The Journal of Open Source Software

LongMemory.jl: Generating, Estimating, and Forecasting Long Memory Models in Julia

J. Eduardo Vera-Valdés [1]

1 Aalborg University, Department of Mathematical Sciences, Aalborg, Denmark

DOI: 10.21105/joss.07708

Software

- Review
- Repository
- Archive

Editor: Oskar Laverny

Reviewers:

- @PieterjanRobbe
- @Santymax98
- @baxmittens

Submitted: 14 January 2025
Published: 31 March 2025

Summary

In time series analysis, the concept of *Long Memory* encompasses datasets including a strong dependency on past values. Hurst (1956) is one of the pioneering works on the subject from the field of Hydrology: while analyzing the flow of the Nile river, he noted that water reservoirs that do not account for its long-term dynamics were still at risk of overflowing. Long memory models are generally used in climate, finance, biology, economics, and many other fields. See Beran et al. (2013) for a textbook on the subject.

We say that a stationary time series x_t has long memory with parameter $-1/2 < d < 1/2$ if it has an autocovariance function $\gamma_x(k)$ that behaves as:

$$\gamma_x(k) \sim C_x k^{2d-1} \quad \text{as} \quad k \to \infty,$$

or if it has an spectral density function $f_x(\lambda)$ that behaves as:

$$f_x(\lambda) \sim C_f \lambda^{-2d} \quad \text{as} \quad \lambda \to 0,$$

where both C_x and C_f are constants. Above equivalences $g(x) \sim h(x)$ as $x \to x_0$ holds when $g(x)/h(x)$ converges to 1 as x tends to x_0.

Both properties above can be analyzed graphically by plotting the autocorrelation and periodogram (an estimator of the spectral density), respectively. As an example, the figure below shows the autocorrelation and periodogram (in logs) for the Nile River minima data. The data are available in `LongMemory.jl` through the `NileData()` function.

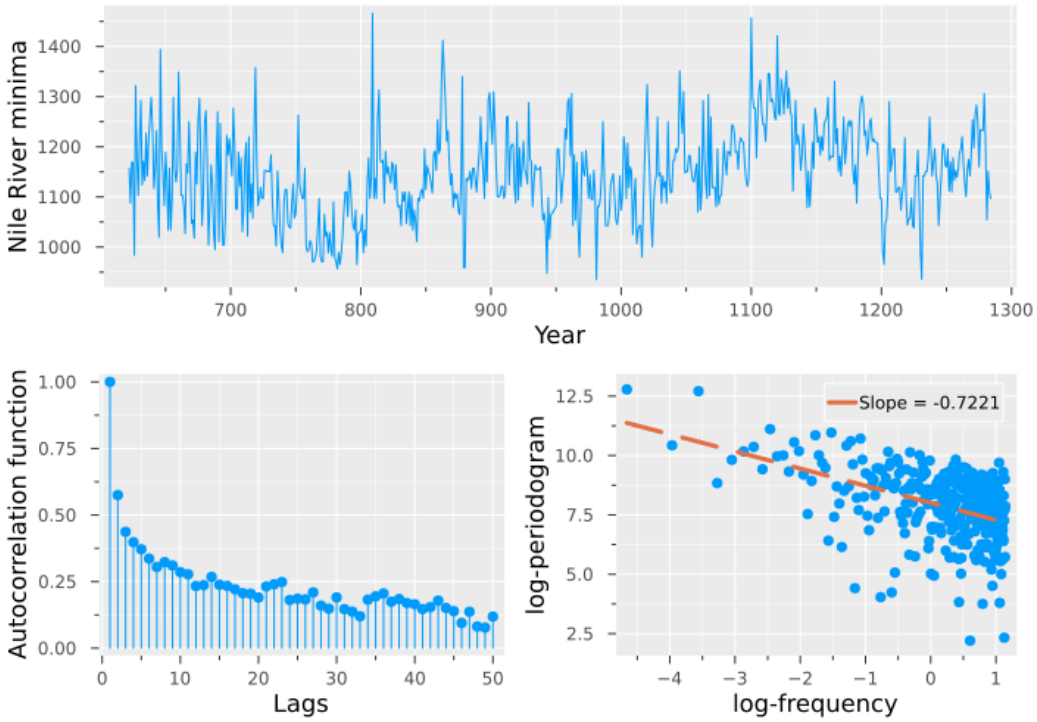

Figure 1: Nile River minima (top), its autocorrelation function (bottom left), and log-periodogram (bottom right)

As the figure shows, the autocorrelation function decays slowly and the periodogram diverges towards infinity near the origin. These are standard features of long memory processes as we just defined them. The `LongMemory.jl` package is concerned with methods for modeling data with this type of behavior.

The following code generates the figure. The `autocorrelation_plot()` and `periodogram_plot()` functions are part of the `LongMemory.jl` package.

```
using LongMemory, Plots
NileMin = NileData()
theme(:ggplot2)
p1 = plot( NileMin.Year , NileMin.NileMin, xlabel="Year",
      ylabel = "Nile River minima" , legend = false )
p2 = autocorrelation_plot(NileMin.NileMin, 50)
p3 = periodogram_plot(NileMin.NileMin, slope = true)
l = @layout [a; b c]
plot(p1, p2, p3, layout = l, size = (700, 500) )
```

Statement of need

`LongMemory.jl` is a package for time series long memory modeling in Julia (Bezanson et al., 2017). The package provides functions to generate long memory time series, estimate model parameters, and forecast data. Generating methods include fractional differencing, stochastic error duration, and cross-sectional aggregation. Estimators include the classic ones used to estimate the Hurst effect, those inspired by log-periodogram regression, and parametric ones. Forecasting is provided for all parametric estimators. Moreover, the package adds plotting capabilities to illustrate long memory dynamics and forecasting. For some of the theoretical developments, `LongMemory.jl` provides the first publicly available implementation

in any programming language. A notable feature of this package is that all functions are implemented in the same programming language, taking advantage of the ease of use and speed provided by Julia. Therefore, all code is accessible to the user. Multiple dispatch, a novel feature of the language, is used to speed computations and provide consistent calls to related methods.

Comparison to existing packages

This section presents benchmarks contrasting different implementations to show the computational efficiency of `LongMemory.jl`. The package is related to the R (R Core Team, 2023) packages `LongMemoryTS` (Leschinski, 2019) and `fracdiff` (Maechler, 2022). The former is the most similar to `LongMemory.jl` in terms of functionality, while the latter is the most popular package for long memory time series in R, measured by the number of CRAN downloads.

The following table presents the summary of the benchmarks. The code for the benchmarks is available on the author's website. The benchmarks were made using `BenchmarkTools.jl` (Chen & Revels, 2016).

Table 1: Comparison of function performance. All sample sizes are 10^4.

Functionality	Package (Language)	Function	Mean	Median
Generation	LongMemory.jl (Julia)	`fi_gen`	7.048E+05	5.812E+05
	fracdiff (R)	`fracdiff.sim`	1.017E+08	1.010E+08
Fractional Differencing	LongMemory.jl (Julia)	`fracdiff`	7.101E+05	5.824E+05
	fracdiff (R)	`diffseries`	2.124E+06	1.892E+06
	LongMemoryTS (R)	`fdiff`	4.150E+06	3.950E+06
Estimation	LongMemory.jl (Julia)	`gph_est`	4.165E+04	3.330E+04
	LongMemoryTS (R)	`gph`	1.662E+05	1.471E+05
	fracdiff (R)	`fdGPH`	7.058E+06	6.022E+06

For long memory generation, the table shows the benchmarks for the function `fi_gen` in `LongMemory.jl` and the function `fracdiff.sim` in the package `fracdiff`. `LongMemoryTS` does not provide a function to directly generate processes with long memory. The results show that `LongMemory.jl` is more than 10^2 times faster than `fracdiff` at this sample size. Regarding fractional differencing, the table shows the benchmarks for the function `fracdiff` in `LongMemory.jl` and the functions `diffseries` and `fdiff` in packages `fracdiff` and `LongMemoryTS`, respectively. Note that `LongMemory.jl` is the fastest implementation by a large margin. Finally, for long memory estimation, the table shows the benchmarks for the function `gph_est` in `LongMemory.jl` and the functions `fdGPH` and `gph` in packages `fracdiff` and `LongMemoryTS`, respectively. The benchmarks show that `LongMemory.jl` is significantly faster, taking advantage of the speed of Julia.

Interestingly, there does not seem to be a package in Python that provides the same functionality as `LongMemory.jl`.

Examples of Research Conducted with `LongMemory.jl`

The package has been used in both (J. E. Vera-Valdés & Kvist, 2024) and (Vera-Valdés J. E. & Kvist, 2025). Furthermore, see the accompanying examples of the package in use at the author's website.

References

Beran, J., Feng, Y., Ghosh, S., & Kulik, R. (2013). Long-memory processes: Probabilistic theories and statistical methods. In *Springer Verlag* (pp. 1–892). Springer. https://doi.org/10.1007/978-3-642-35512-7

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

Chen, J., & Revels, J. (2016). Robust benchmarking in noisy environments. *arXiv e-Prints*. https://doi.org/10.48550/arXiv.1608.04295

Hurst, H. E. (1956). The problem of long-term storage in reservoirs. *International Association of Scientific Hydrology. Bulletin*, *1*, 13–27. https://doi.org/10.1080/02626665609493644

Leschinski, C. (2019). 'LongMemoryTS': Long memory time series. *R Package Version 0.1.0*. https://doi.org/10.32614/CRAN.package.LongMemoryTS

Maechler, M. (2022). 'Fracdiff': Fractionally differenced ARIMA models. *R Package Version 1.5-2*. https://doi.org/10.32614/CRAN.package.fracdiff

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Vera-Valdés, J. E., & Kvist, O. (2024). Breaching 1.5°C: Give me the odds. *arXiv*. https://doi.org/10.48550/arXiv.2412.13855

Vera-Valdés, J. E., & Kvist, O. (2025). Effects of the paris agreement and the COVID-19 pandemic on volatility persistence of stocks associated with the climate crisis: A multiverse analysis. *Advances in Econometrics*, *Forthcoming*. https://everval.github.io/publications/FinancialVolatilityAfterPA.html