Data Mining Practice4

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

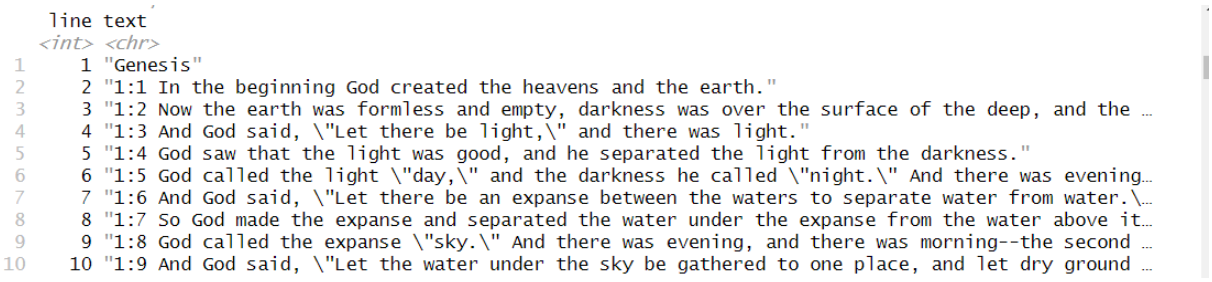
```
library(tidytext)
install.packages("tidytext")
```

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_Bibl
e/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_lines 로 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.

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)
```

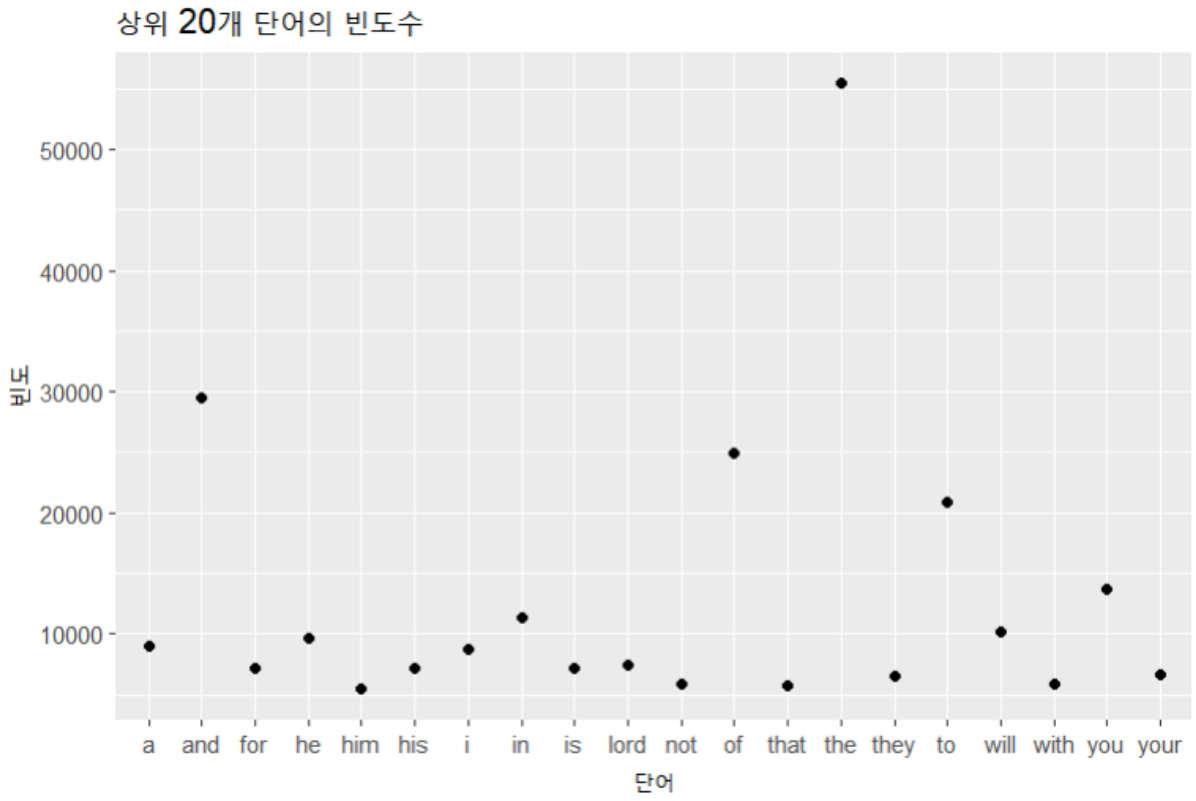
가장 먼저 bible을 토큰화 한 후, 숫자를 세고 정렬함

```
# 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.

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

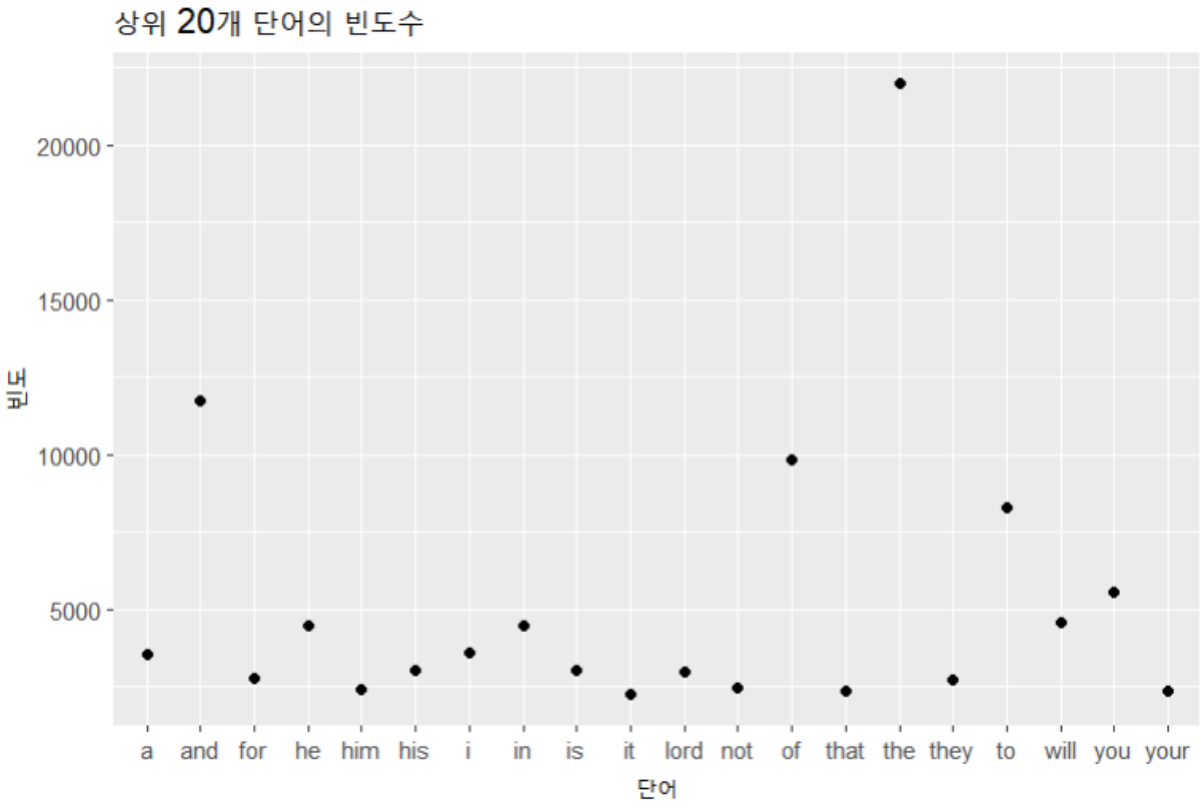
```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book >= "Genesis" & Book <= "Malachi") %>%  # 책의 순서를 기준으로 필터링
  count(word, sort = TRUE)
```

```
# 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
```

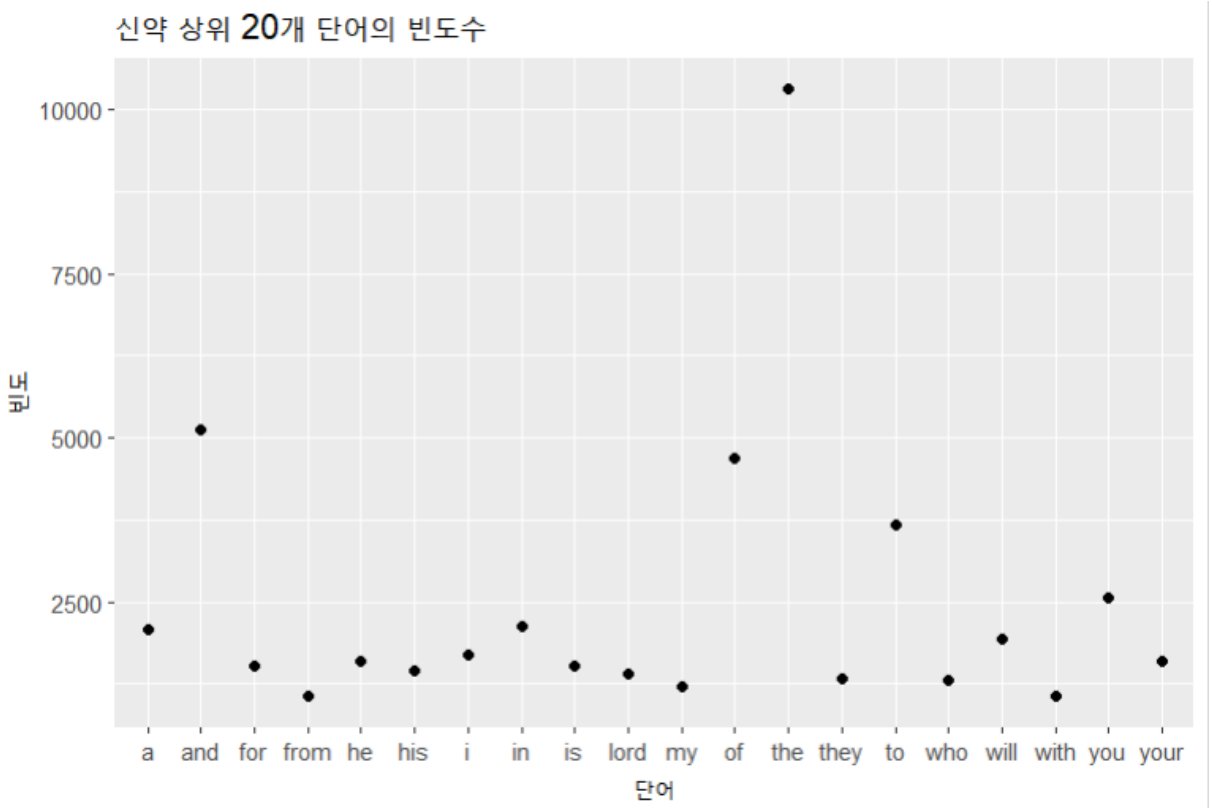
여전히 전체 단어의 빈도수와 매우 유사함을 알 수 있다. 이번에는 이를 시각화하여 살펴보자

```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book >= "Genesis" & Book <= "Malachi") %>%  # 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 >= "Matthew" & Book <= "Revelation") %>%  #
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")
```



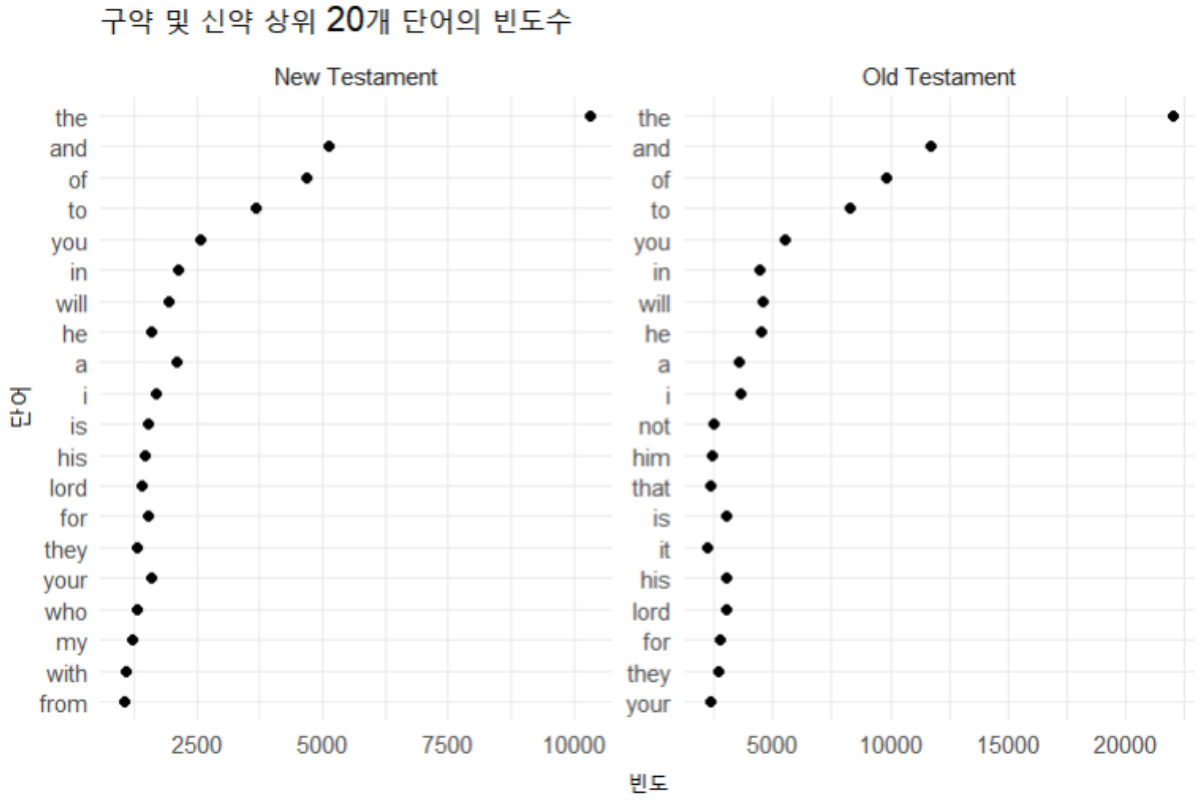
이번에는 두 개의 그래프를 하나로 합쳐보자

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

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

# 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.

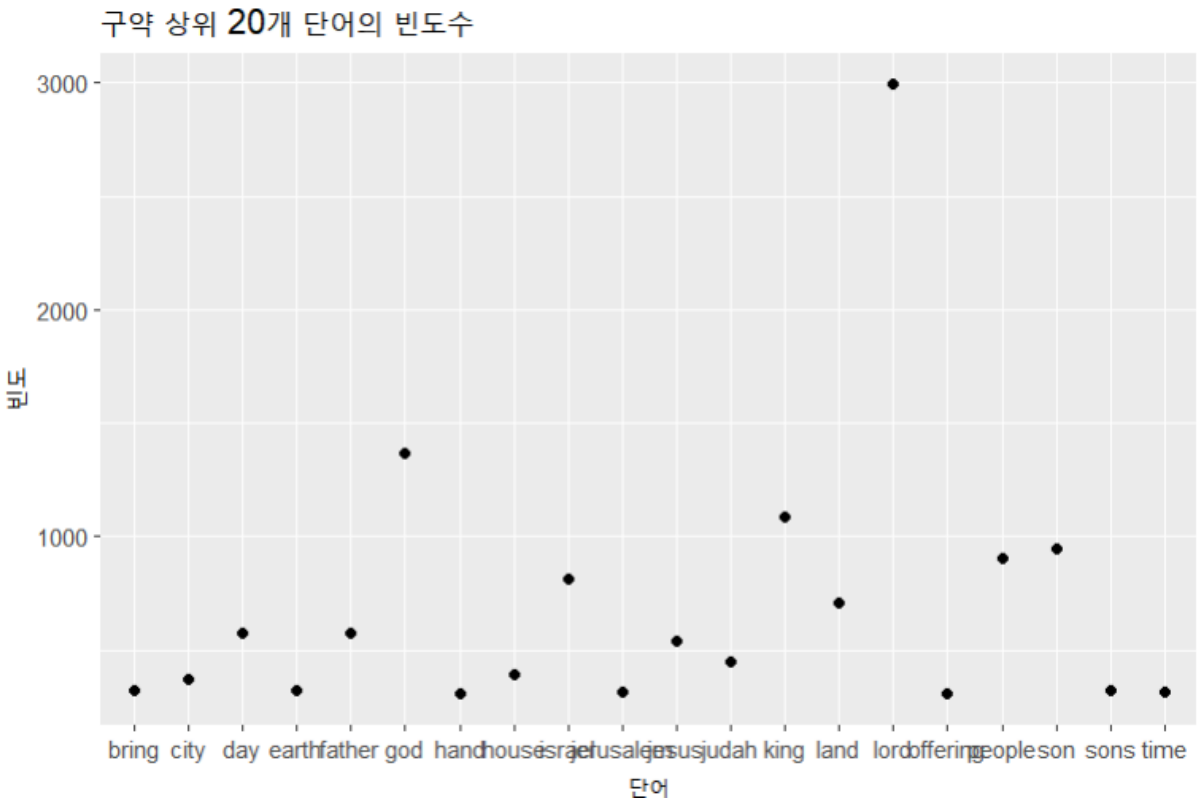
Task2-2 Add the words that should be excluded from the frequency analysis to the stop word list , and apply these in tasks 1 2 and 2 1. (If this was already done in 1

```
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

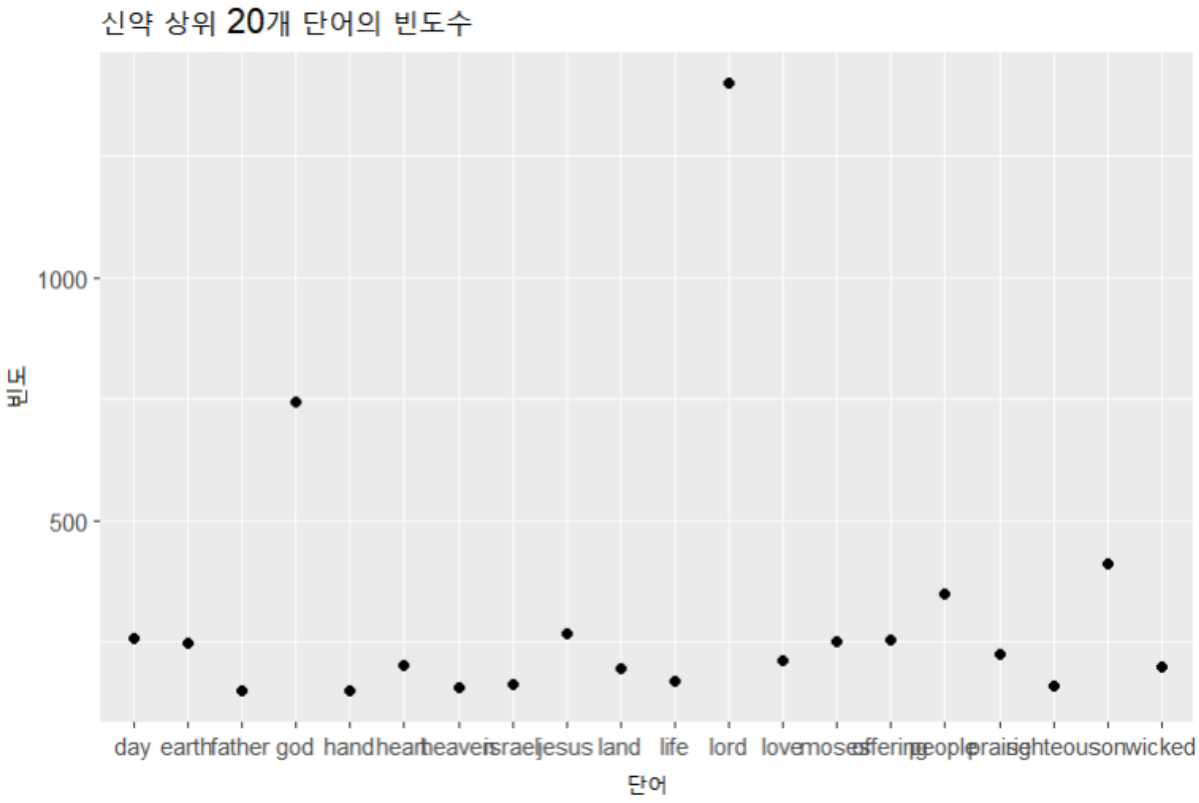
```
bible_df %>%
  unnest_tokens(word, Script)%>%
  filter(Book >= "Genesis" & Book <= "Malachi") %>%  # 책의
순서를 기준으로 필터링
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE)  %>%
  top_n(20, n) %>%
  ggplot(aes(x = word, y = n))+
  geom_point()+
  labs(title = "구약 상위 20개 단어의 빈도수", x = "단어", y =
"빈도")
```

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 >= "Matthew" & Book <= "Revelation") %>%  #
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")
```



동일하게 신약에 대해서도 시각화를 마쳤다. 이제 두 그래프를 하나로 정리해보자

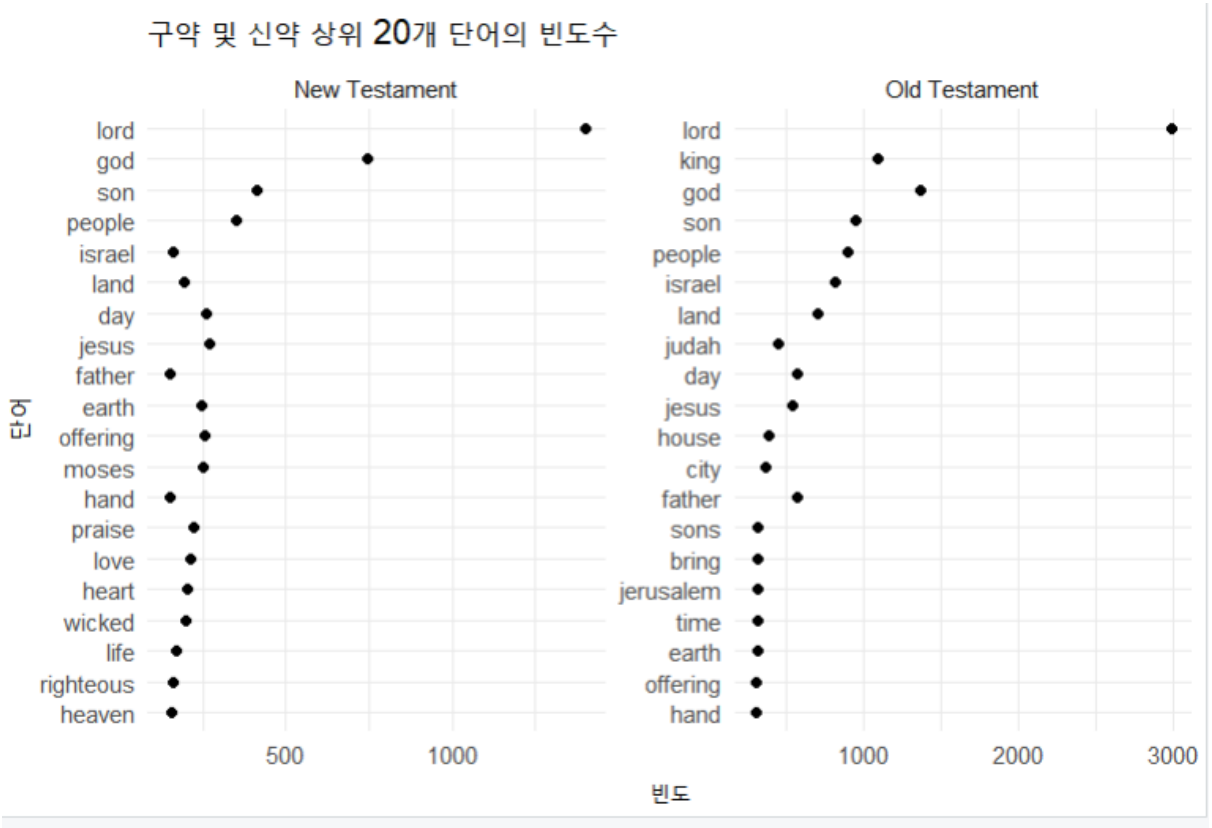
```
ot_top_words1 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book >= "Genesis" & Book <= "Malachi") %>%  # 구약
범위
  anti_join(all_stopwords, by = "word") %>%
  count(word, sort = TRUE) %>%
  top_n(20, n) %>%
  mutate(Testament = "Old Testament")  # 범주 추가 (구약)

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

```r
  mutate(Testament = "New Testament")  # 범주 추가 (신약)

# 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.

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 >= "Genesis" & Book <= "Malachi") %>%
  count(word) %>%
  filter(n >= 10)

nt_words <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book >= "Matthew" & Book <= "Revelation") %>%
  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

```
# 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 innor join으로 합쳐 겹치는 내용만 사용한다

```
# 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

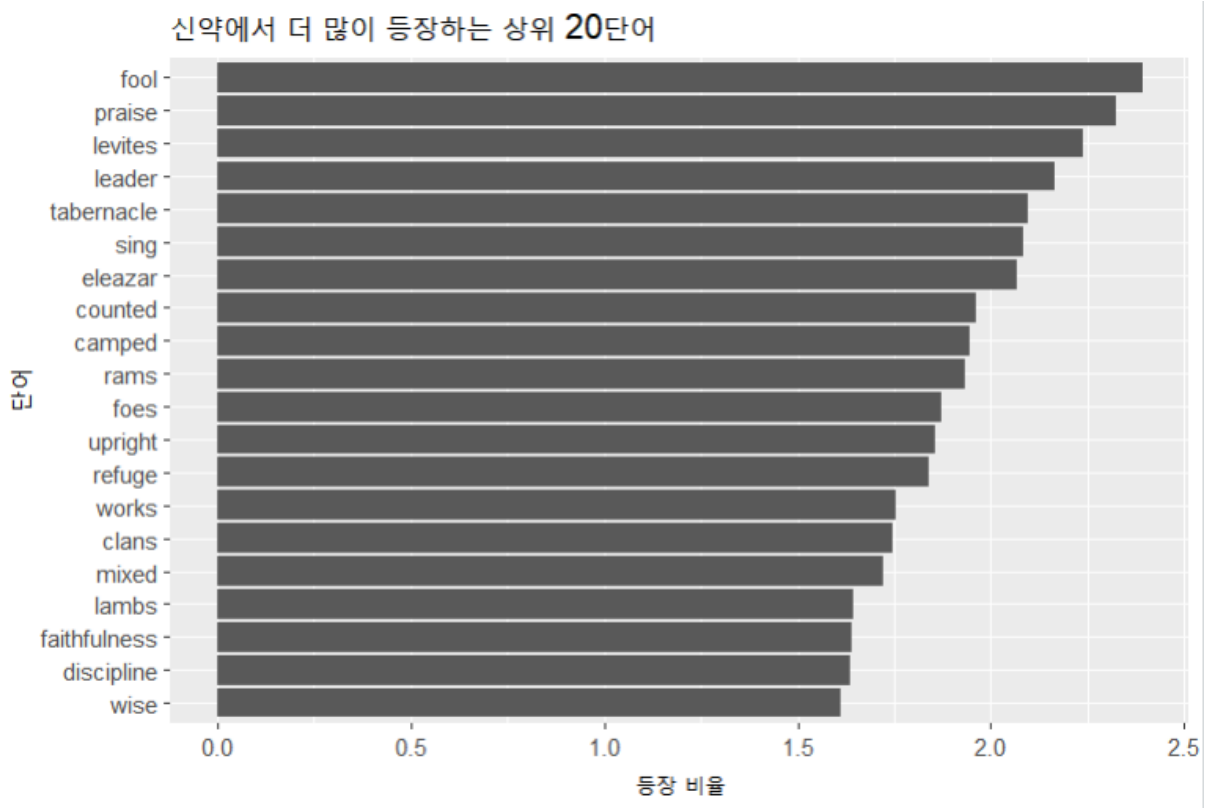
```
# 로그 비율 계산
word_comparison <- word_comparison %>%
  mutate(log_ratio = log(prop_nt / prop_ot))  # 로그 비율 계산
```

이제 상위, 하위 20개를 추출한다

```
# 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)
```

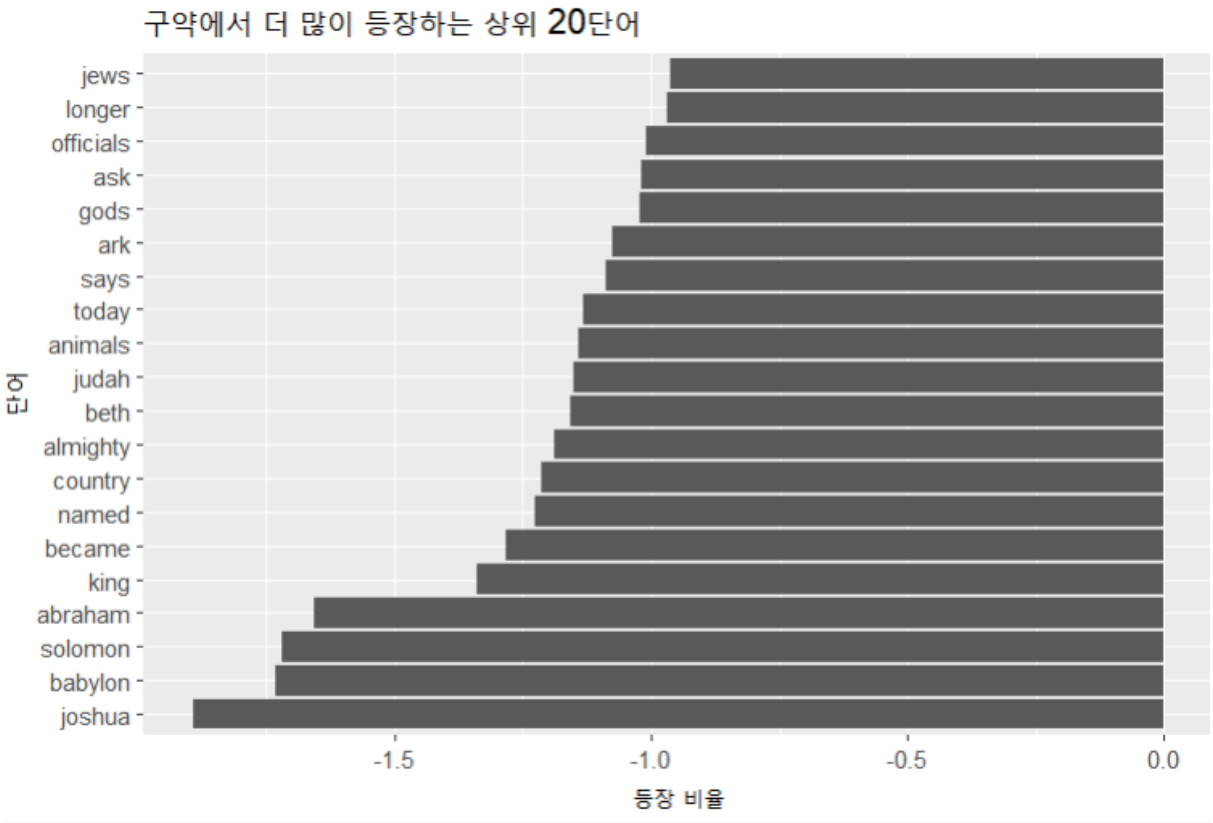
```
top_20%>%
  ggplot(aes(x = reorder(word, log_ratio), y = log_ratio))+
  geom_col()+
  coord_flip() +
  labs(title = "신약에서 더 많이 등장하는 상위 20단어",x = "단어",
y = "등장 비율")
```



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?

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

```r
ot_sentiment <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book >= "Genesis" & Book <= "Malachi") %>%
  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

```
# A tibble: 2 × 2
  sentiment     n
<chr><int>
1 negative   8655
2 positive   6489
```

부정적인 단어는 8655회, 긍정적인 단어는 6489회 등장했다. 신약 역시 이와 동일한 작업을 거치면

```r
nt_sentiment <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book >= "Matthew" & Book <= "Revelation") %>%
  inner_join(sentiments, by = "word") %>%
  count(sentiment)
```

```
# A tibble: 2 × 2
  sentiment     n
<chr><int>
1 negative   4990
2 positive   52473
2 Compare the frequency of sentiment related words between
the New and Old Testaments.
```

신약의 경우에는 긍정적인 단어가 5247, 부정적인 단어가 4990으로 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 >= "Genesis" & Book <= "Malachi") %>%
  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       661
 2 great      241
 3 holy       221
 4 good       207
 5 right      179
 6 love       156
 7 covenant   148
 8 well       148
 9 gold       118
10 heaven     115
11 peace      115
```

우선 구약성경의 긍정적인 단어는 다음과 같다.

```
ot_negative_top10 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book >= "Genesis" & Book <= "Malachi") %>%
  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       196
 2 evil      177
 3 death     150
 4 unclean   142
 5 die       139
 6 anger     120
 7 desert    117
 8 dead      107
 9 fear      103
10 died       99
```

신약에 대해서도 동일하게 확인한 후 두 가지에 대해 시각화를 진행한다

```
nt_positive_top10 <- bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book >= "Matthew" & Book <= "Revelation") %>%
  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 >= "Matthew" & Book <= "Revelation") %>%
  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        426
 2 praise      224
 3 love        214
 4 great       194
 5 good        166
 6 righteous   161
 7 heaven      156
 8 right       127
 9 holy        117
10 work         94
```

```
# A tibble: 10 × 2
   word        n
<chr><int>
 1 wicked    199
 2 evil      130
 3 death     109
 4 sin       102
 5 enemies    97
 6 fear       87
 7 desert     81
 8 trouble    66
 9 poor       64
10 dead       57
```

이제 해당 내용들을 시각화를 해보자

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

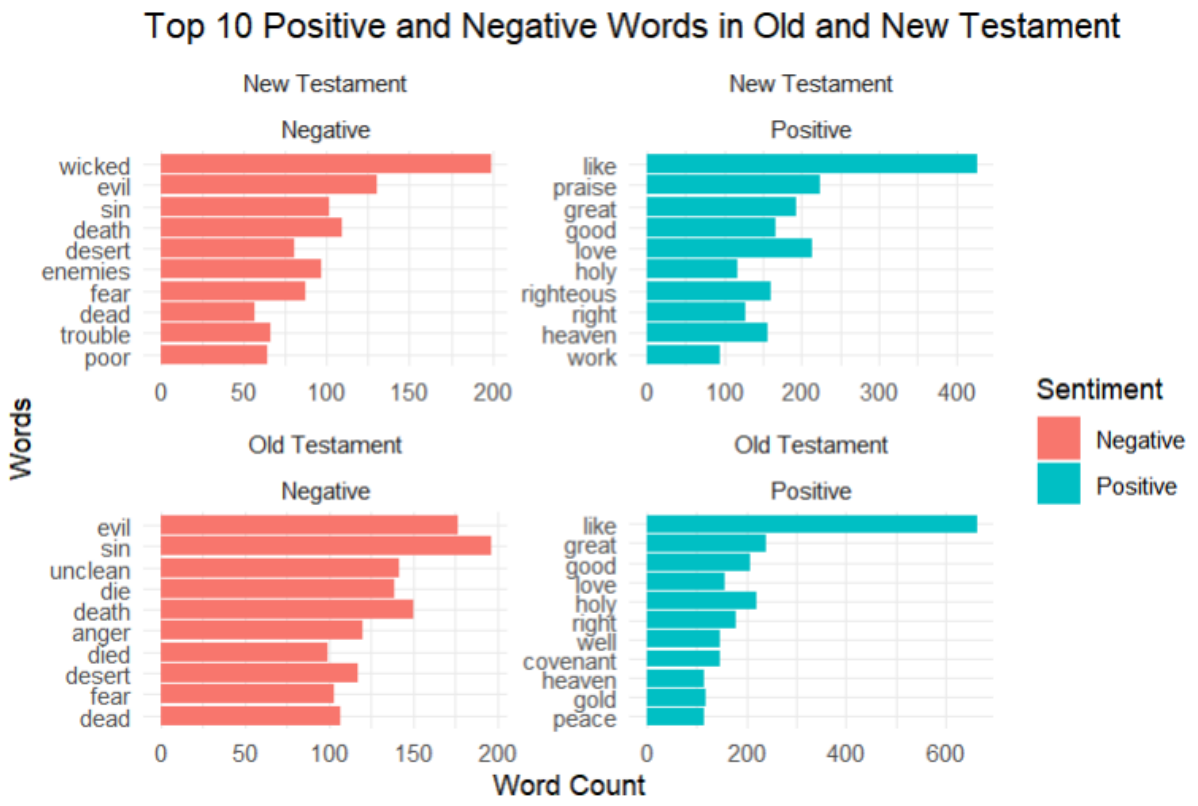
```r
ot_negative_top10 <- ot_negative_top10 %>%
  mutate(Testament = "Old Testament", Sentiment = "Negativ
e")

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
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()
```



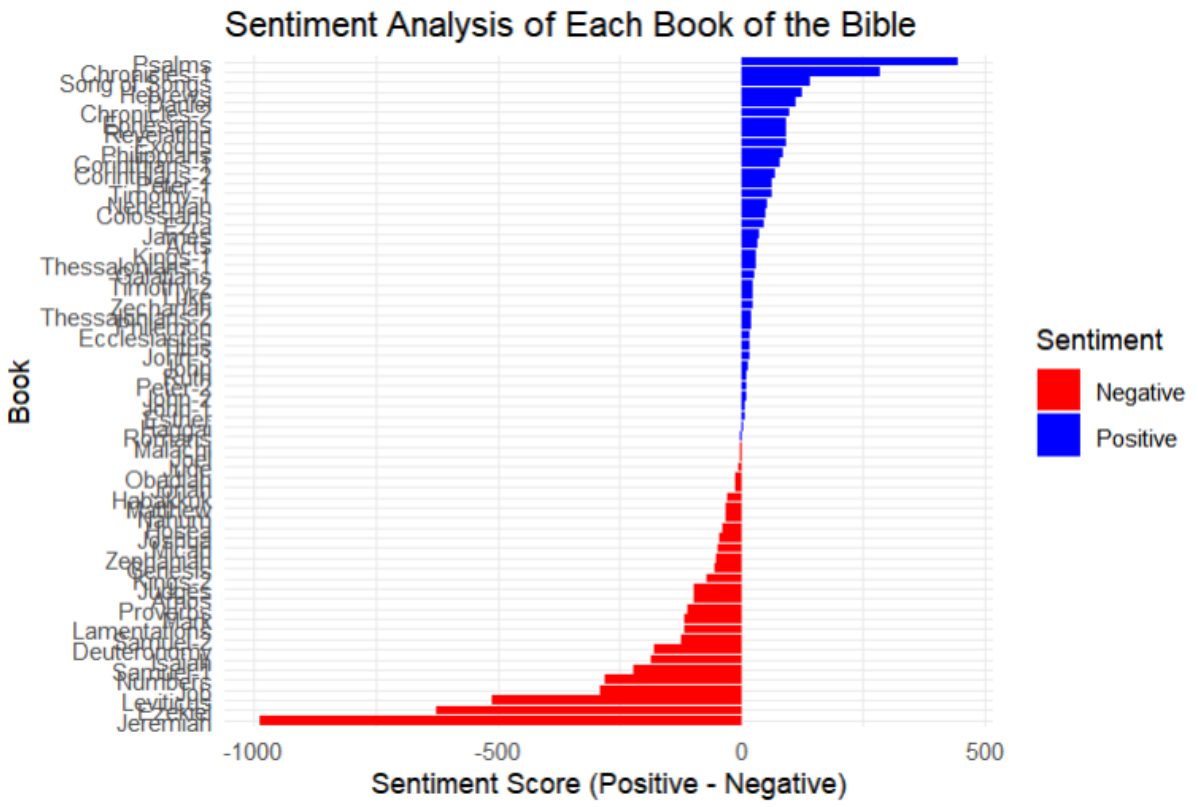
신약과 구약성경 양쪽 모두에서 공통적으로 나타나는 긍정, 부정적 단어가 존재하며 부정적 단어의 경우 evil, sin, die나 death, dead등이 공통적이고, 긍정적 단어에서는 like와 good, great holy등의 단어가 자주 사용되었다.

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 나눔
  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_score), y = sentiment_score, fill = sentiment_score > 0)) +
  geom_col() +
  coord_flip() +
  labs(title = "Sentiment Analysis of Each Book of the Bible",
       x = "Book",
       y = "Sentiment Score (Positive - Negative)") +
  scale_fill_manual(values = c("red", "blue"), name = "Sentiment", labels = c("Negative", "Positive")) +
  theme_minimal()
```



아무래도 전체 책 자체가 너무 많다보니 한눈에 보기 어려운 느낌이다.

Task3-4 C h oose 3~4 books of your interest and p erform sentiment analysis by ch apter 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.

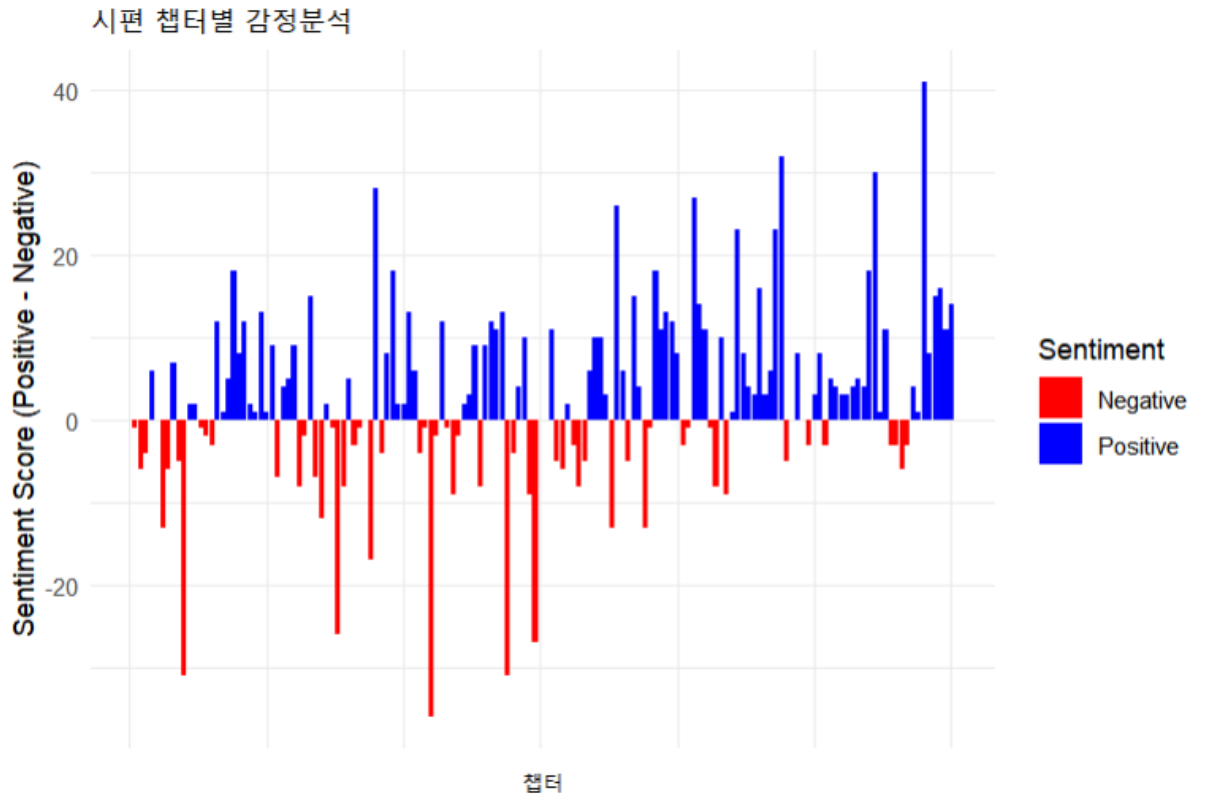
```
  Book     total_words
<chr><int>
1 Psalms         40200
2 Jeremiah       38388
3 Ezekiel        35904
```

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

시편

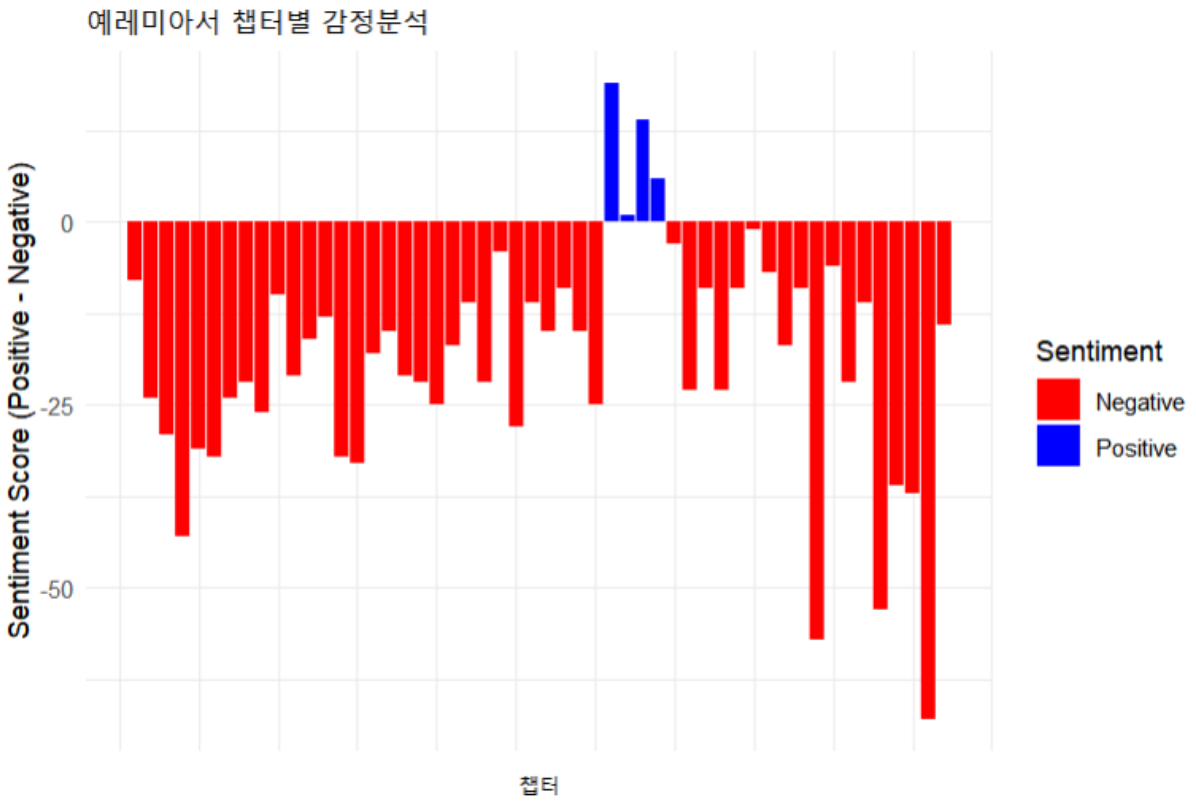
```
bible_df %>%
  unnest_tokens(word, Script) %>%
  filter(Book == "Psalms") %>%  # 시편만 필터링
  inner_join(sentiments, by = "word") %>%
  group_by(Chapter, sentiment) %>%  # 챕터별로 그룹화하고 감정별로 나눔
```

```
  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 = "Sentiment", labels = c("Negative", "Positive"))+
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # X축 텍스트 제거
        axis.ticks.x = element_blank())  # X축 눈금 제거
```



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
```



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 = "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
```