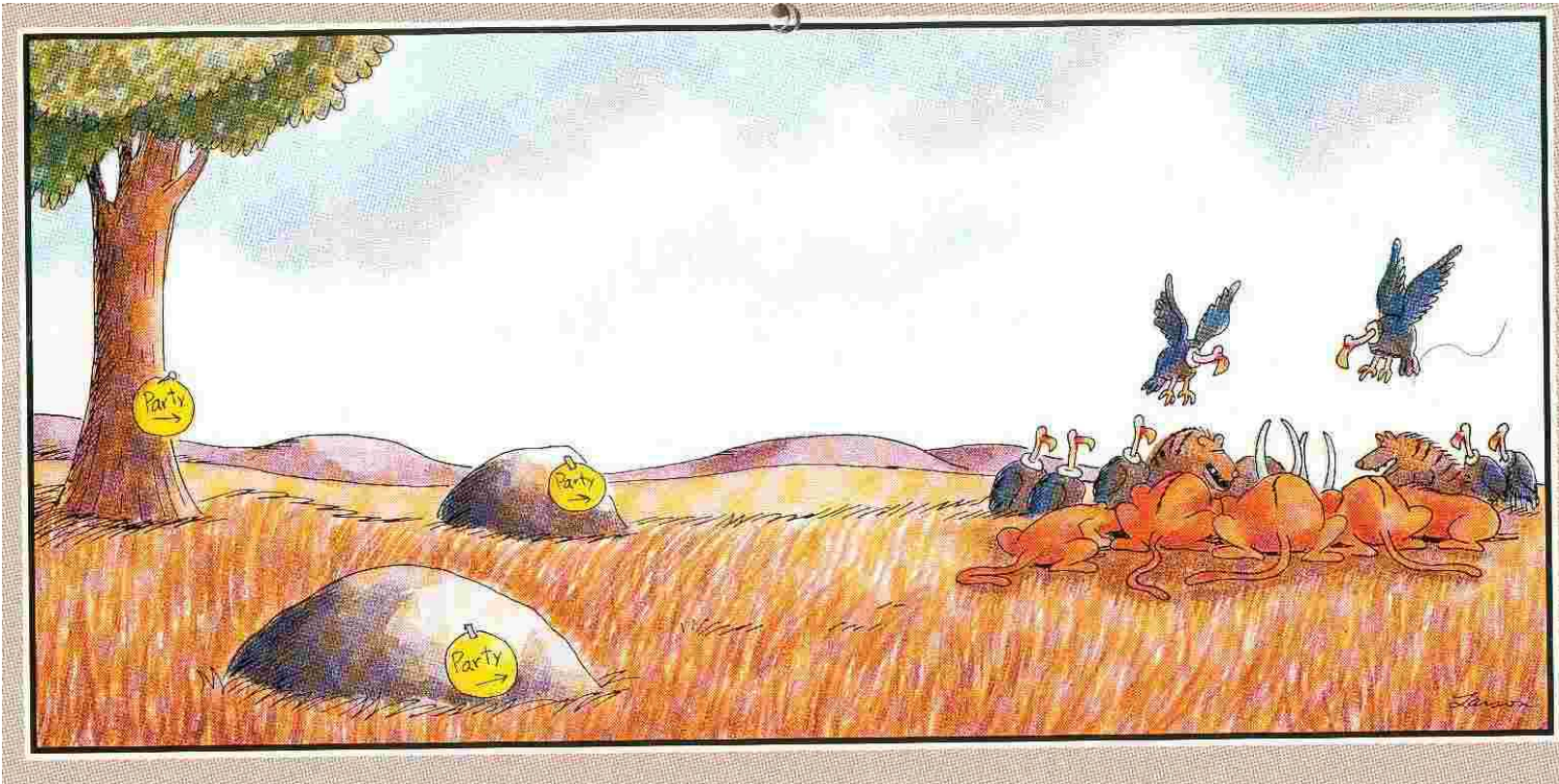


# Loi normale tronquée



# Plan

- 1. Généralités
  - Motivation
  - Définition: troncature vs censure
  - Exemples
- 2. Expression de la densité et inférence
  - Normale réduite et troncature à gauche
  - Cas Général
  - Estimation
- 3. Simulation
  - Acceptation-Rejet
  - Completion
- 4. Traitement en Winbugs
  - Opérateur  $d_{\text{norm}}(\cdot, \cdot)I(a, b)$  et opérateur spécifique
  - Données tronquées, données censurées
- 5. Annexe

# 1-Motivation

Date: Fri, 18 Sep 2009 21:55:28 +0100  
Reply-To: Dave Lunn <david.lunn@MRC-BSU.CAM.AC.UK>  
Sender: "(The BUGS software mailing list)" <BUGS@JISMAIL.AC.UK>  
From: Dave Lunn <david.lunn@MRC-BSU.CAM.AC.UK>

Hi all,

I thought I'd try and clarify matters on the **I(.) construct/notation**. The main reason for the **confusion** appears to be the **subtle difference between truncation and censoring**. Put simply, **truncation is a modelling assumption**, i.e. it represents a belief about the system being modelled, such **as measured drug concentrations should be positive**. **Censoring, on the other hand, represents an observation, e.g. some drug concentration has been observed to be less than 1 mg/L, say**. A simple way to determine which assumption is appropriate in a given setting is to think "would I want to make the same assumption if predicting a new observation?" If yes, then you probably want to truncate; if no, you probably want to use I(.). If truncation is the appropriate assumption, i.e. you wish to actually apply some kind of constraint, then, in general, you should *\*not\** use I(.). However, as discussed below, you can get away with it in certain circumstances. **First, though, I'll illustrate what the problem is...**

# 1-Motivation

for (i in 1:n) {theta[i] ~ dnorm(mu, tau)}

where mu and tau are unknown parameters, and the thetas should all be positive, not because they have been observed to be positive (when negative values are permissible) but because negative values simply don't make sense in the given setting. Hence truncation is appropriate (we would also want to apply the same constraint if predicting a new theta). **If one applies I(0, ) to the distribution above, this simply has the effect of ensuring that no negative values will be sampled for any of the thetas, which is what we want in either case! But, now consider what happens when sampling mu and tau... First note that BUGS does not modify the sampling distributions for mu and tau in any way if we use I(.) above. If the appropriate assumption is truncation, then the correct contribution from each theta[i] to the likelihoods for mu and tau is not dnorm(mu, tau) but dnorm(mu, tau)/const, where const is the integral of dnorm(mu, tau) between theta = 0 and theta = infinity (so that the density for each theta integrates to 1). This normalizing constant (const) is a complex function of both mu and tau, but BUGS is ignoring this (potentially) important contribution to the full conditional distributions of mu and tau. Hence the sampling for mu and tau will be wrong!**

So, back to when we can get away with using I(.) for truncation... Basically, this is ok when the parameters of the distribution to be truncated are known, because it is only they that would otherwise have been sampled incorrectly - if they are known then they are not sampled anyway.

Best wishes,  
Dave Lunn

# 1-Troncature vs Censure

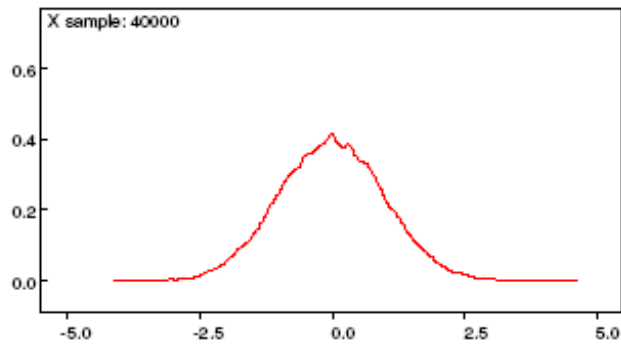
**Truncation:** A truncated sample can be thought of as being equivalent to an underlying sample with **all values outside the bounds entirely omitted**

$$X = (X_i)_{1 \leq i \leq N} \quad X_i \geq c, \forall i$$

**Censoring:** If the sample had been censored, **a record would be of those that were censored, consisting of a note that of whether the lower or upper bound had been passed** and the value of the bound.

$$T_i = \max(X_i, c) \quad \delta_i = I [X_i < c] \quad \text{Observed}$$

# 1-Troncature vs Censure



	var	mean	sd	sample
X		-0.003677	0.9992	40000
Z		0.2855	0.7943	40000
Y		0.07939	0.8667	40000

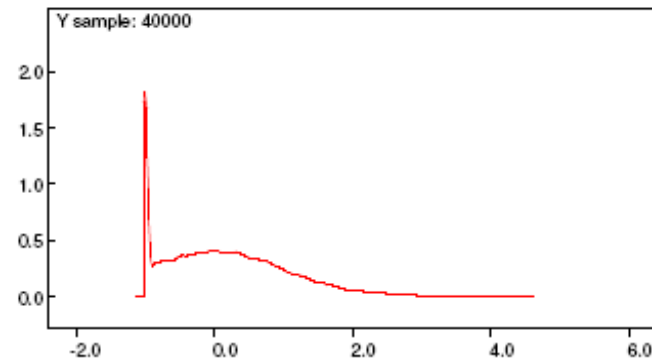
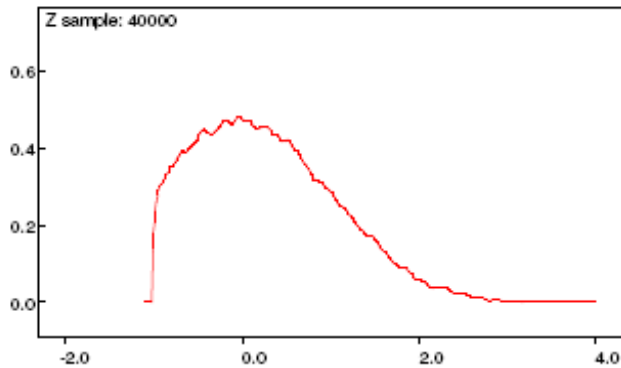
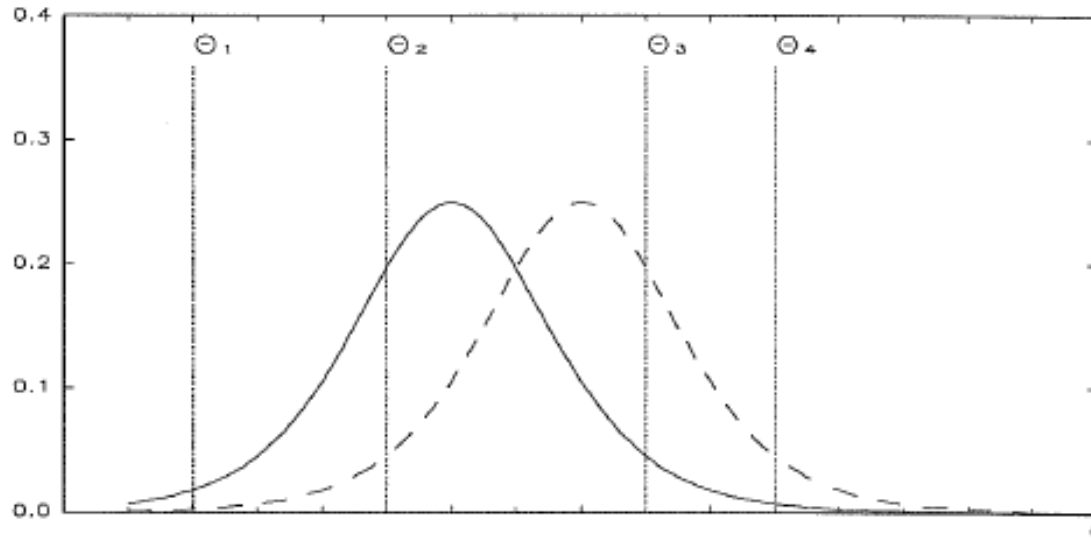


Fig : Densité de loi normale :  $X \sim N(0,1)$  Z tronquée à -1.5, Y censurée à -1.5

# 1. Exemples

1. Sélection directionnelle à seuils (T)
2. Variables ordinales (T)
3. Modélisation d'une fréquence allélique (T)  
modèle Nicholson basé sur une loi normale
4. Limite de détection: LOQ (C)  
Charge virale non mesurable en dessous d'un seuil  
Balance plafonnée  
Durée d'observation fixée

# Données ordinales et variable latente



Assume the "latent variable"  $X_i \sim \mathcal{N}(\eta_i, \sigma^2)$  with fixed thresholds  $\tau_1, \tau_2$

$$\Pr(Y_i = 1) = \pi_{i1} = \Pr(X_i \leq \tau_1) = \Pr\left(\underbrace{\frac{X_i - \eta_i}{\sigma}}_{\mathcal{N}(0,1)} \leq \frac{\tau_1 - \eta_i}{\sigma}\right) = \Phi\left(\frac{\tau_1 - \eta_i}{\sigma}\right)$$

$$\Pr(Y_i = 2) = \pi_{i2} = \Pr(\tau_1 < X_i \leq \tau_2) = \Phi\left(\frac{\tau_2 - \eta_i}{\sigma}\right) - \Phi\left(\frac{\tau_1 - \eta_i}{\sigma}\right)$$

$$\Pr(Y_i = 3) = \pi_{i3} = \Pr(\tau_2 < X_i \leq +\infty) = 1 - \Phi\left(\frac{\tau_2 - \eta_i}{\sigma}\right)$$

$$X_i | Y_i = k \sim N_T(\eta_i, \sigma^2; \tau_{k-1}, \tau_k)$$



## 2. Normale réduite et troncature à gauche

$X \sim N(0,1)$  avec  $X > a$

$$f(x, 0, 1; a, \infty) = I_{(x>a)} \frac{\phi(x)}{\Pr(X > a)} = \boxed{I_{(x \geq s)} \frac{\phi(x)}{1 - \Phi(a)}}$$

$\phi(x)$  = densité normale réduite;  $\Phi(x)$  = f de répartition

$$E(X | X > a) = \frac{\phi(a)}{1 - \Phi(a)} = \frac{\phi(a)}{\Phi(-a)} = \underbrace{\lambda(-a)}_{\text{Mills ratio}}$$

$$\text{Var}(X | X > a) = 1 - \lambda(-a) [\lambda(-a) - a]$$

## 2. Cas général: Normale et double troncature

$X \sim N(\mu, \sigma^2)$  avec  $a < X \leq b$

$$f(x) = I_{[a < X \leq b]} \frac{\sigma^{-1} \phi\left(\frac{x - \mu}{\sigma}\right)}{\Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right)}$$

$$E(X \mid a < X \leq b) = \mu + \sigma \frac{\phi(a^*) - \phi(b^*)}{\Phi(b^*) - \Phi(a^*)}$$

## 2. Inférence

*Logvraisemblance* :  $L(\mu, \sigma^2; x_i > s, 1 \leq i \leq N)$

$$\begin{aligned} -2L &= N \log 2\pi + N \log \sigma^2 + \sum_{i=1}^N \frac{(x_i - \mu)^2}{\sigma^2} \\ &\quad + 2N \log \Phi\left(\frac{\mu - s}{\sigma}\right) \end{aligned}$$

$\sigma$  connu

$$\frac{d[-2L(\mu; x_i > s)]}{d\mu} = 2\sigma^{-2} \left[ -\sum_{i=1}^N (x_i - \mu) + \sigma N \lambda\left(\frac{s - \mu}{\sigma}\right) \right]$$

$$\hat{\mu} = \bar{x} - \sigma \lambda\left(\frac{s - \hat{\mu}}{\sigma}\right) \Leftrightarrow \bar{x} = \hat{\mu} + \sigma \lambda\left(\frac{s - \hat{\mu}}{\sigma}\right)$$

### 3. Simulation/Algorithme Acceptation-Rejet

CP Robert (1995) *Statistics & Computing* 5,121-125; J Geweke (1991)

Standard (left) truncated normal  $\mu_{\min}$

1) Draw  $z$  from a translated exponential

Remember :  $g(z^*; \alpha^*) = \alpha^* \exp(-\alpha^* z^*)$

$z = \mu_{\min} + z^*$  with  $z^* \sim \text{Exp}(\alpha^*) = -(\ln U_{0-1}) / \alpha^*$

2) Let  $\lambda(z) = \exp\left[-\frac{1}{2}(z - \alpha^*)^2\right]$  where  $\alpha^* = \frac{\mu_{\min} + \sqrt{\mu_{\min}^2 + 4}}{2}$

3) Draw  $u$  from  $U_{0-1}$

4) If  $u \leq \lambda(z)$ , keep  $z$ , otherwise go back to 1)

$\mu_{\inf}$	0	0.5	1	1.5	2	2.5	3
$P_{\text{accept}}$	0.76	0.83	0.88	0.91	0.93	0.95	0.96

## 4. Traitement Winbugs/Loi Tronquée/Exemple

Modèle

$$X \sim N(\mu, \sigma^2) \text{ avec } a < X \leq b$$

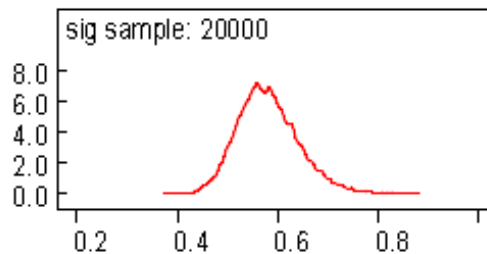
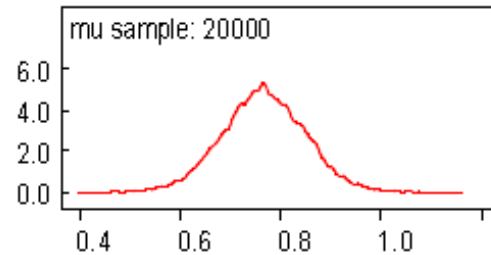
$$\pi(\mu, \sigma^2) \propto \sigma^{-2}$$

## 4. Traitement en Winbugs/Opérateur I(.,.)

Inference on the mean and variance of a truncated normal distribution  
N=50 data obtained from generating N values from  $N(0,1)$  truncated on the left  
( $X>0$ )

```
model {  
  for (i in 1:K) {  
    X[i]~dnorm(mu,tau)I(0,)  
  }  
  mu~dunif(-7,7)  
  tau~dgamma(0.001,0.001)  
  var<-1/tau  
  sig<-pow(tau,-0.5)  
}
```

## 4. Traitement en Winbugs/Opérateur I(.,.)



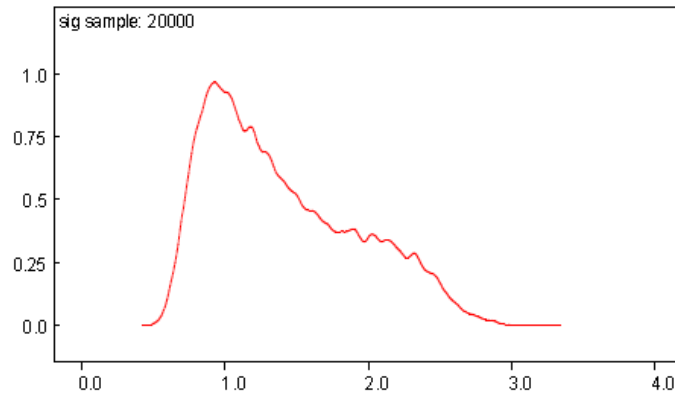
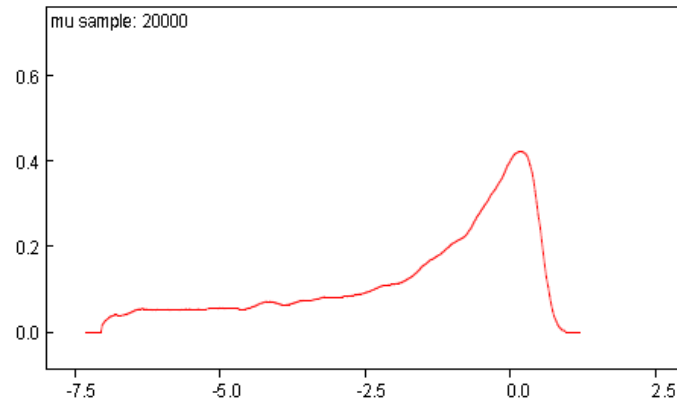
<b>node</b>	<b>mean</b>	<b>sd</b>	<b>MC error</b>	<b>2.5%</b>	<b>median</b>	<b>97.5%</b>	<b>start</b>	<b>sample</b>
mu	0.7661	0.08216	6.5E-4	0.6035	0.7658	0.9273	5001	20000
sig	0.5807	0.06002	4.361E-4	0.4768	0.576	0.7134	5001	20000
<b>node</b>	<b>true</b>							
mean	0.7979							
sd	0.6028							

## 4. Traitement en Winbugs/Fonction spéciale

```
model {  
  for (i in 1:K) {  
    X[i]~djl.dnorm.trunc(mu,tau,0,20)  
    #X[i]~dnorm(mu,tau)|(0,)  
  }  
  mu~dunif(-7,7)  
  tau~dgamma(0.001,0.001)  
  var<- 1/tau  
  sig<-pow(tau,-0.5)  
}
```



## 4. Traitement en Winbugs/Fonction spéciale



node	mean	sd	MC error	2.5%	median	97.5%	mode	start	sample
mu	-1.623	2.011	0.07784	-6.414	-0.9032	0.546	0.088	1001	20000
sig	1.404	0.5213	0.02037	0.6952	1.284	2.506	0.952	1001	20000
var	2.243	1.658	0.0635	0.4833	1.649	6.279	0.862	1001	20000

#### 4. Traitement Winbugs/Données censurées/Exemple

Modèle: Latente  $X_i$ ; Observée  $Y_i$

$$1) c_i = 0, Y_i = X_i, i \in C_0$$

$$2) c_i = 1, Y_i = s, X_i \leq s, i \in C_1 \text{ (censure)}$$

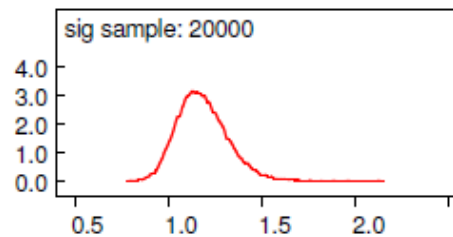
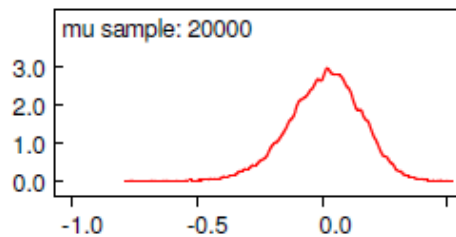
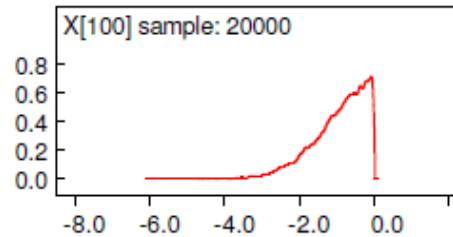
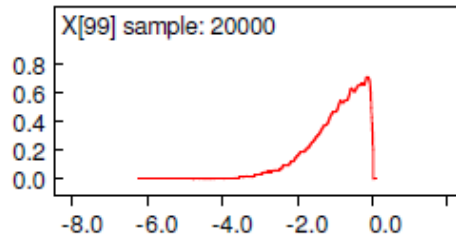
$$X_i \sim N(\mu, \sigma^2) \quad \pi(\mu, \sigma^2) \propto \sigma^{-2}$$

## 4. Traitement Winbugs/Données Censurée/Exemple

```
model {
  for (i in 1:K1) {
    X[i]~dnorm(mu,tau)
  }
  for (i in K1+1:K) {
    X[i]~dnorm(mu,tau)I(,0)
  }
  mu~dunif(-7,7)
  tau~dgamma(0.001,0.001)
  var<-1/tau
  sig<-pow(tau,-0.5)
}
data
list(K=100,K1=52,
X=c(
1.493,1.103,2.183,2.431,0.6758,0.3989,0.7582,0.7711,0.8094,1.83
9,0.5311,0.3288,0.07549,0.1841,0.8946,1.301,1.027,0.2237,1.286,
0.3767,0.3312,0.7693,0.2375,1.257,3.298,0.3022,0.3976,2.318,1.6
35,0.09756,2.173,0.2655,0.9724,0.407,1.645,1.274,0.1394,0.1265,
1.034,1.787,1.651,0.4206,0.3495,0.1654,0.6914,1.987,0.4703,0.24
,0.3701,0.586,0.1293,0.01025,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,N
A,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,N
```

# 4. Traitement Winbugs/Données Censurées

Résultats: Censure à gauche  $Y > 0$   $N=100$  ( $N_0=52$ ,  $N_1=48$ )



X[97]	-0.9356	0.7238	0.005861	-2.688	-0.7821	-0.03552	1	20000
X[98]	-0.9419	0.7316	0.005692	-2.743	-0.7796	-0.03635	1	20000
X[99]	-0.9393	0.7299	0.005793	-2.756	-0.7844	-0.03411	1	20000
X[100]	-0.9402	0.7257	0.005903	-2.678	-0.7883	-0.03499	1	20000
mu	<b>0.009729</b>	0.1448	0.001681	-0.2991	0.01755	0.2732	1	20000
sig	<b>1.183</b>	0.1331	0.00194	0.9574	1.171	1.474	1	20000

## 4. Traitement Winbugs/Données Censurées/Interprétation

Dans un cadre EM Stochastique (Celeux & Diebolt, 1982),  
les statistiques exhaustives de  $\mu$  et  $\sigma^2$  sont

$$\sum_{i=1}^N \tilde{y}_i \text{ et } \sum_{i=1}^N \tilde{y}_i^2$$

$$\tilde{y}_i = y_i^{obs} \text{ si } i \in C_0$$

$$\tilde{y}_i \text{ remplacé par } x_i^{(t)} \sim \mathcal{N}(\mu^{(t)}, \sigma^{2(t)}) I_{[x_i < a]} \text{ si } i \in C_1$$

## 4. Winbugs 1.4 /Fonction spéciale

### WinBUGS Development web-site

Welcome to the WinBUGS Development web-site! This is a new site designed to facilitate the distribution of various extensions of the WinBUGS software. The following features are available:

- The [WinBUGS Differential Interface \(WBDiff\)](#) allows the user to specify complex functions via arbitrary systems of ordinary differential equations. **No bugs have been reported during the beta-testing phase (February -- September, 2004), and so the final version has now been released -- please follow the link to download.**
- Implement your own (hard-wired) specialized functions and distributions within WinBUGS 1.4 -- a **new** version (**as of September 2004**) of the [WinBUGS Development Interface \(WBDev\)](#) is available to download now.
- Our new [Pharmaco](#) interface, together with [WBDiff](#), constitutes Version 2.0 of the [PKBugs](#) software. Please see [here](#) for details. **Note: a bug has been identified which caused some multiple-dosing intravenous-infusion regimens to be specified incorrectly. The current version of Pharmaco includes a fix for this bug -- please follow the link for details.**
- Download specialized components written by other WinBUGS users -- see the [WBDev shared components page](#).
- I am currently working on a suite of reversible jump MCMC components for WinBUGS. This includes algorithms for automatic curve fitting and covariate selection, for example. A beta-release of the [Jump](#) interface can be downloaded from [here](#).
- Links to both the [BUGS](#) and the [PKBugs](#) web-sites are provided for your convenience.

## 5. Annexe/Simulation/Algorithme de completion

Robert & Casella (1999) MC Statistical Methods, Springer

Foulley (2000) GSE, 32, 631-635

Loi normale  $N(0,1)$  tronquée à gauche à  $\mu_{\text{inf}} = s$

$X \sim \mathcal{N}(0,1), X > s$  a une densité  $\propto f(x) = h(x)I_{[x \geq s]}$

où  $h(x) = \exp(-x^2 / 2)$

L'idée est de simuler  $X$  à partir de la loi conjointe de  $(X, Z)$ .

Ici, on peut bâtir une loi conjointe comme suit qui a pour densité

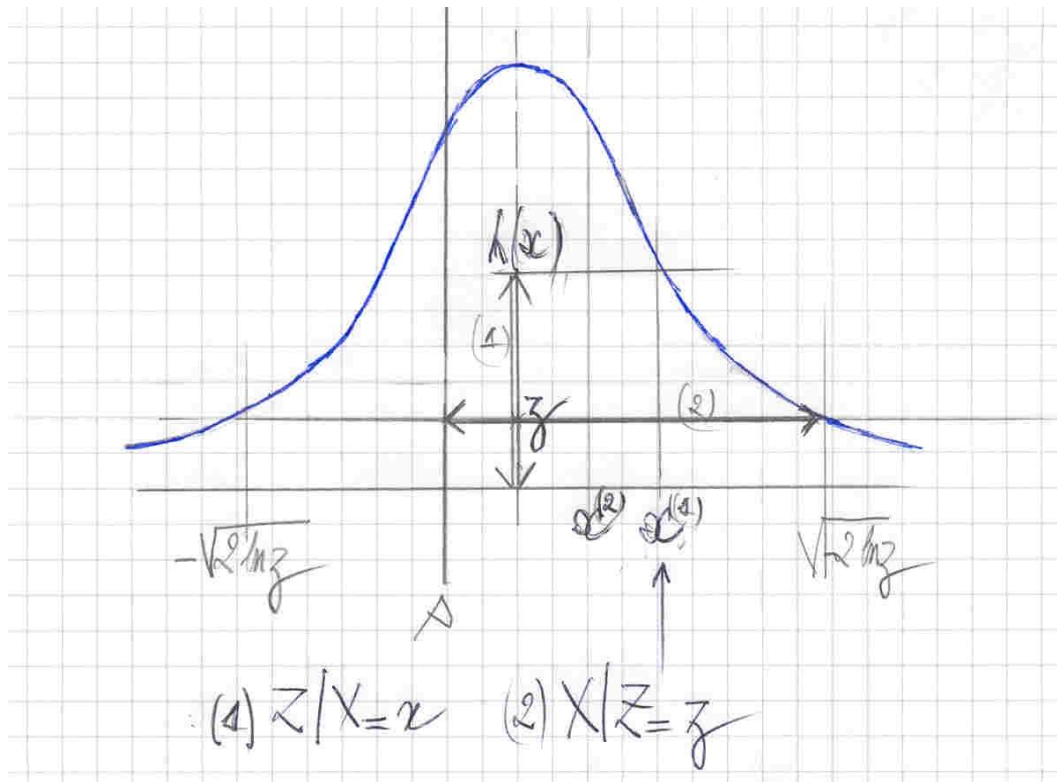
$$g(x, z) = I_{[x \geq s]} I_{[0 < z \leq h(x)]}$$

On vérifie que  $\int g(x, z) dz = f(x)$

En effet

$$\int g(x, z) dz = I_{[x \geq s]} \int I_{[0 < z \leq h(x)]} dz = I_{[x \geq s]} \int_0^{h(x)} dz = h(x) I_{[x \geq s]}$$

## 5. Annexe/Simulation/Algorithme de completion



On génère  $(X, Z)$  par Gibbs

$$1) \boxed{Z|X = x \sim U[0, h(x)]}$$

$$2) X|Z = z \sim U[(x \geq s) \cap (h(x) \geq z)]$$

$$\boxed{X|Z = z \sim U[\max(s, -\sqrt{-2 \ln z}), +\sqrt{-2 \ln z}]}$$