Lab 3.

IQMR 2016

US Senate Speeches

Let's take a look at a US Senate debate on partial birth abortion. As ever, we'll load the texts, make a corpus, then a document term matrix to start off

Then we make a document feature matrix to fit a model to

```
corp <- corpus(textfile("data/abortion-debate-us-senate/*"))
docnames(corp) <- dir("data/abortion-debate-us-senate") ## if you need docnames
docvars(corp, "party") <- c(rep("D", 5), rep("R",7))
summary(corp)


Corpus consisting of 12 documents.

              Text Types Tokens Sentences party
      DEMboxer.txt  2112  17094       798     D
     DEMdurbin.txt   522   1896        89     D
  DEMfeinstein.txt  1379   6439       290     D
     DEMharkin.txt   650   2612       143     D
 DEMlautenberg.txt   715   2300       125     D
     REPallard.txt   400   1206        53     R
  REPbrownback.txt   641   2891       162     R
     REPdewine.txt   460   1473        71     R
     REPensign.txt   413   1256        66     R
      REPhatch.txt   455   1201        59     R
   REPsantorum.txt  1502   9805       422     R
   REPsessions.txt   643   2177       106     R

Source:  /Users/will/wip/iqmr/session3/lab3/* on x86_64 by will
Created: Wed Jun 22 11:28:42 2016
Notes:


corpdfm <- dfm(corp)


Creating a dfm from a corpus ...
   ... lowercasing
   ... tokenizing
   ... indexing documents: 12 documents
   ... indexing features: 3,783 feature types
   ... created a 12 x 3784 sparse dfm
   ... complete.
Elapsed time: 0.101 seconds.
```

The quanteda package has a variety of scaling models, but for ease of examination we'll use the austin package instead. First we'll trim the

```
library(austin)
```

```
Attaching package: 'austin'

The following objects are masked from 'package:quanteda':

    as.wfm, trim
```

```
senatewfm <- wfm(corpdfm, word.margin=2) ## austin wants a wfm object
senatewfm <- trim(senatewfm)
```

```
Words appearing less than 5 times: 2764
Words appearing in fewer than 5 documents: 3286
```

and fit the model on a trimmed version

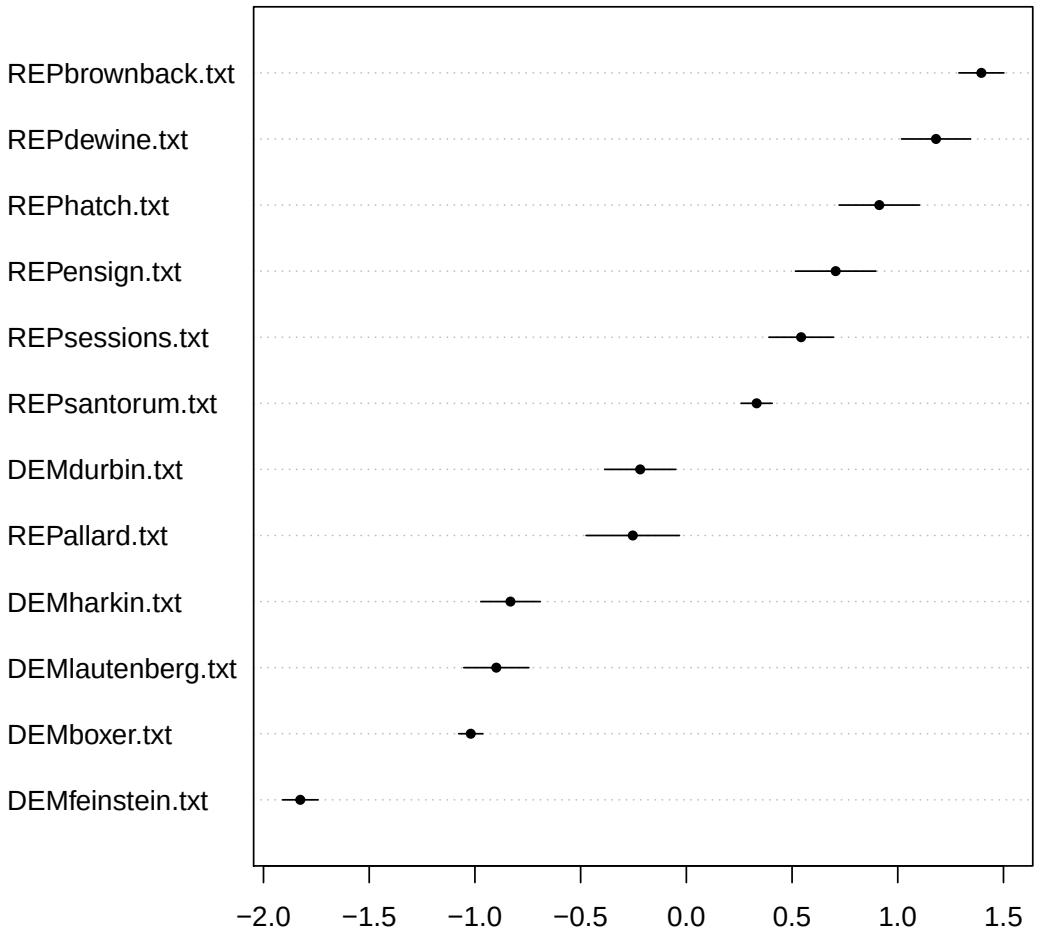
```
mod <- wordfish(senatewfm)
summary(mod)
```

```
Call:
wordfish(wfm = senatewfm)

Document Positions:
                 Estimate Std. Error   Lower    Upper
DEMboxer.txt      -1.0197    0.02922 -1.0770 -0.96243
DEMdurbin.txt     -0.2183    0.08574 -0.3864 -0.05029
DEMfeinstein.txt  -1.8261    0.04294 -1.9102 -1.74190
DEMharkin.txt     -0.8318    0.07174 -0.9724 -0.69122
DEMlautenberg.txt -0.8989    0.07854 -1.0528 -0.74497
REPallard.txt     -0.2534    0.11274 -0.4744 -0.03245
REPbrownback.txt   1.3955    0.05435  1.2889  1.50200
REPdewine.txt      1.1809    0.08333  1.0176  1.34423
REPensign.txt      0.7063    0.09714  0.5159  0.89668
REPhatch.txt       0.9125    0.09703  0.7224  1.10271
REPsantorum.txt    0.3324    0.03759  0.2587  0.40607
REPsessions.txt    0.5430    0.07781  0.3904  0.69547
```

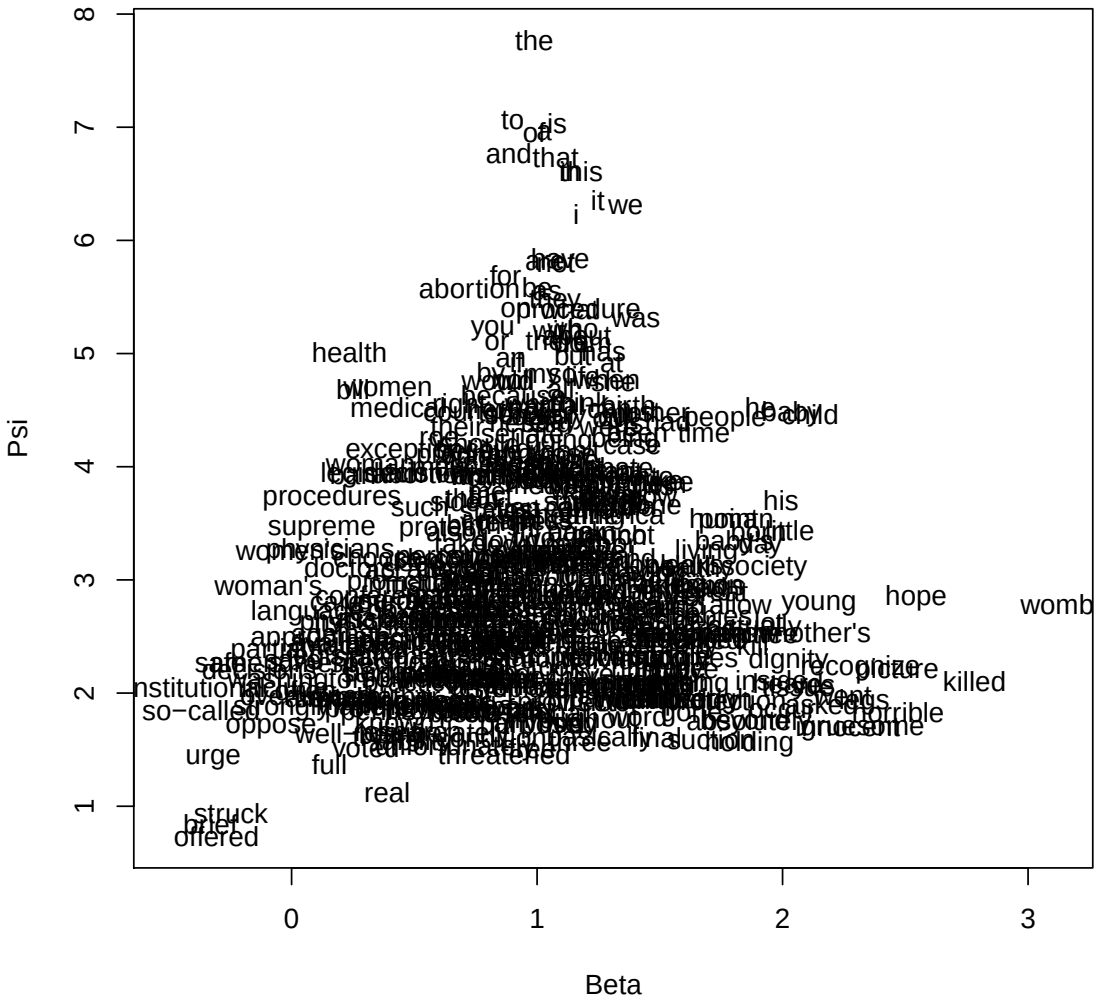
Table summaries are nice but plots are better

```
plot(mod)
```



We can also get one of those nice 'Eiffel Tower' plots that Proksch and Slapin use

```
plot(coef(mod, form='poisson'))
```



A little alpha transparency would proabbly help this (or a really big screen). In any case the word positions are available as β and document positions as θ.

Let's take a look those the word positions and see how they line up on the dimension. Let's plot the slope estimates for some likely looking word stems. But first we have to extract them from all the other parameters.

```
wds <- mod$words
betas <- mod$beta
wparams <- data.frame(word=wds, beta=betas)
wparams <- wparams[order(wparams$beta), ] ## sort by beta
nrow(wparams)
```

```
[1] 498
```

Let's take a quick look at the extremes

```
head(wparams)
```

```
               word       beta
118 unconstitutional -0.4981021
347        so-called -0.3680537
439            brief -0.3302910
345             urge -0.3196375
294          offered -0.3058749
262             safe -0.2896441
```

```
tail(wparams)
```

```
        word     beta
497     legs 2.332637
465  picture 2.466731
273 horrible 2.470805
295     hope 2.543928
496   killed 2.782065
472     womb 3.118871
```

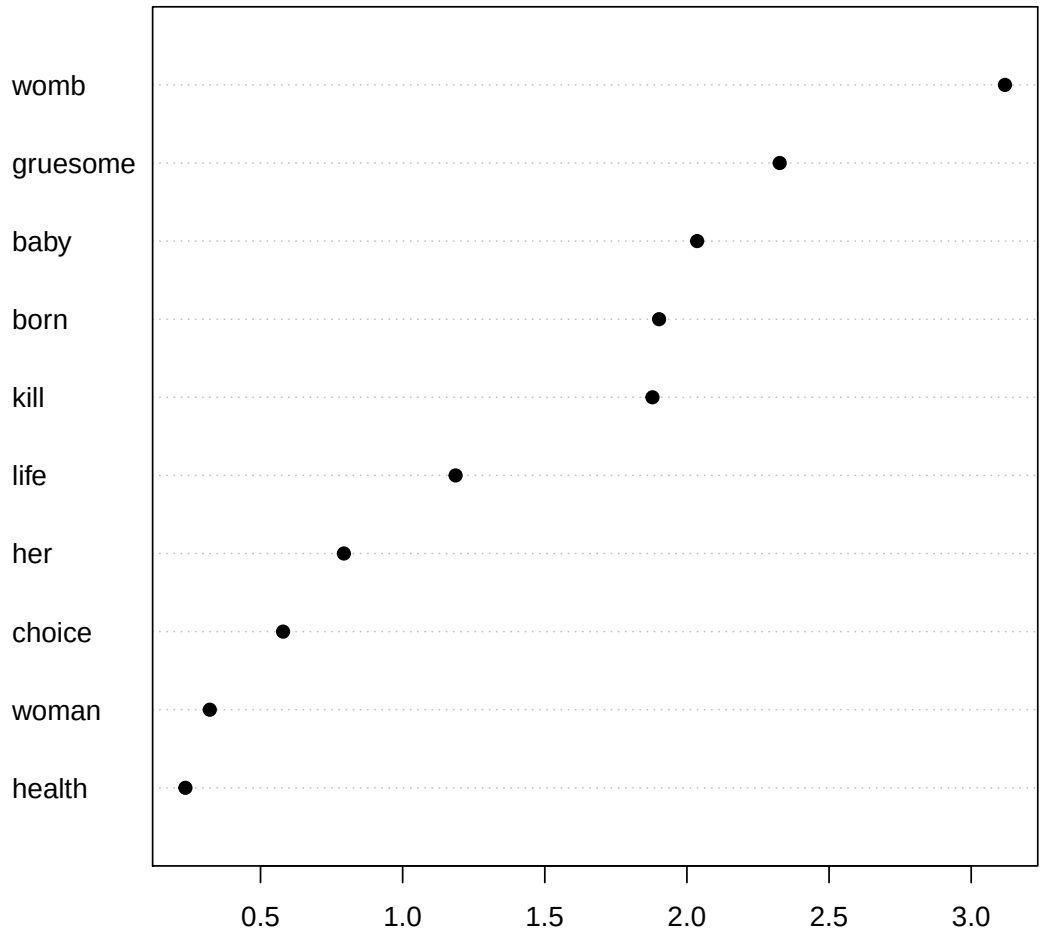
Or we could choose some likely candidates

```
testwords <- c("life", "choice", "womb", "her", "woman", "health",
               "born", "baby", "gruesome", "kill")
samp <- subset(wparams, word %in% testwords)
samp
```

```
        word      beta
50    health 0.2357655
51     woman 0.3215173
301   choice 0.5792851
339      her 0.7932154
93      life 1.1863159
140     kill 1.8788762
434     born 1.9020905
303     baby 2.0359070
484 gruesome 2.3263387
472     womb 3.1188710
```

or plot just these

```
dotchart(samp$beta, samp$word, pch=19)
```



If we were being thorough about these words we'd check they do what we think they do by looking by looking at them in all their contexts, as we did in lab 1.

We can also look at more than one dimension in this data. For this we'll use the `ca` package. You may need use to `install.package` this first.

```
library(ca)
dim(senatewfm) ## we need to flip this around for ca
```

```
[1] 498  12
```

```
mod2 <- ca(t(senatewfm), nf=2) ## note transpose t
```

The `ca` package calls its θs `rowcoord` and β `colcoord`.

Although this is a least squares approximation to the wordfish model, the approximation is pretty good. Let's compare the first dimension with wordfish's document positions. We'll correlate because the (arbitrary) scaling is different between models

```
catheta <- mod2$rowcoord[,1]
cor(catheta, mod$theta)


[1] 0.9942665
```

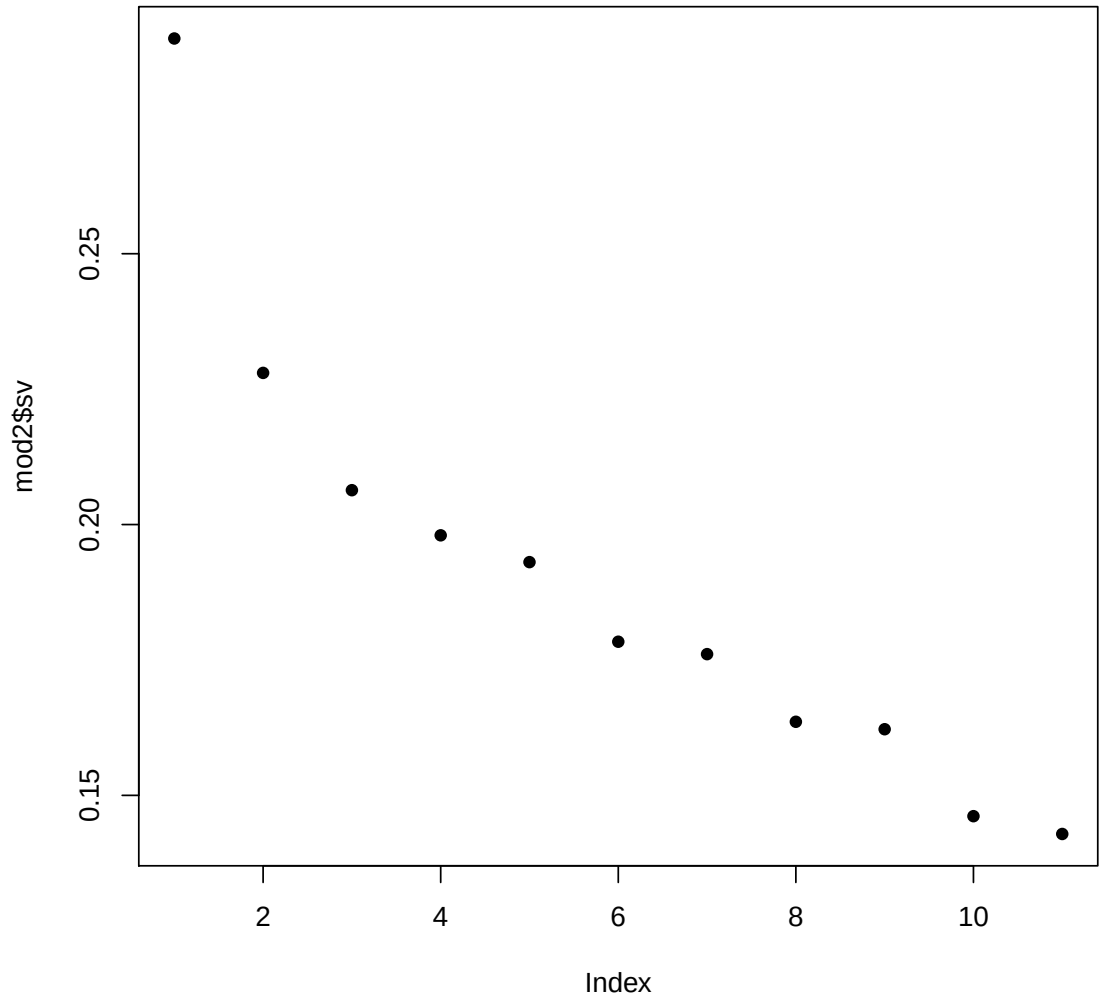
Basically the same, and it's much quicker to fit too...

The `summary` method is pretty comprehensive, though you'll probably want to read some of Greenacre 2007 to make the most of it.

```
summary(mod2)
```

Since we've got multiple dimensions we can check how much variation is being explained in each. What the slides called σ is related to the singular values of the underlying SVD, which we can get from the model. Let's plot these

```
plot(mod2$sv, pch=16)
```



The ‘elbow’ after the first dimension is one (fallible) reason to think that this debate is mostly one dimensional. That is at least theoretically plausible.

If you want to see a biplot of all the words and documents, then

```
plot(mod2)
```

but be warned. It’s big. . . You may want to read the help page to see how to only show some elements, or to change the colors.