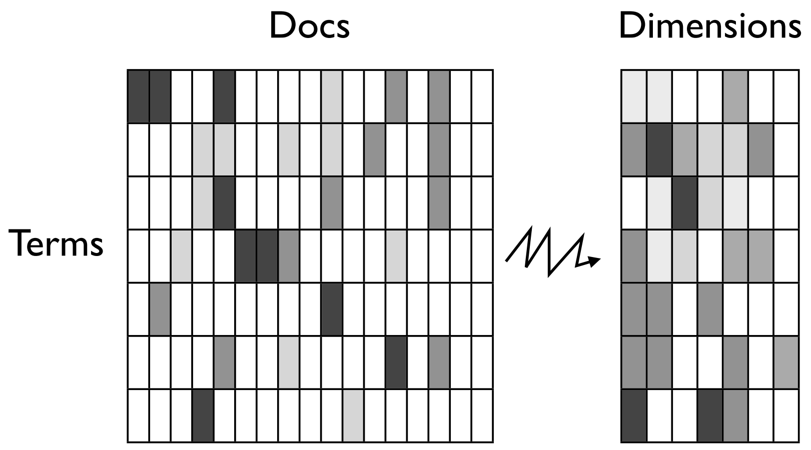
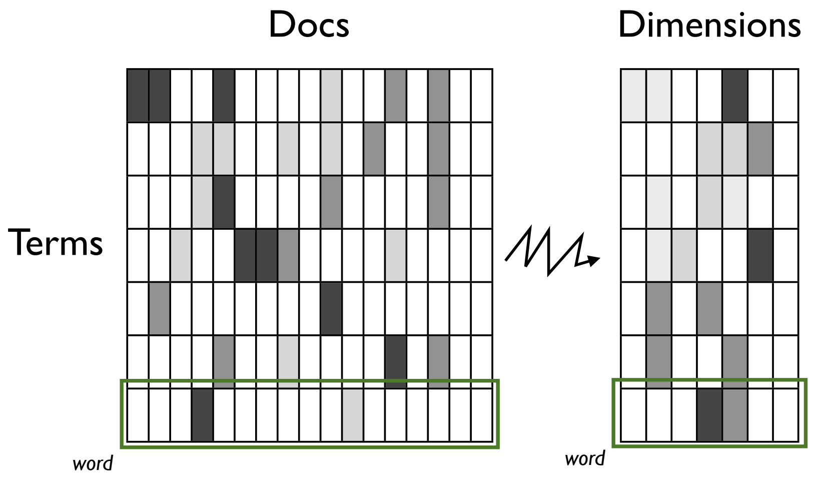
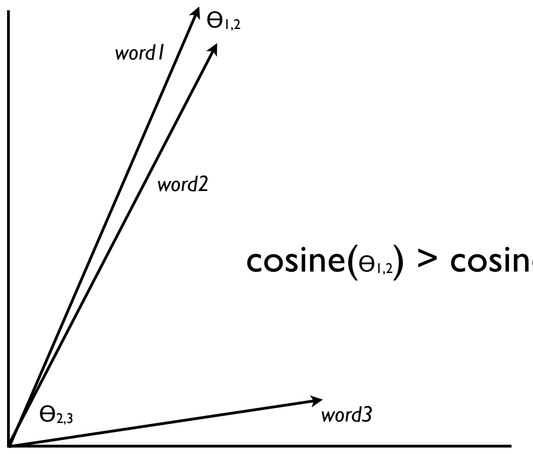
# Feat of Strength

- Follow these slides:
  - build the term X doc matrix
  - build the LSA "model"
  - test a few vectors using cosine
  - hand in: the cosine values in a ***toy 2-dimensional LSA model***...
  - Email to ucmcogsmmms12@gmail.com



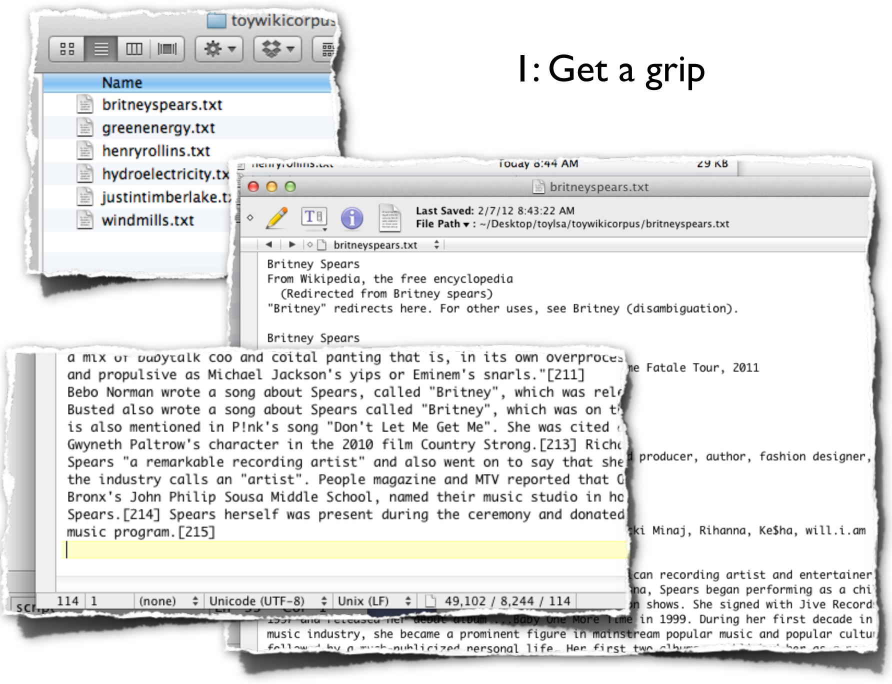




cosine($\theta_{1,2}$) > cosine($\theta_{2,3}$)

cosine = dot(word1,word2) / (norm(word1)*norm(word2))

## 1: Get a grip

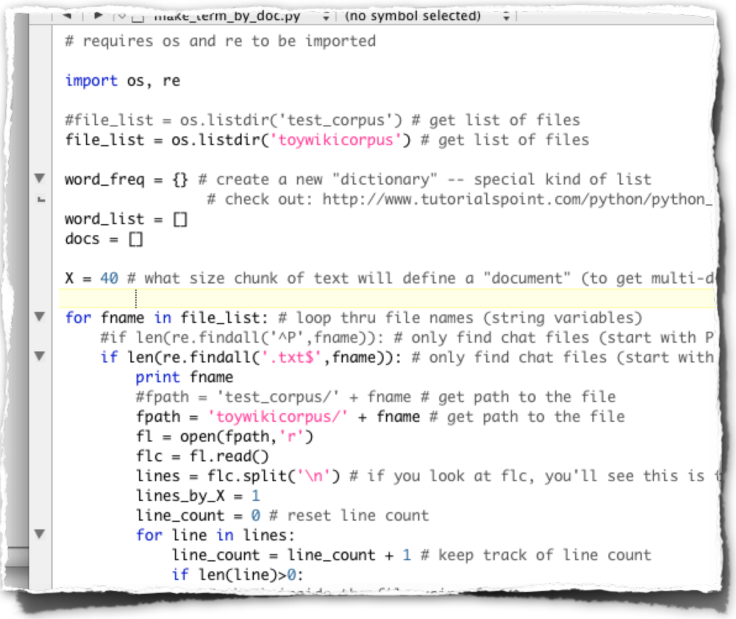


```
toywikicorpus
Name
britneyspears.txt
greenenergy.txt
henryrollins.txt
hydroelectricity.tx
justintimberlake.t
windmills.txt
```

```
britneyspears.txt
Last Saved: 2/7/12 8:43:22 AM
File Path ▾ : ~/Desktop/toylsa/toywikicorpus/britneyspears.txt

Britney Spears
From Wikipedia, the free encyclopedia
  (Redirected from Britney spears)
"Britney" redirects here. For other uses, see Britney (disambiguation).

Britney Spears
```

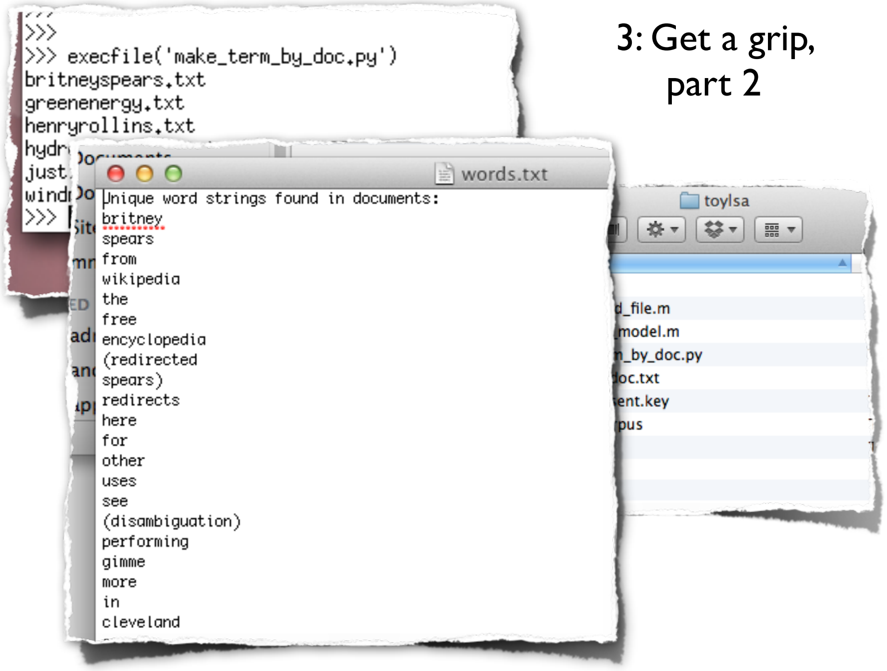
```
and propulsive as Michael Jackson's yips or Eminem's snarls."[211]
Bebo Norman wrote a song about Spears, called "Britney", which was rele
Busted also wrote a song about Spears called "Britney", which was on t
is also mentioned in P!nk's song "Don't Let Me Get Me". She was cited
Gwyneth Paltrow's character in the 2010 film Country Strong.[213] Riche
Spears "a remarkable recording artist" and also went on to say that she
the industry calls an "artist". People magazine and MTV reported that C
Bronx's John Philip Sousa Middle School, named their music studio in ho
Spears.[214] Spears herself was present during the ceremony and donated
music program.[215]

114 | 1    (none)    Unicode (UTF-8)    Unix (LF)    49,102 / 8,244 / 114
```

## 2: Process text with Python



```
make_term_by_doc.py    (no symbol selected)

# requires os and re to be imported

import os, re

#file_list = os.listdir('test_corpus') # get list of files
file_list = os.listdir('toywikicorpus') # get list of files

word_freq = {} # create a new "dictionary" -- special kind of list
                # check out: http://www.tutorialspoint.com/python/python_
word_list = []
docs = []

X = 40 # what size chunk of text will define a "document" (to get multi-d

for fname in file_list: # loop thru file names (string variables)
    #if len(re.findall('^P',fname)): # only find chat files (start with P
    if len(re.findall('.txt$',fname)): # only find chat files (start with
        print fname
        #fpath = 'test_corpus/' + fname # get path to the file
        fpath = 'toywikicorpus/' + fname # get path to the file
        fl = open(fpath,'r')
        flc = fl.read()
        lines = flc.split('\n') # if you look at flc, you'll see this is
        lines_by_X = 1
        line_count = 0 # reset line count
        for line in lines:
            line_count = line_count + 1 # keep track of line count
            if len(line)>0:
```

## 3: Get a grip, part 2

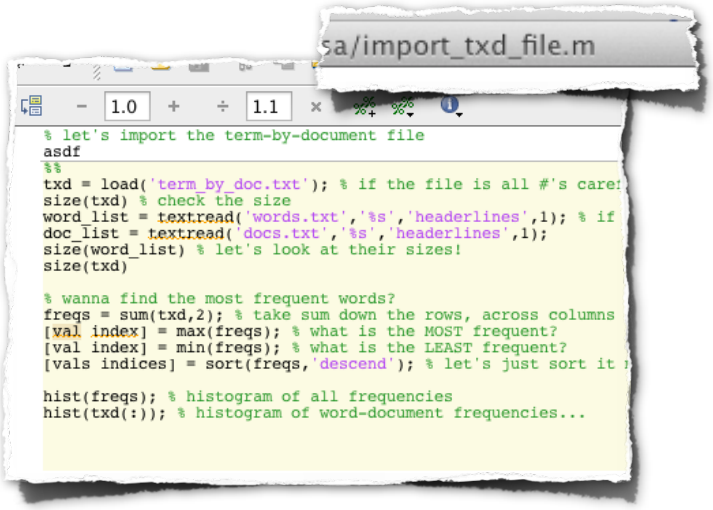


```
>>>
>>> execfile('make_term_by_doc.py')
britneyspears.txt
greenenergy.txt
henryrollins.txt
hydr
just
windi
>>>
```

```
words.txt
Unique word strings found in documents:
britney
spears
from
wikipedia
the
free
encyclopedia
(redirected
spears)
redirects
here
for
other
uses
see
(disambiguation)
performing
gimme
more
in
cleveland
```

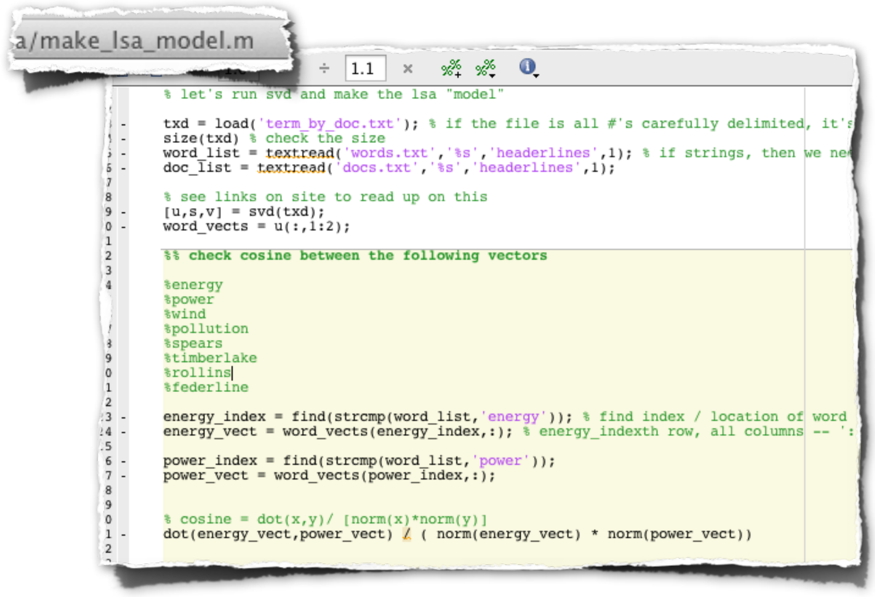
```
toylsa
d_file.m
model.m
n_by_doc.py
doc.txt
ent.key
pus
```

## 4: Import into MATLAB

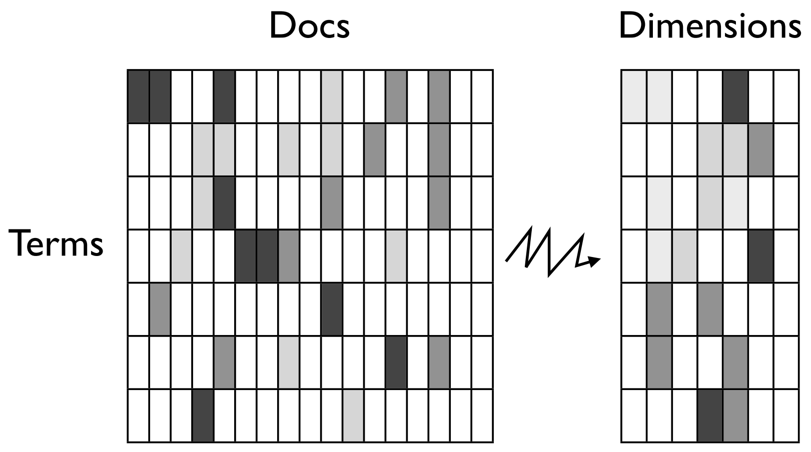


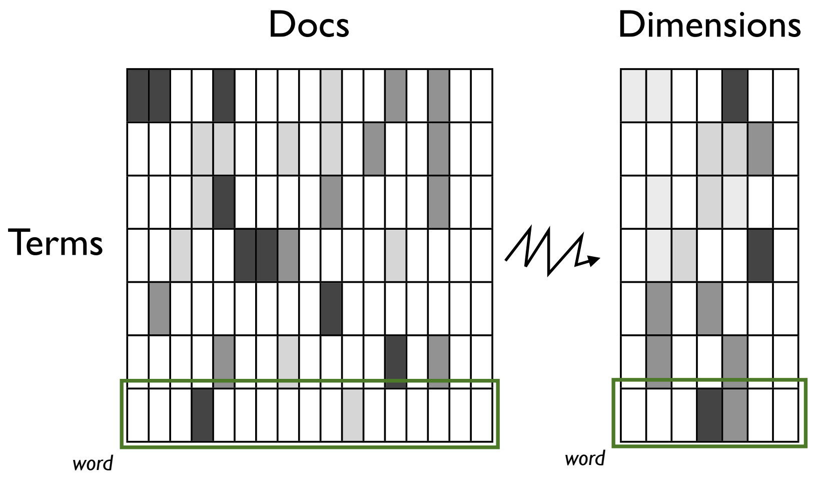
```
sa/import_txd_file.m

% let's import the term-by-document file
asdf
%%
txd = load('term_by_doc.txt'); % if the file is all #'s care
size(txd) % check the size
word_list = textread('words.txt','%s','headerlines',1); % if
doc_list = textread('docs.txt','%s','headerlines',1);
size(word_list) % let's look at their sizes!
size(txd)

% wanna find the most frequent words?
freqs = sum(txd,2); % take sum down the rows, across columns
[val index] = max(freqs); % what is the MOST frequent?
[val index] = min(freqs); % what is the LEAST frequent?
[vals indices] = sort(freqs,'descend'); % let's just sort it

hist(freqs); % histogram of all frequencies
hist(txd(:)); % histogram of word-document frequencies...
```

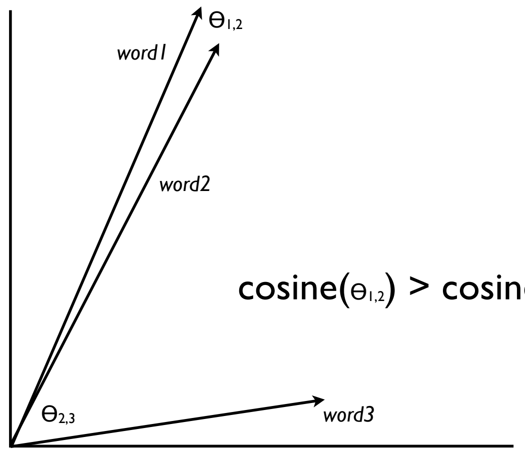
## 5: Use SVD for LSA, then play with vectors!



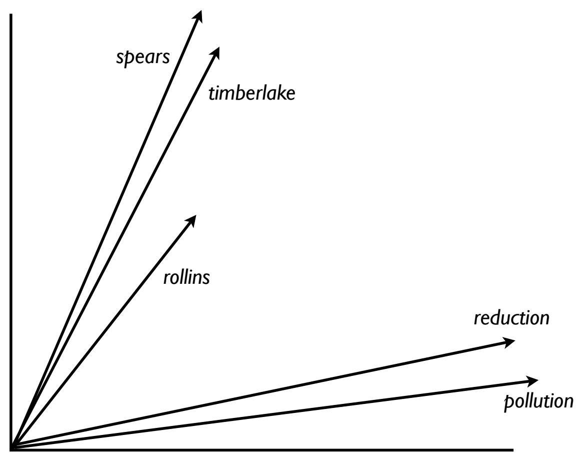
```
% let's run svd and make the lsa "model"

txd = load('term_by_doc.txt'); % if the file is all #'s carefully delimited, it's
size(txd) % check the size
word_list = textread('words.txt','%s','headerlines',1); % if strings, then we ne
doc_list = textread('docs.txt','%s','headerlines',1);

% see links on site to read up on this
[u,s,v] = svd(txd);
word_vects = u(:,1:2);

%% check cosine between the following vectors

%energy
%power
%wind
%pollution
%spears
%timberlake
%rollins
%federline

energy_index = find(strcmp(word_list,'energy')); % find index / location of word
energy_vect = word_vects(energy_index,:); % energy_indexth row, all columns -- ':

power_index = find(strcmp(word_list,'power'));
power_vect = word_vects(power_index,:);


% cosine = dot(x,y)/ [norm(x)*norm(y)]
dot(energy_vect,power_vect) / ( norm(energy_vect) * norm(power_vect))
```







$\text{cosine}(\Theta_{1,2}) > \text{cosine}(\Theta_{2,3})$


spears
timberlake
rollins
reduction
pollution