Advanced topics of hierarchical Bayesian computational modelling

Dr. Lei Zhang

Associate Professor

Adaptive **L**earning **P**sychology & **N**euroscience (ALPN) Lab

Centre for Human Brain Health, School of Psychology

University of Birmingham

lei-zhang.net

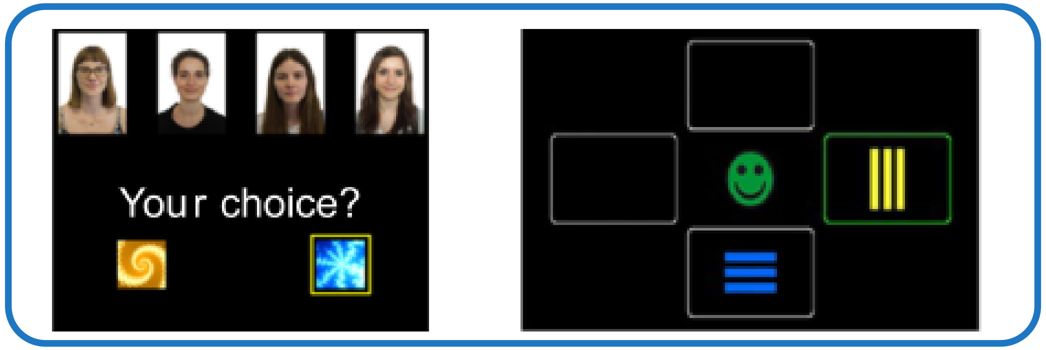
@lei_zhang_lz

HKU, 27.11.2025

My research:

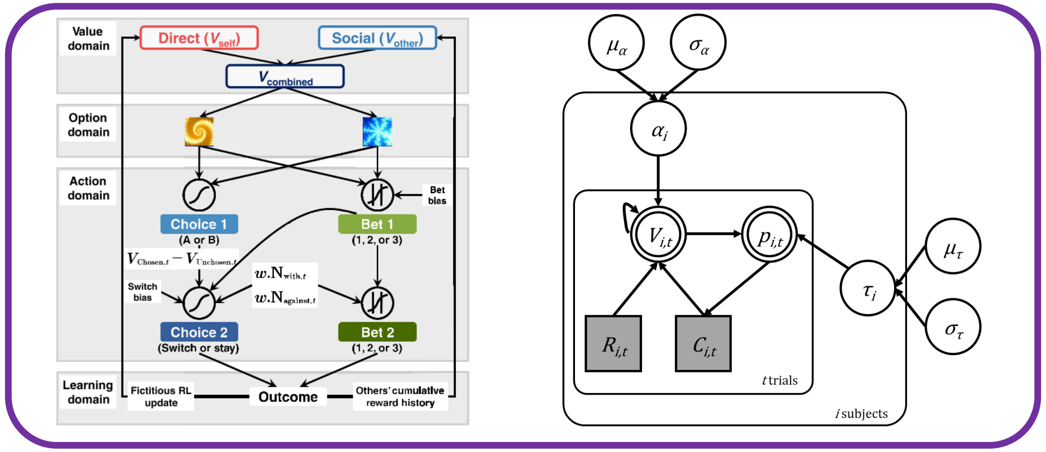
- I ask people to make decisions

Computation



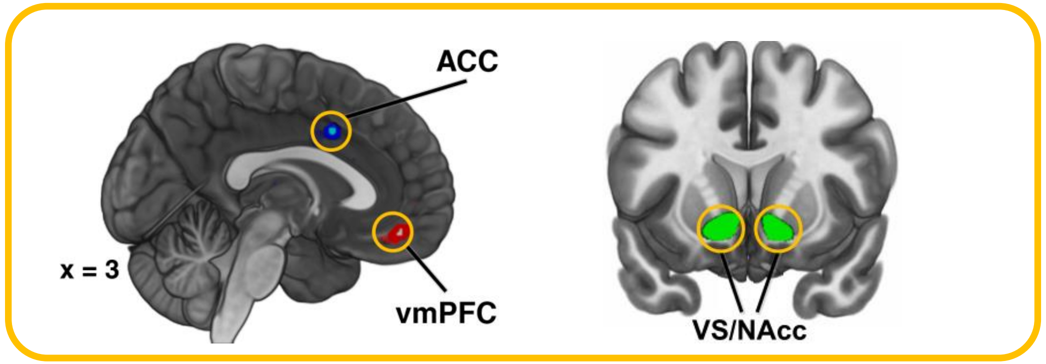
- I build computational models

Algorithm

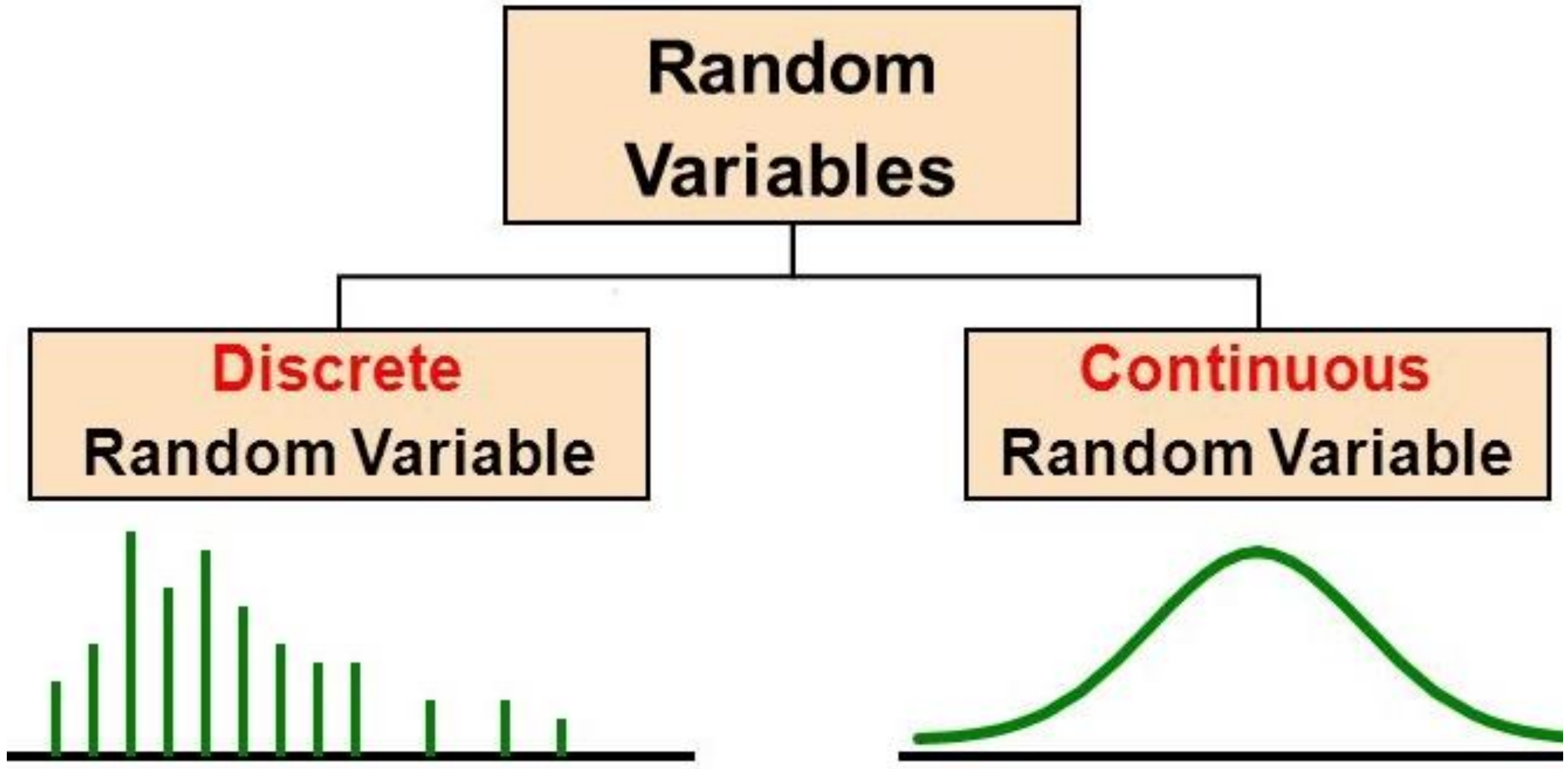


- I examine neural mechanisms

Implementation



Probability Functions



Bayes' theorem

$$p(A,B) = p(B,A)$$

$$p(A,B) = p(A|B)\,p(B)$$

$$p(B,A) = p(B|A)\,p(A)$$

$$p(A|B)\,p(B) = p(B|A)\,p(A)$$

$$\boxed{p\left(A \mid B\right) = \frac{p\left(B \mid A\right)p\left(A\right)}{p\left(B\right)}}$$

Linking Data and Parameter

θ $\quad\quad\quad$ D

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

Linking Data and Parameter

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

Linking Data and Parameter

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?

What if I have multiple parameters?

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

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

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

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

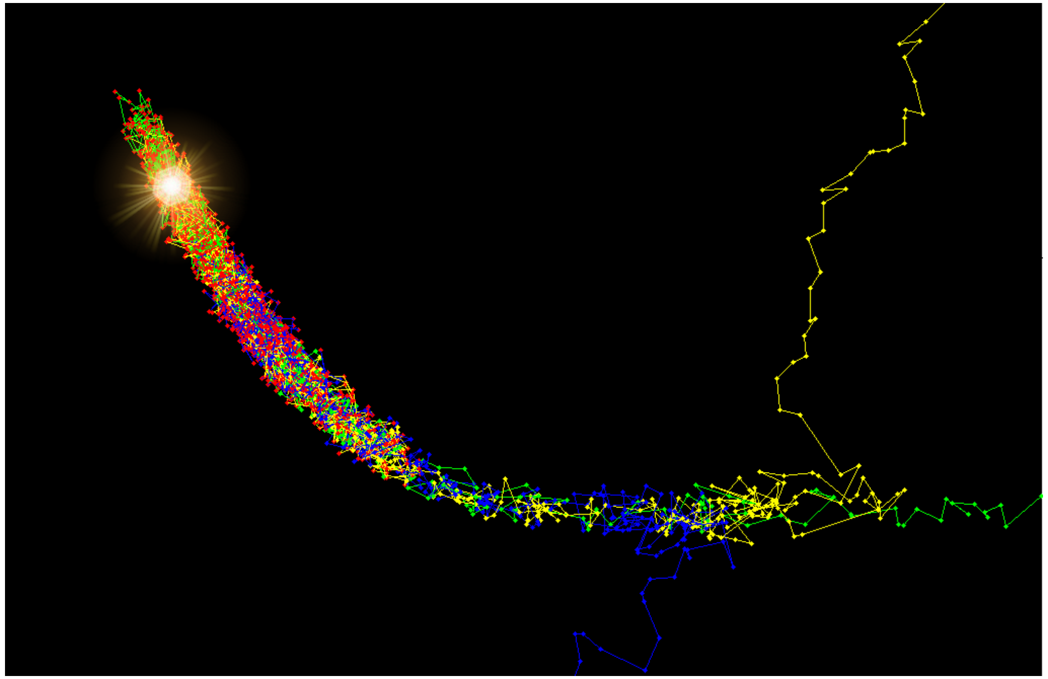
- Analytical solutions (often does not exist)
- Grid approximation (takes too long)

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

- solution: Markov Chain Monte Carlo

MCMC Sampling Algorithms

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

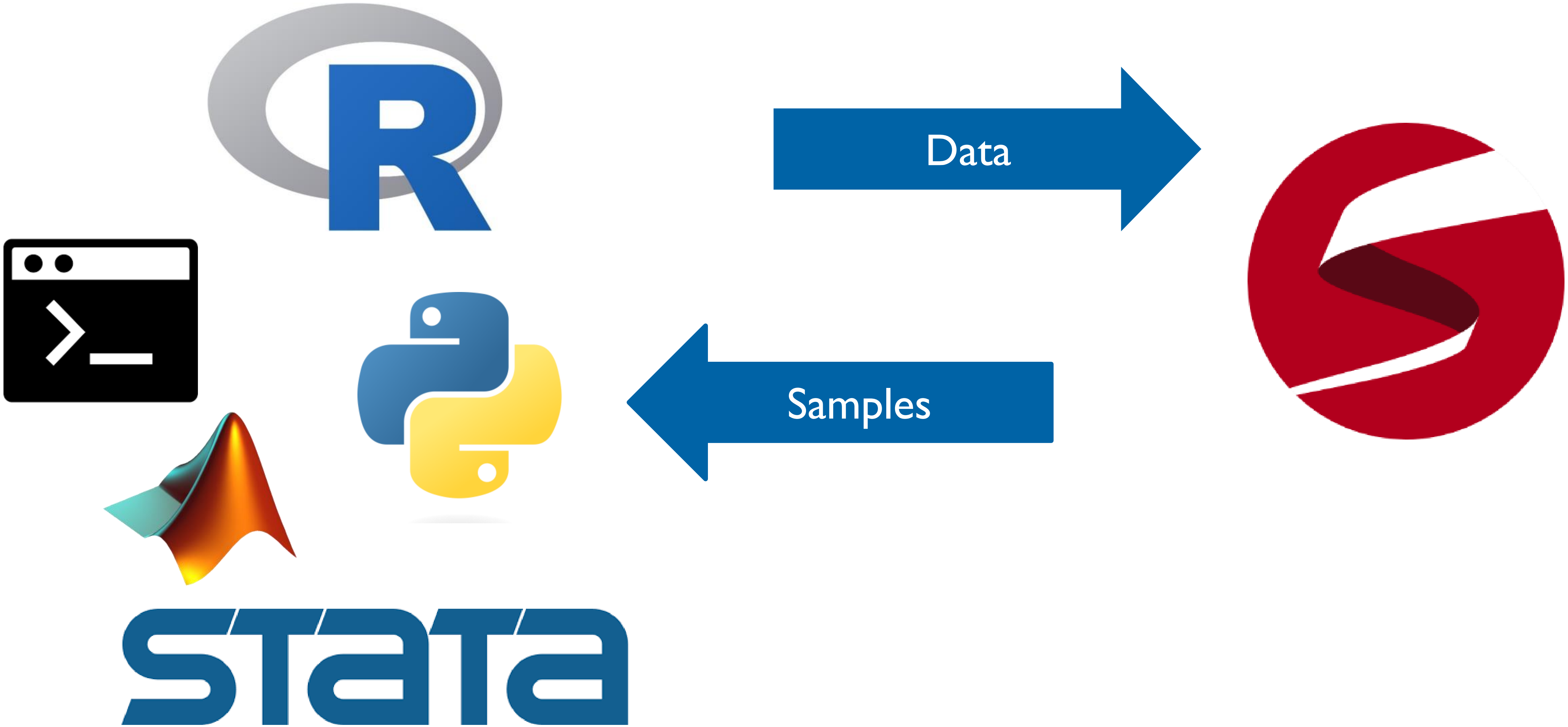


Stan!

*: also PyMC3

PyMC3

Stan and Other Platforms



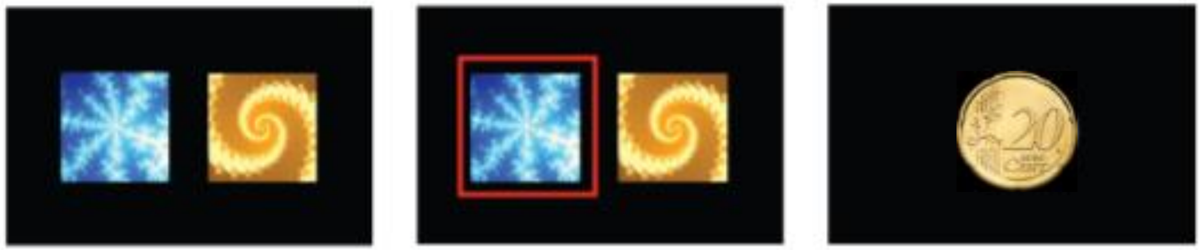
Stan Language

```
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 ...
}
```

reward contingency 80:20

Q-Learning

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

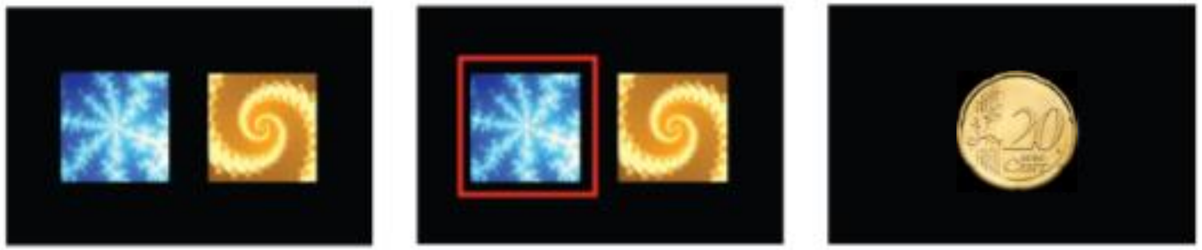


Value update: $V_t = V_{t-1} + \alpha * \mathrm{PE}_{t-1}$

Prediction error: $\mathrm{PE}_{t-1} = R_{t-1} - V_{t-1}$

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

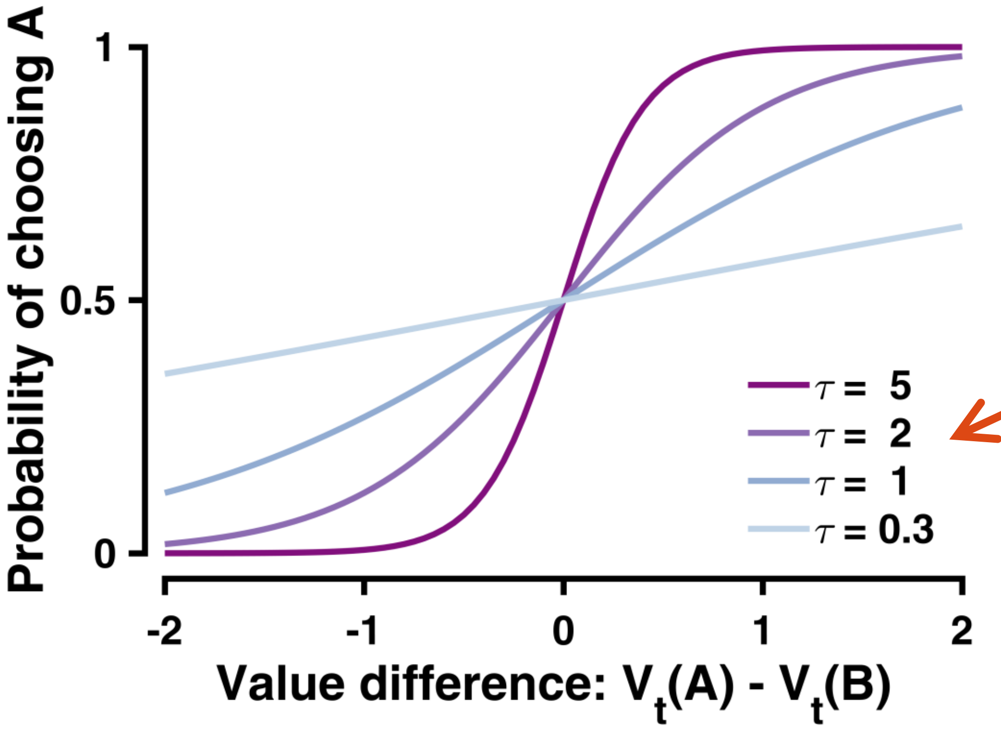
Choice rule: softmax



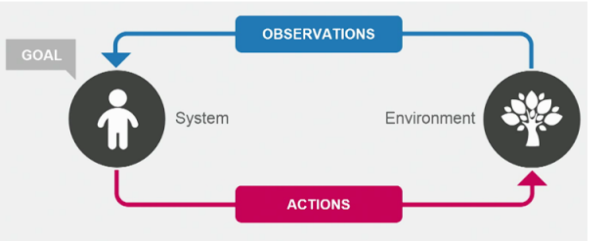
V()
V()

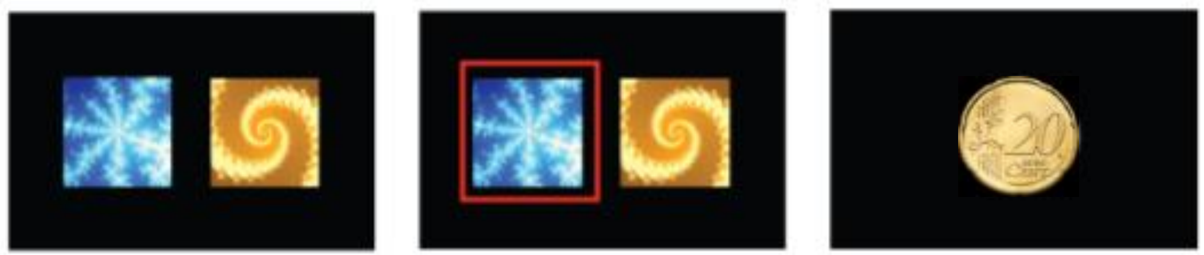
$$\boxed{p_t(A) = \frac{e^{\tau * V_t(A)}}{e^{\tau * V_t(A)} + e^{\tau * V_t(B)}}}$$

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



Q-Learning





Value update:

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

Prediction error:

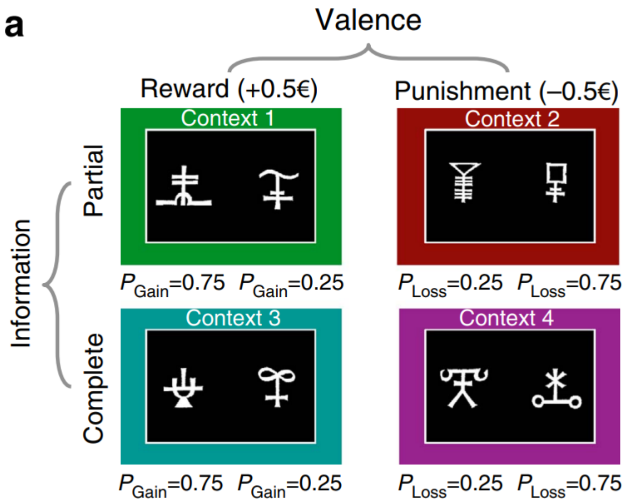
$$PE_t = R_t - V_t$$

choice rule (sigmoid /softmax):

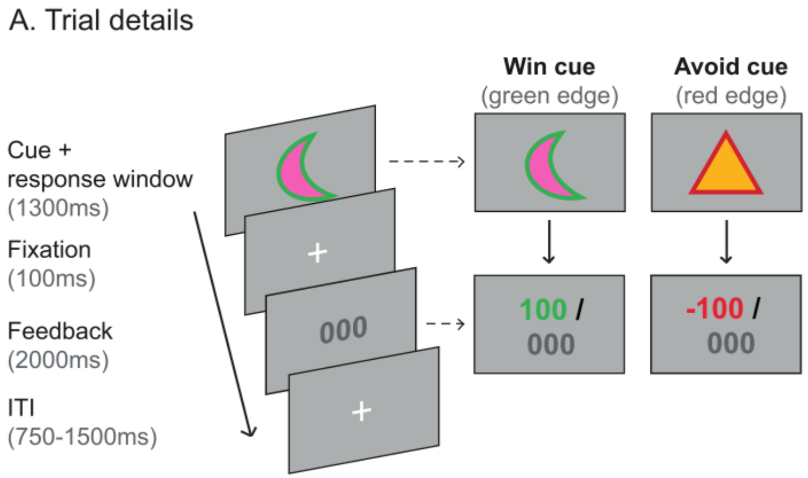
$$p(C{=}a) = \frac{1}{1+e^{\tau*(v(b)-v(a))}}$$

α - learning rate
PE - reward prediction error
V - value
R - reward
τ - softmax temperature

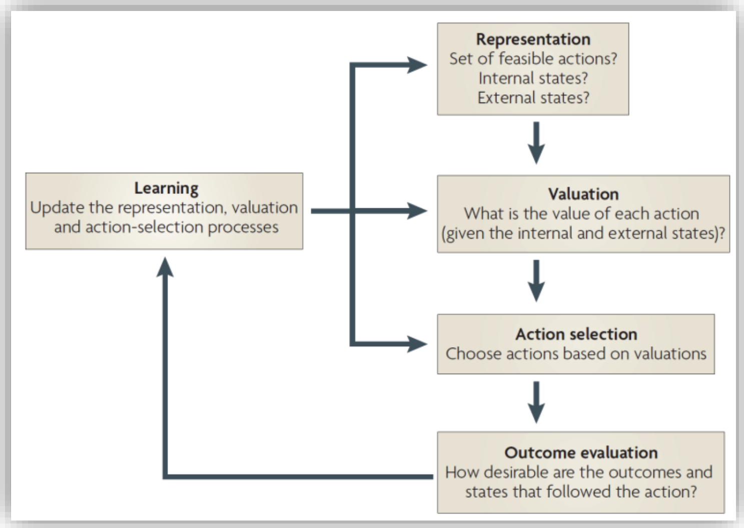
Generalizing RL framework

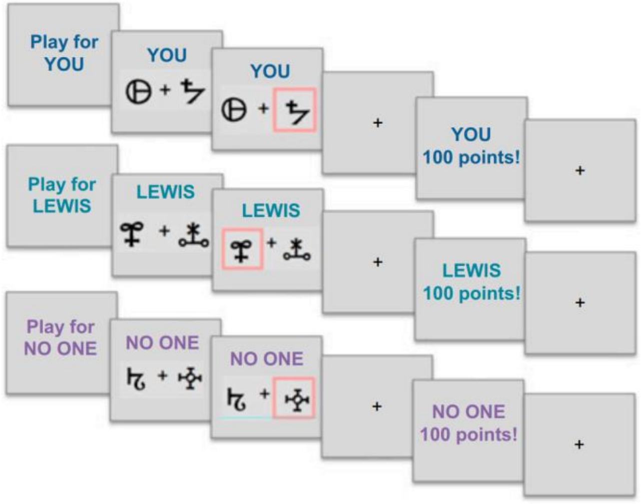


Palminteri et al. (2015)

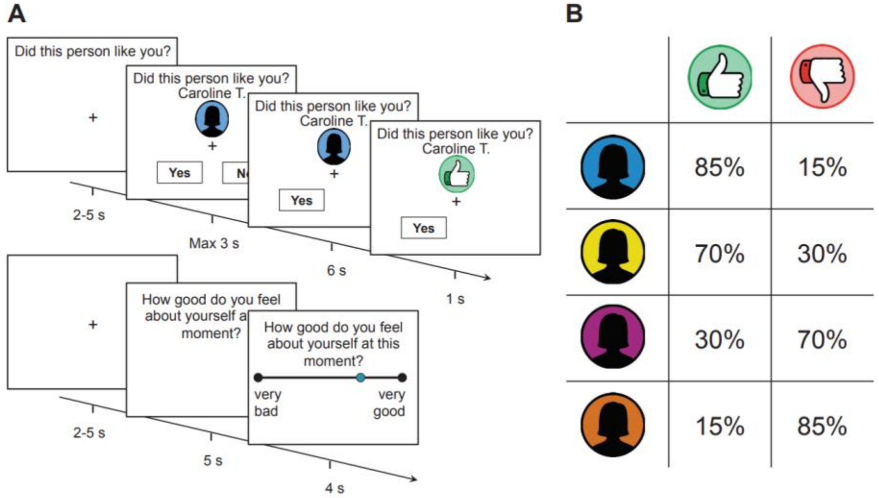


Swart et al. (2017)





Lockwood et al. (2016)



Will et al. (2017)

Some terminologies

simple Q-Learning

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

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

$$\mathrm{LL} = \mathrm{categorical\_lpmf}(\ C_t \mid p_t)$$

simple linear regression

$$\mu = \mathrm{a} + \mathrm{b}\ \mathrm{X}$$

$$\mathrm{LL} = \mathrm{normal\_lpdf}(\ \mathrm{Y} \mid \mu, \delta\ )$$

Some terminologies

simple Q-Learning

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

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

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

simple linear regression

$$\mu = a + b\,X$$

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

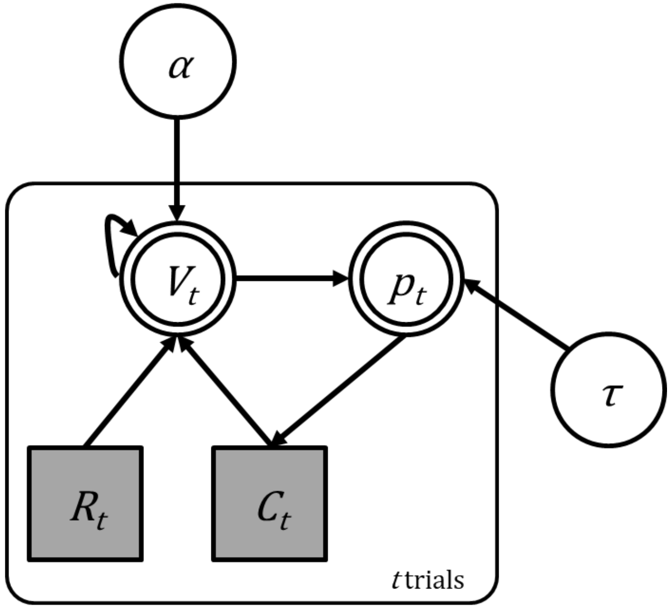
data / observation

free parameter / model parameter

internal/latent (model) variable

likelihood function

RL – Implementation



$$\alpha \sim Uniform(0,1)$$

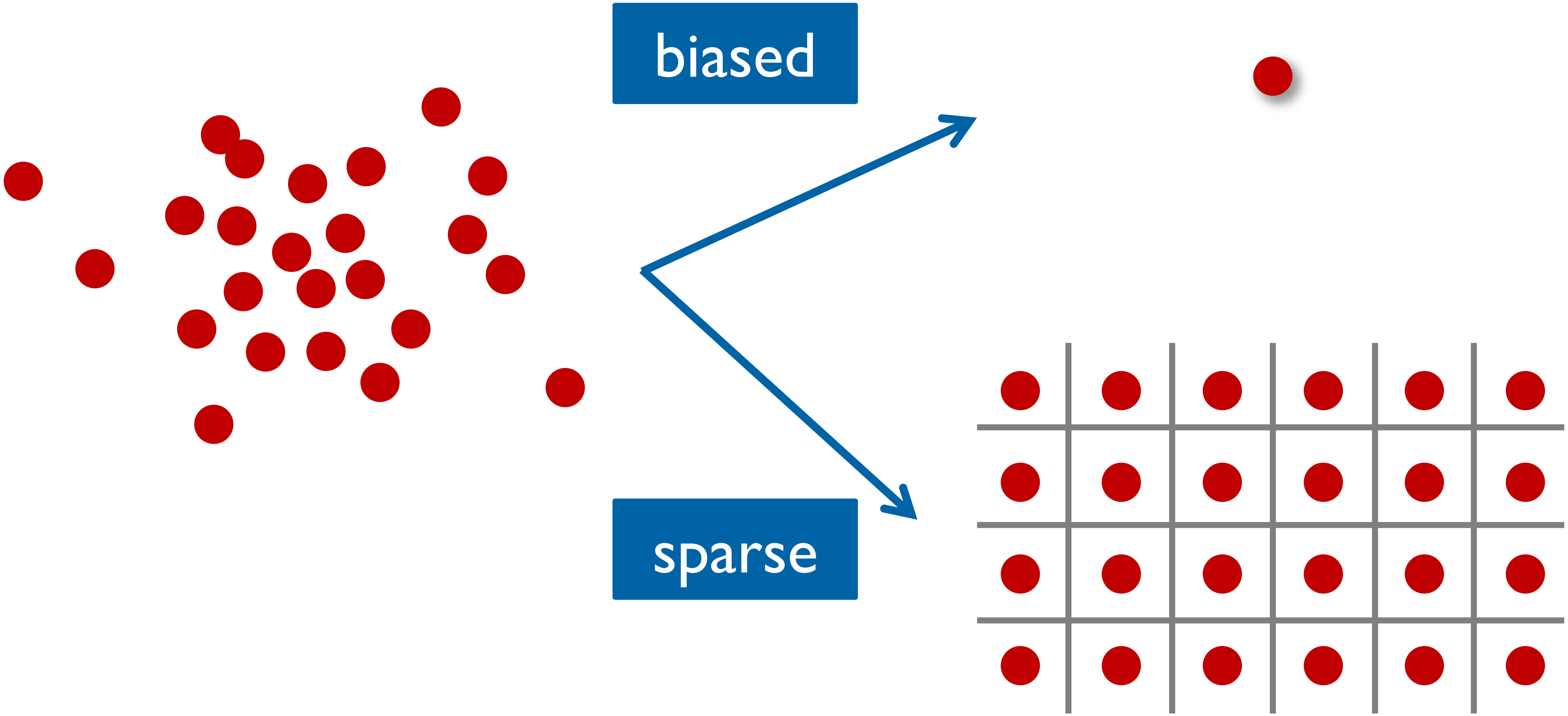
$$\tau \sim Uniform(0,3)$$

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

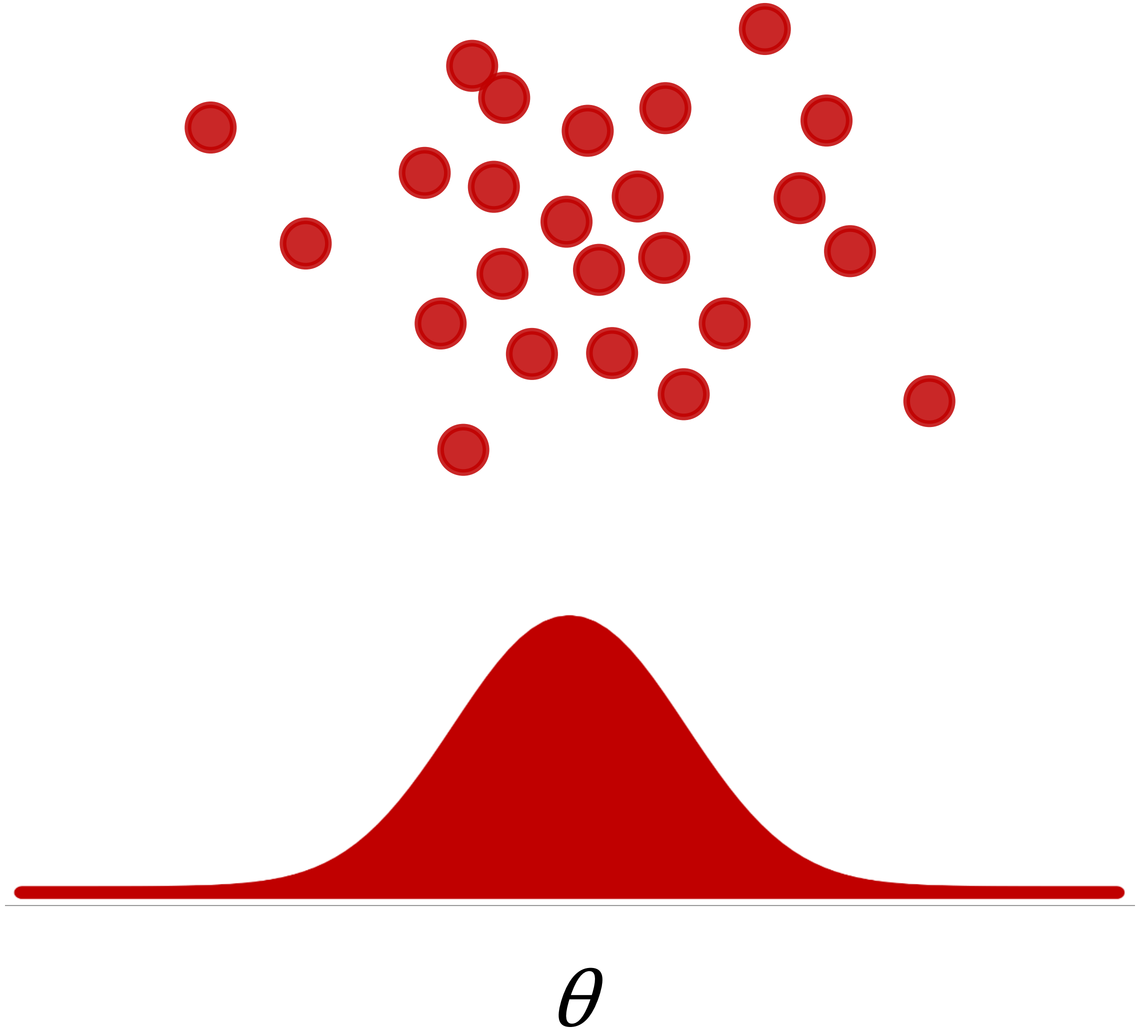
$$V_{t+1}^{c} = V_t^{C} + \alpha\left(R_t - V_t^{C}\right)$$

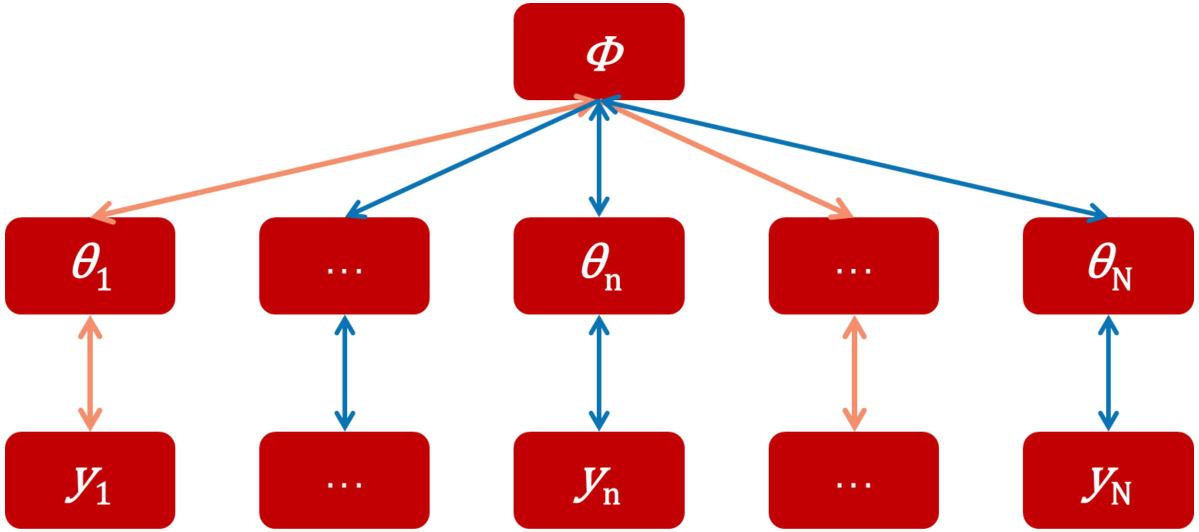
```
transformed data {
  vector[2] initV;
  initV = rep_vector(0.0, 2);
}

model {
  vector[2] v[nTrials+1];
  real pe[nTrials];

  v[1] = initV;

  for (t in 1:nTrials) {
    choice[t] ~ categorical_logit( tau * v[t] );

    pe[t] = reward[t] - v[t,choice[t]];

    v[t+1] = v[t];
    v[t+1, choice[t]] = v[t, choice[t]] + lr * pe[t];
  }
}
```

Fitting Multiple Participants

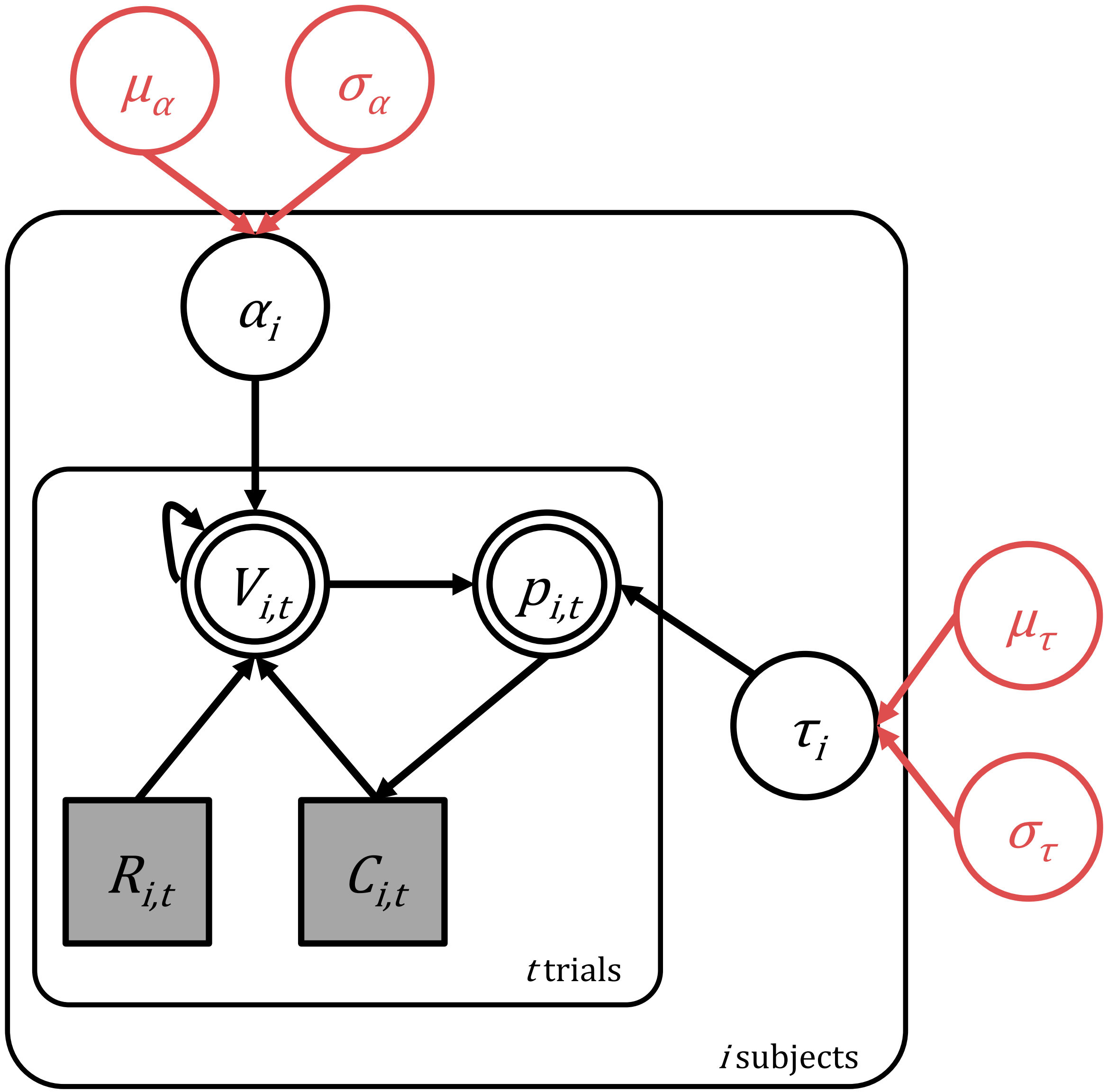


Fitting Multiple Participants with hierarchical Bayesian analysis (HBA)

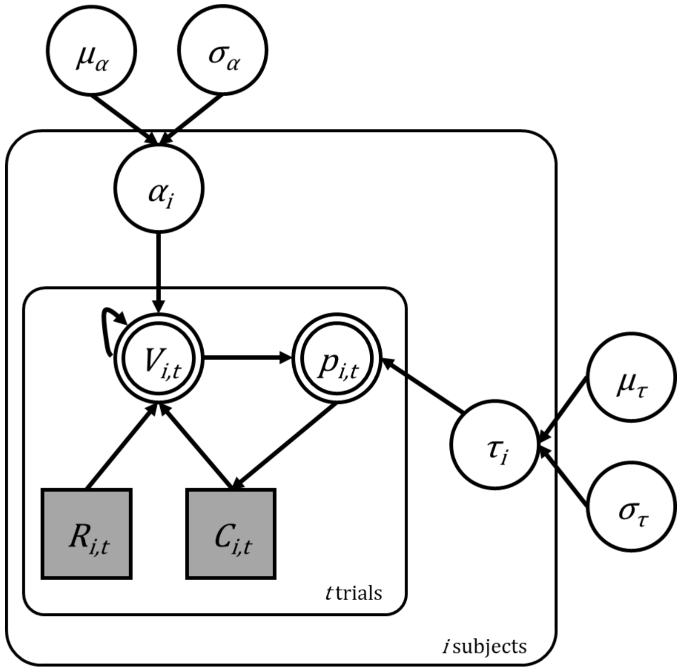




Hierarchical RL Model



Implementing Hierarchical RL Model



$$\mu_\alpha \sim Uniform(0,1)$$
$$\sigma_\alpha \sim halfCauchy(0,1)$$
$$\mu_\tau \sim Uniform(0,3)$$
$$\sigma_\tau \sim halfCauchy(0,3)$$
$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)_{\mathcal{T}(0,1)}$$
$$\tau_i \sim Normal(\mu_\tau, \sigma_\tau)_{\mathcal{T}(0,3)}$$

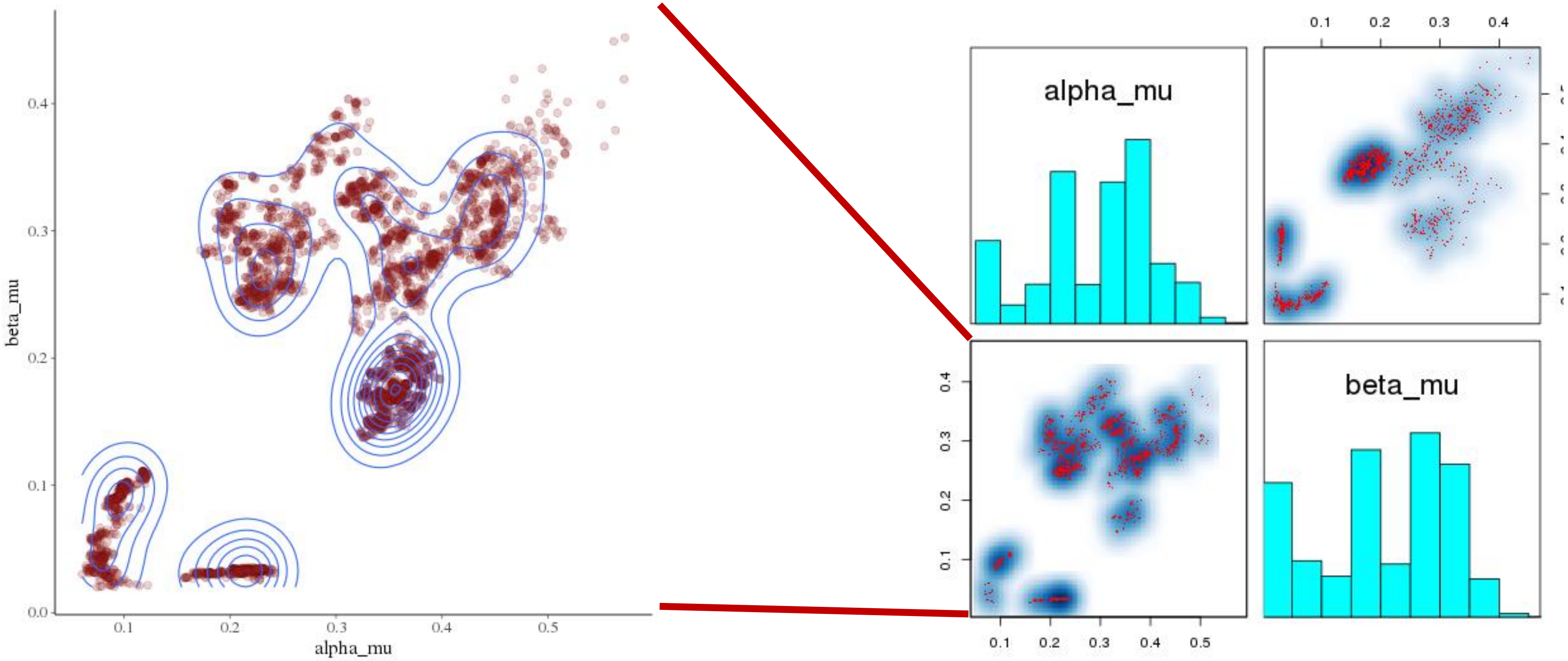
$$p_{i,t}(C=A) = \frac{1}{1+e^{\tau_i(V_{i,t}(B)-V_{i,t}(A))}}$$

$$V^c_{i,t+1} = V^C_{i,t} + \alpha_i(R_{i,t} - V^C_{i,t})$$

```
parameters {
  real<lower=0,upper=1> lr_mu;
  real<lower=0,upper=3> tau_mu;

  real<lower=0> lr_sd;
  real<lower=0> tau_sd;

  real<lower=0,upper=1> lr[nSubjects];
  real<lower=0,upper=3> tau[nSubjects];
}

model {
  lr_sd  ~ cauchy(0,1);
  tau_sd ~ cauchy(0,3);
  lr     ~ normal(lr_mu, lr_sd) ;
  tau    ~ normal(tau_mu, tau_sd) ;

  for (s in 1:nSubjects) {
    vector[2] v;
    real pe;
    v = initV;

    for (t in 1:nTrials) {
      choice[s,t] ~ categorical_logit( tau[s] * v );
      pe = reward[s,t] - v[choice[s,t]];
      v[choice[s,t]] = v[choice[s,t]] + lr[s] * pe;
    }
  }
}
```

Satisfied with the results?

```
Warning messages:
1: There were 3998 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help. See
http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
2: Examine the pairs() plot to diagnose sampling problems
```



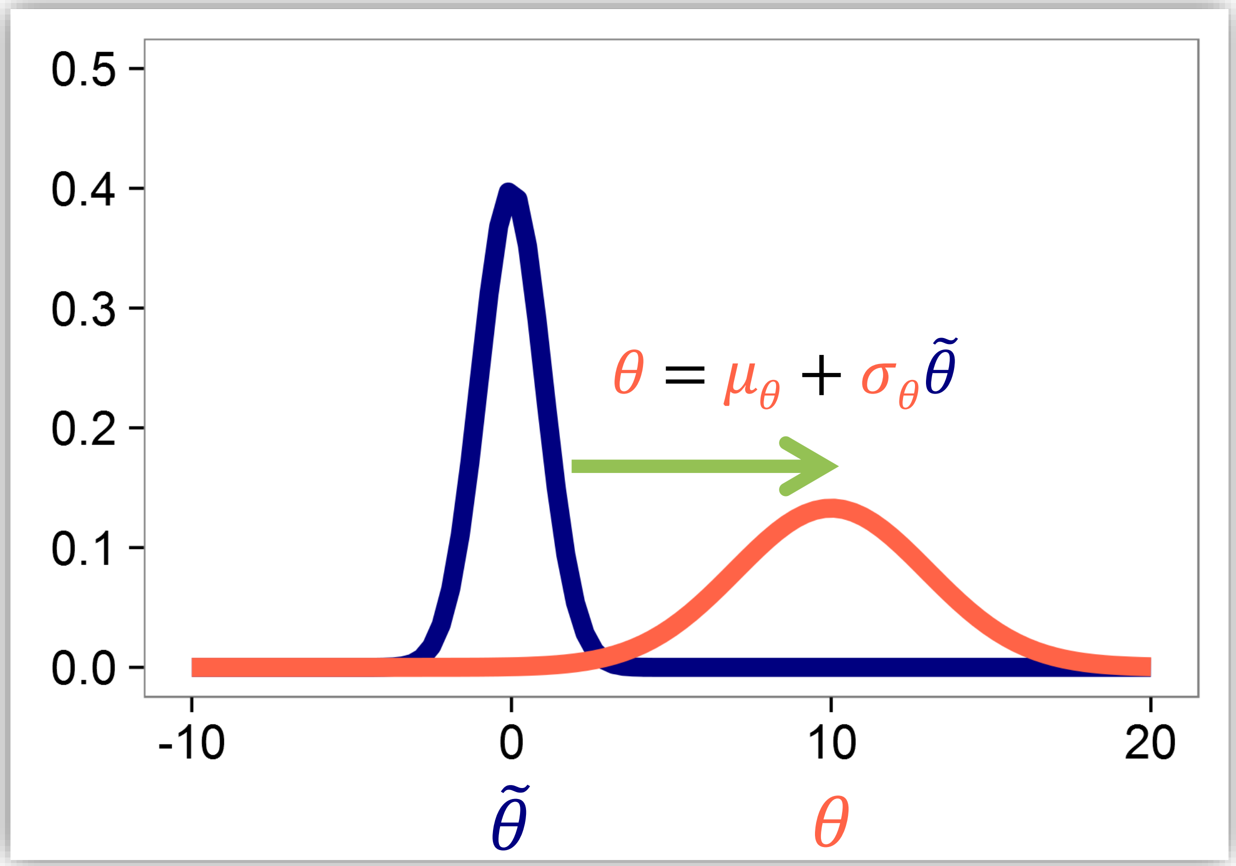
Non-centered Reparameterization*

$$\theta \sim Normal(\mu_\theta, \sigma_\theta)$$

$$\tilde{\theta} \sim Normal(0, 1)$$

$$\theta = \mu_\theta + \sigma_\theta \tilde{\theta}$$

Stan likes **simple** distributions!



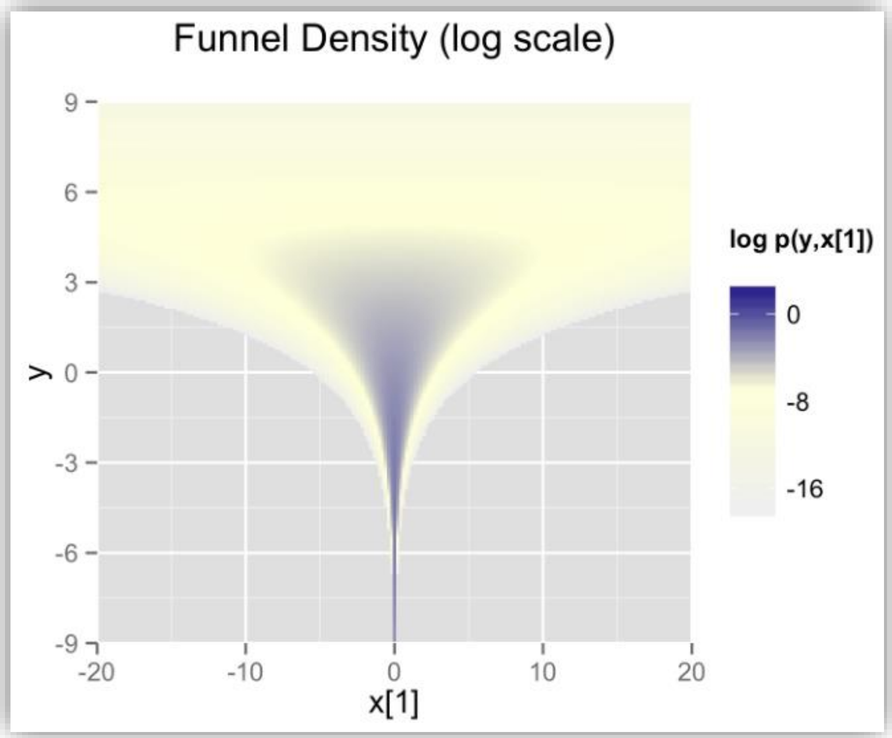
*:Stan Development Team (2016), aka, Matt Trick

Reparameterization

Neal's Funnel

$$p(y,x) = \text{Normal}(y|0,3) \times \prod_{n=1}^{9} \text{Normal}(x_n|0,\exp(y/2))$$

```
parameters {
  real y;
  vector[9] x;
}
model {
  y ~ normal(0,3);
  x ~ normal(0,exp(y/2));
}
```

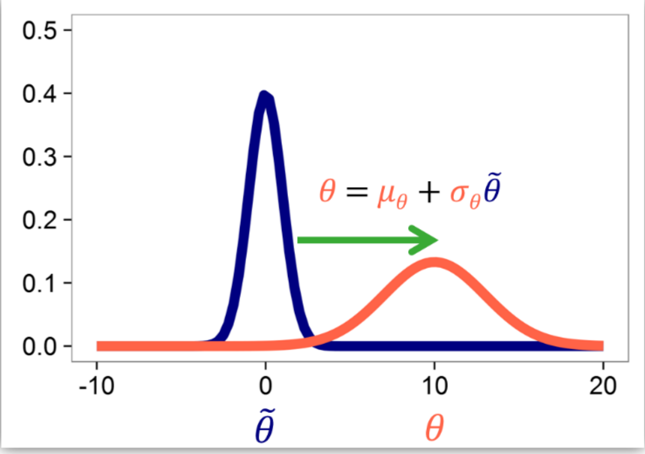


Reparameterization

Neal's Funnel

$$p(y, x) = \text{Normal}(y|0,3) \times \prod_{n=1}^{9} \text{Normal}(x_n|0, \exp(y/2))$$

```
parameters {
  real y;
  vector[9] x;
}
model {
  y ~ normal(0,3);
  x ~ normal(0,exp(y/2));
}
```




```
parameters {
  real y_raw;
  vector[9] x_raw;
}
transformed parameters {
  real y;
  vector[9] x;

  y = 3.0 * y_raw;
  x = exp(y/2) * x_raw;
}
model {
  y_raw ~ normal(0,1);
  x_raw ~ normal(0,1);
}
```

Stan Sampling Parameters

parameter	description	constraint	default
iterations	number of MCMC samples (per chain)	int, > 0	2000
delta: δ	target Metropolis acceptance rate	$\delta \in [0,1]$	0.80
stepsize: ε	initial HMC step size	real, $\varepsilon > 0$	2.0
max_treedepth: L	maximum HMC steps per iteration	int, $L > 0$	10

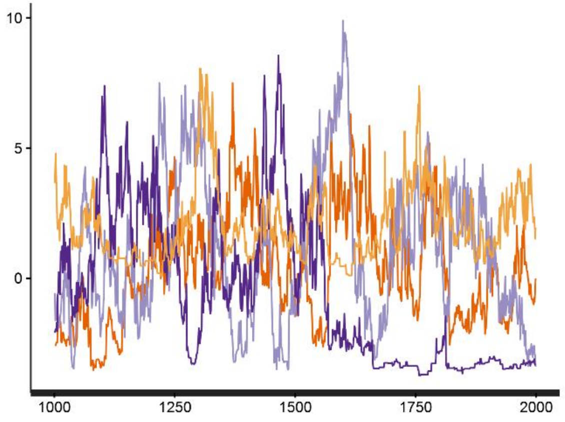
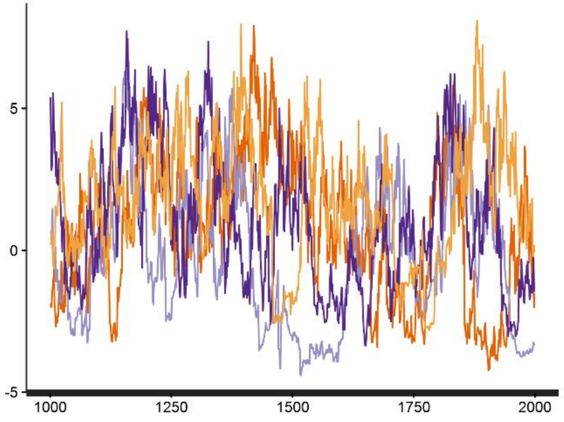
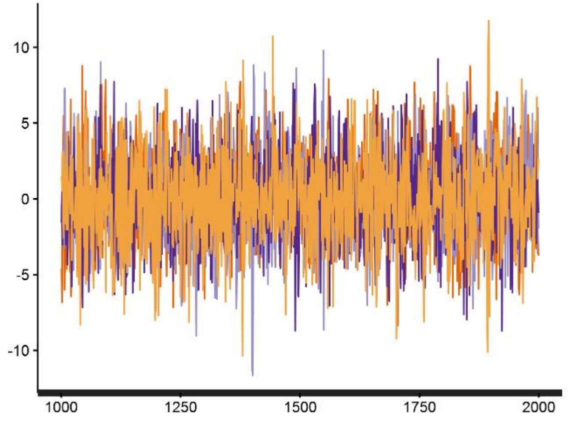
Typical adjustments

- Increase iterations
- Increase delta
- Decrease stepsize
- Might have to increase max_treedepth

```
funnel_fit2 <- stan("_scripts/funnel.stan",
        iter = 4000,
        control = list(adapt_delta = 0.999,
                       stepsize = 1.0,
                       max_treedepth = 20))
```

Stan Development Team (2016)

Neal's Funnel: Comparing Performance

	direct model	adjusted direct model	reparameterized model
Rhat (*y*)	1.22	1.1	1.0
n_eff (*y*)	18	42	3886
runtime*	48.50 sec	50.76 sec	50.12 sec
n_eff (*y*) / runtime	0.37 / sec	0.82 / sec	77.53 / sec
n_divergent	53	0	0
traceplot (*y*)			

*: 2 cores in parallel, including compiling time

How about **Bounded** Parameters?

$\tilde{\theta} \sim Normal(0,1)$

$\theta = \mu_\theta + \sigma_\theta \tilde{\theta}$

$\theta \in (-\infty, +\infty)$

$\tilde{\theta} \sim Normal(0,1)$

$\theta = Probit^{-1}(\mu_\theta + \sigma_\theta \tilde{\theta})$

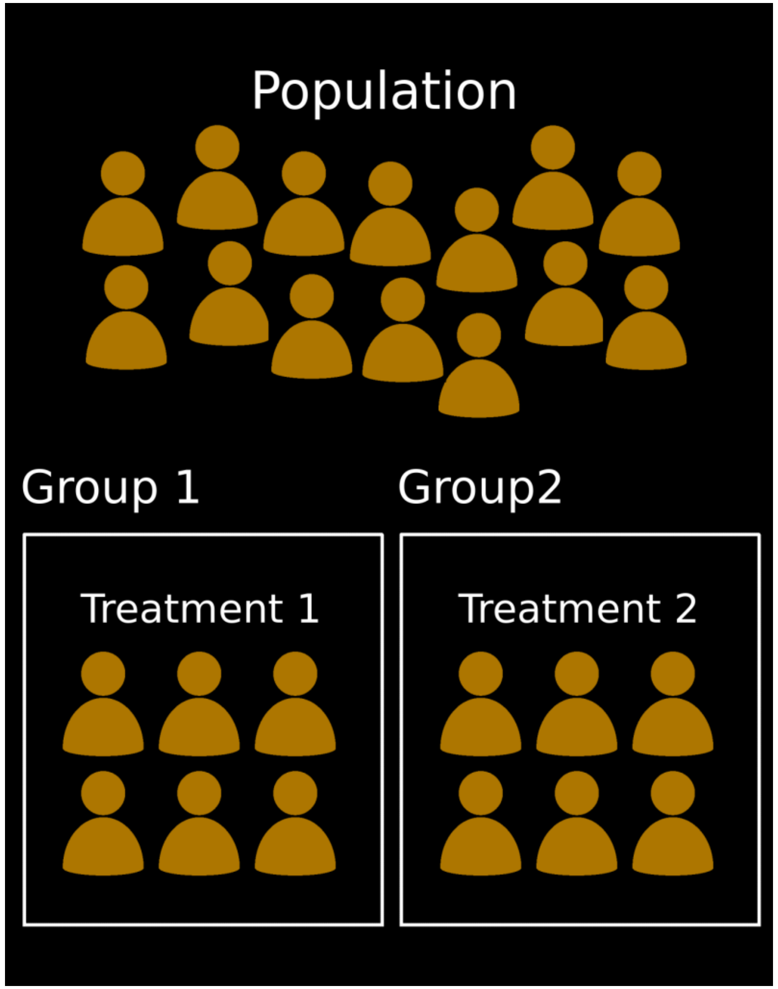
$\theta \in [0, 1]$

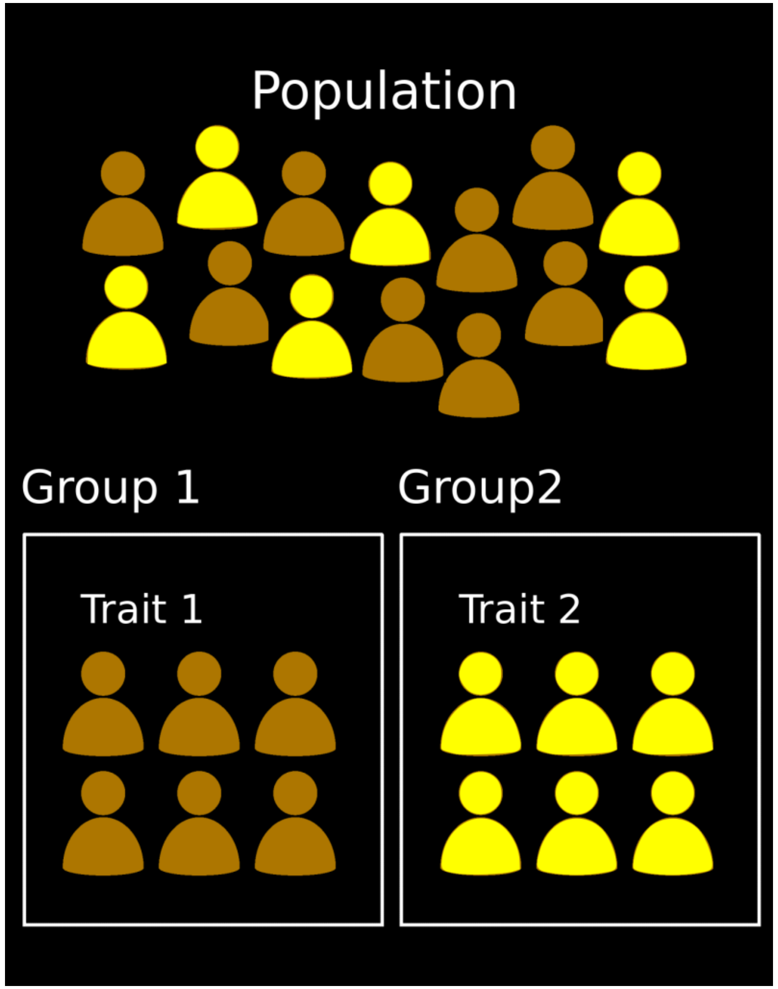
constraint	reparameterization
$\theta \in (-\infty, +\infty)$	$\theta = \mu_\theta + \sigma_\theta \tilde{\theta}$
$\theta \in [0, N]$	$\theta = Probit^{-1}(\mu_\theta + \sigma_\theta \tilde{\theta})$ x N
$\theta \in [M, N]$	$\theta = Probit^{-1}(\mu_\theta + \sigma_\theta \tilde{\theta})$ x $(N–M) + M$
$\theta \in (0, +\infty)$	$\theta = exp(\mu_\theta + \sigma_\theta \tilde{\theta})$; $\theta = log(1+exp(\mu_\theta + \sigma_\theta \tilde{\theta}))$

* $Probit^{-1}$: Normal cumulative distribution function (normcdf)

Between-subject design

- each participant is tested in only one condition
- e.g., mood → IQ performance in the previous example



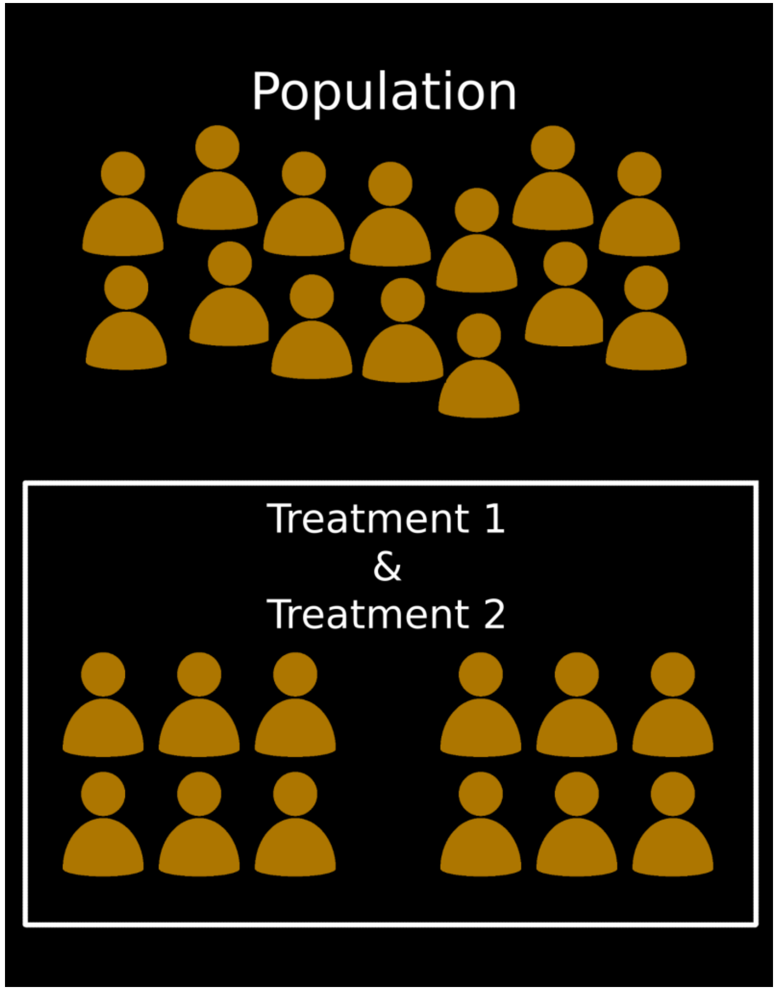


$$\theta_{\text{patient},i} \sim Normal(\mu_{\theta,\text{patient}}, \sigma_{\theta,\text{patient}})$$
$$\theta_{\text{control},i} \sim Normal(\mu_{\theta,\text{control}}, \sigma_{\theta,\text{control}})$$
,

$$\mu_{\theta,\text{patient}} \sim Normal(\mu_{\theta}, \sigma_{\theta})$$
$$\mu_{\theta,\text{control}} \sim Normal(\mu_{\theta}, \sigma_{\theta})$$
$$\theta_{\text{patient},i} \sim Normal(\mu_{\theta,\text{patient}}, \sigma_{\theta,\text{patient}})$$
$$\theta_{\text{control},i} \sim Normal(\mu_{\theta,\text{control}}, \sigma_{\theta,\text{control}})$$

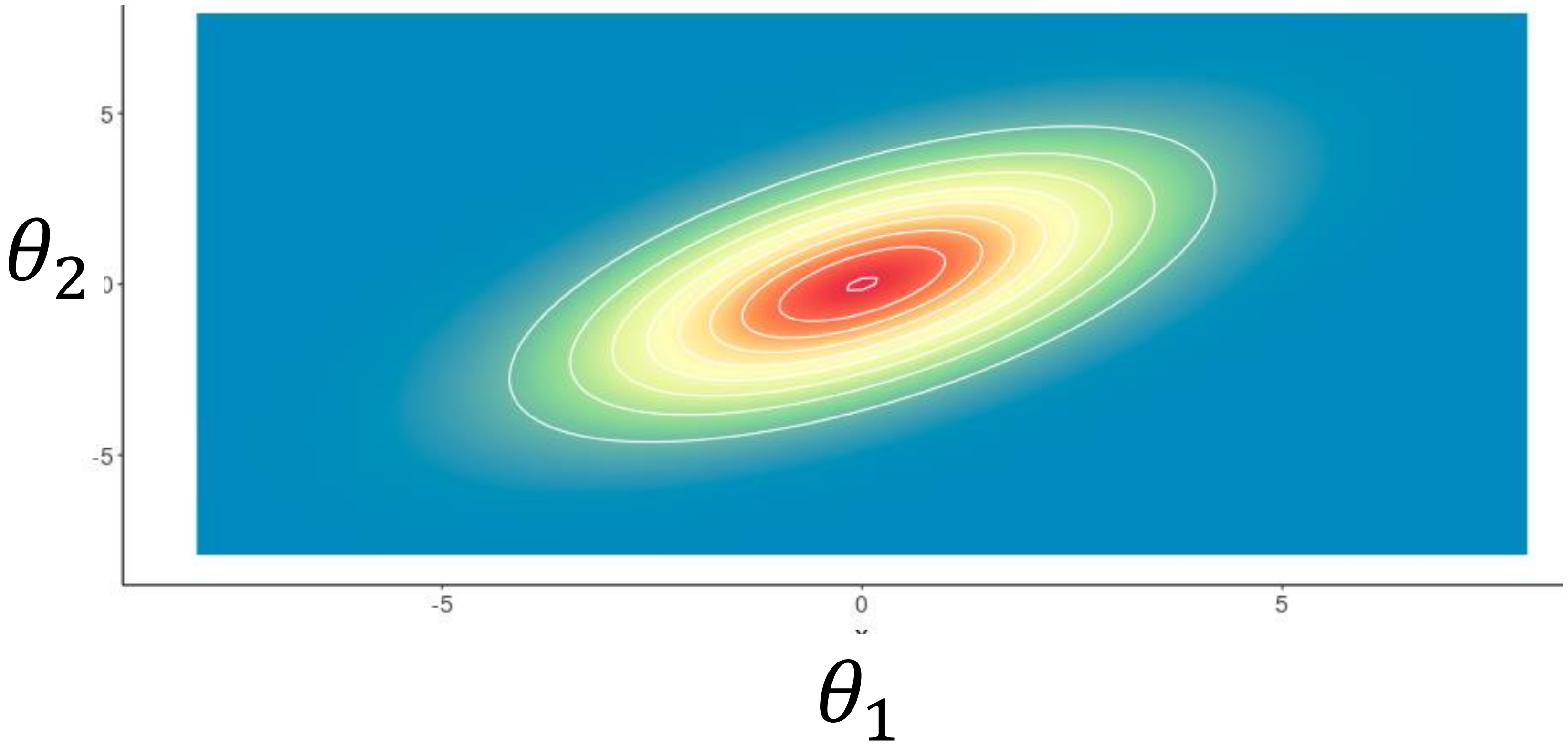
Within-subject design

- each participant is tested under all conditions
- e.g., learning performance at the beginning vs. at the end of the semester



Effect coding

$$\theta_2 = \theta_1 + \Delta\theta$$



Effect coding

	θ_1	$\theta_2 - \theta_1$
θ_1	1	0
θ_2	1	1

Effect coding

	θ_1	$\theta_2 - \theta_1$	$\theta_3 - \theta_1$
θ_1	1	0	0
θ_2	1	1	0
θ_3	1	0	1

Effect coding

$$\theta_i \sim Normal\left(\mu, \Sigma\right)$$

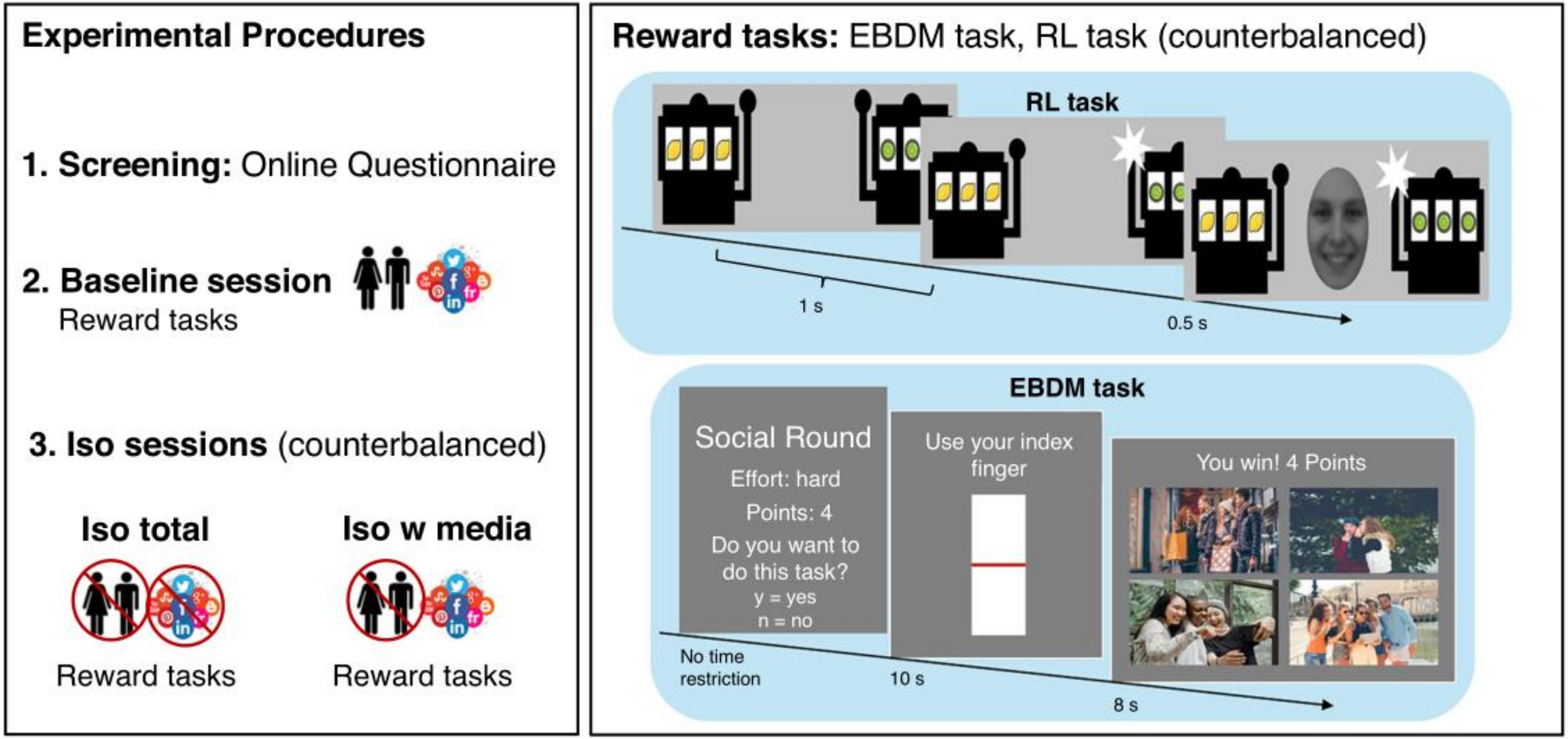
$$\mu = (\mu_1, \mu_2, \mu_3) = \begin{pmatrix} 1\,0\,0 \\ 1\,1\,0 \\ 1\,0\,1 \end{pmatrix} (\mu_1, \mu_2 - \mu_1, \mu_3 - \mu_1)^{\mathrm{T}}$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 \end{pmatrix} = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \Omega \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix}$$

Multiple IV: Factorial design

		Factor A (TV Violence)	
		Level 1 (A1) (Violent)	**Level 2 (A2) (Nonviolent)**
Factor B (Realism)	**Level 1 (B1) (Real)**	A1B1 (Real/Violent)	A2B1 (Real/Nonviolent)
	Level 2 (B2) (Cartoon)	A1B2 (Cartoon/Violent)	A2B2 (Cartoon/Nonviolent)

Lammers & Badia (2013)

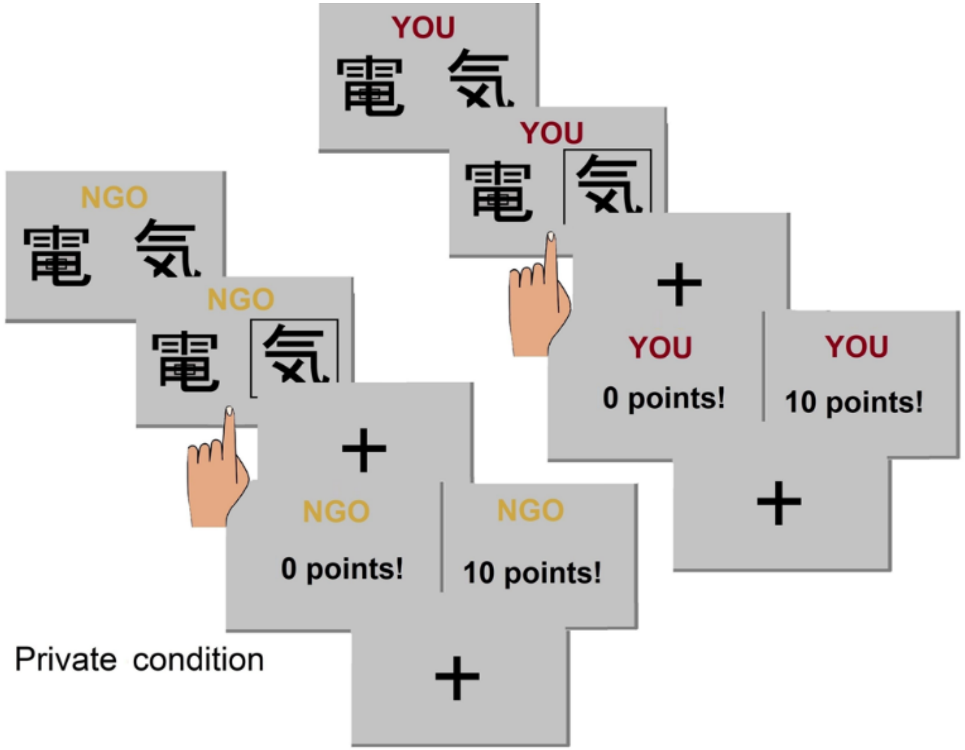


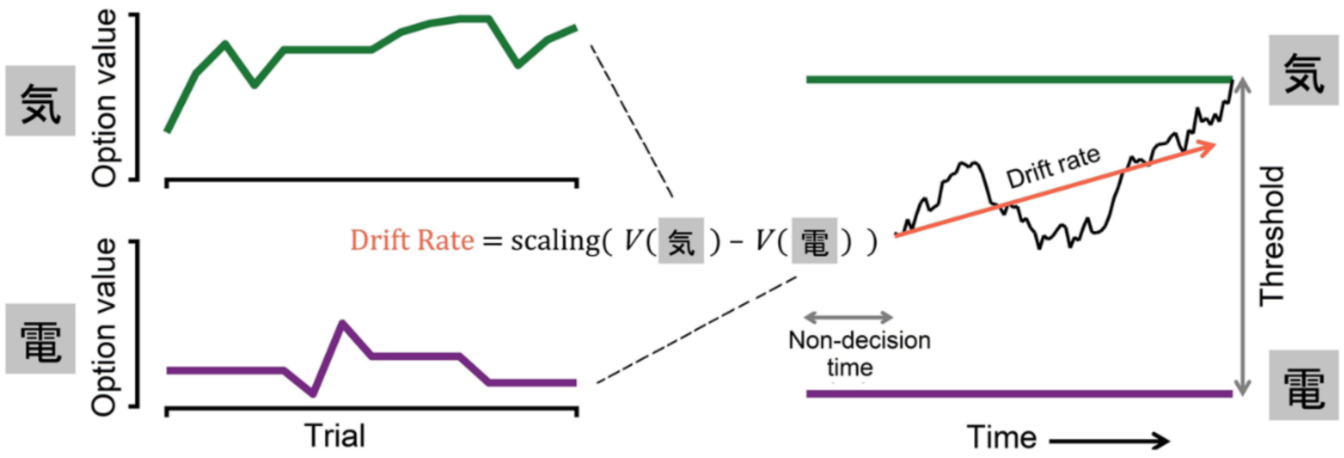
Tomova, …**Zhang,** et al., *Communications Psychology* (2025)

$$
\begin{array}{cc}
 & \begin{array}{cccccc} \text{baseline} & \text{iso w media} & \text{iso total} & \text{baseline} & \text{iso w media} & \text{iso total} \end{array} \\
\begin{array}{c} \\ \text{non-social} \\ \\ \\ \text{social} \\ \\ \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}
\end{array}
$$

Tomova, …**Zhang,** et al., *Communications Psychology* (2025)

RLDDM



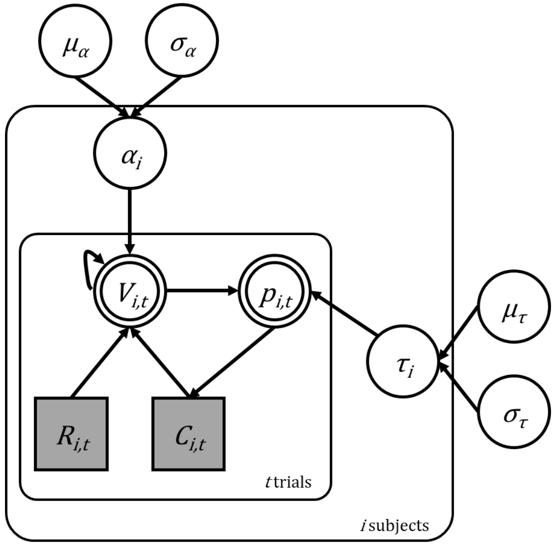


$$v_t = S\left[v_{\text{scaling}}\left(V_{\text{correct},t} - V_{\text{incorrect},t}\right)\right],$$

$$S(x) = 2 \cdot \frac{v_{\max}}{1+e^{-x}} - v_{\max}$$

Kutlikova* & **Zhang***, et al., *Neuropsychopharmacology* (2023)

HBA sounds good, but...



$$\mu_\alpha \sim Uniform(0,1)$$

$$\sigma_\alpha \sim halfCauchy(0,1)$$

$$\mu_\tau \sim Uniform(0,3)$$

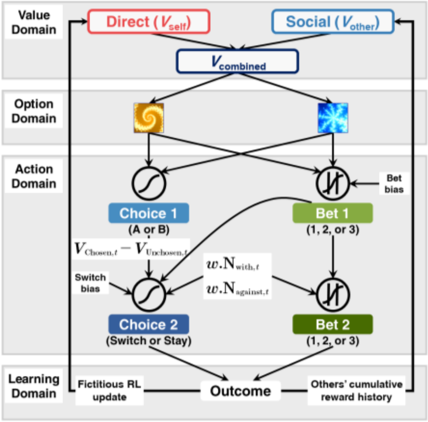
$$\sigma_\tau \sim halfCauchy(0,3)$$

$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)_{\mathcal{T}(0,1)}$$

$$\tau_i \sim Normal(\mu_\tau, \sigma_\tau)_{\mathcal{T}(0,3)}$$

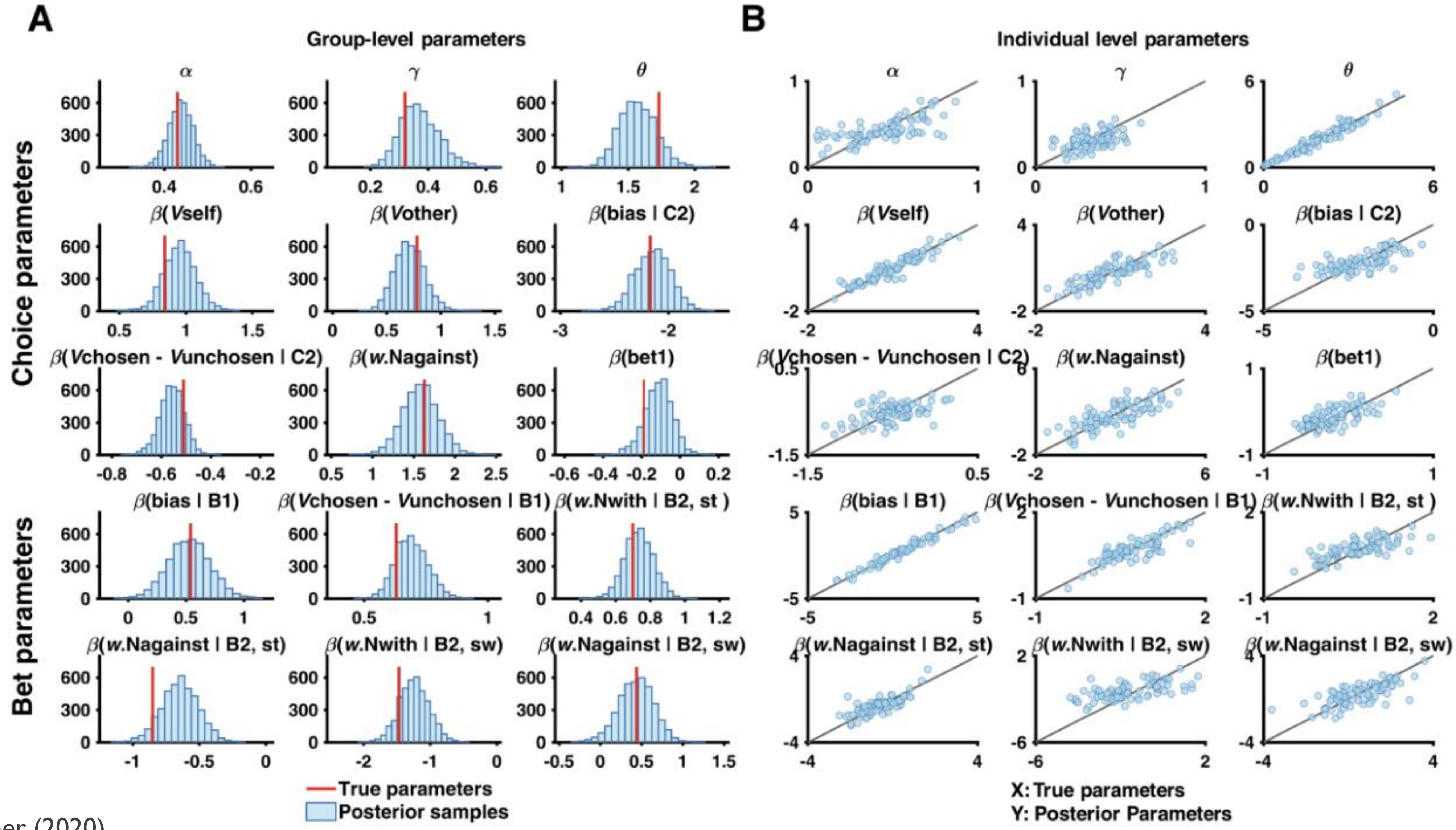
$$p_{i,t}(C = A) = \frac{1}{1 + e^{\tau_i (V_{i,t}(B) - V_{i,t}(A))}}$$

$$V^c_{i,t+1} = V^C_{i,t} + \alpha_i (R_{i,t} - V^C_{i,t})$$



$$
\begin{aligned}
\mathbb{V}_{self,t} &= [V_{self,t}(\mathrm{A}), V_{self,t}(\mathrm{B})] \\
\mathbb{V}_{other,t} &= [V_{other,t}(\mathrm{A}), V_{other,t}(\mathrm{B})] \\
\mathbb{V}_t &= \beta_{vself} \mathbb{V}_{self,t} + \beta_{vother} \mathbb{V}_{other,t} \\
C1_t &\sim Categorical(Softmax(\mathbb{V}_t)) \\
U_{bet1,t} &= \beta_{bias c1} + \beta_{vdiff c1} (V_{chosen,C1,t} - V_{unchosen,C1,t}) \\
B1_t &\sim OrderedLogistic(U_{bet1,t} \mid \theta) \\
w.\mathrm{N}_{against,t} &= \frac{\sum_{s=1}^{K} w_{s,t}}{\sum_{s=1}^{4} w_{s,t}}, K = 0, 1, \ldots, 4 \\
w.\mathrm{N}_{with,t} &= \frac{\sum_{s=1}^{4-K} w_{s,t}}{\sum_{s=1}^{4} w_{s,t}} \\
V_t(\mathrm{switch}) &= \beta_{bias c2} + \beta_{vdiff c2} (V_{chosen,C1,t} - V_{unchosen,C1,t}) + \beta_{against} w.\mathrm{N}_{against,t} \\
C2 &\sim Bernoulli(V_t(\mathrm{switch})) \\
U_{bet2,t} &= \begin{cases} U_{bet1,t} + \beta_{with_{agr}} w.\mathrm{N}_{with,t} + \beta_{against_{agr}} w.\mathrm{N}_{against,t} & \text{, if } C1 = C2 \\ U_{bet1,t} + \beta_{with_{switch}} w.\mathrm{N}_{with,t} + \beta_{against_{switch}} w.\mathrm{N}_{against,t}, & \text{if } C1 \neq C2 \end{cases} \\
B2_t &\sim OrderedLogistic(U_{bet2,t} \mid \theta) \\
\Phi(x) &= \frac{1}{1 + e^{-x}} \\
\delta_{self,chosen,C2,t} &= R_{self,t} - V_{self,chosen,C2,t} \\
\delta_{self,unchosen,C2,t} &= -R_{self,t} - V_{self,unchosen,C2,t} \\
V_{self,chosen,C2,t+1} &= V_{self,chosen,C2,t} + \alpha \delta_{self,chosen,C2,t} \\
V_{self,unchosen,C2,t+1} &= V_{self,unchosen,C2,t} + \alpha \delta_{self,unchosen,C2,t}
\end{aligned}
$$

Parameter recovery:
are parameters identifiable?



Zhang & Gläscher (2020)

hBayesDM package

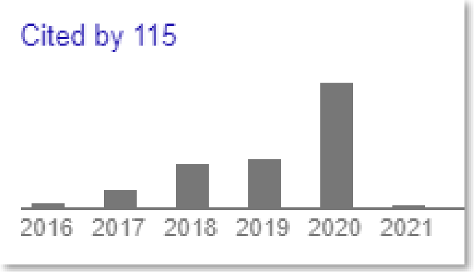
hBayesDM

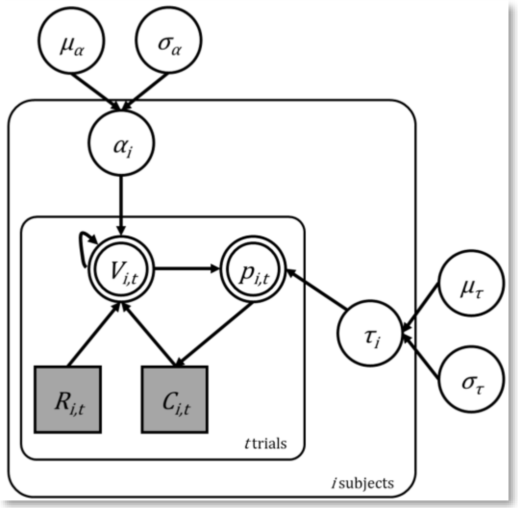
repo status Active | build passing | CRAN 1.0.2 – 2019-11-13 | downloads 33K

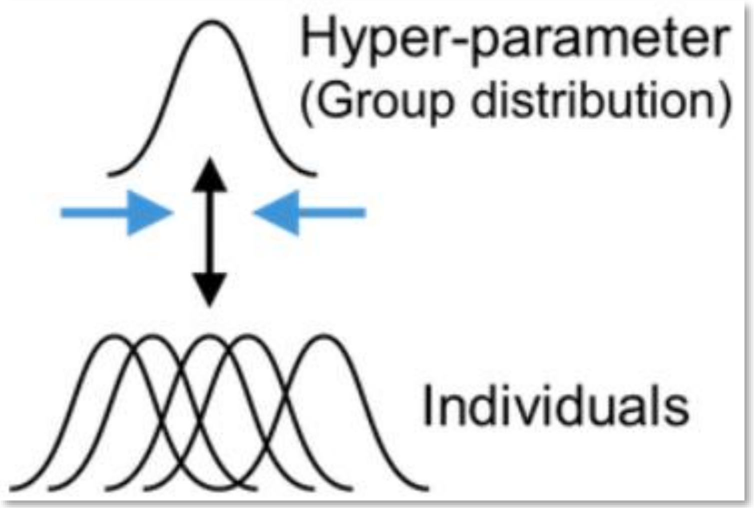
DOI 10.1162/CPSY_a_00002

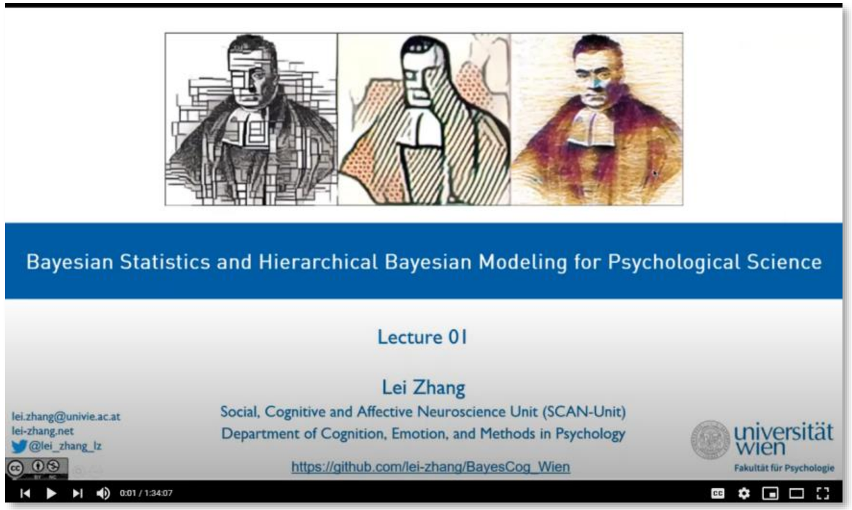
hBayesDM (hierarchical Bayesian modeling of Decision-Making tasks) is a user-friendly package that offers hierarchical Bayesian analysis of various computational models on an array of decision-making tasks. hBayesDM uses Stan for Bayesian inference.

hBayesDM







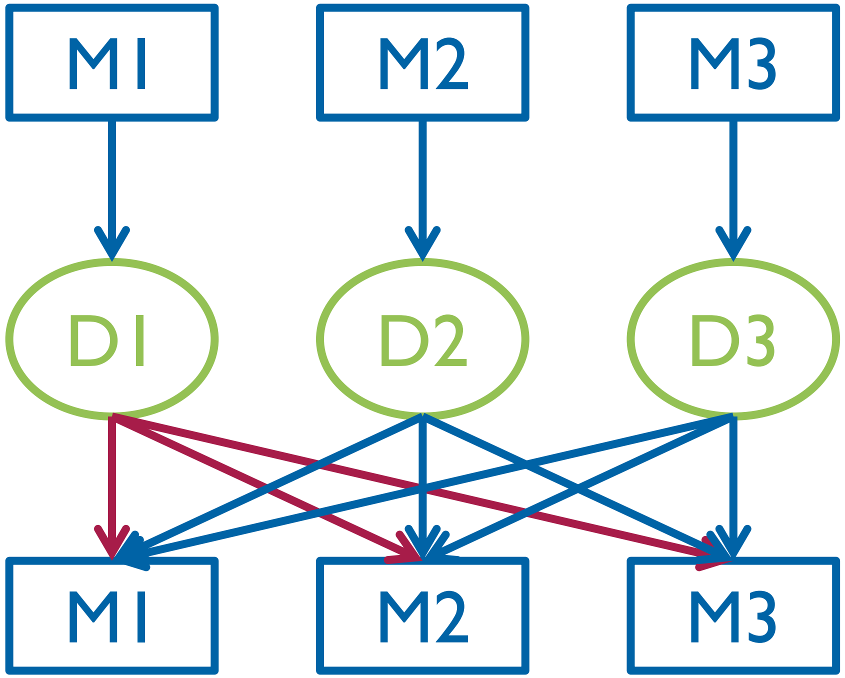


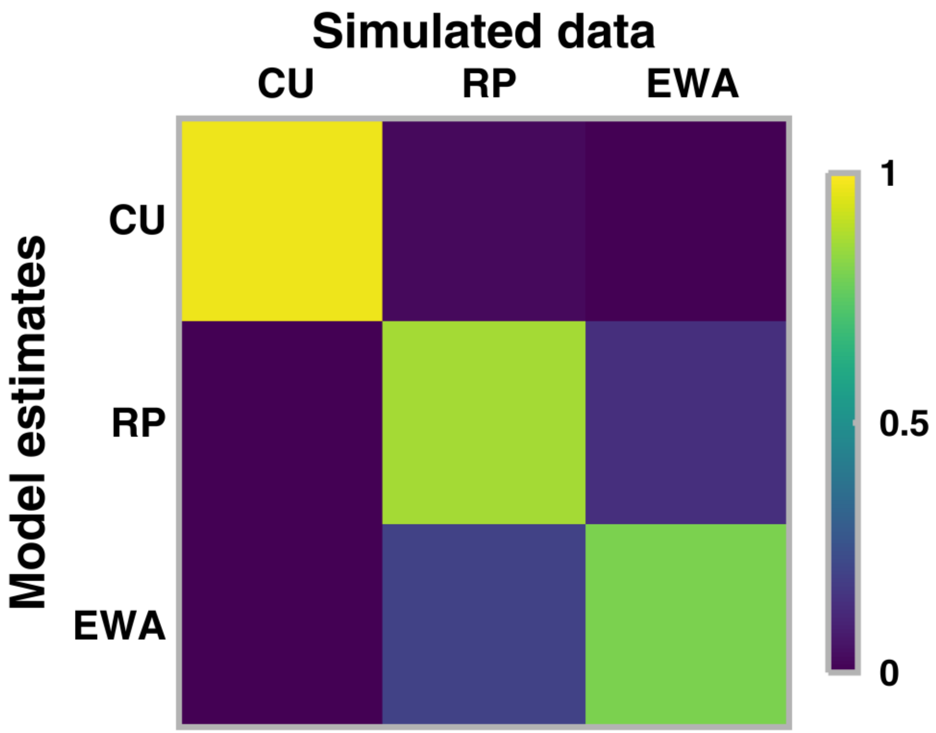
Ahn, Haines, Zhang, (2017)

Model recovery:
are models identifiable?

generative process

fitting process





Crawley* & Zhang* et al., (2020)

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

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

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** ▾

The Importance of Falsification in Computational Cognitive Modeling

Stefano Palminteri,[1,2,*,‡] Valentin Wyart,[1,2,*,‡] and Etienne Koechlin[1,2,*]

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

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, Monja I Froböse, Jennifer L Cook, Dirk EM Geurts, Michael J Frank, Roshan Cools, Hanneke EM den Ouden
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

COGNITIVE NEUROSCIENCE

A brain network supporting social influences in human decision-making

Lei Zhang[1,2]* and Jan Gläscher[1]*†

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

Social threat learning transfers to decision making in humans

Björn Lindström[a,b,c,1], Armita Golkar[c,d], Simon Jangard[c], Philippe N. Tobler[b], and Andreas Olsson[c]

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

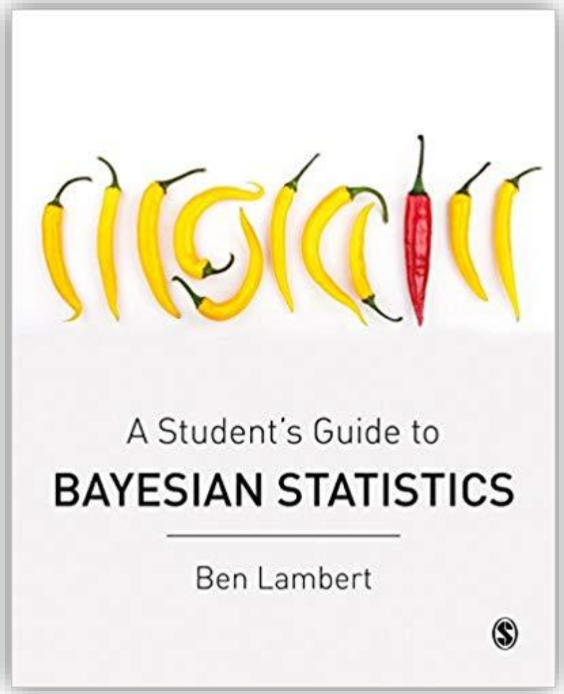
https://www.pnas.org/content/116/10/4732.abstract

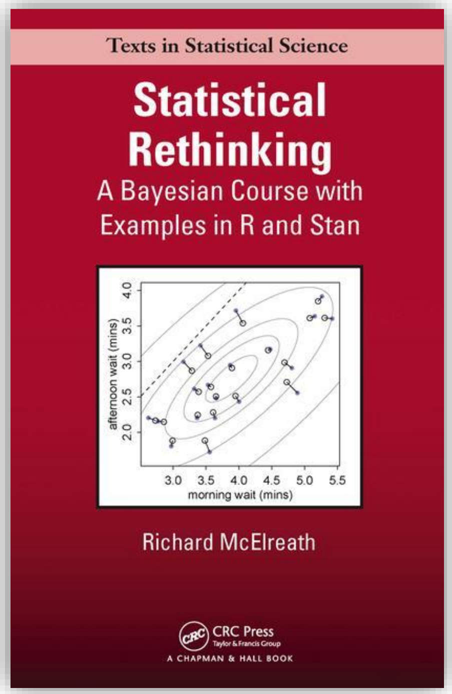
Modeling flexible behavior in childhood to adulthood shows age-dependent learning mechanisms and less optimal learning in autism in each age group

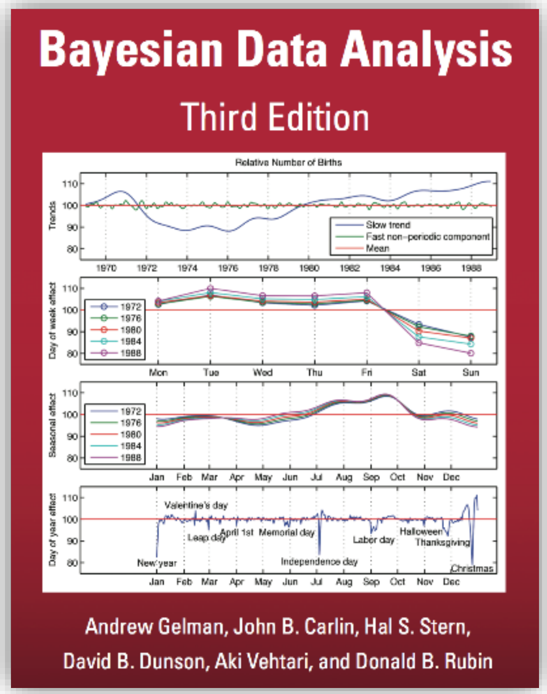
Daisy Crawley[1‡]*, Lei Zhang[2,3,4‡], Emily J. H. Jones[5], Jumana Ahmad[1,6], Bethany Oakley[1], Antonia San José Cáceres[1,7], Tony Charman[8,9], Jan K. Buitelaar[10,11,12], Declan G. M. Murphy[1,9,13], Christopher Chatham[4], Hanneke den Ouden[10‡], Eva Loth[1,13‡], the EU-AIMS LEAP group[¶]

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/stancon202… 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

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

Twitter

Richard McElreath
@rlmcelreath

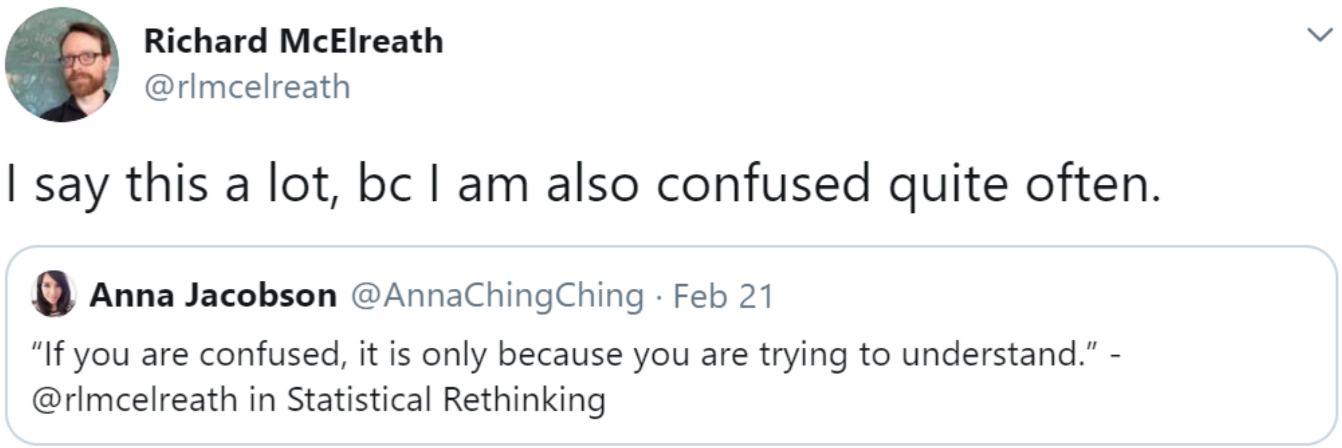
Quiche Lorraine,
@dan_p_simpson

\mathfrak{Michael
@betanalpha

EJ Wagenmakers
@EJWagenmakers

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 programing (e.g., R/Python/Julia/Matlab)
- Learn to seek external help (e.g., existing packages)
- Learn in pairs; practice makes perfect!



Acknowledgement

l.zhang.13@bham.ac.uk

https://lei-zhang.net/

@lei_zhang_lz

@LeiZhang

@lei-zhang

Thank you!


ANY
QUESTIONS
?

Happy Computing!