Introduction to Bayesian Modeling and Inference

Peter Lenk
University of Michigan
plenk@umich.edu

SCECR 2007

Outline

- Motivation
- Bayesian decision theory and inference
- Pooling information and shrinkage
- Markov Chain Monte Carlo
- Hierarchical Bayes (HB) models
- Metropolis Algorhitm
- WinBugs
- Extensive notes and GAUSS code on http://webuser.bus.umich.edu/plenk/downloads.htm

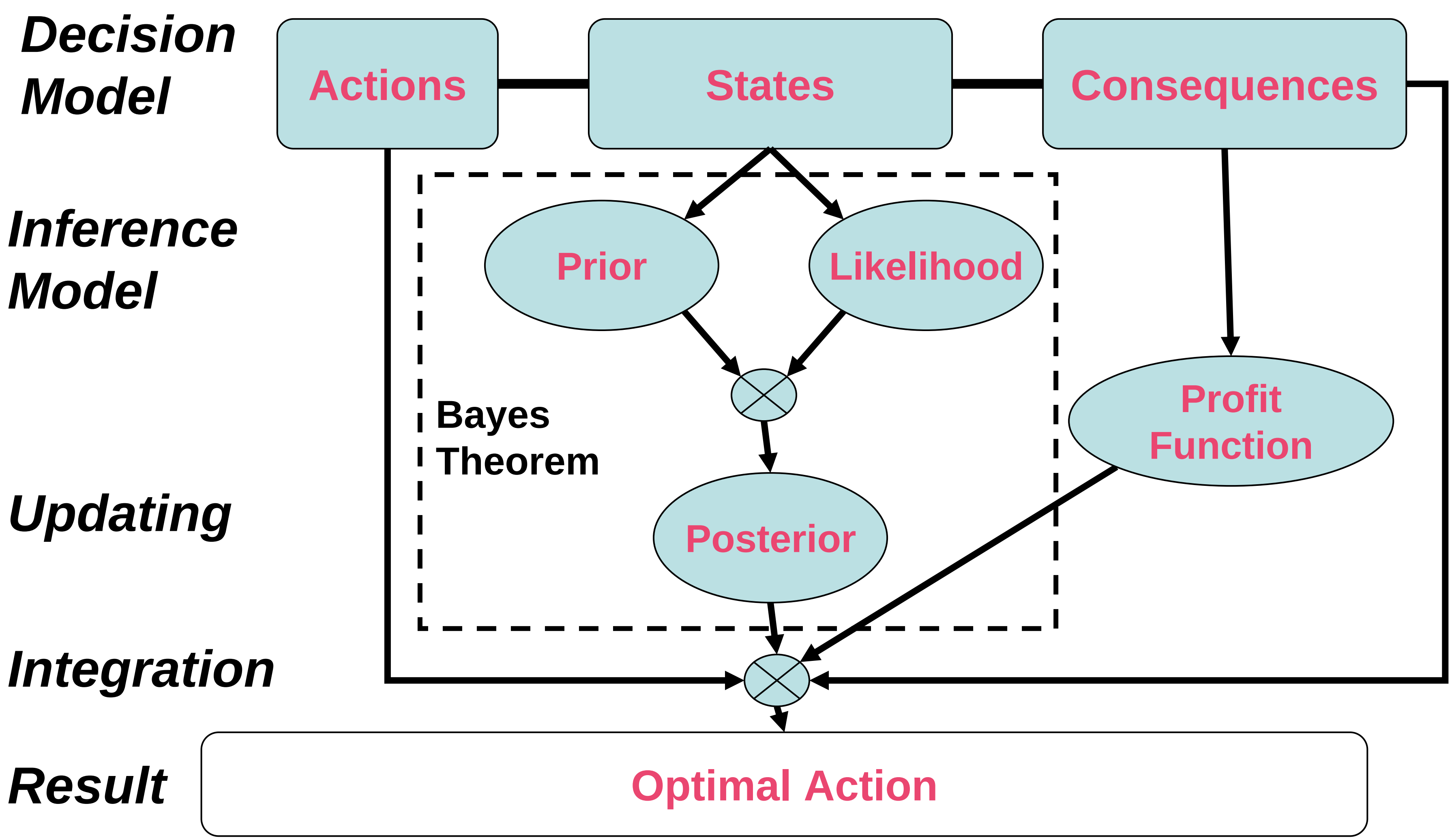
Motivation

- Decision making under uncertainty
  - How to make good decisions with limited information and high uncertainty
- Disaggregate decisions
  - Need to go beyond simplistic aggregates
- Distribution of heterogeneity
  - Recognize individual differences among sampling units - users, web pages, customers, etc
  - Limited information for each sampling unit

Situations where Hierarchical Bayes Rules

		Breath **Number of Units**	
		Narrow **Few**	***Broad*** **Many**
Depth **Observations per Units**	***Shallow*** **Few**	Only Bayes	**HB is Best!**
	Deep **Many**	Methods converge	Massive database

Bayesian Decision Theory



Models, Data, and Parameters

- Joint distribution of data given parameters

$$f[y_1, ..., y_n|\theta]$$

- Conditional independence is useful

$$f[y_1, ..., y_n|\theta] = f[y_1|\theta]\ f[y_2|\theta]...\ f[y_n|\theta]$$

- The likelihood function is the information in the data about θ

$$l(\theta) = f[y_1, ..., y_n|\theta]$$

Priors and Posteriors

- Beliefs about θ before observing the data are encoded in the prior distribution $p[\theta]$
- Beliefs are updated after observing data through Bayes theorem to obtain the posterior distribution
  - $p[\theta|\text{Data}] = l(\theta)p[\theta]/f[y_1 \ldots y_n]$
  - $f[y_1 \ldots y_n]$ is the marginal distribution of the data

Inference: Bayes Rules

- Bayesian inference centers on the posterior distribution
- Given a loss function L(θ,w), the Bayes rule minimizes the expected posterior loss

$$\omega = \arg\max_{w} \int L(\theta, w)\, p[\theta \mid \mathrm{Data}]\, d\theta$$

- Squared error loss: ω = posterior mean
- Absolute error loss: ω = posterior median
- 0/1 loss: ω = posterior mode

Measures of Estimation Uncertainty

- Posterior Risk

$$\rho(\omega) = \int L(\theta, \omega)\, p[\theta \mid \text{Data}]\, d\theta$$

  - Squared error loss: risk = posterior variance

Predictive Distribution

- Distribution for next observation

$$f\left[y_{n+1} \mid y_1 \cdots y_n\right] = \frac{f\left[y_1 \cdots y_{n+1}\right]}{f\left[y_1 \cdots y_n\right]}$$

- Under conditional independence

$$f\left[y_{n+1} \mid y_1 \cdots y_n\right] = \int f\left[y_{n+1} \mid \theta\right] p\left[\theta \mid y_1 \cdots y_n\right] d\theta$$

“Classical” versus Bayes

Classical

- Parameters are fixed
- Data are random
- Centers on sampling distributions
  - Likely values of statistic
- Integrates over sample space
- Protects against sampling errors

Bayes

- Parameters are random
- Data are fixed
- Centers on posterior distributions
  - Likely values of parameters
- Integrates over parameter space
- Learning mechanism

Easy Example
Estimate the Mean

- Model
  - $Y_i \sim N(\mu, \sigma^2)$ for i = 1, .., n
  - Assume that σ is known to keep it simple

- Prior distribution for μ is $N(m_0, v_0^2)$
  - $m_\mathbf{0}$ is your best guess at μ
  - $v_\mathbf{0}^2$ is your uncertainty about your guess

Prior, Likelihood, & Posterior

$$p[\mu] \propto \exp\left[-\frac{1}{2v_0^2}(\mu - m_0)^2\right]$$

$$l[\mu] \propto \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right]$$

$$p[\mu \mid \text{Data}] \propto l[\mu]p[\mu]$$

$$\mu \mid \text{Data} \sim \mathrm{N}\left(\mathrm{m_n}, v_n^2\right)$$

Posterior Distribution

- Posterior distribution is $N(m_n, v_n^2)$
- What happens as
  - n becomes large?
  - v_0 becomes large?
  - σ becomes small?

$$m_n = w\bar{y} + (1-w)m_0$$

$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{v_0^2}} \text{ and } 0 < w < 1$$

$$v_n^2 = \frac{1}{\frac{n}{\sigma^2} + \frac{1}{v_0^2}}$$

Updating Conjugate Models

Posterior distribution is in same family as prior

- Prior Parameters
- Posterior Parameters

$m_0 \longrightarrow m_n = w\bar{y} + (1 - w)m_0$

$v_0^2 \longrightarrow v_n^2 = \dfrac{1}{\dfrac{n}{\sigma^2} + \dfrac{1}{v_0^2}}$

Predictive Distribution

- Predictive distribution for Y_{n+1} is normal with mean m_n and variance $\sigma^2+v_n^2$

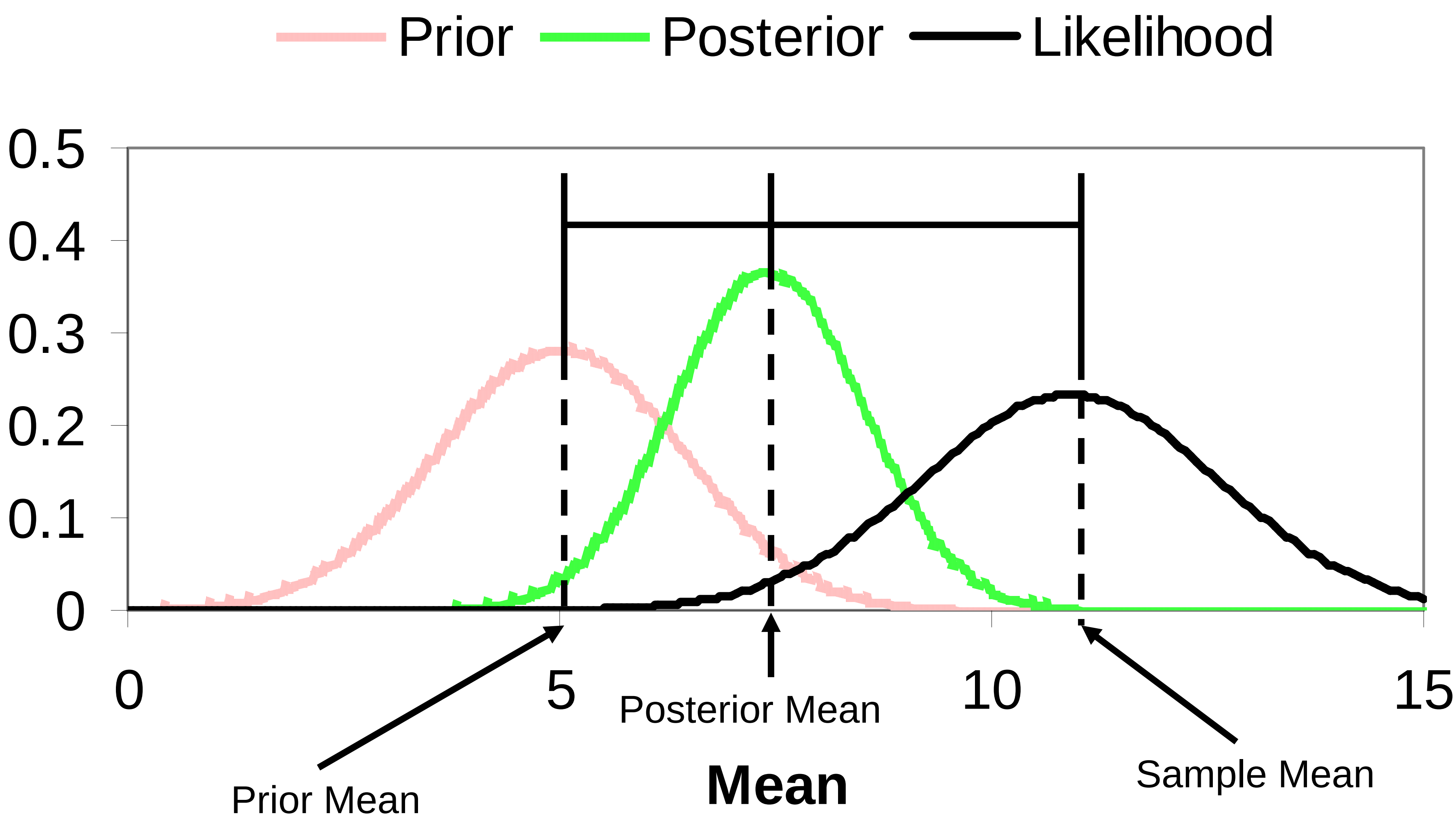
Shrinkage Estimators

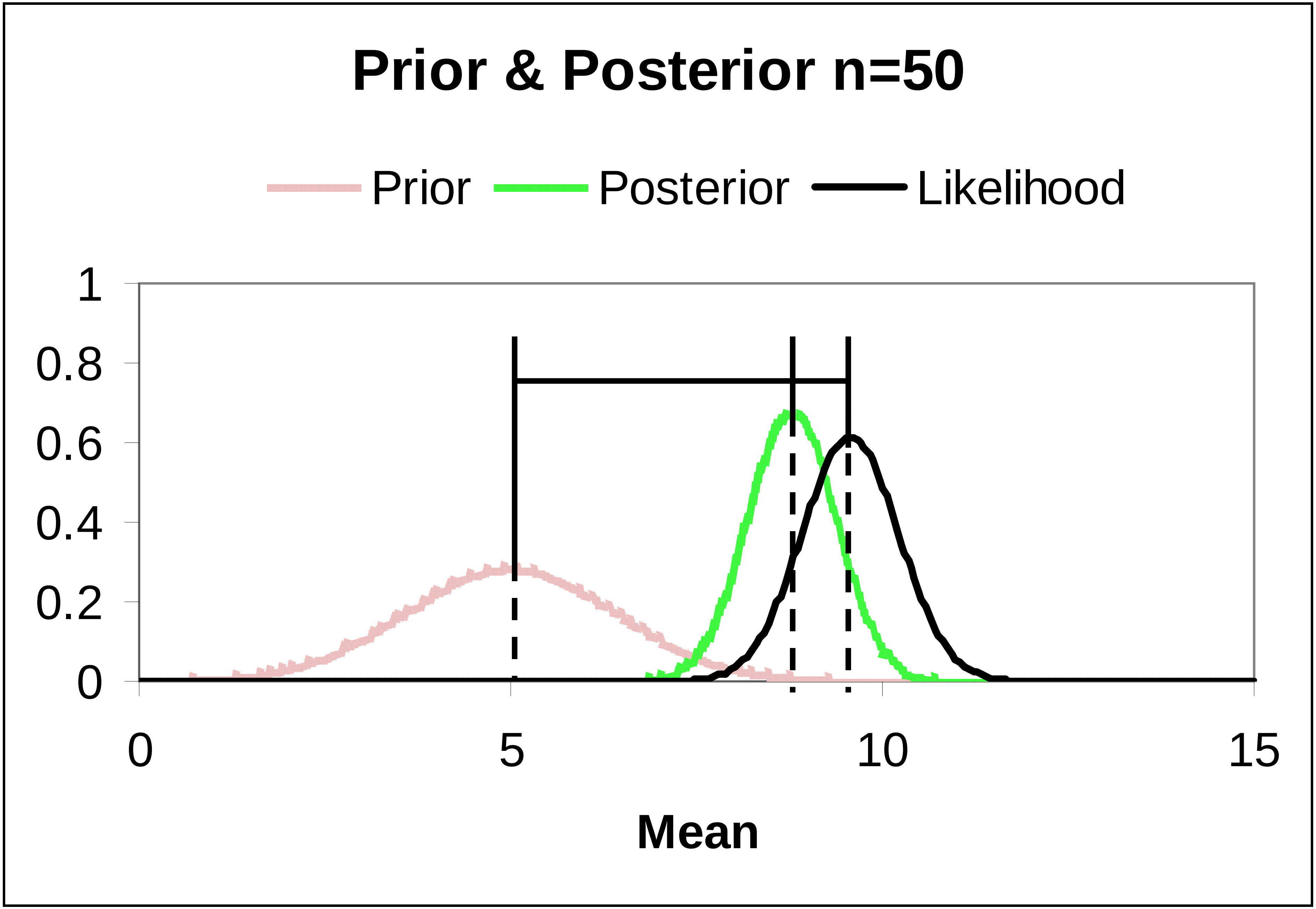
- In the previous example, the Bayes estimator combines the prior mean with sample mean
- The Bayes estimate “shrinks” the sample mean towards your prior guess
- The amount of shrinkage depends on the relative amount of sample information and prior information

Do it with Data

- The truth is that $Y \sim N(\mu=10, \sigma^2=16)$
- Prior for $\mu \sim N(m_0=5,\ v_0^2=2)$
  - Prior is informative and way off
- Data
  - $n = 5$, Average = 10.9, Variance = 14.7
- Posterior for $\mu \sim N(m_0= 7.4,\ v_0^2= 1.2)$

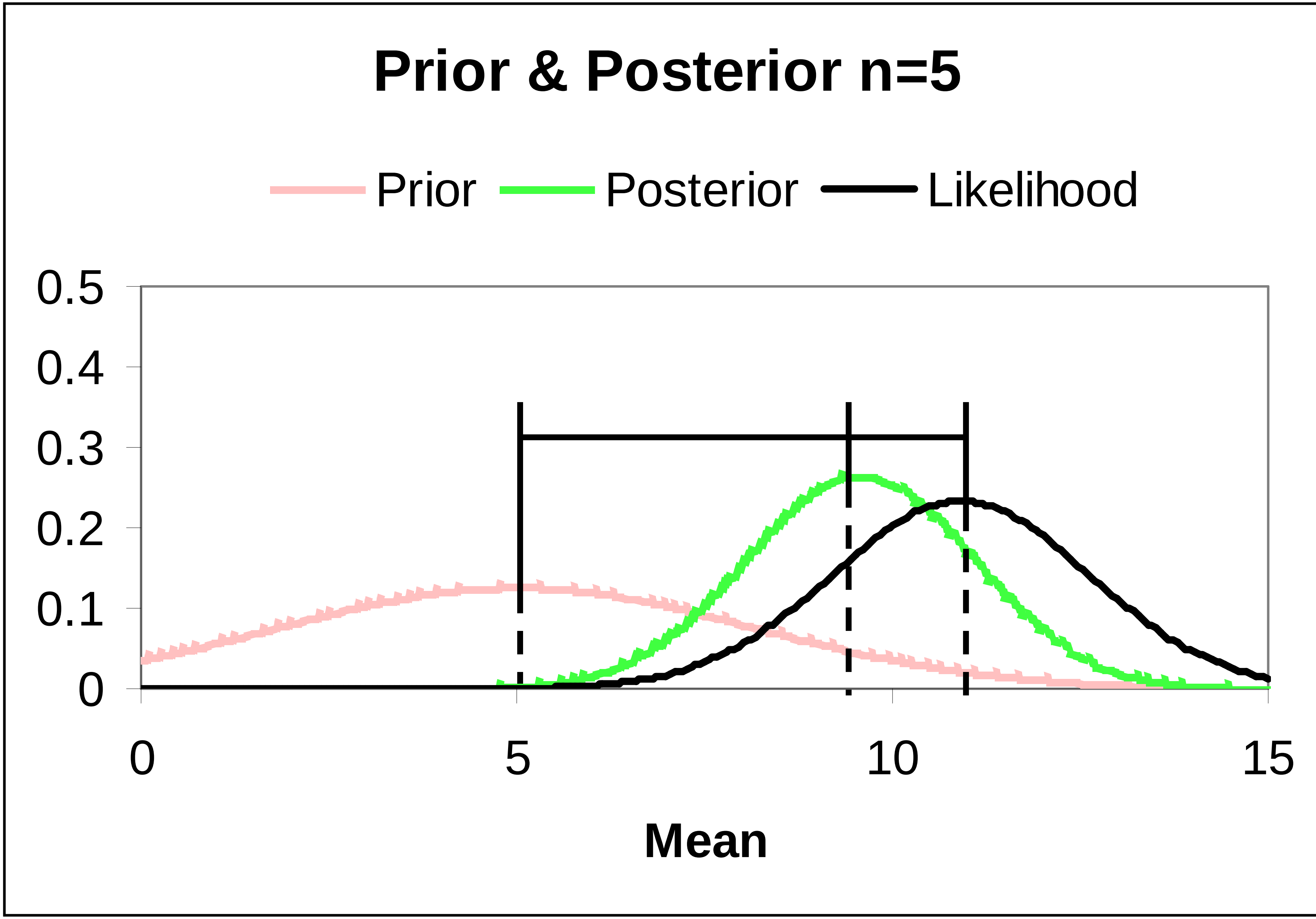
Prior & Posterior n=5

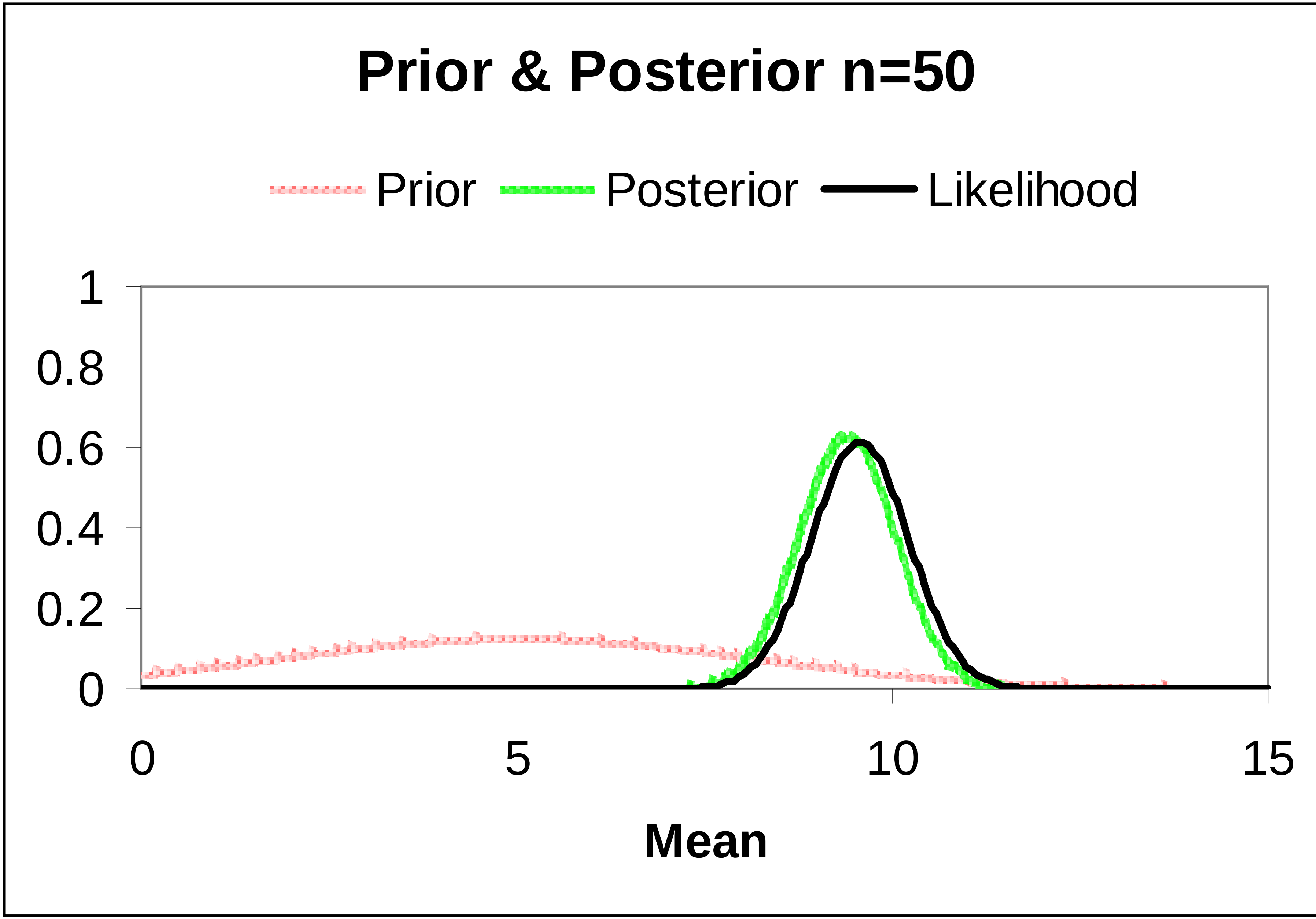



Prior & Posterior n=50
Prior
Posterior
Likelihood
1
0.8
0.6
0.4
0.2
0
0
5
10
15
Mean

Less Informative Prior

- The truth is that $Y \sim N(\mu=10,\sigma^2=16)$
- Prior for $\mu \sim N(m_0=5, v_0^2=10)$
  - Prior variance is 10 instead of 2
- Data
  - n = 5, Average = 10.9, Variance = 14.7
- Posterior for $\mu \sim N(m_0= 7.4, v_0^2= 1.2)$


Prior & Posterior n=5
Prior
Posterior
Likelihood
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
Mean


Prior & Posterior n=50
Prior
Posterior
Likelihood
1
0.8
0.6
0.4
0.2
0
0
5
10
15
Mean

Summary

- Prior has less effect as sample size increases
- Very informative priors give good results with smaller samples if prior information is correct
- If you really don’t know, then use “flatter” or less informative priors

Good & Bad News

- Only simple models result in closed-form equations
- Most models require numerical methods to compute posterior mean, posterior standard deviations, predictions and so on

Monte Carlo

- Compute posterior mean of function $T(\theta)$.

$$E\left[T(\theta) \mid y\right] = \int T(\theta) p(\theta \mid y) d\theta$$

- Generate random draws $\theta_1, \theta_2, \ldots, \theta_m$ from posterior distribution using a random number generator.

$$E\left[T(\theta) \mid y\right] \approx \frac{1}{m} \sum_{j=1}^{m} T\left(\theta_j\right)$$

Good & Bad News

- If your computer has a random number generator for the posterior distribution, Monte Carlo is a snap to do
- Your computer almost never has the correct random number generator

Importance Sampling Approximation: Generate from g instead of f

$$\int T(\theta) f(\theta) d\theta = \int T(\varphi) \frac{f(\varphi)}{g(\varphi)} g(\varphi) d\varphi$$

$$\approx \frac{\sum_{i=1}^{m} T(\varphi_i) r(\varphi_i)}{\sum_{i=1}^{m} r(\varphi_i)} = \sum_{i=1}^{m} T(\varphi_i) w(\varphi_i)$$

$$r(\varphi_i) \propto \frac{f(\varphi_i)}{g(\varphi_i)} \text{ and } w(\varphi_i) = \frac{r(\varphi_i)}{\sum_{j=1}^{m} r(\varphi_j)}$$

Don’t need constants for f and g to compute weights w!

Markov Chain Monte Carlo

- Extension of Monte Carlo
- Random draws are not independent
- Joint distribution $f(\beta,\sigma|Y)$ does not have a convenient random number generator
- Useful if “full” conditional distributions $g(\sigma|\beta,Y)$ and $h(\beta|\sigma,Y)$ have known random number generators

Iterative Generation from Full Conditionals

- Start at σ_0
- Generate $\beta_1 \sim h(\beta|\sigma_0,Y)$
- Generate $\sigma_1 \sim g(\sigma|\beta_1,Y)$

…

- Generate $\beta_{m+1} \sim h(\beta|\sigma_m,Y)$
- Generate $\sigma_{m+1} \sim g(\sigma|\beta_{m+1},Y)$

...

Markov Chain Theory

- $\{(\beta_m,\sigma_m)\}$ forms a Markov chain with stationary distribution $f(\beta,\sigma|Y)$
  - Eventually (β_m,σ_m) are draws from the stationary distribution
  - Delete the first k draws because chain has not converted to the stationary distribution
  - How to pick k?

Regression Model

- Model:
  - $Y_i = x_i'\beta + \varepsilon_i$ and $\varepsilon_i \sim N(0,\sigma^2)$
- Priors
  - $\beta \sim N_p(b_0,V_0)$
  - $\sigma^2 \sim IG(r_0/2,s_0/2)$ the Inverted Gamma Dist

$$f\left(\sigma^2 \mid \frac{r}{2},\frac{s}{2}\right) = \frac{\left(\frac{s}{2}\right)^{\frac{r}{2}}}{\Gamma\left(\frac{r}{2}\right)}\left(\sigma^2\right)^{-\left(\frac{r}{2}+1\right)} \exp\left(-\frac{s}{2\sigma^2}\right) \text{ for } \sigma^2 > 0$$

MCMC Steps 1 & 2

- Step 1: joint distribution
  $[y_1|\beta,\sigma^2] \ldots [y_n|\beta,\sigma^2]\ [\beta][\sigma^2]$
- Step 2: full conditional for β
  $[\beta|Y,\, \sigma^2] \propto [y_1|\beta,\sigma^2] \ldots [y_n|\beta,\sigma^2]\ [\beta][\cancel{\sigma^2}]$
  - $\beta|Y,\, \sigma^2 \sim N(b_n,\, V_n)$
  - $V_n^{-1} = X'X/\,\sigma^2 + V_0^{-1}$
  - $b_n = V_n(X'Y/\,\sigma^2 + V_0^{-1}b_0)$

MCMC Step 3

- Step 3: full conditional for σ^2

$$[\sigma^2 \mid Y, \beta] \propto [y_1|\beta,\sigma^2] \ldots [y_n|\beta,\sigma^2]\, [\cancel{\beta}][\sigma^2]$$

- ➢ $\sigma^2 \mid Y, \beta \sim IG(r_n/2, s_n/2)$
- ➢ $r_n = r_0 + n$
- ➢ $s_n = s_0 + SSE$
- ➢ $SSE = (Y-X\beta)' (Y-X\beta)$

MCMC Steps 4 to 6

- Step 4:Initialize β_0 and σ_0. To what?
- Step 5: Recursively generate

$$\beta_{m+1}|Y, \sigma_m^2 \sim N(b_n, V_n)$$

$$\sigma_{m+1}^2 \,|Y, \beta_{m+1} \sim IG(r_n/2, s_n/2)$$

- Step 6: Stop when __________

MCMC Finished

- Drop first k iterations for burn-in of Markov chain to stationary distribution
- Use remainder of draws to compute
  - ➢ Statistics for the posterior distributions
    - Means, variances, quartile, etc
  - ➢ Historgrams or Box-Plots
  - ➢ Predictive distributions
  - ➢ Other decision variables
    - Elasticities
    - Market shares
    - Live-time value
    - Network connectivity

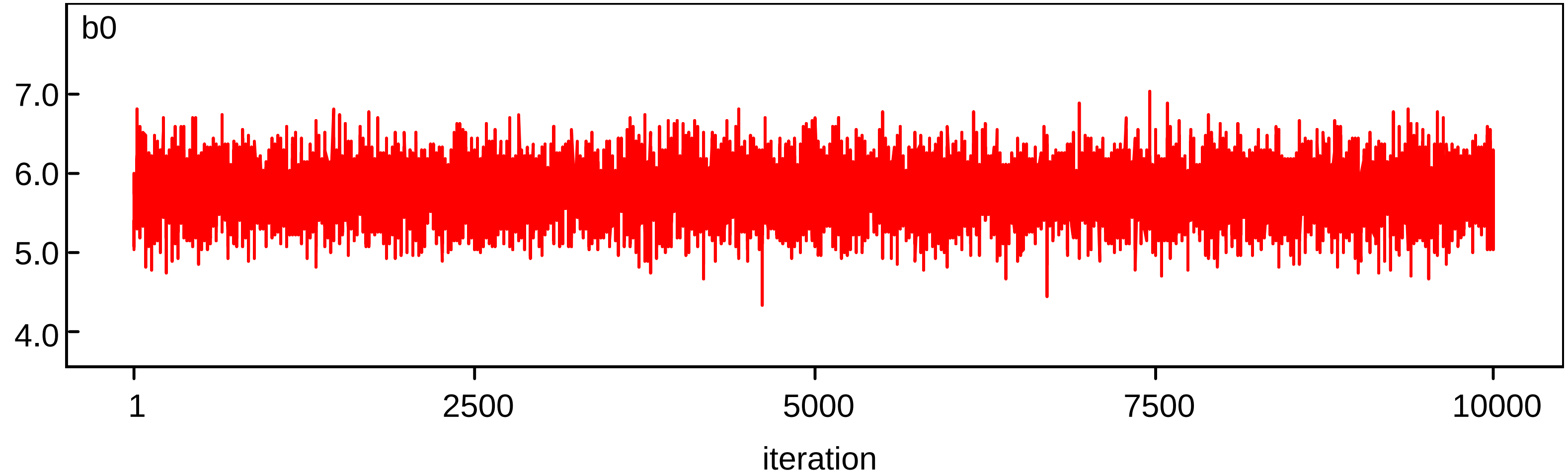
Regression Example

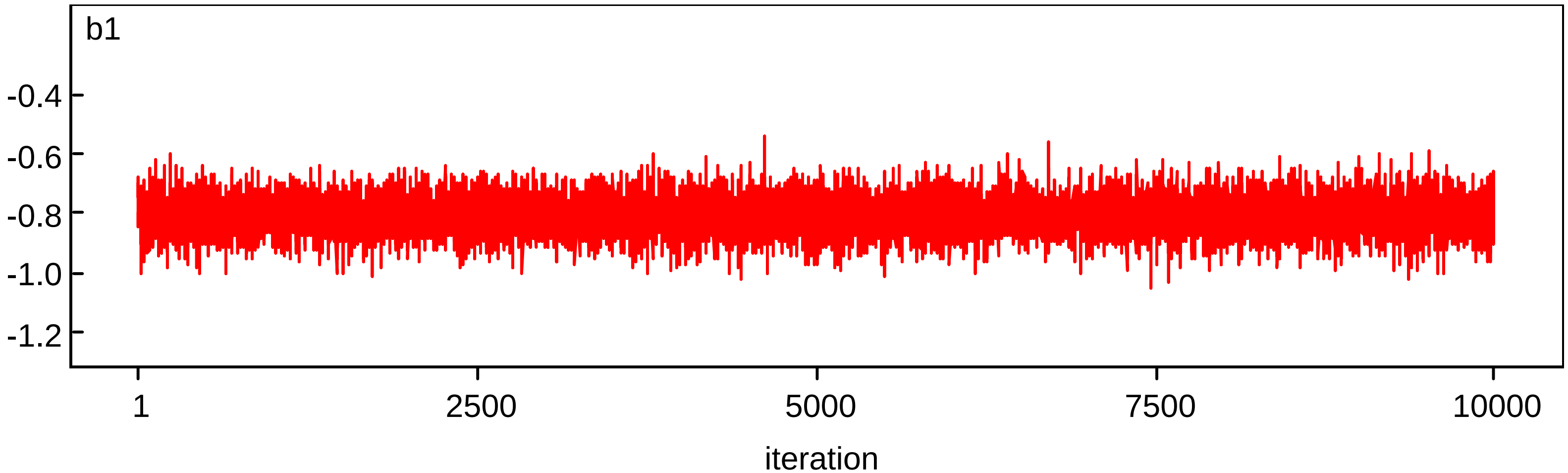
- Movie box office revenue for 2004
  - http://www.boxofficeprophets.com
  - N = 349 releases
  - Y = ln(Total Revenue $m)
  - X1 = ln(Number Opening Screens)
  - X2 = ln(Opening Weekend Revenue $m)
- Log-log model
  - $Y = b0 + b1*X1 + b2*X2 + \varepsilon$

Prior Distributions

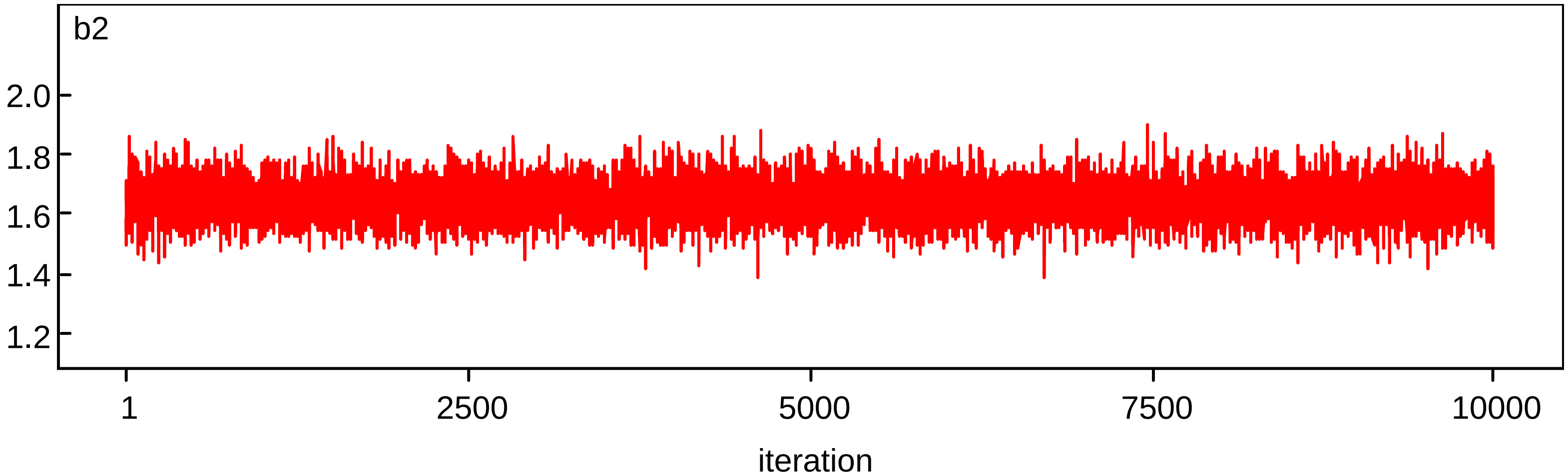
- Regression coefficients are normal with mean 0 and standard deviation 100
- 1/error variance is gamma with parameters 0.1 and 0.1
- Used WinBugs

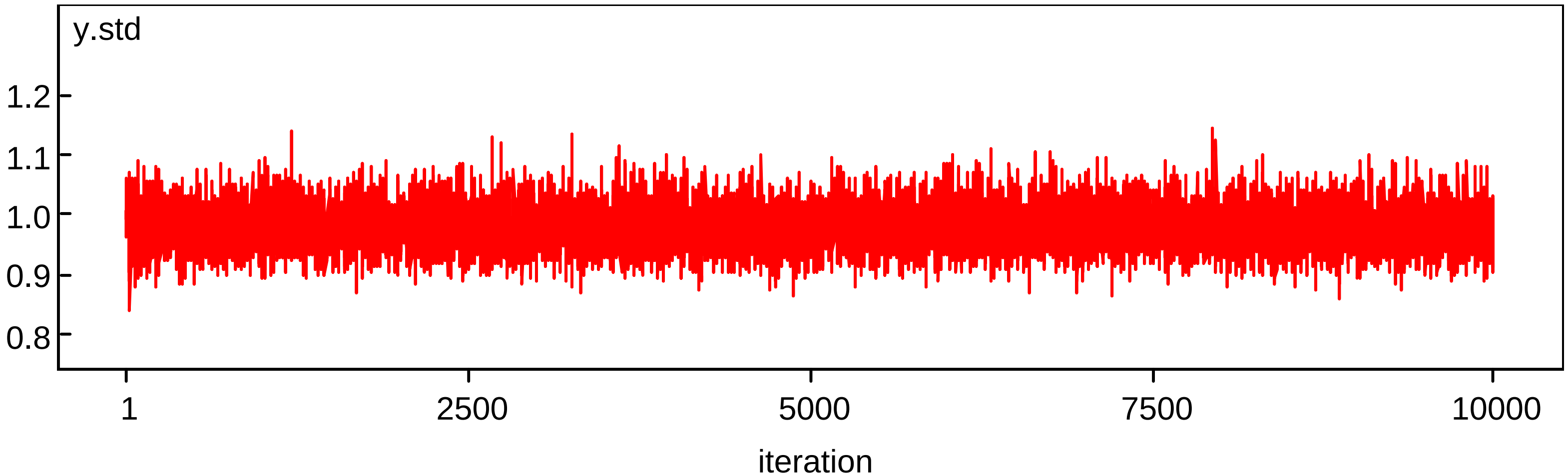
MCMC Iterations for b0 and b1





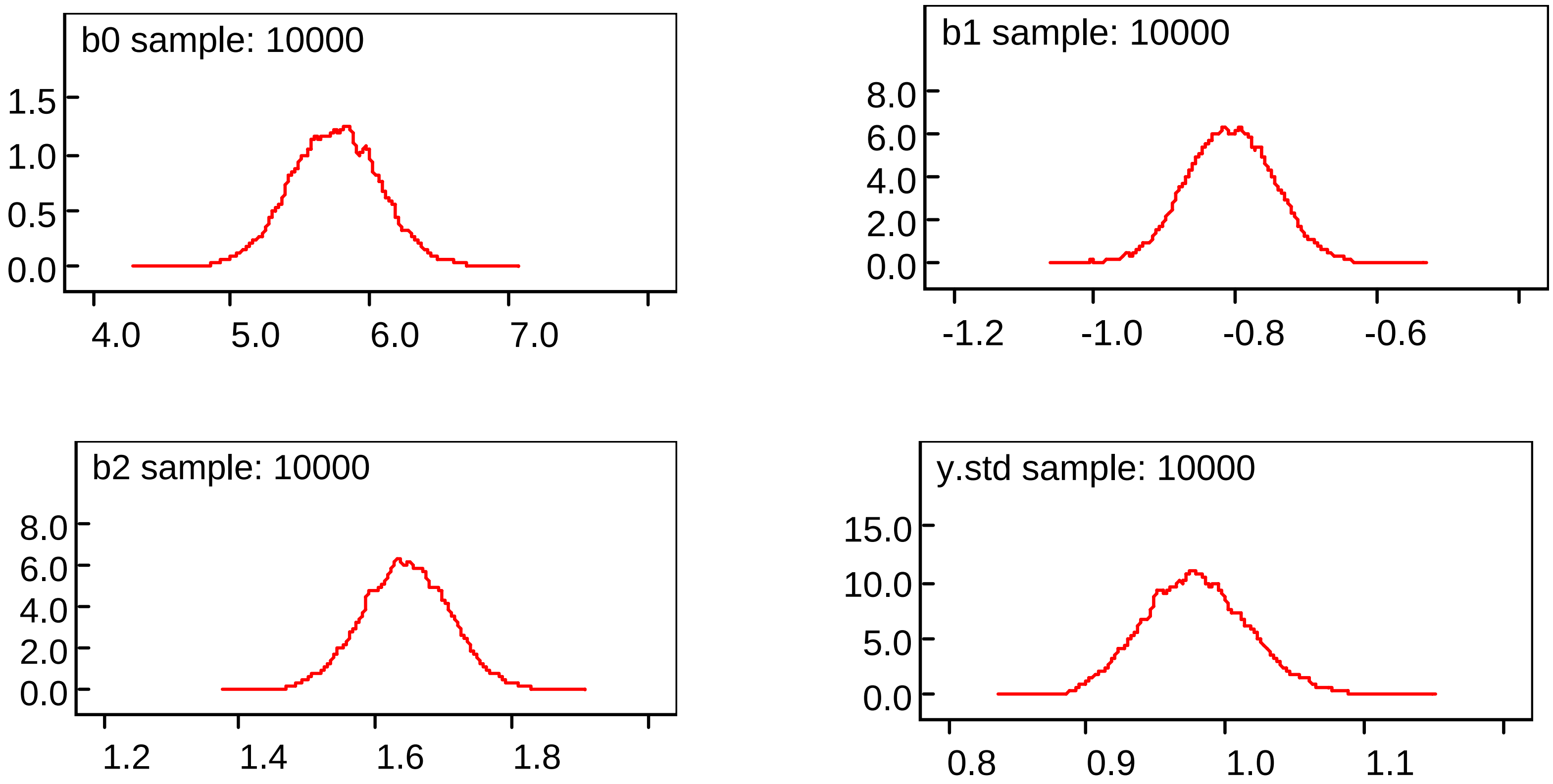
MCMC Iterations for b2 and σ





Posterior Distributions

Histograms from MCMC Iterations



Estimates

Computed from MCMC Iterations

Posterior mean and standard deviation

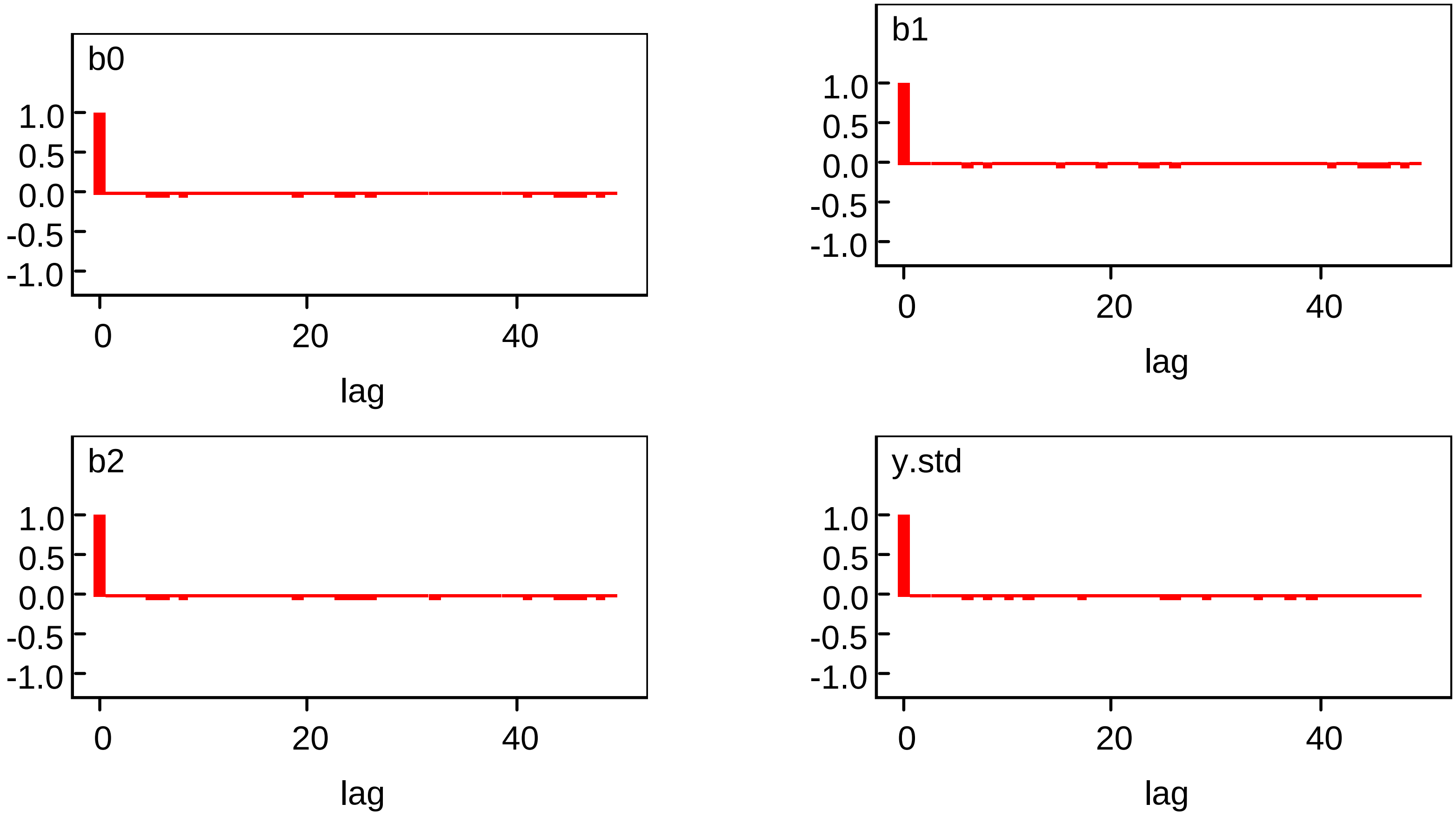
Numerical accuracy of MCMC for mean

Percentiles of the posterior distribution

Node	mean	sd	MC error	2.5%	median	97.5%
b0	5.748	0.323	0.003045	5.11	5.75	6.38
b1	-0.81	0.062	5.69E-4	-0.93	-0.81	-0.68
b2	1.645	0.065	6.2E-4	1.52	1.65	1.77
y.std	0.979	0.038	3.856E-4	0.91	0.98	1.06

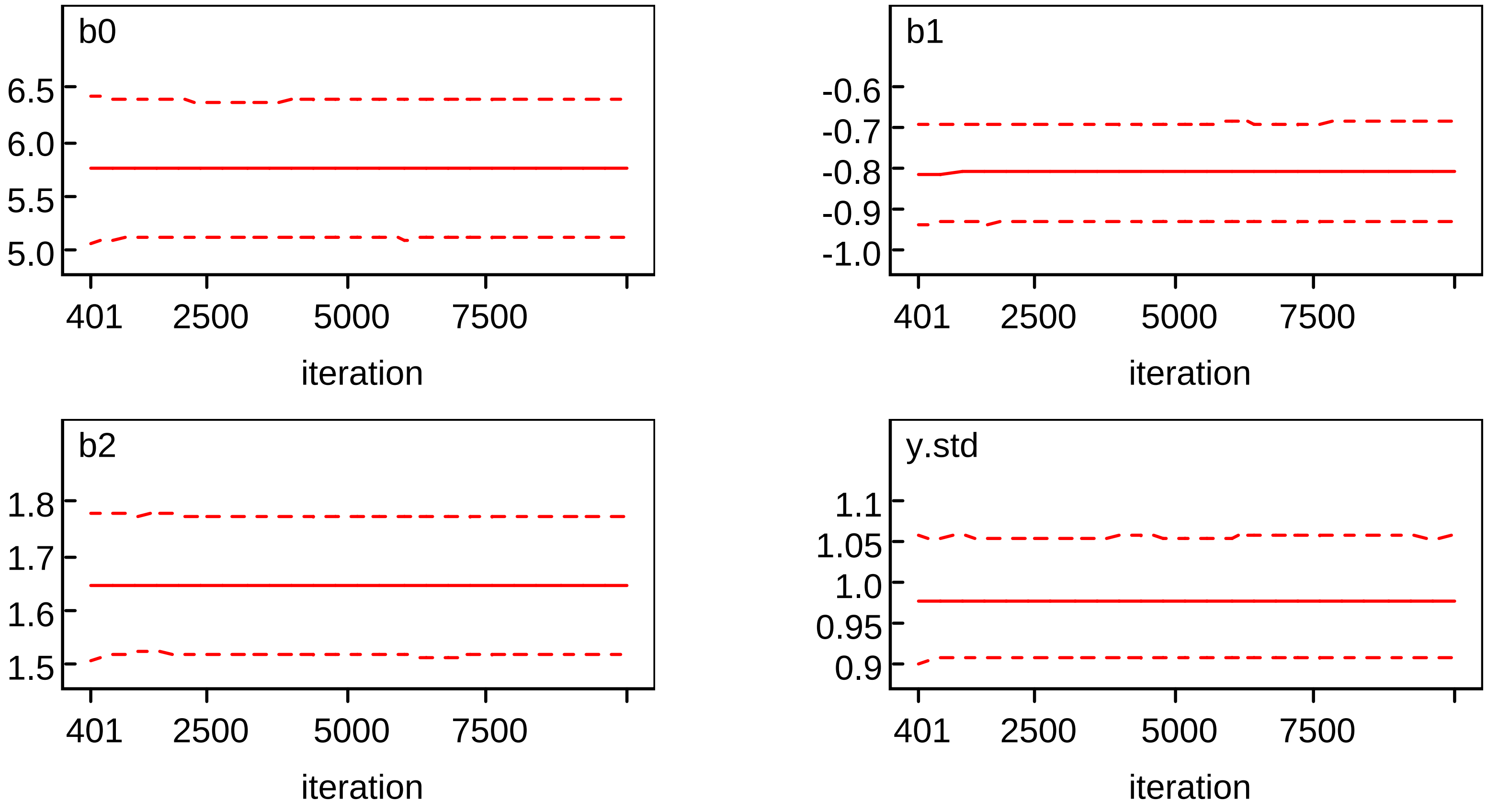
Note: Multicolinearity problem with X1 and X2 and possible endogeneity issues, but this is just an example.

ACF for Parameters



The greater the autocorrelation in the draws, the longer you need to run the chain to obtain a level of numerical accuracy.

Running Quartiles for MCMC



These graphs demonstrates that the estimates are stable over iterations.

FAQ about MCMC

Skip to HB

- How should I initialize the parameters?
- How long of a burn-in period?
- How many iterations should I use for estimation?
- What does “mixing” mean?

Initialize Parameters

- Need to be careful that you don't start chain in a bad area of the parameter space
- Try multiple runs with different starting values
- Use estimates from previous analysis (OLS or MLE), if you have them
- Initialize parameters to reasonable values

Burn-In Period

- Trace of MCMC draws versus iteration should stabilize (not to a constant)
- Gelman & Rubin convergence diagnostic is popular. It requires multiple chains
- Try simulated data where you know the answer
- Hard question in general

Number of Iterations for Estimation

- MC Error gives numerical accuracy of the MCMC estimator of the posterior mean
  - $\pm$ 2*(MC Error) is how far off the estimate should be if you reran the MCMC
- Depends on mixing
  - See next slide
- Try simulated data where you know the true value
- Hard question in general

Mixing

- How well the MCMC algorithm covers the parameter space
- ACF (autocorrelation function)
  - Correlation over time in MCMC draws
  - Large values of ACF (after lag 0) indicates poor mixing
- Need to use more iterations to compensate for large poor mixing

Hierarchical Bayes Model

- Two level model when there are repeated measurements on each unit (subject)
  - Subject level model describes variation of observations within subjects
  - Population level model describes variation of subject-level parameters across the population
- Population-level model
  - Allows sharing of information or pooling across subjects
  - Acts as a “prior” for estimating subject-level parameters

HB Model for Weekly Spending

- Household-level model: Household i and week j
  - $Y_{i,j} \sim N(\mu_i, \sigma_i^2)$ for $i = 1...N$ and $j = 1...n_i$
  - Household mean μ_i depends on household
- Heterogeneity in household means
  - $\mu_i \sim N(\theta, \tau^2)$
  - θ is population mean
- Priors
  - θ is $N(u_0, v_0^2)$
  - Variances σ_i^2 and τ^2 are known

Precisions = 1/Variance

$$\Pr(\theta) = \frac{1}{v_0^2} \text{ is prior precision}$$

$$\Pr(\mu_i \mid \theta) = \frac{1}{\tau^2}$$

$$\Pr(Y_{i,j} \mid \mu_i) = \frac{1}{\sigma_i^2} \text{ and } \Pr(Y_{i,j} \mid \theta) = \frac{1}{\tau^2 + \sigma_i^2}$$

$$\Pr(\overline{Y}_i \mid \mu_i) = \frac{n}{\sigma_i^2} \text{ and } \Pr(\overline{Y}_i \mid \theta) = \frac{1}{\tau^2 + \frac{\sigma_i^2}{n}}$$

Joint Distribution

$$P(Y, \mu, \theta) =$$

$$\underbrace{h\left(\theta \mid u_0, v_0^2\right)}_{\textbf{Prior}} \underbrace{\prod_{i=1}^{N} g\left(\mu_i \mid \theta, \tau^2\right)}_{\textbf{Between Subjects}} \underbrace{\prod_{j=1}^{n_i} f\left(y_{i,j} \mid \mu_i, \sigma_i^2\right)}_{\textbf{Within Subjects}}$$

Posterior Distribution

θ | Data ~ N(u_N,v_N^2)

$$v_N^{-2} = \frac{1}{v_0^2} + \sum_{i=1}^{N} \frac{1}{\tau^2 + \frac{\sigma_i^2}{n_i}}$$

$$u_N = w_0 u_0 + \sum_{i=1}^{N} w_i \bar{Y}_i$$

$$w_0 = \frac{\Pr(\theta)}{\Pr(\theta \mid Y)} \text{ and } w_i = \frac{\Pr(\bar{Y}_i \mid \theta)}{\Pr(\theta \mid Y)}$$

Shrinkage between weighted average of household means and prior mean. Weights depend on relative precisions.

Posterior Mean of μ_i

$$E[\mu_i \mid Y] = \alpha_i \bar{Y}_i + (1 - \alpha_i) u_N$$

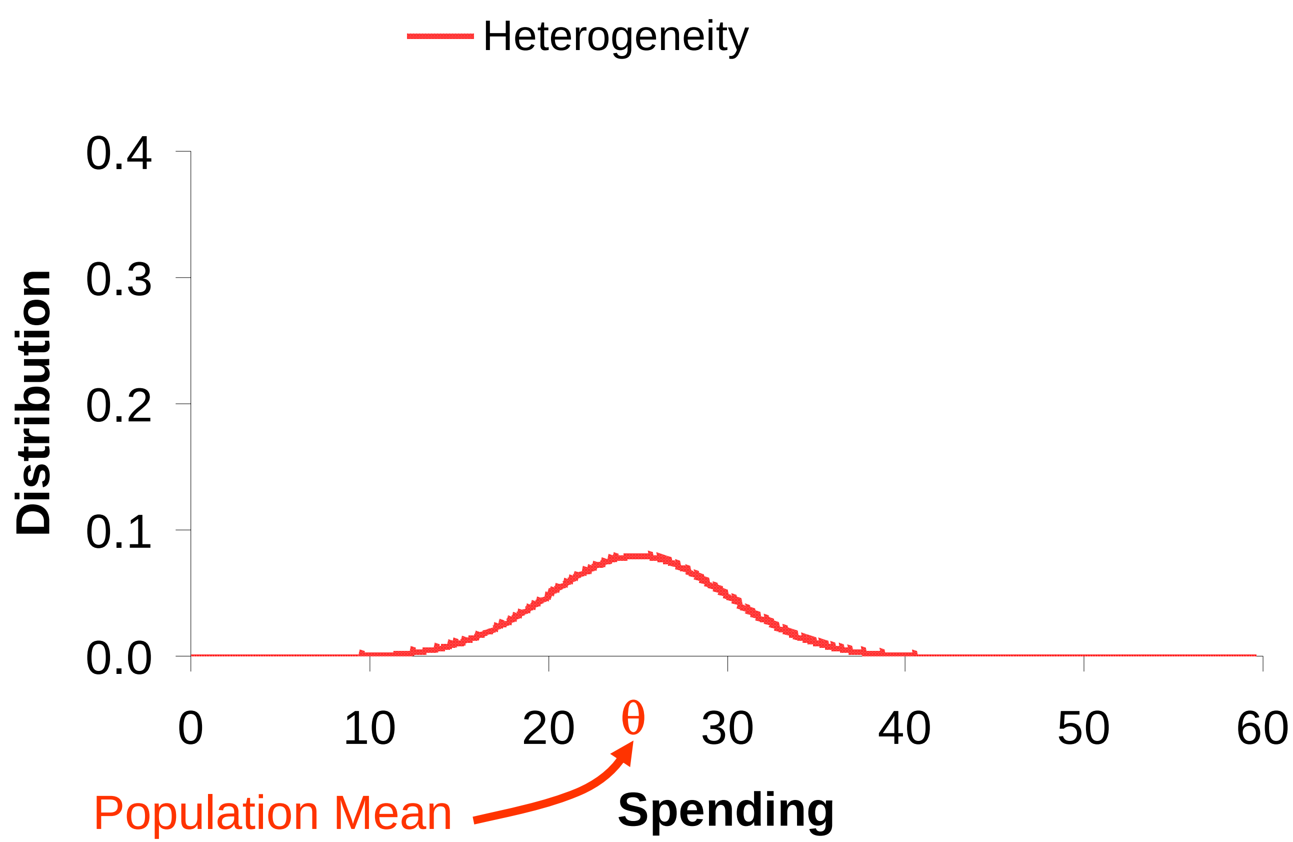
$$\alpha_i = \frac{\Pr(\bar{Y}_i \mid \mu_i)}{\Pr(\mu_i \mid \theta) + \Pr(\bar{Y}_i \mid \mu_i)} = \frac{\frac{n_i}{\sigma_i^2}}{\frac{1}{\tau^2} + \frac{n_i}{\sigma_i^2}}$$

Fantastic!

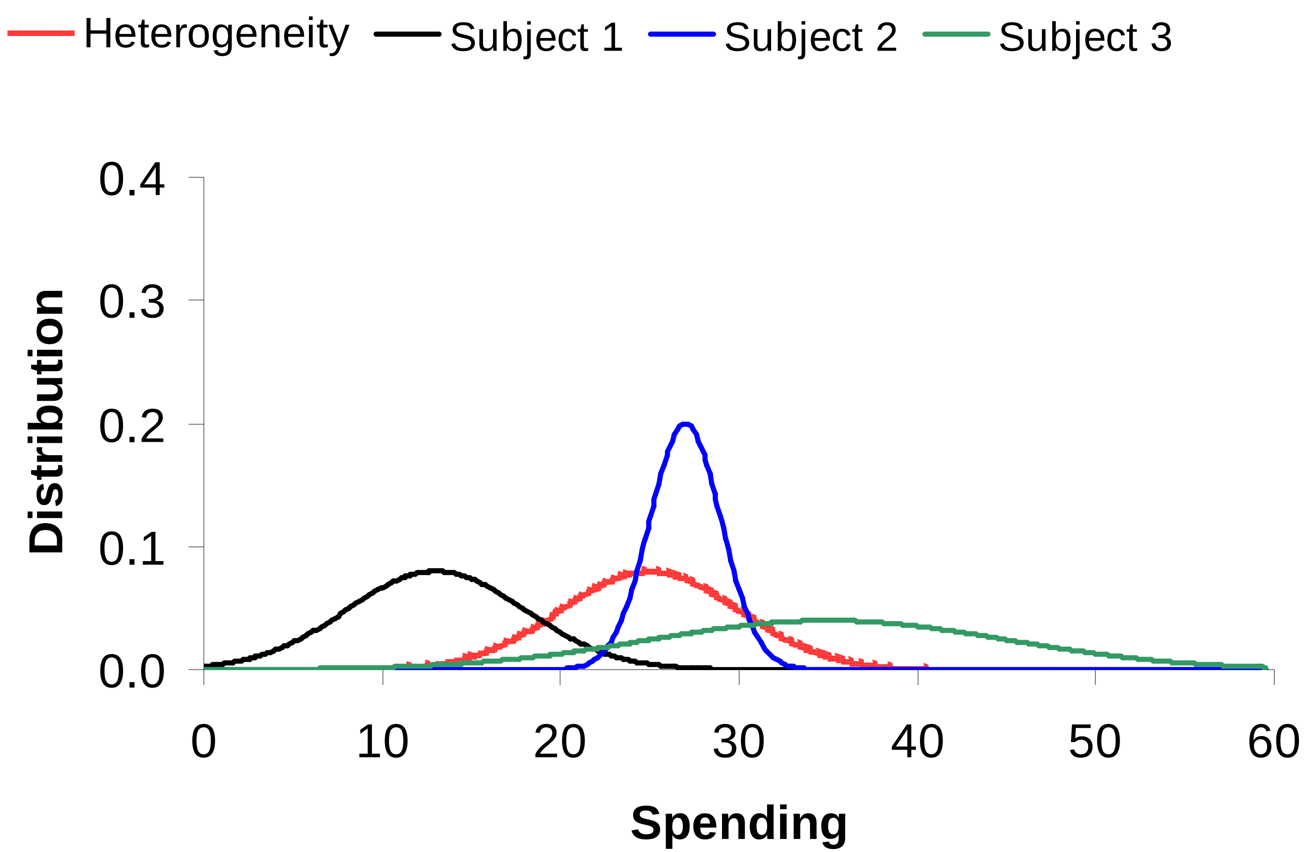
Shrinkage between individual household average and *population estimate*.

Weights depend on relative precisions.

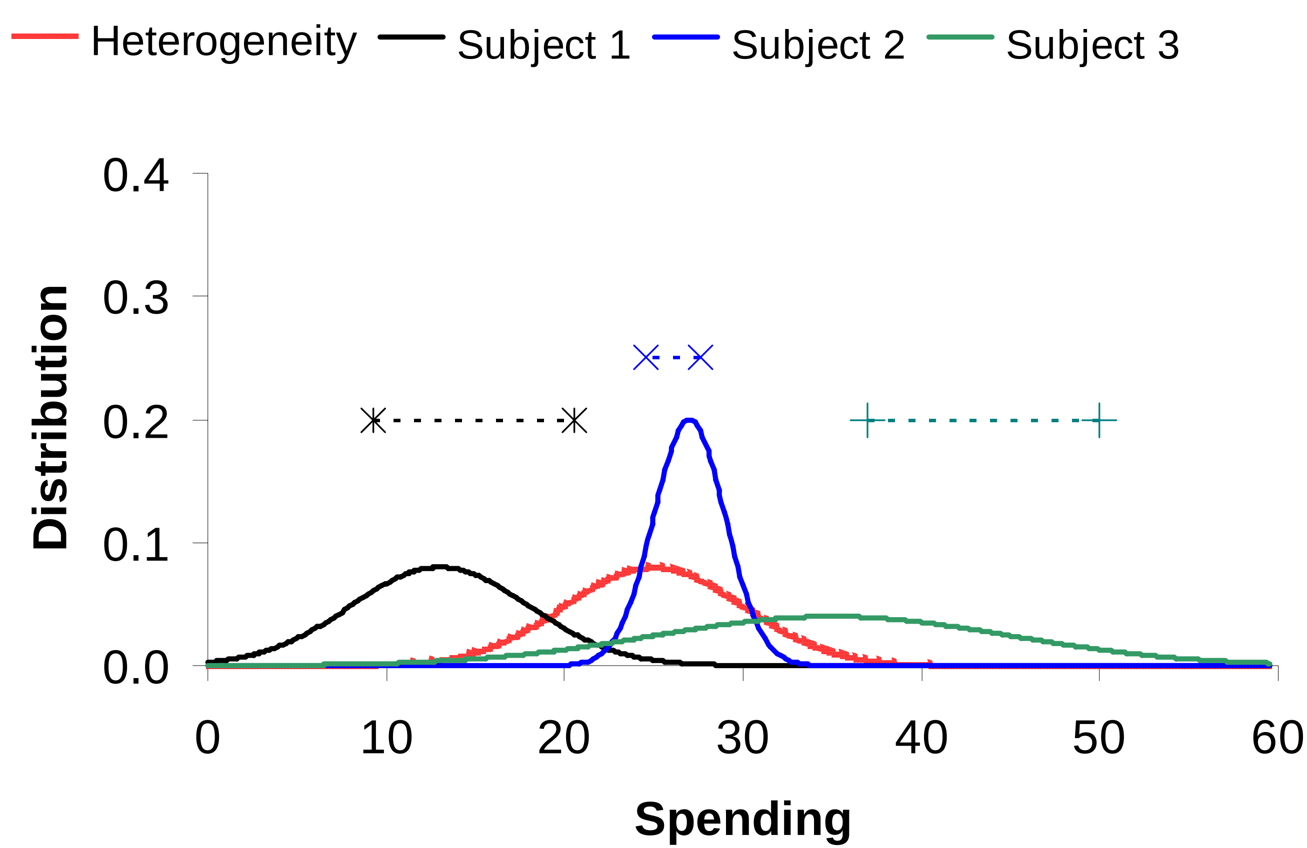
Between-Subject Heterogeneity in Household Mean Spending



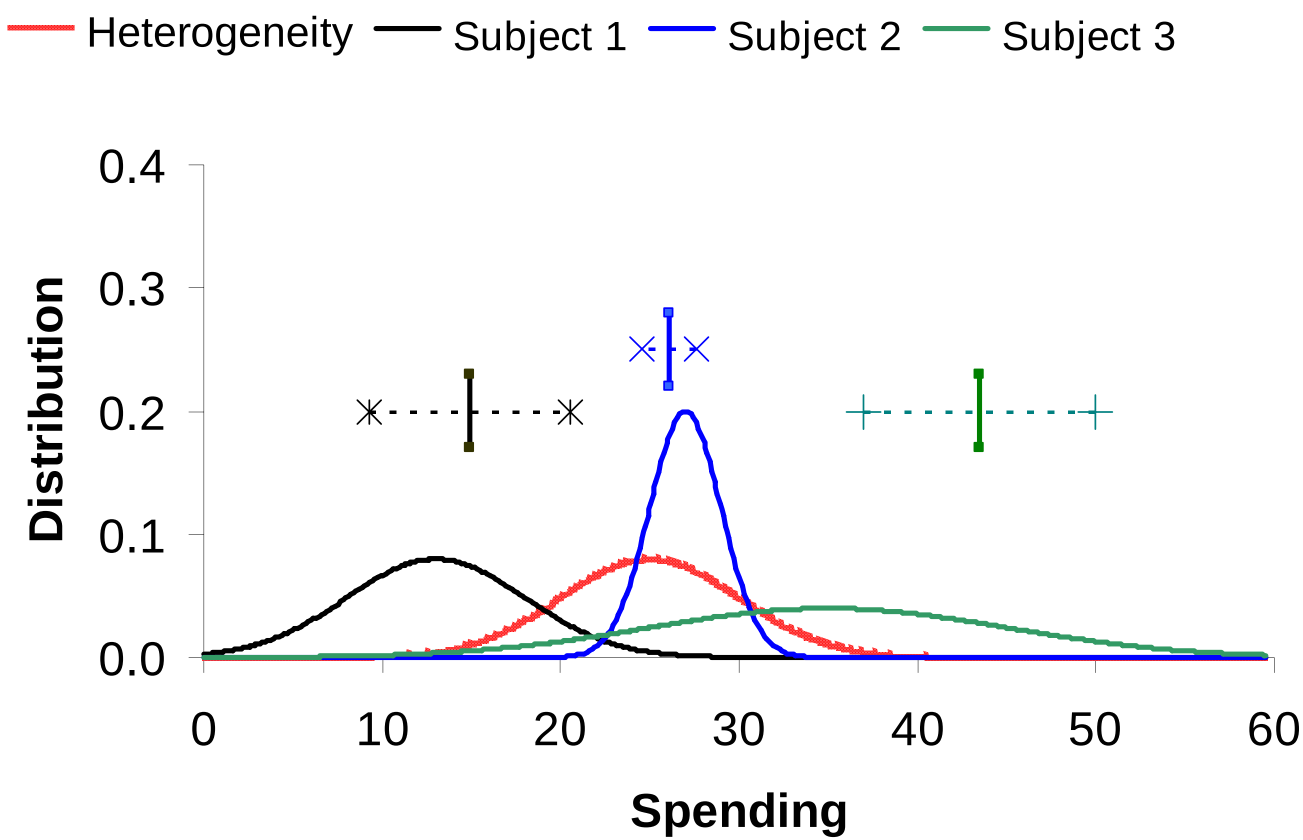
Between & Within Subjects Distributions



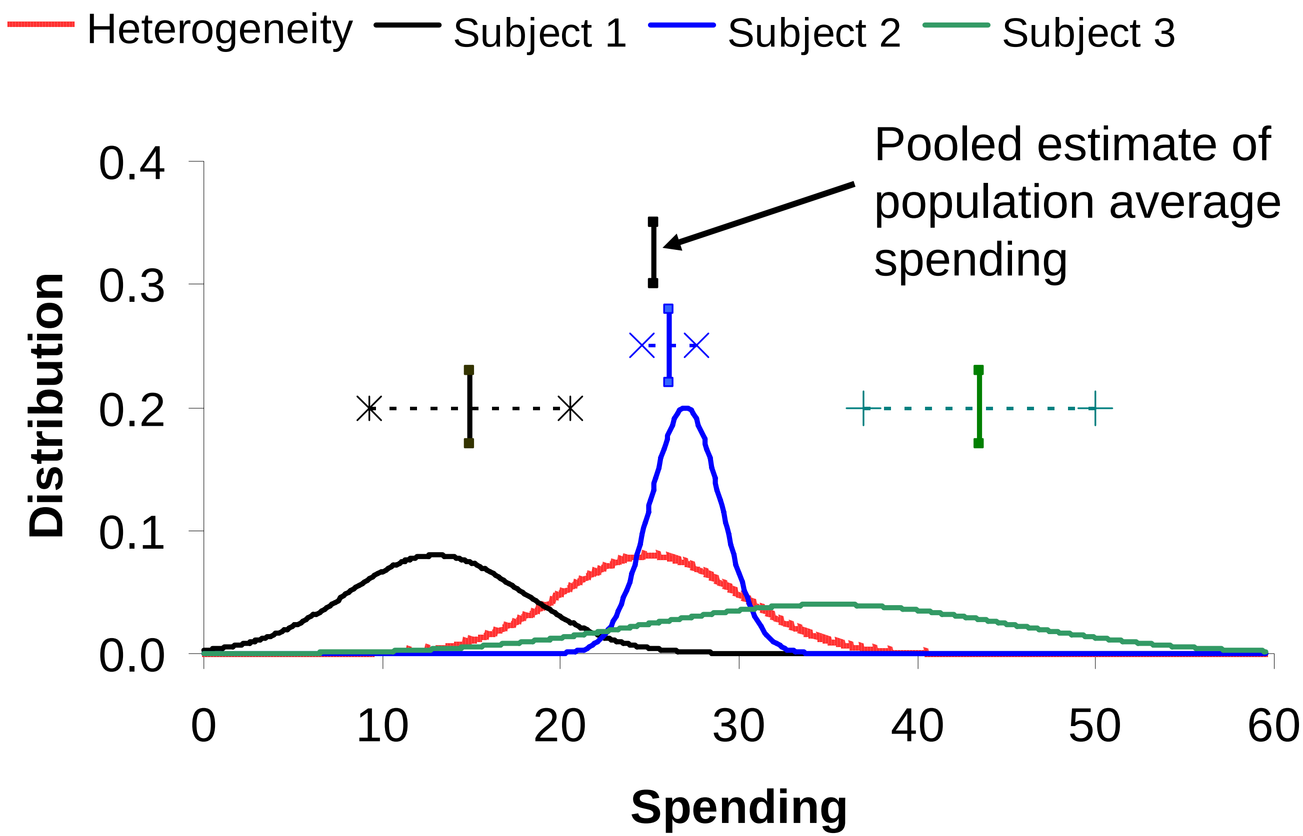
2 Observations per Subject



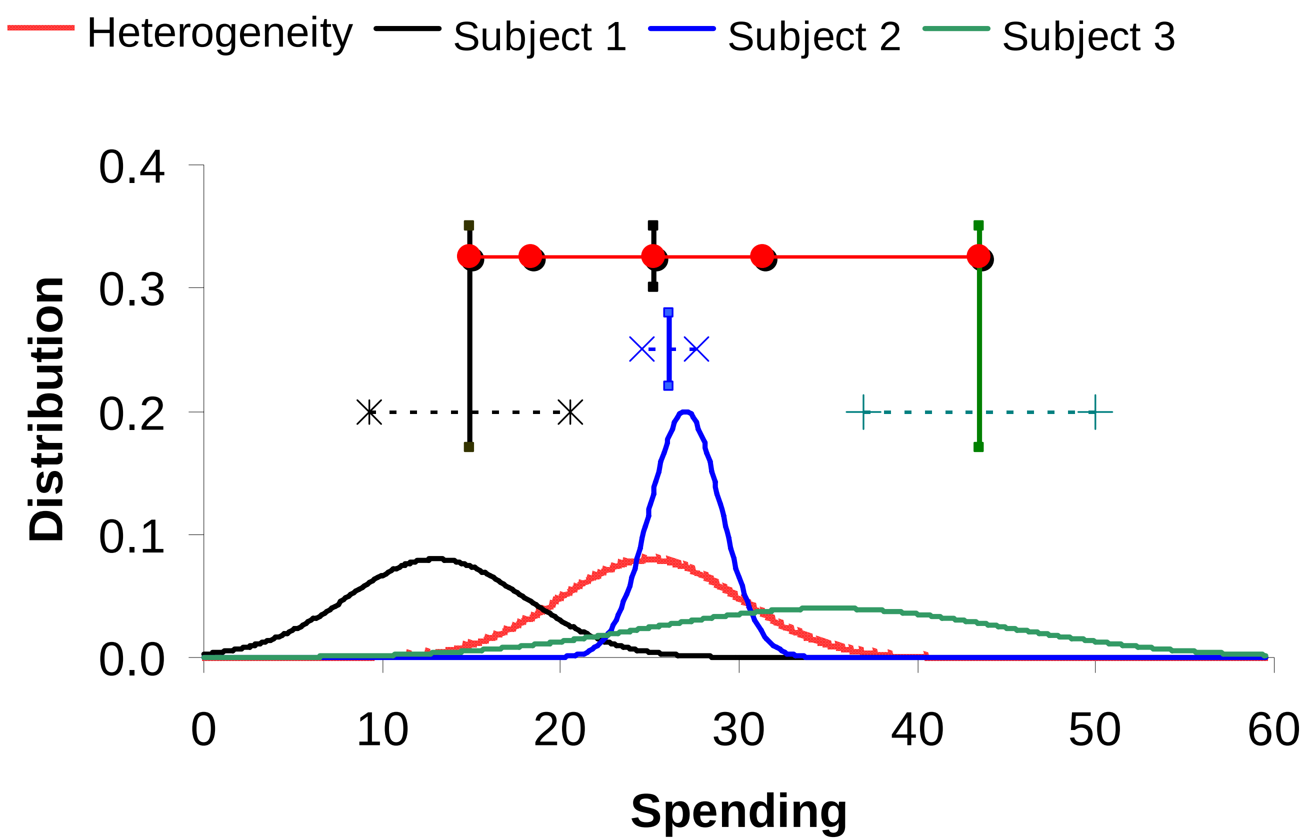
Subject Averages



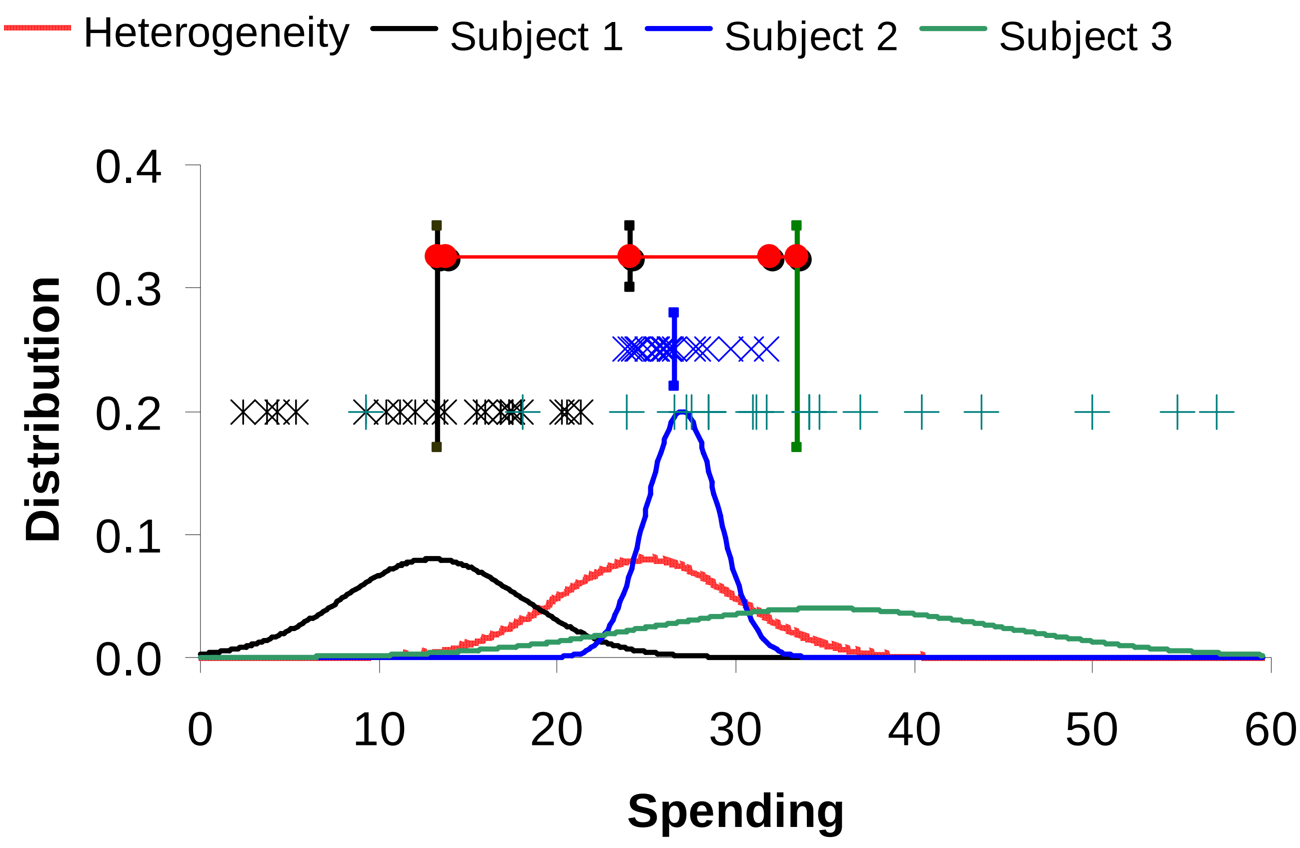
Pooled Estimate of Mean



Shrinkage Estimates



20 Observations per Subject



Bayes & Shrinkage Estimates

- Bayes estimators automatically determine the optimal amount of shrinkage of household estimates to population estimate to minimize MSE
- Borrows strength from all subjects
- Tradeoff some bias for variance reduction
- Allows estimation of models with more parameters than observations!

HB Regression Example

- Metric conjoint
- PC Profiles
- 190 subjects
- 20 profiles per subject
- 7 binary attributes
- 0 to 10 scale for how likely to buy

Variables

- Y = likelihood of purchase
- Attributes
  - X1 = Intercept X2 = RAM
  - X3 = CPU X4 = Screen Size
  - X5 = Channel X6 = Warranty
  - X7 = Software X8 = Price
- Demographics
  - Z1 = Intercept Z2 = Female Dummy
  - Z3 = Nerd Dummy Z4 = Expertise

Model

- Within-Subjects: Subject i, Profile j
  - $Y_{ij} = x_j'\beta_i + \varepsilon_{ij}$ where $\varepsilon_{ij} \sim N(0,\sigma^2)$
  - $i = 1 \ldots n$ and $j = 1 \ldots J_i$
- Between-Subjects or population model
  - $\beta_i = \Theta' z_i + \delta_i$ where $\delta_i \sim N(0,\Lambda)$
- Priors
  - Θ is matrix normal
  - Λ is inverted Wishart
  - σ^2 is inverted gamma

MCMC Full Conditionals

- Joint distribution

$$\underbrace{[Y_1|\beta_1,\sigma^2]\ldots[Y_n|\beta_n,\sigma^2]}_{\boxed{\text{Within Subjects}}}\ \underbrace{[\beta_1|\Theta,\Lambda]\ldots[\beta_n|\Theta,\Lambda]}_{\boxed{\text{Heterogeneity}}}\ \underbrace{[\sigma^2][\Theta][\Lambda]}_{\boxed{\text{Priors}}}$$

- Full conditional of β_i for i = 1…n

$$[\beta_i|\text{ Rest }] \propto [Y_i|\beta_i,\sigma^2]\,[\beta_i|\Theta,\Lambda] \sim \text{Normal}$$

MCMC Full Conditionals (2)

- Full Conditional of Θ

  $[\Theta|$ Rest $] \propto [\beta_1|\Theta,\Lambda]\ldots [\beta_n|\Theta,\Lambda]\ [\Theta]$ ~ Normal

- Full Conditional of Λ

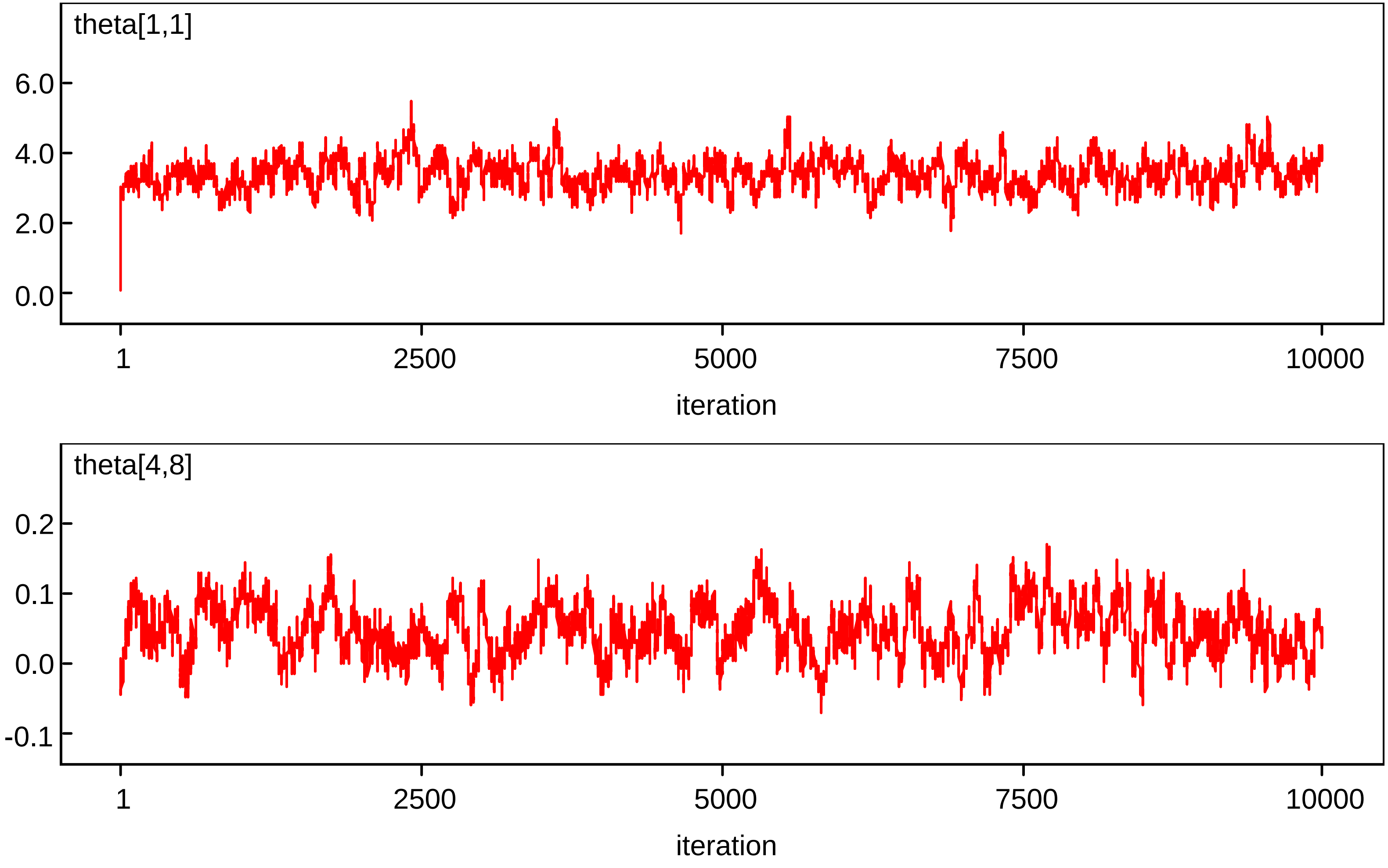
  $[\Lambda|$ Rest $] \propto [\beta_1|\Theta,\Lambda]\ldots [\beta_n|\Theta,\Lambda]\ [\Lambda]$ ~ Inverted Wishart

- Full Conditional of σ^2

  $[\sigma^2\ |$ Rest $] \propto [Y_1|\beta_1,\sigma^2]\ldots [Y_n|\beta_n,\sigma^2]\ [\sigma^2\ ]$
  ~ Inverted Gamma

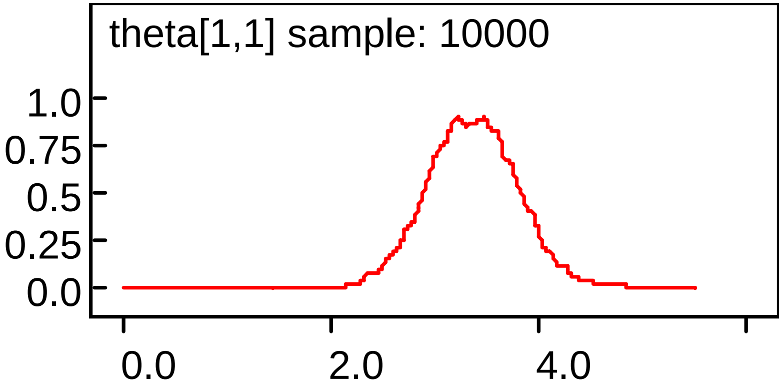
MCMC Iterations

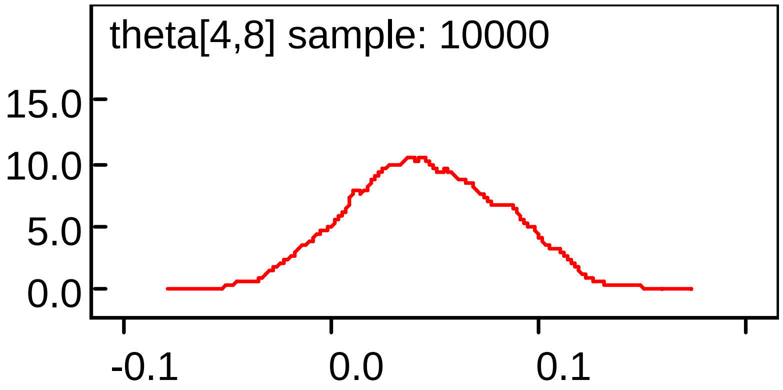
Constant and Coefficient Price x Expert



Posterior Density

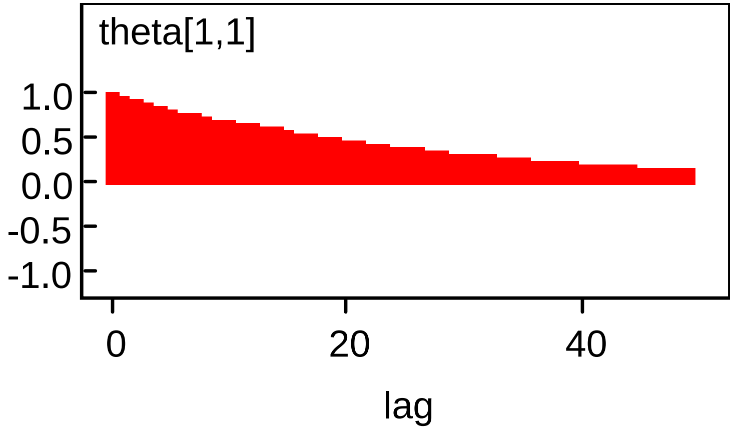
Constant and Price x Expert

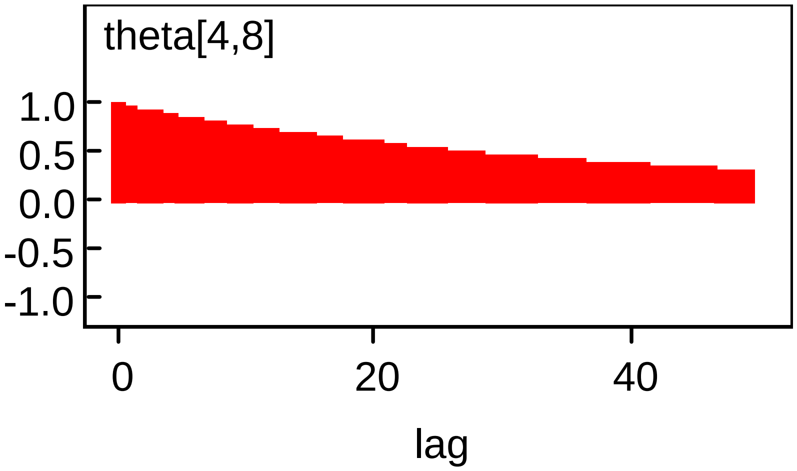




ACF

Constant and Price x Expert





Estimates

Constant and Price x Expert

node	mean	sd	MC error	2.5%	median	97.5%
theta[1,1]	3.377	0.448	0.02831	2.514	3.373	4.281
theta[4,8]	0.047	0.037	0.002639	-0.023	0.045	0.118

Metropolis Algorithm
Baseball Example

- 90 MLB Players in 2000 season.
- Observe at bats (AB) and hits (BA) in May
- Infer distribution of batting averages across players.
- Predict batting averages over season using data from May.

Baseball Batting Averages

Player	May			Season		
	At Bats	Hits	Batting Avg	At Bats	Hits	Batting Avg
Martinez	99	22	0.222	569	147	0.258
Jeter	**54**	**15**	**0.278**	**593**	**201**	**0.339**
O'Neil	109	29	0.266	566	160	0.283
Williams	106	30	0.283	537	165	0.307
Pasada	**82**	**27**	**0.329**	**505**	**145**	**0.287**

In Major League Baseball a batting average above 0.300 is outstanding!

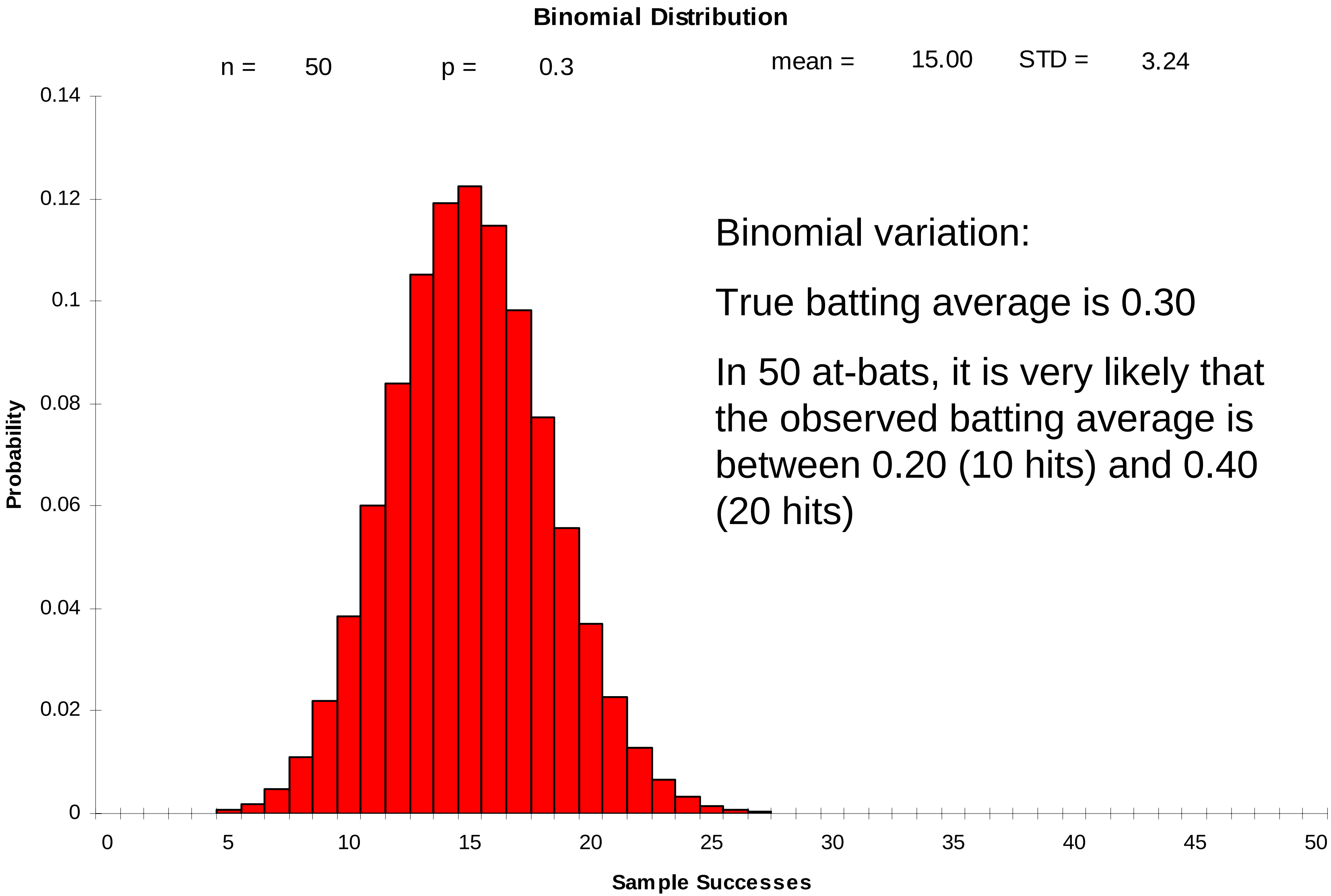
Batting averages around 0.250 are typical.

Notation

- Player i for i = 1...N
- n_i at-bats in may
- x_i hits in May
- x_i/n_i observed batting average
- p_i “true” batting average for player i
  - Assumed to be constant over time and unobserved
  - Varies across players

Model

- $x_i \mid p_i, n_i$ is Binomial(p_i,n_i)
  - This is a strong assumption
- p_i varies across players according to a Beta distribution with parameters α and β
  - $p_i \sim$ Beta(α,β)
- Priors
  - $\alpha \sim$ Gamma(r,s)
  - $\beta \sim$ Gamma(u,v)



Binomial variation:

True batting average is 0.30

In 50 at-bats, it is very likely that the observed batting average is between 0.20 (10 hits) and 0.40 (20 hits)

Beta Distribution

$$f\left[p \mid \alpha, \beta\right] = Beta(p \mid \alpha, \beta)$$

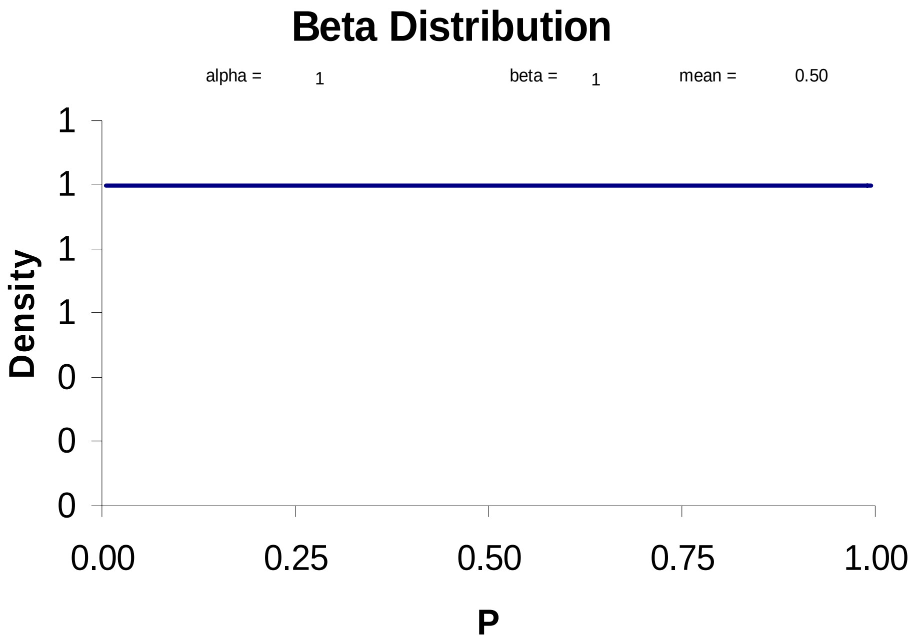
$$= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha - 1}(1 - p)^{\beta - 1} \quad for \quad 0 \le p \le 1$$

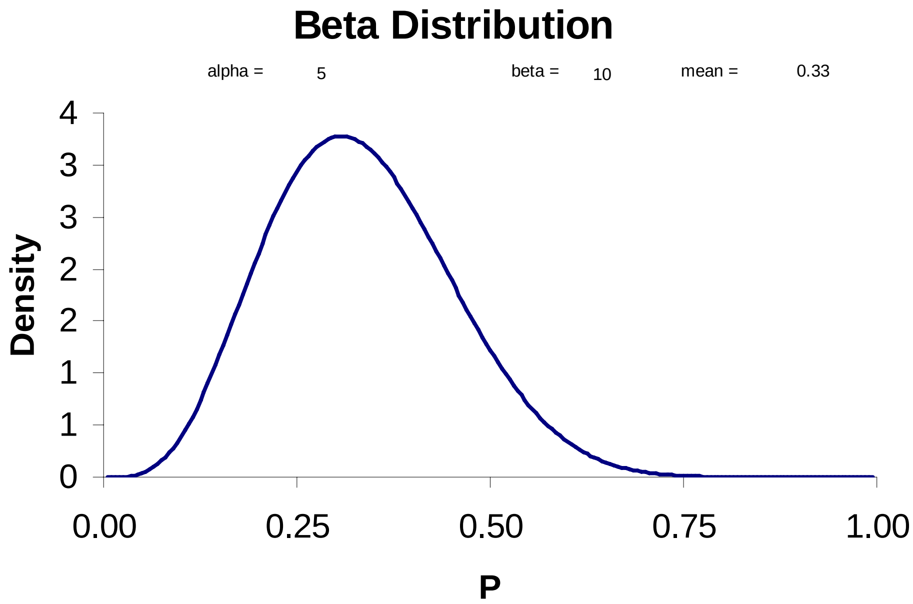
$$\Gamma(\alpha) = \int_0^\infty x^{\alpha - 1} e^{-x} dx$$

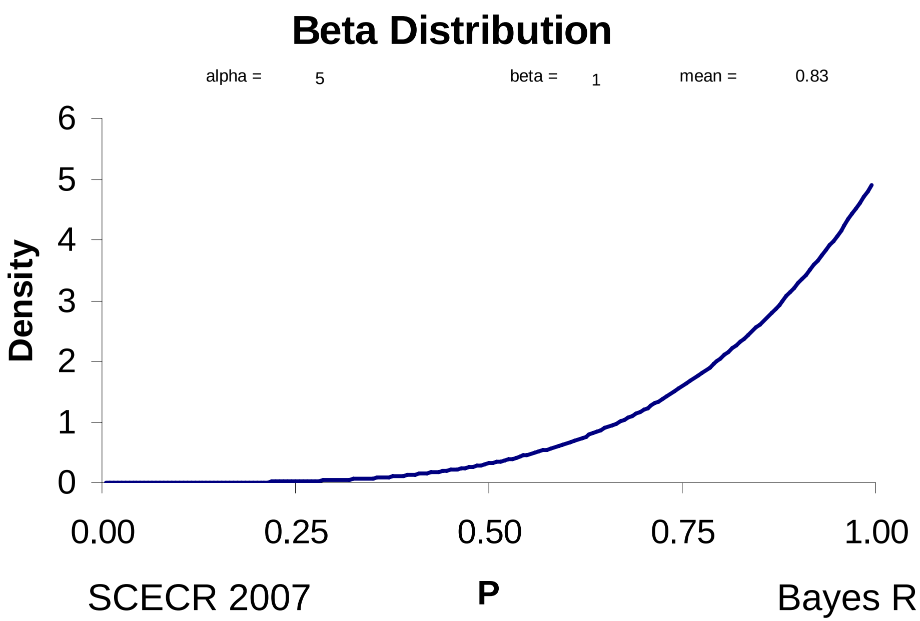
Mean and Variance

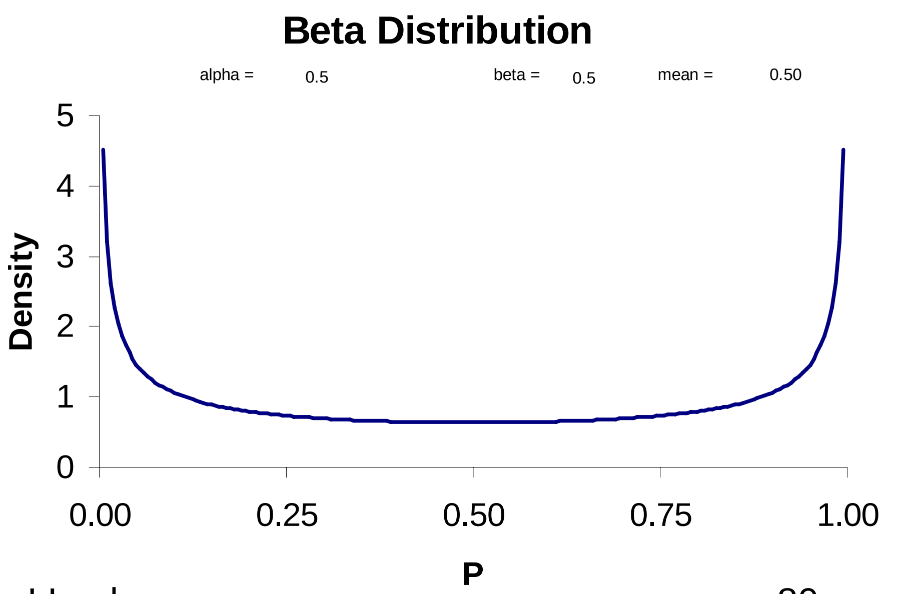
$$E(p) = \frac{\alpha}{\alpha + \beta}$$

$$V(p) = \frac{E(p)\left[1 - E(p)\right]}{\alpha + \beta + 1}$$


Beta Distribution
alpha = 1
beta = 1
mean = 0.50
Density
P
0.00
0.25
0.50
0.75
1.00


Beta Distribution
alpha = 5
beta = 10
mean = 0.33
Density
P
0.00
0.25
0.50
0.75
1.00


Beta Distribution
alpha = 5
beta = 1
mean = 0.83
Density
P
0.00
0.25
0.50
0.75
1.00

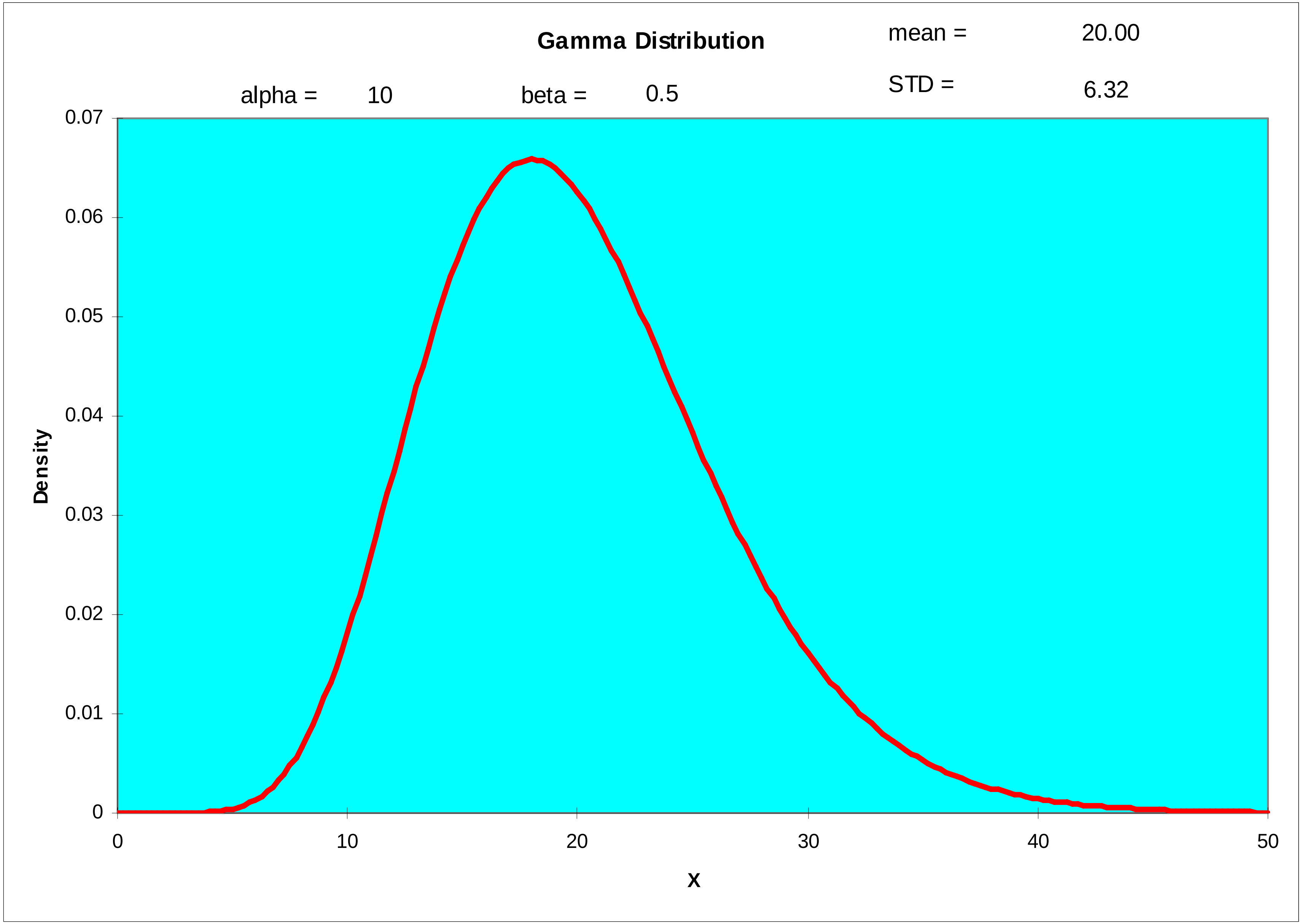

Beta Distribution
alpha = 0.5
beta = 0.5
mean = 0.50
Density
P
0.00
0.25
0.50
0.75
1.00

Gamma Distribution

$$g(y) = G(y \mid r, s)$$

$$= \frac{s^r}{\Gamma(r)} y^{r-1} e^{-sy} \quad for \quad y > 0$$

$$E(Y) = \frac{r}{s} \quad and \quad V(Y) = E(Y)\frac{1}{s}$$


Gamma Distribution
mean = 20.00
STD = 6.32
alpha = 10
beta = 0.5
Density
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
10
20
30
40
50
x

Specify Prior Parameters: r, s, u & v

Skip to MCMC

- Priors: α is G(r,s) & β is G(u,v).
- E(α) = r/s and V(α) = E(α)/s.
- s determines variance relative to mean.
  - I used s = 0.25 or the variance is four times larger than the mean.
  - Same for v.

$$E(p) = E\left[E(p \mid \alpha, \beta)\right]$$
$$= \frac{r}{r+u}$$
$$= p_0$$

$$c = V\left[E(p \mid \alpha, \beta)\right] = \frac{p_0(1-p_0)}{r+u+1}$$

$$r = p_0\left(\frac{p_0(1-p_0)}{c} - 1\right) \quad and$$

$$u = (1-p_0)\left(\frac{p_0(1-p_0)}{c} - 1\right)$$

Specify Prior Parameters

- Guess a mean of all batting averages:
  $$p_0 = 0.25$$
- Measure of my uncertainty of that guess:
  $$c = 0.01$$
- Parameter $r = 4.4$ and $s = 0.25$
- Parameter $u = 13.3$ and $v = 0.25$

MCMC for Batting Averages

- Need full conditionals for p_i give α and β
  - ➢Beta distribution
- Need full conditionals for α and β given p_i .
  - ➢Unknown distribution
  - ➢Use Metropolis algorithm

MCMC: Full Conditionals for Player i Batting Average p_i

$$f\left[p_i \mid x_i, \alpha, \beta\right] \propto \Pr\left(x_i \mid p_i\right) f\left(p_i \mid \alpha, \beta\right)$$

$$\propto p_i^{a+x_i-1}\left(1-p_i\right)^{\beta+n_i-x_i-1}$$

$$= Beta\left(p_i \mid \alpha + x_i, \beta + n_i - x_i\right)$$

MCMC: Full Conditional for α and β

$$g\left(\alpha, \beta \mid x_1, \ldots x_n, p_1, \ldots p_n\right)$$

$$\propto g\left(\alpha \mid r, s\right) g\left(\beta \mid u, v\right) \left[\frac{\Gamma\left(\alpha + \beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\alpha\right)} \right]^n \prod_{i=1}^{n} p_i^{\alpha-1}\left(1 - p_i\right)^{\beta-1}$$

Metropolis Algorithm

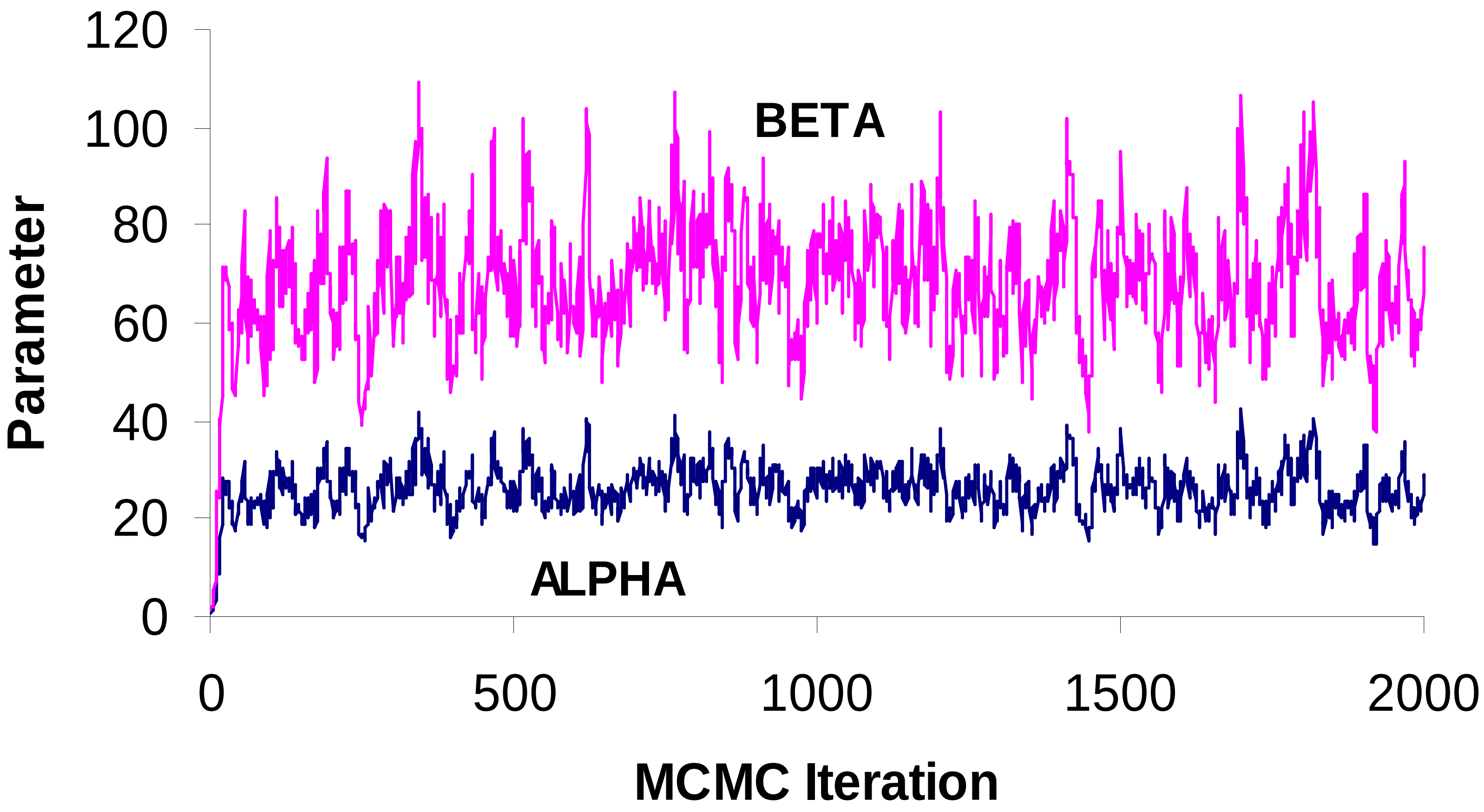
- Want to generate θ from f
- Instead, generate candidate value ϕ from g(.|θ)
  - Density g can depend on θ
  - eg Random walk: $\phi = \theta + \delta$
- With probability $\xi(\theta,\phi)$ accept ϕ as the new value of θ
- With probability $1-\xi(\theta,\phi)$ keep θ
- Markov chain with stationary distribution f

Transition Probability

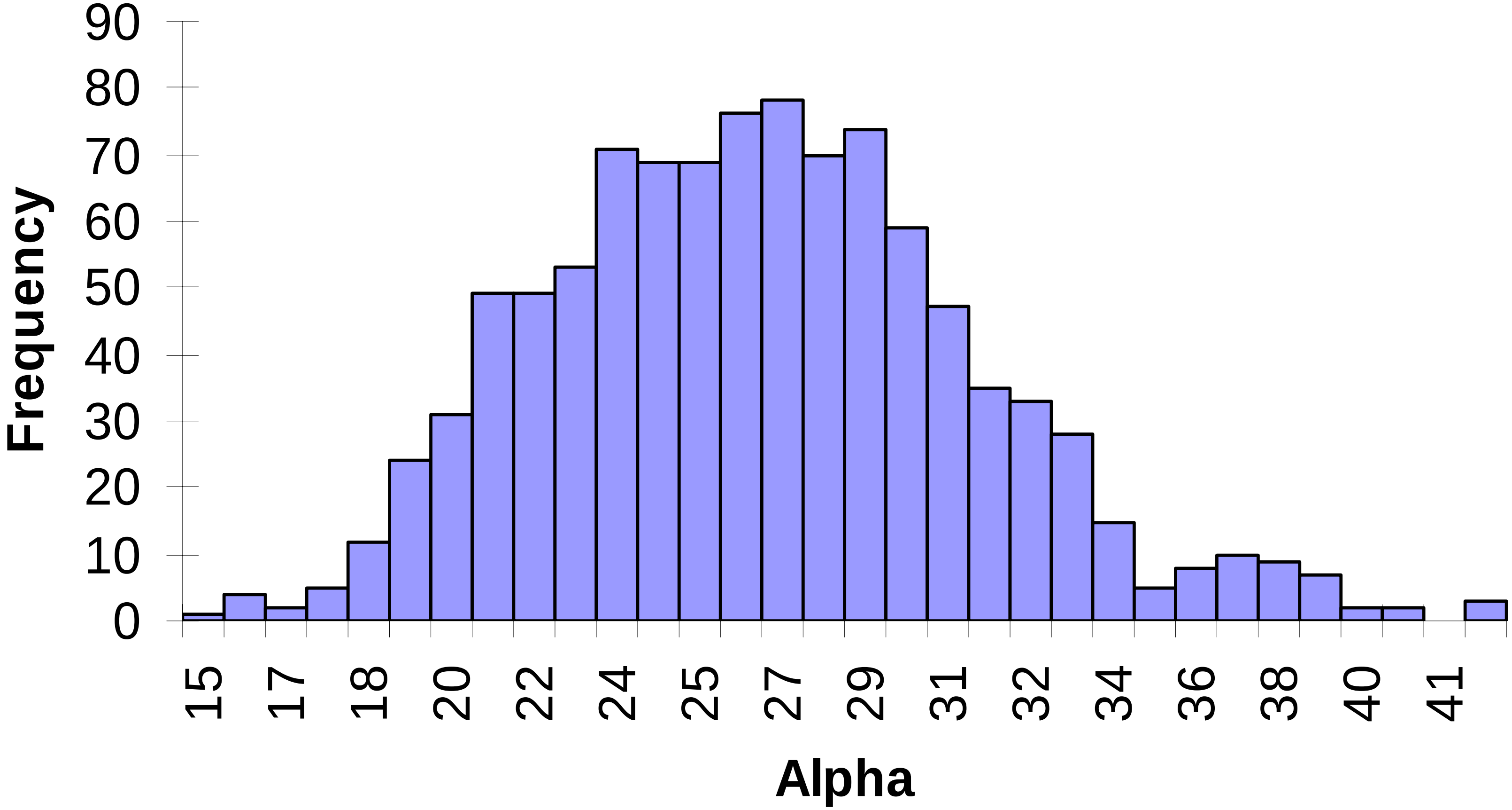
$$\xi(\theta,\varphi) = \min\left\{ \frac{f(\varphi)g(\theta \mid \varphi)}{f(\theta)g(\varphi \mid \theta)}, 1 \right\}$$

- f is the full conditional density of θ
- g is the generating density for ϕ
- Ratios: do not need to know constants
- “Works” if densities are not zero
- Works better if g is close to f

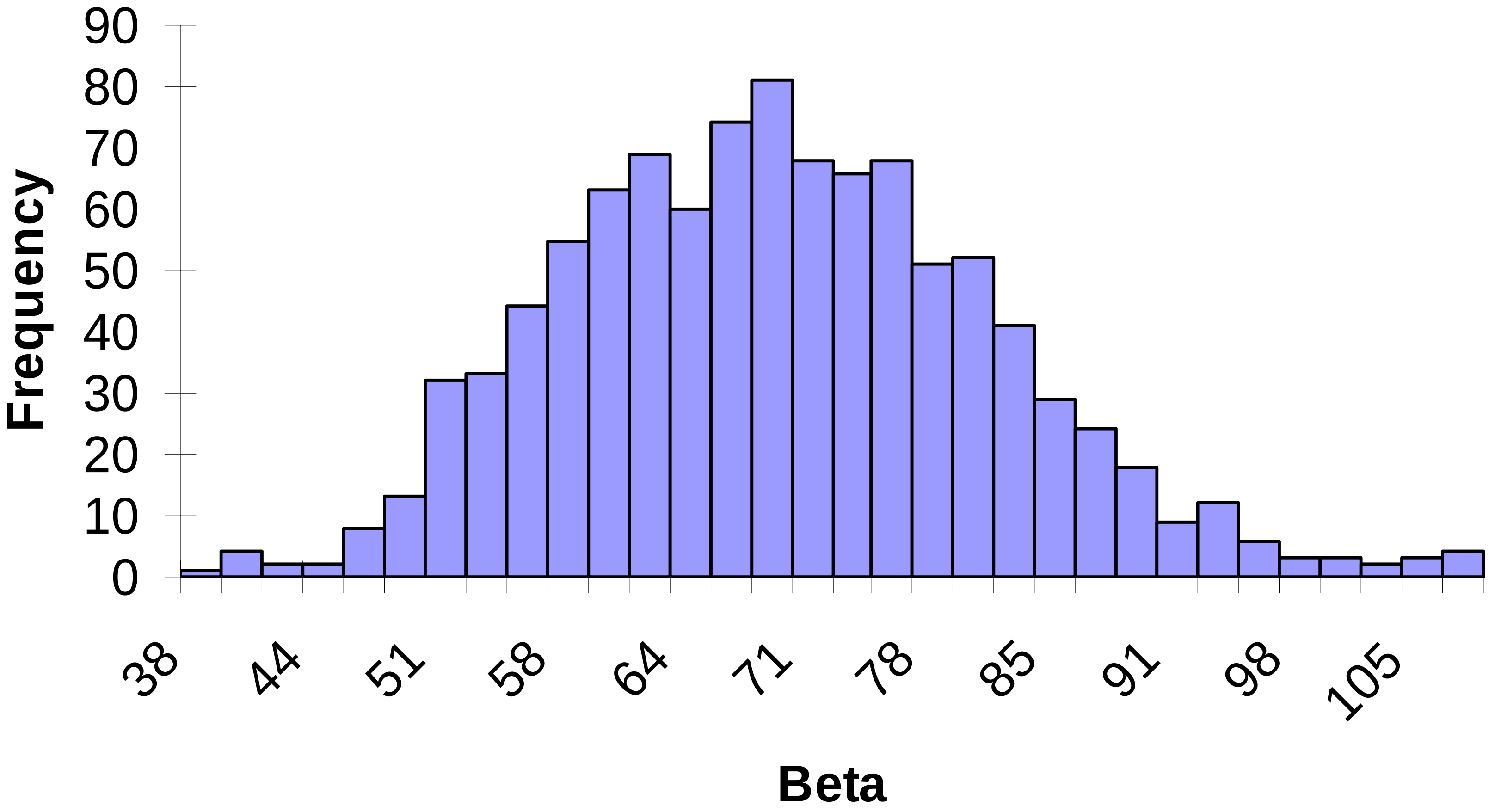
Alpha and Beta vs Iteration



Posterior of Alpha



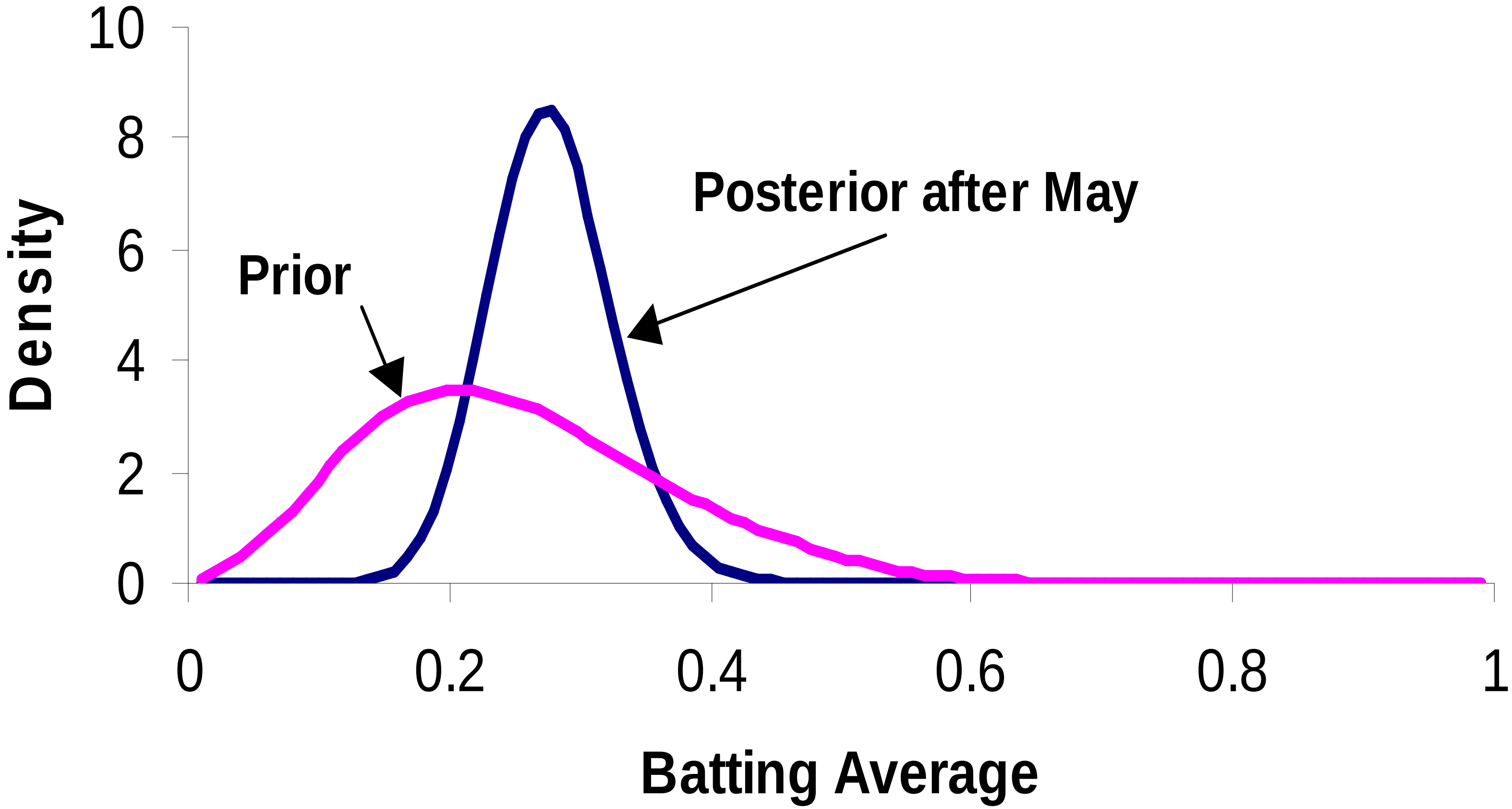
Posterior of Beta



Parameters Estimates

	Prior	Posterior
α	17.8	26.2
(std)	(8.4)	(4.6)
β	53.2	68.2
(std)	(14.6)	(11.7)

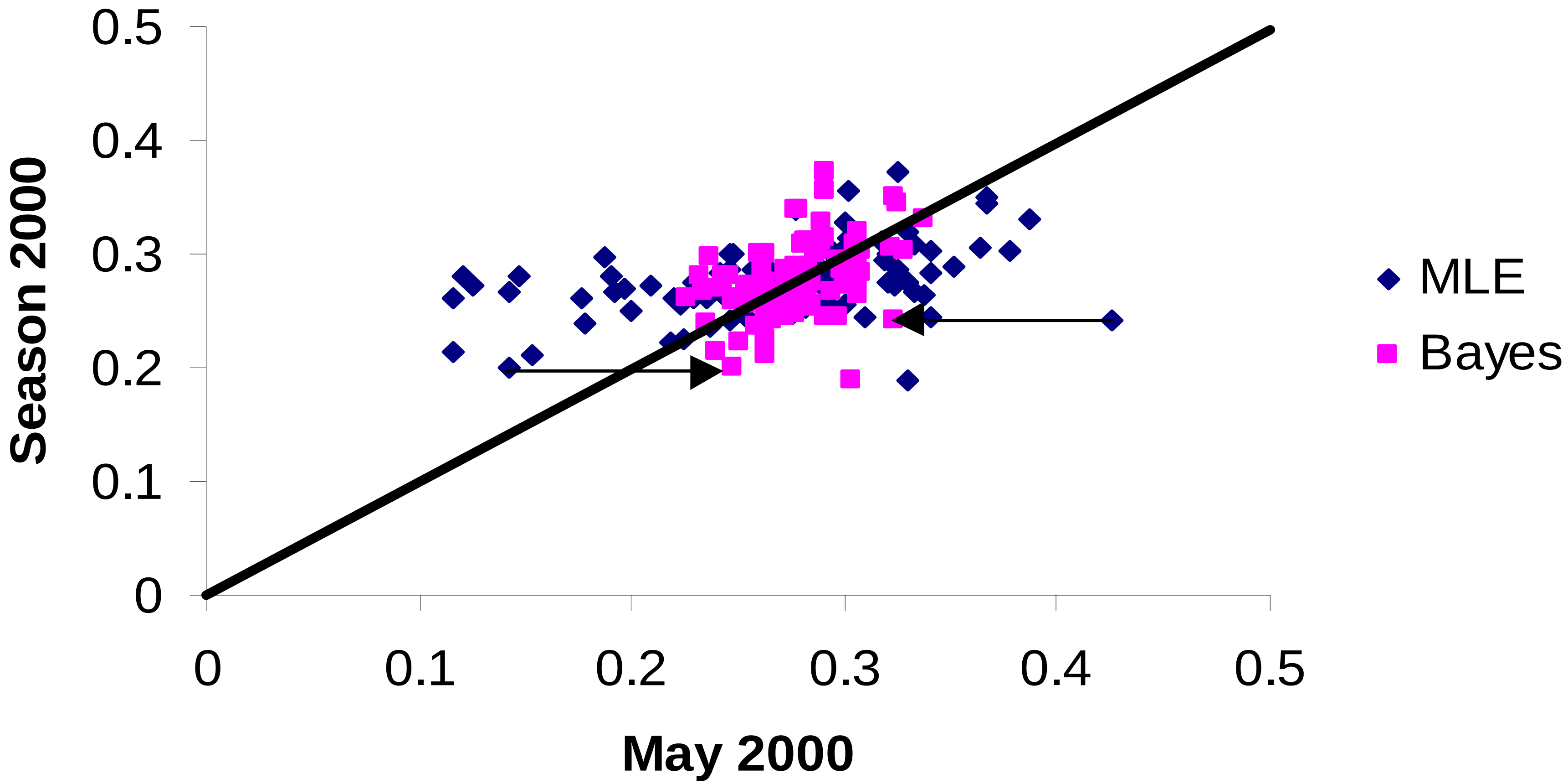
Distribution of Batting Averages



Prediction of Season Averages

	RMSE	MAPE
MLE	0.060	17.0%
Bayes	0.032	9.4%

Batting Averages
Bayes Shrinks MLE



Choice Based Conjoint Example

- Data provided by Sawtooth Software
- Joint work with Robert Zeithammer
- 326 IT purchasing manager
- 5 brands of personal computers
- 8 choice tasks per subject
- 4 alternatives per choice task
  - 3 brands and “None”
  - Choice tasks do not have every brand

Random Utility Model (RUM)

- Subject i has latent utility U for profile j in choice task k: $U_{i,j,k} = x'_{j,k}\beta_i + \varepsilon_{i,j,k}$
  - $x_{i,j,k}$ = attribute levels for profile j
  - β_i = individual level parameters
  - $\varepsilon_{i,j,k}$ = normal distribution with mean 0
    - Errors are associated with brands
    - Σ is error covariance among brand preferences
- Pick profile j* if $U_{i,j^*,k} > U_{i,j,k}$
  - Observed data are the choices
  - Utility “None” = 0

Heterogeneity

- Individual-level parameters follow a multivariate regression model
- $\beta_i = \Theta' z_i + \delta_i$
- Θ = matrix of regression coefficients
- z_i = observed covariates for subject i
- δ_i = multivariate normal errors
  - mean 0 and covariance Λ

Prior Distributions

- Σ, the error covariance matrix for the 5 brands preferences, has an inverted Wishart distribution
- Θ, the regression parameters for heterogeneity, has a matrix normal distribution
- Λ, the error covariance for heterogeneity, has an inverted Wishart distribution.

Probit Model

- Probit probabilities are hard to compute
- Easy fix:
  - Generate latent utilities at each stage
  - Given latent utilities, which are multivariate normal distribution, the rest is easy
- Generate latent utilities
  - Truncated normal distributions
  - Utility for selected profile > Other Utilities.

Posterior Mean of Θ

	CNST	ExPayLow	ExPayHig	Expert	Female	SmallCo	LargeCo
BrandA	0.768				-0.421		0.302
BrandB	0.882		-0.382		-0.406		
BrandC	0.459	0.455	-0.458		-0.471		
BrandD	0.400		-0.584		-0.544		
BrandE			-0.597	-0.354	-0.691		
LowPerfo	-1.574						-0.326
HighPerf	0.566		0.267			0.371	
TeleBuy	-0.192	0.231					
SiteBuy		0.328					
ShortWar							
LongWar	0.401						
MFGFix	-0.679	-0.399					
SiteFix	0.342						
Price2	0.315			-0.291			-0.291
Price3	-0.723	-0.296					
Price4	-0.977	-0.661			0.287		

Brand Preference Covariance Matrix

	BrandA	BrandB	BrandC	BrandD	BrandE
BrandA	1.02	0.01	-0.12	-0.16	-0.41
BrandB	0.01	0.95	0.08	-0.13	-0.45
BrandC	-0.12	0.08	1.18	-0.44	-0.53
BrandD	-0.16	-0.13	-0.44	1.21	-0.08
BrandE	-0.41	-0.45	-0.53	-0.08	1.00

If subject likes Brand A more than expected, he or she will like Brands D and E less than expected.

Not IIA

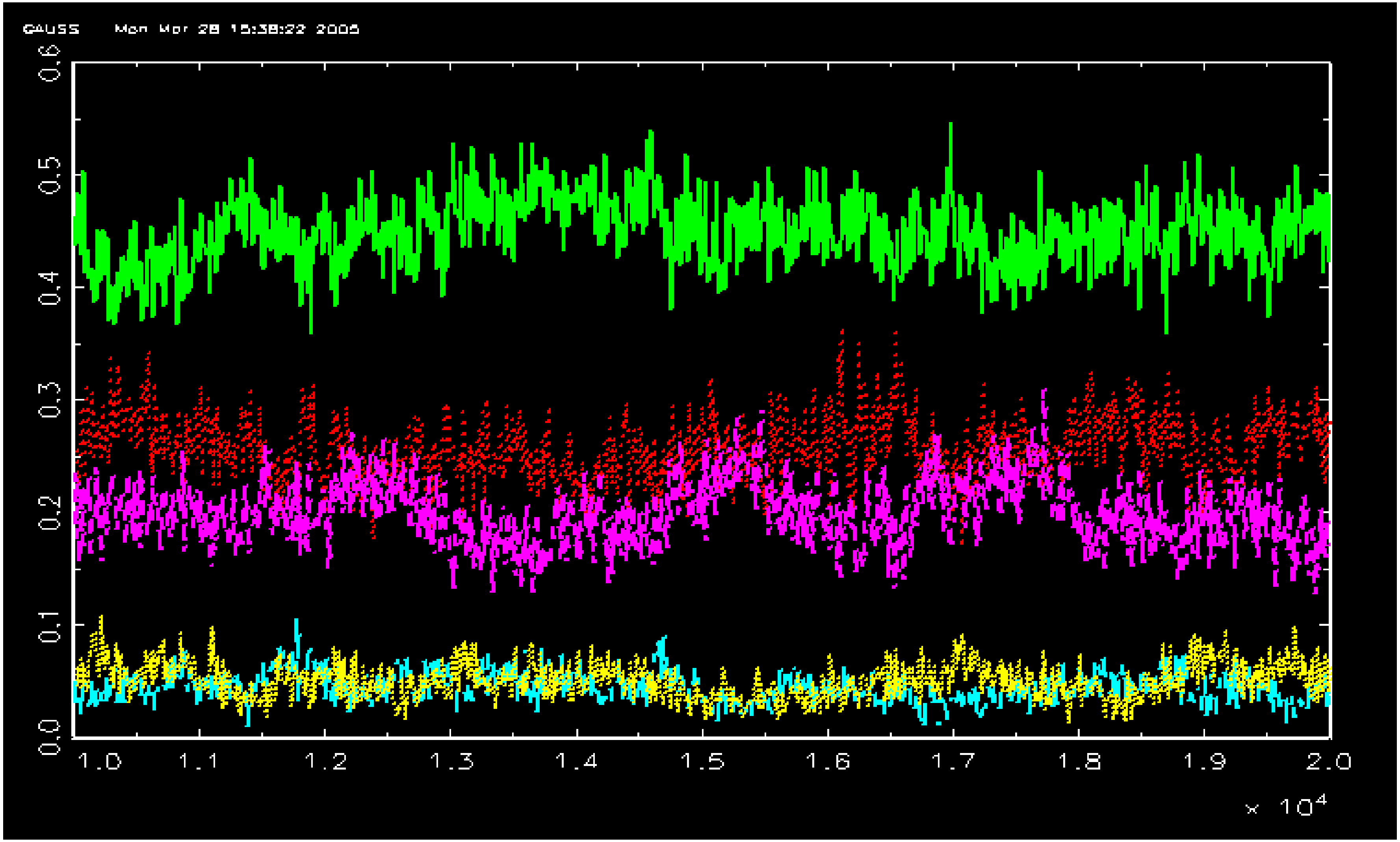
Attribute Significance

- Heterogeneity in partworths β_i
  - "Explained" $\Theta' z_i$
  - "Unexplained" δ_i
- $\theta_{uv} = 0$ is not enough to conclude insignificant partworth.
- Also need var(δ_i) close to zero

Simulated Market Share

- Fix 5 product specifications
- During each iteration
  - Generate subject's latent utility for each product
  - Pick the product with maximum utility
  - Compute market share
- Distribution of market shares

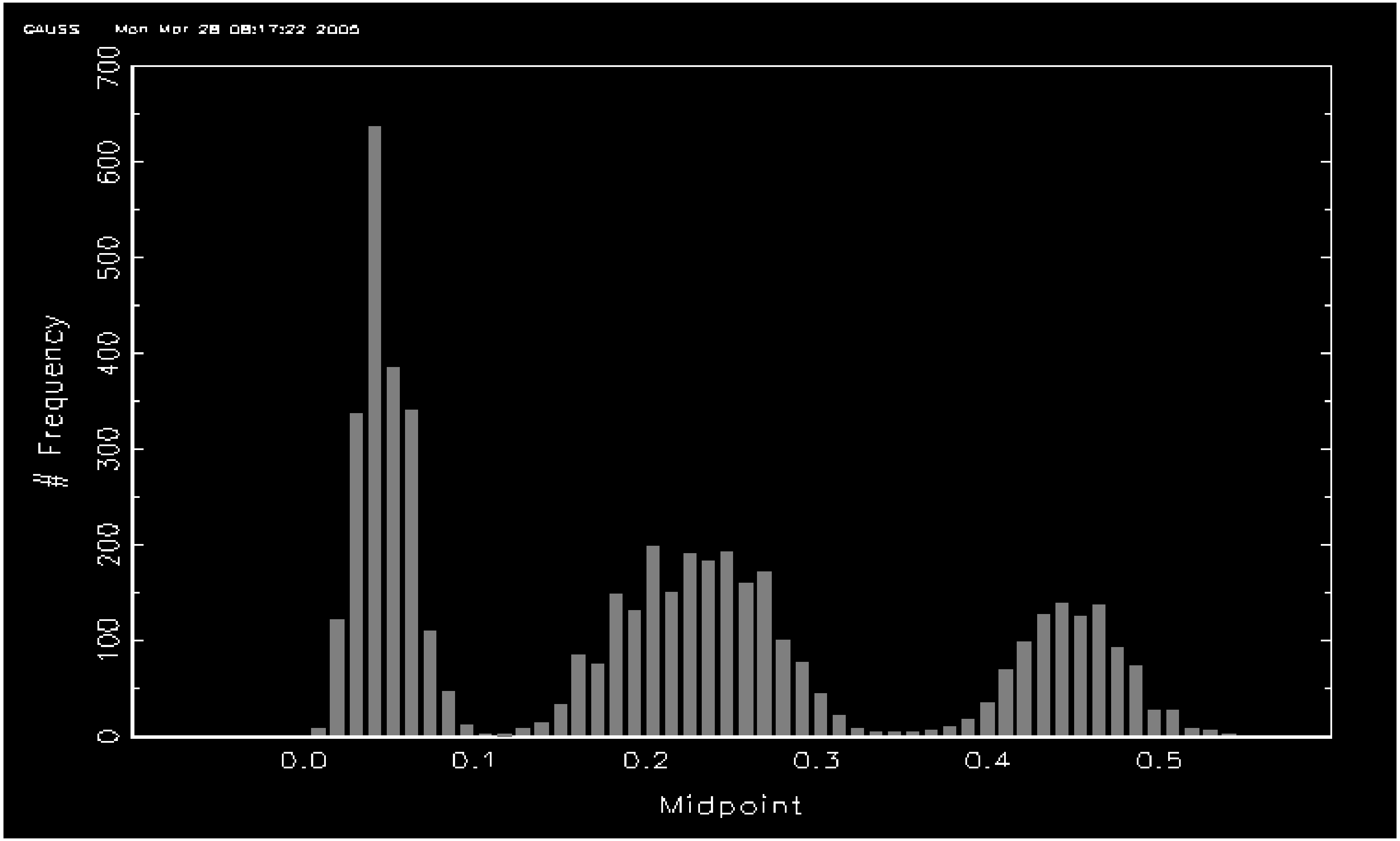
Iteration Plots of Market Shares



Posterior Means and STD DEV

Brand	Mean	STD DEV
A	0.45	0.030
B	0.04	0.014
C	0.26	0.028
D	0.20	0.029
E	0.05	0.015

Histogram of Posterior Distribution for Market Shares



WinBugs

- Free download for Bayesian inference
- Website
  http://www.mrc-bsu.cam.ac.uk/bugs/
- Brief overview of how to run examples provided with WinBugs
- Interpretation of output
- Trying it

Pros and Cons of WinBugs

- Pros
  - Slick software
  - Fairly flexible
  - Comprehensive output
  - Extensive series of examples
  - Large user community
  - Extensive documentation
- Cons
  - Needs some programming skills
  - Does not work for all problems
  - Not designed for "production runs"
  - May be too slow for very large datasets and complex models

WinBugs Language

- WinBugs does not have “canned” models
  - Unlike SAS or SPSS, you cannot use a pull down menu to run standard models
- WinBugs requires some “programming” to specify your model
  - For loops
  - Defining distributions
  - Matrices
  - Variable definitions
- You need to compile your model

Compound Documents

- Examples in manual are written in “compound documents” – way cool
- Text document
  - Describes the model and data
  - Includes model statement
  - Includes data file
  - Includes initial values
- WinBugs runs model from text document

Process

- Specify model
- Attach data
- Attach initial conditions
- Select “nodes” to monitor
- Run MCMC: Winbugs handles the details
  - Conjugate, log-convex, Metropolis
- Analyze output

Back to the Regression Example

- Movie box office revenue for 2004
  - http://www.boxofficeprophets.com
  - N = 349 releases
  - Y = ln(Total Revenue $m)
  - X1 = ln(Number Opening Screens)
  - X2 = ln(Opening Weekend Revenue $m)
- Log-log model
  - Y = b0 + b1*X1 + b2*X2 + e

See compound document "Movie BO.odc"

Prior Distributions

- Regression coefficients are normal with mean 0 and standard deviation 100
- 1/error variance is gamma with parameters 0.1 and 0.1

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Use Tools, Create Fold to insert new double arrow

If you want to work with the Doodle in this document, double click it and give it fuzzy borders.
Touch different nodes to open them up and see their properties.

Model Code

You can get the corresponding commands for the model.
Double click the above Doodle to wake it up. It will have fuzzy borders. Then
Menu, Doodle, Write Code.

A new window will open with the code for the model. Copy and past the code to this document.
Click the fold arrows to see the code. Very nice!

Model Code ➔

Data Definition

The next step is to define the da... ...ta include observations and parameter values. For
example, in the code "N" is the sam... ...size, and this value has to be assigned.

Double arrows hide text. Click it to view or hide.

There are two methods to enter data:
1) list statements that are similar to those in S+. S...
examples of structure and data statements for mat...
2) rectangular arrays

I will use a list statement for N, and a rectangular a...

Data List ➔⬅

Data Rectangular Array
⇨⇦

Note: Rectangular arrays end with "END", followed by a carriage return.

Data do not have to be included in the same document as the model. It is useful to do so if you

start | Re: ART For... | 2 Windows... | Bayes Tutori... | Presentation1 | WinBUGS14 ... | untitled - Paint | 4:00 PM

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Note. You cannot just highlight the Doodle to copy & pasted because every time you touch the screen with the mouse, you create a new node.

If you want to work with the Doodle in this document, double click it and give it fuzzy borders. Touch different nodes to open them up and see their properties.

Model Code

You can get the corresponding commands for the model.
Double click the above Doodle to wake it up. It will have fuzzy borders. Then
Menu, Doodle, Write Code.

A new window will open with the code for the model. Copy and past the code to this document.
Click the fold arrows to see the code. Very nice!

Model Code ⇨

```
model;
{
  for( i in 1 : N ) {
    TotalBO.ln[i] ~ dnorm(mu[i],y.per)
  }
  for( i in 1 : N ) {
    mu[i] <- b0 + b1 * Screens.ln[i] + b2 * OpenBO.ln[i]
  }
  b0 ~ dnorm( 0.0,1.0E-6)
  b1 ~ dnorm( 0.0,1.0E-6)
  b2 ~ dnorm( 0.0,1.0E-6)
  y.per ~ dgamma(0.001,0.001)
  y.std <- 1 / sqrt(y.per)
}
```

⇦

Data Definition

start | Re: ART Forum ... | 2 Windows Exp... | 2 Microsoft Off... | untitled - Paint | WinBUGS14 - [M... | 5:13 PM

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Data Definition

The next step is to define the data. Data include observations and parameter values. For example, in the code "N" is the sample size, and this value has to be assigned.

There are two methods to enter data:
1) list statements that are similar to those in S+. See Help, Examples Vol I or Vol II for lots of examples of structure and data statements for matrices.
2) rectangular arrays

I will use a list statement for N, and a rectangular array for the observations.

Data List ⇨list(N=349)⇦

Data Rectangular Array
➔

Screens.ln[]	OpenBO.ln[]	TotalBO.ln[]
8.334	4.683	6.079
8.331	4.752	5.923
8.021	4.429	5.914
8.166	3.831	5.632
8.277	4.255	5.566
8.257	4.540	5.519
8.139	4.452	5.230
8.060	3.961	5.171
8.012	3.559	5.153
8.202	3.149	5.092
8.298	3.863	5.080
8.137	3.955	4.975
8.135	3.847	4.892
8.099	3.667	4.833
8.186	3.809	4.794
8.182	3.946	4.788
6.766	3.175	4.780
8.194	3.403	4.776
7.899	3.404	4.739
8.224	3.927	4.738
8.085	3.667	4.702
3.689	-0.151	4.631
2.079	-1.715	4.609
8.067	3.207	4.605

start | Re: ART Forum 2... | 2 Windows Expl... | 2 Microsoft Offi... | WinBUGS14 - [Mo... | untitled - Paint | 4:03 PM

Step 1: Model Specification

- Tool bar
  - Model
  - Specification
- Highlight “model” at beginning of code
- Click “check model”
- Look for a happy message at bottom left

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Specification...
Update...
Monitor Met
Save State
Seed...
Script

screen with the mouse, you c .

If you want to work with the D ment, double click it and give it fuzzy borders.
Touch different nodes to ope e their properties.

Model Code

You can get the correspondin the model.
Double click the above Doodle to wake it up. It will have fuzzy borders. Then
Menu, Doodle, Write Code.

A new window will open with the code for the model. Copy and past the code to this document.
Click the fold arrows to see the code. Very nice!

Model Code ⇨

```
model;
{
  for( i in 1 : N ) {
    TotalBO.ln[i] ~ dnorm(mu[i],y.per)
  }
  for( i in 1 : N ) {
    mu[i] <- b0 + b1 * Screens.ln[i] + b2 * OpenBO.ln[i]
  }
  b0 ~ dnorm( 0.0,1.0E-6)
  b1 ~ dnorm( 0.0,1.0E-6)
  b2 ~ dnorm( 0.0,1.0E-6)
  y.per ~ dgamma(0.001,0.001)
  y.std <- 1 / sqrt(y.per)
}
```

⇦

Data Definition

The next step is to define the data. Data include observations and parameter values. For
example, in the code "N" is the sample size, and this value has to be assigned.

start | Re: ART Forum ... | 2 Windows Exp... | 2 Microsoft Off... | WinBUGS14 - [M... | untitled - Paint | 4:16 PM

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

screen with the mouse, you create a new node.

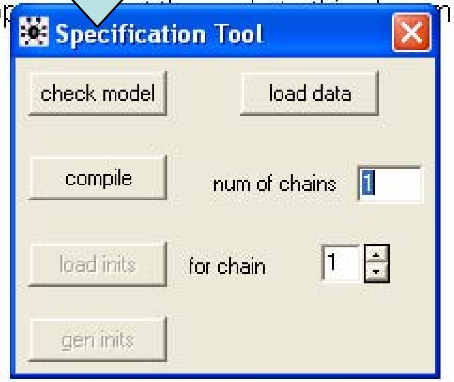
If you want to work with the Doodle in this document, double click it and give it fuzzy borders. Touch different nodes to open them up and see their properties.

Model Code

Highlight Model

You can g nds for the model.
Double cli e it up. It will have fuzzy ders. Then
Menu, Do

A new window will open with the code for the model. Cop ment.
Click the fold arrows to see the code. Very nice!



Model Code ⇨

```
model;
{
  for( i in 1 : N ) {
    TotalBO.ln[i] ~ dnorm(mu[i],y.per)
  }
  for( i in 1 : N ) {
    mu[i] <- b0 + b1 * Screens.ln[i] + b2 * OpenBO.ln[i]
  }
  b0 ~ dnorm( 0.0,1.0E-6)
  b1 ~ dnorm( 0.0,1.0E-6)
  b2 ~ dnorm( 0.0,1.0E-6)
  y.per ~ dgamma(0.001,0.001)
  y.std <- 1 / sqrt(y.per)
}
```

⇦

Data Definition

The next step is to define the data. Data include observations and parameter values. For example, in the code "N" is the sample size, and this value has to be assigned.

start | Re: ART Forum ... | 2 Windows Exp... | 2 Microsoft Off... | WinBUGS14 - [M... | untitled - Paint | 4:17 PM

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

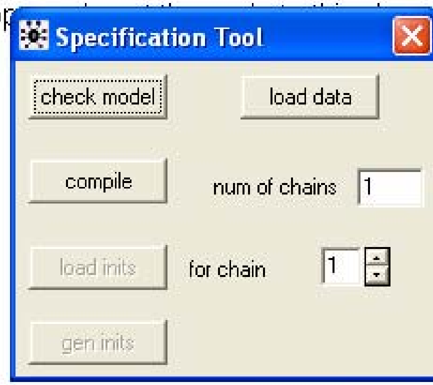
screen with the mouse, you create a new node.

If you want to work with the Doodle in this document, double click it and give it fuzzy borders.
Touch different nodes to open them up and see their properties.

Model Code

You can get the corresponding commands for the model.
Double click the above Doodle to wake it up. It will have fuzzy borders. Then
Menu, Doodle, Write Code.

A new window will open with the code for the model. Cop ... ment.
Click the fold arrows to see the code. Very nice!



Model Code

```
model;
{
  for( i in 1 : N ) {
    TotalBO.ln[i] ~ dnorm(mu[i],y.per)
  }
  for( i in 1 : N ) {
    mu[i] <- b0 + b1 * Screens.ln[i] + b2 * OpenBO.ln[i]
  }
  b0 ~ dnorm( 0.0,1.0E-6)
  b1 ~ dnorm( 0.0,1.0E-6)
  b2 ~ dnorm( 0.0,1.0E-6)
  y.per ~ dgamma(0.001,0.001)
  y.std <- 1 / sqrt(y.per)
}
```

Data Def on

The next s to define the data. Data include observations and parameter values. For
example, in the code "N" is the sample size, and this value has to be assigned.

model is syntactically correct

start Re: ART Forum ... 2 Windows Exp... 2 Microsoft Off... WinBUGS14 - [M... untitled - Paint 4:19 PM

Step 2: Load Data

- If you use data list
  - Highlight “list”
  - Click “load data”
- If data are in rectangular file
  - Make the window with the data active
  - Put cursor at beginning of file
  - Click “load data”
- You can have multiple loads

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Data Definition

The next step is to define the data. Data include observations and parameter values. For example, in the code "N" is the sample size, and this value has to be assigned.

There are two methods to enter data:
1) list statements that are similar to those in S+. See Help, Examples Vol I or Vol II for lots of examples of structure and data statements for matrices.
2) rectangular arrays

I will use a list statement for N, and a rectangular array for the observations.

Data List ⇨list(N=349)⇦

Data Rectangular Array

➔

Screens.ln[]	OpenBO.ln[]	TotalBO.ln[]
8.334	4.683	6.079
8.331	4.752	5.923
8.021	4.429	5.914
8.166	3.831	5.632
8.277	4.255	5.566
8.257	4.540	5.519
8.139	4.452	5.230
8.060	3.961	5.171
8.012	3.559	5.153
8.202	3.149	5.092
8.298	3.863	5.080
8.137	3.955	4.975
8.135	3.847	4.892
8.099	3.667	4.833
8.186	3.809	4.794
8.182	3.946	4.788
6.766	3.175	4.780
8.194	3.403	4.776
7.899	3.404	4.739
8.224	3.927	4.738
8.085	3.667	4.702
3.689	-0.151	4.631
2.079	-1.715	4.609

Specification Tool

check model load data

compile num of chains 1

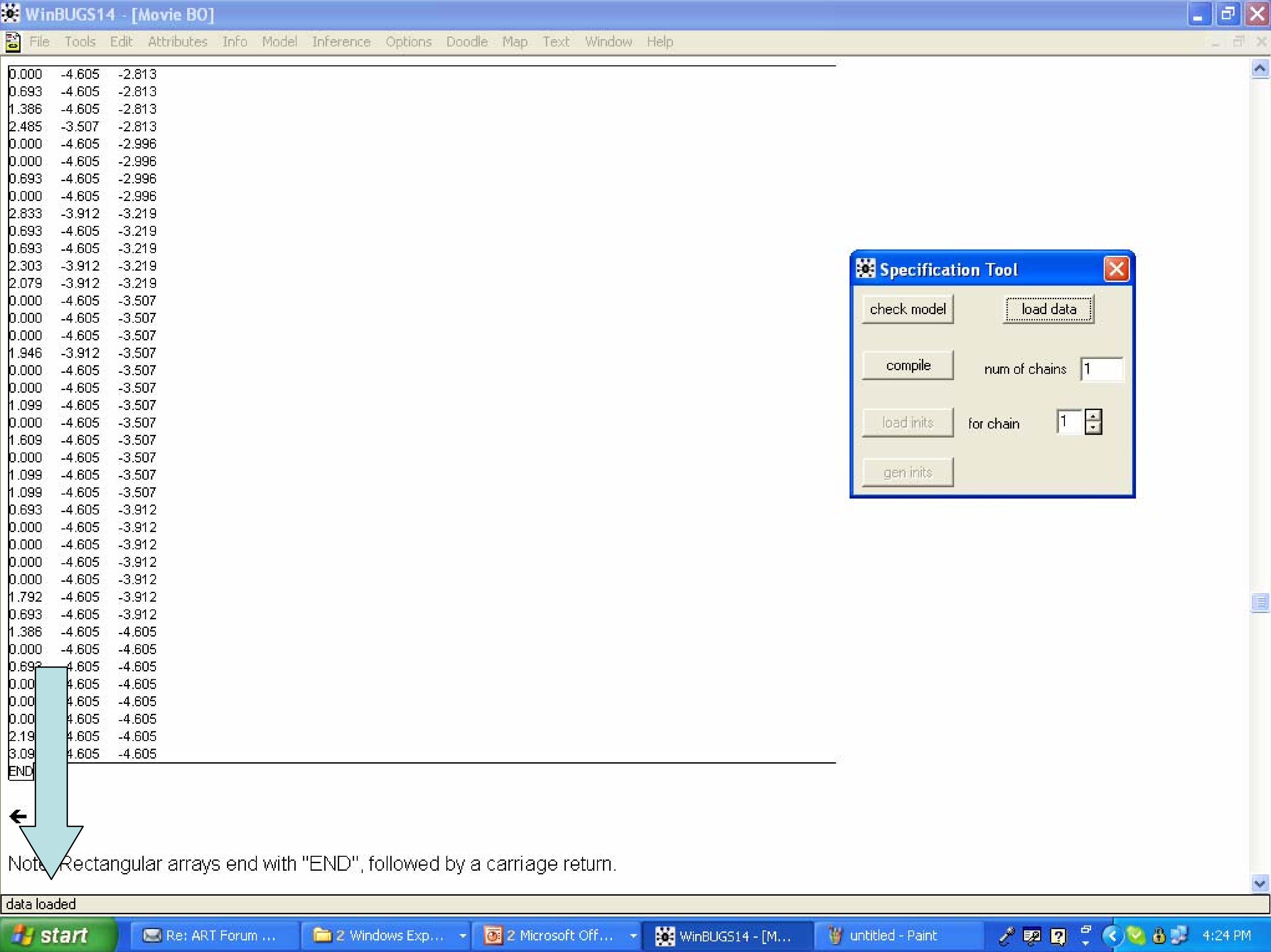
load inits for chain 1

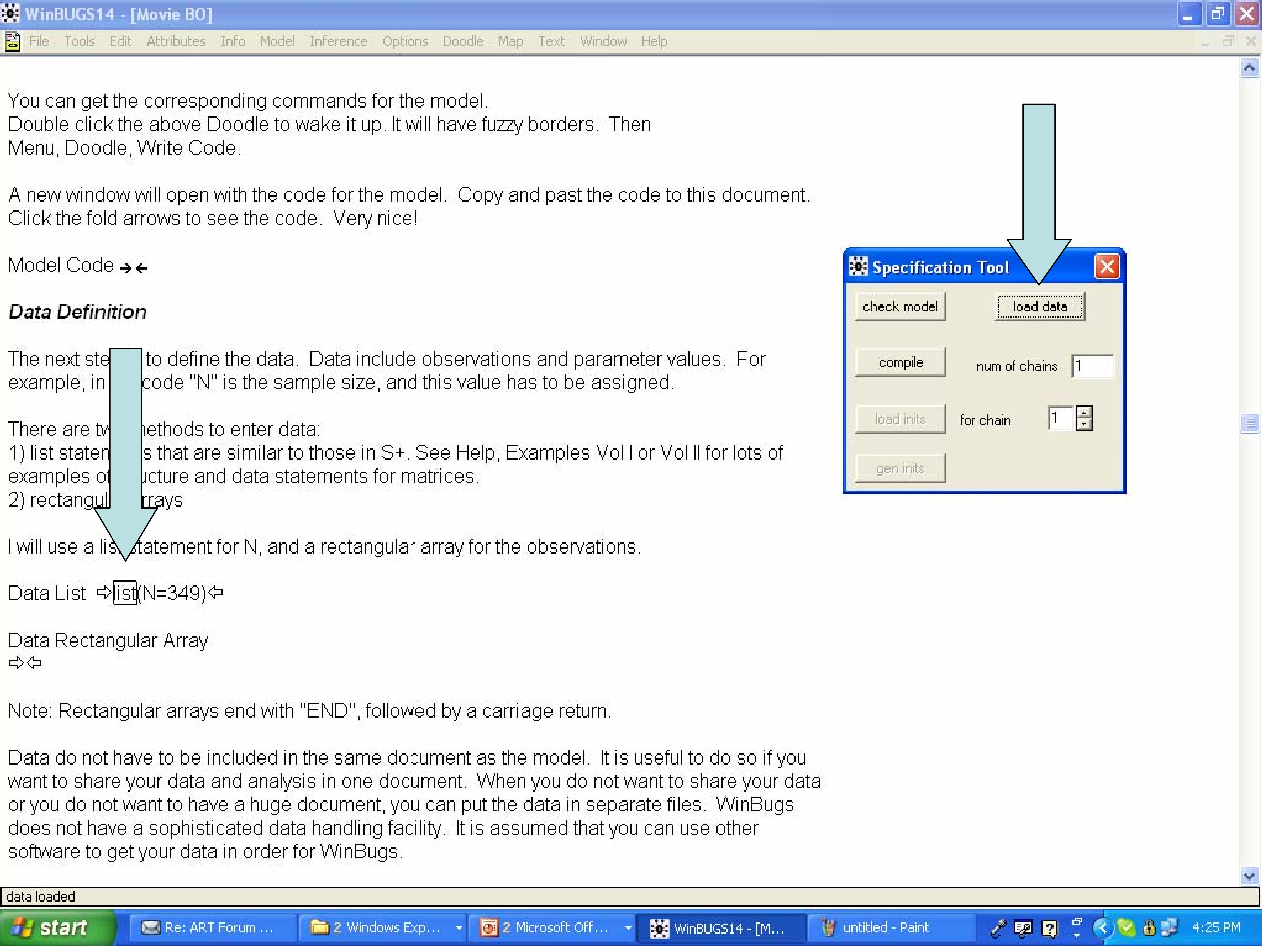
gen inits

Highlight the full array.

model is syntactically correct

start Re: ART Forum ... 2 Windows Exp... 2 Microsoft Off... WinBUGS14 - [M... untitled - Paint 4:21 PM


WinBUGS14 - [Movie BO]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
0.000 -4.605 -2.813
0.693 -4.605 -2.813
1.386 -4.605 -2.813
2.485 -3.507 -2.813
0.000 -4.605 -2.996
0.000 -4.605 -2.996
0.693 -4.605 -2.996
0.000 -4.605 -2.996
2.833 -3.912 -3.219
0.693 -4.605 -3.219
0.693 -4.605 -3.219
2.303 -3.912 -3.219
2.079 -3.912 -3.219
0.000 -4.605 -3.507
0.000 -4.605 -3.507
0.000 -4.605 -3.507
1.946 -3.912 -3.507
0.000 -4.605 -3.507
0.000 -4.605 -3.507
1.099 -4.605 -3.507
0.000 -4.605 -3.507
1.609 -4.605 -3.507
0.000 -4.605 -3.507
1.099 -4.605 -3.507
1.099 -4.605 -3.507
0.693 -4.605 -3.912
0.000 -4.605 -3.912
0.000 -4.605 -3.912
0.000 -4.605 -3.912
0.000 -4.605 -3.912
1.792 -4.605 -3.912
0.693 -4.605 -3.912
1.386 -4.605 -4.605
0.000 -4.605 -4.605
0.69 4.605 -4.605
0.00 4.605 -4.605
0.00 4.605 -4.605
0.00 4.605 -4.605
2.19 4.605 -4.605
3.09 4.605 -4.605
END
Note: Rectangular arrays end with "END", followed by a carriage return.
Specification Tool
check model
load data
compile
num of chains 1
load inits
for chain 1
gen inits
data loaded
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
WinBUGS14 - [M...
untitled - Paint
4:24 PM



You can get the corresponding commands for the model.
Double click the above Doodle to wake it up. It will have fuzzy borders. Then
Menu, Doodle, Write Code.

A new window will open with the code for the model. Copy and past the code to this document.
Click the fold arrows to see the code. Very nice!

Model Code

Data Definition

The next ste to define the data. Data include observations and parameter values. For
example, in code "N" is the sample size, and this value has to be assigned.

There are tw ethods to enter data:
1) list staten s that are similar to those in S+. See Help, Examples Vol I or Vol II for lots of
examples of ucture and data statements for matrices.
2) rectangul rrays

I will use a lis statement for N, and a rectangular array for the observations.

Data List list(N=349)

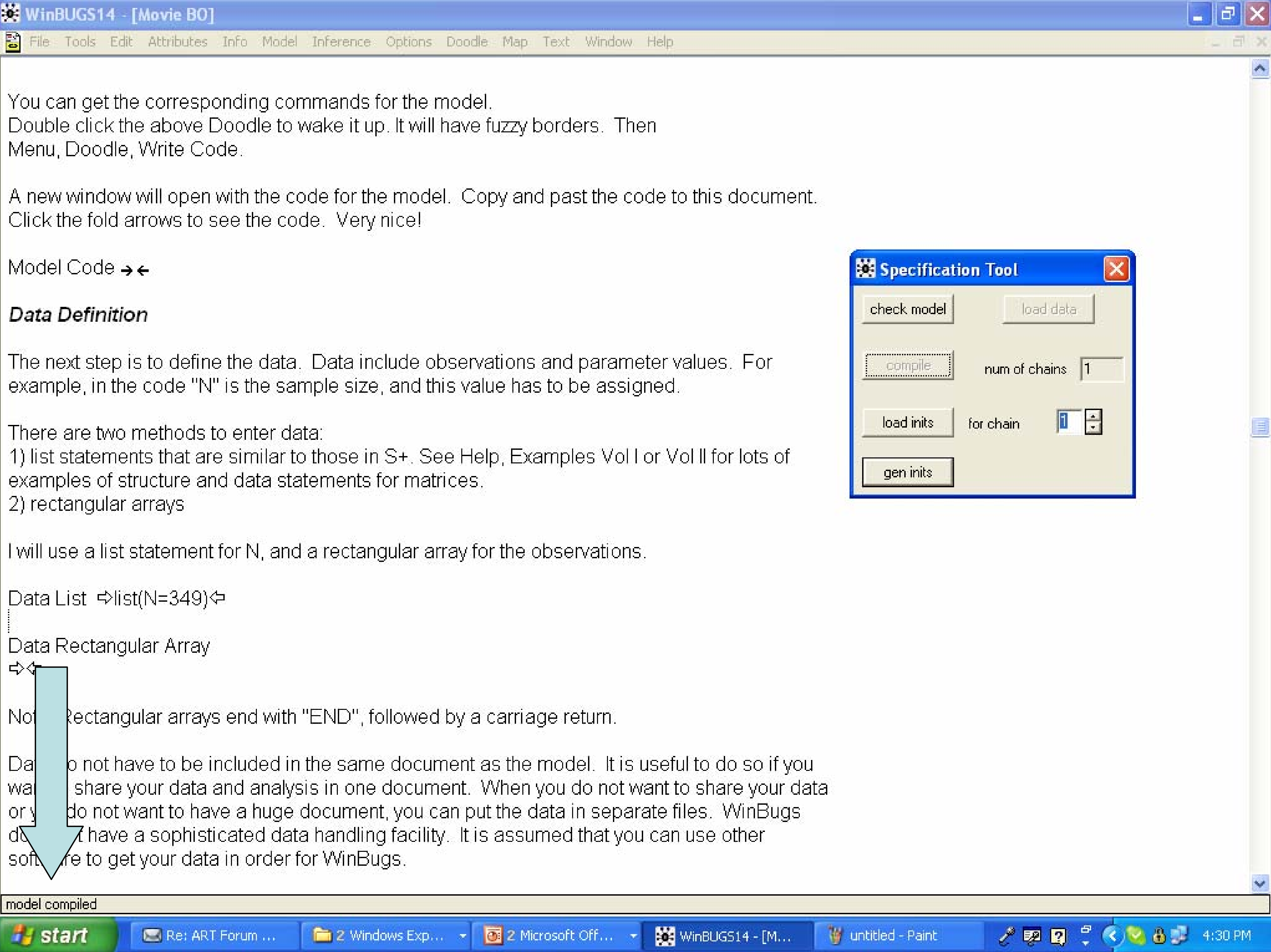
Data Rectangular Array

Note: Rectangular arrays end with "END", followed by a carriage return.

Data do not have to be included in the same document as the model. It is useful to do so if you
want to share your data and analysis in one document. When you do not want to share your data
or you do not want to have a huge document, you can put the data in separate files. WinBugs
does not have a sophisticated data handling facility. It is assumed that you can use other
software to get your data in order for WinBugs.

Step 3: Compile Model

- Specification Tool
- Compile
- Keep your fingers crossed
  - Either a happy message, or
  - Vague message that is very hard to use in debugging



You can get the corresponding commands for the model.
Double click the above Doodle to wake it up. It will have fuzzy borders. Then
Menu, Doodle, Write Code.

A new window will open with the code for the model. Copy and past the code to this document.
Click the fold arrows to see the code. Very nice!

Model Code

Data Definition

The next step is to define the data. Data include observations and parameter values. For example, in the code "N" is the sample size, and this value has to be assigned.

There are two methods to enter data:
1) list statements that are similar to those in S+. See Help, Examples Vol I or Vol II for lots of examples of structure and data statements for matrices.
2) rectangular arrays

I will use a list statement for N, and a rectangular array for the observations.

Data List list(N=349)

Data Rectangular Array

No[illegible]ectangular arrays end with "END", followed by a carriage return.

Da[illegible]o not have to be included in the same document as the model. It is useful to do so if you wa[illegible] share your data and analysis in one document. When you do not want to share your data or y[illegible]do not want to have a huge document, you can put the data in separate files. WinBugs d[illegible]t have a sophisticated data handling facility. It is assumed that you can use other soft[illegible]re to get your data in order for WinBugs.

Step 4: Initial Values

- You can specify the initial values
  - Use same format as data
    - Data list
    - Rectangular file
  - Highlight list or make data window active
  - Click “load init”
- Alternatively, click “gen inits” and WinBugs will generate initial values from your priors

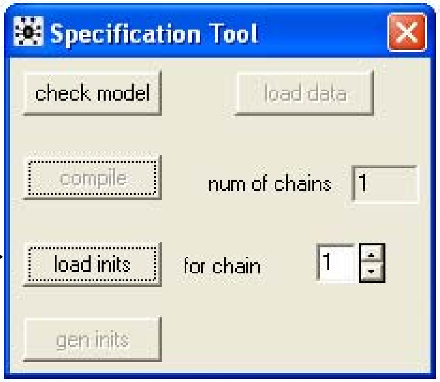
Note: Rectangular arrays end with "END", followed by a carriage return.

Data do not have to be included in the same document as the model. It is useful to do so if you want to share your data and analysis in one document. When you do not want to share your data or you do not want to have a huge document, you can put the data in separate files. WinBugs does not have a sophisticated data handling facility. It is assumed that you can use other software to get your data in order for WinBugs.

Initial Values of Parameters for MCMC

A step before compiling and running the program is to define the starting values for the parameters b0, b1, b2, and y.per. You can define the starting values, or Winbugs can generate them from the prior distributions.

Initial Values for MCMC → list(b0=0, b1=0, b2=0, y.per=1) ←



Running Your Model and Getting Results

Compiling and running your model is a multiple step procedure that is difficult to describe in words.

1) Menu, Model, Specification, Check Model
Either double click the Doodle or highlight the keyword "model" in the code version. Look for a good message at the bottom left corner of this window.

2) Menu, Model, Specification, Load Data
For data, highlight "list". For rectangular arrays within a compound document, highlight the whole array. For a rectangular array in different file, open the file, and make it your active window. Important: you can do more than one Load Data. Always look for "Data Loaded" at the bottom left corner of the screen to verify that it worked.

3) Menu, Model, Specification, Compile
The moment of truth. Look for a happy message at the bottom left hand corner of the screen.

model is initialized

Step 5: Define Sampling Nodes

- Tell WinBugs which parameters to keep track of in MCMC and report on
- Tool bar
  - Inference
  - Samples
- Type parameter names in node box
- The beg and end boxes define iterations to use for estimates
  - You can change these after MCMC is done

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Samples...
Compare...
Correlations
Summary...
Rank...
DIC...

Running Your Model and Gettin

Compiling and running your model ep procedure that is difficult to describe in words.

1) Menu, Model, Specification, Check Model
Either double click the Doodle or highlight the keyword "model" in the code version. Look for a good message at the bottom left corner of this window.

2) Menu, Model, Specification, Load Data
For list data, highlight "list". For rectangular arrays within a compound document, highlight the whole array. For a rectangular array in different file, open the file, and make it your active window. Important: you can do more than one Load Data. Always look for "Data Loaded" at the bottom left corner of the screen to verify that it worked.

3) Menu, Model, Specification, Compile
The moment of truth. Look for a happy message at the bottom left hand corner of the screen.

4) Menu, Model, Specification, Load Inits or Model, Specification, Generate Inits.
If you use Load Inits, then highlight the "list" for the initial values.

5) You can repeat (4) for more than one chain.

6) Next, you have to tell Winbugs how many iterations and what nodes to keep track of.
Menu, Inference, Samples
Type in the name of the node, e.g. b0, and then give the number of iterations.

Do this for each parameter node (b0, b1, b2, and y.per) that you want to estimate.
You may not want to estimate every parameter in the model. In an HB model with 15,000 subjects, you probably do not want to track the individual-level coefficients for all 15,000 subjects.

7) After (6), nothing happens! You need
Menu, Model, Update
to get the thing going

start | Re: ART Forum ... | 2 Windows Exp... | 2 Microsoft Off... | WinBUGS14 - [M... | untitled - Paint | 4:34 PM

WinBUGS14 - [Movie BO]

File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help

Running Your Model and Getting Results

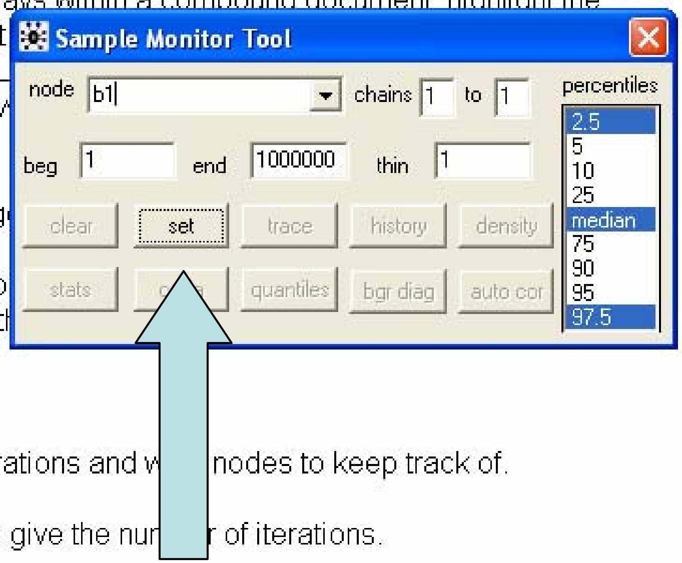
Compiling and running your model is a multiple step procedure that is difficult to describe in words.

1) Menu, Model, Specification, Check Model
Either double click the Doodle or highlight the keyword "model" in the code version. Look for a good message at the bottom left corner of this window.

2) Menu, Model, Specification, Load Data
For list data, highlight "list". For rectangular arrays within a compound document, highlight the
whole array. For a rectangular array in different
window. Important: you can do more than one L
bottom left corner of the screen to verify that it w

3) Menu, Model, Specification, Compile
The moment of truth. Look for a happy messag

4) Menu, Model, Specification, Load Inits or Mo
If you use Load Inits, then highlight the "list" for t



5) You can repeat (4) for more than one chain.

6) Next, you have to tell Winbugs how many iterations and w nodes to keep track of.
Menu, Inference, Samples
Type in the name of the node, e.g. b0, and then give the nur r of iterations.

Do this for each parameter node (b0, b1, b2, and y.per) that you want to estimate.
You may not want to estimate every parameter in the model. In an HB model with 15,000 subjects, you probably do not want to track the individual-level coefficients for all 15,000 subjects.

7) After (6), nothing happens! You need
Menu, Model, Update
to get the thing going

start Re: ART Forum ... 2 Windows Exp... 2 Microsoft Off... WinBUGS14 - [M... untitled - Paint 4:35 PM

Running Your Model and Getting Results

Compiling and running your model is a multiple step procedure that is difficult to describe in words.

1) Menu, Model, Specification, Check Model
Either double click the Doodle or highlight the keyword "model" in the code version. Look for a good message at the bottom left corner of this window.

2) Menu, Model, Specification, Load Data
For list data, highlight "list". For rectangular arrays within a compound document, highlight the whole array. For a rectangular array in different
window. Important: you can do more than one L
bottom left corner of the screen to verify that it w

3) Menu, Model, Specification, Compile
The moment of truth. Look for a happy messag

4) Menu, Model, Specification, Load Inits or Mo
If you use Load Inits, then highlight the "list" for th

5) You can repeat (4) for more than one chain.

6) Next, you have to tell Winbugs how many iterations and what nodes to keep track of.
Menu, Inference, Samples
Type in the name of the node, e.g. b0, and then give the number of iterations.

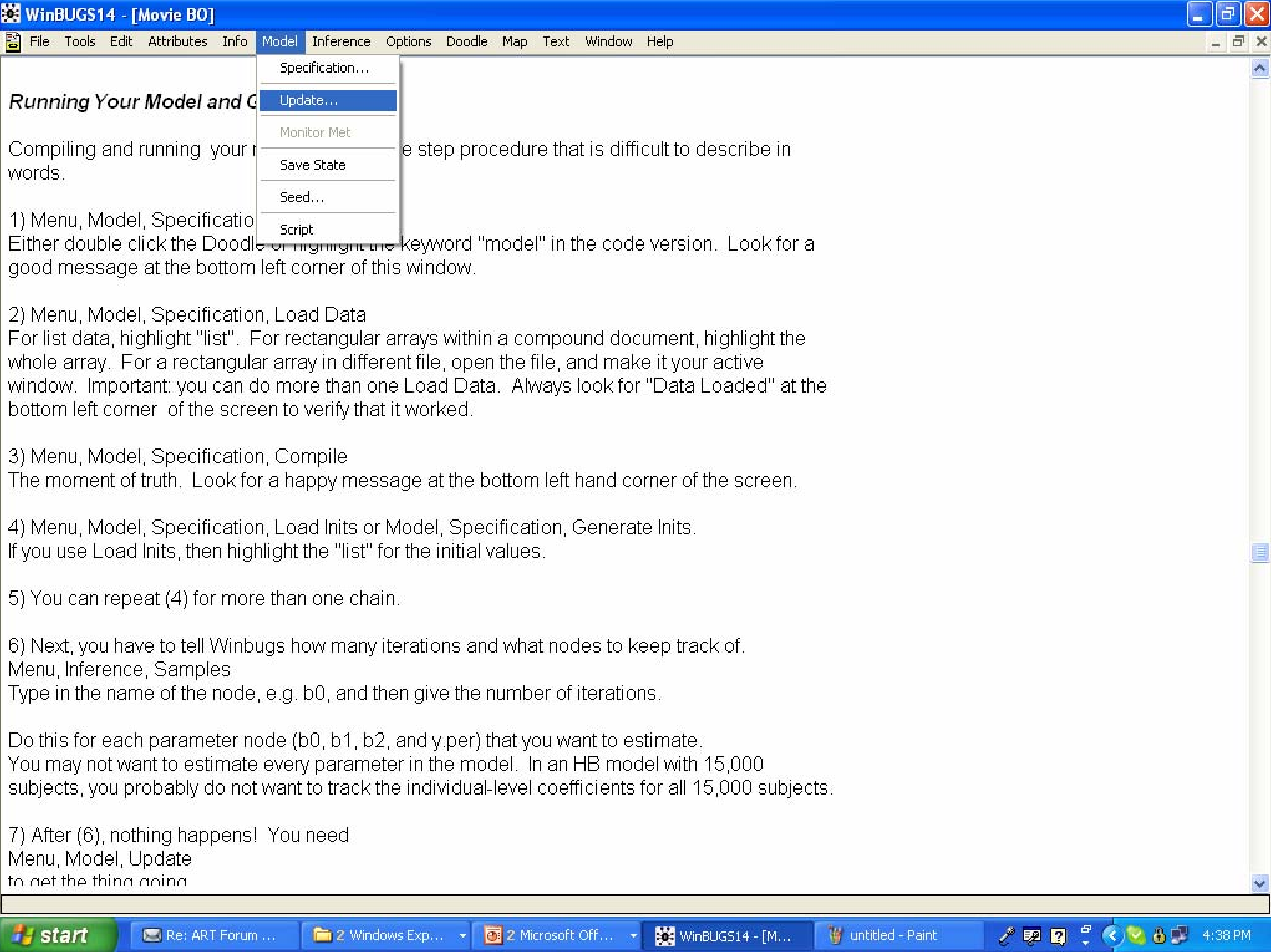
Do this for each parameter node (b0, b1, b2, and y.per) that you want to estimate.
You may not want to estimate every parameter in the model. In an HB model with 15,000 subjects, you probably do not want to track the individual-level coefficients for all 15,000 subjects.

7) After (6), nothing happens! You need
Menu, Model, Update
to get the thing going

Do this for all parameters

Step 6: Run MCMC

- Tool bar
  - Model
  - Update
- Enter number of MCMC iterations in “update” box
- Enter number of iterations used before refreshing the MCMC trace plots
- Enter * in node box of Sample Monitor Tool (see Step 4) to monitor all parameters
- Hit update in Update Tool until you want to stop



Running Your Model and C

Compiling and running your r e step procedure that is difficult to describe in words.

1) Menu, Model, Specificatio
Either double click the Doodle keyword "model" in the code version. Look for a good message at the bottom left corner of this window.

2) Menu, Model, Specification, Load Data
For list data, highlight "list". For rectangular arrays within a compound document, highlight the whole array. For a rectangular array in different file, open the file, and make it your active window. Important: you can do more than one Load Data. Always look for "Data Loaded" at the bottom left corner of the screen to verify that it worked.

3) Menu, Model, Specification, Compile
The moment of truth. Look for a happy message at the bottom left hand corner of the screen.

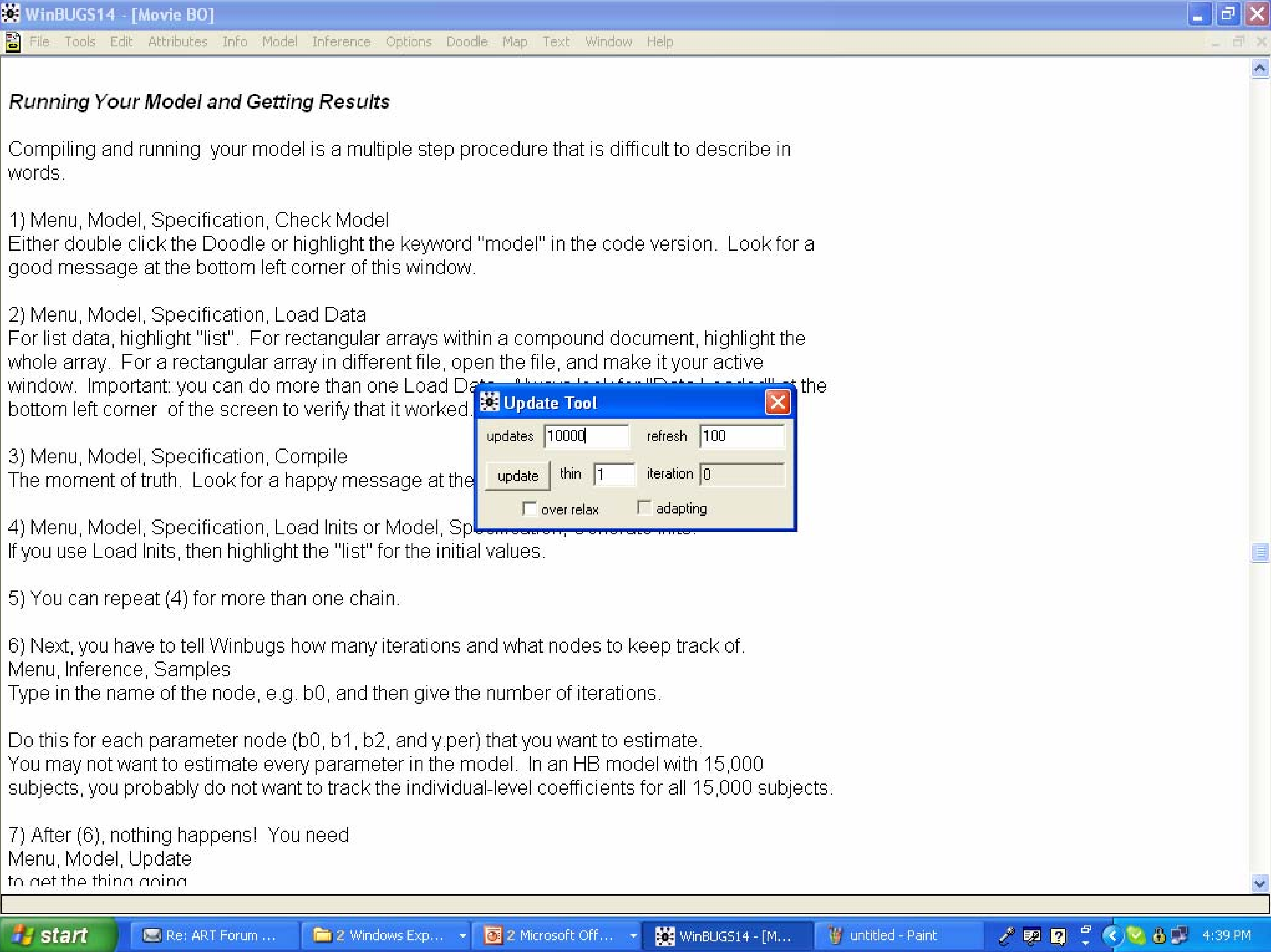
4) Menu, Model, Specification, Load Inits or Model, Specification, Generate Inits.
If you use Load Inits, then highlight the "list" for the initial values.

5) You can repeat (4) for more than one chain.

6) Next, you have to tell Winbugs how many iterations and what nodes to keep track of.
Menu, Inference, Samples
Type in the name of the node, e.g. b0, and then give the number of iterations.

Do this for each parameter node (b0, b1, b2, and y.per) that you want to estimate.
You may not want to estimate every parameter in the model. In an HB model with 15,000 subjects, you probably do not want to track the individual-level coefficients for all 15,000 subjects.

7) After (6), nothing happens! You need
Menu, Model, Update
to get the thing going



Running Your Model and Getting Results

Compiling and running your model is a multiple step procedure that is difficult to describe in words.

1) Menu, Model, Specification, Check Model
Either double click the Doodle or highlight the keyword "model" in the code version. Look for a good message at the bottom left corner of this window.

2) Menu, Model, Specification, Load Data
For list data, highlight "list". For rectangular arrays within a compound document, highlight the whole array. For a rectangular array in different file, open the file, and make it your active window. Important: you can do more than one Load Da ... the
bottom left corner of the screen to verify that it worked.

3) Menu, Model, Specification, Compile
The moment of truth. Look for a happy message at the

4) Menu, Model, Specification, Load Inits or Model, Sp ...
If you use Load Inits, then highlight the "list" for the initial values.

5) You can repeat (4) for more than one chain.

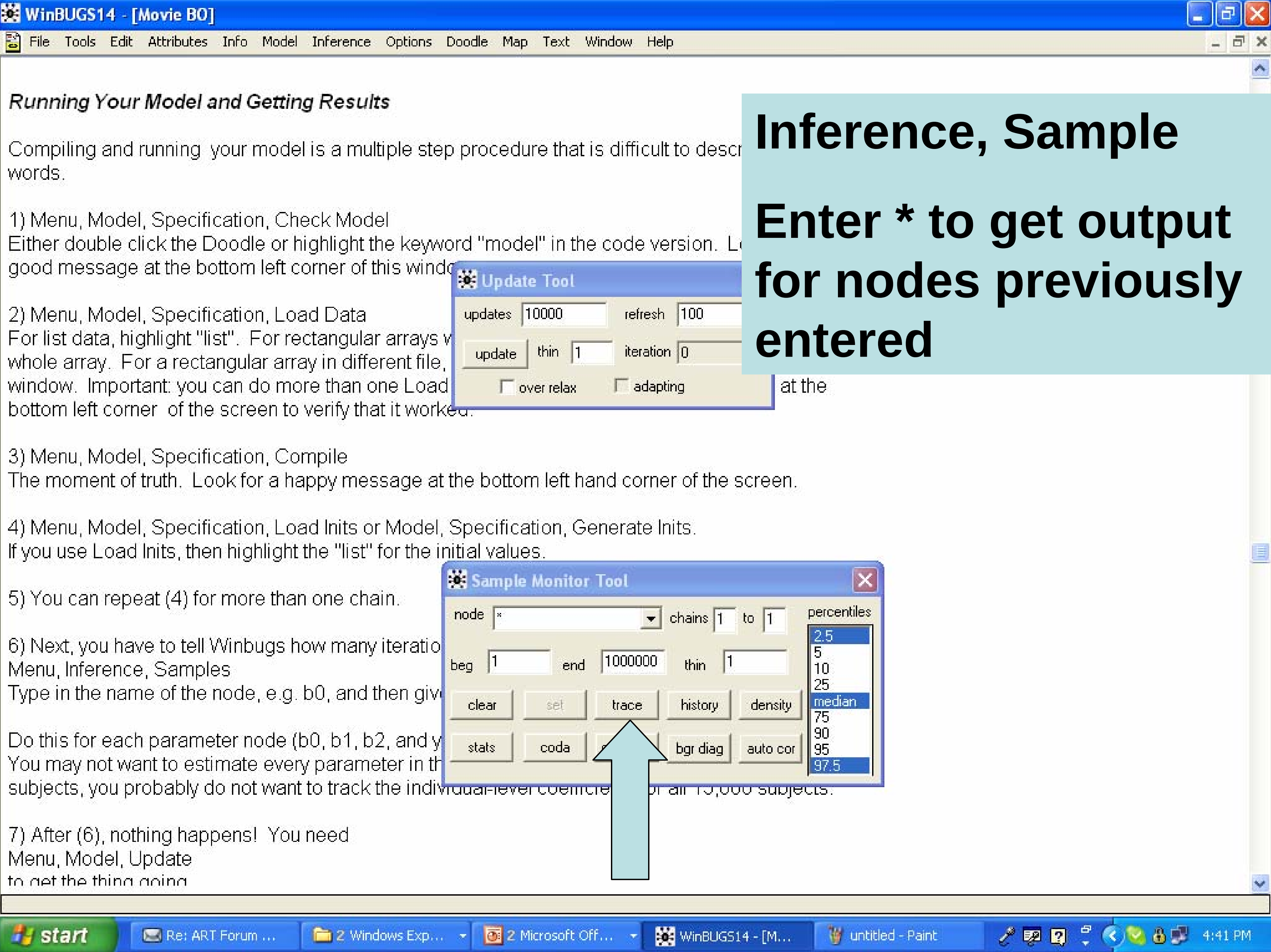
6) Next, you have to tell Winbugs how many iterations and what nodes to keep track of.
Menu, Inference, Samples
Type in the name of the node, e.g. b0, and then give the number of iterations.

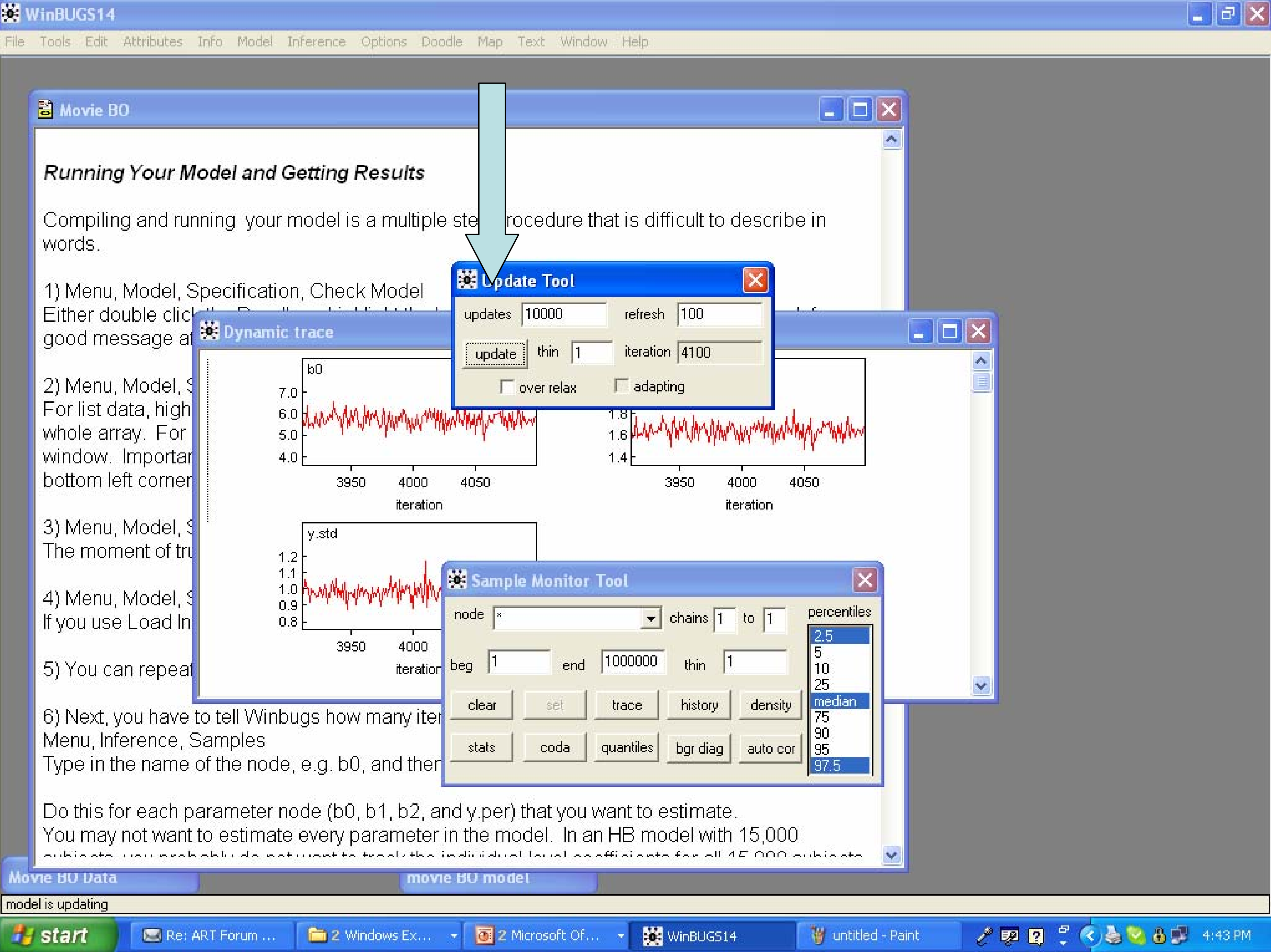
Do this for each parameter node (b0, b1, b2, and y.per) that you want to estimate.
You may not want to estimate every parameter in the model. In an HB model with 15,000 subjects, you probably do not want to track the individual-level coefficients for all 15,000 subjects.

7) After (6), nothing happens! You need
Menu, Model, Update
to get the thing going

Inference, Sample

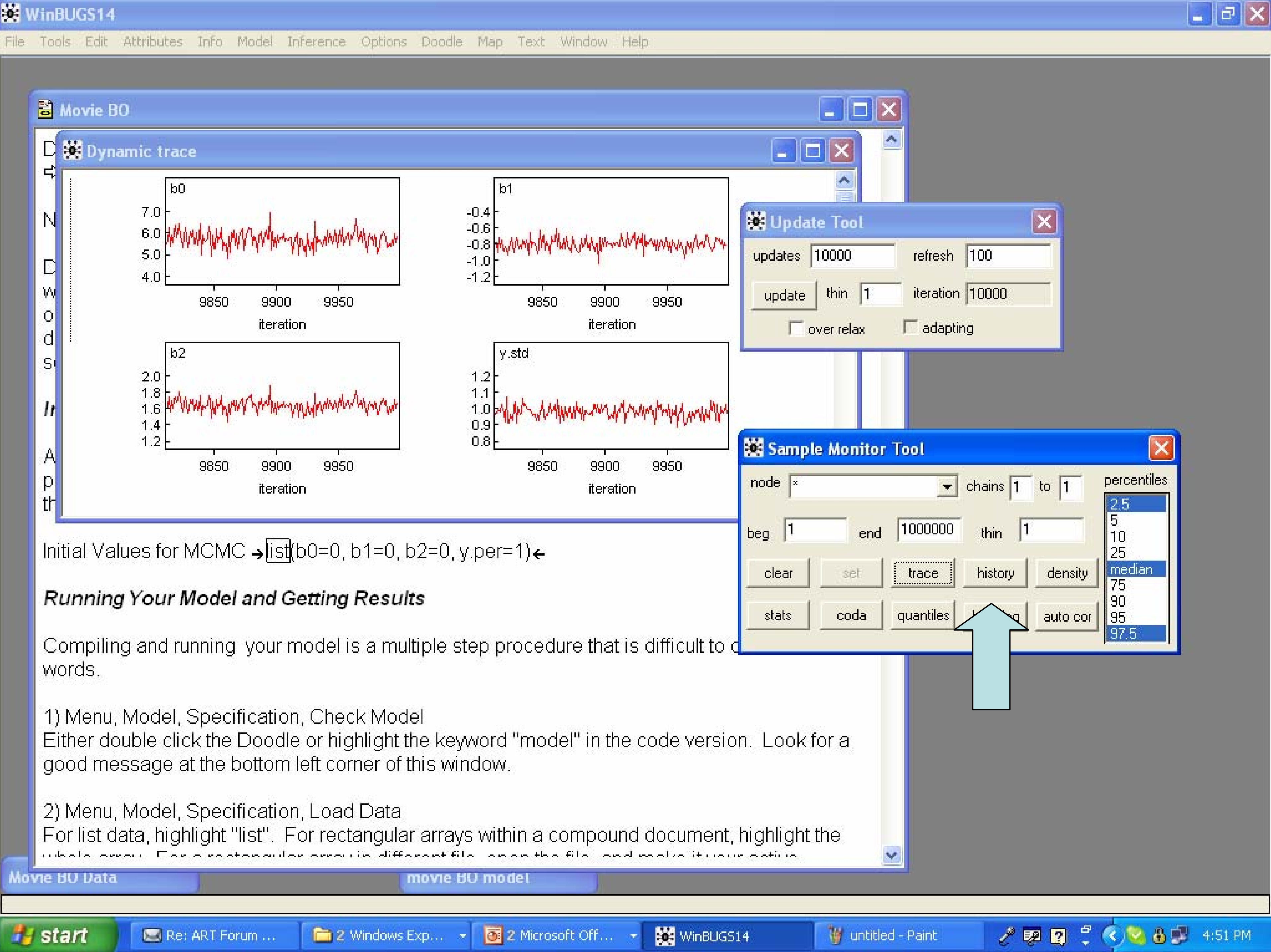
Enter * to get output for nodes previously entered



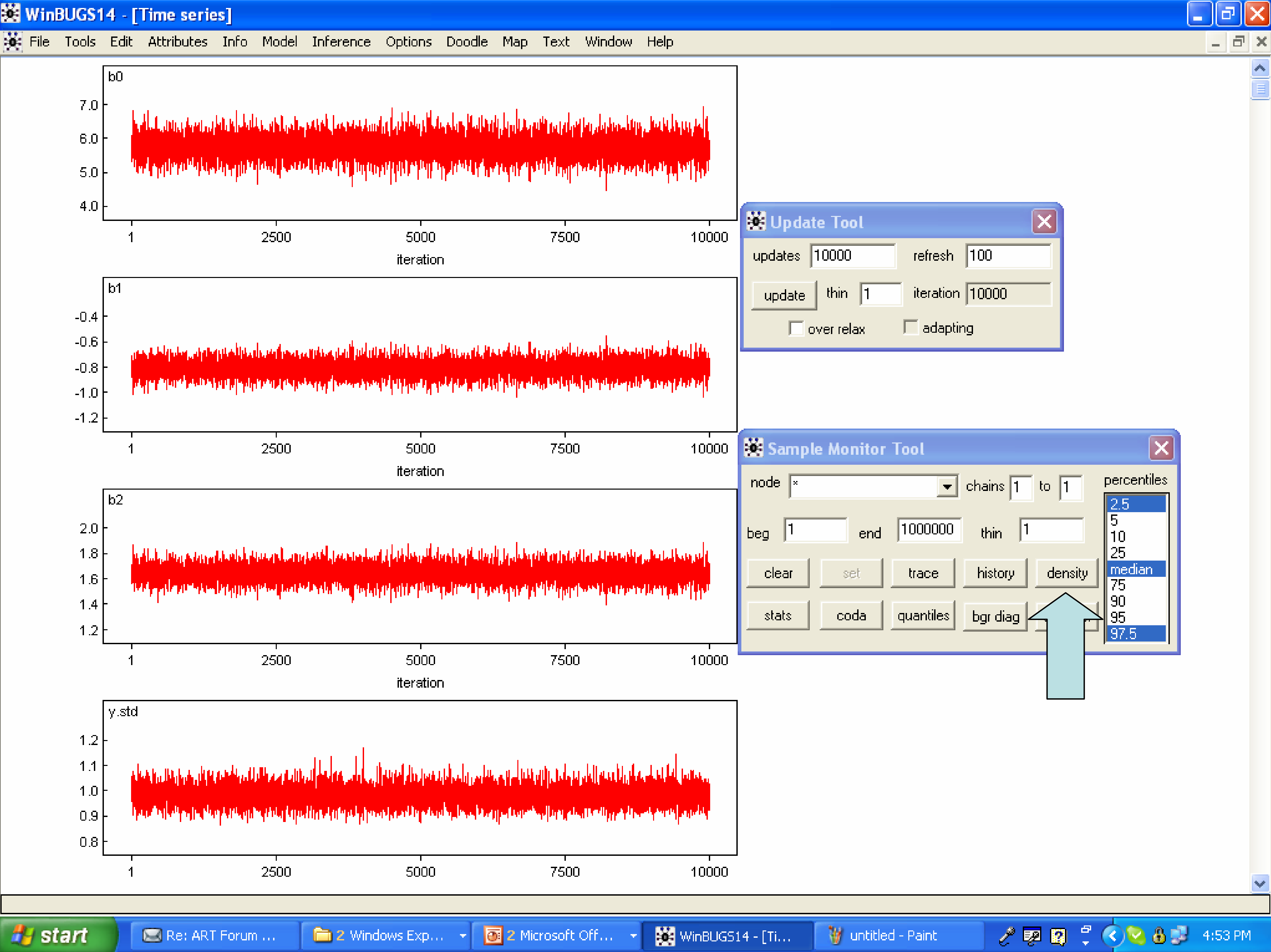

WinBUGS14
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
Movie BO
Running Your Model and Getting Results
Compiling and running your model is a multiple ste rocedure that is difficult to describe in words.
1) Menu, Model, Specification, Check Model
Either double clic
good message at
2) Menu, Model, S
For list data, high
whole array. For
window. Importan
bottom left corner
3) Menu, Model, S
The moment of tru
4) Menu, Model, S
If you use Load In
5) You can repeat
6) Next, you have to tell Winbugs how many iter
Menu, Inference, Samples
Type in the name of the node, e.g. b0, and then
Do this for each parameter node (b0, b1, b2, and y.per) that you want to estimate.
You may not want to estimate every parameter in the model. In an HB model with 15,000
Update Tool
updates 10000
refresh 100
update
thin 1
iteration 4100
over relax
adapting
Dynamic trace
b0
7.0
6.0
5.0
4.0
3950
4000
4050
iteration
1.8
1.6
1.4
3950
4000
4050
iteration
y.std
1.2
1.1
1.0
0.9
0.8
3950
4000
iteration
Sample Monitor Tool
node *
chains 1 to 1
percentiles
beg 1
end 1000000
thin 1
clear
set
trace
history
density
stats
coda
quantiles
bgr diag
auto cor
2.5
5
10
25
median
75
90
95
97.5
Movie BO Data
movie BO model
model is updating
start
Re: ART Forum ...
2 Windows Ex...
2 Microsoft Of...
WinBUGS14
untitled - Paint
4:43 PM

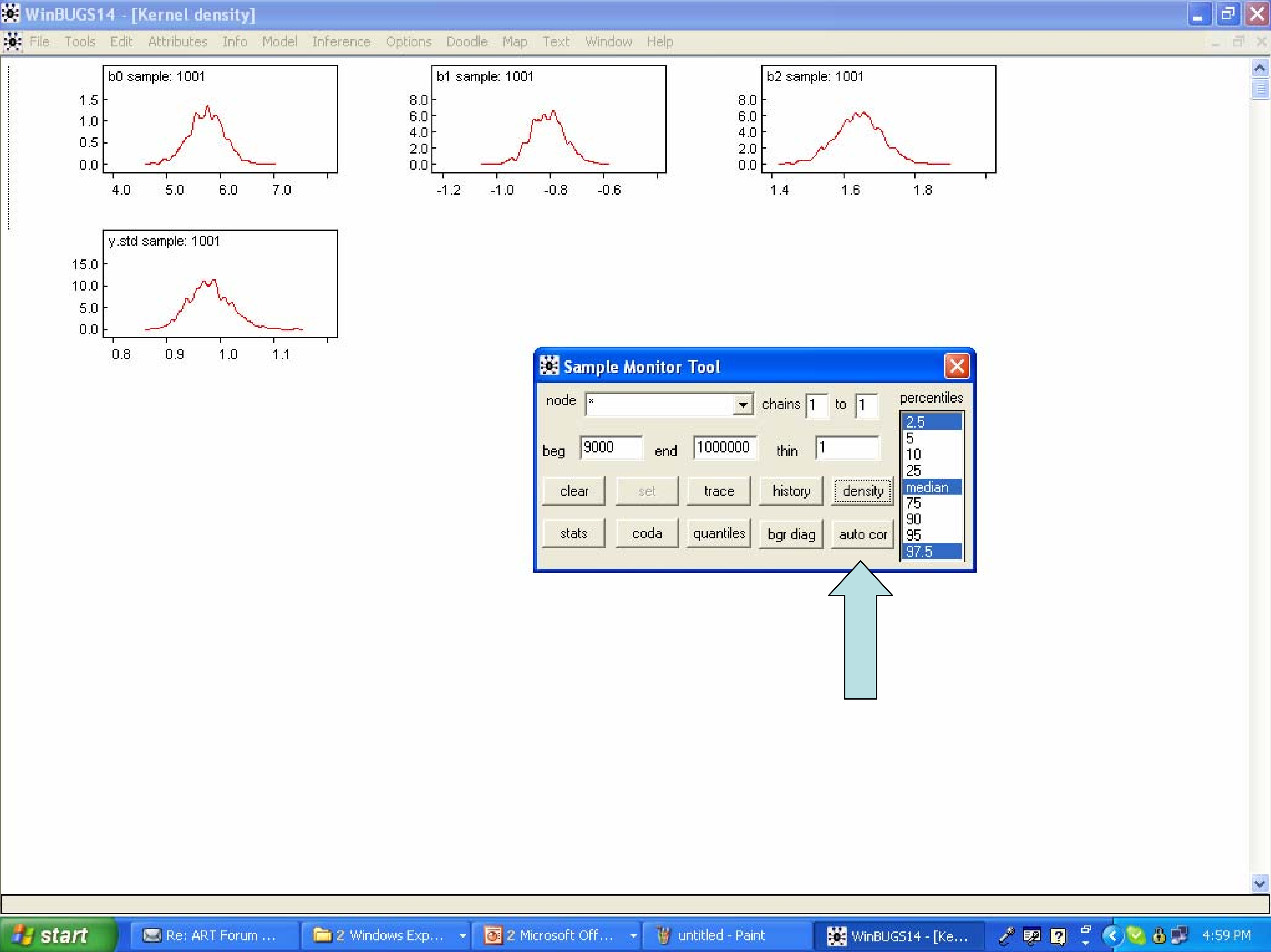
Step 7: WinBugs Output

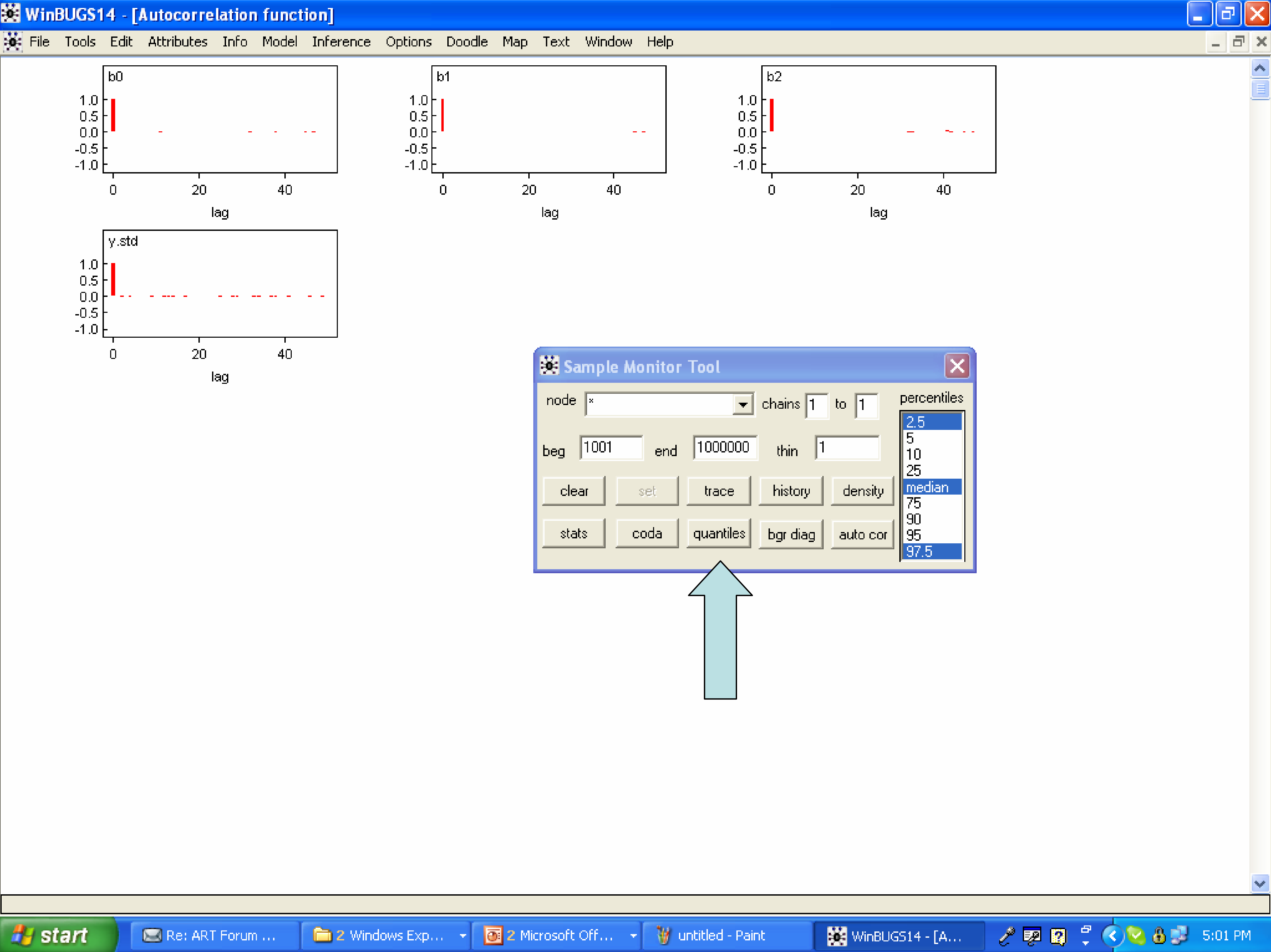
- After finishing runs
  - Modify beg and end values in Sample Monitor Tool to say which iterations you will use
- Output
  - History shows MCMC iterations from beg to end
  - Density plots histograms of iterations
  - Stats give estimates
  - Auto corr gives autocorrelation plots
  - Quartiles give running quartiles
  - Coda gives values of iterations

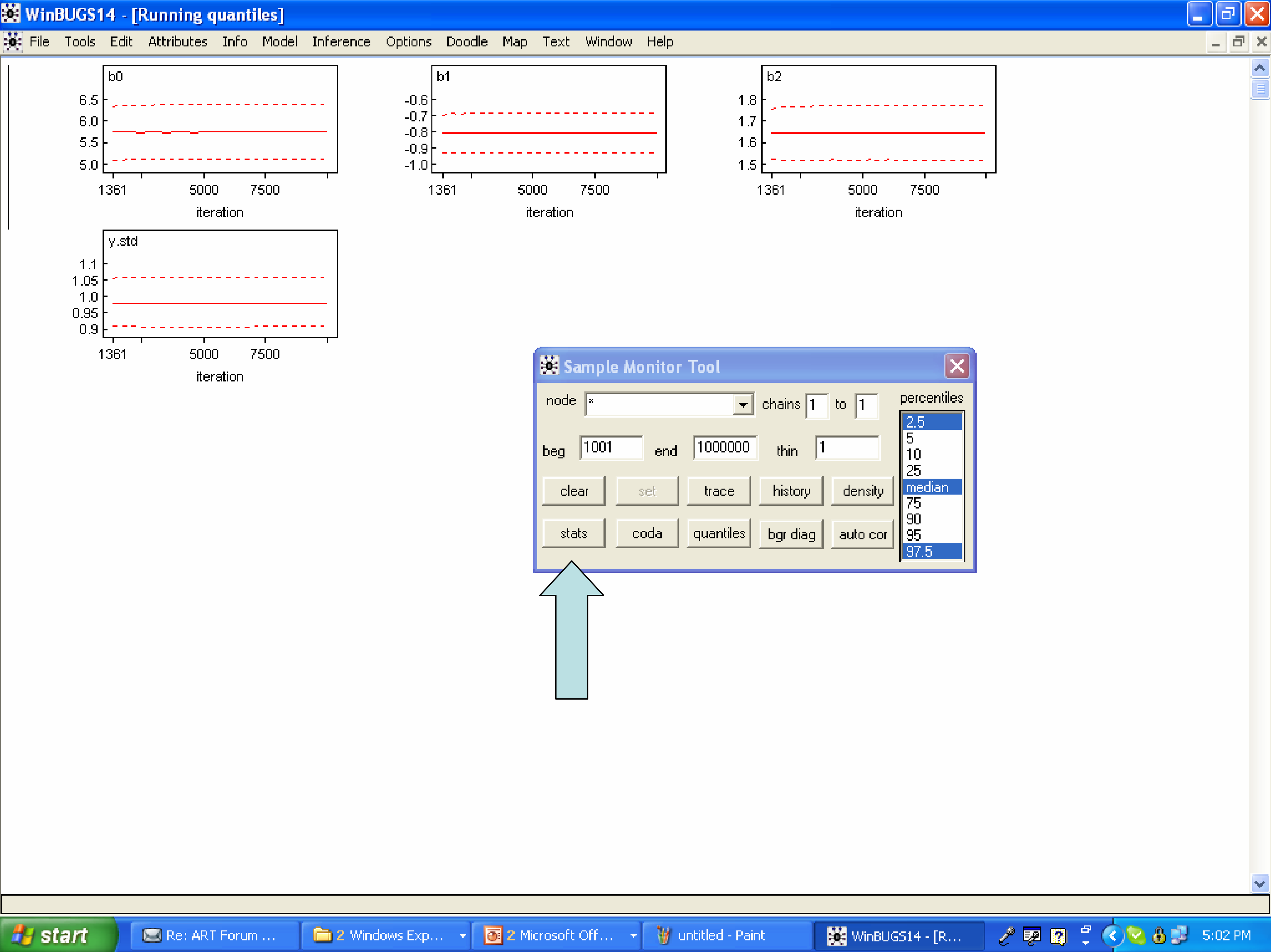


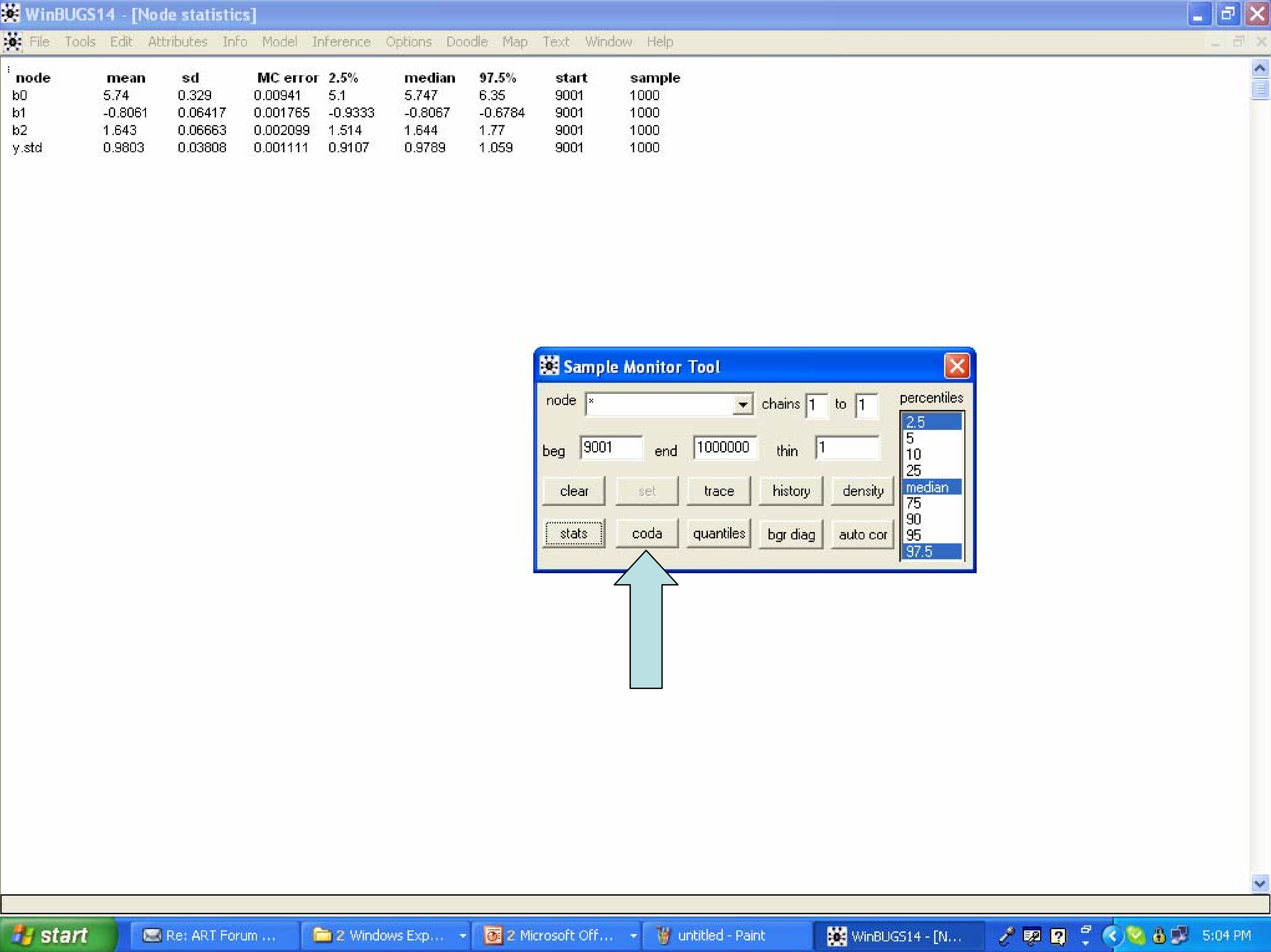
WinBUGS14
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
Movie BO
Dynamic trace
b0
7.0 6.0 5.0 4.0
9850 9900 9950
iteration
b1
-0.4 -0.6 -0.8 -1.0 -1.2
9850 9900 9950
iteration
b2
2.0 1.8 1.6 1.4 1.2
9850 9900 9950
iteration
y.std
1.2 1.1 1.0 0.9 0.8
9850 9900 9950
iteration
Update Tool
updates 10000
refresh 100
update
thin 1
iteration 10000
over relax
adapting
Sample Monitor Tool
node *
chains 1 to 1
percentiles
2.5 5 10 25 median 75 90 95 97.5
beg 1
end 1000000
thin 1
clear
set
trace
history
density
stats
coda
quantiles
auto cor
Initial Values for MCMC list(b0=0, b1=0, b2=0, y.per=1)
Running Your Model and Getting Results
Compiling and running your model is a multiple step procedure that is difficult to c
words.
1) Menu, Model, Specification, Check Model
Either double click the Doodle or highlight the keyword "model" in the code version. Look for a good message at the bottom left corner of this window.
2) Menu, Model, Specification, Load Data
For list data, highlight "list". For rectangular arrays within a compound document, highlight the
Movie BO Data
movie BO model
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
WinBUGS14
untitled - Paint
4:51 PM

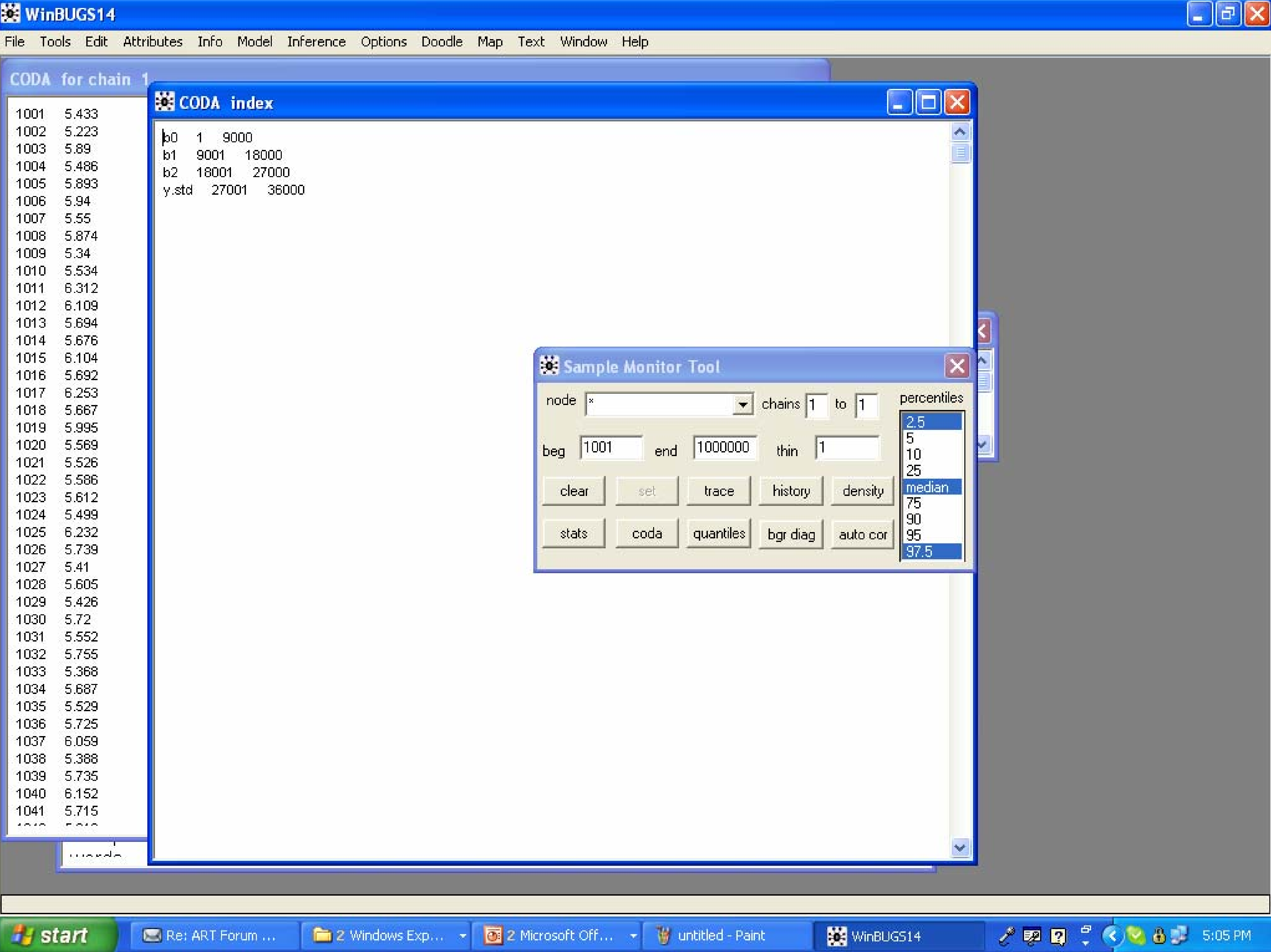

WinBUGS14 - [Time series]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
b0
7.0
6.0
5.0
4.0
1 2500 5000 7500 10000
iteration
b1
-0.4
-0.6
-0.8
-1.0
-1.2
1 2500 5000 7500 10000
iteration
b2
2.0
1.8
1.6
1.4
1.2
1 2500 5000 7500 10000
iteration
y.std
1.2
1.1
1.0
0.9
0.8
1 2500 5000 7500 10000
Update Tool
updates 10000 refresh 100
update thin 1 iteration 10000
over relax adapting
Sample Monitor Tool
node *
chains 1 to 1
percentiles
beg 1 end 1000000 thin 1
clear set trace history density
stats coda quantiles bgr diag
2.5
5
10
25
median
75
90
95
97.5
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
WinBUGS14 - [Ti...
untitled - Paint
4:53 PM


WinBUGS14 - [Kernel density]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
b0 sample: 1001
1.5
1.0
0.5
0.0
4.0 5.0 6.0 7.0
b1 sample: 1001
8.0
6.0
4.0
2.0
0.0
-1.2 -1.0 -0.8 -0.6
b2 sample: 1001
8.0
6.0
4.0
2.0
0.0
1.4 1.6 1.8
y.std sample: 1001
15.0
10.0
5.0
0.0
0.8 0.9 1.0 1.1
Sample Monitor Tool
node *
chains 1 to 1
percentiles
beg 9000
end 1000000
thin 1
clear set trace history density
stats coda quantiles bgr diag auto cor
2.5
5
10
25
median
75
90
95
97.5
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
untitled - Paint
WinBUGS14 - [Ke...
4:59 PM


WinBUGS14 - [Autocorrelation function]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
b0
b1
b2
y.std
1.0
0.5
0.0
-0.5
-1.0
0
20
40
lag
Sample Monitor Tool
node
*
chains
1
to
1
percentiles
beg
1001
end
1000000
thin
1
clear
set
trace
history
density
stats
coda
quantiles
bgr diag
auto cor
2.5
5
10
25
median
75
90
95
97.5
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
untitled - Paint
WinBUGS14 - [A...
5:01 PM


WinBUGS14 - [Running quantiles]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
b0
6.5
6.0
5.5
5.0
1361 5000 7500
iteration
b1
-0.6
-0.7
-0.8
-0.9
-1.0
1361 5000 7500
iteration
b2
1.8
1.7
1.6
1.5
1361 5000 7500
iteration
y.std
1.1
1.05
1.0
0.95
0.9
1361 5000 7500
iteration
Sample Monitor Tool
node *
chains 1 to 1
percentiles
2.5
5
10
25
median
75
90
95
97.5
beg 1001
end 1000000
thin 1
clear
set
trace
history
density
stats
coda
quantiles
bgr diag
auto cor
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
untitled - Paint
WinBUGS14 - [R...
5:02 PM


WinBUGS14 - [Node statistics]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
node mean sd MC error 2.5% median 97.5% start sample
b0 5.74 0.329 0.00941 5.1 5.747 6.35 9001 1000
b1 -0.8061 0.06417 0.001765 -0.9333 -0.8067 -0.6784 9001 1000
b2 1.643 0.06663 0.002099 1.514 1.644 1.77 9001 1000
y.std 0.9803 0.03808 0.001111 0.9107 0.9789 1.059 9001 1000
Sample Monitor Tool
node *
chains 1 to 1
percentiles
beg 9001
end 1000000
thin 1
clear
set
trace
history
density
stats
coda
quantiles
bgr diag
auto cor
2.5
5
10
25
median
75
90
95
97.5
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
untitled - Paint
WinBUGS14 - [N...
5:04 PM


WinBUGS14
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
CODA for chain 1
1001 5.433
1002 5.223
1003 5.89
1004 5.486
1005 5.893
1006 5.94
1007 5.55
1008 5.874
1009 5.34
1010 5.534
1011 6.312
1012 6.109
1013 5.694
1014 5.676
1015 6.104
1016 5.692
1017 6.253
1018 5.667
1019 5.995
1020 5.569
1021 5.526
1022 5.586
1023 5.612
1024 5.499
1025 6.232
1026 5.739
1027 5.41
1028 5.605
1029 5.426
1030 5.72
1031 5.552
1032 5.755
1033 5.368
1034 5.687
1035 5.529
1036 5.725
1037 6.059
1038 5.388
1039 5.735
1040 6.152
1041 5.715
CODA index
b0 1 9000
b1 9001 18000
b2 18001 27000
y.std 27001 36000
Sample Monitor Tool
node *
chains 1 to 1
percentiles
beg 1001
end 1000000
thin 1
clear
set
trace
history
density
stats
coda
quantiles
bgr diag
auto cor
2.5
5
10
25
median
75
90
95
97.5
start
Re: ART Forum ...
2 Windows Exp...
2 Microsoft Off...
untitled - Paint
WinBUGS14
5:05 PM

Scripting

- After you get your program to run successfully, you will become very bored with this lengthy, 7-step procedure
- WinBugs has a scripting language so you can do these activities in batch model
  - http://www.aims.ac.za/~mackay/BUGS/Manuals/Scripts.html
  - http://web.maths.unsw.edu.au/~scott/price.html

Scripting Language

- Need 4 files (.odc or .txt) with
  - Script commands
  - Model commands
  - Data
  - Initial values
- Open script command in WinBugs
- Execute from Tool Bar
  - Model
  - Script

Model File

moviebo.model.odc

```
model;
{
   for( i in 1 : N ) {
      TotalBO.ln[i]~ dnorm(mu[i],y.per)
   }
   for( i in 1 : N ) {
      mu[i] <- b0 + b1 * Screens.ln[i] + b2 * OpenBO.ln[i
   }
   b0 ~ dnorm( 0.0,1.0E-6)
   b1 ~ dnorm( 0.0,1.0E-6)
   b2 ~ dnorm( 0.0,1.0E-6)
   y.per ~ dgamma(0.001,0.001)
   y.std <- 1 / sqrt(y.per)
}
```

Data File

moviebo.data.txt

```
list( N = 349,
TotalBO.ln= c(
6.078719644,
5.922596668,

-4.605170186,
-4.605170186),
OpenBO.ln= c(
4.682501529,
4.752037262,

-4.605170186,
-4.605170186),
Screens.ln = c(
8.333991247,
8.331345425,

2.197224577,
3.091042453) )
```

Initialize Parameters
moviebo.initial.txt

```
list(b0=1.5, b1=0, b2=0, y.per=0.13)
```

Script File
movie.script.odc

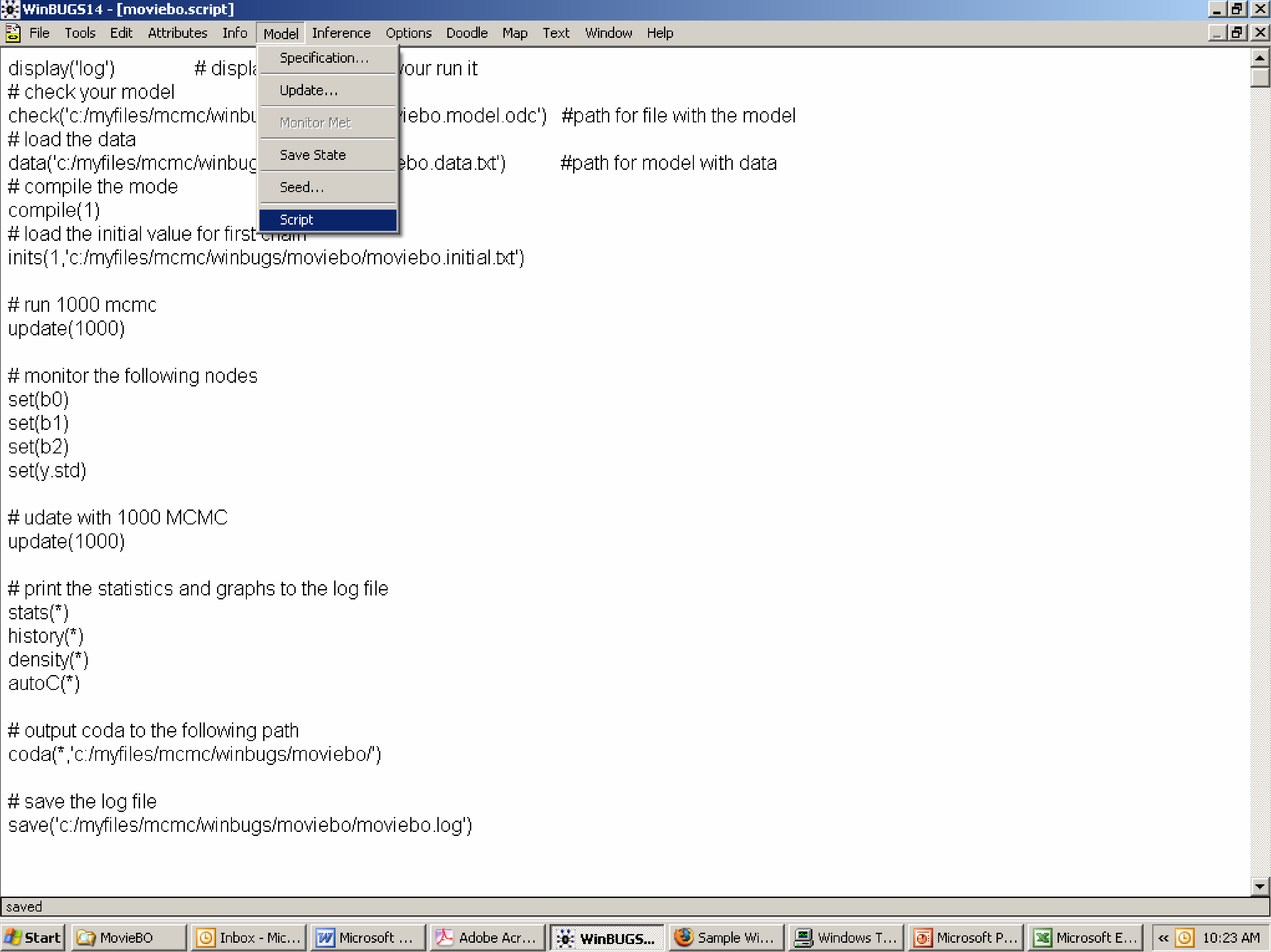
```
display('log')          # display the log file as your run it
# check your model
check('c:/myfiles/mcmc/winbugs/moviebo/moviebo.model.odc')
# load the data
data('c:/myfiles/mcmc/winbugs/moviebo/moviebo.data.txt')
# compile the mode
compile(1)
# load the initial value for first chain
inits(1,'c:/myfiles/mcmc/winbugs/moviebo/moviebo.initial.txt')
```

Script File Continued 1

```
# run 1000 mcmc
update(1000)

# monitor the following nodes
set(b0)
set(b1)
set(b2)
set(y.std)

# udate with 1000 MCMC
update(1000)
```

Script File Continued 2

```
# print the statistics and graphs to the log file
stats(*)
history(*)
density(*)
autoC(*)

# output coda to the following path
coda(*,'c:/myfiles/mcmc/winbugs/moviebo/')

# save the log file
save('c:/myfiles/mcmc/winbugs/moviebo/moviebo.log
```


WinBUGS14 - [moviebo.script]
File Tools Edit Attributes Info Model Inference Options Doodle Map Text Window Help
Specification...
Update...
Monitor Met
Save State
Seed...
Script
display('log') # displ
your run it
check your model
check('c:/myfiles/mcmc/winbu
iebo.model.odc') #path for file with the model
load the data
data('c:/myfiles/mcmc/winbug
ebo.data.txt') #path for model with data
# compile the mode
compile(1)
# load the initial value for first
inits(1,'c:/myfiles/mcmc/winbugs/moviebo/moviebo.initial.txt')
# run 1000 mcmc
update(1000)
# monitor the following nodes
set(b0)
set(b1)
set(b2)
set(y.std)
# udate with 1000 MCMC
update(1000)
# print the statistics and graphs to the log file
stats(*)
history(*)
density(*)
autoC(*)
# output coda to the following path
coda(*,'c:/myfiles/mcmc/winbugs/moviebo/')
# save the log file
save('c:/myfiles/mcmc/winbugs/moviebo/moviebo.log')
saved
Start
MovieBO
Inbox - Mic...
Microsoft ...
Adobe Acr...
WinBUGS...
Sample Wi...
Windows T...
Microsoft P...
Microsoft E...
10:23 AM

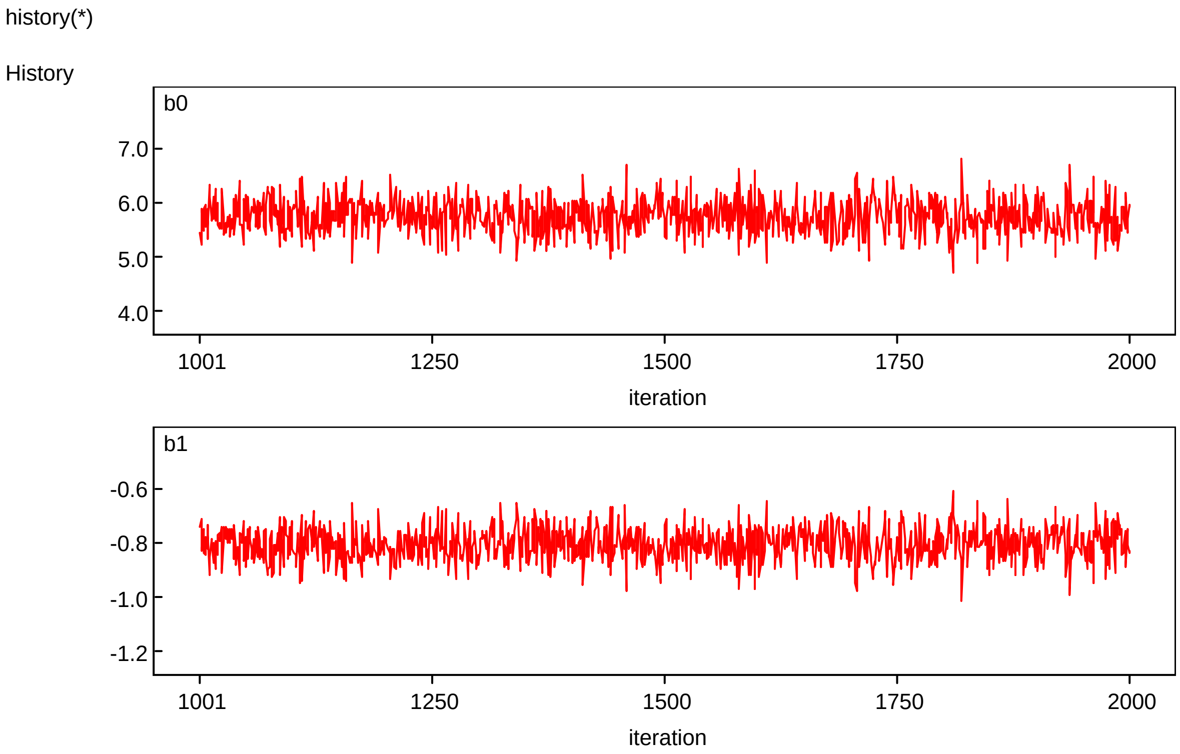
Log File: Statistics

stats(*)

Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b0	5.739	0.3245	0.009024	5.119	5.737	6.372	1001	1000
b1	-0.8058	0.06266	0.001805	-0.9284	-0.8065	-0.685	1001	1000
b2	1.643	0.06433	0.001759	1.518	1.643	1.769	1001	1000
y.std	0.9789	0.03768	0.001245	0.9084	0.9777	1.059	1001	1000

Log File: History



Summary

- Bayesian methods hold great promise for internet and e-commerce applications
- Particularly appropriate when there are many sampling units and sparse observations per unit
- Unifies statistics and decision models
- Tracks sources of uncertainty