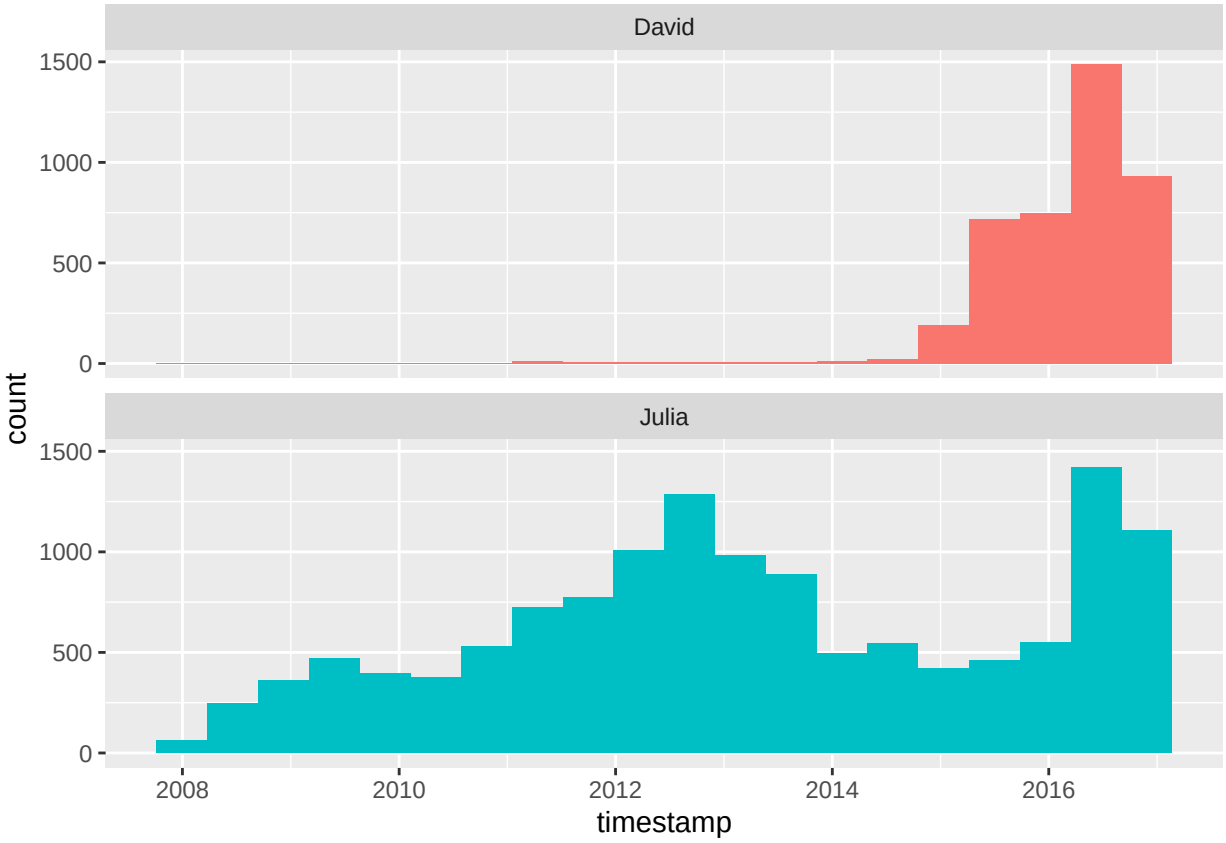
Chapter 07 exercise

John Peach

4/12/2021

```
tweets_julia <- read_csv("tweets_julia.csv")
tweets_dave <- read_csv("tweets_dave.csv")
tweets <- bind_rows(tweets_julia %>%
                      mutate(person = "Julia"),
                    tweets_dave %>%
                      mutate(person = "David")) %>%
  mutate(timestamp = ymd_hms(timestamp))
```

```
ggplot(tweets, aes(x = timestamp, fill = person)) +
  geom_histogram(position = 'identity', bins = 20, show.legend = FALSE) +
  facet_wrap(~person, ncol = 1)
```




```
replace_reg1 <- "https://t.co/[A-Za-z\\d]+|"
replace_reg2 <- "http://[A-Za-z\\d]+|&amp;|&lt;|&gt;|RT|https"
replace_reg <- paste0(replace_reg1, replace_reg2)
unnest_reg <- "([^A-Za-z_\\d#@']|'(?![A-Za-z_\\d#@]))"
```

```
tidy_tweets <- tweets %>%
  dplyr::filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_replace_all(text, replace_reg, "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = unnest_reg) %>%
  dplyr::filter(!word %in% stop_words$word,
                str_detect(word, "[a-z]"))
```

```
frequency <- tidy_tweets %>%
  group_by(person) %>%
  count(word, sort = TRUE) %>%
  left_join(tidy_tweets %>%
              group_by(person) %>%
              summarise(total = n())) %>%
  mutate(freq = n / total)
frequency
```

```
## # A tibble: 20,736 x 5
## # Groups:   person [2]
##    person word               n total    freq
##    <chr>  <chr>          <int> <int>   <dbl>
##  1 Julia  time             584 74572 0.00783
##  2 Julia  @selkie1970      570 74572 0.00764
##  3 Julia  @skedman         531 74572 0.00712
##  4 Julia  day              467 74572 0.00626
##  5 Julia  baby             408 74572 0.00547
##  6 David  @hadleywickham   315 20161 0.0156
##  7 Julia  love             304 74572 0.00408
##  8 Julia  @haleynburke     299 74572 0.00401
##  9 Julia  house            289 74572 0.00388
## 10 Julia  morning          278 74572 0.00373
## # ... with 20,726 more rows
```

```
frequency <- frequency %>%
  select(person, word, freq) %>%
  spread(person, freq) %>%
  arrange(Julia, David)
frequency
```

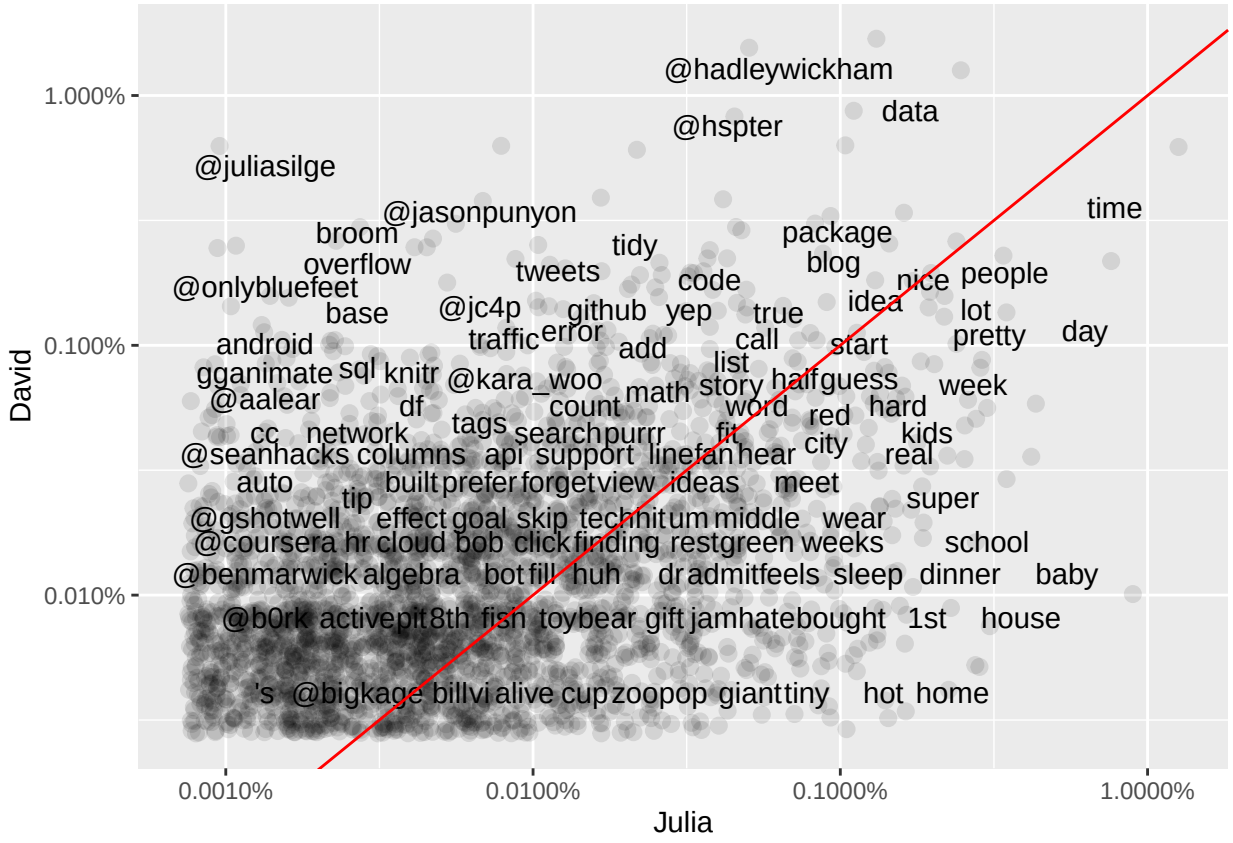
```
## # A tibble: 17,640 x 3
##    word                David     Julia
##    <chr>               <dbl>     <dbl>
##  1 's               0.0000496 0.0000134
##  2 @accidental__art 0.0000496 0.0000134
##  3 @alice_data      0.0000496 0.0000134
##  4 @alistaire       0.0000496 0.0000134
##  5 @corynissen      0.0000496 0.0000134
##  6 @jennybryan's    0.0000496 0.0000134
##  7 @jsvine          0.0000496 0.0000134
##  8 @lizasperling    0.0000496 0.0000134
##  9 @ognyanova       0.0000496 0.0000134
## 10 @rbloggers       0.0000496 0.0000134
## # ... with 17,630 more rows
```

```
ggplot(frequency, aes(Julia, David)) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.25, height = 0.25) +
```

```
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  geom_abline(color = 'red')
```

```
## Warning: Removed 14544 rows containing missing values (geom_point).
```

```
## Warning: Removed 14544 rows containing missing values (geom_text).
```




```
tidy_tweets <- tidy_tweets %>%
  dplyr::filter(timestamp >= as.Date("2016-01-01"),
                timestamp < as.Date("2017-01-01"))
```

```
word_ratio <- tidy_tweets %>%
  dplyr::filter(!str_detect(word, "^@")) %>%
  count(word, person) %>%
  dplyr::filter(sum(n) >= 10) %>%
  ungroup() %>%
  spread(person, n, fill = 0) %>%
  mutate_if(is.numeric, funs((. + 1) / sum(. + 1))) %>%
  mutate(logratio = log(David / Julia)) %>%
  arrange(desc(logratio))
```

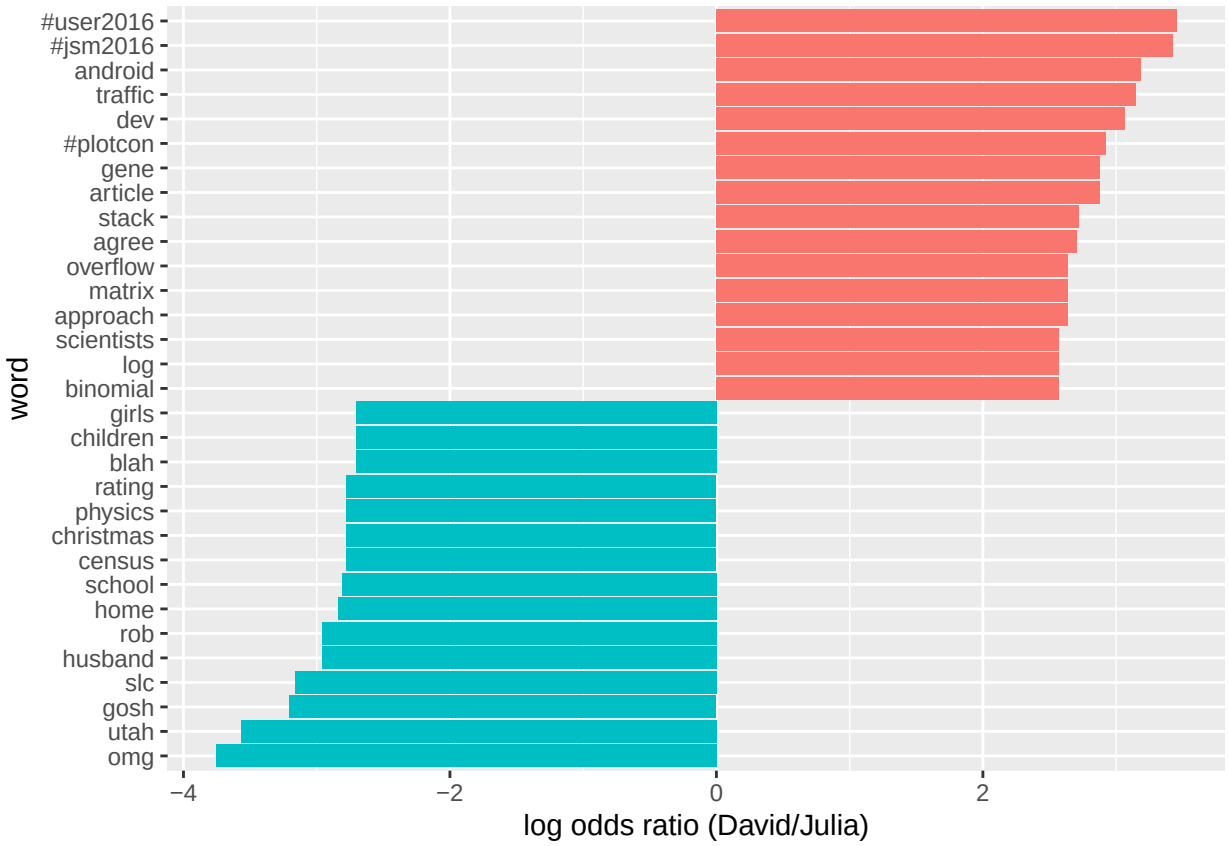
```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
```

```
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
word_ratio %>%
  arrange(abs(logratio))
```

```
## # A tibble: 6,688 x 4
##    word          David    Julia logratio
##    <chr>         <dbl>    <dbl>    <dbl>
##  1 idea       0.00129  0.00133   -0.0245
##  2 map        0.000619 0.000603   0.0263
##  3 science    0.00152  0.00157   -0.0313
##  4 email      0.000563 0.000543   0.0364
##  5 file       0.000563 0.000543   0.0364
##  6 names      0.00101  0.000965   0.0488
##  7 account    0.000450 0.000422   0.0645
##  8 api        0.000450 0.000422   0.0645
##  9 function   0.000900 0.000844   0.0645
## 10 population 0.000450 0.000422   0.0645
## # ... with 6,678 more rows
```

```
word_ratio %>%
  group_by(logratio < 0) %>%
  top_n(15, abs(logratio)) %>%
  ungroup() %>%
  mutate(word = reorder(word, logratio)) %>%
  ggplot(aes(word, logratio, fill = logratio < 0)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    ylab("log odds ratio (David/Julia)") +
    scale_fill_discrete(name = "", labels = c("David", "Julia"))
```




```
words_by_time <- tidy_tweets %>%
  dplyr::filter(!str_detect(word, "^@")) %>%
  mutate(time_floor = floor_date(timestamp, unit = '1 month')) %>%
  count(time_floor, person, word) %>%
  ungroup() %>%
  group_by(person, time_floor) %>%
  mutate(time_total = sum(n)) %>%
  group_by(word) %>%
  mutate(word_total = sum(n)) %>%
  ungroup() %>%
  rename(count = n) %>%
  dplyr::filter(word_total > 30)
words_by_time
```

```
## # A tibble: 970 x 6
##    time_floor          person word    count time_total word_total
##    <dttm>              <chr>  <chr>   <int>      <int>      <int>
##  1 2016-01-01 00:00:00 David  #rstats     2        307        324
##  2 2016-01-01 00:00:00 David  bad         1        307         33
##  3 2016-01-01 00:00:00 David  bit         2        307         45
##  4 2016-01-01 00:00:00 David  blog        1        307         60
##  5 2016-01-01 00:00:00 David  broom       2        307         41
##  6 2016-01-01 00:00:00 David  call        2        307         31
##  7 2016-01-01 00:00:00 David  check       1        307         42
##  8 2016-01-01 00:00:00 David  code        3        307         49
##  9 2016-01-01 00:00:00 David  data        2        307        276
## 10 2016-01-01 00:00:00 David  day         2        307         65
```

```
## # ... with 960 more rows
```

```
nested_data <- words_by_time %>%
  nest(-word, -person)

nested_data
```

```
## # A tibble: 112 x 3
##    person word    data
##    <chr>  <chr>   <list>
##  1 David  #rstats <tibble [12 x 4]>
##  2 David  bad     <tibble [9 x 4]>
##  3 David  bit     <tibble [10 x 4]>
##  4 David  blog    <tibble [12 x 4]>
##  5 David  broom   <tibble [10 x 4]>
##  6 David  call    <tibble [9 x 4]>
##  7 David  check   <tibble [12 x 4]>
##  8 David  code    <tibble [10 x 4]>
##  9 David  data    <tibble [12 x 4]>
## 10 David  day     <tibble [8 x 4]>
## # ... with 102 more rows
```

```
nested_models <- nested_data %>%
  mutate(models = map(data, ~glm(cbind(count, time_total) ~ time_floor, .,
                                 family = "binomial")))
nested_models
```

```
## # A tibble: 112 x 4
##    person word    data              models
##    <chr>  <chr>   <list>            <list>
##  1 David  #rstats <tibble [12 x 4]> <glm>
##  2 David  bad     <tibble [9 x 4]>  <glm>
##  3 David  bit     <tibble [10 x 4]> <glm>
##  4 David  blog    <tibble [12 x 4]> <glm>
##  5 David  broom   <tibble [10 x 4]> <glm>
##  6 David  call    <tibble [9 x 4]>  <glm>
##  7 David  check   <tibble [12 x 4]> <glm>
##  8 David  code    <tibble [10 x 4]> <glm>
##  9 David  data    <tibble [12 x 4]> <glm>
## 10 David  day     <tibble [8 x 4]>  <glm>
## # ... with 102 more rows
```

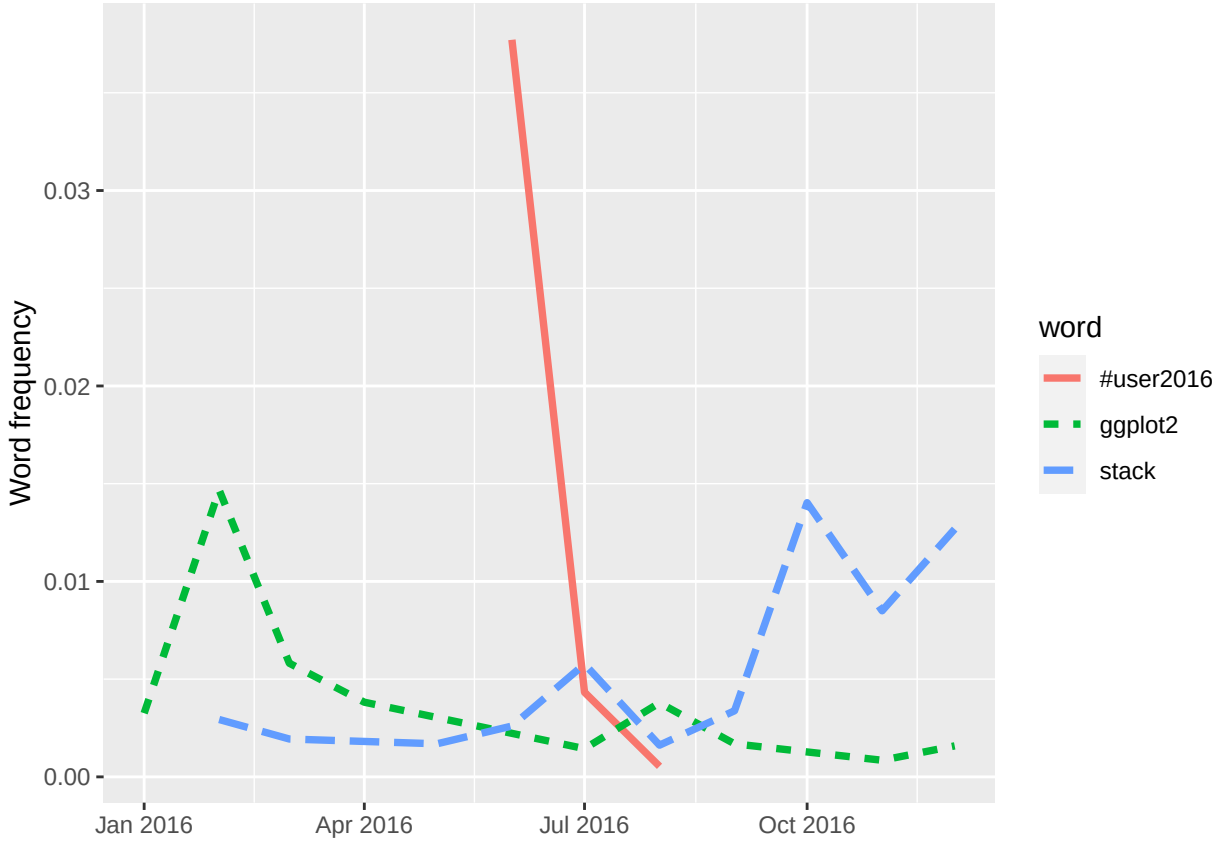
```
nested_models$temp <- map(nested_models$models, tidy)
nested_models %>%
  unnest(temp) %>%
  dplyr::filter(term == "time_floor") %>%
  mutate(adjusted.p.value = p.adjust(p.value)) -> slopes
```

```
top_slopes <- slopes %>%
  dplyr::filter(adjusted.p.value < 0.1) %>%
  select(-statistic, -p.value)
top_slopes
```

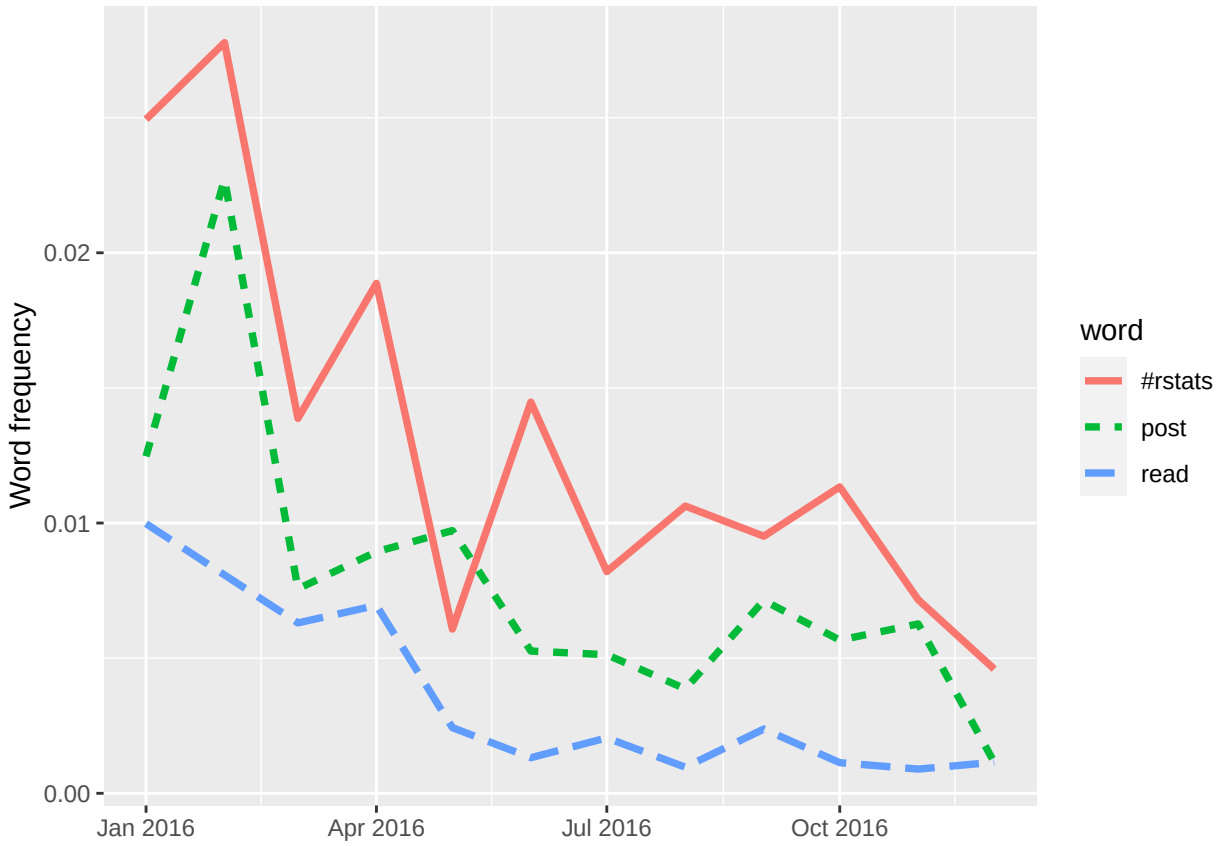
```
## # A tibble: 6 x 8
##   person word    data       models term     estimate  std.error adjusted.p.value
##   <chr>  <chr>   <list>     <list> <chr>       <dbl>      <dbl>             <dbl>
```

```
## 1 David  ggplot2 <tibble [~ <glm>  time_f~  -8.26e-8    1.97e-8        0.00300
## 2 Julia  #rstats <tibble [~ <glm>  time_f~  -4.50e-8    1.12e-8        0.00647
## 3 Julia  post    <tibble [~ <glm>  time_f~  -4.82e-8    1.45e-8        0.0978
## 4 Julia  read    <tibble [~ <glm>  time_f~  -9.33e-8    2.54e-8        0.0263
## 5 David  stack   <tibble [~ <glm>  time_f~   8.04e-8    2.19e-8        0.0263
## 6 David  #user2~ <tibble [~ <glm>  time_f~  -8.18e-7    1.55e-7        0.0000148
```

```
words_by_time %>%
  inner_join(top_slopes, by = c("word", "person")) %>%
  dplyr::filter(person == "David") %>%
  ggplot(aes(time_floor, count / time_total, colour = word, lty = word)) +
    geom_line(size = 1.3) +
    labs(x = NULL, y = "Word frequency")
```




```
words_by_time %>%
  inner_join(top_slopes, by = c("word", "person")) %>%
  dplyr::filter(person == "Julia") %>%
  ggplot(aes(time_floor, count / time_total, color = word, lty = word)) +
    geom_line(size = 1.3) +
    labs(x = NULL, y = "Word frequency")
```




```
tweets_julia <- read_csv("juliasilge_tweets.csv")
tweets_dave <- read_csv("drob_tweets.csv")
tweets <- bind_rows(tweets_julia %>%
                      mutate(person = "Julia"),
                    tweets_dave %>%
                      mutate(person = "David")) %>%
  mutate(created_at = ymd_hms(created_at))
```

```
tweets %>%
  select(-source) %>%
  dplyr::filter(!str_detect(text, "^(RT|@)")) %>%
  mutate(text = str_replace_all(text, replace_reg, "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = unnest_reg) %>%
  anti_join(stop_words)
```

```
## # A tibble: 11,078 x 6
##         id created_at          retweets favorites person word
##      <dbl> <dttm>                 <dbl>     <dbl> <chr>  <chr>
##  1 8.04e17 2016-12-01 16:44:03        0         0 Julia  score
##  2 8.04e17 2016-12-01 16:44:03        0         0 Julia  50
##  3 8.04e17 2016-12-01 16:42:03        0         9 Julia  snowing
##  4 8.04e17 2016-12-01 16:42:03        0         9 Julia  drinking
##  5 8.04e17 2016-12-01 16:42:03        0         9 Julia  tea
##  6 8.04e17 2016-12-01 16:42:03        0         9 Julia  #rstats
##  7 8.04e17 2016-12-01 02:56:10        0        11 Julia  julie
##  8 8.04e17 2016-12-01 02:56:10        0        11 Julia  helping
##  9 8.04e17 2016-12-01 02:56:10        0        11 Julia  python
```

```
## 10 8.04e17 2016-12-01 02:56:10        0        11 Julia  package
## # ... with 11,068 more rows
```

```
# This is broken as the new dataset is missing some columns. I fixed it so that
# it would run but it's output is garbage.
totals <- tidy_tweets %>%
  group_by(person, tweet_id) %>%
  summarise(rts = sum(in_reply_to_status_id)) %>%  # garbage
  group_by(person) %>%
  summarise(total_rts = n())

totals
```

```
## # A tibble: 2 x 2
##   person total_rts
##   <chr>      <int>
## 1 David       2128
## 2 Julia       2252
```

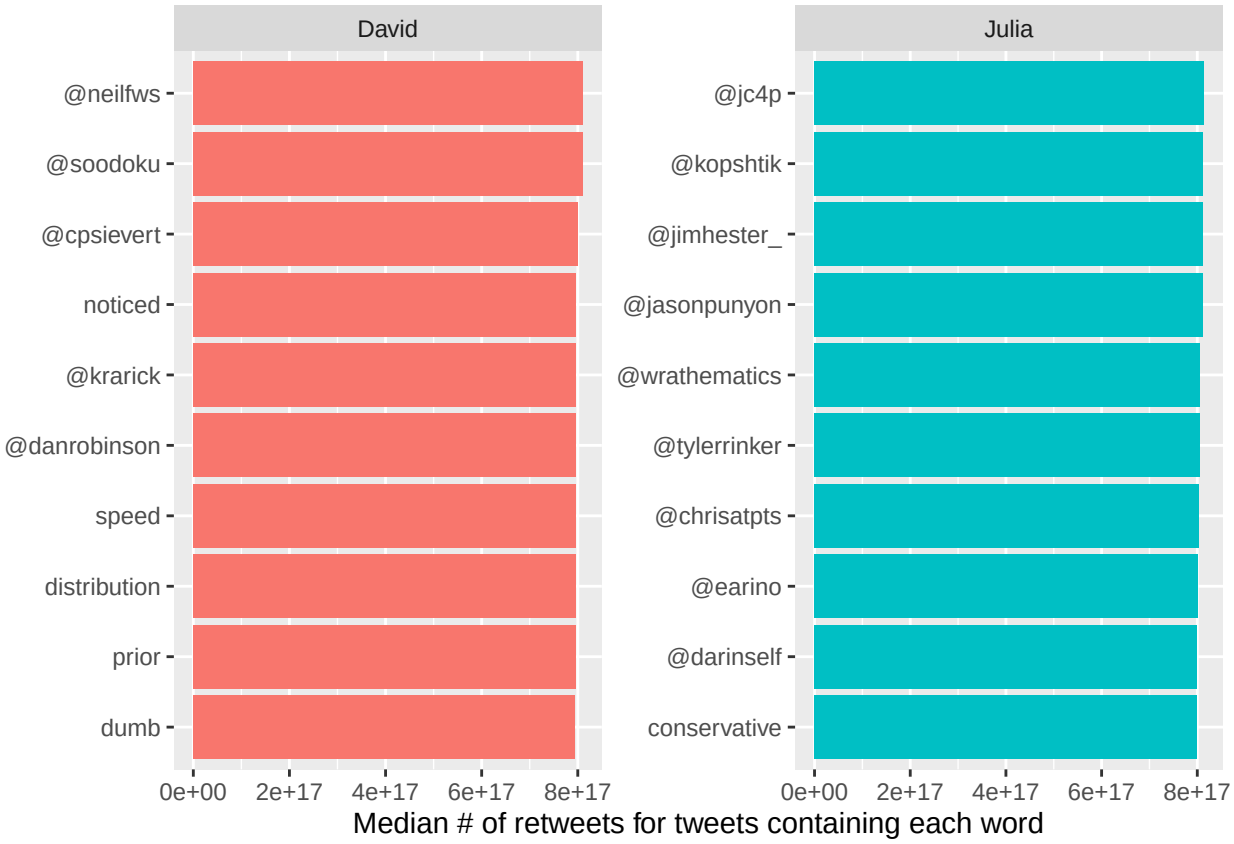
```
# This is broken as the new dataset is missing some columns. I fixed it so that
# it would run but it's output is garbage.
word_by_rts <- tidy_tweets %>%
  group_by(tweet_id, word, person) %>%
  summarise(rts = first(in_reply_to_status_id)) %>%  # garbage
  group_by(person, word) %>%
  summarise(retweets = median(rts), uses = n()) %>%
  left_join(totals) %>%
  dplyr::filter(retweets != 0) %>%
  ungroup()

word_by_rts %>%
  dplyr::filter(uses >= 5) %>%
  arrange(desc(retweets))
```

```
## # A tibble: 250 x 5
##    person word          retweets  uses total_rts
##    <chr>  <chr>            <dbl> <int>     <int>
##  1 Julia  @jc4p          8.13e17     5      2252
##  2 Julia  @kopshtik      8.13e17     5      2252
##  3 Julia  @jimhester_    8.13e17     9      2252
##  4 Julia  @jasonpunyon   8.12e17     5      2252
##  5 David  @neilfws       8.10e17     9      2128
##  6 David  @soodoku       8.10e17     8      2128
##  7 Julia  @wrathematics  8.05e17     9      2252
##  8 Julia  @tylerrinker   8.05e17    12      2252
##  9 Julia  @chrisatpts    8.04e17     6      2252
## 10 Julia  @earino        8.02e17    29      2252
## # ... with 240 more rows
```

```
word_by_rts %>%
  dplyr::filter(uses >= 5) %>%
  group_by(person) %>%
  top_n(10, retweets) %>%
  arrange(retweets) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
```

```
  ungroup() %>%
  ggplot(aes(word, retweets, fill = person)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~ person, scales = 'free', ncol = 2) +
    coord_flip() +
    labs(x = NULL,
         y = "Median # of retweets for tweets containing each word")
```




```
# This is broken as the new dataset is missing some columns. I fixed it so that
# it would run but it's output is garbage.
totals <- tidy_tweets %>%
  group_by(person, tweet_id) %>%
  summarise(favs = n()) %>%  # garbage line
  group_by(person) %>%
  summarise(total_favs = sum(favs))

word_by_favs <- tidy_tweets %>%
  group_by(tweet_id, word, person) %>%
  summarise(favs = first(in_reply_to_status_id)) %>%  #garbage line
  group_by(person, word) %>%
  summarise(favorites = median(favs), uses = n()) %>%
  left_join(totals) %>%
  dplyr::filter(favorites != 0) %>%
  ungroup()
```

```
word_by_favs %>%
  dplyr::filter(uses >= 5) %>%
```

```
  group_by(person) %>%
  top_n(10, favorites) %>%
  arrange(favorites) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
  ungroup() %>%
  ggplot(aes(word, favorites, fill = person)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~ person, scales = "free", ncol = 2) +
    coord_flip() +
    labs(x = NULL,
         y = "Median # of favorites for tweets containing each word")
```