Bayesian Statistics and Bayesian Cognitive Modeling

Lei Zhang
Institute of Systems Neuroscience, University Medical Center Hamburg-Eppendorf

Special thanks to Antonius Wiehler!

10-12. Oct. 2016, Hamburg
lei.zhang@uke.de

Schedule

cognitive model
statistics
computing

DAY1	9:00 – 17:00	Introduction R Basics Probability Basics Bayes' theorem MCMC and Stan Single Parameter Model – Binomial Model
DAY2	9:00 – 17:00	Multiple Parameter Model – Linear Regression Inference, Predictive Check Reinforcement Learning Model Hierarchical Models Optimizing Stan Codes Model Comparison
DAY3	9:00 – 13:00	Stan Style Tip and Debugging Model-Based fMRI Capstone Project: Delay Discounting Task

DAY1

09:00 – 09:30	Overview
09:30 – 10:00	R Basics
10:00 – 10:30	Probability basics
10:30 – 10:45	Coffee break
10:45 – 11:30	Bayes' theorem
11:30 – 12:30	Calculation examples
12:30 – 13:30	Lunch break
13:30 – 14:00	Linking data and parameter
14:00 – 15:00	Binomial model
15:00 – 15:15	Coffee Break
15:15 – 16:15	MCMC and Stan
16:15 – 17:00	Binomial model in Stan

Overview

What is your experience with…

- Statistics?
- R?
- Cognitive Modeling?

You would like to…

- gain knowledge of Bayesian stats?
- be able to read “computational modeling” section in papers?
- write your own model?

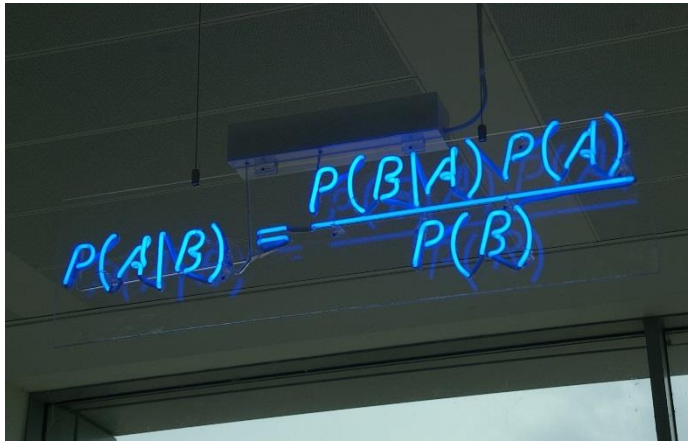
Overview

This workshop is NOT about…

… Bayes in the brain

… Cognitive process that are Bayesian themselves

… Bayesian statistics to supersede classical statistics



However, Bayesian statistics offer great tools to analyze cognitive processes!

- Construct cognitive models
- Estimate posterior distributions of parameters
- Compare models: which is the best one, given the data?

How to Get the Most out of the Workshop

- Ask questions
- Try the exercises
- Talk to each other & help each other

BASICS OF R PROGRAMMING

RStudio®

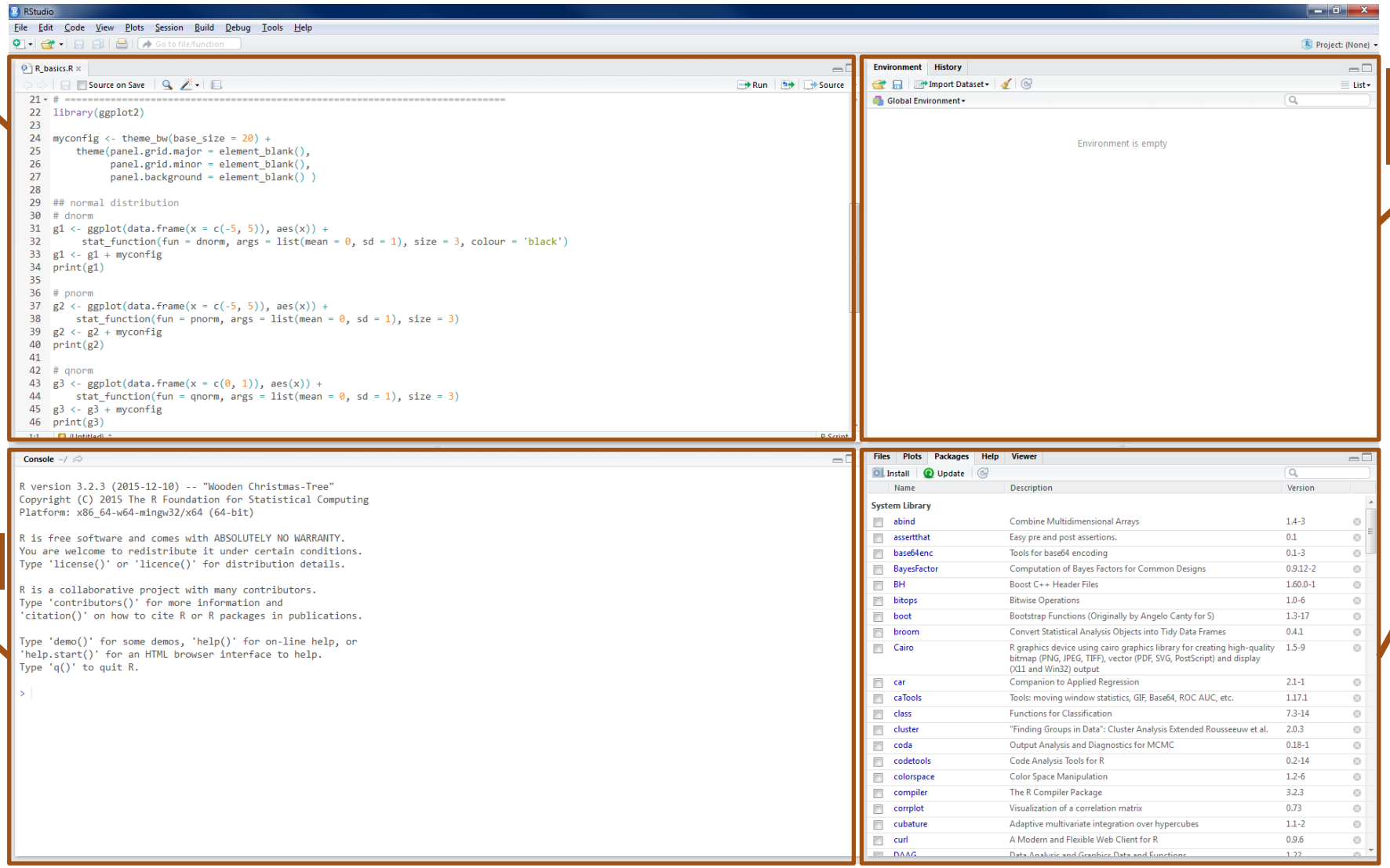
R Basics

cognitive model
statistics
computing

- R
  - a programming language for statistical computing
  - R has its own user interface

- R Studio
  - integrated development environment (IDE) for R
  - a more sophisticated R-friendly editor, with helpful syntax highlight

script editor



console

environment/
command history

file/pkg/img/
etc.

Basic Commands


```
getwd()
setwd('E:/teaching/BayesCog/')
dir()
ls()
print('Hello World!')
cat('Hello', 'World!')
paste0('C:/', 'Group1')
help(func)
? func
a <- 5
a = 5
save(varname, file = "pathname/varname.RData")
load("pathname/varname.RData")
rm(list = ls())
q()
```

RStudio - Shortcuts

cognitive model
statistics
computing

Ctrl + L: clean console

Ctrl + Shift + N: create a new script

↑: command history

Ctrl(hold) + ↑: command history with certain starts

Ctrl + Enter: execute selected scripts

Editor - Shortcuts

Ctrl + home/Pos: go to very top of a script

Ctrl + end/Ende: go to very top of a script

Shift(hold) + ↑/↓: select line(s)

Ctrl(hold) + ←/→: select word(s)

Data Classes

numeric: 1.1 2.0

integer: 1 2 3

character / string: "hello world!"

logical: TRUE FALSE

factors: "male" / "female"

(complex: 1+2i)

Data Types

cognitive model
statistics
computing

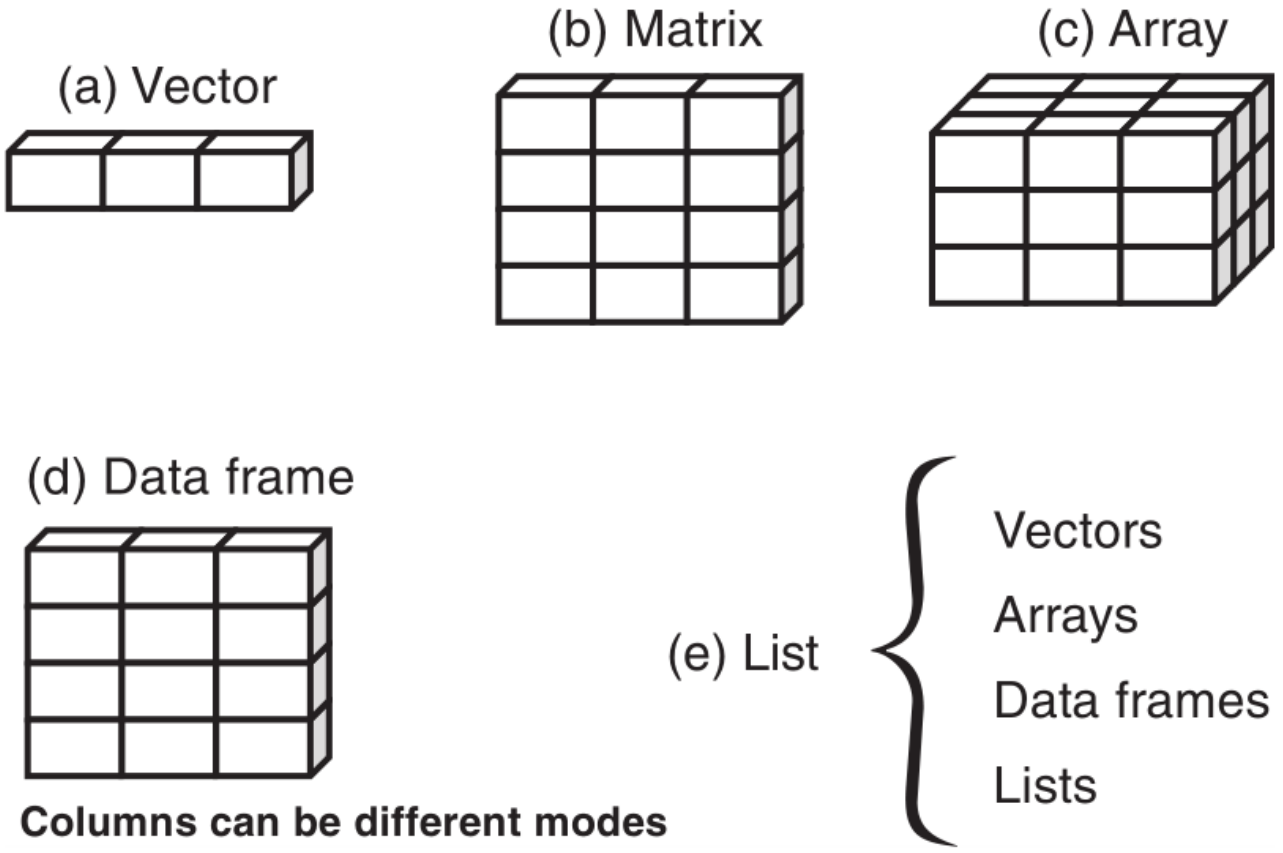
Vector

Matrix

Array

Data Frame

List



Kabacoff (2015)

Exercise I

cognitive model
statistics
computing

`…/BayesCog/01.R_basics/_scripts/R_basics.R`

`up to “Flow Control”`

TASK: practise basic R commands and data type

TIP: `class(), str()`

Flow Control

- if-else

```
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..
   }
}
```

User-defined Function

cognitive model
statistics
computing

```
funname <- function (input_arges) {
    .. function body ..
    .. function body ..
    return(output_arges)
}
```

$$sem = \sqrt{\frac{s^2}{n-1}}$$

```
sem <- function(x) {
    sqrt( var(x,na.rm=TRUE) / (length(na.omit(x))-1) )
}
```

Exercise II

cognitive model
statistics
computing

```
…/BayesCog/01.R_basics/_scripts/R_basics.R
```

TASK: practise flow control and user-defined function

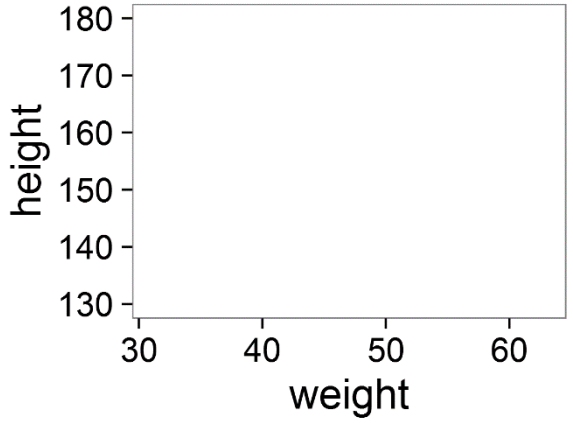
Brief Intro to ggplot2

cognitive model
statistics
computing

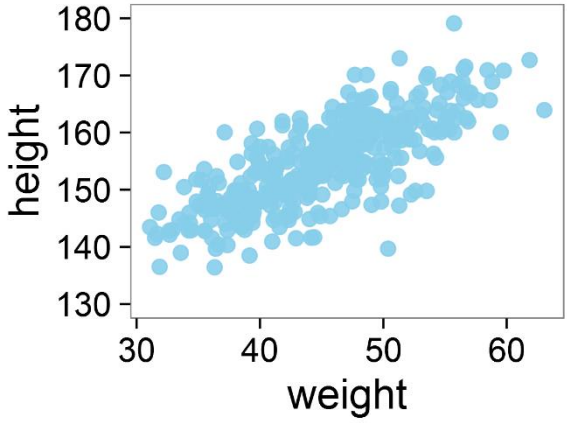
```
plot = geometric (points, lines, bars) + aesthetic (color, shape, size)
```

game of adding layers!

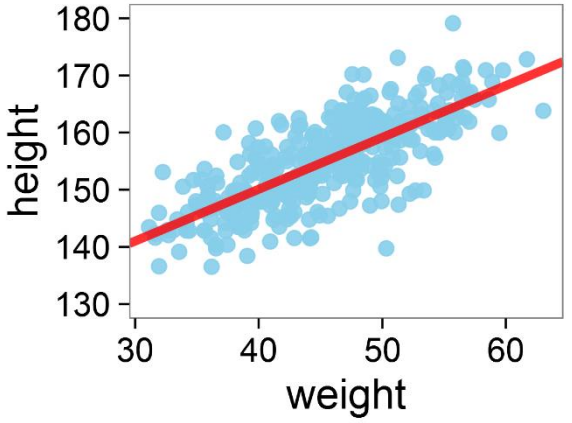
background



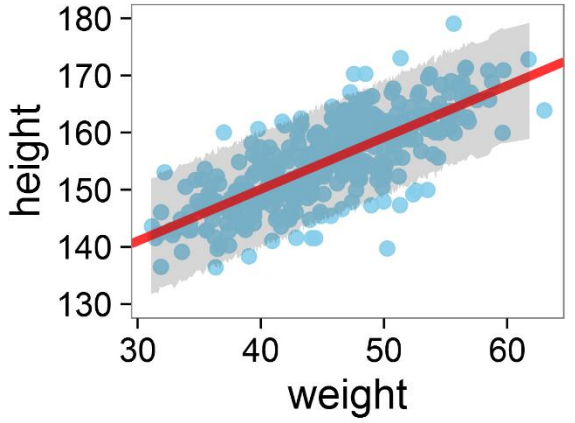
add scatters



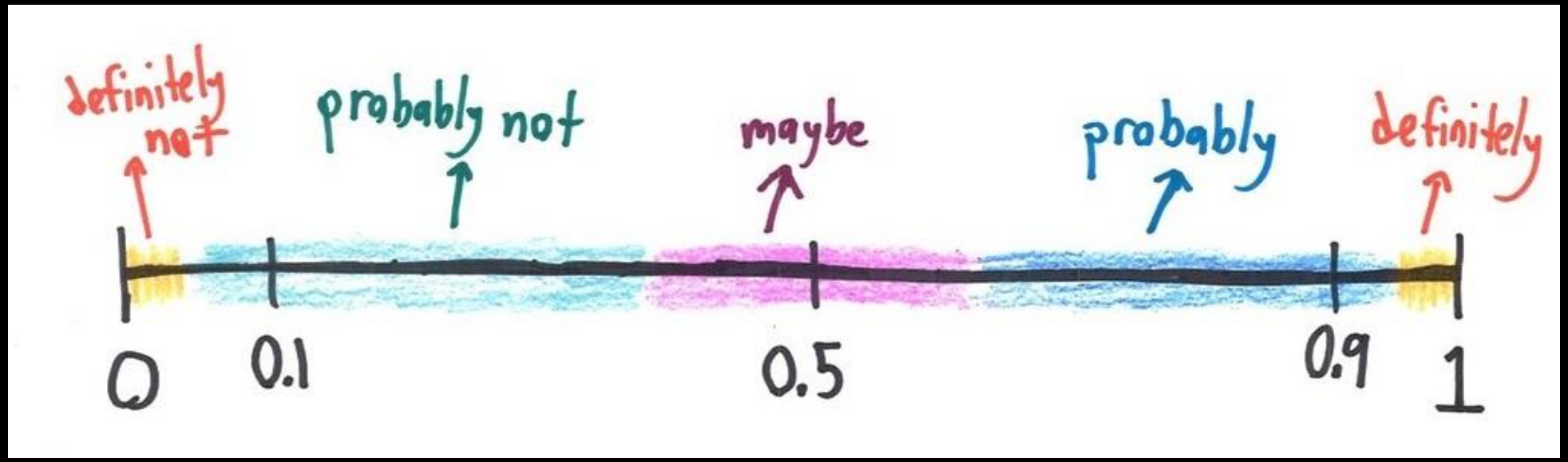
add regression line



add shaded errors



BASICS OF PROBABILITY



Probability

cognitive model
statistics
computing

…assigning numbers to a set of possibilities

Properties (Kolmogorov, 1956)

- $p \in [0,1]$
- $\Sigma p = 1$
- $p(\mathrm{AUB}) = p(\mathrm{A}) + p(\mathrm{B})$, when A and B are *mutually exclusive*

Joint Probability and Conditional Probability

cognitive model
statistics
computing

Joint Probability

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

- e.g., $p(\text{raining})$ and $p(\text{cold})$

Conditional Probability

$p(A|B)$ – 'p of A given B'

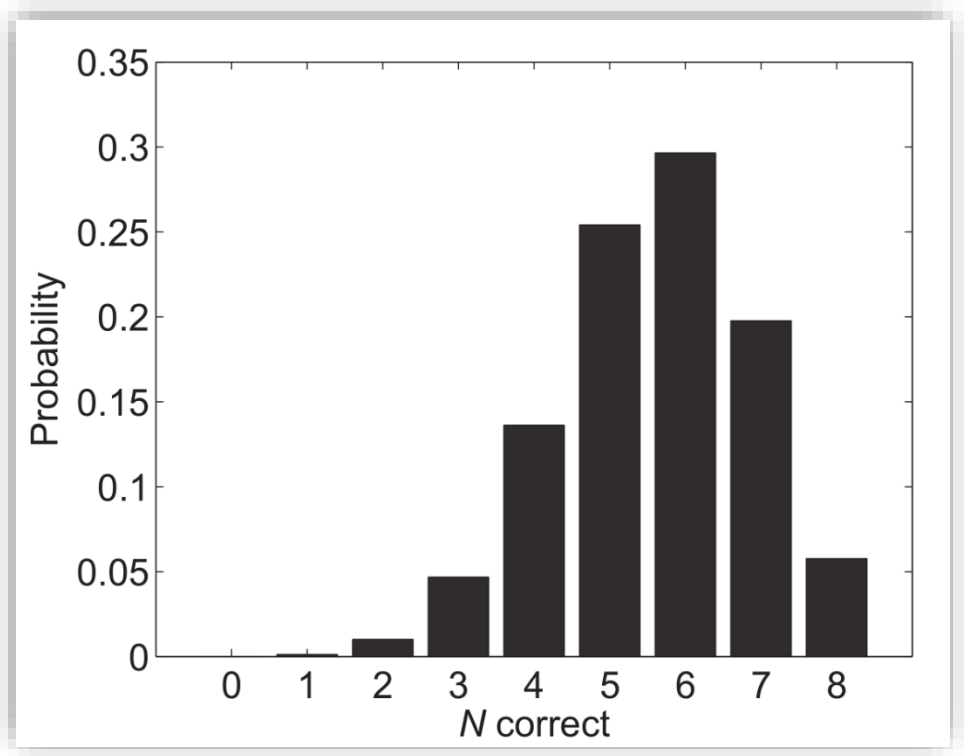
$p(A,B) = p(A|B)p(B)$, when A and B are *independent*

- e.g., $p(\text{raining, cold}) = p(\text{raining}|\text{cold})p(\text{cold})$

Probability Functions

cognitive model
statistics
computing

discrete events



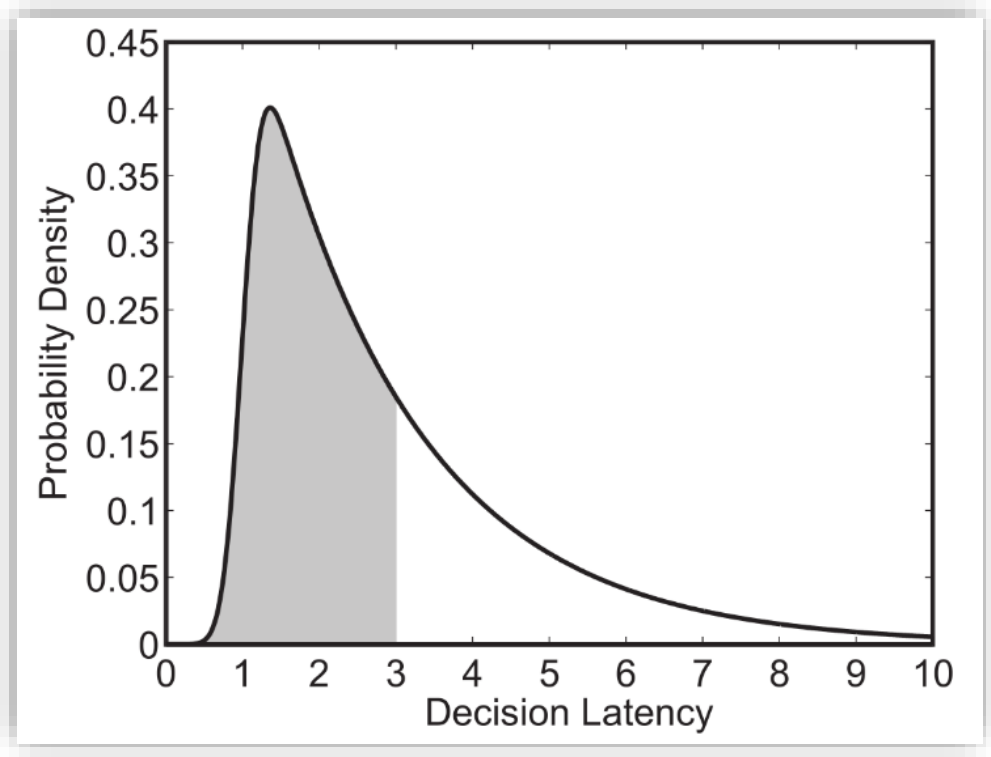
Lewandowsky & Farrell (2011)

Probability Functions

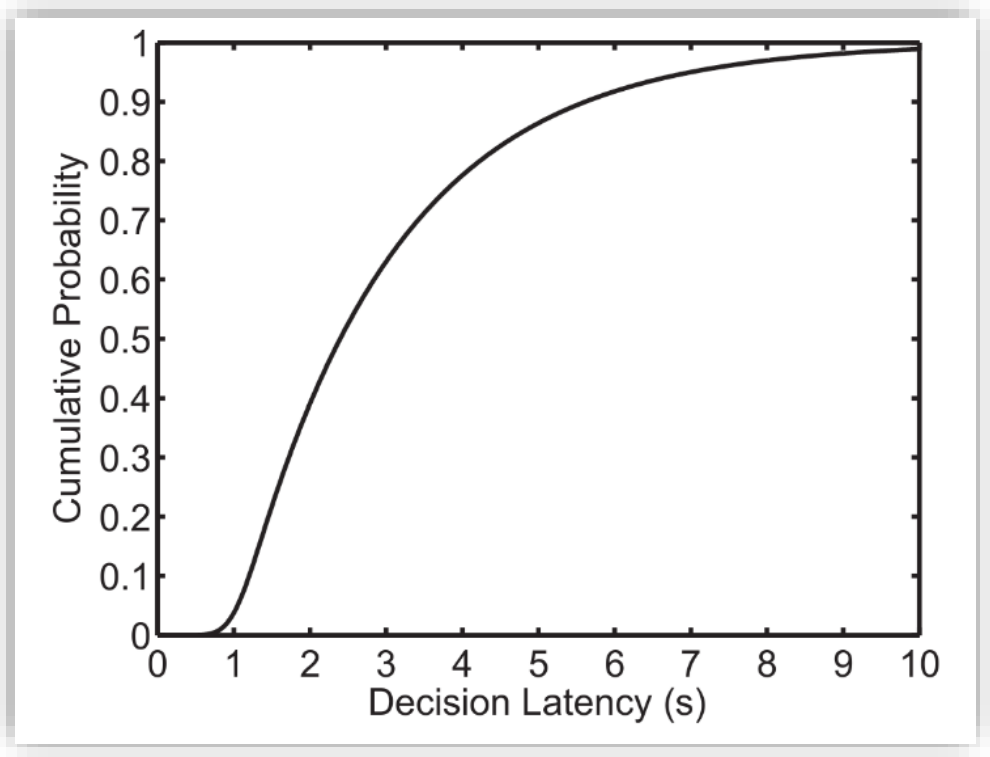
cognitive model
statistics
computing

continuous events

probability density function (PDF)



cumulative distribution function (CDF)



Lewandowsky & Farrell (2011)

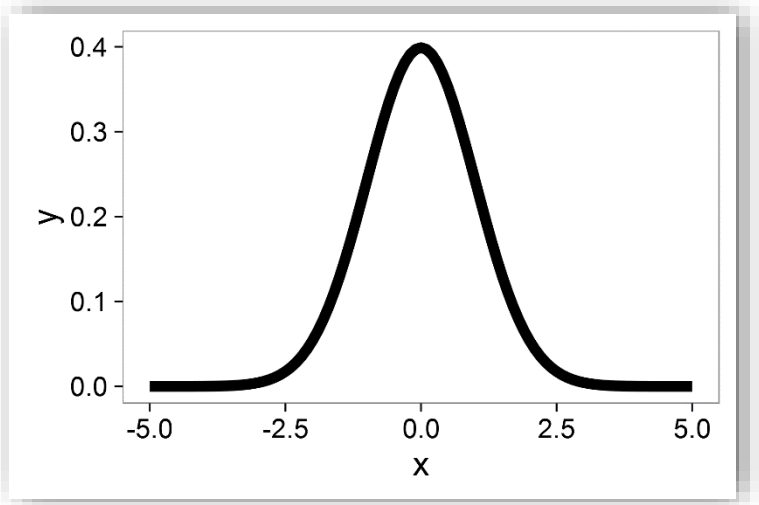
Playing with Probability Functions in R

cognitive model
statistics
computing

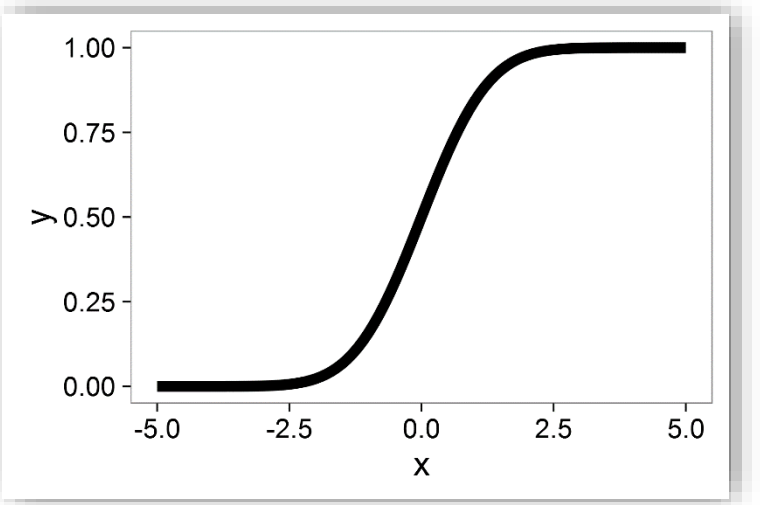
```
dnorm() - PDF
pnorm() - CDF
qnorm() - quantile, inverse cdf
rnorm() - random number generator
```

Example: Normal(0,1)

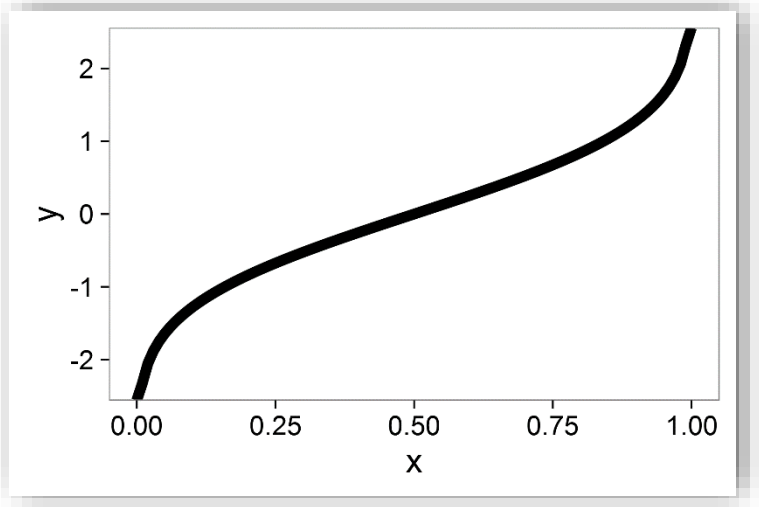
dnorm()



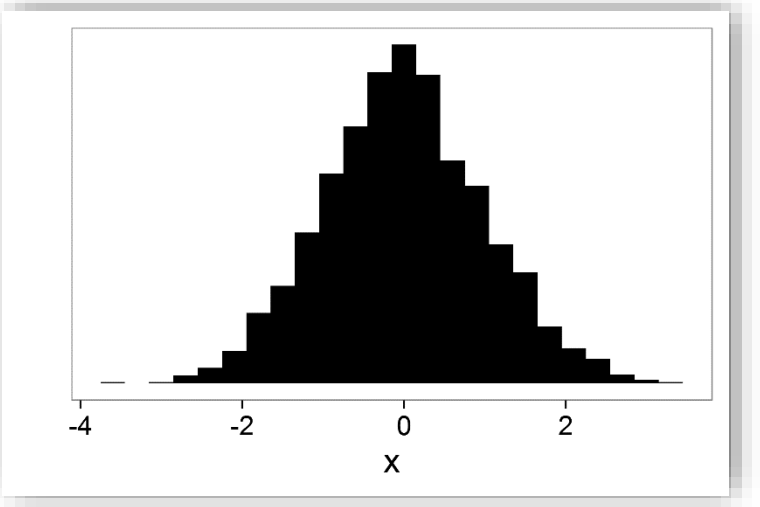
pnorm()



qnorm()



rnorm()



Exercise III

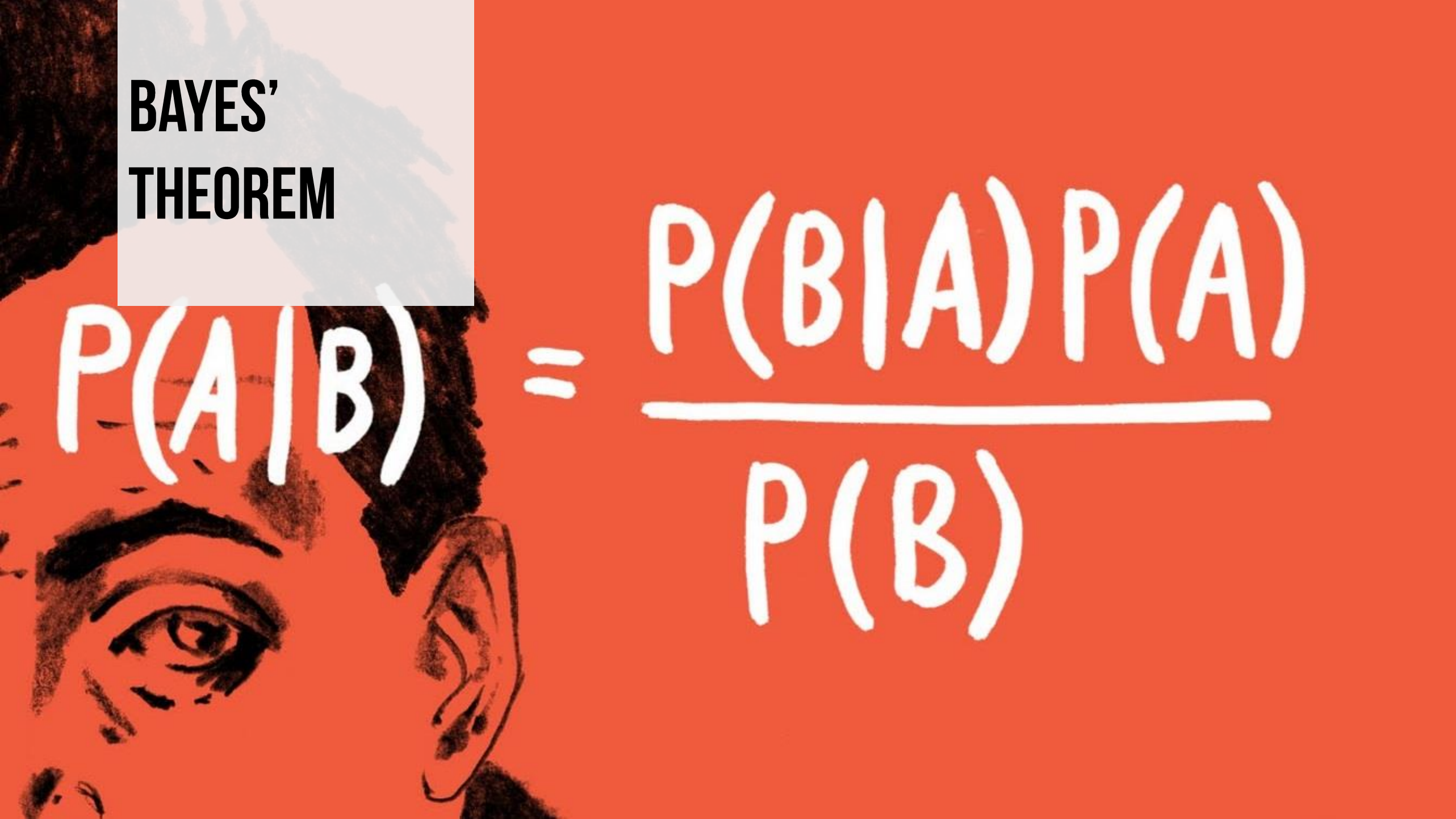
cognitive model
statistics
computing

`…/BayesCog/01.R_basics/_scripts/R_basics.R`

TASK: produce `dnorm`, `pnorm`, `qnorm`, `rnorm` figures for

$$\text{Normal}(0.5, 2)$$

BAYES' THEOREM



Bayes' theorem

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

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

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

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

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

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{\sum_{A^*} p(B \mid A^*)p(A^*)}$$

A^* is a variable that takes on all possible values

One Example

	Column			
Row	...	*c*	...	Marginal
⋮		⋮		
r	...	$p(r, c) = p(r\|c)\, p(c)$	...	$p(r) = \sum_{c^*} p(r\|c^*)\, p(c^*)$
⋮		⋮		
Marginal		$p(c)$		

Kruschke (2015)

One Example

Eye color	Hair color: Black	Brunette	Red	Blond	Marginal (Eye color)
Brown	0.11	0.20	0.04	0.01	0.37
Blue	0.03	0.14	0.03	0.16	0.36
Hazel	0.03	0.09	0.02	0.02	0.16
Green	0.01	0.05	0.02	0.03	0.11
Marginal (hair color)	0.18	0.48	0.12	0.21	1.0

Eye color	Hair color: Black	Brunette	Red	Blond	Marginal (Eye color)
Blue	0.03/0.36 = 0.08	0.14/0.36 = 0.39	0.03/0.36 = 0.08	0.16/0.36 = 0.45	0.36/0.36 = 1.0

Kruschke (2015)

Exercise IV

Suppose that in the general population, the probability of having a rare disease is 1/1000. We denote the true presence or absence of the disease as the value of a parameter, ϑ, that can have the value ϑ = ☹ if disease is present in a person, or the value ϑ = ☺ if the disease is absent. The base rate of the disease is therefore denoted $p(\vartheta = ☹) = 0.001$.

Suppose(1): a test for the disease that has a 99% hit rate: $p(T = +|\vartheta = ☹) = 0.99$

Suppose(2): the test has a false alarm rate of 5%: $p(T = +|\vartheta = ☺) = 0.05$

Q: Suppose we sample a person at random from the population, administer the test, and it comes up positive. What is the posterior probability that the person has the disease?

Exercise IV

Q: What is the posterior probability that the person has the disease?

→ $p(\vartheta = ☹ \mid T = +)$

Exercise IV

Test result	Disease: $\theta =$ ☹ (present)	Disease: $\theta =$ ☺ (absent)	Marginal (test result)
$T = +$	$p(+\|☹)\,p(☹)$ $= 0.99 \cdot 0.001$	$p(+\|☺)\,p(☺)$ $= 0.05 \cdot (1 - 0.001)$	$\sum_\theta p(+\|\theta)\,p(\theta)$
$T = -$	$p(-\|☹)\,p(☹)$ $= (1 - 0.99) \cdot 0.001$	$p(-\|☺)\,p(☺)$ $= (1 - 0.05) \cdot (1 - 0.001)$	$\sum_\theta p(-\|\theta)\,p(\theta)$
Marginal (disease)	$p(☹) = 0.001$	$p(☺) = 1 - 0.001$	1.0

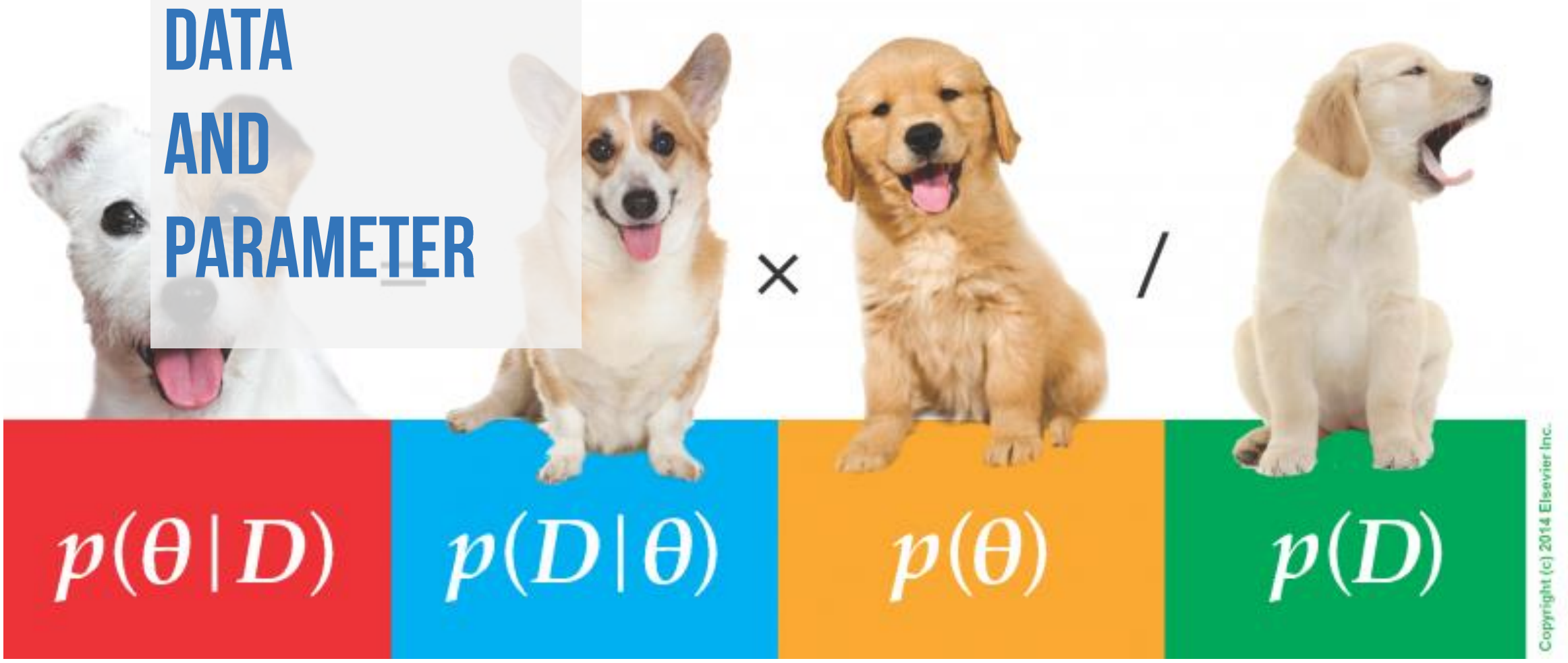
$$p(\theta = ☹ \,|\, T = +) = \frac{p(T = +|\theta = ☹)\,p(\theta = ☹)}{\sum_\theta p(T = +|\theta)\,p(\theta)}$$

$$= \frac{0.99 \cdot 0.001}{0.99 \cdot 0.001 + 0.05 \cdot (1 - 0.001)}$$

$$= 0.019$$

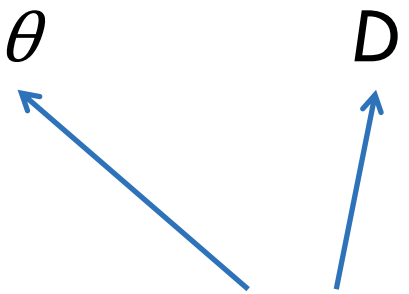
Kruschke (2015)

LINKING DATA AND PARAMETER



Linking Data and Parameter

cognitive model
statistics
computing



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

Linking Data and Parameter

cognitive model
statistics
computing

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

Linking Data and Parameter

cognitive model
statistics
computing

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 is *p*(*Data*)?

cognitive model
statistics
computing

discrete parameters

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

continuous parameters

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

BINOMIAL MODEL

Binomial Model

cognitive model
statistics
computing

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

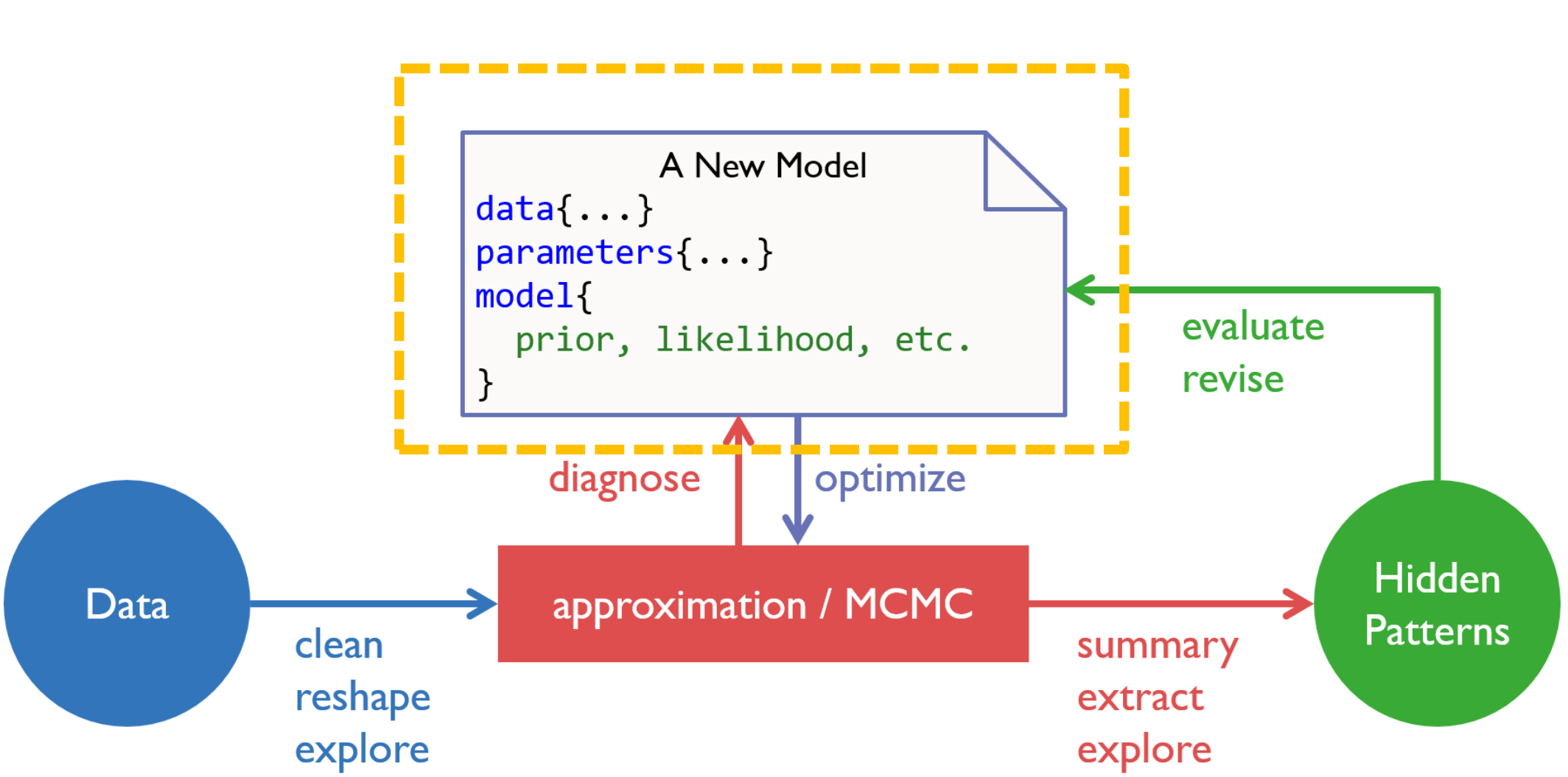
McElreath (2016)

Steps of Bayesian Modeling?

cognitive model
statistics
computing

A data story	Think about how the data might arise. It can be *descriptive* or even *causal*.
Update	Educate your model by feeding it the data. Bayesian Update: update the prior, in light of data, to produce posterior.
Evaluate	Compare model with reality. Revise your model.

McElreath (2016)


cognitive model
statistics
computing
A New Model
data{...}
parameters{...}
model{
prior, likelihood, etc.
}
evaluate
revise
diagnose
optimize
Data
clean
reshape
explore
approximation / MCMC
summary
extract
explore
Hidden
Patterns

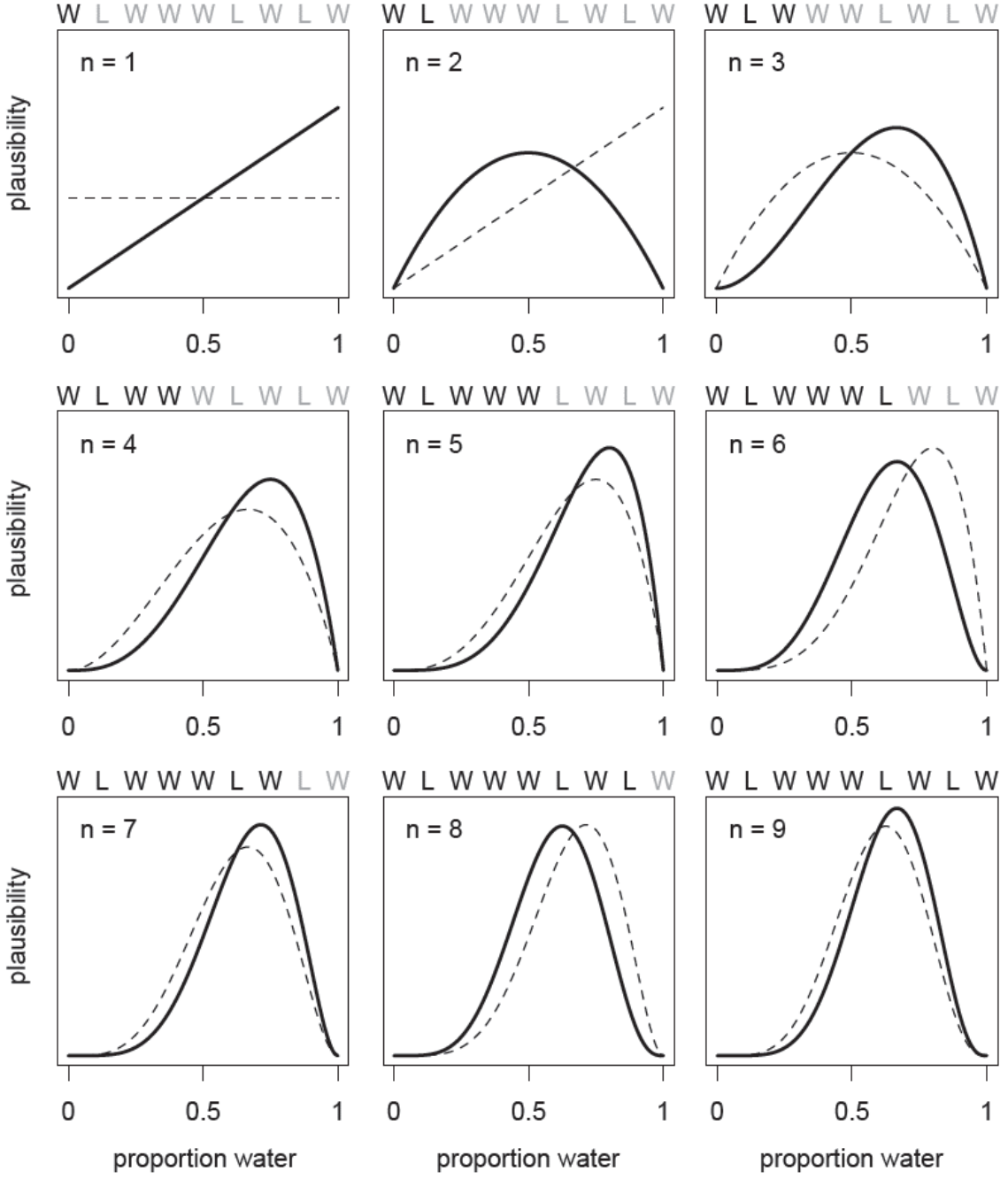
A Data Story of the Globe

cognitive model
statistics
computing

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

McElreath (2016)

Update



cognitive model

statistics

computing

McElreath (2016)

Components of a Model

cognitive model
statistics
computing

think about the likelihood function:

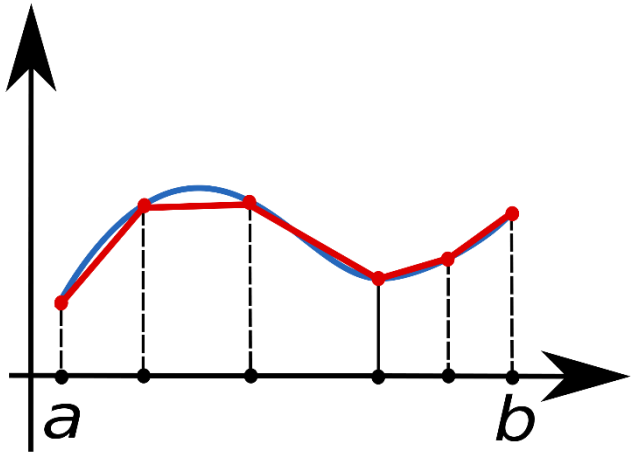
$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\int p(D \mid \theta^*)\, p(\theta^*)\, d\theta^*}$$

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

Binomial Model – Grid Approximation

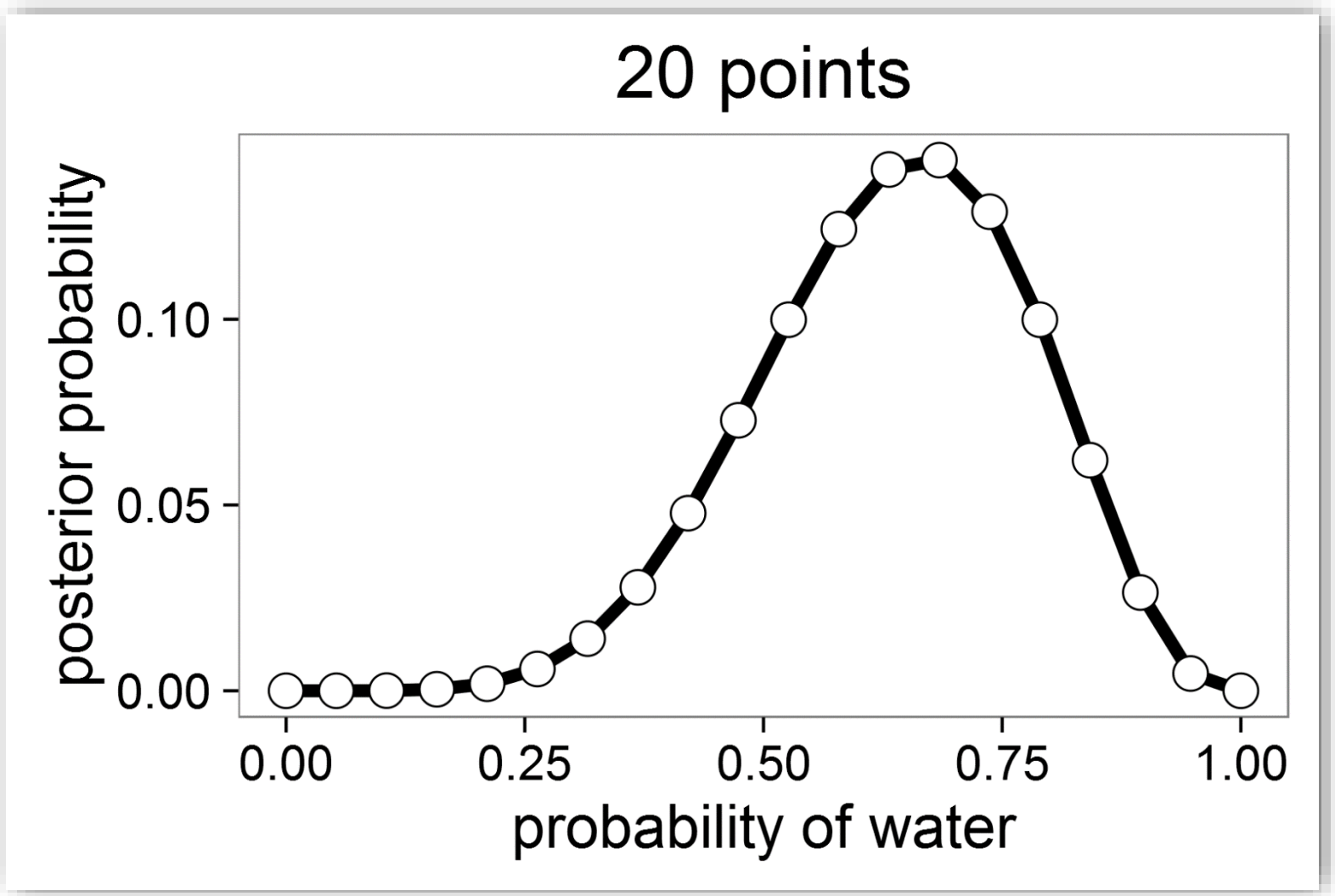
cognitive model
statistics
computing

```
p_start <- 0; p_end <- 1; n_grid <- 20
w <- 6; N <- 9

# define grid
p_grid <- seq( from = p_start ,
          to = p_end , length.out = n_grid )

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

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

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

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



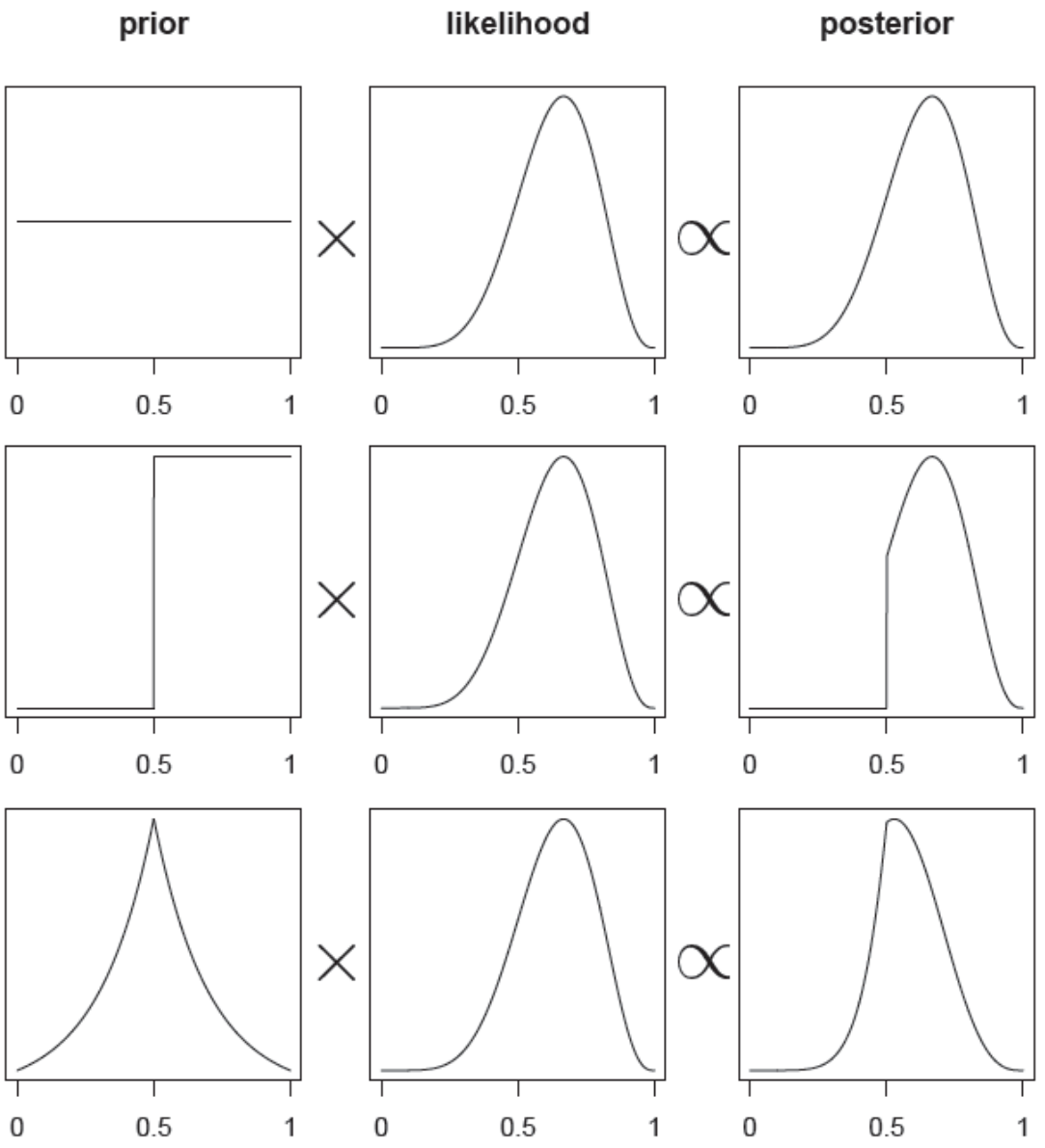
Binomial Model – Grid Approximation

cognitive model
statistics
computing



Impact of Prior

cognitive model
statistics
computing



McElreath (2016)

Exercise V

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

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

Components of a Model

cognitive model
statistics
computing

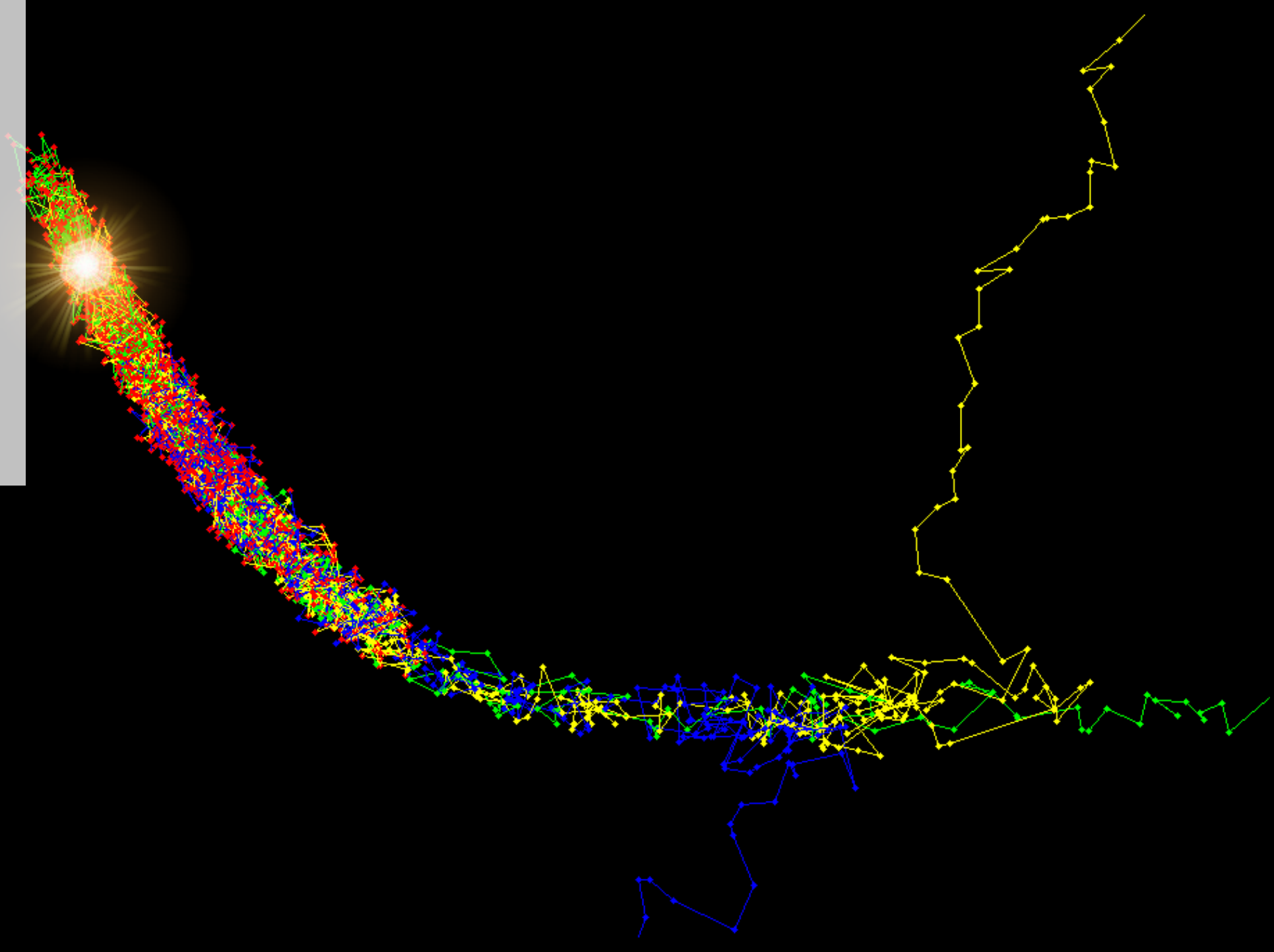
grid approximation for
2 parameters?
5 parameters?
10 parameters

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

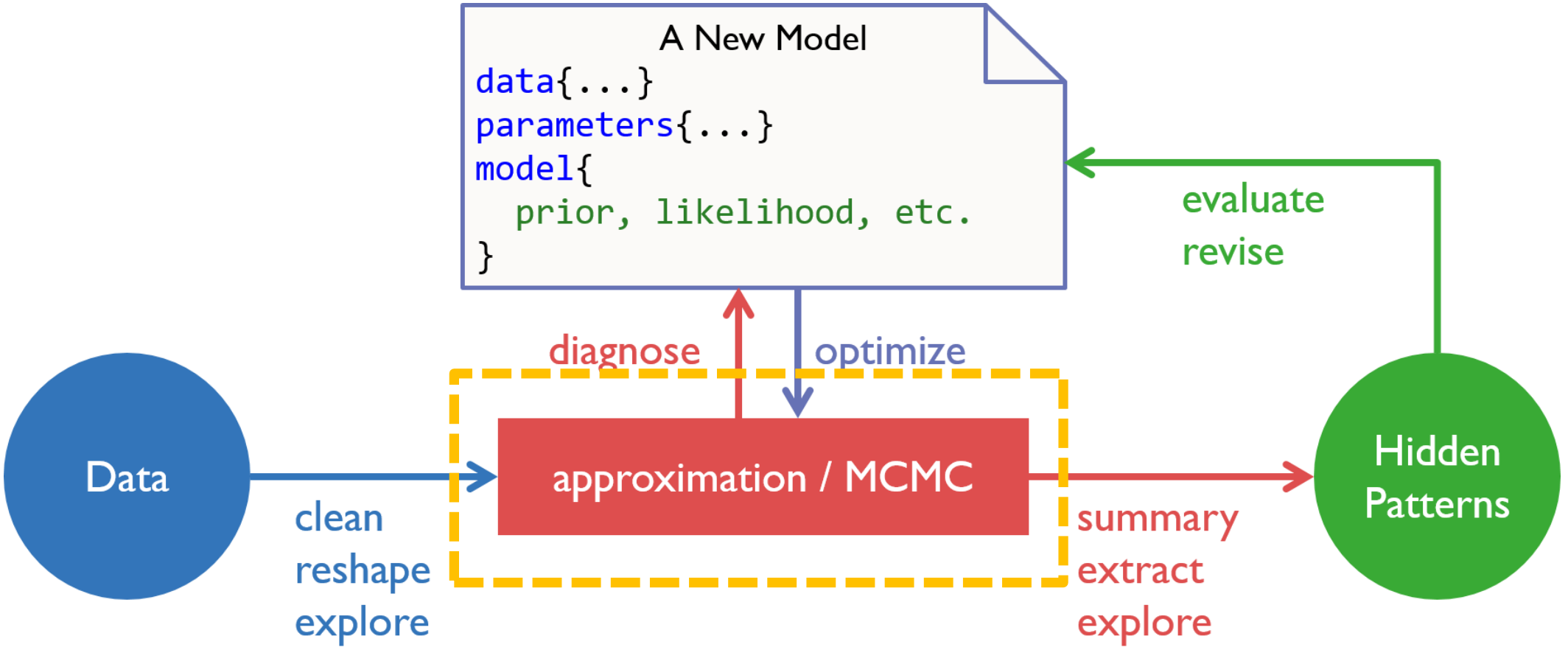
$$p(w \mid N, p) = \binom{N}{w} p^w (1-p)^{N-w}$$

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

MARKOV CHAIN MONTE CARLO



cognitive model
statistics
computing



Solving the Problem by Approximation

cognitive model
statistics
computing

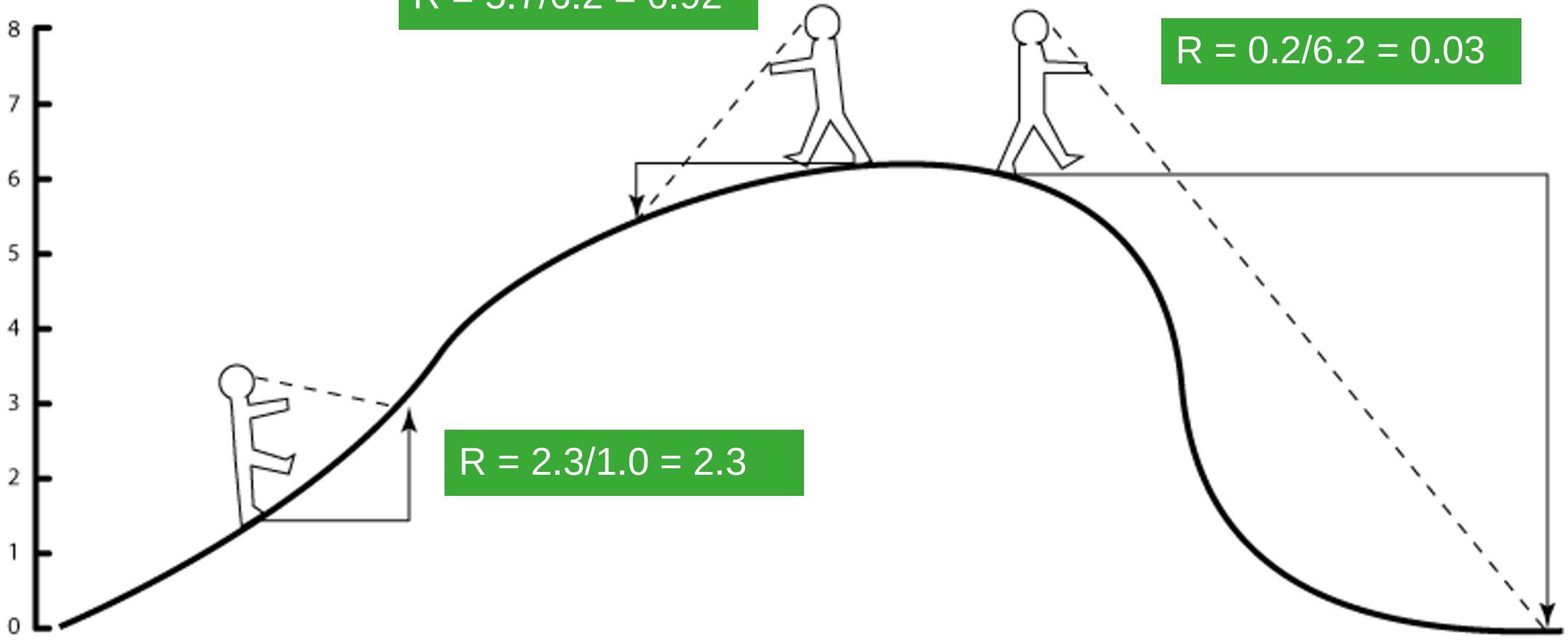
$$p(\theta \mid D) \propto p(D \mid \theta)p(\theta)$$

Deterministic Approximation	Stochastic Approximation
→ Variational Bayes	→ Sampling Methods

An MCMC Robot

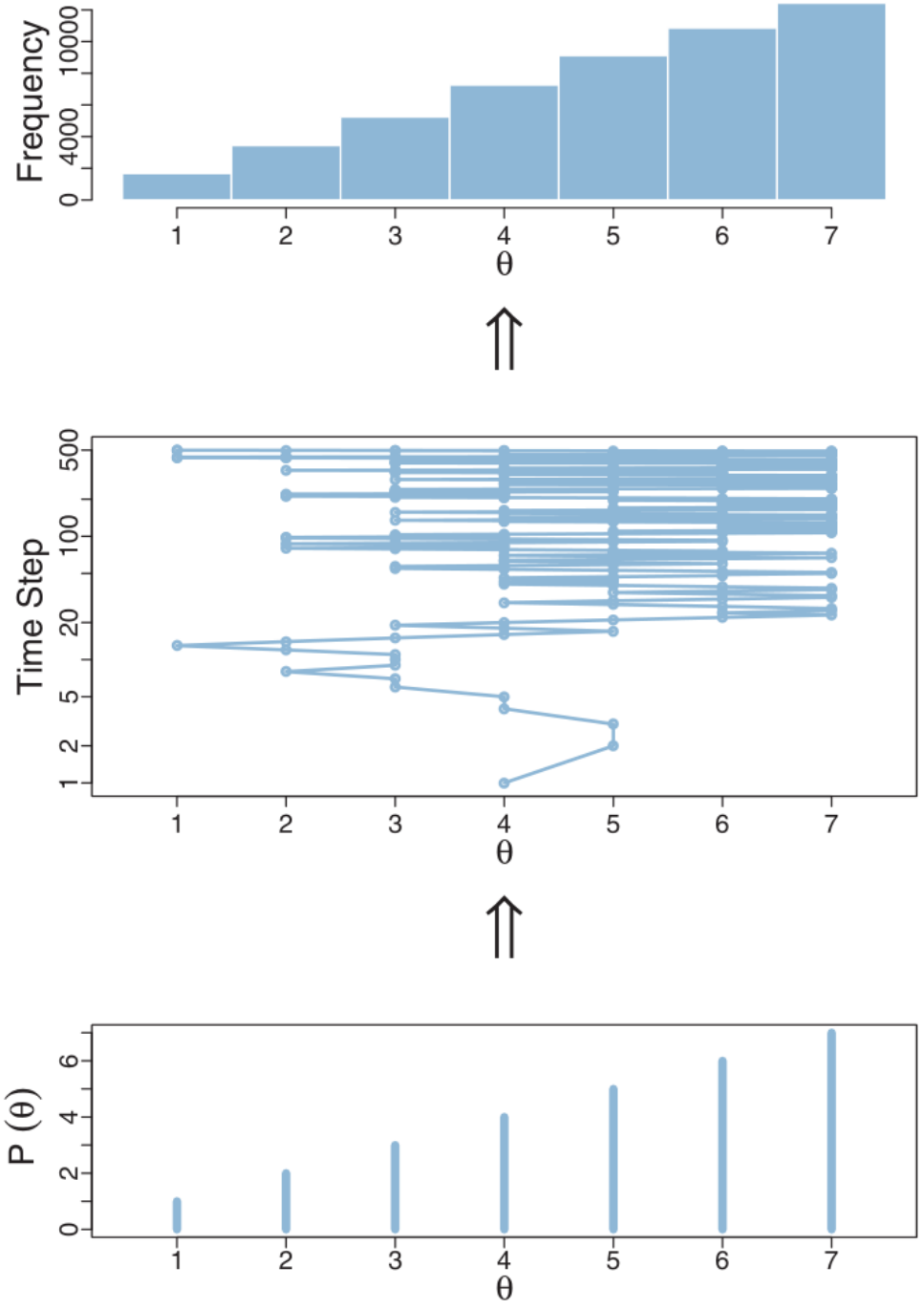
cognitive model
statistics
computing

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



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

Sampling Example



cognitive model

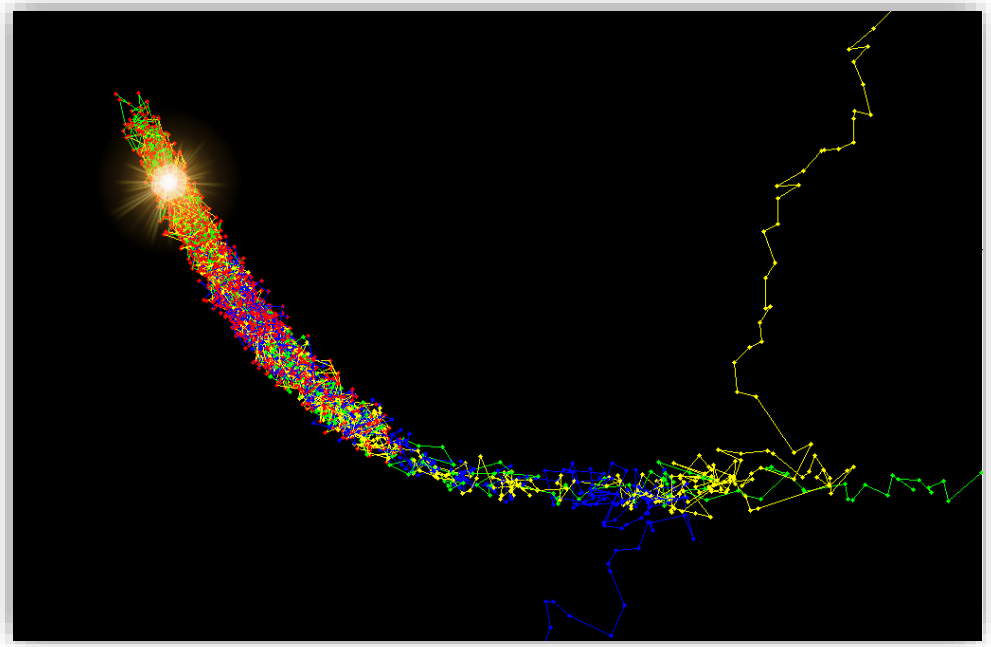
statistics

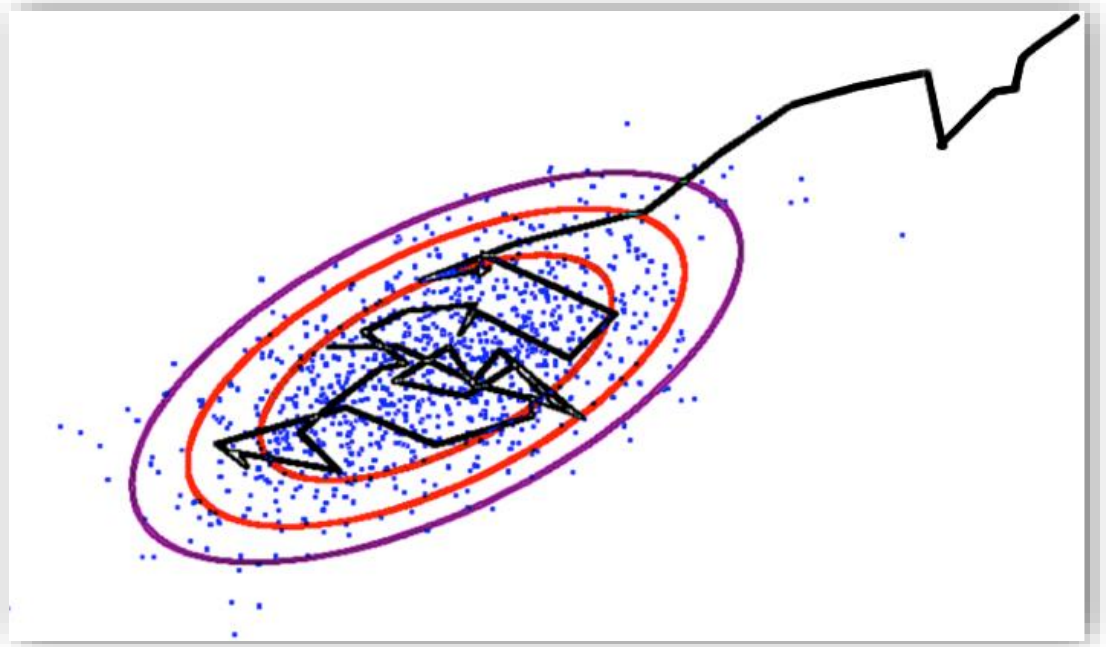
computing

Kruschke (2015)

Visual Example

cognitive model
statistics
computing





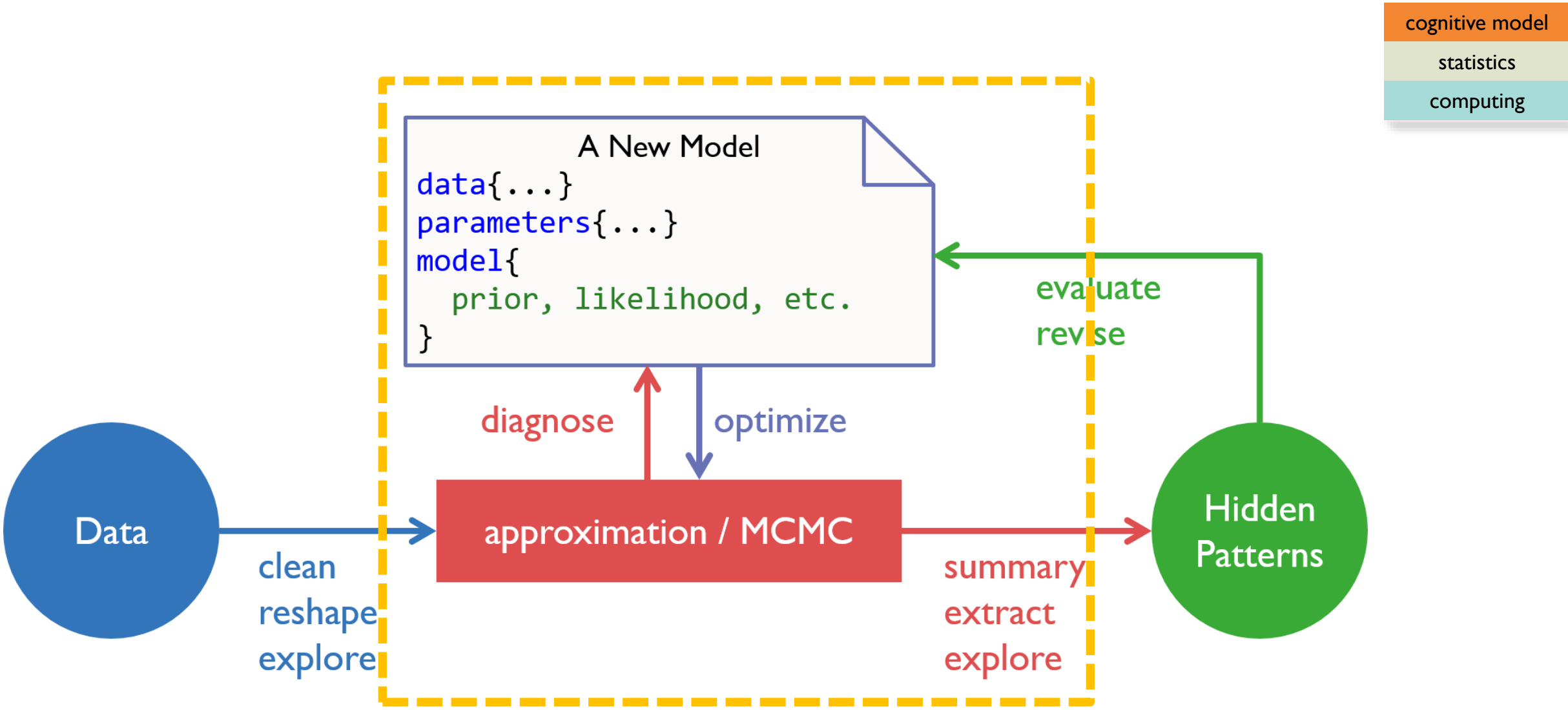
MCMC Sampling Algorithms

cognitive model
statistics
computing

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

Stan!

STAN PROGRAMMING LANGUAGE


cognitive model
statistics
computing
A New Model
data{...}
parameters{...}
model{
prior, likelihood, etc.
}
evaluate
revise
diagnose
optimize
Data
clean
reshape
explore
approximation / MCMC
summary
extract
explore
Hidden
Patterns

Why Use Stan?

cognitive model
statistics
computing

vs. BUGS and JAGS

- Time to converge and per effective sample size:

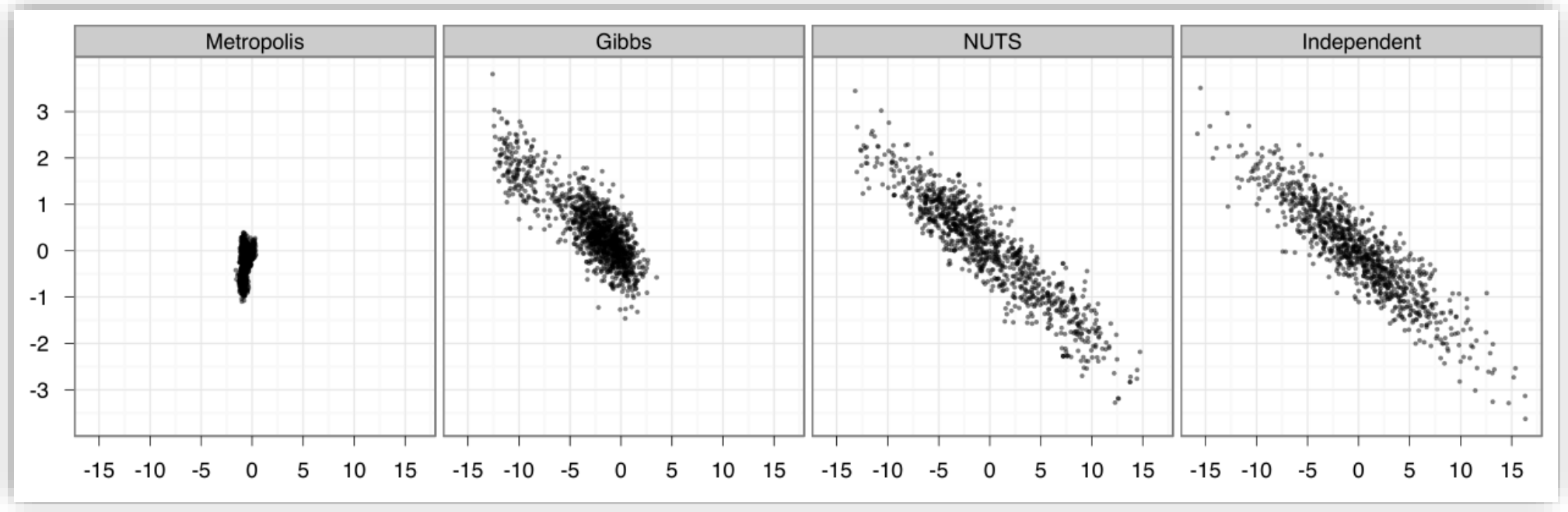
  0.5 - ∞ times faster

- Memory usage: 1 - 10%
- Language features
  - variable overwrite: `a = 4, then a = 5`
  - formal if-else
  - full support of vectorizing

NUTS vs. Gibbs and Metropolis

cognitive model
statistics
computing

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

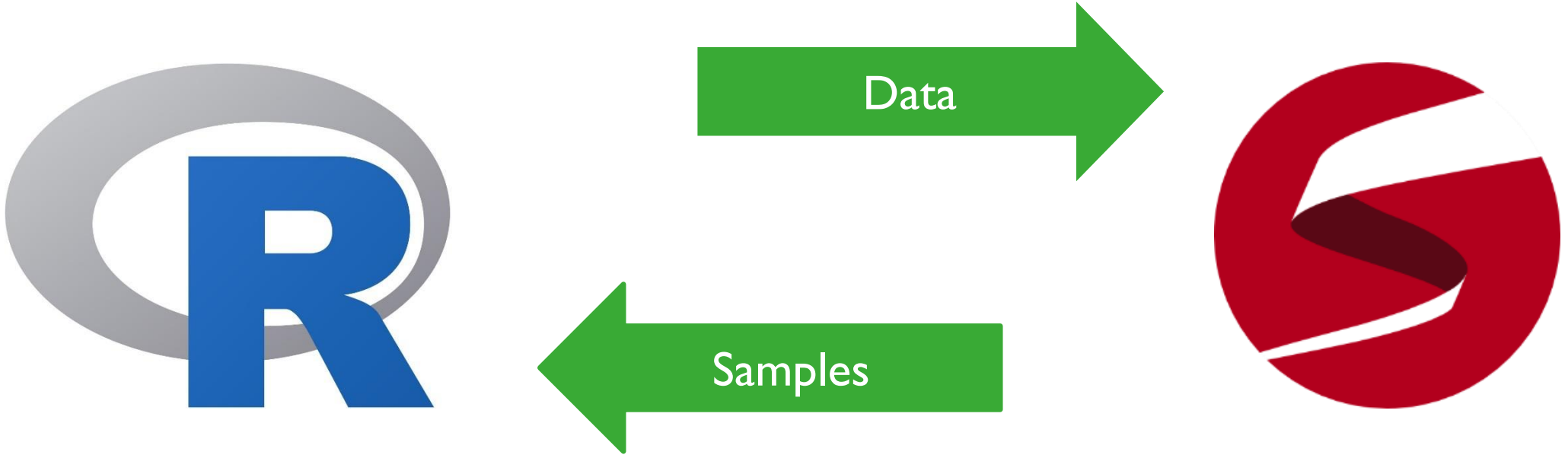


- Two dimensions of highly correlated 250-dim normal
- 1,000,000 draws from Metropolis and Gibbs (thin to 1000)
- 1,000 draws from NUTS; 1000 independent draws

http://mc-stan.org/

Stan and RStan

cognitive model
statistics
computing



Steps of Bayesian Modeling, with Stan

cognitive model
statistics
computing

A data story

Think about how the data might arise.
It can be *descriptive* or even *causal*.
Write a Stan program.

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

cognitive model
statistics
computing

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 (Stan 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 ...
}
```

cognitive model

statistics

computing

General Properties of Stan Language

cognitive model
statistics
computing

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

Variable Declaration

cognitive model
statistics
computing

- Each variable has a type (static type)
- Only values of that type can be assigned to the variable
- Declaration of variables happen at the top of a block (including local blocks)

Scalar Variables

cognitive model
statistics
computing

`real`

- scalar
- continuous

`int`

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

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

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

Constraining Scalar Variables

cognitive model
statistics
computing

```
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-row_vector

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

Flow Control

- if-else

```
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..
   }
}
```

REVISIT BINOMIAL MODEL

Binomial Model

cognitive model
statistics
computing

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*.

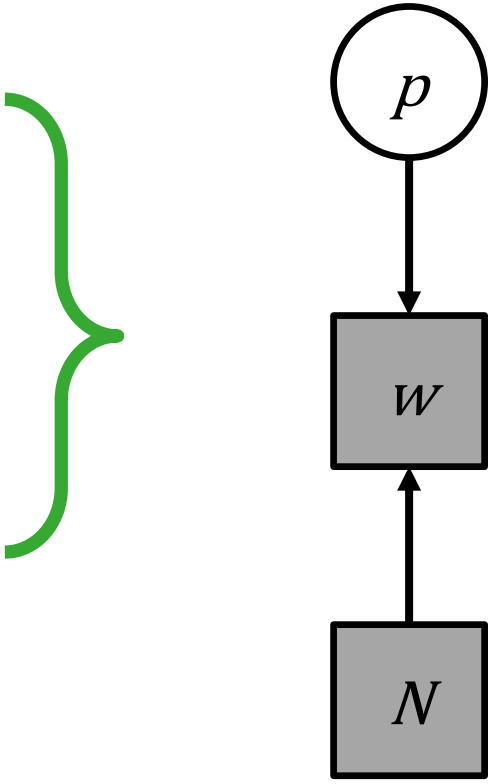
McElreath (2016)

Binomial Model

cognitive model
statistics
computing

W L W W W L W L W

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



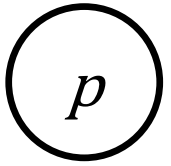
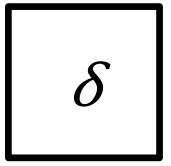
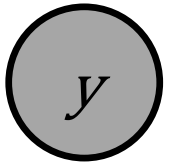
$$p \sim \text{Uniform}(0, 1)$$

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

McElreath (2016)

Graphical Model Notations

cognitive model
statistics
computing

	continuous	discrete
unobserved		
observed		

Lee & Wagenmakers (2013)

Binomial Model

cognitive model
statistics
computing

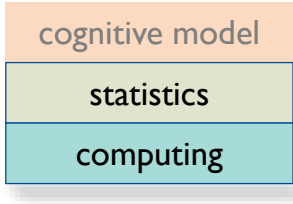
W L W W W L W L W

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

```
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
version.string R version 3.3.1 (2016-06-21)

> stan_version()
[1] "2.12.0"
```

Model Summary

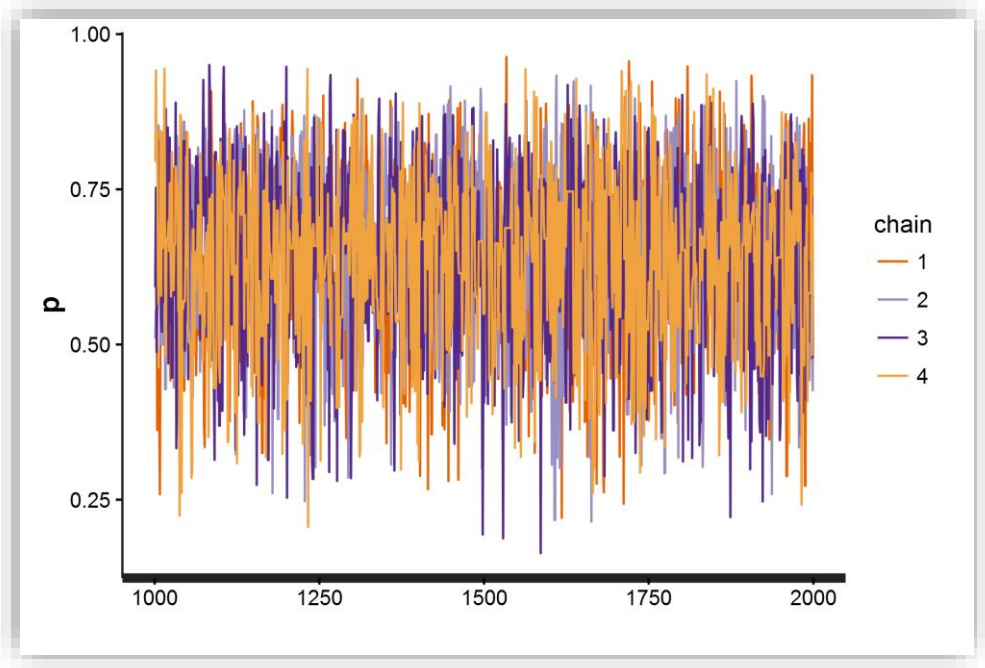
cognitive model
statistics
computing

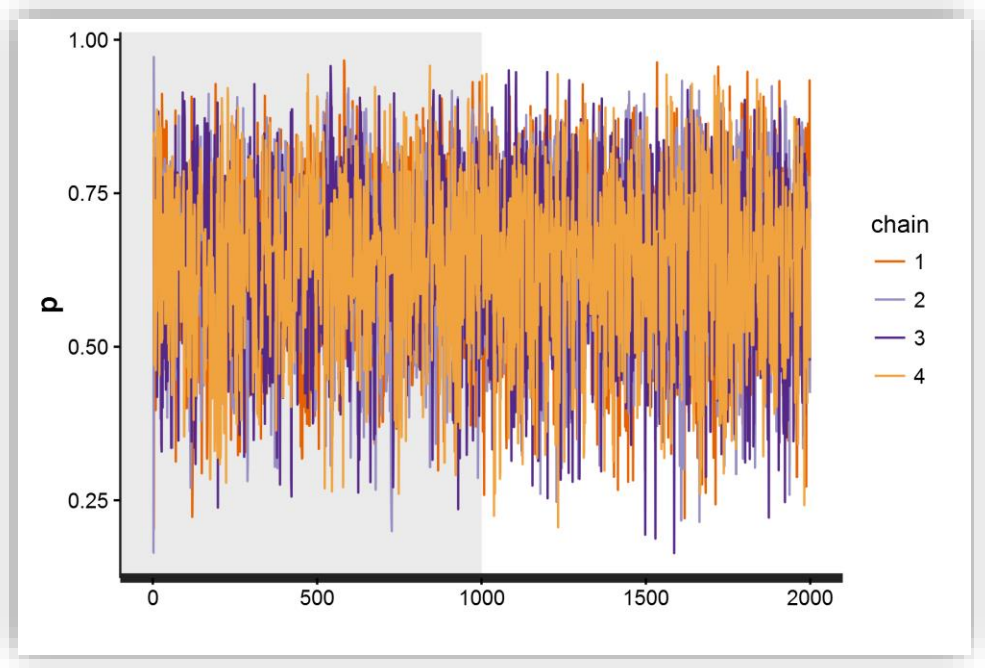
```
> print(fit_globe)
Inference for Stan model: binomial_globe_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

      mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
p     0.64    0.00 0.14  0.34  0.54  0.65  0.74  0.87  2774    1
lp__ -7.73    0.02 0.78 -9.90 -7.89 -7.43 -7.26 -7.21  1242    1

Samples were drawn using NUTS(diag_e) at Tue Sep 06 18:23:15 2016.
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

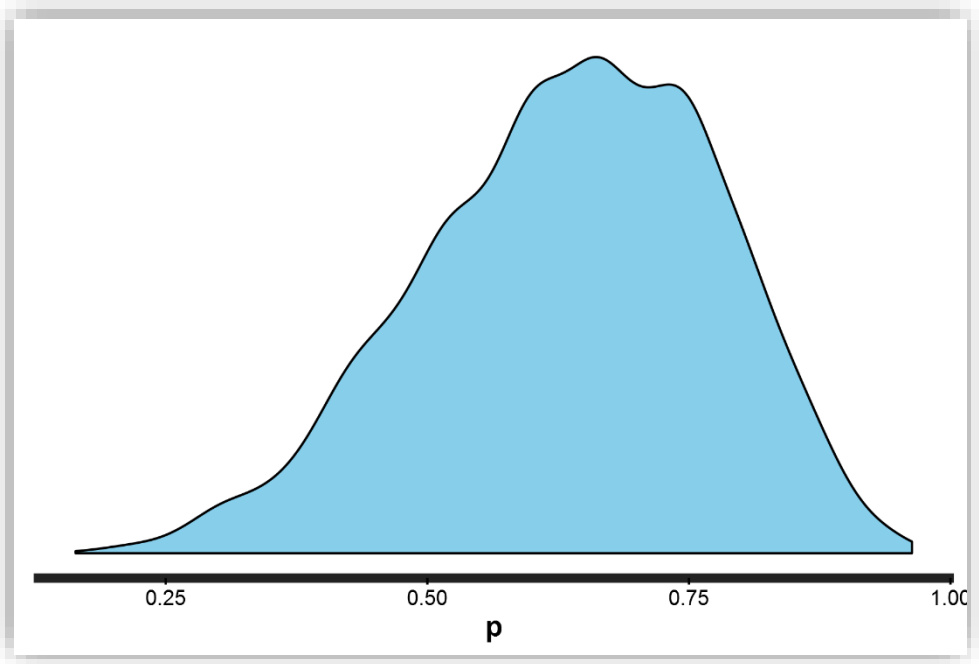
cognitive model
statistics
computing

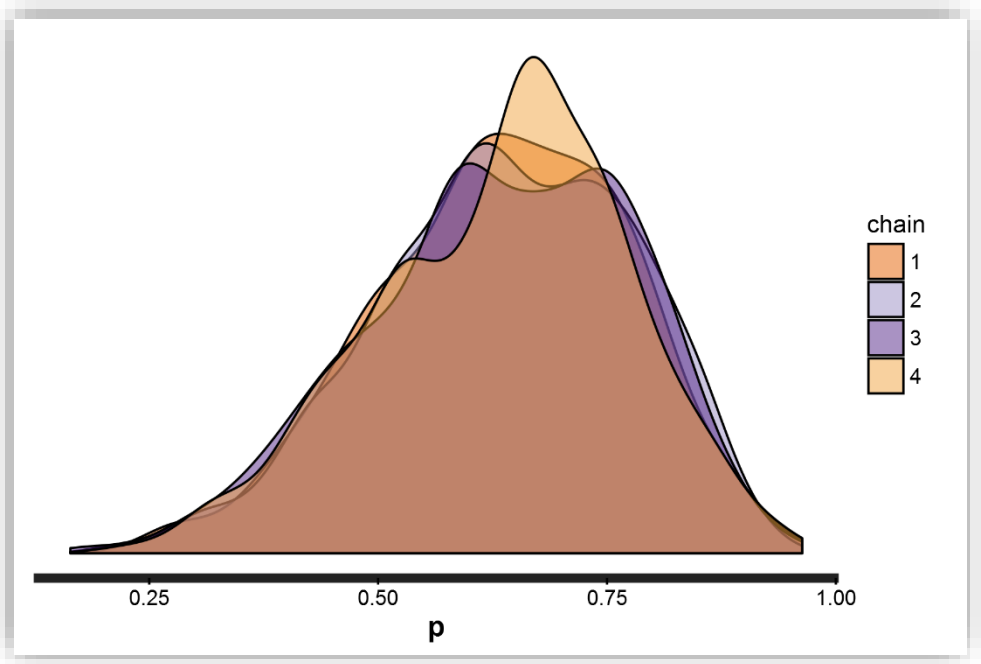




Diagnostics - density

cognitive model
statistics
computing

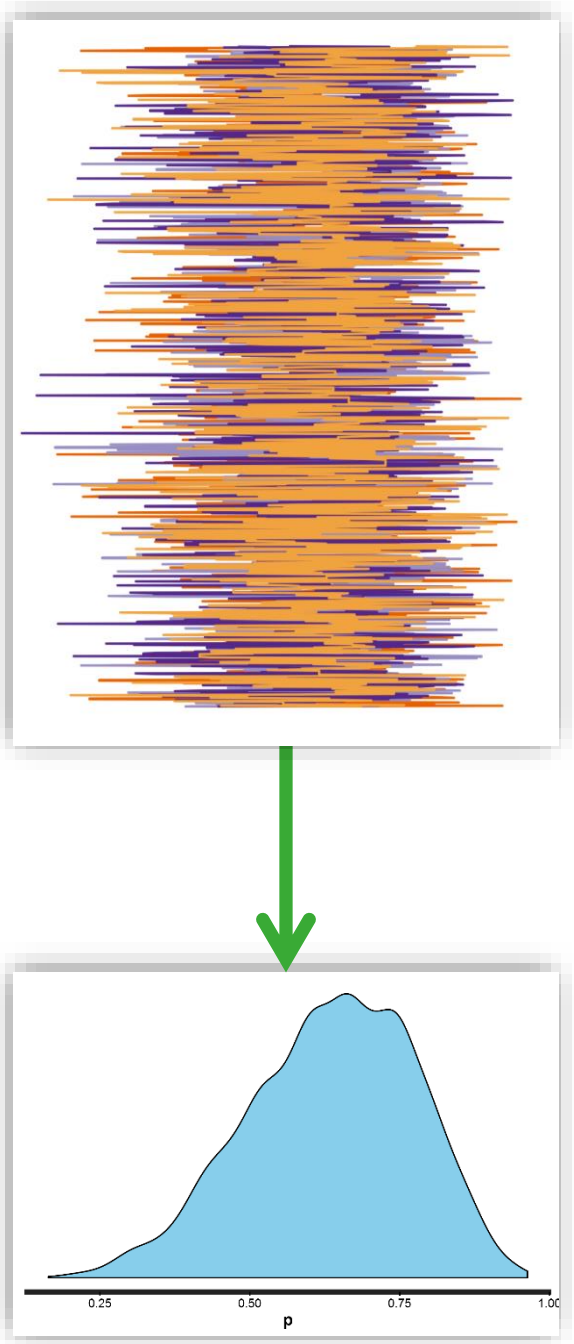




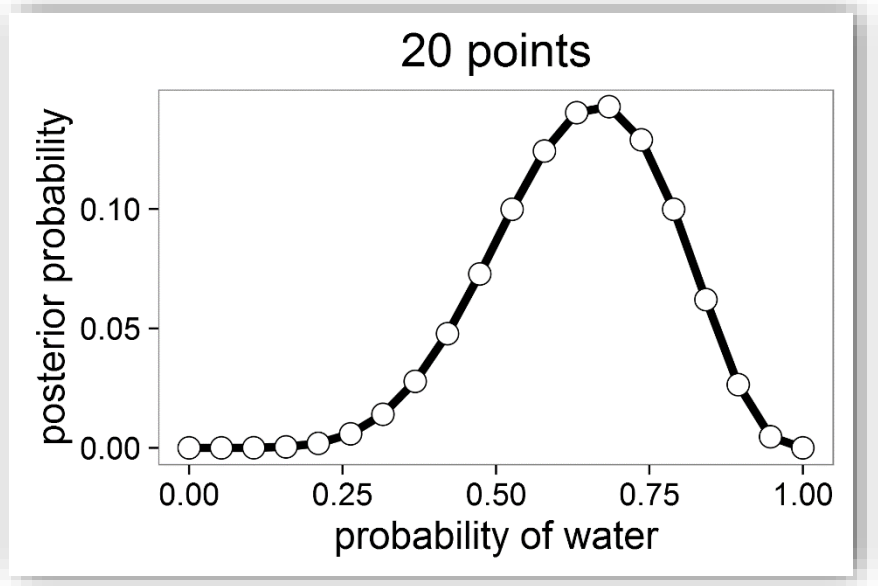
Diagnostics

cognitive model
statistics
computing

MCMC



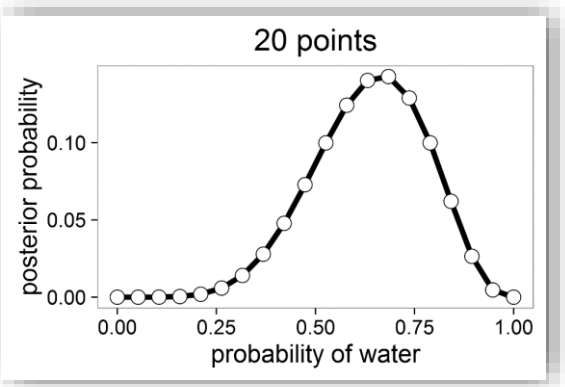
Grid Approximation

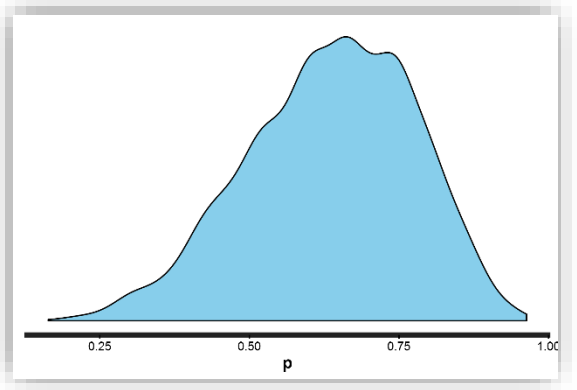


Draw a Conclusion?

cognitive model
statistics
computing

- 6W / 9 trials
- 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);
}
```


ANY
QUESTIONS
?

Stay tuned and
bis morgen!