Stan - an introduction without the scary parts

Sam Abbott
@seabbs
samabbott.co.uk

Aim

High level overview + the tools for learning more

Don’t panic

Hitchhiker's Guide to the Galaxy

Overview

1. Who even are you and why are you here?
2. What is stan anyway?
3. How does stan fit models?
4. What kind of models can stan fit?
5. How do I stan?
6. What is it good for?
7. What is it not good for?
8. How did you find learning stan Sam?
9. Summary

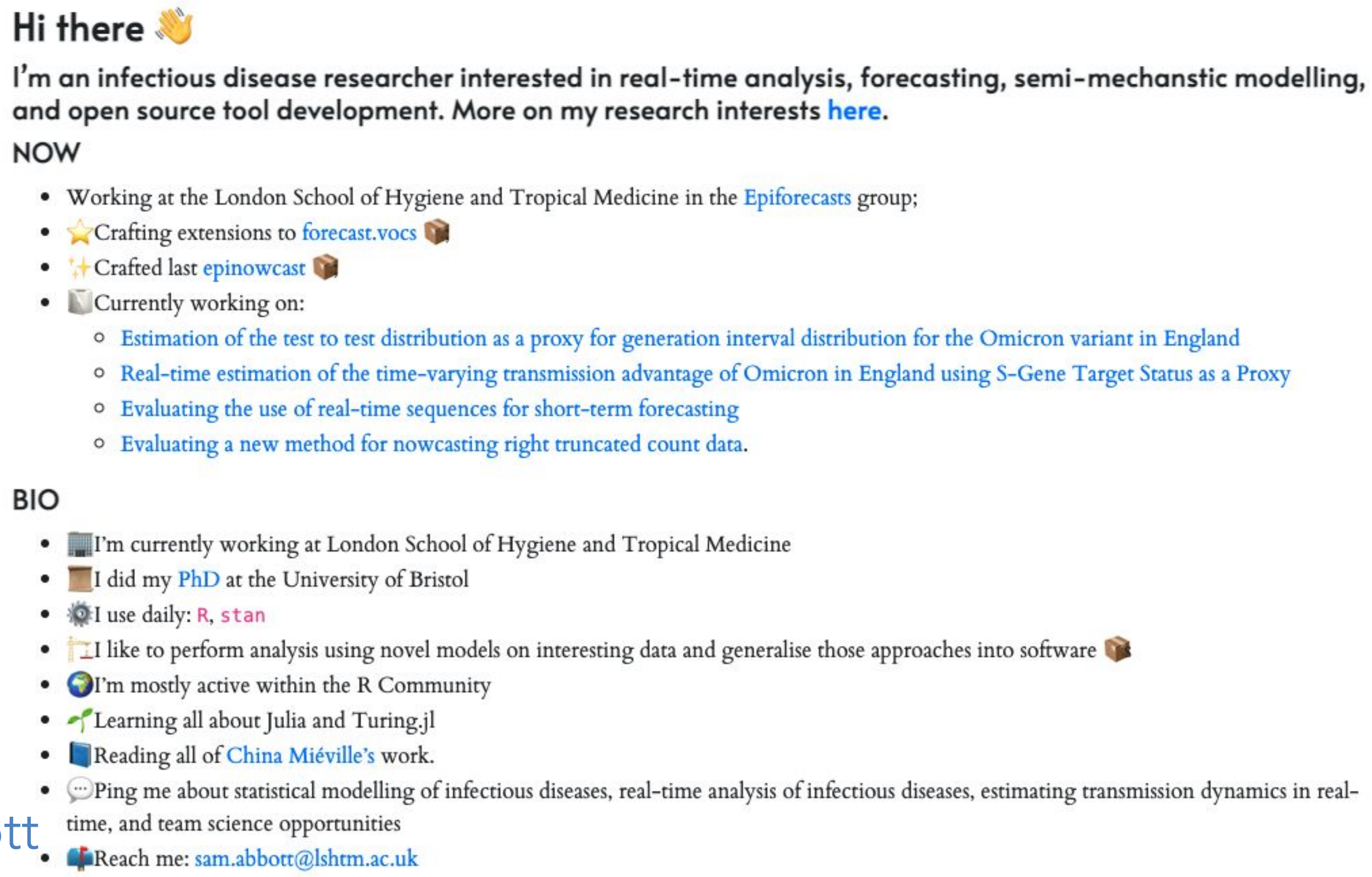
Stan - an introduction without the scary parts

Who even are you and why are you here?

Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE &TROPICAL MEDICINE



Hi there 👋

I'm an infectious disease researcher interested in real-time analysis, forecasting, semi-mechanstic modelling, and open source tool development. More on my research interests here.

NOW

- Working at the London School of Hygiene and Tropical Medicine in the Epiforecasts group;
- ⭐Crafting extensions to forecast.vocs 📦
- ✨Crafted last epinowcast 📦
- 📜Currently working on:
  - Estimation of the test to test distribution as a proxy for generation interval distribution for the Omicron variant in England
  - Real-time estimation of the time-varying transmission advantage of Omicron in England using S-Gene Target Status as a Proxy
  - Evaluating the use of real-time sequences for short-term forecasting
  - Evaluating a new method for nowcasting right truncated count data.

BIO

- 🏢I'm currently working at London School of Hygiene and Tropical Medicine
- 📜I did my PhD at the University of Bristol
- ⚙️I use daily: `R`, `stan`
- 🏗️I like to perform analysis using novel models on interesting data and generalise those approaches into software 📦
- 🌍I'm mostly active within the R Community
- 🌱Learning all about Julia and Turing.jl
- 📘Reading all of China Miéville's work.
- 💬Ping me about statistical modelling of infectious diseases, real-time analysis of infectious diseases, estimating transmission dynamics in real-time, and team science opportunities
- 📫Reach me: sam.abbott@lshtm.ac.uk

Sam Abbott
@seabbs
samabbott.co.uk

I'm a PostDoc, I work for Seb, I think I'm great, and I'm really generally confused

Stan - an introduction without the scary parts

What is stan anyway?

Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE & TROPICAL MEDICINE

What

Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

Users specify log density functions in Stan's probabilistic programming language and get:

- full Bayesian statistical inference with MCMC sampling (NUTS, HMC)
- approximate Bayesian inference with variational inference (ADVI)
- penalized maximum likelihood estimation with optimization (L-BFGS)

Stan's math library provides differentiable probability functions & linear algebra (C++ autodiff). Additional R packages provide expression-based linear modeling, posterior visualization, and leave-one-out cross-validation.

Stan - an introduction without the scary parts

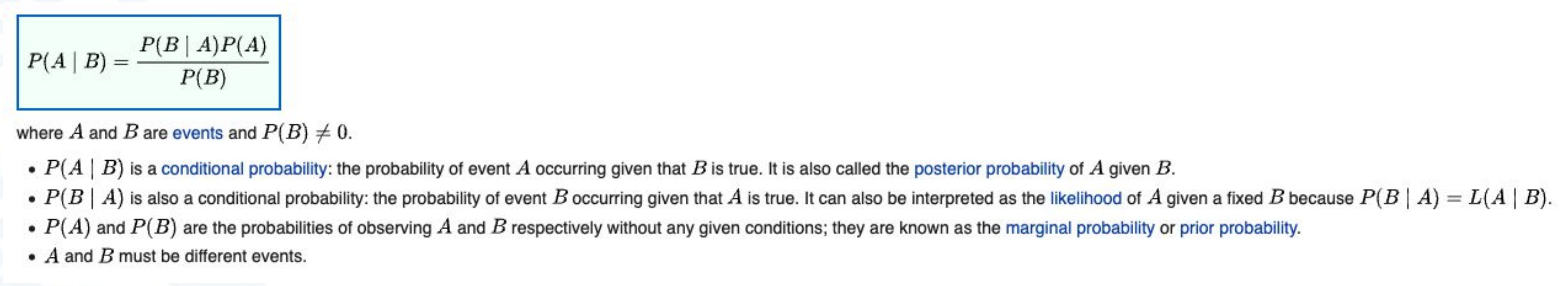
How does stan fit models?

Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE &TROPICAL MEDICINE

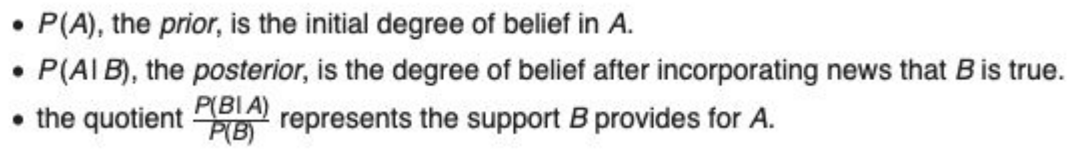
What - Bayes



$$\boxed{P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}}$$

where A and B are events and $P(B) \neq 0$.

- $P(A \mid B)$ is a conditional probability: the probability of event A occurring given that B is true. It is also called the posterior probability of A given B.
- $P(B \mid A)$ is also a conditional probability: the probability of event B occurring given that A is true. It can also be interpreted as the likelihood of A given a fixed B because $P(B \mid A) = L(A \mid B)$.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B respectively without any given conditions; they are known as the marginal probability or prior probability.
- A and B must be different events.



- $P(A)$, the *prior*, is the initial degree of belief in A.
- $P(A|B)$, the *posterior*, is the degree of belief after incorporating news that B is true.
- the quotient $\frac{P(B|A)}{P(B)}$ represents the support B provides for A.

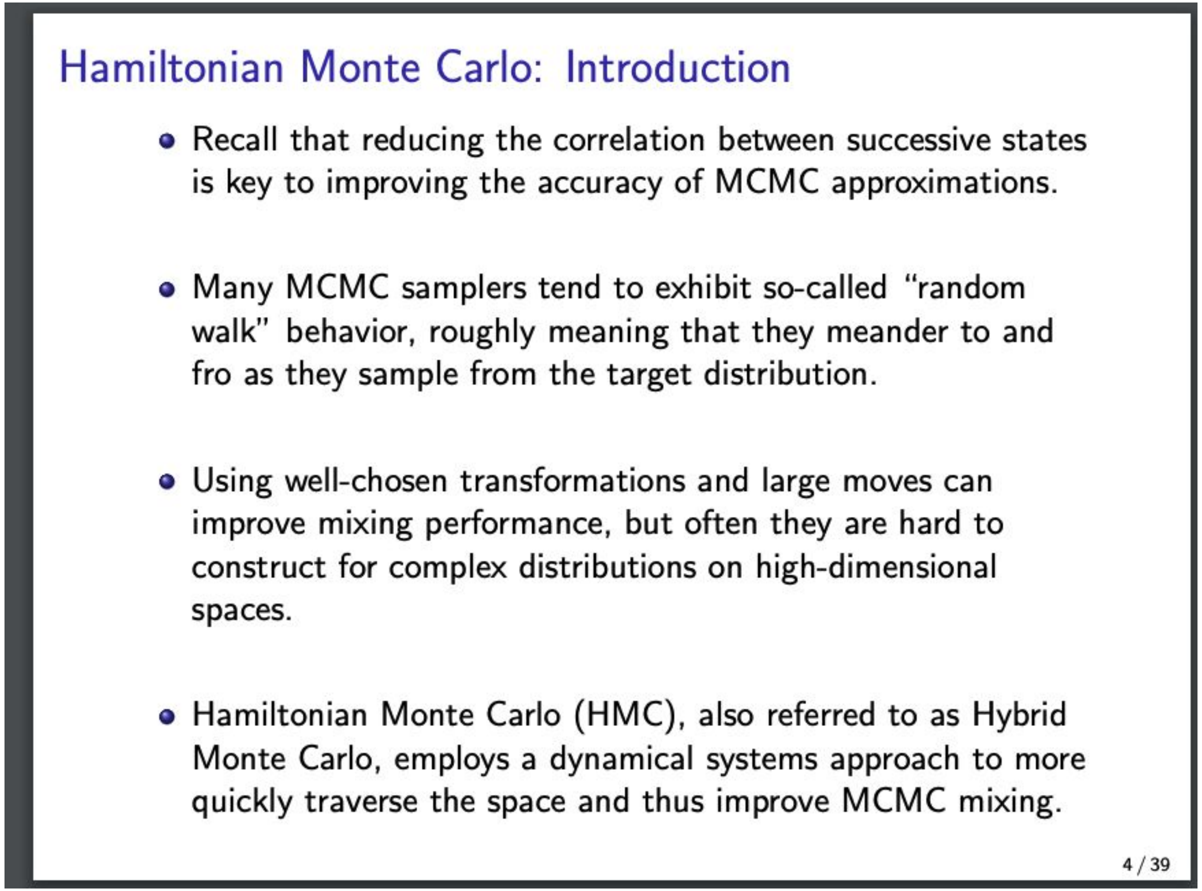
https://en.wikipedia.org/wiki/Bayes%27_theorem

What - MCMC

Markov chain Monte Carlo (MCMC) methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain.

https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo

What - HMC



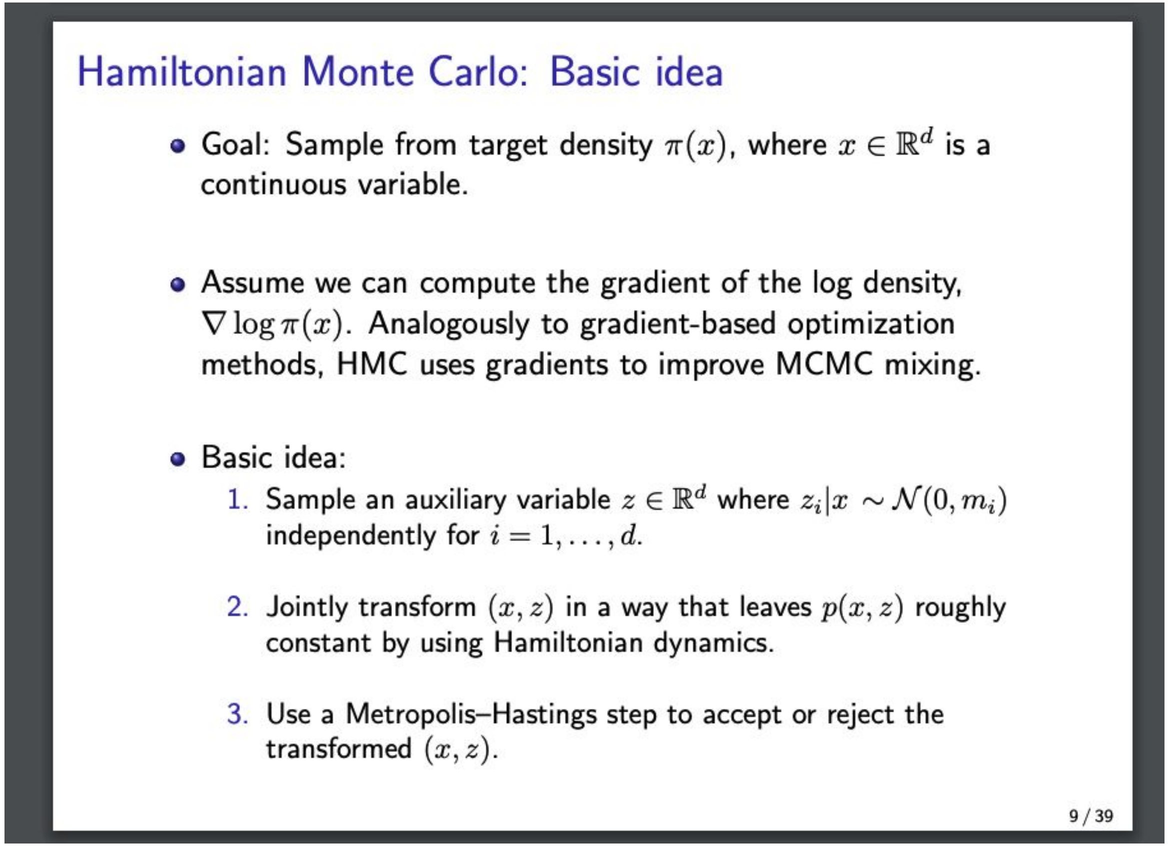
Hamiltonian Monte Carlo: Introduction

- Recall that reducing the correlation between successive states is key to improving the accuracy of MCMC approximations.
- Many MCMC samplers tend to exhibit so-called "random walk" behavior, roughly meaning that they meander to and fro as they sample from the target distribution.
- Using well-chosen transformations and large moves can improve mixing performance, but often they are hard to construct for complex distributions on high-dimensional spaces.
- Hamiltonian Monte Carlo (HMC), also referred to as Hybrid Monte Carlo, employs a dynamical systems approach to more quickly traverse the space and thus improve MCMC mixing.

4 / 39

https:/ /jwmi.github.io/BMB/18-Hamiltonian-Monte-Carlo-and-NUTS.pdf

What - HMC



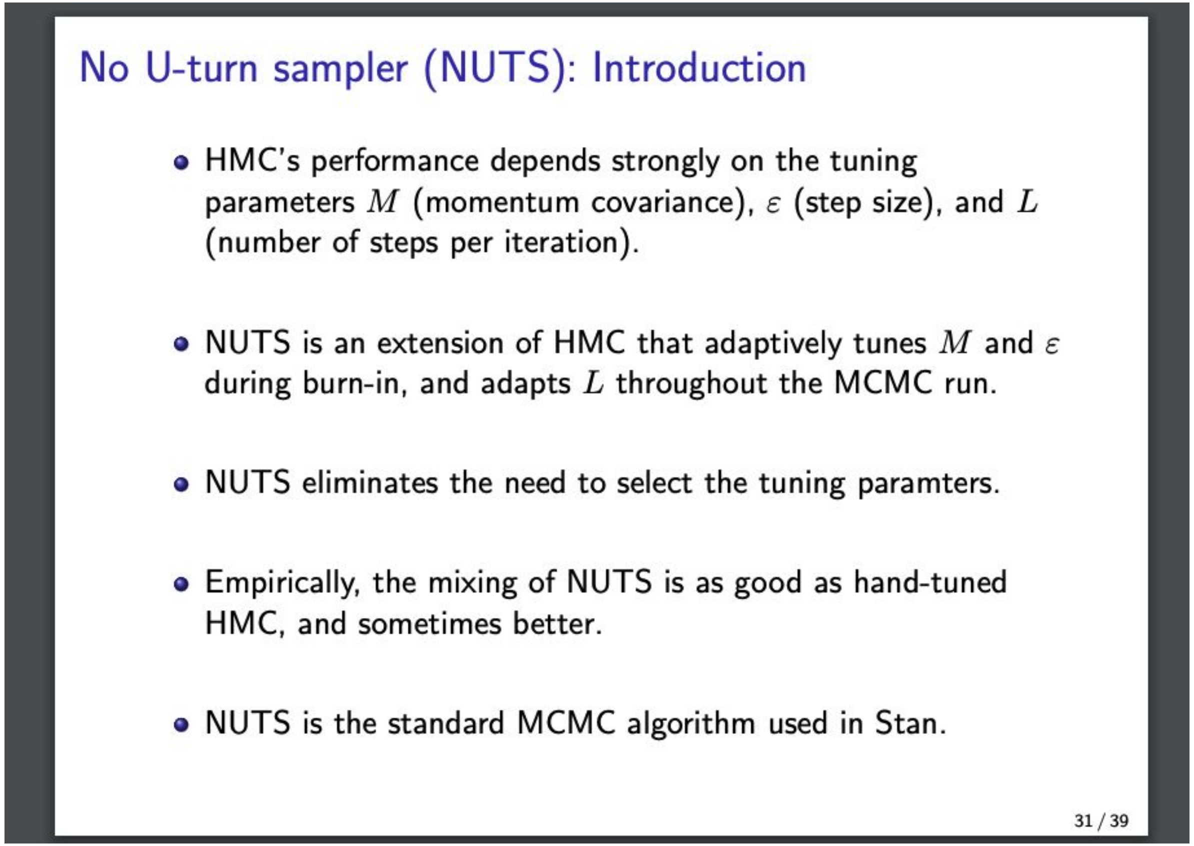
Hamiltonian Monte Carlo: Basic idea

- Goal: Sample from target density $\pi(x)$, where $x \in \mathbb{R}^d$ is a continuous variable.
- Assume we can compute the gradient of the log density, $\nabla \log \pi(x)$. Analogously to gradient-based optimization methods, HMC uses gradients to improve MCMC mixing.
- Basic idea:
  1. Sample an auxiliary variable $z \in \mathbb{R}^d$ where $z_i | x \sim \mathcal{N}(0, m_i)$ independently for $i = 1, \ldots, d$.
  2. Jointly transform (x, z) in a way that leaves $p(x, z)$ roughly constant by using Hamiltonian dynamics.
  3. Use a Metropolis–Hastings step to accept or reject the transformed (x, z).

9 / 39

https:/ /jwmi.github.io/BMB/18-Hamiltonian-Monte-Carlo-and-NUTS.pdf

What - NUTs

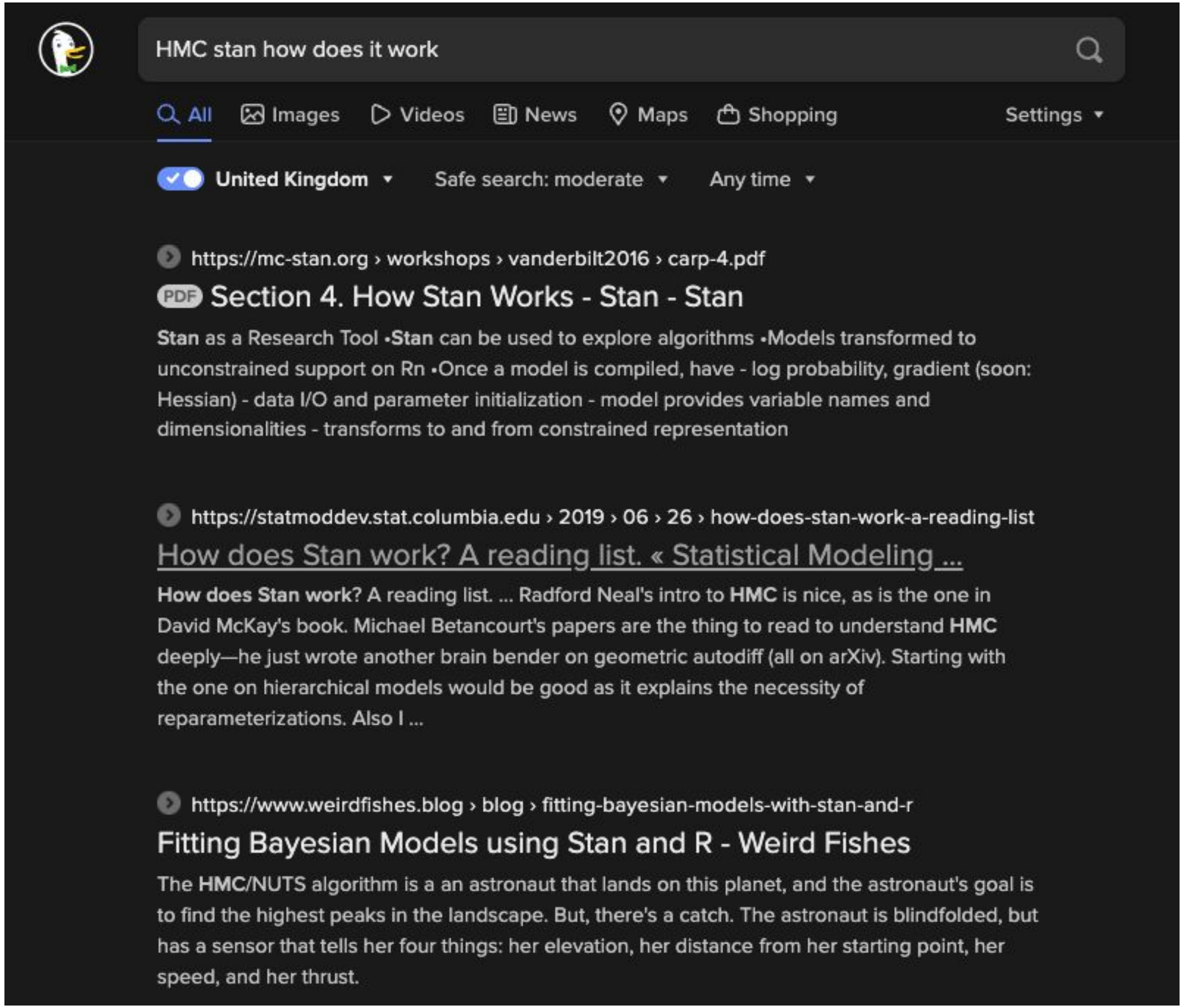


No U-turn sampler (NUTS): Introduction

- HMC's performance depends strongly on the tuning parameters M (momentum covariance), ε (step size), and L (number of steps per iteration).
- NUTS is an extension of HMC that adaptively tunes M and ε during burn-in, and adapts L throughout the MCMC run.
- NUTS eliminates the need to select the tuning paramters.
- Empirically, the mixing of NUTS is as good as hand-tuned HMC, and sometimes better.
- NUTS is the standard MCMC algorithm used in Stan.

31 / 39

https:/ /jwmi.github.io/BMB/18-Hamiltonian-Monte-Carlo-and-NUTS.pdf


HMC stan how does it work

United Kingdom Safe search: moderate Any time

https://mc-stan.org › workshops › vanderbilt2016 › carp-4.pdf

PDF Section 4. How Stan Works - Stan - Stan

Stan as a Research Tool •**Stan** can be used to explore algorithms •Models transformed to unconstrained support on Rn •Once a model is compiled, have - log probability, gradient (soon: Hessian) - data I/O and parameter initialization - model provides variable names and dimensionalities - transforms to and from constrained representation

https://statmoddev.stat.columbia.edu › 2019 › 06 › 26 › how-does-stan-work-a-reading-list

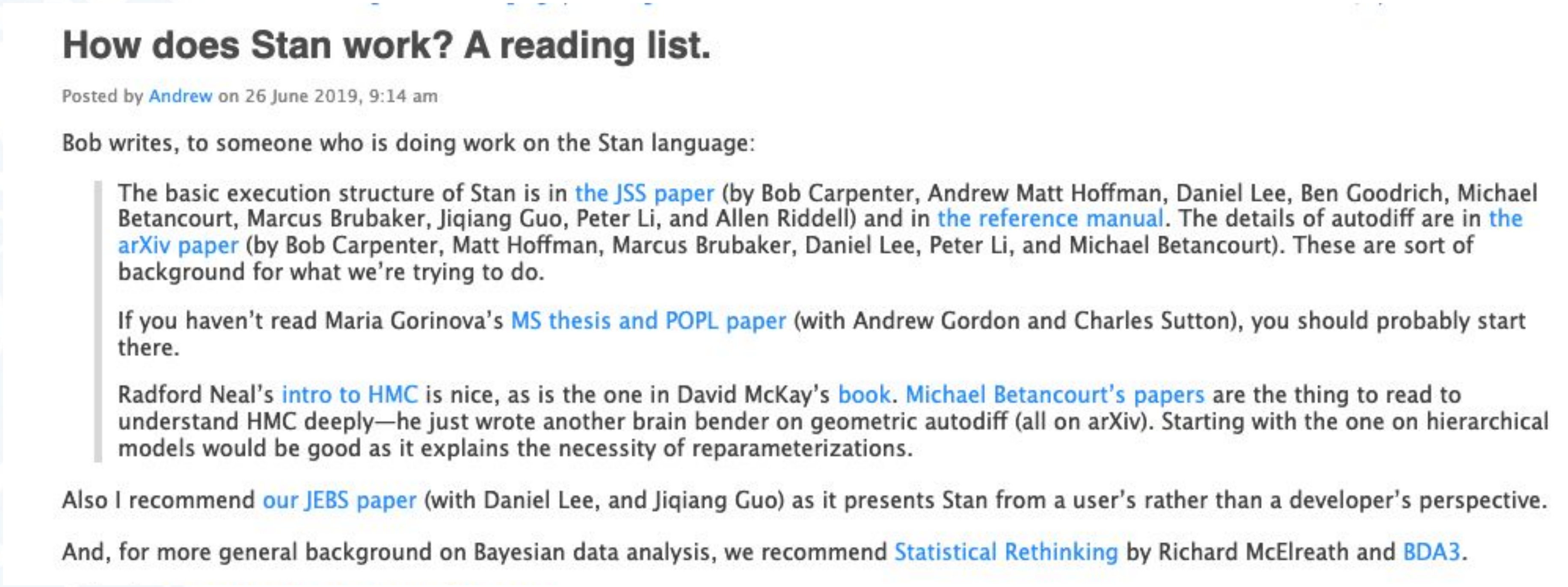
How does Stan work? A reading list. « Statistical Modeling ...

How does Stan work? A reading list. ... Radford Neal's intro to **HMC** is nice, as is the one in David McKay's book. Michael Betancourt's papers are the thing to read to understand **HMC** deeply—he just wrote another brain bender on geometric autodiff (all on arXiv). Starting with the one on hierarchical models would be good as it explains the necessity of reparameterizations. Also I ...

https://www.weirdfishes.blog › blog › fitting-bayesian-models-with-stan-and-r

Fitting Bayesian Models using Stan and R - Weird Fishes

The **HMC**/NUTS algorithm is a an astronaut that lands on this planet, and the astronaut's goal is to find the highest peaks in the landscape. But, there's a catch. The astronaut is blindfolded, but has a sensor that tells her four things: her elevation, her distance from her starting point, her speed, and her thrust.



How does Stan work? A reading list.

Posted by Andrew on 26 June 2019, 9:14 am

Bob writes, to someone who is doing work on the Stan language:

> The basic execution structure of Stan is in the JSS paper (by Bob Carpenter, Andrew Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell) and in the reference manual. The details of autodiff are in the arXiv paper (by Bob Carpenter, Matt Hoffman, Marcus Brubaker, Daniel Lee, Peter Li, and Michael Betancourt). These are sort of background for what we're trying to do.
>
> If you haven't read Maria Gorinova's MS thesis and POPL paper (with Andrew Gordon and Charles Sutton), you should probably start there.
>
> Radford Neal's intro to HMC is nice, as is the one in David McKay's book. Michael Betancourt's papers are the thing to read to understand HMC deeply—he just wrote another brain bender on geometric autodiff (all on arXiv). Starting with the one on hierarchical models would be good as it explains the necessity of reparameterizations.

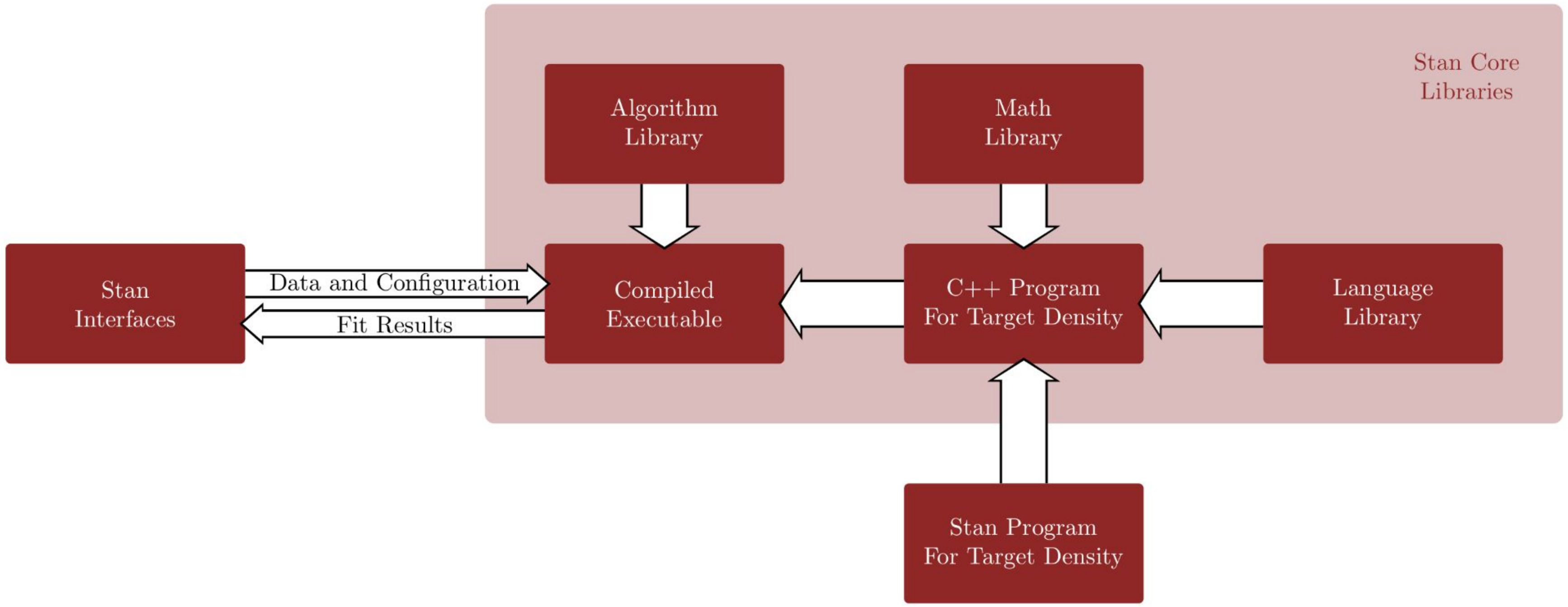
Also I recommend our JEBS paper (with Daniel Lee, and Jiqiang Guo) as it presents Stan from a user's rather than a developer's perspective.

And, for more general background on Bayesian data analysis, we recommend Statistical Rethinking by Richard McElreath and BDA3.

Stan - an introduction without the scary parts

What is stan anyway?

Sam Abbott
@seabbs
samabbott.co.uk



Michael Betancourt betanalpha.github.io/assets/case_studies/stan_intro.html

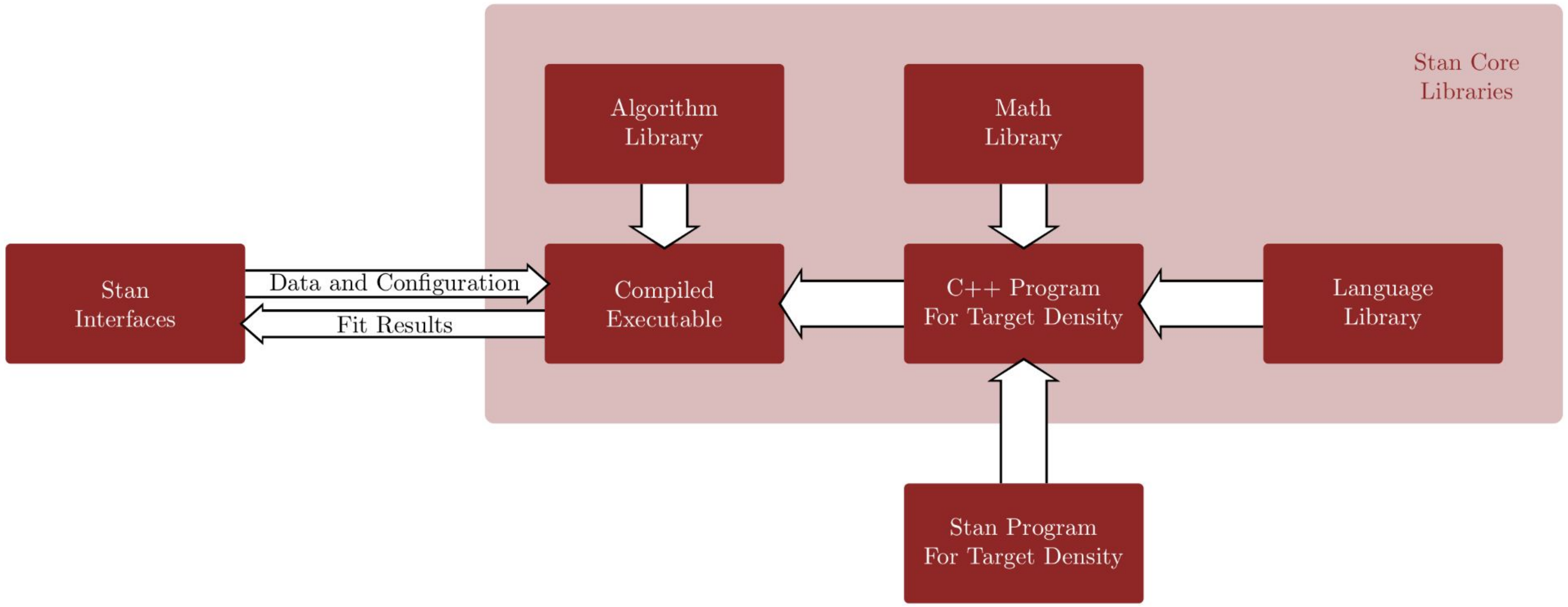
Stan Language Library

- The probabilistic programming that we use to specify our models
- Transpiled into C++
- Think of it as the child of R and C++

Imperative, strongly and statically typed, domain specific probabilistic programming language that defines a log probability density function through programming blocks.

Designed to complement these algorithms [NUTs-HMC] by specifying not just any density function but differentiable density functions defined over continuous product spaces.

Michael Betancourt betanalpha.github.io/assets/case_studies/stan_intro.html



Michael Betancourt betanalpha.github.io/assets/case_studies/stan_intro.html

Stan Math Library

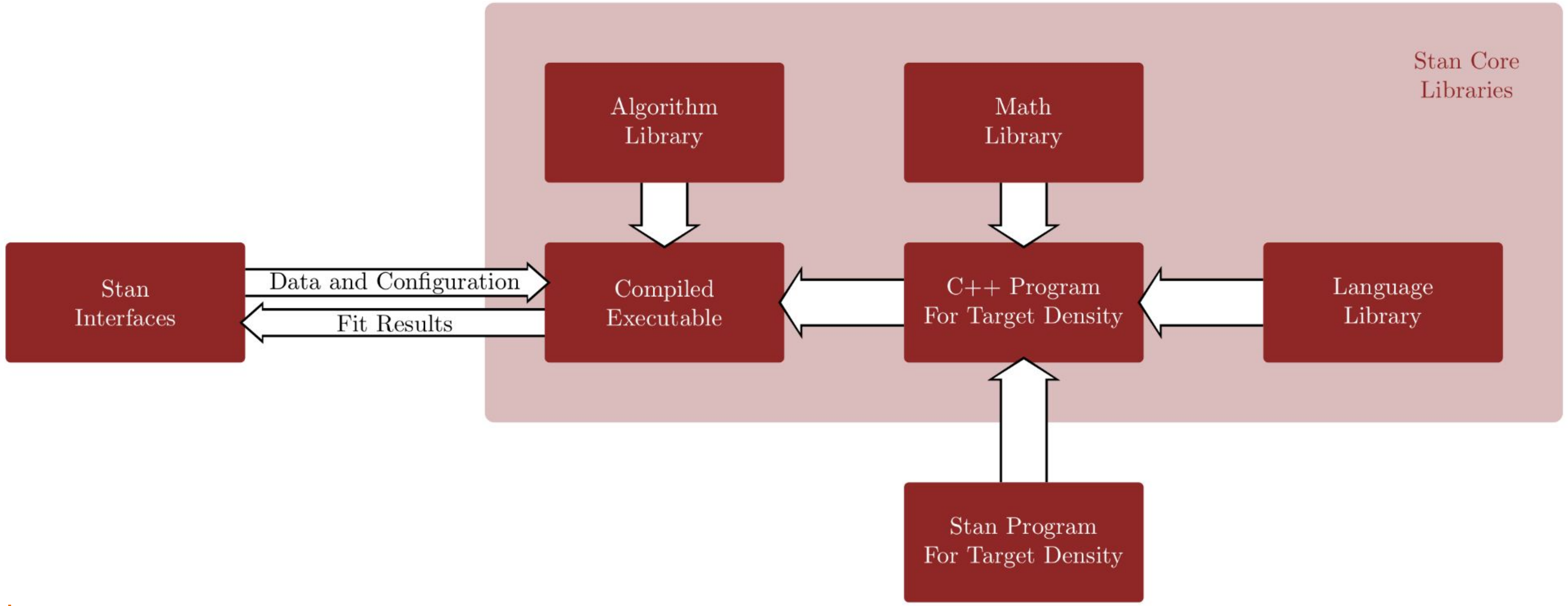
Critically the Stan Math Library also implements automatic differentiation for all of its functions.

Automatic differentiation is a technique for efficiently evaluating the exact values of the gradient of a C++ function at a given set of inputs.

This means that every Stan program defines both a target log probability density function and the corresponding gradient function without any additional effort from the user.

This then allows the use of extremely effective gradient-based algorithms like Hamiltonian Monte Carlo without the user having to pour through pages upon pages of analytic derivative calculations.

Michael Betancourt betanalpha.github.io/assets/case_studies/stan_intro.html



rstan

cmdstanr

Michael Betancourt betanalpha.github.io/assets/case_studies/stan_intro.html

Stan - an introduction without the scary parts

What kind of models can stan fit?

Sam Abbott
@seabbs
samabbott.co.uk

No discrete parameters (unless they can be marginalised away)

Otherwise it is gravy

Integrated ODE solvers

Model complexity limited by computational costs of MCMC

Stan - an introduction without the scary parts

How do I stan?

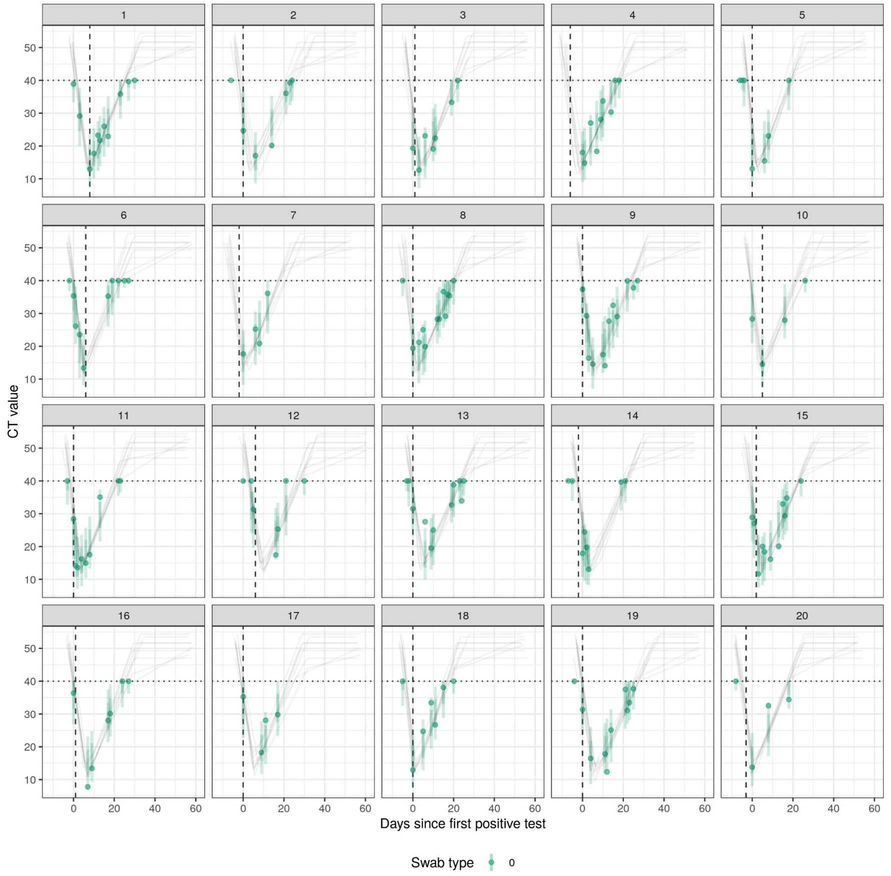
Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE &TROPICAL MEDICINE

My first model - Introduction

- We want to model viral load for COVID-19
- For some number of people we have some number of PCR test results and the date these were done on.
- For these tests we have the Cycle Thresholds (CTs) which we use as a viral load proxy
- We want to model population viral load from infection and understand if it differs by variant etc.



with Timothy Russell et al.

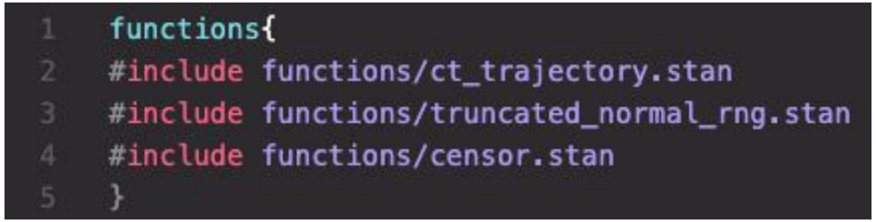
My first model - Functions

- We can use functions in stan as in other languages

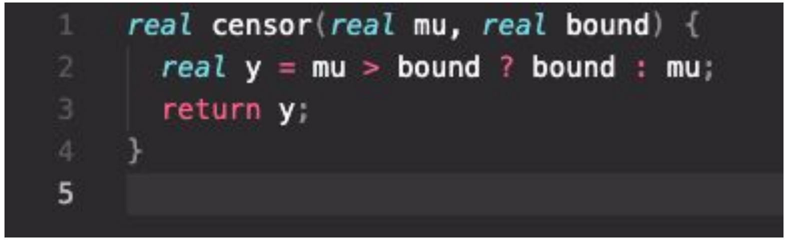
- Like C/C++ we need to define the input and return types.

- We can compile these functions and use them like native R functions for fun and testing if we want to.

- We can't use R functions but we can integrate C++ code if we are very fancy.



```
functions{
#include functions/ct_trajectory.stan
#include functions/truncated_normal_rng.stan
#include functions/censor.stan
}
```

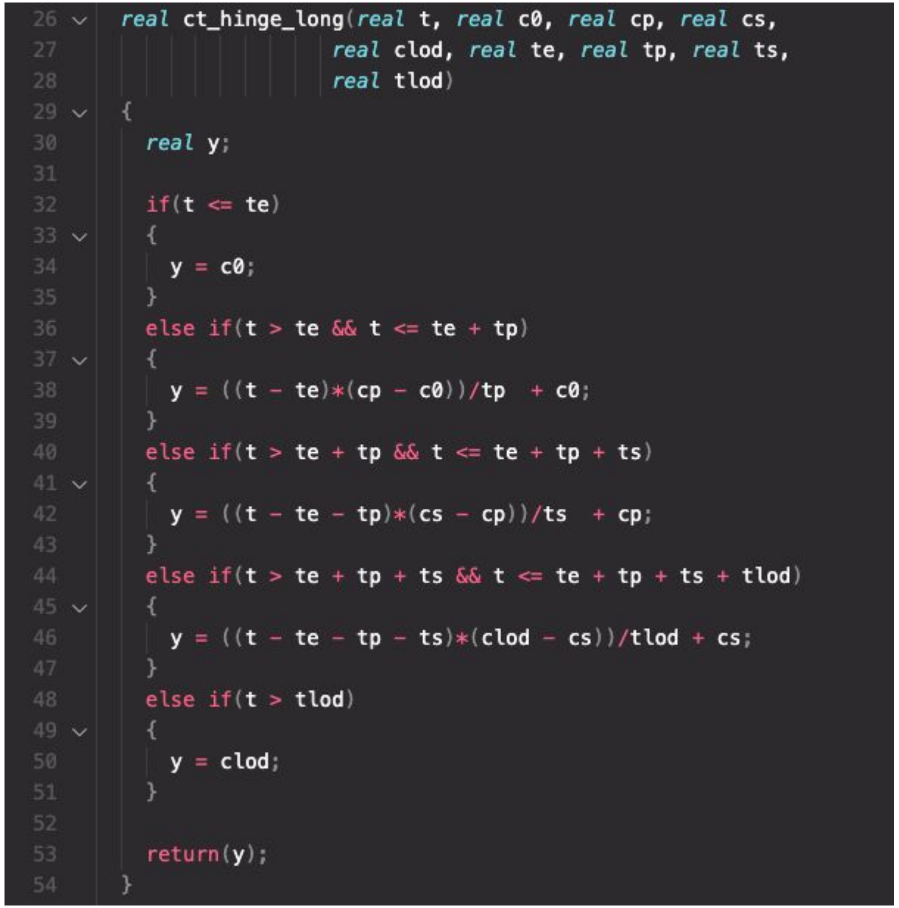


```
real censor(real mu, real bound) {
  real y = mu > bound ? bound : mu;
  return y;
}

```

My first model - Functions

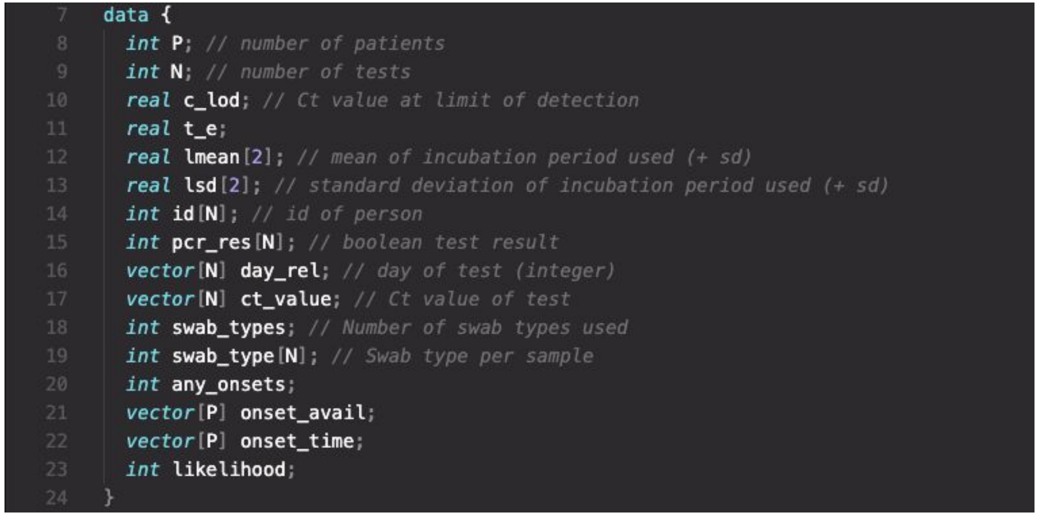
Piecewise linear with 2 change points

- Really complicated and scientific so lets make some assumptions.
- Viral load changes linearly on the log scale.
- Initially load goes up
- Hits a turning point at which time clearance begins
- Later on clearance rate may change or it may not.
- Everything is on the CT scale which is inverted and has a minimum at 0.



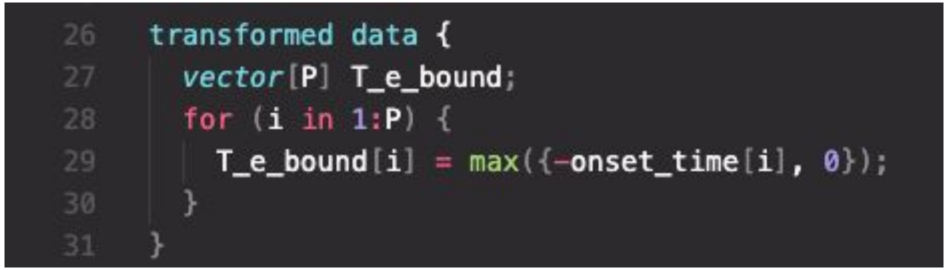
```
real ct_hinge_long(real t, real c0, real cp, real cs,
                   real clod, real te, real tp, real ts,
                   real tlod)
{
  real y;

  if(t <= te)
  {
    y = c0;
  }
  else if(t > te && t <= te + tp)
  {
    y = ((t - te)*(cp - c0))/tp  + c0;
  }
  else if(t > te + tp && t <= te + tp + ts)
  {
    y = ((t - te - tp)*(cs - cp))/ts  + cp;
  }
  else if(t > te + tp + ts && t <= te + tp + ts + tlod)
  {
    y = ((t - te - tp - ts)*(clod - cs))/tlod + cs;
  }
  else if(t > tlod)
  {
    y = clod;
  }

  return(y);
}
```

My first model - Data



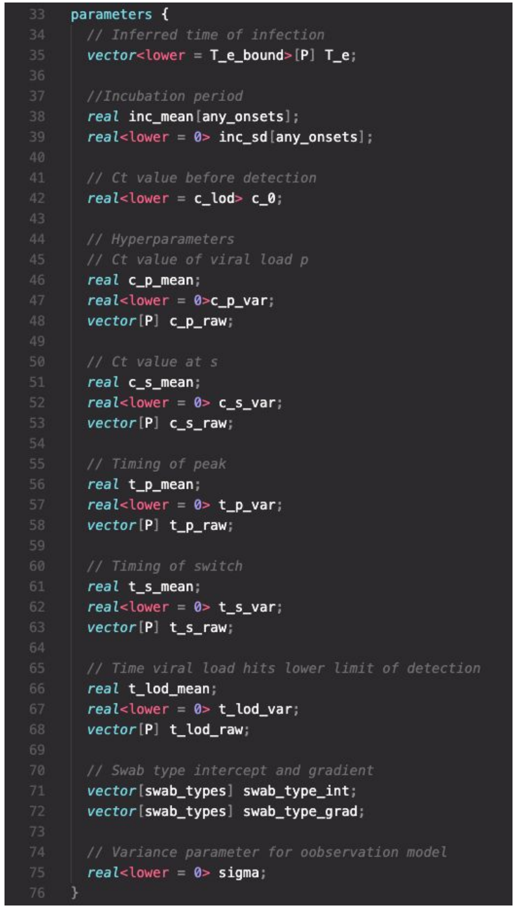
```
data {
  int P; // number of patients
  int N; // number of tests
  real c_lod; // Ct value at limit of detection
  real t_e;
  real lmean[2]; // mean of incubation period used (+ sd)
  real lsd[2]; // standard deviation of incubation period used (+ sd)
  int id[N]; // id of person
  int pcr_res[N]; // boolean test result
  vector[N] day_rel; // day of test (integer)
  vector[N] ct_value; // Ct value of test
  int swab_types; // Number of swab types used
  int swab_type[N]; // Swab type per sample
  int any_onsets;
  vector[P] onset_avail;
  vector[P] onset_time;
  int likelihood;
}
```

My first model - Transformed data



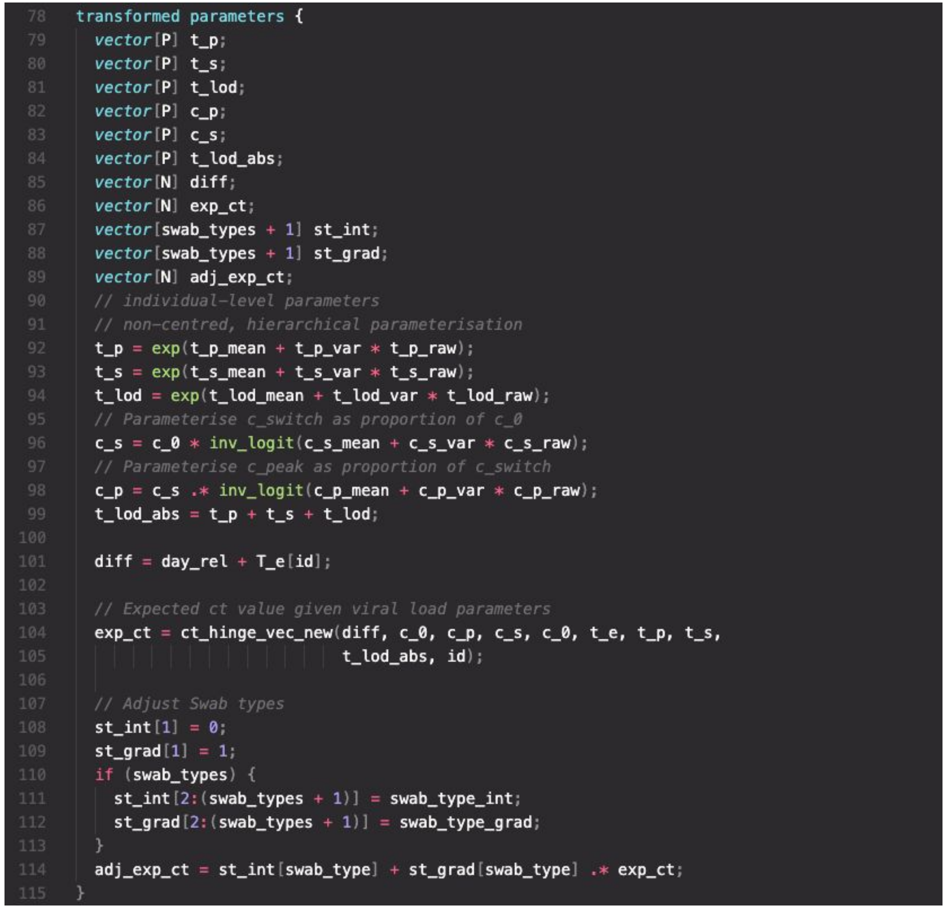
```
transformed data {
  vector[P] T_e_bound;
  for (i in 1:P) {
    T_e_bound[i] = max({-onset_time[i], 0});
  }
}
```

My first model - Parameters



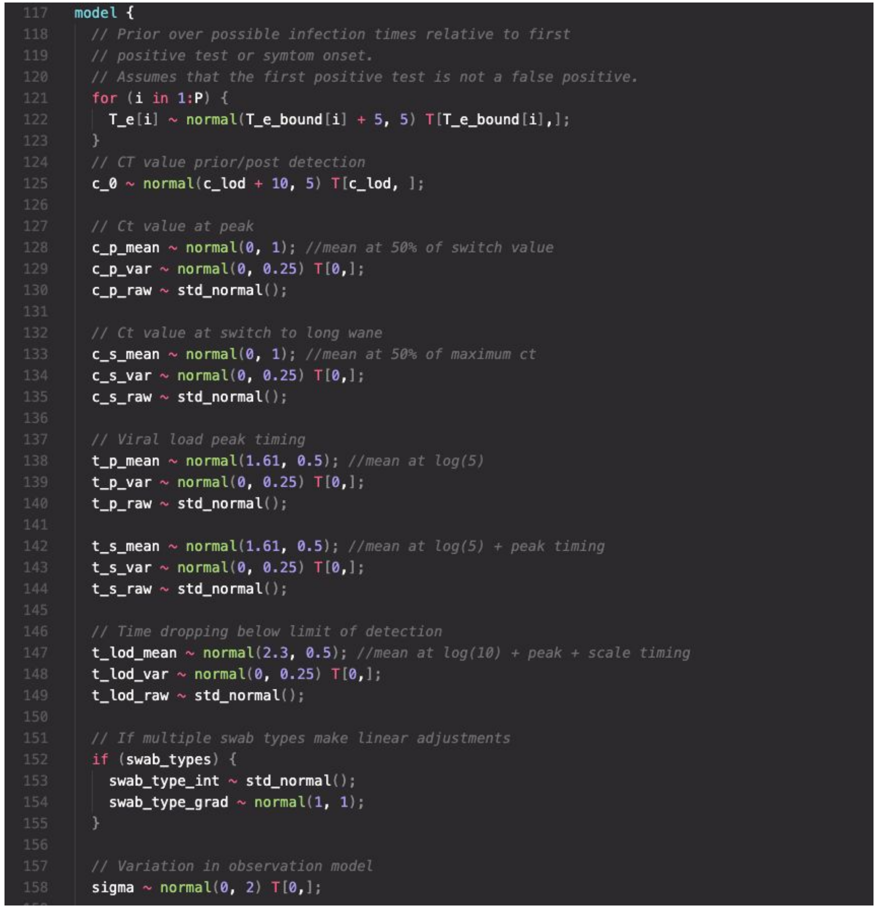
```
parameters {
  // Inferred time of infection
  vector<lower = T_e_bound>[P] T_e;

  //Incubation period
  real inc_mean[any_onsets];
  real<lower = 0> inc_sd[any_onsets];

  // Ct value before detection
  real<lower = c_lod> c_0;

  // Hyperparameters
  // Ct value of viral load p
  real c_p_mean;
  real<lower = 0>c_p_var;
  vector[P] c_p_raw;

  // Ct value at s
  real c_s_mean;
  real<lower = 0> c_s_var;
  vector[P] c_s_raw;

  // Timing of peak
  real t_p_mean;
  real<lower = 0> t_p_var;
  vector[P] t_p_raw;

  // Timing of switch
  real t_s_mean;
  real<lower = 0> t_s_var;
  vector[P] t_s_raw;

  // Time viral load hits lower limit of detection
  real t_lod_mean;
  real<lower = 0> t_lod_var;
  vector[P] t_lod_raw;

  // Swab type intercept and gradient
  vector[swab_types] swab_type_int;
  vector[swab_types] swab_type_grad;

  // Variance parameter for oobservation model
  real<lower = 0> sigma;
}
```

My first model - Transformed parameters



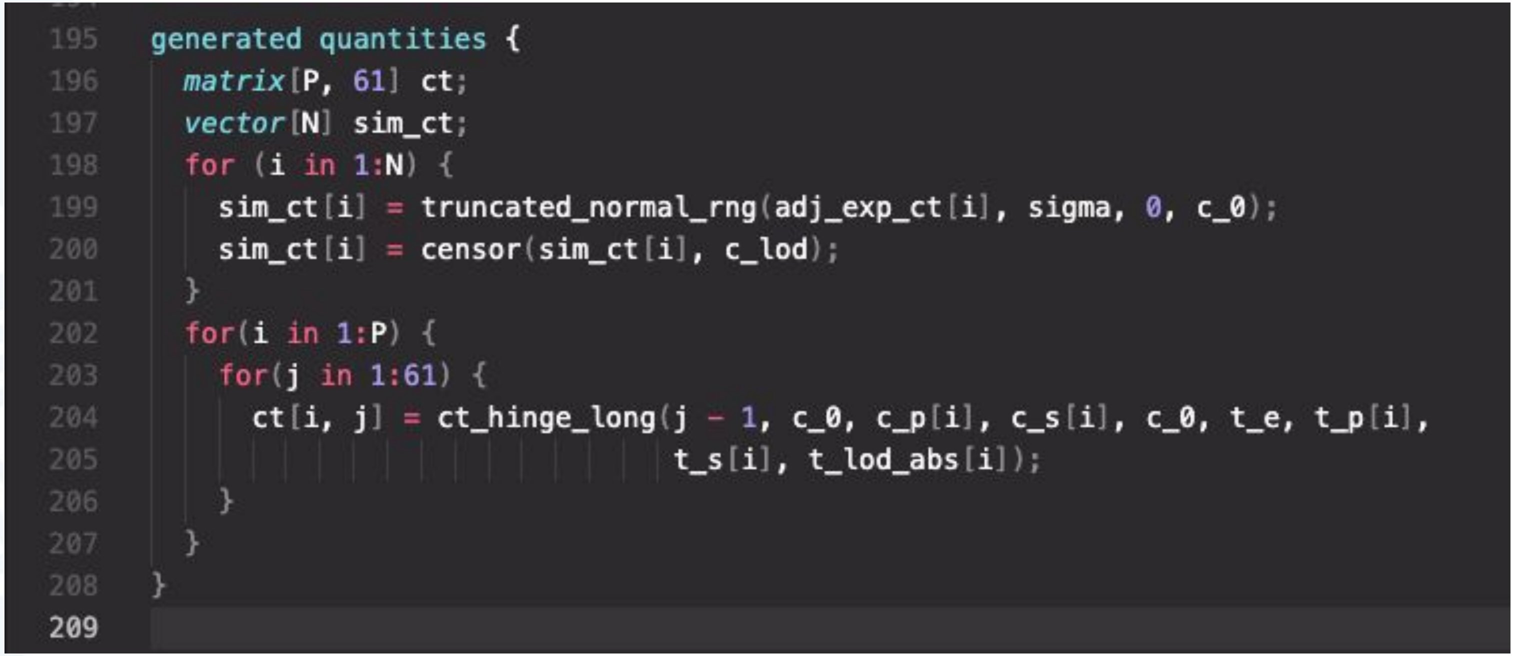
```
transformed parameters {
  vector[P] t_p;
  vector[P] t_s;
  vector[P] t_lod;
  vector[P] c_p;
  vector[P] c_s;
  vector[P] t_lod_abs;
  vector[N] diff;
  vector[N] exp_ct;
  vector[swab_types + 1] st_int;
  vector[swab_types + 1] st_grad;
  vector[N] adj_exp_ct;
  // individual-level parameters
  // non-centred, hierarchical parameterisation
  t_p = exp(t_p_mean + t_p_var * t_p_raw);
  t_s = exp(t_s_mean + t_s_var * t_s_raw);
  t_lod = exp(t_lod_mean + t_lod_var * t_lod_raw);
  // Parameterise c_switch as proportion of c_0
  c_s = c_0 * inv_logit(c_s_mean + c_s_var * c_s_raw);
  // Parameterise c_peak as proportion of c_switch
  c_p = c_s .* inv_logit(c_p_mean + c_p_var * c_p_raw);
  t_lod_abs = t_p + t_s + t_lod;

  diff = day_rel + T_e[id];

  // Expected ct value given viral load parameters
  exp_ct = ct_hinge_vec_new(diff, c_0, c_p, c_s, c_0, t_e, t_p, t_s,
                            t_lod_abs, id);

  // Adjust Swab types
  st_int[1] = 0;
  st_grad[1] = 1;
  if (swab_types) {
    st_int[2:(swab_types + 1)] = swab_type_int;
    st_grad[2:(swab_types + 1)] = swab_type_grad;
  }
  adj_exp_ct = st_int[swab_type] + st_grad[swab_type] .* exp_ct;
}
```

My first model - Model



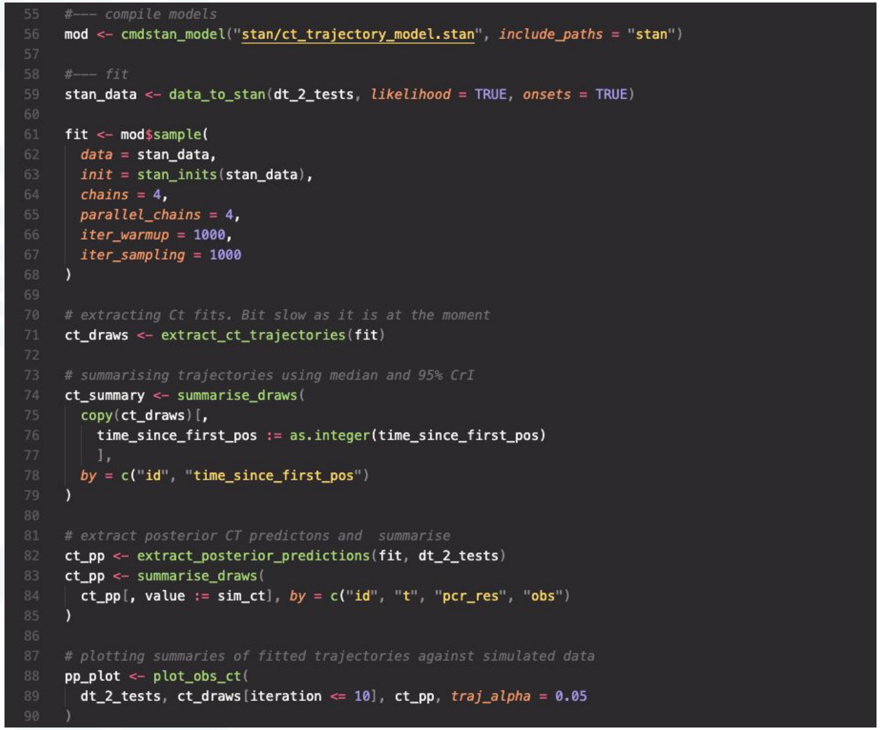
```
model {
  // Prior over possible infection times relative to first
  // positive test or symtom onset.
  // Assumes that the first positive test is not a false positive.
  for (i in 1:P) {
    T_e[i] ~ normal(T_e_bound[i] + 5, 5) T[T_e_bound[i],];
  }
  // CT value prior/post detection
  c_0 ~ normal(c_lod + 10, 5) T[c_lod, ];

  // Ct value at peak
  c_p_mean ~ normal(0, 1); //mean at 50% of switch value
  c_p_var ~ normal(0, 0.25) T[0,];
  c_p_raw ~ std_normal();

  // Ct value at switch to long wane
  c_s_mean ~ normal(0, 1); //mean at 50% of maximum ct
  c_s_var ~ normal(0, 0.25) T[0,];
  c_s_raw ~ std_normal();

  // Viral load peak timing
  t_p_mean ~ normal(1.61, 0.5); //mean at log(5)
  t_p_var ~ normal(0, 0.25) T[0,];
  t_p_raw ~ std_normal();

  t_s_mean ~ normal(1.61, 0.5); //mean at log(5) + peak timing
  t_s_var ~ normal(0, 0.25) T[0,];
  t_s_raw ~ std_normal();

  // Time dropping below limit of detection
  t_lod_mean ~ normal(2.3, 0.5); //mean at log(10) + peak + scale timing
  t_lod_var ~ normal(0, 0.25) T[0,];
  t_lod_raw ~ std_normal();

  // If multiple swab types make linear adjustments
  if (swab_types) {
    swab_type_int ~ std_normal();
    swab_type_grad ~ normal(1, 1);
  }

  // Variation in observation model
  sigma ~ normal(0, 2) T[0,];
```

My first model - Generated quantities

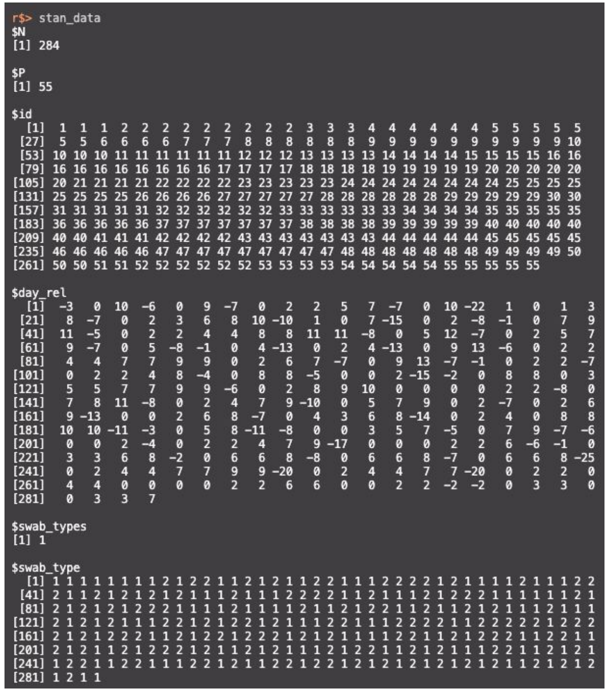


```
generated quantities {
  matrix[P, 61] ct;
  vector[N] sim_ct;
  for (i in 1:N) {
    sim_ct[i] = truncated_normal_rng(adj_exp_ct[i], sigma, 0, c_0);
    sim_ct[i] = censor(sim_ct[i], c_lod);
  }
  for(i in 1:P) {
    for(j in 1:61) {
      ct[i, j] = ct_hinge_long(j - 1, c_0, c_p[i], c_s[i], c_0, t_e, t_p[i],
                               t_s[i], t_lod_abs[i]);
    }
  }
}
```

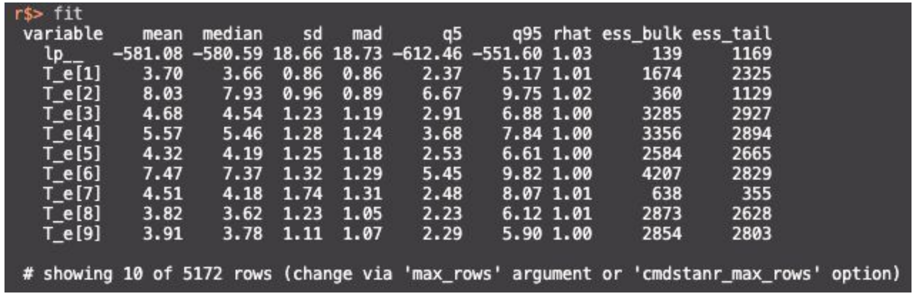
My first model - Fitting



```r
#--- compile models
mod <- cmdstan_model("stan/ct_trajectory_model.stan", include_paths = "stan")

#--- fit
stan_data <- data_to_stan(dt_2_tests, likelihood = TRUE, onsets = TRUE)

fit <- mod$sample(
  data = stan_data,
  init = stan_inits(stan_data),
  chains = 4,
  parallel_chains = 4,
  iter_warmup = 1000,
  iter_sampling = 1000
)

# extracting Ct fits. Bit slow as it is at the moment
ct_draws <- extract_ct_trajectories(fit)

# summarising trajectories using median and 95% CrI
ct_summary <- summarise_draws(
  copy(ct_draws)[,
    time_since_first_pos := as.integer(time_since_first_pos)
    ],
  by = c("id", "time_since_first_pos")
)

# extract posterior CT predictons and  summarise
ct_pp <- extract_posterior_predictions(fit, dt_2_tests)
ct_pp <- summarise_draws(
  ct_pp[, value := sim_ct], by = c("id", "t", "pcr_res", "obs")
)

# plotting summaries of fitted trajectories against simulated data
pp_plot <- plot_obs_ct(
  dt_2_tests, ct_draws[iteration <= 10], ct_pp, traj_alpha = 0.05
)
```



```
r$> stan_data
$N
[1] 284

$P
[1] 55

$id
  [1]  1  1  1  2  2  2  2  2  2  2  2  2  3  3  3  4  4  4  4  4  4  5  5  5  5  5
 [27]  5  5  6  6  6  6  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9 10
 [53] 10 10 10 11 11 11 11 11 11 12 12 12 13 13 13 13 14 14 14 14 15 15 15 15 16 16
 [79] 16 16 16 16 16 16 16 16 17 17 17 17 18 18 18 18 19 19 19 19 19 20 20 20 20 20
[105] 20 21 21 21 21 22 22 22 22 23 23 23 23 23 24 24 24 24 24 24 24 24 25 25 25 25
[131] 25 25 25 25 26 26 26 26 27 27 27 27 27 28 28 28 28 28 28 29 29 29 29 29 30 30
[157] 31 31 31 31 31 32 32 32 32 32 32 33 33 33 33 33 33 34 34 34 34 35 35 35 35 35
[183] 36 36 36 36 36 37 37 37 37 37 37 37 38 38 38 38 39 39 39 39 39 40 40 40 40 40
[209] 40 40 41 41 41 42 42 42 42 43 43 43 43 43 43 43 44 44 44 44 44 45 45 45 45 45
[235] 46 46 46 46 46 47 47 47 47 47 47 47 47 47 48 48 48 48 48 48 48 49 49 49 49 50
[261] 50 50 51 51 52 52 52 52 52 52 53 53 53 53 54 54 54 54 54 55 55 55 55 55

$day_rel
  [1]  -3   0  10  -6   0   9  -7   0   2   2   5   7  -7   0  10 -22   1   0   1   3
 [21]   8  -7   0   2   3   6   8  10 -10   1   0   7 -15   0   2  -8  -1   0   7   9
 [41]  11  -5   0   2   2   4   4   8   8  11  11  -8   0   5  12  -7   0   2   5   7
 [61]   9  -7   0   5  -8  -1   0   4 -13   0   2   4 -13   0   9  13  -6   0   2   2
 [81]   4   4   7   7   9   9   0   2   6   7  -7   0   9  13  -7  -1   0   2   2  -7
[101]   0   2   2   4   8  -4   0   8   8  -5   0   0   2 -15  -2   0   8   8   0   3
[121]   5   5   7   7   9   9  -6   0   2   8   9  10   0   0   0   0   2   2  -8   0
[141]   7   8  11  -8   0   2   4   7   9 -10   0   5   7   9   0   2  -7   0   2   6
[161]   9 -13   0   0   2   6   8  -7   0   4   3   6   8 -14   0   2   4   0   8   8
[181]  10  10 -11  -3   0   5   8 -11  -8   0   0   3   5   7  -5   0   7   9  -7  -6
[201]   0   0   2  -4   0   2   2   4   7   9 -17   0   0   0   2   2   6  -6  -1   0
[221]   3   3   6   8  -2   0   6   6   8  -8   0   6   6   8  -7   0   6   6   8 -25
[241]   0   2   4   4   7   7   9   9 -20   0   2   4   4   7   7 -20   0   2   2   0
[261]   4   4   0   0   0   0   2   2   6   6   0   0   2   2  -2  -2   0   3   3   0
[281]   0   3   3   7

$swab_types
[1] 1

$swab_type
  [1] 1 1 1 1 1 1 1 1 2 1 2 2 1 1 2 1 2 1 1 2 2 1 1 1 2 2 2 2 1 2 1 1 1 1 2 1 1 1 2 2
 [41] 2 1 1 2 1 2 1 2 1 2 1 1 1 2 1 1 1 2 2 2 2 1 1 2 1 1 1 2 1 2 2 2 1 1 1 1 1 1 2 1
 [81] 2 1 2 1 2 1 2 2 2 1 1 1 1 2 1 1 1 1 2 1 1 2 1 2 2 1 1 2 1 1 2 1 2 1 1 1 2 1 1 1
[121] 2 1 2 1 2 1 1 1 2 2 1 2 1 2 2 1 2 1 1 1 2 2 1 1 1 2 2 2 2 1 1 2 2 2 2 2 1 2 2 2
[161] 2 1 2 1 2 2 2 1 1 2 1 2 2 1 2 2 2 1 2 1 2 1 1 1 1 2 2 1 1 2 1 2 2 2 1 1 2 2 1 1
[201] 2 1 2 1 1 2 1 2 2 2 1 2 1 1 2 1 2 1 1 1 2 1 2 2 1 1 2 1 2 1 1 1 2 2 1 1 2 1 2 1
[241] 1 2 2 1 1 2 2 1 1 1 2 2 1 2 1 1 2 2 1 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1 2 1 2 1 2
[281] 1 2 1 1
```

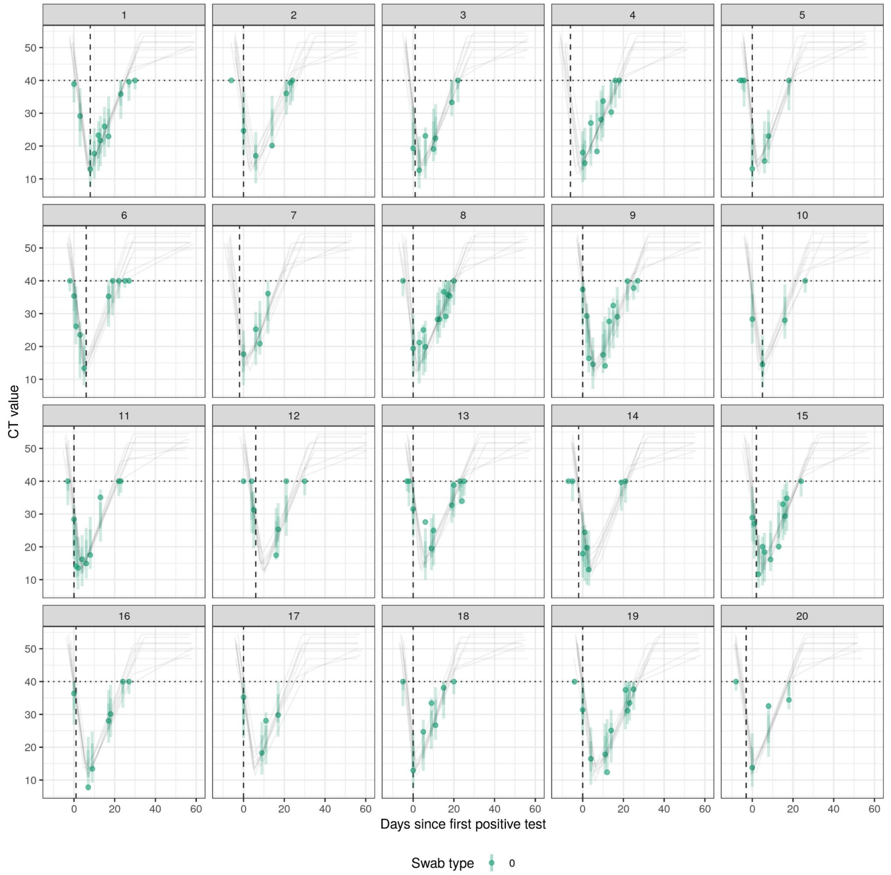


```
r$> fit
 variable    mean  median    sd   mad      q5     q95 rhat ess_bulk ess_tail
   lp__   -581.08 -580.59 18.66 18.73 -612.46 -551.60 1.03      139     1169
   T_e[1]    3.70    3.66  0.86  0.86    2.37    5.17 1.01     1674     2325
   T_e[2]    8.03    7.93  0.96  0.89    6.67    9.75 1.02      360     1129
   T_e[3]    4.68    4.54  1.23  1.19    2.91    6.88 1.00     3285     2927
   T_e[4]    5.57    5.46  1.28  1.24    3.68    7.84 1.00     3356     2894
   T_e[5]    4.32    4.19  1.25  1.18    2.53    6.61 1.00     2584     2665
   T_e[6]    7.47    7.37  1.32  1.29    5.45    9.82 1.00     4207     2829
   T_e[7]    4.51    4.18  1.74  1.31    2.48    8.07 1.01      638      355
   T_e[8]    3.82    3.62  1.23  1.05    2.23    6.12 1.01     2873     2628
   T_e[9]    3.91    3.78  1.11  1.07    2.29    5.90 1.00     2854     2803

 # showing 10 of 5172 rows (change via 'max_rows' argument or 'cmdstanr_max_rows' option)
```

rstan

cmdstanr

My first model - Posterior predictions



My first model - Summary

Stan - an introduction without the scary parts

What is it good for?

Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE &TROPICAL MEDICINE

Why

- Are you an expert in MCMC? No? Then maybe don’t roll your own because it is not the 1990s anymore.

- Community, documentation, and ongoing development.

- A good enough tool for a broad range of problems. Useful if wanting to do anything other than compartmental models

- Clean domain-specific language (DSL) which can be extended with functions written in C++.

- Huge range of tools available for evaluating stan models and handling the output more generally

- Massive amount of work done in stan - easy to understand others work.

Stan - an introduction without the scary parts

What is it not good for?

Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE & TROPICAL MEDICINE

Why not

- MCMC often not a great choice for complex compartmental model systems. For these models tools that support PMCMC or SMC^2 are likely more optimal (Libbi/Birch, ODIN/Dust, Turing.jl, etc).
- Has to compile. You have to interact, debug etc with compiled code. This is gross.
- Unless you are pro debugging involves trying to fit the model again and again. Not much fun.
- Hard to use programmatically across models (you may find yourself writing a DSL generator -though see {adjustr}).
- Not a full language so for software development can be limiting. Better bets here are Turing.jl, PyMc3, numpyro etc.
- Interaction with fit model objects is not ideal but this is an area of work in the community (and the situation is much better than for other tools).

Stan - an introduction without the scary parts

How did learning stan make you feel Sam?

Sam Abbott
@seabbs
samabbott.co.uk

Sad, confused, bemused, frightened, and angry

Stan - an introduction without the scary parts

How do you feel now Sam?

Sam Abbott
@seabbs
samabbott.co.uk

(still no idea what is happening)

Stan - an introduction without the scary parts

Summary

Sam Abbott
@seabbs
samabbott.co.uk

centre *for the* mathematical modelling *of* infectious diseases

LONDON SCHOOL *of* HYGIENE &TROPICAL MEDICINE

Summary

- Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation.
- There are now several of these. Stan's key strength is its community and large suite of tools. It is also as fast or faster than the competition
- Weakness is that limited to what stan supports, it is a compiled language, and not a full programming language.
- Join the #stan slack and get chatting (it is just me talking to myself at the moment).

More from me?

- Some notes on stan for compartmental models + resources: https://github.com/seabbs/cmmid.stan.seir
- Here is a recording + slides of me talking about my work on COVID-19 (all done in stan) with links out to all the code etc: https://bit.ly/covid-19-case-studies