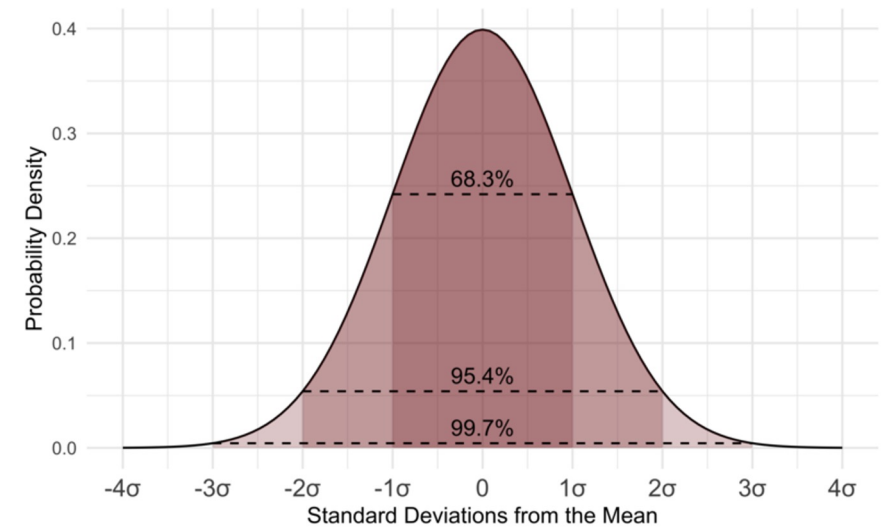
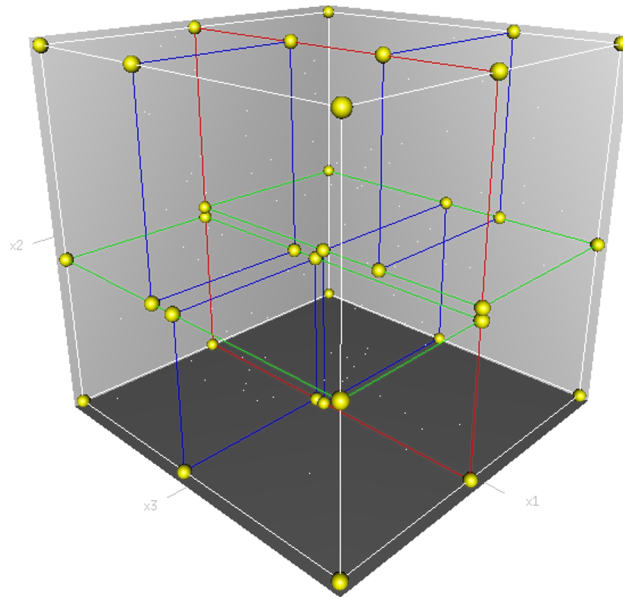
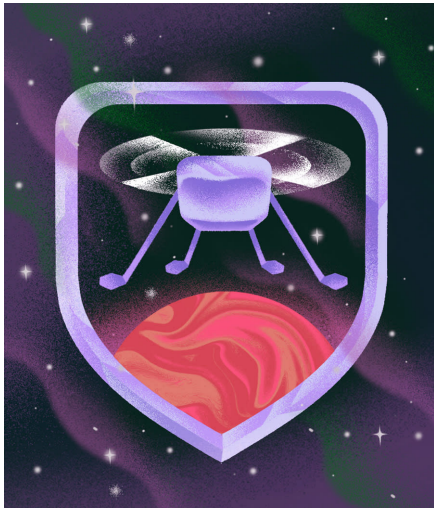




SciPy Library Update



July 13/2022





Meson Source Build Time Improvements



| Hardware | Physical Cores | meson: 2 x cores (m:s) | meson: 1 x cores (m:s) | distutils (m:s) | fold speedup |
|------------------------------|----------------|------------------------|------------------------|-----------------|--------------|
| Intel i9-7900X CPU @ 3.30GHz | 10 | 02:04 | 02:20 | 11:04 | 5.4 |
| AMD EPYC 7702 CPU @ 1.64GHz | 128 | 01:35 | 01:34 | 08:19 | 5.3 |
| ARM Cavium ThunderX2 B0 | 64 | 03:37 | 03:27 | 32:00 | 9.3 |
| IBM POWER9 | 40 | 03:09 | 03:09 | 23:03 | 7.3 |

maintenance/1.9.x hash: c545f52 ; GNU compiler toolchain on Linux

``time python dev.py -j {threads} --build-only` vs. `time python setup.py install``

How to get even faster builds?

We could try breaking the C++ code into smaller compilation units, but there are no plans for that at this time.



Developer CLI Experience Improvements



Reusable infrastructure:

<https://github.com/pydoit/pydevtool>

Broader Adoption in Community?




Usage: dev.py [OPTIONS] COMMAND [ARGS]...

Developer Tool for SciPy



Commands that require a built/installed instance are marked with .

python dev.py --build-dir my-build test -s stats



Options

| | | |
|-------------------------|--------------------|--|
| --help | | Show this message and exit. |
| --build-dir | BUILD_DIR |  Relative path to the build directory. [default: build] |
| --no-build | -n |  do not build the project (note event python only modification require build) |
| --install-prefix | INSTALL_DIR |  Relative path to the install directory. Default is <build-dir>-install. |




build & testing

| | |
|--------------|---|
| build |  build & install package on path |
| test |  Run tests |



static checkers

| | |
|-------------|--|
| lint |  run flake8, and check PEP 8 compliance on branch diff. |
| mypy |  Run mypy on the codebase |

environments

| | |
|----------------|--|
| shell |  Start Unix shell with PYTHONPATH set |
| python |  Start a Python shell with PYTHONPATH set |
| ipython |  Start IPython shell with PYTHONPATH set |

documentation

| | |
|-----------------------|---|
| doc |  Build documentation |
| refguide-check |  Run refguide check |

1.8.x



Highlights From Past Year



1.9.x

- A sparse array API was added for testing
- The sparse SVD library PROPACK is now vendored with SciPy (default off; set `USE_PROPACK=1`)
- A new `scipy.stats.sampling` submodule that leverages the UNU.RAN C library to sample from arbitrary univariate non-uniform continuous and discrete distributions
- All namespaces that were private but happened to miss underscores in their names have been deprecated.

- `scipy.signal.spline`
- `scipy.ndimage.filters`
- `scipy.ndimage.fourier`
- `scipy.ndimage.measurements`
- `scipy.ndimage.morphology`
- ... many others ...

- `scipy.optimize.milp`, new function for mixed-integer linear programming.
- `scipy.stats.fit` for fitting discrete and continuous distributions to data
- Tensor-product spline interpolation modes were added to `scipy.interpolate.RegularGridInterpolator`
- A new global optimizer (Dividing RECTangles algorithm) `scipy.optimize.direct`
- New contributor guide:

conda

pip+venv

With **conda** installed (through [Miniforge](#) or [Mambaforge](#), [Miniconda](#) or [Anaconda](#)), execute the following commands at the terminal from the base directory of your [SciPy](#) clone:

```
# Create an environment with all development dependencies
conda env create -f environment.yml # works with 'mamba' too
# Activate the environment
conda activate scipy-dev
```

Your command prompt now lists the name of your new environment, like so `(scipy-dev)$`.

Finally, build SciPy for development and run the test suite to make sure your installation is successful. On Linux and OSX, you should use:

```
python dev.py
```



Sustainability



- NASA ROSES (+Pandas, NumPy, scikit-learn): Approx. \$1.3 M for Quansight (lead), LANL (subaward), CalPoly (subaward)
- 4 x NumFOCUS SDGs: onboarding grant, PROPACK, mixed integer programming solver, joint sphinx docs improvements
- CZI EOSS Cycle 4 grant (+Pandas, NumPy, Matplotlib): Melissa Mendonça (Quansight)—onboarding, contributor experience, community organization, DEI
- Tidelift & Quansight: Build system improvements
- New team members: Atsushi Sakai, Albert Steppi, Tirth Patel
- Sprint this weekend!