The Journal of Open Source Software

nsEVDx: A Python Library for Stationary and Non-Stationary Extreme Value Modeling

Nischal Kafle [1¶] **and Claudio Meier** [1]

1 Department of Civil, Construction, and Environmental Engineering, University of Memphis, TN, USA
¶ Corresponding author

DOI: 10.21105/joss.11187

Software

- Review
- Repository
- Archive

Editor: Sébastien Boisgérault

Reviewers:

- @manunicholasjacob
- @dosvk

Submitted: 09 July 2026
Published: 05 September 2026

Summary

nsEVDx is an open-source Python package for fitting stationary and non-stationary Extreme Value Distributions (EVDs) to extreme value data. It supports non-stationary models in which the location, scale, and shape parameters can each depend on user-defined covariates. Overall, nsEVDx provides a Python-native interface for frequentist and Bayesian inference of non-stationary EVD models, enabling researchers and practitioners to quantify changing risks associated with rare and extreme events.

Statement of Need

Traditional extreme value frequency analysis assumes that the statistical properties of extreme events remain stationary over time. However, non-stationary behavior in extreme events has been documented across hydroclimatic, engineering, and financial applications. For example, in hydroclimatic systems, non-stationary behavior has been documented in flood and extreme-rainfall processes, with changes linked to factors such as urbanization and climate change (Eccles et al., 2025; Jayaweera et al., 2025; Prosdocimi et al., 2015). Such changes can alter the frequency and magnitude of extreme events over time. Accurately quantifying these evolving risks and return periods requires statistical tools that can incorporate arbitrary, time-varying covariates into all EVD parameters.

Although several existing packages support extreme value analysis, modeling non-stationary EVDs requires substantial statistical and programming expertise, particularly when Bayesian sampling algorithms are involved. `nsEVDx` provides an easy-to-use framework for fitting stationary and non-stationary Generalized Extreme Value (GEV) and Generalized Pareto (GPD) distributions. It makes advanced sampling methods accessible to users with varying levels of expertise in extreme value modeling while retaining the flexibility required for advanced applications.

The target audience for nsEVDx includes hydrologists, climate scientists, engineers, and risk analysts who need a reliable, flexible workflow for non-stationary extreme value modeling. By reducing the effort needed to implement both frequentist optimization and advanced Markov Chain Monte Carlo (MCMC) sampling algorithms, the software allows practitioners to focus on understanding changing risks and their implications for decision-making.

State of the Field

Extreme value distributions (EVDs), particularly the GEV and GPD, are widely used to quantify the frequency and magnitude of rare events in hydrology, climatology, engineering, and finance (Castillo, 1988; Kafle et al., 2026; Katz et al., 2002; McNeil & Frey, 2000). Increasing evidence

of non-stationarity in environmental and socioeconomic systems has motivated the development of methods that allow EVD parameters to vary with time or external covariates.

Several software packages support extreme value analysis. In the R ecosystem, `ismev` (Heffernan J. E. et al., 2003), `extRemes` (Gilleland, 2025), `climextRemes` (Paciorek, 2016), and `NSGEV` (Deville, 2026) provide tools for fitting stationary and non-stationary extreme value models. These packages provide important foundations for EVD-based analysis, although their capabilities differ in areas such as non-stationary modeling and Bayesian inference. Probabilistic programming frameworks, such as Python-based `PyMC` (Abril-Pla et al., 2023) and `NumPyro` (Phan et al., 2019), and `C++`-based Stan with interfaces like `PyStan` (Stan Development Team, 2023b) and `CmdStan` (Stan Development Team, 2023a), offer flexible tools for building custom statistical models, including extreme value models. However, these tools require users to formulate non-stationary likelihood functions explicitly for EVDs along with prior specifications, parameter constraints, and sampling configurations for each application. As a result, they may be too complex for domain-specific practitioners like hydrologists, climate analysts, or risk analysts seeking easy-to-use tools. Within Python, `pyextremes` (Bocharov, 2020) supports extreme value analysis with features such as block maxima and peaks-over-threshold extraction, return-level estimation, frequentist inference through `SciPy`, and Bayesian inference using the affine-invariant ensemble sampler, `emcee` (Foreman-Mackey et al., 2013). However, `pyextremes` is primarily focused on stationary extreme value workflows.

These differences motivate `nsEVDx`, a flexible Python tool that provides a unified framework for both stationary and non-stationary modeling. In particular, `nsEVDx` supports user-defined covariates, parameter constraints, custom priors, and MCMC algorithms such as Metropolis-Adjusted Langevin Algorithm (MALA) and Hamiltonian Monte Carlo (HMC) for fitting non-stationary GEV and GPD models (Betancourt, 2017; Neal, 2011; Roberts & Tweedie, 1996). Compared with general-purpose probabilistic programming frameworks, `nsEVDx` provides predefined EVD likelihoods, parameterizations, and inference workflows, reducing the amount of model construction required for common extreme value analyses.

Software Design

`nsEVDx` is designed to minimize the programming burden typically associated with custom Bayesian implementations while providing a transparent and flexible framework for extreme value modeling. The overall workflow implemented in `nsEVDx` is shown in Figure 1.

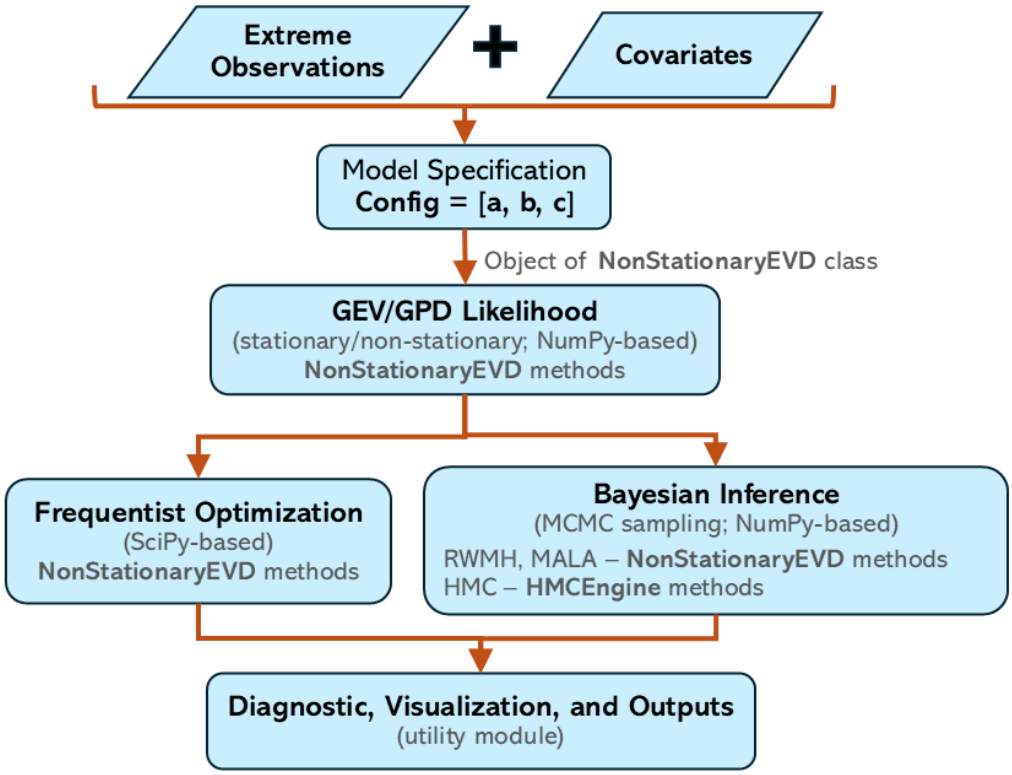

Figure 1: Workflow implemented in `nsEVDx` for specifying, fitting, and diagnosing stationary and non-stationary GEV and GPD models using frequentist optimization or Bayesian MCMC sampling

The concepts of non-stationarity and MCMC techniques used in `nsEVDx` are based on the foundational texts by Robert & Casella (2009) and Coles (2001). The core class `NonStationaryEVD` handles parameter parsing and specification, log-likelihood evaluation, prior assignment, optimization, and MCMC sampling. This unified design allows users to switch between frequentist and Bayesian workflows without redefining model structures. Frequentist methods use `scipy.optimize` to minimize the non-stationary negative log-likelihood, while the Bayesian MCMC methods are implemented in numpy, allowing full transparency and customization. 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. 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.

`nsEVDx` controls non-stationarity 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. An 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. This representation provides a compact and scalable mechanism for defining a wide range of stationary and non-stationary models while maintaining a consistent user interface (Figure 2).

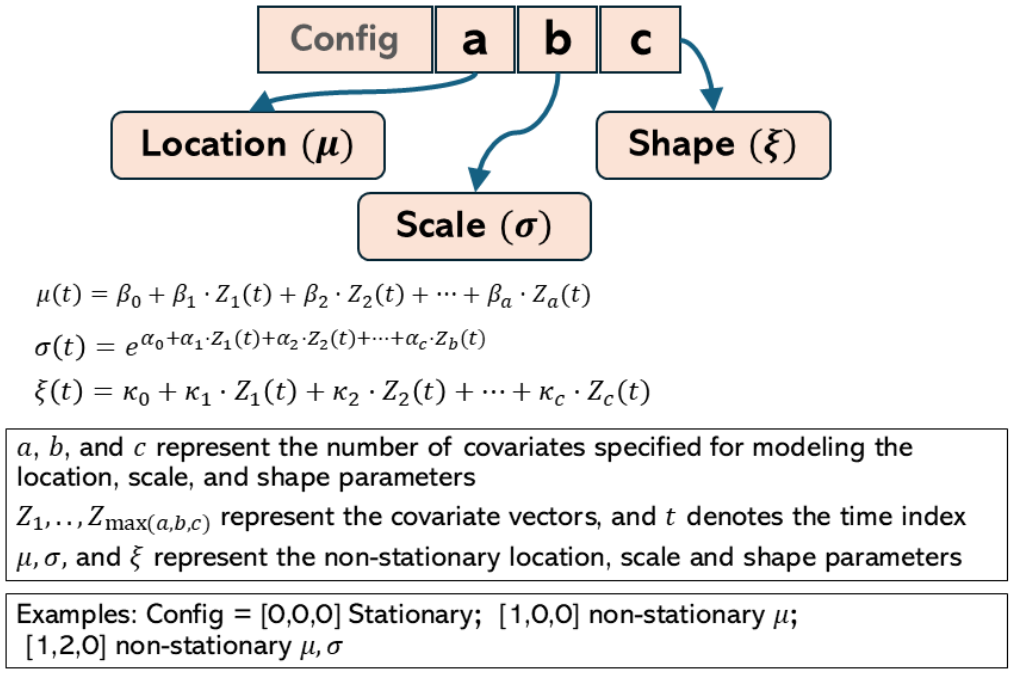

Figure 2: Configuration-vector framework used in `nsEVDx` for specifying stationary and non-stationary EVD models.

Currently, location and shape parameters can be modeled using linear relationships, while the scale parameter is modeled using an exponential link function to ensure positivity. Polynomial relationships can be accommodated by transforming covariates before model fitting.

Another key design decision is the implementation of Bayesian samplers directly in `NumPy`. This approach maximizes transparency, enables users to inspect and modify algorithmic details, and reduces dependencies. The package currently provides Random-Walk Metropolis-Hastings (RWMH), MALA, and HMC samplers, allowing users to balance computational efficiency and inferential robustness according to their application. While RWMH and MALA are integrated within the `NonStationaryEVD` class, HMC is implemented through a dedicated `HMCEngine` component that handles Hamiltonian dynamics.

To support model evaluation and reproducibility, `nsEVDx` includes diagnostic tools for trace plots, posterior summaries, acceptance rates, and Gelman-Rubin ($\hat{R}$) convergence metrics. It also provides model selection options, including DIC, AIC, BIC, and likelihood-ratio tests, enabling complete workflows within a single environment.

The package is intentionally lightweight, relying primarily on `NumPy` and `SciPy` for computation and `Matplotlib` and `Seaborn` for visualization. This minimal dependency footprint improves portability and simplifies installation while preserving extensibility for future methodological developments.

Benchmark Experiment

A simulation-based case study was performed to compare mean absolute errors (MAE) in estimating EVD parameters across multiple packages, including `nsEVDx`. First, we generated $N = 100$ stationary and $N = 50$ non-stationary datasets, each consisting of 500 extreme values sampled using fixed EVD parameter sets (hereafter referred to as the "true" parameter; Figure 3). These datasets were then used to fit EVD models, and MAE was computed by comparing estimated parameters with the corresponding true parameters. For stationary GEV and GPD

models, nsEVDx produced parameter estimates comparable to those obtained with SciPy and pyextremes using both maximum likelihood estimation (MLE) and L-moments (LM) (Figure 3a-b). For non-stationary models, nsEVDx produced parameter estimates comparable to those obtained from an equivalent NumPyro-based NUTS implementation using the same priors and MCMC settings (number of chains, posterior samples, and burn-in period), demonstrating reliable recovery of non-stationary parameters in these simulated examples (Figure 3c-d). Across methods and scenarios, shape parameters generally exhibited the lowest MAE, while differences in the MAE of other parameters were relatively small.

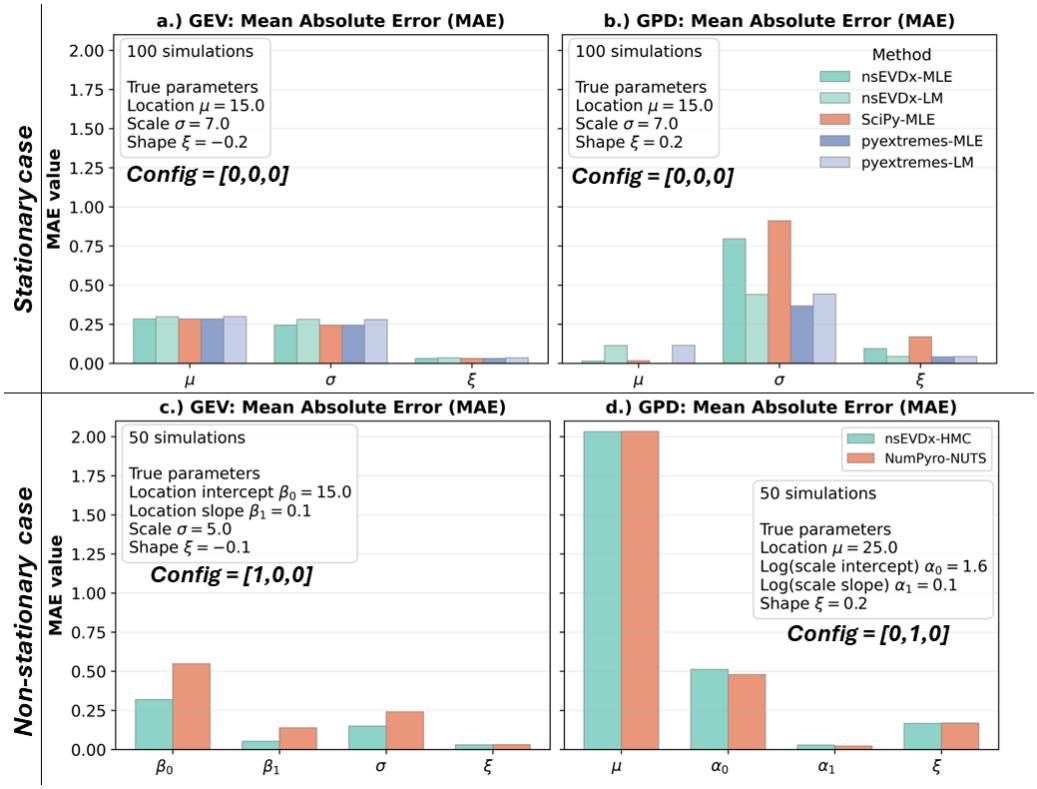

Figure 3: Mean of absolute error in estimating GEV and GPD parameters across the simulations (each using a vector of 500 random values generated using the true parameters) for various methods and Python packages.

Research Impact Statement

nsEVDx lowers the barrier to applying advanced Bayesian methods, including MALA and HMC sampling, by providing predefined extreme value models and inference workflows that accommodate users with different levels of statistical and programming expertise. The package has already been used in ongoing hydroclimatic research focused on trend detection and characterization of extreme precipitation (Kafle et al., 2025; Kafle, 2026). Potential applications extend to climate science, water resources engineering, infrastructure design, and financial risk assessment, e.g., linking rainfall extremes to temperature or maximum stock market losses to market volatility indices. A steady increase in downloads suggests growing interest among researchers and practitioners. The software is openly developed, tested, documented, and distributed through both GitHub and PyPI, supporting long-term reuse and community contributions.

Installation and Example

Install the package via pip or GitHub:

```
pip install nsEVDx
```

or

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

```
import nsEVDx as ns
import numpy as np
from scipy.stats import genextreme
import matplotlib.pyplot as plt

np.random.seed(123)
covariate  = np.array(range(100))
config = [1,0,0] # non-stationary location; other parameters stationary
true_parameter = [30, 0.1, 5, -0.3]
Observations = ns.NonStationaryEVD.ns_EVDrvs(genextreme, true_parameter,
                                     covariate,
                                     config, size=100)
plt.plot(Observations)

# Inferring GEV Parameters from the Observations
model = ns.NonStationaryEVD(config, Observations,
                            covariate,dist='GEV')
results = model.MH_Hmc(num_samples=1500, burn_in=1500,
                    initial_params=[30, 0, 5, -0.1],
                    num_chains=4, T = 1)
# Results
print(f"sample mean : {np.round(np.vstack(results['chains']).mean(axis=0), 3)}")
ns.plot_trace(results['chains'], config, fig_size=(8,8),show=False);
ns.plot_posterior(results['chains'], config, fig_size=(8,8),show=False);
```

Detailed installation instructions, tutorials, API documentation, and executable examples are available in the project GitHub repository and full online documentation.

AI Usage Disclosure

AI tools were used in a limited and supportive role to assist in the development of this library and in drafting the manuscript. Specifically, ChatGPT and Claude were used to refine docstrings and documentation, and to suggest code improvements. All AI-generated docstrings and documentation were carefully reviewed by the authors to ensure that they accurately reflected the intended function and design of the software. Any code suggestions from AI tools were independently tested, verified, and revised before incorporation. The original draft of the manuscript was written by the authors and subsequently improved through feedback and revisions from the advisor, reviewers, editors, and ChatGPT. All underlying scientific motivation, software architecture, analyses, and final interpretations were conceived, verified, and approved by the authors.

Acknowledgements

We thank the JOSS Editor and Reviewers for their thorough and constructive feedback, which significantly improved `nsEVDx`. We also acknowledge the `pyOpenSci` Editors and Reviewers for their valuable suggestions, Vincent Gao for identifying and fixing a bug, and other contributors who helped improve the package.

Funding

No external funding was received for this work.

References

Abril-Pla, O., Andreani, V., Carroll, C., Dong, L. Y., Fonnesbeck, C., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., Osthege, M., Vieira, R., Wiecki, T. V., & Zinkov, R. (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

Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv: Methodology*. https://doi.org/10.48550/arXiv.1701.02434

Bocharov, G. (2020). Pyextremes: Extreme Value Analysis (EVA) in Python. In *GitHub repository* (Version 2.5.0). GitHub. https://georgebv.github.io/pyextremes/

Castillo, E. (1988). *Extreme value theory in engineering*. Academic Press. https://doi.org/10.1016/C2009-0-22169-6

Coles, S. (2001). *An introduction to statistical modeling of extreme values* (4th. printing). Springer. https://doi.org/10.1007/978-1-4471-3675-0

Deville, Y. (2026). *NSGEV: Non-stationary GEV time series*. https://github.com/IRSN/NSGEV/

Eccles, R., Syktus, J., Trancoso, R., Chapman, S., Wasko, C., Evans, J. P., Thatcher, M., Di Virgilio, G., & Stassen, C. (2025). Substantial increases in future precipitation extremes—insights from a large ensemble of downscaled CMIP6 models. *Npj Natural Hazards*, *2*(1), 60. https://doi.org/10.1038/s44304-025-00107-1

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). Emcee: The MCMC hammer. *PASP*, *125*, 306–312. https://doi.org/10.1086/670067

Gilleland, E. (2025). *extRemes: Extreme Value Analysis*. https://doi.org/10.32614/CRAN.package.extRemes

Heffernan J. E., Stephenson A.G., & Gilleland E. (2003). *Ismev: An Introduction to Statistical Modeling of Extreme Values*. https://doi.org/10.32614/CRAN.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://doi.org/10.1017/CBO9780511529443

Jayaweera, L., Wasko, C., & Nathan, R. (2025). Evidence for non-stationarity in the GEV shape parameter when modeling extreme rainfall. *Water Resources Research*, *61*(5), e2023WR036426. https://doi.org/10.1029/2023WR036426

Kafle, N. (2026). *Rain-gauge network effects on the uncertainty and trends in short-duration extreme precipitation* (PhD Dissertation 32785789, The University of Memphis). https://ezproxy.memphis.edu:3443/login?url=https://www.proquest.com/dissertations-theses/rain-gauge-network-effects-on-uncertainty-trends/docview/3369343212/se-2

Kafle, N., Dell'Aira, F., Chadwick, C., & Meier, C. I. (2026). Robustness of regionally derived, short-duration rainfall depth-duration-frequency estimates to the choice of minimum interevent time: Evidence across climates, raingauge densities, and regionalization approaches. *Journal of Hydrologic Engineering*. https://doi.org/10.1061/JHYEFF.HEENG-6729

Kafle, N., Peleg, N., & Meier, C. I. (2025). Detecting spatially consistent trends in sub-hourly extreme rainfall using a neighborhood-based method. *AGU Fall Meeting Abstracts*, *2025*, H13G–07. https://ui.adsabs.harvard.edu/abs/2025AGUFMH13G...07K/abstract

Katz, R. W., Parlange, M. B., & Naveau, P. (2002). Statistics of extremes in hydrology. *Advances in Water Resources*, *25*(8–12), 1287–1304. https://doi.org/10.1016/S0309-1708(02)00056-8

McNeil, A. J., & Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: An extreme value approach. *Journal of Empirical Finance*, *7*(3–4), 271–300. https://doi.org/10.1016/S0927-5398(00)00012-8

Neal, R. M. (2011). MCMC using Hamiltonian Dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). CRC Press. https://doi.org/10.1201/b10905

Paciorek, C. (2016). *climextRemes: Tools for Analyzing Climate Extremes*. https://doi.org/10.32614/CRAN.package.climextRemes

Phan, D., Pradhan, N., & Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv Preprint arXiv:1912.11554*. https://doi.org/10.48550/arXiv.1912.11554

Prosdocimi, I., Kjeldsen, T. R., & Miller, J. D. (2015). Detection and attribution of urbanization effect on flood extremes using nonstationary flood-frequency models. *Water Resources Research*, *51*(6), 4244–4262. https://doi.org/10.1002/2015WR017065

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://doi.org/10.5281/zenodo.1117248

Stan Development Team. (2023b). *PyStan: The Python interface to Stan*. https://doi.org/10.5281/zenodo.1456206