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

```
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: 0.081 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

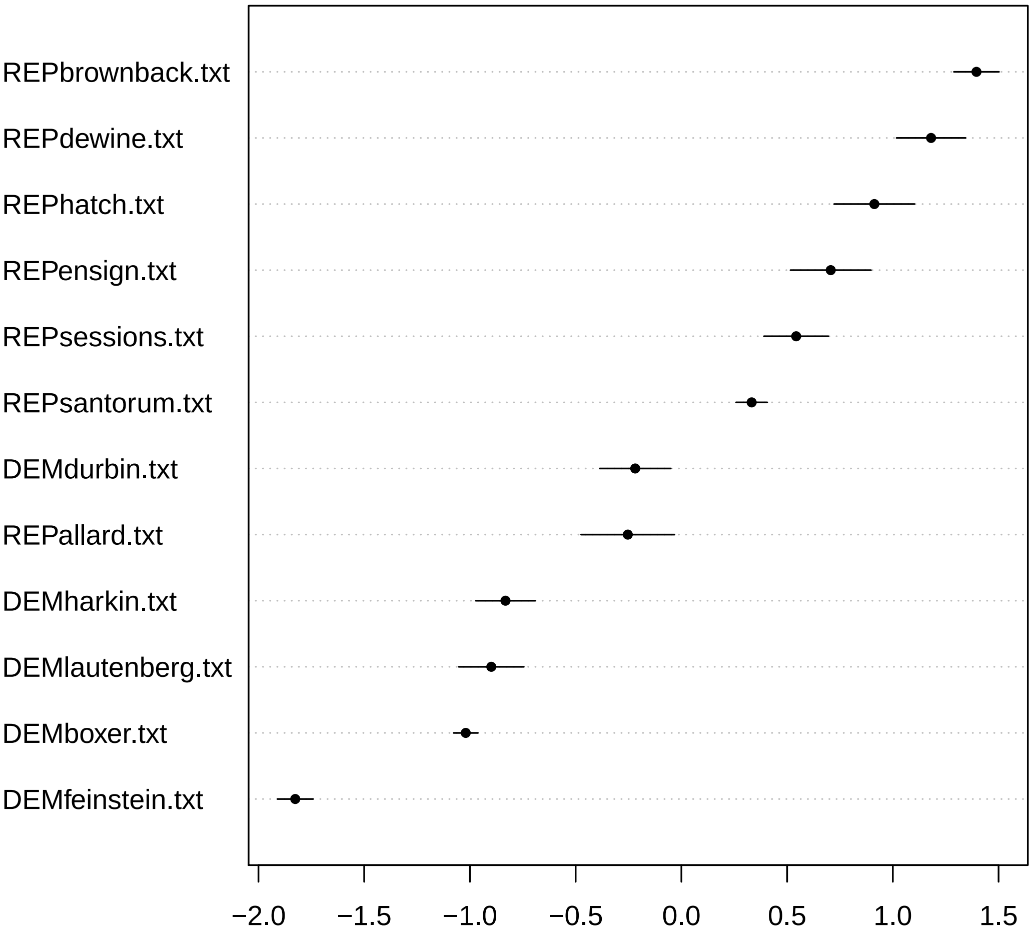
```
mod <- wordfish(senatewfm)
summary(mod)


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
```

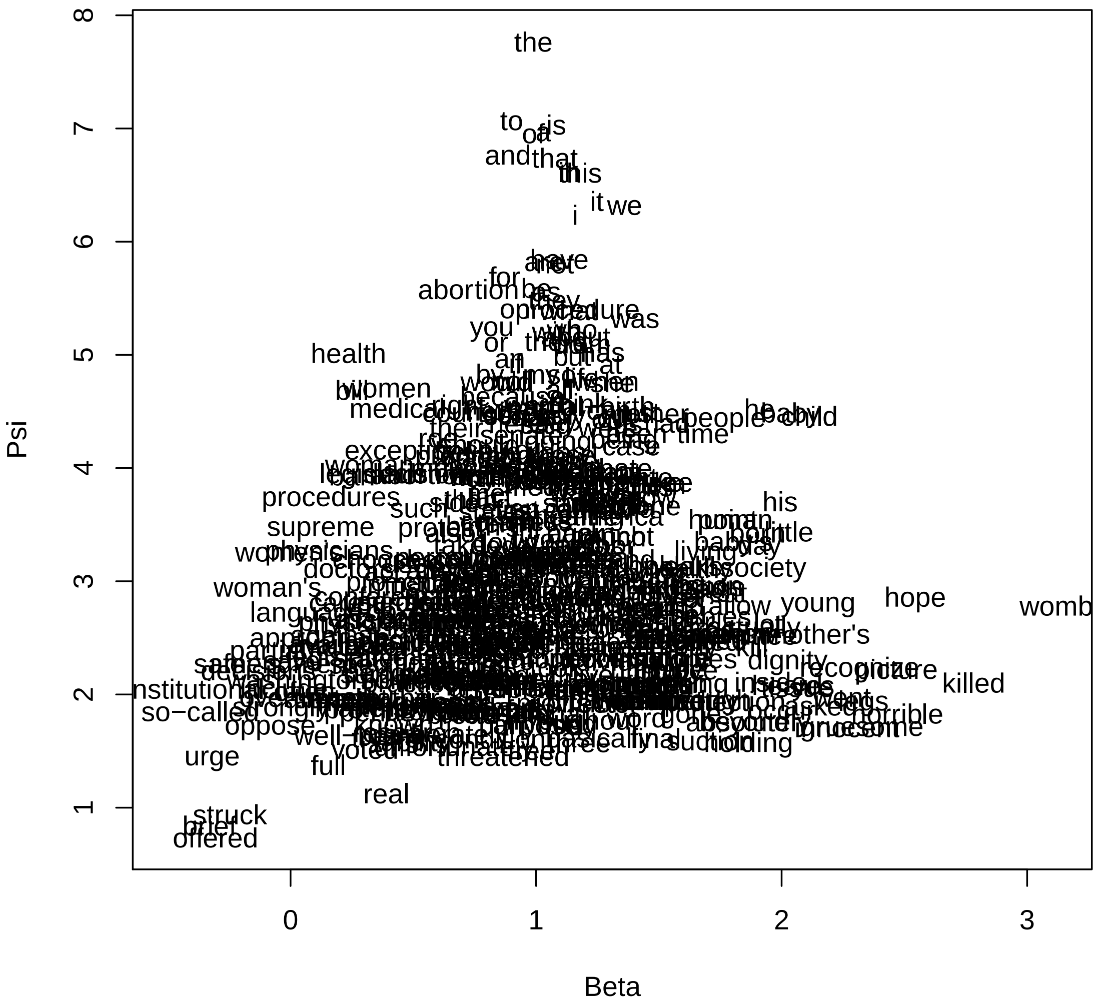
Table summaries are nice but plots are better

```r
plot(mod)
```



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

```r
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.

```r
wdparams <- coef(mod, 'poisson')$words ## just take the word parameters
length(wdparams$beta)


[1] 498
```

There's quite a few. Let's choose some likely candidates

```
wds <- c("life", "choice",
         "womb", "her", "woman", "health","born","baby",
         "gruesome","kill")
mywords <- wdparams[wds,]
mywords[order(mywords$beta),] ## in left to right order


              beta      psi
health   0.2357655 5.014492
woman    0.3215173 4.009126
choice   0.5792851 3.192181
her      0.7932154 4.506371
life     1.1863159 4.797663
kill     1.8788762 2.424454
born     1.9020905 3.428362
baby     2.0359070 4.464929
gruesome 2.3263387 1.672686
womb     3.1188710 2.783132
```

Do the estimates make sense? Again, it's easier to see if we plot them, as in Fig. 1.

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

```
dotchart(mywords$beta, rownames(mywords),pch=19)
```

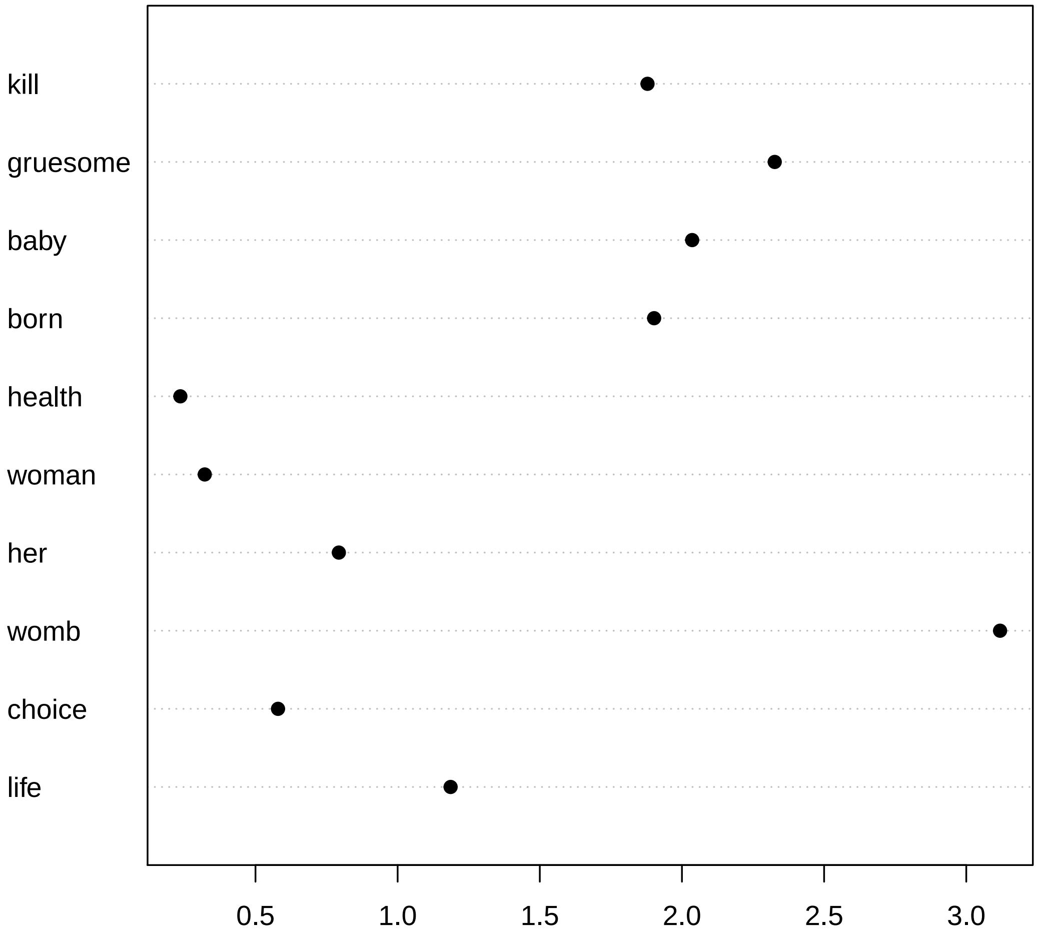
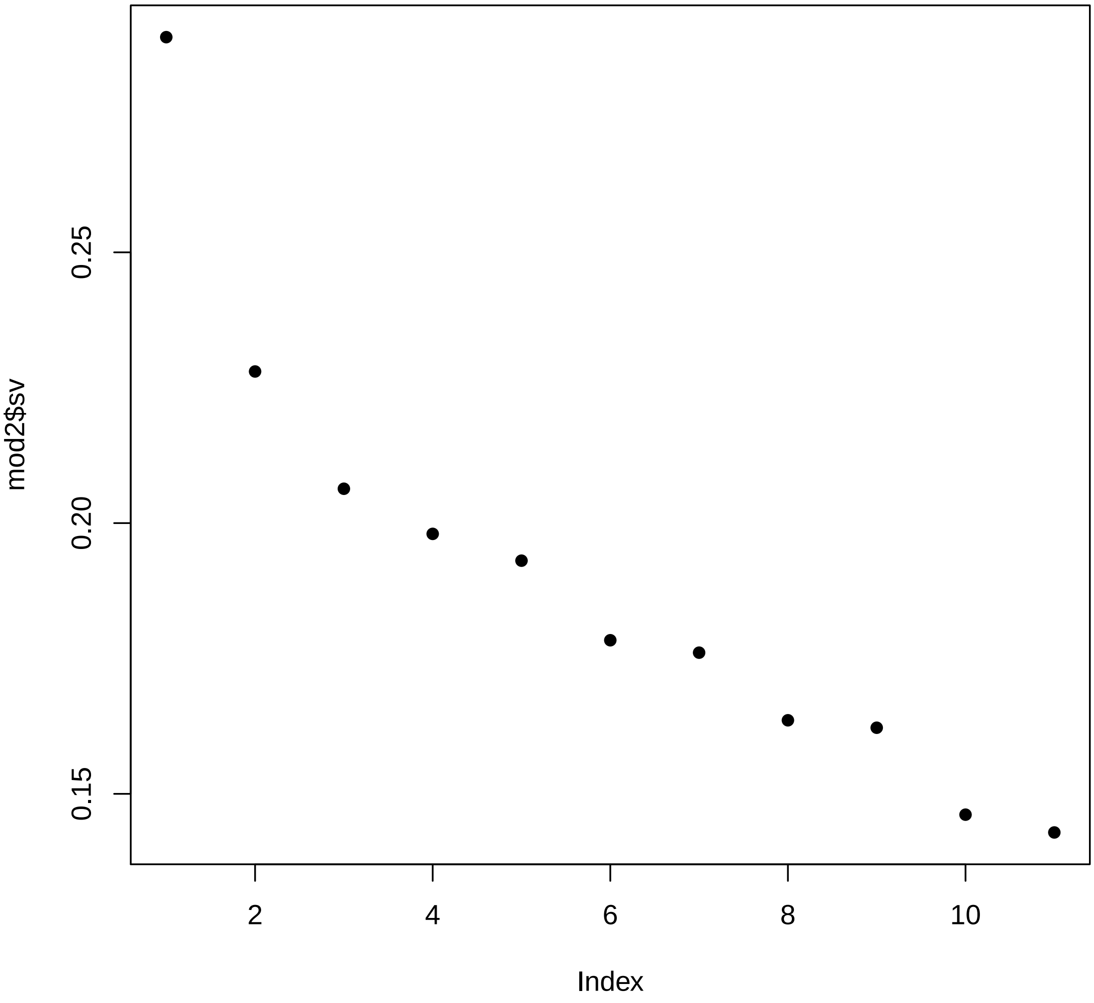


Figure 1: A chart of word slopes (sensitivity to ideological position) for the word stems.

```
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.