




Windrose: A Python Matplotlib, Numpy library to manage wind and pollution data, draw windrose

Lionel Roubeyrie¹ and Sébastien Celles²

DOI: [10.21105/joss.00268](https://doi.org/10.21105/joss.00268)

¹ LIMAIR ² Université de Poitiers - IUT de Poitiers (Poitiers Institute of Technology)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Submitted: 16 May 2017

Published: 04 September 2018

License

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

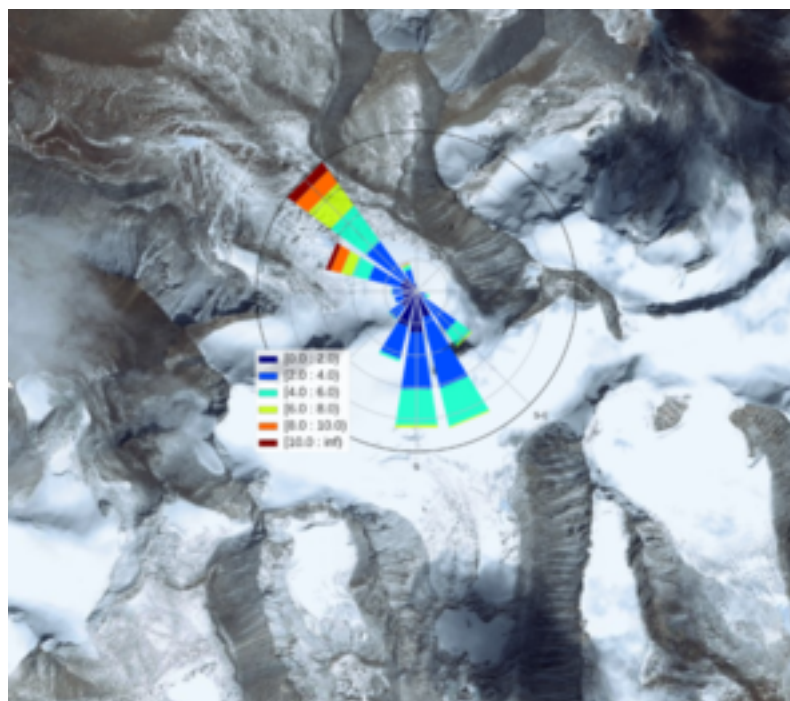
Summary

A [wind rose](#) is a graphic tool used by meteorologists to give a succinct view of how wind speed and direction are typically distributed at a particular location. It can also be used to describe air quality pollution sources. The wind rose tool uses Matplotlib as a backend. Data can be passed to the package using Numpy arrays or a Pandas DataFrame.

Windrose is a Python library to manage wind data, draw windroses (also known as polar rose plots), and fit Weibull probability density functions.

The initial use case of this library was for a technical report concerning pollution exposure and wind distributions analyzes. Data from local pollution measures and meteorologic informations from various sources like Meteo-France were used to generate a pollution source wind rose.

It is also used by some contributors for teaching purpose.



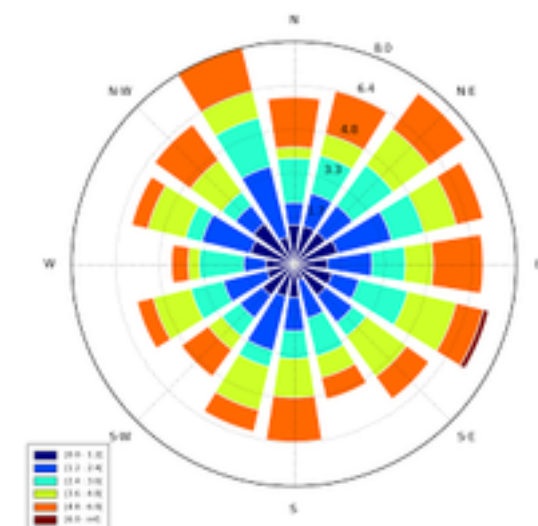
Some others contributors have used it to make figures for a [wind power plant control optimization study](#).

Some academics use it to track lightning strikes during high intensity storms. They are using it to visualize the motion of storms based on the relative position of the lightning

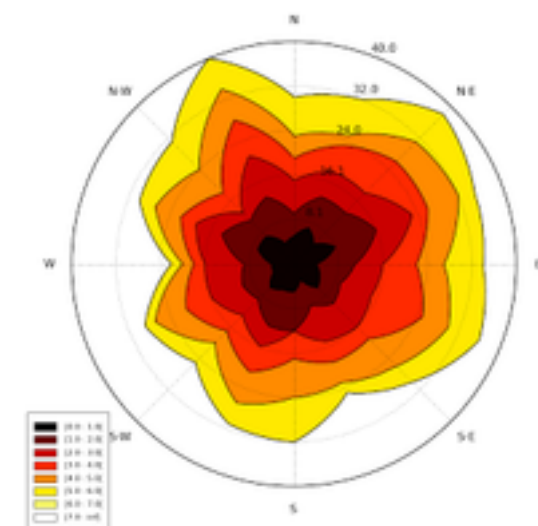
from one strike to the next.

Examples

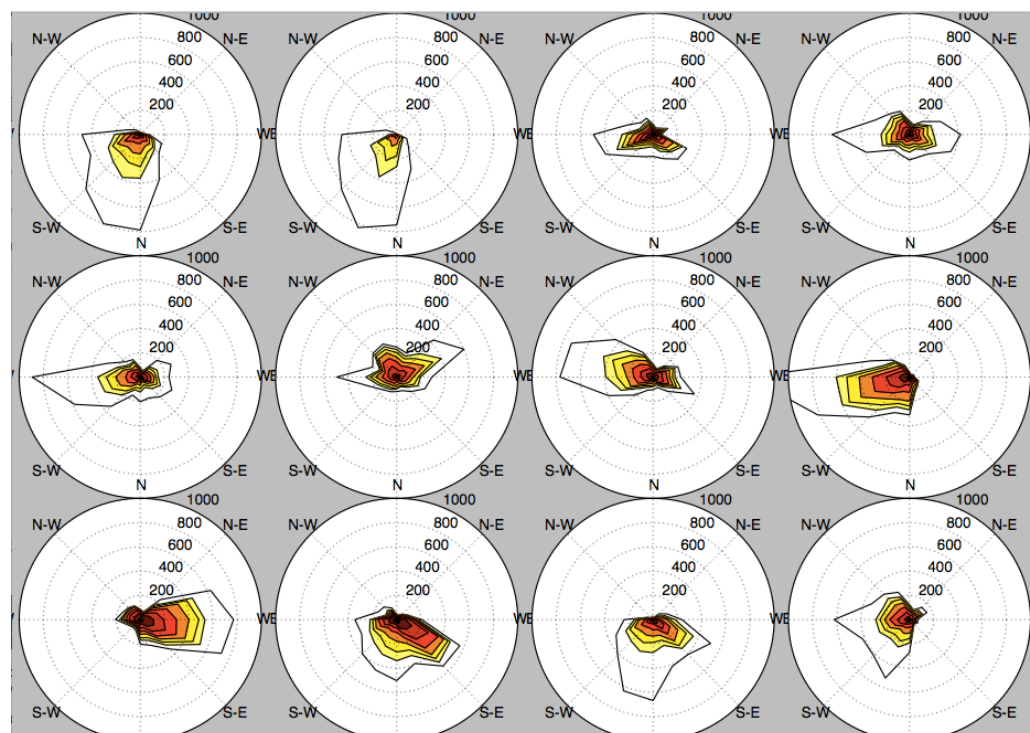
- The bar plot wind rose is the most common plot



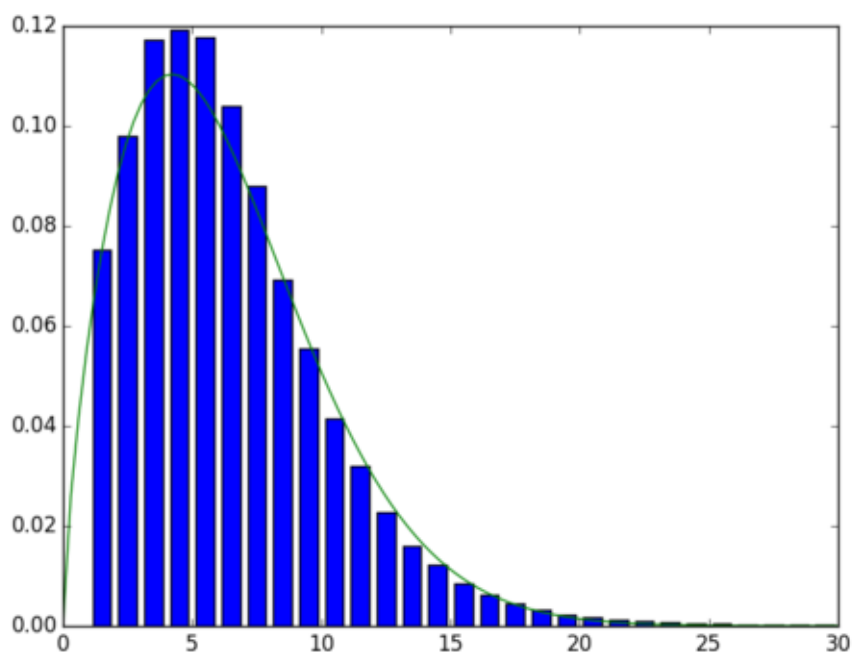
- Contour plots are also possible



- Several windroses can be plotted using subplots to provide a plot per year with for example subplots per month



- Probability density functions (PDF) may be plotted. Fitting Weibull distribution is enabled by Scipy. The Weibull distribution is used in weather forecasting and the wind power industry to describe wind speed distributions, as the natural distribution of wind speeds often matches the Weibull shape



More advanced usages and contributing

Full documentation of library is available at <http://windrose.readthedocs.io/>.

If you discover issues, have ideas for improvements or new features, please report them. [CONTRIBUTING.md](#) explains how to contribute to this project.

List of contributors and/or notable users can be found at [CONTRIBUTORS.md](#).

Future

Windrose is still an evolving library which still need care from its users and developers.

- Map overlay is a feature that could help some users.
- A better API for video exporting could be an interesting improvement.
- Add the capability to make wind roses based on the Weibull shape and scale factors could be considered.
- Make windroses from an histogram table rather than from two arrays of wind speed and wind direction is also a requested feature.

References

- Garver, D., & Brown, R. (2016–2016--). Air quality data analysis using open source tools - united states environmental protection agency | us epa. Retrieved from <http://slideplayer.com/slide/5922455/>
- Harris, R. I., & Cook, N. J. (2014). The parent wind speed distribution: Why weibull? *Journal of wind engineering and industrial aerodynamics*, 131, 72–87. doi:[10.1016/j.jweia.2014.05.005](https://doi.org/10.1016/j.jweia.2014.05.005)
- Horel, J., Crosman, E., Jacques, A., Blaylock, B., Arens, S., Long, A., Sohl, J., et al. (2016). Summer ozone concentrations in the vicinity of the great salt lake. *Atmospheric Science Letters*, 17(9), 480–486. doi:[10.1002/asl.680](https://doi.org/10.1002/asl.680)
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. doi:[10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Jones, E., Oliphant, T., Peterson, P., & others. (2001–2001--). SciPy: Open source scientific tools for Python. Retrieved from <http://www.scipy.org/>
- 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. 51–56).
- Millman, K. J., & Aivazis, M. (2011). Python for scientists and engineers. *Computing in Science & Engineering*, 13(2), 9–12. doi:[10.1109/MCSE.2011.36](https://doi.org/10.1109/MCSE.2011.36)
- Munn, R. (1969). Pollution wind-rose analysis. *Atmosphere*, 7(3), 97–105. doi:[10.1080/00046973.1969.9676573](https://doi.org/10.1080/00046973.1969.9676573)
- Oliphant, T. E. (2006). *A guide to numpy* (Vol. 1). Trelgol Publishing USA.
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3), 10–20. doi:[10.1109/MCSE.2007.58](https://doi.org/10.1109/MCSE.2007.58)
- Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering*, 9(3), 21–29. doi:[10.1109/MCSE.2007.53](https://doi.org/10.1109/MCSE.2007.53)

Quick, J., Annoni, J., King, R., Dykes, K., Fleming, P., & Ning, A. (2017). Optimization under uncertainty for wake steering strategies. In *Journal of physics: Conference series* (Vol. 854, p. 012036). IOP Publishing.

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science and Engg.*, 13(2), 22–30. doi:[10.1109/MCSE.2011.37](https://doi.org/10.1109/MCSE.2011.37)

Water, N. N., & Center, C. (n.d.). NRCS national water and climate center - climate products - climate data - wind rose. Retrieved from <http://www.wcc.nrcs.usda.gov/climate/windrose.html>

Wikipedia contributors. (2018). Wind rose — Wikipedia, the free encyclopedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Wind_rose&oldid=839157228