Exploring and Tagging Text

Ruxin Liu, Anna Cook, Bruce Mallory, Jenna Moscaritolo, Jung Hwa Yeom, Haoqi Wang

11/17/2020

Introduction:

In this assignment, we used the True Numbers (tnum) package to explore and tag text from the book Pride and Prejudice by Jane Austen. We explored the frequency of common words and characters throughout each chapter of the book, and created visualizations to show these frequencies. The process is described in more detail below.

```
# Install the package
# devtools::install_github("truenumbers/tnum/tnum")
library(tnum)
# Get access to the database
tnum.authorize(ip="54.158.136.133")
```

```
## Available spaces: testspace, MEPED, alion-rf, shared-testspace, MSSP-1, wintergreen, testspace, tests
```

```
## Numberspace set to: shared-testspace
```

```
# Explore the text in the Pride and Prejudice
tax1 <-tnum.getDatabasePhraseList("subject", levels = 3)
tax1
```

```
##  [1] ""                                  "austen:jane:sense_and_sensibility"
##  [3] "austen:jane:pride_and_prejudice"   "austen:jane:mansfield_park"
##  [5] "austen:jane:emma"                  "austen:jane:northanger_abbey/"
##  [7] "austen:jane:persuasion"            "university:state:colorado"
##  [9] "university:state:montclair"        "subject"
## [11] "universit:state:montclair"         "university:state:motclair"
```

```
num1 <- tnum.query("*pride* has count:word", max = 5050)
```

```
## Returned 1 thru 5050 of 5050 results
```

```
num1_df <- tnum.objectsToDf(num1)
num2 <- tnum.query("*pride* has text")
```

```
## Returned 1 thru 10 of 5026 results
```

```
num2_df <- tnum.objectsToDf(num2)
```

Creating Functions For Chapter Number

The first step was to create functions to return the chapter numbers where a particular word appears, or where two words co-occur.

```
# For query
ch_num <- function(query) {
  n <- length(query)
  vector <- rep(0, n)
  for(i in 1 : n) {
    vector[i] <- as.numeric(substring(str_split(tnum.getAttrFromList(query[i], "subject"), "[:/]")[[1]]
  }
  return(vector)
}
```

```
# For data frame
ch_num_df <- function(df) {
  n <- nrow(df)
  vector <- rep(0, n)
  subject <- df$subject
  for(i in 1 : n) {
    vector[i] <- as.numeric(substring(str_split(subject[i], "[:/]")[[1]][4], 9))
  }
  return(vector)
}
```

Creating Functions For Paragraph Number

The next step was to create functions to return the paragraph numbers where a particular word appears, or where two words co-occur.

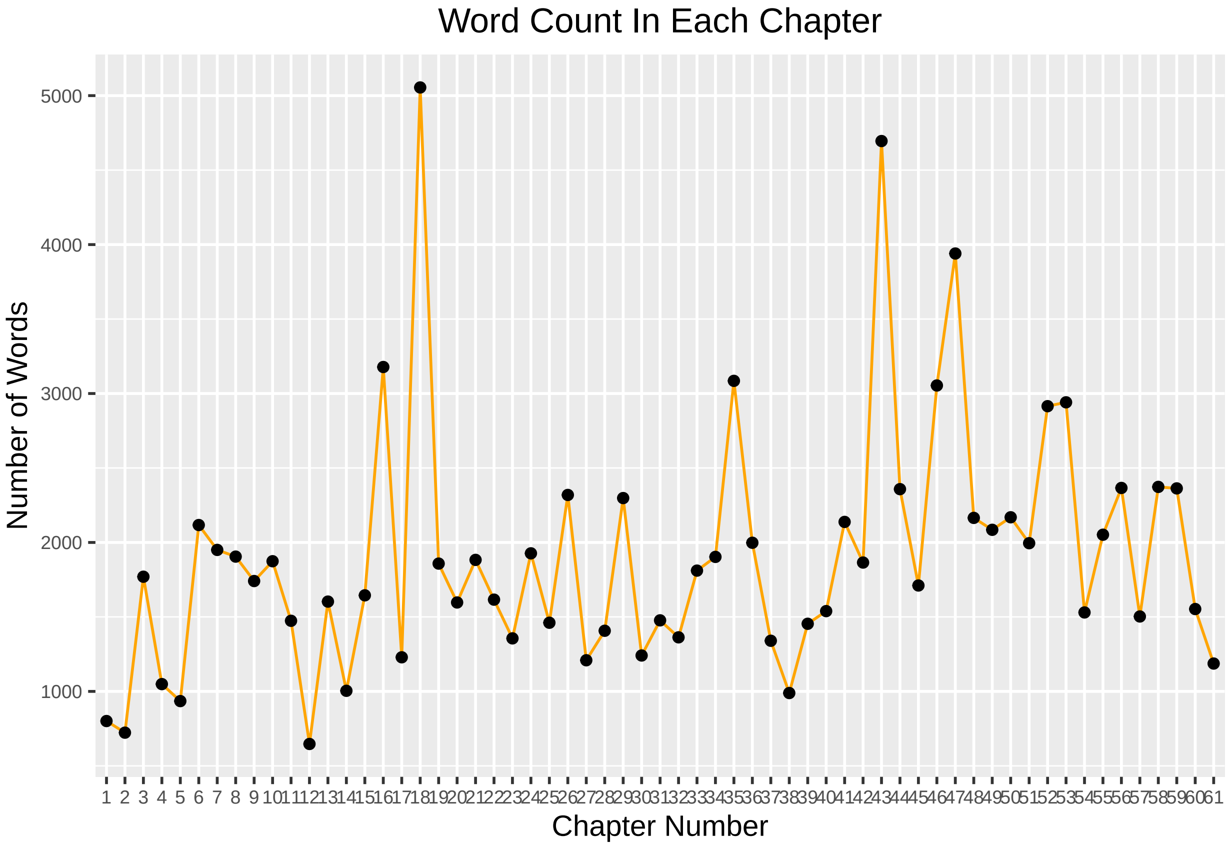
```
# For query
para_num <- function(query) {
  n <- length(query)
  vector <- rep(0, n)
  for(i in 1 : n) {
    vector[i] <- as.numeric(substring(str_split(tnum.getAttrFromList(query[i], "subject"), "[:/]")[[1]]
  }
  return(vector)
}
```

```
# For data frame
para_num_df <- function(df) {
  n <- nrow(df)
  vector <- rep(0, n)
  subject <- df$subject
  for(i in 1 : n) {
    vector[i] <- as.numeric(substring(str_split(subject[i], "[:/]")[[1]][5], 11))
  }
  return(vector)
}
```

```
word_data <- data.frame(ch = ch_num_df(num1_df), para = para_num_df(num1_df), num = num1_df$numeric.valu
word_data <- word_data %>%
  group_by(ch) %>%
  summarise(count = sum(num))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(data = word_data, aes(x= as.factor(ch), y = count, group=1)) +
  geom_line(color = "orange") +
  geom_point() +
  labs(x = "Chapter Number", y = "Number of Words") +
  ggtitle("Word Count In Each Chapter") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text = element_text(size = 7))
```



Exploring and Tagging the Word Love

Since Pride and Prejudice is a romantic novel, the first word that comes to mind is love, and therefore, the love tag is added. It turns out that the word "love" has appeared in the book 111 times.

Below we showed a phrase graph by using tnum package. This graph shows every occurance of the word love in the book. The center of the graph shows the book title (Pride and Prejudice). The chapters where the word love appears surround the title, the paragraph numbers surround the chapters, and the sentence numbers surround the paragraphs. Because the word love is used so many times, this visualization isn't very useful, so next we tried other methods of visualization.

```
tnum.tagByQuery("*pride* has text = REGEXP(\"love\")", "reference:love")
```

```
## list(modifiedCount = 111, tagged = 111, removed = 0)
```

```
love <- tnum.query("*pride* has text = REGEXP(\"love\")", max = 150)
```

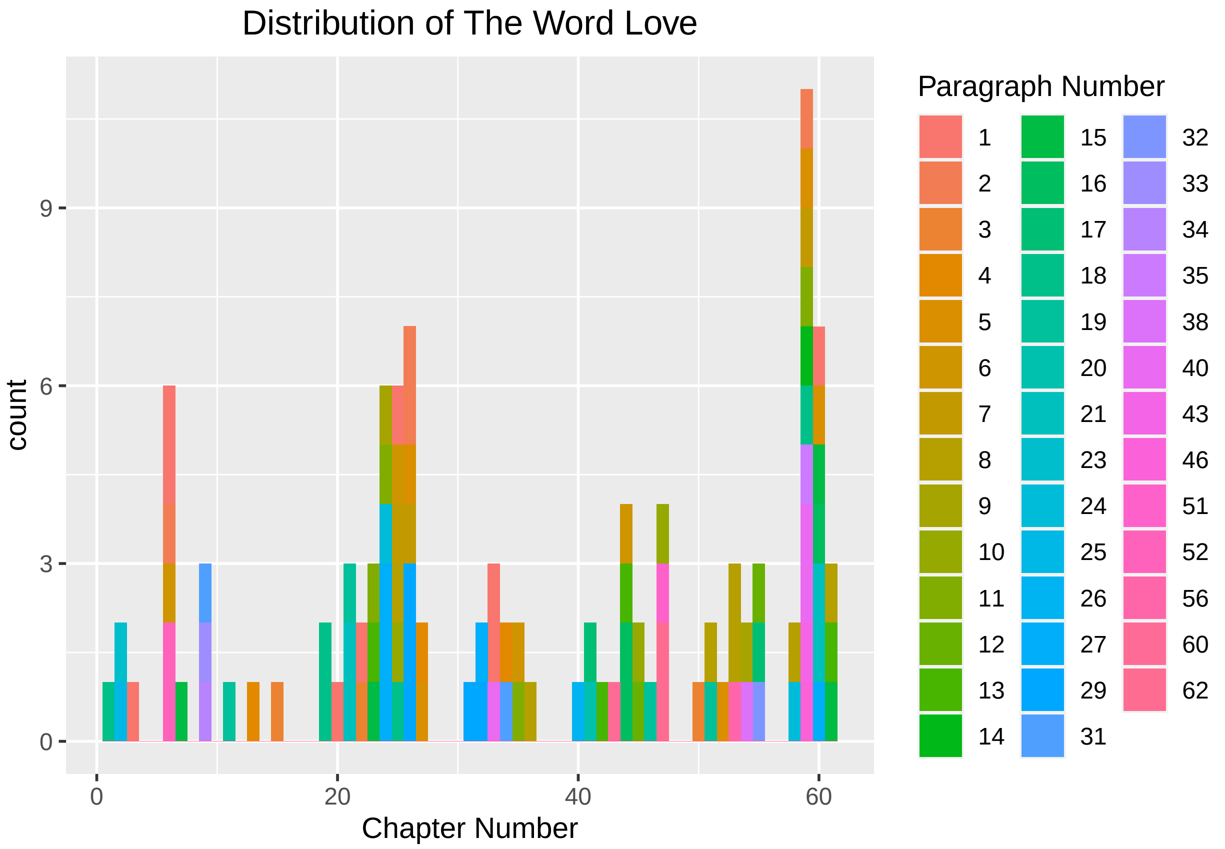
```
## Returned 1 thru 111 of 111 results
```

```
love_df <- tnum.objectsToDf(love)
# View(love_df)
piclove <- tnum.makePhraseGraphFromPathList(love_df$subject)
#tnum.plotGraph(piclove)
```

Histogram of Occurances of the Word Love

Once the functions were written, we were able to produce a distribution plot by using ggplot package, which presents the frequency of the word “love” appearing in different chapters and paragraphs. It is interesting but not surprising to see that the word “love” occurs relatively more often towards the end, where the story is approaching the happy ending. For the paragraph numbers, there isn’t a clear pattern that can be observed.

```
love_data <- data.frame(ch = ch_num(love), para = para_num(love))
ggplot(love_data, aes(ch)) +
  geom_histogram(aes(fill = as.factor(para)), binwidth = 1)+
  labs(title = "Distribution of The Word Love")+
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(fill = "Paragraph Number") +
  xlab("Chapter Number")
```



Exploring and Tagging the Word Marry

Another common word in Pride and Prejudice is the word marry. We followed the same steps as above to create a visualization for the frequency of the word marry appearing in different chapters and paragraphs.

```
tnum.tagByQuery("*pride* has text = REGEXP(\"marry\")", "reference:marry")
```

```
## list(modifiedCount = 58, tagged = 58, removed = 0)
```

```
marry <- tnum.query("*pride* has text = REGEXP(\"marry\")", max = 60)
```

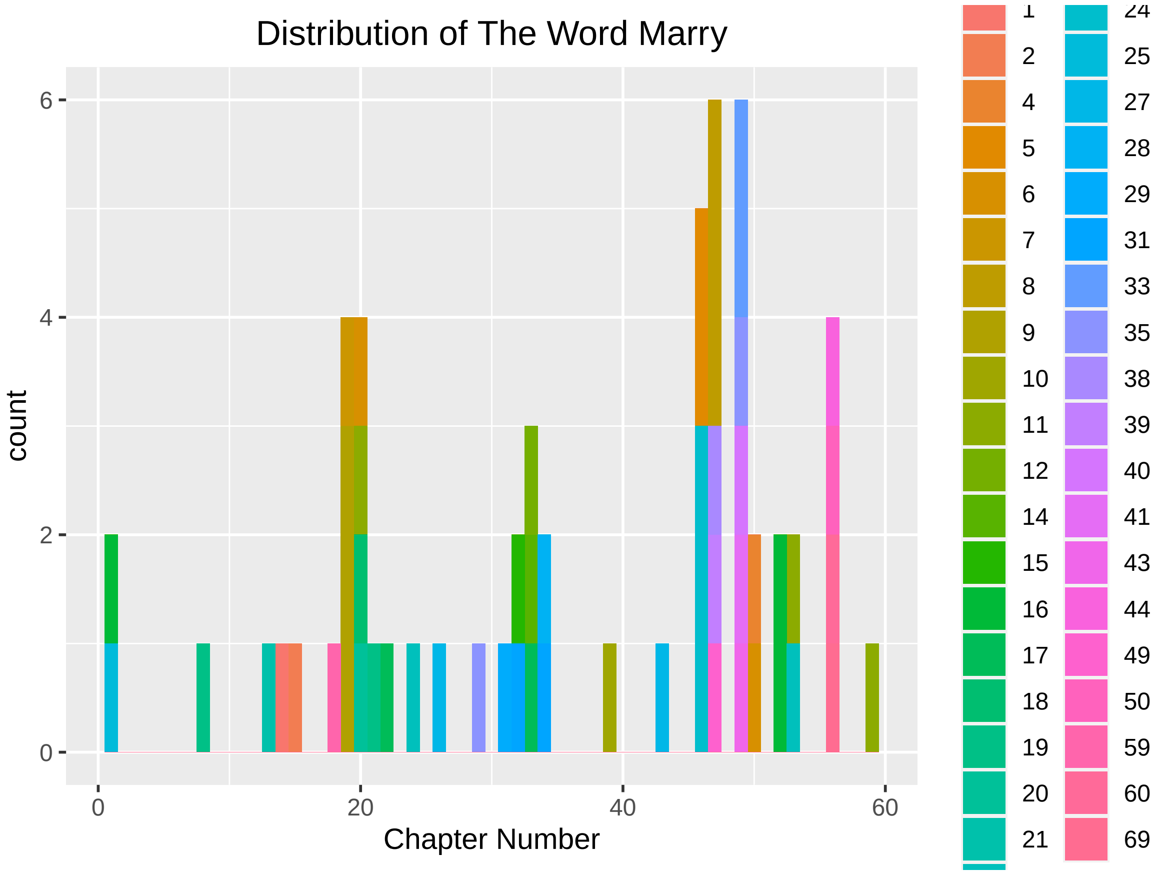
```
## Returned 1 thru 58 of 58 results
```

```
marry_df <- tnum.objectsToDf(marry)
#View(marry_df)
picmarry <- tnum.makePhraseGraphFromPathList(marry_df$subject)
#tnum.plotGraph(picmarry)
```

Histogram of Occurances of the Word Marry

From this plot, you can see that the word marry occurs more frequency toward the end of the book, similar to the word love, because the book is leading up to a happy ending.

```
marry_data <- data.frame(ch = ch_num(marry), para = para_num(marry))
ggplot(marry_data, aes(ch)) +
  geom_histogram(aes(fill = as.factor(para)), binwidth = 1)+
  labs(title = "Distribution of The Word Marry")+
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(fill = "Paragraph Number") +
  xlab("Chapter Number")
```



Three major male characters: Darcy, Bingley & Wickham

The next step in our analysis was to look at the relative frequencies of the names of three of the main male characters, Mr. Darcy, Mr. Bingley, and Mr. Wickham. For this comparison, we used a line graph to show the relative frequencies. From the graph, you can see that Mr. Darcy is mentioned throughout the entire book and he is mentioned extremely often in chapter 18. Mr. Bingley is also mentioned a lot. And Mr. Wickham is mentioned less frequently throughout the book, and his name is not mentioned once before the 15th chapter of the book. This shows that Mr. Darcy seems to be the most important of the male characters.

```
tnum.tagByQuery("*pride* has text = REGEXP(\"Darcy\")", "reference:Darcy")
```

```
## list(modifiedCount = 394, tagged = 394, removed = 0)
```

```
Darcy <- tnum.query("*pride* has text = REGEXP(\"Darcy\")", max = 400)
```

```
## Returned 1 thru 394 of 394 results
```

```
tnum.tagByQuery("*pride* has text = REGEXP(\"Bingley\")", "reference:Bingley")
```

```
## list(modifiedCount = 305, tagged = 305, removed = 0)
```

```
Bingley <- tnum.query("*pride* has text = REGEXP(\"Bingley\")", max = 310)
```

```
## Returned 1 thru 305 of 305 results
```

```
tnum.tagByQuery("*pride* has text = REGEXP(\"Wickham\")", "reference:Wickham")
```

```
## list(modifiedCount = 181, tagged = 181, removed = 0)
```

```
Wickham <- tnum.query("*pride* has text = REGEXP(\"Wickham\")", max = 190)
```

```
## Returned 1 thru 181 of 181 results
```

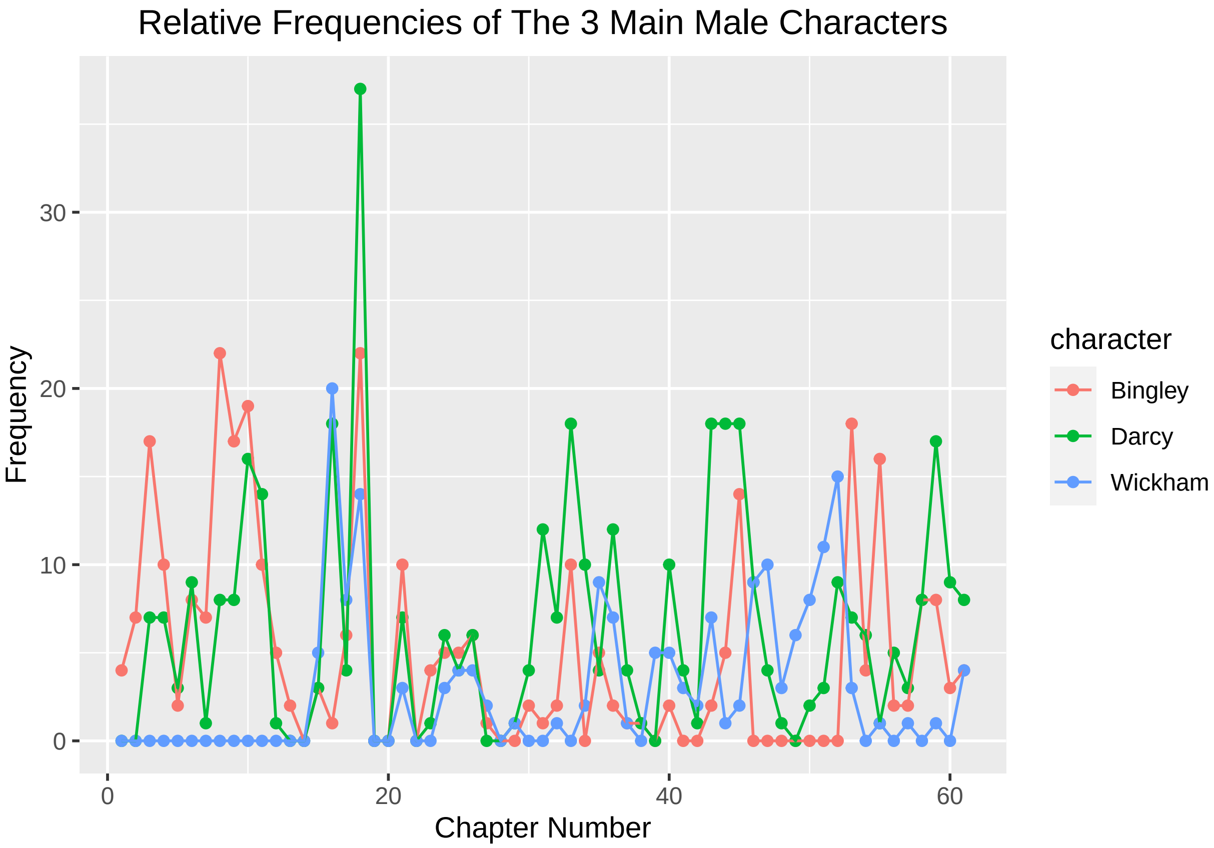
```
Darcy_n <- ch_num(Darcy)
Bingley_n <- ch_num(Bingley)
Wickham_n <- ch_num(Wickham)
```

```
Darcy_table <- as.data.frame(table(Darcy_n))
Bingley_table <- as.data.frame(table(Bingley_n))
Wickham_table <- as.data.frame(table(Wickham_n))
```

```
d <- data.frame(ch = Darcy_n, character = rep("Darcy", 394))
b <- data.frame(ch = Bingley_n, character = rep("Bingley", 305))
w <- data.frame(ch = Wickham_n, character = rep("Wickham", 181))
total <- rbind(b, d, w)
total <- total %>%
  group_by(character, ch) %>%
  summarise(freq = length(ch))
```

```
no_Darcy <- data.frame(character = rep("Darcy", 11),
                       ch = c(1, 2, 13, 14, 19, 20, 22, 27, 28, 39, 49),
                       freq = rep(0, 11))
no_Bingley <- data.frame(character = rep("Bingley", 17),
                       ch = c(14, 19, 20, 22, 28, 29, 34, 39, 41, 42, 46, 47, 48, 49, 50, 51, 52),
                       freq = rep(0, 17))
no_Wickham <- data.frame(character = rep("Wickham", 27),
                       ch = c(c(1:14), 19, 20, 22, 23, 28, 30, 31, 33, 38, 54, 56, 58, 60),
                       freq = rep(0, 27))
total <- rbind(total, no_Darcy, no_Bingley, no_Wickham)
total <- total %>%
  arrange(character, ch)
```

```
ggplot(data = total, aes(x = ch, y = freq, color = character)) +
  geom_point() +
  geom_line() +
  labs(title = "Relative Frequencies of The 3 Main Male Characters")+
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Chapter Number") +
  ylab("Frequency")
```



Exploring and Tagging where Darcy and Elizabeth co-occur

The final step in our analysis was to examining the instances where the names Darcy and Elizabeth co-occur within a sentence. The process was the same as the above analyses, except that we tagged only the sentences that included both names. Below is a histogram of the frequency of the two names co-occuring. As you can see, the two names occur together most frequently toward the end of the book. This makes sense, since Elizabeth and Mr. Darcy fall in love later on in the book, and the book leads to a happy ending for the two characters.

```
tnum.tagByQuery("*pride* has text = REGEXP(\"Elizabeth\")", "reference:Elizabeth_co")
```

```
## list(modifiedCount = 610, tagged = 610, removed = 0)
```

```
Elizabeth <- tnum.query("*pride* has text = REGEXP(\"Elizabeth\")", max = 610)
```

```
## Returned 1 thru 610 of 610 results
```

```
Elizabeth_df <- tnum.objectsToDf(Elizabeth)
tnum.tagByQuery("*pride* has text = REGEXP(\"Darcy\")", "reference:Darcy_co")
```

```
## list(modifiedCount = 394, tagged = 394, removed = 0)
```

```
Darcy_co <- tnum.query("*pride* has text = REGEXP(\"Darcy\")", max = 400)
```

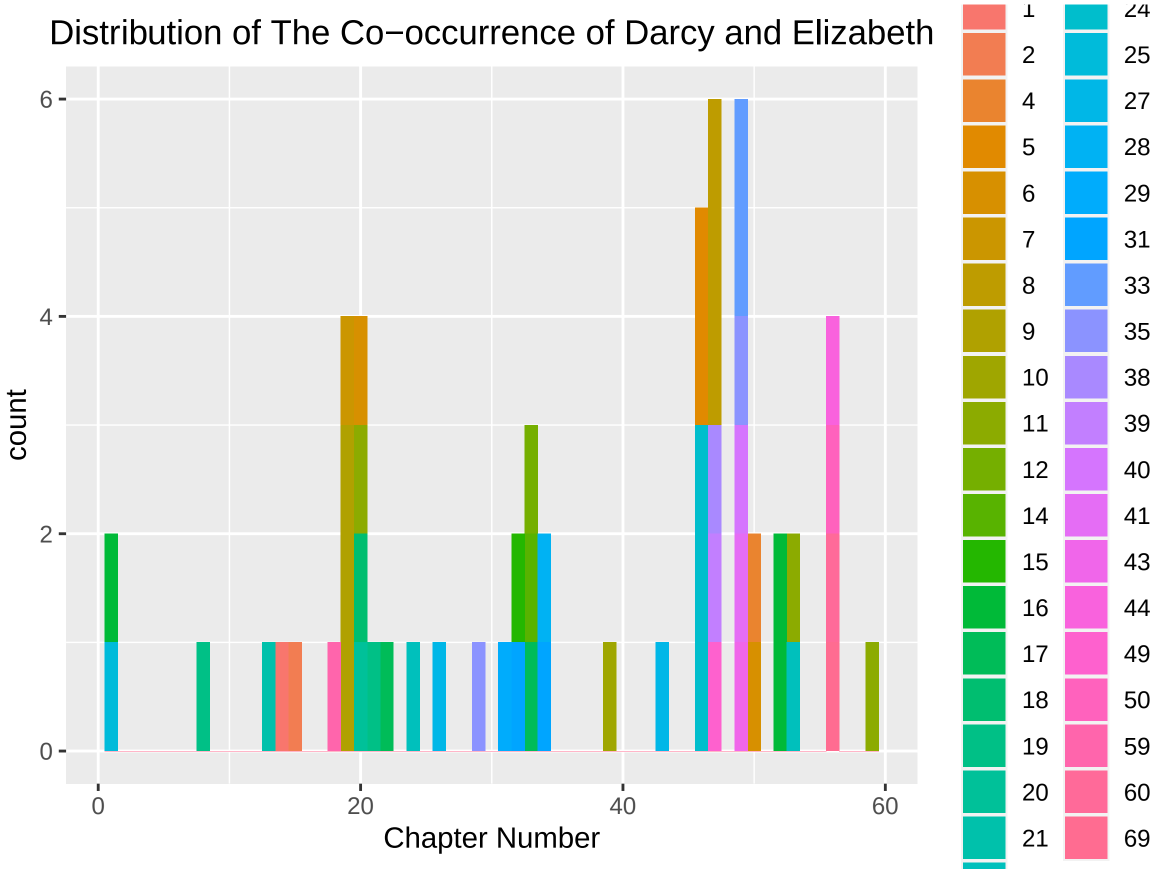
```
## Returned 1 thru 394 of 394 results
```

```
Darcy_co_df <- tnum.objectsToDf(Darcy_co)
```

Since we are working with a data frame, by using the dplyr package, we could find where our tag: Elizabeth_co occurs in the data frame: Darcy__co__df

```
co_occur <- dplyr::filter(Darcy_co_df, grepl('reference:Elizabeth_co', tags))
co_ch <- ch_num_df(co_occur)
co_para <- para_num_df(co_occur)
```

```
co_data <- data.frame(ch = co_ch, para = co_para)
ggplot(marry_data, aes(ch)) +
  geom_histogram(aes(fill = as.factor(para)), binwidth = 1)+
  labs(title = "Distribution of The Co-occurrence of Darcy and Elizabeth")+
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(fill = "Paragraph Number") +
  xlab("Chapter Number")
```



Suggestions for Tnum package

1. One thing we found difficult about working with the tnum package as compared to cran packages, was that there isn't documentation for what the tnum commands do. We had to do some experimenting to figure out what some of the commands were and how to use them.
2. Another thing we notice is that the words are case sensitive, which may probably make the query less accurate. Other than that, we didn't run into too many problems with the package.

```
# tnum.query("*pride* has text = REGEXP(\"elizabeth\")")-- Error in result$data$truenumbers[[1]] : subs
# tnum.query("*pride* has text = REGEXP(\"darcy\")")
```