```
## This is my first attempt to look at term frequcies and topic models in our SGI
governance corpus

install.packages("devtools")
devtools::install_github("dgrtwo/widyr")
devtools::install_github("thomasp85/ggraph")
install.packages("tidyverse")
install.packages("tidytext")
install.packages("igraph")
install.packages("stringr")
install.packages("knitr")
install.packages("Rpoppler")
install.packages("tm")
#install.packages(c('SnowballC', 'wordcloud', 'topicmodels'))
install.packages('pdftools')
install.packages("ggplot2")
install.packages("filehash")
install.packages("dplyr")
install.packages("ggplot")
library(tidyr)
library(ggplot2)
library(ggplot)
library(tidyverse)
library(tidytext)
library(igraph)
library(ggraph)
library(stringr)
library(widyr)
library(knitr)
library(tm)
library(Rpoppler)
library(SnowballC)
library(wordcloud)
library(dplyr)
library(filehash)
library(pdftools)

#connect to pydio files /nfs/urbangi-data-- go to far right corner of
files/plots/packages window on the bottom right and click the box
# with ... and enter /nfs/urbangi-data in the popup box. Look for the urbangi-data
folder and the TestCorpus subfolder


#Used this process to create the test corpus:
https://eight2late.wordpress.com/2015/05/27/a-gentle-introduction-to-text-mining-using-r/
```

```r
#set network drive

#made a corpus
DEPcorp<-PCorpus(
  DirSource("test_DEP"),
  dbControl= list(dbName="DEPcorp.db"),
  readerControl=list(reader=readPDF(engine="Rpoppler")))



#checked to see if test corpus worked
inspect(DEPcorp[[1]])
#fix me! learn how to copy and read from the database to preserve original.

#tm map
test1 <- tm_map(DEPcorp, toSpace, "-")
test1 <- tm_map(DEPcorp, toSpace, ':')
test1 <- tm_map(DEPcorp, removePunctuation)
test1 <- tm_map(DEPcorp, content_transformer(tolower))
test1 <- tm_map(DEPcorp, removeWords,stopwords(kind = "en"))
test1 <- tm_map(DEPcorp, removeWords, c('will','the','however','may','via',
'since','and','in', 'a', 'use','dep','area'))
test1 <- tm_map(DEPcorp, stemDocument)
test1 <- tm_map(DEPcorp, removeNumbers)
test1 <- tm_map(DEPcorp, stripWhitespace)

#test to see how these worked
inspect(DEPcorp[[1]])

#Create a term document matrix and matrix
dtm <- TermDocumentMatrix(DEPcorp)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
fullcorp<-data.frame(m) %>%
  mutate_(word = rownames(m)) %>%
  filter(word %in% names(v[1:10])) %>%
  gather("document", "freq", -word)%>%
  mutate_(document = factor(document))

#plot term frequency
ggplot(fullcorp, aes(y=freq,x=word, fill = document)) +
  geom_bar(stat="identity",show.legend = F) +
  facet_wrap(~document, ncol = 2)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
# Topic Model Experiment

#create a bag of words matrix
dtm <- DocumentTermMatrix(DEPcorp)
as.matrix(dtm[1:4, 1:4])
#checking for outliers
char <- sapply(DEPcorp, function(x) nchar(content(x)))
hist(log10(char))

#inliers--can use this to filter out spurrious docs. This looks at num of characters in
text
#and filters out super small docs.
inlier <- function(x) {
  n <- nchar(content(x))
  n < 10^3 & n > 10
}
test1 <- tm_filter(test1, inlier)
dtm <- DocumentTermMatrix(test1)
dense_dtm <- removeSparseTerms(dtm, 0.999)
dense_dtm <- dense_dtm[rowSums(as.matrix(dense_dtm)) > 0, ]

as.matrix(dense_dtm[1:4, 1:4])

#Term correlations-the findAssocs function checks columns of the
#document-term matrix for correlations.
assoc <- findAssocs(dense_dtm, 'green', 0.2)
assoc
$green

#Latent Dirichelet Allocation
#library(topicmodels)
seed = 12345
fit = LDA(dense_dtm, k = 6, control = list(seed=seed))
terms(fit, 20)

#The topic “weights” can be assigned back to the documents for use in future
analyses.
topics <- posterior(fit, dense_dtm)$topics
topics <- as.data.frame(topics)
colnames(topics) <- c('green', 'monitor', 'infra', 'eval','storm','location')

head(topics)

findFreqTerms(dense_dtm, 100)

?tm
```

```
m <- as.matrix(topics)
v <- sort(rowSums(m),decreasing=TRUE)
topics<-data.frame(m) %>%
  mutate(topic = rownames(m)) %>%
  filter(topic %in% names(v[1:10])) %>%
  mutate(document = factor(document))

ggplot(topics, aes(y=freq,x=topic, fill = document)) +
  geom_bar(stat="identity",show.legend = T) +
  facet_wrap(~document, ncol = 2)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

?ggplot

#find a way to create a good looking table to display term requencies
#https://stackoverflow.com/questions/18101047/list-of-word-frequencies-using-r
#use ggplot2?


### Old experimental code I'm not ready to ditch yet...

#d <- data.frame(word = names(v),freq=v)

#head(v,10)
#head(d, 10)

#plot?
library(ggplot2)

ggplot(fullcorp, aes(y=freq,x=word, fill = document)) +
  geom_bar(stat="identity",show.legend = F) +
  facet_wrap(~document, ncol = 2)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


#Customize document preparation with your own functions. The function must be
wrapped in
#content_transformer if designed to accept and return strings rather than
PlainTextDocuments.
collapse <- function(x) {
  paste(x, collapse = '')
}
test1 <- tm_map(DEPcorp, content_transformer(collapse))

# not sure if I should use this after all, test1 <- tm_map(DEPcorp, Token_Tokenizer)
```

```
#Explore frequent terms and their associations
#You can have a look at the frequent terms in the term-document matrix as follow.
In the example
#below we want to find words that occur at least four times

findFreqTerms(dtm, lowfreq = 4)

#You can analyze the association between frequent terms (i.e., terms which
correlate) using findAssocs() function.
#The R code below identifies which words are associated with “green”
findAssocs(dtm, terms = "green", corlimit = 0.3)

#The frequency table of words
head(d, 10)

#Plot word frequencies. The frequency of the first 10 frequent words are plotted
#from http://www.sthda.com/english/wiki/text-mining-and-word-cloud-
fundamentals-in-r-5-simple-steps-you-should-know
barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,
        col ="lightblue", main ="Most frequent words",
        ylab = "Word frequencies")


inspect(test1[[1]])
inspect(DEPcorp[[1]])
content(DEPcorp[[2]])

test1<-pdf_text("test_DEP/DEP_2011_gi_annual_report_update.pdf")
test2<-
pdf_text("test_DEP/DEP_2012_green_infrastructure_pilot_monitoring_report.pdf")
cat(test1)
cat(test2)

??cluster


#we need to find the subset of docs with high term frequency of keyword interest
```

```
DEPcorp
DEPcorp[1]
inspect(DEPcorp[[1]])

##this didn't work https://dss.iq.harvard.edu/blog/extracting-content-pdf-files
library(pdftools)
plan2011 <- pdf_text("DEP_2011_gi_annual_report_update.pdf")
head(strsplit(plan2011[ [  1 ] ], "\n")[ [ 1 ] ])
```