***Project description***

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.”[1]. In addition to offering the foundational DataFrame object, the library provides a group of methods for tasks ranging from machine learning data preprocessing to exploratory data analysis [2]. 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 [3]. Furthermore, the project is licenced and distributed under the three-clause Berkeley Software Distribution (BSD) license [1].

***Open source***

The most recent stable release was pandas 2.2.3 released on the 20th of September 2024 [1]. The production version of the code is distributed via Pip Installs Packages(Pip)[4]. Pip is recognised as an essential tool for working with Python libraries and dependencies and works by installing precompiled executable binary files onto the system[4]. The package is then stored in the ‘site-packages’ directory as standard source code (figure 1).

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 1: Pandas file path*

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 (authors cannot be sued), maintenance of copyright in redistribution and rights of use of contributors names in promotion/endorsement [5]. Derived from the earlier ‘four clause BSD’, the more recent ‘three clause BSD’ dropped the mandatory advertisement requirement.

In terms of the user community, 61 million users have downloaded the package on anaconda [1] the projects twitter account has 31k followers [6] and 800 members are active on the slack channel [7]. Sponsorship is the main way that the project is funded with individual sponsors and also sponsors from industry [8].

***Joining the community***

By forking the repository, cloning it and submitting a pull request, any programmer can potentially contribute the project – there has been over 3,400 contributors so far [1]. The maturity of the project means there is a comprehensive contribution guide on the official library website [Contributing to pandas](https://pandas.pydata.org/docs/dev/development/contributing.html) to help with this process[9]. As of the date of this report, there are approximately 3,600 issues open on the GitHub (NumPy approximately 2,000 open issues, Matplotlib approximately 1,200 open issues) with the last pull request being authorised on the 21st March 2025 [3],[10].

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 channel where contributors can communicate [1]. Furthermore, there are frequent community meetings (online and through events such as PyData), specific new contributor meetings and under the ‘Contributing to pandas’ section in the README there are helpful links such as ‘good first issue’. In terms of a roadmap, despite being behind schedule, there is a detailed contents for upcoming releases [11].

To conclude this section, community activity on GitHub/Twitter/Slack, the 45,000 stars on the GitHub repository [1], well organised communication channels and dedicated moderators indicate the project is in good health and accessible.

***Contributors***

​The pandas GitHub repository is owned by the pandas-dev organization[1]. Development is conducted openly and hosted in public GitHub repositories under the pandas-dev organization. The project is developed by a team of distributed contributors who participate by submitting, reviewing, and discussing GitHub pull requests and issues. The top contributor based on the number of commits is ‘jbrockmendal’ with 4,844 commits to date however contributions from companies such as anaconda and google have also been important in the projects development [1], [11].

***Code Quality***

Code quality standards can be found in a comprehensive document in the contribution guide [Contributing to the code base](https://pandas.pydata.org/docs/dev/development/contributing_codebase.html#code-standards)[9]. Pandas follows the PEP 8 style guide and makes use of automated tools such as Flake8 for code formatting [12]. From a 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[9]. 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 (figure 2) [12] .

Further recommended development processes include test-driven-development whereby tests are written before code[9]. To support this requirement, it is advised that unit tests are included in the issue – additional test creation is also encouraged to help with the current code coverage deficit. Code test coverage is applied on approximately 250,000 lines of code in the project[13]. At the time of writing this report, the level of coverage is at 85% [13]. To put 85% into context, the popular machine learning library ‘scikit-learn’ has around 110,000 lines of code with a test coverage of around 99% - It’s also worth noting that despite having less than half the amount of code, scikit-learn has managed to attract 3,000(only 400 less than pandas) contributors meaning a higher contributor to line of code ratio and hence, possibly, a better code coverage[14],[15].

A screenshot of a computer program

AI-generated content may be incorrect.

*Figure 2: Pre-commit tests*

***Testing***

Testing is a critical aspect of Pandas development. As the project is a python library, the unit test tool of choice is PyTest which is used for unit testing and regression testing[9]. There is a guide on how to formulate a test, test for warnings, test for exceptions, test for involving files and test for network connectivity [9]. Performance testing is also being incorporated into pandas via ‘asv benchmarks’ – a separate open source library [9], [16].

Currently, continuous integration testing is performed by GitHub actions once a pull request has been submitted (figure 3) ensuring a good standard in submission.. Contributors can also perform this integration testing manually without a pull request[9].

A screenshot of a email

AI-generated content may be incorrect.

*Figure 3: Continuous integration testing*

An area that is still in development would appear to be functional testing as it is not mentioned in the [Contributing to the code base](https://pandas.pydata.org/docs/dev/development/contributing_codebase.html" \l "code-standards) resource [9]. At the moment the issue backlog is split into ‘enhancements’ and ‘bugs’ types, however, issues have not been translated into user stories with acceptance criteria that can be functionally tested [17]. Closer inspection of the issue log also reveals that the test driven development is not always being followed as many of the issues lack an associated unit test [18].

Behind the scenes additional stress and security testing is likely to occur before release however there is no mention of this on the GitHub or website[1], [9].

In terms of reliability, in 2024, a google search revealed there were no significant availability issues with the package. However, in terms of security, as of December 2024 there is a security vulnerability with the ‘Dataframe.query’ method whereby it is vulnerable to injection attacks [19].

***Documentation***

Pandas provides extensive documentation to help users understand its features and functionality. Pandas GitHub has a README headed by a dashboard that provides information about the status of metrics such as code coverage, downloads and version (Figure 4)[1] . A summary of the library follows.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4: README dashboard

As mentioned previously, the project has a dedicated website to explain not only the functionality but also how to contribute to the project at varying levels[9]. This appears to be common among python libraries whereby it is the norm to service documentation via a website [20] . Presumably due to the website, the wiki is not present.

***Feedback***

A screenshot of a social media post

AI-generated content may be incorrect.

*Figure 5: Contributor feedback on Twitter*

Feedback from users is an essential part of the Pandas development cycle. The project has several channels for feedback including monthly meetings, Twitter and Slack [1],[6],[7]. As the project is open source and the pandas community is not an organisation , it’s difficult to establish if there are clear lines of accountability for features/parts of the project – for example it’s not easy to distinguish who to contact if there is a specific issue with the apply() method [1]. Instead, the project appears to be serviced by a group of highly skilled and dedicated developers whereby ownership is shared for the features. For this reason, as suggested in figure 5, the ethos/procedure appears to be; if you can envision an improvement, create an issue, wait for it to be triaged, and then the community will support its implementation. There are currently 1,125 code helpers that have helped triage issues [1].

***References (IEEE)***

[1] Pandas Development Team, "pandas," [Online]. Available: https://github.com/pandas-dev/pandas. [Accessed: Mar. 23, 2025].

[2] McKinney, W., Python for Data Analysis, 2nd ed., Sebastopol, CA, USA: O'Reilly Media, 2017.

[3] NumPy Developers, "NumPy," [Online]. Available: https://github.com/numpy/numpy. [Accessed: Mar. 23, 2025].

[4] Python Package Index, "pip," [Online]. Available: <https://pypi.org/project/pip/>. [Accessed: Mar. 23, 2025].

[5] Dahlander, L. and Magnusson, M., "Relationships between open source software companies and communities: Observations from Nordic firms," Technology Innovation Management Review, [Online]. Available: https://timreview.ca/article/67. [Accessed: Mar. 23, 2025].

[6] pandas\_dev, "pandas\_dev on X (formerly Twitter)," [Online]. Available: https://x.com/pandas\_dev. [Accessed: Mar. 23, 2025].

[7] Pandas Dev Community, "Pandas Dev Community Slack invite," [Online]. Available: https://pandas-dev-community.slack.com/join/shared\_invite/zt-2blg6u9k3-K6\_XvMRDZWeH7Id274UeIg#/shared-invite/email. [Accessed: Mar. 23, 2025].

[8] Pandas, "pandas sponsors," [Online]. Available: <https://pandas.pydata.org/about/sponsors.html#:~:text=pandas%20is%20a%20Sponsored%20Project,and%20sustainability%20of%20the%20project>. [Accessed: Mar. 23, 2025].

[9] Pandas Development Team, "Code standards for contributing to pandas," [Online]. Available: https://pandas.pydata.org/docs/dev/development/contributing\_codebase.html#code-standards. [Accessed: Mar. 23, 2025].

[10] Matplotlib Developers, "Matplotlib," [Online]. Available:

[11] "Authors," Pandas GitHub Repository, 2025. [Online]. Available: https://github.com/pandas-dev/pandas/blob/main/AUTHORS.md. [Accessed: 28-Mar-2025].

[12] https://pandas.pydata.org/docs/dev/whatsnew/v3.0.0.html

https://github.com/matplotlib/matplotlib. [Accessed: Mar. 23, 2025].

[13] "Code Quality Tools," Pandas Documentation, 2025. [Online]. Available: https://pandas.pydata.org/docs/dev/development/code\_quality.html. [Accessed: 28-Mar-2025].

[14] Pre-commit, "pre-commit framework," [Online]. Available: https://pre-commit.com/. [Accessed: Mar. 23, 2025].

[15] Codecov, "pandas Codecov Report," [Online]. Available: https://app.codecov.io/gh/pandas-dev/pandas. [Accessed: Mar. 23, 2025].

[16] Codecov, "scikit-learn Codecov Report," [Online]. Available: https://app.codecov.io/gh/scikit-learn/scikit-learn. [Accessed: Mar. 23, 2025].

[17] Scikit-learn Developers, "scikit-learn," [Online]. Available: https://github.com/scikit-learn/scikit-learn. [Accessed: Mar. 23, 2025].

[18] Airspeed Velocity, ASV Benchmarking Tool, GitHub, 2025. [Online]. Available: https://github.com/airspeed-velocity/asv. [Accessed: 28-Mar-2025].

[19] S.Ovchinnik , "Types of black-box testing."COMP8860 Software Engineering, [Online]. Available: <https://Kent> moodle.com [Accessed: Mar. 26, 2025].

[20] Pandas Development Team, "Issue #61160: [Accept no fields for groupby by

]," GitHub, [Online]. Available: https://github.com/pandas-dev/pandas/issues/61160. [Accessed: Mar. 27, 2025].

[21] Tenable, "Pandas 'DataFrame.query' arbitrary code execution (CVE-2024-42992)," [Online]. Available: https://www.tenable.com/plugins/nessus/213084. [Accessed: Mar. 24, 2025].

[22] Matplotlib, "Visualization with Python," [Online]. Available: https://matplotlib.org/. [Accessed: Mar. 23, 2025].