Assignment 8

Cs432 Web Science Spring 2017

Breon Day

April 14,2017

Question 1:

Part 1:
Create a blog-term matrix.Start by grabbing 100 blogs;include:
http://f-measure.blogspot.com/ ,http://ws-dl.blogspot.com/ and grab 98 more as per the method shown in class.

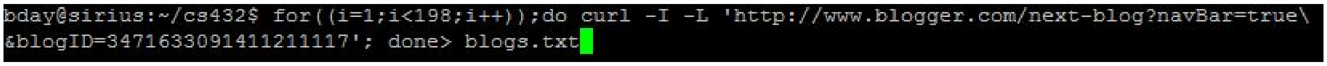
Part 2:
Use the blog title as the identifier for each blog (and row of the matrix). Use the terms from every item/title (RSS) or entry/title (Atom) for the columns of the matrix. The values are the frequency of occurrence.

Part 3:
Limit the number of terms to the most "popular" (i.e., frequent) 1000 terms, this is *after* the criteria on p. 32 (slide 7) has been satisfied.

Answer 1:

Part1

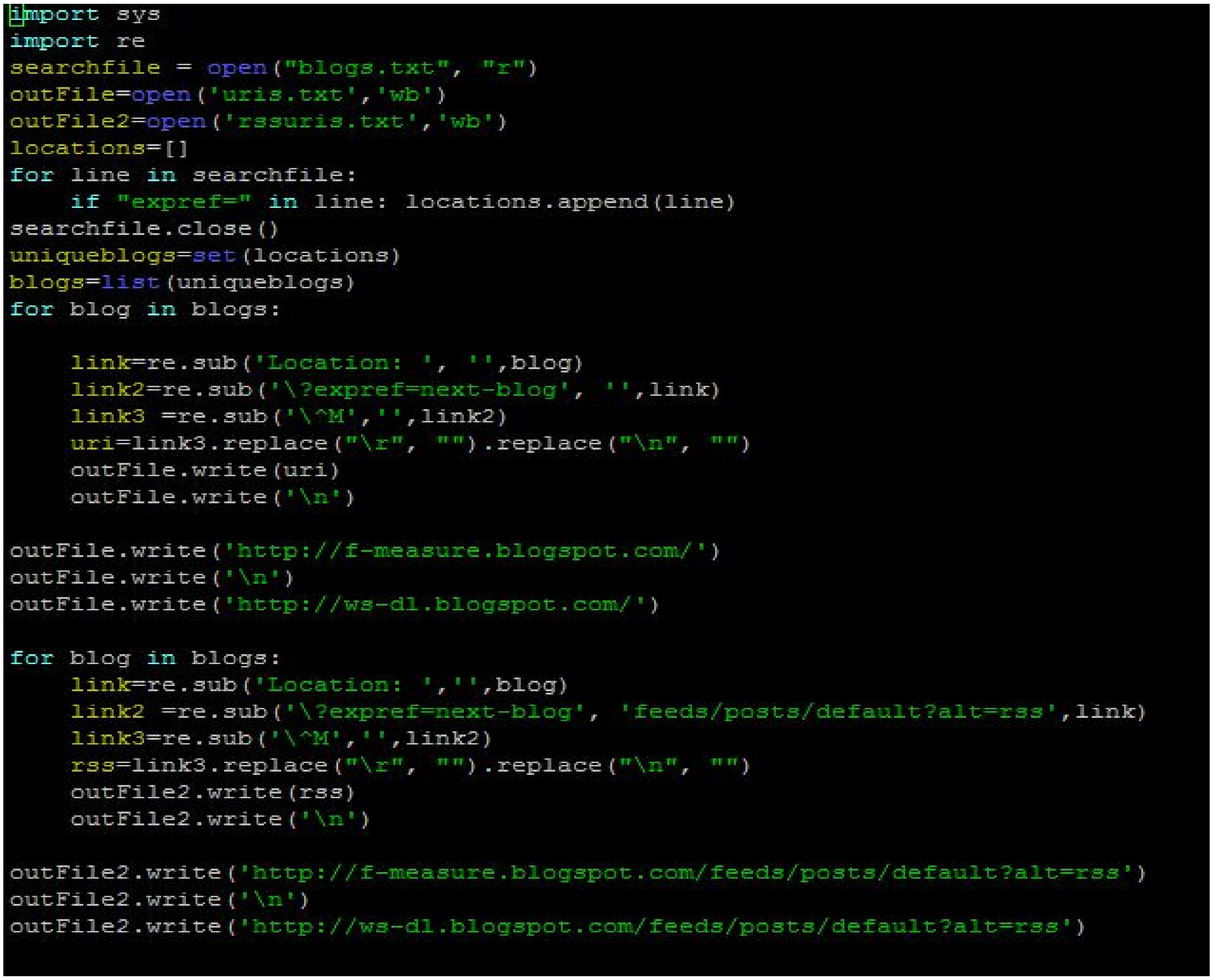
I expanded upon the command line method show in class, curl-I-L'http://www.blogger.com/next-blog?navBar=true\&blogID=3471633091411211117,wrapping it in a for loop and sending the output to the file blogs.txt.



```
bday@sirius:~/cs432$ for((i=1;i<198;i++));do curl -I -L 'http://www.blogger.com/next-blog?navBar=true\
&blogID=3471633091411211117'; done> blogs.txt
```

To account for bad links i increased the initial 98 planned to 198.

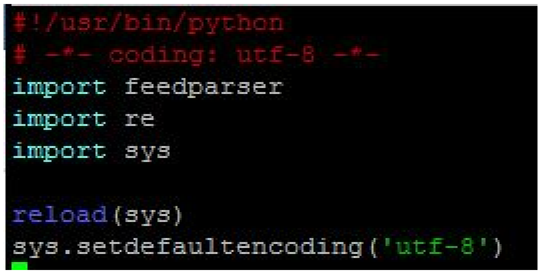
The file blogs.py processes the data into uris and rss links through the use of regular expressions and removes duplicates by converting the original list of links into a set of unique links then back into a list creating two files the final product will be the two files uris.txt and rssuris.txt containing the processed links



```
import sys
import re
searchfile = open("blogs.txt", "r")
outFile=open('uris.txt','wb')
outFile2=open('rssuris.txt','wb')
locations=[]
for line in searchfile:
    if "expref=" in line: locations.append(line)
searchfile.close()
uniqueblogs=set(locations)
blogs=list(uniqueblogs)
for blog in blogs:

    link=re.sub('Location: ', '',blog)
    link2=re.sub('\?expref=next-blog', '',link)
    link3 =re.sub('\^M','',link2)
    uri=link3.replace("\r", "").replace("\n", "")
    outFile.write(uri)
    outFile.write('\n')

outFile.write('http://f-measure.blogspot.com/')
outFile.write('\n')
outFile.write('http://ws-dl.blogspot.com/')

for blog in blogs:
    link=re.sub('Location: ','',blog)
    link2 =re.sub('\?expref=next-blog', 'feeds/posts/default?alt=rss',link)
    link3=re.sub('\^M','',link2)
    rss=link3.replace("\r", "").replace("\n", "")
    outFile2.write(rss)
    outFile2.write('\n')

outFile2.write('http://f-measure.blogspot.com/feeds/posts/default?alt=rss')
outFile2.write('\n')
outFile2.write('http://ws-dl.blogspot.com/feeds/posts/default?alt=rss')
```

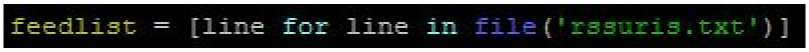
Part 2 and 3:

The creation of the blog matrix was handled by the class generatefeedvector.py from the PCI book chapter3 github, originally i had errors until another student pointed out the default encoding was the culprit,importing and reloading sys then changing default encoding to utf-8 was all that was required.



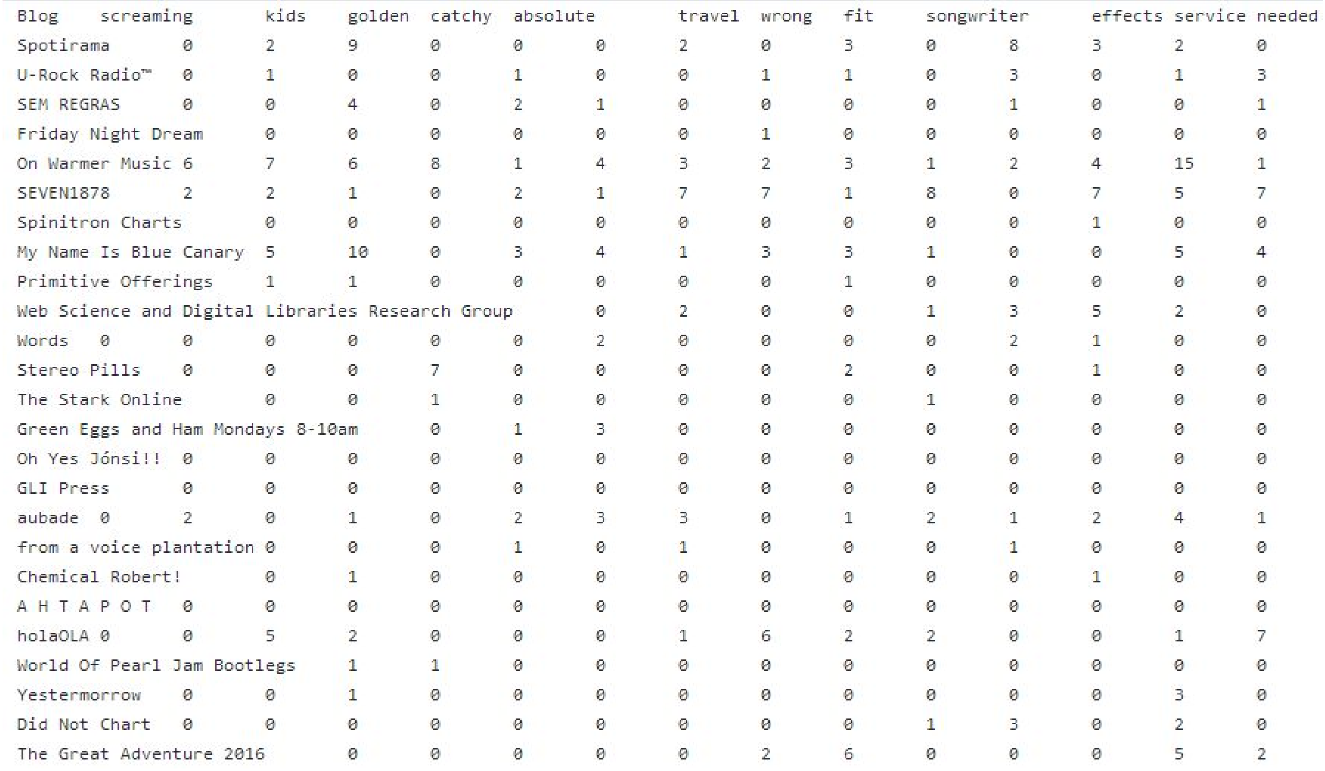
```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import feedparser
import re
import sys

reload(sys)
sys.setdefaultencoding('utf-8')
```

Substituting the original file with my own rssuris.txt allowed me to generate the blog matrixes based on the rss uris i had acquired.



```
feedlist = [line for line in file('rssuris.txt')]
```

Execution of the line python generatefeedvectors.py>blogtitles.txt
produced a portion of the following blog matrix as well as stored their titles
full matrix: https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/Src/Question1/blogdata1.txt
Partial matrix below



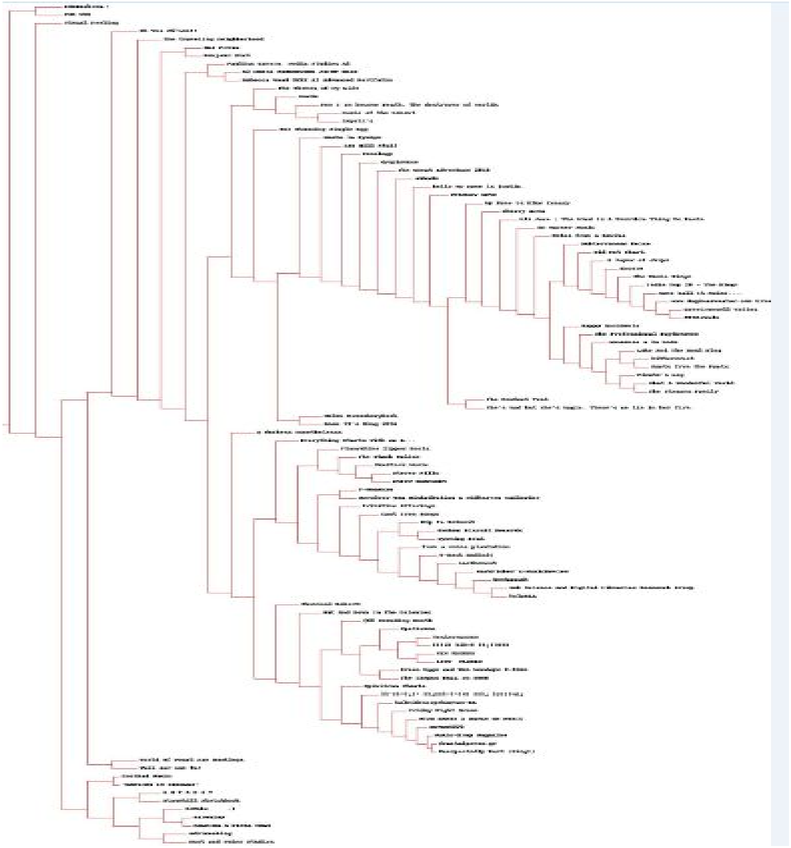
```
Blog    screaming       kids    golden  catchy  absolute        travel  wrong   fit     songwriter      effects service needed
Spotirama       0       2       9       0       0       0       2       0       3       0       8       3       2       0
U-Rock Radio™   0       1       0       0       1       0       0       1       1       0       3       0       1       3
SEM REGRAS      0       0       4       0       2       1       0       0       0       0       1       0       0       1
Friday Night Dream      0       0       0       0       0       0       1       0       0       0       0       0       0
On Warmer Music 6       7       6       8       1       4       3       2       3       1       2       4       15      1
SEVEN1878       2       2       1       0       2       1       7       7       1       8       0       7       5       7
Spinitron Charts        0       0       0       0       0       0       0       0       0       0       1       0       0
My Name Is Blue Canary  5       10      0       3       4       1       3       3       1       0       0       5       4
Primitive Offerings     1       1       0       0       0       0       0       1       0       0       0       0       0
Web Science and Digital Libraries Research Group        0       2       0       0       1       3       5       2       0
Words   0       0       0       0       0       0       2       0       0       0       0       2       1       0       0
Stereo Pills    0       0       0       7       0       0       0       0       2       0       0       1       0       0
The Stark Online        0       0       1       0       0       0       0       0       1       0       0       0       0
Green Eggs and Ham Mondays 8-10am       0       1       3       0       0       0       0       0       0       0       0
Oh Yes Jónsi!!  0       0       0       0       0       0       0       0       0       0       0       0       0       0
GLI Press       0       0       0       0       0       0       0       0       0       0       0       0       0       0
aubade  0       2       0       1       0       2       3       3       0       1       2       1       2       4       1
from a voice plantation 0       0       0       1       0       1       0       0       0       1       0       0       0
Chemical Robert!        0       1       0       0       0       0       0       0       0       0       1       0       0
A H T A P O T   0       0       0       0       0       0       0       0       0       0       0       0       0       0
holaOLA 0       0       5       2       0       0       0       1       6       2       2       0       0       1       7
World Of Pearl Jam Bootlegs     1       1       0       0       0       0       0       0       0       0       0       0
Yestermorrow    0       0       1       0       0       0       0       0       0       0       0       0       3       0
Did Not Chart   0       0       0       0       0       0       0       0       0       1       3       0       2       0
The Great Adventure 2016        0       0       0       0       0       2       6       0       0       0       5       2
```

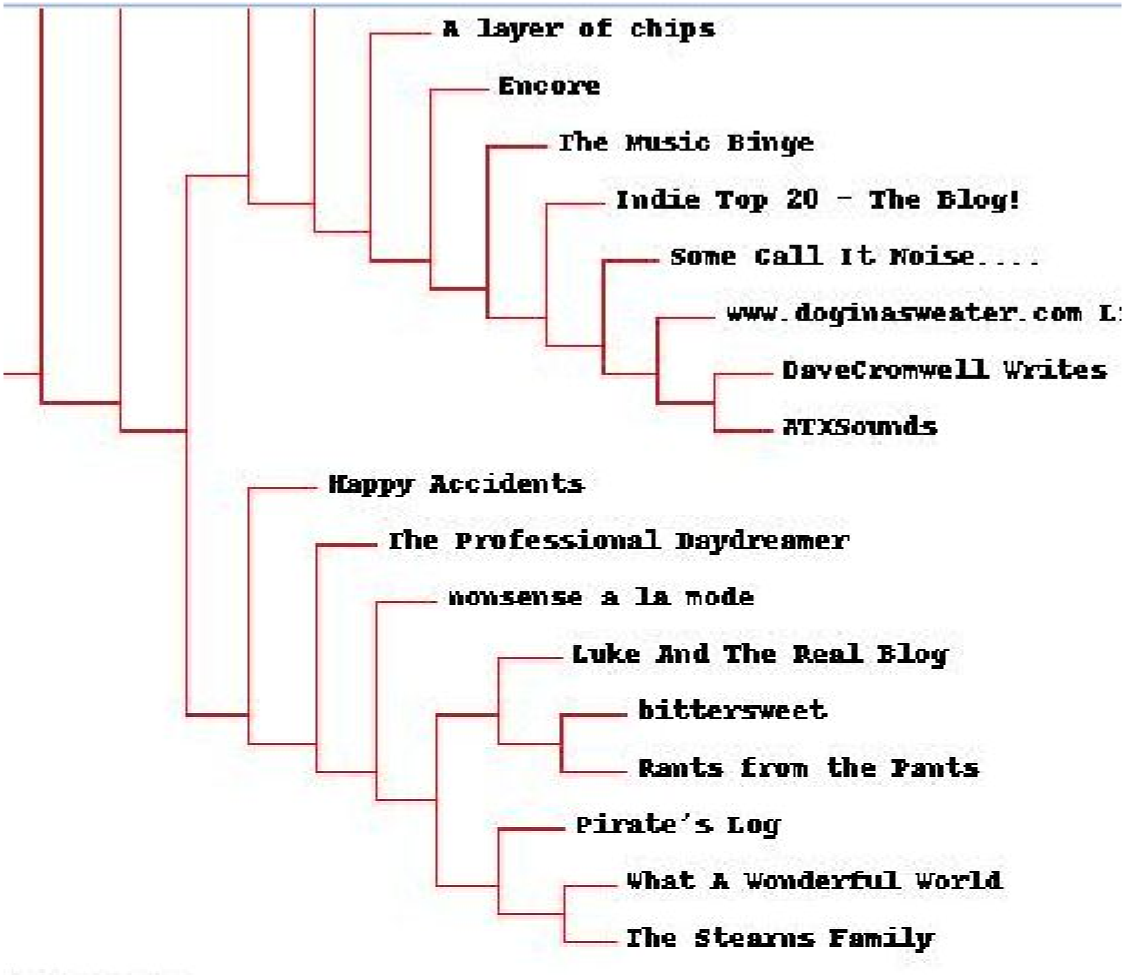
Question 2:

Create an ASCII and JPEG dendrogram that clusters (i.e., HAC) the most similar blogs (see slides 12 & 13).

Answer 2:

First i obtained the prerequisite cluster file from the github repository of the programming collective intelligence chapter 3. Next using the code provided in slides 12 and 13 as well as the previous blog matrix i was able to generate the ascii dendrogram,excluded from this report ,and jpeg dendrogram shown below



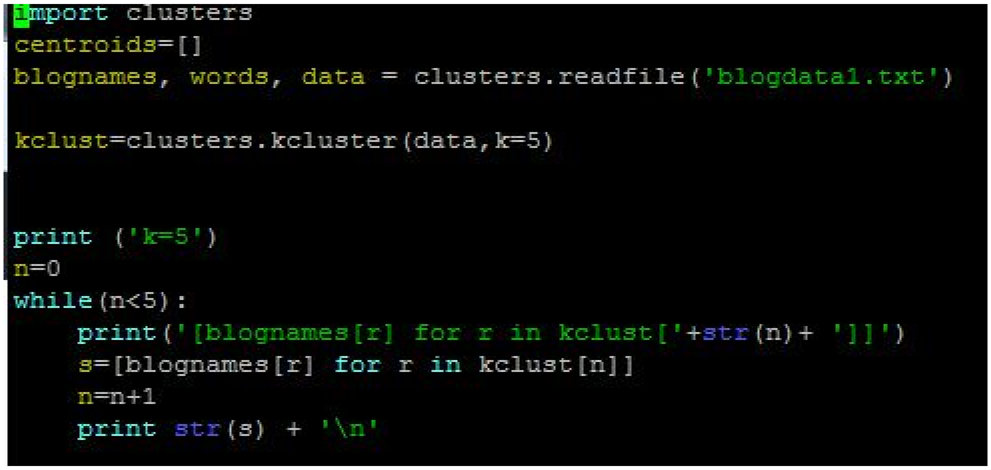

A layer of chips
Encore
The Music Binge
Indie Top 20 - The Blog!
Some Call It Noise....
www.doginasweater.com L
DaveCromwell Writes
ATXSounds
Happy Accidents
The Professional Daydreamer
nonsense a la mode
Luke And The Real Blog
bittersweet
Rants from the Pants
Pirate's Log
What A Wonderful World
The Stearns Family

Question 3:

Cluster the blogs using K-Means, using k=5,10,20. (see slide 18). Print the values in each centroid, for each value of k. How many interations were required for each value of k?

Answer 3:

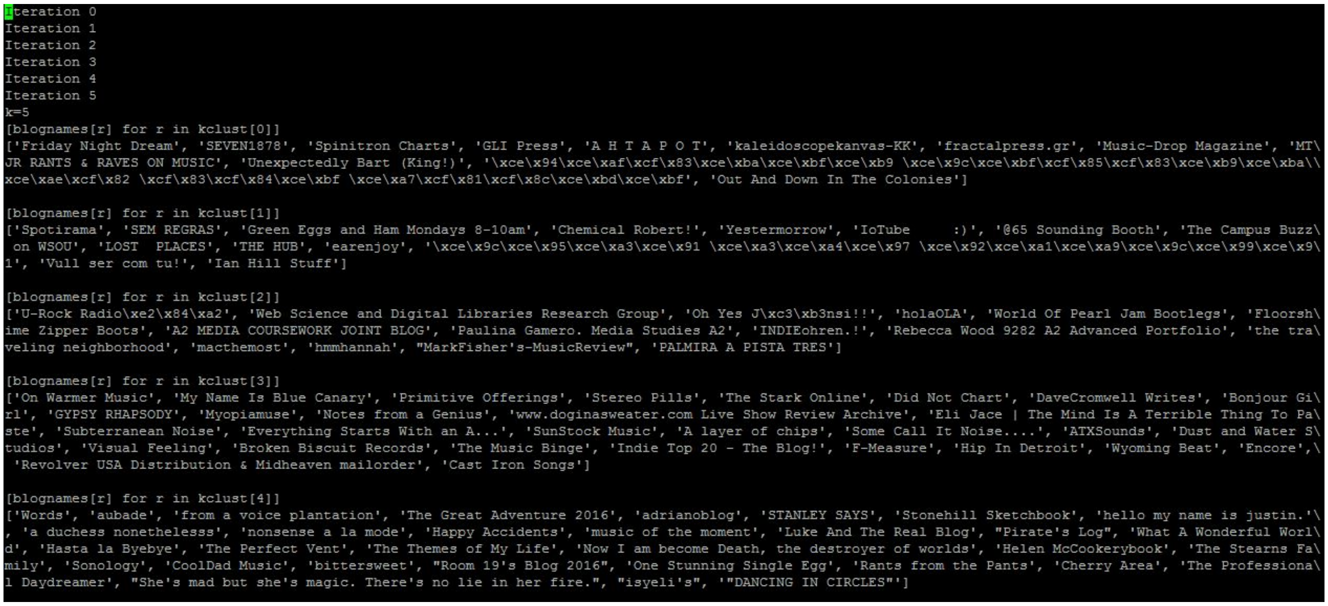
By simply modifying the given code on slide 18 and putting it into a while loop



```
import clusters
centroids=[]
blognames, words, data = clusters.readfile('blogdata1.txt')

kclust=clusters.kcluster(data,k=5)


print ('k=5')
n=0
while(n<5):
    print('[blognames[r] for r in kclust['+str(n)+ ']]')
    s=[blognames[r] for r in kclust[n]]
    n=n+1
    print str(s) + '\n'
```

I was able to achieve the desired output

Full view:



```
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
k=5
[blognames[r] for r in kclust[0]]
['Friday Night Dream', 'SEVEN1878', 'Spinitron Charts', 'GLI Press', 'A H T A P O T', 'kaleidoscopekanvas-KK', 'fractalpress.gr', 'Music-Drop Magazine', 'MT\
JR RANTS & RAVES ON MUSIC', 'Unexpectedly Bart (King!)', '\xce\x94\xce\xaf\xcf\x83\xce\xba\xce\xbf\xce\xb9 \xce\x9c\xce\xbf\xcf\x85\xcf\x83\xce\xb9\xce\xba\\
xce\xae\xcf\x82 \xcf\x83\xcf\x84\xce\xbf \xce\xa7\xcf\x81\xcf\x8c\xce\xbd\xce\xbf', 'Out And Down In The Colonies']

[blognames[r] for r in kclust[1]]
['Spotirama', 'SEM REGRAS', 'Green Eggs and Ham Mondays 8-10am', 'Chemical Robert!', 'Yestermorrow', 'IoTube     :)', '@65 Sounding Booth', 'The Campus Buzz\
 on WSOU', 'LOST  PLACES', 'THE HUB', 'earenjoy', '\xce\x9c\xce\x95\xce\xa3\xce\x91 \xce\xa3\xce\xa4\xce\x97 \xce\x92\xce\xa1\xce\xa9\xce\x9c\xce\x99\xce\x9\
1', 'Vull ser com tu!', 'Ian Hill Stuff']

[blognames[r] for r in kclust[2]]
['U-Rock Radio\xe2\x84\xa2', 'Web Science and Digital Libraries Research Group', 'Oh Yes J\xc3\xb3nsi!!', 'holaOLA', 'World Of Pearl Jam Bootlegs', 'Floorsh\
ime Zipper Boots', 'A2 MEDIA COURSEWORK JOINT BLOG', 'Paulina Gamero. Media Studies A2', 'INDIEohren.!', 'Rebecca Wood 9282 A2 Advanced Portfolio', 'the tra\
veling neighborhood', 'macthemost', 'hmmhannah', "MarkFisher's-MusicReview", 'PALMIRA A PISTA TRES']

[blognames[r] for r in kclust[3]]
['On Warmer Music', 'My Name Is Blue Canary', 'Primitive Offerings', 'Stereo Pills', 'The Stark Online', 'Did Not Chart', 'DaveCromwell Writes', 'Bonjour Gi\
rl', 'GYPSY RHAPSODY', 'Myopiamuse', 'Notes from a Genius', 'www.doginasweater.com Live Show Review Archive', 'Eli Jace | The Mind Is A Terrible Thing To Pa\
ste', 'Subterranean Noise', 'Everything Starts With an A...', 'SunStock Music', 'A layer of chips', 'Some Call It Noise....', 'ATXSounds', 'Dust and Water S\
tudios', 'Visual Feeling', 'Broken Biscuit Records', 'The Music Binge', 'Indie Top 20 - The Blog!', 'F-Measure', 'Hip In Detroit', 'Wyoming Beat', 'Encore',\
 'Revolver USA Distribution & Midheaven mailorder', 'Cast Iron Songs']

[blognames[r] for r in kclust[4]]
['Words', 'aubade', 'from a voice plantation', 'The Great Adventure 2016', 'adrianoblog', 'STANLEY SAYS', 'Stonehill Sketchbook', 'hello my name is justin.'\
, 'a duchess nonethelesss', 'nonsense a la mode', 'Happy Accidents', 'music of the moment', 'Luke And The Real Blog', "Pirate's Log", 'What A Wonderful Worl\
d', 'Hasta la Byebye', 'The Perfect Vent', 'The Themes of My Life', 'Now I am become Death, the destroyer of worlds', 'Helen McCookerybook', 'The Stearns Fa\
mily', 'Sonology', 'CoolDad Music', 'bittersweet', "Room 19's Blog 2016", 'One Stunning Single Egg', 'Rants from the Pants', 'Cherry Area', 'The Professiona\
l Daydreamer', "She's mad but she's magic. There's no lie in her fire.", "isyeli's", '"DANCING IN CIRCLES"']
```

Total iterations were 6,6 and 4 respectively

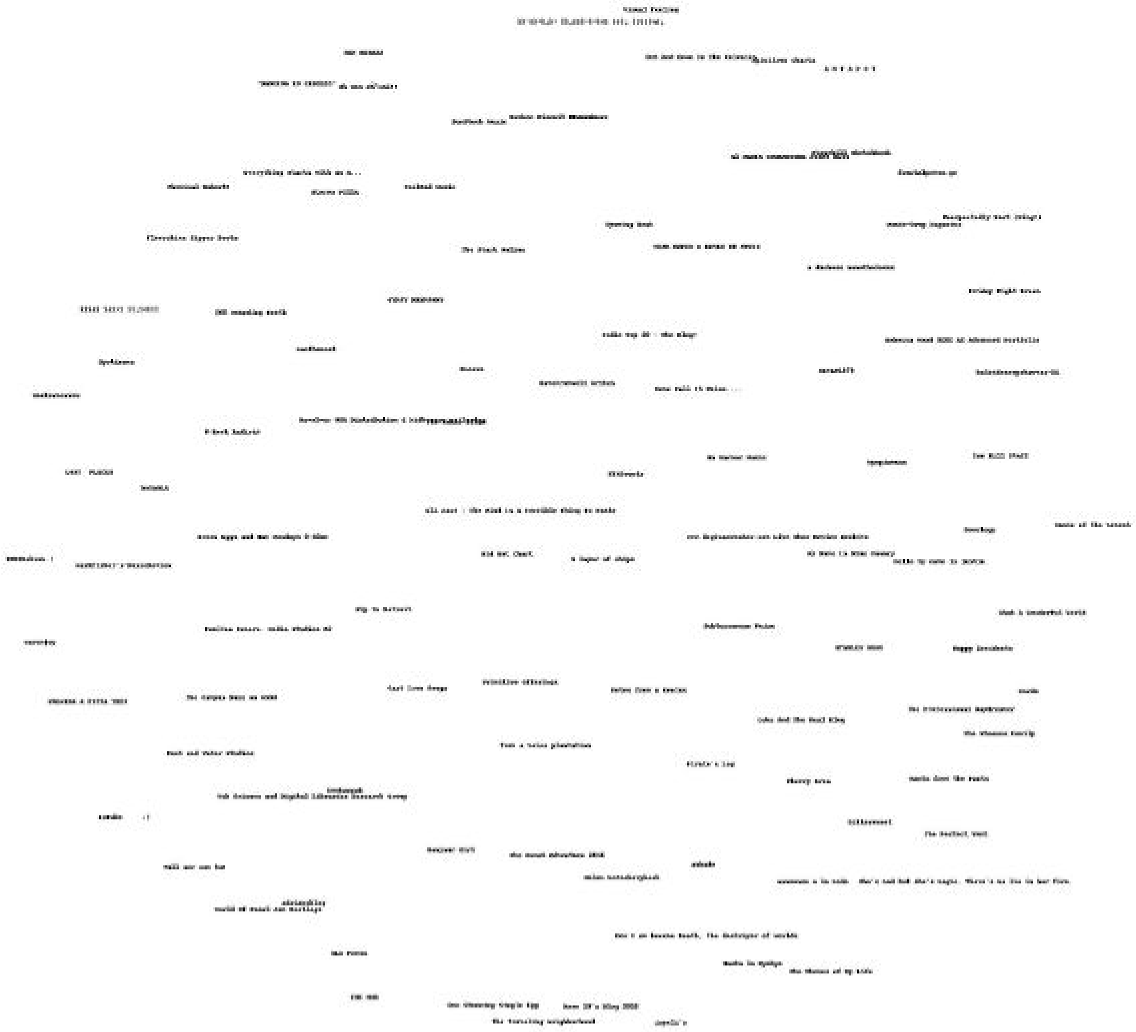
Question 4:

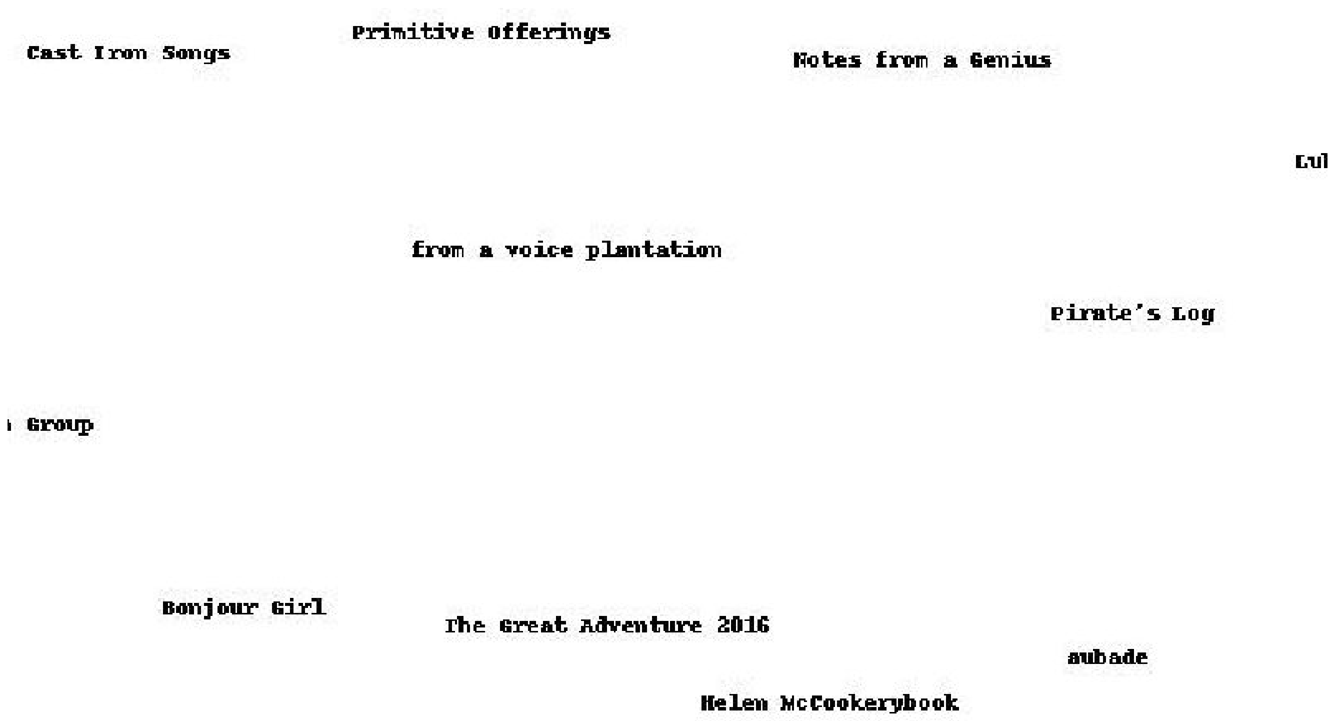
Use MDS to create a JPEG of the blogs similar to slide 29 of the week 12 lecture. How many iterations were required?

Answer 4:

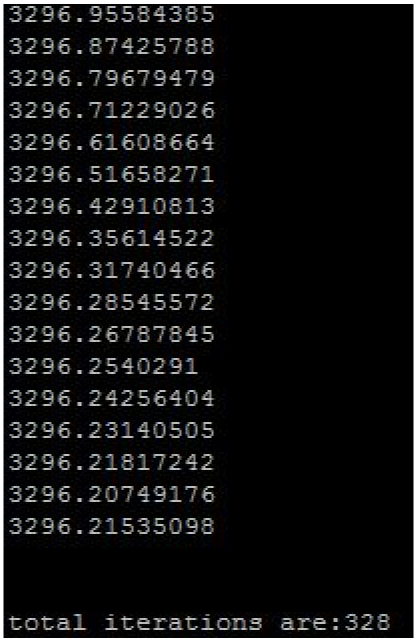
Modified slide 28 code to produce the jpeg and fed the data into a file mds.py>mds.txt that had the iteration count

jpeg:





Iteration count:328*



```
3296.95584385
3296.87425788
3296.79679479
3296.71229026
3296.61608664
3296.51658271
3296.42910813
3296.35614522
3296.31740466
3296.28545572
3296.26787845
3296.2540291
3296.24256404
3296.23140505
3296.21817242
3296.20749176
3296.21535098


total iterations are:328
```

* note this count will not represent the one in the github repo as the screen capture from the file was not coming out the way i wanted