Leveraging Big Data in Introductory Programming Courses

David Reed
Department of Journalism, Media & Computing
Creighton University
Big Data is big!

working definition: the storage & processing of “massive” amounts of data in order to extract useful information

it’s all over the news
- NSA phone & email surveillance
- DHS mining social media
- Google & Facebook collecting and studying online behavior
- CS Principles course includes using “large data sets to explore and discover information and knowledge”

it’s driving the demand for faster computers, expanded storage media, innovative algorithmic approaches and capable technology specialists
Big Data is motivating

- generally accepted that students (especially girls) are motivated by real-world problems
  - even an intro programming level, real-world data can make programming more interesting & relevant

- unfortunately, finding *accessible* data sets is not easy
  - often stored in compressed or proprietary formats
  - may require special tools for data access & viewing

- while specialized formats & tools may be justified at an advanced level, beginning students need transparency
Three examples

this session will present three Big Data examples

- used in introductory programming classes
- students programmed in Python, but any language would do
- the data sets enable students to write simple programs to answer interesting questions

the data sets vary in size and complexity

- ENABLE online wordlist (1 MB)
- NBA/WNBA player stats (5-10 MB)
- Twitter hashtags from 2006-2009 (1.8 GB)
ENABLE (Enhanced North American Benchmark Lexicon) is a public domain list of 172,823 words

- often used for word games such as Scrabble™, Boggle™, ...

when teaching lists/arrays in intro course, can make simple tasks more interesting

- old school: how many grades in a list > 90?
- new school: how many 4-letter words in English? longest & most common word lengths? first and last words (alphabetically)? how many anagrams of stale?
Python’s built-in list-processing make it easy to pose and answer interesting questions (as do other languages)

```python
>>> len([word for word in wordlist if len(word) == 4])
3857
>>> max([len(word) for word in wordlist])
21
>>> counts = 22*[0]
>>> for word in wordlist:
    counts[len(word)] += 1

>>> counts.index(max(counts))
8
>>> sorted(wordlist)[0]
'a'
>>> sorted(wordlist)[-1]
'zyzzyvas'

>>> len([word for word in wordlist if sorted(word) == sorted("stale")])
10
```
Sports stats

http://www.nba.com/statistics/
http://www.wnba.com/statistics/

Sports stats can be interesting to some (but not all) students – *know your audience*

The NBA & WNBA maintain extensive databases of team and players stats online:

- not easily downloadable, but can filter as desired then copy-and-paste into a CSV file
NBA scoring
Leveraging Big Data – CSTA 2014

Student questions

could easily generalize to consider other stats and/or other years
Twitter Hashtags


Infochimp has collected Twitter data from 3/06 to 11/09
- every token (hashtag/URL/smiley) from 500M+ tweets, organized by month or by hour

<table>
<thead>
<tr>
<th>hashtag ag</th>
<th>year</th>
<th>count</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>200910</td>
<td>1</td>
<td>csta</td>
<td></td>
</tr>
<tr>
<td>200909</td>
<td>4</td>
<td>cstagego</td>
<td></td>
</tr>
<tr>
<td>200907</td>
<td>1</td>
<td>cst ar</td>
<td></td>
</tr>
<tr>
<td>200811</td>
<td>1</td>
<td>cst c</td>
<td></td>
</tr>
<tr>
<td>200812</td>
<td>5</td>
<td>cst c</td>
<td></td>
</tr>
<tr>
<td>200903</td>
<td>3</td>
<td>cst c</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Finally, BIG data!

the Twitter file contains 41M+ lines, takes 1.8 GB
  - too big for standard viewing tools (e.g., Word, Excel)
  - too slow to process in its entirety

first task for students is to filter the data
  - filter out hashtag entries only, separate by year
    hashtags2006 → 4 KB    hashtags2007 → 49 KB
    hashtags2008 → 1.9 MB  hashtags2009 → 33.4 MB

```python
def filterByYear(year):
    year = str(year)
    infile = open("tags.tsv", "r")
    outfile = open("hashtags"+year+".tsv", "w")

    line = infile.readline()
    while line != "
        lineList = line.split("\t")
        if lineList[0] == "hashtag" and lineList[1][:4] == year:
            outfile.write(lineList[1]+"\t"+lineList[2]+"\t"+lineList[3])
        line = infile.readline()
    infile.close()
    outfile.close()
```
Student questions

can then pose interesting, historical questions (e.g., Google *sixwords*, *blogher*)

```
MONTHS = ["January", "February", "March", "April", "May", "June", 
          "July", "August", "September", "October", "November", "December"
]

def tagVolume(year):
    year = str(year)
    total = 0

    data = readData("hashtags"+year+".tsv")
    for month in MONTHS:
        monthStr = year + ("0"+str(MONTHS.index(month)+1))[-2:]
        count = sum([int(line[1]) for line in data if line[0] == monthStr])
        total += count
        print(month.ljust(9), str(count).rjust(8))

    print(18*"")
    print(year.ljust(9), str(total).rjust(8))

def mostFrequentTag(year):
    year = str(year)
    data = readData("hashtags"+year+".tsv")
    for month in MONTHS:
        monthStr = year + ("0"+str(MONTHS.index(month)+1))[-2:]
        pairs = [(0, "NONE") + 
                 ([int(line[1]), line[2]] for line in data 
                  if line[0] == monthStr]
                  (count, tag) = max(pairs)
        print(month.ljust(9), tag, count)
```
Other sources

more datasets at www.infochimps.com/datasets
- 60K+ UFO sightings
- crime rates by state
- Social Security benefits & payments

detailed major league baseball play-by-play data at http://www.retrosheet.org/game.htm

airplane departure/arrival/delay data at stat-computing.org/dataexpo/2009/the-data.html

U.S. census data at www.census.gov/main/www/access.html
From the audience

World Health Organization data repository (from Barb Ericson)


IBM’s Many Eyes data repository (from Barb Ericson)


SAS’s Data Depot, as part of its Curriculum Pathways (from Mason Matthews)

- [http://www.sascurriculumpathways.com/portal/Launch?id=3001](http://www.sascurriculumpathways.com/portal/Launch?id=3001)