

# nsEVDx: A Python library for modeling non-Stationary extreme value distributions

2025-07-08

## Summary

**nsEVDx** is an open-source Python package for fitting stationary and non-stationary Extreme Value Distributions (EVDs) to extreme value data. It can be used to model extreme events in fields like hydrology, climate science, finance, and insurance, using both frequentist and Bayesian methods. For Bayesian inference it employs advanced Monte Carlo sampling techniques such as the Metropolis-Hastings, Metropolis-adjusted Langevin (MALA), and Hamiltonian Monte Carlo (HMC) algorithms. Unlike many existing extreme value theory (EVT) tools, which can be complex or lack Bayesian options, **nsEVDx** offers an intuitive, python-native interface that is both user-friendly and extensible. It requires only standard scientific python libraries (**numpy**, **scipy**) for its core functionality, while optional features like plotting and diagnostics use **matplotlib** and **seaborn**. A key feature of **nsEVDx** is its flexible support for non-stationary modeling, where the location, scale, and shape parameters can each depend on arbitrary, user-defined covariates. This enables practical applications such as linking extremes to other variables (e.g., rainfall extremes to temperature or maximum stock market losses to market volatility indices). Overall, **nsEVDx** aims to serve as a practical, easy-to-use, and extensible tool for researchers and practitioners analyzing extreme events in non-stationary environments.

## Statement of Need

Probabilistic modeling of extreme events is essential across disciplines, from resilient infrastructure design and climate adaptation to insurance pricing and financial risk management. In many real-world processes, the statistical properties of the extremes are often non-stationary, driven by long-term changes such as climate change, urbanization, or economic shifts. Accurately estimating return periods and risks under these evolving conditions requires fitting non-stationary extreme value distributions (EVDs) to observations.

Several R packages currently support EVD modeling, including `ismev` (Heffernan J. E. et al., 2003), `extRemes` (Gilleland, 2025), `climextRemes` (Paciorek, 2016), and `NSGEV` (IRSN, 2024). However, these packages differ in their ability to handle non-stationary models and Bayesian inference. Moreover, extending their functionality and integrating modern inference techniques can be challenging. Probabilistic programming frameworks, such as python-based `PyMC` (Oriol Abril-Pla et al., 2023), and C++ based Stan with interfaces like `PyStan` (Stan development Team, 2023b) and `CmdStan` (Stan development Team, 2023a), offer powerful tools for building custom statistical models, including those for extreme value analyses. However, these tools require significant expertise in both statistics and programming to develop, tune, and validate the models effectively. As a result, they may be too complex for domain experts like hydrologists, climate scientists, or risk analysts seeking easy-to-use methods.

Based on this synopsis, there is a clear need for a python tool that balances flexibility and ease of use, while supporting arbitrary covariates, parameter constraints, custom priors, and advanced MCMC algorithms such as MALA (Roberts & Tweedie, 1996), HMC (Betancourt, 2017), for fitting non-stationary Generalized Extreme Value (GEV) and Generalized Pareto (GPD) distributions, the two most prominent EVDs.

To bridge this gap, we developed `nsEVDx`, a flexible, user-friendly python package that streamlines non-stationary EVD modeling without compromising statistical rigor. Developed as part of N.Kafle’s PhD research, `nsEVDx` has been applied in hydrology (Kafle & Meier, 2025) and is applicable to fields like climate science, finance, and engineering, where it is critical to understand the frequency and intensity of extremes under non-stationarity conditions. Its application is also reflected in an upcoming technical paper on trends in short-duration extreme rainfall in the Southeastern U.S. (Kafle & Meier, in preparation).

## Features

- Supports both the Generalized Extreme Value (GEV) and Generalized Pareto (GPD) distributions
- Non-stationary modeling via linear and log-linear relationships between parameters and covariates
- Independent non-stationarity in location, scale, and shape parameters
- Frequentist and Bayesian inference support
- MCMC algorithms: Random Walk, Metropolis-adjusted Langevin (MALA), and Hamiltonian Monte Carlo (HMC)
- Custom priors, parameter bounds, and temperature scaling for tuning MCMC
- Integrated diagnostics: trace plots, convergence checks, and posterior visualization
- Modular and extensible API designed for ease of use by domain scientists
- Bayesian metrics and likelihood ratio tests

## Implementation

The core class `NonStationaryEVD` handles parameter parsing, log-likelihood construction, prior specification, and proposal generation. Frequentist method uses `scipy.optimize` to minimize the non-stationary negative log likelihood, while the Bayesian MCMC methods are implemented from scratch in numpy, allowing full transparency and customization. The concepts of non-stationarity and MCMC techniques used in `nsEVDx` are based on the foundational texts by Robert & Casella (2009) and Coles (2007). The implementation of L-moments in some utility methods follows the approach described by Hosking & Wallis (1997). Currently, `nsEVDx` supports linear modeling for the location and shape parameters, and exponential (log-linear) modeling for the scale parameter, to ensure positivity.

Non-stationarity is controlled via a configuration vector `config = [a, b, c]`, where each entry specifies the number of covariates used to model the location, scale, and shape parameters of the EVD. Entry with a value of 0 implies stationarity (i.e., no covariate dependence), while integer values  $> 0$  indicate non-stationary modeling using the corresponding number of covariates for the parameter.

In Bayesian estimation, `nsEVDx` can infer prior specifications based on the data and configuration or accept user-defined priors. In the frequentist mode, it can determine suitable parameter bounds automatically. However, user-defined priors or bounds are recommended for better convergence and interpretability.

Future updates will potentially include mixed population models using categorical covariates to represent different distributions, an emerging area in hydroclimatic extremes. Additionally, efforts to optimize and accelerate code execution for faster runtimes are planned.

## Installations

Install the package via pip: `pip install nsEVDx`

or alternatively, clone the repository and install manually:

```
git clone https://github.com/Nischalcs50/nsEVDx.git
cd nsEVDx
pip install .
```

## Example usage

```
from nsEVDx import NonStationaryEVD
from scipy.stats import genextreme

sampler = NonStationaryEVD(data,
```

```

covariate, config=[1,0,0],
dist=genextreme)
# config = [1,0,0] means, location parameter is modeled linearly
# with covariate, while scale and shape are treated as stationary
# Priors are inferred from the data if not provided while
# declaring the sampler
Print(sampler.descriptions) # provides the parameter descriptions
samples, acceptance_rate = sampler.MH_RandWalk(
    num_samples=10000,
    initial_params=[10, 0.02, 5, 0.1],
    # B0(location intercept), B1(location slope), scale, shape
    proposal_widths=[0.01, 0.001, 0.01, 0.001],
    T=1.0
)

```

See full documentation at: <https://github.com/Nischalcs50/nsEVDx/docs/API.md>

## Acknowledgements

I gratefully acknowledge the support and encouragement of my wife, Koshika Timsina, whose constant belief in me has been a source of strength throughout this project. I also extend my heartfelt thanks to my family for their unwavering love, patience, and support.

## References

- Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv: Methodology*.
- Coles, S. (2007). *An introduction to statistical modeling of extreme values* (4th. printing). Springer. <https://doi.org/10.1007/978-1-4471-3675-0>
- Gilleland, E. (2025). *extRemes: Extreme Value Analysis*. <https://doi.org/10.1175/JTECH-D-20-0070.1>
- Heffernan J. E., Stephenson A.G., & Gilleland E. (2003). *Ismev: An Introduction to Statistical Modeling of Extreme Values*. <https://CRAN.R-project.org/package=ismev>
- Hosking, J. R. M., & Wallis, J. R. (1997). *Regional Frequency Analysis: An Approach Based on L-Moments* (Vol. 93). Cambridge University Press. <https://ui.adsabs.harvard.edu/abs/1997rfa..book.....H>
- IRSN. (2024). *NSGEV: Non-Stationary GEV Time Series*. <https://github.com/IRSN/NSGEV/>
- Kaffe, N., & Meier, C. (2025). Evaluating Methodologies for Detecting Trends in Short-Duration Extreme Rainfall in the Southeastern United States. *Extreme Hydrological or Critical Event Analysis-III, EWRI Congress 2025, Anchorage, AK, U.S.* <https://alaska2025.eventscribe.net>

- Kafle, N., & Meier, C. (in preparation). Detecting trends in short duration extreme precipitation over SEUS using neighborhood based method. *Manuscript in Preparation*.
- Oriol Abril-Pla, Virgile Andreani, C. Carroll, L. Y. Dong, Christopher Fonnesbeck, Maxim Kochurov, Ravin Kumar, Junpeng Lao, Christian C. Luhmann, Osvaldo A. Martin, Michael Osthege, Ricardo Vieira, Thomas V. Wiecki, & Robert Zinkov. (2023). PyMC: A modern, and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, 9, e1516–e1516. <https://doi.org/10.7717/peerj-cs.1516>
- Paciorek, C. (2016). *climextRemes: Tools for Analyzing Climate Extremes*. <https://CRAN.R-project.org/package=climextRemes>
- Robert, C. P., & Casella, G. (2009). *Introducing Monte Carlo Methods with R*. <https://doi.org/10.1007/978-1-4419-1576-4>
- Roberts, G. O., & Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4), 341. <https://doi.org/10.2307/3318418>
- Stan development Team. (2023a). *CmdStan: The command-line interface to Stan*. <https://mc-stan.org/users/interfaces/cmdstan>
- Stan development Team. (2023b). *PyStan: The python interface to Stan*. <https://pystan.readthedocs.io/>