



Bayesian Statistics and Bayesian Cognitive Modeling

Lei Zhang

Institute of Systems Neuroscience, University Medical Center Hamburg-Eppendorf

8-10. Oct. 2018, Hamburg
lei.zhang@uke.de
lei-zhang.net
 [@lei_stone](https://twitter.com/lei_stone)



Universitätsklinikum
Hamburg-Eppendorf

Schedule

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, Posterior 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	Introduction and 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? (and / or Matlab?)
- Cognitive Modeling?

You would like to...

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

About me

- Current position: Postdoc @ Gläscher Lab, ISN UKE
- Ph.D. Cognitive/computational neuroscience
- M.Sc. Cognitive neuroscience
- B.Sc. Psychology
- My journey through computational modeling
 - Started with MLE (@fminsearch in Matlab)
 - Switched to Bayesian: first JAGS, then Stan.

Overview

This workshop is **NOT** about...

- ... Bayes in the brain (e.g. predictive coding)
- ... Cognitive process that are Bayesian themselves
- ... Bayesian statistics to supersede classic statistics



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

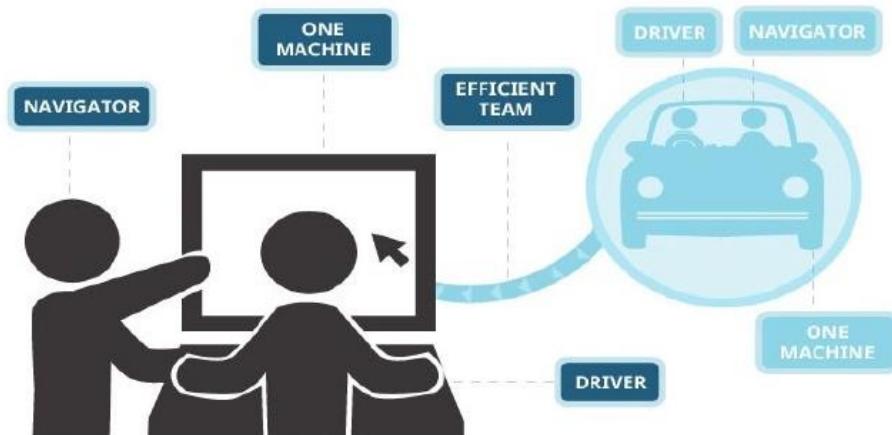
- Construct cognitive models
- Estimate posterior distributions of parameters
- Compare models: which is the best one, given the data
- Perform model-based analysis, e.g. model-based fMRI/EEG

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$
A photograph of a blackboard with the Bayes' theorem formula written in blue marker. The formula is $P(A|B) = \frac{P(B|A) P(A)}{P(B)}$. The board also shows some other faint writing and lines.

How to Get the **Most** out of the Workshop

- Work in pairs: Talk to each other & help each other
- Ask questions
- Try the exercises

PAIR PROGRAMMING



BASICS OF R PROGRAMMING



R Basics

cognitive model
statistics
computing

- R
 - a programming language for statistical computing
 - R has its own user interface
 - freely available on Windows, Mac, and Linux



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



R advanced statistics - Falk Eippert, Leipzig

22.10. - 24.10.2018

22.10.: 9:00-15:30

23.10.: 9:00-17:00

24.10.: 9:00-15:00

script editor

```
21 # -----
22 library(ggplot2)
23
24 myconfig <- theme_bw(base_size = 20) +
25   theme(panel.grid.major = element_blank(),
26         panel.grid.minor = element_blank(),
27         panel.background = element_blank() )
28
29 ## normal distribution
30 # dnorm
31 g1 <- ggplot(data.frame(x = c(-5, 5)), aes(x)) +
32   stat_function(fun = dnorm, args = list(mean = 0, sd = 1), size = 3, colour = 'black')
33 g1 <- g1 + myconfig
34 print(g1)
35
36 # pnorm
37 g2 <- ggplot(data.frame(x = c(-5, 5)), aes(x)) +
38   stat_function(fun = pnorm, args = list(mean = 0, sd = 1), size = 3)
39 g2 <- g2 + myconfig
40 print(g2)
41
42 # qnorm
43 g3 <- ggplot(data.frame(x = c(0, 1)), aes(x)) +
44   stat_function(fun = qnorm, args = list(mean = 0, sd = 1), size = 3)
45 g3 <- g3 + myconfig
46 print(g3)
```

console

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

environment/
command history

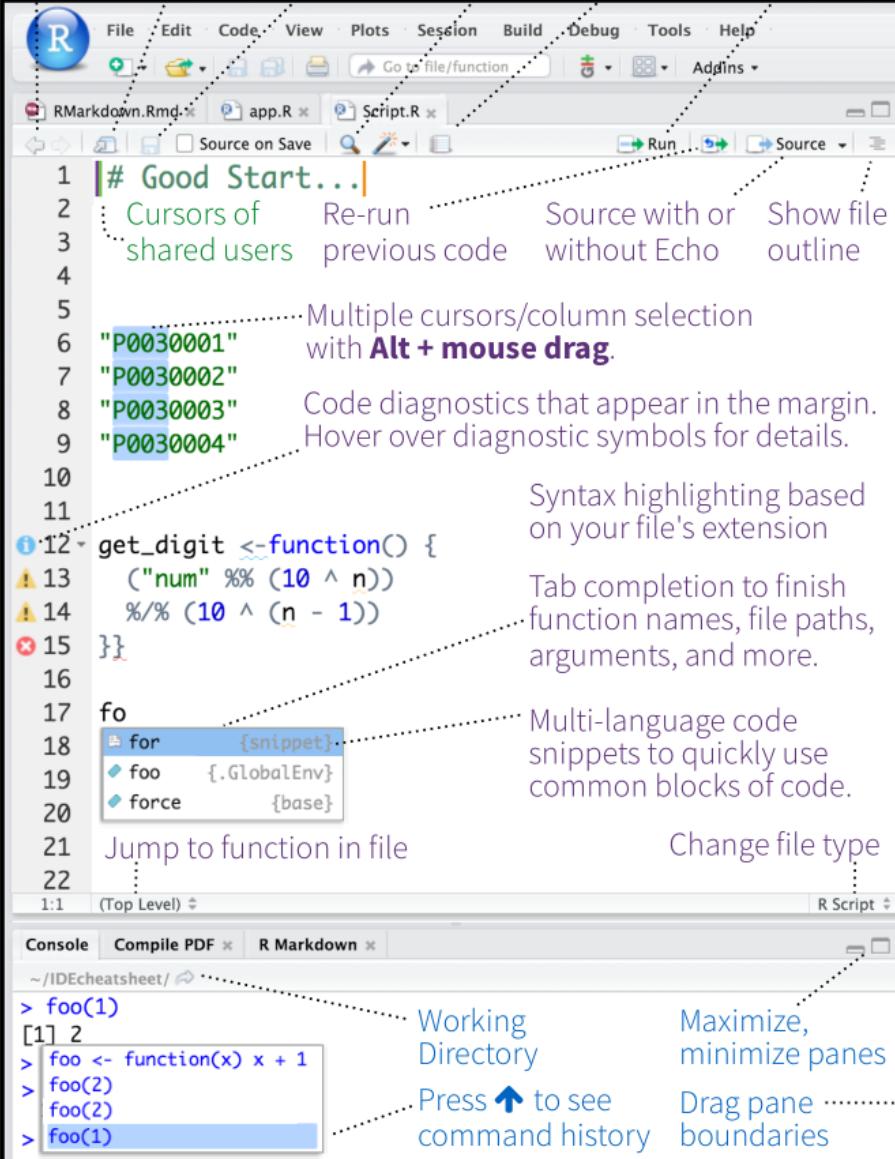
Environment is empty

file/pkg/img/
etc.

Name	Description	Version
System Library		
abind	Combine Multidimensional Arrays	1.4-3
assertthat	Easy pre and post assertions.	0.1
base64enc	Tools for base64 encoding	0.1-3
BayesFactor	Computation of Bayes Factors for Common Designs	0.912-2
BH	Boost C++ Header Files	1.60.0-1
bitops	Bitwise Operations	1.0-6
boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-17
broom	Convert Statistical Analysis Objects into Tidy Data Frames	0.4.1
Cairo	R graphics device using cairo graphics library for creating high-quality bitmap (PNG, JPEG, TIFF), vector (PDF, SVG, PostScript) and display (X11 and Win32) output	1.5-9
car	Companion to Applied Regression	2.1-1
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
class	Functions for Classification	7.3-14
cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.3
coda	Output Analysis and Diagnostics for MCMC	0.18-1
codetools	Code Analysis Tools for R	0.2-14
colorspace	Color Space Manipulation	1.2-6
compiler	The R Compiler Package	3.2.3
complot	Visualization of a correlation matrix	0.73
cubature	Adaptive multivariate integration over hypercubes	1.1-2
curl	A Modern and Flexible Web Client for R	0.9.6
DAAG	Data Analytic and Graphic Data and Functions	1.22

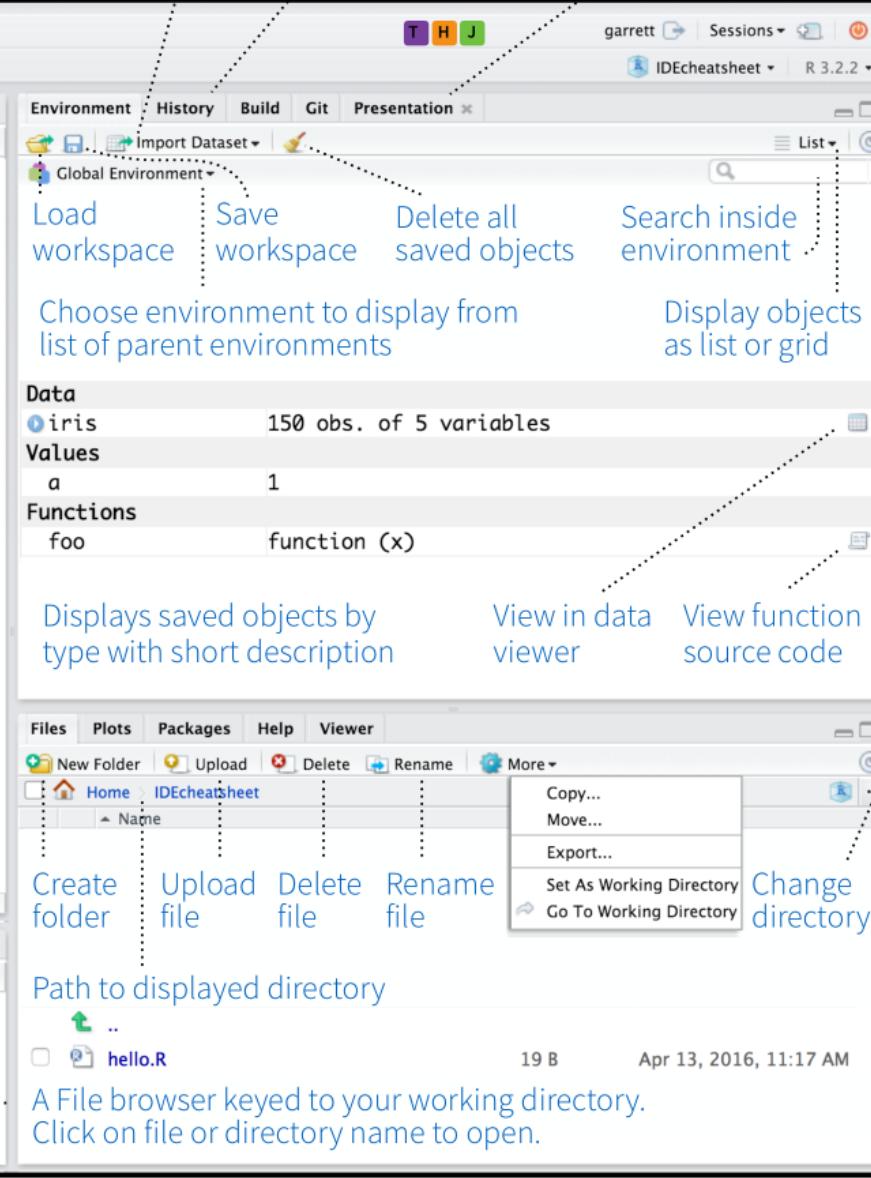
Write Code

Navigate tabs Open in new window Save Find and replace Compile as notebook Run selected code



R Support

Import data with wizard History of past commands to run/copy Display .RPres slideshows
File > New File > R Presentation



cognitive model
statistics
computing

<https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf>

R Console as a Calculator

Addition and Subtraction

```
> 3+2  
[1] 5
```

```
> 3-2  
[1] 1
```

Multiplication and Division

```
> 3*2  
[1] 6
```

```
> 3/2  
[1] 1.5
```

Exponents in R

```
> 3^2  
[1] 9
```

```
> 2^3  
[1] 8
```

Constants in R

```
> pi  
[1] 3.141593
```

```
> exp(1) base of the natural logarithm  
[1] 2.718282
```

Special values

Infinite Values

```
> Inf  
[1] Inf
```

```
> 1+Inf  
[1] Inf
```

Machine Epsilon

```
> .Machine$double.eps  
[1] 2.220446e-16
```

```
> 0>.Machine$double.eps  
[1] FALSE
```

Empty Values

```
> NULL  
NULL
```

```
> 1+NULL  
numeric(0)
```

Missing Values

```
> NA  
[1] NA
```

```
> 1+NA  
[1] NA
```

Storing and manipulating variables

cognitive model
statistics
computing

Define objects `x` and `y` with values of 3 and 2, respectively:

```
> x=3  
> y=2
```

Some calculations with the defined objects `x` and `y`:

```
> x+y  
[1] 5
```

```
> x*y  
[1] 6
```

Warning: R is case sensitive, so `x` and `X` are not the same object.

Basic R functions

Combine

```
> c(1,3,-2)  
[1] 1 3 -2
```

```
> c("a","a","b","b","a")  
[1] "a" "a" "b" "b" "a"
```

Sum and Mean

```
> sum(c(1,3,-2))  
[1] 2
```

```
> mean(c(1,3,-2))  
[1] 0.6666667
```

Variance and Std. Dev.

```
> var(c(1,3,-2))  
[1] 6.333333
```

```
> sd(c(1,3,-2))  
[1] 2.516611
```

Minimum and Maximum

```
> min(c(1,3,-2))  
[1] -2
```

```
> max(c(1,3,-2))  
[1] 3
```

Basic R functions (cont.)

Define objects `x` and `y`:

```
> x=c(1,3,4,6,8)  
> y=c(2,3,5,7,9)
```

Calculate the correlation:

```
> cor(x,y)  
[1] 0.988765
```

Calculate the covariance:

```
> cov(x,y)  
[1] 7.65
```

Combine as columns

```
> cbind(x,y)  
      x  y  
[1, ] 1  2  
[2, ] 3  3  
[3, ] 4  5  
[4, ] 6  7  
[5, ] 8  9
```

Combine as rows

```
> rbind(x,y)  
      [,1] [,2] [,3] [,4] [,5]  
x     1     3     4     6     8  
y     2     3     5     7     9
```

Basic Commands

```
getwd()
setwd('E:/teaching/BayesCog/')
dir() # folders/files in the wd
ls() # anything in the environment/workspace
print('Hello World!')
cat('Hello', 'World!')
paste0('C:/', 'Group1')
help(func)
? func
a <- 5
a = 5
head(d) # first 6 entries
tail(d) # last 6 entries
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 codes (in a script)

Editor (WIN general) - Shortcuts

cognitive model
statistics
computing

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

Ctrl + end/Ende: go to the very end 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"

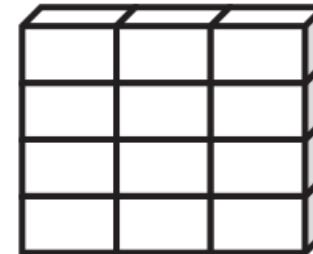
(complex: 1+2i)

Data Types

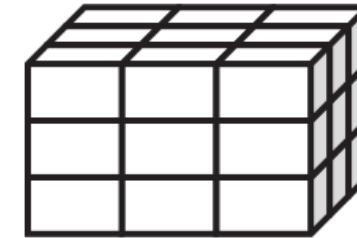
Vector



(b) Matrix



(c) Array



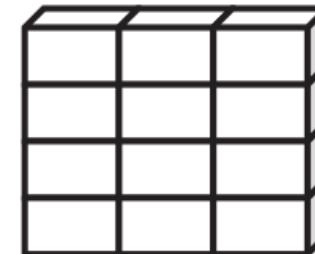
Matrix

Array

Data Frame

List

(d) Data frame



Columns can be different modes

(e) List

Vectors
Arrays
Data frames
Lists

Exercise I

cognitive model
statistics
computing

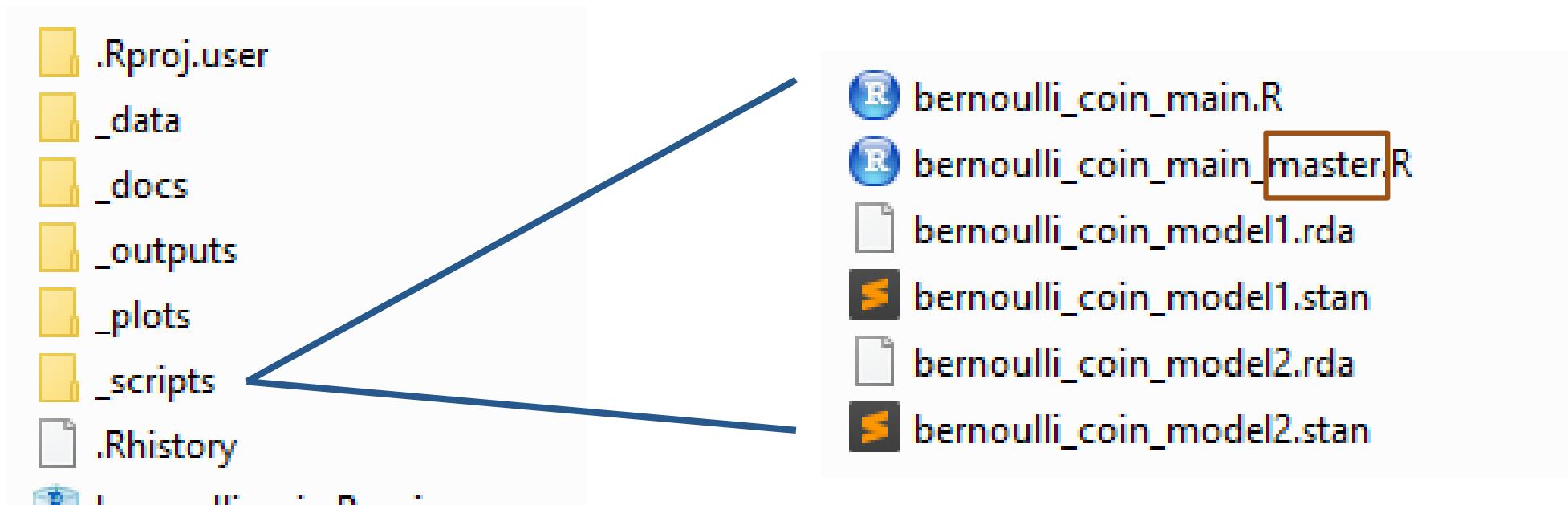
.../BayesCog/01.R_basics/_scripts/R_basics.R

up to “Control Flow”

TASK: practise basic R commands and data type

TIP: `class()`, `str()`

Side note: folder structure



click this to start each exercise,
then no need to set directory

Logical Operators

Operator	Summary
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to
!x	NOT x
x y	x OR y
x&y	x AND y

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..  
    }  
}
```

User-defined Function

```
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

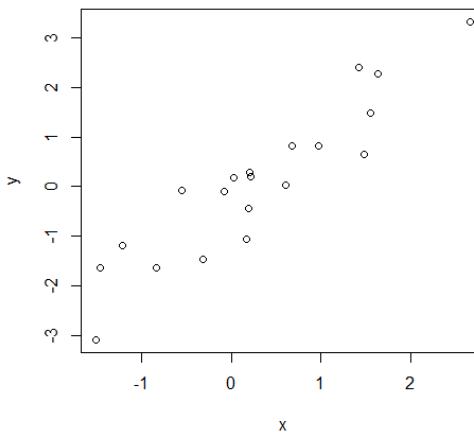
TASK: practise control flow and user-defined function

Visualization

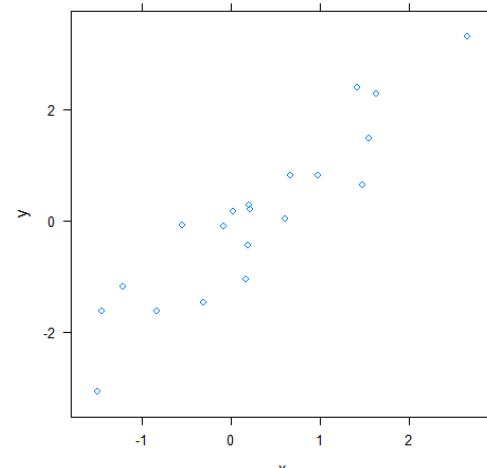
cognitive model
statistics
computing

- built-in plotting functions – first attempt / quick look / exploratory
- **{lattice}** – making nicer, similar to basic plotting functions
- **{ggplot2}** – making nicer, a layering philosophy

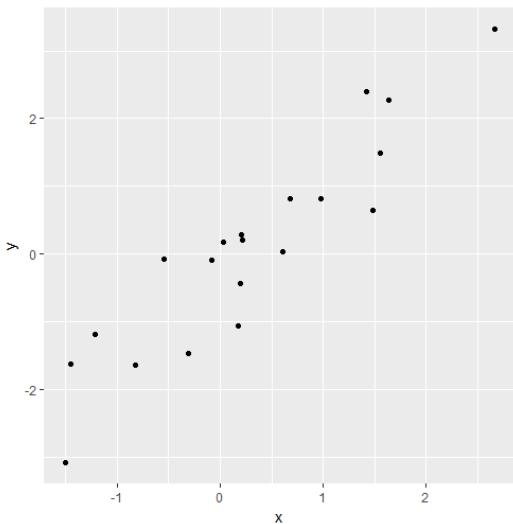
`plot(x,y)`



`lattice::xyplot(y~x)`



`ggplot2::qplot(x,y)`

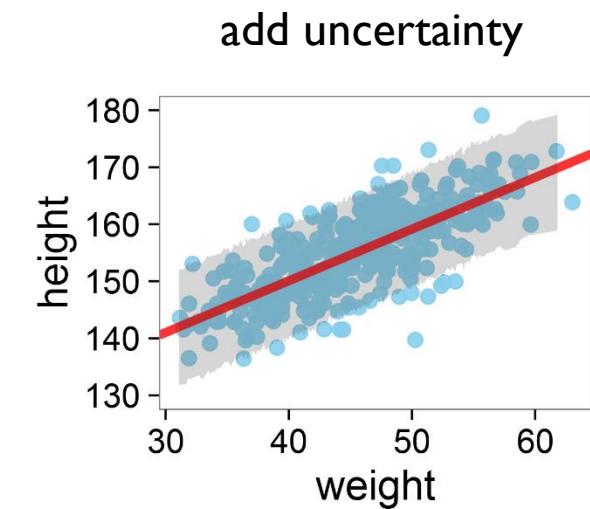
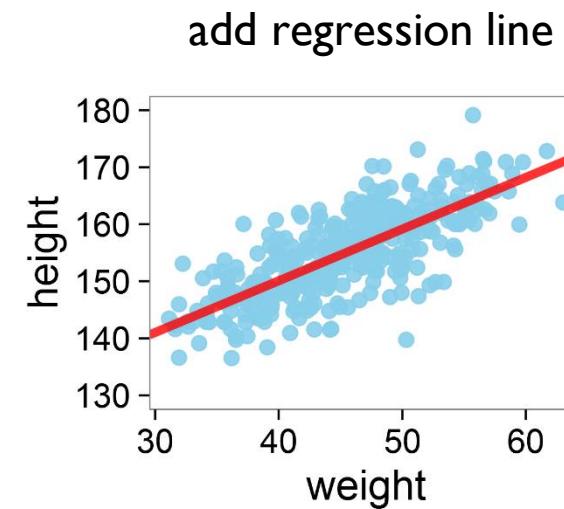
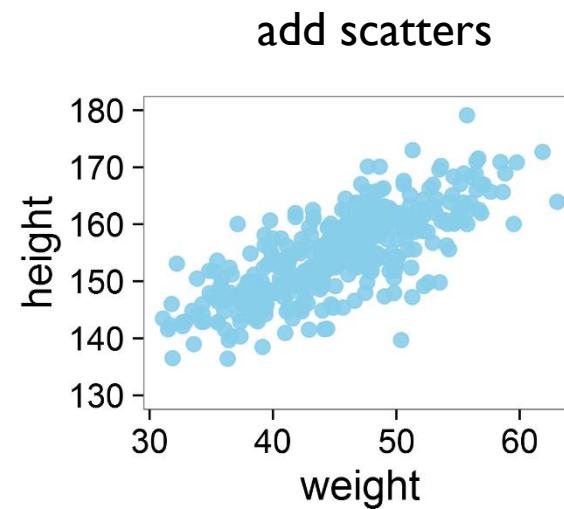
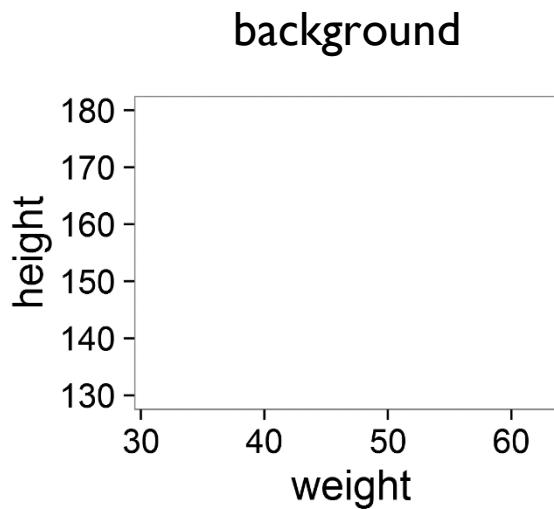


Brief Intro to ggplot2

cognitive model
statistics
computing

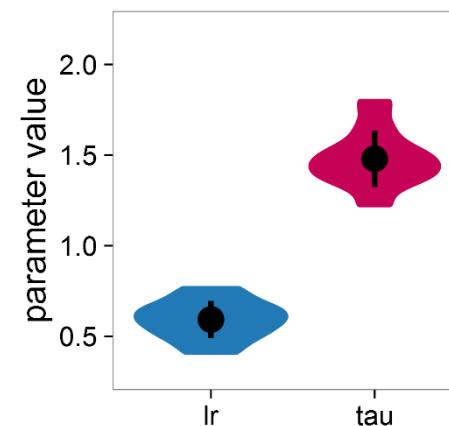
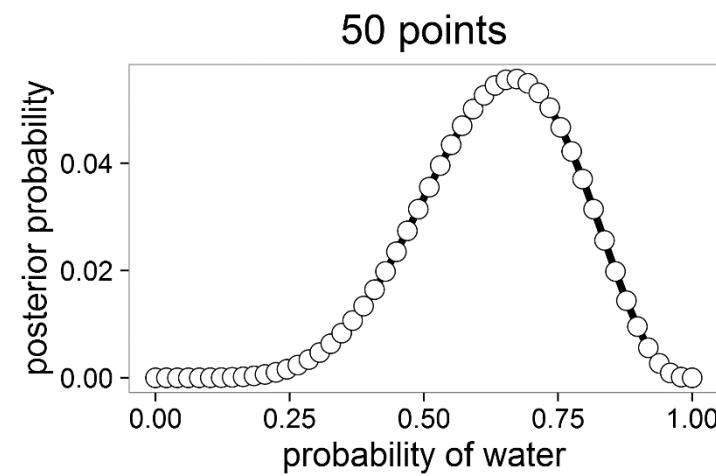
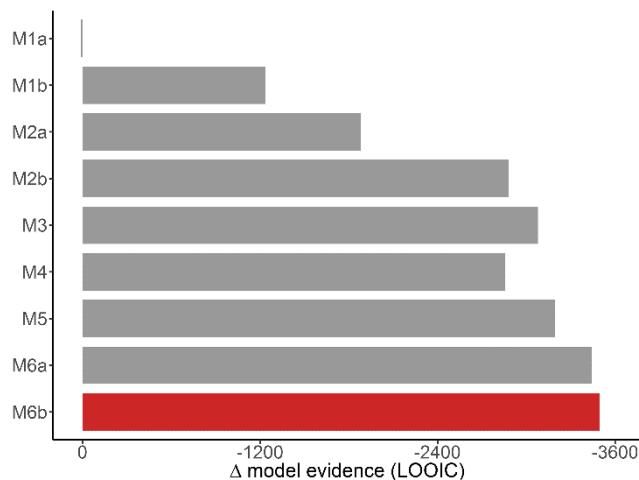
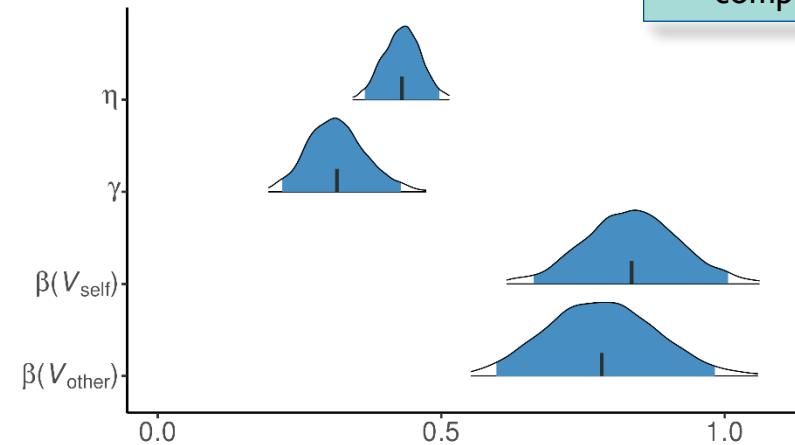
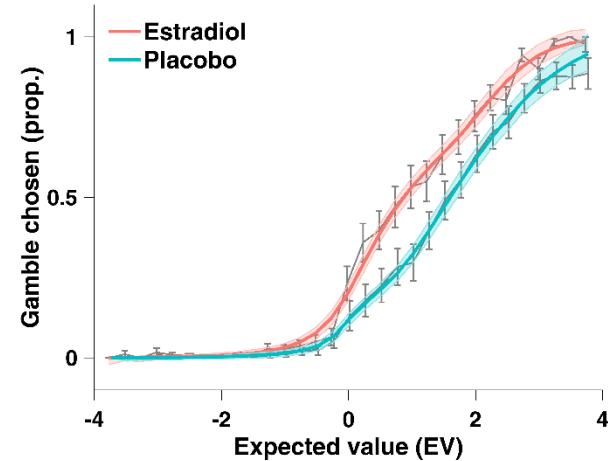
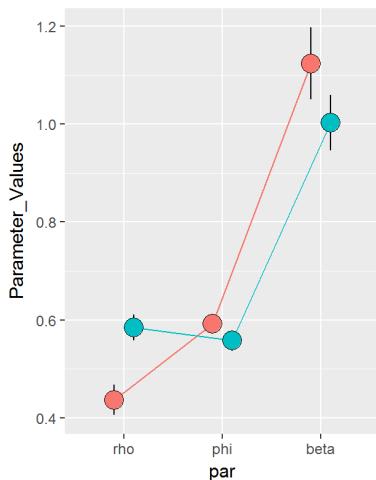
`plot = geometric (points, lines, bars) + aesthetic (color, shape, size)`

game of adding layers!



A taste of ggplot2

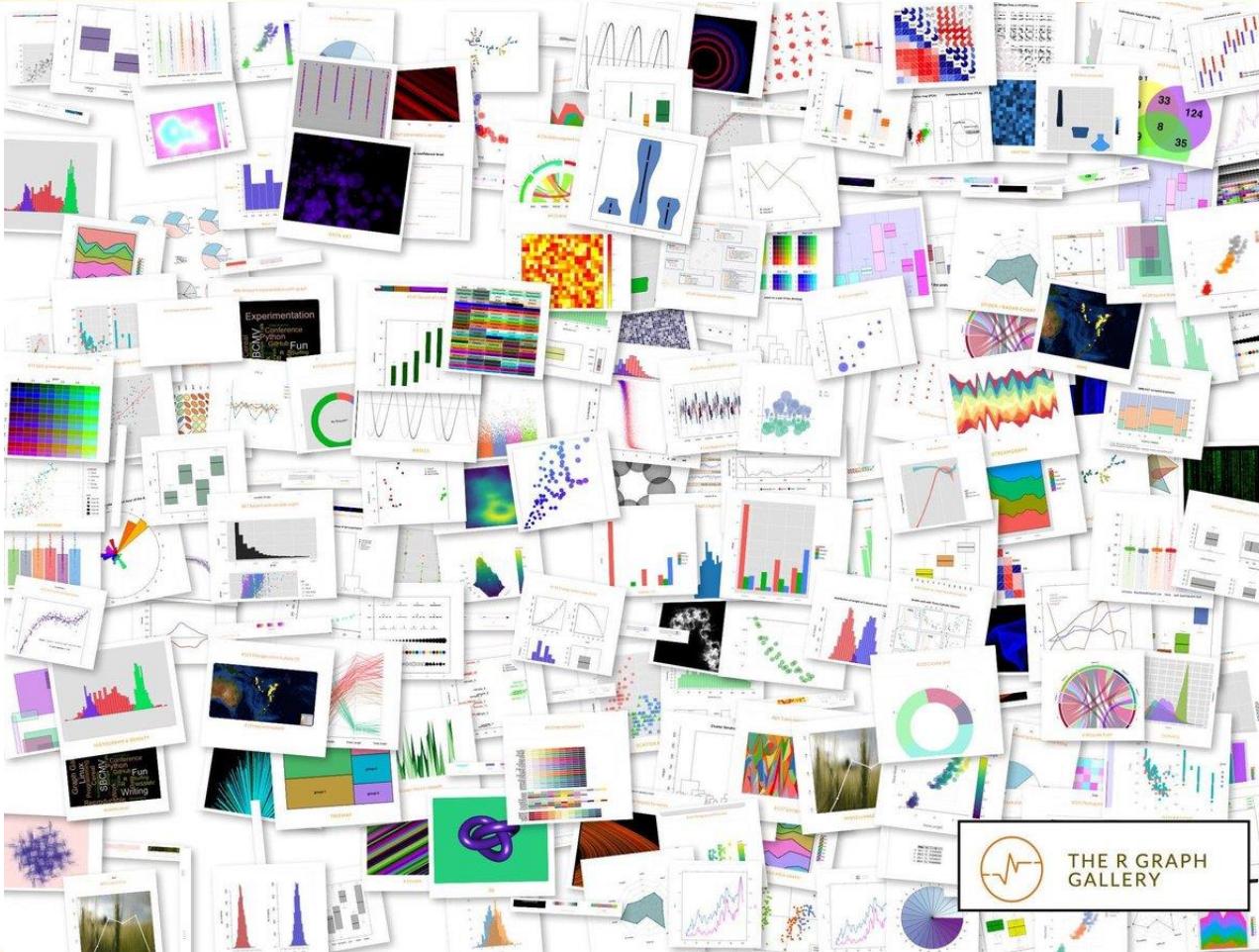
cognitive model
statistics
computing



cognitive model

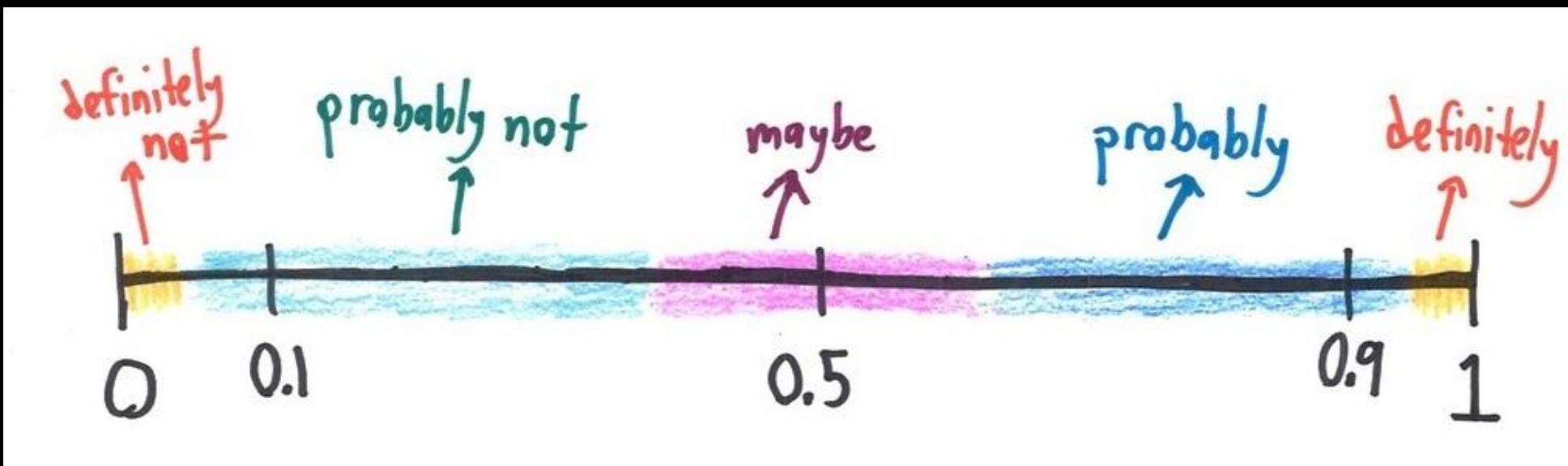
statistics

computing



<https://www.r-graph-gallery.com/>

BASICS OF PROBABILITY



to respondents' estimate of likelihood

Word or phrase

Always

Certainly

Slam dunk

Almost certainly

Almost always

With high probability

Usually

Likely

Frequently

Probably

Often

Serious possibility

More often than not

Real possibility

With moderate probability

Maybe

Possibly

Might happen

Not often

Unlikely

With low probability

Rarely

Never

0% 50% 100%

Probability

cognitive model
statistics
computing

...assigning numbers to a set of possibilities

Properties (Kolmogorov, 1956)

- $p \in [0,1]$
- $\sum p = 1$
- $p(A \cup B) = p(A) + p(B)$, 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

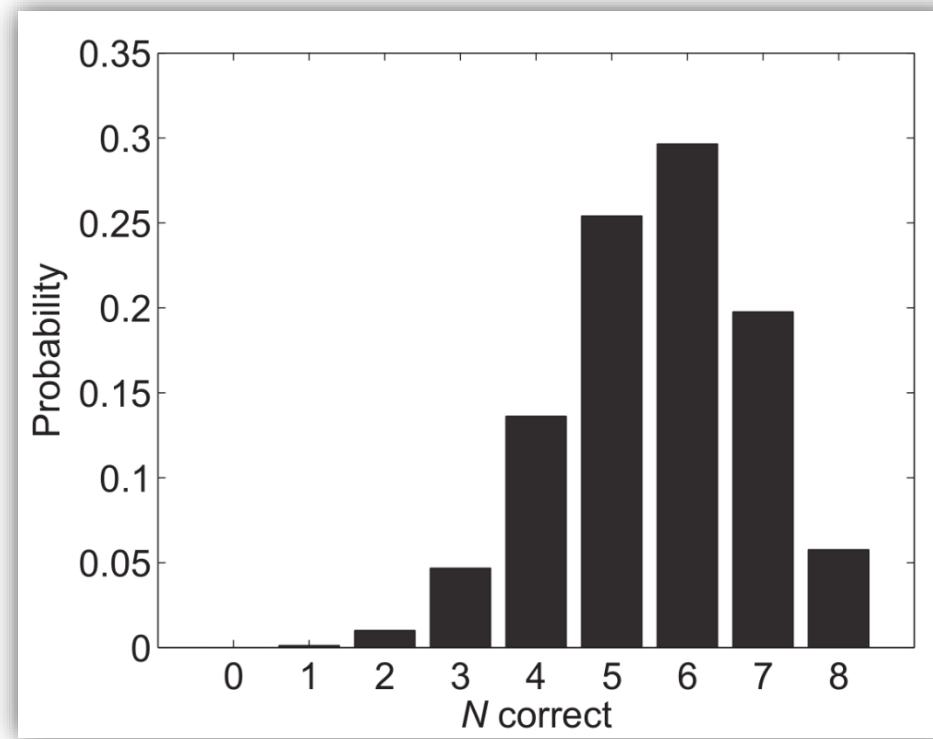
$$p(A|B) - \text{'p of A given B'}$$

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

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

Probability Functions

discrete events – we talk about mass

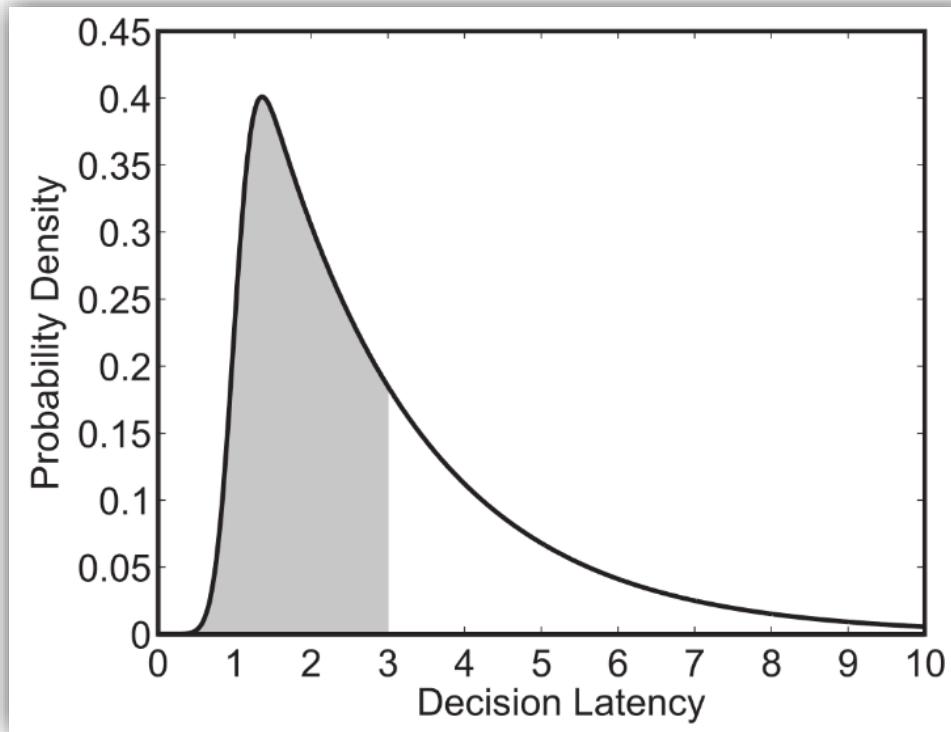


Probability Functions

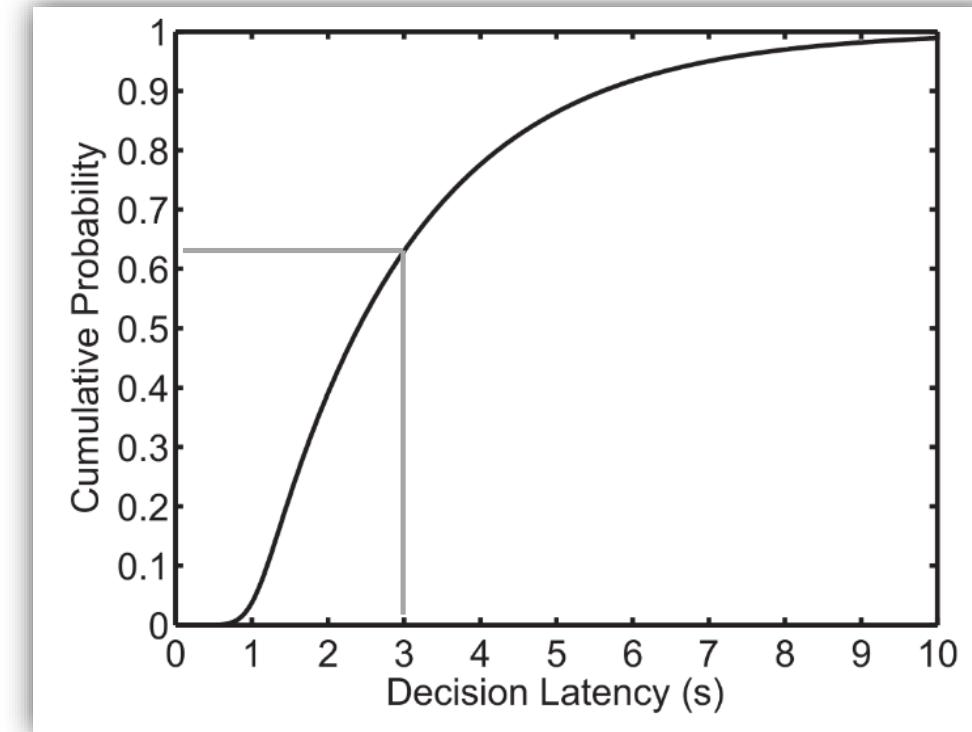
cognitive model
statistics
computing

continuous events – we talk about density

probability density function (PDF)



cumulative distribution function (CDF)



Playing with Probability Functions in R

cognitive model
statistics
computing

`dnorm()` – PDF

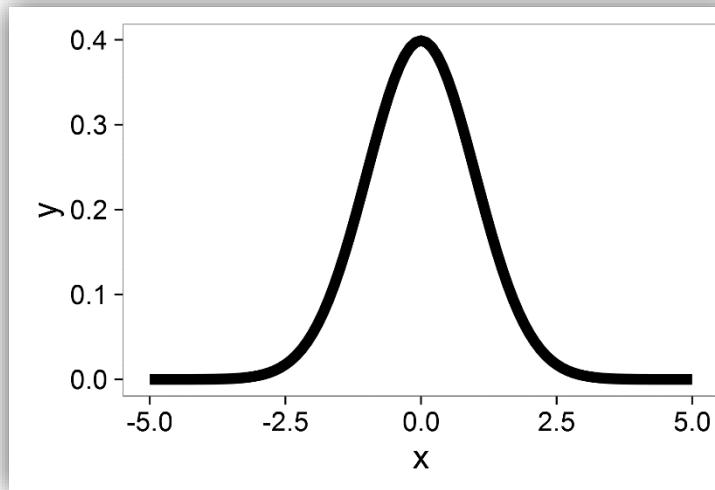
`pnorm()` – CDF

`qnorm()` – quantile, inverse cdf

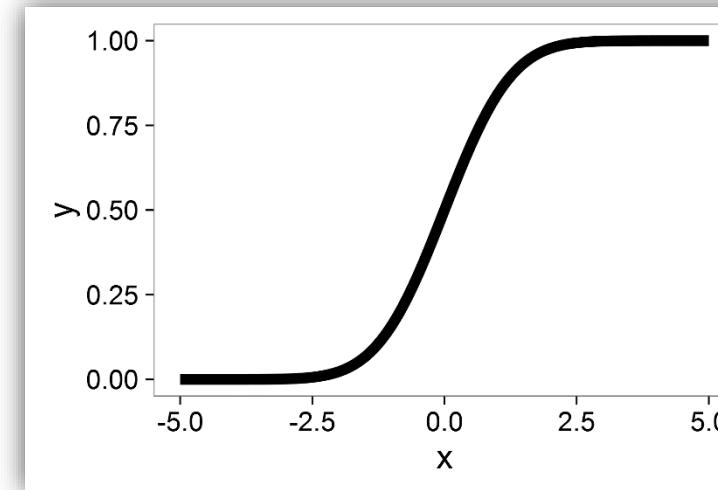
`rnorm()` – random number generator

Example: Normal(0,1)

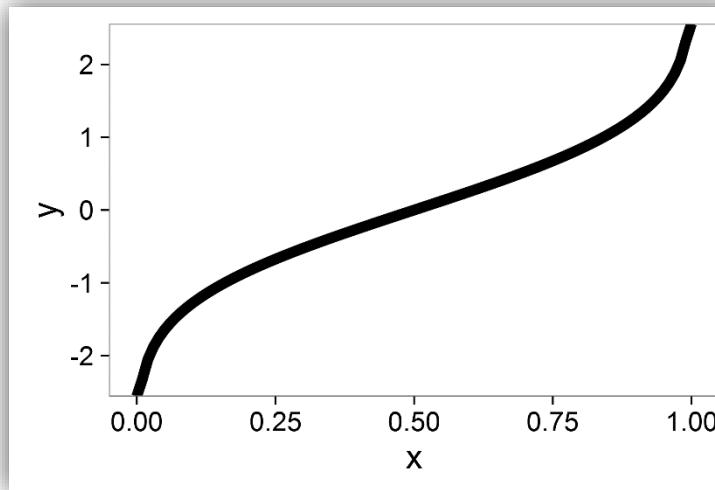
`dnorm()`



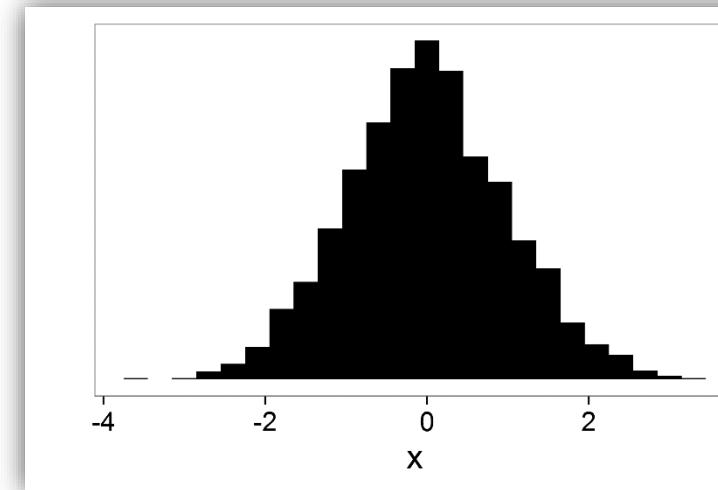
`pnorm()`



`qnorm()`



`rnorm()`

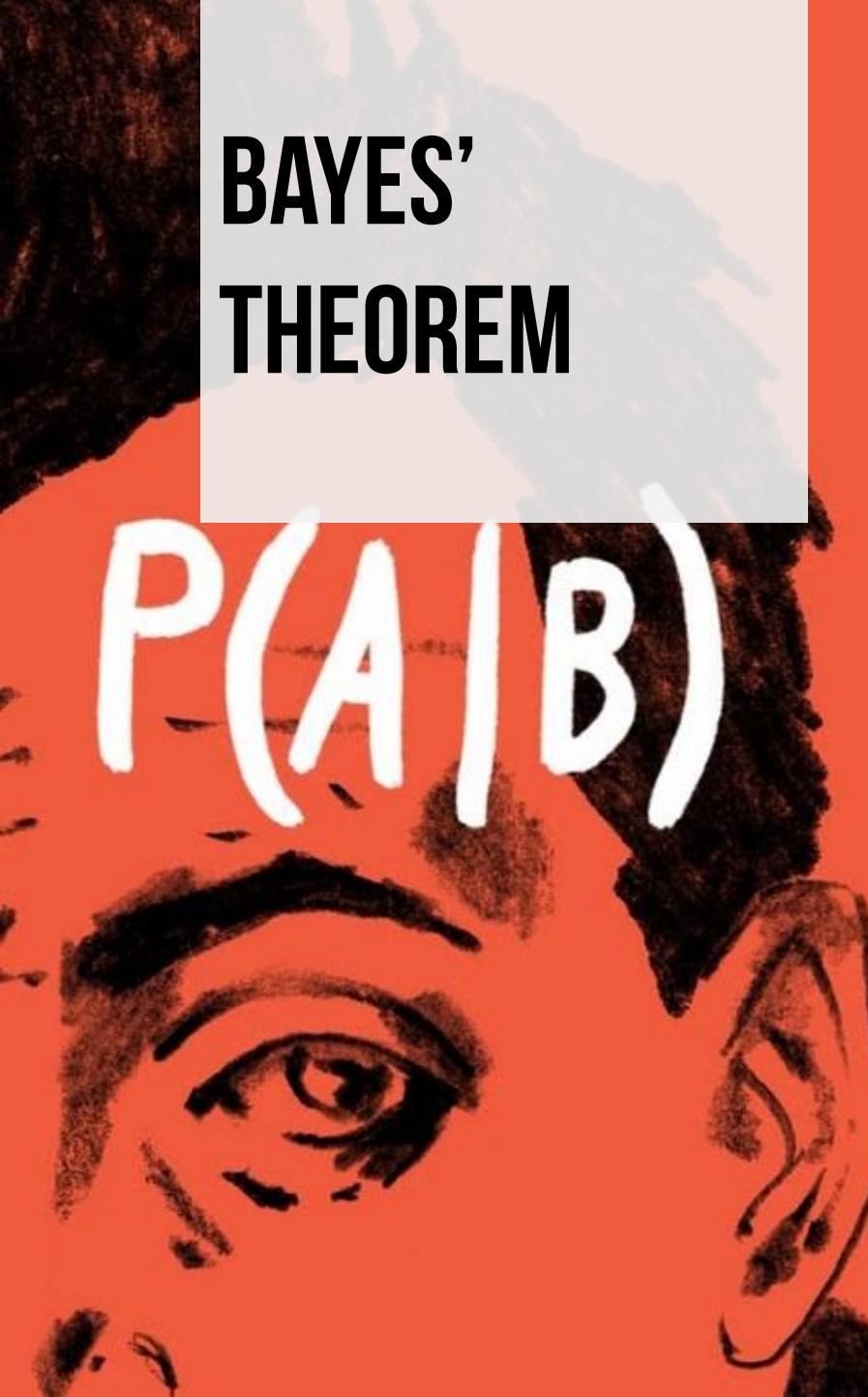


Exercise III

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

$P(A|B)$

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

Bayes' theorem

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

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

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

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

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

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

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

One Example

cognitive model
statistics
computing

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

One Example

Eye color	Hair color				Marginal (Eye color)
	Black	Brunette	Red	Blond	
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

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

Exercise IV

cognitive model
statistics
computing

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 $\vartheta = \text{😊}$ if disease is present in a person, or the value $\vartheta = \text{☺}$ if the disease is absent. The base rate of the disease is therefore denoted $p(\vartheta = \text{😊}) = 0.001$.

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

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

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

cognitive model
statistics
computing

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

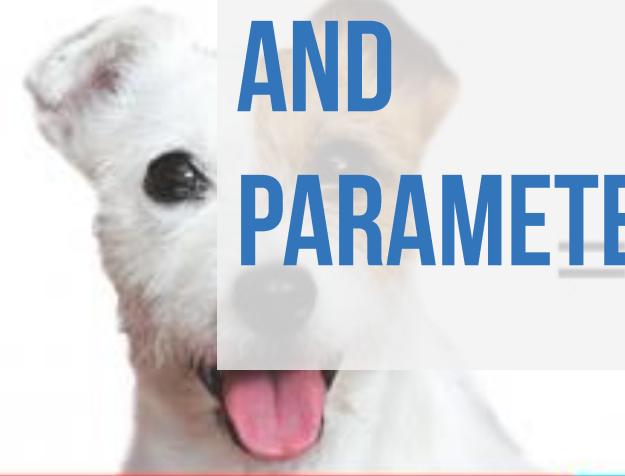
$$\rightarrow p(\vartheta = \text{患病} | T = +)$$

Exercise IV

Test result	Disease		Marginal (test result)
	$\theta = \ddot{\circ}$ (present)	$\theta = \circ$ (absent)	
$T = +$	$p(+ \ddot{\circ}) p(\ddot{\circ})$ $= 0.99 \cdot 0.001$	$p(+ \circ) p(\circ)$ $= 0.05 \cdot (1 - 0.001)$	$\sum_{\theta} p(+ \theta) p(\theta)$
$T = -$	$p(- \ddot{\circ}) p(\ddot{\circ})$ $= (1 - 0.99) \cdot 0.001$	$p(- \circ) p(\circ)$ $= (1 - 0.05) \cdot (1 - 0.001)$	$\sum_{\theta} p(- \theta) p(\theta)$
Marginal (disease)	$p(\ddot{\circ}) = 0.001$	$p(\circ) = 1 - 0.001$	1.0

$$\begin{aligned}
 p(\theta = \ddot{\circ} | T = +) &= \frac{p(T = + | \theta = \ddot{\circ}) p(\theta = \ddot{\circ})}{\sum_{\theta} p(T = + | \theta) p(\theta)} \\
 &= \frac{0.99 \cdot 0.001}{0.99 \cdot 0.001 + 0.05 \cdot (1 - 0.001)} \\
 &= 0.019
 \end{aligned}$$

LINKING DATA AND PARAMETER



$p(\theta | D)$

$p(D | \theta)$

$p(\theta)$

$p(D)$

Linking Data and Parameter

cognitive model
statistics
computing

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

A diagram illustrating the components of the Bayes' rule formula. On the left, there is a term $p(A|B)$. Two blue arrows point towards it: one from the symbol θ above and another from the symbol D to its right.

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 $p(\text{Data})$?

cognitive model
statistics
computing

discrete parameters

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

continuous parameters

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

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
- $\rightarrow 6/9 = 0.666667?$
- Is it right? If not, what to do next?



cognitive model
statistics
computing

Steps of Bayesian Modeling?

A data story

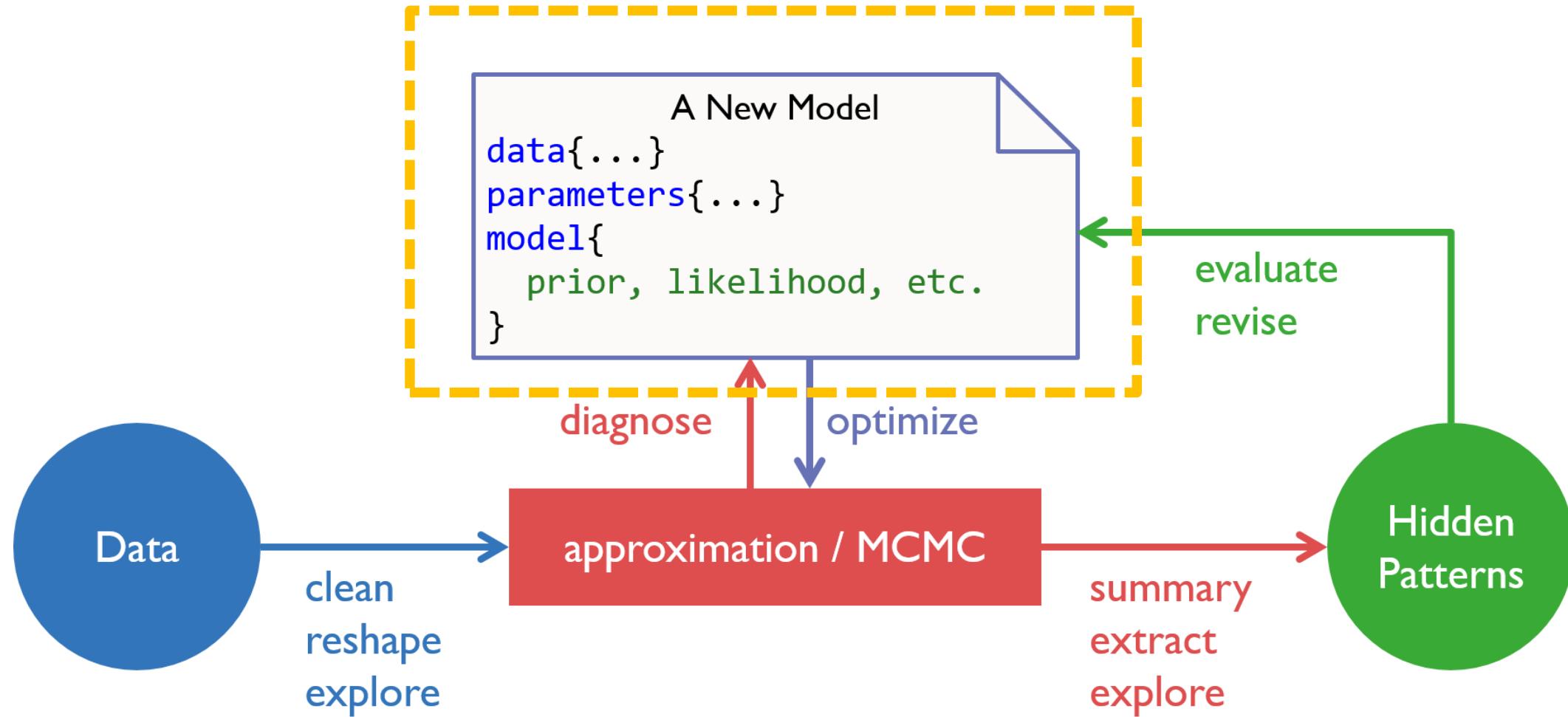
Think about how the data might arise.
It can be *descriptive* or even *causal*.

Update

Educate your model by feeding it with data.
Bayesian Update:
update the prior, in light of data, to produce posterior
the updated posterior then becomes the prior of next update

Evaluate

Compare model with reality.
Revise your model.



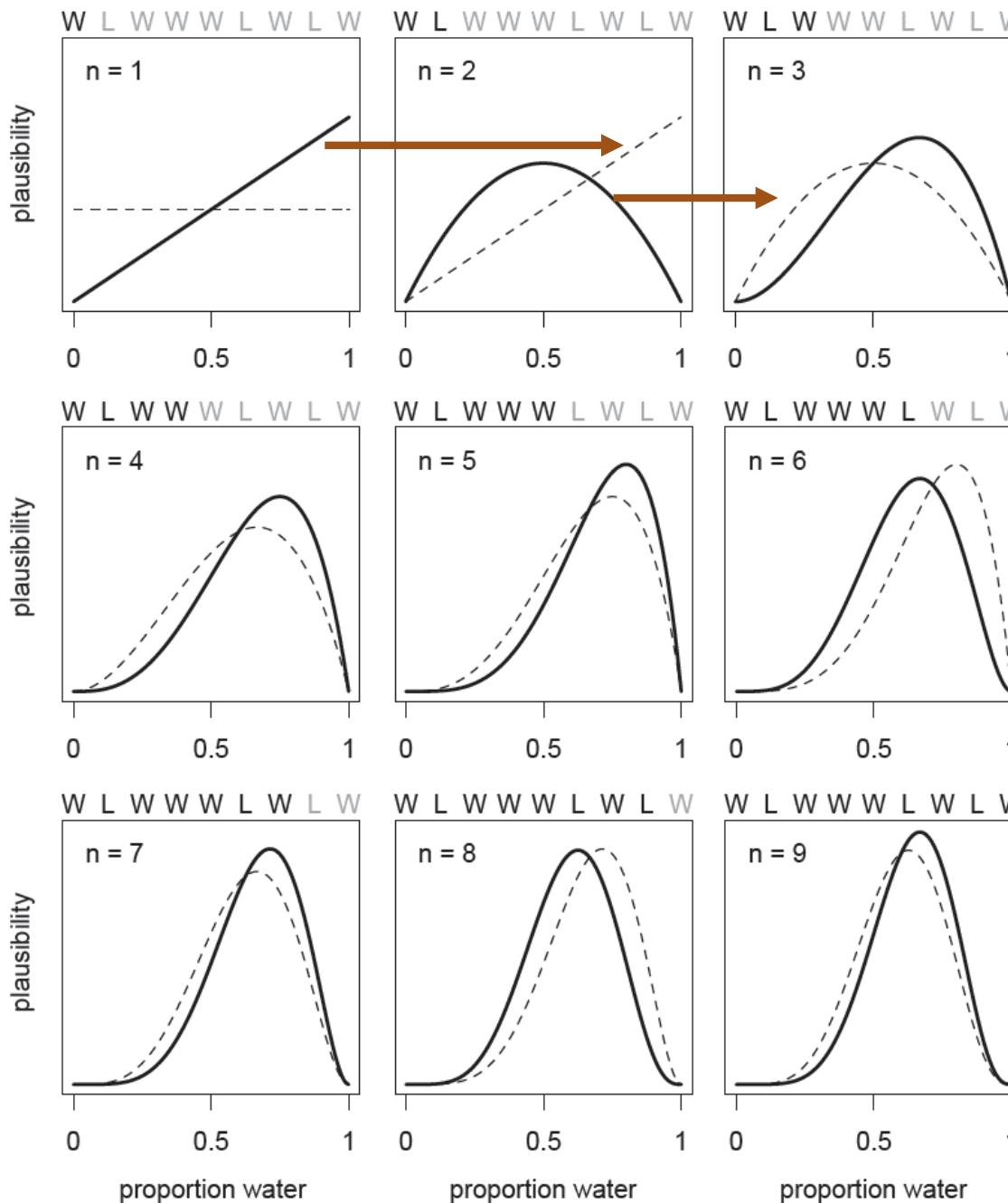
A Data Story of the Globe

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.



Update



- order doesn't matter
- 2/3 is most likely
- others are not ruled out

Components of a Model

think about the likelihood function (of Binomial):

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

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

N : total number of observations

w : number of water

p : proportion of water



known (data)

unknown (parameter)

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)

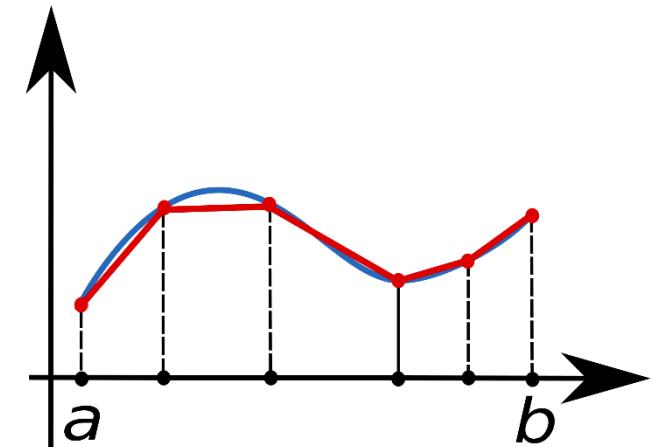
# compute likelihood at each value in grid
likelihood <- dbinom(w , size = N , prob = p_grid )

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

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

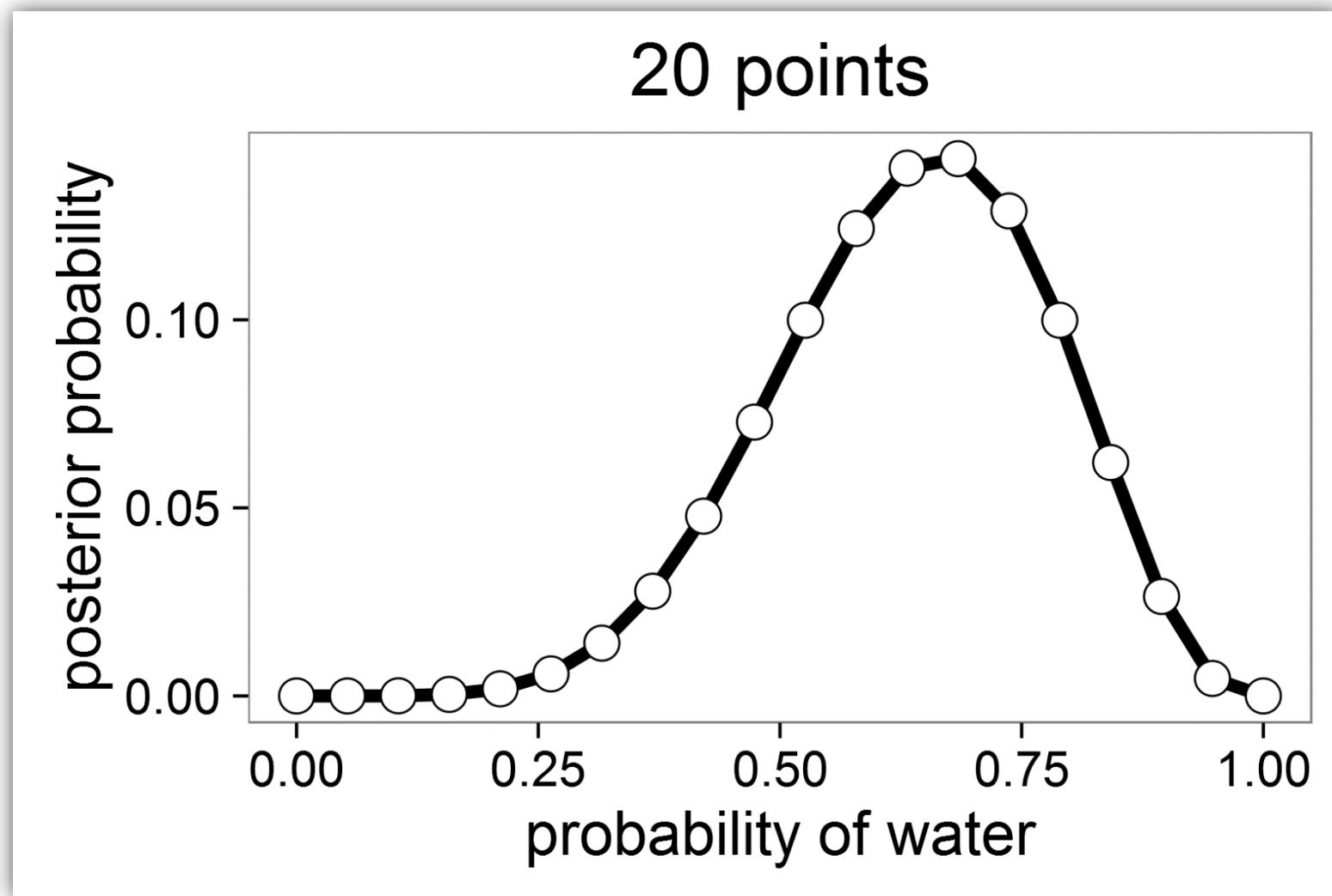
$$p(\theta | D) = \frac{p(D | \theta)p(\theta)}{\int p(D | \theta^*)p(\theta^*)d\theta^*}$$

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



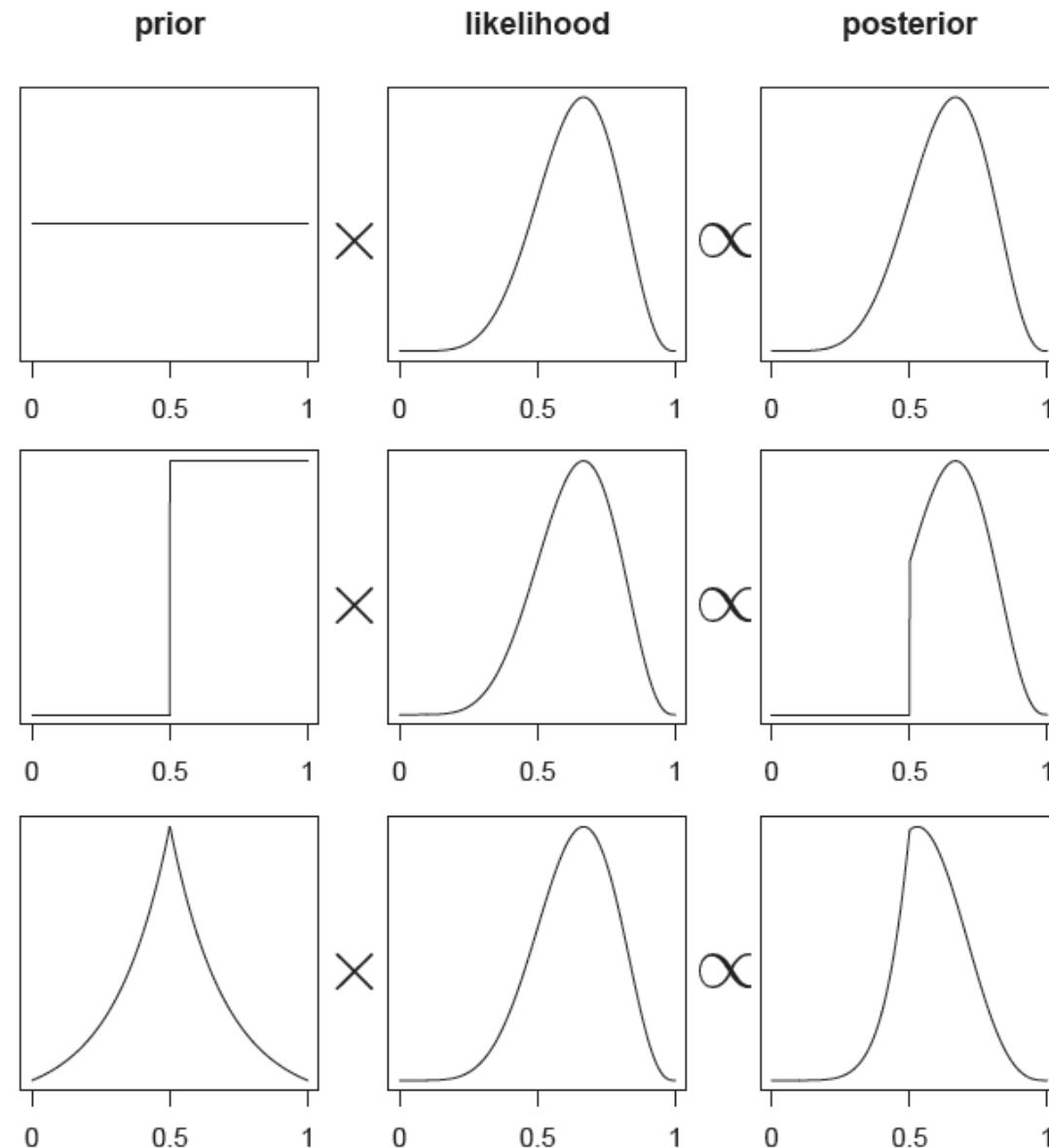
Binomial Model – Grid Approximation

cognitive model
statistics
computing



Impact of Prior

cognitive model
statistics
computing



Exercise V

cognitive model
statistics
computing

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

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

Components of a Model

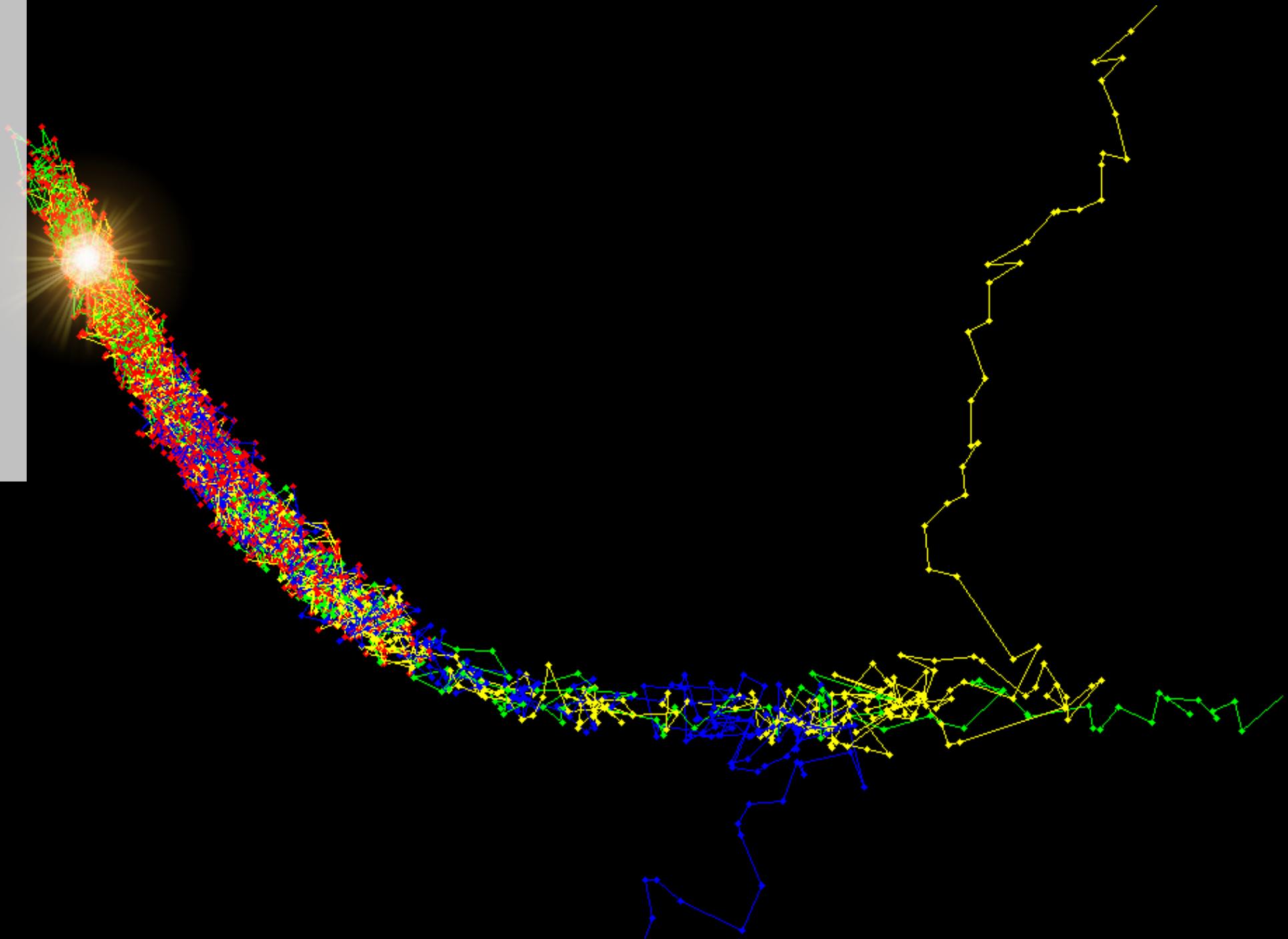
grid approximation for
2 parameters?
5 parameters?
10 parameters?

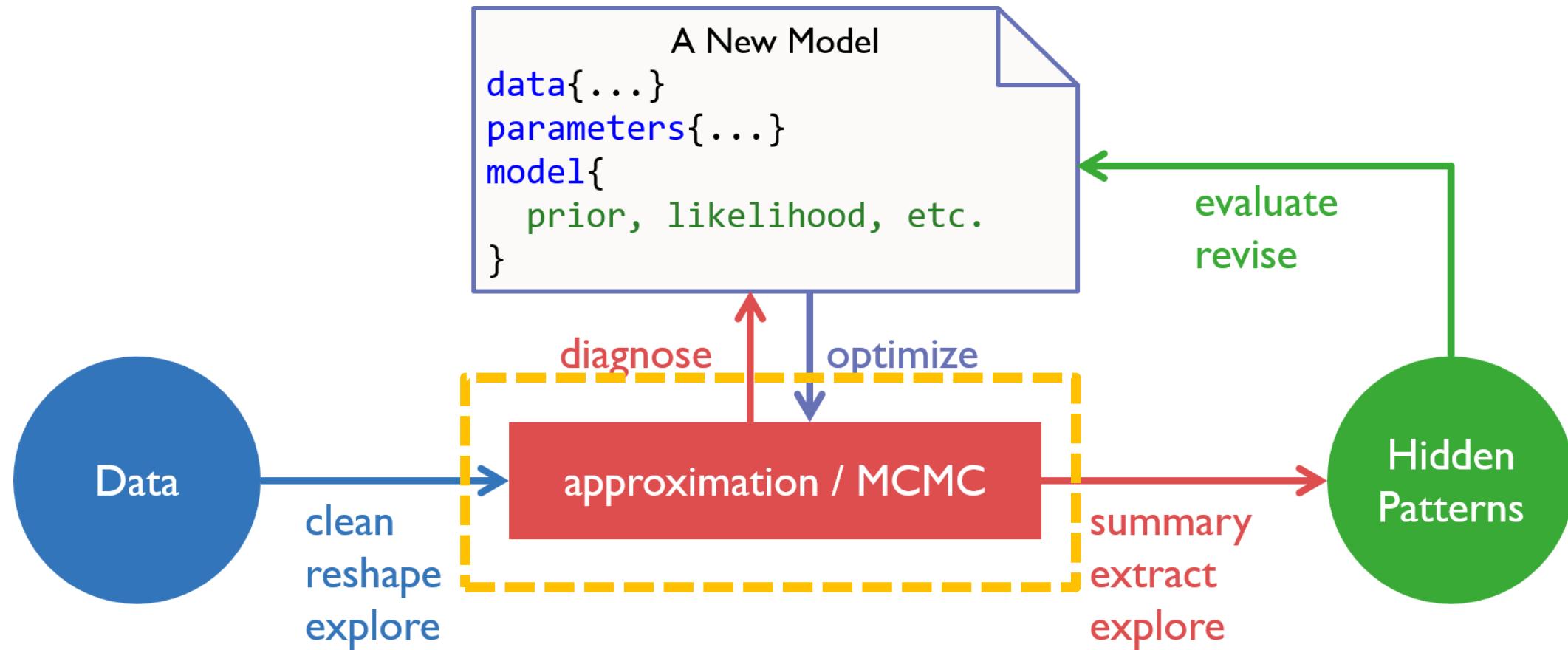
$$p(\theta | D) = \frac{p(D | \theta)p(\theta)}{\int p(D | \theta^*)p(\theta^*) d\theta^*}$$

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

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

MARKOV CHAIN MONTE CARLO





Solving the Problem by Approximation

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

Deterministic
Approximation

→ Variational Bayes

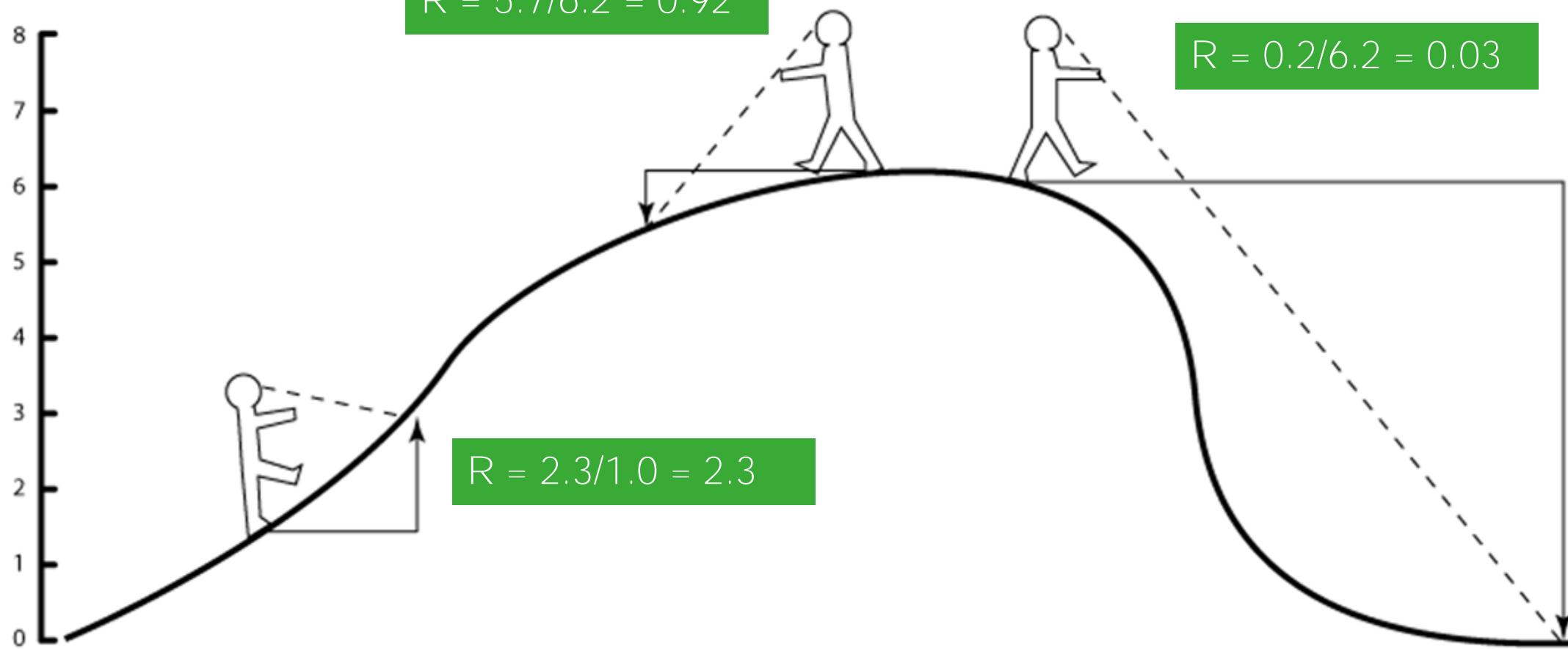
Stochastic
Approximation

→ Sampling Methods

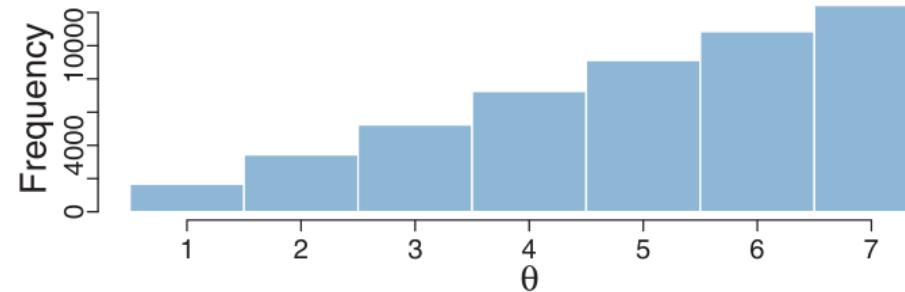
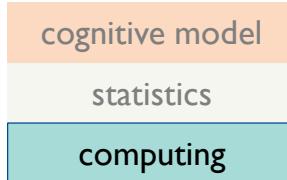
An MCMC Robot

cognitive model
statistics
computing

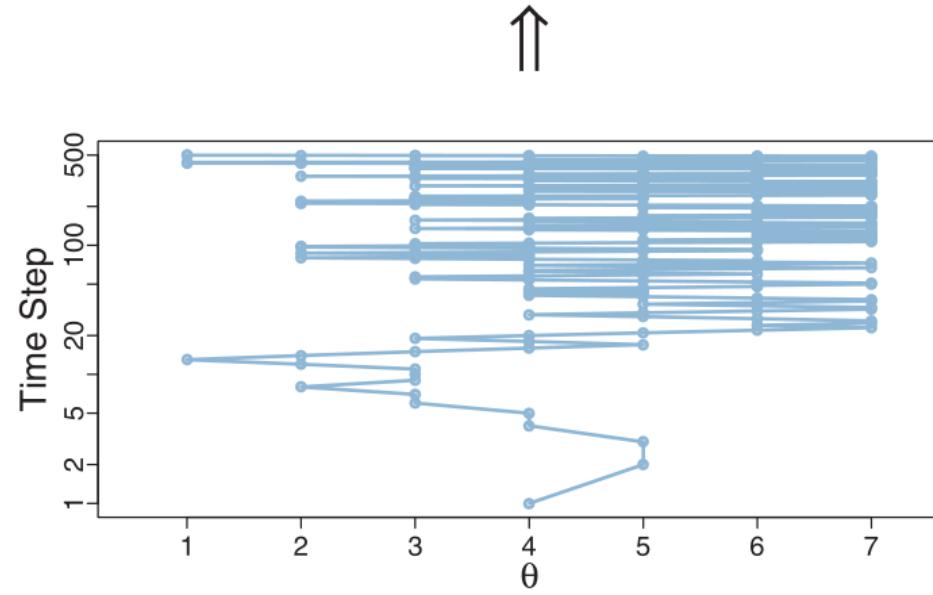
$$p(\theta | D) \propto p(D | \theta)p(\theta)$$



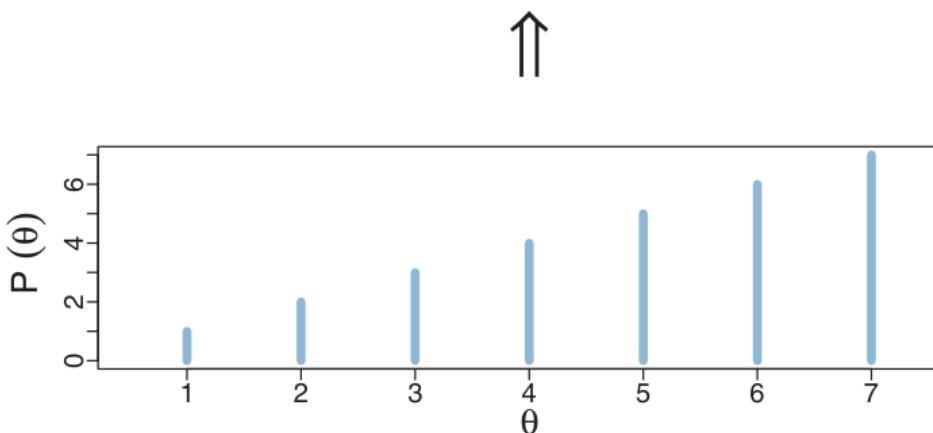
Sampling Example



MCMC summary



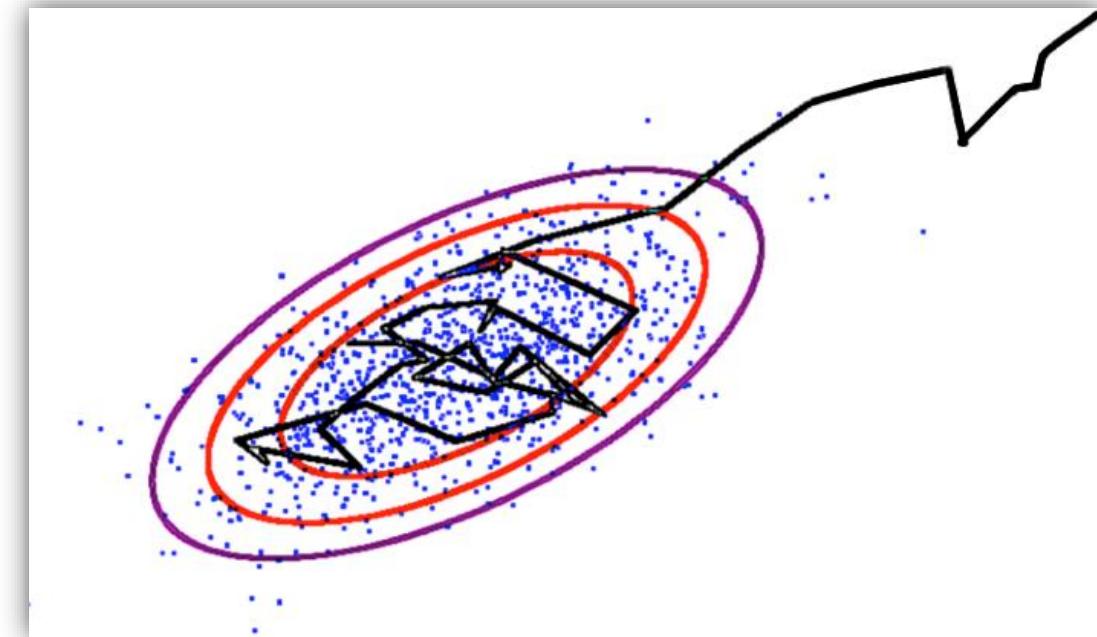
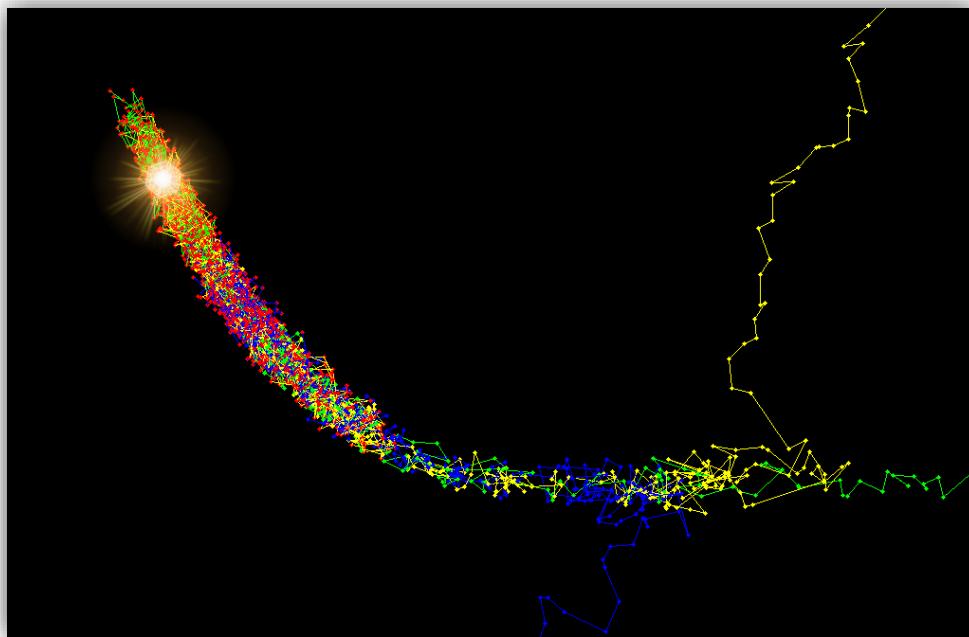
MCMC trace



True distribution

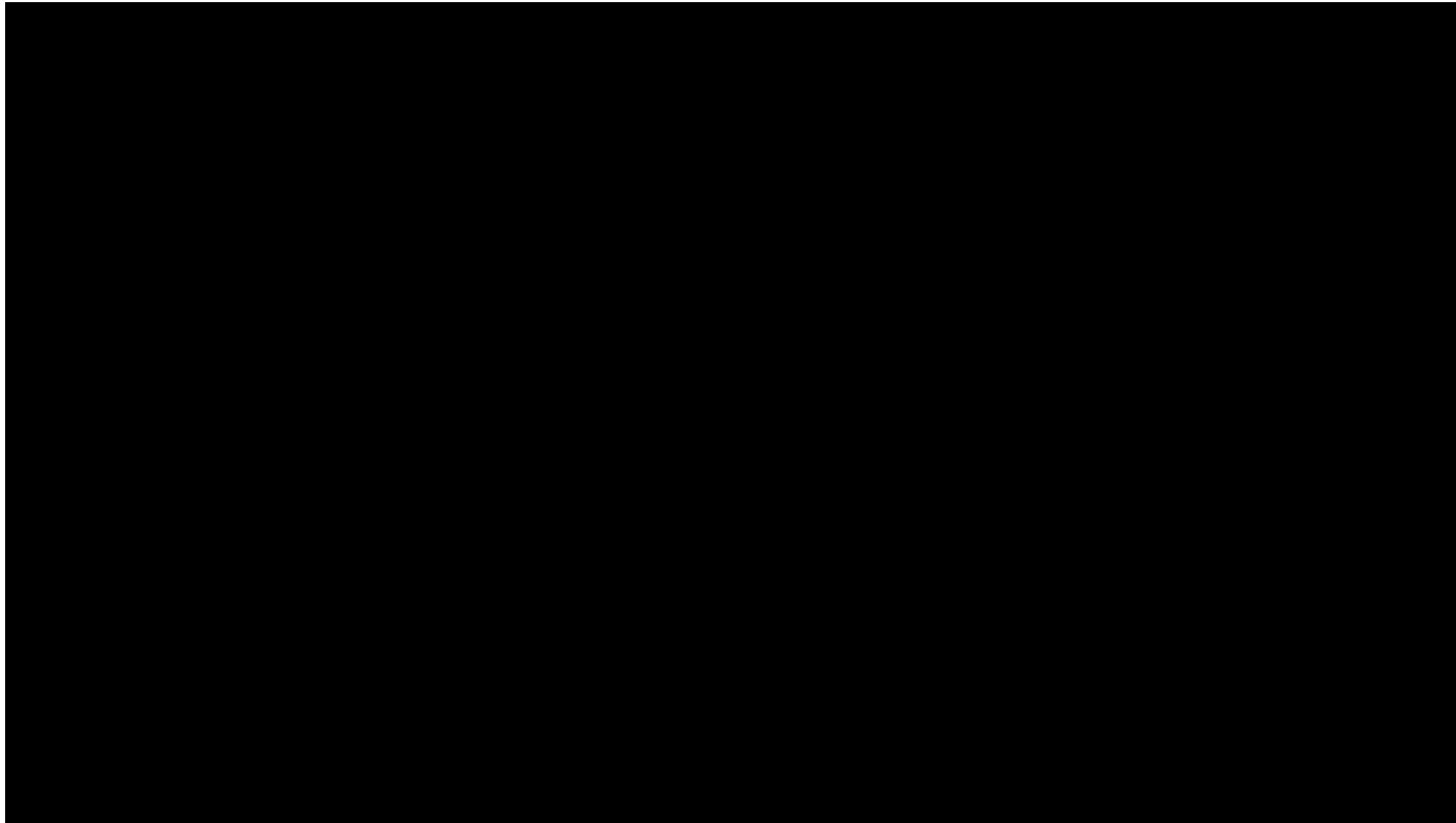
Visual Example

cognitive model
statistics
computing



Let's watch a video!

cognitive model
statistics
computing



MCMC Sampling Algorithms

cognitive model
statistics
computing

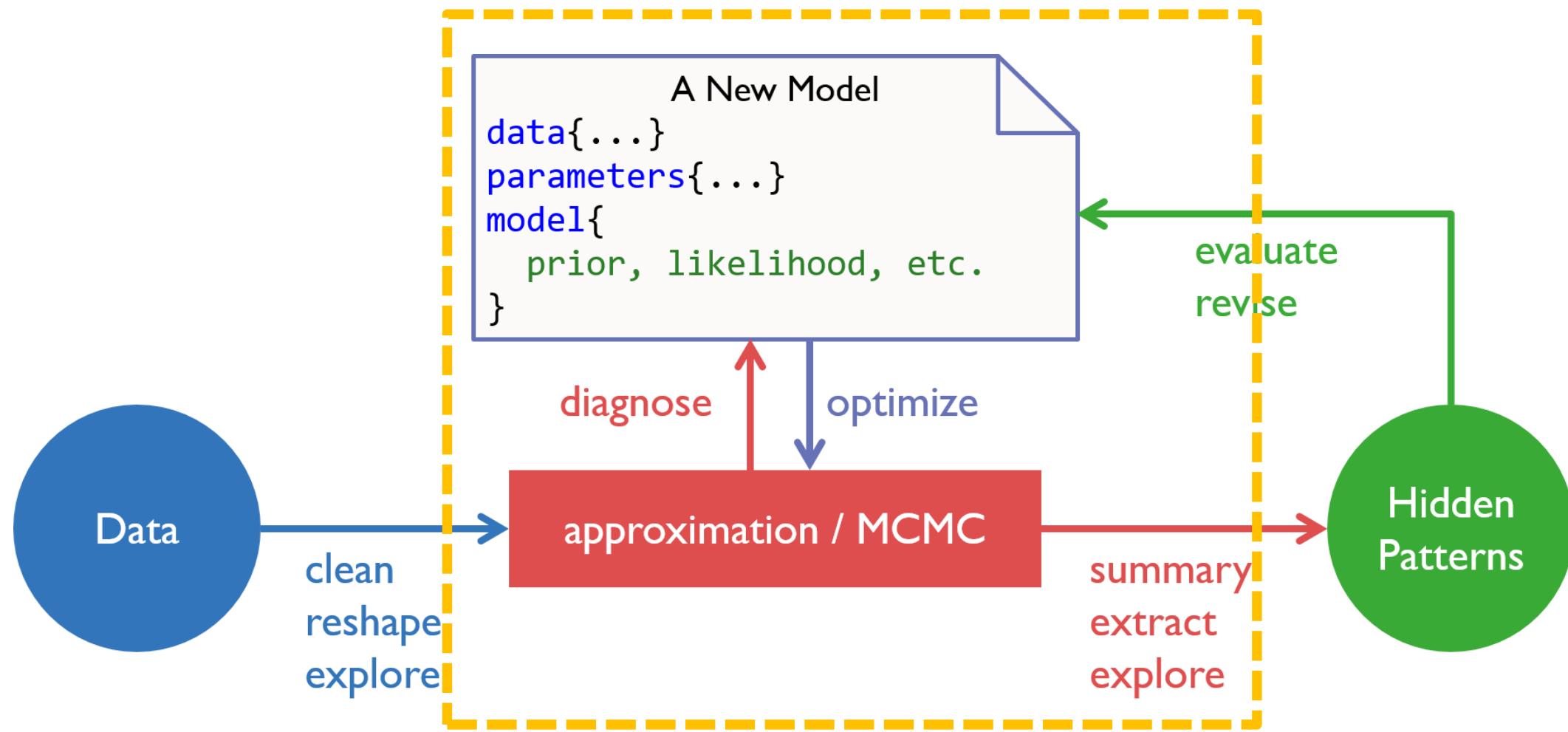
- Rejection sampling
- Importance sampling
- Metropolis algorithm
- Gibbs sampling → JAGS
- HMC sampling*



Stan!

STAN PROGRAMMING LANGUAGE





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 control flow
 - full support of vectorizing



Krzysztof Sakrejda (@sakrejda)

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



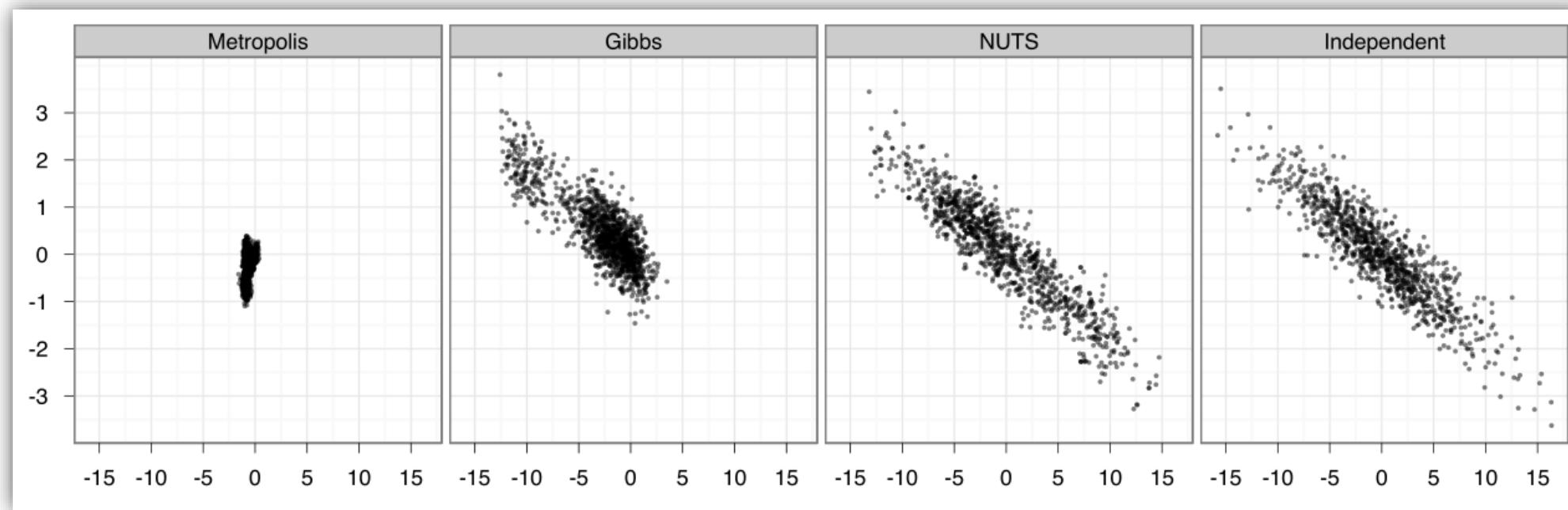
"Selling" Stan
discourse.mc-stan.org

27.03.18, 16:01

NUTS vs. Gibbs and Metropolis

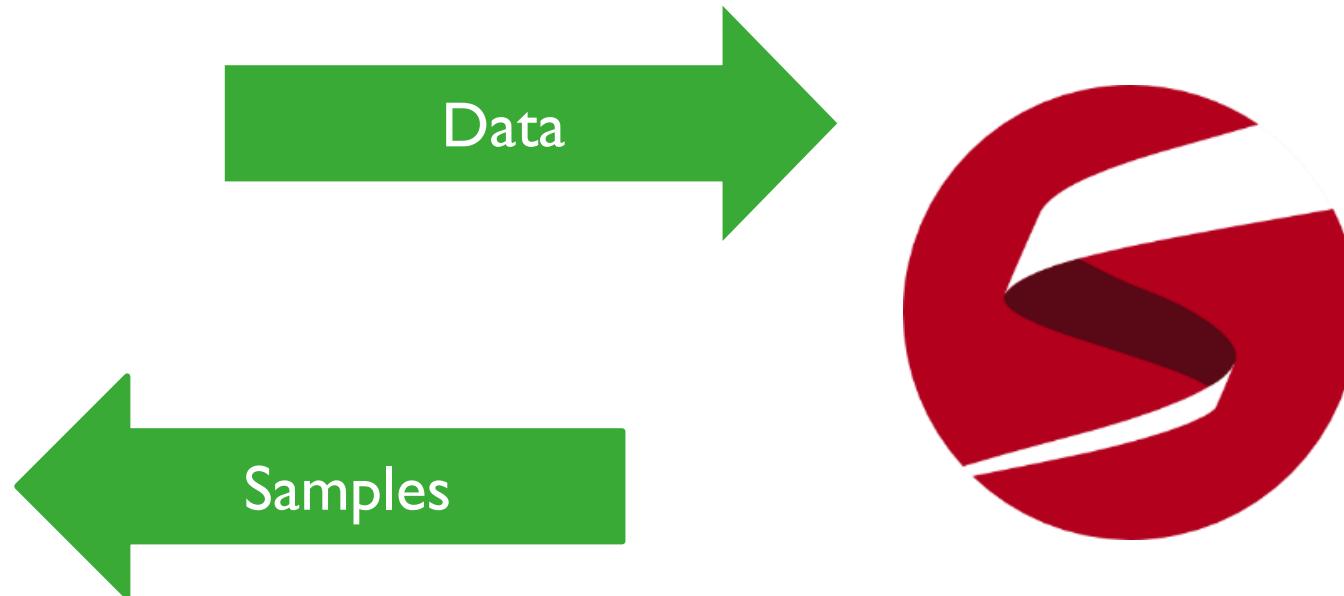
cognitive model
statistics
computing

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

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.

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.

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 / 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 generated by Stan version 2.24  
 * at file: bernoulli.stan, line: 1, column: 1  
 */  
#include <stan/math.hpp>  
#include <stan/math/prim.hpp>  
#include <stan/math/prim/meta.hpp>  
#include <stan/math/prim/meta.hpp>
```

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
 - //
 - /* ... */
- 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

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

cognitive model
statistics
computing

```
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 ;

REVISIT BINOMIAL MODEL



Binomial Model

cognitive model
statistics
computing

W L W W W L W L W

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



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

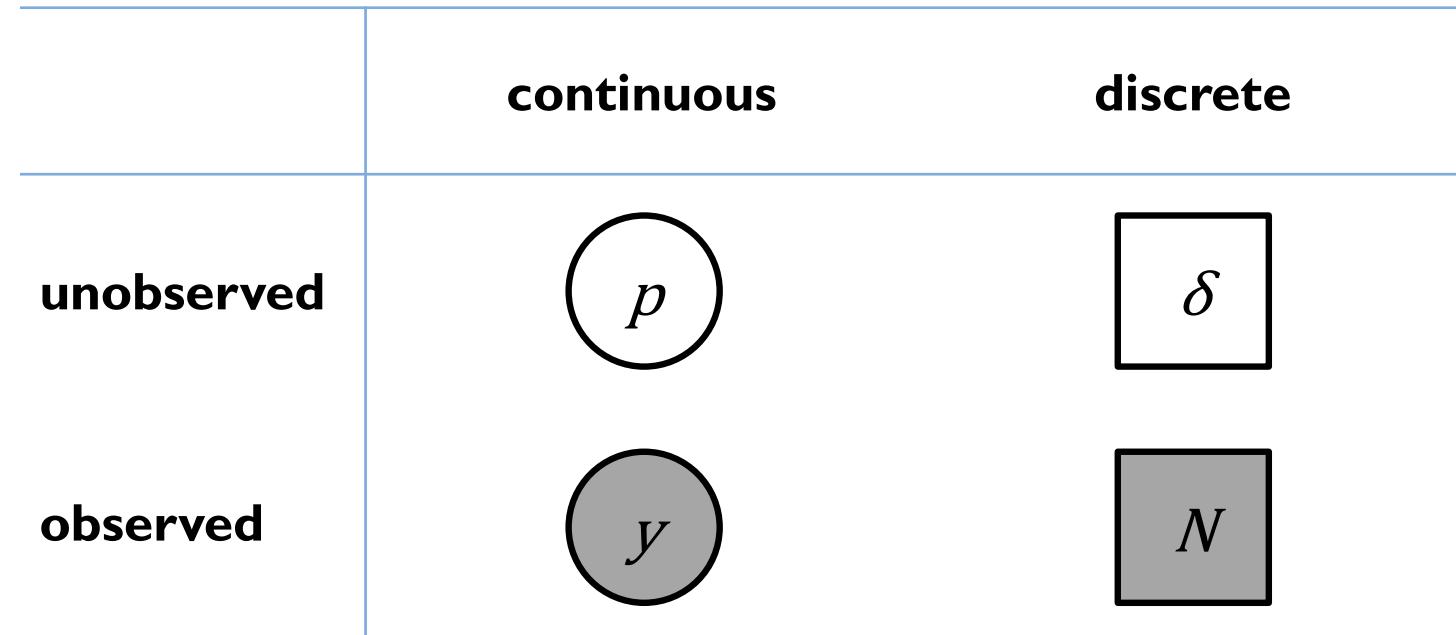
reads as:

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



Graphical Model Notations

cognitive model
statistics
computing

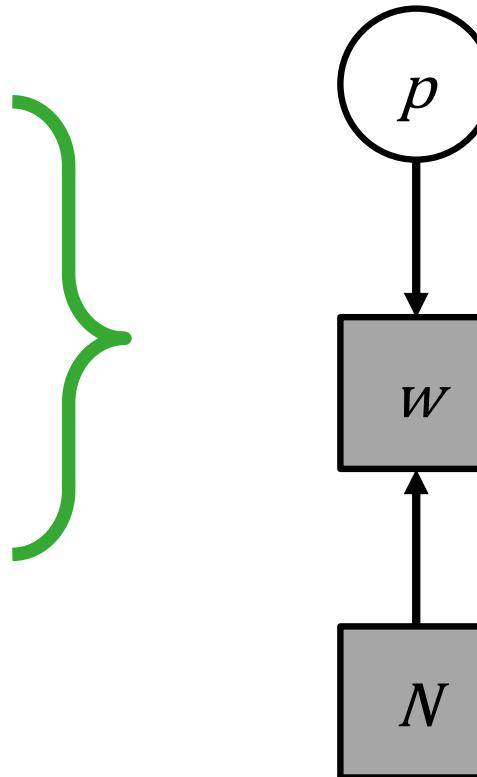


Binomial Model

cognitive model
statistics
computing

W L W W W L W L W

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



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

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



	continuous	discrete
unobserved	p	δ
observed	y	N

Binomial Model

cognitive model
statistics
computing

W L W W W L W L W

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



```
data {  
    int<lower=0> w;  
    int<lower=0> N;  
}  
  
parameters {  
    real<lower=0,upper=1> p;  
}  
  
model {  
    w ~ binomial(N, p);  
}
```

Running Binomial Model with Stan

cognitive model
statistics
computing

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

```
> R.version  
R version 3.5.0 (2018-04-23)
```

```
> stan_version()  
[1] "2.17.0"
```

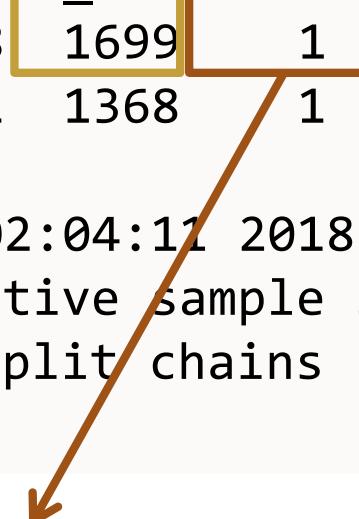
Model Summary

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	0.36	0.54	0.64	0.74	0.88	1699	1
lp__	-7.73	0.02	0.73	-9.92	-7.91	-7.44	-7.27	-7.21	1368	1

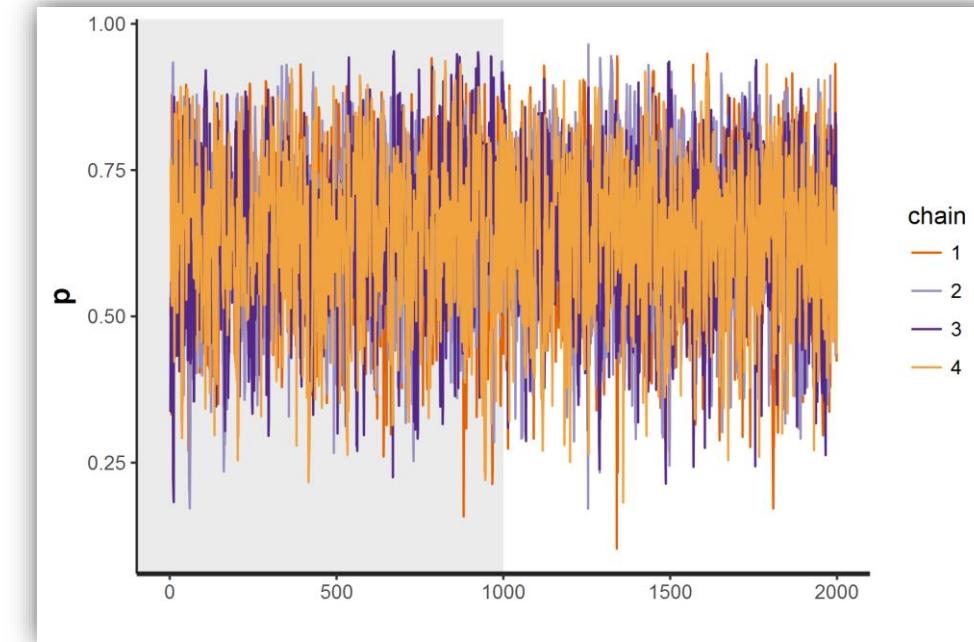
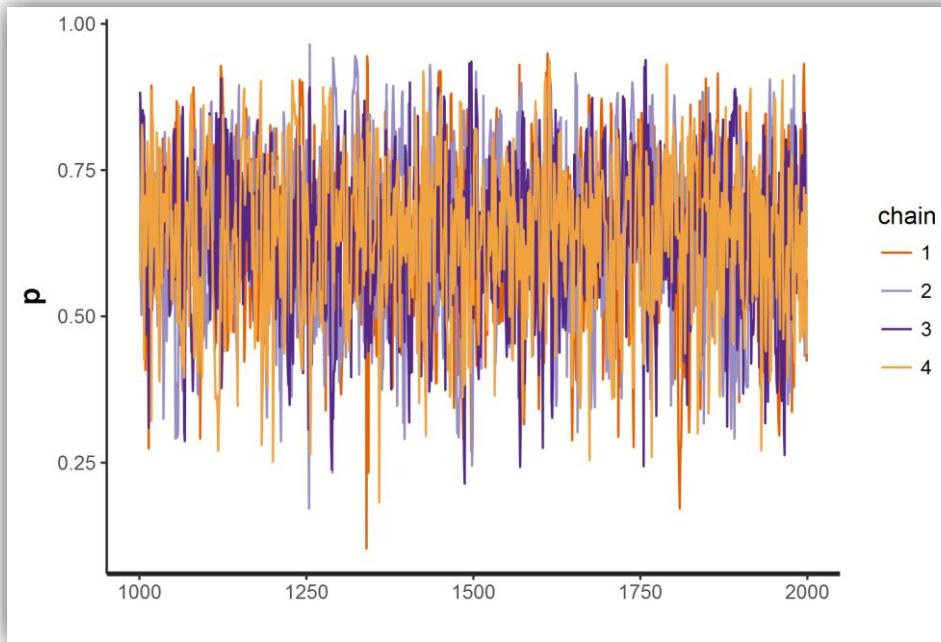
Samples were drawn using NUTS(diag_e) at Thu Jun 28 02:04:11 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).



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

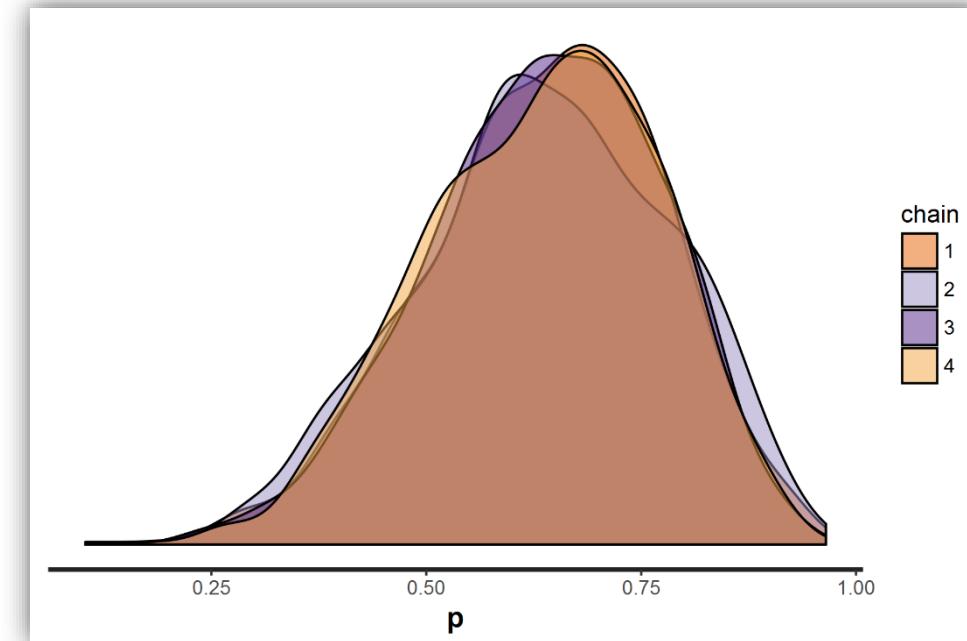
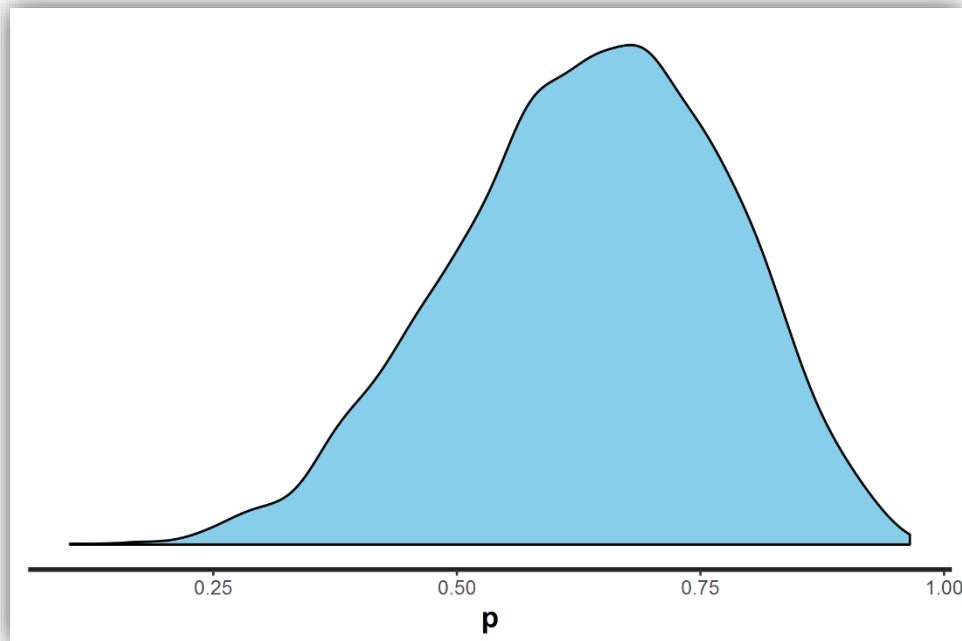
Diagnostics - traceplot

cognitive model
statistics
computing



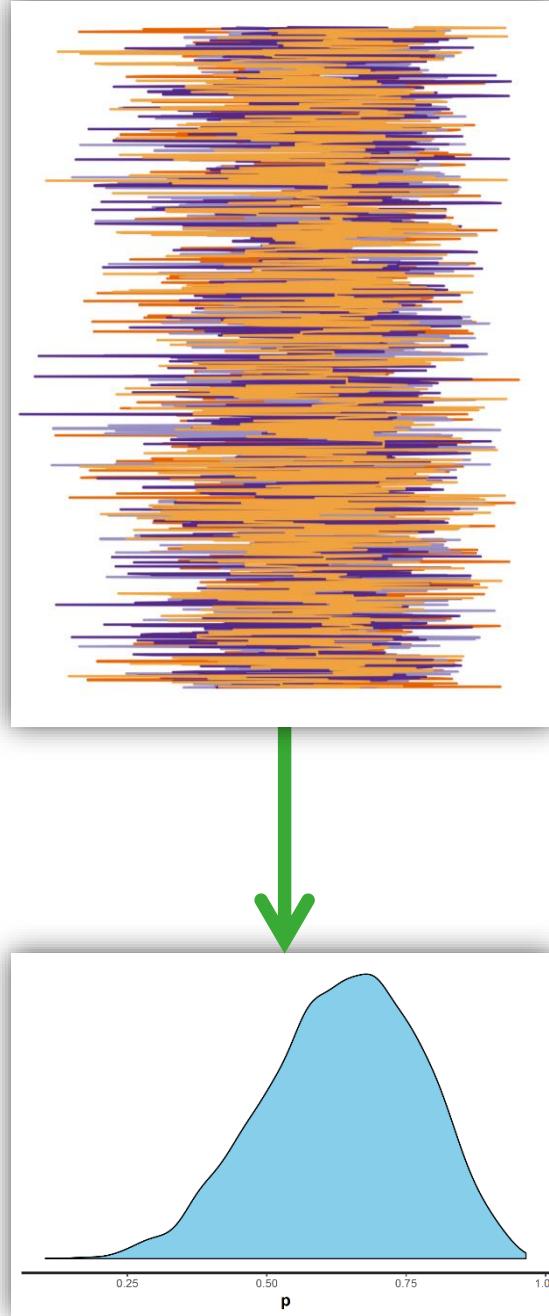
Diagnostics - density

cognitive model
statistics
computing

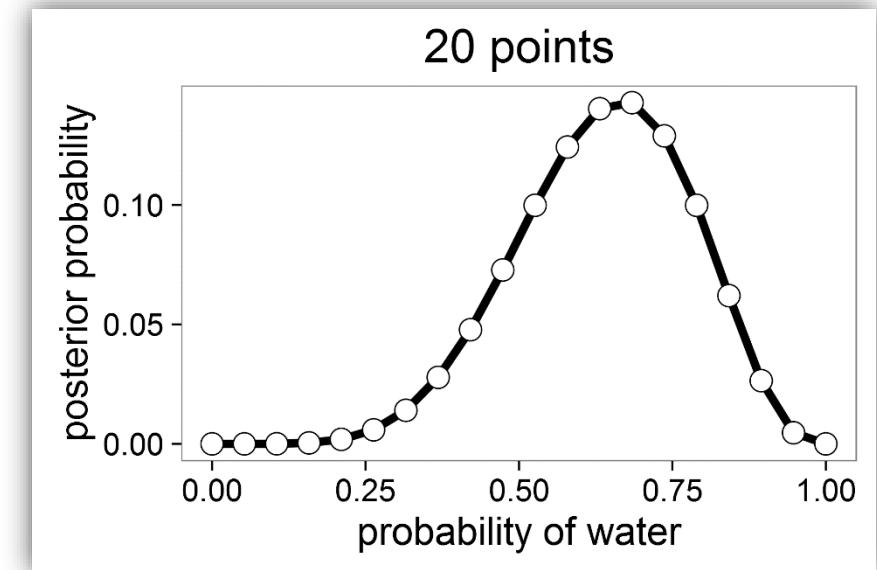


Diagnostics

MCMC



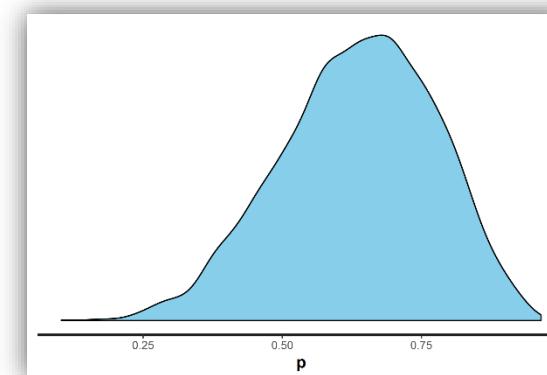
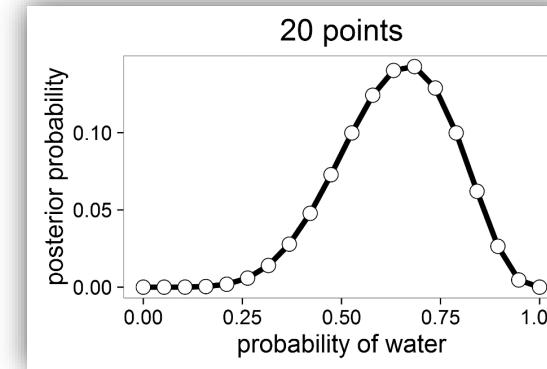
Grid Approximation



Draw a Conclusion?

cognitive model
statistics
computing

- $W = 6$ 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

```
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!