Bayesian Hierarchical Modeling

3: Bayesian Hierarchical and Mixture Modeling

David Draper

Department of Applied Mathematics and Statistics
University of California, Santa Cruz

and (1 Jul 2014–30 Sep 2015) *eBay Research Labs*

{draper@ams.ucsc.edu, davdraper@ebay.com}
www.ams.ucsc.edu/∼draper

Short Course (Day 2)
University of Reading (UK)

26 Nov 2014

Hierarchical Model Selection: A Case Study

Case Study: *In-home geriatric assessment (IHGA)*. In an experiment conducted in the 1980s (Hendriksen et al. 1984), 572 elderly people living in a number of villages in Denmark were randomized, 287 to a **control** (C) group (who received standard care) and 285 to an **experimental** (E) group (who received standard care plus IHGA: a kind of **preventive medicine** in which each person's medical and social needs were assessed and acted upon individually).

One important outcome was the number of **hospitalizations** during the two-year life of the study (Table 4.1).

Table 4.1. Distribution of number of hospitalizations in the IHGA study over a two-year period.

	Number of Hospitalizations										
Group	0	1	2	3	4	5	6	7	n	Mean	SD
Control	138	77	46	12	8	4	0	2	287	0.944	1.24
Experimental	147	83	37	13	3	1	1	0	285	0.768	1.01

Evidently IHGA lowered the mean hospitalization rate (for these elderly Danish people, at least) by $(0.944 - 0.768) = \mathbf{0.176}$, which is about a $100\left(\frac{0.768-0.944}{0.944}\right) = \mathbf{19\%}$ reduction from the control level, a difference that's **large in clinical terms**.

Modeling the IHGA Data

An **off-the-shelf** analysis of this experiment might pretend (**Model 0**) that the data are Gaussian,

$$\begin{aligned} \left(C_i|\mu_C,\sigma_C^2\right) &\overset{\text{IID}}{\sim} N\left(\mu_C,\sigma_C^2\right), i = 1,\ldots,n_C, \\ \left(E_j|\mu_E,\sigma_E^2\right) &\overset{\text{IID}}{\sim} N\left(\mu_E,\sigma_E^2\right), j = 1,\ldots,n_E, \end{aligned} \quad (35)$$

and use the ordinary frequentist **two-independent-samples "z-machinery"**:

```
rosalind 15> R

R : Copyright 2001, The R Development Core Team
Version 1.2.1  (2001-01-15)

> C <- c( rep( 0, 138 ), rep( 1, 77 ), rep( 2, 46 ),
    rep( 3, 12 ), rep( 4, 8 ), rep( 5, 4 ), rep( 7, 2 ) )

> print( n.C <- length( C ) )

[1] 287                 # sample size in the control group

> mean( C )

[1] 0.9442509           # control group mean

> sqrt( var( C ) )

[1] 1.239089            # control group
                        # standard deviation (SD)
> table( C )

   0  1  2  3 4 5 7     # control group
 138 77 46 12 8 4 2     # frequency distribution
```

Analysis of Model 0

```
> E <- c( rep( 0, 147 ), rep( 1, 83 ), rep( 2, 37 ),
    rep( 3, 13 ),rep( 4, 3 ), rep( 5, 1 ), rep( 6, 1 ) )

> print( n.E <- length( E ) )

[1] 285                 # sample size in the
                        # experimental group
> mean( E )

[1] 0.7684211           # experimental group mean

> sqrt( var( E ) )

[1] 1.008268            # experimental group SD

> table( E )

   0  1  2  3 4 5 6     # experimental group
 147 83 37 13 3 1 1     # frequency distribution

> print( effect <- mean( E ) - mean( C ) )

[1] -0.1758298          # mean difference ( E - C )

> effect / mean( C )

[1] -0.1862109          # relative difference ( E - C ) / C

> SE.effect <- sqrt( var( C ) / n.C + var( E ) / n.E )

[1] 0.09442807          # standard error of the difference

> print( CI <- c( effect - 1.96 * SE.effect,
    effect + 1.96 * SE.effect ) )

[1] -0.3609 0.009249   # the 95% confidence interval from
                       # model 0 runs from -.36 to +.01
```

Deficiencies of Model 0

The frequentist analysis of Model 0 is equivalent to a Bayesian analysis of the same model with **diffuse priors** on the control and experimental group means and SDs $(\mu_C, \sigma_C, \mu_E, \sigma_E)$, and is summarized in Table 4.2.

Table 4.2. Summary of analysis of Model 0.

	Posterior		
	Mean	SD	95% Interval
Treatment effect $(\mu_E - \mu_C)$	-0.176	0.0944	$(-0.361, 0.009)$

However, both distributions have long right-hand tails; in fact they look rather **Poisson**.

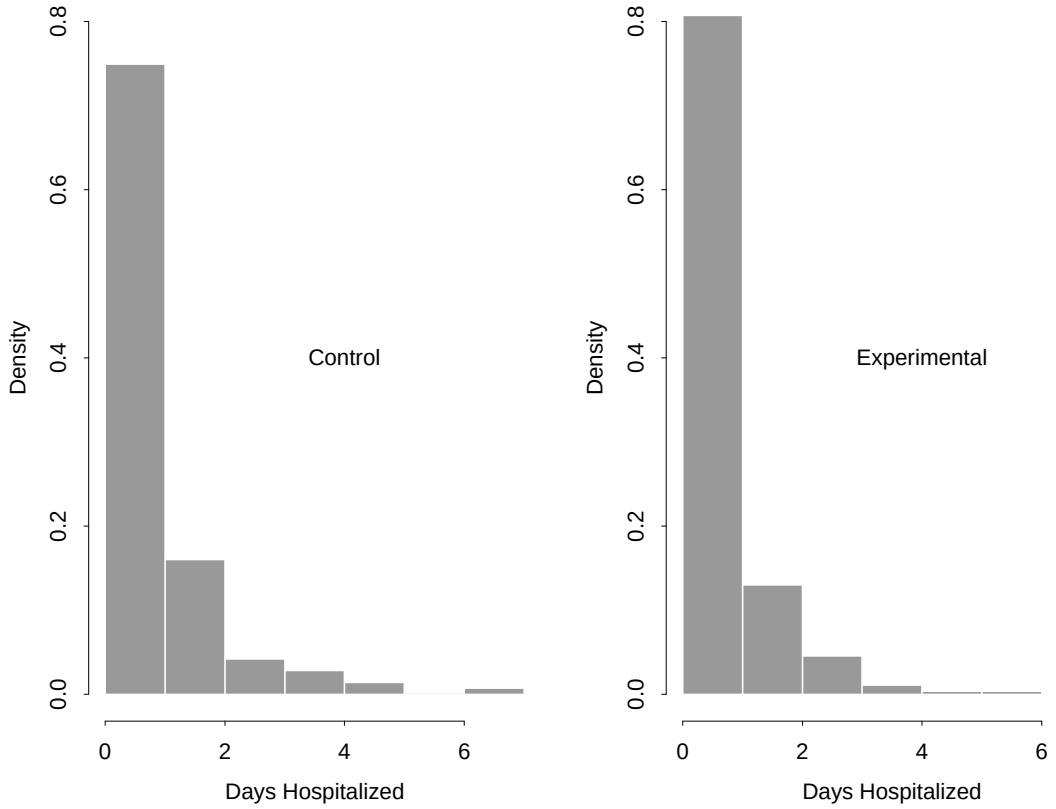


Figure 4.1. Histograms of control and experimental numbers of hospitalizations.

4.1.1 Poisson Fixed-Effects Modeling

R code to make the histograms:

```
> x11( )                              # to open a
                                      #   graphics window
> par( mfrow = c( 1, 2 ) )            # to plot two histograms

> hist( C, nclass = 8, probability = T,
    xlab = 'Days Hospitalized', ylab = 'Density',
    xlim = c( 0, 7 ), ylim = c( 0, 0.8 ) )

> text( 4, 0.4, 'Control' )

> hist( E, nclass = 8, probability = T,
    xlab = 'Days Hospitalized', ylab = 'Density',
    xlim = c( 0, 7 ), ylim = c( 0, 0.8 ) )

> text( 4, 0.4, 'Experimental' )
```

So I **created** a classicBUGS file called poisson1.bug that looked like this:

```
model poisson1;

const

  n.C = 287, n.E = 285;

var

  lambda.C, lambda.E, C[ n.C ], E[ n.E ], effect;

data C in "poisson-C.dat", E in "poisson-E.dat";

inits in "poisson1.in";
```

Initial Poisson Modeling (continued)

```
{

  lambda.C ~ dgamma( 0.001, 0.001 );
  lambda.E ~ dgamma( 0.001, 0.001 );

  for ( i in 1:n.C ) {

    C[ i ] ~ dpois( lambda.C );

  }

  for ( j in 1:n.E ) {

    E[ j ] ~ dpois( lambda.E );

  }

  effect <- lambda.E - lambda.C;

}
```

`poisson1.in` initializes both λ_C and λ_E to 1.0; the $\Gamma(0.001, 0.001)$ priors for λ_C and λ_E are chosen (as usual to create diffuseness) to be **flat** in the region in which the **likelihood** is **appreciable**:

```
> sqrt( var( C ) / n.C )

[1] 0.07314114

> sqrt( var( E ) / n.E )

[1] 0.05972466

> c( mean( C ) - 3.0 * sqrt( var( C ) / n.C ),
    mean( C ) + 3.0 * sqrt( var( C ) / n.C ) )
```

Initial Poisson Modeling (continued)

```
[1] 0.7248275 1.1636743

> c( mean( E ) - 3.0 * sqrt( var( E ) / n.E ),
    mean( E ) + 3.0 * sqrt( var( E ) / n.E ) )

[1] 0.5892471 0.9475950

> lambda.grid <- seq( 0.01, 2.0, 0.01 )

> plot( lambda.grid, 0.001 * dgamma( lambda.grid, 0.001 ),
    type = 'l', xlab = 'Lambda', ylab = 'Density' )
```

The likelihood under the Gaussian model is **concentrated** for λ_C from about 0.7 to 1.2, and that for λ_E from about 0.6 to 1; you can see from the plot that across those ranges the $\Gamma(0.001, 0.001)$ prior is **essentially constant**.

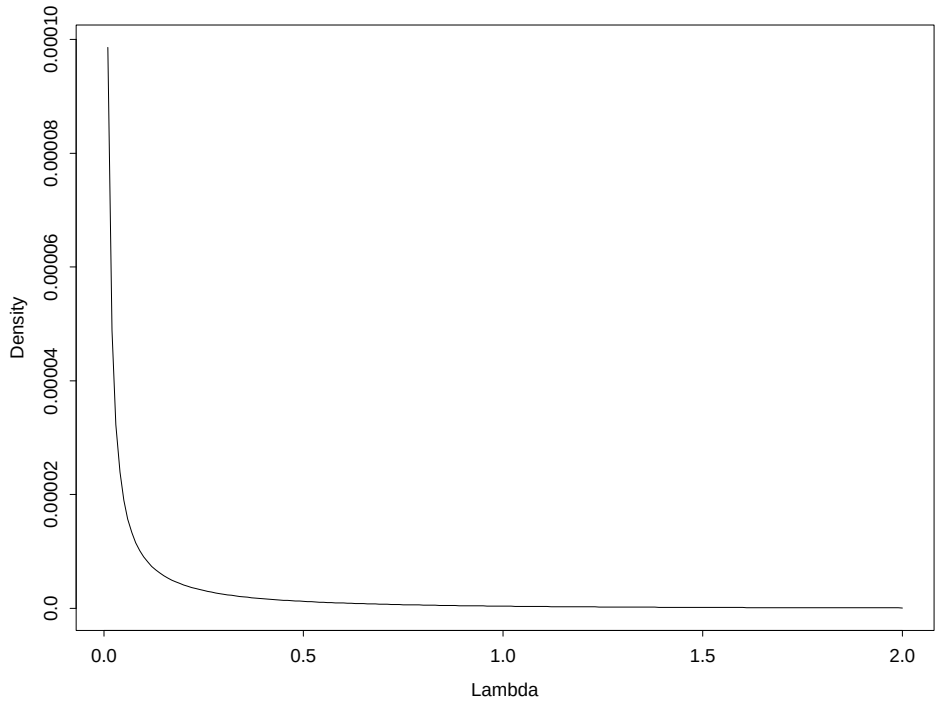
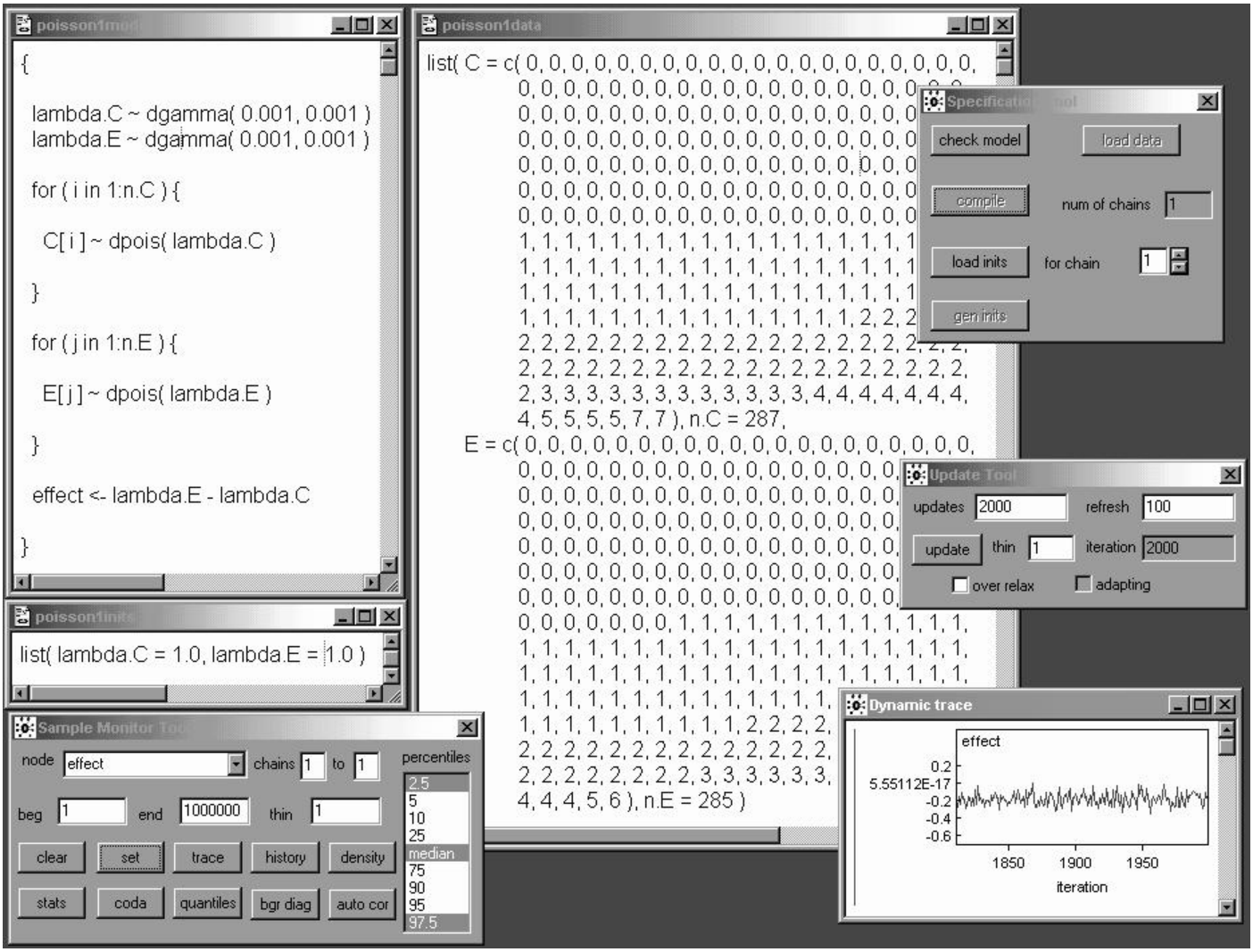


Figure 4.2. The $\Gamma(0.001, 0.001)$ distribution in the region in which the likelihoods for λ_C and λ_E are appreciable.

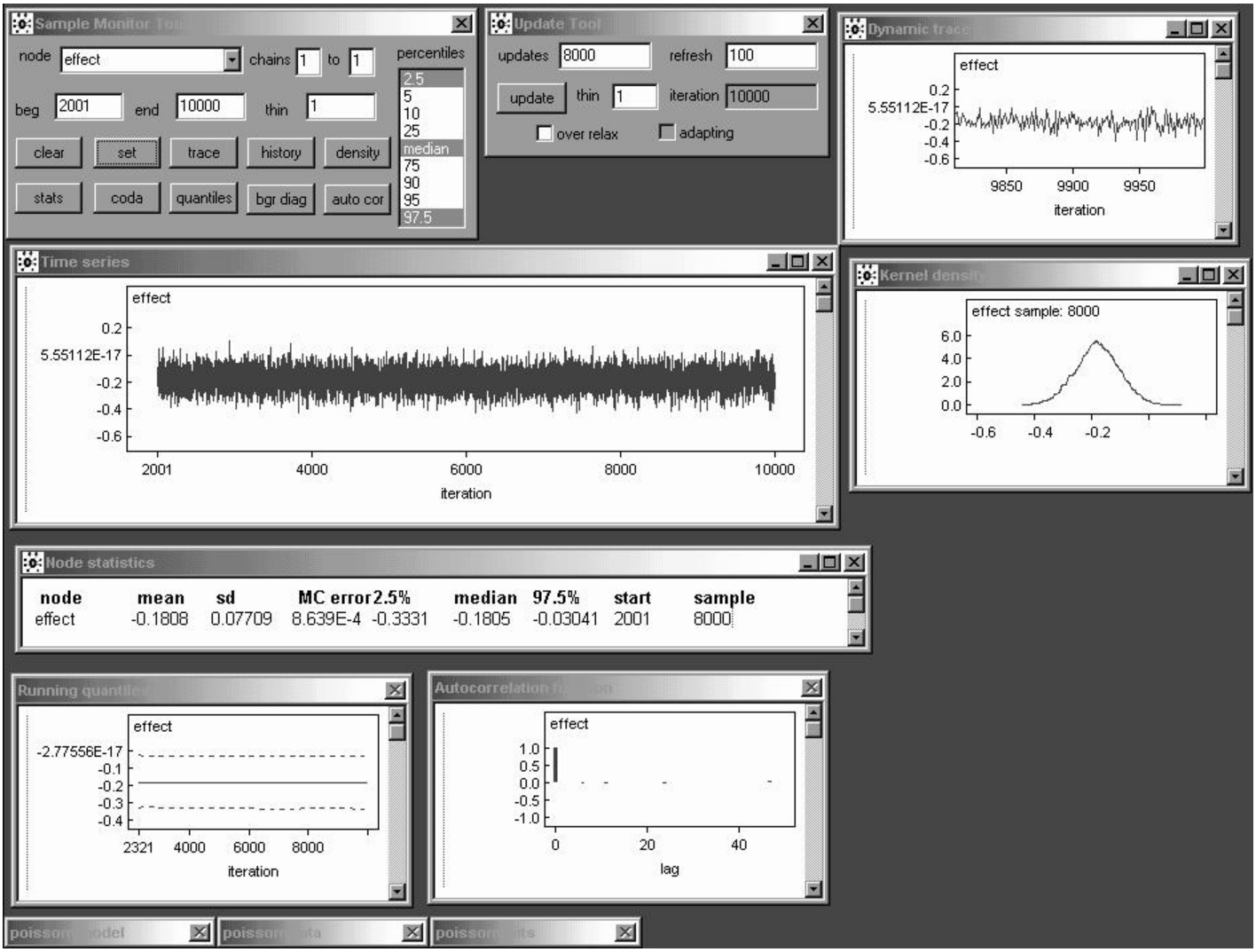
WinBUGS Implementation



The **screendump** above presents part of the results of fitting the **2-independent-samples additive Poisson model** at the top of page 8 in WinBUGS.

A **burn-in of 2,000** was almost **instantaneous at 2.0 PC GHz** and revealed **good mixing** for the three main quantities of interest.

WinBUGS Implementation (continued)



A monitoring run of 8,000 reveals that the effect parameter in the **2-independent-samples Poisson model** is behaving like **white noise**, so that already with only 8,000 iterations the posterior mean has a Monte Carlo standard error of **less than 0.001**.

Initial Poisson Modeling (continued)

Thus a burn-in of 2,000 and a monitoring run of 8,000 yields **good MCMC diagnostics** and permits a comparison between model 0 (Gaussian) and model 1 (Poisson), as in Table 4.3.

Table 4.3. Comparison of inferential conclusions from models 0 and 1.

$\boxed{\lambda_C}$ Model	Posterior Mean	Posterior SD	Central 95% Interval
Gaussian	0.944	0.0731	$(0.801, 1.09)$
Poisson	0.943	0.0577	$(0.832, 1.06)$

$\boxed{\lambda_E}$ Model	Posterior Mean	Posterior SD	Central 95% Interval
Gaussian	0.768	0.0597	$(0.651, 0.885)$
Poisson	0.769	0.0521	$(0.671, 0.875)$

$\boxed{\Delta = \lambda_E - \lambda_C}$ Model	Posterior Mean	Posterior SD	Central 95% Interval
Gaussian	-0.176	0.0944	$(-0.361, 0.009)$
Poisson	-0.174	0.0774	$(-0.325, -0.024)$

The two models produce **almost identical point estimates**, but the Poisson model leads to **sharper inferences** (e.g., the posterior SD for the treatment effect $\Delta = \lambda_E - \lambda_C$ is **22%** larger in model 0 than in model 1).

Additive and Multiplicative Treatment Effects

This is the same point we noticed with the NB10 data—when a location parameter is the only thing at issue, the Gaussian is a **conservative** modeling choice (intuitively, the Poisson gains its "extra accuracy" from the variance and the mean being equal, which permits **second-moment** information to help in estimating the λ values along with the usual first-moment information).

Both the Gaussian and Poisson models so far implicitly assume that the treatment effect is **additive**:

$$E \stackrel{\text{st}}{=} C + \text{effect}, \tag{36}$$

where $\stackrel{\text{st}}{=}$ means *is stochastically equal to*; in other words, apart from random variation the effect of the IHGA is to **add or subtract a constant** to or from each person's underlying rate of hospitalization.

However, since the outcome variable is non-negative, it is plausible that a **better model** for the data is

$$E \stackrel{\text{st}}{=} (1 + \text{effect})\, C. \tag{37}$$

Additive vs. Multiplicative Effect

Here the treatment effect is **multiplicative**—in other words, apart from random variation the effect of the IHGA is to **multiply** each person's underlying rate of hospitalization by a constant above or below 1.

A **qqplot** of the control and experimental outcome values can in some cases be helpful in choosing between additive and multiplicative models:

```
> CEqq <- qqplot( C, E, plot = F )

> table( CEqq$y, CEqq$x )

                Interpolated C values
     0 0.965  1 1.5  2 2.82 3 3.91 4 4.96 5 6.99 7
  0 137     1  9   0  0    0 0    0 0    0 0    0 0
  1   0     0 66   1 16    0 0    0 0    0 0    0 0
  2   0     0  0   0 29    1 7    0 0    0 0    0 0
E 3   0     0  0   0  0    0 4    1 7    1 0    0 0
  4   0     0  0   0  0    0 0    0 0    0 3    0 0
  5   0     0  0   0  0    0 0    0 0    0 0    1 0
  6   0     0  0   0  0    0 0    0 0    0 0    0 1

> symbols( c( 0, 0.964798, 1, 1, 1.5, 2, 2, 2.823944, 3, 3,
    3.908447, 4, 4.964813, 5, 6.985962, 7 ), c( rep( 0, 3 ),
    rep( 1, 3 ), rep( 2, 3 ), rep( 3, 4 ), 4, 5, 6 ),
    circles = c( 137, 1, 9, 66, 1, 16, 29, 1, 7, 4, 1, 7, 1,
    3, 1, 1 ), xlab = 'C', ylab = 'E' )
```

Additive vs. Multiplicative Effect

```
> abline( 0, 1 )                    # E = C (no effect)

> abline( 0, 0.793, lty = 2 )       # E = 0.816 C
                                    #   (multiplicative)

> abline( -0.174, 1, lty = 3 )      # E = C - 0.174 (additive)
```

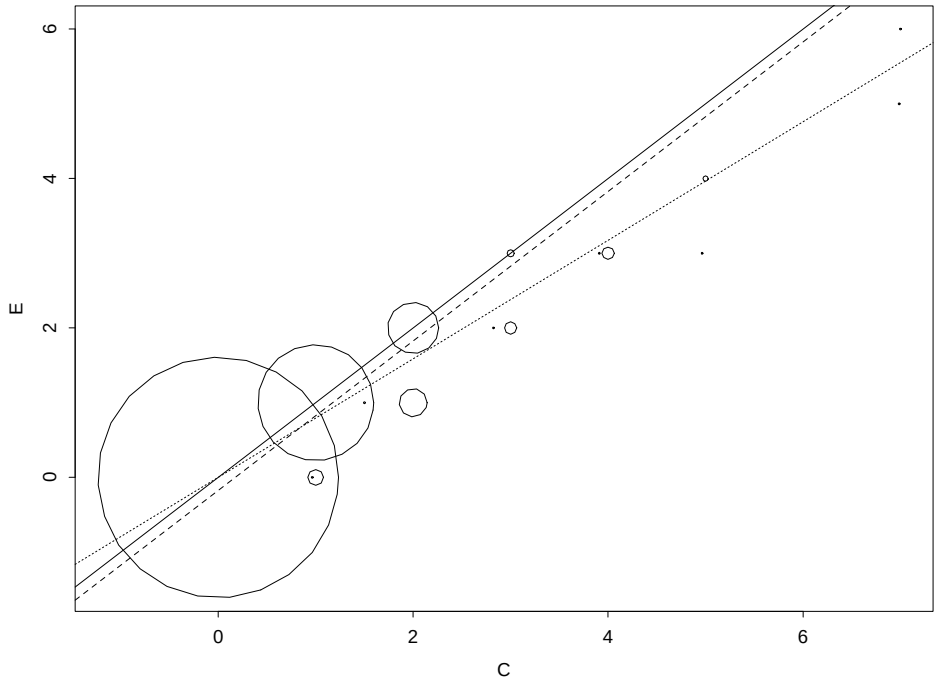


Figure 4.3. QQplot of E versus C values, with the radii of the plotted circles proportional to the number of observations at the indicated point. The solid line corresponds to no treatment effect, the small dotted line to the best-fitting multiplicative model ($E \stackrel{\text{st}}{=} 0.816\,C$), and the large dotted line to the best-fitting additive model ($E \stackrel{\text{st}}{=} C - 0.174$).

Here, because the Poisson model has only **one parameter** for both location and scale, the multiplicative and additive formulations **fit equally well**, but the multiplicative model **generalizes** more readily (see below).

A Multiplicative Poisson Model

A simple way to write the multiplicative model is to re-express the data in the form of a **regression** of the outcome y on a **dummy variable** x which is 1 if the person was in the experimental group and 0 if he/she was in the control group:

i	1	2	$\cdots$	287	288	289	$\cdots$	572
x_i	0	0	$\cdots$	0	1	1	$\cdots$	1
y_i	1	0	$\cdots$	2	0	3	$\cdots$	1

Then for $i = 1, \ldots, n = 572$ the **multiplicative** model can be written

$$\begin{array}{rcl} (y_i \,|\lambda_i) & \overset{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \gamma_0 + \gamma_1 x_i \\ (\gamma_0, \gamma_1) & \sim & \text{diffuse} \end{array} \tag{38}$$

In this model the **control** people have

$$\log(\lambda_i) = \gamma_0 + \gamma_1(0) = \gamma_0, \quad \text{i.e.,} \quad \lambda_C = e^{\gamma_0}, \tag{39}$$

and the **experimental** people have

$$\begin{array}{rcl} \log(\lambda_i) & = & \gamma_0 + \gamma_1(1) = \gamma_0 + \gamma_1, \quad \text{i.e.,} \\ \lambda_E & = & e^{\gamma_0+\gamma_1} = e^{\gamma_0} e^{\gamma_1} = \lambda_C e^{\gamma_1}. \end{array} \tag{40}$$

Now you may remember from basic **Taylor series** that for γ_1 not too far from 0

$$e^{\gamma_1} \doteq 1 + \gamma_1, \tag{41}$$

A Multiplicative Poisson Model

so that finally (for γ_1 **fairly near 0**)

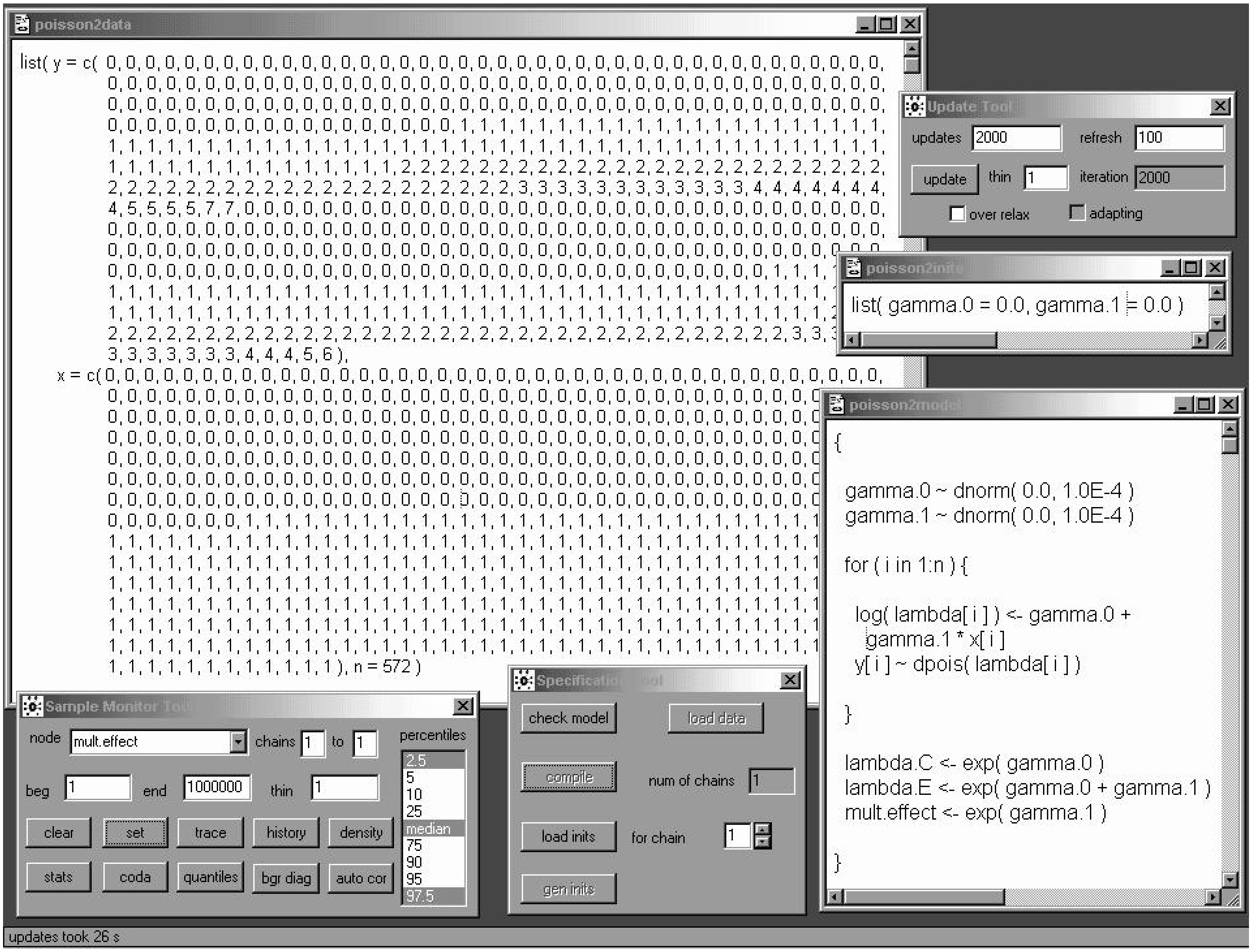
$$\lambda_E \doteq (1 + \gamma_1)\,\lambda_C, \tag{42}$$

which is a way of expressing equation (3) in **Poisson language**.

Fitting this model in BUGS is easy:

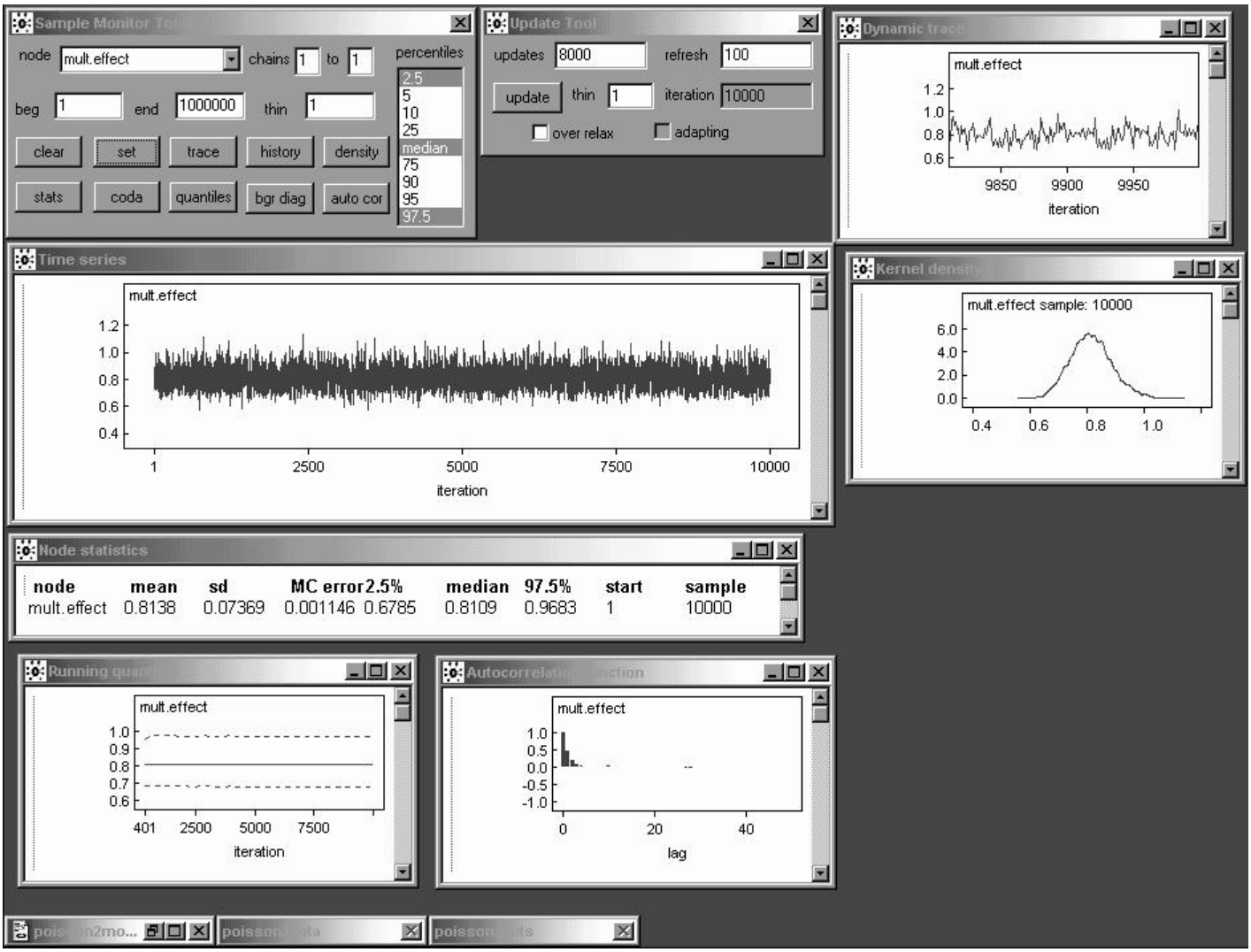
```
model poisson2;

const

  n = 572;

var

  gamma.0, gamma.1, lambda[ n ], x[ n ], y[ n ], lambda.C,
  lambda.E, mult.effect;

data x in "poisson-x.dat", y in "poisson-y.dat";
inits in "poisson2.in";

{
  gamma.0 ~ dnorm( 0.0, 1.0E-4 );    # flat priors for
  gamma.1 ~ dnorm( 0.0, 1.0E-4 );    # gamma.0 and gamma.1

  for ( i in 1:n ) {

    log( lambda[ i ] ) <- gamma.0 + gamma.1 * x[ i ];
    y[ i ] ~ dpois( lambda[ i ] );

  }

  lambda.C <- exp( gamma.0 );
  lambda.E <- exp( gamma.0 + gamma.1 );
  mult.effect <- exp( gamma.1 );
}
```

WinBUGS Implementation (continued)



The **multiplicative Poisson model** (11) takes longer to run—2,000 burn-in iterations now take about **4 seconds at 2.0 PC GHz**—but still exhibits **fairly good mixing**, as we'll see below.

WinBUGS Implementation (continued)



A total of **10,000 iterations** (the chain started essentially in equilibrium, so the burn-in can be absorbed into the monitoring run) reveals that the **multiplicative effect parameter** e^{γ_1} in model (11) behaves like an AR_1 series with $\widehat{\rho}_1 \doteq 0.5$, but the Monte Carlo standard error for the posterior mean is still only about **0.001** with a run of this length.

Additive versus Multiplicative Fit

A burn-in of 2,000 and a monitoring run of 8,000 again yields **good MCMC diagnostics** and permits a comparison between the additive and multiplicative Poisson models, as in Table 4.4.

Table 4.4. Comparison of inferential conclusions from the additive and multiplicative Poisson models.

$\boxed{\lambda_C}$ Model	Posterior Mean	Posterior SD	Central 95% Interval
additive	0.943	0.0577	$(0.832, 1.06)$
multiplicative	0.945	0.0574	$(0.837, 1.06)$

$\boxed{\lambda_E}$ Model	Posterior Mean	Posterior SD	Central 95% Interval
additive	0.769	0.0521	$(0.671, 0.875)$
multiplicative	0.768	0.0518	$(0.671, 0.872)$

$\boxed{\text{effect}}$ Model	Posterior Mean	Posterior SD	Central 95% Interval
additive	-0.174	0.0774	$(-0.325, -0.024)$
multiplicative	-0.184	0.0743	$(-0.324, -0.033)$

With this model it is as if the experimental people's average underlying rates of hospitalization have been **multiplied by 0.82**, give or take about 0.07.

The additive and multiplicative effects are **similar** here, because both are not too far from zero.

Extra-Poisson Variability

However, none of this has verified that the **Poisson model is reasonable** for these data—the histograms show that the Gaussian model is clearly unreasonable, but the diagnostic plots in WinBUGS and CODA only check on the adequacy of the **MCMC** sampling, not the model.

In fact we had a good clue that the data are **not** Poisson back on page 2: as noted in part 2, the Poisson(λ) distribution has mean λ and also variance λ—in other words, the **variance-to-mean-ratio** (VTMR) for the Poisson is 1. But

```
> var( C ) / mean( C )
[1] 1.62599
> var( E ) / mean( E )
[1] 1.322979
```

i.e., the data exhibit **extra-Poisson variability** (VTMR > 1).

This actually **makes good sense** if you think about it, as follows.

The Poisson model assumes that everybody in the control group has the **same underlying rate** λ_C of hospitalization, and similarly everybody in the experimental group has the **same rate** λ_E.

Unobserved Predictor Variables

In reality it's far more reasonable to think that each person has his/her **own** underlying rate of hospitalization that depends on **baseline health status**, **age**, and various other things.

Now Hendriksen **forgot to measure** (or at least to report on) these other variables (he may have hoped that the randomization would balance them between C and E)—the only predictor we have is x, the **experimental status dummy variable**—so the best we can do is to lump all of these other **unobserved** predictor variables together into a kind of **"error" term** e.

This amounts to **expanding** the second Poisson model (11) above: for $i = 1, \ldots, n = 572$ the new model is

$$\begin{array}{rcll} (y_i \,|\lambda_i) & \overset{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) & \\ \log(\lambda_i) & = & \gamma_0 + \gamma_1 x_i + e_i & (43) \\ e_i & \overset{\text{IID}}{\sim} & N\left(0, \sigma_e^2\right) & \\ \left(\gamma_0, \gamma_1, \sigma_e^2\right) & \sim & \text{diffuse.} & \end{array}$$

Random-
Effects Poisson Regression

The **Gaussian** choice for the error distribution is **conventional**, not dictated by the science of the problem (although if there were a lot of unobserved predictors hidden inside the e_i their weighted sum would be close to normal by the **Central Limit Theorem**).

Model (16) is an **expansion** of the earlier model (11) because you can obtain model (11) from (16) by setting $\sigma_e^2 = 0$, whereas with (16) we're letting σ_e^2 vary and **learning about it from the data**.

The addition of the **random effects** e_i to the model is one way to address the extra-Poisson variability: this model would be called a **lognormal mixture of Poisson distributions** (or a **random effects Poisson regression** (REPR) model) because it's as if each person's λ is drawn from a lognormal distribution and then his/her number of hospitalizations y is drawn from a Poisson distribution with his/her λ, and this mixing process will make the variance of y **bigger than its mean**.

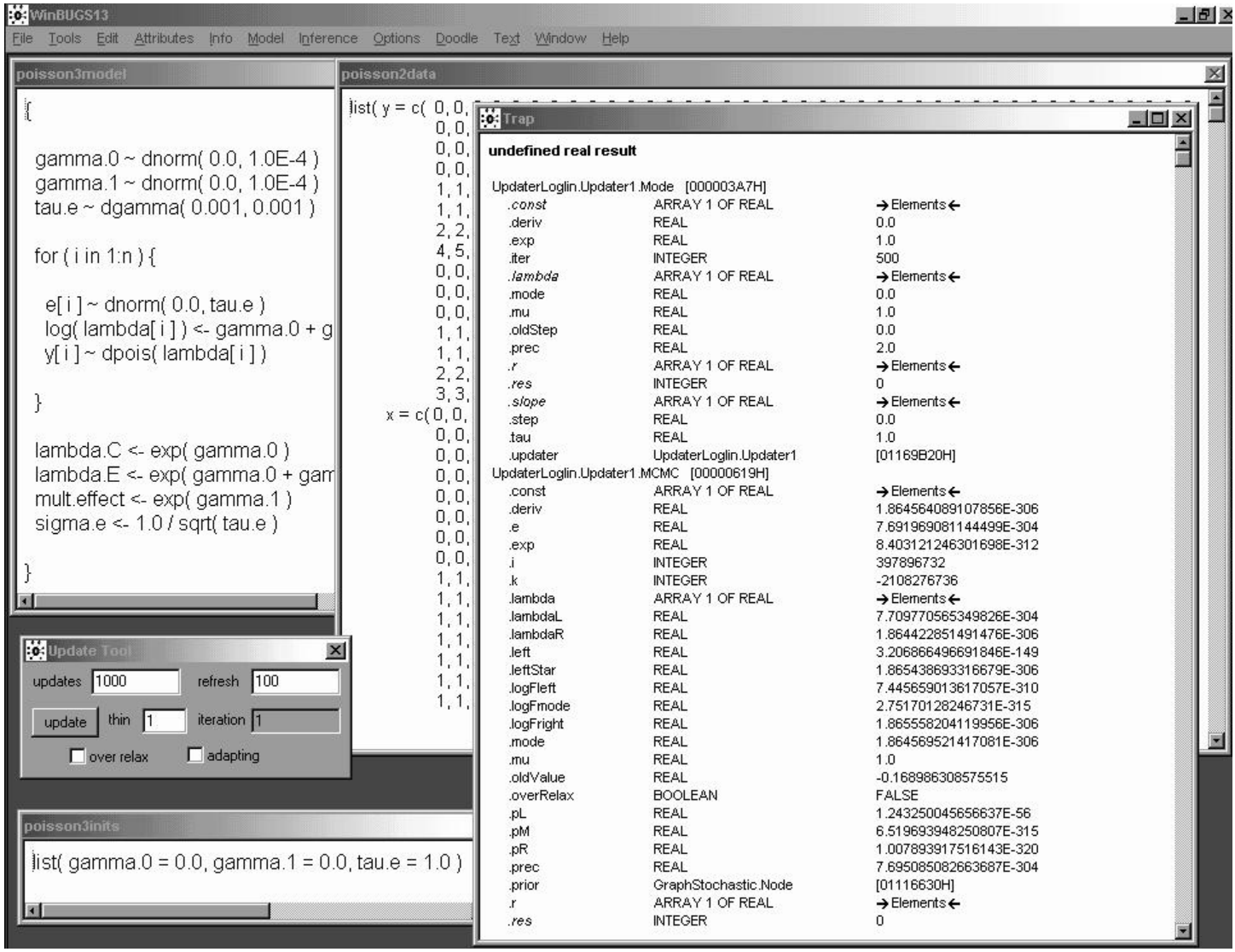
WinBUGS Implementation

The new WinBUGS model is

```
{

  gamma.0 ~ dnorm( 0.0, 1.0E-4 )
  gamma.1 ~ dnorm( 0.0, 1.0E-4 )
  tau.e ~ dgamma( 0.001, 0.001 )

  for ( i in 1:n ) {

    e[ i ] ~ dnorm( 0.0, tau.e )
    log( lambda[ i ] ) <- gamma.0 + gamma.1 * x[ i ] +
      e[ i ]
    y[ i ] ~ dpois( lambda[ i ] )

  }

  lambda.C <- exp( gamma.0 )
  lambda.E <- exp( gamma.0 + gamma.1 )
  mult.effect <- exp( gamma.1 )
  sigma.e <- 1.0 / sqrt( tau.e )

}
```

I again use a **diffuse** $\Gamma(\epsilon, \epsilon)$ prior (with $\epsilon = 0.001$) for the **precision** τ_e of the random effects.

WinBUGS Implementation (continued)



With a model like that in equation (16), there are n **random effects** e_i that need to be sampled as nodes in the graph (the e_i play the role of **auxiliary variables** in the MCMC) along with the fixed effects (γ_0, γ_1) and the variance parameter σ_e^2.

In earlier releases of the software, at least, this made it more crucial to give WinBUGS **good starting values**.

Here WinBUGS release **1.3** has figured out that random draws like $\boxed{1.66 \cdot 10^{-316}}$ result from the **generic** (and quite poor) initial values $(\gamma_0, \gamma_1, \tau_e) = (0.0, 0.0, 1.0)$ and has **refused to continue sampling**.

Sensitivity to Initial Values

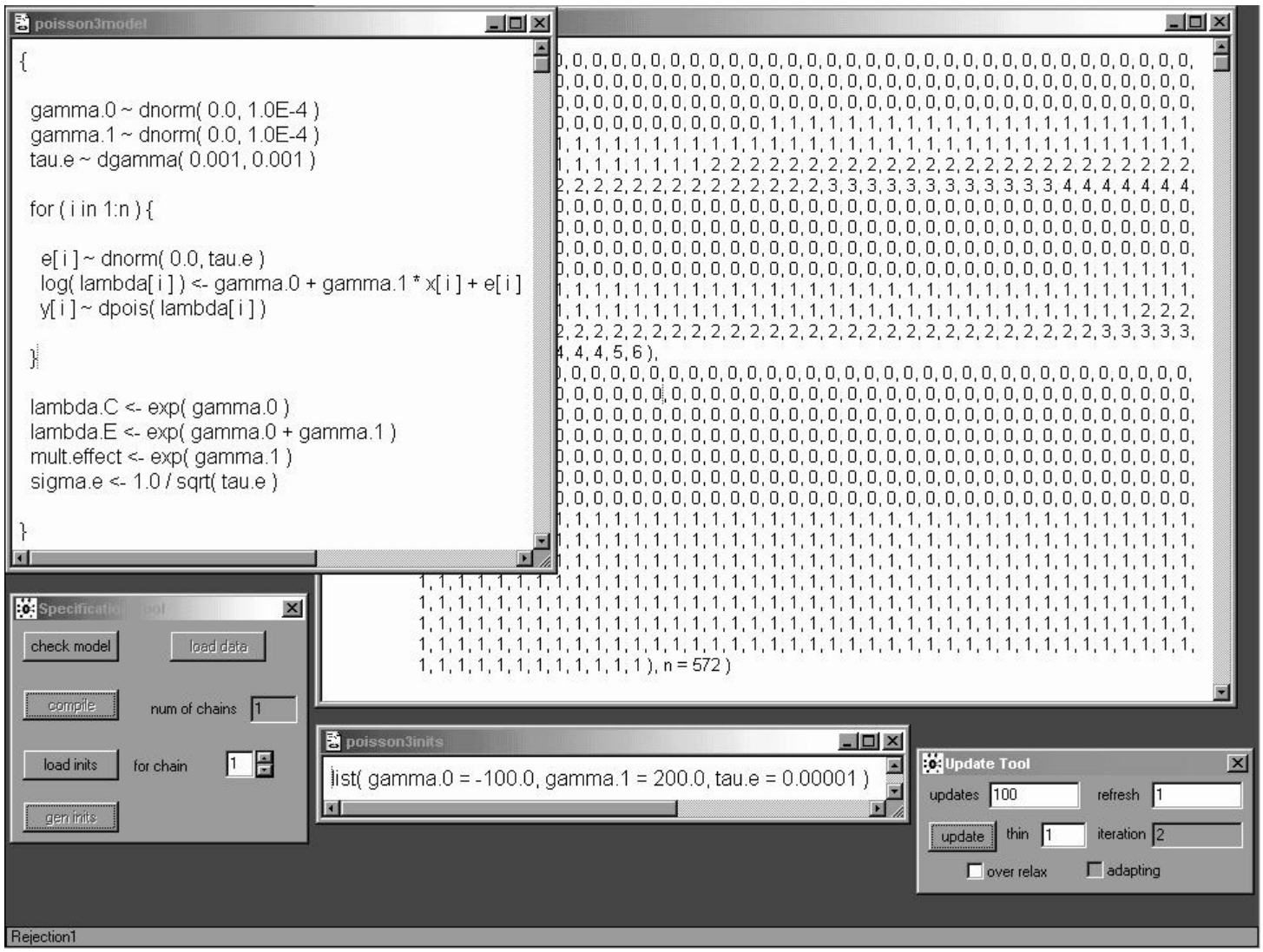
Warning `WinBUGS` can fail, particularly in random-effects models, when you give it initial values that are not very close to the final posterior means; an example in release 1.3 is the REPR model (16) on the IHGA data with the **"generic"** starting values $(\gamma_0, \gamma_1, \tau_e) = (0.0, 0.0, 1.0)$.

When this problem arises there are two ways out in `WinBUGS`: **trial and error**, or a calculation (see below).

NB `MLwiN` does not have this problem—it gets its starting values from **maximum likelihood** (the mode of the likelihood function is often a decent approximation to the mean or mode of the posterior).

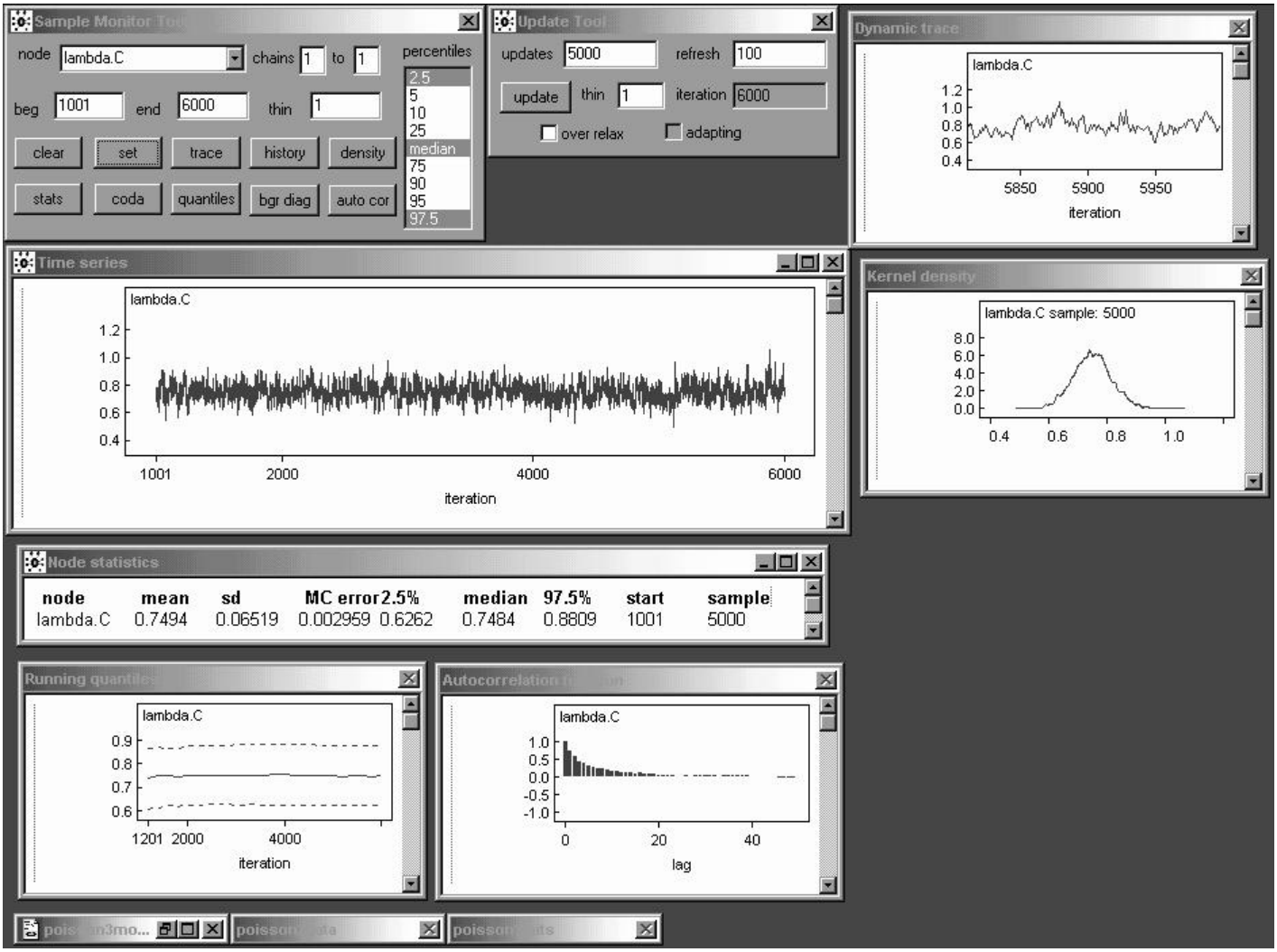
Technical note. To get a decent starting value for τ_e in model (16) you can calculate as follows: renaming the random effects η_i to avoid confusion with the number e, (1) $V(y_i) = V[E(y_i\,|\eta_i)] + E[V(y_i\,|\eta_i)]$, where (2) $(y_i\,|\eta_i) \sim$ Poisson $\left(e^{\gamma_0+\gamma_1 x_i+\eta_i}\right)$, so $E(y_i\,|\eta_i) = V(y_i\,|\eta_i) = e^{\gamma_0+\gamma_1 x_i+\eta_i}$. Then (3) $V[E(y_i\,|\eta_i)] = V\left(e^{\gamma_0+\gamma_1 x_i+\eta_i}\right) = e^{2(\gamma_0+\gamma_1 x_i)} V(e^{\eta_i})$ and $E[V(y_i\,|\eta_i)] = E\left(e^{\gamma_0+\gamma_1 x_i+\eta_i}\right) = e^{\gamma_0+\gamma_1 x_i} E(e^{\eta_i})$. Now (4) e^{η_i} is lognormal with mean 0 and variance σ_e^2 on the log scale, so $E(e^{\eta_i}) = e^{\frac{1}{2}\sigma_e^2}$ and $V(e^{\eta_i}) = e^{\sigma_e^2}\left(e^{\sigma_e^2} - 1\right)$, yielding finally $V(y_i) = e^{2(\gamma_0+\gamma_1 x_i)+\frac{1}{2}\sigma_e^2} + e^{\gamma_0+\gamma_1 x_i+\sigma_e^2}\left(e^{\sigma_e^2} - 1\right)$. (5) Plugging in $x_i = 0$ for the C group, whose sample variance is 1.54, and using the value $\gamma_0 = -0.29$ from runs with previous models, gives an equation for σ_e^2 that can be solved numerically, yielding $\sigma_e^2 \doteq 0.5$ and $\tau_e \doteq 2$.

WinBUGS Implementation (continued)



Interestingly, WinBUGS release 1.4 is able to **sample successfully** with the generic starting values $(\gamma_0, \gamma_1, \tau_e) = (0.0, 0.0, 1.0)$, although of course a **longer burn-in period** would be needed when they're used; you have to try **truly absurd** initial values to get it to fall over, and when it does so the **error message** ("Rejection1") in the lower left corner is **more discreet**.

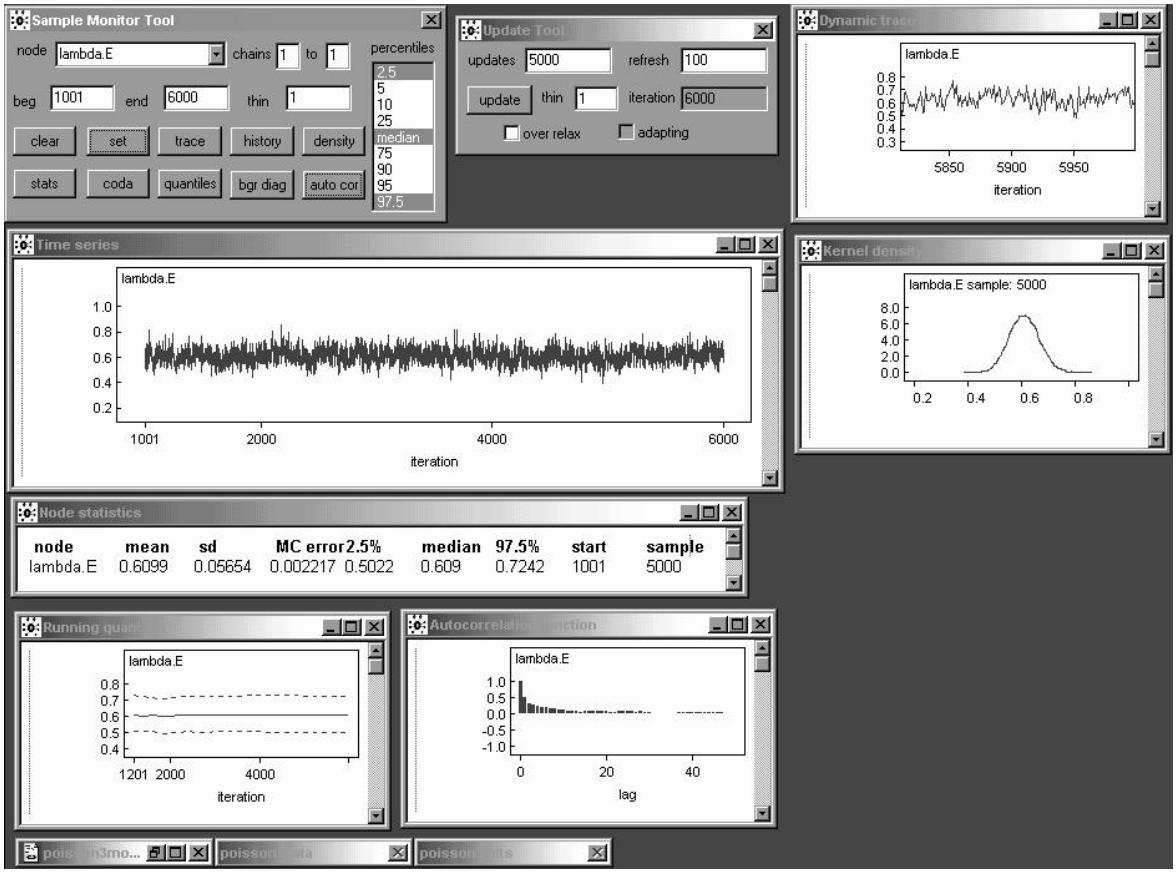
WinBUGS Implementation (continued)

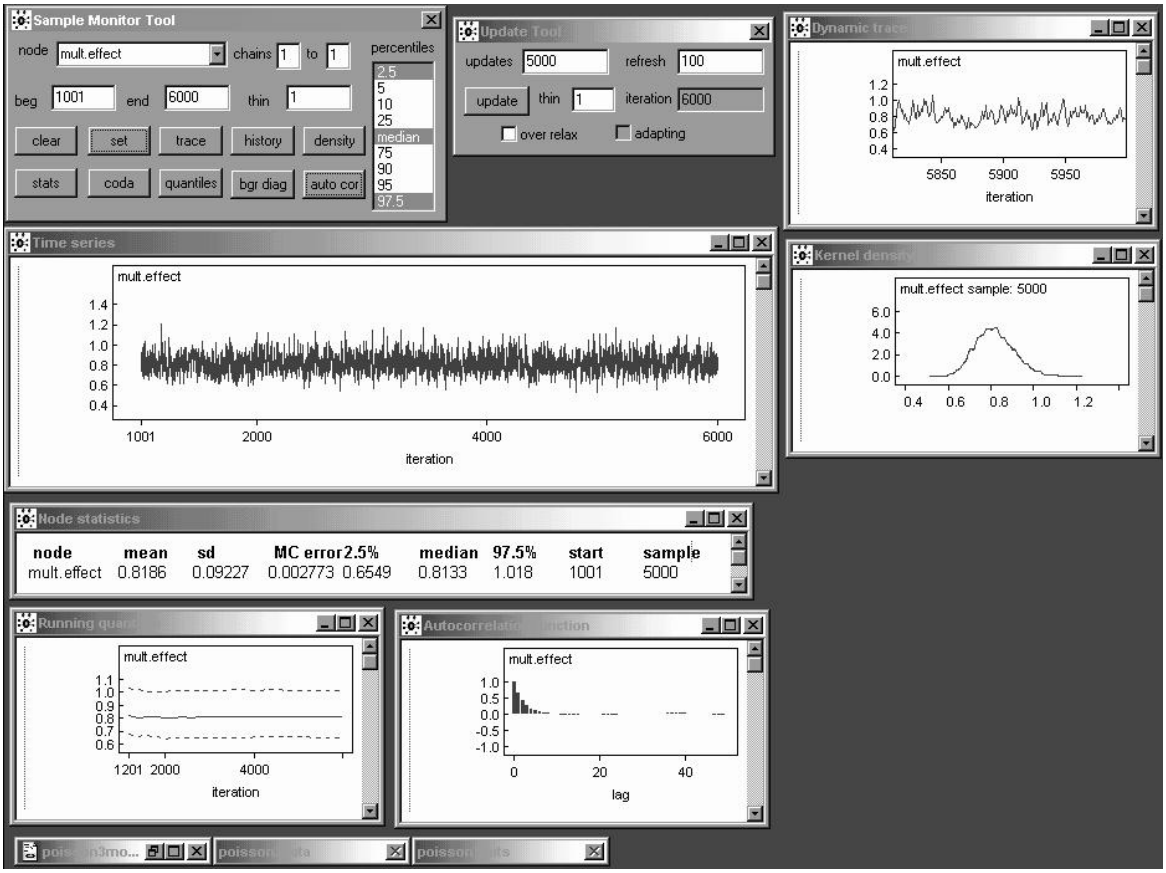


With a **better set of initial values**—$(\gamma_0, \gamma_1, \tau_e) = (-0.058, -0.21, 2.0)$, obtained from (a) the earlier Poisson models (in the case of the regression parameters γ_j) and (b) either a calculation like the one on the bottom of page 29 or trial and error—WinBUGS is able to make progress, although this model takes **a fairly long time to fit** in release 1.4: a burn-in of 1,000 takes 11 seconds at 1.0 PC GHz (the code runs about twice as fast in release 1.3 for some reason).

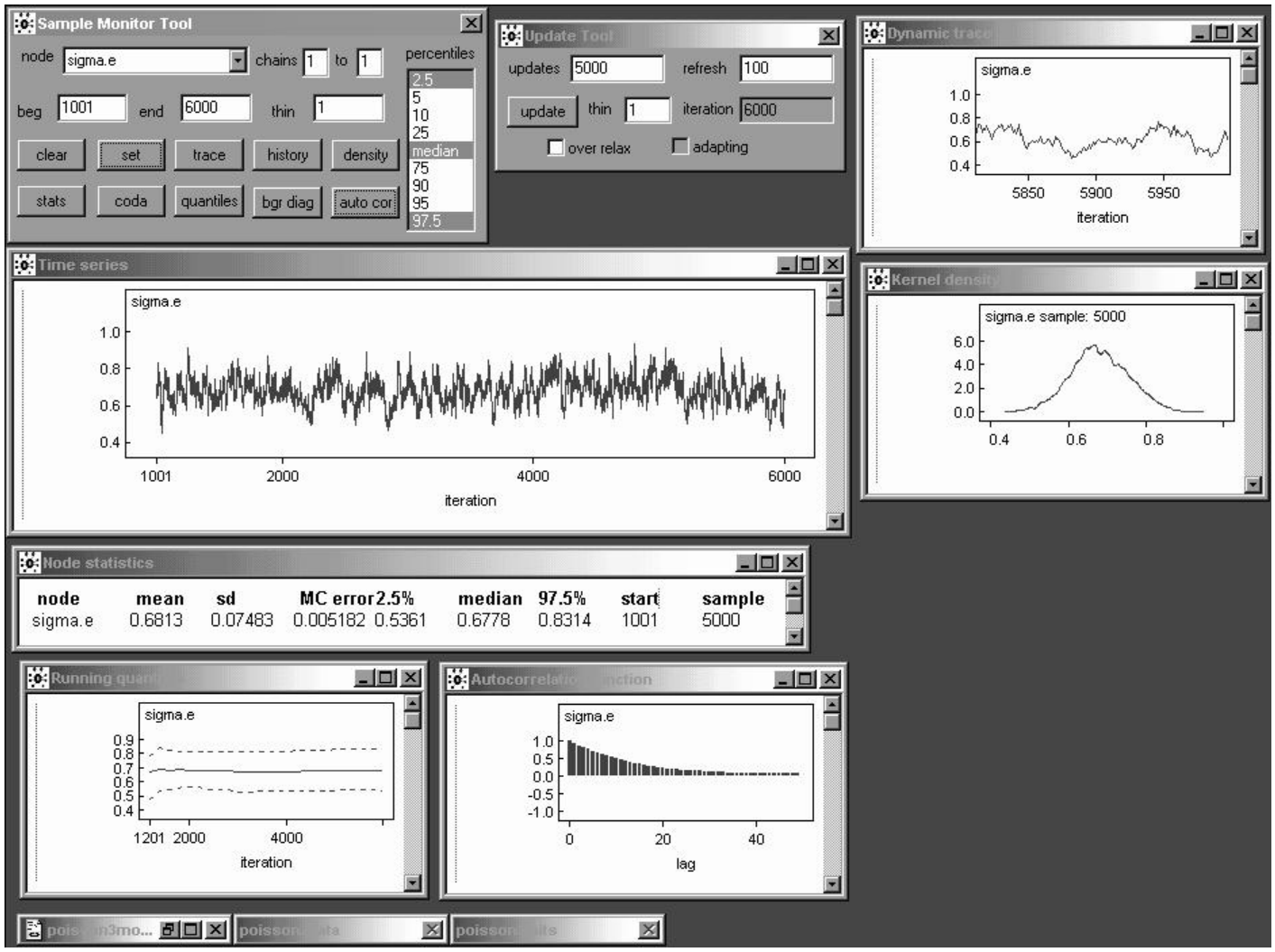
A monitoring run of **5,000** iterations reveals that the random effects make everything **mix more slowly**: λ_C (this page) and λ_E and the multiplicative effect (next page) all behave like AR_1 series with $\widehat{\rho}_1 \doteq 0.7$, 0.5, and 0.6, respectively.

WinBUGS Implementation (continued)





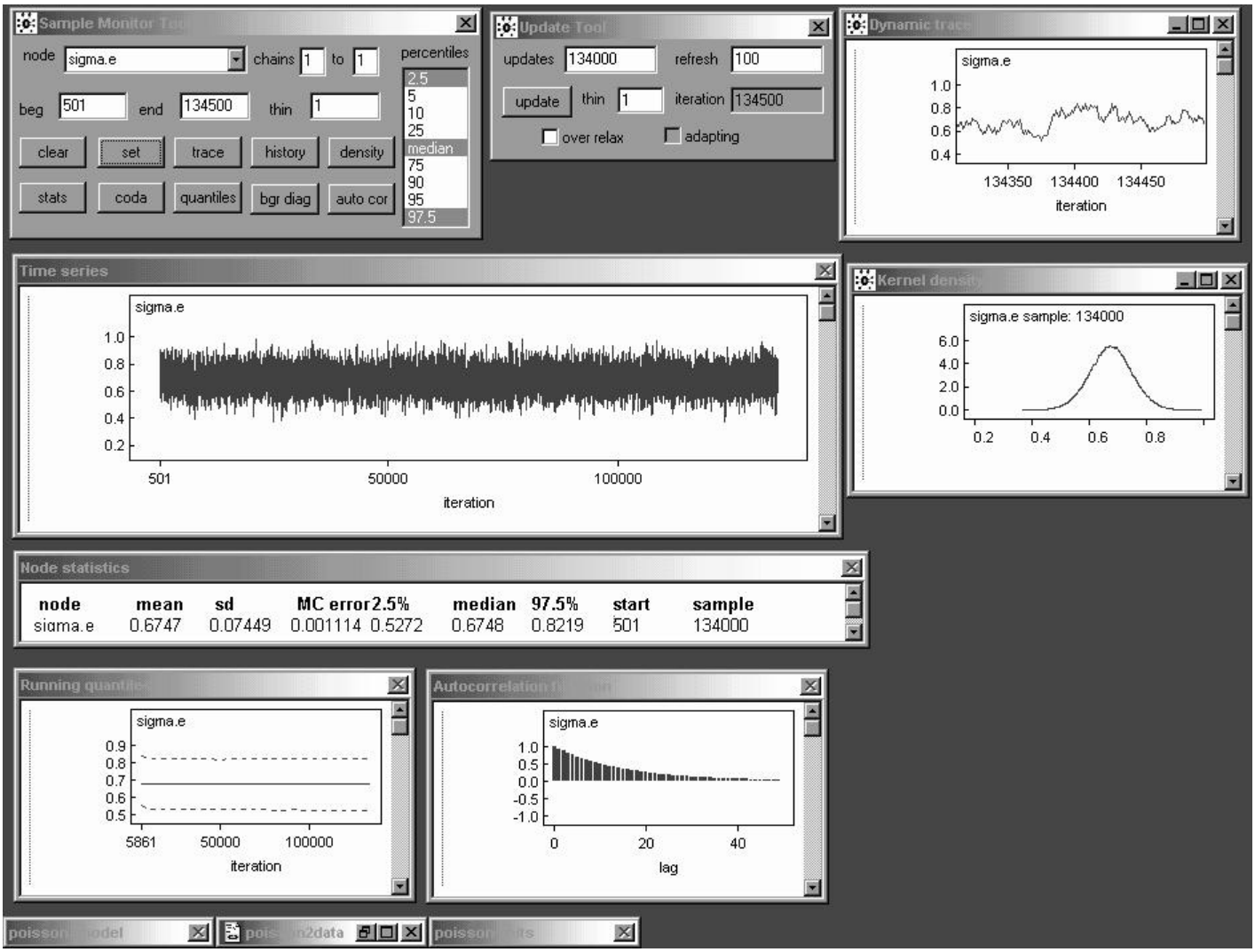
WinBUGS Implementation (continued)



Learning about σ_e in this model is **slow**: its autocorrelation function is that of an AR_1 with a **high value** of $\widehat{\rho}_1$ (equation (55) on page 76 of part 3 of the lecture notes gives $\widehat{\rho}_1 \doteq \mathbf{0.92}$).

The MCSE of the posterior mean for σ_e based on 5,000 draws is **0.005182**—to get this down to (say) **0.001** I need to increase the length of the monitoring run by a factor of $\left(\frac{0.005182}{0.001}\right)^2 \doteq \mathbf{26.9}$, meaning a total run of about $(26.9)(5,000) \doteq 134,000$ iterations (this takes about **half an hour** at 1 PC GHz).

WinBUGS Implementation (continued)



There is **clear evidence** that σ_e is **far from 0**—its posterior mean and SD are estimated as **0.675** (with an MCSE of about **0.001** after 134,000 iterations) and **0.074**, respectively—meaning that the **model expansion** from (11) to (16) was **amply justified**.

REPR Model Results

(Another way to achieve the goal of describing the extra-Poisson variability would be to fit different **negative binomial** distributions to the observed counts in the C and E groups—the negative binomial is a **gamma mixture of Poissons**, and the gamma and lognormal distributions often fit long-tailed data about equally well, so you would not be surprised to find that the two approaches give **similar results**.)

Table 4.5. Comparison of inferential conclusions about the multiplicative effect parameter e^{γ_1} from the fixed-effects and random-effects Poisson regression models.

Model	Posterior Mean	Posterior SD	Central 95% Interval
FEPR	0.816	0.0735	(0.683, 0.969)
REPR	0.830	0.0921	(0.665, 1.02)

Table 4.5 compares the REPR model inferential results with those from model (11), which could also be called a **fixed-effects Poisson regression** (FEPR) model.

The "error" SD σ_e has posterior mean **0.68**, give or take about 0.07 (on the $\log(\lambda)$ scale), corresponding to substantial extra-Poisson variability, which translates into **increased uncertainty** about the multiplicative effect parameter e^{γ_1}.

I'll argue later that the REPR model **fits the data well**, so the conclusion I'd publish from these data is that IHGA reduces the average number of hospitalizations per two years by about $100\,(1 - 0.083)\% = \boxed{\textbf{17\%}}$ give or take about 9% (ironically this conclusion is similar to that from the Gaussian model, but this is **coincidence**).

References

Bryk AS, Raudenbush SW (1992). *Hierarchical Linear Models: Applications and Data Analysis Methods*. London: Sage.

Carlin BP, Louis TA (1996). *Bayes and Empirical Bayes Methods for Data Analysis*. London: Chapman & Hall.

Draper D, Gaver D, Goel P, Greenhouse J, Hedges L, Morris C, Tucker J, Waternaux C (1993a). *Combining Information: Statistical Issues and Opportunities for Research*. Contemporary Statistics Series, No. 1. American Statistical Association, Alexandria VA.

Draper D, Hodges JS, Mallows CL, Pregibon D (1993b). Exchangeability and data analysis (with discussion). *Journal of the Royal Statistical Society, Series A*, **156**, 9–37.

Draper D (1995). Inference and hierarchical modeling in the social sciences (with discussion). *Journal of Educational and Behavioral Statistics*, **20**, 115–147, 233–239.

Hedges LV, Olkin I (1985). *Statistical Methods for Meta-Analysis*. New York: Academic Press.

Morris CN (1983). Parametric empirical Bayes inference: Theory and applications. *Journal of the American Statistical Association*, **78**, 47–59.

Morris CN (1988). Determining the accuracy of Bayesian empirical Bayes estimators in the familiar exponential families. In *Proceedings of the Fourth Purdue Symposium on Statistical Decision Theory and Related Topics IV, part 1.*, SS Gupta, JO Berger, eds. New York: Springer-Verlag, 251–263.

Raudenbush SW (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 19 experiments. *Journal of Educational Psychology*, **76**, 85–97.