

L21

July 24, 2017

1 Lecture 21: Open Data Science

CSCI 1360E: Foundations for Informatics and Analytics

1.1 Overview and Objectives

At this point, you're ready to dive into data science headfirst. One of the best ways to learn more about specific problems is to see how those problems have been addressed so far; reading through existing code can help immensely. But this touches on an important and emerging area of science called "Open Science." By the end of this lecture, you should be able to

- Define "open science" and its importance to scientific inquiry
- Recall the core strategies for reproducing and replicating results
- Package and release your Python code via distribution channels

1.2 Part 1: What is "Open Science"?

Simply put, Open Science is the **movement to make all scientific data, methods, and materials accessible to all levels of society.**

Why is this a good thing?

- The vast majority of research is publicly funded; it would seem logical that the public have access to it!
- Open and transparent research makes peer review and replication much easier.
- There is some convincing evidence that Open Science gives projects more downstream impact in the scientific community

Nonetheless, there are some downsides to making everything openly available.

- The deluge of science will overwhelm already heavily-burdened researchers.
- The tools could be used for more nefarious purposes (a good example is a particularly virulent strain of influenza that researchers were experimenting with a few years back that could potentially be used as a bioweapon).

My opinion--as you've probably guessed by the title of the lecture--is that the benefits of good Open Science practices outweigh the drawbacks, for the following reason:

I learn best when I can dig in and get my hands dirty.

Reading a paper or even a blog post that vaguely describes a method is one thing. Actually seeing the code, changing it, and re-running it to observe the results is something else entirely and, so I believe, is vastly superior in educational terms.

The scientific deluge is legitimate, though this was already happening even without the addition of open data, open access, and open source. And it would seem that, while we do absolutely need to exercise caution in our research and not pursue the ends by any means necessary, science is the pursuit of knowledge for its own sake and that should also be respected to the highest degree.

To that end, there are **six** main themes that comprise the Open Science guidelines.

1. Open data: all data used in the project should be made available.
2. Open source: all code written in the project should be publicly available.
3. Open methods: the exact procedure of the project is publicly documented.
4. Open review: correspondence between reviewers and authors is public.
5. Open access: resulting publications are publicly available.
6. Open education: all education materials are publicly available.

1.2.1 1: Open Data

If you had to pick the "core" of Open Science, this would probably be it. All of the data used in your study and experiments are published online.

This is definitely a shift from prior precedent; most raw data from scientific experiments remain cloistered.

The situation is further complicated by Terms of Service agreements that prohibit the sharing of data collected. For example: if you hooked up a Python client to listen to and capture public Twitter posts, you are forbidden from sharing the Twitter data publicly. Which seems odd, given that the data are public anyway, but there you go.

Repositories and online data banks have sprung up around this idea. Many research institutions host their own open data repositories, as do some large tech companies.

- CERN, the organization behind the Large Hadron Collider, has posted its data online: <http://opendata.cern.ch/about/CMS>
- Amazon has released its own set of large public datasets: <https://aws.amazon.com/public-data-sets/>
- DataHub is a general-purpose repository for anyone to submit their own datasets. <https://datahub.io/en/dataset>

1.2.2 2: Open Source

This is probably the part you're most familiar with. Any (and *all*) code that's used in your project is published somewhere publicly for download.

There are certainly conditions where code can't be fully open sourced--proprietary corporate secrets, pending patents, etc--but to fully adhere to Open Science, the code has to be made completely available for anyone.

Like with open data, there are numerous **repositories** across the web that specialize in providing publicly-available versioning systems for both maintaining and publishing your code.

- GitHub is easily the most popular, and is where the materials for this course are published! <https://github.com/>

- BitBucket is another option that also uses git to manage team codebases <https://bitbucket.org/>
- SourceForge is one of the oldest and most well-known online repositories <https://sourceforge.net/>

1.2.3 3: Open Methods

This is probably the trickiest item. How does one make *methods* reproducible?

Open source code is part of it, but even more important is the effort put into making the methods in the code understandable. This takes several forms:

- Documentation, both as accompanying doc files (e.g. JavaDoc) as well as in-code comments
- Proofs of the methods devised if they're novel, or references to their original sources
- Pre-packaged examples that will run with little or no prior configuration on the user's part
- Self-contained virtual container scripts that satisfy all the prerequisites (more on this later)

One good example of making sure your code documentation is up to par is Continuum IO's new automated README generator, [kapsel](#).

1.2.4 4: Open Review

The cornerstone of the scientific process is that of *peer review*: your peers, your colleagues, your fellow researchers should vet your work before it's officially included as part of the scientific literature.

However, this process is fraught with ambiguity and opacity. Conflicts of interest can potentially lead to biased reviews (if you're reviewing the paper of a competitor, it isn't exactly in your best interest to go easy on them), and it can be difficult to assess a published paper in the public sphere without a trail of edits from which to begin.

Online open review journals such as [The Journal of Open Source Software \(JOSS\)](#) have begun proliferating to address this shortcoming.

Reviews essentially take the form of GitHub tickets, and researchers in the field can discuss and debate the merits of the work in an open forum.

1.2.5 5: Open Access

Once a project is published, the paper should be made publicly available for anyone, anywhere to download and read for themselves.

Easily the most popular open access paper repository is [arXiv](#) (pronounced "archive"...geddit?). Other repositories modeled directly after its success, such as bioRxiv, have already started springing up.

arXiv is already *hugely* popular.

In fact, so many papers are archived here on a regular basis, that someone created their own open source "aggregator" service: <http://www.arxiv-sanity.com>, which collates the papers you want to read and helps filter out all the others.

Submission: LAS: an integrated language analysis tool for multiple languages #35

 **Open** whedon opened this issue 26 days ago · 8 comments



whedon commented 26 days ago • edited



Submitting author: @jiemakel (Eetu Mäkelä)
Repository: <https://github.com/jiemakel/las>
Version: v1.4.8

Status

JOSS Under Review

Status badge code:

```
HTML: <a href="http://joss.theoj.org/papers/87f0093e30a01b1078d8dea41476a08a">

### 1.3.5 Licensing

Yes, this is super boring, I agree. Fortunately, the wonderful folks at GitHub have created an awesome resource for you to use.

**Choose a License:** It provides a handy choose-your-adventure flowchart for picking the perfect open software license for your project. I highly recommend it.

However, if you're just looking for a **tl;dr version** and want the basic lay of the land, GitHub's default licenses for new projects include

- MIT License (pretty much the "no warranty whatsoever, use at your own risk" license)
- Apache 2.0 License (almost exactly like the MIT, but allows for patents and trademarks)
- GPLv3 License (requires anyone who modifies your code to publish those modifications under GPLv3 too)

## 1.4 Review Questions

Some questions to discuss and consider:

- 1: What are your thoughts on Open Science?
- 2: Have you ever developed open source code before? If so, what projects? If not, are there any projects you'd be interested in working on?
- 3: Have you ever dealt with open source licenses? Have you heard of any before this lecture?
- 4: Have you ever written unit tests before? If so, what did you test? If not, what do you think the "unit" refers to?

## 1.5 Course Administrivia

- **Assignment 10 due tonight!** Then it's all over for assignments!
- **Final exam review session Wednesday, 10am-12pm!** Watch the Slack chat for the Hangouts link!
- **Final exam is Friday, July 28!**

## 1.6 Additional Resources

1. GitHub template for open source Python projects <https://github.com/uwescience/shablona>
2. Mozilla Open Science Lab <https://science.mozilla.org/>
3. Open Science Framework (OSF) for all phases of the scientific pipeline <https://osf.io/>
4. The Center for Open Science (COS), which supports OSF <https://cos.io/>