One parameter models in Stan - part 1

Data Analytics

Jerzy Baranowski

Data analytics in times of Covid-19

- Main lecture content on Youtube (around 2 videos per week)
    - Questions in comments and through e-mail (assumed anonymity unless otherwise specified)
- Answers to questions in the form of Q&A videos (once per week)
- Laboratory work assigned through GitHub

Reading announcements

Books

- Bayesian Data Analysis 3 is now free for non-commercial use
  - At least third of material on this course comes from this book
  - You can find it here: https://github.com/avehtari/BDA_course_Aalto (along with lot of the material)
- Second edition of Statistical Rethinking was published
  - Second pillar of our course
  - All the details here: https://xcelab.net/rm/statistical-rethinking/
- Michael Betancourt's intro to Stan: https://betanalpha.github.io/assets/case_studies/stan_intro.html

In the previous episode

- During our last lecture we have:
  - Introduced concepts of Bayesian statistics
    - Prior predictive distribution $p(y) = \int p(y, \theta) d\theta = \int p(\theta) p(y \mid \theta) d\theta$
    - Posterior predictive distribution $$p(\tilde{y} \mid y) = \int p(\tilde{y}, \theta \mid y) d\theta = \int p(\tilde{y} \mid \theta, y) p(\theta \mid y) d\theta = \int p(\tilde{y} \mid \theta) p(\theta \mid y) d\theta$$
  - We analysed our first one-parameter models
    - By analytical computation
    - By grid approximation

How can we analyse Bayesian models?

- Grid approximation
- Quadratic approximation
- **Probabilistic programming**

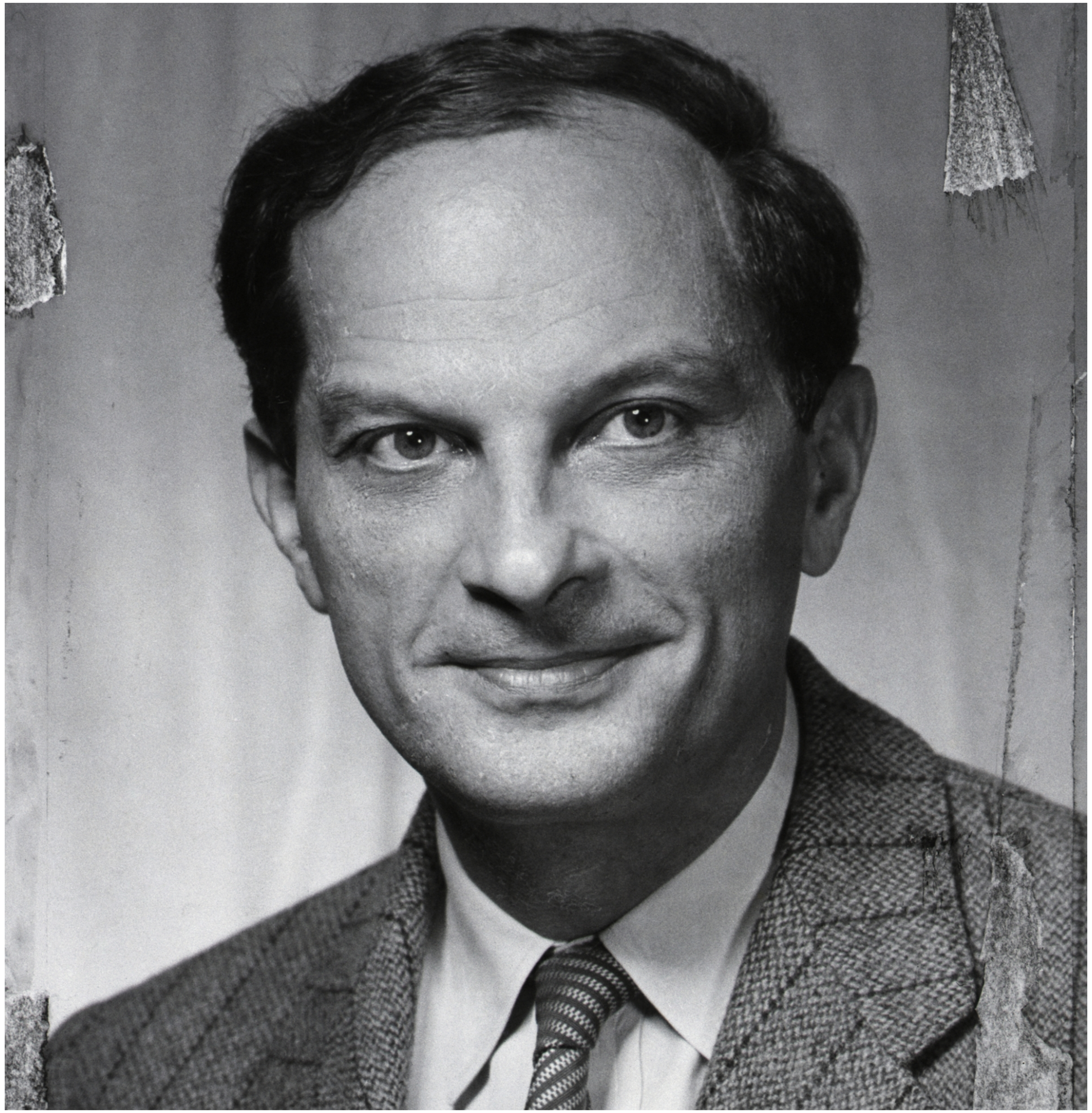
Stan is a statistical modeling language

It offers:

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

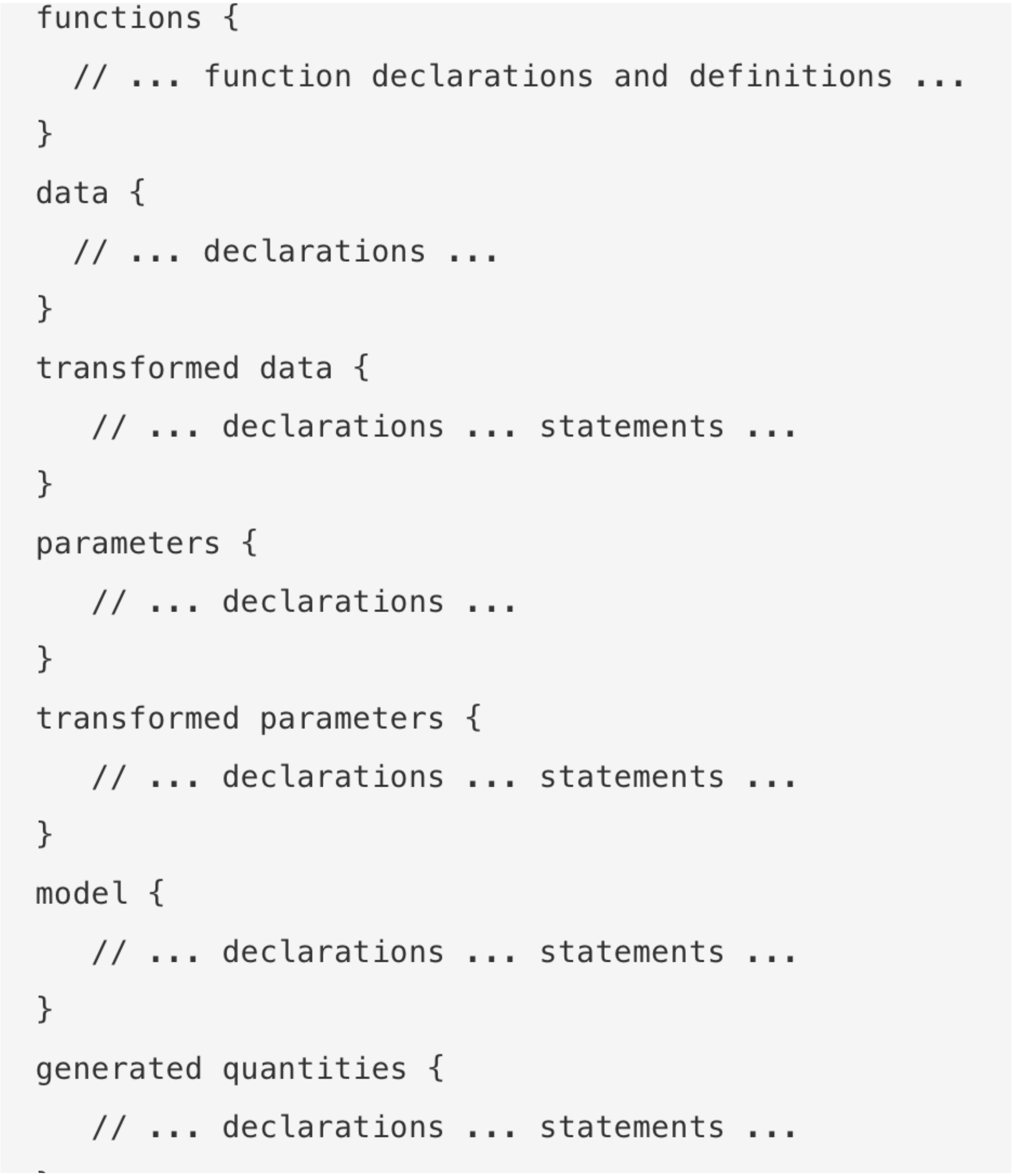
Name is not acronym

It is named after Stanisław Ulam, one of the creators of Monte Carlo methods who worked in Los Alamos developing atomic bomb



Stan programs

- In Stan, we create probabilistic models in stan language
- They are then converted to c++ code, which is subsequently compiled
- Complied model are run from the level of Python, R, or other stuff.



```
functions {
  // ... function declarations and definitions ...
}
data {
  // ... declarations ...
}
transformed data {
   // ... declarations ... statements ...
}
parameters {
   // ... declarations ...
}
transformed parameters {
   // ... declarations ... statements ...
}
model {
   // ... declarations ... statements ...
}
generated quantities {
   // ... declarations ... statements ...
```

Revisiting globe tossing

- We wanted to estimate the water coverage of planet by random sampling of its surface.
- Or in other words we have been throwing an earth shaped ball around and recording whether right thumb of the catcher was on the water or on the land

$$p(y \mid \theta) = \mathrm{Binomial}(y \mid n, \theta) = \binom{n}{y} \theta^y (1 - \theta)^{n-y}$$