

Hypothesize: Robust Statistics for Python

Allan Campopiano¹ and Rand R. Wilcox²

¹ Halton Catholic District School Board ² University of Southern California

DOI: [10.21105/joss.02241](https://doi.org/10.21105/joss.02241)

Software

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

Editor: [Charlotte Soneson](#) ↗

Reviewers:

- [@martinagvilas](#)
- [@arbennett](#)

Submitted: 20 May 2020

Published: 19 June 2020

License

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

Summary

Hypothesize is a robust null hypothesis significance testing (NHST) library for Python. In general, robust hypothesis testing uses techniques which minimize the effects of violating standard statistical assumptions. In particular, robust methods based on the trimmed mean (Tukey, 1960) and/or bootstrapping (Efron & Tibshirani, 1993), routinely outperform traditional statistical approaches in terms of power and accuracy. This is especially true when dealing with distributions that produce outliers (Wilcox, 1998, 2013).

Hypothesize is based on Rand R. Wilcox's collection of [R functions](#) which contains hundreds of robust methods developed since the 1960's. Hypothesize brings many of these functions into the Python library ecosystem with the goal of making robust hypothesis testing easy for researchers, even if they have not had extensive training in statistics or computer science. It is, however, assumed that users have a basic understanding of the concepts and terms related to robust hypothesis testing (e.g., trimmed mean and bootstrapping).

In contrast to other statistical libraries in Python (Ho, Tumkaya, Aryal, Choi, & Claridge-Chang, 2019; Seabold & Perktold, 2010; Vallat, 2018), Hypothesize is focused solely on robust methods for comparing groups and measuring associations. Researchers who are familiar with traditional NHST and related concepts (e.g., t-test, ANOVA, Pearson's correlation) will find analogous approaches in Hypothesize. A broad range of choices exist in Hypothesize both in terms of the supported statistical designs as well as options for fine-grained control over how tests are computed. For example:

- Where applicable, many hypothesis tests allow the specification of an estimator. That is, users may choose when to use the mean, median, trimmed mean, winsorized correlation, percentage bend correlation, or any other compatible statistical estimator.
- Single- and multi-factor designs are supported, and this includes supporting independent, dependent, and mixed groups.
- Family-wise error can be robustly controlled with sequentially rejective methods (Benjamini & Hochberg, 1995; Hochberg, 1988; Rom, 1990).

In terms of learning to use the software, Hypothesize keeps the barrier to entry low for researchers. For example:

- To easily incorporate Hypothesize with standard data processing tools (see McKinney, 2010), all top-level functions take a Pandas DataFrame/Series as input and return a Python Dictionary.
- The API maps cleanly onto features of the user's statistical design. This makes it easier to find and discover the set of appropriate functions for a given use case.
- All top-level functions can be run directly in the browser alongside the documentation via [Google Colab Notebooks](#) (no local installation required).

Acknowledgements

The authors would like to thank James Desjardins, Stefon van Noordt, Lisa Collimore, Martina G. Vilas, Andrew Bennett, Charlotte Soneson, Whedon, the Journal of Open Source Software, and the Halton Catholic District School Board for their support of this project.

References

- Benjamini, Y., & Hochberg. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57, 289–300. doi:[10.1111/j.2517-6161.1995.tb02031.x](https://doi.org/10.1111/j.2517-6161.1995.tb02031.x)
- Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. *Monographs on Statistics and Applied Probability*, 57.
- Ho, J., Tumkaya, T., Aryal, S., Choi, H., & Claridge-Chang, A. (2019). Moving beyond p values: Data analysis with estimation graphics. *Nature Methods*, 16(7), 565–566. doi:[10.1038/s41592-019-0470-3](https://doi.org/10.1038/s41592-019-0470-3)
- Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4), 800–802. doi:[10.1093/biomet/75.4.800](https://doi.org/10.1093/biomet/75.4.800)
- McKinney. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). doi:[10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a)
- Rom, D. M. (1990). A sequentially rejective test procedure based on a modified Bonferroni inequality. *Biometrika*, 77(3), 663–665. doi:[10.1093/biomet/77.3.663](https://doi.org/10.1093/biomet/77.3.663)
- Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *9th python in science conference*. doi:[10.25080/majora-92bf1922-011](https://doi.org/10.25080/majora-92bf1922-011)
- Tukey, J. W. (1960). A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics*, 448–485. Retrieved from <https://ci.nii.ac.jp/naid/20000755025/en/>
- Vallat, R. (2018). Pingouin: Statistics in Python. *Journal of Open Source Software*, 3(31), 1026. doi:[10.21105/joss.01026](https://doi.org/10.21105/joss.01026)
- Wilcox, R. R. (1998). How many discoveries have been lost by ignoring modern statistical methods? *American Psychologist*, 53(3), 300. doi:[10.1037/0003-066X.53.3.300](https://doi.org/10.1037/0003-066X.53.3.300)
- Wilcox, R. R. (2013). *Introduction to robust estimation and hypothesis testing*. Academic press. doi:[10.1016/c2010-0-67044-1](https://doi.org/10.1016/c2010-0-67044-1)