correlaid_text_analysis

November 3, 2018

Scraping webpages

```
url1 <- "https://correlaid.org/blog/posts/understand-p-values"
url2 <- "https://correlaid.org/blog/posts/blockchain-explained"
url3 <- "https://correlaid.org/blog/posts/music-with-r"

doc1 <- gettxt(url1)
# doc11 <- getURL(url1)
# dic11 <- htmlTreeParse(doc11)
# doc1111 <- readLines(url1)
doc2 <- gettxt(url2)
doc3 <- gettxt(url3)

raw.text <- as.data.frame(rbind(cbind("doc1",doc1),
                                cbind("doc2",doc2),
                                cbind("doc3",doc3)))
colnames(raw.text)  <-  c("article", "text")
raw.text$article <- as.character(raw.text$article)
raw.text$text <- as.character(raw.text$text)
str(raw.text)
```

```
## 'data.frame':    3 obs. of  2 variables:
##  $ article: chr  "doc1" "doc2" "doc3"
##  $ text   : chr  "Skip to main content\n\nToggle navigation Menu\n\n•\n• Zurück zu Correlaid.org\n• B
```

tidy text

```
tidy_webtext <- raw.text %>%
  # unnest_tokens(sentence, text, token = "sentences") %>%
  unnest_tokens(word, text)%>%
  mutate(word = str_extract(word , "[a-z'\\s]+")) %>%
  anti_join(stop_words, by= "word") %>%
  filter(!word %in% "[\\s]+",
         !word %in% "",
         !word %in% NA)
```

Chapter 1

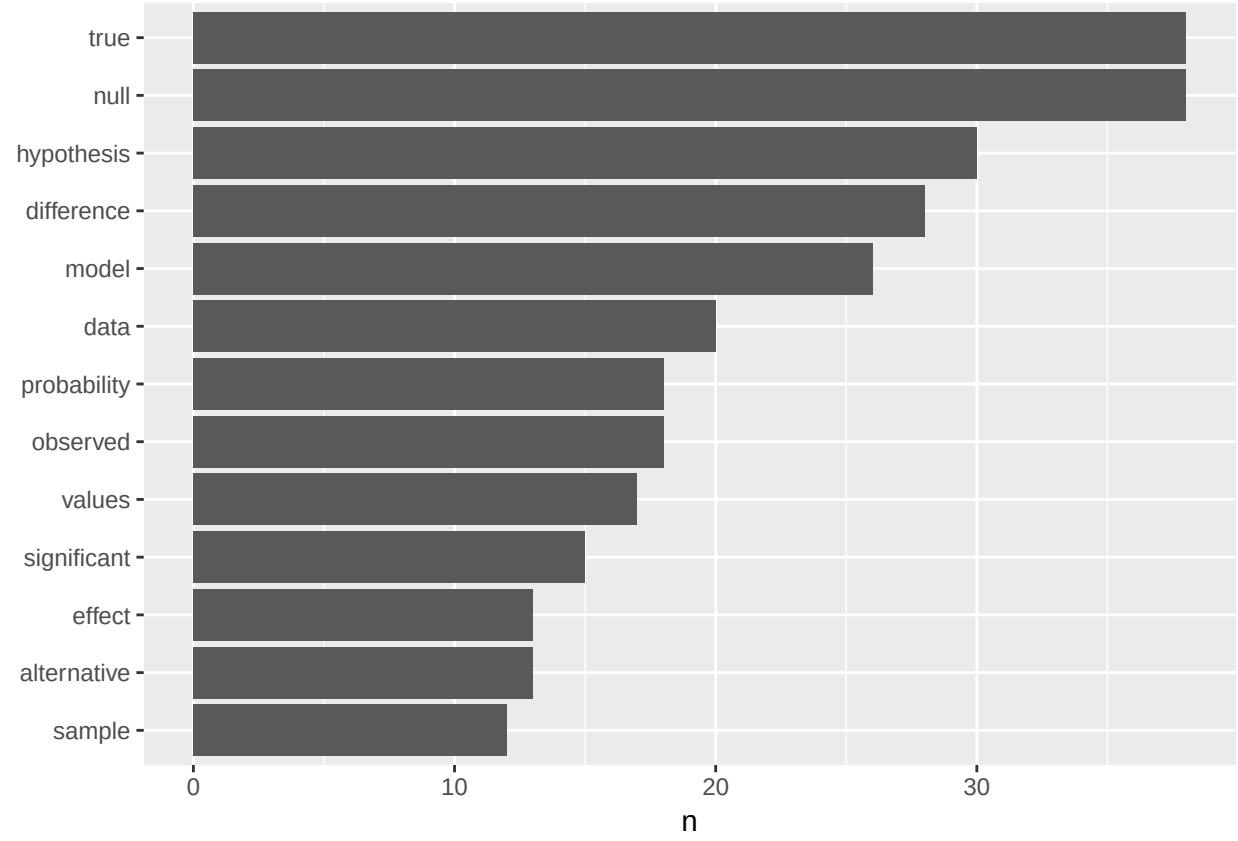
look at single word frequency and visualize

first at doc1

```
text1_count <- tidy_webtext %>%
  filter(article == "doc1" )%>%
```

```
  count(word, sort = TRUE)

text1_count %>%
  filter(n > 10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```

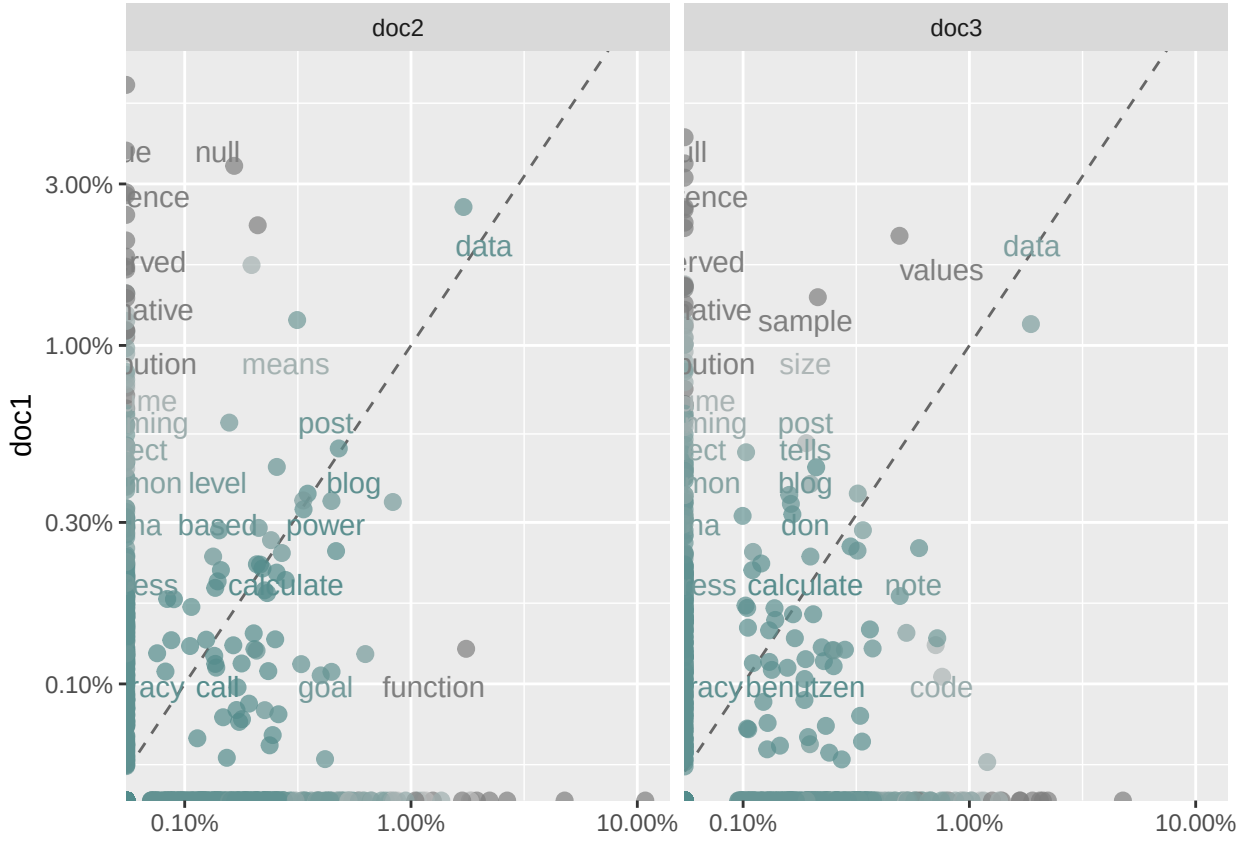


plotting and comparing the three articles

```
frequency <- tidy_webtext %>%
  count(article, word) %>%
  group_by(article) %>%
  mutate(proportion = n/sum(n) )%>%
  select(-n) %>%
  spread(article, proportion) %>%
  gather(article, proportion, 'doc2' : 'doc3')
#fill all NA (word proporton) with zero
frequency[is.na(frequency)] <- 0


ggplot(frequency, aes(x = proportion, y = `doc1`, color = abs(`doc1` - proportion))) +
```

```
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = .7, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  scale_color_gradient(limits = c(0, 0.01), low = "darkslategray4", high = "gray75") +
  facet_wrap(~ article, ncol = 2) +
  theme(legend.position="none") +
  labs(y = "doc1", x = NULL)
```




```
cor.test(data = frequency[frequency$article == "doc2",],
         ~ proportion + `doc1`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and doc1
## t = -0.90635, df = 800, p-value = 0.365
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.10103135  0.03728257
## sample estimates:
##         cor
## -0.03202772
```

```
cor.test(data = frequency[frequency$article == "doc3",],
         ~ proportion + `doc1`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and doc1
## t = -1.3419, df = 800, p-value = 0.18
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.11623528  0.02191054
## sample estimates:
##         cor
## -0.04738898
```

As we saw in the plots, the word frequencies have little frequencies in three articles.

Chapter 2 Sentiment analysis

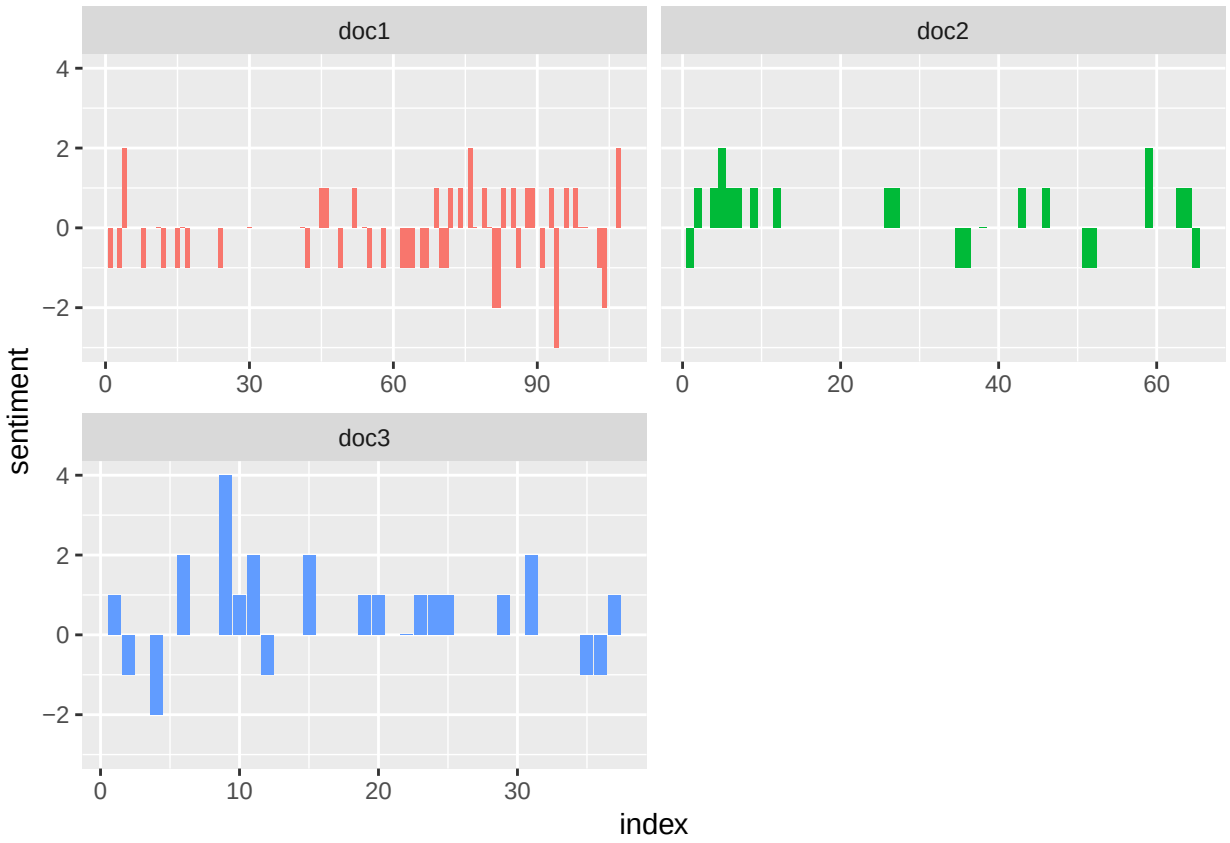
```
afinn <- get_sentiments("afinn")
bing <- get_sentiments("bing")
nrc <- get_sentiments("nrc")
```

bing sentiment analysis

```
sentitext <-  raw.text %>%
  unnest_tokens(sentence, text, token = "sentences") %>%
  group_by(article) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, sentence) %>%
  anti_join(stop_words, by = "word")

bing_analysis <- sentitext %>%
  inner_join(bing, by = "word") %>%
  count(article, index = linenumber , sentiment)%>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)

ggplot(bing_analysis, aes(index, sentiment, fill = article)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ article, ncol = 2, scales = "free_x")
```



Comparing the three sentiment dictionaries

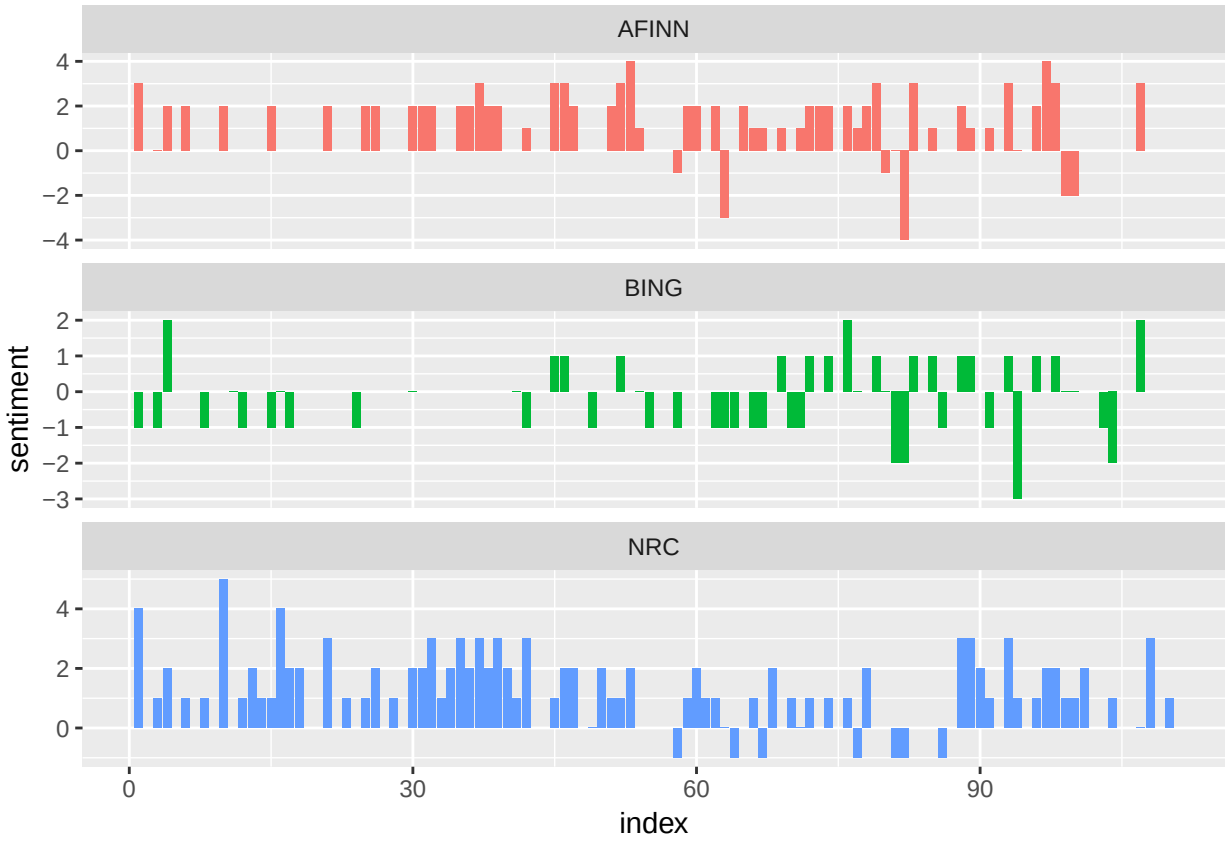
```
sentidoc1 <-  sentitext %>%
  filter(article == "doc1")

afinnsenti <- sentidoc1 %>%
  inner_join(afinn, by = "word") %>%
  group_by(index = linenumber ) %>%
  summarise(sentiment = sum(score)) %>%
  mutate(method = "AFINN")

bingnrcsenti <- bind_rows(sentidoc1 %>%
                            inner_join(bing) %>%
                            mutate(method = "BING"),
                          sentidoc1 %>%
                            inner_join(nrc %>%
                                         filter(sentiment %in% c("positive",
                                                                 "negative"))) %>%
                            mutate(method = "NRC")) %>%
  count(method, index = linenumber, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)

bind_rows(afinnsenti,
```

```
          bingnrcsenti) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



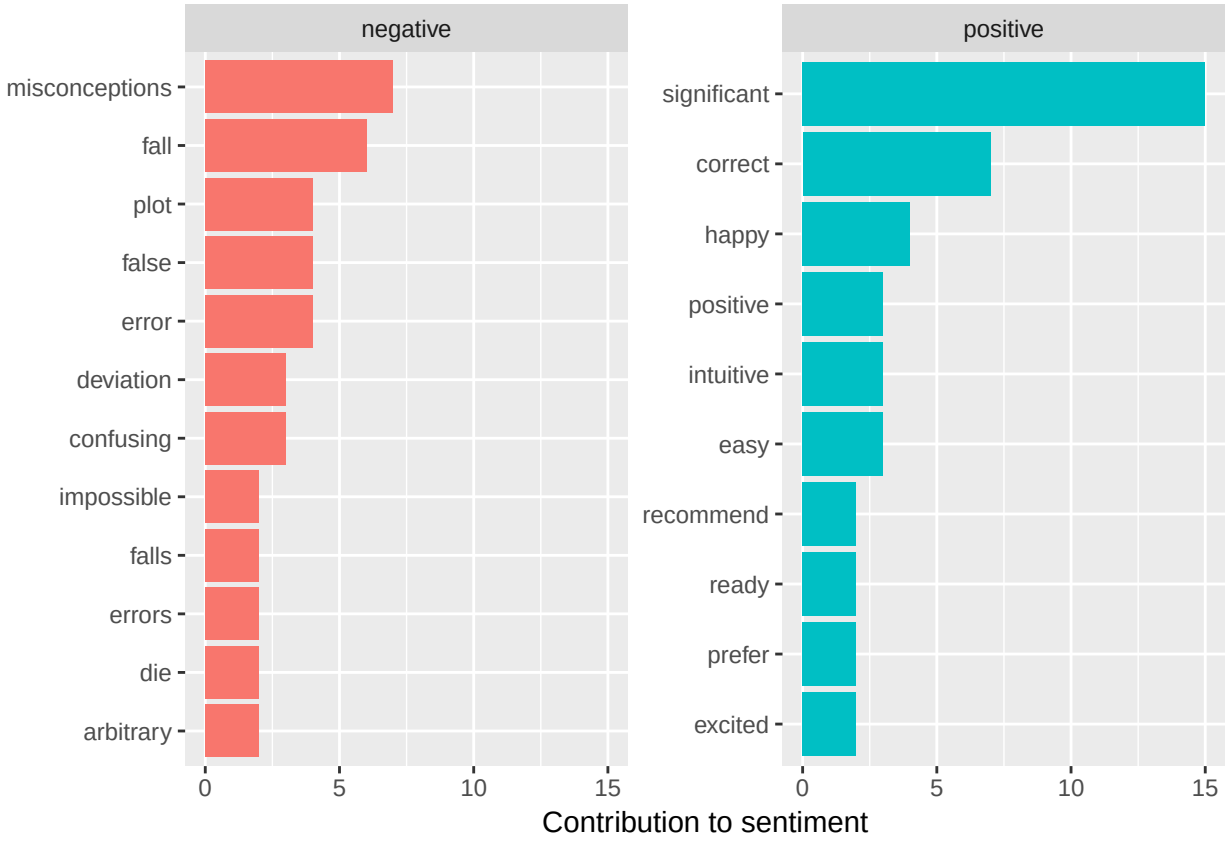
Most common sentiment words

```
bing_counts <- sentitext %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
bing_counts
```

```
## # A tibble: 79 x 3
##    word           sentiment     n
##    <chr>          <chr>     <int>
##  1 significant    positive     15
##  2 correct        positive      7
##  3 misconceptions negative      7
##  4 fall           negative      6
##  5 error          negative      4
##  6 false          negative      4
##  7 happy          positive      4
##  8 plot           negative      4
```

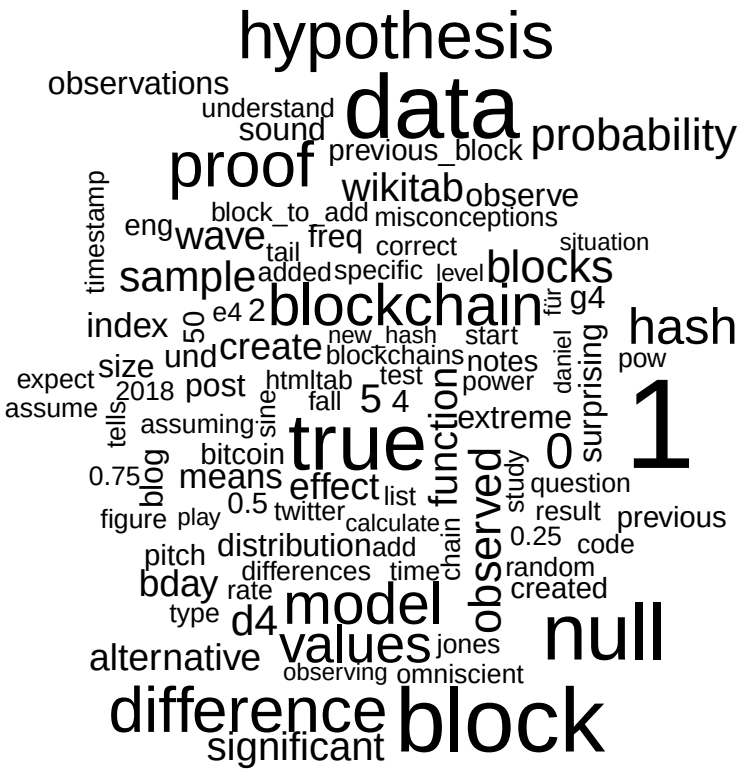
```
##  9 confusing      negative      3
## 10 deviation      negative      3
## # ... with 69 more rows
```

```
bing_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL) +
  coord_flip()
```



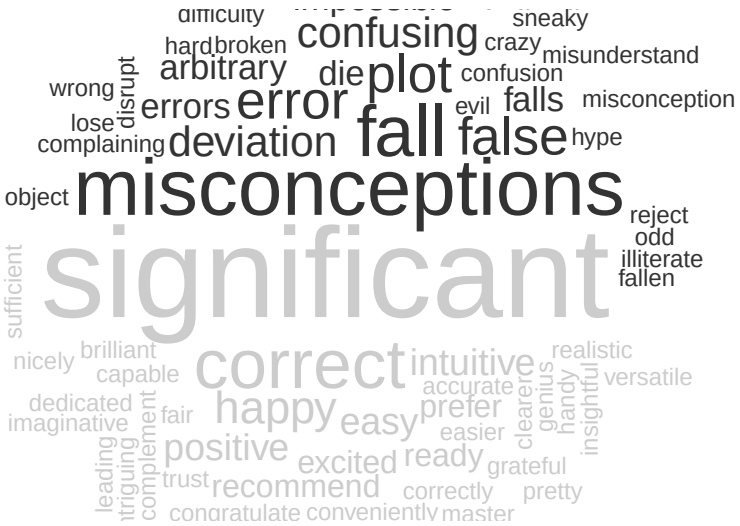
Wordclouds

```
sentitext %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```




```
library(reshape2)

sentitext %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"),
                   max.words = 100)
```


difficulty
sneaky
confusing
crazy
hard
broken
misunderstand
arbitrary
die
plot
confusion
wrong
disrupt
errors
error
evil
falls
misconception
lose
fall
false
complaining
deviation
hype
object
misconceptions
reject
odd
illiterate
fallen
sufficient
significant
nicely
brilliant
correct
intuitive
realistic
capable
accurate
versatile
dedicated
happy
prefer
genius
handy
insightful
imaginative
fair
easy
easier
clearer
positive
leading
intriguing
complement
excited
ready
grateful
trust
recommend
correctly
pretty
congratulate
conveniently
master