Scientific
Research
Publishing

pysky: An Application for the Planning of Multi-Target Astronomical Observations

Allen E. Rocha, William D. Parker

Department of Mathematics and Physics, University of Wisconsin-Parkside, Kenosha, USA
Email: rocha021@rangers.uwp.edu, parkerw@uwp.edu

Abstract

To automate the process of planning and curating multi-target observation sessions, pysky application builds on the astroplan Python package to identify visible objects during an observation window and produce relevant information about those objects in visual and graphical form. The package calculates object visibility based on a provided time window and observing location as well as maximum airmass and limiting magnitude requested by the user. The pysky application images of the target objects with identifying and astrometric data to provide context for the images. In addition, pysky creates polar plots of each object's horizontal coordinates, and the images and plots are designed to be shown side-by-side. The package also generates an HTML table of the selected target objects with their related data to relay the entire target list as one. The pysky application draws on a variety of Python packages to collect and process data from databases such as JPL Horizons and SIMBAD. Results for a test event were verified by hand using database web interfaces. The pysky application provides a platform for further integration of automated observation planning with websites and apps to enhance multi-target observation sessions.

1. Introduction

Astronomical observation of specified objects requires advance planning to identify the objects available during the time frame of observation. Hand calculation to project object availability to the observer has been replaced by computer-based calculation, and, in the astropy universe [1] [2], the astroplan package provides this functionality [3]. In many contexts, an observation session may

focus on an array of objects to observe within a scheduled timeframe rather than a single specific target, and judging which targets to include may be aided by the relative astrometry and visualization of the target objects.

For educational observational programming, the program designer usually seeks to curate an array of observation targets to present to program participants, and visualization is an important component to support participants' astronomical learning [4]. Presenting astrometric data may supplement prior knowledge, which has been shown to increase awe during science learning experiences [5], and, by making the gaps in knowledge salient, awe has been shown to promote scientific investigation [6], an important goal of observatories offering public programming.

The pysky application seeks to fill the need for multi-target observation planning and curation by: 1) automating the identification of available observation targets within a time window given observational constraints; 2) collecting pertinent data about the targets; 3) formatting and visualizing these data for session planning as well as public event promotion and presentation.

2. System Overview

The pysky source code is hosted on GitHub[1]. After the pysky application is installed, the user edits a preferences file to specify an observing latitude, longitude, and elevation, minimum apparent magnitude, maximum air mass, and particular solar system objects and stellar systems that the user wishes to check visibility for. By default, the application searches all of the Messier [7] and Caldwell [8] catalog objects for visibility within specified observing parameters.

The user interacts with the application via their terminal or command prompt by invoking "pysky" with a required start date and start time in ISO 8601 format to query their specified celestial bodies as well as the default catalogs. If the ending day or ending time is not specified, the application will set the observation time to be for one hour after the specified starting time. The user may also specify a multi-threaded model to speed up the calculation (see Section 1 for results).

Using astroplan [3], the application downloads and caches the IERS Bulletin A to accurately predict the apparent position of a celestial object from the Earth. Next, the application loads the predefined dictionaries of Messier and Caldwell objects. For solar system system objects defined by the user, the application queries JPL Horizons [9] to retrieve and store the objects' right ascension, declination, azimuth, elevation, magnitude, constellation, and illumination. Next, the application makes a request to SkyView [10] to retrieve the image of the object using the beautifulsoup4, urllib3, and requests libraries for Python. For the non-solar objects, the application uses astroquery [11] to retrieve the object's astrometry, brightness, and classification from SIMBAD ([12]). The application then uses astroplan [3] and astropy [1] [2] to compare the object's airmass and

---

[1]GitHub: http://github.com/allenerocha/pysky, static Zenodo archive: https://doi.org/10.5281/zenodo.5262732.

apparent magnitude to the user's specified cutoffs and eliminate disqualified objects from the observation list.

Finally, the application generates outputs for each of the visible objects. The first output (if generated) is the image of the object overlaid with astrometry and brightness information (see **Figures 2-5**). This information is retrieved from JPL Horizons [9] for solar-system objects or from astroquery [11] for extrasolar objects. The images of stars and star systems are retrieved from SkyView [10] using Python's Pillow package. The next output generated is the HTML table containing each object's name, classification, astrometry, and brightness as shown in **Table 1**. The final output generated is a polar scatter plot for each object using astroplan [3], astropy [1] [2], and NumPy. One point is plotted for every 15 minutes in the observation time (see **Figures 2-5**). All figures and plots are saved with the HTML table in one directory for convenient display of the images in a slideshow.

Table 1. Table of select objects of apparent magnitude $V > 5.0$ and zenith angle $\sec(z) < 11.0$ on 2020 December 21 22:00:00 UT to 2020 December 22 00:00:00 UT viewed from the Hawthorn Hollow Schoolyard Observatory in Kenosha, Wisconsin, USA. Table highlights information about potential viewing objects for scrutiny by observatory users. Object-type descriptions, constellation, brightness, and distance values for star systems and deep-sky objects are pulled from the SIMBAD database using astroquery. The same values are retrieved from JPL Horizons using astroplan, and the starting and ending horizontal coordinates for all objects are calculated using astroplan.

Name	Type	Start Alt. (°)	Start Az. (°)	End Alt. (°)	End Az. (°)	Constellation	Brightness	Distance (Pm)
Achird	High Proper-Motion Star	61	44	74	18	Cassiopeia	3.44	183
Albireo	Double Or Multiple Star	55	259	33	280	Cygnus	3.08	4108
Capella	Spectroscopic Binary	20	45	37	58	Auriga	0.08	404
Deneb	Evolved Supergiant Star	74	287	53	293	Cygnus	1.25	13357
Jupiter	Planet	21	210	6	235	Capricornus	−2.0	0.00088673
Mars	Planet	37	114	53	149	Pisces	−0.5	0.00012113
Mizar A	Spectroscopic Binary	16	334	9	350	Ursa Major	2.22	783
Neptune	Planet	39	158	40	198	Aquarius	7.8	0.00450591
Polaris	Classical Cepheid	43	1	43	1	Ursa Minor	2.02	4092
Saturn	Planet	21	210	6	235	Capricornus	0.7	0.00161996
Uranus	Planet	30	99	50	127	Aries	5.7	0.00286838
Vega	Variable Star Of Delta Sct Type	50	283	29	297	Lyra	0.03	236
Moon	Satellite (Phase: First Quarter)	35	144	42	181	Pisces	−10.1	0.0000004

Continued

M31	Spiral Galaxy	62	80	84	100	Andromeda	3.4	23,000,000
M45	Open Cluster	20	75	42	95	Taurus	1.6	3700
M52	Open Cluster	68	23	70	344	Cassiopeia	6.9	47,000
NGC 869	Open Cluster	49	47	65	43	Perseus	3.7	69,000
NGC 884	Open Cluster	49	47	65	43	Perseus	3.8	69,000
NGC 7000	Nebula	77	284	55	290	Cygnus	4.0	17,000
Mel 25	Open Cluster	8	76	30	96	Taurus	0.5	1400

2020-12-21T22:00Z/2020-12-22T00:00Z. From Latitude: +42.6499, Longitude: −87.8791, Elevation: 0.204 km, Min V: 5.0, sec(z) max: 11.0.

Figure 1 depicts system overview described above. Parallelization over threads is implemented for JPL Horizons and SIMBAD queries, SkyView image retrieval, visibility cutoff evaluation, image overlays and plot generation. In each case, the total number of objects processed will be divided by the number of threads requested. Tests of the efficiency of thread parallelization are described in Section 1.

3. Main Results

For the following experiments, we choose an observation time frame that includes the Great Conjunction of 2020 from 2020 December 21 at 22:00:00 UT to 2020 December 22 at 00:00:00 UT, observed from the Schoolyard Observatory at Hawthorn Hollow Nature Preserve and Arboretum[2] in Kenosha, Wisconsin, USA, located at −87.8791˚ +42.6499˚. We choose a minimum apparent magnitude threshold of 5.0 and a maximum sec z threshold of 11.0. The astronomical objects parsed for each experiment are: all Messier objects, all Caldwell catalogue objects, the eight solar planets, and a selection of bright and multiple star systems: Polaris, Mizar A, Vega, Sirius, Capella, Procyon, Castor, Albireo, 145 CMa, Achird, and ι Cnc. Parallelization tests were performed on three different machines, increasing the number of threads and repeating the calculation. At the end of the run, the application produces a sky plot for each object's movement during the observation time window, an image of the object with its properties overlayed, and a row in a generated HTML table.

Figures 2-5 show the resulting overlaid images and plots under the conditions described above for a prototypical object in each major category: solar system (Jupiter), deep sky (Andromeda Galaxy), stellar (Polaris), and the Moon. The image of Jupiter was taken from the Hubble Space Telescope and scaled to approximate the apparent diameter seen in the observatory telescope. The image of Andromeda Galaxy and all deep-sky objects, as well as the Moon phase images, are sourced from Wikimedia Commons. The images of star systems such as Polaris come from the Mellinger Optical Survey ([13]).

The astrometry data overlaid on each image is designed to place the object
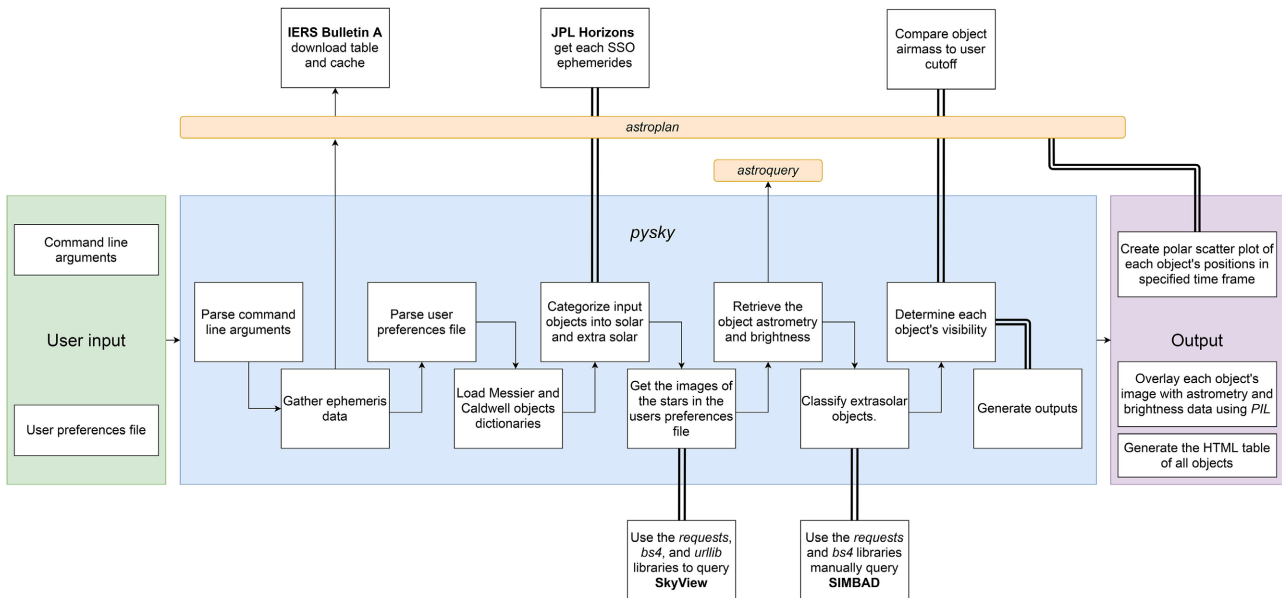
---

[2]http://www.hawthornhollow.org/observatory/.

Figure 1. Diagram depicting high-level view of the pysky application architecture. User inputs (depicted on the left) [green] lead to the application flow [blue], which calls on various other Python packages (astroplan, astroquery italicized) [orange] or external tools (IERS Bulletin A, JPL Horizons, SIMBAD, SkyView in boldface), and generates the requested outputs (depicted on the right) [purple]. Double lines signify processes that are parallelized.

Figure 2. Image of Jupiter with overlay attributes and plot of Jupiter horizontal coordinates starting on 2020 December 21 22:00:00 UT and ending on 2020 December 22 00:00:00 UT viewed from longitude −87.8791, latitude +42.6499. Images are overlaid with naming, categorizing, and astrometry information to contextualize the data. A polar plot of the horizontal coordinates during the observing window is generated for each of the identified visible objects (see also **Figures 2-4**). Such plots are intended to ease observatory use to look for the location and movement of the object in the night sky away from the telescope.

in its astronomical context. The polar plot identifies each object's position in horizontal coordinates during the viewing window to enable observatory users to find that object in the sky away from the telescope.
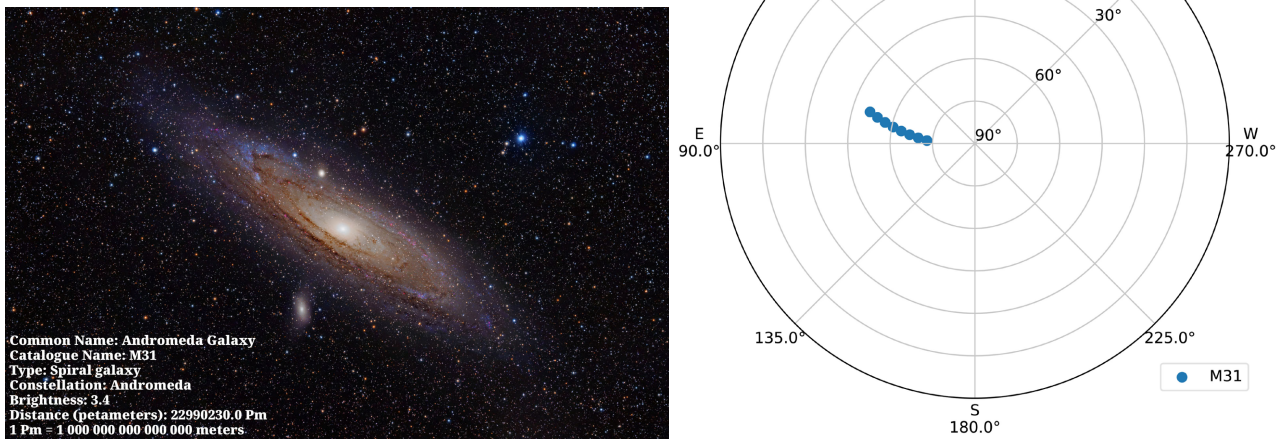
Figure 3. Image of M31 with overlay attributes and plot of M31 horizontal coordinates starting on 2020 December 21 22:00:00 UT and ending on 2020 December 22 00:00:00 UT viewed from longitude −87.8791, latitude +42.6499. pysky searches the Messier and Caldwell catalogs for objects visible during the observation window, and, for each of objects found, an image of the object is output with the information retrieved from SIMBAD overlaying the image. As with planets and stars (user-specified objects), a polar plot of the object's horizontal coordinates during the viewing window is output after the image is retrieved and annotated.
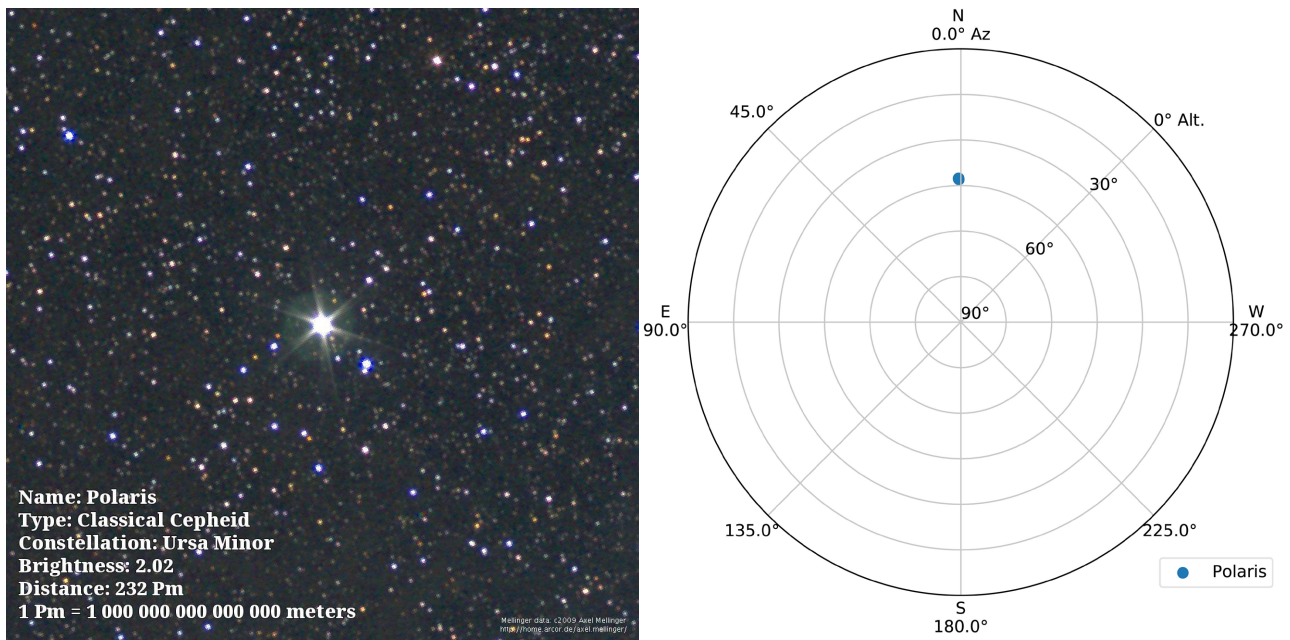

Figure 4. Image of Polaris with overlay attributes and plot of Polaris horizontal coordinates starting on 2020 December 21 22:00:00 UT and ending on 2020 December 22 00:00:00 UT viewed from longitude −87.8791, latitude +42.6499. For each of the visible stars requested by the user in the input file, an image of the star is output with the information retrieved from SIMBAD overlaying the image. As with the planets and catalog objects, the polar plot of the star's horizontal coordinates during the viewing window is output (in this case, showcasing Polaris's visibly insignificant movement during two hours).

Table 1 shows the output HTML table containing name, category, horizontal coordinates, and astrometry information designed for online publication to provide
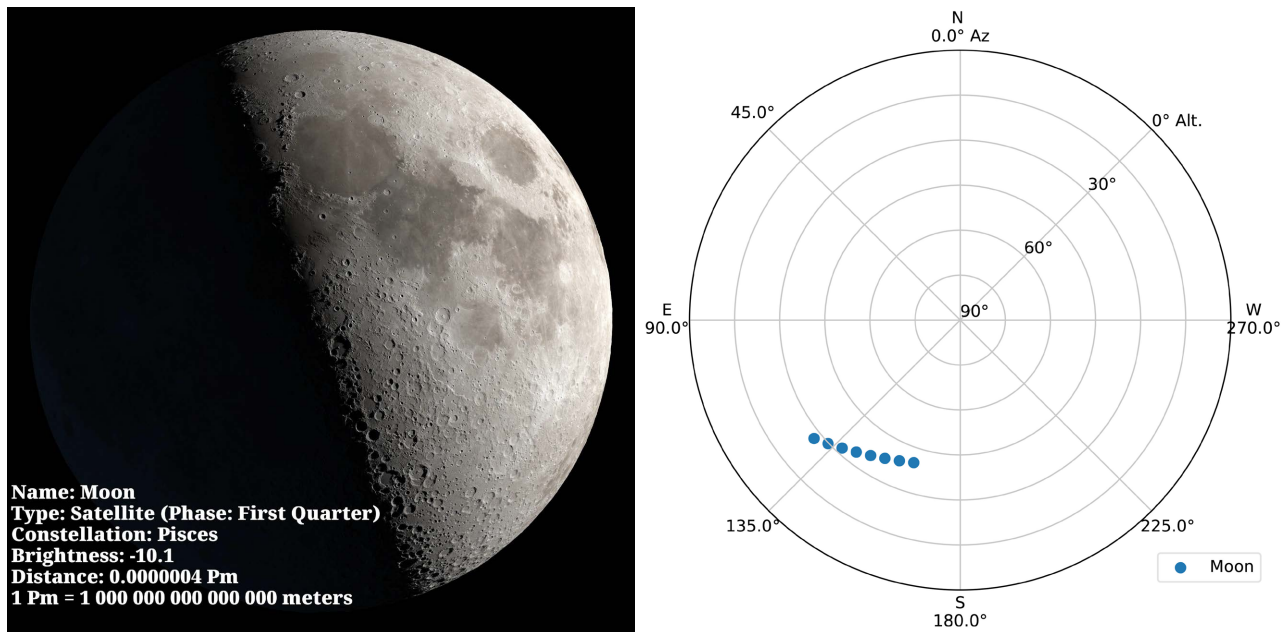
Figure 5. Image of the Moon with overlay attributes and plot of the Moon's horizontal coordinates starting on 2020 December 21 22:00:00 UT and ending on 2020 December 22 00:00:00 UT viewed from longitude −87.8791, latitude +42.6499. Moon phase, brightness, and distance information for the observation window are retrieved from JPL Horizons, and the relevant image of the Moon in the current phase is sourced from Wikimedia Commons and overlaid with the information. Just as with other objects in the observing list, a polar plot depicts the Moon's positions during the observing window.

information beforehand to observatory users about potential viewing targets during an observing session. The HTML table caption contains the observation start and stop times in ISO 8601 format followed by the observing location's geographic coordinates together with the cut-off values for the apparent visual magnitude and airmass. Horizontal coordinates were rounded to the nearest degree since visual acuity does not require further precision. These coordinates for objects in Table 1 were manually cross-checked with Stellarium [14] and, for solar-system objects, the web interface of JPL Horizons. All distances were converted to petameters (10^{15} m) for uniform depiction of solar and interstellar distances in a unit comparable to a light-year yet easily reducible to a human-scale length.

The application was developed to leverage multithreaded CPUs in order to reduce the execution time. Results of parallelization tests depicted in Figure 6 show a reduction in wall time with increasing thread number that appears to become insignificant after reaching 50% of single thread time at four threads. The wall time leveling out at $N_{thread} > 4$ is likely due to network bottleneck when retrieving the images from the SkyView server. This bottleneck could be overcome by increasing the network bandwidth, or by caching the SkyView images after downloading so that the application will reuse an image in another run instead of redownloading. Still, the total computational time of only a few minutes together with the limited scale of the number of potential target objects in an observation window diminishes interest in further algorithmic optimization.
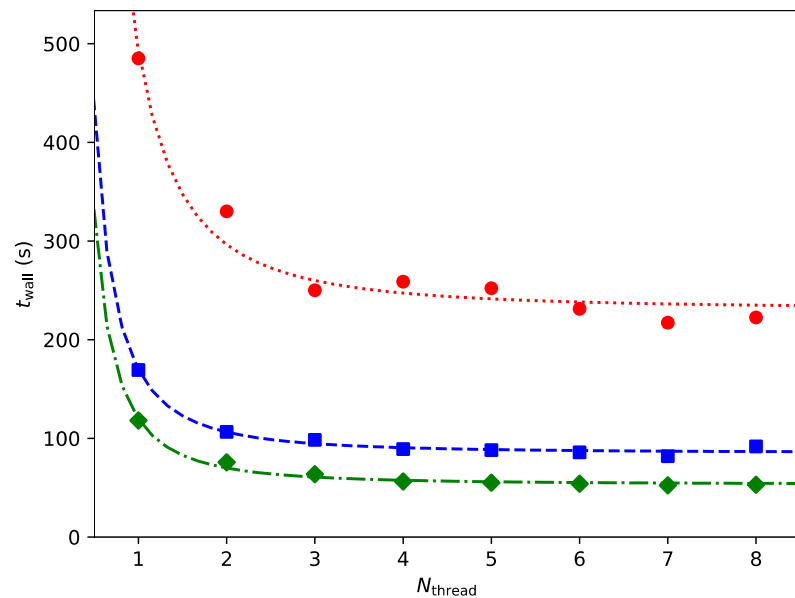
Figure 6. Wall time of pysky calculations with differing number of computational threads on three different machines (red circles, green diamonds, blue squares) for parameters described in Section 1. Corresponding curve fits guide the eye. Reduction in wall time from increased thread number appears to become insignificant after reaching 50% of single thread time at four threads.

4. Conclusions and Suggestions

The pysky application is a Python-based multi-target observation session planning application that produces overlaid images and sky plots of potential viewing objects together with an HTML table of their basic information. The pysky application allows observation planners to focus on their session goals, reducing the need to sort through maps or visualization software to bring together information on the fly while promoting ready visualization and curation of information for an observation session.

Future directions for pysky include: sorting the viewing list by the right ascension to assist with planning the order of viewing, adding a marathon functionality for all-night observation sessions focused on a particular catalog or object type or observing field, integration with mobile apps designed to provide easier access to the output data, and use of astronomical visualization metadata standard [15].

Acknowledgements

---

[3]https://beautiful-soup-4.readthedocs.io/en/latest/.
[4]https://docs.python-requests.org/en/master/.
[5]https://github.com/urllib3/urllib3.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] 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., Unther, H.M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J.E.H., Singer, L., Fox, R., Weaver, B.A., Zabalza, V., Edwards, Z.I., Azalee Bostroem, K., Burke, D.J., Casey, A.R., Crawford, S.M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P.L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M. and Streicher, O. (2013) Astropy: A Community Python Package for Astronomy. *Astronomy & Astrophysics*, **558**, A33.

[2] The 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., VanderPlas, J.T., Bradley, L.D., Pérez-Suárez, D., de Val-Borro, M., (Primary Paper Contributors), Aldcroft, T.L., Cruz, K.L., Robitaille, T.P., Tollerud, E.J., (Astropy Coordination Committee), Ardelean, C., Babej, T., Bach, Y.P., Bachetti, M., Bakanov, A.V., Bamford, S.P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K.L., Biscani, F., Boquien, M., Bostroem, K.A., Bouma, L.G., Brammer, G.B., Bray, E.M., Breytenbach, H., Buddelmeijer, H., Burke, D.J., Calderone, G., Cano Rodríguez, J.L., Cara, M., Cardoso, J.V.M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D'Avella, D., Deil, C., Depagne, E., Dietrich, J.P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L.A., Finethy, T., Fox, R.T., Garrison, L.H., Gibbons, S.L.J., Goldstein, D.A., Gommers, R., Greco, J.P., Greenfield, P., Groener, A.M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A.J., Hosseinzadeh, G., Hu, L., Hunkeler, J.S., Ivezić, Ž., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N.S., Kerzendorf, W.E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A.M., Kumar, A., Lee, A., Lenz, D., Littlefair, S.P., Ma, Z., Macleod, D.M., Mastropietro, M., McCully, C., Montagnac, S., Morris, B.M., Mueller, M., Mumford, S.J., Muna, D., Murphy, N.A., Nelson, S., Nguyen, G.H., Ninan, J.P., Nöthe, M., Ogaz, S., Oh, S., Parejko, J.K., Parley, N., Pascual, S., Patil, R., Patil, A.A., Plunkett, A.L., Prochaska, J.X., Rastogi, T., Reddy Janga, V., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L.E., Sherwood-Taylor, H., Shih, A.Y., Sick, J., Silbiger, M.T., Singanamalla, S., Singer, L.P., Sladen, P.H., Sooley, K.A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S.W., Tremblay, G.R., Turner, J.E.H., Terrón, V., van Kerkwijk, M.H., de la Vega, A., Watkins, L.L., Weaver, B.A., Whitmore, J.B., Woillez, J. and Zabalza, V. (Astropy Contributors) (2018) The Astropy Project: Building an Open-Science Project and Status of the v2.0 Core Package. *The Astronomical Journal*, **156**, Article ID: 123. https://doi.org/10.3847/1538-3881/aabc4f

[3] 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. and Jeschke, E. (2018) Astroplan: An Open Source Observation Planning Package in Python. *The Astronomical Journal*, **155**, Article ID: 128. https://doi.org/10.3847/1538-3881/aaa47e

[4] Cole, M., Cohen, C., Wilhelm, J. and Lindell, R. (2018) Spatial Thinking in Astronomy Education Research. *Physical Review Physics Education Research*, **14**, Article ID: 010139. https://doi.org/10.1103/PhysRevPhysEducRes.14.010139

[5] Price, C.A., Greenslit, J.N., Applebaum, L., Harris, N., Segovia, G., Quinn, K.A. and

Krogh-Jespersen, S. (2021) Awe & Memories of Learning in Science and Art Museums. *Visitor Studies*, **24**, 137-165. https://doi.org/10.1080/10645578.2021.1907152

[6]   McPhetres, J. (2019) Oh, the Things You Don't Know: Awe Promotes Awareness of Knowledge Gaps and Science Interest. *Cognition and Emotion*, **33**, 1599-1615. https://doi.org/10.1080/02699931.2019.1585331

[7]   Messier, C. (1784) Catalogue des Nébuleuses et des Amas d'Étoiles (Catalog of Nebulae and Star Clusters). Connoissance des Temps ou des Mouvements Célestes, Paris, 227-267.

[8]   Moore, P. and Pepin, M.B. (1995) Beyond Messier: The Caldwell Catalogue. Sky and Telescope Vol. 90, Cambridge, MA.

[9]   Giorgini, J.D., Yeomans, D.K., Chamberlin, A.B., Chodas, P.W., Jacobson, R.A., Keesey, M.S., Lieske, J.H., Ostro, S.J., Standish, E.M. and Wimberly, R.N. (1996) JPL's On-Line Solar System Data Service. *Bulletin of the American Astronomical Society*, **28**, Article ID: 1158.

[10] McGlynn, T., Scollick, K. and White, N. (1998) Skyview: The Multi-Wavelength Sky on the Internet. *Symposium-International Astronomical Union*, **179**, 465-466. https://doi.org/10.1017/S0074180900129316

[11] 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., Tollerud, E.J., de Val-Borro, M., Valtchanov, I., Woillez, J. and the Astroquery Collaboration, a Subset of Astropy Collaboration. (2019) Astroquery: An Astronomical Web-Querying Package in Python. *The Astronomical Journal*, **157**, Article ID: 98. https://doi.org/10.3847/1538-3881/aafc33

[12] Wenger, M., Ochsenbein, F., Egret, D., Dubois, P., Bonnarel, F., Borde, S., Genova, F., Jasniewicz, G., Laloë, S., Lesteven, S. and Monier, R. (2000) The SIMBAD Astronomical Database. The CDS Reference Database for Astronomical Objects. *Astronomy & Astrophyiscs*, **143**, 9-22.

[13] Mellinger, A. (2009) A Color All-Sky Panorama Image of the Milky Way. *Publications of the Astronomical Society of the Pacific*, **121**, 1180-1187. https://doi.org/10.1086/648480 http://www.jstor.org/stable/10.1086/648480

[14] Zotti, G., Hoffmann, S.M., Wolf, A., Chéreau, F. and Chéreau, G. (2021) The Simulated Sky: Stellarium for Cultural Astronomy Research. *Journal of Skyscape Archaeology*, **6**, 221-258. https://doi.org/10.1558/jsa.17822 https://journal.equinoxpub.com/JSA/article/view/17822

[15] Hurt, R.L., Gauthier, A.J., Christensen, L.L. and Wyatt, R. (2008) Sharing Images Intelligently: The Astronomy Vizualisation Metadata Standard. In: Christensen, L.L., Zoulias, M. and Robson, I., Eds., *Communicating Astronomy with the Public*, Eugenides Foundation, Athens, 450 p.

[16] Hunter, J.D. (2007) Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, **9**, 90-95. https://doi.org/10.1109/MCSE.2007.55

[17] 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., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perk-told, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F. and van Mulbregt, P. (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

[18]   Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Courna-
       peau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S.,
       van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P.,
       Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke,
       C. and Oliphant, T.E. (2020) Array Programming with NumPy. *Nature*, **585**,
       357-362. https://doi.org/10.1038/s41586-020-2649-2

[19]   van Kemenade, H., Murray, A., Clark, A., Karpinsky, A., Baranovič, O., Gohlke, C.,
       Dufresne, J., Crowell, B., Schmidt, D., Kopachev, K., Houghton, A., Mani, S., Lan-
       dey, S., Vashek, Ware, J., Douglas, J., Caro, D., Martinez, U., Kossouho, S., Lahd, R.,
       Lee, A., Brown, E.W., Tonnhofer, O., Bonfill, M., Rowlands, P., Al-Saidi, F., Novi-
       kov, G. and Górny, M. (2021) Python-Pillow/Pillow: 8.2.0.