The University of Newcastle

Kerrie Mengersen

Introduction to
Bayesian Modelling - 2

Armidale 2004

INTRODUCTION TO BAYESIAN COMPUTATION

- Markov chain Monte Carlo
- Introduction to BUGS
- Convergence diagnostics

Markov chain Monte Carlo

- "Decompose" joint posterior distribution into a sequence of conditional distributions – these are often much simpler (eg, simple univariate normals, etc)
- Simulate from each conditional distribution in turn. We use a simulation method that resembles a Markov chain (so that the new simulated value relies only on the previous value), giving a set of simulated values

  $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(i)}, \ldots$

  which converges to the required conditional,
  The resulting simulations will come from the required joint distribution
- We can use Markov chain theory to make statements about behaviour and convergence of the chain

MCMC Algorithms

- **Gibbs sampling**: sample from full conditionals
- **Metropolis-Hastings**: sample from an "easy" distribution and accept only some of the values
- Lots of variations: reversible jump, slice sampling, particle filters, perfect sampling, adaptive rejection sampling, etc
- Need to ensure conditions, eg *detailed balance, reversibility*

Gibbs sampling

Joint posterior $p(\theta_1, \theta_2, \ldots, \theta_k \mid y)$

1. Choose starting values $\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)}$

At *i*th iteration $^{(i+1)}$

2. Sample $\theta_1^{(i+1)}$ from $p(\theta_1^{(i)} \mid \theta_2^{(i)}, \theta_3^{(i)}, \ldots, \theta_k^{(i)}, y)$

   Sample $\theta_2^{(i+1)}$ from $p(\theta_2^{(i)} \mid \theta_1^{(i+1)}, \theta_3^{(i)}, \ldots, \theta_k^{(i)}, y)$

   …

   Sample $\theta_k^{(i+1)}$ from $p(\theta_k^{(i)} \mid \theta_1^{(i+1)}, \theta_2^{(i+1)}, \ldots, \theta_{k-1}^{(i+1)}, y)$

3. Repeat step 2 many times

Estimation using MCMC

Have simulations:

$$
\begin{array}{llll}
\theta_1^{(0)} & \theta_2^{(0)} & \dots & \theta_k^{(0)} \\
\theta_1^{(1)} & \theta_2^{(1)} & \dots & \theta_k^{(1)} \\
\theta_1^{(2)} & \theta_2^{(2)} & \dots & \theta_k^{(2)} \\
\dots & & & \\
\theta_1^{(l)} & \theta_2^{(l)} & \dots & \theta_k^{(l)}
\end{array}
$$

Easy to estimate expected values:

$$E_{\underline{\theta}|Y}\left(\frac{\theta_1}{\theta_2}\right) \approx \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\theta_1^{(i)}}{\theta_2^{(i)}}\right)$$

Easy to estimate quantiles (credible intervals)

Easy to estimate densities.

Metropolis sampling

- Often we can't simulate from conditional dist'n.
- Instead, simulate from "easy" (proposal) distribution and accept only some of the values.
  - **Conditional distribution $p(\theta|...)$**
  - **Proposal distribution $q(\theta)$**
  - **Suppose we have $\theta^{(i-1)}$ and we want $\theta^{(i)}$**
  - **Simulate possible $\theta^{(i)}$ (θ^* say) from $q(\theta)$ centred on $\theta^{(i)}$**
  - **Accept θ^* with probability:**

$$\alpha = \min \{ 1, \; p(\theta^* |...) / p(\theta^{(i-1)} |...) \}$$

  - **If θ^* is accepted, $\theta^{(i)} = \theta^*$; otherwise $\theta^{(i)} = \theta^{(i-1)}$**

Hastings sampler

- If the proposal $q(\theta)$ is *not* symmetric, the acceptance probability becomes:
  - **Accept θ^* with probability:**

$$\alpha = \min(1, \frac{q(\theta^{(i-1)})}{q(\theta^*)} \frac{p(\theta^* \mid ...)}{p(\theta^{(i-1)} \mid ...)})$$

Graphical Representation
(conditional independence graphs)

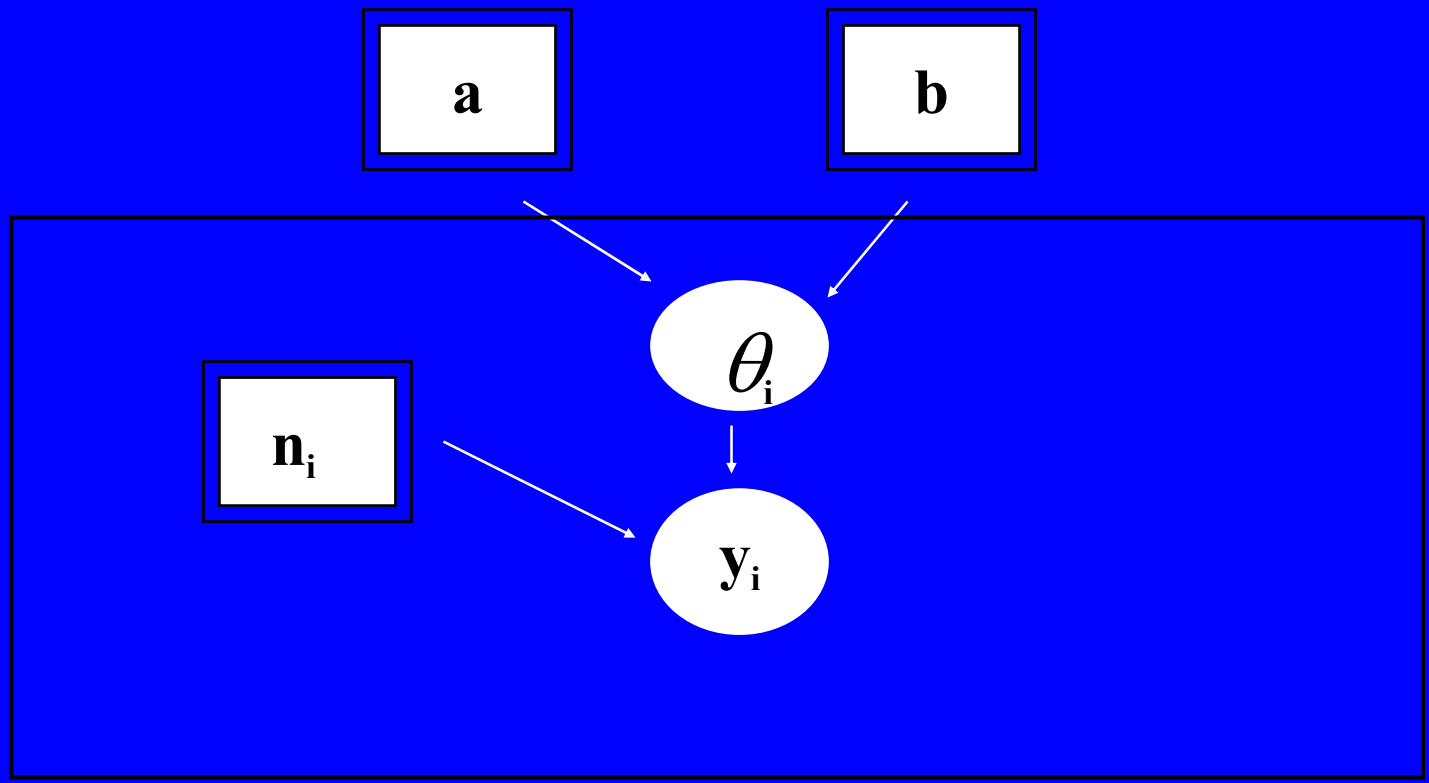
- Concentrate on **structural** relationships
- Directed, undirected and chain graphs
  - nodes represent random quantities
  - links represent relationships
  - missing links represent conditional independence
- Use graphs to:
  - break complex models into simple components
  - communicate essential structure
  - provide basis for computation

Example: Binomial model

- **Model**

$y_i \sim Binomial\ (\theta_i, n_i)$

$\theta_i \sim Beta\ (a,b)$



Explanation of Graph

3 types of node:

- Constants: double edged boxes
  no parents
- Stochastic: circles
  variables (data or parameters)
  given a probability distribution
  have solid arrows pointing to them
- Deterministic: circles
  logical functions of other nodes
  have dashed arrows pointing to them

Example: Logistic model

- **Model:**

  $y_i \sim Binomial\ (p_i, n_i)$

  $logit(p_i) = b_i$

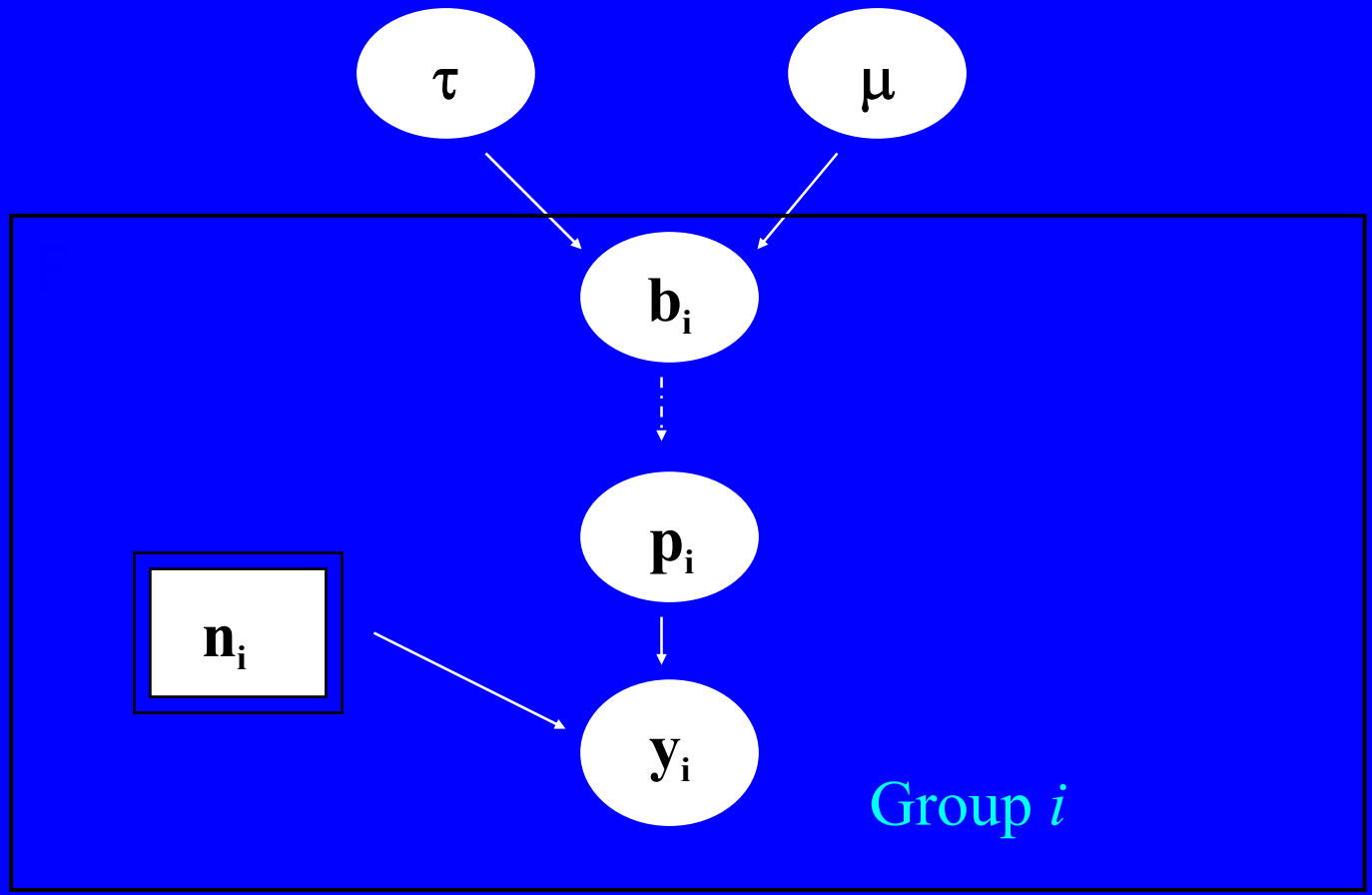
  $b_i \sim Normal\ (\mu, \tau)\ ,\ \tau = 1/\ \sigma^2$

- **Priors**

  $\mu \sim Normal\ (0,\ 1E\text{-}6\ )$

  $\tau \sim Gamma\ (1E\text{-}3,\ 1E\text{-}3\ )$

DAG for logistic model



Example: More rats

- **Explanation**

  30 rats, weighed weekly for 5 weeks.

  Model as random effects linear growth curve.

Weights Y_{ij} of rat i on day x_j

$x_j =$	8	15	22	29	36
Rat 1	151	199	246	283	320
Rat 2	145	199	249	293	354
…					
Rat 30	153	200	244	286	324

Model for Rats

- **Model**

$$y_{ij} \sim Normal\ (\alpha_i + \beta_i\ (x_j - x),\ \tau_C\ )$$

- **Priors**

$$\alpha_i \sim Normal\ (\ \alpha_C\ ,\ r_\alpha\ )$$

$$\beta_i \sim Normal\ (\ \beta_C\ ,\ r_\beta\ )$$

$$\alpha_C \sim Normal\ (\ 0\ ,\ 1E\text{-}4\ )$$

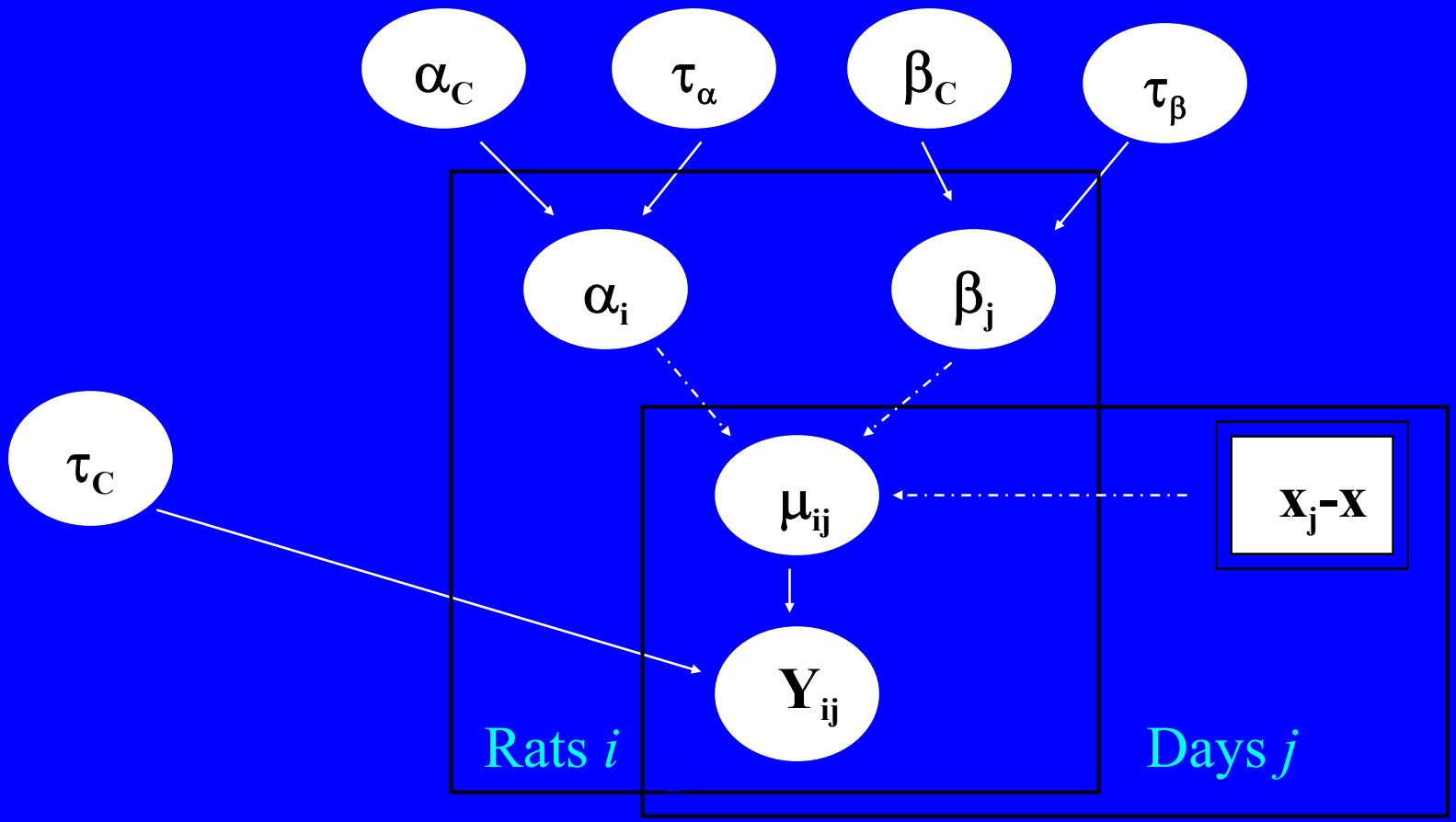
$$\beta_C \sim Normal\ (\ 0\ ,\ 1E\text{-}4\ )$$

$$\tau_C \sim Gamma\ (\ 1E{-}3,\ 1E{-}3\ )$$

$$\tau_\alpha \sim Gamma\ (\ 1E{-}3,\ 1E{-}3\ )$$

$$\tau_\beta \sim Gamma\ (\ 1E{-}3,\ 1E{-}3\ )$$

DAG for rats Logistic model



BUGS

Three current trends:

- Complex hierarchical (random-effects) models being analysed using S-plus, SAS etc
- Graphical models used in multivariate analysis
- Markov chain Monte Carlo (MCMC) methods turning Bayesian into mainstream statistics

Brought together in BUGS:

Bayesian Inference Using Gibbs Sampling

The BUGS Program

- Language for specifying complex directed graphical models
- Constructs graph by identifying parents and children
- Simulates via Gibbs and Metropolis-Hastings algorithms
- Currently restricted to particular distributions (discrete, conjugate, log-concave)

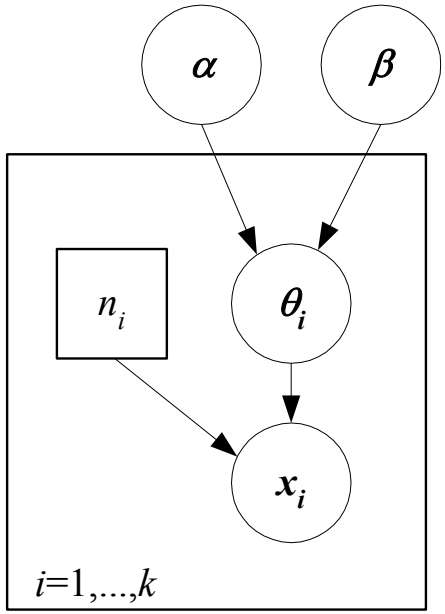
Example: probability of (death) after cardiac surgery?

- **12 hospitals**
- **Sample size (n) , deaths (y):**

n	**47**	**148**	**119**	**810**	**211**	**196**
y	**0**	**18**	**8**	**46**	**8**	**13**

n	**148**	**215**	**207**	**97**	**256**	**360**
y	**9**	**31**	**14**	**8**	**29**	**24**

PGM for Hospitals



stochastic

constant

plate

BUGS code

```
model surgical;
const
    N = 12 ;          # number of hospitals
var
    r[N], p[N], n[N], b[N], mu, tau, sigma, pop.mean;
data r, n in “surgical.dat”
inits in “surgical.in”
{
    for (i in 1:N)  {
    r[i] ~ dbin( p[i], n[i] );
    logit(p[i]) <- b[i];
    b[i] ~ dnorm( mu, tau);
}
# Priors:
mu ~ dnorm(0.0, 1.0E-6)
pop.mean <- exp(mu) / (1 + exp(mu));          # population mean
tau ~ dgamma (1.0E-3, 1.0E-3);      # 1/sigma^2
sigma <- 1/sqrt(tau)
}
```

Related BUGS files

- "surgical.dat" (BUGS data file)

r	*n*
0	47
…	
24	360

- "surgical.init" (BUGS initial value file)

list (tau=1, mu = 0)

Running BUGS: log file

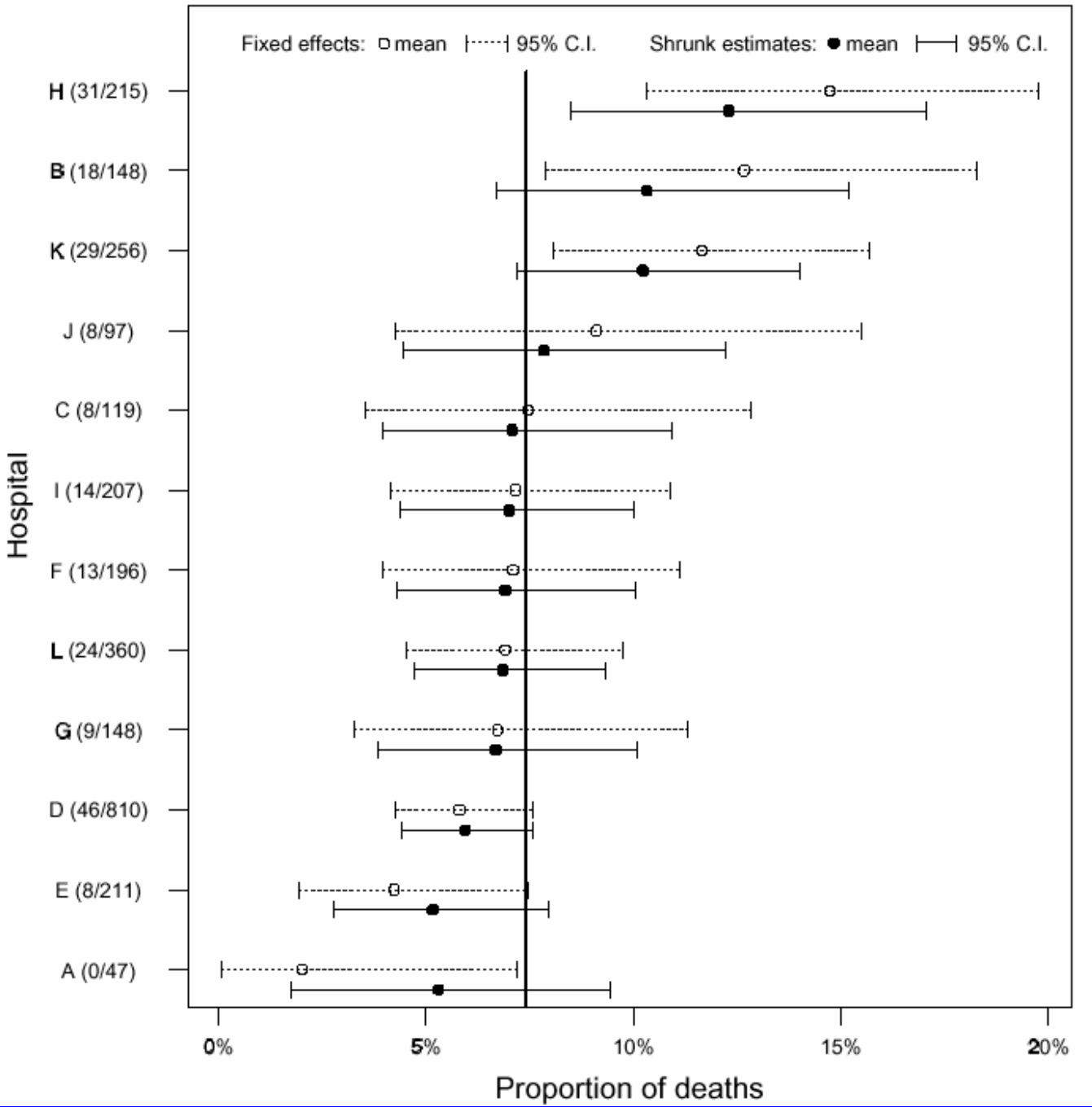
Bugs> compile ("surgical.bug")
Bugs> update(500)
500 updates took 00:00:04
Bugs> monitor (p)
Bugs> monitor (pop.mean)
Bugs> monitor (sigma)
Bugs> update (1000)
1000 updates took 00:00:08
Bugs> stats (p)

	mean	sd	2.5% : 97.5% CI	median	sample
[1]	5.17E-2	2.08E-2	1.50E-2 9.42E-2	5.01E-2	1000
…					
[12]	6.81E-2	1.20E-2	4.62E-2 9.33E-2	6.72E-2	1000

Bugs> stats (pop.mean)

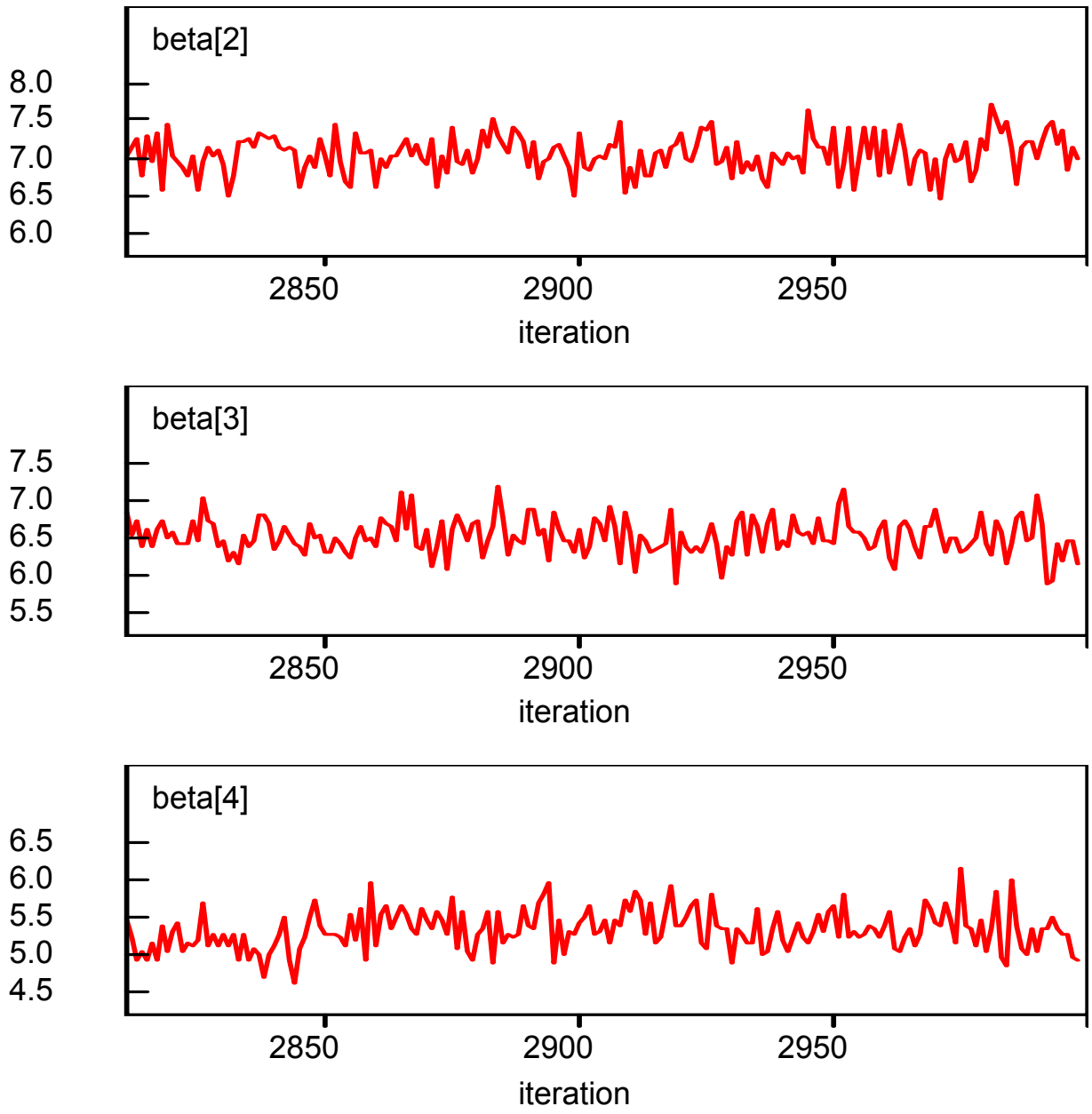
mean	sd	2.5% : 97.5% CI	median	sample
7.30E-2	1.07E-2	5.17E-2 9.49E-2	7.29E-2	1000

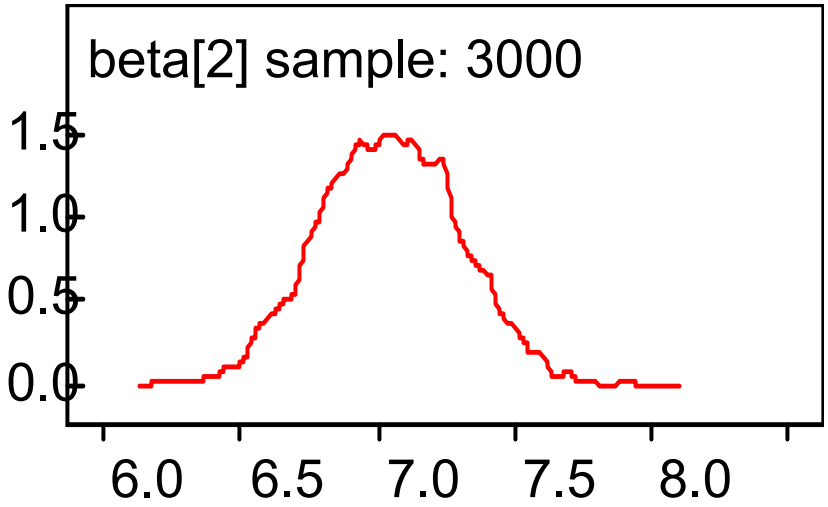
Bugs> q( )

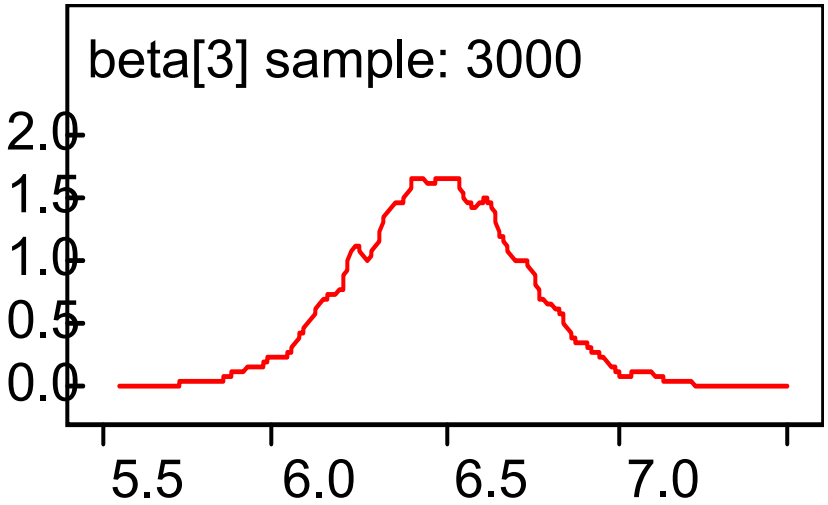

Fixed effects: mean 95% C.I.
Shrunk estimates: mean 95% C.I.
H (31/215)
B (18/148)
K (29/256)
J (8/97)
C (8/119)
I (14/207)
F (13/196)
L (24/360)
G (9/148)
D (46/810)
E (8/211)
A (0/47)
Hospital
0%
5%
10%
15%
20%
Proportion of deaths

Example: BUGS code for rats

```
model rats;
const
      N = 30 ;              # number of rats
      T = 5; # number of time points
var
      tau.c, alpha0, alpha.c, beta.c, x[T], mu[N,T], Y[N,T], alpha[N], beta[N], tau.alpha, tau.beta, x.bar;
data Y in “rats_y.dat”, x in “rats_x.dat”;
inits in “rats.in”
{
      for (i in 1:N)  {
            for (j in 1:T) {
                  mu[i,j] <- alpha[i] + beta[i] * (x[j] - x.bar);
                  Y[i,j] ~ dnorm(mu[i,j], tau.c)
            }
            alpha[i] ~ dnorm(alpha.c, tau.alpha);
            beta[i] ~ dnorm(beta.c, tau.beta);
      }
alpha.c ~ dnorm(0, 1.0E-4);
beta.c ~ dnorm(0, 1.0E-4);
tau.c ~ dgamma(1.0E-3, 1.0E-3);
tau.alpha ~ dgamma(1.0E-3, 1.0E-3);
tau.beta ~ dgamma(1.0E-3, 1.0E-3);
sigma <- 1.0 / sqrt(tau.c);
x.bar <- mean( x[] );
alpha0 <- alpha.c - beta.c * x.bar;
}
```


beta[2]
8.0
7.5
7.0
6.5
6.0
2850
2900
2950
iteration
beta[3]
7.5
7.0
6.5
6.0
5.5
2850
2900
2950
iteration
beta[4]
6.5
6.0
5.5
5.0
4.5
2850
2900
2950
iteration


beta[2] sample: 3000
1.5
1.0
0.5
0.0
6.0
6.5
7.0
7.5
8.0


beta[3] sample: 3000
2.0
1.5
1.0
0.5
0.0
5.5
6.0
6.5
7.0

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta[1]	6.063	0.2411	0.004325	5.595	6.065	6.521	1	3000
beta[2]	7.048	0.257	0.005173	6.563	7.049	7.548	1	3000
beta[3]	6.48	0.2471	0.004511	5.994	6.48	6.968	1	3000
beta[4]	5.345	0.2576	0.005856	4.851	5.345	5.864	1	3000
beta[5]	6.565	0.2532	0.005627	6.058	6.569	7.053	1	3000
beta[6]	6.178	0.2384	0.003631	5.72	6.174	6.65	1	3000
beta[7]	5.972	0.2469	0.005217	5.484	5.971	6.46	1	3000
beta[8]	6.413	0.2452	0.004439	5.919	6.414	6.889	1	3000
beta[9]	7.055	0.2542	0.005396	6.564	7.051	7.553	1	3000
beta[10]	5.848	0.2464	0.004784	5.353	5.85	6.34	1	3000

Example: regression

Consider a set of 5 observed (x, Y) pairs (1, 1), (2, 3), (3, 3), (4, 3), (5, 5). We shall fit a simple linear regression of Y on x, using the notation

$$Y_i \sim \text{Normal}(\mu_i, \tau)$$

$$\mu_i = \alpha + \beta(x_i - x.bar)$$

where *x.bar* represents the mean of the *x*'s. Note that we parameterise the normal distribution in terms of its precision τ, which is 1/variance.

```
model
{
      for(i in 1:N){
            Y[i] ~ dnorm(mu[i], tau)
            mu[i] <- alpha + beta * (x[i] -
mean(x[]))
      }
      sigma <- 1/sqrt(tau)
      alpha ~ dnorm(0, 1.0E-6)
      beta ~ dnorm(0, 1.0E-6)
      tau ~ dgamma(1.0E-3, 1.0E-3)
}
```

What about convergence?

- Theoretical

$$\sup_{x \in C} |P^n(x,C) - P^\infty(C)| \leq M \rho^n_C$$
$$\int P(x,dy) V(y) \leq (1-\beta) V(x) + I_C(x)$$

- Diagnostics

CODA

- **Output processor for BUGS**
- **Menu-driven set of S-Plus functions for:**
- **Convergence diagnosis**
  - specific methods
  - autocorrelations and cross-correlations
- **Summary statistics**
  - empirical mean, sd, quantiles
  - standard error of the mean
- **Graphical**
  - sample trace for each variable
  - kernel density
  - plots of some convergence diagnostics

Convergence: Geweke (1992)

- Look at a single long run
- Test for equal mean for "early" part (1st quarter) and "late" part (second half) of the chain.
- Test statistic is $Z \sim N(0,1)$ if the sample is all from the same distribution.
- Careful: this is only a test of "non-convergence" and can be misleading.

Convergence: Gelman & Rubin (1992)

- Many long runs
- Widely different starting points
- Convergence assessed via an "analysis of variance" between and within the chains.
- Monitor convergence by R: a conservative estimate of how much extra information about the variable that we could expect to gain by running the chains indefinitely
  - R tends to 1 as n tends to infinity
  - R is subject to sampling variation so monitor
    - R *and* is upper 97.5% confidence limit
- Works best when posterior is approx. normal (may need to transform some variables, eg probs, variances)

Convergence: Raftery & Lewis (1992)

- Look at a single long run
- Diagnostic estimates:
  n0: length of burnin
  N = no. additional iterations needed to estimate a posterior quantile adequately
- Chain must be run for at least Nmin iterations before computing diagnostic
- Can give radically different estimates depending on starting values and required accuracy of estimation
- Can *under-estimate* n0 for extreme quantiles
- Must re-diagnose convergence for each quantile.
- Based on 2-state Markov chain theory.

Convergence: Heidelberger & Welch (1983)

- Look at a single long run
- Hypothesis test based on Brownian bridge theory and spectral density estimation
- Iterative procedure:
  - test H0: entire sample of values for a given variable form a stationary process
  - if H0 rejected, discard first 10% and repeat test
  - continue discarding until H0 accepted or 50% samples are discarded (need to run chain for longer)
- Also estimates numerical S.E. of mean and tests size of C.I.
- Test has *very low power* to detect lack of convergence for small sample size.

CODA Menus

- CODA Main Menu:
  - Output Analysis - Diagnostics
  - List/Change Defaults - Quit
- CODA Output Analysis Menu
  - Plots - Statistics
  - List/Change Defaults - Return to Main
- CODA Diagnostics Menu
  - Geweke, Gelman and Rubin, Raftery and Lewis, Heidelberger and Welch, Autocorrelations, Cross-Correlations
  - List/Change Defaults - Return to Main

CODA Output: Surgical Eg

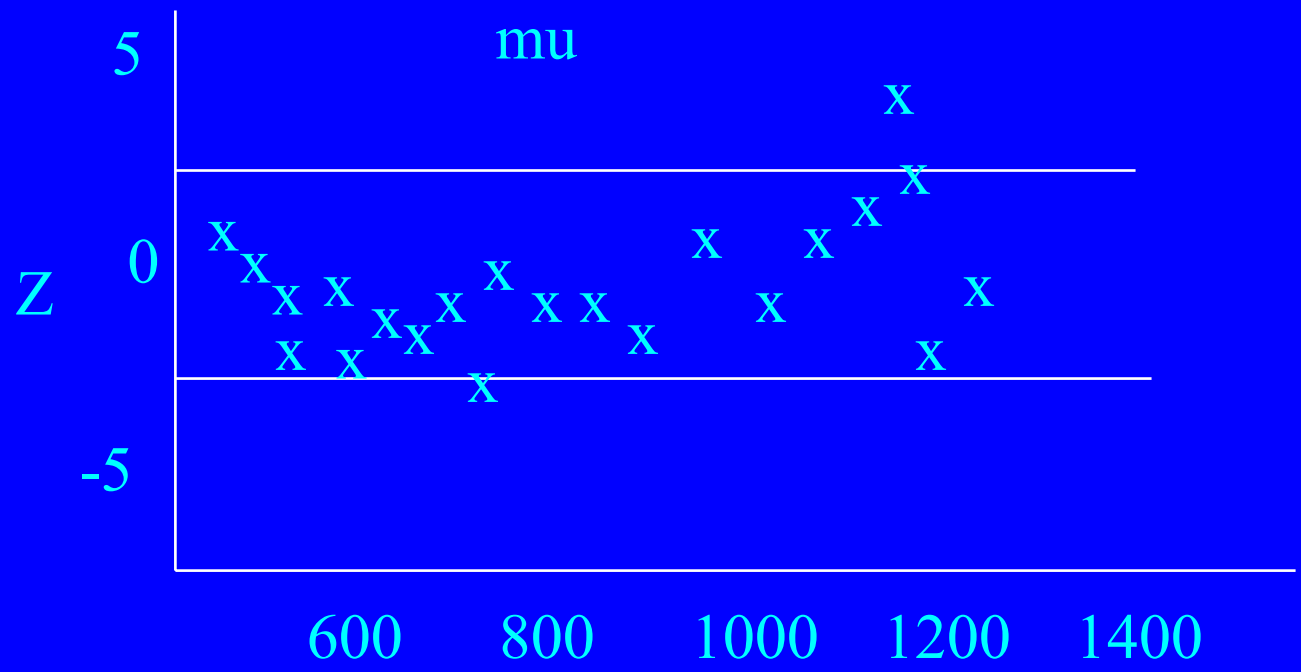
- Trace plot, Kernel density plot
- Summary statistics
- Quantiles for each variable
- Autocorrelations
- Cross-correlations

Geweke Z-score

Iterations used = 501:1500
Thinning interval = 1
Sample size per chain = 1000
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

Var:	mu	p[1]	p[2]	sigma
Z:	0.372	1.650	-2.550	-1.150

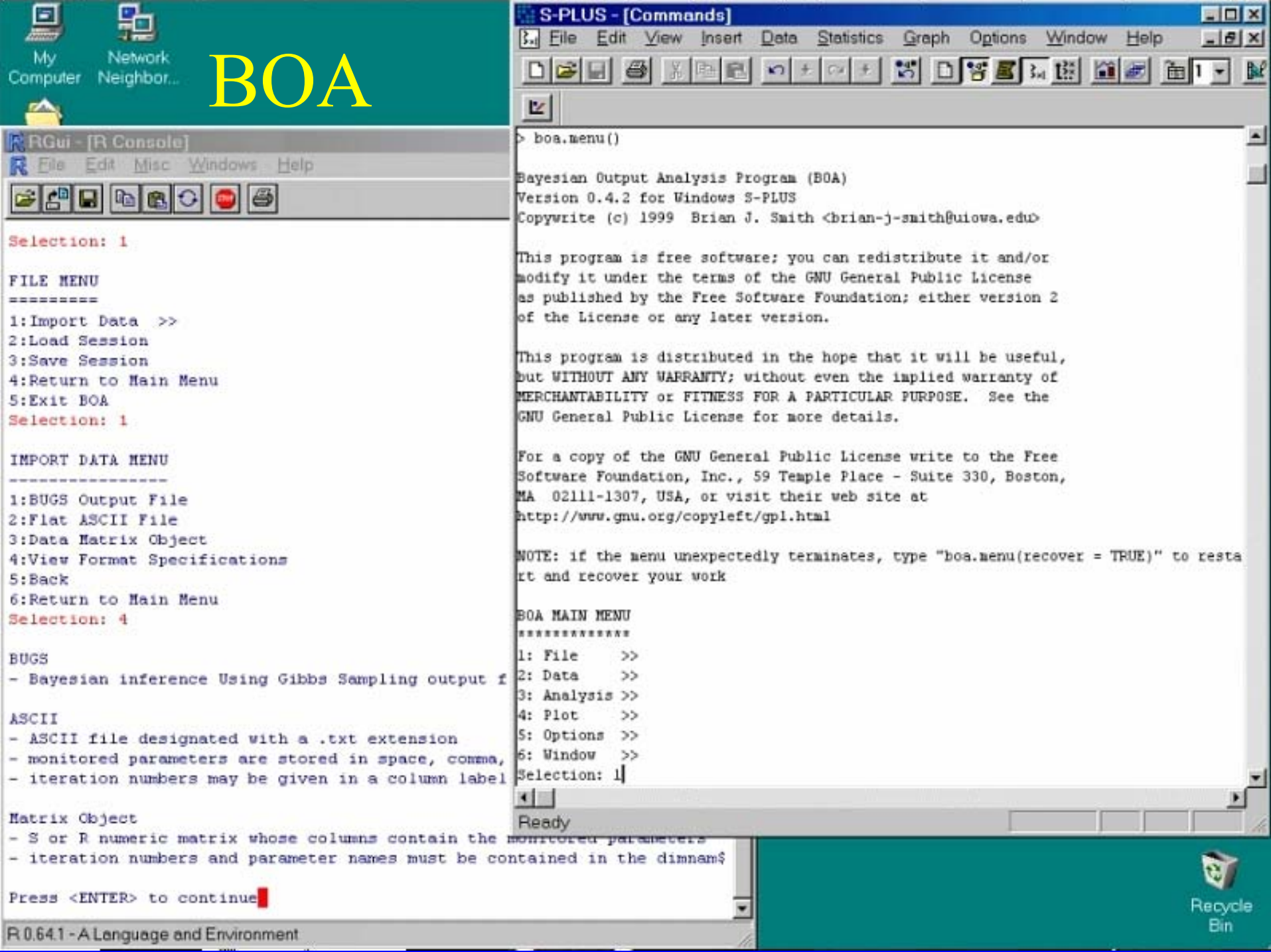
Geweke z-plots



Gelman and Rubin 50% and 97.5% shrink factors

Variable	Point est.	97.5% quantile
mu	1.0	1.01
p[1]	1.02	1.10
p[2]	1.00	1.00
sigma	1.02	1.10

Trace plots, shrink factor plots


My Computer
Network Neighbor...
BOA
RGui - [R Console]
File Edit Misc Windows Help
Selection: 1
FILE MENU
1:Import Data >>
2:Load Session
3:Save Session
4:Return to Main Menu
5:Exit BOA
Selection: 1
IMPORT DATA MENU
1:BUGS Output File
2:Flat ASCII File
3:Data Matrix Object
4:View Format Specifications
5:Back
6:Return to Main Menu
Selection: 4
BUGS
- Bayesian inference Using Gibbs Sampling output f
ASCII
- ASCII file designated with a .txt extension
- monitored parameters are stored in space, comma,
- iteration numbers may be given in a column label
Matrix Object
- S or R numeric matrix whose columns contain the monitored parameters
- iteration numbers and parameter names must be contained in the dimnam$
Press <ENTER> to continue
R 0.64.1 - A Language and Environment
S-PLUS - [Commands]
File Edit View Insert Data Statistics Graph Options Window Help
> boa.menu()
Bayesian Output Analysis Program (BOA)
Version 0.4.2 for Windows S-PLUS
Copywrite (c) 1999 Brian J. Smith <brian-j-smith@uiowa.edu>
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License or any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
For a copy of the GNU General Public License write to the Free
Software Foundation, Inc., 59 Temple Place - Suite 330, Boston,
MA 02111-1307, USA, or visit their web site at
http://www.gnu.org/copyleft/gpl.html
NOTE: if the menu unexpectedly terminates, type "boa.menu(recover = TRUE)" to resta
rt and recover your work
BOA MAIN MENU
1: File >>
2: Data >>
3: Analysis >>
4: Plot >>
5: Options >>
6: Window >>
Selection: 1
Ready
Recycle Bin

BOA (S-PLUS for Linux)

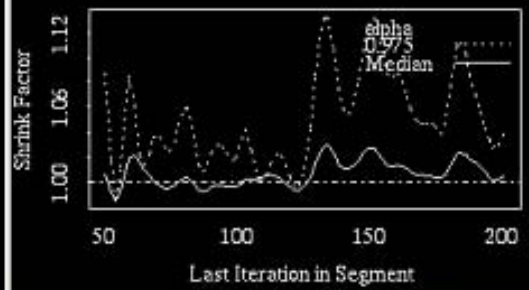
```
BROOKS, GELMAN AND RUBIN CONVERGENCE DIAGNOSTICS:
================================================

Iterations used = 101:200

Potential Scale Reduction Factors
---------------------------------

   alpha    beta   sigma
1.00316 1.00171 1.00841

Multivariate Potential Scale Reduction Factor = 1.026086

Corrected Scale Reduction Factors
---------------------------------

      Estimate   0.975
alpha  1.00596 1.03859
 beta  1.00940 1.03702
sigma  1.01147 1.06440

Press <ENTER> to continue

CONVERGENCE DIAGNOSTICS MENU
----------------------------
1: Brooks, Gelman & Rubin
2: Geweke
3: Heidelberger & Welch
4: Raftery & Lewis
5: Back
6: Return to Main Menu
Selection: 2

GEWEKE CONVERGENCE DIAGNOSTIC:
==============================

Fraction in first window = 0.1
Fraction in last window = 0.5

Chain: line1
------------

          alpha     beta   sigma
Z-Score 0.251456 -3.33816 3.07513

Press <ENTER> to continue

Chain: line2
------------

           alpha      beta     sigma
Z-Score -0.466613 -0.993461 -0.818046

Press <ENTER> to continue
```

S-PLUS <2>

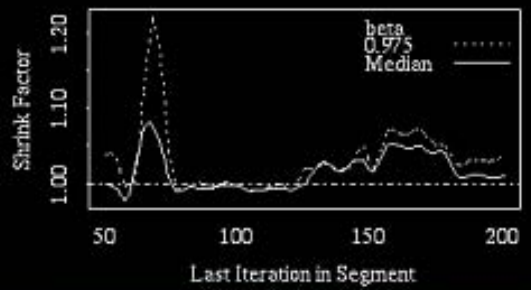
Graph Options Help

BUGS Line Example

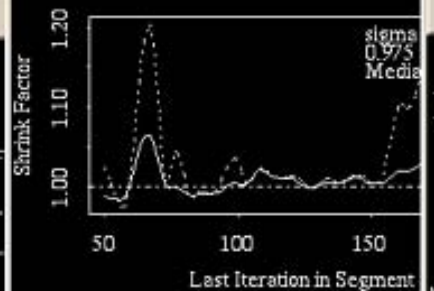
Gelman & Rubin Shrink Factors



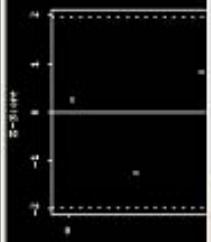
Gelman & Rubin Shrink Factors



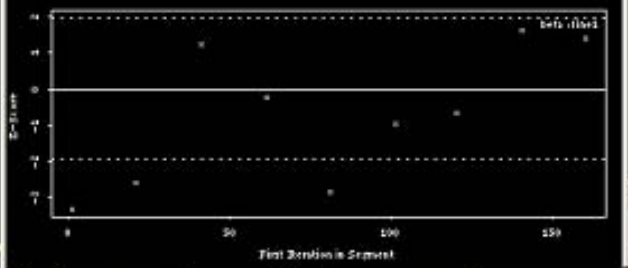
Gelman & Rubin Shrink



Graph Options



Geweke Convergence Diagnostic

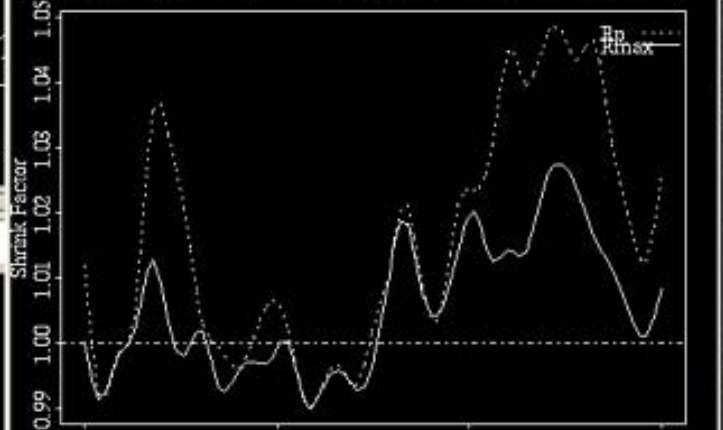


S-PLUS

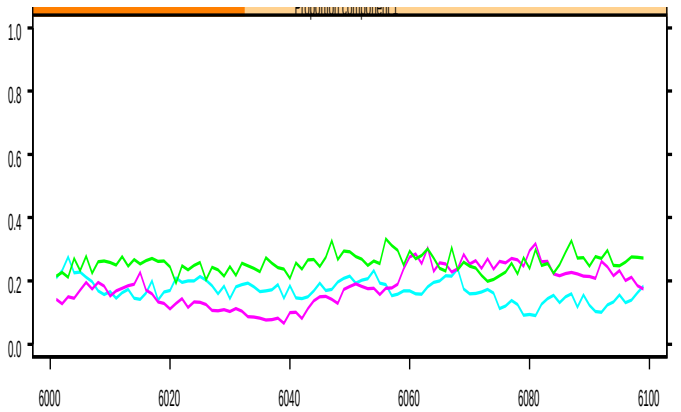
Graph Options Help

BUGS Line Example

Brooks & Gelman Multivariate Shrink Factors



A new diagnostic: phase randomisation



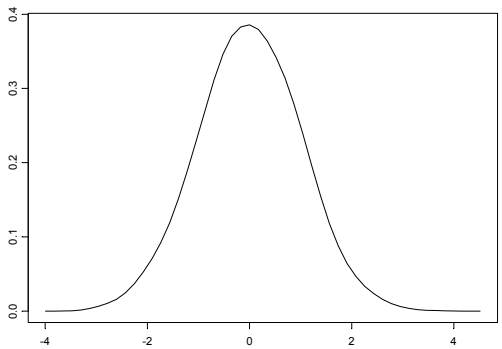
- Run a single chain
- Take Fourier transform
- Randomise phase
- Backtransform

(Phase scrambling, Fourier bootstrap)

Modality and scale of 3rd Cumulant tells us about:

- Linearity
- Stationarity

of the original series



Imagine

Imagine you're a Bayesian
It's easy if you try,
You just adopt a prior,
And the data updates π.
Statistics is so simple
With subjective probabilityyyyy -- ah-ah! ah ah...

Now imagine you're a frequentist,
Worrying about what might have been,
Spending your whole lifetime
Analyzing data you've never seen.
And if you want an interval,
You'll need a pivotal quantityyyyy -- ah-ah! ah ah...

You may say I sound like Nozer --
But I'm not the only one:
Every four years we all get together,
To talk, drink beer, and lie in the sun.