

# Cracking hierarchical Bayesian computational modeling with Stan

**Dr. Lei Zhang**

Associate Professor

Centre for Human Brain Health, School of Psychology  
University of Birmingham



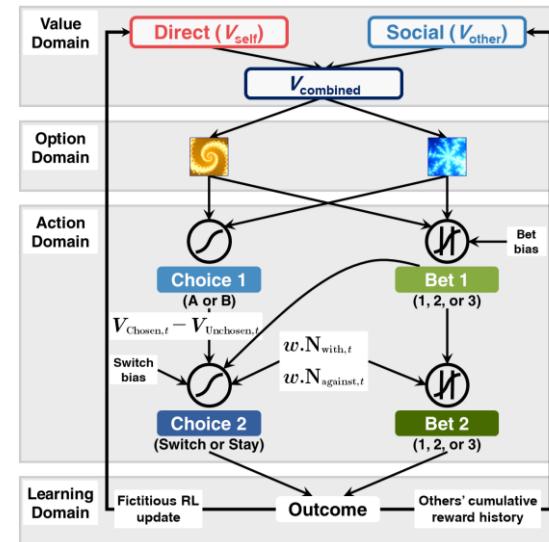
COSMOS 2023, Konstanz  
<https://shorturl.at/mrEGI>

lei-zhang.net  
@lei\_zhang\_lz



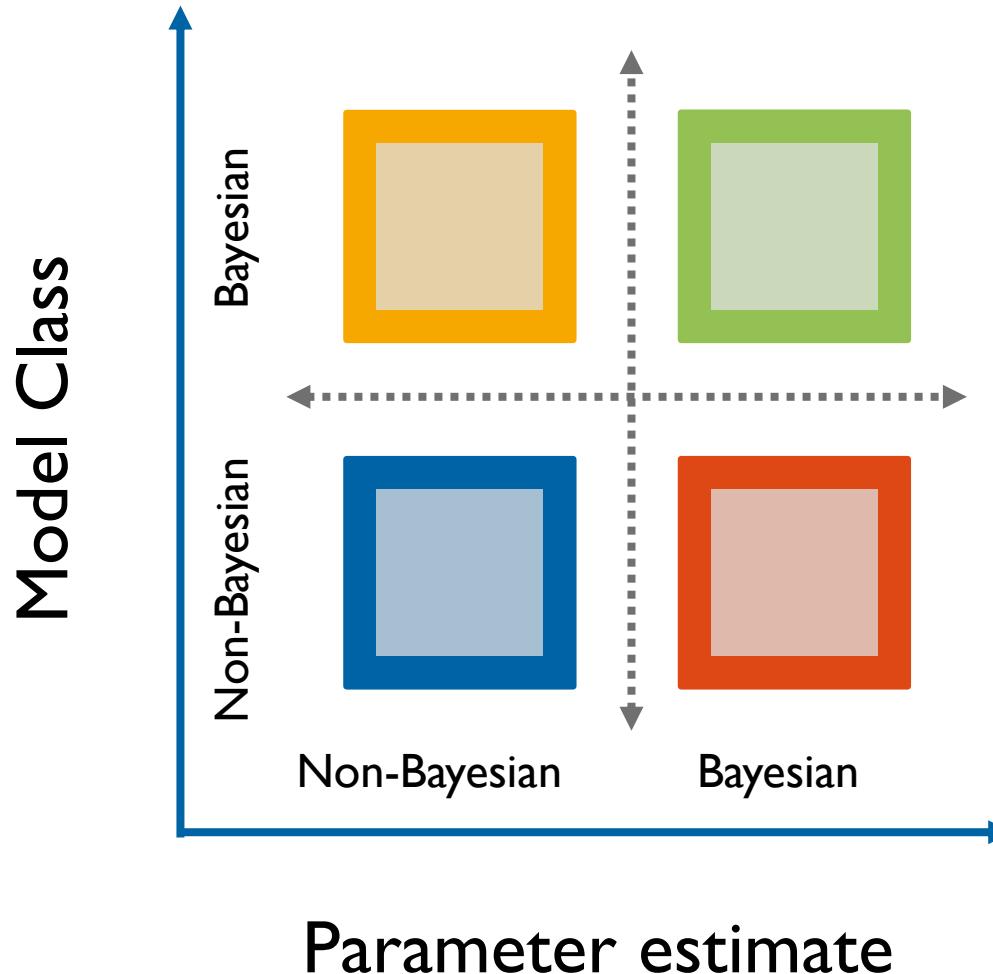
# About me

- My journey through computational modeling
  - Started with MLE (@fminsearch in Matlab)
  - Switched to Bayesian: first JAGS, then Stan
  - Why switching to Stan? Look at Fig. 2A and Table S3 in [Zhang & Gläscher \(2020\)](#)



$$\begin{aligned}
 V_{self,t} &= [V_{self,t}(A), V_{self,t}(B)] \\
 V_{other,t} &= [V_{other,t}(A), V_{other,t}(B)] \\
 V_t &= \beta_{self} V_{self,t} + \beta_{other} V_{other,t} \\
 C1_t &\sim \text{Categorical}(\text{Softmax}(V_t)) \\
 U_{bet1,t} &= \beta_{bias} + \beta_{val}(V_{chosen,C1,t} - V_{unchosen,C1,t}) \\
 B1_t &\sim \text{OrderedLogistic}(U_{bet1,t} | \theta) \\
 &= \sum_{j=1}^K w_{j,t}, K = 0, 1, \dots, 4 \\
 w.N_{against,t} &= \sum_{j=1}^{K-1} w_{j,t} \\
 w.N_{with,t} &= \sum_{j=1}^K w_{j,t} \\
 V_t(\text{switch}) &= \beta_{bias} + \beta_{val}(V_{chosen,C1,t} - V_{unchosen,C1,t}) + \beta_{against} w.N_{against,t} \\
 C2 &\sim \text{Bernoulli}(V_t(\text{switch})) \\
 U_{bet2,t} &= U_{bet1,t} + \beta_{with} w.N_{with} + \beta_{against} w.N_{against}, \text{ if } C1 = C2 \\
 B2_t &\sim \text{OrderedLogistic}(U_{bet2,t} | \theta) \\
 \Phi(x) &= \frac{1}{1+e^{-x}} \\
 \delta_{self,chosen,C1,t} &= R_{self,t} - V_{self,chosen,C1,t} \\
 \delta_{self,unchosen,C1,t} &= -R_{self,t} - V_{self,unchosen,C1,t} \\
 V_{self,chosen,C1,t+1} &= V_{self,chosen,C1,t} + \alpha \delta_{self,chosen,C1,t} \\
 V_{self,unchosen,C1,t+1} &= V_{self,unchosen,C1,t} + \alpha \delta_{self,unchosen,C1,t}
 \end{aligned}$$

# What We Talk About When We Talk About “Bayesian” Models



# Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

# Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

# Recap on Bayesian modeling

## Likelihood

How plausible is the data given our parameter is true?

## Prior

How plausible is our parameter before observing the data?

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

## Posterior

How plausible is our parameter given the observed data?

## Evidence

How plausible is the data under all possible parameters?

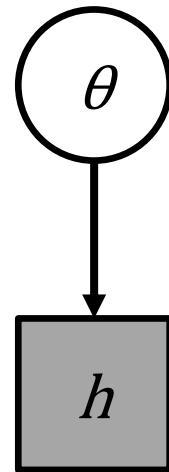
# Bernoulli Model

- You are interested in if a coin is biased.
- You will flip the coin.
- You will record whether it comes up a head (h) or a tail (t).
- You might observe 15 heads out of 20 flips.
- What is your degree of belief about the biased parameter  $\vartheta$ ?



# Bernoulli Model

$$p(h | \theta) = \theta^h (1 - \theta)^{1-h}$$

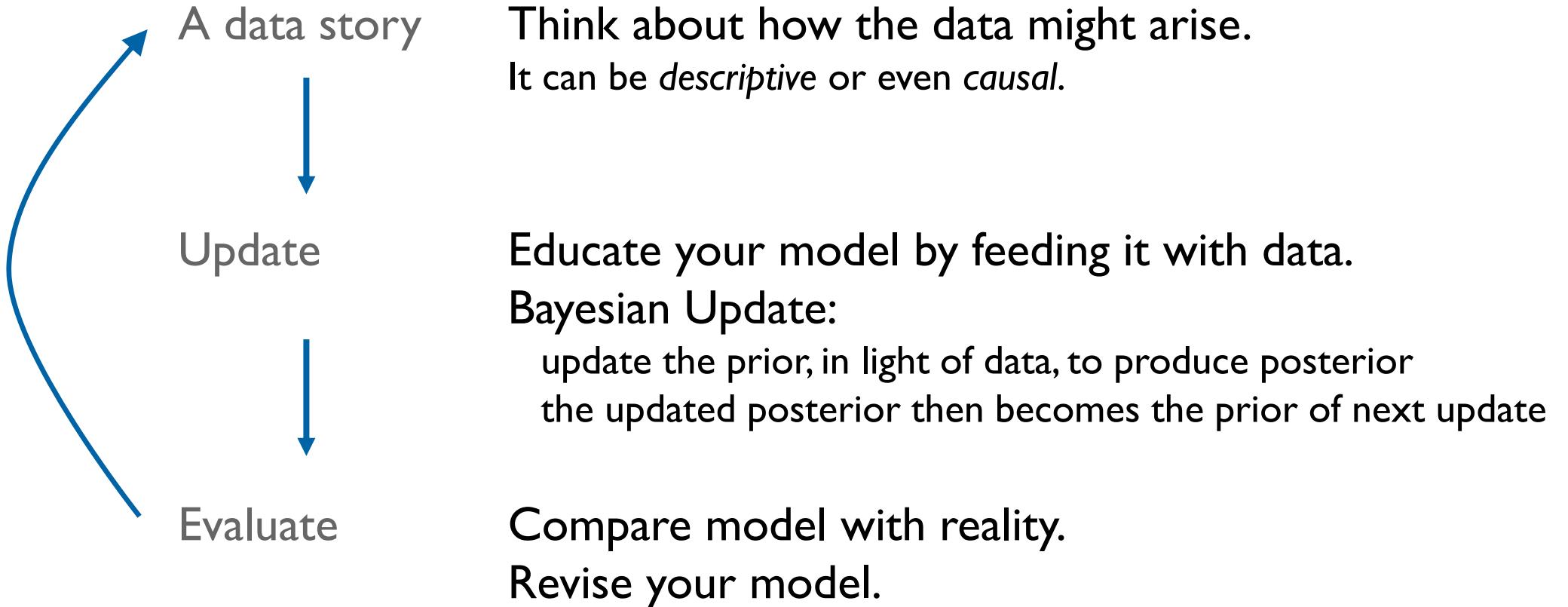


$\theta \sim \text{Uniform}(0, 1)$

$h \sim \text{Bernoulli}(\theta)$

|            | continuous | discrete |
|------------|------------|----------|
| unobserved | $\theta$   | $\delta$ |
| observed   | $y$        | $N$      |

# Steps of (Bayesian) Modeling?



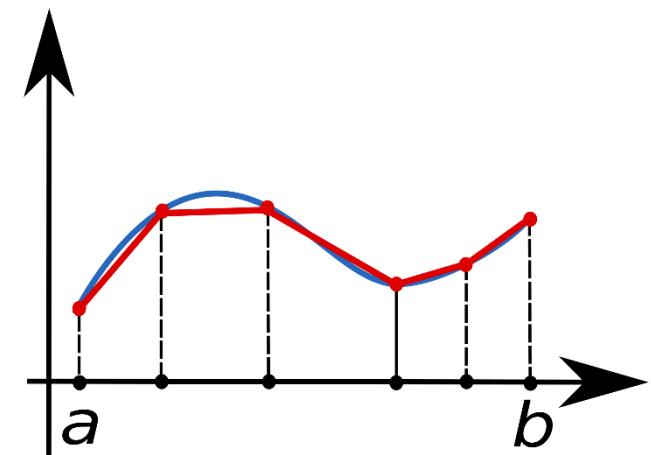
# Solve it by Grid Approximation

discrete parameters

$$p(\theta | D) = \frac{p(D | \theta)p(\theta)}{\sum_{\theta^*} p(D | \theta^*)p(\theta^*)}$$

continuous parameters

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



# What if I have multiple parameters?

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

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

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

$$p(data) = \int_{\mu_1} \int_{\sigma_1} \dots \int_{\mu_{100}} \int_{\sigma_{100}} \underbrace{p(data | \mu_1, \sigma_1, \dots, \mu_{100}, \sigma_{100})}_{\text{likelihood}} \times \underbrace{p(\mu_1, \sigma_1, \dots, \mu_{100}, \sigma_{100})}_{\text{prior}} \\ d\mu_1 d\sigma_1 \dots d\mu_{100} d\sigma_{100},$$

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

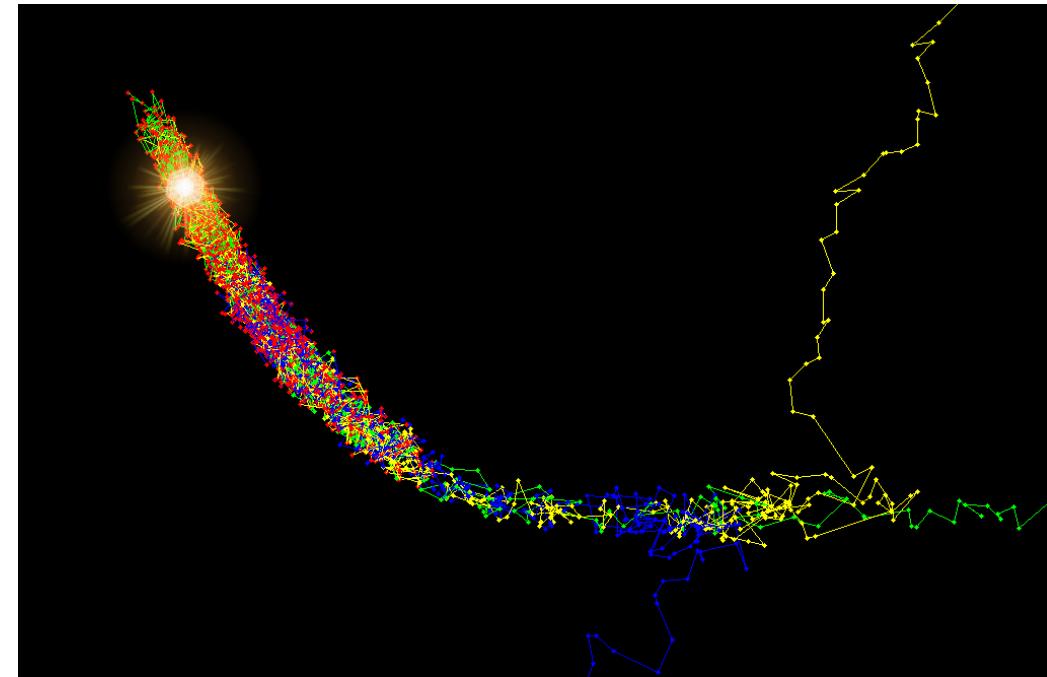
$$p(\theta | D) \propto p(D | \theta) p(\theta)$$

# MCMC Sampling Algorithms

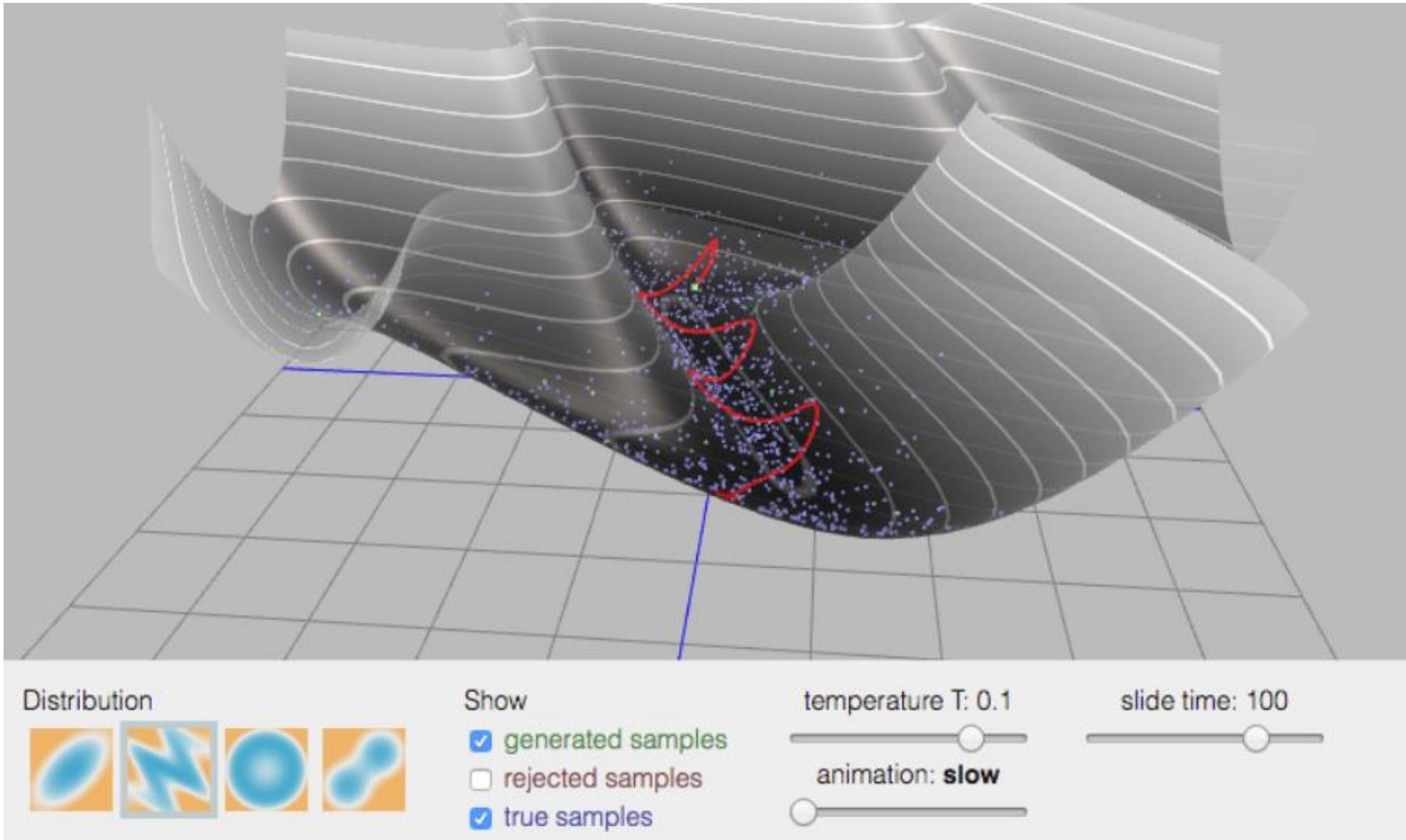
- Rejection sampling
- Importance sampling
- Metropolis algorithm
- Gibbs sampling → JAGS
- HMC sampling\*



Stan!



# build some intuition



# Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

# Is Stan popular?

2021 Summer

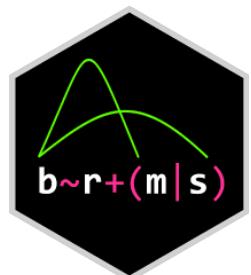
## Stan: A Probabilistic Programming Language

Carpenter, Bob; Gelman, Andrew; Hoffman, Matthew D.; Lee, Daniel; Goodrich, Ben; Betancourt, Michael; Brubaker, Marcus A.; Guo, Jiqiang; Li, Peter; Riddell, Allen

Grantee Submission, Journal of Statistical Software v76 n1 p1-32 Jan 2017

Stan is a probabilistic programming language for specifying statistical models. A Stan program imperatively defines a log probability function over parameters conditioned on specified data and constants. As of version 2.14.0, Stan provides full Bayesian inference for continuous-variable models through Markov chain Monte Carlo methods such as the No-U-Turn sampler, an adaptive form of Hamiltonian Monte Carlo sampling. Penalized maximum likelihood estimates are calculated using optimization methods such as the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm. Stan is also a platform for computing log densities and their gradients and Hessians, which can be used in alternative algorithms such as variational Bayes, expectation propagation, and marginal inference using approximate integration. To this end, Stan is set up so that the densities, gradients, and Hessians, along with intermediate quantities of the algorithm such as acceptance probabilities, are easily accessible. Stan can be called from the command line using the "cmdstan" package, through R using the "rstan" package, and through Python using the "pystan" package. All three interfaces support sampling and optimization-based inference with diagnostics and posterior analysis. "rstan" and "pystan" also provide access to log probabilities, gradients, Hessians, parameter transforms, and specialized plotting.

Descriptors: [Programming Languages](#), [Probability](#), [Bayesian Statistics](#), [Monte Carlo Methods](#), [Statistical Inference](#), [Maximum Likelihood Statistics](#), [Computation](#), [Statistical Distributions](#), [Computer Software](#)



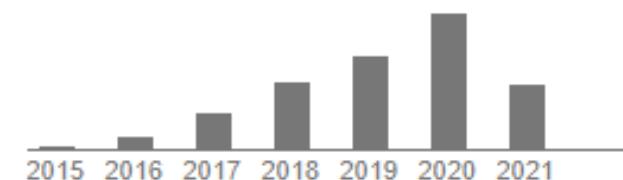
## brms: An R package for Bayesian multilevel **models** using Stan

[PC Bürkner - Journal of statistical software, 2017 - jstatsoft.org](#)

... The **brms** package does not fit **models** itself but uses Stan on the back-end. Accordingly, all samplers implemented in Stan can be used to fit **brms** **models**. Currently, these are the static ...

[☆ Save](#) [99 Cite](#) [Cited by 5534](#) [Related articles](#) [All 29 versions](#) [»»](#)

Cited by 3989



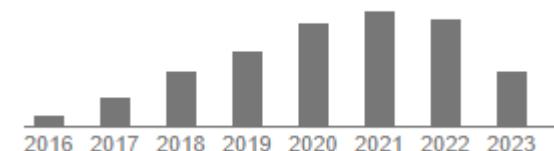
[PDF] **JAGS**: A program for analysis of Bayesian graphical models using Gibbs sampling

[M Plummer - Proceedings of the 3rd international workshop on ..., 2003 - ci.tuwien.ac.at](#)

JAGS is a program for Bayesian Graphical modelling which aims for compatibility with Classic BUGS. The program could eventually be developed as an R package. This article explains the motivations for this program, briefly describes the architecture and then ...

[☆ 99 Cited by 4543](#) [Related articles](#) [All 8 versions](#) [»»](#)

Cited by 6594



2023 Summer

[PDF] **JAGS**: A program for analysis of Bayesian graphical models using Gibbs sampling

[M Plummer - Proceedings of the 3rd international workshop on ..., 2003 - r-project.org](#)

... JAGS is a program for Bayesian Graphical modelling which aims for compatibility ... JAGS and so avoid having to write a new program for each application. A second motivation for JAGS ...

[☆ Save](#) [99 Cite](#) [Cited by 6277](#) [Related articles](#) [All 7 versions](#) [»»](#)

# Who are using Stan?



# Why Use Stan?

vs. BUGS / JAGS

- Less spatial correlation → effective samples
- Time to converge and per effective sample size:  
1 - ∞ times faster
- Memory usage: 1–10%
- Language features
  - variable overwrite: `a = 4`, then `a = 5`
  - formal control flow (same as R)
  - full support of vectorizing & matrix calculation



Krzysztof Sakrejda  
@sakrejda

I keep getting asked why people should use [@mcmc\\_stan](#) so I wrote an answer:

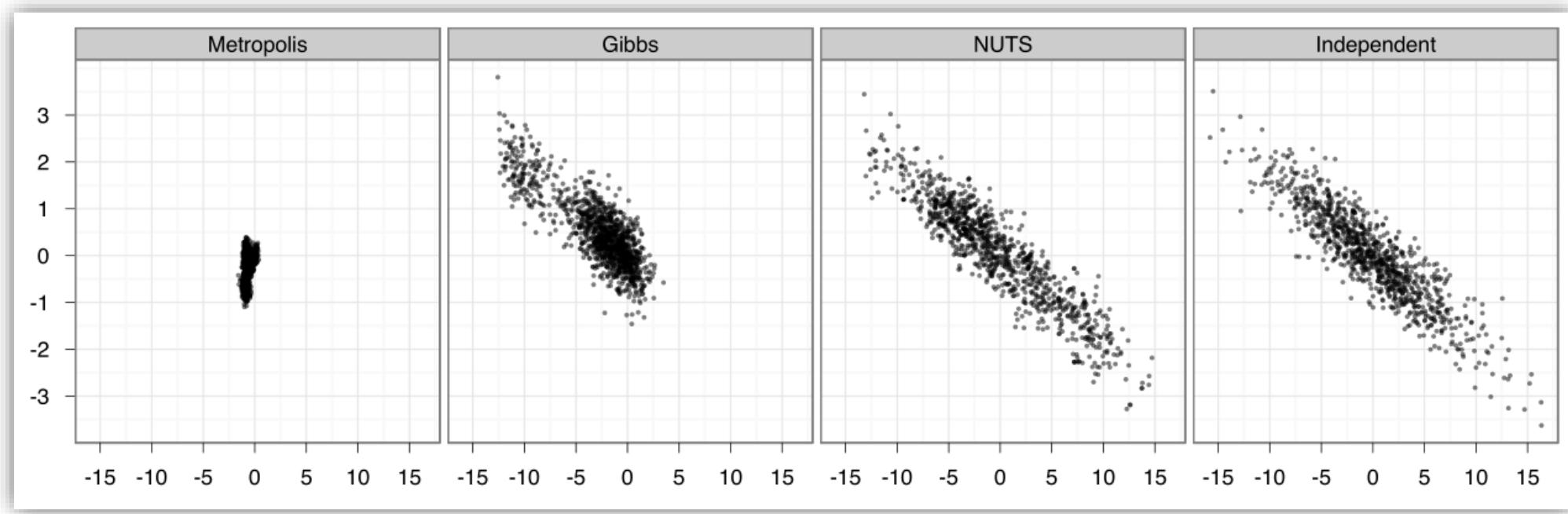


"Selling" Stan  
[discourse.mc-stan.org](http://discourse.mc-stan.org)

27.03.18, 16:01

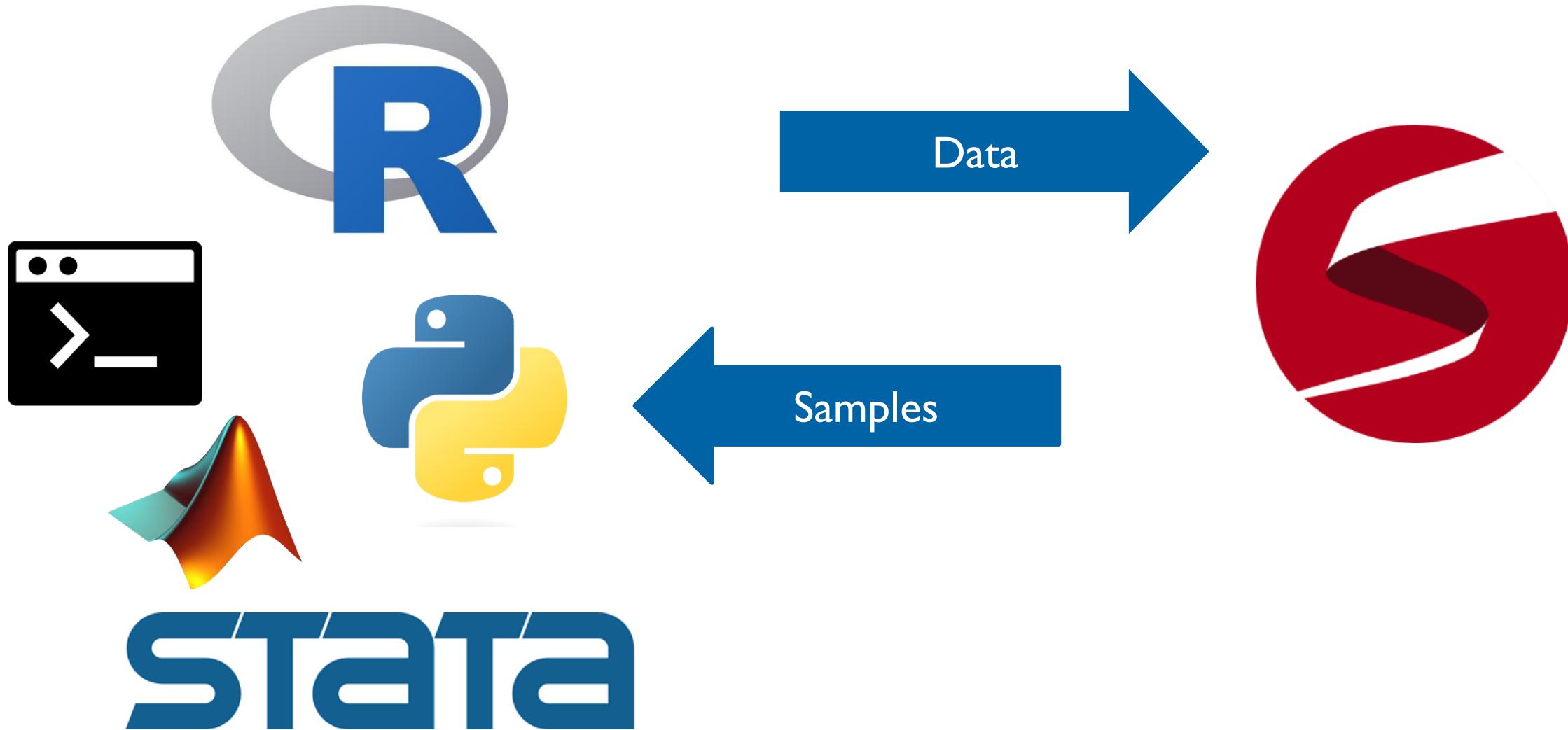
# HMC vs. Gibbs and Metropolis

Hamilton MC (HMC) implements No-U-Turn Sampler (NUTS)



- Two dimensions of highly correlated 250-dim normal
- 1,000,000 draws from Metropolis and Gibbs (thin to 1000)
- 1,000 draws from NUTS; 1000 independent draws

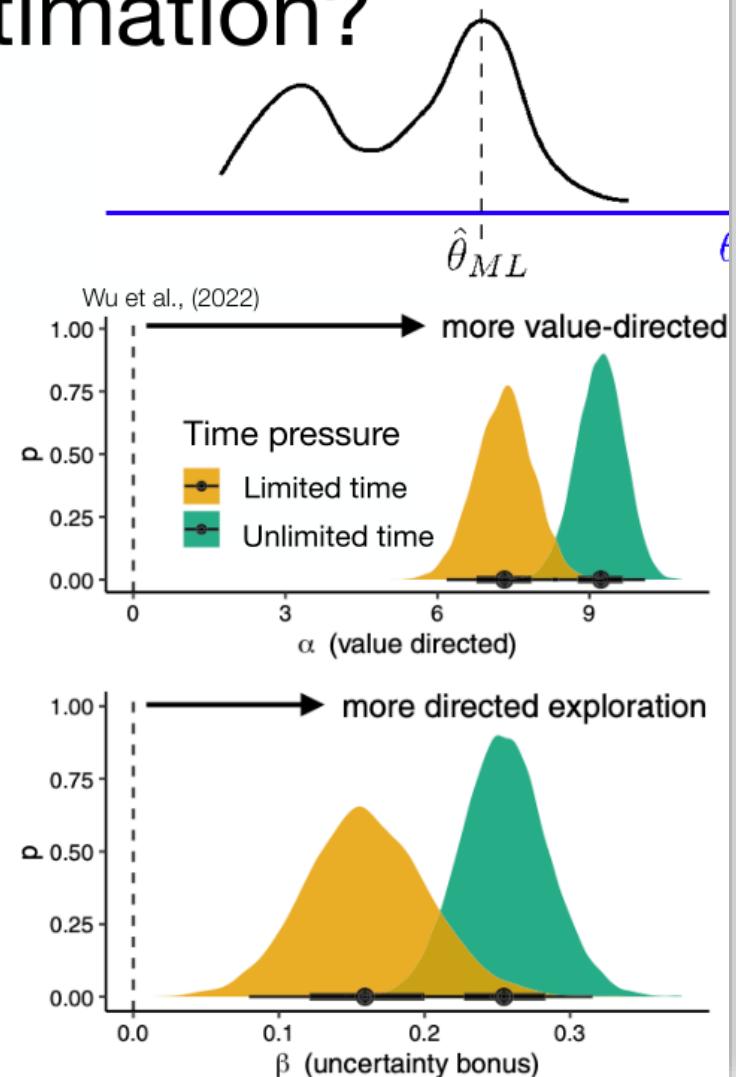
# Stan and Other Platforms



# From tutorial 3

## Why Bayesian model estimation?

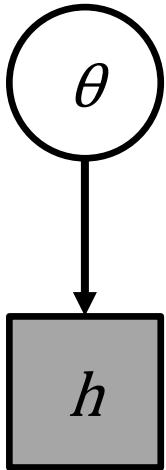
1. Not just a point estimate, but an entire **probability distribution over parameters**
2. Rather than only assuming participants are independent samples, we can model **hierarchical relationships**
3. Naturally avoid overfitting through **Bayesian Occam's Razor**, since we evaluate the model across the entire range of parameters



## But, any cons?

- HMC provides huge improvements in computational efficiency, but **mathematical foundations** are more difficult to follow, at least sometimes.
- Stan cannot sample from the posterior distribution of **discrete parameters** (e.g., [1, 2]).
  - → with additional effort, it can be achieved through marginalisation, see the Stan User Manual.
- There are a few unwritten **tips and tricks**.
  - Practise makes perfect and follow discussions on the forum.
- **Block-based language** → might seem rigid in the first place.

# A quick look at Stan



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

$h \sim \text{Bernoulli}(\theta)$

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

# Stan Language

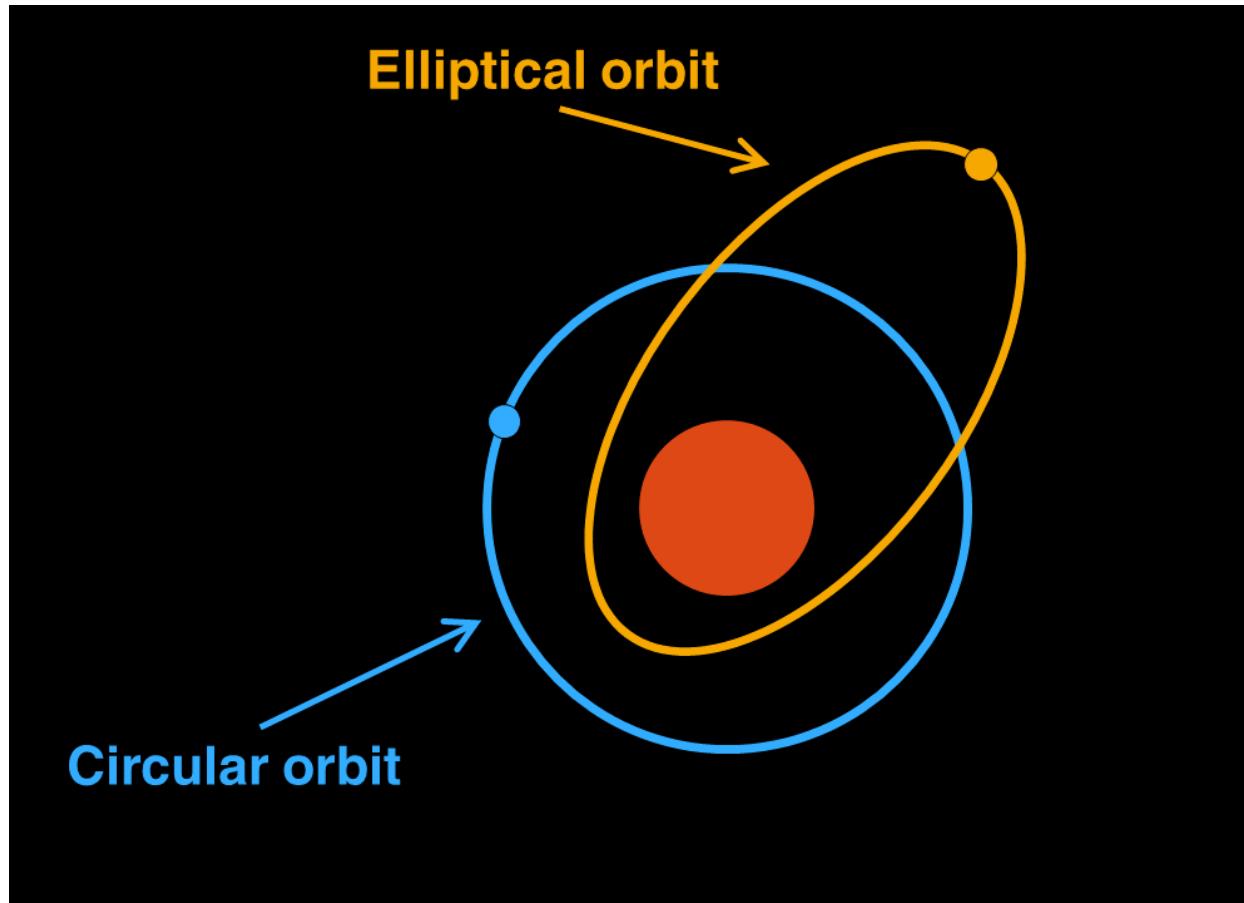
```
data {  
    //... read in external data...  
}  
  
transformed data {  
    //... pre-processing of data ...  
}  
  
parameters {  
    //... parameters to be sampled by HMC ...  
}  
  
transformed parameters {  
    //... pre-processing of parameters ...  
}  
  
model {  
    //... statistical/cognitive model ...  
}  
  
generated quantities {  
    //... post-processing of the model ...  
}
```

# Outline

- Recap on Bayesian modeling
- Why Stan?
- **What is (computational) cognitive modeling?**
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

# The idea of computational modeling is never new

Scientists use mathematical models to approximate certain processes (physical or mental), in order to explain and to predict.



# Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- **The idea of the simple Q-Learning model**
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

# Simple reinforcement learning: 2-armed bandit task



a simple task often used in the laboratory:

- **repeated choice** between N options (**N-armed bandit**)
- ...whose properties (reward amounts, probabilities) are learned through **trial-and-error**
- ...with a **goal** in mind: maximize the overall reward

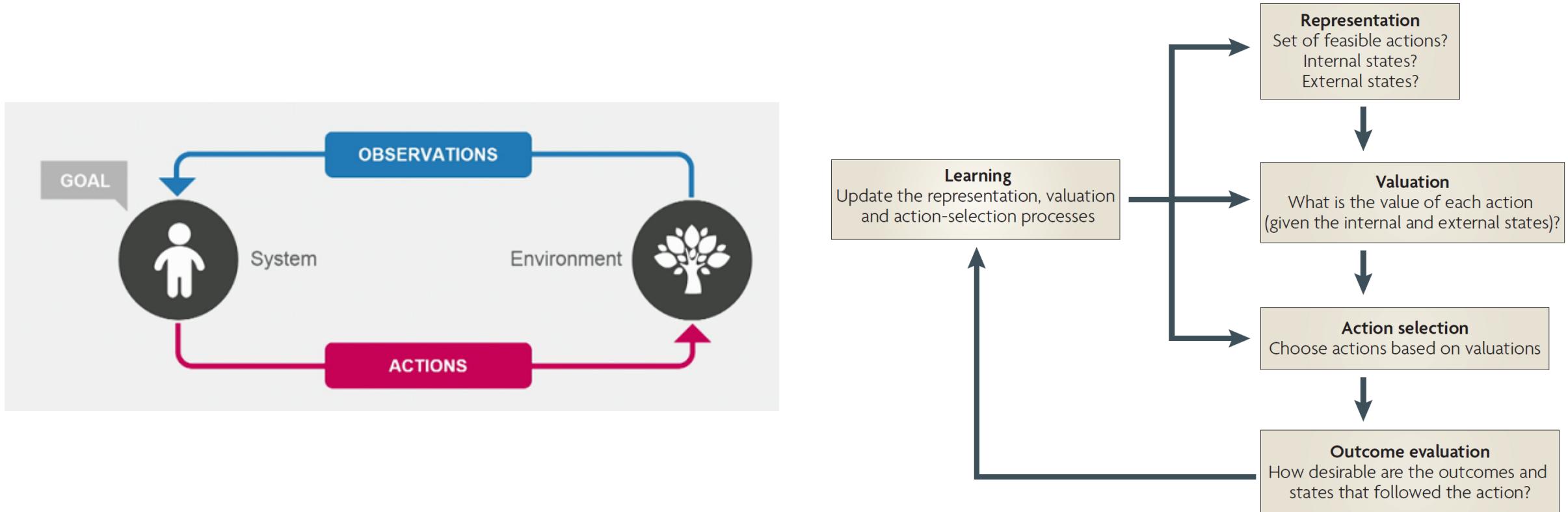
# 2-armed bandit task



What can be your **strategies**:

1. **predict** the value of each deck
2. **choose** the best
3. **learn** from outcome to update predictions  
**(repeat)**

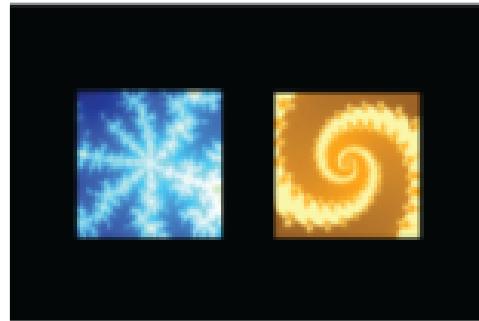
# How is prediction shaped by learning?



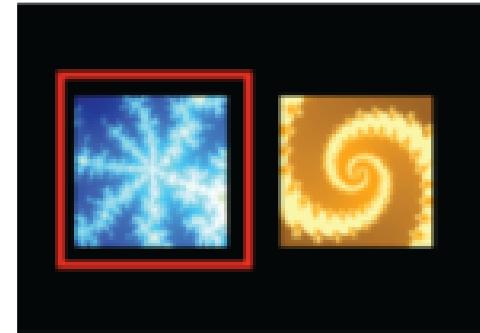
reward contingency 80:20



# One simple experiment: two choice task



choice presentation



action selection



outcome

what do we know?

what can we measure?

what do we not know?

choice & outcome

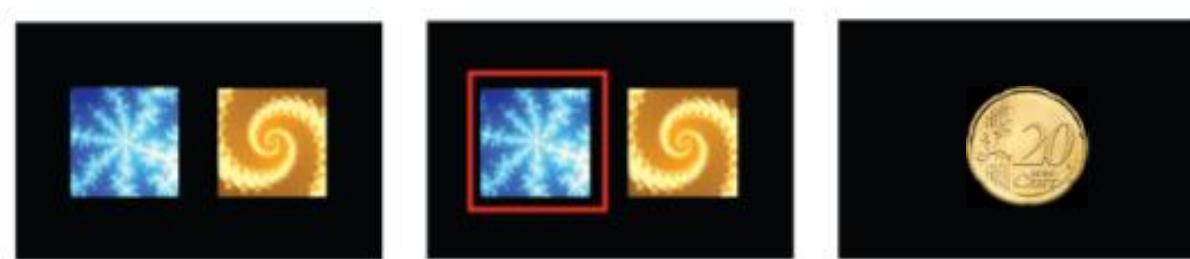
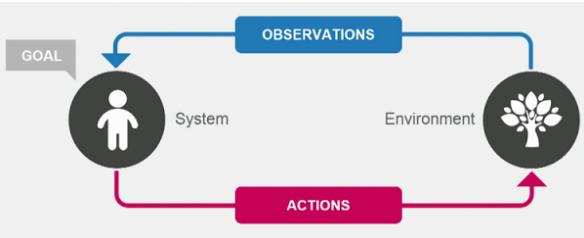
choice accuracy

RL update

$p(\text{choosing the better option})$

| subjID | trialID | choice | outcome |   |
|--------|---------|--------|---------|---|
| 1      | 1       | 1      | 1       | 1 |
| 2      | 1       | 2      | 1       | 1 |
| 3      | 1       | 3      | 1       | 1 |

# Q-learning Value Update



## Cognitive Model

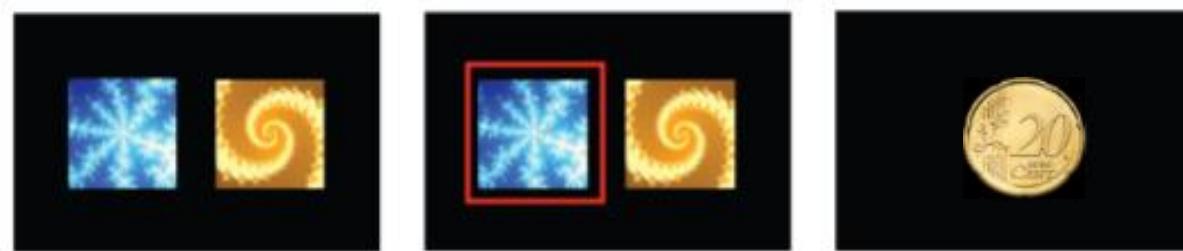
- cognitive process
- using internal variables and free parameters

## Observation Model (Data Model)

- relate model to observed data
- has to account for noise

# Q-Learning

- The idea: **error-driven** learning
- Change in value is proportional to the difference between actual and predicted outcome



Value update:  $V_t = V_{t-1} + \alpha * PE_{t-1}$

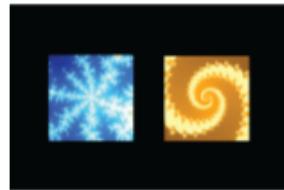
Prediction error:  $PE_{t-1} = R_{t-1} - V_{t-1}$

*Expectations on the next trial = the expectation on the current trial + learning rate \* prediction error (reward – current expectation)*

# Understand the learning rate

Value update:  $V_t = V_{t-1} + \alpha * PE_{t-1}$

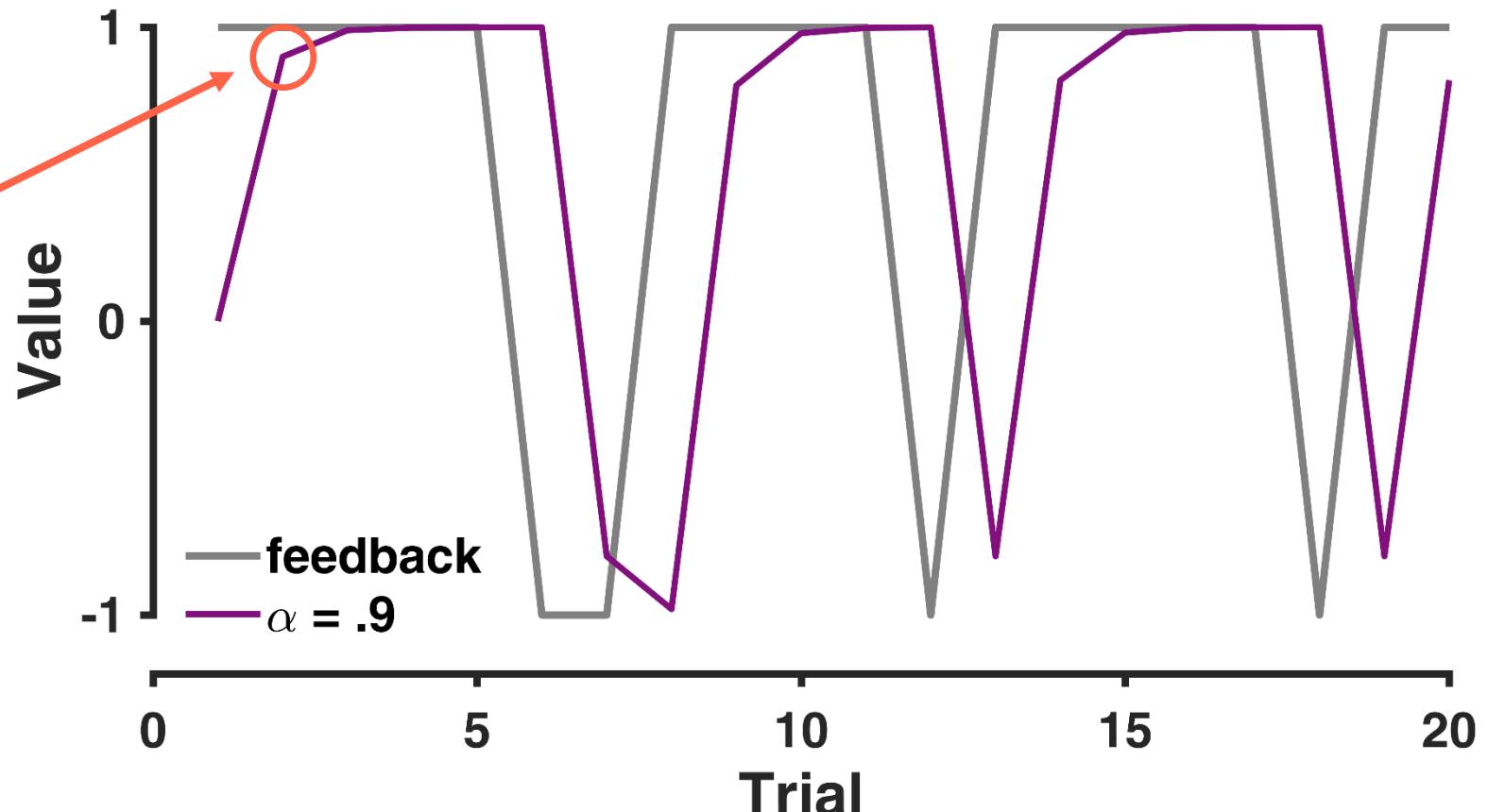
Prediction error:  $PE_{t-1} = R_{t-1} - V_{t-1}$



if  $\alpha = 0.9$

$$V_1 = 0$$

$$\begin{aligned} V_2 &= V_1 + 0.9 * (1 - V_1) \\ &= 0 + 0.9 * (1 - 0) \\ &= 0.9 \end{aligned}$$

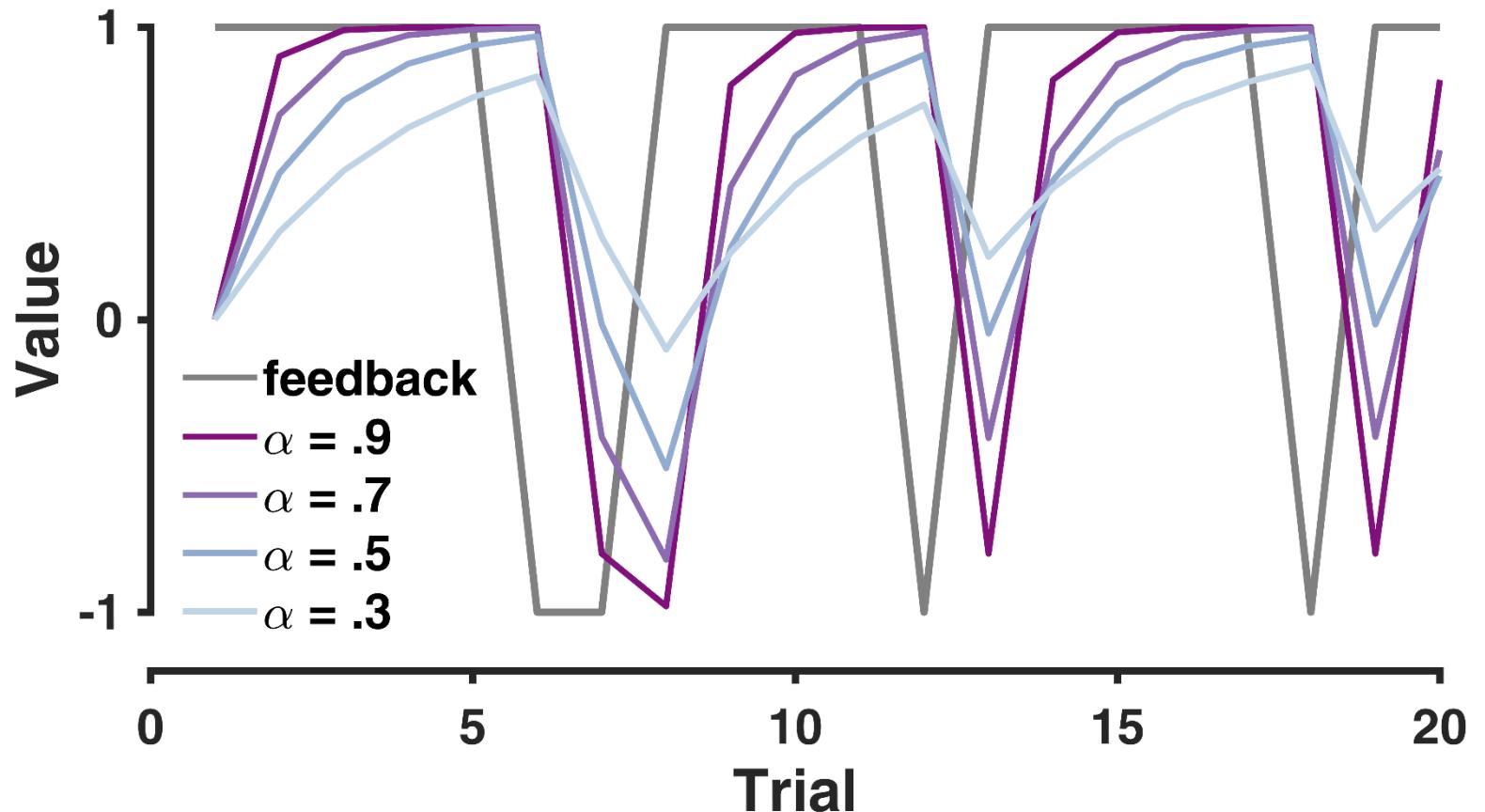


reward contingency – 80:20

# Understand the learning rate

Value update:  $V_t = V_{t-1} + \alpha * PE_{t-1}$

Prediction error:  $PE_{t-1} = R_{t-1} - V_{t-1}$



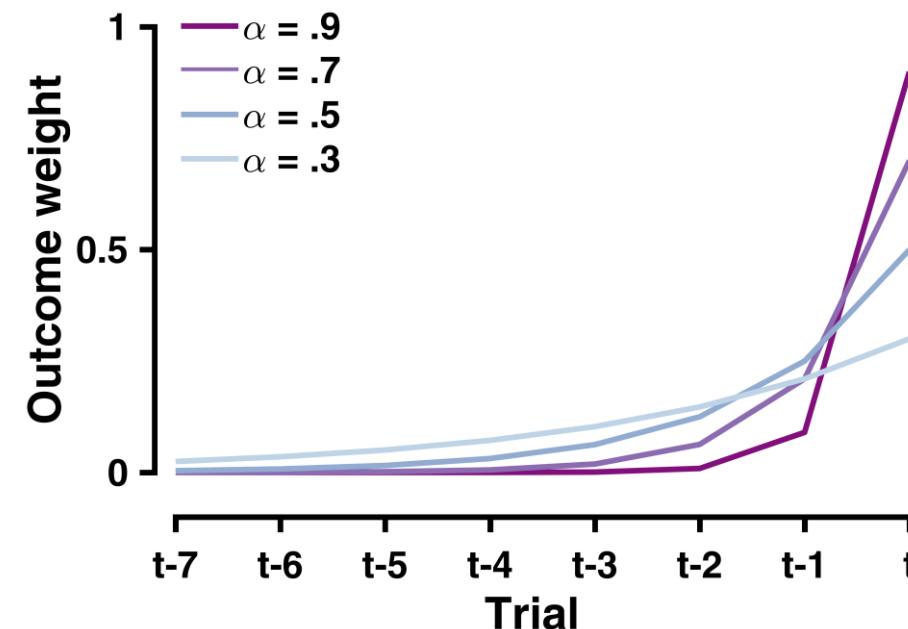
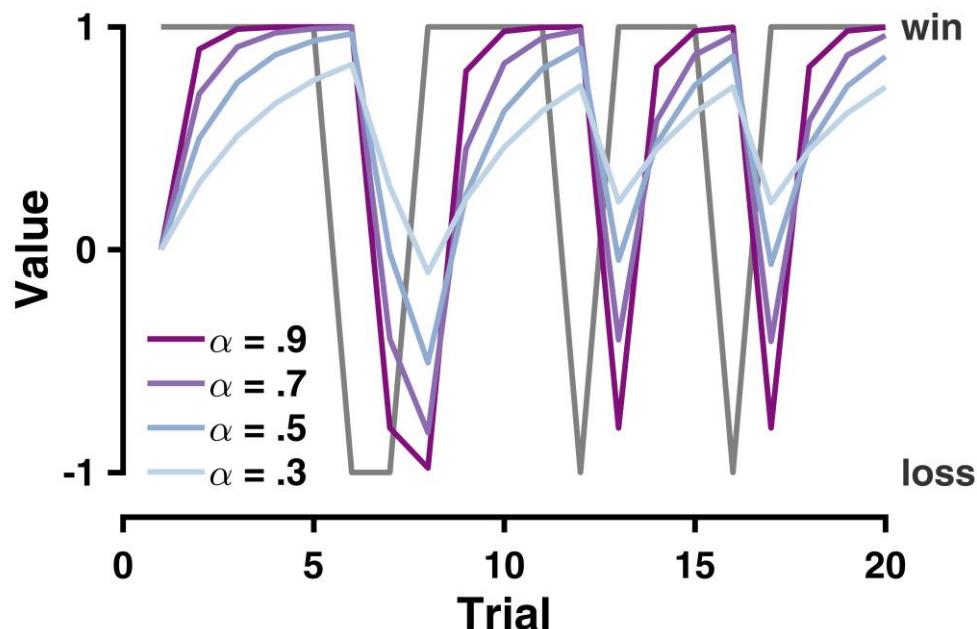
reward contingency – 80:20

# Understand the learning rate

Value update:  $V_t = V_{t-1} + \alpha * PE_{t-1}$

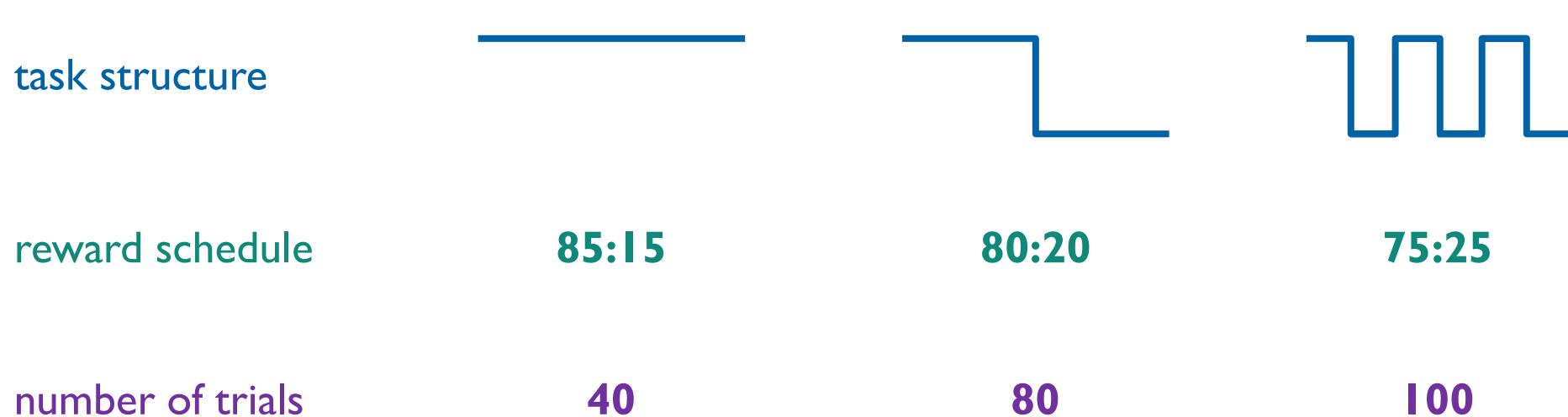
Prediction error:  $PE_{t-1} = R_{t-1} - V_{t-1}$

$$\begin{aligned}V_t &= (1 - \alpha) V_{t-1} + \alpha R_{t-1} \\&= (1 - \alpha)(V_{t-2} + \alpha(R_{t-2} - V_{t-2})) + \alpha R_{t-1} \\&= (1 - \alpha)^t V_0 + \sum_{i=1}^t (1 - \alpha)^{t-i} \alpha R_i\end{aligned}$$



# Is there an optimal learning rate?

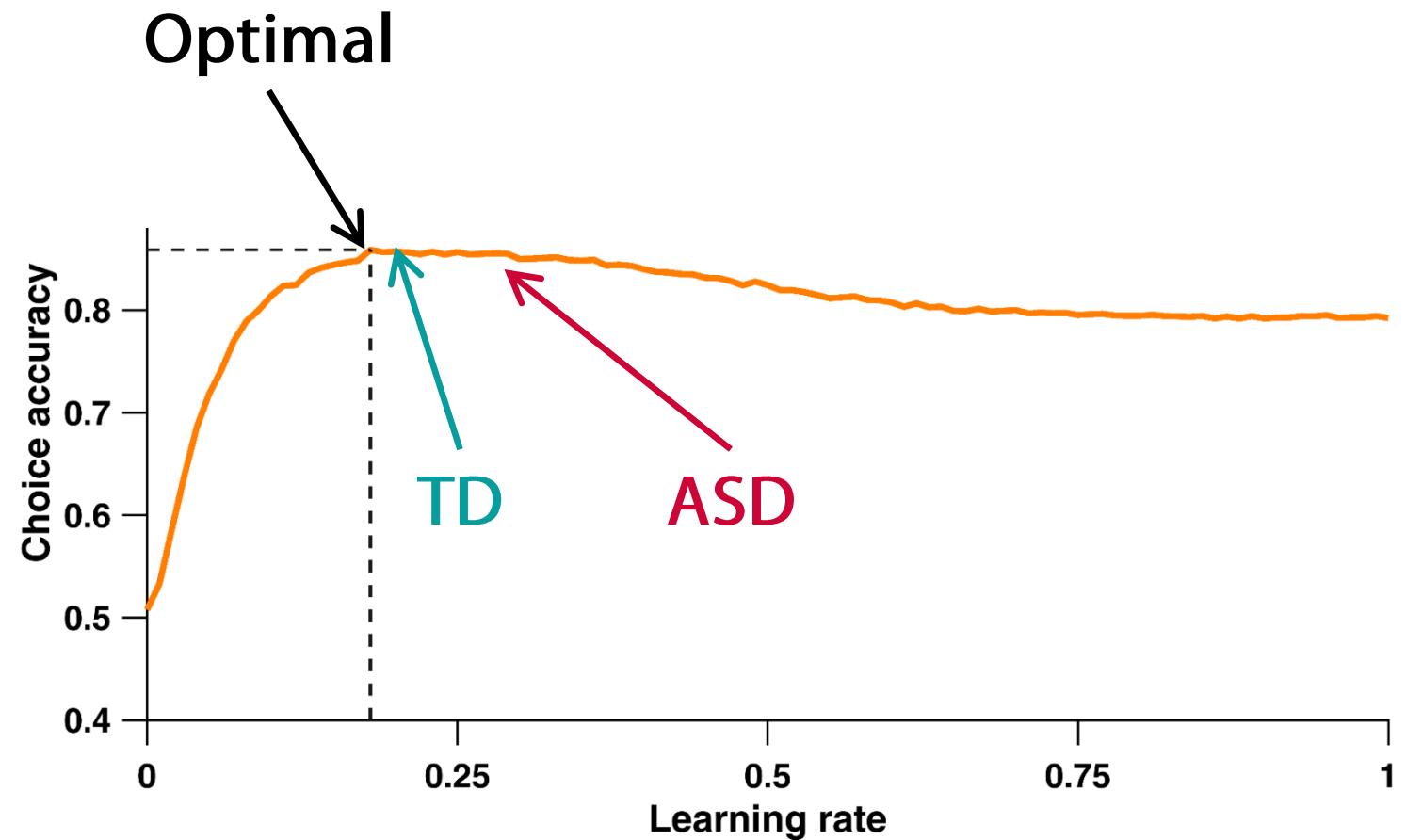
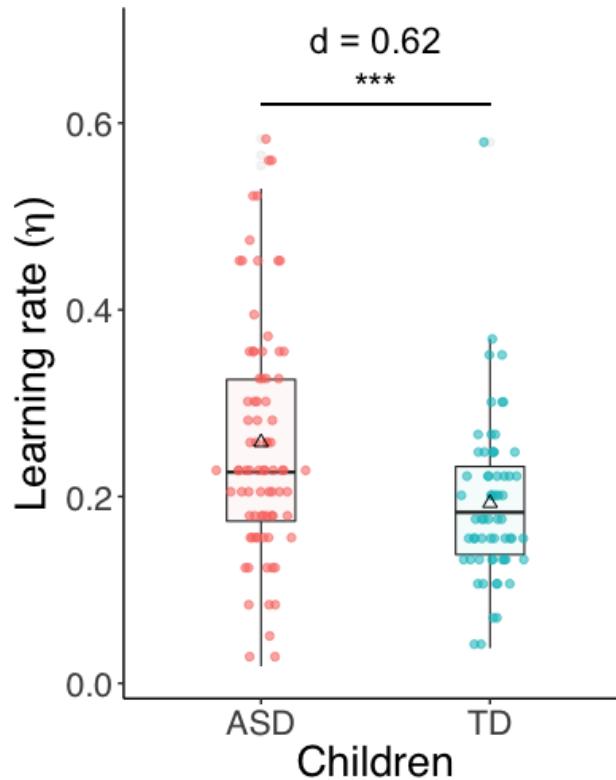
- it hugely depends on the task!



a task = task structure + reward schedule + number of trials

An optimal learning rate exists only when the task is settled.

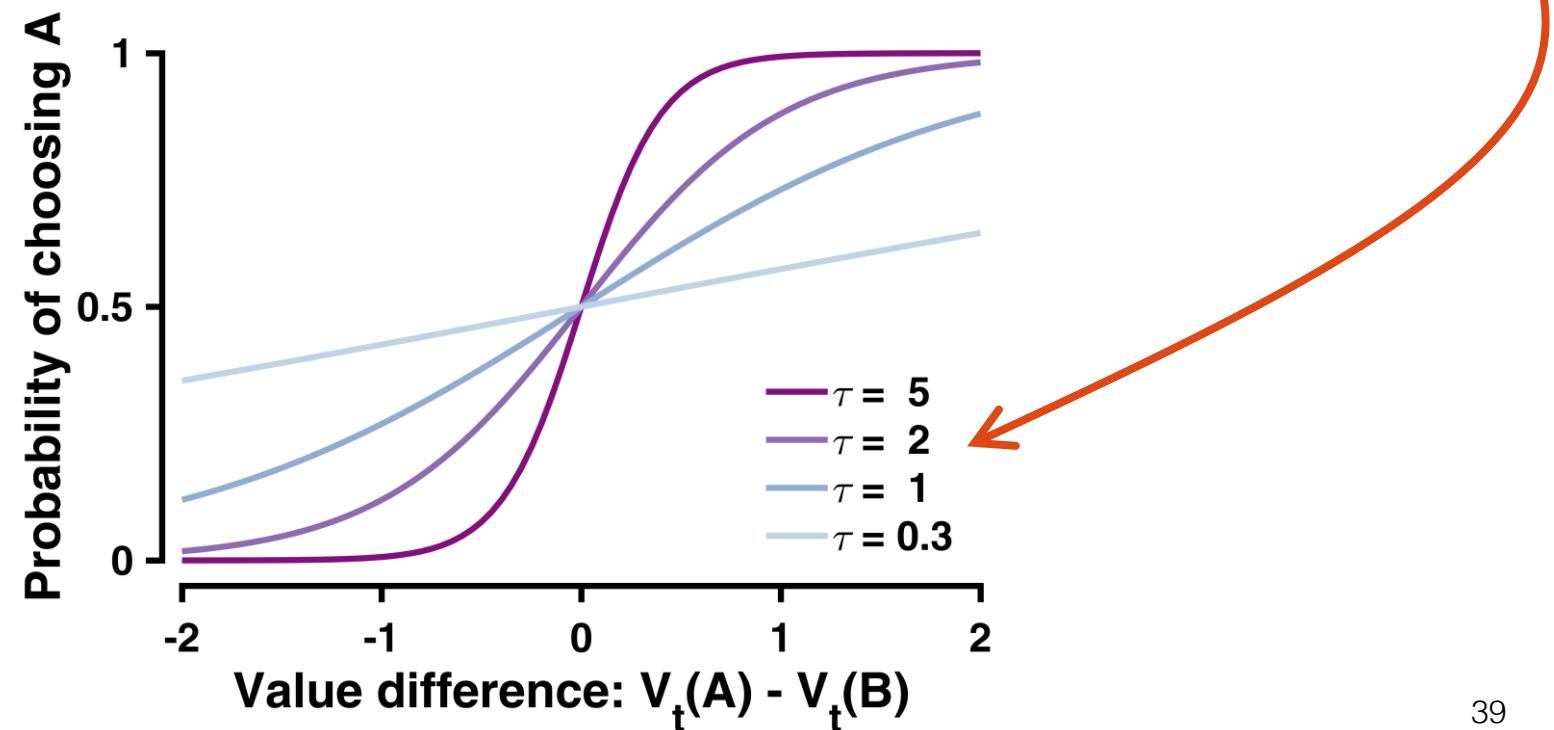
# Optimal learning rate?



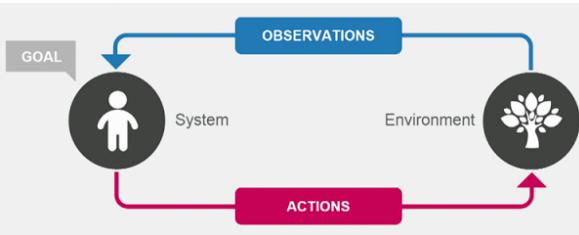
## Choice rule: softmax



$$p_t(A) = \frac{e^{\tau * V_t(A)}}{e^{\tau * V_t(A)} + e^{\tau * V_t(B)}}$$
$$= \frac{1}{1 + e^{-\tau * (V_t(A) - V_t(B))}}$$



# Q-Learning



Value update:

$$V_{t+1} = V_t + \alpha * PE_t$$

Prediction error:

$$PE_t = R_t - V_t$$

choice rule (sigmoid /softmax):

$$p(C=a) = \frac{1}{1+e^{\tau*(v(b)-v(a))}}$$

$\alpha$  - learning rate

PE - reward prediction error

V - value

R - reward

$\tau$  - softmax temperature

# Some terminologies

simple Q-Learning

$$V_{t+1} = V_t + \alpha (R_t - V_t)$$

$$p_t(A) = \frac{1}{1+e^{-\tau*(v(A)-v(B))}}$$

$$\text{LL} = \text{categorical\_lpmf}(C_t | p_t)$$

simple linear regression

$$\mu = a + b X$$

$$\text{LL} = \text{normal\_lpdf}(Y | \mu, \delta)$$

# Some terminologies

## simple Q-Learning

$$V_{t+1} = V_t + \alpha (R_t - V_t)$$

$$p_t(A) = \frac{1}{1+e^{-\tau*(v(A)-v(B))}}$$

$$\text{LL} = \text{categorical_lpmf}(C_t | p_t)$$

## simple linear regression

$$\mu = a + b X$$

$$\text{LL} = \text{normal_lpdf}(Y | \mu, \delta)$$

data / observation

free parameter / model parameter

internal/latent (model) variable

likelihood function

# How do I know which likelihood to use?



## The distribution zoo

by

Ben Lambert and Fergus Cooper

Last month: used by 285 people over 451 sessions in 41 countries

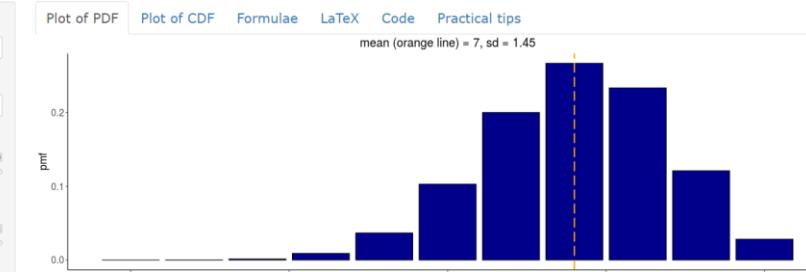
Since created: used by 4072 people over 6785 sessions in 107 countries

Category of distribution: Discrete Univariate

Distribution type: Binomial

size: 10

probability: 0.7



Category of distribution: Multivariate

Distribution type: Multivariate Normal

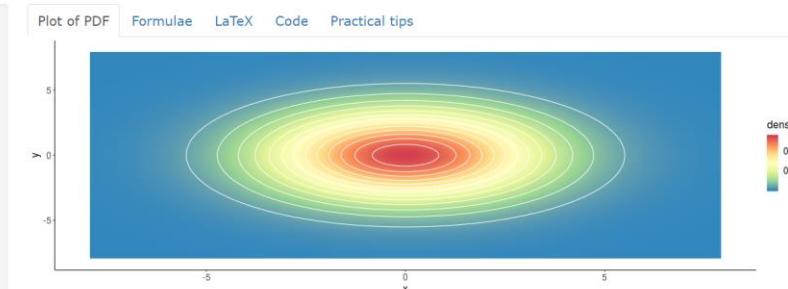
Mean of X: 0

Mean of y: 0

Standard deviation of x: 2.4

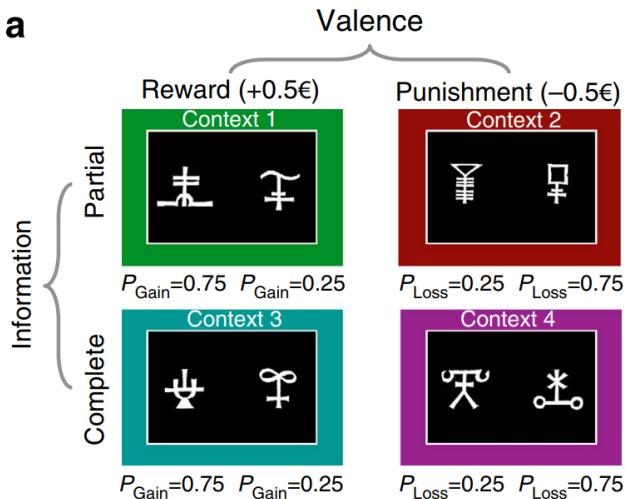
Standard deviation of y: 2.4

Correlation between x and y: 0

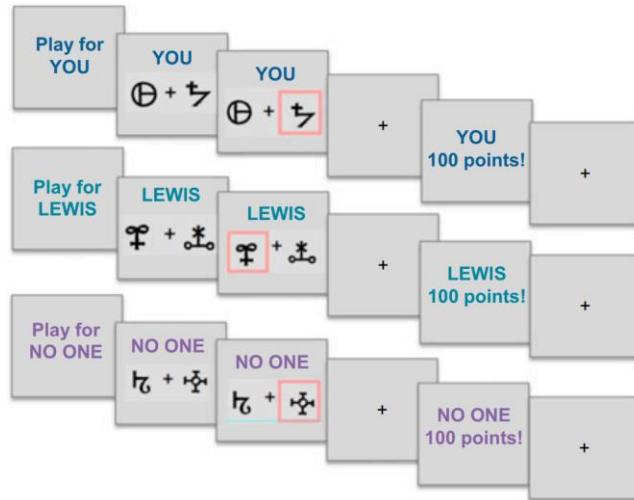


# Generalizing RL framework

a

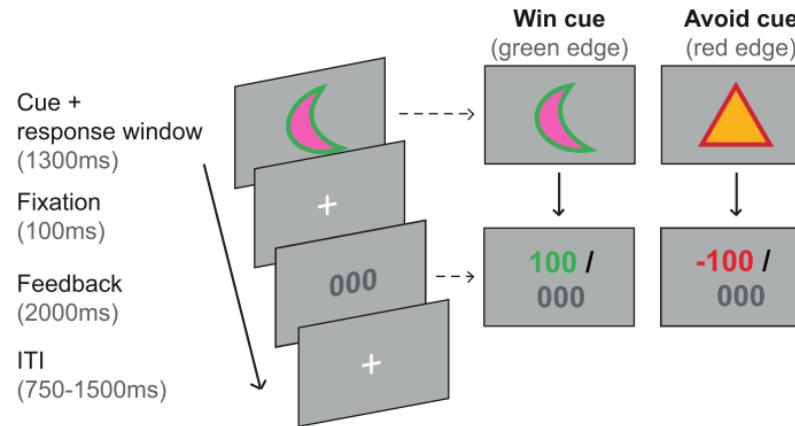


[Palminteri et al. \(2015\)](#)



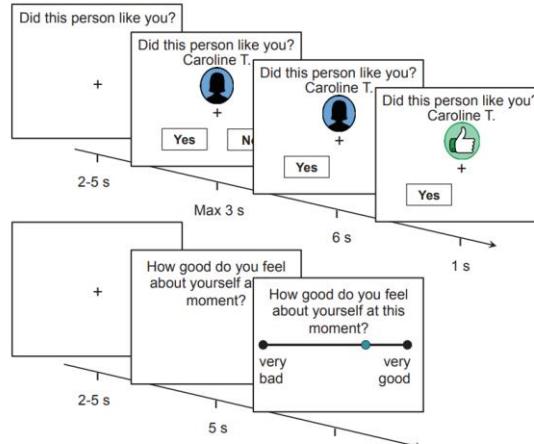
[Lockwood et al. \(2016\)](#)

A. Trial details



[Swart et al. \(2017\)](#)

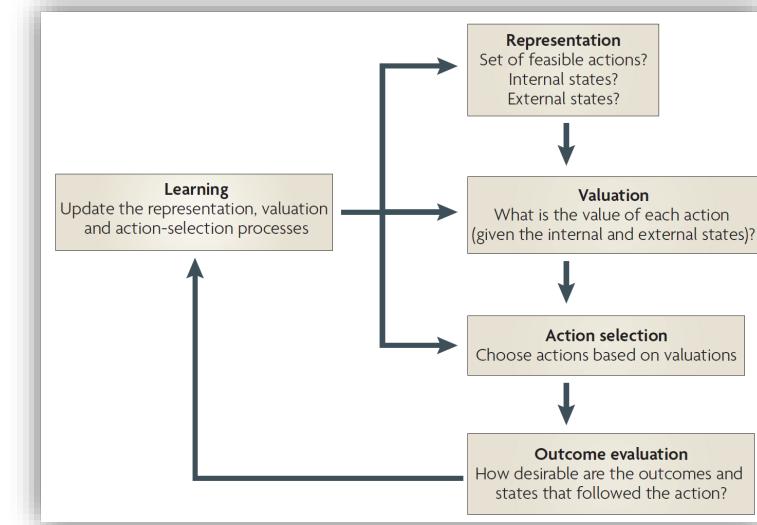
A



B

|                   | Thumbs Up (%) | Thumbs Down (%) |
|-------------------|---------------|-----------------|
| Blue silhouette   | 85%           | 15%             |
| Yellow silhouette | 70%           | 30%             |
| Purple silhouette | 30%           | 70%             |
| Orange silhouette | 15%           | 85%             |

[Will et al. \(2017\)](#)

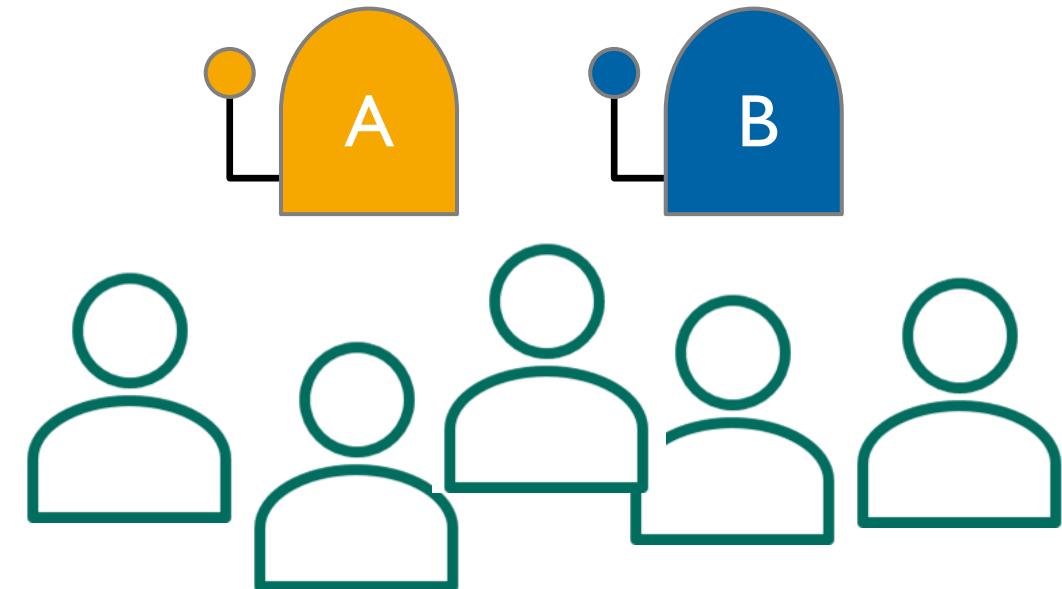
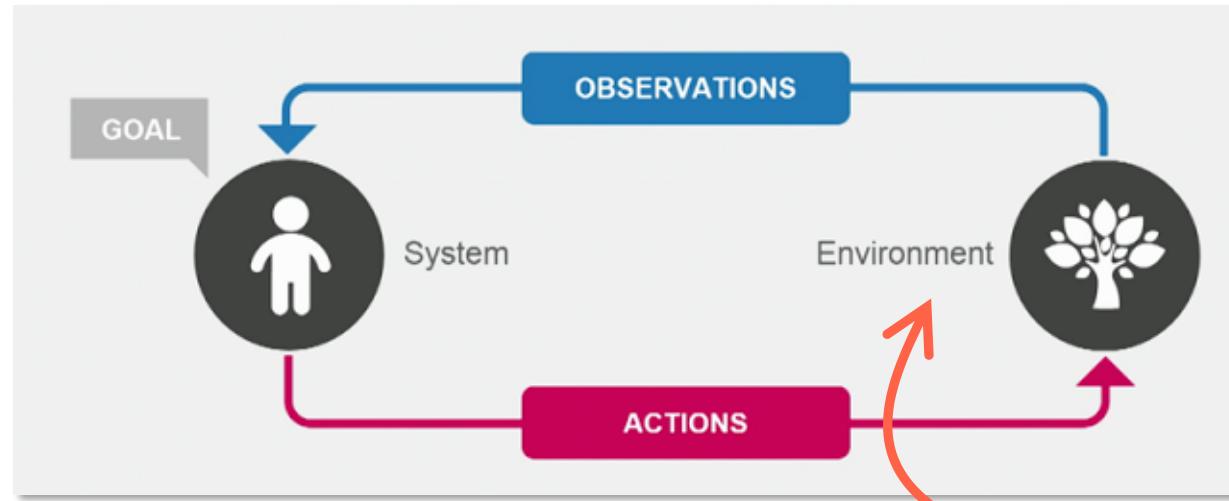


# Social learning & decision-making

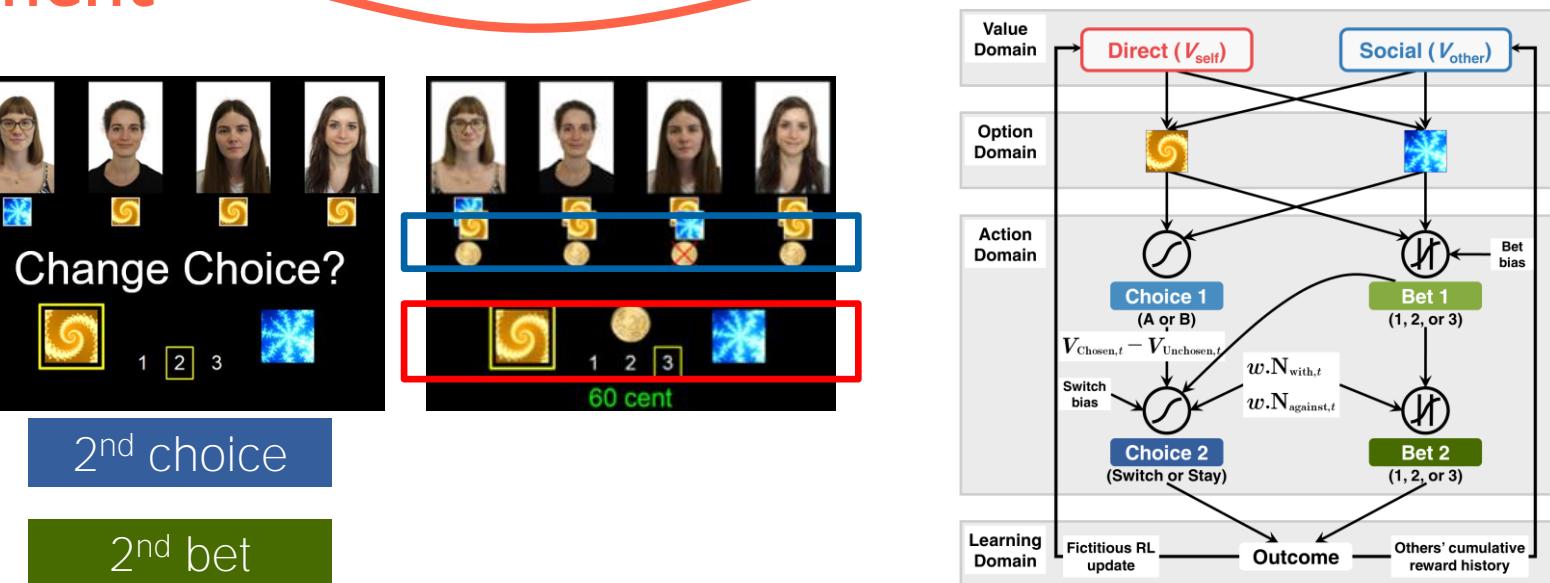
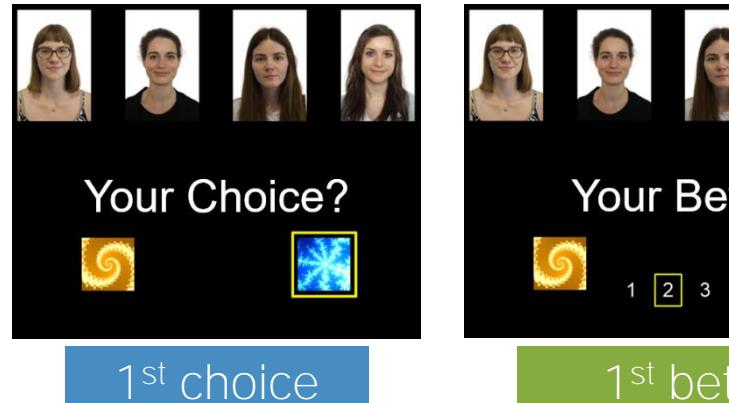
| Decision-making stage (value type)            | Target (and reference frame) of valuation |  |  |
|---|---|--|--|
|   | Value of other people (for oneself)       | Value of other people's experiences (for them) | Value of social constellation (for normative social principle) |
| Choice (decision value and anticipated value) | Deciding whether to marry someone         | Choosing a school for your child               | Altruistic punishment of norm violations                       |
| Outcome (experienced value)                   | Being smiled at                           | Empathy with someone's pain                    | Enjoying fair distributions                                    |
| Learning (anticipated value)                  | Learning about someone's trustworthiness  | Learning about someone else's preferences      | Changing your opinion to increase social conformity            |



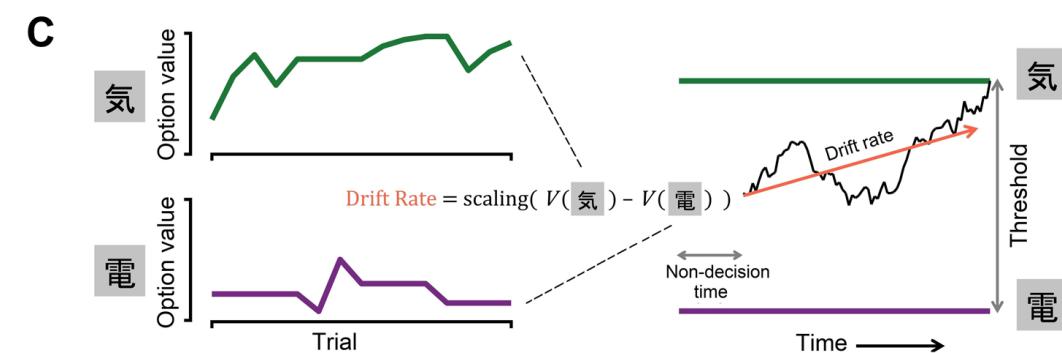
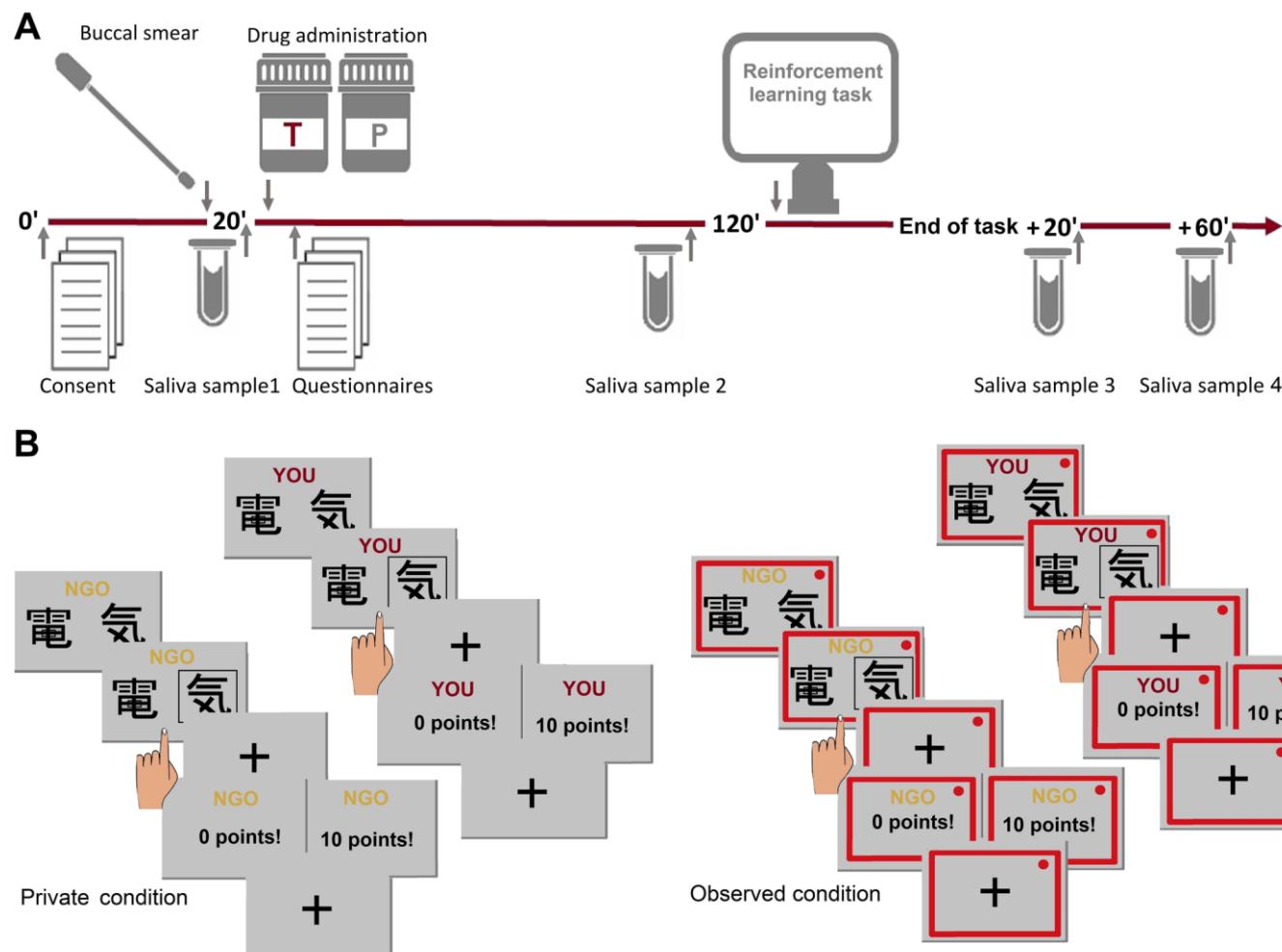
# learn from others (to benefit oneself)



Social environment



# learn for others (pro-social)



# Some more food for thought

What is a social RL task?

There are commonly two variations: either reward learning in social contexts (e.g. learn to expect monetary reward for a social partner) or social feedback learning (e.g. learn to expect social status or social evaluation). When the goal is to compare different types of feedback (e.g. social vs. non-social feedback), we suggest matching the feedback as closely as possible on ‘domain general’ properties, such as salience or preference

Lockwood et al., 2016; Will et al., 2017

Is the learning rate static across trials, or dynamically adapting along the course of the experiment?

The learning rate does not necessarily have to be constant. But in the case of the Rescorla–Wagner model (and related models), the learning rate is indeed static. A dynamic learning rate, however, is possible when other types of models are applied. Note that the interpretation of the learning rate we discussed in the main text is independent of this constant vs dynamic property

Li et al., 2011; Mathys et al., 2011

Is it possible to use RL models in the absence of choice data?

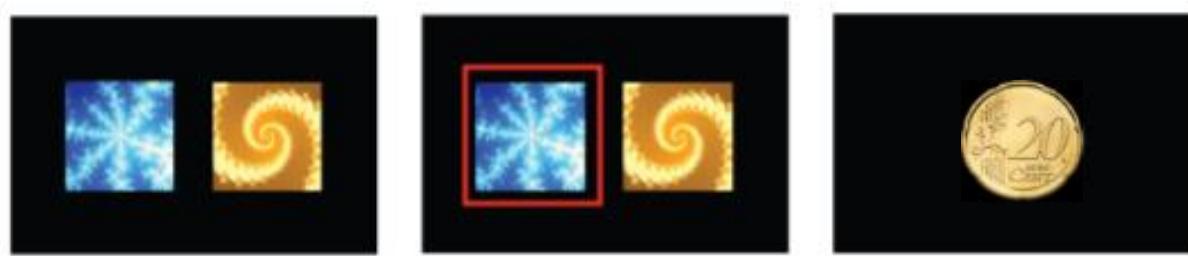
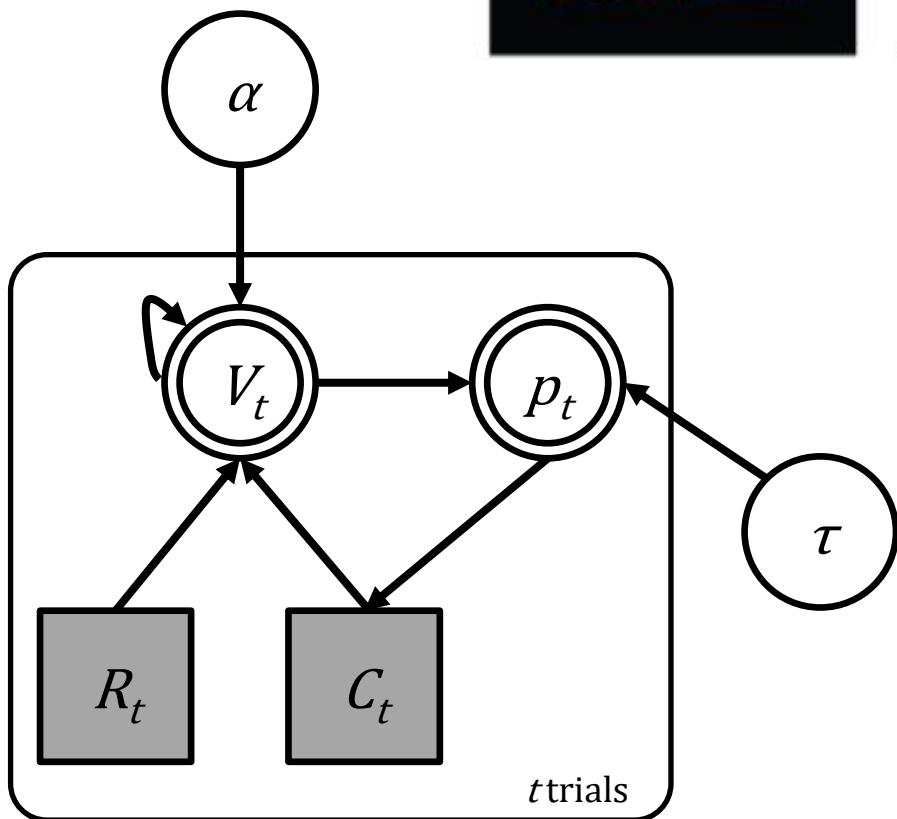
At least some sort of data is needed to perform model estimation. For example, skin conductance response (SCR) or pupil size response (PSR) have been used to fit RL models in associative fear learning tasks, where choice data was not available

Li et al., 2011; Tzovara et al., 2018

# Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- **Implementing Q-Learning model for one subject in Stan**
- Fitting multiple subjects with the hBayesDM package
- Q&A

# RL – Implementation



$$\alpha \sim \text{Uniform}(0, 1)$$

$$\tau \sim \text{Uniform}(0, 3)$$

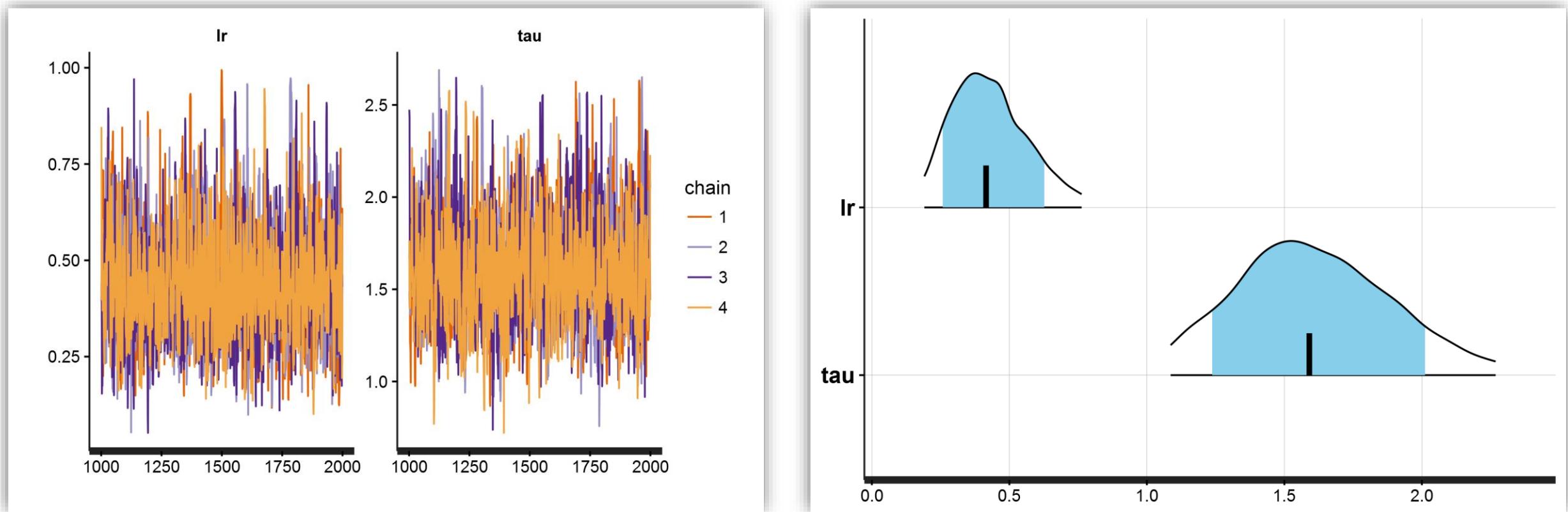
$$p_t(C = A) = \frac{1}{1 + e^{\tau(V_t(B) - V_t(A))}}$$

$$V_{t+1}^c = V_t^C + \alpha (R_t - V_t^C)$$

○ deterministic variable

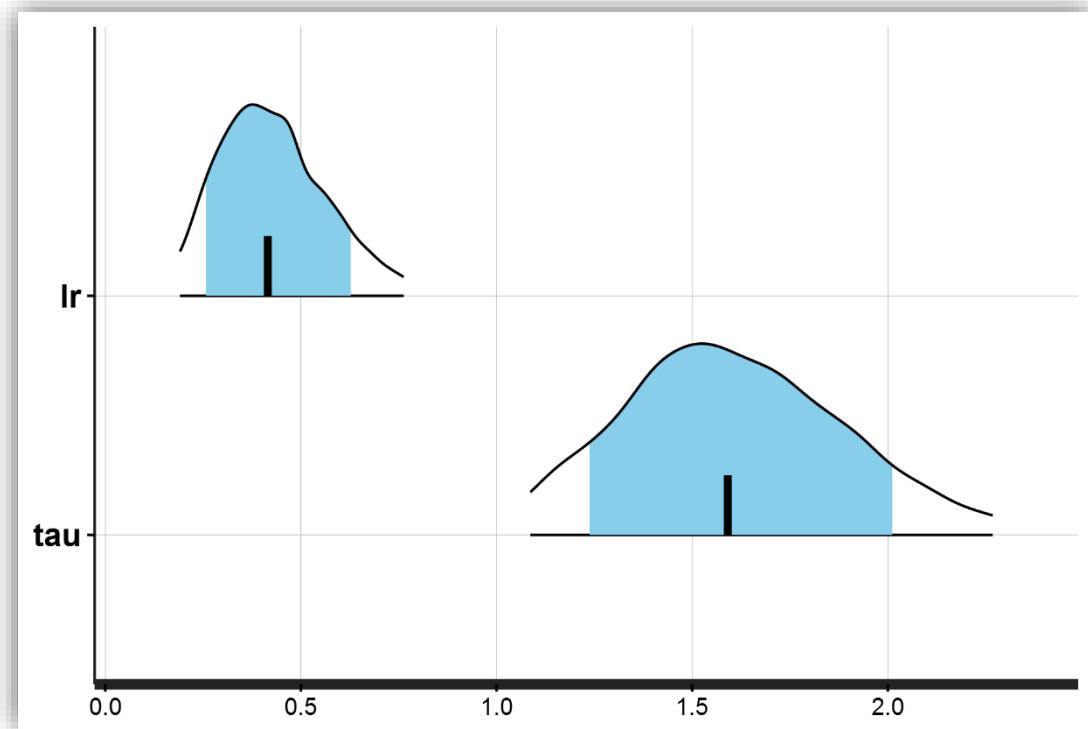
**Live demo of building the Q-Learning model  
model in Stan from scratch**

# RL - MCMC Output

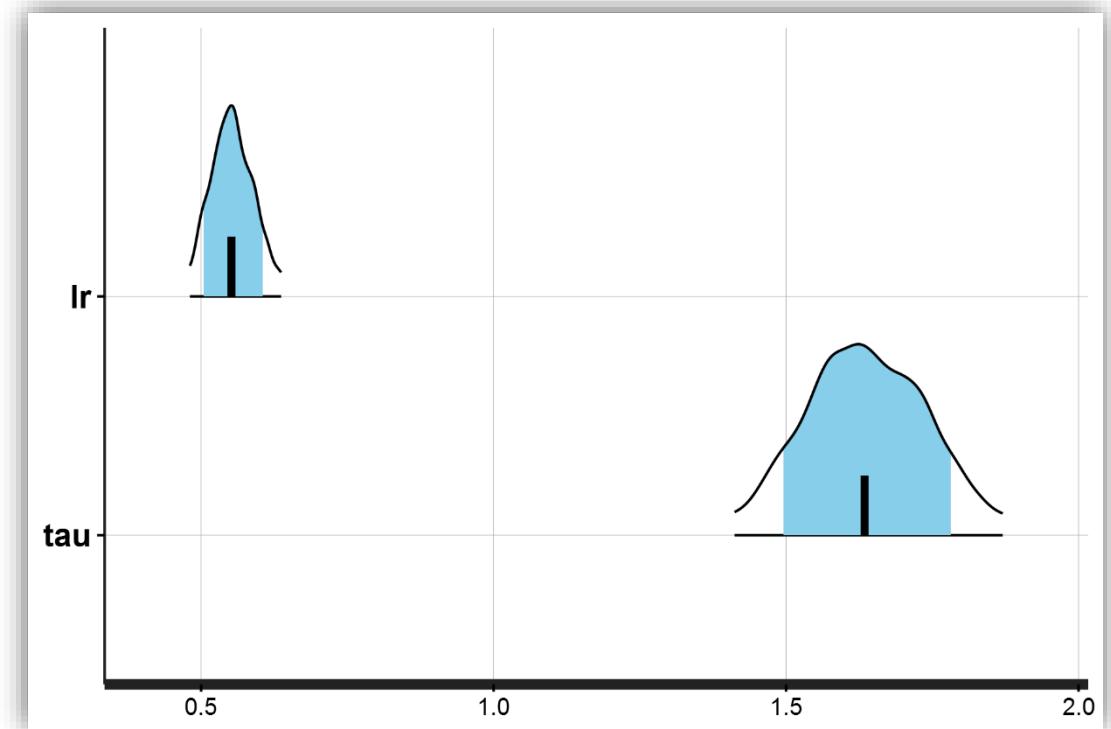


# What if we have more data?

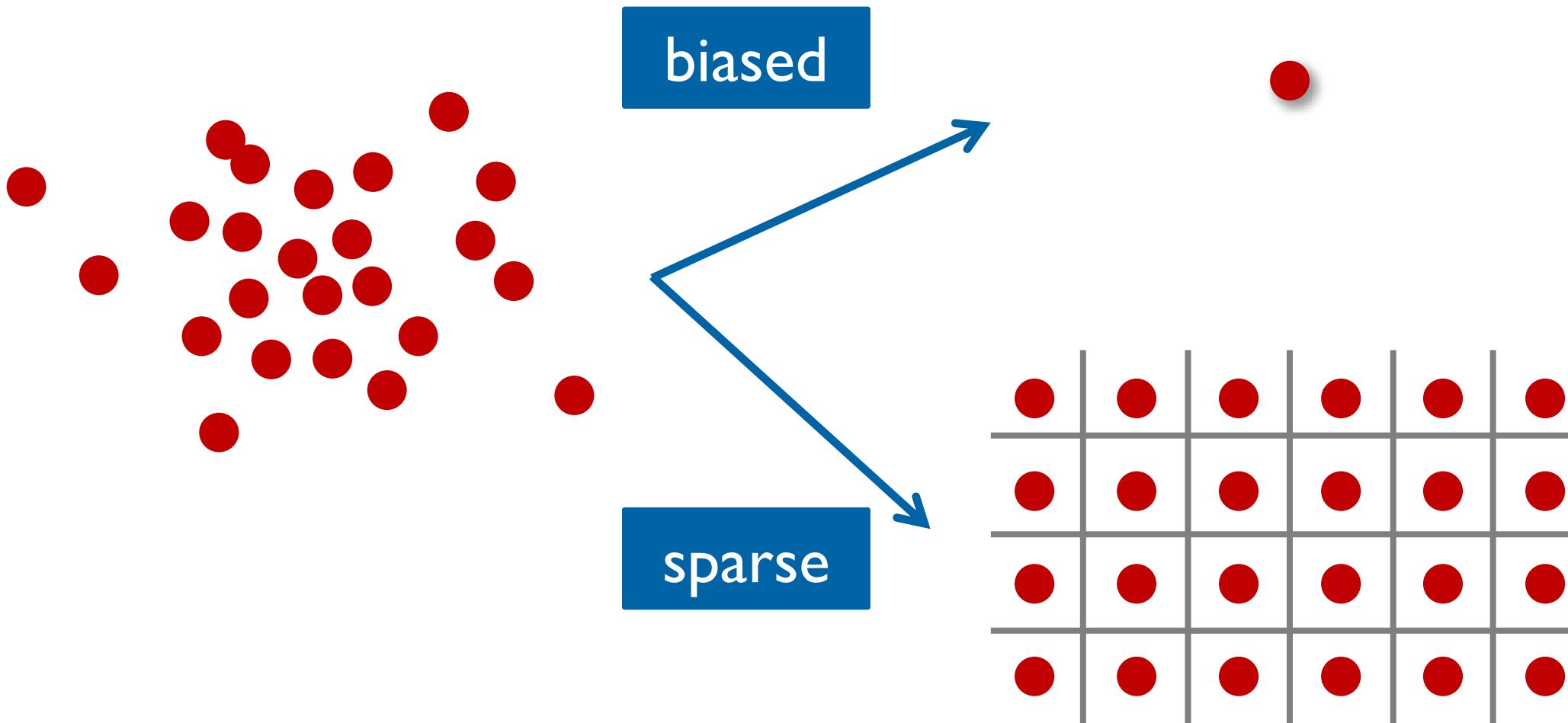
$N = 1$



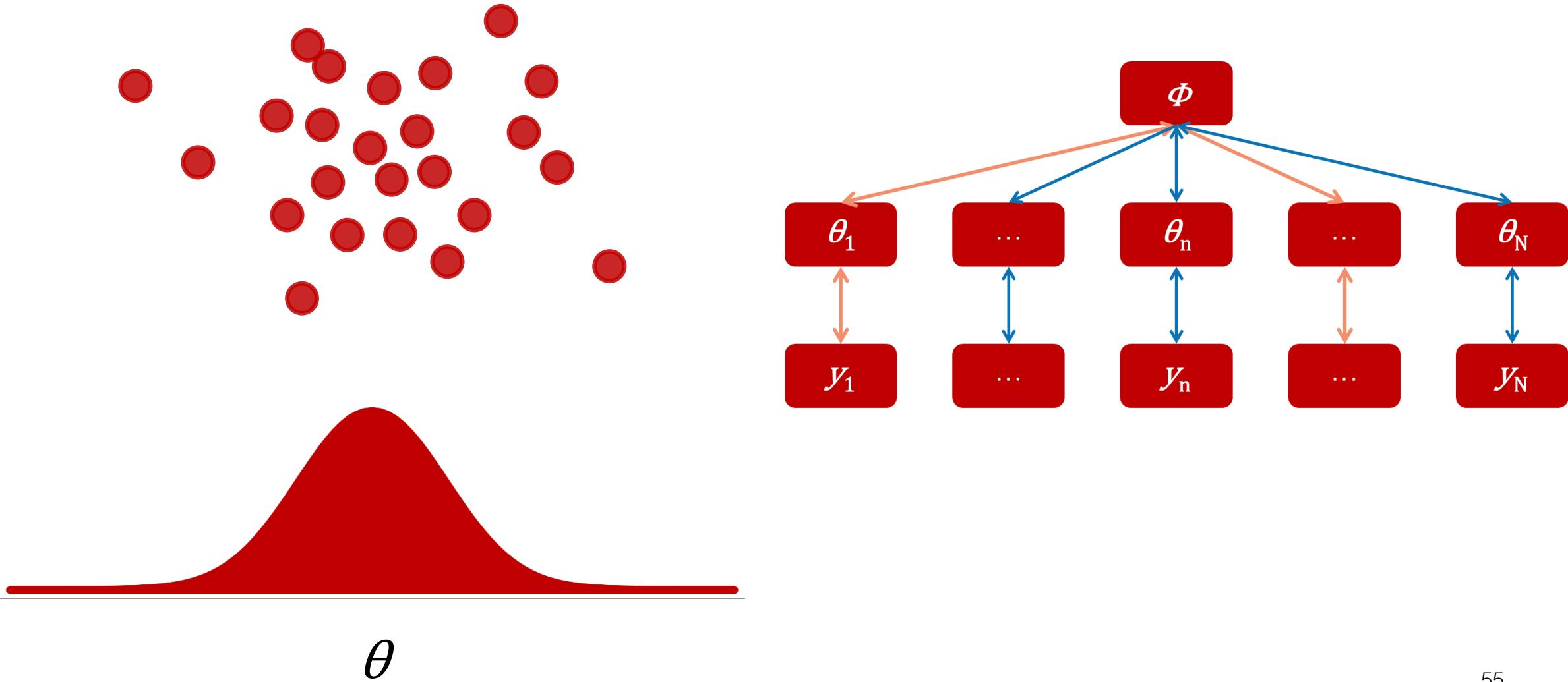
$N = 10$



# Fitting Multiple Participants

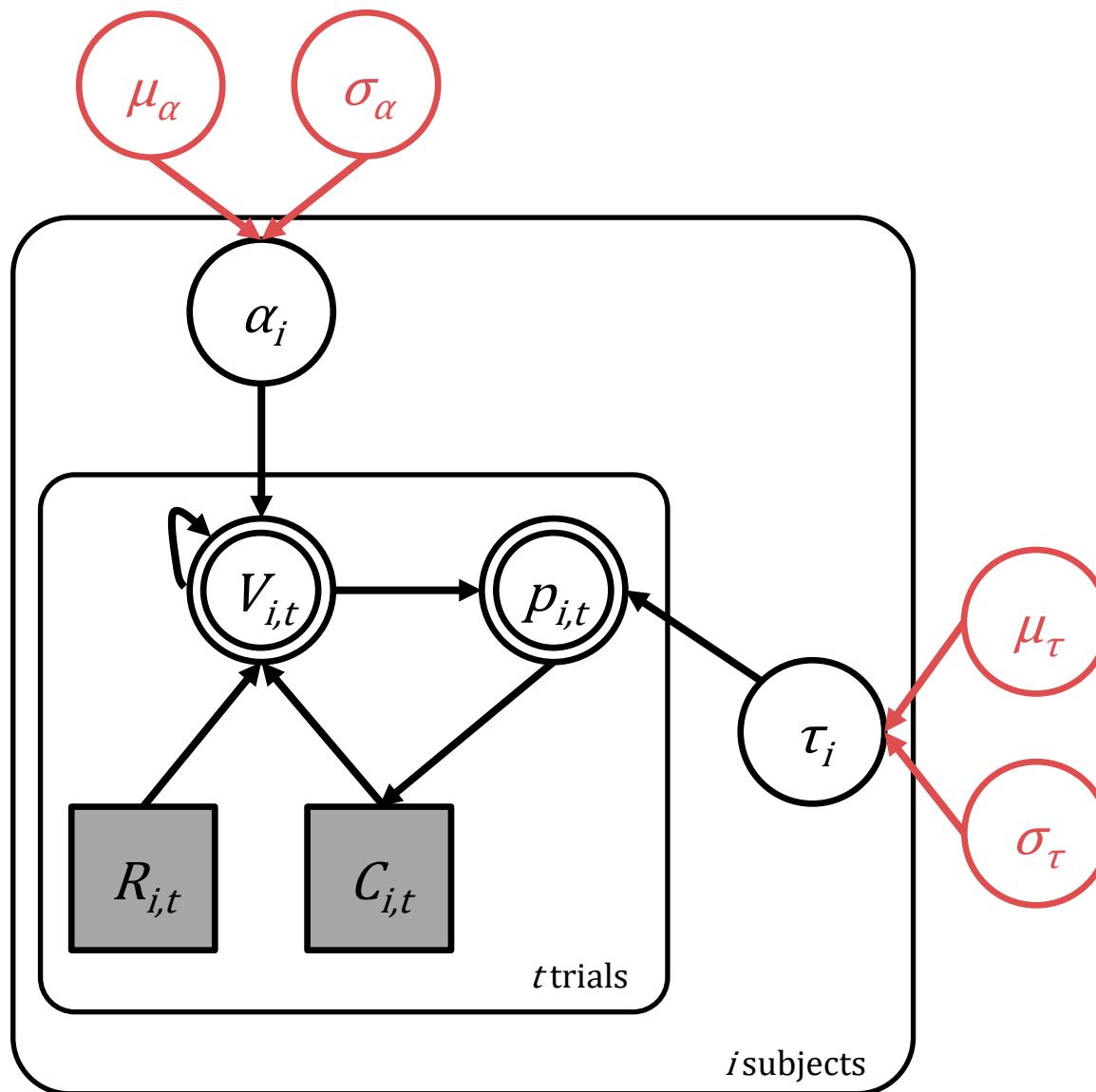


# Fitting Multiple Participants with hierarchical Bayesian analysis (HBA)



# Hierarchical RL Model

Voillà!



# Information Criteria

AIC – Akaike information criterion

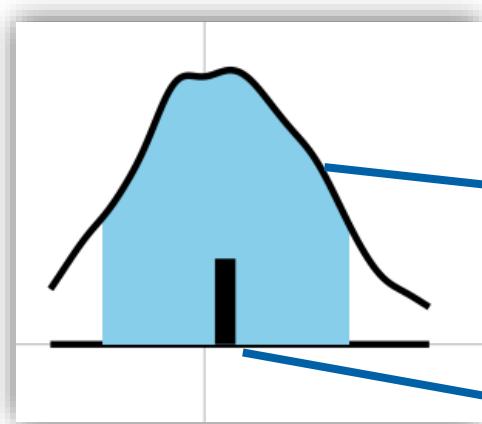
DIC – Deviance Information Criterion

WAIC – Widely Applicable Information Criterion

(Watanabe–Akaike information criterion)

finding the model that has  
the highest out-of-sample  
predictive accuracy

approximation to LOO



WAIC: using **entire posterior** distribution

AIC/DIC: using **point estimation**

# Compute WAIC from Likelihood

$$\text{WAIC} = -2 \widehat{\text{elpd}}_{\text{waic}}$$

expected log pointwise predictive density

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{lpd}} - \widehat{p}_{\text{waic}}$$

$$\begin{aligned}\widehat{\text{lpd}} &= \text{computed log pointwise predictive density} \\ &= \sum_{i=1}^n \log \left( \frac{1}{S} \sum_{s=1}^S p(y_i | \theta^s) \right).\end{aligned}$$

$$\begin{aligned}\widehat{p}_{\text{waic}} &= \text{estimated effective number of parameters} \\ &= \sum_{i=1}^n V_{s=1}^S (\log p(y_i | \theta^s))\end{aligned}$$

```
lpd <- log(colMeans(exp(log_lik)))
```

```
p_waic <- colVars(log_lik)
```

# The {loo} Package

```
> library(loo)
> LL1    <- extract_log_lik(stanfit)
> loo1   <- loo(LL1)    # PSIS leave-one-out
> waic1 <- waic(LL1)   # WAIC
```

Computed from 4000 by 20 log-likelihood matrix

|          | Estimate | SE  |
|----------|----------|-----|
| elpd_loo | -29.5    | 3.3 |
| p_loo    | 2.7      | 1.0 |
| looic    | 58.9     | 6.7 |

Pareto Smoothed Importance Sampling

# Recording the Log-Likelihood in Stan

```
generated quantities {
  ...
  real log_lik[nSubjects];
  ...

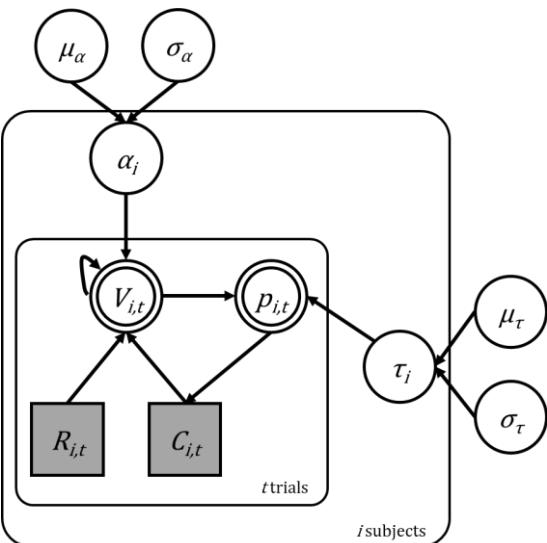
  { # Local section, this saves time and space
    for (s in 1:nSubjects) {
      vector[2] v;
      real pe;

      log_lik[s] = 0;
      v = initV;

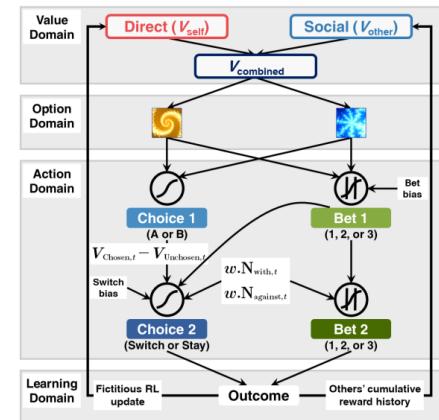
      for (t in 1:nTrials) {
        log_lik[s] = log_lik[s] + categorical_logit_lpmf(choice[s,t] | tau[s] * v);

        pe = reward[s,t] - v[choice[s,t]];
        v[choice[s,t]] = v[choice[s,t]] + lr[s] * pe;
      }
    }
  }
}
```

# HBA sounds good, but...



$$\begin{aligned}
 \mu_\alpha &\sim Uniform(0,1) \\
 \sigma_\alpha &\sim halfCauchy(0,1) \\
 \mu_\tau &\sim Uniform(0,3) \\
 \sigma_\tau &\sim halfCauchy(0,3) \\
 \alpha_i &\sim Normal(\mu_\alpha, \sigma_\alpha)_{\mathcal{T}(0,1)} \\
 \tau_i &\sim Normal(\mu_\tau, \sigma_\tau)_{\mathcal{T}(0,3)} \\
 p_{i,t}(C = A) &= \frac{1}{1 + e^{\tau_i(V_{i,t}(B) - V_{i,t}(A))}} \\
 V_{i,t+1}^c &= V_{i,t}^c + \alpha_i(R_{i,t} - V_{i,t}^c)
 \end{aligned}$$



$$\begin{aligned}
 V_{\text{self},t} &= [V_{\text{self},t}(\text{A}), V_{\text{self},t}(\text{B})] \\
 V_{\text{other},t} &= [V_{\text{other},t}(\text{A}), V_{\text{other},t}(\text{B})] \\
 V_t &= \beta_{\text{self}} V_{\text{self},t} + \beta_{\text{other}} V_{\text{other},t} \\
 C1_t &\sim \text{Categorical}(\text{Softmax}(V_t)) \\
 U_{\text{bet1},t} &= \beta_{\text{bias}_{\text{self}}} + \beta_{\text{valf}_{\text{self}}} (V_{\text{chosen},C1,t} - V_{\text{unchosen},C1,t}) \\
 B1_t &\sim \text{OrderedLogistic}(U_{\text{bet1},t} | \theta) \\
 w.N_{\text{against},t} &= \sum_{s=1}^{s=1} \sum_{n=1}^{n=4} w_{s,t} \\
 &= \sum_{s=1}^{s=1} \sum_{n=1}^{n=4} w_{s,t} \\
 w.N_{\text{with},t} &= \sum_{s=1}^{s=1} \sum_{n=1}^{n=4} w_{s,t} \\
 V_t(\text{switch}) &= \beta_{\text{bias}_{\text{self}}} + \beta_{\text{valf}_{\text{self}}} (V_{\text{chosen},C1,t} - V_{\text{unchosen},C1,t}) + \beta_{\text{against}} w.N_{\text{against},t} \\
 C2 &\sim \text{Bernoulli}(V_t(\text{switch})) \\
 U_{\text{bet2},t} &= \begin{cases} U_{\text{bet1},t} + \beta_{\text{with}_{\text{self}}} w.N_{\text{with},t} + \beta_{\text{against}_{\text{self}}} w.N_{\text{against},t}, & \text{if } C1 = C2 \\ U_{\text{bet1},t} + \beta_{\text{with}_{\text{self}}} w.N_{\text{with},t} + \beta_{\text{against}_{\text{self}}} w.N_{\text{against},t}, & \text{if } C1 \neq C2 \end{cases} \\
 B2_t &\sim \text{OrderedLogistic}(U_{\text{bet2},t} | \theta) \\
 \Phi(x) &= \frac{1}{1 + e^{-x}} \\
 \delta_{\text{self,chosen},C2,t} &= R_{\text{self},t} - V_{\text{self,chosen},C2,t} \\
 \delta_{\text{self,unchosen},C2,t} &= -R_{\text{self},t} - V_{\text{self,unchosen},C2,t} \\
 V_{\text{self,chosen},C2,t+1} &= V_{\text{self,chosen},C2,t} + \alpha \delta_{\text{self,chosen},C2,t} \\
 V_{\text{self,unchosen},C2,t+1} &= V_{\text{self,unchosen},C2,t} + \alpha \delta_{\text{self,unchosen},C2,t}
 \end{aligned}$$

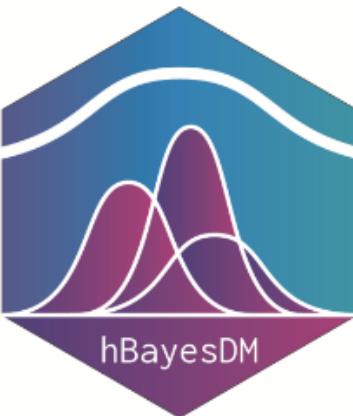
# hBayesDM package

## hBayesDM

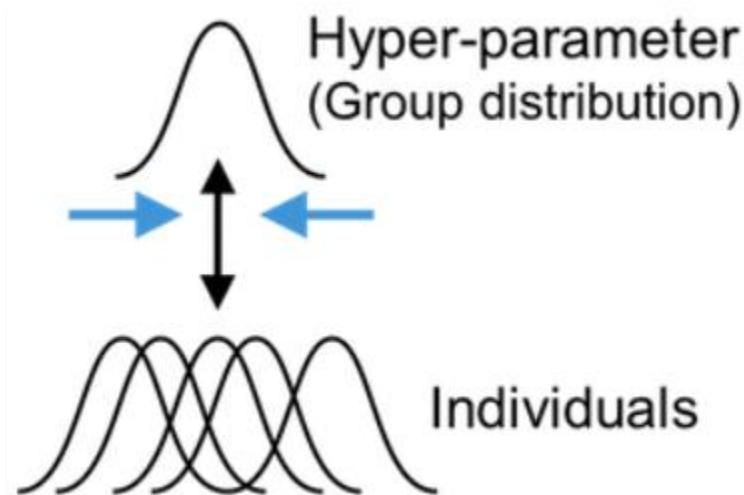
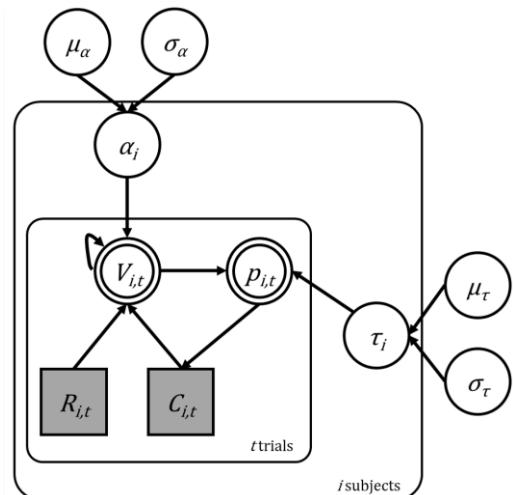
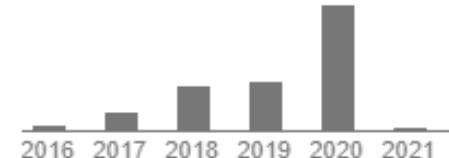
repo status Active build passing CRAN 1.0.2 – 2019-11-13 downloads 33K

DOI 10.1162/CPSY\_a\_00002

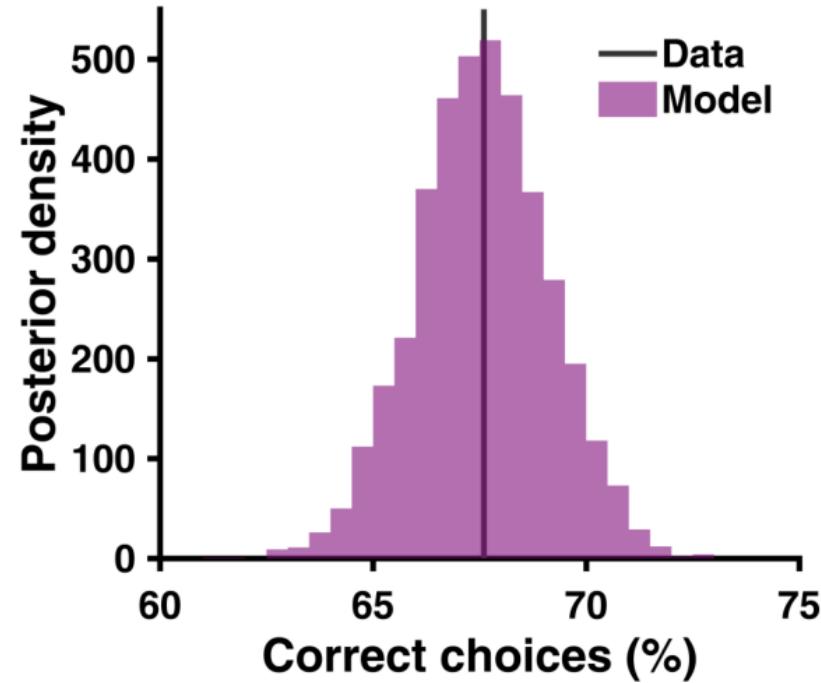
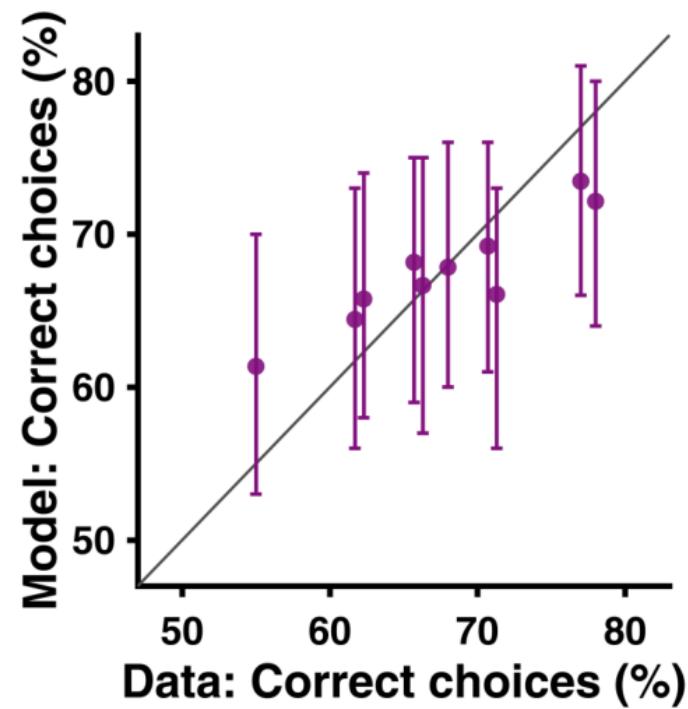
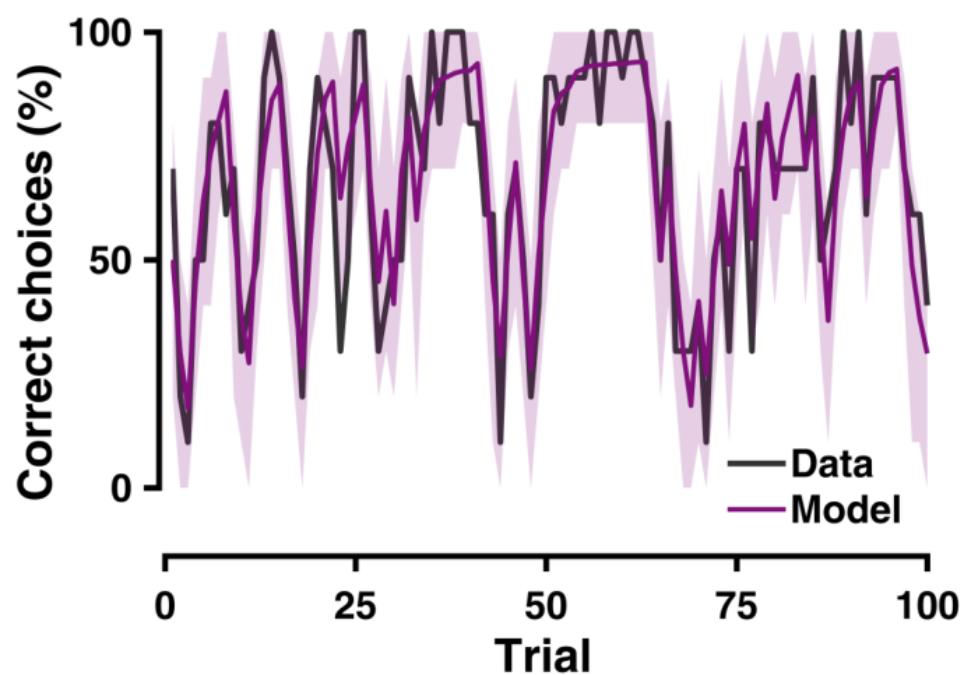
hBayesDM (hierarchical Bayesian modeling of Decision-Making tasks) is a user-friendly package that offers hierarchical Bayesian analysis of various computational models on an array of decision-making tasks. hBayesDM uses Stan for Bayesian inference.



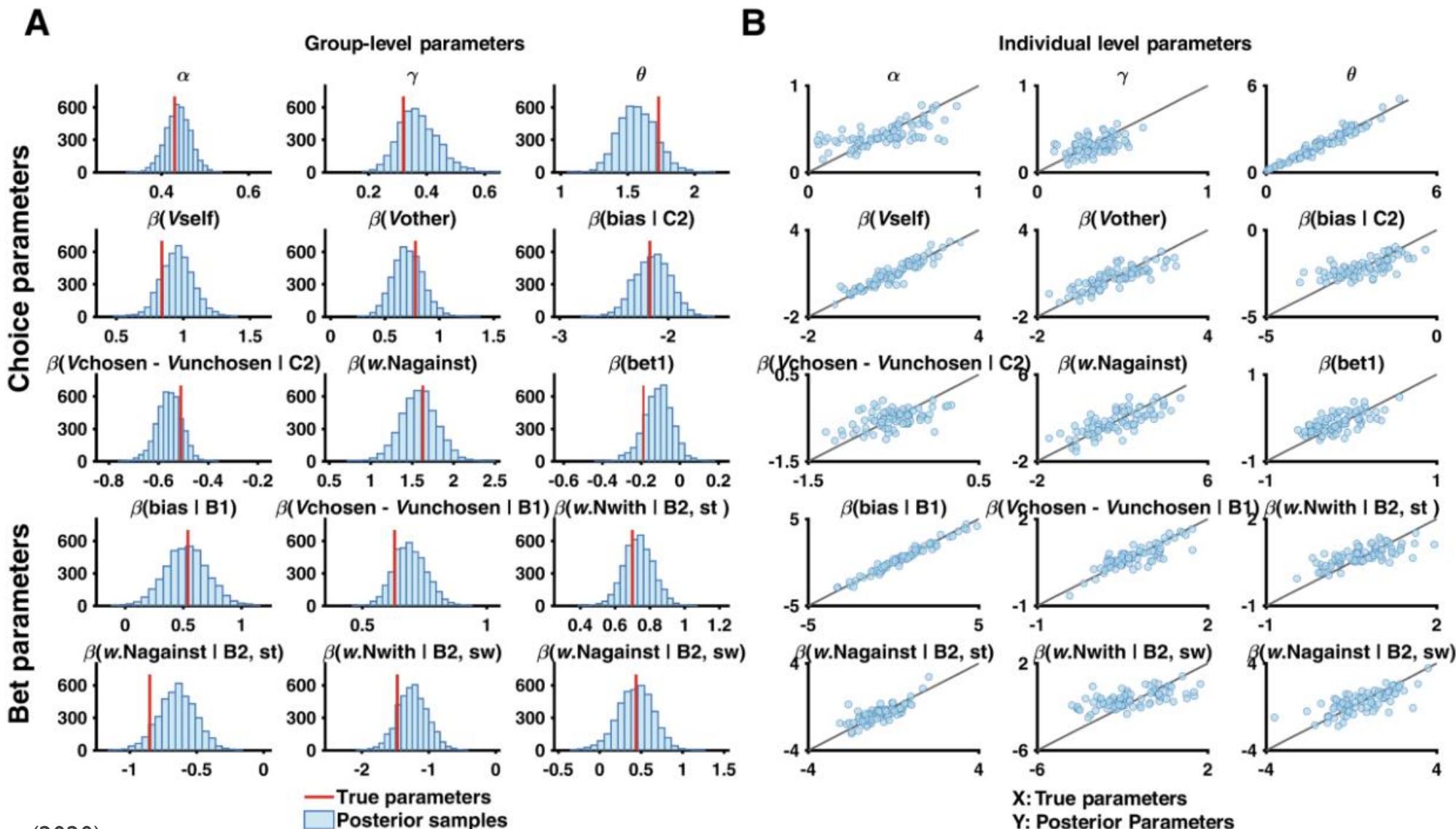
Cited by 115



# Posterior predictive check: *can the model resonate with data?*



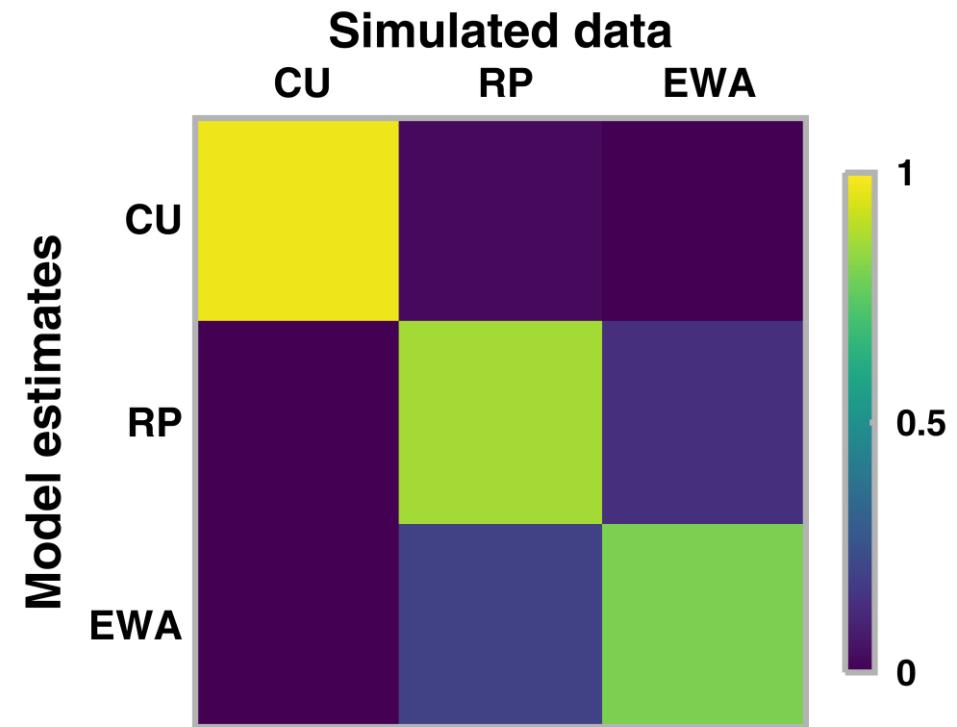
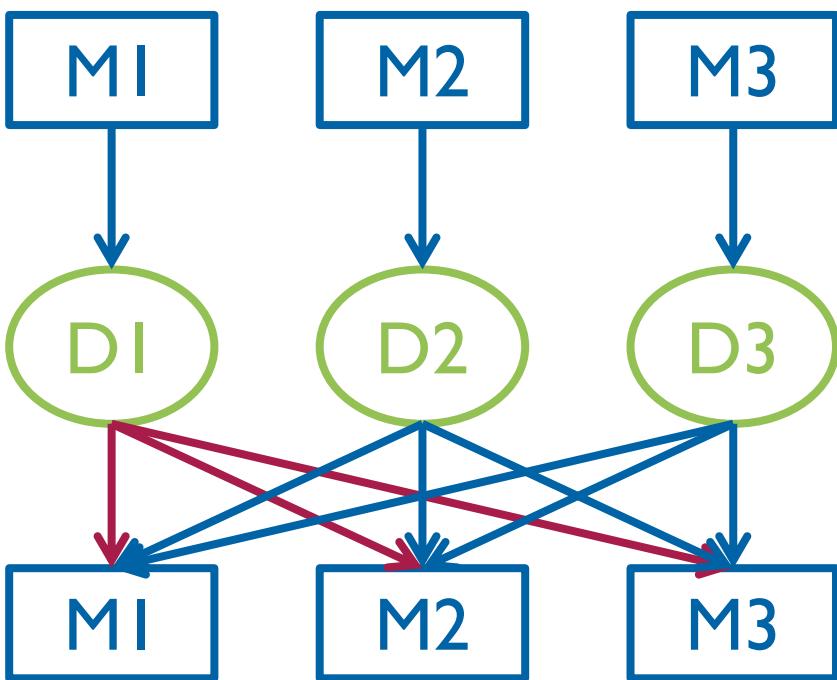
# Parameter recovery: are parameters identifiable?



# Model recovery: are models identifiable?

generative  
process

fitting  
process



# Recommended reading: tutorial

## Using reinforcement learning models in social neuroscience: frameworks, pitfalls and suggestions of best practices

Lei Zhang , Lukas Lengersdorff, Nace Mikus, Jan Gläscher, Claus Lamm [Author Notes](#)

*Social Cognitive and Affective Neuroscience*, Volume 15, Issue 6, June 2020, Pages 695–707, <https://doi.org/10.1093/scan/nsaa089>

<https://doi.org/10.1093/scan/nsaa089>

ACCEPTED MANUSCRIPT

## Computational modelling of social cognition and behaviour—a reinforcement learning primer

Patricia L Lockwood , Miriam Klein-Flügge 

*Social Cognitive and Affective Neuroscience*, nsaa040, <https://doi.org/10.1093/scan/nsaa040>

Published: 30 March 2020 [Article history](#) 

<https://doi.org/10.1093/scan/nsaa040>

## Ten simple rules for the computational modeling of behavioral data



Robert C Wilson , Anne GE Collins 

University of Arizona, United States; University of California, Berkeley, United States

<https://elifesciences.org/articles/49547>

## The Importance of Falsification in Computational Cognitive Modeling

Stefano Palminteri,<sup>1,2,\*‡</sup> Valentin Wyart,<sup>1,2,\*‡</sup> and Etienne Koechlin<sup>1,2,\*</sup>

<https://doi.org/10.1016/j.tics.2017.03.011>

# Recommended reading: empirical work

## Catecholaminergic challenge uncovers distinct Pavlovian and instrumental mechanisms of motivated (in)action



Jennifer C Swart<sup>✉</sup>, Monja I Froböse, Jennifer L Cook, Dirk EM Geurts, Michael J Frank, Roshan Cools, Hanneke EM den Ouden<sup>✉</sup>  
Radboud University, The Netherlands; University of Birmingham, United Kingdom; Radboud University Medical Center, The Netherlands; Linguistic and Psychological  
Sciences, Brown University, United States; Brown University, United States

<https://elifesciences.org/articles/22169>

## Social threat learning transfers to decision making in humans

Björn Lindström<sup>a,b,c,1</sup>, Armita Golkar<sup>c,d</sup>, Simon Jangard<sup>c</sup>, Philippe N. Tobler<sup>b</sup>, and Andreas Olsson<sup>c</sup>

<sup>a</sup>Department of Social Psychology, University of Amsterdam, 1018 WT Amsterdam, The Netherlands; <sup>b</sup>Laboratory for Social and Neural Systems Research, Department of Economics, University of Zürich, 8001 Zürich, Switzerland; <sup>c</sup>Section for Psychology, Department of Clinical Neuroscience, Karolinska Institutet, 171 77 Stockholm, Sweden; and <sup>d</sup>Department of Clinical Psychology, University of Amsterdam, 1018 WT Amsterdam, The Netherlands

<https://www.pnas.org/content/116/10/4732.abstract>

## COGNITIVE NEUROSCIENCE

## A brain network supporting social influences in human decision-making

Lei Zhang<sup>1,2\*</sup> and Jan Gläscher<sup>1,†</sup>

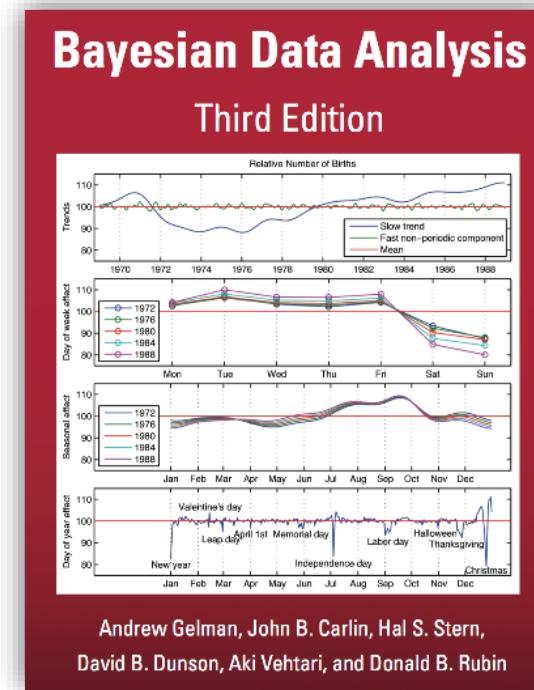
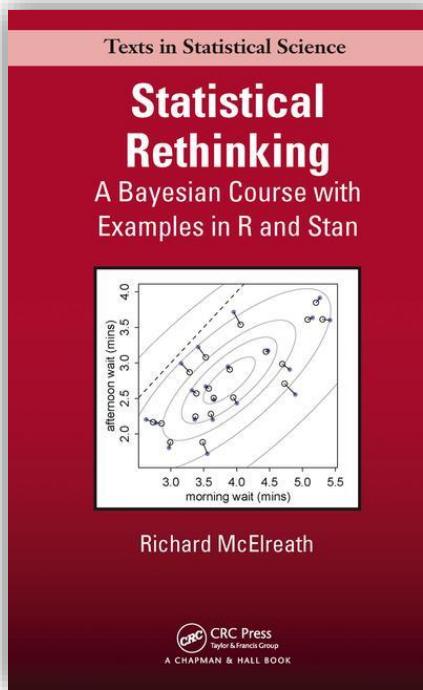
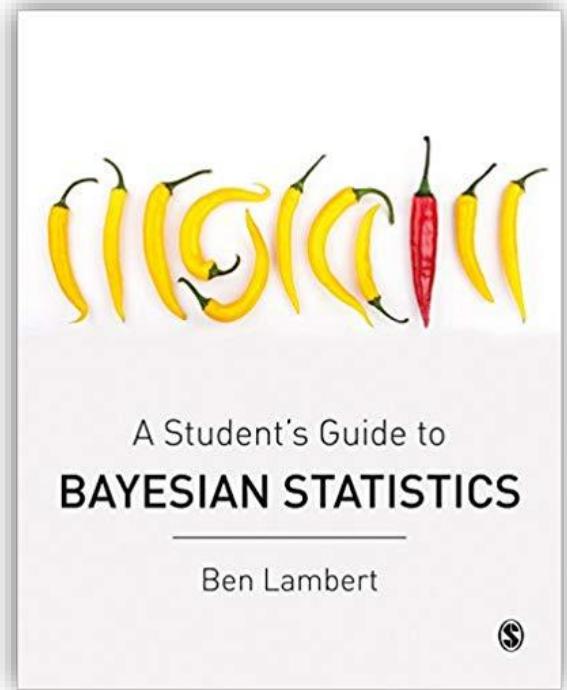
<https://advances.sciencemag.org/content/6/34/eabb4159>

Modeling flexible behavior in childhood to adulthood shows age-dependent learning mechanisms and less optimal learning in autism in each age group

Daisy Crawley<sup>1,2\*</sup>, Lei Zhang<sup>1,2,3,4†</sup>, Emily J. H. Jones<sup>5</sup>, Jumana Ahmad<sup>1,6</sup>,  
Bethany Oakley<sup>1,6</sup>, Antonia San José Cáceres<sup>1,7</sup>, Tony Charman<sup>1,8,9</sup>, Jan  
K. Buitelaar<sup>1,10,11,12</sup>, Declan G. M. Murphy<sup>1,9,13</sup>, Christopher Chatham<sup>4</sup>, Hanneke den  
Ouden<sup>10†</sup>, Eva Loth<sup>1,13‡</sup>, the EU-AIMS LEAP group<sup>11</sup>

<https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000908>

# Recommended reading: book



# Want to learn more about Stan?

# *Workshops*

- StanCon 2019 Hierarchical Models
  - PyData NYC 2019
  - StanCon 2018 Intro Stan
  - StanCon 2018 Hierarchical Models

<https://mc-stan.org/workshops/>

# Stan forum

StanCon 2020, August 11-14 at Oregon State University

Announcements stancon

StanCon 2020 will be at Oregon State University! There will be two days of tutorials followed by two days of talks, open discussions, and statistical modeling. Up-to-date details at <https://mc-stan.org/events/stancon2020>. [read more](#)



 Welcome to the Stan Forums!

The Stan Forums provide a community for asking and answering questions about all aspects of Stan. Before creating a new topic please search the Forums to see if your questions has already been answered, or check out the [read more](#).



Unable to retrieve parameters from a dynamic model

## ■ Modeling ■ fitting-issues ■ specification



Twitter



**Richard McElreath**  
@rlmcelreath



## **Quiche Lorraine,** @dan\_p\_simpson



\mathfrak{Michael}  
@betanalpha



**EJ Wagenmakers**  
@EJWagenmakers

<https://discourse.mc-stan.org/>

# Summary

- Computational modeling is never new → don't let it fear you!
- Learn some statistics (e.g., different statistical distributions)
- Learn some math (e.g., linear algebra)
- Learn some programming (e.g., R/Python/Julia/Matlab)
- Learn to seek external help (e.g., existing packages)
- Learn in pairs; practice makes perfect!



Richard McElreath  
@rlmcelreath



I say this a lot, bc I am also confused quite often.



Anna Jacobson @AnnaChingChing · Feb 21

"If you are confused, it is only because you are trying to understand." -  
@rlmcelreath in Statistical Rethinking

# Acknowledgement



[l.zhang.l3@bham.ac.uk](mailto:l.zhang.l3@bham.ac.uk)



<https://lei-zhang.net/>



[@lei\\_zhang\\_lz](https://twitter.com/lei_zhang_lz)



[@LeiZhang](https://www.youtube.com/@LeiZhang)



[@lei-zhang](https://github.com/lei-zhang)

Thank you!

ANY  
QUESTIONS?  
?

Happy Computing!