





# fwb: An R package for the fractional weighted bootstrap

Noah Greifer<sup>1</sup> 

<sup>1</sup> Institute for Quantitative Social Science, Harvard University, United States  Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

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

Editor: [Susan Holmes](#) 

## Reviewers:

- [@tobiasschoch](#)

Submitted: 16 June 2025

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

## Summary

The nonparametric bootstrap is a simple method for performing inference on estimates of statistical parameters without making strong assumptions about the data or a statistical model and without needing to derive the formula for an estimate's variance (Efron & Tibshirani, 1993). The traditional nonparametric bootstrap involves resampling the observations from the sample with replacement, computing the estimates in the bootstrap sample, performing these steps many times, and summarizing the resulting distribution of bootstrap estimates to compute their confidence intervals. This is equivalent to drawing integer weights from a multinomial distribution and computing weighted estimates in each bootstrap sample. However, integer weights are not the only weights that can be used, and fractional (non-integer) weights carry certain advantages, described below. fwb implements the fractional weighted bootstrap, also known as the Bayesian bootstrap, using a simple interface and with options for parallelization.

## Statement of need

The fractional weighted bootstrap was described originally by Rubin (1981) and, for a more applied audience, by Xu et al. (2020). In each bootstrap sample, weights are drawn from a specified distribution, and the statistics of interest are computed in the weighted sample. In contrast to the traditional bootstrap, which drops any units that are not included in a given bootstrap sample in computing that sample's estimates, the fractional weighted bootstrap retains all units in every bootstrap sample. This yields several benefits, including that rare values are included in every bootstrap sample, and the resulting bootstrap distribution will be smoother (when the weights are drawn from a continuous distribution) (Xu et al., 2020).

This is demonstrated in Xu et al. (2020), who use the fractional weighted bootstrap to estimate the center and shape of a Weibull distribution characterizing failure times for a set of 1703 aircraft engines, among which only 6 failures were observed, data previously analyzed by Meeker & Escobar (1998). In cases with rare outcomes, rare covariate profiles, or highly influential observations, the traditional bootstrap can fail to yield valid inference. The fractional weighted bootstrap avoids these issues by retaining all units and allowing the weights to take on strictly positive, non-integer values.

The traditional bootstrap is implemented in the boot package (Canty & Ripley, 2024; Davison & Hinkley, 1997), which uses a simple interface to perform bootstrapping. The user supplies a dataset and a function that computes the desired statistics on each bootstrap sample from the dataset to `boot::boot()`. `boot::boot.ci()` is used to compute confidence intervals using a variety of methods. fwb provides a drop-in replacement for `boot::boot()` in `fwb::fwb()`, which similarly takes in the original dataset and a function that computes the desired statistics incorporating the bootstrap weights. `fwb()` supports parallelization through the `pbapply` package (Solymos & Zawadzki, 2023), which provides interfaces to the `parallel` and `future`

packages (Bengtsson, 2021) for parallel processing, as well as stratified bootstrapping (i.e., bootstrapping within strata), clustered bootstrapping (bootstrapping of clusters), and different distributions from which the weights are drawn. These distributions include the Dirichlet/Exponential weights originally described by Rubin (1981), “Mammen” weights that mimic the form of the distribution of weights proposed by Mammen (1993) for the wild bootstrap, multinomial weights to exactly reproduce the traditional nonparametric bootstrap by boot, and Poisson frequency weights (Hanley & MacGibbon, 2006).

Like boot, fwb provides tools for plotting the bootstrap distributions and computing confidence intervals, including the Normal approximation, percentile, bias-corrected percentile, and bias-corrected and accelerated (BCa) confidence intervals implemented in `fwb.ci()` (Davison & Hinkley, 1997). fwb also provides additional S3 methods for fwb objects, including a `summary()` method to facilitate extraction of estimates and confidence intervals and perform hypothesis tests, which are otherwise fairly challenging in boot.

A common use of the bootstrap is to compute an estimate of the asymptotic covariance matrix of parameter estimates in regression models. Bootstrapped confidence intervals avoid some of the often strict and unrealistic assumptions of maximum likelihood-based intervals, in particular, that the model is correctly specified (e.g., that the conditional variance is constant in a linear regression model) (Berk et al., 2021). The sandwich package (Zeileis et al., 2020) provides `sandwich::vcovBS()` to compute bootstrapped covariance matrices of the regression parameters, and fwb provides `vcovFWB()` for the same purpose but supporting the fractional weighted bootstrap<sup>1</sup>. These functions can be used as a drop-in for `vcov()` for many regression functions (in particular, those that have an `update()` method). fwb also includes utility functions for computing weighted statistics and variable transformations, such as `w_mean()` for computing the weighted mean and `w_std()` for standardizing (scaling and centering), which automatically incorporate bootstrap weights when used inside `fwb()` or `vcovFWB()`.

The bayesboot package (Bååth, 2025) also implements the Bayesian bootstrap, but it does so in an explicitly Bayesian fashion, focusing on visualizing the bootstrap distribution as a Bayesian posterior distribution. In contrast, fwb is designed to be part of a frequentist workflow similar to traditional bootstrap implementations, e.g., in how it reports point estimates, confidence intervals, and hypothesis tests rather than using the bootstrap distribution to characterize the posterior of the estimate. bayesboot supports bootstrapping with Dirichlet weights and supports the use of statistics that cannot directly incorporate weights, but it does not support other weight distributions, clustering, or confidence interval estimation (other than highest density credible intervals that are part of standard Bayesian analysis workflows). fwb is designed to provide an alternative to boot for frequentist inference using the more robust fractional weighted bootstrap rather than implementing the Bayesian bootstrap as a specific Bayesian procedure.

fwb is available on [CRAN](https://cran.r-project.org/web/packages/fwb/index.html).

## References

- Bååth, R. (2025). *Bayesboot: An implementation of rubin's (1981) bayesian bootstrap*. <https://doi.org/10.32614/CRAN.package.bayesboot>
- Bengtsson, H. (2021). A Unifying Framework for Parallel and Distributed Processing in R using Futures. *The R Journal*, 13(2), 208. <https://doi.org/10.32614/rj-2021-048>
- Berk, R., Buja, A., Brown, L., George, E., Kuchibhotla, A. K., Su, W., & Zhao, L. (2021). Assumption lean regression. *The American Statistician*, 75(1), 76–84. <https://doi.org/10.1080/00031305.2019.1592781>

<sup>1</sup>Note that `sandwich::vcovBS()` supports a basic form of the fractional weighted bootstrap but does not support different distributions of weights or other options provided by fwb.

- 88 Canty, A., & Ripley, B. (2024). *Boot: Bootstrap functions (originally by angelo canty for s)*.  
89 The R Foundation. <https://doi.org/10.32614/cran.package.boot>
- 90 Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap methods and their application* (Reprinted  
91 with corrections 2003). Cambridge University Press. ISBN: 978-0-521-57471-6
- 92 Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. Chapman & Hall.  
93 ISBN: 978-0-412-04231-7
- 94 Hanley, J. A., & MacGibbon, B. (2006). Creating non-parametric bootstrap samples using  
95 Poisson frequencies. *Computer Methods and Programs in Biomedicine*, 83(1), 57–62.  
96 <https://doi.org/10.1016/j.cmpb.2006.04.006>
- 97 Mammen, E. (1993). Bootstrap and wild bootstrap for high dimensional linear models. *The*  
98 *Annals of Statistics*, 21(1). <https://doi.org/10.1214/aos/1176349025>
- 99 Meeker, W. Q., & Escobar, L. A. (1998). *Statistical methods for reliability data*. Wiley.
- 100 Rubin, D. B. (1981). The bayesian bootstrap. *The Annals of Statistics*, 9(1), 130–134.  
101 <https://doi.org/10.1214/aos/1176345338>
- 102 Solymos, P., & Zawadzki, Z. (2023). *Pbapply: Adding progress bar to '\*apply' functions*. The  
103 R Foundation. <https://doi.org/10.32614/cran.package.pbapply>
- 104 Xu, L., Gotwalt, C., Hong, Y., King, C. B., & Meeker, W. Q. (2020). Applications of  
105 the Fractional-Random-Weight Bootstrap. *The American Statistician*, 74(4), 345–358.  
106 <https://doi.org/10.1080/00031305.2020.1731599>
- 107 Zeileis, A., Köll, S., & Graham, N. (2020). Various Versatile Variances: An Object-Oriented  
108 Implementation of Clustered Covariances in R. *Journal of Statistical Software*, 95(1).  
109 <https://doi.org/10.18637/jss.v095.i01>