

# Expectations.jl: Quick and Accurate Expectation Operators in Julia

Arnav Sood<sup>1</sup>

<sup>1</sup> Vancouver School of Economics, University of British Columbia

DOI: [10.21105/joss.02065](https://doi.org/10.21105/joss.02065)

## Software

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

Editor: [Pending Editor](#) ↗

Submitted: 03 February 2020

Published: 03 February 2020

## License

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

## Summary

Many statistical problems require taking an expectation of some function  $f(x)$ , where  $x$  is drawn from some known distribution. For example, a well-known economic model of job search (McCall, 1970) involves calculating an expected value  $\mathbb{E}[f(w)]$ , where  $w$  is a random wage offer, and  $f(\cdot)$  is the lifetime value of that offer.

Julia's `Distributions.jl` (Lin et al., 2019) package provides many random variable objects, but taking expectations is still a laborious process. Traditional approaches include Monte Carlo simulation (slow, and potentially inaccurate), or custom numerical integration (inaccessible for non-statisticians.) And both of these approaches fail to capitalize on one of Julia's key features: the similarity between math and Julia code.

The `Expectations.jl` package addresses these weaknesses. By implementing custom Gaussian integration (also known as *quadrature*) schemes around well-known distributions, we provide fast and compact expectation operators. By making these callable objects, we allow these to be used as valid linear operators (acting on vectors, supporting scalar multiplication, etc.) Accuracy is not compromised; in testing, two pairs of 32-node vectors are sufficient to compute expectations to machine precision. For distributions without a custom quadrature rule, we give generic fallbacks.

## Mathematical and Computational Details

For a (continuous) univariate random variable  $X$ , following a cumulative distribution function  $G(\cdot)$ , the expectation is defined as:

$$\mathbb{E}[f(X)] = \int_S f(x) dG(x)$$

Where  $S$  is the *support* of  $X$ , or the set of values for which  $X$  is nonzero.

The integral is what makes this quantity challenging to compute. A Monte Carlo method might approximate it by drawing a large sample of points  $S = \{x_1, x_2, \dots, x_N\}$ , and then simply taking the average  $\mathbb{E}[f(X)] = \frac{1}{N} \sum_{i=1}^N f(x_i)$ .

While the estimator  $\hat{\mathbb{E}}$  has several attractive statistical properties, Monte Carlo methods tend to be resource-intensive (one must draw a large enough sample, store it in memory, and compute the average.)

We compute the integral via so-called *Gaussian quadrature* (specifically, via calls to the Julia package [FastGaussQuadrature.jl](#).) There are several flavors (Gauss-Legendre, Gauss-Hermite,

Gauss-Laguerre, etc.) which are suitable for various domains  $S$  (compact, infinite, semi-infinite, etc.), and therefore for various distributions.

The core of each, however, is the approximation of an integral by the dot product  $\mathbf{n} \cdot \mathbf{w}$ , where  $\mathbf{n}$  is called the *node vector*, and  $\mathbf{w}$  the *weight vector*. To take the expectations of arbitrary functions, we simply apply the transformation to the nodes. That is, if:

$$\mathbb{E}[X] \approx \mathbf{n} \cdot \mathbf{w}$$

Then:

$$\mathbb{E}[f(X)] \approx f(\mathbf{n}) \cdot \mathbf{w}$$

Where  $f(\mathbf{n})$  is the function  $f$  applied to each element of  $\mathbf{n}$ .

The computation of these weights and nodes is a literature in its own right. We refer to the introduction of (Townsend, Trogdon, & Olver, 2014) for an exposition (the authors also maintain the FastGaussQuadrature library mentioned above.)

## Acknowledgements

The authors gratefully acknowledge support from Patrick Kofod Mogensen, by allowing us to refer to his [DistQuads.jl](#) package for the underlying calls to the quadrature library.

The [QuantEcon](#) organization, which supported this work, is a NumFocus Fiscally Sponsored Project currently funded primarily by the Alfred P. Sloan foundation.

## References

- Lin, D., White, J. M., Byrne, S., Bates, D., Noack, A., Pearson, J., Arslan, A., et al. (2019). JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions. doi:[10.5281/zenodo.2647458](https://doi.org/10.5281/zenodo.2647458)
- McCall, J. J. (1970). Economics of information and job search. *The Quarterly Journal of Economics*, 84(1), 113–126. Retrieved from <https://EconPapers.repec.org/RePEc:oup:qjecon:v:84:y:1970:i:1:p:113-126>.
- Townsend, A., Trogdon, T., & Olver, S. (2014). Fast computation of gauss quadrature nodes and weights on the whole real line. Retrieved from <http://arxiv.org/abs/1410.5286>