

# CS532 Web Science: Assignment 2

Finished on February 11, 2016

*Dr. Michael L. Nelson*

**Naina Sai Tipparti**  
ntippart@cs.odu.edu

# Contents

<b>Problem 1</b>	<b>2</b>
Question . . . . .	2
Answer . . . . .	2
<b>Problem 2</b>	<b>6</b>
Question . . . . .	6
Answer . . . . .	6
<b>Problem 3</b>	<b>9</b>
Question . . . . .	9
Answer . . . . .	9

# Listings

1	urifinder.py . . . . .	2
2	Sample of 1000 Links . . . . .	5
3	mementofinder.py . . . . .	6
4	build_histogram.r . . . . .	7
5	Sample of Memento Links . . . . .	7
6	carbodate.py . . . . .	9
7	build_ecd_mementos.py . . . . .	10
8	ecd_mementos_graph.r . . . . .	10
9	Sample of Carbon Data Links . . . . .	11

# List of Figures

1	Histogram of Site Mementos . . . . .	8
2	Estimated Age to Memento Count . . . . .	12

# Problem 1

## Question

Write a Python program that extracts 1000 unique links from Twitter. You might want to take a look at:

<http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/>

But there are many other similar resources available on the web. Note that only Twitter API 1.1 is currently available; version 1 code will no longer work.

Also note that you need to verify that the final target URI (i.e., the one that responds with a 200) is unique. You could have many different shortened URIs for [www.cnn.com](http://www.cnn.com) (t.co, bit.ly, goo.gl, etc.).

You might want to use the search feature to find URIs, or you can pull them from the feed of someone famous (e.g., Tim O'Reilly).

Hold on to this collection – we'll use it later throughout the semester.

## Answer

Using the python module requests made this task a breeze as well as the initial code provided by Thomas Sileo's blog post.

```

1  # -*- encoding: utf-8 -*-
   import requests
   from requests_oauthlib import OAuth1
   from urllib import quote

6  REQUEST_TOKEN_URL = "https://api.twitter.com/oauth/request_token"
   AUTHORIZE_URL = "https://api.twitter.com/oauth/authorize?oauth_token="
   ACCESS_TOKEN_URL = "https://api.twitter.com/oauth/access_token"

   CONSUMER_KEY = "c7VFfCTtUqFDg69MRHvGnpSwt"
11  CONSUMER_SECRET = "4r8yeBQmzi08HuXY3UQN3qGigz0hgrbQxXpD4w2nR7fBRIRLqU"

   OAUTH_TOKEN = "798668178-bH8DbMpNuWkfAHxu0DgWSHwQE65B1WZnc4Ahtej"
   OAUTH_TOKEN_SECRET = "FhykPKnQcgKQBE43os2bDZ31ugH9RVSG3HYo0L7QG7RNC"

16  SEARCH_URI = "https://api.twitter.com/1.1/search/tweets.json?q="

   SEARCH_ITEMS = map(quote, [
                                'allu arjun',
                                'vijayawada',
                                'Cristiano Ronaldo',
21                                'Adolf Hitler',
                                'Emma Watson',
                                'Anushka Shetty',
                                'Rajini Kanth',
                                'Lionel Messi',
26                                'Ricardo Kaka',
                                'Wayne Rooney',
                                'Gareth Bale',
                                'Neymar',

```

31	'Malaika Arora Khan',
	'Isco',
	'Adam Gilchrist',
	'Kevin de bruyne',
	'Kevin Pietersen',
36	'Julia Roberts',
	'Scarlett Johansson',
	'Kate Winslate',
	'Kamal Hassan',
	'Kumar sangakkara',
41	'Mahela jayawardene',
	'Roger Federer',
	'Maria Sharapova',
	'Serena Williams',
	'Venus Williams',
46	'Justin Henin',
	'Justin Langer',
	'Sania Mirza',
	'Shoaib Malik',
	'Shoaib Akhtar',
51	'Sourav Ganguly',
	'Rafeal Nadal',
	'Real Madrid CF',
	'Rahul Dravid',
	'Manchester United',
56	'FC Barcelona',
	'Manchester City',
	'Chelsea FC',
	'Liverpool FC',
	'Aston Villa FC',
61	'Aston Martin',
	'lamborghini',
	'FC Bayern Munich',
	'Borussia Dortmund',
	'Philipp Lahm',
66	'Sergio Ramos',
	'Sergio Aguero',
	'Marcelo Vieira',
	'Luis Suarez',
	'Luis Enrique',
71	'Jose Mourinho',
	'Alex Ferguson',
	'Pep Guardiola',
	'Shane Watson',
	'Shane Warne',
76	'Bill Gates',
	'Rothschild',
	'Mark Zuckerberg',
	'Chetan Bhagat',
	'Ishant Sharma',
81	'Virat Kohli',
	'Sachin Tendulkar',
	'Ricky Ponting',
	'Matthew Hayden',
	'Rohit Sharma',
86	'Irfan Pathan',
	'yusuf Pathan',
	'Katrina Kaif',
	'Anushka Sharma',
	'Salman Khan',
91	'Ranbir Kapoor',
	'Puneeth Rajkumar',
	'Rajkumar',
	'Ganesh',
	'Upendra',
96	'Darshan',
	'Shivraj Kumar',
	'Rakhul Preeth Singh',

```

101         'Soundarya',
        'Savitramma',
        'Anusuya',
        'Roja',
        'Ram Gopal Varma',
        'Pawn Kalyan',
        'Ram Charan',
106        'Allu Arvind',
        'Allu Ramalinga',
        'Kotta Srinivas Rao',
        'Surya',
        'Joythika',
        'Nagarjun',
111        'Akhil Akkeneni',
        'Naga Chaitanya Akkeneni',
        'Amala Akkeneni',
        'Barack Obama',
        'Abdul Kalam',
116        'Subhash Chandra Bose',
        'Shreya Ghoshal',
        'Shreya Sharan',
        'Sidharth',
        'Mahesh Babu',
121        'Jenelia ',
        'Salam Khan',
        'Sharukh Khan',
        'Hrithik Roshan',
        'Deepika Padukone',
126        'Pooja Ghandhi',
        'Ravichandranan',
        'Arjun Sarja',
        'Shankar Nag',
        'Prakash Raj',
131        'Tennis Krishna',
        'Nagesh',
        'Lokanath',
        'Meena',
        'Arundathi Nag',
136        'Nagma',
        'Laila Mehdiin',
        'Sindhu Menon',
        'Jaylalithaa',
        'Vishnuvardhan',
141        'Suman Nagarkar']]

def get_oauth():
    return OAuth1(CONSUMER_KEY,
                  client_secret=CONSUMER_SECRET,
146                  resource_owner_key=OAUTH_TOKEN,
                  resource_owner_secret=OAUTH_TOKEN_SECRET)

def find_uris(uris):
    with open('output', 'a') as outfile:
151        for search_item in SEARCH_ITEMS:
            result = requests.get(SEARCH_URI + search_item + '&filter%3Alinks&count=10000',
                                auth=oauth)
            for status in result.json()['statuses']:
                for url in status['entities']['urls']:
                    if len(uris) == 10000:
156                        return
                    if 'expanded_url' in url:
                        try:
                            result = requests.get(url['expanded_url'], timeout=4)
                            # only add expanded uris if they aren't in the list already
161                            if result.status_code == 200 and result.url not in uris:
                                add_uri(uris, result.url)
                                outfile.write('%s\n' % result.url)
                        except Exception as e:

```

```

166             print e
                continue

def add_uri(uris, uri):
    uris.add(uri)
    print 'added uri %d: %s' % (len(uris), uri)
171
if __name__ == "__main__":
    oauth = get_oauth()
    uris = set()
    # read in previous set of uris
176    try:
        with open('output', 'r') as infile:
            for line in infile.readlines():
                add_uri(uris, line.strip())
    except IOError:
181        pass
    find_uris(uris)

```

Listing 1: urifinder.py

The script was run multiple times to get the desired 1000 unique URIs. It would end prematurely at times, so the data set was initialized with the data of the previous run and then passed on to the `find_uris` function to preserve work performed.

```

https://www.youtube.com/watch?v=fEhXgOKpE-o&feature=youtu.be&a
https://twitter.com/FutNaRedonda/status/696095799013478402
3 https://twitter.com/dr_javi/status/693995331164463104/photo/1
https://www.youtube.com/watch?v=Ig-kBwOdLg&feature=youtu.be&a
https://www.youtube.com/watch?v=yypauzQV8d0&feature=youtu.be&a
https://twitter.com/asksparati/status/696367298525978624
https://twitter.com/History_Futbol/status/694720215087652864/photo/1
8 http://www.dainikbhaskar.info/sports/kids-should-learn-from-soaring-leicester-not-superstars
-lionel-messi-and-cristiano-ronaldo-they-show-the-value-of-hard-graft/
https://twitter.com/Juezcentral/status/696515201743659008
http://www.oldpicsarchive.com/selected-photos-part-4-33-rare-pics/4/?utm_content=buffer840ce
&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
http://www.georgewalkerbush.net/bushfamilyfundedhitler.htm
http://odia.ig.com.br/noticia/rio-de-janeiro/2016-02-03/justica-proibe-venda-e-divulgacao-de
-livro-escrito-por-adolf-hitler.html
13 http://www.telegraph.co.uk/news/worldnews/donald-trump/12038640/Who-said-it-Donald-Trump-or-
Adolf-Hitler.html
https://www.youtube.com/watch?v=d3r70E6Dvfs&feature=youtu.be&a
https://www.youtube.com/watch?v=sI1E4Vs7cbk&feature=youtu.be&a
http://www.newsweek.com/adolf-hitler-black-holocaust-dark-secrets-423735?rx=us
http://www.flimper.com/events/4
18 https://www.facebook.com/robcaiafa/posts/10208639359731659
http://linkis.com/www.youtube.com/lrpOF
https://www.youtube.com/watch?v=QnpBN-ltVtE&feature=youtu.be

```

Listing 2: Sample of 1000 Links

## Problem 2

### Question

Download the TimeMaps for each of the target URIs. We'll use the ODU

Memento Aggregator, so for example:

URI-R = <http://www.cs.odu.edu/>

URI-T = <http://mementoproxy.cs.odu.edu/aggr/timemap/link/1/http://www.cs.odu.edu/>

Create a histogram\* of URIs vs. number of Mementos (as computed from the TimeMaps). For example, 100 URIs with 0 Mementos, 300 URIs with 1 Memento, 400 URIs with 2 Mementos, etc.

\* = <https://en.wikipedia.org/wiki/Histogram>

### Answer

The python script in Listing 3 was used to retrieve the timemaps and then parse the returned html, traveling down the rabbit hole if the target URI has more than 1000 mementos.

```

# -*- encoding: utf-8 -*-
#! /usr/bin/python

import requests
5 import re

MW_URI = "http://mementoproxy.cs.odu.edu/aggr/timemap/link/1/"

10 if __name__ == '__main__':
    with open('output', 'r') as f:
        output = open('site_mementos', 'w')
        mementos = {}
        for uri in f.read().split('\n'):
            if uri is '':
                continue
            count = 0
            target_uri = MW_URI + uri
            while True:
                result = requests.get(target_uri)
                if result.ok:
                    count = count + result.text.count('rel="memento"')
                    last_line = result.text.split('\n')[-1]
                    if 'rel="timemap"' not in last_line:
                        break
                15 sites = re.findall(r'<([^\>]+)>', last_line)
                target_uri = sites[1]
            mementos[uri] = count
            print 'found %d mementos for uri: %s' % (count, uri)
            output.write('%s %d\n' % (uri, count))
20
30 output.close()

```

Listing 3: mementofinder.py

The dataset created Listing 3. A log scale was used along the y-axis to show more detail among the results. The script in Listing 4 was used to create the histogram in Figure 1, which shows the distribution of mementos per site from the dataset of Question 1.

```
#!/usr/bin/Rscript

data <- read.table("D:/cs532/q2/results", header=TRUE, comment.char="")
counts <- table(data$Mementos)
5 pdf("histogram.pdf")
barplot(counts, log="y", ylim=c(.75, nrow(data)), ylab="Sites", xlab="Memento Count", main="
  Memento Count per Site")
dev.off()
```

Listing 4: build\_histogram.r

```
https://twitter.com/asksparati/status/696367298525978624 0
https://twitter.com/History_Futbol/status/694720215087652864/photo/1 0
3 http://www.dainikbhaskar.info/sports/kids-should-learn-from-soaring-leicester-not-superstars
  -lionel-messi-and-cristiano-ronaldo-they-show-the-value-of-hard-graft/ 0
https://twitter.com/Juecentral/status/696515201743659008 0
http://www.oldpicsarchive.com/selected-photos-part-4-33-rare-pics/4/?utm_content=buffer840ce
  &utm_medium=social&utm_source=twitter.com&utm_campaign=buffer 0
http://www.georgewalkerbush.net/bushfamilyfundedhitler.htm 79
http://odia.ig.com.br/noticia/rio-de-janeiro/2016-02-03/justica-proibe-venda-e-divulgacao-de
  -livro-escrito-por-adolf-hitler.html 0
8 http://www.telegraph.co.uk/news/worldnews/donald-trump/12038640/Who-said-it-Donald-Trump-or-
  Adolf-Hitler.html 7
https://www.youtube.com/watch?v=d3r70E6Dvfs&feature=youtu.be&a 16
https://www.youtube.com/watch?v=sI1E4Vs7cbk&feature=youtu.be&a 16
http://www.newsweek.com/adolf-hitler-black-holocaust-dark-secrets-423735?rx=us 2
http://www.flimper.com/events/4 0
13 https://www.facebook.com/robcaiafa/posts/10208639359731659 0
http://linkis.com/www.youtube.com/lrpOF 0
https://www.youtube.com/watch?v=QnpBN-ltVtE&feature=youtu.be 16
http://daveschlueteronline.com/the-penguin-updates-and-seo/ 3
https://twitter.com/MrJohnQZombie/status/664324668149493760/photo/1 0
18 https://twitter.com/MrJohnQZombie/status/664324668149493760 0
http://www.nzherald.co.nz/entertainment/news/article.cfm?c_id=1501119&objectid=11586082&ref=
  rss&utm_source=dlvr.it&utm_medium=twitter 0
http://frtyb.com/go/Ogl_bnwHp-j4D2/DEFAULT 0
https://www.youtube.com/watch?v=cZKeqenZbJk&feature=youtu.be&a 16
```

Listing 5: Sample of Memento Links



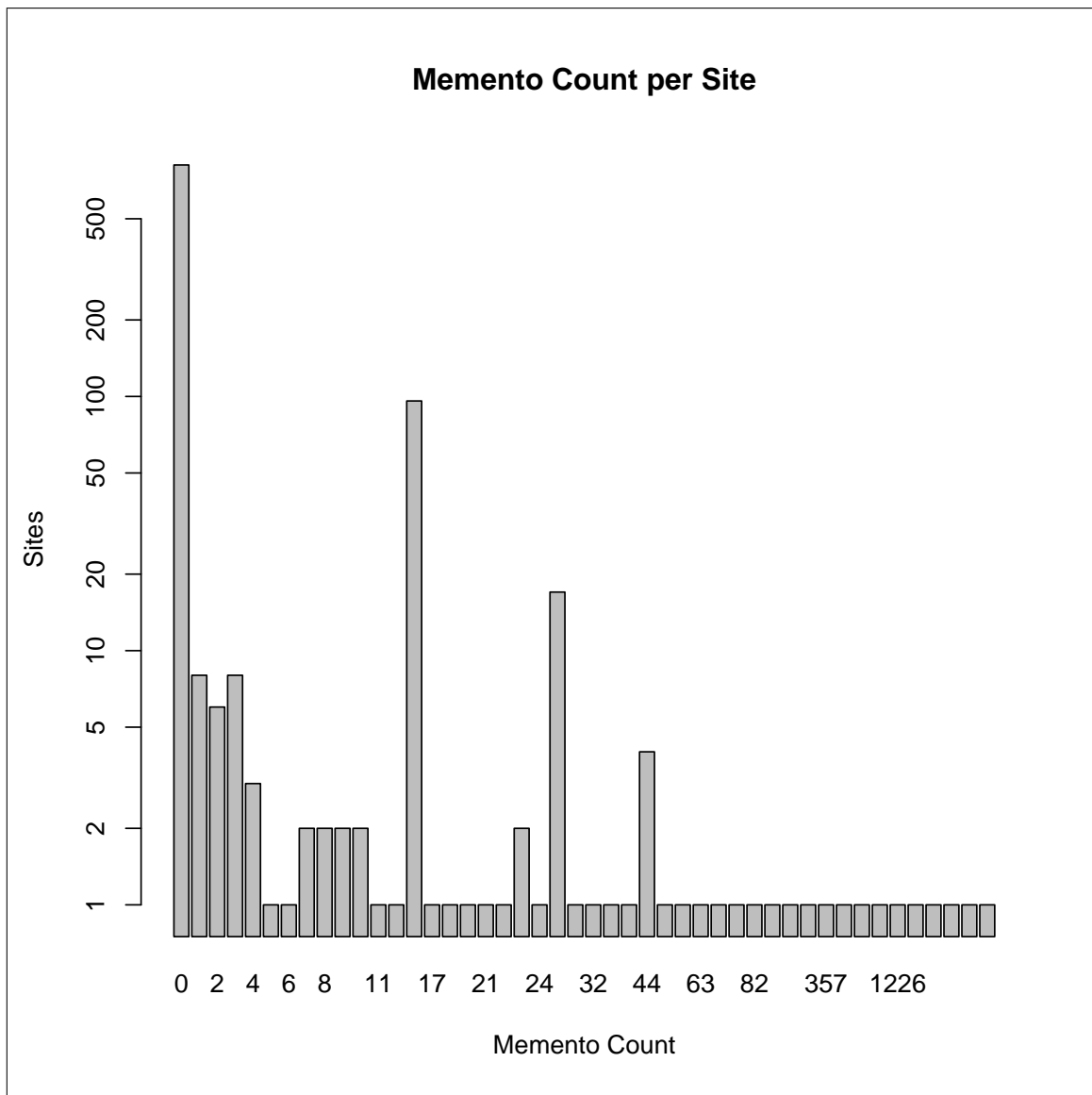


Figure 1: Histogram of Site Mementos

## Problem 3

### Question

Estimate the age of each of the 1000 URIs using the "Carbon Date" tool:

<http://ws-dl.blogspot.com/2014/11/2014-11-14-carbon-dating-web-version-20.html>

Note: you'll should download the library and run it locally; don't try to use the web service.

For URIs that have  $\geq 0$  Mementos and an estimated creation date, create a graph with age (in days) on one axis and number of mementos on the other.

Not all URIs will have Mementos, and not all URIs will have an estimated creation date. State how many fall into either categories.

### Answer

Using the python script in Listing 6 to utilize the tools from the "Carbon Date" tool provided by H.S. Eldeen, estimated creation dates were found for 843 of the original URIs. These were stored in the file `site_ecd_all`.

```

3  #!/usr/bin/python
from local import cd
import json
import concurrent.futures

ECD = 'Estimated Creation Date'

8  def getdate(uri):
    uri = uri.strip()
    print 'Searching for uri: {}'.format(uri)
    uri_json = json.loads(cd(uri))
13    if uri_json[ECD]:
        print 'Found creation date: {}'.format(uri_json[ECD])
        return (uri, uri_json[ECD])
    else:
        print 'Found no ECD'
18    return None

if __name__ == '__main__':
    # Remove already completed links
    with open('site_mementos') as infile:
23        input_uris = [line.split(' ')[0] for line in infile if line.rstrip('\n').split(' ')[1] != '0']
    with open('site_ecd_all') as prevfile:
        prev = [line.split(' ')[0] for line in prevfile]
    uris = [uri for uri in input_uris if uri not in prev]
    print 'Starting on uri {}'.format(len(uris))
28
    # Work on the rest
    with open('site_ecd_all','a') as outfile:
        with concurrent.futures.ThreadPoolExecutor(max_workers=8) as executor:
            urifutures = [executor.submit(getdate, uri) for uri in uris]
33        for future in concurrent.futures.as_completed(urifutures):

```

```

38         try:
            data = future.result()
        except Exception as exc:
            print '{} generated an exception: {}'.format(uri, exc)
        if len(data) == 2:
            print 'Writing data: {}'.format(data)
            outfile.write('{} {}\n'.format(data[0], data[1]))
        else:
            print 'Found no data'

```

Listing 6: carbondate.py

To prepare the dataset for graphing, the Python script in Listing 7 was used to capture the desired subset of data from the `site_ecd_all` file; i.e, those site-memento pairs where the memento count was greater than zero. The results were stored in the file `ecd_mementos`.

```

3  #!/usr/bin/python
  if __name__ == '__main__':
      with open('site_ecd_all') as ecdfile:
          ecdmap = dict([line.rstrip('\n').split() for line in ecdfile])
      with open('site_mementos') as memfile:
          mementos = dict([line.rstrip('\n').split() for line in memfile if line.rstrip('\n').
8              split()[1] != '0'])
      with open('ecd_mementos', 'w') as outfile:
          outfile.write('ECD Mementos\n')
          for uri, mem in mementos.iteritems():
              try:
13                  outfile.write('{} {}\n'.format(ecdmap[uri], mem))
              except KeyError as e:
                  print '{}\n'.format(e))

```

Listing 7: build\_ecd\_mementos.py

Using the R script in Listing 8, with the dataset obtained from Listing 7 (the `ecd_mementos` file), the graph in Figure 2 was created. This graph shows that the older a site is the higher that site's memento count tends to be. It also shows that there are a far greater number of new sites than old ones.

```

1  #!/usr/bin/Rscript
  data <- read.table("D:/cs532/q3/ecd_mementos", header=TRUE)
  strtodate = function(x) {
      result = Sys.time() - strptime(x, "%Y-%m-%dT%H:%M:%S")
6      return(result)
  }
  data$ECD <- lapply(data$ECD, strtodate)
  pdf("ecd_mementos.pdf")
  plot(data, log="y", xlab="Estimated Site Age in days", ylab="Number of Mementos", main="
11  Estimated Site Age to Memento Count")
  dev.off()

```

Listing 8: ecd\_mementos\_graph.r

```

http://cinerocks.in/Latest-Tamil-Actress-gallery/Anushka-Shetty-Stills/236/11 2015-05-04T19:47:27
http://www.deccanchronicle.com/entertainment/bollywood/040216/shah-rukh-khan-takes-onus-of-dilwale-loss-refunds-50-percent-to-the-distributors.html 2016-02-04T00:00:00
https://www.youtube.com/watch?v=I8DBBiwecq8&feature=youtu.be&a 2008-06-20T06:57:31
4 https://www.youtube.com/watch?v=a_qhLAwnCRY 2008-06-20T06:57:31
https://www.youtube.com/watch?v=Eub52KkPNXQ&feature=youtu.be&a 2008-06-20T06:57:31
https://www.youtube.com/watch?v=3BiHvbH4Wow&feature=youtu.be 2008-06-20T06:57:31
https://www.youtube.com/watch?v=MkD6AqAVEh0&feature=youtu.be&a 2008-06-20T06:57:31
https://www.youtube.com/watch?v=Xl2FpTqzp10&feature=youtu.be&a 2008-06-20T06:57:31
9 https://www.888sport.com/treble888.htm 2015-04-13T20:39:30
http://www.dailymail.co.uk/sport/football/article-3436302/Kids-learn-soaring-Leicester-not-superstars-Lionel-Messi-Cristiano-Ronaldo-value-hard-graft.html 2016-02-08T00:26:51
https://www.youtube.com/watch?v=j9u1Uj5CPmg&feature=youtu.be 2008-06-20T06:57:31
https://www.youtube.com/watch?v=atZssXHTjWg&feature=youtu.be&a 2008-06-20T06:57:31
http://www.dailymail.co.uk/sport/football/article-3435795/Levante-v-Barcelona-LIVE-Follow-action-Lionel-Messi-Luis-Suarez-Neymar-La-Liga-action.html 2016-02-07T12:14:15
14 http://www.dailymail.co.uk/sport/football/article-3436302/Kids-learn-soaring-Leicester-not-superstars-Lionel-Messi-Cristiano-Ronaldo-value-hard-graft.html?ITO=1490&ns_mchannel=rss&ns_campaign=1490 2009-10-29T00:00:00
https://www.youtube.com/watch?v=bCgajKUvQow 2008-06-20T06:57:31
https://www.facebook.com/photo.php?fbid=894707853975568 2010-01-05T05:44:35
https://www.youtube.com/watch?v=bCgajKUvQow&feature=youtu.be 2008-06-20T06:57:31
https://www.youtube.com/watch?v=4g2m1CuG_0U&feature=youtu.be 2008-06-20T06:57:31
19 https://www.facebook.com/photo.php?fbid=1090325967667020 2010-01-05T05:44:35

```

Listing 9: Sample of Carbon Data Links

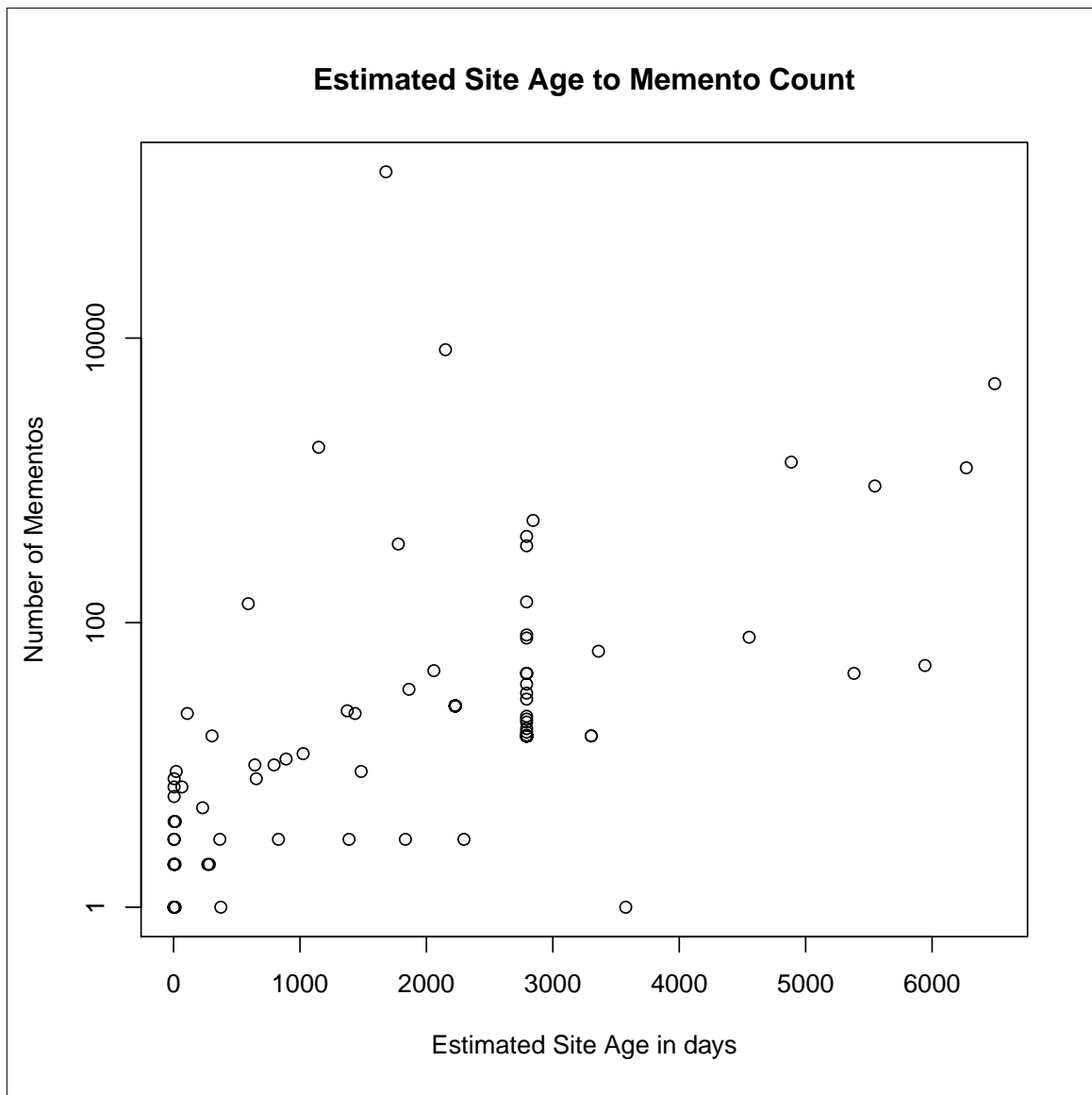


Figure 2: Estimated Age to Memento Count

## References

- [1] Getting Started with Twitter API: <http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/>
- [2] Twitter Search API: <https://dev.twitter.com/rest/public/search>
- [3] Twitter API - Get / Search Tweets: <https://dev.twitter.com/rest/reference/get/search/tweets>
- [4] R: <http://www.cs.odu.edu/~sainswor/uploads/Teaching/cs595f13-R.pdf>
- [5] R functions: <http://stackoverflow.com/questions/16860200/row-by-row-operations-and-updates-in-data-table>
- [6] R lapply: <http://stackoverflow.com/questions/20766617/using-apply-functions-in-place-of-for-loop-in-r>
- [7] R strptime: <https://stat.ethz.ch/R-manual/R-devel/library/base/html/strptime.html>
- [8] <https://github.com/phonedude/>