Q-STEP R 'HOW TO' GUIDES:

COMPUTATIONAL TEXT ANALYSIS USING THE QUANTEDA PACKAGE

Creator: Dr James Weinberg

As social scientists, we often explore complex problems through case studies, interviews, policy documents, or focus groups. In each case, these methods provide a wealth of qualitative data to analyse, but the process can be laborious and subject to significant human error. However, it is also possible to analyse these data using quantitative methods from the field of computational text analysis. This is an exciting methodology that remains largely untapped as a resource because of the software-specific expertise required. By contrast to researcher-led coding and content analysis by hand or in software such as NVivo, computational analyses in syntax-based software such as R allow researchers to interrogate larger volumes of qualitative data in greater depth and to present interesting comparative trends across documents.

In this 'how to' guide, you will learn how to use the 'quanteda' package in R to conduct 5 sets of analyses. The examples used in this article are based on a collection of policy documents, Government reports and curriculum guidance, and focus groups about citizenship education policy in the United Kingdom. These analyses and the surrounding discussions can be found in the following forthcoming article:

Weinberg, J., Mills, S., and Keating, A. (In Preparation). The Future of Citizenship Education: A Comparison of Stakeholder and Government Responses to Policy Recommendations in the House of Lord's Select Committee Report on Citizenship and Civic Engagement.

Step 1: Data Preparation

As with quantitative analysis of numerical survey data, it is important to prepare your data properly before you begin textual analysis in R. Before we look at what you need to do, take a minute to install and/or load the following packages: quanteda, readtext, topicmodels, SentimentAnalysis, ggplot2, tm, viridis, and RColorBrewer.

Before you reach this point, you should have identified a clear research question and collated documents that relate to it. You can then isolate these in a zip file in a safe location and load your working directory accordingly. Textual data can be stored in a variety of formats and it is likely that you may have documents in more than one of these, e.g. pdf, word, html. Although R natively supports flat text files such as csv. and txt., you will need to use the 'readtext' package to import many different types of textual data in a uniform format. The 'readtext' function will automatically import all available files from a predetermined file path. You can then use the 'corpus' function to tie your texts together.

Q-Step A step-change in quantitative social science skills. Funded by the Nuffield Foundation, ESRC and HEFCE. Nuffield Foundation. Sheffield Methods Institute. The University Of Sheffield.

```
filepath <- "E:/_Civic Education/APPG and PLOG/Future of Cit Ed - Data and Wr
ite Up/Lords Report Analysis - All Groups.zip"

rt <- readtext(filepath, text_field = "texts")
full_corpus <- corpus(rt) # create quanteda corpus
summary(full_corpus)

## Corpus consisting of 12 documents:
##
##                                                           Text Types Tokens
##  Evidence - GCSE_subject_content_for_citizenship_studies.pdf   760   2918
##   Evidence - SECONDARY_national_curriculum_-_Citizenship.pdf   321    857
##                                   Focus Groups - Cit Ed.docx  1948  17856
##                                      Focus Groups - NCS.docx  1276   8496
##                               Government Evidence Cit Ed.docx   427   1448
##                                  Government Evidence NCS.docx   288    830
##                             Government Response - Cit Ed.docx   534   1687
##                                Government Response - NCS.docx   438   1259
##                                   Lords report - Cit Ed.docx   982   3464
##                                      Lords report - NCS.docx   472   1333
##                           Stakeholders - Citizenship Ed.docx  2769  19659
##                                      Stakeholders - NCS.docx  2031  11321
##  Sentences
##         31
##         16
##        857
##        434
##         44
##         35
##         63
##         40
##        132
##         59
##        635
##        406
```

For some of the analyses that follow, you may want to isolate particular texts in your corpus. To do this, you can add an extra document variable - essentially another identifier for each text. In this instance I add a number to each document and then use the 'corpus_subset' functions to create two new subsets in my data.

```
## Add a new document variable so that you can subset the data for analysis

docvars(full_corpus, "Document") <- 1:12
summary(full_corpus)

## Corpus consisting of 12 documents:
##
##                                                           Text Types Tokens
##  Evidence - GCSE_subject_content_for_citizenship_studies.pdf   760   2918
##   Evidence - SECONDARY_national_curriculum_-_Citizenship.pdf   321    857
##                                     Focus Groups - Cit Ed.docx  1948  17856
##                                        Focus Groups - NCS.docx  1276   8496
##                                 Government Evidence Cit Ed.docx   427   1448
##                                    Government Evidence NCS.docx   288    830
##                               Government Response - Cit Ed.docx   534   1687
##                                  Government Response - NCS.docx   438   1259
##                                     Lords report - Cit Ed.docx   982   3464
##                                        Lords report - NCS.docx   472   1333
##                          Stakeholders - Citizenship Ed.docx  2769  19659
##                                        Stakeholders - NCS.docx  2031  11321
##  Sentences Document
##         31        1
##         16        2
##        857        3
##        434        4
##         44        5
##         35        6
##         63        7
##         40        8
##        132        9
##         59       10
##        635       11
##        406       12
```

```
## Create desired subsets

corpus_FGDFE1 = corpus_subset(full_corpus, Document %in% c("1", "3"), select
= Document)
summary(corpus_FGDFE1)

## Corpus consisting of 2 documents:
##
##                                                          Text Types Tokens
##  Evidence - GCSE_subject_content_for_citizenship_studies.pdf   760   2918
##                                     Focus Groups - Cit Ed.docx  1948  17856
##  Sentences Document
##         31        1
##        857        3
```

Before analysing your texts, you will also need to conduct a number of pre-processing tasks. For example, you can tokenise your texts (splitting them into single words as units of analysis); normalise each document to modify uppercase letters and strip symbols that might impact the computer's ability to recognise words with the same spellings and meanings; and remove 'stopwords' (common words in the English language that are not analytically informative). Not only do these processes reduce the comutational load of the data, they also improve the accuracy of analyses by reducing the size of the dataset and filtering words or symbols or little or no relevance. In quanteda, the 'stopwords' function

returns a character vector of stopwords for a given language, but it is also possible to create your own list of stopwords and read these into R before you pre-process your texts.
You can pre-process your texts at the same time that you turn your corpus into a document-term matrix. A DTM is the most common way to represent a corpus of texts in a 'bag-of-words' format. In other words, it turns your texts into matrix where rows are documents, columns are tokens or terms, and cells indicate how many times each term occurs in each document. The 'dfm' function in quanteda allows you to transform your corpus with one command, at the same time as pre-processing the data.

```
## Import your own stopwords list (remember to encode your list as UTF-8)

stopwords <- readLines("E:/_Civic Education/APPG and PLOG/Future of Cit Ed -
Data and Write Up/Englishstopwords.txt", encoding = "UTF-8")

## Or alternatively load quanteda's own list of stopwords

sw <- stopwords("english")
head(sw)

## [1] "i"      "me"     "my"     "myself" "we"     "our"

## Turn your corpus into a DTM and pre-process

dtm <- dfm(full_corpus, tolower = TRUE, remove_numbers = TRUE,
           remove_punct = TRUE, remove_separators = TRUE,
           stem = FALSE, remove = stopwords)
dtm

## Document-feature matrix of: 12 documents, 4,188 features (81.9% sparse).
```

Step 2: Analysis

Once you have processed your data, you can begin analysing it. As per the advice of Boumans and Trilling (2016), I recommend using a mixture of counting and dictionary methods, supervised machine learning, and unsupervised machine learning. By following this scheme, you can use complementary deductive analyses (where you specify a priori what the computer is looking for in the texts) and inductive analyses (where you might not know what you're looking for exactly, and let the computer extract meaningful trends). I will talk through 5 of these different analyses.

1. Counting and Dictionary Analysis

You can use the dictionary approach to count how often particular words or phrases - usually those related to key concepts in your research - occur in each of your texts. The first step is create a dictionary object (here called myDict) which uses the 'dictionary' function. In this instance, I create a dictionary of two key concepts (active citizenship and character) from academic debates about what should be taught in citizenship studies in schools. Each concept is comprised of a list of indicators that I have theoretically devised beforehand. The 'dfm_lookup' function is then used to apply the dictionary to my DTM in the same way that you might hand-code a document. The output provides a new DTM in which columns represent your dictionary codes.

```
myDict <- dictionary(list(active_citizenship =
                            c("community", "engage", "engagement", "participa
te", "participation", "democracy", "inclusive", "democratic", "active", "coop
erate", "cooperation", "respect", "civic", "active", "politics", "openminded"
, "openness", "dignity", "rights", "responsibilities", "responsibility", "gov
ernment", "democracy", "justice", "fairness", "equality", "tolerance", "diver
sity", "culture", "religion",                               "global", "envrion
ment", "sustainability", "interpretation",                "ethic", "ethical", "
interaction", "cooperation", "questioning",
"collective", "power", "structure"),
                          character =
                            c("resilient", "resilience", "work", "character",
                              "development", "develop", "improve",
                              "improvement", "adult", "adulthood", "job",
                              "service", "autonomy", "critical", "curiosity",
                              "judgement", "reasoning", "reflection",
"resourcefulness",            "confidence", "determination", "motivation",
                              "perseverance", "resilience", "teamwork",
                              "neighbourliness", "service", "volunteering",
                              "compassion", "courage", "gratitude", "honesty"
, "humility",        "integrity", "justice", "respect")))

dict_dtm <- dfm_lookup(dtm, myDict, nomatch = "_unmatched")
show(dict_dtm)
```

Document	'Active Citizenship'	'Character Education'
	Overall frequency (relative proportion of all tokens in the document)	
(1) DfE – Subject guidance for CE GCSE	112 (8.3%)	16 (1.2%)
(2) DfE – Secondary curriculum for CE	33 (8.5%)	14 (3.6%)
(3) Policy recipients on CE	114 (2.4%)	40 (0.8%)
(4) Policy recipients on the NCS	40 (1.9%)	41 (1.9%)
(5) Government - Evidence to the Lords on CE	16 (3.2%)	6 (1.2%)
(6) Government - Evidence to the Lords on the NCS	10 (3.6%)	6 (2.2%)
(7) Lords report – Chp.3 on CE	36 (2.6%)	9 (0.6%)
(8) Lords report – Chp. 4 on NCS	20 (3.8%)	22 (4.2%)
(9) Stakeholders – Evidence to Lords on CE	377 (4.6%)	142 (1.7%)
(10) Stakeholders – Evidenceto Lords on the NCS	151 (3.4%)	123 (2.8%)

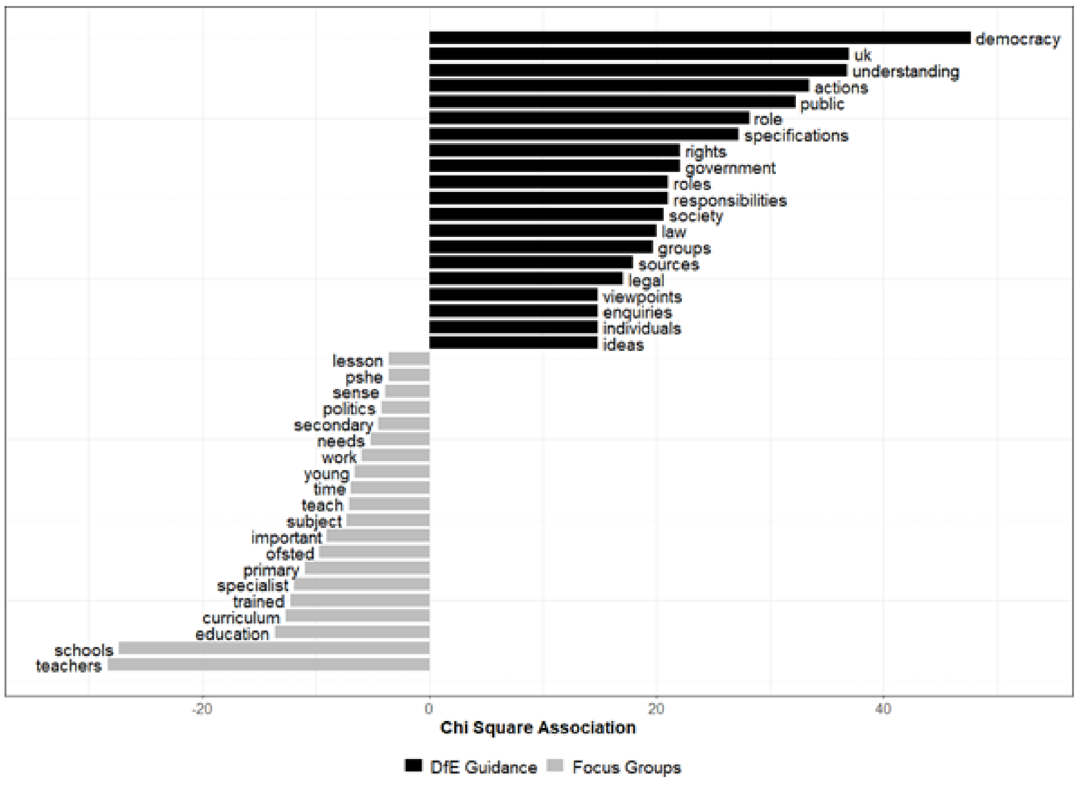
'Note: This table was reformatted in word after performing the initial analysis in R.

2. Keyness Plots

Various statistics can be used to describe and analyse a text corpus. One particularly useful technique is to compare the term frequencies of two corpora (i.e. two texts in your corpus or a partcular subset you have already created) and visualise these in a keyness plot. Not only can this provide a quick exploration of how different documents discuss a particular topic, but it can provide the stimulus to further discussions or inquiries.

In the following example I compare the guidance on citizenship studies issued to schools in England by the Department for Education (DfE) in 2015 and focus groups conducted with policy recipients. Keyness plots measure the chi-square (??^{2}) associations of individual tokens in these texts, thus determining differences between the expected frequencies and the observed frequencies of individual tokens. I use the ggplot2 package to customise R's native 'textplot' function. In this particular example, you can see that the words 'UK', 'government' and 'society' were used with greater frequency by the DfE than focus group participants, while 'curriculum', 'schools' and 'teachers' were more likely to be used by the focus group participants.

```
dtm_FGDFE1 = dfm(corpus_FGDFE1, groups = "Document",
                    remove = stopwords, remove_punct = TRUE, remove_numbers =
TRUE)
keyness = textstat_keyness(dtm_FGDFE1, target = "1")
textplot_keyness(keyness, show_reference = TRUE, show_legend = TRUE,
                 labelcolor = "black", labelsize = 5, font = NULL) +
  ggtitle ("Chi  Square Association of Frequent Terms") +
  xlab("Chi Square Association")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.text=element_text(size=14),
        legend.position="bottom") +
  scale_color_manual(labels = c("DfE Guidance", "Focus Groups"), values = c("
black", "grey")) +
  guides(color = guide_legend(""))
```



3. Latent Factors

Another inductive analysis is unsupervised machine learning, in which no coding rules are specified (unlike the dictionary function). Instead, an algorithm comes up with a model by identifying patterns in the texts. As a researcher, you can specify the parameters of this analysis, such as the number of categories used to classify terms in documents. The advantage of this technique is that it may produce groupings of terms that you, as the researcher, had not previously considered important or had not expected in the texts.

In this example, I use the 'topicmodels' package to extract 5 categories of related terms in the full corpus of documents. To increase the number of texts to model, and in order to reduce the size of the vocabulary, I first split the texts by paragraphs and remove terms with a frequency of less than 5.

```
texts = corpus_reshape(full_corpus, to = "paragraphs")

par_dtm <- dfm(texts, stem = FALSE, # create a document-term matrix with usua
l pre-processing
               remove_punct = TRUE, remove = stopwords,
               remove_numbers = TRUE)

par_dtm <- dfm_trim(par_dtm, min_termfreq = 5) # remove rare terms
par_dtm <- convert(par_dtm, to = "topicmodels") # convert to topicmodels form
at

set.seed(1)
lda_model <- topicmodels::LDA(par_dtm, method = "Gibbs", k = 5)
terms(lda_model, 8)

##      Topic 1     Topic 2       Topic 3       Topic 4     Topic 5
## [1,] "work"      "citizenship" "citizenship" "young"     "government"
## [2,] "schools"   "education"   "teachers"    "social"    "understanding"
## [3,] "community" "curriculum"  "schools"     "action"    "issues"
## [4,] "time"      "schools"     "education"   "programme" "democracy"
## [5,] "young"     "subject"     "primary"     "service"   "role"
## [6,] "different" "national"    "curriculum"  "citizen"   "citizens"
## [7,] "students"  "learning"    "subject"     "part"      "society"
## [8,] "active"    "political"   "specialist"  "local"     "skills"
```

4. Text Similarities

You may wish to start your analysis by looking at the overall similarity of the documents in your corpus. You can do this in quanteda using the 'textstat_simil' function, which looks for correlations between term frequencies in each document. In this example, I have started by weighting my texts to account for differences in document length. In this instance, I have used the 'prop' weight so that the computer uses the proportional frequency of terms in each document as opposed to their overall frequency.

```
mydtm <-  dfm_weight(dtm, "prop")

(sl <- textstat_simil(mydtm, method = "correlation", margin = "documents"))

Corr <- as.matrix(sl)

# Hide upper triangle
upper<- Corr
upper[upper.tri(Corr)]<-""
upper<-as.data.frame(upper)
summary(upper)

##  Evidence - GCSE_subject_content_for_citizenship_studies.pdf
##  0                :1
##  0.162290009132911:1
##  0.266368295630025:1
##  0.267258155125007:1
##  0.299764118345595:1
##  0.327373325837836:1
##  (Other)          :6
##  Evidence - SECONDARY_national_curriculum_-_Citizenship.pdf
##                   :1
##  0                :1
##  0.140279491300854:1
```

Sheffield
Methods
Institute.

```
##  0.158908549258068:1
##  0.178768052757388:1
##  0.207096982952784:1
##  (Other)          :6
##      Focus Groups - Cit Ed.docx      Focus Groups - NCS.docx
##                   :2                                :3
##  0                :1           0                    :1
##  0.154392026231859:1           0.346152137105964:1
##  0.349631715228474:1           0.37899693594797 :1
##  0.371175132654846:1           0.429623370581584:1
##  0.501926065101342:1           0.4868351311454 7 :1
##  (Other)          :5           (Other)          :4
##   Government Evidence Cit Ed.docx    Government Evidence NCS.docx
##                   :4                                  :5
##  0                :1              0                   :1
##  0.205734757935846:1              0.306049361232224:1
##  0.260375404028313:1              0.363392272316252:1
##  0.38288797169148 :1              0.380087716585415:1
##  0.387121481567471:1              0.38128993958719 :1
##  (Other)          :3              (Other)          :2
##  Government Response - Cit Ed.docx   Government Response - NCS.docx
##                   :6                                   :7
##  0                :1                0                  :1
##  0.215477505051008:1                0.141121468155891:1
##  0.369063318460564:1                0.251322318430208:1
##  0.539560957133171:1                0.412203977618608:1
##  0.760035657184936:1                0.667330399577106:1
##  0.776669765623733:1
##      Lords report - Cit Ed.docx      Lords report - NCS.docx
##                   :8                                :9
##  0                :1           0                    :1
##  0.340138459091588:1           0.498087716570306:1
##  0.560557279815752:1           0.600771908091996:1
##  0.890839878033076:1
##
##
##  Stakeholders - Citizenship Ed.docx Stakeholders - NCS.docx
##                   :10                 :11
##  0                : 1                0: 1
##  0.488967712928273: 1
##
##
##
##
```

5. Sentiment Analysis

Another angle you can take is to examine the tone of each text and thus get a general sense of the affective orientation of each document towards the topic of interest. The analysis I've conducted here uses the Lexicoder Sentiment Dictionary. The pre-populated dictionary consists of 2,858 “negative” sentiment words and 1,709 “positive” sentiment words, as well as a further set of 2,860 and 1,721 negations of negative and positive words. In this example, I start by examining the general sentiment of each document in the corpus, before narrowing the analysis to examine the sentiment attached to specific words of interest (or rather their stems). In the latter analysis, I specify a window of 5 words either side of the specified terms.

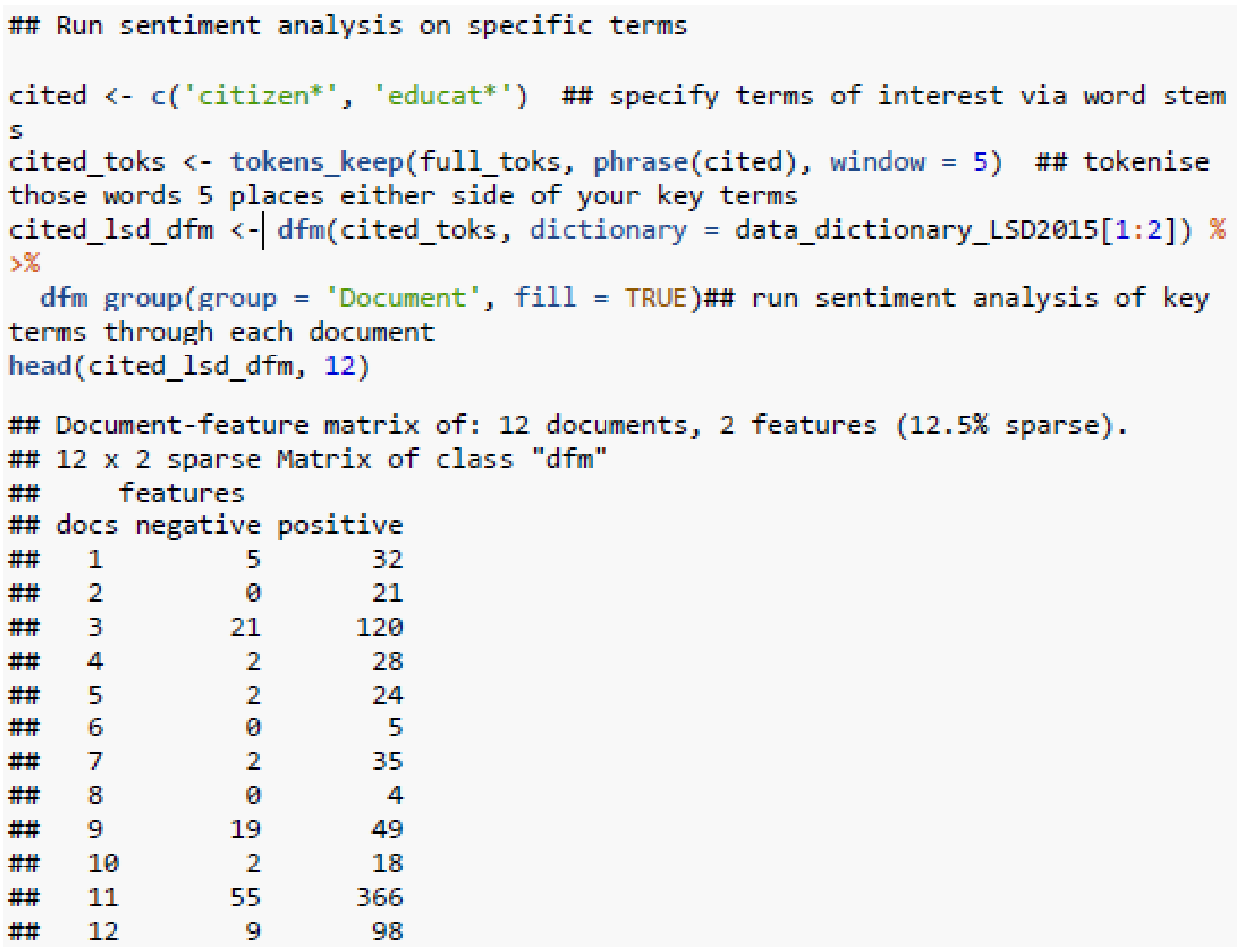
```
## Retokenise your corpus in a format recognised by the SentimentAnalysis pac
kage

full_toks <- tokens(full_corpus, remove_punct = TRUE,
                    remove_numbers = TRUE)

## Run sentiment analysis on the full corpus of documents

lsd_toks <- tokens_lookup(full_toks, data_dictionary_LSD2015[1:2])
lsd_dfm <- dfm(lsd_toks)
head(lsd_dfm, 12)

## Document-feature matrix of: 12 documents, 2 features (0% sparse).
## 12 x 2 sparse Matrix of class "dfm"
##                                                             features
## docs                                                         negative
##   Evidence - GCSE_subject_content_for_citizenship_studies.pdf       47
##   Evidence - SECONDARY_national_curriculum_-_Citizenship.pdf        11
##   Focus Groups - Cit Ed.docx                                       199
##   Focus Groups - NCS.docx                                           62
##   Government Evidence Cit Ed.docx                                    8
##   Government Evidence NCS.docx                                       5
##   Government Response - Cit Ed.docx                                  5
##   Government Response - NCS.docx                                     3
##   Lords report - Cit Ed.docx                                        51
##   Lords report - NCS.docx                                            4
##   Stakeholders - Citizenship Ed.docx                               233
##   Stakeholders - NCS.docx                                          108
##                                                             features
## docs                                                         positive
##   Evidence - GCSE_subject_content_for_citizenship_studies.pdf      170
##   Evidence - SECONDARY_national_curriculum_-_Citizenship.pdf        67
##   Focus Groups - Cit Ed.docx                                       706
##   Focus Groups - NCS.docx                                          296
##   Government Evidence Cit Ed.docx                                   68
##   Government Evidence NCS.docx                                      32
##   Government Response - Cit Ed.docx                                104
##   Government Response - NCS.docx                                    85
##   Lords report - Cit Ed.docx                                       123
##   Lords report - NCS.docx                                           71
##   Stakeholders - Citizenship Ed.docx                               992
##   Stakeholders - NCS.docx                                          507
```



```
## Run sentiment analysis on specific terms

cited <- c('citizen*', 'educat*')  ## specify terms of interest via word stem
s
cited_toks <- tokens_keep(full_toks, phrase(cited), window = 5)  ## tokenise
those words 5 places either side of your key terms
cited_lsd_dfm <- dfm(cited_toks, dictionary = data_dictionary_LSD2015[1:2]) %
>%
  dfm_group(group = 'Document', fill = TRUE)## run sentiment analysis of key
terms through each document
head(cited_lsd_dfm, 12)

## Document-feature matrix of: 12 documents, 2 features (12.5% sparse).
## 12 x 2 sparse Matrix of class "dfm"
##     features
## docs negative positive
##   1         5       32
##   2         0       21
##   3        21      120
##   4         2       28
##   5         2       24
##   6         0        5
##   7         2       35
##   8         0        4
##   9        19       49
##   10        2       18
##   11       55      366
##   12        9       98
```

Using text-as-data is an innovative way of expanding your research portfolio and extracting new insights from qualitative data. This guide is intended as an introduction to just 5 techniques available to researchers using the quanteda package in R. Other packages and other techniques are available and I urge researchers to explore these before commencing.

Useful References:

Boumans, J. W., & Trilling, D. (2016). Taking stock of the toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars. Digital Journalism, 4(1), 8-23.

Welbers, K., Van Atteveldt, W., & Benoit, K. (2017). Text Analysis in R. Communication Methods and Measures, 11(4), 245-265. http://www.tandfonline.com/doi/10.1080/19312458.2017.1387238