# Bayesian Modeling, Inference, Prediction and Decision-Making

## 5: Bayesian Model Specification (Section 2)

**David Draper**

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

`draper@ams.ucsc.edu`

`www.ams.ucsc.edu/∼draper`

*eBay/Google*

10 Fridays, 11 Jan–22 Mar 2013 (except 25 Jan)

**Short course web page:**
`www.ams.ucsc.edu/∼draper/eBay-Google-2013.html`

(This section of the notes is based in part on unpublished joint work with a recent Ph.D. student of mine, Milovan Krnjajić.)

# What is a Bayesian Model?

$\boxed{\textbf{Definition:}}$ A **Bayesian model** is a **mathematical framework** (embodying **assumptions** and **judgments**) for **quantifying uncertainty about unknown quantities** by relating them to **known quantities**.

Desirable for the **assumptions** and **judgments** in the model to arise as directly as possible from **contextual information** in the problem under study.

The most satisfying approach to **achieving this goal** appears to be that of de Finetti (1990): a **Bayesian model** is a **joint predictive distribution**

$$p(y) = p(y_1, \ldots, y_n) \tag{1}$$

for as-yet-unobserved **observables** $y = (y_1, \ldots, y_n)$.

$\boxed{\textbf{Example 1:}}$ Data = **health outcomes** for all patients at one hospital with heart attack admission diagnosis.

Simplest possible: $y_i = 1$ if patient $i$ **dies within 30 days of admission**, 0 otherwise.

de Finetti (1930): **in absence of any other information**, my predictive uncertainty about $y_i$ is **exchangeable**.

**Representation theorem** for binary data: if I'm willing to regard $(y_1, \ldots, y_n)$ as part of an **infinitely exchangeable sequence** (meaning that I judge all **finite subsets** exchangeable; this is like **thinking** of the $y_i$ as having been **randomly sampled** from the **population** $(y_1, y_2, \ldots)$), then to be **coherent** my joint predictive distribution $p(y_1, \ldots, y_n)$ must have the simple **hierarchical** form

$$\begin{aligned} \theta &\sim p(\theta) \\ (y_i|\theta) &\overset{\text{IID}}{\sim} \text{Bernoulli}(\theta), \end{aligned} \tag{2}$$

where $\theta = P(y_i = 1)$ = **limiting value of mean of** $y_i$ in infinite sequence.

## Model = Prior (Sometimes)

**Mathematically** $p(\theta)$ is **mixing distribution** in

$$p(y_1, \ldots, y_n) = \int_0^1 \theta^s (1-\theta)^{n-s} \, p(\theta) \, d\theta, \qquad (3)$$

where $s = \sum_{i=1}^n y_i$; **statistically**, $p(\theta)$ provides opportunity to quantify **prior information** about $\theta$ and combine with information in $y$.

Thus, in simplest situation, **Bayesian model specification** = choice of **scientifically appropriate prior distribution** $p(\theta)$.

**Example 2 (elaborating Example 1):** Now I want to predict real-valued **sickness-at-admission score** instead of mortality (still no **covariates**).

Uncertainty about $y_i$ still **exchangeable**; de Finetti's (1937) **representation theorem** for real-valued data: if $(y_1, \ldots, y_n)$ part of **infinitely exchangeable sequence**, all **coherent** joint predictive distributions $p(y_1, \ldots, y_n)$ must have hierarchical form

$$\begin{array}{rcl} F & \sim & p(F) \\ (y_i | F) & \stackrel{\text{IID}}{\sim} & F, \end{array} \qquad (4)$$

where $F$ = **limiting empirical cumulative distribution function** (CDF) of infinite sequence $(y_1, y_2, \ldots)$.

## Bayesian Nonparametrics

Thus here Bayesian model specification = choosing **scientifically appropriate mixing (prior) distribution** $p(F)$ for $F$.

However, $F$ is **infinite-dimensional parameter**; putting probability distribution on $\mathcal{D} = \{$all possible CDFs$\}$ is harder.

Specifying distributions on **function spaces** is task of Bayesian **nonparametric** (BNP) modeling (e.g., Dey et al. 1998).

$\boxed{\textbf{Example 3 (elaborating Example 2):}}$ In practice, in addition to **outcomes** $y_i$, **covariates** $x_{ij}$ will typically be available.

For instance (Hendriksen et al. 1984), 572 elderly people **randomized**, 287 to **control** ($C$) group (standard care) and 285 to **treatment** ($T$) group (standard care plus **in-home geriatric assessment** (IHGA): **preventive medicine** in which each person's medical/social needs assessed, acted upon individually).

One important **outcome** was **number of hospitalizations** (in two years).

$y_i^T$, $y_j^C$ = numbers of hospitalizations for **treatment** person $i$, **control** person $j$, respectively.

Suppose **treatment/control** (T/C) status is **only available covariate**.

## Conditional Exchangeability

**Unconditional** judgment of exchangeability across all 572 outcomes **no longer automatically scientifically appropriate**.

Instead **design of experiment** compels (at least initially) judgment of **conditional exchangeability given T/C status** (e.g., de Finetti 1938, Draper et al. 1993), as in

$$\begin{array}{c} (F_T, F_C) \sim p(F_T, F_C) \\ (y_i^T | F_T, F_C) \overset{\text{IID}}{\sim} F_T \Big| (y_j^C | F_T, F_C) \overset{\text{IID}}{\sim} F_C \end{array} \qquad (5)$$

This framework, in which (a) **covariates** specify **conditional exchangeability judgments**, (b) de Finetti's **representation theorem** reduces model specification task to placing appropriate prior distributions on CDFs, covers much of field of **statistical inference/prediction**.

Note that even in this **rather general nonparametric framework** it will be necessary to have a **good tool** for **discriminating between the quality of two models** (here: **unconditional** exchangeability ($F_T = F_C$; $T$ has **same effect** as $C$) versus **conditional** exchangeability ($F_T \neq F_C$; $T$ and $C$ effects **differ**)).

## Data-Analytic Model Specification

**Basic problem** of Bayesian **model choice**: Given future observables $y = (y_1, \ldots, y_n)$, I'm **uncertain** about $y$ (**first-order**), but I'm also **uncertain about how to specify my uncertainty** about $y$ (**second-order**).

Standard (**data-analytic**) approach to model specification involves initial choice, for **structure** of model, of **standard parametric family**, followed by **modification** of initial choice—once data begin to arrive—if data suggest **deficiencies** in original specification.

This approach (e.g., Draper 1995) is **incoherent** (unless I pay an **appropriate price** for **shopping around** for the model): it uses data both to specify **prior distribution on structure space** and to **update** using **data-determined prior** (result will typically be **uncalibrated** (too narrow) predictive distributions for future data).

Dilemma is example of **Cromwell's Rule** (if $p(\theta) = 0$ then $p(\theta|y) = 0$ for all $y$): initial model choice placed **0 prior probability** on **large regions of model space**; formally all such regions **must also have 0 posterior probability** even if data indicate **different prior on model space** would have been better.

## Two Possible Solutions

• If use prior on $F$ that places **non-zero probability on all Kullback-Leibler neighborhoods of all densities** (Walker et al. 2003; e.g., Pólya trees, Dirichlet process mixture priors, when chosen well), then BNP **directly avoids** Cromwell's Rule dilemma, at least for large $n$: as $n \rightarrow \infty$ posterior on $F$ will **shrug off** any incorrect details of prior specification, will **fully adapt** to actual data-generating $F$ ($\boxed{\textbf{NB}}$ this assumes correct exchangeability judgments).

• **Three-way cross-validation** (3CV; Draper and Krnjajić 2005): taking usual cross-validation idea one step further,

**(1) Partition** data at random into *three* (non-overlapping and exhaustive) subsets $S_i$.

**(2)** Fit tentative {likelihood + prior} to $S_1$. **Expand** initial model in all feasible ways suggested by data exploration using $S_1$. **Iterate** until you're happy.

**(3)** Use final model (fit to $S_1$) from (2) to create predictive distributions for all data points in $S_2$. Compare actual outcomes with these distributions, checking for **predictive calibration**. Go back to (2), change likelihood as necessary, **retune prior** as necessary, to get good calibration. **Iterate** until you're happy.

**(4)** Announce **final model** (fit to $S_1 \cup S_2$) from (3), and report **predictive calibration** of this model on data points in $S_3$ as indication of how well it would perform with new data.

With **large** $n$ probably only need to do this **once**; with **small** and **moderate** $n$ probably best to **repeat** (1–4) several times and **combine** results in some appropriate way (e.g., **model averaging**).

## Model Selection as a Decision Problem

Given method like 3CV which permits **hunting around in model space** without forfeiting calibration, two kinds of model specification questions (in both **parametric** and **nonparametric** Bayesian modeling) arise:

(1) Is $M_1$ **better than** $M_2$? (this tells me **when it's OK to discard a model in my search**)

(2) Is $M_1$ **good enough**? (this tells me **when it's OK to stop searching**)

It would seem self-evident that **to specify a model you have to say to what purpose the model will be put**, for how else can you answer these two questions?

Specifying this purpose demands **decision-theoretic basis for model choice** (e.g., Draper 1996; Key et al. 1998).

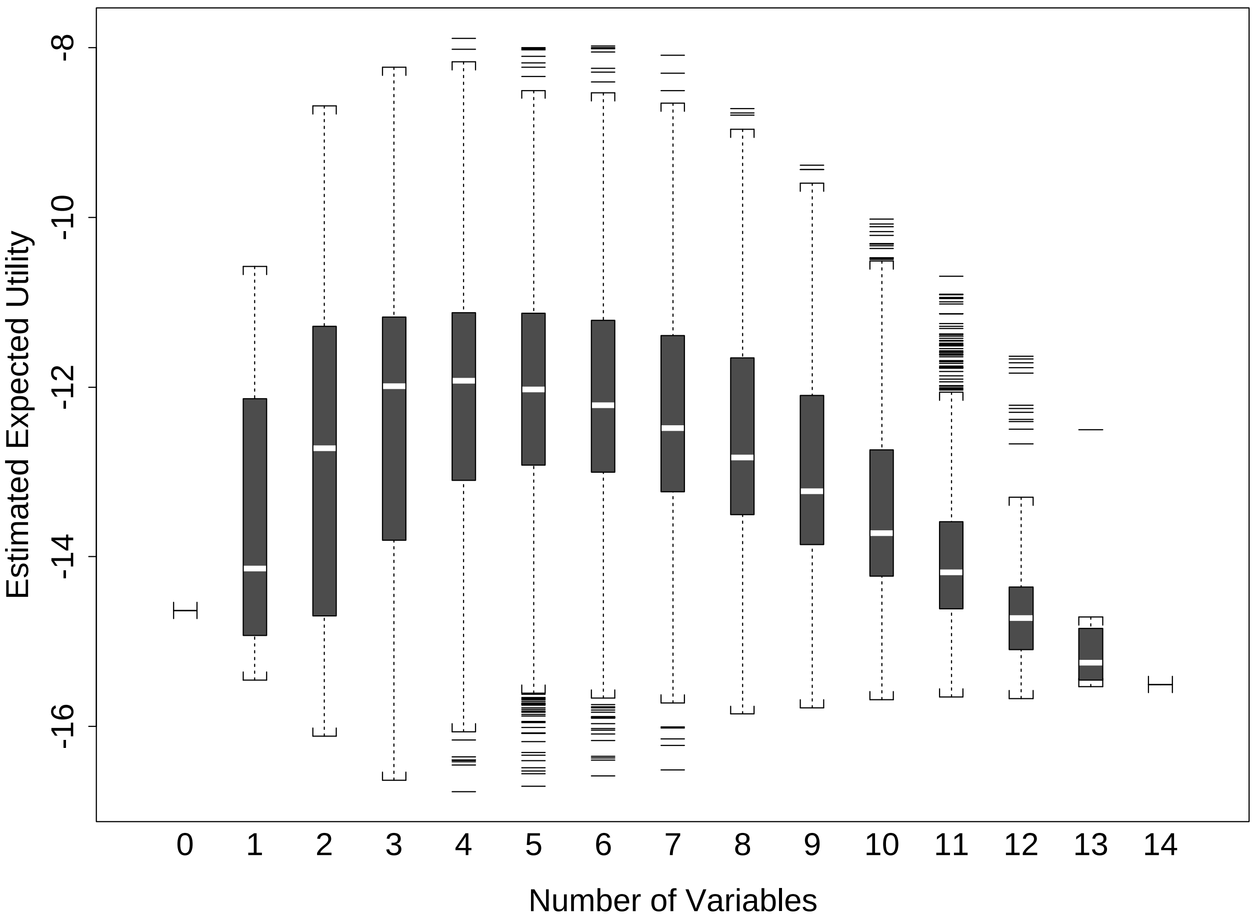
To take **two examples**,

$\boxed{\textbf{(Case 1)}}$ If you're going to choose which of several ways to behave in future, then model has to be **good enough to reliably aid in choosing best behavior** (see, e.g., Draper and Fouskakis example below); or

$\boxed{\textbf{(Case 2)}}$ If you wish to make scientific summary of what's known, then—remembering that hallmark of good science is good prediction—the model has to be **good enough to make sufficiently accurate predictions of observable outcomes** (in which dimensions along which accuracy is to be monitored are driven by what's **scientifically relevant**; see, e.g., log score results below).

## Utility-Based Variable Selection

**Example 4 (Case 1):** Draper and Fouskakis (2000, 2004) (also see Fouskakis and Draper 2002) give one example of decision-theoretic model choice in action, demonstrating that **variable selection in regression models** should often be governed by principle that final model should only contain variables that predict well enough **given how much they cost to collect** (see the figure below, which compares $2^{14} =$ **16,384** models).



*Estimated expected utility as function of number of predictor variables, in problem involving construction of cost-effective scale to measure sickness at hospital admission of elderly pneumonia patients. Best models only have 4–6 sickness indicators out of 14 possible predictors.*

## Choosing Utility Function

Any reasonable utility function in Example 4 will have two components, one quantifying **data collection costs** associated with construction of given sickness scale, other rewarding and penalizing scale's **predictive successes, failures**.

$\boxed{\textbf{(Case 2)}}$ Sometimes the main goal instead is **summary of scientific knowledge**, which suggests (as noted above) a **utility function** that rewards **predictive accuracy**.

How can such a **utility function** be specified in a **reasonably general way** to answer **model specification question (1)** above? (Is $M_1$ **better than** $M_2$?)

Need **scoring rule** that measures **discrepancy** between observation $y^*$ and predictive distribution $p(\cdot|y, M_i)$ for $y^*$ under model $M_i$ given data $y$.

As noted (e.g.) by Good (1950) and O'Hagan and Forster (2004), **the optimal (impartial, symmetric, proper)** scoring rules are linear functions of $\boxed{\log p(y^*|y)}$.

On **calibration** grounds it would $\boxed{\textbf{seem}}$ to be a mistake to **use data twice** in measuring this sort of thing (once to make predictions, again with same data to see how good they are; but ...).

**Out-of-sample predictive validation** (e.g., Geisser and Eddy 1979, Gelfand et al. 1992) solves this problem: e.g., successively remove each observation $y_j$ one at a time, construct predictive distribution for $y_j$ based on $y_{-j}$ (data vector with $y_j$ removed), see where $y_j$ falls in this distribution.

## Log Score as Utility

This motivates **cross-validated** version of **log scoring rule** (e.g., Gelfand and Dey 1994; Bernardo and Smith 1994): with $n$ data values $y_j$, when choosing among $k$ models $M_i, i \in I$, find that model $M_i$ which maximizes

$$LS_{CV}(M_i|y) = \frac{1}{n} \sum_{j=1}^{n} \log p(y_j|M_i, y_{-j}). \qquad (6)$$

It has been argued that this can be given direct **decision-theoretic justification**: with utility function for model $i$

$$U(M_i|y) = \log p(y^*|M_i, y), \qquad (7)$$

where $y^*$ is **future data value**, expectation in MEU is over **uncertainty about** $y^*$; Gelfand et al. (1992) and Bernardo and Smith (1994) claim that this expectation can be accurately **estimated** (assuming exchangeability) by (6) (I'll revisit this claim below).

With approximately **Gaussian** predictive distributions it can also be revealing to compute **predictive** $z$**–scores**, for observation $j$ under model $i$:

$$z_{ij} = \frac{y_j - E(y_j|M_i, y_{-j})}{\sqrt{V(y_j|M_i, y_{-j})}}. \qquad (8)$$

For **good predictive calibration** of $M_i$, $\{z_{ij}, j = 1, \ldots, n\}$ should have **mean 0**, **standard deviation** (SD) **1**; often find instead that SD is **larger than 1** (predictive uncertainty bands **not wide enough**).

## Approximating Log Score Utility

With **large data sets**, in situations in which **predictive distribution** has to be **estimated by MCMC**, direct calculation of $LS_{CV}$ is **computationally expensive**; need **fast approximation** to it.

To see how this might be obtained, examine log score in **simplest possible model** $M_0$: for $i = 1, \ldots, n$,

$$\begin{aligned} \mu & \sim N\left(\mu_0, \sigma_\mu^2\right) \\ (Y_i|\mu) & \overset{\text{IID}}{\sim} N(\mu, \sigma^2) \end{aligned} \tag{9}$$

with $\sigma$ known, take **highly diffuse prior** on $\mu$ so that **posterior** for $\mu$ is approximately

$$(\mu|y) = (\mu|\bar{y}) \mathrel{\dot\sim} N\left(\bar{y}, \frac{\sigma^2}{n}\right), \tag{10}$$

where $y = (y_1, \ldots, y_n)$.

Then **predictive distribution** for next observation is approximately

$$(y_{n+1}|y) = (y_{n+1}|\bar{y}) \mathrel{\dot\sim} N\left[\bar{y}, \sigma^2\left(1 + \frac{1}{n}\right)\right], \tag{11}$$

and $LS_{CV}$, ignoring linear scaling constants, is

$$LS_{CV}(M_0|y) = \sum_{j=1}^{n} \ln p(y_j|y_{-j}), \tag{12}$$

where as before $y_{-j}$ is $y$ with observation $j$ **set aside**.

But by **same reasoning**

$$p(y_j|y_{-j}) \doteq N\left(\bar{y}_{-j}, \sigma_n^2\right), \tag{13}$$

where $\bar{y}_{-j}$ is **sample mean** with observation $j$ **omitted**, $\sigma_n^2 = \sigma^2\left(1 + \frac{1}{n-1}\right)$, so that

## $LS_{CV}$ Approximation (continued)

$$\begin{aligned} \ln p(y_j | y_{-j}) &\doteq c - \frac{1}{2\sigma_n^2}(y_j - \bar{y}_{-j})^2 \quad \text{and} \\ LS_{CV}(M_0|y) &\doteq c_1 - c_2 \sum_{j=1}^{n} (y_j - \bar{y}_{-j})^2 \qquad (14) \end{aligned}$$

for some **constants** $c_1$ and $c_2$ with $c_2 > 0$.

Now it's **interesting fact** (related to behavior of **jackknife**), which you can prove by **induction**, that

$$\sum_{j=1}^{n} (y_j - \bar{y}_{-j})^2 = c \sum_{j=1}^{n} (y_j - \bar{y})^2 \qquad (15)$$

for some $c > 0$, so finally for $c_2 > 0$ the **result** is that

$$LS_{CV}(M_0|y) \doteq c_1 - c_2 \sum_{j=1}^{n} (y_j - \bar{y})^2, \qquad (16)$$

i.e., in this model log score is **almost perfectly negatively correlated with sample variance**.

But in this model the **deviance** (minus twice the log likelihood) is

$$\begin{aligned} D(\mu) &= -2 \ln l(\mu|y) = c_0 - 2 \ln p(y|\mu) \\ &= c_0 + c_3 \sum_{j=1}^{n} (y_j - \mu)^2 \qquad (17) \end{aligned}$$

for some $c_3 > 0$, encouraging suspicion that **log score should be strongly related to deviance**.

## Deviance Information Criterion ($DIC$)

Given parametric model $p(y|\theta)$, Spiegelhalter et al. (2002) define **deviance information criterion** ($DIC$) (by analogy with other information criteria) to be estimate $D(\bar{\theta})$ of model (lack of) **fit** (as measured by deviance) plus **penalty for complexity** equal to twice **effective number of parameters** $p_D$ of model:

$$DIC(M|y) = D(\bar{\theta}) + 2\,\hat{p}_D, \tag{18}$$

where $\bar{\theta}$ is posterior mean of $\theta$; they suggest that models with **low** $DIC$ value are to be **preferred** over those with higher value.

When $p_D$ is **difficult to read directly from model** (e.g., in **complex hierarchical models**, especially those with **random effects**), they motivate the following **estimate**, which is easy to compute from standard MCMC output:

$$\hat{p}_D = \overline{D(\theta)} - D(\bar{\theta}), \tag{19}$$

i.e., difference between **posterior mean of deviance** and **deviance evaluated at posterior mean** of parameters (`WinBUGS` release 1.4 will **estimate** these quantities).

In **model** $M_0$, $p_D$ is of course 1, and $\bar{\theta} = \bar{y}$, so

$$DIC(M_0|y) = c_0 + c_3 \sum_{j=1}^{n} (y_j - \bar{y})^2 + 2 \tag{20}$$

and conclusion is that

$$-DIC(M_0|y) \doteq c_1 + c_2 LS_{CV}(M_0|y) \tag{21}$$

for $c_2 > 0$, i.e., (if this generalizes) **choosing model by maximizing $LS_{CV}$ and by minimizing $DIC$ are approximately equivalent behaviors**.

(This connection was **hinted at** in discussion of Spiegelhalter et al. 2002 but never really made **explicit**.)

## $LS_{CV} \leftrightarrow DIC$?

Milovan and I are now (work in progress) **exploring the scope** of (21); in several **simple models** $M$ so far we find for $c_2 > 0$ that

$$-DIC(M|y) \doteq c_1 + c_2 LS_{CV}(M|y), \qquad (22)$$

i.e., across repeated data sets generated from given model, even with small $n$ $DIC$ and $LS_{CV}$ can be **fairly strongly negatively correlated**.

Above argument **generalizes** to **any situation** in which **predictive distribution** is **approximately Gaussian** (e.g., **Poisson**($\lambda$) likelihood with **large** $\lambda$, **Beta**($\alpha, \beta$) likelihood with **large** ($\alpha + \beta$), etc.).

$\boxed{\textbf{Example 3 continued.}}$ With **one-sample count data** (like number of hospitalizations in the $T$ and $C$ portions of IHGA data), people often choose between **fixed-** and **random-effects Poisson model formulations**: for $i = 1, \ldots, n$, and, e.g., with **diffuse priors**,

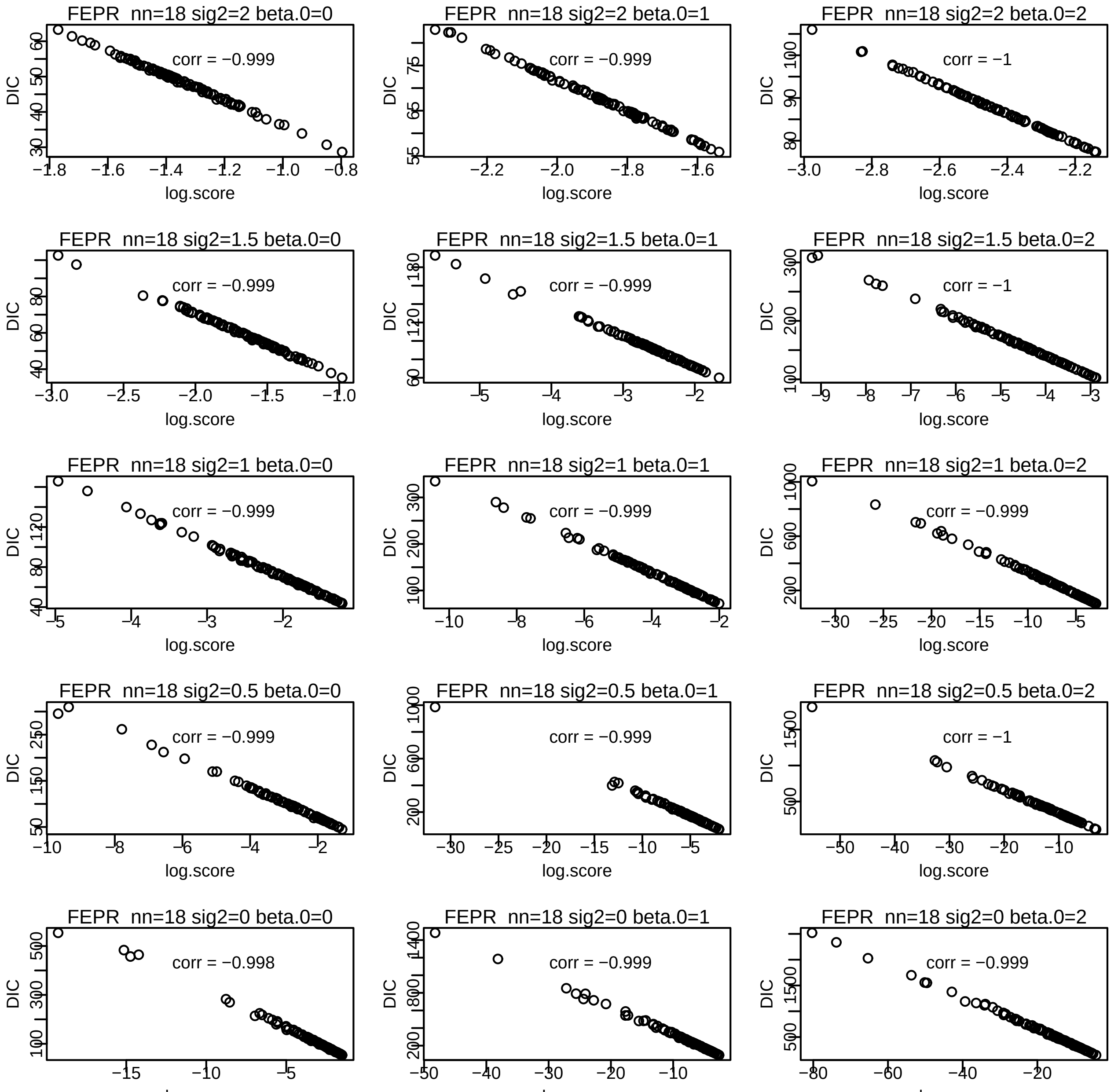
$$M_1\text{:} \left\{ \begin{array}{ccc} \lambda & \sim & p(\lambda) \\ (y_i|\lambda) & \stackrel{\text{IID}}{\sim} & \text{Poisson}(\lambda) \end{array} \right\} \quad \text{versus} \qquad (23)$$

$$M_2\text{:} \left\{ \begin{array}{ccc} (\beta_0, \sigma^2) & \sim & p(\beta_0, \sigma^2) \\ (y_i|\lambda_i) & \stackrel{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \beta_0 + e_i \\ e_i & \stackrel{\text{IID}}{\sim} & N(0, \sigma^2) \end{array} \right\} \qquad (24)$$

$M_1$ is **special case** of $M_2$ with $\left(\sigma^2 = 0, \lambda = e^{\beta_0}\right)$; likelihood in $M_2$ is **Lognormal mixture of Poissons** (often similar to fitting **negative binomial** distribution, which is **Gamma mixture of Poissons**).

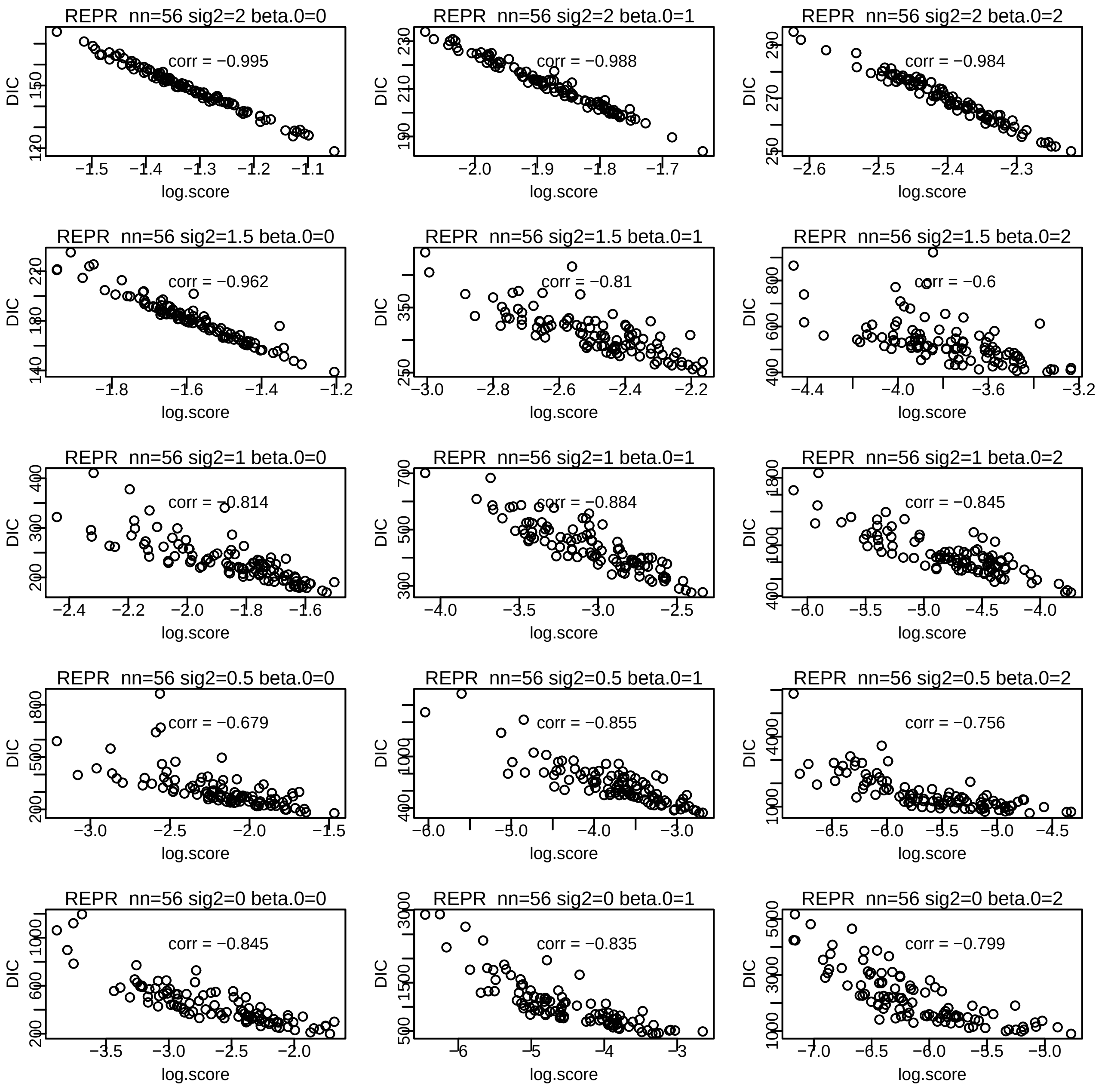
## $LS_{CV} \leftrightarrow DIC$? (continued)

We conducted **partial-factorial simulation study** with factors $\{n = 18, 32, 42, 56, 100\}$, $\{\beta_0 = 0.0, 1.0, 2.0\}$, $\{\sigma^2 = 0.0, 0.5, 1.0, 1.5, 2.0\}$ in which (**data-generating mechanism**, **assumed model**) = $\{(M_1, M_1), (M_1, M_2), (M_2, M_1), (M_2, M_2)\}$; in each cell of this grid we used 100 **simulation replications**.



When **assumed model** is $M_1$ (**fixed-effects** Poisson), $LS_{CV}$ and $DIC$ are **almost perfectly negatively correlated** (we have **mathematical explanation** of this).

## $LS_{CV} \leftrightarrow DIC$? (continued)



When **assumed model** is $M_2$ (**random-effects** Poisson), $LS_{CV}$ and $DIC$ are **less strongly negatively correlated** ($DIC$ can **misbehave** with **mixture models**; see below), but **correlation increases with** $n$.

# Example 3

As example of **correspondence between** $LS_{CV}$ **and** $DIC$ in real problem, IHGA data were as follows:

*Distribution of number of hospitalizations in IHGA study over 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 |
| Treatment | 147 | 83 | 37 | 13 | 3 | 1 | 1 | 0 | 285 | 0.768 | 1.01 |

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

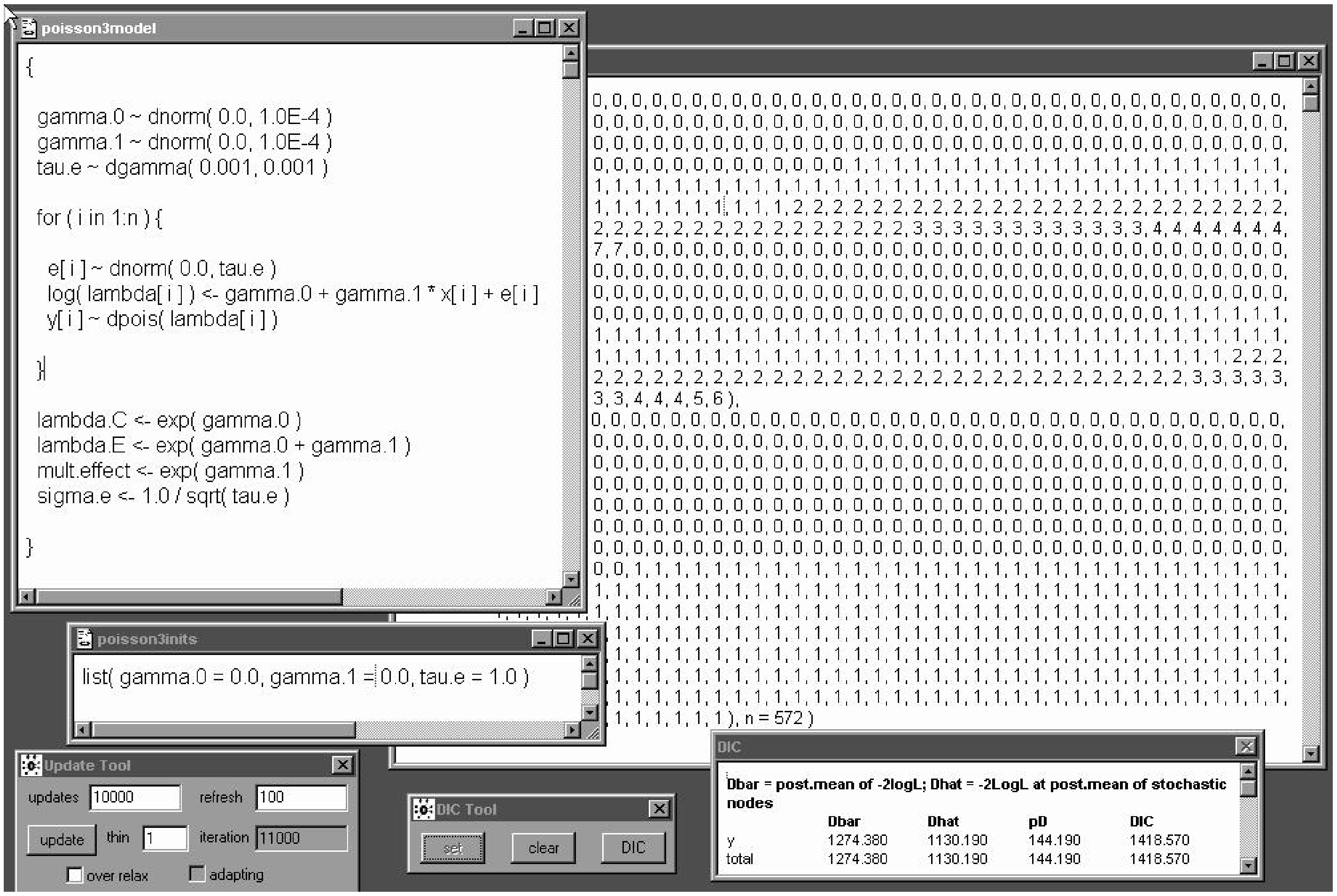
Four **possible models** for these data (not all of them good):

- **Two-independent-sample Gaussian** (diffuse priors);
- **One-sample Poisson** (diffuse prior), pretending treatment and control $\lambda$s are equal;
- **Two-independent-sample Poisson** (diffuse priors), which is equivalent to **fixed-effects Poisson regression** (FEPR); and
- **Random-effects Poisson regression** (REPR), because $C$ and $T$ **variance-to-mean ratios** (VTMRs) are 1.63 and 1.32, respectively:

$$\begin{array}{rcl} (y_i \,|\, \lambda_i) & \overset{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \beta_0 + \beta_1 x_i + e_i \\ e_i & \overset{\text{IID}}{\sim} & N\left(0, \sigma_e^2\right) \\ \left(\beta_0, \beta_1, \sigma_e^2\right) & \sim & \text{diffuse} \, , \end{array} \qquad (25)$$

where $x_i = 1$ is a **binary indicator** for $T/C$ status.

# DIC Example



poisson3model

```
{

  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 )

}
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
7, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3,
3, 3, 4, 4, 4, 5, 6 ),
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1 ), n = 572 )
```

poisson3inits

```
list( gamma.0 = 0.0, gamma.1 = 0.0, tau.e = 1.0 )
```

Update Tool

```
updates 10000    refresh 100
update   thin 1   iteration 11000
☐ over relax   ☐ adapting
```

DIC

```
Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic
nodes
          Dbar        Dhat        pD        DIC
y         1274.380    1130.190    144.190   1418.570
total     1274.380    1130.190    144.190   1418.570
```

To use the **DIC feature** in `WinBUGS` to produce the screen shot above, I **fit** the REPR model as usual, did a **burn-in** of 1,000, **selected** `DIC` as a pull-down option from the `Inference` menu, **clicked** the `set` button in the `DIC Tool` window that popped up, **changed** the 1,000 to 10,000 in the `updates` window of the `Update Tool`, **clicked** `update`, and then **clicked** `DIC` in the `DIC Tool` when the monitoring run of 10,000 was finished—the `DIC` **results window** appears, with the `Dbar` ($\overline{D(\theta)}$), `Dhat` ($D(\bar{\theta})$), `pD` ($\hat{p}_D$), and `DIC` ($DIC(y)$) values.

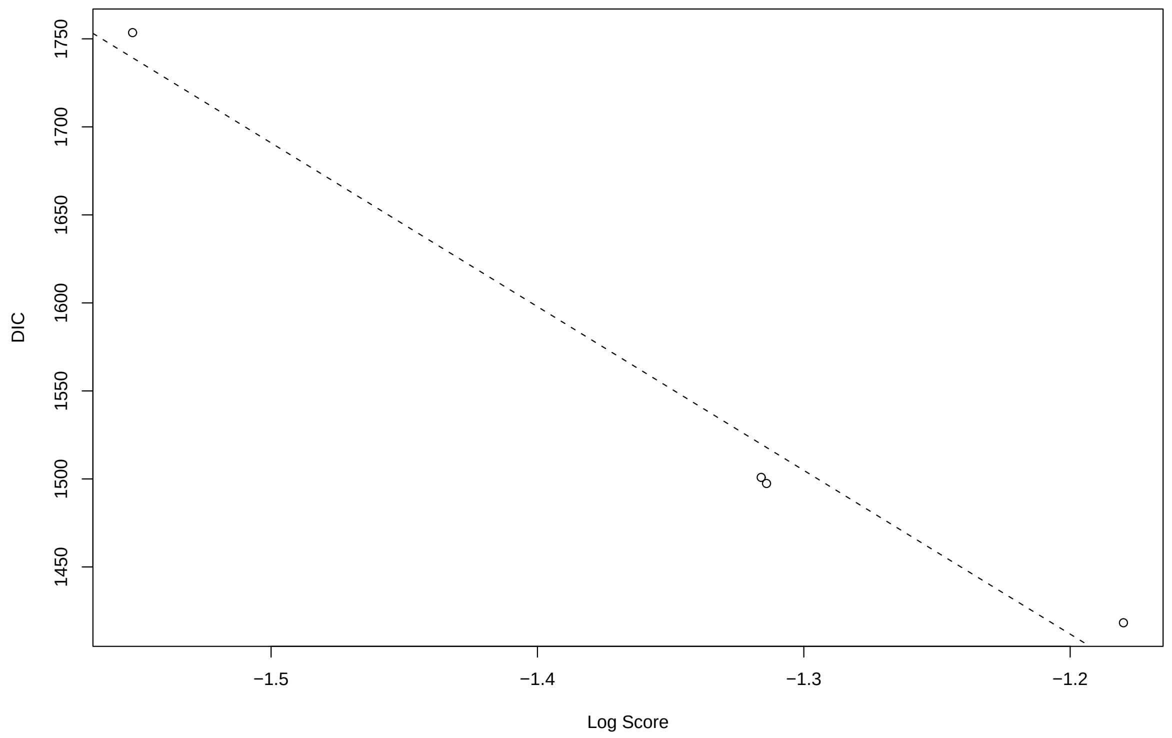
# DIC Example (continued)

**DIC** and **LS** results on these four models:

| Model | $\overline{D(\theta)}$ | $D(\bar{\theta})$ | $\widehat{p}_D$ | $DIC(y)$ | $LS(y)$ |
|---|---|---|---|---|---|
| 1 (Gaussian) | 1749.6 | 1745.6 | 3.99 | 1753.5 | $-1.552$ |
| 2 (Poisson, common $\lambda$) | 1499.9 | 1498.8 | 1.02 | 1500.9 | $-1.316$ |
| 3 (FEPR, different $\lambda$s) | 1495.4 | 1493.4 | 1.98 | 1497.4 | $-1.314$ |
| 4 (REPR) | 1275.7 | 1132.0 | 143.2 | 1418.3 | $-1.180$ |
|  | 1274.7 | 1131.3 | 143.5 | 1418.2 |  |
|  | 1274.4 | 1130.2 | 144.2 | 1418.6 |  |

(3 REPR rows were based on **different monitoring runs**, all of length 10,000, to give idea of Monte Carlo noise level.)

As $\sigma_e \to 0$ in **REPR** model, you get **FEPR** model, with $p_D = 2$ parameters; as $\sigma_e \to \infty$, in effect **all subjects in study have their own** $\lambda$ and $p_D$ would be 572; in between at $\sigma_e \doteq 0.675$ (posterior mean), `WinBUGS` estimates that there are about **143 effective parameters in REPR model**, but its deviance $D(\bar{\theta})$ is so much lower that it **wins DIC contest** hands down.

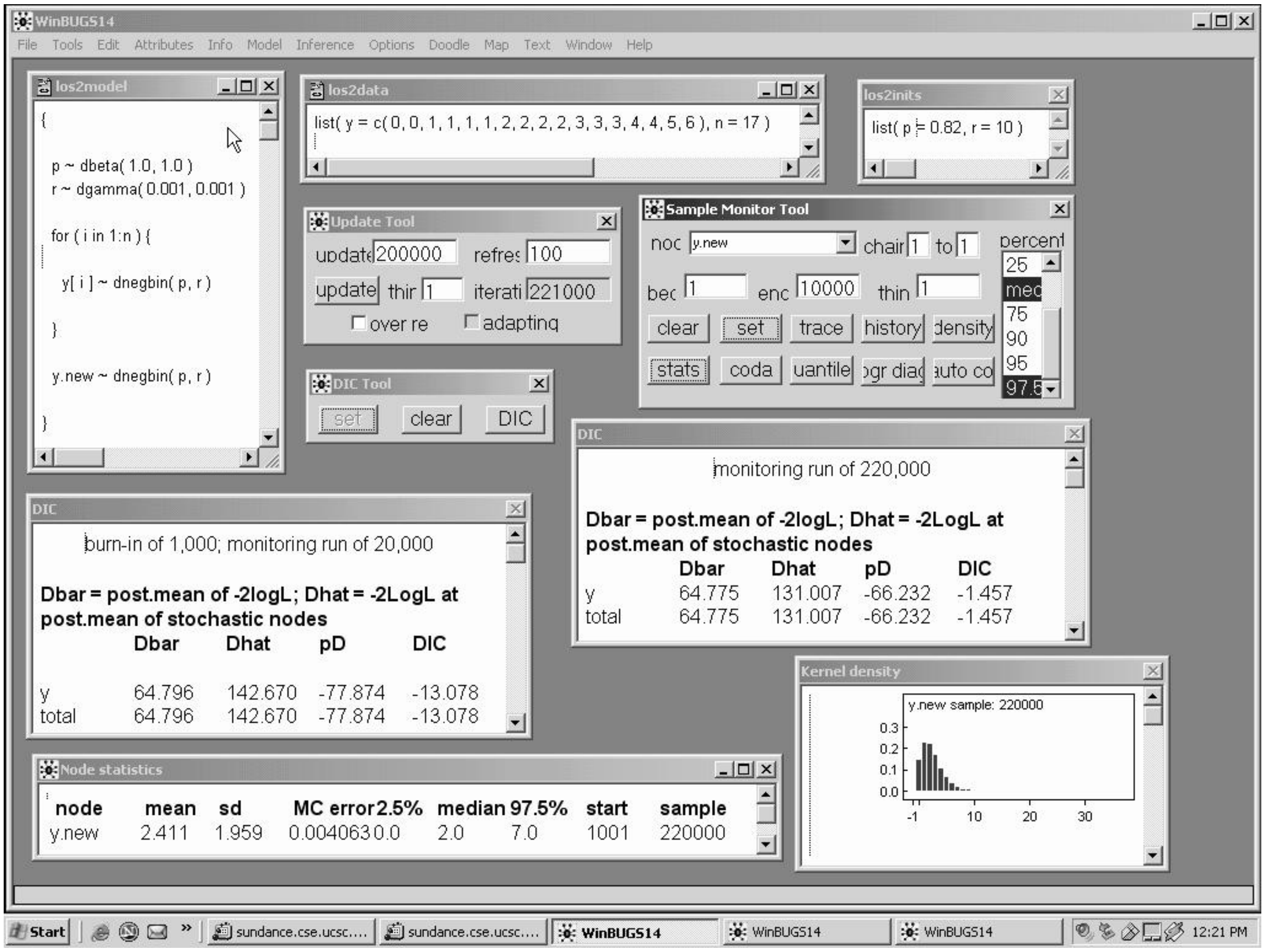


**Correlation** between LS and DIC across these four models is **−0.98**.

## But $DIC$ Can Misbehave

$y = (0,0,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6)$ is a data set generated from the **negative binomial** distribution with parameters $(p, r) = (0.82, 10.8)$ (in `WinBUGS` notation); $y$ has `mean 2.35` and `VTMR 1.22`.

Using **standard diffuse priors** for $p$ and $r$ as in the `BUGS` examples manuals, the **effective number of parameters** $p_D$ for the negative binomial model (which **fits the data quite well**) is estimated at **−66.2**:



los2model

```
{

  p ~ dbeta( 1.0, 1.0 )
  r ~ dgamma( 0.001, 0.001 )

  for ( i in 1:n ) {

    y[ i ] ~ dnegbin( p, r )

  }

  y.new ~ dnegbin( p, r )

}
```

los2data

```
list( y = c( 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 6 ), n = 17 )
```

los2inits

```
list( p = 0.82, r = 10 )
```

Update Tool: updates 200000, refresh 100, update, thin 1, iteration 221000, over relax, adapting

Sample Monitor Tool: node y.new, chains 1 to 1, beg 1, end 10000, thin 1, percentiles 25, median, 75, 90, 95, 97.5; clear, set, trace, history, density, stats, coda, quantiles, bgr diag, auto cor

DIC

```
burn-in of 1,000; monitoring run of 20,000

Dbar = post.mean of -2logL; Dhat = -2LogL at
post.mean of stochastic nodes
          Dbar      Dhat      pD       DIC

y         64.796    142.670   -77.874  -13.078
total     64.796    142.670   -77.874  -13.078
```

DIC

```
monitoring run of 220,000

Dbar = post.mean of -2logL; Dhat = -2LogL at
post.mean of stochastic nodes
          Dbar      Dhat      pD       DIC
y         64.775    131.007   -66.232  -1.457
total     64.775    131.007   -66.232  -1.457
```

Node statistics

```
node    mean   sd     MC error  2.5%  median  97.5%  start  sample
y.new   2.411  1.959  0.004063  0.0   2.0     7.0    1001   220000
```

The basic problem here is that the MCMC estimate of $p_D$ can be **quite poor** if the marginal posteriors for one or more parameters (using the **parameterization** that defines the **deviance**) are **far from normal**; **reparameterization** helps but can still lead to **poor estimates** of $p_D$.

## Fast (Direct) Approximation to $LS_{CV}$

We've seen above that $DIC$ can sometimes provide an accurate and **fast (indirect) approximation** to $LS_{CV}$; what about a **fast** $\boxed{\textbf{direct}}$ **approximation**?

An obvious thing to try is the following **full-sample** version of $LS$: in the one-sample situation, for instance, compute a **single predictive distribution** $p(\cdot|y, M_i)$ for a future data value with each model $M_i$ under consideration, based on the **entire data set** $y$ (without omitting any observations), and define

$$LS_{FS}(M_i|y) = \frac{1}{n} \sum_{j=1}^{n} \log p(y_j|y, M_i). \tag{26}$$

The **naive** approach to calculating $LS_{CV}$, when MCMC is needed to compute the predictive distributions, requires $n$ MCMC runs, **one for each omitted observation**; by contrast $LS_{FS}$ needs only a **single** MCMC run, making its computational speed (a) $n$ **times faster** than naive implementations of $LS_{CV}$ and (b) **equivalent** to that of $DIC$.

• The **log score approach** works equally well with **parametric** and **nonparametric** Bayesian models; $DIC$ is **only defined for parametric models**.

• When **parametric** model $M_i$ with parameter vector $\theta_i$ is fit via **MCMC**, the **predictive ordinate** $p(y^*|y, M_i)$ in $LS_{FS}$ is easy to approximate: with $m$ identically distributed (not necessarily independent) MCMC **monitoring** draws $(\theta_i)^*_k$ from $p(\theta_i|y, M_i)$,

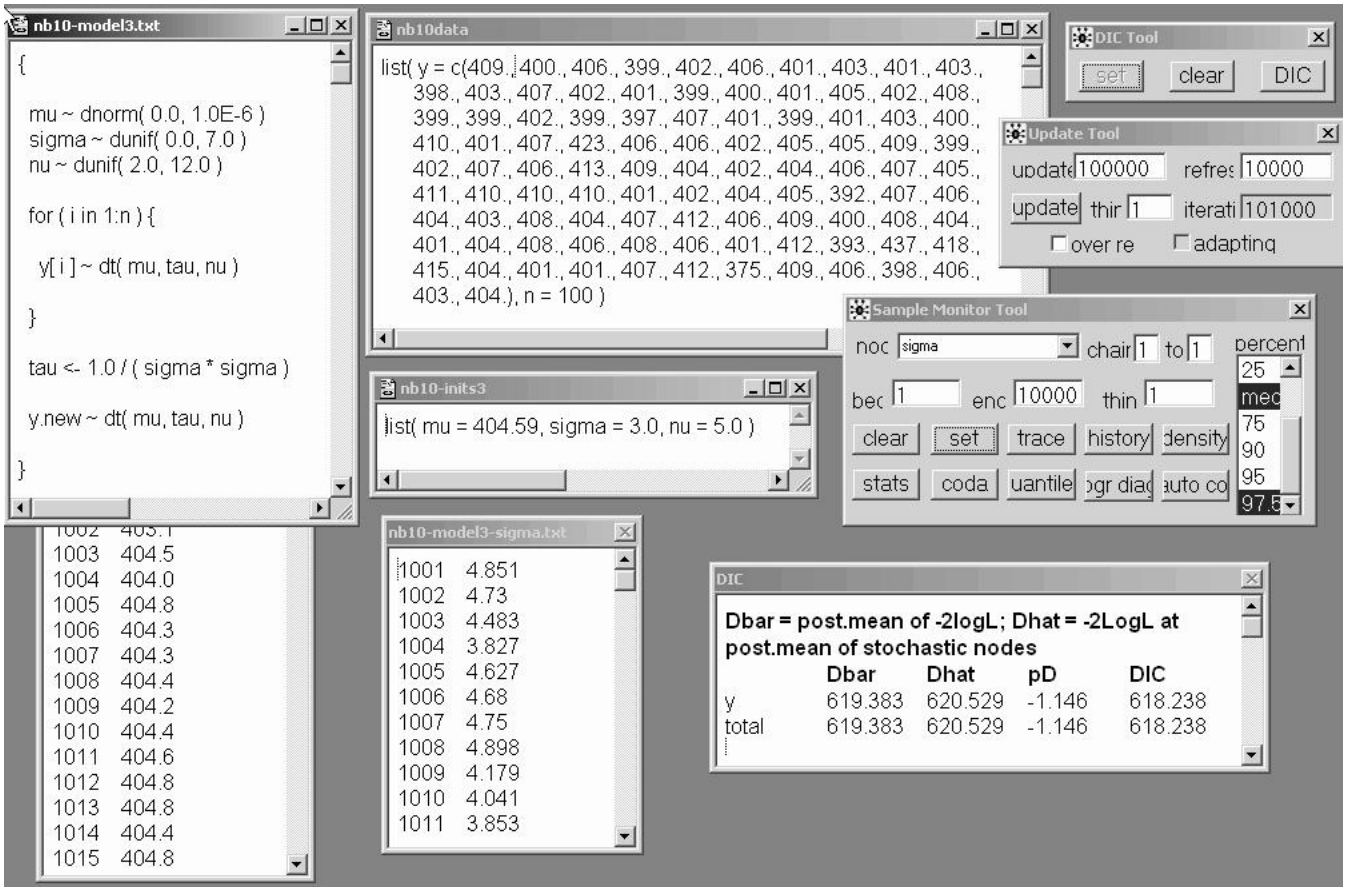
$$\begin{aligned} p(y^*|y, M_i) &= \int p(y^*|\theta_i, M_i)\, p(\theta_i|y, M_i) d\theta_i \\ &= E_{(\theta_i|y, M_i)} \left[ p(y^*|\theta_i, M_i) \right] \qquad (27) \\ &\doteq \frac{1}{m} \sum_{k=1}^{m} p(y^*|(\theta_i)^*_k, M_i). \end{aligned}$$

## Example of $LS_{FS}$ Calculations

**Example.** I'd like to use $LS_{FS}$ and $DIC$ to **compare** the **Gaussian** and $t$ models we discussed earlier for the **NB10** data.

The files `NB10-model-2.txt`, `NB10-data.txt`, and `NB10-initial-values-2.txt` on the course web page contain the `WinBUGS` **implementation** of

$$M_2\text{: } \mu \sim N(0, \text{precision} = 1.0\text{E-6}), \sigma \sim U(0, 9.0),$$
$$\nu \sim U(2.0, 12.0), (y_i | \mu, \sigma, \nu) \overset{\text{IID}}{\sim} t_\nu(\mu, \sigma^2)$$



```
{

  mu ~ dnorm( 0.0, 1.0E-6 )
  sigma ~ dunif( 0.0, 7.0 )
  nu ~ dunif( 2.0, 12.0 )

  for ( i in 1:n ) {

    y[ i ] ~ dt( mu, tau, nu )

  }

  tau <- 1.0 / ( sigma * sigma )

  y.new ~ dt( mu, tau, nu )

}
```

```
list( y = c(409., 400., 406., 399., 402., 406., 401., 403., 401., 403.,
   398., 403., 407., 402., 401., 399., 400., 401., 405., 402., 408.,
   399., 399., 402., 399., 397., 407., 401., 399., 401., 403., 400.,
   410., 401., 407., 423., 406., 406., 402., 405., 405., 409., 399.,
   402., 407., 406., 413., 409., 404., 402., 404., 406., 407., 405.,
   411., 410., 410., 410., 401., 402., 404., 405., 392., 407., 406.,
   404., 403., 408., 404., 407., 412., 406., 409., 400., 408., 404.,
   401., 404., 408., 406., 408., 406., 401., 412., 393., 437., 418.,
   415., 404., 401., 401., 407., 412., 375., 409., 406., 398., 406.,
   403., 404.), n = 100 )
```

```
list( mu = 404.59, sigma = 3.0, nu = 5.0 )
```

```
1002    403.1
1003    404.5
1004    404.0
1005    404.8
1006    404.3
1007    404.3
1008    404.4
1009    404.2
1010    404.4
1011    404.6
1012    404.8
1013    404.8
1014    404.4
1015    404.8
```

```
1001    4.851
1002    4.73
1003    4.483
1004    3.827
1005    4.627
1006    4.68
1007    4.75
1008    4.898
1009    4.179
1010    4.041
1011    3.853
```

```
Dbar = post.mean of -2logL; Dhat = -2LogL at
post.mean of stochastic nodes
         Dbar      Dhat      pD       DIC
y        619.383   620.529   -1.146   618.238
total    619.383   620.529   -1.146   618.238
```

I collect **100,000 monitoring iterations** for $M_2$, remembering to hit the `set` button on the `DIC tool` before the monitoring begins; I use the `coda` button to store the $\mu, \sigma$, and $\nu$ columns of the MCMC data set in files called `nb10-model-2-mu.txt`, `nb10-model-2-sigma.txt`, and `nb10-model-2-nu.txt`, respectively; and I hit the `DIC` button on the `DIC tool` to record that the $DIC$ value for this model is **618.2** (note that $DIC$ has **misbehaved** again: $p_D$ is estimated to be **-1.1**).

## $LS_{FS}$ Calculations (continued)

I go through a **similar process** with the files `NB10-model-1.txt`, `NB10-data.txt`, and `NB10-initial-values-1.txt` to fit

$$M_1\text{: } \mu \sim N(0, \text{precision} = 1.0\text{E-6}), \sigma \sim U(0, 9.0),$$
$$(y_i|\mu,\sigma) \overset{\text{IID}}{\sim} N(\mu, \sigma^2)$$

and store the $\mu$ and $\sigma$ columns of the MCMC data set in files called `nb10-model-1-mu.txt` and `nb10-model-1-sigma.txt`, respectively; this time the $DIC$ value is **660.1** and DIC is **better-behaved** ($p_D$ is estimated to be **1.9**, which is **about right**).

On the basis of $DIC$ I would conclude that $M_2$ (**618.2** with 3 parameters) is (substantially) better than $M_1$ (**660.1** with 2).

Here is some `R` **code** (also available on the web page) to compute the **log score** values for **both models**.

```
> y <- dget( "nb10-data.txt" )

> y <- sort( y$y )

> mu.G <- matrix( scan( "nb10-model-1-mu.txt" ),
    100000, 2, byrow = T )[ , 2 ]

> sigma.G <- matrix( scan( "nb10-model-1-sigma.txt" ),
    100000, 2, byrow = T )[ , 2 ]

> mu.t <- matrix( scan( "nb10-model-2-mu.txt" ),
    100000, 2, byrow = T )[ , 2 ]

> sigma.t <- matrix( scan( "nb10-model-2-sigma.txt" ),
    100000, 2, byrow = T )[ , 2 ]

> nu.t <- matrix( scan( "nb10-model-2-nu.txt" ),
    100000, 2, byrow = T )[ , 2 ]
```

## $LS_{FS}$ Calculations (continued)

```
> dt.s <- function( y, mu, sigma, nu ) {

>   exp( lgamma( ( nu + 1 ) / 2 ) - ( ( nu + 1 ) / 2 ) *
>     log( 1 + ( y - mu )^2 / ( nu * sigma^2 ) ) -
>     lgamma( nu / 2 ) - log( nu * pi ) / 2 - log( sigma ) )

> }

> LS.contributions <- matrix( 0, 100, 2 )

> for ( j in 1:100 ) {

>   LS.contributions[ j, 1 ] <- log( mean( dt.s( y[ j ],
>     mu.t, sigma.t, nu.t ) ) )

>   LS.contributions[ j, 2 ] <- log( mean( dnorm( y[ j ],
>     mu.G, sigma.G ) ) )

> }

> cbind( y, LS.contributions,
>   0 + LS.contributions[ , 1 ] > LS.contributions[ , 2 ] )

                                t
                                better
                                than
              t        Gaussian G

  [1,] 375 -8.586208 -12.204954 1
  [2,] 392 -5.349809  -4.639139 0
  [3,] 393 -5.077313  -4.362693 0
  [4,] 397 -3.903555  -3.475233 0
  [5,] 398 -3.602015  -3.309458 0
  [6,] 398 -3.602015  -3.309458 0
  [7,] 399 -3.307381  -3.166624 0
  [8,] 399 -3.307381  -3.166624 0
```

## $LS_{FS}$ Calculations (continued)

```
 [9,] 399 -3.307381  -3.166624 0
[10,] 399 -3.307381  -3.166624 0
[11,] 399 -3.307381  -3.166624 0
[12,] 399 -3.307381  -3.166624 0
[13,] 399 -3.307381  -3.166624 0
[14,] 400 -3.028685  -3.046933 1
[15,] 400 -3.028685  -3.046933 1
[16,] 400 -3.028685  -3.046933 1
[17,] 400 -3.028685  -3.046933 1
[18,] 401 -2.778176  -2.950552 1
[19,] 401 -2.778176  -2.950552 1
[20,] 401 -2.778176  -2.950552 1
[21,] 401 -2.778176  -2.950552 1
[22,] 401 -2.778176  -2.950552 1
[23,] 401 -2.778176  -2.950552 1
[24,] 401 -2.778176  -2.950552 1
[25,] 401 -2.778176  -2.950552 1
[26,] 401 -2.778176  -2.950552 1
[27,] 401 -2.778176  -2.950552 1
[28,] 401 -2.778176  -2.950552 1
[29,] 401 -2.778176  -2.950552 1
[30,] 402 -2.571441  -2.877618 1
[31,] 402 -2.571441  -2.877618 1
[32,] 402 -2.571441  -2.877618 1
[33,] 402 -2.571441  -2.877618 1
[34,] 402 -2.571441  -2.877618 1
[35,] 402 -2.571441  -2.877618 1
[36,] 402 -2.571441  -2.877618 1
[37,] 402 -2.571441  -2.877618 1
[38,] 403 -2.426129  -2.828236 1
[39,] 403 -2.426129  -2.828236 1
[40,] 403 -2.426129  -2.828236 1
[41,] 403 -2.426129  -2.828236 1
[42,] 403 -2.426129  -2.828236 1
[43,] 403 -2.426129  -2.828236 1
[44,] 404 -2.358212  -2.802475 1
```

## $LS_{FS}$ Calculations (continued)

```
[45,] 404 -2.358212  -2.802475 1
[46,] 404 -2.358212  -2.802475 1
[47,] 404 -2.358212  -2.802475 1
[48,] 404 -2.358212  -2.802475 1
[49,] 404 -2.358212  -2.802475 1
[50,] 404 -2.358212  -2.802475 1
[51,] 404 -2.358212  -2.802475 1
[52,] 404 -2.358212  -2.802475 1
[53,] 405 -2.376305  -2.800373 1
[54,] 405 -2.376305  -2.800373 1
[55,] 405 -2.376305  -2.800373 1
[56,] 405 -2.376305  -2.800373 1
[57,] 405 -2.376305  -2.800373 1
[58,] 406 -2.477698  -2.821932 1
[59,] 406 -2.477698  -2.821932 1
[60,] 406 -2.477698  -2.821932 1
[61,] 406 -2.477698  -2.821932 1
[62,] 406 -2.477698  -2.821932 1
[63,] 406 -2.477698  -2.821932 1
[64,] 406 -2.477698  -2.821932 1
[65,] 406 -2.477698  -2.821932 1
[66,] 406 -2.477698  -2.821932 1
[67,] 406 -2.477698  -2.821932 1
[68,] 406 -2.477698  -2.821932 1
[69,] 406 -2.477698  -2.821932 1
[70,] 407 -2.649778  -2.867123 1
[71,] 407 -2.649778  -2.867123 1
[72,] 407 -2.649778  -2.867123 1
[73,] 407 -2.649778  -2.867123 1
[74,] 407 -2.649778  -2.867123 1
[75,] 407 -2.649778  -2.867123 1
[76,] 407 -2.649778  -2.867123 1
[77,] 407 -2.649778  -2.867123 1
[78,] 408 -2.875393  -2.935880 1
[79,] 408 -2.875393  -2.935880 1
[80,] 408 -2.875393  -2.935880 1
```

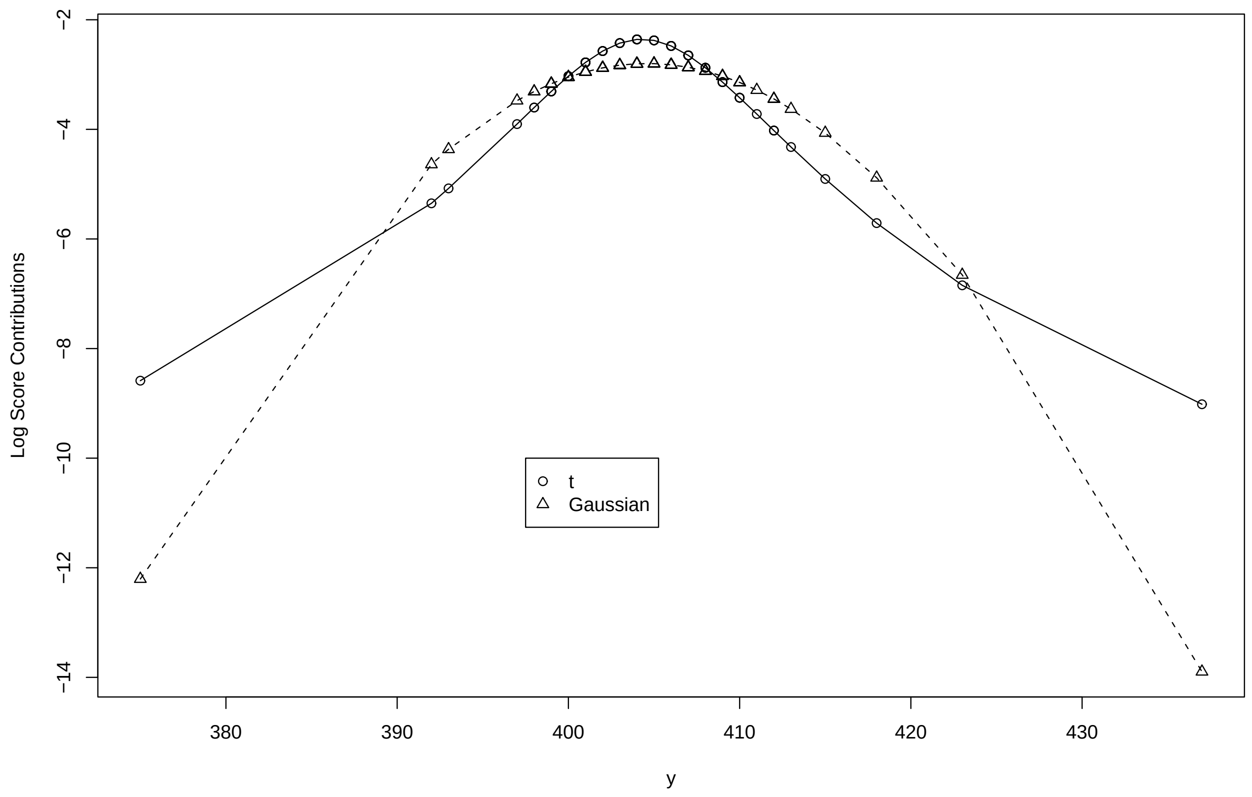
## $LS_{FS}$ Calculations (continued)

```
 [81,] 408 -2.875393  -2.935880 1
 [82,] 408 -2.875393  -2.935880 1
 [83,] 409 -3.137771  -3.028107 0
 [84,] 409 -3.137771  -3.028107 0
 [85,] 409 -3.137771  -3.028107 0
 [86,] 409 -3.137771  -3.028107 0
 [87,] 409 -3.137771  -3.028107 0
 [88,] 410 -3.422943  -3.143672 0
 [89,] 410 -3.422943  -3.143672 0
 [90,] 410 -3.422943  -3.143672 0
 [91,] 410 -3.422943  -3.143672 0
 [92,] 411 -3.720225  -3.282413 0
 [93,] 412 -4.021816  -3.444136 0
 [94,] 412 -4.021816  -3.444136 0
 [95,] 412 -4.021816  -3.444136 0
 [96,] 413 -4.322196  -3.628616 0
 [97,] 415 -4.905384  -4.064801 0
 [98,] 418 -5.710652  -4.882504 0
 [99,] 423 -6.845648  -6.656119 0
[100,] 437 -9.016222 -13.896384 1

> sum( LS.contributions[ , 1 ] > LS.contributions[ , 2 ] ) /
>   length( y )

[1] 0.71

# Thus t model is predictively better than Gaussian for
# 71% of the data points.

LS.t <- mean( LS.contributions[ , 1 ] )

LS.G <- mean( LS.contributions[ , 2 ] )

c( LS.t, LS.G )

[1] -3.082331 -3.262142
```

## $LS_{FS}$ Calculations (continued)

Although it's not immediately **obvious**, the **log score** for the $t$ model ($-3.08$) is **substantially higher** than that for the Gaussian model ($-3.26$), so $LS$ and $DIC$ have reached the **same conclusion** here.

```
> plot( y, LS.contributions[ , 1 ],
>   ylim = c( min( LS.contributions ),
>   max( LS.contributions ) ),
>   ylab = 'Log Score Contributions' )

> lines( y, LS.contributions[ , 1 ], lty = 1 )

> points( y, LS.contributions[ , 2 ], pch = 2 )

> lines( y, LS.contributions[ , 2 ], lty = 2 )

> legend( 397.5, -10, c( "t", "Gaussian" ), pch = c( 1, 2 ) )
```



The $t$ model **fits better** both **in the tails** (where the **most influential observations** are from the Gaussian point of view) and in the **center** (where **most** of the data values are).

## Asymptotic Properties of $LS_{FS}$

Recall the claim that $LS_{CV}$ **approximates expectation of logarithmic utility**:

$$E\left[U(M_i|y)\right] \approx LS_{CV} = \frac{1}{n}\sum_{j=1}^{n} \log p(y_j|M_i, y_{-j}) \quad (28)$$

Berger et al. (2005) recently proved that **difference** between LHS and RHS of (28) **does not vanish** for large $n$ but is instead $O_p(\sqrt{n})$.

(However **unpleasant**, this fact does not automatically invalidate use of $LS_{CV}$ as estimated expected utility, since when comparing two models we effectively look at the **difference** between two $LS_{CV}$ values, and the discrepancy should largely **cancel out**.)

We have proved for a simple model that $LS_{FS}$ is **free from this deficiency**: the difference between $E[U(M_i|y)]$ and $LS_{FS} = \frac{1}{n}\sum_{j=1}^{n} \log p(y_j|y, M_i)$ is $O_p(1)$ (we expect the **general proof** to go through as well).

$\boxed{Q:}$ Does this **asymptotic superiority** of $LS_{FS}$ over $LS_{CV}$ translate into **better small-sample performance**?

## $LS_{CV}$, $LS_{FS}$ and $DIC$ Model Discrimination

We now have **three behavioral rules**: **maximize** $LS_{CV}$, **maximize** $LS_{FS}$, **minimize** $DIC$.

With (e.g.) two models to choose between, how **accurately** do these behavioral rules **discriminate** between $M_1$ and $M_2$?

$\boxed{\textbf{Example:}}$ Recall that in **earlier simulation study**, for $i = 1, \dots, n$, and with **diffuse priors**, we considered

$$M_1\text{:} \left\{ \begin{array}{ccc} \lambda & \sim & p(\lambda) \\ (y_i|\lambda) & \overset{\text{IID}}{\sim} & \text{Poisson}(\lambda) \end{array} \right\} \quad \text{versus}$$

$$M_2\text{:} \left\{ \begin{array}{ccc} (\beta_0, \sigma^2) & \sim & p(\beta_0, \sigma^2) \\ (y_i|\lambda_i) & \overset{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \beta_0 + e_i \\ e_i & \overset{\text{IID}}{\sim} & N(0, \sigma^2) \end{array} \right\}$$

## Model Discrimination (continued)

As **extension** of previous simulation study, we generated data from $M_2$ and computed $LS_{CV}$, $LS_{FS}$, and $DIC$ for models $M_1$ and $M_2$ in **full-factorial grid** $\{n = 32, 42, 56, 100\}$, $\{\beta_0 = 0.0, 1.0\}$, $\sigma^2 = 0.1, 0.25, 0.5, 1.0, 1.5, 2.0\}$, with **100** simulation replications in each cell, and monitored **percentages of correct model choice** (here $M_2$ is always correct).

**Examples** of **results** for (e.g.) $LS_{CV}$:

$n = 32$

% Correct Decision

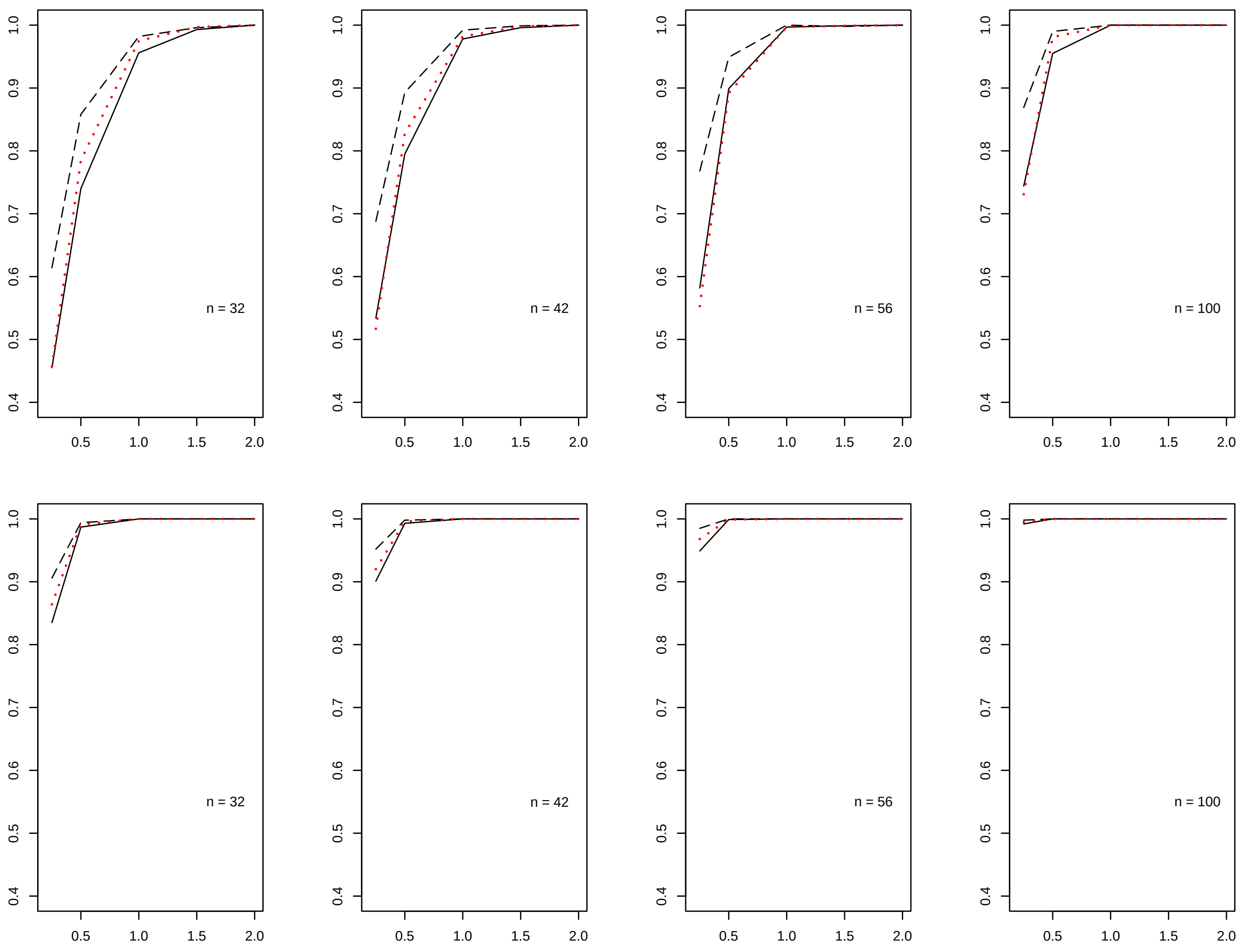
| | $\beta_0$ | |
|---|---|---|
| $\sigma^2$ | 0 | 1 |
| 0.10 | 31 | 47 |
| 0.25 | 49 | 85 |
| 0.50 | 76 | 95 |
| 1.00 | 97 | 100 |
| 1.50 | 98 | 100 |
| 2.00 | 100 | 100 |

Mean Absolute Difference in $LS_{CV}$

| | $\beta_0$ | |
|---|---|---|
| $\sigma^2$ | 0 | 1 |
| 0.10 | 0.001 | 0.002 |
| 0.25 | 0.002 | 0.013 |
| 0.50 | 0.017 | 0.221 |
| 1.00 | 0.237 | 4.07 |
| 1.50 | 1.44 | 17.4 |
| 2.00 | 12.8 | 63.9 |

Even with $n$ only **32**, $LS_{CV}$ makes the right model choice **more than 90% of the time** when $\sigma^2 > 0.5$ for $\beta_0 = 1$ and when $\sigma^2 > 1.0$ for $\beta_0 = 0$.

# Model Discrimination (continued)



The plots above compare **Bayesian decision-theoretic power curves** for $LS_{CV}$ (**solid** lines), $LS_{FS}$ (**long dotted** lines), and $DIC$ (**short dotted** lines) (column 1: $\beta_0 = 0$; column 2: $\beta_0 = 1$).

Remarkably, not only is $LS_{FS}$ **much quicker computationally** than $LS_{CV}$, it's also **more accurate** at identifying the correct model than $LS_{CV}$ or $DIC$.

To summarize, in **computational efficiency**

$$LS_{CV} < DIC \doteq LS_{FS} \tag{29}$$

and in **fixed-** and **random-effects Poisson modeling** the results in **model discrimination power** are

$$LS_{CV} \doteq DIC < LS_{FS} \tag{30}$$

## Why Not Bayes Factors?

Much has been written about use of **Bayes factors** for **model choice** (e.g., Jeffreys 1939, Kass and Raftery 1995; excellent recent book by O'Hagan and Forster 2004 devotes almost **40 pages** to this topic).

Why not use **probability scale** to choose between $M_1$ and $M_2$?

$$\begin{aligned} \left[\frac{p(M_1|y)}{p(M_2|y)}\right] &= \left[\frac{p(M_1)}{p(M_2)}\right] \cdot \left[\frac{p(y|M_1)}{p(y|M_2)}\right] \qquad (31) \\ \left(\begin{array}{c}\text{posterior}\\\text{odds}\end{array}\right) &= \left(\begin{array}{c}\text{prior}\\\text{odds}\end{array}\right) \cdot \left(\begin{array}{c}\text{Bayes}\\\text{factor}\end{array}\right) \end{aligned}$$

Kass and Raftery (1995) **note** that

$$\begin{aligned} \log\left[\frac{p(y|M_1)}{p(y|M_2)}\right] &= \log p(y|M_1) - \log p(y|M_2) \qquad (32) \\ &= LS^*(M_1|y) - LS^*(M_2|y), \end{aligned}$$

**where**

$$\begin{aligned} LS^*(M_i|y) &\equiv \log p(y|M_i) \\ &= \log\left[p(y_1|M_i)\, p(y_2|y_1, M_i) \cdots p(y_n|y_1, \ldots, y_{n-1}, M_i)\right] \\ &= \log p(y_1|M) + \sum_{j=2}^{n} \log p(y_j|y_1, \ldots, y_{j-1}, M_i). \end{aligned}$$

Thus **log Bayes factor** equals **difference** between models in **something that looks like a log score**, i.e., aren't $LS_{CV}$ and $LS_{FS}$ equivalent to choosing $M_i$ whenever the Bayes factor in favor of $M_i$ **exceeds 1**?

## LS $\neq$ BF

$\boxed{\text{No}}$; crucially, LS* is defined via **sequential** prediction of $y_2$ from $y_1$, $y_3$ from $(y_1, y_2)$, etc., whereas $LS_{CV}$ and $LS_{FS}$ are based on **averaging over all possible out-of-sample predictions**.

This distinction **really matters**: as is well known, with **diffuse priors** Bayes factors are **hideously sensitive** to particular **form** in which diffuseness is **specified**, but this defect is **entirely absent** from $LS_{CV}$ and $LS_{FS}$ (and from other properly-defined **utility-based model choice criteria**).

$\boxed{\textbf{Example:}}$ Integer-valued data $y = (y_1, \ldots, y_n)$;

$M_1 =$ **Geometric**$(\theta_1)$ likelihood with **Beta**$(\alpha_1, \beta_1)$ prior on $\theta_1$;

$M_2 =$ **Poisson**$(\theta_2)$ likelihood with **Gamma**$(\alpha_2, \beta_2)$ prior on $\theta_2$.

**Bayes factor** in favor of $M_1$ over $M_2$ is

$$\frac{\Gamma(\alpha_1+\beta_1)\Gamma(n+\alpha_1)\Gamma(n\bar{y}+\beta_1)\Gamma(\alpha_2)(n+\beta_2)^{n\bar{y}+\alpha_2}\left(\prod_{i=1}^n y_i!\right)}{\Gamma(\alpha_1)\Gamma(\beta_1)\Gamma(n+n\bar{y}+\alpha_1+\beta_1)\Gamma(n\bar{y}+\alpha_2)\beta_2^{\alpha_2}}.$$

**Diffuse** priors: take $(\alpha_1, \beta_1) = (1, 1)$ and $(\alpha_2, \beta_2) = (\epsilon, \epsilon)$ for some $\epsilon > 0$.

**Bayes factor** reduces to

$$\frac{\Gamma(n+1)\Gamma(n\bar{y}+1)\Gamma(\epsilon)(n+\epsilon)^{n\bar{y}+\epsilon}\left(\prod_{i=1}^n y_i!\right)}{\Gamma(n+n\bar{y}+2)\Gamma(n\bar{y}+\epsilon)\epsilon^{\epsilon}}.$$

## LS $\neq$ BF (continued)

This goes to $+\infty$ as $\epsilon \downarrow 0$, i.e., you can make the evidence in **favor** of the **Geometric model** over the **Poisson** as **large** as you want as a function of a quantity near 0 that **scientifically** you have **no basis** to specify.

By **contrast**, e.g.,

$$\begin{aligned} LS_{CV}(M_1|y) &= \log\Big[\frac{(\alpha_1+n-1)\Gamma(\beta_1+s)}{\Gamma(\alpha_1+n+\beta_1+s)}\Big] \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\log\Big[\frac{\Gamma(\alpha_1+n-1+\beta_1+s_i)}{\Gamma(\beta_1+s_i)}\Big] \end{aligned}$$

and

$$\begin{aligned} LS_{CV} = (M_2|y) &= \frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{\Gamma(\alpha_2+s)}{\Gamma(y_i+1)\Gamma(\alpha_2+s_i)}\right. \\ &\quad \left.\cdot\Big(\frac{\beta_2+n}{\beta_2+n+1}\Big)^{\alpha_2+s_i}\Big(\frac{1}{\beta_2+n+1}\Big)^{y_i}\right] \end{aligned}$$

(with similar expressions for $LS_{FS}$); both of these quantities are **entirely stable** as a function of $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ near zero.

(Various **attempts** have been made to **fix** this defect of Bayes factors, e.g., {partial, intrinsic, fractional} Bayes factors, well calibrated priors, conventional priors, intrinsic priors, expected posterior priors, ... (e.g., Pericchi 2004); all of these methods appear to require an appeal to **ad-hockery** which is **not required by the log score approach**.)

(Some **bridges** can be built between **LS** and **BF**, e.g., Berger et al. (2005) re-interpret $LS_{CV}$ as the "Gelfand-Dey (1994) **predictive Bayes factor**" $BF^{GD}$; connections like these are the subject of **ongoing investigation**.)

## What $LS_{FS}$ Is Not

(1) **Likelihood** part of (parametric) model $M_j$: $(y_i|\theta_j, M_j) \overset{\text{IID}}{\sim} p(y_i|\theta_j, M_j) (j = 1, 2)$, with **prior** $p(\theta_j|M_j)$ for model $M_j$.

**Ordinary Bayes factor** involves comparing quantities of the form

$$\begin{aligned} p(y|M_j) &= \int \left[ \prod_{i=1}^{n} p(y_i|\theta_j, M_j) \right] p(\theta_j|M_j)\, d\theta_j, \\ &= E_{(\theta_j|M_j)} L(\theta_j|y, M_j), \end{aligned} \qquad (33)$$

i.e., Bayes factor involves comparing **expectations** of **likelihoods** with respect to the **priors** in the models under comparison (this is **why ordinary Bayes factors behave so badly with diffuse priors**).

Aitkin (1991; **posterior Bayes factors**): compute expectations instead with respect to the **posteriors**, i.e., $\boxed{\textbf{PBF:}}$ favor model $M_1$ if log $\bar{L}_1^A >$ log $\bar{L}_2^A$, where

$$\log \bar{L}_j^A = \log \int \left[ \prod_{i=1}^{n} p(y_i|\theta_j, M_j) \right] p(\theta_j|y, M_j)\, d\theta_j. \qquad (34)$$

This **solves** the problem of sensitivity to a diffuse prior but **creates new problems of its own**, e.g., it's **incoherent**.

It may **seem** at first glance (e.g., O'Hagan and Forster (2004) think so) that **PBF is the same thing as** $LS_{FS}$: favor model $M_1$ if

$$n\, LS_{FS}(M_1|y) > n\, LS_{FS}(M_2|y). \qquad (35)$$

But **not so**:

$$nLS_{FS}(M_j|y) = \log \prod_{i=1}^{n} \left[ \int p(y_i|\theta_j, M_j)\, p(\theta_j|y, M_j)\, d\theta_j \right], \qquad (36)$$

and this is **not the same** because the **integral** and **product** operators **do not commute**.

## What $LS_{FS}$ Is Not (continued)

Also, some people (e.g., Geweke (2005)) like to compare models based on the **posterior expectation of the log likelihood** (this is **one of the ingredients** in $DIC$), and this is **not the same** as $LS_{FS}$ either: by **Jensen's inequality**

$$\begin{aligned} n LS_{FS}(M_j|y) &= \sum_{i=1}^{n} \log p(y_i|y, M_j) \\ &= \sum_{i=1}^{n} \log \int p(y_i|\theta_j, M_j)\, p(\theta_j|y, M_j)\, d\theta_j \\ &= \sum_{i=1}^{n} \log E_{(\theta_j|y,M_j)} L(\theta_j|y_i, M_j) \\ &> \sum_{i=1}^{n} E_{(\theta_j|y,M_j)} \log L(\theta_j|y_i, M_j) \qquad (37) \\ &= E_{(\theta_j|y,M_j)} \sum_{i=1}^{n} \log L(\theta_j|y_i, M_j) \\ &= E_{(\theta_j|y,M_j)} \log \prod_{i=1}^{n} L(\theta_j|y_i, M_j) \\ &= E_{(\theta_j|y,M_j)} \log L(\theta_j|y, M_j). \end{aligned}$$

# When Is a Model Good Enough?

$LS_{FS}$ **method** described here (**not** LS$^*$ method) can **stably** and **reliably** help in choosing between $M_1$ and $M_2$; but suppose $M_1$ has a (substantially) **higher** $LS_{FS}$ than $M_2$.

This doesn't say that $M_1$ is **adequate**—it just says that $M_1$ **is better than** $M_2$, i.e., what about model specification question (2): Is $M_1$ **good enough**?

As mentioned above, a **full judgment of adequacy** requires **real-world input** (to what purpose will the model be put?), but you can answer a somewhat related question—**could the data have arisen from a given model**?—in a general way by **simulating** from that model many times, **developing** a distribution of (e.g.) $LS_{FS}$ values, and **seeing how unusual** the actual data set's log score is in this distribution (Draper and Krnjajić 2004).

This is related to the **posterior predictive model-checking** method of Gelman, Meng and Stern (1996); however, this sort of thing cannot be done **naively**, or result will be **poor calibration**—indeed, Robins et al. (2000) demonstrated that the Gelman et al. procedure may be (sharply) **conservative**.

Using **modification** of idea in Robins et al., we have developed method for **accurately calibrating the log score scale**.

**Inputs** to our procedure: (1) A **data set** (e.g., with regression structure); (2) A **model** (can be parametric, non-parametric, or semi-parametric).

**Simple example:** data set $y = (1, 2, 2, 3, 3, 3, 4, 6, 7, 11)$, $n = 10$.

Given **model** $(*)$

$$\begin{aligned} (\lambda) &\sim \text{Gamma}(0.001, 0.001) \\ (y_i|\lambda) &\overset{\text{IID}}{\sim} \text{Poisson}(\lambda) \end{aligned} \qquad (38)$$

# Calibrating $LS_{FS}$ Scale

**Step 1:**

Calculate $LS_{FS}$ for this data set; say get $LS_{FS} = -1.1$; call this **actual log score** (ALS).

Obtain posterior for $\lambda$ given $y$ based on this data set; call this **actual posterior**.

**Step 2:**

```
for ( i in 1:m1 ) {

  make a lambda draw from the actual posterior;
    call it lambda[ i ]

  generate a data set of size n from the second
    line of model (*) above, using
    lambda = lambda[ i ]

  compute the log score for this generated
    data set; call it LS[ i ]

}
```

Output of this loop is a vector of log scores; call this **V.LS**.

Locate ALS in distribution of $LS_{FS}$ values by computing percentage of $LS_{FS}$ values in V.LS that are $\leq$ ALS; call this percentage **unadjusted actual tail area** (say this is 0.22).

So far this is just Gelman et al. with $LS_{FS}$ as the **discrepancy function**.

We know from our own simulations and the literature (Robins et al. 2000) that this tail area (a $p$-value for a **composite null hypothesis**, e.g., Poisson($\lambda$) with $\lambda$ unspecified) is **conservative**, i.e., with the 0.22 example above an adjusted version of it that is well calibrated would be **smaller**.

## Calibrating $LSFS$ Scale (continued)

We've **modified** and implemented one of the ways suggested by Robins et al., and we've shown that it does indeed work even in rather small-sample situations, although our approach to implementing the basic idea can be **computationally intensive**.

$\boxed{\textbf{Step 3:}}$

```
for ( j in 1:m2 ){

  make a lambda draw from the actual posterior;
    call it lambda*.

  generate a data set of size n from the second line
    of model (*) above, using lambda = lambda*;
    call this the simulated data set

  repeat steps 1, 2 above on this
    simulated data set

}
```

The result will be a vector of unadjusted tail areas; call this **V.P**.

Compute the percentage of tail areas in V.P that are $\leq$ the unadjusted actual tail area; this is the **adjusted actual tail area**.

## Calibrating $LS_{FS}$ Scale (continued)

The claim is that the 3-step procedure above is **well-calibrated**, i.e., if the sampling part of model $(*)$ really did generate the observed data, the distribution of adjusted actual tail areas obtained in this way would be **uniform**, apart from simulation noise.

Step 3 in this procedure **solves the calibration problem** by applying the old idea that if $X \sim F_X$ then $F_X(X) \sim U(0,1)$.

This claim can be verified by building a **big loop** around steps 1–3 as follows:

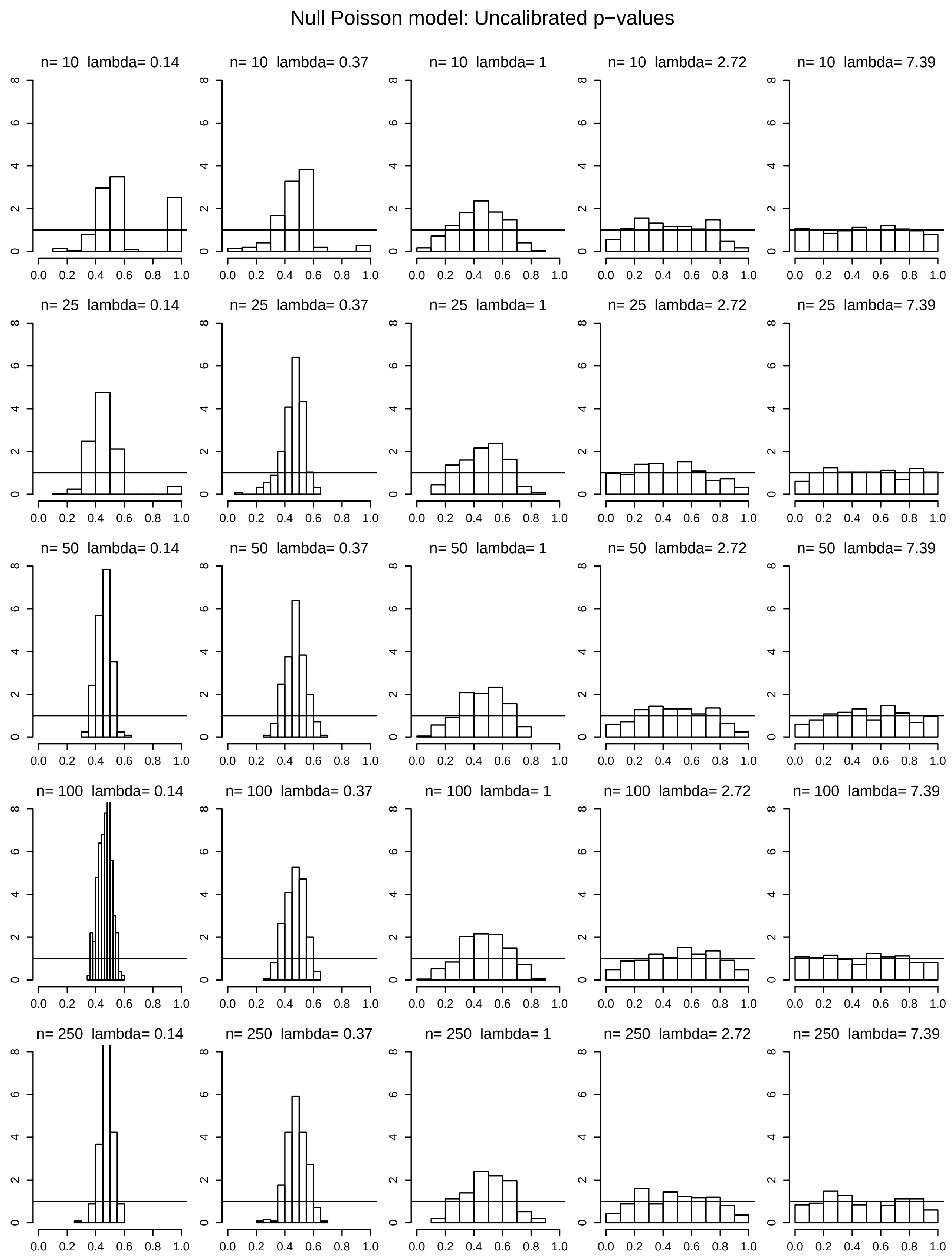
```
Choose a lambda value of interest; call it lambda.sim

for ( k in 1:m3 ) {

  generate a data set of size n from the
    second line of model (*) above, using
    lambda = lambda.sim; call this the
    validation data set

  repeat steps 1-3 on the validation data set

}
```

The result will be a vector of **adjusted P-values**; call this **V.Pa**.

We have **verified** (via simulation) in several simple (and some less simple) situations that the values in V.Pa are close to $U(0,1)$ in distribution.

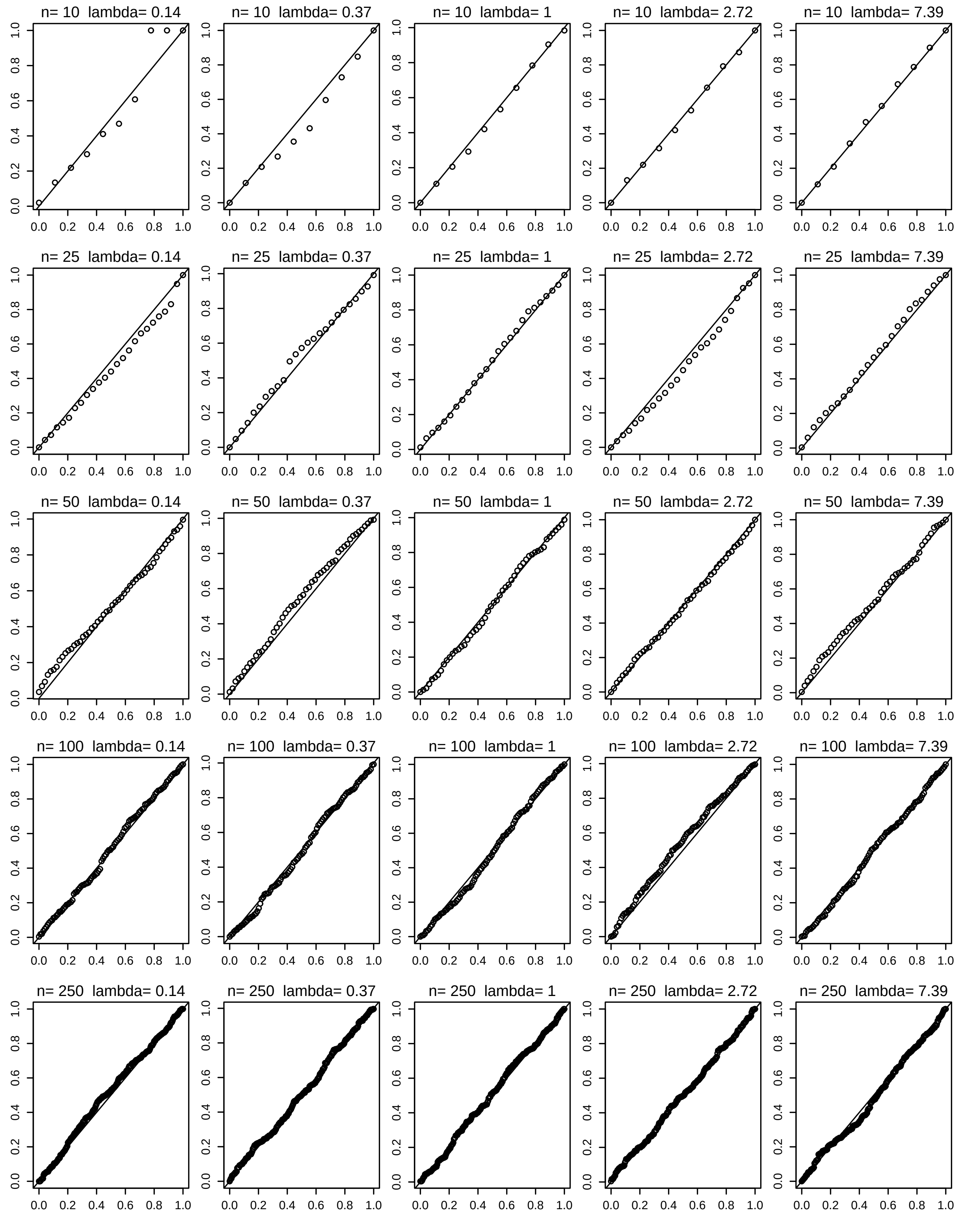
Two **examples**—Poisson($\lambda$) and Gaussian($\mu, \sigma^2$):

# Uncalibrated p-values

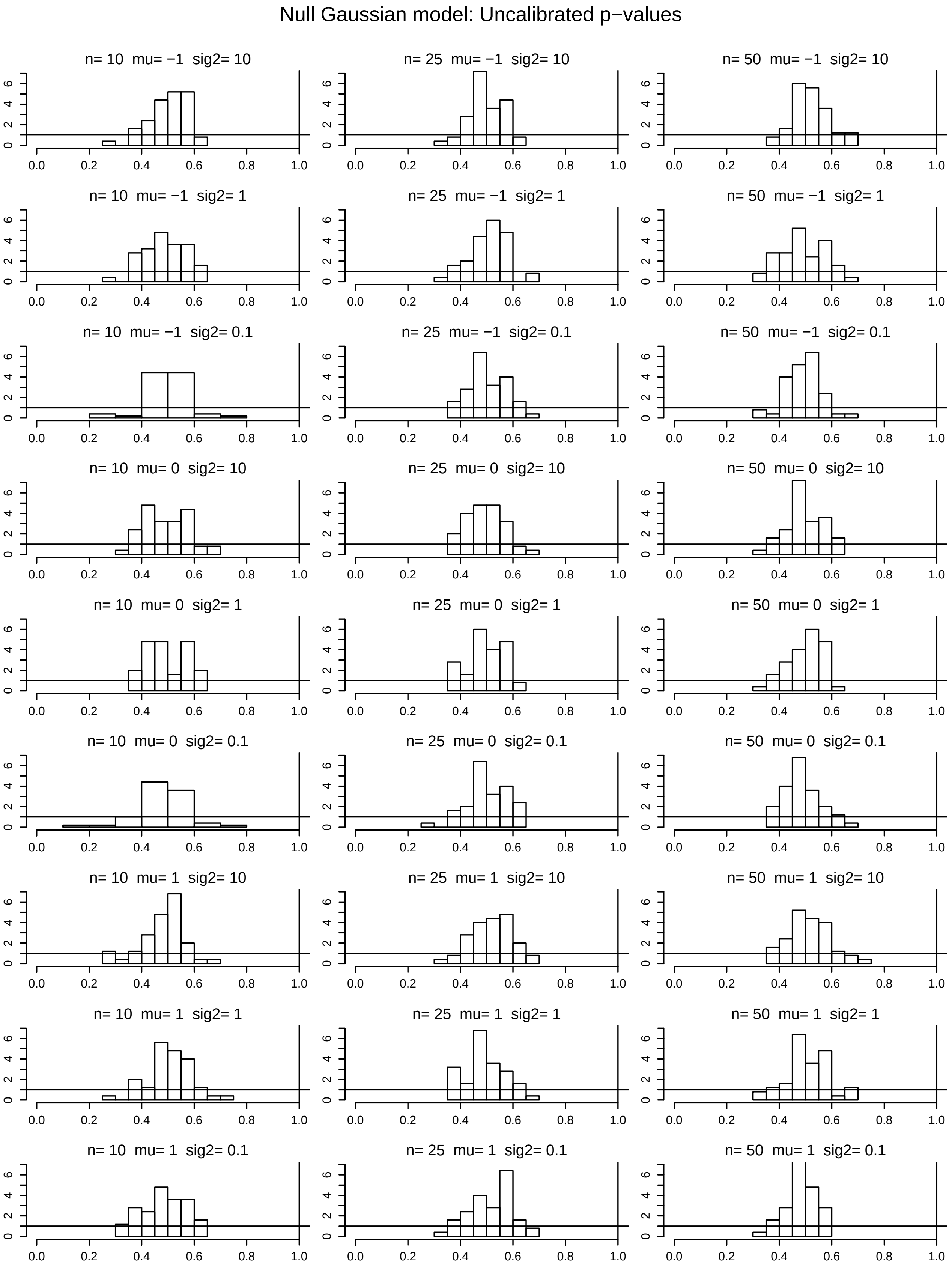


# Calibrated p-values

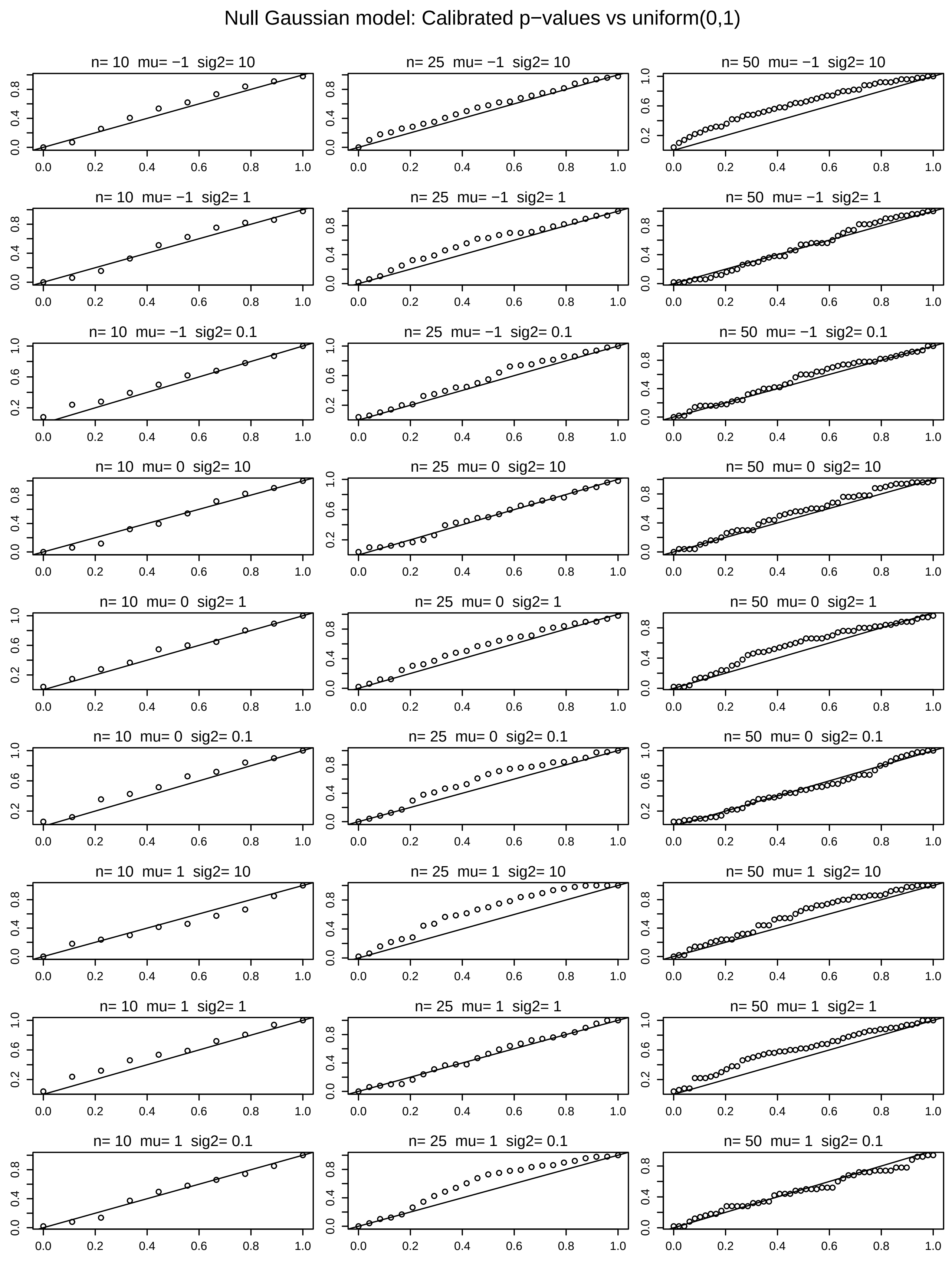
Null Poisson model: Calibrated p−values vs uniform(0,1)



# Uncalibrated p-values



# Calibrated p-values



## R Implementation

Here's some R code (available at the course web site) to **implement** our method for **calibrating** the **log score** scale in a **one-sample Poisson** setting, applied first to the **length of stay** data from part 2b and then to a **simulated data set** that was **not generated by the Poisson model**.

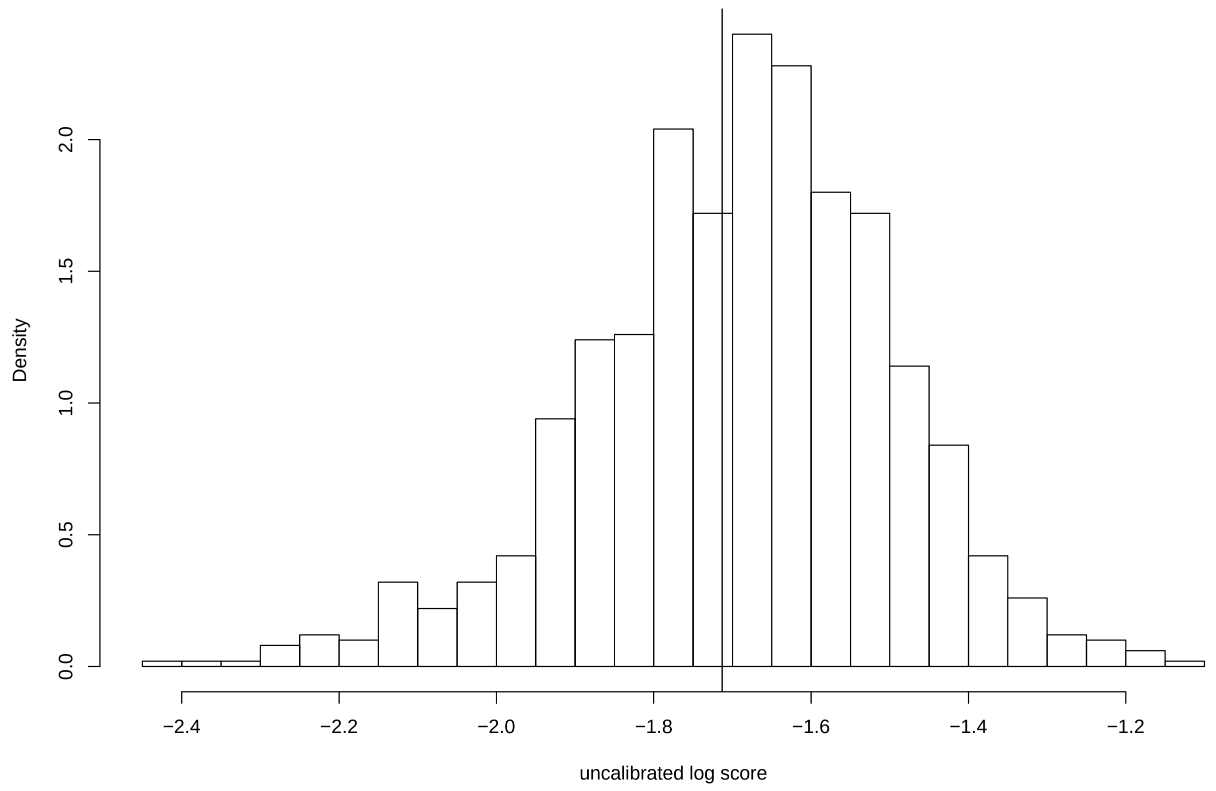
```
> print( y <- c( 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 6 ) )
 [1] 0 1 1 1 1 1 2 2 2 2 3 3 4 6

> print( epsilon <- 0.001 )
[1] 0.001

> ln.poisson.gamma <- function( y, alpha, beta ) {
+
+   lgamma( alpha + y ) + alpha * log( beta /
+     ( beta + 1 ) ) + y * log( 1 / ( beta + 1 ) ) -
+     lgamma( alpha ) - lgamma( y + 1 )
+
+ }

> step1 <- function( y, epsilon ) {
+
+   n <- length( y )
+
+   s <- sum( y )
+
+   als <- mean( ln.poisson.gamma( y, epsilon + s,
+     epsilon + n ) )
+
+   return( c( n, s, als ) )
+
+ }

> print( step1.result <- step1( y, epsilon ) )
[1] 14.00000 29.00000 -1.71309
```

So the **actual log score** for the LoS data set is $-1.71$, but is this **unusually small if the data really were Poisson**?

## R Implementation (continued)

```
> step2 <- function( n, s, epsilon, als, m1 ) {
+
+   lambda <- rgamma( m1, epsilon + s, epsilon + n )
+
+   ls <- rep( 0, m1 )
+
+   for ( i in 1:m1 ) {
+
+     y.star <- rpois( n, lambda[ i ] )
+
+     s.star <- sum( y.star )
+
+     ls[ i ] <- mean( ln.poisson.gamma( y.star,
+       epsilon + s.star, epsilon + n ) )
+
+   }
+
+   uata <- sum( ls <= als ) / m1
+
+   write( ls, "ls.out" )
+
+   return( uata )
+
+ }

> m1 <- 1000
>
> print( step2.result <- step2( step1.result[ 1 ],
+   step1.result[ 2 ], epsilon, step1.result[ 3 ], m1 ) )
[1] 0.418

> v.ls <- scan( "ls.out" )
Read 1000 items
>
> hist( v.ls, nclass = 20, probability = T,
+   main = '', xlab = 'uncalibrated log score' )
>
> abline( v = step1.result[ 3 ] )
```

## R Implementation (continued)



The **actual log score** doesn't look at all **unusual** in this plot, but recall from the discussion above that it **may not yet be properly calibrated**.

```
> step3 <- function( y, epsilon, m1, m2 ) {
+
+   step1.result <- step1( y, epsilon )
+
+   n <- step1.result[ 1 ]
+
+   s.actual <- step1.result[ 2 ]
+
+   uata <- step2( step1.result[ 1 ], step1.result[ 2 ],
+    epsilon, step1.result[ 3 ], m1 )
+
+   v.p <- rep( 0, m2 )
```

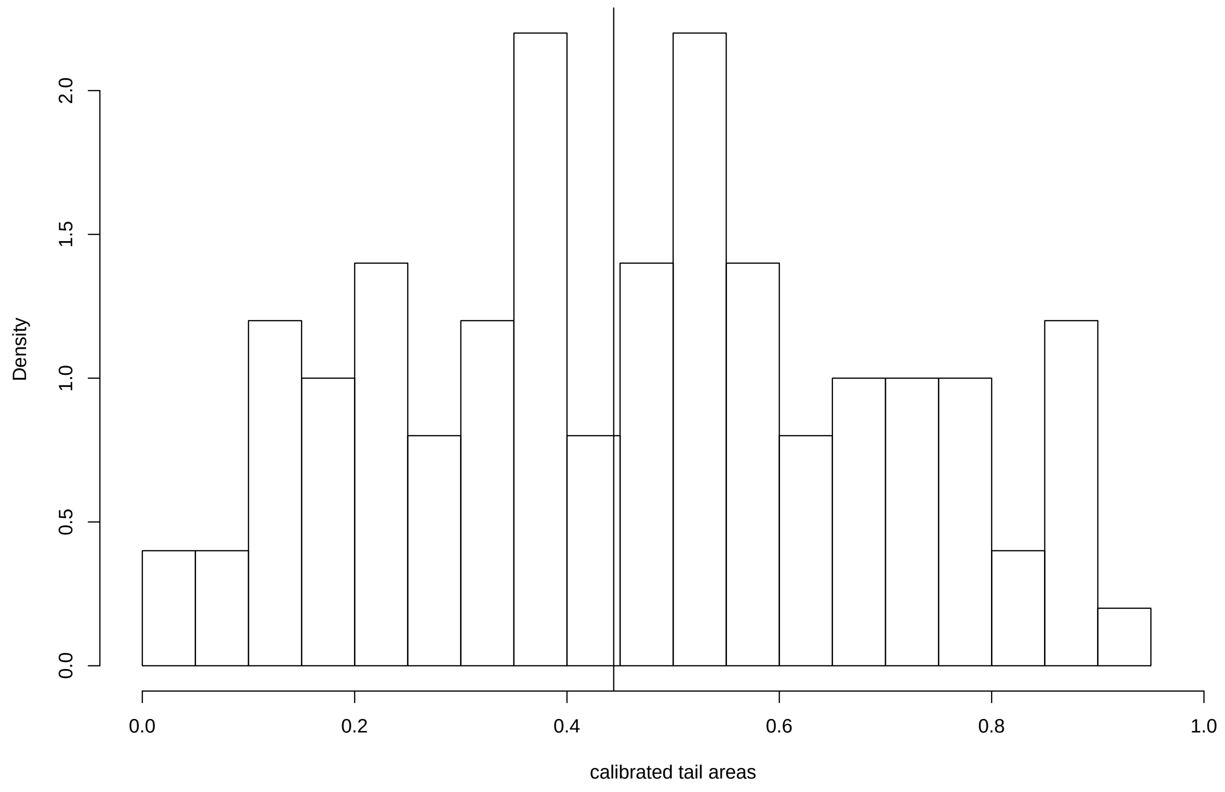
## R Implementation (continued)

```
+   for ( j in 1:m2 ) {
+
+     lambda.star <- rgamma( 1, epsilon + s.actual,
+      epsilon + n )
+
+     y.sim <- rpois( n, lambda.star )
+
+     step1.result <- step1( y.sim, epsilon )
+
+     v.p[ j ] <- step2( step1.result[ 1 ],
+       step1.result[ 2 ], epsilon, step1.result[ 3 ], m1 )
+
+   }
+
+   aata <- sum( v.p <= uata ) / m2
+
+   write( v.p, "v.p.out" )
+
+   return( aata )
+
+ }

> m2 <- 100
>
> print( step3.result <- step3( y, epsilon, m1, m2 ) )
[1] 0.4
```

Here the **recalibration** has **not had much effect**, but (as the plots above showed) **this will not always be the case**.

## R Implementation (continued)

```
> v.p <- scan( "v.p.out" )
Read 100 items
>
> hist( v.p, nclass = 20, probability = T, xlim = c( 0, 1 ),
+   main = '', xlab = 'calibrated tail areas' )
>
> abline( v = step2.result )
```



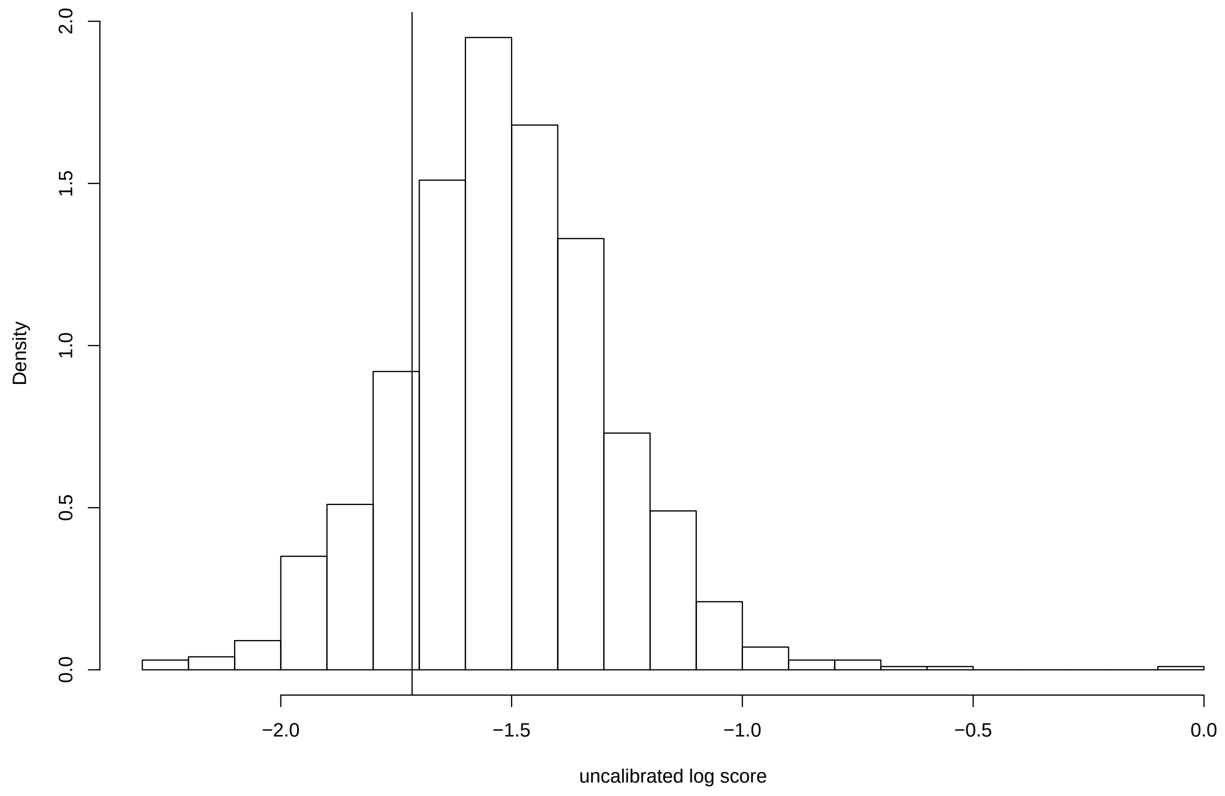
For a **second example** let's look at a **data set** generated as a **lognormal mixture of Poissons** with a **substantial VTMR**.

```
> n <- 10
>
> e <- rnorm( n, 0.0, 0.5 )
>
> mu <- 0
>
> lambda <- rep( 0, n )
```

## R Implementation (continued)

```
> y <- rep( 0, n )

> for ( i in 1:n ) {
+
+   lambda[ i ] <- exp( mu + e[ i ] )
+
+   y[ i ] <- rpois( 1, lambda[ i ] )
+
+ }

> print( y <- sort( y ) )

[1] 0 0 0 1 1 1 2 3 4 4

> var( y ) / mean( y )

[1] 1.555556

> print( step1.result <- step1( y, epsilon ) )

[1] 10.000000 16.000000 -1.715601

> print( step2.result <- step2( step1.result[ 1 ],
+   step1.result[ 2 ], epsilon, step1.result[ 3 ], m1 ) )

[1] 0.178

> v.ls <- scan( "ls.out" )

> hist( v.ls, nclass = 20, probability = T,
+   main = '', xlab = 'uncalibrated log score' )

> abline( v = step1.result[ 3 ] )
```

## R Implementation (continued)

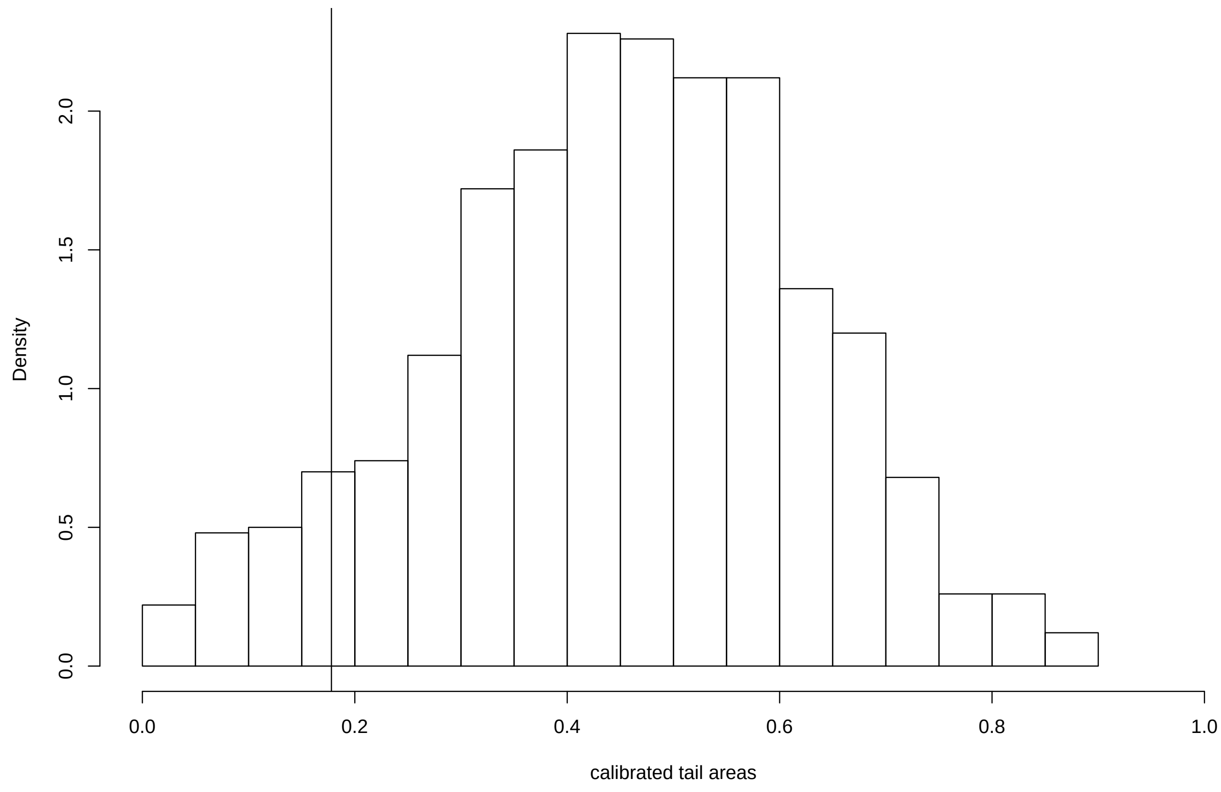


```
> m2 <- 1000

> print( step3.result <- step3( y, epsilon, m1, m2 ) )

[1] 0.099
```

So here's an example where the **uncalibrated tail area** is **about twice as big as it should be**.

```
> v.p <- scan( "v.p.out" )

> hist( v.p, nclass = 20, probability = T, xlim = c( 0, 1 ),
+   main = '', xlab = 'calibrated tail areas' )

> abline( v = step2.result )
```

## R Implementation (continued)



The **true calibrated tail-area distribution** is **far from uniform**, so 0.178 is actually **substantially farther out in the true tail than it seems**.

## Conclusions

- {**Exchangeability judgments** plus **nonparametric** (BNP) modeling} = **Bayesian model specification** in many problems.

- **BNP** is one way to avoid the **dilemma** posed by **Cromwell's Rule** in Bayesian model specification; **three-way cross-validation** (3CV) is another.

- Model choice is really a **decision problem** and should be approached via **MEU**, with a utility structure that's **sensitive to the real-world context**.

- When the goal is to make an **accurate scientific summary** of what's known about something, the **predictive log score** has a **sound generic utility basis** and can yield **stable** and **accurate** model specification decisions.

- $DIC$ can be thought of as a fast approximation to the **leave-one-out predictive log score** ($LS_{CV}$), but $DIC$ can behave **unstably** as a function of **parameterization**.

- The **full-sample log score** ($LS_{FS}$) is $n$ times **faster** than naive implementations of $LS_{CV}$, has better **small-sample model discrimination power** than either $LS_{CV}$ or $DIC$, and **has better asymptotic behavior** than $LS_{CV}$.

- **Generic Bayes factors** are **highly unstable** when context suggests **diffuse prior information**; many methods for fixing this have been proposed, most of which seem to require an **appeal to ad-hockery** which is **absent** from the $LS_{FS}$ approach.

- The basic Gelman et al. (1996) method of posterior predictive model-checking is **badly calibrated**: when it gives you a tail area of, e.g., **0.4**, the calibrated equivalent may well be **0.04** or even **0.004**.

- We have modified an **approach** suggested by Robins et al. (2000) to help answer the question "Could the data have arisen from model $M$?" in a **well-calibrated** way.

# References

Bernardo JM, Smith AFM (1994). *Bayesian Theory.* New York: Wiley.

Dey D, Mueller P, Sinha D (1998). *Practical Nonparametric and Semiparametric Bayesian Statistics*. New York: Springer Verlag (Lecture Notes in Statistics, Volume 133).

de Finetti B (1930). Funzione caratteristica de un fenomeno aleatorio. *Mem. Acad. Naz. Lincei*, **4**, 86–133.

de Finetti B (1937). La prévision: ses lois logiques, ses sources subjectives. *Ann. Inst. H. Poincaré*, **7**, 1–68.

de Finetti B (1938). Sur la condition d'equivalence partielle. *Actualités Scientifiques et Industrielles*, **739**.

de Finetti B (1990). *Theory of Probability*. New York: Wiley Classics Library.

Draper D (1995). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society Series B*, **57**, 45–97.

Draper D (1996). Utility, sensitivity analysis, and cross-validation in Bayesian model-checking. *Statistica Sinica*, **6**, 760–767 (discussion of "Posterior predictive assessment of model fitness via realized discrepancies," by A Gelman, X-L Meng, and H Stern).

Draper D, Fouskakis D (2000). A case study of stochastic optimization in health policy: problem formulation and preliminary results. *Journal of Global Optimization*, **18**, 399–416.

Draper D, Fouskakis D (2004). Stochastic optimization methods for cost-effective quality assessment in health. Submitted.

Draper D, Krnjajić M (2005). Three-way cross-validation for well-calibrated model exploration. In preparation.

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

Fouskakis D, Draper D (2002). Stochastic optimization: a review. *International Statistical Review*, **70**, 315–349.

## References (continued)

Geisser S, Eddy WF (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, **74**, 153–160.

Gelfand AE, Dey DK, Chang H (1992). Model determination using predictive distributions, with implementation via sampling-based methods (with discussion). In *Bayesian Statistics 4* (Bernardo JM, Berger JO, Dawid AP, Smith AFM, editors), Oxford: Oxford University Press, 147–167.

Gelman A, Meng X-L, Stern H (1996). Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica Sinica*, **6**, 733–760.

Good IJ (1950). *Probability and the Weighing of Evidence.* London: Griffin.

Hendriksen C, Lund E, Stromgard E (1984). Consequences of assessment and intervention among elderly people: a three year randomised controlled trial. *British Medical Journal*, **289**, 1522–1524.

Jeffreys H (1939). *Theory of Probability*. Oxford: Oxford University Press.

Kass RE, Raftery AE (1995). Bayes factors. *Journal of the American Statistical Association*, **90**, 773–795.

Key JT, Pericchi LR, Smith AFM (1998). Bayesian model choice: what and why? (with discussion). In *Bayesian Statistics 6*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (editors). Oxford University Press, 343–370.

O'Hagan A, Forster J (2004). *Bayesian Inference*, second edition. London: Arnold.

Pericchi L (2004). Model selection and hypothesis testing based on objective probabilities and Bayes factors. Manuscript.

Robins JM, van der Vaart A, Ventura V (2000). Asymptotic distribution of $P$ values in composite null models. *Journal of the American Statistical Association*, **95**, 1143–1156.

Spiegelhalter DJ, Best NG, Carlin BR, van der Linde A (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society Series B*, **64**, 583–616.

Walker S, Damien P, Lenk P (2004). On priors with a Kullback-Leibler property. *Journal of the American Statistical Association*, **99**, 404–408.