Fundamentals of Stan

StanCon 2023

mc-stan.org

Instructor: Charles Margossian
cmargossian@flatironinstitute.org
https://charlesm93.github.io./

Teaching Assistant: Manny Mokel
emmanuel.mokel@duke.edu

Course description

This half-a-day course serves as a rigorous introduction to Stan and can be used as a stepping stone to then take more advanced tutorials.

We will discuss the tenants of the *Bayesian workflow*, i.e. the iterative process through which we build, fit, and criticize models, with the latter step often motivating useful revisions to our model. We will see how Stan supports each of the steps and apply them to several illustrative examples. In an effort to thoroughly understand the outputs returned by Stan, we will also delve a little bit under-the-hood, and review Stan's Hamiltonian Monte Carlo sampler—currently one of the most successful Markov chain Monte Carlo (MCMC) methods—and cover essential diagnostics for running MCMC. This will be a hands-on workshop: participants are expected to code and attempt several exercises.

Goals:

- Learn the Stan language
- Understand how to do Bayesian inference with MCMC
- Learn the fundamentals of the Bayesian workflow
- Apply these principles to some illustrative examples

Helpful pre-requisites:

- Familiarity with probability (i.e. different types of distributions; conditional distributions; definitions of expectation values, variance, quantiles, etc.)
- Familiarity in a coding language, e.g. R or Python. The focus will be on coding in Stan. Participants are not expected to know Stan.

About the instructor: *Charles is a Research Fellow at the Flatiron Institute, Center for Computational Mathematics. He holds a PhD in Statistics from Columbia University. For 7 years, he has been a member of the Stan development team. His contributions include support for several implicit functions (e.g. matrix exponentials, algebraic solvers, hidden Markov models), and the co-creation of Torsten, an extension of Stan for pharmacometrics. His research interests include Bayesian hierarchical models, Markov chain Monte Carlo, variational inference and automatic differentiation.*

About the teaching assistant: *Manny is a rising senior at Duke University studying mathematics and statistics, as well as a Summer Research Assistant at the Flatiron Institute, Center for Computational Mathematics. He is primarily interested in Bayesian statistical modeling for problems in the natural and physical sciences.*

Course Outline

- Review of Bayesian analysis
  - Bayesian model
  - Bayesian inference
  - Beyond inference: the Bayesian workflow
- Markov chain Monte Carlo
  - Probing the posterior distribution
  - Controlling the error of Monte Carlo estimators
  - Hamiltonian Monte Carlo
- Stan
  - Basics of Stan
  - Example 1: Bayesian linear regression
  - Checking the inference ($\widehat{R}$, ESS, trace and density plots)
  - Checking the model (posterior predictive checks)
  - Exercise: improve the Bayesian linear regression model

- Epidemiology model
  - Susceptible-Infected-Recovered (SIR) model
  - Solving an ODE in Stan
  - Exercise: build, fit, and criticize an SIR model.

Installation Instructions

For this tutorial, we will use `R` as a scripting language. The code in `R` is already written and we will write `Stan` code together. The `R` code can be found at

`https://github.com/charlesm93/stanTutorial`

From the GitHub page you can open `Stan_tutorial.ipynb` and then click on the button "Open in Colab". If you do so, you will be able to run the `R` script on Colab's cloud server. The first cells of the notebook install all the requisite `R` packages and `Stan`.

You also have the option to run `R` on your local machine, in which case you must install the following packages:

- `cmstanr`, `ggplot2`, `rjson`, `posterior`, `bayesplot`.

If you're not able to get something installed, don't hesitate to email the instructor and teaching assistant, or post an inquiry on the Stan forum (`https://discourse.mc-stan.org/`). If you're not able to install the packages before the tutorial, that's ok! We'll help you the day of, and you still have the option to use Colab.

- For instructions on installing `R`, see `https://www.r-project.org/`. I also recommend downloading `RStudio` (`https://posit.co/products/open-source/rstudio/`) — the free version is all I've ever used.

- `cmdstanr` is a light-weight wrapper around `Stan` and the most up-to-date `R` interface for `Stan`.[1] For instructions on how to install `cmdstanr`, see `https://mc-stan.org/cmdstanr/`.

- All other `R` packages are available on CRAN and can be installed directly from `R` using the install function,
  e.g. `install.packages("ggplot2")`.

[1] A similar package exists in `Python` called `cmdstanPy`.