Bayesian Statistics and Hierarchical Bayesian Modeling for Psychological Science

Lecture 05

Lei Zhang

Social, Cognitive and Affective Neuroscience Unit (SCAN-Unit)

Department of Cognition, Emotion, and Methods in Psychology

lei.zhang@univie.ac.at

lei-zhang.net

@lei_zhang_lz

https://github.com/lei-zhang/BayesCog_Wien

universität wien

Fakultät für Psychologie

Bayesian warm-up?

Binomial Model

- You are curious how much of the surface is covered in water.
- You will toss the globe up in the air.
- You will record whether or not the surface under your right index finger is water (W) or land (L).
- You might observe: W L W W W L W L W
- → 6/9 = 0.666667?
- Is it right? If not, what to do next?

McElreath (2016)

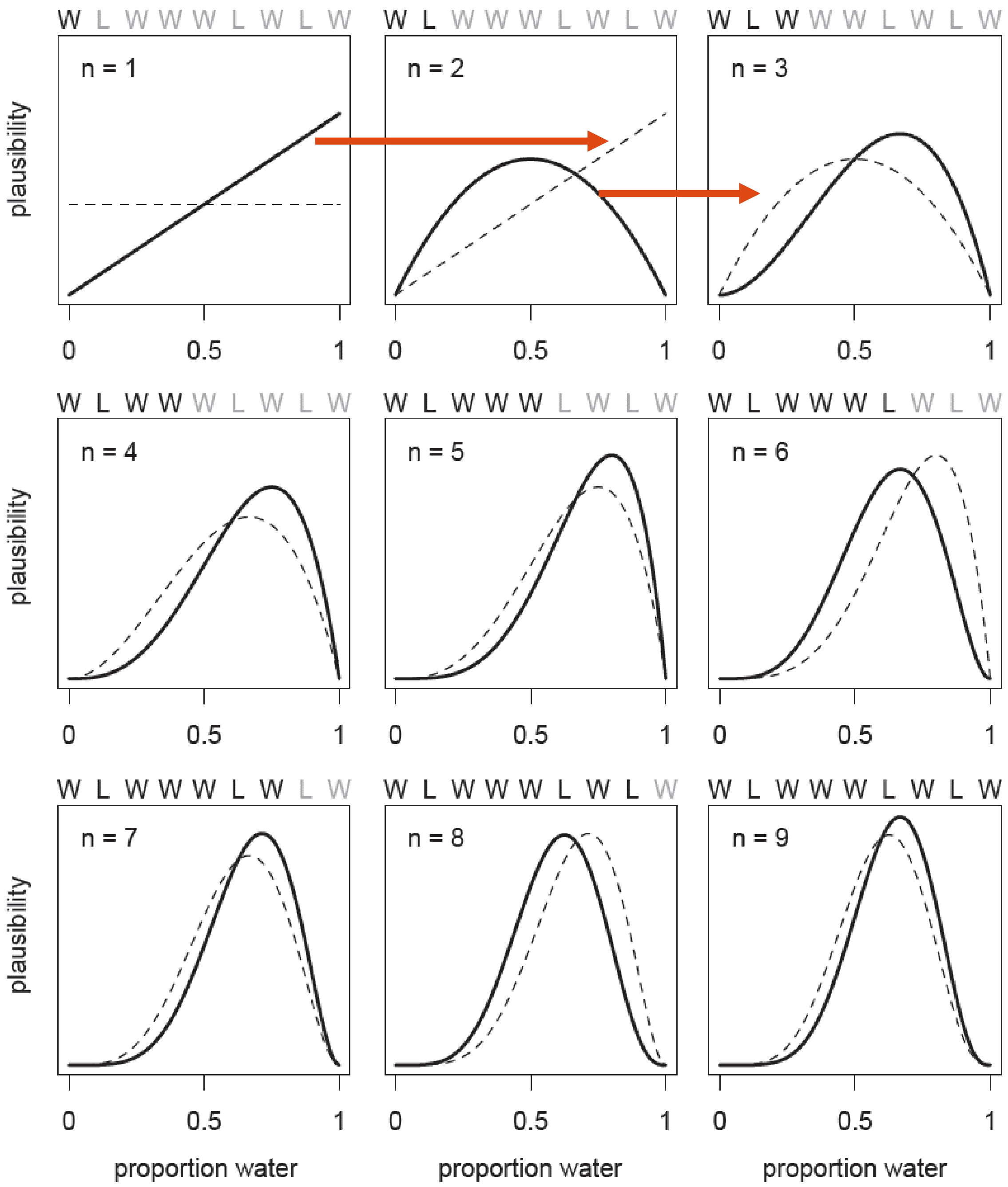
A Data Story of the Globe

cognitive model
statistics
computing

- The true proportion of water covering the globe is ϑ.
- A single toss of the globe has a probability ϑ of producing a water (W) observation.
- It has a probability $(1 - \vartheta)$ of producing a land (L) observation.
- Each toss of the globe is independent of the others.

McElreath (2016)

Update



- order doesn't matter
- 2/3 is most likely
- others are not ruled out

cognitive model
statistics
computing

McElreath (2016)

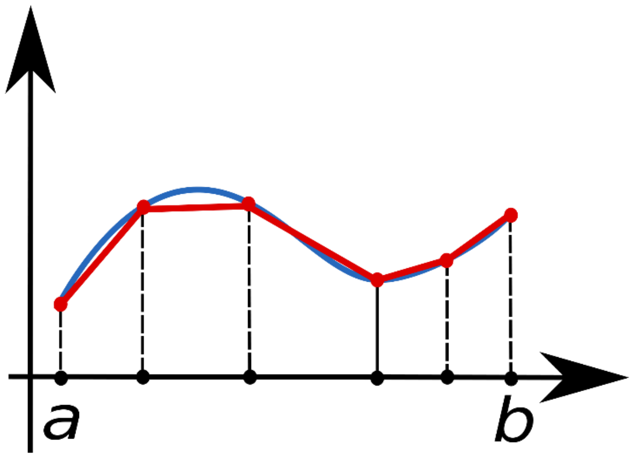
Solve it by Grid Approximation

discrete parameters

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\boxed{\sum_{\theta^*} p(D \mid \theta^*)\, p(\theta^*)}}$$

continuous parameters

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\boxed{\int p(D \mid \theta^*)\, p(\theta^*)\, d\theta^*}}$$

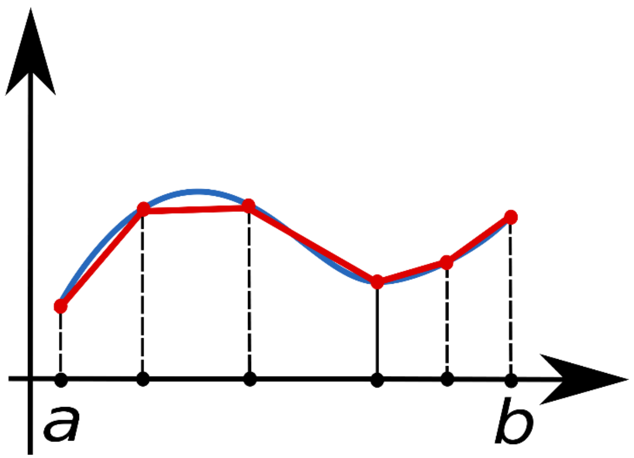


Binomial Model – Grid Approximation


```r
theta_start <- 0; theta_end <- 1; n_grid <- 20
w <- 6; N <- 9

# define grid
theta_grid <- seq(from = theta_start, to = theta_end,
                  length.out = n_grid)

# define prior
prior <- rep(1 , n_grid)

# compute likelihood at each value in grid
likelihood <- dbinom(w, size = N, prob = theta_grid)

# compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

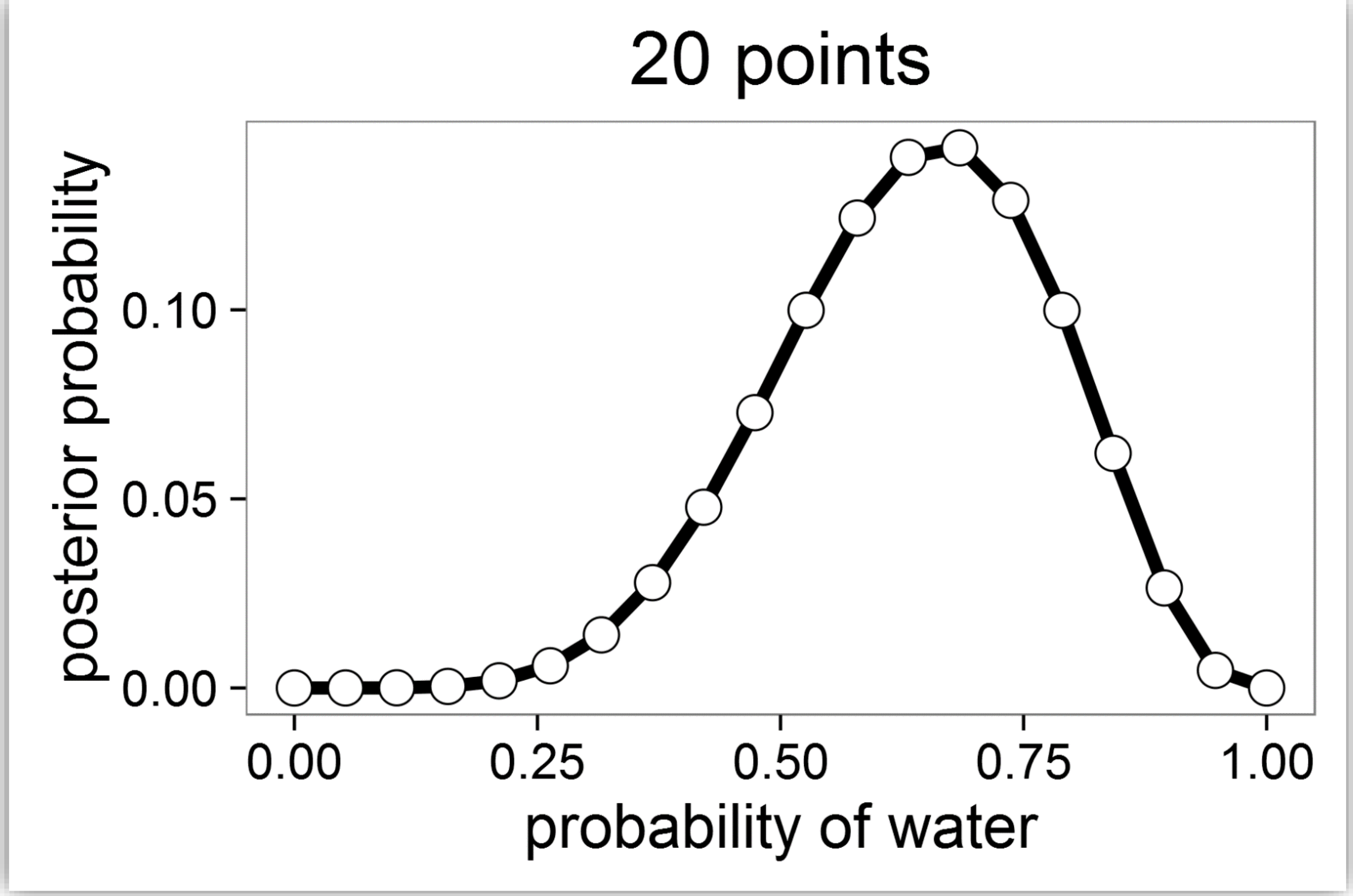
$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\int p(D \mid \theta^*)\, p(\theta^*)\, d\theta^*}$$

$$p(w \mid N, \theta) = \binom{N}{w} \theta^w (1-\theta)^{N-w}$$

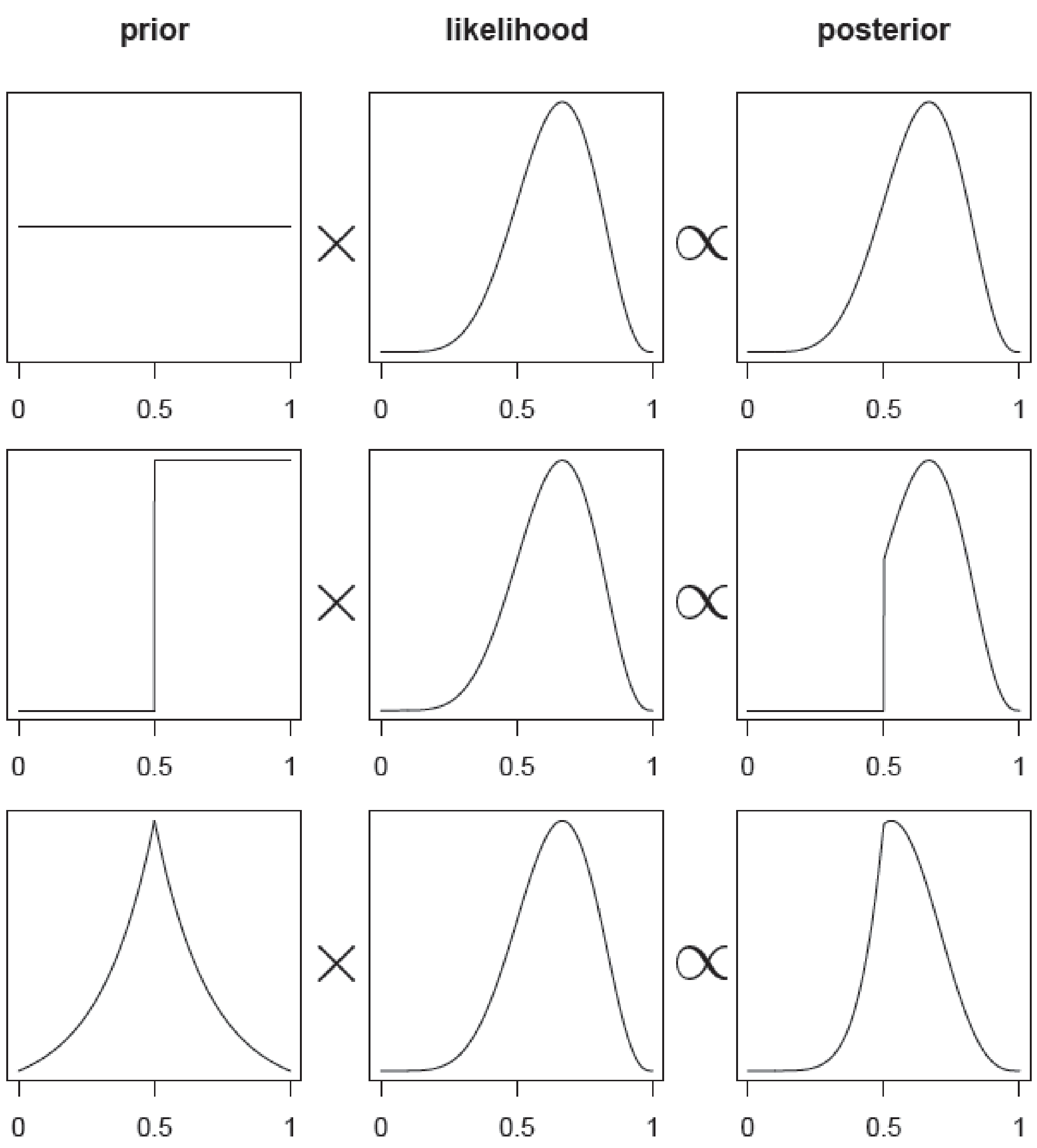


Binomial Model – Grid Approximation

cognitive model
statistics
computing



Impact of Prior



McElreath (2016)

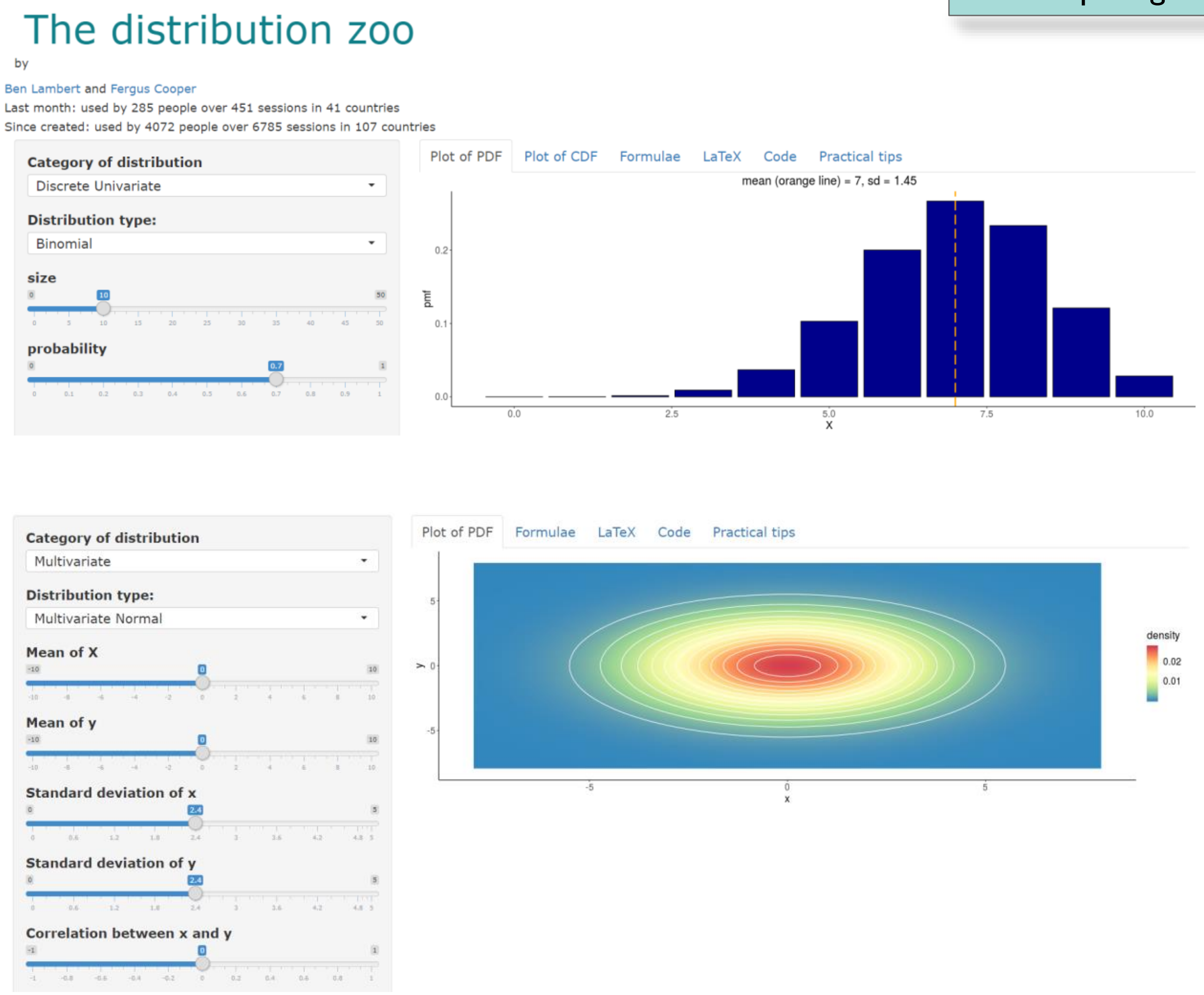
Exercise VII

cognitive model
statistics
computing

```
…/BayesCog/02.binomial_globe/_scripts/binomial_globe_grid.R
```

TASK: run a grid approximation with `grid_size = 50`

How do I know which likelihood to use?



Lambert (2018)

https://ben18785.shinyapps.io/distribution-zoo/

What if I have multiple parameters?

grid approximation for
2 parameters?
5 parameters?
10 parameters?

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\boxed{\int p(D \mid \theta^*)\, p(\theta^*)\, d\theta^*}}$$

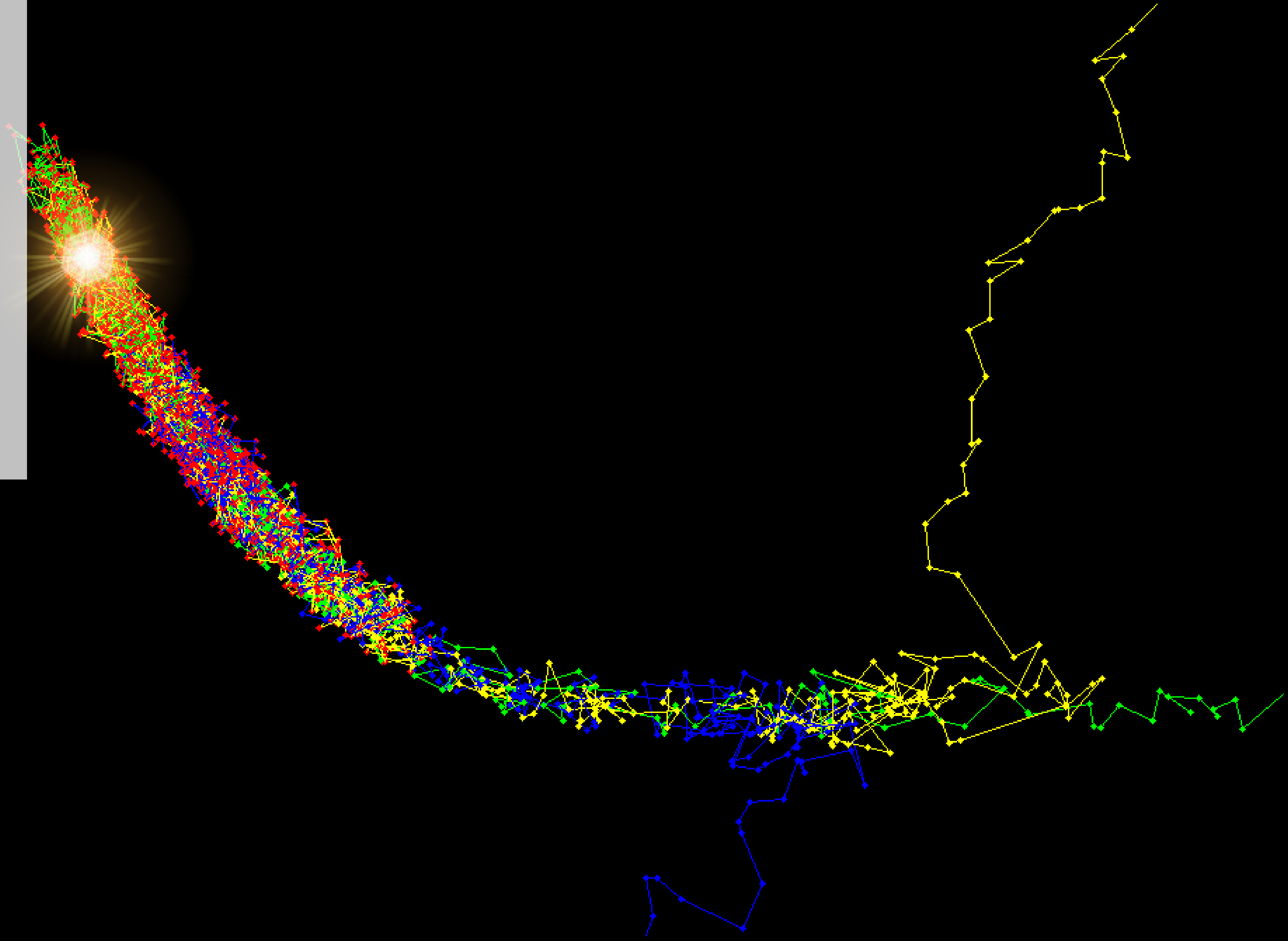
$$p(data) = \int_{\text{All}\,\theta_1} \int_{\text{All}\,\theta_2} p(data, \theta_1, \theta_2)\, \mathrm{d}\theta_1 \mathrm{d}\theta_2$$

$$p(data) = \int_{\mu_1} \int_{\sigma_1} \cdots \int_{\mu_{100}} \int_{\sigma_{100}} \underbrace{p(data \mid \mu_1, \sigma_1, \ldots, \mu_{100}, \sigma_{100})}_{\text{likelihood}} \times \underbrace{p(\mu_1, \sigma_1, \ldots, \mu_{100}, \sigma_{100})}_{\text{prior}}$$
$$\mathrm{d}\mu_1 \mathrm{d}\sigma_1 \ldots \mathrm{d}\mu_{100} \mathrm{d}\sigma_{100},$$

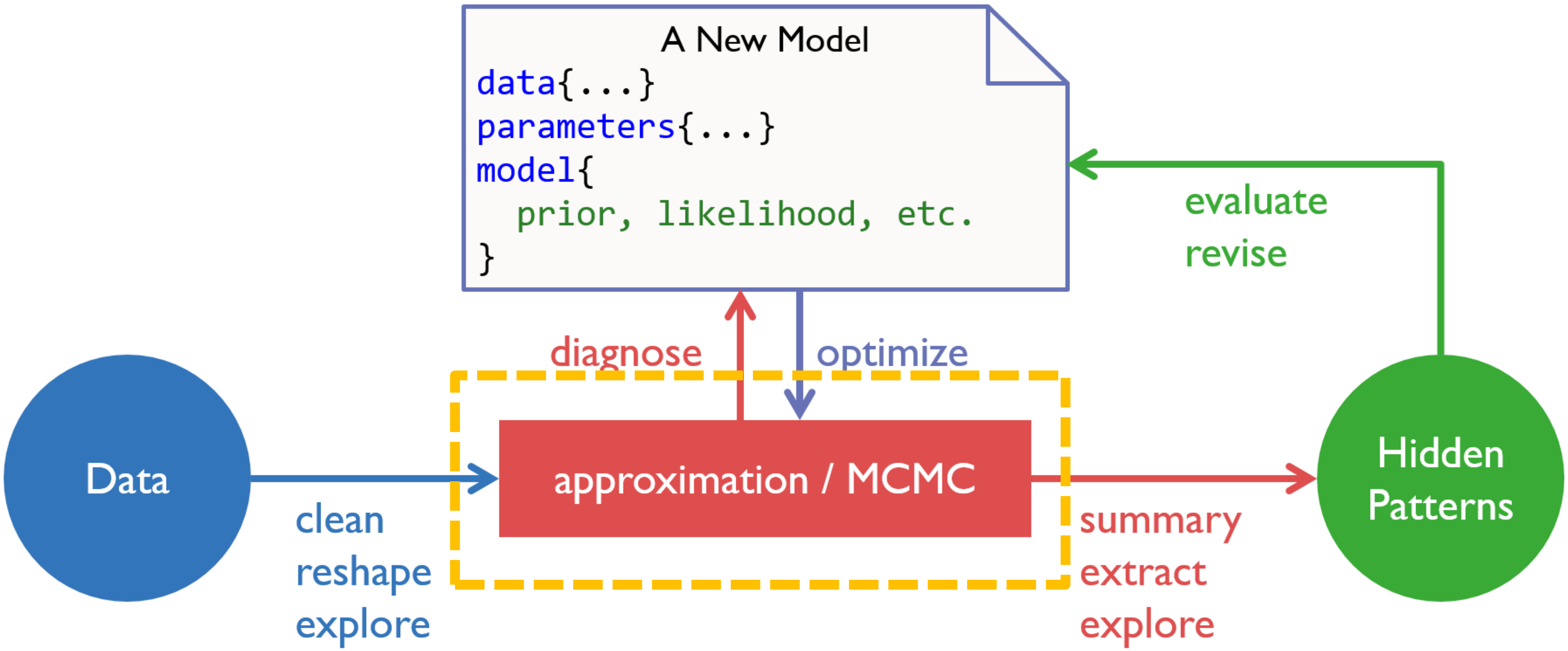
- Analytical solutions (often does not exist)
- Grid approximation (takes too long)
- Markov Chain Monte Carlo

$$\boxed{p(\theta \mid D) \propto p(D \mid \theta)\, p(\theta)}$$

MARKOV CHAIN MONTE CARLO



cognitive model
statistics
computing



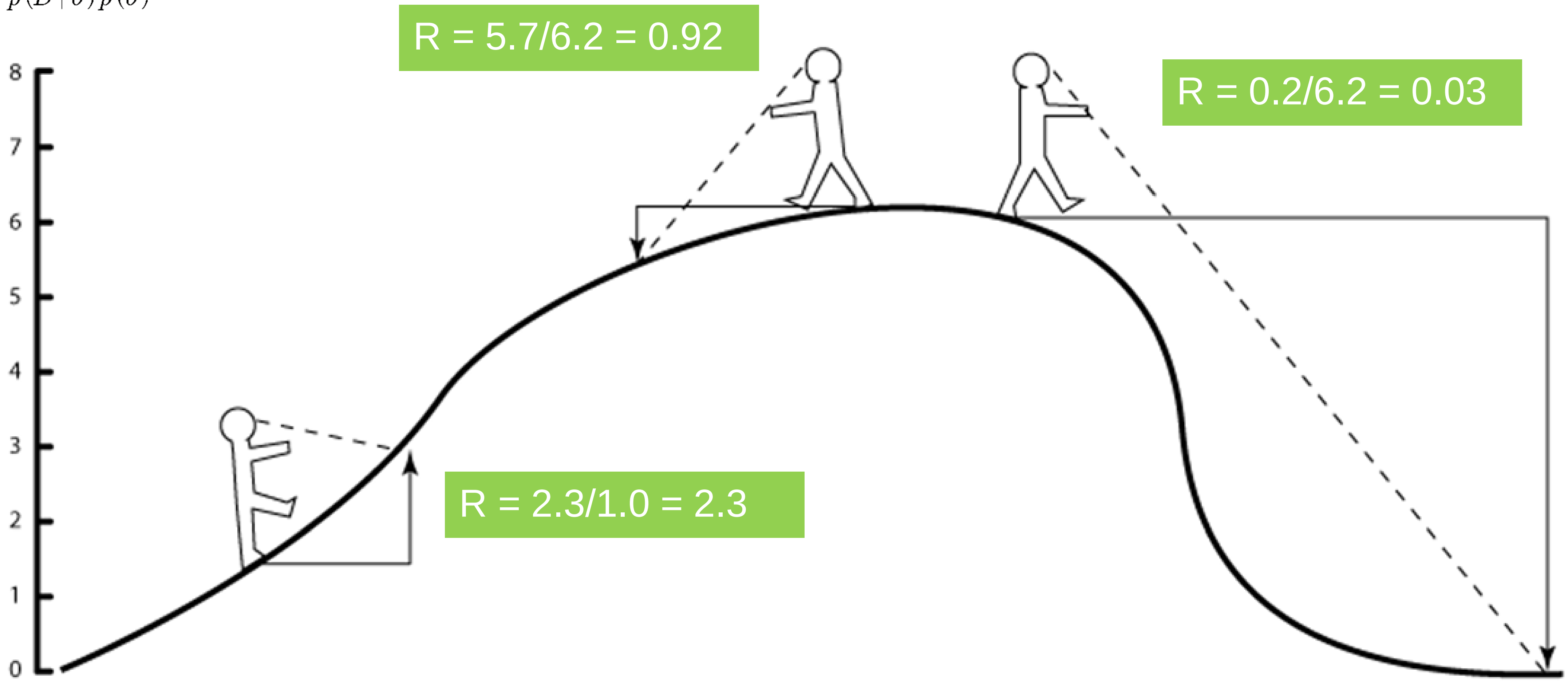
Solving the Problem by Approximation

$$p(\theta \mid D) \propto p(D \mid \theta)\, p(\theta)$$

Deterministic Approximation	Stochastic Approximation
→ Variational Bayes	→ Sampling Methods

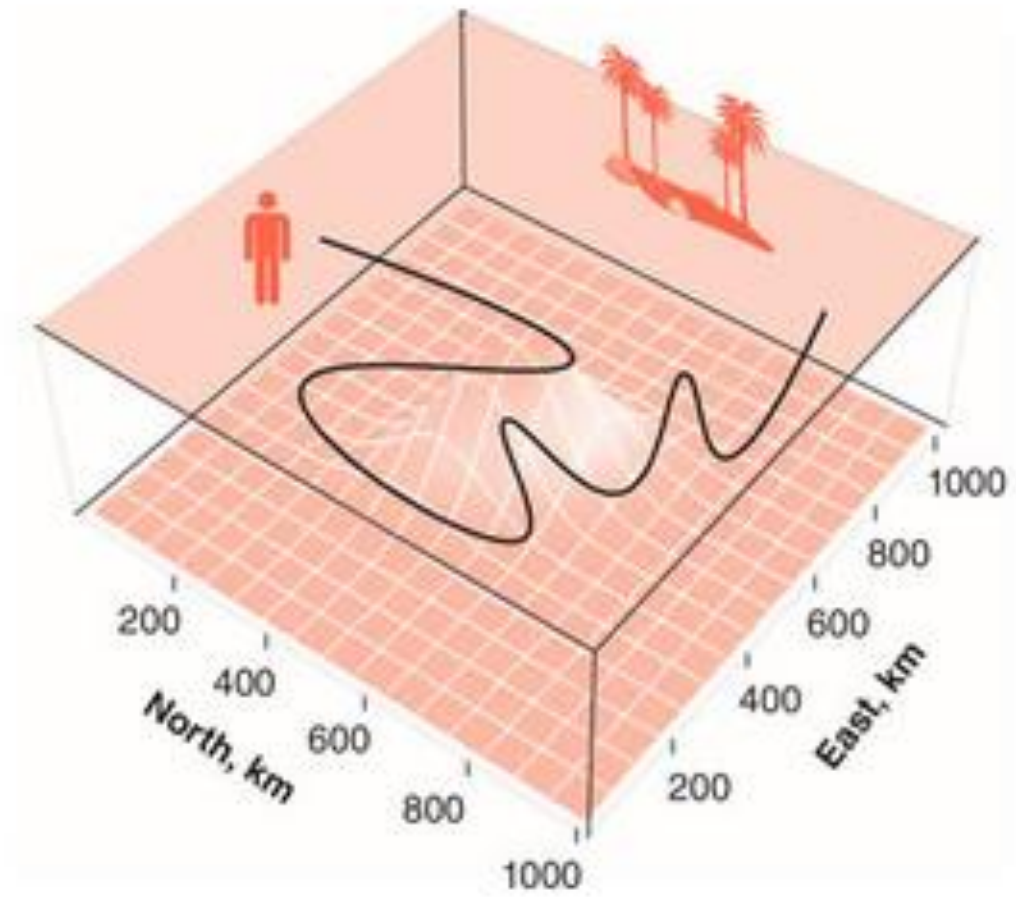
An MCMC Robot

$p(\theta \mid D) \propto p(D \mid \theta) p(\theta)$



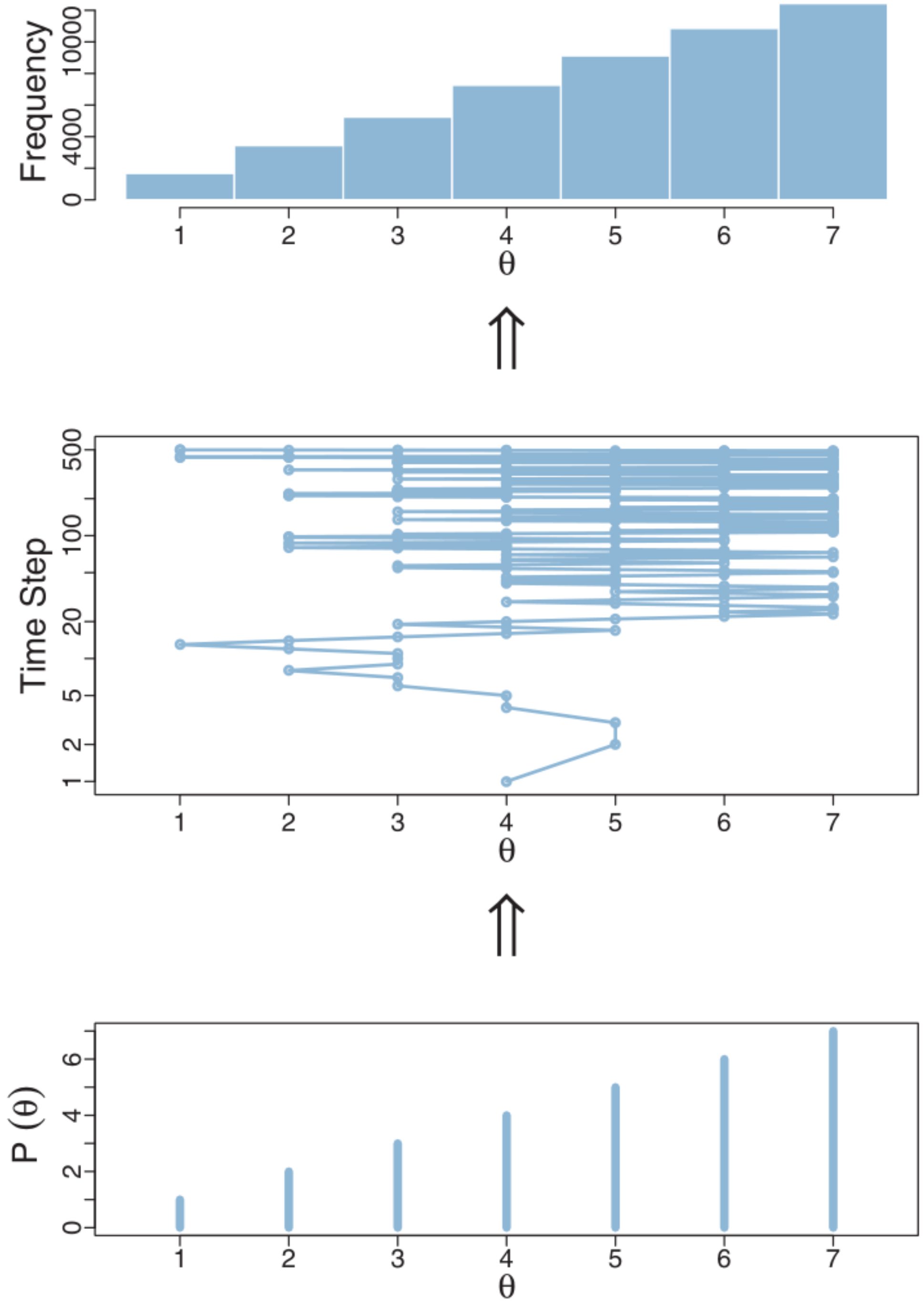
http://marple.eeb.uconn.edu/mcmcrobot/

An MCMC Robert in 3D



Lambert (2018)

Sampling Example: Discrete



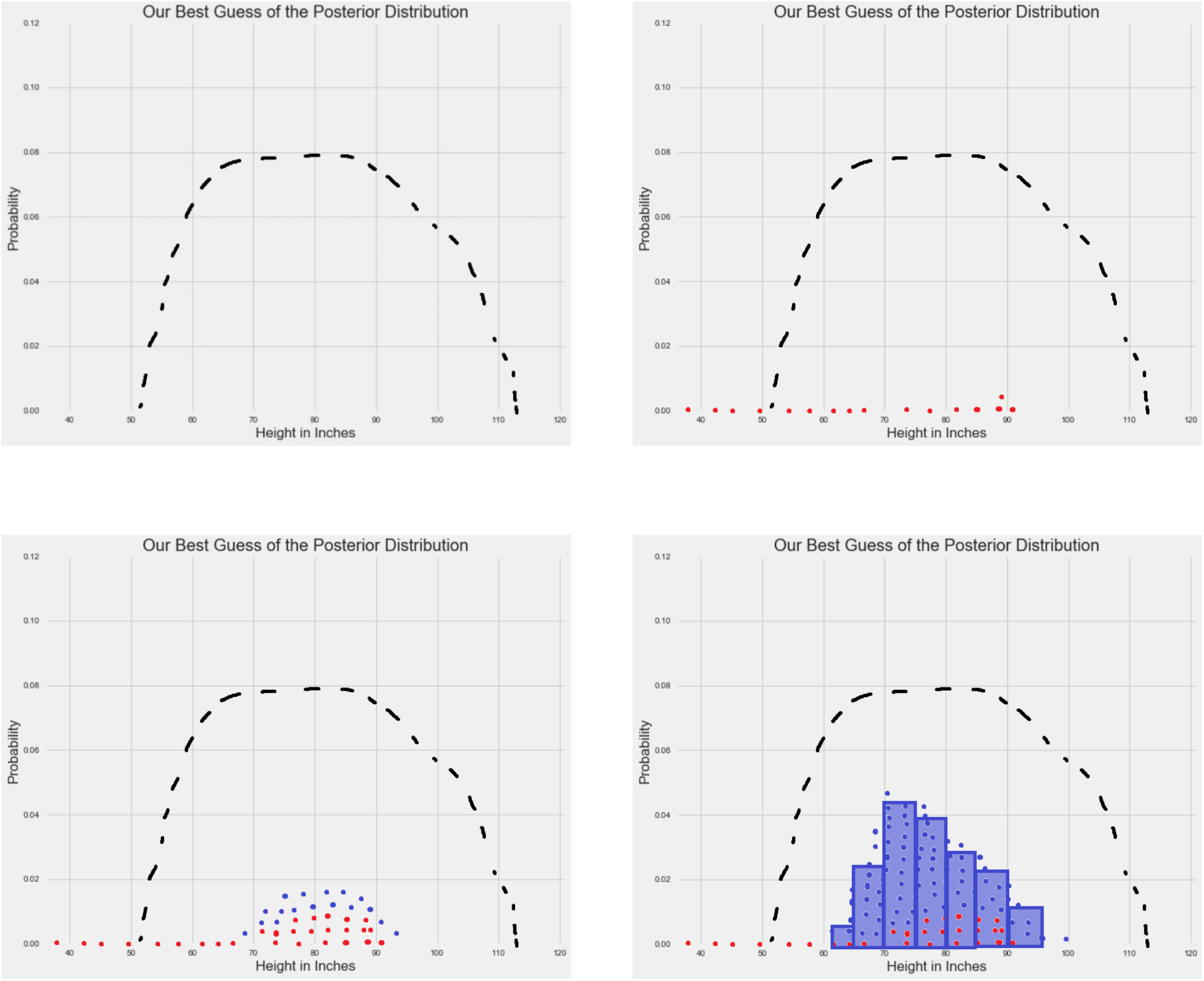
MCMC summary

MCMC trace

True distribution

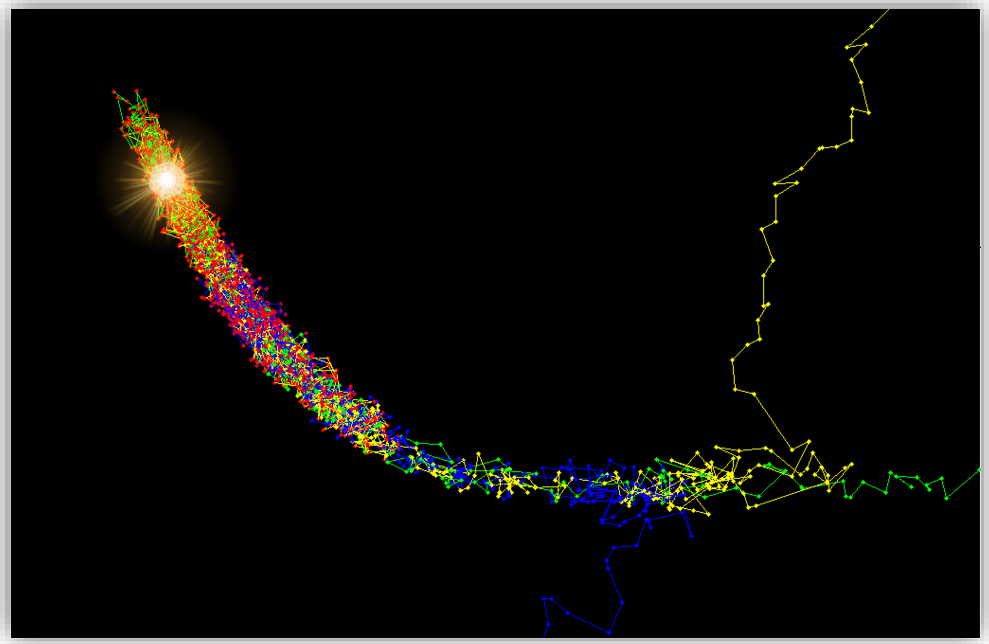
Kruschke (2015)

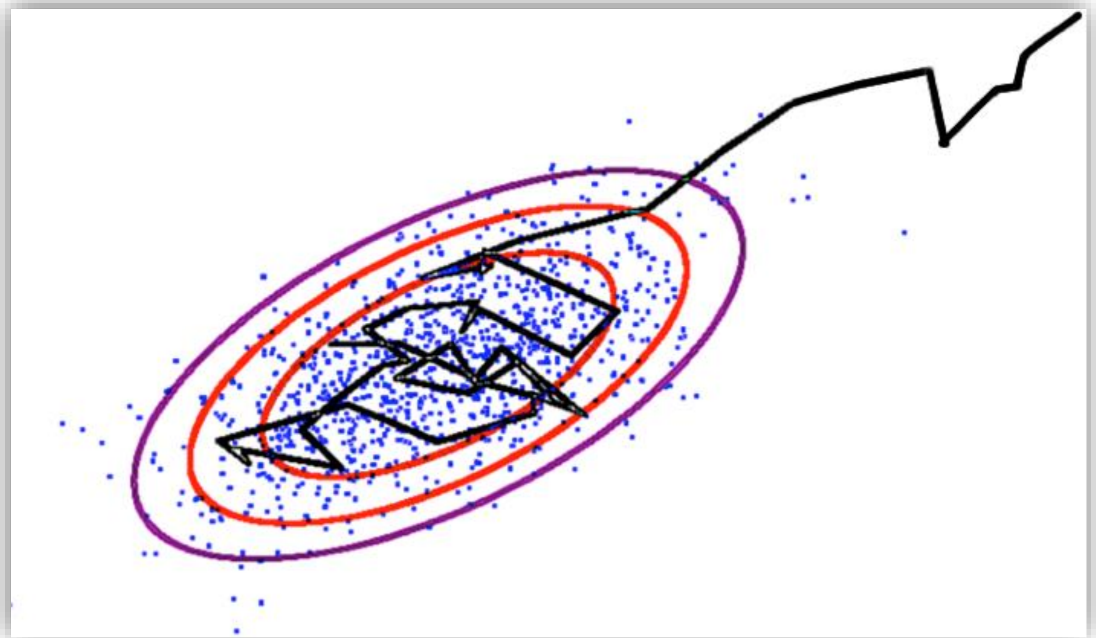
Sampling Example: Continuous



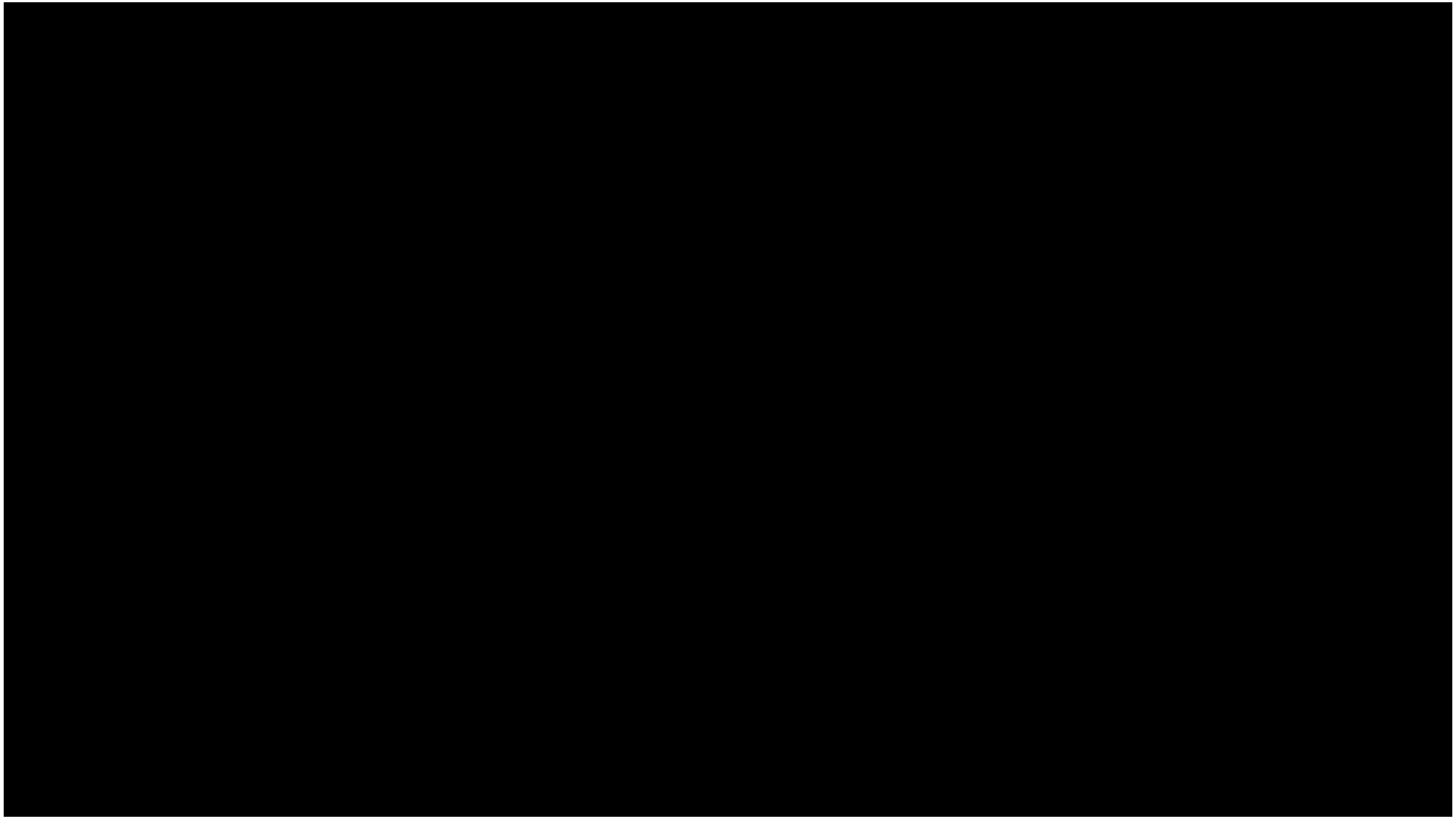
[A Zero-Math Introduction to Markov Chain Monte Carlo Methods]

Visual Example





Let’s watch a video!



Credit: Thomas Wiecki

MCMC Sampling Algorithms

- Rejection sampling
- Importance sampling
- Metropolis algorithm
- Gibbs sampling → JAGS
- HMC sampling*

Stan!

*: also PyMC3 PyMC3

Bayesian Statistics and Hierarchical Bayesian Modeling for Psychological Science

Lecture 06

Lei Zhang

Social, Cognitive and Affective Neuroscience Unit (SCAN-Unit)

Department of Cognition, Emotion, and Methods in Psychology

lei.zhang@univie.ac.at

lei-zhang.net

@lei_zhang_lz

https://github.com/lei-zhang/BayesCog_Wien

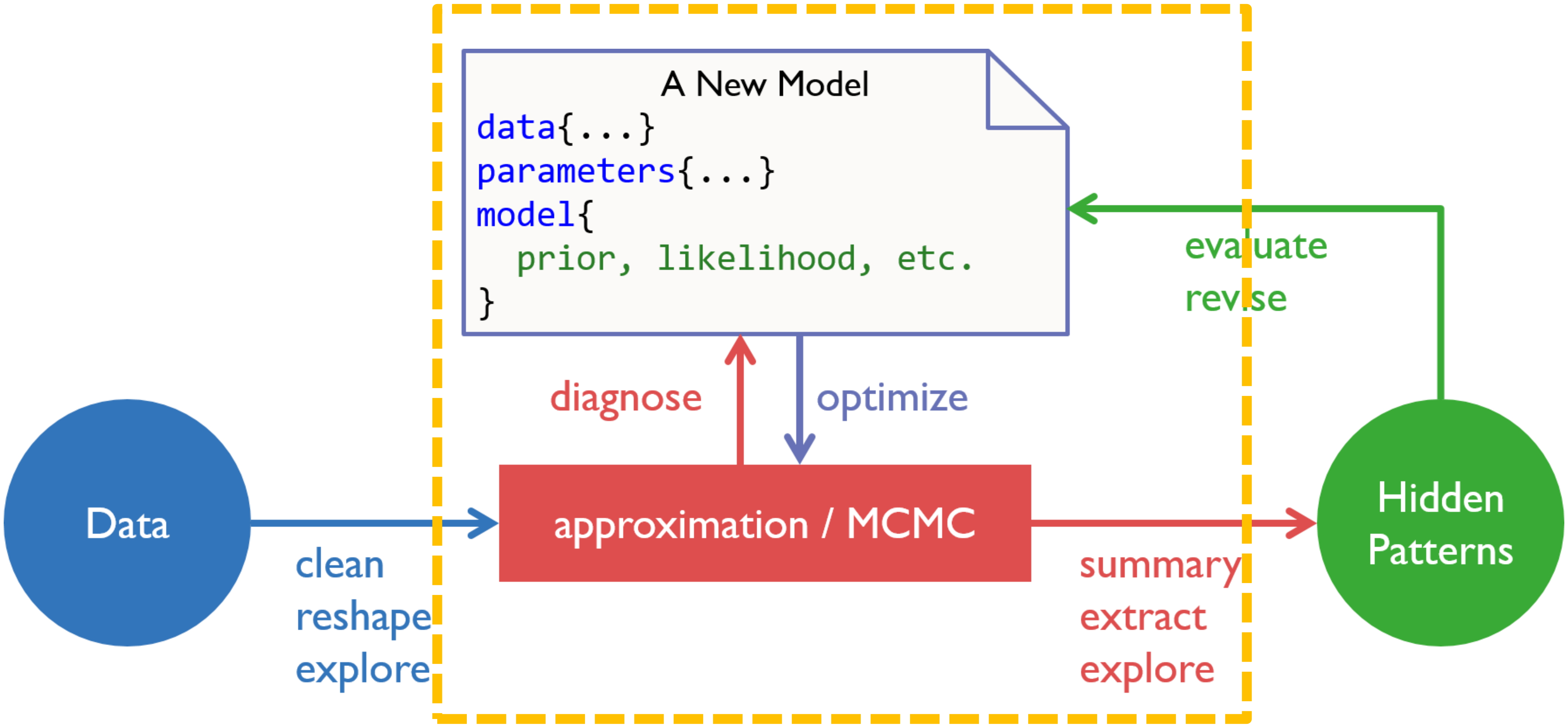
universität wien

Fakultät für Psychologie

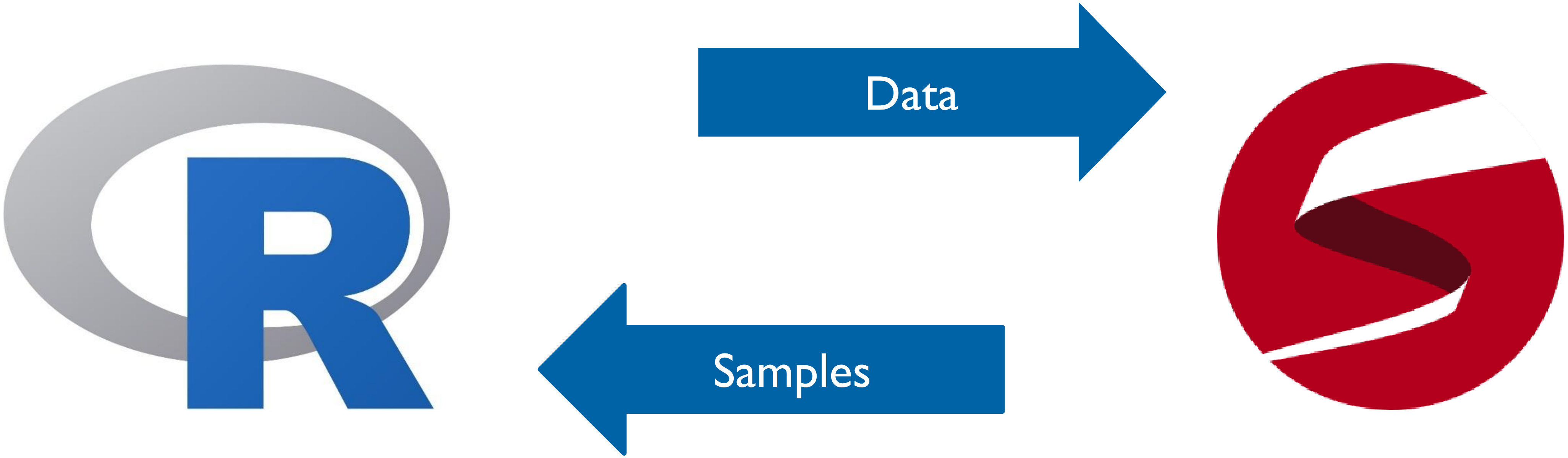
Bayesian **warm-up**?

STAN PROGRAMMING LANGUAGE I

cognitive model
statistics
computing



Stan and RStan



Steps of Bayesian Modeling, with Stan

cognitive model
statistics
computing

A data story	Think about how the data might arise. It can be *descriptive* or even *causal*. Write a Stan program (*.stan).
Update	Educate your model by feeding it the data. Bayesian Update: update the prior, in light of data, to produce posterior. Run Stan using RStan (PyStan, MatlabStan etc.)
Evaluate	Compare model with reality. Revise your model. Evaluate in RStan and ShinyStan.

McElreath (2016)

Steps of Using Stan

1. Stan program read into memory
2. Source-to-source transformation into C++
3. C++ compiled and linked (takes a while)
4. Run Stan program
5. Posterior analysis / interface

```
data {
  int<lower=0> N;
  int<lower=0,upper=1> y[N];
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  y ~ bernoulli(theta);
}
```

Stan Language

model blocks

```
data {
 //... read in external data...
}

transformed data {
 //... pre-processing of data ...
}

parameters {
 //... parameters to be sampled by HMC ...
}

transformed parameters {
 //... pre-processing of parameters ...
}

model {
 //... statistical/cognitive model ...
}

generated quantities {
 //... post-processing of the model ...
}
```

REVISIT BINOMIAL MODEL

Binomial Model

cognitive model

statistics

computing

W L W W W L W L W

$$p(w \mid N, \theta) = \binom{N}{w} \theta^{w} (1-\theta)^{N-w}$$

$$w \sim \text{Binomial}(N, \theta)$$

reads as:
w is distributed as a binomial distribution, with number of trials *N*, and success rate ϑ.

McElreath (2016)

Graphical Model Notations

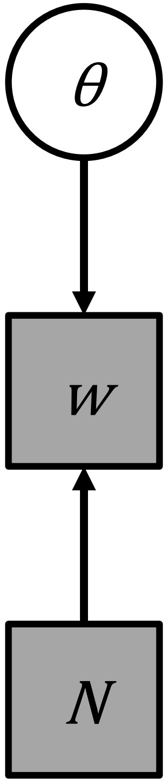
	continuous	discrete
unobserved	θ	δ
observed	y	N

Lee & Wagenmakers (2013)

Binomial Model

W L W W W L W L W

$$p(w \mid N, \theta) = \binom{N}{w} \theta^{w} (1-\theta)^{N-w}$$



$$\theta \sim \mathrm{Uniform}(0, 1)$$

$$w \sim \mathrm{Binomial}(N, \theta)$$

	continuous	discrete
unobserved	θ	δ
observed	y	N

McElreath (2016)

Binomial Model

cognitive model
statistics
computing

W L W W W L W L W

$$p(w \mid N, \theta) = \binom{N}{w} \theta^{w} (1-\theta)^{N-w}$$

```
data {
    int<lower=0> w;
    int<lower=0> N;
}

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

model {
    w ~ binomial(N, theta);
}
```

McElreath (2016)

Running Binomial Model with Stan


```
…/BayesCog/02.binomial_globe/_scripts/binomial_globe_main.R
```

```
> R.version
R version 3.5.1 (2018-07-02)

> stan_version()
[1] "2.18.0"
```

Model Summary

cognitive model
statistics
computing

```
> print(fit_globe)
Inference for Stan model: binomial_globe_model.
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%   25%   50%   75% 97.5% n_eff Rhat
theta  0.64    0.00 0.14  0.35  0.54  0.65  0.74  0.87  1278    1
lp__  -7.72    0.02 0.69 -9.77 -7.89 -7.46 -7.27 -7.21  1824    1

Samples were drawn using NUTS(diag_e) at Tue Apr 09 12:44:04 2019.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Gelman-Rubin convergence diagnostic (Gelman & Rubin, 1992)


ANY
QUESTIONS
?

Happy Computing!