

MANUAL DE USO DE BRMUW

por Dr. Jorge Luis Bazan

Departamento de Ciencias. Pontificia Universidad Católica del Peru

e-mail: jlbazan@pucp.edu.pe home page <http://argos.pucp.edu.pe/~jlbazan/>

Índice

Introducción

1. Modelo de Regresión Binaria
2. Enlaces Asimétricos en Regresión Binaria
3. Inferencia Bayesiana
4. Ejemplo de Aplicación
5. Uso del BMRUW
 - 5.1 Generar la sintaxis del modelo
 - 5.2 Generar la sintaxis para los datos.
 - 5.3 Estimación Bayesiana usando WinBUGS u OpenBugs
 - 5.4 Estimación Bayesiana usando WinBUGS u OpenBugs dentro R

Introducción

BRMUW (Bayesian Regression Model using WinBUGS) es un software para generar sintaxis de varios modelos de Regresión Binaria bajo un enfoque bayesiano usando MCMC que pueden ser ejecutadas en los programas [OpenBUGS](#), [WinBUGS](#) o en [R](#) a través de las librerías [R2WinBUGS](#) o [Brugs](#).

Este aplicativo es para uso personal y no debe ser usado para cualquier propósito comercial sin mi autorización escrita explícita. El aplicativo es proporcionado “como está” sin garantía de cualquier tipo. Para implementar apropiadamente los diferentes modelos mencionados en el aplicativo Ud. debe leer mínimamente la literatura sugerida en las referencias y debe estar familiarizado con la Inferencia Bayesiana usando MCMC.

BRMUW está pensado en usuarios aplicados que dada una base de datos desean conocer la sintaxis de diversos modelos de Regresión binaria usualmente no disponibles en diversos programas estadísticos incluyendo el programa *R*.

Este aplicativo básico ha sido desarrollado como parte de los proyectos DAI 3412, 4031 y 2009-0033 de la [Pontificia Universidad Católica del Perú](#). Actualmente estamos trabajando en un proyecto que implementa estos mismos modelos y otros adicionales en un aplicativo denominado BAYESPUCP.

El aplicativo junto a algunos modelos disponibles ha sido desarrollado por 3 años y a lo largo de ese tiempo diversas personas han colaborado en el proyecto por lo que deseo expresar mis agradecimientos. Entre ellos a los colegas Oscar Millones y Cristian Bayes por su ayuda durante la revisión de la actual versión del aplicativo y el manual. También deseo agradecer a Adrian Paucar, Enver Tarazona, Luisa Ames, Margareht Sequeiros y Pedro Curich por el apoyo informático en alguna de las etapas del proyecto. Muchas gracias a mi familia por su paciencia y apoyo.

1. Modelo de Regresión Binaria

Considere un modelo de regresión binaria

$$y_i \sim \text{Bernoulli}(\pi_i)$$
$$\pi_i = F(\mathbf{x}_i^T \boldsymbol{\beta})$$

Siendo y_i una variable binaria tal que $y_i = 1$ ocurre con probabilidad π_i , $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ik})^T$ un vector con los valores de k variables explicativas, $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_k)^T$ un vector de k coeficientes de regresión y $F(\cdot)$ denota una función de distribución acumulada (fda). La función inversa es comúnmente denominada función de enlace y $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$ es el i -ésimo predictor lineal.

Cuando F es una fda de una distribución simétrica la función de enlace resultante es simétrica y tiene una forma simétrica alrededor de $\pi_i = 0.5$. En el caso que F sea la fda de una normal estándar tenemos el enlace *probit*,

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

y en el caso de F sea la fda de una distribución logística obtenemos el enlace *logit*,

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

Estos enlaces probit y logit son implementados en BRMUW.

2. Enlaces Asimétricos en Regresión Binaria

Chen *et al* (1999) sostienen que cuando la probabilidad de una respuesta binaria se aproxima a 0 en una tasa diferente que cuando se aproxima a 1, los enlaces simétricos para el ajuste de datos pueden ser inadecuados. En este caso, hay que considerar enlaces asimétricos. El programa BRMUW considera la fda de distribuciones asimétricas para construir enlaces asimétricos. Un ejemplo muy popular es el enlace log-log complementario o *cloglog*, donde la fda usada en el enlace corresponde a la Distribución de Gumbel.

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

En este caso, la fda está completamente especificada, no depende de ningún parámetro adicional desconocido y no presenta como caso particular un enlace simétrico.

Información de como implentar la estimación bayesiana de la regresión binaria usando los enlaces cloglog, probit y logit en WinBUGS u OpenBUGS puede ser vista en el Ejemplo [Beetles:logistic,probit and extreme value models](#) del Manual. Sin embargo la regresión binaria bayesiana considerando otros enlaces como los discutidos en Bazán, Bolfarine y Branco (2006 y 2010) actualmente no se encuentran disponibles.

Otros enlaces considerados en BRMUW es aquel que se obtiene considerando otras funciones de distribución acumuladas como las siguientes:

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

estos enlaces son logit asimetrizados y son conocidos como *scobit* y *power logit*, respectivamente, e incluyen al enlace logit como caso especial cuando el parámetro $\lambda = 1$. Para una revisión de estos enlaces ver Prentice (1976) y Nagler (1994).

En BRMUW también son implementados tres enlaces que se basan en la fda de una distribución normal asimétrica, esta acumulada puede ser representada de manera general la siguiente manera:

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

donde $\mathbf{x} = (t, 0)'$; $\boldsymbol{\theta} = (\mu, \sigma^2, \lambda)'$; $\Phi_2(\cdot)$ representa la distribución acumulada de una distribución normal bivariada con parámetros $\boldsymbol{\mu} = (\mu, 0)'$ y $\boldsymbol{\Omega} = \begin{bmatrix} \sigma^2 & -\delta \\ -\delta & 1 \end{bmatrix}$; y $\delta = \frac{\lambda}{\sqrt{1 + \lambda^2}}$, ver Bazán, Bolfarine y Branco (2006 y 2010)

Luego, los enlaces implementados en el BRMUW son los siguientes:

- Si $\boldsymbol{\theta} = (0, 1 + \lambda^2, -\lambda)$, se obtiene el enlace probit asimetrizado propuesto en Chen *et al* (1999) denominado aquí *CDS skew probit*.
- Si $\boldsymbol{\theta} = (0, 1, \lambda)$, se obtiene el enlace propuesto por Bazán, Branco y Bolfarine (2006) denominado aquí *BBB skew probit*

- Si $\theta = \left(-\frac{\sqrt{2}\delta}{\sqrt{\pi - 2\delta^2}}, \frac{\pi}{\pi - 2\delta^2}, \lambda \right)$, obtenemos el enlace denominado estándar probit asimétrizado (Bazán, Bolfarine y Branco 2006 y 2010), denominado aquí *Standard skew probit*.

En estos tres enlaces, λ es el parámetro que controla la asimetría, así tenemos que para valores negativos (positivos) de λ tenemos asimetría negativa (positiva).

Esta clase de modelos puede verse también como perteneciente a la clase de mezclas de distribuciones elípticas propuestas por Basu y Mukhopadhyay (2000) dada por:

$$F = \{F(\cdot) = \int_{[0, \infty]} H(\cdot | \nu) dG(\nu)\},$$

donde G es la función de distribución acumulada $[0, \infty)$ y H es una distribución elíptica. En este caso el CDS skew probit considera una clase de mezclas de normales donde la medida de mezcla es la distribución normal positiva con función de densidad dada por $g(x) = 2\phi(x)$, $x > 0$, con $\phi(\cdot)$ siendo la función de densidad de la normal estándar. Otro caso interesante cuando se mezcla la normal positiva con H la función de distribución acumulada de la distribución logística es conocida como skew logistic o *skew logit* (ver Chen, Dey y Shao, 2001) que también está implementado en BRMUW.

En resumen, los modelos de regresión binaria implementados en BRMUW clasificados según sus enlaces son

- *simétricos: probit, logit.*
- *Asimétricos: cloglog, scobit, power logit, skew logit, skew probit (CDS, BBB y standard).*

3. Ejemplo de Aplicación

El programa BRMUW genera la sintaxis necesaria para la estimación bayesiana de varios modelos de regresión binaria, en el programa WinBUGS (ver Spiegelhalter *et al* 1996) o OpenBUGS (Spiegelhalter *et al* 2007), usando diversos métodos MCMC. Para ello solo es necesario contar con un archivo de texto con los datos, generado de cualquier programa estadístico o de Excel. En las columnas usualmente aparece los nombres de las variables en la primera línea y la primera columna deberá contener la variable respuesta.

Como ejemplo consideramos una data que contiene algunas variables de un estudio con agricultores beneficiarios de un programa favorable a la erradicación de cultivos de coca (Bazán y Millones, 2008). La data se denomina concoca.txt y se encuentra dentro de la descarga del programa. Las variables en concoca.txt son

- sierr : si se muestra favorable a erradicar el cultivo de coca
- permedyc : índice de percepción de que el cultivo de coca produce daño al medio ambiente
- partco : índice de participación comunal
- concoca : índice acerca de si consume coca
- pobrez : niveles de pobreza

El archivo de datos tiene la siguiente estructura

sierr	permedyc	partco	concoca	pobrez
1	2	2	1	2
0	0	6	1	2
.
.
.
1	2	9	0	3

Como un ejemplo de aplicación consideremos el siguiente modelo

$$sierr_i \sim Bernoulli(\pi_i)$$

$$\pi_i = F(\eta_i)$$

$$\eta_i = \beta_1 + \beta_2 permedyc_i + \beta_3 partco_i + \beta_4 concoca_i + \beta_5 pobrez_i, \quad i = 1, 2, \dots, 1947$$

donde la F corresponde al enlace skew logit. Mayores detalles en Bazán y Millones (2008)

4. Estimación Bayesiana

Considerando la distribución Bernoulli para la variable respuesta, la función de verosimilitud es dada por

$$L(\beta, \theta, \lambda | y, X) = \prod_{i=1}^n [F_\theta[m(x'_i \beta, \lambda)]^{y_i} [1 - F_\theta[m(x'_i \beta, \lambda)]]^{1-y_i}$$

donde $F_\theta(\cdot)$ es la función de densidad de una distribución asimétrica indexada por el parámetro θ , que no necesariamente es unidimensional y $m(\cdot)$ es una función continua

del predictor lineal que también incluye la función identidad con λ como el parámetro de forma o parámetro asimétrico.

Para los enlaces logit, probit, cloglog, scobit y power logit se ha considerado esta función de verosimilitud, para los enlaces skew probit y skew logit se ha considerado otras versiones de la función de verosimilitud considerando versiones aumentadas que son discutidas en las referencias específicas de estos modelos.

En la Inferencia bayesiana, a diferencia de la inferencia clásica, los parámetros de interés se asumen como variables aleatorias y así se establecen diferentes distribuciones de probabilidad a priori que reflejan nuestro conocimiento previo de su conducta. Combinando la función de verosimilitud y las distribuciones a priori podemos obtener la distribución posterior de los parámetros de interés. En este trabajo, nosotros consideramos prioris que son vagas, prioris propias con distribuciones conocidas pero varianza grande así como independencia entre las prioris.

Inferencia Bayesiana es facilitada con el uso de diferentes métodos MCMC implementados en los programas WinBUGS y OpenBUGS por lo que usando una programación mínima es posible implementar estos métodos. Para una revisión nosotros sugerimos el libro de Congdon (2005).

5. Uso del BMRUW

Vamos a usar el BRMUW para implementar el modelo de regresión binaria con enlace skew logit para los datos de concoca.txt descrito en la sección 3. Para mayores detalles de la aplicación revisar Bazán y Millones (2008).

5.1 Generar la sintaxis del modelo

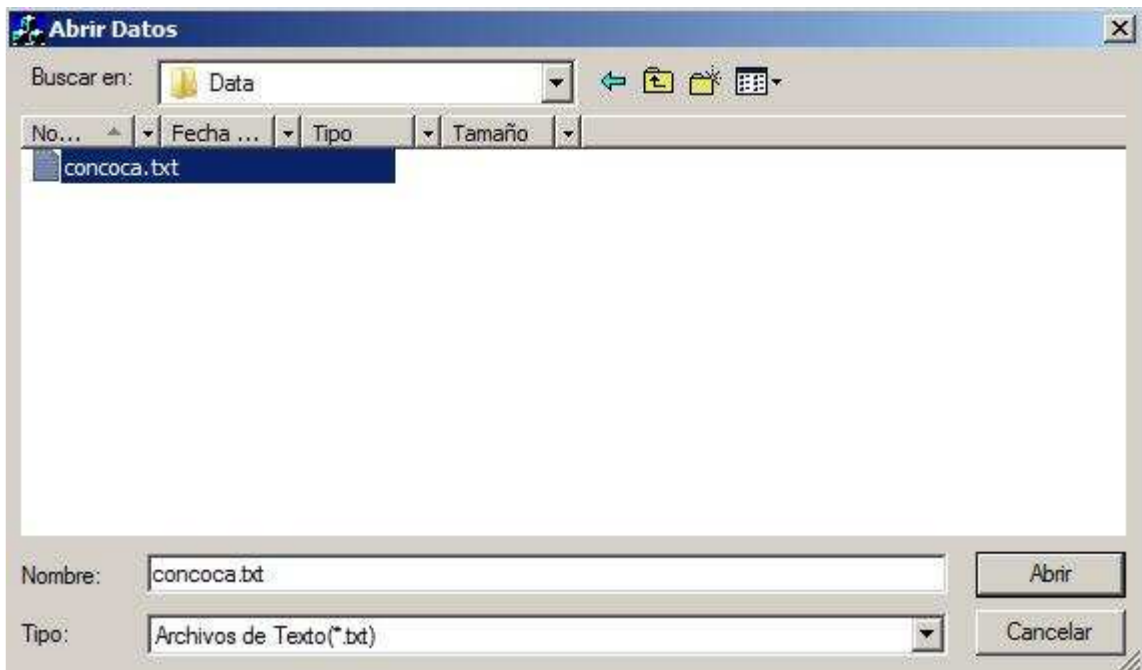
1. Escoger skew logit de *Modelos > Enlaces Asimétricos*.



2. Esto abrirá la caja de dialogo "Generación del Modelo de Regresión Binaria para WinBUGS"



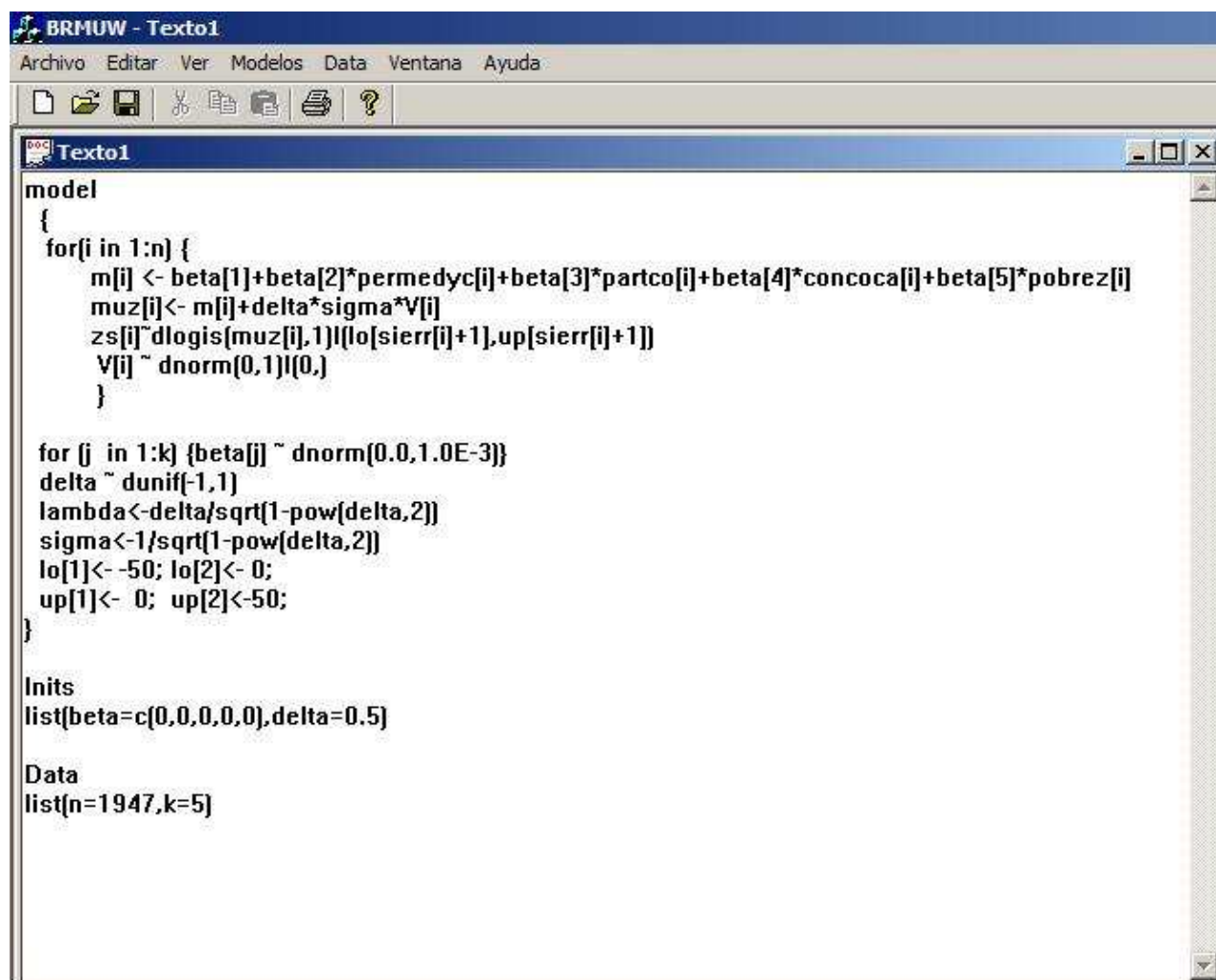
3. Hacer click en *Incluir Datos* y abrir el archivo con los datos.



4. Hacer click en *Aceptar* en "Generación del Modelo de Regresión Binaria para WinBUGS".



5. Esto genera la sintaxis del modelo.



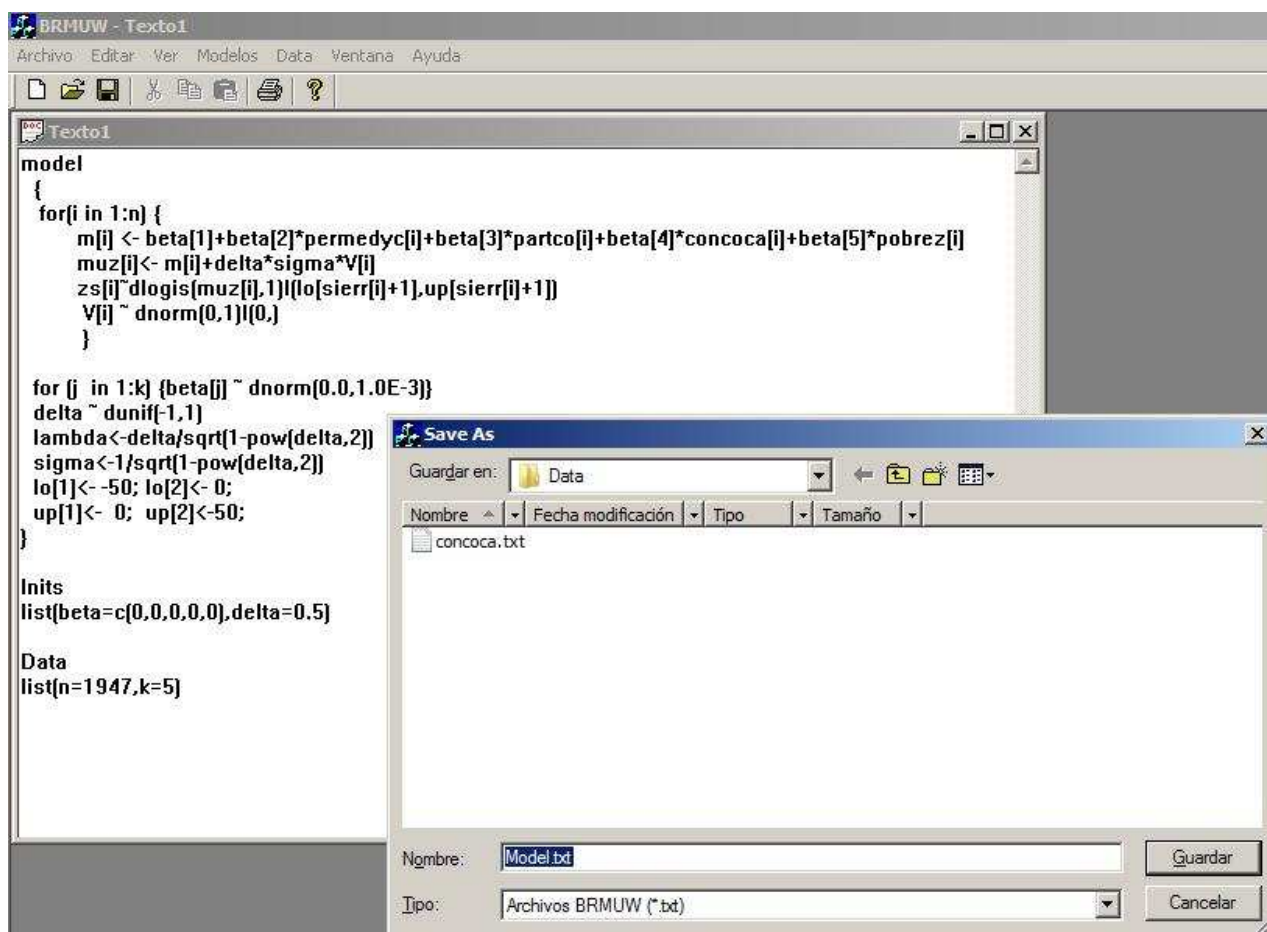
```
BRMUW - Texto1
Archivo  Editar  Ver  Modelos  Data  Ventana  Ayuda
[Icons]
Text01
model
{
  for(i in 1:n) {
    m[i] <- beta[1]+beta[2]*permedyc[i]+beta[3]*partco[i]+beta[4]*concoca[i]+beta[5]*pobrez[i]
    muz[i]<- m[i]+delta*sigma*V[i]
    zs[i]~dlogis(muz[i],1)|(lo[sierr[i]+1],up[sierr[i]+1])
    V[i] ~ dnorm(0,1)|(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;
}

Inits
list(beta=c(0,0,0,0,0),delta=0.5)

Data
list(n=1947,k=5)
```

6. Guardar la sintaxis del modelo, *Archivo > Guardar*

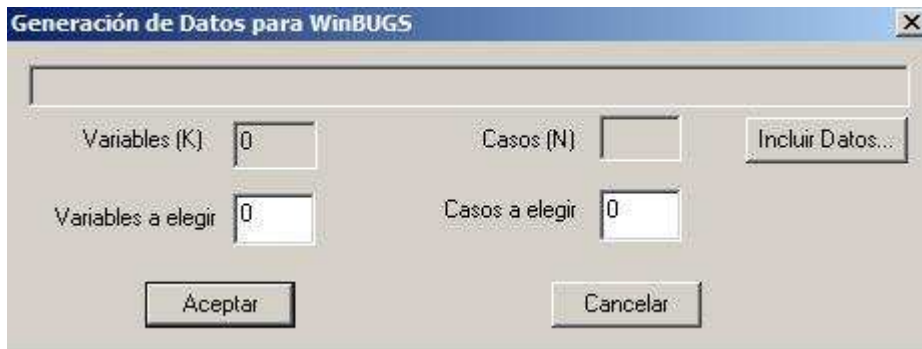


5.2 Generar la sintaxis para los datos.

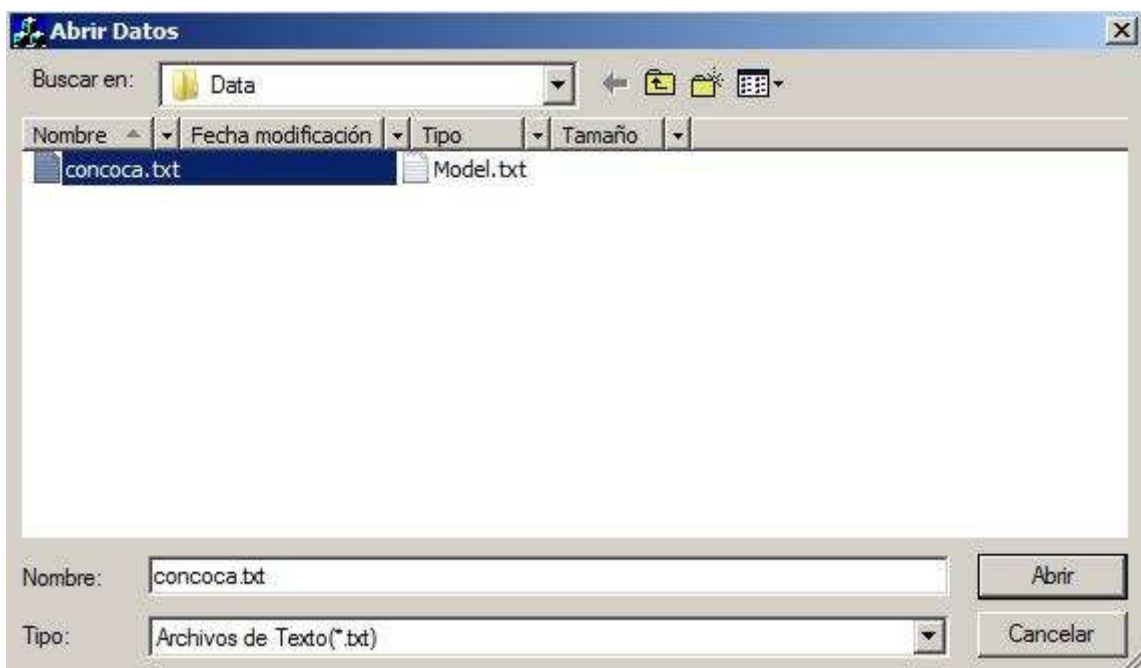
1. Hacer click en *Data > Generar data para WinBUGS*.



2. Esto abrirá la caja de dialogo “*Generación de Datos para WinBUGS*”.



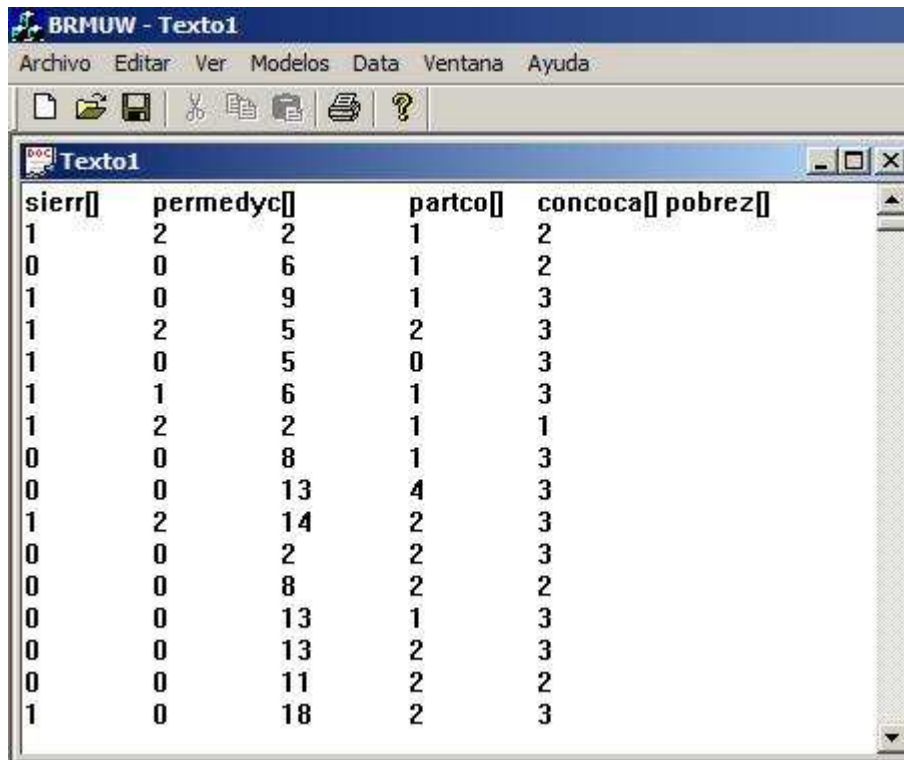
3. Hacer click en *Incluir Datos* y abrir el archivo con los datos.



4. Hacer click en *Aceptar* en “*Generación de Datos para WinBUGS*”.



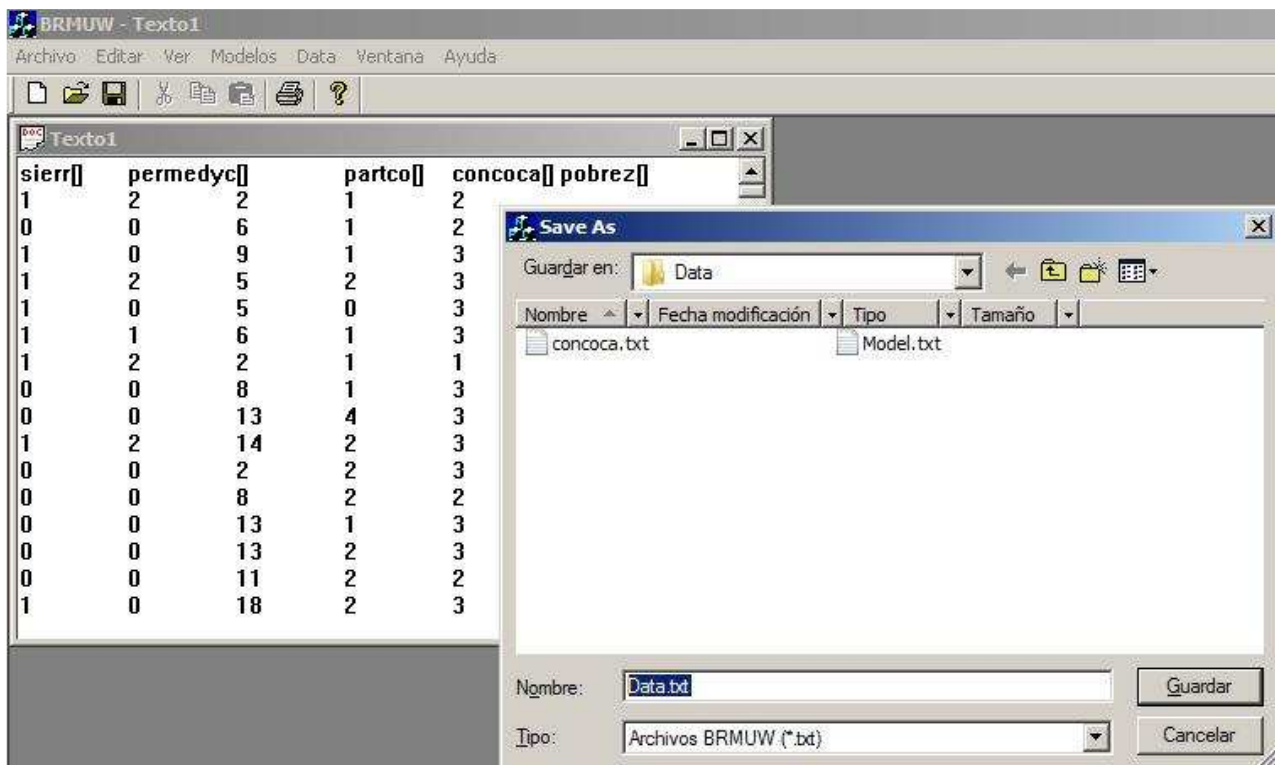
5. Esto genera la sintaxis para los datos.



The screenshot shows a window titled "BRMUW - Texto1" with a menu bar (Archivo, Editar, Ver, Modelos, Data, Ventana, Ayuda) and a toolbar. The main content area displays a table with five columns: sierr[], permedyc[], partco[], concoca[], and pobrez[]. The data is as follows:

sierr[]	permedyc[]	partco[]	concoca[]	pobrez[]
1	2	2	1	2
0	0	6	1	2
1	0	9	1	3
1	2	5	2	3
1	0	5	0	3
1	1	6	1	3
1	2	2	1	1
0	0	8	1	3
0	0	13	4	3
1	2	14	2	3
0	0	2	2	3
0	0	8	2	2
0	0	13	1	3
0	0	13	2	3
0	0	11	2	2
1	0	18	2	3

6. Guardar la sintaxis para los datos, *Archivo > Guardar*

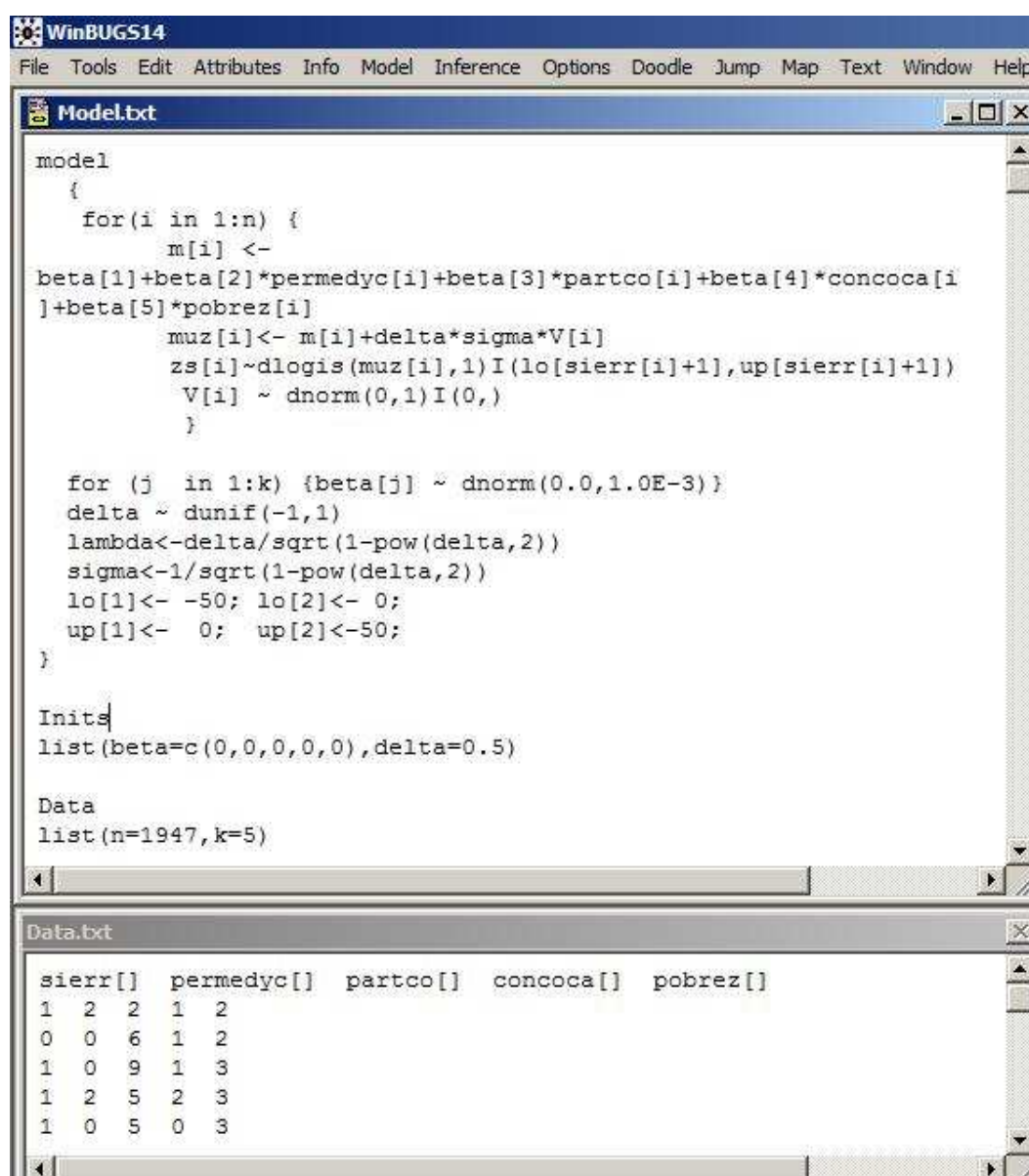


Un video tutorial de como BRMUW puede ser usado para generar sintaxis de diferentes modelos de regresión binaria así como generar sintaxis para datos leídos puede verse <http://videos.pucp.edu.pe/videos/ver/b55ab3b7633c6dab0cad8eec47066e40>.

5.3 Estimación Bayesiana usando WinBUGS u OPENBUGS

Como se ha presentado el BRMUW genera dos archivos, uno que contiene el modelo de regresión binaria con el enlace seleccionado y otro que contiene el conjunto de datos. Ambos archivos en formato txt deben ser abiertos dentro del programa WinBUGS u OpenBUGS para realizar el análisis de inferencia bayesiana correspondiente.

1. Abrir los archivos con la sintaxis del modelo y de los datos previamente generados por el BMRWU en WinBUGS u OpenBUGS.



The screenshot shows the WinBUGS14 interface. The 'Model.txt' window contains the following code:

```
model
{
  for(i in 1:n) {
    m[i] <-
beta[1]+beta[2]*permedyc[i]+beta[3]*partco[i]+beta[4]*concooca[i]
]+beta[5]*pobrez[i]
    muz[i]<- m[i]+delta*sigma*V[i]
    zs[i]~dlogis(muz[i],1)I(lo[sierr[i]+1],up[sierr[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;
}

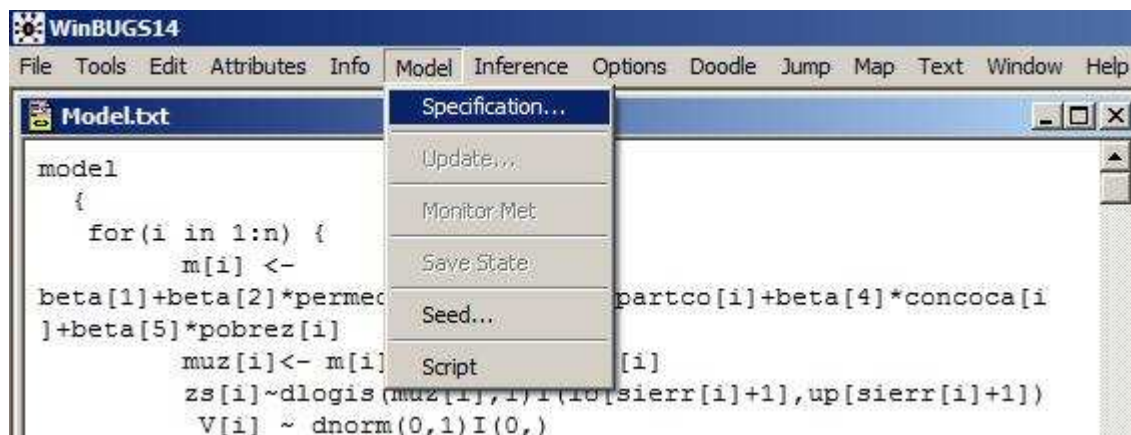
Inits
list(beta=c(0,0,0,0,0),delta=0.5)

Data
list(n=1947,k=5)
```

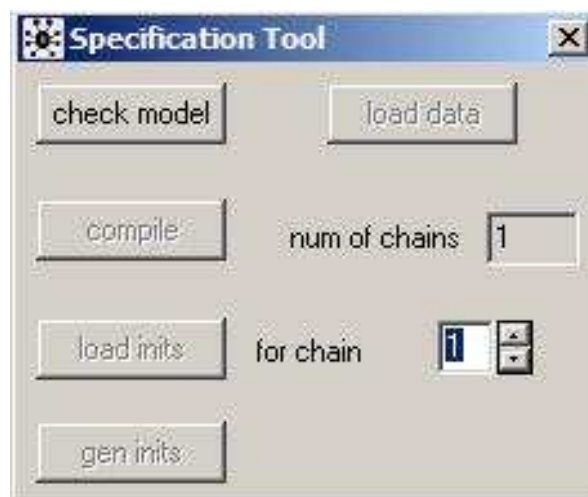
The 'Data.txt' window contains the following data:

sierr[]	permedyc[]	partco[]	concooca[]	pobrez[]
1	2	2	1	2
0	0	6	1	2
1	0	9	1	3
1	2	5	2	3
1	0	5	0	3

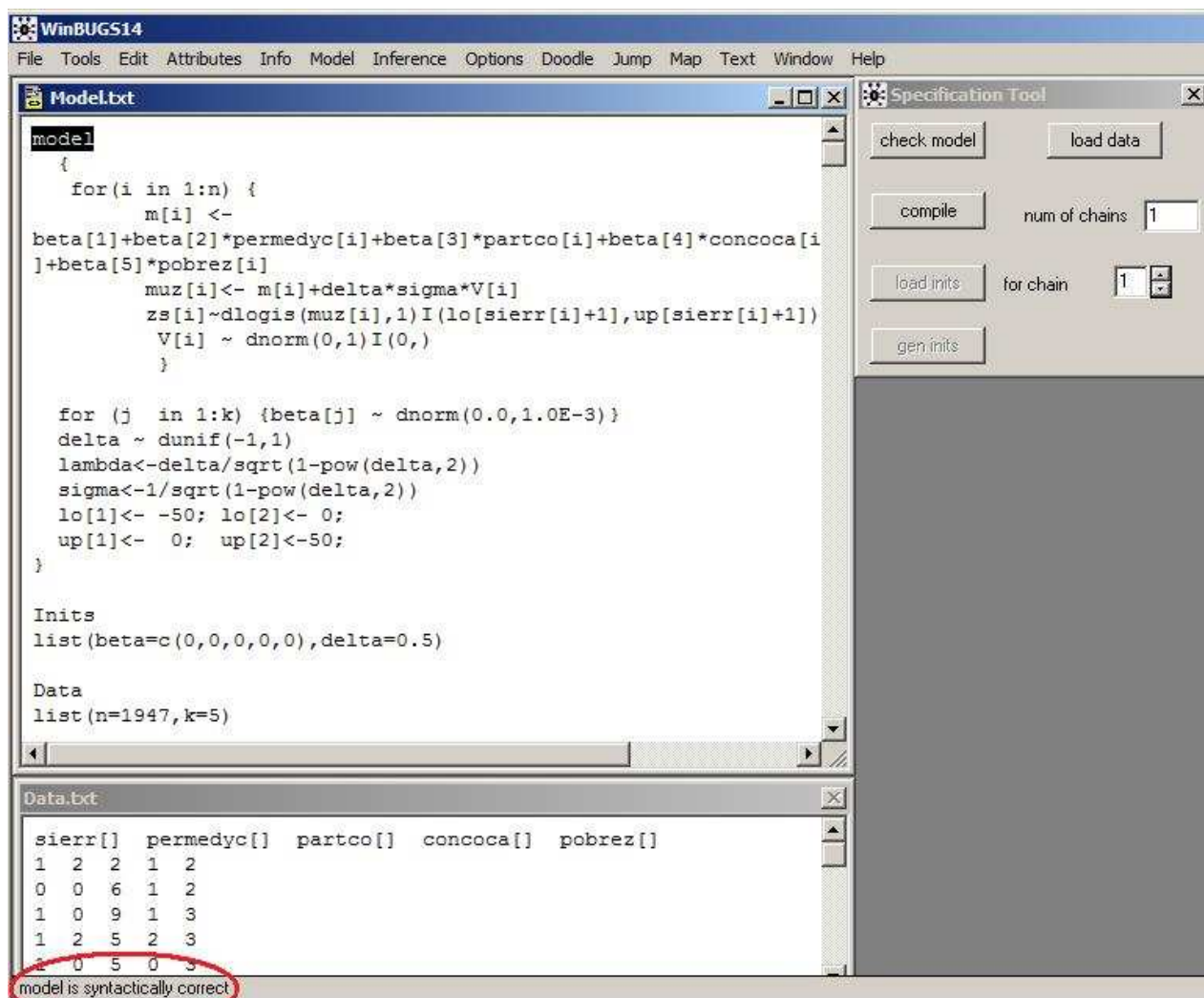
2. Hacer click en *Model > Especification*



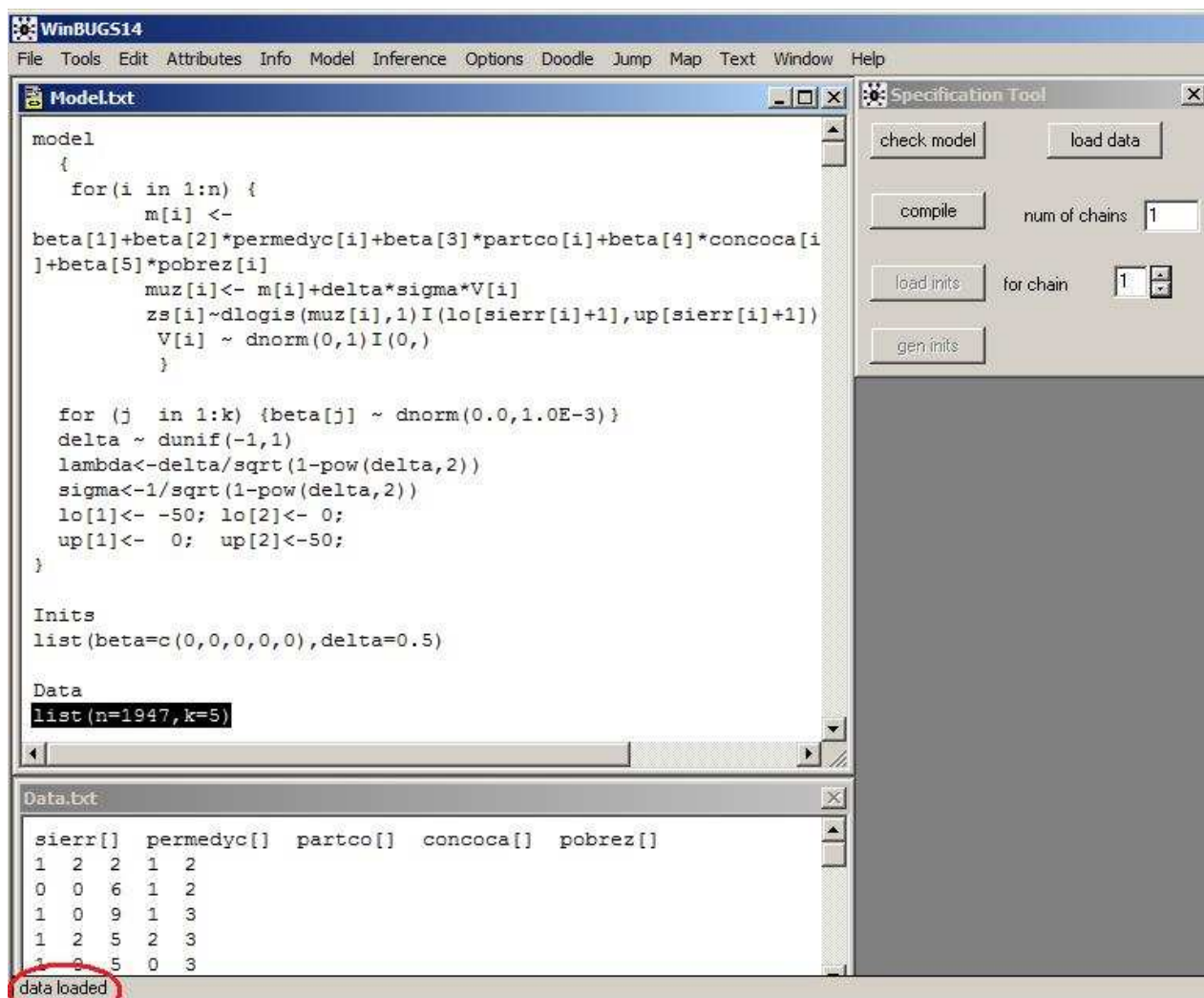
3. Esto abrirá la caja de dialogo "Specification Tool".



4. Seleccionar el modelo y hacer click en *check model*. En la parte inferior izquierda debe aparecer “*model is syntactically correct*” que indica que la sintaxis del modelo esta correctamente formulada.



5. Seleccionar en el archivo del modelo la línea debajo de Data y hacer click en load data. En la parte inferior izquierda aparece “data loaded” indicando que los datos han sido leídos .



6. En el archivo de los datos seleccionar los nombres de las variables y hacer click en load data.

The screenshot shows the WinBUGS14 software interface. The main window displays a model file named 'Model.txt' with the following code:

```
model
{
  for(i in 1:n) {
    m[i] <-
beta[1]+beta[2]*permedyc[i]+beta[3]*partco[i]+beta[4]*concoca[
]+beta[5]*pobrez[i]
    muz[i]<- m[i]+delta*sigma*V[i]
    zs[i]~dlogis(muz[i],1)I(lo[sierr[i]+1],up[sierr[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;
}

Inits
list(beta=c(0,0,0,0,0),delta=0.5)

Data
list(n=1947,k=5)
```

The 'Specification Tool' panel on the right contains the following controls:

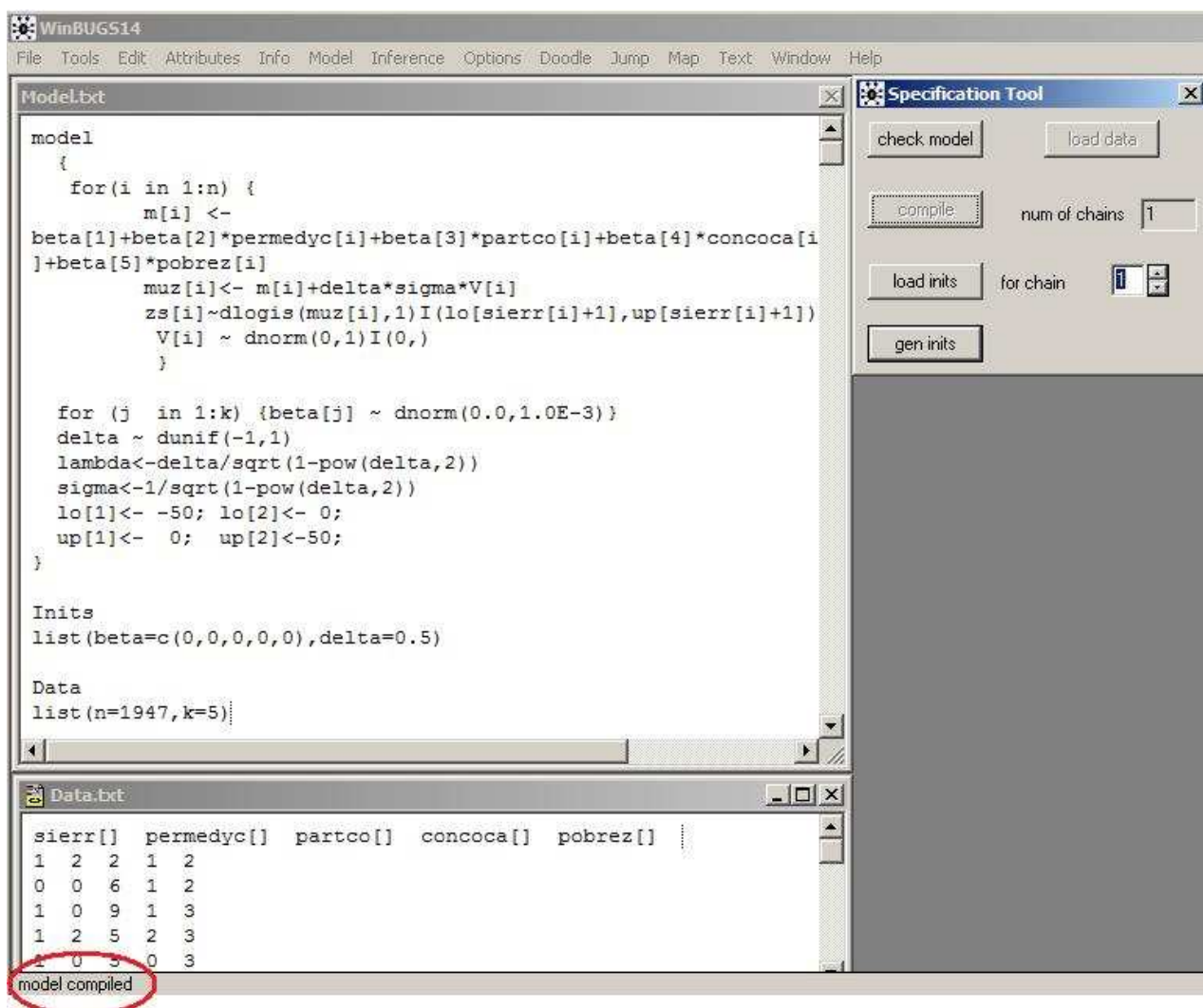
- check model
- load data
- compile
- num of chains: 1
- load inits
- for chain: 1
- gen inits

The 'Data.txt' window shows the following data:

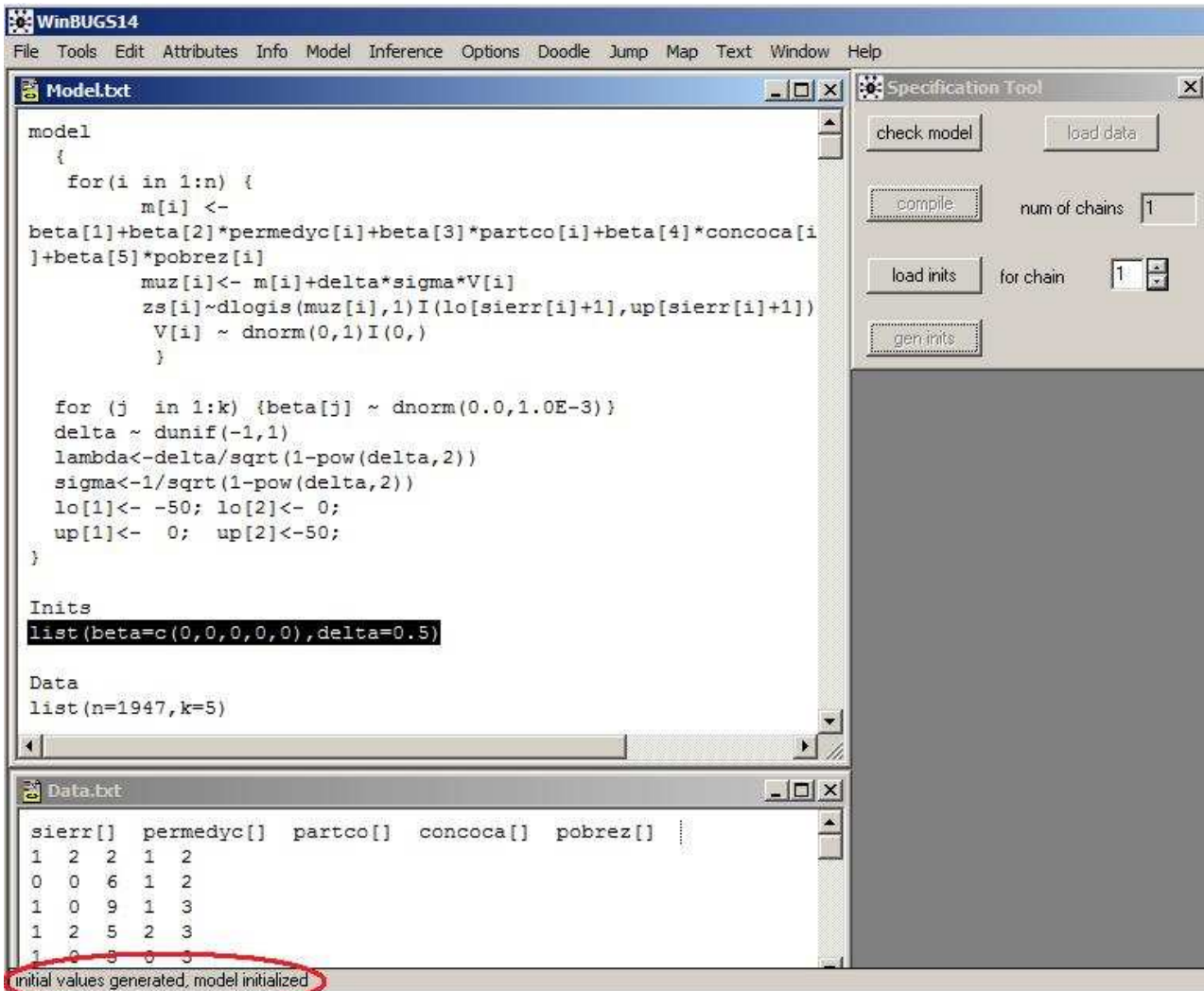
sierr[]	permedyc[]	partco[]	concoca[]	pobrez[]
1	2	2	1	2
0	0	6	1	2
1	0	9	1	3
1	2	5	2	3
1	0	5	0	3

The status bar at the bottom indicates 'data loaded'.

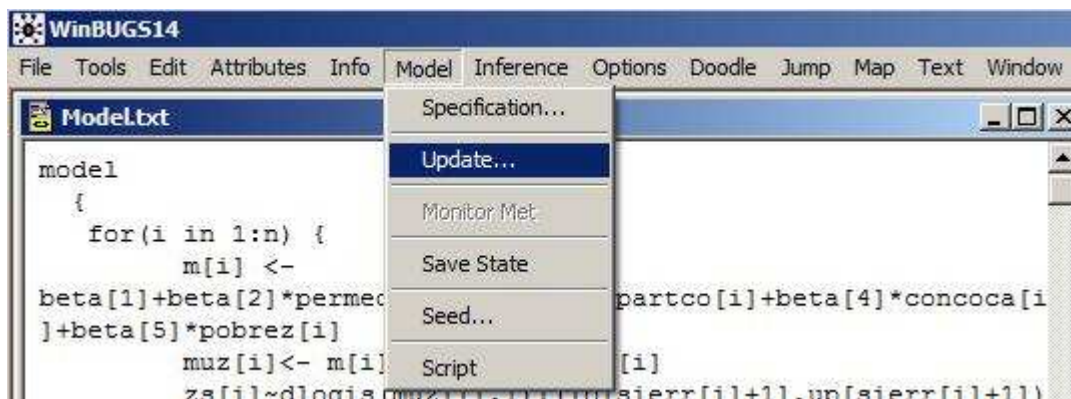
- En la caja de dialogo "Specification tool", se especifica el numero de cadenas que queremos generar en la caja de texto en "num of chains". Una vez especificado el número de cadenas a generar (en este ejemplo 1 cadena) hacer click en *compile*. En la parte inferior debe aparecer "model compiled".



8. Seleccionar la línea debajo de Inits en el archivo del modelo y hacer click en *load inits*. Luego hacer click en *gen inits*. Esto genera los valores iniciales para la Estimación Bayesiana. En la parte inferior debe aparece “*inicial values generated, model initialized*”



9. Hacer click en *Model > Update*



10. Esto abrirá la caja de dialogo “*Update tool*”. En la caja de texto *updates* se ingresa el número de iteraciones que uno requiera y luego se hace click en *update*.



11. Luego debemos especificar que parámetros necesitamos que el programa guarde, para esto vamos a *Inference > Samples*, lo cual abrirá la caja de dialogo “*Sample monitor tool*”. En la caja de texto *node* se escribe el nombre del parámetro y luego se hace click en *set*, esto debe ser hecho para cada parámetro.



12. Repetimos el paso 10 generando mas iteraciones que ahora están siendo guardadas por el WinBUGS u OpenBUGS. En la caja de dialogo “*Sample Monitor Tool*”, podemos calcular estadísticas a posteriori de los parámetros haciendo click en *stats*, un histórico de las cadenas haciendo click en *history*, una estimativa de la densidad a posteriori haciendo click en *density* y otras estadísticas de las cadenas pueden ser calculadas a través de esta caja de dialogo.

Un video tutorial de como WinBUGS puede ser usado para ejecutar la estimación bayesiana de diferentes modelos de regresión binaria puede verse

<http://videos.pucp.edu.pe/videos/ver/db8373ad4703990c51fd196ef2500c9f>

5.4 Estimación Bayesiana usando WinBUGS u OPENBUGS dentro de R

Como hemos visto en la sección anterior con los dos archivos que genera el BRMUW podemos implementar la Estimación Bayesiana con el programa WinBUGS u OpenBUGS.

Alternativamente también se puede implementar la estimación bayesiana utilizando interfaces del R con WinBUGS u OpenBUGS. Para esto necesitaremos el archivo de texto original con los datos en columna:

sierr	permedyc	partco	concooca	pobrez
1	2	2	1	2
0	0	6	1	2
.
.
.
1	2	9	0	3

este archivo lo guardaremos con el nombre datos.txt y del archivo de sintaxis del modelo generado en BRMUW deberemos copiar solo la sintaxis del modelo. En el ejemplo abajo implementamos el modelo de regresión binaria Skew logit.

```

Model.txt
model
{
  for(i in 1:n) {
    m[i] <-
beta[1]+beta[2]*permedyc[i]+beta[3]*partco[i]+beta[4]*concooca[i
]+beta[5]*pobrez[i]
    muz[i]<- m[i]+delta*sigma*V[i]
    zs[i]~dlogis(muz[i],1)I(lo[sierr[i]+1],up[sierr[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;
}

Inits
list(beta=c(0,0,0,0,0),delta=0.5)

Data
list(n=1947,k=5)

```

esto es debemos copiar toda la sintaxis antes de “Inits” y guardarla en un archivo, en este ejemplo modelo.txt. Entonces el archivo modelo.txt quedaría como

```

model
{
  for(i in 1:n) {
    m[i] <-
beta[1]+beta[2]*permedyc[i]+beta[3]*partco[i]+beta[4]*concoca[i
]+beta[5]*pobrez[i]
    muz[i]<- m[i]+delta*sigma*V[i]
    zs[i]~dlogis(muz[i],1)I(lo[sierr[i]+1],up[sierr[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;
}

```

Luego para implementar la estimación bayesiana en el R seguiremos los siguientes pasos para usar la librería R2WinBUGS.

1. Dentro de R, cargar la librería R2WinBUGS con el siguiente comando que ha sido instalada previamente

```
library(R2WinBUGS)
```

2. Leer los datos (el archivo datos.txt en este ejemplo esta en la carpeta C:\BRMUW\)

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

3. Crear una lista que contenga los datos y la información que esta debajo de Data en el archivo generado por el BRMUW

```
Data
list(n=1947,k=5)
```

Esto la hacemos con el siguiente comando

```
data<-c(as.list(datos),list(n=1947,k=5))
```

4. Crear un programa que genere los valores iniciales.

```
Inits
list(beta=c(0,0,0,0,0),delta=0.5)
```



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

5. Finalmente con el comando `bugs` se implementa la estimación bayesiana. Aquí explicaremos brevemente la sintaxis del comando `Bugs`

`parameters.to.save` = es un vector con los nombres de los parámetros del modelo cuyas simulaciones deseamos guardar.

`model.file` = es la dirección donde se encuentra el archivo del modelo.

`n.chains` = es el número de cadenas a ser generadas.

`n.iter` = es el número iteraciones totales de cada cadena.

`n.burnin` = es el número de iteraciones a ser descartadas como *burn-in*.

`program` = es el programa que se utilizará para implementar la estimación bayesiana.

`n.burnin` = es el número de iteraciones a ser descartadas como *burn-in*.

Luego con el siguiente comando se implementa la estimación bayesiana y las simulaciones son guardadas en el objeto `salida`.

```
salida<-bugs(data,inits,parameters.to.save=c("beta","lambda"),
model.file="C:/BRMUW/modelo.txt", n.chains=1, n.iter=2000,
n.burnin=1000,program="OpenBUGS")
```

Si ponemos `salida` en la línea de comandos del R obtenemos un resumen de la simulación

```
> salida
```

```
Inference for Bugs model at "C:/Users/vaio/Documents/Jorge/modelo.txt", fit
using OpenBUGS,
 1 chains, each with 20000 iterations (first 10000 discarded), n.thin = 10
 n.sims = 1000 iterations saved

```

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	-2.8	0.9	-4.2	-3.5	-2.9	-1.8	-1.5
beta[2]	0.7	0.1	0.5	0.6	0.7	0.7	0.8
beta[3]	0.1	0.0	0.1	0.1	0.1	0.1	0.1
beta[4]	-0.2	0.1	-0.3	-0.2	-0.2	-0.2	-0.1
beta[5]	0.5	0.1	0.4	0.4	0.5	0.7	0.8
lambda	0.8	0.9	-0.8	0.0	0.8	1.6	2.3
deviance	4984.5	155.5	4664.7	4862.6	5043.7	5123.0	5151.8

```
DIC info (using the rule, pD = Dbar-Dhat)
pD = -90.0 and DIC = 4894.0
DIC is an estimate of expected predictive error (lower deviance is better).
```

6. Finalmente, para mayores detalles sobre el comando `bugs` se puede consultar la ayuda escribiendo en la línea de comandos

`?bugs`

Referencias

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

Basu, S. and Mukhopadhyay, S. (2000). Binary response regression with normal scale mixtures links, in *Generalized Linear Models: A Bayesian Perspective*, eds. D.K. Dey, S.K. Ghosh, and B.K. Mallick, New York: Marcel Dekker.

Bazán, J., Branco, M. D. y Bolfarine, H. (2006) A skew item response model. *Bayesian Analysis*, **1**: 861- 892.

Bazán, J. L., Bolfarine, H. y Branco, M. D. (2006) A generalized skew probit class link for binary regression. *Technical report* (RT-MAE-2006-05). Department of Statistics. University of São Paulo.

Bazán, J. L. , Bolfarine, H. y Branco, D. M. (2010) A framework for skew-probit links in Binary regression (aceptado para publicacion *Communications in Statistics - Theory and Methods*)

Bazán, J. L., Millones, O. (2008). A classification of binary asymmetric regression models: The use of BRMUW in an application to the decision to eradicate illegal crops of coca leaf. *Simposio Nacional de Probabilidade e Estatística*. SINAPE, São Pedro, Julio 2008.

Bazán, J. L., Millones, O (2008). Una clasificación de modelos de regresión binaria asimétrica: el uso del BAYES-PUCP en una aplicación sobre la decisión del cultivo ilícito de hoja de coca. *Economía* 29 (62), 17-32. PUCP.

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

Chen, M. H., Dey, D. K., y Shao, Q-M. (1999) A new skewed link model for dichotomous quantal response data. *Journal of the American Statistical Association*, 94: 1172-1186.

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

Nagler J. (1994) Scobit: an alternative estimator to logit and probit. *American Journal Political Science*, 38: 230-255.

Prentice, R. L. (1976) A Generalization of the probit and logit methods for dose-response curves. *Biometrika*, 32: 761-768.

Spiegelhalter, D. J., Thomas, A., Best, N. G. y Gilks, W.R. (1996) BUGS 0.5 examples (Vol. 1 Version i). Cambridge, UK: University of Cambridge.

Spiegelhalter, D. J., Thomas, A., Best, N. G., Lunn, D (2007) OpenBUGS User Manual version 3.0.2. MRC Biostatistics Unit, Cambridge.