1. Extract Webpage Contents using XML

Code

In [ ]:
```
rm(list=ls())
library(XML)
library(rvest)
library(data.table)
u="https://en.wikipedia.org/wiki/ASCII"
dt1=read_html(u)%>%html_table(fill=TRUE)%>%.[[3]]
dt2=read_html(u)%>%html_table(fill=TRUE)%>%.[[4]]
dt3=read_html(u)%>%html_table(fill=TRUE)%>%.[[5]]

dt_all <- as.data.table(rbind(dt1, dt2, dt3))
colnames(dt_all) <- c("Binary", "Oct", "Dec", "Hex", "Glyph'63", "Glyph'65",
"Glyph'67")

#Get rid of the rows with column names
dt_all <- dt_all[Binary!="Binary"]

dim(dt_all)
str(dt_all)
head(dt_all)
tail(dt_all)
```

Output

In [ ]:
```
> dim(dt_all)
[1] 95  7

> head(dt_all)
     Binary Oct Dec Hex Glyph'63 Glyph'65 Glyph'67
1: 010 0000 040  32  20  (space)  (space)  (space)
2: 010 0001 041  33  21        !        !        !
3: 010 0010 042  34  22        "        "        "
4: 010 0011 043  35  23        #        #        #
5: 010 0100 044  36  24        $        $        $
6: 010 0101 045  37  25        %        %        %

> tail(dt_all)
     Binary Oct Dec Hex Glyph'63 Glyph'65 Glyph'67
1: 111 1001 171 121  79                 y        y
2: 111 1010 172 122  7A                 z        z
3: 111 1011 173 123  7B                 {        {
4: 111 1100 174 124  7C       ACK       ¬        |
5: 111 1101 175 125  7D                 }        }
6: 111 1110 176 126  7E       ESC       |        ~
```

2. Pandas for Data Analysis

(a) Choose a dataset. Give a summary of the data.

In [2]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

matplotlib.style.use('ggplot')

df=pd.read_csv('boston.csv', header=0, index_col=0)

# part (a) summarize data
df.shape
```

Out[2]:
```
(506, 14)
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 506 entries, 1 to 506
Data columns (total 14 columns):
crim       506 non-null float64
zn         506 non-null float64
indus      506 non-null float64
chas       506 non-null int64
nox        506 non-null float64
rm         506 non-null float64
age        506 non-null float64
dis        506 non-null float64
rad        506 non-null int64
tax        506 non-null int64
ptratio    506 non-null float64
black      506 non-null float64
lstat      506 non-null float64
medv       506 non-null float64
dtypes: float64(11), int64(3)
memory usage: 59.3 KB
```

In [4]:
```
df.describe()
```

Out[4]:

	crim	zn	indus	chas	nox	rm	age
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.00
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.9000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500
75%	3.677082	12.500000	18.100000	0.000000	0.624000	6.623500	94.075
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.00

In [5]:
```
df.head()
```

Out[5]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	r
1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	2
2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	2
3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	3
4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	3
5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	3

In [6]:
```
df.tail(3)
```

Out[6]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat
504	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64
505	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48
506	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88

(b) Name your favorite and provide the link

Greg Reda has a nice three part tutorial that deals with dataframe using Pandas. It's written in jupyter notebook style. http://www.gregreda.com/2013/10/26/intro-to-pandas-data-structures/ (http://www.gregreda.com/2013/10/26/intro-to-pandas-data-structures/)

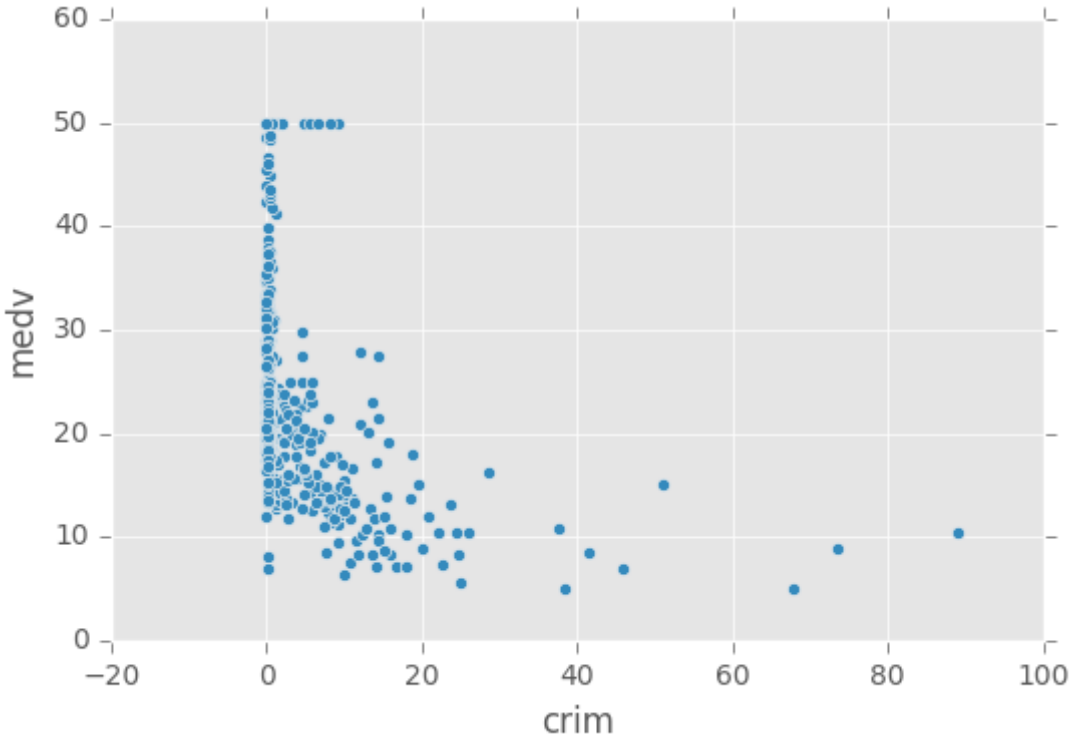
Pandas' official document is also very handy. http://pandas.pydata.org/pandas-docs/stable/10min.html (http://pandas.pydata.org/pandas-docs/stable/10min.html)

(c) Do a simple data analysis, including plots, a regression model, with parameter estimates

In [7]:
```
#df.plot.scatter(x='crim', y='medv')
```

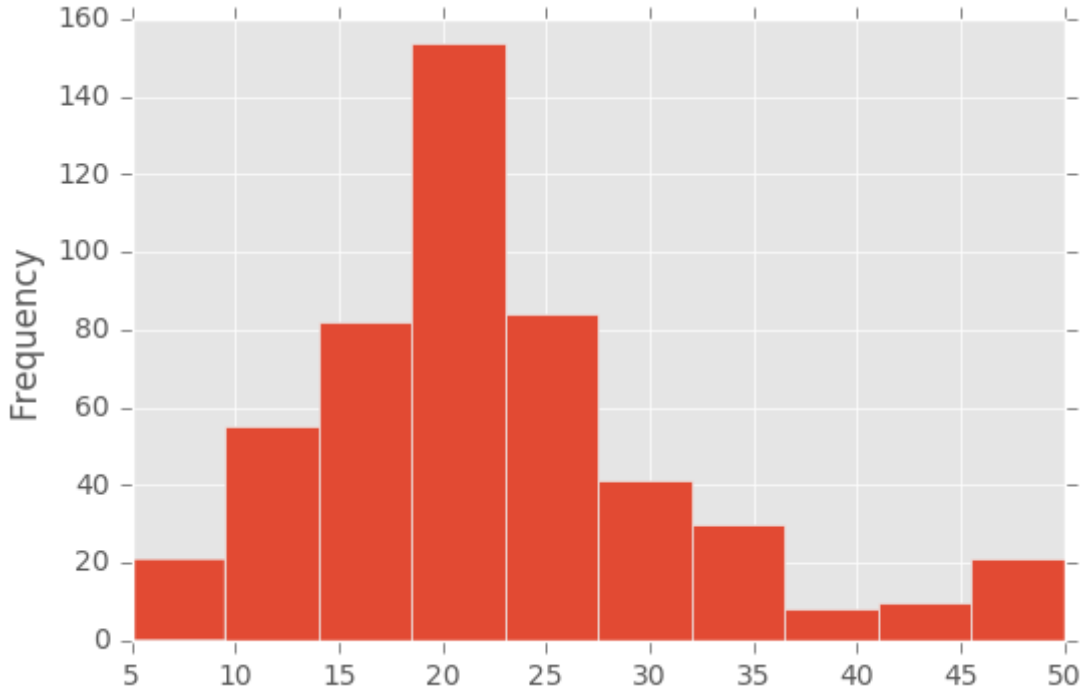
In [8]:
```
df.plot(kind='scatter', x='crim', y='medv')
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1efcafa4ba8>



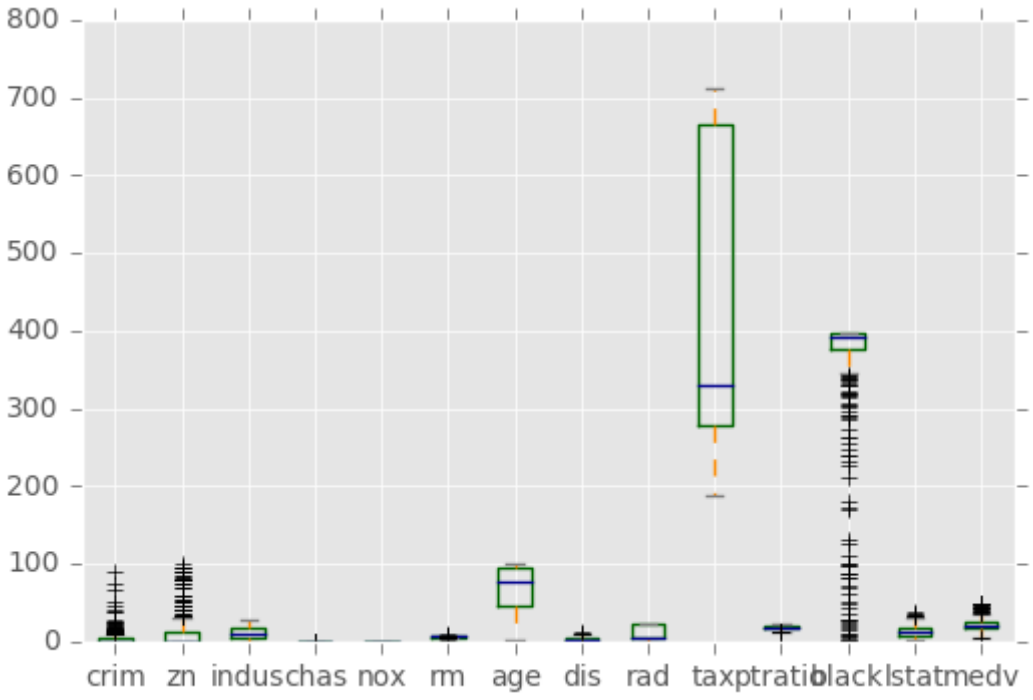
In [9]:
```
# histogram
df['medv'].plot.hist(orientation='vertical', cumulative=False)
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1efcaf459b0>



In [10]:
```
# box plot
color = dict(boxes='DarkGreen', whiskers='DarkOrange', medians='DarkBlue', cap
s='Gray')
df.plot.box(color=color, sym='r+')
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1efcb92da90>



In [11]:
```
# regression
import statsmodels.formula.api as smf
X="+".join(df.columns.difference(['medv']))
result=smf.ols(formula="medv ~"+X, data=df).fit()
```

In [12]:
```
print(result.params)
```

```
Intercept    36.459488
age           0.000692
black         0.009312
chas          2.686734
crim         -0.108011
dis          -1.475567
indus         0.020559
lstat        -0.524758
nox         -17.766611
ptratio      -0.952747
rad           0.306049
rm            3.809865
tax          -0.012335
zn            0.046420
dtype: float64
```

In [13]:
```
print(result.summary())
```

OLS Regression Results

Dep. Variable:	medv	R-squared:	0.741
Model:	OLS	Adj. R-squared:	0.734
Method:	Least Squares	F-statistic:	108.1
Date:	Mon, 28 Nov 2016	Prob (F-statistic):	6.72e-135
Time:	21:54:07	Log-Likelihood:	-1498.8
No. Observations:	506	AIC:	3026.
Df Residuals:	492	BIC:	3085.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
Intercept	36.4595	5.103	7.144	0.000	26.432	46.487
age	0.0007	0.013	0.052	0.958	-0.025	0.027
black	0.0093	0.003	3.467	0.001	0.004	0.015
chas	2.6867	0.862	3.118	0.002	0.994	4.380
crim	-0.1080	0.033	-3.287	0.001	-0.173	-0.043
dis	-1.4756	0.199	-7.398	0.000	-1.867	-1.084
indus	0.0206	0.061	0.334	0.738	-0.100	0.141
lstat	-0.5248	0.051	-10.347	0.000	-0.624	-0.425
nox	-17.7666	3.820	-4.651	0.000	-25.272	-10.262
ptratio	-0.9527	0.131	-7.283	0.000	-1.210	-0.696
rad	0.3060	0.066	4.613	0.000	0.176	0.436
rm	3.8099	0.418	9.116	0.000	2.989	4.631
tax	-0.0123	0.004	-3.280	0.001	-0.020	-0.005
zn	0.0464	0.014	3.382	0.001	0.019	0.073

```
=
Omnibus:                      178.041   Durbin-Watson:                   1.07
8
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              783.12
6
Skew:                           1.521   Prob(JB):                     8.84e-17
1
Kurtosis:                       8.281   Cond. No.                      1.51e+0
4
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 1.51e+04. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

(d) Provide a 95% confidence interval to one of the regression coefficients

For the variable 'crim', the 95% confidence interval is [-0.173, -0.043]

(e) Comment on pros and cons of using Pandas vs R.

For data analysis that entails exploration and visualization, I think Python Pandas and R are both very powerful, and the difference is not great at all. Most of the work can be readily done in both. In terms of summarizing the data, R has built in function 'summary' while python has describe() method. When it comes to linear regression, R has small functions that are built in while python relies on packages one has to import first. The result in python comes in as more integrated than R though. For now, R has more statistical support in general than Python. For data manipulation, I personlly prefer R data.table which is both intuitive and fast.

3. Text Analysis Using Python

(a) Show code for downloading

In [14]:
```
# Part (a) Download the file to local disc
import urllib.request
from urllib.request import urlretrieve
url='http://www.stat.uchicago.edu/~meiwang/courses/pg18.txt'
urlretrieve(url, "federal.txt")
```

Out[14]:
```
('federal.txt', <http.client.HTTPMessage at 0x1efcda4cf28>)
```

(b) Extract all title lines along with line numbers. Show code and (partial) output.

In [15]:
```
# Part (b) Read in titles
import re
with open('federal.txt','r') as fh:
        count=1
        dic={}
        for line in fh:
                line=line.rstrip()
                if re.search('^FEDERALIST.*?No\.\s[0-9]*', line):
                        #print(line)
                        dic[count]={'Title': line,
                                                'Number': re.findall('^FEDERAL
IST.*?No\.\s([0-9]*)', line)}
                count+=1

sorted(dic.items())[:10]
```

Out[15]:
```
[(43, {'Number': ['1'], 'Title': 'FEDERALIST. No. 1'}),
 (225, {'Number': ['2'], 'Title': 'FEDERALIST No. 2'}),
 (415, {'Number': ['3'], 'Title': 'FEDERALIST No. 3'}),
 (592, {'Number': ['4'], 'Title': 'FEDERALIST No. 4'}),
 (788, {'Number': ['5'], 'Title': 'FEDERALIST No. 5'}),
 (956, {'Number': ['6'], 'Title': 'FEDERALIST No. 6'}),
 (1217, {'Number': ['7'], 'Title': 'FEDERALIST. No. 7'}),
 (1474, {'Number': ['8'], 'Title': 'FEDERALIST No. 8'}),
 (1709, {'Number': ['9'], 'Title': 'FEDERALIST No. 9'}),
 (1940, {'Number': ['10'], 'Title': 'FEDERALIST No. 10'})]
```

(c) How many articles are in the ebook file? Are there any duplicates?

In [16]:
```
# Part (c)
# How many articles
print('There are {0} articles in the ebook file, including possible duplicate
s'.format(len(dic)))
# Are there duplicates? Yes!
len(dic) is len(set([dic[a]['Number'][0] for a in dic.keys()]))
```

```
There are 86 articles in the ebook file, including possible duplicates
```

Out[16]:
```
False
```

There are indeed duplicates

(d) How many single authored papers were written by each author? Show code and obtain the counts.

In [17]:
```
# Part (d)
# full text is preceded by 'To the People of the State of New York:'
# 'PUBLIUS': exception No. 37 no 'publius'

# Add an "PUBLIUS" identifier at the end of No. 37, tidy format
import fileinput
import re

for line in fileinput.FileInput('federal.txt',inplace=1):
        if re.search('.+PUBLIUS\.', line):
                line=line.replace(line, re.findall('(.+)PUBLIUS\.', line)
[0]+"\n\nPUBLIUS.")
                print(line, )
        elif "diminished by delays or by new experiments." in line:
                line=line.replace(line, line + "\n\nPUBLIUS.")
                print(line, )
        else:
                print(line, end='')
```

In [18]:
```
# Obtain the full dictionary
with open('federal.txt','r') as fh:
        count=1
        dic={}
        for line in fh:
                line=line.rstrip()
                if re.search('^FEDERALIST.*?No\.\s[0-9]*', line):
                        #print(line)
                        dic[count]={'Title': line,
                                                'Number': re.findall('^FEDERAL
IST.*?No\.\s([0-9]*)', line)}
                        next_line=fh.readline().rstrip()
                        while next_line not in ['HAMILTON', 'JAY', 'MADISON',
'HAMILTON AND MADISON', 'HAMILTON OR MADISON']:
                                next_line=fh.readline().rstrip()
                        dic[count]['Author']=next_line

                        next_line=fh.readline().rstrip()
                        while not re.search('^To the People of the State of Ne
w York',next_line):
                                next_line=fh.readline().rstrip()
                        fh.readline()
                        next_line=fh.readline().rstrip()

                        text_lst=[]
                        while next_line!='PUBLIUS.':
                                text_lst.append(next_line)
                                next_line=fh.readline().rstrip()
                        dic[count]['FullText']=text_lst

                count+=1

sorted(dic.items())[0]
```

Out[18]:

```
(43,
 {'Author': 'HAMILTON',
  'FullText': ['AFTER an unequivocal experience of the inefficacy of the',
   'subsisting federal government, you are called upon to deliberate on',
   'a new Constitution for the United States of America. The subject',
   'speaks its own importance; comprehending in its consequences',
   'nothing less than the existence of the UNION, the safety and welfare',
   'of the parts of which it is composed, the fate of an empire in many',
   'respects the most interesting in the world. It has been frequently',
   'remarked that it seems to have been reserved to the people of this',
   'country, by their conduct and example, to decide the important',
   'question, whether societies of men are really capable or not of',
   'establishing good government from reflection and choice, or whether',
   'they are forever destined to depend for their political',
   'constitutions on accident and force. If there be any truth in the',
   'remark, the crisis at which we are arrived may with propriety be',
   'regarded as the era in which that decision is to be made; and a',
   'wrong election of the part we shall act may, in this view, deserve',
   'to be considered as the general misfortune of mankind.',
   '',
   'This idea will add the inducements of philanthropy to those of',
   'patriotism, to heighten the solicitude which all considerate and',
   'good men must feel for the event. Happy will it be if our choice',
   'should be directed by a judicious estimate of our true interests,',
   'unperplexed and unbiased by considerations not connected with the',
   'public good. But this is a thing more ardently to be wished than',
   'seriously to be expected. The plan offered to our deliberations',
   'affects too many particular interests, innovates upon too many local',
   'institutions, not to involve in its discussion a variety of objects',
   'foreign to its merits, and of views, passions and prejudices little',
   'favorable to the discovery of truth.',
   '',
   'Among the most formidable of the obstacles which the new',
   'Constitution will have to encounter may readily be distinguished the',
   'obvious interest of a certain class of men in every State to resist',
   'all changes which may hazard a diminution of the power, emolument,',
   'and consequence of the offices they hold under the State',
   'establishments; and the perverted ambition of another class of men,',
   'who will either hope to aggrandize themselves by the confusions of',
   'their country, or will flatter themselves with fairer prospects of',
   'elevation from the subdivision of the empire into several partial',
   'confederacies than from its union under one government.',
   '',
   'It is not, however, my design to dwell upon observations of this',
   'nature. I am well aware that it would be disingenuous to resolve',
   'indiscriminately the opposition of any set of men (merely because',
   'their situations might subject them to suspicion) into interested or',
   'ambitious views. Candor will oblige us to admit that even such men',
   'may be actuated by upright intentions; and it cannot be doubted',
   'that much of the opposition which has made its appearance, or may',
   'hereafter make its appearance, will spring from sources, blameless',
   'at least, if not respectable--the honest errors of minds led astray',
   'by preconceived jealousies and fears. So numerous indeed and so',
   'powerful are the causes which serve to give a false bias to the',
   'judgment, that we, upon many occasions, see wise and good men on the',
   'wrong as well as on the right side of questions of the first',
   'magnitude to society. This circumstance, if duly attended to, would',
```

```
'furnish a lesson of moderation to those who are ever so much',
'persuaded of their being in the right in any controversy. And a',
'further reason for caution, in this respect, might be drawn from the',
'reflection that we are not always sure that those who advocate the',
'truth are influenced by purer principles than their antagonists.',
'Ambition, avarice, personal animosity, party opposition, and many',
'other motives not more laudable than these, are apt to operate as',
'well upon those who support as those who oppose the right side of a',
'question. Were there not even these inducements to moderation,',
'nothing could be more ill-judged than that intolerant spirit which',
'has, at all times, characterized political parties. For in',
'politics, as in religion, it is equally absurd to aim at making',
'proselytes by fire and sword. Heresies in either can rarely be',
'cured by persecution.',
'',
'And yet, however just these sentiments will be allowed to be, we',
'have already sufficient indications that it will happen in this as',
'in all former cases of great national discussion. A torrent of',
'angry and malignant passions will be let loose. To judge from the',
'conduct of the opposite parties, we shall be led to conclude that',
'they will mutually hope to evince the justness of their opinions,',
'and to increase the number of their converts by the loudness of',
'their declamations and the bitterness of their invectives. An',
'enlightened zeal for the energy and efficiency of government will be',
'stigmatized as the offspring of a temper fond of despotic power and',
'hostile to the principles of liberty. An over-scrupulous jealousy',
'of danger to the rights of the people, which is more commonly the',
'fault of the head than of the heart, will be represented as mere',
'pretense and artifice, the stale bait for popularity at the expense',
'of the public good. It will be forgotten, on the one hand, that',
'jealousy is the usual concomitant of love, and that the noble',
'enthusiasm of liberty is apt to be infected with a spirit of narrow',
'and illiberal distrust. On the other hand, it will be equally',
'forgotten that the vigor of government is essential to the security',
'of liberty; that, in the contemplation of a sound and well-informed',
'judgment, their interest can never be separated; and that a',
'dangerous ambition more often lurks behind the specious mask of zeal',
'for the rights of the people than under the forbidden appearance of',
'zeal for the firmness and efficiency of government. History will',
'teach us that the former has been found a much more certain road to',
'the introduction of despotism than the latter, and that of those men',
'who have overturned the liberties of republics, the greatest number',
'have begun their career by paying an obsequious court to the people;',
'commencing demagogues, and ending tyrants.',
'',
'In the course of the preceding observations, I have had an eye,',
'my fellow-citizens, to putting you upon your guard against all',
'attempts, from whatever quarter, to influence your decision in a',
'matter of the utmost moment to your welfare, by any impressions',
'other than those which may result from the evidence of truth. You',
'will, no doubt, at the same time, have collected from the general',
'scope of them, that they proceed from a source not unfriendly to the',
'new Constitution. Yes, my countrymen, I own to you that, after',
'having given it an attentive consideration, I am clearly of opinion',
'it is your interest to adopt it. I am convinced that this is the',
'safest course for your liberty, your dignity, and your happiness. I',
'affect not reserves which I do not feel. I will not amuse you with',
```

```
 'an appearance of deliberation when I have decided. I frankly',
 'acknowledge to you my convictions, and I will freely lay before you',
 'the reasons on which they are founded. The consciousness of good',
 'intentions disdains ambiguity. I shall not, however, multiply',
 'professions on this head. My motives must remain in the depository',
 'of my own breast. My arguments will be open to all, and may be',
 'judged of by all. They shall at least be offered in a spirit which',
 'will not disgrace the cause of truth.',
 '',
 'I propose, in a series of papers, to discuss the following',
 'interesting particulars:',
 '',
 'THE UTILITY OF THE UNION TO YOUR POLITICAL PROSPERITY',
 '',
 'THE INSUFFICIENCY OF THE PRESENT CONFEDERATION',
 'TO PRESERVE THAT UNION  THE NECESSITY OF A GOVERNMENT AT LEAST',
 'EQUALLY ENERGETIC WITH THE ONE PROPOSED, TO THE ATTAINMENT OF THIS',
 'OBJECT  THE CONFORMITY OF THE PROPOSED CONSTITUTION TO THE TRUE',
 'PRINCIPLES OF REPUBLICAN GOVERNMENT',
 'ITS ANALOGY TO YOUR OWN STATE CONSTITUTION',
 'and lastly, THE ADDITIONAL SECURITY WHICH ITS',
 'ADOPTION WILL AFFORD TO THE PRESERVATION OF THAT SPECIES OF',
 'GOVERNMENT, TO LIBERTY, AND TO PROPERTY.',
 '',
 'In the progress of this discussion I shall endeavor to give a',
 'satisfactory answer to all the objections which shall have made',
 'their appearance, that may seem to have any claim to your attention.',
 '',
 'It may perhaps be thought superfluous to offer arguments to',
 'prove the utility of the UNION, a point, no doubt, deeply engraved',
 'on the hearts of the great body of the people in every State, and',
 'one, which it may be imagined, has no adversaries. But the fact is,',
 'that we already hear it whispered in the private circles of those',
 'who oppose the new Constitution, that the thirteen States are of too',
 'great extent for any general system, and that we must of necessity',
 'resort to separate confederacies of distinct portions of the',
 'whole.1 This doctrine will, in all probability, be gradually',
 'propagated, till it has votaries enough to countenance an open',
 'avowal of it. For nothing can be more evident, to those who are',
 'able to take an enlarged view of the subject, than the alternative',
 'of an adoption of the new Constitution or a dismemberment of the',
 'Union. It will therefore be of use to begin by examining the',
 'advantages of that Union, the certain evils, and the probable',
 'dangers, to which every State will be exposed from its dissolution.',
 'This shall accordingly constitute the subject of my next address.',
 '',
 ''],
'Number': ['1'],
'Title': 'FEDERALIST. No. 1'})
```

In [19]:
```
# tidy the fulltext
def TidyText(textlst):
        textlst=[a for a in textlst if a is not '']
        text=' '.join(textlst)
        return text

def TidyDic(dic):
        for i in dic.keys():
                dic[i]['FullText']=TidyText(dic[i]['FullText'])
        # remove duplicates [595, 612]
        Index=[i for i in dic.keys() if dic[i]['Number'][0]=='70']
        print('Article No. 70 has {0} duplicates at line {1}'.format(len(Index
ndex))
        # : kill the second one line 612
        dic.pop(max(Index), None)

TidyDic(dic)
```

```
Article No. 70 has 2 duplicates at line [595, 612]
```

In [20]:
```
import json
with open('data.json','w') as f:
    json.dump(dic,f,indent=4,sort_keys=True)
```

In [21]:
```
def count_single_authored(dic):
        Hamilton={i:dic[i] for i in dic.keys() if dic[i]
['Author']=='HAMILTON'}
        Madison={i:dic[i] for i in dic.keys() if dic[i]['Author']=='MADISON'}
        return len(Hamilton), len(Madison)

a, b=count_single_authored(dic)
print('{0} papers are single authored by Hamilton\n{1} papers are single autho
red by Madison'.format(a,b))
```

```
51 papers are single authored by Hamilton
15 papers are single authored by Madison
```

(e) Who wrote the longest articles on average? Show code and averages.

In [22]:
```
# Part (e) who wrote longer articles on average
def clean_text(rawtext):
        word_lst=rawtext.split()
        new=[]
        for i in word_lst:
                if re.search('.*[,\.:;\?]',i):
                        i=re.findall('(.*)[,\.:;\?]',i)[0]
                i=i.lower()
                new.append(i)
        return new

import numpy as np
def count_length(dic):
        Hamilton={int(dic[i]['Number'][0]):len(clean_text(dic[i]['FullText']))
 for i in dic.keys() if dic[i]['Author']=='HAMILTON'}
        Madison={int(dic[i]['Number'][0]):len(clean_text(dic[i]['FullText']))
for i in dic.keys() if dic[i]['Author']=='MADISON'}
        return np.mean(list(Hamilton.values())),
np.mean(list(Madison.values()))

a, b=count_length(dic)
print('Average length of articles authored by Halmilton is {0}\nAverage length
 of articles authored by Madison is {1}'.format(a,b))
```

```
Average length of articles authored by Halmilton is 2173.8627450980393
Average length of articles authored by Madison is 2721.9333333333334
```

Madison writes longer articles on average.

(f) Show code of counting and create the table of word counts.

In [23]:
```
# Part (f) Do word count
words=['by', 'enough', 'from', 'this', 'that', 'to', 'upon', 'while',
'whilst']
def word_count(bag_of_words):
        counts={}
        for i in bag_of_words:
                if i in words:
                        counts[i]=counts.get(i,0) + 1
        return counts

def count_words_all(dic):
        bag_of_words_H=[]
        bag_of_words_M=[]
        for k,v in dic.items():
                if dic[k]['Author']=='HAMILTON':
                        bag_of_words_H+=clean_text(v['FullText'])
                elif dic[k]['Author']=='MADISON':
                        bag_of_words_M+=clean_text(v['FullText'])
        count_H=word_count(bag_of_words_H)
        count_M=word_count(bag_of_words_M)
        return {'Hamilton': count_H, 'Madison': count_M}

count_dic=count_words_all(dic)
```

In [24]:
```
import pandas as pd
df=pd.DataFrame(count_dic)
df=df.T
df=df.fillna(0)
df=df.astype(int)
df['total']=df.sum(axis=1)
observed=df.ix[:,0:9]
df
```

Out[24]:

	by	enough	from	that	this	to	upon	while	whilst	total
Hamilton	847	32	618	1678	904	4467	372	36	1	8955
Madison	474	0	201	567	259	1296	7	0	12	2816

(g) Chi-square test to compare the frequencies of the words used by the two authors. What are the hypotheses? What is your conclusion?

In [25]:
```
# Part (g) Chi-square test of independence
import scipy.stats as stats
stats.chi2_contingency(observed)
```

Out[25]:
```
(270.7059036717892,
 6.9642235696186058e-54,
 8,
 array([[  1.00497451e+03,   2.43445757e+01,   6.23068983e+02,
           1.70792414e+03,   8.84773171e+02,   4.38430592e+03,
           2.88331068e+02,   2.73876476e+01,   9.88998386e+00],
        [  3.16025486e+02,   7.65542435e+00,   1.95931017e+02,
           5.37075864e+02,   2.78226829e+02,   1.37869408e+03,
           9.06689321e+01,   8.61235239e+00,   3.11001614e+00]]))
```

The null hypothesis is that word frequencies are independent of authors. The Chi-square test is essentially a test of independence. The test statistic is 270.706 and the p value is 6.964e-54. Hence we reject the null hypothsis and conclude that the word frequencies are dependent on authorship. This is useful because we can determine authorship of an ambiguous document by looking at the word frequency counts.

In [ ]: