The Journal of Open Source Software

# AutoHist.jl: A Julia package for fast and automatic histogram construction

**Oskar Høgberg Simensen** [1]

**1** Scientific Computing Center, Karlsruhe Institute of Technology

**DOI:** 10.21105/joss.08850

**Software**

- Review
- Repository
- Archive

**Editor:** Mehmet Hakan Satman

**Reviewers:**

- @PieterjanRobbe
- @ajinkya-k

**Submitted:** 17 July 2025
**Published:** 23 September 2025



## Summary

`AutoHist.jl` is a Julia (Bezanson et al., 2017) package for fitting histograms to univariate data, with an automatic and data-based selection of the histogram partition. It currently supports 8 irregular and 12 regular automatic histogram procedures from the statistics literature. Additionally, `AutoHist.jl` provides extensions for `Plots.jl` (Christ et al., 2023) and `Makie.jl` (Danisch & Krumbiegel, 2021), allowing for simple visualization of the resulting histogram fits.

## Statement of need

To this day, histograms remain one of the most widely used nonparametric density estimators. Their popularity is in no doubt due to their simplicity and interpretability, and they are routinely used by practitioners with limited mathematical backgrounds as a means of visualizing the distribution of data sets. Although the construction of a histogram for a given interval partition is a simple task, the quality of the resulting density estimate is very sensitive to the choice of bins. As a result, the task of designing automatic histogram procedures, where the number of bins and their location are chosen in a data-driven fashion, has received considerable interest in the statistics community. Despite advances in our understanding of different bin selection rules, many popular software libraries only include a limited number of simple rules for selecting a regular histogram partition, and typically do not provide any support for automatic irregular histogram construction. `AutoHist.jl` fills this gap by providing a fast implementation of state-of-the-art regular and irregular bin selection algorithms from the statistics literature. A complete overview of the bin selection procedures that have been implemented so far is given in **Table 1**.

**Table 1:** Implemented bin selection procedures so far. For methods with type=both, a regular and an irregular variant of the criterion is supported.

| Rule | Type (regular/irregular) | Reference |
|---|---|---|
| Sturges' rule | regular | Sturges (1926) |
| Scott's rule | regular | Scott (1979) |
| Freedman & Diaconis' rule | regular | Freedman & Diaconis (1981) |
| Akaike's Information Criterion (AIC) | regular | Hall (1990) |
| Bayesian Information Criterion (BIC) | regular | Davies et al. (2009) |
| Birgé & Rozenholc's rule | regular | Birgé & Rozenholc (2006) |
| Minimum Description Length (MDL) | regular | Hall & Hannan (1988) |
| Wand's rule | regular | Wand (1997) |
| Random Regular Histogram (RRH), Knuth | regular | Knuth (2019) |
| Rozenholc et al. penalty A | irregular | Rozenholc et al. (2010) |

| Rule | Type (regular/irregular) | Reference |
|---|---|---|
| Rozenholc et al. penalty B | irregular | Rozenholc et al. (2010) |
| Rozenholc et al. penalty R | irregular | Rozenholc et al. (2010) |
| Bayesian Blocks | irregular | Scargle et al. (2013) |
| Random Irregular Histogram (RIH) | irregular | Simensen et al. (2025) |
| Normalized Maximum Likelihood (NML) | both | Kontkanen & Myllymäki (2007) |
| L2 cross-validation (L2CV) | both | Rudemo (1982) |
| Kullback-Leibler cross-validation (KLCV) | both | Hall (1990), Simensen et al. (2025) |

We note that some automatic histogram selection rules have been implemented in Julia and in other programming languages, typically as part of plotting libraries. The `Plots.jl` package and the `hist` function from base R (R Core Team, 2025) include support for some plug-in rules used to construct regular histograms, including Sturges' rule, Scott's rule, and Freedman & Diaconis' rule. Python's `matplotlib` (Hunter, 2007) implements regular L2 cross-validation in addition to the three aforementioned rules. In Julia, the `StatsPlots.jl` library (Christ et al., 2023) also includes an implementation of equal-area histograms. The R package `histogram` (Mildenberger et al., 2019) supports a wider range of more sophisticated regular and irregular histogram methods, but their implementation covers fewer criteria than `AutoHist.jl`.

# Installation and usage

The `AutoHist` package is part of the Julia general registry, and can as such be installed via the built-in package manager,

```
using Pkg
Pkg.add("AutoHist")
```

To illustrate the basic use of the software, we fit a histogram based on the Random Irregular Histogram criterion and a regular histogram based on Akaike's Information Criterion to a standard Normal random sample of size $n = 10^6$.

```
using AutoHist, Random, Distributions

n = 10^6
x = rand(Xoshiro(1812), Normal(), n)         # synthetic data

h_irr = fit(AutomaticHistogram, x, RIH())    # fit an irregular histogram
h_reg = fit(AutomaticHistogram, x, AIC())    # fit a regular histogram
```

The call to the `fit` method returns an object of type `AutomaticHistogram`, with fields recording the chosen histogram partition, estimated density and bin counts. `AutoHist.jl` provides plotting recipes for `Plots.jl` and `Makie.jl`, which allows the user to easily visualize the fit via `Plots.plot(h_irr)` or `Makie.plot(h_reg)`. Below, we show in more detail how to plot the irregular and the regular histogram fitted in the above code snippet using `Makie`.

```
import CairoMakie, Makie # using the CairoMakie backend

fig = Makie.Figure(size=(670, 320))
ax1 = Makie.Axis(fig[1, 1], title="Irregular histogram", xlabel="x",
                 ylabel="Density")
ax2 = Makie.Axis(fig[1, 2], title="Regular histogram", xlabel="x")
p_irr = Makie.plot!(ax1, h_irr, alpha=0.4, color="black")
```

```
p_reg = Makie.plot!(ax2, h_reg, alpha=0.4, color="red")
fig
```

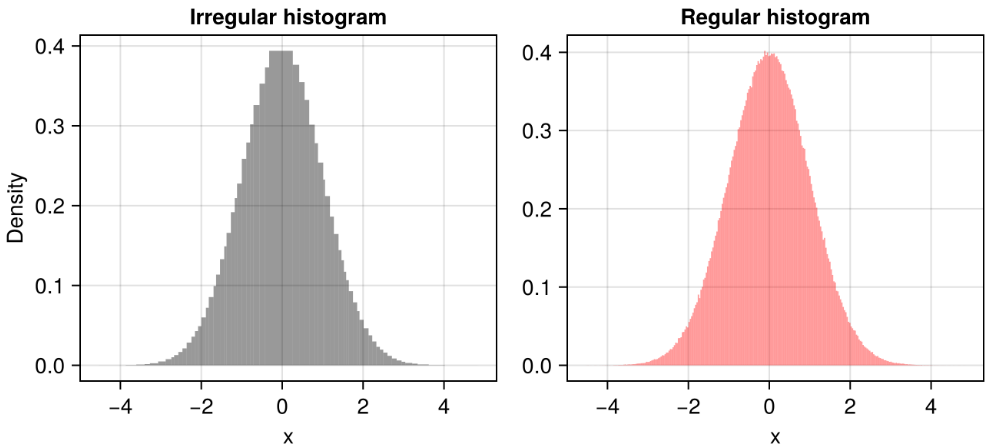


**Figure 1:** Plot of the irregular and regular histogram fit to the standard Normal sample.


# Acknowledgements

The author thanks the two reviewers for their valuable suggestions that improved the software and this paper.


# References


Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Birgé, L., & Rozenholc, Y. (2006). How many bins should be put in a regular histogram. *ESAIM: Probability and Statistics*, *10*, 24–45. https://doi.org/10.1051/ps:2006001

Christ, S., Schwabeneder, D., Rackauckas, C., Borregaard, M. K., & Breloff, T. (2023). *Plots.jl - A user extendable plotting API for the Julia programming language*. https://doi.org/10.5334/jors.431

Danisch, S., & Krumbiegel, J. (2021). Makie.jl: Flexible high-performance data visualization for Julia. *Journal of Open Source Software*, *6*, 3349. https://doi.org/10.21105/joss.03349

Davies, P. L., Gather, U., Nordman, D., & Weinert, H. (2009). A comparison of automatic histogram constructions. *ESAIM: Probability and Statistics*, *13*, 181–196. https://doi.org/10.1051/ps:2008005

Freedman, D., & Diaconis, P. (1981). On the histogram as a density estimator: L2 theory. *Zeitschrift Für Wahrscheinlichkeitstheorie Und Verwandte Gebiete*, *57*, 453–476. https://doi.org/10.1007/BF01025868

Hall, P. (1990). Akaike's information criterion and Kullback–Leibler loss for histogram density estimation. *Probability Theory and Related Fields*, *85*, 449–467. https://doi.org/10.1007/BF01203164

Hall, P., & Hannan, E. J. (1988). On stochastic complexity and nonparametric density estimation. *Biometrika*, *75*, 705–714. https://doi.org/10.1093/biomet/75.4.705

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science &*

*Engineering*, *9*, 90–95. https://doi.org/10.1109/MCSE.2007.55

Knuth, K. H. (2019). Optimal data-based binning for histograms and histogram-based probability density models. *Digital Signal Processing*, *95*. https://doi.org/10.1016/j.dsp.2019.102581

Kontkanen, P., & Myllymäki, P. (2007). MDL histogram density estimation. *Proceedings of Machine Learning Research*, *2*, 219–226. https://proceedings.mlr.press/v2/kontkanen07a.html

Mildenberger, T., Rozenholc, Y., & Zasada, D. (2019). *Histogram: Construction of regular and irregular histograms with different options for automatic choice of bins*. https://doi.org/10.32614/cran.package.histogram

R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://doi.org/10.32614/r.manuals

Rozenholc, Y., Mildenberger, T., & Gather, U. (2010). Combining regular and irregular histograms by penalized likelihood. *Computational Statistics & Data Analysis*, *54*, 3313–3323. https://doi.org/10.1016/j.csda.2010.04.021

Rudemo, M. (1982). Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, *9*, 65–78. https://www.jstor.org/stable/4615859

Scargle, J. D., Norris, J. P., Jackson, B., & Chiang, J. (2013). Studies in astronomical time series analysis. VI. Bayesian block representations. *The Astrophysical Journal*, *764*, 1–26. https://doi.org/10.1088/0004-637X/764/2/167

Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika*, *66*, 605–610. https://doi.org/10.1093/biomet/66.3.605

Simensen, O. H., Christensen, D., & Hjort, N. L. (2025). Random irregular histograms. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2505.22034

Sturges, H. A. (1926). The choice of a class interval. *Journal of the American Statistical Association*, *21*, 65–66. https://doi.org/10.1080/01621459.1926.10502161

Wand, M. P. (1997). Data-based choice of histogram bin width. *The American Statistician*, *51*, 59–64. https://doi.org/10.2307/2684697