Practical 6: Latent Dirichlet Allocation

In this practical, we will process the text data that includes the abstracts of all papers in the *Journal of Statistical Software (JSS)*, up to 08/05/2010.

The JSS data is available as a list matrix in the package corpus.JSS.papers which can be installed and loaded by:

```
install.packages("corpus.JSS.papers", repos =
"http://datacube.wu.ac.at/", type = "source")
data("JSS_papers", package = "corpus.JSS.papers")
```

ID	Title	Creator	Subjects	Description	…
1	A Diagnostic to Assess the Fit of a Variogram Model to Spatial Data	Ronald Barry	c("Statistics", "Software")	The fit of a variogram model to spatially-distributed data is often difficult to assess. A graphical diagnostic written in S-plus is introduced that allows the user to determine both the general quality of the fit of a variogram model, and to find specific pairs of locations that do not have measurements that are consonant with the fitted variogram. It can help identify nonstationarity, outliers, and poor variogram fit in general. Simulated data sets and a set of soil nitrogen concentration data are examined using this graphical diagnostic.	
…	…				

I. Processing text data

In this section, we use the *tm* and *XML* package to process the JSS dataset.

1. Install the tm and XML packages.
2. We use only abstracts published up to 2010-08-05 and omit those containing non-ASCII characters in the abstracts.

```
JSS_papers <- JSS_papers[JSS_papers[,"date"] < "2010-08-05",]

JSS_papers <- JSS_papers[sapply(JSS_papers[, "description"],
Encoding) == "unknown",]

dim(JSS_papers)
```

3. The final data set contains 348 documents. Before analysis we transform it to a "Corpus" using package tm. HTML markup in the abstracts for greek letters, subscripting, etc., is removed using package XML. Install *tm*, *XML* and *SnowballC* packages to perform this task.

```
library("tm")
library("XML")

remove_HTML_markup <- function(s) tryCatch({
      doc <- htmlTreeParse(paste("", s), asText = TRUE, trim = FALSE)
 xmlValue(xmlRoot(doc))
}, error = function(s) s)

corpus <- Corpus(VectorSource(sapply(JSS_papers[, "description"],
remove_HTML_markup)))
```

4. The corpus is exported to a document-term matrix using function *DocumentTermMatrix()* from package *tm*. The terms are stemmed and the stop words, punctuation, numbers and terms of length less than 3 are removed using the control argument.

```
Sys.setlocale("LC_COLLATE", "C") #this is just to make sure we will have the
   same results

JSS_dtm <- DocumentTermMatrix(corpus, control = list(stemming =
TRUE, stopwords = TRUE, minWordLength = 3, removeNumbers = TRUE,
   removePunctuation = TRUE))
```

5. The mean term frequency-inverse document frequency (tf-idf) over documents containing this term is used to select the vocabulary. This measure allows to omit terms which have low frequency as well as those occurring in many documents. In this step, we need to install the package *stam*.

```
library("slam")

summary(col_sums(JSS_dtm))

term_tfidf <- tapply(JSS_dtm$v/row_sums(JSS_dtm)[JSS_dtm$i],
JSS_dtm$j, mean) * log2(nDocs(JSS_dtm)/col_sums(JSS_dtm > 0))

summary(term_tfidf)

JSS_dtm <- JSS_dtm[,term_tfidf >= 0.1]

JSS_dtm <- JSS_dtm[row_sums(JSS_dtm) > 0,]

summary(col_sums(JSS_dtm))
```

After this pre-processing we have the following document-term matrix with a reduced vocabulary which we can use to fit topic models.

```
dim(JSS_dtm)
```

II. Fitting the Latent Dirichlet Allocation (LDA) model

In this section, we fit an LDA model with 30 unknown topics to the dataset using the *topicmodels* package. We need to install the *topicmodels* package to perform this step.

```
   library("topicmodels")
   k <- 30
   SEED <- 2010
jss_TM <- LDA(JSS_dtm, k = k, method = "Gibbs", control =
list(seed = SEED, burnin = 1000, thin = 100, iter = 1000))
```

1. The most likely topic for each document is obtained by:

   Topic <- topics(jss_TM,1)
   Topic

2. The five most frequent terms of each topic

```
Terms <- terms(jss_TM, 5)
Terms[,1:5] #list the frequent terms of the first 5 topics
```

III. (This section is optional) Fitting and Visualising LDA model with lda and LDAVis packages

In this section, we use an alternative package, *lda*, for modelling the author-topic model. This package can be used together with the LDAVis package for visualising the results.

Please following the following link to complete the example:
http://cpsievert.github.io/LDAvis/reviews/reviews.html