2014-06-05 Teaching Analysis and Presentation of Open Data

From the announcement e-mail on readings for this RIT Reading Group meeting:

> _Raymond Yee, who teaches the "Working with Open Data" course at the School of Information, will facilitate a discussion about "Teaching Analysis and Presentation of Open Data" [...]._
>
> _Here are Raymond's readings for this group (written by him, hence the first-person in italics):_
>
> As a way of jumpstarting a discussion on the topic of teaching open data analysis, I'd like to briefly share my experiences of teaching "Working with Open Data". Here's a write-up no the 2013 edition of the course.
>
> _http://blog.fperez.org/2013/05/exploring-open-data-with-ipython.html_
>
> I've not written up an analysis of the 2014 course yet. But you can find the course materials for the course at
>
> _http://is.gd/wwod14_
>
> Specifically, the final projects can be found at
>
> _https://github.com/working-with-open-data-2014/project-organization/wiki/Working-with-Open-Data-2014-Projects_
>
> I mention Software Carpentry below, so it'd be good for us to take a look at:
>
> _http://software-carpentry.org/faq.html_
>
> _Our discussion will focus on these questions (a mix suggested by Steve and Raymond):_
>
> _(a) What technical issues, if any, are presented by open data that differ from copyrighted, restricted, or sensitive data sets used in some campus research?_
>
> _(b) What kinds of technical / infrastructural resources could be made available to instructors to facilitate teaching and focus class/student time on topics and activities of core value in the context of a course/project?_
>
> _(c) What is suggested by the following brainstormed list of areas for improvement for a next version of "Working with Open Data" -- improvements Raymond would try to make if more time and resources were available:_
>
> _* expand the implementation of bCourses/Canvas API to make it easier to do more frequent quizzes and automated grading_
> _* more video, screencasts to enable students to revisit my lectures outside (and inside the classroom)_
> _* having students make screencasts_
> _* teach students how to use the whole range of the Census API and how to process bulk census data_
> _* basic front end web development skills, including JavaScript_
> _* Firmer integration of software carpentry and introductory Python review_
> _* Statistics training_
> _* Test driven development_
> _* Python outside of the IPython Notebook_
> _* Systematic overview and coverage of the open data intellectual property regime and movement_
> _* Formal writing and presentation skills_
> _* Working Real world clients_
> _* Coverage of scaling up computation_
>
> _(d) In general, what can be done to lower the intellectual and technical barriers in working with the combination of open data, open source software, and scalable on demand computational infrastructure?_

Participating:

- Steve Masover, RIT
- David Greenbaum, RIT
- Arvin, EDW
- [name?], EDW
- Tim Dennis, Library

- John Lowe, RIT
- Chris Hoffman, RIT
- Sharon Goetz, Library
- Patrick Schmitz, RIT
- Michael Black, Pahma
- Greg Hamilton, EDW
- Rick Jaffe, RIT
- Raymond Yee, School of Information
- Aaron Culich, RIT
- Chris Pachorik, Statistics
- Aron Roberts, RIT
- Mark Chiang, EDW

Notes:

Raymond
=============

* Working with Open Data course: 34 students this year, 24 students last year. 1 TA plus Raymond. Students arrive with at least 1 semester of Python programming under their belt
* Course text is Python for Data Analysis (O'Reilly). Author Wes McKinney is author of Pandas library (naively: dealing with rectangular data -- data that fits into rows and columns).
* Heavy use of IPython in the course, specifically IPython notebooks.
* Data sets used in course included focus on census data. Example: using IPython notebook to generate maps of race/ethnicity in US counties. This example uses Javascript D3 to render map from these data. Cf. GitHub link in reading list communication, above.
* Class deliverable was an IPython notebook that described and contextualized an analysis, and the code / outputs that implement that analysis.
* Raymond structures the class to aim toward a public open house in which students showcase their work, which motivates them to polish their articulation and presentation
* In general, Raymond uses a 'dive right in' approach to teaching ... engage with problems, with data ... that before theory or systematic treatment of the problem space(s).
* Interestingly, Raymond put a lot of time/effort into coming up with example projects that interested him, and he presented them to students ... and it turned out they were pretty much all interested in different things. Students gravitated toward their own interests.

Arvin: Data cleansing first? E.g., discrepancy in county naming standards between census data and pollution data available in data.gov
Raymond: A more systematic treatment of issues in file formats, data cleanliness, etc. might be an area in which the course could be augmented /improved. Dirty data sets -- esp. dirty aspects of generally clean data sets -- are a teaching opportunity: how the real world of data actually is.
Aaron: All APIs, some web scraping?
Raymond: A mix. APIs, web scraping, download data sets.

Aron: When you run into challenges with particular data sets, what kinds of sharing is there among people who have dealt with the same or similar data sets?
Raymond: Hard to generalize.
Chris P: Nobody's going to get academic credit for cleaning data or for explaining how they did it. So probably not a lot of that.
Tim: I see more help published on techniques, but not in the context of specific data sets.

Patrick: A naive question. Open data analogy to open source: is there no publication of cleaned data sets, derivatives, algorithms that clean data sets?
Raymond: A Github of data ... not there yet. US Open Data Institute: http://usodi.org/ ... aware of but haven't looked into this. Waldo Jaquith, http://dat-data.com/.
Patrick: Given that lots of time goes into cleaning data, do you or you students or others find that once you get to that step in an overall analysis that you've produced something of any use to others, and consider publishing that?
Raymond: Hard to find motivation / reward for doing that. So, generally no due to lack of incentive.
Chris H: In stat (Chris P), Library (Tim D), EDW (many) -- do you find occasion to publish patterns, algorithms, etc. re: data cleaning? In CollectionSpace (current migration of BAM/PFA data) we're finding that it's very hard to find/justify the expense of time.
Tim: Emergent model of literate programming (as in IPython notebooks). Should be documented with the code, discoverable, published in a repository is good too.
Patrick: Reputational motiviation in communities who deal with big, well-known data sets?
Chris P: from Statistics dept perspective, we're generally working with scientists, and rely on them to handle that aspect of work prior to our engagement.
Steve: What if any technical infrastructure that UCB might provide would make teaching your course easier, or otherwise support pedagogy or research
David: Poli Sci, Classics -- data sets w/ rich mark up and description
John Lowe: perhaps the analogy of software:data is false. Data as facts has different model from software as process.

Steve: On second question: what kinds of technical/infrastructural resources could be made available to instructors...?
Raymond: BCE @ D-Lab. Interested in getting students started in problem-solving -- including data cleansing headaches -- without spending too much time getting environments, software versions, etc. set up? Virtualization may solve some of this, but I question whether we can fully address usability issues. Will we exchange one set of problems (usabiity re: VMs / Virtual Workstations) for others (install packages, version issues, etc.).
Tim: How much time gets expended in software / environment set up?
Raymond: Haven't tracked. It happens that you think you've solved environmental set up but then you have to return to it as problems crop up.
Patrick: IPython on the web environments?
Raymond: wakari ... had students try this ... had its own flakiness ... free versions underprovisioned (can't blame Continuum [vendor] for that) ... maybe there was a network bandwith issue in South Hall
Aron: ~20% of Python lab at D-Lab day two students were still at that point struggling a bit with their environments
Raymond: We didn't get to the point of scaling up ... if we had gotten there, we could have used Amazon credits to migrate into more powerful computation resources, but that's not a permanent solution -- not sure how long Amazon grants will last. Good to educate students about available options from Amazon to Rackspace to Google to (in the future) some Berkeley service ... might also be useful to educate about scripted solutions to VM provision, as an approach to avoiding vendor lock-in
John Lowe: Don't students need to know something about statistics in addition to knowledge of Python
Raymond: Sure ... most projects we did did not turn on statistical analysis, but that point is very well taken.

Next (6/19): Data Curation, organized by Chris Hoffman