# Positive and Negative Words in Dracula

Ron Richardson

November 20, 2017

# Outline

# Install and Load Libraries

- ```
  library(dplyr)
  ```
  ```

# Install and Load Libraries

- ```
  library(dplyr)
  ```
- ```
  library(gutenbergr)
  ```

# Install and Load Libraries

- `library(dplyr)`
- `library(gutenbergr)`
- `library(tidytext)`

# Install and Load Libraries

- `library(dplyr)`
- `library(gutenbergr)`
- `library(tidytext)`
- `library(stringr)`

# Install and Load Libraries

- `library(dplyr)`
- `library(gutenbergr)`
- `library(tidytext)`
- `library(stringr)`
- `library(ggplot2)`

# Searching for a Novel

```
df<-gutenberg_works(str_detect(title, 'Dracula'))
df$gutenberg_id

## [1]   345 10150

df$title

## [1] "Dracula"          "Dracula's Guest"
```

# Download Dracula from Project Gutenberg

```
dracula<-gutenberg_download(345)
colnames(dracula)

## [1] "gutenberg_id" "text"

substr(dracula$text[500], 1, 21)

## [1] "my own disappointment"
```

# Unpack the Words

```
words<-dracula%>%
  unnest_tokens(word, text)
colnames(words)

## [1] "gutenberg_id" "word"

words[500,]

## # A tibble: 1 x 2
##   gutenberg_id  word
##          <int> <chr>
## 1          345  have
```

# The Bing Lexicon

```
bing<-get_sentiments('bing')
colnames(bing)

## [1] "word"      "sentiment"

bing[500,]

## # A tibble: 1 x 2
##      word sentiment
##     <chr>     <chr>
## 1 bereft  negative
```

# Joining Dracula with Bing

```
words<-inner_join(words, bing)
words$gutenberg_id<-NULL
colnames(words)

## [1] "word"      "sentiment"
```

# Top 10 Positive Words I

```
words_pos<-words%>%
  filter(sentiment=='positive')%>%
  group_by(word)%>%
  summarize(count=n(), sentiment=first(sentiment))%>%
  arrange(count)%>%
  top_n(10, wt=count)
```

# Top 10 Positive Words II

```
words_pos

## # A tibble: 10 x 3
##       word count sentiment
##      <chr> <int>     <chr>
##  1  sweet    66  positive
##  2  ready    71  positive
##  3 better    77  positive
##  4   love    84  positive
##  5  right    99  positive
##  6   work   146  positive
##  7  great   183  positive
##  8   well   245  positive
##  9   good   258  positive
## 10   like   292  positive
```

# Top 10 Negative Words I

```
words_neg<-words%>%
  filter(sentiment=='negative')%>%
  group_by(word)%>%
  summarize(count=n(), sentiment=first(sentiment))%>%
  arrange(count)%>%
  top_n(10, wt=count)
```

# Top 10 Negative Words II

```
words_neg

## # A tibble: 10 x 3
##         word count sentiment
##        <chr> <int>     <chr>
##  1  trouble    53  negative
##  2     fell    59  negative
##  3     miss    60  negative
##  4     dark    77  negative
##  5  strange    90  negative
##  6    death    94  negative
##  7 terrible   100  negative
##  8     dead   109  negative
##  9     fear   137  negative
## 10     poor   193  negative
```

# Joining Positive and Negative Words I

```
words_pos$word<-factor(words_pos$word, levels=words_pos$wo
words_neg$word<-factor(words_neg$word, levels=words_neg$wo

words<-rbind(words_pos, words_neg)
```

# Joining Positive and Negative Words II

```
words

## # A tibble: 20 x 3
##         word count sentiment
##       <fctr> <int>     <chr>
##  1    sweet    66  positive
##  2    ready    71  positive
##  3   better    77  positive
##  4     love    84  positive
##  5    right    99  positive
##  6     work   146  positive
##  7    great   183  positive
##  8     well   245  positive
##  9     good   258  positive
## 10     like   292  positive
## 11  trouble    53  negative
## 12     fell    59  negative
## 13     miss    60  negative
```

# The Comparison Barplot I

```
plot<-ggplot()+
  geom_bar(data=words, aes(x=word, y=count, fill=sentiment,
  xlab("Word")+
  ylab("Count")+
  coord_flip()+
  ggtitle("Top 10 Positive/Negative Words in Dracula")+
  facet_wrap(~sentiment, scales='free_y')+  # which column
  scale_fill_manual(values=c('#000000', '#ea6205'))+
  scale_color_manual(values=c('#ea6205', '#000000'))
```

# The Comparison Barplot II