The Journal of Open Source Software

# distr6: The Complete R6 Probability Distributions Interface

**Raphael E.B. Sonabend**[1] **and Franz J. Kiraly**[1, 2]

**1** Department of Statistical Science, University College London, Gower Street, London WC1E 6BT, United Kingdom **2** Shell

**DOI:** 10.21105/joss.0XXXX

**Software**

- Review
- Repository
- Archive

**Editor:** Editor Name

**Submitted:** 01 January 1900
**Published:** 01 January 3030



## Summary

The two gold-standard ways of interacting with probability distributions in `R` (R Core Team, 2017) is with the `d/p/q/r` functions in the `stats` (R Core Team, 2017) package or with distributions as objects in the `distr` (Ruckdeschel, Kohl, Stabla, & Camphausen, 2006) family of packages. `distr6` officially upgrades `distr` by using the state-of-the-art `R6` (Chang, 2018) object-oriented paradigm. It enables probability distributions to be used as objects, which is fundamental for probabilistic machine learning.

`distr6` makes use of novel ways to implement tried and tested design patterns (Gamma, Helm, Johnson, & Vlissides, 1996) to implement a clean, unified, extensible, and scalable interface for probability distributions. 42 probability distributions, with a further 11 kernels, are currently implemented, with many more in development. As well as these distributions, `distr6` allows extensions in the form of wrappers that can scale, truncate, or huberize distributions, and compositions such as mixtures and products.

`distr6` is currently being used in `mlr3proba` (Sonabend et al., 2019), which uses machine learning to predict probability distributions. Additional uses of `distr6` include sampling, and analysis of custom distributions via method imputation and visualisation.

The speed and efficiency of `R6`, combined with its scalability, allows `distr6` to be a complete interface for interacting with probability distributions as objects. `distr6` has the ambitious long-term goal of implementing all probability distributions defined throughout `R`.

Related software includes the `distr` (Ruckdeschel et al., 2006) family of packages, which uses the S4 object-oriented paradigm, `distributions3` (Hayes & Moller-Trane, 2019), which uses S3, and `Distributions.jl` (Lin et al., 2019), which is implemented in `Julia`. `distr6` was developed alongside the authors of `distr`.

## Key Design Principles

As `distr6` upgrades the `distr` family of packages, which has been available for over a decade, extensive discussions were had with the `distr` authors in order to learn from the experience of `distr` to provide the best possible user-journey in `distr6`. Therefore `distr6` adheres to the following design principles:

1. **Unified design interface** - Every implemented distribution/kernel as well as any custom distribution built by the user, has an identical interface. This helps make the package easy to navigate and the documentation simple to read.

2. **Separation of analytical and numerical results** - Implemented distributions only contain analytical results by default. Decorators (Gamma et al., 1996) are used so that numerical results can be imputed if analytical ones are unavailable. This allows users to guarantee precision of results, and to allow a choice of imputation methods.
3. **Inheritance but not over-inheritance** - Learning from the design choices of `distr`, `distr6` implements relatively few abstract classes for a simple inheritance structure. Deocorators, adaptors, and compositors (Gamma et al., 1996) are used to prevent over-inheritance, which with many distributions can lead to a complicated and messy class tree. This allows for the interface to be as flexible and scalable as possible.
4. **Full inspection and manipulation of distribution parameters** - `R stats` generally only allows one parameterisation for distributions, despite there often being several choices. `distr6` allows all common parameterisations for every distribution and after construction any parameter can be updated.
5. **Flexible object oriented (OO) paradigms** - `R6` is a new OO paradigm that relatively few packages currently use and it is appreciated that for new users there is a learning curve. Hence the package `R62S3` (Sonabend, 2019) is used to allow users to choose between calling methods with `R6` or `S3`, e.g. for some distribution `d`, both `d$pdf(1)` and `pdf(d, 1)` are possible.

# Key Use-Cases

1. **Constructing and querying probability distributions** - Currently 42 parameteric and non-parametric distributions can be constructed. Each can be queried for common representations, e.g. pdf, cdf, quantile, as well as simulating from the distribution. Additionally, mathematical methods are available, such as mean, variance, kurtosis, and skewness.
2. **Imputing numerical methods for custom user-build distributions** - Users can construct their own probability distributions with `Distribution$new`, decorators can then be used to impute numerical methods and functions. This is useful for understanding and learning properties of new distributions.
3. **Construction of composite distributions** - Wrappers in `distr6` exist for mixture, product, and vector distributions, as well as for truncation and scaling (and several more). Therefore distributions can be arbitrarily complex to serve any use-case.
4. **Probabilistic supervised learning** - `distr6` is used in the probabilistic machine learning package `mlr3proba` (Sonabend et al., 2019), which uses `distr6` in order to make supervised predictions of probability distributions.
5. **Visualisation** - The `plot` function can be used to visualise the shape of the probability distributions, with a choice of one or multiple representations, including the density, distribution, quantile, survival, hazard, and cumulative hazard function. Additionally the `qqplot` function can compare empirical distributions to any distribution implemented in `distr6`.

# Software Availability

`distr6` is available on GitHub and CRAN. It can either be installed from GitHub using the `devtools` (Wickham, Hester, & Chang, 2019) library or directly from CRAN with `install.packages`. The package uses the MIT open-source licence. Contributions, issues, feature requests, and general feedback can all be found and provided on the project GitHub. Full tutorials and further details are available on the project website.

## Acknowledgements

We acknowledge contributions from the authors of `distr`, Prof. Dr. Peter Ruckdeschel and Prof. Dr. Matthias Kohl, as well as a group of interns at UCL: Shen Chen, Jordan Deenichin, Chengyang Gao, Chloe Zhaoyuan Gu, Yunjie He, Xiaowen Huang, Shuhan Liu, Runlong Yu, Chijing Zeng and Qian Zhou.

## References

Chang, W. (2018). R6: Classes with Reference Semantics. Retrieved from https://cran.r-project.org/package=R6

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1996). Design Patterns: Elements of Reusable Software. doi:10.1093/carcin/bgs084

Hayes, A., & Moller-Trane, R. (2019). Distributions3. CRAN.

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

R Core Team. (2017). R: A Language and Environment for Statistical Computing. Vienna: R Foundation for Statistical Computing.

Ruckdeschel, P., Kohl, M., Stabla, T., & Camphausen, F. (2006). S4 Classes for Distributions. R News.

Sonabend, R. (2019, May). R62S3: Automatic Method Generation from R6. CRAN. Retrieved from https://cran.r-project.org/package=R62S3

Sonabend, R., Kiraly, F., & Lang, M. (2019, December). mlr3proba: Probabilistic Supervised Learning for 'mlr3'. CRAN. Retrieved from https://cran.r-project.org/package=mlr3proba

Wickham, H., Hester, J., & Chang, W. (2019). devtools: Tools to Make Developing R Packages Easier. CRAN.