Bayesian Statistics and Hierarchical Bayesian Modeling for Psychological Science

Lecture 05

Lei Zhang

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

Department of Basic Psychological Research and Research Methods

lei.zhang@univie.ac.at

lei-zhang.net

@lei_zhang_lz

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

universität wien

Fakultät für Psychologie

Review of a paper?

bioRxiv beta
THE PREPRINT SERVER FOR BIOLOGY

1

2

- decision-making
- cognitive modeling
- no fMRI
- similar length/difficulty

After L05	students 1:12	students 13:23
After L11	students 13:23	students 1:12

https://github.com/lei-zhang/BayesCog_Wien/tree/master/review_a_paper

Review of a paper?


bioRxiv preprint first posted online Apr. 24, 2018; doi: http://dx.doi.org/10.1101/303982. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC 4.0 International license.

CATECHOLAMINE CHALLENGE UNCOVERS DISTINCT MECHANISMS FOR DIRECT VERSUS INDIRECT, BUT NOT SOCIAL VERSUS NON-SOCIAL, LEARNING

Jennifer L. Cook[1*], Jennifer C. Swart[2], Monja I. Froböse[2], Andreea Diaconescu[3], Dirk E. M. Geurts[2,4], Hanneke E. M. den Ouden[2], & Roshan Cools[2,4].

1 School of Psychology, University of Birmingham, Edgbaston, Birmingham, United Kingdom

2 Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands

3 Translational Neuromodeling Unit, Institute for Biomedical Engineering, University of Zurich & ETH Zurich, Switzerland

4 Department of Psychiatry, Radboud University Medical Centre, Nijmegen, The Netherlands

* Correspondence: j.l.cook@bham.ac.uk

1

1 Baliko

2 Eder

3 Garber

4 Gianinazzi

5 Goltermann

6 Gyimesi

7 Hartmann

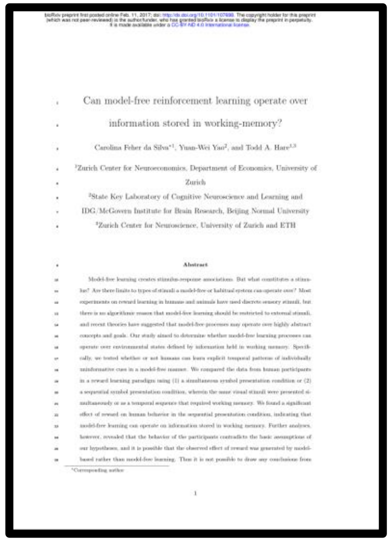
8 Kern

9 Kim

10 Kriegleder

11 Malik

12 Marschner


bioRxiv preprint first posted online Feb. 11, 2017; doi: http://dx.doi.org/10.1101/107698. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-ND 4.0 International license.

Can model-free reinforcement learning operate over information stored in working-memory?

Carolina Feher da Silva[*1], Yuan-Wei Yao[2], and Todd A. Hare[1,3]

[1]Zurich Center for Neuroeconomics, Department of Economics, University of Zurich

[2]State Key Laboratory of Cognitive Neuroscience and Learning and IDG/McGovern Institute for Brain Research, Beijing Normal University

[3]Zurich Center for Neuroscience, University of Zurich and ETH

Abstract

Model-free learning creates stimulus-response associations. But what constitutes a stimulus? Are there limits to types of stimuli a model-free or habitual system can operate over? Most experiments on reward learning in humans and animals have used discrete sensory stimuli, but there is no algorithmic reason that model-free learning should be restricted to external stimuli, and recent theories have suggested that model-free processes may operate over highly abstract concepts and goals. Our study aimed to determine whether model-free learning processes can operate over environmental states defined by information held in working memory. Specifically, we tested whether or not humans can learn explicit temporal patterns of individually uninformative cues in a model-free manner. We compared the data from human participants in a reward learning paradigm using (1) a simultaneous symbol presentation condition or (2) a sequential symbol presentation condition, wherein the same visual stimuli were presented simultaneously or as a temporal sequence that required working memory. We found a significant effect of reward on human behavior in the sequential presentation condition, indicating that model-free learning can operate on information stored in working memory. Further analyses, however, revealed that the behavior of the participants contradicts the basic assumptions of our hypotheses, and it is possible that the observed effect of reward was generated by model-based rather than model-free learning. Thus it is not possible to draw any conclusions from

[*]Corresponding author

1

13 Pfeiffer

14 Renz

15 Riedl

16 Rosenow

17 Rother

18 Schmeckenbecher

19 Schmid

20 Vilsmeier

21 Xu

22Song

23Muth

https://github.com/lei-zhang/BayesCog_Wien/tree/master/review_a_paper

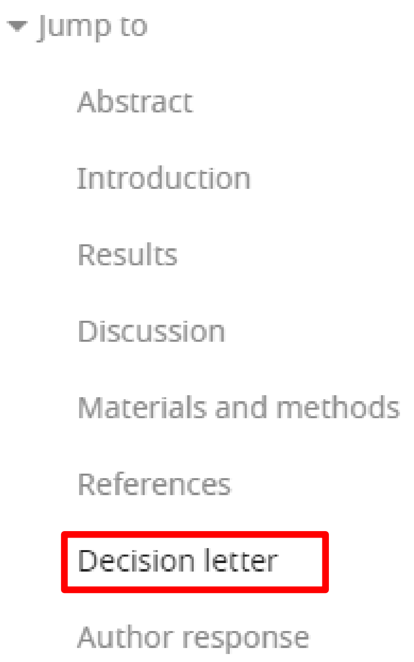
How to review a paper?

- Suppose you are invited by a journal editor to review a paper
- Of course, you have to read it☺, <u>carefully and critically</u>
- Then write a review report to the editor
  - (1) Make a summary. What is this paper about? What was done? What was the conclusion?
  - (2) List your concerns. Is the design appropriate? Are the analyses sound? Do their data support the conclusion? What can be done better?
- For this course:
  - be independent: okay to discuss HOW to review, but do NOT discuss WHAT to review

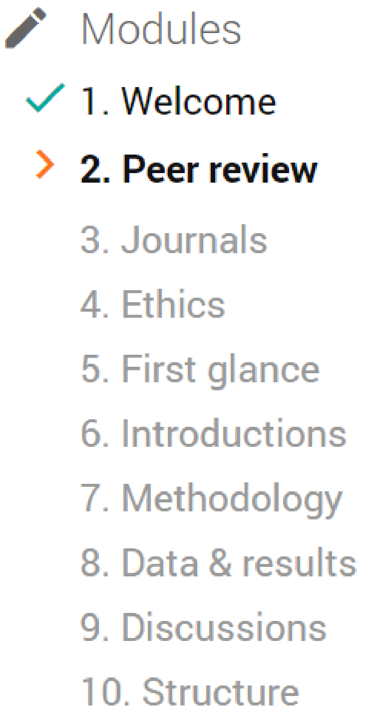
https://github.com/lei-zhang/BayesCog_Wien/tree/master/review_a_paper

Where to learn to review a paper?

- Publicly available review reports:
  - [Nature Communications]
  - [eLife]



- Structured online course
  - [Publons Academy]



https://github.com/lei-zhang/BayesCog_Wien/tree/master/review_a_paper

Review in action

- paper#_lastname_matriculatenumber.docx
- paper1_Cook_etal_2018.pdf
- paper2_daSilva_etal_2017.pdf
- paper_assign_list.txt

“Title of the paper”

Summary of the paper

In this paper xx et al., investigated xxx…

Strength of the paper

[theoretical contribution, experimental design, methodological endeavor, etc.]

Major concerns

[lacking literatures, inappropriate analyses, conclusion cannot be directly supported by the results etc.]

Minor concerns

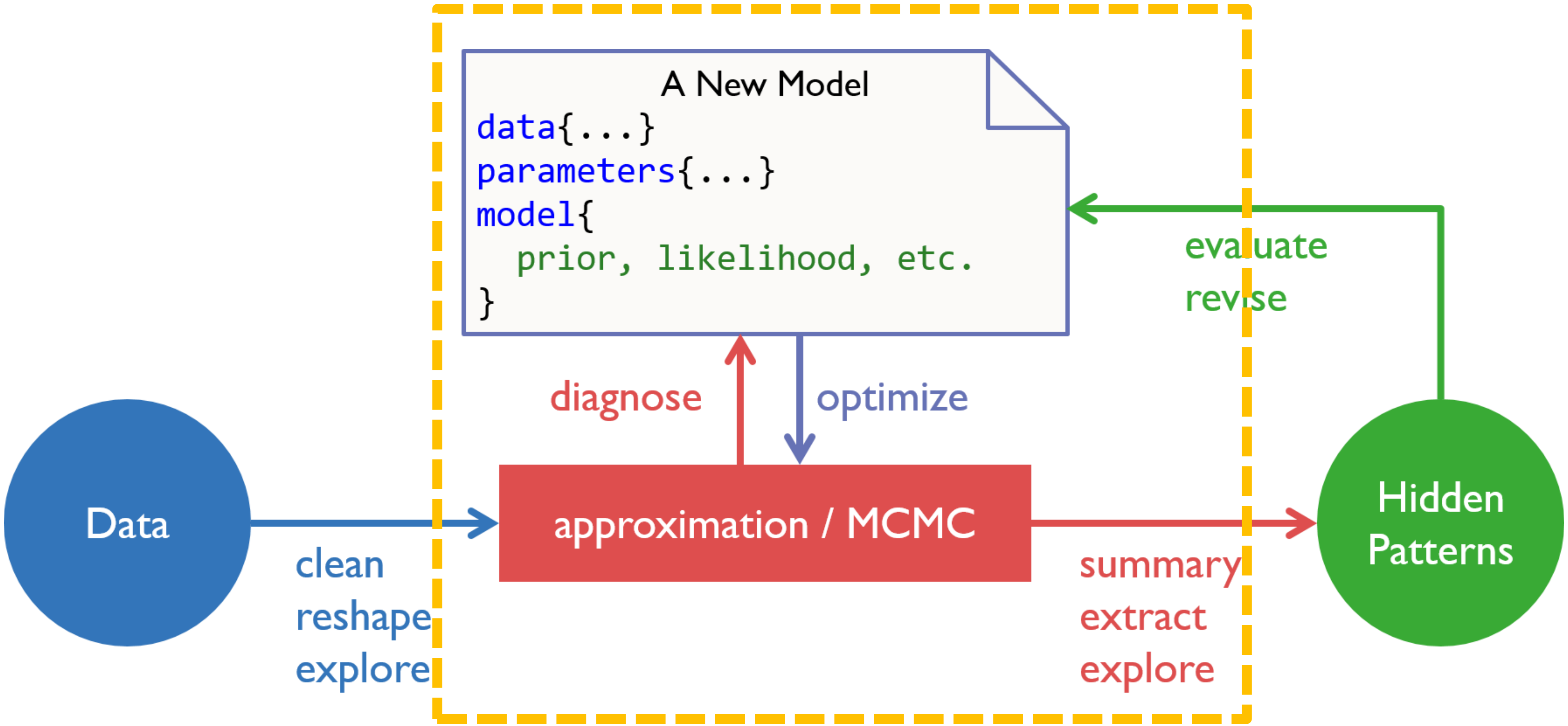
[typo, imprecise statistics (e.g., missing degrees of freedom), grammar mistakes, etc.]

- up to 3 pages (12pt, 1.5 space)
- send it via email to me
- New Due!: Sunday 05.05.2019

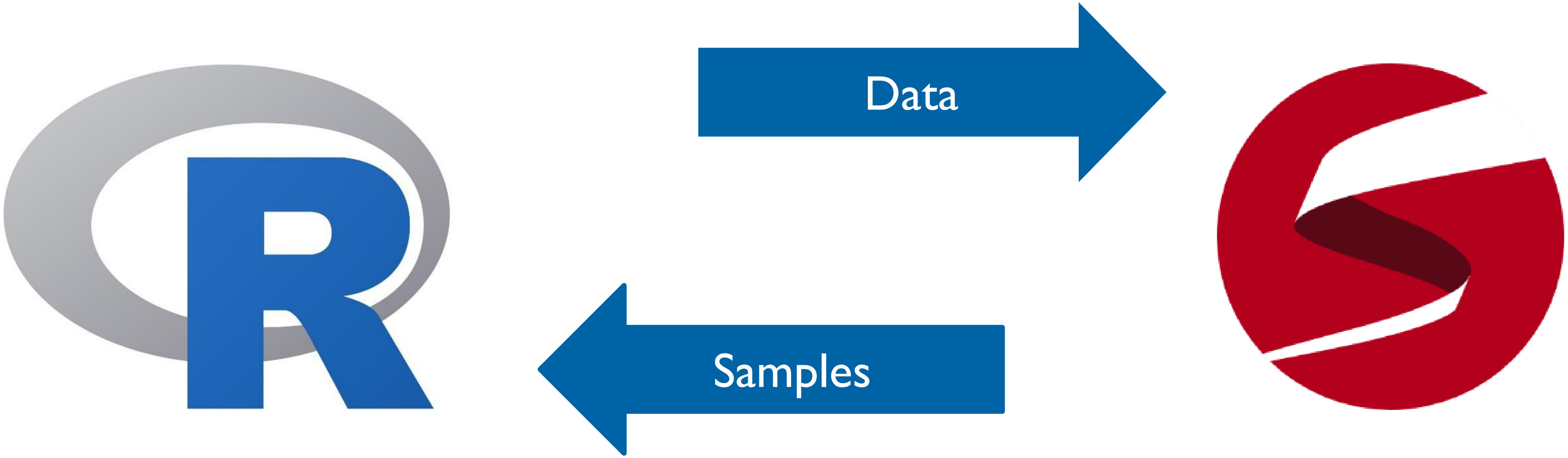
https://github.com/lei-zhang/BayesCog_Wien/tree/master/review_a_paper

STAN PROGRAMMING LANGUAGE I

cognitive model
statistics
computing



Stan and RStan



Steps of Bayesian Modeling, with Stan

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

McElreath (2016)

Steps of Using Stan

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

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

Stan Language

model blocks

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

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

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

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

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

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

REVISIT BINOMIAL MODEL

Binomial Model

W L W W W L W L W

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

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

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

McElreath (2016)

Graphical Model Notations

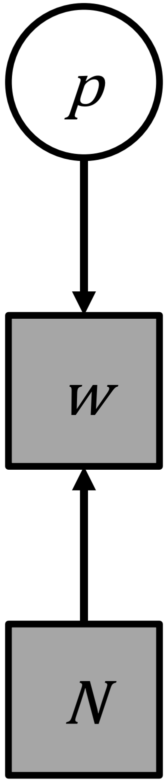
	continuous	discrete
unobserved	p	δ
observed	y	N

Lee & Wagenmakers (2013)

Binomial Model

W L W W W L W L W

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



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

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

	continuous	discrete
unobserved	p	δ
observed	y	N

McElreath (2016)

Binomial Model

W L W W W L W L W

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

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

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

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

McElreath (2016)

Running Binomial Model with Stan

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

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

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

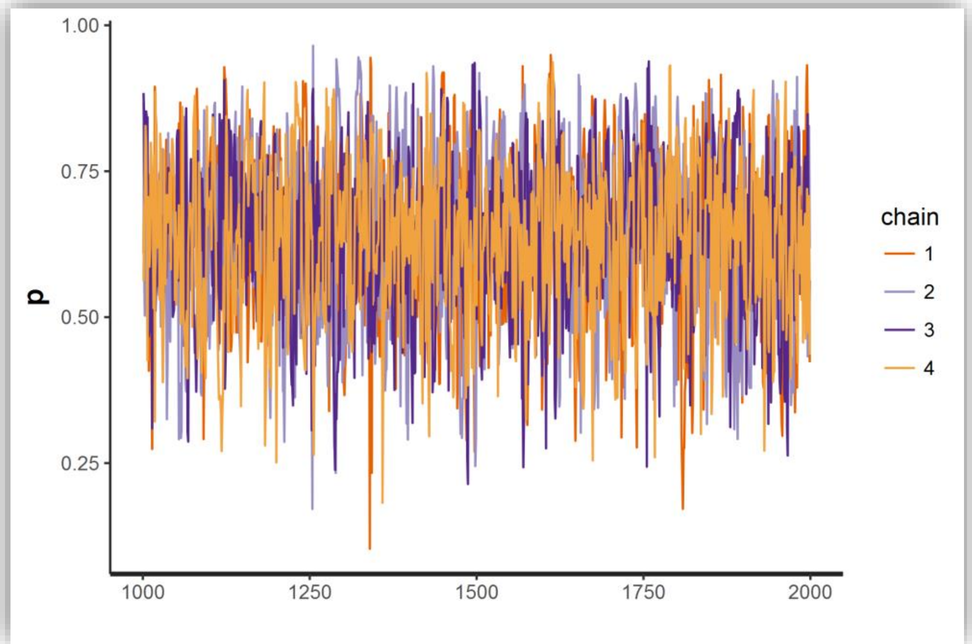
Model Summary

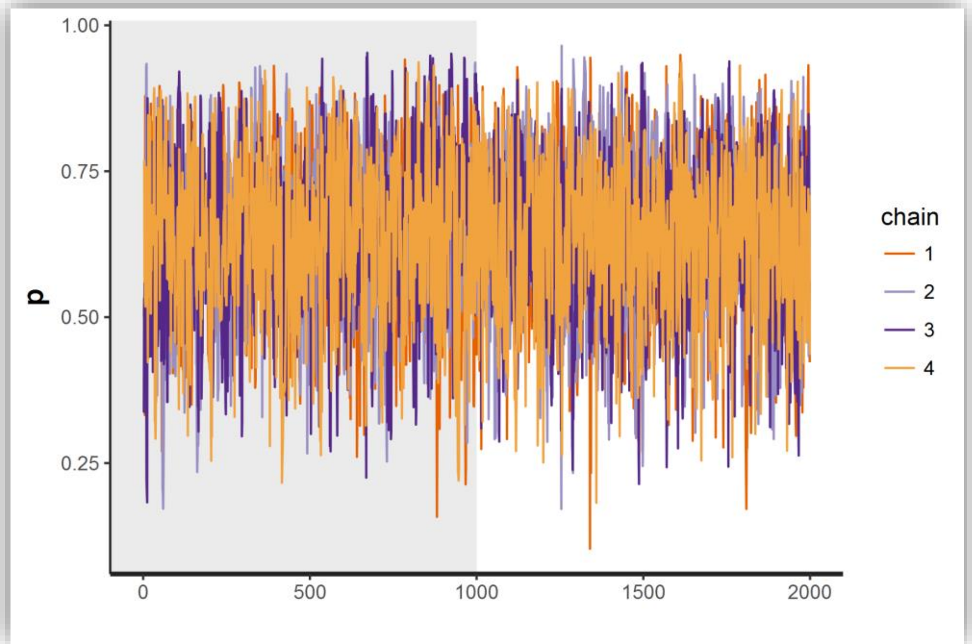
cognitive model
statistics
computing

```
> print(fit_globe)
Inference for Stan model: binomial_globe_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

      mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
p     0.64    0.00 0.14  0.35  0.54  0.65  0.74  0.87  1278    1
lp__ -7.72    0.02 0.69 -9.77 -7.89 -7.46 -7.27 -7.21  1824    1

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

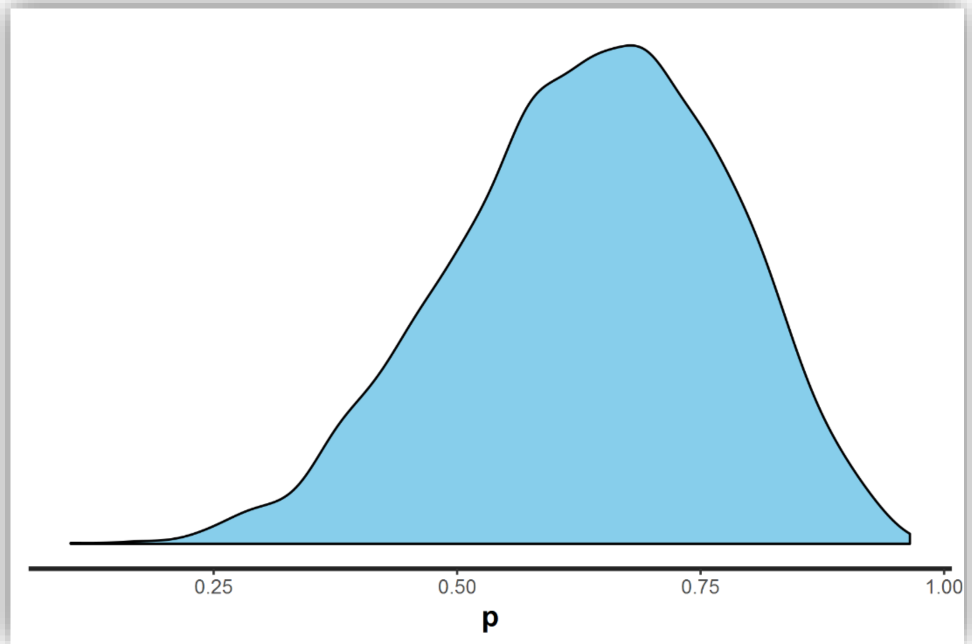
Gelman-Rubin convergence diagnostic (Gelman & Rubin, 1992)

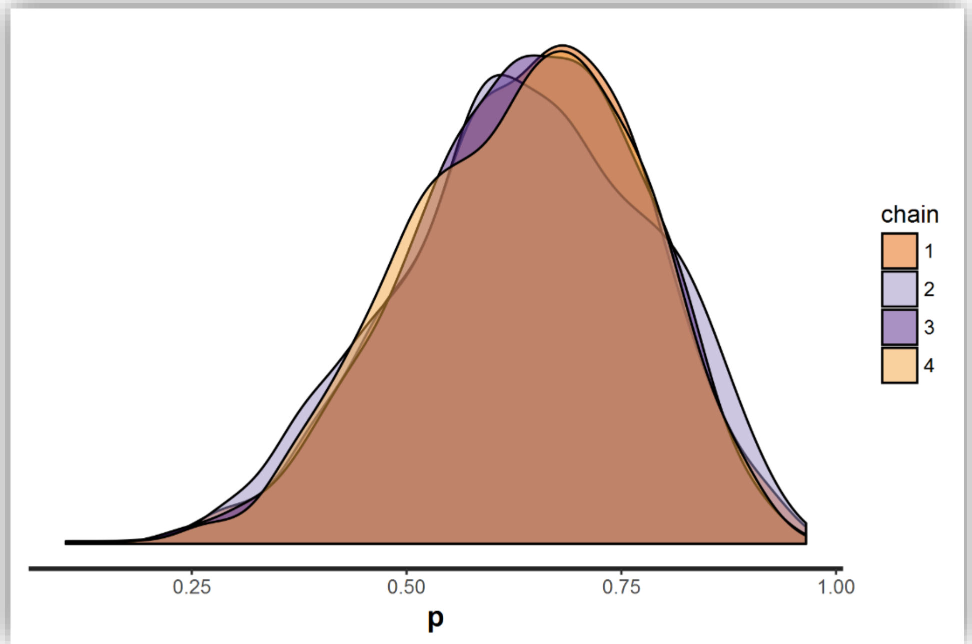
Diagnostics - traceplot





Diagnostics - density





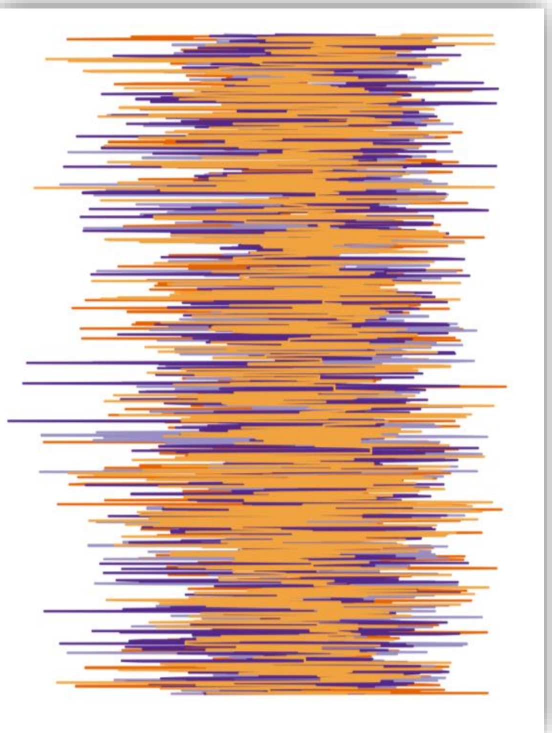
Diagnostics

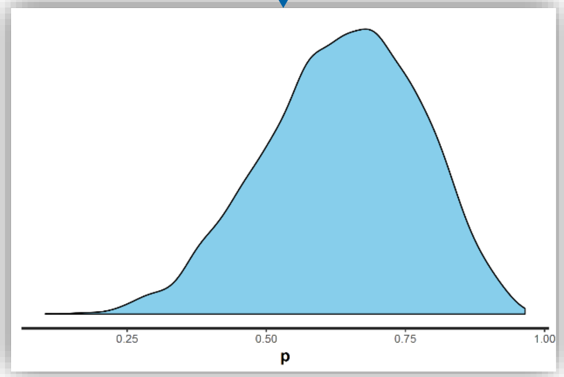
cognitive model

statistics

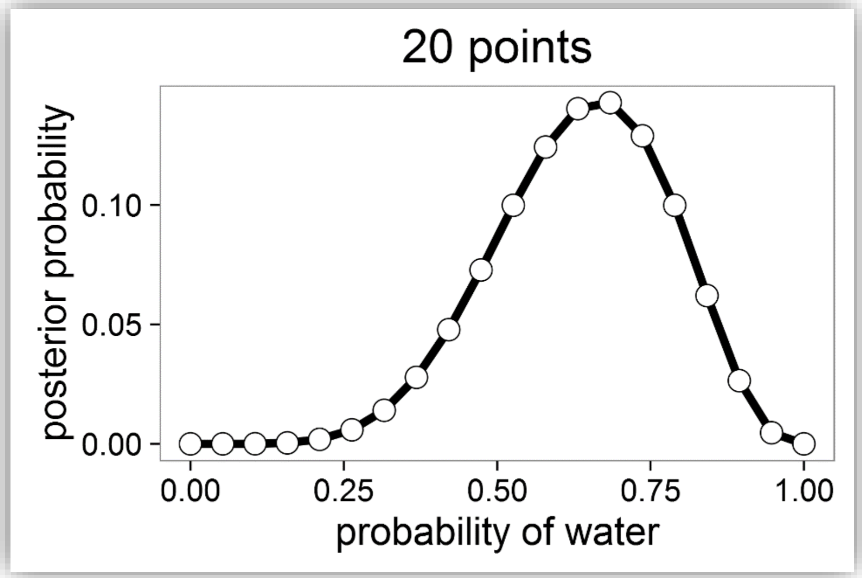
computing

MCMC



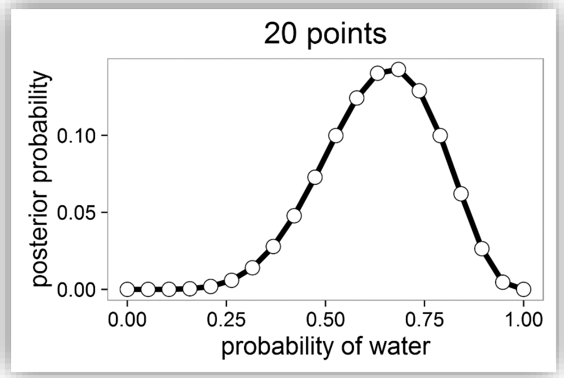


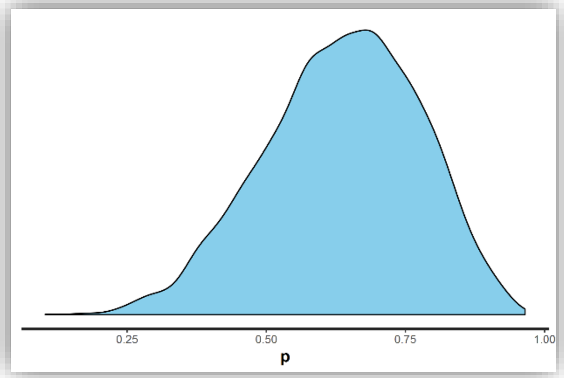
Grid Approximation



Draw a Conclusion?

- W = 6 out of N = 9
- uncertainty (relative plausibility) of all p values
- the relative plausibility of p = 0.63 is the highest, but it never rules out the possibility of p being other values, e.g., 0.5, 0.75
- → when p = 0.5, you may still observe 6W / 9 trials





Is Anything Missing? – NO

cognitive model
statistics
computing

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

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

model {
  p ~ uniform(0,1);
  w ~ binomial(N, p);
}
```

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

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

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

STAN PROGRAMMING LANGUAGE II

Why Use Stan?

vs. BUGS and JAGS

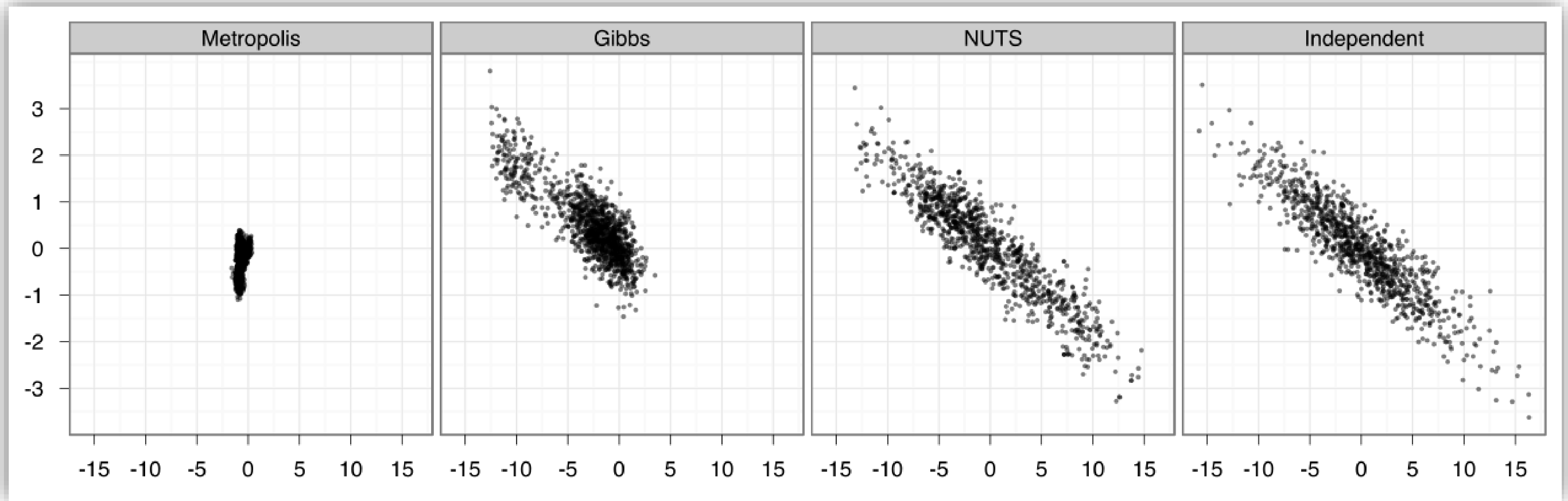
- Time to converge and per effective sample size:

  0.5 - ∞ times faster
- Memory usage: 1 - 10%
- Language features
  - variable overwrite: `a = 4, then a = 5`
  - formal control flow
  - full support of vectorizing

http://mc-stan.org/
https://discourse.mc-stan.org/t/selling-stan/3693/6

NUTS 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

http://mc-stan.org/

General Properties of Stan Language

- Whitespace does not matter
- Comments
  - //
  - /* … */
- Must use semicolon (;)
- Variables are typed and scoped

Variable's Scope

	data	transformed data	parameters	transformed parameters	model	generated quantities
Variable Declarations	Yes	Yes	Yes	Yes	Yes	Yes
Variable Scope	Global	Global	Global	Global	Local	Local
Variables Saved?	No	No	Yes	Yes	No	Yes
Modify Posterior?	No	No	No	No	Yes	No
Random Variables	No	No	No	No	No	Yes

Variable Declaration

- Each variable has a type (static type; scalar, vector, matrix etc.)
- Only values of that type can be assigned to the variable
  - e.g. cannot assign [1 2 3] to a (declared as a scalar)
- Declaration of variables happen at the top of a block (including local blocks)

Scalar Variables

cognitive model
statistics
computing

`real`

- scalar
- continuous

```
data {
  real y;
}
```

`int`

- scalar
- integer
- can't be used in `parameters` or `transformed parameters` blocks

```
data {
  int n;
}
```

Constraining Scalar Variables


```
data {
  int<lower=1> m;
  int<lower=0,upper=1> n;
  real<lower=0> x;
  real<upper=0> y;
  real<lower=-1,upper=1> rho;
}
```

Vector & Matrix

```
vector[3] a;
// column vector

row_vector[4] b;
// row vector

matrix[3,4] A;
// A is a 3x4 matrix
// A[1] returns a 4-element row vector

vector<lower=0,upper=1>[5] rhos;
row_vector<lower=0>[4] sigmas;
matrix<lower=-1, upper=1>[3,4] Sigma;
```

Control Flow

- if-else

```
if (cond) {
   ..statement..
} else {
   ..statement..
}
```

```
if (cond) {
   ..statement..
} else if (cond) {
   ..statement..
} else {
   ..statement..
}
```

- for-loop

```
for ( j in 1:n) {
   ..statement..
}
```

```
for ( j in 1:J ) {
   for ( k in 1:K ) {
      ..statement..
   }
}
```

same as the R syntax, but
terminate each line with ;

BERNOULLI MODEL

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 ϑ?

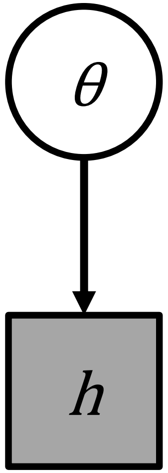
Kruschke (2015)

Bernoulli Model

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

$N = 1$

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



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

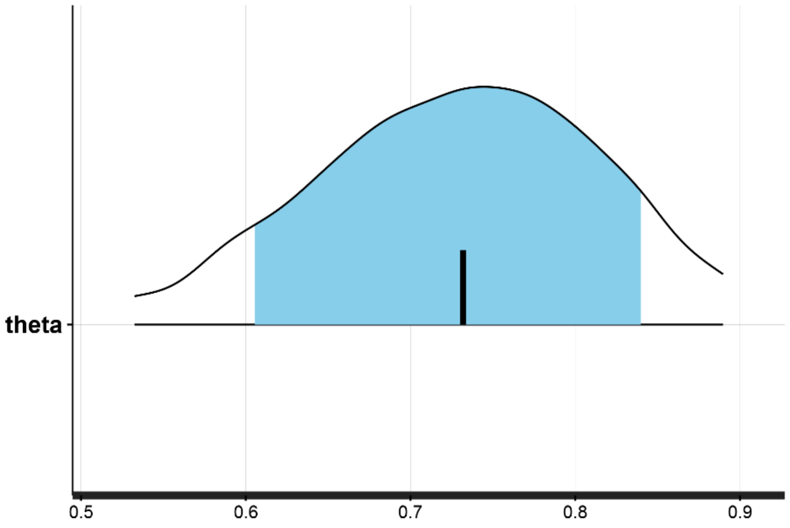
$h \sim \text{Bernoulli}\ (\theta)$

Exercise VIII

```
…/BayesCog/03.bernoulli_coin/_scripts/bernoulli_coin_main.R
```

TASK: fit the Bernoulli model

```
> dataList
$`flip`
 [1] 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1

$N
[1] 20
```



Possible Optimization?

```
model {
  for (n in 1:N) {
    flip[n] ~ bernoulli(theta);
  }
}
```

61.59 secs*

```
model {
  flip ~ bernoulli(theta);
}
```

53.25 secs*

Thinking before looping!

* compiling time included