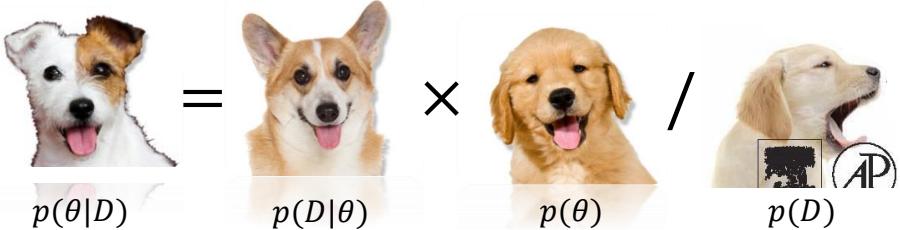


Doing Bayesian Data Analysis



John K. Kruschke

Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

Bayesian Reasoning

The role of data is to re-allocate credibility:

Prior Credibility with New Data
→ Posterior Credibility

via Bayes' rule

© John K. Kruschke, 2013

3

Bayesian Reasoning

The role of data is to re-allocate credibility:

Bayesian reasoning in everyday life is
intuitive:

© John K. Kruschke, 2013

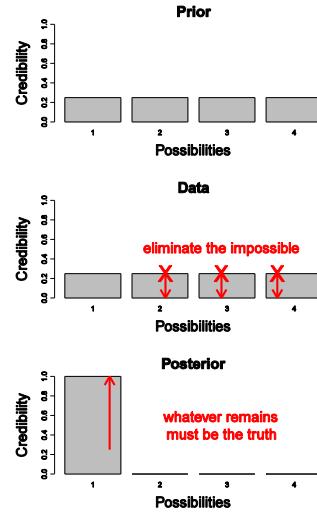
4

Bayesian Reasoning

The role of data is to re-allocate credibility:

Bayesian reasoning in everyday life is intuitive:

Sherlock Holmes: "How often have I said to you that when you have eliminated the impossible, whatever remains, however improbable, must be the truth?" (Doyle, 1890)



© John K. Kruschke, 2013

5

Bayesian Reasoning

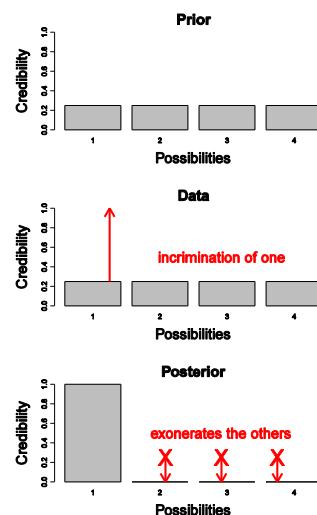
The role of data is to re-allocate credibility:

Bayesian reasoning in everyday life is intuitive:

Sherlock Holmes: "How often have I said to you that when you have eliminated the impossible, whatever remains, however improbable, must be the truth?" (Doyle, 1890)

Judicial exoneration: For unaffiliated suspects, the incrimination of one exonerates the others.

Credibility of the claim that the suspect committed the crime.



© John K. Kruschke, 2013

6

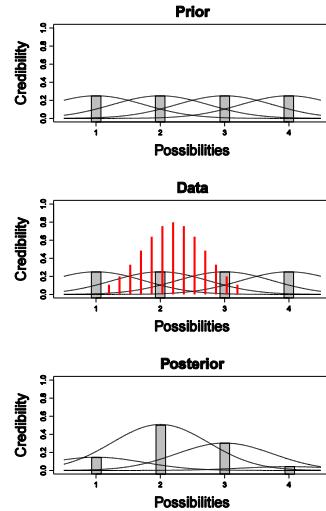
Bayesian Data Analysis

The role of data is to re-allocate credibility:

Bayesian reasoning in data analysis is intuitive:

Possibilities are *parameter values* in a model, such as the *mean* of a normal distribution.

We reallocate credibility to parameter values that are consistent with the data.



© John K. Kruschke, 2013

7

Bayesian Data Analysis

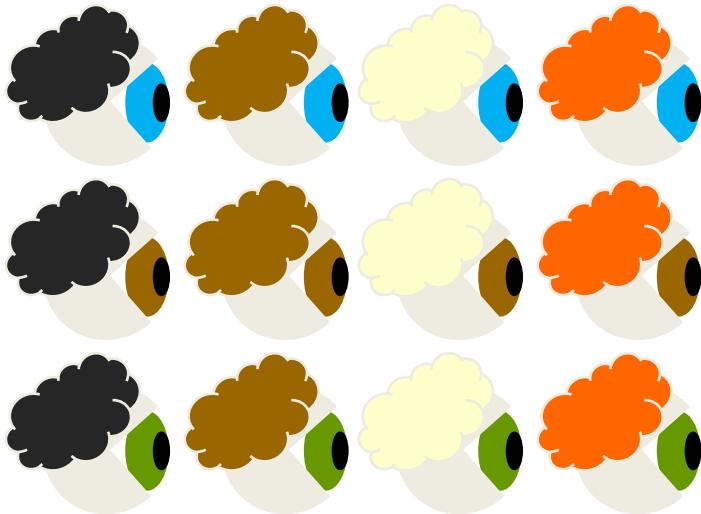
The role of data is to re-allocate credibility:

1. Define a meaningful descriptive model.
2. Establish prior credibility regarding parameter values in the model. The prior credibility must be acceptable to a skeptical scientific audience.
3. Collect data.
4. Use Bayes' rule to re-allocate credibility to parameter values that are most consistent with the data.

© John K. Kruschke, 2013

8

Joint, marginal and conditional probability



© John K. Kruschke, 2013

10

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-----------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/ Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

Proportion of sample of University of Delaware students 1974,
N=592. Data adapted from Snee (1974).

© John K. Kruschke, 2013

11

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

Joint probabilities: $p(e,h)$

For example, $p(e=\text{blue}, h=\text{black}) = .03$

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

Marginal probabilities: $p(e)$

For example, $p(e=\text{blue}) = .36$
 $= \sum_h p(e=\text{blue}, h)$

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |



Marginal probabilities: $p(h)$
For example, $p(h=\text{black}) = .18$
 $= \sum_e p(e,h=\text{black})$

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |



**Marginal probabilities:
 $p(h)$ without info about e**

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .21 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

*Conditional/ probabilities:
 $p(h|e=blue)$ is $p(h)$ with info that $e=blue$*

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .21 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

*Conditional/ probabilities:
 $p(h|e=blue) = p(h,e=blue) / p(e=blue)$*

| Eye Color | Hair Color | | | | |
|-----------|-----------------|-----------------|-----------------|-----------------|----------------|
| | Black | Brunette | Blond | Red | |
| Blue | .03/.36 =.08 | .14/.36 =.39 | .16/.36 =.45 | .03/.36 =.08 | .36/.36 =.1 |

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

Conditional probabilities:
 $p(h|e=\text{blue}) = p(h,e=\text{blue}) / p(e=\text{blue})$

| Eye Color | Hair Color | | | | |
|-----------|-----------------|-----------------|-----------------|-----------------|---------------|
| | Black | Brunette | Blond | Red | |
| Blue | .03/.36 =.08 | .14/.36 =.39 | .16/.36 =.45 | .03/.36 =.08 | .36/.36 =1 |

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

Conditional probabilities:
 $p(h|e=\text{brown}) = p(h,e=\text{brown}) / p(e=\text{brown})$

| Eye Color | Hair Color | | | | |
|-----------|-----------------|-----------------|-----------------|-----------------|---------------|
| | Black | Brunette | Blond | Red | |
| Brown | .12/.37 =.32 | .20/.37 =.54 | .01/.37 =.03 | .04/.37 =.11 | .37/.37 =1 |

Joint, marginal and conditional probability

| Eye Color | Hair Color | | | | |
|-------------|------------|----------|-------|-----|-----|
| | Black | Brunette | Blond | Red | |
| Blue | .03 | .14 | .16 | .03 | .36 |
| Brown | .12 | .20 | .01 | .04 | .37 |
| Hazel/Green | .03 | .14 | .04 | .05 | .27 |
| | .18 | .48 | .21 | .12 | |

Conditional probabilities:
 $p(h | e=\text{brown}) = p(h,e=\text{brown}) / p(e=\text{brown})$

| Eye Color | Hair Color | | | | |
|-----------|-----------------|-----------------|-----------------|-----------------|---------------|
| | Black | Brunette | Blond | Red | |
| Brown | .12/.37 =.32 | .20/.37 =.54 | .01/.37 =.03 | .04/.37 =.11 | .37/.37 =1 |

Bayes' rule

By definition, $p(h | e) = \frac{p(e, h)}{p(e)}$ and $p(e | h) = \frac{p(e, h)}{p(h)}$.

Bayes' rule

By definition, $p(h | e) = \frac{p(e, h)}{p(e)}$ and $p(e | h) = \frac{p(e, h)}{p(h)}$.

Hence $p(h | e)p(e) = p(e, h) = p(e | h)p(h)$, and

Bayes' rule

By definition, $p(h | e) = \frac{p(e, h)}{p(e)}$ and $p(e | h) = \frac{p(e, h)}{p(h)}$.

Hence $p(h | e)p(e) = p(e, h) = p(e | h)p(h)$, and

$$p(h | e) = p(e | h) p(h) / p(e)$$

Bayes' rule

By definition, $p(h | e) = \frac{p(e, h)}{p(e)}$ and $p(e | h) = \frac{p(e, h)}{p(h)}$.

Hence $p(h | e)p(e) = p(e, h) = p(e | h)p(h)$, and

$$p(h | e) = p(e | h) p(h) / p(e)$$

↑ ↑
with info without
about e info
 about e

Bayes' rule

$$p(h | e) = p(e | h) p(h) / p(e)$$

So what's the big deal?

→ Ramifications when applied to data and parameter values!

$$\underbrace{p(\theta | D)}_{\text{posterior}} = \underbrace{p(D | \theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}} / \underbrace{p(D)}_{\text{evidence}}$$

↑ ↑
with info without
about D info
 about D

Bayes' rule applied to data and parameter values

| Data | Parameter | | | |
|------|-----------|--------------------------------------|-----|--------|
| | ... | θ | ... | |
| : | | : | | : |
| D | ... | $p(D \theta) = p(D \theta)p(\theta)$ | ... | $p(D)$ |
| : | | : | | : |
| | ... | $p(\theta)$ | ... | |

Possible values of parameter


Possible values of data


Bayes' rule applied to data and parameter values

| Data | Parameter | | | |
|------|-----------|--------------------------------------|-----|--------|
| | ... | θ | ... | |
| : | | : | | : |
| D | ... | $p(D \theta) = p(D \theta)p(\theta)$ | ... | $p(D)$ |
| : | | : | | : |
| | ... | $p(\theta)$ | ... | |

Marginal probabilities:
 $p(\theta)$ without info about D
a.k.a. "prior"

Bayes' rule applied to data and parameter values

| Data | Parameter | | | |
|----------|-----------|--------------------------------------|-----|----------|
| | ... | θ | ... | |
| \vdots | | \vdots | | \vdots |
| D | ... | $p(D \theta) = p(D \theta)p(\theta)$ | ... | $p(D)$ |
| \vdots | | \vdots | | \vdots |
| | ... | $p(\theta)$ | ... | |

Conditional probabilities:
 $p(\theta|D)$ is $p(\theta)$ with info about D
a.k.a. "posterior"

Bayes' rule applied to data and parameter values

| Data | Parameter | | | |
|----------|-----------|--------------------------------------|-----|----------|
| | ... | θ | ... | |
| \vdots | | \vdots | | \vdots |
| D | ... | $p(D,\theta) = p(D \theta)p(\theta)$ | ... | $p(D)$ |
| \vdots | | \vdots | | \vdots |
| | ... | $p(\theta)$ | ... | |

Conditional probabilities:
 $p(\theta|D)$ is $p(\theta)$ with info about D
a.k.a. "posterior"

$$\begin{aligned}
 p(\theta|D) &= p(D,\theta) / p(D) \\
 &= p(D|\theta) p(\theta) / p(D) \\
 &= p(D|\theta) p(\theta) / \sum_{\theta_c} p(D|\theta_c) p(\theta_c)
 \end{aligned}$$

Bayes' rule applied to data and parameter values

| Data | Parameter | | | |
|----------|-----------|--------------------------------------|-----|----------|
| | ... | θ | ... | |
| \vdots | | \vdots | | \vdots |
| D | ... | $p(D \theta) = p(D \theta)p(\theta)$ | ... | $p(D)$ |
| \vdots | | | | \vdots |
| | ... | $p(\theta)$ | ... | |



Reallocate credibility across candidate parameter values, given data value.

Conditional probabilities:
 $p(\theta|D)$ is $p(\theta)$ with info about D
a.k.a. "posterior"

$$\begin{aligned} p(\theta|D) &= p(D,\theta) / p(D) \\ &= p(D|\theta) p(\theta) / p(D) \\ &= p(D|\theta) p(\theta) / \sum_{\theta_c} p(D|\theta_c) p(\theta_c) \end{aligned}$$

Example: Estimating bias of a coin

Possible data values:

$D=1$ for "heads" and $D=0$ for "tails"

Likelihood function:

$p(D=1|\theta) = \theta$ and $p(D=0|\theta) = 1-\theta$
hence $p(D|\theta) = \theta^D(1-\theta)^{1-D}$

Prior on θ :

θ can be any value in interval $[0,1]$,
but for illustration we'll restrict θ to
three possible values with these prior
probabilities:

$$\begin{aligned} p(\theta=.25) &= .25 \\ p(\theta=.50) &= .50 \\ p(\theta=.75) &= .25 \end{aligned}$$



Example: Estimating bias of a coin

Possible data values:

$D=1$ for "heads" and $D=0$ for "tails"

Outcomes are nominal values, but re-coded as 1 and 0 for convenience in likelihood function.

Likelihood function:

$$p(D=1|\theta) = \theta \text{ and } p(D=0|\theta) = 1-\theta \\ \text{hence } p(D|\theta) = \theta^D(1-\theta)^{1-D}$$

Prior on θ :

θ can be any value in interval $[0,1]$, but for illustration we'll restrict θ to three possible values with these prior probabilities:

$$p(\theta = .25) = .25 \\ p(\theta = .50) = .50 \\ p(\theta = .75) = .25$$

Example: Estimating bias of a coin

Possible data values:

$D=1$ for "heads" and $D=0$ for "tails"

Likelihood function:

$$p(D=1|\theta) = \theta \text{ and } p(D=0|\theta) = 1-\theta \\ \text{hence } p(D|\theta) = \theta^D(1-\theta)^{1-D}$$

Probability of data value is a function of the parameter θ .
 D is an observed constant.

Prior on θ :

θ can be any value in interval $[0,1]$, but for illustration we'll restrict θ to three possible values with these prior probabilities:

$$p(\theta = .25) = .25 \\ p(\theta = .50) = .50 \\ p(\theta = .75) = .25$$

Bayesian Data Analysis

The role of data is to re-allocate credibility:

- 1. Define a meaningful descriptive model.**
- 2. Establish prior credibility regarding parameter values in the model. The prior credibility must be acceptable to a skeptical scientific audience.**
- 3. Collect data.**
- 4. Use Bayes' rule to re-allocate credibility to parameter values that are most consistent with the data.**

Model for bias of coin is simplistic, but it *is* a model, and the parameter carries the meaning of the model.

© John K. Kruschke, 2013

34

Example: Estimating bias of a coin

Possible data values:

D=1 for "heads" and D=0 for "tails"

Likelihood function:

$$p(D=1|\theta) = \theta \text{ and } p(D=0|\theta) = 1-\theta$$

hence $p(D|\theta) = \theta^D(1-\theta)^{1-D}$

Prior on θ :

θ can be any value in interval [0,1], but for illustration we'll restrict θ to three possible values with these prior probabilities:

$$\begin{aligned} p(\theta=.25) &= .25 \\ p(\theta=.50) &= .50 \\ p(\theta=.75) &= .25 \end{aligned}$$

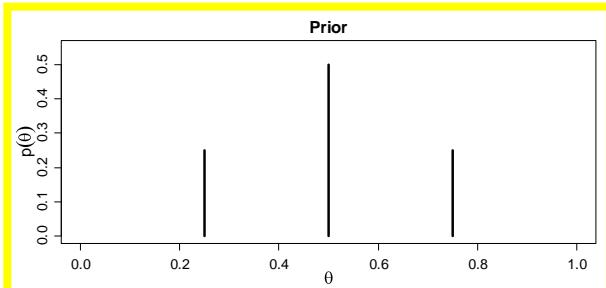
$p(\theta)$ can be thought of as the probability that a mint makes a coin that has bias θ .

Or, $p(\theta)$ can be thought of as the credibility that the coin has bias θ .

© John K. Kruschke, 2013

35

Example: Estimating bias of a coin

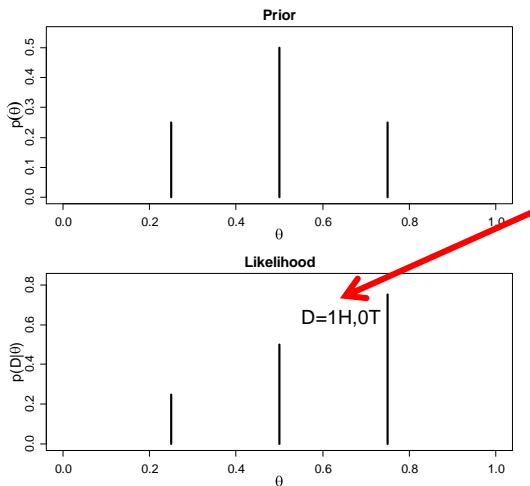


Prior on θ :

θ can be any value in interval $[0,1]$, but for illustration we'll restrict θ to three possible values with these prior probabilities:

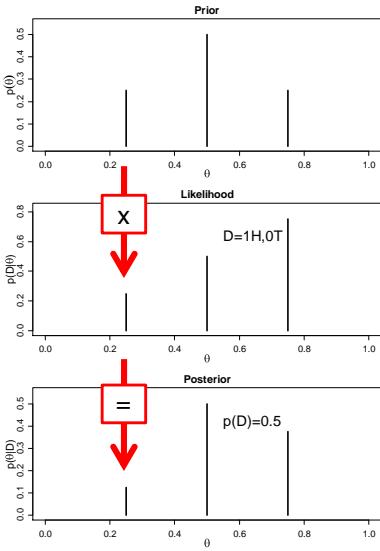
$$\begin{aligned} p(\theta = .25) &= .25 \\ p(\theta = .50) &= .50 \\ p(\theta = .75) &= .25 \end{aligned}$$

Example: Estimating bias of a coin



Flip the coin once
and get "heads":
Data contain 1
Heads and 0 Tails.

Example: Estimating bias of a coin

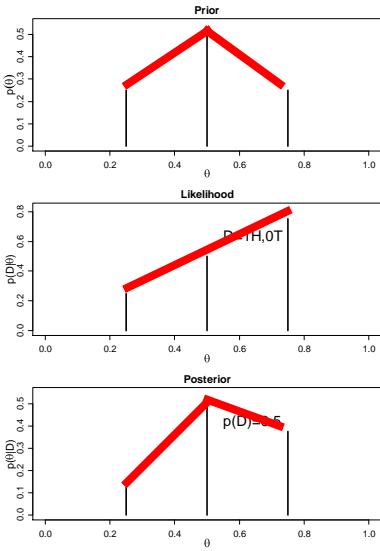


Posterior at θ is
prior, $p(\theta)$,
times
likelihood, $p(D|\theta)$,
normalized by
 $p(D) = \sum_{\theta} p(D|\theta)p(\theta)$.

© John K. Kruschke, 2013

38

Example: Estimating bias of a coin

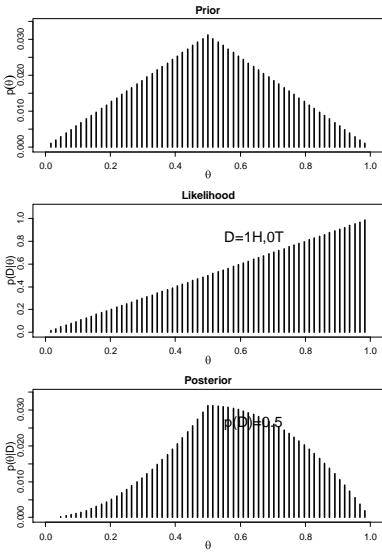


Posterior is a
compromise between
prior and likelihood.
Posterior shows re-
allocation of beliefs
across candidate
parameter values.

© John K. Kruschke, 2013

39

Example: Estimating bias of a coin

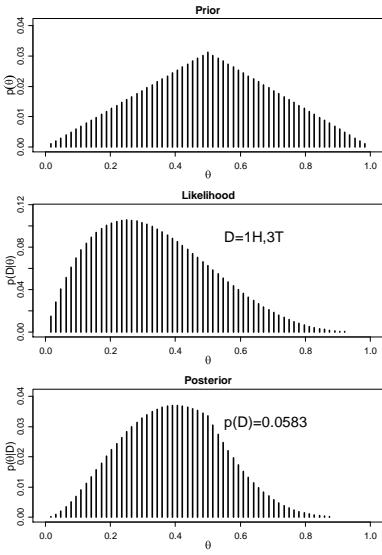


Prior has more candidate values of theta.

© John K. Kruschke, 2013

40

Example: Estimating bias of a coin

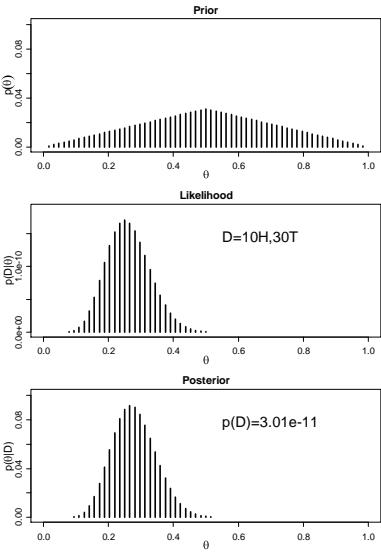


25% heads in small sample.
Strong influence of prior.

© John K. Kruschke, 2013

41

Example: Estimating bias of a coin

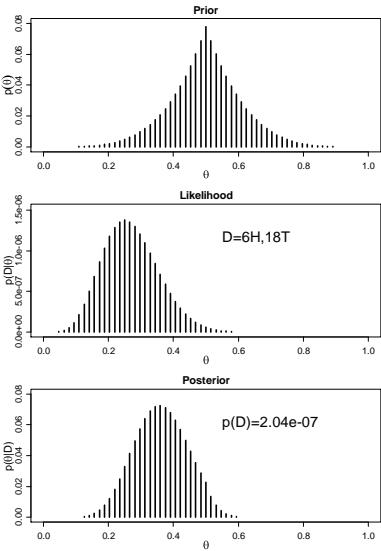


25% heads in larger sample.
Prior is overwhelmed by data.

© John K. Kruschke, 2013

42

Example: Estimating bias of a coin

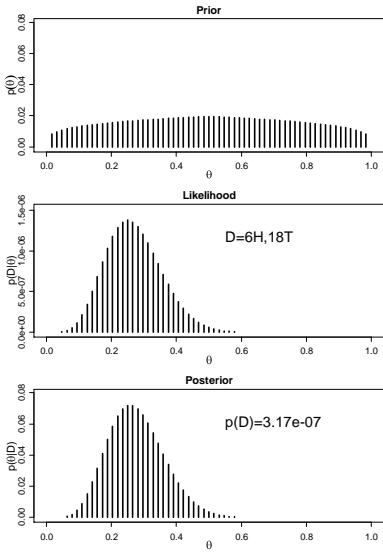


Strongly informed prior can have influence on posterior for modest amounts of data.

© John K. Kruschke, 2013

43

Example: Estimating bias of a coin

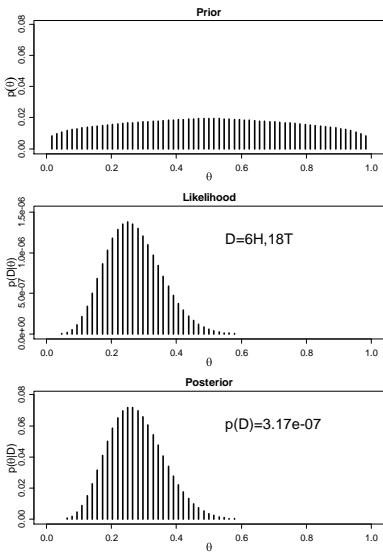


Vague prior
has little influence on
posterior.

© John K. Kruschke, 2013

44

Example: Estimating bias of a coin



Vague prior
has little influence on
posterior.

*Prior is chosen w.r.t. a
skeptical scientific
audience. Prior is overt,
not capricious. Prior can
be strongly informed,
mildly informed, or vague
depending on actual prior
knowledge and audience.*

© John K. Kruschke, 2013

45

Installation: See Blog Entry

<http://doingbayesiandataanalysis.blogspot.com/2012/01/complete-steps-for-installing-software.html>

The screenshot shows a blog post titled "Complete steps for installing software and programs". The post contains five numbered steps for installing JAGS and R. Step 1: Install the general programming language R. Step 2: Install the Bayesian sampling program JAGS. Step 3: Install the package that lets R talk to JAGS. Step 4: Get the programs used in the book. Step 5: Install the R editor RStudio.

Tips for running the programs:

- Many of the programs call others when running, so
- put all the programs together in the same folder and
- be sure that it has that folder as its working directory. Set the working directory in RStudio from the menu: Tools -> Set Working Dir. Or, if you're using R's built-in editor, set the working directory in R's command console from the menu: File -> Change dir...
- Mac users: Having trouble with graphics when running the programs? See [this blog post](#).

© John K. Kruschke, 2013

46

Using R

Help in R:

`help("keyword")`

If html help isn't working, try

`help("keyword", h="text")`

Handy list of R commands:

<http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

© John K. Kruschke, 2013

47

Running a program in R

Get the programs from

<http://www.indiana.edu/~kruschke/DoingBayesianDataAnalysis/>
and
<http://www.indiana.edu/~kruschke/BEST/BEST.zip>

Change directory to the folder in which you have your R programs saved:

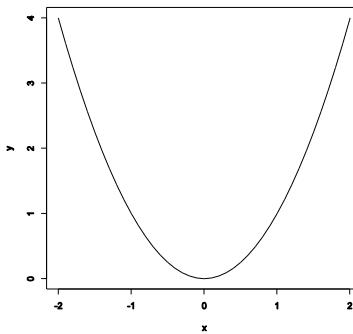
File → Change Dir...

Open the program SimpleGraph.R:

File → Open Script...

SimpleGraph.R

```
x = seq( from = -2 , to = 2 , by = 0.1 )      # Specify vector of x values.
y = x^2                                         # Specify corresponding y values.
plot( x , y , type = "l" )                      # Make a graph of the x,y points.
dev.copy2eps( file = "SimpleGraph.eps" )        # Save the plot to an EPS file.
```



File → Open Script... BayesUpdate.R

```
# Theta is the vector of candidate values for the parameter theta.
# nThetaVals is the number of candidate theta values.
# To produce the examples in the book, set nThetaVals to either 3 or 63.
nThetaVals = 3
# Now make the vector of theta values:
Theta = seq( from = 1/(nThetaVals+1) , to = nThetaVals/(nThetaVals+1) ,
            by = 1/(nThetaVals+1) )

# pTheta is the vector of prior probabilities on the theta values.
pTheta = pmin( Theta , 1-Theta ) # Makes a triangular belief distribution.
pTheta = pTheta / sum(pTheta) # Makes sure that beliefs sum to 1.

# Specify the data. To produce the examples in the book, use either
# Data = c(1,1,1,0,0,0,0,0,0,0) or Data = c(1,0,0,0,0,0,0,0,0,0).
Data = c(1,1,1,0,0,0,0,0,0,0)
nHeads = sum( Data == 1 )
nTails = sum( Data == 0 )

# Compute the likelihood of the data for each value of theta:
pDataGivenTheta = Theta^nHeads * (1-Theta)^nTails

# Compute the posterior:
pPosterior = pDataGivenTheta * pTheta / pData # This is Bayes' rule!

# Plot the results.
windows(7,10) # create window of specified width,height inches.
ylim(0,1) # Create axes with y-axis from 0 to 1.
par(mar=c(1,3,1,0)) # number of margin lines: bottom,left,top,right
par(mar=c(1,3,1,0)) # which margin lines to use for labels
par(mar=c(0.5, 0.5, 0.3, 0.1)) # margin size in inches: bottom,left,top,right

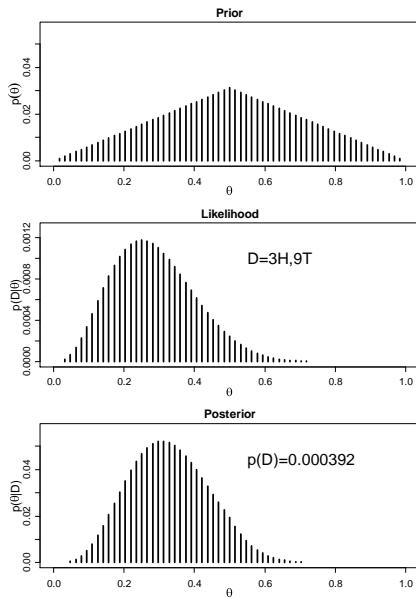
# Plot the prior:
plot(Theta , pTheta , type="h" , lwd=3 , main="Prior" ,
      xlab="theta" , ylab="p(theta)" ,
      ylim=0:1,max(pThetaGivenTheta) , ylab=deQuote(p(theta)) ,
      main=deQuote(p(theta))) , xaxs="i" , cex.lab=1.5 , cex.main=1.5)

# Plot the likelihood:
plot(Theta , pDataGivenTheta , type="h" , lwd=3 , main="Likelihood" ,
      xlab="theta" , ylab="p(D|theta)" ,
      ylim=0:1,max(pThetaGivenTheta) , ylab=deQuote(paste("p(D|",theta,")")) ,
      main=deQuote(paste("D=",nHeads,"/",nTails,"; p(D)=",p(D))) ,
      text=.55 , xlab="theta" , ylab="p(D|theta)" , xaxs="i" ,
      cex.lab=1.5 , cex.main=1.5)

# Plot the posterior:
plot(Theta , pPosterior , type="h" , lwd=3 , main="Posterior" ,
      xlab="theta" , ylab="p(theta|D)" ,
      ylim=0:1,max(pThetaGivenTheta) , ylab=deQuote(paste("p(",theta,"|D)") ,
      main=deQuote(paste("D=",nHeads,"/",nTails,"; p(D)=",p(D)))) ,
      text=.55 , xlab="theta" , ylab="p(theta|D)" , xaxs="i" ,
      cex.lab=1.5 , cex.main=1.5)

# Save the plot as an EPS file.
# windows(7,10) # create window of specified width,height inches.
if (nThetaVals == 3) {modelType = "simpleModel" }
if (nThetaVals == 63) {modelType = "complexModel" }
if (nHeads == 1 & nTails == 11) {datatype = "complexData" }
if (nHeads == 11 & nTails == 1) {datatype = "complexData" }
filename = paste("BayesUpdate",modelType,".",datatype,".eps",sep="")
pD = pPosterior[nHeads+1]
text(.55 , .55,max(pThetaGivenTheta) , cex=2 ,
      bgquote("p(D)=", .5*(max(pThetaGivenTheta))) , adj=c(0,.5))

# Try help("dev.copyEPS") for info about saving graphs in other file formats.
dev.copyEPS(file=filename)
```



© John K. Kruschke, 2013

50

File → Open Script... BayesUpdate.R

```
# Theta is the vector of candidate values for the parameter theta.
# nThetaVals is the number of candidate theta values.
# To produce the examples in the book, set nThetaVals to either 3 or 63.
nThetaVals = 3
# Now make the vector of theta values:
Theta = seq( from = 1/(nThetaVals+1) , to = nThetaVals/(nThetaVals+1) ,
            by = 1/(nThetaVals+1) )
Theta is a vector of values.

# pTheta is the vector of prior probabilities on the theta values.
pTheta = pmin( Theta , 1-Theta ) # Makes a triangular belief distribution.
pTheta = pTheta / sum(pTheta) # Makes sure that beliefs sum to 1.
pmin is component-wise minimum.

# Specify the data. To produce the examples in the book, use either
# Data = c(1,1,1,0,0,0,0,0,0,0) or Data = c(1,0,0,0,0,0,0,0,0,0).
Data = c(1,1,1,0,0,0,0,0,0,0)
nHeads = sum( Data == 1 )
nTails = sum( Data == 0 )

# Compute the likelihood of the data for each value of theta:
pDataGivenTheta = Theta^nHeads * (1-Theta)^nTails

# Compute the posterior:
pData = sum( pDataGivenTheta * pTheta )
pPosterior = pDataGivenTheta * pTheta / pData # This is Bayes' rule!

# Plot the results.
windows(7,10) # create window of specified width,height inches.
```

© John K. Kruschke, 2013

51

File → Open Script... BayesUpdate.R

```

# Theta is the vector of candidate values for the parameter theta.
# nThetaVals is the number of candidate theta values.
# To produce the examples in the book, set nThetaVals to either 3 or 63.
nThetaVals = 3
# Now make the vector of theta values:
Theta = seq( from = 1/(nThetaVals+1) , to = nThetaVals/(nThetaVals+1) ,
             by = 1/(nThetaVals+1) )

# pTheta is the vector of prior probabilities on the theta values.
pTheta = pmin( Theta , 1-Theta ) # Makes a triangular belief distribution.
pTheta = pTheta / sum( pTheta ) # Makes sure that beliefs sum to 1.

# Specify the data. To produce the examples in the book, use either
# Data = c(1,1,1,0,0,0,0,0,0,0,0) or Data = c(1,0,0,0,0,0,0,0,0,0,0).
Data = c(1,1,1,0,0,0,0,0,0,0,0)
nHeads = sum( Data == 1 )
nTails = sum( Data == 0 ) power is applied to each component of the vector

# Compute the likelihood of the data for each value of theta:
pDataGivenTheta = Theta^nHeads * (1-Theta)^nTails multiplication is component by component

# Compute the posterior:
pData = sum( pDataGivenTheta * pTheta )
pThetaGivenData = pDataGivenTheta * pTheta / pData # This is Bayes' rule!

# Plot the results.
windows(7,10) # create window of specified width,height inches.

```

© John K. Kruschke, 2013

52

Agenda

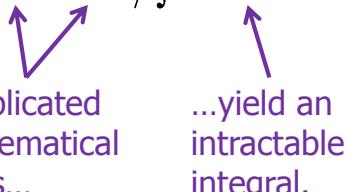
- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

© John K. Kruschke, 2013

53

Why Bayesian inference can be difficult

$$\begin{aligned}
 p(\theta | D) &= p(D | \theta)p(\theta) / p(D) \\
 &= p(D | \theta)p(\theta) / \sum_c p(D | \theta_c)p(\theta_c) \quad \text{for discrete } \theta \\
 &= p(D | \theta)p(\theta) / \int d\theta p(D | \theta)p(\theta) \quad \text{for continuous } \theta
 \end{aligned}$$



Complicated mathematical forms... ...yield an intractable integral.

Moreover,
 grid approximation
 will not work for
 models with many
 parameters.

Approximating the posterior by a random sample

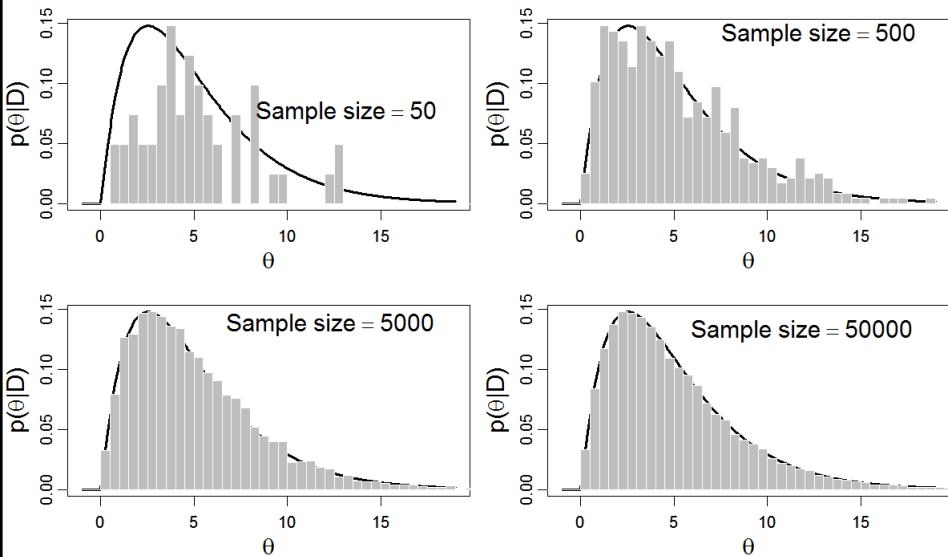
Problem:

Analytical mathematical derivation is unknown, and grid approximation is intractable.

Solution:

- Approximate the posterior by a large random sample of representative values of the parameter.
- This approach works because we can generate random values from the posterior without solving the integral for $p(D)$.

Approximating a distribution by a random sample

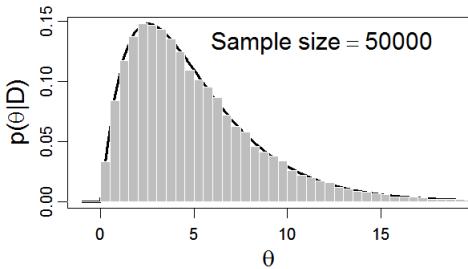


© John K. Kruschke, 2013

57

Approximating a distribution by a random sample

Any aspect of the underlying continuous posterior distribution can be approximated from the sample of parameter values: Mean, variance, percentiles, etc.



© John K. Kruschke, 2013

58

How to generate a random sample

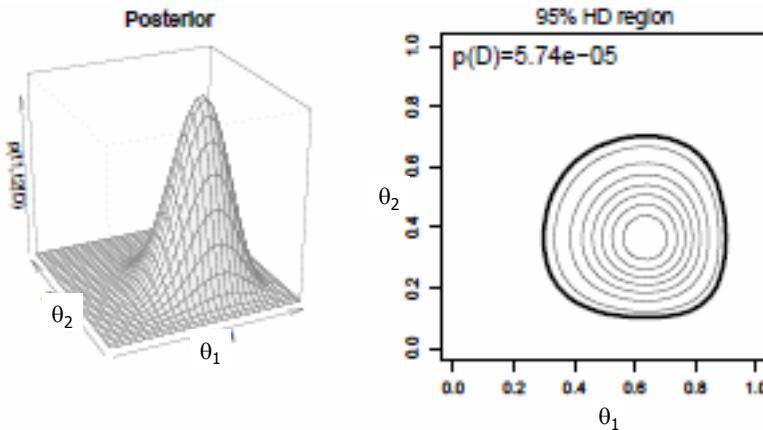
Some distributions have known, efficient random-sample generators:
 Uniform, normal, exponential, beta, gamma, etc.

Use the known random sample generators to indirectly sample from a complicated posterior distribution. There are several ways, of which we'll consider two:

- *Gibbs Sampling*
- *Metropolis Algorithm*

Example

Consider this posterior distribution on two parameters, θ_1 and θ_2 . Goal is to generate a random sample without computing the integral $p(D)$.



Gibbs Sampling

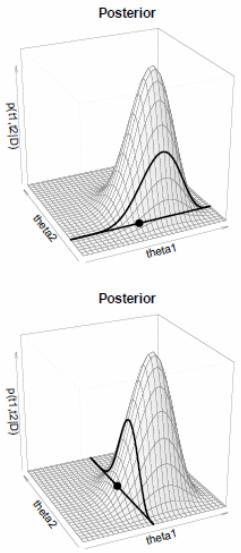
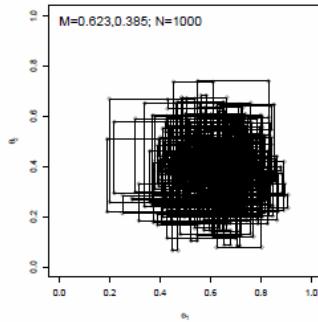


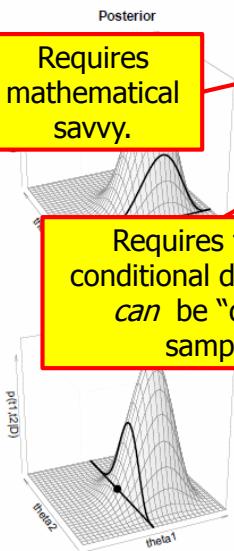
Figure 8.4: Two steps in a Gibbs sampling. The top panel shows a random generation of a value for θ_1 , conditional on a value for θ_2 . The heavy lines show a slice through the posterior at the conditional value of θ_2 , and the large dot shows a random value of θ_1 sampled from the conditional density. The bottom panel shows a random generation of a value for θ_2 , conditional on the value for θ_1 determined by the previous step. The heavy lines show a slice through the posterior at the conditional value of θ_1 , and the large dot shows a random value of θ_2 sampled from the conditional density.



© John K. Kruschke, 2013

61

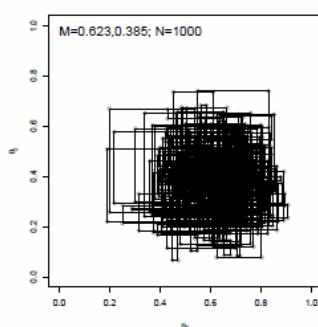
Gibbs Sampling



Requires that the conditional distributions *can* be “directly” sampled.

1. Derive conditional distribution for each parameter: $p(\theta_i | D, \{\theta_{j \neq i}\})$
2. Sample directly from each conditional distribution.

In a Gibbs panel shows a random generation of a value for θ_1 , conditional on a value for θ_2 . The heavy lines show a slice through the posterior at the conditional value of θ_2 , and the large dot shows a random value of θ_1 sampled from the conditional density. The bottom panel shows a random generation of a value for θ_2 , conditional on the value for θ_1 determined by the previous step. The heavy lines show a slice through the posterior at the conditional value of θ_1 , and the large dot shows a random value of θ_2 sampled from the conditional density.

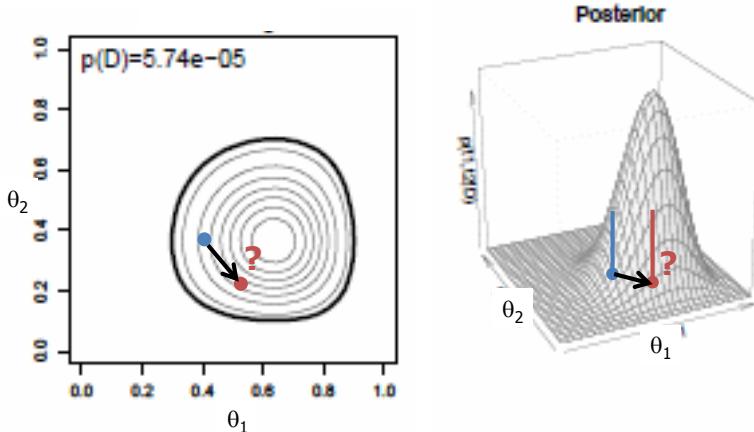


© John K. Kruschke, 2013

62

Metropolis Algorithm

1. Propose a new parameter position, θ_{pro} , directly sampled from a known distribution such as a normal.
2. Decide whether to accept the proposed position. Decision rule only needs to evaluate $p(D|\theta_{\text{pro}})$ and $p(\theta_{\text{pro}})$, not $p(D)$.



© John K. Kruschke, 2013

63

Metropolis Algorithm

1. Propose a new parameter position, θ_{pro} , directly sampled from a known distribution such as a normal.
2. Decide whether to accept the proposed position. Decision rule only needs to evaluate $p(D|\theta_{\text{pro}})$ and $p(\theta_{\text{pro}})$, not $p(D)$.

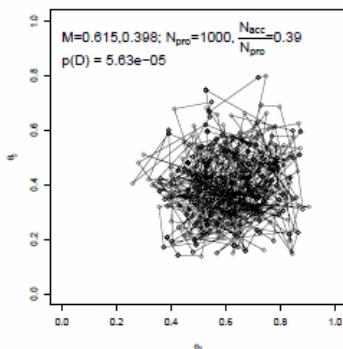


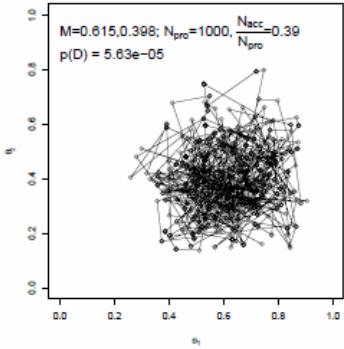
Figure 8.3: Metropolis algorithm applied to the prior and likelihood shown in Figure 8.1, p. 131. Compare this scatter plot with the lower right contour plot of Figure 8.1. N_{pro} is the number of proposed jumps (after burn in), and N_{acc} is the number of accepted proposals. R code is in Section 8.8.2 (BernTwoMetropolis.R).

© John K. Kruschke, 2013

64

Metropolis Algorithm

1. Propose a new parameter position, θ_{pro} , directly sampled from a known distribution such as a normal.
2. Decide whether to accept the proposed position. Decision rule only needs to evaluate $p(D|\theta_{\text{pro}})$ and $p(\theta_{\text{pro}})$, not $p(D)$.



If the proposal distribution does not resemble the posterior distribution (e.g., proposal is too wide or too narrow), then many proposals are rejected and the sample gets very “clumpy”.

Gibbs and Metropolis: Both “MCMC” Sampling

Both are “Markov chain Monte Carlo” (MCMC) methods:

From current point (\leftarrow Markov chain), randomly generate a new point (\leftarrow Monte Carlo).

Doing it with JAGS

“JAGS” = Just Another Gibbs Sampler
but other sampling methods are incorporated.

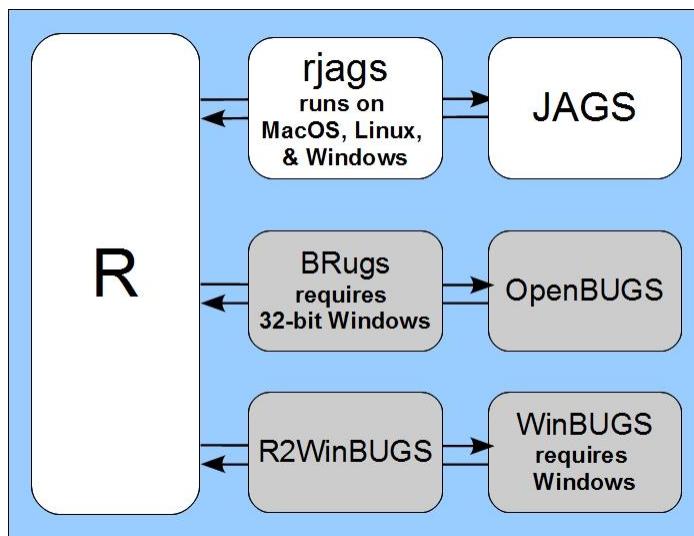


JAGS makes it easy. You specify only the

- prior function
- likelihood function

and JAGS does the rest! You do no math, no selection of sampling methods.

JAGS and BUGS



Installation: See Blog Entry

<http://doingbayesiandataanalysis.blogspot.com/2012/01/complete-steps-for-installing-software.html>

The screenshot shows a blog post titled "Complete steps for installing software and programs". The post provides instructions for installing R, JAGS, and R packages related to the book. It includes a link to a zip file containing all programs and a note about using JAGS instead of BUGS. The sidebar features a book cover for "Doing Bayesian Data Analysis" by John Kruschke, a search bar, and a blog archive.

Saturday, January 28, 2012

Complete steps for installing software and programs

To use the programs, there are five basic installation steps:

1. Install the general programming language R. Go to the [R web site](#) and install the latest version of R appropriate for your computer (Windows, Mac, Linux).
2. Install the Bayesian sampling program JAGS. Go to the [JAGS web site](#) and install the latest version of JAGS appropriate for your computer (Windows, Mac, Linux). Caution to Mac users: If you find that R is having trouble working with JAGS, it might be because `zjags` (in the next step) is compiled for use with JAGS 2.* instead of JAGS 3.*. If you have this problem, uninstall JAGS 3.* and install JAGS 2.* instead.
3. Install the package that lets R talk to JAGS. Invoke R. At the command line, type
`install.packages("rjags")`
 You may be prompted to select an internet archive to get the package from; select a site geographically near you.
4. Get the programs used in the book. For a list of individually downloadable programs, [click here](#). For a zip file (named `ProgramsDoingBayesianDataAnalysis.zip`) that contains all the programs in a single file, [click here](#). Be sure to unzip (extract) the programs from the zip file. Programs are updated occasionally. Click on the first link, above, and sort the list by date modified to see if you have the most recent versions. I now recommend using JAGS instead of BUGS. All the programs now have JAGS versions in addition to the original BUGS versions. Any program that had "BUGS" or "BUGS" in its file name now has a JAGS equivalent with "JAGS" in its file name. More info is available at [this blog post](#).
5. Install the R editor RStudio. R comes with its own built-in editor, but it is not very useful for dealing with long programs. Go to the [RStudio web site](#) and install the latest version of RStudio appropriate for your computer (Windows, Mac, Linux).

Tips for running the programs:
 Many of the programs call others when running, so
 • put all the programs together in the same folder and
 • be sure that R has that folder as its working directory. Set the working directory in RStudio from the menu: Tools -> Set Working Dir. Or, if you're using R's built-in editor, set the working directory in R's command console from the menu: File -> Change dir...
 Mac users: Having trouble with graphics when running the programs? See [this blog post](#).

© John K. Kruschke, 2013

69

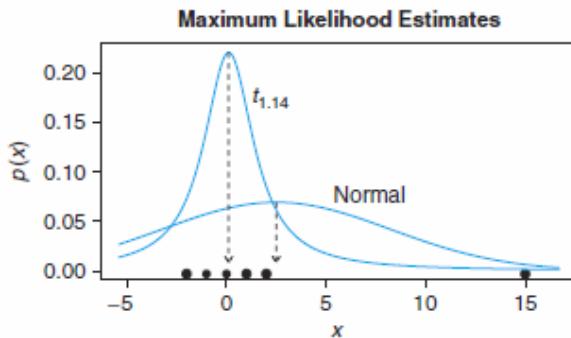
Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

© John K. Kruschke, 2013

70

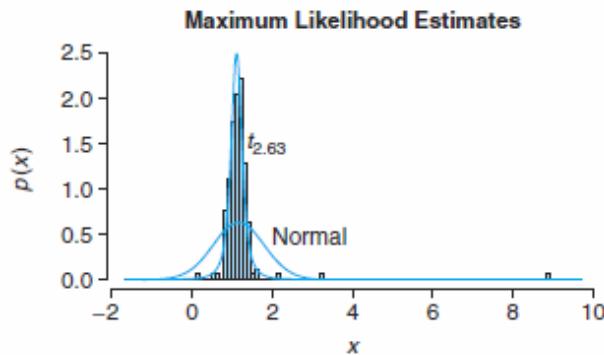
Robust estimation with outliers



Normal is pulled by outliers, but t distribution is not.

t distribution is used here as a description of data,
NOT as a sampling distribution for p values!

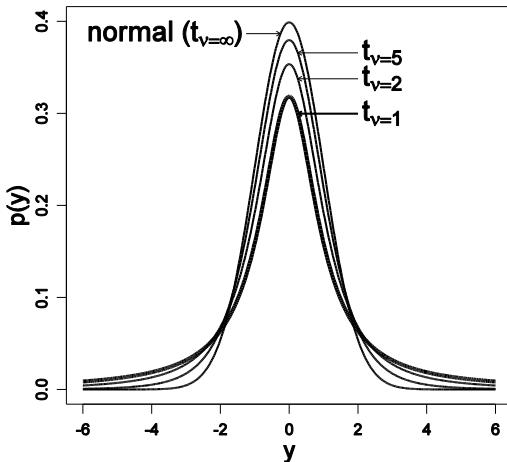
Robust estimation with outliers



Normal is expanded by outliers, but t distribution is not.

t distribution is used here as a description of data,
NOT as a sampling distribution for p values!

Robust estimation with outliers

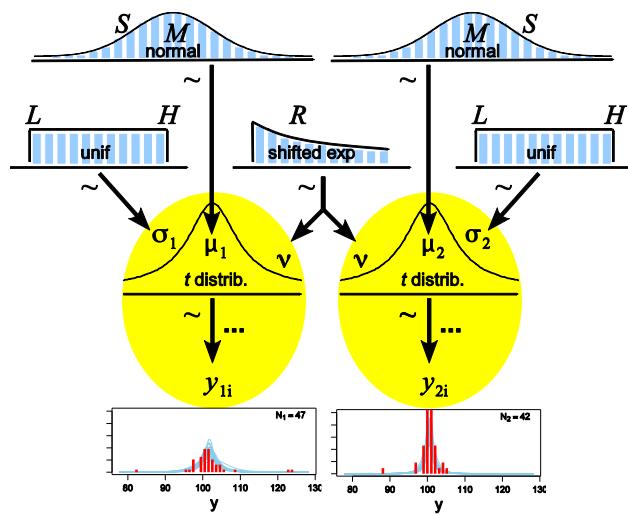


The t distribution has normality controlled by the parameter v .

© John K. Kruschke, 2013

73

Robust Bayesian estimation for comparing two groups

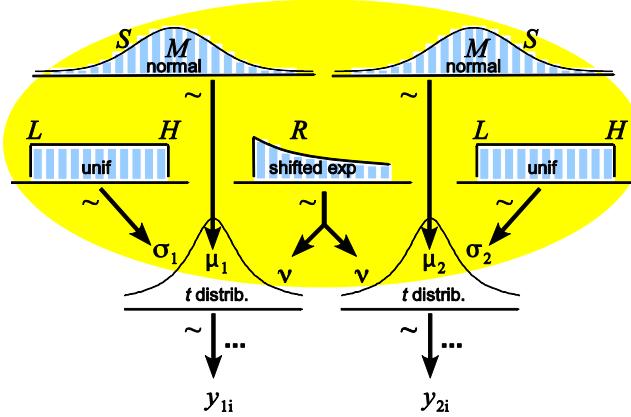


Step 1:
The data from each group are described by t distributions, using five parameters altogether.

© John K. Kruschke, 2013

74

Robust Bayesian estimation for comparing two groups

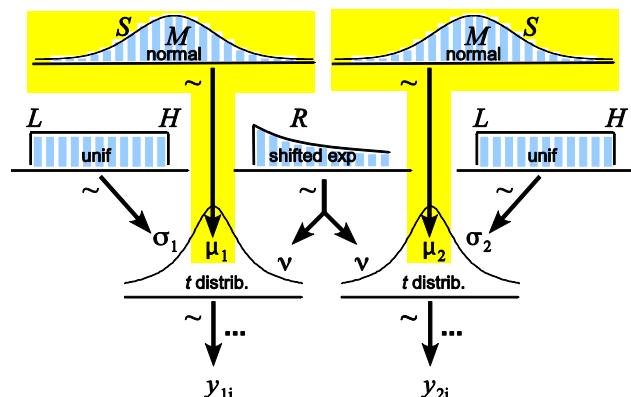


Step 2: Specify
the prior.

© John K. Kruschke, 2013

75

Robust Bayesian estimation for comparing two groups

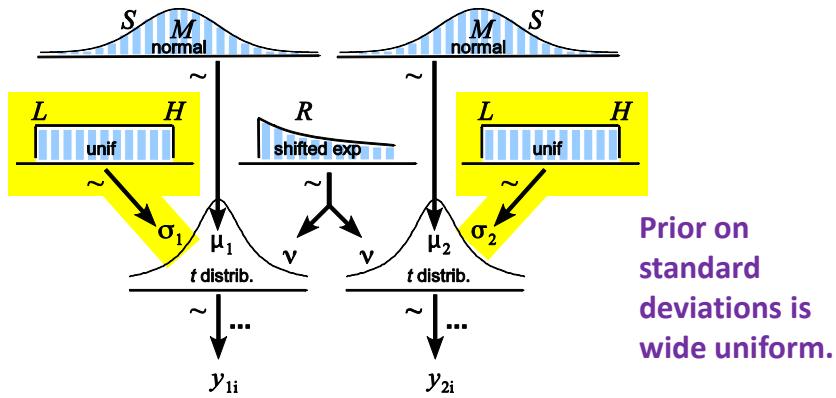


Prior on means
is wide normal.

© John K. Kruschke, 2013

76

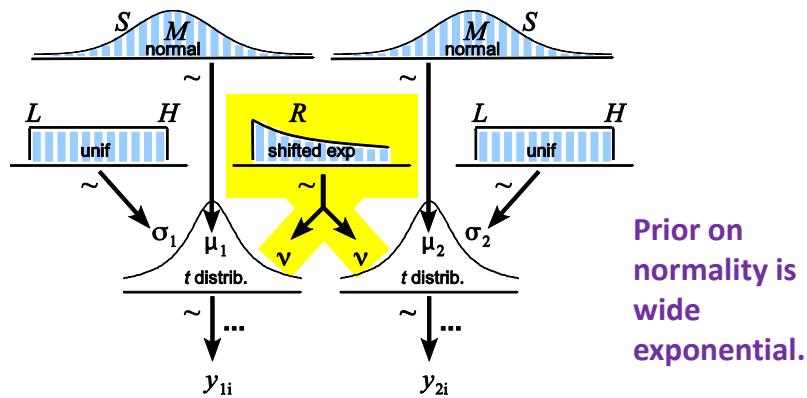
Robust Bayesian estimation for comparing two groups



© John K. Kruschke, 2013

77

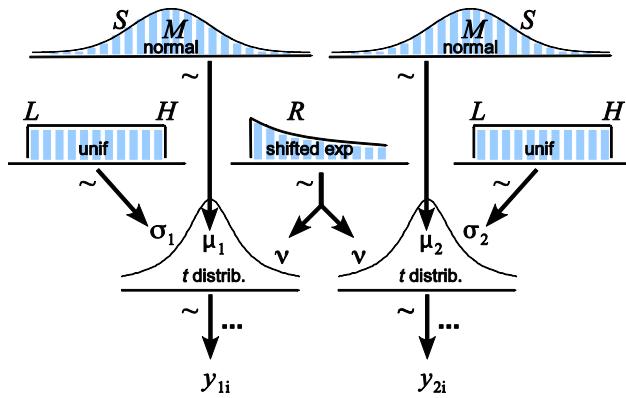
Robust Bayesian estimation for comparing two groups



© John K. Kruschke, 2013

78

Robust Bayesian estimation for comparing two groups

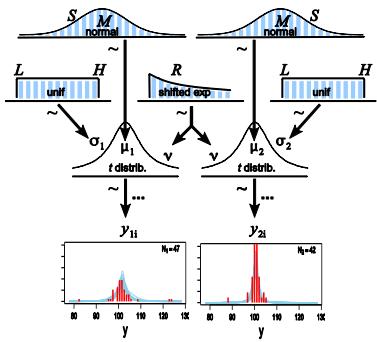


Parameter distributions will be represented by histograms: A huge number of representative parameter values.

© John K. Kruschke, 2013

70

Step 3: Collect Data.

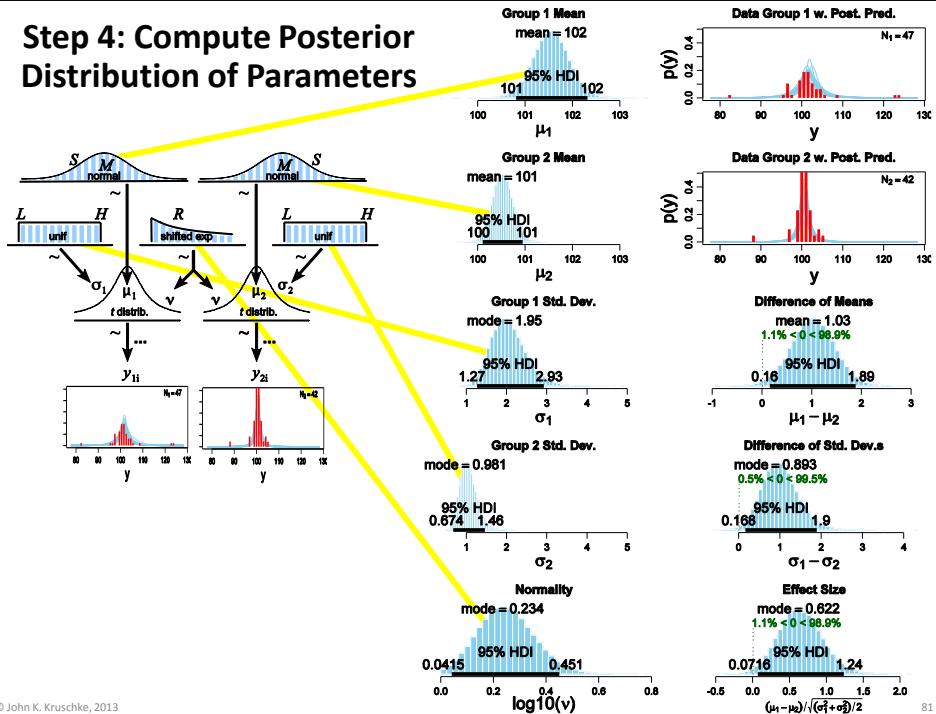


One fixed data set,
shown as red
histograms.

© John K. Kruschke, 2013

80

Step 4: Compute Posterior Distribution of Parameters



© John K. Kruschke, 2013

81

Computer Software:

Packaged for easy use!
Underlying program is never seen.

```
source("BEST.R") # load the program

# Specify data as vectors (replace with your own data):
y1 = c(101,100,102,104,102,97,105,105,98,101,...,101)
y2 = c(99,101,100,101,102,100,97,101,104,101,...,99)

# Run the Bayesian analysis:
mcmcChain = BESTmcmc( y1 , y2 )

# Plot the results of the Bayesian analysis:
BESTplot( y1 , y2 , mcmcChain )
```

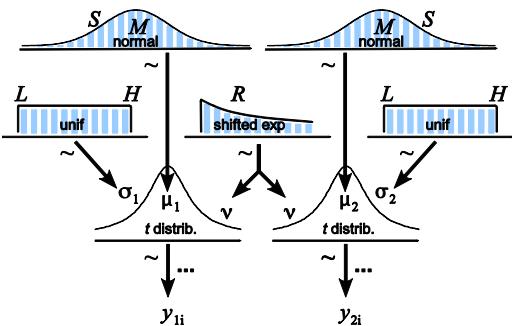
© John K. Kruschke, 2013

82

Robust Bayesian estimation for comparing two groups

Download the programs from
<http://www.indiana.edu/~kruschke/BEST/BEST.zip>

Now for a look
under the hood



<http://www.autonationconnect.com/2010/07/backseat-mechanic-under-the-hood/>

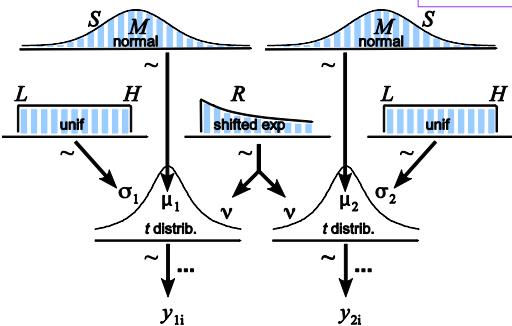
© John K. Kruschke, 2013

83

Robust Bayesian estimation for comparing two groups

Program BEST.R: JAGS model specification.

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29)
}
```



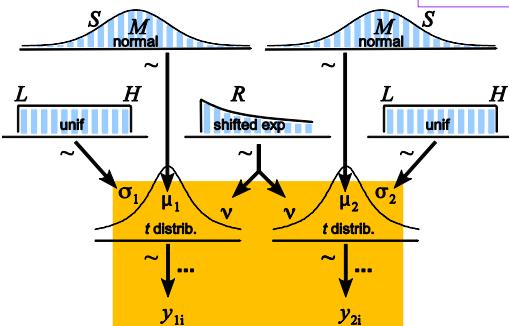
© John K. Kruschke, 2013

84

Robust Bayesian estimation for comparing two groups

Program BEST.R: JAGS model specification.

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29)
}
```



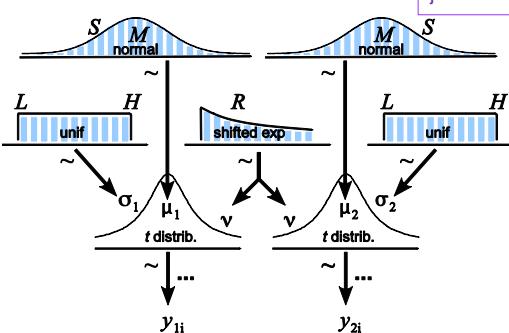
© John K. Kruschke, 2013

85

Robust Bayesian estimation for comparing two groups

Program BEST.R: JAGS model specification.

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29)
}
```



Nested indexing:
 $x[i]$ is the group (1 or 2)
of the i^{th} score.

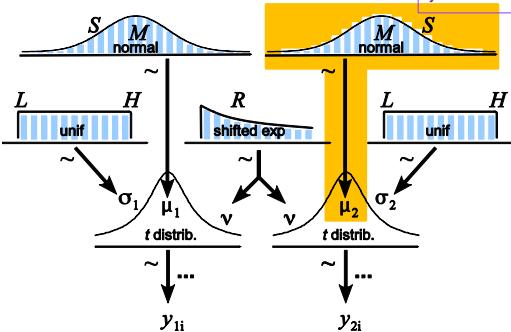
© John K. Kruschke, 2013

86

Robust Bayesian estimation for comparing two groups

Program BEST.R: JAGS model specification.

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29)
}
```



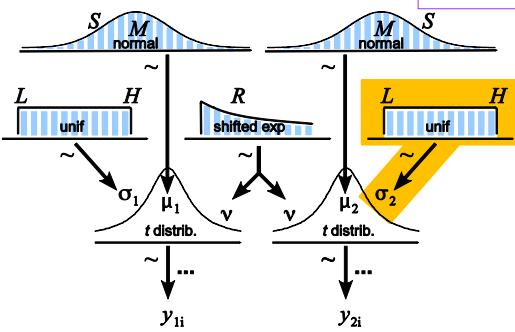
© John K. Kruschke, 2013

87

Robust Bayesian estimation for comparing two groups

Program BEST.R: JAGS model specification.

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29)
}
```



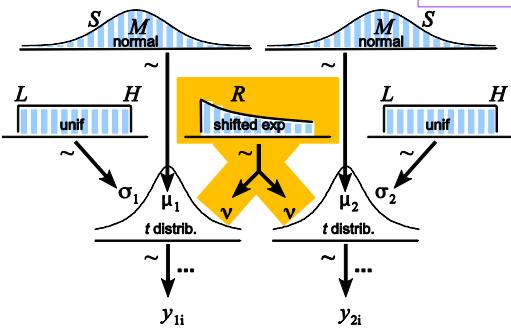
© John K. Kruschke, 2013

88

Robust Bayesian estimation for comparing two groups

Program BEST.R: JAGS model specification.

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29)
}
```



© John K. Kruschke, 2013

89

Programs in R + rjags + JAGS:

Five main sections in all programs:

1. **Specify model** (we just did this).
2. **Load data.**
3. **Initialize the MCMC chain.**
4. **Run the MCMC chain.**
5. **Examine the results.**

© John K. Kruschke, 2013

90

BEST.R

```

BESTmcmc = function( y1, y2, numSavedSteps=100000, thinSteps=1, showMCMC=FALSE) {
  # This function generates an MCMC sample from the posterior distribution.
  # Description of arguments:
  # showMCMC is a flag for displaying diagnostic graphs of the chains.
  #   If F (the default), no chain graphs are displayed. If T, they are.

  require(rjags)

  #-----
  # THE MODEL.
  modelString = "
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+
  nuMinusOne ~ dexp(1/29)
}
" # close quote for modelString
# Write out modelString to a text file
writeLines( modelString , con="BESTmodel.txt" )

#-----
# THE DATA.
# Load the data:
y = c( y1 , y2 ) # combine data into one vector
x = c( rep(1,length(y1)) , rep(2,length(y2)) ) # create group membership code
Ntotal = length(y)
# Specify the data in a list, for later shipment to JAGS:
dataList = list(
  y = y ,
  x = x ,
  Ntotal = Ntotal .

```

© John K. Kruschke, 2013

91

BEST.R

```

BESTmcmc = function( y1, y2, numSavedSteps=100000, thinSteps=1, showMCMC=FALSE) {
  # This function generates an MCMC sample from the posterior distribution.
  # Description of arguments:
  # showMCMC is a flag for displaying diagnostic graphs of the chains.
  #   If F (the default), no chain graphs are displayed. If T, they are.

  require(rjags)

  #-----
  # THE MODEL.
  modelString = "
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt( mu[x[i]] , tau[x[i]] , nu )
  }
  for ( j in 1:2 ) {
    mu[j] ~ dnorm( muM , muP )
    tau[j] <- 1/pow( sigma[j] , 2 )
    sigma[j] ~ dunif( sigmaLow , sigmaHigh )
  }
  nu <- nuMinusOne+
  nuMinusOne ~ dexp(1/29)
}
" # close quote for modelString
# Write out modelString to a text file
writeLines( modelString , con="BESTmodel.txt" )

#-----
# THE DATA.
# Load the data:
y = c( y1 , y2 ) # combine data into one vector
x = c( rep(1,length(y1)) , rep(2,length(y2)) ) # create group membership code
Ntotal = length(y)
# Specify the data in a list, for later shipment to JAGS:
dataList = list(
  y = y ,
  x = x ,
  Ntotal = Ntotal .

```

© John K. Kruschke, 2013

92

BEST.R

```

}
nu <- nuMinusOne+1
nuMinusOne ~ dexp(1/29)
}
# close quote for modelString
# Write out modelString to a text file
writeLines( modelString , con="BESTmodel.txt" )

#-----#
# THE DATA.
# Load the data:
y = c( y1 , y2 ) # combine data into one vector
x = c( rep(1,length(y1)) , rep(2,length(y2)) ) # create group membership code
Ntotal = length(y)
# Specify the data in a list, for later shipment to JAGS:
dataList = list(
  y = y ,
  x = x ,
  Ntotal = Ntotal ,
  muM = mean(y) ,
  muP = 0.000001 * 1/sd(y)^2 ,
  sigmaLow = sd(y) / 1000 ,
  sigmaHigh = sd(y) * 1000
)

#-----#
# INITIALIZE THE CHAINS.
# Initial values of MCMC chains based on data:
mu = c( mean(y1) , mean(y2) )
sigma = c( sd(y1) , sd(y2) )
# Regarding initial values in next line: (1) sigma will tend to be too big if
# the data have outliers, and (2) nu starts at 5 as a moderate value. These
# initial values keep the burn-in period moderate.
initsList = list( mu = mu , sigma = sigma , nuMinusOne = 4 )

#-----#
# RUN THE CHAINS

parameters = c( "mu" , "sigma" , "nu" )      # The parameters to be monitored
adaptSteps = 500                                # Number of steps to "tune" the samplers
burnInSteps = 1000
nChains = 3
#chain = ceiling( ( nSavedChains * thinStep ) / nChains )

```

© John K. Kruschke, 2013

93

BEST.R

```

}
nu <- nuMinusOne+1
nuMinusOne ~ dexp(1/29)
}
# close quote for modelString
# Write out modelString to a text file
writeLines( modelString , con="BESTmodel.txt" )

#-----#
# THE DATA.
# Load the data:
y = c( y1 , y2 ) # combine data into one vector
x = c( rep(1,length(y1)) , rep(2,length(y2)) ) # create group membership code
Ntotal = length(y)
# Specify the data in a list, for later shipment to JAGS:
dataList = list(
  y = y ,
  x = x ,
  Ntotal = Ntotal ,
  muM = mean(y) ,
  muP = 0.000001 * 1/sd(y)^2 ,
  sigmaLow = sd(y) / 1000 ,
  sigmaHigh = sd(y) * 1000
)

#-----#
# INITIALIZE THE CHAINS.
# Initial values of MCMC chains based on data:
mu = c( mean(y1) , mean(y2) )
sigma = c( sd(y1) , sd(y2) )
# Regarding initial values in next line: (1) sigma will tend to be too big if
# the data have outliers, and (2) nu starts at 5 as a moderate value. These
# initial values keep the burn-in period moderate.
initsList = list( mu = mu , sigma = sigma , nuMinusOne = 4 )

#-----#
# RUN THE CHAINS

parameters = c( "mu" , "sigma" , "nu" )      # The parameters to be monitored
adaptSteps = 500                                # Number of steps to "tune" the samplers
burnInSteps = 1000
nChains = 3
#chain = ceiling( ( nSavedChains * thinStep ) / nChains )

```

© John K. Kruschke, 2013

94

```

# initial values keep the burn-in period moderate.
initsList = list( mu = mu , sigma = sigma , nuMinusOne = 4 )
BEST.R
#-----#
# RUN THE CHAINS

parameters = c( "mu" , "sigma" , "nu" )      # The parameters to be monitored
adaptSteps = 500                                # Number of steps to "tune" the samplers
burnInSteps = 1000
nChains = 3
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )
# Create, initialize, and adapt the model:
jagsModel = jags.model( "BESTmodel.txt" , data=dataList , inits=initsList ,
                       n.chains=nChains , n.adapt=adaptSteps )
# Burn-in:
cat( "Burning in the MCMC chain...\n" )
update( jagsModel , n.iter=burnInSteps )
# The saved MCMC chain:
cat( "Sampling final MCMC chain...\n" )
codaSamples = coda.samples( jagsModel , variable.names=parameters ,
                           n.iter=nIter , thin=thinSteps )
# resulting codaSamples object has these indices:
#   codaSamples[[ chainIdx ]][ stepIdx , paramIdx ]

#-----#
# EXAMINE THE RESULTS
if ( showMCMC ) {
  windows()
  autocorr.plot( codaSamples[[1]] , ask=FALSE )
}

# Convert coda-object codaSamples to matrix object for easier handling.
# But note that this concatenates the different chains into one long chain.
# Result is mcmcChain[ stepIdx , paramIdx ]
mcmcChain = as.matrix( codaSamples )
return( mcmcChain )

} # end function BESTmcmc

```

© John K. Kruschke, 2013

95

```

# initial values keep the burn-in period moderate.
initsList = list( mu = mu , sigma = sigma , nuMinusOne = 4 )
BEST.R
#-----#
# RUN THE CHAINS

parameters = c( "mu" , "sigma" , "nu" )      # The parameters to be monitored
adaptSteps = 500                                # Number of steps to "tune" the samplers
burnInSteps = 1000
nChains = 3
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )
# Create, initialize, and adapt the model:
jagsModel = jags.model( "BESTmodel.txt" , data=dataList , inits=initsList ,
                       n.chains=nChains , n.adapt=adaptSteps )
# Burn-in:
cat( "Burning in the MCMC chain...\n" )
update( jagsModel , n.iter=burnInSteps )
# The saved MCMC chain:
cat( "Sampling final MCMC chain...\n" )
codaSamples = coda.samples( jagsModel , variable.names=parameters ,
                           n.iter=nIter , thin=thinSteps )
# resulting codaSamples object has these indices:
#   codaSamples[[ chainIdx ]][ stepIdx , paramIdx ]

#-----#
# EXAMINE THE RESULTS
if ( showMCMC ) {
  windows()
  autocorr.plot( codaSamples[[1]] , ask=FALSE )
}

# Convert coda-object codaSamples to matrix object for easier handling.
# But note that this concatenates the different chains into one long chain.
# Result is mcmcChain[ stepIdx , paramIdx ]
mcmcChain = as.matrix( codaSamples )
return( mcmcChain )

} # end function BESTmcmc

```

© John K. Kruschke, 2013

96

Computer Software:

Packaged for easy use!
Underlying program is never seen.

```
source("BEST.R") # load the program

# Specify data as vectors (replace with your own data):
y1 = c(101,100,102,104,102,97,105,105,98,101,...,101)
y2 = c(99,101,100,101,102,100,97,101,104,101,...,99)

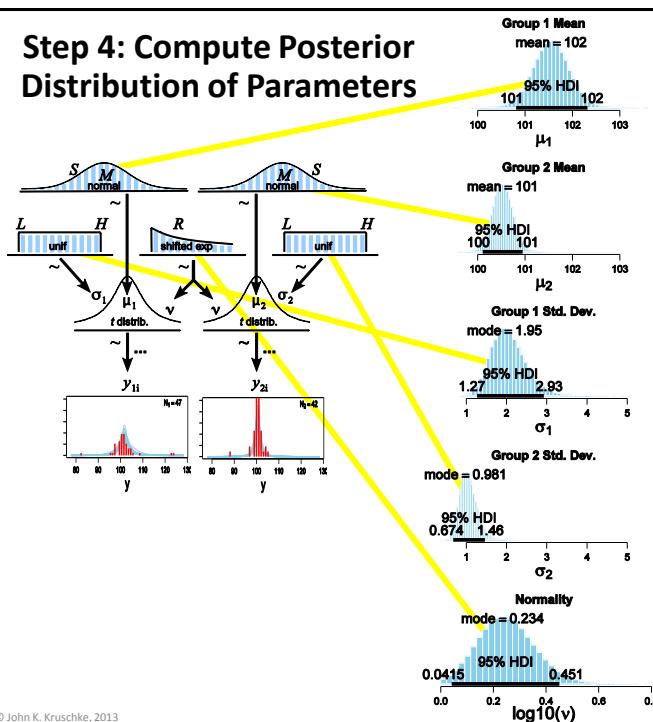
# Run the Bayesian analysis:
mcmcChain = BESTmcmc( y1 , y2 )

# Plot the results of the Bayesian analysis:
BESTplot( y1 , y2 , mcmcChain )
```

© John K. Kruschke, 2013

97

Step 4: Compute Posterior Distribution of Parameters



Important:
These are histograms of parameter values from the posterior distribution:
A huge number of combinations of μ_1 , μ_2 , σ_1 , σ_2 , v that are jointly credible given the data.

These are not data distributions, and not sampling distributions from a null hypothesis.

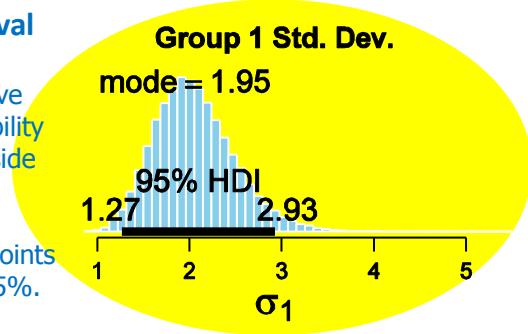
© John K. Kruschke, 2013

95% HDI: Highest density interval

Points within the HDI have higher credibility (probability density) than points outside the HDI.

The total probability of points within the 95% HDI is 95%.

Points outside the HDI may be deemed not credible.



© John K. Kruschke, 2013

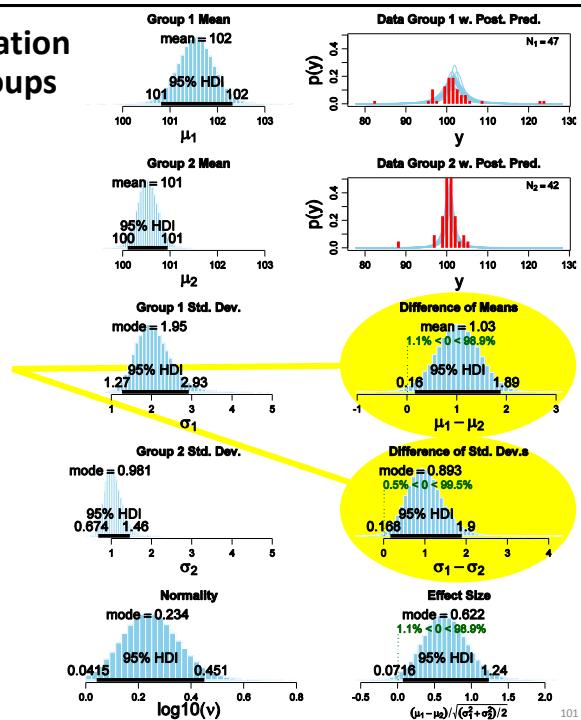
90

Robust Bayesian estimation for comparing two groups

Differences between groups?
Compute $\mu_1 - \mu_2$ and $\sigma_1 - \sigma_2$ at each of the many credible combinations.

Here, both differences are credibly non-zero.

(NHST would require two tests...)



© John K. Kruschke, 2013

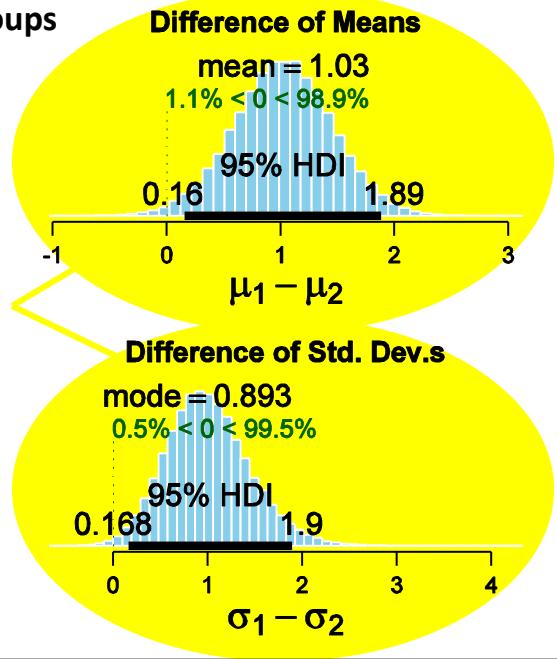
Robust Bayesian estimation for comparing two groups

Differences between
groups?

Compute $\mu_1 - \mu_2$
and $\sigma_1 - \sigma_2$
at each of the many
credible combinations.

Here, both differences
are credibly non-zero.

(NHST would require
two tests...)

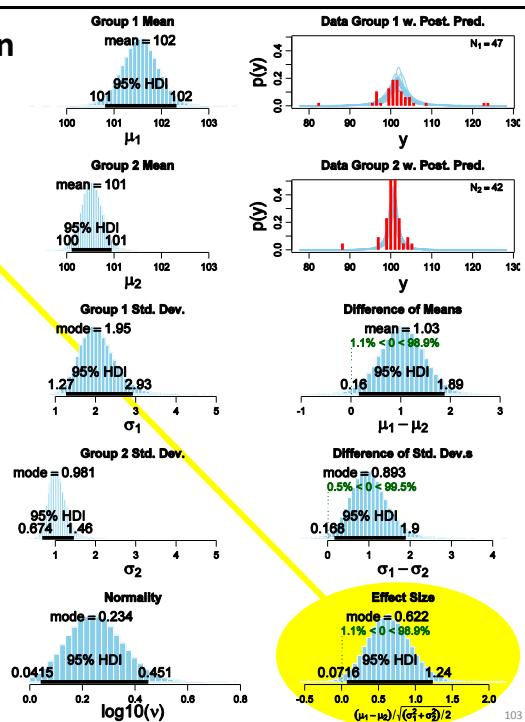


© John K. Kruschke, 2013

103

Robust Bayesian estimation for comparing two groups

Complete distribution
on effect size!

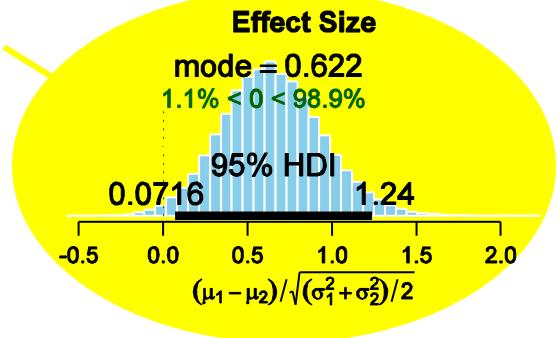


© John K. Kruschke, 2013

103

Robust Bayesian estimation for comparing two groups

Complete distribution
on effect size!



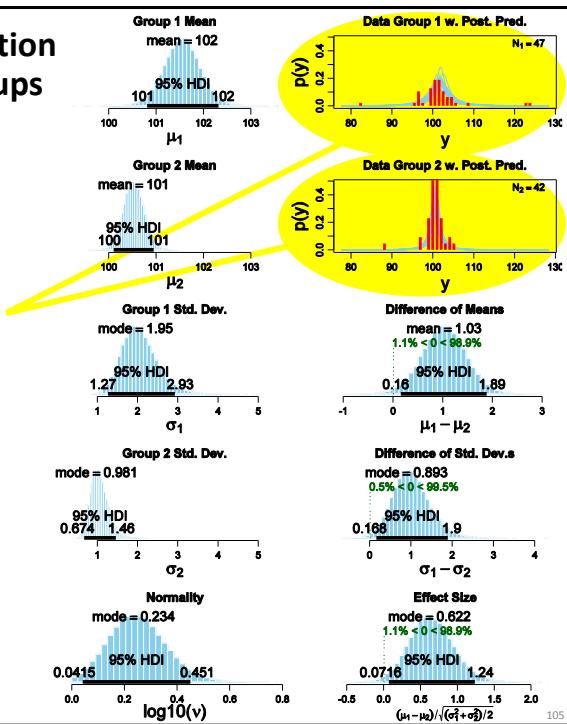
© John K. Kruschke, 2013

104

Robust Bayesian estimation for comparing two groups

Are the data described
well by the model?

Superimpose a
smattering of credible
descriptive distributions
on data.
= “posterior predictive
check”



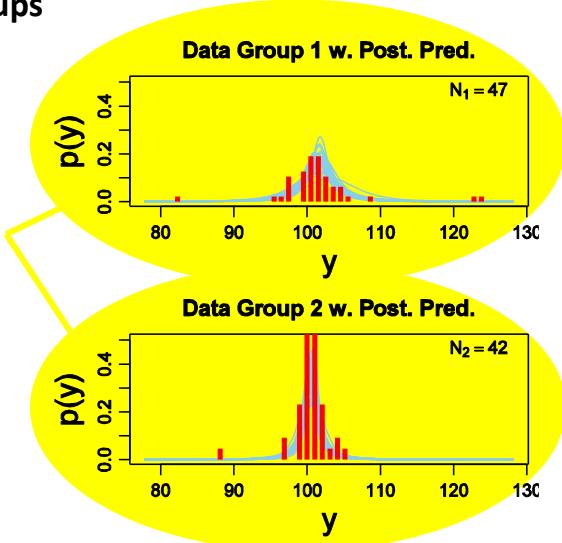
© John K. Kruschke, 2013

105

Robust Bayesian estimation for comparing two groups

**Are the data described
well by the model?**

Superimpose a
smattering of credible
descriptive distributions
on data.
= “posterior predictive
check”

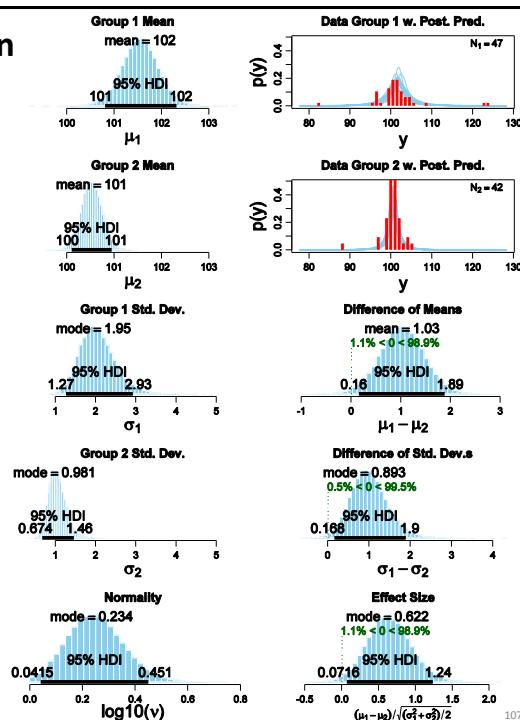


© John K. Kruschke, 2013

106

Robust Bayesian estimation for comparing two groups

Summary:
**→ Complete distribution
of credible parameter
values** (not merely point
estimate with ends of
confidence interval).
**→ Decisions about
multiple aspects of
parameters** (without
reference to p values).
**→ Flexible descriptive
model, robust to outliers**
(unlike NHST t test).



© John K. Kruschke, 2013

107

Agenda

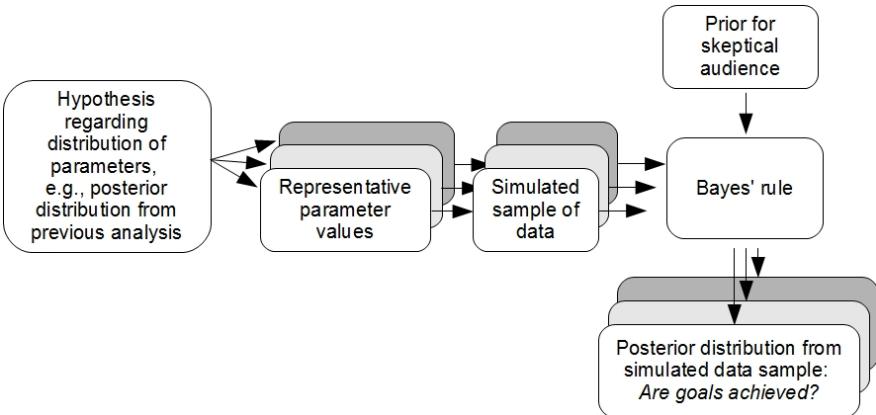
- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

Power Analysis

Power is
the probability of achieving your research goal,
 such as excluding a null value
 or achieving a desired precision,
if the world is as specified,
 e.g. by a previously-obtained posterior distribution,
and if you collect data a specified way,
 e.g. with sample size N.

Challenging to compute power meaningfully in NHST because the world is specified only as a punctate hypothesis, instead of as a distribution of possibilities.

Bayesian Power Analysis

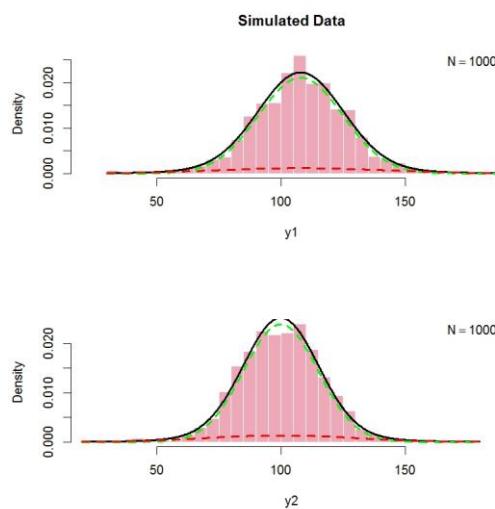


© John K. Kruschke, 2013

110

Prospective Power Analysis

The research hypothesis expressed as plausible data.



© John K. Kruschke, 2013

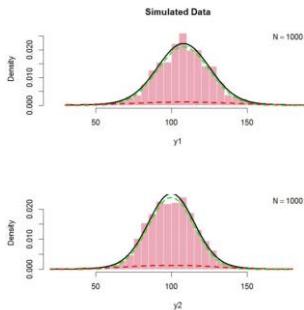
111

Prospective Power Analysis

The research hypothesis expressed as plausible data.

BESTexamplePower.R

```
# PROSPECTIVE POWER ANALYSIS, using fictitious strong data.
# Generate large fictitious data set that expresses hypothesis:
prospectData = makeData( mu1=108, sd1=17, mu2=100, sd2=15, nPerGrp=1000,
                         pcntOut=10, sdOutMult=2.0, rnd.seed=NULL )
y1pro = prospectData$y1 # Merely renames simulated data for convenience below.
y2pro = prospectData$y2 # Merely renames simulated data for convenience below.
```



© John K. Kruschke, 2013

112

Prospective Power Analysis

The research hypothesis expressed as
parameter distribution implied by plausible data.

BESTexamplePower.R

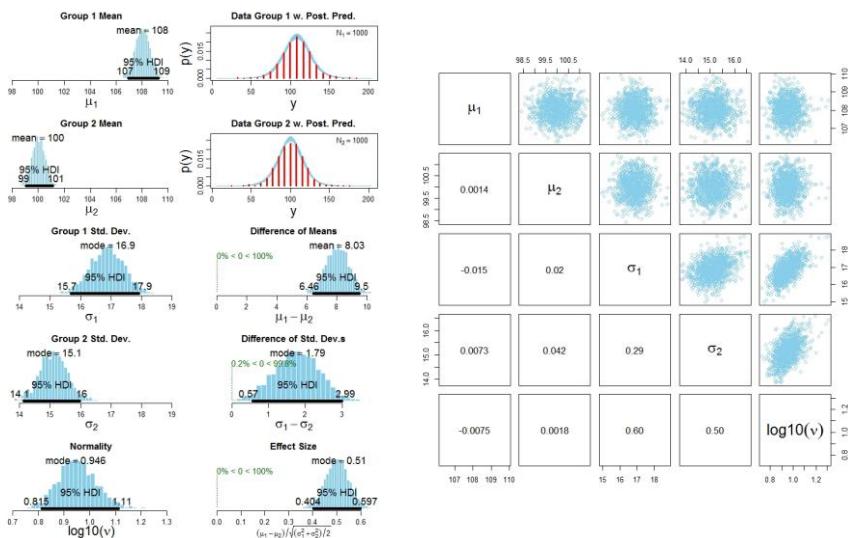
```
# Generate Bayesian posterior distribution from fictitious data:
# (uses fewer than usual MCMC steps because it only needs nRep credible
# parameter combinations, not a high-resolution representation)
mcmcChainPro = BESTmcmc( y1pro , y2pro , numSavedSteps=2000 )
postInfoPro = BESTplot( y1pro , y2pro , mcmcChainPro , pairsPlot=TRUE )
save( y1pro, y2pro, mcmcChainPro, postInfoPro,
      file="BESTexampleProPowerMCMC.Rdata" )
```

© John K. Kruschke, 2013

113

Prospective Power Analysis

The research hypothesis expressed as
parameter distribution implied by plausible data.

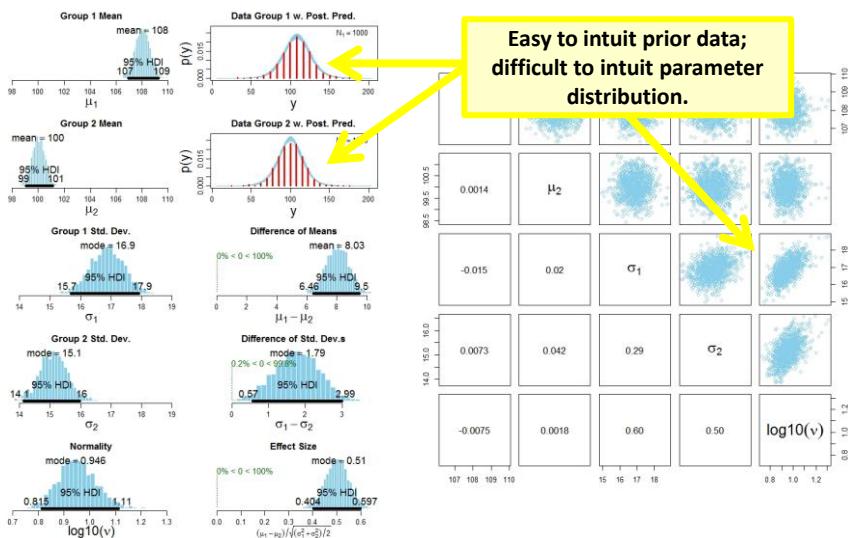


© John K. Kruschke, 2013

114

Prospective Power Analysis

The research hypothesis expressed as
parameter distribution implied by plausible data.



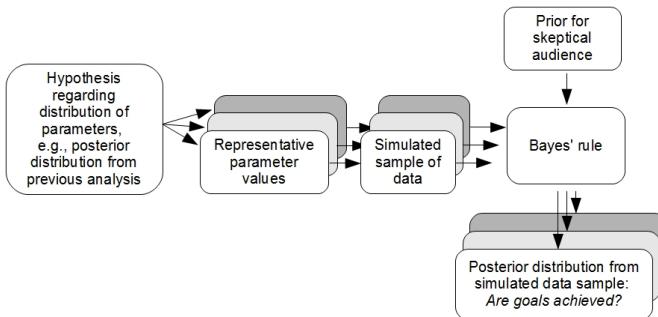
© John K. Kruschke, 2013

115

Prospective Power Analysis

BESTexamplePower.R

```
# Now compute the prospective power for planned sample sizes:
N1plan = N2plan = 50 # specify planned sample size
powerPro = BESTpower( mcmcChainPro , N1=N1plan , N2=N2plan , showFirstNrep=5 ,
ROPEm=c(-1.5,1.5) , ROPEsd=c(-0.0,0.0) , ROPEeff=c(-0.0,0.0) ,
maxHDIWm=15.0 , maxHDIWsd=10.0 , maxHDIWeff=1.0 , nRep=1000 ,
mcmcLength=10000 , saveName = "BESTexampleProPower.Rdata" )
```



© John K. Kruschke, 2013

116

Prospective Power Analysis

| Goal | Based on 1,000 simulated replications | | |
|--|---------------------------------------|-------------|-------------|
| | Bayesian Power | Lower Bound | Upper Bound |
| 95% HDI on the difference of means excludes ROPE of (-1.5, 1.5). | 40.1% | 37.1% | 43.1% |
| 95% HDI on the difference of means has width less than 15.0. | 72.6% | 69.8% | 75.3% |
| 95% HDI on the difference of standard deviations is greater than zero. | 10.5% | 8.6% | 12.4% |
| 95% HDI on the difference of standard deviations has width less than 10.0. | 15.8% | 13.5% | 18.0% |
| 95% HDI on the effect size is greater than zero. | 54.4% | 51.4% | 57.5% |
| 95% HDI on the effect size has width less than 1.0. | 97.8% | 96.9% | 98.7% |

Table 1

Bayesian prospective power analysis for parameter distribution in Figure 8, using $N_1 = N_2 = 50$. “Lower Bound” and “Upper Bound” refer to limits of the 95% HDI on the beta posterior for estimated power, which get closer together as the number of simulated replications increases.

© John K. Kruschke, 2013

117

NHST Power Analysis

```
tTestInfo = t.test(y1,y2)
show( tTestInfo )

varTestInfo = var.test(y1,y2,alternative="two.sided")
show( varTestInfo )

n = sqrt( length(y1) * length(y2) ) # Harmonic mean sample size.
sd = sqrt( (sd(y1)^2+sd(y2)^2)/2 ) # Pooled SD
ctPower = power.t.test( n=n , delta=mean(y1)-mean(y2) , sd=sd , power=NULL ,
                        sig.level=0.05 , type="two.sample" , alt="two.sided" , strict=TRUE )
loPower = power.t.test( n=n , delta=tTestInfo$conf.int[1] , sd=sd , power=NULL ,
                        sig.level=0.05 , type="two.sample" , alt="two.sided" , strict=TRUE )
hiPower = power.t.test( n=n , delta=tTestInfo$conf.int[2] , sd=sd , power=NULL ,
                        sig.level=0.05 , type="two.sample" , alt="two.sided" , strict=TRUE )
cat( "NHST low est power: " , round(loPower$power,3) , "\n" )
cat( "NHST mid est power: " , round(ctPower$power,3) , "\n" )
cat( "NHST high est power: " , round(hiPower$power,3) , "\n" )
```

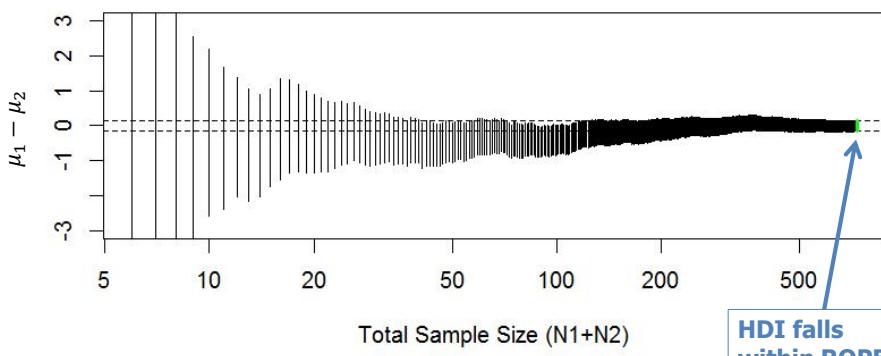
© John K. Kruschke, 2013

118

Sequential Testing

For simulated data *from the null hypothesis*:

Example of Correct Acceptance

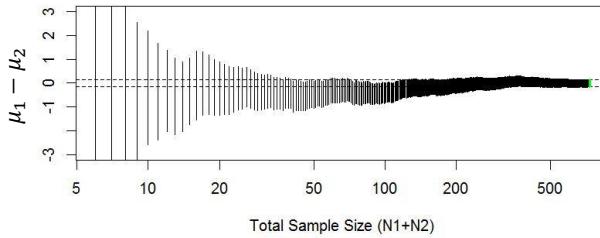


© John K. Kruschke, Nov. 2012

Sequential Testing

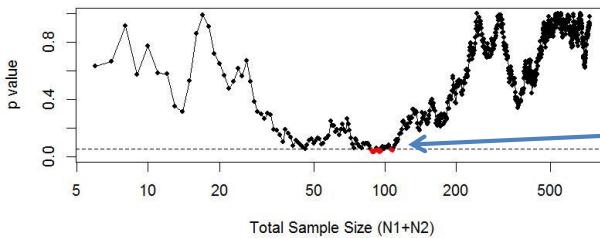
For simulated data from the null hypothesis:

Example of Correct Acceptance



HDI falls within ROPE:
Null correctly accepted.

p values for same data as above



NHST has no way to accept null.

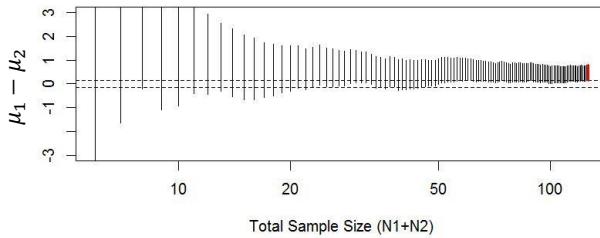
NHST falsely rejects null.

© John K. Kruschke

Sequential Testing

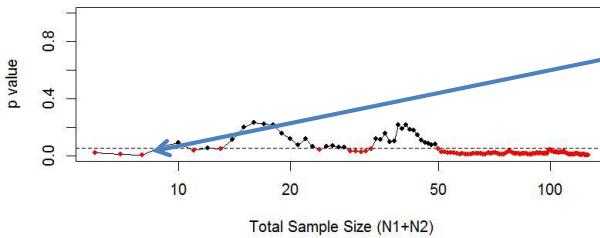
For simulated data from the null hypothesis:

Example of False Alarm



HDI falls outside ROPE:
Null falsely rejected.

p values for same data as above



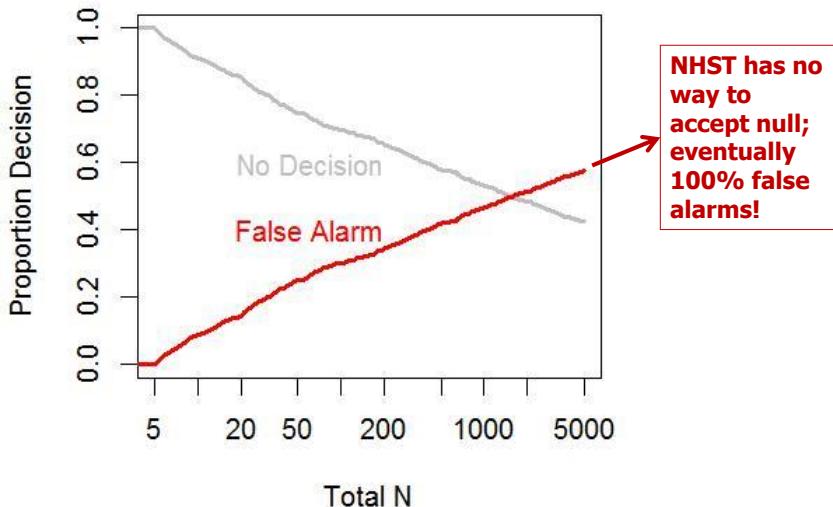
NHST falsely rejects null.

© John K. Kruschke

Sequential Testing

For simulated data from the null hypothesis:

Sequential testing by NHST

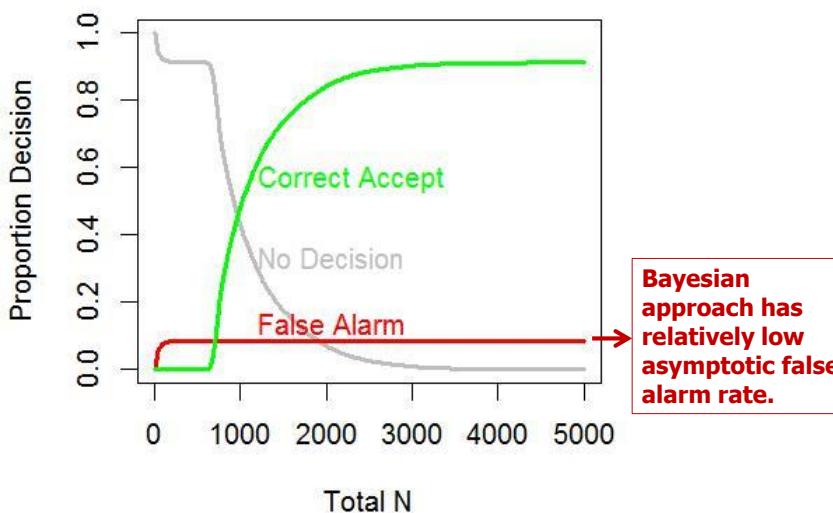


© John K. Kruschke, Nov. 2012

Sequential Testing

For simulated data from the null hypothesis:

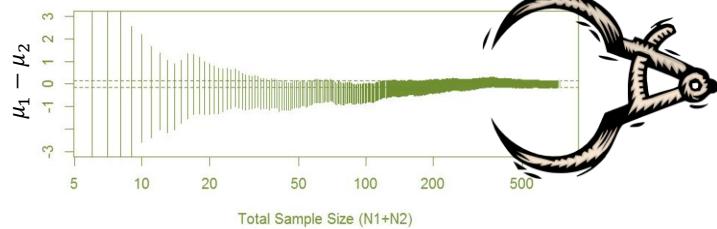
Sequential testing by HDI with ROPE



© John K. Kruschke, Nov. 2012

Goal: Precision in Estimation

Precision:
Width of
95% HDI

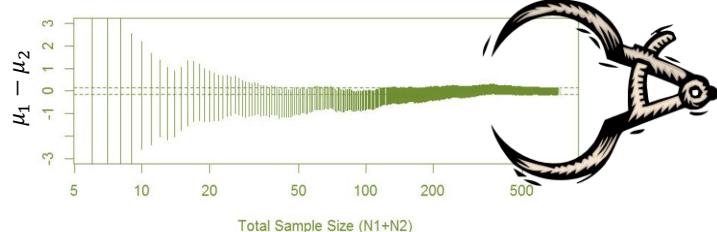


Better goal than rejecting null.
E.g., in election polling, want precise estimate.

© John K. Kruschke, Nov. 2012

Goal of Precision in Estimation

Precision:
Width of
95% HDI



- Easy to pursue in Bayesian, but not in frequentist.
- Bayesian power analysis for this goal is straight forward.

© John K. Kruschke, Nov. 2012

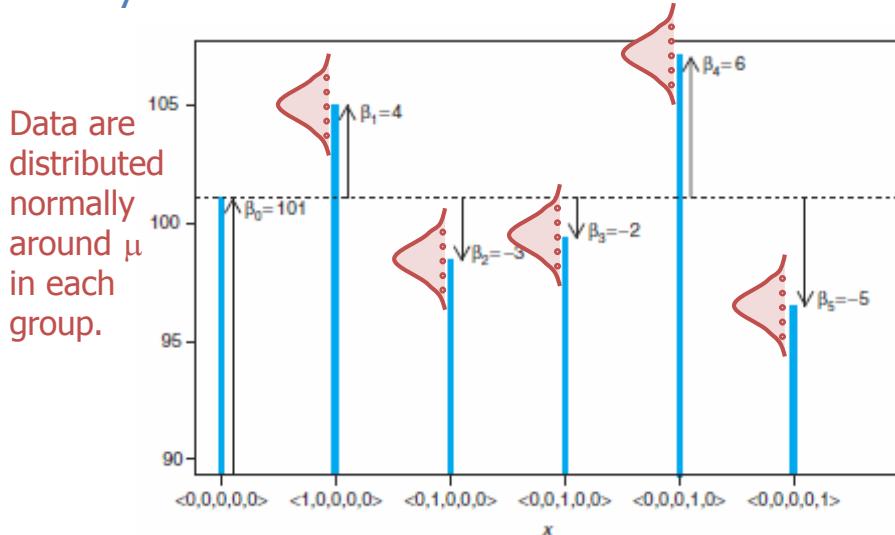
Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

© John K. Kruschke, 2013

126

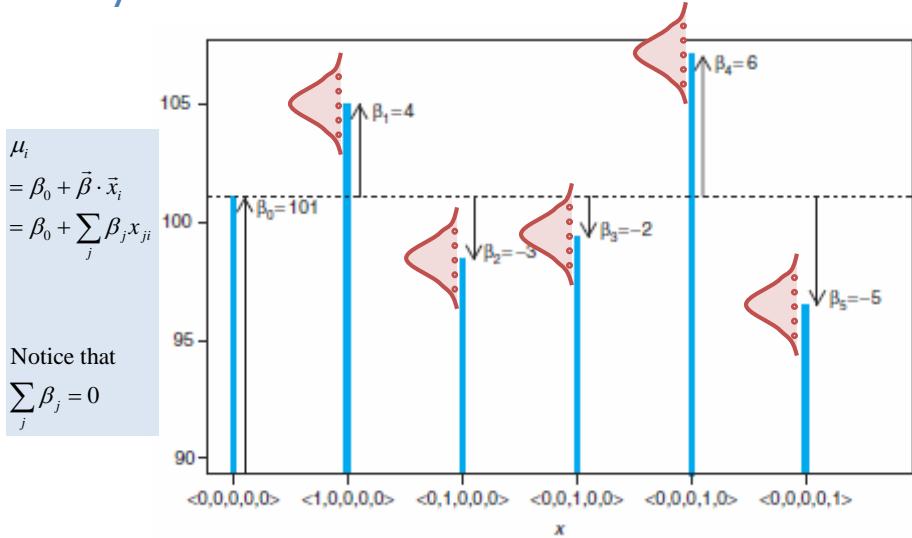
Bayesian hierarchical ANOVA: The data



© John K. Kruschke, 2013

127

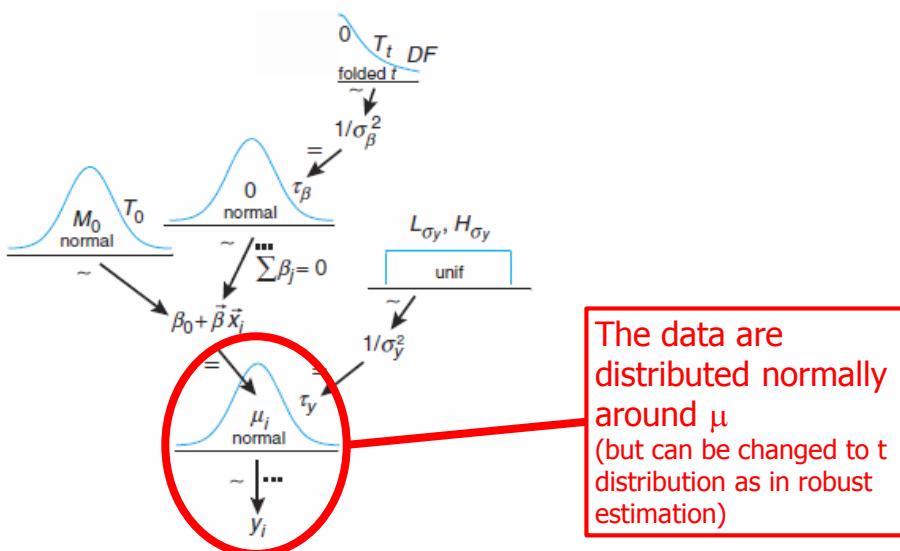
Bayesian hierarchical ANOVA: The model



© John K. Kruschke, 2013

128

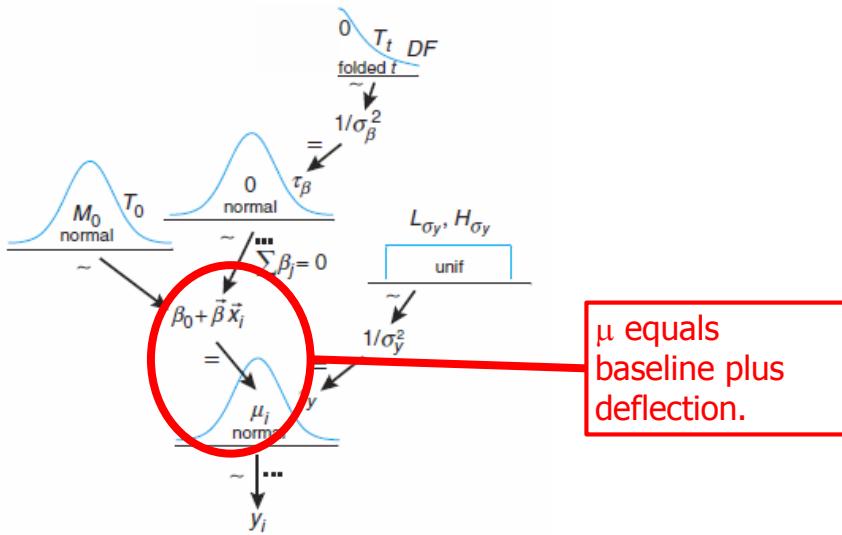
Bayesian hierarchical ANOVA



© John K. Kruschke, 2013

129

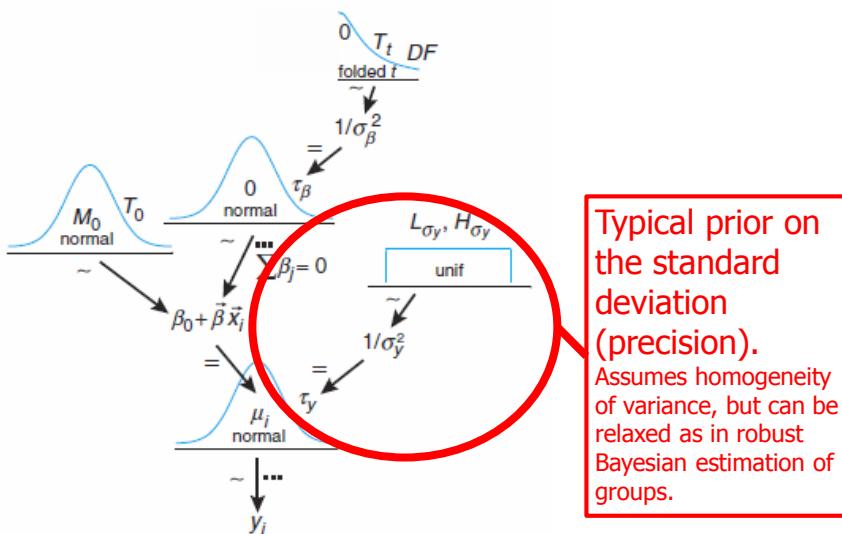
Bayesian hierarchical ANOVA



© John K. Kruschke, 2013

130

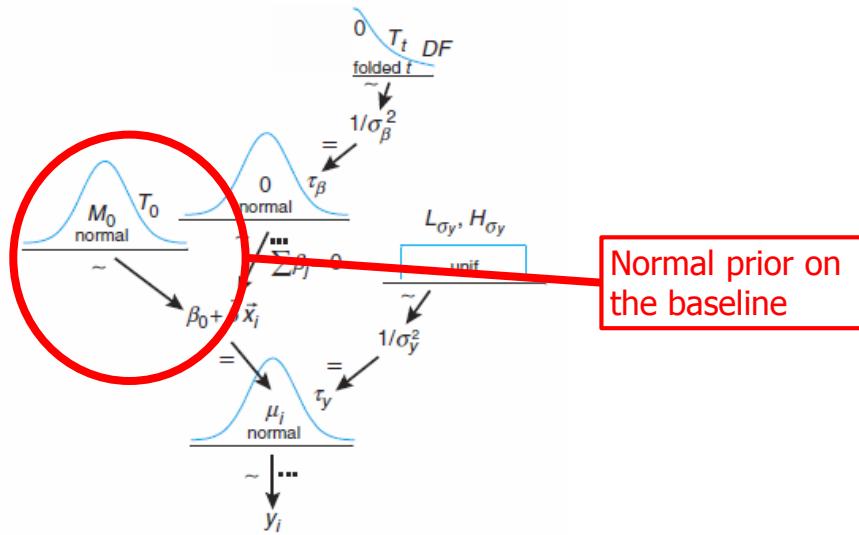
Bayesian hierarchical ANOVA



© John K. Kruschke, 2013

131

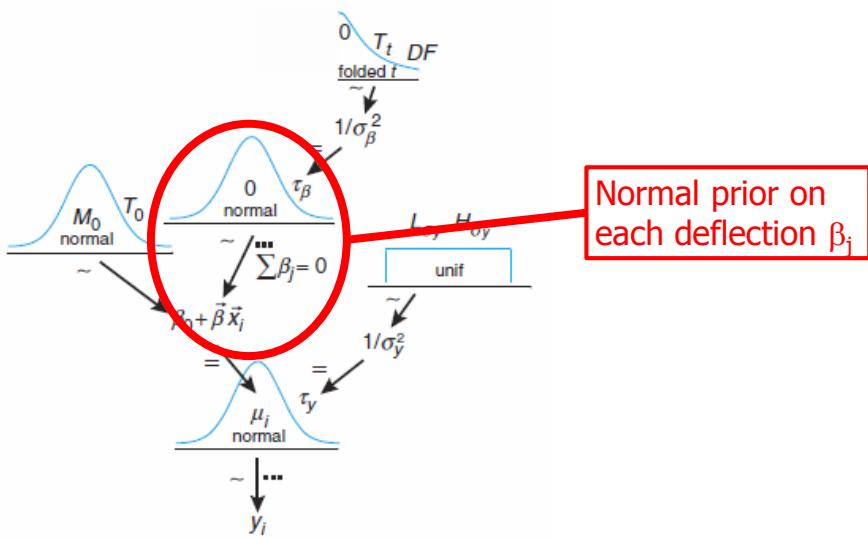
Bayesian hierarchical ANOVA



© John K. Kruschke, 2013

132

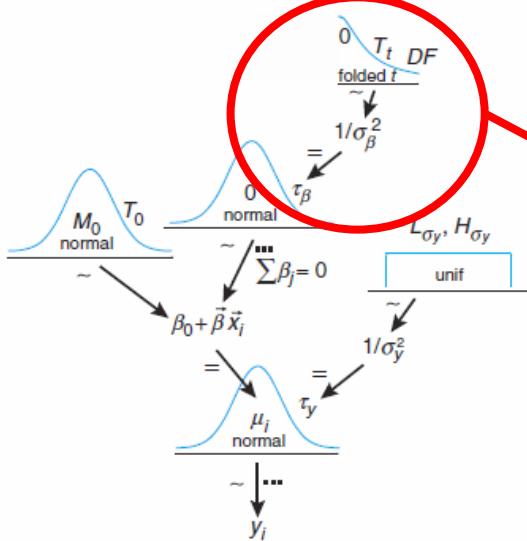
Bayesian hierarchical ANOVA



© John K. Kruschke, 2013

133

Bayesian hierarchical ANOVA

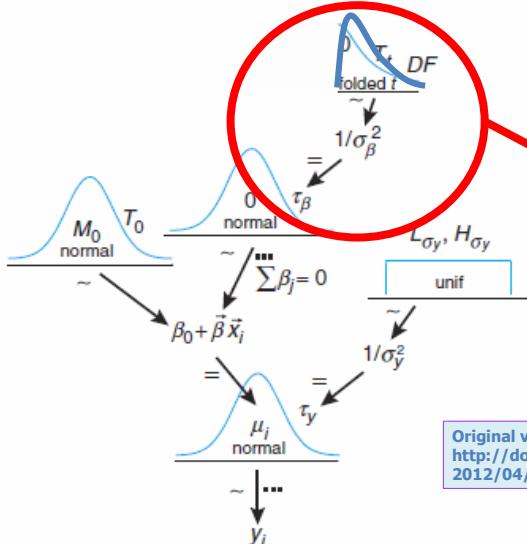


Hierarchical ANOVA: Estimate the standard deviation (precision) of the deflections across levels of x.

© John K. Kruschke, 2013

134

Bayesian hierarchical ANOVA



Hierarchical ANOVA: Estimate the standard deviation (precision) of the deflections across levels of x.

Original version is now revised. See
<http://doingbayesiandataanalysis.blogspot.com/2012/04/improved-programs-for-hierarchical.html>

© John K. Kruschke, 2013

135

Digression: The gamma distribution

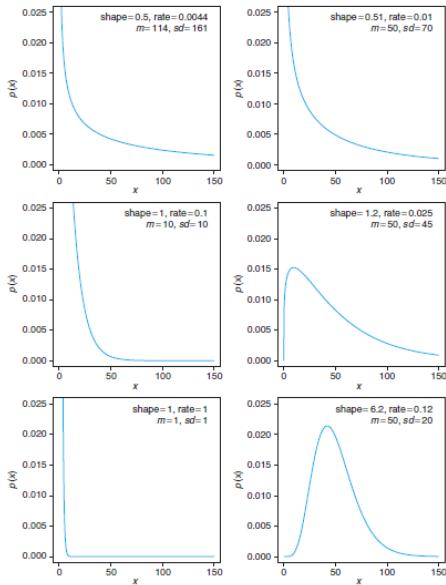


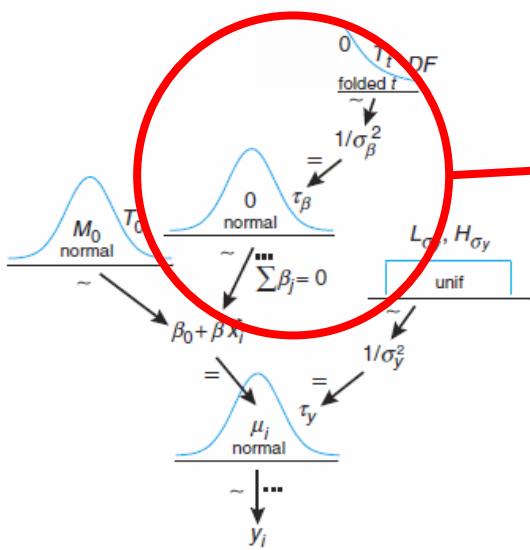
Figure 9.8: Examples of the gamma density distribution. The $\text{gamma}(x|s, r)$ distribution is a probability density for $x \geq 0$, given by $\text{gamma}(x|s, r) = \frac{r^s}{\Gamma(s)} x^{s-1} e^{-rx}$, where $\Gamma(s)$ is the gamma function: $\Gamma(s) = \int_0^\infty dt t^{s-1} e^{-t}$. The gamma function is a generalization of the factorial, because for positive integers, $\Gamma(s) = (s-1)!$. In the specification of the distribution, s is called the “shape” parameter and r is called the “rate” (or “inverse scale”) parameter. The mean of the gamma distribution is $m = s/r$, and the standard deviation of the gamma distribution is $sd = \sqrt{s}/r$. Hence $s = m^2/sd^2$ and $r = m/sd^2$. In R, the gamma density is provided by `dgamma(x, shape=s, rate=r)`, and the gamma function is provided by `gamma(s)`. Conveniently, BUGS parameterizes the gamma distribution the same way as R, i.e., with shape and rate parameters in that order.

For parameterization by mode, see the blog.

© John K. Kruschke, 2013

136

Bayesian hierarchical ANOVA



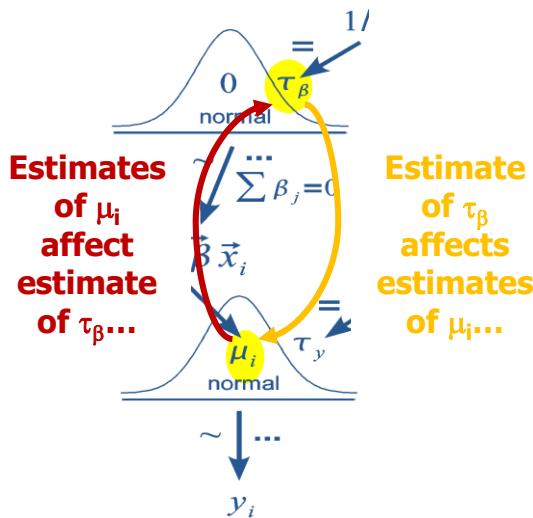
Shrinkage:
Estimate of β_j
is informed by
estimates of $\beta_{k \neq j}$
via estimate of σ_β

© John K. Kruschke, 2013

137

Shrinkage

Overall level



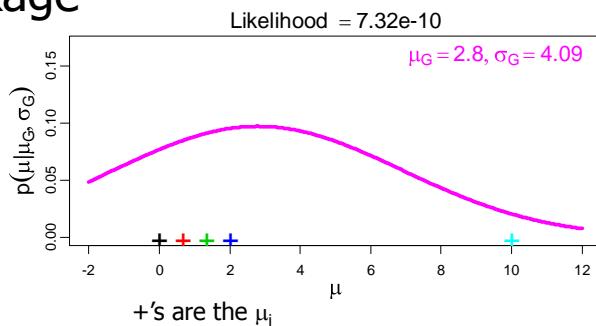
Groups level

© John K. Kruschke, 2013

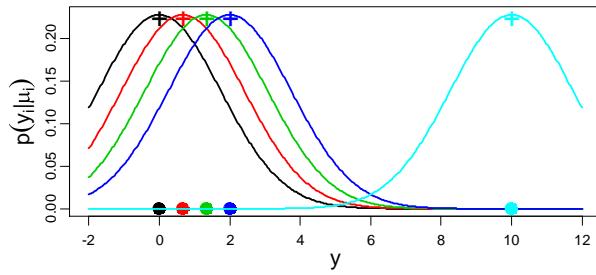
138

Shrinkage

Overall level



Groups level



© John K. Kruschke, 2013

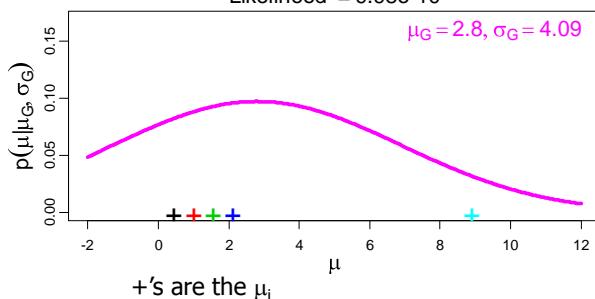
139

Shrinkage

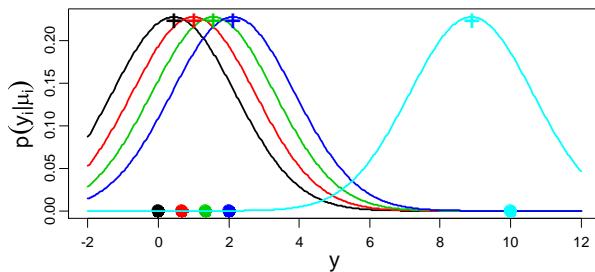
Likelihood = 9.98e-10

$$\mu_G = 2.8, \sigma_G = 4.09$$

Overall
level



Groups
level



© John K. Kruschke, 2013

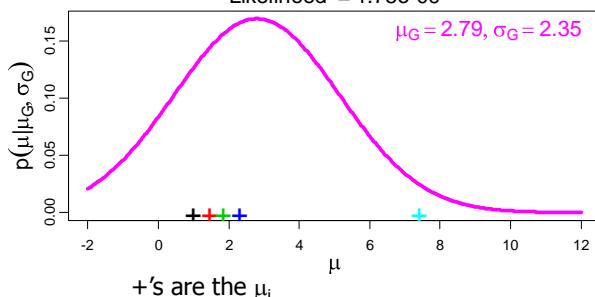
140

Shrinkage

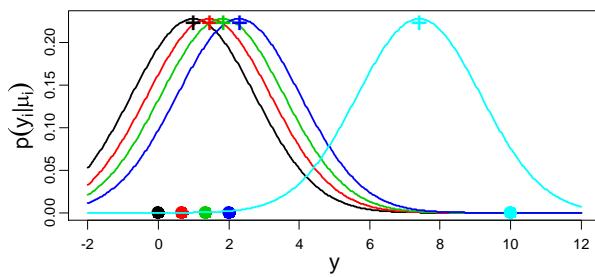
Likelihood = 1.75e-09

$$\mu_G = 2.79, \sigma_G = 2.35$$

Overall
level



Groups
level



© John K. Kruschke, 2013

141

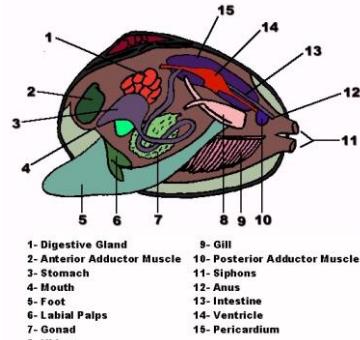
An Example

Consider mussel muscles. Is the size of the mussel muscle dependent upon geographical location? **Predicted variable** is relative length of AA muscle scar.



**Mytilus trossulus
from Alaska**

<http://www.bily.com/pnwsc/web-content/Photos/Bivalves/Mytilus%20complex,%20LS%20photo.jpg>



<http://www.marietta.edu/~biol/mussels/anatomy.html>

© John K. Kruschke, 2013

142

Predictor Groups

Five locations (groups): Tillamook, Oregon; Newport, Oregon; Petersburg, Alaska; Magadan, Russia; Tvarminne, Finland (data from McDonald, Seed & Koehn, 1991).

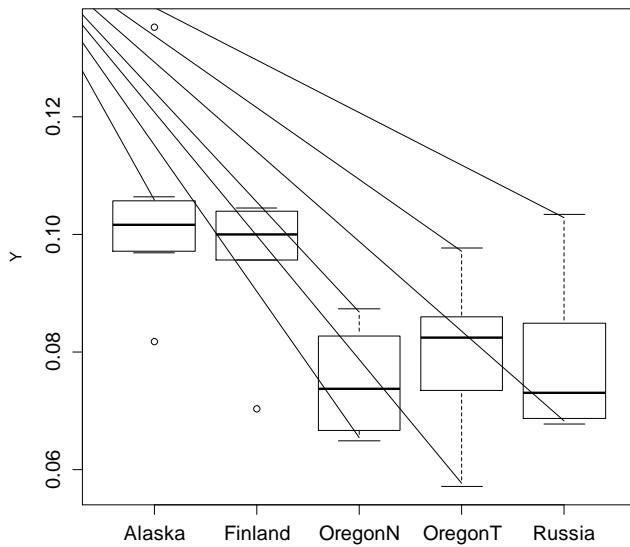


http://world_map_road.tripod.com/

© John K. Kruschke, 2013

143

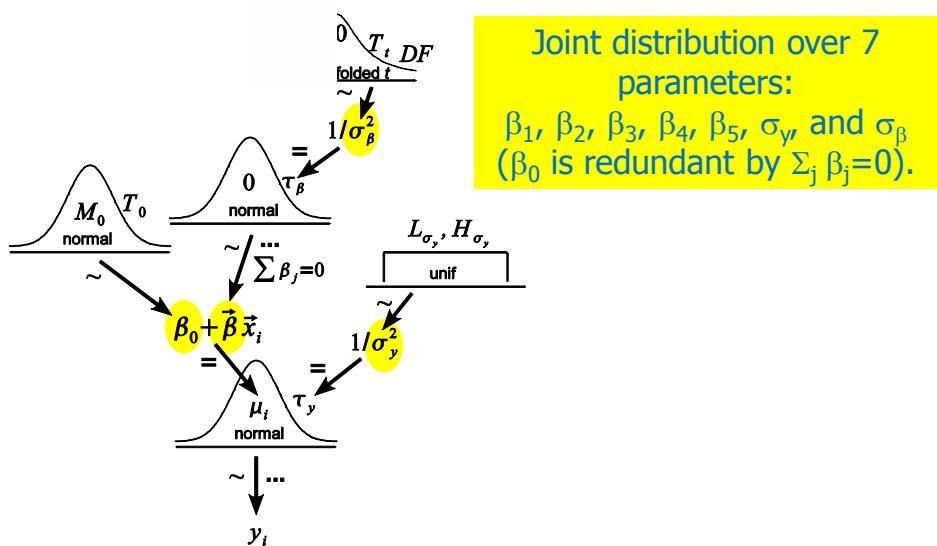
The Data (McDonald, Seed & Koehn, 1991)



© John K. Kruschke, 2013

144

Hierarchical Bayesian Model

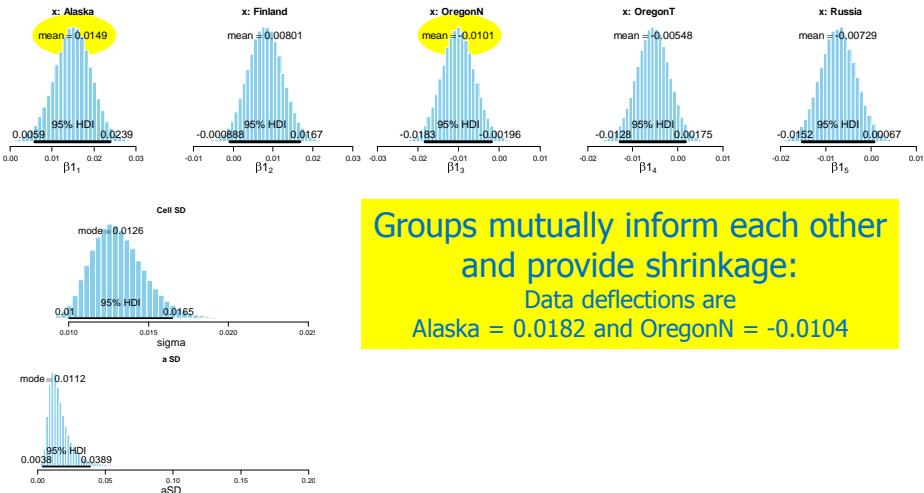


© John K. Kruschke, 2013

145

Posterior Distribution

Program: ANOVAonewayJagsSTZ.R

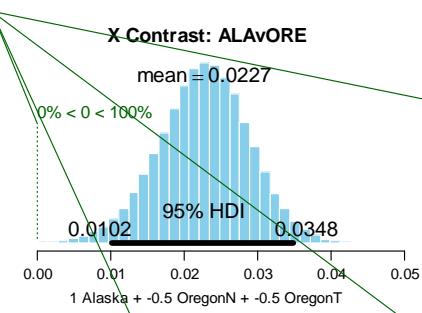


146

Posterior Distribution: Comparisons

Most interest usually is in comparisons across groups or combinations of groups.

Alaska versus average of two Oregon sites:
 $\beta_{\text{Alaska}} - (\beta_{\text{OregonN}} + \beta_{\text{OregonT}})/2$
 computed at each step in chain.



© John K. Kruschke, 2013

147

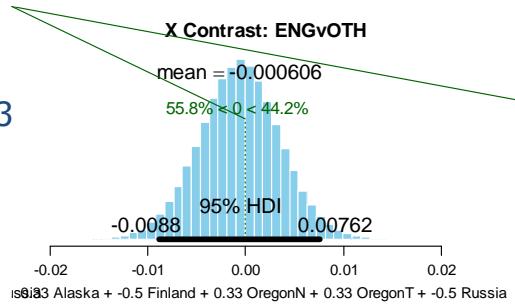
Posterior Distribution: Comparisons

Most interest usually is in comparisons across groups or combinations of groups.

English speaking versus non-English sites:

$$\begin{aligned} & (\beta_{\text{Alaska}} + \beta_{\text{OregonN}} + \beta_{\text{OregonT}})/3 \\ & - (\beta_{\text{Finland}} + \beta_{\text{Russia}})/2 \end{aligned}$$

computed at each step in chain.

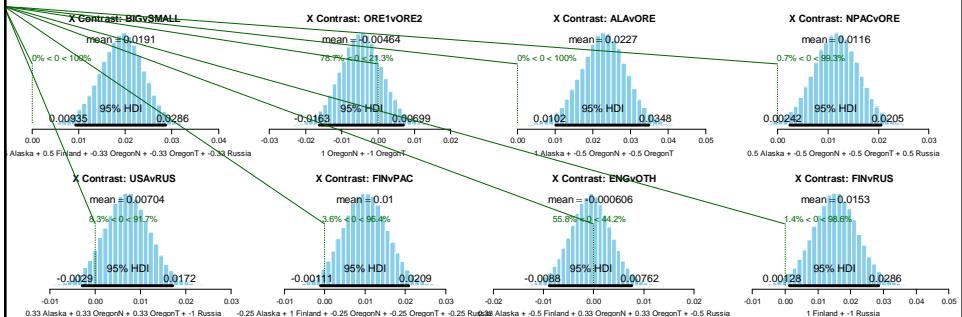


© John K. Kruschke, 2013

148

Posterior Distribution: Comparisons

Can investigate all comparisons of interest without any “corrections” for multiple comparisons



© John K. Kruschke, 2013

149

Bayesian Multiple Comparisons

- No analysis is immune to false alarms, because data are random and accidental conspiracies of rogue data can arise. But the Bayesian posterior distribution is the best inference we can make given the data we have.
- *Bayesian hierarchical ANOVA diminishes false alarms by letting the data and prior inform shrinkage of group estimates.*
- Attenuation of false alarms from hierarchical model does not use unknowable intentions: There is no notion of “planned” comparisons versus “post hoc” comparisons, and no counting of comparisons, as in NHST.

© John K. Kruschke, 2013

150

Bayesian hierarchical ANOVA vs classical NHST ANOVA

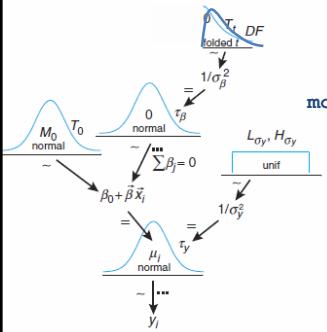
Omnibus test: De-emphasized. Rarely is the omnibus test of interest.

Classical ANOVA means partitioning of variance into between-group and within-group components via least-squares, which plays no direct role here. But σ_w^2 and σ_B^2 are estimated.

© John K. Kruschke, 2013

151

Bayesian hierarchical ANOVA



ANOVAonewayJagsSTZ.R

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- a0 + a[x[i]]
  }
  tau <- 1 / pow( sigma , 2 )
  sigma ~ dunif(0,10) # y values standardized
  a0 ~ dnorm(0,0.001) # y values standardized
  for ( j in 1:NxLvl ) {
    a[j] ~ dnorm( 0.0 , atau )
  }
  atau <- 1 / pow( aSD , 2 )
  aSD ~ dgamma(1.01005,0.1005) # mode=0.1,sd=10.0

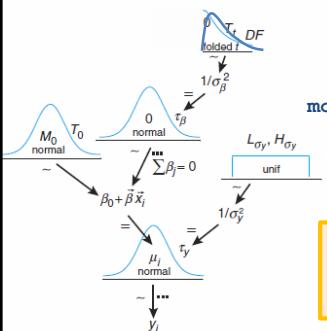
  # Convert a0,a[] to sum-to-zero b0,b[] :
  for ( j in 1:NxLvl ) { m[j] <- a0 + a[j] }
  b0 <- mean( m[1:NxLvl] )
  for ( j in 1:NxLvl ) { b[j] <- m[j] - b0 }
}
```

Each arrow in diagram has corresponding line in model specification.

© John K. Kruschke, 2013

152

Bayesian hierarchical ANOVA



ANOVAonewayJagsSTZ.R

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- a0 + a[x[i]]
  }
  a[j] ~ dnorm( 0.0 , atau )
}
atau <- 1 / pow( aSD , 2 )
aSD ~ dgamma(1.01005,0.1005) # mode=0.1,sd=10.0

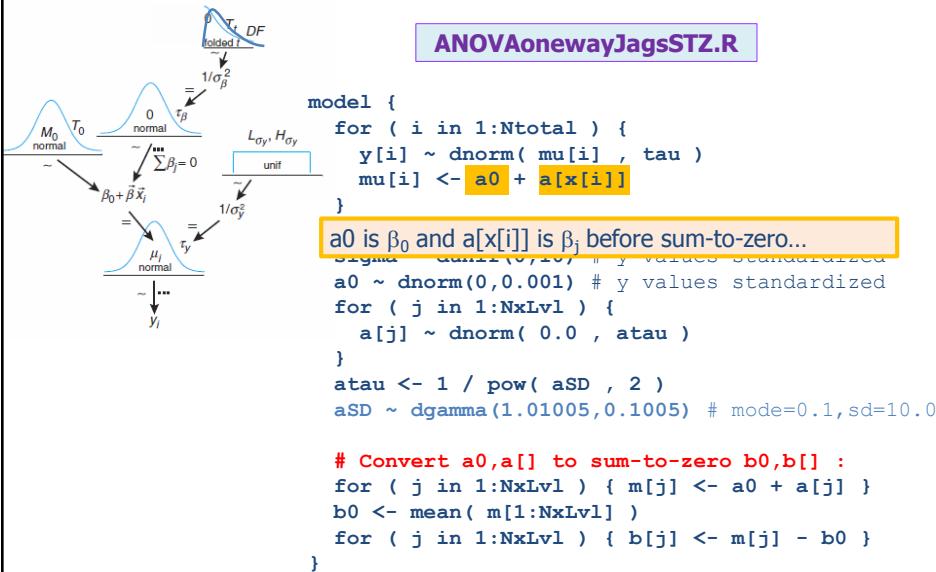
# Convert a0,a[] to sum-to-zero b0,b[] :
for ( j in 1:NxLvl ) { m[j] <- a0 + a[j] }
b0 <- mean( m[1:NxLvl] )
for ( j in 1:NxLvl ) { b[j] <- m[j] - b0 }
```

Each row of data contains $y[i]$ and $x[i]$ for individual i , where $x[i]$ indicates the group j of individual i .

© John K. Kruschke, 2013

153

Bayesian hierarchical ANOVA



© John K. Kruschke, 2013

154

ANOVAonewayJagsSTZ.R: Data

| Group | Size | Site |
|-------|--------|---------|
| 1 | 0.0571 | OregonT |
| 1 | 0.0813 | OregonT |
| 1 | 0.0831 | OregonT |
| ... | | |
| 2 | 0.0873 | OregonN |
| 2 | 0.0662 | OregonN |
| 2 | 0.0672 | OregonN |
| ... | | |
| 3 | 0.0974 | Alaska |
| 3 | 0.1352 | Alaska |
| 3 | 0.0817 | Alaska |
| ... | | |
| 4 | 0.1033 | Russia |
| 4 | 0.0915 | Russia |
| 4 | 0.0781 | Russia |
| ... | | |
| 5 | 0.0703 | Finland |
| 5 | 0.1026 | Finland |
| 5 | 0.0956 | Finland |
| ... | | |

```

datarecord = read.table( "McDonaldSK1991data.txt" , header=T ,
  colclasses=c("factor","numeric") )
y = as.numeric(datarecord$Size)
Ntotal = length(datarecord$Size)
x = as.numeric(datarecord$Site)
xnames = levels(datarecord$Site)
NxLvl = length(unique(datarecord$Site))
normalize = function( v ){ return( v / sum(v) ) }
contrastList = list(
  BIGvSMALL = normalize(xnames=="Alaska" | xnames=="Finland") -
    normalize(xnames=="OregonN" | xnames=="OregonT" | xnames=="Russia") ,
  ORE1vORE2 = (xnames=="OregonN")-(xnames=="OregonT") ,
  ALAvORE = (xnames=="Alaska")-normalize(xnames=="OregonN" | xnames=="OregonT") ,
  NPACvORE = normalize(xnames=="Alaska" | xnames=="Russia") -
    normalize(xnames=="OregonN" | xnames=="OregonT") ,
  USAvRUS = normalize(xnames=="Alaska" | xnames=="OregonN" | xnames=="OregonT") -
    (xnames=="Russia") ,
  FINvPAC = (xnames=="Finland") -
    normalize(xnames=="Alaska" | xnames=="Russia" |
      xnames=="OregonN" | xnames=="OregonT") ,
  ENGvOTH = normalize(xnames=="Alaska" | xnames=="OregonN" | xnames=="OregonT") -
    normalize(xnames=="Finland" | xnames=="Russia") ,
  FINvRUS = (xnames=="Finland")-(xnames=="Russia")
)

```

© John K. Kruschke, 2013

155

ANOVAonewayJagsSTZ.R: Initialize chains

```
theData = data.frame( y=dataList$y , x=factor(x,labels=xnames) )
a0 = mean( theData$y )
a = aggregate( theData$y , list( theData$x ) , mean )[,2] - a0
ssw = aggregate( theData$y , list( theData$x ) ,
                  function(x){var(x)*(length(x)-1)} )[,2]
sp = sqrt( sum( ssx ) / length( theData$y ) )
initsList = list( a0 = a0 , a = a , sigma = sp , aSDunabs = sd(a) )
```

© John K. Kruschke, 2013

156

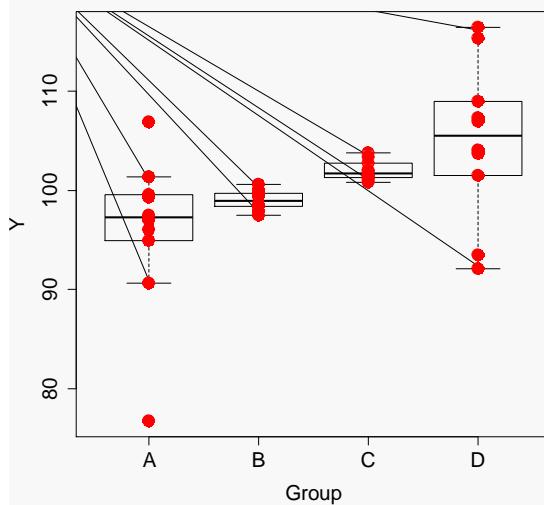
Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

© John K. Kruschke, 2013

157

Non-homogeneous Variances



<http://doingbayesiandataanalysis.blogspot.com/2011/04/anova-with-non-homogeneous-variances.html>

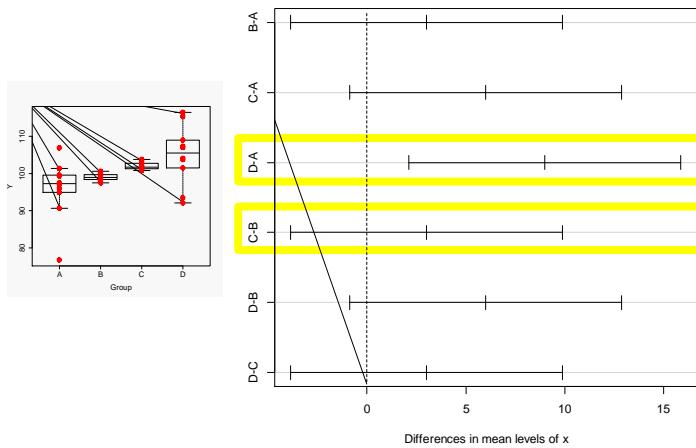
© John K. Kruschke, 2013

158

Non-homogeneous Variances

NHST ANOVA: R's `aov()` and `TukeyHSD()` *assumes homogeneity of variance*.

95% family-wise confidence level

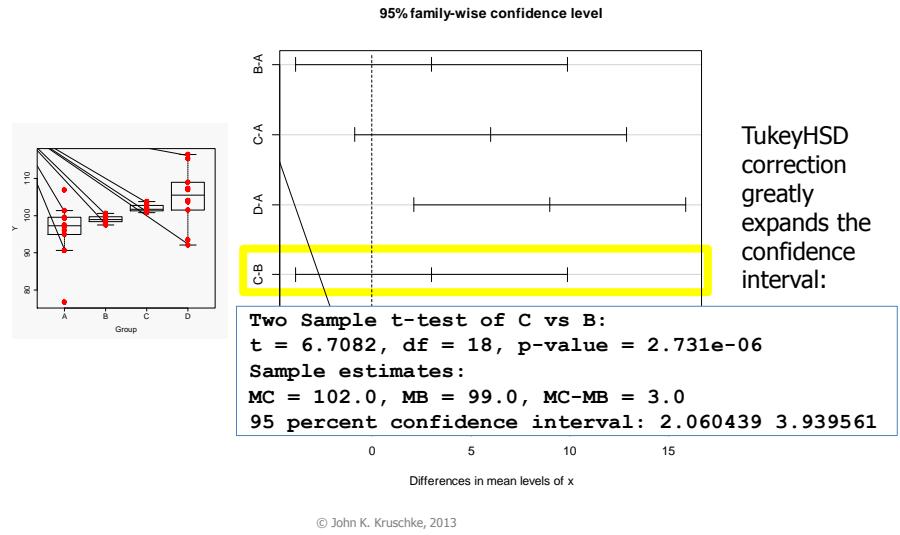


© John K. Kruschke, 2013

159

Non-homogeneous Variances

NHST ANOVA: R's `aov()` and `TukeyHSD()` *assumes homogeneity of variance.*

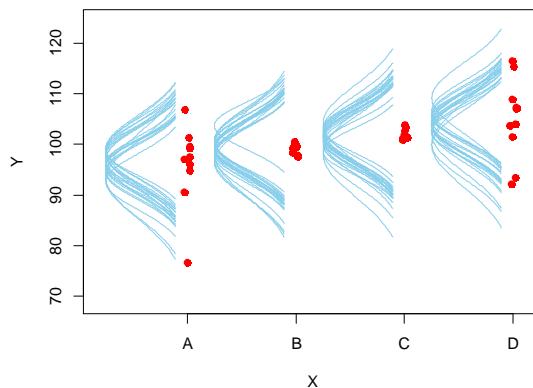


160

Non-homogeneous Variances

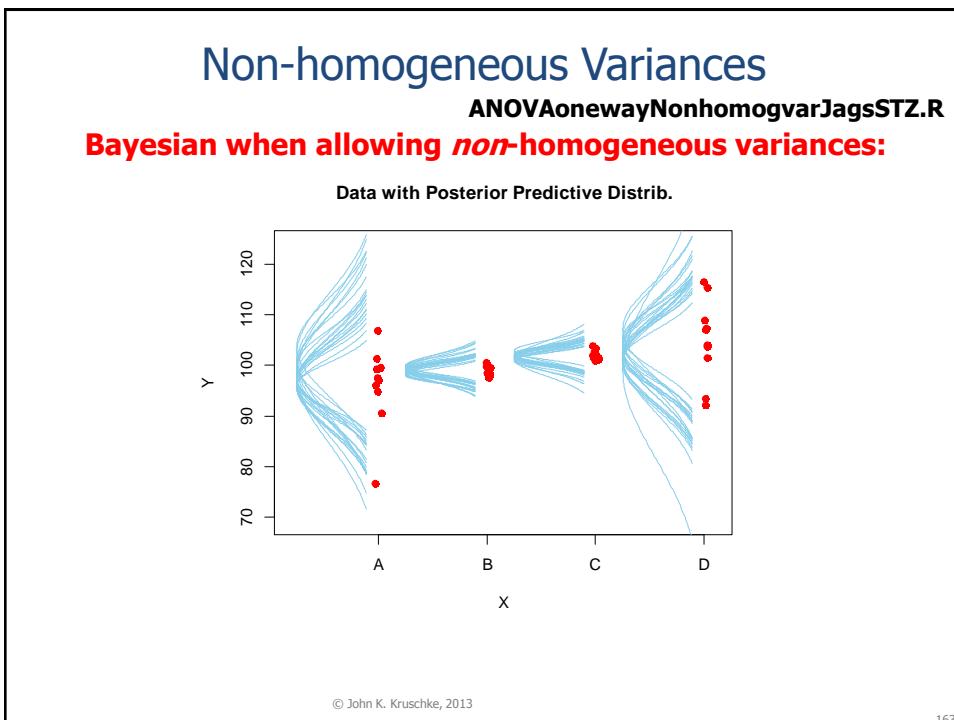
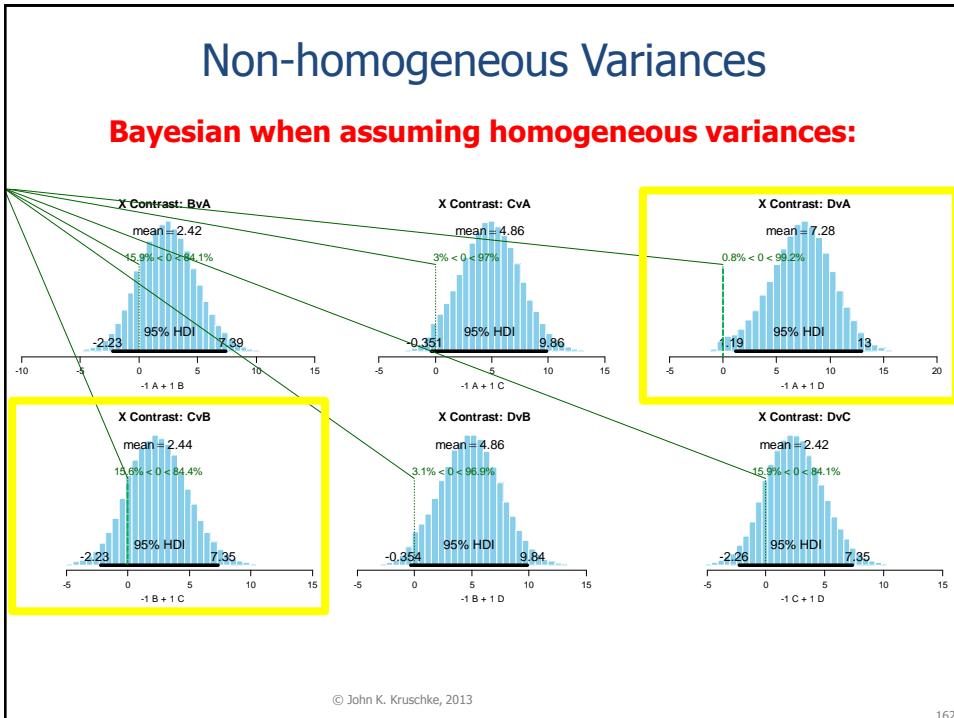
Bayesian when assuming homogeneous variances:

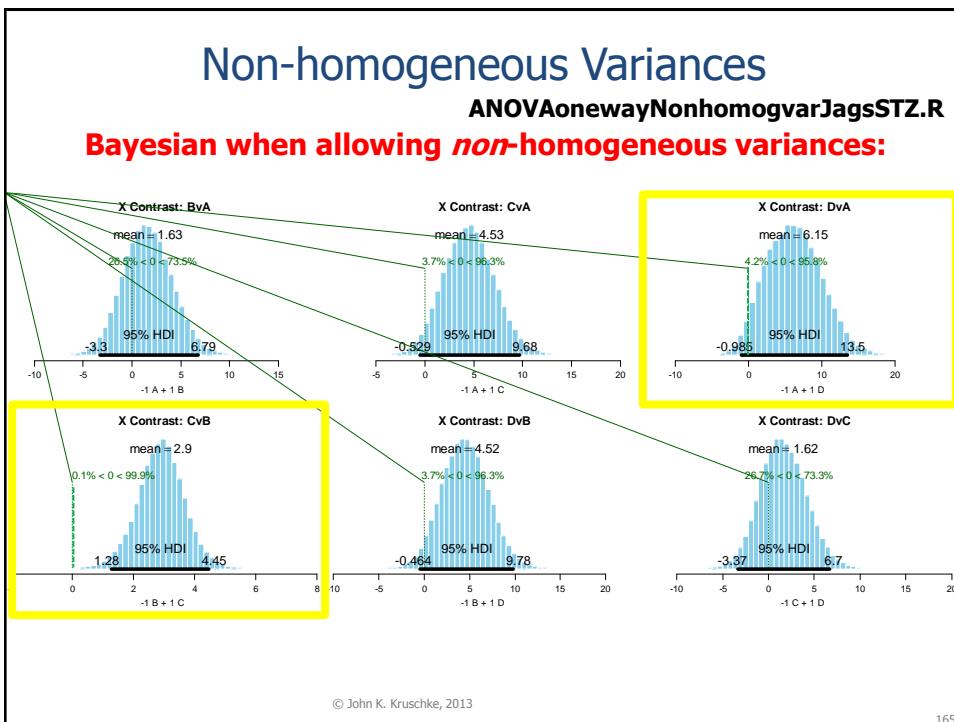
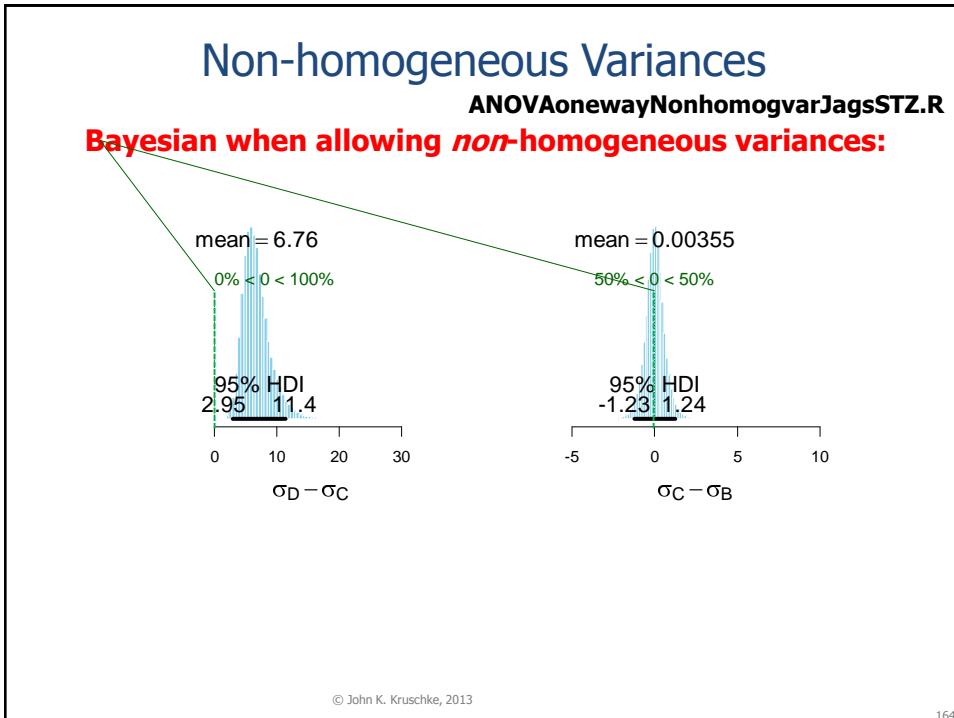
Data with Posterior Predictive Distrib.



© John K. Kruschke, 2013

161





Non-homogeneous Variances

ANOVAonewayNonhomogvarJagsSTZ.R

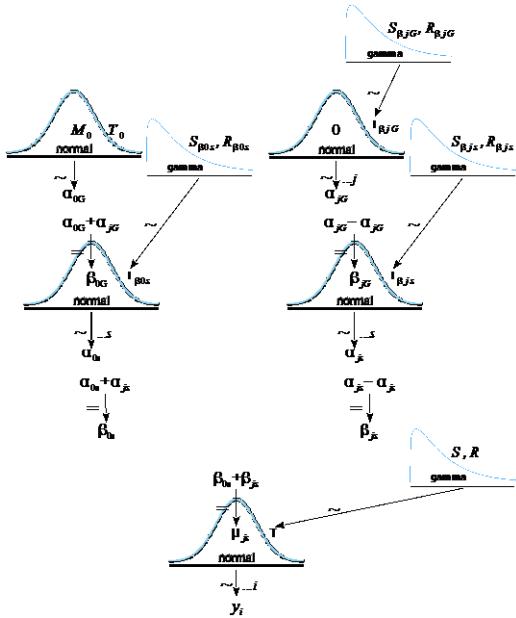
```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dnorm( mu[i] , tau[x[i]] )
    mu[i] <- a0 + a[x[i]]
  }
  a0 ~ dnorm(0,0.001)
  for ( j in 1:NxLvl ) {
    a[j] ~ dnorm( 0.0 , atau )
    tau[j] ~ dgamma( sG , rG )
  }
  sG <- pow(m,2)/pow(d,2)
  rG <- m/pow(d,2)
  m ~ dgamma(1,1)
  d ~ dgamma(1,1)
  atau <- 1 / pow( aSD , 2 )
  aSD ~ dgamma(1.01005,0.1005) # mode=0.1, sd=10.0
# Convert a0,a[] to sum-to-zero b0,b[] :
  for ( j in 1:NxLvl ) { mpred[j] <- a0 + a[j] }
  b0 <- mean( mpred[1:NxLvl] )
  for ( j in 1:NxLvl ) { b[j] <- mpred[j] - b0 }
}
```

*Again, the flexibility
of JAGS/BUGS for
expressing
hierarchical models!*

© John K. Kruschke, 2013

166

Single-factor ("oneway") within-subjects "ANOVA"



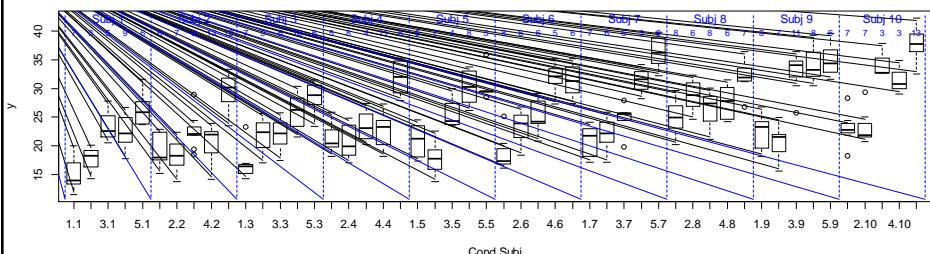
NEW!

and
improvable!

© John K. Kruschke, 2013

167

Single-factor ("oneway") within-subjects "ANOVA"



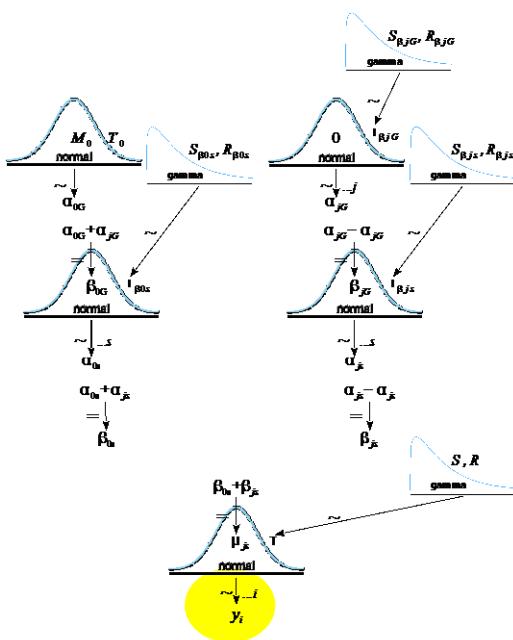
The data.

- 5 conditions.
- Each of 10 subjects contributes multiple measures to every condition.
- Different n in each cell (indicated above boxplots).

© John K. Kruschke, 2013

168

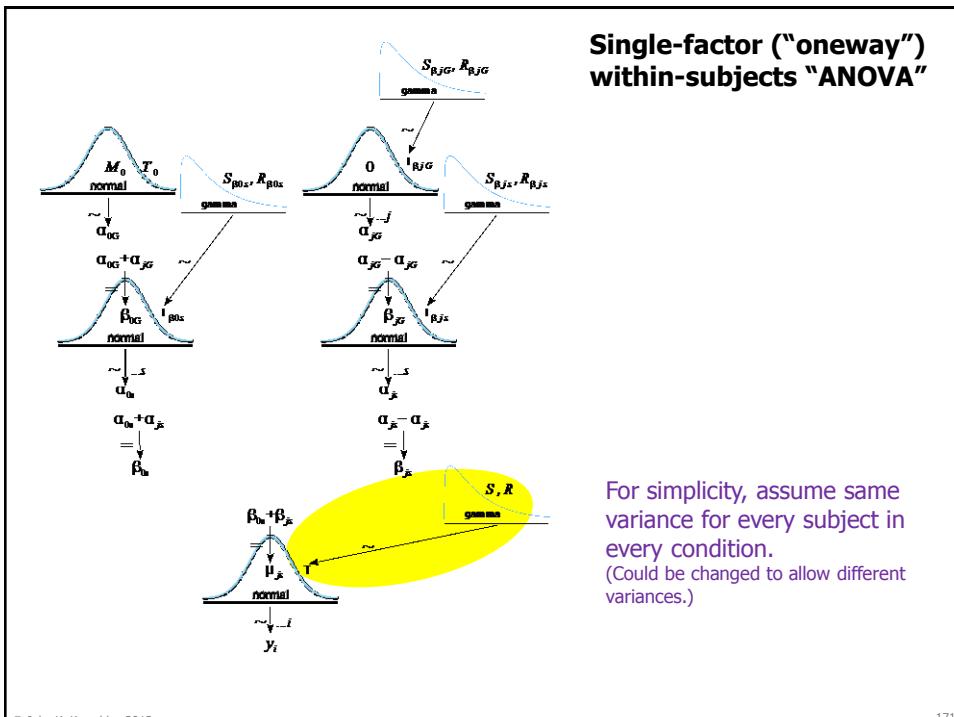
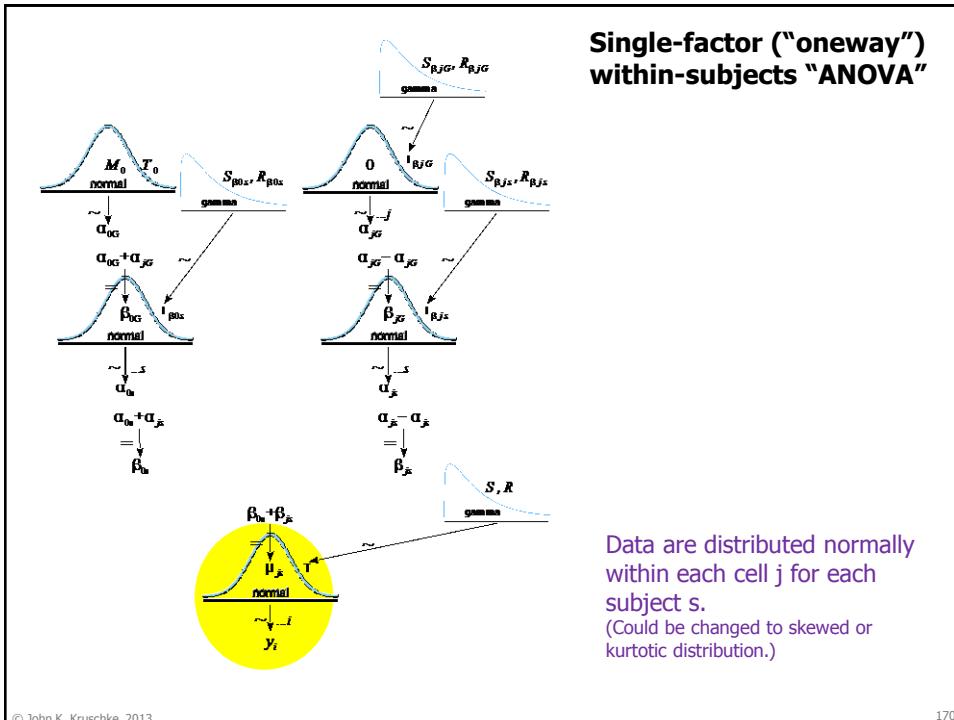
Single-factor ("oneway") within-subjects "ANOVA"

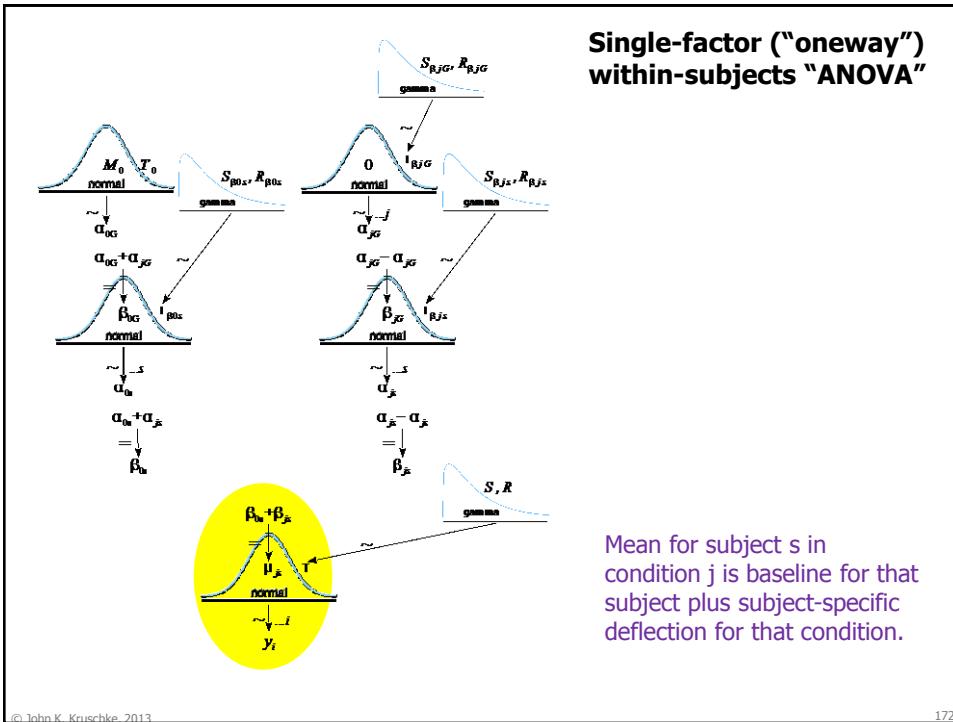


Repeated measures for
subject s within condition j .
Each instance is indexed by i .
(No need for "balanced" design.)

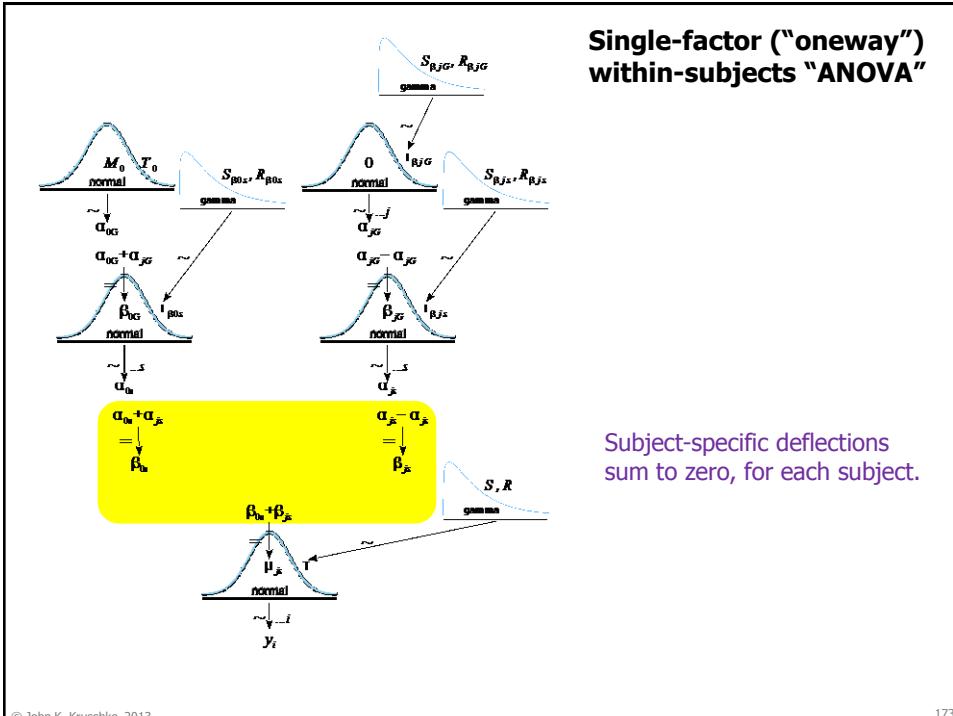
© John K. Kruschke, 2013

169

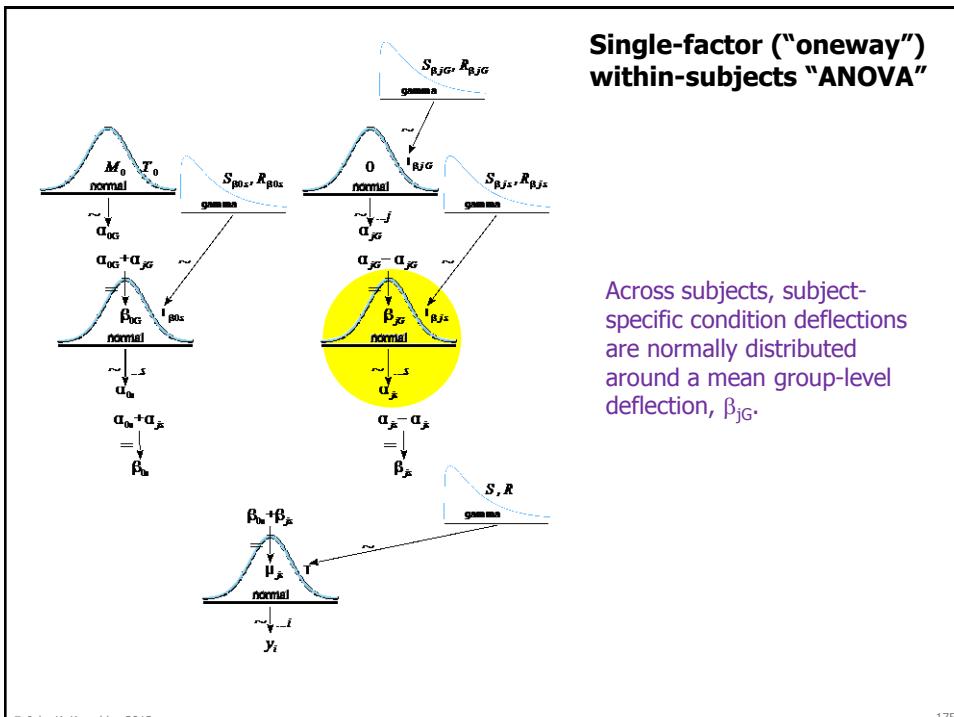
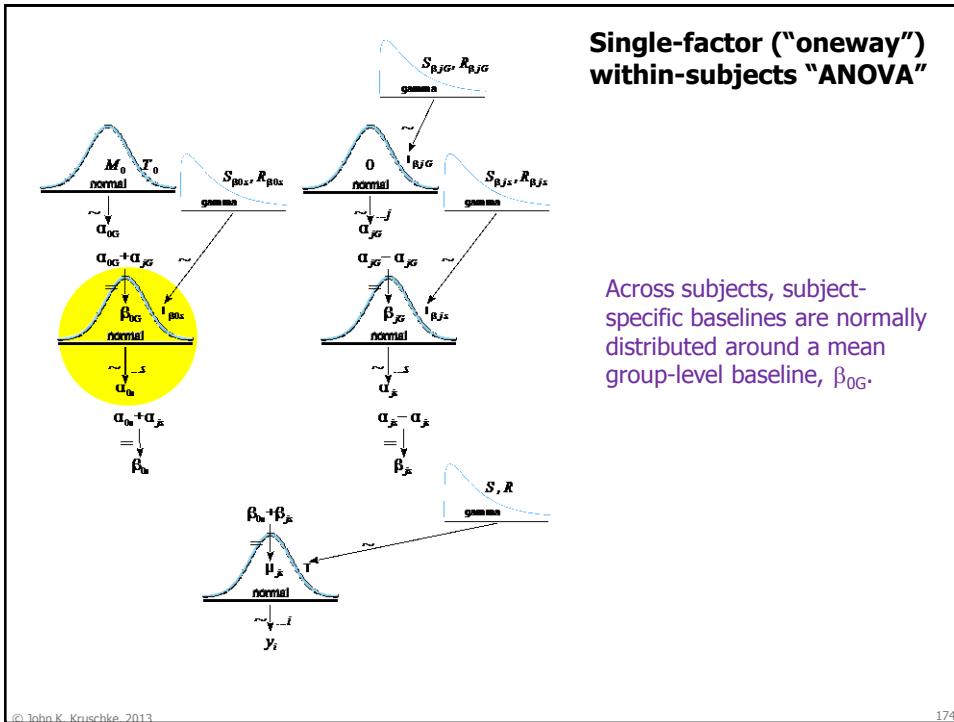


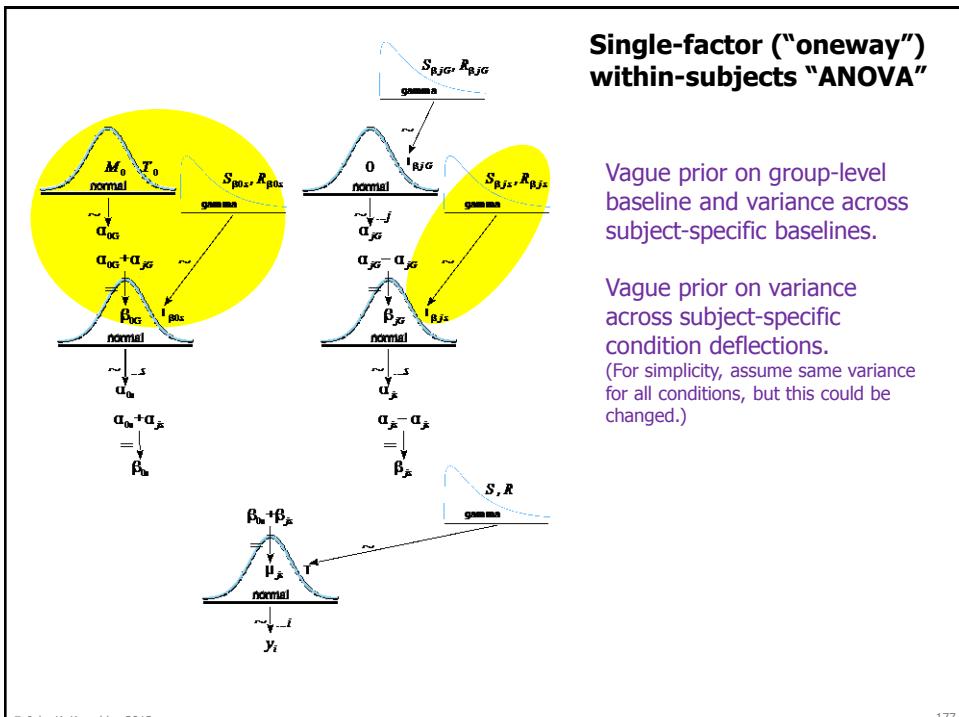
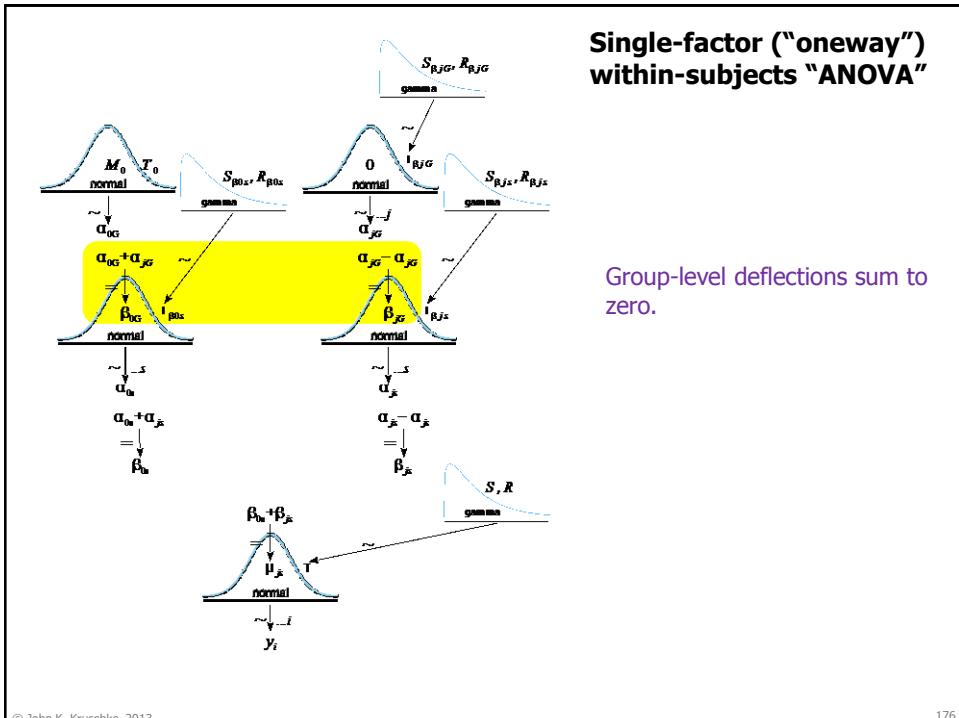


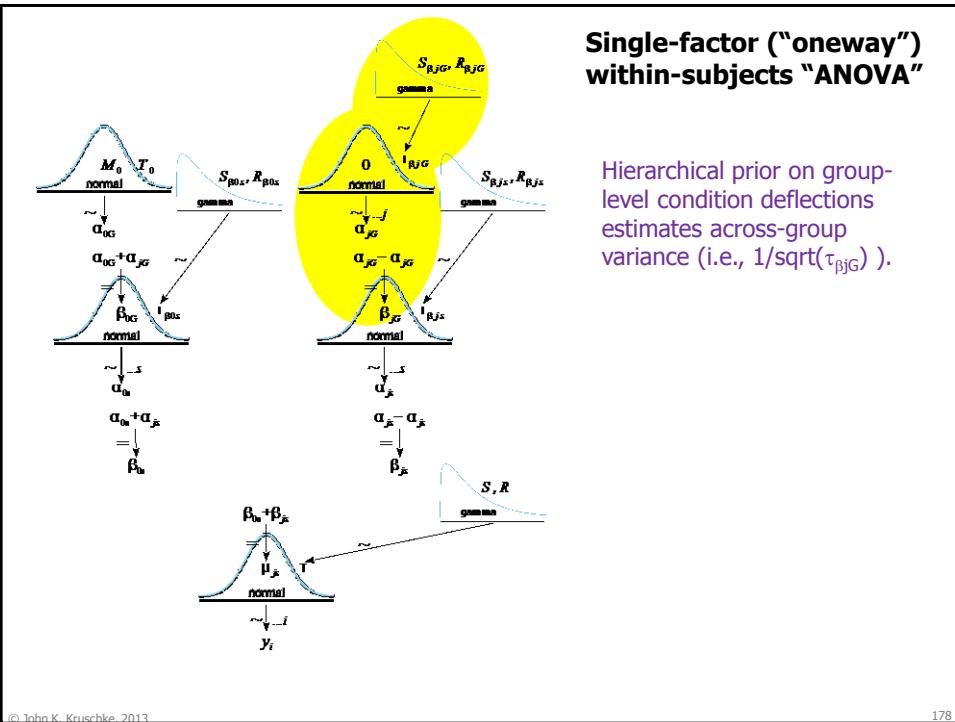
172



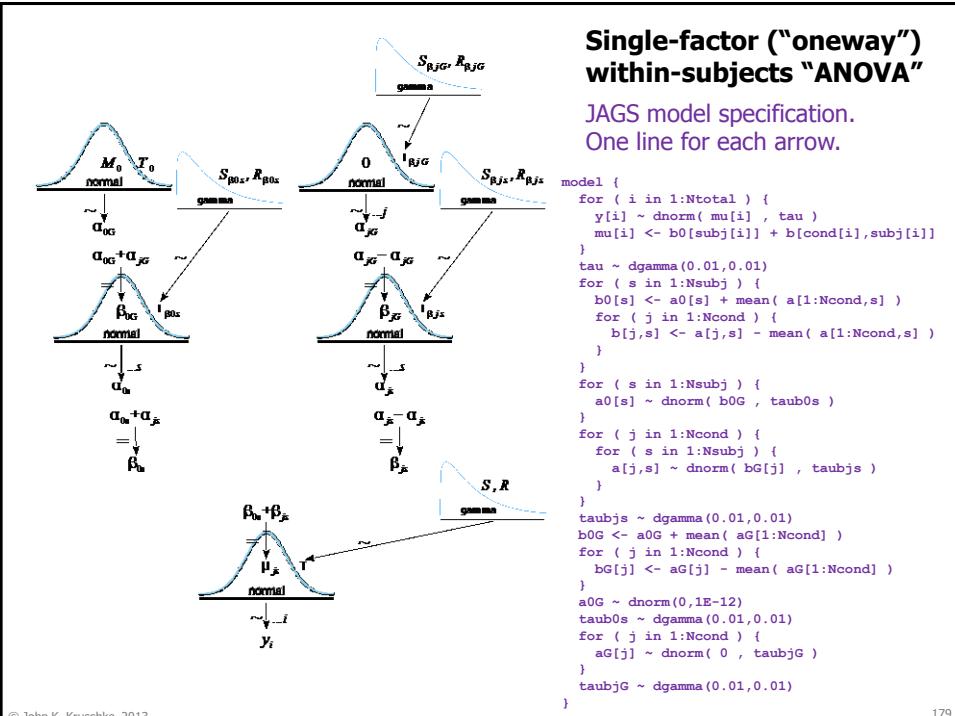
173



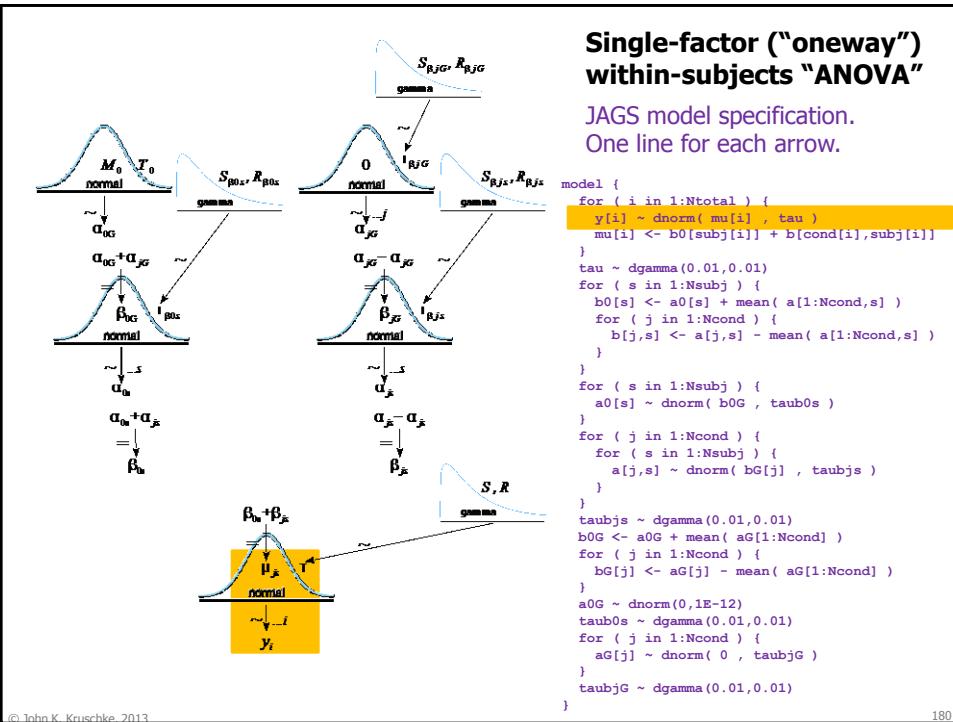




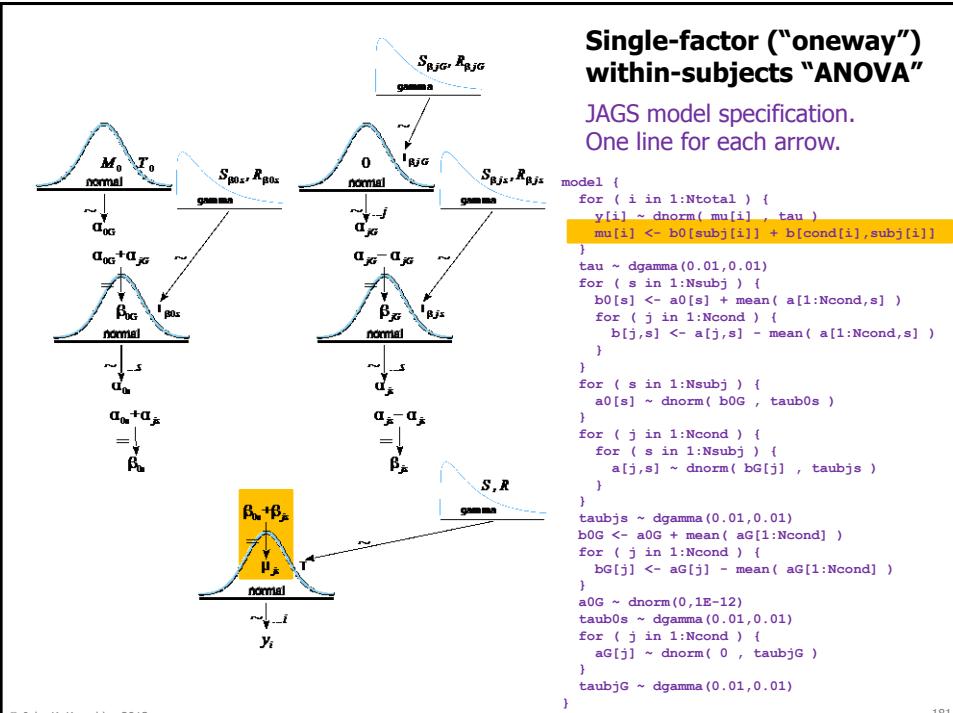
178



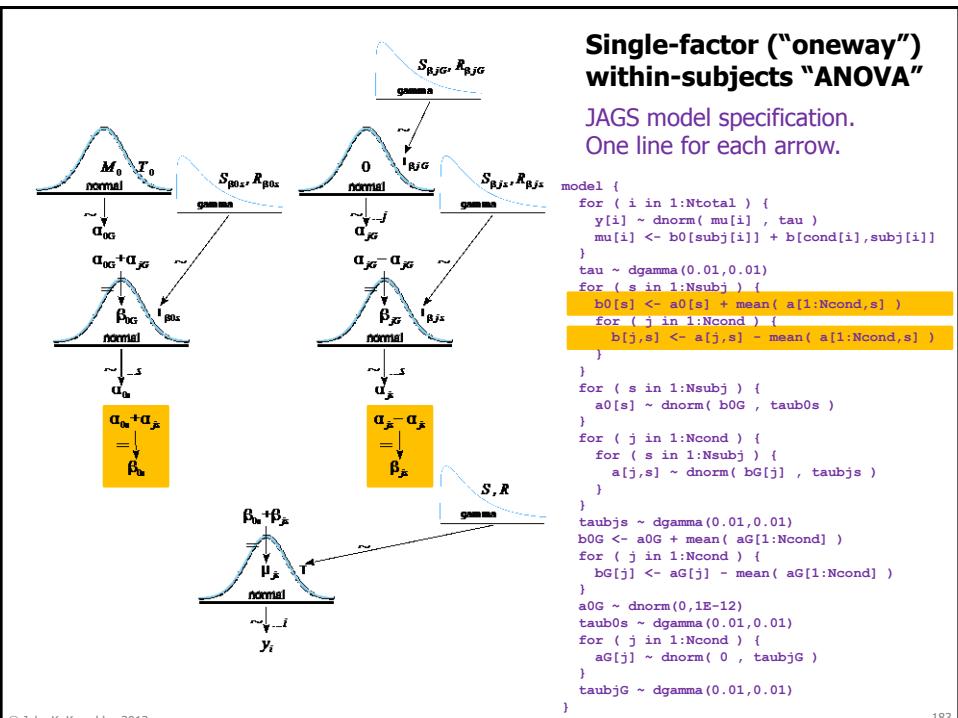
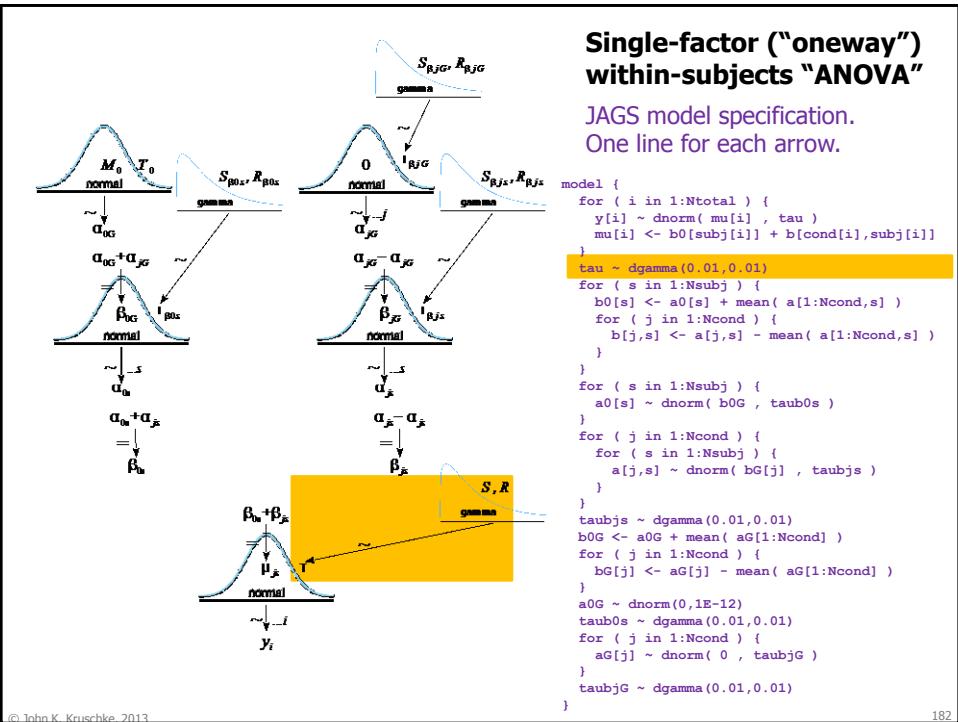
179

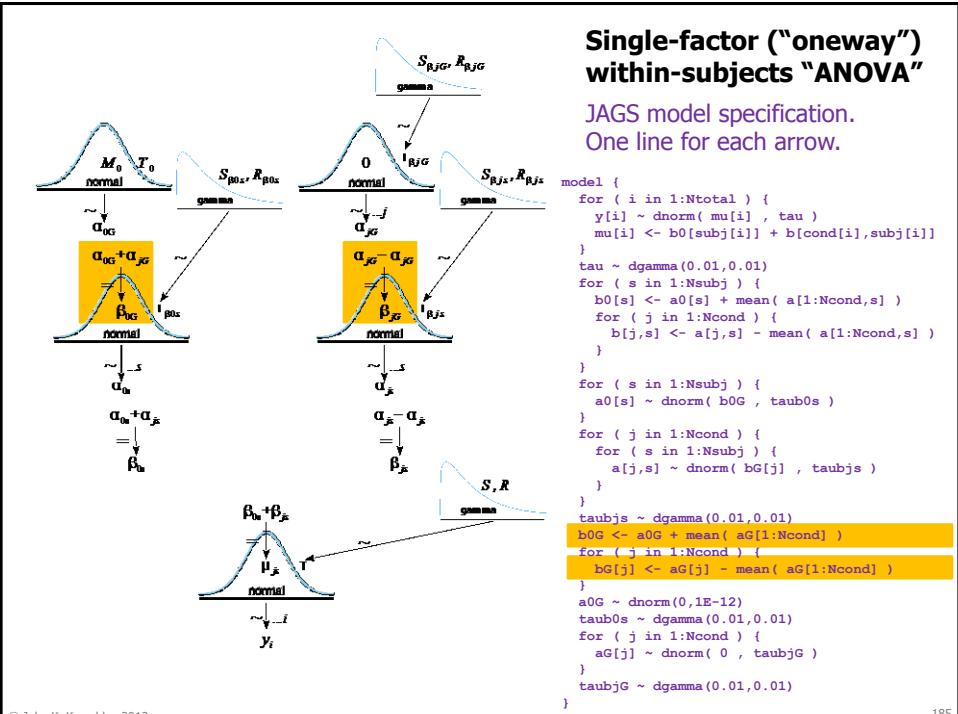
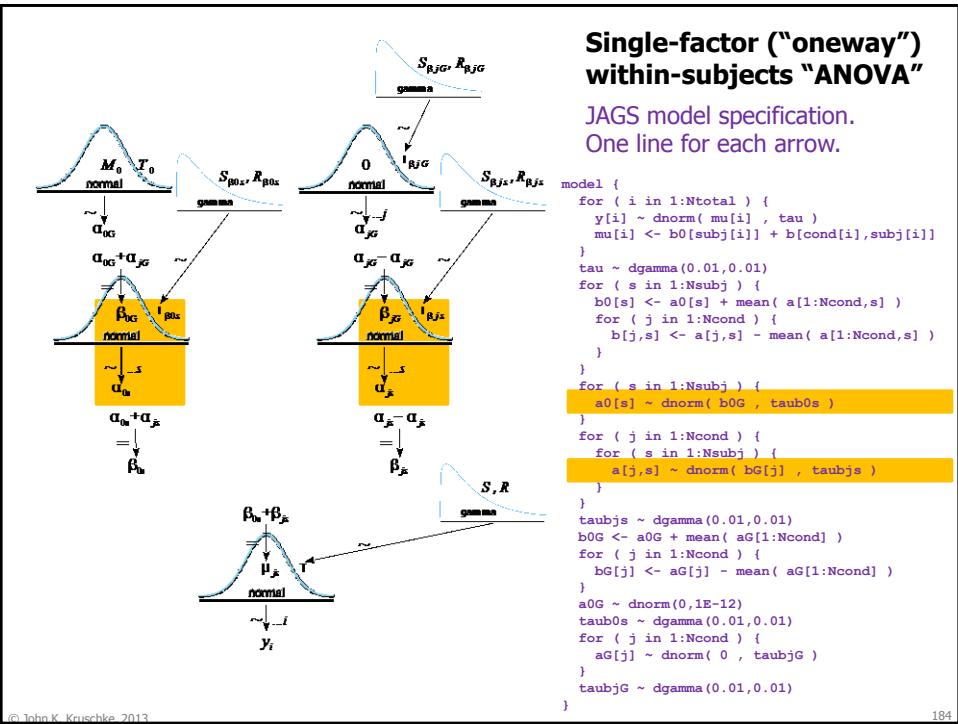


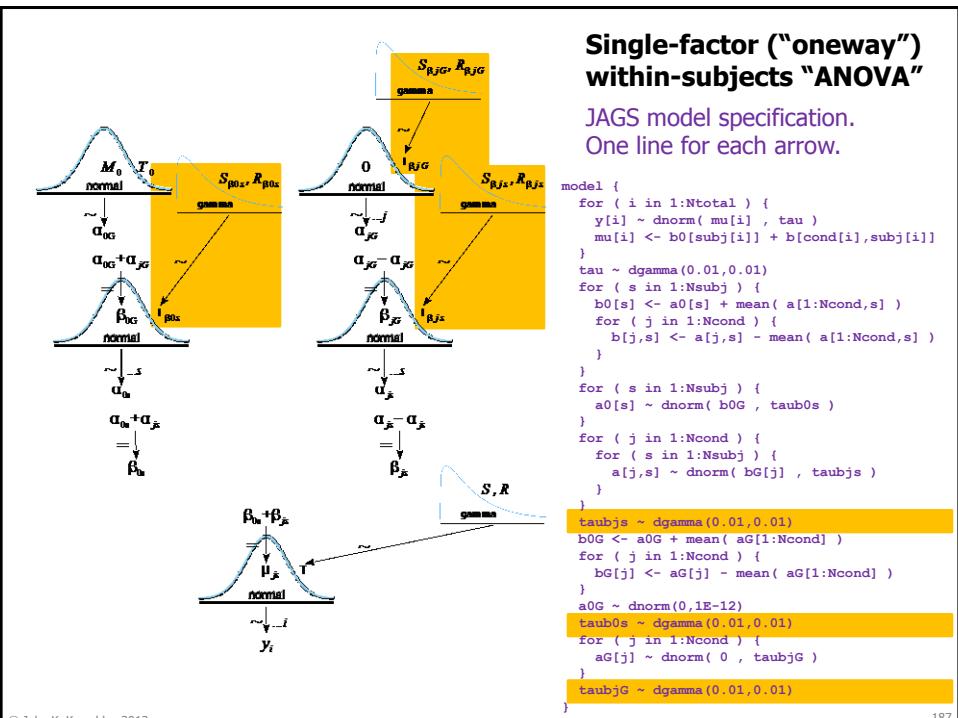
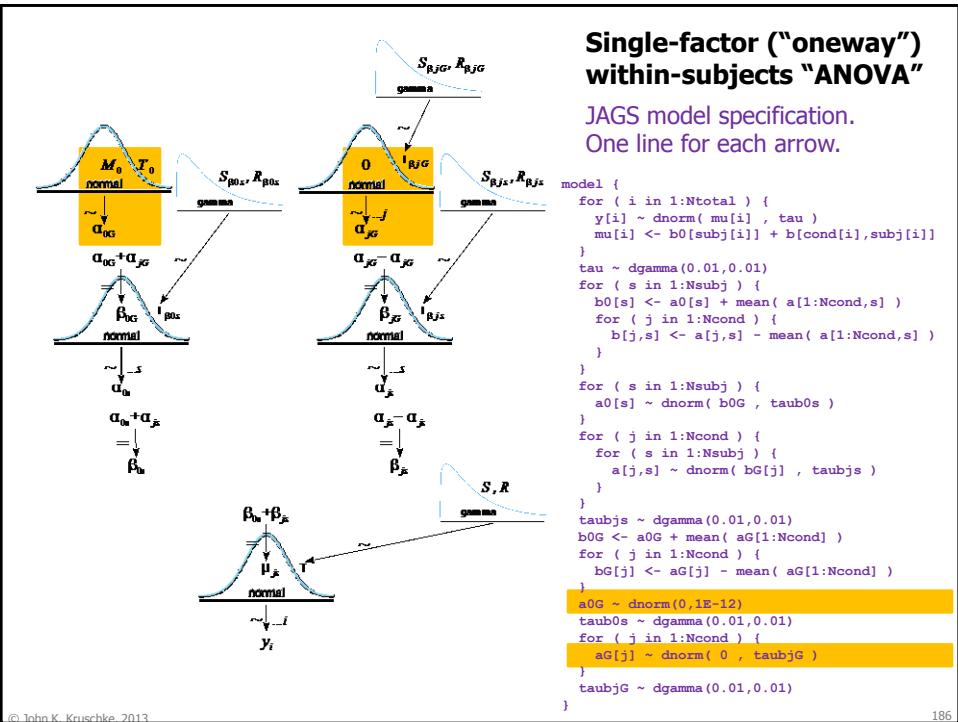
180

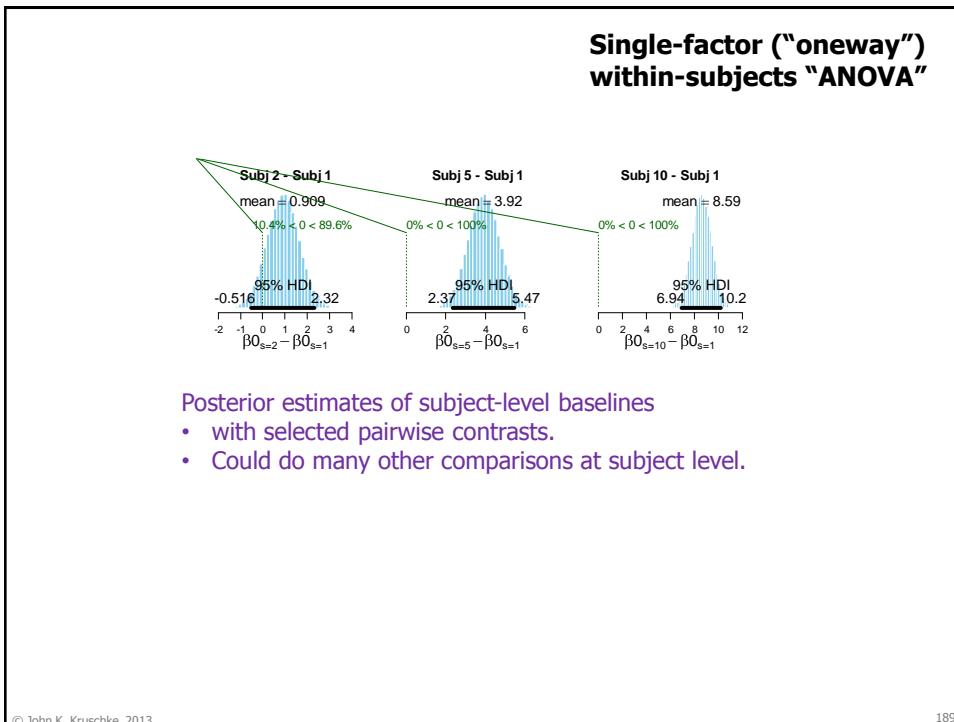
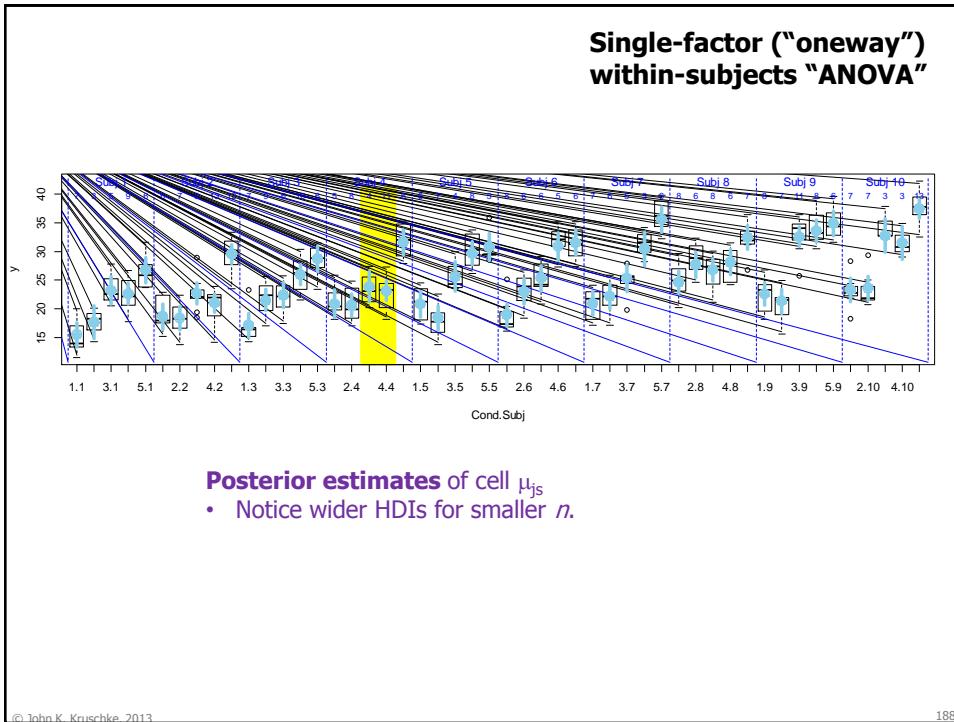


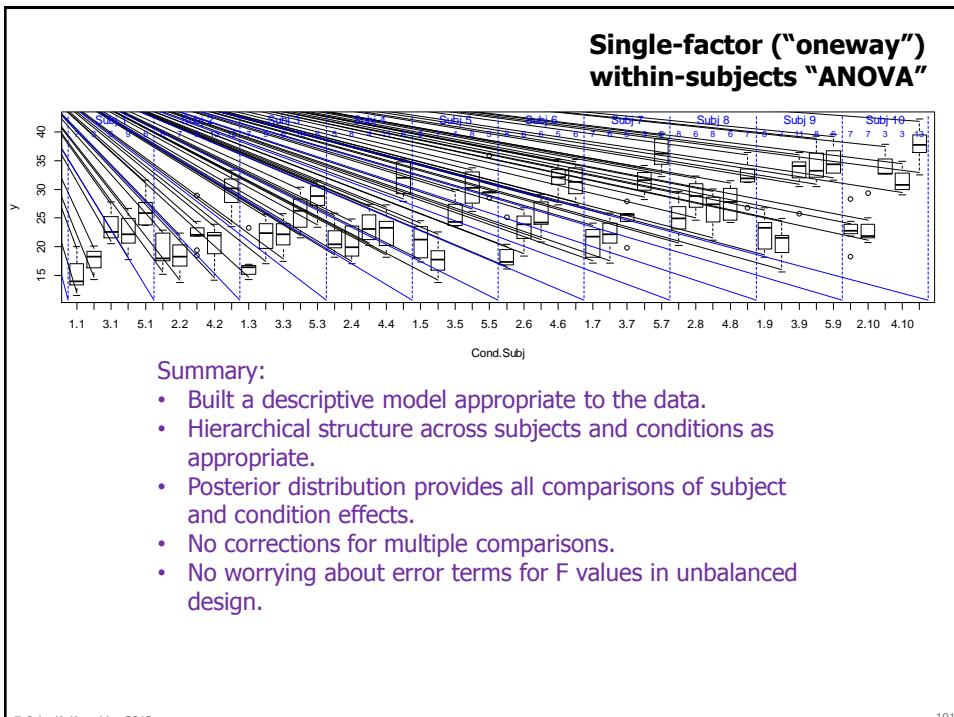
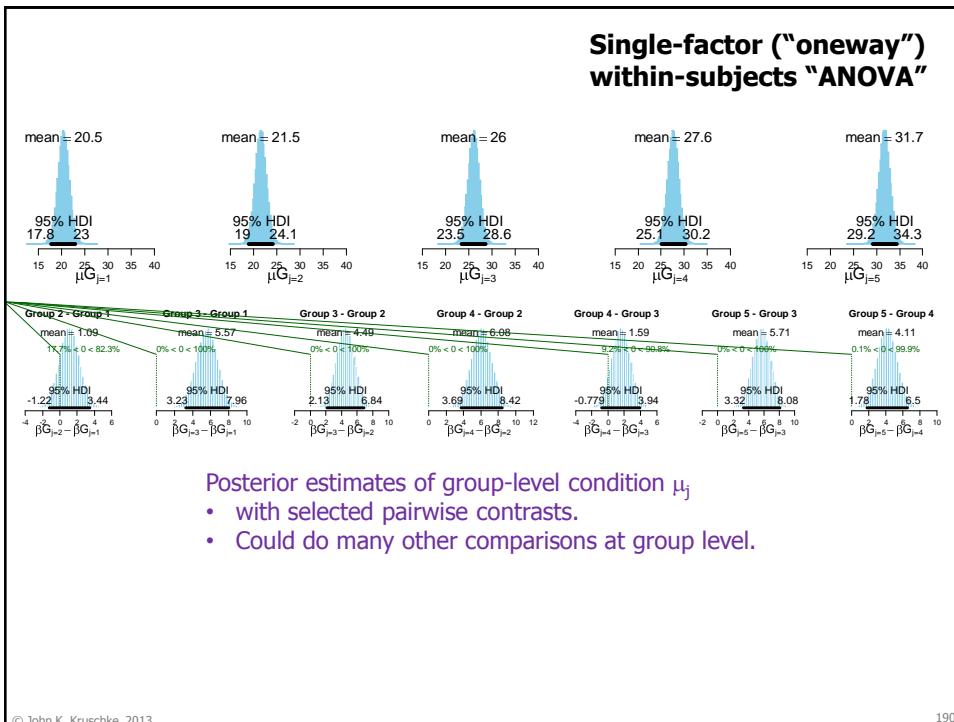
181











Split plot design

| Y | Subj | Angle | Age |
|-----|------|-----------|-----------|
| 450 | 1 | B1(Zero) | A1(Young) |
| 510 | 1 | B2(Four) | A1(Young) |
| 630 | 1 | B3(Eight) | A1(Young) |
| 390 | 2 | B1(Zero) | A1(Young) |
| 480 | 2 | B2(Four) | A1(Young) |
| 540 | 2 | B3(Eight) | A1(Young) |
| ... | ... | ... | ... |
| 510 | 10 | B1(Zero) | A1(Young) |
| 540 | 10 | B2(Four) | A1(Young) |
| 660 | 10 | B3(Eight) | A1(Young) |
| 420 | 11 | B1(Zero) | A2(Old) |
| 570 | 11 | B2(Four) | A2(Old) |
| 690 | 11 | B3(Eight) | A2(Old) |
| 600 | 12 | B1(Zero) | A2(Old) |
| 720 | 12 | B2(Four) | A2(Old) |
| 810 | 12 | B3(Eight) | A2(Old) |
| ... | ... | ... | ... |
| 510 | 20 | B1(Zero) | A2(Old) |
| 690 | 20 | B2(Four) | A2(Old) |
| 810 | 20 | B3(Eight) | A2(Old) |

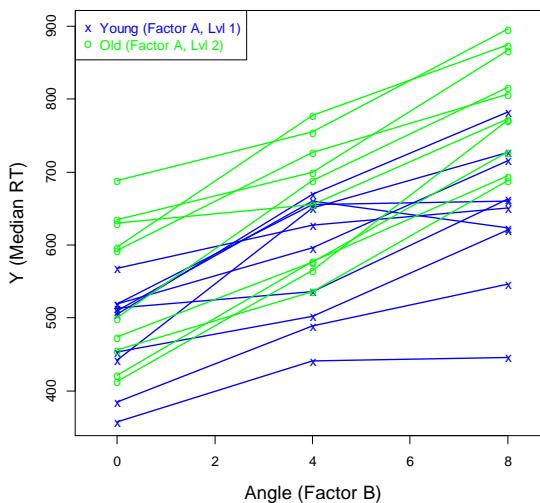
Consider an example provided by Maxwell & Delaney (2004, *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, 2nd Edition, Erlbaum; Ch. 12). A perceptual psychologist is studying response times for identifying letters flashed on a screen. The letters can be rotated off of vertical by zero degrees, four degrees, or eight degrees. Every subject responds many times to letters at each of the three angles. The experimenter (for unknown reasons) analyzes only the median response time of each subject at each angle. Thus, *each subject contributes only one datum at each level of angle (factor B)*. There are two types of subjects: "young" and "old." Age of subject is factor A. Y is the median response time, in milliseconds. There are 10 subjects per Age group.

(Bayesian analysis does not require or prefer equal subjects per group.)

<http://doingbayesiandataanalysis.blogspot.com/2012/05/split-plot-design-in-jags-revised.html>

Split plot design

| Y | Subj | Angle | Age |
|-----|------|-----------|-----------|
| 450 | 1 | B1(Zero) | A1(Young) |
| 510 | 1 | B2(Four) | A1(Young) |
| 630 | 1 | B3(Eight) | A1(Young) |
| 390 | 2 | B1(Zero) | A1(Young) |
| 480 | 2 | B2(Four) | A1(Young) |
| 540 | 2 | B3(Eight) | A1(Young) |
| ... | ... | ... | ... |
| 510 | 10 | B1(Zero) | A1(Young) |
| 540 | 10 | B2(Four) | A1(Young) |
| 660 | 10 | B3(Eight) | A1(Young) |
| 420 | 11 | B1(Zero) | A2(Old) |
| 570 | 11 | B2(Four) | A2(Old) |
| 690 | 11 | B3(Eight) | A2(Old) |
| 600 | 12 | B1(Zero) | A2(Old) |
| 720 | 12 | B2(Four) | A2(Old) |
| 810 | 12 | B3(Eight) | A2(Old) |
| ... | ... | ... | ... |
| 510 | 20 | B1(Zero) | A2(Old) |
| 690 | 20 | B2(Four) | A2(Old) |
| 810 | 20 | B3(Eight) | A2(Old) |



<http://doingbayesiandataanalysis.blogspot.com/2012/05/split-plot-design-in-jags-revised.html>

Split plot design

In classical ANOVA, the model analyzes the data variance into five components plus residual noise. The five components are:

- (1) the (between-subject) main effect, A,
- (2) the (within-subject) main effect, B,
- (3) the main effect of subject within levels of A, S/A,
- (4) the interaction AxB, and
- (5) the interaction of B with subjects within levels of A, BxS/A.

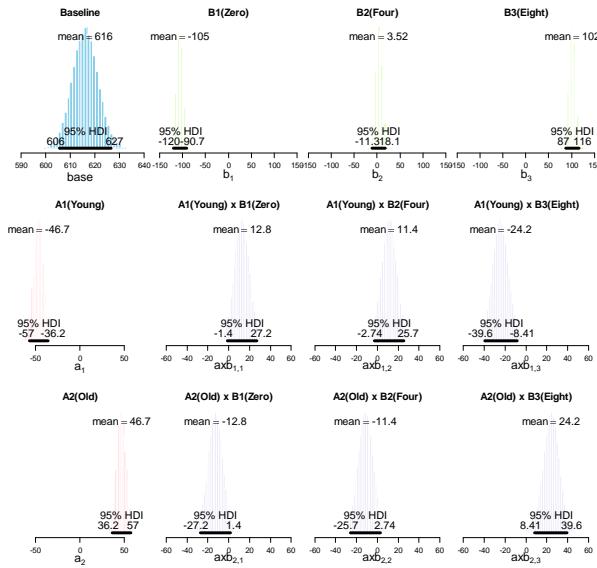
It turns out that the five components actually use up all the variance, so there is zero residual noise remaining. Another way of thinking about it is that *the BxS/A interaction term cannot be identified separately from the noise*, because there is only one measurement per cell of the design. That is also why the BxS/A term is used as the error term for some of the *F* ratios.

Therefore the model I've used in the Bayesian analysis does *not* include a BxS/A term.

Split plot design

```
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- base + a[aLvl[i]] + s[sLvl[i]] + b[bLvl[i]] + axb[aLvl[i],bLvl[i]]
    # The model has no BxS/A term because that would leave zero noise variance.
  }
  # Prior:
  tau <- pow( sigma , -2 )
  sigma ~ dunif(0,1000)
  base ~ dnorm(500,1.0E-6)
  for ( j in 1:NaLvl ) { a[j] ~ dnorm( 0.0 , aTau ) }
  aTau <- 1 / pow( aSD , 2 )
  aSD ~ dgamma(1.221,0.003683) # mode=60,sd=300. Change for scale of data.
  for ( j in 1:NbLvl ) { b[j] ~ dnorm( 0.0 , bTau ) }
  bTau <- 1 / pow( bSD , 2 )
  bSD ~ dgamma(1.221,0.003683)
  for ( j in 1:NsLvl ) { s[j] ~ dnorm( 0.0 , sTau ) }
  sTau <- 1 / pow( sSD , 2 )
  sSD ~ dgamma(1.221,0.003683) #
  for ( j in 1:NaLvl ) { for ( k in 1:NbLvl ) { axb[j,k] ~ dnorm( 0.0 , axbTau ) } }
  axbTau <- 1 / pow( axbSD , 2 )
  axbSD ~ dgamma(1.221,0.003683)
  # Conversion to sum-to-zero effects happens outside JAGS in R
}
```

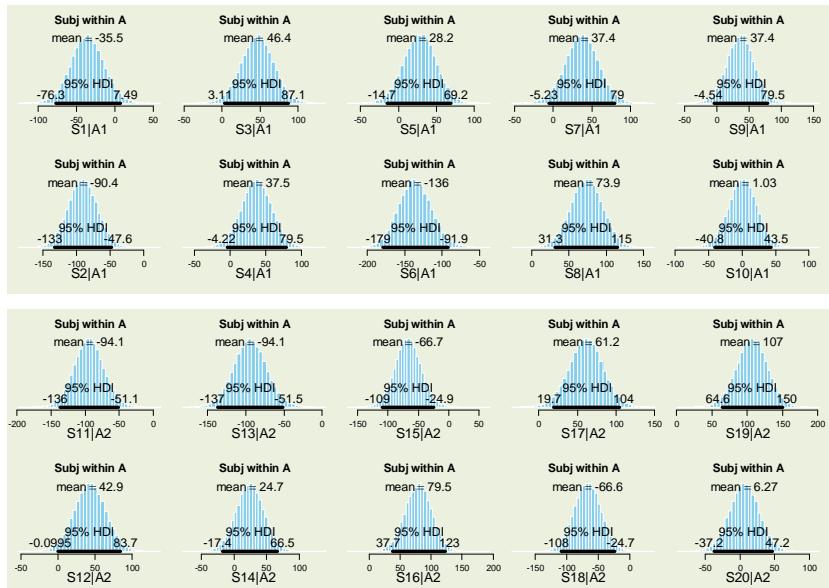
Split plot design



© John K. Kruschke, 2013

196

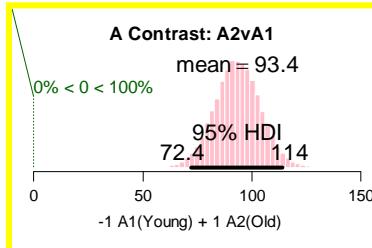
Split plot design



© John K. Kruschke, 2013

197

Split plot design



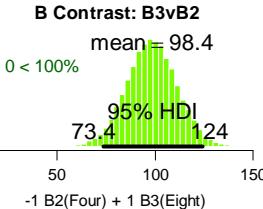
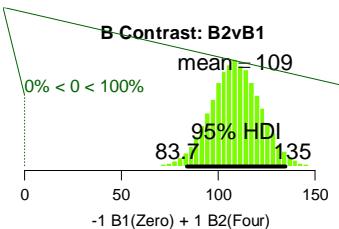
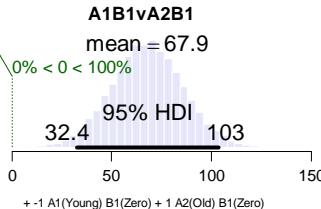
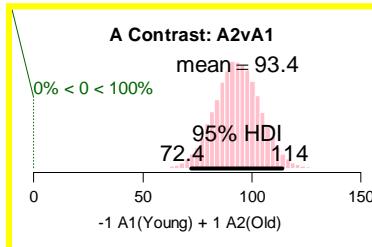
HDI is very precise compared to NHST confidence interval.

F ratio for this between-subject contrast uses larger denominator than other contrasts. Bayesian inference uses one joint posterior distribution for all decisions.

© John K. Kruschke, 2013

198

Split plot design



© John K. Kruschke, 2013

199

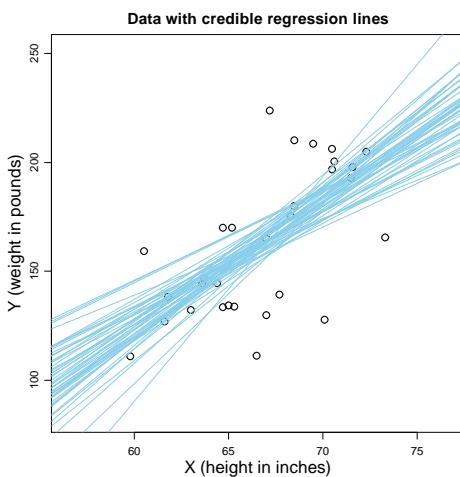
Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

© John K. Kruschke, 2013

200

Example: Linear Regression

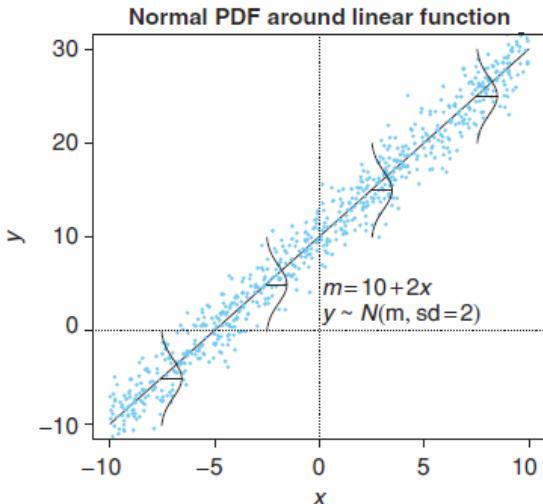


For every individual, there is a **predictor** value, x , and a **predicted** value, y .

© John K. Kruschke, 2013

201

Example of data generated from the model.



Three parameters:
intercept β_0 is 10,
slope β_1 is 2,
SD σ is 2.

Vertical scatter is
normally distributed,
with same SD for all x.

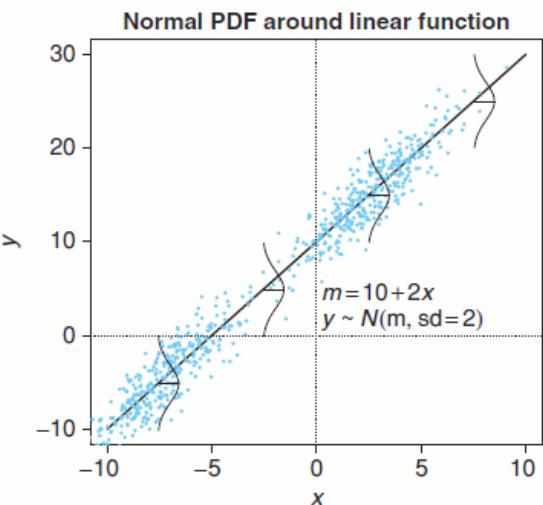
$$\begin{aligned} m &= \beta_0 + \beta_1 x \\ y &\sim N(m, \sigma) \end{aligned}$$

(Here x is uniformly
distributed; model says
nothing about
distribution of x.)

© John K. Kruschke, 2013

202

Example of data generated from the model.



Three parameters:
intercept β_0 is 10,
slope β_1 is 2,
SD σ is 2.

Vertical scatter is
normally distributed,
with same SD for all x.

$$\begin{aligned} m &= \beta_0 + \beta_1 x \\ y &\sim N(m, \sigma) \end{aligned}$$

(Here x is *bimodally*
distributed; model says
nothing about
distribution of x.)

© John K. Kruschke, 2013

203

Bayesian Linear Regression

Given the data D, what should we believe about β_0 , β_1 , and σ ? Bayesian answer:

$$p(\beta_0, \beta_1, \sigma | D) = p(D | \beta_0, \beta_1, \sigma) p(\beta_0, \beta_1, \sigma) / p(D)$$

↑ ↑ ↑

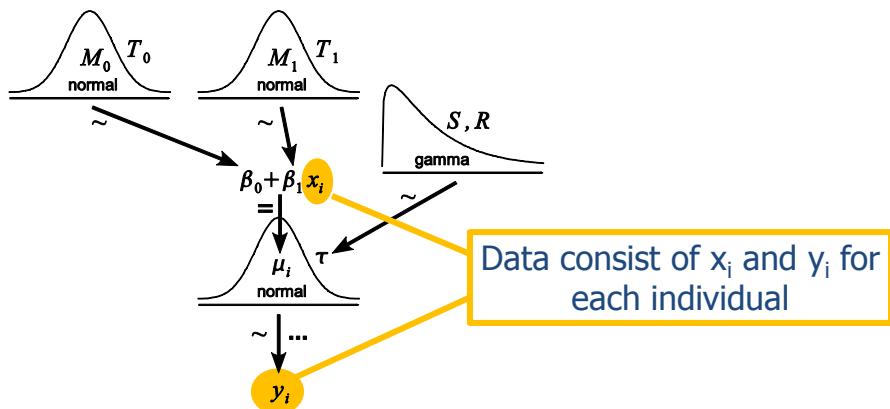
Posterior

linear regress.
likelihood
function

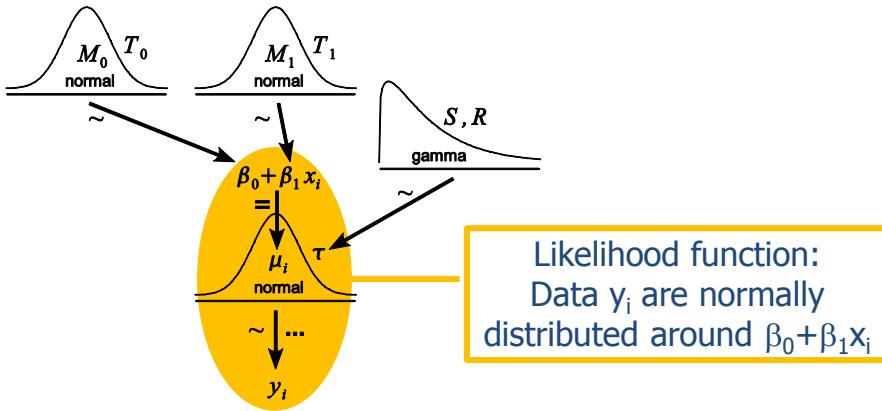
Prior

Bayesian inference is re-allocation of credibility.
Any estimated parameter has credibility
re-allocated from prior to posterior.

Bayesian Linear Regression



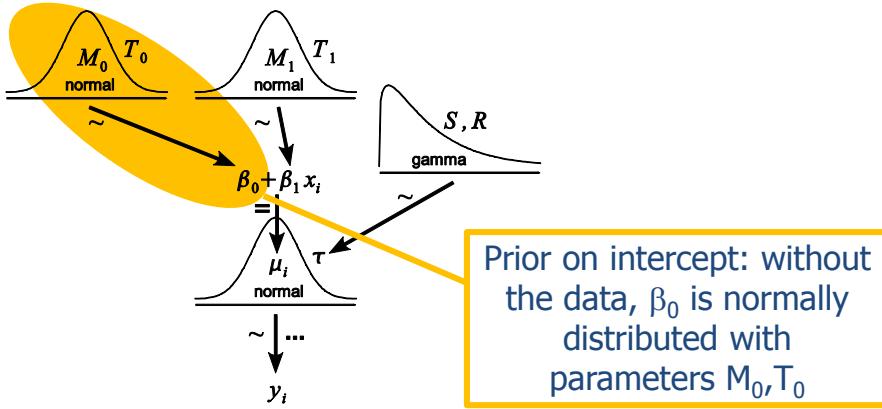
Bayesian Linear Regression



© John K. Kruschke, 2013

206

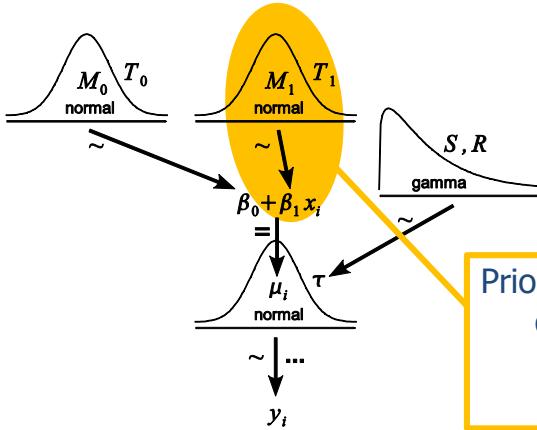
Bayesian Linear Regression



© John K. Kruschke, 2013

207

Bayesian Linear Regression

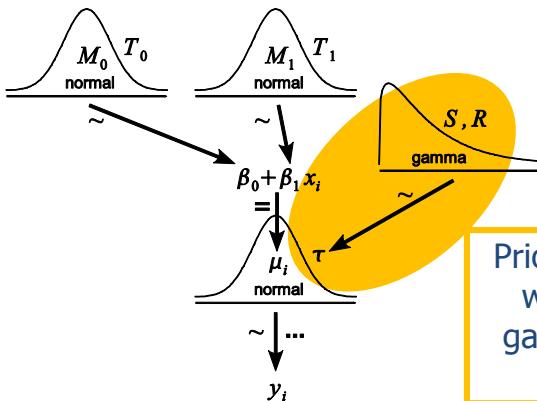


Prior on slope: without the data, β_1 is normally distributed with parameters M_1, T_1

© John K. Kruschke, 2013

208

Bayesian Linear Regression

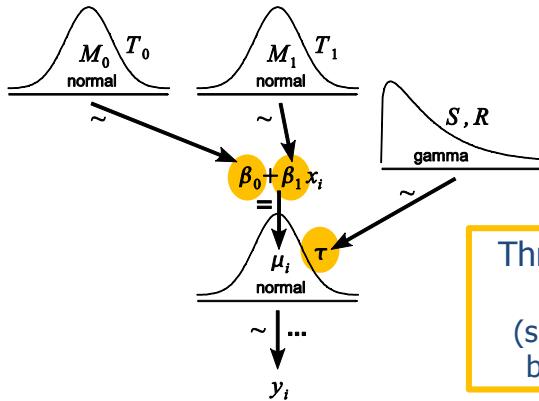


Prior on precision ($1/\sigma^2$): without the data, τ is gamma distributed with parameters S, R

© John K. Kruschke, 2013

209

Bayesian Linear Regression

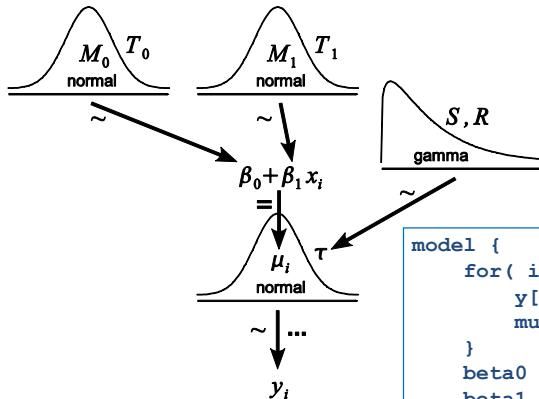


Three parameters with a
joint prior
(specified as independent
beliefs for convenience)

© John K. Kruschke, 2013

210

Bayesian Linear Regression in JAGS

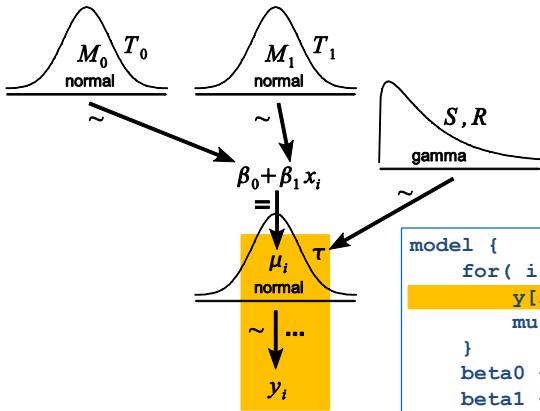


```
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + betal * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  betal ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}
```

© John K. Kruschke, 2013

211

Bayesian Linear Regression in JAGS

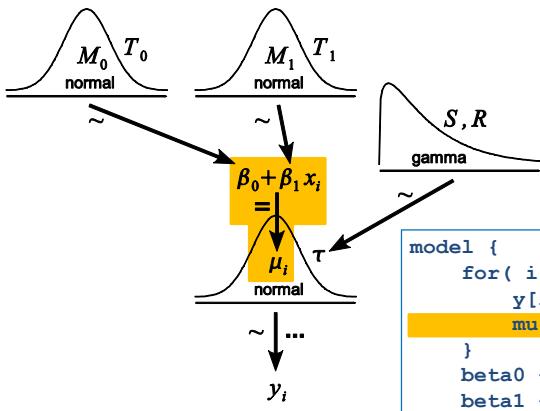


```
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + betal * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  betal ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}
```

© John K. Kruschke, 2013

212

Bayesian Linear Regression in JAGS

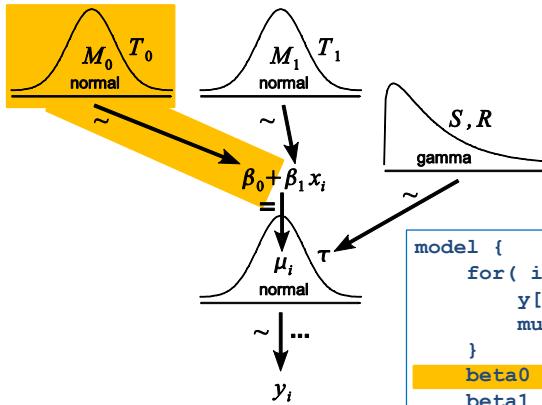


```
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + betal * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  betal ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}
```

© John K. Kruschke, 2013

213

Bayesian Linear Regression in JAGS



```

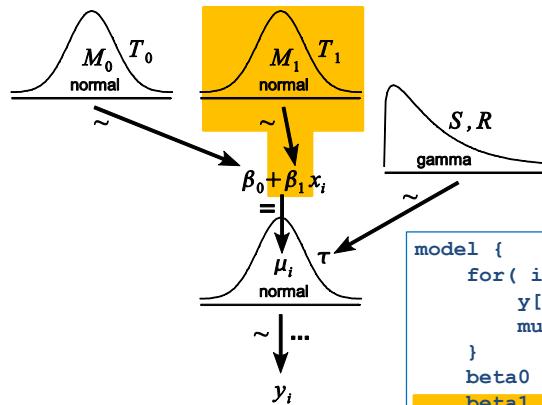
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + betal * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  betal ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}

```

© John K. Kruschke, 2013

214

Bayesian Linear Regression in JAGS



```

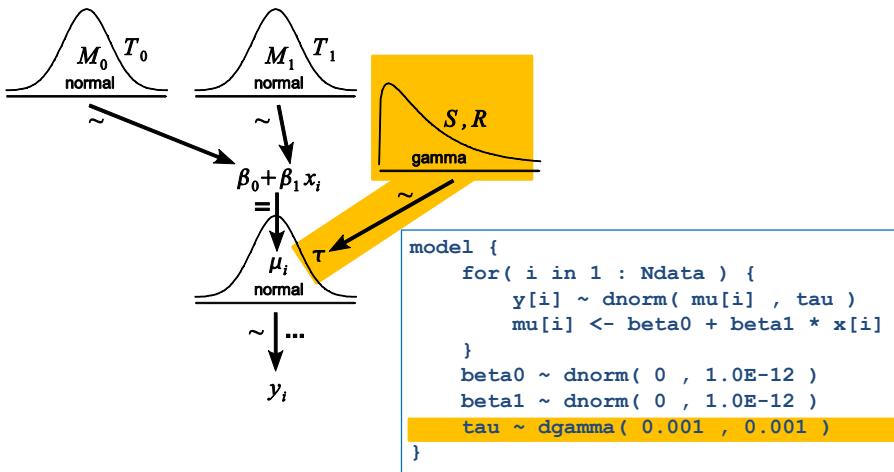
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + betal * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  betal ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}

```

© John K. Kruschke, 2013

215

Bayesian Linear Regression in JAGS



© John K. Kruschke, 2013

216

Program for Linear Regression

1. Specify model (we just did this).

2. Load data.

Standardize the data for MCMC efficiency.

3. Initialize the MCMC chain.

Use least-squares solution as start.

4. Run the MCMC chain.

5. Examine the results.

Transform back to original data scale.

© John K. Kruschke, 2013

217

Standardizing the data and transforming back to the original scale

$$z(x) = \frac{(x - M_x)}{SD_x} \quad \text{and} \quad z(y) = \frac{(y - M_y)}{SD_y} \quad (16.1)$$

Denote the intercept and slope for standardized data as ζ_0 and ζ_1 (Greek letter "zeta"), and denote the predicted value of y as \hat{y} . Then

$$\begin{aligned} z\hat{y} &= \zeta_0 + \zeta_1 z_x \quad \text{by definition of the model} \\ \frac{(\hat{y} - M_y)}{SD_y} &= \zeta_0 + \zeta_1 \frac{(x - M_x)}{SD_x} \quad \text{from Eqn. 16.1} \\ \hat{y} &= \underbrace{\zeta_0 SD_y + M_y}_{\beta_0} - \underbrace{\zeta_1 SD_y M_x / SD_x}_{\beta_1} + \underbrace{\zeta_1 SD_y / SD_x}_{} x \end{aligned} \quad (16.2)$$

Thus, for every believable combination of ζ_0, ζ_1 values, there is a corresponding believable combination of β_0, β_1 values specified by Equation 16.2.

Program for Linear Regression

- Invoke R from within Windows or WINE.
- Change directory to the folder in which you have your R programs saved:
File → Change Dir...

- Open the program

SimpleLinearRegressionJags.R:

File → Open Script...

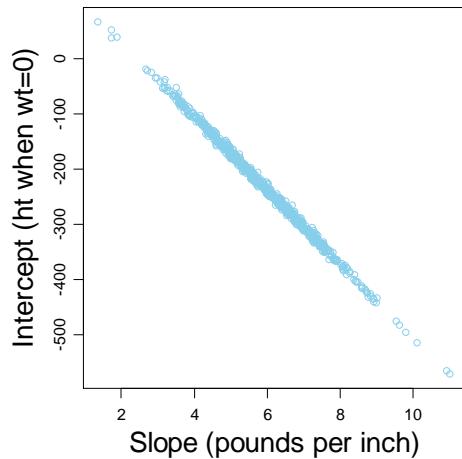
You will also need the file

HtWt30.csv in the same folder

and **plotChains.R**, **plotPost.R** and **HDIofMCMC.R**
in the same folder.

Linear Regression: Posterior Distribution

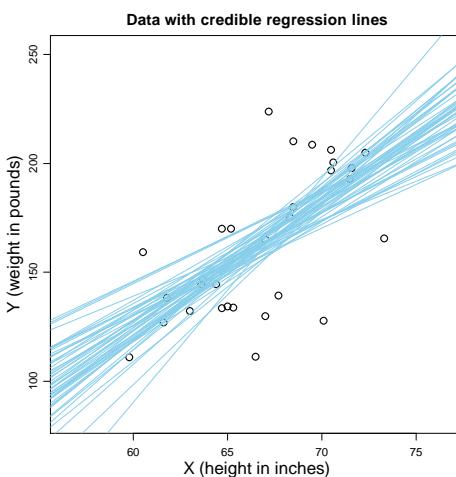
Posterior distribution is joint over *three* parameters, only two of which are shown here.



© John K. Kruschke, 2013

220

Posterior slopes and intercepts



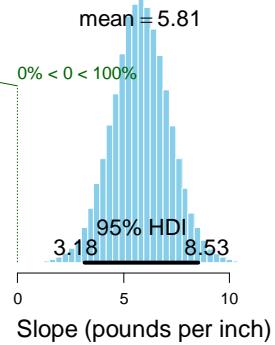
Each line also has a corresponding SD (precision), not shown.

© John K. Kruschke, 2013

221

Posterior of slope parameter

A slope of zero may be deemed not credible.

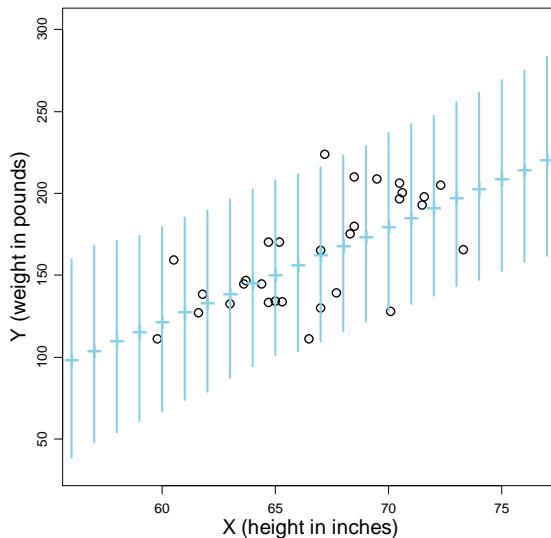


© John K. Kruschke, 2013

222

Posterior Predictions

Data with 95% HDI & Mean of Posterior Predictions



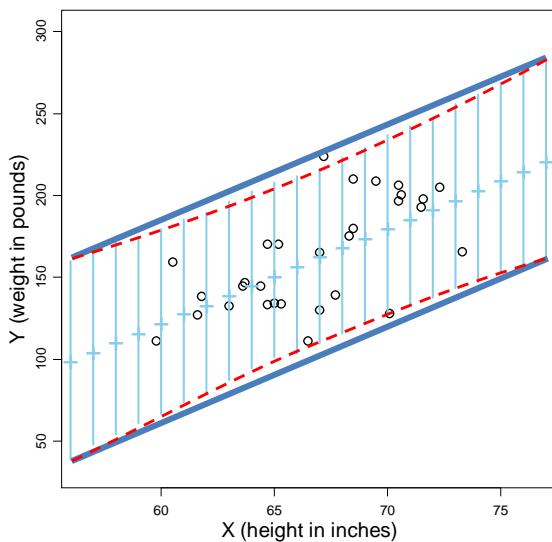
Vertical bars summarize simulated data as predicted from credible parameter values.

© John K. Kruschke, 2013

223

Posterior Predictions

Data with 95% HDI & Mean of Posterior Predictions

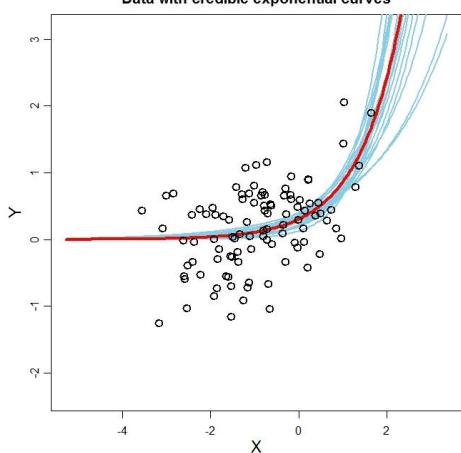


© John K. Kruschke, 2013

224

Exponential Regression

Data with credible exponential curves



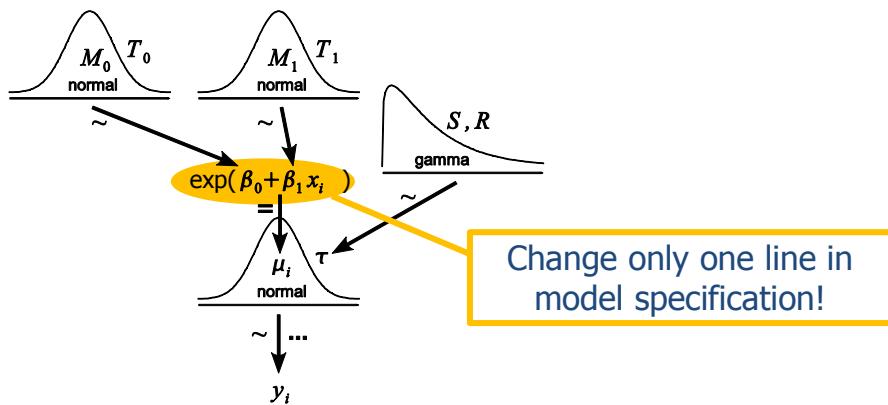
For every individual, there is a **predictor** variable, x , and a **predicted** variable, y .

Exponential regression models y as a random distribution around the curve $y_{\text{pred}} = \exp(\beta_1(x - \beta_0))$

© John K. Kruschke, 2013

225

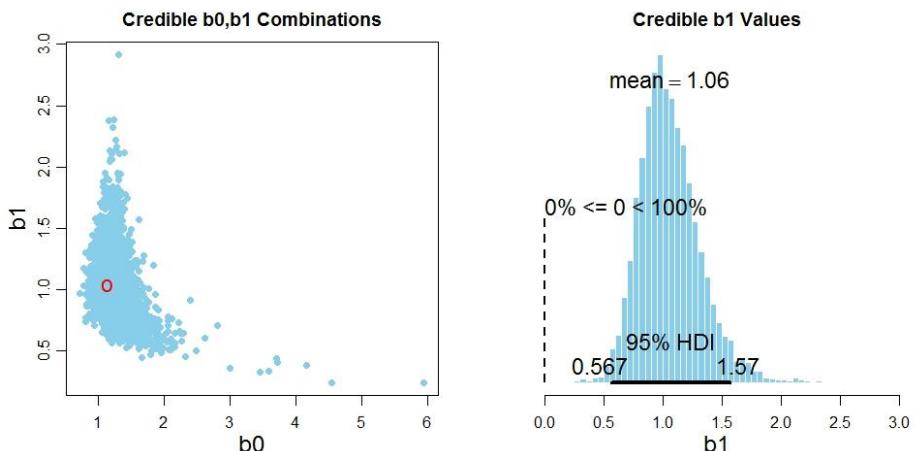
Bayesian Exponential Regression



© John K. Kruschke, 2013

226

Exponential Regression: Posterior Distrib.

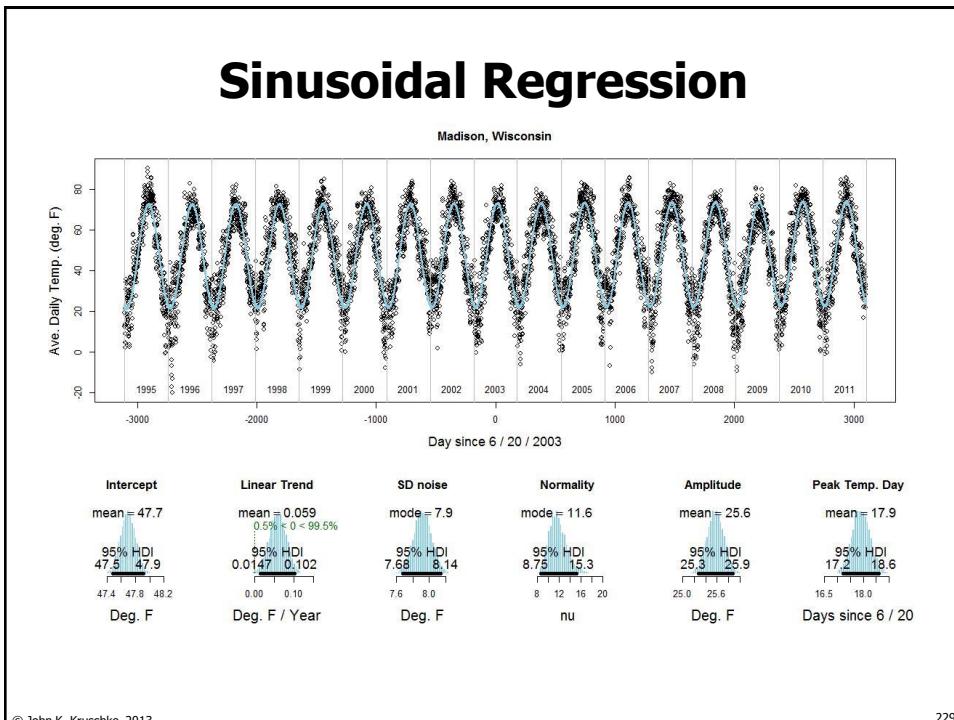
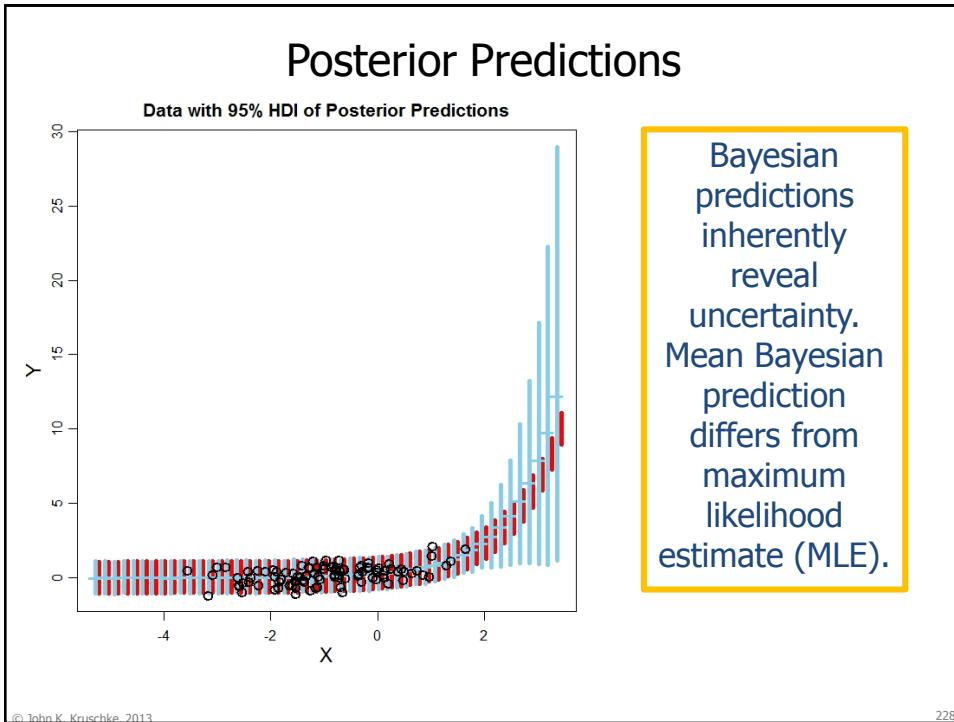


Curved trade-off between
values of b_0 and b_1 .
Not evident by MLE!

Easy to see that b_1 is non-zero. ***But not in NHST!***

© John K. Kruschke, 2013

227



Sinusoidal Regression

Madison, Wisconsin



```
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dt( mu[i] , tau , nu )
    mu[i] <- beta0+beta1*x[i] + amp*cos((x[i]-thresh)/wl)
  }
  beta0 ~ dnorm(0,1.0E-12)
  beta1 ~ dnorm(0,1.0E-12)
  tau ~ dgamma(0.001,0.001)
  amp ~ dunif(0,50)
  thresh ~ dunif(-183,183)
  nu <- nuMinusOne + 1
  nuMinusOne ~ dexp(1/29)
}
```

© John K. Kruschke, 2013

230

Sinusoidal Regression

Madison, Wisconsin



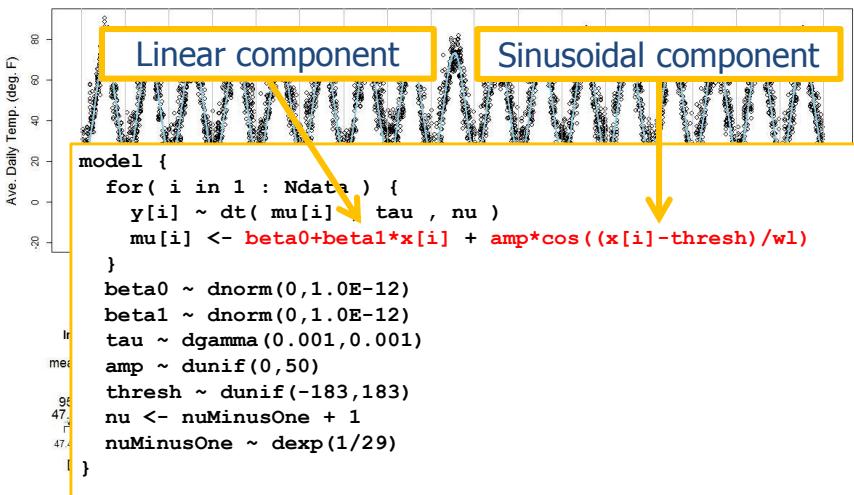
```
model {
  for( i in 1 : Ndata ) {
    y[i] ~ dt( mu[i] , tau , nu )
    mu[i] <- beta0+beta1*x[i] + amp*cos((x[i]-thresh)/wl)
  }
  beta0 ~ dnorm(0,1.0E-12)
  beta1 ~ dnorm(0,1.0E-12)
  tau ~ dgamma(0.001,0.001)
  amp ~ dunif(0,50)
  thresh ~ dunif(-183,183)
  nu <- nuMinusOne + 1
  nuMinusOne ~ dexp(1/29)
}
```

© John K. Kruschke, 2013

231

Sinusoidal Regression

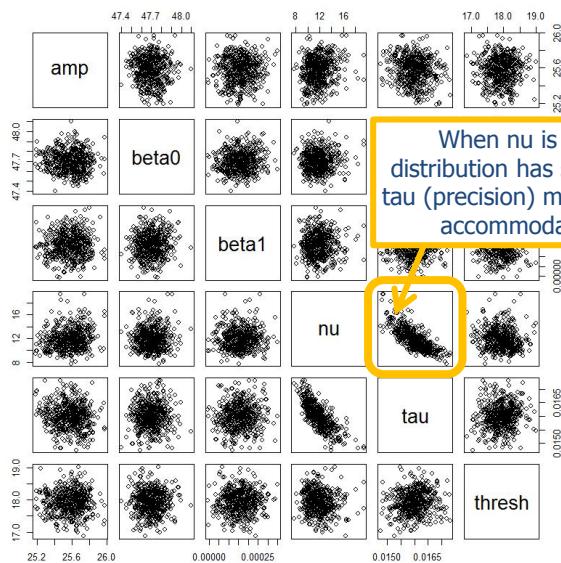
Madison, Wisconsin



© John K. Kruschke, 2013

232

Sinusoidal Regression

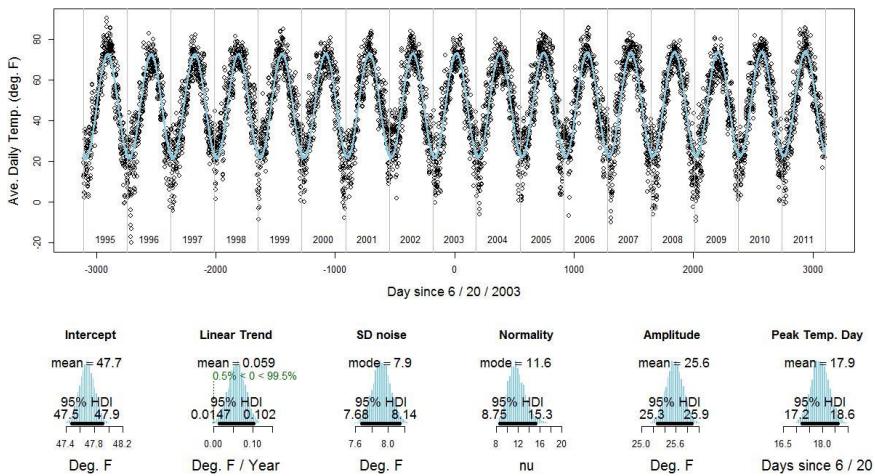


© John K. Kruschke, 2013

233

Sinusoidal Regression

Madison, Wisconsin

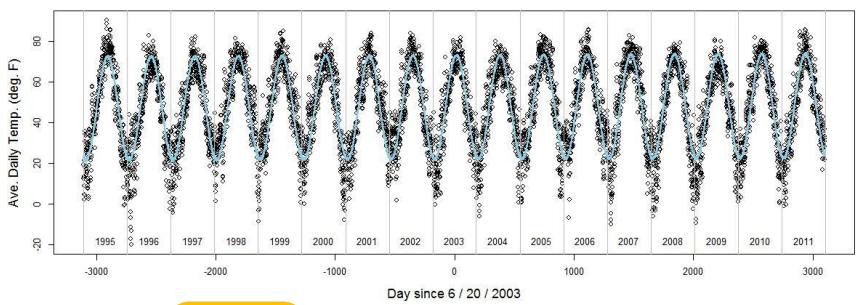


© John K. Kruschke, 2013

234

Sinusoidal Regression

Madison, Wisconsin

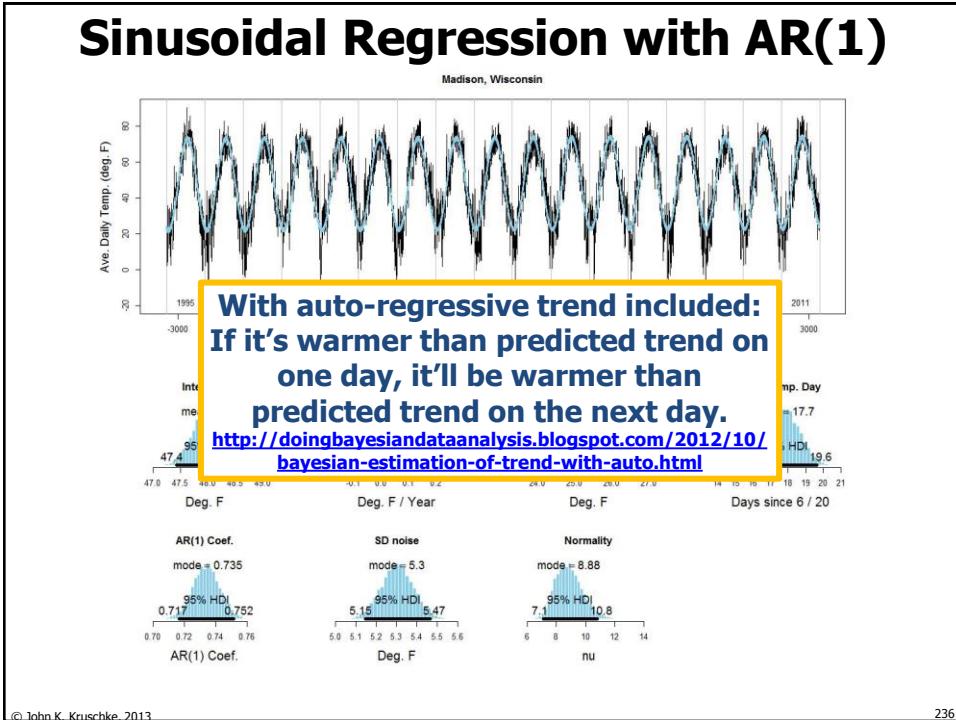


It's getting warmer!
Discussion at
<http://doingbayesiandataanalysis.blogspot.com/2012/07/its-getting-warmer-in-wisconsin.html>

© John K. Kruschke, 2013

235

Sinusoidal Regression with AR(1)



Sinusoidal Regression with AR(1)

```
model {
  trend[1] <- beta0 + beta1*x[1] + amp*cos((x[1]-thresh)/wl)
  for( i in 2 : Ndata ) {
    y[i] ~ dt( mu[i] , tau , nu )
    mu[i] <- trend[i] + ar1 * ( y[i-1] - trend[i-1] )
    trend[i] <- beta0 + beta1*x[i] + amp*cos((x[i]-thresh)/wl)
  }
  ar1 ~ dunif(-1.1,1.1)
  beta0 ~ dnorm(0,1.0E-12)
  beta1 ~ dnorm(0,1.0E-12)
  tau ~ dgamma( 0.001 , 0.001 )
  amp ~ dunif(0,50)
  thresh ~ dunif(-183,183)
  nu <- nuMinusOne + 1
  nuMinusOne ~ dexp(1/29)
}
```

Trend is linear plus sinusoid,
as before...

Sinusoidal Regression with AR(1)

```

model {
  trend[1] <- beta0 + beta1*x[1] + amp*cos((x[1]-thresh)/wl)
  for( i in 2 : Ndata ) {
    y[i] ~ dt( mu[i] , tau , nu )
    mu[i] <- trend[i] + ar1 * ( y[i-1] - trend[i-1] )
    trend[i] <- beta0 + beta1*x[i] + amp*cos((x[i]-thresh)/wl)
  }
  ar1 ~ dunif(-1.1,1.1)
  beta0 ~ dnorm(0,1.0E-12)
  beta1 ~ dnorm(0,1.0E-12)
  tau ~ dgamma( 0.001 , 0.001 )
  amp ~ dunif(0,50)
  thresh ~ dunif(-183,183)
  nu <- nuMinusOne + 1
  nuMinusOne ~ dexp(1/29)
}

```

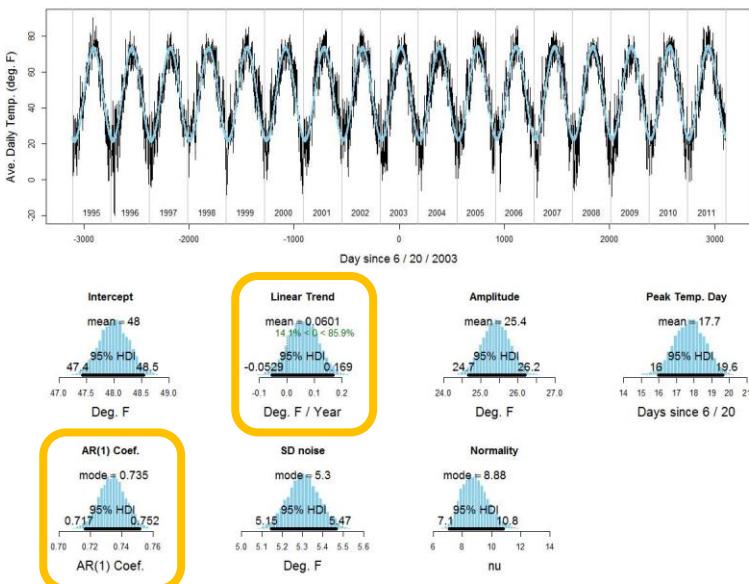
But predicted value (μ) is trend *plus* proportion of deviation of data from trend at previous time. This is the AR(1) part of the model.

© John K. Kruschke, 2013

238

Sinusoidal Regression with AR(1)

Madison, Wisconsin



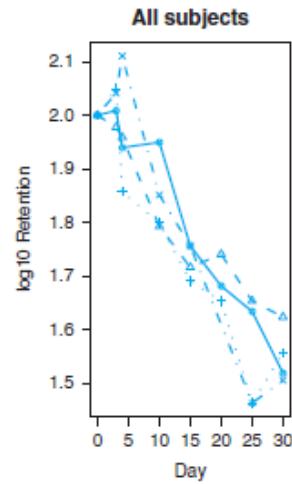
© John K. Kruschke, 2013

239

Hierarchical regression: Individual and group levels

As another concrete example of this situation, consider an experiment in which the investigators were interested in how quickly different organs clear themselves of contaminants (Feldman, 1988, reported data from an unpublished experiment by S. B. Weinstock and J. D. Brain). The researchers administered iron oxide particles to rats, because the iron oxide remaining in the body could be assayed noninvasively via magnetometry. Four rats were given intravenous injection of iron oxide, the particles of which are taken up by liver endothelial cells. Four other rats were given the iron oxide by tracheal instillation, so that the particles were taken up by lung macrophages. Although the researchers were specifically interested in comparing the clearance rates of the two groups, we will consider only the "lung" group.

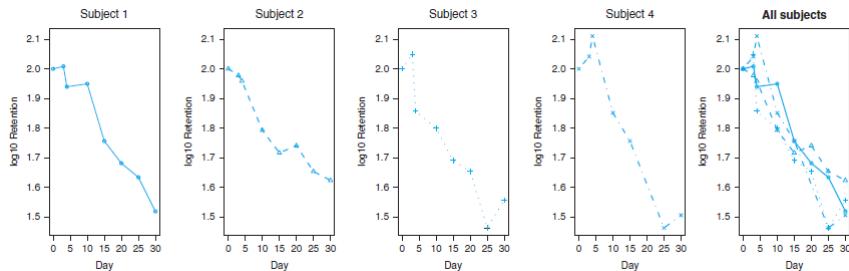
The amount of iron oxide retained in the body, as a percentage of the initially assayed amount, was measured at various times during the following 30 days. The data for the lung group are shown in the top row of Figure 16.10. The retention amount is plotted on a logarithmic scale, so that the retention curves are approximately linear. Notice that all the curves start at a y value of 2.0, because the first measurement establishes the baseline that defines 100%, and $\log_{10}(100) = 2.0$. Some curves rise above the initial value, which presumably does not indicate spontaneous creation of iron, but instead indicates measurement noise, either at the initial or subsequent times. The graphs indicate a reduction through time. The goal of our analysis is to determine what the believable rates of reduction are, given the data. We may also want to know if the apparent reduction really is believably nonzero.



© John K. Kruschke, 2013

240

Hierarchical regression: Individual and group levels



© John K. Kruschke, 2013

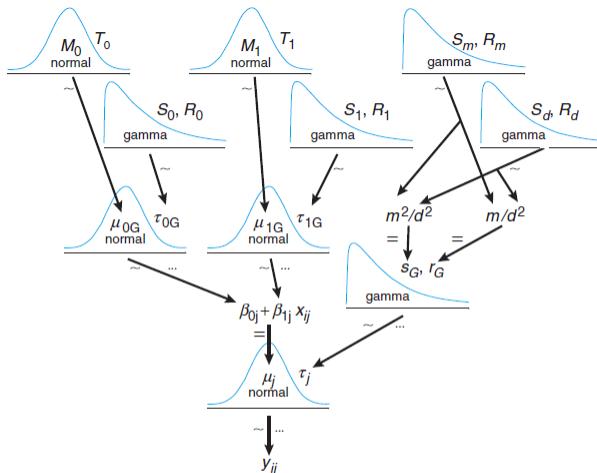
241

Hierarchical regression: Individual and group levels

To model this situation, each subject's data set is described by an individual linear regression, and the regression coefficients of the individuals are, in turn, modeled by group-level distributions. The group-level distributions are controlled by parameters that describe the central tendency and variability of the group, and it is these group-level parameters in which we are primarily interested.

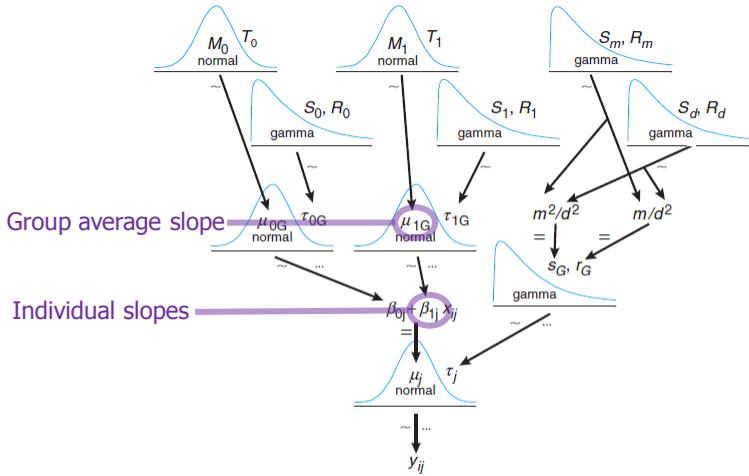
Figure 16.11 shows the hierarchy of dependencies. At the lowest level, we just see a simple linear regression for each individual, with the same components as Figure 16.3. The regression coefficients for the j^{th} individual, namely, β_{0j} , β_{1j} , and τ_j , in turn come from distributions that describe the group. For example, the individual slope coefficients, β_{1j} , are assumed to come from a normal distribution, with mean μ_{1G} and precision τ_{1G} . We are interested in estimating those group-level parameters; therefore each is given a prior distribution at the top level of the diagram.

Hierarchical regression: Individual and group levels

**FIGURE 16.11**

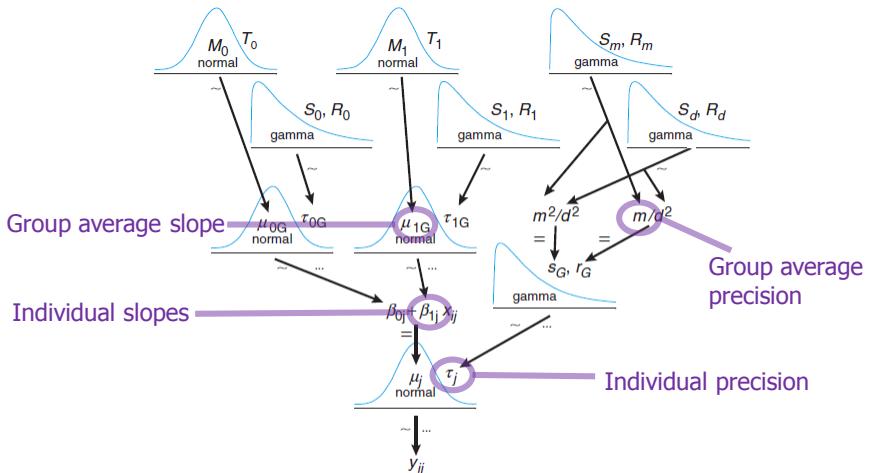
A model of dependencies for repeated scores from N subjects drawn independently from the same group. The slope for the j^{th} subject is β_{1j} . Across subjects, it is distributed normally, with mean μ_{1G} and precision τ_{1G} (i.e., across subjects the variance of the slopes is $1/\tau_{1G}$).

Hierarchical regression: Individual and group levels

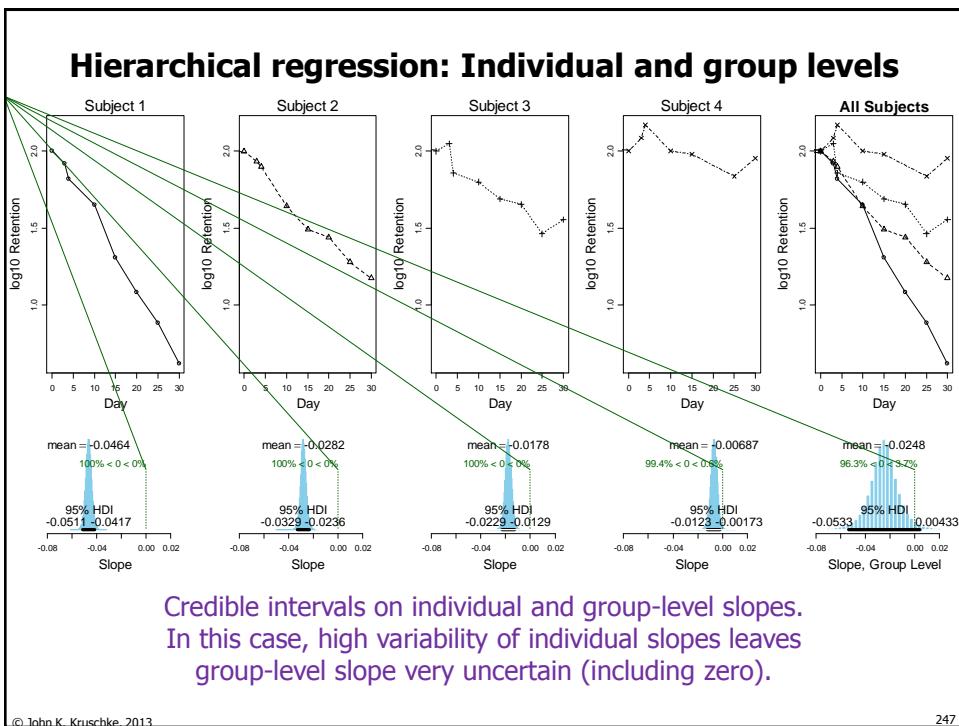
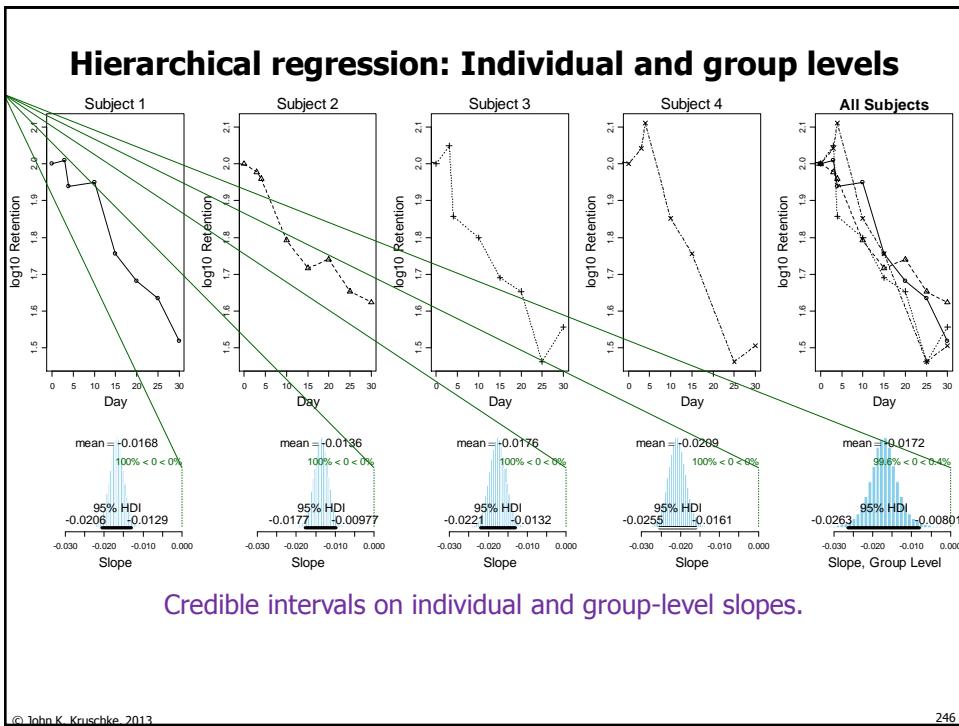
**FIGURE 16.11**

A model of dependencies for repeated scores from N subjects drawn independently from the same group. The slope for the j^{th} subject is β_{1j} . Across subjects, it is distributed normally, with mean μ_{1G} and precision τ_{1G} (i.e., across subjects the variance of the slopes is $1/\tau_{1G}$).

Hierarchical regression: Individual and group levels

**FIGURE 16.11**

A model of dependencies for repeated scores from N subjects drawn independently from the same group. The slope for the j^{th} subject is β_{1j} . Across subjects, it is distributed normally, with mean μ_{1G} and precision τ_{1G} (i.e., across subjects the variance of the slopes is $1/\tau_{1G}$).



Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

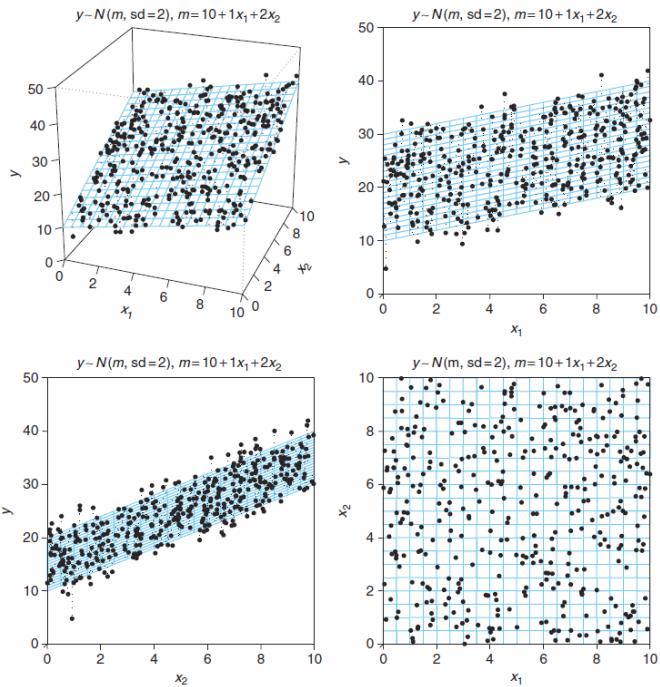
© John K. Kruschke, 2013

248

Multiple Linear Regression

$$\mu = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$y \sim \text{normal}(\mu, \sigma)$$

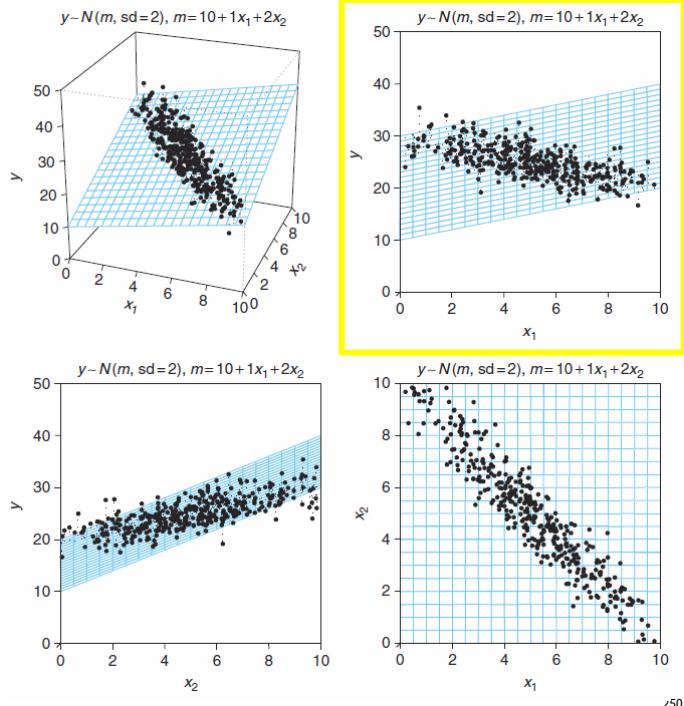


© John K. Kruschke, 2013

249

Multiple Linear Regression

Perils of correlated predictors



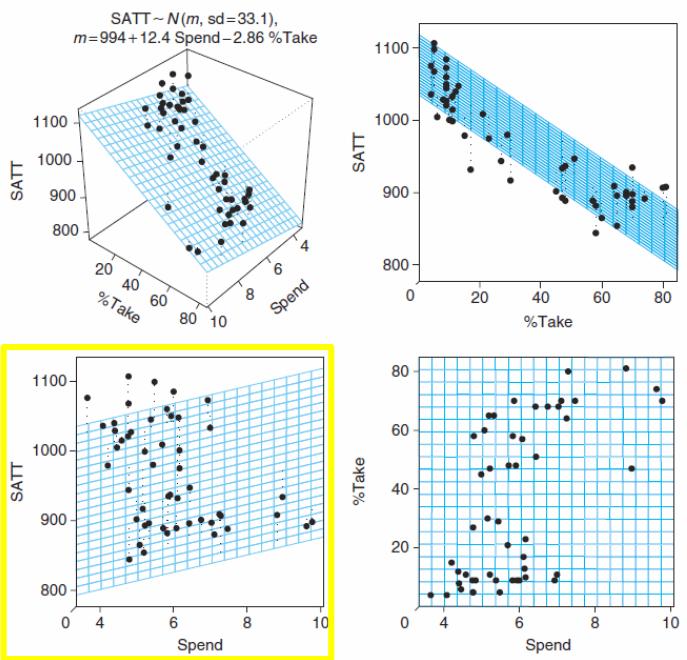
© John K. Kruschke, 2013

250

Predicted: SAT score.

Predictors:
Spending per pupil and
% of students taking the exam.

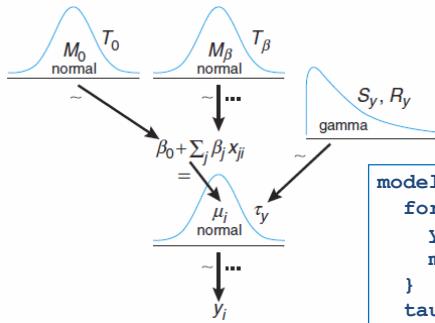
Data from Guber (1999)



© John K. Kruschke, 2013

251

Bayesian Multiple Linear Regression in JAGS



```
model {
  for( i in 1 : nData ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- b0 + inprod( b[] , x[i,] )
  }
  tau ~ dgamma(.01,.01)
  b0 ~ dnorm(0,1.0E-12)
  for ( j in 1:nPredictors ) {
    b[j] ~ dnorm(0,1.0E-12)
  }
}
```

Program for Multiple Linear Regression

1. Specify model (we just did this).

2. Load data.

Standardize the data for MCMC efficiency.

3. Initialize the MCMC chain.

Use least-squares solution as start.

4. Run the MCMC chain.

5. Examine the results.

Transform back to original data scale.

Standardizing the data and transforming back to the original scale

$$z_{\hat{y}} = \zeta_0 + \sum_j \zeta_j z_{x_j}$$

$$\frac{(\hat{y} - M_y)}{SD_y} = \zeta_0 + \sum_j \zeta_j \frac{(x_j - M_{x_j})}{SD_{x_j}}$$

$$\hat{y} = \underbrace{\zeta_0 SD_y + M_y - \sum_j \zeta_j SD_y M_{x_j} / SD_{x_j}}_{\beta_0} + \sum_j \underbrace{\zeta_j SD_y / SD_{x_j}}_{\beta_j} x_j$$

The estimate of σ_y is merely $\sigma_{z_y} SD_y$.

Initializing the MCMC chain at “best” values

```
genInitList <- function( nPred=nPredictors ) {
  lmInfo = lm( dataList$y ~ dataList$x )
  bInit = lmInfo$coef[-1]
  tauInit = length(dataList$y) / sum(lmInfo$res^2)
  list(
    b0 = 0 , # could be b0 = lmInfo$coef[1]
    b = bInit ,
    tau = tauInit
  )
}
for ( chainIdx in 1 : nChain ) {
  modelInits( bugsInits( genInitList ) )
}
```

Program for Multiple Linear Regression

MultipleLinearRegressionJags.R

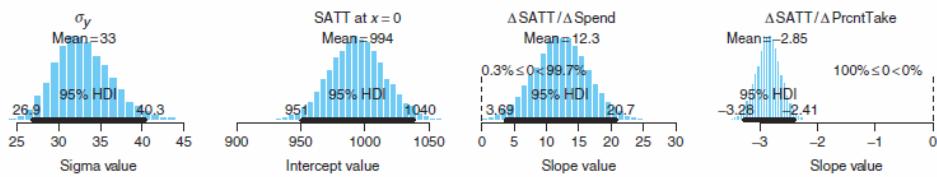
Specify

```
dataSource = c("Guber1999", "McIntyre1994", "random") [1]
```

© John K. Kruschke, 2013

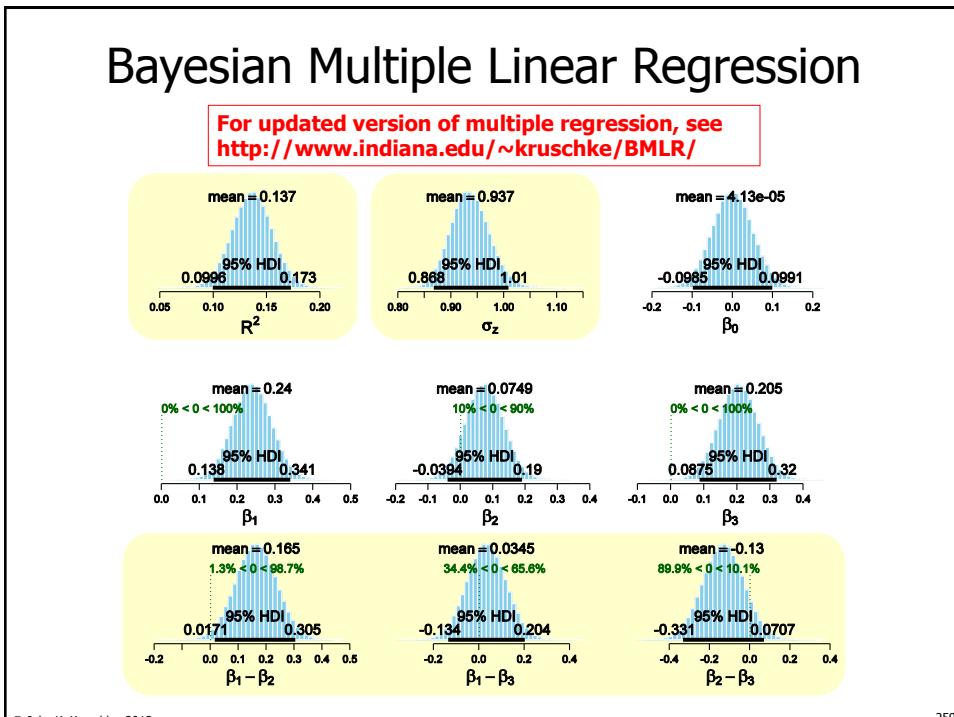
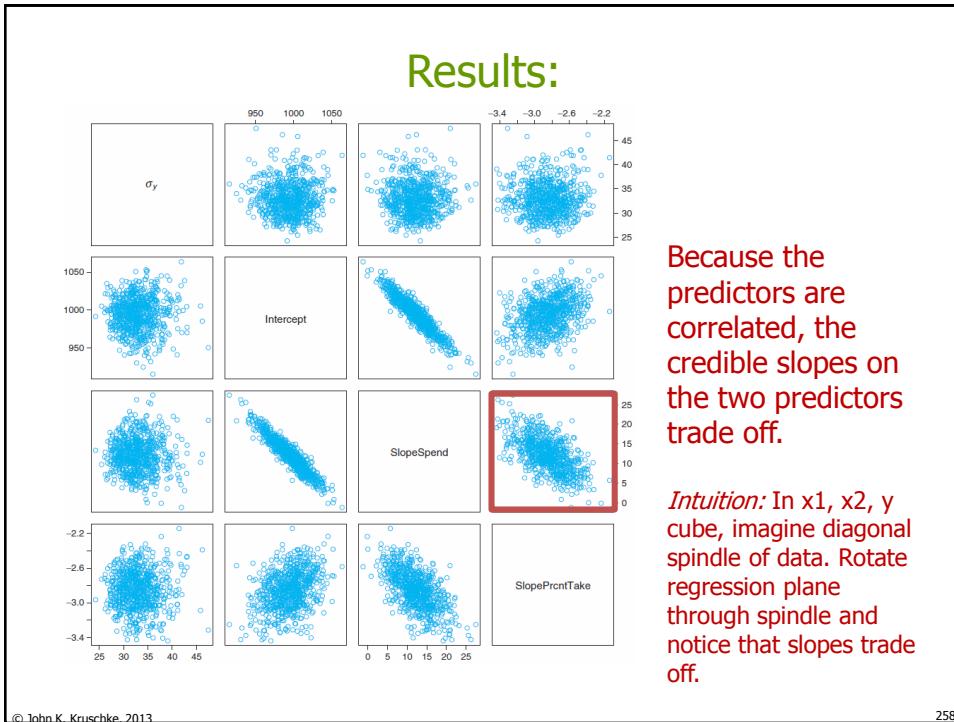
256

Results:

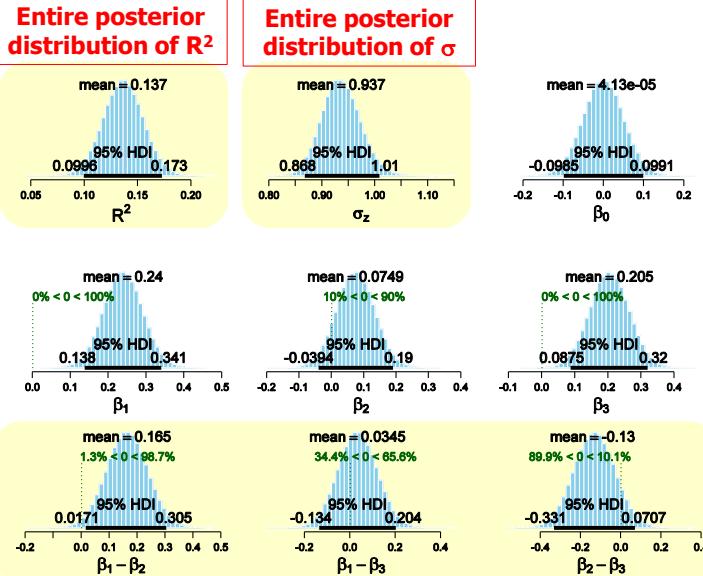


© John K. Kruschke, 2013

257



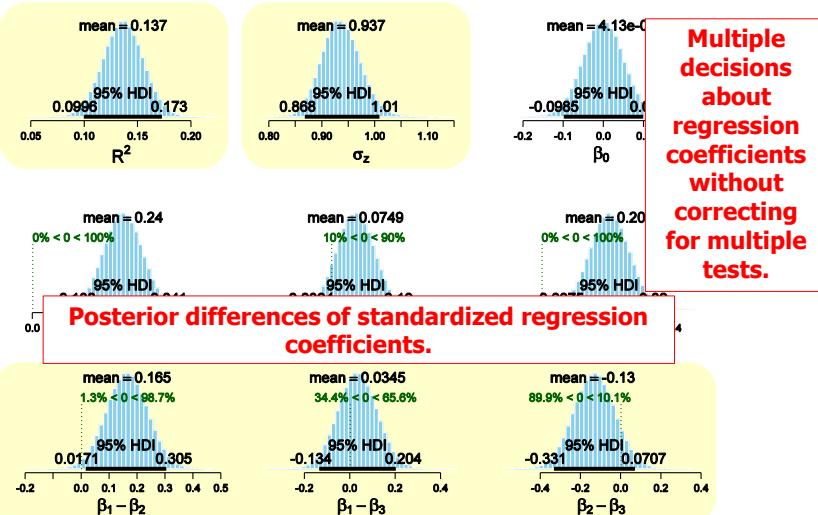
Bayesian Multiple Linear Regression



© John K. Kruschke, 2013

260

Bayesian Multiple Linear Regression



© John K. Kruschke, 2013

261

Agenda

- Bayes' rule, grid approximation, and R.
- Markov Chain Monte Carlo and JAGS.
- Robust Bayesian estimation for two groups.
 - Power
 - Sequential testing
- Bayesian hierarchical ANOVA.
 - Multiple comparisons
 - Heterogeneous variances
 - Other within-subject (repeated measures) designs
 - Split-plot design
- Bayesian regression.
 - Linear
 - Non-linear: Exponential, Sinusoidal, Auto-regressive AR(1)
 - Hierarchical regression for individuals and groups
- How to report a Bayesian analysis.

How to report a Bayesian analysis (book Ch. 23)

Essential points:

- **Motivate the Bayesian analysis** (versus NHST). In general, Bayesian analysis yields richer information and does not use ill-defined p values.
- **Describe the model and its parameters.** The posterior refers to the parameters, so they must be explained!
- **Describe and justify the prior.** Typically easy because the prior is only mildly informed by scale of data. See also optional robustness check.
- **Mention the MCMC details**, such as burn in, autocorrelation, and length of chain.
- **Interpret the posterior distribution.** HDI's, ROPE's, contrasts, shrinkage, etc.

Optional points:

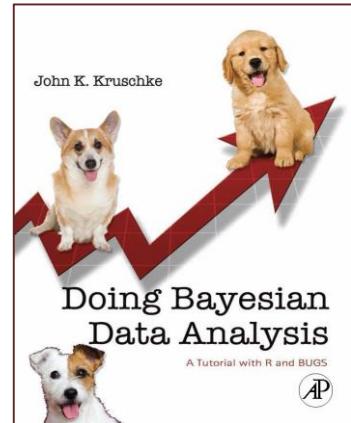
- **Robustness of posterior when prior is changed.** Conclusions from posterior do not change when prior is reasonably changed.
- **Posterior predictive check.** Do simulated data "look like" actual data?
- **Power analysis.** Can be very useful as additional measure of robustness of result.

Helpful points:

- **Post the raw data.** Other researchers can conduct other analyses. Enhances longevity of the results.
- **Post the MCMC chain.** Other researchers can examine other comparisons, or do power analyses and posterior predictive checks. Enhances longevity of the results.

Many other topics are in the book (and articles), e.g.

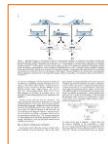
- ❖ **Power:** Probability of achieving the goals of research.
- ❖ **Model comparison and Bayes factors** and why it's a dangerous approach to null hypothesis testing.
- ❖ The **generalized linear model**.
- ❖ Many types of **regression**, including logistic regression and ordinal regression.
- ❖ Bayesian hierarchical **two way ANOVA with interaction contrasts**.
- ❖ **Log-linear models** vs chi-square test.
- ❖ All preceded by **extensive introductory chapters** covering notions of probability, Bayes' rule, MCMC, model comparison, etc.



© John K. Kruschke, 2013

264

The blog: <http://doingbayesiandataanalysis.blogspot.com/>



Kruschke, J. K. (2012). **Bayesian estimation supersedes the *t* test.**
Journal of Experimental Psychology: General.



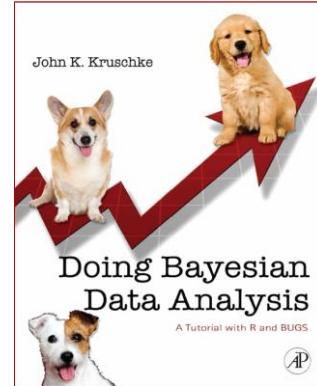
Kruschke, J. K. (2011). **Bayesian assessment of null values via parameter estimation and model comparison.**
Perspectives on Psychological Science, 6(3), 299-312.



Kruschke, J. K. (2010). **What to believe: Bayesian methods for data analysis.**
Trends in Cognitive Sciences, 14(7), 293-300.



Kruschke, J. K. (2010). **Bayesian data analysis.**
Wiley Interdisciplinary Reviews: Cognitive Science, 1(5), 658-676.



Kruschke, J. K. (2011).
Doing Bayesian Data Analysis: A Tutorial with R and BUGS.
Academic Press / Elsevier.

© John K. Kruschke, 2013

265