Bayesian Methods for Data Science (DATS 6450 - 11)

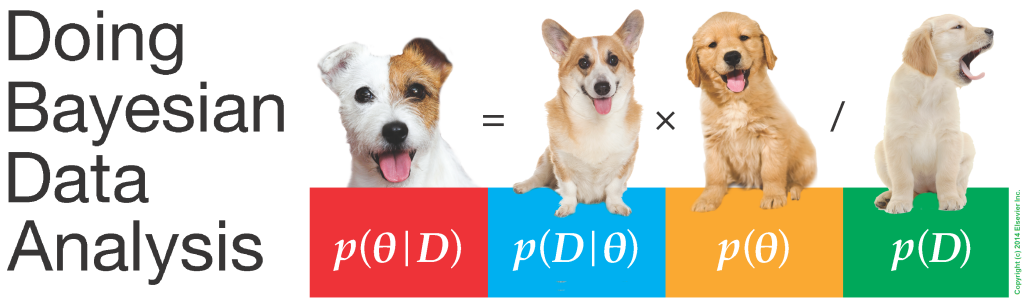
JAGS

Yuxiao Huang

Data Science, Columbian College of Arts & Sciences
George Washington University
yuxiaohuang@gwu.edu

October 9, 2019

Reference



Picture courtesy of the book website

- This set of slices is an excerpt of the book by Professor John K. Kruschke, with some trivial changes by the creator of the slides
- Please find the reference to and website of the book below:
  - Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier
  - `https://sites.google.com/site/doingbayesiandataanalysis/`

Overview

1. A complete example
2. Simplified scripts for frequently used analyses
3. Example: difference of biases
4. Sampling from the prior distribution in JAGS
5. Probability distributions available in JAGS

Introduction

- JAGS (Just Another Gibbs Sampler) is a system that automatically builds Markov chain Monte Carto (MCMC) samplers for complex hierarchical models
- JAGS succeeded the pioneering system named BUGS (Bayesian inference using Gibbs sampling), with different samplers and better usability across different computer-operating systems

Introduction

- JAGS takes as input a user's description of a hierarchical model for data, and returns an MCMC sample of the posterior distribution
- Figure 8.1 (see next page) shows the packages that we will be using for this course
- In particular, we will learn how to use JAGS from R via the `rjags` and `runjags` packages

Figure 8.1

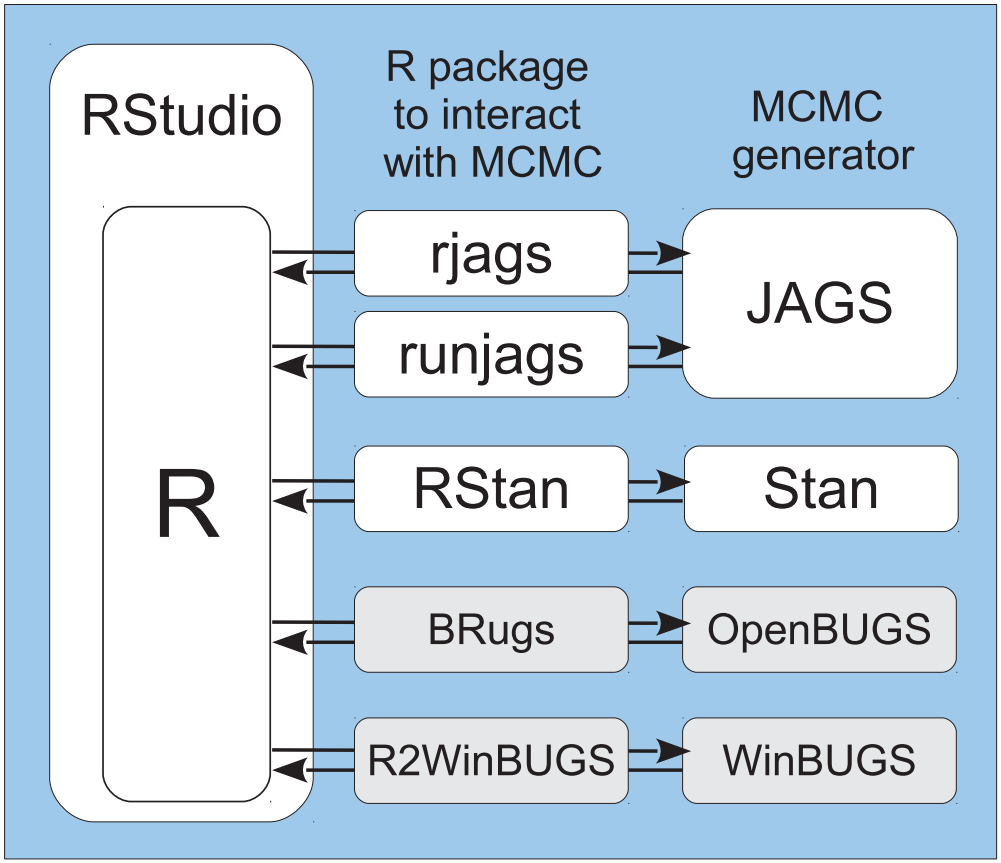


Figure 8.1: Relation of R programming language to other software tools. On the left, RStudio is an editor for interacting with R. The items on the right are various programs for generating MCMC samples of posterior distributions. The items in the middle are packages in R that interact with the MCMC generators. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Introduction

- To install JAGS, see `http://mcmc-jags.sourceforge.net/` (Google just another Gibbs sampler if broken link)
- Once JAGS is installed on your computer, invoke `R` and, at its command line, type
  - `install.packages("rjags")`
  - `install.packages("runjags")`

Parameterized model

- JAGS takes as input a parameterized model and outputs the posterior distributions of the parameters
- In the coin-flipping example, the parameterized model includes
  - a bernoulli likelihood with parameter θ
  - a beta prior with parameters A and B
- Figure 8.2 (see next page) shows the parameterized model
- Such diagrams are very useful for two important reasons:
  - they help with conceptual clarity, especially as we deal with more complex models
  - every arrow in the diagram has a corresponding line of code in the JAGS model specification

Figure 8.2

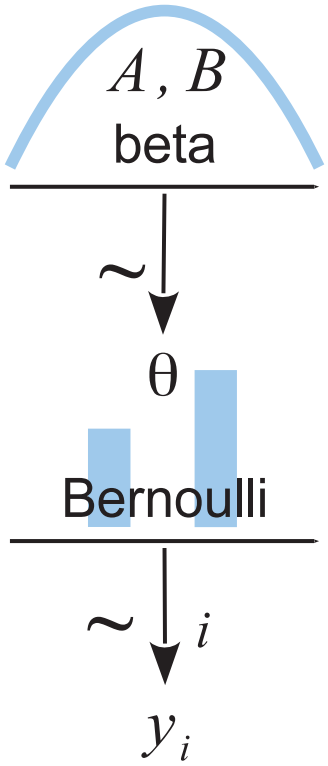


Figure 8.2: Diagram of model with Bernoulli likelihood and beta prior. The pictures of the distributions are intended as stereotypical icons, and are not meant to indicate the exact forms of the distributions. Diagrams like this should be scanned from the bottom up, starting with the data y_i and working upward through the likelihood function and prior distribution. Every arrow in the diagram has a corresponding line of code in JAGS model specification. *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Load data

- For comma separated values (CSV) files, they can be read into `R` with the `read.csv` function
- To bundle the data for JAGS, we put the data and length into a list

Specify model

- Again, every arrow in Figure 8.2 has a corresponding line of code in the JAGS model specification
- Particularly, the specification follows the order:

$$\text{likelihood} \rightarrow \text{prior}$$

- We create the specification as a character string in `R`, then save the string to a text file, then send the text file to JAGS
- See `Jags-ExampleScript.R` for details (available on the book website)

Initialize chains

- The efficiency of the MCMC process can be improved by reasonable starting values of the parameters
- In general, a useful choice for initial values of the parameters is their maximum likelihood estimate (MLE)
  - MLE is the value of the parameter that maximizes the likelihood
- **Q:** Why MLE?

Initialize chains

- The efficiency of the MCMC process can be improved by reasonable starting values of the parameters
- In general, a useful choice for initial values of the parameters is their maximum likelihood estimate (MLE)
  - MLE is the value of the parameter that maximizes the likelihood
- **Q:** Why MLE?
- **A:** Because the posterior distribution is usually not significantly different from the likelihood, if the prior is noncommittal
- For the bernoulli likelihood, the MLE is

$$\theta = \frac{z}{N},$$

that is, the proportion of heads

Initialize chains

- When there are multiple chains, we can specify different or identical initial values for each chain
- The rjags package provides three ways of initializing chains:
  - specify a single initial point for all the chains, by using a `list`
  - specify a specific initial point for each chain, by using:
    - a `list` of `list`
    - function call
- See `Jags-ExampleScript.R` for details (available on the book website)

Generate chains

- There are three steps:
  1. JAGS figure out the appropriate samplers for the model based on the provided information
  2. runs the chains for a burn-in period
  3. runs and records the MCMC sample that we will subsequently examine
- See `Jags-ExampleScript.R` for details (available on the book website)

Examine chains

- **Q:** Recall, what should be checked about the representativeness of the MCMC samples?

Examine chains

- **Q:** Recall, what should be checked about the representativeness of the MCMC samples?
- **A:**
  - trace plot
  - density plot
  - shrink factor
  - autocorrelation
- Figure 8.3 (see next page) shows such information for the θ parameter in the example
- See `Jags-ExampleScript.R` for details (available on the book website)

Figure 8.3

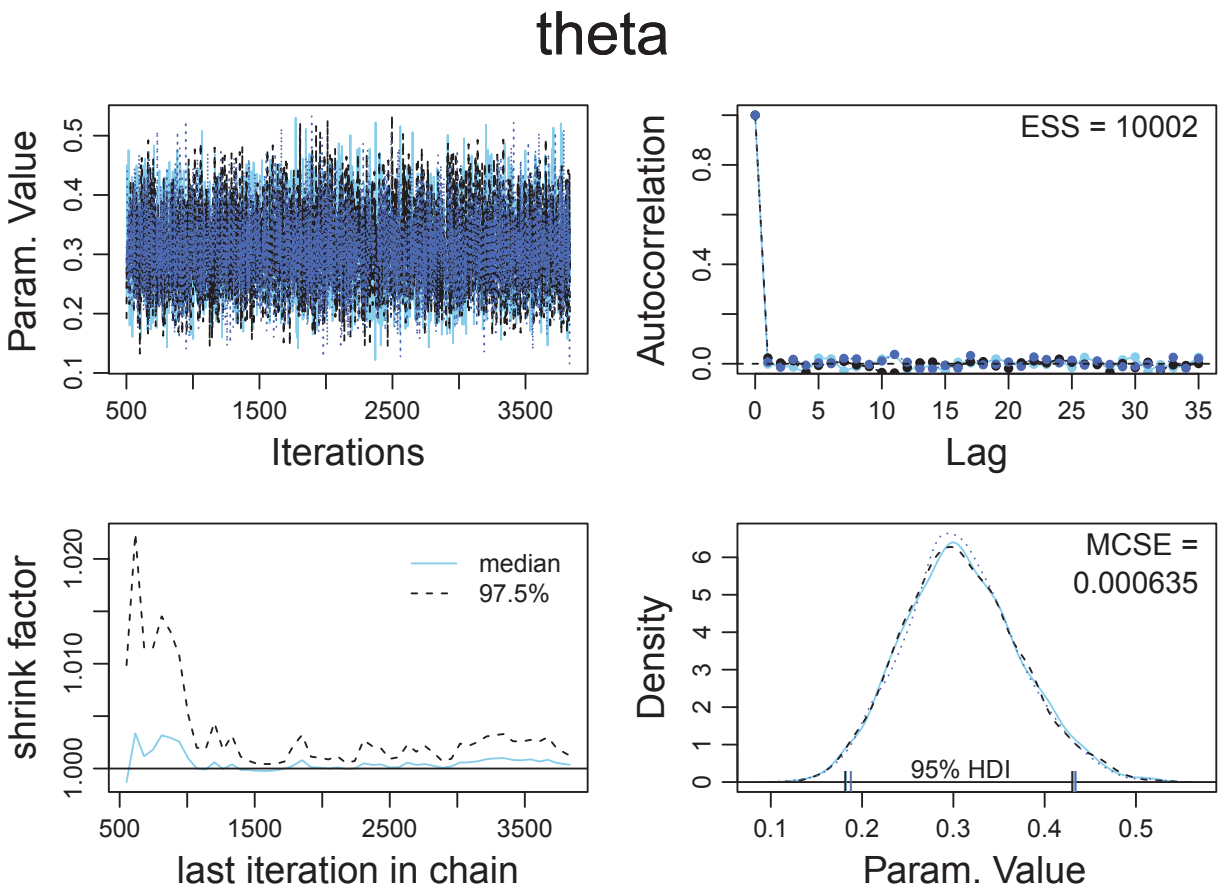


Figure 8.3: Convergence diagnostics for output of JAGS... Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

The `plotPost` function

- Figure 8.4 (see next page) shows the MCMC sample from the posterior distribution
- This figure was created by function `plotPost`
- See `Jags-ExampleScript.R` for details (available on the book website)

Figure 8.4

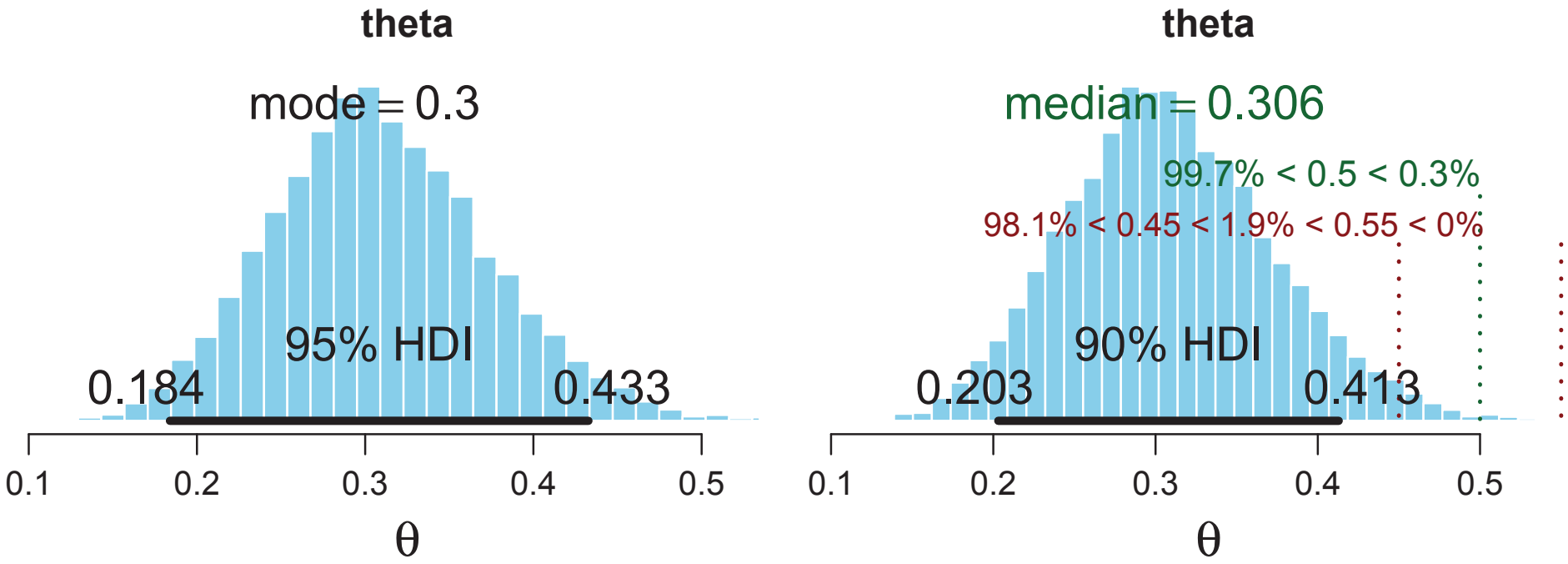


Figure 8.4: Posterior distribution based on output from JAGS, plotted twice using different options in the `plotPost` function. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Simplified scripts for frequently used analyses

- The previous script assembled:
  - the data list
  - the model specification
  - the initial values
  - the running of the chains
- This has been collapsed into a single function called `genMCMC`
- See `Jags-Ydich-Xnom1subj-MbernBeta-Example.R` for details (available on the book website)

Example: difference of biases

- There are cases where we want to compare the difference between two parameters (e.g., the biases of two coins)
- We can use JAGS to:
  - generate MCMC chain for each parameter
  - compare the difference between the two parameters
- This is shown in Figure 8.6 (see next page)
- See `Jags-Ydich-XnomSsubj-MbernBeta-Example.R` for details (available on the book website)

Figure 8.6

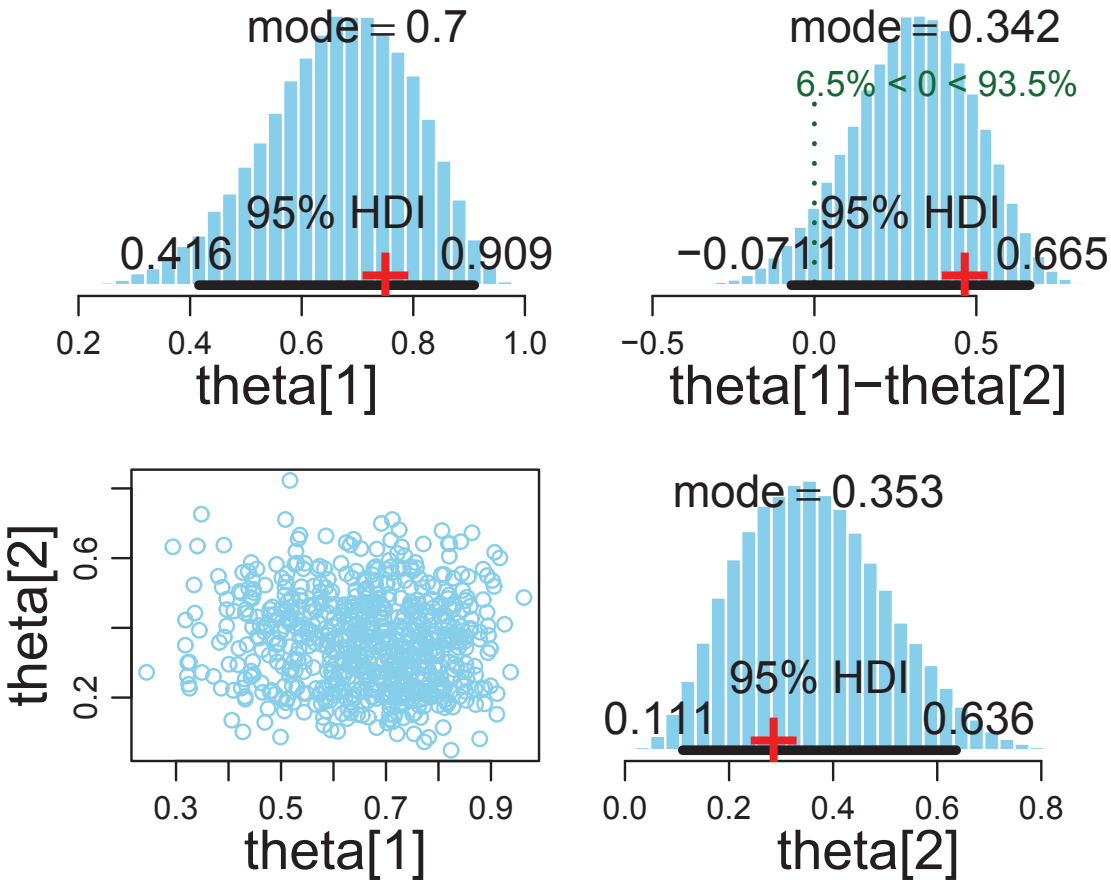


Figure 8.6: Posterior distribution created by JAGS. Compare the upper-right panel with Figure 7.9. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Sampling from the prior distribution in JAGS

- Besides the posterior, there are times where we may want to examine the prior, to check whether the prior that we explicitly or implicitly specified actually matches what we intended
- **Q:** Can we have JAGS generate an MCMC sample from, not the posterior, but the prior?

Sampling from the prior distribution in JAGS

- Besides the posterior, there are times where we may want to examine the prior, to check whether the prior that we explicitly or implicitly specified actually matches what we intended
- **Q:** Can we have JAGS generate an MCMC sample from, not the posterior, but the prior?
- **A:** Yes, by simply running JAGS with no data included
- Figure 8.7 (see next page) shows the prior sampled by JAGS
- See `Jags-Ydich-XnomSsubj-MbernBeta.R` for details (available on the book website)

Figure 8.7

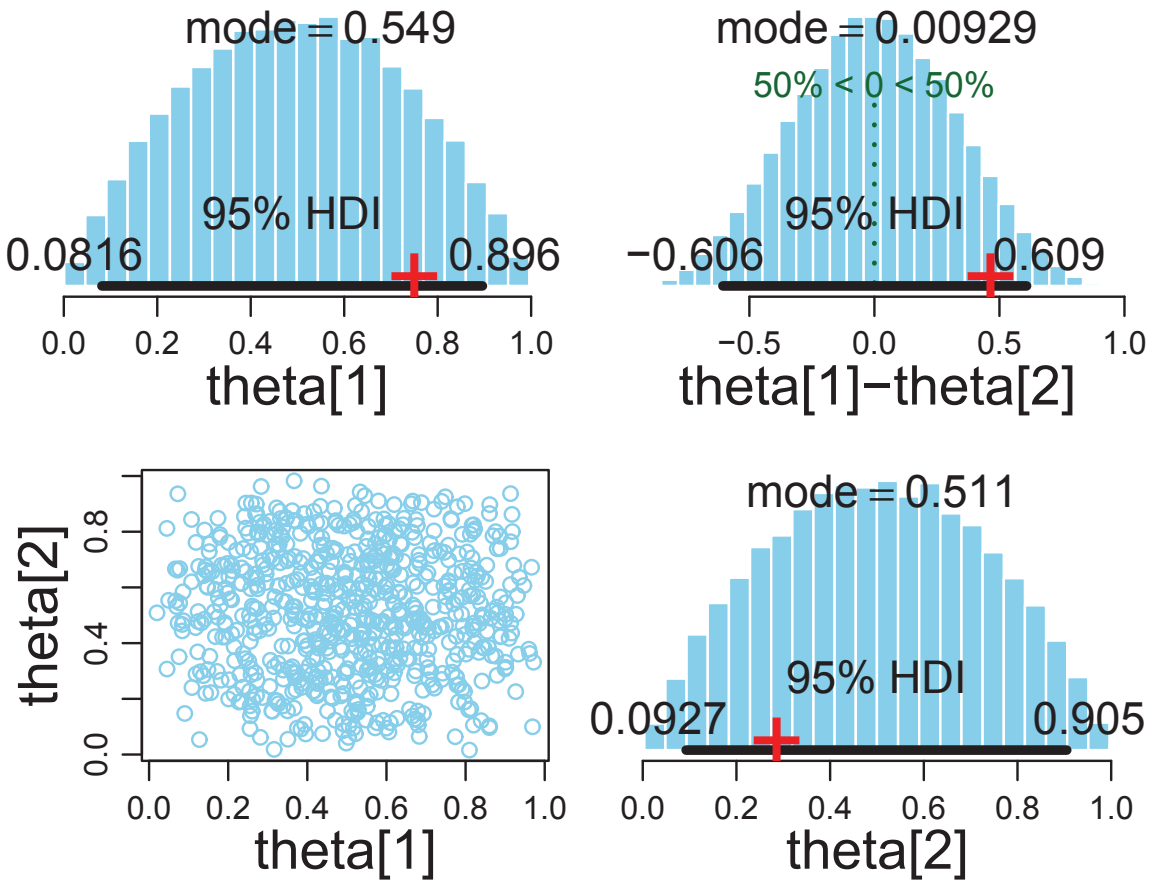


Figure 8.7: The prior distribution for difference of biases. Compare with Figure 8.6. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Functions that are not built in

- JAGS has a large collection of frequently used probability distributions that are built-in
  - beta
  - gamma
  - normal
  - bernoulli
  - binomial
  - ...
- **Q:** What if the prior or likelihood distribution is not built-in?

Functions that are not built in

- JAGS has a large collection of frequently used probability distributions that are built-in
  - beta
  - gamma
  - normal
  - bernoulli
  - binomial
  - ...
- **Q:** What if the prior or likelihood distribution is not built-in?
- **A:**
  - "Bernoulli ones trick"
  - "Poisson zeros trick"

Defining new likelihood functions

- Consider a likelihood function that is not built-in

$$\text{pdf} = 6\theta(1-\theta)$$

- If the function were built-in, then we would be able to specify in JAGS something like

$$y[i] \sim \text{dpdf}(\theta)$$

- **Q:** What does this do in JAGS?

Defining new likelihood functions

- Consider a likelihood function that is not built-in

$$\text{pdf} = 6\theta(1-\theta)$$

- If the function were built-in, then we would be able to specify in JAGS something like

$$y[i] \sim \text{dpdf}(\theta)$$

- **Q:** What does this do in JAGS?
- **A:** It calculates:
  1. the prior, given the prior function and θ (generated by the propositional distribution)
  2. the likelihood, given the likelihood function ($y[i]$) and θ
  3. the product of the above prior and likelihood

Defining new likelihood functions

- Now consider a magical built-in function, X, such that

$$y[i] \sim dX(\theta)$$

  produces a likelihood density that equals θ
- **Q:** What happens when we replace θ with the likelihood density of $6x(1-x)$?

Defining new likelihood functions

- Now consider a magical built-in function, X, such that

$$y[i] \sim dX(\theta)$$

  produces a likelihood density that equals θ
- **Q:** What happens when we replace θ with the likelihood density of $6x(1-x)$?
- **A:** The above code produces likelihood density that equals that of $6x(1-x)$, problem solved
- **Q:** What is this magical built-in function, X?

Bernoulli ones trick

- **A:**
  - for bernoulli distribution, when we do

$$1 \sim \text{dbern}(\theta),$$

    it yields a likelihood density that equals θ
  - thus, bernoulli distribution can be the magical function X
  - see `Jags-ExampleScript-NotBuiltIn-bernoulli.R` for details (available on the blackboard)

Poisson zeros trick

- **A:**
  - for poisson distribution, when we do

$$0 \sim \text{dpois}(-\log(\theta)),$$

    it yields a likelihood density that equals θ
  - thus, poisson distribution can also be the magical function X
  - wee `Jags-ExampleScript-NotBuiltIn-poisson.R` for details (available on the blackboard)

Faster sampling with parallel processing in runjags

- When called from `rjags`, JAGS run chains sequentially by using only one core
- When called from `runjags`, JAGS can run chains in parallel by using multiple cores, thereby decreasing the total time needed
  - to install the package, at R's command line, type

    ```
    install.packages("runjags")
    ```

  - to load the package into R's memory, type

    ```
    library("runjags")
    ```

  - for details go to: `https://cran.r-project.org/web/packages/runjags/` (Google `runjags` if broken link)