




Fast Resampling and Monte Carlo Methods in Python

Matt Haberland  ¹✉

¹ California Polytechnic State University, San Luis Obispo, USA ✉ Corresponding author

DOI: [10.21105/joss.05092](https://doi.org/10.21105/joss.05092)

Software

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

Editor: [Mehmet Hakan Satman](#) 



Reviewers:

- [@SaranjeetKaur](#)
- [@kose-y](#)

Submitted: 04 January 2023

Published: 13 June 2023

License

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

Summary

In the 1935 book *The Design of Experiments* ([Fisher, 1951](#)), Ronald Fisher analyzed the results of an experiment performed by Charles Darwin to determine the effect of cross-fertilization on plant growth ([Darwin, 1877](#)). In these experiments, Darwin collected seeds from both cross- and self-fertilized flowers of a parent plant and germinated the seeds under identical conditions. When a pair of cross- and self-fertilized seeds happened to germinate at the same time, the two were planted in opposite sides of the same pot and nurtured in the same environment. At the end of the study, the heights of the plants were measured.

To infer whether the greater height of the cross-fertilized plants should be attributed to random chance or the change in fertilization method, Fisher computed the differences in heights between paired plants and used a paired-sample t-test ([Student, 1908](#)) to estimate the statistical significance of the results. He noted that the t-test relies on assumptions about the population of height differences, and he proposed the following statistical test, which makes no such assumptions.

Under the null hypothesis that the fertilization method has no effect on plant height, the heights of the two plants in a pair are sampled from the same population. Therefore, under the null hypothesis, the sign of the difference in heights would be positive or negative with equal probability. To perform the test, Fisher considers the distribution of mean height differences under all possible combinations of signs: the “null distribution”. If the observed mean height difference is greater than almost all elements in the null distribution, it is unlikely to be observed under the null hypothesis, and this is taken as evidence against the null hypothesis in favor of the alternative that cross-fertilized plants tend to be taller.

This is an early example of a *permutation test*. Like *Monte Carlo tests* ([Robert et al., 1999](#)) and the *bootstrap* ([Efron & Tibshirani, 1994](#)), permutation tests replace many assumptions and mathematical approximations with *lots* of computation. Naturally, these methods did not become very popular until the advent of the computer, but even today, some popular programming languages lack efficient implementations of these essential statistical techniques. This paper discusses the addition of functions implementing these techniques to SciPy ([Virtanen et al., 2020](#)), the de facto standard library for scientific computing in Python.

Statement of need

Scientists and engineers frequently analyze data by considering questions of the following forms, illustrated with examples based on Darwin’s experiment ([Darwin, 1877](#)).

1. Are the observations sampled from a hypothesized distribution? For instance, are the differences in heights between paired plants normally distributed?
2. Are the samples drawn from the *same* distribution? For example, does the fertilization method have no effect on plant height?
3. What can be inferred from the samples about the true value of a population statistic? For instance, what is the uncertainty in the true mean difference in heights between

paired plants?

Monte Carlo hypothesis tests, permutation tests, and the bootstrap are general procedures for answering questions of these forms, respectively.

Although there are a variety of specialized statistical procedures for answering such questions (e.g., normality tests, t-tests, standard error of the mean), most computer implementations provide answers that are only guaranteed to be accurate when assumptions that simplify the mathematics are met (e.g., the sample size is large, there are no ties in the data). At the expense of additional computation, the `scipy.stats` functions `monte_carlo_test`, `permutation_test`, and `bootstrap` provide accurate answers to questions 1-3 even when the simplifying assumptions of specialized procedures are not met. Furthermore, these functions can be used to answer questions 1-3 even when specialized procedures have not been *developed*, giving users tremendous flexibility in the analysis that they can perform.

Before the release of SciPy 1.9.0, the need for these procedures was partially met in the scientific Python ecosystem by tutorials (Omar, 2021; Saenger, 2019; Ye, 2020) and niche packages (Millman, 2021; Saxton & Dembinski, 2020), but the functions in SciPy have several advantages:

- Prevalence: SciPy is one of the most frequently downloaded scientific Python packages. If a Python user finds the need for these statistical methods, chances are that they already have SciPy, eliminating the need to find and install a new package.
- Speed: the functions take advantage of vectorized user code, avoiding slow Python loops.
- Easy-of-use: the function API reference and tutorials are thorough, and the interfaces share common features and complement other SciPy functions.
- Dependability: as with all SciPy code, these functions were rigorously peer-reviewed, and unit tests are extensive.

Acknowledgements

We gratefully acknowledge the support of Chan Zuckerberg Initiative Essential Open Source Software for Science Grant EOSS-0000000432 and the 2022 Round 1 NumFOCUS Small Development Grant “Introducing Users to Powerful New Features of SciPy”. Thanks to all those who have submitted feature requests, bug reports, and review comments related to these features, and especially to SciPy maintainers Christoph Baumgarten, Robert Lucas, Tirth Patel, Pamphile Roy, and Warren Weckesser who reviewed this work.

References

- Darwin, C. (1877). *The effects of cross and self fertilisation in the vegetable kingdom*. D. Appleton; Company.
- Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.
- Fisher, R. A. (1951). *The design of experiments*. Oliver & Boyd. ISBN: 9780050008683
- Millman, J. (2021). *Permute 0.2*. <https://github.com/statlab/permute>
- Omar, H. (2021). A guide on how to simulate and visualize hypothesis tests using Python code. In *Analytics Vidhya*. Medium. <https://medium.com/analytics-vidhya/66861e0d0f74>
- Robert, C. P., Casella, G., & Casella, G. (1999). *Monte Carlo statistical methods* (Vol. 2). Springer.
- Saenger, V. (2019). How to assess statistical significance in your data with permutation tests. In *Towards Data Science*. Medium. <https://towardsdatascience.com/8bb925b2113d>

- Saxton, D., & Dembinski, H. (2020). *Resample 1.0.0*. <https://github.com/scikit-hep/resample/>
- Student. (1908). The probable error of a mean. *Biometrika*, 1–25. <https://doi.org/10.2307/2331554>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., & others. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Ye, L. (2020). Bootstrap and statistical inference in Python. In *Towards Data Science*. Medium. <https://towardsdatascience.com/a06d098a8bfd>