Data Mining practice4 team2

22000283	박예준
22000245	문벼리
22001030	신혜리

Task 1-1 Read the entire text f iles and organize it in the following format:

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

tidytext를 설치 후 라이브러리를 이용해 텍스트 마이닝을 시작한다

```
bible1 <- readLines("C:/Users/silkj/Desktop/한동대학교/5학기/데이터 마이닝 실습/Data-Mining-Practicum/myR/NIV_English_Bible/01-Genesis.txt")

bible <- data_frame(line =1:4, text = bible1)
# 'bible1'이 1533줄이므로 이에 맞게 line 열 생성
bible <- tibble(line = 1:length(bible1), text = bible1)

# 결과 확인
print(bible)
```

```
   line text
  <int> <chr>
1     1 "Genesis"
2     2 "1:1 In the beginning God created the heavens and the earth."
3     3 "1:2 Now the earth was formless and empty, darkness was over the surface of the deep, and the …
4     4 "1:3 And God said, \"Let there be light,\" and there was light."
5     5 "1:4 God saw that the light was good, and he separated the light from the darkness."
6     6 "1:5 God called the light \"day,\" and the darkness he called \"night.\" And there was evening…
7     7 "1:6 And God said, \"Let there be an expanse between the waters to separate water from water.\…
8     8 "1:7 So God made the expanse and separated the water under the expanse from the water above it…
9     9 "1:8 God called the expanse \"sky.\" And there was evening, and there was morning--the second …
10   10 "1:9 And God said, \"Let the water under the sky be gathered to one place, and let dry ground …
```

가장 먼저 readLines를 사용하여 창세기의 text를 로드한 후, 데이터 프레임으로 변환함

맨 처음 data_frame으로 라인을 정했으나 로드한 창세기가 4줄이 아니므로 오류, tibble을 사용하여 bible1의 길이만큼 라인을 정함

```
# 책 이름은 첫 번째 행에 있음
book_name <- bible$text[1]

# 나머지 줄 (Chapter:Verse와 Script 포함) 추출
script_lines <- bible$text[-1]

# 데이터프레임을 만들기 위한 빈 벡터 초기화
chapter <- c()
verse <- c()
script <- c()
```

text의 가장 첫 번째 행의 Genesis 를 book_name으로 따로 빼고 나머지를 script_lines로 할당 다음으로 각각을 데이터프레임으로 만들기 위한 벡터를 생성함

```
# 각 줄을 분석해서 "Chapter", "Verse", "Script"로 분리
for (line in script_lines) {
  # 정규식을 사용해 Chapter:Verse와 나머지 Script 부분 분리
  match <- regexpr("^(\\d+):(\\d+)\\s(.+)$", line, perl=TRU
E)

  # 패턴이 맞는 경우에만 처리
  if (match[1] != -1) {
    # Chapter:Verse 추출
    chapter_verse <- regmatches(line, regexpr("^(\\d+):(\\d
+)", line))
    chapter_verse_split <- strsplit(chapter_verse, ":")
[[1]]

    # Chapter와 Verse로 나누기
    chapter <- c(chapter, as.numeric(chapter_verse_split
[1]))
    verse <- c(verse, as.numeric(chapter_verse_split[2]))
```

```
    # Script 추출 (정규식으로 나머지 텍스트 추출)
    script <- c(script, regmatches(line, regexpr("\\s(.+)
$", line)))
  }
}
```

정규식을 사용해 장:절 script로 분리함

^ → 시작신호

(\\d+) →숫자형 데이터를 분리 +로 하나 이상의 숫자를 할당받음

: chapter:Verse 를 구분하는 :를 매칭

\\s(.+)$ → 공백을 매칭한 후 하나 이상의 모든 문자를 매칭함

일련의 과정을 통해 기존의 챕터:절 script로 구성된 문장이 정리됨

이후 해당 패턴이 맞는 경우 다시 챕터와 절로 구분한 후 script를 추출함

```
# 데이터프레임 생성
bible_df <- tibble(
  Book = book_name,
  Chapter = chapter,
  Verse = verse,
  Script = script
)

# 결과 확인
print(bible_df)
```

마지막으로 각 벡터에 저장된 내용들을 가지고 데이터 프레임을 생성

	Book	Chapter	Verse	Script
	<chr>	<dbl>	<dbl>	<chr>
1	Genesis	1	1	" In the beginning God created the heavens and the earth."
2	Genesis	1	2	" Now the earth was formless and empty, darkness was over the surface of the d…
3	Genesis	1	3	" And God said, \"Let there be light,\" and there was light."
4	Genesis	1	4	" God saw that the light was good, and he separated the light from the darknes…
5	Genesis	1	5	" God called the light \"day,\" and the darkness he called \"night.\" And ther…
6	Genesis	1	6	" And God said, \"Let there be an expanse between the waters to separate water…
7	Genesis	1	7	" So God made the expanse and separated the water under the expanse from the w…
8	Genesis	1	8	" God called the expanse \"sky.\" And there was evening, and there was morning…
9	Genesis	1	9	" And God said, \"Let the water under the sky be gathered to one place, and le…
10	Genesis	1	10	" God called the dry ground \"land,\" and the gathered waters he called \"seas…

제대로 만들어 졌으나, 이런 과정을 구약,신양 성경 모두에 적용하려고 하면 너무 복잡하고 시간도 오래걸린다. 그렇기 때문에 이번에는 해당 text가 저장된 폴더 전체를 다운받은 후 한 번에 데이터프레임으로 만드는 방법을 사용하려고 한다.

```
folder_path <- "C:/Users/silkj/Desktop/한동대학교/5학기/데이터
마이닝 실습/Data-Mining-Practicum/myR/NIV_English_Bible/"

# 폴더 내의 모든 텍스트 파일을 리스트로 가져오기
file_list <- list.files(folder_path, pattern = "*.txt", ful
l.names = TRUE)
```

우선 파일 경로를 txt파일이 있는 폴더로 지정한 후, list.files로 폴더 내 텍스트 파일을 리스트로 가져온다.

```
# 빈 리스트 생성
bible_texts <- list()

# 파일 리스트를 반복하면서 파일 읽기
for (file in file_list) {
  # 파일 내용 읽기
  bible_content <- readLines(file, encoding = "UTF-8")

  # 파일 이름(책 이름) 추출
  book_name <- tools::file_path_sans_ext(basename(file))  #
확장자 제거

  # 리스트에 저장
  bible_texts[[book_name]] <- bible_content
}
```

이후 빈 리스트를 생성한 후, 파일들을 LeadLines함수를 통해 파일 한줄씩 리스트에 읽어들인다.

이와 함께 파일의 이름 또한 추출하여 리스트에 저장하는데, basename(file)을 통해 각 파일의 이름을 추출하고 그 안에서 tools::file_path_sans_ext을 통해 확장명인 .txt를 제거하여 book_name에 저장한다

이후 해당 내용들을 리스트에 저장하는데, book_name을 키로 bible_content를 저장한다

이 과정을 통해 성경의 전체 내용이 제목-1:1 ~ 의 형태로 저장이 되어 있으며 이제 이 챕터와 절, 내용의 형태로 구분하는 과정이 필요하다.

```
# 최종 데이터를 저장할 데이터프레임
bible_df <- tibble(Book = character(), Chapter = numeric(),
Verse = numeric(), Script = character())
```

최종 데이터를 저장할 데이터프레임을 만들고,

```
for (book_name in names(bible_texts)) {
  script_lines <- bible_texts[[book_name]][-1]  # 첫 줄(책 이
름)을 제외한 나머지

  # 각 줄을 분석해서 "Chapter", "Verse", "Script"로 분리
  chapter <- c()
  verse <- c()
  script <- c()

  for (line in script_lines) {
    # 정규식으로 Chapter:Verse와 Script 분리
    match <- regexpr("^(\\d+):(\\d+)\\s(.+)$", line, perl=T
RUE)

    if (match[1] != -1) {
      # Chapter와 Verse 추출
      chapter_verse <- regmatches(line, regexpr("^(\\d+):
(\\d+)", line))
      chapter_verse_split <- strsplit(chapter_verse, ":")
[[1]]
      chapter <- c(chapter, as.numeric(chapter_verse_split
[1]))
      verse <- c(verse, as.numeric(chapter_verse_split[2]))

      # Script 부분 추출
      script <- c(script, regmatches(line, regexpr("\\s(.+)
$", line)))
```

```
    }
  }

  # 각 파일의 결과를 데이터프레임으로 추가
  temp_df <- tibble(
    Book = rep(book_name, length(chapter)),
    Chapter = chapter,
    Verse = verse,
    Script = script
  )

  # 최종 데이터프레임에 추가
  bible_df <- bind_rows(bible_df, temp_df)
}
```

반복문을 통해 book_name별로 변환을 시도한다. 이전과 똑같이 정규식 변환을 통해 Character과 verse, script를 구분하여 저장한다.

```
if (match[1] != -1) {
      # Chapter와 Verse 추출
      chapter_verse <- regmatches(line, regexpr("^(\\d+):
(\\d+)", line))
      chapter_verse_split <- strsplit(chapter_verse, ":")
[[1]]
      chapter <- c(chapter, as.numeric(chapter_verse_split
[1]))
      verse <- c(verse, as.numeric(chapter_verse_split[2]))

      # Script 부분 추출
      script <- c(script, regmatches(line, regexpr("\\s(.+)
$", line)))
    }
```

이 부분에서는 list가 매칭이 완료되었을 때, 챕터와 구문을 추출하는데 :를 기준으로 추출하며 각각을 수치형으로 변환하는 과정을 포함한다.

이후 결과를 확인하면

```
   Book       Chapter Verse Script
   <chr>        <dbl> <dbl> <chr>
 1 01-Genesis       1     1 " In the beginning God created the heavens and the earth."
 2 01-Genesis       1     2 " Now the earth was formless and empty, darkness was over the surface of th…
 3 01-Genesis       1     3 " And God said, \"Let there be light,\" and there was light."
 4 01-Genesis       1     4 " God saw that the light was good, and he separated the light from the dark…
 5 01-Genesis       1     5 " God called the light \"day,\" and the darkness he called \"night.\" And t…
 6 01-Genesis       1     6 " And God said, \"Let there be an expanse between the waters to separate wa…
 7 01-Genesis       1     7 " So God made the expanse and separated the water under the expanse from th…
 8 01-Genesis       1     8 " God called the expanse \"sky.\" And there was evening, and there was morn…
 9 01-Genesis       1     9 " And God said, \"Let the water under the sky be gathered to one place, and…
10 01-Genesis       1    10 " God called the dry ground \"land,\" and the gathered waters he called \"s…
```

다음과 같이 Book, Chapter, Verse, Script별 내용이 들어가 있는데, book의 경우 각 txt의 이름을 추출한 것이기에 01-genesis와 같이 의도하지 않은 값이 포함되어있다. 이를 제거하기 위해서

```
# Book 변수에서 숫자와 하이픈 제거
bible_df$Book <- gsub("^\\d+-", "", bible_df$Book)
```

gsub을 사용해 숫자와 -를 제거한다.

이후 다시 결과를 확인하면

```
  Book    Chapter Verse Script
  <chr>     <dbl> <dbl> <chr>
1 Genesis       1     1 " In the beginning God created the heavens and the earth."
2 Genesis       1     2 " Now the earth was formless and empty, darkness was over the surface of the de…
3 Genesis       1     3 " And God said, \"Let there be light,\" and there was light."
4 Genesis       1     4 " God saw that the light was good, and he separated the light from the darkness…
5 Genesis       1     5 " God called the light \"day,\" and the darkness he called \"night.\" And there…
6 Genesis       1     6 " And God said, \"Let there be an expanse between the waters to separate water …
```

다음과 같이 잘 나왔다.

결과에 포함된 \는 인용절을 표현하기 위한 “를 구분하기 위해 포함되어있으며 이를 제거할 수 있지만 이후 전처리를 위해 남겨두었다.

```
# 구약성서
ot_books <- c("Genesis", "Exodus", "Leviticus", "Numbers",
"Deuteronomy", "Joshua", "Judges",
              "Ruth", "Samuel-1", "Samuel-2", "Kings-1", "K
ings-2", "Chronicles-1", "Chronicles-2",
              "Ezra", "Nehemiah", "Esther", "Job", "Psalm
s", "Proverbs", "Ecclesiastes",
              "Song of Solomon", "Isaiah", "Jeremiah", "Lam
entations", "Ezekiel", "Daniel",
              "Hosea", "Joel", "Amos", "Obadiah", "Jonah",
```

```
"Micah", "Nahum", "Habakkuk",
              "Zephaniah", "Haggai", "Zechariah", "Malach
i")

#신약성서
nt_books <- c("Matthew", "Mark", "Luke", "John", "Acts", "R
omans", "Corinthians-1", "Corinthians-2",
              "Galatians", "Ephesians", "Philippians", "Col
ossians", "Thessalonians-1",
              "Thessalonians-2", "Timothy-1", "Timothy-2",
"Titus", "Philemon", "Hebrews",
              "James", "Peter-1", "Peter-2", "John-1", "Joh
n-2", "John-3", "Jude", "Revelation")
```

이후 필터 작업에서 사용하기 위해, 구약과 신약에 대한 내용을 정리하여, 구약성서와 신약성서의 이름을 저장하였다.

Task 1-2 Perform tokenization and analyze word frequency, including visualizing frequent words. (Plusprovide interpretations)

지금 가지고 있는 데이터프레임에 대하여 토큰화를 진행한 후 단어의 빈도를 분석해보자,

```
bible_df %>%
  unnest_tokens(word, Script)%>%
  count(word,sort = TRUE)
```

가장 먼저 bible을 토큰화 한 후, 숫자를 세고 정렬함

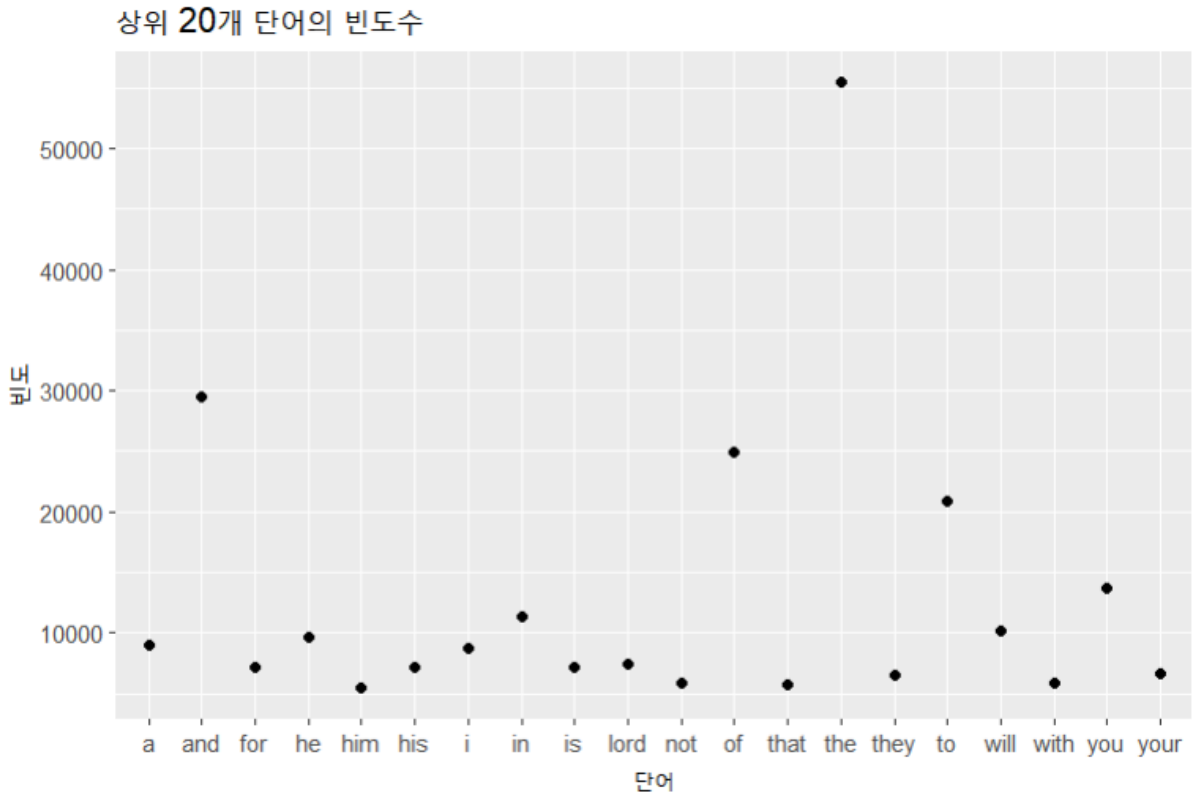
```
# A tibble: 14,302 × 2
   word      n
<chr><int>
 1 the   55481
 2 and   29541
 3 of    24958
 4 to    20867
 5 you   13700
 6 in    11333
 7 will  10156
```

```
 8 he      9630
 9 a       9015
10 i       8719
```

가장 많이 사용된 단어는 관사 the이며 그 다음으로 and, of, to 등의 단어가 사용되었다. 정확한 명사나 동사 보다는 아무래도 전치사, 관사 등의 단어가 많이 사용된 것으로 보인다.

이를 조금 더 보기 쉽도록 시각화를 진행해 보았다.

```
bible_df %>%
  unnest_tokens(word, Script)%>%
  count(word,sort = TRUE)%>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "상위 20개 단어의 빈도수", x = "단어", y = "빈도")
```



명확한 명사는 딱히 눈에 띄지 않으며 그나마 보이는 것에는 lord, 주님을 부르는 명사가 눈에 띈다.

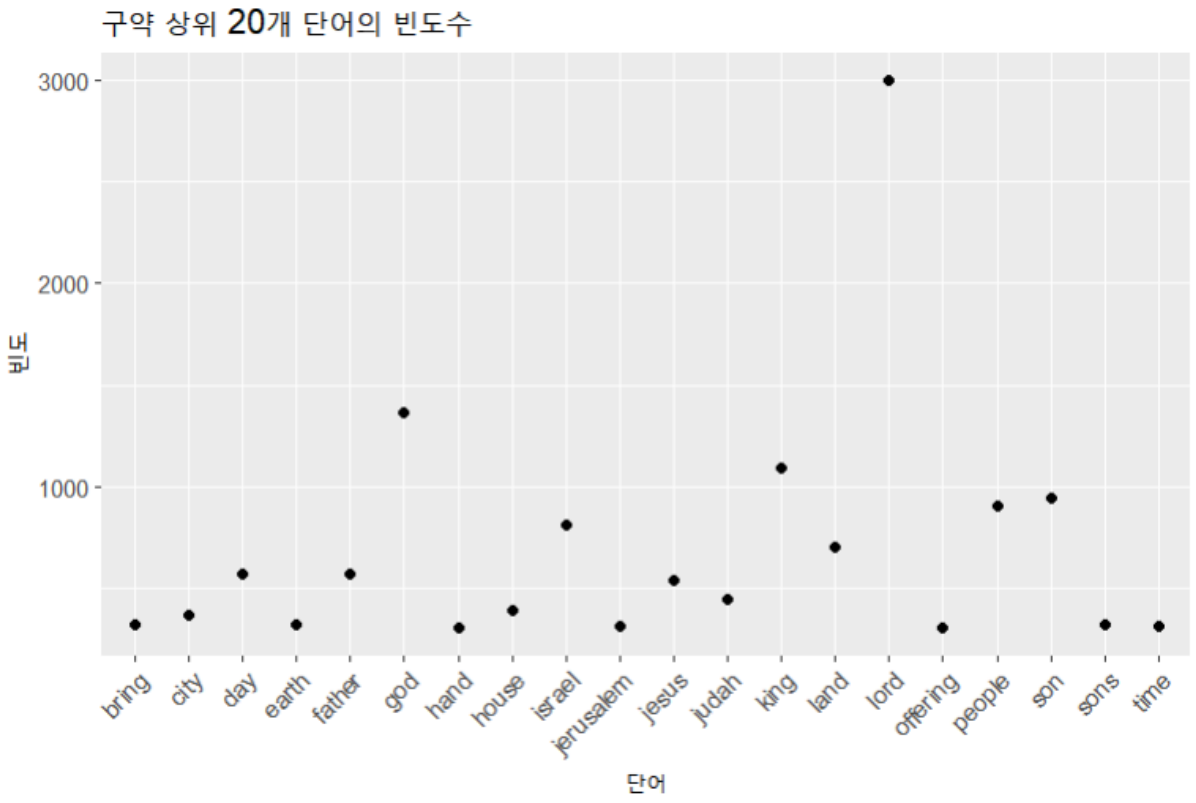
이번에는 자주 등장한 단어들을 불용어 처리한 후 다시 확인해보자

```
data("stop_words")
custom_stopwords <- tibble(word = c("the", "and", "of", "t
o", "you", "in","will","he","a","i","is","his",
                                    "for","they","your","wh
o","my","with","from","him","that","it"))
all_stopwords <- bind_rows(stop_words, custom_stopwords)
```

우선 현재 자주 등장하는 명사가 아닌 단어들을 전부 불용어로 처리한다

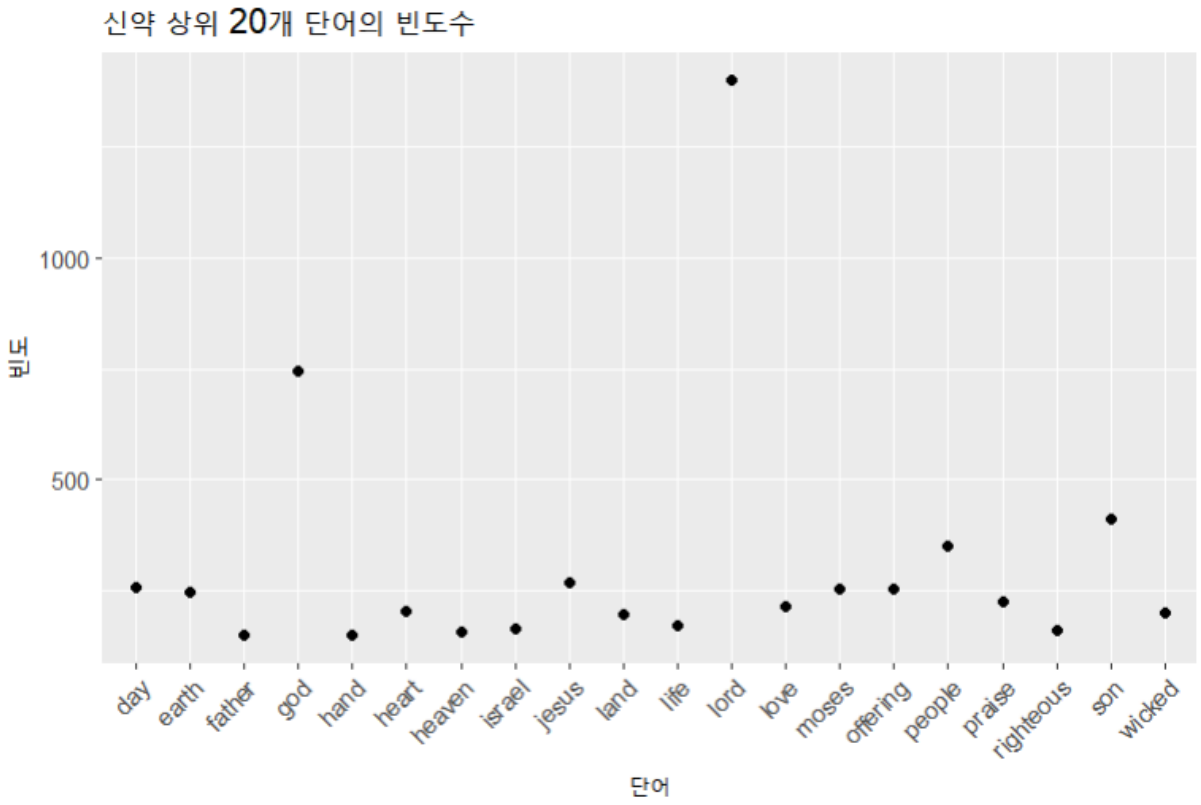
```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% ot_books) %>%  # 책의 순서를 기준으로 필터링
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "구약 상위 20개 단어의 빈도수", x = "단어", y =
"빈도")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

이후 구약 성경에 대해 불용어를 제거한 후 시각화를 한다.



확실히 이번에는 다른 명사들이 자주 등장하는 것을 볼 수 있다.

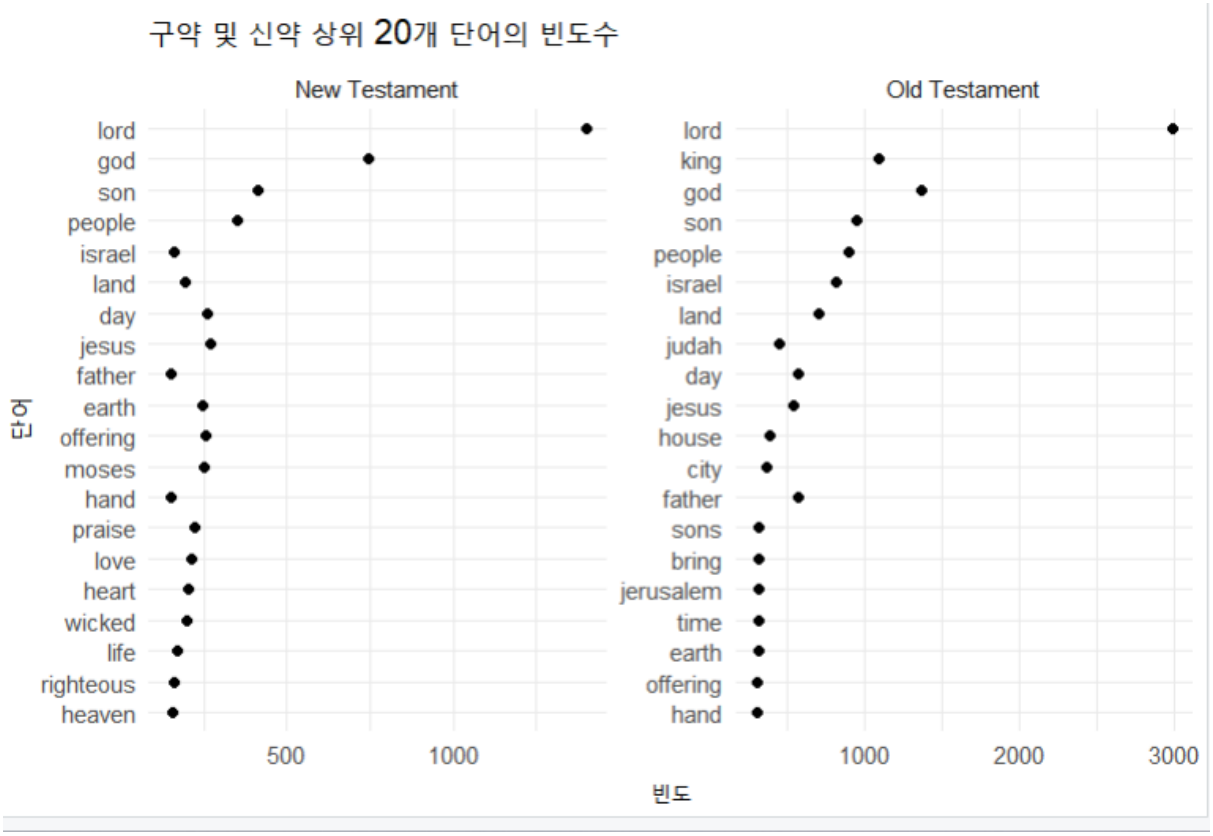
```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% nt_books) %>%  # 책의 순서를 기준으로 필터링
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "신약 상위 20개 단어의 빈도수", x = "단어", y =
"빈도")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



동일하게 신약에 대해서도 시각화를 마쳤다. 이제 두 그래프를 하나로 정리해보자

```
ot_top_words1 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>% # 구약 범위
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE) %>%
  top_n(20, n) %>%
  mutate(Testament = "Old Testament")  # 범주 추가 (구약)

# 신약 성경 (Matthew ~ Revelation) 상위 20개 단어 데이터프레임 생
성
nt_top_words1 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% nt_books) %>%  # 신약 범위
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE) %>%
  top_n(20, n) %>%
  mutate(Testament = "New Testament")  # 범주 추가 (신약)
```

```r
# 두 데이터프레임을 결합
top_words1 <- bind_rows(ot_top_words1, nt_top_words1)

# 하나의 그래프로 결합하여 시각화
ggplot(top_words1, aes(x = reorder(word, n), y = n)) +
  geom_point() +
  facet_wrap(~Testament, scales = "free") +  # 구약, 신약으로 구분
  coord_flip() +  # 단어를 보기 쉽게 가로로 회전
  labs(title = "구약 및 신약 상위 20개 단어의 빈도수",
       x = "단어", y = "빈도") +
  theme_minimal()
```

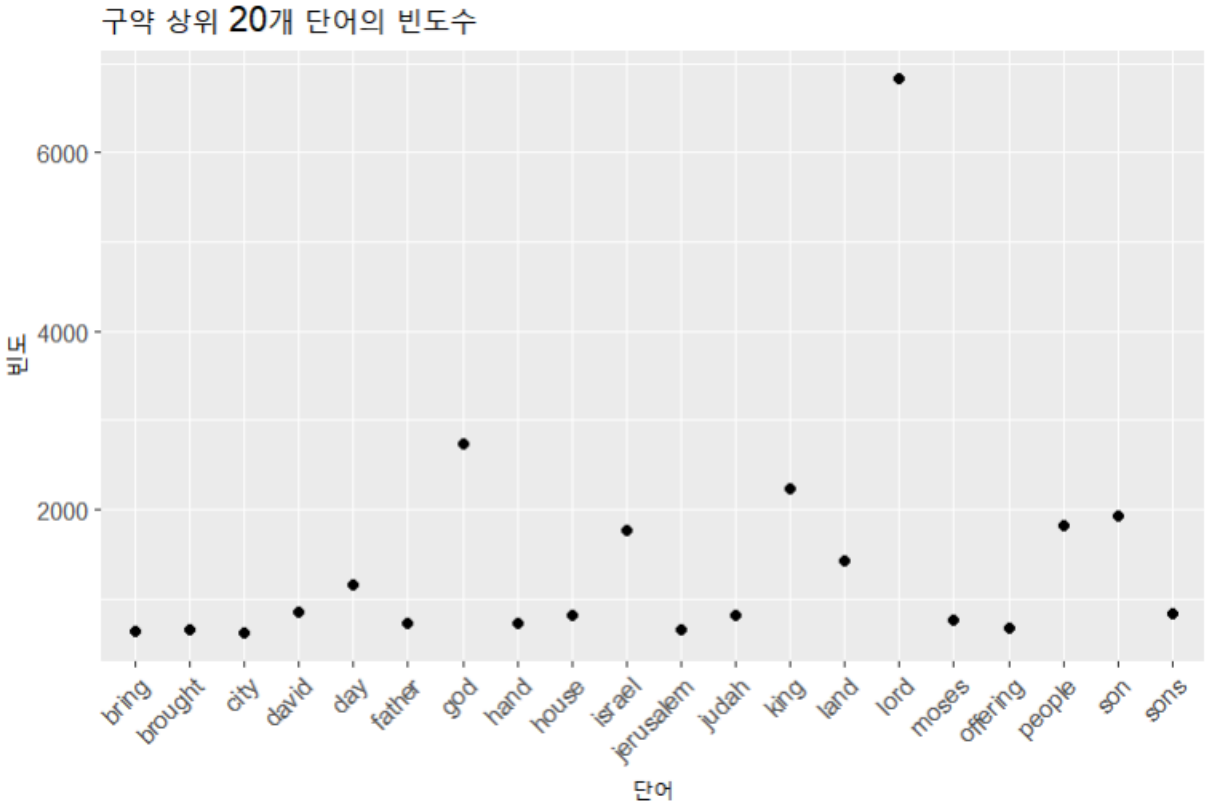


구약과 신약 사이 자주 사용된 상위 20개에 해당하는 단어 빈도수가 조금 다르게 나온 모습이다.

이번에는 불용어를 기본적으로 제공하는 stop_word로 설정해서 제거해 보자

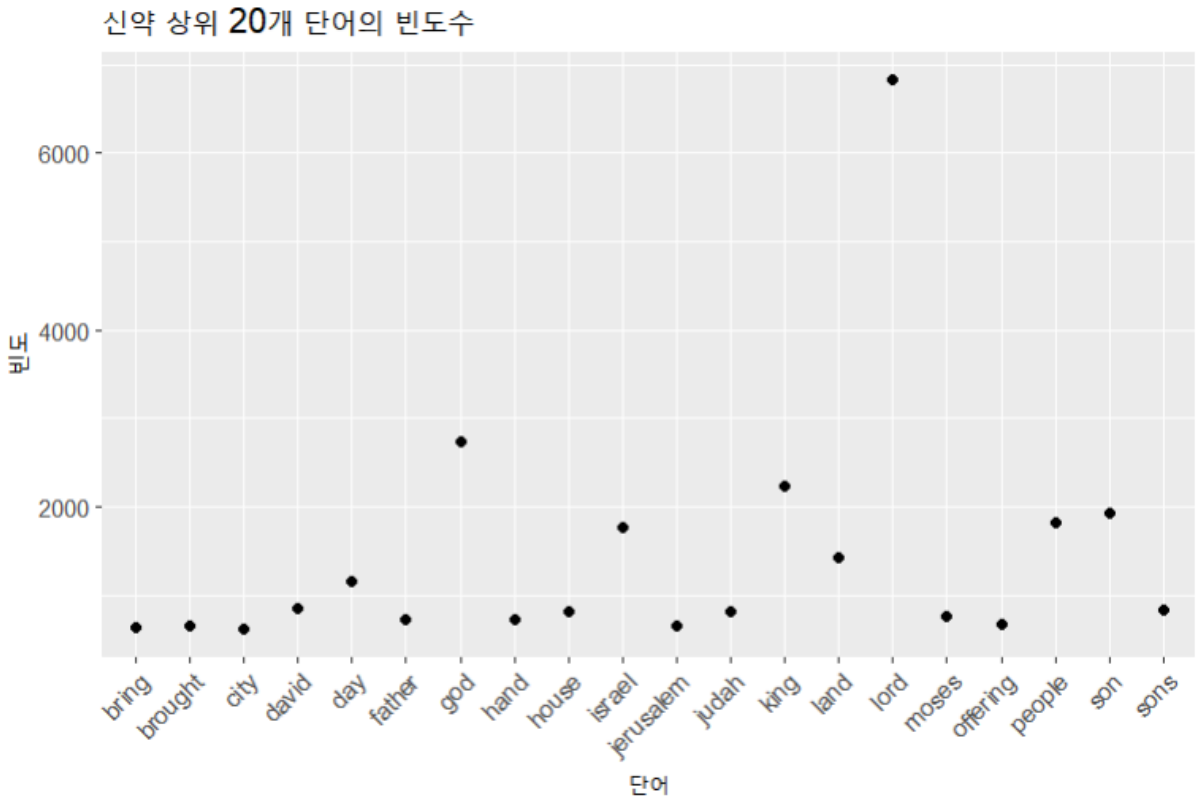
```r
data("stop_words")

bible_df %>%
  unnest_tokens(word, Script) %>%
  anti_join(stop_words, by = "word")
```

```r
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% ot_books) %>%  # 책의 순서를 기준으로 필터링
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "구약 상위 20개 단어의 빈도수", x = "단어", y = "빈도")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% nt_books) %>%  # 책의 순서를 기준으로 필터링
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "신약 상위 20개 단어의 빈도수", x = "단어", y =
"빈도")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

상위 20개를 기준으로 했을때라 이미 제거한 불용어와 일치하여 시각화 그래프에서는 큰 티가 나지 않다



Task 2-1 Analyze word frequency separately for the Old Testament (Genesis to Malachi) and the New Testament (starting from Matthew), and compare how frequent words differ between them.

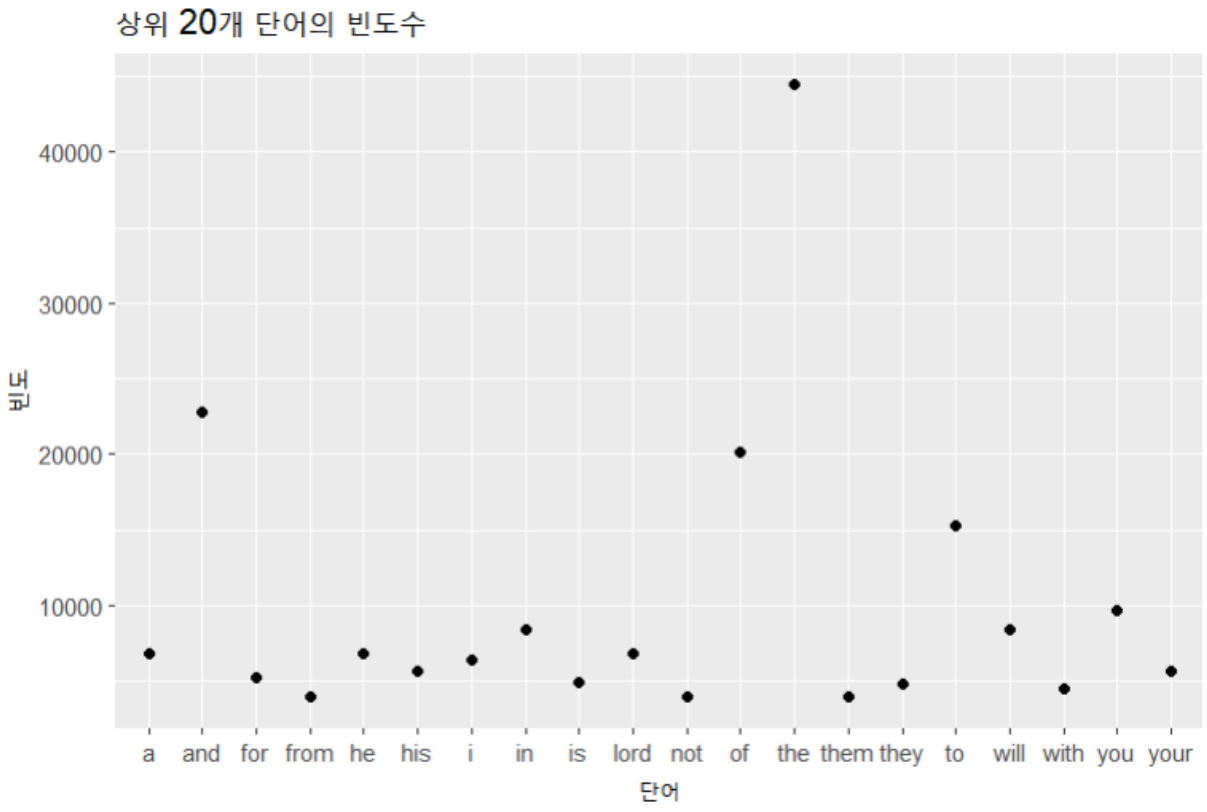
가장 먼저 구약성경 Genesis~Malachi에 해당하는 내용에서 단어숫자를 세어보자

```r
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% ot_books) %>%  # 책의 순서를 기준으로 필터링
  count(word, sort = TRUE)
```

```
# A tibble: 9,492 × 2
   word      n
<chr><int>
 1 the   21993
 2 and   11716
 3 of     9830
 4 to     8278
 5 you    5547
 6 will   4572
 7 he     4475
 8 in     4466
 9 i      3598
10 a      3547
```

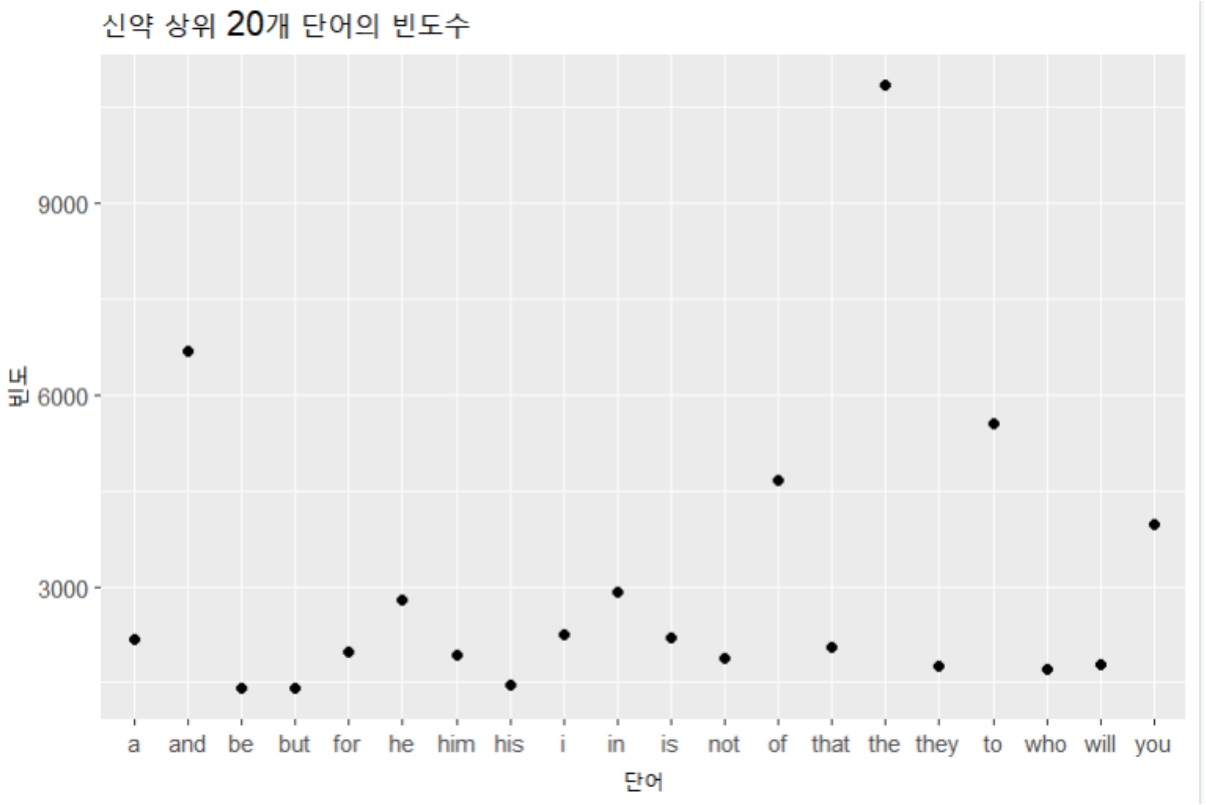
여전히 전체 단어의 빈도수와 매우 유사함을 알 수 있다. 이번에는 이를 시각화하여 살펴보자

```r
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% ot_books) %>%  # 책의 순서를 기준으로 필터링
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "상위 20개 단어의 빈도수", x = "단어", y = "빈도")
```



다음으로 Matthew 부터 나머지 신약성서의 빈도를 세어보자

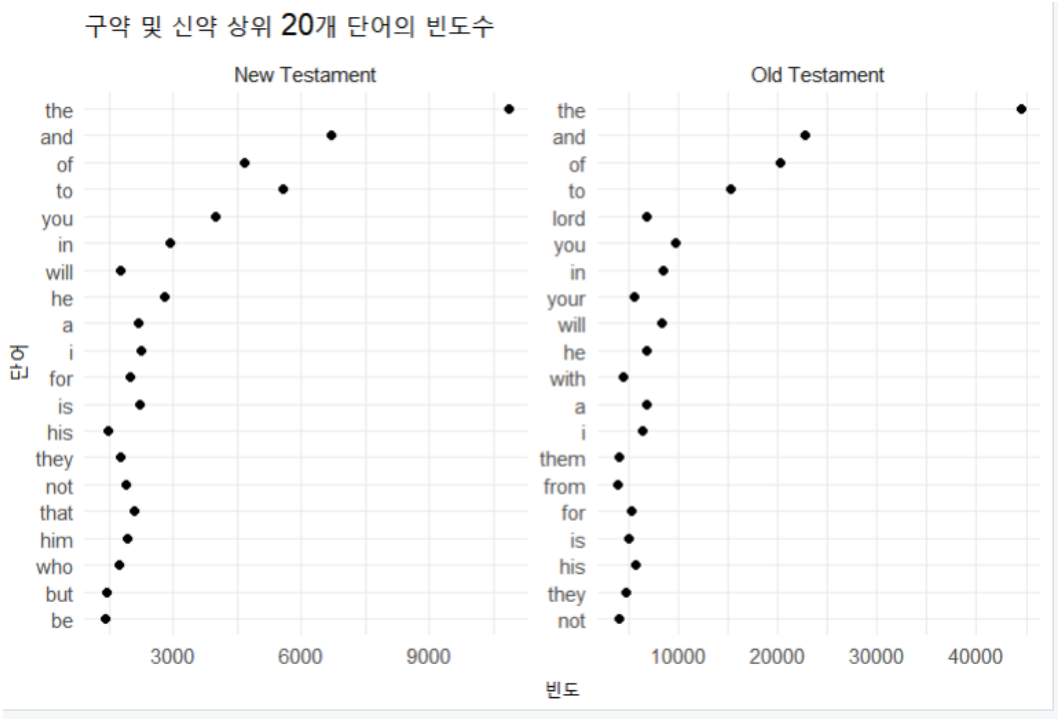
```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book %in% nt_books) %>% # 책의 순서를 기준으로 필터링
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "신약 상위 20개 단어의 빈도수", x = "단어", y =
"빈도")
```



이번에는 두 개의 그래프를 하나로 합쳐보자

```
# 구약 성경 (Genesis ~ Malachi) 상위 20개 단어 데이터프레임 생성
ot_top_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>%  # 구약 범위
  count(word, sort = TRUE) %>%
  top_n(20, n) %>%
  mutate(Testament = "Old Testament")  # 범주 추가 (구약)

# 신약 성경 (Matthew ~ Revelation) 상위 20개 단어 데이터프레임 생
성
nt_top_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% nt_books) %>%  # 신약 범위
  count(word, sort = TRUE) %>%
  top_n(20, n) %>%
  mutate(Testament = "New Testament")  # 범주 추가 (신약)

# 두 데이터프레임을 결합
top_words <- bind_rows(ot_top_words, nt_top_words)
```

```r
# 하나의 그래프로 결합하여 시각화
ggplot(top_words, aes(x = reorder(word, n), y = n)) +
  geom_point() +
  facet_wrap(~Testament, scales = "free") +  # 구약, 신약으로 구분
  coord_flip() +  # 단어를 보기 쉽게 가로로 회전
  labs(title = "구약 및 신약 상위 20개 단어의 빈도수",
       x = "단어", y = "빈도") +
  theme_minimal()
```



구약에서 동일한 단어의 반복이 더 많았으며, 각각 상위 20개를 비교해서는 조금 차이가 있다.

Task2-2 Add the words that should be excluded from the frequency analysis to the stop word list , and apply these in tasks 1 2 and 2 1. (If this was already done in 1

가장 많이 사용된 단어들 중, 명사를 제외한

"the", "and", "of", "to", "you", "in","will","he","a","i","is","his", "for","they","your","who","my","with","from","him","that","it"

이상의 단어들을 제외하였고, 그 후에는 stop_word를 그대로 불용어로 처리하였다.

Task2-3 Extract a list of words that appear more than 10 times in both the Old and New Testaments. Calculate the appearance ratio of these words (frequency of the word / total number of words in the o ld/new testament) for both the New Testament and Old Testament separately. Then, calculate the log ratio: log(propA / propB) where p ropA is the appearance r atio in the New Testament, and propB is the appearance ratio in the Old Testament. Extract the top 20 and bottom 20 words by log ratio, and explain your interpretation

가장 먼저 구약과 신약의 10번 이상 사용된 단어들에 대한 단어 빈도를 계산한다

```
ot_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>%
  count(word) %>%
  filter(n >= 10)

nt_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% nt_books) %>%
  count(word) %>%
  filter(n >= 10)
```

이후, 구약과 신약의 전체 단어를 세어 저장한다

```
# 구약의 전체 단어 수
total_ot_words <- sum(ot_words$n)

# 신약의 전체 단어 수
total_nt_words <- sum(nt_words$n)
```

다음으로 구약과 신약에서 사용된 단어수를 전체 단어로 나누어 비율을 구한다

```r
# 구약에서의 출현 비율 계산
ot_words <- ot_words %>%
  mutate(prop_ot = n / total_ot_words)  # 구약에서의 출현 비율

# 신약에서의 출현 비율 계산
nt_words <- nt_words %>%
  mutate(prop_nt = n / total_nt_words)  # 신약에서의 출현 비율
```

이제 두 구약과 신약, 두 데이터프레임을 inner join으로 합쳐 겹치는 내용만 사용한다

```r
# 구약과 신약의 단어 데이터를 병합 (inner join 사용)
word_comparison <- inner_join(ot_words, nt_words, by = "wor
d", suffix = c("_ot", "_nt"))
```

다음으로 신약에서의 출현비율과, 구약에서의 출현 비율을 계산한 log ratio를 계산한다

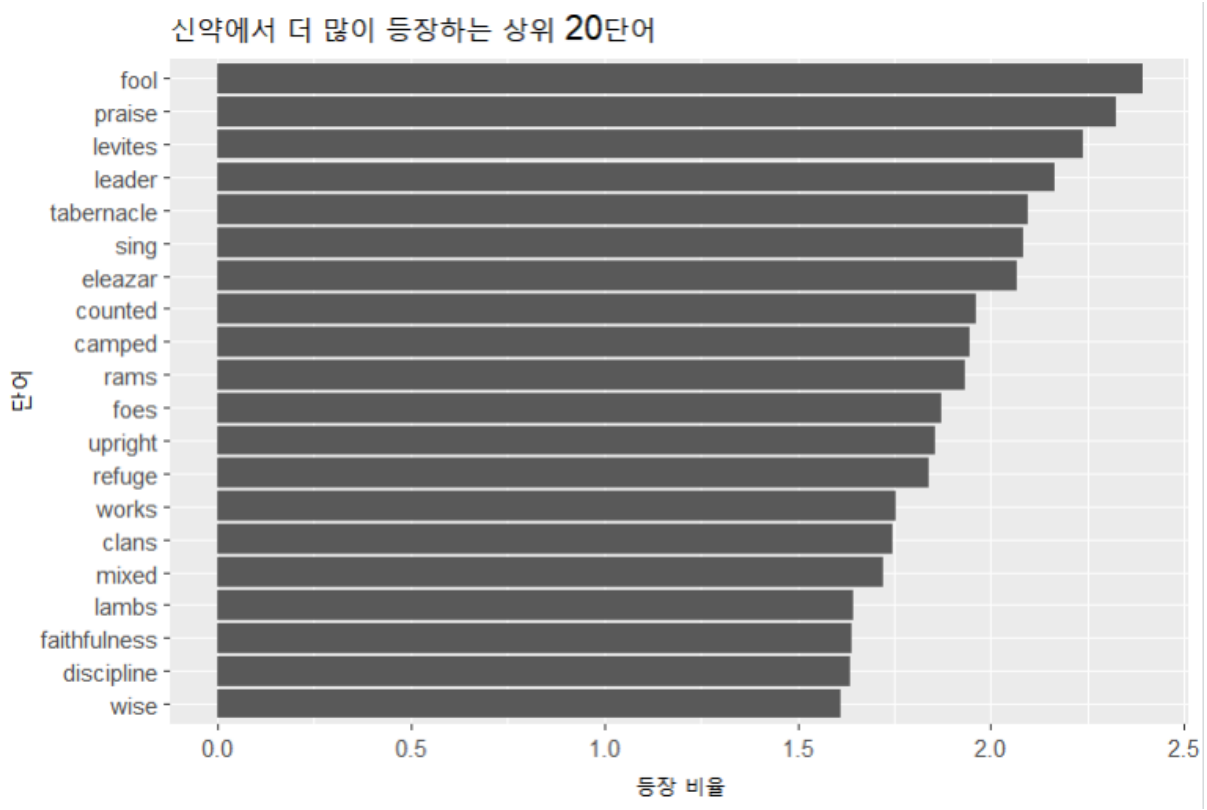
```r
# 로그 비율 계산
word_comparison <- word_comparison %>%
  mutate(log_ratio = log(prop_nt / prop_ot))  # 로그 비율 계산
```

이제 상위, 하위 20개를 추출한다

```r
# 상위 20개 (신약에서 더 많이 등장하는 단어)
top_20 <- word_comparison %>%
  arrange(desc(log_ratio)) %>%
  head(20)

# 하위 20개 (구약에서 더 많이 등장하는 단어)
bottom_20 <- word_comparison %>%
  arrange(log_ratio) %>%
  head(20)
```

```r
top_20%>%
  ggplot(aes(x = reorder(word, log_ratio), y = log_ratio))+
  geom_col()+
  coord_flip() +
```

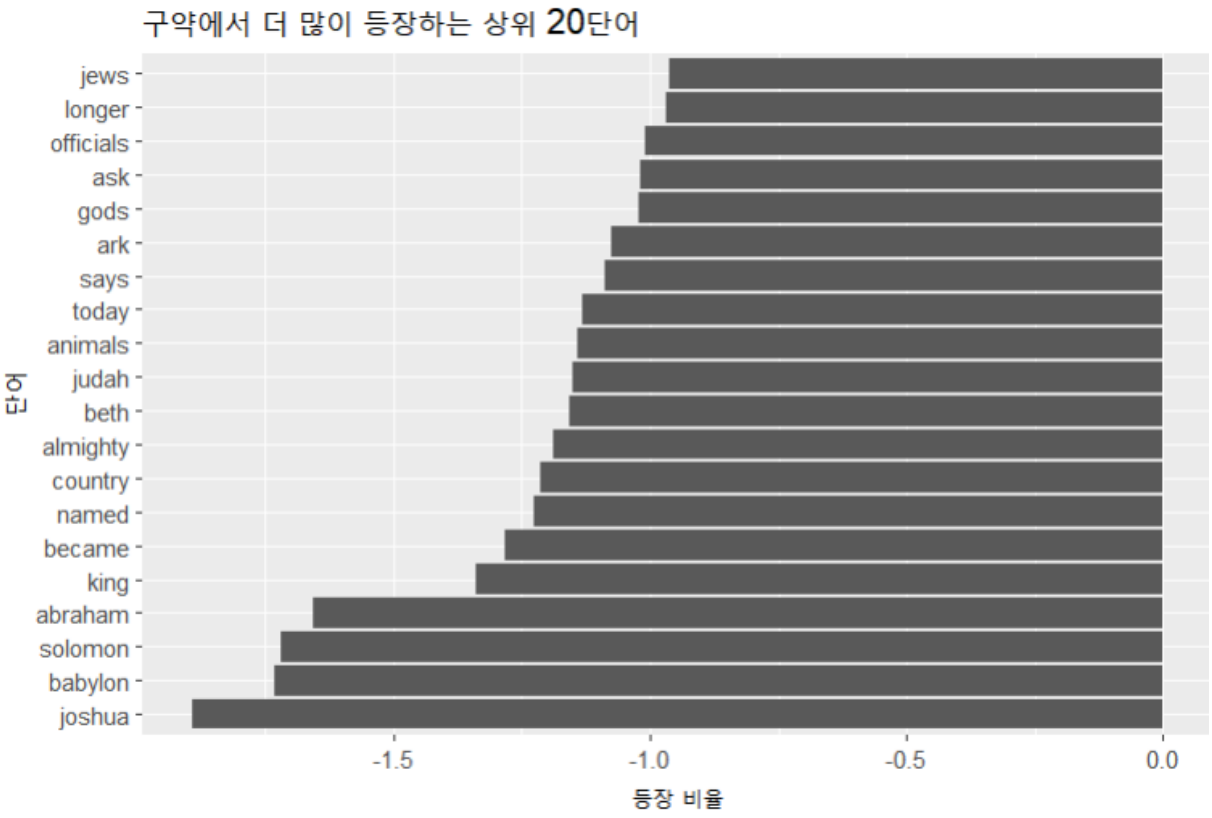
```
  labs(title = "신약에서 더 많이 등장하는 상위 20단어",x = "단어",
y = "등장 비율")
```



log_ratio가 양수인 단어들은 구약보다 신약에서 더 많이 등장하는 단어들이며, 그 내용에는 fool, praise, levites 등의 단어가 많이 등장했다.

이번에는 구약에서 더 많이 등장한 단어를 살펴보면

```
bottom_20%>%
  ggplot(aes(x = reorder(word, log_ratio), y = log_ratio))+
  geom_col()+
  coord_flip() +
  labs(title = "구약에서 더 많이 등장하는 상위 20단어",x = "단어",
y = "등장 비율")
```



Task 2-4 Attempt to improve the result in 2 1 ~ 2 3 to find clear message and insight , what do you think you need for the im provement?

→ 2-3전체단어 당 단어 갯수

이번에는 구약 단어와 신약 단어에 대해 전체 성경에서의 빈도수를 체크한 후, 구약과 신약 각각에 대한 비율을 구해 더 많이 사용된 단어를 찾아보고자 한다.

```
ot_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>%
  count(word) %>%
  filter(n >= 50)

nt_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
```

```
  filter(Book %in% nt_books) %>%
  count(word) %>%
  filter( n > 50)
```

구약 단어와 신약 단어를 각각 20개 이상 사용된 단어들로 필터를 거친다.

```
total_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  count(word)%>%
  filter(n > 50)
```

이번에는 전체 각 단버열 숫자를 세어 저장한다

```
word_comparison <- total_words %>%
  left_join(ot_words, by = "word", suffix = c("_total", "_o
t")) %>%
  left_join(nt_words, by = "word", suffix = c("_ot", "_n
t")) # 신약에 "_nt" 접미사 추가
head(word_comparison)
# NA 처리 후 비율 계산 (NA는 0으로 처리)
word_comparison <- word_comparison %>%
  mutate(
    n_ot = ifelse(is.na(n_ot), 0, n_ot),    # 구약에서 등장하지
않은 단어는 0으로 처리
    n_nt = ifelse(is.na(n), 0, n),          # 신약에서 등장하지
않은 단어는 0으로 처리
    prop_ot = n_ot / n_total,               # 구약에서의 비율
    prop_nt = n_nt / n_total                # 신약에서의 비율
  )
```

구약과 신약, 그리고 전체 성경에서 사용된 count를 단어별로 한 데이터프레임에 저장한다.

이후 구약에서만 등장한다던지, 신약에서만 등장한 단어 NA값들을 0으로 필터링한다

```
word_comparison <- word_comparison %>%
  mutate(diff = abs(prop_ot - prop_nt))  # 구약과 신약 비율 차
이 계산
```

```r
# 차이가 큰 상위 20개 단어 추출
top_20_diff <- word_comparison %>%
  arrange(desc(diff)) %>%
  head(20)
```

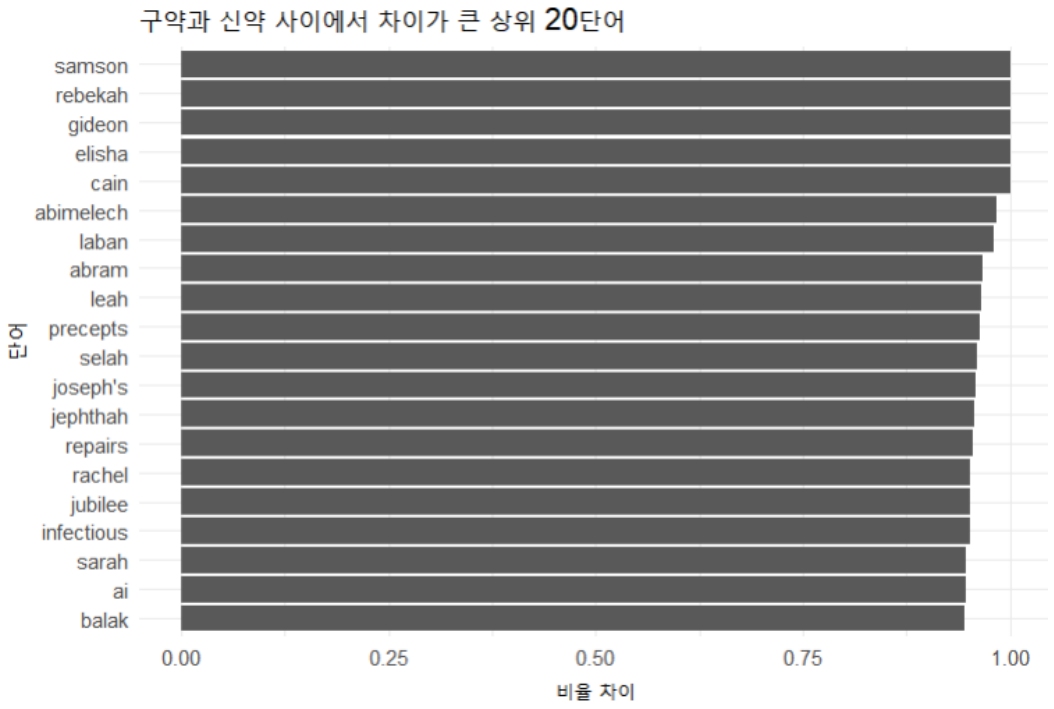
이제 구약과 신약 사이의 비율차이를 계산한 후, diff값이 큰 20개의 단어를 추출한다

```
# A tibble: 20 × 8
   word        n_total  n_ot     n  n_nt prop_ot prop_nt  d
iff
   <chr>         <int> <dbl> <int> <dbl>   <dbl>   <dbl> <d
bl>
 1 abimelech        60    60    NA     0       1       0
1
 2 abner            56    56    NA     0       1       0
1
 3 abram            60    60    NA     0       1       0
1
 4 absalom          91    91    NA     0       1       0
1
 5 ahab             85    85    NA     0       1       0
1
 6 alien            73    73    NA     0       1       0
1
 7 ammonites        88    88    NA     0       1       0
1
 8 amorites         76    76    NA     0       1       0
1
 9 apostles         64     0    64    64       0       1
1
10 aram             84    84    NA     0       1       0
1
11 assyria         122   122    NA     0       1       0
1
12 attacked         61    61    NA     0       1       0
1
13 babylonians      54    54    NA     0       1       0
1
```

```
14 bashan            60    60    NA     0       1       0
1
15 beth             126   126    NA     0       1       0
1
16 bethel            72    72    NA     0       1       0
1
17 boundary          59    59    NA     0       1       0
1
18 bull              98    98    NA     0       1       0
1
19 camped            79    79    NA     0       1       0
1
20 canaanites        57    57    NA     0       1       0
1
```

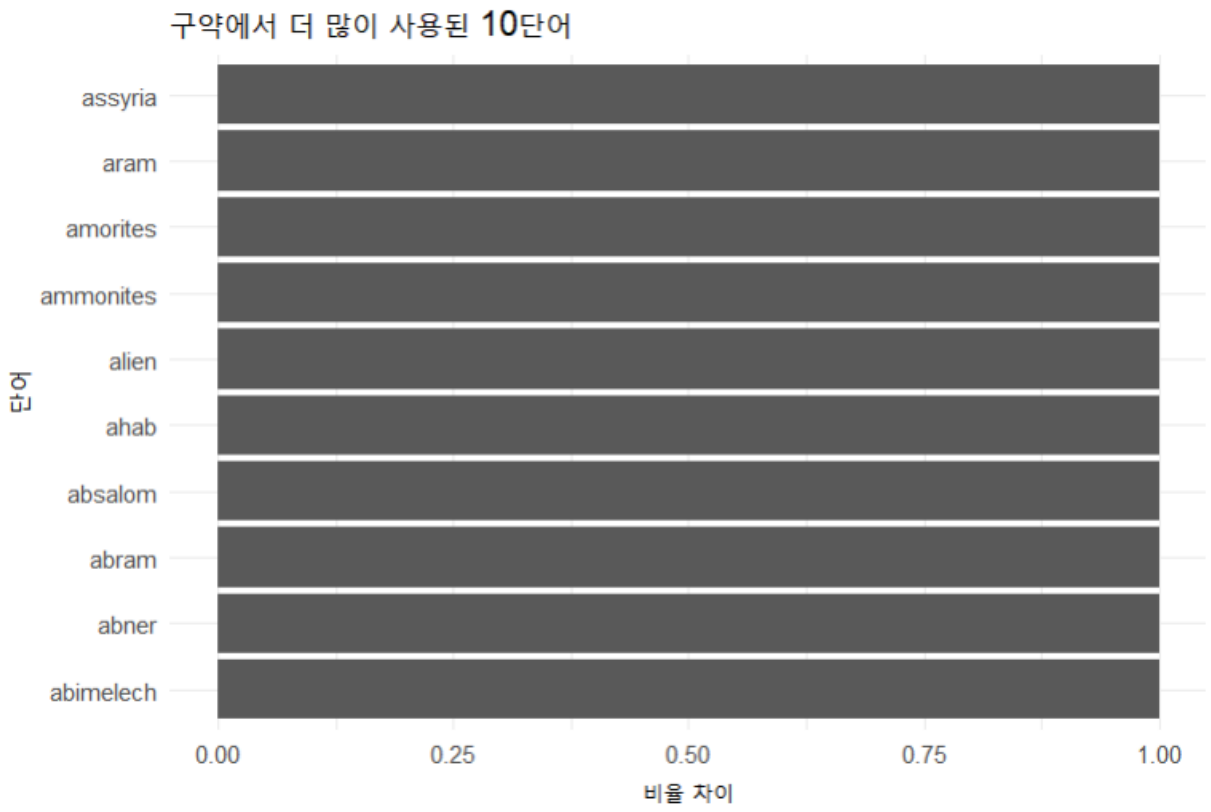
다음과 같은 결과가 나왔다. 구약에서 많이 사용된 단어나, 신약에서 많이 사용된 단어가 섞여서 상위 20개가 출력이 되었다.

```r
top_20_diff %>%
  ggplot(aes(x = reorder(word, diff), y = diff)) +
  geom_col() +
  coord_flip() +
  labs(title = "구약과 신약 사이에서 차이가 큰 상위 20단어",
       x = "단어", y = "비율 차이") +
  theme_minimal()
```

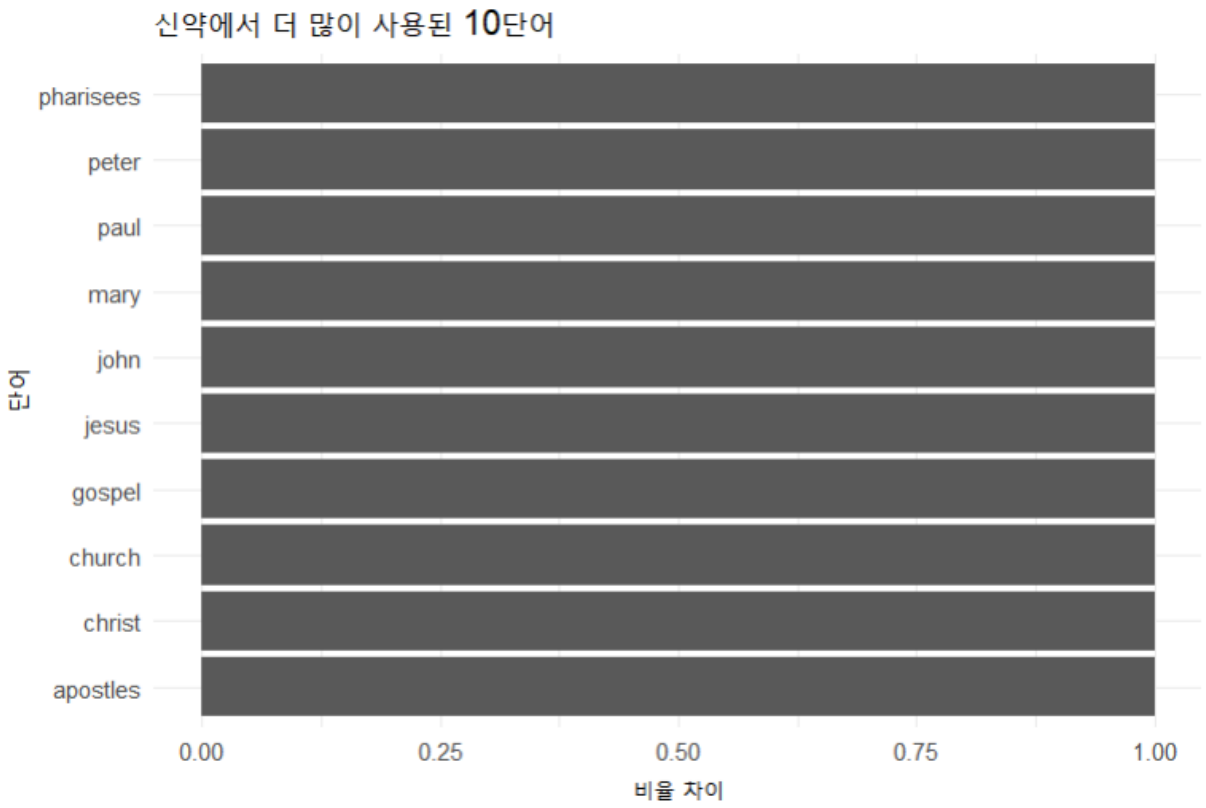


이번에는 구약에서 많이 사용된 단어 10개, 신약에서 많이 사용된 단어 10개를 각각 시각화 해보자

```
top_10_ot <- word_comparison %>%
  arrange(desc(prop_ot)) %>%
  head(10)

top_10_ot %>%
  ggplot(aes(x = reorder(word, diff), y = diff)) +
  geom_col() +
  coord_flip() +
  labs(title = "구약에서 더 많이 사용된 10단어",
       x = "단어", y = "비율 차이") +
  theme_minimal()
```




```r
top_10_nt <- word_comparison %>%
  arrange(desc(prop_nt)) %>%
  head(10)

top_10_nt %>%
  ggplot(aes(x = reorder(word, diff), y = diff)) +
  geom_col() +
  coord_flip() +
  labs(title = "신약에서 더 많이 사용된 10단어",
       x = "단어", y = "비율 차이") +
  theme_minimal()
```



구약과 신약 각각에서 많이 사용된 단어들에 대한 비율을 보면 구약이나 신약에서 각각 한쪽에서만 50회 이상 사용된 단어들이 62개로, 대부분이 명사 위주의 단어가 많이 사용되었다.

Task 3-1 Perform sentiment analysis for both the New and Old Testament s separately.

우선 tidytext패키지에서 제공하는 감정 사전을 가지고 감정분석을 시도한다

```
sentiments <- get_sentiments("bing")
```

감정 사전 bing을 확인한 후

```
ot_sentiment <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>%
  inner_join(sentiments, by = "word") %>%  # 감정 사전과 매칭
  count(sentiment)
```

구약 성경에서 등장한 단어들에 대해 감정사전과 매칭을 통해 감정이 나타난 단어들의 숫자를 센다

이후 확인해보면

```
  sentiment     n
  <chr>     <int>
1 negative  16373
2 positive  13388
```

부정적인 단어는 16373회, 긍정적인 단어는 13388회 등장했다. 신약 역시 이와 동일한 작업을 거치면

```
nt_sentiment <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% nt_books) %>%
  inner_join(sentiments, by = "word") %>%
  count(sentiment)
```

```
# A tibble: 2 × 2
  sentiment     n
  <chr>     <int>
1 negative   4738
2 positive   5610
```

신약의 경우에는 긍정적인 단어가 5610, 부정적인 단어가 4738으로 나타났다.

Task 3-2 Compare the frequency of sentiment related words between the New and Old Testaments.

구약 성경의 긍정적인 단어와 부정적인 단어들에 대해 상위 10개씩을 확인해보자

```
ot_positive_top10 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>%
  inner_join(sentiments, by = "word") %>%  # 감정 사전과 매칭
  filter(sentiment == "positive") %>%  # 긍정적인 단어만 선택
```

```
  count(word, sort = TRUE) %>%  # 단어 빈도 계산
  top_n(10, n)
print(ot_positive_top10)
```

```
# A tibble: 11 × 2
   word         n
   <chr>    <int>
 1 like      1223
 2 great      530
 3 gold       427
 4 holy       392
 5 good       356
 6 right      317
 7 love       302
 8 work       299
 9 praise     296
10 covenant   262
```

우선 구약성경의 긍정적인 단어는 다음과 같다.

```
ot_negative_top10 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% ot_books) %>%
  inner_join(sentiments, by = "word") %>%  # 감정 사전과 매칭
  filter(sentiment == "negative") %>%  # 긍정적인 단어만 선택
  count(word, sort = TRUE) %>%  # 단어 빈도 계산
  top_n(10, n)
print(ot_negative_top10)
```

```
# A tibble: 10 × 2
   word        n
   <chr>   <int>
 1 sin       346
 2 wicked    342
 3 evil      327
 4 death     306
 5 desert    275
```

```
 6 die       254
 7 enemies   250
 8 anger     248
 9 destroy   213
10 fear      210
```

신약에 대해서도 동일하게 확인한 후 두 가지에 대해 시각화를 진행한다

```r
nt_positive_top10 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% nt_books) %>%
  inner_join(sentiments, by = "word") %>%  # 감정 사전과 매칭
  filter(sentiment == "positive") %>%  # 긍정적인 단어만 선택
  count(word, sort = TRUE) %>%  # 단어 빈도 계산
  top_n(10, n)

nt_negative_top10 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book %in% nt_books) %>%
  inner_join(sentiments, by = "word") %>%  # 감정 사전과 매칭
  filter(sentiment == "negative") %>%  # 긍정적인 단어만 선택
  count(word, sort = TRUE) %>%  # 단어 빈도 계산
  top_n(10, n)

print(nt_positive_top10)
print(nt_negative_top10)
```

```
# A tibble: 10 × 2
   word       n
   <chr>  <int>
 1 like     261
 2 faith    252
 3 good     252
 4 heaven   236
 5 love     226
 6 holy     192
```

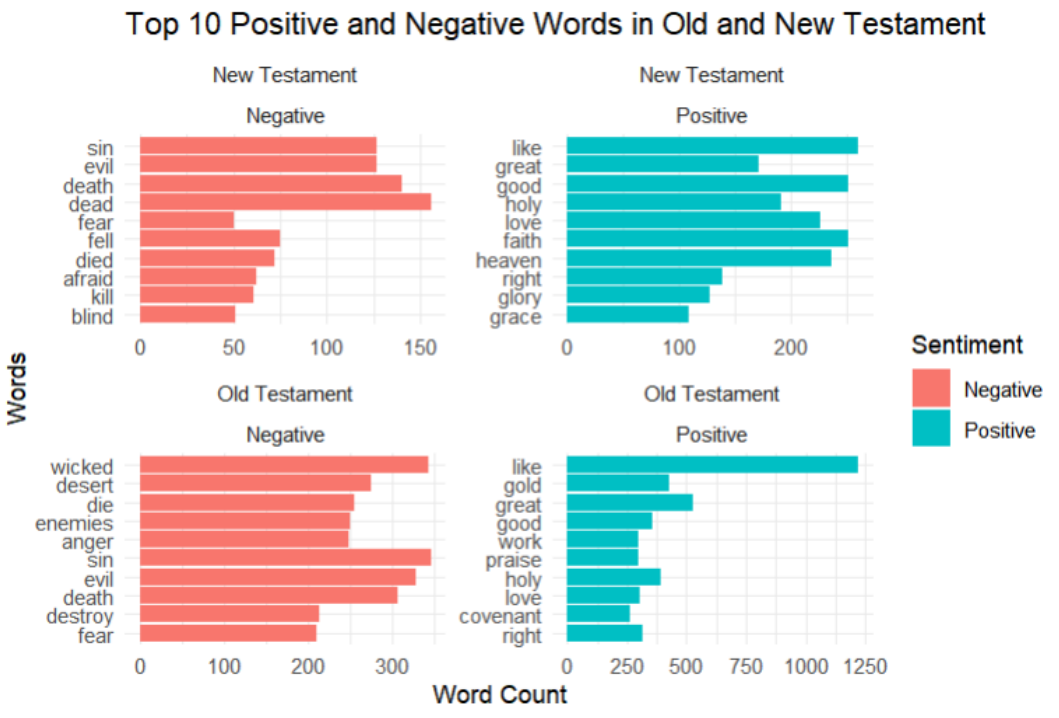
```
 7 great    172
 8 right    139
 9 glory    127
10 grace    109
```

```
# A tibble: 10 × 2
   word       n
   <chr>  <int>
 1 dead     156
 2 death    140
 3 evil     127
 4 sin      127
 5 fell      75
 6 died      72
 7 afraid    62
 8 kill      61
 9 blind     51
10 fear      50
```

이제 해당 내용들을 시각화를 해보자

```
ot_positive_top10 <- ot_positive_top10 %>%
  mutate(Testament = "Old Testament", Sentiment = "Positiv
e")

ot_negative_top10 <- ot_negative_top10 %>%
  mutate(Testament = "Old Testament", Sentiment = "Negativ
e")

nt_positive_top10 <- nt_positive_top10 %>%
  mutate(Testament = "New Testament", Sentiment = "Positiv
e")

nt_negative_top10 <- nt_negative_top10 %>%
  mutate(Testament = "New Testament", Sentiment = "Negativ
e")

# 2. 네 개의 데이터프레임을 하나로 결합
```

```r
top_words <- bind_rows(ot_positive_top10, ot_negative_top1
0, nt_positive_top10, nt_negative_top10)

# 3. 시각화: 각 감정과 성경 범위별 단어 빈도 시각화
ggplot(top_words, aes(x = reorder(word, n), y = n, fill = S
entiment)) +
  geom_col() +
  facet_wrap(~Testament + Sentiment, scales = "free") +  #
구약/신약, 긍정/부정으로 구분
  coord_flip() +
  labs(title = "Top 10 Positive and Negative Words in Old a
nd New Testament",
       x = "Words", y = "Word Count") +
  theme_minimal()
```



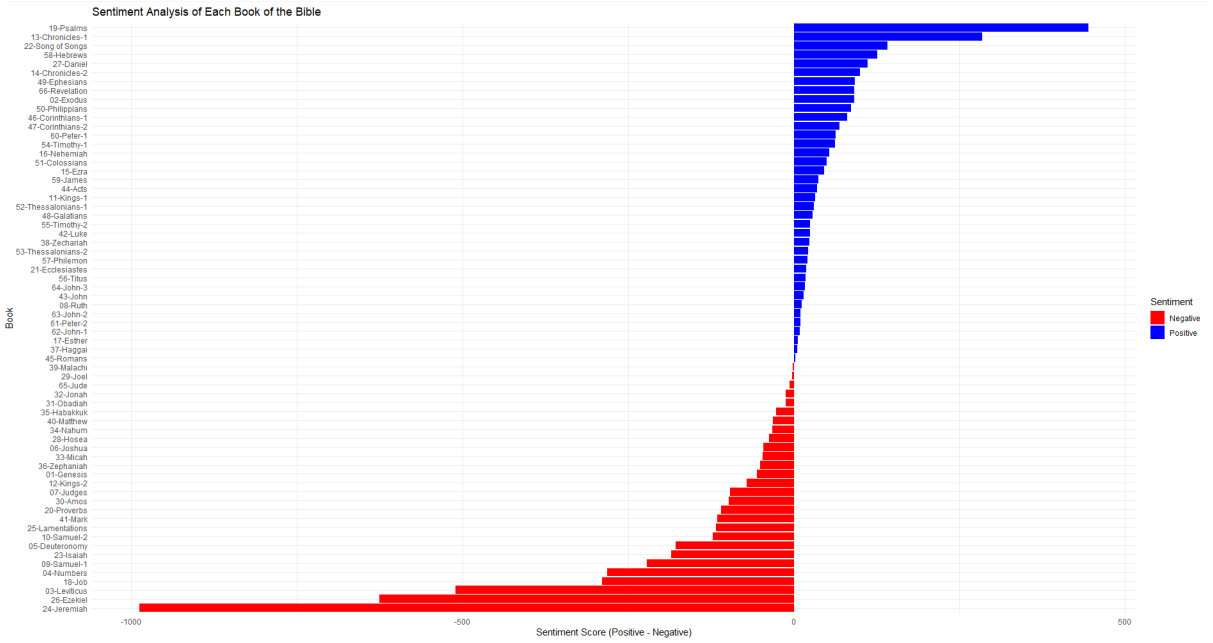
신약과 구약성경 양쪽 모두에서 공통적으로 나타나는 긍정, 부정적 단어가 존재하며 부정적 단어의 경우 evil, sin, die나 death, dead등이 공통적이고, 긍정적 단어에서는 like와 good, great holy등의 단어가 자주 사용되었다.

Task 3-3 Analyze sentiment for each book of the Bible. Identify which books have positive sentiment and which have negative sentiment.

```
book_sentiment <- bible_df %>%
  unnest_tokens(word, Script) %>%
  inner_join(sentiments, by = "word") %>%  # 감정 사전과 매칭
  group_by(Book, sentiment) %>%  # 책별로 그룹화하고 감정별로 나
눔
  count() %>%  # 각 감정별로 단어 수 세기
  spread(sentiment, n, fill = 0)%>%  # 긍정과 부정 단어 개수를
나란히 표시
  mutate(sentiment_score = positive - negative)
```

책별로 그룹화를 한 후, 감정별 단어 수를 세고 그 개수를 체크한 후 최종적인 스코어를 매기는 방법으로 책이 긍정적인가, 부정적인가를 나타냈다. 이제 이를 시각화해보면

```
ggplot(book_sentiment, aes(x = reorder(Book, sentiment_scor
e), y = sentiment_score, fill = sentiment_score > 0)) +
  geom_col() +
  coord_flip() +
  labs(title = "Sentiment Analysis of Each Book of the Bibl
e",
       x = "Book",
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sent
iment", labels = c("Negative", "Positive")) +
  theme_minimal()
```



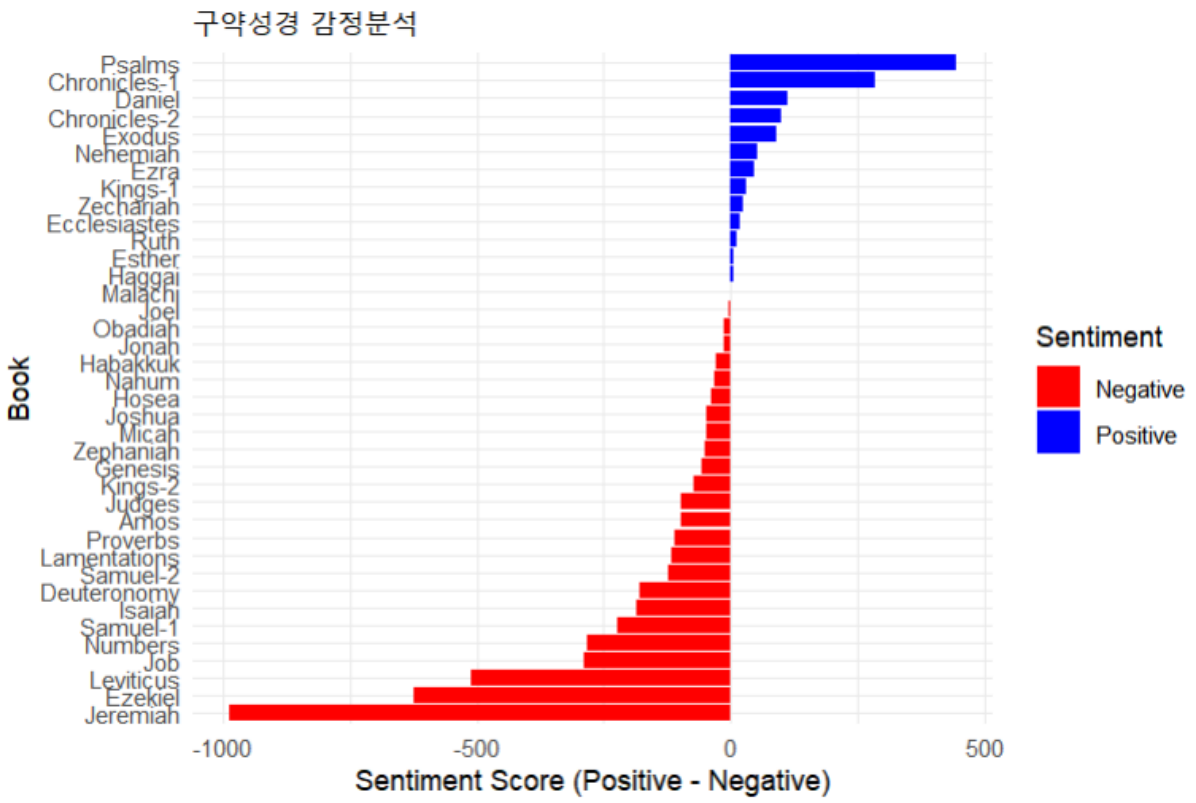
아무래도 전체 책 자체가 너무 많다보니 한눈에 보기 어려운 느낌이다.

신약과 구약에 대해 각각 긍정-부정 찾아보기

```
book_sentiment <- bible_df %>%
  unnest_tokens(word, Script) %>%
  inner_join(sentiments, by = "word", relationship = "many-
to-many") %>%  # 감정 사전과 매칭
  filter(Book %in% ot_books) %>%
  group_by(Book, sentiment) %>%  # 책별로 그룹화하고 감정별로 나
눔
  count() %>%  # 각 감정별로 단어 수 세기
  spread(sentiment, n, fill = 0)%>%  # 긍정과 부정 단어 개수를
나란히 표시
  mutate(sentiment_score = positive - negative)
```

```
ggplot(book_sentiment2, aes(x = reorder(Book, sentiment_sco
re), y = sentiment_score, fill = sentiment_score > 0)) +
  geom_col() +
  coord_flip() +
  labs(title = "Sentiment Analysis of Each Book of the Bibl
e",
       x = "Book",
```

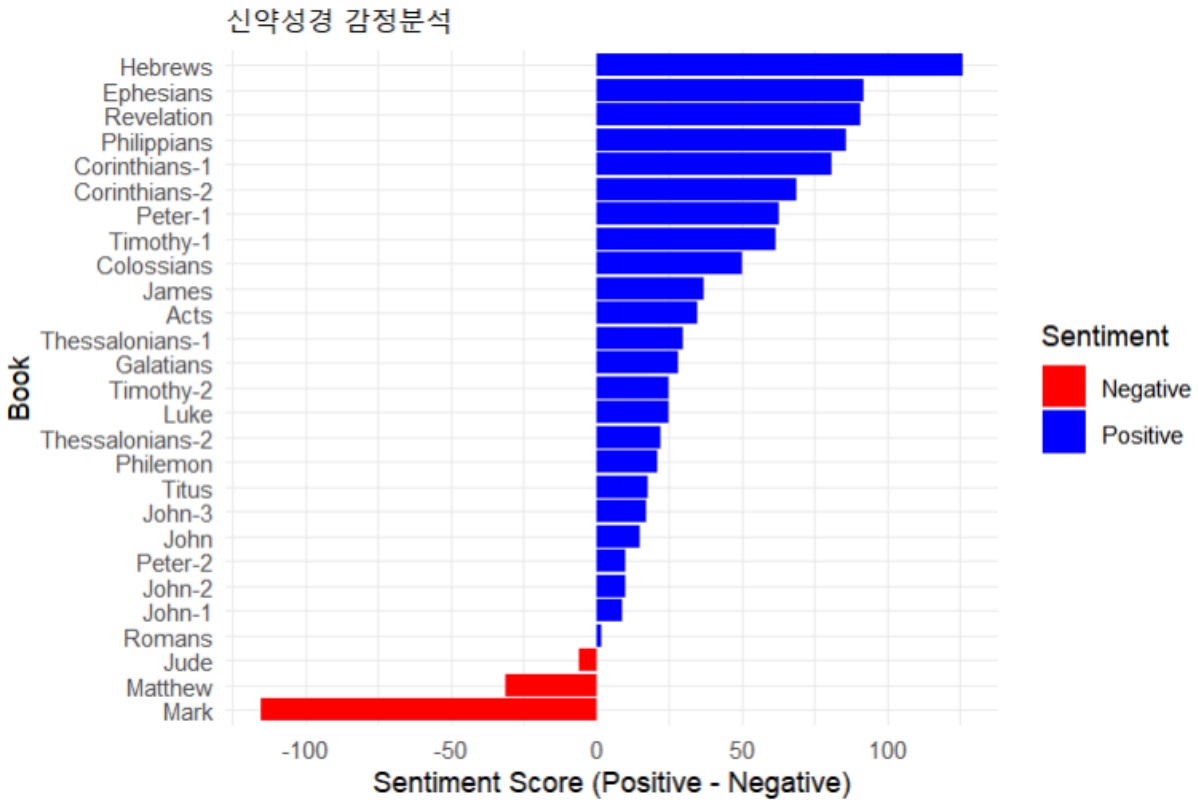
```
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sent
iment", labels = c("Negative", "Positive")) +
  theme_minimal()
```



구약성경의 감정을 분석

```
book_sentiment1 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  inner_join(sentiments, by = "word", relationship = "many-
to-many") %>%  # 감정 사전과 매칭
  filter(Book %in% nt_books) %>%
  group_by(Book, sentiment) %>%  # 책별로 그룹화하고 감정별로 나
눔
  count() %>%  # 각 감정별로 단어 수 세기
  spread(sentiment, n, fill = 0)%>%  # 긍정과 부정 단어 개수를
나란히 표시
  mutate(sentiment_score = positive - negative)
```

```r
ggplot(book_sentiment1, aes(x = reorder(Book, sentiment_sco
re), y = sentiment_score, fill = sentiment_score > 0)) +
  geom_col() +
  coord_flip() +
  labs(title = "신약성경 감정분석",
       x = "Book",
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sent
iment", labels = c("Negative", "Positive")) +
  theme_minimal()
```



Task3-4 Choose 3~4 books of your interest and perform sentiment analysis by chapter for certain books, such as Genesis, Chronicles, Kings, Psalms, etc., which are relatively longer than others . Analyze how the sentiment changes throughout the chapters and compare your findings with your knowledge of the Bible.

우선, 챕터별 단어 빈도수를 세어, 가장 빈도수가 많은 3개의 책에 대해 감정분석을 시도해 보겠다

```
bible_df %>%
  unnest_tokens(word, Script) %>%
  group_by(Book) %>%
  summarise(total_words = n()) %>%  # 각 책에서의 전체 단어 수 계산
  top_n(3, total_words) %>%  # 상위 3개의 책 추출
  arrange(desc(total_words))  # 내림차순 정렬
```

book으로 그룹화 후, 전체 단어수를 세고 상위 3개를 추출했다.

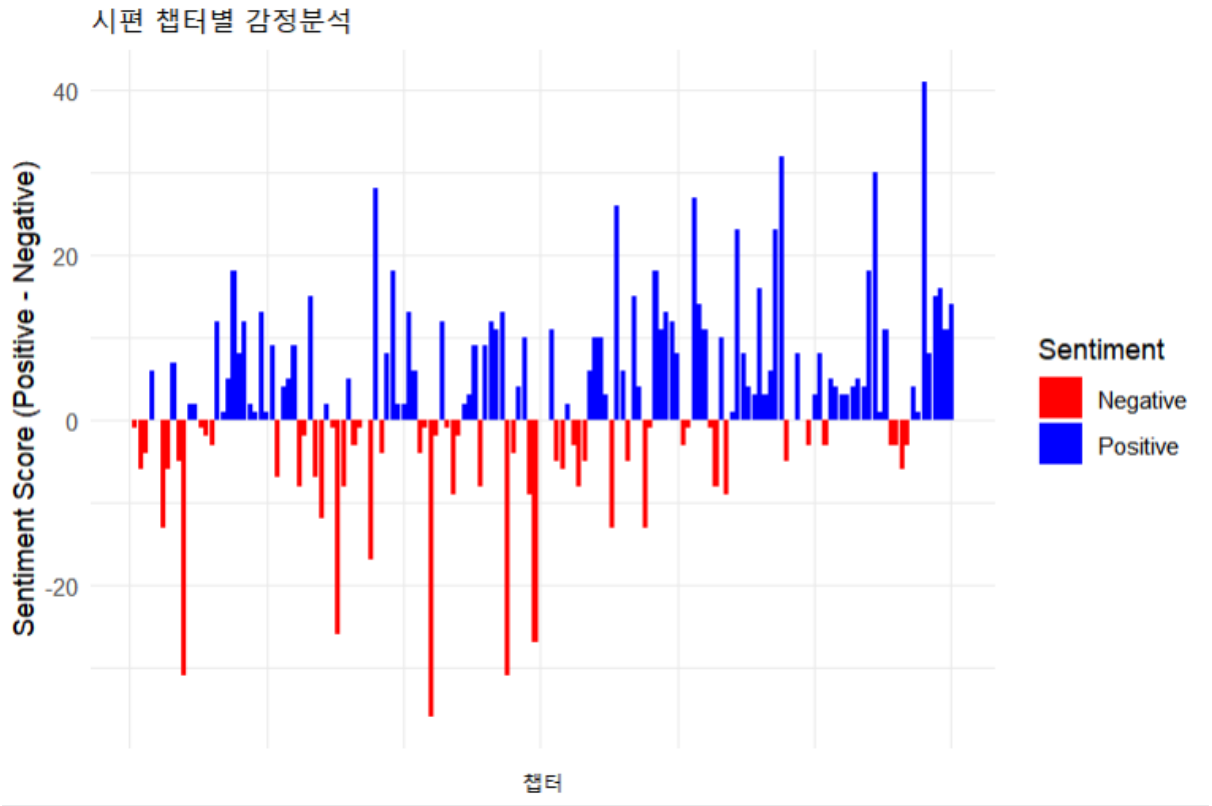
```
  Book     total_words
<chr><int>
1 Psalms         40200
2 Jeremiah       38388
3 Ezekiel        35904
```

시편과 예레미아서, 그리고 에스겔서 3개의 책이 나왔고, 이제 이를 사용하여 챕터별 감정분석을 수행해보자

시편

```
bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book == "Psalms") %>%  # 시편만 필터링
  inner_join(sentiments, by = "word") %>%
  group_by(Chapter, sentiment) %>%  # 챕터별로 그룹화하고 감정별로 나눔
  count() %>%  # 각 감정별로 단어 수 세기
  spread(sentiment, n, fill = 0) %>%  # 긍정과 부정 단어 개수를 나란히 표시
  mutate(sentiment_score = positive - negative) %>%  # 긍정 - 부정으로 감정 점수 계산
  arrange(Chapter)%>%
  ggplot(aes(x = Chapter, y= sentiment_score, fill = sentiment_score > 0))+
  geom_col()+
```

```
  labs(title = "시편 챕터별 감정분석",
       x = "챕터",
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sent
iment", labels = c("Negative", "Positive"))+
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # X축 텍스트 제거
        axis.ticks.x = element_blank())  # X축 눈금 제거
```



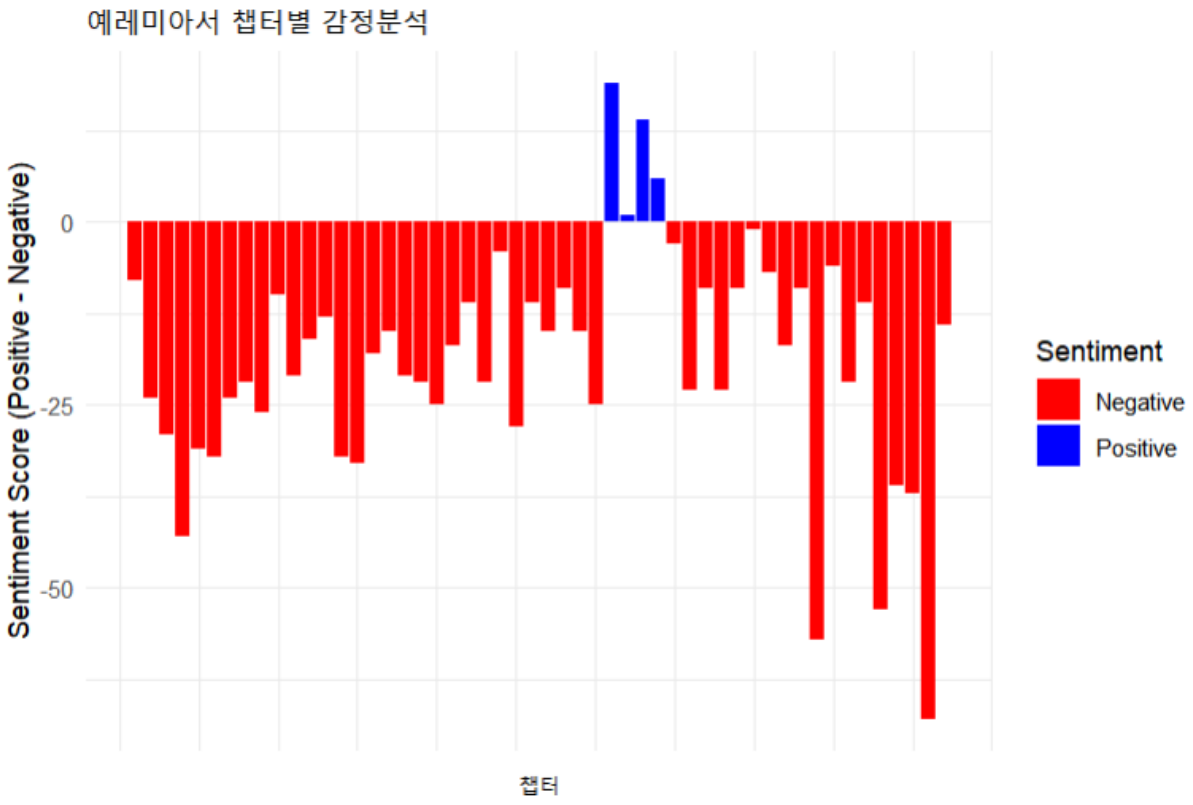
시편은 솔로몬, 다윗을 포함한 많은 저자들이 하나님께 드리는 시의 모음집이다. 이에는 예언, 찬양과 경배, 탄식과 기도, 감사와 회복의 내용이 모두 포함된다. 이 내용들이 한 편에 모두 담겨있는 경우도 있고, 장별로 이어지는 경우도 있다.

내용을 파악한 결고 일반적으로 탄식 → 회개 → 하나님께 올려드리는 감사와 찬양으로의 구조로 이루어진다고 볼 수 있다. 따라서 부정 → 긍정으로 변환하는 패턴을 보일 수 있음.

예레미아서

```
bible_df %>%
  unnest_tokens(word, Script) %>%
```

```
  filter(Book == "Jeremiah") %>%  # 시편만 필터링
  inner_join(sentiments, by = "word") %>%
  group_by(Chapter, sentiment) %>%  # 챕터별로 그룹화하고 감정별로 나눔
  count() %>%  # 각 감정별로 단어 수 세기
  spread(sentiment, n, fill = 0) %>%  # 긍정과 부정 단어 개수를 나란히 표시
  mutate(sentiment_score = positive - negative) %>%  # 긍정 - 부정으로 감정 점수 계산
  arrange(Chapter)%>%
  ggplot(aes(x = Chapter, y= sentiment_score, fill = sentiment_score > 0))+
  geom_col()+
  labs(title = "예레미아서 챕터별 감정분석",
       x = "챕터",
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sentiment", labels = c("Negative", "Positive"))+
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # X축 텍스트 제거
        axis.ticks.x = element_blank())  # X축 눈금 제거
```



회복을 약속하기 이전:
예레미야서의 초기 부분은 이스라엘과 유다의 불순종과 우상숭배로 인한 하나님의 심판을 예고

하나님은 이스라엘과 유다가 그들의 죄악으로 인해 기근, 전염병, 전쟁과 같은 재앙을 겪게 될 것임을 경고

예레미야를 통해 백성들에게 회개하라고 말씀하심.

이스라엘과 유다의 반역으로 인한 하나님의 진노는 바벨론을 심판의 도구로 삼아 예루살렘의 함락과 유다의 멸망을 불러올 것이라는 예언으로 이어짐.

회복의 약속:
심판의 경고 속에서도 하나님은 유다와 이스라엘에 대한
회복의 약속을 명확히 밝히심

하나님은 포로로 잡혀간 백성들이 다시 그들의 땅으로 돌아오고, 그 땅이 재건될 것을 약속 (30~33장).

그들은 다시 하나님의 은혜 아래에서 번영을 누리며, 하나님은 그들과 새로운 언약을 맺으실 것이라고 선언

특히, 하나님은 다윗의 후손 중에서 의로운 지도자가 일어날 것이며, 이스라엘을 공의로 다스릴 것이라고 예언

이 회복은 하나님의 은혜와 자비를 보여줌.

회복 이후:
회복의 약속 이후에도 경고의 메시지는 지속

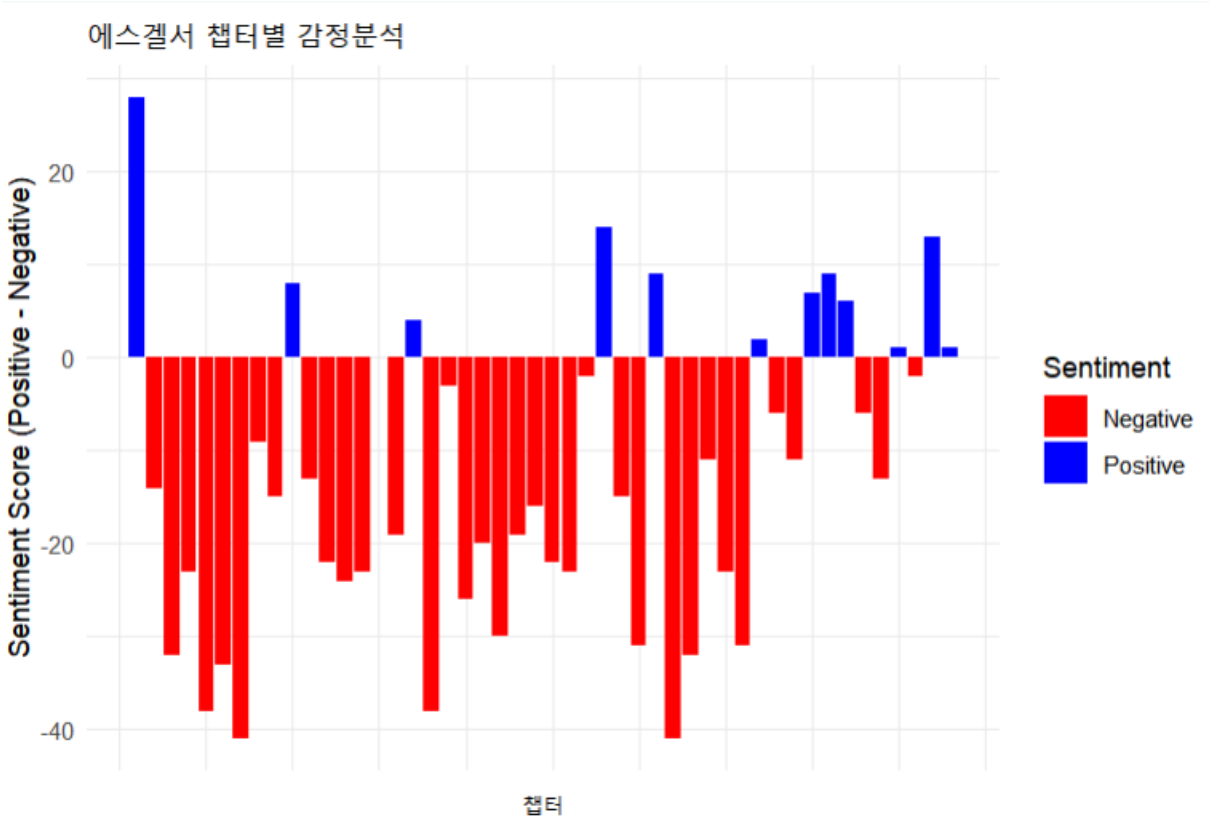
하나님은 바벨론의 멸망을 예고하며, 비록 바벨론이 이스라엘을 심판하는 도구로 사용되었지만, 그들의 죄악 또한 심판을 면치 못할 것임을 예언

.유다와 이스라엘이 회복된 후에도 과거의 죄악을 반복하지 않도록 하나님은 지속적인 경고를 주심

→ 감성분석의 결과와 전체적인 내용의 일치도가 매우 높았음.

에스겔서

```
bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book == "Ezekiel") %>%  # 시편만 필터링
  inner_join(sentiments, by = "word") %>%
  group_by(Chapter, sentiment) %>%  # 챕터별로 그룹화하고 감정별로 나눔
  count() %>%  # 각 감정별로 단어 수 세기
  spread(sentiment, n, fill = 0) %>%  # 긍정과 부정 단어 개수를 나란히 표시
  mutate(sentiment_score = positive - negative) %>%  # 긍정 - 부정으로 감정 점수 계산
  arrange(Chapter)%>%
  ggplot(aes(x = Chapter, y= sentiment_score, fill = sentiment_score > 0))+
  geom_col()+
  labs(title = "에스겔서 챕터별 감정분석",
       x = "챕터",
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sentiment", labels = c("Negative", "Positive"))+
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # X축 텍스트 제거
        axis.ticks.x = element_blank())  # X축 눈금 제거
```



에스겔서는 닥쳐올 재난에 대해서 하나님을 떠난 이스라엘 백성들의 반응을 잘 보여준다. '바벨론의 침공'이라는 재난을 마주하고도 이스라엘 백성은 두가지 반응을 보인다. 1. 현실을 거부한다. 2. 절망에 빠져서 아무것도 보지 못한다.

그러나 에스겔은 이스라엘 백성들이 보지 못했던 것들, 즉 하나님께서 이 상황 가운데에서 일하고 계심을 보았다. 에스겔서는 그 하나님의 일하심을 세밀하고 굵게 그려낸 책이라고 할 수 있다. 책을 먹고, 괴상한 모습의 형상들, 마른뼈들이 되살아남 등등...

그 환상들 가운데 뭔가 극단적이고 무서운 형상들이 있다면 의도와는 상관 없이 부정적인 챕터가 되는 경우가 많았음.

챕터별 감성분석의 결과와 실제 내용을 비교해보니, 감성분석의 한계점이 보였다.

1. **부정적인 단어가 긍정적으로 활용된 경우에도 분석은 부정으로 된다.**
    a. eg: 에스겔서 2장은 현재 부정으로 분류되어 있는데 이는 'do not fear'라고 말씀하는 부분이 많기 때문이다. 그러나 'Do not Fear'에서 Fear이라는 단어는 부정적이나 문장 상에서는 두려워 말라, 안심시키고 위로하려는 의도가 담겨있다. 실제 내용은 부정적이지 않다는 말이다.
2. **감성분석은 문장/ 문단의 의도를 파악하지는 못한다.**

부정적인 단어가 많다고 해서 꼭 부정적인 글이 아니라는 이야기이다.

SNS에서 짧은 댓글이나, 글들을 크롤링해서 트렌드를 분석하기에는 적합하나, 깊은 의미가 담겨 있는 글에서는 글의 내용과 의도가 다를 경우의 가능성이 다분하다. 문맥의 관계를 반영할 수 있는 BERT와 같은 모델을 사용하는 것 이 더 적합할 것 같다.

3. **감성분석을 위한 패키지는 과연 몇 개의 단어들을 분류할 수 있는 것인가?**

   NRC의 경우 14000개, bing의 경우는 6800개밖에 존재하지 않는다. 특히 구어체와 성경에서만 사용하는 salvation과 같은 단어들이 있을 수 있다. 그런 단어들은 적절하게 분류되지 않았을 가능성 역시 고려해야할 것이다.