Topic 3: Modelling Textual Data

Taylor Arnold[1] Lauren Tilton[2]

04 July 2017

[1] Assistant Professor of Statistics
University of Richmond
@statsmaths

[2] Assistant Professor of Digital Humanities
University of Richmond
@nolauren

Setup

Packages

To run the following code, we again make sure that the following packages are loaded (and installed) the following packages.

```
library(cleanNLP)
library(dplyr)
library(readr)
library(stringi)
library(ggplot2)
library(topicmodels)
library(glmnet)
library(ggrepel)
library(viridis)
library(magrittr)
theme_set(theme_minimal())
```

You will also need to download and set-up the tutorial's datasets.

The Data

The President of the United States is constitutionally obligated to provide a report known as the 'State of the Union'. The report summarizes the current challenges facing the country and the president's upcoming legislative agenda.

We have run the spaCy NLP pipeline over this corpus and provide the output data in the GitHub repository.

```
sotu_nlp <- read_csv("data/sotu.csv.gz")
sotu_meta <- read_csv("data/sotu_meta.csv")
```

Exploratory Analysis

Sentence lengths

Just because we are doing text analysis is no excuse for not doing basic exploratory analysis of our data. What, for example, is the distribution of sentence lengths in the corpus?

```
sotu_nlp %>%
  count(id, sid) %$%
  quantile(n, seq(0,1,0.1))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    1   11   16   19   23   27   31   37   44   58  681
```

Common nouns

What are the most common nouns in the corpus?

```
sotu_nlp %>%
  filter(upos == "NOUN") %>%
  count(lemma) %>%
  top_n(n = 40, n) %>%
  use_series(lemma)
```

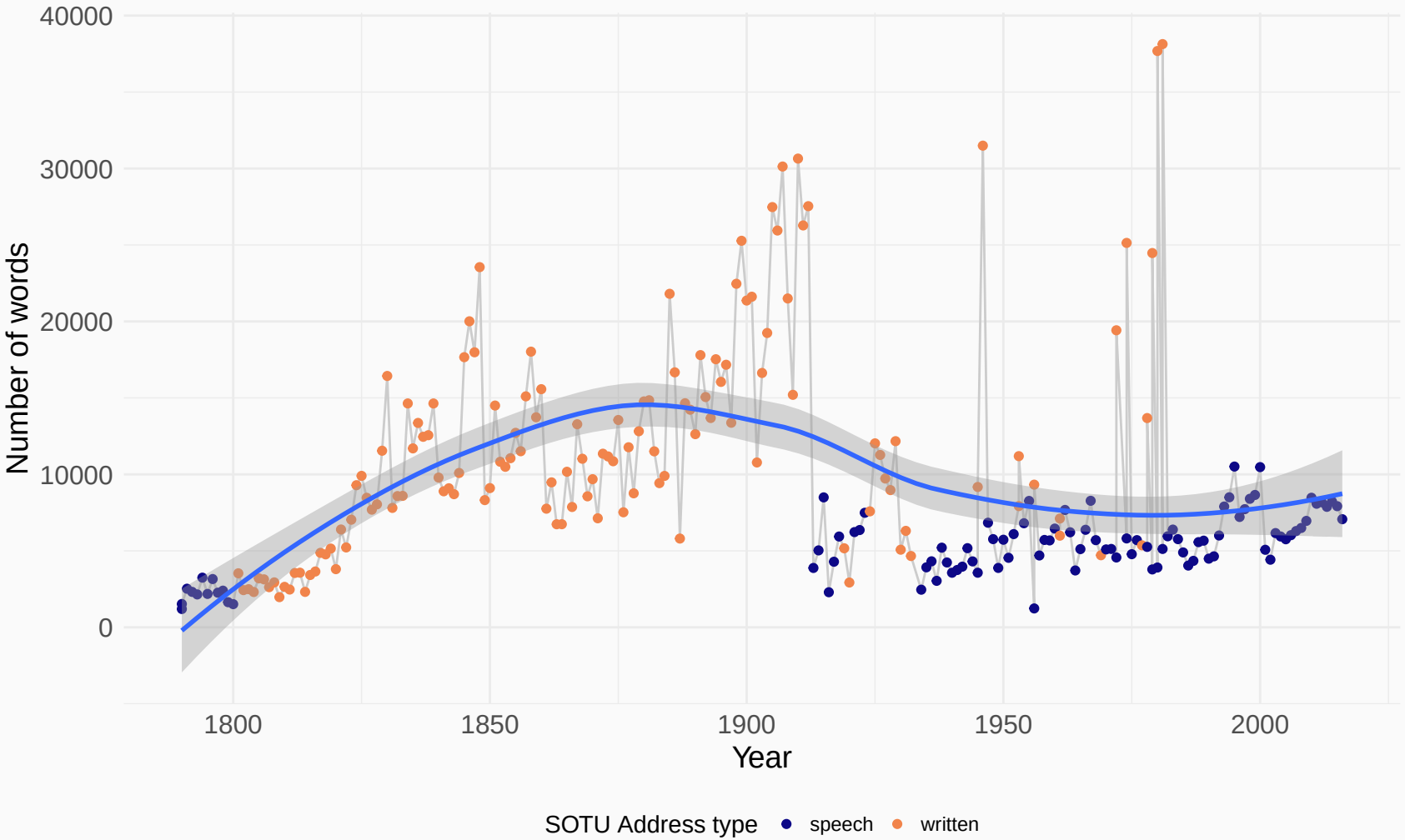
```
##  [1] "act"        "action"     "business"   "citizen"
##  [5] "condition"  "country"    "duty"       "effort"
##  [9] "force"      "government" "interest"   "land"
## [13] "law"        "legislation" "man"       "measure"
## [17] "nation"     "part"       "peace"      "people"
## [21] "policy"     "power"      "program"    "purpose"
## [25] "question"   "right"      "service"    "state"
## [29] "subject"    "system"     "tax"        "time"
## [33] "treaty"     "war"        "way"        "what"
## [37] "who"        "work"       "world"      "year"
```

Length in words

Now, how long is each State of the Union in words? Does this differ based on whether it was given as a speech or a written document?

```
sotu_nlp %>%
  count(id) %>%
  left_join(sotu_meta, by = "id") %>%
  ggplot(aes(year, n)) +
    geom_line(color = grey(0.8)) +
    geom_point(aes(color = sotu_type)) +
    geom_smooth()
```

Length in words



Document Summarisation

Summarising with dependencies

A straightforward way of extracting a high-level summary of the content of a speech is to extract all direct object object dependencies where the target noun is not a very common word.

Here is an example of this using the first address made by George W. Bush in 2001:

```
summary_2001 <- sotu_nlp %>%
  left_join(sotu_meta, by = "id") %>%
  filter(year == 2001, relation == "dobj") %>%
  left_join(word_frequency, by = "word") %>%
  filter(frequency < 0.001) %>%
  select(id, word, word_source) %$%
  sprintf("%s => %s", word_source, word)
```

George W. Bush (2001)

```
summary_2001
```

```
##  [1] "take => oath"                      "increasing => layoffs"
##  [3] "buying => prescriptions"           "protects => trillion"
##  [5] "makes => welcoming"                "accelerating => cleanup"
##  [7] "fight => homelessness"             "allowing => taxpayers"
##  [9] "provide => mentor"                 "fight => illiteracy"
## [11] "promotes => compassion"            "end => profiling"
## [13] "stopping => abuses"                "pay => trillion"
## [15] "throw => darts"                    "restores => fairness"
## [17] "restructure => defenses"     "promoting => internationalism"
## [19] "makes => downpayment"              "deploy => defenses"
## [21] "discard => relics"                 "confronting => shortage"
## [23] "sound => footing"                  "bridge => divides"
## [25] "minding => manners"                "divided => conscience"
## [27] "done => servants"
```

George W. Bush (2002)

```
head(summary_2002, 34)
```

```
##  [1] "faces => dangers"          "urged => followers"
##  [3] "brought => sorrow"         "found => diagrams"
##  [5] "hold => hostages"          "eliminate => parasites"
##  [7] "prevent => regimes"        "flaunt => hostility"
##  [9] "develop => anthrax"        "kicked => inspectors"
## [11] "match => hatred"           "attack => allies"
## [13] "deploy => defenses"        "permit => regimes"
## [15] "increased => vigilance"    "develop => vaccines"
## [17] "fight => anthrax"          "expand => patrols"
## [19] "track => arrivals"         "mean => neighborhoods"
## [21] "thank => attendants"       "defeat => recession"
## [23] "want => paycheck"          "set => posturing"
## [25] "reduce => dependency"      "offer => dignity"
## [27] "enact => safeguards"       "keeping => commitments"
## [29] "saw => selves"             "embracing => ethic"
## [31] "extending => compassion"   "extend => compassion"
## [33] "await => knock"            "owns => aspirations"
```

Woodrow Wilson (1919)

```
head(summary_1919,  34)
```

```
##  [1] "save => inconvenience"        "produce => stagnation"
##  [3] "produce => stagnation"        "made => interruption"
##  [5] "keep => armies"               "-and => necessaries"
##  [7] "arrive => permitting"         "urge => necessity"
##  [9] "produced => bitterness"       "remove => grievances"
## [11] "produces => dissatisfaction" "stir => disturbances"
## [13] "shown => willingness"         "bring => democratization"
## [15] "analyze => particulars"       "bid => pause"
## [17] "saps => vitality"             "treat => manifestations"
## [19] "touch => tissues"             "come => unrest"
## [21] "settle => disputes"           "devise => tribunal"
## [23] "lose => composure"            "realize => fruition"
```

Term Frequency Matrices

NLP and matrices

So far, we have done all of our analysis using a data frame where each token is given its own row. For modelling purposes, we often want to calculate the term frequency matrix. This matrix has one row per document and one column per unique token in the data set (although we can limit which tokens actually have a column).

Conveniently, **cleanNLP** provides the function `get_tfidf` for calculated this matrix.

Document term frequency matrix

	I	and	...	commute	...	lol
Text #0001	20	55	...	0	...	0
Text #0002	34	72	...	5	...	0
Text #0003	6	34	...	0	...	4
⋮	⋮	⋮	...	⋮	⋱	⋮
Text #9500	150	87	...	0	...	30

get_tfidf()

Here we will construct a term frequency matrix from only non-proper nouns:

```
sotu_tfidf <- sotu_nlp %>%
  filter(pos %in% c("NN", "NNS")) %>%
  get_tfidf(min_df = 0.05, max_df = 0.95,
                      type = "tfidf", tf_weight = "dnorm")
```

get_tfidf()

The output is a list with three elements: the term frequency inverse document frequency matrix, the ids of the documents corresponding to row names, and the vocabulary corresponding to the column names.

```
head(sotu_tfidf$vocab, 20)
```

```
##  [1] "world"       "citizen"     "service"     "duty"
##  [5] "system"      "right"       "man"         "program"
##  [9] "policy"      "work"        "act"         "condition"
## [13] "subject"     "legislation" "force"       "effort"
## [17] "treaty"      "purpose"     "land"        "business"
```

```
head(sotu_tfidf$id, 20)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
dim(sotu_tfidf$tfidf)
```

```
## [1]  236 2356
```

PCA

What specifically can we do with this data? As a starting point, we will compute the principal components of the matrix. While base-R has great functions for doing this, we'll make use of the **cleanNLP** function `tidy_pca` which returns a data frame that makes plotting in **ggplot2** easier:

```
sotu_pca <- tidy_pca(sotu_tfidf$tfidf, sotu_meta)
select(sotu_pca, president, party, PC1, PC2)
```

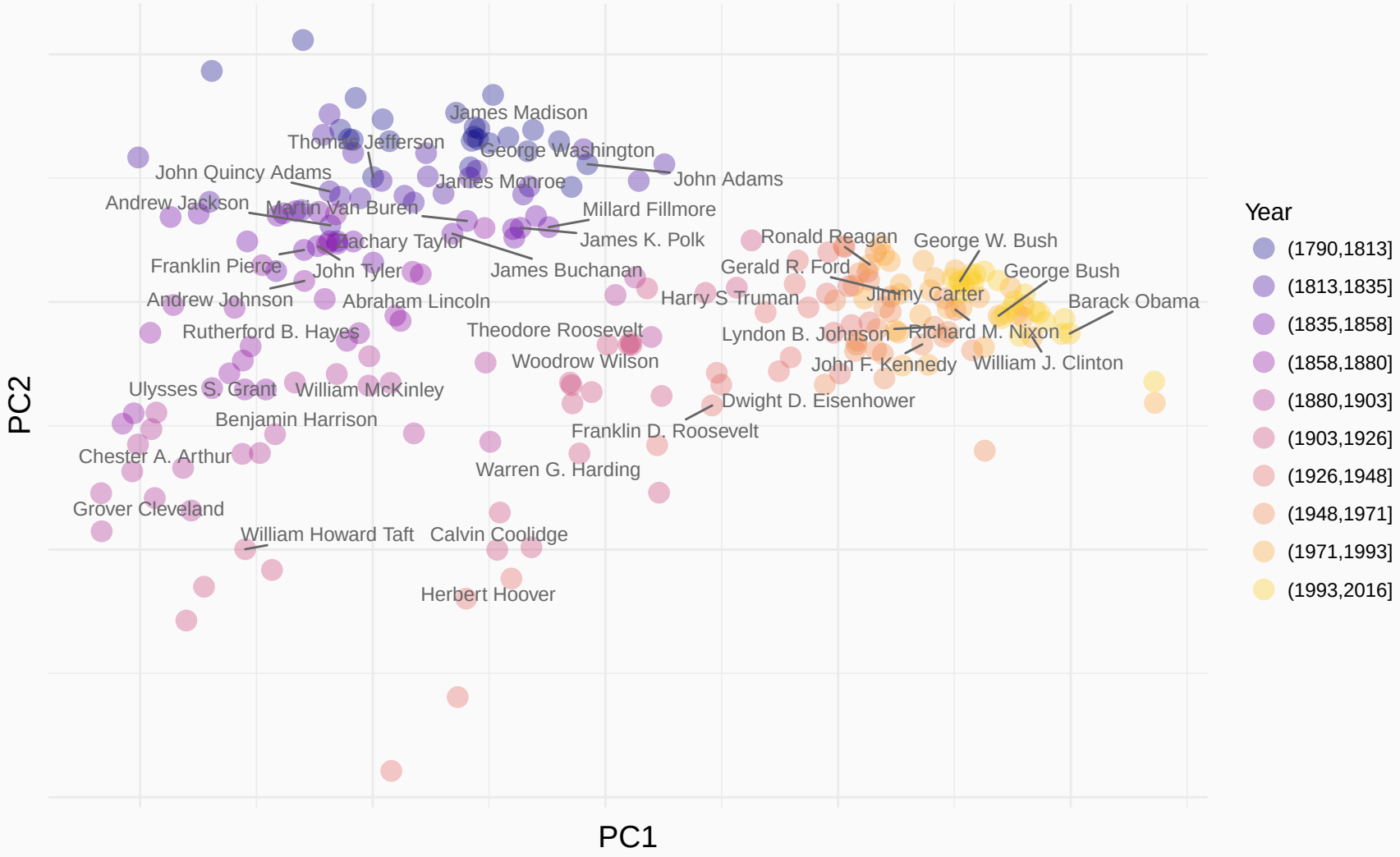
```
## # A tibble: 236 x 4
##            president       party        PC1      PC2
##                <chr>       <chr>      <dbl>    <dbl>
## 1 George Washington Nonpartisan  -1.994792 12.97925
## 2 George Washington Nonpartisan  -4.829730 16.72253
## 3 George Washington Nonpartisan  -5.744745 13.01544
## 4 George Washington Nonpartisan  -3.341501 12.16999
## 5 George Washington Nonpartisan -16.921023 18.65898
## 6 George Washington Nonpartisan  -5.658502 13.34691
## # ... with 230 more rows
```

PCA plot

While a simple scatter plot of this is easy to construct, we can tweak some of the default settings to get a really nice visualization of where each President's speeches cluster:

```
ggplot(sotu_pca, aes(PC1, PC2)) +
  geom_point(aes(color = cut(year, 10, dig.lab = 4))) +
  geom_text(data = filter(sotu_pca, !duplicated(president)))
```

PCA plot, cont.



Topic Models

Topic Models

Topic models are a collection of statistical models for describing abstract themes within a textual corpus. Each theme is characterized by a collection of words that commonly co-occur; for example, the words ‘crop’, ‘dairy’, ‘tractor’, and ‘hectare’, might define a *farming* theme.

One of the most popular topic models is latent Dirichlet allocation (LDA), a Bayesian model where each topic is described by a probability distribution over a vocabulary of words. Each document is then characterized by a probability distribution over the available topics.

LDA

To fit LDA on a corpus of text parsed by the **cleanNLP** package, the output of `get_tfidf` can be piped directly to the `LDA` function in the package **topicmodels**. The topic model function requires raw counts, so the type variable in `get_tfidf` is set to "tf".

```
sotu_tf <- sotu_nlp %>%
  filter(pos %in% c("NN", "NNS")) %>%
  get_tfidf(min_df = 0.05, max_df = 0.95,
                       type = "tf", tf_weight = "raw")
tm <- LDA(sotu_tf$tf, k = 16, control = list(verbose = 1))
```

Describing topics

We can describe each topic by giving the five most important words in each topic:

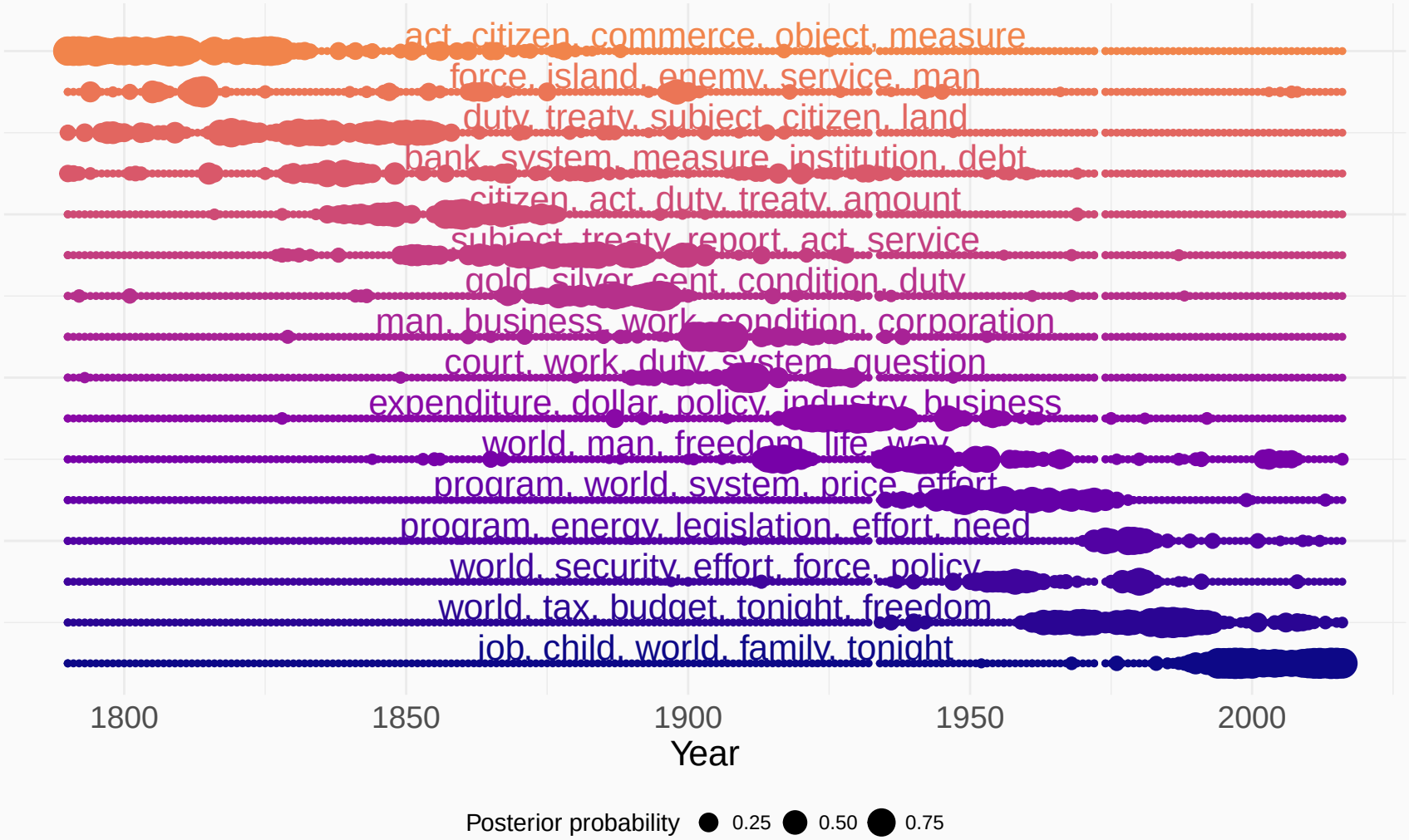
```
terms <- posterior(tm)$terms
topics <- posterior(tm)$topics
topic_df <- data_frame(topic = as.integer(col(topics)),
                       id = sotu_meta$id[as.integer(row(topics))],
                       val = as.numeric(topics)) %>%
              left_join(sotu_meta, by = "id")
top_terms <- apply(terms, 1,
                   function(v) paste(sotu_tf$vocab[order(v,
                      decreasing = TRUE)[1:5]], collapse = ", "))
top_terms <- as.character(top_terms)
```

Describing topics

```
top_terms
```

```
##  [1] "act, citizen, commerce, object, measure"
##  [2] "man, business, work, condition, corporation"
##  [3] "duty, treaty, subject, citizen, land"
##  [4] "world, man, freedom, life, way"
##  [5] "bank, system, measure, institution, debt"
##  [6] "program, world, system, price, effort"
##  [7] "world, tax, budget, tonight, freedom"
##  [8] "job, child, world, family, tonight"
##  [9] "world, security, effort, force, policy"
## [10] "program, energy, legislation, effort, need"
## [11] "citizen, act, duty, treaty, amount"
## [12] "expenditure, dollar, policy, industry, business"
## [13] "court, work, duty, system, question"
## [14] "gold, silver, cent, condition, duty"
## [15] "force, island, enemy, service, man"
## [16] "subject, treaty, report, act, service"
```

Topics over time



Authorship Detection

Predictive models

A classifier that distinguishes speeches made by two presidents will be constructed here for the purpose of illustrating the topical and stylistic differences between them and their speech writers.

As a first step, a term-frequency matrix is extracted using the same technique as was used with the topic modeling function. However, here the frequency is computed for each sentence in the corpus rather than the document as a whole.

‘George Bush (2001-2008)’

‘Barack Obama (2009-2016)’

Design matrix

The ability to do this seamlessly with a single additional `mutate` function defining a new id illustrates the flexibility of the `get_tfidf` function.

```
df <- sotu_nlp %>%
  left_join(sotu_meta, by = "id") %>%
  filter(president %in% c("Barack Obama", "George W. Bush")) %>%
  mutate(new_id = paste(id, sid, sep = "-")) %>%
  filter(pos %in% c("NN", "NNS"))
mat <- get_tfidf(df, min_df = 0, max_df = 1, type = "tf",
                 tf_weight = "raw", doc_var = "new_id")
```

Training and testing sets

It will be necessary to define a response variable `y` indicating whether this is a speech made by President Obama as well as a training flag indicating which speeches were made in odd numbered years.

```
m2 <- data_frame(new_id = mat$id) %>%
  left_join(df[!duplicated(df$new_id),]) %>%
  mutate(y = as.numeric(president == "Barack Obama")) %>%
  mutate(train = (year %% 2 == 0))
```

Elastic net

The output may now be used as input to the elastic net function provided by the **glmnet** package. This function fits a model of the form:

$$\beta = \text{argmin}_b \left\{ ||y - Xb||_2 + \lambda \cdot (\alpha)||b||_1 + \lambda \cdot (1-\alpha)||b||_2^2 \right\}$$

Cross-validation is used in order to select the best value of the model's tuning parameter λ.

```
model <- cv.glmnet(mat$tf[m2$train,], m2$y[m2$train],
                   family = "binomial", alpha = 0.9)
```

The response is set to the binomial family given the binary nature of the response and training is trained on only those speeches occurring in odd-numbered years.

Predicted probabilities

We can add the predicted probabilites to the dataset m2 with the following:

```
m2$pred <- predict(model, newx = mat$tf, type = "response",
                   s = model$lambda.1se)
select(m2, new_id, id, sid, president, year, pred)
```

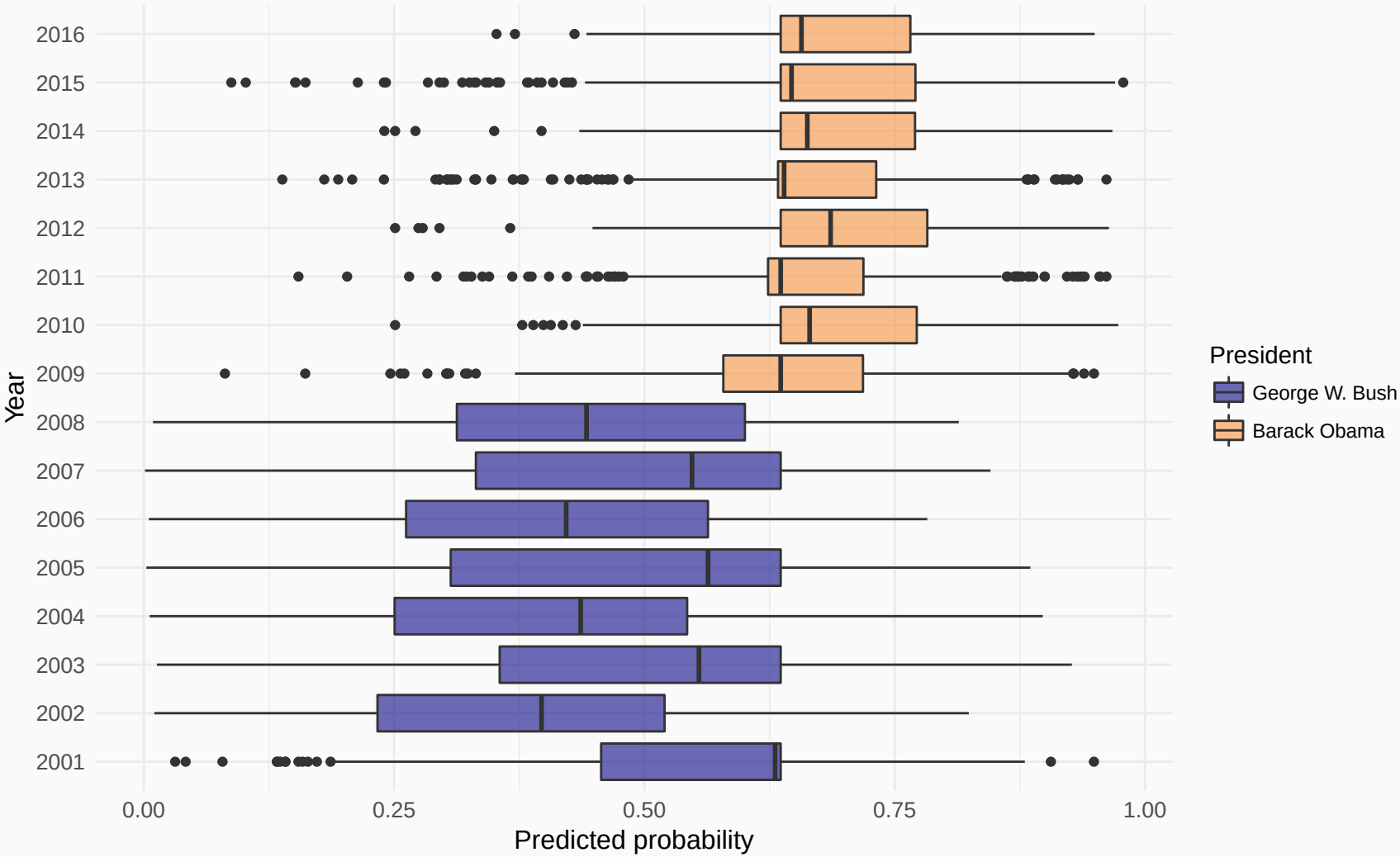
```
## # A tibble: 4,821 x 6
##   new_id    id   sid      president  year      pred
##    <chr> <int> <int>          <chr> <int>     <dbl>
## 1  221-1   221     1 George W. Bush  2001 0.5976356
## 2  221-2   221     2 George W. Bush  2001 0.6361072
## 3  221-3   221     3 George W. Bush  2001 0.6361072
## 4  221-4   221     4 George W. Bush  2001 0.7841981
## 5  221-5   221     5 George W. Bush  2001 0.6361072
## 6  221-6   221     6 George W. Bush  2001 0.7841981
## # ... with 4,815 more rows
```

Predicted probabilities

A boxplot of the predicted classes for each sentence within a speach is a good way of evaluating the model:

```
ggplot(m2, aes(factor(year), pred)) +
  geom_boxplot(aes(fill = president))
```

Predicted probabilities



Model coefficients

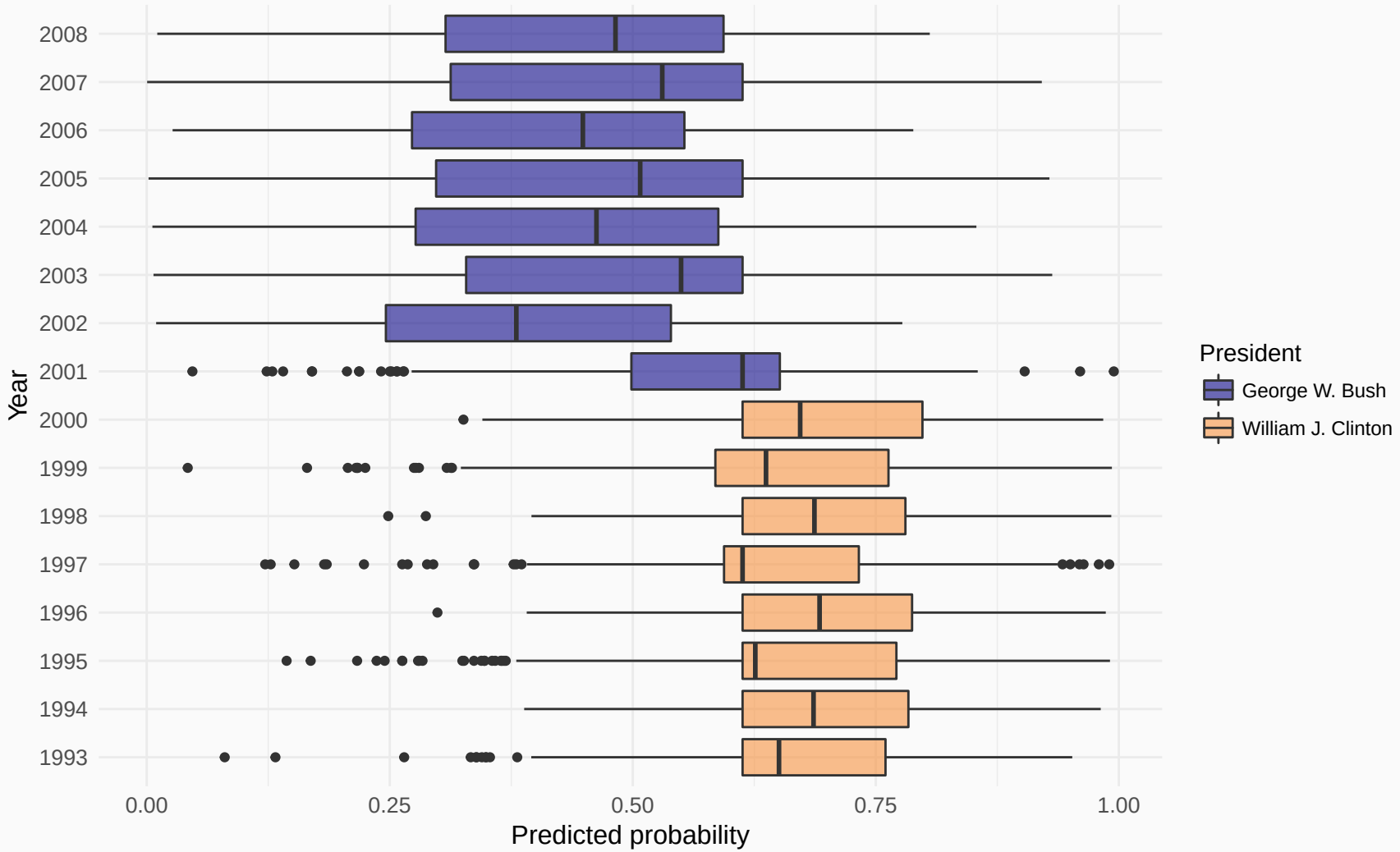
One benefit of the penalized linear regression model is that it is possible to interpret the coefficients in a meaningful way. Here are the non-zero elements of the regression vector, coded as whether the have a positive (more Obama) or negative (more Bush) sign:

```
beta <- coef(model, s = model[["lambda"]][10])[-1]
sprintf("%s (%d)", mat$vocab, sign(beta))[beta != 0]
```

```
##  [1] "job (1)"          "nation (-1)"      "business (1)"
##  [4] "child (-1)"       "terrorist (-1)"   "freedom (-1)"
##  [7] "college (1)"      "company (1)"      "thing (1)"
## [10] "peace (-1)"       "change (1)"       "enemy (-1)"
## [13] "terror (-1)"      "hope (-1)"        "drug (-1)"
## [16] "kid (1)"          "regime (-1)"      "class (1)"
## [19] "industry (1)"     "member (-1)"      "relief (-1)"
## [22] "liberty (-1)"     "compassion (-1)"  "enforcement (-1)"
## [25] "medicine (-1)"    "death (-1)"       "11th (-1)"
## [28] "homeland (-1)"    "will (-1)"        "character (-1)"
## [31] "culture (-1)"
```

‘Bill Clinton (1993-2000)’

Predicted probabilities (Bush vs. Clinton)

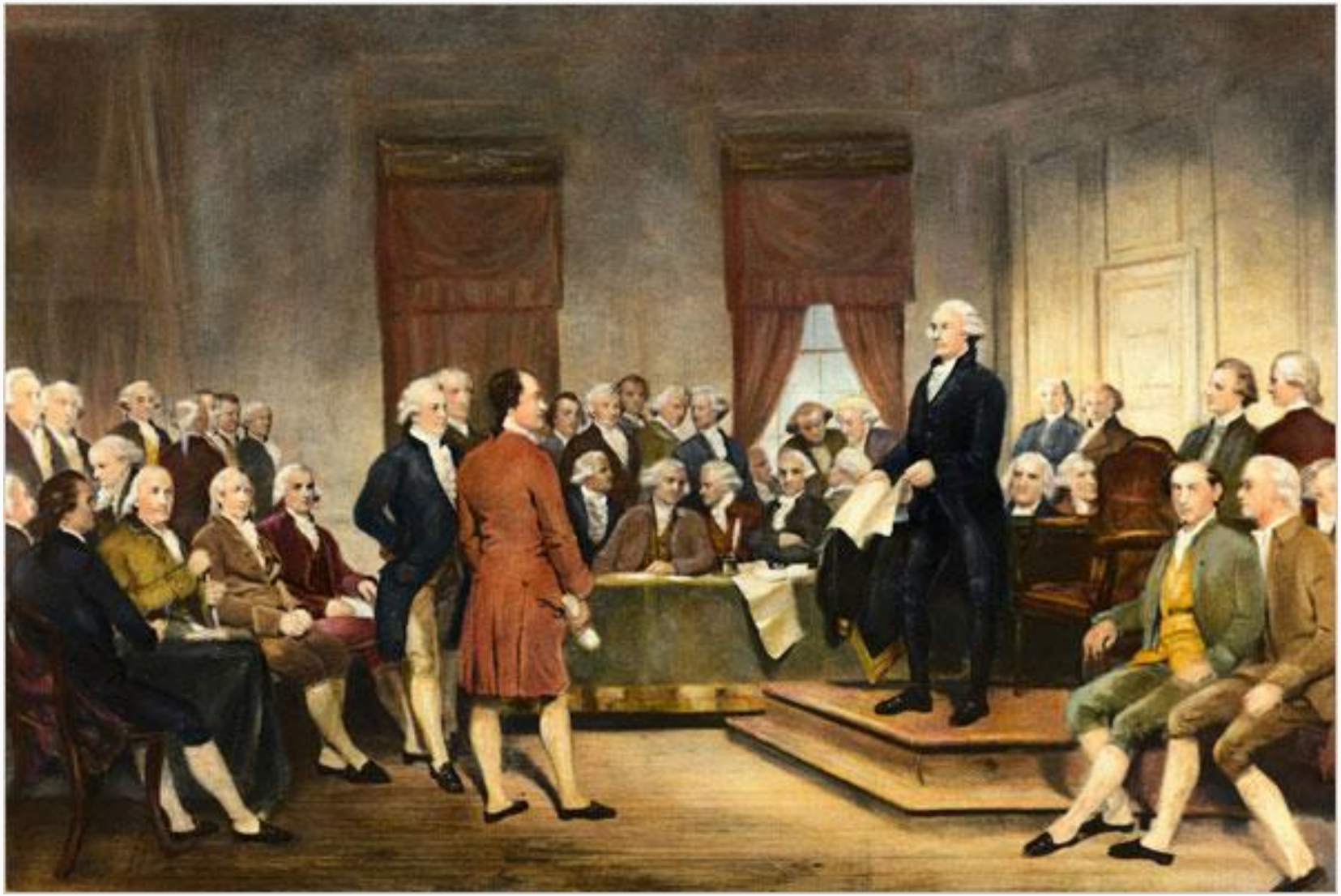


Model coefficents (Bush vs. Clinton)

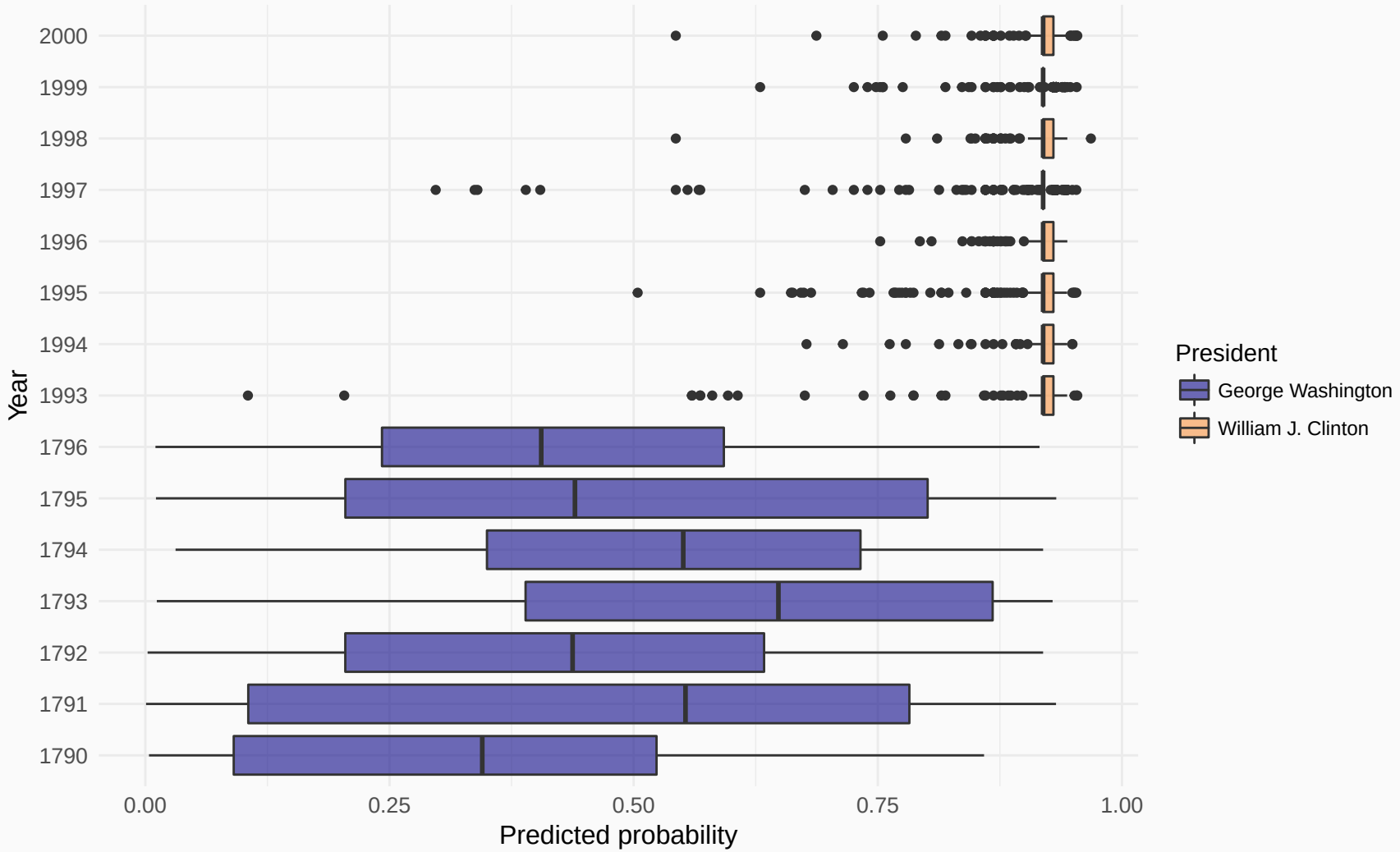
```
beta <- coef(model, s = model[["lambda"]][9])[-1]
sprintf("%s (%d)", mat$vocab, sign(beta))[beta != 0]
```

```
##  [1] "year (1)"          "child (1)"         "family (1)"
##  [4] "care (1)"          "community (1)"     "freedom (-1)"
##  [7] "century (1)"       "parent (1)"        "thing (1)"
## [10] "welfare (1)"       "terrorist (-1)"    "challenge (1)"
## [13] "woman (-1)"        "crime (1)"         "terror (-1)"
## [16] "enemy (-1)"        "hope (-1)"         "something (1)"
## [19] "idea (1)"          "troop (-1)"        "gun (1)"
## [22] "regime (-1)"       "relief (-1)"       "liberty (-1)"
## [25] "danger (-1)"       "attack (-1)"       "institution (-1)"
## [28] "compassion (-1)"   "coalition (-1)"    "dignity (-1)"
## [31] "11th (-1)"         "oil (-1)"          "homeland (-1)"
```

‘George Washington (1789-1797)’



Predicted probabilities (Washington vs. Clinton)



Model coefficients (Washington vs. Clinton)

```
##  [1] "provision (-1)"     "measure (-1)"       "object (-1)"
##  [4] "attention (-1)"     "mean (-1)"          "session (-1)"
##  [7] "consideration (-1)" "militia (-1)"     "establishment (-1)"
## [10] "circumstance (-1)"  "commerce (-1)"     "satisfaction (-1)"
## [13] "tribe (-1)"         "commissioner (-1)"  "execution (-1)"
## [16] "case (-1)"
```