hBayesDM: Hands-On!

Prepared by Nate Haines

Presented by Woo-Young (Young) Ahn

Installation (actually the hardest part)

`0_install.R`

- For MAC users:
  1. Make sure you have Xcode installed: https://developer.apple.com/xcode/
  2. Run the following commands through the R console:

```
# install 'devtools' if required
if (!require(devtools)) install.packages("devtools")
devtools::install_github("CCS-Lab/hBayesDM")
```

- For Windows users:
  1. Download R tools: https://cran.r-project.org/bin/windows/Rtools/
     - For more details, see: https://github.com/stan-dev/rstan/wiki/Installing-RStan-on-Windows
  2. Run the following command through the R console:

```
install.packages("hBayesDM", dependencies = T)
```

1_test_stan.R

Testing Stan!

- We use Stan's "8 schools" example to make sure your computer can run Stan without errors. The model and its interpretation are unimportant.
  - If interested, see more on the model here:

    https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started#example-1-eight-schools

```
# Fit model using 1000 iterations on 2 different chains (not parallel)
# Note that the model compiles at the beginning, so it takes time to start
fit1 <- stan(model_code = m1, data = schools_dat,
             iter = 1000, chains = 2)

# Same fit as above, but test parallel computing
fit2 <- stan(model_code = m1, data = schools_dat,
             iter = 1000, chains = 2, cores = 2)
```

Distribute chains across 2 CPUs for faster fitting

2_help_files.R

Exploring hBayesDM

```
# Load hBayesDM library
library(hBayesDM)

# hBayesDM is documented extensively. To view the tasks/models included in
# the package, use the following command:
?hBayesDM
```

Click on model names for model-specific documentation

hBayesDM-package {hBayesDM}

Hierarchical Bayesian Modeling of Decision-Making Tasks

Description

Fit an array of decision-making tasks with computational models in a hierarchical Bayesian framework. Can perform hierarchical Bayesian analysis of various computational models with a single line of coding. Bolded tasks, followed by their respective models, are itemized below.

Bandit

2-Armed Bandit (Rescorla-Wagner (delta)) — bandit2arm_delta
4-Armed Bandit with fictive updating + reward/punishment sensitvity (Rescorla-Wagner (delta)) — bandit4arm_4par
4-Armed Bandit with fictive updating + reward/punishment sensitvity + lapse (Rescorla-Wagner (delta)) — bandit4arm_lapse

Delay Discounting

Help Files: Arguments

2_help_files.R

See help files for any model to see different arguments → `?dd_exp`

Arguments

`data`	A .txt file containing the data to be modeled. Data columns should be labelled as follows: "subjID", "delay_later", "amount_later", "delay_sooner", "amount_sooner", and "choice". See **Details** below for more information.
`niter`	Number of iterations, including warm-up.
`nwarmup`	Number of iterations used for warm-up only.
`nchain`	Number of chains to be run.
`ncore`	Integer value specifying how many CPUs to run the MCMC sampling on. Defaults to 1.
`nthin`	Every `i == nthin` sample will be used to generate the posterior distribution. Defaults to 1. A higher number can be used when auto-correlation within the MCMC sampling is high.
`inits`	Character value specifying how the initial values should be generated. Options are "fixed" or "random" or your own initial values.
`indPars`	Character value specifying how to summarize individual parameters. Current options are: "mean", "median", or "mode".
`saveDir`	Path to directory where .RData file of model output (`modelData`) can be saved. Leave blank if not interested.
`email`	Character value containing email address to send notification of completion. Leave blank if not interested.
`modelRegressor`	Exporting model-based regressors? TRUE or FALSE. Currently not available for this model.
`adapt_delta`	Floating point number representing the target acceptance probability of a new sample in the MCMC chain. Must be between 0 and 1. See **Details** below.
`stepsize`	Integer value specifying the size of each leapfrog step that the MCMC sampler can take on each new iteration. See **Details** below.
`max_treedepth`	Integer value specifying how many leapfrog steps that the MCMC sampler can take on each new iteration. See **Details** below.

Help Files: Data format

`2_help_files.R`

See help files for any model to see necessary data format → `?dd_exp`

Details

This section describes some of the function arguments in greater detail.

data should be assigned a character value specifying the full path and name of the file, including the file extension (e.g. ".txt"), that contains the behavioral data of all subjects of interest for the current analysis. The file should be a **tab-delimited** text (.txt) file whose rows represent trial-by-trial observations and columns represent variables. For the Delay Discounting Task, there should be six columns of data with the labels "subjID", "delay_later", "amount_later", "delay_sooner", "amount_sooner", and "choice". It is not necessary for the columns to be in this particular order, however it is necessary that they be labelled correctly and contain the information below:

`"subjID"`

A unique identifier for each subject within data-set to be analyzed.

`"delay_later"`

An integer representing the delayed days for the later option within the given trial. (e.g., 1 6 15 28 85 170).

`"amount_later"`

A floating number representing the amount for the later option within the given trial. (e.g., 10.5 38.3 13.4 31.4 30.9, etc.).

`"delay_sooner"`

An integer representing the delayed days for the sooner option (e.g., 0 0 0 0).

`"amount_sooner"`

A floating number representing the amount for the sooner option (e.g., 10 10 10 10).

`"choice"`

An integer value representing the chosen option within the given trial (e.g., 0=instant amount, 1=delayed amount)

Help Files: Data format

2_help_files.R

See help files for any model to see necessary data format → ?dd_exp

Details

This section describes some of the function arguments in greater detail.

data should be assigned a character value specifying the full path and name of the file, including the file extension (e.g. ".txt"), that contains the behavioral data of all subjects of interest for the current analysis. The file should be a **tab-delimited** text (.txt) file whose rows represent trial-by-trial observations and columns represent variables. For the Delay Discounting Task, there should be six columns of data with the labels "subjID", "delay_later", "amount_later", "delay_sooner", "amount_sooner", and "choice". It is not necessary for the columns to be in this particular order, however it is necessary that they be labelled correctly and contain the information below:

`"subjID"`

A unique identifier for each subject within data-set to be analyzed.

`"delay_later"`

An integer representing the delayed days for the later option within the given trial. (e.g., 1 6 15 28 85 170).

`"amount_later"`

A floating number representing the amount for the later option within the given trial. (e.g., 10.5 38.3 13.4 31.4 30.9, etc.).

`"delay_sooner"`

An integer representing the delayed days for the sooner option (e.g., 0 0 0 0).

`"amount_sooner"`

A floating number representing the amount for the sooner option (e.g., 10 10 10 10).

`"choice"`

An integer value representing the chosen option within the given trial (e.g., 0=instant amount, 1=delayed amount)

Required column headers in **tab-delimited** .txt file

Help Files: Returned Values

2_help_files.R

See help files for any model to see returned values → `?dd_exp`

Value

`modelData` A class `"hBayesDM"` object with the following components:

`model`

Character string with the name of the model (`"dd_exp"`).

`allIndPars`

`"data.frame"` containing the summarized parameter values (as specified by `"indPars"`) for each subject.

`parVals`

A `"list"` where each element contains posterior samples over different model parameters.

`fit`

A class `"stanfit"` object containing the fitted model.

`rawdata`

`"data.frame"` containing the raw data used to fit the model, as specified by the user.

Help Files: Returned Values

2_help_files.R

See help files for any model to see returned values → `?dd_exp`

Value

`modelData` A class `"hBayesDM"` object with the following components:

`model`

Character string with the name of the model (`"dd_exp"`).

`allIndPars`

`"data.frame"` containing the summarized parameter values (as specified by `"indPars"`) for each subject.

`parVals`

A `"list"` where each element contains posterior samples over different model parameters.

`fit`

A class `"stanfit"` object containing the fitted model.

`rawdata`

`"data.frame"` containing the raw data used to fit the model, as specified by the user.

Used for most post-hoc analyses

Workflow

See `dd_exp.stan`

```
fit <- dd_exp("~/my_data.txt")
```

`my_data.txt`

subjID	trial	delay_later	amount_later	delay_sooner	amount_sooner	choice
1	1	6	10.5	0	10	1
1	2	170	38.3	0	10	1
1	3	28	13.4	0	10	1
1	4	28	31.4	0	10	1
1	5	85	30.9	0	10	1
1	6	28	21.1	0	10	1
1	7	28	13	0	10	1
1	8	1	21.3	0	10	1
1	9	28	21.1	0	10	1
1	10	15	30.1	0	10	1
1	11	1	10.7	0	10	1
1	12	85	36.1	0	10	1
1	13	15	10.5	0	10	1

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 ...
}

`fit`

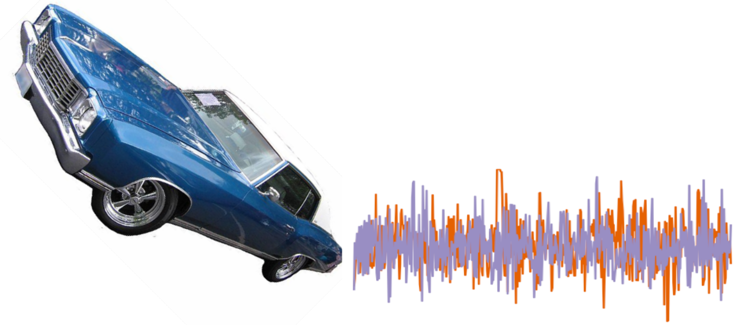
Diagnostics

Plotting

Inference

Model Comparison

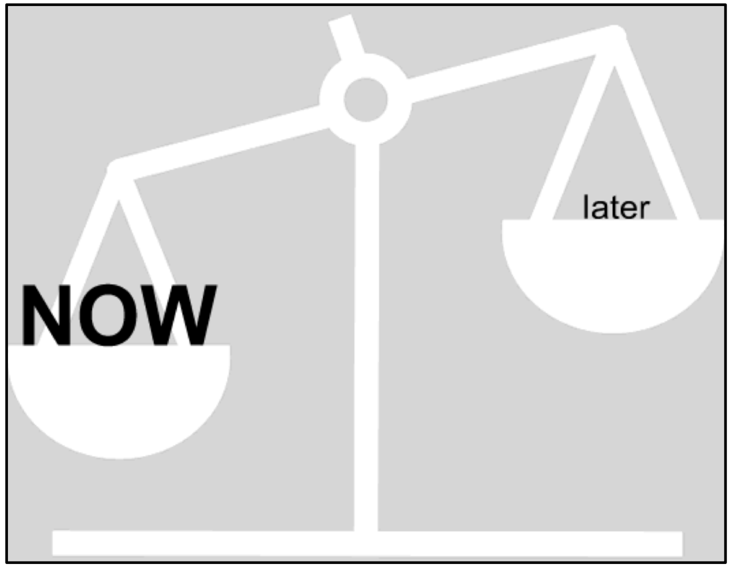
Let's get started!



Goals for today →

1. **Learn to fit models to three different tasks** (time allowing):
    1. Delay (i.e. temporal) Discounting
    2. Risk Aversion
    3. Go/ No-go
2. **Develop intuition for diagnosing poor convergence**
    1. Viewing MCMC traceplots and R-hat statistics
3. **Understand model comparison and interpretation**
    1. Information criteria to compare models assuming different cognitive processes
    2. Interpretation of parameter differences across different task conditions

Delay Discounting: Task



Mischel, Ebbesen, & Raskoff (1972, *J. of Personality and Social Psych.*)

Ability to delay gratification related to later:

1. Academic success,
2. Coping abilities, and
3. Other positive outcomes

see Mischel, Shoda, & Rodriguez (1989, *Science*)

Delay Discounting: Task

?

Now

In 2 weeks

Rachlin, Raineri, & Cross (1991, *J. Experimental Analysis of Behavior*)

Working Memory
(Hinson, Jameson, & Whitney, 2003)

Intelligence
(Shamosh et al., 2008)

Addictive behaviors
(MacKillop, 2013)

Schizophrenia
(Ahn et al., 2011; Heerey, Matveeva, & Gold, 2011; Heerey, et al., 2007)

Bipolar Disorder
(Ahn et al., 2011)

Delay Discounting: Models

3_dd_models.R

```
# Exponential model
?dd_exp
```

$$V_D = V_A \cdot e^{-r \cdot d}$$

Samuelson (1937, *Rev. Econ. Studies*)

```
# Hyperbolic model
?dd_hyperbolic
```

$$V_D = \frac{V_A}{1 + k \cdot d}$$

Mazur (1987, *Quant. Analysis of Beh.*)

Delay Discounting: Models

`3_dd_models.R`

```
# Exponential model
?dd_exp
```

$$V_D = V_A \cdot e^{-r \cdot d}$$

Samuelson (1937, *Rev. Econ. Studies*)

```
# Hyperbolic model
?dd_hyperbolic
```

$$V_D = \frac{V_A}{1 + k \cdot d}$$

Mazur (1987, *Quant. Analysis of Beh.*)

Discounting rates

- Between 0 and 1
- Closer to 0 = less impulsive/more patient
- Closer to 1 = more impulsive/ less patient

Delay Discounting: Models

`3_dd_models.R`

```
# Exponential model
?dd_exp
```

$$V_D = V_A \cdot e^{-r \cdot d}$$

Samuelson (1937, *Rev. Econ. Studies*)

Logistic link (also known as the "softmax" function)

$$Pr(Later) = \frac{1}{1 + e^{-((V_D - V_A) \cdot \beta)}}$$

```
# Hyperbolic model
?dd_hyperbolic
```

$$V_D = \frac{V_A}{1 + k \cdot d}$$

Mazur (1987, *Quant. Analysis of Beh.*)

Delay Discounting: <u>Models</u>

```
3_dd_models.R
```

```
# Exponential model
?dd_exp
```

$$V_D = V_A \cdot e^{-r \cdot d}$$

Samuelson (1937, *Rev. Econ. Studies*)

```
# Hyperbolic model
?dd_hyperbolic
```

$$V_D = \frac{V_A}{1 + k \cdot d}$$

Mazur (1987, *Quant. Analysis of Beh.*)

Logistic link (also known as the "softmax" function)

$$Pr(Later) = \frac{1}{1 + e^{-((V_D - V_A) \cdot \beta)}}$$

Inverse temperature

- Between 0 and 5
- Closer to 0 = more random choices with respect to ***V***
- Closer to 5 = more deterministic choice with respect to ***V***

Delay Discounting: Models

3_dd_models.R

```
# Exponential model
?dd_exp
```

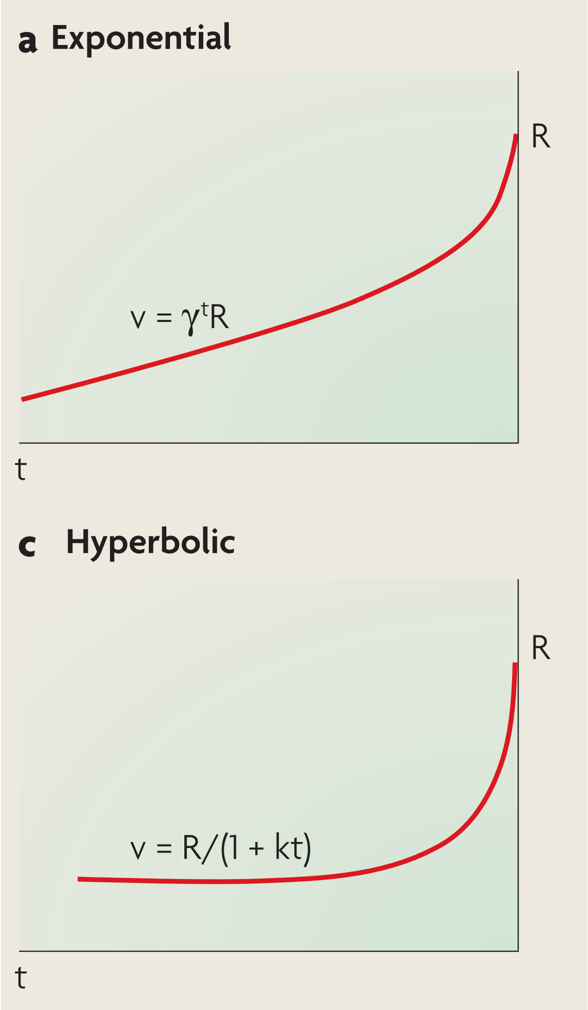
$$V_D = V_A \cdot e^{-r \cdot d}$$

Samuelson (1937, *Rev. Econ. Studies*)

```
# Hyperbolic model
?dd_hyperbolic
```

$$V_D = \frac{V_A}{1 + k \cdot d}$$

Mazur (1987, *Quant. Analysis of Beh.*)



Rangel, Camerer, & Montague (2008, *Nature Rev. Neuro.*)

Delay Discounting: Models

`3_dd_models.R`

```
# Exponential model
?dd_exp
```

$$V_D = V_A \cdot e^{-r \cdot d}$$

Samuelson (1937, *Rev. Econ. Studies*)

```
# Hyperbolic model
?dd_hyperbolic
```

$$V_D = \frac{V_A}{1 + k \cdot d}$$

Mazur (1987, *Quant. Analysis of Beh.*)

Which model better account for these data?

subjID	trial	delay_later	amount_later	delay_sooner	amount_sooner	choice
1	1	6	10.5	0	10	1
1	2	170	38.3	0	10	1
1	3	28	13.4	0	10	1
1	4	28	31.4	0	10	1
1	5	85	30.9	0	10	1
1	6	28	21.1	0	10	1
1	7	28	13	0	10	1
1	8	1	21.3	0	10	1
1	9	28	21.1	0	10	1
1	10	15	30.1	0	10	1
1	11	1	10.7	0	10	1
1	12	85	36.1	0	10	1
1	13	15	10.5	0	10	1

Delay Discounting: Fitting

3_dd_models.R

How to fit in hBayesDM? → Simple!

```r
# Fit the exponential model
fit_exp <- dd_exp("example", niter = 2000, nwarmup = 1000, nchain = 2, ncore = 2)
```

Delay Discounting: Fitting

3_dd_models.R

How to fit in hBayesDM? → Simple!

Number of iterations to use for tuning Hamiltonian Monte Carlo Sampler (not used in analyses)

Distribute chains over 2 CPUs

```r
# Fit the exponential model
fit_exp <- dd_exp("example", niter = 2000, nwarmup = 1000, nchain = 2, ncore = 2)
```

Use example data (included for each model)

Number of iterations to run for each chain

Run 2 chains in total

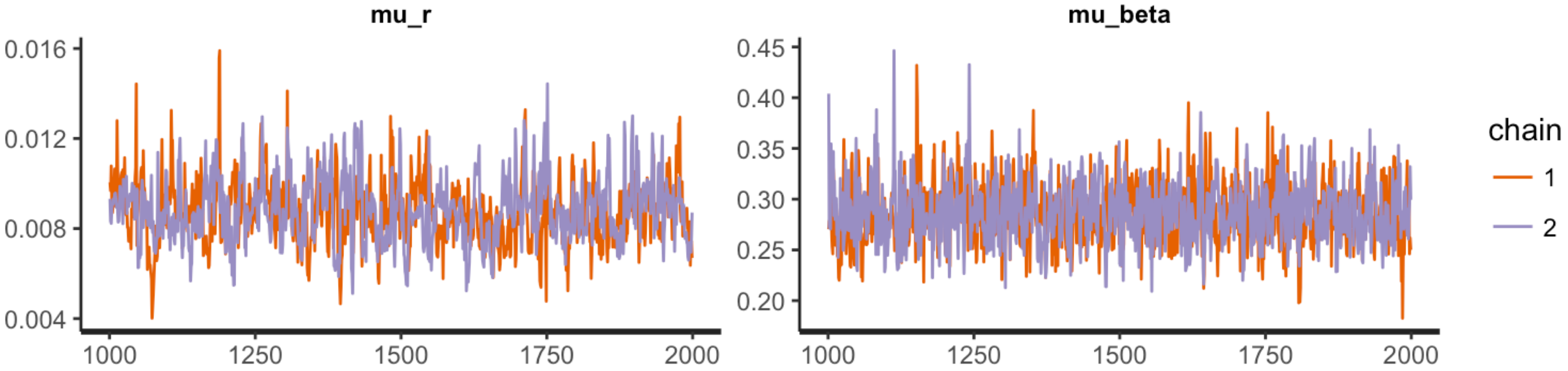
Note that we use 2 chains for toy examples. Run **at least 4** for real applications.

Delay Discounting: Diagnostics

3_dd_models.R

Visualize chains (i.e. "traceplots") →

```
# Make sure the chains are "mixing" well ("furry caterpillars")
plot(fit_exp, type = "trace")
```

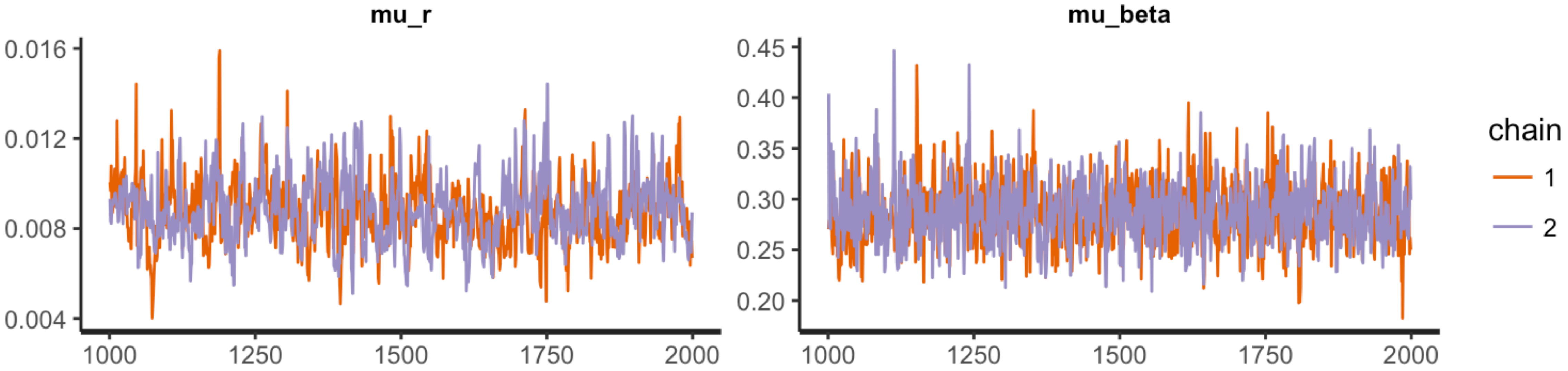


Delay Discounting: Diagnostics

3_dd_models.R

Visualize chains (i.e. "traceplots") →

```
# Make sure the chains are "mixing" well ("furry caterpillars")
plot(fit_exp, type = "trace")
```



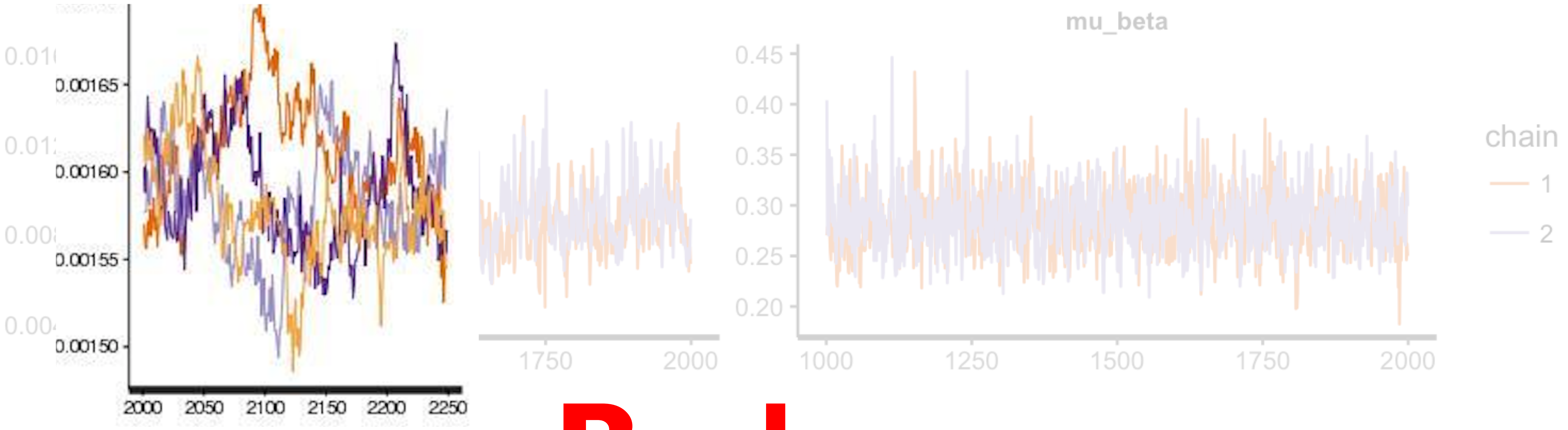
Good!

Delay Discounting: Diagnostics

3_dd_models.R

Visualize chains (i.e. "traceplots") →

```
# Make sure the chains are "mixing" well ("furry caterpillars")
plot(fit_exp, type = "trace")
```



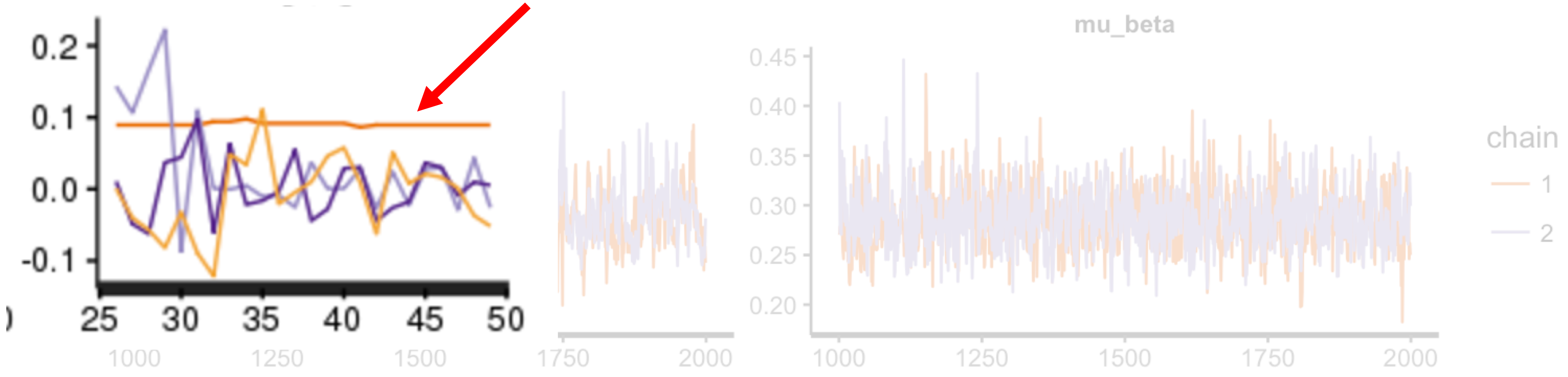
Bad...

Delay Discounting: Diagnostics

3_dd_models.R

Visualize chains (i.e. "traceplots") →

```
# Make sure the chains are "mixing" well ("furry caterpillars")
plot(fit_exp, type = "trace")
```



Really bad!

Delay Discounting: Diagnostics

3_dd_models.R

Check R-hat values →

```
# Return Rhat for each parameter
  rhat(fit_exp)
```

```
> rhat(fit_exp)
                Rhat
mu_r       1.0055032
mu_beta    1.0000060
sigma[1]   1.0015803
sigma[2]   1.0044683
r[1]       0.9995365
r[2]       0.9992113
r[3]       1.0004604
r[4]       0.9994235
     .
     .
     .
```

Or

```
# Simple check (i.e. True/False)
  rhat(fit_exp, less = 1.1)
```

```
> rhat(fit_exp, less = 1.1)
TRUE: All Rhat values are less than 1.1
[1] TRUE
```

Delay Discounting: Diagnostics

`3_dd_models.R`

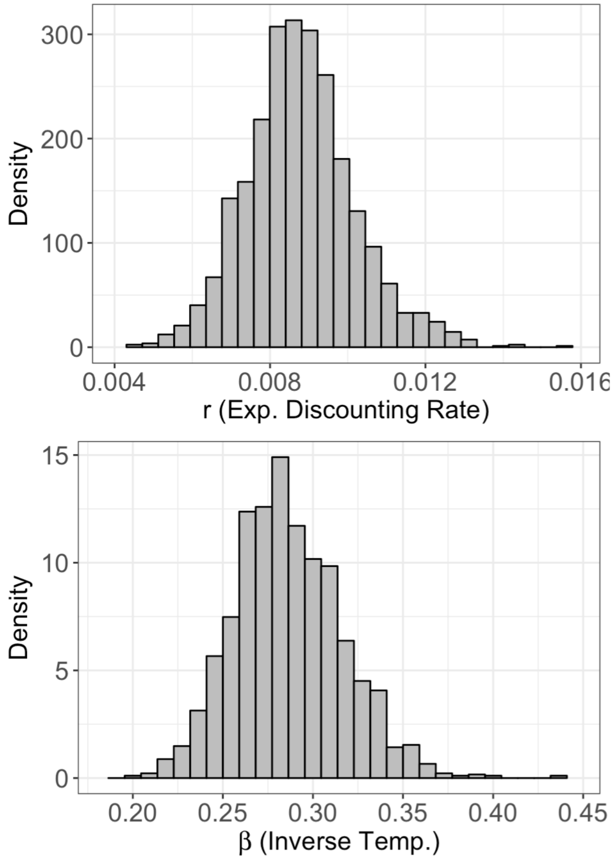
Very important to *always* check convergence

Delay Discounting: Visualizing

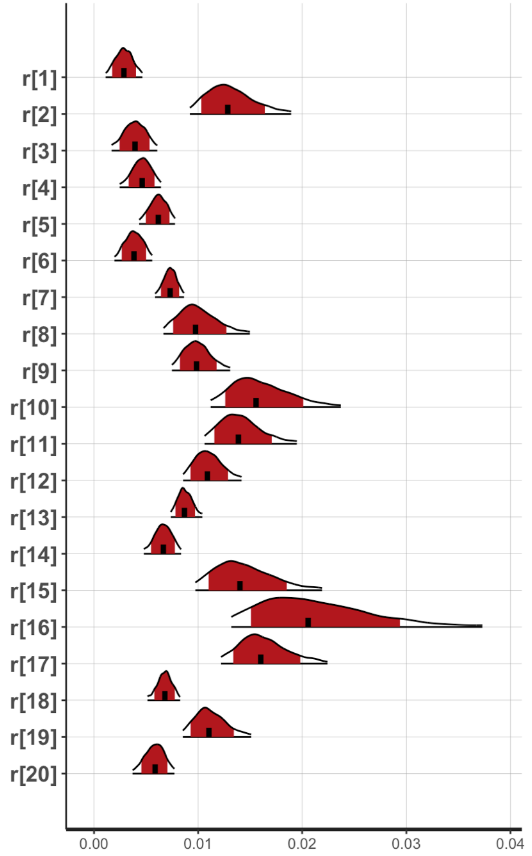
```
3_dd_models.R
```

hBayesDM offers **multiple methods** for visualizing parameters:

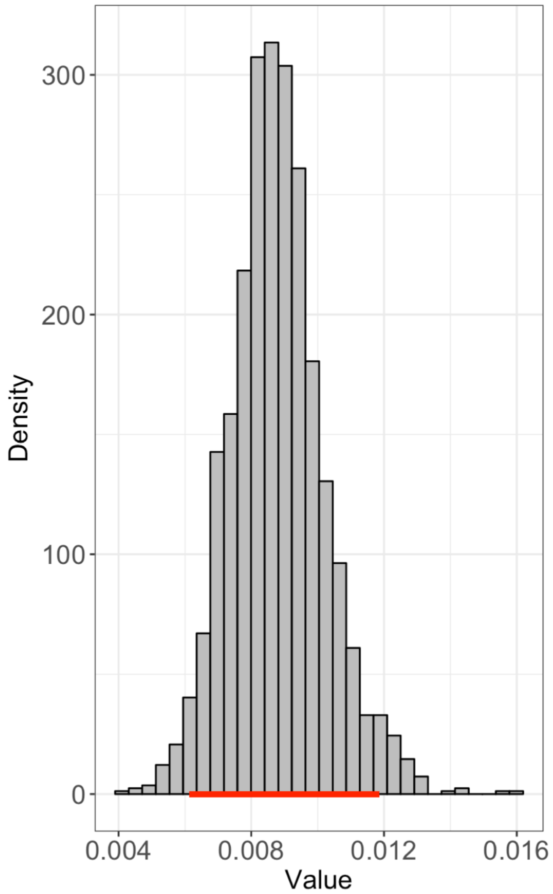
```
plot(fit_exp)
```




```
plotInd(fit_exp, pars = "r")
```




```
plotHDI(fit_exp$parVals$mu_r)
```



Delay Discountin Model Compariso

```
> printFit(fit_exp, fit_hyp)
          Model    LOOIC
1        dd_exp 1936.525
2 dd_hyperbolic 1887.354
```

3_dd_models.R

Use Leave-One-Out Information Criterion (LOOIC) to compare models:

```
printFit(fit_exp, fit_hyp)
```

```
> printFit(fit_exp, fit_hyp)
          Model    LOOIC LOOIC Weights
1        dd_exp 1941.713  5.802994e-13
2 dd_hyperbolic 1885.362  1.000000e+00
```

Model with lowest (i.e. closest to –∞) LOOIC has best fit

The hyperbolic model provides the best fit!

Table with rows for each model

Delay Discounting: Inference

`3_dd_models.R`

Access the model output to make inference:

```
fit_hyp$allIndPars

> fit_hyp$allIndPars
             k      beta subjID
1  0.003903576 0.5585866      1
2  0.027381461 0.3005429      2
3  0.007602515 0.2640412      3
4  0.007642756 0.3353525      4
5  0.011381992 0.4080799      5
6  0.005857174 0.4638293      6
7  0.013448884 0.5326219      7
8  0.022207471 0.2770387      8
9  0.019426198 0.3548839      9
10 0.030384833 0.3382569     10
```

Means of posterior distributions are shown for each subject/parameter

Discounting rate (k) can be used for further analyses:

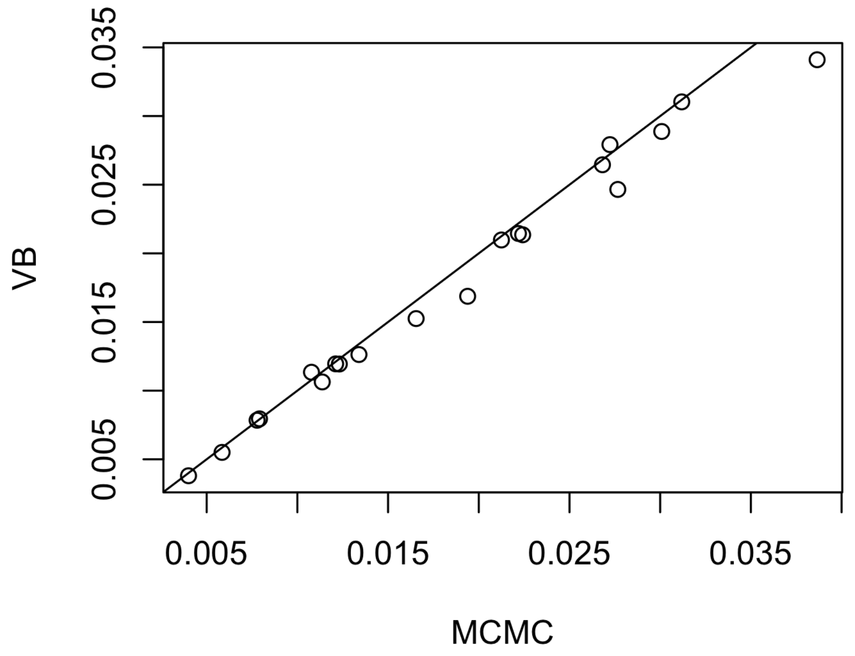
1. Correlation with impulsivity measures?
2. As an independent variable in a new model?
3. As the dependent variable in a new model?
4. Whatever your research question is!

3_dd_models.R

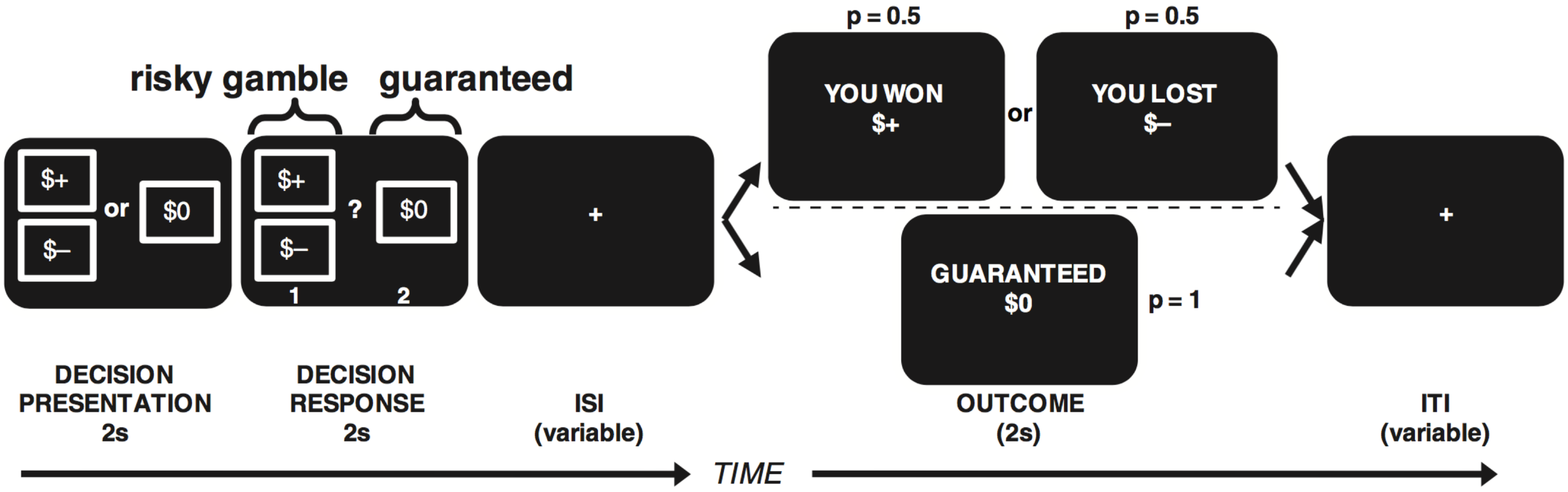
Delay Discounting: Compare MCMC and VB

```r
# Let's compare MCMC estimates and VB estimates
fit_hyp_vb <- dd_hyperbolic("example", niter = 2000, nwarmup = 1000,
                            nchain = 2, ncore = 2, vb=TRUE)
```

```r
# plot posterior means
plot(fit_hyp$allIndPars$k,
     fit_hyp_vb$allIndPars$k,
     xlab = "MCMC", ylab="VB")
# plot y=x line
abline(0,1)
```



Risk Aversion: Task



Sokol-Hessner et al. (2009, *PNAS; 2012, SCAN*)

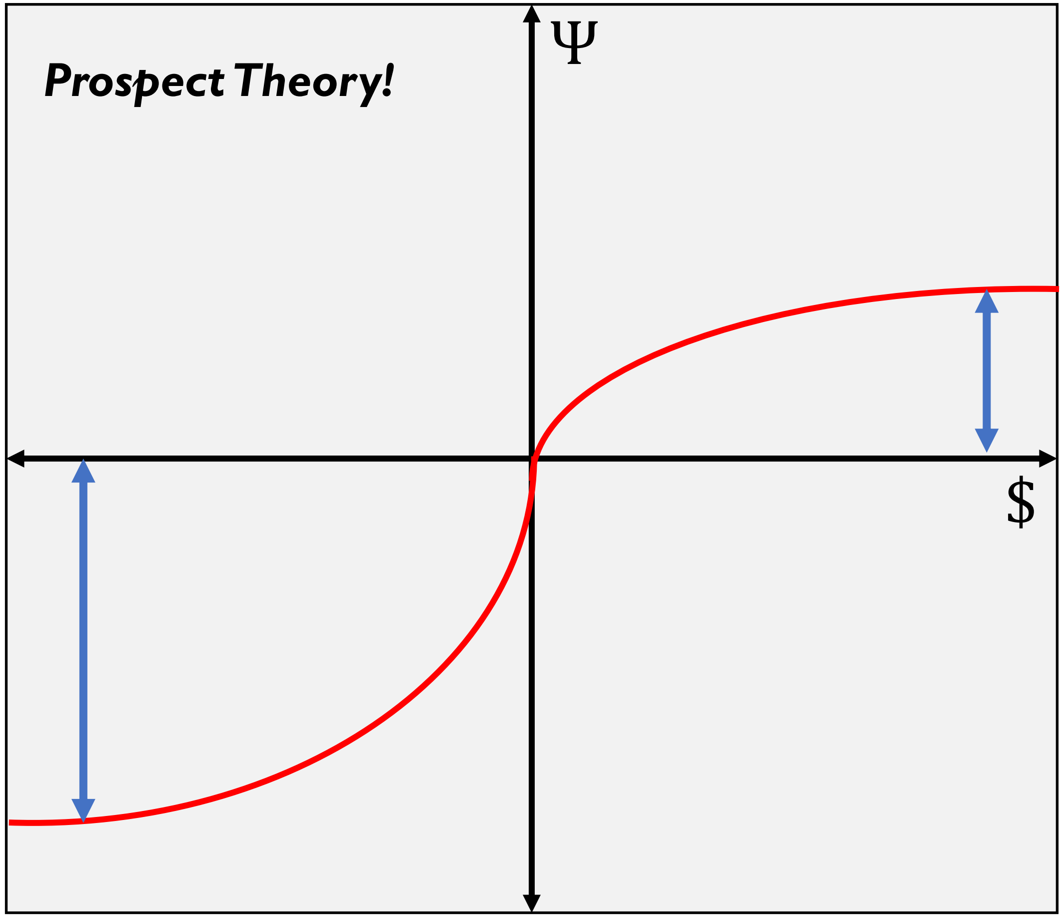
Risk Aversion: Models

If x is the possible outcome, the subjective utility of x is given by:

$$u(x) = \begin{cases} x^{\rho} & \text{if } x \geq 0 \\ -\lambda \cdot (-x)^{\rho} & x < 0 \end{cases}$$

Loss Aversion

- Between 0 and 5
- Closer to 0 = losses are lesser than equivalent gains
- Closer to 5 = losses are larger than equivalent gains



Sokol-Hessner et al. (2009, *PNAS; 2012, SCAN*)

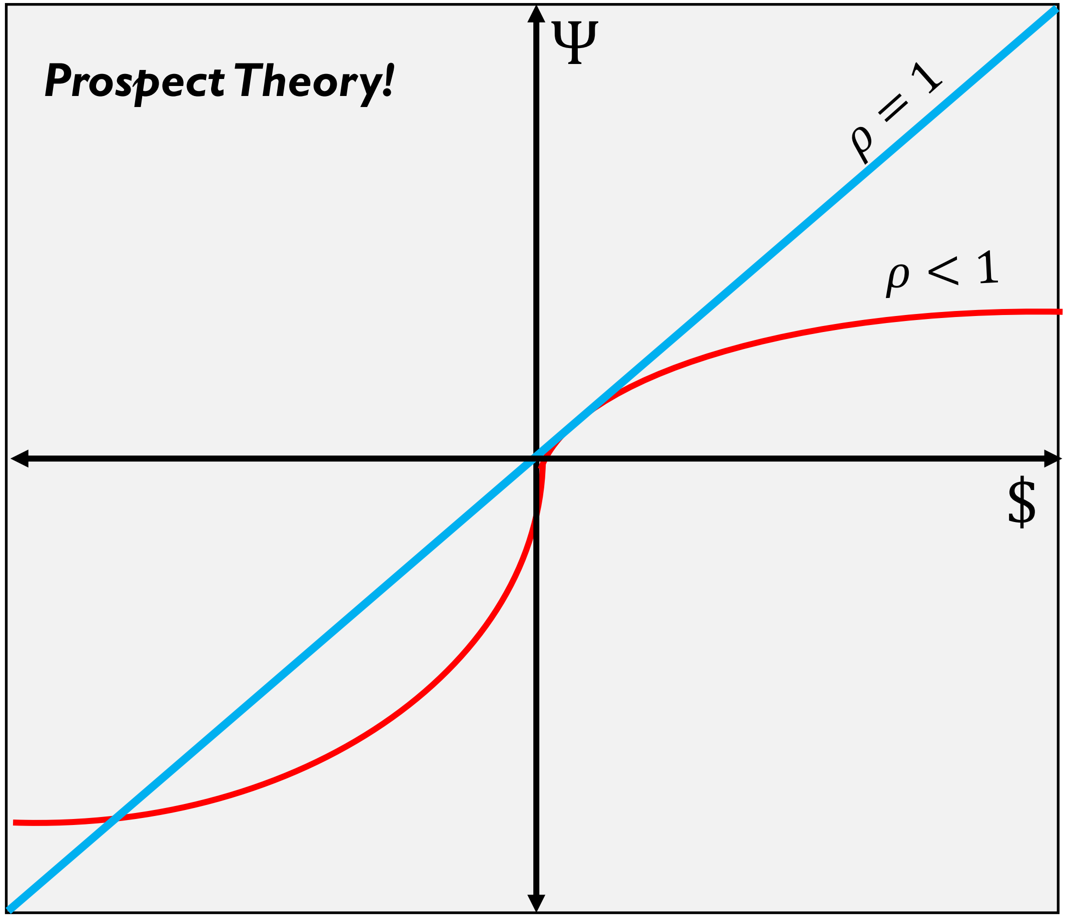
Risk Aversion: Models

If x is the possible outcome, the subjective utility of x is given by:

$$u(x) = \begin{cases} x^{\rho} & \text{if } x \geq 0 \\ -\lambda \cdot (-x)^{\rho} & x < 0 \end{cases}$$

Risk Aversion

- Between 0 and 2
- Controls utility shape



Sokol-Hessner et al. (2009, *PNAS; 2012, SCAN*)

Risk Aversion: Models

The value of the gamble is a weighted sum of the utility for winning versus losing:

$$V_{gamble} = 0.5 \cdot u(gain) + 0.5 \cdot u(loss)$$

And the value of the safe option is just its subjective utility:

$$V_{safe} = u(safe)$$

Sokol-Hessner et al. (2009, *PNAS; 2012, SCAN*)

Risk Aversion: Models

The value of the gamble is a weighted sum of the utility for winning versus losing:

$$V_{gamble} = 0.5 \cdot u(gain) + 0.5 \cdot u(loss)$$

And the value of the safe option is just its subjective utility:

$$V_{safe} = u(safe)$$

The probability of taking the gamble is the logistic link from before!

$$Pr(gamble) = \frac{1}{1 + e^{-((V_{gamble} - V_{safe}) \cdot \tau)}}$$

Inverse temperature

- Same as the delay discounting models

Sokol-Hessner et al. (2009, *PNAS; 2012, SCAN*)

Risk Aversion: Extract data

4_ra_models.R

hBayesDM contains data from Sokol-Hessner (2009)

- We will fit this data next
- To access the data, use the following commands:

```
# These are data collected when subjects were asked to attend to each trial
path_to_attend_data <- system.file("extdata/ra_data_attend.txt", package="hBayesDM")
```

```
# These are data collected when subjects were asked to view their choice as one
# within a large portfolio (i.e. think like a stock trader!)
path_to_regulate_data <- system.file("extdata/ra_data_reappraisal.txt", package="hBayesDM")
```

Risk Aversion: Fitting

`4_ra_models.R`

Fit the data using the same command as before!

- Just replace the task/model name and file path

```
# Fit the full risk aversion model to "attend" and "regulate" data (i.e. prospect theory)
fit_att_1 <- ra_prospect(path_to_attend_data, niter = 2000, nwarmup = 1000, nchain = 2, ncore = 2)
fit_reg_1 <- ra_prospect(path_to_regulate_data, niter = 2000, nwarmup = 1000, nchain = 2, ncore = 2)
```

Risk Aversion: Fitting

```
4_ra_models.R
```

Fit the data using the same command as before!

- Just replace the task/model name and file path

```r
# Fit the full risk aversion model to "attend" and "regulate" data (i.e. prospect theory)
fit_att_1 <- ra_prospect(path_to_attend_data, niter = 2000, nwarmup = 1000, nchain = 2, ncore = 2)
fit_reg_1 <- ra_prospect(path_to_regulate_data, niter = 2000, nwarmup = 1000, nchain = 2, ncore = 2)
```

Remember to check for convergence!

```r
# Check convergence for both models
plot(fit_att_1, "trace"); rhat(fit_att_1, 1.1)
plot(fit_reg_1, "trace"); rhat(fit_reg_1, 1.1)
```

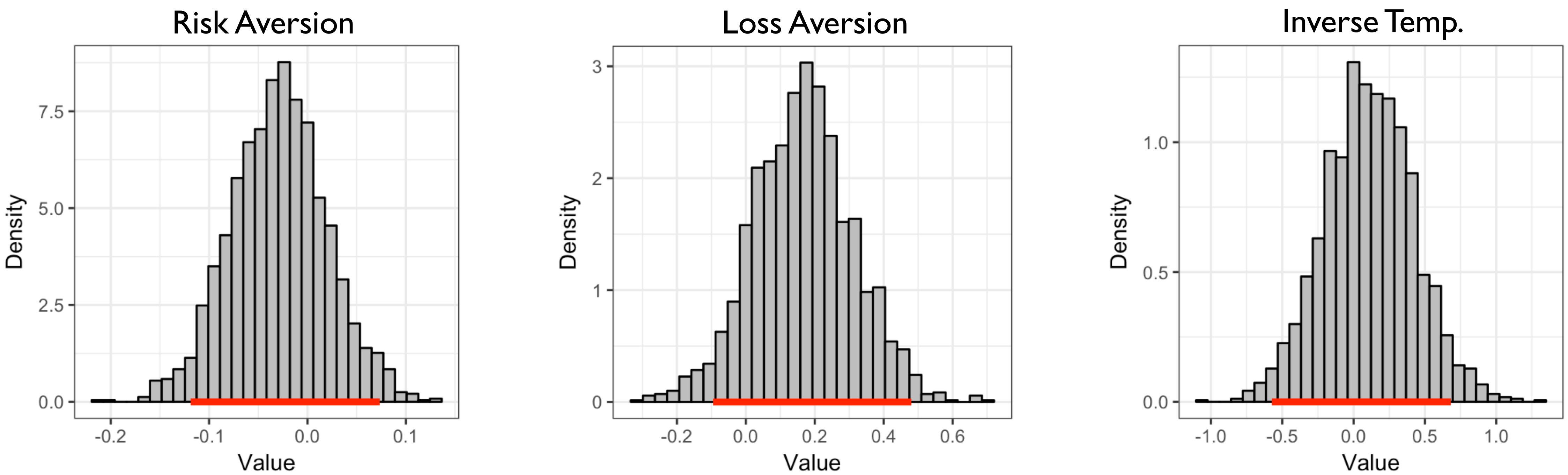
Risk Aversion: Inference

`4_ra_models.R`

Compare parameters across conditions

- Taking a difference in posterior distribution shows how the probability mass is different across conditions (particularly for Loss Aversion):

```
plotHDI(fit_att_1$parVals$mu_rho - fit_reg_1$parVals$mu_rho)
```



Examples → Kruschke (2014, *DBDA*) Ahn et al. (2014, *Frontiers in Psychology*) Beylergil et al. (2017, *NeuroImage: Clinical*)

Risk Aversion: Inference

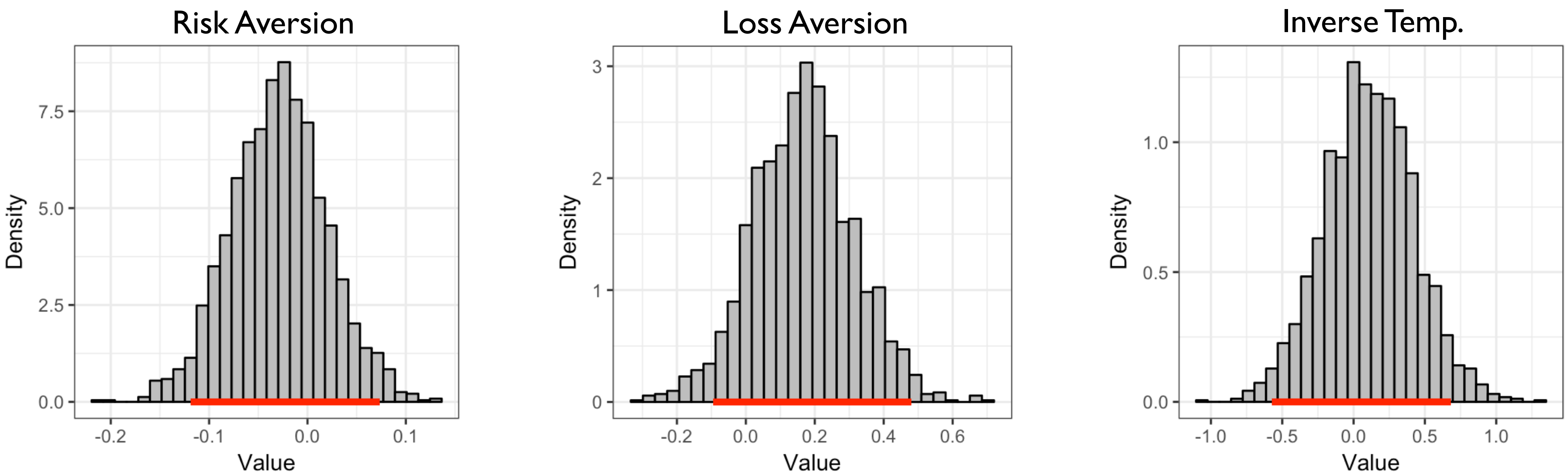
4_ra_models.R

Compare parameters across conditions

- Taking a difference in posterior distribution shows how the probability mass is different across conditions (particularly for Loss Aversion):

mu_lambda…mu_tau…

```
plotHDI(fit_att_1$parVals$mu_rho - fit_reg_1$parVals$mu_rho)
```



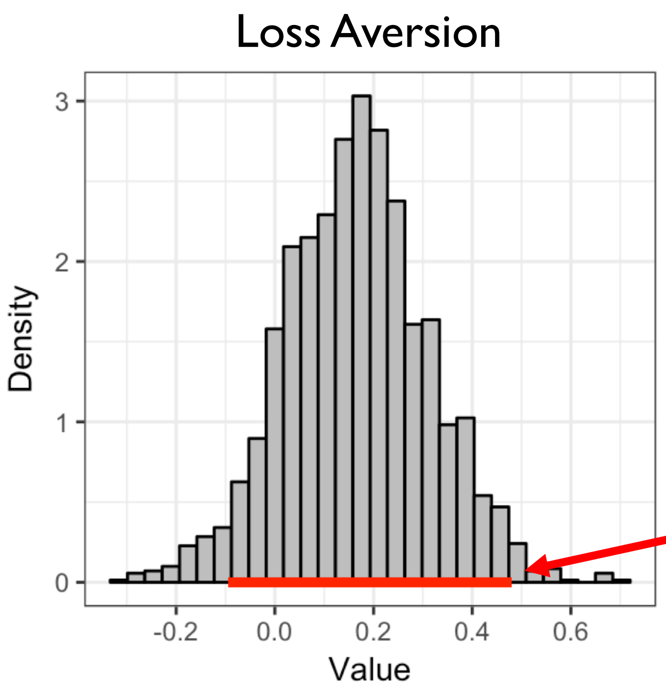
Examples → Kruschke (2014, *DBDA*) Ahn et al. (2014, *Frontiers in Psychology*) Beylergil et al. (2017, *NeuroImage: Clinical*)

Risk Aversion: Inference

```
4_ra_models.R
```

Compare parameters across conditions

- Taking a difference in posterior distribution shows how the probability mass is different across conditions:




```
> plotHDI(fit_att_1$parVals$mu_lambda - fit_reg_1$parVals$mu_lambda)
[1] "95% Highest Density Interval (HDI):"
[1] "Lower bound=-0.0948, Upper bound=0.4793"
```

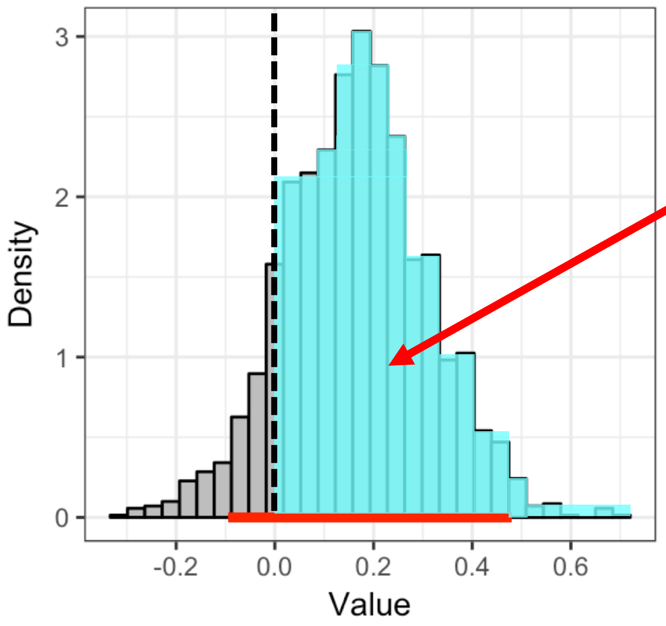
Risk Aversion: Inference

4_ra_models.R

Compare parameters across conditions

- Taking a difference in posterior distribution shows how the probability mass is different across conditions:

```
mean((fit_att_1$parVals$mu_lambda - fit_reg_1$parVals$mu_lambda)>0)
```




```
> mean((fit_att_1$parVals$mu_lambda - fit_reg_1$parVals$mu_lambda)>0)
[1] 0.88
```

Proportion of probability mass greater than 0 →

- Can be used in other models!
- ~0.5 would be expected if there was no change between conditions

Go/ No-go: Background

5_gng_models.R

Multiple, competing decision making systems

Box 1 | **Examples of behaviours driven by different valuation systems**

Valuation system	**Valence**	
	Appetitive (rewards)	**Avoidance (punishments)**
Pavlovian	Eat all food on plate	Cross street upon seeing dangerous person
	Reward obtained: food	Punishment avoided: potential harm
Habitual	Morning cup of coffee	Drive usual route to work
	Reward obtained: stimulant	Punishment avoided:traffic
Goal-directed	Movie selection	Go for a run
	Reward obtained: entertainment	Punishment avoided: obesity

Rangel, Camerer, & Montague (2008, *Nature Rev. Neuro.*)

Go/ No-go: Background

`5_gng_models.R`

Multiple, competing decision making systems

Box 1 | Examples of behaviours driven by different valuation systems

Valuation system	Valence	
	Appetitive (rewards)	Avoidance (punishments)
Pavlovian	Eat all food on plate	Cross street upon seeing dangerous person
	Reward obtained: food	Punishment avoided: potential harm
Habitual	Morning cup of coffee	Drive usual route to work
	Reward obtained: stimulant	Punishment avoided:traffic
Goal-directed	Movie selection	Go for a run
	Reward obtained: entertainment	Punishment avoided: obesity

Rangel, Camerer, & Montague (2008, *Nature Rev. Neuro.*)

We are biased to approach reward and avoid punishment

BUT…

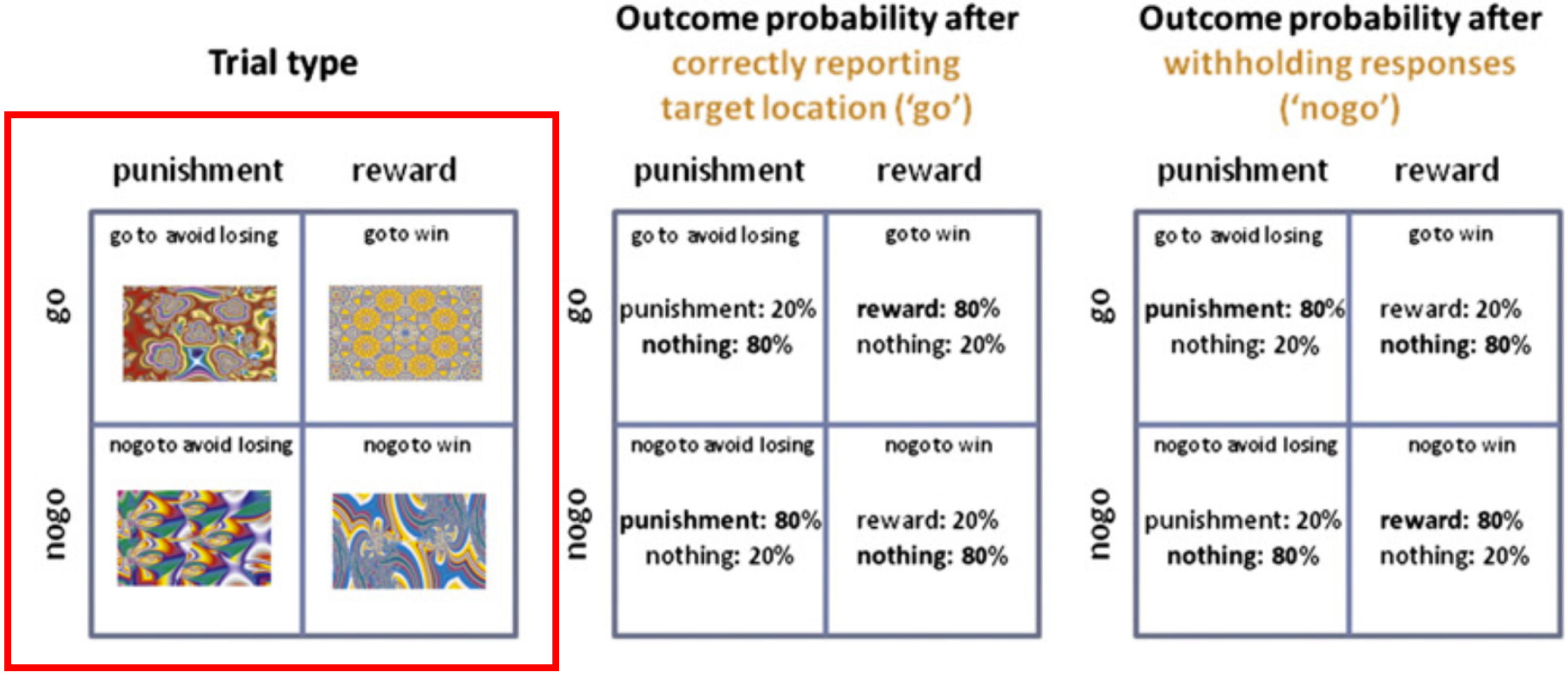
In most behavioral tasks, decision makers must approach (i.e. press button) a stimulus to be rewarded.

<u>This does not allow us to understand the role of ***Pavlovian Bias***.</u>

Go/ No-go: Task

5_gng_models.R

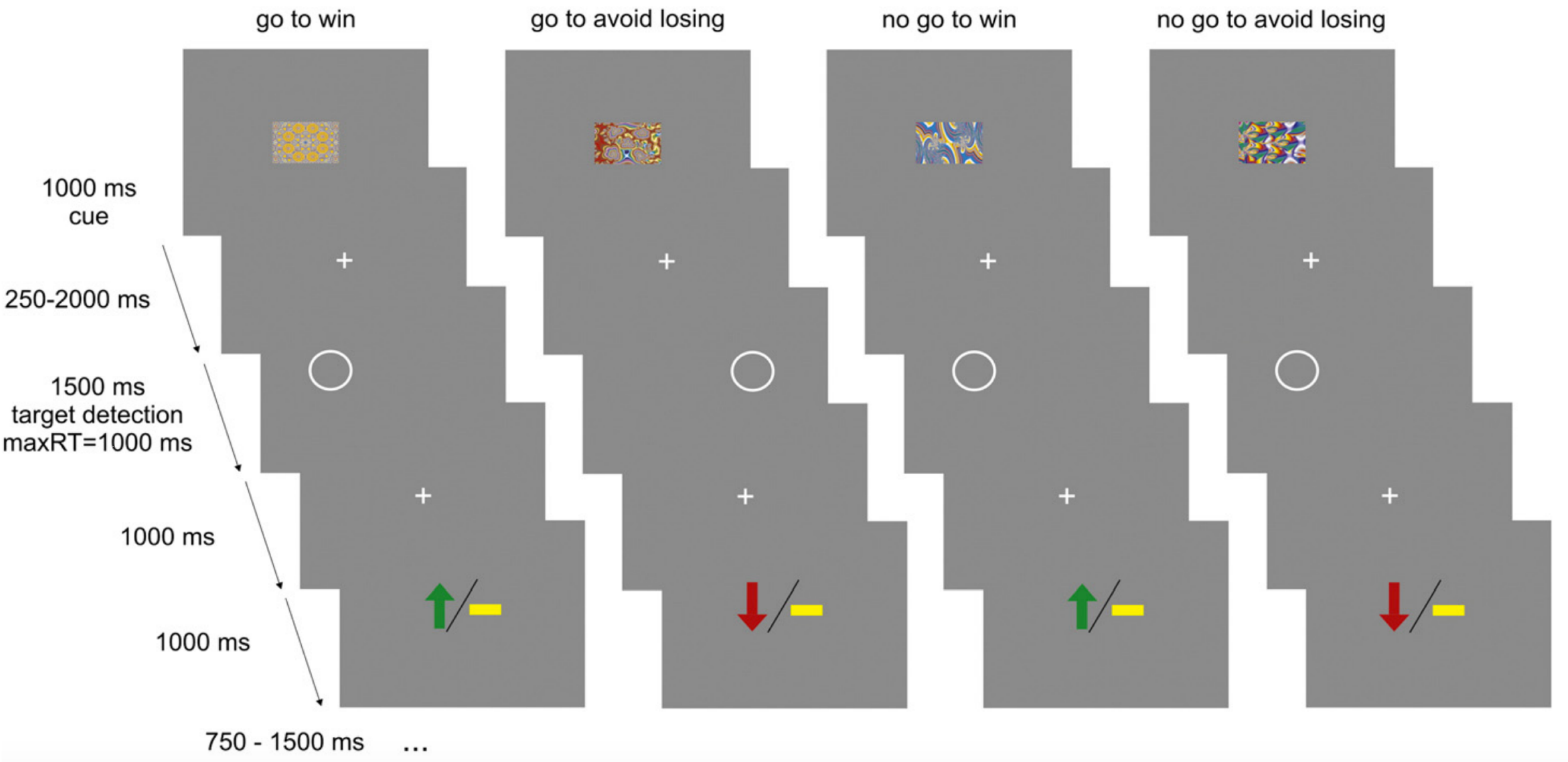
If we orthogonalize action and valence, we can study ***Pavlovian Bias:***



Guitart-Masip et al. (2012, *NeuroImage*)

Go/ No-go: Task

5_gng_models.R



Guitart-Masip et al. (2012, *NeuroImage*)

Go/ No-go: Models

`5_gng_models.R`

We will skip the equations for these. Conceptually, though →

hBayesDM contains **four models** for the orthogonalized Go/No-go:

1. gng_m1 → Assumes actions are learned without Go Bias or Pavlovian Bias
2. gng_m2 → Same as gng_m1, but adds Go Bias
3. gng_m3 → Same as gng_m2, but adds Pavlovian Bias
4. gng_m4 → Same as gng_m3, but assumes that rewards and punishments are evaluated differently

Guitart-Masip et al. (2012, *NeuroImage*) Cavanagh et al. (2013, *J. Neuro.*)

Go/ No-go: Fitting

5_gng_models.R

We simulated data from one of the four Go/No-go models, which model was it?

Start by locating data:

```
# Locate file path to go/ no-go data
gng_dat <- file.choose()
```

Go/ No-go: <u>Fitting</u>

`6_gng_models.R`

We simulated data from one of the four Go/No-go models, which model was it?

See **`6_gng_models.R`** for the answer!

To be posted on our GitHub…

Details of Stan modeling (e.g., how to set priors, likelihood functions, parameter bounds, etc.)

cpsy

an open access journal

RESEARCH

Revealing Neurocomputational Mechanisms of Reinforcement Learning and Decision-Making With the hBayesDM Package

Woo-Young Ahn[1], **Nathaniel Haines**[1], **and Lei Zhang**[2]

[1]Department of Psychology, The Ohio State University, Columbus, OH

[2]Institute for Systems Neuroscience, University Medical Center Hamburg-Eppendorf, Hamburg, Germany

Keywords: reinforcement learning, decision-making, hierarchical Bayesian modeling, model-based fMRI

For parameters that are bounded between 0 and 1 (e.g., learning rate), we use the inverse probit transformation (the cumulative distribution function of a unit normal distribution) to convert the unconstrained values into this range. In fact, given the mathematical relationship between the probability density function (pdf) and the cumulative density function (cdf) of the unit normal distribution, this transformation guarantees that the converted prior will be uniformly distributed between 0 and 1. Several studies have demonstrated the robustness and effectiveness of this transformation (e.g., Ahn et al., 2014; Wetzels et al., 2010). To effectively implement this, Stan provides a fast approximation of the inverse probit transformation (i.e., the `Phi_approx` function), which we adopted:

$$\mu_{\xi'} \sim \text{Normal}(0, 1)$$

$$\sigma_{\xi'} \sim \text{half-Cauchy}(0, 5)$$

$$\xi' \sim \text{Normal}(\mu_{\xi'}, \sigma_{\xi'})$$

$$\xi = \text{Probit}^{-1}(\xi')$$

https://en.wikipedia.org/wiki/Probit

Optimizing Approaches in Stan

Hierarchical models often suffer from highly correlated group-level parameters in their posterior distributions, creating challenges in terms of model convergence and estimation time (Gelman et al., 2013; Kruschke, 2014). To address these challenges, we practice reparameterization and vectorization in order to optimize the model specification in hBayesDM.

A Normal(μ, σ) distribution, like other distributions in the location–scale distribution family, can be reparameterized to be sampled from a unit normal distribution that is multiplied by the scale parameter σ and then shifted with the location parameter μ. Formally,

$$\xi \sim \text{Normal}(\mu_\xi, \sigma_\xi)$$

is mathematically equivalent to

$$\xi' \sim \text{Normal}(0, 1),$$

$$\xi = \mu_\xi + \xi' \cdot \sigma_\xi.$$

Such transformation is referred to as *noncentered parameterization* (a.k.a. the "Matt trick") by the Stan Development Team (2016), and it effectively reduces the dependence between μ_ξ, ξ, and σ_ξ and increases the effective sample size.

Where can I find actual Stan and R codes (in GitHub)?

- *Stan codes: https://github.com/CCS-Lab/hBayesDM/tree/master/exec*
- *R codes: https://github.com/CCS-Lab/hBayesDM/tree/master/R*
- *Sample data: https://github.com/CCS-Lab/hBayesDM/tree/master/inst/extdata*

References

Mischel, W., Ebbesen, E. B., & Raskoff Zeiss, A. (1972). Cognitive and attentional mechanisms in delay of gratification. *Journal of Personality and Social Psychology*, *21*(2), 204–218. http://doi.org/10.1037/h0032198

Mischel, W., Shoda, Y., & Rodriguez, M. L. (1989). Delay of gratification in children. *Science*.

Rachlin, H., Raineri, A., & Cross, D. (1991). SUBJECTIVE PROBABILITY AND DELAY. *Journal of the Experimental Analysis of Behavior*, *55*(2), 233–244. http://doi.org/10.1901/jeab.1991.55-233

Hinson, J. M., Jameson, T. L., & Whitney, P. (2003). Impulsive decision making and working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*(2), 298–306. http://doi.org/10.1037/0278-7393.29.2.298

Shamosh, N. A., DeYoung, C. G., Green, A. E., Reis, D. L., Johnson, M. R., Conway, A. R. A., et al. (2008). Individual Differences in Delay Discounting Relation to Intelligence, Working Memory, and Anterior Prefrontal Cortex. Psycholical Science, 19(9), 904–911. http://doi.org/10.1111/j.1467-9280.2008.02175.x

Ahn, W.-Y., & Vassileva, J. (2016). Machine-learning identifies substance-specific behavioral markers for opiate and stimulant dependence. *Drug and Alcohol Dependence*, *161*, 247–257. http://doi.org/10.1016/j.drugalcdep.2016.02.008

MacKillop, J. (2013). Integrating behavioral economics and behavioral genetics: delayed reward discounting as an endophenotype for addictive disorders. *Journal of the Experimental Analysis of Behavior*, *99*(1), 14–31. http://doi.org/10.1002/jeab.4

Ahn, W.-Y., Rass, O., Fridberg, D. J., Bishara, A. J., Forsyth, J. K., Breier, A., et al. (2011b). Temporal discounting of rewards in patients with bipolar disorder and schizophrenia. *Journal of Abnormal Psychology*, *120*(4), 911–921. http://doi.org/10.1037/a0023333

Heerey, E. A., Robinson, B. M., McMahon, R. P., & Gold, J. M. (2007). Delay discounting in schizophrenia. *Cognitive Neuropsychiatry*, *12*(3), 213–221. http://doi.org/10.1080/13546800601005900

Heerey, E. A., Matveeva, T. M., & Gold, J. M. (2011). Imagining the future: Degraded representations of future rewards and events in schizophrenia. *Journal of Abnormal Psychology*, *120*(2), 483–489. http://doi.org/10.1037/a0021810

Samuelson, P. A. (1937). A Note on Measurement of Utility. The Review of Economic Studies, 4(2), 155.

References continued

Mazur, J. E. (1987). An adjusting procedure for studying delayed reinforcement. In M. L. Commons, J. E. Mazur, J. A. Nevin, & H. Rachlin (Eds.), *Quantitative Analyses of Behavior* (pp. 55–73). Commons. http://doi.org/10.2307/1449146?ref=search- gateway:bae0459b9d53bb6ebb88e2d2afcdc47a

Ebert, J., & Prelec, D. (2007). The fragility of time: Time-insensitivity and valuation of the near and far future. *Management Science*, *53*(9), 1423–1438.

Rangel, A., Camerer, C. F., Camerer, C., & Montague, P. R. (2008). A framework for studying the neurobiology of value-based decision making. *Nature Reviews Neuroscience*, *9*(7), 545–556. http://doi.org/10.1038/nrn2357

Sokol-Hessner, P., Hsu, M., Curley, N. G., Delgado, M. R., Camerer, C. F., Phelps, E. A., & Smith, E. E. (2009). Thinking like a Trader Selectively Reduces Individuals' Loss Aversion. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(13), 5035–5040. http://doi.org/10.2307/40455144?ref=search-gateway:1f452c8925000031ef87ca756455c9e3

Sokol-Hessner, P., Camerer, C. F., & Phelps, E. A. (2012). Emotion regulation reduces loss aversion and decreases amygdala responses to losses. Social Cognitive and Affective Neuroscience, nss002. http://doi.org/10.1093/scan/nss002

Kruschke, J. (2014). Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan. Academic Press.

Ahn, W. Y., Vasilev, G., Lee, S. H., Busemeyer, J. R., Kruschke, J. K., Bechara, A., & Vassileva, J. (2014). Decision-making in stimulant and opiate addicts in protracted abstinence: evidence from computational modeling with pure users. Frontiers in psychology, 5, 849.

Beylergil, S. B., Beck, A., Deserno, L., Lorenz, R. C., Rapp, M. A., Schlagenhauf, F., ... & Obermayer, K. (2017). Dorsolateral prefrontal cortex contributes to the impaired behavioral adaptation in alcohol dependence. NeuroImage: Clinical, 15, 80-94.

Guitart-Masip, M., Huys, Q. J. M., Fuentemilla, L., Dayan, P., Duzel, E., & Dolan, R. J. (2012). Go and no-go learning in reward and punishment: Interactions between affect and effect. Neuroimage, 62(1), 154–166. http://doi.org/10.1016/j.neuroimage.2012.04.024

Cavanagh, J. F., Eisenberg, I., Guitart-Masip, M., Huys, Q., & Frank, M. J. (2013). Frontal Theta Overrides Pavlovian Learning Biases. Journal of Neuroscience, 33(19), 8541–8548. http://doi.org/10.1523/JNEUROSCI.5754-12.2013