The Journal of Open Source Software

exoatlas: Friendly Python Code for Exoplanet Populations

Zach K. Berta-Thompson [1], **Patcharapol Wachiraphan** [1], **Autumn Stephens** [1], **Mirielle Caradonna** [1], **Catriona A. Murray** [1], **Valerie Arriero** [1], **Jackson Avery** [1], **Girish M. Duvvuri** [2], **and J. Sebastian Pineda** [3]

1 Department of Astrophysical and Planetary Sciences, University of Colorado Boulder, USA **2** Department of Physics and Astronomy, Vanderbilt University, USA **3** Laboratory for Atmospheric and Space Physics, University of Colorado Boulder, USA

DOI: 10.21105/joss.09655

Software

- Review
- Repository
- Archive

Editor: Ivelina Momcheva

Reviewers:

- @LorenzoMugnai
- @tedjohnson12

Submitted: 01 July 2025
Published: 08 November 2026

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/1538-4357/ae77fd, published in the The Astrophysical Journal.

Summary

Planets are complicated. Understanding how they work requires connecting individual objects to the context of broader populations. Exoplanets are easier to picture next to their closest Solar System archetypes, and planets in the Solar System are richer when seen alongside a growing community of known exoplanets in the Milky Way. The `exoatlas` toolkit provides a friendly Python interface for retrieving and working with populations of planets, aiming to simplify the process of placing worlds in context.

Statement of need

`exoatlas` was designed to meet a need among both researchers and educators for an intuitive Python tool to access planet populations. Particularly in working with students and junior scientists, for whom easy avenues for exploration and play would have particular benefit, significant barriers often frustrate attempts to perform the following tasks:

- retrieving basic properties for exoplanets and Solar System objects
- calculating derived planet quantities with propagated uncertainties
- comparing individual exoplanets to relevant comparison samples
- planning future telescope observations of known exoplanet systems
- creating beautiful and up-to-date planet data visualizations

Online planetary data archives merge incredible curatorial efforts with powerful tools for data access and visualization. These archives include the NASA Exoplanet Archive (Christiansen et al., 2025), exoplanet.eu, exo.MAST, the University of Geneva Data Analysis Center for Exoplanets (Buchschacher & Alesina, 2019), TEPCat (Southworth, 2011), and the Open Exoplanet Catalog (Rein, 2012) for exoplanets, and the JPL Solar System Dynamics, IAU Minor Planet Center, and the NSSDCA Planetary Facts sheets for Solar System objects. Through web portals, many of these archives enable the user to find or quickly create up-to-date plots of planetary properties. Through downloadable tables and/or application programming interfaces (APIs), they can provide raw or curated planetary properties with clear citations to the scientific literature. Drawing from these and other archives, tools like NASA's Eyes and Stellarium (Zotti et al., 2021) allow the easy visualization of planetary orbits, and tools like Tapir (Jensen, 2013), the Exoplanet Archive Transit and Ephemeris Service (Christiansen et al., 2025), and VarAstro help plan new exoplanet observations.

`exoatlas` does not intend to replace any of these important archival efforts, web visualization

portals, or planning tools. The main gap that `exoatlas` aims to address is to support new calculations and creative visualizations by users who are fluent in Python but uninterested in carefully parsing complicated tables, calculating replacements for missing quantities, or manually propagating uncertainties into derived quantities. By trying to do a good job of these steps behind the scenes, `exoatlas` aims to provide an approachable interface for exploratory analysis, centering a strong focus on scientific storytelling and illuminating visualizations. The target audience is the scientist, educator, or student who wants to curiously explore one step beyond a web plot, to playfully interact with exoplanet and Solar System data in Python, to fussily adjust how data are filtered or plotted, and/or to easily integrate new planet discoveries into their workflow.

Mapping populations with `exoatlas`

The user interface for `exoatlas` centers around the Python `Population` class, with each `Population` object containing a standardized table of planet properties and methods for interacting with that table. `exoatlas` makes extensive use of Astropy (Astropy Collaboration, 2026) to be as familiar as possible for modern astronomers, and it is thoroughly documented with astronomical audiences in mind.

- To retrieve planet data, `exoatlas` provides `Population` objects that automatically access archive data for exoplanets as well as Solar System major planets, minor planets, and moons. Exoplanet data come from the NASA Exoplanet Archive (Christiansen et al., 2025) API, and Solar System data come from the JPL Solar System Dynamics API or small reformatted data tables included in the package repository itself. Whatever the original data source, all `Population` objects act similarly and have uniform nomenclature for accessing data columns. Planet quantities all have physical units attached with `astropy.units` to facilitate unit conversions and minimize conceptual errors. New `Population` objects can also be created from `astropy.table` tables, enabling custom datasets to be included.
- To calculate derived quantities for planets, a set of default methods is included within the core `Population` definition, or users may attach their own calculation methods. If data for a quantity is missing, calculations can be used to swap in alternate estimates; for example, a planet's semimajor axis a will attempt first to pull from the original data table, and then second to calculate a from the planet's period P and the star's mass $M_\star$ assuming Newton's Version of Kepler's Third Law $P^2 = 4\pi^2 a^3 / G M_\star$, and then third to calculate from a transit-derived ratio a/R_*. Uncertainties on all derived quantities can be numerically propagated using the `astropy.uncertainty` framework, where distributions of samples are generated for each original table quantity, carried through calculations, and then used to estimate confidence intervals.
- To extract sets of planets meeting particular criteria, `Population` objects can be indexed, sliced, and masked to generate new smaller `Population` objects. Coupling familiar array or table operations into the creation of subpopulations enables nuanced filtering of datasets based on any combination of original archival table quantities, derived quantities, and/or quantity uncertainties.
- To plan telescope observations of known exoplanet systems, `exoatlas` can estimate the signal-to-noise (S/N) ratio achievable for exoplanet observables. As derived quantities, all S/N estimates can include propagated uncertainties, to help avoid biasing target samples toward systems with larger uncertainties. Using `astroplan` (Morris et al., 2018) and `astropy.coordinates` (Astropy Collaboration, 2026), `exoatlas` can determine local altitudes and azimuths for all elements in a `Population`, as well as upcoming opportunities to observe transits from ground-based telescopes (when the star is above the horizon, the Sun is down, and the planet is passing in front of the star). Including basic observation planning tools enables a workflow where targets can be filtered both by the estimated detectability of a signal and by whether a telescope can actually point at the system.
- To make explanatory illustrations of planet data, `exoatlas` includes a visual language to

build up population comparison plots. The core elements of this visual language are the `Plottable` (a quantity that should be represented with certain scaling, limits, and labels), the `Map` (a panel expressing plottable quantities with position, size, or color), and the `Gallery` (a collection of maps with linked axes and/or datasets). Preset visualizations built from this language can offer quick contextual reference for a planet's fundamental properties, as in Figure 1, which was generated with only six lines of code:

```
from exoatlas import *
from exoatlas.visualizations import *

e = TransitingExoplanets()
s = SolarSystem()
h = e["HD209458b"]

PlanetGallery().build([e, s, h])
```

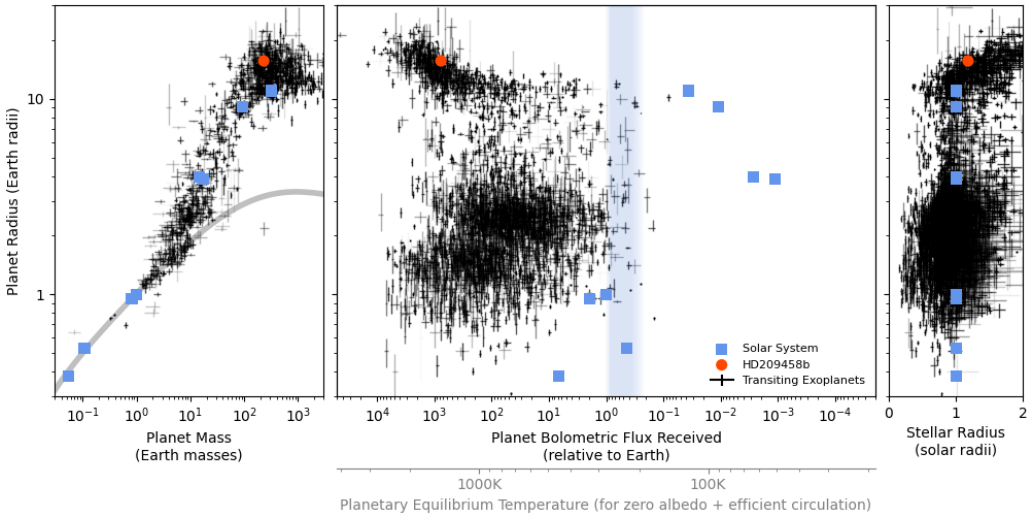


Figure 1: Example `exoatlas` visualization placing the first discovered transiting exoplanet HD209458b in context with other transiting exoplanets and the eight major Solar System planets. Errorbars use a color intensity that scales inversely with quantity uncertainties, to avoid giving undue visual weight to the least precise data.

Research, teaching, and learning

`exoatlas` was designed to support the researcher who wants to contextualize planet populations in papers, proposals, or talks, the educator who wants help connect lessons more immediately to real exoplanet and Solar System data, and the student who simply wants to learn a little more about oodles of neat planets. All these communities are encouraged try `exoatlas`, to ask for help, to suggest improvements, and/or to contribute code.

AI usage disclosure

No generative AI tools were used in the development of this software, the writing of this manuscript, or the preparation of supporting materials.

Acknowledgements

We acknowledge the long commitment federally-funded archives have made to preserving and sharing data with the scientific community, and the heroic efforts of the people who build, maintain, and continually improve those archives. This material is based upon work supported by the National Science Foundation under Grant No. 1945633.

This work made use of the following software packages: Astropy (Astropy Collaboration et al., 2013, 2018, 2022; Astropy Collaboration, 2026), Jupyter (Kluyver et al., 2016; Perez & Granger, 2007), Matplotlib (Hunter, 2007), pandas (McKinney, 2010; The pandas development team, 2026), Python (Van Rossum & Drake, 2009), SciPy (Gommers et al., 2026; Virtanen et al., 2020), ArviZ (Kumar et al., 2019; Martin et al., 2026), Astroquery (Ginsburg et al., 2019, 2025), `tqdm` (Costa-Luis et al., 2026), and Xarray (Stephan Hoyer et al., 2025; S. Hoyer & Hamman, 2017). Software citation information aggregated using the Software Citation Station (Wagg et al., 2025; Wagg & Broekgaarden, 2024).

This research has made use of data and/or services provided by the International Astronomical Union's Minor Planet Center. This research has made use of data obtained from or tools provided by the portal exoplanet.eu of The Extrasolar Planets Encyclopaedia. This research has made use of the exo.MAST service at the Mikulski Archive for Space Telescopes (MAST), operated by the Space Telescope Science Institute (STScI). STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS5-26555.

References

Astropy Collaboration. (2026). *Astropy* (Version v8.0.1). Zenodo. https://doi.org/10.5281/zenodo.21262391

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., N"othe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., … Astropy Project Contributors. (2022). The Astropy Project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package. *The Astrophysical Journal*, *935*(2), 167. https://doi.org/10.3847/1538-4357/ac7c74

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., Vand erPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., … Astropy Contributors. (2018). The Astropy Project: Building an open-science project and status of the v2.0 core package. *The Astronomical Journal*, *156*(3), 123. https://doi.org/10.3847/1538-3881/aabc4f

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., … Streicher, O. (2013). Astropy: A community Python package for astronomy. *Astronomy and Astrophysics*, *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Buchschacher, N., & Alesina, F. (2019). DACE: New available visualisation and analysis tools for exoplanet research. In M. Molinaro, K. Shortridge, & F. Pasian (Eds.), *Astronomical Data Analysis Software and Systems XXVI* (Vol. 521, p. 757).

Christiansen, J. L., McElroy, D. L., Harbut, M., Ciardi, D. R., Crane, M., Good, J., Hardegree-Ullman, K. K., Kesseli, A. Y., Lund, M. B., Lynn, M., Muthiar, A., Nilsson, R., Oluyide, T., Papin, M., Rivera, A., Swain, M., Susemiehl, N. D., Tam, R., van Eyken, J., & Beichman, C. (2025). The NASA Exoplanet Archive and exoplanet follow-up observing program: Data, tools, and usage. *The Planetary Science Journal*, *6*(8), 186. https:

//doi.org/10.3847/PSJ/ade3c2

Costa-Luis, C. da, Larroque, S. K., Altendorf, K., Mary, H., richardsheridan, Korobov, M., Raphael, N., Ivanov, I., Bargull, M., Rodrigues, N., Shawn, Dektiarev, M., Górny, M., mjstevens777, Pagel, M. D., Zugnoni, M., CrazyPython, Newey, C., Lee, A., … Rothberg, A. (2026). *tqdm: A fast, extensible progress bar for Python and CLI* (Version v4.70.0). Zenodo. https://doi.org/10.5281/zenodo.21624223

Ginsburg, A., Sipőcz, B. M., Brasseur, C. E., Cowperthwaite, P. S., Craig, M. W., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lian Lim, P., Lockhart, K. E., Mommert, M., Morris, B. M., Norman, H., Parikh, M., Persson, M. V., Robitaille, T. P., Segovia, J.-C., Singer, L. P., … a subset of the astropy collaboration. (2019). astroquery: An astronomical web-querying package in Python. *The Astronomical Journal*, *157*, 98. https://doi.org/10.3847/1538-3881/aafc33

Ginsburg, A., Sipőcz, B., Brasseur, C. E., Parikh, M., jcsegovia, Groener, A., Norman, H., derdon, Marchand, M., Kelley, M., Bianco, S., Robitaille, T., Lim, P. L., Vaher, E., Deil, C., Mommert, M., Medina, J. V., Nilsson, R., Tollerud, E., … Mesh, C. (2025). *astropy/astroquery: v0.4.11* (Version v0.4.11). Zenodo. https://doi.org/10.5281/zenodo.17163526

Gommers, R., Virtanen, P., Haberland, M., Burovski, E., Reddy, T., Weckesser, W., Nelson, A., Oliphant, T. E., Cournapeau, D., Polat, I., alexbrc, Roy, P., Peterson, P., Colley, L., Wilson, J., endolith, Bowhay, J., Mayorov, N., Steppi, A., … Schmitz, D. (2026). *scipy/scipy: SciPy 1.18.0rc1* (Version v1.18.0rc1). Zenodo. https://doi.org/10.5281/zenodo.20383145

Hoyer, S., & Hamman, J. (2017). xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Hoyer, Stephan, Roos, M., Joseph, H., Magin, J., Cherian, D., Fitzgerald, C., Hauser, M., Fujii, K., Maussion, F., Imperiale, G., Clark, S., Kleeman, A., Nicholas, T., Kluyver, T., Westling, J., Munroe, J., Amici, A., Barghini, A., Banihirwe, A., … Littlejohns, O. (2025). *xarray* (Version v2025.12.0). Zenodo. https://doi.org/10.5281/zenodo.17834519

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jensen, E. (2013). *Tapir: A web interface for transit/eclipse observability*. Astrophysics Source Code Library, record ascl:1306.007.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). *Jupyter Notebooks – a publishing format for reproducible computational workflows* (F. Loizides & B. Schmidt, Eds.; pp. 87–90). IOS Press.

Kumar, R., Carroll, C., Hartikainen, A., & Martin, O. (2019). ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, *4*(33), 1143. https://doi.org/10.21105/joss.01143

Martin, O. A., Abril-Pla, O., Deklerk, J., Axen, S. D., Carroll, C., Hartikainen, A., & Vehtari, A. (2026). *ArviZ: A modular and flexible library for exploratory analysis of Bayesian models* (Version v1.2.0). Zenodo. https://doi.org/10.5281/zenodo.20670650

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in science conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Morris, B. M., Tollerud, E., Sipőcz, B., Deil, C., Douglas, S. T., Berlanga Medina, J., Vyhmeister, K., Smith, T. R., Littlefair, S., Price-Whelan, A. M., Gee, W. T., & Jeschke, E. (2018). astroplan: An open source observation planning package in Python. *The Astronomical Journal*, *155*(3), 128. https://doi.org/10.3847/1538-3881/aaa47e

Perez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, *9*(3), 21–29. https://doi.org/10.1109/MCSE.2007.53

Rein, H. (2012). A proposal for community driven and decentralized astronomical databases and the Open Exoplanet Catalogue. *arXiv e-Prints*, arXiv:1211.7121. https://doi.org/10.48550/arXiv.1211.7121

Southworth, J. (2011). Homogeneous studies of transiting extrasolar planets - IV. Thirty systems with space-based light curves. *Monthly Notices of the Royal Astronomical Society*, *417*(3), 2166–2196. https://doi.org/10.1111/j.1365-2966.2011.19399.x

The pandas development team. (2026). *pandas-dev/pandas: pandas* (Version v3.0.5). Zenodo. https://doi.org/10.5281/zenodo.21500199

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace. ISBN: 1441412697

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wagg, T., & Broekgaarden, F. S. (2024). Streamlining and standardizing software citations with The Software Citation Station. *arXiv e-Prints*, arXiv:2406.04405. https://arxiv.org/abs/2406.04405

Wagg, T., Broekgaarden, F., Van-Lane, P., Wu, K., & Gültekin, K. (2025). *TomWagg/software-citation-station: v1.4* (Version v1.4). Zenodo. https://doi.org/10.5281/zenodo.17654855

Zotti, G., Hoffmann, S. M., Wolf, A., Chéreau, F., & Chéreau, G. (2021). Simulated Sky: Stellarium for cultural astronomy research. *Journal of Skyscape Archaeology*, *6*(2). https://doi.org/10.1558/jsa.17822