Code sharing, Discovery and Credit

Steven K. Morley
Shameless self-promotion
github.com/drsteve – please check out spacepy and others…

SpacePy
• Python/Fortran/C

PyForecastTools
• Python

RAM-SCB
• Fortran

LANLGeoMag
• C

MinimalSubstormModel
• Python
I made this
Officially out in the open

- Choose a license*
  - see https://opensource.org/licenses
  - Most of my projects use BSD license
  - GPL can restrict use…

- Choose a host
  - E.g. GitHub, SourceForge, BitBucket

- Make it discoverable!
  - LANLGeoMag is AUL 9, in use for several operational needs, but is not widely known

UNCLASSIFIED
I made this
Officially out in the open

2009:
- SpacePy is born
- VCS using local SVN

2010:
- Open-source approval from LANL (PSF license)
- Code migrated to SourceForge
- SpacePy presentation at SciPy conference!

2011:
- Migrate to git
- SciPy conference proceedings published
Hey, you! I made this!
Making code discoverable

For Python, PyPI is key!
  • Discoverability
  • Installability

Strong web presence with good keywords helps discoverability

Often “discoverability” is just “what comes up when I GDS?”

UNCLASSIFIED
Hey, you! I made this!
Making code discoverable

Other options:
• Astrophysics Source Code Library
• code.nasa.gov (for NASA employees)
• Journal articles
where credit is due…

Making code citeable

Digital Object Identifiers (DOIs) were introduced to provide permanent links to online resources.
- URLs can change…

But my code is on GitHub!
- GitHub provides a URL
- ASCL records URLs
- Journal articles have DOIs, but aren’t a good way of storing code
Your code is data
Various variants of versioning

AGU requires that data are made available at the time of publication

For the purposes of this policy, data include, but are not limited to, the following:

* New code/computer software used to generate results or analyses reported in the paper.

Open data repositories that issue DOIs include:

* Zenodo (zenodo.org)
* Dryad (datadryad.org)
* Some institutions and gov’t agencies

UNCLASSIFIED
Your code is data
Examples of data and code at Zenodo

CXD Energetic Particle Data (Selection of SEP Events from 2003 - 2017)

Matthew R. Carver; John P. Sullivan; Steven K. Morley; Benjamin Norman

Energetic particle data (electrons and protons) from the CXD instrument aboard GPS satellites. This is a subset of the data described at doi:10.1002/2017SW001604 and contains data from the following SEP events (listed by onset date):


The work to create this data was funded and performed under the auspices of the US Department of Energy.

UNCLASSIFIED
Zenodo-GitHub integration

When should a “release” be made?

1. Flip the switch
   Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.

2. Create a release
   Go to GitHub and create a release. Zenodo will automatically download a zip-ball of each new release and register a DOI.

3. Get the badge
   After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below.

Enabled Repositories:
- drsteve/LANLGeoMag
  DOI 10.5281/zenodo.1195041
- drsteve/PyForecastTools
  DOI 10.5281/zenodo.1256922
I found your GitHub

Now what?

CI badge shows that unit tests are passing

DOI badge links to Zenodo

Where are the docs?

PyForecastTools

A Python module to provide model validation and forecast verification tools. The module builds on the scientific Python stack (Python, Numpy) and uses the dmarrow class from SpacePy's datamodel.

SpacePy is available through the Python Package Index, MacPorts, and is under version control at sourceforge.net/p/spacepy/ If SpacePy is not available a reduced functionality implementation of the class is provided with this package.

To install (local user), run

```bash
python setup.py install --user
```

The module can then be imported (within a Python script or interpreter) by

```python
import verify
```

For help, please see the docstrings for each function and/or class.

Additional documentation is under development using Github pages at drsteve.github.io/PyForecastTools, and source for this is in the docs folder.
Please publish me!

Where can code papers go?

- JGR: Space Physics
  - E.g., Stoneback et al. (2018), PYSAT: Python Satellite Data Analysis Toolkit
- SciPy Conference Proceedings
  - E.g., Morley et al. (2010), SpacePy: A Python-based library of tools for the space sciences
- Journal of Open Source Software
  - “Developer-friendly”: Peer-reviewed short summaries with code
  - Reviews are done as GitHub tickets; code is archived on Zenodo
- Journal of Open Research Software
- Journal of Computational Science
- Geoscientific Model Development
So my code is open source
- It’s under version control at github
- I have it indexed at PyPI and ASCL
- My paper has been published and archived at Zenodo

So now what do I cite?
- GitHub URL? (Quasi-citeable)
- The Zenodo archive? (Citeable; Has a DOI)
- The ASCL record? (Citeable; Does not have a DOI)
- The journal article? (Citeable; Has a DOI)

Now imagine there have been several “releases”, each with multiple DOIs
The citation stealer
Pitfalls of proliferating products

Publicize preferred citation.

Make preferred citation data available in code!

```python
>>> import spacepy
>>> spacepy.__citation__
```

Zenodo provides a “master” DOI that always points to the latest version.
The citation stealer
Pitfalls of proliferating products

Dryad includes preferred citation data

Can be done in code, e.g., in README markdown file. Good for GitHub projects.

When using this data, please cite the original publication:


Additionally, please cite the Dryad data package:

Code Sharing
Releasing code is not enough

- Release code with open-source license
  - Make code discoverable
    - Make code citeable
  - Engage with, and support, your community
- Establish community best practices to make all this easier!

Make code useable and trustable!
(Document and unit-test!)