Assignment 7

Dr. Nwala

Nathaniel Everett

March 31, 2018

Contents:

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. Note that this method randomly chooses blogs and each student will separately do this process, so it is unlikely that these 98 blogs will be shared among students. In other words, no sharing of blog data. Upload to github your code for grabbing the blogs and provide a list of blog URIs, both in the report and in github.

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. Essentially you are replicating the format of the "blogdata.txt" file included with the PCI book code. Limit the number of terms to the most "popular" (i.e., frequent) 1000 terms, this is *after* the criteria on p. 32 (slide 8) has been satisfied. Remember that blogs are paginated.

My first attempt at this problem was very long-winded and had lots of variables. I ultimately abandoned it for something a bit more simpler and favorable. My solution for this problem took several steps. The first step was where I had trouble with the first time, simply getting the URLs for random blogs. I decided it is better to use a shell script instead of a python script. A do-while loop is implemented:

```
while [ $counter -le 100 ]
do
        url="$(curl -L -s -o /dev/null -w %{url_effective} 'http://www.blogger.com/next-blog?
navBar=true&blogID=3471633091411211117')"
        echo $url >> url.txt
```

As it shows, the URLs go to the file which is created known as url.txt. The files that enter url.txt will not be the same each and every time. The counter is set to stop after 100 blog urls have been received, and can be changed as to how see fit. I added the f-measure.blogspot.com and ws-dl.blogspot.com domains manually, but it is easy to simply add them into the file with the script. The shell script is called blog.sh.

The next step is to use url.txt as the input in a python script, which will be called blogs2.py. The blogs in url.txt have the ?expref=next-blog suffix added to them. This python script simply removes them, but also outputs them into a new file, url.dat. These are the actual blog URLs. The third step is to use another python script, blogs3.py. Again, this one simply does one thing for each URL, and that is to add the feeds/posts/default to the end of the URL. The output file is feed.dat. The final step involves getting the blog matrix set up, taking in feed.dat as the input file. The file used is blogs4.py, taking in several imports and utilizing several functions to get the word counts and the words used in each and every blog. The output is known as blogdata.txt, and is a large file that takes in the words and word counts present in every blog. This output file will be used for the remaining three problems. The blog titles I got from my output of url.dat are shown in the next few pages:

http://f-measure.blogspot.com/
http://ws-dl.blogspot.com/
http://londynsky.blogspot.com/
http://mileinmine.blogspot.com/
http://floorshimezipperboots.blogspot.com/
http://therunoutgroov3.blogspot.com/
http://mtjrrantsravesonmusic.blogspot.com/
http://travelingneighborhood.blogspot.com/
http://davecromwellwrites.blogspot.com/
http://mts-dailythemes.blogspot.com/
http://semregrasluispink.blogspot.com/
http://thefastbreakofchampions.blogspot.com/
http://simonegoes.blogspot.com/
http://hellomynameisjustin.blogspot.com/
http://ohyesjonsi.blogspot.com/
http://www.sonology.com/
http://myopiamuse.blogspot.com/
http://bleakbliss.blogspot.com/
http://coyotedocmusic.blogspot.com/
http://psilabtapes.blogspot.com/
http://flipmpip.blogspot.com/
http://theonionfield.blogspot.com/
http://my-name-is-blue-canary.blogspot.com/
http://nonsensealamode.blogspot.com/
http://didnotchart.blogspot.com/
http://fridaynightrecordparty.blogspot.com/
http://thesehungrytimes.blogspot.com/
http://www.ringtonelirik.com/
http://earenjoy.blogspot.com/
http://musicblog89.blogspot.com/
http://dustandwaterstudios.blogspot.com/
http://mccookerybook.blogspot.com/
http://skiptrack.blogspot.com/
http://cloudbusting87.blogspot.com/
http://rantsfromthepants.blogspot.com/
http://noradiorecs.blogspot.com/
http://thesportsmith.blogspot.com/
http://seven1878.blogspot.com/
http://thefleshyfresh.blogspot.com/
http://doyouneedatv.blogspot.com/
http://stephanieveto.blogspot.com/
http://davecromwellwrites.blogspot.com/
http://musicneedshelp.blogspot.com/
http://organmyth.blogspot.com/
http://cino-pacino.blogspot.com/
http://paulswinnipeg.blogspot.com/
http://bleakbliss.blogspot.com/

http://captainpandamusic.blogspot.com/
http://kerryoconnorsother.blogspot.com/
http://myopiamuse.blogspot.com/
http://musicneedshelp.blogspot.com/
http://thecheatcodesfordrugs.blogspot.com/
http://300vinylchallenge2017.blogspot.com/
http://earenjoy.blogspot.com/
http://www.pewterandpuddles.com/
http://lost-places-hamburg.blogspot.com/
http://kaleidoscopekanvas-kk.blogspot.com/
http://stephanieveto.blogspot.com/
http://flipmpip.blogspot.com/
http://www.tnggames.co.uk/
http://mccookerybook.blogspot.com/
http://laespiraldejosephk.blogspot.com/
http://earenjoy.blogspot.com/
http://www.stereopills.com/
http://www.sonology.com/
http://johnandmaureensanto.blogspot.com/
http://thefastbreakofchampions.blogspot.com/
http://www.sonology.com/
http://blog.spinitron.com/
http://morganflag.blogspot.com/
http://www.punkrockteaching.org/
http://adrianomarquesblog.blogspot.com/
http://avidsblog.blogspot.com/
http://cuzmusicrocks.blogspot.com/
http://doginasweatershowreviews.blogspot.com/
http://youshouldbeinsweden.blogspot.com/
http://encorenorthernireland.blogspot.com/
http://nogiftforthegab.blogspot.com/
http://elijace.blogspot.com/
http://alayerofchips.blogspot.com/
http://zonaboblikessandwiches.blogspot.com/
http://superpersonamia.blogspot.com/
http://fractalpress.blogspot.com/
http://palmiraapistatres.blogspot.com/
http://thefastbreakofchampions.blogspot.com/
http://hiiijaaackie.blogspot.com/
http://theworldsfirstinternetbaby.blogspot.com/
http://justplayingfavorites.blogspot.com/
http://mobbie2.blogspot.com/
http://deeky.blogspot.com/
http://myopiamuse.blogspot.com/
http://superpersonamia.blogspot.com/
http://www.tnggames.co.uk/
http://globalgoon.blogspot.com/

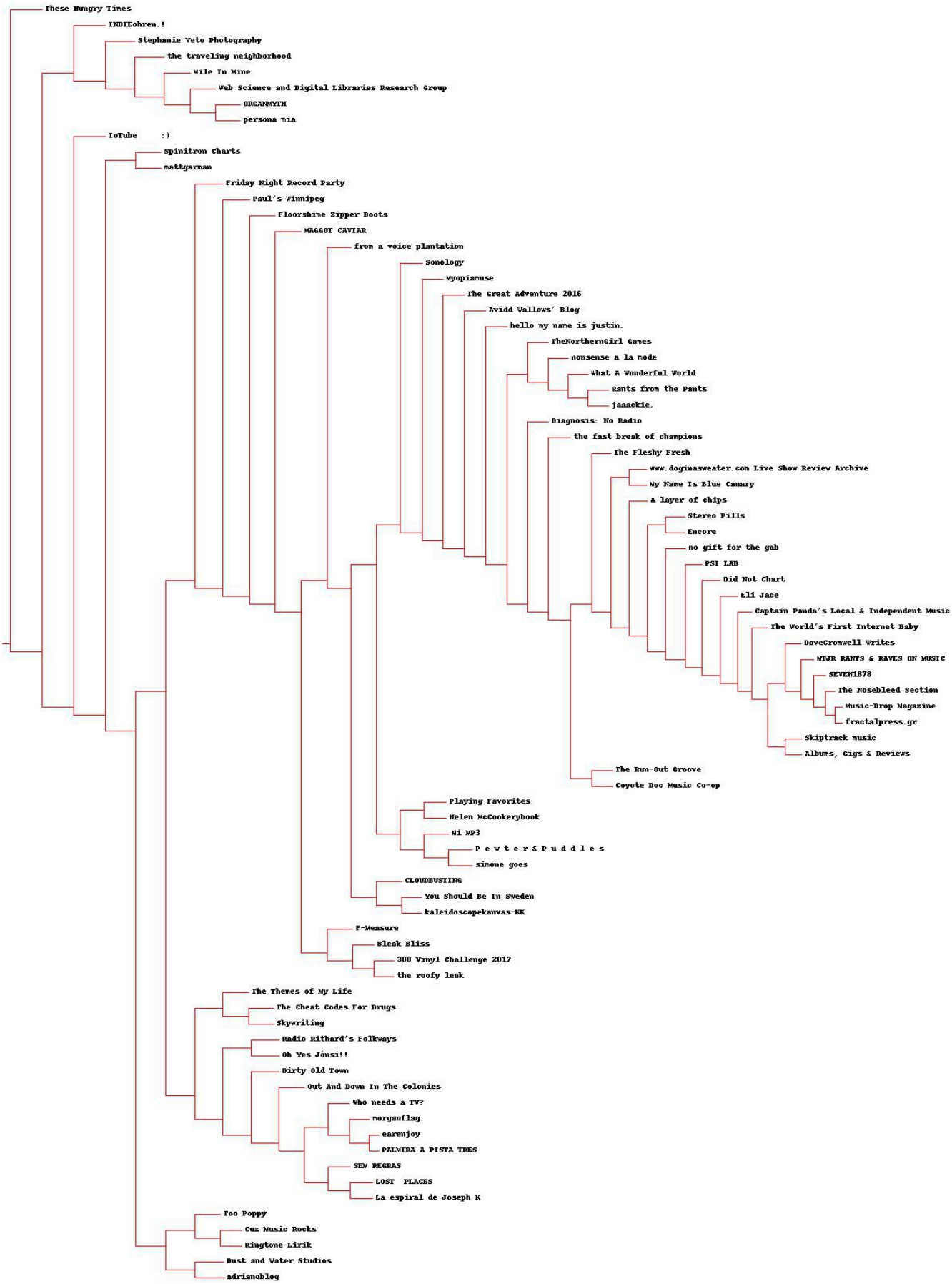
http://radiorithard.blogspot.com/
http://www.toopoppy.com/
http://www.stereopills.com/
http://theworldsfirstinternetbaby.blogspot.com/
http://miemepetres.blogspot.com/
http://maggotcaviar.blogspot.com/
http://outanddowninthecolonies.blogspot.com/
http://cino-pacino.blogspot.com/

---

2. Create an ASCII and JPEG dendrogram that clusters (i.e., HAC) the most similar blogs (see slides 13 & 14). Include the JPEG in your report and upload the ascii file to github (it will be too unwieldy for inclusion in the report).

Again, I was using the final output file from the previous problem (blogdata.txt). I also used clusters.py, which was provided in the problem, although it had to be modified to support python 3, so I included the modified clusters.py in this assignment. The python script here is diagrams.py, outputting both the ASCII diagram and the JPEG diagram. The ASCII is shown via the console using the clusters function printclust, and from there, I copied the output to a txt file, ascii.txt. The JPEG diagram is generated by the clusters script function drawdendrogram, and is also shown in the next page:

```
clusters.printclust(cl,labels=blognames) #ascii diagram
clusters.drawdendrogram(cl,blognames,jpeg='blogcluster.jpg') #drawing the dendrogram
```


These Hungry Times
INDIEohren.!
Stephanie Veto Photography
the traveling neighborhood
Mile In Mine
Web Science and Digital Libraries Research Group
ORGANMYTH
persona mia
IoTube :)
Spinitron Charts
mattyarman
Friday Night Record Party
Paul's Wimipeg
Floorshime Zipper Boots
MAGGOT CAVIAR
from a voice plantation
Sonology
Myopiamuse
The Great Adventure 2016
Avidd Wallows' Blog
hello my name is justin.
TheNorthernGirl Games
nonsense a la mode
What A Wonderful World
Rants from the Pants
jaaackie.
Diagnosis: No Radio
the fast break of champions
The Fleshy Fresh
www.doginasweater.com Live Show Review Archive
My Name Is Blue Canary
A layer of chips
Stereo Pills
Encore
no gift for the gab
PSI LAB
Did Not Chart
Eli Jace
Captain Panda's Local & Independent Music
The World's First Internet Baby
DaveCromwell Writes
MIJR RANTS & RAVES ON MUSIC
SEVEN1878
The Nosebleed Section
Music-Drop Magazine
fractalpress.gr
Skiptrack music
Albums, Gigs & Reviews
The Run-Out Groove
Coyote Doc Music Co-op
Playing Favorites
Helen McCookerybook
Mi MP3
P e w t e r & P u d d l e s
simone goes
CLOUDBUSTING
You Should Be In Sweden
kaleidoscopekamvas-KK
F-Measure
Bleak Bliss
300 Vinyl Challenge 2017
the roofy leak
The Themes of My Life
The Cheat Codes For Drugs
Skywriting
Radio Rithard's Folkways
Oh Yes Jónsi!!
Dirty Old Town
Out And Down In The Colonies
Who needs a TV?
morganflag
earenjoy
PALMIRA A PISTA TRES
SEM REGRAS
LOST PLACES
La espiral de Joseph K
Too Poppy
Cuz Music Rocks
Ringtone Lirik
Dust and Water Studios
adrismoblog

3. Cluster the blogs using K-Means, using k=5,10,20. (see slide 25). Print the values in each centroid, for each value of k. How many iterations were required for each value of k?

Again, using clusters.py and blogdata.txt as input files, this time with kmod.py as the python script used. The function from clusters is kcluster. The code was modified a few times to use each of the k values shown in the problem, ending at k=20, which is the file shown. Omit the lines in the code to get the proper iterations and output for k=5 and k=10. The results I received are shown below.

K=5: 5 iterations required
['adrianoblog', 'Mi MP3']
['300 Vinyl Challenge 2017', 'Bleak Bliss']
['Music-Drop Magazine', 'The Nosebleed Section', 'SEVEN1878', 'kaleidoscopekanvas-KK', 'fractalpress.gr']
['F-Measure', 'INDIEohren.!']
['Sonology', 'The Great Adventure 2016', 'hello my name is justin.', 'ORGANMYTH', 'Web Science and Digital Libraries Research Group']

K=10: 4 iterations required
['300 Vinyl Challenge 2017', 'Bleak Bliss', 'Helen McCookerybook']
['www.doginasweater.com Live Show Review Archive', 'F-Measure', 'The Fleshy Fresh']
['the traveling neighborhood', 'PSI LAB', 'Myopiamuse']
['LOST PLACES', 'SEM REGRAS', 'Out And Down In The Colonies']
['Sonology', 'the fast break of champions', 'Rants from the Pants', 'Dirty Old Town', 'hello my name is justin.', 'nonsense a la mode', 'Spinitron Charts', 'TheNorthernGirl Games', 'Stephanie Veto Photography', 'What A Wonderful World']
['The Run-Out Groove', 'Stereo Pills', 'Floorshime Zipper Boots', 'Did Not Chart', 'the roofy leak', 'Encore', "The World's First Internet Baby", 'Albums, Gigs & Reviews']
[]
['DaveCromwell Writes', "Radio Rithard's Folkways", 'MAGGOT CAVIAR', 'My Name Is Blue Canary']
['The Themes of My Life', 'INDIEohren.!', 'Mi MP3', 'ORGANMYTH', 'persona mia', "Captain Panda's Local & Independent Music Showcase"]
['no gift for the gab', 'Friday Night Record Party', 'Coyote Doc Music Co-op', 'Eli Jace']

K=20: 5 iterations required
['the fast break of champions', 'Friday Night Record Party']
["Paul's Winnipeg", 'Diagnosis: No Radio', 'Myopiamuse']
[]
['Too Poppy', 'MTJR RANTS & RAVES ON MUSIC']
['The Themes of My Life']
['MAGGOT CAVIAR', 'Web Science and Digital Libraries Research Group']
[]
['Mile In Mine', 'Stereo Pills', 'Spinitron Charts', 'These Hungry Times', 'ORGANMYTH', 'persona mia', "Captain Panda's Local & Independent Music Showcase", 'mattgarman']
[]
['A layer of chips', 'The Run-Out Groove', 'Ringtone Lirik', 'TheNorthernGirl Games']

['Music-Drop Magazine', 'The Nosebleed Section', 'SEVEN1878', 'kaleidoscopekanvas-KK', 'fractalpress.gr']
[]
['the traveling neighborhood', 'LOST PLACES', 'Who needs a TV?', 'morganflag', 'La espiral de Joseph K', 'SEM REGRAS', 'earenjoy', 'You Should Be In Sweden', 'PALMIRA A PISTA TRES', 'Out And Down In The Colonies', 'INDIEohren.!']
['Rants from the Pants', 'Dirty Old Town', 'CLOUDBUSTING', 'The Cheat Codes For Drugs', 'Skywriting', 'jaaackie.', 'IoTube :)', "Avidd Wallows' Blog"]
['Sonology', 'hello my name is justin.', 'nonsense a la mode', 'Stephanie Veto Photography', 'What A Wonderful World']
[]
['Playing Favorites', 'Oh Yes Jónsi!!', 'Helen McCookerybook', 'simone goes']
['www.doginasweater.com Live Show Review Archive', "Radio Rithard's Folkways", 'from a voice plantation', 'My Name Is Blue Canary', 'The Fleshy Fresh']
['DaveCromwell Writes', 'no gift for the gab', 'Dust and Water Studios', 'Cuz Music Rocks', 'Floorshime Zipper Boots', 'PSI LAB', 'Did Not Chart', 'Skiptrack music', 'Coyote Doc Music Co-op', 'Encore', 'F-Measure', 'Mi MP3', "The World's First Internet Baby", 'Albums, Gigs & Reviews', 'Eli Jace']
['300 Vinyl Challenge 2017', 'Bleak Bliss', 'The Great Adventure 2016', 'the roofy leak', 'adrianoblog', 'P e w t e r & P u d d l e s']

---

4. Use MDS to create a JPEG of the blogs similar to slide 29 of the week 11 lecture. How many iterations were required?

Once more, clusters.py and blogdata.txt were used. The scaledown and draw2d functions were used. The JPEG file is called blogs2d.jpg and is also shown below:

```
clusters.draw2d(coordinates,blognames,jpeg='blogs2d.jpg') # for mds
```


Spinitron Charts
IRDIEohvnn.!
the traveling neighborhood
Out And Down In The Colonies
These Hungry Tines
The Cheat Codes For Drugs
fractalpress.gr
Ringtone Lirik
Skywriting
persona nia
nattgarman
The Nosebleed Section
Music Drop Magazine
simone goes
You Should Be In Sweden
La espiral de Joseph K
ORGANOVIK
Who needs a TV?
kaleidoscopekanvas-KK
Hi MP3
P o w t e r & P u d d l e s
merymnflay
Cuz Music Rocks
CLOUDBUSTING
Too Poppy
earenjoy
SEVEN1878
Mile In Mine
MIJR RANTS & RAVES ON MUSIC
what A Wonderful World
PALMIRA A FIOTA TNES
The Fleshy Fresh
Wyopiamuse
Friday Night Record Party
PSI LAB
jaaackie.
Diagnosis: No Radio
Encore
TheNorthernGirl Games
Floorshine Zipper Boots
LOST PLACES
Albums, Gigs & Reviews
DaveCromwell Writes
no gift for the gab
www.doginasweater.com Live Show Review Archive
hello my name is justin.
My Name Is Blue Canary
Captain Panda's Local & Independent Music Showcase
Rants frm the Pants
Stereo Pills
Skiptrack music
A layer of chips
Playing Favorites
300 Vinyl Challenge 2017
Coyote Doc Music Co-op
nonsense a la mode
Bleak Bliss
Bid Not Chart
Avidd Wallows' Blog
the roofy leak
Eli Jace
the fast break of champions
frws a voice plantation
Sonology
The World's First Internet Baby
MAGGOT CAVIAR
The Run-Out Groove
The Great Adventure 2016
Stephanie Veto Photography
F-Measure
Helen McCookerybook
Oh Yes Jonsi!!
Paul's Winnipeg
SEM REGRAS
Radio Rithard's Folkways
adrimsblog
The Theses of My Life
Web Science and Digital Libraries Research Group
Dust and Water Studies
Dirty Old Town
IoTube :)

Files included:
clusters.py (required for all questions except the first one, *main reference*)
blog.sh (a shell script to output url.txt, 102 URLs, can change to whatever number needed)
url.txt (output from blog.sh)
blogs2.py (python, removes ?expref=next-blog from the urls in url.txt and outputs them to url.dat)
url.dat (output from blogs2.py)
blogs3.py (python, adds feeds/posts/default to the urls in url.dat and outputs them into feed.dat)
feed.dat (output from blogs3.py)
blogs4.py (creates the blogmatrix in blogdata.txt, uses feed.dat as input
blogdata.txt (blogmatrix output from blogs4.py)
diagrams.py (inputs blogdata.txt outputs a dendrogram in ASCII format (looks terrible) and in a JPEG blogcluster.jpg)
ascii.txt (ASCII representation of the dendrogram)
blogcluster.jpg (jpg of dendrogram from diagrams.py)
kmod.py (calculates kmeans, look at the comments)
mds.py (inputs blogdata.txt, creates a JPEG of blogs in blogs2d.jpg)
blogs2d.jpg (data output from mds.py)

(note that even though the diagrams are included in the report, it is better to view them on github rather than in the report itself).