Visualization and Learning Structure: Problem Session 7/15/16

Let's go back to our OkCupid profiles. There were ten optional essay questions. We might be interested in clustering profiles by the text and topics they use.

1. Download the profiles from our website and read into R.

   ```
   profiles<-read.csv("profiles.csv",header=T)
   ```

   Use `attach(profiles)` to create a variable for each column.

   We're going to be taking a look at the first essay: `essay0`: "My self-summary".

2. We first need to convert the essay texts into a corpus that can be used for clustering and word clouds. The code below looks at a sample of 100 profiles.

   ```
   essays<-Corpus(VectorSource(sample(essay0,100)))
   ```

   Take a look at your sampled essays to get an idea of what you have.

3. Now we'll convert them into a Document Term Matrix for clustering. The code uses the standard removals and conversions: Because some people left the essay question blank, there will be some empty rows that need to be removed.

   ```
   doc.term.m<-DocumentTermMatrix(essays,control=list(removePunctuation=TRUE,
   removeNumbers=TRUE, stopwords=TRUE,weighting=weightTfIdf))
   doc.term.m<-as.matrix(doc.term.m)
   row.sum<-rowSums(doc.term.m)
   doc.term.m<-doc.term.m[-which(row.sum==0),]
   ```

   What is the size of your matrix? `dim(doc.term.m)`

   Check out some of the words in your matrix using:

   ```
   word.list<-colnames(doc.term.m); n.word.list<-length(word.list)
   ```

4. Now we'll cluster the profiles. Download/install/load the spherical k-means library into R: `library(skmeans)`. It may ask you to download some additional support libraries.

   First, we can reduce the space by only clustering the 50 "best" words:

   ```
   tf.idf.sums<-colSums(doc.term.m)  ##summing up the tf-idf
   keep<-rev(order(tf.idf.sums))[1:50] ##keeping the ones with the biggest values
   ```

   We can use an elbow plot to choose the number of clusters (see class code). Here I'll just choose to look at four clusters. Adapt if you would like.

   ```
   skm.reduced<-skmeans(doc.term.m[,keep],4)
   ```

   (Check that `doc.term.m[,keep]` doesn't have any all zero rows that need to be removed)

5. Now we'll look at the most important words in each cluster. First we partition the profiles into the four groups and then sum up the tf-idf values to find each cluster's most important words.

```
cl1<-doc.term.m[skm.reduced$cl==1,]  ##getting the profiles in cluster 1
cl1.sums<-colSums(cl1)   ##summing up the tf-idf values for cluster 1
cl2<-doc.term.m[skm.reduced$cl==2,]; cl2.sums<-colSums(cl2)
cl3<-doc.term.m[skm.reduced$cl==3,]; cl3.sums<-colSums(cl3)
cl4<-doc.term.m[skm.reduced$cl==4,]; cl4.sums<-colSums(cl4)
```

   Looking at the 25 most important words in each cluster:

```
word.list[order(cl1.sums)[(n.word.list-25):n.word.list]]
word.list[order(cl2.sums)[(n.word.list-25):n.word.list]]
word.list[order(cl3.sums)[(n.word.list-25):n.word.list]]
word.list[order(cl4.sums)[(n.word.list-25):n.word.list]]
```

   Do you see any themes in the clusters?

6. Turning to word clouds, you'll need to download/install/load the wordcloud library in R: `library(wordcloud)`.

   Word clouds are build on frequency values (rather than TF-IDF value). The below code finds a TermDocument matrix (just the transpose of our DocumentTerm matrix) based on frequencies and then converts the matrix into the form needed for the word cloud function (essentially a list of words and freqs). Take a look at `help(wordcloud)` to see what other options you could change/add.

```
term.doc.m<-TermDocumentMatrix(essays,control=list(removePunctuation=TRUE,
removeNumbers=TRUE, stopwords=TRUE))
tdm<-as.matrix(term.doc.m)

#overall word cloud
v<-sort(rowSums(tdm),decreasing=TRUE)  ##finding total frequencies
d<-data.frame(word=names(v),freq=v)  ##creating a list of words and freqs
wordcloud(d$word, d$freq)
```

   We can also visualize each cluster's word cloud by subsetting the matrix:

```
tdm1<-tdm[,skm.reduced$cl==1]
v1<-sort(rowSums(tdm1),decreasing=TRUE)  ##finding total frequencies
d1<-data.frame(word=names(v1),freq=v1)  ##creating list of words and freqs
wordcloud(d1$word, d1$freq)
```

   Change all the 1's to 2's, 3's, and 4's to see the different clusters.

   Any themes emerge?