Text mining - Tweets by Donald Trump

R Cafe - Jonathan - j.debruin1@uu.nl

10/28/2019

Text Mining

The process of deriving high-quality information from text. (Wikipedia, 2019)

Applications

- Text categorization
- Entity extraction
- Document summarization
- Sentiment analysis

Text Mining with R

Text mining in R is challenging (without external tools).

> *We developed the tidytext R package because we were familiar with many methods for data wrangling and visualization, but couldn't easily apply these same methods to text. (Silge and Robinson 2016)*

- Text mining package `tidytext`
- Book "Text mining with R (Silge and Robinson, 2016)"
- www.tidytextmining.com/

Text Mining with R

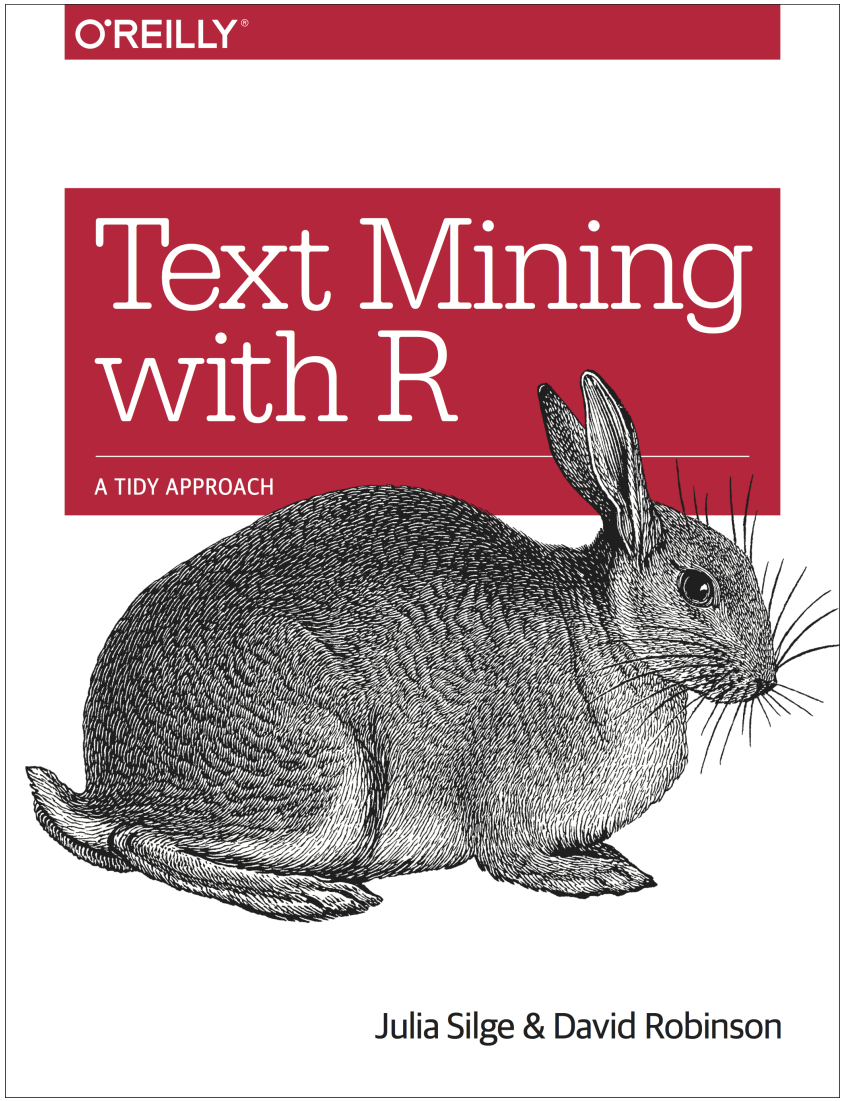


Figure 1:

Recap: Tidy data

Tidy data has a specific structure (Wickham 2014):

- Each variable is a column
- Each observation is a row
- Each type of observational unit is a table

Recap: Non-tidy data (iris)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Recap: Tidy data (iris)

```
##     id Species      measure value
## 1    1  setosa  Sepal.Width   3.5
## 2    1  setosa Sepal.Length   5.1
## 3    1  setosa  Petal.Width   0.2
## 4    1  setosa Petal.Length   1.4
## 5    2  setosa  Sepal.Width   3.0
## 6    2  setosa Sepal.Length   4.9
## 7    2  setosa  Petal.Width   0.2
## 8    2  setosa Petal.Length   1.4
## 9    3  setosa  Sepal.Width   3.2
## 10   3  setosa Sepal.Length   4.7
```

Tidy text

Definition: tidy text format is a table with ***one-token-per-row***

- A token is a meaningful unit of text, such as a word, that we are interested in using for analysis, and tokenization is the process of splitting text into tokens.

Packages for text mining

```
# default tidyverse packages
library(tidyverse)
library(lubridate)

# text mining related
library(tidytext)
library(textdata)
library(wordcloud)
```

Tweets by Donald Trump - load data

- http://trumptwitterarchive.com/
- https://github.com/mkearney/trumptweets/
- https://github.com/UtrechtUniversity/R-data-cafe/

```
tweets_trump <- read_csv(
  "https://raw.githubusercontent.com/UtrechtUniversity/R-data-cafe/m
)
```

```
## Parsed with column specification:
## cols(
##   status_id = col_double(),
##   created_at = col_datetime(format = ""),
##   text = col_character(),
##   favorite_count = col_double(),
##   retweet_count = col_double()
## )
```

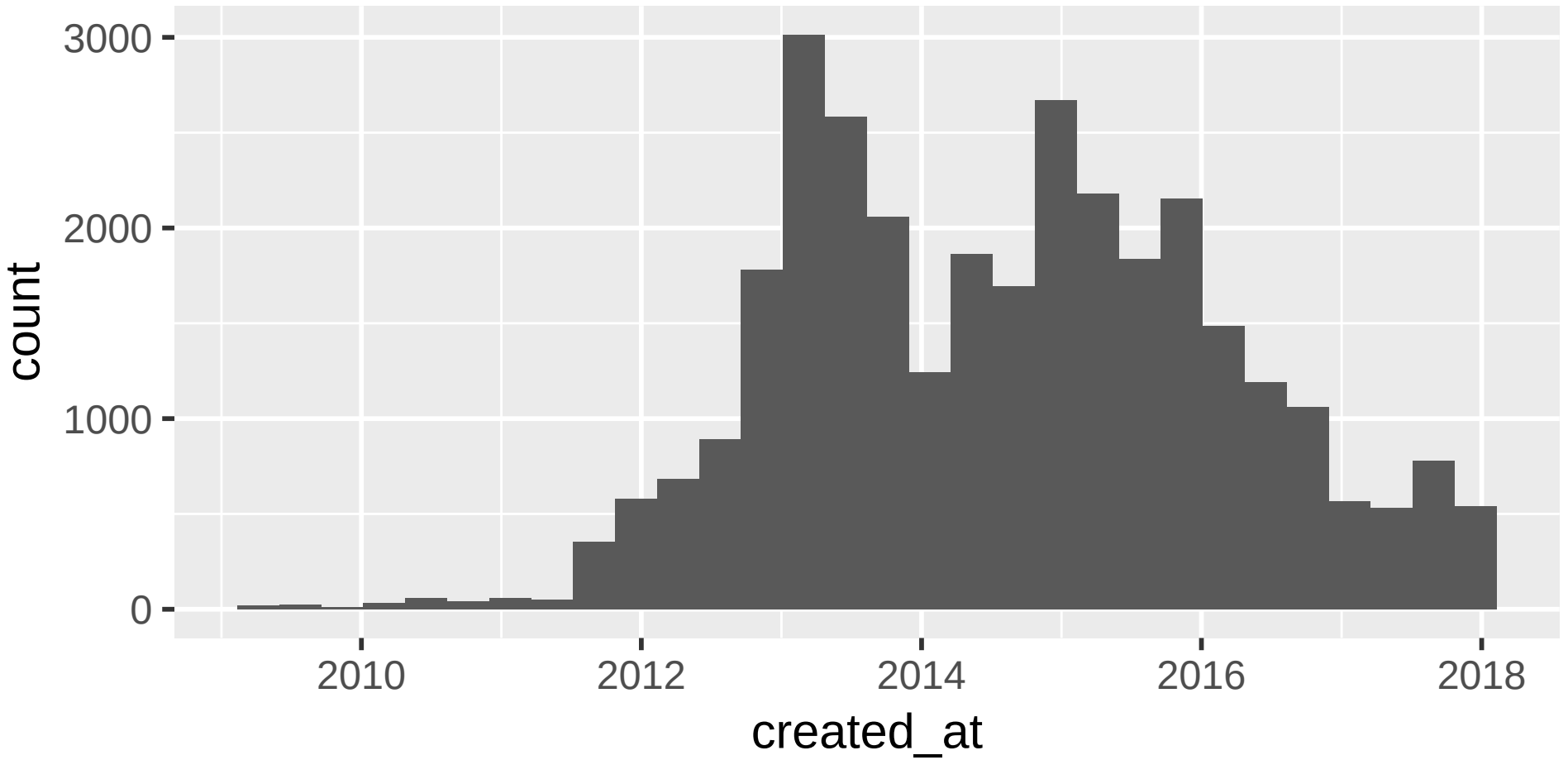
Tweets by Donald Trump - preview

```
head(tweets_trump)
```

```
## # A tibble: 6 x 5
##   status_id created_at          text              favorite_count re
##       <dbl> <dttm>              <chr>                      <dbl>
## 1   1.86e 9 2009-05-20 22:29:47 Read a great ~                11
## 2   9.27e15 2010-11-29 15:52:46 Congratulatio~                7
## 3   2.90e10 2010-10-28 18:53:40 I was on The ~                6
## 4   7.48e15 2010-11-24 17:20:54 Tomorrow nigh~               17
## 5   5.78e 9 2009-11-16 21:06:10 Donald Trump ~                3
## 6   1.48e16 2010-12-14 20:55:30 I'll be appea~               40
```

Tweets by Donald Trump - timeline

```
tweets_trump %>%
  ggplot(aes(created_at)) +
  geom_histogram()
```



Tweets by Donald Trump - tokenizing & tidy text

```
unnest_tokens(tweets_trump, word, text)
```

```
## # A tibble: 554,898 x 5
##     status_id created_at          favorite_count retweet_count word
##         <dbl> <dttm>                       <dbl>         <dbl> <chr>
##  1 1864367186 2009-05-20 22:29:47             11            11 read
##  2 1864367186 2009-05-20 22:29:47             11            11 a
##  3 1864367186 2009-05-20 22:29:47             11            11 great
##  4 1864367186 2009-05-20 22:29:47             11            11 interview
##  5 1864367186 2009-05-20 22:29:47             11            11 with
##  6 1864367186 2009-05-20 22:29:47             11            11 donald
##  7 1864367186 2009-05-20 22:29:47             11            11 trump
##  8 1864367186 2009-05-20 22:29:47             11            11 that
##  9 1864367186 2009-05-20 22:29:47             11            11 appeared
## 10 1864367186 2009-05-20 22:29:47             11            11 in
## # ... with 554,888 more rows
```

Tweets by Donald Trump - tokenizing & tidy text

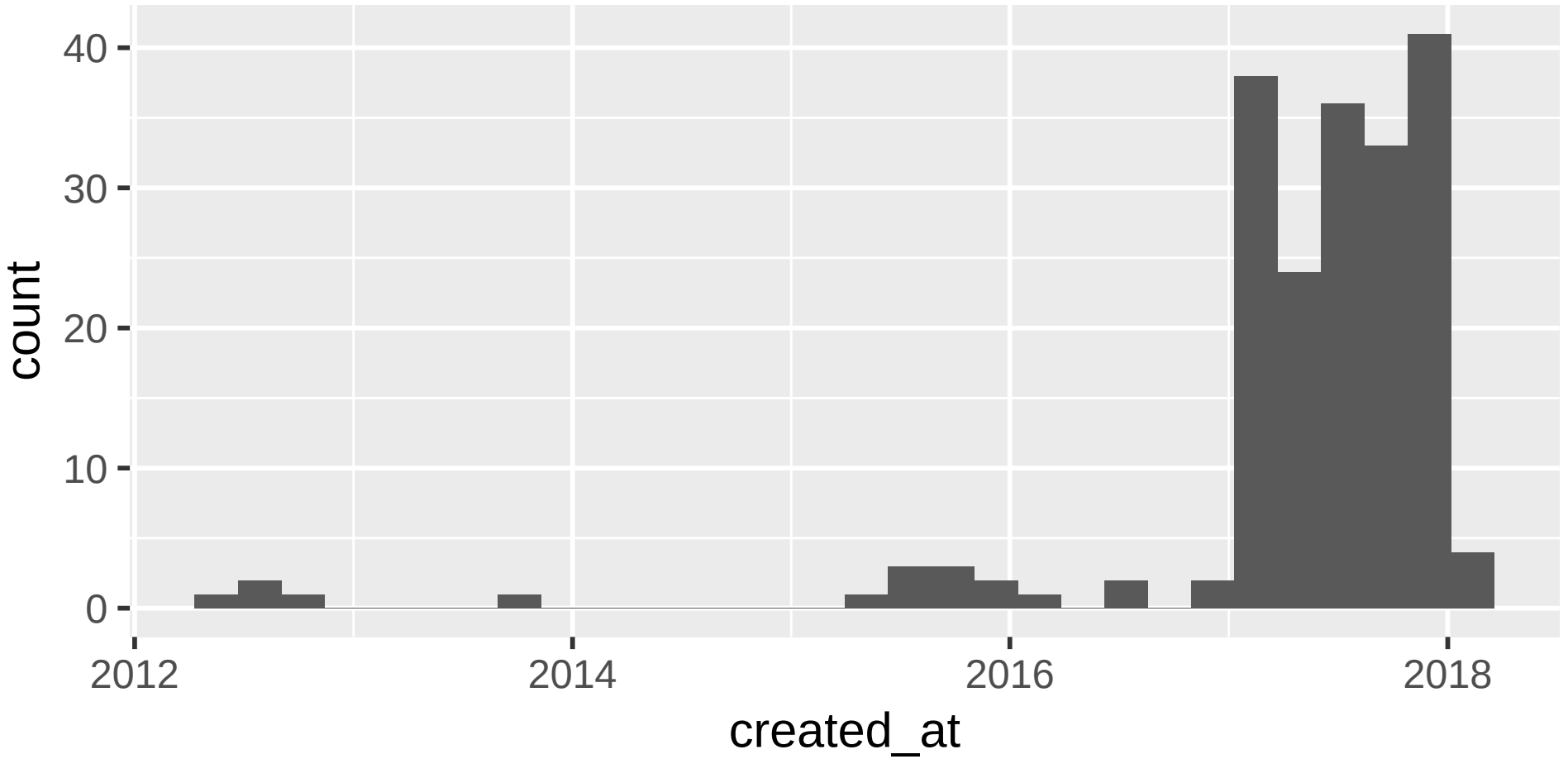
```
(tweets_trump_tokens <- unnest_tokens(tweets_trump, word, text, token="tweets"))
```

```
## Using `to_lower = TRUE` with `token = 'tweets'` may not preserve URLs.
```

```
## # A tibble: 532,698 x 5
##     status_id created_at          favorite_count retweet_count word
##         <dbl> <dttm>                       <dbl>         <dbl> <chr>
##  1 1864367186 2009-05-20 22:29:47             11            11 read
##  2 1864367186 2009-05-20 22:29:47             11            11 a
##  3 1864367186 2009-05-20 22:29:47             11            11 great
##  4 1864367186 2009-05-20 22:29:47             11            11 interview
##  5 1864367186 2009-05-20 22:29:47             11            11 with
##  6 1864367186 2009-05-20 22:29:47             11            11 donald
##  7 1864367186 2009-05-20 22:29:47             11            11 trump
##  8 1864367186 2009-05-20 22:29:47             11            11 that
##  9 1864367186 2009-05-20 22:29:47             11            11 appeared
## 10 1864367186 2009-05-20 22:29:47             11            11 in
## # ... with 532,688 more rows
```

Tweets by Donald Trump - timeline 'fake'

```
tweets_trump_tokens %>%
  filter(word == "fake") %>%
  ggplot(aes(created_at)) +
  geom_histogram()
```



Tweets by Donald Trump - Most common words

Use `count()`, a function from the `dplyr` package!

```
tweets_trump_tokens %>%
  count(word, sort = TRUE) %>%
  head(10)
```

```
## # A tibble: 10 x 2
##    word                n
##    <chr>           <int>
##  1 the             19136
##  2 to              11991
##  3 a                9368
##  4 is               8227
##  5 @realdonaldtrump 8095
##  6 and              8067
##  7 you              7770
##  8 in               7419
##  9 of               7364
## 10 i                6567
```

Tweets by Donald Trump - Filter stopwords

Use `anti_join()`, a function from the `dplyr` package!

```
data(stop_words)

tweets_trump_tokens %>% anti_join(stop_words, by="word")
```

```
## # A tibble: 259,964 x 5
##    status_id created_at          favorite_count retweet_count word
##        <dbl> <dttm>                       <dbl>         <dbl> <chr>
##  1   1.86e 9 2009-05-20 22:29:47             11            11 read
##  2   1.86e 9 2009-05-20 22:29:47             11            11 interview
##  3   1.86e 9 2009-05-20 22:29:47             11            11 donald
##  4   1.86e 9 2009-05-20 22:29:47             11            11 trump
##  5   1.86e 9 2009-05-20 22:29:47             11            11 appeared
##  6   1.86e 9 2009-05-20 22:29:47             11            11 york
##  7   1.86e 9 2009-05-20 22:29:47             11            11 times
##  8   1.86e 9 2009-05-20 22:29:47             11            11 magazine
##  9   1.86e 9 2009-05-20 22:29:47             11            11 http://tinyu~
## 10   9.27e15 2010-11-29 15:52:46              7            32 congratulati~
## # ... with 259,954 more rows
```

Tweets by Donald Trump - Filter stopwords

```
tweets_trump_tokens %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort = TRUE) %>%
  head(10)
```

```
## # A tibble: 10 x 2
##    word                n
##    <chr>           <int>
##  1 @realdonaldtrump  8095
##  2 trump             4223
##  3 amp               2644
##  4 president         1885
##  5 donald            1650
##  6 people            1495
##  7 america           1261
##  8 obama             1225
##  9 country           1101
## 10 time              1066
```

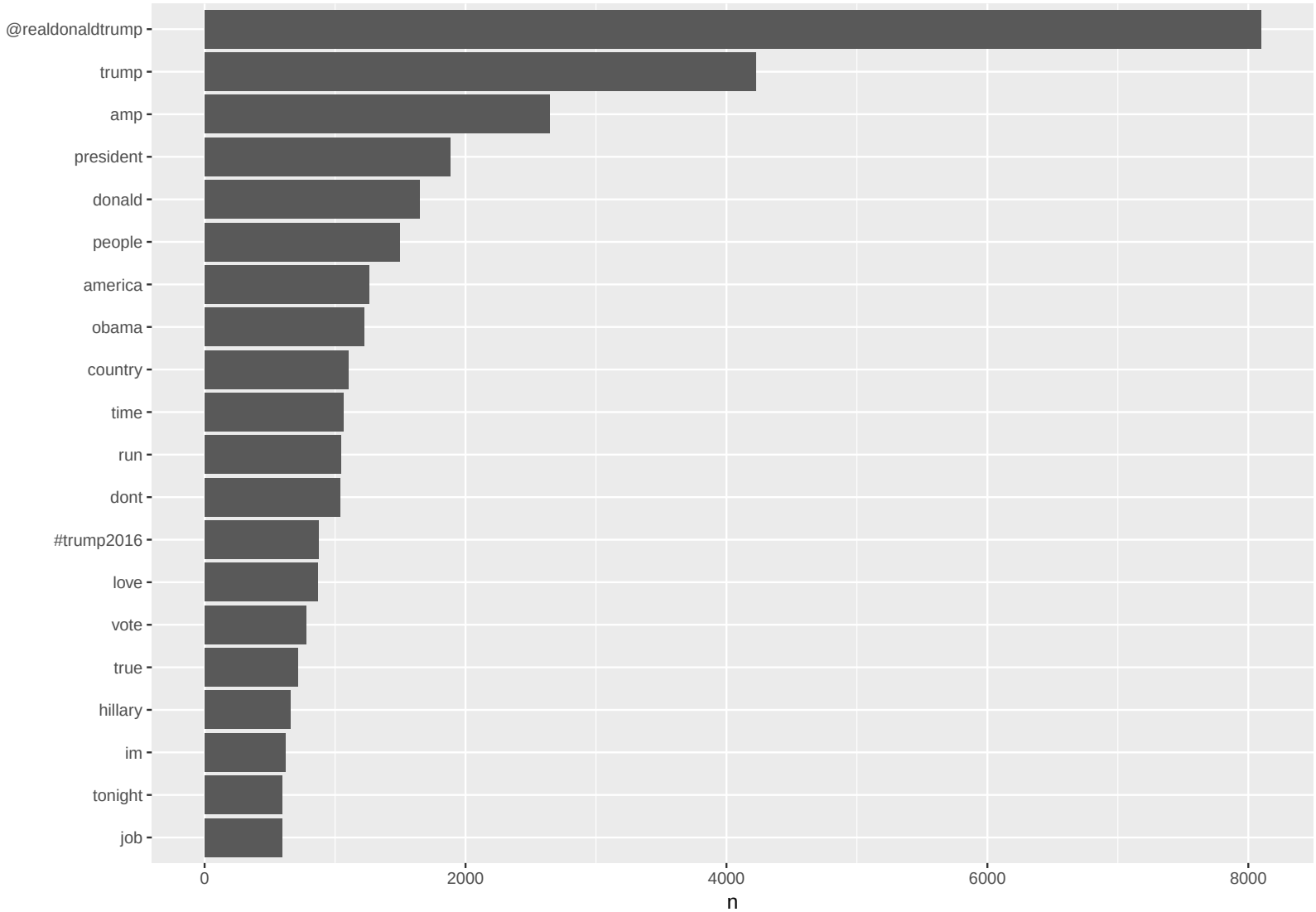
Tweets by Donald Trump - Filter stopwords

```
# top 20 non-stop words
tweets_trump_tokens %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort = TRUE) %>%
  head(20) %>%

  # trick to reorder factor (for plotting purposes)
  mutate(word = reorder(word, n)) %>%

  # create plot with ggplot
  ggplot(aes(word, n)) +
    geom_col() +
    xlab(NULL) +
    coord_flip()
```

Tweets by Donald Trump - Filter stopwords

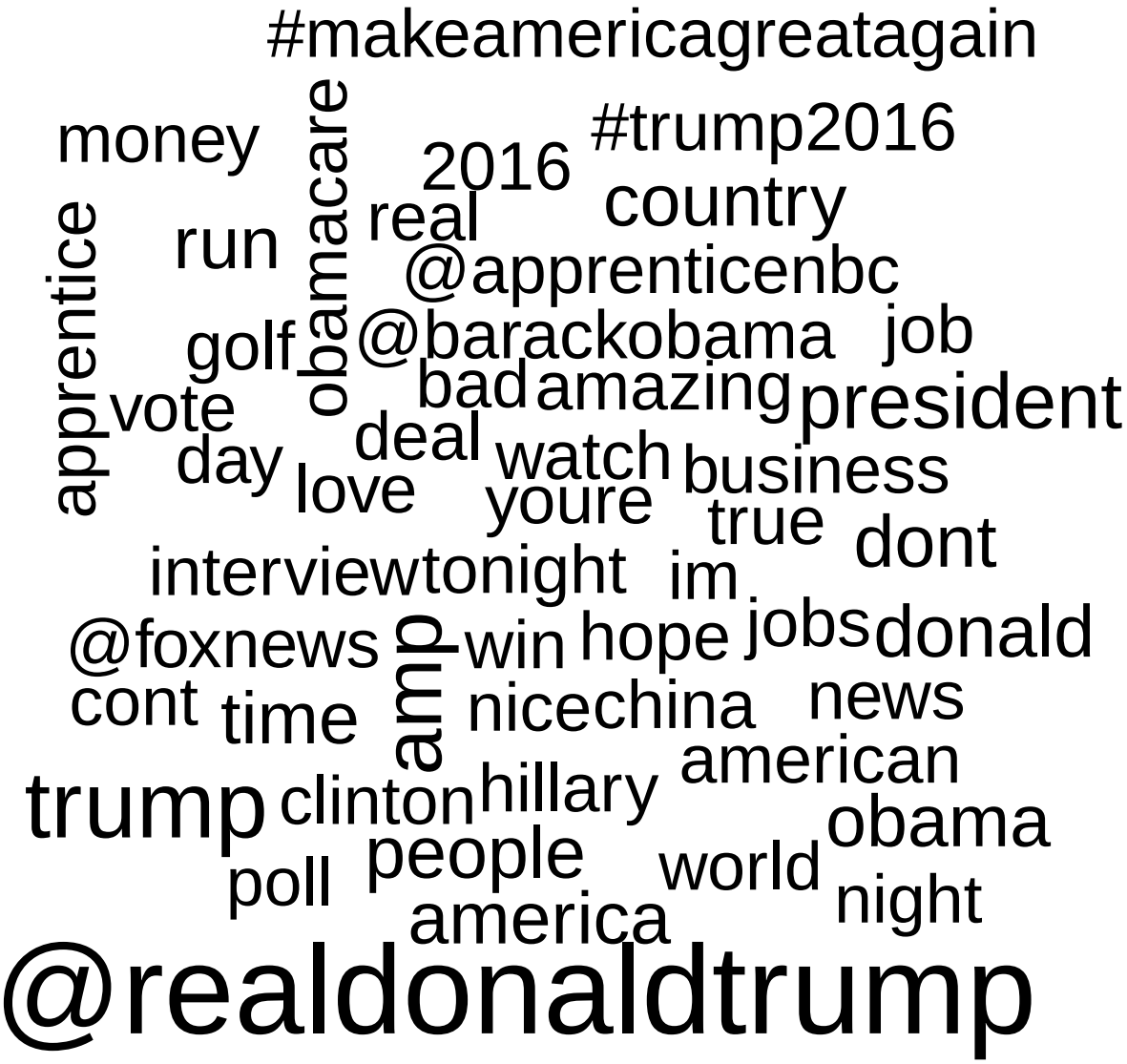


Tweets by Donald Trump - Word cloud

```
library(wordcloud)

tweets_trump_tokens %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort = TRUE) %>%
  # apply wordcloud to our data
  with(wordcloud(word, n, scale=c(2, 1), max.words = 50))
```

Tweets by Donald Trump - Word cloud



Sentiment analysis

Systematically identify, extract, quantify, and study affective states and subjective information. (Wikipedia, 2019)

Sentiment libraries

- Avialable in package `textdata`.

```
get_sentiments("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
```

Sentiment libraries

- Not every English word is in the lexicons because many English words are pretty neutral.

```
get_sentiments("bing")
```

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
```

Sentiment libraries

- It is important to keep in mind that these methods do not take into account qualifiers before a word, such as in “no good” or “not true”

```
get_sentiments("nrc")
```

```
## # A tibble: 13,901 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
```

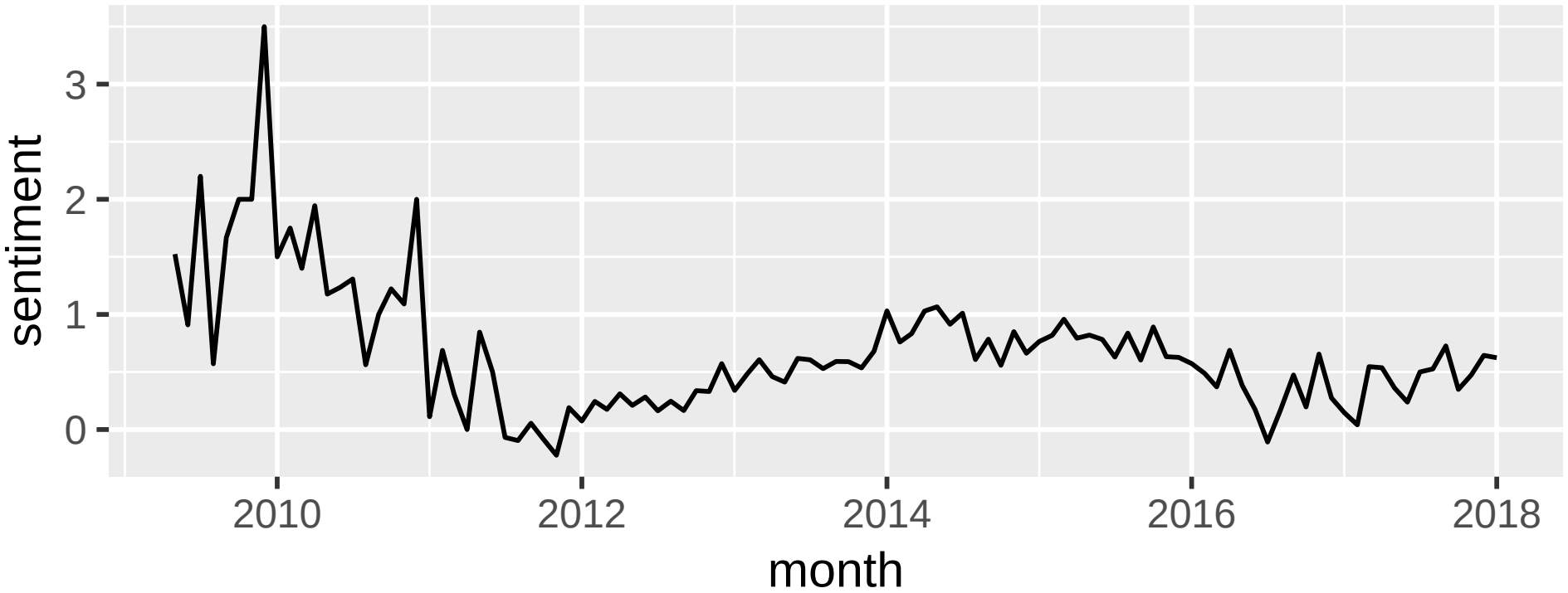
Tweets by Donald Trump - Sentiment analysis [bing]

```
(tweet_sentiment <- tweets_trump_tokens %>%
   # append score/value to each word (if and only if available)
   left_join(get_sentiments("bing"), by="word") %>%
   count(created_at, status_id, sentiment) %>%
   # untidy the dataset to compute the sentiment
   spread(sentiment, n, fill = 0) %>%
   # sentiment is number of positive words - negative words
   mutate(sentiment = positive - negative))
```

```
## # A tibble: 32,037 x 6
##    created_at          status_id negative positive `<NA>` sentiment
##    <dttm>                  <dbl>    <dbl>    <dbl>  <dbl>     <dbl>
##  1 2009-05-04 18:54:25 1698308935        0        2     21         2
##  2 2009-05-05 01:00:10 1701461182        0        3     19         3
##  3 2009-05-08 13:38:08 1737479987        1        2     13         1
##  4 2009-05-08 20:40:15 1741160716        0        0     13         0
##  5 2009-05-12 14:07:28 1773561338        0        1     19         1
##  6 2009-05-12 19:21:55 1776419923        1        1     17         0
##  7 2009-05-13 17:38:28 1786560616        0        3     13         3
##  8 2009-05-14 16:30:40 1796477499        0        2     12         2
##  9 2009-05-15 14:13:13 1806258917        0        2     10         2
## 10 2009-05-16 22:22:45 1820624395        0        1     14         1
## # ... with 32,027 more rows
```

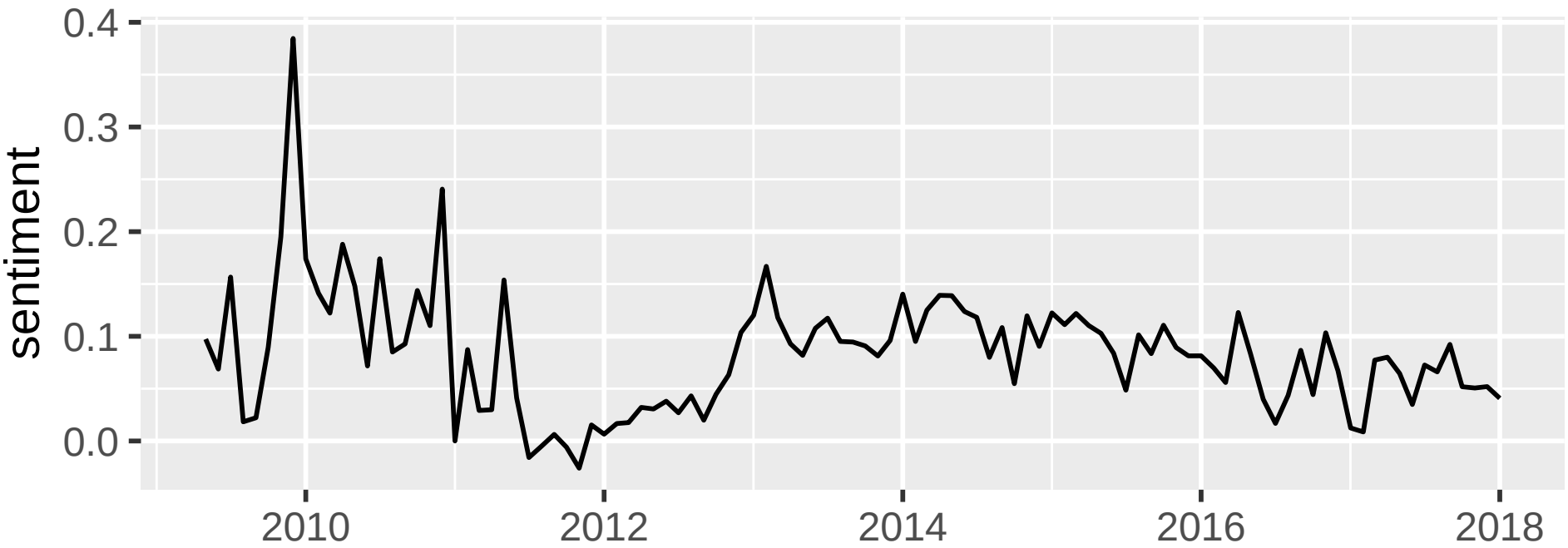
Tweets by Donald Trump - Sentiment analysis [bing]

```
tweet_sentiment %>%
  group_by(month=floor_date(created_at, "month")) %>%
  summarize(sentiment=mean(sentiment)) %>%
  ggplot(aes(month, sentiment)) +
    geom_line()
```



Tweets by Donald Trump - Sentiment analysis [afinn]

```
tweets_trump_tokens %>%
  left_join(get_sentiments("afinn"), by="word") %>%
  mutate(value = replace_na(value, 0)) %>%
  group_by(month=floor_date(created_at, "month")) %>%
  summarize(sentiment=mean(value)) %>%
  ggplot(aes(month, sentiment)) +
    geom_line()
```



Questions?

Thanks for attending.

R Cafe 15:00 - 17:00