

Text, Tweets, and Sentiment

Part 3

Prof. Bisbee

Vanderbilt University

Slides Updated: 2024-03-31

Returning to Trump

```
require(tidyverse)
tweet_words <-
read_rds(file="https://github.com/jbisbee1/DS1000_S2024/raw/main/data/T
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: 45,221 × 5  
##   word      `FALSE` `TRUE` totFALSE totTRUE  
##   <chr>      <dbl> <dbl>    <dbl>    <dbl>  
## 1 a           6      27   189217   257487  
## 2 aa          1       2   189217   257487  
## 3 aaa        11       1   189217   257487  
## 4 aamp        1       0   189217   257487  
## 5 aand        0       1   189217   257487  
## 6 aaron        2       1   189217   257487  
## 7 ab          1       2   189217   257487  
## 8 abaco        0       1   189217   257487  
## 9 abandon      6       8   189217   257487  
## 10 abandoned  13      11   189217   257487  
## # i 45,211 more rows
```

Odds Step 2

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

```
## # A tibble: 45,221 × 7  
##   word    `FALSE` `TRUE` totFALSE totTRUE propFALSE propTRUE  
##   <chr>    <dbl> <dbl>    <dbl>  <dbl>    <dbl>    <dbl>  
## 1 a         6      27   189217  257487   3.70e-5  1.09e-4  
## 2 aa        1       2   189217  257487   1.06e-5  1.17e-5  
## 3 aaa       11       1   189217  257487   6.34e-5  7.77e-6  
## 4 aamp      1       0   189217  257487   1.06e-5  3.88e-6  
## 5 aand      0       1   189217  257487   5.28e-6  7.77e-6  
## 6 aaron     2       1   189217  257487   1.59e-5  7.77e-6  
## 7 ab        1       2   189217  257487   1.06e-5  1.17e-5  
## 8 abaco     0       1   189217  257487   5.28e-6  7.77e-6  
## 9 aband...  6       8   189217  257487   3.70e-5  3.50e-5  
## 10 aband... 13      11   189217  257487   7.40e-5  4.66e-5  
## # i 45,211 more rows
```

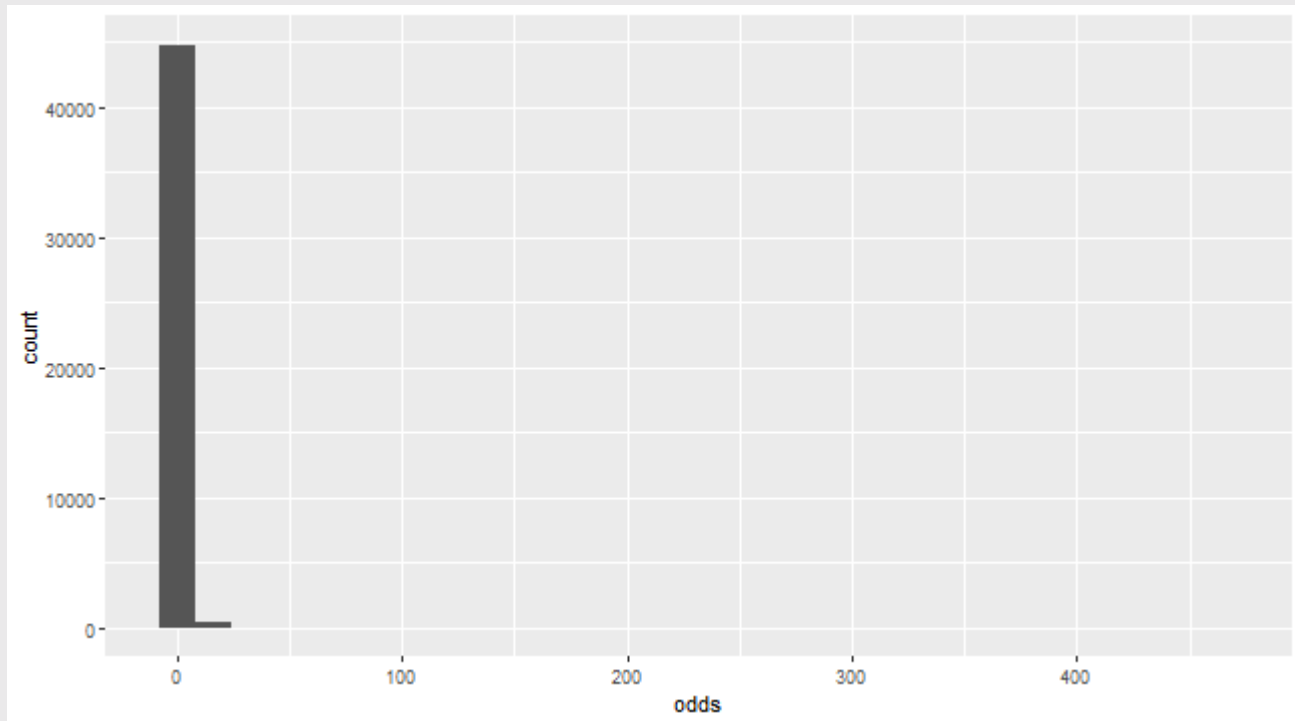
Odds Step 3

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

```
## # A tibble: 45,221 × 8  
##   word      `FALSE` `TRUE` totFALSE totTRUE propFALSE propTRUE  
##   <chr>      <dbl> <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  
## 1 a             6      27   189217   257487    3.70e-5    1.09e-4  
## 2 aa            1       2   189217   257487    1.06e-5    1.17e-5  
## 3 aaa          11       1   189217   257487    6.34e-5    7.77e-6  
## 4 aamp         1       0   189217   257487    1.06e-5    3.88e-6  
## 5 aand         0       1   189217   257487    5.28e-6    7.77e-6  
## 6 aaron        2       1   189217   257487    1.59e-5    7.77e-6  
## 7 ab           1       2   189217   257487    1.06e-5    1.17e-5  
## 8 abaco        0       1   189217   257487    5.28e-6    7.77e-6  
## 9 aband...     6       8   189217   257487    3.70e-5    3.50e-5  
## 10 aband...   13      11   189217   257487    7.40e-5    4.66e-5  
## # i 45,211 more rows  
## # i 1 more variable: odds <dbl>
```

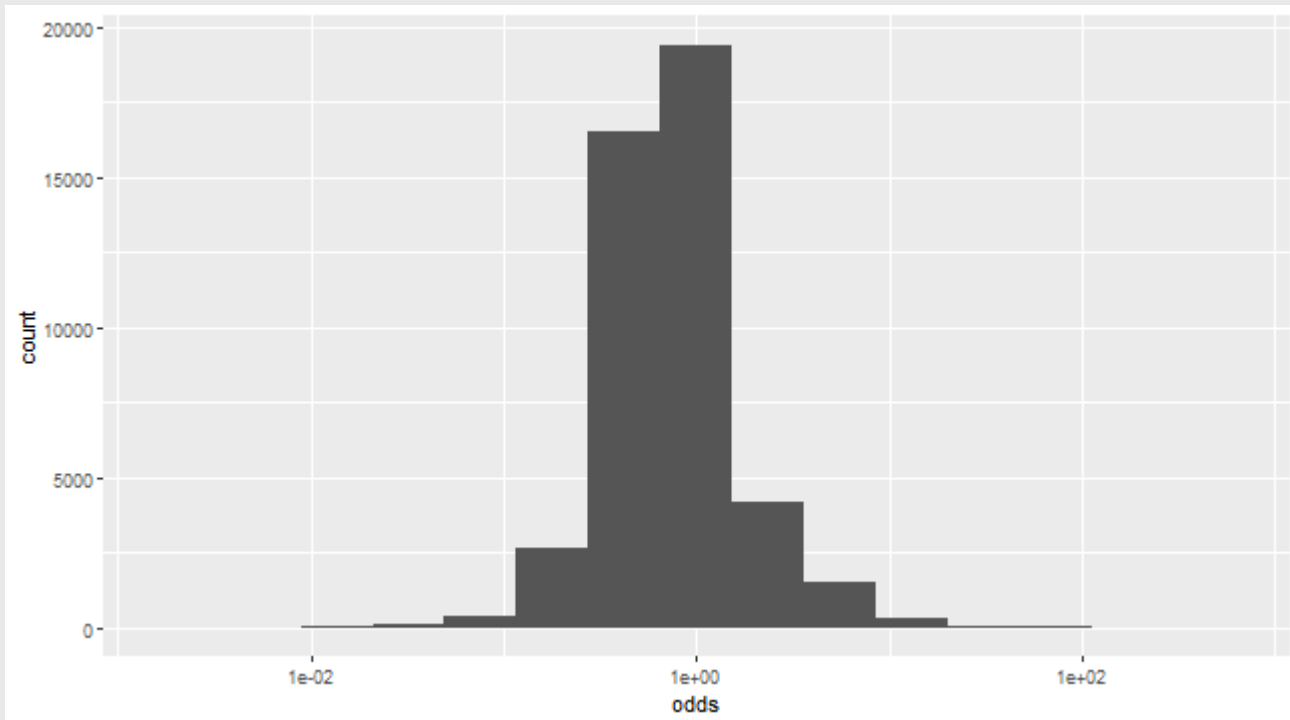
Why log?

```
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)))
```

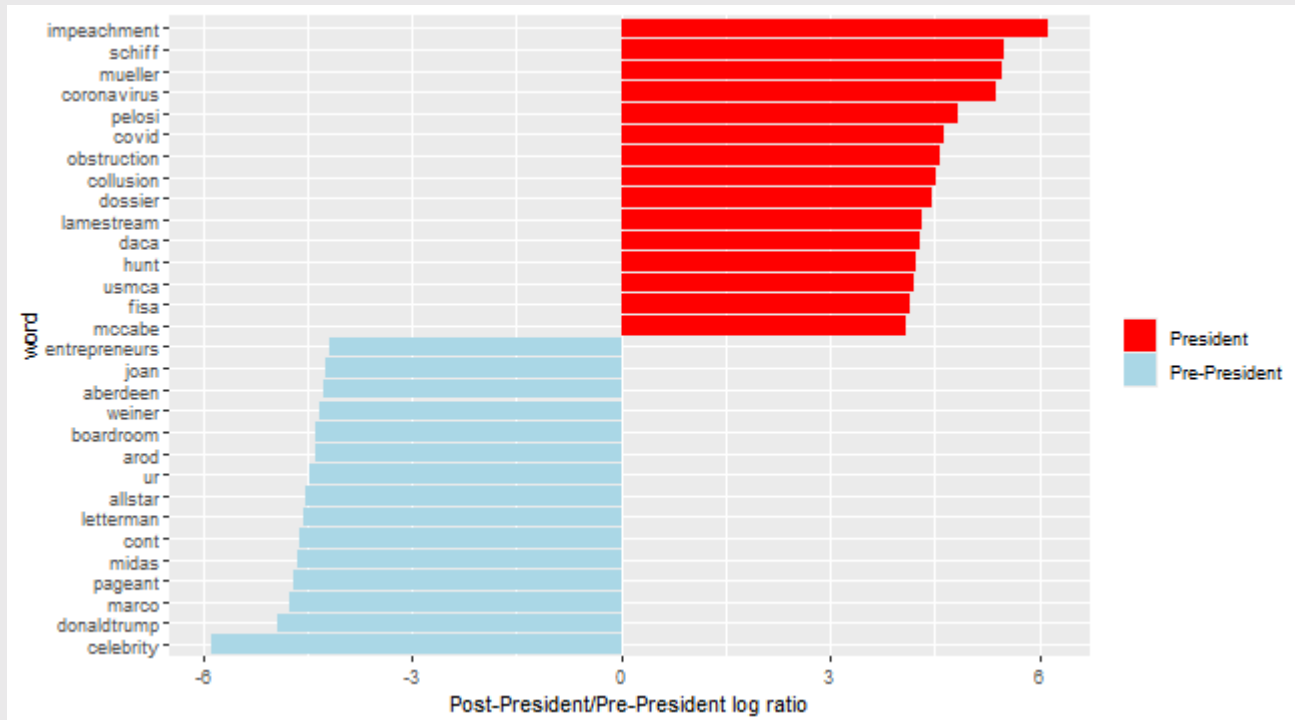
```
## # A tibble: 45,221 × 9  
##   word      `FALSE` `TRUE` totFALSE totTRUE propFALSE propTRUE  
##   <chr>      <dbl> <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  
## 1 a           6      27   189217   257487   3.70e-5   1.09e-4  
## 2 aa          1       2   189217   257487   1.06e-5   1.17e-5  
## 3 aaa         11       1   189217   257487   6.34e-5   7.77e-6  
## 4 aamp         1       0   189217   257487   1.06e-5   3.88e-6  
## 5 aand         0       1   189217   257487   5.28e-6   7.77e-6  
## 6 aaron         2       1   189217   257487   1.59e-5   7.77e-6  
## 7 ab           1       2   189217   257487   1.06e-5   1.17e-5  
## 8 abaco         0       1   189217   257487   5.28e-6   7.77e-6  
## 9 aband...      6       8   189217   257487   3.70e-5   3.50e-5  
## 10 aband...    13      11   189217   257487   7.40e-5   4.66e-5  
## # i 45,211 more rows  
## # i 2 more variables: odds <dbl>, logodds <dbl>
```

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) # Might need to install.packages('textdata')
```

```
## 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_S2024/raw/main/data/nrc.Rd')
```

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
## # i 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")
```

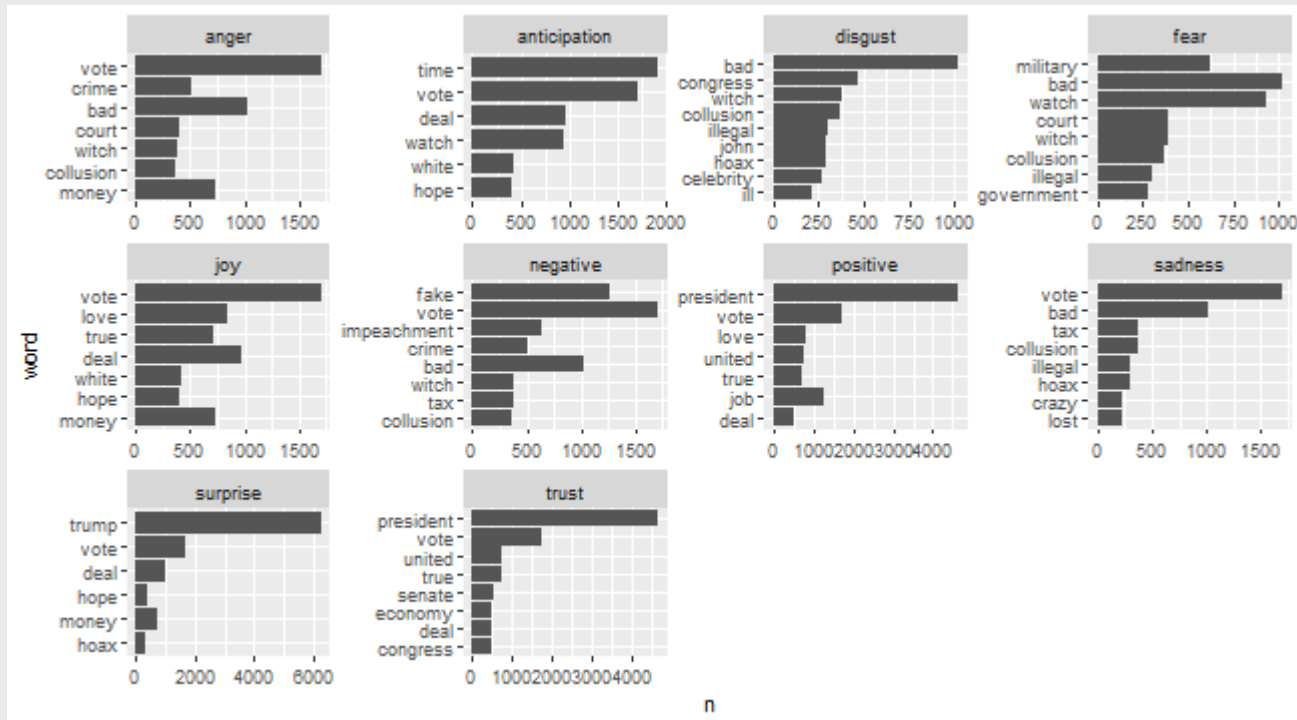
```
## Warning in inner_join(., nrc, by = "word"): Detected an unexpected  
## many-to-many relationship between  
## `x` and `y`.  
## i Row 7 of `x` matches multiple rows in `y`.  
## i Row 2 of `y` matches multiple rows in `x`.  
## i If a many-to-many relationship is expected, set  
## `relationship = "many-to-many"` to silence this warning.
```

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: 45,592 × 13
## # Groups:   PostPresident [2]
##   PostPresident document anger anticipation disgust fear
##   <lg1>          <dbl> <int>      <int>      <int> <int>
## 1 FALSE      1701461182     1         3         1     0
## 2 FALSE      1741160716     1         1         1     0
## 3 FALSE      1924074459     1         1         0     0
## 4 FALSE      2045871770     1         0         0     0
## 5 FALSE      2317112756     1         0         0     1
## 6 FALSE      2346367430     2         1         1     2
## 7 FALSE      2403435685     1         2         1     2
## 8 FALSE      3688564134     1         0         1     1
## 9 FALSE      7677152231     1         1         1     0
## 10 FALSE     8083871612     1         1         1     0
## # i 45,582 more rows
## # i 7 more variables: joy <int>, negative <int>,
## #   positive <int>, sadness <int>, surprise <int>,
## #   trust <int>, sentiment <int>
```

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  
##   PostPresident anger anticipation disgust fear joy  
##   <lgl>          <int>          <int>    <int> <int> <int>  
## 1 FALSE          8138          13333     5356  7999 12440  
## 2 TRUE           13892          14095     8933 14051 10973  
## # i 7 more variables: negative <int>, positive <int>,  
## #   sadness <int>, surprise <int>, trust <int>,  
## #   sentiment <int>, ntweet <dbl>
```

Sentiment by presidency

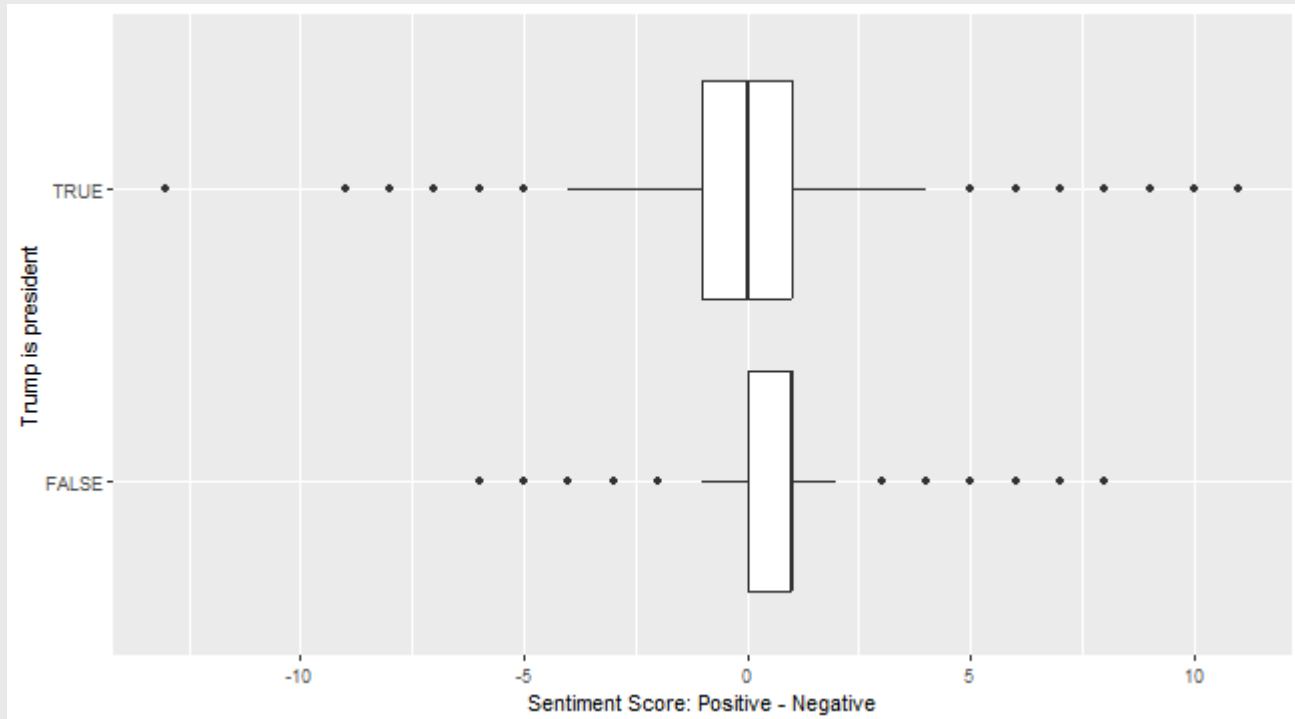
- Univariate distributions!

```
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

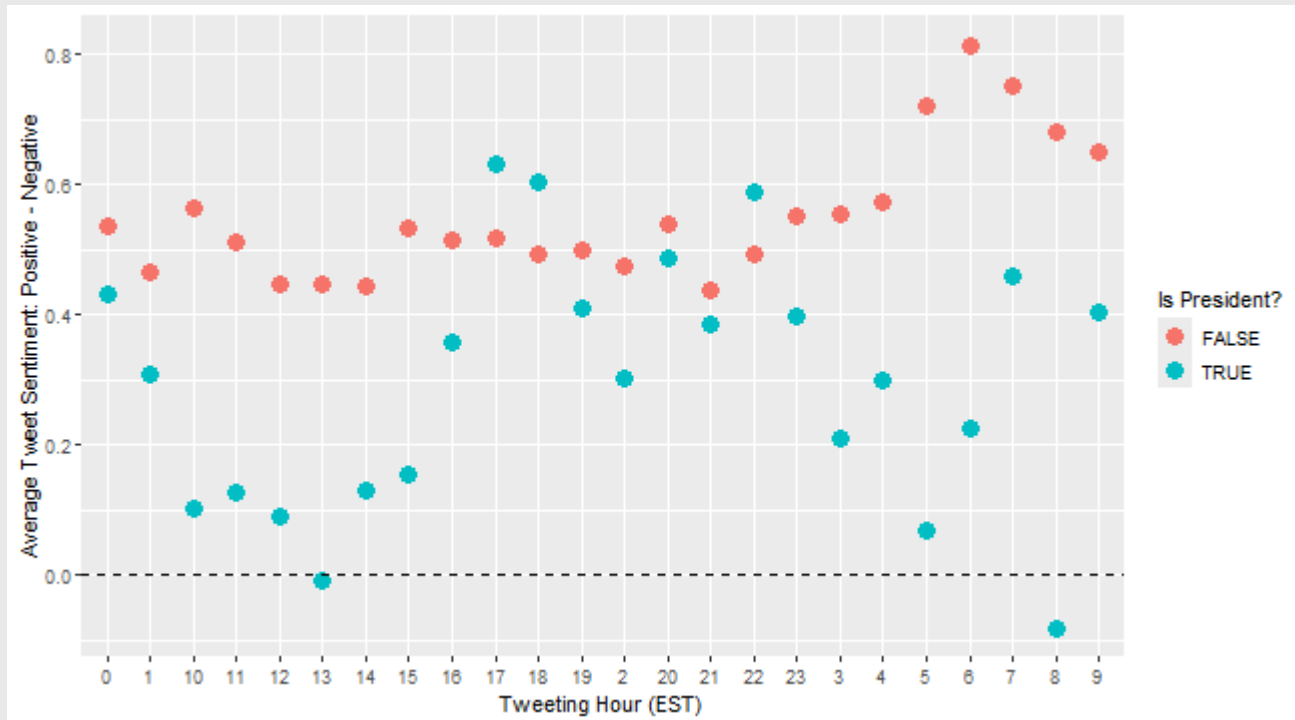
- Univariate distributions
 - Comparing sentiment by hour

```
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



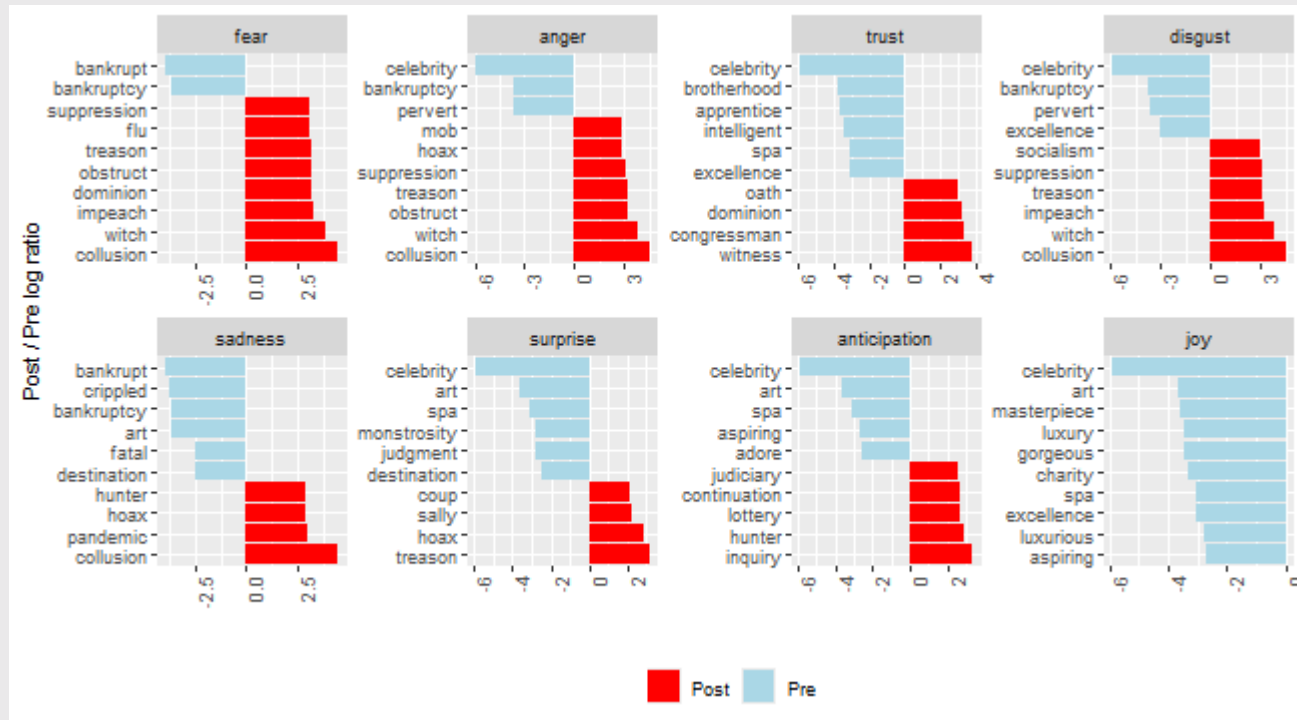
Understanding Trump

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

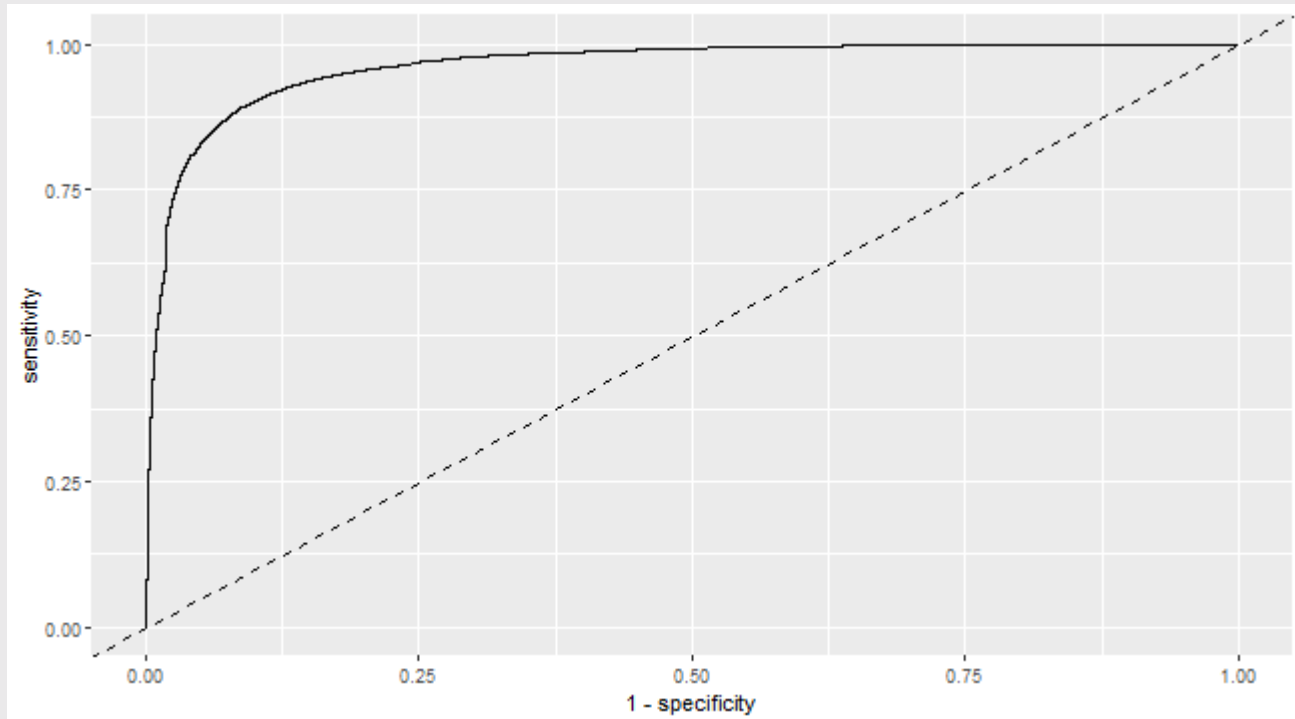
roc_auc(forAUC, 'truth', 'preds')
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>       <dbl>
## 1 roc_auc binary      0.962
```

```
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 "RT @ShannonBream: BREAKING: POTUS commutes Roger Stone..."
## 2 "Congratulations to a future STAR of the Republican Part..."
## 3 "@Seanelmi Thanks Sean."
## 4 "\"\"@0071Lisav: @CNN @realDonaldTrump @CNNPolitics ca..."
## 5 "RT @thejtlewis: @realDonaldTrump https://t.co/W7L9kCZK3..."
## 6 "RT @TrumpWarRoom: WATCH: @KatrinaPierson explains Joe B..."
## 7 "The State Department's 'shadow government' #DrainTheSwa..."
## 8 "Sexual pervert Anthony Weiner has zero business holding..."
## 9 "TO MY FAVORITE PEOPLE IN THE WORLD! https://t.co/38DbQt..."
## 10 "Small businesses will have an ally in the White House w..."
```

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: 3 × 4  
##   content                PostPresident preds pred_binary  
##   <chr>                  <lgl>      <dbl> <lgl>  
## 1 @Seanelmi Thanks Sean. FALSE      0.985 TRUE  
## 2 The State Department's 's... FALSE      0.731 TRUE  
## 3 TO MY FAVORITE PEOPLE IN ... TRUE       0.316 FALSE
```

```
# We only make 3 mistakes!
```

Can we do better if we add sentiment?

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

```
## Joining with `by = join_by(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         0.966 logodds  
## 2 roc_auc binary         0.967 logodds & net sentiment  
## 3 roc_auc binary         0.968 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?)