Kabira Pentateuch Blog Post

Introduction

As a person with a ministry background, I find the Bible fascinating at both devotional and scholarly levels. The first five books of the Bible, the Pentateuch, are the foundation of both Jewish and Christian faith traditions. The Pentateuch tells the story of the Israelites becoming a people. It begins at creation, but most of it discusses the period between the Israelites' bondage in Egypt and their entry into the promised land. It was a unique story in the ancient Near East because it outlined a people's relationship with a single god. At that time, people didn't think of the concept of such a relationship.

My questions for this analysis were about the nature of the relationship between the Israelites and God. How does the text describe the evolution of this relationship? What characteristics does the relationship have?

I think this is an important question for people in Christianity because it is no longer assumed that people will have any particular belief system. I believe that looking to this foundational text can help clarify our own beliefs and make sense of our relationships with God, too.

```
# scraping the data and reading it in

gen_1r <- "http://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Genesis%201

gen_1 <- read_html(gen_1r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

gen_2r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Genesis%20

gen_2 <- read_html(gen_2r) %>%
```

```r
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

gen_3r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Genesis%20

gen_3 <- read_html(gen_3r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

gen_4r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Genesis%20

gen_4 <- read_html(gen_4r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

ex_1r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Exodus%201.

ex_1 <- read_html(ex_1r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

ex_2r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Exodus%2019

ex_2 <- read_html(ex_2r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")
```

```
ex_3r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Exodus%2033

ex_3 <- read_html(ex_3r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

lev_1r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Leviticus%

lev_1 <- read_html(lev_1r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

lev_2r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Leviticus%

lev_2 <- read_html(lev_2r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

lev_3r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Leviticus%

lev_3 <- read_html(lev_3r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

num_1r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Numbers%20

num_1 <- read_html(num_1r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

num_2r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Numbers%20

num_2 <- read_html(num_2r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

num_3r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Numbers%20

num_3 <- read_html(num_3r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

deu_1r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Deuteronom

deu_1 <- read_html(deu_1r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")

deu_2r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Deuteronom

deu_2 <- read_html(deu_2r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")
```

```
deu_3r <- "https://bible.oremus.org/?version=NRSV&vnum=YES&headings=YES&passage=Deuteronom

deu_3 <- read_html(deu_3r) %>%
  html_elements(".bibletext p") %>%
  html_text() %>%
  tibble() %>%
  rename(text = ".") %>%
  filter(text!="")
```

```
# aggregating the books by book

gen_whole <- gen_1 %>%
  bind_rows(gen_2, gen_3, gen_4)

ex_whole <- ex_1 %>%
  bind_rows(ex_2, ex_3)

lev_whole <- lev_1 %>%
  bind_rows(lev_2, lev_3)

num_whole <- num_1 %>%
  bind_rows(num_2, num_3)

deu_whole <- deu_1 %>%
  bind_rows(deu_2, deu_3)
```

```
# first pass tidying the data

gen_tidy <- gen_whole %>%
  unnest_tokens(output = word, input = text) %>%
  separate(word, into = c("chap_verse", "word"), sep = "(?<=\\d)(?=\\D)", fill = "left")


ex_tidy <- ex_whole %>%
  unnest_tokens(output = word, input = text) %>%
  separate(word, into = c("chap_verse", "word"), sep = "(?<=\\d)(?=\\D)", fill = "left")


lev_tidy <- lev_whole %>%
  unnest_tokens(output = word, input = text) %>%
  separate(word, into = c("chap_verse", "word"), sep = "(?<=\\d)(?=\\D)", fill = "left")
```

```
num_tidy <- num_whole %>%
  unnest_tokens(output = word, input = text) %>%
  separate(word, into = c("chap_verse", "word"), sep = "(?<=\\d)(?=\\D)", fill = "left")


deu_tidy <- deu_whole %>%
  unnest_tokens(output = word, input = text) %>%
  separate(word, into = c("chap_verse", "word"), sep = "(?<=\\d)(?=\\D)", fill = "left")
```

```
# GENESIS

# addressing issues with data not being read correctly


# 1. creating index so that the corrected data can be re-joined with the rest of the data

gen_tidy$gen_index <- c(1:length(row_number(gen_tidy)))

# 2. finding the columns that have numbers in them
gen_trouble <- gen_tidy %>%
  filter_all(any_vars(word %in% c(1:99)))

# 3. using the NA values in the chap_verse column to identify numbers in the word column
# and then copying the number from the word column to the chap_verse column
gen_trouble2 <- gen_trouble %>%
  mutate(chap_verse = ifelse(is.na(chap_verse), word, chap_verse))

# 4. replacing the numbers in the word colum with a blank space
gen_trouble2 <- gen_trouble2 %>%
  mutate(word = if_else(chap_verse == word, "", word))

# 5. joining the corrected data with the rest of the dataset
gen_trouble3 <-
  left_join(gen_tidy, gen_trouble2, by="gen_index")

# 6. transferring the corrected chap_verse values to the whole dataset's chap_verse column
gen_trouble41 <- gen_trouble3 %>%
  mutate(chap_verse.x = if_else(word.x == chap_verse.y & is.na(chap_verse.x), chap_verse.y
# 7. transferring the corrected word values to the whole dataset's word column
gen_trouble4 <- gen_trouble41 %>%
  mutate(word.x = if_else(word.x %in% c(1:99), "", word.x))

# 8. cleaning up the dataset: renaming chap_verse.x to chap_verse; renaming word.x to word
# deleting chap_verse.y and word.y; and changing chap_verse and gen_index datatypes to num

gen_trouble4 <- gen_trouble4 %>%
  mutate(chap_verse = as.numeric(chap_verse.x), word = word.x, gen_index = as.numeric(gen_
  select(chap_verse, word, gen_index)

# Genesis

# reading in csv of chapter - verse indices
gen_cvind <- read_csv("blog_post_gen_cvindex.csv")
#str(gen_cvind)

# reading in csv of section headings
gen_head <- read_csv("genesis_headings.csv")
#str(gen_head)

# assigning an incrementing value for each chap_verse entry
gen_tidy_cv <- gen_trouble4 %>%
  mutate(c_v_index = as.numeric(cumsum(grepl("^[-]{0,1}[0-9]{0,}.{0,1}[0-9]{1,}$", chap_ve

# joining csv chapter chapter-verse indices to incremented value to add chapter and verse
gen_tidy_cv <-
  left_join(gen_tidy_cv, gen_cvind, by='c_v_index')

# joining csv section headings on chapter and verse
gen_tidy_cv <-
  left_join(gen_tidy_cv, gen_head, by= c("chapter"="chapter", "verse"="verse"))

# filling in section headings for each word and adding book name
gen_tidy_cv2 <- gen_tidy_cv %>%
  fill(heading, .direction="downup") %>%
  mutate(document='genesis')



new_gen <- gen_tidy_cv2 %>%
  select(document, word, heading, chapter, verse)
#head(new_gen)
```

```
#EXODUS

# addressing issues with data not being read correctly


# 1. creating index so that the corrected data can be re-joined with the rest of the data

ex_tidy$ex_index <- c(1:length(row_number(ex_tidy)))

# 2. finding the columns that have numbers in them
ex_trouble <- ex_tidy %>%
  filter_all(any_vars(word %in% c(1:99)))

# 3. using the NA values in the chap_verse column to identify numbers in the word column
# and then copying the number from the word column to the chap_verse column
ex_trouble2 <- ex_trouble %>%
  mutate(chap_verse = ifelse(is.na(chap_verse), word, chap_verse))

# 4. replacing the numbers in the word colum with a blank space
ex_trouble2 <- ex_trouble2 %>%
  mutate(word = if_else(chap_verse == word, "", word))

# 5. joining the corrected data with the rest of the dataset
ex_trouble3 <-
  left_join(ex_tidy, ex_trouble2, by="ex_index")

# 6. transferring the corrected chap_verse values to the whole dataset's chap_verse column
ex_trouble41 <- ex_trouble3 %>%
  mutate(chap_verse.x = if_else(word.x == chap_verse.y & is.na(chap_verse.x), chap_verse.y

# 7. transferring the corrected word values to the whole dataset's word column
ex_trouble4 <- ex_trouble41 %>%
  mutate(word.x = if_else(word.x %in% c(1:99), "", word.x))

# 8. cleaning up the dataset: renaming chap_verse.x to chap_verse; renaming word.x to word
# deleting chap_verse.y and word.y; and changing chap_verse and ex_index datatypes to nume

ex_trouble4 <- ex_trouble4 %>%
  mutate(chap_verse = as.numeric(chap_verse.x), word = word.x, ex_index = as.numeric(ex_in
  select(chap_verse, word, ex_index)


# Exodus
# reading in csv of chapter - verse indices
ex_cvind <- read_csv("blog_post_ex_cvindex.csv")
#str(ex_cvind)

# reading in csv of section headings
ex_head <- read_csv("exodus_headings.csv")
#str(ex_head)

# assigning an incrementing value for each chap_verse entry
ex_tidy_cv <- ex_trouble4 %>%
  mutate(c_v_index = as.numeric(cumsum(grepl("^[-]{0,1}[0-9]{0,}.{0,1}[0-9]{1,}$", chap_ve

# joining csv chapter chapter-verse indices to incremented value to add chapter and verse
ex_tidy_cv <-
  left_join(ex_tidy_cv, ex_cvind, by='c_v_index')

# joining csv section headings on chapter and verse
ex_tidy_cv <-
  left_join(ex_tidy_cv, ex_head, by= c("chapter"="chapter", "verse"="verse"))

# filling in section headings for each word and adding book name
ex_tidy_cv2 <- ex_tidy_cv %>%
  fill(heading, .direction="downup") %>%
  mutate(document='exodus')


new_ex <- ex_tidy_cv2 %>%
  select(document, word, heading, chapter, verse)
#head(new_ex)
```

```
# LEVITICUS

# addressing issues with data not being read correctly


# 1. creating index so that the corrected data can be re-joined with the rest of the data
```

```
lev_tidy$lev_index <- c(1:length(row_number(lev_tidy)))

# 2. finding the columns that have numbers in them
lev_trouble <- lev_tidy %>%
  filter_all(any_vars(word %in% c(1:99)))

# 3. using the NA values in the chap_verse column to identify numbers in the word column
# and then copying the number from the word column to the chap_verse column
lev_trouble2 <- lev_trouble %>%
  mutate(chap_verse = ifelse(is.na(chap_verse), word, chap_verse))

# 4. replacing the numbers in the word colum with a blank space
lev_trouble2 <- lev_trouble2 %>%
  mutate(word = if_else(chap_verse == word, "", word))

# 5. joining the corrected data with the rest of the dataset
lev_trouble3 <-
  left_join(lev_tidy, lev_trouble2, by="lev_index")

# 6. transferring the corrected chap_verse values to the whole dataset's chap_verse colum
lev_trouble41 <- lev_trouble3 %>%
  mutate(chap_verse.x = if_else(word.x == chap_verse.y & is.na(chap_verse.x), chap_verse.y

# 7. transferring the corrected word values to the whole dataset's word column
lev_trouble4 <- lev_trouble41 %>%
  mutate(word.x = if_else(word.x %in% c(1:99), "", word.x))

# 8. cleaning up the dataset: renaming chap_verse.x to chap_verse; renaming word.x to word
# deleting chap_verse.y and word.y; and changing chap_verse and lev_index datatypes to num

lev_trouble4 <- lev_trouble4 %>%
  mutate(chap_verse = as.numeric(chap_verse.x), word = word.x, lev_index = as.numeric(lev_
  select(chap_verse, word, lev_index)



# Leviticus
# reading in csv of chapter - verse indices
lev_cvind <- read_csv("blog_post_lev_cvindex.csv")
#str(lev_cvind)
```

```
# reading in csv of section headings
lev_head <- read_csv("leviticus_headings.csv")
#str(lev_head)

# assigning an incrementing value for each chap_verse entry
lev_tidy_cv <- lev_trouble4 %>%
  mutate(c_v_index = as.numeric(cumsum(grepl("^[-]{0,1}[0-9]{0,}.{0,1}[0-9]{1,}$", chap_ve

# joining csv chapter chapter-verse indices to incremented value to add chapter and verse
lev_tidy_cv <-
  left_join(lev_tidy_cv, lev_cvind, by='c_v_index')

# joining csv section headings on chapter and verse
lev_tidy_cv <-
  left_join(lev_tidy_cv, lev_head, by= c("chapter"="chapter", "verse"="verse"))

# filling in section headings for each word and adding book name
lev_tidy_cv2 <- lev_tidy_cv %>%
  fill(heading, .direction="downup") %>%
  mutate(document='leviticus')


new_lev <- lev_tidy_cv2 %>%
  select(document, word, heading, chapter, verse)
#head(new_lev)
```

```
# NUMBERS
# addressing issues with data not being read correctly


# 1. creating index so that the corrected data can be re-joined with the rest of the data

num_tidy$num_index <- c(1:length(row_number(num_tidy)))

# 2. finding the columns that have numbers in them
num_trouble <- num_tidy %>%
  filter_all(any_vars(word %in% c(1:99)))

# 3. using the NA values in the chap_verse column to identify numbers in the word column
# and then copying the number from the word column to the chap_verse column
num_trouble2 <- num_trouble %>%
  mutate(chap_verse = ifelse(is.na(chap_verse), word, chap_verse))

# 4. replacing the numbers in the word colum with a blank space
num_trouble2 <- num_trouble2 %>%
  mutate(word = if_else(chap_verse == word, "", word))

# 5. joining the corrected data with the rest of the dataset
num_trouble3 <-
  left_join(num_tidy, num_trouble2, by="num_index")

# 6. transferring the corrected chap_verse values to the whole dataset's chap_verse column
num_trouble41 <- num_trouble3 %>%
  mutate(chap_verse.x = if_else(word.x == chap_verse.y & is.na(chap_verse.x), chap_verse.y

# 7. transferring the corrected word values to the whole dataset's word column
num_trouble4 <- num_trouble41 %>%
  mutate(word.x = if_else(word.x %in% c(1:99), "", word.x))

# 8. cleaning up the dataset: renaming chap_verse.x to chap_verse; renaming word.x to word
# deleting chap_verse.y and word.y; and changing chap_verse and num_index datatypes to num

num_trouble4 <- num_trouble4 %>%
  mutate(chap_verse = as.numeric(chap_verse.x), word = word.x, num_index = as.numeric(num_
  select(chap_verse, word, num_index)


# Numbers
# reading in csv of chapter - verse indices
num_cvind <- read_csv("blog_post_num_cvindex.csv")
#str(num_cvind)

# reading in csv of section headings
num_head <- read_csv("numbers_headings.csv")
#str(num_head)

# assigning an incrementing value for each chap_verse entry
num_tidy_cv <- num_trouble4 %>%
  mutate(c_v_index = as.numeric(cumsum(grepl("^[-]{0,1}[0-9]{0,}.{0,1}[0-9]{1,}$", chap_ve

# joining csv chapter chapter-verse indices to incremented value to add chapter and verse
num_tidy_cv <-
```

```
  left_join(num_tidy_cv, num_cvind, by='c_v_index')

# joining csv section headings on chapter and verse
num_tidy_cv <-
  left_join(num_tidy_cv, num_head, by= c("chapter"="chapter", "verse"="verse"))

# filling in section headings for each word and adding book name
num_tidy_cv2 <- num_tidy_cv %>%
  fill(heading, .direction="downup") %>%
  mutate(document='numbers')


new_num <- num_tidy_cv2 %>%
  select(document, word, heading, chapter, verse)
#head(new_num)
```

```
# DEUTERONOMY

# addressing issues with data not being read correctly


# 1. creating index so that the corrected data can be re-joined with the rest of the data

deu_tidy$deu_index <- c(1:length(row_number(deu_tidy)))

# 2. finding the columns that have numbers in them
deu_trouble <- deu_tidy %>%
  filter_all(any_vars(word %in% c(1:99)))

# 3. using the NA values in the chap_verse column to identify numbers in the word column
# and then copying the number from the word column to the chap_verse column
deu_trouble2 <- deu_trouble %>%
  mutate(chap_verse = ifelse(is.na(chap_verse), word, chap_verse))

# 4. replacing the numbers in the word colum with a blank space
deu_trouble2 <- deu_trouble2 %>%
  mutate(word = if_else(chap_verse == word, "", word))

# 5. joining the corrected data with the rest of the dataset
deu_trouble3 <-
  left_join(deu_tidy, deu_trouble2, by="deu_index")
```

```
# 6. transferring the corrected chap_verse values to the whole dataset's chap_verse column
deu_trouble41 <- deu_trouble3 %>%
  mutate(chap_verse.x = if_else(word.x == chap_verse.y & is.na(chap_verse.x), chap_verse.y

# 7. transferring the corrected word values to the whole dataset's word column
deu_trouble4 <- deu_trouble41 %>%
  mutate(word.x = if_else(word.x %in% c(1:99), "", word.x))

# 8. cleaning up the dataset: renaming chap_verse.x to chap_verse; renaming word.x to word
# deleting chap_verse.y and word.y; and changing chap_verse and deu_index datatypes to num

deu_trouble4 <- deu_trouble4 %>%
  mutate(chap_verse = as.numeric(chap_verse.x), word = word.x, deu_index = as.numeric(deu_
  select(chap_verse, word, deu_index)


# Deuteronomy
# reading in csv of chapter - verse indices
deu_cvind <- read_csv("blog_post_deu_cvindex.csv")
#str(deu_cvind)

# reading in csv of section headings
deu_head <- read_csv("deuteronomy_headings.csv")
#str(deu_head)

# assigning an incrementing value for each chap_verse entry
deu_tidy_cv <- deu_trouble4 %>%
  mutate(c_v_index = as.numeric(cumsum(grepl("^[-]{0,1}[0-9]{0,}.{0,1}[0-9]{1,}$", chap_ve

# joining csv chapter chapter-verse indices to incremented value to add chapter and verse
deu_tidy_cv <-
  left_join(deu_tidy_cv, deu_cvind, by='c_v_index')

# joining csv section headings on chapter and verse
deu_tidy_cv <-
  left_join(deu_tidy_cv, deu_head, by= c("chapter"="chapter", "verse"="verse"))

# filling in section headings for each word and adding book name
deu_tidy_cv2 <- deu_tidy_cv %>%
  fill(heading, .direction="downup") %>%
  mutate(document='deuteronomy')
```

```
new_deu <- deu_tidy_cv2 %>%
  select(document, word, heading, chapter, verse)
#head(new_deu)
```

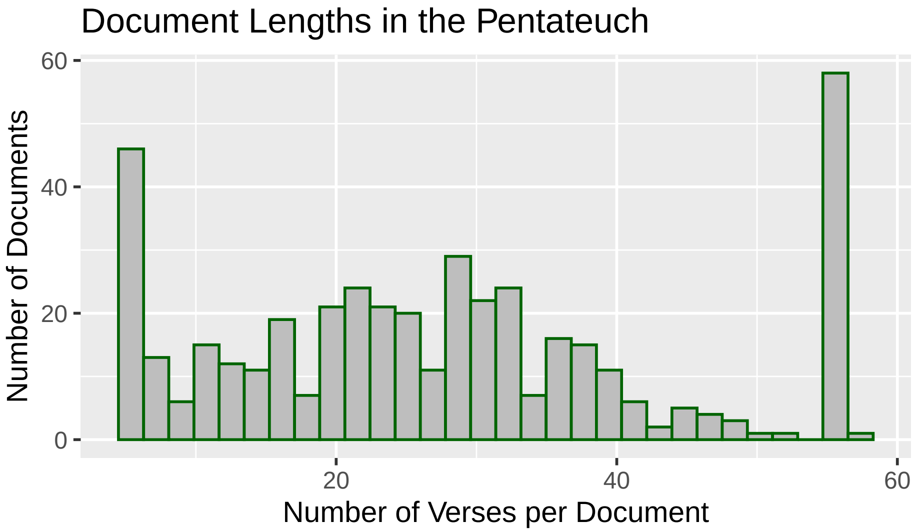
```
# making a full pentateuch document

pent_raw <- new_gen %>%
  bind_rows(new_ex, new_lev, new_num, new_deu)

# writing out the full pentateuch document to save time for future analysis
write_csv(pent_raw, "full_raw_pentateuch.csv")
```

```
# reading in the clean file
pent_raw <-read_csv("full_raw_pentateuch.csv")
```

```
# changing the data type of the document number
pent_extra <- pent_raw %>%
  mutate(doc_no = as.numeric(case_when(document=="genesis" ~ 1,
                                      document=="exodus" ~ 2,
                                      document=="leviticus" ~ 3,
                                      document=="numbers" ~ 4,
                                      document=="deuteronomy" ~ 5)))

# counting number of verses per document
pent_sum <- pent_extra %>%
  group_by(doc_no, chapter, verse) %>%
  summarize(n=n())


# adding a sequential line number
pent_sum$head_id <- c(1:5852)

# joining the dataframes to have a master list of the documents
pent_ref1 <-
  left_join(pent_extra, pent_sum, by = c("doc_no"="doc_no", "chapter"="chapter", "verse"="
pent_ref <- pent_ref1 %>%
  mutate(head_find = if_else(lag(heading, n=1) == heading, 0, 1)) %>%
  mutate(head_find = if_else(is.na(head_find), 1, head_find)) %>%
  mutate(bible_book_no = doc_no) %>%
  filter(head_find==1)
```

```
colnames(pent_ref)[7] <- 'doc_length'
```

```
pent_ref_gr <-pent_ref %>%
  ggplot() +
  geom_histogram(data=pent_ref,aes(x=doc_length), color="darkgreen", fill="grey") +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  labs(title="Document Lengths in the Pentateuch", x="Number of Verses per Document", y="N
pent_ref_gr
```



```
#ggsave("pent_ref.jpg", plot=pent_ref_gr)

summary(pent_ref[ , 'doc_length'])
```

```
  doc_length
Min.   : 5.00
1st Qu.:16.50
Median :27.00
Mean   :27.68
3rd Qu.:36.00
Max.   :57.00
```

```
word_count<-pent_ref1 %>%
  group_by(heading) %>%
  summarize(n=n())
colnames(word_count)[2] <- 'num_words'
```

Data Description

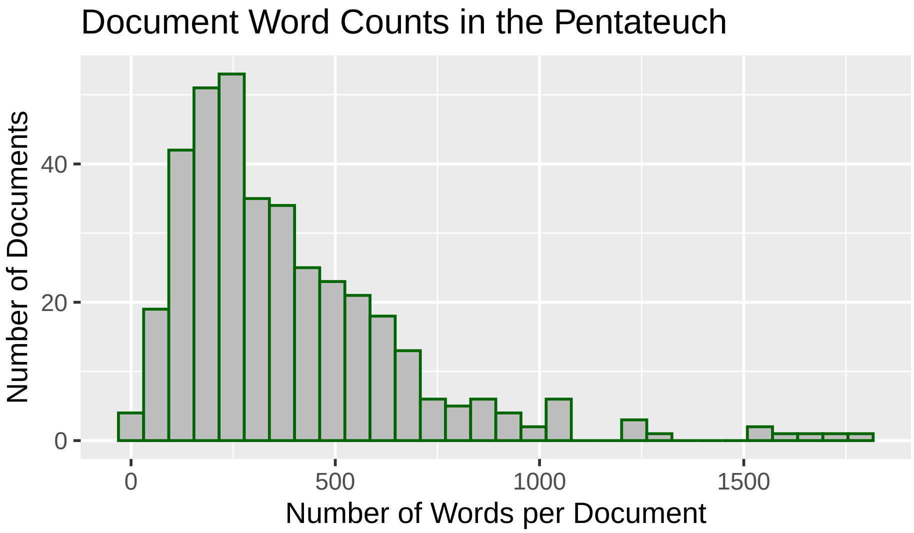
The data was obtained from bible.oremus.org. It is all of the text of the first five books of the Bible, which together are referred to as the Pentateuch. The texts are from the New Revised Standard Version translation, which is considered to be one of the top scholarly translations.

In addition to the Bible text that was downloaded, I added section headings as a more logical way to break up the text. The section headings were not available from the oremus website so I entered them into a csv file by hand using the headings provided in a physical copy of the NRSV Bible. The section headings are my 'documents' for this project. Some summary statistics and graphs are shown below. There is a wide variety in the numbers of words per document, as well as the number of verses per document.

Documents have anywhere from 5 to 57 verses, with a median of 27 verses. The word counts sound a bit odd because the counts exclude stopwords: documents have as few as 17 words and as many as 1803 words. The median number of words per document is 317.

Because of the high variance in numbers of words and numbers of verses per document, I used weightings for much of my analysis, which you will find below.

```
word_count_gr <- word_count %>%
  ggplot() +
  geom_histogram(data=word_count,aes(x=num_words), color="darkgreen", fill="grey") +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  labs(title="Document Word Counts in the Pentateuch", x="Number of Words per Document",
  y="Number of Documents")

word_count_gr
```



```
summary(word_count[ , 'num_words'])
```

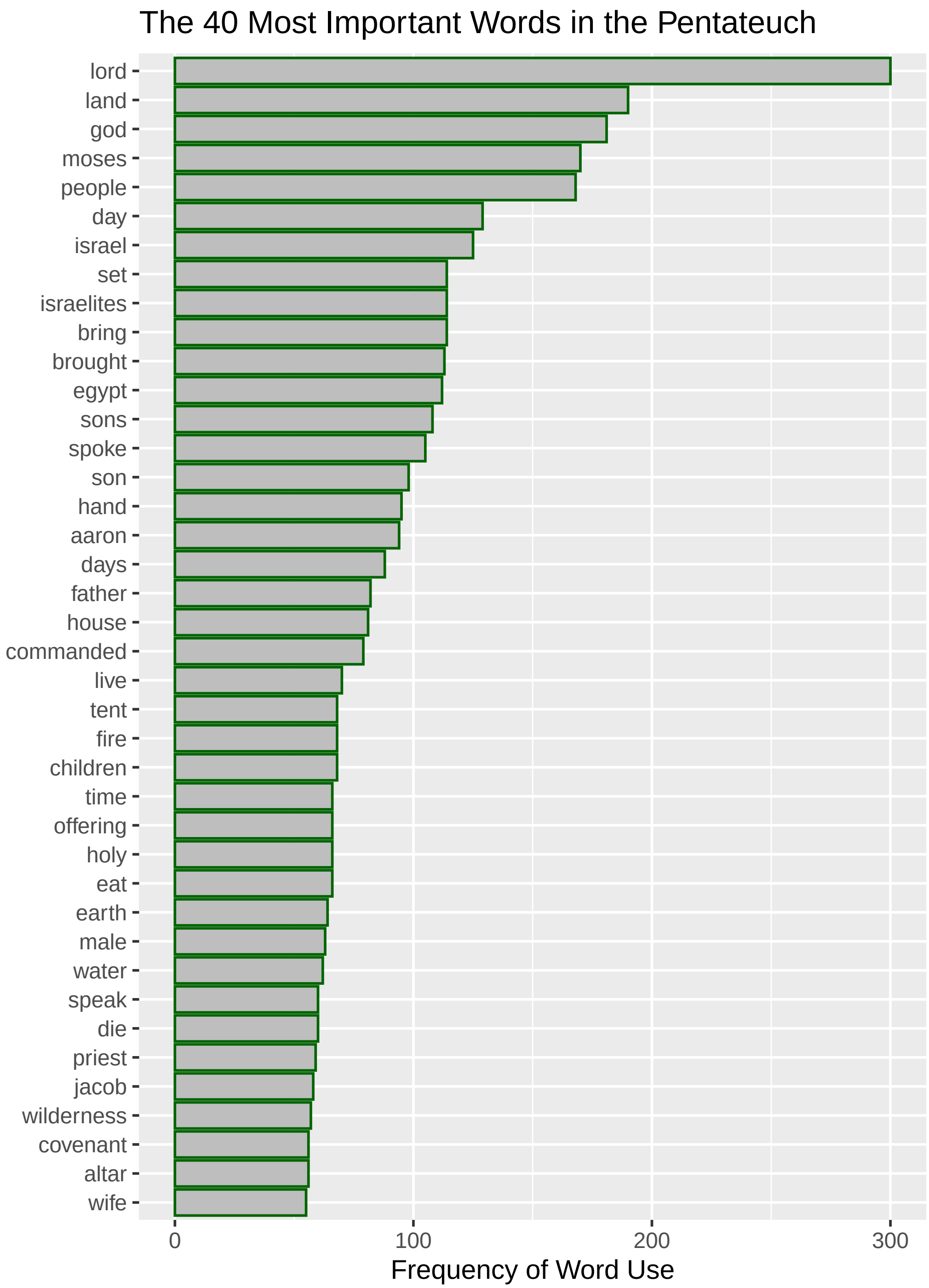
```
  num_words     
 Min.   :  17.0  
 1st Qu.: 193.0  
 Median : 317.0  
 Mean   : 388.7  
 3rd Qu.: 511.0  
 Max.   :1803.0  
```

```
pent_join <- pent_ref %>%
  group_by(bible_book_no, head_id, heading) %>%
  summarize(n=n())

#ggsave("word_count.jpg", plot=word_count_gr)
```

```
# removing bible book names, chapters, and verses for analysis purposes
pent_mod <- pent_raw %>%
  filter(word!="") %>%
  select(word, heading)
```

```
# removing stop words and blank words
pent_clean <- pent_mod %>%
  anti_join(stop_words) %>%
  count(heading, word, sort = T)
```

```
pent_clean_gr<-pent_clean %>%
  count(word, sort=T) %>%
  top_n(n=40, wt = n) %>%
  ggplot() +
  geom_col(aes(x=n, y=reorder(word, n)), color="darkgreen", fill="grey") +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  labs(title="The 40 Most Important Words in the Pentateuch", x="Frequency of Word Use", y
pent_clean_gr
```


The 40 Most Important Words in the Pentateuch
lord
land
god
moses
people
day
israel
set
israelites
bring
brought
egypt
sons
spoke
son
hand
aaron
days
father
house
commanded
live
tent
fire
children
time
offering
holy
eat
earth
male
water
speak
die
priest
jacob
wilderness
covenant
altar
wife
0
100
200
300
Frequency of Word Use

```
#ggsave("forty.jpg", plot=pent_clean_gr)
```

```
# TOPIC MODELING
# prep - casting into DTM
pent_dtm <- pent_clean %>%
  cast_dtm(document = heading,
           term = word,
           value = n)
```

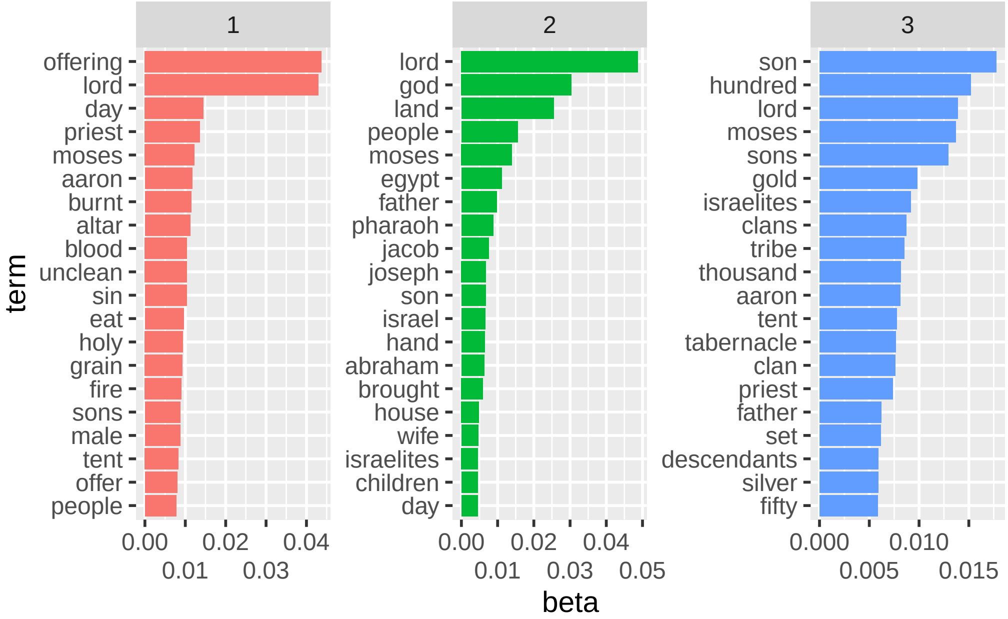
Data Analysis

I began my analysis with topic modeling and tried 3, 4, 5, and 6-topic models. The 3 and 4-topic models did not capture enough detail. The 6-topic model started to break up the topics in a strange way.

The 5-topic model ended up being the best one, as will be explained below.

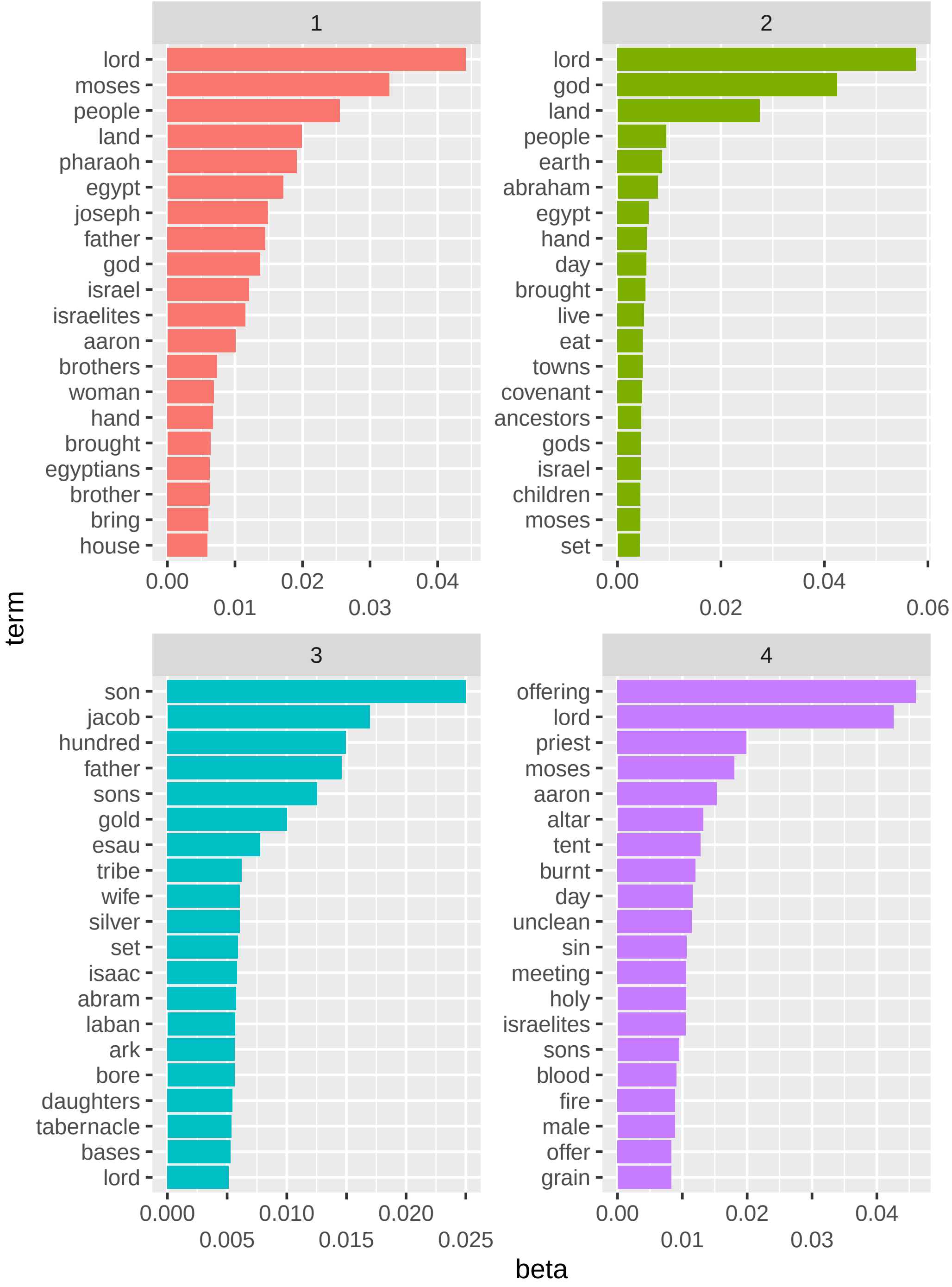
I chose to do topic modeling in order to find out what the authors of the texts emphasized.

```
# part 1 betas (word-topic)
# 3 Topic model
pent_lda_3 <- LDA(pent_dtm,k = 3,
              control = list(seed = 1233))

pent_terms_3 <- tidy(pent_lda_3, matrix = "beta")

pent_terms_3_graph <- pent_terms_3 %>%
  group_by(topic) %>%
  slice_max(beta, n =20) %>%
  ungroup() %>%
  arrange(topic,-beta) %>%
  mutate(term = reorder_within(term,beta,topic)) %>%
  ggplot() +
  geom_col(aes(x = beta, y = term, fill = factor(topic)), show.legend=FALSE) +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered()

pent_terms_3_graph
```



```
#ggsave("pent_3.jpg", plot=pent_terms_3_graph)
```

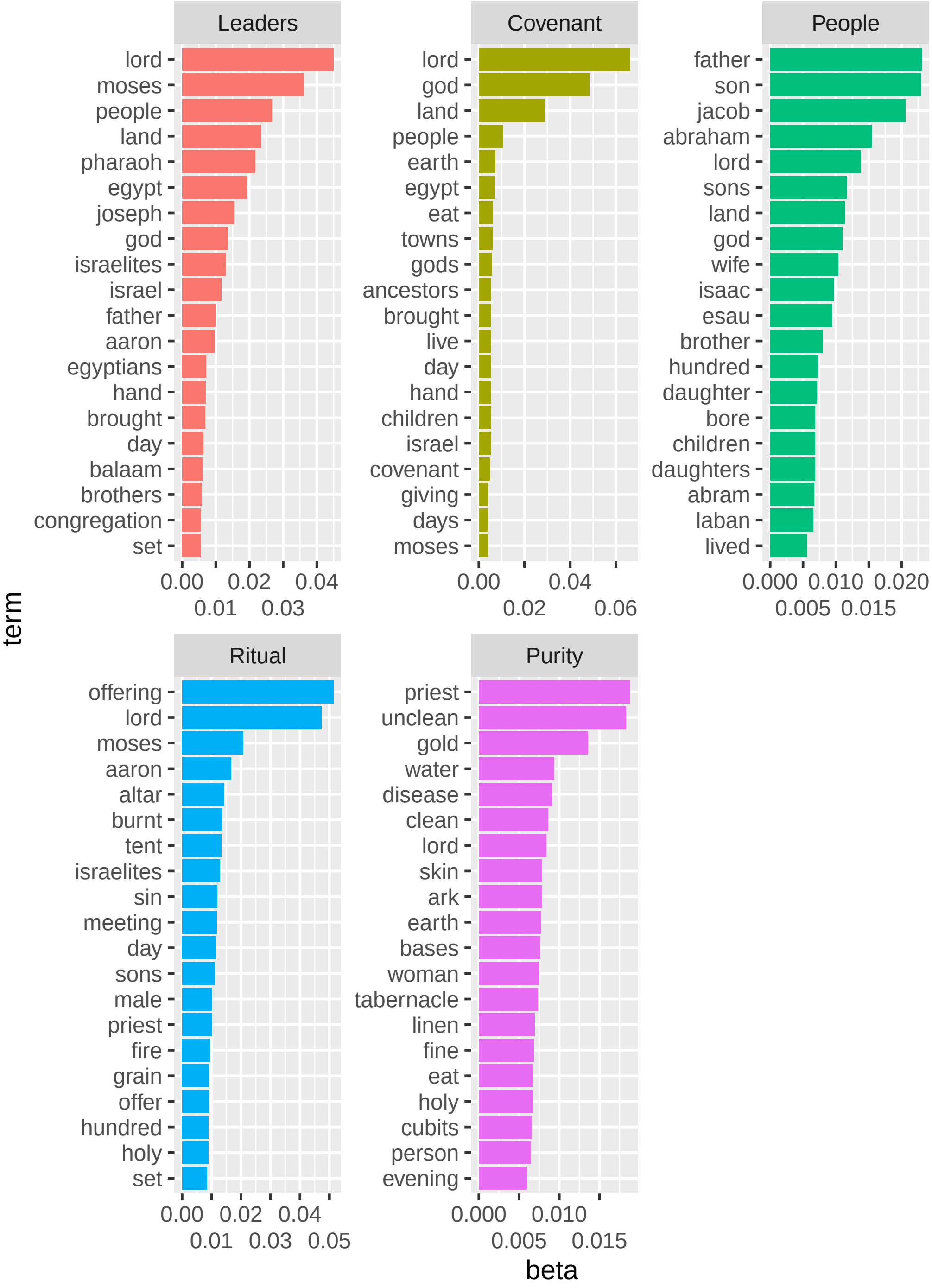
```
# 4 Topic model
pent_lda_4 <- LDA(pent_dtm,k = 4,
              control = list(seed = 1234))

pent_terms_4 <- tidy(pent_lda_4, matrix = "beta")

pent_terms_4_graph <- pent_terms_4 %>%
  group_by(topic) %>%
  slice_max(beta, n =20) %>%
  ungroup() %>%
  arrange(topic,-beta) %>%
  mutate(term = reorder_within(term,beta,topic)) %>%
  ggplot() +
  geom_col(aes(x = beta, y = term, fill = factor(topic)), show.legend=FALSE) +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered()

pent_terms_4_graph
```


1
lord
moses
people
land
pharaoh
egypt
joseph
father
god
israel
israelites
aaron
brothers
woman
hand
brought
egyptians
brother
bring
house
0.00
0.01
0.02
0.03
0.04
2
lord
god
land
people
earth
abraham
egypt
hand
day
brought
live
eat
towns
covenant
ancestors
gods
israel
children
moses
set
0.00
0.02
0.04
0.06
term
3
son
jacob
hundred
father
sons
gold
esau
tribe
wife
silver
set
isaac
abram
laban
ark
bore
daughters
tabernacle
bases
lord
0.000
0.005
0.010
0.015
0.020
0.025
4
offering
lord
priest
moses
aaron
altar
tent
burnt
day
unclean
sin
meeting
holy
israelites
sons
blood
fire
male
offer
grain
0.00
0.01
0.02
0.03
0.04
beta

```
#ggsave("pent_4.jpg", plot=pent_terms_4_graph)
```

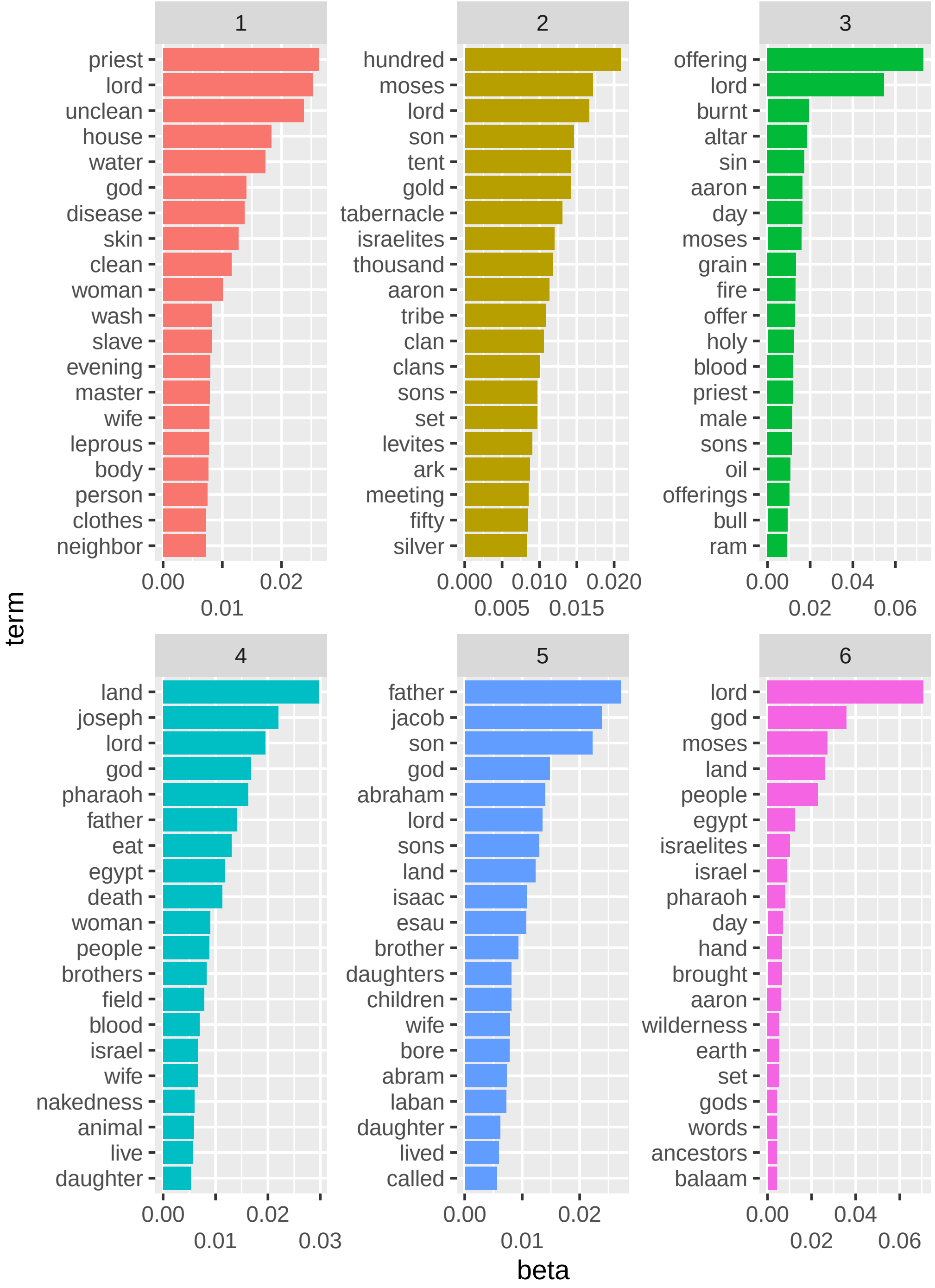
I used prior subject matter knowledge to determine the topics. The topics address the main concerns of the needs of the ancient Israelites. The covenant topic is about God's relationship with the Israelites. The leaders topic discusses the leaders that were chosen for the Israelites. There was a lot of push and pull between the leaders and the people. The people topic includes much of the 'history' and ancestry narratives. The ritual topic describes the actions of and requirements for the religious leaders. Finally, the purity topic addresses the requirements for the people in general.

```
# 5 Topic model - This is the model chosen for analysis
pent_lda_5 <- LDA(pent_dtm,k = 5,
               control = list(seed = 1235))

pent_terms_5 <- tidy(pent_lda_5, matrix = "beta")

pent_terms_5_graph <- pent_terms_5 %>%
  group_by(topic) %>%
  slice_max(beta, n =20) %>%
  ungroup() %>%
  arrange(topic,-beta) %>%
  mutate(term = reorder_within(term,beta,topic)) %>%
  mutate(topic = factor(topic)) %>%
  mutate(topic =
    fct_recode(topic,
               "Leaders" = "1",
               "Covenant" = "2",
               "People" = "3",
               "Ritual" = "4",
               "Purity" = "5")) %>%
  ggplot() +
  geom_col(aes(x = beta, y = term, fill = factor(topic)), show.legend=FALSE) +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered()

pent_terms_5_graph
```


Leaders
lord
moses
people
land
pharaoh
egypt
joseph
god
israelites
israel
father
aaron
egyptians
hand
brought
day
balaam
brothers
congregation
set
0.00
0.01
0.02
0.03
0.04
Covenant
lord
god
land
people
earth
egypt
eat
towns
gods
ancestors
brought
live
day
hand
children
israel
covenant
giving
days
moses
0.00
0.02
0.04
0.06
People
father
son
jacob
abraham
lord
sons
land
god
wife
isaac
esau
brother
hundred
daughter
bore
children
daughters
abram
laban
lived
0.000
0.005
0.010
0.015
0.020
term
Ritual
offering
lord
moses
aaron
altar
burnt
tent
israelites
sin
meeting
day
sons
male
priest
fire
grain
offer
hundred
holy
set
0.00
0.01
0.02
0.03
0.04
0.05
Purity
priest
unclean
gold
water
disease
clean
lord
skin
ark
earth
bases
woman
tabernacle
linen
fine
eat
holy
cubits
person
evening
0.000
0.005
0.010
0.015
beta

```
#ggsave("pent_5.jpg", plot=pent_terms_5_graph)
```

```
# 6 Topic model
pent_lda_6 <- LDA(pent_dtm,k = 6,
              control = list(seed = 1236))

pent_terms_6 <- tidy(pent_lda_6, matrix = "beta")

pent_terms_6_graph <- pent_terms_6 %>%
  group_by(topic) %>%
  slice_max(beta, n =20) %>%
  ungroup() %>%
  arrange(topic,-beta) %>%
  mutate(term = reorder_within(term,beta,topic)) %>%
  ggplot() +
  geom_col(aes(x = beta, y = term, fill = factor(topic)), show.legend=FALSE) +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered()

pent_terms_6_graph
```


1
priest
lord
unclean
house
water
god
disease
skin
clean
woman
wash
slave
evening
master
wife
leprous
body
person
clothes
neighbor
0.00
0.01
0.02
2
hundred
moses
lord
son
tent
gold
tabernacle
israelites
thousand
aaron
tribe
clan
clans
sons
set
levites
ark
meeting
fifty
silver
0.000
0.005
0.010
0.015
0.020
3
offering
lord
burnt
altar
sin
aaron
day
moses
grain
fire
offer
holy
blood
priest
male
sons
oil
offerings
bull
ram
0.00
0.02
0.04
0.06
4
land
joseph
lord
god
pharaoh
father
eat
egypt
death
woman
people
brothers
field
blood
israel
wife
nakedness
animal
live
daughter
0.00
0.01
0.02
0.03
5
father
jacob
son
god
abraham
lord
sons
land
isaac
esau
brother
daughters
children
wife
bore
abram
laban
daughter
lived
called
0.00
0.01
0.02
6
lord
god
moses
land
people
egypt
israelites
israel
pharaoh
day
hand
brought
aaron
wilderness
earth
set
gods
words
ancestors
balaam
0.00
0.02
0.04
0.06
term
beta

```
#ggsave("pent_6.jpg", plot=pent_terms_6_graph)
```

```
# part 2 gammas (document-topic)
pent_documents <- tidy(pent_lda_5,matrix = "gamma")
```

I used the gamma values to characterize each document. In addition, I weighted the gamma values according to how strongly they described the document. If the value for the highest gamma was over .85, I rated it Strong and gave it a 1 weighting. If it was .7 - .85, I rated it as Moderate and gave it a .7 weighting. If it was .51 to .7 I rated it as Weak and gave it a .51 weighting. The weightings show up in the heatmap below, with the stronger weightings leading to darker colors.

The heatmap is meant to show the evolution of the writers' concerns through the Pentateuch. As expected, Genesis is mostly concerned with the people, the stories of the Israelites' ancestry. Exodus, Leviticus, and Numbers continue the narrative, but the focus is on the Israelites' time in the wilderness when they were learning how to become free people. Therefore, the concerns are largely about leadership, ritual, and purity. How do they become a nation of people in relationship with God? Deuteronomy shifts strongly into the covenant topic. This is the book where the Israelites are instructed about how to live up to their covenant responsibilities.

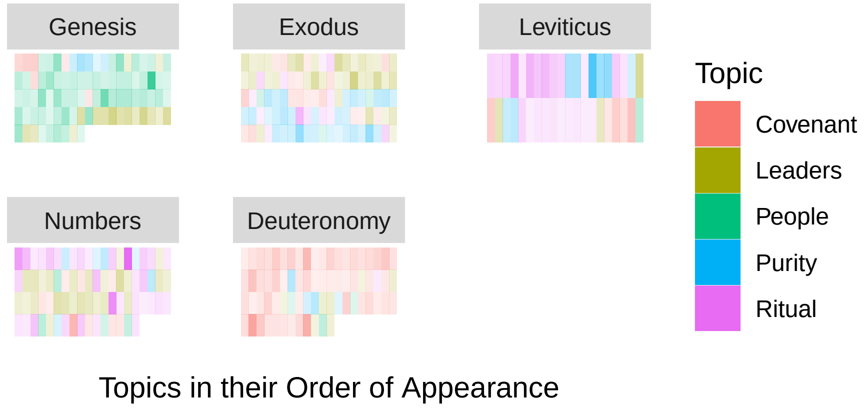
```
# pivoting the gamma values wider


wide_gamma <- pivot_wider(pent_documents, names_from=topic, values_from = gamma)
wide_gamma <- wide_gamma %>%
  mutate(gamma_sel = colnames(wide_gamma[,2:6])[max.col(wide_gamma[,2:6])]) %>%
  mutate(row_maximum = apply(wide_gamma[,-1], 1, max)) %>%
  mutate(gamma_name = case_when(gamma_sel == 1 ~ "Leaders",
                                gamma_sel == 2 ~ "Covenant",
                                gamma_sel == 3 ~ "People",
                                gamma_sel == 4 ~ "Ritual",
                                gamma_sel == 5 ~ "Purity")) %>%
  mutate(gamma_strength_name = case_when(row_maximum >= .85 ~ "Strong",
                                    row_maximum < .85 & row_maximum >= .7 ~ "Moderate",
                                    TRUE ~ "Weak")) %>%
  mutate(gamma_strength = as.numeric(case_when(row_maximum >= .85 ~ "1",
                                    row_maximum < .85 & row_maximum >= .7 ~ ".7",
                                    TRUE ~ ".51")))


# joining the pent_join ordering document to the gamma designation document
```

```
ordered_gamma <-
  merge(wide_gamma, pent_join, by.x = "document", by.y = "heading")
```

```
# putting the documents in order to display them
ordered_gamma_2 <- ordered_gamma[order(ordered_gamma$head_id), ] %>%
  select(head_id, bible_book_no, gamma_sel, gamma_name, document, gamma_strength, gamma_st
  mutate(doc_weight = lead(head_id, n=1) - head_id) %>%
  mutate(doc_weight = if_else(head_id == 5841, 12, doc_weight)) %>%
  mutate(bible_book_no = as.factor(bible_book_no))

ordered_gamma_2$doc_order <- c(1:377)

#ordered_gamma_2 %>%
#  group_by(bible_book_no, doc_order) %>%
#  summarize(n=n())

# manually formatting the heatmaps so they display correctly
ordered_gamma_2_2 <- ordered_gamma_2%>%
  mutate(doc_row = as.numeric(case_when(doc_order %in% c(1:20) ~ 1,
                                        doc_order %in% c(21:40) ~ 2,
                                        doc_order %in% c(41:60) ~ 3,
                                        doc_order %in% c(61:80) ~ 4,
                                        doc_order %in% c(81:89) ~ 5,
                                        doc_order %in% c(90:109) ~ 6,
                                        doc_order %in% c(121:140) ~ 7,
                                        doc_order %in% c(141:160) ~ 8,
                                        doc_order %in% c(161:180) ~ 9,
                                        doc_order %in% c(181:200) ~ 10,
                                        doc_order %in% c(201:220) ~ 11,
                                        doc_order %in% c(221:240) ~ 12,
                                        doc_order %in% c(241:260) ~ 13,
                                        doc_order %in% c(261:280) ~ 14,
                                        doc_order %in% c(281:300) ~ 15,
                                        doc_order %in% c(301:320) ~ 16,
                                        doc_order %in% c(321:340) ~ 17,
                                        doc_order %in% c(341:360) ~ 18,
                                        doc_order %in% c(361:377) ~ 19)))

ordered_gamma_2_2$doc_col <- c(1:20, 1:20, 1:20, 1:20, 1:20, 1:20,1:20, 1:20, 1:20,
                             1:20, 1:20, 1:20,1:20, 1:20, 1:20,1:20, 1:20, 1:20,
                             1:17)
```

```
# Gen 1-89
# Ex 90-189
# Lev 190-229
# Num 230-305
# Deu 306-377


ordered_gamma_3 <- ordered_gamma_2%>%
  mutate(doc_row = as.numeric(case_when(doc_order %in% c(1:20) ~ 1,
                                        doc_order %in% c(21:40) ~ 2,
                                        doc_order %in% c(41:60) ~ 3,
                                        doc_order %in% c(61:80) ~ 4,
                                        doc_order %in% c(81:89) ~ 5,
                                        doc_order %in% c(90:109) ~ 6,
                                        doc_order %in% c(110:129) ~ 7,
                                        doc_order %in% c(130:149) ~ 8,
                                        doc_order %in% c(150:169) ~ 9,
                                        doc_order %in% c(170:189) ~ 10,
                                        doc_order %in% c(190:209) ~ 11,
                                        doc_order %in% c(210:229) ~ 12,
                                        doc_order %in% c(230:249) ~ 13,
                                        doc_order %in% c(250:269) ~ 14,
                                        doc_order %in% c(270:289) ~ 15,
                                        doc_order %in% c(290:305) ~ 16,
                                        doc_order %in% c(306:325) ~ 17,
                                        doc_order %in% c(326:345) ~ 18,
                                        doc_order %in% c(346:365) ~ 19,
                                        doc_order %in% c(366:377) ~ 20)))

ordered_gamma_3$doc_col <- c(1:20, 1:20, 1:20, 1:20, 1:9, 1:20,1:20, 1:20, 1:20,
                             1:20, 1:20, 1:20,1:20, 1:20, 1:20,1:16, 1:20, 1:20,
                             1:20,1:12)



gamma_plot <- ordered_gamma_3 %>%
  mutate(bible_book_no =
    fct_recode(bible_book_no,
               "Genesis" = "1",
               "Exodus" = "2",
               "Leviticus" = "3",
               "Numbers" = "4",
               "Deuteronomy" = "5")) %>%
  ggplot(aes(y=doc_row, x=doc_col, fill=gamma_name)) +
  geom_tile(aes(alpha=doc_weight*gamma_strength)) +
  guides(alpha="none") +
  scale_y_reverse() +
  scale_fill_discrete(name="Topic") +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme(plot.title.position = "panel") +
  theme(panel.background = element_rect(fill = NA)) +
  theme(axis.text = element_text(colour = NA)) +
  theme(axis.ticks = element_line(linewidth=NA)) +
  labs(title="    The Pentateuch by Topic", x = "Topics in their Order of Appearance",
       y=NULL, caption="       Darker values indicate stronger emphasis") +
  theme(plot.caption = element_text(size=8, face = "italic", hjust=0, margin=margin(t=20))
  theme(axis.title.x = element_text(margin=margin(t=0))) +
  labs(legend = "Topic")+
  facet_wrap(~bible_book_no, scales="free") # without this line, one big heatmap [need to

gamma_plot
```

The Pentateuch by Topic



Darker values indicate stronger emphasis

```
#ggsave("gamma.jpg", plot=gamma_plot)
```

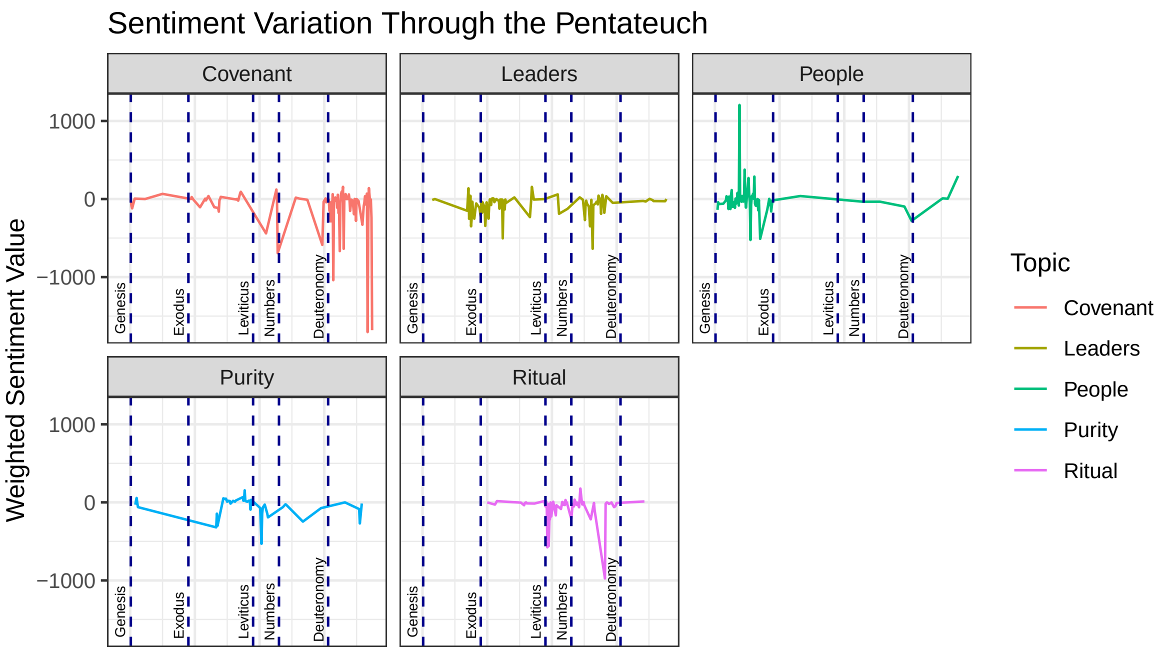
The second analysis I did was sentiment analysis. I wanted to see how the sentiments expressed in the text compared from topic to topic, as well as how the sentiments changed over time for each topic. I used the afinn sentiments in order to use the numeric scales for comparison purposes.

I assigned new sentiment values to quite a few words because the words' meanings for general use today don't capture the values from the Biblical text. Also, in assigning the weights of the sentiments I took the number of verses in the documents into consideration. This means that longer documents are weighted more heavily than shorter ones. I also used the gamma weights that I assigned above in order to try to be more accurate about the likely intensity of the sentiments. The "value weight" variable I created is the number of verses * the gamma weight * the afinn value.

On the whole, the distribution of the sentiments across the topics is fairly uniform. The graphs of the sentiments for each topic over time show considerable variety.

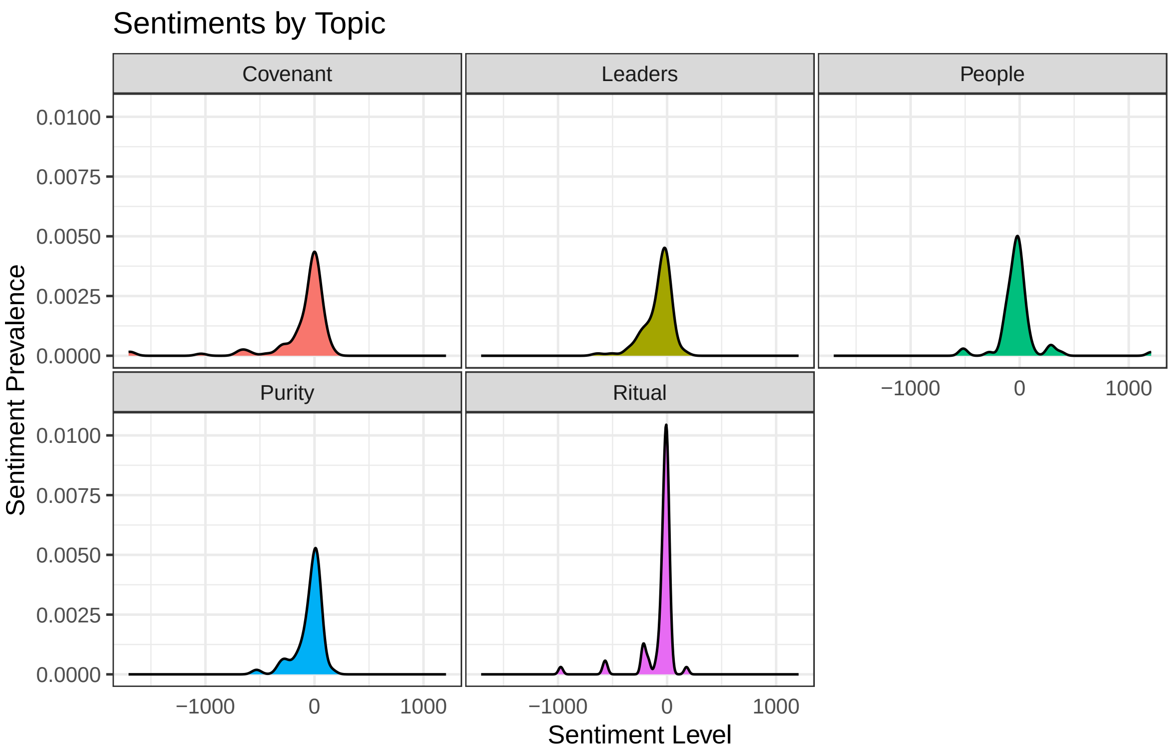
```
# SENTIMENT ANALYSIS

P_sent <-pent_clean %>%
  inner_join(get_sentiments("afinn"), by = "word",
             relationship = "many-to-many") %>%
  count(heading, word, value)

# recoding some words for afinn - this is based on knowledge of the texts
val_4 <- c("god", "promise", "promised", "spirit", "pray", "restore", "forgive", "restored
              "faith", "free", "heaven", "bless", "mercy", "glory", "redeemed", "steadfast
              "pardon", "acquit", "rescue", "blesses", "blessing", "praise", "vindicate",
           "absolve", "absolved", "pardoned", "beloved", "freedom", "save", "compassionate
val_3 <- c("treasure", "safe", "zealous", "favor", "entrusted", "integrity", "authority",
           "comfort", "honored", "justice", "grant", "worth", "secure", "generous", "accep
val_2 <- c("desire", "increase", "visions", "vision", "capable", "embrace")
val_1 <- c("loose", "fire", "swear")
val_0 <- c("nuts", "ass")
val_neg_2 <- c("exposed", "demands", "broken", "blind")
val_neg_3 <- c("harsh", "provoked", "ruined", "shattered", "rejected", "provoke", "reject"
               "crushed", "unjust", "forgotten", "impatient", "doubt", "outcry", "degraded
               "complacent", "deny", "seduced")
val_neg_4 <- c("sinful", "abandoned", "wrathful", "murder", "shame", "complained", "compla
               "crush", "pollute", "pollutes", "oppressed", "violated", "violating", "cond
               "disgrace", "disaster", "contempt")
```

```
P_sent_gam <- P_sent%>%
  left_join(ordered_gamma_2, by= c("heading" = "document"))

# renaming n so the value can be used in calculations without confusing R
colnames(P_sent_gam)[4] <- 'times_appear'

P_sent_gam <- P_sent_gam %>%
  mutate(value_weight = doc_weight*value*gamma_strength)%>%
  mutate(value_count = value * times_appear)
P_sent_gam <- P_sent_gam %>%
  mutate(value2=as.numeric(case_when(word %in% val_4 ~ 4,
                                    word %in% val_3 ~ 3,
                                    word %in% val_2 ~ 2,
                                    word %in% val_1 ~ 1,
                                    word %in% val_0 ~ 0,
                                    word %in% val_neg_2 ~ -2,
                                    word %in% val_neg_3 ~ -3,
                                    word %in% val_neg_4 ~ -4)))
P_sent_gam <- P_sent_gam %>%
  mutate(value=if_else(is.na(value2), value,value2))



P_sent_gam_pl <- P_sent_gam %>%
  group_by(bible_book_no, gamma_name, heading) %>%
  summarize(sent_vals = sum(value_weight), sort=TRUE)


P_sent_gam_plot <-
ggplot(data=P_sent_gam_pl, aes(x=sent_vals, group=gamma_name, fill=gamma_name)) +
    geom_density(adjust=1.5) +
    theme_bw() +
    facet_wrap(~gamma_name) +
    labs(title="Sentiments by Topic", x="Sentiment Level", y="Sentiment Prevalence", legen
    theme(legend.position="none",
      panel.spacing = unit(0.1, "lines"),
      axis.ticks.x=element_blank())

P_ridges <- ggplot(P_sent_gam_pl, aes(x = sent_vals, y = gamma_name, fill = gamma_name)) +
  geom_density_ridges() +
  theme_ridges() +
  labs(title="Sentiments by Topic", x="Sentiment Level", y=NULL, legend = NULL) +
  theme(legend.position = "none")

sent_prog <- P_sent_gam %>%
  group_by(gamma_name, doc_order) %>%
  summarize(tot_val_wt = sum(value_weight))


sent_prog_plot <- sent_prog %>%
  ggplot() +
  geom_path(aes(x=doc_order, y=tot_val_wt, color=gamma_name)) +
  labs(title="Sentiment Variation Through the Pentateuch", y="Weighted Sentiment Value", x
       caption = "Dotted lines mark the beginnings of the listed books") +
  theme_bw() +
  theme(axis.text.x = element_blank()) +
  theme(axis.ticks.x = element_blank()) +
  geom_vline(xintercept=c(1, 90, 190, 230, 306), linetype = 2, color="darkblue") +
  annotate("text", x=-15.7, y=-1400, label="Genesis", angle=90, size=2) +
  annotate("text", x=75, y=-1450, label="Exodus", angle=90, size=2) +
  annotate("text", x=175, y=-1400, label="Leviticus", angle=90, size=2) +
  annotate("text", x=215, y=-1400, label="Numbers", angle=90, size=2) +
  annotate("text", x=291, y=-1250, label="Deuteronomy", angle=90, size=2) +
  theme(plot.caption = element_text(size=8, face = "italic", hjust=0, margin=margin(t=20))
  facet_wrap(vars(gamma_name))

sent_prog_plot
```

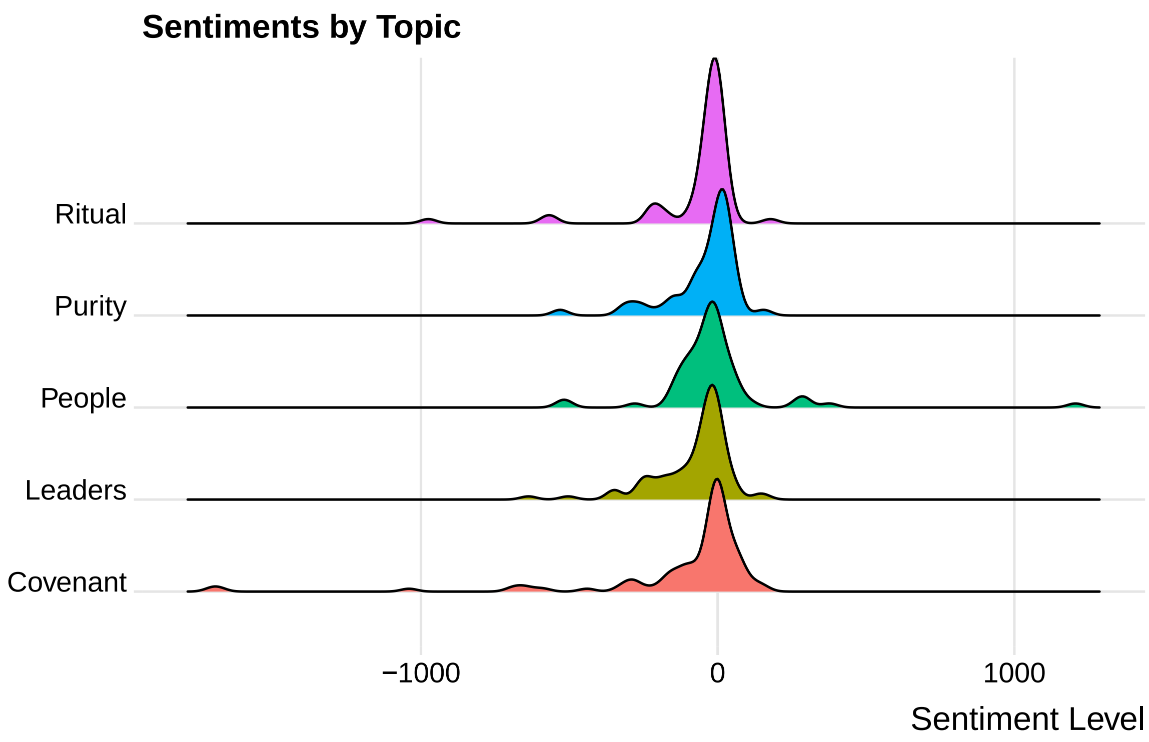


Dotted lines mark the beginnings of the listed books

```
P_sent_gam_plot
```


Sentiments by Topic
Covenant
Leaders
People
Purity
Ritual
Sentiment Prevalence
0.0100
0.0075
0.0050
0.0025
0.0000
−1000
0
1000
Sentiment Level

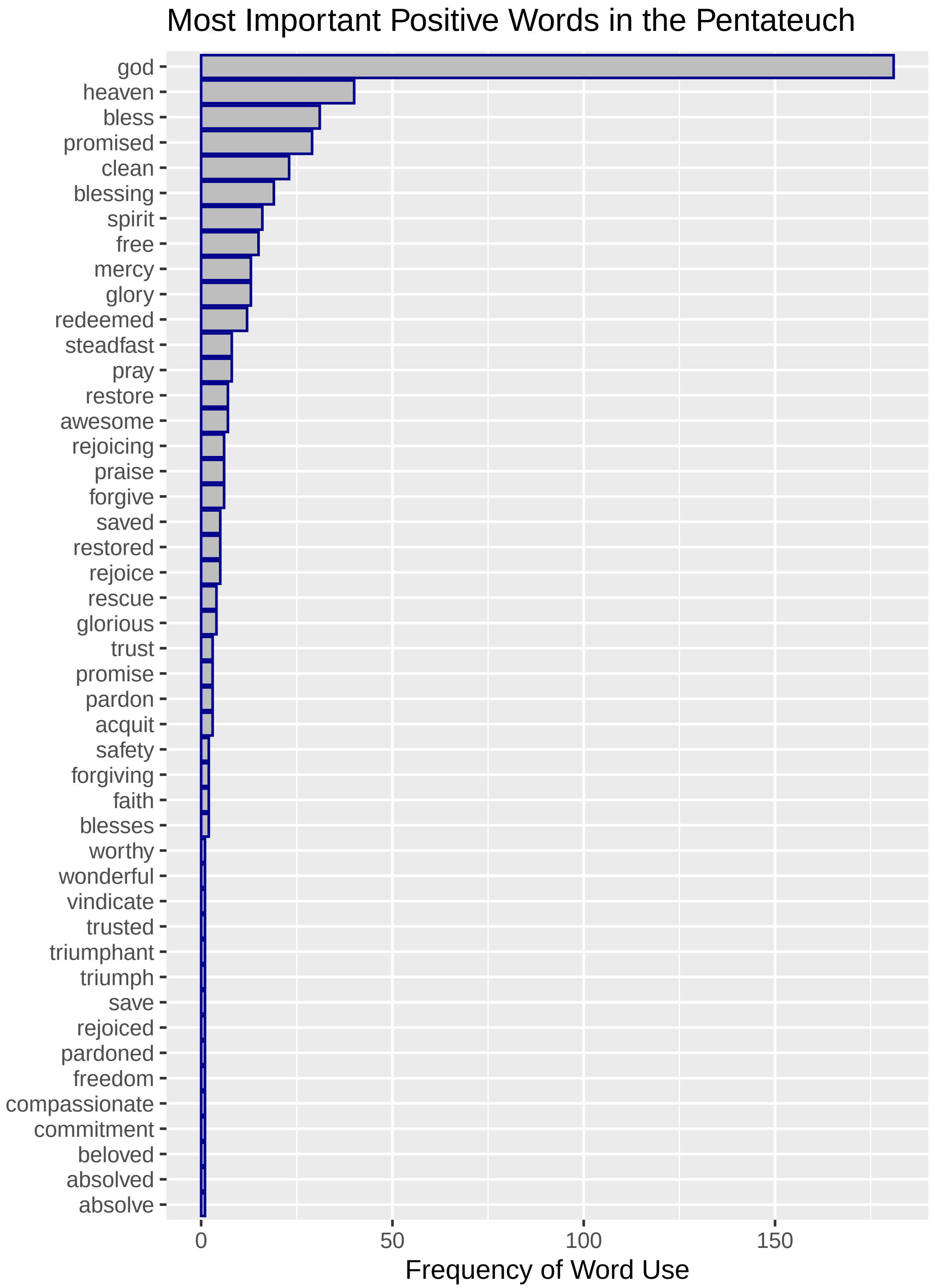
```
P_ridges
```



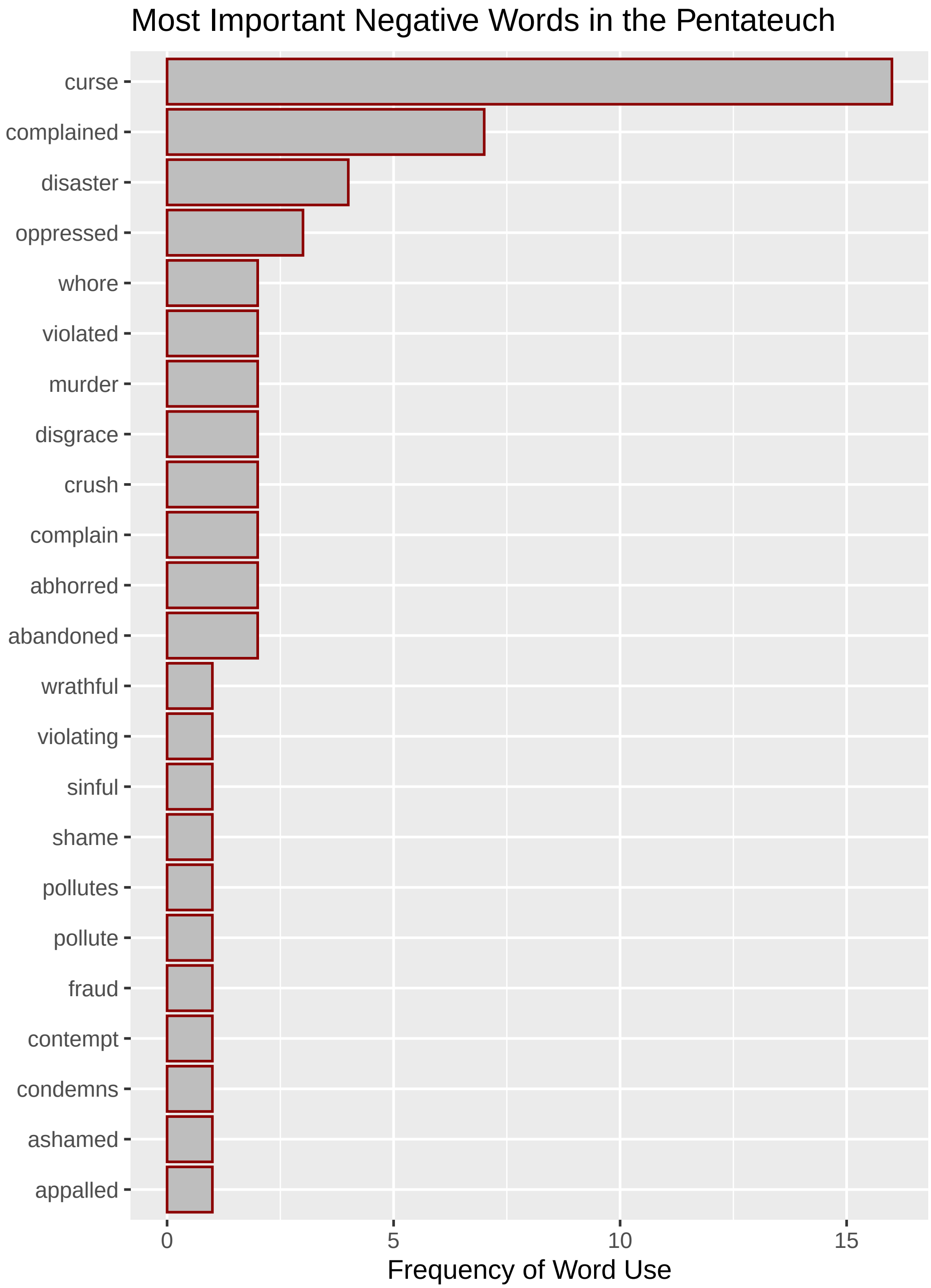
```
#ggsave("sent_prog.jpg", plot=sent_prog_plot)
#ggsave("p_sent_gam.jpg", plot=P_sent_gam_plot)
#ggsave("P_ridges.jpg", plot=P_ridges)
```

Having looked at the sentiments in general, I examined the most important positive and negative words in the corpus. Words like blessing, mercy, redeemed, and steadfast highlight the importance of relationships. Conversely, the negative terms cursed, complained, oppressed, disgrace highlight how the breaking of relationships matters.

```
pos_gam <-P_sent_gam %>%
  filter(value==4) %>%
  count(word, sort=T) %>%
  top_n(n=40, wt = n) %>%
  ggplot() +
  geom_col(aes(x=n, y=reorder(word, n)), color="darkblue", fill="grey") +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  labs(title="Most Important Positive Words in the Pentateuch", x="Frequency of Word Use",
pos_gam
```


Most Important Positive Words in the Pentateuch
god
heaven
bless
promised
clean
blessing
spirit
free
mercy
glory
redeemed
steadfast
pray
restore
awesome
rejoicing
praise
forgive
saved
restored
rejoice
rescue
glorious
trust
promise
pardon
acquit
safety
forgiving
faith
blesses
worthy
wonderful
vindicate
trusted
triumphant
triumph
save
rejoiced
pardoned
freedom
compassionate
commitment
beloved
absolved
absolve
0
50
100
150
Frequency of Word Use

```
neg_gam<-P_sent_gam %>%
  filter(value==-4) %>%
  count(word, sort=T) %>%
  top_n(n=40, wt = n) %>%
  ggplot() +
  geom_col(aes(x=n, y=reorder(word, n)), color="darkred", fill="grey") +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  labs(title="Most Important Negative Words in the Pentateuch", x="Frequency of Word Use",
neg_gam
```


Most Important Negative Words in the Pentateuch
curse
complained
disaster
oppressed
whore
violated
murder
disgrace
crush
complain
abhorred
abandoned
wrathful
violating
sinful
shame
pollutes
pollute
fraud
contempt
condemns
ashamed
appalled
0
5
10
15
Frequency of Word Use

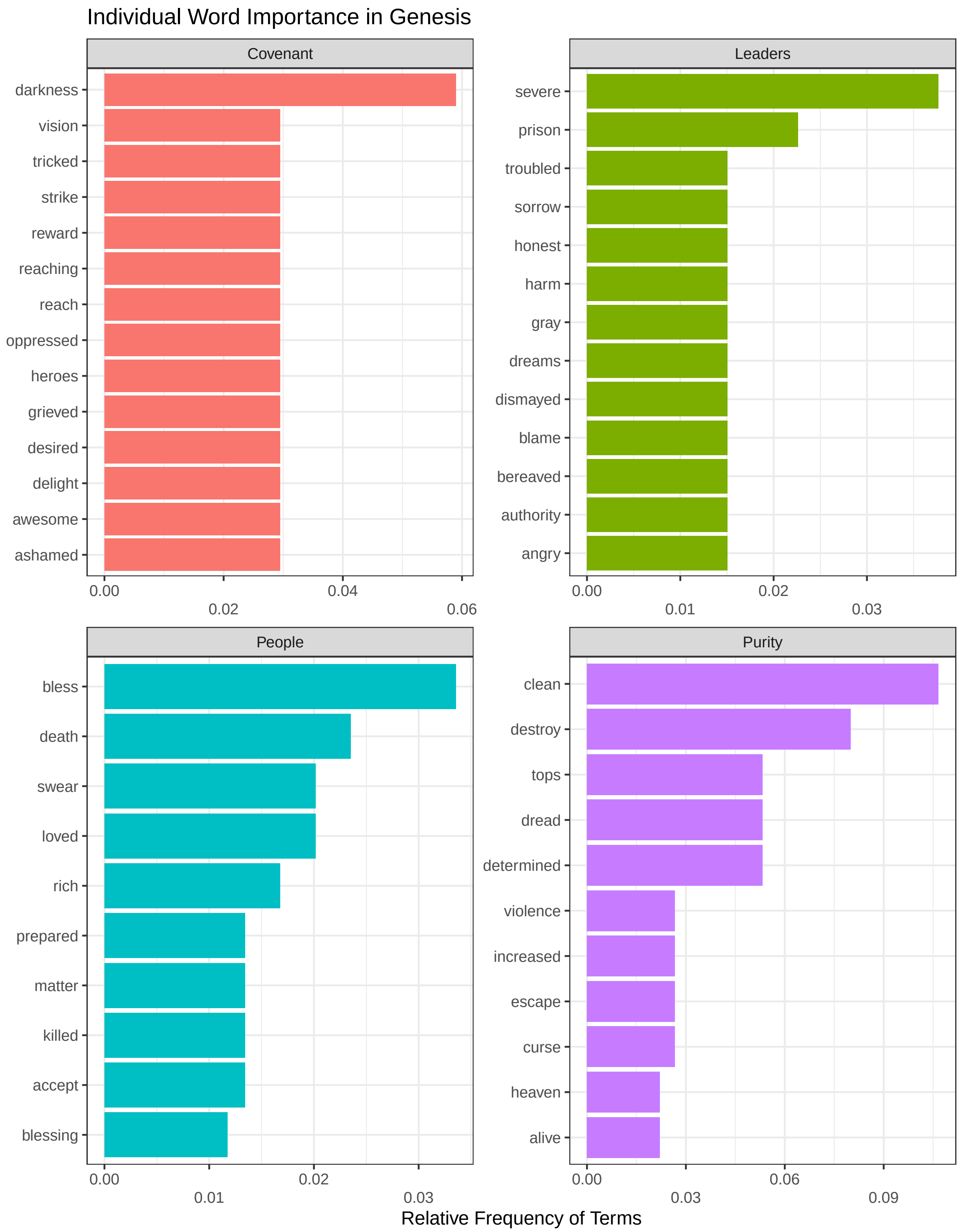
```
#ggsave("pos_gam.jpg", plot=pos_gam)
#ggsave("neg_gam.jpg", plot=neg_gam)
```

After looking at the important words for the whole corpus, I did a tf-idf analysis to look at the words at a topic level for each book of the Bible. Interestingly, but perhaps not surprisingly, most of the words for every topic and every book are negative. Tf-idf privileges words that are more specifically related to an individual document. I imagine that the ‘positive’ words are likely shared broadly across all of the documents and therefore do not appear as often.

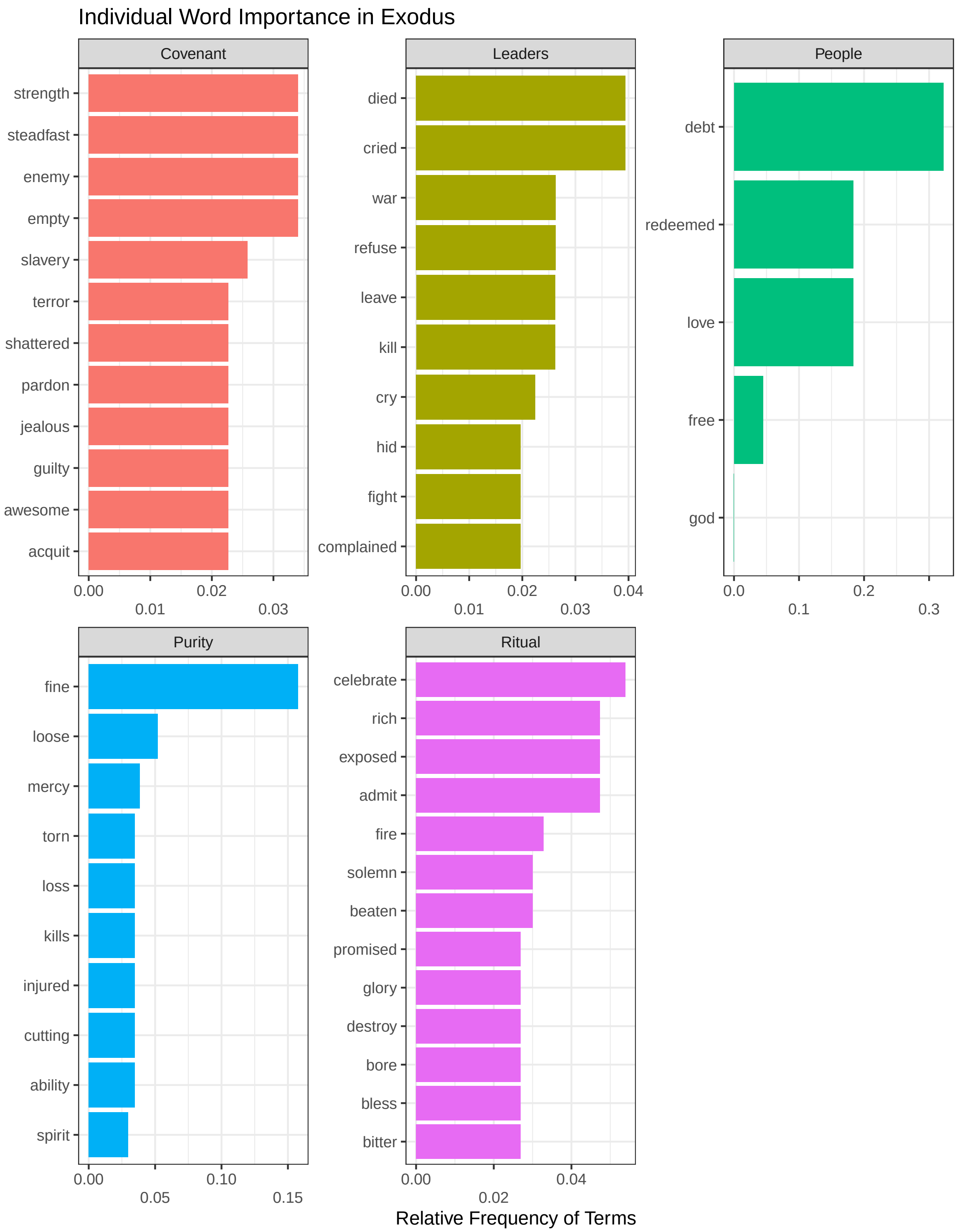
```
# gamma_name = topic = tf-idf document
# word = word
# bible_book_no

# Modifying P_sent_gam to aggregate words per book of bible
# word-value only for later merge
w_v_only <- P_sent_gam %>%
  select(word, value) %>%
  unique()

# Modifying P_sent_gam to aggregate words per book of bible for tf-idf
P_sel <- P_sent_gam %>%
  select(bible_book_no, gamma_name, word, times_appear) %>%
  group_by(bible_book_no, gamma_name, word) %>%
  summarize(times_appear=sum(times_appear))
```

```
# tf-idf
# one for each book of the Bible
# Genesis
# Exodus
# Leviticus
# Numbers
# Deuteronomy


# Genesis tf-idf
tf_Gen <-P_sent_gam %>%
  filter(bible_book_no==1) %>%
  count(gamma_name, word) %>%
  bind_tf_idf(term = word,
              document=gamma_name,
              n=n) %>%
  group_by(gamma_name) %>%
  top_n(10, wt=tf_idf)

tf_Gen_plot <- tf_Gen %>%
  ggplot() +
  geom_col(aes(x=tf_idf,
               y=reorder(word, tf_idf),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance in Genesis") +
  xlab("Relative Frequency of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

# Exodus tf-idf
tf_Ex <-P_sent_gam %>%
  filter(bible_book_no==2) %>%
  count(gamma_name, word) %>%
  bind_tf_idf(term = word,
              document=gamma_name,
              n=n) %>%
  group_by(gamma_name) %>%
  top_n(10, wt=tf_idf)

tf_Ex_plot <- tf_Ex %>%
  ggplot() +
  geom_col(aes(x=tf_idf,
               y=reorder(word, tf_idf),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance in Exodus") +
  xlab("Relative Frequency of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()
```

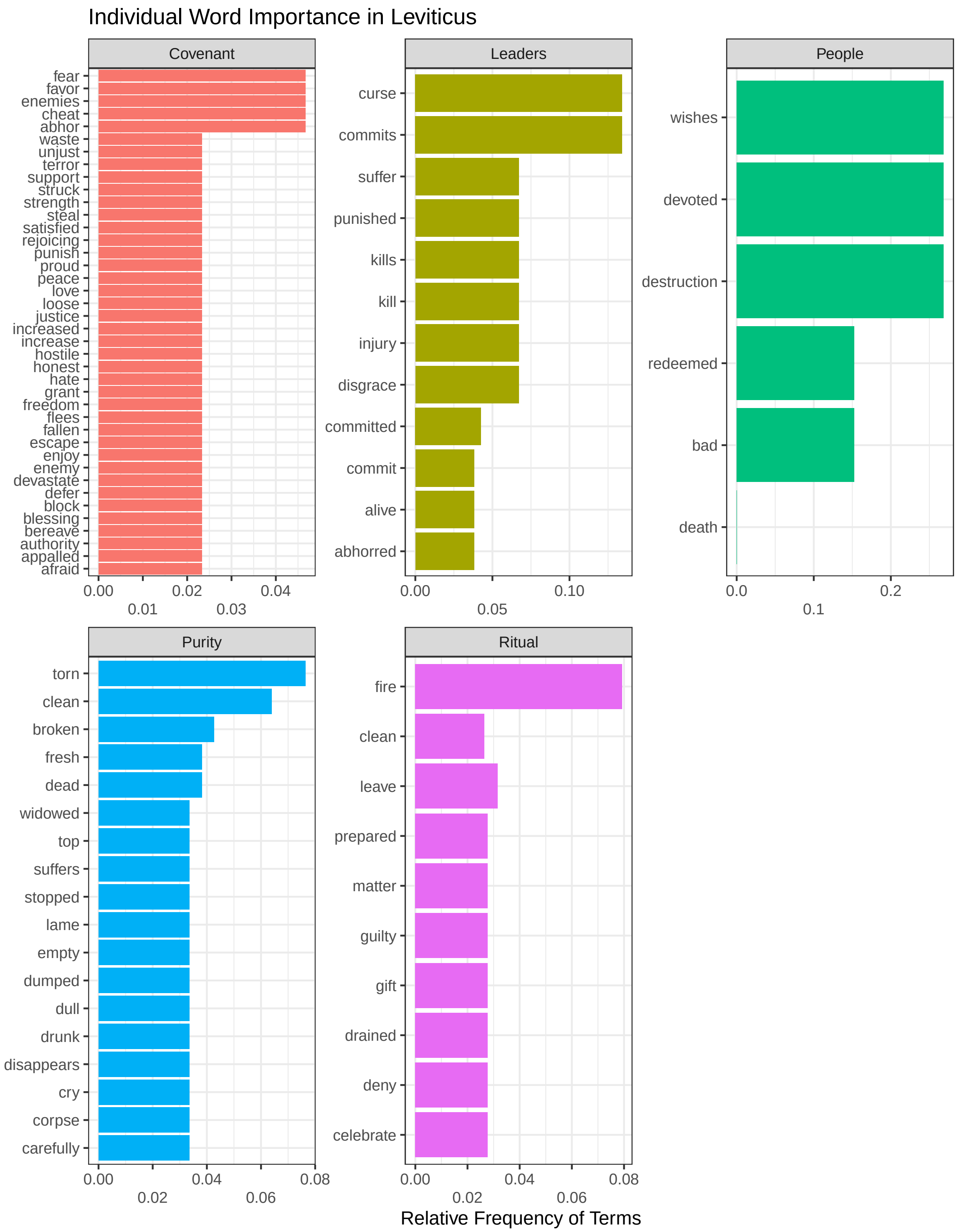
```r
# Leviticus tf-idf
tf_Lev <-P_sent_gam %>%
  filter(bible_book_no==3) %>%
  count(gamma_name, word) %>%
  bind_tf_idf(term = word,
              document=gamma_name,
              n=n) %>%
  group_by(gamma_name) %>%
  top_n(10, wt=tf_idf)

tf_Lev_plot <- tf_Lev %>%
  ggplot() +
  geom_col(aes(x=tf_idf,
               y=reorder(word, tf_idf),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance in Leviticus") +
  xlab("Relative Frequency of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

# Numbers tf-idf
tf_Num <-P_sent_gam %>%
  filter(bible_book_no==4) %>%
  count(gamma_name, word) %>%
  bind_tf_idf(term = word,
              document=gamma_name,
              n=n) %>%
  group_by(gamma_name) %>%
  top_n(10, wt=tf_idf)

tf_Num_plot <- tf_Num %>%
  ggplot() +
  geom_col(aes(x=tf_idf,
               y=reorder(word, tf_idf),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance in Numbers") +
  xlab("Relative Frequency of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

# Deuteronomy tf-idf
tf_Deu <-P_sent_gam %>%
  filter(bible_book_no==5) %>%
  count(gamma_name, word) %>%
  bind_tf_idf(term = word,
              document=gamma_name,
              n=n) %>%
  group_by(gamma_name) %>%
  top_n(10, wt=tf_idf)

tf_Deu_plot <- tf_Deu %>%
  ggplot() +
  geom_col(aes(x=tf_idf,
               reorder(word, tf_idf),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance in Deuteronomy") +
  xlab("Relative Frequency of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()



tf_Gen_plot
```


Individual Word Importance in Genesis
Covenant
darkness
vision
tricked
strike
reward
reaching
reach
oppressed
heroes
grieved
desired
delight
awesome
ashamed
0.00
0.02
0.04
0.06
Leaders
severe
prison
troubled
sorrow
honest
harm
gray
dreams
dismayed
blame
bereaved
authority
angry
0.00
0.01
0.02
0.03
People
bless
death
swear
loved
rich
prepared
matter
killed
accept
blessing
0.00
0.01
0.02
0.03
Purity
clean
destroy
tops
dread
determined
violence
increased
escape
curse
heaven
alive
0.00
0.03
0.06
0.09
Relative Frequency of Terms

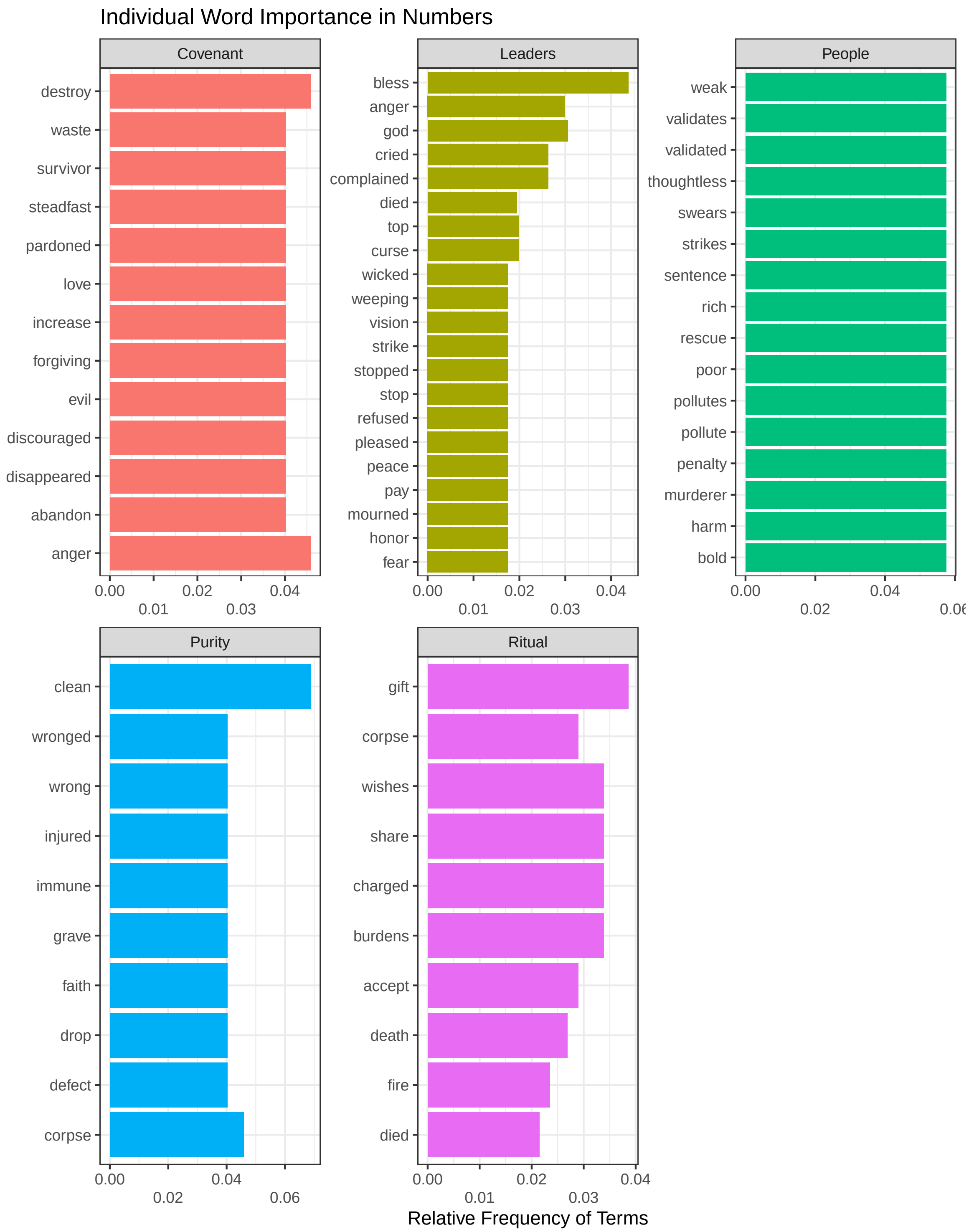
```
tf_Ex_plot
```


Individual Word Importance in Exodus
Covenant
strength
steadfast
enemy
empty
slavery
terror
shattered
pardon
jealous
guilty
awesome
acquit
0.00
0.01
0.02
0.03
Leaders
died
cried
war
refuse
leave
kill
cry
hid
fight
complained
0.00
0.01
0.02
0.03
0.04
People
debt
redeemed
love
free
god
0.0
0.1
0.2
0.3
Purity
fine
loose
mercy
torn
loss
kills
injured
cutting
ability
spirit
0.00
0.05
0.10
0.15
Ritual
celebrate
rich
exposed
admit
fire
solemn
beaten
promised
glory
destroy
bore
bless
bitter
0.00
0.02
0.04
Relative Frequency of Terms

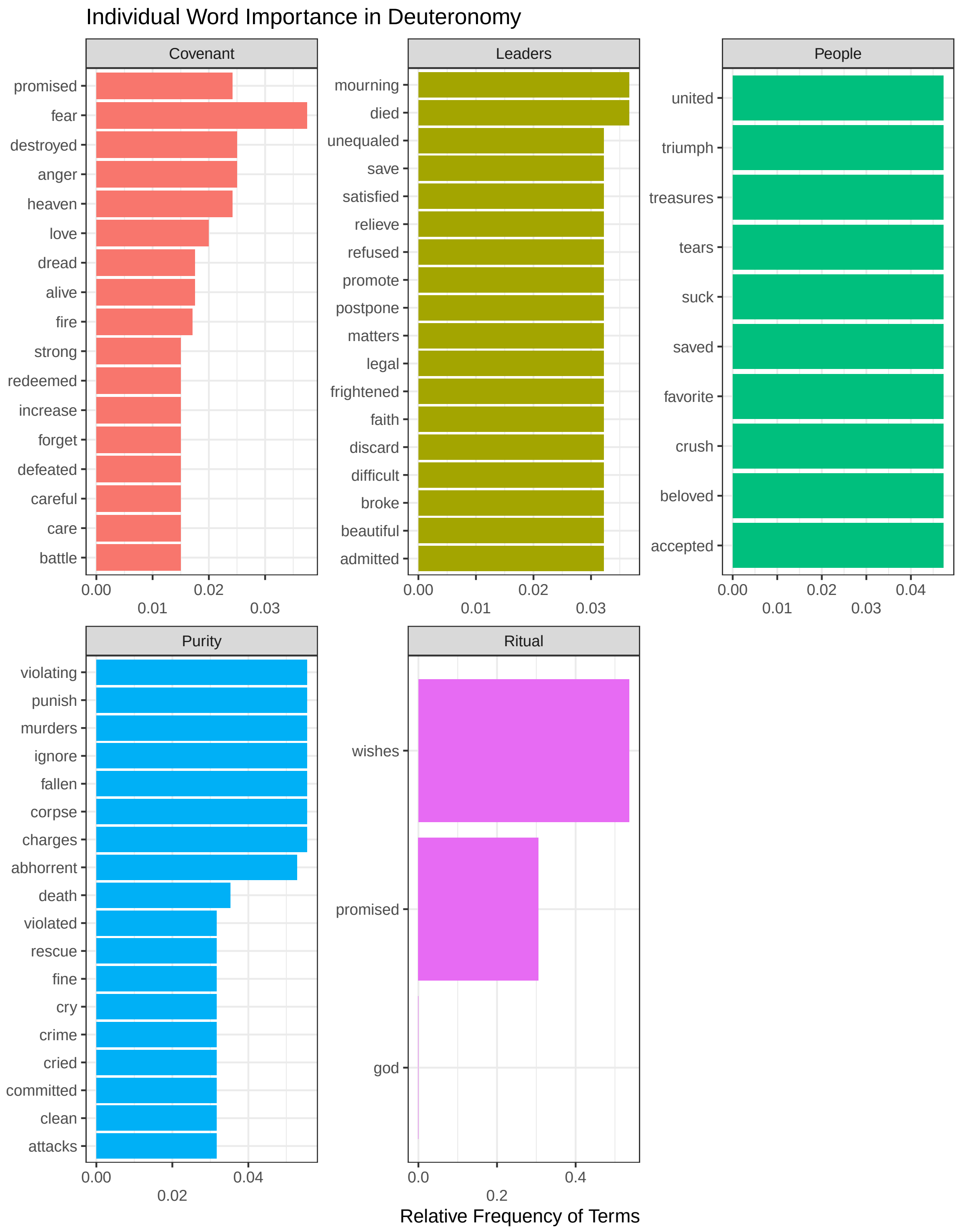
```
tf_Lev_plot
```


Individual Word Importance in Leviticus
Covenant
fear
favor
enemies
cheat
abhor
waste
unjust
terror
support
struck
strength
steal
satisfied
rejoicing
punish
proud
peace
love
loose
justice
increased
increase
hostile
honest
hate
grant
freedom
flees
fallen
escape
enjoy
enemy
devastate
defer
block
blessing
bereave
authority
appalled
afraid
0.00
0.01
0.02
0.03
0.04
Leaders
curse
commits
suffer
punished
kills
kill
injury
disgrace
committed
commit
alive
abhorred
0.00
0.05
0.10
People
wishes
devoted
destruction
redeemed
bad
death
0.0
0.1
0.2
Purity
torn
clean
broken
fresh
dead
widowed
top
suffers
stopped
lame
empty
dumped
dull
drunk
disappears
cry
corpse
carefully
0.00
0.02
0.04
0.06
0.08
Ritual
fire
clean
leave
prepared
matter
guilty
gift
drained
deny
celebrate
0.00
0.02
0.04
0.06
0.08
Relative Frequency of Terms

```
tf_Num_plot
```


Individual Word Importance in Numbers
Covenant
destroy
waste
survivor
steadfast
pardoned
love
increase
forgiving
evil
discouraged
disappeared
abandon
anger
0.00
0.01
0.02
0.03
0.04
Leaders
bless
anger
god
cried
complained
died
top
curse
wicked
weeping
vision
strike
stopped
stop
refused
pleased
peace
pay
mourned
honor
fear
0.00
0.01
0.02
0.03
0.04
People
weak
validates
validated
thoughtless
swears
strikes
sentence
rich
rescue
poor
pollutes
pollute
penalty
murderer
harm
bold
0.00
0.02
0.04
0.06
Purity
clean
wronged
wrong
injured
immune
grave
faith
drop
defect
corpse
0.00
0.02
0.04
0.06
Ritual
gift
corpse
wishes
share
charged
burdens
accept
death
fire
died
0.00
0.01
0.02
0.03
0.04
Relative Frequency of Terms

```
tf_Deu_plot
```


Individual Word Importance in Deuteronomy
Covenant
promised
fear
destroyed
anger
heaven
love
dread
alive
fire
strong
redeemed
increase
forget
defeated
careful
care
battle
0.00
0.01
0.02
0.03
Leaders
mourning
died
unequaled
save
satisfied
relieve
refused
promote
postpone
matters
legal
frightened
faith
discard
difficult
broke
beautiful
admitted
0.00
0.01
0.02
0.03
People
united
triumph
treasures
tears
suck
saved
favorite
crush
beloved
accepted
0.00
0.01
0.02
0.03
0.04
Purity
violating
punish
murders
ignore
fallen
corpse
charges
abhorrent
death
violated
rescue
fine
cry
crime
cried
committed
clean
attacks
0.00
0.02
0.04
Ritual
wishes
promised
god
0.0
0.2
0.4
Relative Frequency of Terms

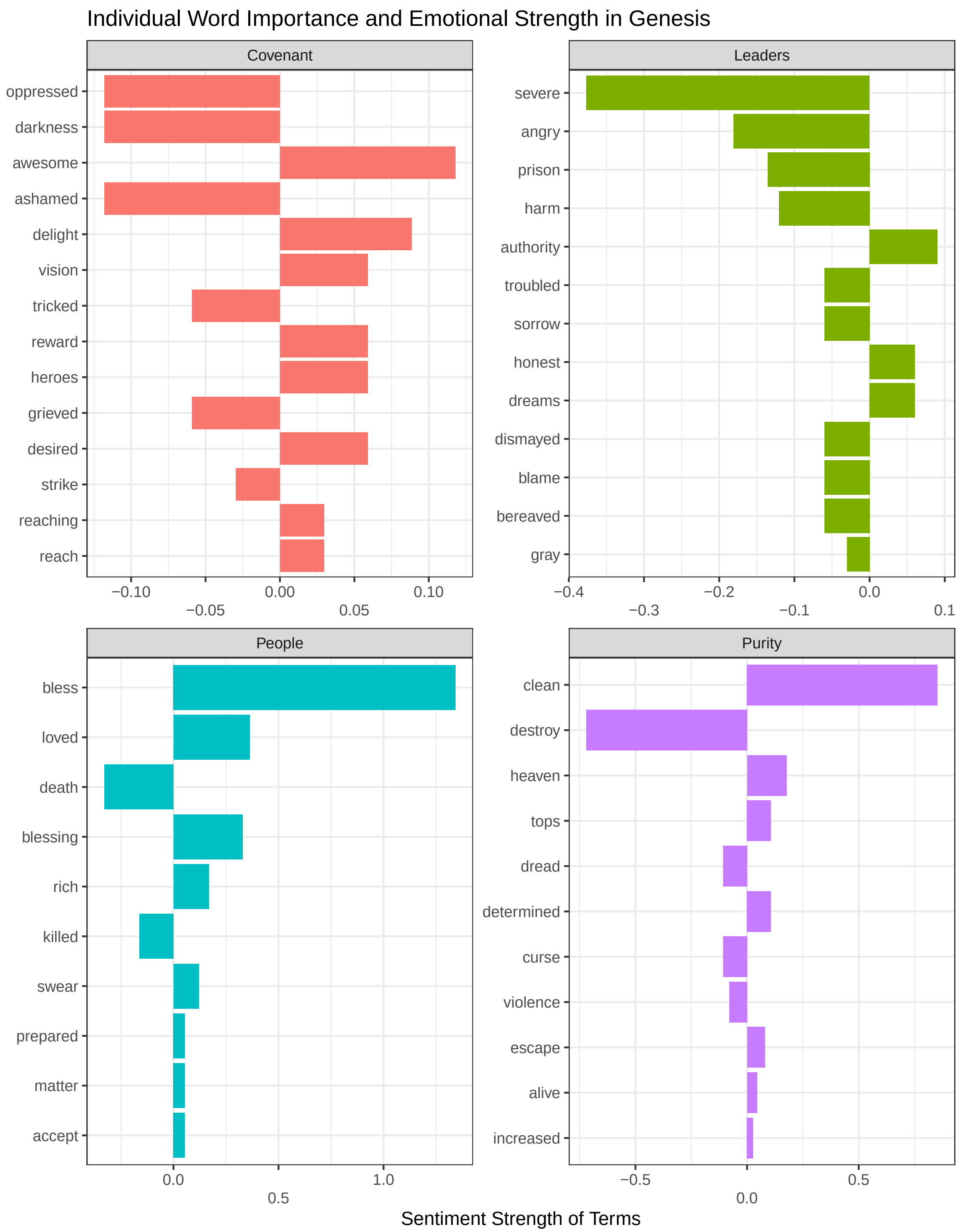
```
#ggsave("tf_Gen.jpg", plot=tf_Gen_plot)
#ggsave("tf_Ex.jpg", plot=tf_Ex_plot)
#ggsave("tf_Lev.jpg", plot=tf_Lev_plot)
#ggsave("tf_Num.jpg", plot=tf_Num_plot)
#ggsave("tf_Deu.jpg", plot=tf_Deu_plot)
```

Because I am very interested in the sentiments, I decided to do a sentiment analysis with tf-idf. I created a val_tf_strength variable which I calculated by taking the tf_idf value and multiplying it by the afinn value and the number of times that the word appears in the book of the Bible. One general observation is that, when leaders have more negative terms as the most important, people has more positive ones, and vice versa. There was a lot of conflict!

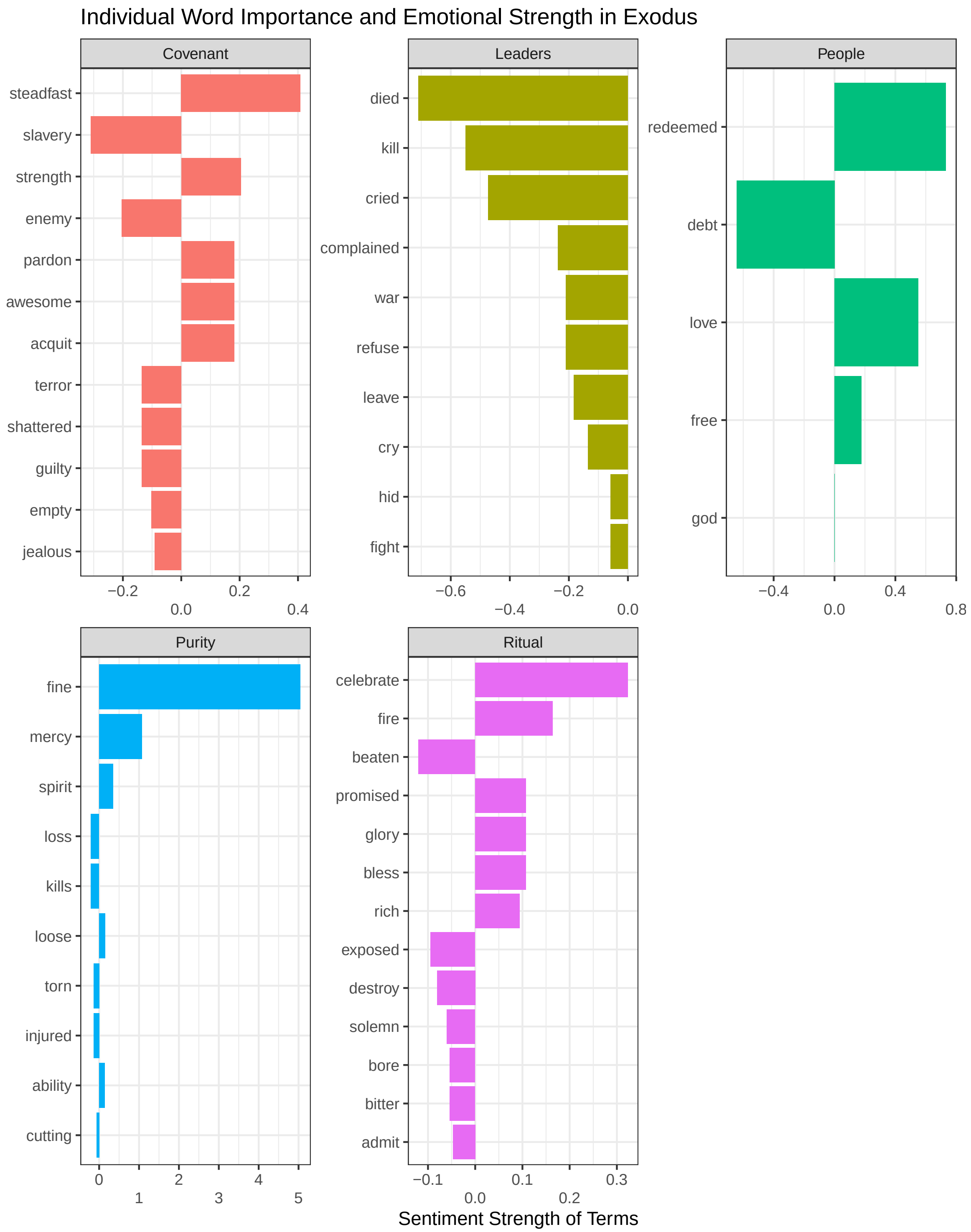
```
# combining tf-idf with sentiment levels


# renaming n so the value can be used in calculations without confusing R
colnames(tf_Gen)[3] <- 'times_appear'
colnames(tf_Ex)[3] <- 'times_appear'
colnames(tf_Lev)[3] <- 'times_appear'
colnames(tf_Num)[3] <- 'times_appear'
colnames(tf_Deu)[3] <- 'times_appear'

# merging the tf files with sentiment values
tf_sent_Gen <-
  left_join(tf_Gen, w_v_only, by="word") %>%
  mutate(val_tf_strength = value * times_appear*tf_idf)

tf_sent_Ex<-
  left_join(tf_Ex, w_v_only, by="word") %>%
  mutate(val_tf_strength = value * times_appear*tf_idf)

tf_sent_Lev <-
  left_join(tf_Lev, w_v_only, by="word") %>%
  mutate(val_tf_strength = value * times_appear*tf_idf)

tf_sent_Num <-
  left_join(tf_Num, w_v_only, by="word") %>%
  mutate(val_tf_strength = value * times_appear*tf_idf)

tf_sent_Deu <-
  left_join(tf_Deu, w_v_only, by="word") %>%
  mutate(val_tf_strength = value * times_appear*tf_idf)
```

```
# plotting combination of tf-idf importance and sentiment and frequency of sentiment

# one for each book of the Bible

# Genesis
# Exodus
# Leviticus
# Numbers
# Deuteronomy

tf_sent_Gen_plot <- tf_sent_Gen %>%
  ggplot() +
  geom_col(aes(x=val_tf_strength,
               y=reorder(word, abs(val_tf_strength)),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance and Emotional Strength in Genesis") +
  xlab("Sentiment Strength of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

tf_sent_Ex_plot <- tf_sent_Ex %>%
  ggplot() +
  geom_col(aes(x=val_tf_strength,
               y=reorder(word, abs(val_tf_strength)),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance and Emotional Strength in Exodus") +
  xlab("Sentiment Strength of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

tf_sent_Lev_plot <- tf_sent_Lev %>%
  ggplot() +
  geom_col(aes(x=val_tf_strength,
               y=reorder(word, abs(val_tf_strength)),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance and Emotional Strength in Leviticus") +
  xlab("Sentiment Strength of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

tf_sent_Num_plot <- tf_sent_Num %>%
  ggplot() +
  geom_col(aes(x=val_tf_strength,
               y=reorder(word, abs(val_tf_strength)),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance and Emotional Strength in Numbers") +
  xlab("Sentiment Strength of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()

tf_sent_Deu_plot <- tf_sent_Deu %>%
  ggplot() +
  geom_col(aes(x=val_tf_strength,
               y=reorder(word, abs(val_tf_strength)),
               fill=gamma_name),
           show.legend=F) +
  facet_wrap(~gamma_name, scales = "free") +
  ggtitle("Individual Word Importance and Emotional Strength in Deuteronomy") +
  xlab("Sentiment Strength of Terms") +
  ylab("") +
  guides(x = guide_axis(n.dodge = 2)) +
  theme(plot.margin=unit(c(1,1,1,1), 'cm')) +
  theme_bw()
```

```
tf_sent_Gen_plot
```

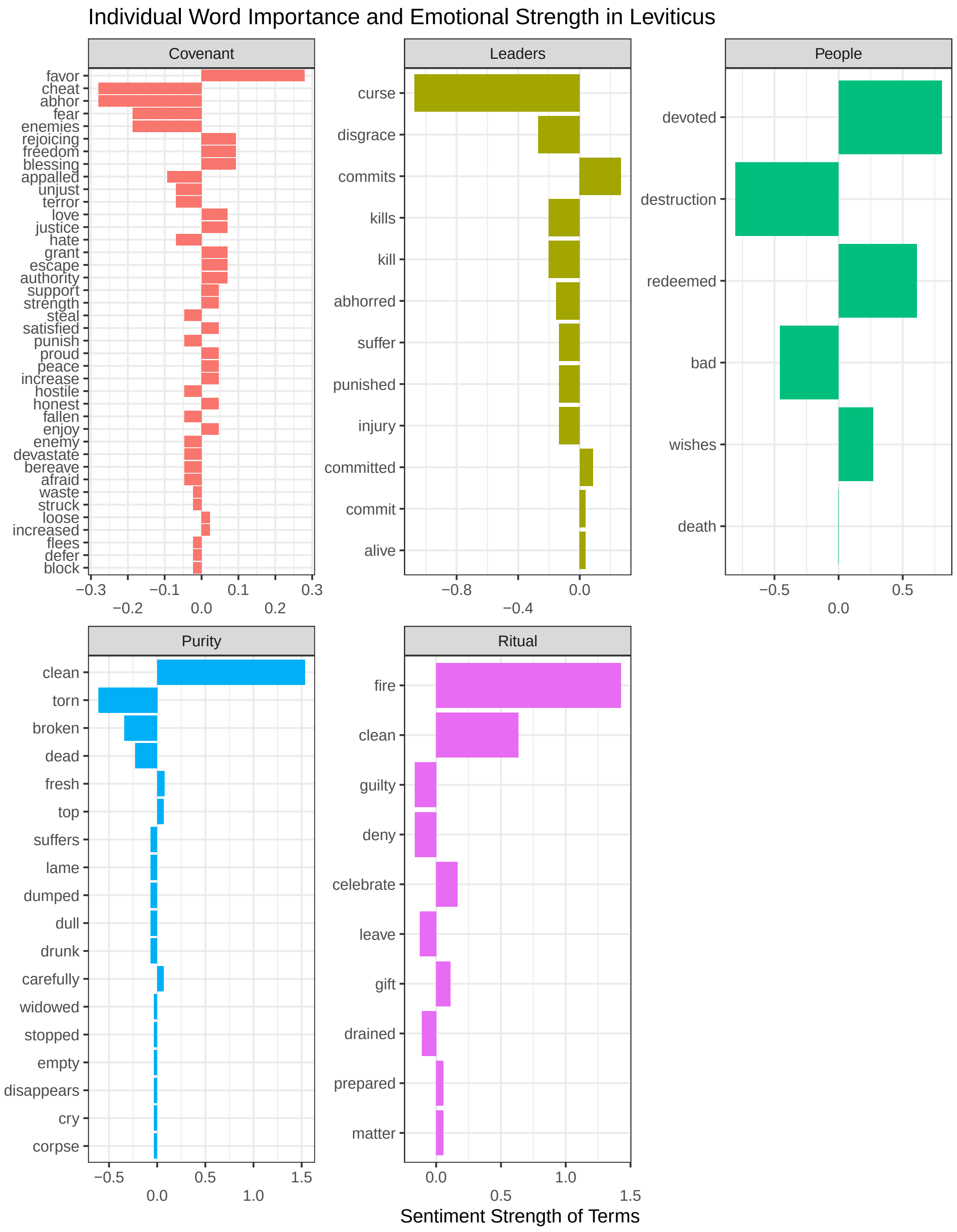

Individual Word Importance and Emotional Strength in Genesis
Covenant
oppressed
darkness
awesome
ashamed
delight
vision
tricked
reward
heroes
grieved
desired
strike
reaching
reach
−0.10
−0.05
0.00
0.05
0.10
Leaders
severe
angry
prison
harm
authority
troubled
sorrow
honest
dreams
dismayed
blame
bereaved
gray
−0.4
−0.3
−0.2
−0.1
0.0
0.1
People
bless
loved
death
blessing
rich
killed
swear
prepared
matter
accept
0.0
0.5
1.0
Purity
clean
destroy
heaven
tops
dread
determined
curse
violence
escape
alive
increased
−0.5
0.0
0.5
Sentiment Strength of Terms

```
tf_sent_Ex_plot
```

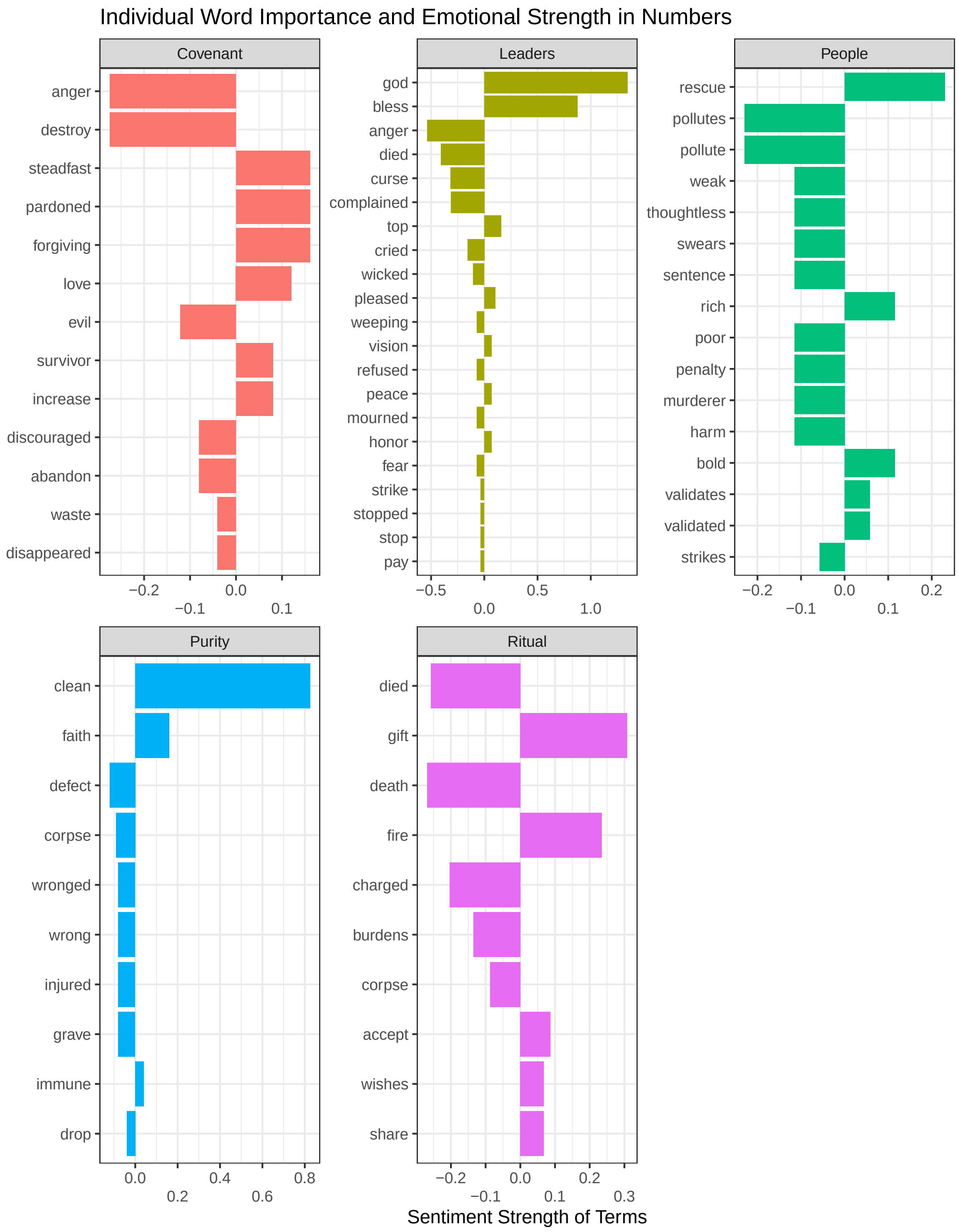


Individual Word Importance and Emotional Strength in Exodus

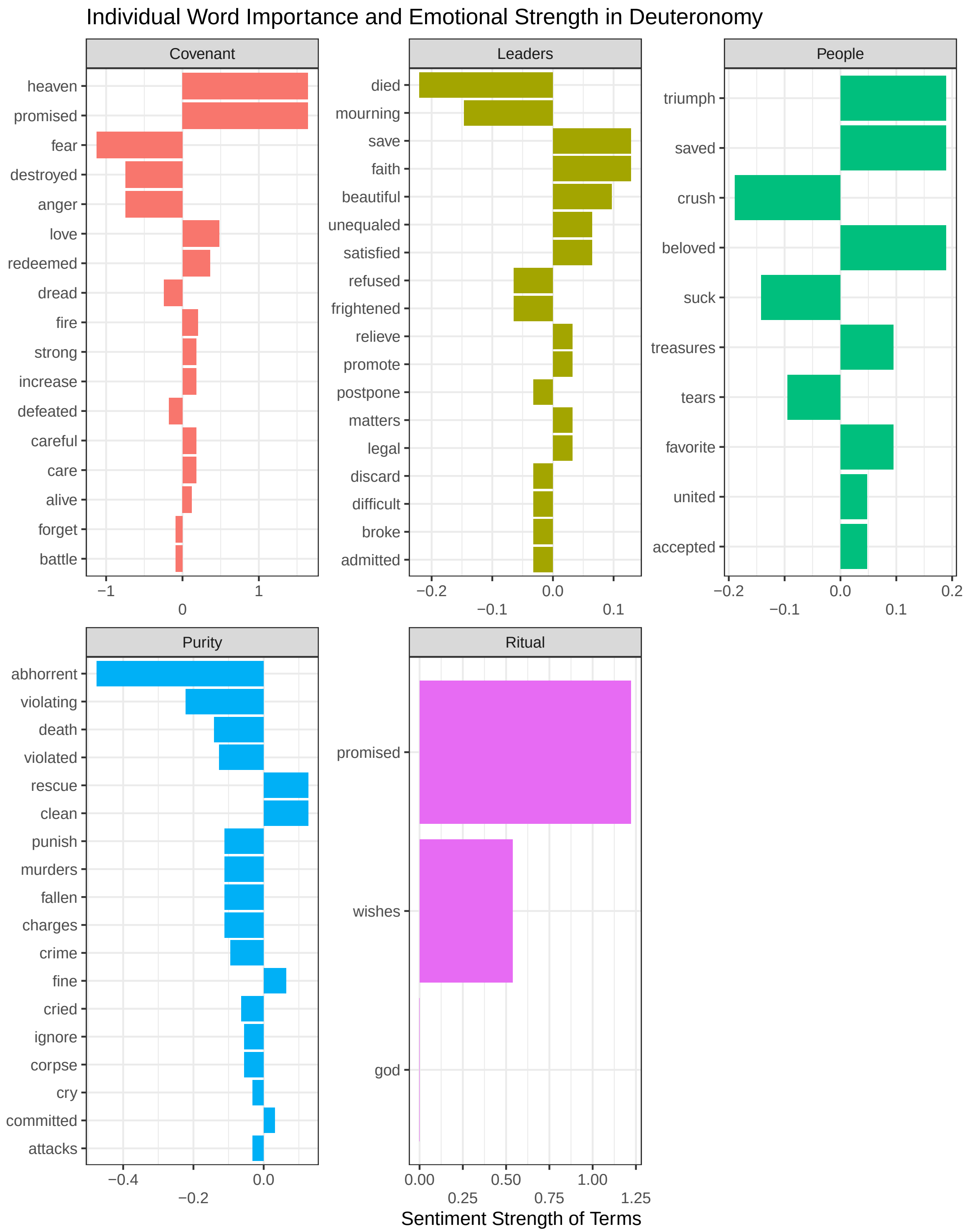
```
tf_sent_Lev_plot
```


Individual Word Importance and Emotional Strength in Leviticus
Covenant
favor
cheat
abhor
fear
enemies
rejoicing
freedom
blessing
appalled
unjust
terror
love
justice
hate
grant
escape
authority
support
strength
steal
satisfied
punish
proud
peace
increase
hostile
honest
fallen
enjoy
enemy
devastate
bereave
afraid
waste
struck
loose
increased
flees
defer
block
−0.3
−0.1
0.1
0.3
−0.2
0.0
0.2
Leaders
curse
disgrace
commits
kills
kill
abhorred
suffer
punished
injury
committed
commit
alive
−0.8
0.0
−0.4
People
devoted
destruction
redeemed
bad
wishes
death
−0.5
0.5
0.0
Purity
clean
torn
broken
dead
fresh
top
suffers
lame
dumped
dull
drunk
carefully
widowed
stopped
empty
disappears
cry
corpse
−0.5
0.5
1.5
0.0
1.0
Ritual
fire
clean
guilty
deny
celebrate
leave
gift
drained
prepared
matter
0.0
1.0
0.5
1.5
Sentiment Strength of Terms

```
tf_sent_Num_plot
```


Individual Word Importance and Emotional Strength in Numbers
Covenant
anger
destroy
steadfast
pardoned
forgiving
love
evil
survivor
increase
discouraged
abandon
waste
disappeared
−0.2
−0.1
0.0
0.1
Leaders
god
bless
anger
died
curse
complained
top
cried
wicked
pleased
weeping
vision
refused
peace
mourned
honor
fear
strike
stopped
stop
pay
−0.5
0.0
0.5
1.0
People
rescue
pollutes
pollute
weak
thoughtless
swears
sentence
rich
poor
penalty
murderer
harm
bold
validates
validated
strikes
−0.2
−0.1
0.0
0.1
0.2
Purity
clean
faith
defect
corpse
wronged
wrong
injured
grave
immune
drop
0.0
0.2
0.4
0.6
0.8
Ritual
died
gift
death
fire
charged
burdens
corpse
accept
wishes
share
−0.2
−0.1
0.0
0.1
0.2
0.3
Sentiment Strength of Terms

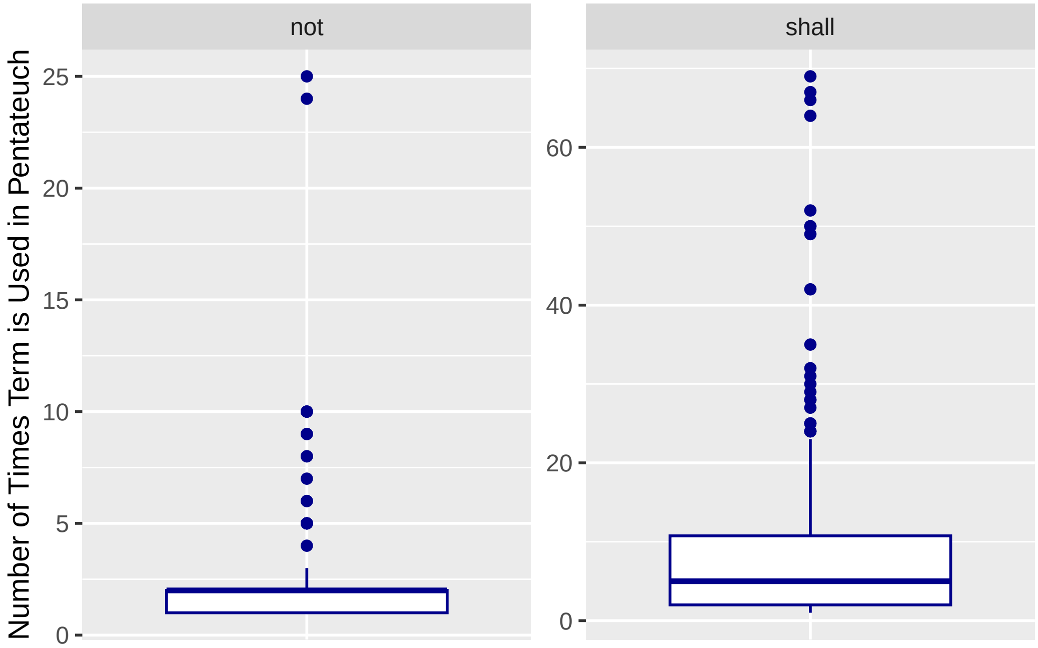
```
tf_sent_Deu_plot
```


Individual Word Importance and Emotional Strength in Deuteronomy
Covenant
heaven
promised
fear
destroyed
anger
love
redeemed
dread
fire
strong
increase
defeated
careful
care
alive
forget
battle
−1
0
1
Leaders
died
mourning
save
faith
beautiful
unequaled
satisfied
refused
frightened
relieve
promote
postpone
matters
legal
discard
difficult
broke
admitted
−0.2
−0.1
0.0
0.1
People
triumph
saved
crush
beloved
suck
treasures
tears
favorite
united
accepted
−0.2
−0.1
0.0
0.1
0.2
Purity
abhorrent
violating
death
violated
rescue
clean
punish
murders
fallen
charges
crime
fine
cried
ignore
corpse
cry
committed
attacks
−0.4
−0.2
0.0
Ritual
promised
wishes
god
0.00
0.25
0.50
0.75
1.00
1.25
Sentiment Strength of Terms

```
#ggsave("tf_sent_Gen.jpg", plot=tf_sent_Gen_plot)
#ggsave("tf_sent_Ex.jpg", plot=tf_sent_Ex_plot)
#ggsave("tf_sent_Lev.jpg", plot=tf_sent_Lev_plot)
#ggsave("tf_sent_Num.jpg", plot=tf_sent_Num_plot)
#ggsave("tf_sent_Deu.jpg", plot=tf_sent_Deu_plot)
```

Finally, I wanted to further explore the big drop in sentiment for the covenant topic in the book of Deuteronomy. A cursory look showed that there were a lot of "shall" and "shall not" commands. I wondered if that was the reason for the negative sentiment. The data suggests that that is the case.

```
# shall and shall not analysis

# finding the occurrences of "shall" and "shall not"
comm <- pent_mod
comm$ind <- c(1:146460)
#head(comm)

comm2_1 <- comm %>%
  mutate(shall_find = if_else(word == "shall" & lead(word, n=1) !="not", 1, 0)) %>%
  mutate(not_find = if_else(lag(word, n=1) == "shall" & word == "not", 1, 0))


comm2 <- comm2_1 %>%
  filter(not_find ==1 | shall_find ==1) %>%
  mutate(shall_not = if_else(shall_find == 1, "shall", "not"))

comm3 <- comm2 %>%
  group_by(heading, word) %>%
  summarize(n = n())
colnames(comm3)[3] <- "s_n_count"

shall_not_box<-comm3 %>%
  ggplot() +
  geom_boxplot(aes(x = factor(word), y = s_n_count), color="darkblue") +
  theme(axis.text.x = element_blank()) +
  theme(axis.ticks.x = element_blank()) +
  labs(x=NULL, y="Number of Times Term is Used in Pentateuch") +
  facet_wrap(~ word, scales="free")

shall_not_box
```

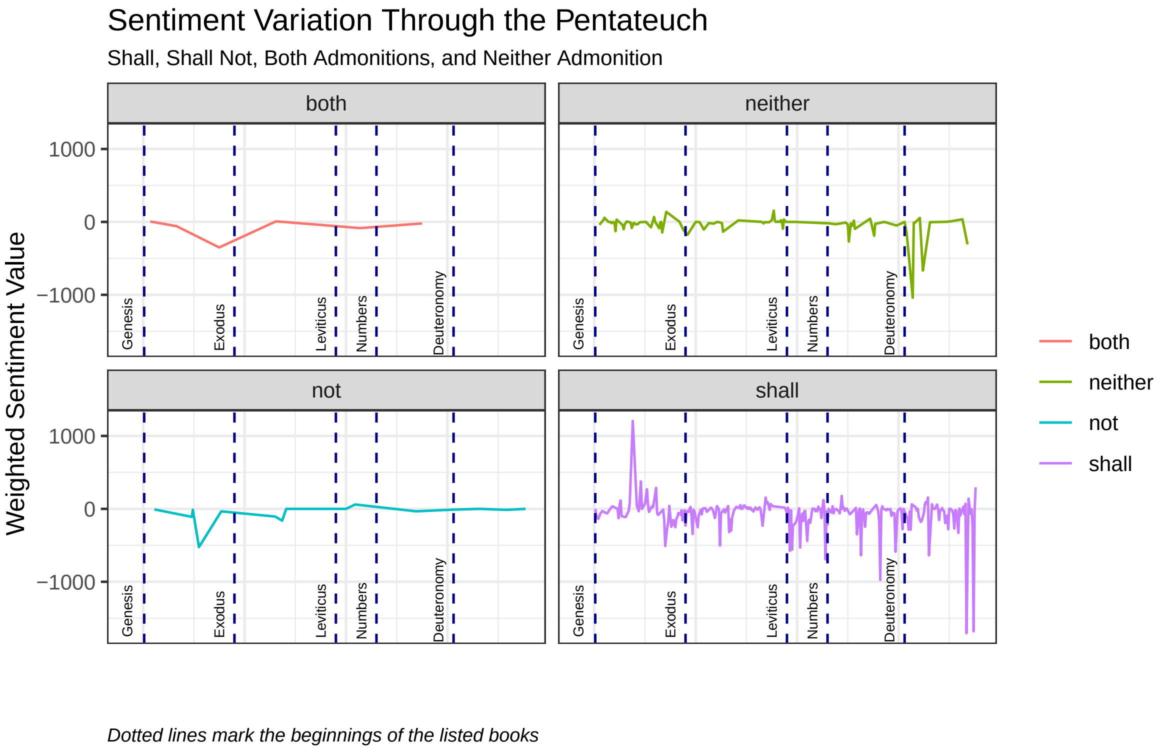


```
#ggsave("shall_box.jpg", plot=shall_not_box)
```

```
# reducing P_sent_gam file to merge with shall/shall not
#ad_gam <- P_sent_gam %>%
#  select(heading, head_id, bible_book_no, gamma_name, doc_order)
#ad_gam2 <- unique(ad_gam)

# reducing ordered_gamma_3 file to merge with shall/shall not
# ordered_gamma_3 is the master file from before sentiment analysis and
# therefore contains sections that are not included in sentiment analysis
ad_gam <- ordered_gamma_3 %>%
  rename(heading=document) %>%
  select(heading, head_id, bible_book_no, gamma_name, doc_order)
ad_gam2 <- unique(ad_gam)



# merging the comm3 file with P_sent_gam to get everything in same order

admonition <-
  left_join(ad_gam2, comm3, by = "heading") %>%
  mutate(word=if_else(is.na(word), "neither", word),
         s_n_count=if_else(is.na(s_n_count), 0, s_n_count))

admonition2 <- admonition %>%
  mutate(s_n_class = case_when(head_id==1 ~ word,
                               word=="neither" ~ "neither",
                               lag(heading, n=1)==heading & lag(s_n_count, n=1) == s_n_cou
                               lag(heading, n=1)==heading & lag(s_n_count, n=1) < s_n_coun
                               lead(heading, n=1) == heading & lead(s_n_count, n=1) < s_n_
                               lag(heading, n=1)!=heading & lead(heading, n=1) !=heading ~
                               head_id==5841 ~ word,
                               TRUE ~ "discard")) %>%
  mutate(s_n_net = case_when(head_id==1 ~ s_n_count,
                               word=="neither" ~ 0,
                               lag(heading, n=1)==heading & lag(s_n_count, n=1) == s_n_cou
                               lag(heading, n=1)==heading & lag(s_n_count, n=1) < s_n_coun
                               lead(heading, n=1) == heading & lead(s_n_count, n=1) < s_n_
                               lag(heading, n=1)!=heading & lead(heading, n=1) !=heading ~
                               head_id==5841 ~ s_n_count,
                               TRUE ~ 999))

admonition3 <- admonition2 %>%
  filter(s_n_class != "discard" & s_n_net!=999)
```

```
# P_sent_gam_pl has the sentiment values aggregated by heading
# P_sent_trunc is truncated version of P_sent_gam_pl
# joining the shall / shall not admonition3 file and P_sent_trunc to look at
# sentiments specifically in relation to shall / shall not

P_sent_trunc <- P_sent_gam_pl %>%
  select(., -sort)


admonition_sent <-
  left_join(admonition3, P_sent_trunc, by="heading")

admonition_sent <- admonition_sent %>%
  mutate(sent_vals = if_else(is.na(sent_vals), 0.0000000000000000000000, sent_vals))
```

```r
sent_ad_plot <-
  ggplot() +
  geom_path(data=admonition_sent, aes(x=doc_order, y=sent_vals, color=s_n_class)) +
  labs(title="Sentiment Variation Through the Pentateuch",
       subtitle = "Shall, Shall Not, Both Admonitions, and Neither Admonition",
       y="Weighted Sentiment Value", x="", color = "",
       caption = "Dotted lines mark the beginnings of the listed books") +
  theme_bw() +
  theme(axis.text.x = element_blank()) +
  theme(axis.ticks.x = element_blank()) +
  geom_vline(xintercept=c(1, 90, 190, 230, 306), linetype = 2, color="darkblue") +
  annotate("text", x=-15.7, y=-1400, label="Genesis", angle=90, size=2) +
  annotate("text", x=75, y=-1450, label="Exodus", angle=90, size=2) +
  annotate("text", x=175, y=-1400, label="Leviticus", angle=90, size=2) +
  annotate("text", x=215, y=-1400, label="Numbers", angle=90, size=2) +
  annotate("text", x=291, y=-1250, label="Deuteronomy", angle=90, size=2) +
  theme(plot.caption = element_text(size=8, face = "italic", hjust=0, margin=margin(t=20))
  theme(plot.subtitle = element_text(size=9)) +
  facet_wrap(vars(s_n_class)) #, scales="free")

sent_ad_plot
```



```
#ggsave("sent_ad.jpg", plot=sent_ad_plot)
```

Discussion

My questions were about the evolution of the Israelites' relationship with God and the nature of that relationship.

The data suggests that the relationship evolved over time. The sentiments in Genesis, especially, were positive about the people. Conflicts were evident in the word choices and sentiments in Exodus, Leviticus, and Numbers. Then in Deuteronomy, as the Israelites prepare to enter the promised land, they are given a description of the expectations for their behavior for holding up their end of the covenant.

If I were to sum it up, I would say that the Pentateuch is the story of a people growing into maturity. They begin with dependence and end with responsibility.