

# Week 8/9 - Applied Bayes

February 28, 2011 — Michael Escobar

## Model Checking<sup>(1)</sup>

### Introduction

- Basically good applied statisticians check their assumptions.
  - J. R. Savage : "Applied statisticians are statistician who take their Assumptions Seriously!"
- ⇒ So, we will discuss ways of looking at the data, the model, and the results to see if everything makes sense.
- ⇒ Some of the objectives :
- Ⓐ look for unusual observations .
  - Ⓑ Comparison of different models
  - Ⓒ General Goodness of fit .

<sup>(1)</sup> follows closely Bugs Manual Version 5 (1996) pg 40-47

NB: Is this Bayesian? ~~Presently we~~ Presently we  
 will look ~~at~~ at more ad hoc methods. It is  
 possible to make this more formal. Next week  
 we look at more formal Bayesian methods  
 for model selection. ~~There are~~ There are  
 ways to put our "Beliefs" on the "Mathematical  
 form of the ~~model~~ model". But this is too advanced  
 for this ~~course~~ course.

---

### Look my at individual observations

Gelfand (1992) suggests:

$$\textcircled{1} \text{ Residuals: } y_i - E(y_i)$$

$$\textcircled{2} \text{ Standardized Residual: } \frac{(y_i - E(y_i))}{\sqrt{V(y_i)}}$$

\textcircled{3} The "chance of getting a more extreme obs."  $\min \left[ P(\bar{Y}_i < y_i), P(\bar{Y}_i > y_i) \right]$

\textcircled{4} Chance of a More Surprised obs.  $P(\bar{Y}_i | p(\bar{Y}_i) \leq p(y_i))$

\textcircled{5} The predictive ordinate of the observations:  $p(y_i)$

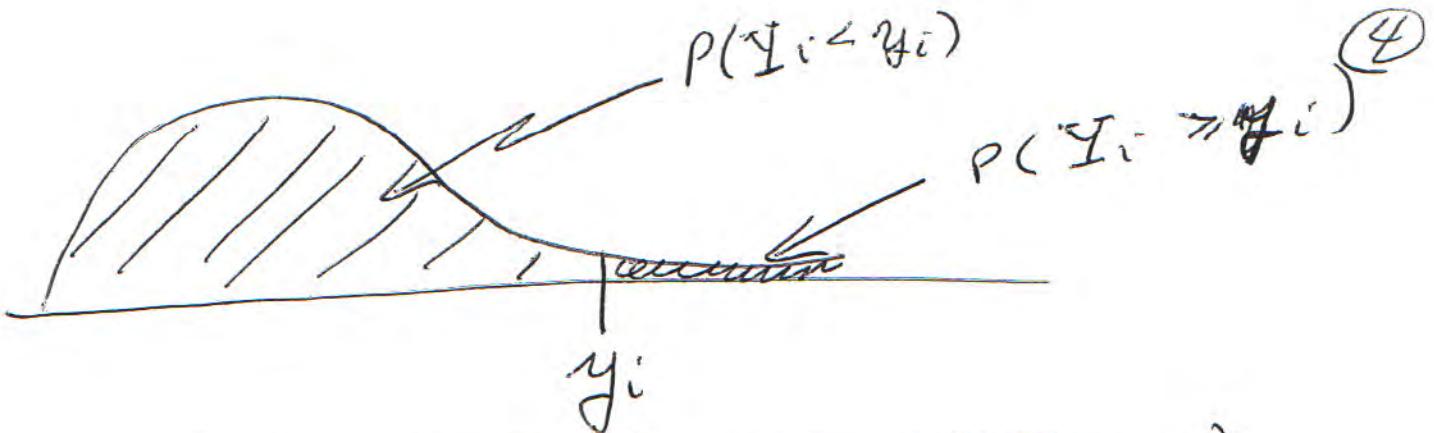
where  $y_i$  is the observation. ③

- The probability  $P(\cdot)$ ,  $p(y_i)$  are the Predictive Distributions of  $y_i$ .
- $E(y_i)$ ,  $V(y_i)$  are the expected value (mean) and variance of the random variable  $y_i$ .



Some Comments:

- for ① and ②, we will need to "calibrate" these statistics to know what's good or bad.
- for ③: chance of getting a more extreme observation  $\min(p(Y_i \leq y_i), p(Y_i \geq y_i))$ ,  
This is similar to a 2-sided p-value.  
That is, you look at the two tail areas and take the one that is smaller. This is the probability of getting a more extreme observation.



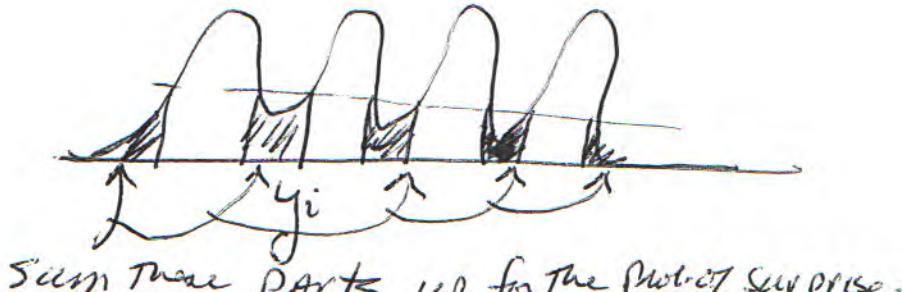
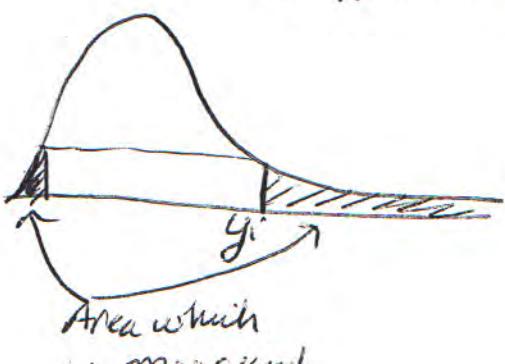
So here the smaller value is  $P(Y_i \geq y_i)$ .

- ④ "Surprise" observation:

$$P(Y_i) p(Y_i) \leq p(y_i)$$

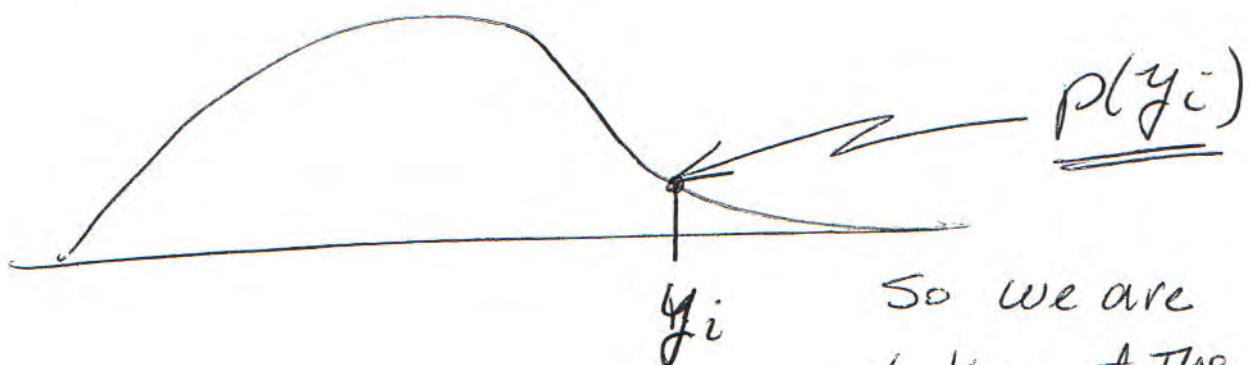
We look at the set of all  $Y$ 's where the value of the probability density is smaller than value of the probability density at the observed value.

For a Unimodal density, this will be the same as #3: The <sup>2-sided</sup>  $p$ -value. There is a potential difference when the density is Multi-modal.



(5)

- ⑤ The predictive ordinate:



So we are looking at the density value.

You do not expect many of your observed values to have low predictive ordinates → many should be in the mode...

Now what are the Distributions based on?

- ⑥ 2 data sets → one for "Training" and the other for "Testing".

So, use the Training set to get the distribution. (posterior based on training) and then look at set of  $(y_1, \dots, y_n)$  in test and calculate the above statistics.

(6)

(b) "Leave one out" Analysis.  
 (Cross-validation). Call the set  $\bar{y}_{\setminus i}$  to be the set of  $y$ 's where the  $i^{\text{th}}$  value ( $y_i$ ) is ~~left out~~  
 left out. So calculate the statistic for  $y_i$  based on posterior distribution from data =  $\boxed{\bar{y}_{\setminus i}}$ .

- One way to do this (which is a ~~difficult~~) is to run  $n$  different analyses, each with one data point left out. In WinBugs, this will drive you crazy.
- Alternatively there is a trick:

$$\begin{aligned} \overline{p(y_i | \bar{y}_{\setminus i})} &= E_{\theta} \left[ \frac{1}{n} p(y_i | \theta_k) \mid \bar{y}_{\setminus i} \right] \\ &= \operatorname{Avg}_{\theta \sim \theta_k | \bar{y}_{\setminus i}} \left[ \frac{1}{n} p(y_i | \theta_k) \right] \end{aligned}$$

For more details  $\rightarrow$  see Attached Bugs 0.5 manual.

Not quite formal Bayes methods  
 of model comparison =>

Negative Cross-Validated log likelihood

$$-\sum \log [P(y_i | y_{-i})]$$

→ compare different models by taking difference

N.B: Also called

Pseudo Bayes Factor

measures the overall goodness of fit

Deviance:  $E\left[-2 \sum \log (\text{likelihood}) \middle| \text{DATA}\right]$

Pearson Type  $E\left(\sum_{i=1}^n \frac{y_i - E(y_i)}{\sqrt{V(y_i)}} \middle| \text{data}\right)$

## Worked Example

Basic Problem = DATA from Pump P

Example in Win bugs

Example I Manual.

Overview = DATA from pump failures. Number of failures,  $(X_i)$ , follows a Poisson distribution. Have data on length of operation,  $(t_i)$ . So have

$X_i \sim \text{Poisson}(\theta_i t_i), i = 1, \dots, 10$   
Let  $\theta_i t_i = \lambda_i$

(See Sheet from Manual for more details)

### 3 Models considered

"conjugate" model:  $\theta_i \sim \text{Gamma}(\alpha, \beta)$   
 $\alpha \sim \text{Exponential}(1)$   
 $\beta \sim \text{Gamma}(0.1, 0.1)$

SLM Constant:  $\theta_i t_i = \lambda_i$   
 $\log(\lambda_i) = \mu + \log(t_i)$   $\mu \sim \text{Normal}$

SLM VARYing:  $\log(\lambda_i) = \beta_i + \log(t_i)$



## BUGS Pumps: conjugate gamma-Poisson hierarchical model

George et al (1993) discuss Bayesian analysis of hierarchical models where the conjugate prior is adopted at the first level, but for any given prior distribution of the hyperparameters, the joint posterior is not of closed form. The example they consider relates to 10 power plant pumps. The number of failures  $x_i$  is assumed to follow a Poisson distribution

$$x_i \sim \text{Poisson}(\theta_i t_i) \quad i = 1, \dots, 10$$

where  $\theta_i$  is the failure rate for pump  $i$  and  $t_i$  is the length of operation time of the pump (in 1000s of hours). The data are shown below.

Pump	$t_i$	$x_i$
1	94.5	5
2	15.7	1
3	62.9	5
4	126	14
5	5.24	3
6	31.4	19
7	1.05	1
8	1.05	1
9	2.1	4
10	10.5	22

A conjugate gamma prior distribution is adopted for the failure rates:

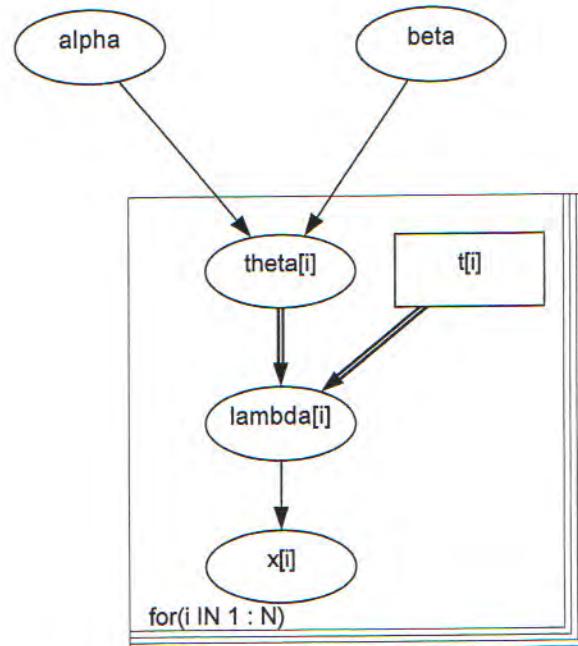
$$\theta_i \sim \text{Gamma}(\alpha, \beta), \quad i = 1, \dots, 10$$

George et al (1993) assume the following prior specification for the hyperparameters  $\alpha$  and  $\beta$

$$\begin{aligned}\alpha &\sim \text{Exponential}(1.0) \\ \beta &\sim \text{Gamma}(0.1, 1.0)\end{aligned}$$

They show that this gives a posterior for  $\beta$  which is a gamma distribution, but leads to a non-standard posterior for  $\alpha$ . Consequently, they use the Gibbs sampler to simulate the required posterior densities.

*Graphical model for pump example:*



*BUGS language for pump example:  
Conjugate Model*

```

model
{
  for (i in 1 : N) {
    theta[i] ~ dgamma(alpha, beta)
    lambda[i] <- theta[i] * t[i]
    x[i] ~ dpois(lambda[i])
  }
  alpha ~ dexp(1)
  beta ~ dgamma(0.1, 1.0)
}
  
```

[ Since Gamma  
is Conjugate  
Prior for Poisson ]

Data → click on one of the arrows to open the data ←

Init → click on one of the arrows to open the initial values ←

## Results

A burn in of 1000 updates followed by a further 10000 updates gave the parameter estimates:

→ Model for GLM Constraint ←

model {  
 for ( $i$  in 1:N) {  
 $x[i] \sim dpois(\lambda[i])$   
 $\log(\lambda[i]) \leftarrow \mu + \log(t[i])$   
 }  
 $\mu \sim dnorm(mu\phi, tau\phi)$

data = list( $\mu\phi = 0, tau\phi = .001$ )

Model for GLM Varying

model {  
 for ( $i$  in 1:N) {  
 $x[i] \sim dpois(\lambda[i])$   
 $\log(\lambda[i]) \leftarrow \beta[i] + \log(t[i])$   
 $\beta[i] \sim dnorm(\mu, tau)$   
 }  
 $\mu \sim dnorm(mu\phi, tau\phi)$   
 $tau \sim dgamma(tau.a, tau.b)$

DATA: list( $\mu\phi = \phi, tau\phi = .001, tau.a = 1, tau.b = 1$ )

(12)

## So look at the 3 models

- look for unusual observations
- look for general goodness of fit.

Below

① each model is fit

② for each model → different diagnostics  
are fit using Win bug . and  
also some values are output to  
Excel for more processing .

③ given these values , some comments  
about the models are  
made .

```
# PumpDiagScp.txt      November 5, 2002
#
# runs the models for model diagnostic example
#   The data is from the Pump example in the manual
#
display('log')
check('C:/mike/BayesCourse2/ModelDiag/pumpGLMconsMdl.txt')
data('C:/mike/BayesCourse2/ModelDiag/pumpGLMconsData.txt')
compile(1)
inits(1, 'C:/mike/BayesCourse2/ModelDiag/pumpGLMconsInit.txt')
update(1000)
# getting the residuals
set(res)
set(stdres)
# getting the calibration statistics
set(res.rep)
set(stdres.rep)
# To get the probability of more extreme
set(p.smaller)
# predictive ordinates:
set(p.inv)
# some goodness of fit statistics:
set(chidev1.pval)
set(chidev2.pval)
set(chidev2.obs)
set(chidev2.rep)
# the deviance statistics and its distribution
set(deviance)
set(deviance.rep)
set(Dev.pval)
update(10000)
stats(*)
save('C:/mike/BayesCourse2/ModelDiag/pumpGLMconsOut.txt')
```

Script file  
to run  
the Model  
with Constant  
rate

→ Set the nodes we  
will sample from

→ get the statistics

→ take this file and put  
in Excel to calculate  
some other stats.

```
# note, here is the data file (with out the "#" of course):
#
# list(mu0=0, tau0=.001, tau.a=1, tau.b=1,
# t = c(94.3, 15.7, 62.9, 126, 5.24, 31.4, 1.05, 1.05, 2.1, 10.5),
# x = c(5,1,5,14, 3,19,1,1, 4,22), N = 10)
```

```
# Note, here is the initialization file. If you don't initialize y.rep, then
# you will need to gen.inits() command in script file
#
# list(mu=1,tau=1,x.rep=c(1,1,1, 1,1, 1,1,1, 1,1))
```

Model file is given on the  
next page

$$X_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \mu + \log(t_i)$$

## GLM: Constant Model

```

model{
for(i in 1:N){
  :[i]~dpois(lambda[i])
  log(lambda[i])<- mu + log(t[i])
  : beta[i]~dnorm(mu,tau)

#####
:      model checking steps are here.....
:      getting the residuals for the observed values...
:      note: I am deviating from the bugs manual... not getting the moments.

res[i]<-(x[i]-lambda[i])           # estimate of the residuals for this model
stdres[i]<-res[i]/sqrt(lambda[i])   # for the standardized residuals
                                     # Note: the variance of a poisson is the mean...

lev1.obs[i]<-pow(res[i],2)
lev2.obs[i]<-pow(stdres[i],2)

:      getting a replicated sample.... This is a sample of the predictive distribution
:rep[i]~dpois(lambda[i])
:smaller[i] <-step(x[i]-x.rep[i])    # check to see the probability of getting a more extrem
:value

:      residual and moments of replicated data.... this gives the predicted distribution for these
values.

res.rep[i]<-x.rep[i] - lambda[i]
stdres.rep[i]<-res.rep[i]/sqrt(lambda[i])

lev1.rep[i]<-pow(res.rep[i],2)
lev2.rep[i]<-pow(stdres.rep[i],2)

:      likelihood for each observed and replicated data....
:      note: need to know the density function of the probability model

oglike[i]<-x[i]*log(lambda[i]) -lambda[i] -logfact(x[i])
oglike.rep[i]<-x.rep[i]*log(lambda[i]) -lambda[i] -logfact(x.rep[i])

).inv[i]<- 1/exp(loglike[i])          # this is to find the predictive ordinate of the obs
:rvations

mu~dnorm(mu0,tau0)
tau~dgamma(tau.a,tau.b)

#####
:      summing the diagnostic values

hiddev1.obs <- sum(dev1.obs[])
hiddev2.obs <- sum(dev2.obs[])

hiddev1.rep <- sum( dev1.rep[] )
hiddev2.rep <- sum( dev2.rep[] )

```

chidev1.pval<-step(chidev1.obs-chidev1.rep)  
chidev2.pval<-step(chidev2.obs-chidev2.rep)

(15)

```
# Deviance statistic  
  
deviance<- -2*sum(loglike[])  
deviance.rep <- -2*sum(loglike.rep[])  
  
Dev.pval<-step(deviance-deviance.rep)
```

}

## Calculating VARIOus Diagnostics

① for  $\text{res} = y_i - E(y_i)$

get Posterior Mean of  $(y_i - E(y_i))$

N.B:  $E(y_i)$  moves around... with MCMC.

in  $\left[ \text{for } (i \text{ in } 1:N) \right]$  loops define

" $\text{res} \leftarrow \{ X[i] - \lambda[i] \}$ "

- Since  $\lambda[i]$  is  ~~$E(y_i)$~~  everything else in the MCMC.
- To calculate, need samples of predictive ~~distribution~~ distribution. Call these  $X.\text{rep}[i]$ , with
  - " $X.\text{rep}[i] \sim dpois(\lambda[i])$ ".

So in each iteration of MCMC, a new value of  $X.\text{rep}[i]$  will be sampled.

(16)

- to get predictive distribution of  $\text{res}[i]$ ,

define random variable  $\text{res.rep}[i]$  as:

"

$$\text{res.rep}[i] \leftarrow \text{x.rep}[i] - \lambda[i]$$

That is, replace " $\text{x}[i]$ " by " $\text{x.rep}[i]$ ". The posterior distribution of "res.rep[i]" will give you the "Typical distribution" of what you expect  $\text{res}[i]$  to be.

So, look at a ~~frequency histogram~~

Summary statistics of  $\text{res.rep}[i]$  to

see if  $\text{res}[i]$  is "typical".

(2)

### Standardize residual

$$\text{stres} = [y_i - E(y_i)] / \sqrt{V(y_i)}$$

- Similar to res, but need  $\text{Var}(y_i)$ . Fit model, get the variance based on model. Here, we have Poisson model, and it is known  $VAR = \lambda[i]$  for Poisson random variables.

- So define (in the for loop.) (17)

~~Calibration~~

- " $\text{Stres}[i] \leftarrow \text{res}[i] / \sqrt{\lambda[i]}$ "
- This value should be vaguely t-ish since it is standardised. But this depends on  $\text{res}[i]$  being approximately normal. So, usually don't need to calibrate, but it is easy to do. Do it the same way "Res" was calibrated, by using  $x.\text{rep}[i]$  again.

### ③ Prob of more Extreme Observation:

$$\min(P(Y_i < y_i), P(Y_i > y_i))$$

⇒ with the sampled value from the predictive distribution, this is fairly easy.

- Define a variable, say  $z = \begin{cases} 1 & \text{when } \\ = 0 & \text{other} \end{cases}$

$x.\text{rep} \leq x$

(18) This is done w/ "step" function in WinBugs

$$\text{Step}(e) = \begin{cases} 1 & \text{when } e \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

So

$$\text{Let } p.\text{smaller}[i] \leftarrow \text{step}(x[i] - x.\text{rep}[i])$$

then the posterior distribution of

$$p.\text{smaller}[i] \text{ is the } P(Y_i \leq x_i).$$

So, extreme values happen when this mean is either near 1 or 0.

---

(4) "Suprised obs"

Since Poisson is unimodal, we don't worry about this diagnostic.

---

(5) Predicted ordinates

---

↳ this is done for the crossel validated value.

(19)

Step 1

first calculate log-like function for each observation.

$$\begin{aligned} \text{"loglike[i] } &\leftarrow x[i] * \log(\lambda_{\text{obs}}[i]) \\ &- \lambda_{\text{obs}}[i] - \log \text{fact}(x[i]) \end{aligned}$$

→ Note: taking logs is more stable than first calculating large values and dividing.

→ Note use of log fact(-) functions-

→ Please Note that this is for the Poisson model. Other models will need different likelihoods.

→ For Poisson likelihood

$$P(X|\lambda) = \frac{\lambda^X}{X!} \exp(-\lambda)$$

↳ The poisson density function.

(20)

Step 2get " $p_{inv}[i] = 1 / \exp(\text{log like}[i])$ "Step 3get  $E(p_{inv}[i] | \text{data})$ 

as posterior mean. this is obtained

by monitoring "p.inv" nodes and  
getting mean from Winbugs.Step 4

$$\hat{P}(y_i | y_{\setminus i}) = \frac{1}{E(1 | p(y_i | \theta) | y_{\setminus i})}$$

The predictive ordinates (cross validated)

is obtained by getting inverse of the  
output to step 3.

⇒ Note : easiest way is to cut & paste results  
from Winbugs STMT command and PCT  
into Excel.

21

## Deviance + Goodness of fit, etc

Pseudo Bayes factor  $\Rightarrow$  "Bayes log. like ratio STATISTIC"

$\Rightarrow$  get [X-validated] log likelihood.

This is  $\ln(\hat{P}(y_i | y_{-i})) = \text{log like}_{\Sigma i}$

$\Rightarrow$  get this from Pg 20 Step 4, and  
take logs.

$\Rightarrow$  Pseudo BF = loglike =  $-2 \sum \text{log like}_{\Sigma i}$

Do this in EXCEL.

## Chi-square goodness of fit

$\Rightarrow$  from Standard Residuals.

~~All figures displayed on this slide are deleted statement~~

$\Rightarrow$  in for loop define  $\left( \frac{y_i - E(y_i)}{\sqrt{v_i y_i}} \right)^2$

" $\text{devZ}^{obs}[i] \leftarrow \text{Row(stres}[i], 2)$

⇒ outside for loop and  
in Winbugs Model -  
define node chiderv2

$$\text{chiderv2} \leftarrow \text{sum}(\text{dev2-obs}[3])$$

⇒ to calibrate → define

chiderv2-rep from  $X_{\text{rep}}$ .

---



---

### deviance

⇒ Remember, in for loop, had defined

$\loglik[3]$

⇒ outside of for loop, but in Model <sup>Winbugs</sup> sum  
these values up:

$$\text{deviance} \leftarrow -2 * \text{sum}(\loglik[3])$$

⇒ to calibrate could use  $X_{\text{rep}}$  values  
and look at distribution.

# Results of Winbugs Model for Constant rate

23

Model with constant rate

node	mean	sd	MC error	2.50%	median	97.50%	sample
Dev.pval	1	0	1.00E-12	1	1	1	10000
chidev1.pval	1	0	1.00E-12	1	1	1	10000
chidev2.obs	261	31.14	0.3445	209.7	257.4	331.4	10000
chidev2.pval	1	0	1.00E-12	1	1	1	10000
chidev2.rep	10.04	5.814	0.05531	3.02	8.727	24.65	10000
deviance	159	1.448	0.01415	158	158.4	163	10000
deviance.rep	36.42	4.6	0.04674	29.2	35.8	47.11	10000
p.inv[1]	1.68E+05	1.34E+06	14060	936.3	20660	1.04E+06	10000
p.inv[2]	9.052	2.718	0.02985	5.321	8.556	15.82	10000
p.inv[3]	364	703.1	7.701	35.84	189.6	1784	10000
p.inv[4]	2582	20260	211.5	38.14	413.7	15200	10000
p.inv[5]	13.58	3.15	0.03474	8.867	13.07	21.12	10000
p.inv[6]	69480	248300	2478	1506	19360	438100	10000
p.inv[7]	5.618	0.5178	0.005732	4.719	5.574	6.751	10000
p.inv[8]	5.618	0.5178	0.005732	4.719	5.574	6.751	10000
p.inv[9]	1023	460.5	5.049	424.4	924.1	2188	10000
p.inv[10]	3.88E+15	3.79E+16	3.72E+14	2.59E+12	1.98E+14	2.38E+16	10000
p.smaller[1]	4.00E-04	0.02	1.97E-04	0	0	0	10000
p.smaller[2]	0.1567	0.3635	0.0034	0	0	1	10000
p.smaller[3]	0.0118	0.108	0.001192	0	0	0	10000
p.smaller[4]	0.0124	0.1107	0.001074	0	0	0	10000
p.smaller[5]	0.9691	0.173	0.00177	0	1	1	10000
p.smaller[6]	1	0	1.00E-12	1	1	1	10000
p.smaller[7]	0.9775	0.1483	0.001403	1	1	1	10000
p.smaller[8]	0.977	0.1499	0.00158	1	1	1	10000
p.smaller[9]	0.9997	0.01732	1.71E-04	1	1	1	10000
p.smaller[10]	1	0	1.00E-12	1	1	1	10000
res[1]	-15.25	2.371	0.02623	-20.19	-15.17	-10.87	10000
res[2]	-2.371	0.3947	0.004367	-3.195	-2.358	-1.643	10000
res[3]	-8.507	1.581	0.0175	-11.81	-8.453	-5.589	10000
res[4]	-13.06	3.168	0.03505	-19.66	-12.95	-7.211	10000
res[5]	1.875	0.1317	0.001458	1.6	1.879	2.118	10000
res[6]	12.26	0.7895	0.008734	10.61	12.28	13.71	10000
res[7]	0.7745	0.0264	2.92E-04	0.7195	0.7754	0.8232	10000
res[8]	0.7745	0.0264	2.92E-04	0.7195	0.7754	0.8232	10000
res[9]	3.549	0.0528	5.84E-04	3.439	3.551	3.646	10000
res[10]	19.75	0.264	0.002921	19.19	19.75	20.23	10000
res.rep[1]	-0.01142	4.501	0.04487	-8.42	-0.1875	9.272	10000
res.rep[2]	0.01364	1.814	0.01729	-3.064	-0.1466	3.986	10000
res.rep[3]	-0.005802	3.678	0.03871	-6.593	-0.1652	7.688	10000
res.rep[4]	0.02985	5.237	0.05232	-9.838	-0.1607	10.81	10000
res.rep[5]	0.006549	1.072	0.01038	-1.304	-0.1263	2.764	10000
res.rep[6]	-0.03201	2.576	0.029	-4.624	-0.2084	5.527	10000
res.rep[7]	0.00422	0.4793	0.004434	-0.2762	-0.2152	0.826	10000
res.rep[8]	0.00462	0.4824	0.005646	-0.2772	-0.2155	0.83	10000
res.rep[9]	0.005641	0.6771	0.006616	-0.545	-0.4061	1.582	10000
res.rep[10]	-0.0146	1.486	0.01517	-2.348	-0.1949	3.362	10000

↑ calibration of  
residues

Extreme values  
"p-values"

Residual

All factors for  
Residuals

24

stdres[1]	-3.375	0.3288	0.003641	-4.023	-3.378	-2.729	10000
stdres[2]	-1.286	0.1395	0.001545	-1.56	-1.287	-1.011	10000
stdres[3]	-2.301	0.2953	0.00327	-2.88	-2.305	-1.717	10000
stdres[4]	-2.487	0.4632	0.00513	-3.389	-2.494	-1.566	10000
stdres[5]	1.784	0.2294	0.002541	1.352	1.775	2.255	10000
stdres[6]	4.763	0.5849	0.006478	3.663	4.74	5.965	10000
stdres[7]	1.643	0.1524	0.001688	1.358	1.636	1.958	10000
stdres[8]	1.643	0.1524	0.001688	1.358	1.636	1.958	10000
stdres[9]	5.317	0.392	0.004342	4.591	5.298	6.133	10000
stdres[10]	13.23	0.9554	0.01058	11.46	13.18	15.22	10000
stdres.rep[1]	-0.00264	0.9995	0.009838	-1.84	-0.04071	2.065	10000
stdres.rep[2]	0.006431	0.9888	0.009377	-1.73	-0.08153	2.169	10000
stdres.rep[3]	-0.001963	1.001	0.01061	-1.801	-0.04576	2.091	10000
stdres.rep[4]	0.00564	1.006	0.01005	-1.886	-0.03139	2.076	10000
stdres.rep[5]	0.006237	1.01	0.009752	-1.142	-0.119	2.498	10000
stdres.rep[6]	-0.01257	0.9915	0.01109	-1.769	-0.0826	2.111	10000
stdres.rep[7]	0.009114	1.01	0.009337	-0.5255	-0.4639	1.98	10000
stdres.rep[8]	0.009984	1.013	0.0118	-0.5265	-0.4642	2.013	10000
stdres.rep[9]	0.008384	1.008	0.00988	-0.7382	-0.6372	2.446	10000
stdres.rep[10]	-0.009938	0.9903	0.01009	-1.532	-0.1316	2.22	10000

Predictive ordinates	1/p(y)	p(y)	Log(like)
1	1.68E+05	5.95E-06	-12.03172
2	9.05E+00	1.10E-01	-2.202986
3	3.64E+02	2.75E-03	-5.897154
4	2.58E+03	3.87E-04	-7.85632
5	1.36E+01	7.36E-02	-2.608598
6	6.95E+04	1.44E-05	-11.14879
7	5.62E+00	1.78E-01	-1.725976
8	5.62E+00	1.78E-01	-1.725976
9	1.02E+03	9.78E-04	-6.930495
10	3.88E+15	2.58E-16	-35.89538

$$\log(p(y)) = \log(\text{like})$$

-2 \* Sum of the  
column = 176.0468

Pseudo -2 LogLike:

176.0468

## Results for constant Model

25

$\Rightarrow \chi^2$  goodness of fit

mean =

observed  $\text{chidev2-obs} = 261 \rightarrow \text{very big}$

calibration  $\text{chidev2-rep} \Rightarrow (3.02, 24.65)$

261 outside range  $\rightarrow$  does not seem to fit too well

Deviance  $\rightarrow$  Deviance = 159 and it should be between (29 and 47) ...

P.IRV These are the values of  
 $E(\hat{p}(y_i | \theta))$

for  $i=1$  This equals  $1.68 \times 10^5$

$\Rightarrow$  So estimate  $\hat{p}(y_i | y_{\setminus i})$

$$= 1/E( ) = \frac{1}{1.68 \times 10^5} = \underline{\underline{3.95 \times 10^{-6}}}$$

which is very small



More on these after STAT results  
on bottom of page

<u>P. smaller</u>	<u>p smaller</u>
1	$4(10)^{-4}$
2	•1567
3	•0118
4	•0124
⋮	⋮

→ These are the ~~ex~~ probability of seeing an observation this extreme. ≈ The "p-value" type of statistic.

→ N.B: All values are close to 1000.  
So all the values are fairly extreme and none are really in the middle.

<u>Residuals</u>	<u><math>y_i - E(y_i)</math></u>
1	-15.25
2	-2.371
3	-8.507
4	-13.06
⋮	⋮

⇒ Note: difficult to see what is going on without calibration

⇒ Calibration is in res-per rep

1	-15.25	(-8.42, 9.272) ← Extreme
2	-2.371	(-3.064, 3.986) ← not too bad
3	-8.507	(-6.593, 7.688) ← Extreme
4	-13.06	(-9.838, 10.81) ← Extreme
⋮	⋮	⋮

⇒ Please, Note that each obs has a different variance + shape, so we need these calibrations supplied by res. rep 3 modes

<u>Standardize residuals</u>		$\frac{y_i - \bar{y}_i}{\sqrt{\text{VAR}(y_i)}}$
	st. res	( 2.5%, 97.5% )
1	-3.375	( -1.84, 2.065 )
2	-1.286	( -1.73, 2.065 )
3	-2.301	( -1.801, 2.091 )
4	-2.427	( -1.886, 2.076 )
:	:	:

⇒ So again, many large residuals. (the 2nd one is not bad.)

⇒ N.B: Although n standardized - Please Note that the distribution of these residuals is skewed (it's a Poisson model). So there is some value to calibration.

→ Predictive ordinates

Calculated by taking

$$\frac{1}{E(p_{inv})}$$

so get

	<u>p<sub>inv</sub></u>	<u>Pred<sub>ord</sub></u>
1	$1.68(10)^5$	$5.95(10)^{-6}$
2	$9.05(10^8)$	$1.10(10)^{-1}$
3	$3.64(10)^2$	$2.75(10)^{-3}$
.	⋮	⋮

$$\log \text{like} = \ln(\text{Pred<sub>ord</sub>})$$

$$-12.03$$

$$-2.202$$

$$-5.89$$

⋮

$$-2 \sum (\uparrow)$$

$$= 176.0468$$

→ Compare w/ other  
models by getting  
difference.

→ like using

-2 LR test  
sort of -

Conclusion: This Model

does not seem to fit  
too well

# GLM Model w/ VARYING mean

```

model{
for(i in 1:N){
x[i]~dpois(lambda[i])
log(lambda[i])<- beta[i] + log(t[i])
beta[i]~dnorm(mu,tau)
}
#####
# model checking steps are here.....
#
# getting the residuals for the observed values...
# note: I am deviating from the bugs manual... not getting the moments.

res[i]<- (x[i]-lambda[i]) # estimate of the residuals for this model
stdres[i]<-res[i]/sqrt(lambda[i]) # for the standardized residuals
# Note: the variance of a poisson is the mean...

dev1.obs[i]<-pow(res[i],2)
dev2.obs[i]<-pow(stdres[i],2)

# getting a replicated sample.... This is a sample of the predictive distribution

x.rep[i]~dpois(lambda[i])
p.smaller[i] <-step(x[i]-x.rep[i]) # check to see the probability of getting a more extreme value

# residual and moments of replicated data.... this gives the predicted distribution for these values.

res.rep[i]<-x.rep[i] - lambda[i]
stdres.rep[i]<-res.rep[i]/sqrt(lambda[i])

dev1.rep[i]<-pow(res.rep[i],2)
dev2.rep[i]<-pow(stdres.rep[i],2)

# likelihood for each observed and replicated data...
# note: need to know the density function of the probability model

loglike[i]<-x[i]*log(lambda[i]) -lambda[i] -logfact(x[i])
loglike.rep[i]<-x.rep[i]*log(lambda[i]) -lambda[i] -logfact(x.rep[i])

p.inv[i]<- 1/exp(loglike[i]) # this is to find the predictive ordinate of the observations

}

mu~dnorm(mu0,tau0)
tau~dgamma(tau.a,tau.b) } Also changed from other Model.

#####
# summing the diagnostic values

chidev1.obs <- sum(dev1.obs[])
chidev2.obs <- sum(dev2.obs[])

chidev1.rep <- sum( dev1.rep[] )
chidev2.rep <- sum( dev2.rep[] )

```

20

```
: hidev1.pval<-step(chidev1.obs-chidev1.rep)
: hidev2.pval<-step(chidev2.obs-chidev2.rep)

: Deviance statistic

leviance<- -2*sum(loglike[])
leviance.rep <- -2*sum(loglike.rep[])

dev.pval<-step(deviance-deviance.rep)
```

---

Also data : set

list ( $\mu\phi=0$ ,  $\tau\mu\phi=0.001$ ,  $\tau\alpha=1$ ,  $\tau\beta=1$ )

---

Same basic script file

---

Output on Next page,

(31)

Pump Model with varying rate parameter

node	mean	sd	MC error	2.50%	median	97.50%	sample
Dev.pval	0.6801	0.4664	0.0047	0	1	1	10000
chidev1.pval	0.5099	0.4999	0.005002	0	1	1	10000
chidev2.obs	13.77	11.24	0.1291	3.521	11.29	38.08	10000
chidev2.pval	0.6169	0.4861	0.00483	0	1	1	10000
chidev2.rep	9.979	5.337	0.05528	3.219	8.859	23.16	10000
deviance	43.37	4.34	0.04506	36.77	42.75	53.39	10000
deviance.rep	40.25	4.827	0.05552	32.35	39.74	51.17	10000
p.inv[1]	22.32	326.3	3.317	5.702	7.163	70.95	10000
p.inv[2]	6.545	54	0.5306	2.72	3.266	22.35	10000
p.inv[3]	15.81	93.4	1.106	5.702	7.032	60.43	10000
p.inv[4]	40.15	1386	13.81	9.439	11.83	110.1	10000
p.inv[5]	12.06	150.3	1.471	4.466	5.544	42.41	10000
p.inv[6]	48.38	965.4	9.48	10.98	13.79	152.2	10000
p.inv[7]	5.238	7.343	0.07815	2.719	3.481	17.89	10000
p.inv[8]	5.313	11.68	0.1168	2.72	3.503	17.55	10000
p.inv[9]	26.55	199.4	2.009	5.123	7.213	134.3	10000
p.inv[10]	65.91	1476	14.7	11.81	15.23	220.3	10000
p.smaller[1]	0.4982	0.5	0.005564	0	0	1	10000
p.smaller[2]	0.5567	0.4968	0.004849	0	1	1	10000
p.smaller[3]	0.5239	0.4994	0.004857	0	1	1	10000
p.smaller[4]	0.5227	0.4995	0.005146	0	1	1	10000
p.smaller[5]	0.6905	0.4623	0.004513	0	1	1	10000
p.smaller[6]	0.5975	0.4904	0.005198	0	1	1	10000
p.smaller[7]	0.825	0.38	0.003797	0	1	1	10000
p.smaller[8]	0.8291	0.3764	0.0042	0	1	1	10000
p.smaller[9]	0.7647	0.4242	0.003865	0	1	1	10000
p.smaller[10]	0.6199	0.4854	0.005024	0	1	1	10000
res[1]	-0.9991	2.397	0.02592	-6.514	-0.7006	2.706	10000
res[2]	-0.7133	1.154	0.01312	-3.597	-0.4576	0.7535	10000
res[3]	-0.7636	2.321	0.0247	-6.227	-0.4513	2.826	10000
res[4]	-0.6807	3.785	0.04206	-9.213	-0.3654	5.759	10000
res[5]	0.2063	1.528	0.01452	-3.524	0.4849	2.299	10000
res[6]	0.4102	4.284	0.04336	-8.956	0.7997	7.745	10000
res[7]	0.2627	0.684	0.007231	-1.583	0.4706	0.9367	10000
res[8]	0.2759	0.6764	0.007422	-1.56	0.4802	0.9362	10000
res[9]	0.8735	1.663	0.0162	-3.163	1.172	3.2	10000
res[10]	1.132	4.585	0.04778	-8.766	1.438	9.26	10000
res.rep[1]	-0.02024	2.461	0.0258	-4.411	-0.2025	5.277	10000
res.rep[2]	6.05E-04	1.31	0.01327	-2.277	-0.1618	3.066	10000
res.rep[3]	-0.004072	2.396	0.02314	-4.243	-0.1857	5.219	10000
res.rep[4]	0.01999	3.881	0.0366	-7.153	-0.1627	8.142	10000
res.rep[5]	-0.009365	1.653	0.01607	-2.905	-0.18	3.712	10000
res.rep[6]	-0.02355	4.315	0.04085	-7.985	-0.2255	8.876	10000
res.rep[7]	0.002066	0.8542	0.008452	-1.384	-0.1769	2.139	10000
res.rep[8]	0.001348	0.8567	0.009282	-1.369	-0.181	2.169	10000
res.rep[9]	0.001713	1.765	0.01495	-3.107	-0.171	3.906	10000
res.rep[10]	0.1292	4.567	0.04242	-8.407	-0.04895	9.801	10000

32

stdres[1]	-0.226	0.9552	0.01007	-1.92	-0.2934	1.787	10000
stdres[2]	-0.3073	0.8285	0.009069	-1.678	-0.379	1.517	10000
stdres[3]	-0.133	0.955	0.01019	-1.858	-0.1933	1.917	10000
stdres[4]	-0.05217	0.9866	0.0108	-1.912	-0.09641	2.006	10000
stdres[5]	0.4073	1.05	0.009756	-1.38	0.3057	2.746	10000
stdres[6]	0.2132	1.022	0.01031	-1.694	0.1875	2.309	10000
stdres[7]	0.8217	1.23	0.0118	-0.9851	0.6467	3.724	10000
stdres[8]	0.8408	1.232	0.01319	-0.9749	0.666	3.707	10000
stdres[9]	0.8219	1.198	0.01225	-1.182	0.6972	3.577	10000
stdres[10]	0.3652	1.054	0.01086	-1.58	0.3171	2.594	10000
stdres.rep[1]	-0.007027	1.009	0.01018	-1.74	-0.08954	2.174	10000
stdres.rep[2]	-7.93E-04	0.9973	0.009735	-1.428	-0.1561	2.28	10000
stdres.rep[3]	-0.002611	1.001	0.009758	-1.716	-0.083	2.153	10000
stdres.rep[4]	0.005106	1.011	0.009416	-1.836	-0.0431	2.109	10000
stdres.rep[5]	-0.007208	0.9905	0.009668	-1.572	-0.1237	2.239	10000
stdres.rep[6]	-0.007524	0.9979	0.009374	-1.855	-0.05555	2.032	10000
stdres.rep[7]	-0.002811	0.988	0.00952	-1.138	-0.3818	2.534	10000
stdres.rep[8]	-0.002784	0.995	0.01092	-1.138	-0.3886	2.57	10000
stdres.rep[9]	0.002291	0.9982	0.008455	-1.583	-0.1033	2.216	10000
stdres.rep[10]	0.02688	1.001	0.009247	-1.836	-0.01108	2.129	10000

Predictive Ordinat 1/p(y)	p(y)	log(like)
1	22.32	0.044803
2	6.545	0.152788
3	15.81	0.063251
4	40.15	0.024907
5	12.06	0.082919
6	48.38	0.02067
7	5.238	0.190913
8	5.313	0.188218
9	26.55	0.037665
10	65.91	0.015172

Psuedo LogLike:

57.19969

# GLM Model w/ Varying Mean

33

## Diagnostic Results

$$\chi^2 \text{ goodness of fit} = 13.77$$

w/ expected range (3.219, 23.16)

So it's within expected range.

$$\text{Deviance} = 43.37 \text{ w/ expected range } (32.35, 51.17)$$

So again within expected range

P-smaller  $\Rightarrow$  "p-value" diagnostic

Value range between .049 to .829

(Actually, if the value is bigger than .5, the real diagnostic is  $1 - \frac{\text{value}}{2}$ )

So All within range.

Residual  $\Rightarrow$  All within range

for example, for 1st obs,  $\text{res}[1] = -0.991$

and range from  $\text{res}.\text{rep}[1]$  is  $(-4.411, 5.277)$   
So within range.

Std. Res

again all within range-

for example  $\text{stdres}[1] \approx -0.226$

and range should be  $(-1.74, 2.174)$

Pseudo likelihood = 57.199 which is  
much smaller than 176.

So Pseudo Bayes factor =  $176 - 57.199$

$$\approx \underline{\underline{119}}$$

```

# for conjugate model
#
model{
for(i in 1:N){
x[i]~dpois(lambda[i])
lambda[i]<- theta[i]*t[i]
theta[i]~dgamma(alpha,beta)
#####
# model checking steps are here.....
}

# getting the residuals for the observed values...
# note: I am deviating from the bugs manual... not getting the moments.

res[i]<- (x[i]-lambda[i]) # estimate of the residuals for this model
stdres[i]<-res[i]/sqrt(lambda[i]) # for the standardized residuals
# Note: the variance of a poisson is the mean...

dev1.obs[i]<-pow(res[i],2)
dev2.obs[i]<-pow(stdres[i],2)

# getting a replicated sample..... This is a sample of the predictive distribution

x.rep[i]~dpois(lambda[i])
p.smaller[i] <-step(x[i]-x.rep[i]) # check to see the probability of getting a more extrem
e value

# residual and moments of replicated data.... this gives the predicted distribution for these
values.
res.rep[i]<-x.rep[i] - lambda[i]
stdres.rep[i]<-res.rep[i]/sqrt(lambda[i])

dev1.rep[i]<-pow(res.rep[i],2)
dev2.rep[i]<-pow(stdres.rep[i],2)

# likelihood for each observed and replicated data...
# note: need to know the density function of the probability model
loglike[i]<-x[i]*log(lambda[i]) -lambda[i] -logfact(x[i])
loglike.rep[i]<-x.rep[i]*log(lambda[i]) -lambda[i] -logfact(x.rep[i])

p.inv[i]<- 1/exp(loglike[i]) # this is to find the predictive ordinate of the obs
ervations
}

alpha~dexp(1)
beta~dgamma(0.1,1.0)
#####
# summing the diagnostic values

chidev1.obs <- sum(dev1.obs[])
chidev2.obs <- sum(dev2.obs[])

chidev1.rep <- sum( dev1.rep[] )
chidev2.rep <- sum( dev2.rep[] )

chidev1.pval<-step(chidev1.obs-chidev1.rep)
chidev2.pval<-step(chidev2.obs-chidev2.rep)

# Deviance statistic
deviance<- -2*sum(loglike[])
deviance.rep <- -2*sum(loglike.rep[])

```

Model w/ gamma prior

→ Changed here

35

→ Changed here

ev.pval<-step(deviance-deviance.rep)

36

## With conjugate model

node	mean	sd	MC error	2.50%	median	97.50%	sample
Dev.pval	0.6221	0.4849	0.004691	0	1	1	10000
chidev1.pval	0.495	0.5	0.00513	0	0	1	10000
chidev2.obs	12.7	17.87	0.1767	3.046	10.01	35.34	10000
chidev2.pval	0.5654	0.4957	0.004875	0	1	1	10000
chidev2.rep	9.914	5.241	0.04638	3.16	8.834	23.43	10000
deviance	42.76	4.154	0.04192	36.45	42.12	52.33	10000
deviance.rep	40.76	4.822	0.04804	32.8	40.22	51.71	10000
p.inv[1]	16.93	118.5	1.183	5.702	7.167	74.17	10000
p.inv[2]	6.931	29.29	0.2995	2.719	3.351	28.71	10000
p.inv[3]	21.59	243.9	2.348	5.702	7.177	71.17	10000
p.inv[4]	26.41	116.6	1.337	9.44	11.84	112.4	10000
p.inv[5]	11.1	52.72	0.6286	4.466	5.446	43.08	10000
p.inv[6]	34.49	283.1	2.87	10.98	13.85	120.1	10000
p.inv[7]	4.838	10.5	0.09719	2.719	3.182	16.02	10000
p.inv[8]	4.823	8.604	0.1005	2.719	3.176	16.27	10000
p.inv[9]	16.17	96.74	0.8922	5.122	6.664	70.49	10000
p.inv[10]	67	2505	25	11.81	15.07	169.2	10000
p.smaller[1]	0.5367	0.4987	0.00563	0	1	1	10000
p.smaller[2]	0.5934	0.4912	0.004802	0	1	1	10000
p.smaller[3]	0.5429	0.4982	0.004372	0	1	1	10000
p.smaller[4]	0.5432	0.4981	0.004568	0	1	1	10000
p.smaller[5]	0.6262	0.4838	0.005218	0	1	1	10000
p.smaller[6]	0.559	0.4965	0.004637	0	1	1	10000
p.smaller[7]	0.7564	0.4293	0.004256	0	1	1	10000
p.smaller[8]	0.7612	0.4264	0.003932	0	1	1	10000
p.smaller[9]	0.7375	0.44	0.004534	0	1	1	10000
p.smaller[10]	0.6245	0.4843	0.004708	0	1	1	10000
res[1]	-0.621	2.378	0.02907	-6.273	-0.2925	3.035	10000
res[2]	-0.5843	1.23	0.01202	-3.799	-0.2847	0.8679	10000
res[3]	-0.6474	2.4	0.0241	-6.292	-0.3165	3.048	10000
res[4]	-0.5399	3.803	0.03398	-8.962	-0.1799	5.869	10000
res[5]	-0.182	1.678	0.01833	-4.263	0.1009	2.237	10000
res[6]	-0.06853	4.302	0.04174	-9.117	0.2441	7.317	10000
res[7]	0.05052	0.7696	0.008067	-1.926	0.2454	0.9232	10000
res[8]	0.05618	0.7625	0.007368	-1.956	0.2487	0.9251	10000
res[9]	0.6797	1.593	0.01766	-3.078	0.9406	3.007	10000
res[10]	1.102	4.459	0.0472	-8.474	1.424	8.807	10000
res.rep[1]	0.06247	2.387	0.02484	-4.199	-0.1151	5.326	10000
res.rep[2]	-0.00857	1.227	0.009869	-2.241	-0.1801	2.823	10000
res.rep[3]	0.03561	2.355	0.02082	-4.151	-0.1371	5.201	10000
res.rep[4]	-0.02005	3.774	0.03786	-7.116	-0.183	7.811	10000
res.rep[5]	-9.31E-04	1.773	0.01889	-3.17	-0.1589	3.919	10000
res.rep[6]	-8.31E-04	4.414	0.03969	-8.26	-0.1729	9.098	10000
res.rep[7]	0.009624	0.9683	0.009326	-1.61	-0.1688	2.396	10000
res.rep[8]	-0.005422	0.9598	0.009065	-1.591	-0.1861	2.259	10000
res.rep[9]	-0.003875	1.846	0.01847	-3.211	-0.1785	4.152	10000
res.rep[10]	0.008293	4.562	0.0444	-8.51	-0.1858	9.675	10000

stdres[1]	-0.05276	1.019	0.01229	-1.868	-0.1271	2.165	10000
stdres[2]	-0.09895	1.152	0.01072	-1.734	-0.2512	2.388	10000
stdres[3]	-0.06209	1.024	0.01033	-1.873	-0.1373	2.181	10000
stdres[4]	-0.01148	1.004	0.009126	-1.87	-0.04779	2.058	10000
stdres[5]	0.1679	1.048	0.01117	-1.582	0.05927	2.56	10000
stdres[6]	0.0977	1.005	0.009909	-1.719	0.05637	2.141	10000
stdres[7]	0.4847	1.209	0.01155	-1.126	0.2825	3.332	10000
stdres[8]	0.4916	1.225	0.01308	-1.138	0.287	3.38	10000
stdres[9]	0.6347	1.062	0.01131	-1.157	0.5378	3.018	10000
stdres[10]	0.3505	1.015	0.01055	-1.535	0.3138	2.425	10000
stdres.rep[1]	0.02438	1.004	0.009914	-1.71	-0.05127	2.202	10000
stdres.rep[2]	-0.002626	0.9926	0.008673	-1.364	-0.243	2.377	10000
stdres.rep[3]	0.01154	0.9914	0.008519	-1.657	-0.06095	2.155	10000
stdres.rep[4]	-0.004765	0.9886	0.009871	-1.819	-0.04829	2.064	10000
stdres.rep[5]	-3.11E-04	0.9919	0.01061	-1.596	-0.09944	2.199	10000
stdres.rep[6]	-0.002156	1.01	0.009054	-1.881	-0.03921	2.079	10000
stdres.rep[7]	0.002803	0.9799	0.009571	-1.231	-0.3234	2.421	10000
stdres.rep[8]	-0.003137	0.9941	0.009066	-1.216	-0.3375	2.485	10000
stdres.rep[9]	-0.004114	1.007	0.01036	-1.609	-0.1138	2.244	10000
stdres.rep[10]	0.001005	0.9975	0.009852	-1.875	-0.04151	2.106	10000

Predictive Order	1/p(y)	p(y)	log(like)
1	16.93	0.059067	-2.829087
2	6.931	0.144279	-1.936004
3	21.59	0.046318	-3.07223
4	26.41	0.037864	-3.273743
5	11.1	0.09009	-2.406945
6	34.49	0.028994	-3.540669
7	4.838	0.206697	-1.576501
8	4.823	0.20734	-1.573396
9	16.17	0.061843	-2.783158
10	67	0.014925	-4.204693

Pseudo LogLike:

54.39285

$\chi^2$  GOF: 12.7 Range should be (3.16 to 23.43)  
So fine

deviance: 42.76 w/ range (32.8, 51.7) So fine

P. smaller, Res, and stdres are all fine

Pseudo Likelihood is 54.39

54.39 compares to 57.19

A is 2.8 so it is very similar  $\Rightarrow$  More Next week about comparing these #'s

# Selected Pages from earlier Version of The BUGS Manual.

## BUGS 0.5 \* Bayesian inference Using Gibbs Sampling Manual (version ii)

David Spiegelhalter

Andrew Thomas  
Wally Gilks

Nicky Best

MRC Biostatistics Unit, Institute of Public Health,  
Robinson Way, Cambridge CB2 2SR

Tel: 44-1223-330300

Fax: 44-1223-330388

e-mail: bugs@mrc-bsu.cam.ac.uk

ftp: ftp.mrc-bsu.cam.ac.uk

August 14, 1996

### Permission and Disclaimer

First the formal stuff.

Permission to use, copy and distribute this software and its documentation for any purpose is hereby granted without fee, provided that both copyright notice and this permission notice appears in all copies.

The MRC Biostatistics Unit disclaims all warranties with regard to this software, including all special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

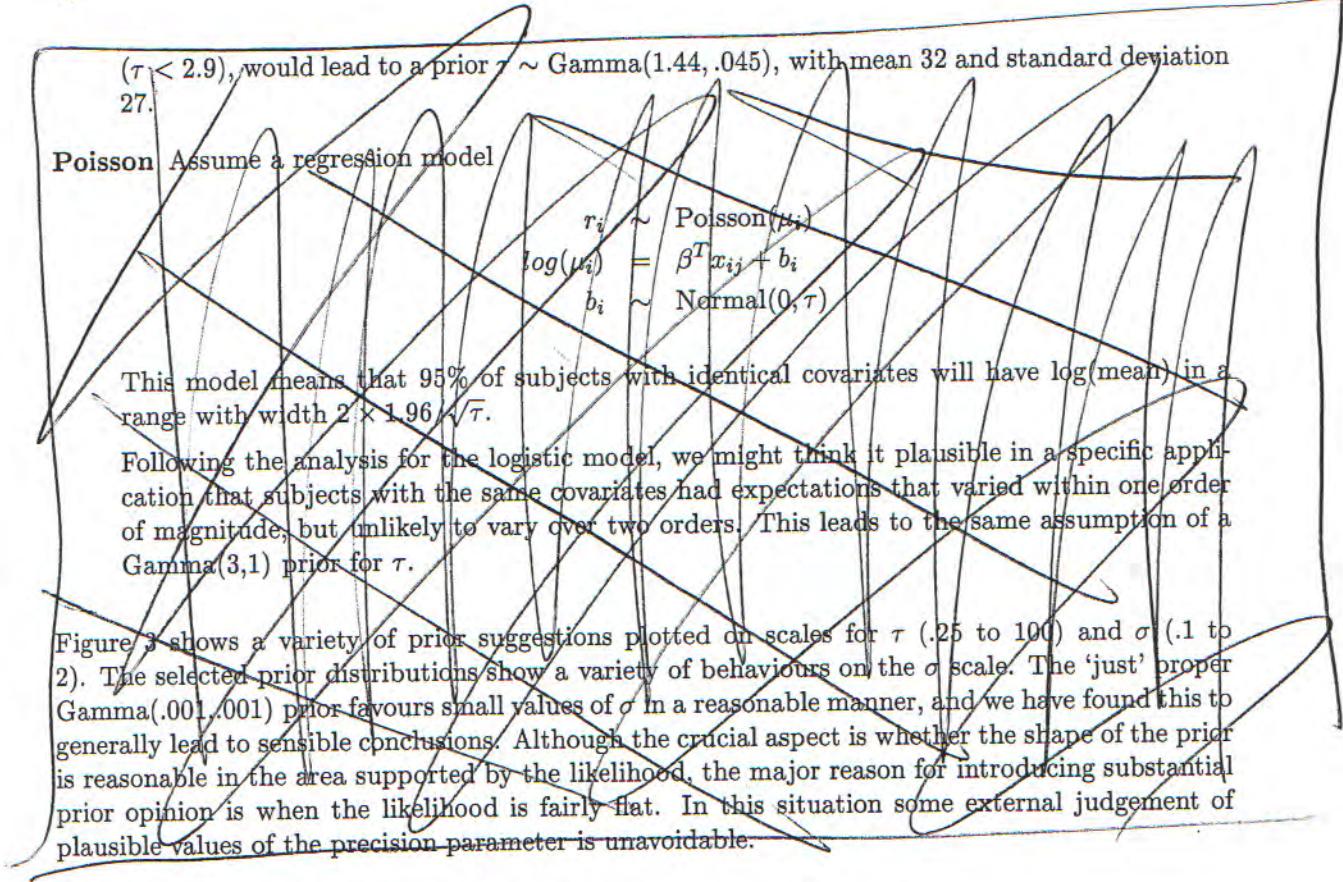
More informally, potential users are reminded to be extremely careful if using this program for serious statistical analysis. Appropriate parameterisation, monitoring for convergence, and model checking are only briefly discussed in this manual, and are left entirely to the user's responsibility. We strongly recommend an incremental approach to modelling, assessing at each stage sensitivity to inputs and assumptions.

We have tested the program on quite a wide set of examples (see accompanying *Examples 1* and *Examples 2* documents), but be particularly careful with types of model that are currently not featured. If there is a problem, BUGS might just crash, which is not very good, but it might well carry on and produce answers that are wrong, which is even worse. Please let us know of any successes or failures.

**Beware - Gibbs sampling can be dangerous!.**

---

\*BUGS ©copyright MRC Biostatistics Unit 1995. ALL RIGHTS RESERVED. The support of the Economic and Social Research Council (UK) is gratefully acknowledged. The work was funded in part by ESRC (UK) Award Number H519 25 5023.



### 9.3 Model criticism and selection

Although the emphasis in this manual and examples is firmly on drawing inferences assuming a model is an appropriate assumption, serious statistical science requires us to check these assumptions and justify our particular choice of model. We briefly outline some recent suggestions for model checking and comparison using MCMC methods - see the cited references for more details. We shall indicate which aspects can be easily included in a BUGS analysis.

We shall distinguish three objectives: model checking through examination of individual observations, comparison between two or more competitor models, and global checks of goodness-of-fit. In all cases we exploit the idea of comparing observed statistics with our expectations, were we making predictions conditional on the truth of a particular model assumption.

#### 9.3.1 Checking through examination of individual observations.

Consider data  $y_1, \dots, y_I$  and parameters  $\theta$  under the assumed model. Gelfand *et al.* (1992) suggest a series of 'checking functions' which can be calculated for each observation, involving a comparison of the observed  $y_i$  with a predictive distribution  $p(Y_i)$ . Leaving the basis for this predictive distribution aside for the moment, their suggestions include

1. the residual:  $y_i - E(Y_i)$
2. the standardised residual:  $(y_i - E(Y_i)) / \sqrt{V(Y_i)}$

3. the chance of getting a more extreme observation:  $\min(p(Y_i < y_i), p(Y_i \geq y_i))$
4. the chance of getting a more ‘surprising’ observation:  $p(Y_i : p(Y_i) \leq p(y_i))$
5. the predictive ordinate of the observation:  $p(y_i)$

The first and last require some standard against which to compare, whereas the middle three options speak for themselves.

Two situations can be identified: when the data  $y_1, \dots, y_I$  form a separate evaluation set, and when they were used for deriving the posterior distribution of the parameters.

1. *Separate evaluation data available.* In this case the posterior distribution is based on a ‘training set’  $\underline{x}$ . The predictive distribution is given by

$$p(Y_i|\underline{x}) = \int p(Y_i|\underline{x}, \theta)p(\theta|\underline{x})d\theta.$$

In many cases, unless say there is an explicit dependence on previous observations, the  $y$ 's are conditionally independent of the  $x$ 's given the total set of unknowns  $\theta$ . Thus the required integral is simply

$$p(Y_i|\underline{x}) = \int p(Y_i|\theta)p(\theta|\underline{x})d\theta.$$

This just requires adding additional random nodes  $Y_i$  in the graph with the appropriate parents, and monitoring the samples generated for  $Y_i$ .

The observed values  $y_i$  can then be compared with their predictive distributions which may be summarised using the `stats` command. As an application of the third checking function, one could then simply see how often the true observations lie within the nominal predictive intervals. Alternatively, one could calculate the first four checking functions above by explicitly including new nodes representing, for example,  $E(Y_i|\theta)$  and  $V(Y_i|\theta)$  and so obtain estimates of  $E(Y_i)$  and  $V(Y_i)$  directly by monitoring these nodes and noting the empirical mean.

2. *No separate evaluation data available.* In this case the predictive distribution of  $Y_i$  should ideally be conditional on the model and remainder of the data in order to perform “cross-validation”. Hence for each observation  $y_i$  we would require a distribution  $p(Y_i|\underline{y}_{\setminus i})$ , where  $\underline{y}_{\setminus i}$  is the rest of the data excluding  $y_i$ .

Unfortunately this is generally difficult to do within BUGS. An exception is the final checking function, since we can explicitly calculate the predictive ordinate within BUGS. Gelfand and Dey (1994) point out the interesting fact that

$$\frac{1}{p(y_i|\underline{y}_{\setminus i})} = \int \frac{1}{p(y_i|\underline{y}_{\setminus i}, \theta)} p(\theta|\underline{y}) d\theta$$

$\approx \text{Avg}_{\theta \sim \text{post}} \left[ \frac{1}{p(Y_i|\theta, \underline{y}_{\setminus i})} \right]$

and hence a Monte Carlo estimate of  $p(y_i|\underline{y}_{\setminus i})$  is obtained as a harmonic mean of  $p(y_i|\underline{y}_{\setminus i}, \theta)$ . This may be accomplished in BUGS by constructing a node which takes on values  $p(y_i|\underline{y}_{\setminus i}, \theta)^{-1}$ , and then taking the inverse of its empirical mean. However, harmonic means are notoriously unstable so care is required regarding convergence.

An approximation to the cross-validatory method is to use the methods for a separate evaluation set, but replacing  $\underline{x}$  by  $\underline{y}$ . Hence our predictive distribution is

$$p(Y_i|\underline{y}) = \int p(Y_i|\underline{y}, \theta)p(\theta|\underline{y})d\theta$$

Magic  
formulas

which will usually be expressible as

$$p(Y_i|y) = \int p(Y_i|\theta)p(\theta|y)d\theta$$

and hence just requires adding additional random nodes  $Y_i$  in the graph with the same parents as  $y_i$ .

If we do wish to sample from the correct cross-validatory predictive distribution, this can be carried out using an additional importance sampling step to remove the effect of  $y_i$  when repredicting  $Y_i$  (Gelfand *et al.*, 1992), although this would have to be carried out external to BUGS.

### 9.3.2 Comparison between two or more candidate models

#### Bayes factor approaches.

The Bayesian theory of model comparison is based on the Bayes factor, which for two competing models  $M_1$  and  $M_2$  is defined as the ratio of the marginal ordinates of the observed data

$$\frac{p(y|M_1)}{p(y|M_2)} = \frac{\int p(y|\theta_1, M_1)p(\theta_1|M_1) d\theta_1}{\int p(y|\theta_2, M_2)p(\theta_2|M_2) d\theta_2}$$

where  $\theta_i$  are the unobserved quantities in model  $i$ . There is a great deal of recent literature on the difficult issues surrounding the calculation and interpretation of Bayes factors: see for example Kass and Raftery (1995); Gelfand and Dey (1994). Carlin and Chib (1995) describe one means of calculating Bayes factors within a Monte Carlo run, and this is illustrated in our pines example.

#### Cross-validatory measures.

We note the analysis first described by Smith (1991) in which it is argued that a Bayes factor approach is only strictly appropriate if we truly believe that one, and only one, of the candidate models is true. If, as would generally be the case, the assumed models are simply considered as useful proxies for some theoretically and practically indeterminate ‘true’ model, then model comparison based on cross-validatory measures may be more suitable.

Comparison between models can therefore be based on an accumulation over observations of the checking functions described above. Examples might include the sum of squared or absolute unstandardised residuals  $y_i - E(Y_i)$ , or the negative cross-validatory log-likelihood

$$-\sum_{i=1}^n \log p(y_i|y_{\setminus i}).$$

The latter results in a comparison between models based on the ‘pseudo-Bayes factor’ (Geisser and Eddy, 1979; Gelfand and Dey, 1994).

We note that such accumulation needs to be carried out after a BUGS run using the estimates obtained from the monitors.

#### ‘Deviance’ measures

Dempster (1974) suggested examining the posterior distribution of the log-likelihood of the observed data: this is straightforward to calculate with a BUGS run by forming a variable with the value  $-2 \log p(y|\theta)$ . For non-hierarchical models, the minimum feasible value of this quantity is the

traditional deviance statistic, and in these circumstances a natural connection is made to classical model comparison. However, in for hierarchical models the minimum is likely to be very poorly estimated by the sample minimum, and the mean might be a more reasonable measure: this has been implemented by, for example, Gilks *et al.* (1992) and Zeger and Karim (1991) for model comparison. This quantity is illustrated in the `seeds` example for logistic models, the `salm` example for Poisson models, the `lsat` and `beetles` examples for binary data, the `alli` example for multinomial data and the `jaw` example for multivariate normal data. However, further work is required before giving recommendations on its use.

### 9.3.3 Global goodness-of-fit tests based on Bayesian p-values

Rubin (1984) introduced a ‘Bayesianly-justifiable’ frequentist model checking device. Suppose we calculate a statistic  $d(y)$  that we expect might be sensitive to departures of interest from the assumed model. For example, if we doubted the normal assumption for the population distribution in random-effects meta-analysis, we might calculate the range of the observed empirical log-odds ratios in the trials. Allowing design aspects of the study to be fixed, such as the sample sizes of the trials, we could then generate replicate sets of data based on our current beliefs about the parameters  $p(\theta|y)$ , and re-calculate for each replicate set  $y^{rep}$  the statistic  $d(y^{rep})$  which we may abbreviate to  $d^{rep}$ . Our observed statistic can then be compared with the distribution  $p(d^{rep})$  generated under the assumed model. If  $d(y)$  lies in the extreme tails of this distribution then evidence exists against the model.

This approach can be easily accommodated in BUGS by producing replicate data-sets and constructing nodes to calculate  $d$  based on the original and replicate data. The empirical distribution of  $d^{rep}$  can then be monitored using the `stats` command.

Gelman *et al.* (1996) extend this work of Rubin to allow discrepancy measures  $d(y, \theta)$  that depend both on the data and parameters, and this is straightforward to program within BUGS.

## 9.4 Implementation of selected model checking criteria in BUGS

We use the trivial `line` example introduced in Section 1.4 of this manual to illustrate how to implement a number of the above checking criteria in BUGS. (Note that we have changed the data point (2,3) in the `line.dat` file to (2,7) to represent an ‘outlier’).

The first 3 checking functions listed in Section 9.3.1 may be calculated directly in BUGS. Note that in this example,  $E(Y_i)$  is denoted by `mu[i]`. To compute  $p(Y_i < y_i)$  (see checking function 3), we first obtain values of the random variable  $Y_i$  by generating a replicate data set `Y.rep[i]` which depends on the current values of `mu[]` and `tau` at each iteration. The `step()` function is then used to calculate the variable `p.smaller[i]` which takes the value 1 if  $Y[i] - Y.rep[i] \geq 0$  and zero otherwise. Hence the posterior mean of `p.smaller[i]` is simply the proportion of iterations for which  $Y.rep[i] < Y[i]$  (*i.e.*  $p(Y_i < y_i)$ ). It follows that  $p(Y_i \geq y_i) = 1 - \text{posterior mean of } p.smaller$ ; the chance of observing a more *extreme* value for  $Y_i$  is thus the minimum of these two probabilities. To compute the fifth checking function listed in section 9.3.1, we calculate the variable `p.inv[i]` (the inverse of the likelihood for observation  $i$ ) at each iteration. Having completed a BUGS run, we then calculate the inverse of the posterior mean of `p.inv[i]` to obtain the predictive ordinate  $p(y_i|y_{-i})$ . In addition, these values may be used to compute the negative cross-validatory log-likelihood discussed in section 9.3.2.

The model 'deviance' may be computed directly in the BUGS code by calculating the log-likelihood contribution (`log.like[i]`) for each observation, and then summing and multiplying by  $-2$ . Monitoring this sum over iterations will yield a posterior distribution for the deviance.

The replicate data set `Y.rep[]` mentioned earlier is also used to compute the Bayesian p-values described in section 9.3.3. Here we consider 2 different statistics for  $d(y)$  which may be sensitive to outlying observations in a Normal model. These are the coefficients of skewness and kurtosis, which are estimated by the average value of the third and fourth moments about the mean divided by the third and fourth powers of the standard deviation respectively. For samples from a Normal distribution, the skewness coefficient has expectation zero, whilst the kurtosis coefficient has expectation 3. Positive skewness indicates an extended right tail, whilst negative skewness implies an extended left tail. A kurtosis coefficient  $> 3$  indicates an excess of values near the mean and far from it, with a corresponding depletion elsewhere: this is the manner in which the  $t$ -distribution departs from the Normal. Values  $< 3$  result from distributions with a flatter top than Normal. For large sample sizes ( $n \approx 150$  for skewness and  $n \approx 1000$  for kurtosis), the sampling distributions of the coefficients are approximately Normal with known variances. For smaller sample sizes, these sampling distributions are not appropriate for testing whether the observed coefficients depart significantly from their expected values. The problem of an unknown sampling distribution may be overcome by implementing the ideas of Gelman *et al.* (1996), who suggest computing the *empirical* distribution of the relevant statistic  $d(y, \theta)$  conditional on current beliefs about the 'true' model parameters. In the `line` example, this is achieved by computing the average of the standardised third (`skew.rep`) and fourth (`kurtosis.rep`) moments using the replicate data set `Y.rep[]` at each iteration. These values are then compared to the relevant average moment (`skew.obs` or `kurtosis.obs`) computed from the observed data. The proportion of iterations for which `skew.obs < skew.rep` and `kurtosis.obs < kurtosis.rep` correspond to the Bayesian p-values for testing significant departures from the values of skewness and kurtosis expected under Normality.

The BUGS code below shows the declarations needed to implement each of the above checking criteria for the `line` example.

```

model line;
const
  N = 5, # number of observations
  PI = 3.141593;

var
  x[N], Y[N], mu[N], alpha, beta, tau, sigma, x.bar, resid[N], sresid[N],
  m3[N], m4[N], skew.obs, kurtosis.obs, p.skew, p.kurtosis, Y.rep[N],
  resid.rep[N], sresid.rep[N], m3.rep[N], m4.rep[N], skew.rep, kurtosis.rep,
  p.smaller, like[N], p.inv[N], log.like[N], deviance;

data in "line.dat";
inits in "line.in";

{
  # Model

  for (i in 1:N) {
    mu[i] <- alpha + beta*(x[i] - x.bar);
  }
}

```

```

Y[i] ~ dnorm(mu[i],tau);
}
x.bar <- mean(x[]);
alpha ~ dnorm(0.0,1.0E-4); beta ~ dnorm(0.0,1.0E-4);
tau ~ dgamma(1.0E-3,1.0E-3); sigma <- 1.0/sqrt(tau);

# Model checking

for (i in 1:N) {
# Residuals and moments for observed data
  resid[i] <- Y[i] - mu[i];           # Checking fn. 1
  sresid[i] <- resid[i]*sqrt(tau);    # Checking fn. 2

  m3[i] <- pow(sresid[i],3);
  m4[i] <- pow(sresid[i],4);

# Replicate data set
  Y.rep[i] ~ dnorm(mu[i],tau);
  p.smaller[i] <- step(Y[i]-Y.rep[i]); # Checking fn. 3

# Residuals and moments for replicate data
  resid.rep[i] <- Y.rep[i] - mu[i];
  sresid.rep[i] <- resid.rep[i]*sqrt(tau);

  m3.rep[i] <- pow(sresid.rep[i],3);
  m4.rep[i] <- pow(sresid.rep[i],4);

# Likelihood for each observed Y[i]
  like[i] <- sqrt(tau/(2*PI))*exp(-0.5*pow(sresid[i],2));
  p.inv[i] <- 1/like[i];            # For checking fn. 5
  log.like[i] <- log(like[i]);    # Log-likelihood for deviance
}

# Bayesian p-values
skew.obs <- sum(m3[])/N;      skew.rep <- sum(m3.rep[])/N;
p.skew <- step(skew.rep-skew.obs);

kurtosis.obs <- sum(m4[])/N;    kurtosis.rep <- sum(m4.rep[])/N;
p.kurtosis <- step(kurtosis.rep-kurtosis.obs);

# Deviance distribution
deviance <- -2*sum(log.like[]);
}

```

Table 6 gives the results for each of these model checks based on a 10000 iteration BUGS run.

Recall that we edited observation  $y_2$  to create an outlier. Each of the checking functions described in section 9.3.1 indicate that  $y_2$  is indeed the most outlying value:  $y_2$  has the largest residuals,

46

Residuals $y_i - E(Y_i)$			
resid[1]	-2.05	95% interval:	(-8.09, 4.03)
resid[2]	3.57	95% interval:	(-0.70, 7.74)
resid[3]	-0.82	95% interval:	(-4.41, 2.64))
resid[4]	-1.20	95% interval:	(-5.54, 3.12)
resid[5]	0.41	95% interval:	(-5.69, 6.52)
Standardized Residuals $(y_i - E(Y_i))/\sqrt{V(Y_i)}$			
sresid[1]	-0.74	95% interval:	(-2.44, 0.89)
sresid[2]	1.31	95% interval:	(-0.14, 2.91)
sreisd[3]	-0.30	95% interval:	(-1.21, 0.60)
sreisd[4]	-0.44	95% interval:	(-1.60, 0.67)
sresid[5]	0.15	95% interval:	(-1.36, 1.65)
Probability of more extreme observation $\min(p(Y_i \leq y_i), p(Y_i \geq y_i))$			
min(p.smaller[1], 1 - p.smaller[1])	0.29		
min(p.smaller[2], 1 - p.smaller[2])	0.15		
min(p.smaller[3], 1 - p.smaller[3])	0.40		
min(p.smaller[4], 1 - p.smaller[4])	0.35		
min(p.smaller[5], 1 - p.smaller[5])	0.45		
Predictive ordinate $p(y_i   \underline{y}_{\setminus i})$			
1/ p.inv[1]	0.039		
1/ p.inv[2]	0.008		
1/ p.inv[3]	0.010		
1/ p.inv[4]	0.086		
1/ p.inv[5]	0.069		
Negative cross-validatory log-likelihood $-\sum_{i=1}^n \log p(y_i   \underline{y}_{\setminus i})$			
$-\sum_{i=1}^5 \log(1/p.\text{inv}[i])$	15.47		
Bayesian p-values $p(d < d^{rep})$			
<i>d</i> = Skewness coefficient:			
skew.obs	0.45	95% interval:	(-3.06, 4.90)
skew.rep	-0.08	95% interval:	(-23.42, 21.49)
p.skew	0.44		
<i>d</i> = Kurtosis coefficient:			
kurtosis.obs	3.46	95% interval:	(0.06, 17.29)
kurtosis.rep	39.87	95% interval:	(0.06, 228.90)
p.kurtosis	0.70		
Deviance $-2 \log p(\underline{y}   \theta)$			
deviance	25.27	Minimum:	20.92

Table 6: Results of selected model diagnostics for the line example

47

the smallest probability of observing a more extreme value and the smallest predictive ordinate. However, the Bayesian p-values suggest that the observed skewness and kurtosis coefficients are consistent with the values expected from a random sample of 5 points from a Normal distribution. That is, there is no evidence against the assumption of Normal errors for this data. Notice the wide 95% credible intervals for the empirical distribution of `skew.rep` and `kurtosis.rep`, suggesting large sampling variation for these coefficients when based on small sample sizes.

The deviance and negative cross-validatory log-likelihood are intended for comparing 2 or more alternative models. To illustrate this, we fitted a second model to the `line` data, in which the observations were assumed to follow a Student's *t* distribution on 3 degrees of freedom. This yielded a posterior mean deviance of 24.56 (minimum = 19.71) and a negative cross-validatory log-likelihood of 15.44, all of which are slightly smaller than the corresponding values for the Normal model. This suggests that a *t*-distributed error structure on 3 degrees of freedom provides a marginally better fit to the edited `line` data than does a Normal error structure.

## 9.5 Ranking

~~Besag *et al.* (1995) emphasise how MCMC methods are particularly suitable for deriving posterior probabilities of complex functions of multiple parameters, simply by counting the proportion of iterations for which the specific condition obtains. A special but useful example is when inferring the rank order of a set of parameters, for example the mortality rates in a set of hospitals. This is described in detail in the surgical example, in which it is shown how to establish the rank order at each iteration by means of the step function.~~

## 9.6 Measurement error

~~There has been increasing acknowledgement of the importance of measurement error in epidemiology, and this has led to a large literature (Gail, 1990) on alternative methods for adjusting inferences based on the standard assumption of that measured covariates represent the true 'exposure' to whatever risk factor is of interest. Richardson and Gilks (1993, 1994) discuss the Bayesian approach to measurement error using graphical models and Gibbs sampling: here we shall indicate how alternative models for measurement error may be accommodated using BUGS, and also issue a warning about the lack of robustness that can result from a full Bayesian analysis.~~

~~Suppose we measure a covariate  $z_i$  on an individual  $i$ , but we feel this is only an approximation for the 'true' covariate value which we shall denote  $x_i$ . We consider two possible reasons for this approximation: first, that there is 'measurement error' either due to the measurement instrument used or due to random fluctuations around a long-term mean, and second that there is 'rounding error' due simply to recording with insufficient inaccuracy: an example of the latter is the notorious preference for numbers ending in 0 or 5 in records of blood pressure. Of course, it is possible for both of these causes to exist together.~~

~~For each of these causes, we can consider two possible structures relating the observed  $z_i$  to the true underlying  $x_i$ : first, in the 'classical error' model  $z_i$  depends on  $x_i$ , and we explicitly assume an 'exposure model', i.e. a probability distribution for  $x_i$  in the population; second, we can adopt the 'Berkson' model and consider the true  $x_i$  as depending on  $z_i$  - while this may seem odd at first it is essentially equivalent to assuming a uniform exposure distribution.~~

~~The four alternatives are briefly discussed below, with their possible representation in BUGS, and~~

## Acknowledgements

We are grateful to our colleagues David Clayton and Teresa Smith for help concerning the examples and the manual. Brad Carlin and David Mackay provided many useful comments on previous drafts of the manual.

## References

- Baker, S. G. (1995). The multinomial-poisson transformation. *The Statistician*, **43**, 495–504.
- Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley and Sons, Ltd., Chichester.
- Besag, J., Green, P. J., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems (with discussion). *Statistical Science*, **10**, 3–66.
- Best, N. G., Cowles, M. K., and Vines, S. K. (1995). *CODA: Convergence diagnosis and output analysis software for Gibbs sampling output, Version 0.3*. MRC Biostatistics Unit, Cambridge.
- Best, N. G., Spiegelhalter, D. J., Thomas, A., and Brayne, C. E. G. (1996). Bayesian analysis of realistically complex models. *J Roy Statist Soc A*, **159**, 323–42.
- Breslow, N. E. and Day, N. E. (1980). *Statistical Methods on Cancer Research Volume 1: Case-Control Studies*. International Agency for Cancer Research, Lyon.
- Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. *J Roy Statist Soc B*, **57**, 473–84.
- Carlin, J. B. (1992). Meta-analysis for 2 x 2 tables: a Bayesian approach. *Statistics in Medicine*, **11**, 141–59.
- Cox, D. R. and Wermuth, N. (1993). Linear dependencies represented by chain graphs. *Statistical Science*, **8**, 204–18.
- De Groot, M. E. (1970). *Optimal Statistical Decisions*. McGraw-Hill, New York.
- Dempster, A. P. (1974). The direct use of likelihood for significance testing. In *Proceedings of Conference on Foundational Questions in Statistical Inference*, (ed. O. Barndorff-Nielsen, P. Blaeild, and G. Schou), pp. 335–52. Department of Theoretical Statistics: University of Aarhus.
- DuMouchel, W. (1990). Bayesian meta-analysis. In *Statistical methodology in the pharmaceutical sciences*, (ed. D. Berry), pp. 509–29. Marcel Dekker, New York.
- DuMouchel, W. and Waternaux, C. (1992). Hierarchical models for combining information and for meta-analyses (discussion). In *Bayesian statistics 4*, (ed. J. Bernardo, J. Berger, A. Dawid, and A. Smith), pp. 338–41. Clarendon Press, Oxford.
- Frydenberg, M. (1990). The chain graph Markov property. *Scandinavian Journal of Statistics*, **17**, 333–53.
- Gail, M. H. (1990). A bibliography and comments on the use of statistical models in epidemiology in the 1980's. *Statistics in Medicine*, **10**, 1819–85.
- Geisser, S. and Eddy, W. (1979). A predictive approach to model selection. *J Amer Statist Assoc*, **74**, 153–60.
- Gelfand, A. E. and Dey, D. K. (1994). Bayesian model choice: asymptotics and exact calculations. *J Roy Statist Soc B*, **56**.
- Gelfand, A. E., Dey, D. K., and Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods. In *Bayesian statistics 4*, (ed. J. M.

- Bernardo, J. O., Berger, A. P., Dawid, and A. F. M. Smith), pp. 147–68. Oxford University Press.
- Gelfand, A. E., Sahu, S. K., and Carlin, B. P. (1995). Efficient parameterizations for normal linear models. *Biometrika*, **82**, 479–88.
- Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J Amer Statist Assoc*, **85**, 398–409.
- Gelman, A., Meng, X. L., and Stern, H. S. (1996). Bayesian tests for goodness of fit using tail area probabilities. *Statistica Sinica*. (to appear).
- Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, **7**, 457–72.
- Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In *Bayesian Statistics 4*, (ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith). Clarendon Press, Oxford, UK.
- Gilks, W. R. (1992). Derivative-free adaptive rejection sampling for gibbs sampling. In *Bayesian Statistics 4*, (ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith), pp. 169–94. Clarendon Press, Oxford, UK.
- Gilks, W. R., Clayton, D. G., Spiegelhalter, D. J., Best, N. G., McNeil, A. J., Sharples, L. D., and Kirby, A. J. (1993). Modelling complexity: applications of Gibbs sampling in medicine. *J Roy Statist Soc B*, **55**, 39–52.
- Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (ed.) (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall, New York.
- Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994). A language and program for complex Bayesian modelling. *The Statistician*, **43**, 169–78.
- Gilks, W. R., Wang, C. C., Yvonne, B., and Coursaget, P. (1992). Random-effects models for longitudinal data using gibbs sampling. *Biometrics*, **48**.
- Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, **41**, 337–48.
- Kass, R. and Raftery, A. (1995). Bayes factors and model uncertainty. *J Amer Statist Assoc*, **90**, 773–95.
- MacMahon, S., Peto, R., Cutler, J., Collins, R., Sorlie, P., Neaton, J., Abbott, R., Godwin, J., Dyer, A., and Stamler, J. (1990). Blood pressure, stroke, and coronary heart disease. Part 1, prolonged differences in blood pressure: prospective observational studies corrected for the regression dilution bias. *The Lancet*, **335**, 765–74.
- Richardson, S. and Gilks, W. R. (1993). Conditional independence models for epidemiological studies with covariate measurement error. *Statistics in Medicine*, **12**, 1703–22.
- Richardson, S. and Gilks, W. R. (1994). A bayesian approach to measurement error problems in epidemiological studies using conditional independence models. *American Journal of Epidemiology*, (to appear).
- Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics*, **12**, 1151–72.
- Smith, A. F. M. (1991). Discussion of ‘Posterior Bayes factors’ by M Aitken. *J Roy Statist Soc B*, **53**, 132–3.
- Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion). *J Roy Statist Soc B*, **55**, 3–24.

A handwritten circled number 50.

- Smith, T. C., Spiegelhalter, D. J., and Thomas, A. (1995). Bayesian graphical modelling applied to random effects meta-analysis. *Statistics in Medicine*, **14**, 2685–99.
- Spiegelhalter, D. J., Best, N. G., Gilks, W. R., and Inskip, H. (1995a). Hepatitis: a case study in MCMC methods. In *Markov chain Monte Carlo Methods in practice*, (ed. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter), pp. 21–43. Chapman and Hall, New York.
- Spiegelhalter, D. J., Thomas, A., and Best, N. G. (1995b). Computation on Bayesian graphical models. In *Bayesian Statistics 5*. Clarendon Press, Oxford, UK.
- Spiegelhalter, D. J., Thomas, A., Best, N. G., and Gilks, W. R. (1995c). *BUGS: Bayesian inference Using Gibbs Sampling, Version 0.50*. MRC Biostatistics Unit, Cambridge.
- Wermuth, N. and Lauritzen, S. L. (1990). On substantive research hypotheses, conditional independence graphs and graphical chain models (with discussion). *J Roy Statist Soc B*, **52**, 21–72.
- Zeger, S. L. and Karim, M. R. (1991). Generalized linear models with random effects: a Gibbs sampling approach. *J Amer Statist Assoc*, **86**, 79–86.