Text, Tweets, and Sentiment

Part 3

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/04/10

Slides Updated: 2023-04-09

Returning to Trump

```
require(tidyverse)
tweet_words <- readRDS(file="../data/Trump_tweet_words.Rds")
tweet_words <- tweet_words %>% mutate(PostPresident =
Tweeting.date > as.Date('2016-11-06'))
```

Log-Odds

- **Odds**: Probability a word is used pre/post presidency
- **Log**: Useful for removing skew in data!
- Interactive code time!

Odds Step 1

```
(odds1 <- tweet_words %>%
  count(word, PostPresident) %>%
  filter(sum(n) >= 5) %>%
  spread(PostPresident, n, fill = 0) %>%
  ungroup() %>%
  mutate(totFALSE = sum(`FALSE`),
         totTRUE = sum(`TRUE`)))
```

```
## # A tibble: 23,453 × 5
##    word      `FALSE` `TRUE` totFALSE totTRUE
##    <chr>       <dbl>  <dbl>    <dbl>   <dbl>
##  1 aa              1      0   183927  114138
##  2 aaa            11      1   183927  114138
##  3 aand            0      1   183927  114138
##  4 aaron           2      0   183927  114138
##  5 aarons          1      0   183927  114138
##  6 ab              1      0   183927  114138
##  7 abandon         6      4   183927  114138
##  8 abandoned      15      8   183927  114138
##  9 abbas           0      2   183927  114138
## 10 abbott          1      1   183927  114138
```

Odds Step 2

```
(odds2 <- odds1 %>%
  mutate(propFALSE = (`FALSE` + 1) / (totFALSE + 1),
         propTRUE = (`TRUE` + 1) / (totTRUE + 1)))
```

```
## # A tibble: 23,453 × 7
##    word      `FALSE` `TRUE` totFALSE totTRUE propF…¹ propT…²
##    <chr>       <dbl>  <dbl>    <dbl>   <dbl>   <dbl>   <dbl>
##  1 aa              1      0   183927  114138 1.09e-5 8.76e-6
##  2 aaa            11      1   183927  114138 6.52e-5 1.75e-5
##  3 aand            0      1   183927  114138 5.44e-6 1.75e-5
##  4 aaron           2      0   183927  114138 1.63e-5 8.76e-6
##  5 aarons          1      0   183927  114138 1.09e-5 8.76e-6
##  6 ab              1      0   183927  114138 1.09e-5 8.76e-6
##  7 abandon         6      4   183927  114138 3.81e-5 4.38e-5
##  8 abandoned      15      8   183927  114138 8.70e-5 7.89e-5
##  9 abbas           0      2   183927  114138 5.44e-6 2.63e-5
## 10 abbott          1      1   183927  114138 1.09e-5 1.75e-5
## # … with 23,443 more rows, and abbreviated variable names
## #   ¹propFALSE, ²propTRUE
```

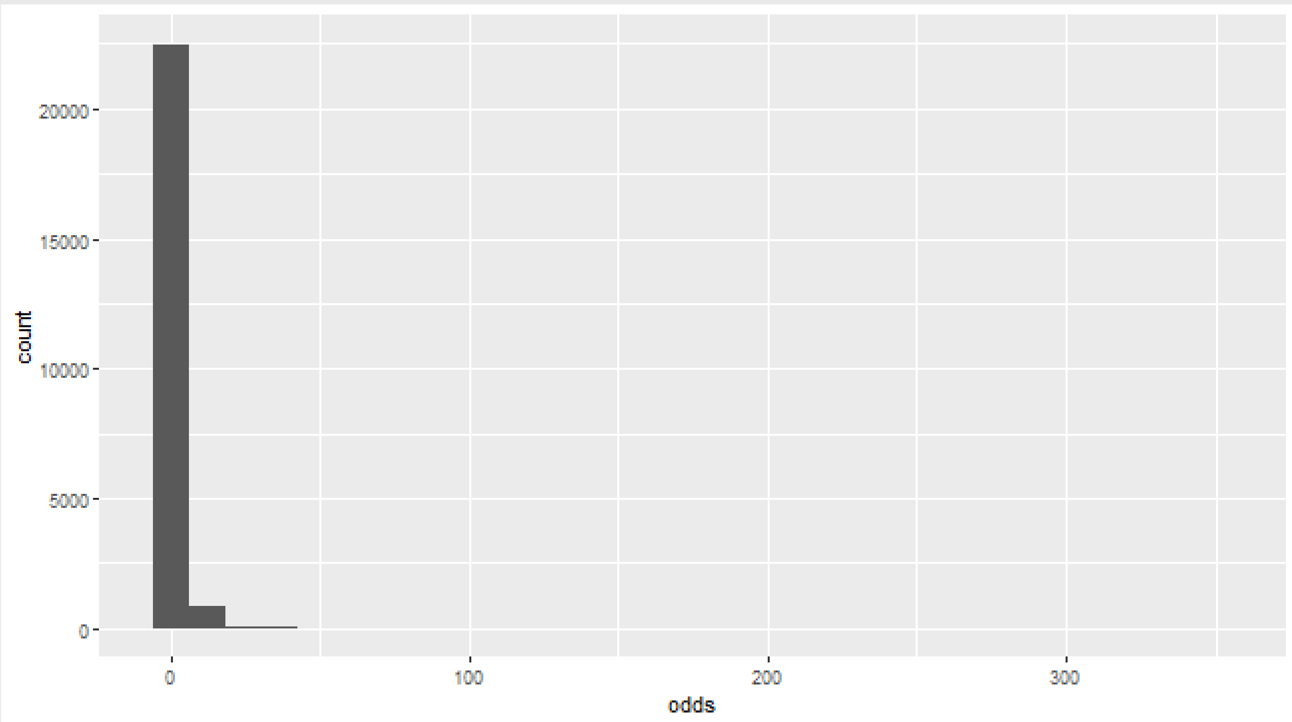
Odds Step 3

```
(odds3 <- odds2 %>%
  mutate(odds = propTRUE / propFALSE))
```

```
## # A tibble: 23,453 × 8
##    word      `FALSE` `TRUE` totFALSE totTRUE propF…¹ propT…²
##    <chr>       <dbl>  <dbl>    <dbl>   <dbl>   <dbl>   <dbl>
##  1 aa              1      0   183927  114138 1.09e-5 8.76e-6
##  2 aaa            11      1   183927  114138 6.52e-5 1.75e-5
##  3 aand            0      1   183927  114138 5.44e-6 1.75e-5
##  4 aaron           2      0   183927  114138 1.63e-5 8.76e-6
##  5 aarons          1      0   183927  114138 1.09e-5 8.76e-6
##  6 ab              1      0   183927  114138 1.09e-5 8.76e-6
##  7 abandon         6      4   183927  114138 3.81e-5 4.38e-5
##  8 abandoned      15      8   183927  114138 8.70e-5 7.89e-5
##  9 abbas           0      2   183927  114138 5.44e-6 2.63e-5
## 10 abbott          1      1   183927  114138 1.09e-5 1.75e-5
## # … with 23,443 more rows, 1 more variable: odds <dbl>, and
## #   abbreviated variable names ¹propFALSE, ²propTRUE
```

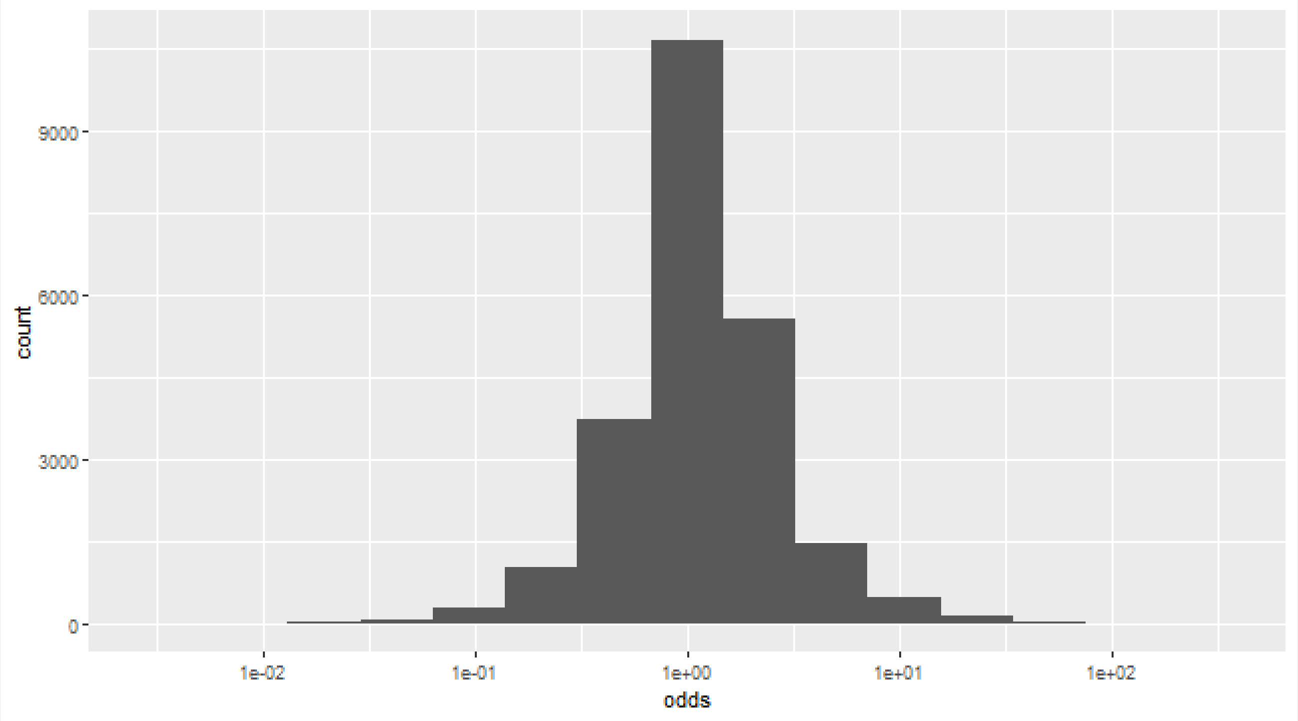
Why log?

```
odds3 %>%
  ggplot(aes(x = odds)) +
  geom_histogram()
```



Why log?

```
odds3 %>%
  ggplot(aes(x = odds)) +
  geom_histogram(bins = 15) +
  scale_x_log10()
```



Odds Step 4

```
(prepost_logodds <- odds3 %>%
  mutate(logodds = log(odds)))
```

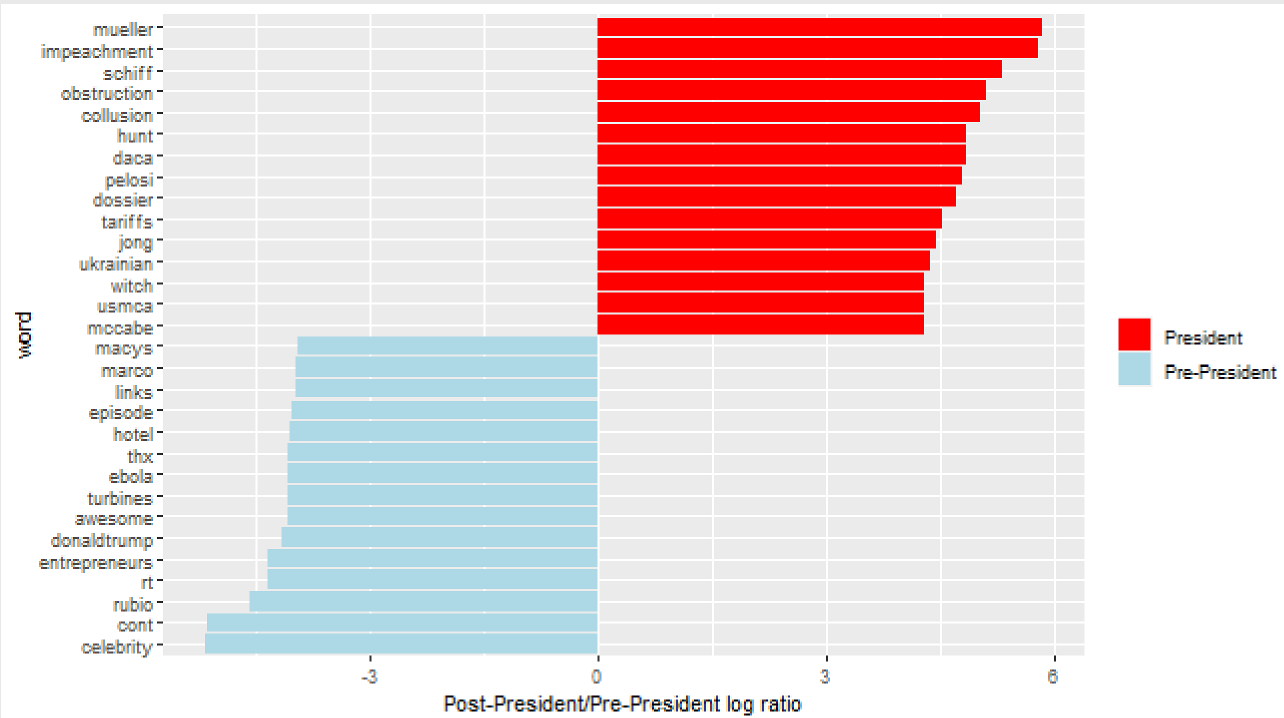
```
## # A tibble: 23,453 × 9
##    word      `FALSE` `TRUE` totFALSE totTRUE propF…¹ propT…²
##    <chr>       <dbl>  <dbl>    <dbl>   <dbl>   <dbl>   <dbl>
##  1 aa              1      0   183927  114138 1.09e-5 8.76e-6
##  2 aaa            11      1   183927  114138 6.52e-5 1.75e-5
##  3 aand            0      1   183927  114138 5.44e-6 1.75e-5
##  4 aaron           2      0   183927  114138 1.63e-5 8.76e-6
##  5 aarons          1      0   183927  114138 1.09e-5 8.76e-6
##  6 ab              1      0   183927  114138 1.09e-5 8.76e-6
##  7 abandon         6      4   183927  114138 3.81e-5 4.38e-5
##  8 abandoned      15      8   183927  114138 8.70e-5 7.89e-5
##  9 abbas           0      2   183927  114138 5.44e-6 2.63e-5
## 10 abbott          1      1   183927  114138 1.09e-5 1.75e-5
## # … with 23,443 more rows, 2 more variables: odds <dbl>,
## #   logodds <dbl>, and abbreviated variable names
## #   ¹propFALSE, ²propTRUE
```

Effect of becoming president

```
p <- prepost_logodds %>%
  group_by(logodds > 0) %>%
  top_n(15, abs(logodds)) %>%
  ungroup() %>%
  mutate(word = reorder(word, logodds)) %>%
  ggplot(aes(word, logodds, fill = logodds < 0)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  ylab("Post-President/Pre-President log ratio") +
  scale_fill_manual(name = "", labels = c("President", "Pre-
President"),
                    values = c("red", "lightblue"))
```

Effect of becoming president

p



Meaning

- Thus far, everything is **topic**-related
  - How often he talks about things
- But what does he **mean** when he talks about Mueller?
  - We can probably guess
- But we want a more systematic method
  - **Sentiment**: the *feeling* behind words

Meaning

- **Sentiment** analysis is based on **dictionaries**
  - Just like **stop words** from last week!
  - Prepared lists of words, but tagged according to **emotion**
- Good dictionary included in `tidytext` package

```
require(tidytext)
```

```
## Loading required package: tidytext
```

```
nrc <- get_sentiments("nrc")
# If this doesn't work on your computer, just load it with
read_rds()
nrc <-
read_rds('https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectur
raw=true')
```

Meaning

```
nrc
```

```
## # A tibble: 13,901 × 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
## # … with 13,891 more rows
```

Sentiment by Pre/Post Presidency

- Measure sentiment by proportion of words
- Divide by pre/post presidency

```
word_freq <- tweet_words %>%
  group_by(PostPresident) %>%
  count(word) %>%
  filter(sum(n) >= 5) %>%
   mutate(prop = prop.table(n)) # Faster way of calculating
proportions!
```

Sentiment by Pre/Post Presidency

- Attaching sentiment from `nrc`
    - `inner_join()`: only keeps words that appear in `nrc`

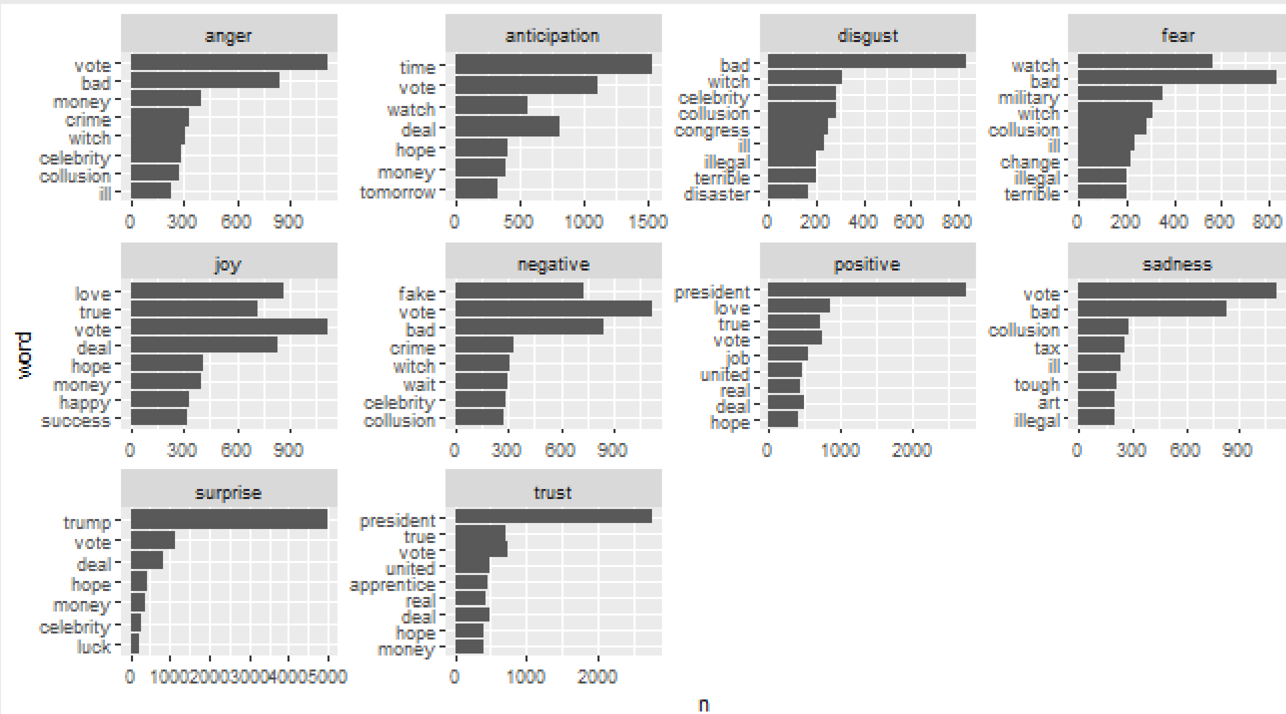
```
word_freq_sentiment <- word_freq %>%
    inner_join(nrc, by = "word")
```

Sentiment overall

```
p <- word_freq_sentiment %>%
  group_by(sentiment) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(y = word, x = n)) +
  facet_wrap(~ sentiment, scales = "free", nrow = 3) +
  geom_bar(stat = "identity")
```

Sentiment Overall

```
p
```



Sentiment overall

- Could also just calculate positive sentiments - negative sentiments
  - Want to do this at the tweet level

```
tweet_sentiment <- tweet_words %>%
    inner_join(nrc, by = "word")

tweet_sentiment_summary <- tweet_sentiment %>%
  group_by(PostPresident, sentiment) %>%
  count(document,sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>% # same as spread()!
  mutate(sentiment = positive - negative)
```

Sentiment overall

```
tweet_sentiment_summary
```

```
## # A tibble: 33,480 × 13
## # Groups:   PostPresident [2]
##    PostP…¹ docum…² anger antic…³ disgust  fear   joy negat…⁴
##    <lgl>     <dbl> <int>   <int>   <int> <int> <int>   <int>
##  1 FALSE    1.70e9     1       3       1     0     2       1
##  2 FALSE    1.74e9     1       1       1     0     1       1
##  3 FALSE    1.92e9     1       1       0     0     1       1
##  4 FALSE    2.05e9     1       0       0     0     0       1
##  5 FALSE    2.32e9     1       0       0     1     0       1
##  6 FALSE    2.35e9     2       1       1     2     1       2
##  7 FALSE    2.40e9     1       2       1     2     1       0
##  8 FALSE    3.69e9     1       0       1     1     0       2
##  9 FALSE    7.68e9     1       1       1     0     2       2
## 10 FALSE    8.08e9     1       1       1     0     1       1
## # … with 33,470 more rows, 5 more variables:
## #   positive <int>, sadness <int>, surprise <int>,
## #   trust <int>, sentiment <int>, and abbreviated variable
## #   names ¹PostPresident, ²document, ³anticipation,
## #   ⁴negative
```

Sentiment by presidency

- Calculate total number of tweets by sentiment

```
tweet_sentiment_summary  %>%
  group_by(PostPresident) %>%
  mutate(ntweet = 1) %>%
  summarize(across(-document, sum))
```

```
## # A tibble: 2 × 13
##   PostPr…¹ anger antic…² disgust  fear   joy negat…³ posit…⁴
##   <lgl>    <int>   <int>   <int> <int> <int>   <int>   <int>
## 1 FALSE     8326   13803    5527  8213 12800   15319   28141
## 2 TRUE      7108    6826    4894  6827  5554   12667   14959
## # … with 5 more variables: sadness <int>, surprise <int>,
## #   trust <int>, sentiment <int>, ntweet <dbl>, and
## #   abbreviated variable names ¹PostPresident,
## #   ²anticipation, ³negative, ⁴positive
```

Sentiment by presidency

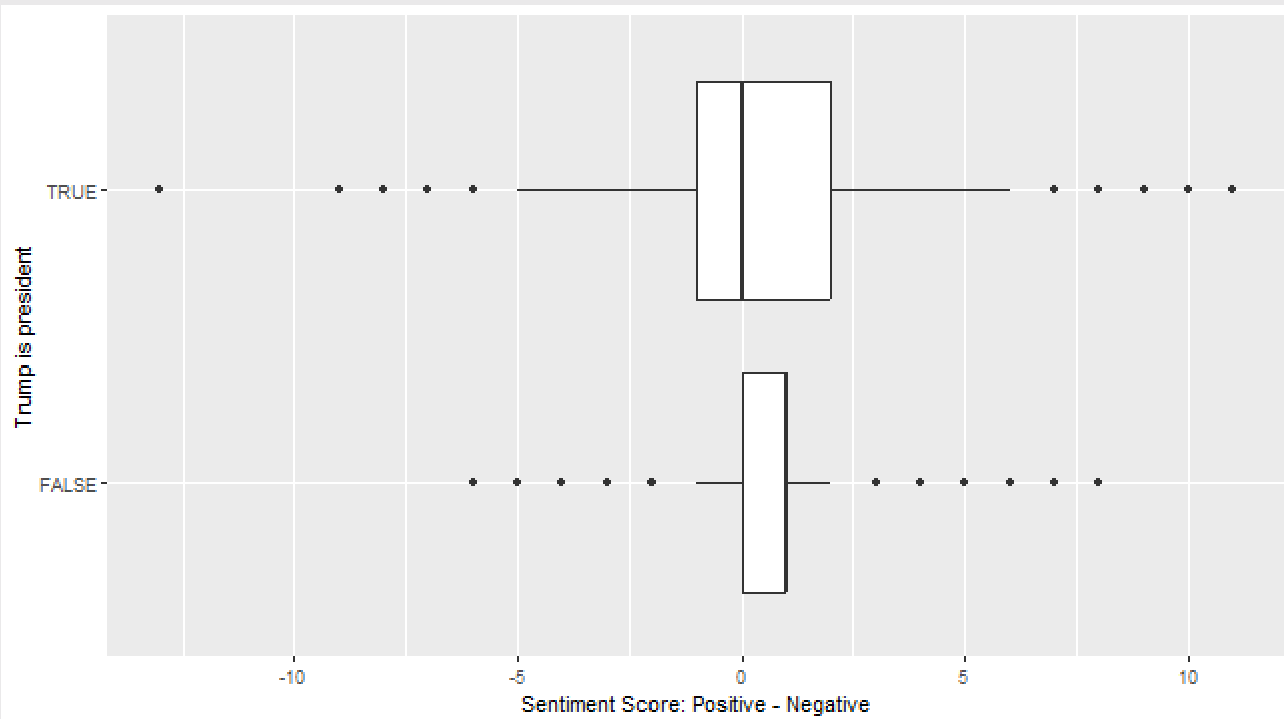
- Univariate distributions!

```
p <- tweet_sentiment_summary %>%
  ggplot(aes(x = sentiment, y = PostPresident)) +
  geom_boxplot() +
  labs(y= "Trump is president", x = "Sentiment Score: Positive -
Negative")
```

Sentiment by presidency

- Univariate distributions!

```
p
```



Sentiment by hour

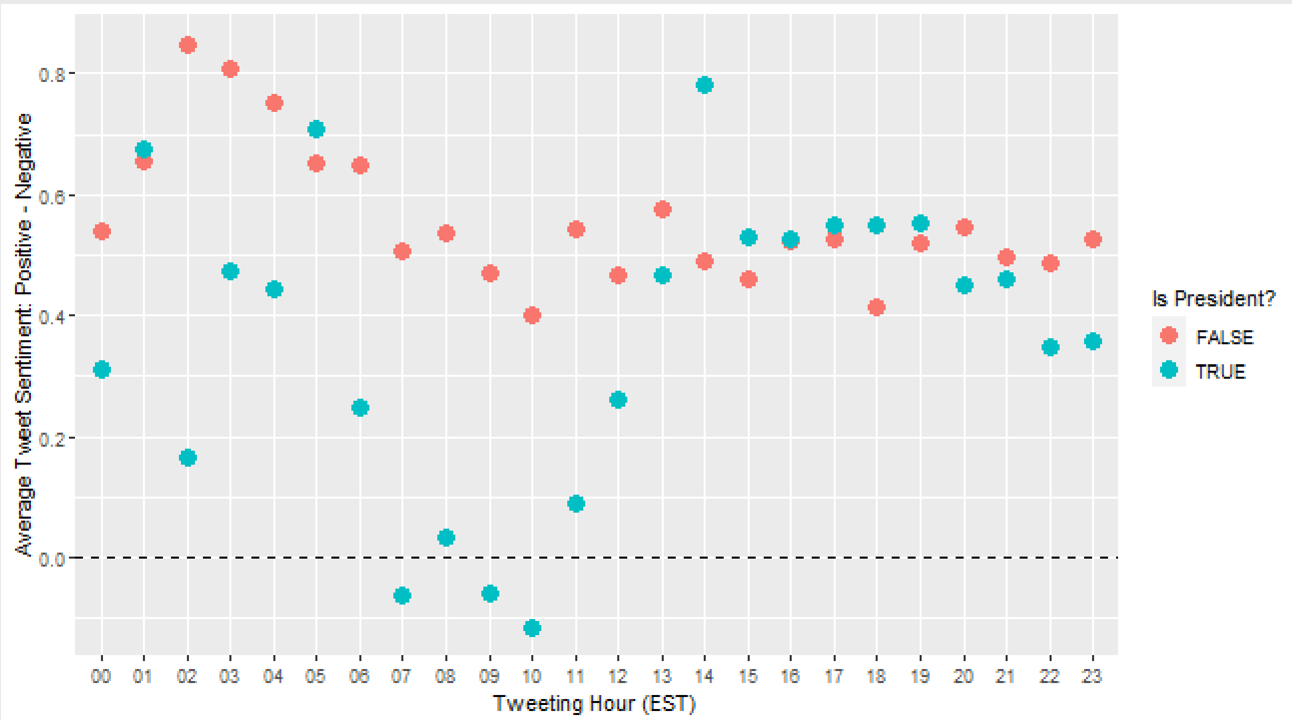
- Univariate distributions
    - Comparing sentiment by hour

```
p <- tweet_sentiment %>%
  group_by(PostPresident,Tweeting.hour,sentiment) %>%
  count(document,sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill
= 0) %>%
  mutate(sentiment = positive - negative) %>%
  summarize(AvgSentiment = mean(sentiment)) %>%
  ggplot(aes(y = AvgSentiment, x= Tweeting.hour,
color=PostPresident)) +
  geom_point(size = 4) +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  labs(x = "Tweeting Hour (EST)", y = "Average Tweet Sentiment:
Positive - Negative", color = "Is President?")
```

Sentiment by hour

- Comparing sentiment by hour

```
p
```



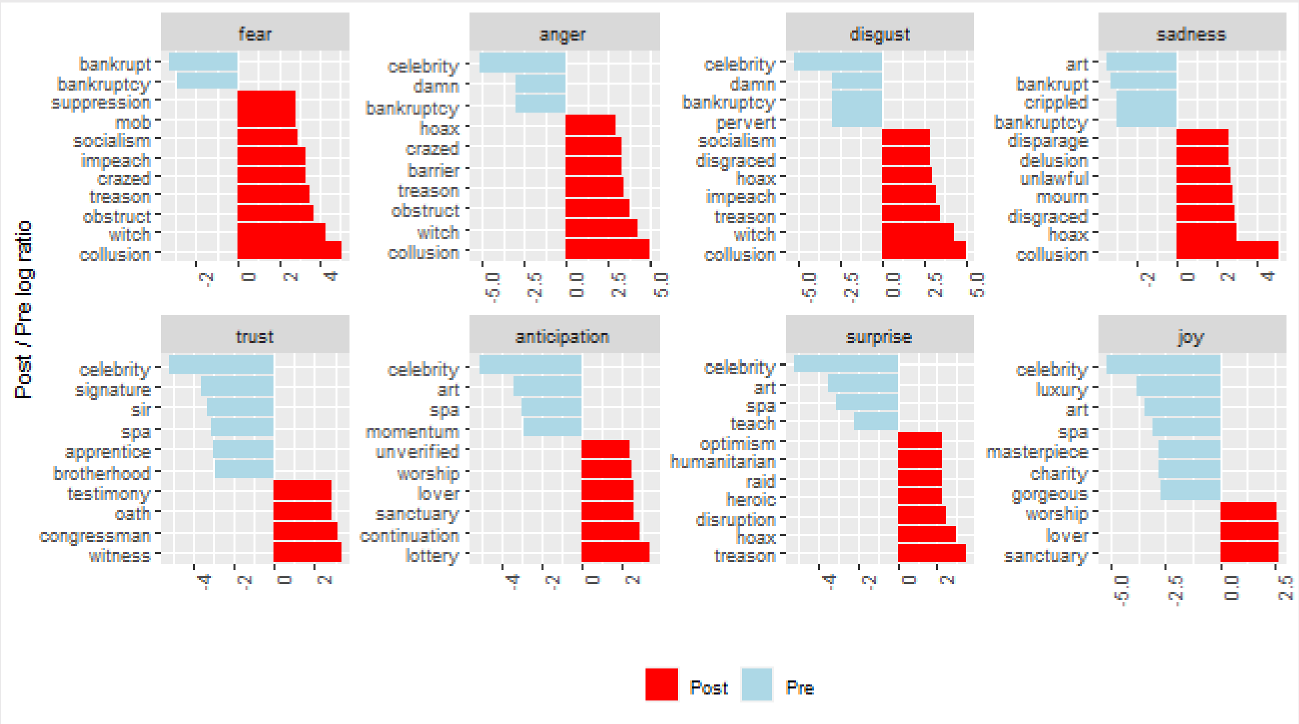
Understanding Trump

- When Trump is coded as "positive" or "negative", what is he saying?
- Look at log-odds ratio words, matched to sentiment!

```
p <- prepost_logodds %>%
  inner_join(nrc, by = "word") %>%
  filter(!sentiment %in% c("positive", "negative")) %>%
  mutate(sentiment = reorder(sentiment, -logodds),
         word = reorder(word, -logodds)) %>%
  group_by(sentiment) %>%
  top_n(10, abs(logodds)) %>%
  ungroup() %>%
  ggplot(aes(y = word, x = logodds, fill = logodds < 0)) +
  facet_wrap(~ sentiment, scales = "free", nrow = 2) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "", y = "Post / Pre log ratio") +
  scale_fill_manual(name = "", labels = c("Post", "Pre"),
                    values = c("red", "lightblue")) +
  theme(legend.position = 'bottom')
```

Understanding Trump

p



Text as predictors

- Let's say we didn't know when each tweet was written
- Could we predict whether it was written during his presidency or not?
    - Logit model using **text** as predictors

Text as Data

- Predict tweets by average of words' log-odds!

```
toanal <- tweet_words %>%
  select(document,word,PostPresident) %>%
  left_join(prepost_logodds %>% select(word,logodds)) %>% # Link
data with log-odds
  group_by(document,PostPresident) %>%
  summarise(logodds = mean(logodds)) %>% # Calculate average log-
odds by document
  ungroup()

m <- glm(PostPresident ~ logodds,toanal,family = binomial) # Logit
regression
```

Text as Data

- Evaluate the performance

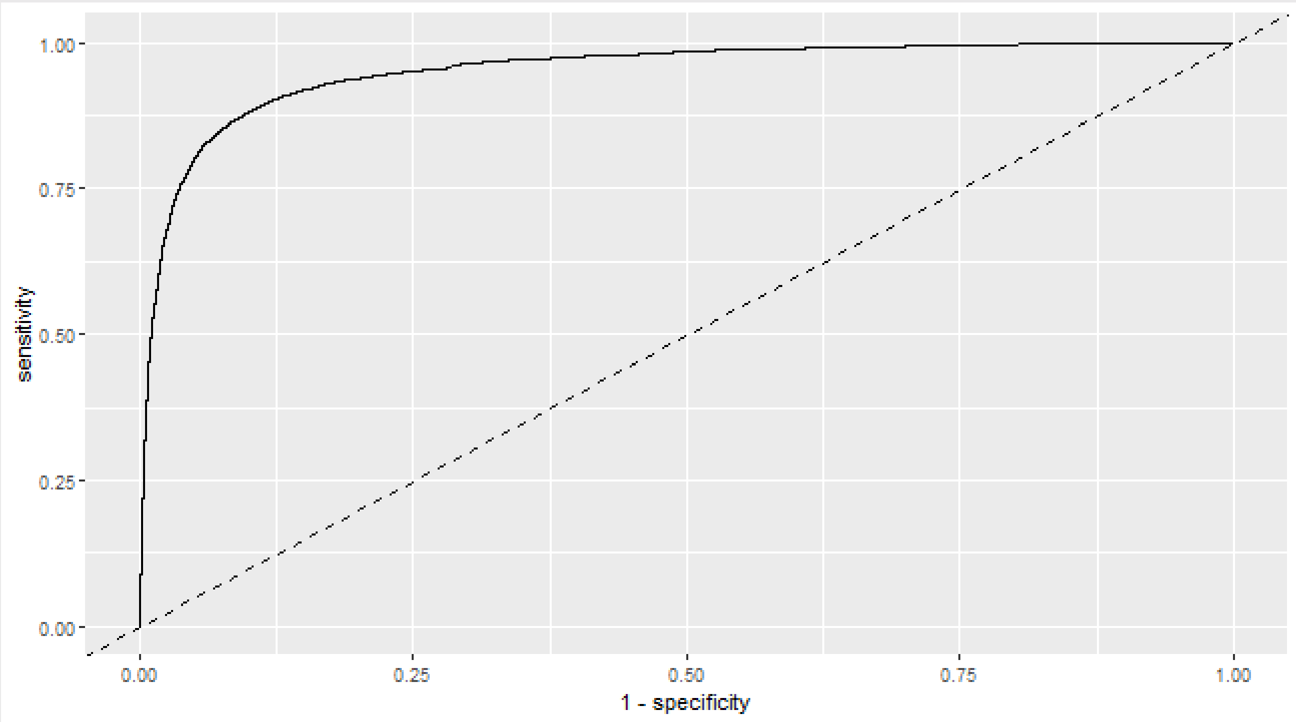
```
require(tidymodels)
forAUC <- toanal %>% # Evaluate model performance
  mutate(preds = predict(m,type = 'response'),
         truth = factor(PostPresident,levels = c('TRUE','FALSE')))

roc_auc(forAUC,'truth','preds')
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.952
```

```
p <- roc_curve(forAUC,'truth','preds') %>%
  ggplot(aes(x = 1-specificity,y = sensitivity)) +
  geom_line() +
  geom_abline(intercept = 0,slope = 1,linetype = 'dashed')
```

Evaluate performance

```
p
```



Evaluate on some sample tweets

```
raw_tweets <- read_rds('../data/Trumptweets.Rds')
set.seed(20)
toCheck <- raw_tweets %>% slice(sample(1:nrow(.),size = 10))

toCheck %>%
  select(content)
```

```
## # A tibble: 10 × 1
##    content
##    <chr>
##  1 "Getting ready to leave for Europe. First meeting - NATO…
##  2 "We should remember that during this entire Petraeus epi…
##  3 "\" @ ZacharySmitty: @ realDonaldTrump I just listened t…
##  4 "Arena was packed, totally electric!"
##  5 "Without momentum there’s a lack of energy that can lead…
##  6 "# CelebrityApprentice Listening to the advice from @ jo…
##  7 "@ Abspara @ pennjillette @ CelebApprentice March 3rd. T…
##  8 "@ sassybrowning Thanks Sassy"
##  9 "How can Hillary run the economy when she can't even sen…
## 10 "\" @ hasantaleb: If @ realDonaldTrump was The President…
```

Evaluate on some sample tweets

```
toTest <- toCheck %>% left_join(toanal,by = c('id' = 'document'))
# Merge the raw text with the log-odds

toTest %>%
  mutate(preds = predict(m,newdata = toTest,type = 'response'))
%>%
  select(content,PostPresident,preds) %>%
  mutate(pred_binary = preds > .5) %>%
  filter(PostPresident != pred_binary)
```

```
## # A tibble: 1 × 4
##   content                            PostPr…¹ preds pred_…²
##   <chr>                              <lgl>    <dbl> <lgl>
## 1 Arena was packed, totally electric! FALSE    0.533 TRUE
## # … with abbreviated variable names ¹PostPresident,
## #   ²pred_binary
```

```
# We only make 1 mistake! And it is on a tough tweet
```

Can we do better if we add sentiment?

```
toanal <- toanal %>%
  left_join(tweet_sentiment_summary) %>%
  drop_na()
```

```
## Joining, by = c("document", "PostPresident")
```

```
m1 <- glm(PostPresident ~ logodds,toanal,family = binomial)
m2 <- glm(PostPresident ~ logodds + sentiment,toanal,family =
binomial)
m3 <- glm(PostPresident ~ logodds + anger + anticipation + disgust
+ fear + joy + sadness + surprise + trust,toanal,family =
binomial)

forAUC <- toanal %>%
  mutate(preds1 = predict(m1,type = 'response'),
         preds2 = predict(m2,type = 'response'),
         preds3 = predict(m3,type = 'response'),
         truth = factor(PostPresident,levels = c('TRUE','FALSE')))
```

Can we do better if we add sentiment?

```
roc_auc(forAUC,'truth','preds1') %>% mutate(model = 'logodds') %>%
  bind_rows(roc_auc(forAUC,'truth','preds2') %>% mutate(model =
'logodds & net sentiment')) %>%
  bind_rows(roc_auc(forAUC,'truth','preds3') %>% mutate(model =
'logodds & detailed sentiment'))
```

```
## # A tibble: 3 × 4
##   .metric .estimator .estimate model
##   <chr>   <chr>          <dbl> <chr>
## 1 roc_auc binary         0.962 logodds
## 2 roc_auc binary         0.963 logodds & net sentiment
## 3 roc_auc binary         0.964 logodds & detailed sentiment
```

- Not really

Conclusion

- Sentiment can...
  - ...help us describe the data (i.e., infer what someone meant)
  - ...help us predict the data (RQ: do positive tweets get more likes?)
- Housekeeping stuff!
  - Pset 8 due Friday, April 14th
  - Pset 9 due Friday, April 21st (EC for using advanced ML!)
  - Final exam due Friday, April 28th