Statistical Text Mining with the works of Mark Twain: An Introduction to Latent Dirichlet Allocation Method

---

A Thesis
Presented to
The Division of Statistics Department
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Olivia Xu

February 27, 2017

Approved for the Division
(Department of Statistics)

_____

Advisor N. Horton

Acknowledgements

I want to thank the Statistics Department of Amherst College, especially my advisor, Professor Nicholas Horton, for all that they have done for me. Through Nick's faith in my ability, I've become more a more confident and capable student and person.

Table of Contents

List of Tables

List of Figures

Abstract

Mark Twain is one of the most notable writers in American history. After Andrew and I completed our project using the texts of Katherine Mansfield, I knew I wanted to do something similar with a much larger and more complicated body of work. Mark Twain was an easy choice; he has written many famous works. Through sentiment analysis and stylometric analysis, I was able to see content and style changes in just one body of work, as well as across works over the span of his lifetime. I also included a short introduction to LDA in this project because it is a different way to analyze text, and it was something that Andrew and I had not done before. Lastly, to make this project easily replicable, I put the text files in a package called `marktwainr` so that other students can readily access them and play with them.

Dedication

To my parents.

Introduction

0.0.1 Mark Twain

Samuel Langhorne Clemens, or better known under his pen name Mark Twain, was a famous American author known for writinf "The Great American Novel" *The Adventures of Tom Sawyer*. Born in 1835, he wrote many works in his lifetime up until his death in 1910. Twain is a good choice for this text mining project because he wrote so much and because his works spanned across many years of his life and tackled so many different subjects.

0.0.2 Previously, Katherine Mansfield

I decided to do my extension on the works of Mark Twain because after tackling the short stories of Katherine Mansfield, I wanted to expand to works of much greater length. Additionally, because Mansfield's works were so short, they did not lend themselves well to the Latent Dirichlet Allocation method (my expository review topic).

0.0.3 Text Mining with Twain

I will be applying the same methods Andrew and I had previously used to textually analyze the works of Mansfield onto the works of Twain. Because Twain's works are much longer, I decided to pick six of his works, two from each of his "eras" (early, middle, and late), that summed up each of his phases according to experts. Then, using these six works, I will give an expository review of LDA for my peers.

Chapter 1

Twain Basics: *The Adventures of Tom Sawyer*

As I stated in the **Introduction**, due to Twain having such a large and extensive body of work, it was necessary to pick and choose works that I wanted to use for my analysis. A quick google search easily told me some of his most famous works, and so I tried to pick novels that he was famous for, while also making sure I had a selection from each phase of his writing career (early, middle, and end). The works that ended up meeting those criteria are listed below.

1. *Innocents Abroad* (early - 1869)
2. *Roughing It* (early - 1872)
3. *The Adventures of Tom Sawyer* (middle - 1876)
4. *The Adventures of Huckleberry Finn* (middle - 1884)
5. *A Connecticut Yankee in King Arthur's Court* (late - 1889)
6. *Pudd'nhead Wilson* (late - 1894)

These texts are all available online for free at the Project Gutenberg website `http://www.gutenberg.org/`. After extracting and cleaning the text from the website, (for a more detailed process, please see my *TwainWrangle.Rmd* file in GitHub) we are ready for some text analysis.

1.0.1 Goals and Objections

Using the works of Mark Twain, I hope to capture the essence of his writings using stylometric and sentiment markers. I will then also use his works to introduce a different type of text analysis, the Latent Dirichlet Allocation method.

1.0.2 Case Study: *The Adventures of Tom Sawyer*

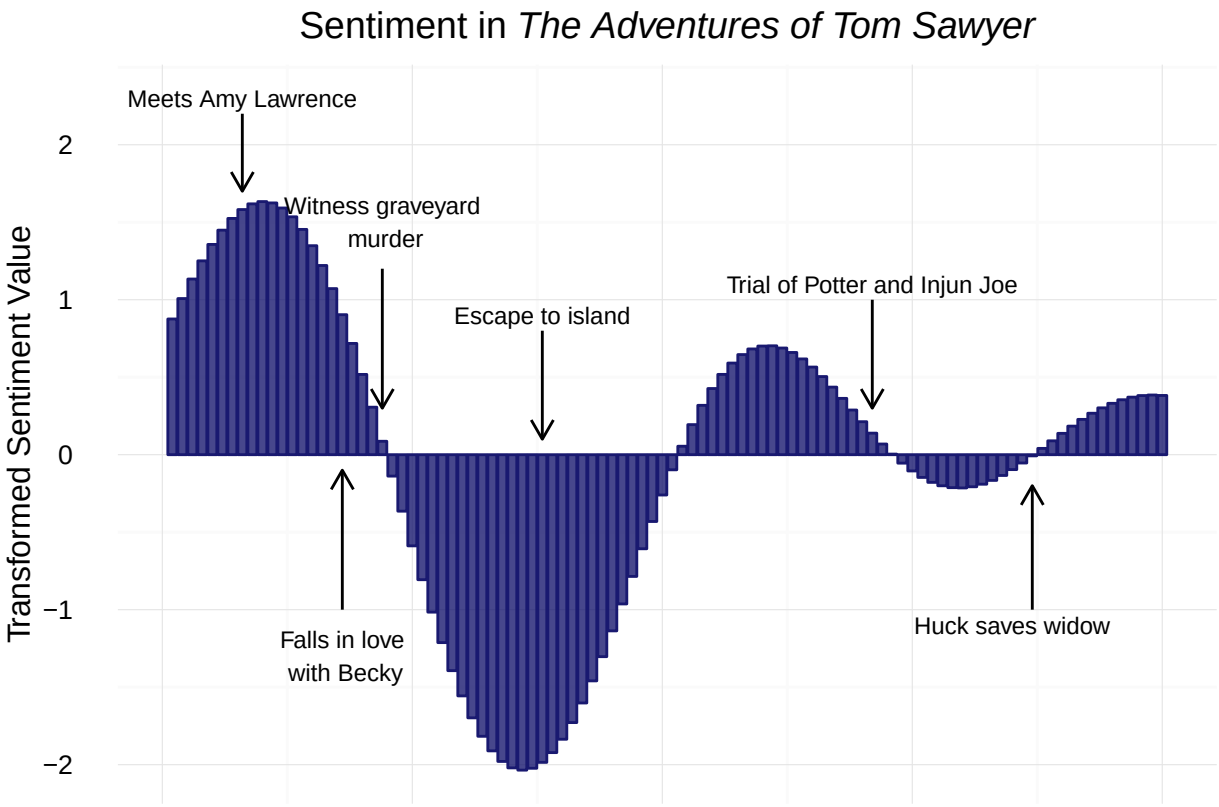
Like we did with our Mansfield project, let's first start off by looking at one novel before looking at them all together.

About: Tom Sawyer

A story that is familiar to most people, *The Adventures of Tom Sawyer* was Twain's "Great American novel". The book starts off with young Tom Sawyer whitewashing a fence as punishment. Tom is a mischievous little boy, and with his friend Huckleberry Finn, the two witness a murder at a graveyard. Scared that the murderer will come after them next, they escape to an island. Out of remorse, the two return home and help solve the murder mystery.

Sentiment Analysis

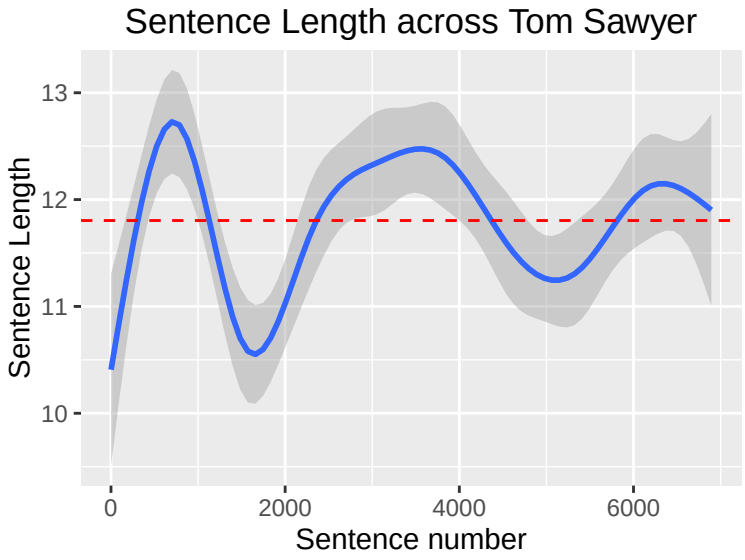
I will be using the `syuzhet` package in R to perform sentiment analysis (see the Mansfield project for more information and reasons why we chose syuzhet). The graph we get of the sentiment with the Fourier transform applied is exactly what we expect. The story starts off positive and ends positively. The highs and lows of the graph also match up nicely with key plot points; for example, we see an increase in positive sentiment when Tom meets these two pretty girls, but then the novel becomes much darker after the boys witness the murder of the doctor.

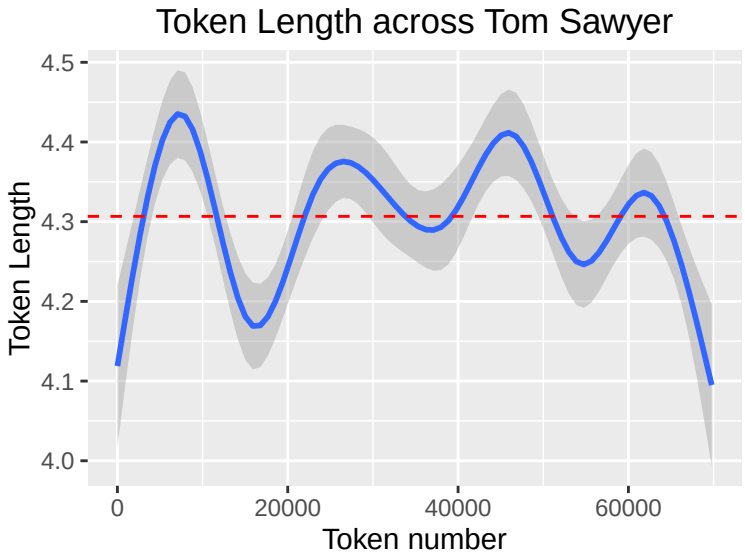


Stylometric Analysis

Using the `katherinemansfieldr` package that Andrew and I built last semester, I will be applying the functions we put in that package to measure certain stylometric

markers in Tom Sawyer. Below are two graphs depicting the sentence length over time in the book and token length over time. As we can see, while the graphs are interesting to look at, there doesn't seem to be a clear cut pattern in sentence or token length throughout the book. However, the two graphs do seem to loosely follow the Fourier sentiment graph from above; longer sentences and tokens seem to be related to the more positive portions of the novel.





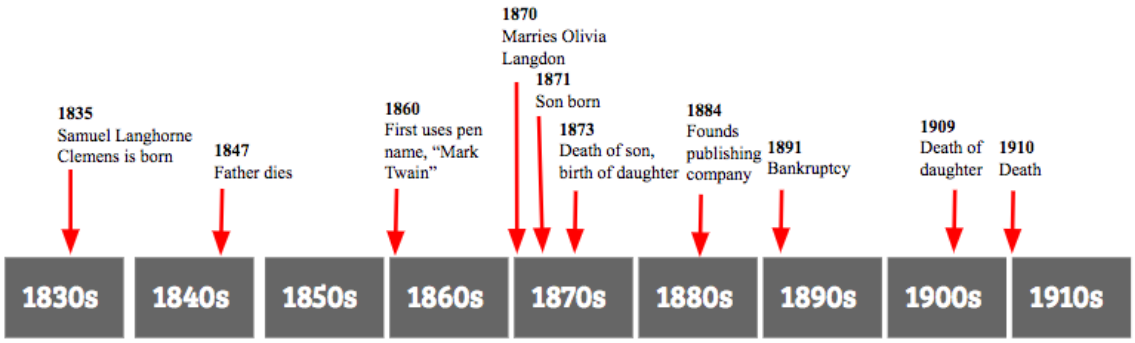
Chapter 2

The Adventures of Mark Twain

Now that I've gone through *The Adventures of Tom Sawyer*, I wanted to take a look at Twain's works as a whole (using the six novels stated in **Chapter 1**) and compare them to important milestones in his life. I'll use the same methods as I did in **Chapter 1**, and in **Chapter 3** I will introduce a different method for textual analysis.

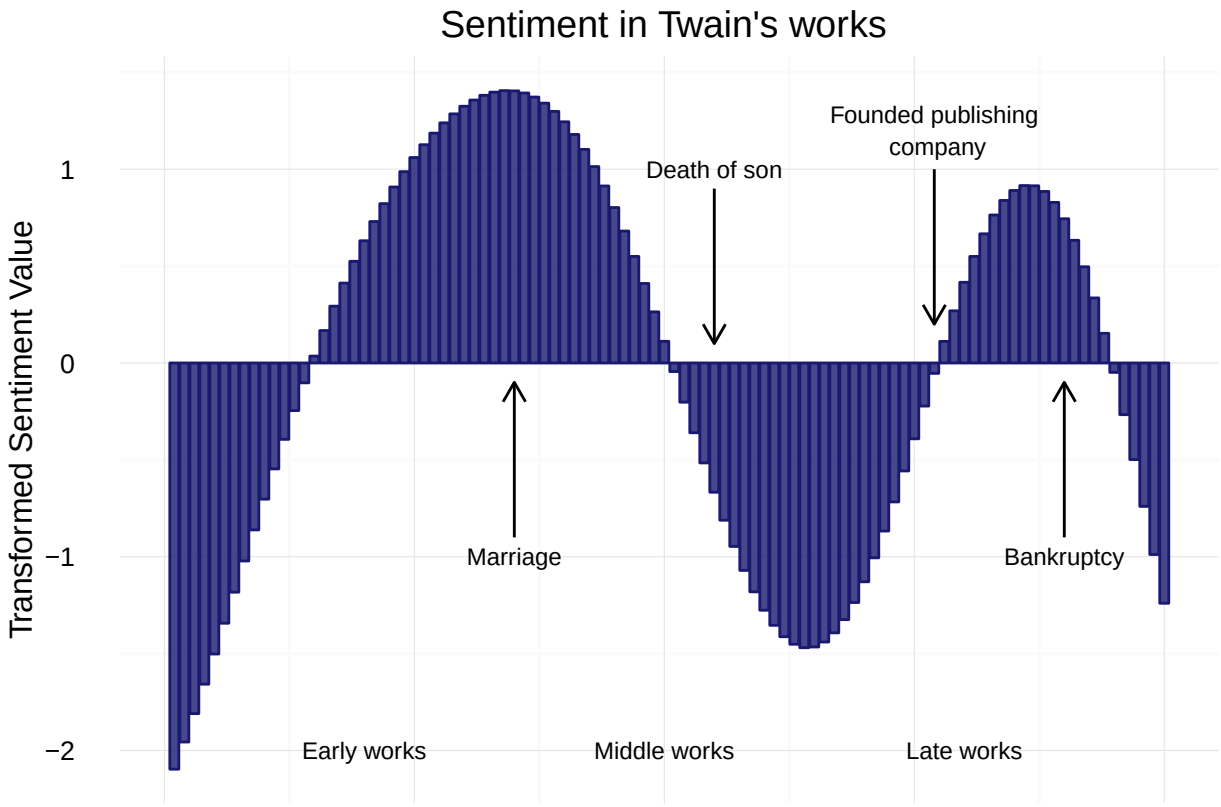
2.0.1 Important dates in Twain's life

Twain led a very exciting and inventive life. When he was young, his family moved to the city of Hannibal, MO. Hannibal was a frequent stop for steam boats, and it came to be quite the inspiration for Twain later in his works. Twain was educated at a local private school until he droppped out to become a printer's apprentice, where he discovered his love for writing. Later in his life, while living in California, Twain met his future wife, Olivia Langdon. They had one son and three daughters together, only of which one of them would make it to adulthood. Twain frequently invested in new inventions, and towards the end of his career, he was broke from poor investments. Below is a timeline of Twain's life in picture format.



2.0.2 Twain's life as seen through his works

Sentiment Analysis



As expected, milestones in Twain's life match up with the sentiment throughout his works. Appropriately, the sentiment reaches an all time high around the time he meets his wife, Olivia, and gets married. The sentiment plummets after the death of their first child and never quite recovers to the original happiness. The works end on a sad note when half his children have died and he's bankrupt.

Stylometric Analysis

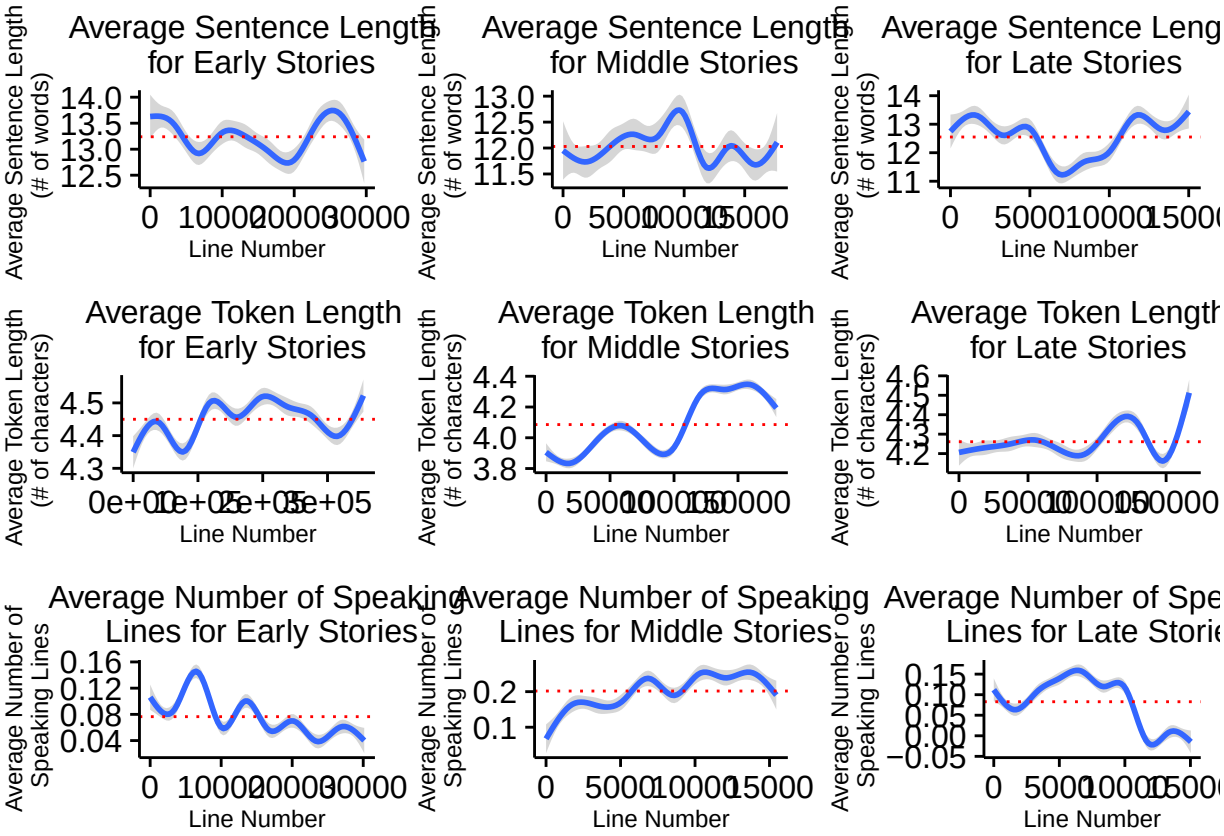
As we can see from the plots below, general trends between early, middle, and late works do not seem to differ drastically. One difference to note among sentence length is that Twain's earlier works seem to have longer sentences in general, which isn't surprising because his earlier works described landscapes and nature scenery. His later works had a larger variety in sentence length, which could be due to becoming a more critical and experienced writer over the years.
Among token length, the general trend across is that tokens are longer as the stories progress. The earlier works again have longer tokens than middle or late works, and that could be due to the emphasis on nature and scenery in those works. There are more speaking lines in the middle works than early or late works. Again this isn't surprising because of the content in those works. The trends are different between the

three phases, speaking lines increase as the novels progress in early and late works, but decrease for the middle works.
Because there were only two works used for each period, this analysis is of course limited because I didn't analyze Twain's full body of work for each period. However, I did pick his most notable pieces to analyze, so these graphs could be able to say something about the general essence of his writing from those periods.

```
Attaching package: 'cowplot'

The following object is masked from 'package:ggplot2':

    ggsave
```



Chapter 3

Latent Dirichlet Allocation: An Expository Review

3.0.1 What is LDA?

LDA is a statistical model used to detect underlying groups that would explain why some parts of the data look similar. It is commonly used to identify patterns that are not easily interpretable. In this chapter, I will be outlining how I applied the LDA method on Twain's works. As a resource and for inspiration, I used the "Data Until I Die!" blog post on topic modeling called "A Rather Nosy Topic Model Analysis of the Enron Email Corpus".

3.0.2 Twain and LDA

Because LDA is used to find patterns and meaning in a large amount of documents, I needed to break up my Twain documents in a logical fashion so that I could have more than six documents (from the original six books). I decided to break each book up by chapters, which seemed like the best way to break the stories up into smaller increments but still have them make sense and be complete in themselves. I ended up with somewhere close to 300 chapters after I run my code.

The first thing the blog did was to create a string with stop words. A stop word is a word that we're not interested in looking at, so it gets thrown out in the process of finding the natural groupings. Using the blog post's stop words list as an inspiration, I added my own words to the list. I kept going back and forth between the output and the stop words list to add words that I didn't want in the output (mostly every day words that didn't tell me much about the topics).

Next, I had to clean up the Corpus. A Corpus is just a collection of text files, in this case, it is the collection of all the individual chapters from the six books. I did this by making all the text lower case (`tolower` parameter, ensures that "Twain" doesn't show up as something from "twain"), removing punctuation (`removePunctuation` parameter), removing numbers (`removeNumbers` parameter), removing stop words (`stopwords` parameter), removing stemming (`stemming`, which reduces words of the same root to the same thing, ie "dog", "dog-like", "puppy" are all the same thing),

and looking at words of three characters or more (`wordLengths`). The last parameter, `weighting`, indicates that we are going to weight the importance of words by the standard term frequency (`tf`) method.

```
dtm.control = list(
  tolower = T,
  removePunctuation = T,
  removeNumbers = T,
  stopwords = c(stopwords("english"),extendedstopwords),
  stemming = T,
  wordLengths = c(3,Inf),
  weighting = weightTf)

dtm = DocumentTermMatrix(twaincorpus, control=dtm.control)
dtm = removeSparseTerms(dtm,0.999)
dtm = dtm[rowSums(as.matrix(dtm))>0,]
```

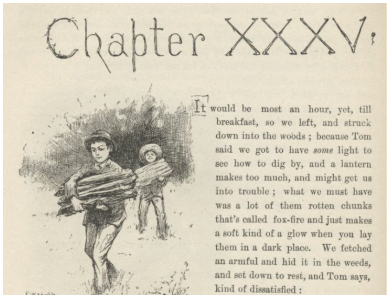
After cleaning the Corpus, we're ready to find our topics. The last step is to decide how many topics we want `R` to find. For this example, I decided to use 4 categories (k=4). I went back and forth between this output and the extended stop words list to add necessary words I didn't want included in the topics.

```
k = 4
lda.model = LDA(dtm, k)
terms(lda.model,20)
```

```
      Topic 1    Topic 2    Topic 3    Topic 4
 [1,] "day"      "mile"     "day"      "king"
 [2,] "hundr"    "feet"     "peopl"    "tell"
 [3,] "peopl"    "littl"    "mine"     "littl"
 [4,] "littl"    "hundr"    "hundr"    "jim"
 [5,] "citi"     "water"    "dollar"   "sir"
 [6,] "hand"     "hors"     "hand"     "boy"
 [7,] "look"     "day"      "littl"    "hand"
 [8,] "church"   "mountain" "hous"     "day"
 [9,] "thousand" "lake"     "street"   "head"
[10,] "eye"      "night"    "friend"   "look"
[11,] "beauti"   "hour"     "name"     "ill"
[12,] "stone"    "look"     "pass"     "night"
[13,] "pictur"   "thousand" "half"     "tri"
[14,] "world"    "tree"     "citi"     "talk"
[15,] "land"     "rock"     "world"    "dat"
[16,] "feet"     "desert"   "thousand" "peopl"
[17,] "wall"     "found"    "offic"    "that"
[18,] "live"     "wide"     "matter"   "told"
```

```
[19,] "life"     "eye"      "night"    "mind"
[20,] "water"    "island"   "hors"     "pretti"
```

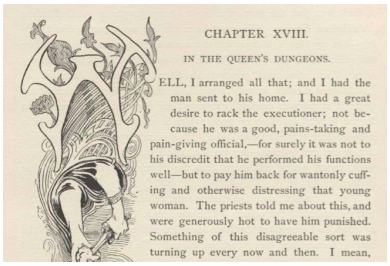
Topic 1 looks like it is related to time of day and relationships with other people. Topic 2 looks like it has to do with time of day again and speaking. The third topic has to do with nature and the outdoors. Lastly, topic 4 looks like it has to do with noblemen and religion. To see what these topics look like in context, let's look at some randomly selected chapters that scored highly in each category.



Chapter XXXV.

It would be most an hour, yet, till breakfast, so we left, and struck down into the woods; because Tom said we got to have *some* light to see how to dig by, and a lantern makes too much, and might get us into trouble; what we must have was a lot of them rotten chunks that's called fox-fire and just makes a soft kind of a glow when you lay them in a dark place. We fetched an armful and hid it in the weeds, and set down to rest, and Tom says, kind of dissatisfied:

This is one of the final chapters from *The Adventures of Huckleberry Finn*. This chapter describes the scene where Tom and Huck are planning their escape. There is a lot of arguing going on because the boys are pressed for time and need a plan quick. Topic 1 very loosely captures the essence of this chapter.

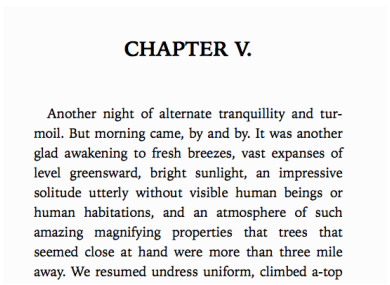


CHAPTER XVIII.

IN THE QUEEN'S DUNGEONS.

ELL, I arranged all that; and I had the man sent to his home. I had a great desire to rack the executioner; not because he was a good, pains-taking and pain-giving official,—for surely it was not to his discredit that he performed his functions well—but to pay him back for wantonly cuffing and otherwise distressing that young woman. The priests told me about this, and were generously hot to have him punished. Something of this disagreeable sort was turning up every now and then. I mean,

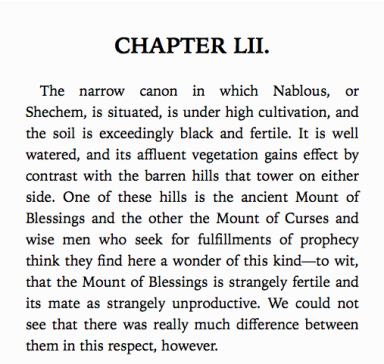
This is a chapter from *A Connecticut Yankee in King Arthur's Court*. Here, an imprisoned man states that a king could go unrecognized in the streets if wearing peasant's clothing. The king tests it out and finds that it is indeed true. There is a lot of dialogue and inner stream of consciousness in this chapter, which again, is very loosely captured by Topic 2.



CHAPTER V.

Another night of alternate tranquillity and turmoil. But morning came, by and by. It was another glad awakening to fresh breezes, vast expanses of level greensward, bright sunlight, an impressive solitude utterly without visible human beings or human habitations, and an atmosphere of such amazing magnifying properties that trees that seemed close at hand were more than three mile away. We resumed undress uniform, climbed a-top

This excerpt is from *Roughing It*. This chapter describes a journey by mule, and they are currently somewhere in Nevada. Topic 3 perfectly sums up this chapter, as it is full of flowery descriptions of nature and the natural world around them.



CHAPTER LII.

The narrow canon in which Nablous, or Shechem, is situated, is under high cultivation, and the soil is exceedingly black and fertile. It is well watered, and its affluent vegetation gains effect by contrast with the barren hills that tower on either side. One of these hills is the ancient Mount of Blessings and the other the Mount of Curses and wise men who seek for fulfillments of prophecy think they find here a wonder of this kind—to wit, that the Mount of Blessings is strangely fertile and its mate as strangely unproductive. We could not see that there was really much difference between them in this respect, however.

This is a chapter taken from *Innocents Abroad*. In this chapter, Twain wonders about Jesus's life as a boy because they are at Nazareth, Jesus's boyhood home. Topic 4 also succintly describes the essence of this chapter.

Chapter 4

marktwainr

4.0.1 About

To easily replicate the analysis performed in this project, I have compiled an R package called `marktwainr` so that anybody can easily use and play with the selected works of Mark Twain. To install this package, follow the code below.

```
devtools::install_github("oliviaxu17/marktwainr")
library(marktwainr)
```

Conclusion

Mark Twain wrote a full body of works that are an excellent source of data for statistics practice and text mining. In this project, I've explored different methods of stylometric analysis and explored sentiment analysis using the `szyuzhet` package. Twain's writing style and content certainly changed over his life, and through these project, we were able to see how they changed and postulate reasons for these changes. Hopefully, future students will be able to use this project as a guide to learning text mining.

Appendix A

Appendix

A.1 Chapter 1

A.1.1 Sentiment Analysis

```
# packages needed
library(mosaic)
library(XML)
library(readr)
library(devtools)
library(stringr)
library(RCurl)
library(syuzhet)
library(tm)
```

```
# tom sawyer text website
tomsawyerURL <- "http://www.gutenberg.org/files/74/74-0.txt"
tomsawyertext <- scan(tomsawyerURL, what = "character", sep = "\n")
```

```
# cleaning tom sawyer text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I") {
      startend <- i
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "CONCLUSION") {
      startend <- c(startend, j)
      return(startend)
```

```
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
  return (realtext)
}

# tom sawyer text
tomsawyerstartend <- findstart(tomsawyertext)
tomsawyertext <- actualtext(tomsawyertext, tomsawyerstartend)
tomsawyertext <- iconv(tomsawyertext,"WINDOWS-1252","UTF-8")
```

```
#sentiment data for tom sawyer
tomsent <- get_sentiment(tomsawyertext, "syuzhet")
tomsenttrans <- as.numeric(get_transformed_values(tomsent,
                                  low_pass_size = 3,
                                  scale_vals = TRUE,
                                  scale_range = FALSE))
tomsentdata <- data.frame(cbind(linenumber = seq_along(tomsenttrans), ft = tomsenttrans
```

```
# tom sawyer sentiment graph with Fourier transform
ggplot(data = tomsentdata, aes(x = linenumber, y = ft)) +
        geom_bar(stat = "identity", alpha = 0.8, color = "midnightblue", fill = "midnigh
        theme_minimal() +
        ylab("Transformed Sentiment Value") +
        ggtitle(expression(paste("Sentiment in ", italic("The Adventures of Tom Sawyer"
        theme(axis.title.x=element_blank()) +
        theme(axis.ticks.x=element_blank()) +
        theme(axis.text.x=element_blank()) +
ggplot2::annotate("text", size = 3, x = c(8, 18, 22, 38, 71, 85),
                  y = c(2.3, -1.3, 1.5, .9, 1.1, -1.1),
                  label = c("Meets Amy Lawrence", "Falls in love\n with Becky",
                            "Witness graveyard\n murder", "Escape to island",
                            "Trial of Potter and Injun Joe", "Huck saves widow")) +
        ggplot2::annotate("segment", arrow = arrow(length = unit(0.03, "npc")),
                  x = c(8, 18, 22, 38, 71, 87), xend = c(8, 18, 22, 38, 71, 87),
                  y = c(2.2, -1, 1.2, .8, 1, -1),
                  yend = c(1.7, -.1 , 0.3, .1, .3, -.2))
```

A.1.2 Stylometric Analysis

```
# installing katherinemansfield package
devtools::install_github("Amherst-Statistics/katherinemansfieldr")
library(katherinemansfieldr)
```

```
# finding sentence length in Tom Sawyer
sentlen <- function(sentences) {
  senlen <- 0
  for (i in 1:length(sentences)) {
    sentlen <- sapply(gregexpr("\\W+", sentences[i]), length) + 1
    senlen <- c(senlen, sentlen)
  }
  senlen <- senlen[-1]
  return(senlen)
}

tomSentences <- extract_sentences(tomsawyertext)
tomsenlen <- sentlen(tomSentences)
```

```
# plotting tom sawyer sentence length
data <- as.data.frame(cbind(tomsenlen, c(1:length(tomsenlen))))
ggplot(data, aes(x = V2, y = tomsenlen)) + geom_smooth() +
  geom_hline(yintercept = mean(data$tomsenlen), color = "red", linetype = "dashed"
```

```
# finding token length in tom sawyer
tomToken <- extract_token(tomsawyertext)

tokenlen <- function(token) {
  toklen <- 0
  for (i in 1:length(token)) {
    tokenlen <- nchar(token[i])
    toklen <- c(toklen, tokenlen)
  }
  toklen <- toklen[-1]
  return(toklen)
}

tokenlength <- tokenlen(tomToken)
```

```
# plotting token length in tom sawyer
data <- as.data.frame(cbind(tokenlength, c(1:length(tokenlength))))
ggplot(data, aes(x = V2, y = tokenlength)) + geom_smooth() +
  geom_hline(yintercept = mean(data$tokenlength), color = "red", linetype = "dashe
```

A.2 Chapter 2

```
# Mark Twain life timeline
library(png)
library(grid)
img <- readPNG("twaintimeline.png")
grid.raster(img)
```

A.2.1 Twain's life as seen through his works

Sentiment Analysis

```
# packages needed
library(mosaic)
library(XML)
library(readr)
library(devtools)
library(stringr)
library(RCurl)
library(syuzhet)
library(tm)
```

```
# importing texts
roughingitURL <- "http://www.gutenberg.org/files/3177/3177.txt"
roughingittext <- scan(roughingitURL, what = "character", sep = "\n")
innocentsURL <- "http://www.gutenberg.org/files/3176/3176-0.txt"
innocentstext <- scan(innocentsURL, what = "character", sep = "\n")
huckfinnURL <- "http://www.gutenberg.org/files/76/76-0.txt"
huckfinntext <- scan(huckfinnURL, what = "character", sep = "\n")
tomsawyerURL <- "http://www.gutenberg.org/files/74/74-0.txt"
tomsawyertext <- scan(tomsawyerURL, what = "character", sep = "\n")
connyankeeURL <- "http://www.gutenberg.org/files/86/86-0.txt"
connyankeetext <- scan(connyankeeURL, what = "character", sep = "\n")
puddURL <- "http://www.gutenberg.org/files/102/102-0.txt"
puddtext <- scan(puddURL, what = "character", sep = "\n")
```

```
# cleaning roughing it text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I.") {
      startend <- i
```

```
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "End of Project Gutenberg's Roughing It, by Mark Twain (Samuel
      startend <- c(startend, j)
      return(startend)
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
  return (realtext)
}

# roughing it text
roughstartend <- findstart(roughingittext)
roughingittext <- actualtext(roughingittext, roughstartend)
```

```
# cleaning innocents abroad text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I.") {
      startend <- i
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "End of the Project Gutenberg EBook of The Innocents Abroad") {
      startend <- c(startend, j)
      return(startend)
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
  return (realtext)
}

# innocents abroad text
innocentstartend <- findstart(innocentstext)
innocentstext <- actualtext(innocentstext, innocentstartend)
```

```
# cleaning tom sawyer text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I") {
      startend <- i
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "CONCLUSION") {
      startend <- c(startend, j)
      return(startend)
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
  return (realtext)
}

# tom sawyer text
tomsawyerstartend <- findstart(tomsawyertext)
tomsawyertext <- actualtext(tomsawyertext, tomsawyerstartend)
tomsawyertext <- iconv(tomsawyertext,"WINDOWS-1252","UTF-8")
```

```
# cleaning huckleberry finn text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I.") {
      startend <- i
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "End of the Project Gutenberg EBook of Adventures of Huckleberry Finn
      startend <- c(startend, j)
      return(startend)
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
```

```
  return (realtext)
}

# huckleberry finn text
huckfinnstartend <- findstart(huckfinntext)
huckfinntext <- actualtext(huckfinntext, huckfinnstartend)
```

```
# cleaning connecticut yankee text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I") {
      startend <- i
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "End of the Project Gutenberg EBook of A Connecticut Yankee in
      startend <- c(startend, j)
      return(startend)
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
  return (realtext)
}

# connecticut yankee text
connyankeestartend <- findstart(connyankeetext)
connyankeetext <- actualtext(connyankeetext, connyankeestartend)
```

```
# cleaning pudd'n text
findstart <- function(text){
  startend <- 0
  for (i in 1:length(text)){
    if (text[i] == "CHAPTER I.") {
      startend <- i
    }
  }
  for (j in 50:length(text)){
    if (text[j] == "and the creditors sold him down the river.") {
      startend <- c(startend, j)
      return(startend)
```

```
    }
  }
}

actualtext <- function(text, startend){
  realtext <- text[startend[1]:startend[2]]
  return (realtext)
}

# pudd'n text
puddstartend <- findstart(puddtext)
puddtext <- actualtext(puddtext, puddstartend)
```

```
# sentiment across entire life
early <- c(roughingittext, innocentstext)
earlylen <- length(early)
middle <- c(huckfinntext, tomsawyertext)
midlen <- length(middle)
late <- c(connyankeetext, puddtext)

allworks <- c(early, middle, late)

allworkssent <- get_sentiment(allworks, "syuzhet")
allworkssenttrans <- as.numeric(get_transformed_values(allworkssent,
                                low_pass_size = 3,
                                scale_vals = TRUE,
                                scale_range = FALSE))
allworksdata <- data.frame(cbind(linenumber = seq_along(allworkssenttrans), ft = allwor
```

```
# sentiment across all works with Fourier transform
ggplot(data = allworksdata, aes(x = linenumber, y = ft)) +
        geom_bar(stat = "identity", alpha = 0.8, color = "midnightblue", fill = "midnigh
        theme_minimal() +
        ylab("Transformed Sentiment Value") +
        ggtitle(expression(paste("Sentiment in Twain's works "))) +
        theme(axis.title.x=element_blank()) +
        theme(axis.ticks.x=element_blank()) +
        theme(axis.text.x=element_blank()) +
  ggplot2::annotate("text", size = 3, x = c(20, 50, 80, 35, 55, 77, 90),
                y = c(-2, -2, -2, -1, 1, 1.2, -1),
                label = c("Early works", "Middle works", "Late works", "Marriage", "Dea
  ggplot2::annotate("segment", arrow = arrow(length = unit(0.03, "npc")),x = c(35, 55,
```

Stylometric Analysis

```
# packages needed
library(katherinemansfieldr)
# sentence length function
sentlen <- function(sentences) {
  senlen <- 0
  for (i in 1:length(sentences)) {
    sentlen <- sapply(gregexpr("\\W+", sentences[i]), length) + 1
    senlen <- c(senlen, sentlen)
  }
  senlen <- senlen[-1]
  return(senlen)
}

# finding sentence length in early, middle, late works
earlySent <- extract_sentences(early)
earlysenlen <- sentlen(earlySent)
earlysendata <- as.data.frame(cbind(earlysenlen, c(1:length(earlysenlen))))

middleSent <- extract_sentences(middle)
middlesenlen <- sentlen(middleSent)
middlesendata <- as.data.frame(cbind(middlesenlen, c(1:length(middlesenlen))))

lateSent <- extract_sentences(late)
latesenlen <- sentlen(lateSent)
latesendata <- as.data.frame(cbind(latesenlen, c(1:length(latesenlen))))
```

```
# packages needed
library(ggplot2)
# early sentence length plot
earlysenlenplot <- ggplot(earlysendata,
                    aes(x = V2, y = earlysenlen)) +
  geom_smooth() +
  geom_hline(yintercept = mean(earlysendata$earlysenlen),
             colour = "red", linetype = "dotted") +
  ylab("Average Sentence Length \n (# of words)") +
  xlab("Line Number") +
  ggtitle("Average Sentence Length \n for Early Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# middle sentence length plot
middlesenlenplot <- ggplot(middlesendata,
                    aes(x = V2, y = middlesenlen)) +
  geom_smooth() +
  geom_hline(yintercept = mean(middlesendata$middlesenlen),
             colour = "red", linetype = "dotted") +
  ylab("Average Sentence Length \n (# of words)") +
  xlab("Line Number") +
  ggtitle("Average Sentence Length \n for Middle Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# late sentence length plot
latesenlenplot <- ggplot(latesendata,
                    aes(x = V2, y = latesenlen)) +
  geom_smooth() +
  geom_hline(yintercept = mean(latesendata$latesenlen),
             colour = "red", linetype = "dotted") +
  ylab("Average Sentence Length \n (# of words)") +
  xlab("Line Number") +
  ggtitle("Average Sentence Length \n for Late Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# token length function
tokenlen <- function(token) {
  toklen <- 0
  for (i in 1:length(token)) {
    tokenlen <- nchar(token[i])
    toklen <- c(toklen, tokenlen)
  }
  toklen <- toklen[-1]
  return(toklen)
}

# token length in early, middle, late works
earlyToken <- extract_token(early)
earlytokenlength <- tokenlen(earlyToken)
earlytokendata <- as.data.frame(cbind(earlytokenlength, c(1:length(earlytokenlength))))

middleToken <- extract_token(middle)
middletokenlength <- tokenlen(middleToken)
```

```
middletokendata <- as.data.frame(cbind(middletokenlength, c(1:length(middletokenle

lateToken <- extract_token(late)
latetokenlength <- tokenlen(lateToken)
latetokendata <- as.data.frame(cbind(latetokenlength, c(1:length(latetokenlength))
```

```
# early token length plot
earlytokenplot <- ggplot(earlytokendata,
                         aes(x = V2, y = earlytokenlength)) +
  geom_smooth() +
  geom_hline(yintercept = mean(earlytokendata$earlytokenlength),
             colour = "red", linetype = "dotted") +
  ylab("Average Token Length \n (# of characters)") +
  xlab("Line Number") +
  ggtitle("Average Token Length \n for Early Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# middle token length plot
middletokenplot <- ggplot(middletokendata,
                         aes(x = V2, y = middletokenlength)) +
  geom_smooth() +
  geom_hline(yintercept = mean(middletokendata$middletokenlength),
             colour = "red", linetype = "dotted") +
  ylab("Average Token Length \n (# of characters)") +
  xlab("Line Number") +
  ggtitle("Average Token Length \n for Middle Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# late token length plot
latetokenplot <- ggplot(latetokendata,
                         aes(x = V2, y = latetokenlength)) +
  geom_smooth() +
  geom_hline(yintercept = mean(latetokendata$latetokenlength),
             colour = "red", linetype = "dotted") +
  ylab("Average Token Length \n (# of characters)") +
  xlab("Line Number") +
  ggtitle("Average Token Length \n for Late Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# early quotation plot
earlypunc <- freq_punct_line(early, """)

earlyquoteplot <- ggplot(earlypunc,
                    aes(x = line_index, y = left_quote)) +
  geom_smooth() +
  geom_hline(yintercept = mean(earlypunc$left_quote),
             colour = "red", linetype = "dotted") +
  ylab("Average Number of\n Speaking Lines") +
  xlab("Line Number") +
  ggtitle("Average Number of Speaking\n Lines for Early Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# middle quotation plot
middlepunc <- freq_punct_line(middle, """)

middlequoteplot <- ggplot(middlepunc,
                    aes(x = line_index, y = left_quote)) +
  geom_smooth() +
  geom_hline(yintercept = mean(middlepunc$left_quote),
             colour = "red", linetype = "dotted") +
  ylab("Average Number of\n Speaking Lines") +
  xlab("Line Number") +
  ggtitle("Average Number of Speaking\n Lines for Middle Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# late quotations plot
latepunc <- freq_punct_line(late, """)

latequoteplot <- ggplot(latepunc,
                    aes(x = line_index, y = left_quote)) +
  geom_smooth() +
  geom_hline(yintercept = mean(latepunc$left_quote),
             colour = "red", linetype = "dotted") +
  ylab("Average Number of\n Speaking Lines") +
  xlab("Line Number") +
  ggtitle("Average Number of Speaking\n Lines for Late Stories") +
  theme(axis.title.y = element_text(size = 9)) +
  theme(axis.title.x = element_text(size = 9)) +
  theme(plot.title = element_text(size = 12))
```

```
# packages needed
library(cowplot)
library(ggplot2)
# code to graph all 9 plots together
plot_grid(earlysenlenplot, middlesenlenplot, latesenlenplot,
          earlytokenplot, middletokenplot, latetokenplot,
          earlyquoteplot, middlequoteplot, latequoteplot,
          nrow = 3, ncol = 3)
```

A.3 Chapter 3

A.3.1 Twain and LDA

```
# packages needed
library(stringr)
library(plyr)
library(tm)
library(tm.plugin.mail)
library(SnowballC)
library(topicmodels)
```

```
# creating Corpus of Twain texts
twaincorpus <- Corpus(DirSource("/home/class17/sxu17/XuACComps/TwainTexts"))

# stop words list
extendedstopwords=c("a","about","above","across","after","MIME Version","forwarded
```

```
# cleaning Corpus
dtm.control = list(
  tolower = T,
  removePunctuation = T,
  removeNumbers = T,
  stopwords = c(stopwords("english"),extendedstopwords),
  stemming = T,
  wordLengths = c(3,Inf),
  weighting = weightTf)

dtm = DocumentTermMatrix(twaincorpus, control=dtm.control)
dtm = removeSparseTerms(dtm,0.999)
dtm = dtm[rowSums(as.matrix(dtm))>0,]
```

```
# results from LDA
k = 4
lda.model = LDA(dtm, k)
terms(lda.model,20)
```

```
# code to see how closely certain chapters match with each topic
twain.topics = posterior(lda.model, dtm)$topics
df.twain.topics = as.data.frame(twain.topics)
df.twain.topics = cbind(email=as.character(rownames(df.twain.topics)),
                        df.twain.topics, stringsAsFactors=F)
```

```
# finding chapters that are highly correlated to topic 1
sample(which(df.twain.topics$"1" > .95), 10)
topic1 <- twaincorpus[[72]]
topic1$content[1:10]
```

```
# displaying part of chapter that is high on topic 1
library(png)
library(grid)
img <- readPNG("huckfinn35.png")
grid.raster(img)
```

```
# finding chapters that are highly correlated to topic 2
sample(which(df.twain.topics$"2" > .95), 10)
topic2 <- twaincorpus[[10]]
topic2$content[1:10]
```

```
# displaying part of chapter that is high on topic 2
img <- readPNG("connyankee.png")
grid.raster(img)
```

```
# finding chapters highly correlated to chapter 3
sample(which(df.twain.topics$"3" > .95), 10)
topic3 <- twaincorpus[[208]]
topic3$content[1:10]
```

```
# dispalying part of chapter that is high on topic 3
img <- readPNG("roughing.png")
grid.raster(img)
```

```
# finding chapters highly correlated to chapter 4
sample(which(df.twain.topics$"4" > .95), 10)
topic4 <- twaincorpus[[132]]
topic4$content[1:10]
```

```
# displaying part of chapter that is high on topic 4
img <- readPNG("innocents.png")
grid.raster(img)
```

Appendix B

Chapter 4

B.0.1 About

```
# load marktwainr package code
devtools::install_github("oliviaxu17/marktwainr")
library(marktwainr)
```

References