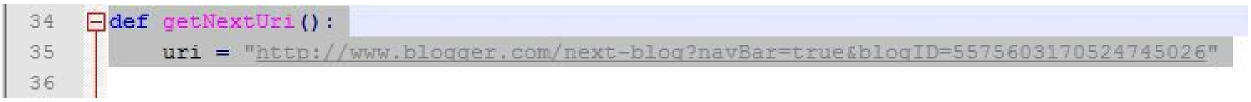
Alexander Wilson

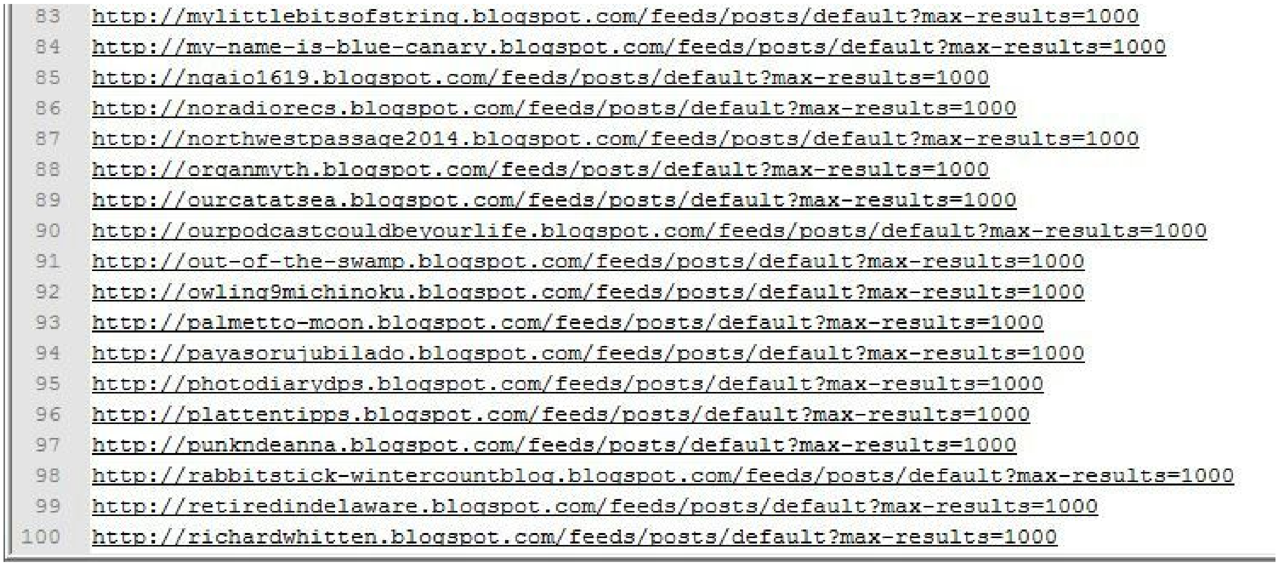
1) Create a blog-term matrix.

I began this assignment by running fetchURL.py in putty, using the linux enviroment. I have found over the course of this class that using the school's environment is easier to run because of python libraries. I ran fetchURI three times each outputting to a different text file. The reason I did it three times was each time I ran a different blog address to get further away from the source. (below is where I would change the blog location by changing the ID)



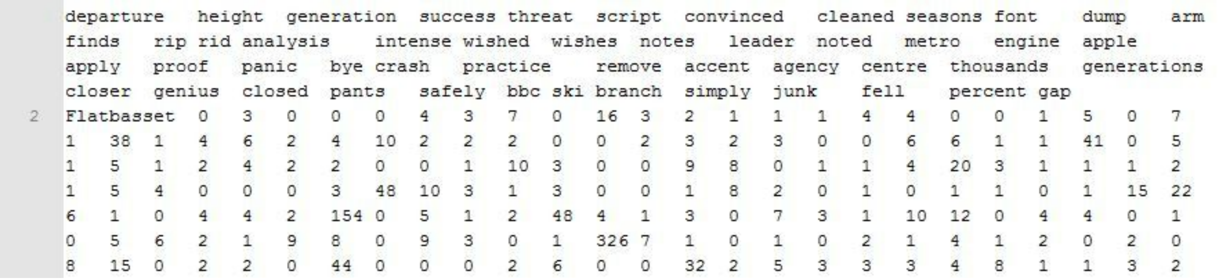
```
def getNextUri():
    uri = "http://www.blogger.com/next-blog?navBar=true&blogID=5575603170524745026"

```

I then combined the URIs into one file. (see feedlist.txt)
I then used the linux command -u *filename* to sort the list and get rid of duplicate URIs.



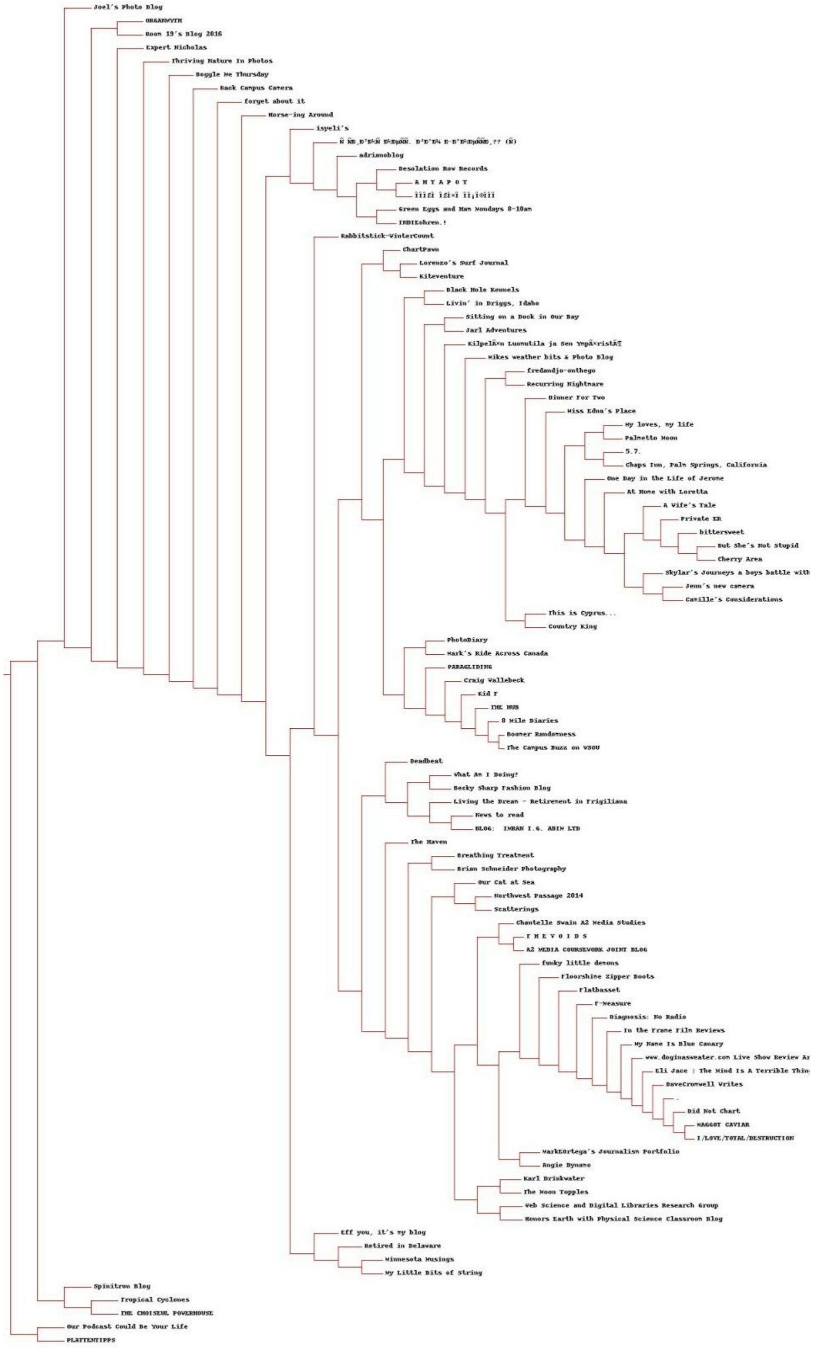
```
http://mylittlebitsofstring.blogspot.com/feeds/posts/default?max-results=1000
http://my-name-is-blue-canary.blogspot.com/feeds/posts/default?max-results=1000
http://ngaio1619.blogspot.com/feeds/posts/default?max-results=1000
http://noradiorecs.blogspot.com/feeds/posts/default?max-results=1000
http://northwestpassage2014.blogspot.com/feeds/posts/default?max-results=1000
http://organmyth.blogspot.com/feeds/posts/default?max-results=1000
http://ourcatatsea.blogspot.com/feeds/posts/default?max-results=1000
http://ourpodcastcouldbeyourlife.blogspot.com/feeds/posts/default?max-results=1000
http://out-of-the-swamp.blogspot.com/feeds/posts/default?max-results=1000
http://owling9michinoku.blogspot.com/feeds/posts/default?max-results=1000
http://palmetto-moon.blogspot.com/feeds/posts/default?max-results=1000
http://payasorujubilado.blogspot.com/feeds/posts/default?max-results=1000
http://photodiarydps.blogspot.com/feeds/posts/default?max-results=1000
http://plattentipps.blogspot.com/feeds/posts/default?max-results=1000
http://punkndeanna.blogspot.com/feeds/posts/default?max-results=1000
http://rabbitstick-wintercountblog.blogspot.com/feeds/posts/default?max-results=1000
http://retiredindelaware.blogspot.com/feeds/posts/default?max-results=1000
http://richardwhitten.blogspot.com/feeds/posts/default?max-results=1000
```

I then ended up with 143 URIs. I tried completing the whole assignment with 143 blogs locations but found it overloaded the generatefeedvector.py program. I chopped my list down to 100 and that cleared all my issues. (I relocated your two blogs to the top of the URL list to make them easier to find.)I was then able to generate all the proper data with generatefeedvector.py which outputted to blogdata.txt.



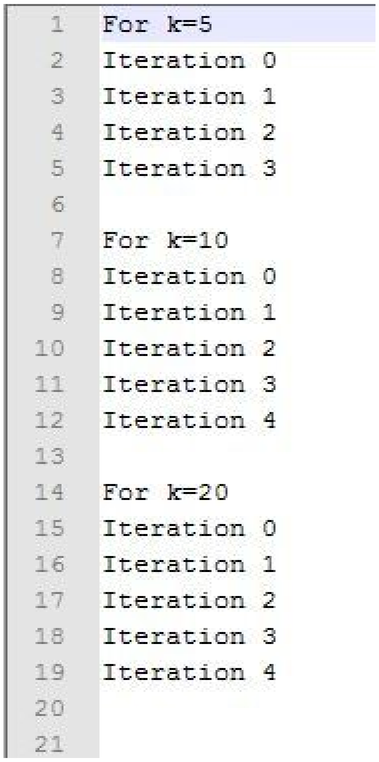
```
departure   height  generation  success threat  script  convinced   cleaned seasons font    dump    arm
finds   rip rid analysis    intense wished  wishes  notes   leader  noted   metro   engine  apple
apply   proof   panic   bye crash   practice    remove  accent  agency  centre  thousands   generations
closer  genius  closed  pants   safely  bbc ski branch  simply  junk    fell    percent gap
Flatbasset  0   3   0   0   0   4   3   7   0   16  3   2   1   1   1   4   4   0   0   1   5   0   7
1   38  1   4   6   2   4   10  2   2   2   0   0   2   3   2   3   0   0   6   6   1   1   41  0   5
1   5   1   2   4   2   2   0   0   1   10  3   0   0   9   8   0   1   1   4   20  3   1   1   1   2
1   5   4   0   0   0   3   48  10  3   1   3   0   0   1   8   2   0   1   0   1   1   0   1   15  22
6   1   0   4   4   2   154 0   5   1   2   48  4   1   3   0   7   3   1   10  12  0   4   4   0   1
0   5   6   2   1   9   8   0   9   3   0   1   326 7   1   0   1   0   2   1   4   1   2   0   2   0
8   15  0   2   2   0   44  0   0   0   2   6   0   0   32  2   5   3   3   3   4   8   1   1   3   2
```

2) Create an ASCII and JPEG dendrogram that clusters.
I used the script makeDendogram.py which I ran with my blogdata.txt from q1. It took me a while to figure out how to get clusters to run properly. I found that putting the clusters.py inside all my folder that call it fixed the issues. I found this question to be difficult my first time around because my text file was formatted incorrectly due to 143 different blogs. After cutting down the list I was able to produce a working graph.



3) Cluster the blogs using K-Means, using k=5,10,20.

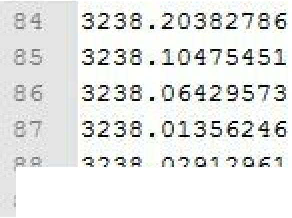
I made a separate folder for each question however in hindsight it would have been much easier to keep it all the files and data in the same folder, mainly because of clusters.py. I preformed question three by running the script iterator.py. It produced k=5 got 4 iterations, K=10 we get 5 iterations, and K=20 I got 5 iterations.



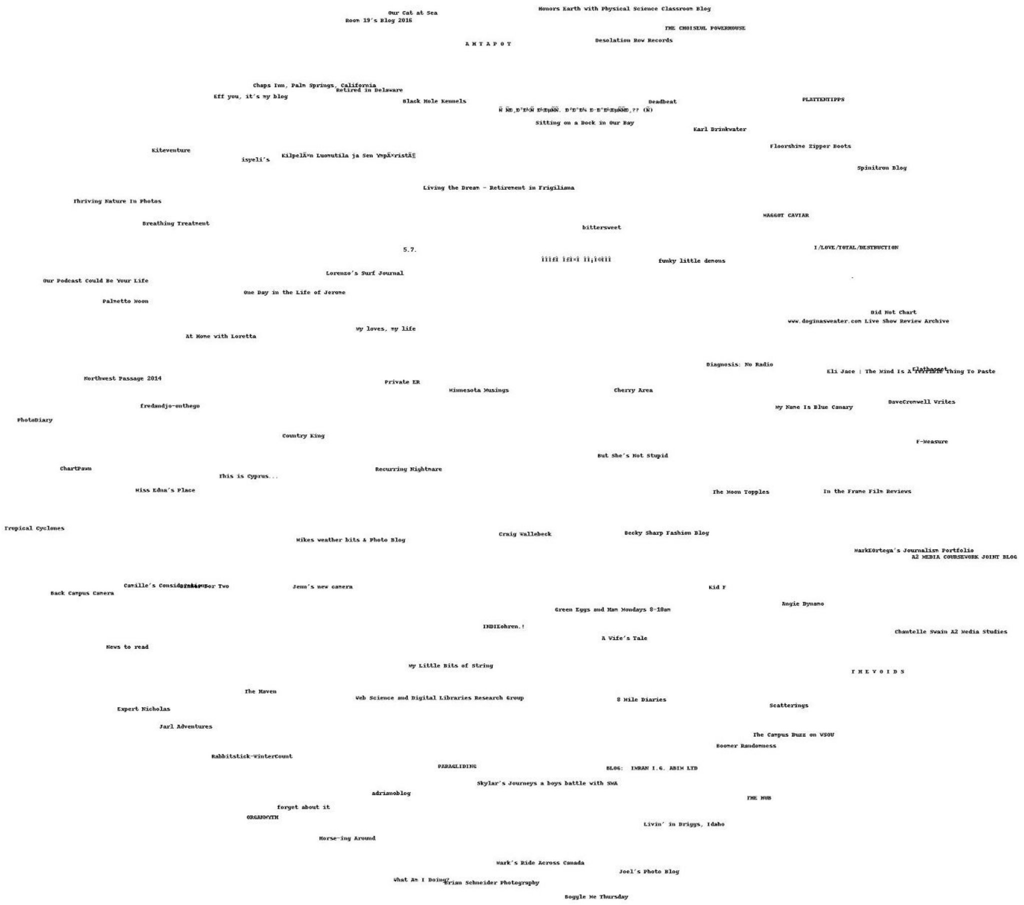
```
For k=5
Iteration 0
Iteration 1
Iteration 2
Iteration 3

For k=10
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4

For k=20
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4


```

4) Use MDS to create a JPEG of the blogs similar to slide 29,

I created the blog space graph using multidimensional scaling (MDS) and the script MDS.py. The book offered a lot of clues as to how to approach this question. I was able to create a nice graph that was readable. To make the graph it required iter = 88.



```
84  3238.20382786
85  3238.10475451
86  3238.06429573
87  3238.01356246
88  3238.02912961
```



P.S. Some of my data is labeled incorrectly because I had to put it in 4 different folders and renamed it just about every time. I attempted the extra credit but was unable to produce a working graph.