**Introduction to Stan**

Aim: Ensure that Stan will run correctly on a university desktop or on your own machine.

Stan is a programming language that we will use to fit Bayesian models. It uses Monte Carlo methods that we will see during the lectures; however, it is not necessary to understand these methods to begin fitting models. If you’re interested, there are more details here: <https://mc-stan.org>

We need to install several pieces of software before we can use Stan. If you’re on a university desktop, you’ll already have access to RStudio and Notepad++. If you’re using your own computer, a Google search for `install R’, ‘install RStudio’ and ‘install Notepad++’ will find the correct web pages for downloading these pieces of software.

We need to install Rtools from the ‘Rtools42 installer’ link: [Rtools42 for Windows (r-project.org)](https://cran.r-project.org/bin/windows/Rtools/rtools42/rtools.html)

Then we need to configure C++ from within RStudio using this R code.

install.packages("StanHeaders", repos = c("https://mc-stan.org/r-packages/", getOption("repos")))

A key feature of Stan is that it can be run from RStudio. To be able to do this, we need to install the ‘rstan’ package using the R code below. This will also install packages that Stan depends on.

install.packages("rstan", repos = c("https://mc-stan.org/r-packages/", getOption("repos")),dependencies=TRUE)

I’ve used my Desktop as the working directory for this example. You can change the working directory either using code such as that below, or through the menu system in RStudio.

setwd("C:/Users/James/Desktop")

We then need to create a .stan file that contains the Stan code for fitting the model. This Stan code will be unfamiliar to you at the moment, but for now just copy the code below into a Notepad++ file and save it as ‘schools.stan’ in your working directory. Note that you need to add a blank line at the bottom of the file.

data {

int<lower=0> J;

real y[J];

real<lower=0> sigma[J];

}

parameters {

real mu;

real<lower=0> tau;

vector[J] eta;

}

transformed parameters {

vector[J] theta;

theta = mu + tau \* eta;

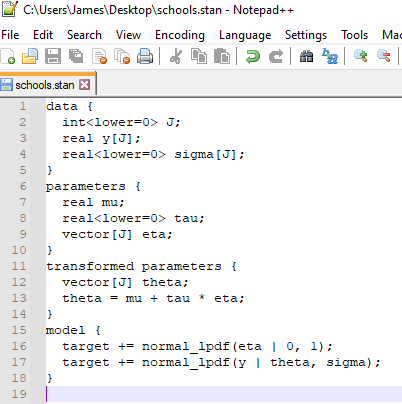
}

model {

target += normal\_lpdf(eta | 0, 1);

target += normal\_lpdf(y | theta, sigma);

}



We can now fit this model and analyse the results entirely within RStudio.

Firstly, we need to load the data into an R object.

schools\_data <- list(

J = 8,

y = c(28, 8, -3, 7, -1, 1, 18, 12),

sigma = c(15, 10, 16, 11, 9, 11, 10, 18)

)

Then we fit the model. The meaning of the options will become familiar to you as the module progresses, but for now you should just run this code.

library(rstan)

fit1 <- stan(

file = "schools.stan", # Stan program

data = schools\_data, # named list of data

chains = 4, # number of Markov chains

warmup = 1000, # number of warmup iterations per chain

iter = 2000, # total number of iterations per chain

cores = 1, # number of cores (could use one per chain)

refresh = 0 # no progress shown

)

plot(fit1)

You should see a plot that looks like this.

Chart

Description automatically generated

If not, something has gone wrong, so go back and check that each step has been carried out correctly.