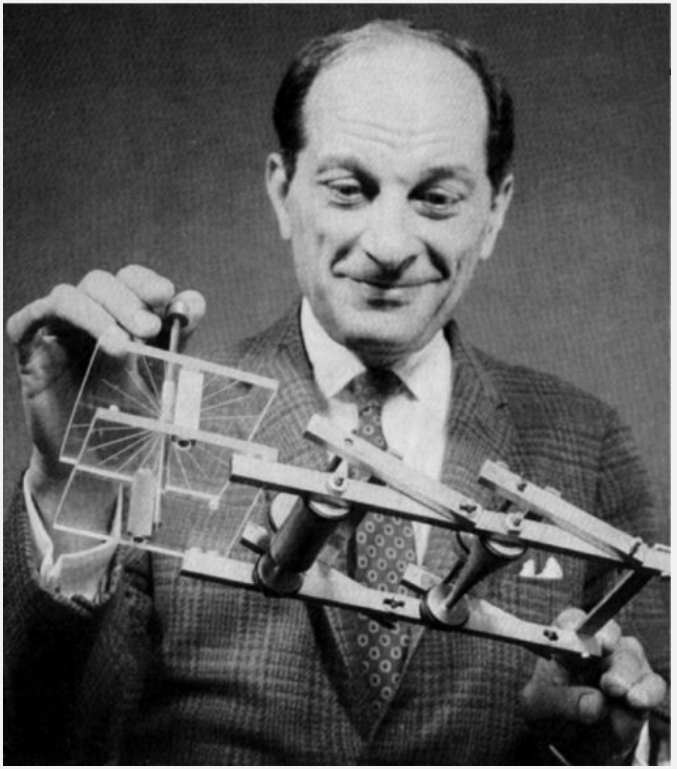
The RStan package

EdinbR July 2019

The Stan project

- Language for probabilistic programming
- Differentiable mathematics and probability library
- Algorithms for Bayesian posterior inference and analysis
- Platforms: Windows, MacOS X, Linux.
- Interfaces: R, Python, Julia, Matlab, Mathematica, Stata
- Documentation:
  - ▷ Stan Users Guide (311 pages)
  - ▷ Stan Reference Manual and Users Guide (642 pages)
- Web `http://mc-stan.org`



Stan Ulam holding the Fermiac. Ulam, von Neumann and Metropolis invented the Monte Carlo method in the 1950s

Other Bayesian programs

- CRAN task view 'Bayesian inference'

  Long list of Bayesian packages, mostly focused on a particular type of model.

- WinBUGS/OpenBUGS/JAGS

  General purpose, based on Gibbs sampling.
  Code is declarative/interpreted.
  Code → Directed Acyclic Graph

- Stan

  Based on Hamiltonian Monte Carlo. HMC sampling is more robust, scalable and efficient than Gibbs.
  Stan code is imperative/converted to C++ and compiled.
  Code → log probability function and its derivatives

Bayes' formula

Data Y, unknown parameter(s) θ,

$$\begin{array}{lcl} P(\theta \mid Y) & \propto & P(Y \mid \theta) * P(\theta) \\ \text{(posterior)} & \propto & \text{(likelihood)} * \text{(prior)} \end{array}$$

Multiplicative factors in the likelihood which do not involve θ can be discarded. Choice of prior can be a problem: stan has reasonable defaults, which depend on parameter geometry.

Support	Transformation	Example
$(-\infty, +\infty)$	identity	normal mean
$(0, +\infty)$	log	normal variance
$(0, 1)$	log odds	probability

The user can override the defaults.

Hamiltonian Monte Carlo

- A frictionless puck slides over a surface of variable height. State of system is described by its *position* (q) and its *momentum* (p). Potential energy $U(q)$ is proportional to current height of puck, and kinetic energy $K(p)$ is proportional to $|p|^2$.

- On the level, puck moves at constant velocity. On a rising slope $K(p)$ decreases and $U(q)$ increases, until $K(p)$ is zero and the puck slides back down the slope. The Hamiltonian $H(q, p) = U(q) + K(p)$ is constant.

- In MCMC, position coords q correspond to values of the parameters, potential energy is minus the log p.d.f. evaluated at these values. Momentum variables are introduced artificially.

From Neal (2011) MCMC using Hamiltonian dynamics. In Brooks et al, Handbook of Markov Chain Monte Carlo, 116-162, Chapman & Hall/CRC

Genetic linkage

Maize plants are categorized as type A, B or C

phenotype	A	B	C
probability	$\frac{2+\theta}{4}$	$\frac{1-\theta}{2}$	$\frac{\theta}{4}$
number	1997	1810	32

The parameter θ measures linkage between two segregating genetic loci.

The frequencies Y_1 ... Y_3 have a multinomial distn and the log-likelihood is

$$Y_1 \log(2+\theta) + Y_2 \log(1-\theta) + Y_3 \log \theta$$

Stan code for genetic linkage (1st version)

```
parameters {
real<lower=0,upper=1> theta;
}

model {
target += 1997 * log(2 + theta) + 1810 * log(1 - theta)
            + 32 * log(theta);
}
```

Model block does not specify a probability distn for θ

Hence default prior (uniform on logit scale).

Stan code for genetic linkage (2nd version)

```
data {
int<lower=0> y[3];
}
parameters {
real<lower=0,upper=1> theta;
}
transformed parameters {
simplex[3] p;
}
p[1] = 0.25 * (2 + theta);
p[2] = 0.5 * (1 - theta);
p[3] = 1 - p[1] - p[2];
}
model {
y ~ multinomial(p);
}
```

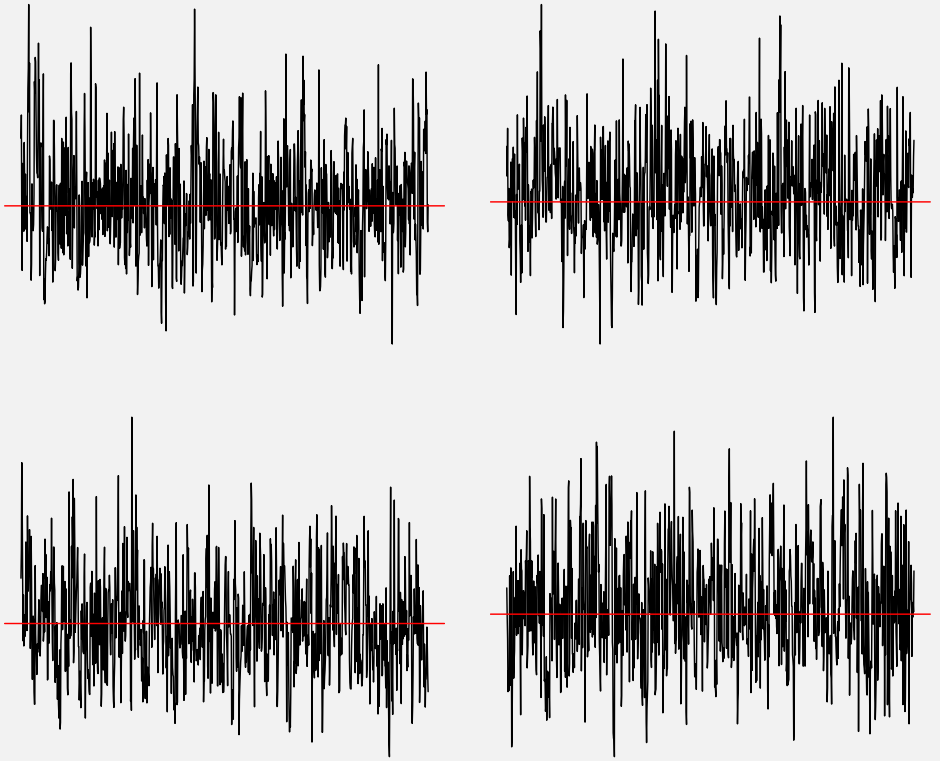
R code for genetic linkage example

```
library(rstan)

y <- c(1997, 1810, 32)

fit <- stan('linkage.stan')

## linkage.stan is a file containing the stan code

print(fit, par = 'theta', digits = 4, probs = c(0.025, 0.975))

4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

        mean se_mean     sd   2.5%  97.5% n_eff   Rhat
theta 0.0366   2e-04 0.0061 0.0256 0.0491  1482 1.0023

Samples were drawn using NUTS(diag_e) at Mon Jul 15 16:27:42 2019.
```

Graphical output

```
## rstan functions (based on ggplot2)
traceplot(fit)
stan_hist(fit)

## or do it yourself
out <- extract(fit, 'theta', permute = FALSE)$theta

oldpar <- par(mar = c(1,1,1,1), mfrow = c(2,2))

for (chain in 1:4) {
plot(out[,chain,], type = 'l', axes = FALSE, ann = FALSE)
abline(h = 0.037, col = 'red')
}
par(oldpar)

plot(density(out), las = 1, ann = FALSE)
hist(out, freq = FALSE, las = 1, border = 'lightblue', add = TRUE)
```

Trace plots



Histogram and density plot