Chapter 4

John Peach

3/28/2021

```
austen_bigrams <- austen_books() %>%
  unnest_tokens(bigram, text, token = 'ngrams', n = 2)

austen_bigrams
```

```
## # A tibble: 675,025 x 2
##    book                bigram
##    <fct>               <chr>
##  1 Sense & Sensibility sense and
##  2 Sense & Sensibility and sensibility
##  3 Sense & Sensibility <NA>
##  4 Sense & Sensibility by jane
##  5 Sense & Sensibility jane austen
##  6 Sense & Sensibility <NA>
##  7 Sense & Sensibility <NA>
##  8 Sense & Sensibility <NA>
##  9 Sense & Sensibility <NA>
## 10 Sense & Sensibility <NA>
## # ... with 675,015 more rows
```

```
austen_bigrams %>%
  count(bigram, sort = TRUE)
```

```
## # A tibble: 193,210 x 2
##    bigram      n
##    <chr>   <int>
##  1 <NA>    12242
##  2 of the   2853
##  3 to be    2670
##  4 in the   2221
##  5 it was   1691
##  6 i am     1485
##  7 she had  1405
##  8 of her   1363
##  9 to the   1315
## 10 she was  1309
## # ... with 193,200 more rows
```

```
library(tidyr)

bigrams_seperated <- austen_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_filtered <- bigrams_seperated %>%
```

```
  dplyr::filter(!word1 %in% stop_words$word) %>%
  dplyr::filter(!word2 %in% stop_words$word)

# new bigram counts:
bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

bigram_counts
```

```
## # A tibble: 28,975 x 3
##    word1   word2         n
##    <chr>   <chr>     <int>
##  1 <NA>    <NA>      12242
##  2 sir     thomas      266
##  3 miss    crawford    196
##  4 captain wentworth   143
##  5 miss    woodhouse   143
##  6 frank   churchill   114
##  7 lady    russell     110
##  8 sir     walter      108
##  9 lady    bertram     101
## 10 miss    fairfax      98
## # ... with 28,965 more rows
```

```
bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = ' ')

bigrams_united
```

```
## # A tibble: 51,155 x 2
##    book                bigram
##    <fct>               <chr>
##  1 Sense & Sensibility NA NA
##  2 Sense & Sensibility jane austen
##  3 Sense & Sensibility NA NA
##  4 Sense & Sensibility NA NA
##  5 Sense & Sensibility NA NA
##  6 Sense & Sensibility NA NA
##  7 Sense & Sensibility NA NA
##  8 Sense & Sensibility NA NA
##  9 Sense & Sensibility chapter 1
## 10 Sense & Sensibility NA NA
## # ... with 51,145 more rows
```

```
austen_books() %>%
  unnest_tokens(trigram, text, token = 'ngrams', n = 3) %>%
  separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%
  dplyr::filter(!word1 %in% stop_words$word,
                !word2 %in% stop_words$word,
                !word3 %in% stop_words$word) %>%
  count(word1, word2, word3, sort = TRUE)
```

```
## # A tibble: 6,141 x 4
##    word1     word2     word3         n
##    <chr>     <chr>     <chr>     <int>
```

```
##  1 <NA>      <NA>      <NA>      13260
##  2 dear      miss      woodhouse    20
##  3 miss      de        bourgh       17
##  4 lady      catherine de           11
##  5 poor      miss      taylor       11
##  6 sir       walter    elliot       10
##  7 catherine de        bourgh        9
##  8 dear      sir       thomas        8
##  9 replied   miss      crawford      7
## 10 sir       william   lucas         7
## # ... with 6,131 more rows
```

```
bigrams_filtered %>%
  dplyr::filter(word2 == 'street') %>%
  count(book, word1, sort = TRUE)
```

```
## # A tibble: 33 x 3
##    book                word1           n
##    <fct>               <chr>       <int>
##  1 Sense & Sensibility harley          16
##  2 Sense & Sensibility berkeley        15
##  3 Northanger Abbey    milsom          10
##  4 Northanger Abbey    pulteney        10
##  5 Mansfield Park      wimpole          9
##  6 Pride & Prejudice   gracechurch      8
##  7 Persuasion          milsom           5
##  8 Sense & Sensibility bond             4
##  9 Sense & Sensibility conduit          4
## 10 Persuasion          rivers           4
## # ... with 23 more rows
```

```
bigram_tf_idf <- bigrams_united %>%
  count(book, bigram) %>%
  bind_tf_idf(bigram, book, n) %>%
  arrange(desc(tf_idf))

bigram_tf_idf
```

```
## # A tibble: 31,397 x 6
##    book                bigram                n     tf   idf tf_idf
##    <fct>               <chr>             <int>  <dbl> <dbl>  <dbl>
##  1 Mansfield Park      sir thomas          266 0.0244  1.79 0.0438
##  2 Persuasion          captain wentworth   143 0.0232  1.79 0.0416
##  3 Mansfield Park      miss crawford       196 0.0180  1.79 0.0322
##  4 Persuasion          lady russell        110 0.0179  1.79 0.0320
##  5 Persuasion          sir walter          108 0.0175  1.79 0.0314
##  6 Emma                miss woodhouse      143 0.0129  1.79 0.0231
##  7 Northanger Abbey    miss tilney          74 0.0128  1.79 0.0229
##  8 Sense & Sensibility colonel brandon      96 0.0115  1.79 0.0205
##  9 Sense & Sensibility sir john             94 0.0112  1.79 0.0201
## 10 Emma                frank churchill     114 0.0103  1.79 0.0184
## # ... with 31,387 more rows
```

```
bigrams_seperated %>%
  dplyr::filter(word1 == 'not') %>%
  count(word1, word2, sort = TRUE)
```

```
## # A tibble: 1,178 x 3
##    word1 word2     n
##    <chr> <chr> <int>
##  1 not   be      580
##  2 not   to      335
##  3 not   have    307
##  4 not   know    237
##  5 not   a       184
##  6 not   think   162
##  7 not   been    151
##  8 not   the     135
##  9 not   at      126
## 10 not   in      110
## # ... with 1,168 more rows
```

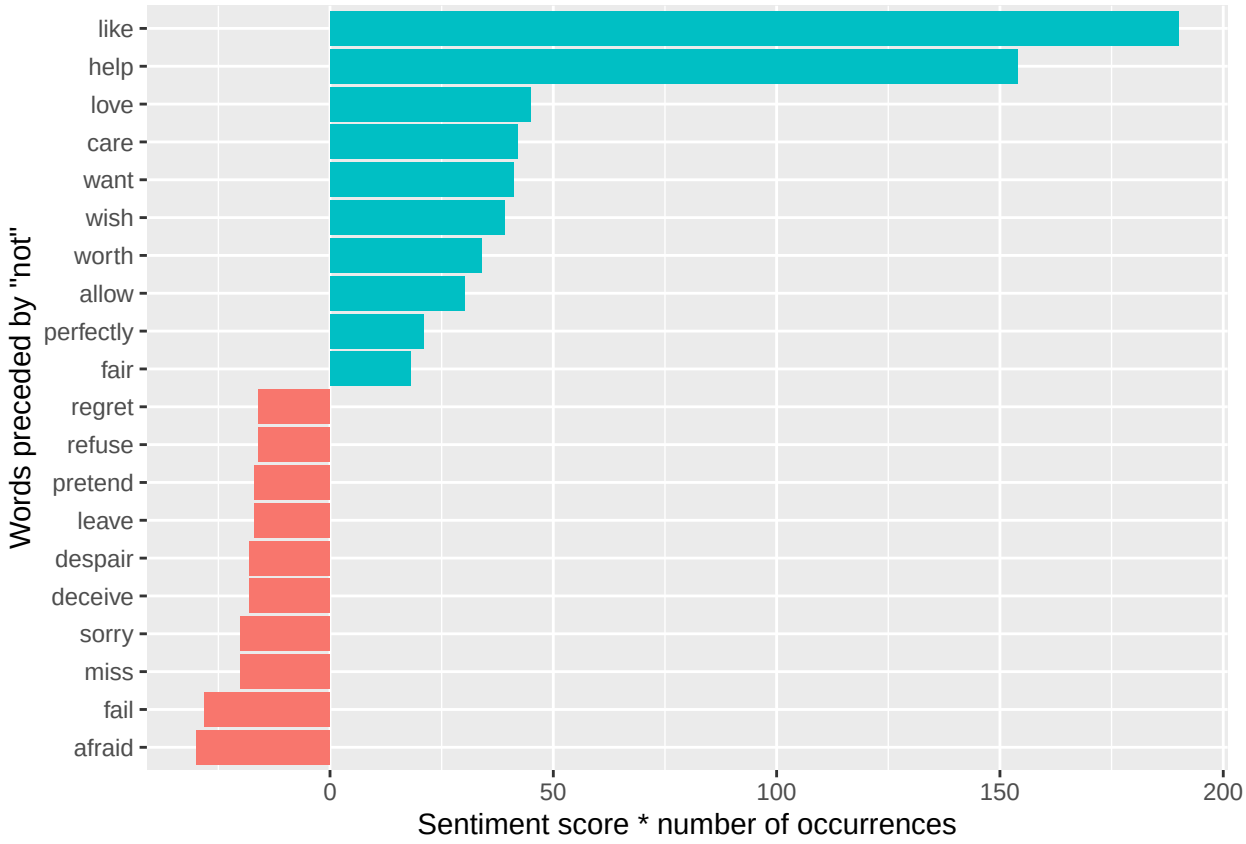
```
AFINN <- get_sentiments('afinn')
AFINN
```

```
## # A tibble: 2,477 x 2
##    word       value
##    <chr>      <dbl>
##  1 abandon       -2
##  2 abandoned     -2
##  3 abandons      -2
##  4 abducted      -2
##  5 abduction     -2
##  6 abductions    -2
##  7 abhor         -3
##  8 abhorred      -3
##  9 abhorrent     -3
## 10 abhors        -3
## # ... with 2,467 more rows
```

```
not_words <- bigrams_seperated %>%
  dplyr::filter(word1 == 'not') %>%
  inner_join(AFINN, by = c(word2 = 'word')) %>%
  count(word2, value, sort = TRUE) %>%
  ungroup() %>%
  rename(score = value)

not_words
```

```
## # A tibble: 229 x 3
##    word2   score     n
##    <chr>   <dbl> <int>
##  1 like        2    95
##  2 help        2    77
##  3 want        1    41
##  4 wish        1    39
##  5 allow       1    30
##  6 care        2    21
##  7 sorry      -1    20
##  8 leave      -1    17
##  9 pretend    -1    17
## 10 worth       2    17
## # ... with 219 more rows
```

```
not_words %>%
  mutate(contribution = n * score) %>%
  arrange(desc(abs(contribution))) %>%
  head(20) %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * score, fill = n * score > 0)) +
    geom_col(show.legend = FALSE) +
    xlab("Words preceded by \"not\"") +
    ylab("Sentiment score * number of occurrences") +
    coord_flip()
```




```
negation_words <- c("not", "no", "never", "without")

negation_words <- bigrams_seperated %>%
  dplyr::filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by = c(word2 = 'word')) %>%
  count(word1, word2, value, sort = TRUE) %>%
  ungroup()
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing

## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following object is masked from 'package:base':
##
##     union
```

```
bigram_counts
```

```
## # A tibble: 28,975 x 3
##    word1   word2         n
##    <chr>   <chr>     <int>
##  1 <NA>    <NA>      12242
##  2 sir     thomas      266
##  3 miss    crawford    196
##  4 captain wentworth   143
##  5 miss    woodhouse   143
##  6 frank   churchill   114
##  7 lady    russell     110
##  8 sir     walter      108
##  9 lady    bertram     101
## 10 miss    fairfax      98
## # ... with 28,965 more rows
```

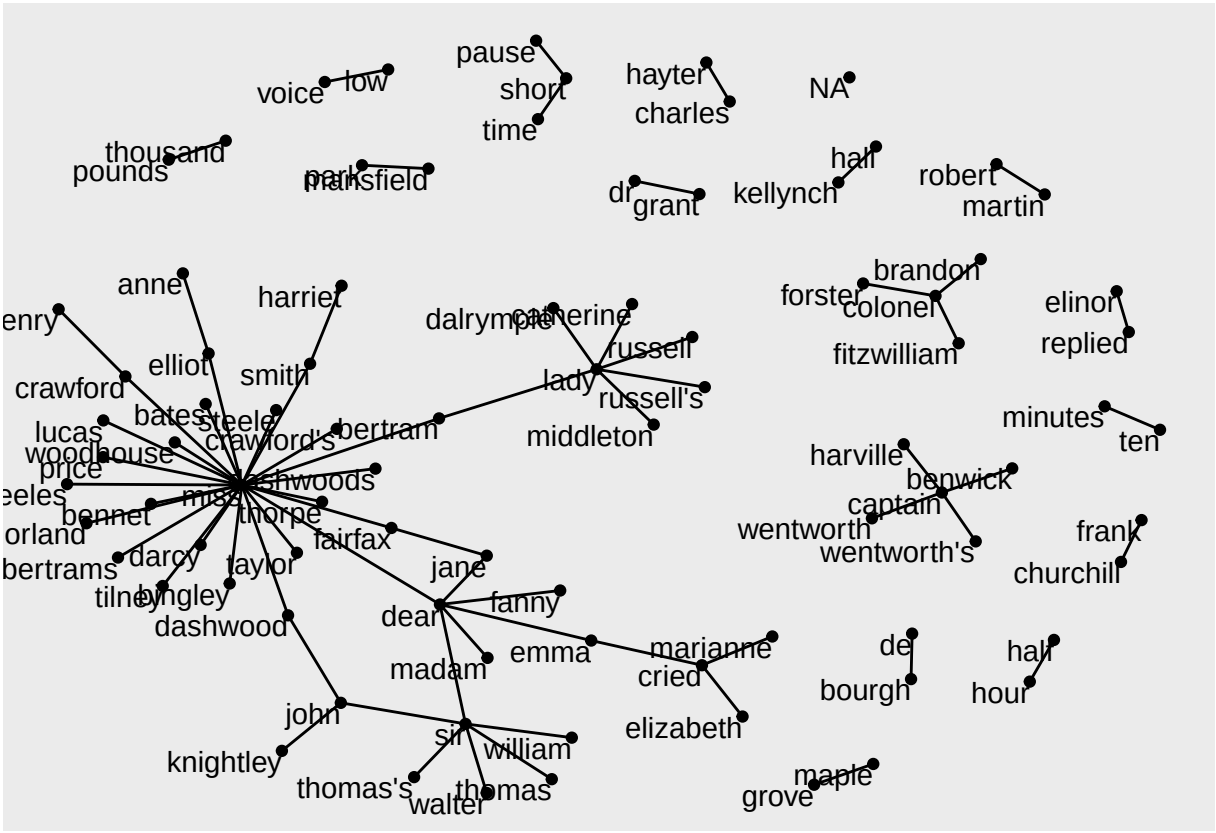
```
bigram_graph <- bigram_counts %>%
  dplyr::filter(n > 20) %>%
  graph_from_data_frame()
```

```
## Warning in graph_from_data_frame(.): In `d' `NA' elements were replaced with
## string "NA"
```

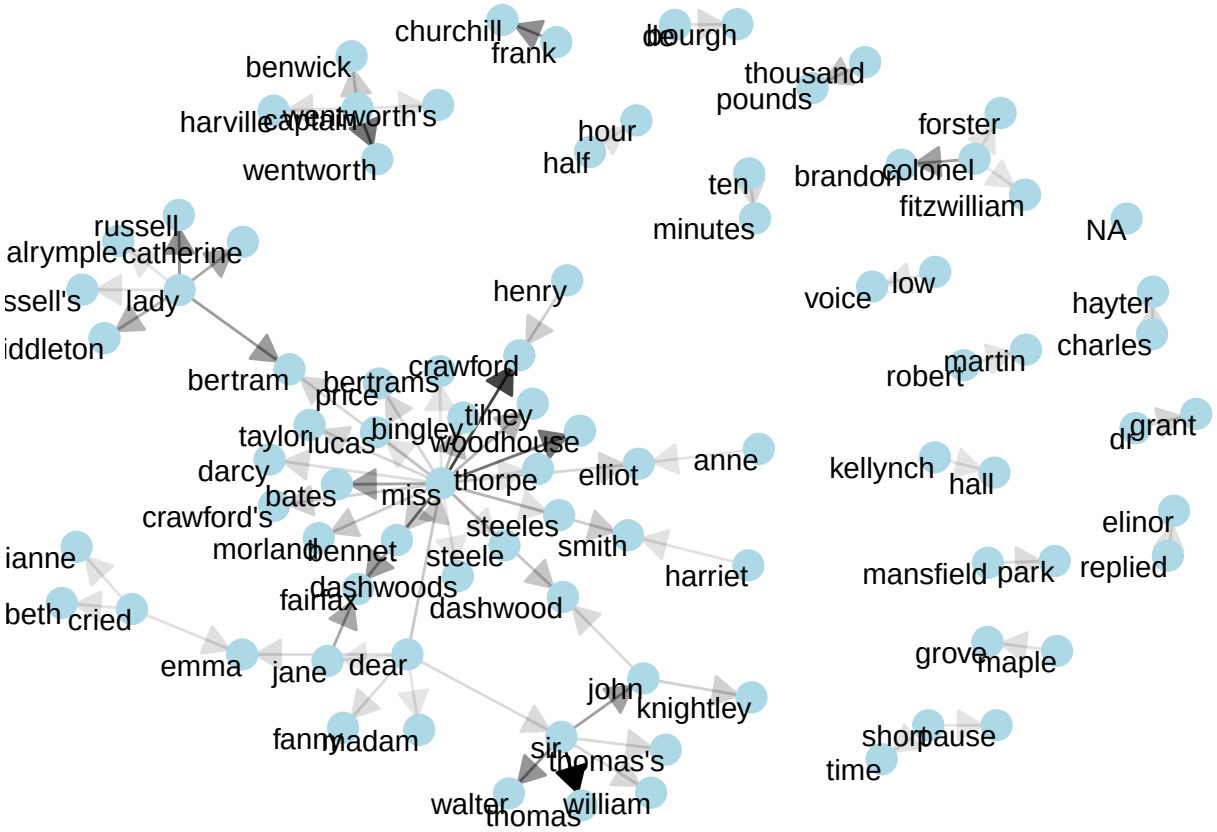
```
bigram_graph
```

```
## IGRAPH 2644a90 DN-- 86 71 --
## + attr: name (v/c), n (e/n)
## + edges from 2644a90 (vertex names):
##  [1] NA      ->NA         sir     ->thomas     miss    ->crawford
##  [4] captain ->wentworth  miss    ->woodhouse  frank   ->churchill
##  [7] lady    ->russell    sir     ->walter     lady    ->bertram
## [10] miss    ->fairfax    colonel ->brandon    sir     ->john
## [13] miss    ->bates      jane    ->fairfax    lady    ->catherine
## [16] lady    ->middleton  miss    ->tilney     miss    ->bingley
## [19] thousand->pounds     miss    ->dashwood   dear    ->miss
## [22] miss    ->bennet     miss    ->morland    captain ->benwick
## + ... omitted several edges
```

```
library(ggraph)
set.seed(2017)

ggraph(bigram_graph, layout = 'fr') +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```




```
set.seed(2016)
a <- grid::arrow(type = 'closed', length = unit(0.15, 'inches'))

ggraph(bigram_graph, layout = 'fr') +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(0.07, 'inches')) +
  geom_node_point(color = 'lightblue', size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```

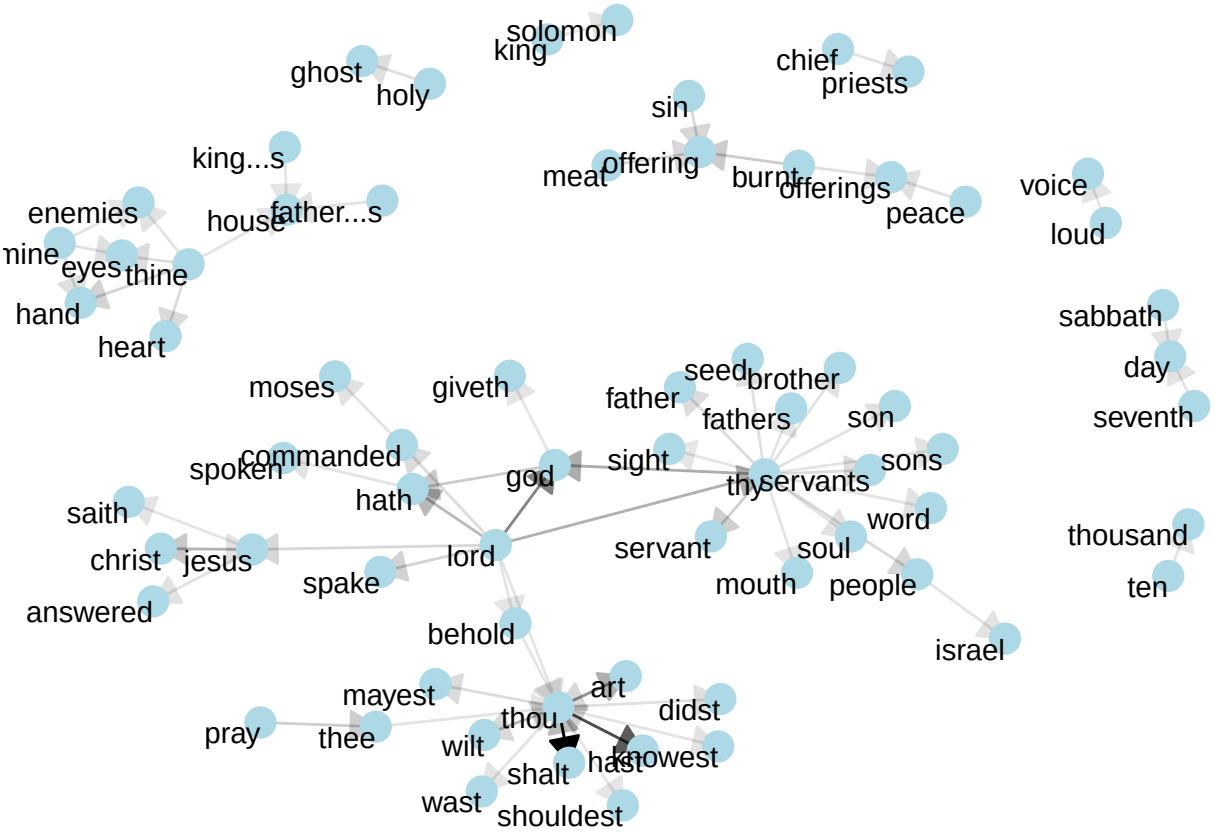



```
library(dplyr)
library(tidyr)
library(tidytext)
library(ggplot2)
library(igraph)
library(ggraph)

count_bigrams <- function(dataset) {
  dataset %>%
    unnest_tokens(bigram, text, token = 'ngrams', n = 2) %>%
    separate(bigram, c('word1', 'word2'), sep = ' ') %>%
    dplyr::filter(!word1 %in% stop_words$word,
                  !word2 %in% stop_words$word) %>%
    count(word1, word2, sort = TRUE)
}

visualize_bigrams <- function(bigrams){
  set.seed(2016)
  a <- grid::arrow(type = 'closed', length = unit(0.15, 'inches'))

  bigrams %>%
    graph_from_data_frame() %>%
    ggraph(layout = 'fr') +
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, arrow = a) +
    geom_node_point(color = 'lightblue', size = 5) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
    theme_void()
```

```
}
```

```
library(gutenbergr)
kjv <- gutenberg_download(10, mirror = 'http://eremita.di.uminho.pt/gutenberg/')

library(stringr)

kjv_bigrams <- kjv %>%
  count_bigrams()

kjv_bigrams %>%
  dplyr::filter(n > 40,
                !str_detect(word1, '\\d'),
                !str_detect(word2, '\\d')) %>%
  visualize_bigrams()
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'father's' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'father's' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'father's' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'king's' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'king's' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'king's' in 'mbcsToSbcs': dot substituted for <99>
```




```
austen_section_words <- austen_books() %>%
  dplyr::filter(book == 'Pride & Prejudice') %>%
  mutate(section = row_number() %/% 10) %>%
  dplyr::filter(section > 0) %>%
  unnest_tokens(word, text) %>%
  dplyr::filter(!word %in% stop_words$word)
```

```
library(widyr)
```

```
word_pair <- austen_section_words %>%
  pairwise_count(word, section, sort = TRUE)
```

```
## Warning: `distinct_()` is deprecated as of dplyr 0.7.0.
## Please use `distinct()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Warning: `tbl_df()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
word_pair
```

```
## # A tibble: 796,008 x 3
##    item1     item2         n
##    <chr>     <chr>     <dbl>
##  1 darcy     elizabeth   144
```

```
##  2 elizabeth darcy       144
##  3 miss      elizabeth   110
##  4 elizabeth miss        110
##  5 elizabeth jane        106
##  6 jane      elizabeth   106
##  7 miss      darcy        92
##  8 darcy     miss         92
##  9 elizabeth bingley      91
## 10 bingley   elizabeth    91
## # ... with 795,998 more rows
```

```
word_pair %>%
  dplyr::filter(item1 == 'darcy')
```

```
## # A tibble: 2,930 x 3
##    item1 item2         n
##    <chr> <chr>     <dbl>
##  1 darcy elizabeth   144
##  2 darcy miss         92
##  3 darcy bingley      86
##  4 darcy jane         46
##  5 darcy bennet       45
##  6 darcy sister       45
##  7 darcy time         41
##  8 darcy lady         38
##  9 darcy friend       37
## 10 darcy wickham      37
## # ... with 2,920 more rows
```

```
word_cors <- austen_section_words %>%
  group_by(word) %>%
  dplyr::filter(n() >= 20) %>%
  pairwise_cor(word, section, sort = TRUE)

word_cors
```

```
## # A tibble: 154,842 x 3
##    item1     item2     correlation
##    <chr>     <chr>           <dbl>
##  1 bourgh    de              0.951
##  2 de        bourgh          0.951
##  3 pounds    thousand        0.701
##  4 thousand  pounds          0.701
##  5 william   sir             0.664
##  6 sir       william         0.664
##  7 catherine lady            0.663
##  8 lady      catherine       0.663
##  9 forster   colonel         0.622
## 10 colonel   forster         0.622
## # ... with 154,832 more rows
```

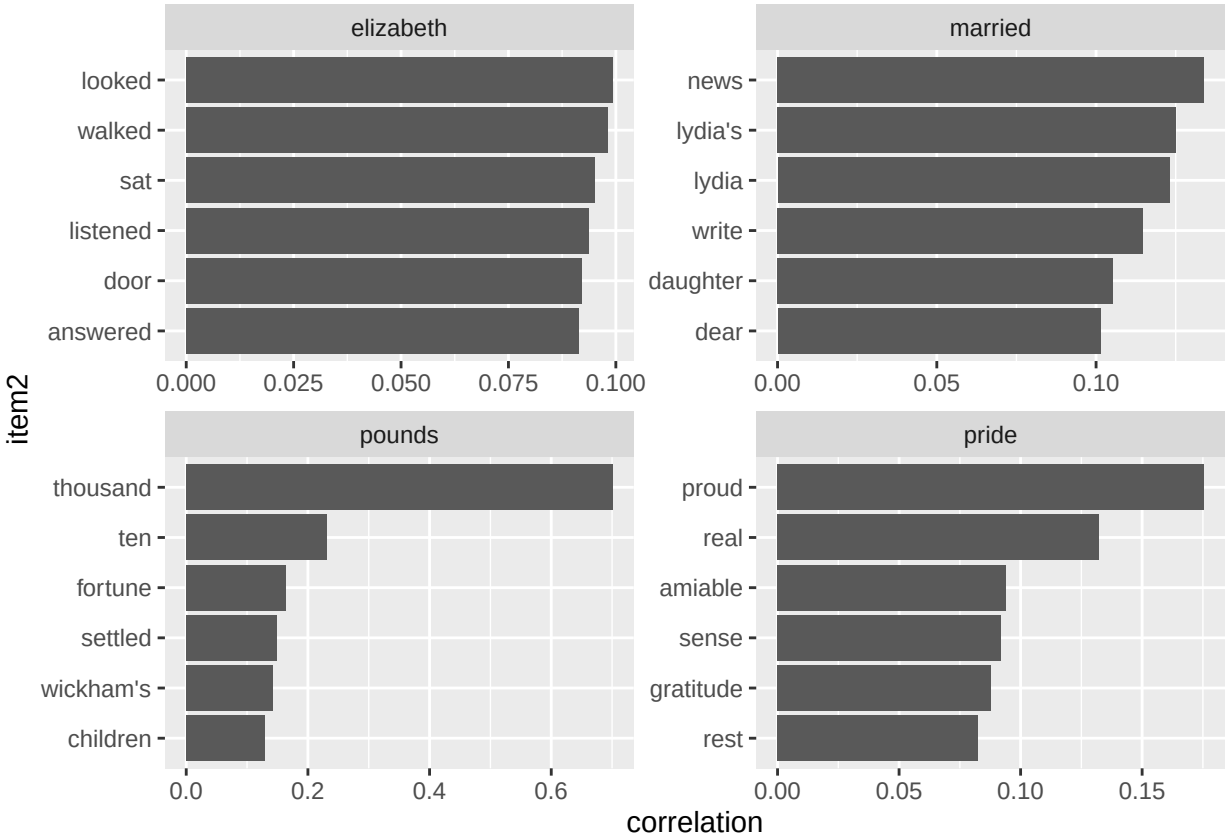
```
word_cors %>%
  dplyr::filter(item1 == 'pounds')
```

```
## # A tibble: 393 x 3
##    item1  item2     correlation
```

```
##    <chr>  <chr>           <dbl>
##  1 pounds thousand       0.701
##  2 pounds ten            0.231
##  3 pounds fortune        0.164
##  4 pounds settled        0.149
##  5 pounds wickham's      0.142
##  6 pounds children       0.129
##  7 pounds mother's       0.119
##  8 pounds believed       0.0932
##  9 pounds estate         0.0890
## 10 pounds ready          0.0860
## # ... with 383 more rows
```

```
word_cors %>%
  dplyr::filter(item1 %in% c('elizabeth', 'pounds', 'married', 'pride')) %>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item2 = reorder(item2, correlation)) %>%
  ggplot(aes(item2, correlation)) +
    geom_bar(stat = 'identity') +
    facet_wrap(~ item1, scales = 'free') +
    coord_flip()
```

```
## Selecting by correlation
```




```
set.seed(2016)
```

```
word_cors %>%
  dplyr::filter(correlation > 0.15) %>%
  graph_from_data_frame() %>%
  ggraph(layout = 'fr') +
    geom_edge_link(aes(edge_alpha = correlation), show.legend = FALSE) +
    geom_node_point(color = 'lightblue', size = 5) +
    geom_node_text(aes(label = name), repel = TRUE) +
    theme_void()
```