Stata goes BUGS (via R)

Susumu Shikano
Department of Political Science I
University of Mannheim

31. March 2006
4th German Stata Users’ Group Meeting

press “ctrl + l” to start presentation
Problem 1

Problem

You are a Stata user . . .

- and have a complicated likelihood function hard (or impossible) to maximize, or . . .
- have not enough number of observations for a large amount of parameters to estimate.
You are a Stata user . . .

- and have a complicated likelihood function hard (or impossible) to maximize, or . . .
- have not enough number of observations for a large amount of parameters to estimate.

One possible solution

- You forget the maximization (at a moment or forever) and take the Bayesian methods.
Another philosophy

- observed data: fixed; unknown parameters: random
- MCMC (Markov Chain Monte Carlo) provides estimated distribution of interested parameters.
Bayesian Methods

Another philosophy

- observed data: fixed; unknown parameters: random
- MCMC (Markov Chain Monte Carlo) provides estimated distribution of interested parameters.

Some practical advantages

- Fitting a wider range of models
  - Modelling latent variables
  - Estimation of hierarchical models
- Analyzing a wider range of data
  - Analyzing small samples
  - Treating missing data properly
- Systematic incorporation of your prior knowledge
- Intuitive interpretation of results
Problem

Bayesian methods and MCMC are not incorporated in Stata.
Problem 2

Problem
Bayesian methods and MCMC are not incorporated in Stata.

Solution?
BUGS: Bayesian updating using Gibbs sampling
- WinBUGS (for Windows)
- OpenBUGS (for Linux)
- JAGS (platform independent)
Problem 3

Problem

No interface for WinBUGS is available in Stata.
Problem 3

Problem
No interface for WinBUGS is available in Stata.

Solution?
R
- is well equipped with interfaces to Stata as well as WinBUGS.
- can be used as interface between Stata and WinBUGS.
Stata goes BUGS

Stata goes BUGS step by step

1. You call R from inside Stata.
Stata goes BUGS

Stata goes BUGS step by step

1. You call R from inside Stata.
2. R reads, transforms, and hands your data to WinBUGS
You call R from inside Stata.
R reads, transforms, and hands your data to WinBUGS which can be started from R.
Stata goes BUGS

Stata goes BUGS step by step

1. You call R from inside Stata.
2. R reads, transforms, and hands your data to WinBUGS which can be started from R.
3. WinBUGS gives estimation results to R
Stata goes BUGS

Stata goes BUGS step by step

1. You call R from inside Stata.
2. R reads, transforms, and hands your data to WinBUGS which can be started from R.
3. WinBUGS gives estimation results to R
4. which are transformed into the Stata format.
Stata goes BUGS step by step

1. You call R from inside Stata.
2. R reads, transforms, and hands your data to WinBUGS which can be started from R.
3. WinBUGS gives estimation results to R which are transformed into the Stata format.
4. You have results in Stata.
Stata goes BUGS via R step by step

1. You call R from inside Stata.
2. R reads, transforms, and hands your data to WinBUGS which can be started from R.
3. WinBUGS gives estimation results to R which are transformed into the Stata format.
4. You have results in Stata.
Setting up

Software to install

- R and R2WinBUGS (Additional Package)
- WinBUGS
  (should be installed in “c:/Programme/WinBUGS14“)

Files

- GoWinBUGS.R (You don’t have to edit it.)
- GoWinBUGSModel.bug (your model code)
To run WinBUGS

<table>
<thead>
<tr>
<th>You need ...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>model</strong> (including priors)</td>
</tr>
<tr>
<td><strong>data file</strong></td>
</tr>
<tr>
<td><strong>initial values</strong></td>
</tr>
</tbody>
</table>
You save `GoWinBUGS.R` and `GoWinBUGSMModel.bug` in a directory and change your stata working directory to it.
You save `GoWinBUGS.R` and `GoWinBUGSModel.bug` in a directory and change your Stata working directory to it.

You save your Stata-data as `dataToR.dta`.

- Only needed variables!
How to go WinBUGS

- You save `GoWinBUGS.R` and `GoWinBUGSMModel.bug` in a directory and change your Stata working directory to it.
- You save your Stata-data as `dataToR.dta`.
  - Only needed variables!
- You edit `GoWinBUGSMModel.bug` according to the model to be estimated.
How to go WinBUGS

- You save GoWinBUGS.R and GoWinBUGSMModel.bug in a directory and change your stata working directory to it.
- You save your Stata-data as dataToR.dta.
  - Only needed variables!
- You edit GoWinBUGSMModel.bug according to the model to be estimated.
- Run following command:
  ```shell```
  shell "C:/Programme/R/R-2.2.1/bin/R.exe" CMD BATCH "GoWinBUGS.R"
  ```shell```
How to go WinBUGS

- You save GoWinBUGS.R and GoWinBUGSModel.bug in a directory and change your stata working directory to it.
- You save your Stata-data as dataToR.dta.
  - Only needed variables!
- You edit GoWinBUGSModel.bug according to the model to be estimated.
- Run following command:
  shell “C:/Programme/R/R-2.2.1/bin/R.exe“ CMD BATCH “GoWinBUGS.R“
- You will have following output files in your working directory:
  - dataFromR.dta: Posterior distribution in Stata Format
  - bugOutput1.pdf and bugOutput2.pdf: some graphics
  - GoWinBUGS.Rout: log of R
Simulated data based on a probit model

\[ Pr(y_i = 1) = F(\beta_1 + \beta_2 x_i) \]  

F is here the probit CDF transformation.
GoWinBUGS.do

clear
cd "c:/Konferenzen/StataMeeting"

/* setting parameters */
set obs 100 /* setting the number of obs.*/
scalar beta1=5 /* constant */
scalar beta2=7 /* coef for x */
scalar wt=3 /* weight of error term */

/* generating data */
gen x =uniform()*2 -1 /* generating independent variable */
gen e = invnorm(uniform()) /* generating error term */
gen y = beta1+beta2*x+wt*e /* generating latent variable */
replace y =0 if y<=0.5
replace y =1 if y>0.5
GoWinBUGS.do

clear
cd "c:/Konferenzen/StataMeeting"

/* setting parameters */
set obs 100 /* setting the number of obs.*/
scalar beta1=5 /* constant */
scalar beta2=7 /* coef for x */
scalar wt=1 /* weight of error term */

/* generating data */
gen x =uniform()*2 -1 /* generating independent variable */
gen e = invnorm(uniform()) /* generating error term */
gen y = beta1+beta2*x+wt*e /* generating latent variable */
replace y =0 if y<=0.5 /* generating latent variable */
replace y =1 if y>0.5
your do-file

GoWinBUGS.do

```stata
clear

cd "c:/Konferenzen/StataMeeting"

/* setting parameters */
set obs 100 /* setting the number of obs.*/
scalar beta1=5 /* constant */
scalar beta2=7 /* coef for x */
scalar wt=3 /* weight of error term */

/* generating data */
gen x =uniform()*2 -1 /* generating independent variable */
gen e = invnorm(uniform()) /* generating error term */
gen y = beta1+beta2*x+wt*e /* generating latent variable */
replace y =0 if y<=0.5 /* generating latent variable */
replace y =1 if y>0.5
```
GoWinBUGS.do

/* probit analysis */
probit y x

/* going BUGS */
keep y x /* keep only variables for WinBUGS */
save dataToR,replace /* save data set */
shell "C:/Programme/R/R-2.2.1/bin/R.exe" CMD BATCH "GoWinBUGS.R"

/* getting results from R */
use dataFromR, clear
/* summary statistics of posterior distribution */
sum, detail
your do-file

GoWinBUGS.do

/* probit analysis */
probit y x

/* going BUGS */
keep y x /* keep only variables for WinBUGS */
save dataToR,replace /* save data set */
shell "C:/Programme/R/R-2.2.1/bin/R.exe" CMD BATCH "GoWinBUGS.R"

/* getting results from R */
use dataFromR, clear /* summary statistics of posterior distribution */
sum, detail
Model

GoWinBUGSModel.bug (preamble: not read by WinBUGS, but by R)

```plaintext
# #
# # interested parameters: beta
# # initial values: beta=0,3 ←initial values; Comma (,) between values
# n.burnin: 5000 ←length of burn in
# n.iter: 10000 ←no of iterations
# n.thin: 1 ←thinning rate
# N (no. of rows)?: yes
# J (no. of cols)?: no ←if yes, you can inspect results in WinBUGS.
# debug?: no
# #
# binary probit
# #
```

Susumu Shikano

Stata-Meeting
GoWinBUGSModel.bug (model)

```plaintext
model{
  for (i in 1:N){
    mu[i] <- beta[1] + beta[2]* x[i];
    # # the trick from Jackman (2000)
    # # otherwise WinBUGS would give an error
    ystar[i] ~ dnorm(mu[i],1)I(lo[y[i]+1],up[y[i]+1]);
    probit(p[i]) <- ystar[i];
  }
  lo[1] <- -50; up[1] <- 0; # # interval for ystar | y=0
  lo[2] <- 0; up[2] <- 50; # # interval for ystar | y=1
  # # priors
  beta[1] ~ dnorm(0,0.0001) ← Here you can
  beta[2] ~ dnorm(0,0.0001) ← change priors.
}
```

Ideological Positions of Political Actors

Relevant for Analysis of . . .

- Voting behavior
- Party competition
- Coalition building
- Policy making process
- etc.
Ideological Positions of Political Actors

Relevant for Analysis of . . .

- Voting behavior
- Party competition
- Coalition building
- Policy making process
- etc.

Data source

- Expert Survey
- Mass Survey
- Party Manifesto
- Recorded vote, or “roll call vote”
Applying Item-Response Theory

\[ Pr(y_{ij} = \text{Yes}) = F(\gamma_j - \beta_j x_i) \]  

(2)

F is here the logit CDF transformation.
Item-Response Theory

Applying Item-Response Theory

\[ Pr(y_{ij} = \text{Yes}) = F(\gamma_j - \beta_j x_i) \]  

(2)

F is here the logit CDF transformation.
Applying Item-Response Theory

\[ Pr(y_{ij} = \text{Yes}) = F(\gamma_j - \beta_j x_i) \]  

where \( F \) is the logit CDF transformation.

F is here the logit CDF transformation.
Ideological Position of German Federal States

Source: Frankfurter Allgemeine Zeitung
Roll calls in German Bundesrat

Bundesrat

- The second chamber consisted of the representatives of 16 state governments
- The representatives of a state vote unanimously.

Data

- 729.-813. Session (1998-2005; during the red-green government)
- J= 20 (all but unanimous roll calls); N=16
Roll calls in German Bundesrat

Data

- 729.-813. Session (1998-2005; during the red-green government)
- J= 20 (all but unanimous roll calls); N=16

|     | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | v12 | v13 | v14 | v15 | v16 | v17 | v18 | v19 | v20 |
|-----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| BW  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1   | 1   | 1   | 0   | 0   | 1   | 1   | 1   | 1   | 1   | 1   |
| BY  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  | 1   | 1   | 1   | 0   | 1   | 0   | 1   | 1   | 1   | 1   | 1   |
| BE  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 1   |
| BB  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| HB  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   |
| HH  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| HE  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| MV  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| NI  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| NW  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| RP  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| SL  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| SA  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| ST  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| SH  | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| TH  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
Roll calls in German Bundesrat

Data
- 729.-813. Session (1998-2005; during the red-green government)
- J= 20 (all but unanimous roll calls); N=16

Challenge
- With 16 × 20 data
- 20 (γ) + 20 (β) + 16 (x) parameters to estimate
Model

GoWinBUGSModel.bug (preamble: not read by WinBUGS, but by R)

```plaintext
# # interested parameters: beta, gamma, x
# # initial values: beta=3,-3,-3,-1,3,2,3,3,2,1,-1,-3,-2,1,-1,3,-1,-1,1,0;gamma=0,0,0,-3,-1,-1,-1,-1,-1,-1,-3,-1,-1,-1,-1,-3,0,-1,0,1,-2;x=1,1,-1,-1,0,0,1,-1,0,-1,-1,1,1,0,-1,1
# # n.burnin: 5000
# # n.iter: 10000
# # n.thin: 1
# # N (no. of rows)?: yes
# # J (no. of cols)?: yes
# # Matrix? : yes
# # debug?: no
# # IRT
```

← Data are read as matrix y[,]
GoWinBUGSModel.bug (model)

model{
  for (i in 1:N) {
    for (j in 1:J) {
      logit(p[i,j]) <- gamma[j] - beta[j] * x[i] ;
      y[i,j] ~ dbern(p[i,j]);
    }
  }
  # # prior
  for (i in 1:N) {
    x[i] ~ dnorm(0,1) # # prior for ideal points
  }
  for (j in 1:J) {
    gamma[j] ~ dnorm(0,0.25)
    beta[j] ~ dnorm(0,0.25)
  }
}

Susumu Shikano
Estimation results for $x$
Further possibilities

- Change of government - Introducing missing values
Further possibilities

- Change of government - Introducing missing values

<table>
<thead>
<tr>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
<th>v5</th>
<th>v6</th>
<th>v7</th>
<th>v8</th>
<th>v9</th>
<th>v10</th>
<th>v11</th>
<th>v12</th>
<th>v13</th>
<th>v14</th>
<th>v15</th>
<th>v16</th>
<th>v17</th>
<th>v18</th>
<th>v19</th>
<th>v20</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BY</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BE</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BB</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>HB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HH</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>HE</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MV</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NI</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>NW</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>RP</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SL</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SA</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ST</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SH</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TH</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Further possibilities

- Change of government - Introducing missing values
- Not only “Yeas“ and “Nays“, but also Abstention
  - Ordinal Item-Response Model
Further possibilities

- Change of government - Introducing missing values
- Not only “Yeas“ and “Nays“, but also Abstention - Ordinal Item-Response Model

![Graph showing probability of Yes and No responses against ideology.](image-url)
Further possibilities

- Change of government - Introducing missing values
- Not only “Yeas“ and “Nays“, but also Abstention
  - Ordinal Item-Response Model
Main messages

- Stata user can now use MCMC.
Main messages

- Stata user can now use MCMC.
- but, still in a roundabout way.
Conclusion

Main messages

- Stata user can now use MCMC.
- but, still in a roundabout way.
- Why not its own implementation or, at least, an interface to WinBUGS?
FAQ, or frequently heard arguments

1. MCMC is a technique only for hard core scientists. Demands for MCMC are limited.
FAQ, or frequently heard arguments

1. MCMC is a technique only for hard core scientists. Demands for MCMC are limited. Development of technology and software can encourage a wider range of users to apply MCMC.
Conclusion

**FAQ, or frequently heard arguments**

1. MCMC is a technique only for hard core scientists. Demands for MCMC are limited.
2. Why Stata? You can use R or learn WinBUGS.
FAQ, or frequently heard arguments

1. MCMC is a technique only for hard core scientists. Demands for MCMC are limited.

2. Why Stata? You can use R or learn WinBUGS.
   - Same answer to point 1.
   - Teaching
FAQ, or frequently heard arguments

1. MCMC is a technique only for hard core scientists. Demands for MCMC are limited.
2. Why Stata? You can use R or learn WinBUGS.
3. Writing model for themselves is also painful.
FAQ, or frequently heard arguments

1. MCMC is a technique only for hard core scientists. Demands for MCMC are limited.
2. Why Stata? You can use R or learn WinBUGS.
3. Writing model for themselves is also painful.
   - Problem specific packages (IRT, MNP, etc.)
   - Writing own model is didactically meaningful.
Acknowledgements

Special thanks to Alex Herzog for his advices in Stata.

Disclaimer

- Programs presented here are downloadable under:
  http://webrum.uni-mannheim.de/sowi/shikanos/software
- They were written for relatively simple models.
- Choose your initial values carefully. If your choice is bad, WinBUGS can go on strike.
Useful Links

- http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml
  Homepage of WinBUGS

- http://www.r-project.org/
  Homepage of R

- http://cran.r-project.org/src/contrib/Descriptions/
  R2WinBUGS.html
  R2WinBUGS (also can be installed per internet from inside R)
Literature

