Assignment 8

Samriddh Gupta

Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'readr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## Warning: package 'stringr' was built under R version 3.6.3

## Warning: package 'forcats' was built under R version 3.6.3

## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 3.6.3
```

```
library(janeaustenr)
```

```
## Warning: package 'janeaustenr' was built under R version 3.6.3
```

```
library(stringr)
library(gutenbergr)
```

```
## Warning: package 'gutenbergr' was built under R version 3.6.3
```

```
library(ggplot2)
library(dplyr)
library(scales)
```

```
## Warning: package 'scales' was built under R version 3.6.3
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

Exercise 1:

Getting all the books

```
hgwells<-gutenberg_works(author == "Wells, H. G. (Herbert George)")
hgwells1<-gutenberg_download(hgwells$gutenberg_id)
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

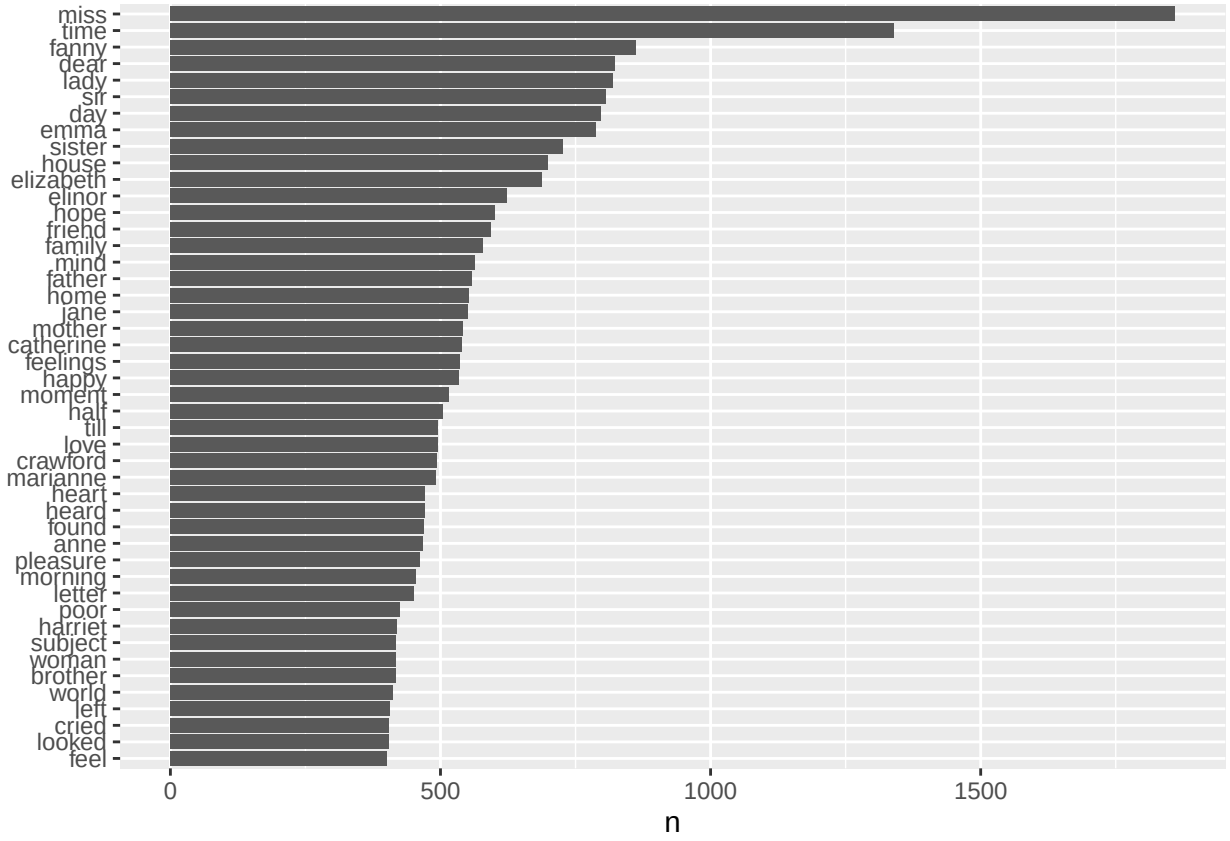
```
bronte<-gutenberg_works(str_detect(author, "Brontë,"))
bronte1<-gutenberg_download(bronte$gutenberg_id)

orig_books <- austen_books() %>%
  group_by(book) %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(str_detect(text,
                                     regex("^chapter [\\divxlc]",
                                           ignore_case = TRUE)))) %>%
  ungroup() %>%
  select(chapter, linenumber, everything())
```

Getting word frequncy for each author

```
tidy_books <- orig_books %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidy_books %>%
  count(word, sort = TRUE) %>%
  filter(n > 400) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```




```
tidy_hgwells <- hgwells1 %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidy_bronte <- bronte1 %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

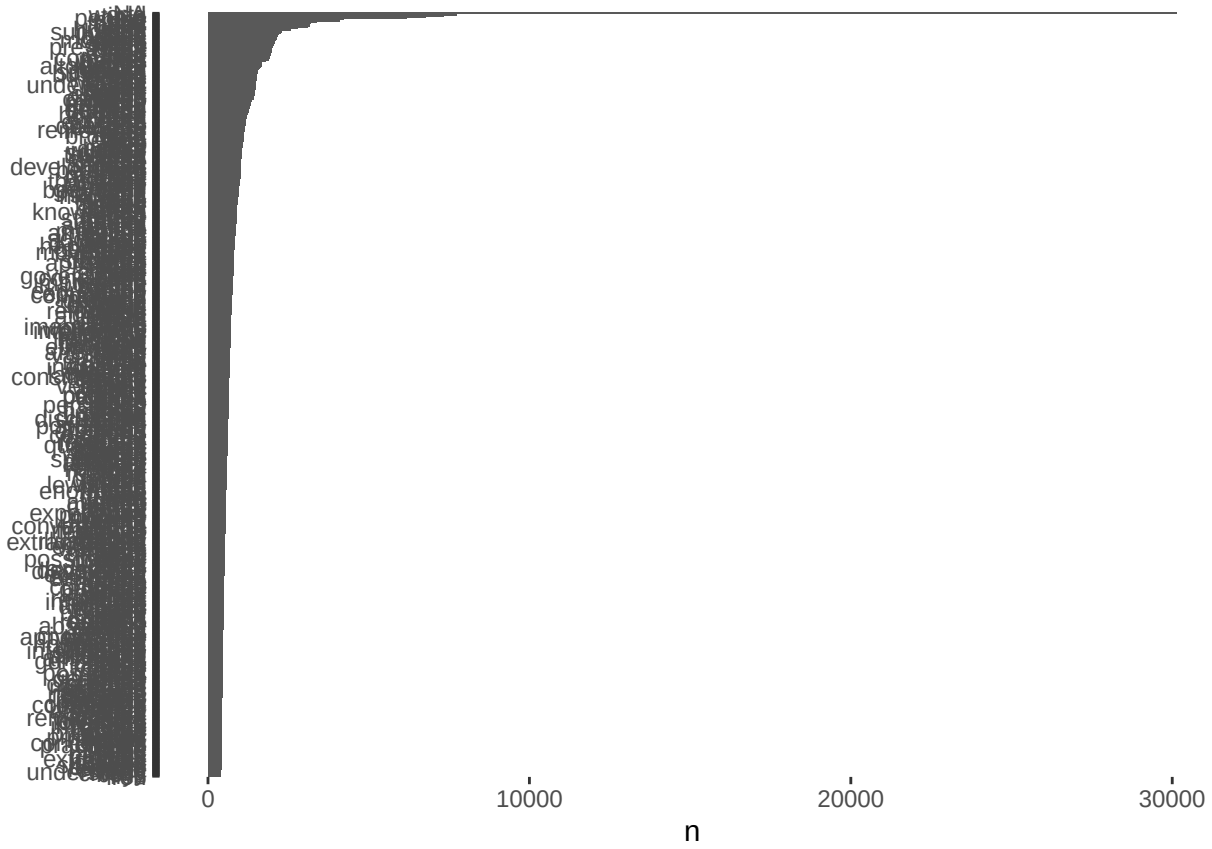
```
tidy_hgwells %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 54,988 x 2
##    word       n
##    <chr>  <int>
##  1 <NA>   30146
##  2 world   7728
##  3 time    7327
##  4 people  6758
##  5 life    6186
##  6 mind    4212
##  7 day     4107
##  8 sort    3385
##  9 hand    3197
## 10 found   3172
## # ... with 54,978 more rows
```

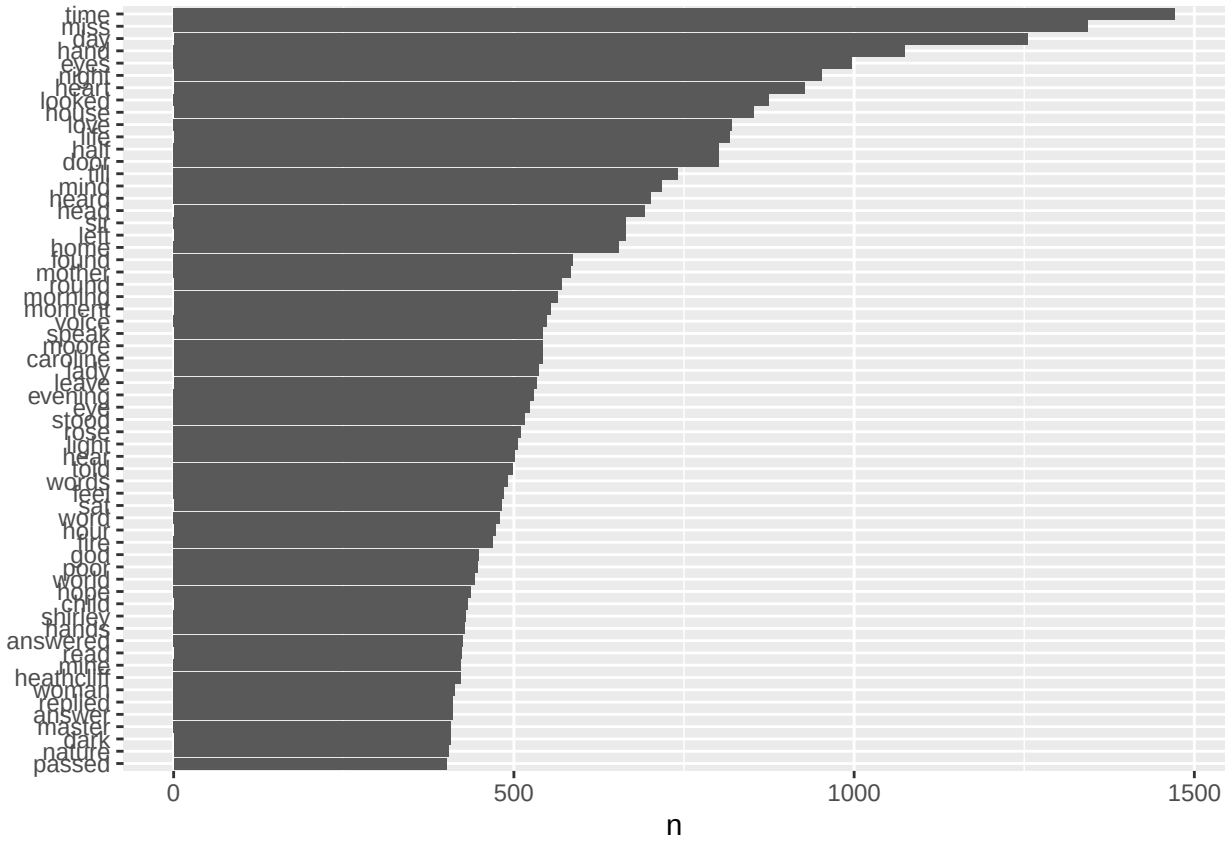
```
tidy_bronte %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 26,864 x 2
##    word       n
##    <chr>  <int>
##  1 time    1472
##  2 miss    1344
##  3 day     1255
##  4 hand    1075
##  5 eyes     997
##  6 night    952
##  7 heart    927
##  8 looked   875
##  9 house    852
## 10 love     821
## # ... with 26,854 more rows
```

```
tidy_hgwells %>%
  count(word, sort = TRUE) %>%
  filter(n > 400) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```




```
tidy_bronte %>%
  count(word, sort = TRUE) %>%
  filter(n > 400) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```



comparing the frequnency

```
frequency_by_word_across_authors <- bind_rows(
  mutate(tidy_bronte,author = "Bronte"),
  mutate(tidy_hgwells, author = "Wells"),
  mutate(tidy_books, author = "Austen")) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(author, proportion)

frequency_by_word_across_authors
```

```
## # A tibble: 62,010 x 4
##    word           Austen     Bronte        Wells
##    <chr>           <dbl>      <dbl>        <dbl>
##  1 '          NA          NA          0.00000126
##  2 a'ch       NA          NA          0.000000628
##  3 a'chitect  NA          NA          0.00000126
##  4 a'eplane   NA          NA          0.000000628
##  5 a'hm       NA          NA          0.000000628
```

```
##  6 a'll      NA          NA          0.00000188
##  7 a'most    NA           0.0000109  0.00000251
##  8 a'n't      0.00000462 NA         NA
##  9 a'penny   NA          NA          0.000000628
## 10 aa        NA          NA          0.000000628
## # ... with 62,000 more rows
```

```
frequency <- frequency_by_word_across_authors %>%
  gather(author, proportion, `Bronte`:`Wells`)

frequency
```

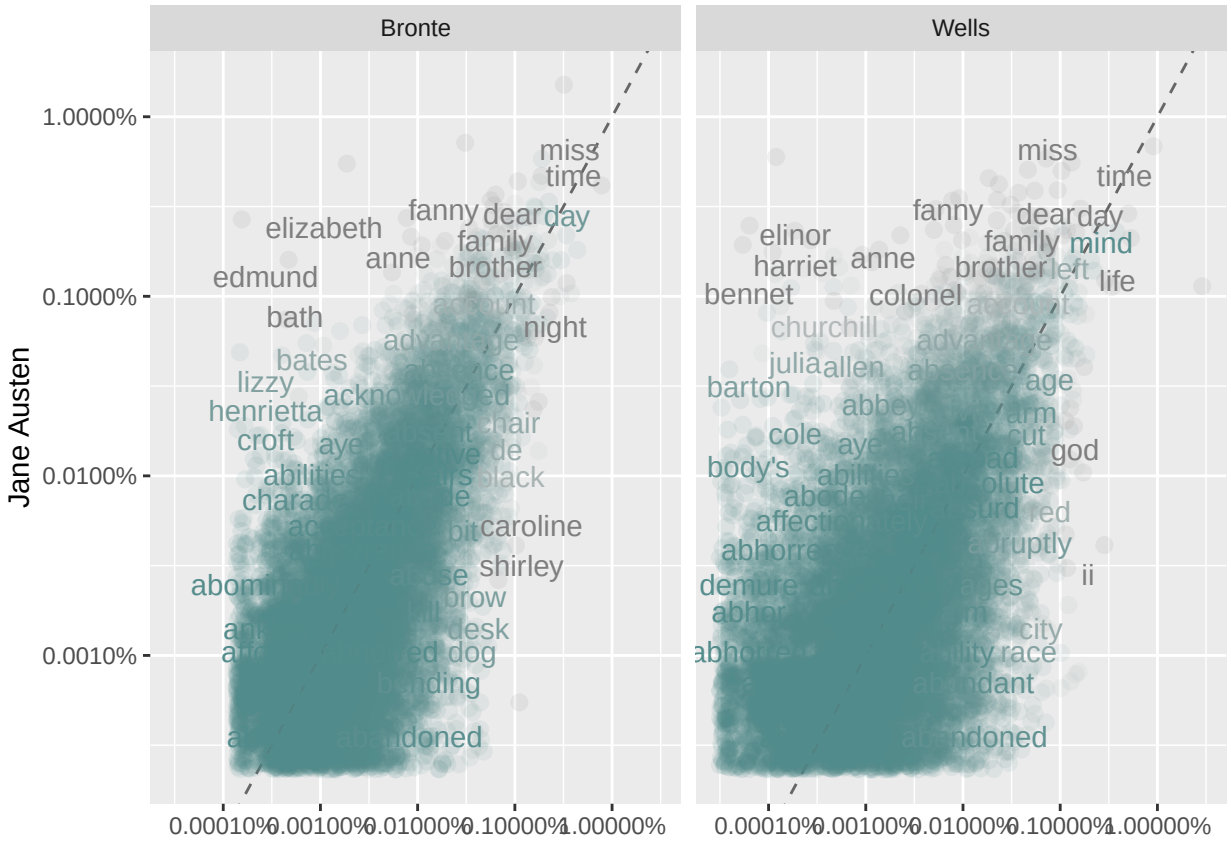
```
## # A tibble: 124,020 x 4
##    word           Austen author proportion
##    <chr>           <dbl> <chr>       <dbl>
##  1 '          NA         Bronte NA
##  2 a'ch       NA         Bronte NA
##  3 a'chitect  NA         Bronte NA
##  4 a'eplane   NA         Bronte NA
##  5 a'hm       NA         Bronte NA
##  6 a'll       NA         Bronte NA
##  7 a'most     NA         Bronte  0.0000109
##  8 a'n't       0.00000462 Bronte NA
##  9 a'penny    NA         Bronte NA
## 10 aa         NA         Bronte NA
## # ... with 124,010 more rows
```

Frequency Graph

```
frequency %>% ggplot(aes(x = proportion,
                         y = `Austen`,
                         color = abs(`Austen` - proportion))) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, 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, 0.001),
                       low = "darkslategray4",
                       high = "gray75") +
  facet_wrap(~author, ncol = 2) +
  theme(legend.position="none") +
  labs(y = "Jane Austen", x = NULL)
```

```
## Warning: Removed 101222 rows containing missing values (geom_point).
```

```
## Warning: Removed 101224 rows containing missing values (geom_text).
```




```
df_Bronte <- frequency[frequency$author == "Bronte",]
df_Bronte
```

```
## # A tibble: 62,010 x 4
##    word           Austen author proportion
##    <chr>           <dbl> <chr>       <dbl>
##  1 '          NA         Bronte NA
##  2 a'ch       NA         Bronte NA
##  3 a'chitect  NA         Bronte NA
##  4 a'eplane   NA         Bronte NA
##  5 a'hm       NA         Bronte NA
##  6 a'll       NA         Bronte NA
##  7 a'most     NA         Bronte  0.0000109
##  8 a'n't       0.00000462 Bronte NA
##  9 a'penny    NA         Bronte NA
## 10 aa         NA         Bronte NA
## # ... with 62,000 more rows
```

```
cor.test(data = df_Bronte, ~ proportion + `Austen`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Austen
## t = 118.23, df = 10943, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7405885 0.7570447
## sample estimates:
##      cor
## 0.748932
```

```
df_Wells <- frequency[frequency$author == "Wells",]
df_Wells
```

```
## # A tibble: 62,010 x 4
##    word           Austen author   proportion
##    <chr>           <dbl> <chr>         <dbl>
##  1 '          NA         Wells   0.00000126
##  2 a'ch       NA         Wells   0.000000628
##  3 a'chitect  NA         Wells   0.00000126
##  4 a'eplane   NA         Wells   0.000000628
##  5 a'hm       NA         Wells   0.000000628
##  6 a'll       NA         Wells   0.00000188
##  7 a'most     NA         Wells   0.00000251
##  8 a'n't       0.00000462 Wells  NA
##  9 a'penny    NA         Wells   0.000000628
## 10 aa         NA         Wells   0.000000628
## # ... with 62,000 more rows
```

```
cor.test(data = df_Wells, ~ proportion + `Austen`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Austen
## t = 44.839, df = 11851, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3653507 0.3961354
## sample estimates:
##       cor
## 0.3808486
```

Exercise 2:

Getting all the books

```
mark_Twain<-gutenberg_works(author == "Twain, Mark")
mark_Twain1<-gutenberg_download(mark_Twain$gutenberg_id)
```

```
## Warning in .f(.x[[i]], ...): Could not download a book at http://
## aleph.gutenberg.org/1/9/6/8/19682/19682.zip
```

```
## Warning in .f(.x[[i]], ...): Could not download a book at http://
## aleph.gutenberg.org/1/9/8/4/19841/19841.zip
```

```
leo_tolstoy<-gutenberg_works(author == "Tolstoy, Leo, graf")
leo_tolstoy1<-gutenberg_download(leo_tolstoy$gutenberg_id)

charles_dickens<-gutenberg_works(author == "Dickens, Charles")
charles_dickens1<-gutenberg_download(charles_dickens$gutenberg_id)
```

Getting word frequncy for each author

```
tidy_twain <- mark_Twain1 %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidy_tolstoy <- leo_tolstoy1 %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidy_dickens <- charles_dickens1 %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidy_twain %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 46,964 x 2
##    word        n
##    <chr>   <int>
##  1 <NA>    13608
##  2 time    10504
##  3 day      6651
##  4 people   4861
##  5 night    3815
##  6 hundred  3650
##  7 tom      3543
##  8 life     3394
##  9 head     3361
## 10 house    3230
## # ... with 46,954 more rows
```

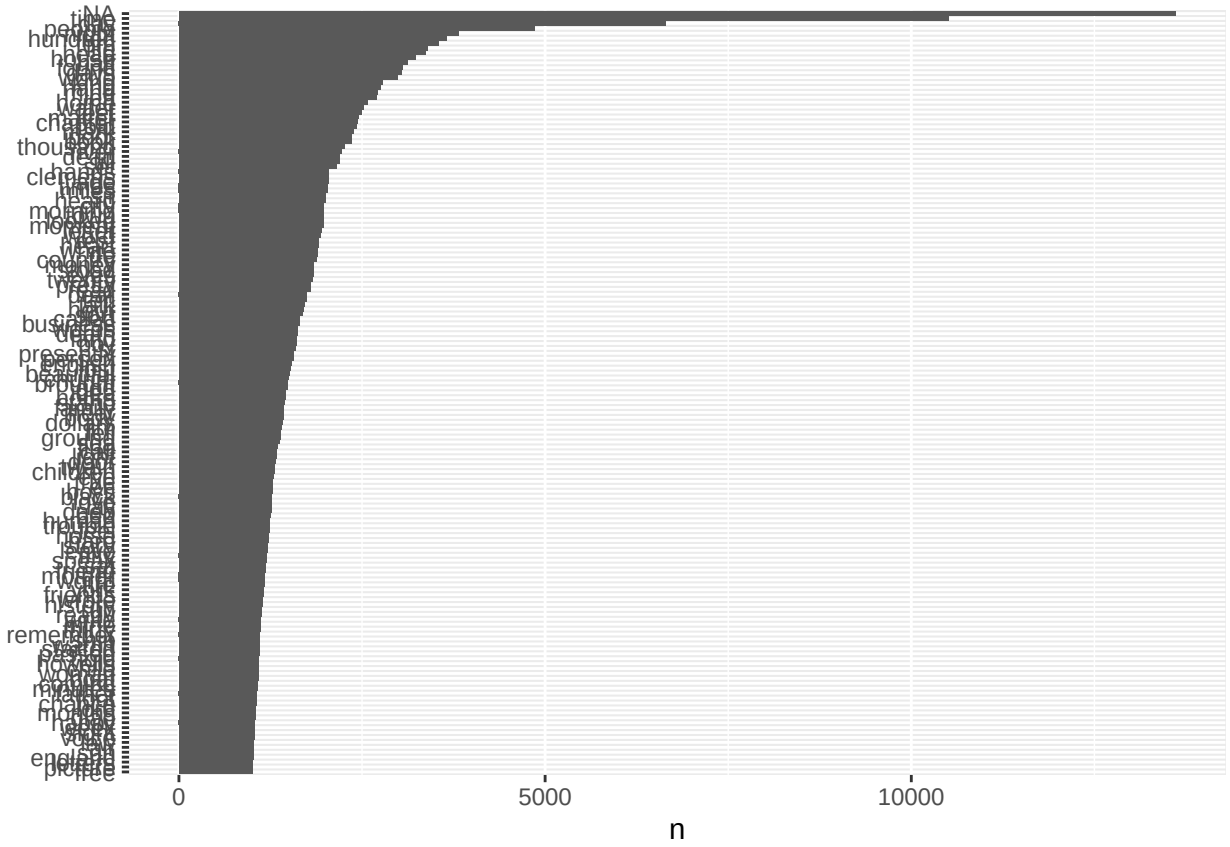
```
tidy_tolstoy %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 38,680 x 2
##    word       n
##    <chr>  <int>
##  1 <NA>   10826
##  2 life    7212
##  3 time    5320
##  4 people  5262
##  5 love    3212
##  6 day     3193
##  7 eyes    2924
##  8 god     2778
##  9 prince  2602
## 10 head    2463
## # ... with 38,670 more rows
```

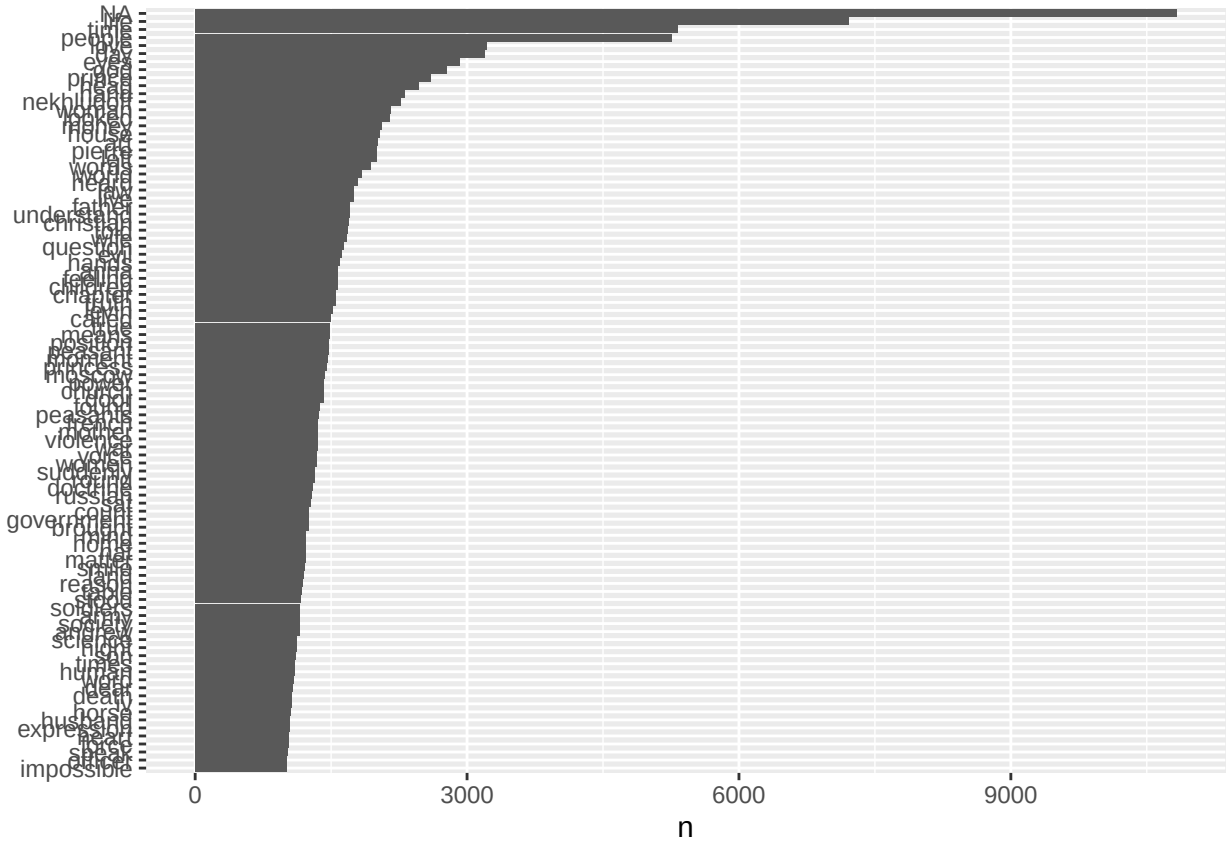
```
tidy_dickens %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 48,730 x 2
##    word      n
##    <chr> <int>
##  1 time  13115
##  2 sir   12993
##  3 dear   9336
##  4 <NA>   8722
##  5 hand   8492
##  6 miss   8348
##  7 head   8327
##  8 day    8322
##  9 night  8081
## 10 house  7716
## # ... with 48,720 more rows
```

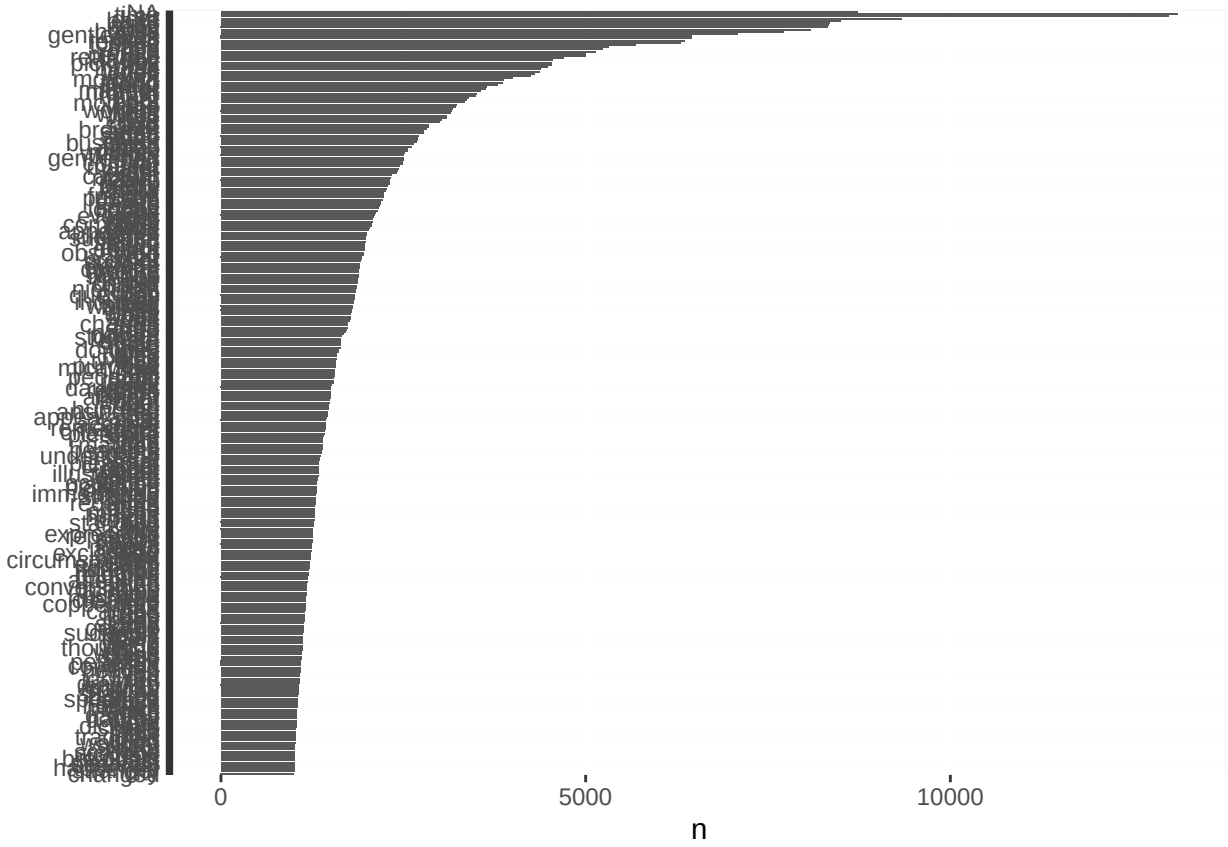
```
tidy_twain %>%
  count(word, sort = TRUE) %>%
  filter(n > 1000) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```




```
tidy_tolstoy %>%
  count(word, sort = TRUE) %>%
  filter(n > 1000) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```




```
tidy_dickens %>%
  count(word, sort = TRUE) %>%
  filter(n > 1000) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```



comparing the frequnency

```
frequency_by_word_across_authors <- bind_rows(
  mutate(tidy_twain,author = "Twain"),
  mutate(tidy_tolstoy, author = "Tolstoy"),
  mutate(tidy_dickens, author = "Dickens")) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(author, proportion)

frequency_by_word_across_authors
```

```
## # A tibble: 78,343 x 4
##    word            Dickens Tolstoy        Twain
##    <chr>             <dbl>   <dbl>        <dbl>
##  1 a'beckett   0.00000126       NA NA
##  2 a'exposer  NA                NA  0.000000622
##  3 a'hoy      NA                NA  0.000000622
##  4 a'int       0.000000420      NA NA
##  5 a'mighty's  0.000000420      NA NA
```

```
##  6 a'most      0.0000264        NA NA
##  7 a'ms       NA                NA  0.000000622
##  8 a'n't       0.00000252       NA NA
##  9 a'nt        0.00000168       NA  0.000000622
## 10 a'purpose   0.000000840      NA NA
## # ... with 78,333 more rows
```

```
frequency <- frequency_by_word_across_authors %>%
  gather(author, proportion, `Twain`:`Tolstoy`)

frequency
```

```
## # A tibble: 156,686 x 4
##    word            Dickens author   proportion
##    <chr>             <dbl> <chr>         <dbl>
##  1 a'beckett   0.00000126  Twain  NA
##  2 a'exposer  NA           Twain   0.000000622
##  3 a'hoy      NA           Twain   0.000000622
##  4 a'int       0.000000420 Twain  NA
##  5 a'mighty's  0.000000420 Twain  NA
##  6 a'most      0.0000264   Twain  NA
##  7 a'ms       NA           Twain   0.000000622
##  8 a'n't       0.00000252  Twain  NA
##  9 a'nt        0.00000168  Twain   0.000000622
## 10 a'purpose   0.000000840 Twain  NA
## # ... with 156,676 more rows
```

Frequency Graph

```
frequency %>% ggplot(aes(x = proportion,
                         y = `Dickens`,
                         color = abs(`Dickens` - proportion))) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, 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, 0.001),
                       low = "darkslategray4",
                       high = "gray75") +
  facet_wrap(~author, ncol = 2) +
  theme(legend.position="none") +
  labs(y = "Charles Dickens", x = NULL)
```

```
## Warning: Removed 103669 rows containing missing values (geom_point).
```

```
## Warning: Removed 103671 rows containing missing values (geom_text).
```


```
df_Twain <- frequency[frequency$author == "Twain",]
df_Twain
```

```
## # A tibble: 78,343 x 4
##    word            Dickens author   proportion
##    <chr>             <dbl> <chr>         <dbl>
##  1 a'beckett   0.00000126  Twain  NA
##  2 a'exposer  NA           Twain   0.000000622
##  3 a'hoy      NA           Twain   0.000000622
##  4 a'int       0.000000420 Twain  NA
##  5 a'mighty's  0.000000420 Twain  NA
##  6 a'most      0.0000264   Twain  NA
##  7 a'ms       NA           Twain   0.000000622
##  8 a'n't       0.00000252  Twain  NA
##  9 a'nt        0.00000168  Twain   0.000000622
## 10 a'purpose   0.000000840 Twain  NA
## # ... with 78,333 more rows
```

```
cor.test(data = df_Twain, ~ proportion + `Dickens`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Dickens
## t = 200.1, df = 28400, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7600149 0.7696674
## sample estimates:
##       cor
## 0.7648841
```

```
df_Tolstoy <- frequency[frequency$author == "Tolstoy",]
df_Tolstoy
```

```
## # A tibble: 78,343 x 4
##    word            Dickens author  proportion
##    <chr>             <dbl> <chr>        <dbl>
##  1 a'beckett   0.00000126  Tolstoy         NA
##  2 a'exposer  NA           Tolstoy         NA
##  3 a'hoy      NA           Tolstoy         NA
##  4 a'int       0.000000420 Tolstoy         NA
##  5 a'mighty's  0.000000420 Tolstoy         NA
##  6 a'most      0.0000264   Tolstoy         NA
##  7 a'ms       NA           Tolstoy         NA
##  8 a'n't       0.00000252  Tolstoy         NA
##  9 a'nt        0.00000168  Tolstoy         NA
## 10 a'purpose   0.000000840 Tolstoy         NA
## # ... with 78,333 more rows
```

```
cor.test(data = df_Tolstoy, ~ proportion + `Dickens`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Dickens
## t = 148.8, df = 24613, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6815388 0.6946925
## sample estimates:
##       cor
## 0.6881722
```

Charles Dickens have more Similarity with Mark Twain's Works.

Exercise 3:

Getting the data

```
Les_Miserables<- gutenberg_download(c(135))
A_Tale_of_Two_Cities<- gutenberg_download(c(98))
```

cleaning the data

```
tidy_victor <- Les_Miserables %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidy_charles <- A_Tale_of_Two_Cities %>%
  unnest_tokens(word, text) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

Analysis

```
bing_word_counts1 <- tidy_victor %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

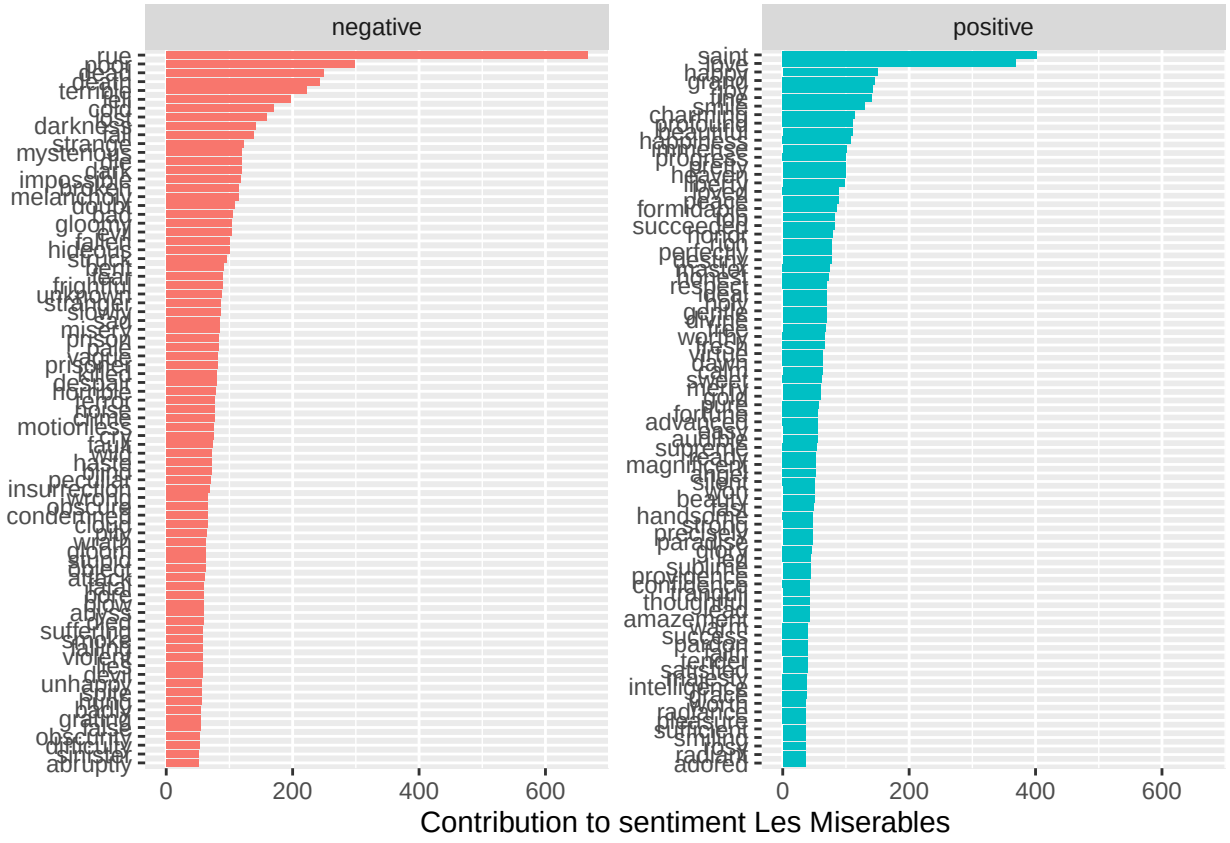
```
bing_word_counts1
```

```
## # A tibble: 2,863 x 3
##    word      sentiment     n
##    <chr>     <chr>     <int>
##  1 rue       negative    667
##  2 saint     positive    402
##  3 love      positive    369
##  4 poor      negative    299
##  5 dead      negative    249
##  6 death     negative    243
##  7 terrible negative    223
##  8 fell      negative    197
##  9 cold      negative    170
## 10 lost      negative    159
## # ... with 2,853 more rows
```

```
bing_word_counts1 %>%
  group_by(sentiment) %>%
  top_n(81) %>%
  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 Les Miserables",
       x = NULL) +
  coord_flip()
```

```
## Selecting by n
```




```
bing_word_counts2 <- tidy_charles %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

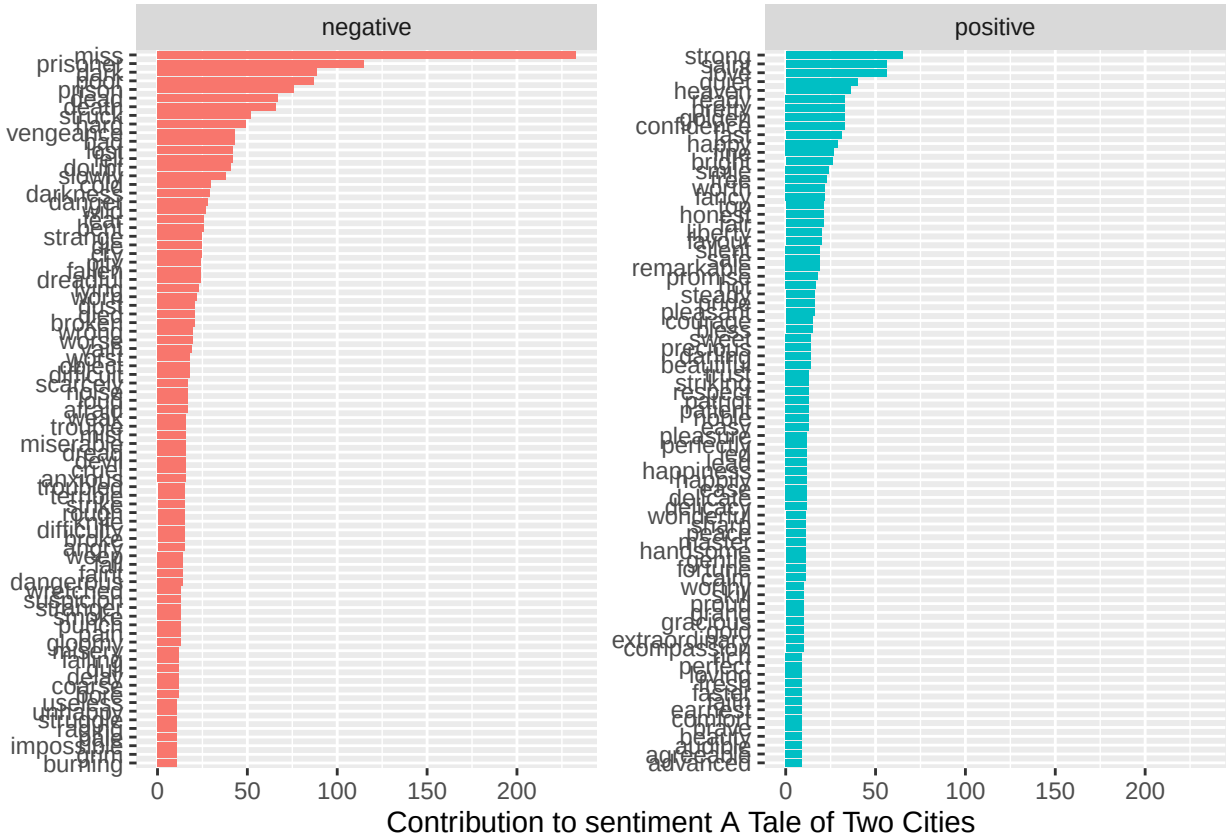
```
bing_word_counts2
```

```
## # A tibble: 1,840 x 3
##    word     sentiment     n
##    <chr>    <chr>     <int>
##  1 miss     negative    233
##  2 prisoner negative    115
```

```
##  3 dark     negative     89
##  4 poor     negative     87
##  5 prison   negative     76
##  6 dead     negative     67
##  7 death    negative     66
##  8 strong   positive     65
##  9 love     positive     56
## 10 saint    positive     56
## # ... with 1,830 more rows
```

```
bing_word_counts2 %>%
  group_by(sentiment) %>%
  top_n(81) %>%
  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 A Tale of Two Cities",
       x = NULL) +
  coord_flip()
```

```
## Selecting by n
```



joy

```
bing_with_joy <- get_sentiments("bing") %>%
  filter(word == "joy")

bing_word_counts1 <- tidy_charles %>%
  inner_join(bing_with_joy) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

```
bing_word_counts1
```

```
## # A tibble: 1 x 3
##   word  sentiment     n
##   <chr> <chr>     <int>
## 1 joy   positive      6
```

```
bing_word_counts2 <- tidy_victor %>%
  inner_join(bing_with_joy) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

```
bing_word_counts2
```

```
## # A tibble: 1 x 3
##   word  sentiment     n
##   <chr> <chr>     <int>
## 1 joy   positive    143
```