• Project description. Start by describing what the project is about, what the software does, what it’s used for.

Originally developed by Wes McKinney in 2008 whilst working at AQR capital management, **“Pandas** is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or ‘labelled’ data both easy and intuitive.” Pandas Development Team(2025). Beyond providing the foundational data frame object (derived from pythons native dictionary data type), the library provides a series of methods serving use cases from machine learning data preprocessing to exploratory data analysis McKinney, W. (2017). The library has a key dependency of NumPy, another library that adds support for N-dimensional arrays and corresponding mathematical functions to operate on these arrays NumPy(2025). Furthermore, the projected is licenced and distributed under the three-clause Berkeley Software Distribution BSD license Pandas Development Team (2025).

• Open-Source. What open-source licence does the project use? Who can contribute to the repository? Who owns the repository? Does the project receive any funding from the industry or the community? Are there developer guidelines for new contributors?

• Joining the project community. How easy is it for a new developer to join the project community. What documentation is available for new developers? Are there contributing guidelines? Is there a development roadmap? How can a new developer start communicating with the rest of the community?

BSD licenses are a group of permissive licenses that impose minimal restrictions of the use and distribution of software. Developed in 1999, a ‘three clause BSD’ license mandates terms and conditions of redistribution (no liability of the authors), maintenance of copyright in redistribution and rights of use of contributors names in promotion/endorsement Montague (2007). Derived from the earlier ‘four clause BSD’, the more recent ‘three clause BSD’ dropped the advertisement requirement.

For comparison, another license is the general proprietary license (GPL). The goal of the GPL was introduced to prevent software from becoming proprietary. Compared to BSD, the GPL licence is a more layered license with clauses such as discretionary distribution/support costs that can be imposed by authors, resell restrictions and requirements that must be followed with associated patents i.e. everyone must be able to use the software Montague (2007).

By forking the repository and submitting a pull request, any programmer can contribute the project – so far there have been over 3,400 contributors .The maturity of the project means there is a comprehensive contribution guide on the official library website [Contributing to pandas — pandas 3.0.0.dev0+2022.gdc8401afea documentation](https://pandas.pydata.org/docs/dev/development/contributing.html). As of the date of this essay , there are approximately 3600 issues open on the GitHub (NumPy approximately 1,600 open issues, Matplotlib approximately 1,500 open issues) with the last pull request being authorised on the 21st March 2025.

Support for community contribution includes 5 moderators, named under the ‘contributor code of conduct’, that help to maintain the professionalism of the page as well as an official Slack where contributors can communicate. Furthermore, there are frequent community meetings and specific new contributor meetings that are held monthly. To conclude, community activity, the 45,000 stars on the repository, well organised communication channels and dedicated moderators indicate the project is in good health.

• Code quality. What code quality standards are being followed? How are these standards ensured in development? Are any automated tools being used to ensure code quality?

Code quality standards can be found in a 5,000 word document in the contribution guide. [Contributing to the code base — pandas 3.0.0.dev0+2022.gdc8401afea documentation](https://pandas.pydata.org/docs/dev/development/contributing_codebase.html#code-standards). From the contributors perspective there are tools that help ensure that additional development integrates well with the existing codebase. For example, ./ci/code\_checks.sh is a script that can be run to ensure imported modules and the formatting of doctests and docstrings is correct. Furthermore, ‘Pre-commit’ tests can be run to pre-emptively trigger any issues that may arise in continuous integration or that may be flagged at code-review. ‘Pre-commit’ uses git hooks to identify simple issues with the output of some of the tests seen below.

A screenshot of a computer program

AI-generated content may be incorrect.

Further recommended development processes include test-driven-development whereby tests are written before code. To support this requirement, the test is often included in the issue – additional test creation is also encouraged to help with the current code coverage deficit. Code coverage is applied on approximately 250,000 lines of code in the project. At the time of writing this essay, the level of coverage is currently at 85% Codecov (2025).

• Testing. Does the project use testing tools, such as unit testing libraries? Does the project provide guidelines for functional testing? Does the project have CI tools configured for automatic testing?

As the project is a python library, the unit test tool of choice is pytest and the documentation has a guide on how to formulate a test, test for warnings, test for exceptions, test for involving files and test for network connectivity. Performance testing is also being incorporated into pandas via asv benchmarks – a separate open source library.

Currently, Integration testing is performed by GitHub actions once a pull request has been submitted. Contributors can also perform this integration testing without a pull request.A screenshot of a email

AI-generated content may be incorrect.

• Documentation. How is the project documented? Does it have a wiki? Does it have a separate web resource for documentation? How easy is it for a new user to get started?

• Community Feedback. Does the project accept feedback from users and developers? Is there a way to report bugs and issues? How are those issues being dealt with?

References

[Codecov](https://app.codecov.io/gh/pandas-dev/pandas)