統計モデリング入門 筑波大 (大塚) 集中講義 [09]

階層ベイズモデル

久保拓弥 kubo@ees.hokudai.ac.jp, @KuboBook

筑波大集中講義 http://goo.gl/HvRhXn

2015–03–01

ファイル更新時刻: 2015–03–01 14:18

この時間に説明したいこと

1. **GLMM と階層ベイズモデル**
   GLMM をベイズモデルと考えると……
2. **階層ベイズモデルの推定**
   ソフトウェア WinBUGS を使ってみる
3. **複数ランダム効果の階層ベイズモデル**
   個体差 + グループ差，など

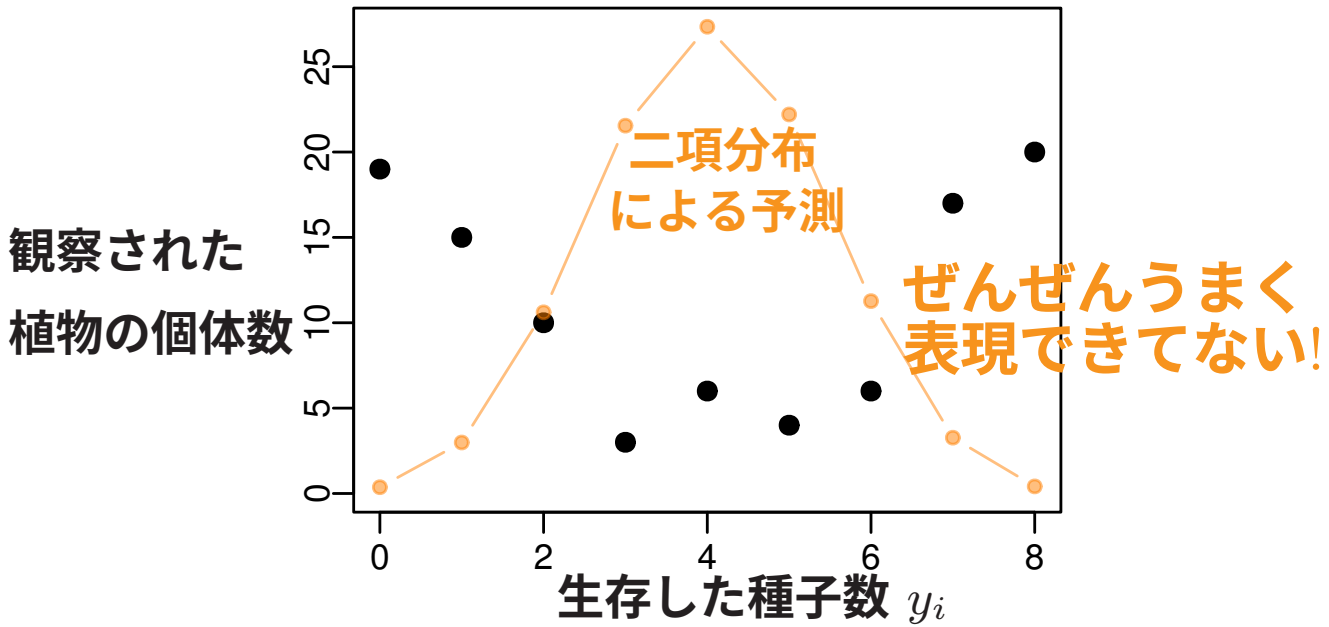
1. GLMM と階層ベイズモデル

GLMM をベイズモデルと考えると……

階層ベイズモデルとなる

二項分布では説明できない観測データ!

100 個体の植物の合計 800 種子中 403 個の生存が見られたので，平均生存確率は 0.50 と推定されたが……

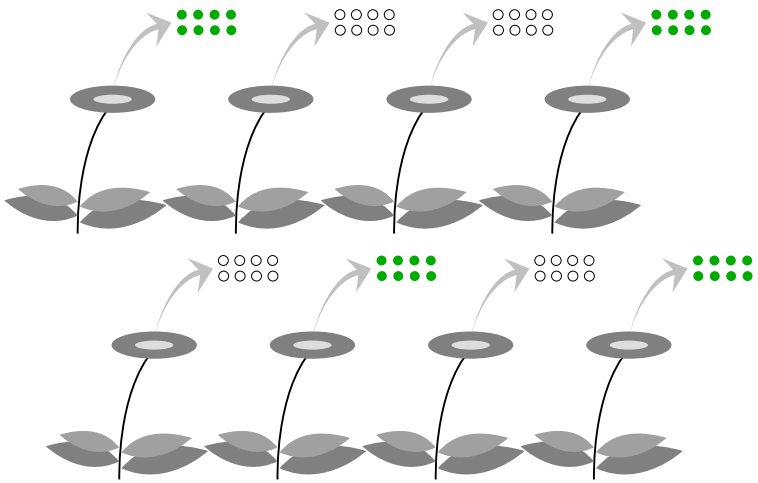


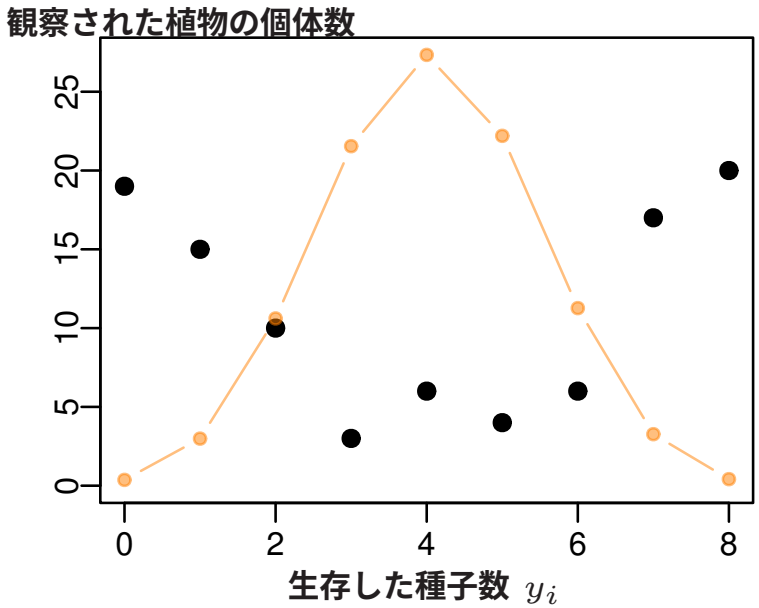
さっきの例題と同じようなデータなのに?

(「統計モデリング入門」第 10 章の最初の例題)

個体差 → 過分散 (overdispersion)

極端な過分散の例





- 種子全体の平均生存確率は 0.5 ぐらいかもしれないが……
- 植物個体ごとに種子の生存確率が異なる: 「個体差」
- 「個体差」があると overdispersion が生じる
- 「個体差」の原因は観測できない・観測されていない

モデリングやりなおし: 個体差を考慮する

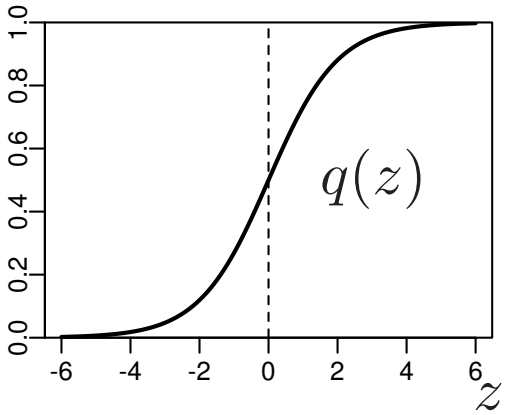
- 生存確率を推定するために 二項分布という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率は二項分布

$$p(y_i \mid q_i) = \binom{N_i}{y_i} q_i^{y_i} (1 - q_i)^{N_i - y_i},$$

- ここで仮定していること
  - 個体差があるので個体ごとに生存確率 q_i が異なる

GLM わざ: ロジスティック関数で表現する生存確率

- **生存確率** $q_i = q(z_i)$ **をロジスティック関数** $q(z) = 1/\{1 + \exp(-z)\}$ **で表現**



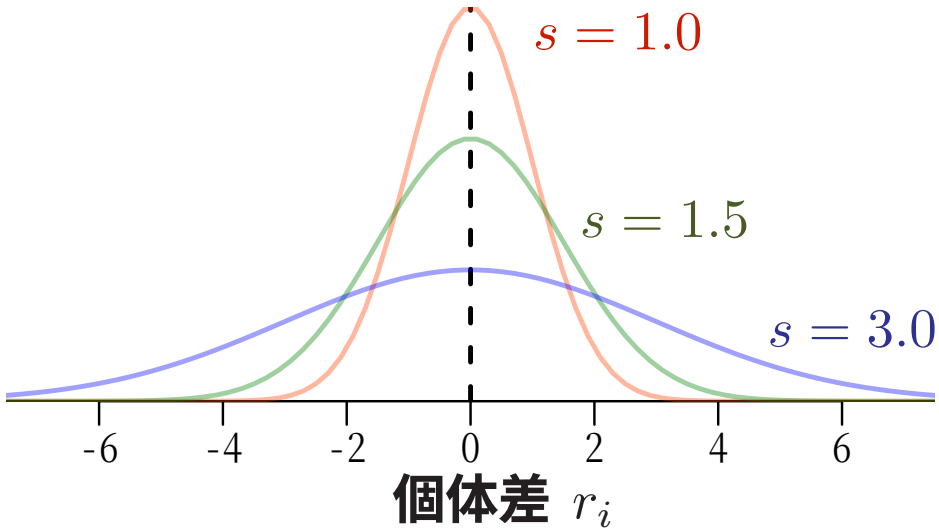
- **線形予測子** $z_i = a + r_i$ **とする**
  - **パラメーター** a**: 全体の平均**
  - **パラメーター** r_i**: 個体** i **の個体差 (ずれ)**

個々の個体差 r_i を最尤推定するのはまずい

- 100 個体の生存確率を推定するために**パラメーター 101 個** (a と $\{r_1, r_2, \cdots, r_{100}\}$) を推定すると……
- 個体ごとに生存数 / 種子数を計算していることと同じ! (「データのよみあげ」と同じ)

そこで，次のように考えてみる

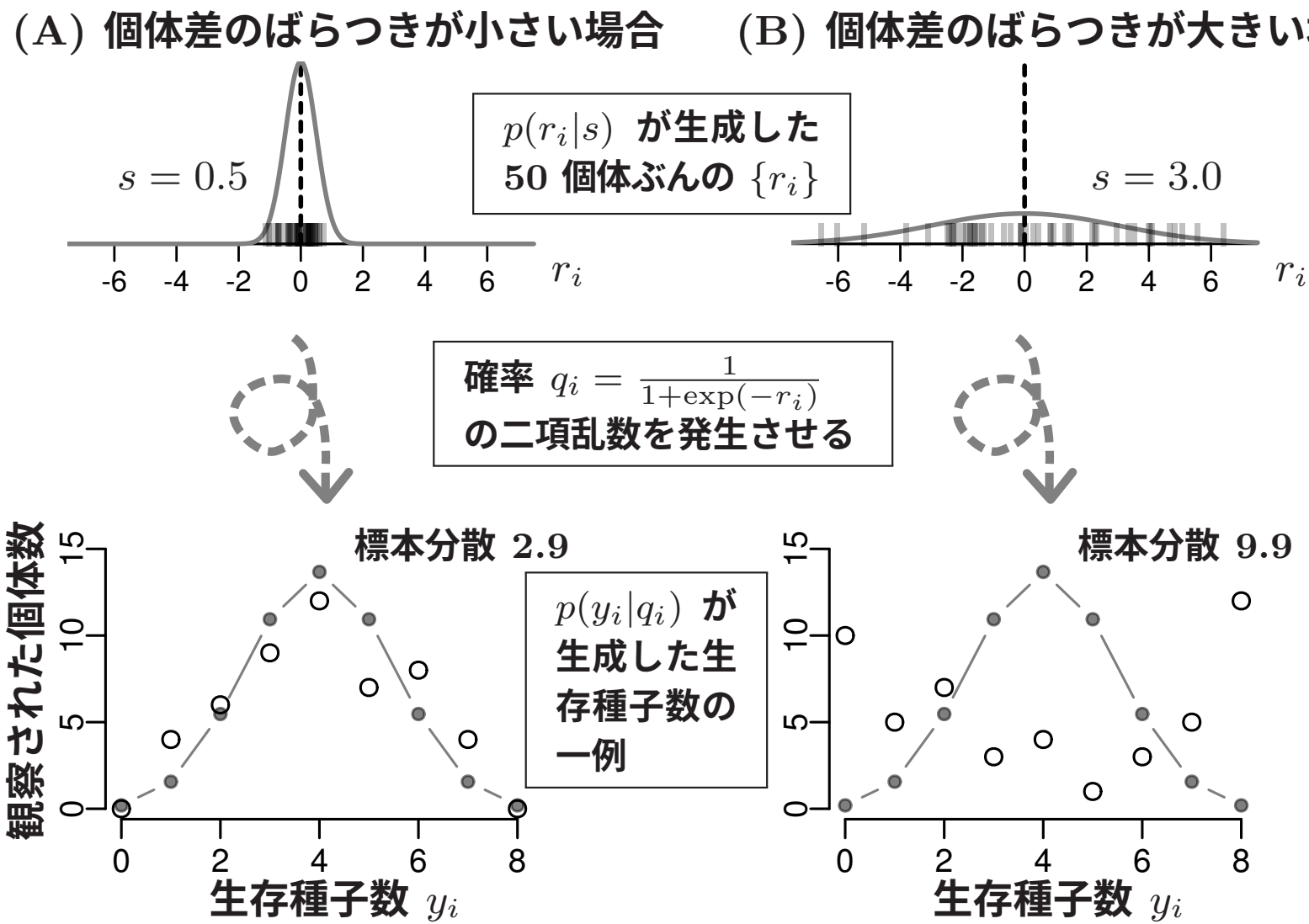
$\{r_i\}$ のばらつきは正規分布だと考えてみる



$$p(r_i \mid s) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{r_i^2}{2s^2}\right)$$

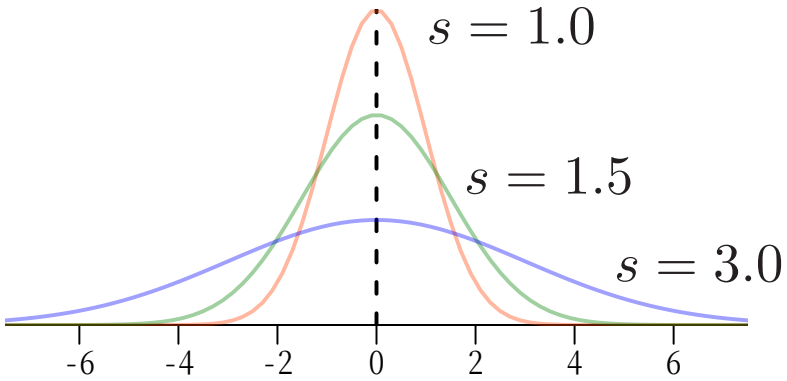
この確率密度 $p(r_i \mid s)$ は r_i の「出現しやすさ」をあらわしていると解釈すればよいでしょう．r_i がゼロにちかい個体はわりと「ありがち」で，r_i の絶対値が大きな個体は相対的に「あまりいない」．

ひとつの例示: 個体差 r_i の分布と過分散の関係



これは r_i の事前分布の指定，ということ

前回の授業で $\{r_i\}$ は正規分布にしたがうと仮定したが
ベイズ統計モデリングでは 「100 個の r_i たちに
共通する事前分布として正規分布 を指定した」
ということになる

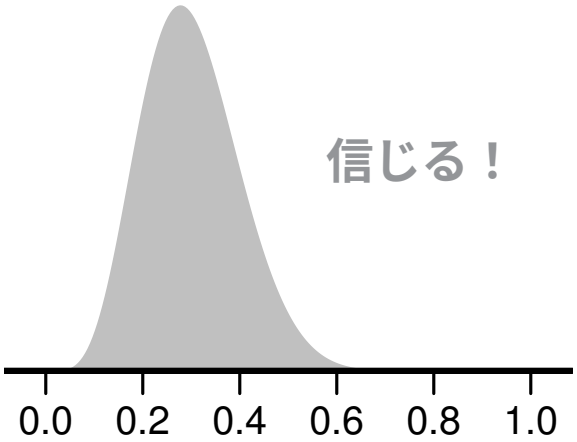


$$p(r_i \mid s) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{r_i^2}{2s^2}\right)$$

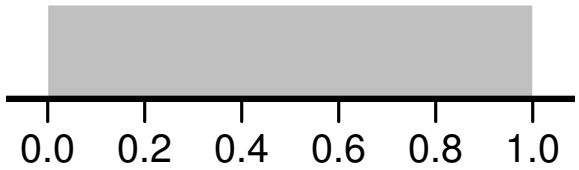
ベイズ統計モデルでよく使われる三種類の事前分布

たいていのベイズ統計モデルでは，ひとつのモデルの中で複数の種類の事前分布を混ぜて使用する．

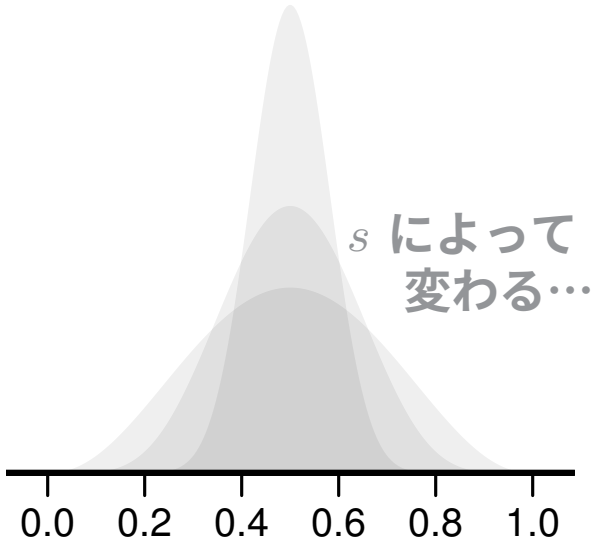
(A) **主観的な事前分布**
(できれば使いたくない!)



(B) **無情報事前分布**



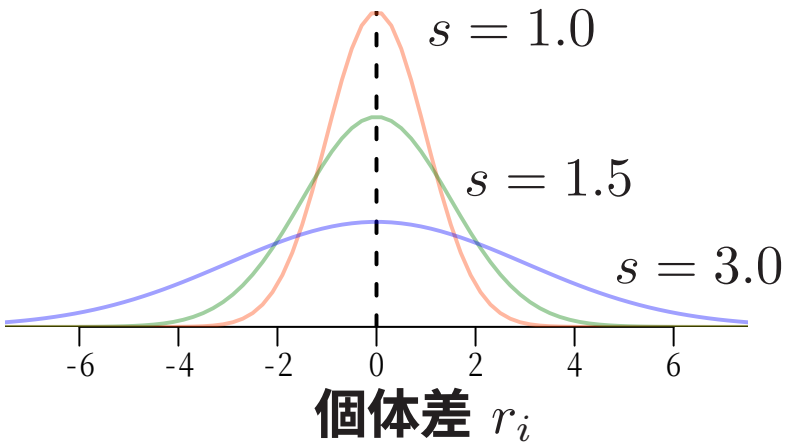
(C) **階層事前分布**



r_i の事前分布として階層事前分布を指定する

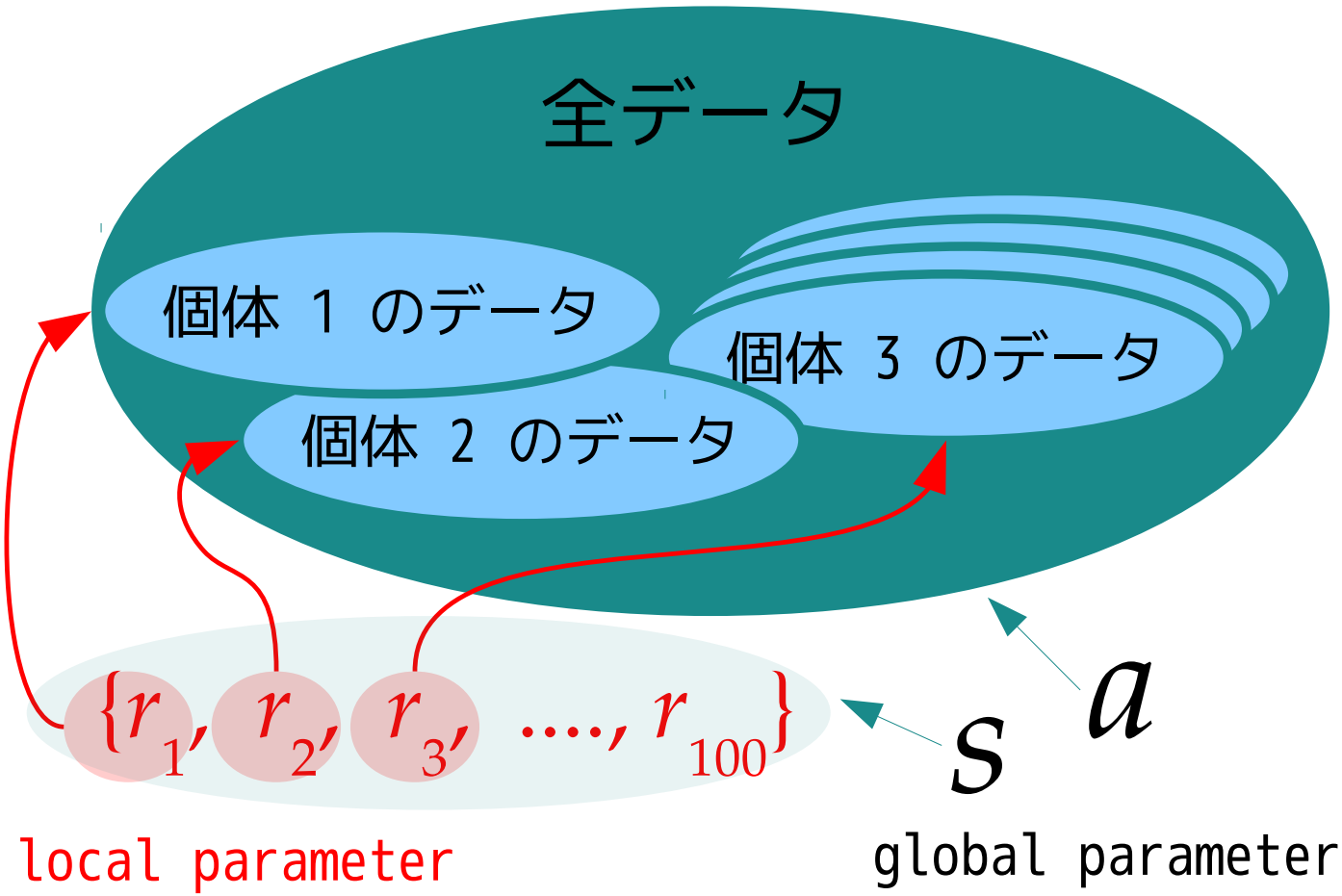
階層事前分布の利点

「データにあわせて」事前分布が変形!



$$p(r_i \mid s) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{r_i^2}{2s^2}\right)$$

統計モデルの大域的・局所的なパラメーター



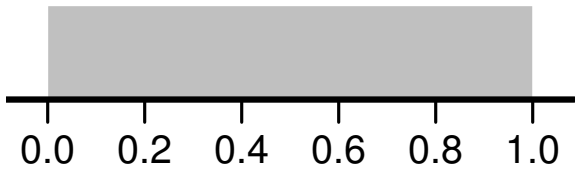
データのどの部分を説明しているのか?

パラメーターごとに適切な事前分布を選ぶ

(B) 無情報事前分布

$a,\ s$

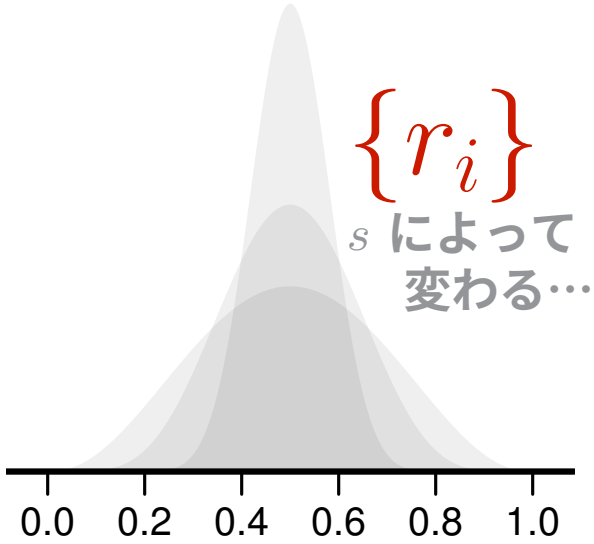
わからない？



(C) 階層事前分布

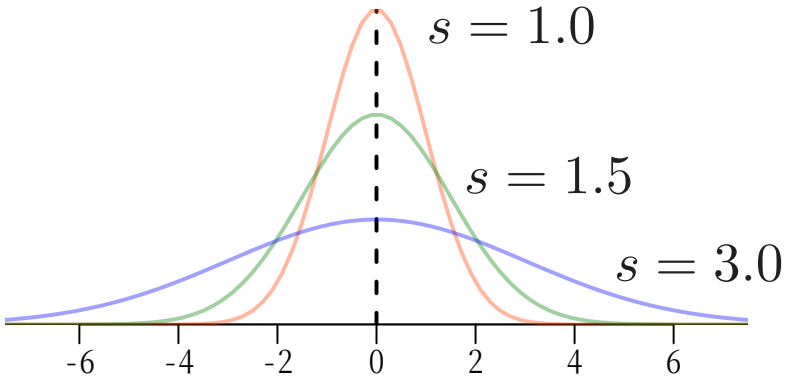
$\{r_i\}$

s によって変わる…



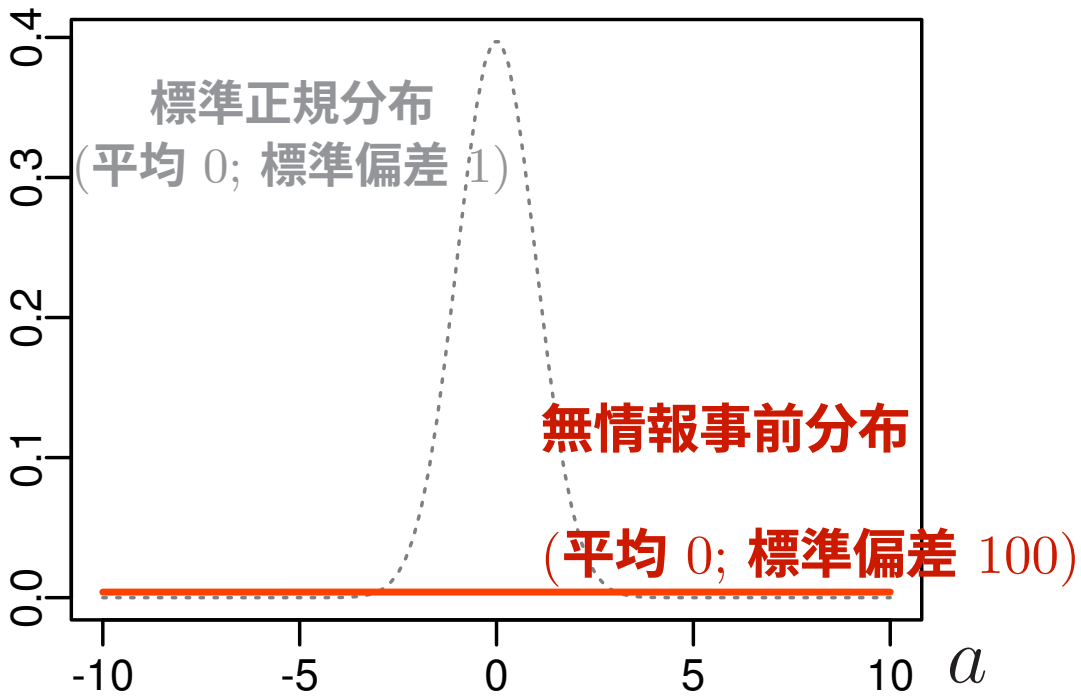
パラメーターの種類	説明する範囲	事前分布
全体に共通する平均・ばらつき	大域的	無情報事前分布
個体・グループごとのずれ	局所的	階層事前分布

個体差 $\{r_i\}$ のばらつき s の無情報事前分布



- s はどのような値をとってもかまわない
- そこで s の事前分布は **無情報事前分布** (non-informative prior) とする
- たとえば一様分布，ここでは $0 < s < 10^4$ の一様分布としてみる

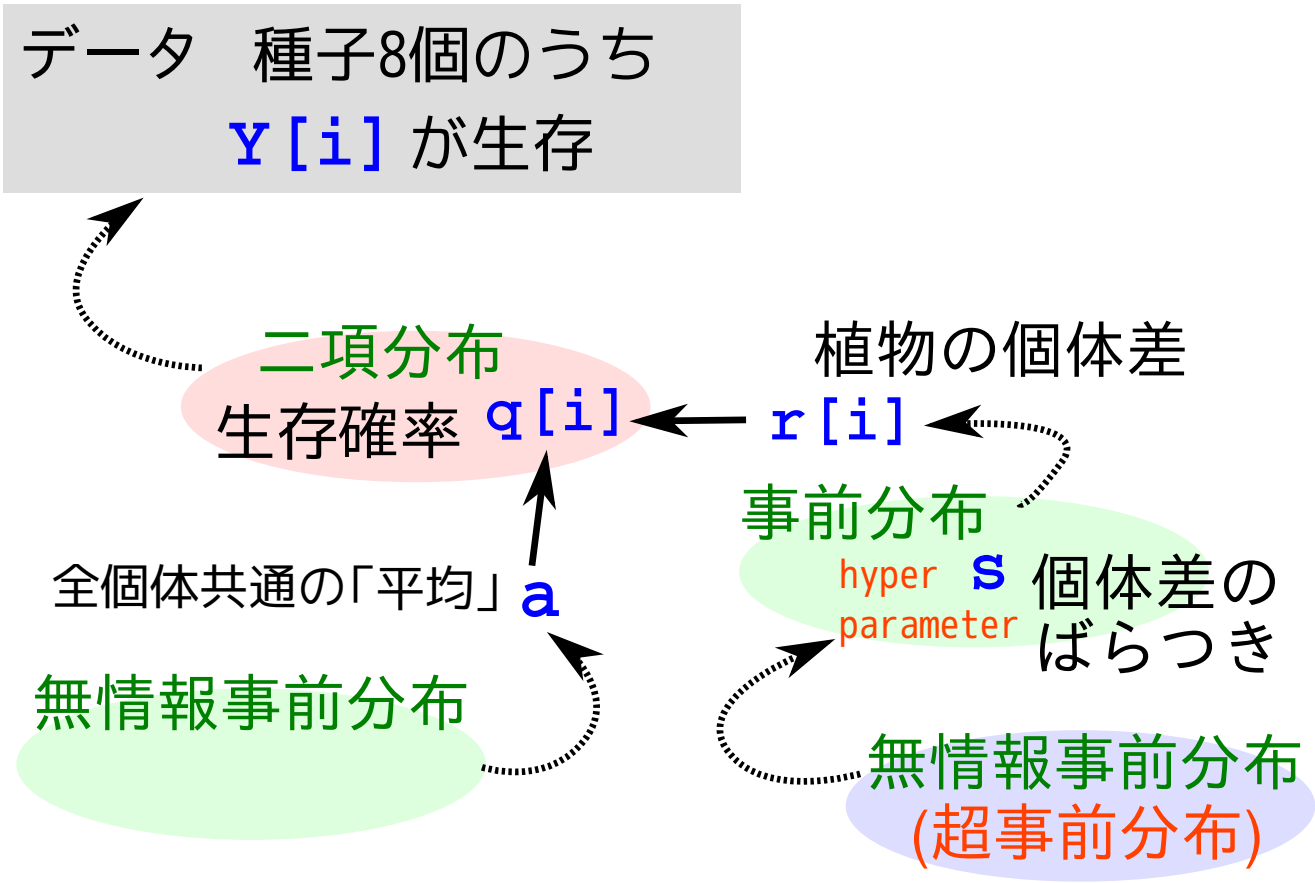
全個体の「切片」 a の無情報事前分布



「生存確率の (logit) 平均 a は何でもよい」と表現している

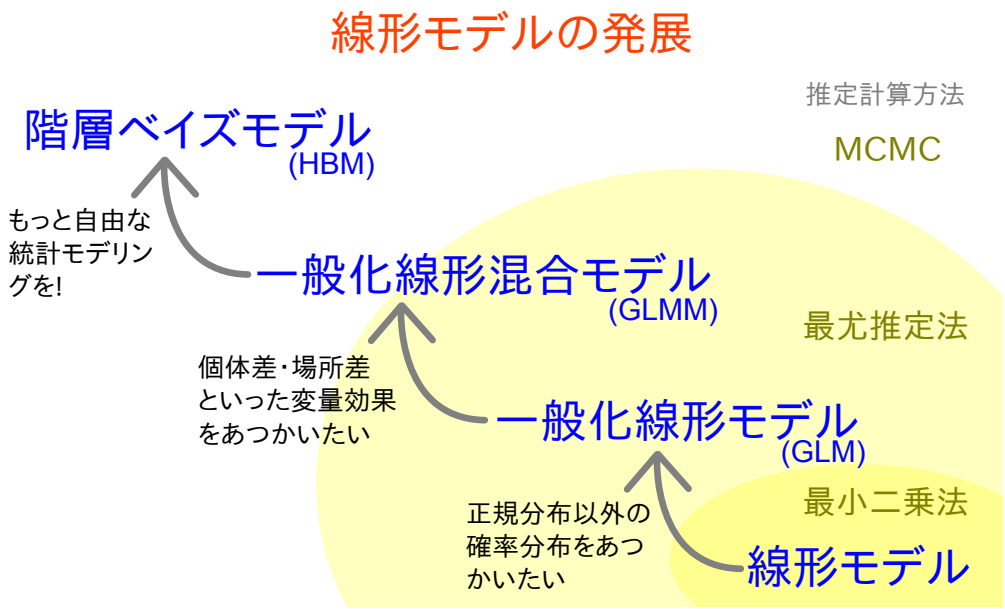
階層ベイズモデル: 事前分布の階層性

超事前分布 → 事前分布という階層があるから



矢印は手順ではなく，依存関係をあらわしている

階層ベイズモデルと GLMM の関係



一般化線形混合モデル (Generalized Linear Mixed Model) **は階層ベイズモデルのひとつ**

- global parameter **は** fixed effects
- local parameter **は** random effects

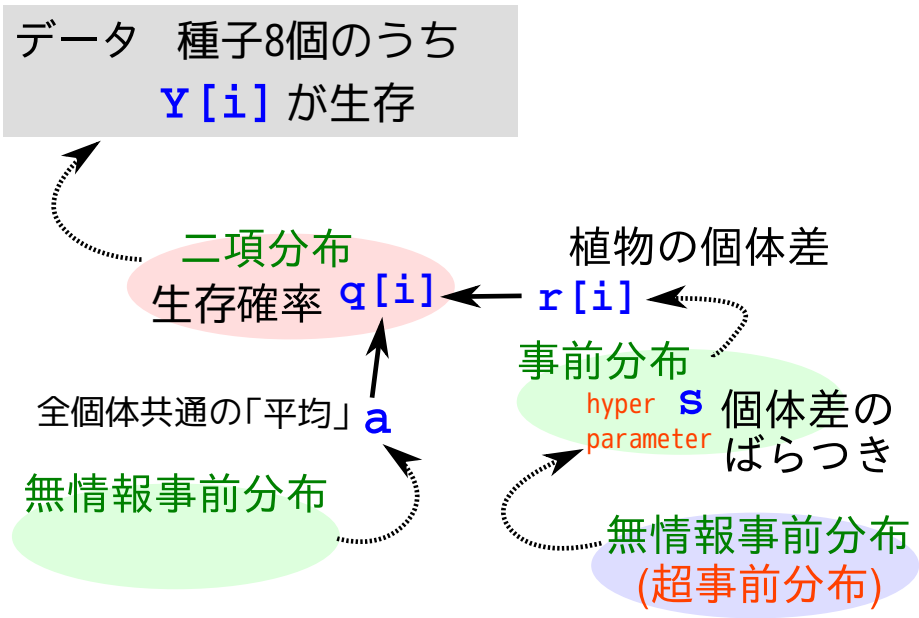
2. 階層ベイズモデルの推定

ソフトウェア WinBUGS を使ってみる

BUGS 言語で統計モデルを指定，R と連携する

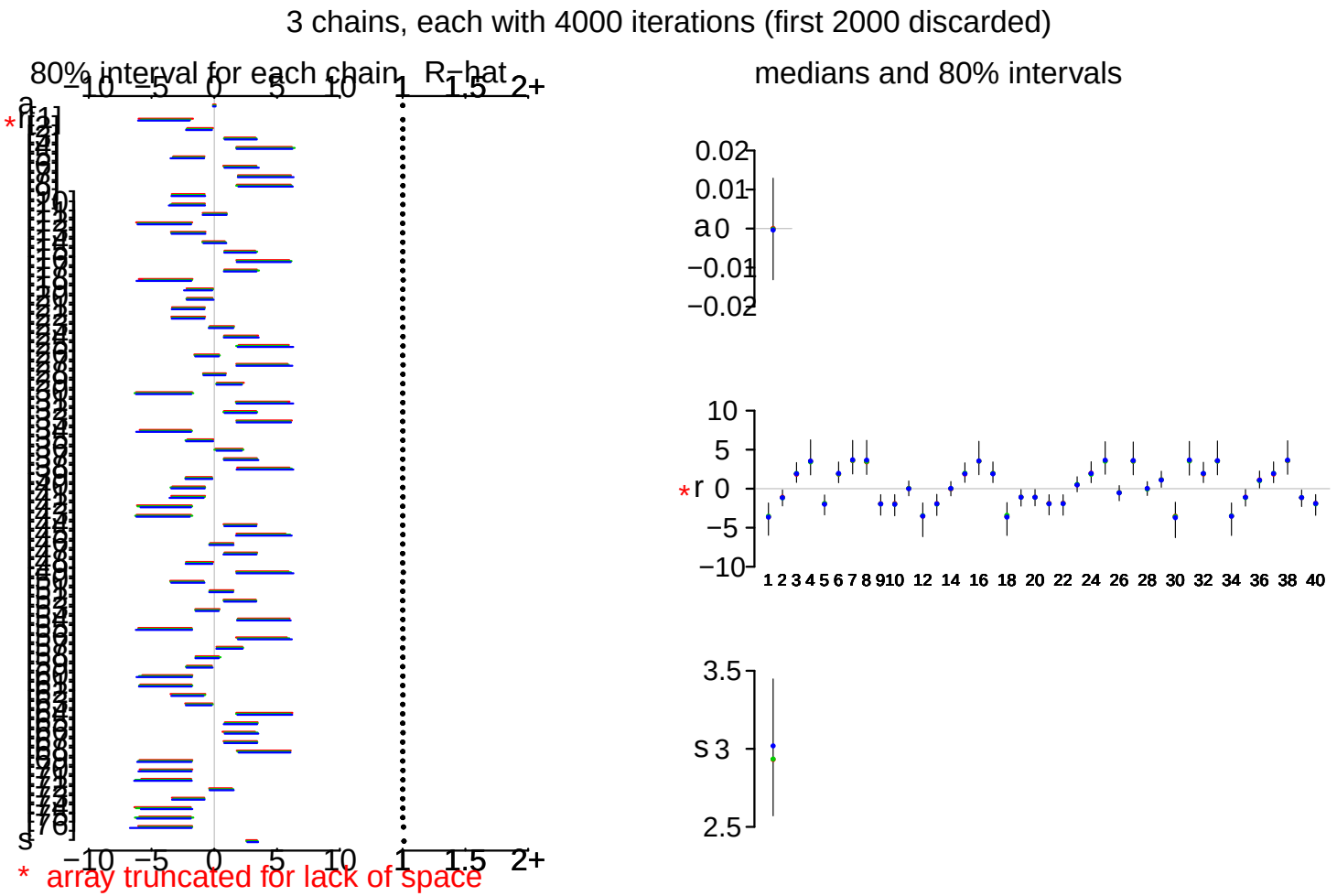
階層ベイズモデルを BUGS コードで記述する

```
model
{
  for (i in 1:N.data) {
    Y[i] ~ dbin(q[i], 8)
    logit(q[i]) <- a + r[i]
  }
  a ~ dnorm(0, 1.0E-4)
  for (i in 1:N.data) {
    r[i] ~ dnorm(0, tau)
  }
  tau <- 1 / (s * s)
  s ~ dunif(0, 1.0E+4)
}
```



JAGS で得られた事後分布サンプルの要約

```
> source("mcmc.list2bugs.R")                    # なんとなく便利なので…
> post.bugs <- mcmc.list2bugs(post.mcmc.list) # bugs クラスに変換
```

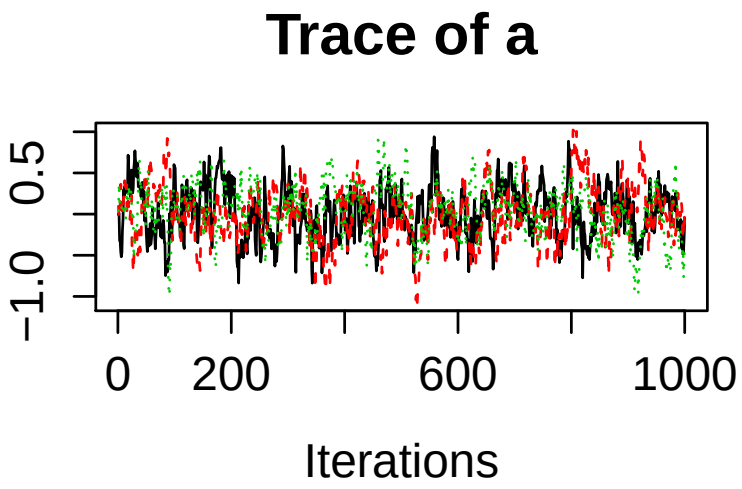


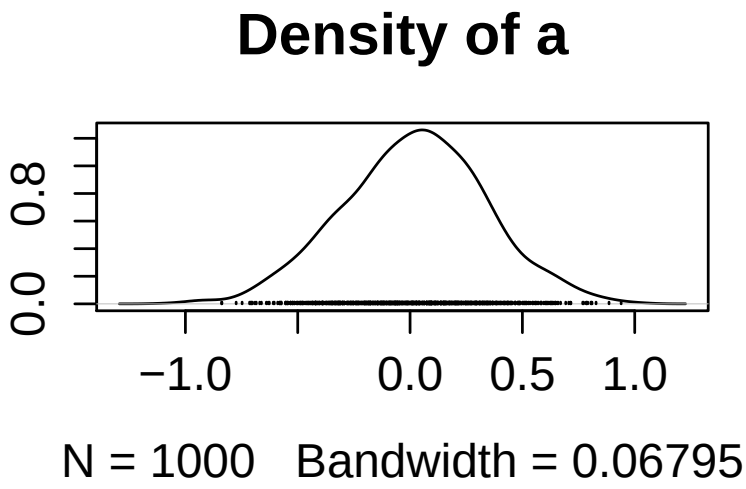
bugs オブジェクトの post.bugs を調べる

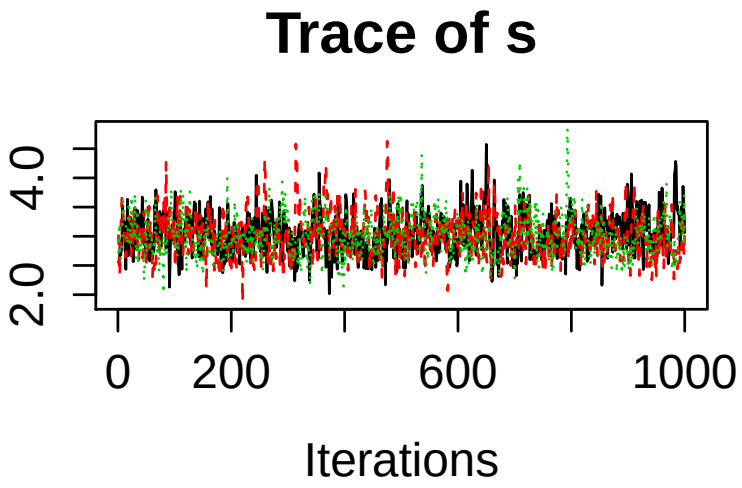
- print(post.bugs, digits.summary = 3)
- **事後分布の 95% 信頼区間などが表示される**

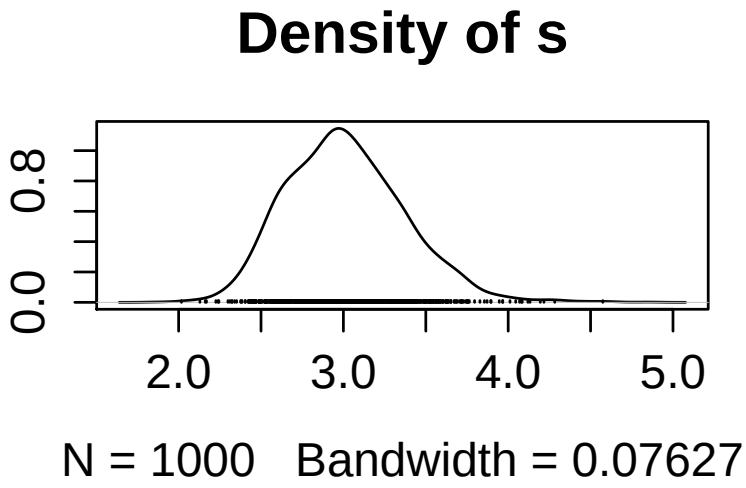
```
 3 chains, each with 4000 iterations (first 2000 discarded), n.thin = 2
 n.sims = 3000 iterations saved
         mean    sd   2.5%    25%    50%    75%  97.5%  Rhat n.eff
a       0.020 0.321 -0.618 -0.190  0.028  0.236  0.651 1.007   380
s       3.015 0.359  2.406  2.757  2.990  3.235  3.749 1.002  1200
r[1]   -3.778 1.713 -7.619 -4.763 -3.524 -2.568 -1.062 1.001  3000
r[2]   -1.147 0.885 -2.997 -1.700 -1.118 -0.531  0.464 1.001  3000
r[3]    2.014 1.074  0.203  1.282  1.923  2.648  4.410 1.001  3000
r[4]    3.765 1.722  0.998  2.533  3.558  4.840  7.592 1.001  3000
r[5]   -2.108 1.111 -4.480 -2.775 -2.047 -1.342 -0.164 1.001  2300
... (中略)
r[99]   2.054 1.103  0.184  1.270  1.996  2.716  4.414 1.001  3000
r[100] -3.828 1.766 -7.993 -4.829 -3.544 -2.588 -1.082 1.002  1100
```

各パラメーターの事後分布サンプルを R で調べる



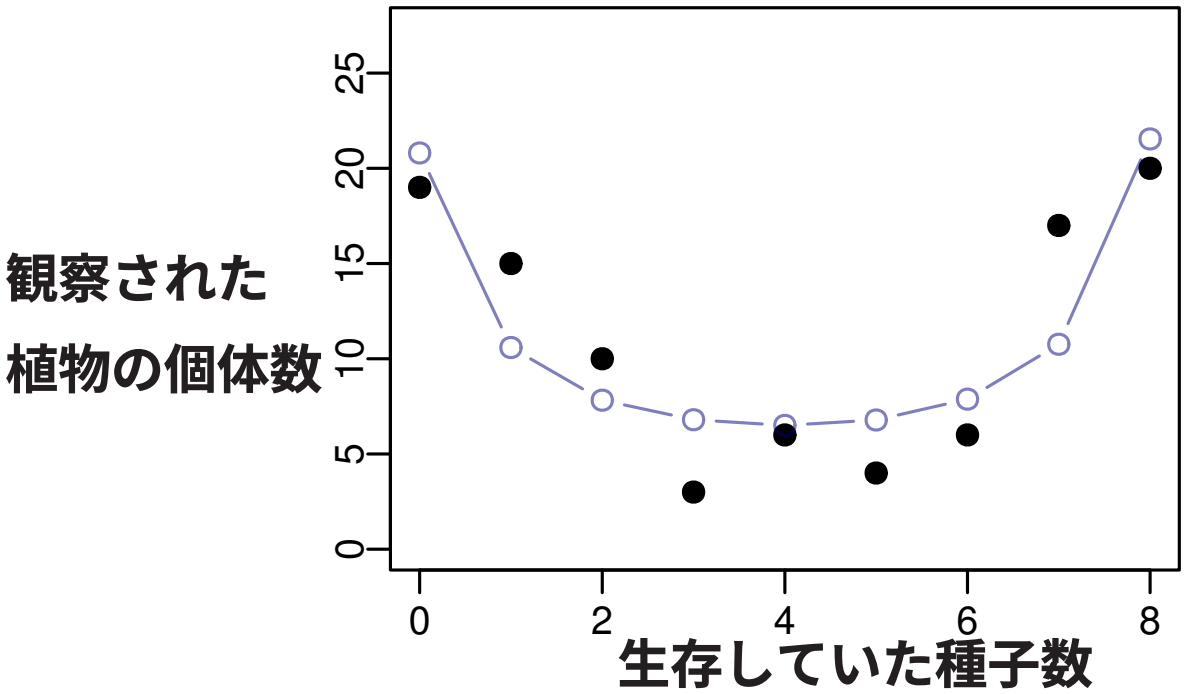






得られた事後分布サンプルを組みあわせて予測

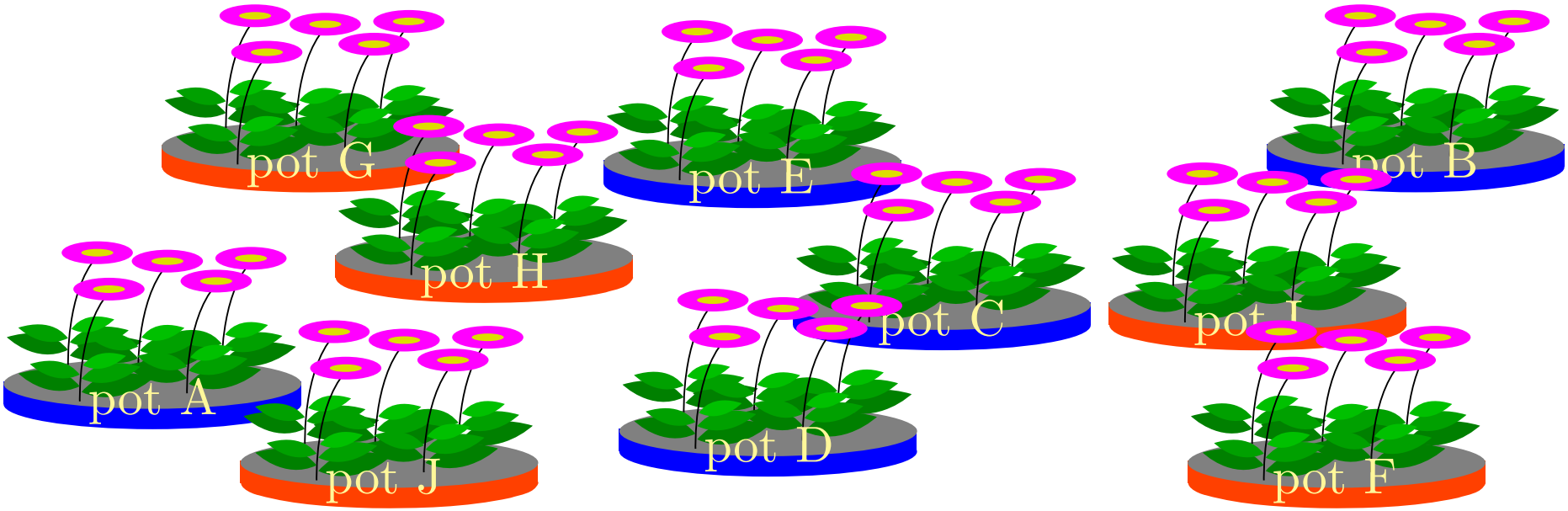
- `post.mcmc <- to.mcmc(post.bugs)`
- **これは `matrix` と同じようにあつかえるので，作図に便利**



3. 複数ランダム効果の階層ベイズモデル

個体差 + グループ差，など

架空植物の例題: またまた種子数データ



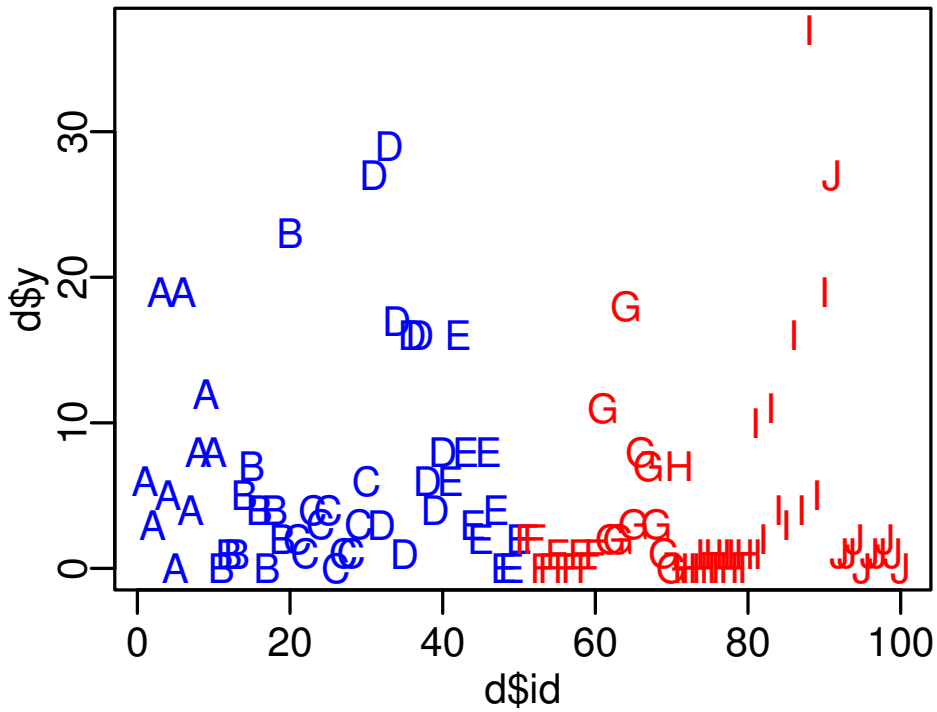
- 肥料をやったら個体ごとの種子数 y_i が増えるかどうかを調べたい
- 植木鉢 10 個，各鉢に 10 個体の架空植物 (合計 100 個体)
  - コントロール ($f_j = \mathbf{C}$) 5 鉢 (合計 50 個体)
  - 肥料をやる処理 ($f_j = \mathbf{T}$) 5 鉢 (合計 50 個体)

データはこのように格納されている

```
> d <- read.csv("d1.csv")
> head(d)
  id pot f  y
1  1   A C  6
2  2   A C  3
3  3   A C 19
4  4   A C  5
5  5   A C  0
6  6   A C 19
```

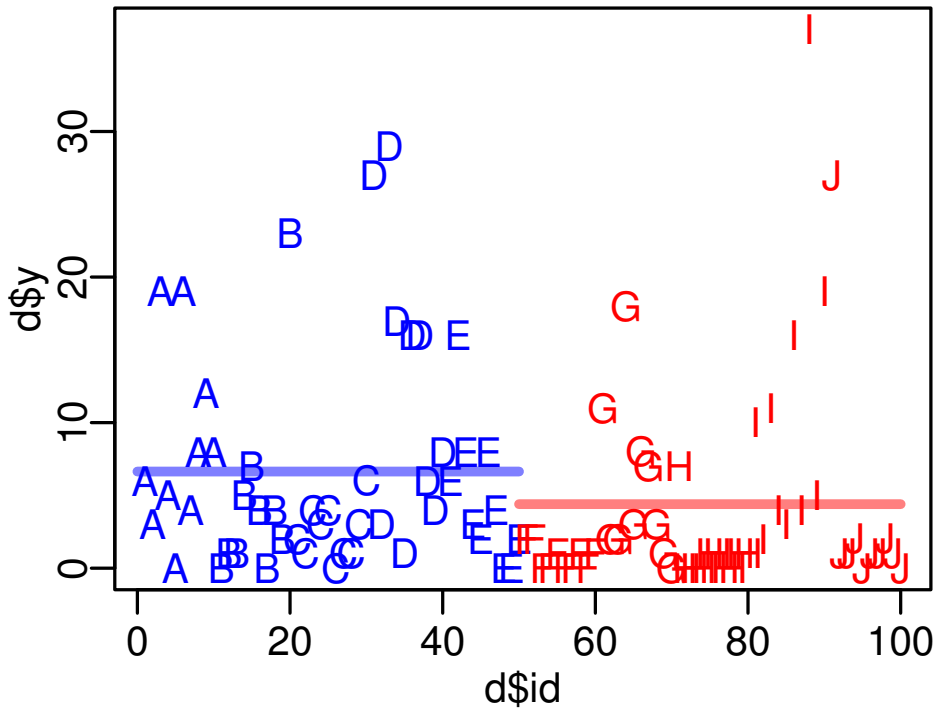
- id **列**: **個体番号** $\{1, 2, 3, \cdots, 100\}$
- pot **列**: **植木鉢名** $\{\mathrm{A}, \mathrm{B}, \mathrm{C}, \cdots, \mathrm{J}\}$
- f **列**: **処理**: **コントロール** C, **肥料** T
- y **列**: **種子数** (**応答変数**)

データはとにかく図示する!!



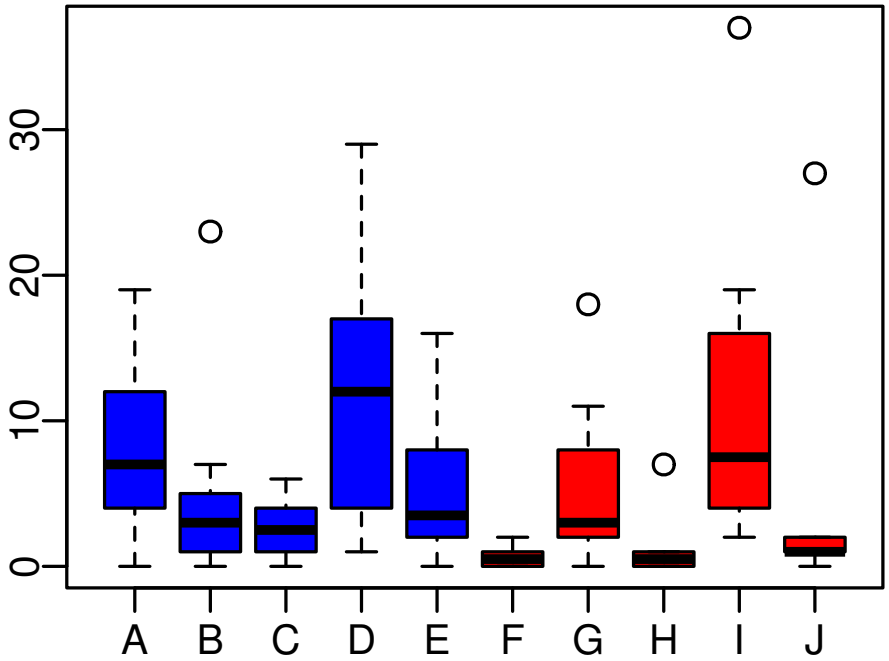
- `plot(d$id, d$y, pch = as.character(d$pot), ...)`
- **コントロール・処理 でそんなに差がない**?

処理ごとの平均も図に追加してみる



- むしろ 処理 のほうが平均種子数が低い?
- (注) この架空データは 肥料の効果はゼロ と設定して生成した

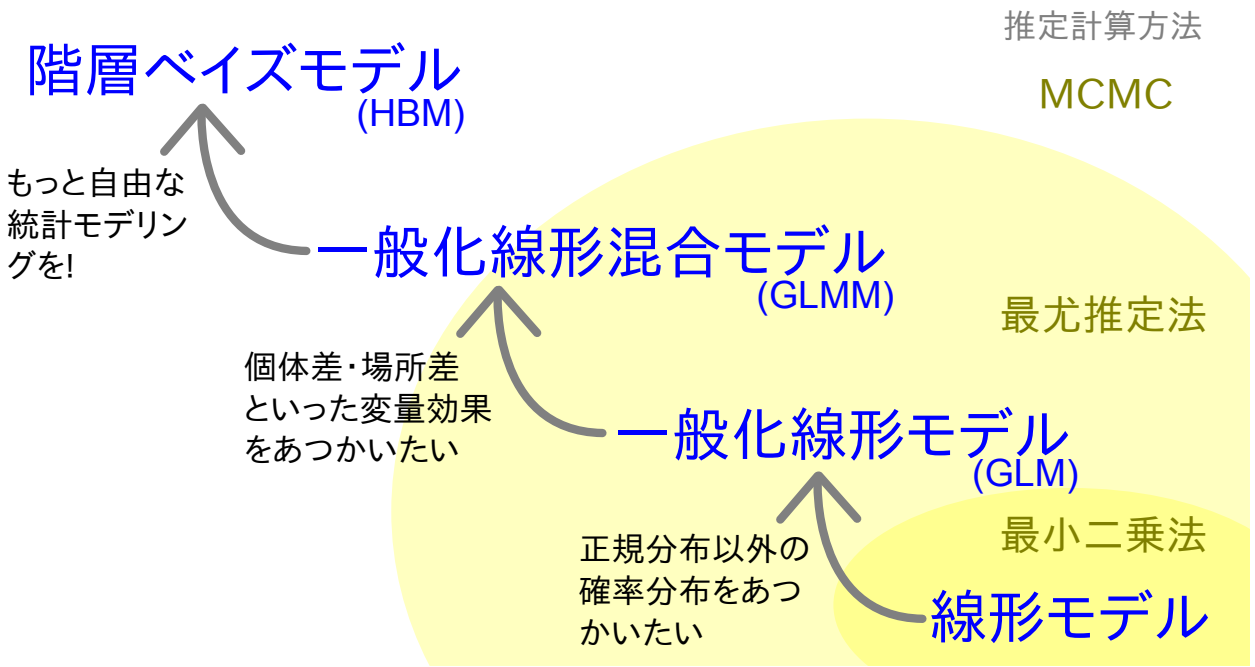
個体差だけでなく植木鉢差もありそう?



- `plot(d$pot, d$y, col = rep(c("blue", "red"), each = 5))`
- **植木鉢由来の** random effects **みたいなものはブロック差と呼ばれる**

(一般化な) 線形モデルのわくぐみで，とりあえず考えてみる

線形モデルの発展



GLM: 個体差もブロック差も無視

```
> summary(glm(y ~ f, data = d, family = poisson))
...(略)...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.8931     0.0549   34.49  < 2e-16
fT           -0.4115     0.0869   -4.73  2.2e-06
...(略)...
```

- 肥料をやる処理 (f) をすると，平均種子数が下がる?
- AIC でモデル選択しても同じような結果に

GLMM: 個体差だけ考慮，ブロック差は無視

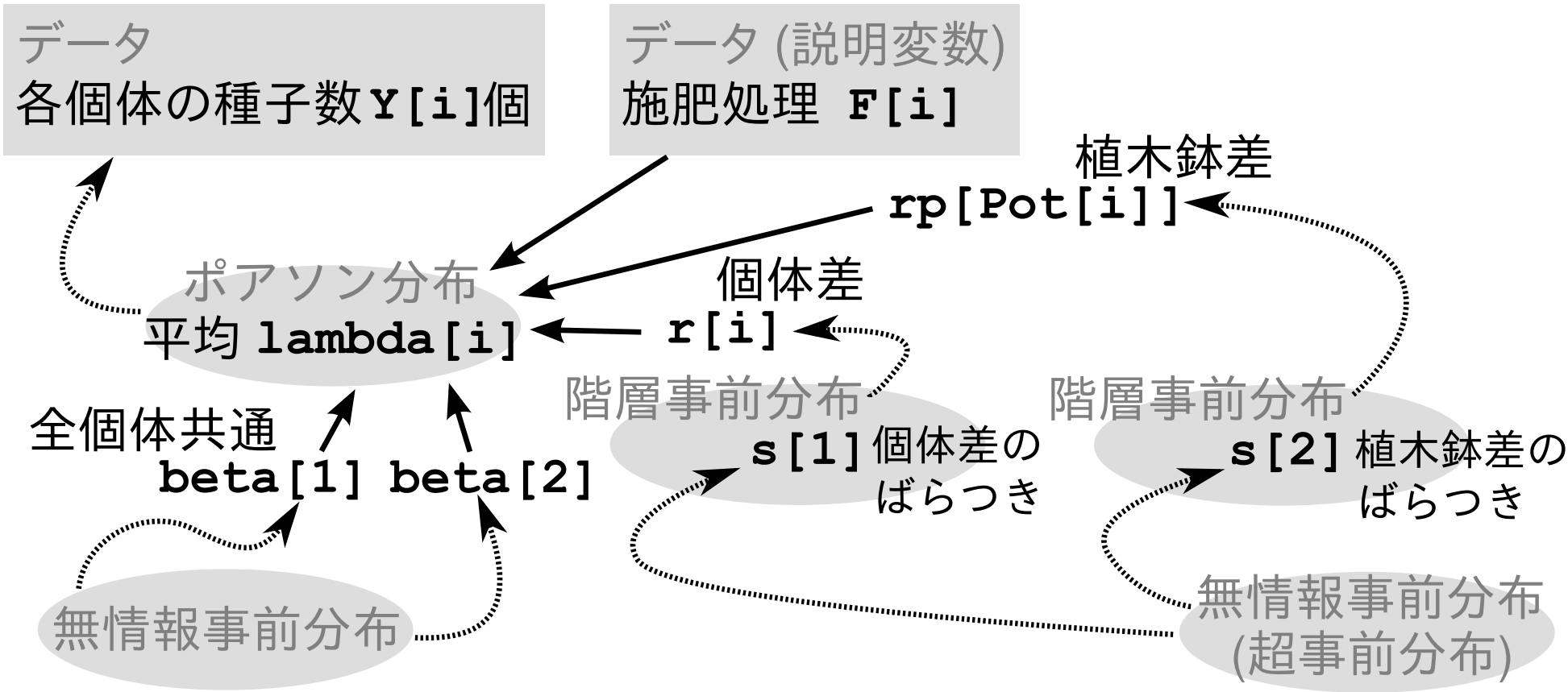
```
> library(glmmML)
> summary(glmmML(y ~ f, data = d, family = poisson,
+ cluster = id))
...(略)...
                coef se(coef)     z Pr(>|z|)
(Intercept)  1.351    0.192  7.05  1.8e-12
fT          -0.737    0.280 -2.63  8.4e-03
...(略)...
```

- やっぱり同じ?
- むしろ肥料処理の悪影響が強い?

個体差 + ブロック差を考える階層ベイズモデル

- ここでは log リンク関数を使う
- 平均の対数 $\log(\lambda_i) = a + bf_i +$ (個体差) + (ブロック差)
- 事前分布の設定
  - 切片 a と f_i の係数 b は無情報事前分布 (すごく平らな正規分布)
  - 個体差とブロック差は階層的な事前分布 (それぞれ標準偏差 σ_1, σ_2 の正規分布，平均はゼロ)
  - 標準偏差 σ_* は無情報事前分布 ($[0, 10^4]$ の一様分布)

植木鉢問題の階層ベイズモデルの図示



個体差 + ブロック差を考える階層ベイズモデル

- ここでは log リンク関数を使う
- 平均の対数 $\log(\lambda_i) = a + bf_i +$ (個体差) + (ブロック差)
- 事前分布の設定
  - 切片 a と f_i の係数 b は無情報事前分布 (すごく平らな正規分布)
  - 個体差とブロック差は階層的な事前分布 (それぞれ標準偏差 σ_1, σ_2 の正規分布，平均はゼロ)
  - 標準偏差 σ_* は無情報事前分布 ($[0, 10^4]$ の一様分布)

個体差 + ブロック差のあるポアソン回帰の BUGS code (1)

```
model
{
  for (i in 1:N.sample) {
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- a + b * F[i] + r[i] + rp[Pot[i]]
  }
  # 次のページの事前分布の定義につづく
```

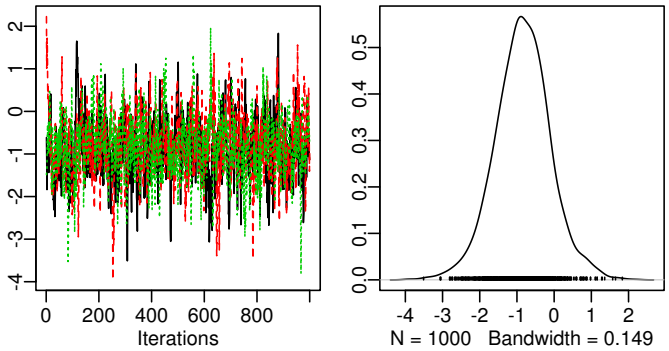
ここでの BUGS coding のポイント

- **因子型の説明変数** $f_i \in \{\mathtt{C}, \mathtt{T}\}$ **は，それぞれ** `F[i]` **を** `0`，`1` **と置きかえる**
- `Pot[i]` **は** `1`，`2`，`...`，`10` **と数字になおした植木鉢名をいれておいて，植木鉢の効果** `rp[...]` **を参照させる**

個体差 + ブロック差のあるポアソン回帰の BUGS code (2)

```
  # 前のページからのつづき
  a ~ dnorm(0, 1.0E-4) # 切片
  b ~ dnorm(0, 1.0E-4) # 肥料の効果
  for (i in 1:N.sample) {
    r[i] ~ dnorm(0, tau[1])  # 個体差
  }
  for (j in 1:N.pot) {
    rp[j] ~ dnorm(0, tau[2]) # 植木鉢の差（ブロック差）
  }
  for (k in 1:N.tau) {
    tau[k] <- 1.0 / (sigma[k] * sigma[k]) # 個体・植木鉢のばらつき
    sigma[k] ~ dunif(0, 1.0E+4)
  }
}
```

肥料の効果 (パラメーター b) はなさそう?

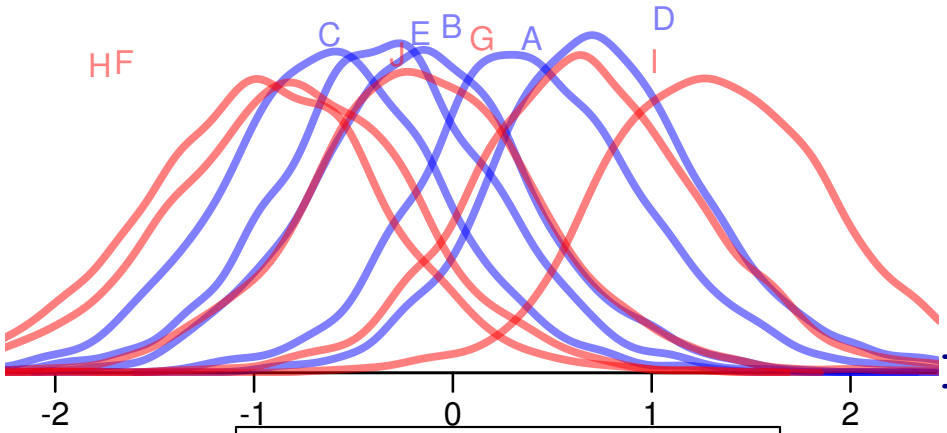


	mean	sd	2.5%	25%	50%	75%	97.5%	Rha
a	1.501	0.529	0.482	1.157	1.493	1.852	2.565	1.03
b	-1.016	0.706	-2.436	-1.476	-0.993	-0.565	0.395	1.01
sigma[1]	1.020	0.114	0.822	0.939	1.014	1.089	1.265	1.00

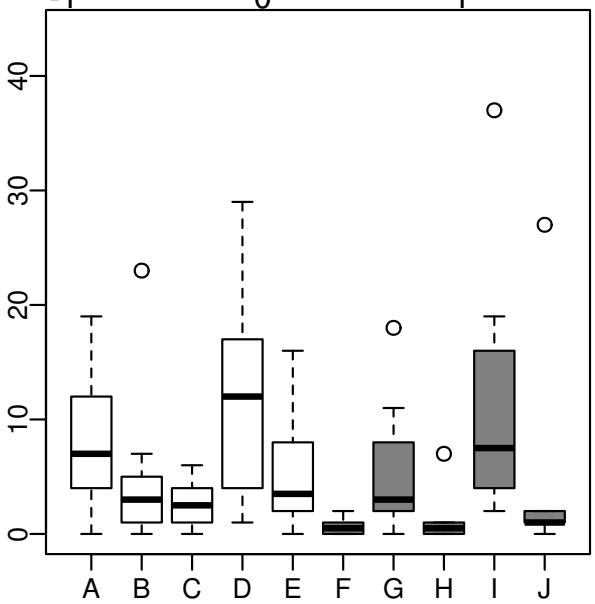
...(略)...

この架空データを生成した種子数シミュレイションでは，肥料の効果はまったく無いと設定していた

推定された植木鉢の差（ブロック差）



ブロック差
rp[j]



統計モデリングの手ぬきは危険!

- random effects つまり 個体差・ブロック差が大きい
- random effects の影響が大きいときには，fixed effects の大きさが見えにくくなる— ニセの「効果」が見えることもあれば，見えるはずの傾向が隠されることも
  - **個体差・ブロック差の階層ベイズモデルが必要!**
- もしブロック差を人為的に小さくできないなら，ブロック数をもっと増やして，より正確な植木鉢の効果のばらつきを正確に推定するしかない