CS 432/532

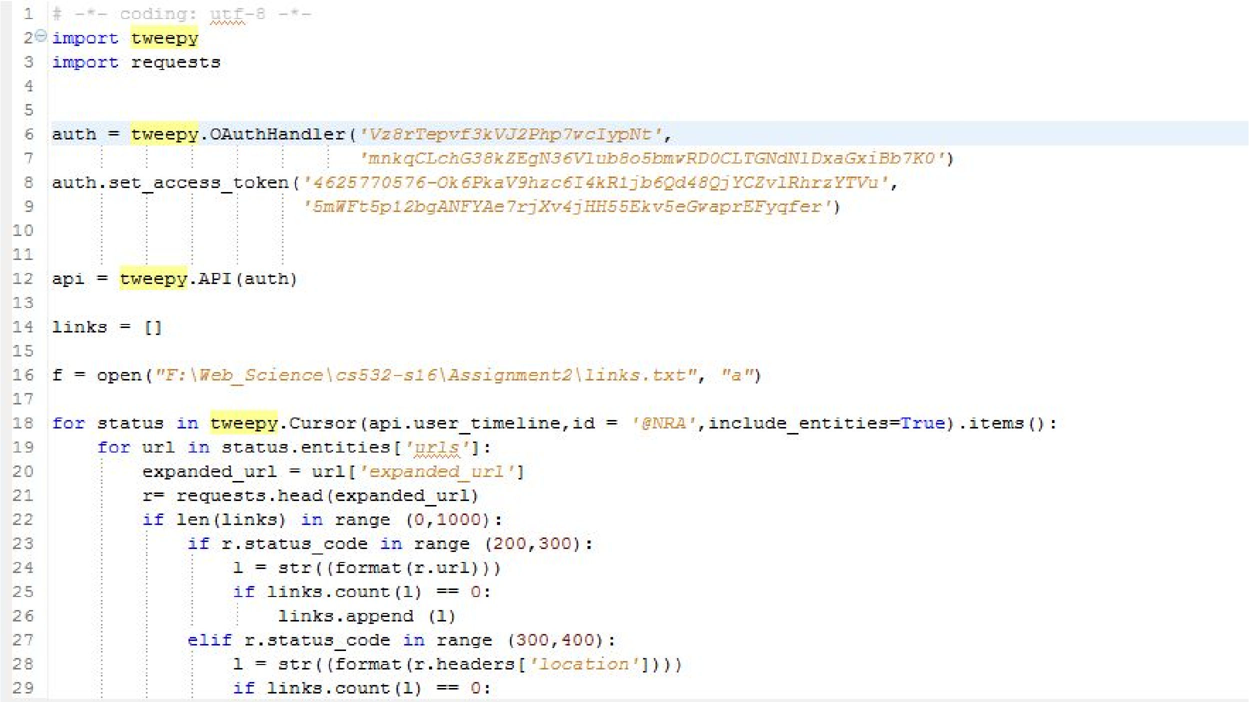
Assignment 2

2/11/2016

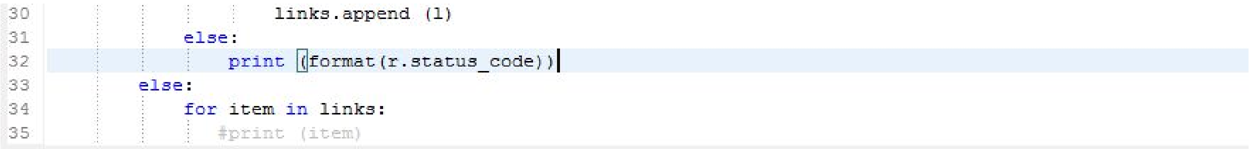
Daniel Moore

Problem 1:

Write a python program that extracts 1000 unique links from Twitter.



```
# -*- coding: utf-8 -*-
import tweepy
import requests


auth = tweepy.OAuthHandler('Vz8rTepvf3kVJ2Php7wcIypNt',
                           'mnkqCLchG38kZEgN36Vlub8o5bmwRD0CLTGNdN1DxaGxiBb7K0')
auth.set_access_token('4625770576-Ok6PkaV9hzc6I4kR1jb6Qd48QjYCZvlRhrzYTVu',
                      '5mWFt5p12bgANFYAe7rjXv4jHH55Ekv5eGwaprEFyqfer')


api = tweepy.API(auth)

links = []

f = open("F:\Web_Science\cs532-s16\Assignment2\links.txt", "a")

for status in tweepy.Cursor(api.user_timeline,id = '@NRA',include_entities=True).items():
    for url in status.entities['urls']:
        expanded_url = url['expanded_url']
        r= requests.head(expanded_url)
        if len(links) in range (0,1000):
            if r.status_code in range (200,300):
                l = str((format(r.url)))
                if links.count(l) == 0:
                    links.append (l)
            elif r.status_code in range (300,400):
                l = str((format(r.headers['location'])))
                if links.count(l) == 0:
```

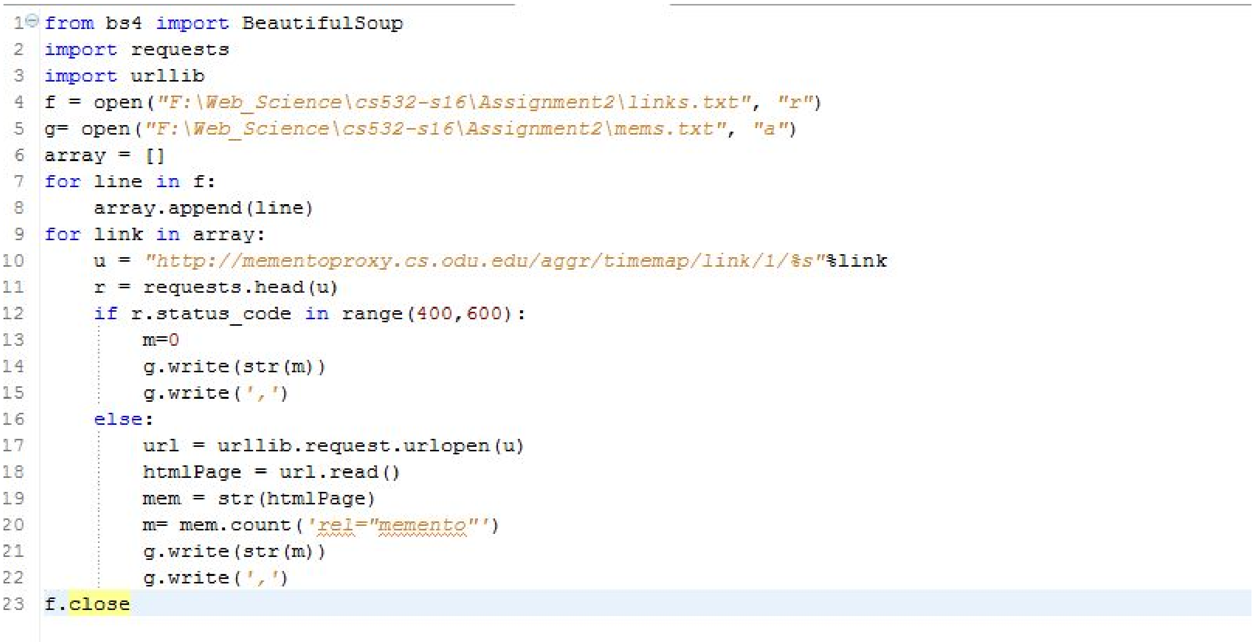


```
                    links.append (l)
            else:
                print (format(r.status_code))
        else:
            for item in links:
                #print (item)
```

Above is a program written in Python 3.4 which extracts the links.

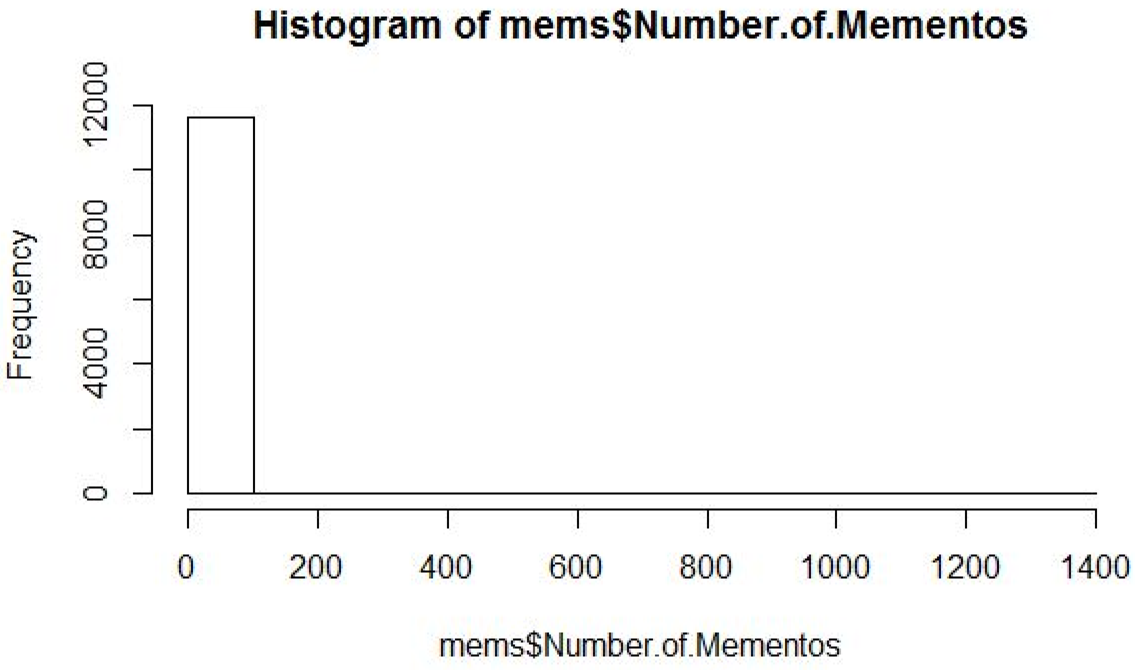
Problem 2:

Using the ODU Memento Aggregator download the time maps for the 1000 links collected in phase 1.



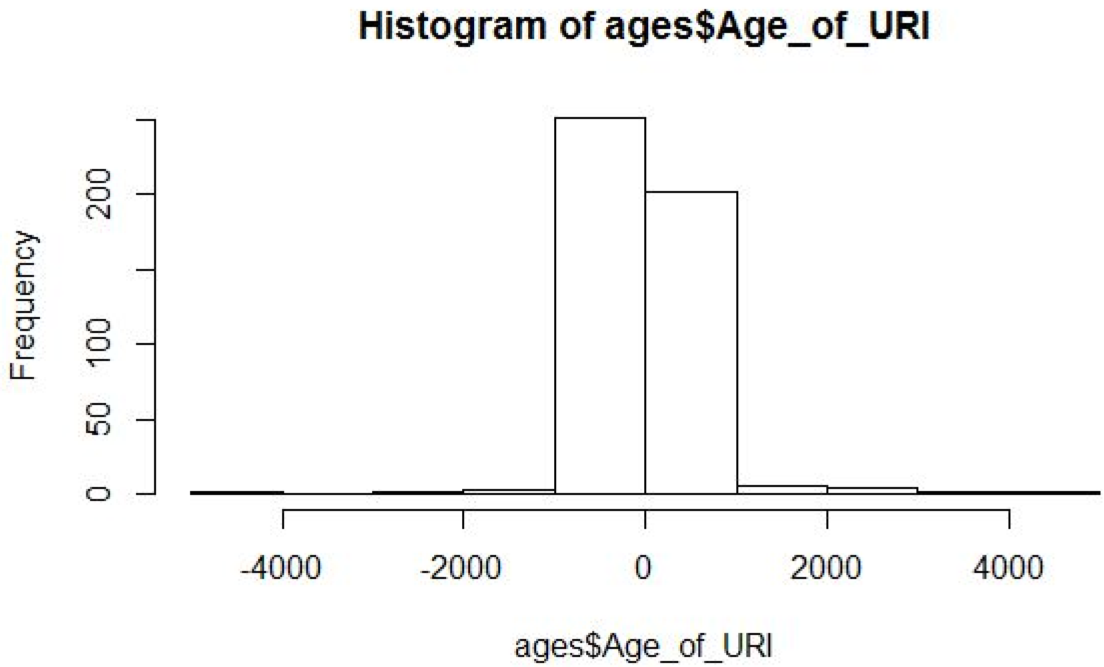
```python
from bs4 import BeautifulSoup
import requests
import urllib
f = open("F:\Web_Science\cs532-s16\Assignment2\links.txt", "r")
g= open("F:\Web_Science\cs532-s16\Assignment2\mems.txt", "a")
array = []
for line in f:
    array.append(line)
for link in array:
    u = "http://mementoproxy.cs.odu.edu/aggr/timemap/link/1/%s"%link
    r = requests.head(u)
    if r.status_code in range(400,600):
        m=0
        g.write(str(m))
        g.write(',')
    else:
        url = urllib.request.urlopen(u)
        htmlPage = url.read()
        mem = str(htmlPage)
        m= mem.count('rel="memento"')
        g.write(str(m))
        g.write(',')
f.close
```

Above is the code for downloading the time maps and recording the number of mementos for each URI in a file. Second part of this portion was to plot the frequenies of each number of mementos.



Somehow my code looped too many times and generated almost 12,000 results. The number of URIs which either had 0 mementos or returned a 400 or 500 error were very disproportionate.

ThT



The number of URIs which contained no origination dates when ran though the carbon date tool was also disproportionate.