Chapter 3

John Peach

3/23/2021

```
book_words <- austen_books() %>%
  unnest_tokens(word, text) %>%
  count(book, word, sort = TRUE) %>%
  ungroup()

total_words <- book_words %>%
  group_by(book) %>%
  summarise(total = sum(n))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
book_words <- left_join(book_words, total_words)
```

```
## Joining, by = "book"
```

```
book_words
```

```
## # A tibble: 40,379 x 4
##    book              word      n  total
##    <fct>             <chr> <int>  <int>
##  1 Mansfield Park    the    6206 160460
##  2 Mansfield Park    to     5475 160460
##  3 Mansfield Park    and    5438 160460
##  4 Emma              to     5239 160996
##  5 Emma              the    5201 160996
##  6 Emma              and    4896 160996
##  7 Mansfield Park    of     4778 160460
##  8 Pride & Prejudice the    4331 122204
##  9 Emma              of     4291 160996
## 10 Pride & Prejudice to     4162 122204
## # ... with 40,369 more rows
```

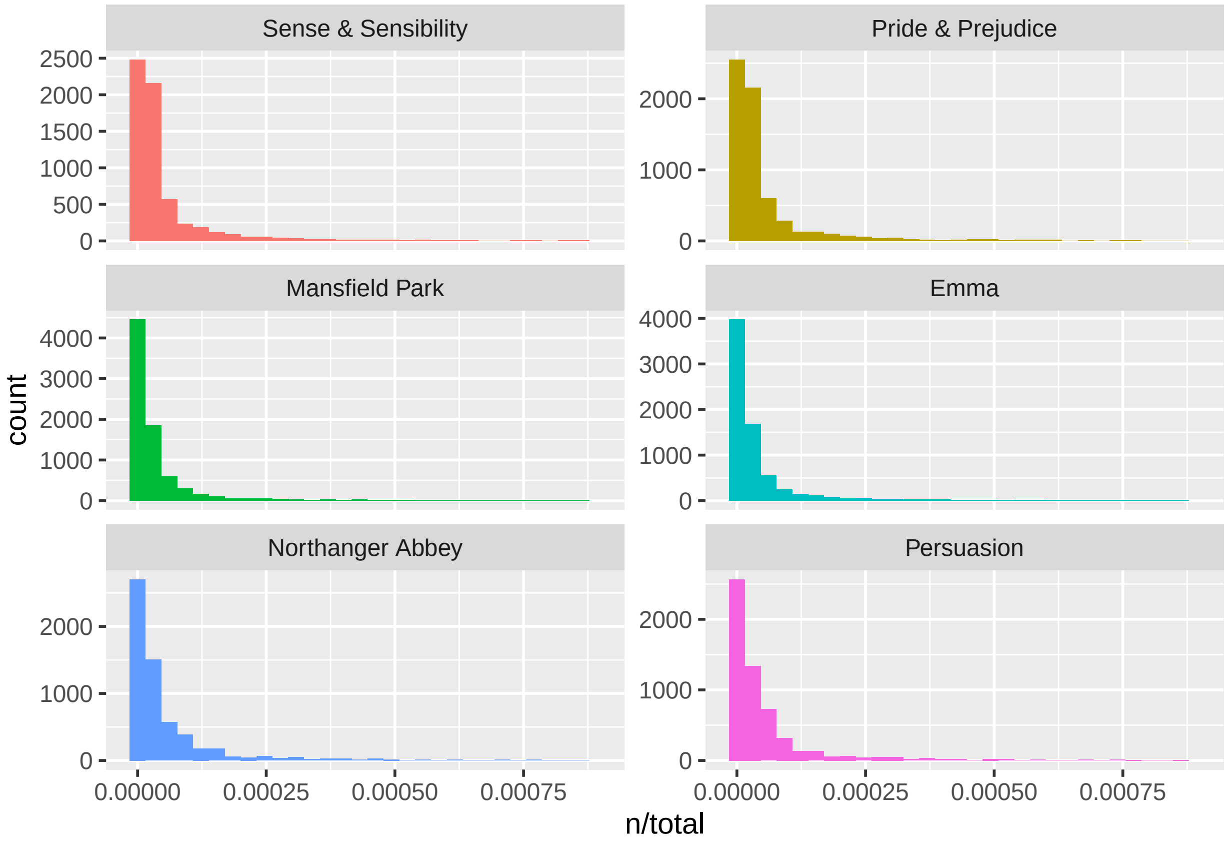
```
library(ggplot2)

ggplot(book_words, aes(n/total, fill = book)) +
  geom_histogram(show.legend = FALSE) +
  xlim(NA, 0.0009) +
  facet_wrap(~book, ncol = 2, scales = 'free_y')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 896 rows containing non-finite values (stat_bin).

## Warning: Removed 6 rows containing missing values (geom_bar).
```

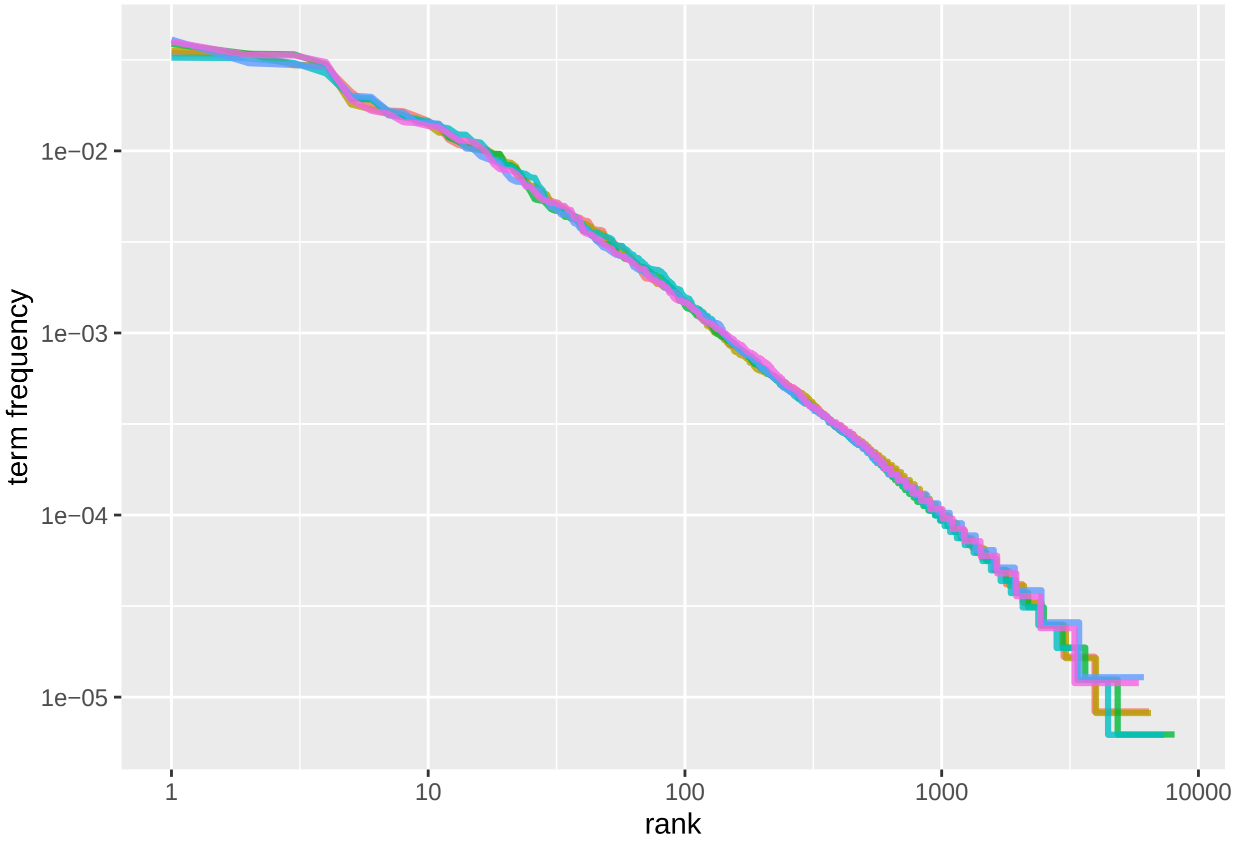


```
freq_by_rank <- book_words %>%
  group_by(book) %>%
  mutate(rank = row_number(), `term frequency` = n/total)

freq_by_rank
```

```
## # A tibble: 40,379 x 6
## # Groups:   book [6]
##    book              word      n  total  rank `term frequency`
##    <fct>             <chr> <int>  <int> <int>            <dbl>
##  1 Mansfield Park    the    6206 160460     1           0.0387
##  2 Mansfield Park    to     5475 160460     2           0.0341
##  3 Mansfield Park    and    5438 160460     3           0.0339
##  4 Emma              to     5239 160996     1           0.0325
##  5 Emma              the    5201 160996     2           0.0323
##  6 Emma              and    4896 160996     3           0.0304
##  7 Mansfield Park    of     4778 160460     4           0.0298
##  8 Pride & Prejudice the    4331 122204     1           0.0354
##  9 Emma              of     4291 160996     4           0.0267
## 10 Pride & Prejudice to     4162 122204     2           0.0341
## # ... with 40,369 more rows
```

```
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color = book)) +
    geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
    scale_x_log10() +
    scale_y_log10()
```

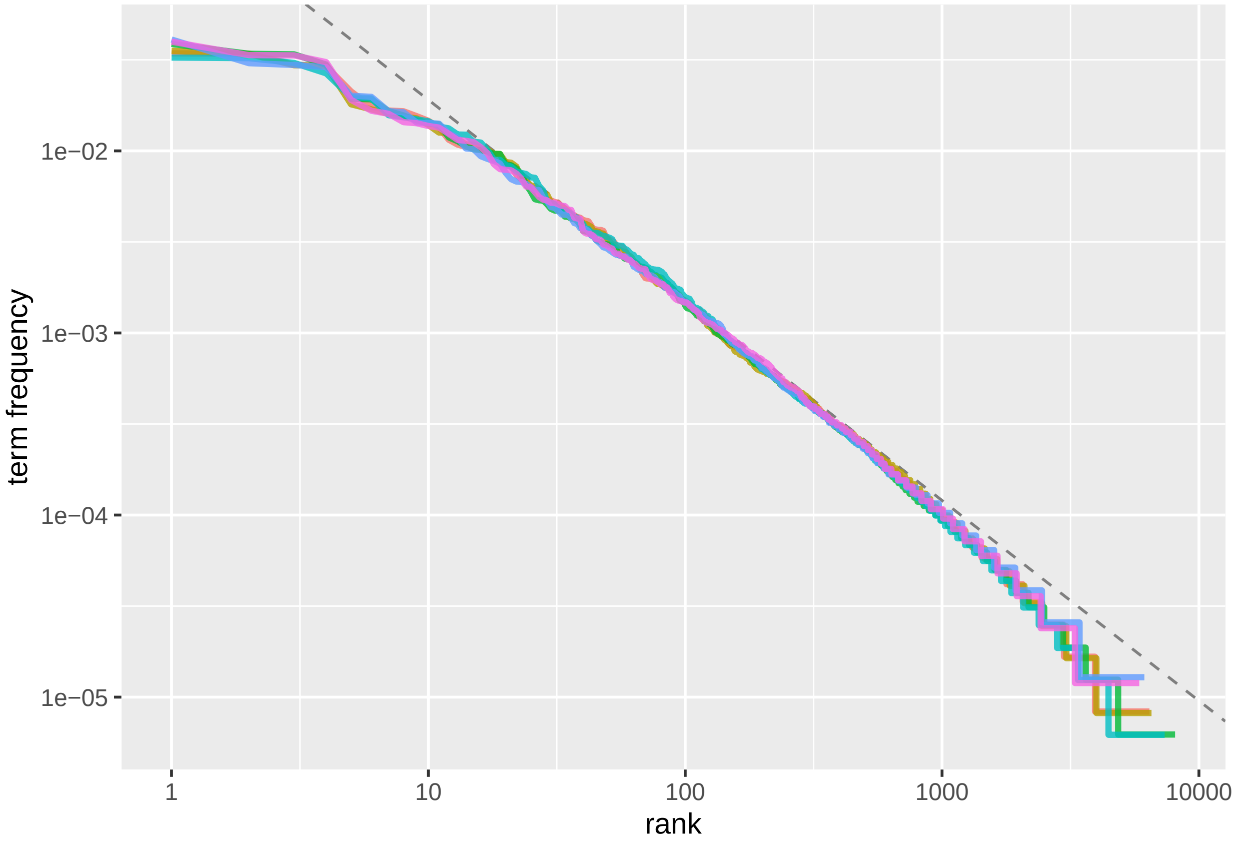


```
rank_subset <- freq_by_rank %>%
  dplyr::filter(rank < 500, rank > 10)

lm(log10(`term frequency`) ~ log10(rank), data = rank_subset)
```

```
##
## Call:
## lm(formula = log10(`term frequency`) ~ log10(rank), data = rank_subset)
##
## Coefficients:
## (Intercept)  log10(rank)
##     -0.6226      -1.1125
```

```
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color = book)) +
    geom_abline(intercept = -0.62, slope = -1.1, color = 'gray50', linetype = 2) +
    geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
    scale_x_log10() +
    scale_y_log10()
```



```
book_words <- book_words %>%
  bind_tf_idf(word, book, n)

book_words
```

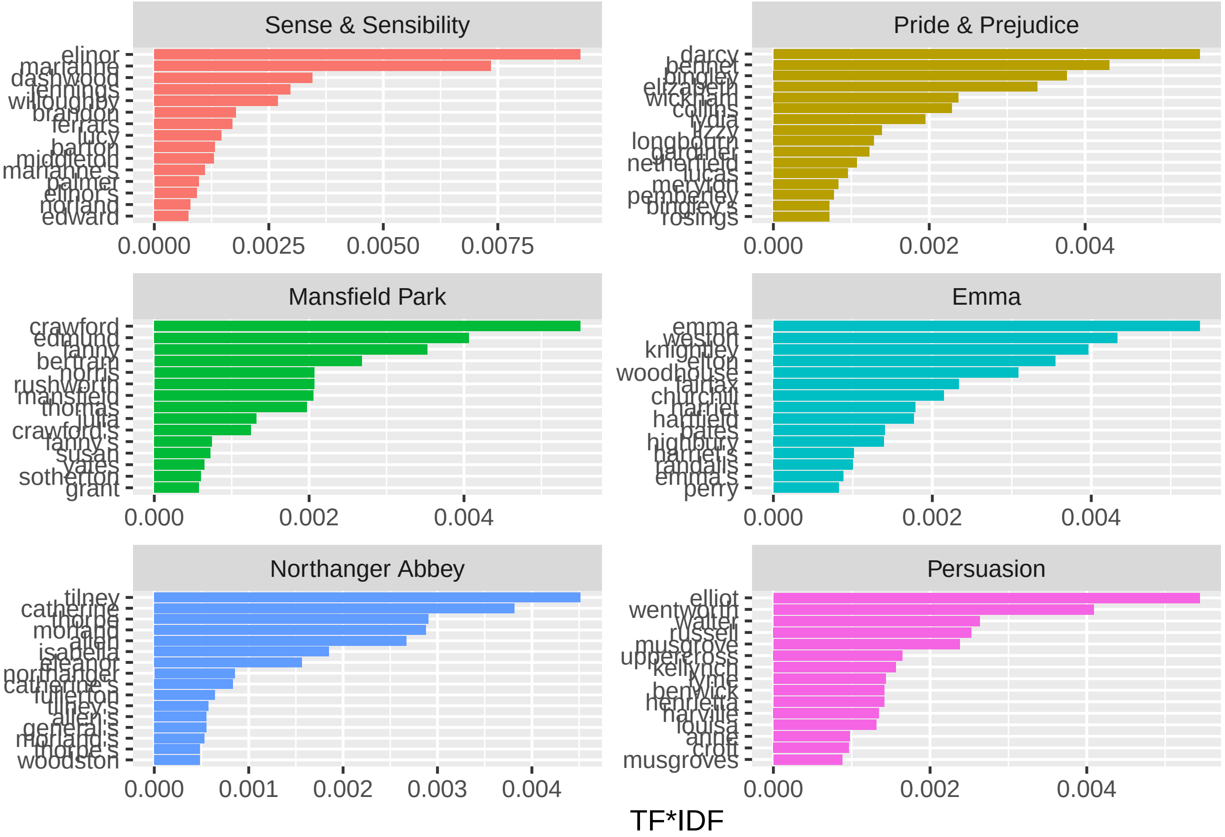
```
## # A tibble: 40,379 x 7
##    book                word      n  total     tf   idf tf_idf
##    <fct>               <chr> <int>  <int>  <dbl> <dbl>  <dbl>
##  1 Mansfield Park      the    6206 160460 0.0387     0      0
##  2 Mansfield Park      to     5475 160460 0.0341     0      0
##  3 Mansfield Park      and    5438 160460 0.0339     0      0
##  4 Emma                to     5239 160996 0.0325     0      0
##  5 Emma                the    5201 160996 0.0323     0      0
##  6 Emma                and    4896 160996 0.0304     0      0
##  7 Mansfield Park      of     4778 160460 0.0298     0      0
##  8 Pride & Prejudice the    4331 122204 0.0354     0      0
##  9 Emma                of     4291 160996 0.0267     0      0
## 10 Pride & Prejudice to     4162 122204 0.0341     0      0
## # ... with 40,369 more rows
```

```
book_words %>%
  select(-total) %>%
  arrange(desc(tf_idf))
```

```
## # A tibble: 40,379 x 6
##    book                word           n      tf   idf  tf_idf
##    <fct>               <chr>      <int>   <dbl> <dbl>   <dbl>
##  1 Sense & Sensibility elinor       623 0.00519  1.79 0.00931
##  2 Sense & Sensibility marianne    492 0.00410  1.79 0.00735
##  3 Mansfield Park      crawford    493 0.00307  1.79 0.00551
##  4 Pride & Prejudice   darcy       373 0.00305  1.79 0.00547
##  5 Persuasion          elliot      254 0.00304  1.79 0.00544
##  6 Emma                emma        786 0.00488  1.10 0.00536
##  7 Northanger Abbey    tilney      196 0.00252  1.79 0.00452
##  8 Emma                weston      389 0.00242  1.79 0.00433
##  9 Pride & Prejudice   bennet      294 0.00241  1.79 0.00431
## 10 Persuasion          wentworth   191 0.00228  1.79 0.00409
## # ... with 40,369 more rows
```

```
book_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(book) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill = book)) +
    geom_col(show.legend = FALSE) +
    labs(x = NULL, y = 'TF*IDF') +
    facet_wrap(~book, ncol = 2, scales = 'free') +
    coord_flip()
```

```
## Selecting by tf_idf
```



```
library(gutenbergr)
physics <- gutenberg_download(c(37729, 14725, 13476, 5001),
                              meta_fields = 'author',
                              mirror ='http://eremita.di.uminho.pt/gutenberg/')
```

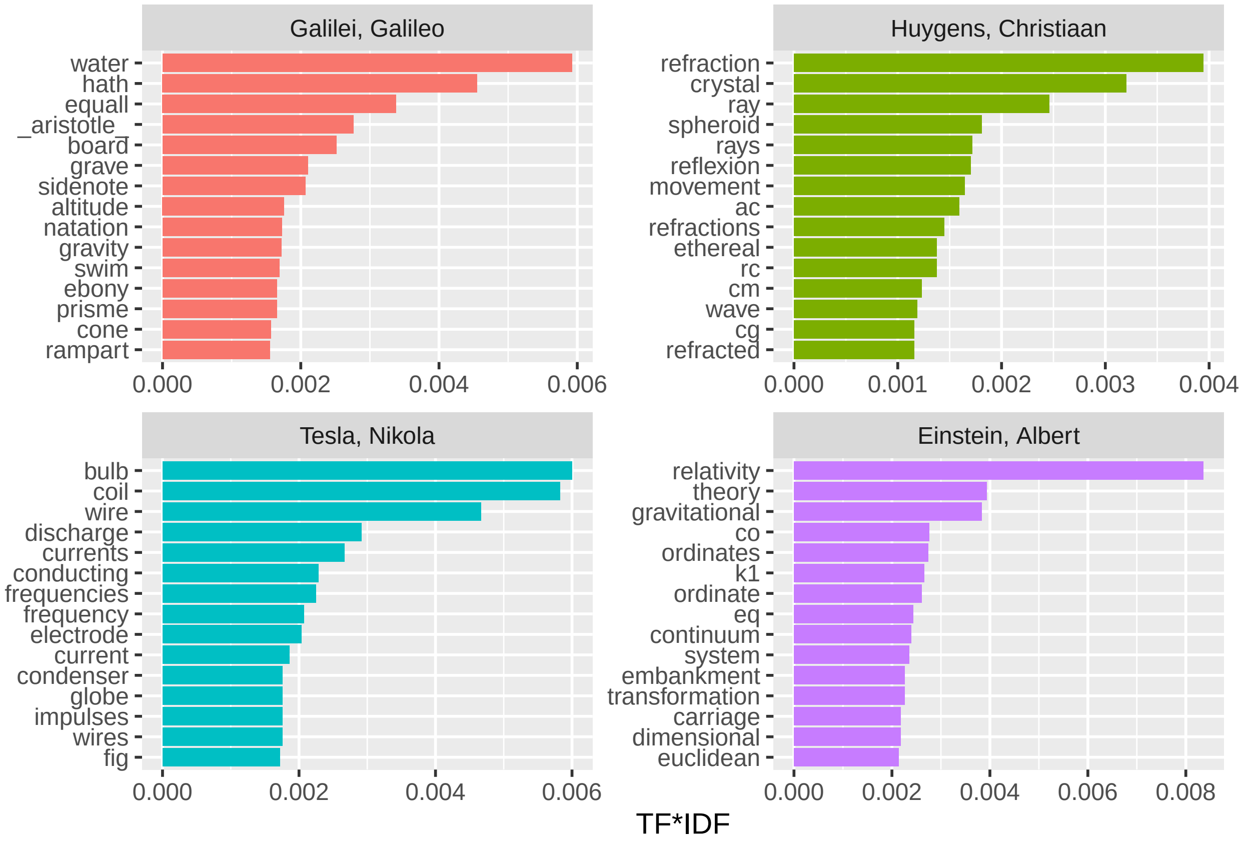
```
physics_word <- physics %>%
  unnest_tokens(word, text) %>%
  count(author, word, sort = TRUE) %>%
  ungroup()

physics_word
```

```
## # A tibble: 12,592 x 3
##    author              word      n
##    <chr>               <chr> <int>
##  1 Galilei, Galileo    the    3760
##  2 Tesla, Nikola       the    3604
##  3 Huygens, Christiaan the    3553
##  4 Einstein, Albert    the    2994
##  5 Galilei, Galileo    of     2049
##  6 Einstein, Albert    of     2030
##  7 Tesla, Nikola       of     1737
##  8 Huygens, Christiaan of     1708
##  9 Huygens, Christiaan to     1207
## 10 Tesla, Nikola       a      1176
## # ... with 12,582 more rows
```

```
plot_physics <- physics_word %>%
  bind_tf_idf(word, author, n) %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  mutate(author = factor(author, levels = c('Galilei, Galileo',
                                            'Huygens, Christiaan',
                                            'Tesla, Nikola',
                                            'Einstein, Albert')))

plot_physics %>%
  group_by(author) %>%
  top_n(15, tf_idf) %>%
  ungroup() %>%
  mutate(word = reorder(word, tf_idf)) %>%
  ggplot(aes(word, tf_idf, fill = author)) +
    geom_col(show.legend = FALSE) +
    labs(x = NULL, y = 'TF*IDF') +
    facet_wrap(~author, ncol = 2, scales = 'free') +
    coord_flip()
```



```r
library(stringr)

physics %>%
  dplyr::filter(str_detect(text, "eq\\.")) %>%
  select(text)
```

```
## # A tibble: 55 x 1
##    text
##    <chr>
##  1 "                         eq. 1: file eq01.gif"
##  2 "                         eq. 2: file eq02.gif"
##  3 "                         eq. 3: file eq03.gif"
##  4 "                         eq. 4: file eq04.gif"
##  5 "                       eq. 05a: file eq05a.gif"
##  6 "                       eq. 05b: file eq05b.gif"
##  7 "the distance between the points being eq. 06 ."
##  8 "direction of its length with a velocity v is eq. 06 of a metre."
##  9 "velocity v=c we should have eq. 06a ,"
## 10 "the rod as judged from K1 would have been eq. 06 ;"
## # ... with 45 more rows
```

```r
physics %>%
  dplyr::filter(str_detect(text, "K1")) %>%
  select(text)
```

```
## # A tibble: 59 x 1
##    text
##    <chr>
##  1 to a second co-ordinate system K1 provided that the latter is
##  2 condition of uniform motion of translation. Relative to K1 the
##  3 tenet thus: If, relative to K, K1 is a uniformly moving co-ordinate
##  4 with respect to K1 according to exactly the same general laws as with
##  5 does not hold, then the Galileian co-ordinate systems K, K1, K2, etc.,
##  6 Relative to K1, the same event would be fixed in respect of space and
##  7 to K1, when the magnitudes x, y, z, t, of the same event with respect
##  8 of light (and of course for every ray) with respect to K and K1. For
##  9 reference-body K and for the reference-body K1. A light-signal is sent
## 10 immediately follows. If referred to the system K1, the propagation of
## # ... with 49 more rows
```

```
physics %>%
  dplyr::filter(str_detect(text, "AK")) %>%
  select(text)
```

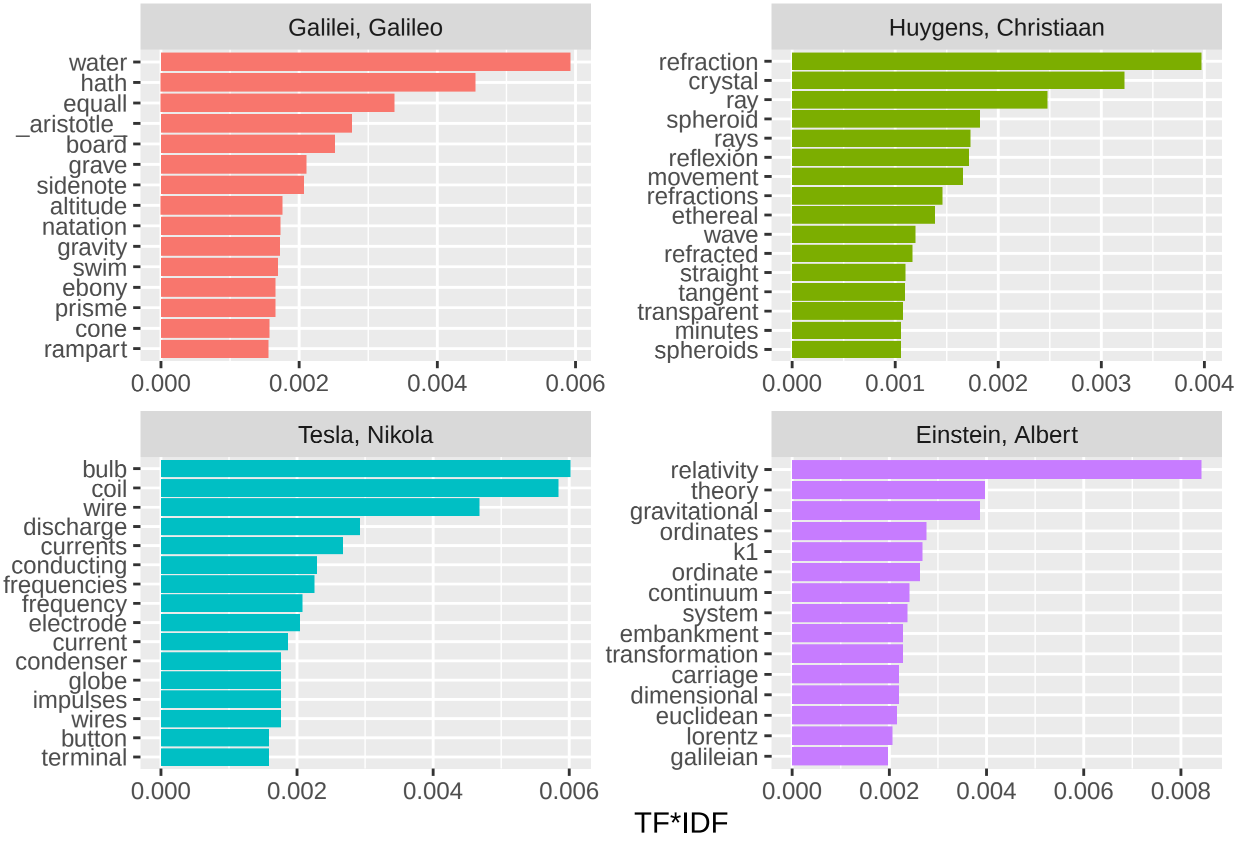
```
## # A tibble: 34 x 1
##    text
##    <chr>
##  1 Now let us assume that the ray has come from A to C along AK, KC; the
##  2 be equal to the time along KMN. But the time along AK is longer than
##  3 that along AL: hence the time along AKN is longer than that along ABC.
##  4 And KC being longer than KN, the time along AKC will exceed, by as
##  5 line which is comprised between the perpendiculars AK, BL. Then it
##  6 ordinary refraction. Now it appears that AK and BL dip down toward the
##  7 side where the air is less easy to penetrate: for AK being longer than
##  8 than do AK, BL. And this suffices to show that the ray will continue
##  9 surface AB at the points AK_k_B. Then instead of the hemispherical
## 10 along AL, LB, and along AK, KB, are always represented by the line AH,
## # ... with 24 more rows
```

```
mystopwords <- data_frame(word = c('eq', 'co', 'rc', 'ac', 'ak', 'bn',
                                   'fig', 'file', 'cg', 'cb', 'cm'))
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
physics_word <- anti_join(physics_word, mystopwords, by = 'word')
plot_physics <- physics_word %>%
  bind_tf_idf(word, author, n) %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(author) %>%
  top_n(15, tf_idf) %>%
  ungroup %>%
  mutate(author = factor(author, levels = c('Galilei, Galileo',
                                            'Huygens, Christiaan',
                                            'Tesla, Nikola',
                                            'Einstein, Albert')))

ggplot(plot_physics, aes(word, tf_idf, fill = author)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "TF*IDF") +
  facet_wrap(~author, ncol = 2, scales = 'free') +
coord_flip()
```



Galilei, Galileo
water
hath
equall
aristotle
board
grave
sidenote
altitude
natation
gravity
swim
ebony
prisme
cone
rampart
0.000
0.002
0.004
0.006
Huygens, Christiaan
refraction
crystal
ray
spheroid
rays
reflexion
movement
refractions
ethereal
wave
refracted
straight
tangent
transparent
minutes
spheroids
0.000
0.001
0.002
0.003
0.004
Tesla, Nikola
bulb
coil
wire
discharge
currents
conducting
frequencies
frequency
electrode
current
condenser
globe
impulses
wires
button
terminal
0.000
0.002
0.004
0.006
Einstein, Albert
relativity
theory
gravitational
ordinates
k1
ordinate
continuum
system
embankment
transformation
carriage
dimensional
euclidean
lorentz
galileian
0.000
0.002
0.004
0.006
0.008
TF*IDF