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

**[10 marks] Project description Provide a clear description of what the project is and what it’s used for.**

***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.” (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 Developers, 2025). Furthermore, the project 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?

***[15 marks] Open-source aspects How is the source code distributed? Are there stable releases? Is it available in non-source forms such as installers, executable binaries, etc.? What open-source licence is used? How big is the user community? How is the project funded?***

***Open source***

The most recent stable release was pandas 2.2.3 released on the 20th of September 2024 (Pandas Development Team, 2025). The production version of the code is distributed via Pip Installs Packages(Pip). 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. 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 : 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 (Dahlander, L. and Magnusson, M., 2005). 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 (Pandas Development Team, 2025), the twitter account has 31k followers (pandas\_dev, 2025) and 800 members are on the slack channel (Pandas Dev Community, 2025) . Sponsorship is the main way that the project is funded with individual sponsors and also sponsors from industry.

• 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?

***[15 marks] Joining the project community What documentation is there for new members to read? Is there a code of conduct and what is in it? Are there development guidelines, issue reporting guidelines, development roadmap? Is there a community area for people to discuss the project (forum, project website, discord server, etc.)?***

***Joining the community***

By forking the repository, cloning it and submitting a pull request, any programmer can contribute the project – there has been over 3,400 contributors so far. 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) (Pandas Development Team, 2025) . As of the date of this essay , there are approximately 3600 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. (GitHub, 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 (Pandas Development Team, 2025). Furthermore, there are frequent community meetings and specific new contributor meetings that are held monthly. To conclude, community activity on GitHub/Twitter/Slack, the 45,000 stars on the repository (Pandas Development Team, 2025), well organised communication channels and dedicated moderators indicate the project is in good health – although it is worth noting that there is currently no clear roadmap.

***[10 marks] Contributors Who is the project owner(s)? Who are the main contributors to the project? Who reviews and merges new pull requests? How many total contributors are there? How is the communication established between contributors?***

***Contributors***

​The pandas GitHub repository is owned by the pandas-dev organization. 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, as well as engaging in public project discussions on GitHub and other channels. The top contributors based on the number of commits is jbrockmendal with 4,844 commits to date (Pandas Development Team, 2025).

• 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?

***[15 marks] Code quality What code quality standard does the codebase follow? Is there documentation on project-specific code quality requirements? Is there automated code style checking?***

***Code Quality***

Code quality standards can be found in a comprehensive 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 (figure 2).

A screenshot of a computer program

AI-generated content may be incorrect.

Figure : Pre-commit tests

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 at 85% Codecov (2025). In terms of other popular python libraries, 85% represents ‘good’ code coverage. For comparison, the popular machine learning library ‘scikit-learn’ has around 110,000 lines of code with a coverage of around 99% Codecov (2025) - It’s also worth noting that despite having less than half the amount of code, scikit-learn has managed to attract 3,000 contributors (only 400 less than pandas).

***[15 marks] Testing How is the software being tested? Are there unit tests? Are there automated tests? Are there ways to run testing manually? Is there a CI tool that runs automated tests on new versions? Are there instructions for functional testing? Is there any evidence of security testing, stress testing, etc.?***

• 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?

***Testing***

As the project is a python library, the unit test tool of choice is Pytest. 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. Performance testing is also being incorporated into pandas via ‘asv benchmarks’ – a separate open source library.

Currently, continuous integration testing is performed by GitHub actions once a pull request has been submitted. Contributors can also perform this integration testing manually without a pull request.

A screenshot of a email

AI-generated content may be incorrect.

Figure : Continuous integration testing

An area that is still in development would appear to be functional testing. At the moment the reality is that the issue backlog is split into ‘enhancements’ and ‘bugs’ type, however they have not been translated into user stories with acceptance criteria that can be functionally tested. 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. In general, the issue log appears to be quite intimidating and unwieldy with no clear prioritisation or road map.

As this is only a public repository, behind the scenes additional stress and security testing is likely to occur before the release. Its worth noting that in 2024 there was no availability issues reported with the package however as of December 2024 there is a security vulnerability with the ‘Dataframe.query’ method whereby it is vulnerable to injection attacks (Tenable, 2025).

***[10 marks] Documentation How well is the software documented? Is there a detailed README page? Is there a wiki and how detailed is it? Is there any external documentation? How easy is it for a new user to start using the software? How easy is it for an experienced user to learn about advanced functionalities?***

• 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?

***Documentation***

Pandas has a README headed by a dashboard that provides information about the status of metrics such as code coverage, downloads and version. A summary of the package then follows.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : 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 on entry. This appears to be common among python libraries whereby it the cultural norm is to service documentation via a website. Due to the website, the wiki is not present.

***[10 marks] Feedback How can users provide feedback to the developers? Is there a way to submit bug reports, feature requests, usability issues, etc.? Are there special ways to submit feedback on sensitive topics, such as security vulnerabilities? How much feedback is received from the user community? How is this feedback being addressed?***

• 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?

***Feedback***

A screenshot of a social media post

AI-generated content may be incorrect.

Figure : Contributor feedback on Twitter

The project has several channels for feedback including monthly meetings, Twitter and Slack. As the project is open source and pandas is not a product, it’s hard to establish if there are clear lines of accountability for features/ parts of the project – who ‘owns’ the apply() method for example? It’s more likely that it’s a perpetual work in progress serviced by a group highly skilled and dedicated developers that make things happen. For this reason, as suggested in figure 5, the ethos appears to be if you can envision an improvement, then do it yourself and people will support you. It’s a credit to the quality of pandas that users believe it’s structured like an organisation.

***References***

Codecov (n.d.) pandas Codecov Report [Online]. Available at: https://app.codecov.io/gh/pandas-dev/pandas (Accessed: 23 March 2025).

Codecov (n.d.) scikit-learn Codecov Report [Online]. Available at: https://app.codecov.io/gh/scikit-learn/scikit-learn (Accessed: 23 March 2025).

Dahlander, L. and Magnusson, M. (2005) 'Relationships between open source software companies and communities: Observations from Nordic firms', Technology Innovation Management Review. Available at: https://timreview.ca/article/67 (Accessed: 23 March 2025).

Matplotlib Developers (n.d.) Matplotlib [Online]. Available at: https://github.com/matplotlib/matplotlib (Accessed: 23 March 2025).

NumPy Developers (n.d.) NumPy [Online]. Available at: https://github.com/numpy/numpy (Accessed: 23 March 2025).

pandas\_dev (n.d.) pandas\_dev on X (formerly Twitter) [Online]. Available at: https://x.com/pandas\_dev (Accessed: 23 March 2025).

Pandas Dev Community (n.d.) *Pandas Dev Community Slack invite* [Online]. Available at: <https://pandas-dev-community.slack.com/join/shared_invite/zt-2blg6u9k3-K6_XvMRDZWeH7Id274UeIg#/shared-invite/email> (Accessed: 23 March 2025).

Pandas Development Team (n.d.) pandas [Online]. Available at: https://github.com/pandas-dev/pandas (Accessed: 23 March 2025).

Pandas Development Team (n.d.) Code standards for contributing to pandas [Online]. Available at: https://pandas.pydata.org/docs/dev/development/contributing\_codebase.html#code-standards (Accessed: 23 March 2025).

Pre-commit (n.d.) pre-commit framework [Online]. Available at: https://pre-commit.com/ (Accessed: 23 March 2025).

Scikit-learn Developers (n.d.) scikit-learn [Online]. Available at: https://github.com/scikit-learn/scikit-learn (Accessed: 23 March 2025).

Tenable (2024) Pandas 'DataFrame.query' arbitrary code execution (CVE-2024-42992). Available at: https://www.tenable.com/plugins/nessus/213084 (Accessed: 24 March 2025).

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

***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.” (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 Developers, 2025). Furthermore, the project 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?

***Open source***

The most recent stable release was pandas 2.2.3 released on the 20th of September 2024 (Pandas Development Team, 2025). The production version of the code is distributed via Pip Installs Packages(Pip). 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. The package is then stored in the ‘site-packages’ directory as standard source code (figure 1).

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 (Dahlander, L. and Magnusson, M., 2005). 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 (Pandas Development Team, 2025), the twitter account has 31k followers (pandas\_dev, 2025) and 800 members are on the slack channel (Pandas Dev Community, 2025) . Sponsorship is the main way that the project is funded with individual sponsors and also sponsors from industry.

***Joining the community***

By forking the repository, cloning it and submitting a pull request, any programmer can contribute the project – there has been over 3,400 contributors so far. 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) (Pandas Development Team, 2025) . As of the date of this essay , there are approximately 3600 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. (GitHub, 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 (Pandas Development Team, 2025). Furthermore, there are frequent community meetings and specific new contributor meetings that are held monthly. To conclude, community activity on GitHub/Twitter/Slack, the 45,000 stars on the repository (Pandas Development Team, 2025), well organised communication channels and dedicated moderators indicate the project is in good health – although it is worth noting that there is currently no clear roadmap.

***Contributors***

​The pandas GitHub repository is owned by the pandas-dev organization. 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, as well as engaging in public project discussions on GitHub and other channels. The top contributors based on the number of commits is jbrockmendal with 4,844 commits to date (Pandas Development Team, 2025).

***Code Quality***

Code quality standards can be found in a comprehensive 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 (figure 2).

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 at 85% Codecov (2025). In terms of other popular python libraries, 85% represents ‘good’ code coverage. For comparison, the popular machine learning library ‘scikit-learn’ has around 110,000 lines of code with a coverage of around 99% Codecov (2025) - It’s also worth noting that despite having less than half the amount of code, scikit-learn has managed to attract 3,000 contributors (only 400 less than pandas).

***Testing***

As the project is a python library, the unit test tool of choice is Pytest. 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. Performance testing is also being incorporated into pandas via ‘asv benchmarks’ – a separate open source library.

Currently, continuous integration testing is performed by GitHub actions once a pull request has been submitted. Contributors can also perform this integration testing manually without a pull request.

An area that is still in development would appear to be functional testing. At the moment the reality is that the issue backlog is split into ‘enhancements’ and ‘bugs’ type, however they have not been translated into user stories with acceptance criteria that can be functionally tested. 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. In general, the issue log appears to be quite intimidating and unwieldy with no clear prioritisation or road map.

As this is only a public repository, behind the scenes additional stress and security testing is likely to occur before the release. Its worth noting that in 2024 there was no availability issues reported with the package however as of December 2024 there is a security vulnerability with the ‘Dataframe.query’ method whereby it is vulnerable to injection attacks (Tenable, 2025).

***Documentation***

Pandas has a README headed by a dashboard that provides information about the status of metrics such as code coverage, downloads and version. A summary of the package then follows.

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 on entry. This appears to be common among python libraries whereby it the cultural norm is to service documentation via a website. Due to the website, the wiki is not present.

The project has several channels for feedback including monthly meetings, Twitter and Slack. As the project is open source and pandas is not a product, it’s hard to establish if there are clear lines of accountability for features/ parts of the project – who ‘owns’ the apply() method for example? It’s more likely that it’s a perpetual work in progress serviced by a group highly skilled and dedicated developers that make things happen. For this reason, as suggested in figure 5, the ethos appears to be if you can envision an improvement, then do it yourself and people will support you. It’s a credit to the quality of pandas that users believe it’s structured like an organisation.

***References***

Codecov (n.d.) pandas Codecov Report [Online]. Available at: https://app.codecov.io/gh/pandas-dev/pandas (Accessed: 23 March 2025).

Codecov (n.d.) scikit-learn Codecov Report [Online]. Available at: https://app.codecov.io/gh/scikit-learn/scikit-learn (Accessed: 23 March 2025).

Dahlander, L. and Magnusson, M. (2005) 'Relationships between open source software companies and communities: Observations from Nordic firms', Technology Innovation Management Review. Available at: https://timreview.ca/article/67 (Accessed: 23 March 2025).

Matplotlib Developers (n.d.) Matplotlib [Online]. Available at: https://github.com/matplotlib/matplotlib (Accessed: 23 March 2025).

NumPy Developers (n.d.) NumPy [Online]. Available at: https://github.com/numpy/numpy (Accessed: 23 March 2025).

pandas\_dev (n.d.) pandas\_dev on X (formerly Twitter) [Online]. Available at: https://x.com/pandas\_dev (Accessed: 23 March 2025).

Pandas Dev Community (n.d.) *Pandas Dev Community Slack invite* [Online]. Available at: <https://pandas-dev-community.slack.com/join/shared_invite/zt-2blg6u9k3-K6_XvMRDZWeH7Id274UeIg#/shared-invite/email> (Accessed: 23 March 2025).

Pandas Development Team (n.d.) pandas [Online]. Available at: https://github.com/pandas-dev/pandas (Accessed: 23 March 2025).

Pandas Development Team (n.d.) Code standards for contributing to pandas [Online]. Available at: https://pandas.pydata.org/docs/dev/development/contributing\_codebase.html#code-standards (Accessed: 23 March 2025).

Pre-commit (n.d.) pre-commit framework [Online]. Available at: https://pre-commit.com/ (Accessed: 23 March 2025).

Scikit-learn Developers (n.d.) scikit-learn [Online]. Available at: https://github.com/scikit-learn/scikit-learn (Accessed: 23 March 2025).

Tenable (2024) Pandas 'DataFrame.query' arbitrary code execution (CVE-2024-42992). Available at: https://www.tenable.com/plugins/nessus/213084 (Accessed: 24 March 2025).

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