Machine Learning & Global Health Network Day

Tutorial: Implementing distributions in STAN

Anna Freni Sterrantino, Zhi Ling, Oliver Ratmann

May 22nd, 2025
Imperial College London

Aims and scope

This short hands-on tutorial introduces participants to implementing custom statistical distributions in Stan, with a focus on the Yule-Simon distribution. Through guided examples, we will demonstrate three progressively technical methods: writing the distribution directly in the Stan language, using external C++ code, and finally, developing it in a deeper "dev-style" C++ integration. The session is designed for researchers and practitioners who are familiar with Stan and want to extend its capabilities with custom models.

What You Will Learn

By the end of this tutorial, you will:

- Understand the structure and syntax for defining custom distributions in Stan.
- Implement the Yule-Simon distribution in:
  - Pure Stan language using user-defined functions.
  - External C++ via the Stan Math library interface.
  - Developer-style C++ for deeper integration and efficiency.
- Gain insight into when and why to use each approach.
- Learn debugging tips and best practices for custom Stan functions.

Expected Prior Knowledge

- Basic understanding of Bayesian modelling and Stan
- Familiarity with R or Python scripting
- Some experience with the command line
- Optional: basic C++ familiarity

For the tutorial

To run the tutorial smoothly, make sure you have the following installed and working.

- RStudio
- R packages: `tidyr, rstan, loo, ggplot2, gridExtra, bayesplot, cmdstanr,posterior`
- Optional: emacs

Pre-tutorial Checklist

1. Install Required Software

Option A: R + CmdStanR

- Install **R** (version 4.3 or higher): `https://cran.r-project.org/`
- Install **RStudio**: `https://posit.co/download/rstudio-desktop/`
- In R, install CmdStanR and CmdStan:

```
install.packages("cmdstanr", repos = c("https://mc-stan.org/r-packages/", getOptio
cmdstanr::install_cmdstan()
```

Option B: Python + CmdStanPy

- Install **Python** (version 3.8 or higher): `https://www.python.org/`
- Recommended: Install via **Anaconda**: `https://www.anaconda.com/`
- In terminal or Anaconda Prompt, install CmdStanPy and CmdStan:

```
pip install cmdstanpy
python -c "import␣cmdstanpy;␣cmdstanpy.install_cmdstan()"
```

2. Install a C++ Toolchain

- **macOS**: `xcode-select --install`
- **Windows (R)**: Install RTools: `https://cran.r-project.org/bin/windows/Rtools/`
- **Windows (Python)**: Use WSL with GCC or MSVC toolchain
- **Linux**: `sudo apt install build-essential`

3. Install a Code Editor

Have one of the following installed:

- Visual Studio Code
- RStudio
- Sublime Text
- Emacs

4. Verification Scripts

R Version

```
library(cmdstanr)
check_cmdstan_toolchain()
cmdstanr::cmdstan_version()
##  set_cmdstanr_path() add your path

mod <- cmdstan_model(write_stan_file("data{
real<lower=0>alpha;
}
parameters{
real<lower=0>y;
}
model{
y~exponential(alpha);
}"))
fit <- mod$sample(data = list(alpha = 1), chains = 1, iter_sampling = 100)
print(fit$summary())
```

Python Version

```
from cmdstanpy import CmdStanModel, install_cmdstan
install_cmdstan()

stan_code = """
data {
  real<lower=0> alpha;
}
parameters {
  real<lower=0> y;
}
model {
  y ~ exponential(alpha);
}
"""
with open("simple_model.stan", "w") as f:
    f.write(stan_code)

model = CmdStanModel(stan_file="simple_model.stan")
fit = model.sample(data={"alpha": 1}, chains=1, iter_sampling=100)
print(fit.summary())
```

References

How to install https://mc-stan.org/docs/cmdstan-guide/installation.html#cpp-toolchain
Yule-Simon Distribution https://www.statisticshowto.com/yule-simon-distribution/
Distributions installed in Stan https://github.com/stan-dev/math/tree/develop/stan/math/prim/prob
Cmdstanr vignette https://mc-stan.org/cmdstanr/articles/cmdstanr.html
Adding new functions to Stan https://mc-stan.org/math/md_doxygen_2contributor__help__pages_2getting__started.html
Adding distribution in Stan Mathhttps://mc-stan.org/math/md_doxygen_2contributor__help__pages_2adding__new__distributions.html
Define Custom Response Distributions with brms https://cran.r-project.org/web/packages/brms/vignettes/brms_customfamilies.html
Full stan example for lpmf, cdf etc https://jepusto.com/posts/double-poisson-in-Stan/
Custom likelihoods with Stan example
https://aheblog.com/2018/05/18/method-of-the-month-custom-likelihoods-and-bayesian-models-in-
Yule Simon distribution https://par.nsf.gov/servlets/purl/10334078