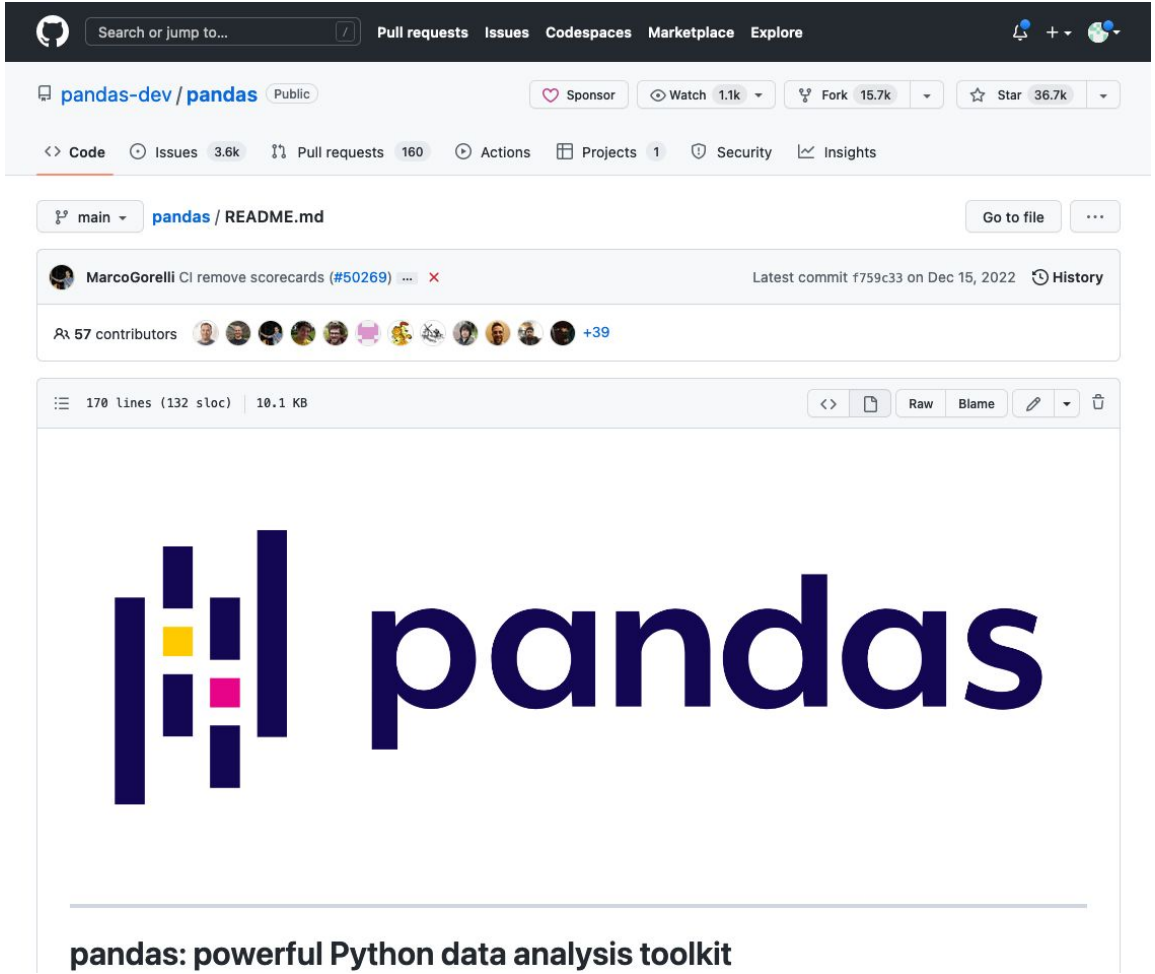
UNIVERSITY of WASHINGTON

Software Design for Data Science

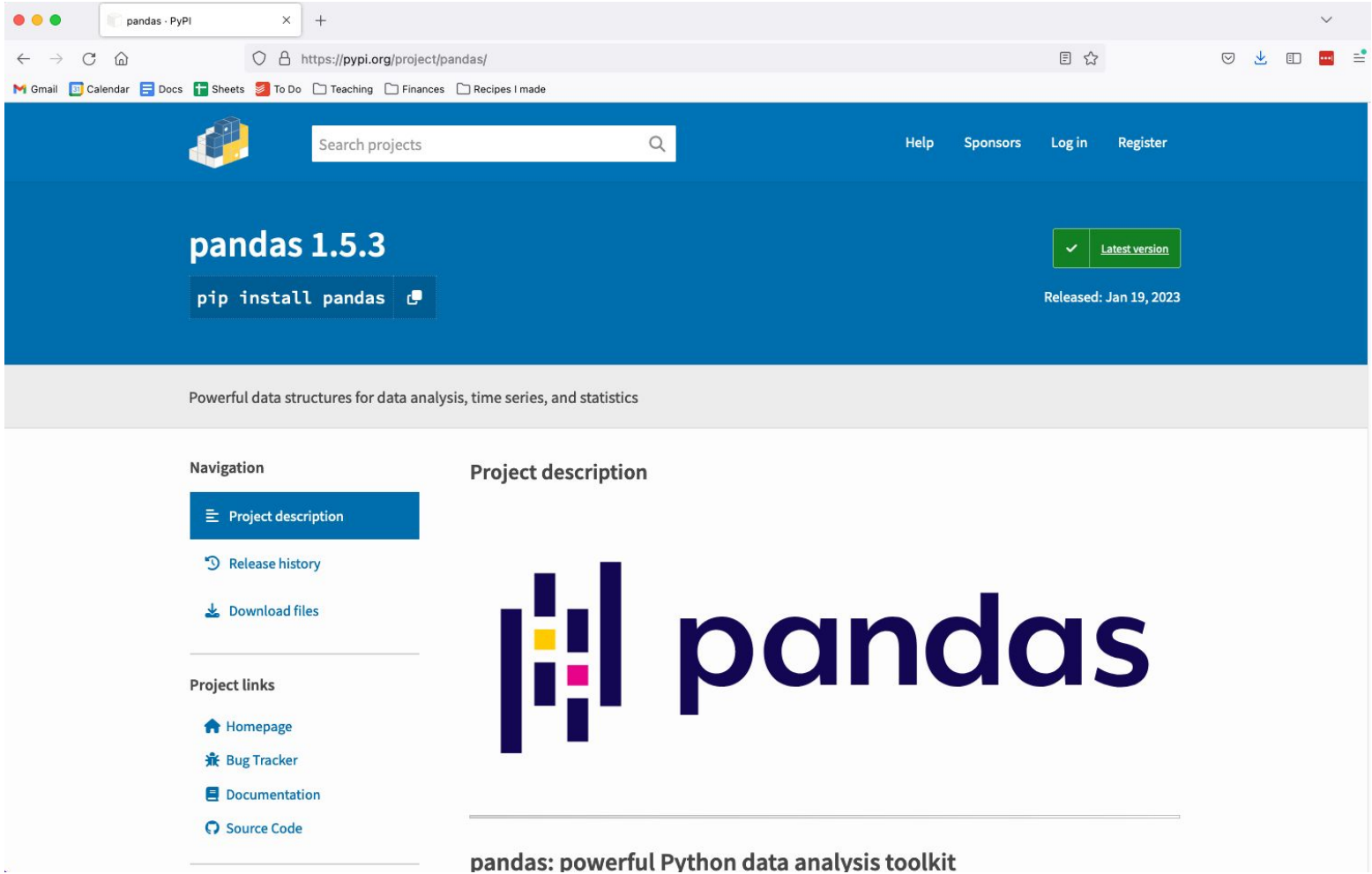
Advanced Packages

Melissa Winstanley
University of Washington
February 29, 2024

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

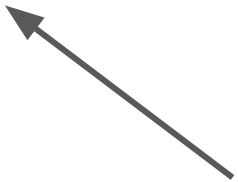
```
myproject/
    README.md
    LICENSE
    myproject/
        __init__.py
        pyproject.toml
        core.py
        subpackage/
        submodule/
        tests/
```

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
    myproject/
        __init__.py
        setup.py
        requirements.txt
        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
    myproject/
        __init__.py
        setup.py
        requirements.txt
        MANIFEST.in
        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 setup.py bdist egg upload [options]
$ python setup.py bdist wininst [options]
$ python setup.py sdist [options]
```

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