Create list of data science URLs

In [2]:
```
# List of articles about data science
urls = ["http://en.wikipedia.org/wiki/Data_science",
        "http://www.zdnet.com/big-data-what-to-trust-data-science-or-the-boss
s-sixth-sense-7000026550/",
        "http://www.datascienceassn.org/content/tachyon-supercharges-spark",
        "https://www.coursera.org/course/datasci",
        "http://www-01.ibm.com/software/data/infosphere/data-scientist/",
        "http://radar.oreilly.com/2010/06/what-is-data-science.html",
        "http://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-ce
ntury/ar/1",
        "http://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history
-of-data-science/",
        "http://cs109.org/",
        "https://education.emc.com/guest/campaign/data_science.aspx",
        "https://www.udacity.com/course/ud359",
        "http://www.forbes.com/sites/gilpress/2012/09/27/data-scientists-the-
definition-of-sexy/",
        "http://gigaom.com/2013/04/16/how-to-hire-data-scientists-and-get-hir
ed-as-one/",
        "http://blogs.hbr.org/2013/03/a-data-scientists-real-job-sto/",
        "http://www.theguardian.com/technology/2014/jan/27/why-data-science-m
atters-to-foursquare"]
```

Pull down the HTML code from the URLs

In [3]:
```
# Pull in HTML from a news article
from urllib import urlopen
pages_html = []
for i in range(0,len(urls)):
    pages_html.append(urlopen(urls[i]).read())
```

In [4]:
```
# HTML code with text surrounded by tags, code, etc.
pages_html[0][1:1000]
```

Out[4]:
```
'!DOCTYPE html>\n<html lang="en" dir="ltr" class="client-nojs">\n<head>\n<met
a charset="UTF-8" />\n<title>Data science - Wikipedia, the free encyclopedia<
/title>\n<meta http-equiv="X-UA-Compatible" content="IE=EDGE" />\n<meta name=
"generator" content="MediaWiki 1.23wmf14" />\n<link rel="alternate" type="app
lication/x-wiki" title="Edit this page" href="/w/index.php?title=Data_science
&amp;action=edit" />\n<link rel="edit" title="Edit this page" href="/w/index.
```

```
php?title=Data_science&amp;action=edit" />\n<link rel="apple-touch-icon" href
="//bits.wikimedia.org/apple-touch/wikipedia.png" />\n<link rel="shortcut ico
n" href="//bits.wikimedia.org/favicon/wikipedia.ico" />\n<link rel="search" t
ype="application/opensearchdescription+xml" href="/w/opensearch_desc.php" tit
le="Wikipedia (en)" />\n<link rel="EditURI" type="application/rsd+xml" href="
//en.wikipedia.org/w/api.php?action=rsd" />\n<link rel="copyright" href="//cr
eativecommons.org/licenses/by-sa/3.0/" />\n<link rel="alternate" type="applic
ation/atom+xm'
```

Extract text content from HTML code

In [6]:
```
# Import nltk (natural language toolkit)
import nltk

# Remove the tags, code from the HTML, leaving just text
pages_raw = []
for i in range(0, len(pages_html)):
    pages_raw.append(nltk.clean_html(pages_html[i]))
```

In [7]:
```
# Cleaned text
pages_raw[0][1:1000]
```

Out[7]:
```
"ata science - Wikipedia, the free encyclopedia \n \n \n \n \n \n \n \n \n \n
 \n \n \n \n \n\n\n\n\n\n \n \n\t\t \n\t\t \n\t\t \n\t\t\t \n\t\t\t \n\t\t\t\
t\t\t \n\t\t\t\t\t\t\t Data science \n\t\t\t \n\t\t\t\t\t\t\t\t\t From Wikipedia,
 the free encyclopedia \n\t\t\t\t\t\t\t\t \n\t\t\t\t\t\t\t\t\t\t\t\t \n\t\t\t
\t\tJump to:\t\t\t\t\t navigation , \t\t\t\t\t search \n\t\t\t\t \n\t\t\t\t\t \
n \n \n \n \n This article's tone or style may not reflect the encyclopedic t
one used on Wikipedia . See Wikipedia's guide to writing better articles for
suggestions. (February 2014) \n \n \n \n \n \n  \nData Science \n \n \n Data
science is the study of the generalizable extraction of knowledge from data ,
 [ 1 ] yet the key word is science . [ 2 ] It incorporates varying elements a
nd builds on techniques and theories from many fields, including signal proce
ssing , mathematics , probability models , machine learning , computer progra
mming , statistics , data engineering , pattern recognition and learning , vi
sualization , uncertainty modeling , data warehousing , and high performance
computing with the goal of extracting meaning from"
```

Tokenize text

In [8]:
```
# Tokenize the news article text
pages_tokenized = []
```

```
for i in range(0, len(pages_html)):
    pages_tokenized.append(nltk.word_tokenize(pages_raw[i]))
```

In [10]:
```
# Tokenized text. Each word and punctuation mark is a different "token"
pages_tokenized[0][1:100]
```

Out[10]:
```
['science',
 '-',
 'Wikipedia',
 ',',
 'the',
 'free',
 'encyclopedia',
 'Data',
 'science',
 'From',
 'Wikipedia',
 ',',
 'the',
 'free',
 'encyclopedia',
 'Jump',
 'to',
 ':',
 'navigation',
 ',',
 'search',
 'This',
 'article',
 "'s",
 'tone',
 'or',
 'style',
 'may',
 'not',
 'reflect',
 'the',
 'encyclopedic',
 'tone',
 'used',
 'on',
 'Wikipedia',
 '.',
 'See',
 'Wikipedia',
 "'s",
 'guide',
```

```
'to',
'writing',
'better',
'articles',
'for',
'suggestions.',
'(',
'February',
'2014',
')',
'Data',
'Science',
'Data',
'science',
'is',
'the',
'study',
'of',
'the',
'generalizable',
'extraction',
'of',
'knowledge',
'from',
'data',
',',
'[',
'1',
']',
'yet',
'the',
'key',
'word',
'is',
'science',
'.',
'[',
'2',
']',
'It',
'incorporates',
'varying',
'elements',
'and',
'builds',
'on',
'techniques',
```

```
 'and',
 'theories',
 'from',
 'many',
 'fields',
 ',',
 'including',
 'signal',
 'processing',
 ',',
 'mathematics']
```

Clean text

In [11]:
```
# Import stopwords "corpus". Stopwords are very common words that are less us
eful for distinguishing documents, finding meaning.
from nltk.corpus import stopwords
stopwords = nltk.corpus.stopwords.words('english')

# Remove punctuation, change to lowercase, and remove stopwords. Use a "list
comprehension"
pages_tokenized_clean = []
for i in range(0, len(pages_tokenized)):
    pages_tokenized_clean.append([w.lower() for w in pages_tokenized[i] if w.
isalpha() and w.lower() not in stopwords])
```

In [12]:
```
pages_tokenized_clean[0][1:100]
```

Out[12]:
```
['science',
 'wikipedia',
 'free',
 'encyclopedia',
 'data',
 'science',
 'wikipedia',
 'free',
 'encyclopedia',
 'jump',
 'navigation',
 'search',
 'article',
 'tone',
 'style',
 'may',
 'reflect',
```

```
'encyclopedic',
'tone',
'used',
'wikipedia',
'see',
'wikipedia',
'guide',
'writing',
'better',
'articles',
'february',
'data',
'science',
'data',
'science',
'study',
'generalizable',
'extraction',
'knowledge',
'data',
'yet',
'key',
'word',
'science',
'incorporates',
'varying',
'elements',
'builds',
'techniques',
'theories',
'many',
'fields',
'including',
'signal',
'processing',
'mathematics',
'probability',
'models',
'machine',
'learning',
'computer',
'programming',
'statistics',
'data',
'engineering',
'pattern',
'recognition',
```

```
 'learning',
 'visualization',
 'uncertainty',
 'modeling',
 'data',
 'warehousing',
 'high',
 'performance',
 'computing',
 'goal',
 'extracting',
 'meaning',
 'data',
 'creating',
 'data',
 'data',
 'science',
 'buzzword',
 'often',
 'used',
 'interchangeably',
 'analytics',
 'big',
 'data',
 'often',
 'abused',
 'marketing',
 'anything',
 'involving',
 'data',
 'processing',
 'particular',
 'existing',
 'competitive',
 'intelligence']
```

Merge documents together

In [13]:
```
# Merge all pages together
pages_merged = []
for i in range(1,len(pages_tokenized_clean)):
    pages_merged = pages_merged + pages_tokenized_clean[i]
```

In [14]:
```
pages_merged[1:100]
```

Out[14]:
```
['data',
 'trust',
 'data',
 'science',
 'boss',
 'sixth',
 'sense',
 'zdnet',
 'zdnet',
 'log',
 'join',
 'zdnet',
 'home',
 'white',
 'papers',
 'hot',
 'topics',
 'downloads',
 'reviews',
 'newsletters',
 'us',
 'edition',
 'available',
 'following',
 'editions',
 'asia',
 'australia',
 'europe',
 'india',
 'united',
 'kingdom',
 'united',
 'states',
 'zdnet',
 'around',
 'globe',
 'zdnet',
 'belgium',
 'zdnet',
 'china',
 'zdnet',
 'france',
 'zdnet',
 'germany',
 'zdnet',
 'korea',
 'zdnet',
```

```
'japan',
'zdnet',
'netherlands',
'topics',
'research',
'mwc',
'windows',
'xp',
'internet',
'things',
'security',
'byod',
'cxo',
'apple',
'writers',
'log',
'log',
'join',
'zdnet',
'read',
'five',
'things',
'need',
'know',
'mobile',
'middle',
'east',
'topic',
'big',
'data',
'discover',
'follow',
'via',
'rss',
'email',
'alert',
'big',
'data',
'trust',
'data',
'science',
'boss',
'sixth',
'sense',
'summary',
'technology',
'run',
```

```
 'projects',
 'may',
 'opening',
 'firms',
 'progress']
```

Basic analysis on word counts

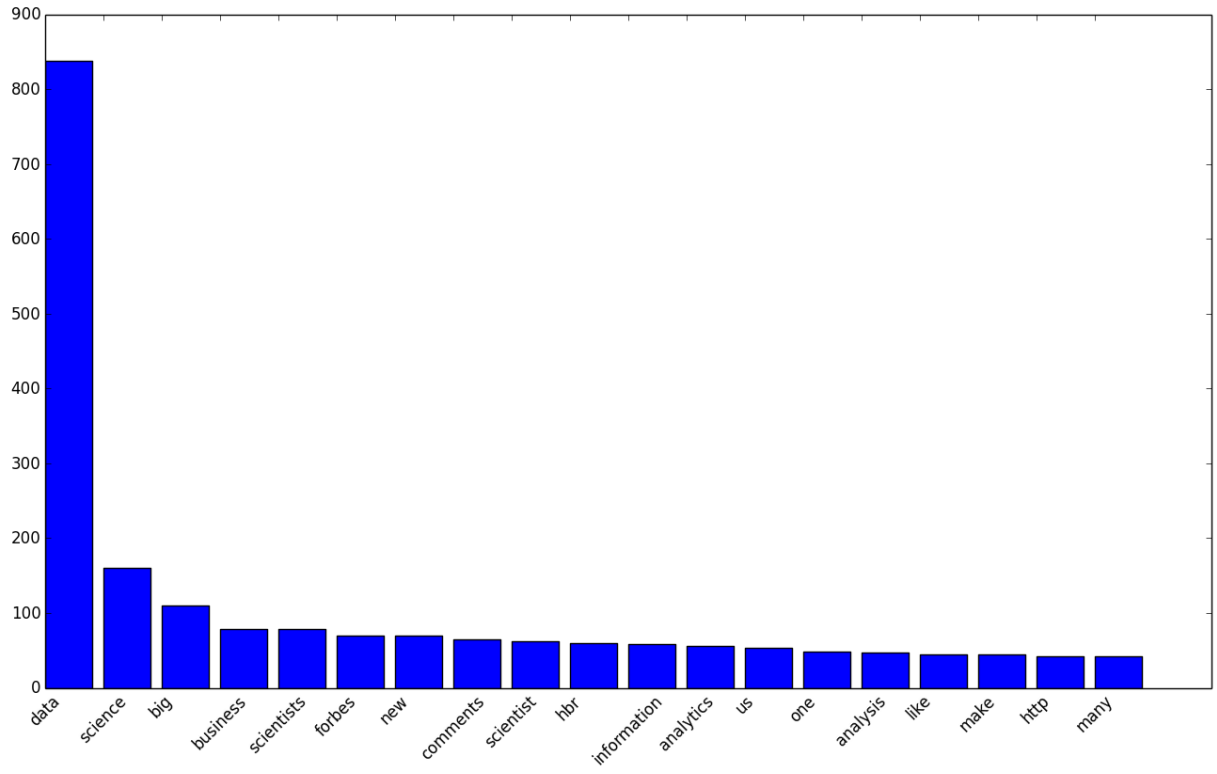
In [15]:
```
# Create frequency distribution of words in articles
freqdist = nltk.FreqDist(w for w in pages_merged)
```

In [16]:
```
#%pylab inline

# Plot frequency distribution
import matplotlib.pyplot as plt
topWords = freqdist.keys()[0:19]
topValues = freqdist.values()[0:19]

plt.bar(range(len(topWords)), topValues)
plt.xticks(range(len(topWords)), topWords, rotation = 45)

plt.show()
```



Overall word counts

In [17]:
```
# How many different words are used in the articles?
len(set(pages_merged))
```

Out[17]:
```
4454
```

- Did we overcount? What about words that are really different versions of the same word, like "analyze" and "analyzing"?
- Use a process called stemming, with nltk.stem

Exploration with nltk

In [18]:
```
# Find occurences of a given word:
# First, convert document to nltk Text object
pages_text = nltk.Text(pages_merged)
# Find words that are unusually close to one another frequently
pages_text.collocations()
```

```
Building collocations list
data science; big data; harvard business; data scientists; data
scientist; machine learning; gil press; permalink flag; flag reply;
december november; november october; summit forbes; months ago; august
july; october september; privacy policy; january december; september
august; business school; july june
```

In []:
```

```