Introduction to Data Programming

CSE 160
University of Washington
Spring 2018
Ruth Anderson

Slides based on previous versions by Michael Ernst and earlier versions by Bill Howe

Agenda for Today

- What is this course?
- Course logistics
- Python!

Welcome to CSE 160!

CSE 160 teaches core programming concepts with an emphasis on real data manipulation tasks from science, engineering, and business

Goal by the end of the quarter: Given a data source and a problem description, you can independently write a complete, useful program to solve the problem

Course staff

- Lecturer:
  - Ruth Anderson
- TAs:
  - Ollin Boer Bohan
  - Linxing Jiang
  - Lauren Martini
  - Zhiheng Qin
  - Siyu Wang
  - Lingyue Zhang
  - Alex Zhou

  Ask us for help!

Learning Objectives

- Computational problem-solving
  - Writing a program will become your "go-to" solution for data analysis tasks
- Basic Python proficiency
  - Including experience with relevant libraries for data manipulation, scientific computing, and visualization.
- Experience working with real datasets
  - astronomy, biology, linguistics, oceanography, open government, social networks, and more.
  - You will see that these are easy to process with a program, and that doing so yields insight.

What this course is not

- A “skills course” in Python
  - …though you will become proficient in the basics of the Python programming language
  - …and you will gain experience with some important Python libraries
- A data analysis / “data science” / data visualization course
  - There will be very little statistics knowledge assumed or taught
- A “project” course
  - the assignments are “real,” but are intended to teach specific programming concepts
- A “big data” course
  - Datasets will all fit comfortably in memory
  - No parallel programming

"It's a great time to be a data geek."

-- Roger Barga, Microsoft Research

"The greatest minds of my generation are trying to figure out how to make people click on ads"

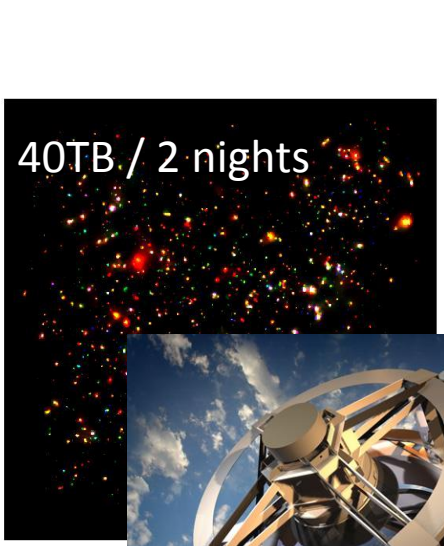
-- Jeff Hammerbacher, co-founder, Cloudera

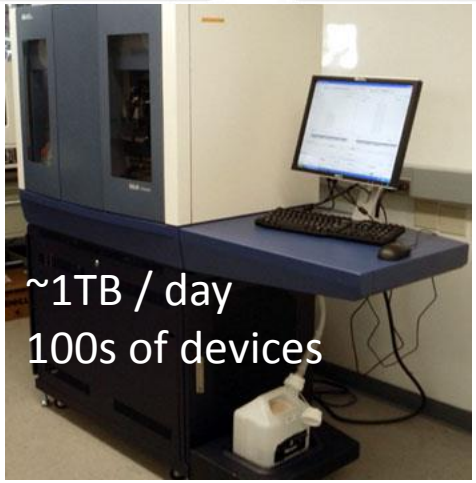
All of science is reducing to computational data manipulation

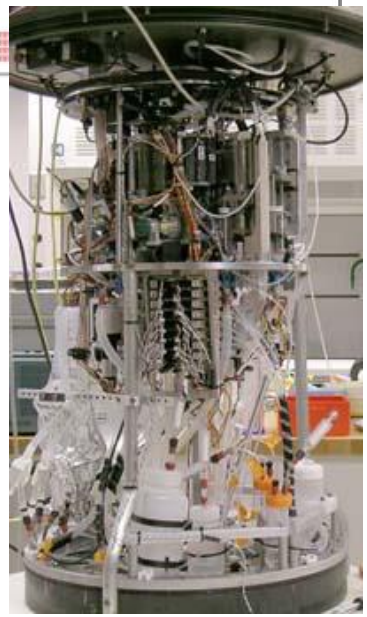
Old model: "Query the world" (Data acquisition coupled to a specific hypothesis)

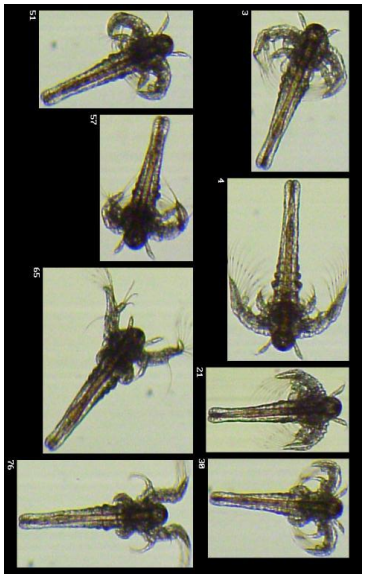
New model: "Download the world" (Data acquisition supports many hypotheses)

- **Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)**
- **Biology: lab automation, high-throughput sequencing,**
- **Oceanography: high-resolution models, cheap sensors, satellites**









Slide from Bill Howe, eScience Institute

Example: Assessing treatment efficacy

	A	B	C	D	E	F	G	H	I	J
1	fu_2wk	fu_4wk	fu_8wk	fu_12wk	fu_16wk	fu_20wk	fu_24wk	total4type_fu	clinic_zip	pt_zip
2	1	3	4	7	9	9	9	12	98405	98405
3	2	4	6	7	8	8	8	8	98405	98403
4	0	0				0	0		98405	98445
5	3	4				5	5		98405	98332
6	0	0				0	0		98405	98405
7	2	2				2	2			98402
8	1	2	5	6	8	10	10	14	98405	98418
9	1	1	2	2	2	2	2	2	98499	98406
10	0	0	1	2	2	2	2	6	98405	98404
11	0	0	0	0	0	0	0	0	98405	98402
12	1	1	2	2	4	4	4	4	98405	98405
13	1								98404	98404
14	2								98499	98498
15	0								98499	98445
16	1								98499	98405
17	1								98499	98498
18	1	3	3	3	3	3	3	3	98499	98499
19	1	1	4	5	7	7	7	7	98499	98371

number of follow ups within 16 weeks after treatment enrollment.

Zip code of clinic

Zip code of patient

Question: Does the distance between the patient's home and clinic influence the number of follow ups, and therefore treatment efficacy?

Python program to assess treatment efficacy

```
# This program reads an Excel spreadsheet whose penultimate
# and antepenultimate columns are zip codes.
# It adds a new last column for the distance between those zip
# codes, and outputs in CSV (comma-separated values) format.
# Call the program with two numeric values:  the first and last
# row to include.
# The output contains the column headers and those rows.

# Libraries to use
import random
import sys
import xlrd        # library for working with Excel spreadsheets
import time
from gdapi import GoogleDirections

# No key needed if few queries
gd = GoogleDirections('dummy-Google-key')

wb = xlrd.open_workbook('mhip_zip_eScience_121611a.xls')
sheet = wb.sheet_by_index(0)

# User input:  first row to process, first row not to process
first_row = max(int(sys.argv[1]), 2)
row_limit = min(int(sys.argv[2]+1), sheet.nrows)

def comma_separated(lst):
  return ",".join([str(s) for s in lst])

headers = sheet.row_values(0) + ["distance"]
print comma_separated(headers)

for rownum in range(first_row,row_limit):
   row = sheet.row_values(rownum)
   (zip1, zip2) = row[-3:-1]
   if zip1 and zip2:
      # Clean the data
      zip1 = str(int(zip1))
      zip2 = str(int(zip2))
      row[-3:-1] = [zip1, zip2]
      # Compute the distance via Google Maps
      try:
         distance = gd.query(zip1,zip2).distance
      except:
         print >> sys.stderr, "Error computing distance:", zip1, zip2
         distance = ""
    # Print the row with the distance
    print comma_separated(row + [distance])
    # Avoid too many Google queries in rapid succession
    time.sleep(random.random()+0.5)
```

23 lines of executable code!

Course logistics

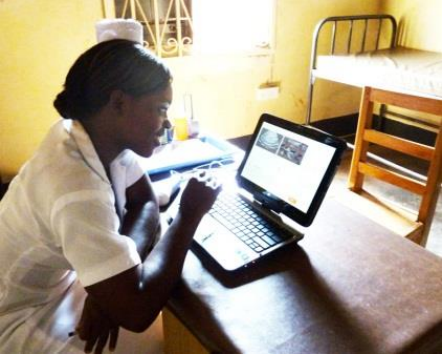
- Website: `http://www.cs.washington.edu/cse160`
- See the website for all administrative details
- Take notes!
- Homework 1 part 1 is due Friday
  - As is a survey
- You get 5 late days throughout the quarter
  - No assignment may be submitted more than 3 days late. (contact the instructor if you are hospitalized)
- If you want to join the class, sign sheet at front of class, email rea@cs.washington.edu, from your @u address

Academic Integrity

- Honest work is required of a scientist or engineer
- Collaboration policy on the course web. **Read it!**
  - Discussion is permitted
  - **Carrying materials from discussion is not permitted**
  - Everything you turn in must be your own work
    - Cite your sources, explain any unconventional action
  - **You may not view others' work**
  - If you have a question about the policy, just ask us
- I trust you completely
- I have no sympathy for trust violations – nor should you

How to succeed

- No prerequisites
- **Non**-predictors for success:
  - Past programming experience
  - Enthusiasm for games or computers
- Programming and data analysis are challenging
- Every one of you can succeed
  - There is no such thing as a "born programmer"
  - Work hard
  - Follow directions
  - Be methodical
  - *Think* before you act
  - Try on your own, then ask for help
  - Start early



Me (Ruth Anderson)

- Grad Student at UW: in Programming Languages, Compilers, Parallel Computing
- Taught Computer Science at the University of Virginia for 5 years
- PhD at UW: in Educational Technology, Pen Computing
- Current Research: Computing and the Developing World, Computer Science Education

Introductions

- Name
- Email address
- Major
- Year (1,2,3,4,5)
- Hometown
- Interesting Fact or what I did over break.