Hideous Progeny: Tidytext Analysis of Frankenstein

2018 May 19
frankenstein / literature / sentiment analysis

> And now, once again, I bid my hideous progeny go forth and prosper. — Mary Wollstonecraft Shelley (London, 15 October 1831)

1 The Moden Prometheus

Mary Wollstonecraft Shelley's Frankenstein (The Modern Prometheus) is one of the best stories anyone can read. If anyone was to start their journey in the wonderful world of stories, Frankenstein will always be my top recommendation. I am so excited talking about Frankenstein, I had to reread it before completing this text analysis.

A lot of people know the legend of Frankenstein. I will avoid (or at least try) giving away summaries or spoilers even though I believe spoilers can do no harm to any aspiring reader of Shelley's masterpiece. Reading Frankenstein is an experience in itself. It is like a great journey into a foreign and mystical country. The traveler may share stories of her adventure, but never will the listeners understand or experience the real joy of the journey she took.

2 Pursuit for Frankenstein Begins

Downloading any public domain book available in Gutenberg is now easy thanks to `gutenbergr`. The `gutenberg_download()` function allows one to download any text in Gutenberg. In its simplest form, the only input needed in `gutenberg_download()` is the `gutenberg_id` or EBook number of the text one wants to download. In my case, I will use this Frankenstein text with EBook #84.

```
frankenstein <-
  gutenberg_download(84)

frankenstein %>%
  head(20) %>%
  kable(
    caption = "Sample Raw Downloaded Frankenstein Text from Guntenberg",
    align = rep("c", ncol(frankenstein))
  ) %>%
  kableExtra::kable_styling(
    full_width = TRUE
```

```
)
```

Table 2.1: Sample Raw Downloaded Frankenstein Text from Guntenberg

gutenberg_id	text
84	Frankenstein,
84	
84	or the Modern Prometheus
84	
84	
84	by
84	
84	Mary Wollstonecraft (Godwin) Shelley
84	
84	
84	
84	
84	Letter 1
84	
84	
84	St. Petersburgh, Dec. 11th, 17–
84	
84	TO Mrs. Saville, England
84	
84	You will rejoice to hear that no disaster has accompanied the

The downloaded text returns a tibble with two columns: `gutenberg_id` and `text`. The `text` column contains the words per line of the book. Note that with the empty spaces, this raw downloaded tibble requires a lot of cleaning.

3 Emptiness Filled

Since Frankenstein is a novel, it will be reasonable to study it by chapter. Unfortunately, the raw downloaded file does not contain a column for chapters, so I had to extract and divide the document into chapters by hand.

The simplest, easiest, most reproducible, and instinctive way to do this is by using the `fill()` function from the `dplyr` package. I first saw this method used in Julia Silge's topic modeling of Sherlock Holmes stories.

```
chapter_headers <-
  c(str_c("Letter", 1:4, sep = " "),
    str_c("Chapter", 1:24, sep = " "))

chapters <-
  frankenstein %>%
  mutate(chapter = ifelse(str_detect(text,
                                     str_c(START,
                                           chapter_headers,
                                           END,
                                           collapse = "|")),
                          text,
                          NA),
         line_number = row_number()) %>%
  fill(chapter) %>%
  drop_na() %>%
  filter(text != chapter, text != "")
```

The idea is to create a new column that will extract all the chapter headings and apply it as an identifier to all lines between chapter headings. For Frankenstein, the uniform format for chapter headings are Letter and Chapter followed by numbers. I created a new column called chapter then copied the text if it follows a chapter heading format and NA if it does not. I then used `fill()` to replace all NA cells with chapter headings above it. I then filtered out all NA rows, i.e. the lines above the first chapter and all empty lines in the text. I also had the help of the `stringr` and `rebus` package in creating and filtering strings. I also included a column to identify line number that will be useful when we separate the text into words.

4 Destruction and Creation

After separating the lines of text into their appropriate chapters, I separated the Frankenstein text into words. The easiest way to do this is with the `unnest_tokens()` function from the `tidytext` package. I also removed all stop words or words that supply unnecessary information by using `anti_join()` and the `get_stopwords()` and specifying the SMART lexicon.

```
words <-
  chapters %>%
  mutate(chapter = factor(chapter,
                          levels = chapter_headers)) %>%
  unnest_tokens(word, text) %>%
  anti_join(get_stopwords(source = "smart"),
            by = "word")
```

`unnest_tokens()` takes a tibble as its first argument, making it easy to use %>%, then the name of the new column of the unnested tokens, and the name of the column we want to unnest.

Now that the Frankenstein text data is tidy, we can start exploring.

5 Of Man, Of Life

One of the very first questions I answer when I explore text data is to determine the most commonly used words.

```
words %>%
  count(word) %>%
  top_n(20, n) %>%
  ggplot(aes(fct_reorder(word, n),
             n)) +
  geom_col(fill = palette_pander(2)[2]) +
  coord_flip() +
  labs(
    x = "Word",
    y = "Count"
  ) +
  theme_pander()
```

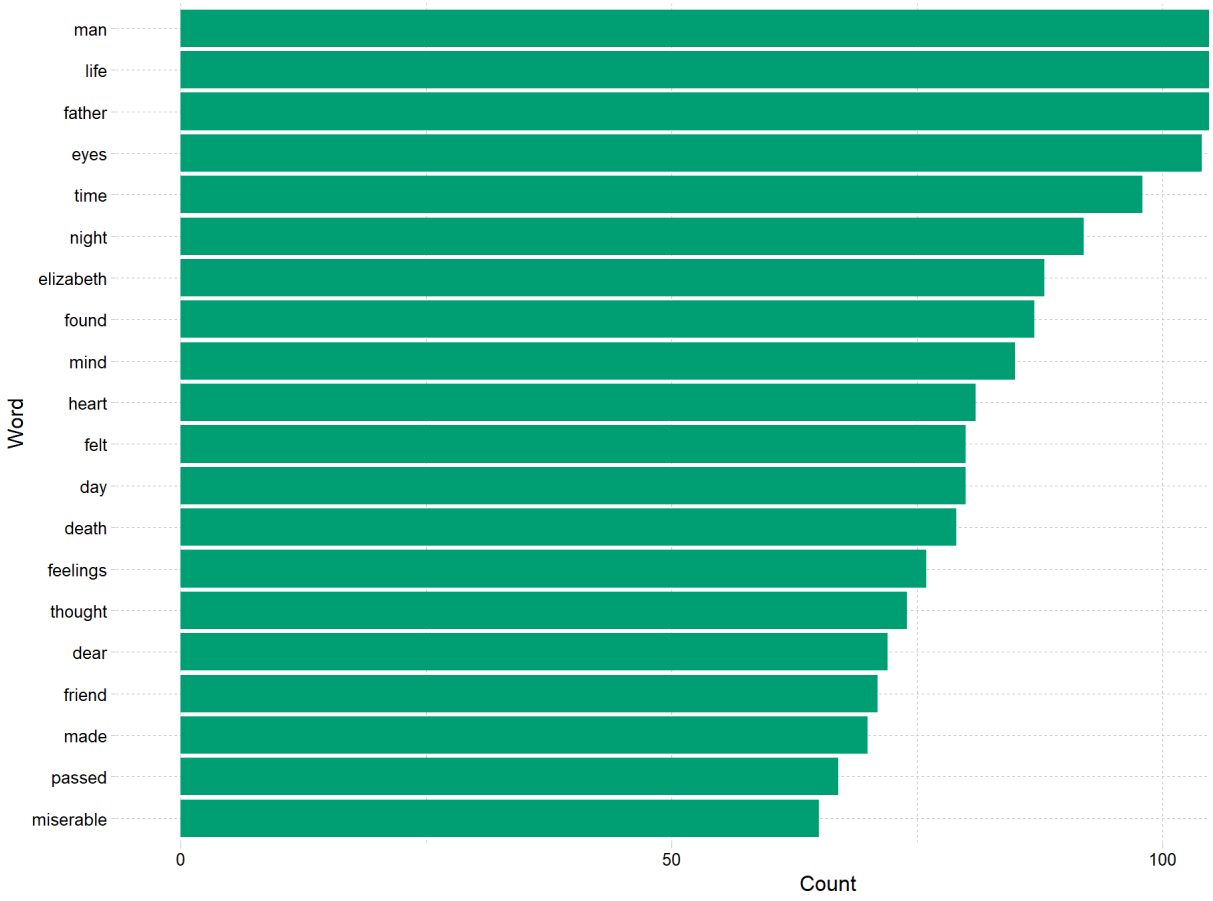

Figure 5.1: Most Used Words in Frankenstein

Fellow readers of Frankenstein should not be surprised that man and life is the top two words. Frankenstein is more about the madness of one man and the monster he created. It is about humanity and the interplay of life and death (these two words also in the top 20).

There are two important fathers in the story. Elizabeth also has an important role both in the story and Victor's life.

6 A Big Ending

After the most common words, I also ask what are the number of words per chapter.

```
n_words_median_chapter <-
  words %>%
  count(chapter) %>%
  with(median(n))

words %>%
  count(chapter) %>%
  ggplot(aes(fct_reorder(chapter,
                         n),
             n)) +
  geom_col(fill = palette_pander(2)[2]) +
  geom_hline(yintercept = n_words_median_chapter,
             color = "red3") +
  coord_flip() +
  scale_y_continuous(labels = comma) +
```

```
  labs(
    x = "Chapter",
    y = "Number of Words"
  ) +
  theme_pander()
```

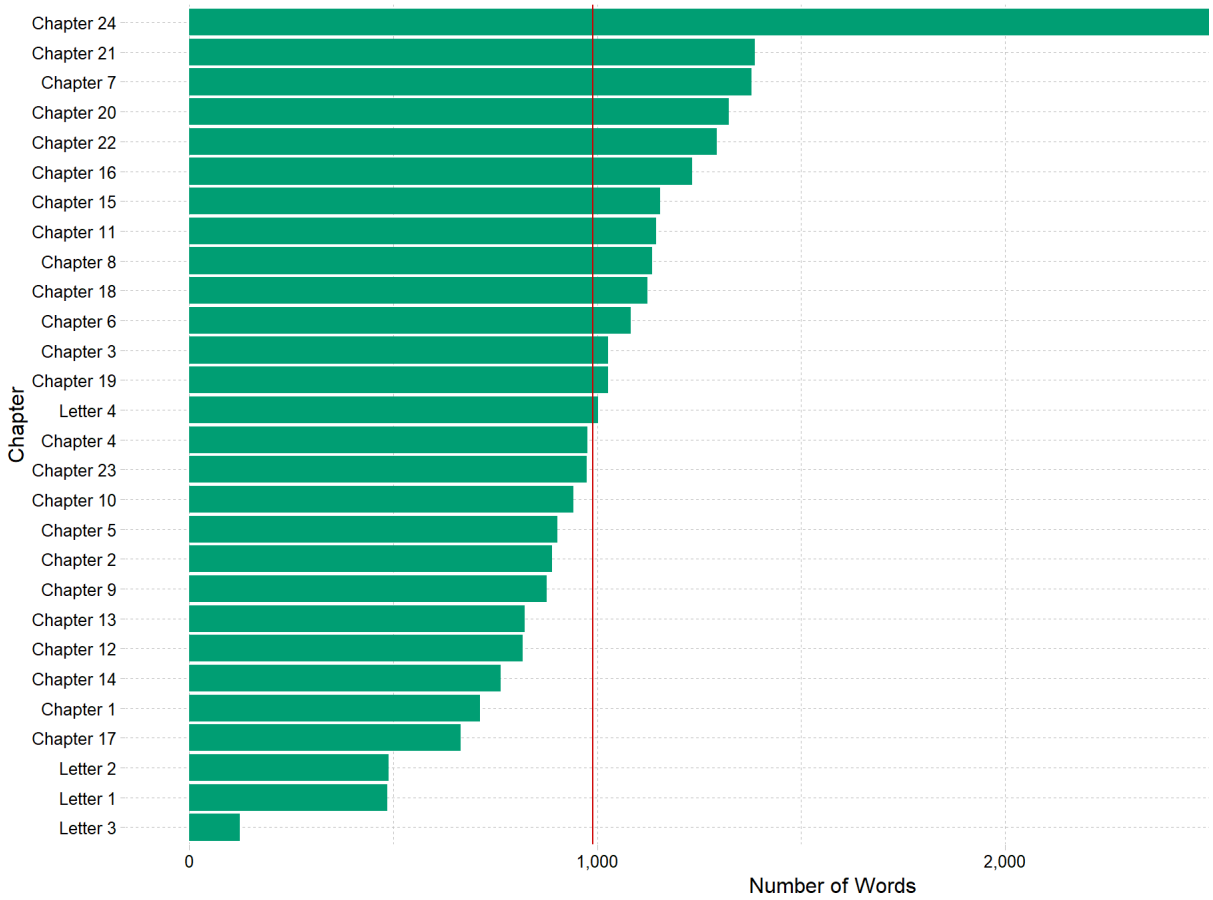

Figure 6.1: Number of Words per Chapter. Red Line is the Median Number of Words (989) per Chapter

It seems like Shelley could have used one or two more chapters, but maybe she was limited to only 24 chapters for her work. One of my favorite quotes is in the last chapter.

We will learn more about chapters later. Time to get more insight about the novel.

7 Uncontrollable Feelings

Sentiment analysis is another staple when working with text data. Words express emotions. Some express more negative emotions, others more positive. We can determine the overall feeling or mood of a text by looking at each word and weighing which emotions dominate the text more.

The `tidytext` package made sentiment analysis a bit easier with the `get_sentiments()` function and with the help of `dplyr`'s `inner_join()`.

Here, I used the AFINN and NRC lexicon.

7.1 Waves of Emotions

The AFINN lexicon scores words from -5 to 5 depending on the words emotions and intensity. I took the AFINN scores of each word in each chapter then took the total AFINN score of each chapter to determine the overall sentiment of the chapter.

```
words %>%
  inner_join(get_sentiments("afinn"),
             by = "word") %>%
  group_by(chapter) %>%
  summarise(sentiment_score = sum(score),
            positive_net = sentiment_score >= 0) %>%
  ggplot(aes(chapter,
             sentiment_score,
             fill = positive_net,
             label = sentiment_score)) +
  geom_col(show.legend = FALSE) +
  geom_hline(yintercept = 0,
             color = "red3") +
  scale_fill_pander() +
  theme_pander() +
  theme(axis.text.x = element_text(angle = 90),
        axis.ticks.x = element_blank()) +
  labs(
    x = "Chapter",
    y = "Net Sentiment Score"
  )
```

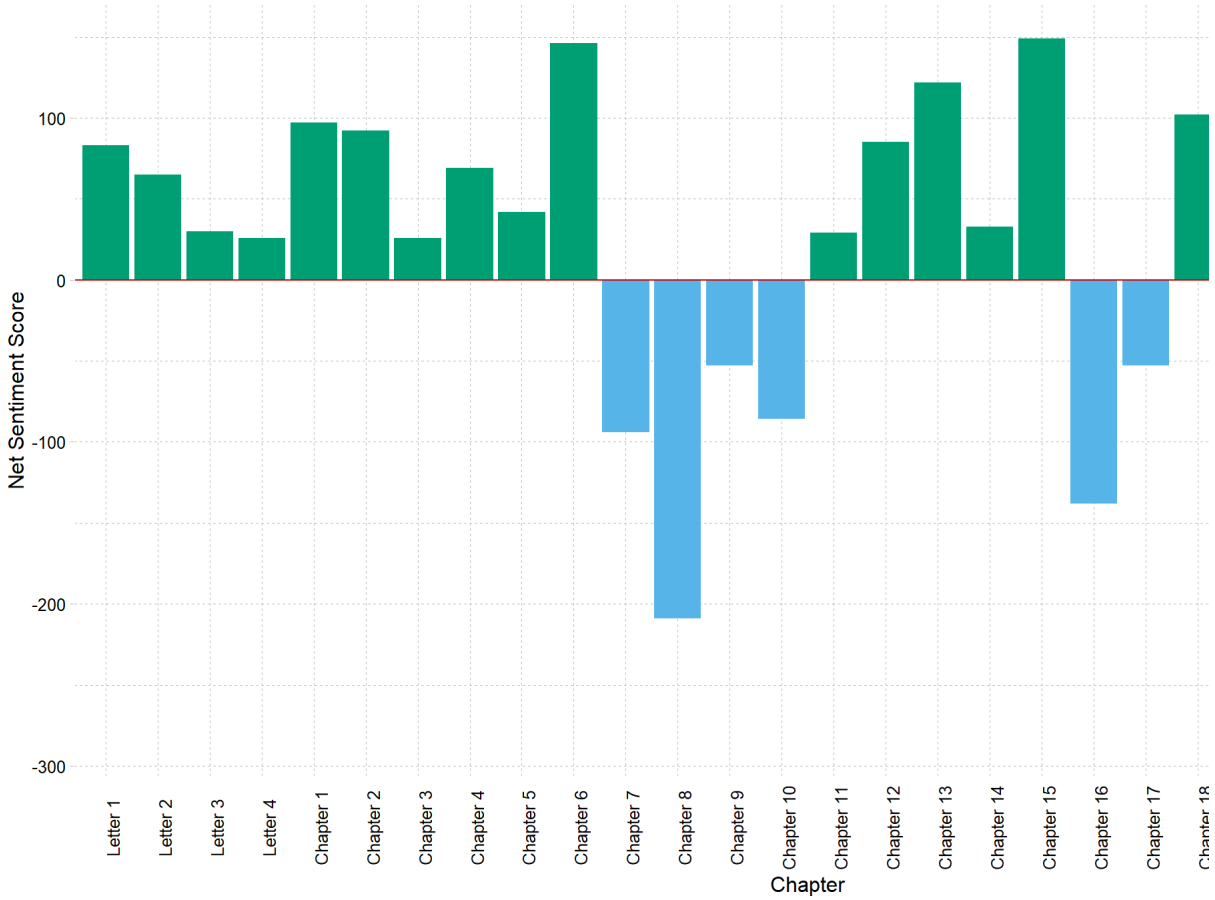

Figure 7.1: Net AFINN Score per Chapter of Frankenstein

To guide people who are yet to read Frankenstein, I renamed all chapters of the novel. I think I did not give away much, but I think we can get a feel of why some chapters are positive while others are negative. Chapter 24's very negative score is also due to it being the chapter with the most words.

```
tibble(chapter_title = c(
  "Letter 1 - Robert Walton",
  "Letter 2 - Vessel and Crew",
  "Letter 3 - In Haste",
  "Letter 4 - Two Strangers on the Ice",
  "Chapter 1 - Alphonse, Caroline, Elizabeth",
  "Chapter 2 - Victor Frankenstein",
  "Chapter 3 - Ingolstadt",
  "Chapter 4 - Eight Feet in Height",
  "Chapter 5 - Daemon",
  "Chapter 6 - News from Home",
  "Chapter 7 - Strangled",
  "Chapter 8 - The Trial",
  "Chapter 9 - Solitude in Nature",
  "Chapter 10 - The Daemon's Proposal",
  "Chapter 11 - Awakening",
  "Chapter 12 - The Cottagers",
  "Chapter 13 - Safie and the Gift of Knowledge",
  "Chapter 14 - Fall of the De Laceys",
  "Chapter 15 - Paradise Lost, Plutarch's Lives and Sorrows of Werter",
  "Chapter 16 - Burned Down",
  "Chapter 17 - The Agreement",
  "Chapter 18 - To England",
  "Chapter 19 - Calm Before the Storm",
  "Chapter 20 - The Threat",
  "Chapter 21 - Accused",
  "Chapter 22 - Overshadowed Marriage",
  "Chapter 23 - Bridal Bier",
  "Chapter 24 - Unconquerable Ice"
)) %>%
  separate(chapter_title,
           into = c("chapter",
                    "title"),
           sep = " - ") %>%
  kable(
    caption = "My Titles for Each Frankenstein Chapter",
    align = c("c", "c")
  ) %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )
```

Table 7.1: My Titles for Each Frankenstein Chapter

chapter	title
Letter 1	Robert Walton
Letter 2	Vessel and Crew
Letter 3	In Haste
Letter 4	Two Strangers on the Ice
Chapter 1	Alphonse, Caroline, Elizabeth
Chapter 2	Victor Frankenstein
Chapter 3	Ingolstadt
Chapter 4	Eight Feet in Height
Chapter 5	Daemon
Chapter 6	News from Home

Chapter 7	Strangled
Chapter 8	The Trial
Chapter 9	Solitude in Nature
Chapter 10	The Daemon's Proposal
Chapter 11	Awakening
Chapter 12	The Cottagers
Chapter 13	Safie and the Gift of Knowledge
Chapter 14	Fall of the De Laceys
Chapter 15	Paradise Lost, Plutarch's Lives and Sorrows of Werter
Chapter 16	Burned Down
Chapter 17	The Agreement
Chapter 18	To England
Chapter 19	Calm Before the Storm
Chapter 20	The Threat
Chapter 21	Accused
Chapter 22	Overshadowed Marriage
Chapter 23	Bridal Bier
Chapter 24	Unconquerable Ice

Above are the titles I made for each chapter. I think this is enough heat for future Frankenstein readers to understand the trend. Better yet, they should read the novel and verify if this sentiment analysis holds for them.

7.2 Down the Precipice

I also started wondering what kind of emotional ride Frankenstein can bring to the reader. Not that I did not feel the emotions, but I want to verify if the data will confirm what I felt.

I took the total AFINN score of a line in the novel and took the cumulative sum of the scores from start to finish. I am interested if there are parts that will keep us feeling down or are there parts that will pull us back and allow us to breath the happiness air.

```
words %>%
  inner_join(get_sentiments("afinn"),
             by = "word") %>%
  group_by(line_number) %>%
  summarise(sentiment_score = sum(score)) %>%
  arrange(line_number) %>%
  mutate(cumsum_sentiment_score = cumsum(sentiment_score)) %>%
  ggplot(aes(line_number,
             cumsum_sentiment_score)) +
  geom_line(color = palette_pander(5)[5]) +
  geom_hline(yintercept = 0,
             color = "red3") +
  theme(axis.text.x = element_blank()) +
  theme_pander() +
  labs(
    x = "Progress of Novel (Start to Finish)",
    y = "Cumulative AFINN Score"
  )
```

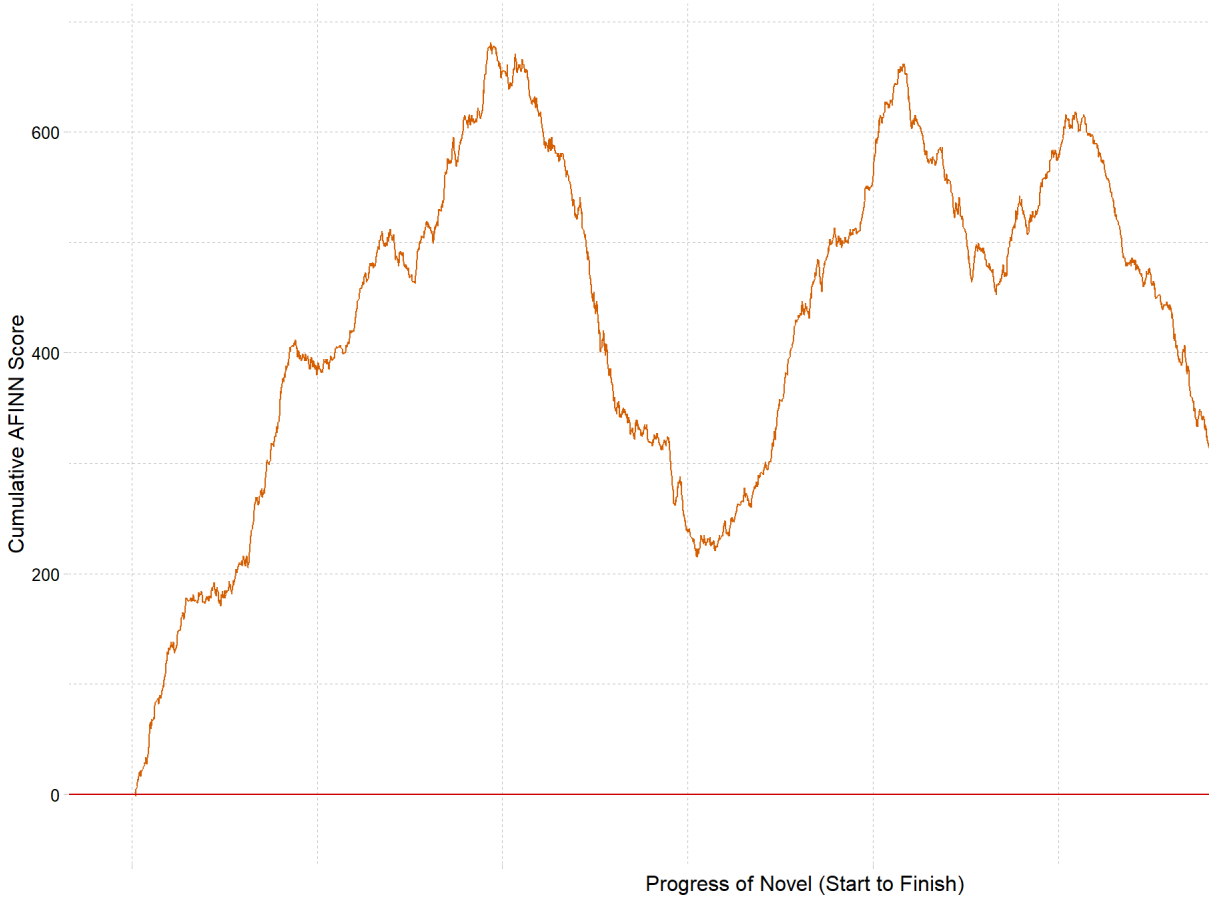

Figure 7.2: Cumulative AFINN Score Through Frankenstein

Overall, Shelley seems to keep the reader floating on the sea of positive emotions but shocks us with a sudden push that sinks our head down into the negative emotions. In the end, we are gasping for breath at all of the negativity. I hope this convinces more readers to try Frankenstein and feel the steep climbs and the sudden drops.

7.3 Fear the Daemon

The NRC lexicon does not score words. Instead, it categorizes words as either positive or negative and into one of the eight basic emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust). What is the most dominant emotion in Frankenstein?

```
words %>%
  inner_join(get_sentiments("nrc"),
             by = "word") %>%
  filter(str_detect(sentiment,
                    "positive|negative") == FALSE) %>%
  count(sentiment) %>%
  mutate(percent_sentiment = n / sum(n)) %>%
  ggplot(aes(fct_reorder(sentiment,
                         percent_sentiment),
             percent_sentiment,
             fill = sentiment)) +
  scale_y_percent() +
  geom_col(show.legend = FALSE) +
  coord_flip() +
```

```
  scale_fill_pander() +
  theme_pander() +
  labs(
    x = "Emotions",
    y = ""
  )
```

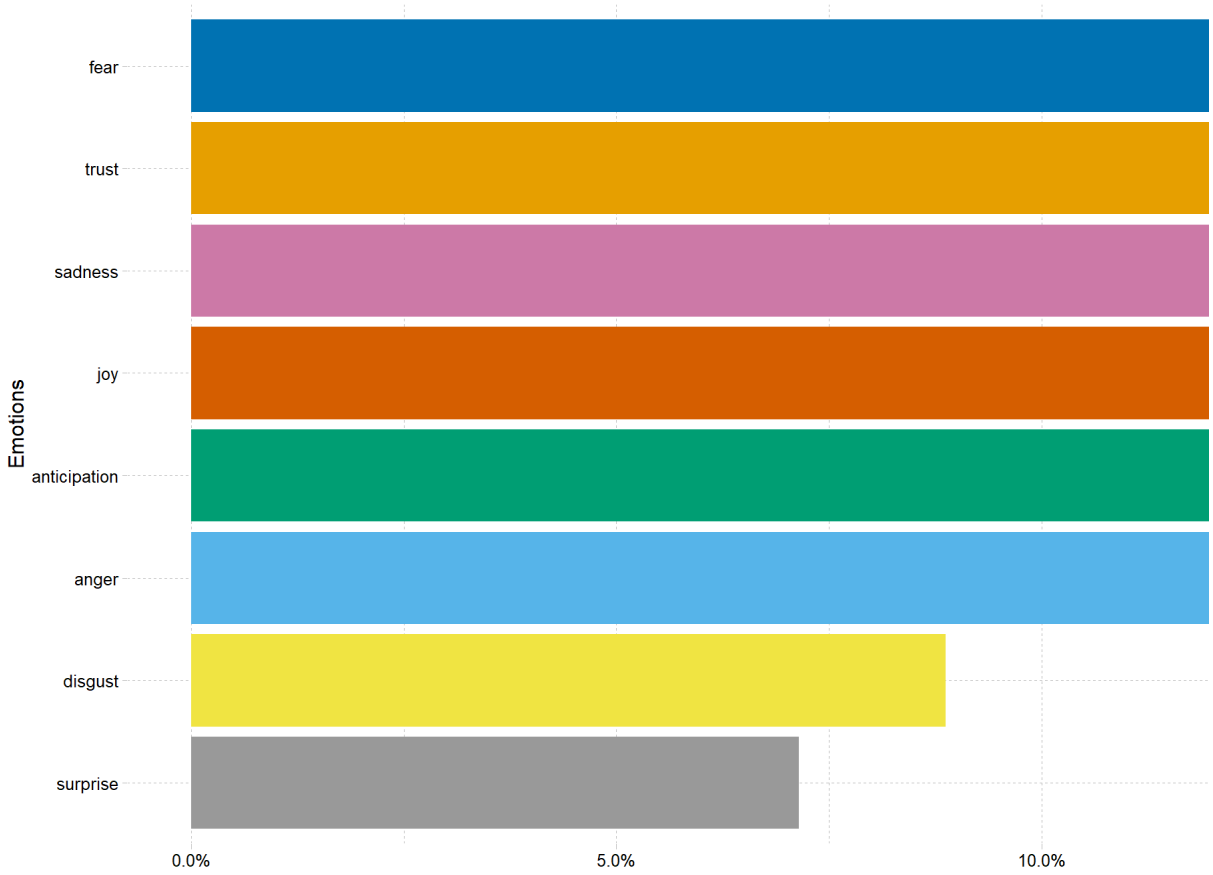

Figure 7.3: Frequency of the Eight Emotions in Frankenstein

Perhaps it is not surprising to the future reader that fear is the dominating emotion in the novel. However, some may wonder why trust comes second and not sadness. Fellow readers will understand why. Trust is also a central theme in the story that could have changed how the story ended at different situations.

8 Ice and Hearts of Fire

> This ice is not made of such stuff as your hearts may be; it is mutable and cannot withstand you if you say that it shall not. — Victor Frankenstein

Alas, this tidytext analysis fails if I have not sparked any interest in you to read Frankenstein. Mary Wollstonecraft Shelley's masterpiece must continue to burn inside the hearts and minds of the people. It might be the only way to caution the Victor and appease the Daemon inside us.

‹ The Crusade: Text Mining Trivium Songs

All About Git and Github in RStudio: A Step-by-Step Guide for Beginners Like Me (With Pictures!) ›