Lab 1

IQMR 2016

1 Introduction to quanteda

Building a corpus

Let's start by reading in the data from the abortion debate analyzed by Bara et al. I've concatenated each speaker's contributions into a single file. (This is certainly not the only way to think about analyzing this data, but it's what Bara et al. did.)

First load the package

```
library(quanteda)
```

```
quanteda version 0.9.7.13
```

```
Attaching package: 'quanteda'
```

```
The following object is masked from 'package:base':

    sample
```

then read in some text files and make a `corpus` from them

```
txts <- textfile("data/abortion-debate-by-speaker/*")
corp <- corpus(txts)
```

Corpora get big quickly, so most functions in the package will not show you all the contents of any object. Call `summary` to get a view of the new corpus object.

It's helpful to add some metadata to the documents, so we can subset them. Here we'll record the vote of each speaker.

```
vote <- rep("yes", 24) ## 16 voted yes
vote[1:3] <- "abs"     ## 3 abstained
vote[4:8] <- "no"      ## 5 voted no
docvars(corp, "vote") <- vote
summary(corp)
```

```
Corpus consisting of 24 documents.

                           Text Types Tokens Sentences vote
          abs-Dr Horace King.txt   104    177        17  abs
       abs-Mr William Deedes.txt   530   1670        81  abs
         abs-Sir John Hobson.txt   717   3108       128  abs
        no-Mr Kevin McNamara.txt   909   3365       135   no
```

```
 no-Mr Norman St John-Stevas.txt   720   2507       118   no
           no-Mr Peter Mahon.txt    57     79         6   no
         no-Mr William Wells.txt   780   2939       140   no
          no-Mrs Jill Knight.txt   765   2642       116   no
           yes-Dr David Owen.txt   588   1818        82  yes
        yes-Dr John Dunwoody.txt   569   2050        81  yes
   yes-Dr Michael Winstanley.txt    49     68         3  yes
        yes-Hon. Sam Silkin.txt    30     34         2  yes
       yes-Miss Joan Vickers.txt   618   2039        92  yes
            yes-Mr Alex Lyon.txt    51     59         3  yes
          yes-Mr Angus Maude.txt   690   2515        93  yes
      yes-Mr Charles Pannell.txt    95    175        11  yes
          yes-Mr David Steel.txt  1160   5268       191  yes
         yes-Mr Edward Lyons.txt   411    860        40  yes
       yes-Mr John Mendelson.txt    48     63         1  yes
            yes-Mr Leo Abse.txt   733   2385        92  yes
          yes-Mr Roy Jenkins.txt   728   2727       102  yes
    yes-Mrs Gwyneth Dunwoody.txt    60     91         4  yes
         yes-Mrs Renee Short.txt   707   2382        91  yes
  yes-Sir Henry Legge-Bourke.txt    68    103         5  yes

Source:  /Users/will/wip/iqmr/session1/lab1/* on x86_64 by will
Created: Mon Jun 20 20:33:56 2016
Notes:
```

where `docvars` adds document specific metadata, here the speaker's vote.

If we want to get the texts *out* of this corpus object we use

```
texts(corp)
```

To just get the contributions of the delightfully named Mr Norman St John-Stevas, we can index into it

```
texts(corp)[5]
```

To see just a few speakers, e.g. the ones that voted against, we can use `subset`

```
nocorp <- subset(corp, vote == "no")
summary(nocorp)
```

```
Corpus consisting of 5 documents.

                          Text Types Tokens Sentences vote
       no-Mr Kevin McNamara.txt   909   3365       135   no
 no-Mr Norman St John-Stevas.txt   720   2507       118   no
           no-Mr Peter Mahon.txt    57     79         6   no
         no-Mr William Wells.txt   780   2939       140   no
          no-Mrs Jill Knight.txt   765   2642       116   no

Source:  /Users/will/wip/iqmr/session1/lab1/* on x86_64 by will
Created: Mon Jun 20 20:33:56 2016
Notes:
```

Exploring text corpora

Let's check to see if there is key terminology that we should be looking out for. One way to do this is to look for collocations. These are word combinations that occur more often than we would expect from

their individual frequencies. Here's the top 40.

```
collocations(corp, n=40)
```

```
      word1      word2 word3 count       G2
 1:      it         is         196 848.3238
 2:     the       bill         211 807.2171
 3:  member        for          88 749.4416
 4:      of        the         386 712.1012
 5:       i         am          73 531.4851
 6:      do        not          66 457.9654
 7: medical profession          36 403.1270
 8:  second    reading          28 387.1623
 9:   there         is          79 363.6020
10:     the      house          83 363.0760
11:  clause          1          25 352.0836
12:   think       that          63 335.7217
13: carried        out          25 310.6485
14:       i      think          59 310.0138
15:  should         be          58 293.2631
16:   would         be          56 264.8196
17:     has       been          37 260.7602
18:   those        who          33 256.9848
19:    have       been          41 254.4129
20:      my        hon          40 251.4309
21:   there        are          44 238.9961
22:    that         it          91 237.2256
23:       i       hope          34 236.7005
24:       i       have          71 232.4643
25:     may         be          41 226.2328
26:   royal    college          15 224.6796
27:      in        the         169 222.7823
28:     for   roxburgh          24 220.7437
29:       i    believe          36 219.1823
30: illegal  abortions          18 209.8923
31:     the        hon          89 204.6584
32:     per       cent          11 197.1157
33:   david      steel          12 189.9803
34:       i         do          46 188.8939
35: selkirk        and          24 187.0869
36:     and    learned          28 186.2599
37: believe       that          33 184.4400
38:    will         be          41 184.4032
39:    does        not          27 184.0270
40:   right        hon          28 183.8125
      word1      word2 word3 count       G2
```

This reminds us that some key terms are `royal college` (of surgeons), `illegal abortions`, `medical profession`, and the more procedural `right hon` and `second reading`.

This works better, the larger the corpus. You can tweak the results by changing the statistic used to score word pairs. If you have a little time on your hands and a corpus larger than this one then you can also look for word triples, although finding them is a bit more computationally intensive.

Here's an example of using pointwise mutual information (`pmi`) rather than a likelihood ratio test (`lr`) to find three word collocations

```
collocations(corp, size=3, n=40, method="pmi")
```

The first couple of terms remind us that this debate happened in the Sixties…

Keywords in context

Since this is an abortion debate, let's see the honourable folk talk about mothers and babies. We'll use the keyword in context function

```
kwic(corp, "mother*")
```

we might benefit from a bit more local context, so maybe set the window a bit wider Here are the babies

```
kwic(corp, "babi*", window=10)
```

you may need to expand your window a bit to see these properly.

Perhaps oddly, there is much less talk of babies than of mothers. In this debate, the other major actors are doctors and their professional association, which you can investigate the same way.

Constructing a document feature matrix

One of the first things we tend to do to a set of documents as preparation from modeling is to make a document feature matrix (`dfm`)

```
corpdfm <- dfm(corp)


Creating a dfm from a corpus ...
   ... lowercasing
   ... tokenizing
   ... indexing documents: 24 documents
   ... indexing features: 3,542 feature types
   ... created a 24 x 3543 sparse dfm
   ... complete.
Elapsed time: 0.069 seconds.


dim(corpdfm)


[1]   24 3543
```

Typically though, we'll want to trim the low frequency and idiosyncratic terms out

```
corpdfm2 <- trim(corpdfm, minCount=5, minDoc=5)


Removing features occurring fewer than 5 times: 2720
Removing features occurring in fewer than 5 documents: 2939


dim(corpdfm2)


[1]  24 604
```

which makes it a fair bit smaller.

There's also a `wordcloud` function for viewing the the document feature matrix, but we won't use it because wordclouds are silly.

Answering questions with text

Now let's prod these documents in a more substantively focused way.

In the debate the Speaker, Mr Horace King, said he would try to give equal time to both sides of the debate. (You can read the original debate as `data/abortion-debate-hansard.html`). Did it happen this way?

It's hard to know whether the debate was persuasive since we do not know the speakers prior beliefs (though we could find out from their previous debates) so let us assume that there was no substantial persuasion. We'll also assume that no speaker spoke particularly slowly. These imply that we can proxy speaking time with number of words said.

```
speakingtime <- rowSums(corpdfm)
speakingtime
```

```
         abs-Dr Horace King.txt       abs-Mr William Deedes.txt         abs-Sir John Hobson.txt
                            147                            1472                            2800
       no-Mr Kevin McNamara.txt no-Mr Norman St John-Stevas.txt           no-Mr Peter Mahon.txt
                           2984                            2238                              70
        no-Mr William Wells.txt          no-Mrs Jill Knight.txt           yes-Dr David Owen.txt
                           2599                            2351                            1647
       yes-Dr John Dunwoody.txt   yes-Dr Michael Winstanley.txt         yes-Hon. Sam Silkin.txt
                           1882                              60                              30
      yes-Miss Joan Vickers.txt            yes-Mr Alex Lyon.txt          yes-Mr Angus Maude.txt
                           1795                              53                            2257
     yes-Mr Charles Pannell.txt          yes-Mr David Steel.txt         yes-Mr Edward Lyons.txt
                            154                            4766                             756
      yes-Mr John Mendelson.txt             yes-Mr Leo Abse.txt          yes-Mr Roy Jenkins.txt
                             57                            2153                            2430
   yes-Mrs Gwyneth Dunwoody.txt         yes-Mrs Renee Short.txt  yes-Sir Henry Legge-Bourke.txt
                             82                            2135                              88
```

Now to break this down by final vote

```
aggregate(speakingtime ~ docvars(corp, "vote"), FUN=sum)
```

```
  docvars(corp, "vote") speakingtime
1                   abs         4419
2                    no        10242
3                   yes        20345
```

It appears that floor time was about 30% eventual no voters and 60% eventual yes voters. However, individual no voters did get on average more time each

```
aggregate(speakingtime ~ docvars(corp, "vote"), FUN=mean)
```

```
  docvars(corp, "vote") speakingtime
1                   abs     1473.000
2                    no     2048.400
3                   yes     1271.562
```

Applying a content analysis dictionary

Let's turn to the content analysis dictionary that Bara used. A content analysis dictionary in `quanteda` terms can be a regular R list of words. (It can also import Wordstat and LIWC format files, if you made a

dictionary in one of those packages). Here we'll just read in the Bara dictionary to be the right kind of list.

I have a copy of the dictionary in Yoshikoder format, an XML format that can be rather easily parsed by the `rvest` web scraping package. So I'll use that. If you have a different format you might have to wrote this yourself (or you can ask us how). It's not pretty, but for the record

```
library(rvest)
```

```
Loading required package: xml2
```

```
dic <- read_html("data/2007_abortion_dictionary.ykd")
cats <- html_nodes(dic, "cnode cnode") # top level categories
getwords <- function(x){ html_attr(html_nodes(x, "pnode"), "name") }
baradic <- lapply(cats, getwords)
names(baradic) <- html_attr(cats, "name")
```

We take this simple list and turn it into a quanteda `dictionary` object

```
bara <- dictionary(baradic)
```

Replicating a little bit of Bara

With dictionary in hand we can now go *category* counting rather than word counting

```
baradfm <- dfm(corp, dictionary=bara)
```

```
Creating a dfm from a corpus ...
   ... lowercasing
   ... tokenizing
   ... indexing documents: 24 documents
   ... indexing features: 3,542 feature types
   ... applying a dictionary consisting of 6 keys
   ... created a 24 x 6 sparse dfm
   ... complete.
Elapsed time: 0.399 seconds.
```

Since this output is not absolutely massive

```
dim(baradfm)
```

```
[1] 24  6
```

let's force it into a regular R matrix to take a look at the whole thing without being swamped in elements

```
dictout <- as.matrix(baradfm)
dictout
```

```
                              features
docs                           advocacy legal medical moral procedural social
  abs-Dr Horace King.txt              3     0       0     0         23      1
  abs-Mr William Deedes.txt          36    11      28    14         87     25
```

```
  abs-Sir John Hobson.txt              36    22      65     9       144     72
  no-Mr Kevin McNamara.txt             68    38      94     8       125     71
  no-Mr Norman St John-Stevas.txt      63    26      50    32       112     35
  no-Mr Peter Mahon.txt                 1     0       1     1         8      2
  no-Mr William Wells.txt              61    24      73    21       128     47
  no-Mrs Jill Knight.txt               49    29      62    21        62     85
  yes-Dr David Owen.txt                42     5      68     8        71     51
  yes-Dr John Dunwoody.txt             42    23      65     9        86     70
  yes-Dr Michael Winstanley.txt         1     2       0     0         3      0
  yes-Hon. Sam Silkin.txt               0     0       0     1         6      0
  yes-Miss Joan Vickers.txt            39    10      54     4       107     76
  yes-Mr Alex Lyon.txt                  1     2       2     1         0      0
  yes-Mr Angus Maude.txt               38    16      48    23       100     58
  yes-Mr Charles Pannell.txt            5     0       7     0        16      6
  yes-Mr David Steel.txt               87    76     119    56       177     89
  yes-Mr Edward Lyons.txt              10    10      40     4        30     27
  yes-Mr John Mendelson.txt             6     0       0     0         9      0
  yes-Mr Leo Abse.txt                  47    14      44    12       103     69
  yes-Mr Roy Jenkins.txt               45    27      46     9       148     34
  yes-Mrs Gwyneth Dunwoody.txt          1     0       0     0         5      7
  yes-Mrs Renee Short.txt              28    52      80     6        90     78
  yes-Sir Henry Legge-Bourke.txt        0     0       0     0         4      6
```

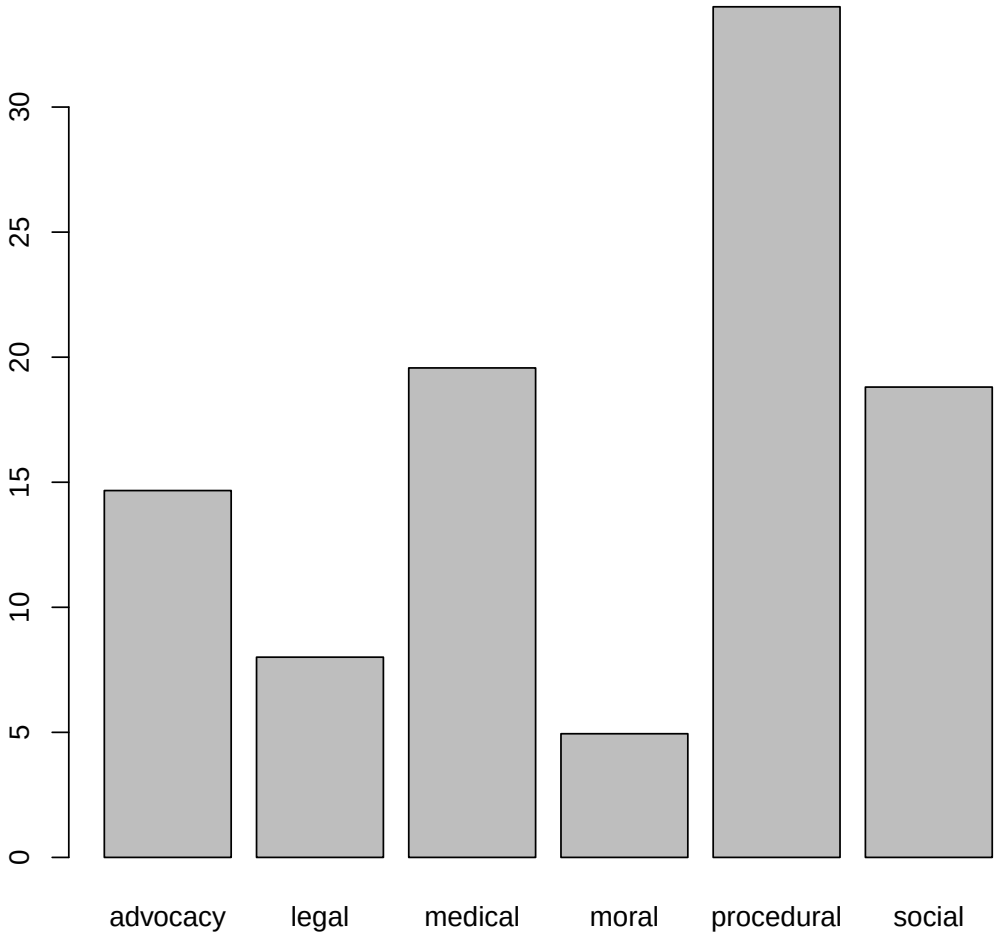
And recreate some of Bara et al.'s Table 3, only as a bar plot.

Table 3: Mean percentage vocabulary* use by 14 major speakers, July 1966 Second Reading debate

	Advocacy vocabulary	Legal vocabulary	Medical vocabulary	Moral vocabulary	Rhetoric of debate vocabulary	Social vocabulary
Mean	13.59	7.82	21.71	4.61	32.17	20.09
Standard deviation	2.98	3.36	4.73	2.51	6.94	4.86

Note: *as % total dictionary present.

```
emph <- colSums(dictout) ## emphasis
propemph <- emph / sum(emph)*100 ## relative emphasis as a percentage
barplot(propemph)
```



Finally, let's revisit the floortime question but this time counting only vocabulary that Bara et al. thought was substantively relevant.

```
relevanttalk <- rowSums(dictout)
aggregate(relevanttalk ~ docvars(corp, "vote"), FUN=sum)
```

```
  docvars(corp, "vote") relevanttalk
1                   abs          576
2                    no         1397
3                   yes         2861
```

Now the balance of floor time spent saying 'relevant' words is even more skewed, at around 3 to 1.