Text analysis

Getting and Cleaning Data

Unstructured Data Types

Text files and documents

Websites and applications

Sensor data

Image files

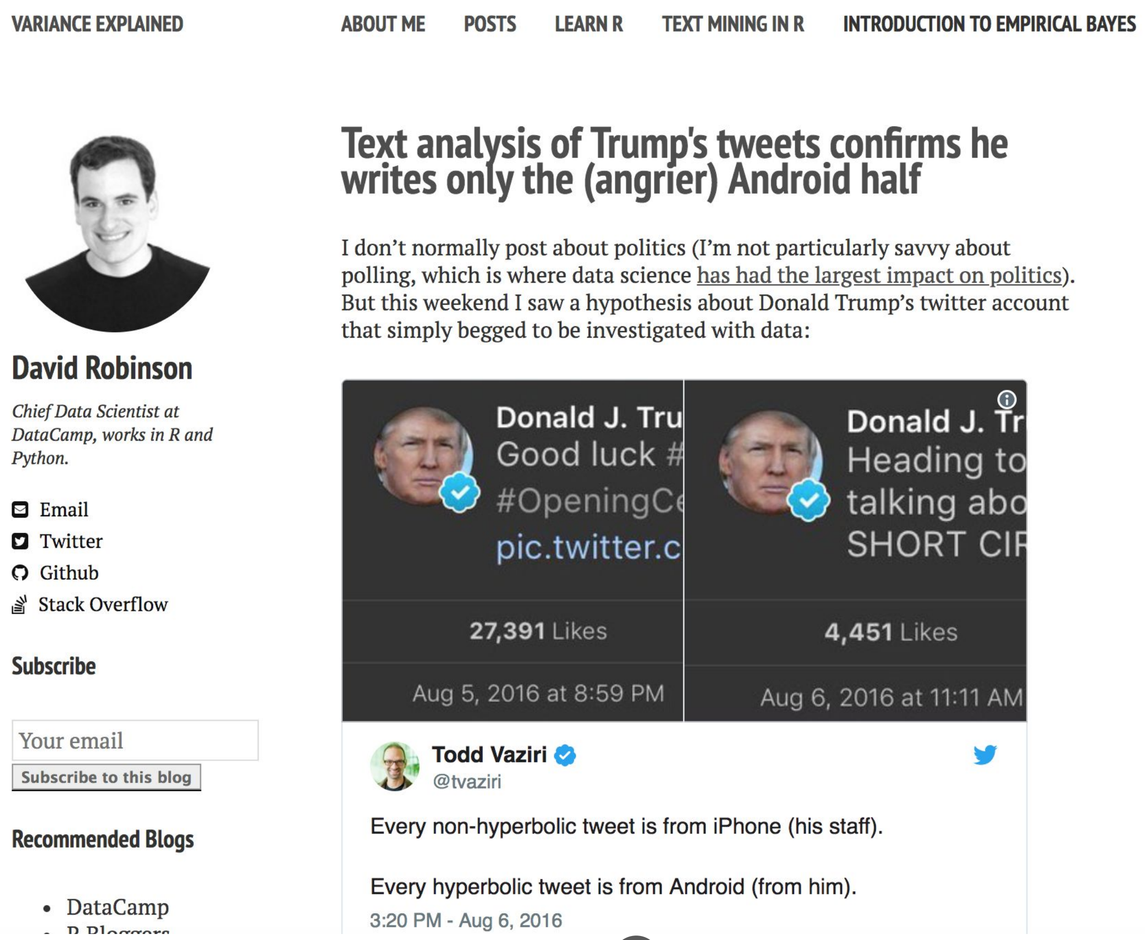
Audio files

Video files

Email data

Social media data

Source: TechTarget



VARIANCE EXPLAINED

ABOUT ME POSTS LEARN R TEXT MINING IN R INTRODUCTION TO EMPIRICAL BAYES

David Robinson

Chief Data Scientist at DataCamp, works in R and Python.

- Email
- Twitter
- Github
- Stack Overflow

Subscribe

Your email

Subscribe to this blog

Recommended Blogs

- DataCamp
- R Bloggers

Text analysis of Trump's tweets confirms he writes only the (angrier) Android half

I don't normally post about politics (I'm not particularly savvy about polling, which is where data science has had the largest impact on politics). But this weekend I saw a hypothesis about Donald Trump's twitter account that simply begged to be investigated with data:

Donald J. Tru
Good luck #
#OpeningCe
pic.twitter.c

27,391 Likes

Aug 5, 2016 at 8:59 PM

Donald J. Tr
Heading to
talking abo
SHORT CIR

4,451 Likes

Aug 6, 2016 at 11:11 AM

Todd Vaziri
@tvaziri

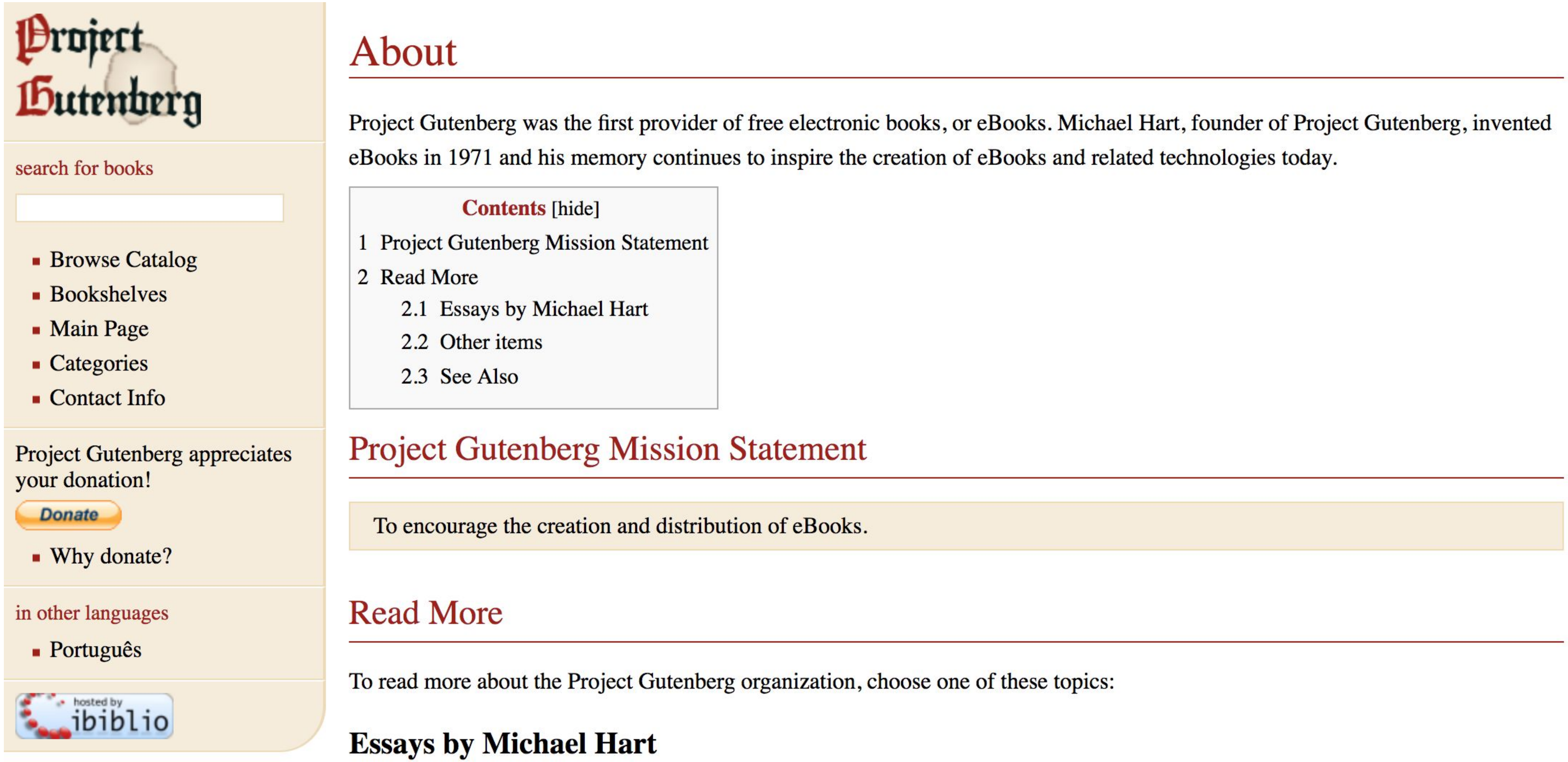
Every non-hyperbolic tweet is from iPhone (his staff).

Every hyperbolic tweet is from Android (from him).

3:20 PM - Aug 6, 2016

http://varianceexplained.org/r/trump-tweets/

TIDYTEXT



About

Project Gutenberg was the first provider of free electronic books, or eBooks. Michael Hart, founder of Project Gutenberg, invented eBooks in 1971 and his memory continues to inspire the creation of eBooks and related technologies today.

Contents [hide]

Project Gutenberg Mission Statement

To encourage the creation and distribution of eBooks.

Read More

To read more about the Project Gutenberg organization, choose one of these topics:

Essays by Michael Hart

https://www.gutenberg.org/

```r
gutenberg_works() %>%
  filter(title == "Dracula")
```

```
# A tibble: 1 x 8
  gutenberg_id title   author     gutenberg_author_… language gutenberg_bookshelf                rights             has_text
         <int> <chr>   <chr>                   <int> <chr>    <chr>                              <chr>              <lgl>
1          345 Dracula Stoker, B…                190 en       Gothic Fiction/Movie Books/Horror/My… Public domain in … TRUE
```

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

```
# A tibble: 15,568 x 2
   gutenberg_id text
          <int> <chr>
 1          345 "                                 DRACULA"
 2          345 ""
 3          345 ""
 4          345 ""
 5          345 ""
 6          345 ""
 7          345 "                                 DRACULA"
 8          345 ""
 9          345 "                                   _by_"
10          345 ""
# ... with 15,558 more rows
```

```
dracula %>%
  unnest_tokens(word, text)

# A tibble: 162,577 x 2
   gutenberg_id word
          <int> <chr>
 1          345 dracula
 2          345 dracula
 3          345 _by_
 4          345 bram
 5          345 stoker
 6          345 illustration
 7          345 colophon
 8          345 new
 9          345 york
10          345 grosset
# ... with 162,567 more rows
```

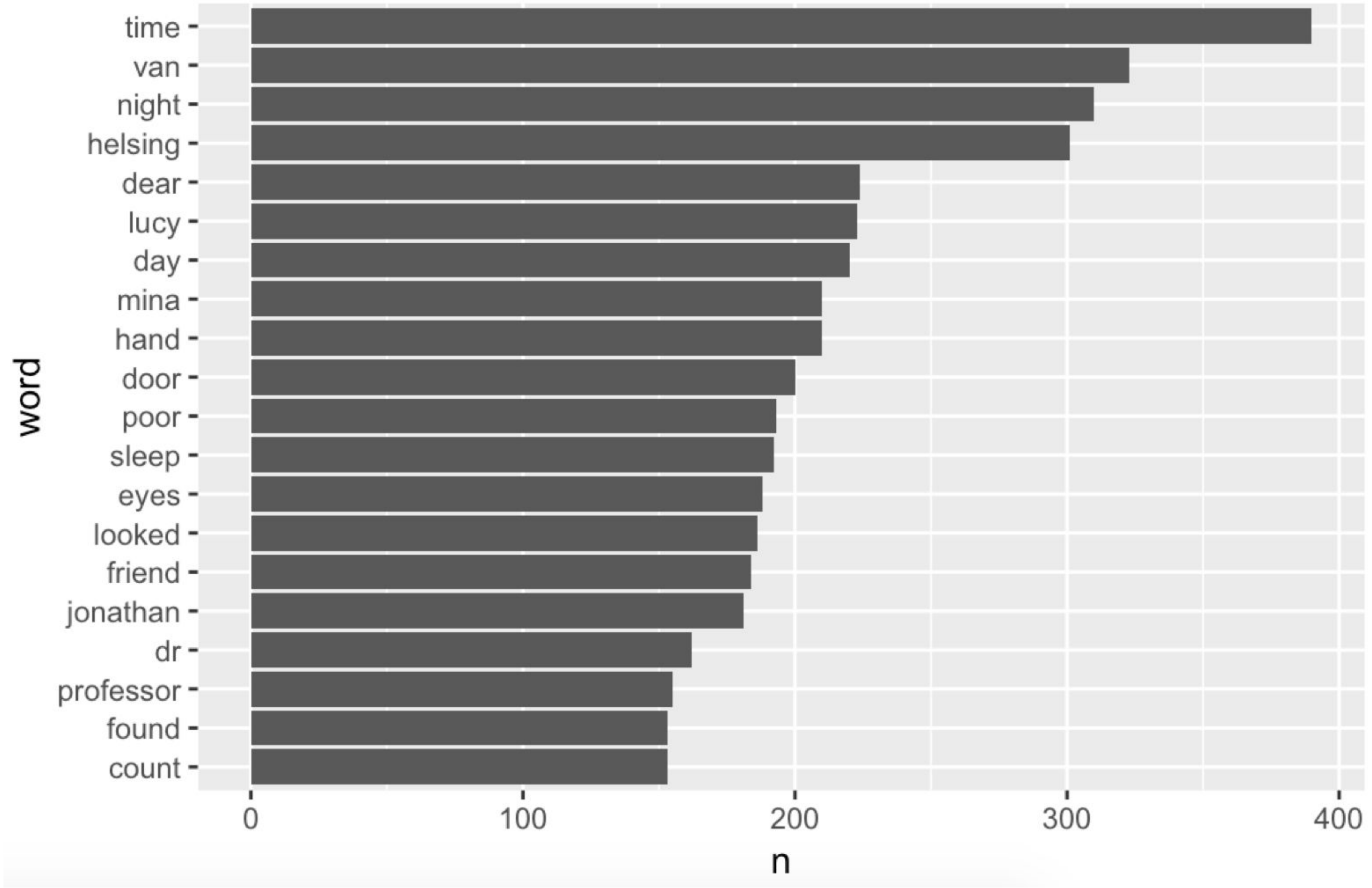
```
> stop_words
# A tibble: 1,149 x 2
   word        lexicon
   <chr>       <chr>
 1 a           SMART
 2 a's         SMART
 3 able        SMART
 4 about       SMART
 5 above       SMART
 6 according   SMART
 7 accordingly SMART
 8 across      SMART
 9 actually    SMART
10 after       SMART
# ... with 1,139 more rows
```

```
dracula %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

```
# A tibble: 48,552 x 2
   gutenberg_id word
          <int> <chr>
 1          345 dracula
 2          345 dracula
 3          345 _by_
 4          345 bram
 5          345 stoker
 6          345 illustration
 7          345 colophon
 8          345 york
 9          345 grosset
10          345 dunlap
# ... with 48,542 more rows
```

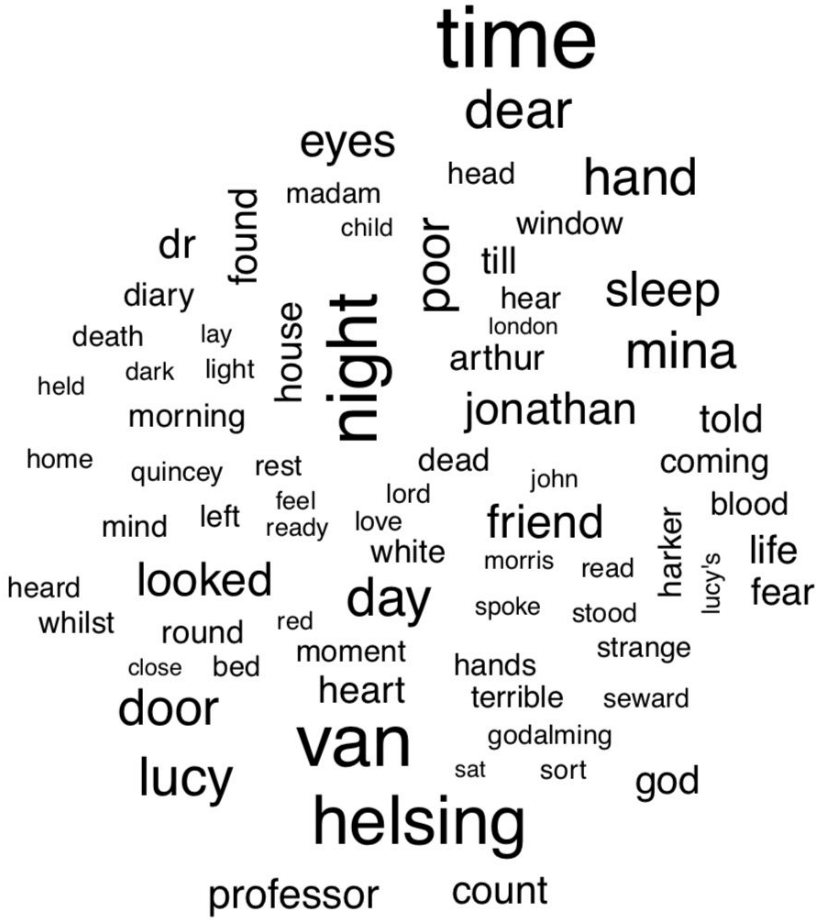
```
dracula %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)

# A tibble: 9,072 x 2
   word        n
   <chr>   <int>
 1 time      390
 2 van       323
 3 night     310
 4 helsing   301
 5 dear      224
 6 lucy      223
 7 day       220
 8 hand      210
 9 mina      210
10 door      200
# ... with 9,062 more rows
```

```r
dracula %>%
  unnest_tokens(word, text) %>%          # Put each word on its own line
  anti_join(stop_words) %>%              # Remove common "stop" words
  count(word, sort = TRUE) %>%           # Count the number of times each word appears
  filter(n > 150) %>%                    # Keep only those that appear more than 150 times
  mutate(word = reorder(word, n)) %>%    # Put them in the order they appear
  ggplot(aes(word, n)) +                 # Plot the number of times each word appears
  geom_bar(stat = "identity") +          # Using a bar plot
  coord_flip                             # Flip the axes so that the words are on the Y
```



time
van
night
helsing
dear
lucy
day
mina
hand
door
poor
sleep
eyes
looked
friend
jonathan
dr
professor
found
count
word
0
100
200
300
400
n

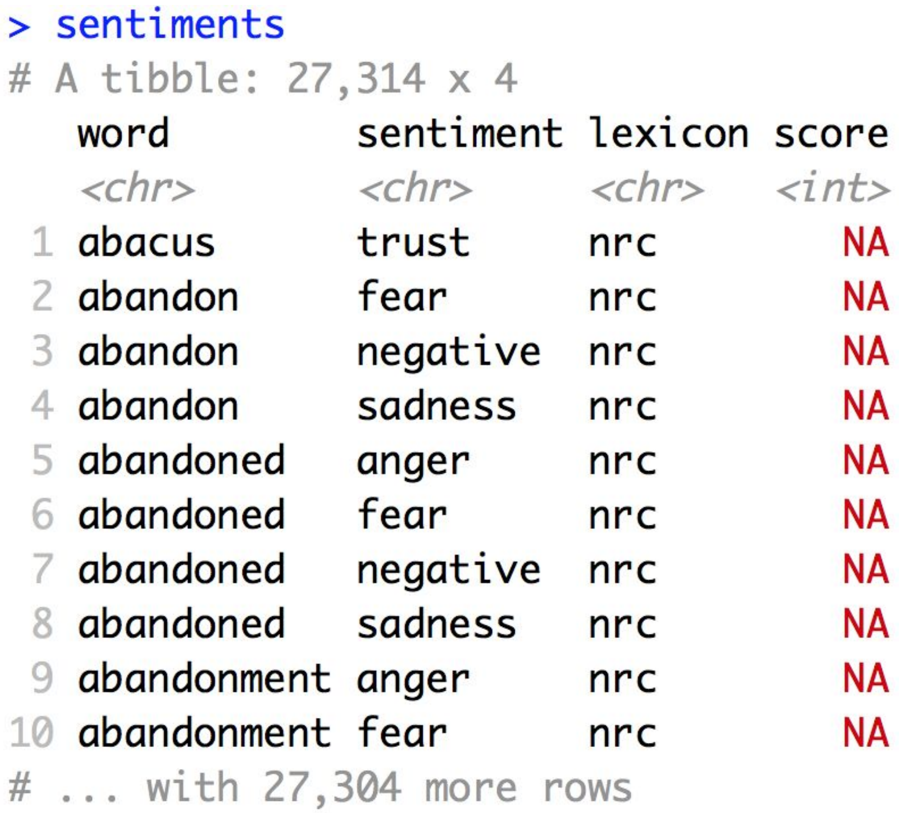
```r
dracula %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 75))
```


time
dear
eyes
head
hand
madam
found
child
poor
window
dr
till
diary
night
hear
sleep
house
london
death
lay
arthur
mina
dark
light
held
jonathan
told
morning
home
quincey
rest
dead
coming
john
lord
feel
blood
harker
mind
left
ready
love
friend
white
life
morris
read
lucy's
heard
looked
day
fear
spoke
stood
whilst
red
round
moment
strange
close
bed
hands
heart
terrible
seward
door
godalming
van
lucy
sat
sort
god
helsing
professor
count

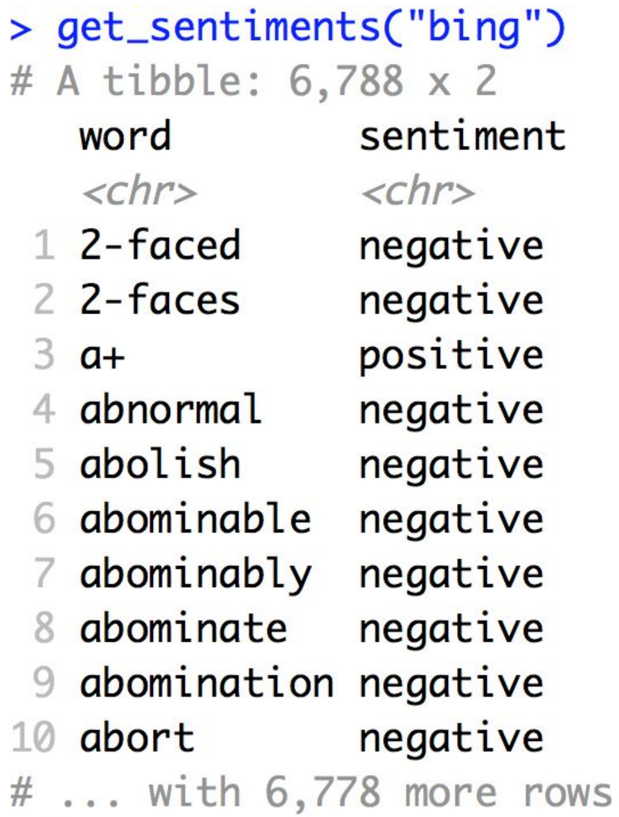
Positive:
70%

Negative:
20%

Neutral:
10%



```
> sentiments
# A tibble: 27,314 x 4
   word        sentiment lexicon score
   <chr>       <chr>     <chr>   <int>
 1 abacus      trust     nrc        NA
 2 abandon     fear      nrc        NA
 3 abandon     negative  nrc        NA
 4 abandon     sadness   nrc        NA
 5 abandoned   anger     nrc        NA
 6 abandoned   fear      nrc        NA
 7 abandoned   negative  nrc        NA
 8 abandoned   sadness   nrc        NA
 9 abandonment anger     nrc        NA
10 abandonment fear      nrc        NA
# ... with 27,304 more rows
```

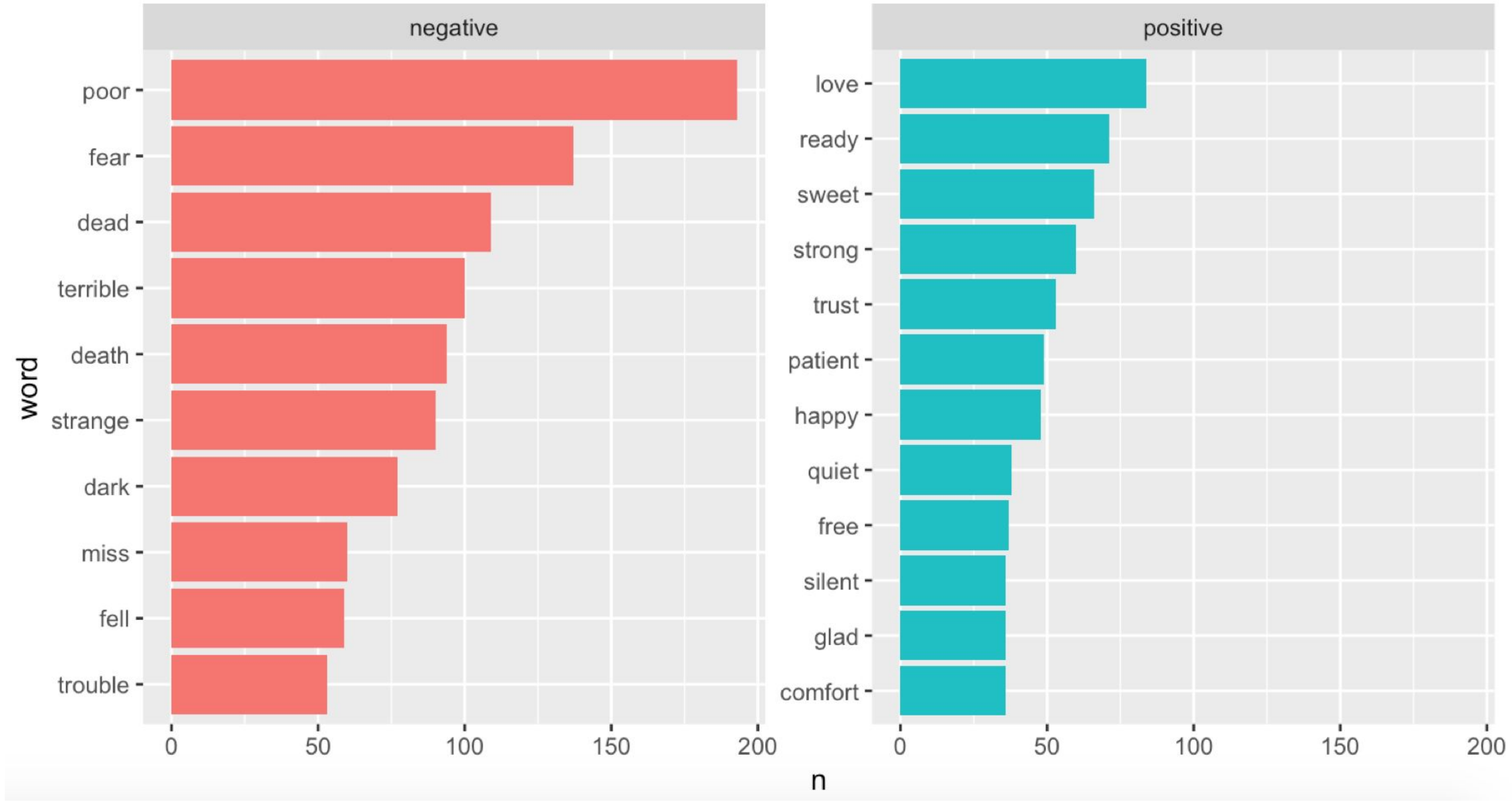


```
> get_sentiments("bing")
# A tibble: 6,788 x 2
   word        sentiment
   <chr>       <chr>
 1 2-faced     negative
 2 2-faces     negative
 3 a+          positive
 4 abnormal    negative
 5 abolish     negative
 6 abominable  negative
 7 abominably  negative
 8 abominate   negative
 9 abomination negative
10 abort       negative
# ... with 6,778 more rows
```

```
dracula %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE)
```

```
# A tibble: 1,611 x 3
   word     sentiment     n
   <chr>    <chr>     <int>
 1 poor     negative    193
 2 fear     negative    137
 3 dead     negative    109
 4 terrible negative    100
 5 death    negative     94
 6 strange  negative     90
 7 love     positive     84
 8 dark     negative     77
 9 ready    positive     71
10 sweet    positive     66
# ... with 1,601 more rows
```

```r
dracula %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment) %>%
  group_by(sentiment) %>%                         # Group the words into positive and negative groups
  top_n(10) %>%                                   # Find the top ten most common words in both the positive and negative groups
  ungroup() %>%                                   # Ungroup the data so that mutate() works in the next step
  mutate(word = reorder(word, n)) %>%             # Reorder the words so that when you plot them it will be in order of most common to least
  ggplot(aes(word, n, fill = sentiment)) +        # Plot the words and their frequencies
  geom_bar(stat = "identity") +                   # In a bar plot
  facet_wrap(~sentiment, scales = "free_y") +     # In two separate plots, one for each sentiment
  coord_flip()                                    # With the X and Y axes flipped for readability
```


negative
positive
word
poor
fear
dead
terrible
death
strange
dark
miss
fell
trouble
love
ready
sweet
strong
trust
patient
happy
quiet
free
silent
glad
comfort
0
50
100
150
200
n

Summarizing: Text analysis

Getting and cleaning data