BayesianModeling
User Manual

Version 1.0 December 2011

Jorge Luis Bazán, PhD

Department of Sciences

Pontifical Catholic University of Peru

email: jlbazan@pucp.edu.pe

Home Page: http://argos.pucp.edu.pe/~jlbazan/

Index **_Page_**

Introduction

BayesianModeling is a java software development tool to generate syntax of several models of Binary Regression and Item Response Theory under a Bayesian approach using Markov chain Monte Carlo (MCMC) methods. Subsequently, the syntax can be executed in the programs OpenBUGS (Spiegelhalter, Thomas, Best, Lunn, 2007), WinBUGS (Spiegelhalter, Thomas, Best, Lunn, 2003) or in R program (R Development Core Team, 2004) through the libraries R2WinBUGS and R2OpenBUGS (Sturtz, Ligges and Gelman, 2005), BRugs (Thomas, et al, 2006) and rbugs.

This application is for your personal use and must not be used for any commercial purpose whatsoever without my explicit written permission. The application is provided "as is" without warranty of any kind. In order appropriately to implement the different models mentioned in the application, you it must read in detail the literature suggested in the references and must be familiarized with the Bayesian Inference using MCMC.

BayesianModeling is thought for practitioners that given a data set they wish to know the syntax of diverse Binary Regression or Item Response models in bugs code, Theory usually non available in diverse statistical programs including the program *R*. This program write two files: a bugs model file for each one of this models considering adequate priors, lists with sensible starting values and size of the data set and a data set file in rectangular format, both readable in `WinBUGS` or `OpenBUGS`.

This basic application can be considered a different version of BRMUW (Bazán, 2010) which was developed as part of the projects DAI 3412, 4031 and 2009-0033 of Pontifical Catholic University of Peru with the purpose to disseminate models developed by the author.

This application together with some models has been developed for three late years and throughout that time various people have collaborated in this project reason why I desire to express my gratefulness. Among them to the colleagues Oscar Millones, Christian Bayes and Miluska Osorio for theirs aid during the revision of the present version of the application and the user guide. I also am thankful to Martin Iberico, Margareth Sequeiros and Pedro Curich for the computational support in some of the stages of the project. Thank you very much to my family for its patience and support.

Installation instructions can be found in the included README file. *BayesianModeling* can run smoothly in any operating systems, such as Windows, Mac OS, Linux, in which the Java Virtual Machine and Perl are supported.
Java: Java SE Runtime Environment 1.6 or later. Perl: Perl v5.10 or later. RAM: ≥ 512M
The *BayesianModeling* package was introduced at the II Conbratri-2011 (http://187.45.202.74/conbratri/).

1. Binary Regression

1.1. A Binary Regression Models

Consider $\mathbf{y} = (y_1, y_2, \ldots, y_n)'$ a $n \times 1$ vector of n independent dichotomous random variables, assuming that $y_i = 1$ with probability p_i and $y_i = 0$ with probability $1 - p_i$, and $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})'$ a $k \times 1$ vector of covariates, where x_{i1} may equals 1, corresponding to an intercept, $i = 1, \ldots, n$. Moreover, $\mathbf{X}$ denotes the $n \times k$ design matrix with rows x_i', and $\beta = (\beta_1, \ldots, \beta_k)'$ is a $k \times 1$ vector of regression coefficients. Binary regression models assume that $p_i = F(\eta_i), \quad i = 1, \ldots, n,$ where $F(.)$ denotes a cumulative distribution function (cdf). The inverse function F^{-1} is typically called the link function and $\eta_i = \mathbf{x}_i'\beta$ $= \beta_1 + \beta_2 x_{i2} + \ldots + \beta_k x_{ik}$ is the linear predictor. Thus, a binary regression model is given by

$$
\begin{gathered}
y_i \sim Bernoulli(p_i) \\
p_i = F(\eta_i) = F(\mathbf{x_i'}\beta), \quad i = 1, \ldots, n,
\end{gathered}
$$

When F a cdf of a symmetric distribution, the response curve is has symmetric form about 0.5. Examples are obtained when F is in the class of the elliptical distributions as, for example, standard normal, logistic, Student-t, double exponential and Cauchy distributions.

In the case that F is the cdf of a standard normal distribution we obtain the *probit link*

$$F(t) = \Phi(t)$$

and in the case that F is the cdf of a logistic distribution we obtain the *logit link*,

$$F(t) = L(t) = \frac{e^t}{1 + e^t}.$$

These links: probit and logit are implemented in *BayesianModeling*.

1.2. Asymmetric Links in Binary Regression

As reported in the literature, symmetric links are not always appropriate for modeling this kind of data. Nagler (1994), Chen, Dey and Shao (1999) among others, showed the importance of appropriately choosing the link function and how sensitive is the inference if a symmetric link function is incorrectly used in the place of an asymmetric link. The problem appears when the probability of a given binary response approaches 0 at a different rate than it approaches1. Moreover, examples are listed in different textbooks (see, for example, Collet, 2003) reporting situations where an asymmetric link is more appropriate than a symmetric one.

In this case, it is necessary to consider asymmetric links. A very popular example of asymmetric link is the complementary log-log link or *cloglog*, where the cdf of the Gumbel distribution is considered as defined by

$$F(t) = 1 - exp(-exp(t))$$

Where the cdf is completely specified and, it does not depend on any unknown additional parameter and it does not include any particular case as a symmetrical link. This link is considered in *Bayesian Modeling*.

Information of how to implement the Bayesian estimation of the binary regression using the cloglog, probit and logit links in WinBUGS or OpenBUGS can be seeing in the Example Beetles: logistic, probit and carries far estimates models of the Manual. Nevertheless Bayesian approach to binary regression models considering other links as the discussed by Bazán, Bolfarine and Branco (2006 and 2010), Prentice (1976), Nagler (1994), Chen, Dey and Shao (1999, 2001) are not available at the moment.

Asymmetric links considered in *BayesianModeling* are those that are obtained considering other cdf like the following:

$$F(t) = 1 - (1 + e^{t})^{-\lambda} \text{ y } F(t) = (1 + e^{-t})^{-\lambda} \quad \lambda > 0$$

these links are asymmetric logit and are known as *scobit* and *power logit*, respectively, and include the logit link as special case when the parameter $\lambda = 1$. For a review of these links see Prentice (1976) and Nagler (1994).

In *BayesianModeling* also are implemented three links that are based in the cdf of a skew normal distribution (see Azzalini, 1985), this cdf can be represented in general by the following way:

$$F(t; \boldsymbol{\psi}) = 2\Phi_2(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Omega})$$

Where $\boldsymbol{x} = (t, 0)'$; $\boldsymbol{\psi} = (\mu, \sigma^2, \lambda)'$; $\Phi_2(.)$ it represents the distribution accumulated of a normal distribution bivariate with parameters $\boldsymbol{\mu} = (\mu, 0)'$ and $\boldsymbol{\Omega} = \begin{bmatrix} \sigma^2 & -\delta \\ -\delta & 1 \end{bmatrix}$; and $\delta = \dfrac{\lambda}{\sqrt{1 + \lambda^2}}$.

This links were proposed by Bazán, Bolfarine y Branco (2006 and 2010) and as especial cases of this general formulation, the implemented links in the *BayesianModeling* are the following:

- If $\boldsymbol{\psi} = (0, 1 + \lambda^2, -\lambda)$, obtains the asymmetric probit link proposed in Chen et al (1999) named as *CDS skew probit*.
- If $\boldsymbol{\psi} = (0, 1, \lambda)$, obtain the asymmetric probit link proposed by Bazán, Branco and Bolfarine (2006) named as *BBB skew probit*.
- If $\boldsymbol{\psi} = \left(-\frac{\sqrt{2}\delta}{\sqrt{\pi - 2\delta^2}}, \frac{\pi}{\pi - 2\delta^2}, \lambda\right)$, with $\delta = \frac{\lambda}{\sqrt{1+\lambda^2}}$ obtain the standard asymmetric probit link (Bazán, Bolfarine y Branco 2006 y 2010), named here as *Standard skew probit*.

In these three links, λ is the shape parameter that controls the asymmetry, so we have for negative values (positive) of λ has negative asymmetry (positive).

These three models can see also as belonging to the kind of mixes of eliptic distributions proposed by Basu and Mukhopadhyay (2000) given by:

$$F = F(.) = \int_{[0,\infty>} H(. \mid v) dG(v),$$

Where G is the cdf of a variable in $[0, \infty >$ and H is an eliptic distribution. For instance, the CDS skew probit considers a kind of mixes of normal where the measure of mix is the positive normal distribution with density function given by $g(x) = 2\phi(x),\, x > 0$, with $\phi(.)$ being the function of density of the standard normal.

Another interesting case when mixture of the positive normal with H the cumulative distribution function of the logistic distribution it is considered as skew logistic or *skew logit* (see Chen, Dey and Shao, 2001) that also is implemented in *BayesianModeling*.

1.3. Bayesian Estimation in Binary Regression

Considering the distribution Bernoulli for the variable response, the likelihood function is given by

$$L(\beta,\theta,\mid y,X)=\prod_{i=1}^{n}\left\{F_{\theta}\left[x_i'\beta\right]\right\}^{y_i}\left\{1-F_{\theta}\left[x_i'\beta\right]\right\}^{1-y_i}$$

Where $F_{\theta}(.)$ is the cdf of an asymmetric distribution indexed by the shape parameter θ associated with the asymmetric link. For Power Logit and Scobit $\theta=\lambda$. For skew probit and skew logit $\theta=\delta$

The logit, probit, cloglog, scobit and power logit links consider this likelihood function; however skew probit and skew logit links consider other versions of the likelihood function considering augmented versions that are discussed in the specific references of these models.

In the Bayesian Inference, the parameters of interest are assumed like random variables and so is need establishes a priori probability distributions that reflects our previous knowledge of its behavior. Combining the likelihood function and the priori distributions we can obtain the posteriori distributions of the parameters of interest. In the present work, we consider priors that they are vague proper priors with known distributions but variance big as well as independence between priors (see Nzoufras, 2009).

In Binary Regression models there is consensus about the prior specification for regression coefficients, thus is assumed $\beta_j\sim N(0,\sigma^2)$ for $j=1,\ldots,k$ with σ^2 to be large, and this case is considered $\sigma^2=1000.$ In relation with the shape parameter associated with the link: for Scobit and Power logit models is assumed $\lambda\sim gamma(1,1)$ ($E(\lambda)=1,$ $V(\lambda)=1$) and for skew probit models $\delta\sim Unif(-1,1)$, where $-1\leq\delta\leq 1$ and $\lambda=\frac{\delta}{\sqrt{1-\delta^2}}$. In addition is assumed independent priors as

$$\pi(\beta,\theta)=\prod_{j}^{k}\pi_1(\beta_j)\pi_2(\theta_j).$$

With $\pi_1(.),\pi_2(.)$ as indicated above.

Bayesian Inference is facilitated with the use of different MCMC methods implemented in WinBUGS or OpenBUGS software using a minimum programming. For more details about the use of these software for Bayesian Inference and usual Binary regression models we suggest the book of Congdon (2005), Congdon (2010) and Ntzoufras (2009). An introduction to MCMC methods is given in Gilks, Richardson, and Spiegelhalter (1996).

Also, Bayesian Inference for some traditional Binary models considering R packages as `arm`, `bayesm`, `DPpackage`, `LaplacesDemon`, `MCMCpack` are available.

However, for most of the models presented here, there is no program that generates codes for WinBUGS or OpenBUGS with exception of BRMUW (Bazán 2010) a previous version of this program in Spanish. In contrast, BUGS codes for all Binary regression models presented here are facilitated using *BayesianModeling*. The Binary regression models implemented in *BayesianModeling* classified according to its links are:

• Symmetric: probit, logit.

• Asymmetric: cloglog, scobit, power logit, skew logit, skew probit (CDS, BBB and standard).

All codes for Binary regression models are established considering the likelihood function presented here and considering the priors suggested with the exception of the skew logit and skew probit models which use an augmented likelihood function version. In *BayesianModeling*, when a specific code is generated for a Binary regression model, also References to justify the model and the choices of priors are showed.

1.4. Application: Beetles data set

The *BayesianModeling* program generates the necessary syntax for the Bayesian estimation of several models of binary regression, in the WinBUGS program (see Spiegelhalter et al, 1996) or OpenBUGS (Spiegelhalter et al, 2007), using diverse methods MCMC. For this only is necessary to have a text file with the data, generated from any statistics program or from Excel. In the columns normally appear the names of the variables in the first line and in the first column should appear the response variable.

As example consider the group of data Beetles: logistic, probit and extreme value models of the WinBUGS. The group of data is denominated as beetles.txt that is found in this downloads of the program. The variables used in beetles.txt are:

y: 1 if the beetle died after 5 hours of being exposed to carbon disulfide, 0 otherwise x: concentration of carbon disulfide that a beetle was exposed

The data file has the following structure

```
y     x
1     1.6907
1     1.6907
.          .
.          .
.          .
1     1.8839
```

As an application example we consider the following model

$$y_i \sim Bernoulli(\pi_i)$$
$$\pi_i = F\left(\eta_i\right)$$
$$\eta_i = \beta_1 + \beta_2 x_i, \quad i = 1, 2, ..., 481$$

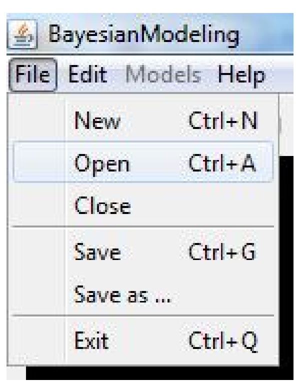
Where F corresponds to the skew logit link (see Chen, Dey and Shao, 2001). More details in Bazán, Bolfarine y Branco (2010)

1.5. Use of *BayesianModeling*

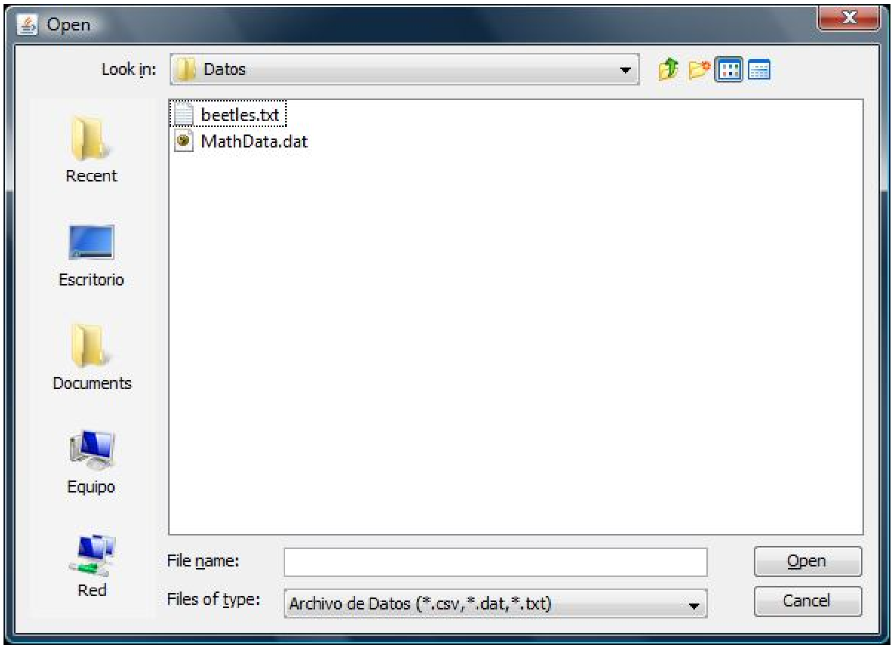
1.5.1 Use of the *BayesianModeling* to generate the syntax of the model

We will use the *BayesianModeling* to implement the model of binary regression with skew logit link for the data beetles.txt described in the previous section. To start using the program you must Open your data file before of choose the models for them.

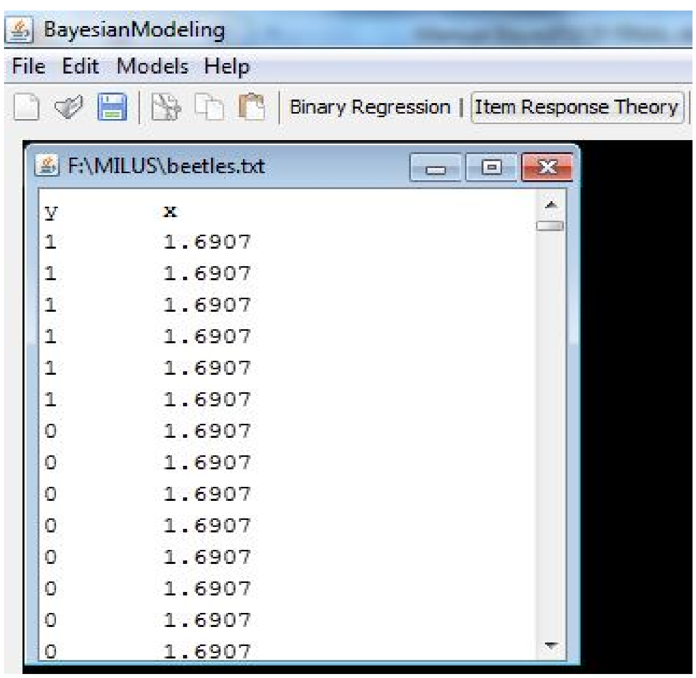
1. Go to *File* > *Open*



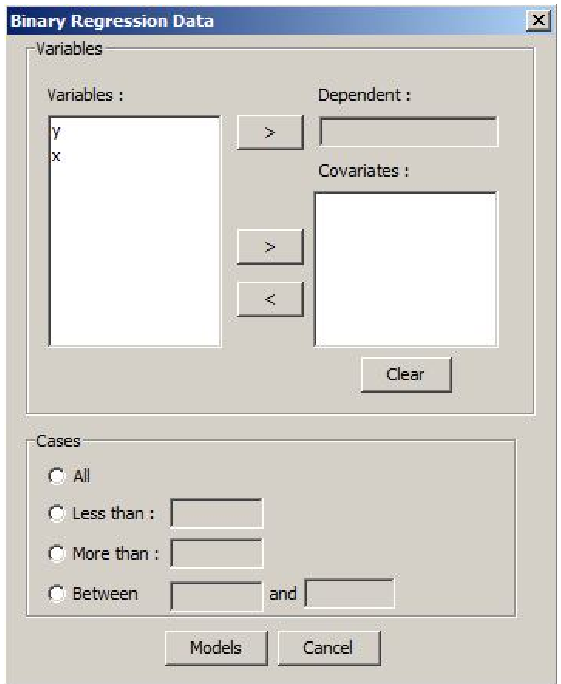
2. Navigate to the directory that contains your dataset files. Select the data set file you want work. The program can open data file in ASCII format (.csv, .txt and .dat)



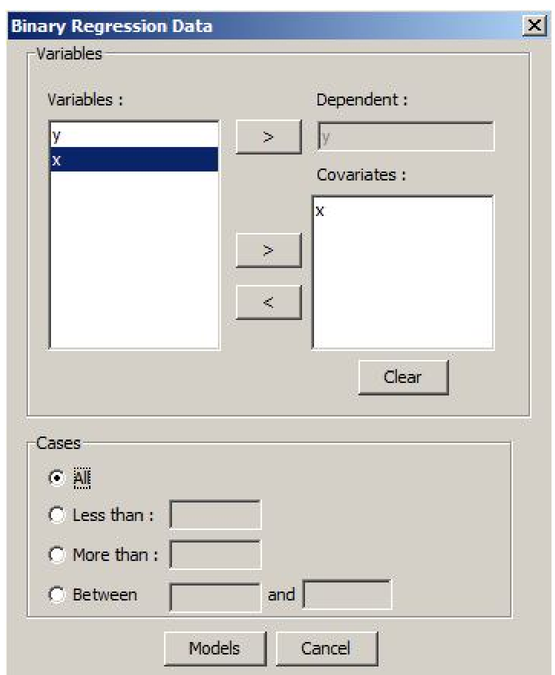
3. To select the model you want Click on Binary Regression button or Go to *Models > Binary Regression*



4. This will open the dialog box "*Binary Regression Data*".

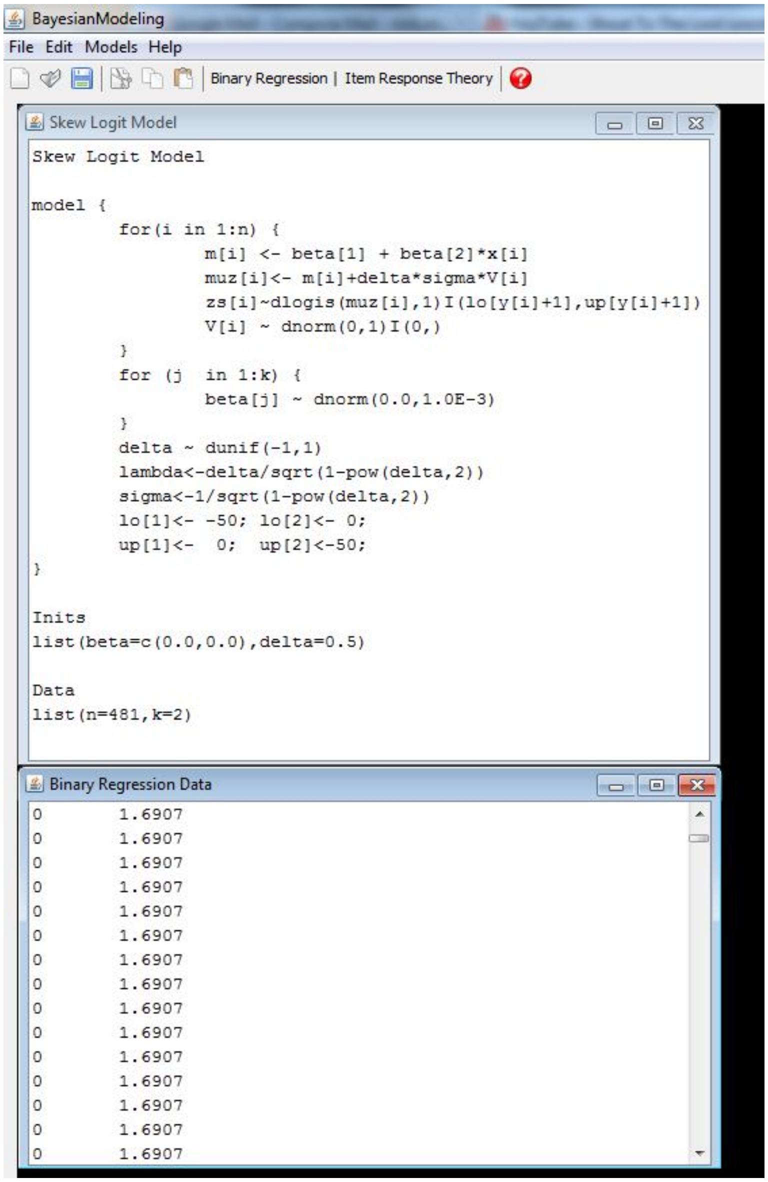


5. Afterward, select the *dependent* variable and *covariates* variable you want. In this case y and x respectively. Click and drag the variables. Also, you must to indicate if will use all the data or only a part of them considering the options in *Cases*. In the example, select *All*.



6. Then, click *Models*. This will open the dialog box “*Binary Regression Models*”. Here you must to select the models that will be used, in this example only select the model skew-logit and do click in OK.

7. Two type of files are generated, corresponding for the selected models and for data. In this case *Skew Logit Model and Binary Regression Data.* Both files are readable in WinBUGS or OpenBUGS.



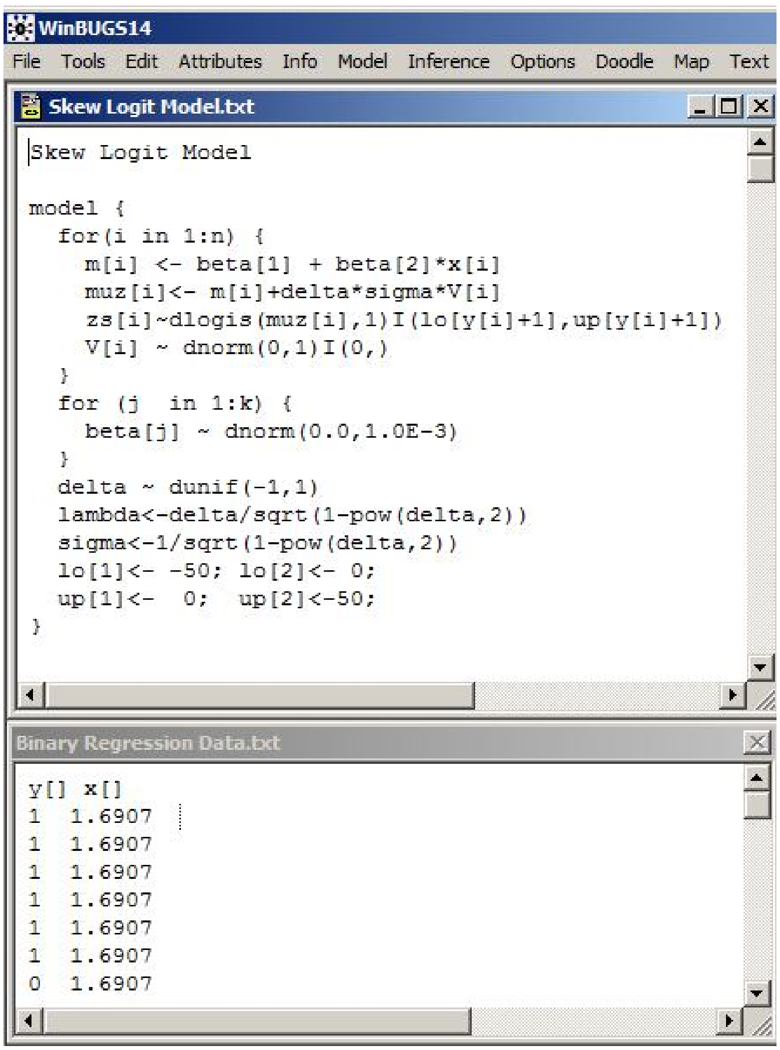
These two files will have to be saved with format *txt*, is to say *Binary Regression Data.txt* and *Skew Logit Model.txt* for its subsequent use.

To generate new models for other data is recommended to restart the program and follow the steps presented.

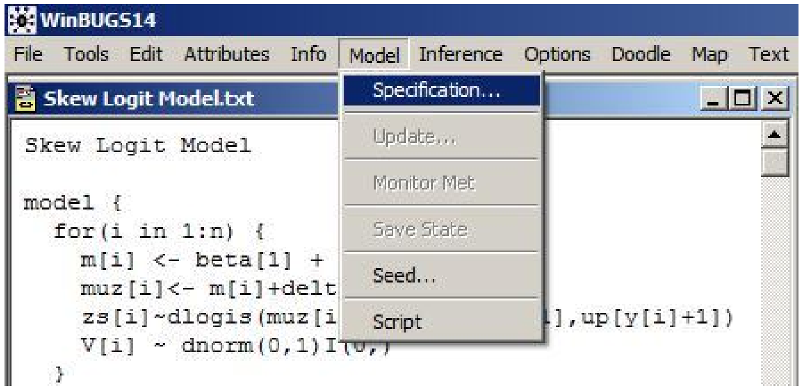
1.5.2 Bayesian estimation using WinBUGS or OpenBUGS

As we have seen the *BayesianModeling* generates two files, one that contains the model of binary regression with the link selected and another that contains the data set. Both files with txt format have to be opened in the WinBUGS or OpenBUGS program to make the correspondent analysis of inference. Here we detail the steps that must be followed to perform the Bayesian inference in WinBUGS. For details, see Chapter 4 of Nzoufras (2009)

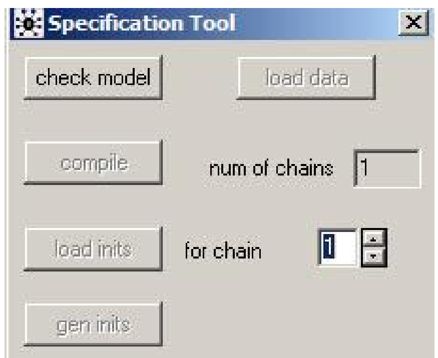
1. Open the files with the syntax of the model and the previous generated data by the *BayesianModeling* in WinBUGS or OpenBUGS.



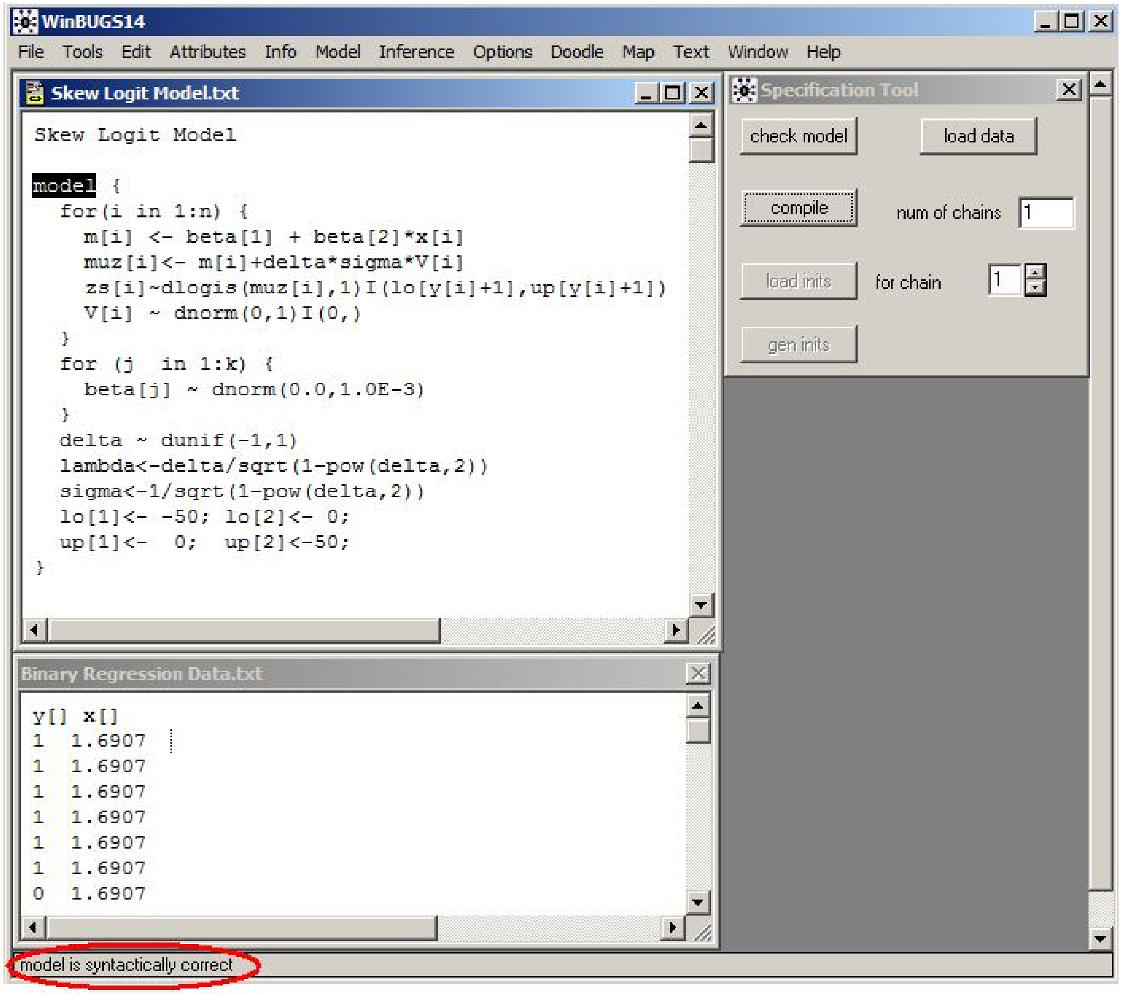
2. Click in *Model* > *Specification* having active the window of the file Skew Logit Model.txt.



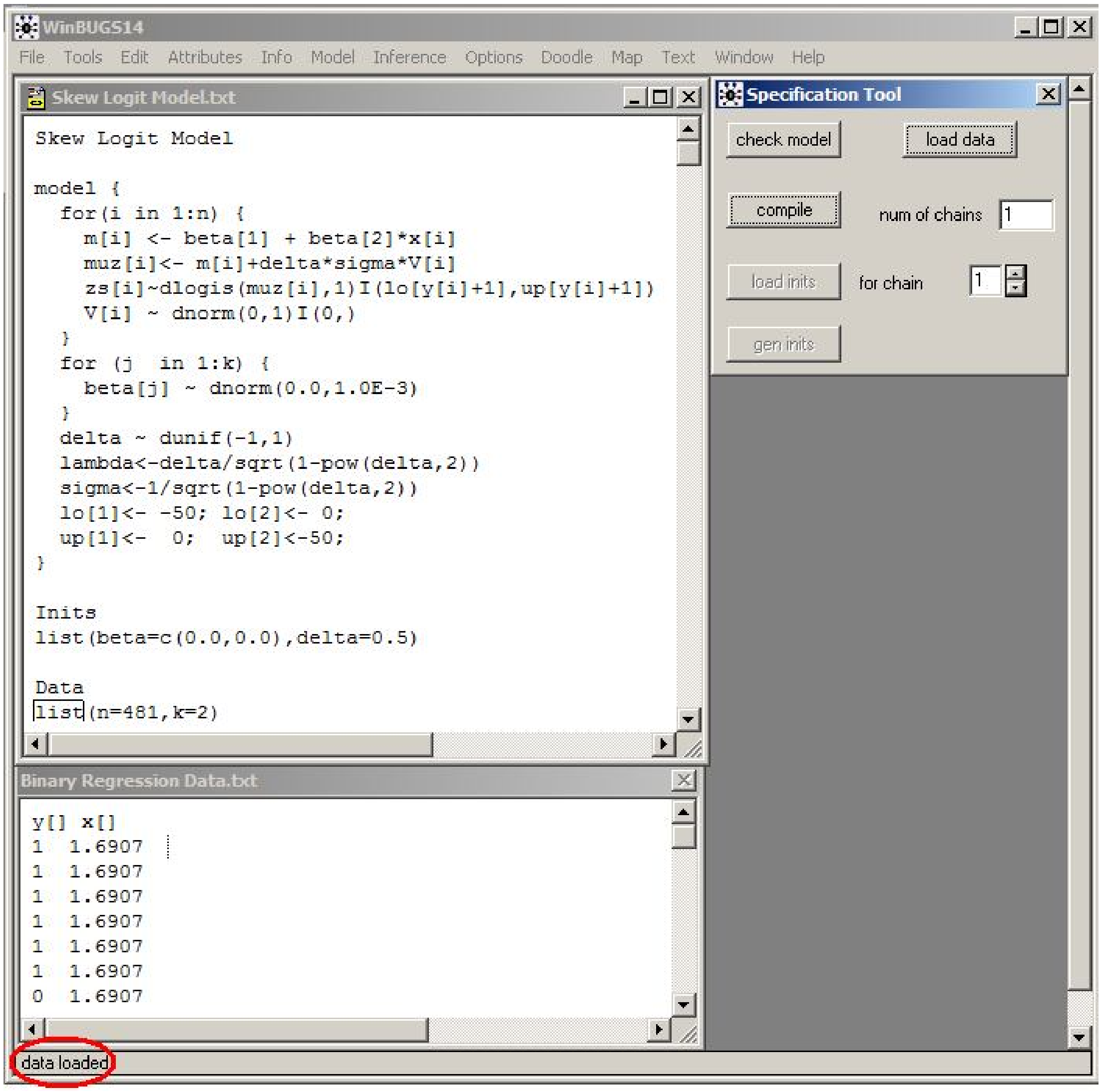
3. This will open the dialogue box "*Specification Tool*".



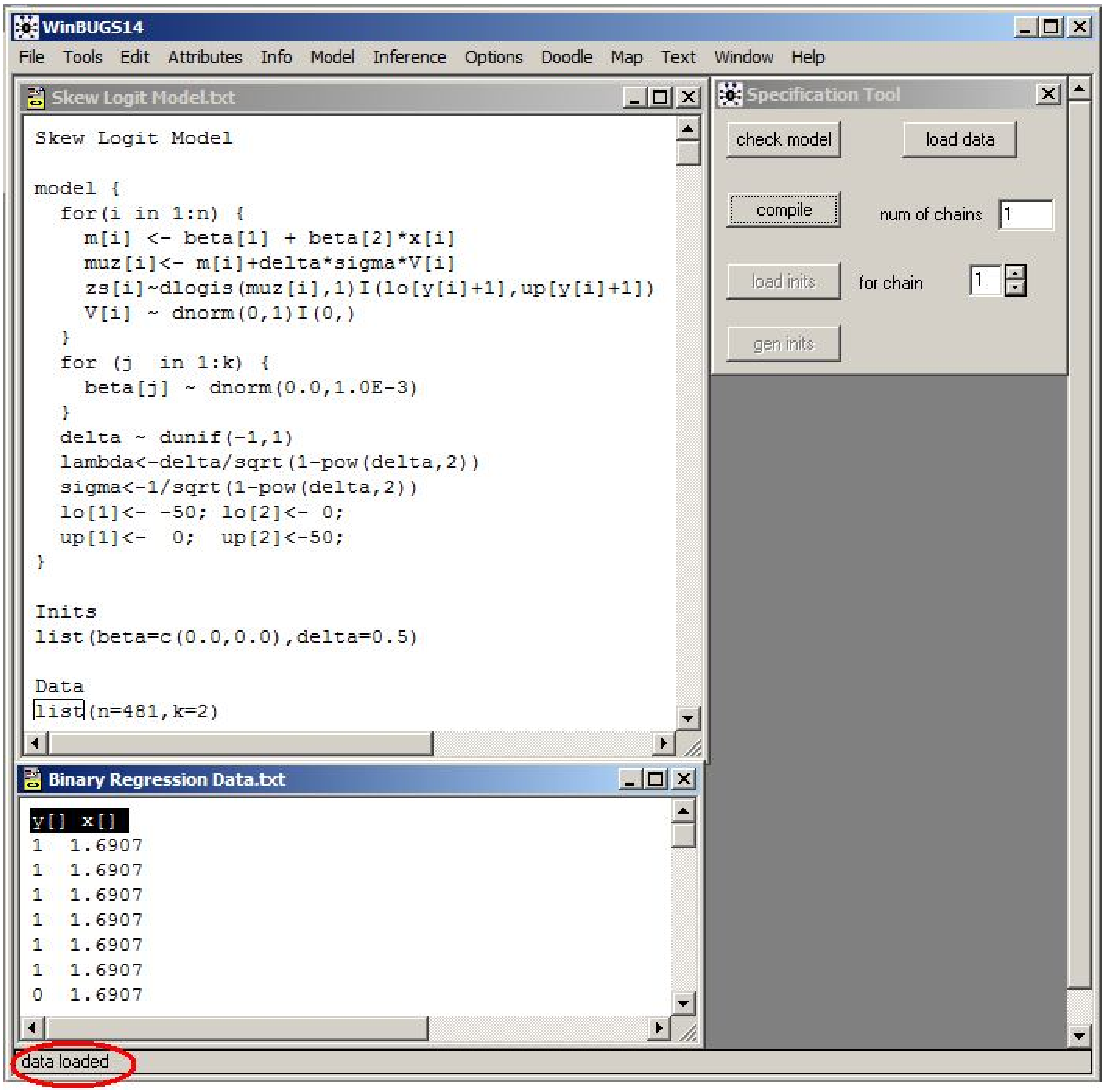
4. Select the word *model* highlighting it with the cursor and do click in *check model*. In the below left corner has to appear "*model is syntactically correct*" that indicates the syntax of the model is properly formulated.



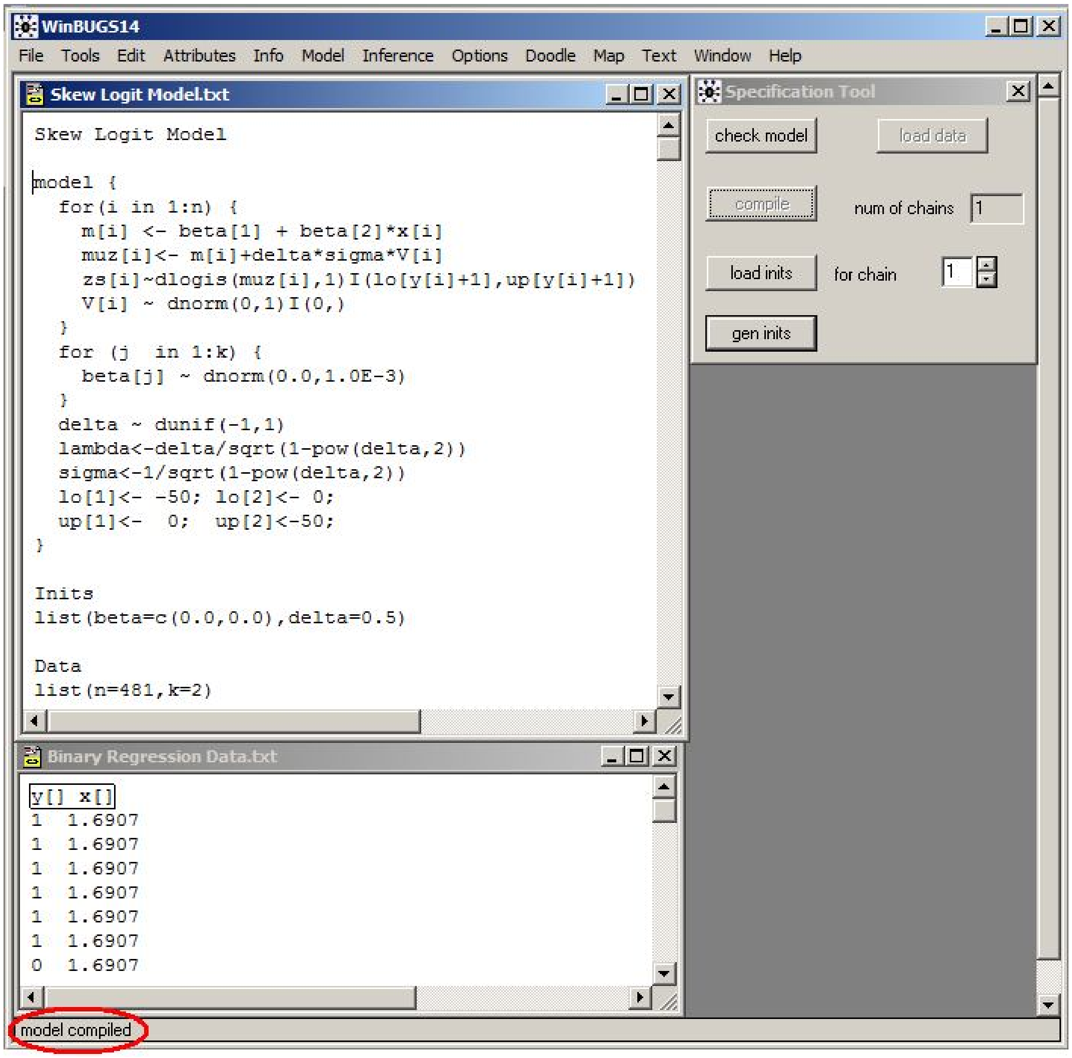
5. Select in the file of the model the line under Data, *list* and do click in *load data*. In the below left corner appears "*data loaded*" indicating that the data have been read.



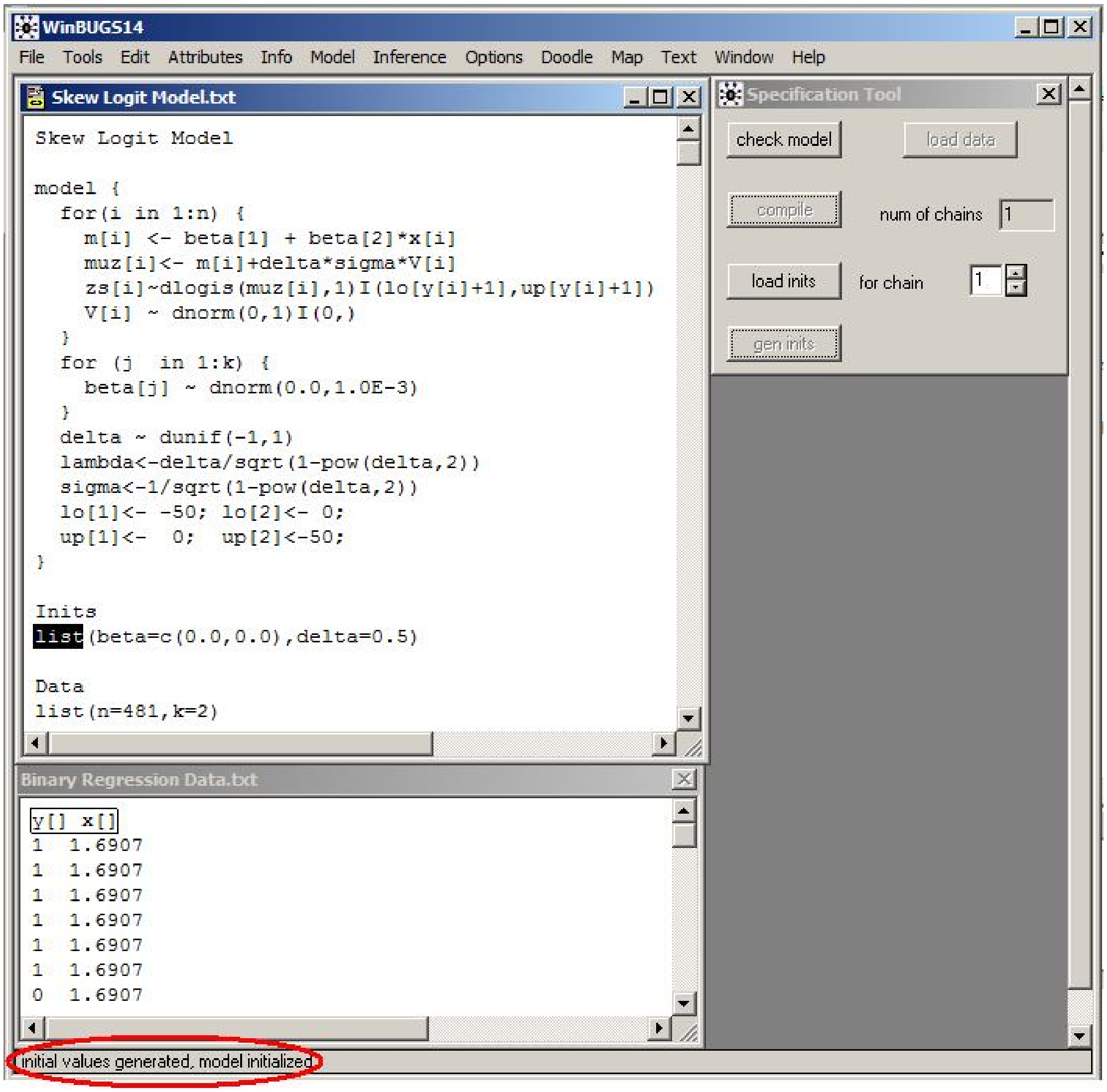
6. In the data file Binary Regression Data.txt select the names of the variables x and y and do click in *load data*.



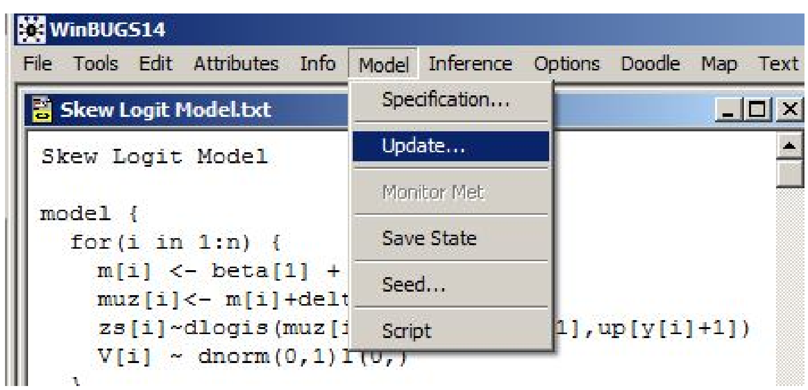
7. The dialogue box "*Specification tool*" specifies the number of chains that want to generate in the box of text "*num of chains*". Once specified the number of chains to generate (in this example 1 chain) do click in *compile*. In the left corner below has to appear "*model compiled*".



8. Select the line under *Inits* in the file of the model and do click in *load inits*. Then do click in *gen inits*. This generates the initial values for the Bayesian Estimation. In the left corner below has to appear "*initial values generated, model initialized*".



9. Do click in *Model* > *Update*



10. This will open the dialogue box "*Update tool*". In the text box *Update Tool* type the number of iterations that are required and afterwards does click in *update.*

11. Then, you must to specify the parameters to be monitored, for this go to *Inference > Samples*, which will open the dialogue box "*Sample monitor tool*". In the text box *node* type the name of the parameters and then do click in *set*, this has to be done for each parameter. In the example is beta and delta.

12. Repeat the step 10 as necessary to generate more iterations. In addition, by considering the dialogue box "*Sample Monitor Tool*", you can: calculate posterior statistics of the parameters doing click in *stats* button, obtain a trace plots of the chains doing click in *history* button, an estimation of the posterior density doing click in *density* button and other more statistics of the chains can be calculated using this dialogue box. If you want to save the actual values for further analysis, click on "coda" on the Sample Monitor Tool.

1.5.3 Bayesian Estimation using WinBUGS or OPENBUGS in R

As we have seen in the previous section with the two files that generates the *BayesianModeling* can implement the Bayesian Estimation using WinBUGS or

OpenBugs programs. Also Bayesian estimation can be implemented by using R2WinBUGS or R2OpenBUGS (Sturtz, Ligges and Gelman, 2005), packages for Running WinBUGS and OpeBUGS from R respectively or BRugs (Thomas, et al, 2006) a collection of R functions that allow users to analyze graphical models using MCMC techniques.

Here we will need the file Beatles.txt and the skewlogitModel.txt syntax of the model generated in the *BayesianModeling*.

We copy all the syntax before "*Inits*" and save it in a file, by example as *modelbr.txt*. Then the file *modelbr.txt* would remain as

```
model {
 for(i in 1:n) {
      m[i] <- beta[1] + beta[2]*x[i]
      muz[i]<- m[i]+delta*sigma*V[i]
      zs[i]~dlogis(muz[i],1)I(lo[y[i]+1],up[y[i]+1])
      V[i] ~ dnorm(0,1)I(0,)
 }
 for (j  in 1:k) {
      beta[j] ~ dnorm(0.0,1.0E-3)
 }
 delta ~ dunif(-1,1)
 lambda<-delta/sqrt(1-pow(delta,2))
 sigma<-1/sqrt(1-pow(delta,2))
 lo[1]<- -50; lo[2]<- 0;
 up[1]<-  0;  up[2]<-50;
}
```

To implement the Bayesian estimation in R we follow the following steps to use the library R2WinBUGS.

1. In R, load the library R2WinBUGS, installed previously, with the following commando:

```
library(R2WinBUGS)
```

2. Read the data (the file beetles.txt for this Example is in F:\MILUS\beetles.txt)

```
datos <- read.table("F:/MILUS/beetles.txt", header=TRUE, sep="",
na.strings="NA", dec=".",strip.white=TRUE)
```

3. Create a list that contain the data and the information of the size of the data set typing:

```
n=nrow(datos)

k=ncol(datos)

data<-c(as.list(datos),list(n=n,k=k))
```

4. Create a program that generates the initials values typing:

```
inits<-function(){list(beta=rep(0,k),delta=0.5)}
```

5. Finally with the command *bugs* implements the Bayesian estimation. Here will explain in brief the syntax of the command Bugs

`parameters.to.save =` is a vector with the names of the parameters of the model which simulations we wish to save.

`model.file =` is the direction where finds the file of the model

`n.chains =` is the number of chains to be generated.

`n.iter =` is the number of the total iterations of each chain.

`n.burnin =` is the number of iterations to be discharged as burn-in.

`program =` is the program that will be used to implement the Bayesian estimation

With the following command the Bayesian estimation is implemented and the simulations are saved in the object `out`.

```
out<-bugs(data,inits,parameters.to.save=c("beta","lambda"),

model.file="F:/MILUS/modelbr.txt", n.chains=1, n.iter=44000,

n.burnin=4000,program="WinBUGS")
```

6. If we type out in the line of commands of the R a summary of the simulation is obtained

```
> out
Inference for Bugs model at "G:/MILUS/modelbr.txt", fit using WinBUGS,
 1 chains, each with 44000 iterations (first 4000 discarded), n.thin =
40
 n.sims = 1000 iterations saved
           mean   sd  2.5%    25%    50%    75%  97.5%
beta[1]   -62.2  3.1 -67.8  -64.4  -62.3  -60.1  -55.9
beta[2]    35.0  1.6  31.4   33.9   35.2   36.2   38.0
lambda      0.2  0.8  -1.2   -0.3    0.1    0.7    1.7
deviance 1029.3 15.7 986.7 1024.7 1035.0 1040.0 1044.0

DIC info (using the rule, pD = Dbar-Dhat)
pD = -9.3 and DIC = 1020.0
DIC is an estimate of expected predictive error (lower deviance is
better).
```

7. Finally, to greater details about the command *bugs* you can obtain help typing in the line of the commands the following

```
?bugs
```

Note. You can specify Bug.directory. The directory that contains the **WinBUGS** executable. If the global option R2WinBUGS.bugs.directory is not NULL, it will be used as the default. Also you can specify the program to use, either winbugs/WinBUGS or openbugs/OpenBUGS, the latter makes use of function openbugs and requires the CRAN package BRugs.

2. Item Response Theory

2. 1. Item Response Theory models

Consider data collected of n persons who have each given responses on k different items of a test. A *Two-Parameter* Item Response Theory (IRT) model one-dimensional and binary is a system in which for each person i has a unidimensional monotone latent variable model $(\mathbf{Y}, U_i)$, defined by the following expressions:

$$Y_{ij}|u_i, \eta_j \sim Bernoulli(p_{ij})$$
$$p_{ij} = P(Y_{ij} = 1 \mid \theta_i, \eta_j) = F(m_{ij})$$
$$m_{ij} = a_j(\theta_i - b_j),$$

$i = 1\ldots, n,\ j = 1, \ldots, k$

where

- Y_{ij} is the manifest variable which model the binary response of the person i that answer to the item j. The items have binary outcomes, i.e., the items are scored as 1 if correct and 0 if no.
- $\eta_j = (a_j, b_j)$ are two parameters that represent, respectively, to the discrimination and the difficulty of the item j.
- θ_i is the value of the latent variable or trait latent Θ_i for the person i, and some occasions it is interpreted as the latent ability of the person i.
- p_{ij} is the conditional probability given $\Theta_i = \theta_i$ to respond correctly *to* item j .
- F *is called* the *item characteristic curve* (ICC) and
- m_{ij} is a latent lineal predictor associated with the latent trait of the person i and the item parameters for the item j.

Observations

The *Two-Parameter* IRT model

- satisfies the property of latent conditional independence; it is, for a person i the response Y_{ij} to the different items are conditionally independent given the latent variable Θ_i, $i = 1\ldots, n$
- satisfies the property of latent monotonicity, because is a function strictly no decreasing of θ_i, $i = 1\ldots, n$
- is one-dimensional latent.

- $F(m_{ij})$, where $i = 1 \ldots, n$ and $j = 1, \ldots, k$, is the same for each case and $F^{-1}(.)$ is called the link function.
- Also is assumed that responses are independent between persons.
- The parameters of difficulty b_j and of discrimination a_j represent the location and inclination of the item, respectively, being a_j a proportional value to the inclination of the ICC in the point b_j and b_j is the point on θ_i where the ICC has its maximum slope. Values as $a_j < 0$ are not expected. The parametric space for the parameter b_j is arbitrary and to be the same as θ_i than by the usual take values in the line real.

Another parameterization very common for the predictor linear latent is $m_{ij} = a_j\theta_i - b_j$. This parametrización is very important from the computational point of view since it facilitates the computational time of convergence. When it used this parameterization, the previous parameter of difficulty can be obtained doing $\frac{b_j}{a_j}$ in the obtained result. Generally, this parameterization is preferred in the Bayesian Inference and also in *BayesianModeling*.

In The *Two-Parameter* IRT model, the conjoint density of the vector of multivariate responses $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n)'$, with $\boldsymbol{Y_i} = (Y_{i1}, \ldots, Y_{ik})$ given the vector of latent variables $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_n)'$ and the vector of parameters of the items $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_k)'$ can be written as:

$$f(\boldsymbol{Y}|\boldsymbol{\Theta}, \boldsymbol{\eta}) = \prod_{i=i}^{n} \prod_{j=1}^{k} F(m_{ij})^{y_{ij}} (1 - F(m_{ij}))^{1-y_{ij}}$$

The proof of this result is direct by considering the latent conditional independence.

The first IRT binary model was introduced by Lord (1952) with an ICC given by $F(x) = \Phi(x)$, being Φ the cdf of a standard normal variable. This model is known in the psychometric literature as *normal ogive* model which corresponds in the context of Generalize Linear Models, for a probit link function and empathizing this can be named as *2P* model.

On the other hand, Birbaum (1968) considered a ICC given by $F(x) = L(x)$, where $L(x) = \dfrac{e^x}{1+e^x}$ denotes the cdf of a logistic variable. This induces, in the language of the generalized linear models, to a logit link function. This model is known as the *logistic* model and empathizing the link is named here as *2L* model.

Particular cases

The IRT model admits diverse formulations, which depend basically of as it considers the ICC. In its simplest version could take $a_j = 1$ and consider an ICC of the form

$$P(Y_{ij} = 1 \mid \theta_i, b_j) = F(\theta_i - b_j).$$

This is called of *one-parameter* IRT model and when probit or logit links are considered we have 1P or 1L IRT model respectively.

In a general way we could consider an ICC of the form

$$P(Y_{ij} = 1 \mid u_i, a_j, b_j, c_j) = c_j + (1 - c_j)F(a_j(\theta_i - b_j)),$$

Where the parameter $c_j \in [0,1]$ indicates the probability that very low ability individuals will get this item j correct by chance, and F is the distribution function. This is known as the *three-parameter* IRT model. If $c_j = 0$, the model is reduced to the two- parameters IRT model. Again, when probit or logit links are considered we have 3P or 3L IRT model respectively.

The IRT model with logit link

The IRT model with logit link or logistic model is probably the model more used in IRT. The version of three parameters for this model establishes that the probability that the person i hit the item j is given by:

$$P(Y_{ij} = 1 | \theta_i, a_j, b_j, c_j) = c_j + (1 - c_j)\frac{1}{1 + e^{Da_j(\theta_i - b_j)}},$$

where usually is assumed that $D=-1$ although some authors consider also the value $D=-1.7$ for approximating this model to the normal ogive model. As particular cases have

- $P(Y_{ij}=1|\theta_i,a_j,b_j)=\dfrac{1}{1+e^{Da_j(\theta_i-b_j)}}$ (with $c_j=0$) and
- $P(Y_{ij}=1|\theta_i,b_j)=\dfrac{1}{1+e^{D(\theta_i-b_j)}}$ (with $c_j=0$ y $a_j=1$).

The last model of a parameter, is knows as well as the Rasch model but it has own interpretations and derivations (see for example Fischer and Molenaar, 1995).

The *BayesianModeling* program allows implementing the code in WinBUGS for the models 1L, 2L, 3L, 1P, 2P and 3P IRT models. By considering this links, these models are symmetric IRT models. In addition, news IRT models with asymmetric links are considered also in *BayesianModeling* which are presented in the next section.

2.2. IRT Models with asymmetric links

In the traditional IRT models, the asymmetric ICC are considered symmetrical; this is the case of the logistic and of normal ogive models. However, as it has observed Samejima (2000), Bazán et al (2006) and Bolfarine and Bazán (2010) asymmetric ICC can be incorporated considering a new parameter of item that controls the shape of the curve. This asymmetry is necessary in many cases for a better modelization of answers with a low proportion of 0´s or 1's. Then will show three of these models.

The skew normal ogive model

The *skew normal ogive* model was proposed by Bazán et al (2006) assuming that the probability of success considering the abilities and the item parameters associates it given by:

$$p_{ij}=P(Y_{ij}=1\mid\theta_i,\eta_j)=\Phi_{SN}(m_{ij};\lambda_j),\,i=1\ldots,n,\;j=1,\ldots,k$$

where $\lambda_j > 0$ is a parameter of asymmetry, $m_{ij} = a_j\theta_i - b_j$ is the latent linear predictor and Φ_{SN} denote the cdf of a skew normal distributions with function of density $\phi_{SN}(x;\lambda) = 2\Phi(x)\Phi(\lambda x)$, being ϕ the pdf of a standard normal variable.

Notice that if $\lambda = 0$, the normal ogive model (2P) is obtained, but as indicated in Bazán et al (2006) if $\lambda > 0$, the probability of correct response has a slow growth for low values of latent variable Θ. On the other hand, if $\lambda < 0$, the probability of correct response has a quick growth for low values of the latent variable Θ. Is because this behavior that this parameter is interpreted as a penalization parameter for item. Main details about this model can be reviewed in Bazán et al (2006). In this formulation the link considered is the BBB skew probit link (see Bazán, Bolfarine and Branco, 2010) and for this reason the model can be named also *two-parameter skew probit* or *2SP* IRT model. When only difficulty parameter is considered we have the *1SP* IRT model.

The LPE and RLPE models

Logistic positive exponent (LPE) was proposed by Samejima (2000). A reversal version, named *Reflection Logistic positive exponent* (RLPE) was formulated by Bolfarine and Bazán (2010). These models, studied in Bolfarine and Bazán (2010), assume that the probability of correct response considering the abilities and the item parameters associates it given by

$$p_{ij} = P(Y_{ij} = 1 \,|\, \theta_i \,,\, a_j \;\; b_j) = F_{\lambda_j}(m_{ij}),$$

where F_{λ_j} is the cdf of the logistic distribution indexed by the parameter $\lambda_j > 0$ and evaluated in m_{ij}. For LPE model $F_{\lambda_j} = L(m_{ij})^{\lambda_j}$ and for RLPE model $F_{\lambda_j} = 1 - L(-m_{ij})^{\lambda_j}$. Depending that function of distribution specifies will have *2LPE* or *2RLPE* IRT models. In the first case, this characterize by

$$F_1(m_{ij}) = 1 - (1 + e^{m_{ij}})^{-\lambda_j}$$

And the second case by:

$$F_2(m_{ij}) = (1 + e^{-m_{ij}})^{-\lambda_j}$$

These correspond to the cdf of the Scobit distribution and Burr of type II, respectively.

Note that $F_1(-m_{ij}) \neq 1 - F_1(m_{ij})$ or $F_2(-m_{ij}) \neq 1 - F_2(m_{ij})$ and F_1 y F_2 are asymmetric but it holds that $F_2(m_{ij}) = 1 - F_1(-m_{ij})$ or $F_1(m_{ij}) = 1 - F_2(-m_{ij})$.

In both models λ_j can also interpret like a parameter of penalty or bonus of similar way to the case of the model of skew normal ogive model given by Bazán, Branco and Bolfarine (2006). More details of this model can review in Bolfarine y Bazán (2010). Particular cases and extensions considering one-parameter or three-parameters are possible. Thus, *1LPE*, *1RLPE*, *3LPE* and *3RLPE* are another IRT models implemented in *BayesianModeling*.

2.3. Bayesian estimation in IRT

Considering the distribution Bernoulli for the response variable, the likelihood function for IRT model in the three-parameter IRT model is given by

$$L(\mathbf{a}, \mathbf{b}, \mathbf{c}, \lambda, \theta, \mid \mathbf{y}) = \prod_{i=1}^{n} \prod_{j=1}^{k} \{F_\lambda\left[a_j(\theta_i - b_j); c_j\right]\}^{y_i} \{1 - F_\lambda\left[a_j(\theta_i - b_j); c_j\right]\}^{1-y_i}$$

Where $F_\lambda\left[a_j(\theta_i - b_j); c_j\right]$ is the cdf of an asymmetric distribution indexed by the parameter λ, associated with the asymmetric ICC.

The Logistic (logit), Normal ogive (probit), LPE and RLPE consider this likelihood function; however skew probit IRT consider other version of the likelihood function considering augmented version that is discussed in the specific references of this model. In WinBUGS, the implementation of this procedure is not direct because it requires of a correct specification of the indicator variables. Main details can find in Bazán, Branco and Bolfarine (2006).

In the Bayesian Inference, the parameters of interest are assumed like random variables and so is need establishes a priori probability distributions that reflects our previous knowledge of its behavior. Combining the likelihood function and the priori distributions we can obtain the posteriori distributions of the parameters of interest. In the present work, we consider priors that they are vague proper priors with known distributions but variance big as well as independence between priors (see Nzoufras, 2009). In traditional

IRT models priors are discussed in Albert (1992), Johnson and Albert (1999), Patz and Junker (1999), Sahu (2002), Rupp, Dey and Zumbo (2004), Bazán, Bolfarine and Leandro (2006), Fox (2010).

In IRT models there is consensus about the prior specification for latent trait, thus is assumed $\theta_i \sim N(0,1)$ for $i = 1, \ldots, n$. However about item parameter there is several proposals. Here is assumed independent priors as

$$\pi(\theta, \mathbf{a}, \mathbf{b}, \mathbf{c}, \lambda) = \prod_{i}^{n} \phi(\theta_i) \prod_{j}^{k} \pi_1(a_j)\pi_2(b_j)\pi_3(c_j)\pi_4(\lambda_j).$$

Where $\phi(.)$ is a pdf of normal distribution and $\pi_1(.)$, $\pi_2(.)$, $\pi_3(.)$, $\pi_4(.)$ correspond to the prior distributions of item parameters $a_j,\ b_j,\ c_j$ and λ_j, respectively.

In the special case of size of small samples we suggested the use of the following prior specification

- $a_j \sim HN(\mu_a, \sigma_a^2)$ with $\mu_a = 1$ and $\sigma_a^2 = 0.5$ ($E(a_j) = 1.1126$ and $V(a_j) = 0.3747$) where $HN(.)$ correspond the positive normal or Half-normal distribution.
- $b_j \sim N(\mu_b, \sigma_b^2)$ with $\mu_b = 0$ and $\sigma_b^2 = 2$, ($E(b_j) = 0$ and $V(b_j) = 2$)
- $c_j \sim Beta(5, 17)$ ($E(c_j) = 0.227$, $V(c_j) = 0.0076$).
- For LPE and RLPE models $\lambda_j \sim gamma(0.25, 0.25)$ ($E(\lambda_j) = 1$, $V(\lambda_j) = 4$)
- For Ogive skew normal model $\delta_j \sim Unif(-1,1)$, where $-1 \leq \delta_j \leq 1$ and $\lambda_j = \frac{\delta_j}{\sqrt{1-\delta_j^2}}$.

Bayesian Inference in IRT models is facilitated with the use of different methods MCMC implemented in WinBUGS or OpenBUGS software. An introduction to MCMC methods is given in Gilks, Richardson, and Spiegelhalter (1996). For more details about the use of these softwares for Bayesian Inference we suggest the book of Congdon (2005), Congdon (2010) and Ntzoufras (2009). For traditional IRT models Bugs codes are available by example in Curtis (2010), Fox (2010), Bazan, Valdivieso and Calderón (2010).

Also, Bayesian Inference for some traditional IRT models using `R` package (`MCMCpack`: Martin, Quinn and Park, 2011) and `Matlab` package: (`IRTuno`: Sheng, 2008a, `IRTmu2no`: Sheng, 2008b, `IRTm2noHA`: Sheng, 2010) are available.

However, for most of the models presented here, there is no program that generates codes for WinBUGS or OpenBUGS. This is facilitated using *BayesianModeling*. The IRT models implemented in *BayesianModeling* classified according to its links are:

• Symmetric: logistic (1L, 2L, 3L), probit (1P, 2P, 3P).

• Asymmetric: LPE (1LPE, 2LPE, 3LPE), RLPE (1RLPE, 2RLPE, 3RLPE), skew probit (1SP, 2SP).

All codes for IRT models are established considering the likelihood function presented here and considering the priors suggested with the exception of the skew probit IRT models which use an augmented likelihood function version. In *BayesianModeling*, when a specific code is generated for a IRT model, also References to justify the model and the choices of priors are showed.

2.4. Application: Math data set

The program *BayesianModeling* generates the syntaxes necessary for the Bayesian estimation of several models of the Item Response Theory, for posterior use in WinBUGS (see Spiegelhalter et al 1996) or OpenBUGS (Spiegelhalter et al 2007) program, using diverse MCMC methods. For this only is necessary to have a file of text with the data, generated from any statistics program or from Excel. In each column, usually appear the names of the items in the first line.

As an example, consider a data set of 14 items from a Mathematical test developed by the Unity of Measurement of the Educative Quality of Peru for the National Evaluation of the sixth degree of 1998 which were applied to a sample of 131 students of sixth degree of high socioeconomic level. These data have been used in Bazán, Branco and Bolfarine (2006) and Bazán, Bolfarine and Leandro (2006).

The released items are a sampling from a test that appears published in the following link:

http://www2.minedu.gob.pe/umc/admin/images/publicaciones/boletines/Boletin-13.pdf

In the table appears the identification corresponding to the number of item with the number in the UMEQ test.

Number of item of Math data	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Number of item in the UMEQ test	1	8	9	11	12	13	21	25	32	5	17	30	2	10

The data file can be found in the file zip of the program and has the following structure:

```
I01  I02  I03  I04  I05  I06  I07  ...  I12  I13  I14
1    1    0    1    1    0    1    ...  0    0    1
1    1    1    1    1    1    1    ...  0    1    1
.    .    .    .    .    .    .    .    .    .    .
.    .    .    .    .    .    .     .   .    .    .
.    .    .    .    .    .    .      .  .    .    .
1    1    0    1    0    0    1    ...  0    1    1
```

As an example of application we consider an IRT model with asymmetric link, in this case we consider the skew normal ogive model with parameters of difficulty and of discrimination, this is a two-parameter skew probit IRT model (2SP)

$$Y_{ij}|\theta_i,\eta_j \sim Bernoulli(p_{ij})$$
$$p_{ij} = P(Y_{ij} = 1 \mid \theta_i,\eta_j) = \Phi_{SN}(m_{ij},\lambda_j)$$
$$m_{ij} = a_j\theta_i - b_j,$$

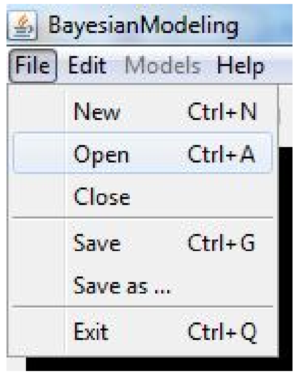
$i = 1\ldots,131,\ j = 1,\ldots,14$, where $\lambda_j > 0$ is a parameter of penalty and Φ_{SN} denote the skew normal cdf.

2.5. Use of the BayesianModeling

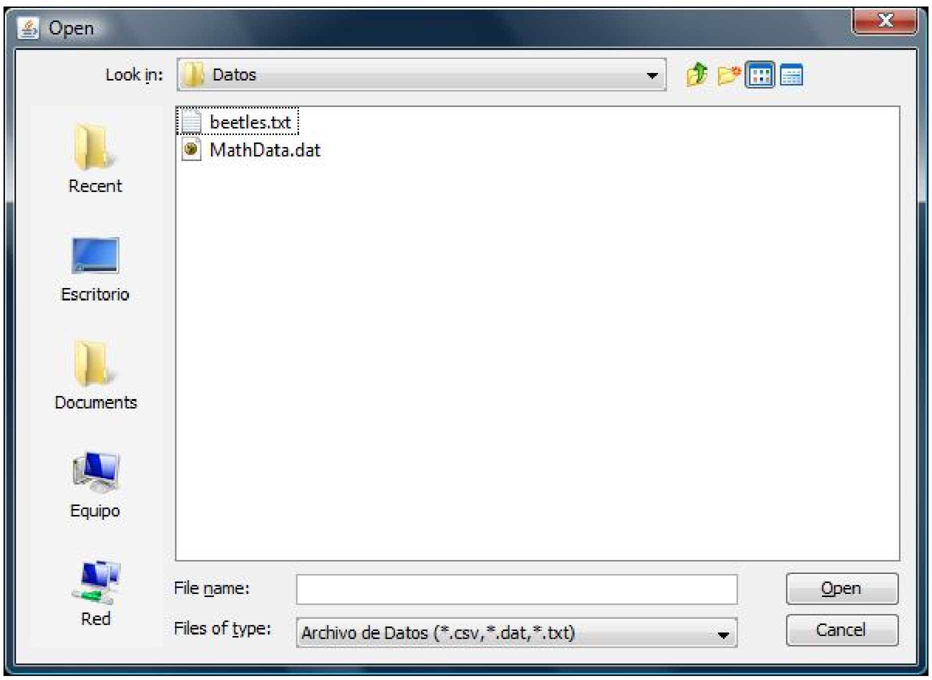
We described the use of the *BayesianModeling* to implements the 2SP IRT model to the data of MathData.dat described in the previous section. For more details of this application, review Bazán, Branco and Bolfarine (2006).

2.5.1 Generate the syntax of the model

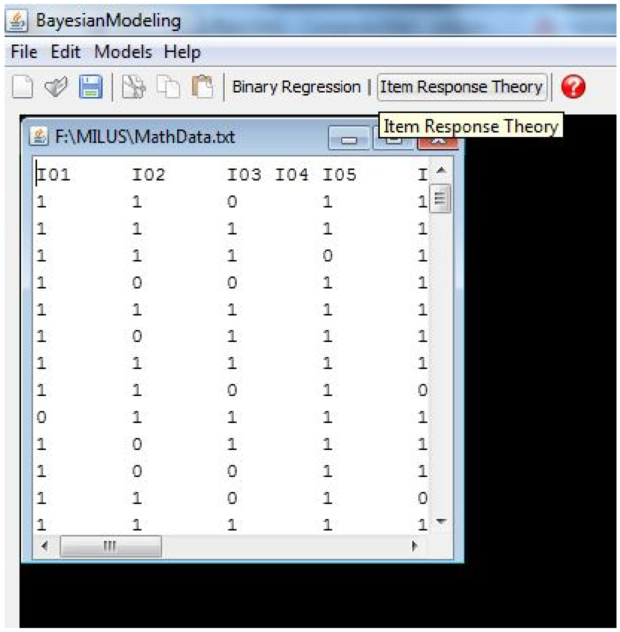
1. Go to *File* > *Open*



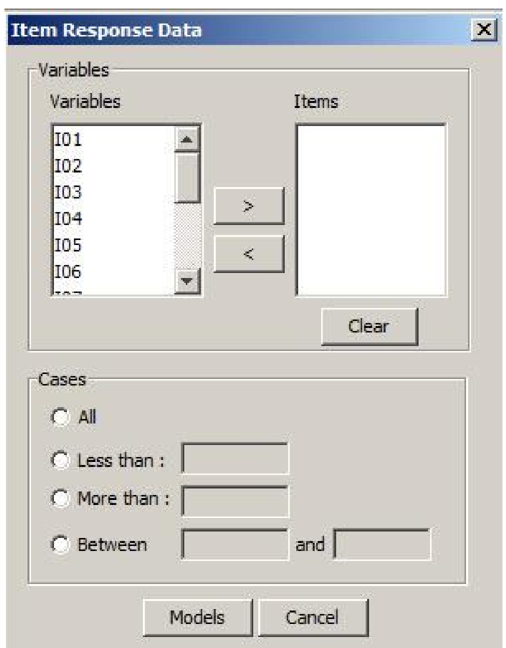
2. Open the file with the data set.



3. Click *Item Response Theory*

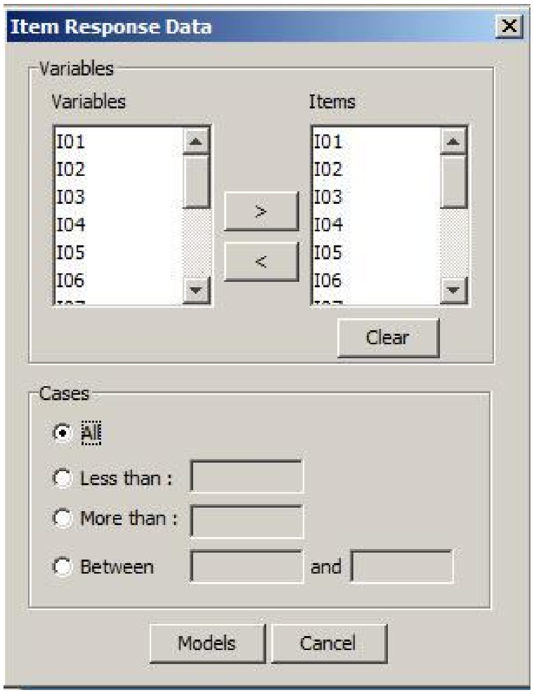


4. This will open the dialogue box “*Item Response Data*”.

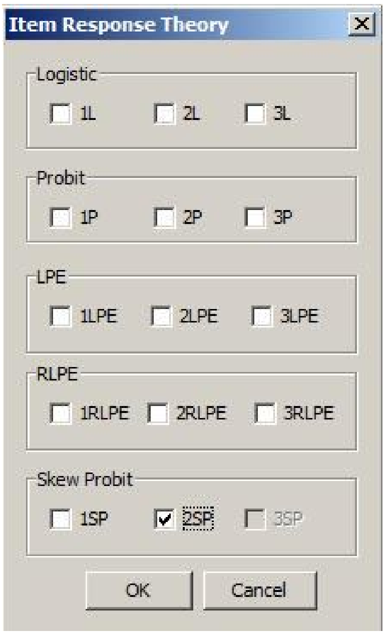


5. Then select the items that will be used. As well as to indicate if will use all the data or only a part of them.

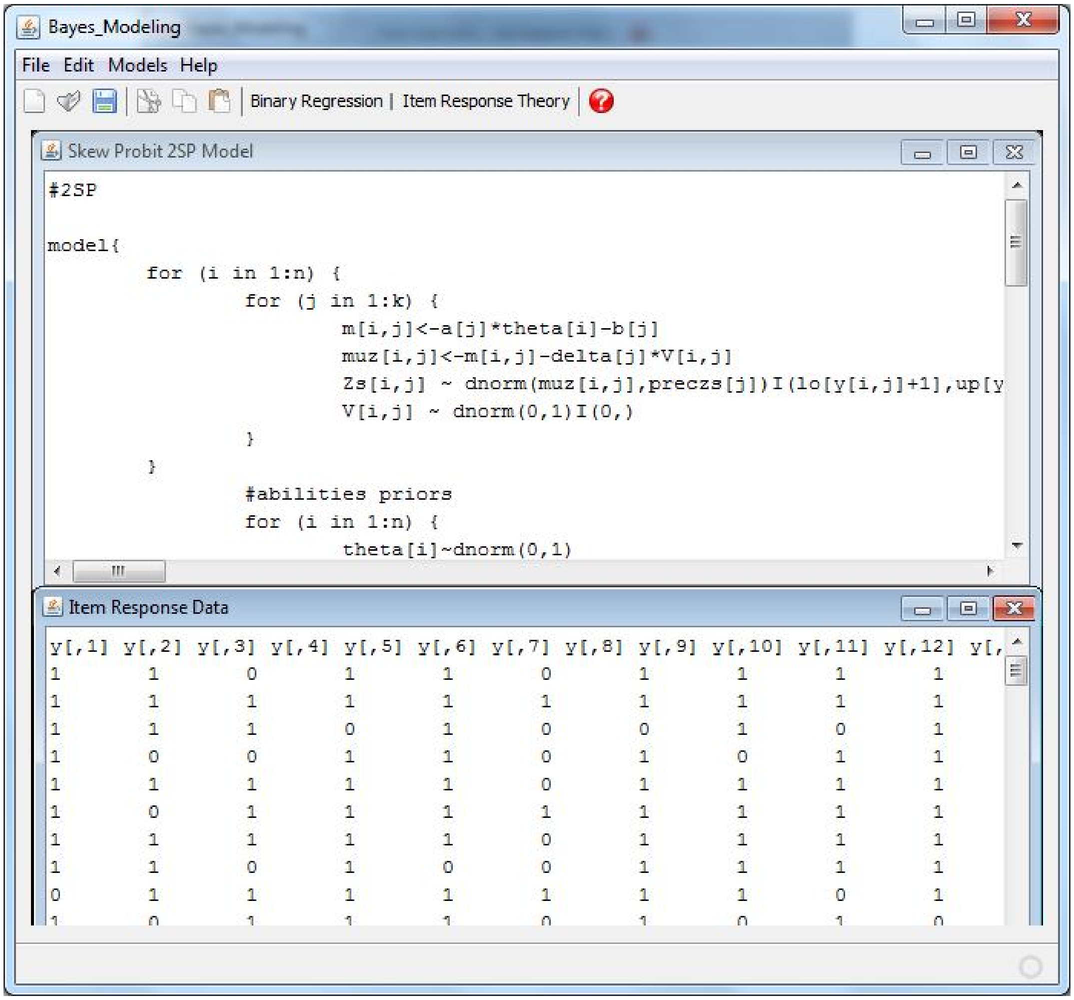
In our case will select all the variables as items and then click in *All*.



6. Then click in *Models* to open the dialogue box "*Item Response Theory*". Here have to select the models that will be used, in this example only select the 2SP model and click OK.



7. This generates two data files: the file with the syntax of the model chosen in WinBUGS (Skew Probit 2SP Model) and another file with the syntax of the data. (Item Response Data).

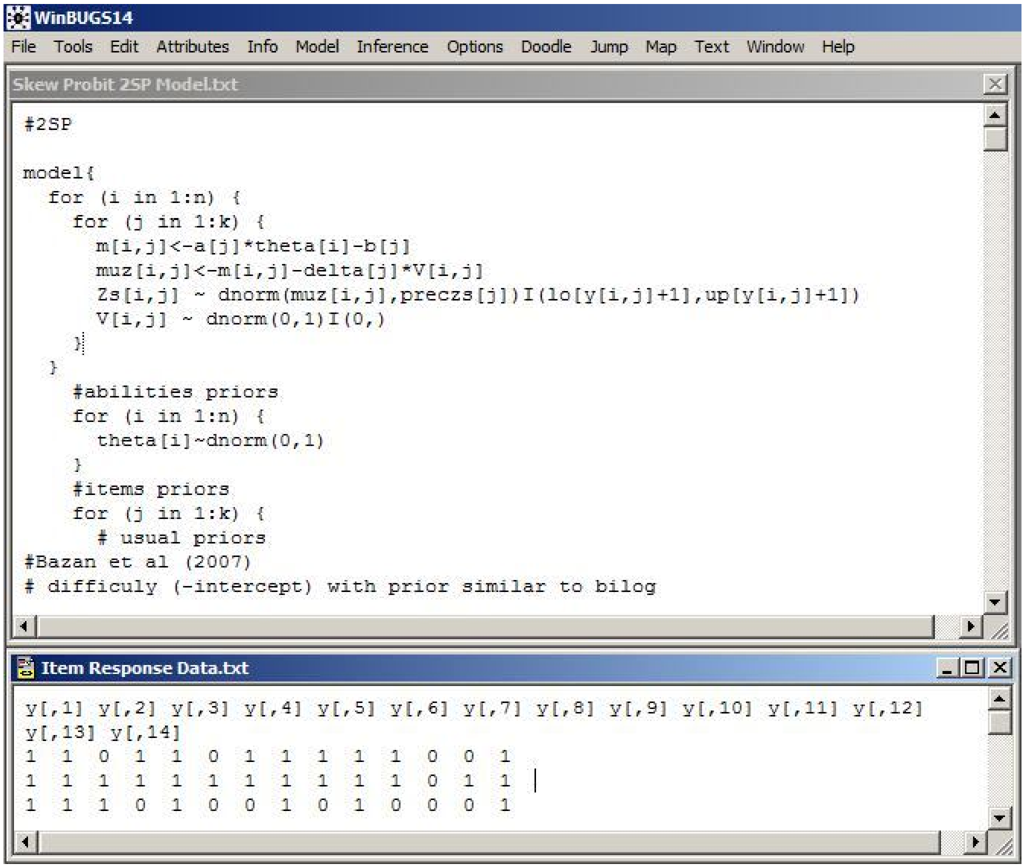


BayesianModeling generates two files, one that contains the model of Binary regression with the link selected and another file that contains the data set. Both files in format txt have to be saved to be opened in the program WinBUGS or OpenBUGS to do the appropriate analysis of Bayesian inference.

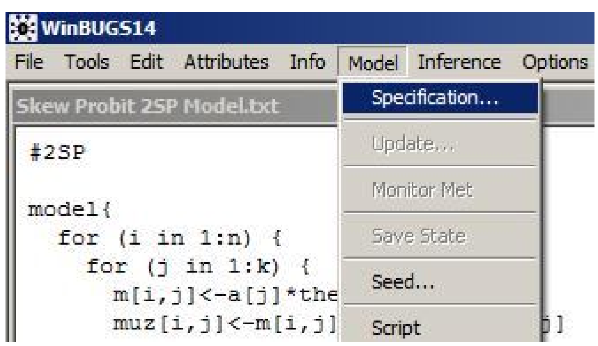
2.5.2 Bayesian Estimation using WinBUGS or OpenBUGS

For a appropriate analysis of Bayesian inference of the model generated make the following.

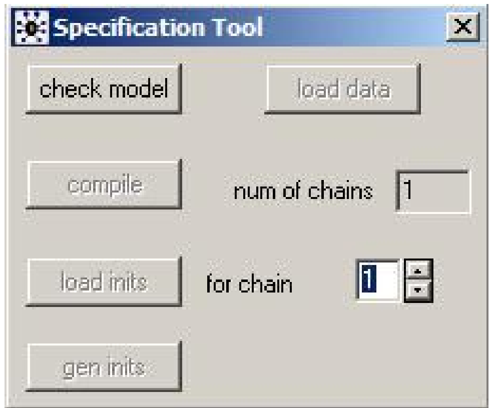
1. Open the files with the syntax of the model and of the data previously generated by the *BayesianModeling* in WinBUGS or OpenBUGS.



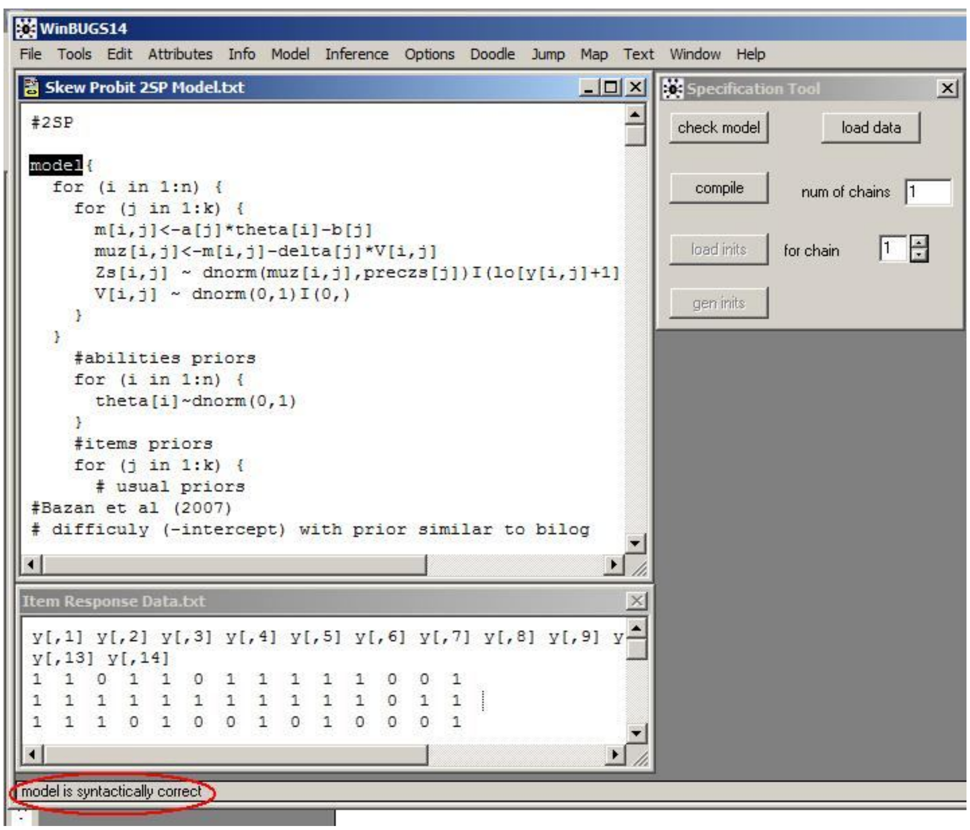
2. Having activated the window of Skew Probit 2SP Model. txt, click *Model* > *Specification*



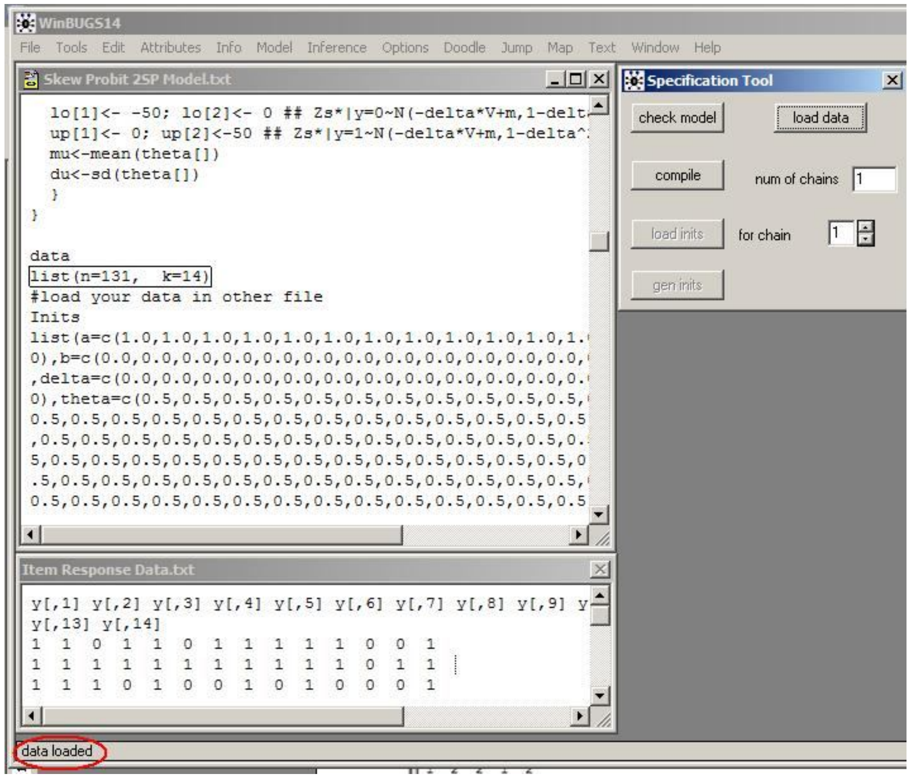
3. This will open the dialogue box “*Specification Tool*”.



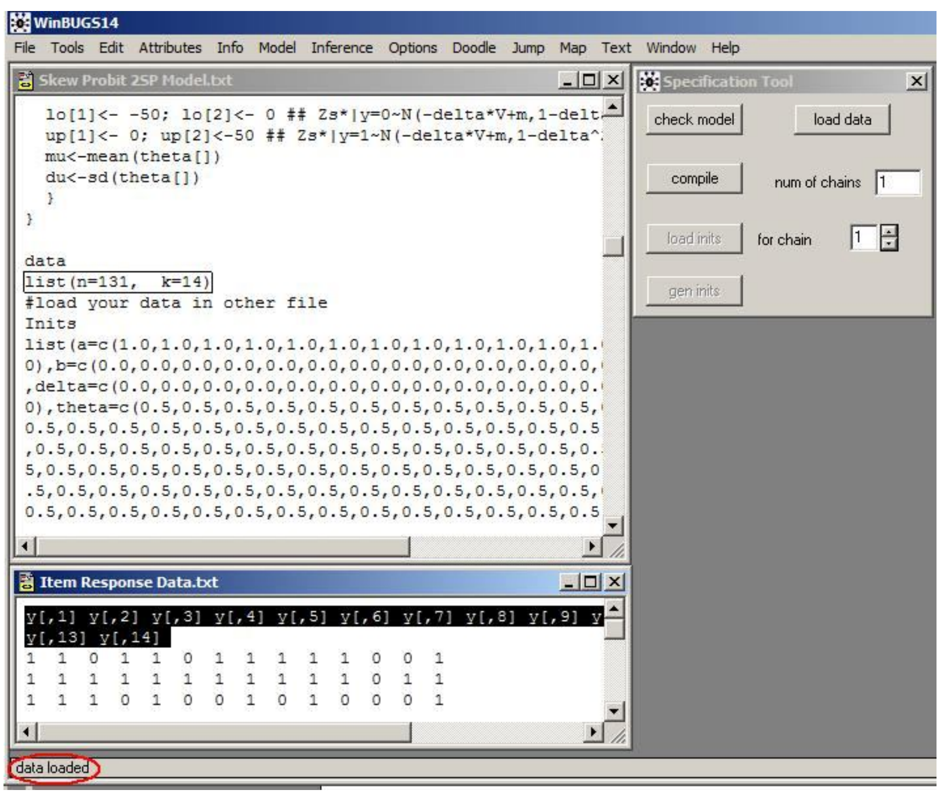
4. Select the model, highlighting the word *model* and click *check model*. In the left corner below has to appear "*model is syntactically correct*" that indicates that the syntax of the model has been properly formulated



5. Select in the Skew Probit 2SP Model.txt file, the line under *data* and do click *load data*. In the left corner below appears "*data loaded*" indicating that the data have been loaded.

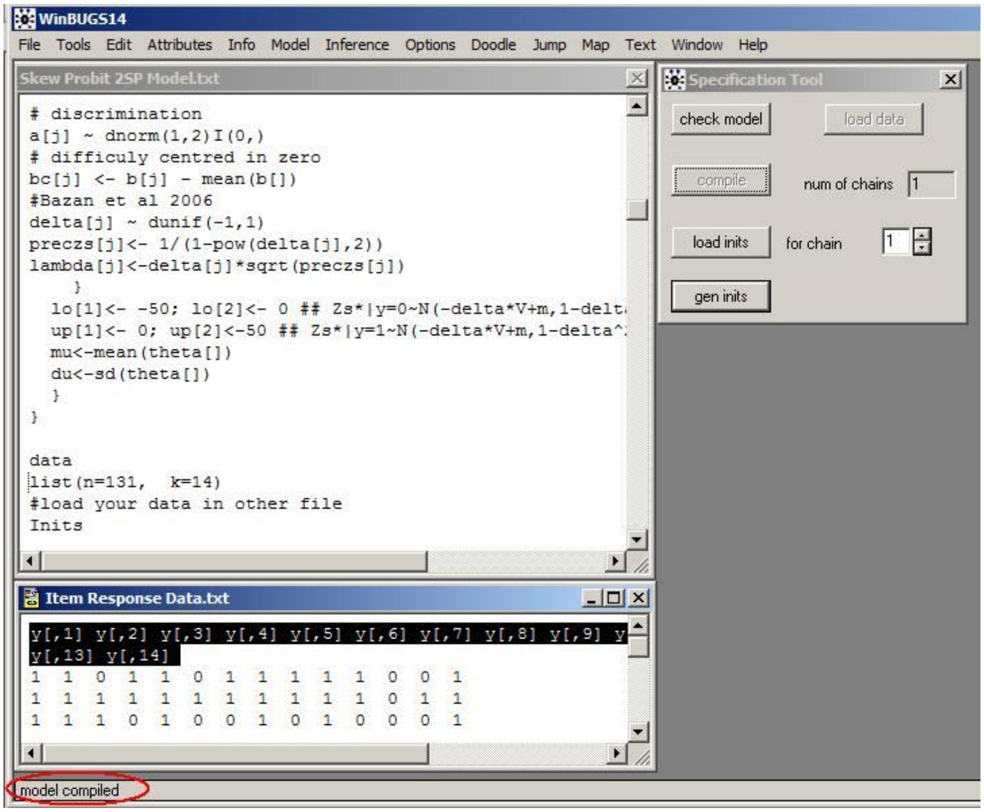


6. In the data file select the list of the variables that are placed in the first row and click *load data*.

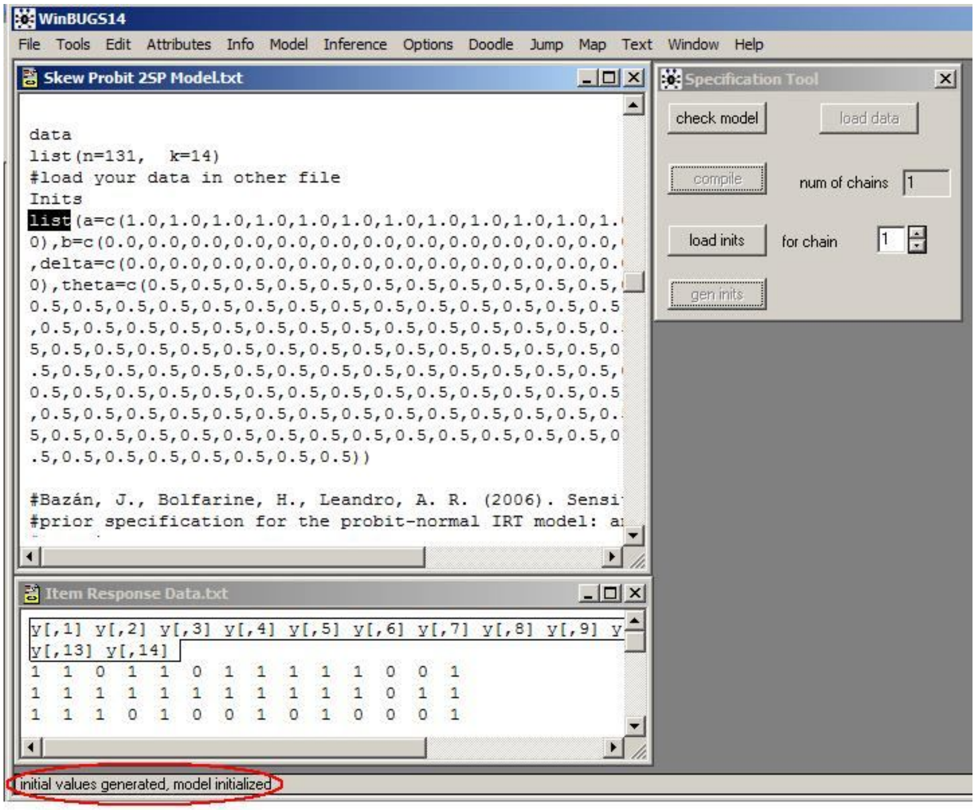


7. In the dialogue box "*Specification tool*" indicate the number of chains that want to generate in the text box "num of chains". Once specified the number of chains to generate (in this example 1 chain) do click *compile.*

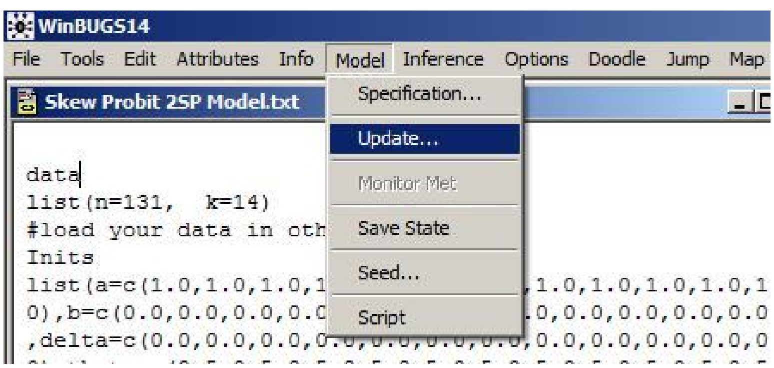
   In the left corner below has to appear "*model compiled*".



8. Select the line under *Inits* in the file of the model and click *load inits*. Then click *gen inits*. This generates the initial values for the Bayesian estimation. In the left corner below has to appears "*initial values generated*, *model initialized*"



9. Click *Model* > *Update*



10. This will open the dialogue box "*Update tool*". In the text box *updates* enter the number of iterations that requires and then click *update*.

While the program does the iterations, in the left corner below will appear the following message "*model is updating*" until the iterations finish when the following message "4000 updates took 61 s" appears.

11. Then should specify that parameters need the program save, for this go to *Inference* > *Samples*, which will open the dialogue box "*Sample monitor tool*". In the text box *node* type the name of the parameter and then click *set*; this has to be done for each parameter.

12. Repeat the step 10 generating more iterations that now have being saved by the WinBUGS or OpenBugs. In the dialogue box "*Sample Monitor Tool*", can calculate posteriori statistics of the parameters clicking *stats*, a historical of the chains clicking *history*, an estimation of the posteriori density and others statistics of the chains can be calculated using this dialogue box.

2.5.3 Bayesian Estimation using WinBUGS or OPENBUGS in R

As we have seen in the previous section with the two files that generates the *BayesianModeling* can implement the Bayesian Estimation using WinBUGS or OpenBugs programs. Also Bayesian estimation can be implemented by using R2WinBUGS or R2OpenBUGS (Sturtz, Ligges and Gelman, 2005), packages for Running WinBUGS and OpeBUGS from R respectively or BRugs (Thomas, et al, 2006) a collection of R functions that allow users to analyze graphical models using MCMC techniques.

For this will need the original text file with the data in columns:

```
I01  I02  I03     I04     I05  I06  ...  I12  I13  I14
1    1    0    1    1    0    1    ...  0    0    1
1    1    1    1    1    1    1    ...  0    1    1
.    .    .    .    .    .    .     .   .    .    .
.    .    .    .    .    .    .     .   .    .    .
.    .    .    .    .    .    .     .   .    .    .
1    1    0    1    0    0    1    ...  0    1    1
```

i.e. the file called MathData.dat (See section 2.5.1) and the syntax of the model generated in BayesianModeling will have to copy only the syntax of the model. In the example below implement the 2SP model.

```
model{
  for (i in 1:n) {
        for (j in 1:k) {
              m[i,j]<-a[j]*theta[i]-b[j]
              muz[i,j]<-m[i,j]-delta[j]*V[i,j]
              Zs[i,j] ~ dnorm(muz[i,j],preczs[j])I(lo[y[i,j]+1],up[y[i,j]+1])
              V[i,j] ~ dnorm(0,1)I(0,)
        }
  }
        #abilities priors
        for (i in 1:n) {
              theta[i]~dnorm(0,1)
        }
        #items priors
        for (j in 1:k) {
              # usual priors
#Bazan et al (2006)
# difficulty (-intercept) with prior similar to bilog
b[j] ~ dnorm(0,0.5)
# discrimination
a[j] ~ dnorm(1,2)I(0,)
# difficulty centred in zero
bc[j] <- b[j] - mean(b[])
#Bazan et al 2006
delta[j] ~ dunif(-1,1)
preczs[j]<- 1/(1-pow(delta[j],2))
lambda[j]<-delta[j]*sqrt(preczs[j])
        }
  lo[1]<- -50; lo[2]<- 0 ## Zs*|y=0~N(-delta*V+m,1-delta^2)I(-50,0)
  up[1]<- 0; up[2]<-50 ## Zs*|y=1~N(-delta*V+m,1-delta^2)I(0,50)
  mu<-mean(theta[])
  du<-sd(theta[])
  }
}

data
list(n=131,  k=14)
#load your data in other file
Inits
list(a=c(1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0),b=c(0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0),delta=c(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0),theta=c(0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,
0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,
0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,
0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,
0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,
0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,
0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5))

#Bazán, J., Bolfarine, H., Leandro, A. R. (2006). Sensitivity analysis of
#prior specification for the probit-normal IRT model: an empirical study.
#Estadística, Journal of the Inter-American Statistical Institute. 58(170-171), 17-
42.
#Available in http://www.ime.usp.br/~jbazan/download/bazanestadistica.pdf

#Bazán, J. L., Branco, D. M. & Bolfarine (2006). A skew item response model.
#Bayesian Analysis, 1, 861-892.
```

This should copy the syntax before “*data*” and save it in a file, for this example *modelirt.txt*. Then the file *modelirt.txt* would remain

```
model{
     for (i in 1:n) {
            for (j in 1:k) {
                   m[i,j]<-a[j]*theta[i]-b[j]
                   muz[i,j]<-m[i,j]-delta[j]*V[i,j]
                   Zs[i,j] ~
dnorm(muz[i,j],preczs[j])I(lo[y[i,j]+1],up[y[i,j]+1])
                   V[i,j] ~ dnorm(0,1)I(0,)
            }
     }
            #abilities priors
            for (i in 1:n) {
                   theta[i]~dnorm(0,1)
            }
            #items priors
            for (j in 1:k) {
                   # usual priors
#Bazan et al (2006)
# difficulty (-intercept) with prior similar to bilog
b[j] ~ dnorm(0,0.5)
# discrimination
a[j] ~ dnorm(1,2)I(0,)
# difficulty centred in zero
bc[j] <- b[j] - mean(b[])
#Bazan et al 2006
delta[j] ~ dunif(-1,1)
preczs[j]<- 1/(1-pow(delta[j],2))
lambda[j]<-delta[j]*sqrt(preczs[j])
            }
     lo[1]<- -50; lo[2]<- 0 ## Zs*|y=0~N(-delta*V+m,1-delta^2)I(-50,0)
     up[1]<- 0; up[2]<-50 ## Zs*|y=1~N(-delta*V+m,1-delta^2)I(0,50)
     mu<-mean(theta[])
     du<-sd(theta[])
     }
}
```

Then, to implement the Bayesian estimation in R will follow the next steps to use the library R2WinBUGS. Remember to install it previously.

1. In R, download the library R2WinBUGS with the following command:

```
library(R2WinBUGS)
```

2. Read the data (the MathData.dat file for this example is placed in the folder F:\MILUS\MathData.dat)

```
datos <- read.table("F:/MILUS/MathData.dat", header=TRUE, sep="",
na.strings="NA", dec=".",strip.white=TRUE)
```

3. Create a *list* that contain the data and the information of the number of persons and items using the following command

```
n=nrow(datos)
k=ncol(datos)
data<-list(y=as.matrix(datos),n=131,k=14)
```

4. Create a program that will generate initial values.

```
inits <- function(){
list(a=rep(1,k),b=rep(0,k),delta=rep(0,k),theta=rep(n,0.5))}
```

5. Finally the command *bugs* implements the Bayesian estimation. Here will explain in brief the syntax of the command *bugs*

`parameters.to.save =` is a vector with the names of the parameters of the model which simulations want to store.

`model.file =` is the name of the file where the model is saved.

`n.chains =` is the number of chains that will be generated.

`n.iter =` is the number of total iterations of each chain.

`n.burnin =` is the number of iterations that will be discharged as burn-in.

`program =` is the program that will be used to implement the Bayesian inference

`n.burnin =` is the number of iterations that will be discharged as burn-in.

Then the following command implements the Bayesian estimation and the simulations are stored in the object `out.`

```
out<-
bugs(data,inits,parameters.to.save=c("a","b","delta"),
model.file="F:/MILUS/modelirt.txt", n.chains=1, n.iter=24000,
n.burnin=4000, program="WinBUGS")
```

6. If type `out` in the line of commands of R obtain a summary of the simulation

```
> out
Inference for Bugs model at "F:/MILUS/modelirt.txt", fit using WinBUGS,
 1 chains, each with 24000 iterations (first 4000 discarded), n.thin = 20
 n.sims = 1000 iterations saved
            mean   sd   2.5%    25%    50%    75%  97.5%
a[1]         0.5  0.2    0.1    0.3    0.5    0.6    1.0
a[2]         0.3  0.2    0.0    0.1    0.2    0.4    0.6
a[3]         0.5  0.2    0.1    0.3    0.5    0.6    0.9
a[4]         0.9  0.3    0.3    0.6    0.8    1.1    1.6
a[5]         0.5  0.2    0.1    0.3    0.4    0.6    1.0
a[6]         0.3  0.2    0.0    0.2    0.3    0.4    0.7
a[7]         0.8  0.3    0.3    0.6    0.8    1.0    1.5
a[8]         0.9  0.3    0.3    0.7    0.9    1.1    1.7
a[9]         0.2  0.1    0.0    0.1    0.2    0.3    0.5
a[10]        0.4  0.2    0.1    0.3    0.4    0.6    0.9
a[11]        1.3  0.5    0.6    1.0    1.3    1.6    2.4
a[12]        0.3  0.2    0.0    0.2    0.3    0.4    0.7
a[13]        0.4  0.2    0.1    0.3    0.4    0.6    0.9
a[14]        0.4  0.3    0.0    0.2    0.4    0.6    1.1
```

```
b[1]         -0.9  0.4   -1.5   -1.2   -0.9   -0.6    0.0
b[2]         -0.9  0.4   -1.6   -1.3   -1.0   -0.6    0.0
b[3]          0.0  0.4   -0.7   -0.4    0.0    0.3    0.7
b[4]         -1.7  0.6   -2.7   -2.1   -1.7   -1.2   -0.5
b[5]         -1.1  0.5   -1.8   -1.4   -1.1   -0.8   -0.1
b[6]          0.3  0.4   -0.5    0.0    0.3    0.6    1.0
b[7]         -1.6  0.6   -2.6   -2.0   -1.6   -1.2   -0.5
b[8]         -1.3  0.6   -2.3   -1.7   -1.4   -0.9   -0.2
b[9]         -0.7  0.4   -1.3   -1.0   -0.7   -0.4    0.2
b[10]        -1.0  0.5   -1.7   -1.4   -1.0   -0.6    0.0
b[11]        -1.9  0.7   -3.3   -2.3   -1.9   -1.5   -0.5
b[12]         0.4  0.4   -0.4    0.0    0.4    0.8    1.0
b[13]        -0.9  0.4   -1.6   -1.3   -1.0   -0.7    0.0
b[14]        -1.6  0.5   -2.4   -1.9   -1.6   -1.2   -0.5
delta[1]      0.1  0.5   -0.8   -0.4    0.0    0.5    1.0
delta[2]     -0.1  0.5   -0.9   -0.5   -0.1    0.2    0.9
delta[3]      0.0  0.5   -0.9   -0.4    0.0    0.4    0.9
delta[4]     -0.1  0.5   -0.9   -0.5   -0.2    0.2    0.9
delta[5]     -0.1  0.5   -0.9   -0.5   -0.1    0.3    0.9
delta[6]      0.0  0.5   -0.9   -0.4    0.0    0.4    1.0
delta[7]     -0.1  0.5   -0.9   -0.6   -0.2    0.3    0.9
delta[8]     -0.1  0.5   -0.9   -0.6   -0.2    0.2    0.9
delta[9]     -0.1  0.5   -0.9   -0.4   -0.1    0.3    0.8
delta[10]    -0.1  0.5   -0.9   -0.5   -0.2    0.3    0.9
delta[11]    -0.3  0.6   -1.0   -0.8   -0.5    0.1    0.9
delta[12]     0.0  0.5   -1.0   -0.5    0.0    0.4    0.9
delta[13]     0.0  0.5   -0.8   -0.4    0.0    0.4    1.0
delta[14]    -0.1  0.5   -0.9   -0.5   -0.1    0.3    0.9
deviance   3855.8 66.3 3703.9 3814.0 3865.0 3904.0 3963.0

DIC info (using the rule, pD = Dbar-Dhat)
pD = -52.0 and DIC = 3803.7
DIC is an estimate of expected predictive error (lower deviance is better).
```

Note that for now we just asked for monitoring the parameters a, b and delta. But if it requires could ask for θ.

7. Finally, for more details in the command *bugs* can consult Help writing in the line of commands

```
?bugs
```

Note. You can specify Bug.directory. The directory that contains the **WinBUGS** executable. If the global option R2WinBUGS.bugs.directory is not NULL, it will be used as the default. Also you can specify the program to use, either winbugs/WinBUGS or openbugs/OpenBUGS, the latter makes use of function openbugs and requires the CRAN package **BRugs**. In addition, because of the large number of parameters in IRT models, execution may be delayed!

References

Albert, J (2009). *Bayesian Computation with R*. Springer Verlag

Basu S, Mukhopadhyay S (2000). "Binary response regression with normal scale Mixtures links." In DK Dey, SK Ghosh, BK Mallick (eds.), "Generalized Linear Models: A Bayesian Perspective". New York: Marcel Dekker.

Bazán JL (2010). *Manual de uso de BRMUW*, *Version 1.0*. Software. Departamento de Ciencias. PUCP. URL http://argos.pucp.edu.pe/~jlbazan/download/ManualdeusoBRMUW.pdf

Bazán JL, Branco MD, Bolfarine H (2006) "A skew item response model." *Bayesian Analysis*, **1**, 861- 892.

Bazán JL, Bolfarine H, Branco MD (2010) "A framework for skew-probit links in Binary regression." *Communications in Statistics - Theory and Methods*, **39**, 678-697

Bazán JL, Bolfarine H, Branco MD (2006). *A generalized skew probit class link for binary regression*. Technical report (RT-MAE-2006-05). Department of Statistics. University of São Paulo. URL http://argos.pucp.edu.pe/~jlbazan/download/gspversion14.pdf

Bazán JL, Bolfarine H Leandro AR (2006). "Sensitivity analysis of prior specification for the probit-normal IRT model: an empirical study." *Estadística*, *Journal of the Inter-American Statistical Institute* **58(**170-171), 17-42. URL http://argos.pucp.edu.pe/~jlbazan/download/bazanestadistica.pdf

Bazán, JL, Valdivieso L, Calderón A (2010). *Enfoque bayesiano en modelos de Teoría de Respuesta al Ítem*. Reporte de Investigación. Serie B. Nro 25. Departamento de Ciencias. PUCP. URL http://argos.pucp.edu.pe/~jlbazan/download/Reportef27.pdf

Birnbaum A (1968). Some Latent Trait Models and Their Use in Infering an Examinee's Ability. In FM Lord, MR Novick (eds) *Statistical Theories of Mental Test Scores*. New York: Addison-Wesley.

Bolfarine H, Bazán JL (2010). “Bayesian estimation of the logistic positive Exponent IRT Model”. *Journal of Educational Behavioral Statistics*, **35**, 6, 693-713

Carlin BP, Louis TA (2000). *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, CRC, London, Boca Raton, FL.

Collet D (2003). *Modelling binary data*. Chapman & Hall/CRC, Second Edition, Boca Raton, USA.

Congdon P (2010). *Applied Bayesian Hierarchical Methods*, Chapman & Hall / CRC.

Congdon P (2005). *Bayesian Models for Categorical Dates*, Wiley.

Chen MH, Dey D, Shao Q-M. (2001). “Bayesian analysis of binary data using Skewed logit models.” *Calcutta Statistical Association Bulletin*, **51**, 201-202.

Curtis MS (2010) “BUGS Code for Item Response Theory.” *Journal of Statistical Software*. **36**(1), 1-34.

Fischer G, Molenaar I (1995). *Rasch Models Foundations, recent development, and applications*. The Nerthelands: Springer-Verlag.

Fox JP (2010). *Bayesian Item Response Modeling: Theory and Applications*. New York: Springer.

Fu ZH, Tao J, Shi NZ (2009). “Bayesian estimation in the multidimensional three-parameter logistic model.” *Journal of Statistical Computation and Simulation*, **79**, 819 - 835.

Gilks W, Richardson S, Spiegelhalter D (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

Gilks, WR, Wild P (1992). "Adaptive rejection sampling for Gibbs sampling." *Applied Statistics*, **41**: 337-348

Johnson V, Albert J (1999). *Ordinal Data Modeling*. New York: Springer-Verlag.

Lord FM (1952). *A theory of test scores*. New York: Psychometric Society.

Martin AD, Quinn KM, Park, JH (2011). "MCMCpack: Markov Chain Monte Carlo in R." *Journal of Statistical* Software, 42(9): 1-21.

Nagler J (1994). "Scobit: An alternative estimator to Logit and Probit." *American Journal of Political Science*, ***38***(1), 230-255.

Ntzoufras I (2009). *Bayesian Modeling Using WinBugs*. Wiley Series in Computational Statistics, Hoboken, USA.

Patz, RJ, Junker, BW (1999). "A straightforward approach to Markov Chain Monte Carlo methods for item response models." *Journal of Educactional and Behavioral Statistics*, ***24***, 146-178.

Prentice RL (1976). "A Generalization of the probit and logit methods for Dose response curves". *Biometrika*, **32**(4), 761-768.

R Development Core Team (2004). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org.

Rupp A, Dey DK, Zumbo B (2004). "To Bayes or Not to Bayes, from Whether to When: Applications of Bayesian Methodology to Item Response Modeling." *Structural Equations Modeling*. **11**, 424-451.

Sahu SK (2002). "Bayesian estimation and model choice in item response models". *Journal Statistical Computing Simulation*, **72**: 217-232.

Samejima F (2000). "Logistic positive exponent family of models: Virtue of asymmetric item characteristic curves." *Psychometrika*, **65**(3): 319-335.

Sheng Y (2008a). “Markov Chain Monte Carlo Estimation of Normal Ogive IRT Models in MATLAB.” *Journal of Statistical Software*, **25**(8), 1–15.

Sheng Y (2008b). “A MATLAB Package for Markov Chain Monte Carlo with a Multi-unidimensional IRT Model.” *Journal of Statistical Software*, **28**(10), 1–19.

Sheng Y (2010). Bayesian Estimation of MIRT Models with General and Specific Latent Traits in MATLAB. *Journal of Statistical Software,* **34**(3), 1-27*.*

Spiegelhalter DJ, Thomas A, Best NG, Gilks, WR (1996) *BUGS 0.5 examples* (Vol. 1 Version i). Cambridge, UK: University of Cambridge.

Spiegelhalter DJ, Thomas A, Best NG, Lunn D (2003). *WinBUGS Version 1.4 Users Manual*. MRC Biostatistics Unit, Cambridge. URL http://www.mrc-bsu.cam.ac.uk/bugs/.

Spiegelhalter DJ, Thomas A, Best NG, Lunn D (2007). *OpenBUGS User Manual version 3.0.2*. MRC Biostatistics Unit, Cambridge.

Sturtz, S, Ligges U, Gelman A. (2005). “R2WinBUGS: A Package for Running WinBUGS from R.” *Journal of Statistical Software,* **12**(3), 1-16.

Thomas A, O’Hara B, Ligges U, Sturtz S (2006). “Making BUGS Open.” *R News*, **6**(1), 12–17. URL http://CRAN.R-project.org/doc/Rnews/.