UNIVERSITY of WASHINGTON

Software Design for Data Science

Advanced Packages

Melissa Winstanley
University of Washington
March 6, 2025

Class Updates

Next week: Practice Presentations

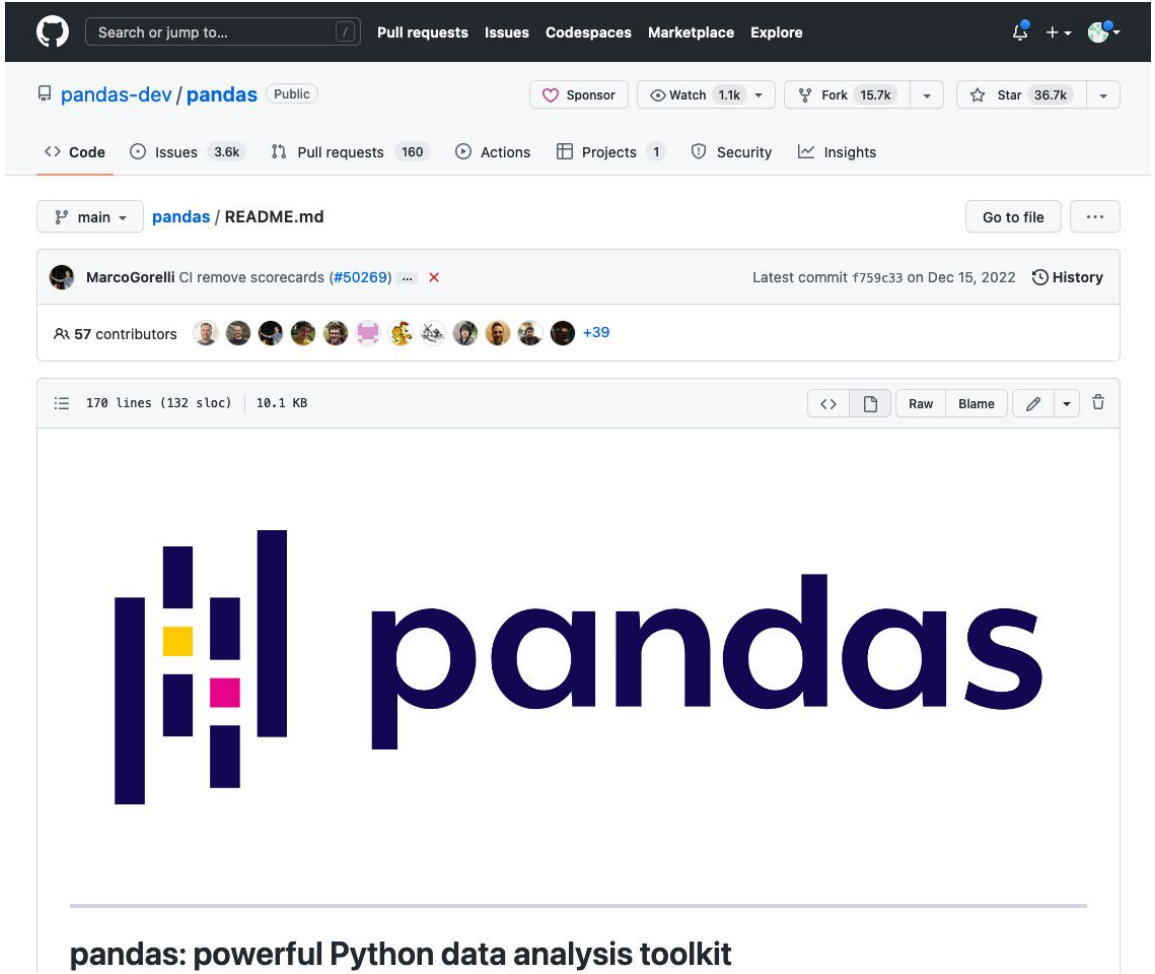
- Similar to your first demo - low stress
- You should ALSO have a first draft of your presentation
  - Definitely does not need to be complete, but we want to see the beginning
- This is also your time to have any last minute discussions with instructors

Project Requirements: stuff we've learned

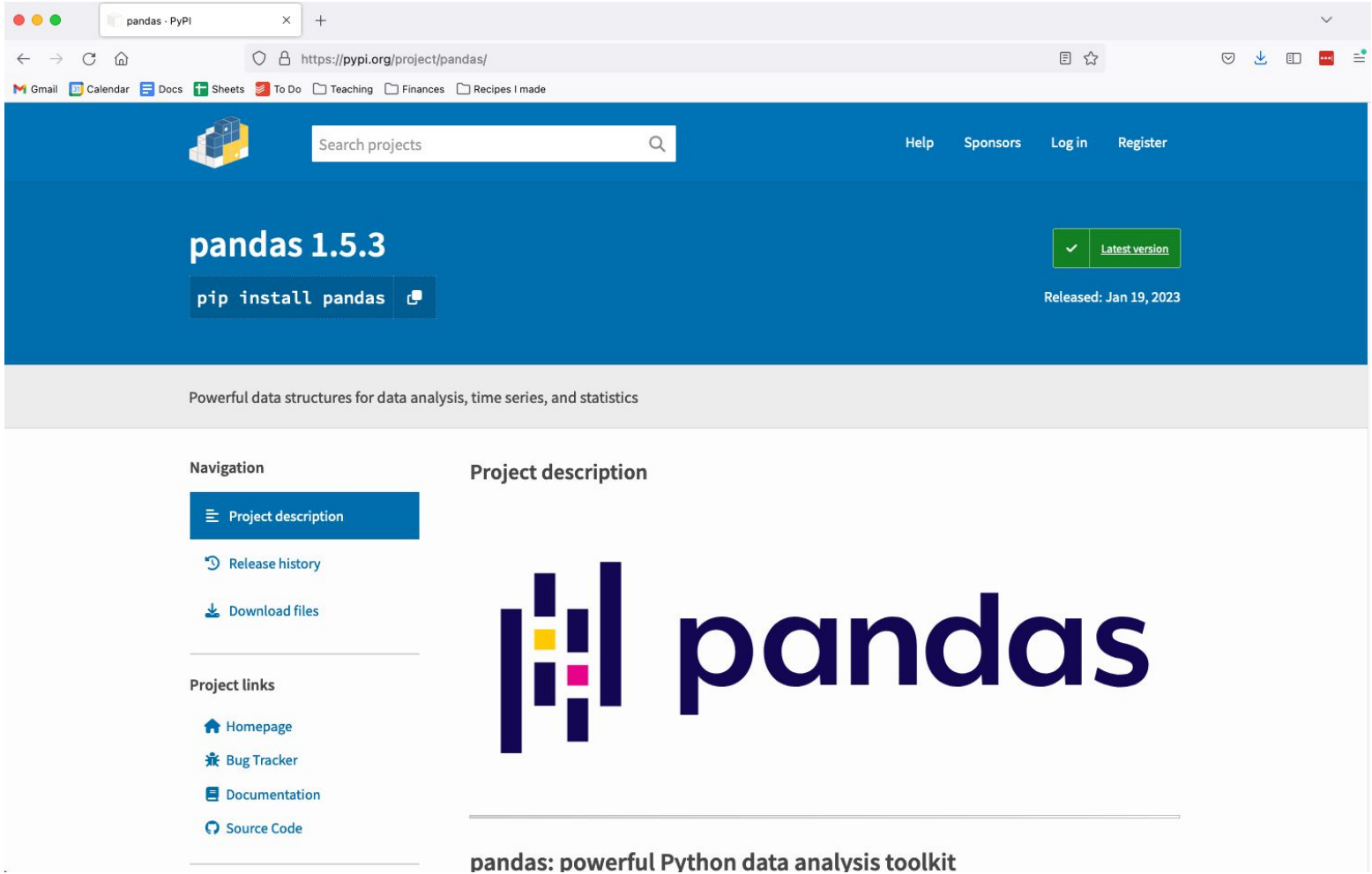
Just to reiterate: you must include the following in your projects.

- **Tests**
  - High test coverage
  - High coverage is not *sufficient* - make sure you write good tests
- **README**
  - Updated!
- **Inline documentation (docstrings)**
  - For functions/modules
- **examples** folder
  - How do various users interact with your system?
  - How do we (the instructors) run the code and tests?
- **Style**: pylint 100%
- **Continuous Integration**
- **Pull requests & reviews**
- **Package distribution** *(today!)*

Distributing a Python package


Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
pandas-dev / pandas
Public
Sponsor
Watch 1.1k
Fork 15.7k
Star 36.7k
Code
Issues 3.6k
Pull requests 160
Actions
Projects 1
Security
Insights
main
pandas / README.md
Go to file
MarcoGorelli CI remove scorecards (#50269)
Latest commit f759c33 on Dec 15, 2022
History
57 contributors
+39
170 lines (132 sloc)
10.1 KB
Raw
Blame
pandas
pandas: powerful Python data analysis toolkit

PyPI


pandas · PyPI
https://pypi.org/project/pandas/
Gmail
Calendar
Docs
Sheets
To Do
Teaching
Finances
Recipes I made
Search projects
Help
Sponsors
Log in
Register
pandas 1.5.3
pip install pandas
Latest version
Released: Jan 19, 2023
Powerful data structures for data analysis, time series, and statistics
Navigation
Project description
Release history
Download files
Project links
Homepage
Bug Tracker
Documentation
Source Code
Project description
pandas
pandas: powerful Python data analysis toolkit

Extending project structure for PyPI

```
myproject/
    README.md
    LICENSE
    pyproject.toml
    myproject/
        __init__.py
        core.py
        subpackage/
        submodule/
        tests/
```

Contains metadata for the package for PyPI. This metadata is then piped to a package builder - often `setuptools` but there are others like `hatchling`

https://packaging.python.org/en/latest/tutorials/packaging-projects/

Extending project structure for PyPI

```
myproject/
    README.md
    LICENSE
    pyproject.toml
    requirements.txt
    myproject/
        __init__.py
        core.py
        subpackage/
        submodule/
        tests/
```

Contains absolute dependencies, especially useful if you're publishing an application.
Can be generated using:

```
conda list --export > requirements.txt
```

OR

```
pip freeze > requirements.txt
```

Extending project structure for PyPI

```
myproject/
    README.md
    LICENSE
    pyproject.toml
    requirements.txt
    MANIFEST.in
    myproject/
        __init__.py
        core.py
        subpackage/
        submodule/
        tests/
```

Specify data and files that should also be packaged in addition to the Python modules

Example `pyproject.toml`

https://github.com/UWDATA515/ci_example/blob/main/pyproject.toml

More examples

Check out existing Python packages. These may be more complicated:

https://github.com/numpy/numpy

https://github.com/pandas-dev/pandas

Submitting your package to PyPI

Update your code and version number. Run your test suites and ensure your code works as intended. Create a PyPI account if you don't have one already

Create your source, and if desired, binary distribution:

```
$ python -m build
```

Install `twine` package to submit builds to PyPI.

(Can install using conda install twine, pip install twine, etc.)

```
$ twine upload dist/*
```

Exercise: Set up your project's `pyproject.toml` file

https://setuptools.pypa.io/en/latest/userguide/index.html

Make sure it builds!

```
python -m build
```

(remember to install the build tools as part of your environment!)

Test it out!

- Move the built tar.gz from dist to a folder outside your repo
- Create a new (empty) conda environment
- `python3 -m pip install ./<your-package-name>.tar.gz`
- Try running python and try importing your modules and functions