07 Case study: comparing Twitter archives

H. David Shea

04 Aug 2021

Contents

Getting the data and distribution of tweets

```
tweets_julia <- read_csv("data/tweets_julia.csv")
tweets_dave <- read_csv("data/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) +
    theme_light()
```

Word frequencies

Cleaning up tweet into text only words

```
remove_reg <- "&amp;|&lt;|&gt;"
tidy_tweets <- tweets %>%
    filter(!str_detect(text, "^RT")) %>% # remove re-tweets
    mutate(text = str_remove_all(text, remove_reg)) %>% # remove hypertext characters
    unnest_tokens(word, text, token = "tweets") %>%
    filter(
```

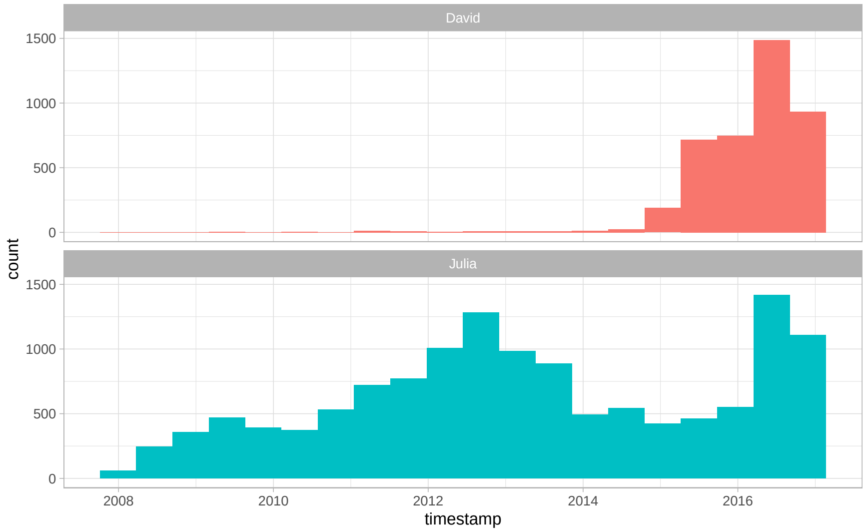

Figure 1: All tweets from the authors' (Julia Silge and David Robinson) accounts

```
        !word %in% stop_words$word,
        !word %in% str_remove_all(stop_words$word, "'"),
        str_detect(word, "[a-z]")
    )
```

Word frequencies

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

frequency %>%
    slice_head(n = 5) %>%
    kable()
```

person	word	n	total	freq
David	@hadleywickham	308	20699	0.0148799
David	#rstats	269	20699	0.0129958
David	@jennybryan	213	20699	0.0102904
David	@quominus	206	20699	0.0099522
David	@hspter	185	20699	0.0089376
Julia	@selkie1970	570	74152	0.0076869
Julia	time	557	74152	0.0075116
Julia	@skedman	531	74152	0.0071610
Julia	day	437	74152	0.0058933
Julia	baby	392	74152	0.0052864

```r
frequency <- frequency %>%
    select(person, word, freq) %>%
    pivot_wider(names_from = person, values_from = freq) %>%
    arrange(Julia, David)

frequency %>%
    slice_head(n = 10) %>%
    kable()
```

word	Julia	David
@accidentalart	1.35e-05	4.83e-05
@alicedata	1.35e-05	4.83e-05
@alistaire	1.35e-05	4.83e-05
@corynissen	1.35e-05	4.83e-05
@jennybryans	1.35e-05	4.83e-05
@jsvine	1.35e-05	4.83e-05
@lewislab	1.35e-05	4.83e-05
@lizasperling	1.35e-05	4.83e-05
@ognyanova	1.35e-05	4.83e-05
@rbloggers	1.35e-05	4.83e-05

```r
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") +
    theme_light()
#> Warning: Removed 18075 rows containing missing values (geom_point).
#> Warning: Removed 18075 rows containing missing values (geom_text).
```

Comparing word usage

Calculate the log odds ratio between David and Julia.

$$\text{log odds ratio} = \ln\left(\frac{\left[\frac{n+1}{\text{total}+1}\right]_{\text{David}}}{\left[\frac{n+1}{\text{total}+1}\right]_{\text{Julia}}}\right)$$

```r
tidy_tweets <- tidy_tweets %>% # Just look at 2016
    filter(timestamp >= as.Date("2016-01-01"),
           timestamp < as.Date("2017-01-01"))

word_ratios <- tidy_tweets %>%
```

Figure 2: Comparing the frequency of words used by Julia and David

```r
    filter(!str_detect(word, "^@")) %>%
    count(word, person) %>%
    group_by(word) %>%
    filter(sum(n) >= 10) %>%
    ungroup() %>%
    pivot_wider(names_from = person,
                values_from = n,
                values_fill = 0) %>%
    mutate_if(is.numeric, list( ~ (. + 1) / (sum(.) + 1))) %>%
    mutate(logratio = log(David / Julia)) %>%
    arrange(desc(logratio))

word_ratios %>%
    arrange(abs(logratio)) %>%
    slice_head(n = 10) %>%
    kable(caption = "Words about equally likely to come from David or Julia's account during 2016")
```

Table 3: Words about equally likely to come from David or Julia's account during 2016

word	David	Julia	logratio
words	0.0037651	0.0037777	-0.0033403
science	0.0065261	0.0064760	0.0077095
idea	0.0057731	0.0059363	-0.0278814
email	0.0025100	0.0024285	0.0330273
file	0.0025100	0.0024285	0.0330273
purrr	0.0025100	0.0024285	0.0330273
test	0.0022590	0.0021587	0.0454498
account	0.0020080	0.0018888	0.0611982
api	0.0020080	0.0018888	0.0611982
sad	0.0020080	0.0018888	0.0611982

```r
word_ratios %>%
    group_by(logratio < 0) %>%
    slice_max(abs(logratio), n = 15) %>%
    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")) +
    theme_light()
```

Changes in word use

Which words' frequencies have changed the fastest in the authors' Twitter feeds?

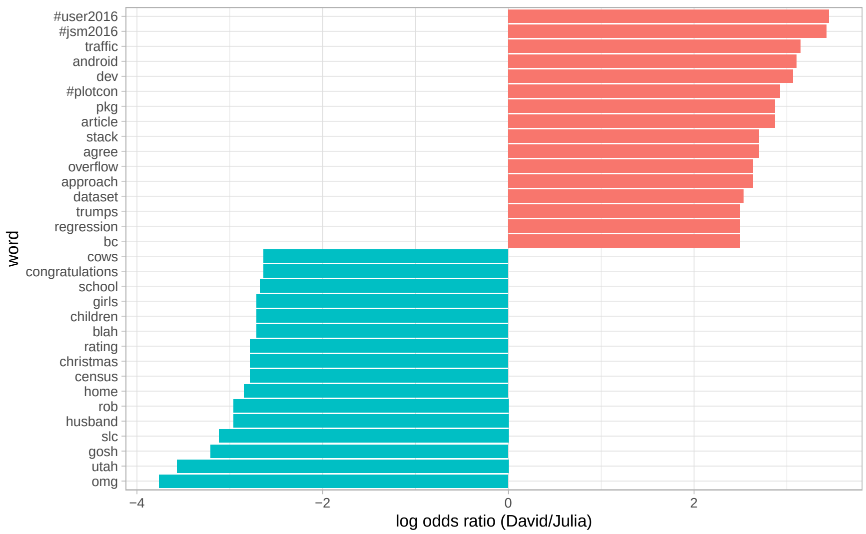

Figure 3: Comparing the odds ratios of words from the authors' accounts

```r
words_by_time <- tidy_tweets %>%
    filter(!str_detect(word, "^@")) %>% # remove user names
    mutate(time_floor = floor_date(timestamp, unit = "1 month")) %>% # measure monthly
    count(time_floor, person, word) %>%
    group_by(person, time_floor) %>%
    mutate(time_total = sum(n)) %>%
    group_by(person, word) %>%
    mutate(word_total = sum(n)) %>%
    ungroup() %>%
    rename(count = n) %>%
    filter(word_total > 30)

words_by_time %>%
    slice_head(n = 10) %>%
    kable(caption = "Data showing a person using a word in a given month")
```

Table 4: Data showing a person using a word in a given month

time_floor	person	word	count	time_total	word_total
2016-01-01	David	#rstats	2	315	205
2016-01-01	David	broom	2	315	34
2016-01-01	David	data	2	315	148
2016-01-01	David	ggplot2	1	315	37
2016-01-01	David	time	2	315	56
2016-01-01	David	tweets	1	315	46
2016-01-01	Julia	#rstats	10	437	116
2016-01-01	Julia	blog	2	437	33
2016-01-01	Julia	data	5	437	105
2016-01-01	Julia	day	1	437	43

"The `count` column tells us how many times that person used that word in that time bin, the `time_total` column tells us how many words that person used during that time bin, and the `word_total` column tells us how many times that person used that word over the whole year."

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

nested_data
#> # A tibble: 32 x 3
#>    person word    data
#>    <chr>  <chr>   <list>
#>  1 David  #rstats <tibble [12 x 4]>
#>  2 David  broom   <tibble [10 x 4]>
#>  3 David  data    <tibble [12 x 4]>
#>  4 David  ggplot2 <tibble [10 x 4]>
#>  5 David  time    <tibble [12 x 4]>
#>  6 David  tweets  <tibble [8 x 4]>
#>  7 Julia  #rstats <tibble [12 x 4]>
#>  8 Julia  blog    <tibble [10 x 4]>
#>  9 Julia  data    <tibble [12 x 4]>
#> 10 Julia  day     <tibble [12 x 4]>
#> # ... with 22 more rows

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

nested_models
#> # A tibble: 32 x 4
#>    person word    data              models
#>    <chr>  <chr>   <list>            <list>
#>  1 David  #rstats <tibble [12 x 4]> <glm>
#>  2 David  broom   <tibble [10 x 4]> <glm>
#>  3 David  data    <tibble [12 x 4]> <glm>
#>  4 David  ggplot2 <tibble [10 x 4]> <glm>
#>  5 David  time    <tibble [12 x 4]> <glm>
#>  6 David  tweets  <tibble [8 x 4]>  <glm>
#>  7 Julia  #rstats <tibble [12 x 4]> <glm>
#>  8 Julia  blog    <tibble [10 x 4]> <glm>
#>  9 Julia  data    <tibble [12 x 4]> <glm>
#> 10 Julia  day     <tibble [12 x 4]> <glm>
#> # ... with 22 more rows

slopes <- nested_models %>%
    mutate(models = map(models, tidy)) %>%
    unnest(cols = c(models)) %>%
    filter(term == "time_floor") %>%
    mutate(adjusted.p.value = p.adjust(p.value))

top_slopes <- slopes %>%
  filter(adjusted.p.value < 0.05)

top_slopes %>%
    select(-data) %>%
    kable(caption = "Words which have changed in frequency at a moderately significant level in the aut
```

Table 5: Words which have changed in frequency at a moderately significant level in the authors' tweets

person	word	term	estimate	std.error	statistic	p.value	adjusted.p.value
David	ggplot2	time_floor	-1e-07	0e+00	-4.044928	0.0000523	0.0016225
Julia	#rstats	time_floor	0e+00	0e+00	-4.037323	0.0000541	0.0016225
Julia	post	time_floor	-1e-07	0e+00	-3.457068	0.0005461	0.0158365
David	overflow	time_floor	1e-07	0e+00	3.119265	0.0018130	0.0489518
David	stack	time_floor	1e-07	0e+00	3.370386	0.0007506	0.0210176
David	#user2016	time_floor	-8e-07	2e-07	-5.266287	0.0000001	0.0000045

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

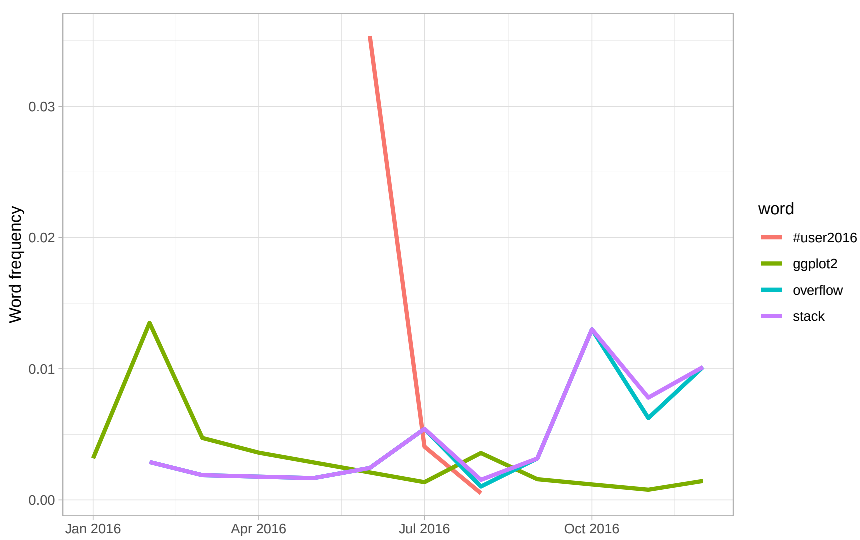

Figure 4: Trending words in David's tweets

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

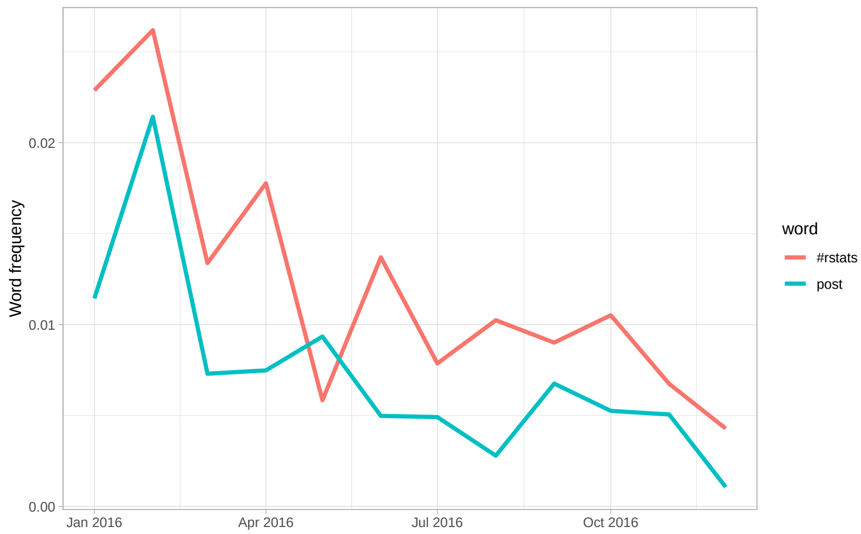

Figure 5: Trending words in Julia's tweets

Favorites and retweets

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

tidy_tweets <- tweets %>%
    filter(!str_detect(text, "^(RT|@)")) %>% # keep re-tweets and favorites
    mutate(text = str_remove_all(text, remove_reg)) %>% # remove hypertext characters
    unnest_tokens(word, text, token = "tweets", strip_url = TRUE) %>%
    filter(!word %in% stop_words$word,
           !word %in% str_remove_all(stop_words$word, "'"))

tidy_tweets %>%
    slice_head(n = 10) %>%
    kable()
```

id	created_at	source	retweets	favorites	person	word
8.043655e+17	2016-12-01 16:44:03	Twitter Web Client	0	0	Julia	score
8.043655e+17	2016-12-01 16:44:03	Twitter Web Client	0	0	Julia	50
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	snowing
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	drinking
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	tea
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	#rstats
8.043650e+17	2016-12-01 16:42:03	Twitter Web Client	0	9	Julia	

id	created_at	source	retweets	favorites	person	word
8.041571e+17	2016-12-01 02:56:10	Twitter Web Client	0	11	Julia	julie

```
totals <- tidy_tweets %>%
  group_by(person, id) %>%
  summarise(rts = first(retweets)) %>%
  group_by(person) %>%
  summarise(total_rts = sum(rts))

totals %>%
    kable()
```

person	total_rts
David	13014
Julia	1750

```
word_by_rts <- tidy_tweets %>%
    group_by(id, word, person) %>%
    summarise(rts = first(retweets)) %>%
    group_by(person, word) %>%
    summarise(retweets = median(rts), uses = n()) %>%
    left_join(totals) %>%
    filter(retweets != 0) %>%
    ungroup()

word_by_rts %>%
    filter(uses >= 5) %>%
    arrange(desc(retweets)) %>%
    slice_max(retweets, n = 10) %>%
    kable()
```

person	word	retweets	uses	total_rts
David	animation	85	5	13014
David	gganimate	75	6	13014
David	error	56	7	13014
David	start	56	6	13014
David	download	52	5	13014
Julia	tidytext	50	7	1750
David	introducing	45	6	13014
David	understanding	37	6	13014
David	ab	36	5	13014
David	bayesian	34	7	13014
David	modeling	34	5	13014
David	python	34	7	13014

```
word_by_rts %>%
    filter(uses >= 5) %>%
```

```
    group_by(person) %>%
    slice_max(retweets, n = 10) %>%
    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") +
    theme_light()
```

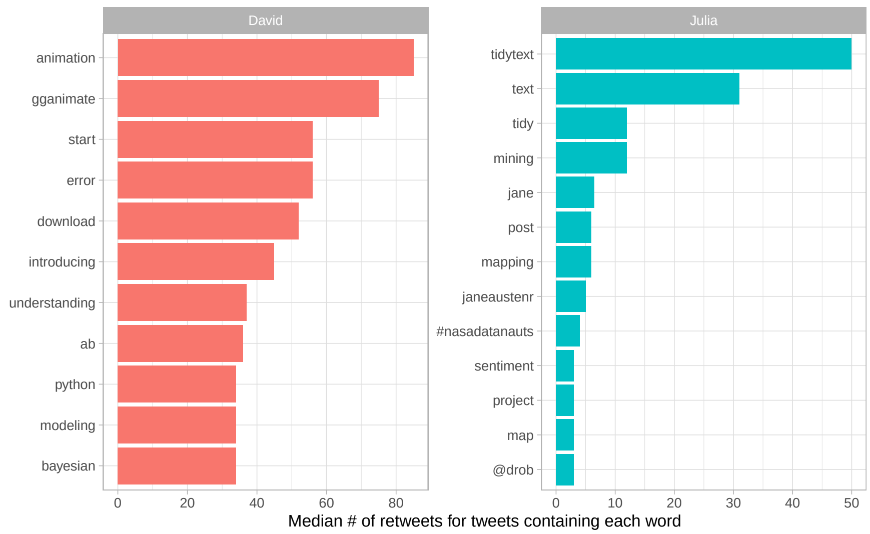

Figure 6: Words with highest median retweets

```
totals <- tidy_tweets %>%
    group_by(person, id) %>%
    summarise(favs = first(favorites)) %>%
    group_by(person) %>%
    summarise(total_favs = sum(favs))

word_by_favs <- tidy_tweets %>%
    group_by(id, word, person) %>%
    summarise(favs = first(favorites)) %>%
    group_by(person, word) %>%
    summarise(favorites = median(favs), uses = n()) %>%
    left_join(totals) %>%
    filter(favorites != 0) %>%
    ungroup()

word_by_favs %>%
    filter(uses >= 5) %>%
    group_by(person) %>%
```

```
    slice_max(favorites, n = 10) %>%
    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") +
    theme_light()
```

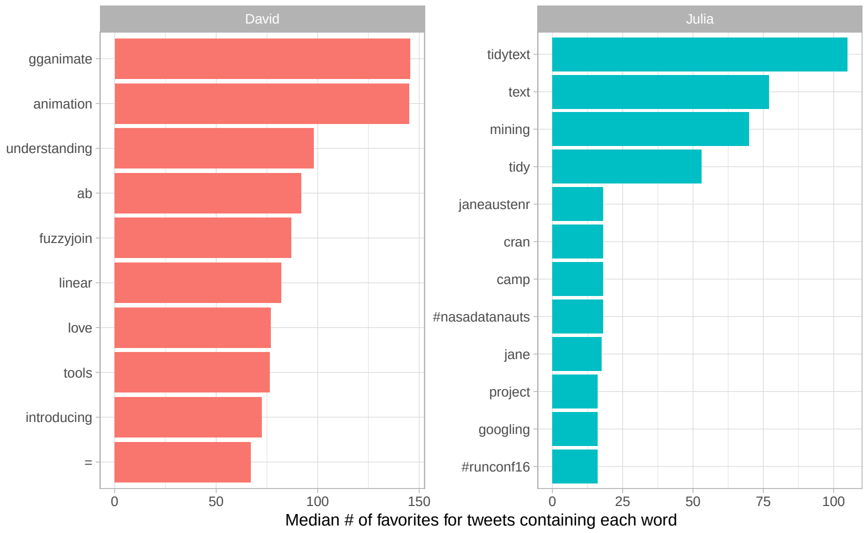

Figure 7: Words with highest median favorites

"In general, the same words that lead to retweets lead to favorites."