개혁         4 0.0109  0.693 0.00755
## # … with 206 more rows
```

TF-IDF가 낮은 단어 살펴보기

- 역대 대통령들이 공통적으로 사용한 흔한 단어, 범용 단어

```
frequecy %>%
  filter(president == "문재인") %>%
  arrange(tf_idf)
```

```
## # A tibble: 688 x 6
##    president word       n       tf   idf tf_idf
##    <chr>     <chr>  <int>    <dbl> <dbl>  <dbl>
##  1 문재인    경쟁       6 0.00456      0      0
##  2 문재인    경제      15 0.0114       0      0
##  3 문재인    고통       4 0.00304      0      0
##  4 문재인    과거       1 0.000760     0      0
##  5 문재인    국민      21 0.0160       0      0
##  6 문재인    기회       5 0.00380      0      0
##  7 문재인    대통령    12 0.00913      0      0
##  8 문재인    동안       2 0.00152      0      0
##  9 문재인    들이       9 0.00684      0      0
## 10 문재인    마음       2 0.00152      0      0
## # … with 678 more rows
```

TF-IDF가 낮은 단어 살펴보기

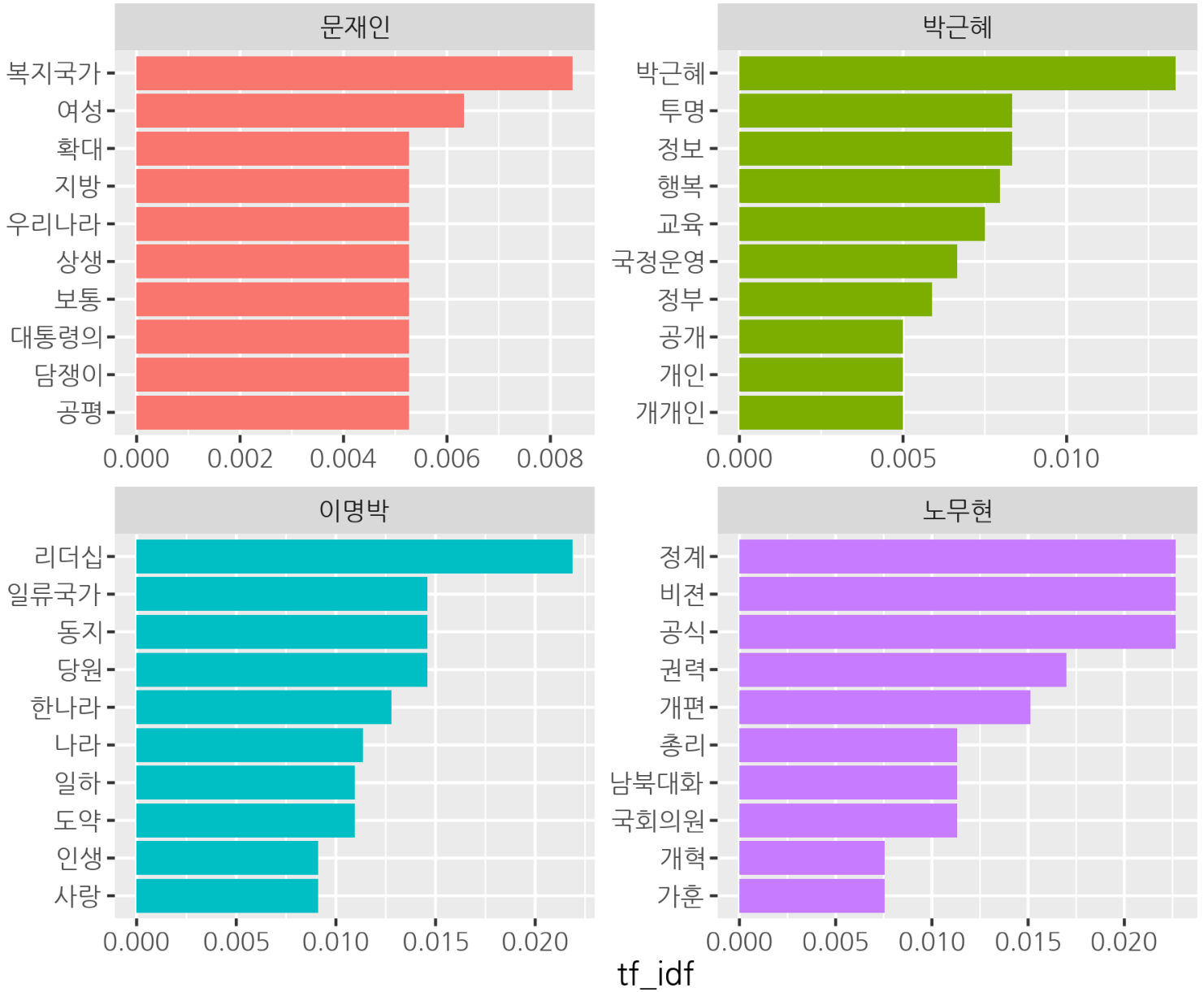
- 역대 대통령들이 공통적으로 사용한 흔한 단어, 범용 단어

```
frequecy %>%
  filter(president == "박근혜") %>%
  arrange(tf_idf)
```

```
## # A tibble: 407 x 6
##    president word       n      tf   idf tf_idf
##    <chr>     <chr>  <int>   <dbl> <dbl>  <dbl>
##  1 박근혜    경쟁       1 0.00120     0      0
##  2 박근혜    경제      15 0.0180      0      0
##  3 박근혜    고통       4 0.00481     0      0
##  4 박근혜    과거       2 0.00240     0      0
##  5 박근혜    국민      72 0.0865      0      0
##  6 박근혜    기회       1 0.00120     0      0
##  7 박근혜    대통령     3 0.00361     0      0
##  8 박근혜    동안       3 0.00361     0      0
##  9 박근혜    들이       3 0.00361     0      0
## 10 박근혜    마음       3 0.00361     0      0
## # … with 397 more rows
```

막대 그래프 만들기

```
# 주요 단어 추출
top10 <- frequecy %>%
  group_by(president) %>%
  slice_max(tf_idf, n = 10, with_ties = F)

# 그래프 순서 정하기
top10$president <- factor(top10$president,
                          levels = c("문재인", "박근혜", "이명박", "노무현"))

# 막대 그래프 만들기
ggplot(top10, aes(x = reorder_within(word, tf_idf, president),
                  y = tf_idf,
                  fill = president)) +
  geom_col(show.legend = F) +
  coord_flip() +
  facet_wrap(~ president, scales = "free", ncol = 2) +
  scale_x_reordered() +
  labs(x = NULL) +
  theme(text = element_text(family = "nanumgothic"))
```


문재인
복지국가
여성
확대
지방
우리나라
상생
보통
대통령의
담쟁이
공평
0.000
0.002
0.004
0.006
0.008
박근혜
박근혜
투명
정보
행복
교육
국정운영
정부
공개
개인
개개인
0.000
0.005
0.010
이명박
리더십
일류국가
동지
당원
한나라
나라
일하
도약
인생
사랑
0.000
0.005
0.010
0.015
0.020
노무현
정계
비젼
공식
권력
개편
총리
남북대화
국회의원
개혁
가훈
0.000
0.005
0.010
0.015
0.020
tf_idf

03 의미망 분석:
어떤 맥락에서 단어를 썼을까?

목차

03-1 동시 출현 단어 분석: Co-occurrence analysis

동시 출현 단어 분석(Co-occurrence analysis)

- 단어 간의 관계를 살펴보는 분석 방법
- '손-장갑', '머리-모자' 처럼 관계가 있는 단어 파악
- 단어의 관계를 표현한 의미망(semantic network) 만드는데 활용

기본적인 전처리

```
# 기생충 기사 댓글 불러오기
library(readr)
raw_news_comment <- read_csv("news_comment_parasite.csv")

# 전처리
library(dplyr)
library(stringr)
library(textclean)

news_comment <- raw_news_comment %>%
  select(reply) %>%
  mutate(reply = str_replace_all(reply, "[^가-힣]", " "),
         reply = str_squish(reply),
         id = row_number())
```

토큰화하기

- 단어가 사용된 맥락을 살펴봐야 하므로 명사, 형용사, 동사 함께 추출

1. 형태소 분석기를 이용해 품사 기준으로 토큰화하기

- `SimplePos22()` : 문장의 단어를 22개의 품사로 구분

```
library(tidytext)
library(KoNLP)

comment_pos <- news_comment %>%
  unnest_tokens(input = reply,
                output = word,
                token = SimplePos22,
                drop = F)

comment_pos %>%
  select(reply, word)
```

```
## # A tibble: 39,956 x 2
##    reply                                    word
##    <chr>                                    <chr>
##  1 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 정말/ma
##  2 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 우리/np
##  3 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 집/nc+에/jc…
##  4 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 좋/pa+은/et…
##  5 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 일/nc+이/jc…
##  6 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 생기/pv+어/e…
##  7 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 기쁘/pa+고/e…
##  8 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 행복한/nc
##  9 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 것/nb+처럼/j…
## 10 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인… 나/np+의/jc…
## # … with 39,946 more rows
```

품사 분리하여 행 구성하기

- 원하는 품사를 추출하기 쉽도록 한 행을 한 품사로 구성하기
- `tidyr::separate_rows()`:
  - 정규 표현식에 따라 텍스트를 여러 행으로 나누기
  - `sep = "[+]"`: `"+"`가 등장할 때마다 행을 나눔

```
# 품사별로 행 분리
library(tidyr)
comment_pos <- comment_pos %>%
  separate_rows(word, sep = "[+]")

comment_pos %>%
  select(word, reply)
```

```
## # A tibble: 70,553 x 2
##    word    reply
##    <chr>   <chr>
##  1 정말/ma 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  2 우리/np 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  3 집/nc   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  4 에/jc   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  5 좋/pa   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  6 은/et   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  7 일/nc   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  8 이/jc   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
##  9 생기/pv 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
## 10 어/ec   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 …
## # … with 70,543 more rows
```

3. 품사 추출하기

(1) 명사 추출하기

- `"/n"`이 붙어있는 단어 추출
- 태그 제거: '/로 시작하는 모든 문자' 제거

```
# 명사 추출하기
noun <- comment_pos %>%
  filter(str_detect(word, "/n")) %>%
  mutate(word = str_remove(word, "/.*$"))

noun %>%
  select(word, reply)
```

```
## # A tibble: 27,457 x 2
##    word   reply
##    <chr>  <chr>
##  1 우리   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  2 집     정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  3 일     정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  4 행복한 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  5 것     정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  6 나     정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  7 일     정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  8 양     정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
##  9 행복   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
## 10 행복   정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 양 행…
## # … with 27,447 more rows
```

(2) 동사, 형용사 추출하기

- 동사 `"/pv"`, 형용사:`"/pa"` 붙어있는 단어 추출
- 단어 뒤에 태그 대신 '다'를 붙여 이해하기 편하게 수정하기
  - ex) "받" → "받다", "멋지" → "멋지다"

```
# 동사, 형용사 추출하기
pvpa <- comment_pos %>%
  filter(str_detect(word, "/pv|/pa")) %>%          # "/pv", "/pa" 추출
  mutate(word = str_replace(word, "/.*$", "다"))  # "/"로 시작 문자를 "다"로 바꾸기

pvpa %>%
  select(word, reply)
```

```
## # A tibble: 5,317 x 2
##    word      reply
##    <chr>     <chr>
##  1 좋다      정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  2 생기다    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  3 기쁘다    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  4 축하드리다… 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  5 기쁘다    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  6 기쁘다    와 너무 기쁘다 이 시국에 정말 내 일같이 기쁘고 감사하다 축하드…
##  7 기쁘다    와 너무 기쁘다 이 시국에 정말 내 일같이 기쁘고 감사하다 축하드…
##  8 축하드리다… 와 너무 기쁘다 이 시국에 정말 내 일같이 기쁘고 감사하다 축하드…
##  9 불다      우리나라의 영화감독분들 그리고 앞으로 그 꿈을 그리는 분들에게 큰…
## 10 크다      우리나라의 영화감독분들 그리고 앞으로 그 꿈을 그리는 분들에게 큰…
## # … with 5,307 more rows
```

(3) 추출한 데이터 결합하기

- 추출한 단어 결합하기
- 이해할 수 있는 두 글자 이상 단어만 남기기

```
# 품사 결합
comment <- bind_rows(noun, pvpa) %>%
  filter(str_count(word) >= 2) %>%
  arrange(id)

comment %>%
  select(word, reply)
```

```
## # A tibble: 26,860 x 2
##    word      reply
##    <chr>     <chr>
##  1 우리      정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  2 행복한    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  3 행복      정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  4 행복      정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  5 좋다      정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  6 생기다    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  7 기쁘다    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  8 축하드리다… 정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
##  9 기쁘다    정말 우리 집에 좋은 일이 생겨 기쁘고 행복한 것처럼 나의 일인 …
## 10 시국      와 너무 기쁘다 이 시국에 정말 내 일같이 기쁘고 감사하다 축하드…
## # … with 26,850 more rows
```

명사, 동사, 형용사를 한 번에 추출하기

- 명사, 동사, 형용사를 추출해 결합한 후 두 글자 이상만 남기기

```
comment_new <- comment_pos %>%
  separate_rows(word, sep = "[+]") %>%
  filter(str_detect(word, "/n|/pv|/pa")) %>%
  mutate(word = ifelse(str_detect(word, "/pv|/pa"),
                       str_replace(word, "/.*$", "다"),
                       str_remove(word, "/.*$"))) %>%
  filter(str_count(word) >= 2) %>%
  arrange(id)
```

단어 동시 출현 빈도 구하기

```
install.packages("widyr")
library(widyr)

pair <- comment %>%
  pairwise_count(item = word,    # 단어
                 feature = id,    # 텍스트 구분 기준
                 sort = T)        # 빈도 높은 순 정렬
pair
```

```
## # A tibble: 245,920 x 3
##    item1      item2      n
##    <chr>      <chr>  <dbl>
##  1 영화       기생충   111
##  2 기생충     영화     111
##  3 감독       봉준호    86
##  4 봉준호     감독      86
##  5 감독님     봉준호    66
##  6 봉준호     감독님    66
##  7 만들다     영화      57
##  8 영화       만들다    57
##  9 기생충     봉준호    54
## 10 블랙리스트 감독      54
## # … with 245,910 more rows
```

- 한 단어를 기준으로 함께 사용된 모든 단어의 빈도를 구함
- 순서를 바꿔가며 같은 빈도를 지니는 두 개의 행으로 구성됨
  - ex) "영화-기생충", "기생충-영화"

특정 단어와 자주 함께 사용된 단어 살펴보기

```
pair %>% filter(item1 == "영화")
```

```
## # A tibble: 2,313 x 3
##    item1 item2      n
##    <chr> <chr>  <dbl>
##  1 영화  기생충   111
##  2 영화  만들다    57
##  3 영화  봉준호    52
##  4 영화  받다      48
##  5 영화  한국      46
##  6 영화  아카데미  42
##  7 영화  같다      41
##  8 영화  감독      39
##  9 영화  아니다    38
## 10 영화  좋다      35
## # … with 2,303 more rows
```

```
pair %>% filter(item1 == "봉준호")
```

```
## # A tibble: 1,579 x 3
##    item1  item2      n
##    <chr>  <chr>  <dbl>
##  1 봉준호 감독      86
##  2 봉준호 감독님    66
##  3 봉준호 기생충    54
##  4 봉준호 영화      52
##  5 봉준호 블랙리스트  48
##  6 봉준호 대한민국  38
##  7 봉준호 자랑      33
##  8 봉준호 축하드리다  30
##  9 봉준호 송강호    30
## 10 봉준호 축하      25
## # … with 1,569 more rows
```

03-2 동시 출현 네트워크: Co-occurrence network

동시 출현 네트워크(co-occurrence network)

- 동시 출현 빈도를 이용해 단어의 관계를 네트워크 형태로 표현
- 단어들이 어떤 맥락에서 함께 사용되었는지 이해할 수 있다

네트워크 그래프 데이터 만들기

- `tidygraph::as_tbl_graph()`
- 동시 출현 빈도 데이터를 '네트워크 그래프 데이터'로 변환하기
  - 단어를 나타내는 노드(node, 꼭짓점)
  - 단어를 연결하는 엣지(edge, 선)

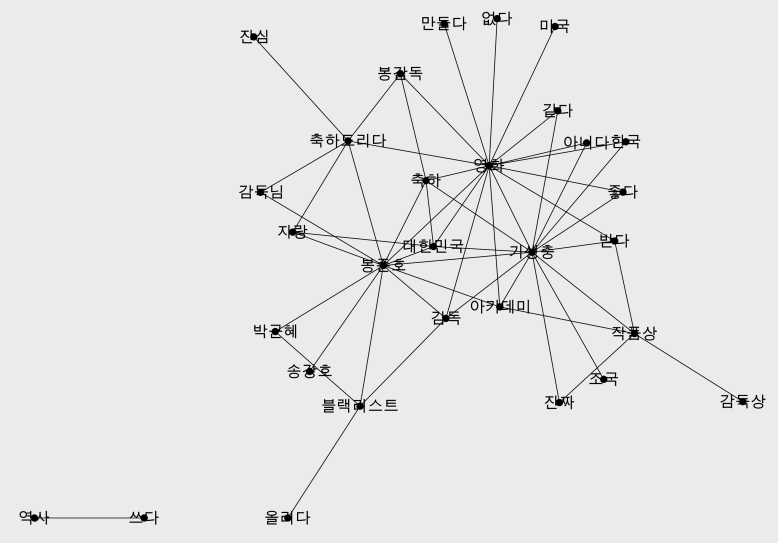
```
install.packages("tidygraph")
library(tidygraph)

graph_comment <- pair %>%
  filter(n >= 25) %>%
  as_tbl_graph()

graph_comment
```

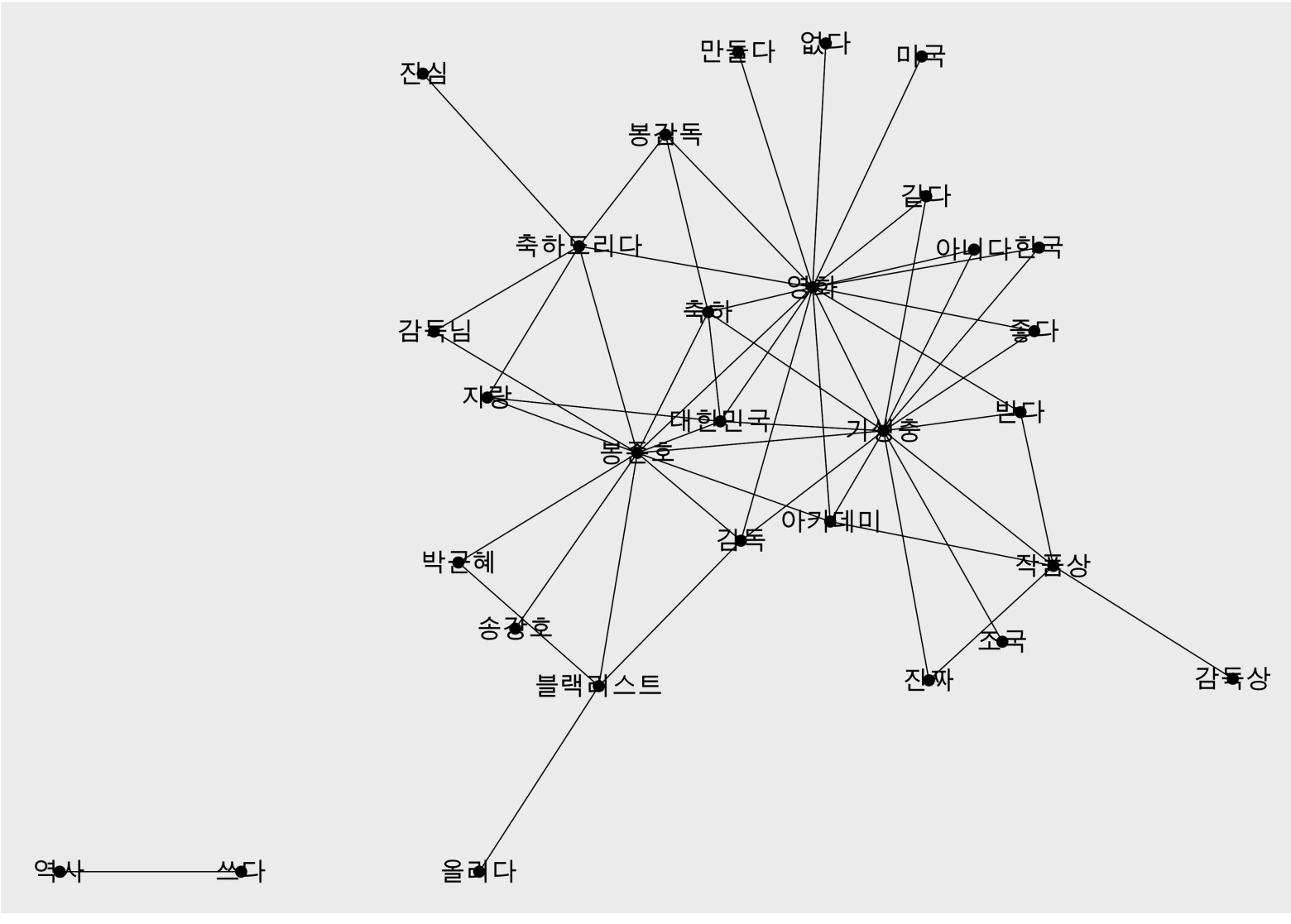
```
## # A tbl_graph: 30 nodes and 108 edges
## #
## # A directed simple graph with 2 components
## #
## # Node Data: 30 x 1 (active)
##   name
##   <chr>
## 1 영화
## 2 기생충
## 3 감독
## 4 봉준호
## 5 감독님
## 6 만들다
## # … with 24 more rows
## #
## # Edge Data: 108 x 3
##    from    to     n
##   <int> <int> <dbl>
## 1     1     2   111
## 2     2     1   111
## 3     3     4    86
## # … with 105 more rows
```

네트워크 그래프 만들기

- `ggraph::ggraph()`

```
install.packages("ggraph")
library(ggraph)

ggraph(graph_comment) +
  geom_edge_link() +                    # 엣지
  geom_node_point() +                   # 노드
  geom_node_text(aes(label = name))  # 텍스트
```




만들다
없다
마국
진심
봉감독
같다
축하드리다
아니다한국
영화
축하
감독님
좋다
자랑
대한민국
기생충
받다
봉준호
아카데미
감독
박관혜
작품상
송강호
조국
블랙리스트
진짜
감독상
역사
쓰다
올리다

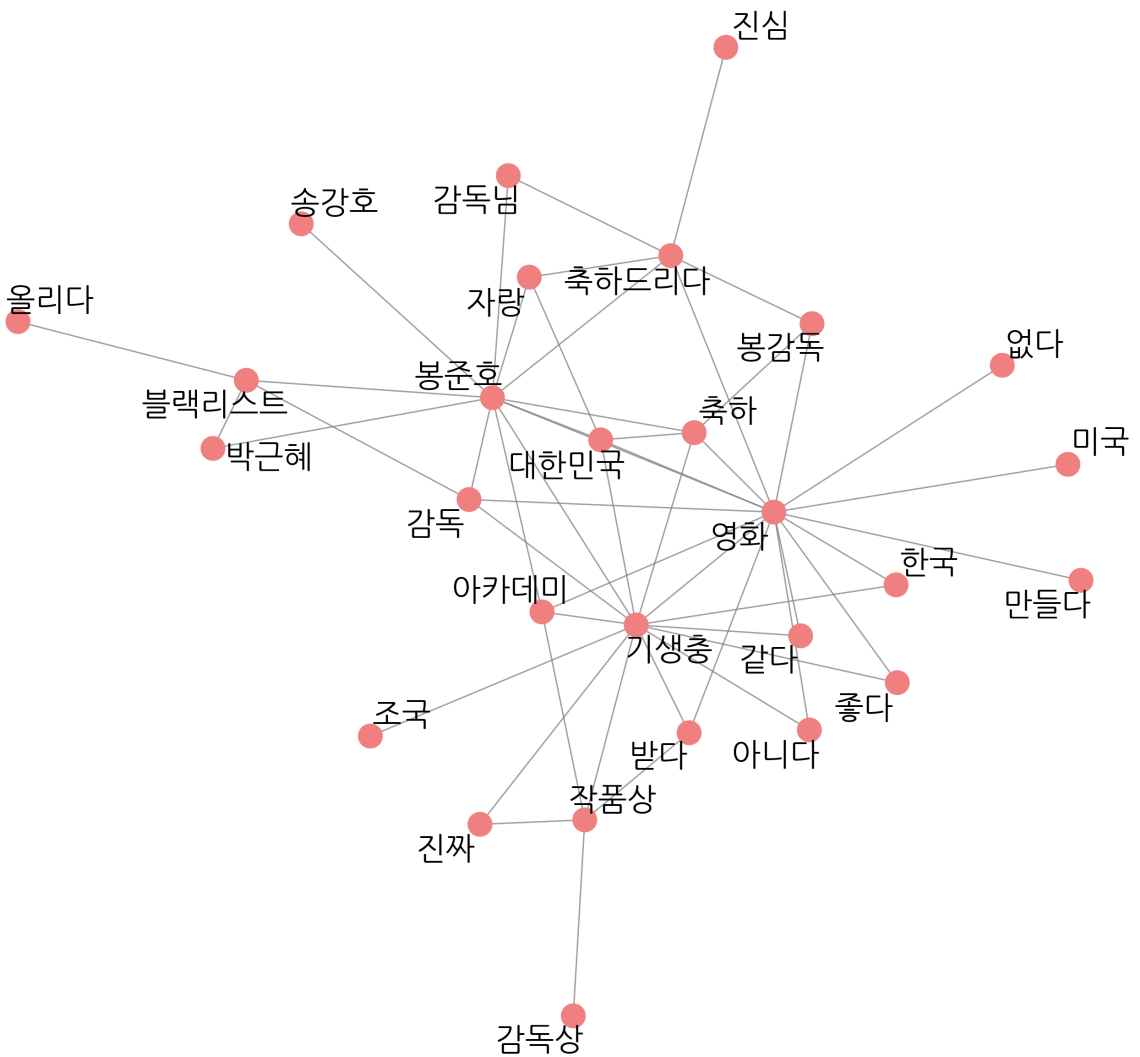
그래프 다듬기

```
# 한글 폰트 설정
library(showtext)
font_add_google(name = "Nanum Gothic", family = "nanumgothic")
showtext_auto()
```

엣지와 노드의 색깔, 크기, 텍스트 위치 수정

- `ggraph(layout = "fr")`: 네트워크 형태 결정
  - 난수를 이용해 매번 형태 달라짐 → `set.seed()`로 난수 고정

```
set.seed(1234)                                  # 난수 고정
ggraph(graph_comment, layout = "fr") +          # 레이아웃

  geom_edge_link(color = "gray50",              # 엣지 색깔
                 alpha = 0.5) +                 # 엣지 명암

  geom_node_point(color = "lightcoral",         # 노드 색깔
                  size = 5) +                   # 노드 크기

  geom_node_text(aes(label = name),             # 텍스트 표시
                 repel = T,                     # 노드밖 표시
                 size = 5,                      # 텍스트 크기
                 family = "nanumgothic") +      # 폰트

  theme_graph()                                 # 배경 삭제
```


진심
쓰다
역사
송강호
감독님
축하드리다
자랑
올리다
봉감독
없다
봉준호
블랙리스트
축하
박근혜
대한민국
미국
감독
영화
한국
아카데미
만들다
기생충
같다
조국
좋다
받다
아니다
작품상
진짜
감독상

네트워크 그래프 함수 만들기

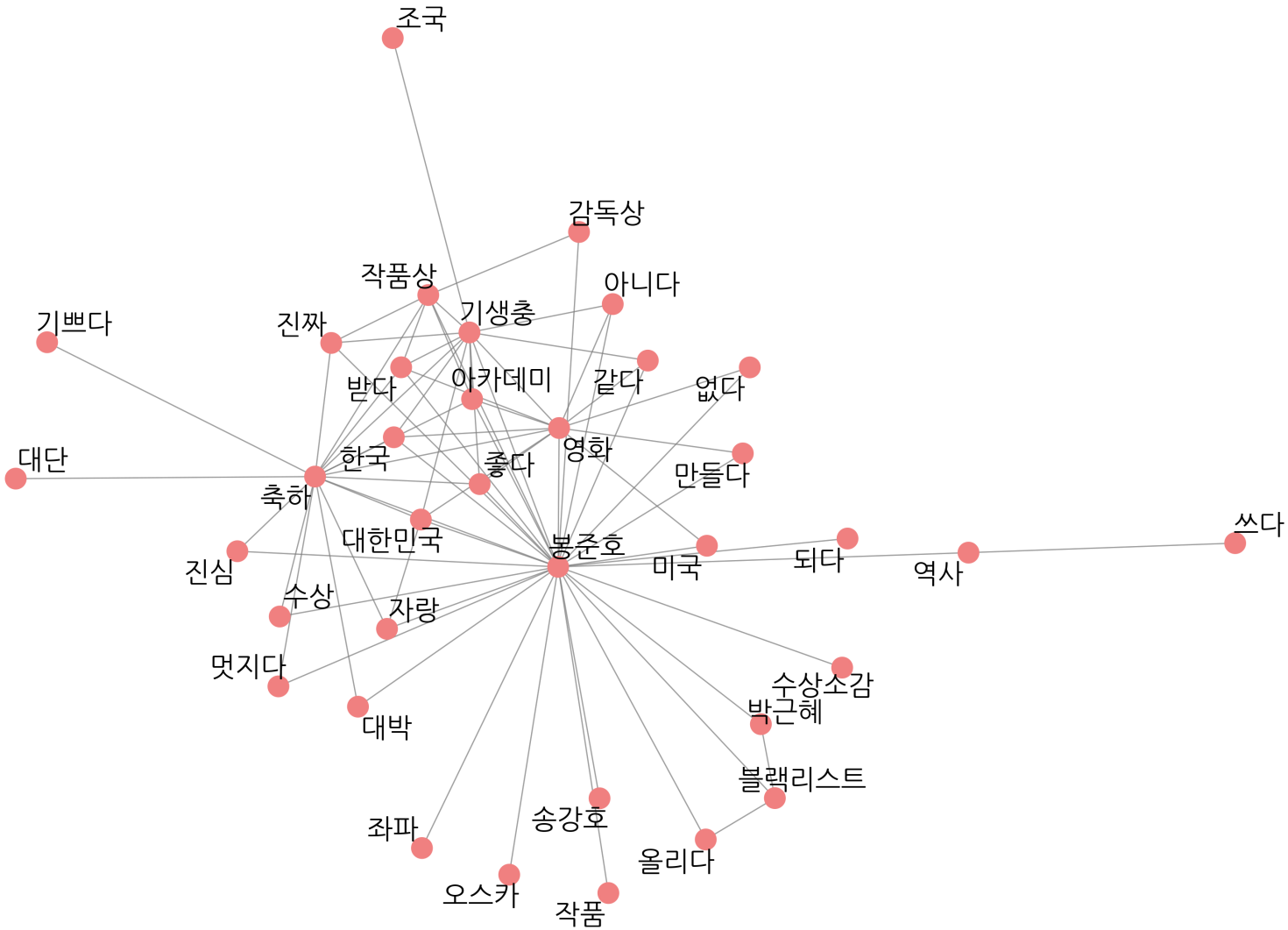
```
word_network <- function(x) {
  ggraph(x, layout = "fr") +
    geom_edge_link(color = "gray50",
                   alpha = 0.5) +
    geom_node_point(color = "lightcoral",
                    size = 5) +
    geom_node_text(aes(label = name),
                   repel = T,
                   size = 5,
                   family = "nanumgothic") +
    theme_graph()
}
```

```
set.seed(1234)
word_network(graph_comment)
```

유의어 처리하기

- 유의어(synonyms): 표현은 다르지만 의미가 비슷한 단어
  - ex) `"감독"`, `"봉감독"`, `"봉준호감독"`
- 유의어 통일하기: 네트워크 구조가 간결해지고 단어의 관계가 좀 더 분명하게 드러남

```r
# 유의어 처리하기
comment <- comment %>%
  mutate(word = ifelse(str_detect(word, "감독") &
                       !str_detect(word, "감독상"), "봉준호", word),
         word = ifelse(word == "오르다", "올리다", word),
         word = ifelse(str_detect(word, "축하"), "축하", word))

# 단어 동시 출현 빈도 구하기
pair <- comment %>%
  pairwise_count(item = word,
                 feature = id,
                 sort = T)

# 네트워크 그래프 데이터 만들기
graph_comment <- pair %>%
  filter(n >= 25) %>%
  as_tbl_graph()

# 네트워크 그래프 만들기
library(ggraph)
set.seed(1234)
word_network(graph_comment)
```


조국
감독상
작품상
아니다
기쁘다
진짜
기생충
받다
아카데미
같다
없다
대단
한국
좋다
영화
만들다
축하
대한민국
봉준호
쓰다
진심
미국
되다
역사
수상
자랑
멋지다
수상소감
박근혜
대박
블랙리스트
좌파
송강호
올리다
오스카
작품

연결 중심성과 커뮤니티 표현하기

- 네트워크 그래프는 단어 노드가 많아 어떤 단어 노드 중심으로 해석할지 판단 어려움
- 연결 중심성과 커뮤니티를 표현하면 단어의 관계를 더 분명하게 파악할 수 있다

연결 중심성(degree centrality)

- 노드가 다른 노드들과 얼마나 밀접하게 연결되는지 나타낸 값
- 연결 중심성으로 노드 크기를 조정하면 어떤 단어를 눈여겨봐야 할지 판단하기 쉬워진다

연결 중심성과 커뮤니티 표현하기

- 네트워크 그래프는 단어 노드가 많아 어떤 단어 노드 중심으로 해석할지 판단 어려움
- 연결 중심성과 커뮤니티를 표현하면 단어의 관계를 더 분명하게 파악할 수 있다

커뮤니티(community)

- 단어 간의 관계가 가까워 빈번하게 연결된 노드 집단
- 노드를 커뮤니티별로 구분 지어 서로 다른 색으로 표현하면 네트워크 구조를 이해하기 쉬워진다

1. 네트워크 그래프 데이터에 연결 중심성, 커뮤니티 변수 추가하기

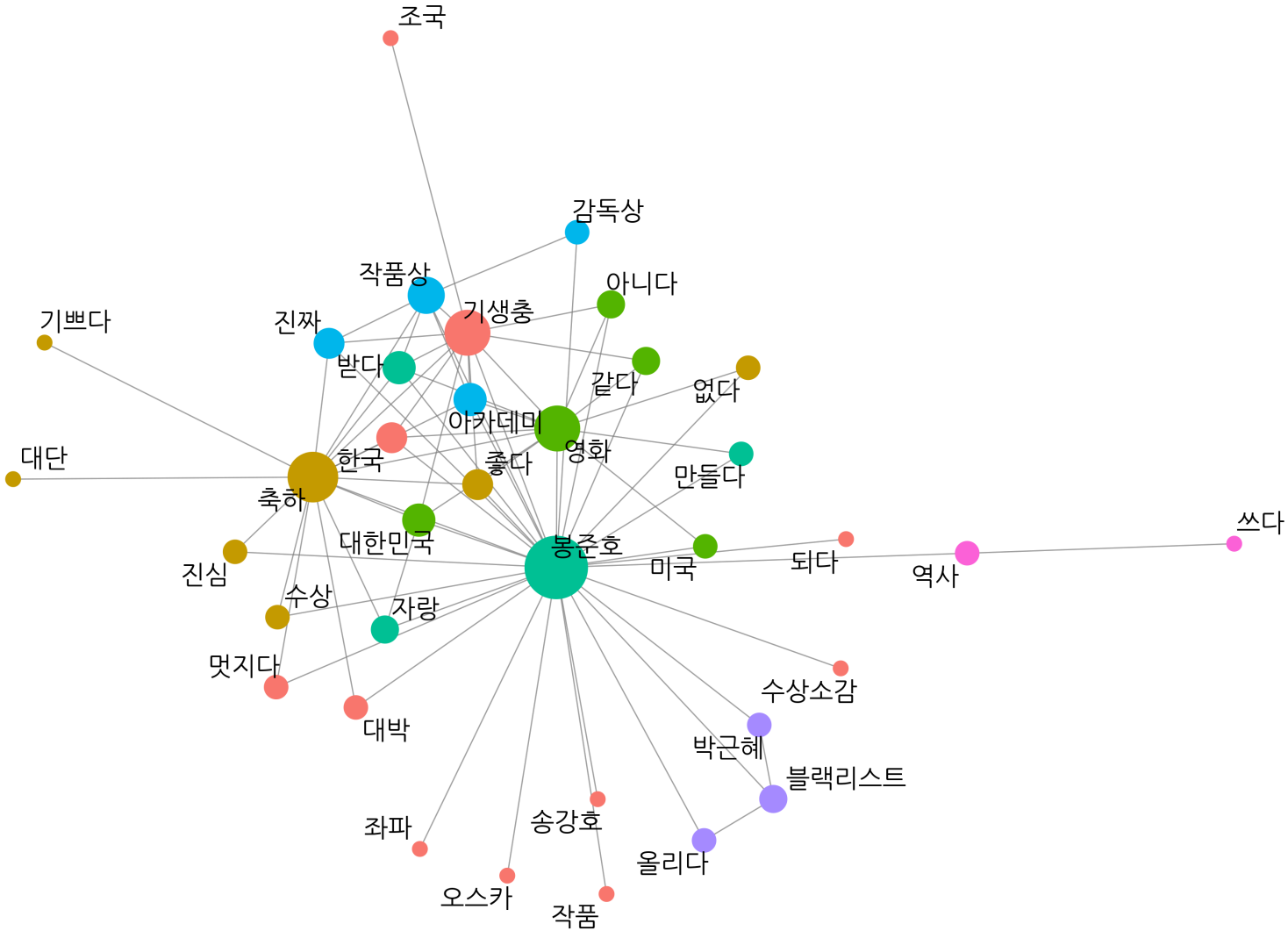
```
set.seed(1234)
graph_comment <- pair %>%
  filter(n >= 25) %>%
  as_tbl_graph(directed = F) %>%
  mutate(centrality = centrality_degree(),    # 연결 중심성
         group = as.factor(group_infomap()))  # 커뮤니티

graph_comment
```

`tidygraph` 패키지의 연결 중심성 지표, 커뮤니티 탐지 알고리즘: tidygraph.data-imaginist.com

```
## # A tbl_graph: 36 nodes and 152 edges
## #
## # An undirected multigraph with 1 component
## #
## # Node Data: 36 x 3 (active)
##   name       centrality group
##   <chr>           <dbl> <fct>
## 1 봉준호             62 4
## 2 축하               34 2
## 3 영화               26 3
## 4 블랙리스트          6 6
## 5 기생충             26 1
## 6 대한민국           10 3
## # … with 30 more rows
## #
## # Edge Data: 152 x 3
##    from    to     n
##   <int> <int> <dbl>
## 1     1     2   198
## 2     1     2   198
## 3     1     3   119
## # … with 149 more rows
```

2. 네트워크 그래프에 연결 중심성, 커뮤니티 표현하기

```
set.seed(1234)
ggraph(graph_comment, layout = "fr") +      # 레이아웃

  geom_edge_link(color = "gray50",          # 엣지 색깔
                 alpha = 0.5) +             # 엣지 명암

  geom_node_point(aes(size = centrality,    # 노드 크기
                      color = group),       # 노드 색깔
                  show.legend = F) +        # 범례 삭제
  scale_size(range = c(5, 15)) +            # 노드 크기 범위

  geom_node_text(aes(label = name),         # 텍스트 표시
                 repel = T,                 # 노드밖 표시
                 size = 5,                  # 텍스트 크기
                 family = "nanumgothic") +  # 폰트

  theme_graph()                             # 배경 삭제
```


조국
감독상
작품상
아니다
기쁘다
진짜
기생충
받다
같다
없다
아카데미
영화
대단
한국
좋다
만들다
축하
대한민국
봉준호
미국
되다
쓰다
역사
진심
수상
자랑
멋지다
수상소감
대박
박근혜
블랙리스트
좌파
송강호
올리다
오스카
작품

03-3 단어 간 상관 분석: Phi coefficient

동시 출현 빈도의 한계

- 대부분의 단어와 자주 함께 사용되는 단어쌍 다수
  - ex) `"영화"-"기생충"`
- 다른 단어에 비해 상대적으로 자주 함께 사용된 단어가 무엇인지 살펴봐야 한다

파이 계수(phi coefficient)

- 두 단어가 함께 사용되는 경우가 각각 사용되는 경우에 비해 얼마나 많은지 나타낸 지표
- 상대적으로 관련성이 큰 단어 파악하는데 활용
  - 어떤 단어와 자주 함께 사용되지만 다른 단어와는 자주 함께 사용되지 않는 단어

파이 계수의 의미

- X, Y 두 단어가 있을 때, 여러 텍스트에서 두 단어의 사용 여부를 놓고 가능한 모든 경우
  - X, Y 모두 있음(a)
  - X, Y 모두 없음(d)
  - X만 있음(b)
  - Y만 있음(c)

	단어 Y 있음	단어 Y 없음	전체
단어 X 있음	a	b	$a+b$
단어 X 없음	c	d	$c+d$
전체	$a+c$	$b+d$	

$$\phi = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$$

파이 계수의 의미

- -1 ~ +1
  - +1에 가까울수록 두 단어가 자주 함께 사용되어 관련성이 크다는 의미
  - -1에 가까울수록 함께 사용되는 경우가 드물어 관련성이 작다는 의미

파이 계수 구하기

- `widyr::pairwise_cor()`
  - `item`: 단어
  - `feature`: 텍스트 구분 기준
  - `sort = T`: 파이 계수 높은순 정렬

```
word_cors <- comment %>%
  add_count(word) %>%
  filter(n >= 20) %>%
  pairwise_cor(item = word,
               feature = id,
               sort = T)

word_cors
```

`add_count()` 원자료에 빈도 나타낸 변수 추가

```
## # A tibble: 26,732 x 3
##    item1      item2      correlation
##    <chr>      <chr>            <dbl>
##  1 올리다     블랙리스트       0.478
##  2 블랙리스트 올리다           0.478
##  3 역사       쓰다             0.370
##  4 쓰다       역사             0.370
##  5 박근혜     블랙리스트       0.322
##  6 블랙리스트 박근혜           0.322
##  7 가족       조국             0.306
##  8 조국       가족             0.306
##  9 작품상     감독상           0.276
## 10 감독상     작품상           0.276
## # … with 26,722 more rows
```

특정 단어와 관련성이 큰 단어 살펴보기

```
word_cors %>%
  filter(item1 == "대한민국")
```

```
## # A tibble: 163 x 3
##    item1    item2  correlation
##    <chr>    <chr>        <dbl>
##  1 대한민국 국민         0.182
##  2 대한민국 자랑         0.158
##  3 대한민국 위상         0.149
##  4 대한민국 국격         0.129
##  5 대한민국 위대한       0.100
##  6 대한민국 세계         0.0910
##  7 대한민국 문화         0.0757
##  8 대한민국 감사합       0.0724
##  9 대한민국 나라         0.0715
## 10 대한민국 오늘         0.0715
## # … with 153 more rows
```

```
word_cors %>%
  filter(item1 == "역사")
```

```
## # A tibble: 163 x 3
##    item1 item2    correlation
##    <chr> <chr>          <dbl>
##  1 역사  쓰다           0.370
##  2 역사  최초           0.117
##  3 역사  한국           0.0982
##  4 역사  순간           0.0910
##  5 역사  한국영화       0.0821
##  6 역사  아니다         0.0774
##  7 역사  감사           0.0654
##  8 역사  영광           0.0640
##  9 역사  영화제         0.0596
## 10 역사  오스카         0.0593
## # … with 153 more rows
```

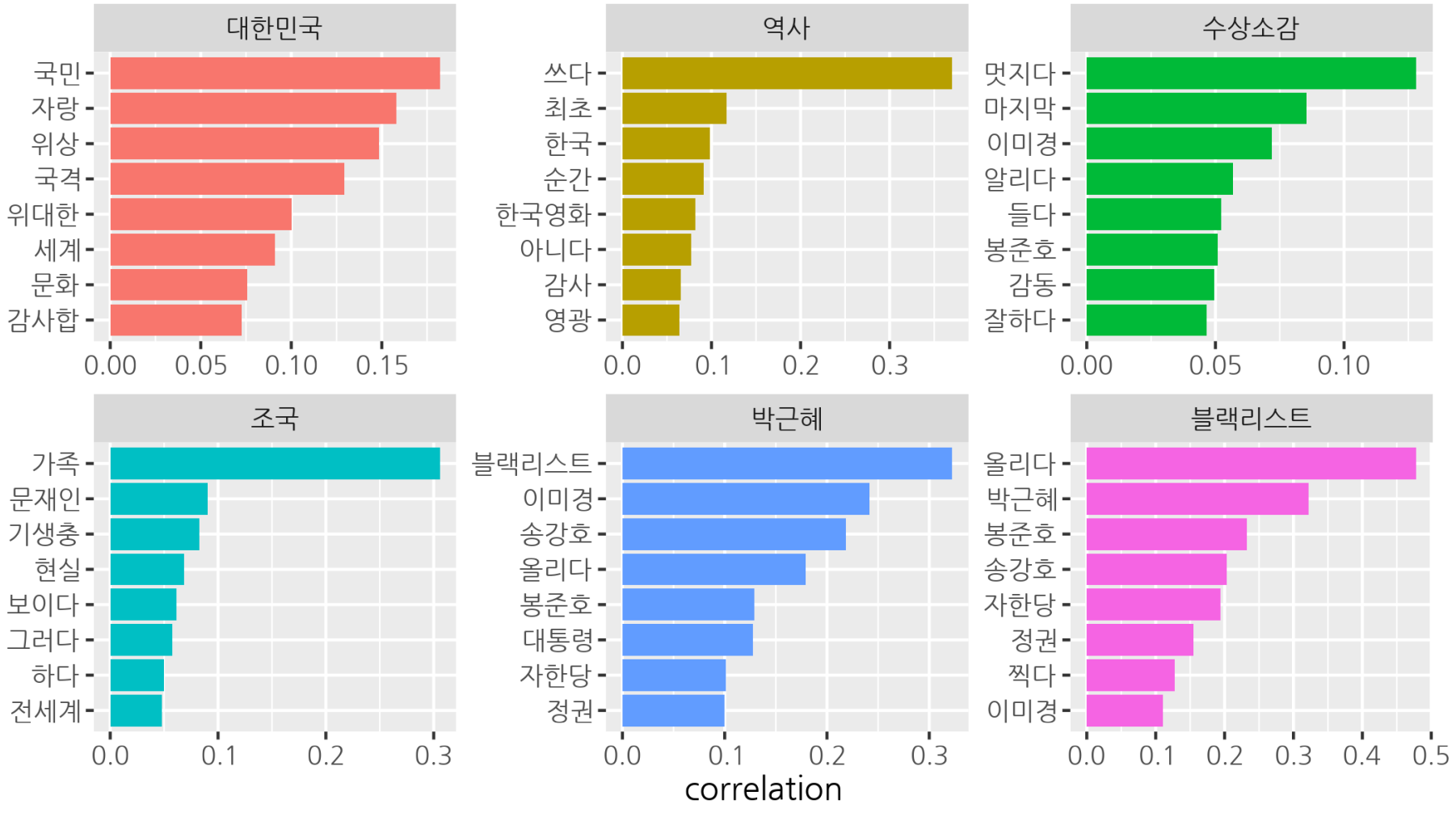
파이 계수로 막대 그래프 만들기

1. 관심 단어별로 파이 계수가 큰 단어 추출하기

```
# 관심 단어 목록 생성
target <- c("대한민국", "역사", "수상소감", "조국", "박근혜", "블랙리스트")

top_cors <- word_cors %>%
  filter(item1 %in% target) %>%
  group_by(item1) %>%
  slice_max(correlation, n = 8)
```

2. 막대 그래프 만들기

```
# 그래프 순서 정하기
top_cors$item1 <- factor(top_cors$item1, levels = target)

library(ggplot2)
ggplot(top_cors, aes(x = reorder_within(item2, correlation, item1),
                     y = correlation,
                     fill = item1)) +
  geom_col(show.legend = F) +
  facet_wrap(~ item1, scales = "free") +
  coord_flip() +
  scale_x_reordered() +
  labs(x = NULL) +
  theme(text = element_text(family = "nanumgothic"))
```


대한민국
국민
자랑
위상
국격
위대한
세계
문화
감사합
0.00
0.05
0.10
0.15
역사
쓰다
최초
한국
순간
한국영화
아니다
감사
영광
0.0
0.1
0.2
0.3
수상소감
멋지다
마지막
이미경
알리다
들다
봉준호
감동
잘하다
0.00
0.05
0.10
조국
가족
문재인
기생충
현실
보이다
그러다
하다
전세계
0.0
0.1
0.2
0.3
박근혜
블랙리스트
이미경
송강호
올리다
봉준호
대통령
자한당
정권
0.0
0.1
0.2
0.3
블랙리스트
올리다
박근혜
봉준호
송강호
자한당
정권
찍다
이미경
0.0
0.1
0.2
0.3
0.4
0.5
correlation

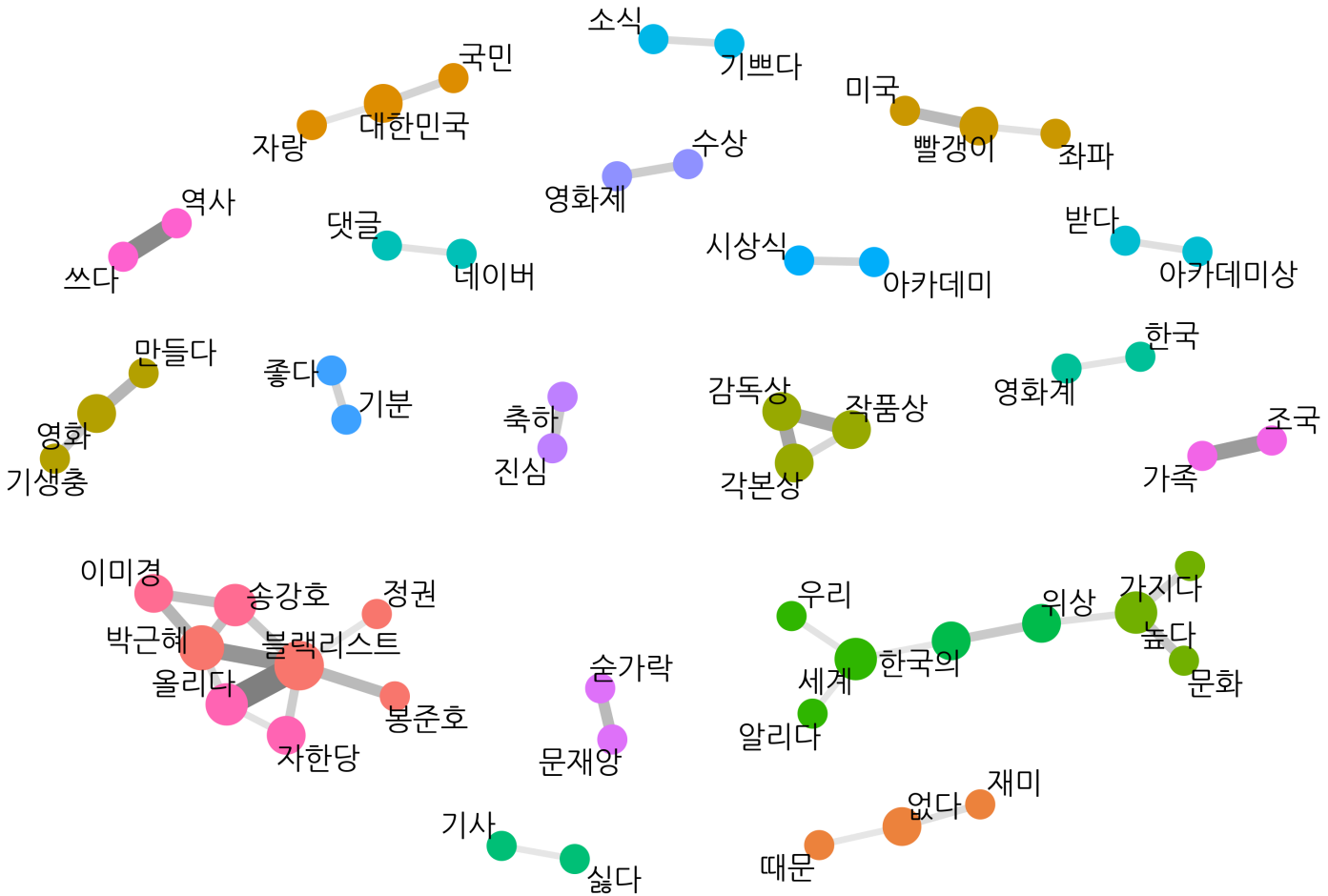
파이 계수로 네트워크 그래프 만들기

1. 네트워크 그래프 데이터 만들기. 연결 중심성과 커뮤니티 추가하기

```
set.seed(1234)
graph_cors <- word_cors %>%
  filter(correlation >= 0.15) %>%
  as_tbl_graph(directed = F) %>%
  mutate(centrality = centrality_degree(),
         group = as.factor(group_infomap()))
```

2. 네트워크 그래프 만들기

```
set.seed(1234)
ggraph(graph_cors, layout = "fr") +

  geom_edge_link(color = "gray50",
                 aes(edge_alpha = correlation,   # 엣지 명암
                     edge_width = correlation),  # 엣지 두께
                 show.legend = F) +              # 범례 삭제
  scale_edge_width(range = c(1, 4)) +            # 엣지 두께 범위

  geom_node_point(aes(size = centrality,
                      color = group),
                  show.legend = F) +
  scale_size(range = c(5, 10)) +

  geom_node_text(aes(label = name),
                 repel = T,
                 size = 5,
                 family = "nanumgothic") +

  theme_graph()
```


소식
기쁘다
국민
대한민국
자랑
미국
빨갱이
좌파
수상
영화제
역사
쓰다
댓글
네이버
시상식
아카데미
받다
아카데미상
한국
영화계
만들다
영화
기생충
좋다
기분
축하
진심
감독상
작품상
각본상
조국
가족
이미경
송강호
정권
박근혜
블랙리스트
올리다
봉준호
자한당
숟가락
문재앙
우리
위상
가지다
높다
한국의
세계
문화
알리다
재미
없다
때문
기사
싫다

03-4 연이어 사용된 단어쌍 분석: n-gram

- 같은 단어도 함께 사용된 단어에 따라 의미가 달라짐
- 어떤 단어는 다른 단어와 연결되어 새로운 의미를 만들어냄
  - ex)
    - '사과를 먹다', '사과를 하다'
    - '감을 잡다', '귀가 얇다'
- 동시 출현 빈도와 파이 계수의 한계: 단어가 함께 사용된 횟수만 고려
  - 단어가 연결될 때 생기는 의미 무시
  - 이해하기 어려운 단어쌍 등장

- 단어가 연결될 때 생기는 의미를 고려하려면 **'자주 연이어 사용된 단어'**를 살펴봐야 한다

엔그램(n-gram)

- 연이어 사용된 n개의 단어
  - 두 단어 연속: 바이그램(bigram) 또는 2-gram
  - 세 단어 연속: 트라이그램(trigram) 또는 3-gram

- **텍스트를 엔그램으로 토큰화하면**
  - 단어 앞뒤에 연이어 사용된 단어를 함께 살펴봄: 얼마나 자주 '연이어' 사용된 단어쌍인가?
  - 단어가 연결될 때 생기는 의미와 맥락을 이해할 수 있음
  - 대다수의 텍스트에 사용된 평범한 단어쌍이 아니라 분명한 의미를 드러내는 단어쌍 발견

엔그램으로 토큰화하기

샘플 텍스트로 엔그램 토큰화해보기

- `tidytext::unnest_tokens()`
  - `token = "ngrams"`
  - `n`: 기준 단어 수

```
text <- tibble(value = "대한민국은 민주공화국이다. 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.")

text
```

```
## # A tibble: 1 x 1
##   value
##   <chr>
## 1 대한민국은 민주공화국이다. 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
```

```r
# 바이그램 토큰화
text %>%
  unnest_tokens(input = value,
                output = word,
                token = "ngrams",
                n = 2)
```

```
## # A tibble: 9 x 1
##   word
##   <chr>
## 1 대한민국은 민주공화국이다
## 2 민주공화국이다 대한민국의
## 3 대한민국의 주권은
## 4 주권은 국민에게
## 5 국민에게 있고
## 6 있고 모든
## 7 모든 권력은
## 8 권력은 국민으로부터
## 9 국민으로부터 나온다
```

```r
# 트라이그램 토큰화
text %>%
  unnest_tokens(input = value,
                output = word,
                token = "ngrams",
                n = 3)
```

```
## # A tibble: 8 x 1
##   word
##   <chr>
## 1 대한민국은 민주공화국이다 대한민국의
## 2 민주공화국이다 대한민국의 주권은
## 3 대한민국의 주권은 국민에게
## 4 주권은 국민에게 있고
## 5 국민에게 있고 모든
## 6 있고 모든 권력은
## 7 모든 권력은 국민으로부터
## 8 권력은 국민으로부터 나온다
```

기사 댓글로 바이그램 만들기

(1) 명사, 동사, 형용사 추출하기

- `comment_pos` 이용: 댓글을 형태소로 토큰화 후 품사별로 행 분리
- 명사, 동사, 형용사를 추출해 결합한 후 두 글자 이상만 남김

```
comment_new <- comment_pos %>%
  separate_rows(word, sep = "[+]") %>%
  filter(str_detect(word, "/n|/pv|/pa")) %>%
  mutate(word = ifelse(str_detect(word, "/pv|/pa"),
                       str_replace(word, "/.*$", "다"),
                       str_remove(word, "/.*$"))) %>%
  filter(str_count(word) >= 2) %>%
  arrange(id)
```

(2) 유의어 처리하기

```
comment_new <- comment_new %>%
  mutate(word = ifelse(str_detect(word, "감독") &
                       !str_detect(word, "감독상"), "봉준호", word),
         word = ifelse(word  == "오르다", "올리다", word),
         word = ifelse(str_detect(word, "축하"), "축하", word))
```

(3) 한 댓글이 하나의 행이 되도록 결합하기

```
comment_new %>%
  select(word)
```

```
## # A tibble: 26,860 x 1
##    word
##    <chr>
##  1 우리
##  2 좋다
##  3 생기다
##  4 기쁘다
##  5 행복한
##  6 행복
##  7 축하
##  8 행복
##  9 기쁘다
## 10 기쁘다
## # … with 26,850 more rows
```

(3) 한 댓글이 하나의 행이 되도록 결합하기

```
line_comment <- comment_new %>%
  group_by(id) %>%
  summarise(sentence = paste(word, collapse = " "))

line_comment
```

```
## # A tibble: 4,007 x 2
##       id sentence
##  * <int> <chr>
##  1     1 우리 좋다 생기다 기쁘다 행복한 행복 축하 행복 기쁘다
##  2     2 기쁘다 시국 기쁘다 감사하다 축하 진심
##  3     3 우리나라 봉준호 불다 크다 영감 봉준호 공동각본쓴 한진 작가님 축하 축하 드리다…
##  4     4 봉준호 봉준호 우리나라 대한민국 자랑 세계 어디 우리 한국인 힘내다 삽시…
##  5     5 노벨상 탄느낌이네요 축하
##  6     6 기생충 받다 박수 치다 감독상 기대다 봉준호 봉준호
##  7     7 대한민국 영화사 쓰다 계시다
##  8     8 아카데미상 받다 태극기 휘날리다 광해 명량 전부문 휩쓸어야겠
##  9     9 다시한번 보이다 영화관
## 10    10 대한민국 봉준호 대단 한국의 문화 자긍심 가지
## # … with 3,997 more rows
```

(4) 바이그램으로 토큰화하기

```
bigram_comment <- line_comment %>%
  unnest_tokens(input = sentence,
                output = bigram,
                token = "ngrams",
                n = 2)

bigram_comment
```

```
## # A tibble: 23,348 x 2
##       id bigram
##    <int> <chr>
##  1     1 우리 좋다
##  2     1 좋다 생기다
##  3     1 생기다 기쁘다
##  4     1 기쁘다 행복한
##  5     1 행복한 행복
##  6     1 행복 축하
##  7     1 축하 행복
##  8     1 행복 기쁘다
##  9     2 기쁘다 시국
## 10     2 시국 기쁘다
## # … with 23,338 more rows
```

연이어 사용된 단어쌍 빈도 구하기

1. 바이그램 분리하기

```
# 바이그램 분리하기
library(tidyr)
bigram_seprated <- bigram_comment %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigram_seprated
```

```
## # A tibble: 23,348 x 3
##       id word1  word2
##    <int> <chr>  <chr>
##  1     1 우리   좋다
##  2     1 좋다   생기다
##  3     1 생기다 기쁘다
##  4     1 기쁘다 행복한
##  5     1 행복한 행복
##  6     1 행복   축하
##  7     1 축하   행복
##  8     1 행복   기쁘다
##  9     2 기쁘다 시국
## 10     2 시국   기쁘다
```

2. 단어쌍 빈도 구하기

```
# 단어쌍 빈도 구하기
pair_bigram <- bigram_seprated %>%
  count(word1, word2, sort = T) %>%
  na.omit()

pair_bigram
```

```
## # A tibble: 19,030 x 3
##    word1      word2          n
##    <chr>      <chr>      <int>
##  1 봉준호     봉준호       155
##  2 블랙리스트 올리다        64
##  3 진심       축하          64
##  4 봉준호     축하          57
##  5 봉준호     송강호        34
##  6 영화       만들다        31
##  7 축하       봉준호        31
##  8 대단       축하          27
##  9 봉준호     블랙리스트    27
## 10 대박       축하          26
## # … with 19,020 more rows
```

`na.omit()`: 결측치 행 제거: 한 단어로 된 문장은 바이그램으로 토큰화하면 `NA`가 됨
ex) '축하합니다', '멋집니다'

3. 단어쌍 살펴보기

```
# 동시 출현 단어쌍
pair %>%
  filter(item1 == "대한민국")
```

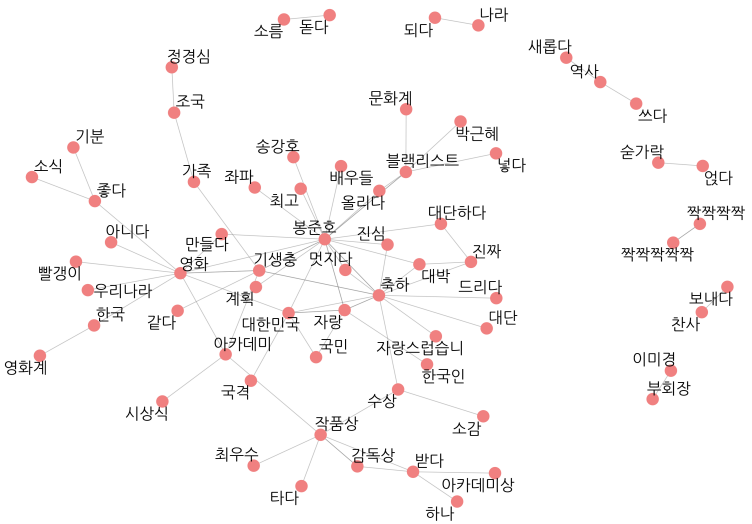
```
## # A tibble: 1,010 x 3
##    item1    item2        n
##    <chr>    <chr>    <dbl>
##  1 대한민국 봉준호      70
##  2 대한민국 축하        54
##  3 대한민국 자랑        44
##  4 대한민국 영화        30
##  5 대한민국 기생충      27
##  6 대한민국 국민        22
##  7 대한민국 세계        16
##  8 대한민국 아카데미    16
##  9 대한민국 위상        15
## 10 대한민국 좋다        14
## # … with 1,000 more rows
```

```
# 바이그램 단어쌍
pair_bigram %>%
  filter(word1 == "대한민국")
```

```
## # A tibble: 109 x 3
##    word1    word2      n
##    <chr>    <chr>  <int>
##  1 대한민국 국민      21
##  2 대한민국 자랑      15
##  3 대한민국 영화      11
##  4 대한민국 국격       8
##  5 대한민국 위상       7
##  6 대한민국 만세       6
##  7 대한민국 봉준호     5
##  8 대한민국 문화       4
##  9 대한민국 영광       4
## 10 대한민국 기생충     3
## # … with 99 more rows
```

엔그램으로 네트워크 그래프 만들기

```
# 네트워크 그래프 데이터 만들기
graph_bigram <- pair_bigram %>%
  filter(n >= 8) %>%
  as_tbl_graph()

# 네트워크 그래프 만들기
set.seed(1234)
word_network(graph_bigram)
```




소름
돋다
되다
나라
새롭다
역사
쓰다
정경심
조국
문화계
박근혜
기분
송강호
블랙리스트
넣다
숟가락
얹다
소식
가족
좌파
배우들
좋다
최고
올리다
짝짝짝짝
대단하다
아니다
봉준호
진심
짝짝짝짝짝
만들다
진짜
빨갱이
영화
기생충
멋지다
대박
우리나라
축하
드리다
보내다
계획
한국
찬사
같다
대한민국
자랑
대단
아카데미
국민
자랑스럽습니
이미경
영화계
한국인
부회장
국격
수상
시상식
작품상
소감
최우수
감독상
받다
아카데미상
타다
하나

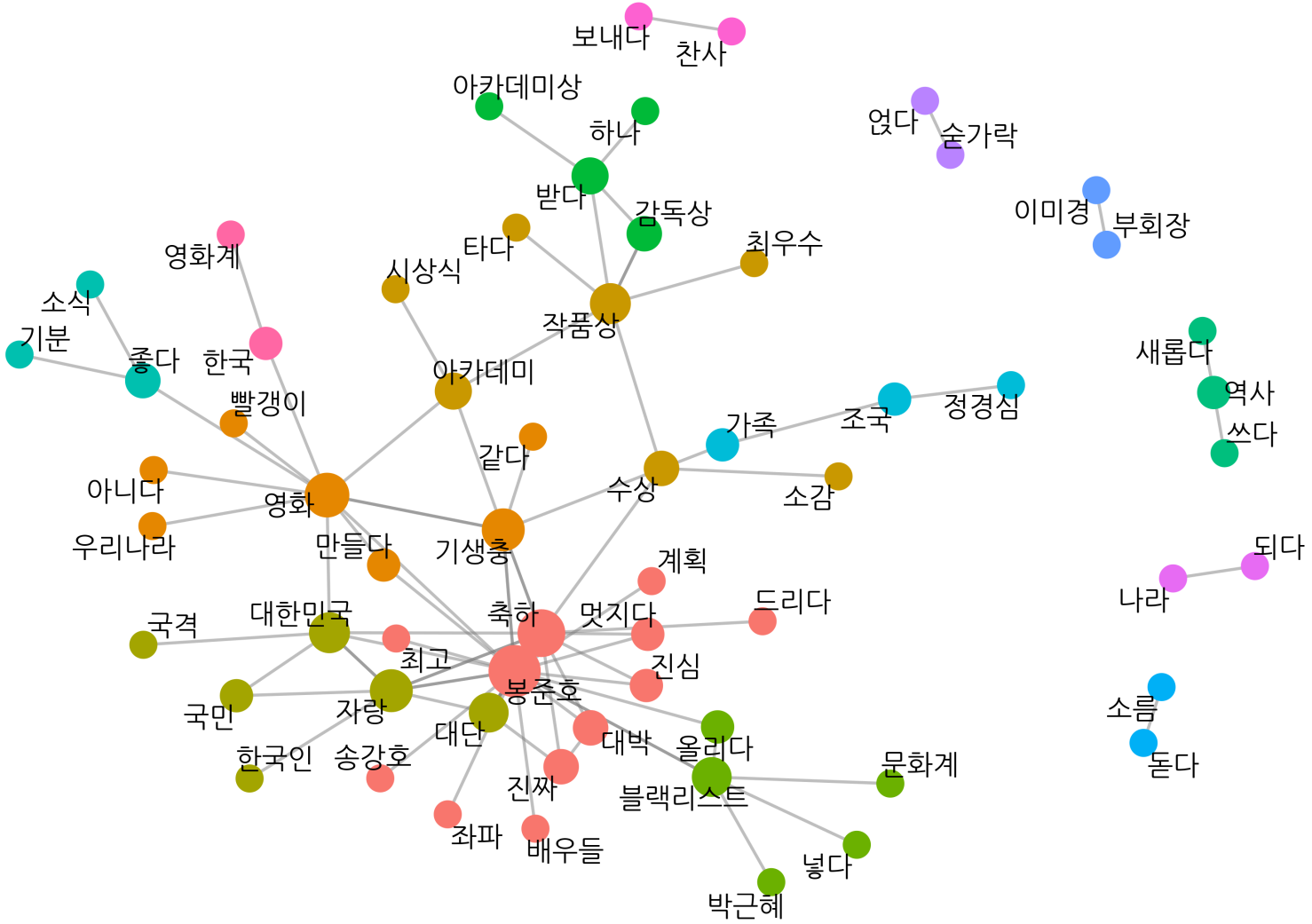
유의어 통일하고 네트워크 그래프 다시 만들기

- `bigram_seprated`의 유의어 통일, 같은 단어 연속 단어쌍 제거
- 단어쌍 빈도 구하고 결측치 제거

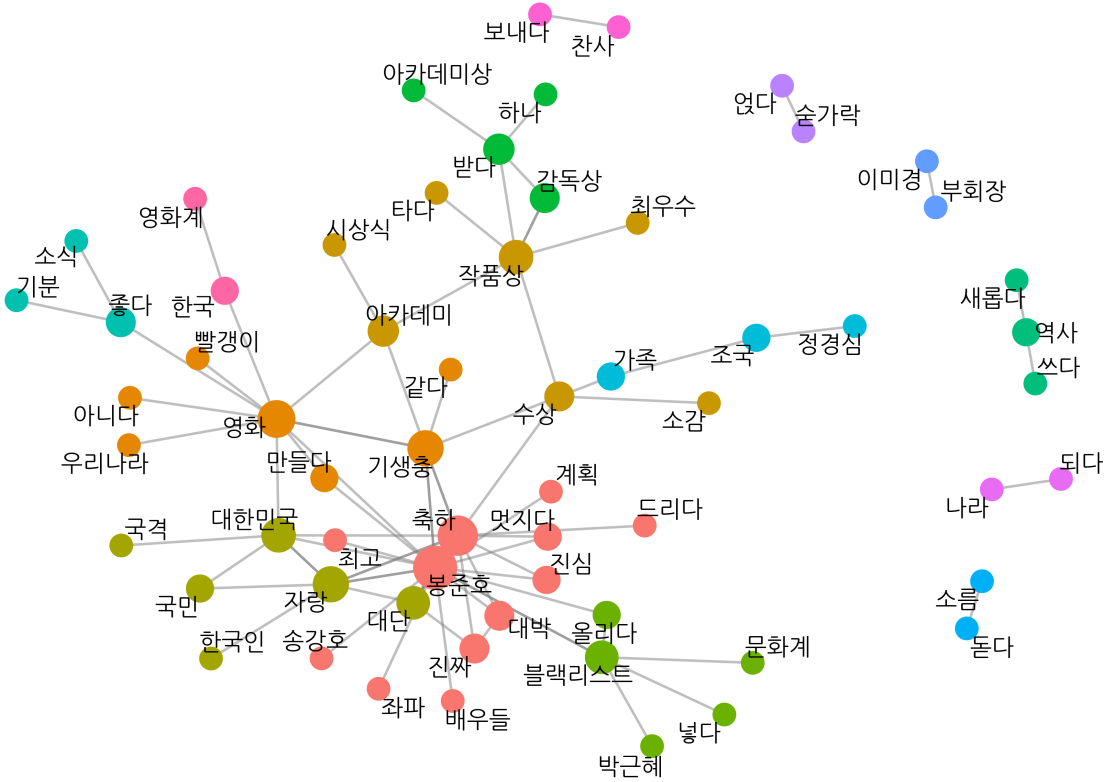
```
# 유의어 처리
bigram_seprated <- bigram_seprated %>%
  mutate(word1 = ifelse(str_detect(word1, "대단"), "대단", word1),
         word2 = ifelse(str_detect(word2, "대단"), "대단", word2),

         word1 = ifelse(str_detect(word1, "자랑"), "자랑", word1),
         word2 = ifelse(str_detect(word2, "자랑"), "자랑", word2),

         word1 = ifelse(str_detect(word1, "짝짝짝"), "짝짝짝", word1),
         word2 = ifelse(str_detect(word2, "짝짝짝"), "짝짝짝", word2)) %>%

  # 같은 단어 연속 제거
  filter(word1 != word2)

# 단어쌍 빈도 구하기
pair_bigram <- bigram_seprated %>%
  count(word1, word2, sort = T) %>%
  na.omit()
```

```r
# 네트워크 그래프 데이터 만들기
set.seed(1234)
graph_bigram <- pair_bigram %>%
  filter(n >= 8) %>%
  as_tbl_graph(directed = F) %>%
  mutate(centrality = centrality_degree(),    # 중심성
         group = as.factor(group_infomap()))  # 커뮤니티
```

```r
# 네트워크 그래프 만들기
set.seed(1234)
ggraph(graph_bigram, layout = "fr") +          # 레이아웃

  geom_edge_link(color = "gray50",             # 엣지 색깔
                 alpha = 0.5) +                # 엣지 명암

  geom_node_point(aes(size = centrality,       # 노드 크기
                      color = group),          # 노드 색깔
                  show.legend = F) +           # 범례 삭제
  scale_size(range = c(4, 8)) +                # 노드 크기 범위

  geom_node_text(aes(label = name),            # 텍스트 표시
                 repel = T,                    # 노드밖 표시
                 size = 5,                     # 텍스트 크기
                 family = "nanumgothic") +     # 폰트

  theme_graph()                                # 배경 삭제
```

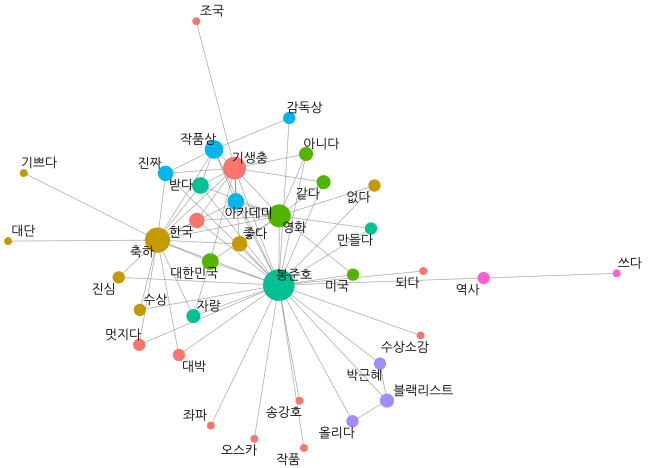

보내다
찬사
아카데미상
하나
받다
감독상
타다
시상식
최우수
작품상
영화계
소식
기분
좋다
한국
빨갱이
아카데미
아니다
영화
우리나라
만들다
같다
기생충
수상
가족
조국
정경심
소감
얹다
숟가락
이미경
부회장
새롭다
역사
쓰다
되다
나라
계획
드리다
대한민국
국격
축하
멋지다
최고
진심
봉준호
국민
자랑
대단
대박
올리다
한국인
송강호
진짜
블랙리스트
문화계
좌파
배우들
넣다
박근혜
소름
돋다

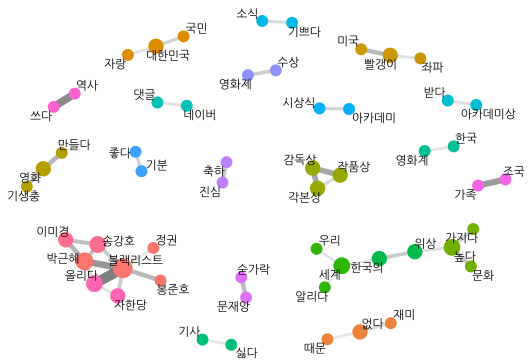
- 자주 연이어 사용된 단어쌍 중심으로 네트워크 형성
- 단어의 맥락과 의미를 구체적으로 이해할 수 있음
- 개별 단어의 빈도는 낮지만 자주 연이어 사용되고 함께 사용할 때 분명한 의미 지니는 단어쌍 발견
  - ex) '이미경-부회장', '조국-가족'

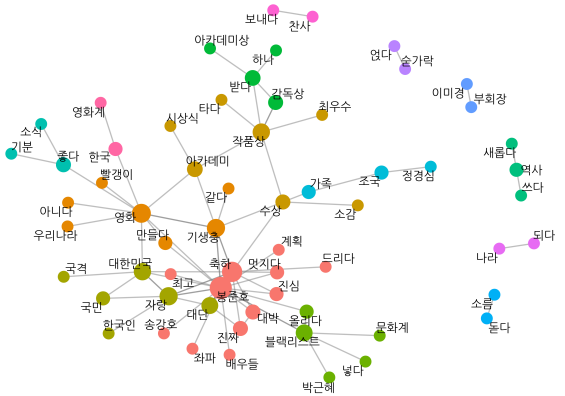


어떤 방법으로 네트워크 그래프를 만드는 게 좋을까

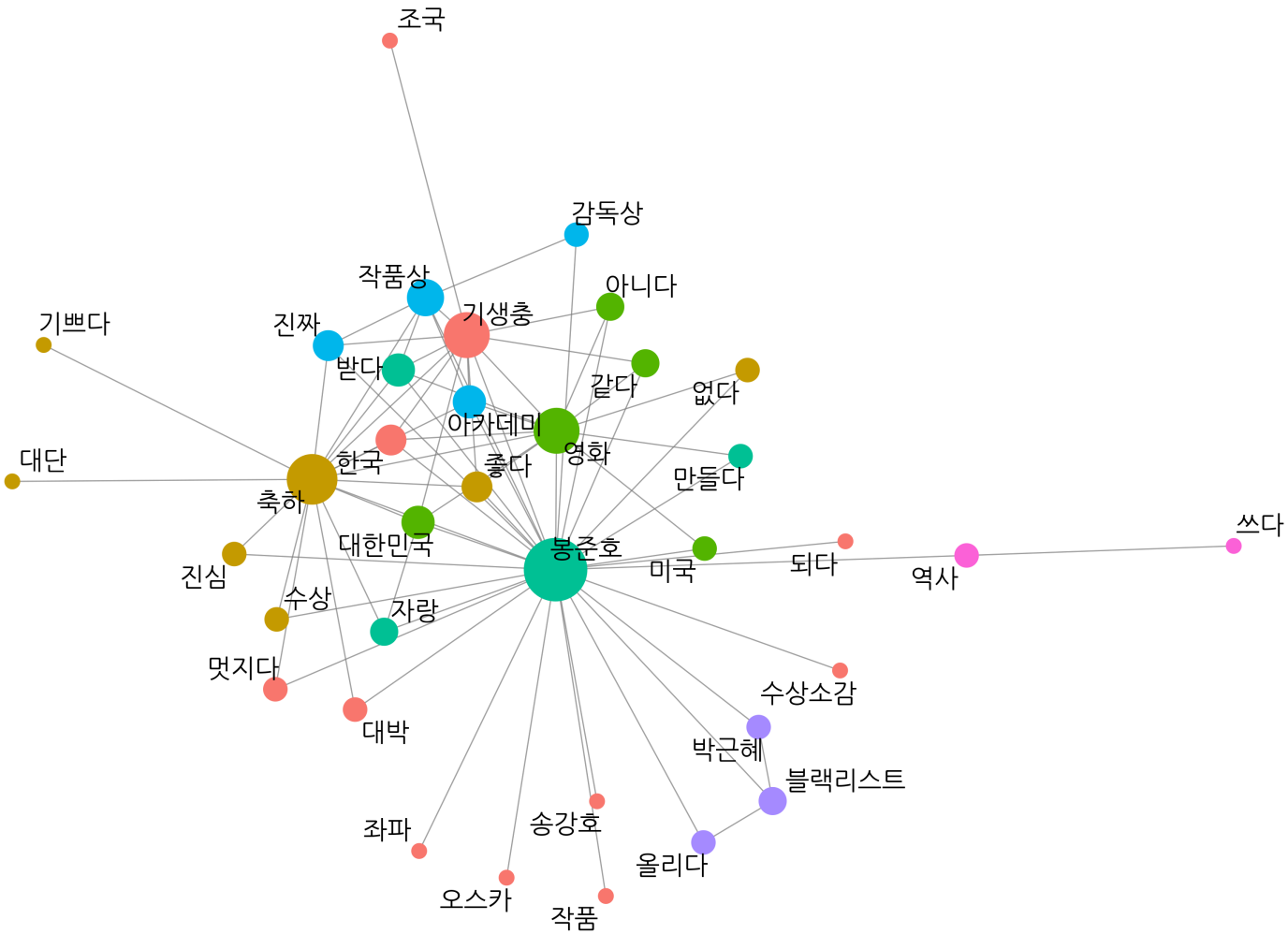
- 각 방법의 특징 다르므로 분석 목적에 맞게 선택
- 세 가지 방법 모두 사용해 분석 결과 비교하면 텍스트를 다각도로 이해할 수 있음



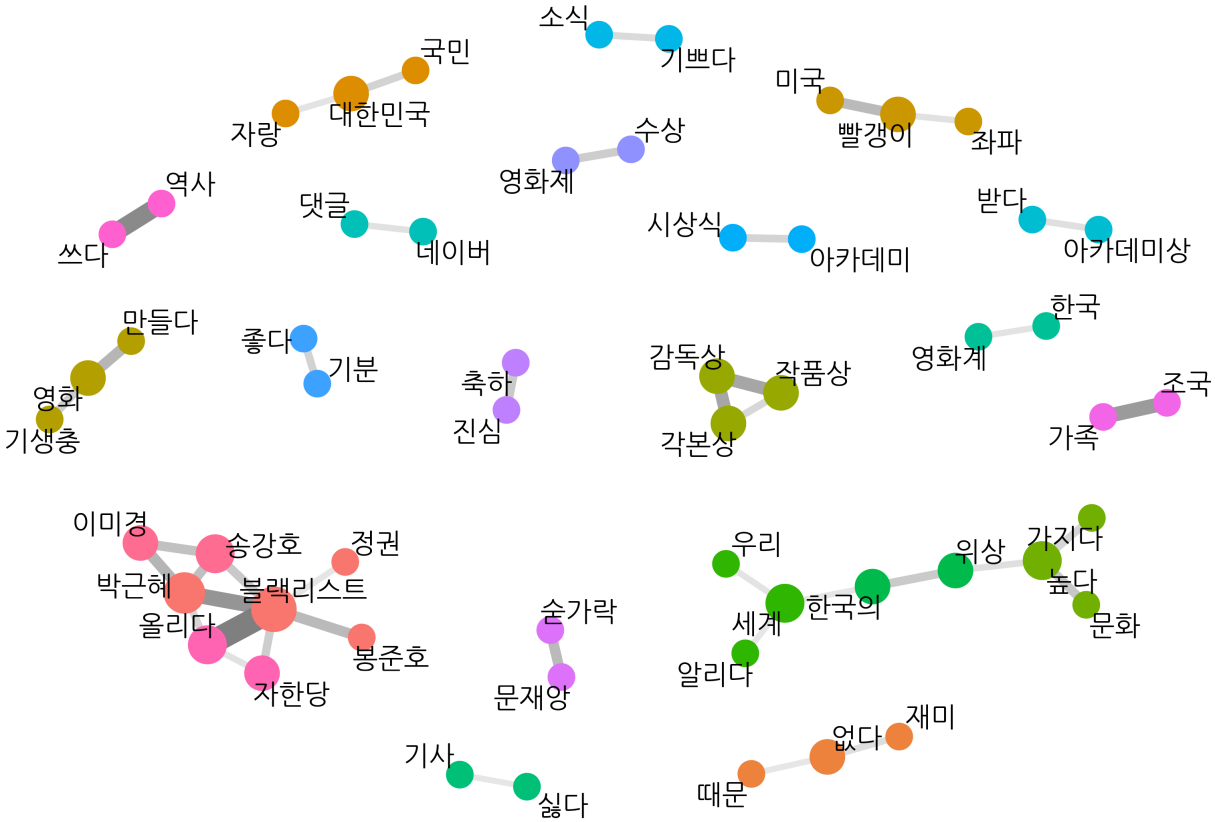




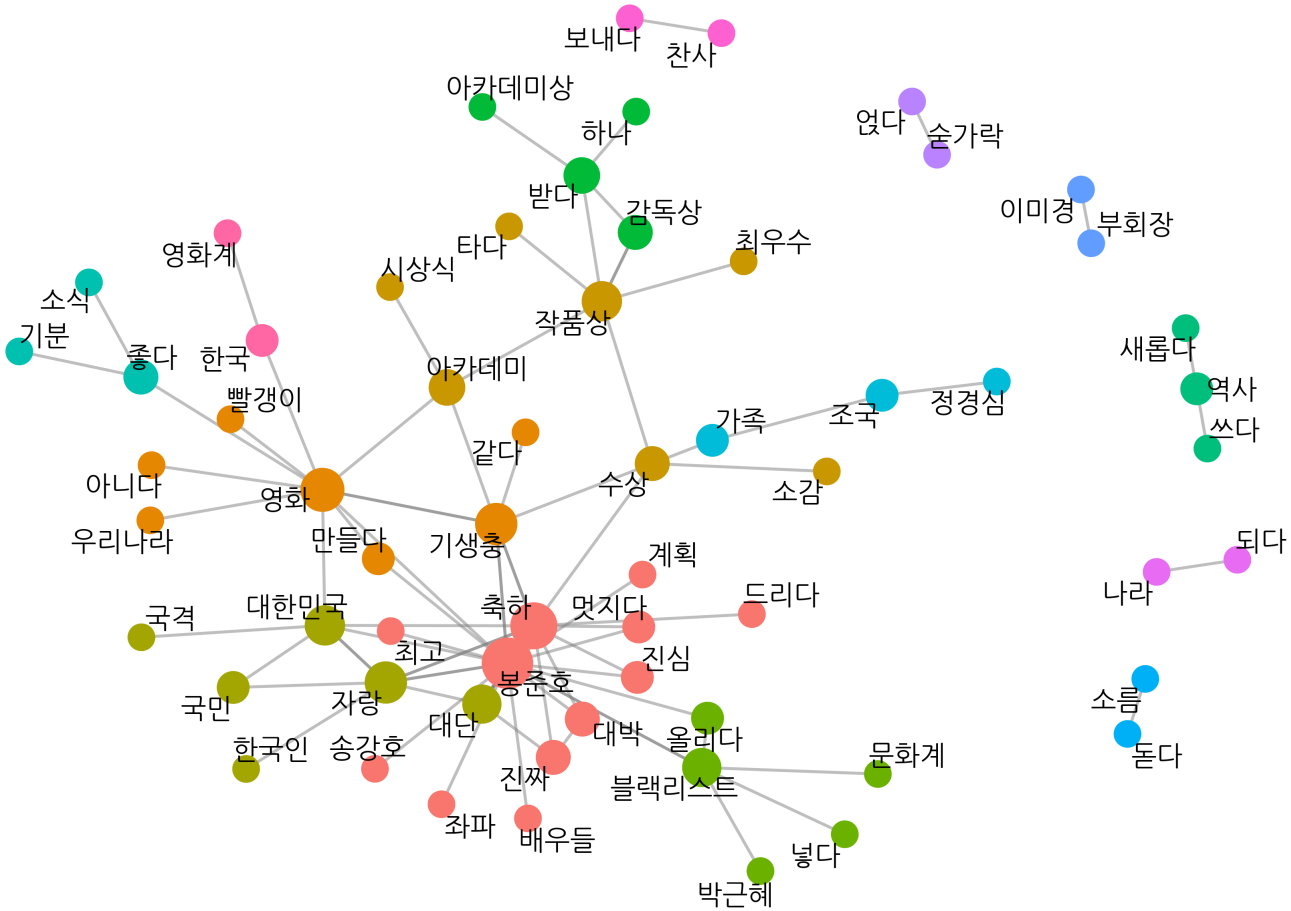
- **동시 출현 빈도**: 자주 사용된 단어 중심으로 단어들의 관계 표현



- **파이 계수**
  - 관련성이 큰 단어쌍 중심으로 표현
  - 단어 군집을 잘 드러내고 싶을 때

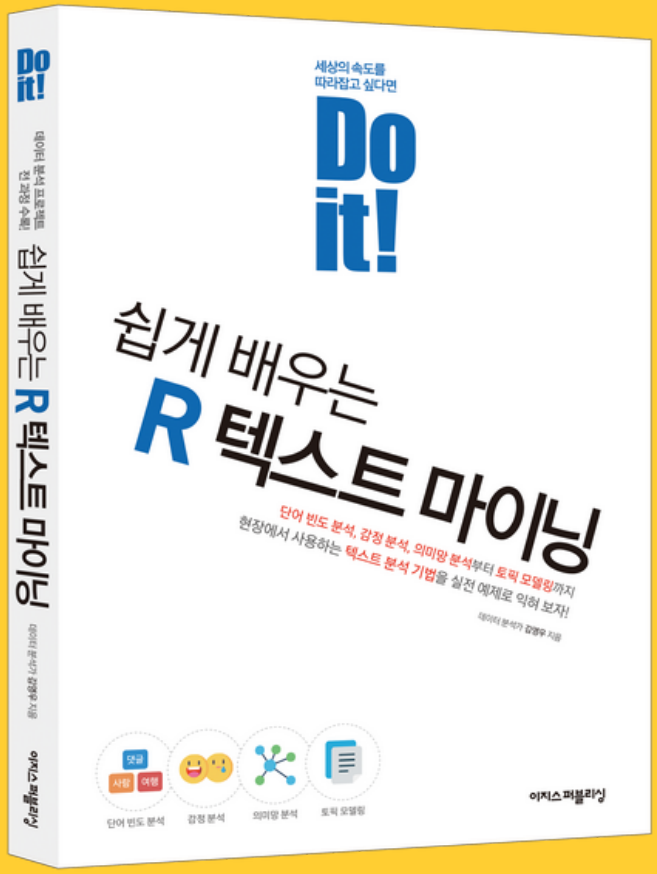


- **엔그램**
  - 연이어 사용될 때 의미를 지니는 단어쌍 중심으로 표현
  - 단어들이 전반적으로 어떤 관계를 형성하고 있는지 표현할 때



Q&A

Quiz


Do it!
데이터 분석 프로젝트 전 과정 수록!
쉽게 배우는 R 텍스트 마이닝
데이터 분석가 김영우 지음
이지스퍼블리싱
세상의 속도를 따라잡고 싶다면
Do it!
쉽게 배우는
R 텍스트 마이닝
단어 빈도 분석, 감정 분석, 의미망 분석부터 토픽 모델링까지
현장에서 사용하는 텍스트 분석 기법을 실전 예제로 익혀 보자!
데이터 분석가 김영우 지음
댓글
사람
여행
단어 빈도 분석
감정 분석
의미망 분석
토픽 모델링
이지스퍼블리싱

stats7445@gmail.com