統計モデリング入門 2014 (7)

階層ベイズモデル

久保拓弥 kubo@ees.hokudai.ac.jp

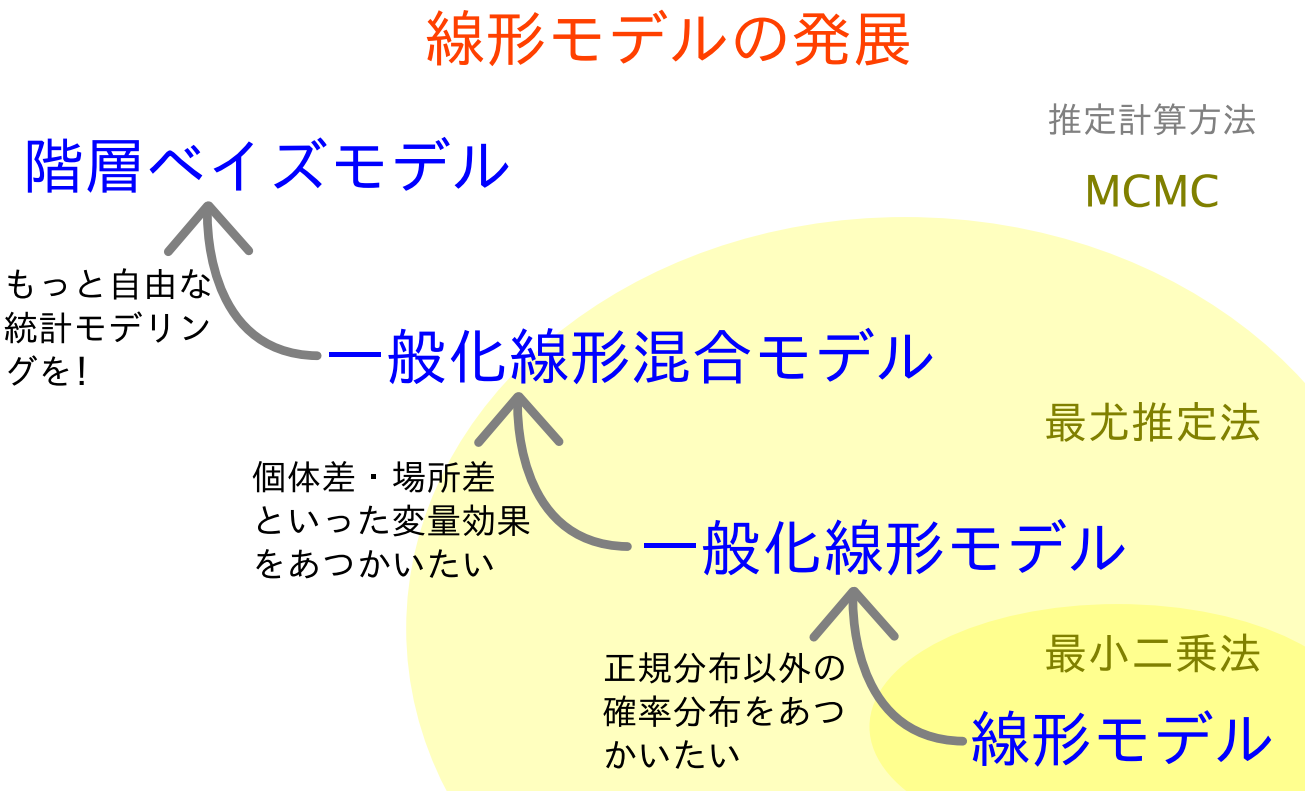
北大環境科学院の講義 http://goo.gl/XeBR2x

2014–07–28

ファイル更新時刻: 2015–01–15 16:14

より現実的・実戦的なデータ解析のために

- **一般化線形モデル** → **階層ベイズモデル**
- **最尤推定** → Markov chain Monte Carlo (MCMC)



今回のハナシで説明したいこと

階層ベイズモデルという
「現象の統計モデル化」は便利，
これを使うためには MCMC という
推定技法もちょっと勉強してみましょう

今日の内容と統計モデリング入門との対応

http://goo.gl/Ufq2

今日はおもに「第 8–10 章」の内容を説明します．

- 著者: 久保拓弥
- 出版社: 岩波書店
- 2012–05–18 刊行

このあとのハナシのながれ

1. **二項分布のパラメーターを最尤推定**

   あえてものすごく簡単な例題

2. **同じような推定を MCMC でやってみる**

   最尤推定と MCMC はちがう!

3. **GLM だけでは実際のデータ解析はできない**

   階層ベイズモデル (である GLMM) の導入

4. **階層ベイズモデルの推定**

   ソフトウェア WinBUGS を使ってみる

5. **階層ベイズモデルの応用**

   複数の階層，時間や空間の構造

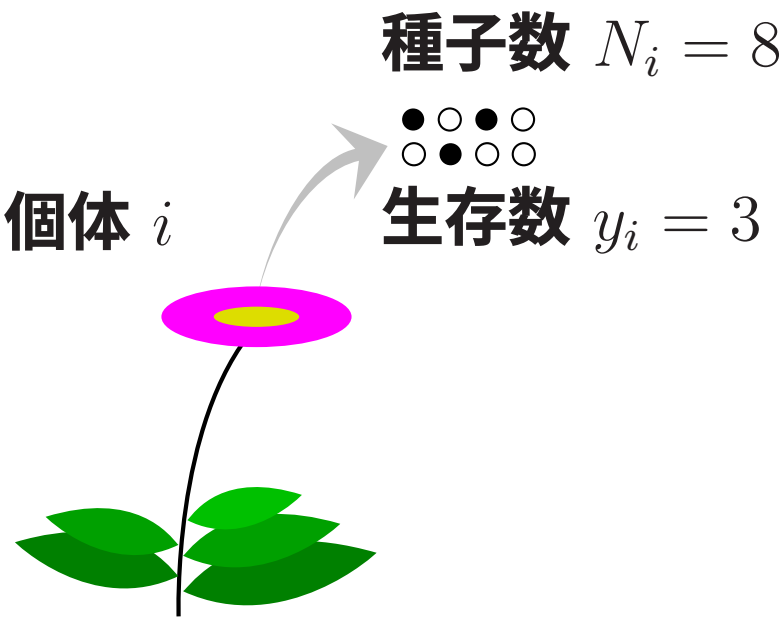
1. 二項分布のパラメーターを最尤推定

あえてものすごく簡単な例題

いちおう GLM 的なモデル

例題: 植物の種子の生存確率

- 架空植物の種子の生存を調べた
- **種子**: 生きていれば発芽する
  - どの個体でも 8 個 の種子を調べた
- **生存確率**: ある種子が生きている確率

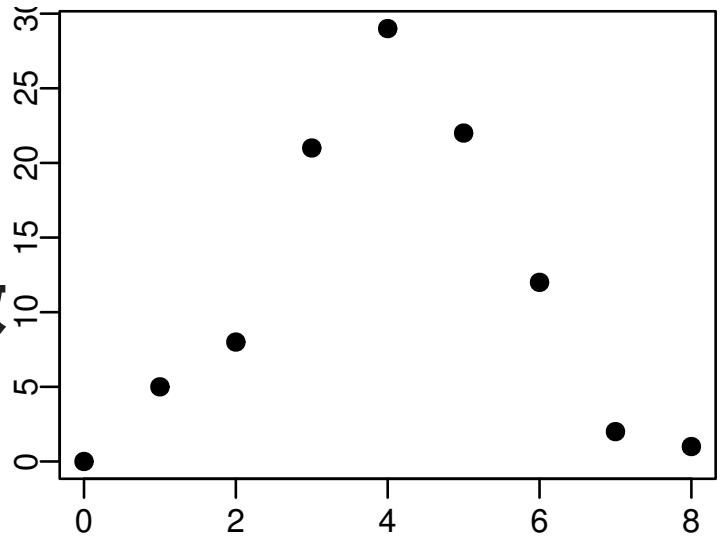


- データ: 植物 100 個体，合計 800 種子の生存の有無を調べた
- 問: この植物の生存確率はどのように統計モデル化できるか?

簡単すぎる例題: 生存確率は全個体で同じ（「個体差」なし）

個体ごとの生存数	0	1	2	3	4	5	6	7	8
観察された個体数	0	5	8	21	29	22	12	2	1

観察された
植物の個体数



生存していた種子数 y_i

これは個体差なしの均質な集団

生存確率 q と二項分布の関係

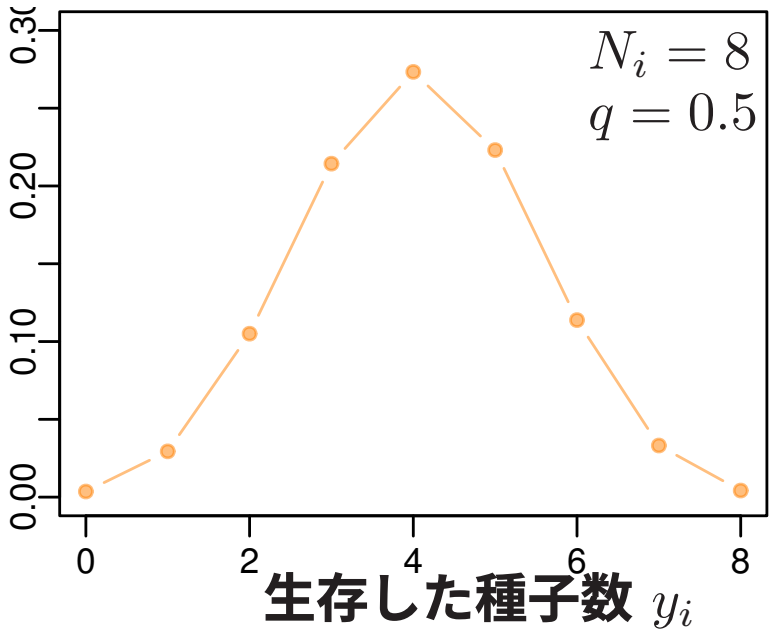
- 生存確率を推定するために**二項分布** という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率

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

- ここで仮定していること
  - **個体差はない**
  - つまり **すべての個体で同じ生存確率** q

二項分布の図示例: 生存確率 $q = 0.5$

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



尤度 (ゆうど): 100 個体ぶんのデータが観察される確率

- 観察データ $\{y_i\}$ が確定しているときに
- パラメータ q は値が自由にとりうると考える
- 尤度 は 100 個体ぶんのデータが得られる確率の積，パラメータ q の関数として定義される

$$L(q|\{y_i\}) = \prod_{i=1}^{100} p(y_i \mid q)$$

個体ごとの生存数	0	1	2	3	4	5	6	7	8
観察された個体数	0	5	8	21	29	22	12	2	1

対数尤度方程式と最尤推定

- この尤度 $L(q \mid \text{データ})$ を最大化するパラメータの推定量 $\hat{q}$ を計算したい
- 尤度を対数尤度になおすと

$$\log L(q \mid \text{データ}) = \sum_{i=1}^{100} \log \binom{N_i}{y_i} + \sum_{i=1}^{100} \{y_i \log(q) + (N_i - y_i) \log(1-q)\}$$

- この対数尤度を最大化するように未知パラメーター q の値を決めてやるのが**最尤推定**

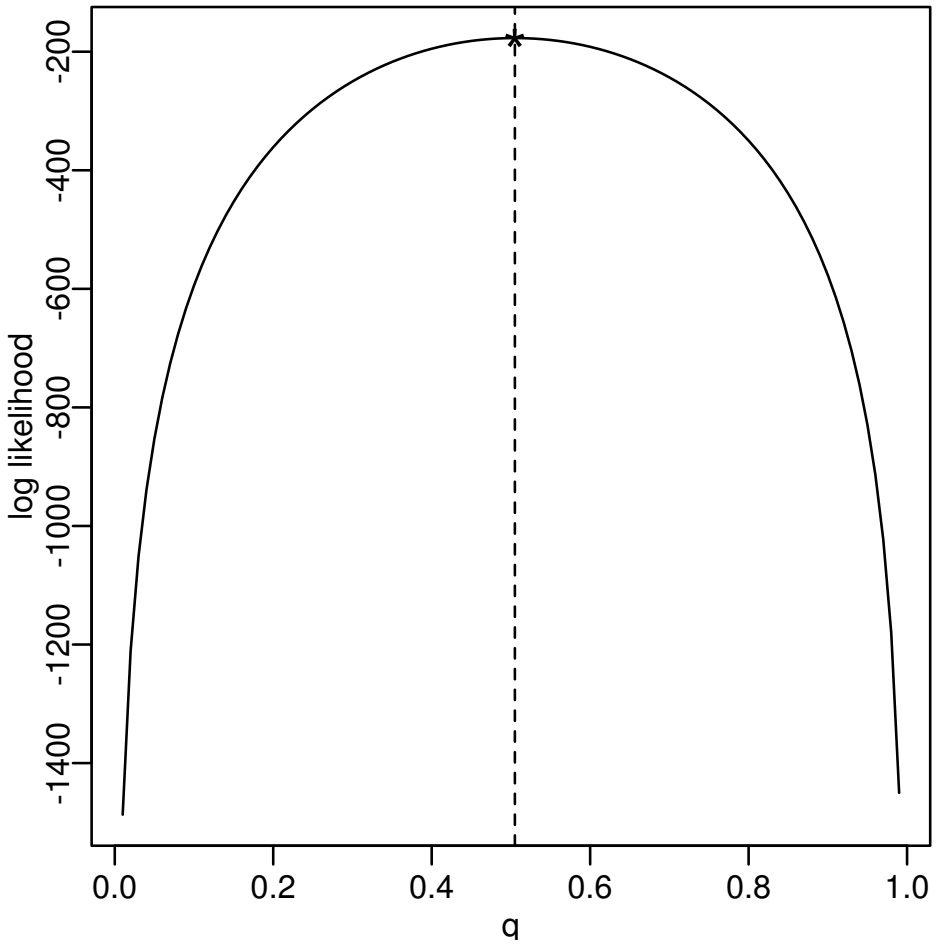
最尤推定 (MLE) とは何か

- 対数尤度 $L(q \mid$ データ$)$ が最大になるパラメーター q の値をさがしだすこと

- 対数尤度 $\log L(q \mid$ データ$)$ を q で偏微分して 0 となる $\hat{q}$ が対数尤度最大

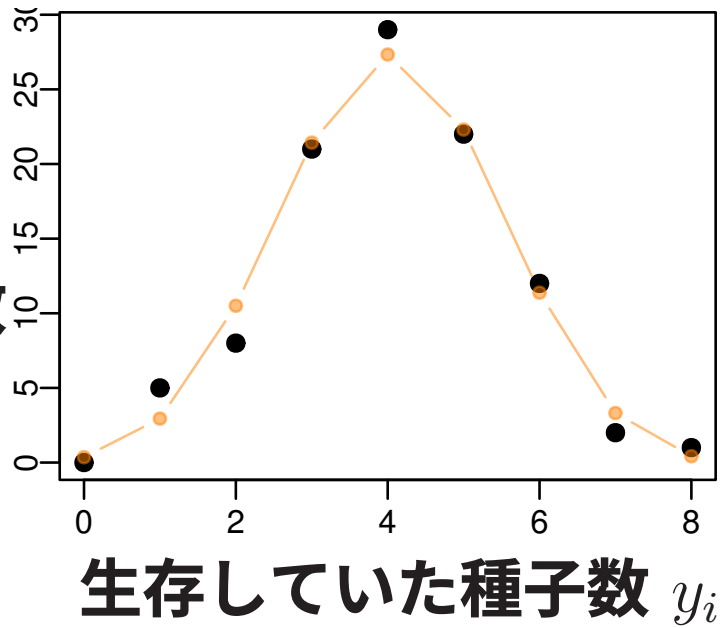
  $\partial \log L(q \mid$ データ$)/\partial q = 0$
- 生存確率 q が全個体共通の場合の最尤推定量・最尤推定値は

$$\hat{q} = \frac{\text{生存種子数}}{\text{調査種子数}} = \frac{404}{800} = 0.505$$



二項分布で説明できる 8 種子中 y_i 個の生存

$\hat{q} = 0.505$ **なので** $\binom{8}{y} 0.505^y 0.495^{8-y}$



とりあえずここまでで
確率分布を適切に選んで統計モデリング，
統計モデルのパラメーターを
最尤（さいゆう）推定 する
…… といったことを説明しました

2. 同じような推定を MCMC でやってみる

最尤推定と MCMC はちがう!

そして「なぜかしら」ベイズ統計モデルと関連づけ

ここでやること: 尤度と MCMC の関係を考える

- さきほどの簡単な例題 (生存確率) のデータ解析を
- 最尤推定ではなく
- Markov chain Monte Carlo (MCMC) 法のひとつであるメトロポリス法 (Metropolis method) であつかう
- 得られる結果: 「パラメーターの値の分布」……??

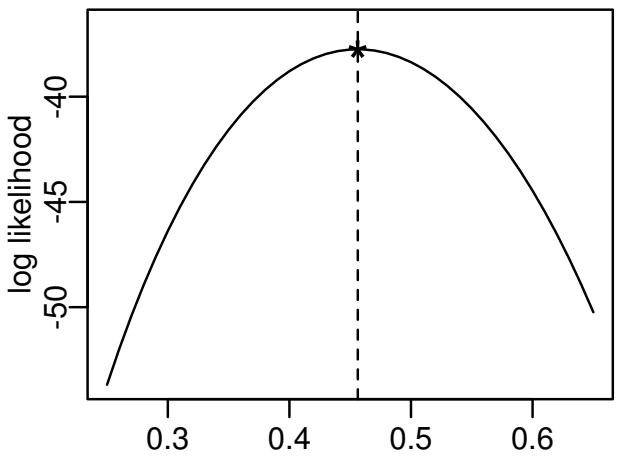
MCMC をもちださなくてもいい簡単すぎる問題

説明のためあえてメトロポリス法を適用してみる

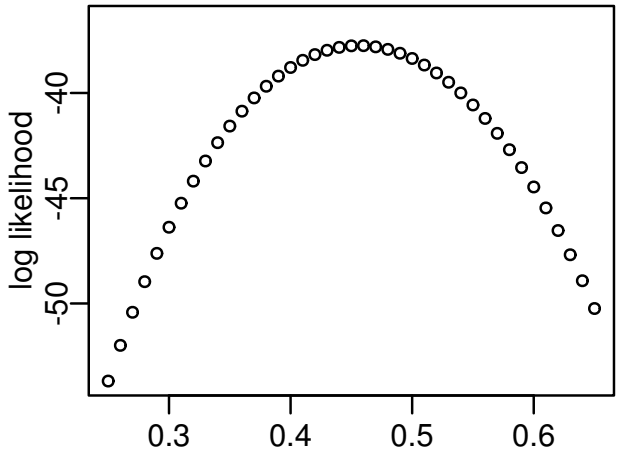
メトロポリス法を説明するための準備

連続的な対数尤度関数

$\log L(q)$



離散化: q がとびとびの値をとる



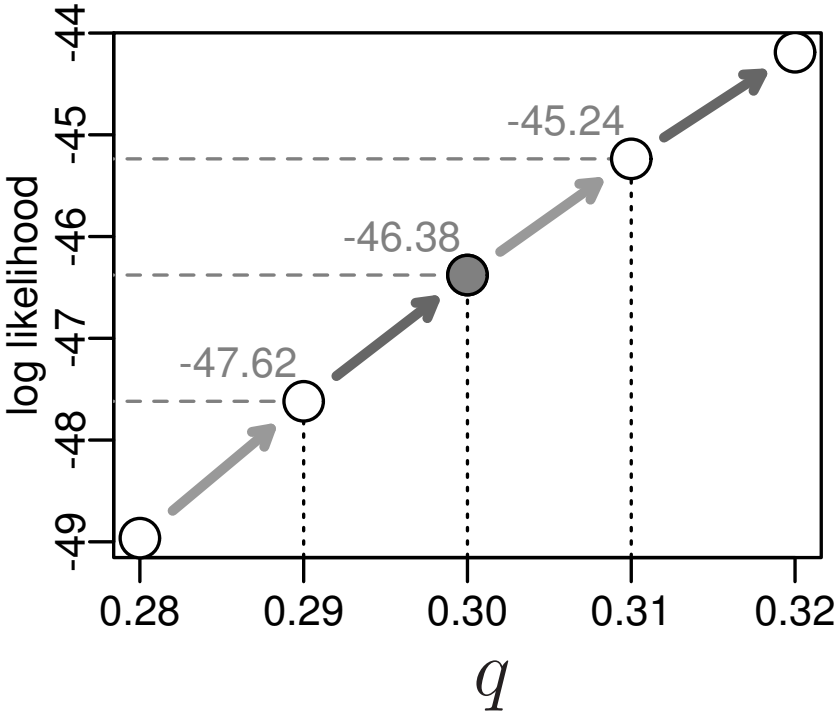
説明を簡単にするため

生存確率 q の軸を離散化する

(実際には離散化する必要などない)

試行錯誤による q の最尤推定値の探索

ちょっと効率の悪い「試行錯誤の最尤推定」



1. q の値の「行き先」を「両隣」どちらかにランダムに決める
2. 「行き先」が現在の尤度より高ければ，q の値をそちらに変更
3. 尤度が変化しなくなるまで $(1), (2)$ をくりかえす

メトロポリス法のルール: この例題の場合

1. パラメーター q の初期値を選ぶ

   (q の初期値が 0.3)

2. q を増やすか減らすかをランダムに決める

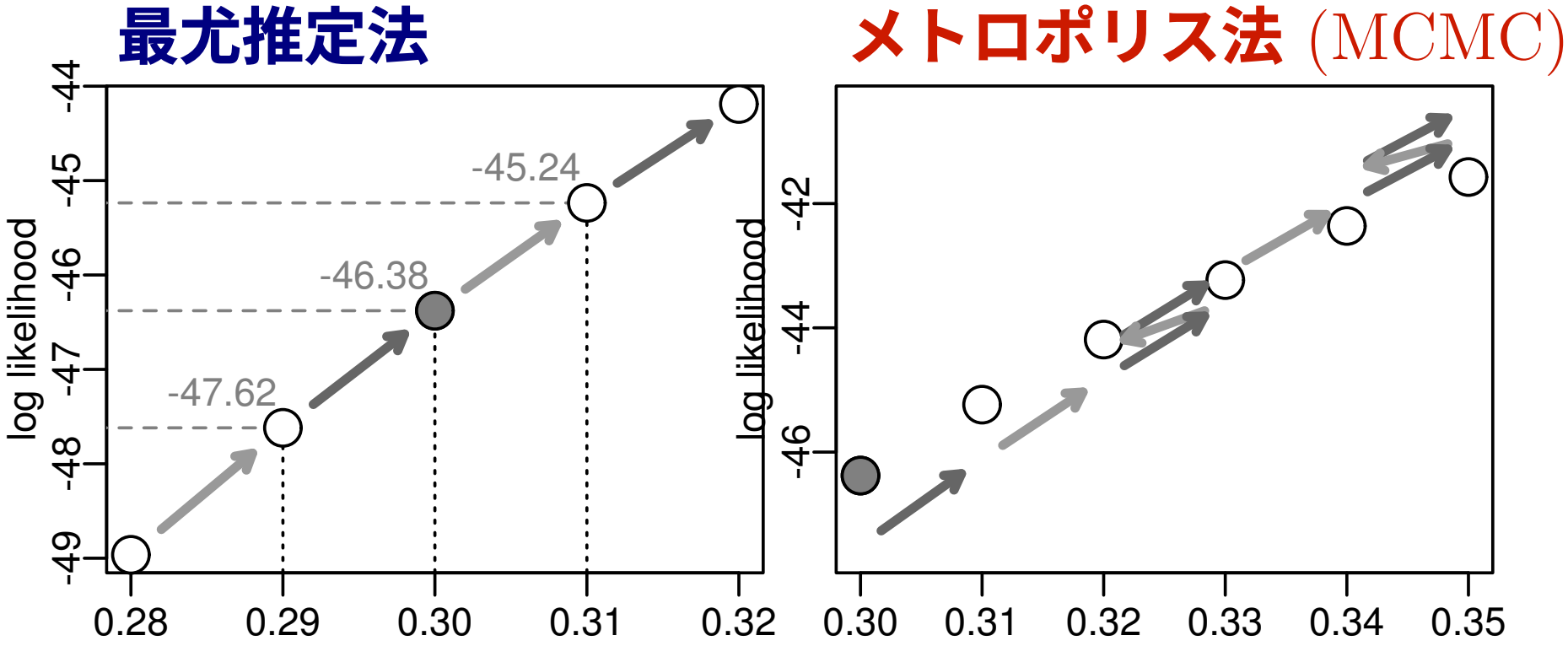
   (新しく選んだ q の値を q_{new} としましょう)

3. q_{new} における尤度 $L(q_{\mathrm{new}})$ ともとの尤度 $L(q)$ を比較
   - $L(q_{\mathrm{new}}) \geq L(q)$ (あてはまり改善): $q \leftarrow q_{\mathrm{new}}$
   - $L(q_{\mathrm{new}}) < L(q)$ (あてはまり改悪):
     - 確率 $r = L(q_{\mathrm{new}})/L(q)$ で $q \leftarrow q_{\mathrm{new}}$
     - 確率 $1 - r$ で q を変更しない

4. 手順 $2.$ にもどる

   ($q = 0.01$ や $q = 0.99$ でどうなるんだ，といった問題は省略)

メトロポリス法のルールで q を動かす

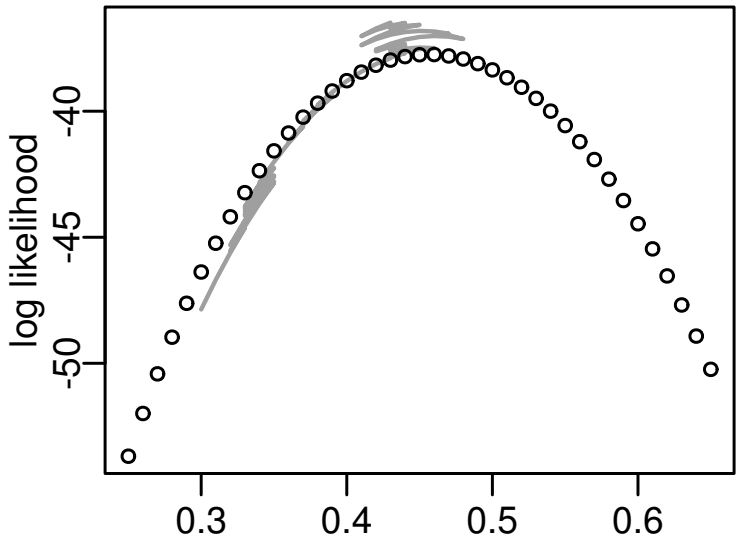


メトロポリス法だと
「単調な山のぼり」にはならない

対数尤度関数の「山」でうろうろする q の値

メトロポリス法 (そして一般の MCMC) は

最適化ではない

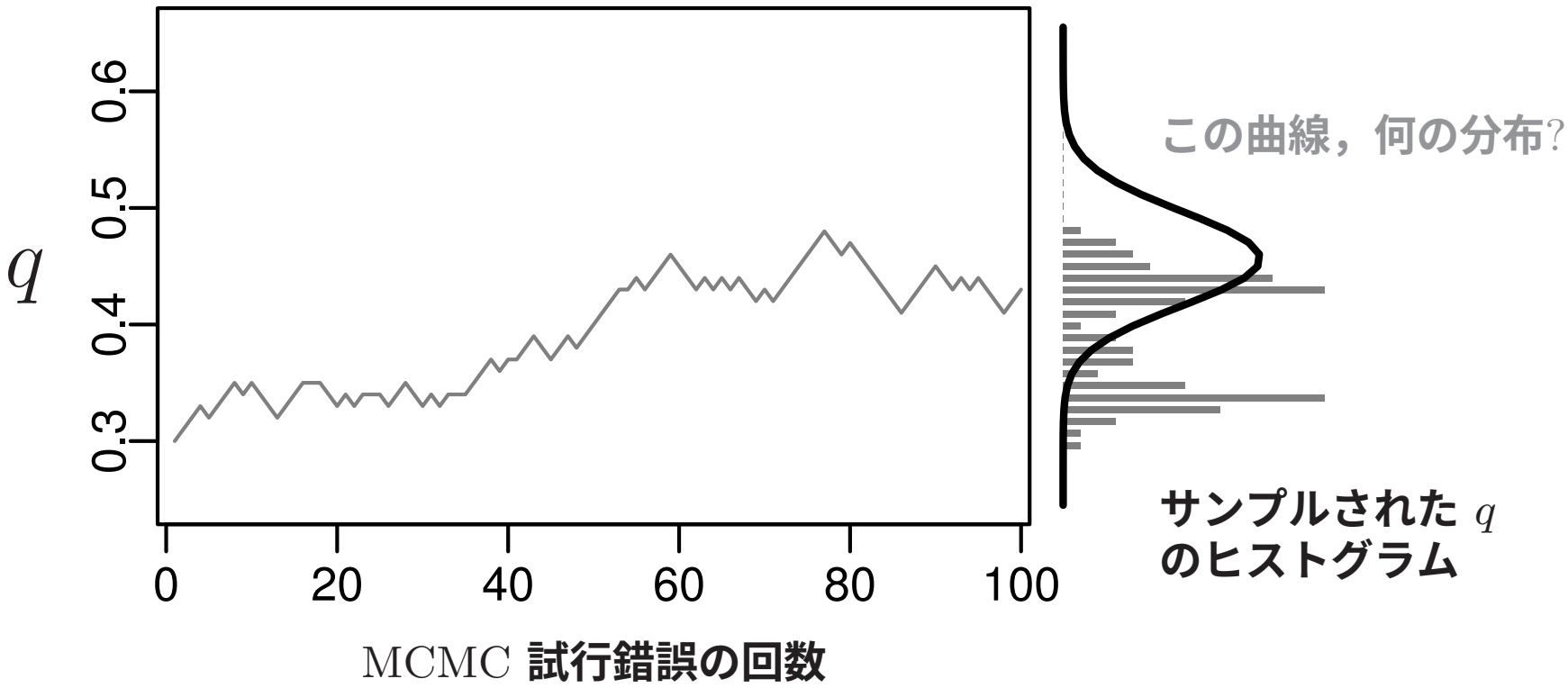


ときどきはでに落っこちる

何のためにこんなことをやるのか?

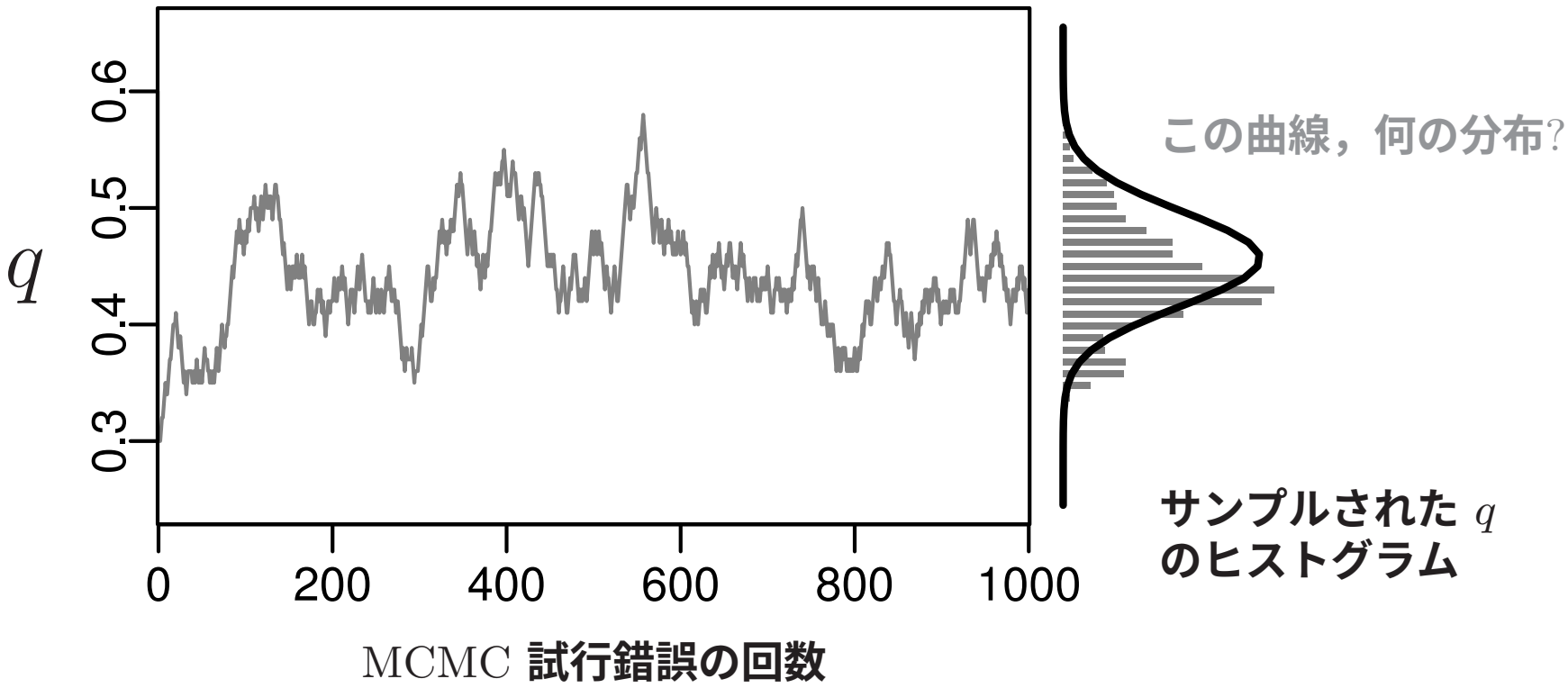
q の変化していく様子を記録してみよう

ステップごとに q の値をサンプリング



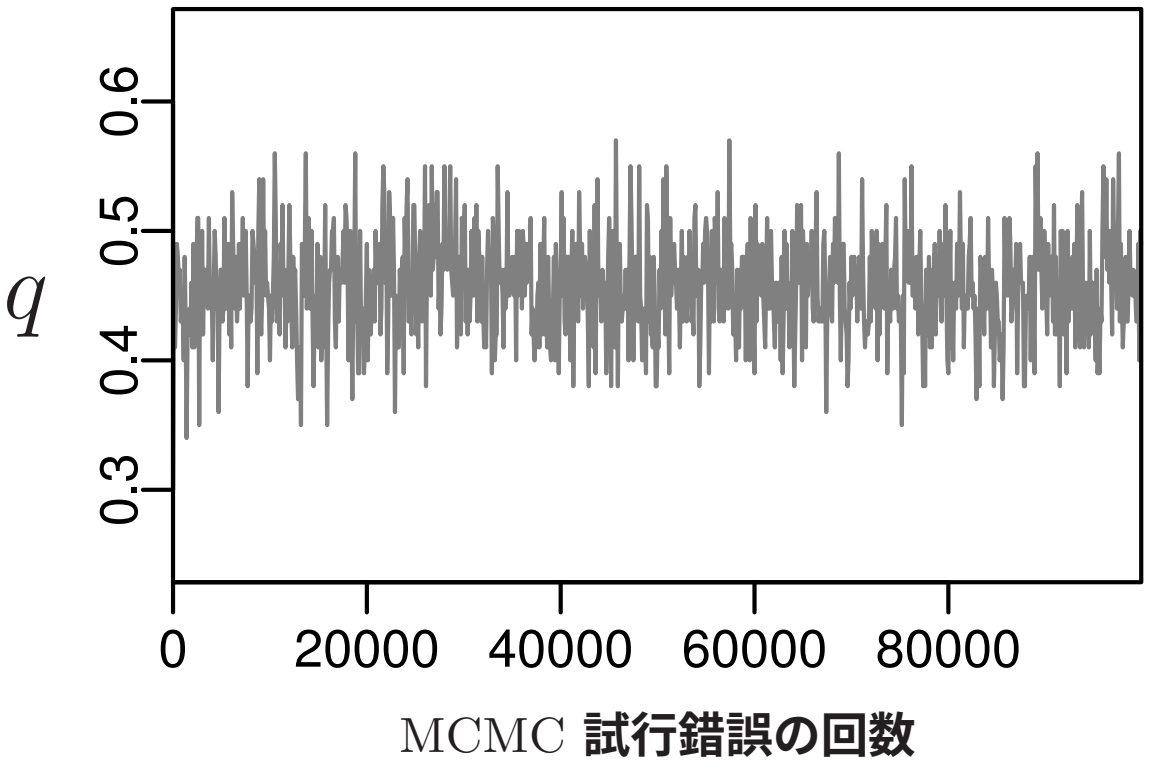
もっと試行錯誤してみたほうがいいのかな?

もっと長くサンプリングしてみる



まだまだ……?

もっともっと長くサンプリングしてみる



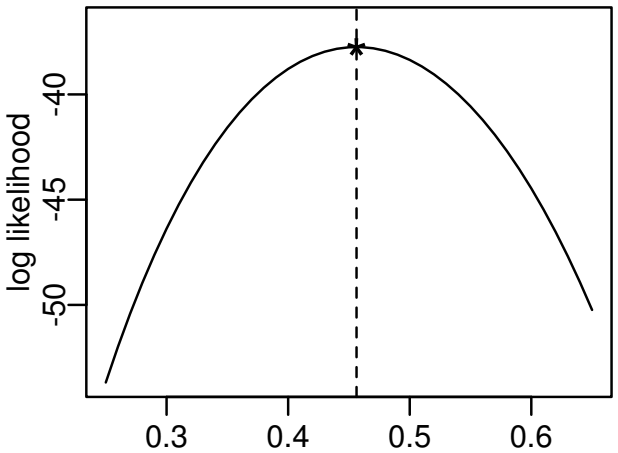
じつはこれは
「q の確率分布」
……このあと説明

サンプルされた q
のヒストグラム

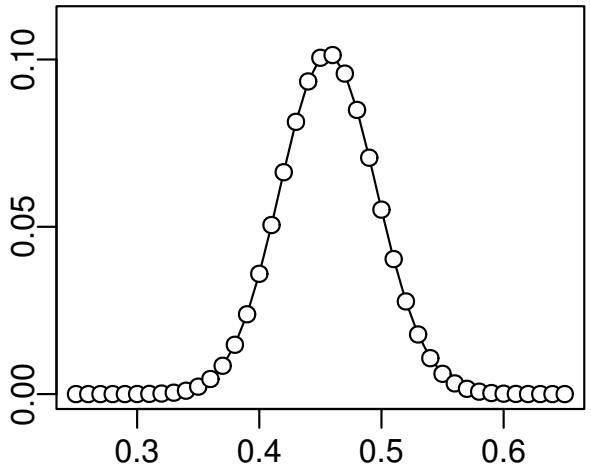
なんだか，ある「山」のかたちにまとまったぞ?

MCMC は何をサンプリングしている?

対数尤度 $\log L(q)$



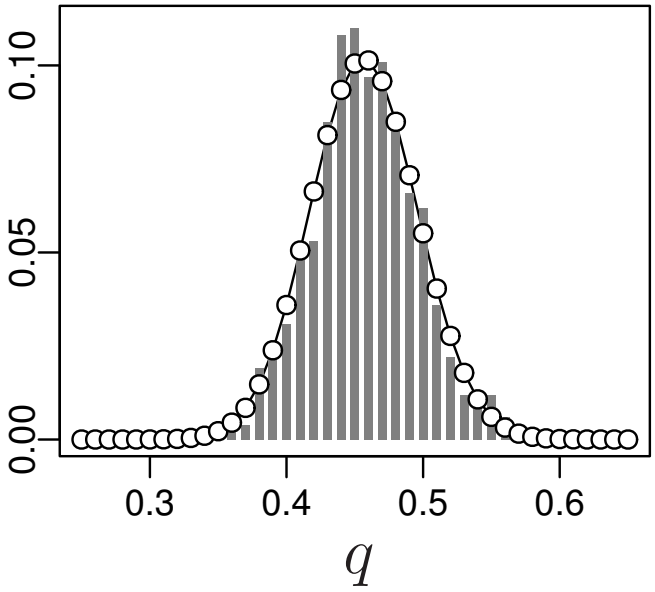
尤度 $L(q)$ **に比例する確率分布**



尤度に比例する確率分布からのランダムサンプル

マルコフ連鎖の定常分布は $p(q) = \dfrac{L(q)}{\sum_q L(q)}$ **となる**

MCMC の結果として得られた q の経験分布



- データと統計モデル (二項分布) を決めて，MCMC サンプリングすると，$p(q)$ からのランダムサンプルが得られる
- このランダムサンプルをもとに，q の平均や 95% 区間などがわかる— 便利じゃないか!

MCMC という推定方法から「パラメーター q の確率分布」というちょっと奇妙な考えかたがでてきた……

「ふつう」の統計学では
「パラメーターの確率分布」といった
考えかたはしない，しかし ……

ベイズ統計学なら「パラメーターの確率分布」はぜんぜん自然な考えかただ

ベイズモデル: 尤度・事後分布・事前分布……

- ベイズの公式　$p(q \mid \mathbf{Y}) = \dfrac{p(\mathbf{Y} \mid q) \times p(q)}{p(\mathbf{Y})}$
- $p(q \mid \mathbf{Y})$ は何かデータ $(\mathbf{Y})$ のもとで何かパラメーター (q) が得られる確率 (事後分布)
- $p(q)$ はあるパラメーター q が得られる確率 (事前分布)
- $p(\mathbf{Y} \mid q)$ パラメーターを決めたときにデータが得られる確率 (尤度に比例)
- $p(\mathbf{Y})$ はデータ $\mathbf{Y}$ が得られる確率 (単なる規格化定数)

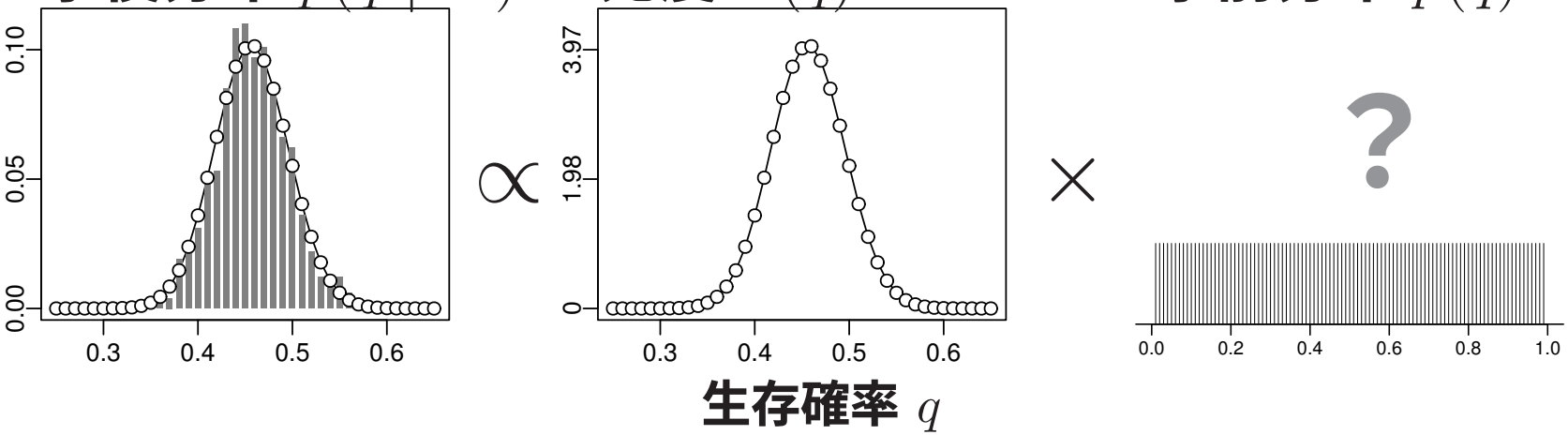
$$(\text{事後分布}) \propto \frac{\text{尤度} \times \text{事前分布}}{(\text{データが得られる確率})}$$

$$\propto \text{尤度} \times \text{事前分布}$$

ベイズ統計にむりやりこじつけてみると?

q の事前分布は一様分布，と考えるとつじつまがあう?

事後分布 $p(q \mid \mathrm{Y})$ $\propto$ 尤度 $L(q)$ $\times$ 事前分布 $p(q)$



事前分布ってのがよくわからない……

以上の説明は，

「MCMC によって得られる結果」

は

「ベイズ統計でいうパラメーターの事後分布」

と考えると解釈しやすいかも

といったことを

ばくぜんかつなんとなく対応づける

ひとつのこころみでありました……

厳密な正当化とかそういったものではありません

3. GLM だけでは実際のデータ解析はできない

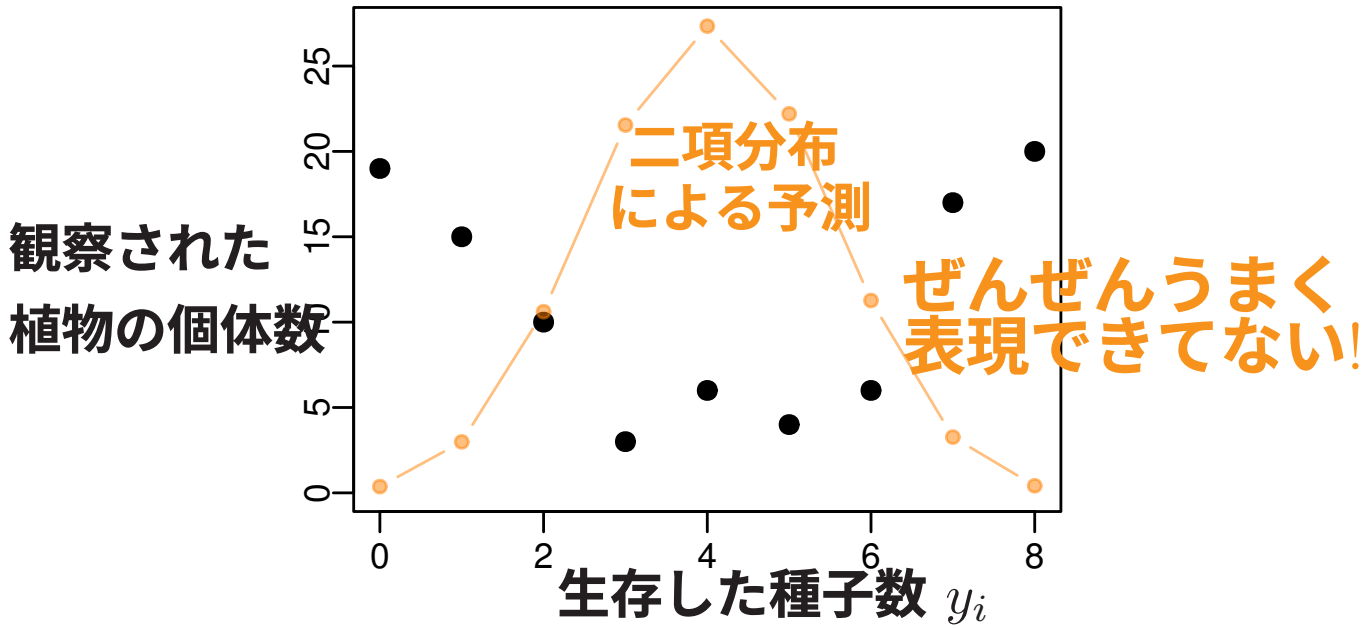
階層ベイズモデル (である GLMM) の導入

(パラメーター推定のハナシのつづきはまたあとで)

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……

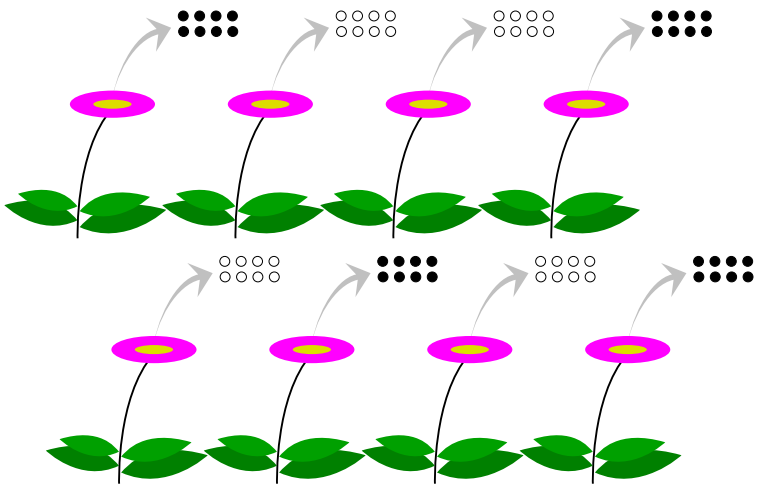


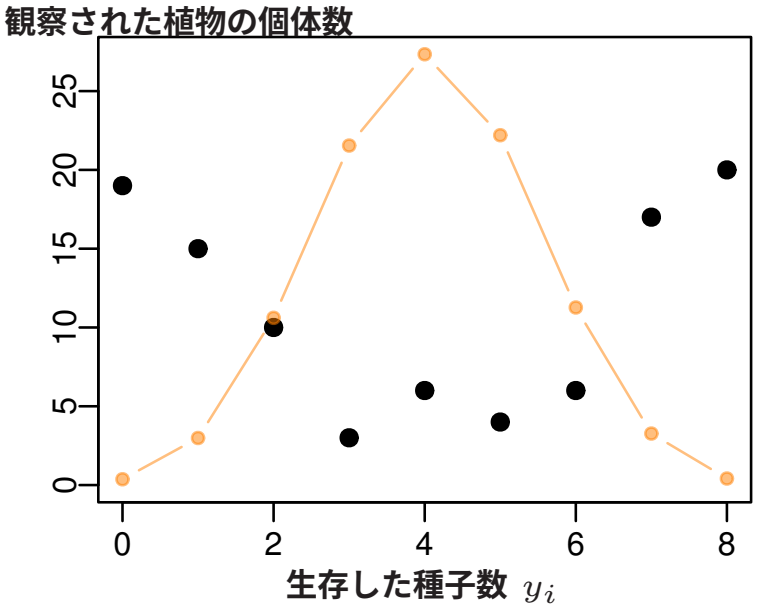
さっきの例題と同じようなデータなのに?

(「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

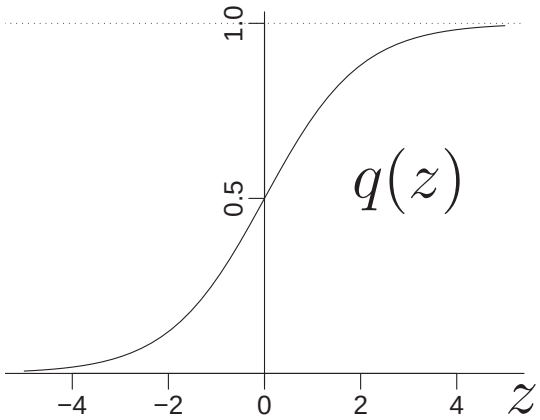
- 生存確率を推定するために 二項分布という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率は二項分布

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

- 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**



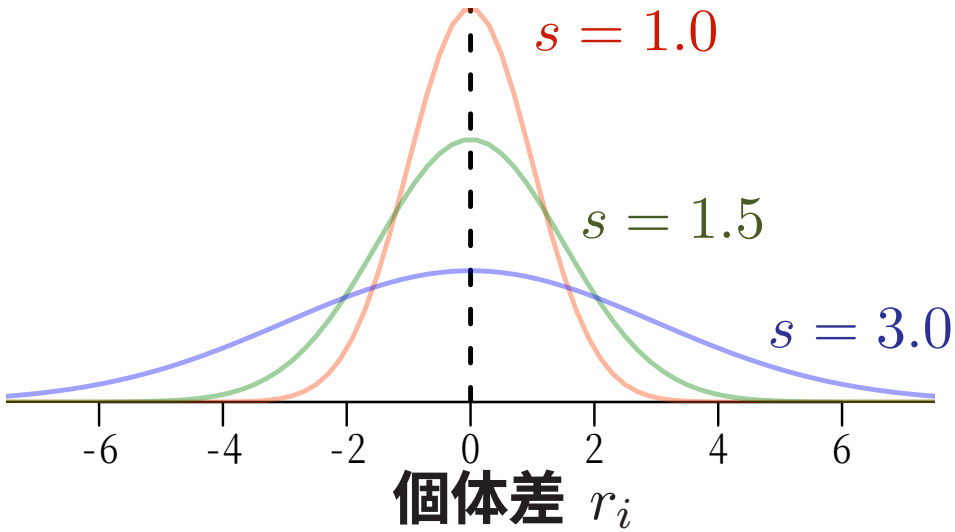
- **線形予測子** $z_i = a + r_i$ **とする**
  - **パラメーター** a**: 全体の平均**
  - **パラメーター** r_i**: 個体** i **の個体差（ずれ）**

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

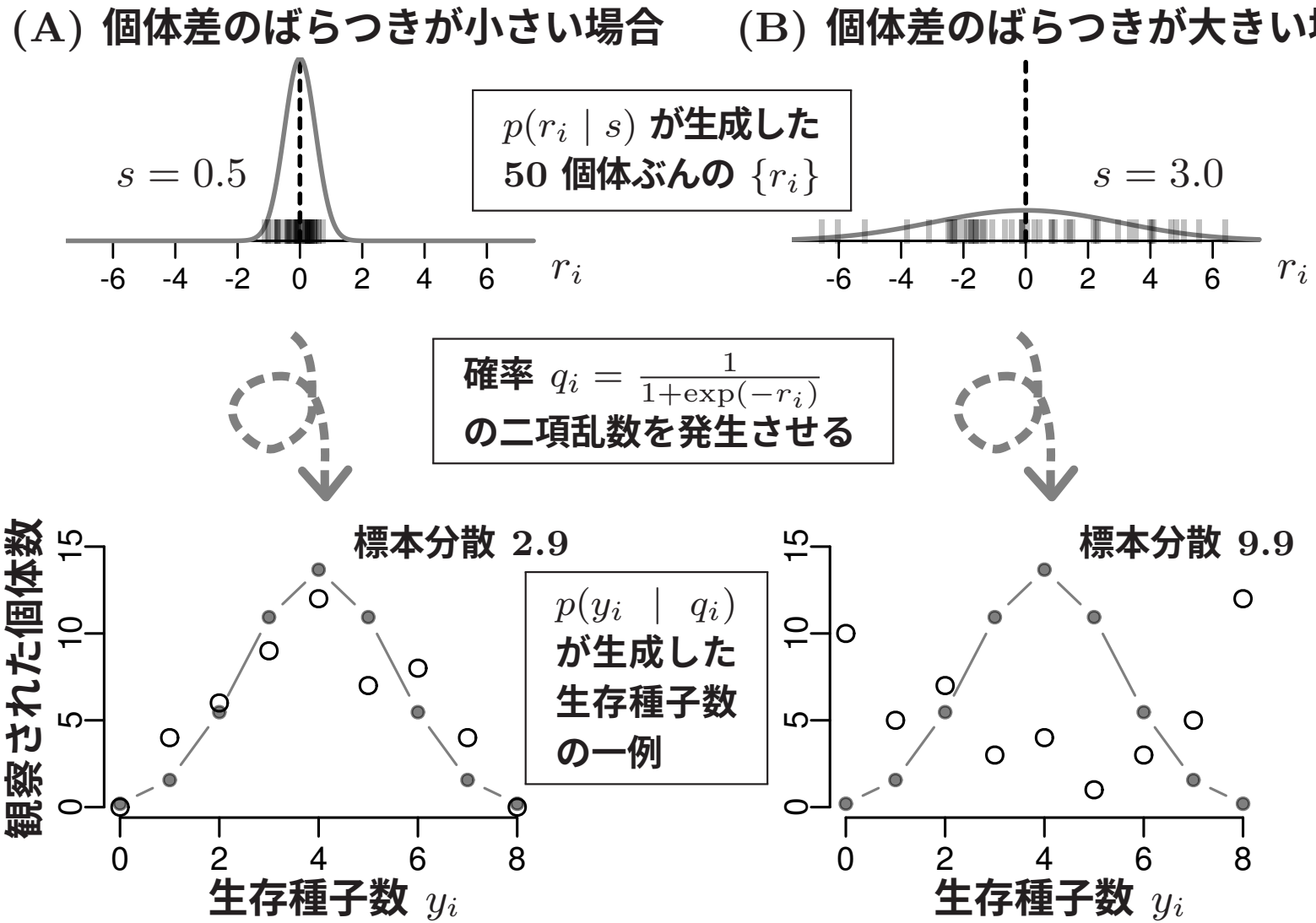
$\{r_i\}$ のばらつきは正規分布だと考えてみる



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

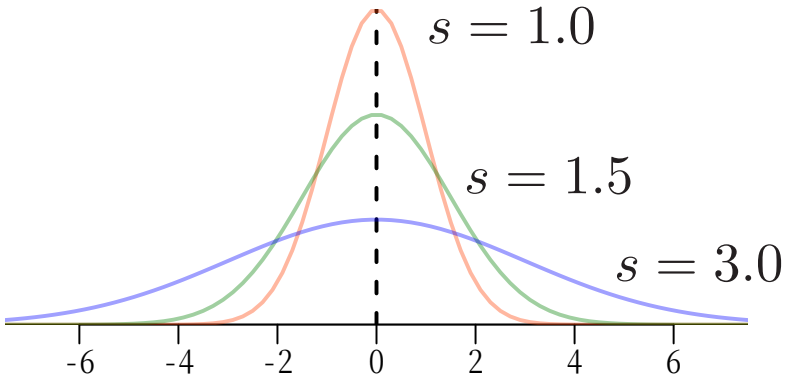
この確率密度 $p(r_i \mid s)$ は r_i の「出現しやすさ」をあらわしていると解釈すればよいでしょう．r_i がゼロにちかい個体はわりと「ありがち」で，r_i の絶対値が大きな個体は相対的に「あまりいない」．

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

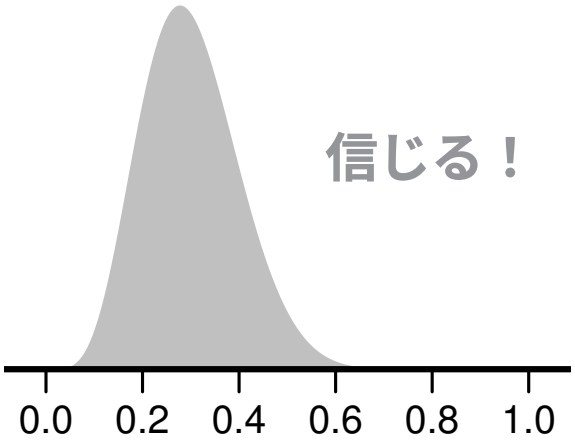


$$p(r_i \mid s) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{r_i^2}{2s^2}\right)$$

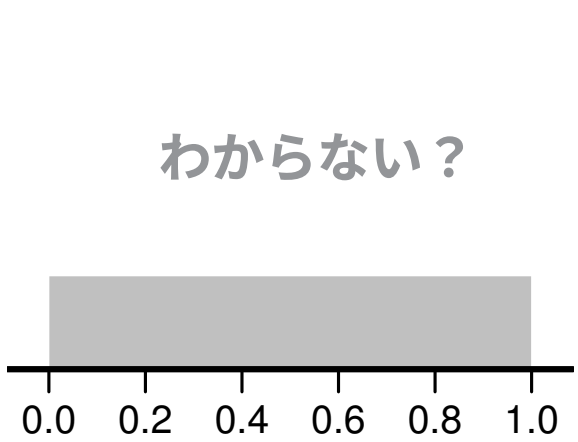
ベイズ統計モデルでよく使われる三種類の事前分布

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．

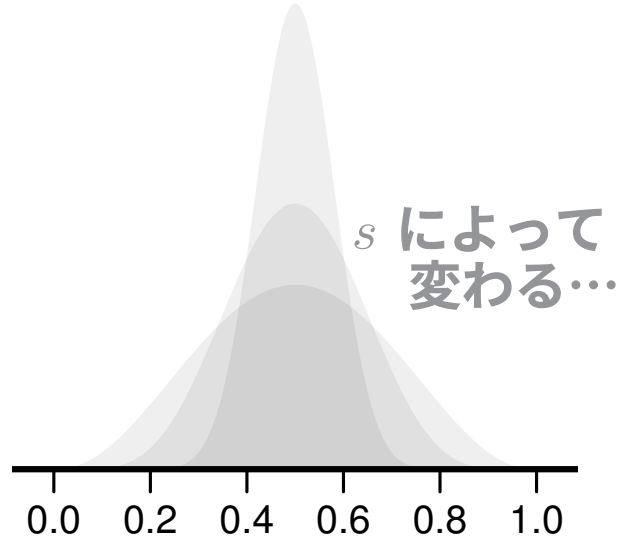
(A) **主観的な事前分布**
(できれば使いたくない!)



(B) **無情報事前分布**



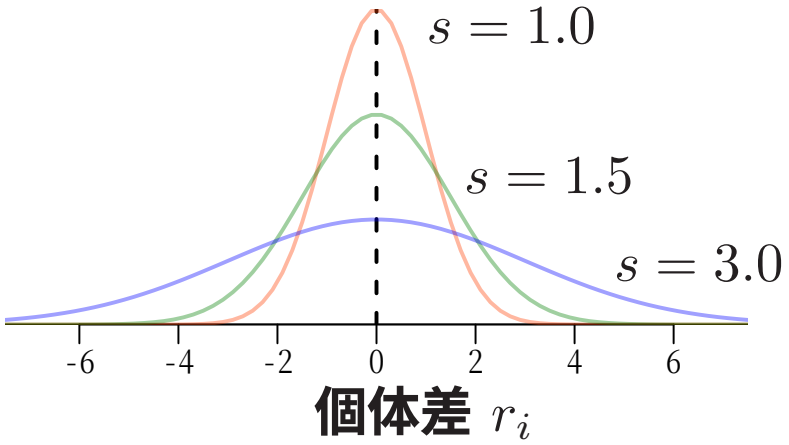
(C) **階層事前分布**



r_i の事前分布として階層事前分布を指定する

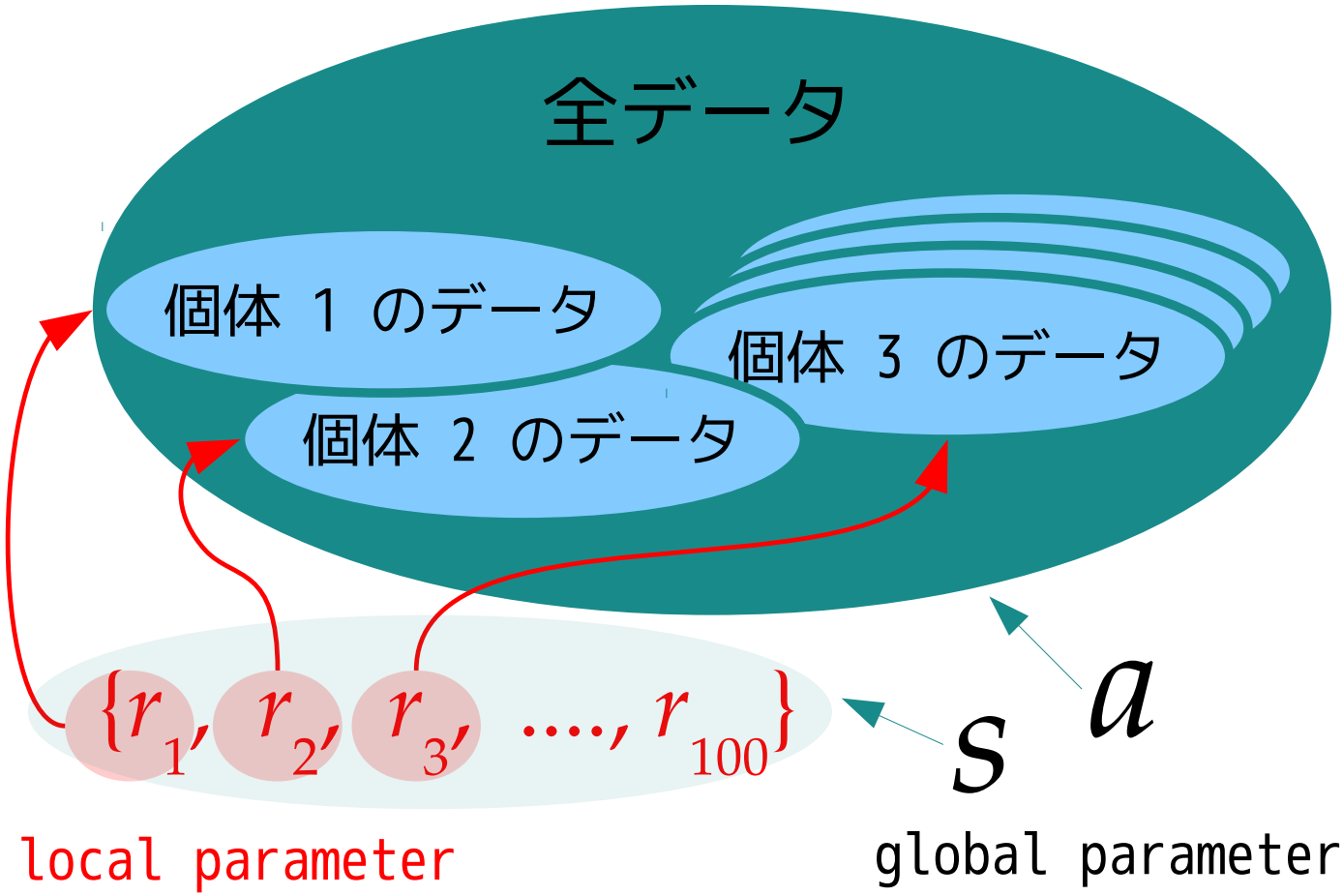
階層事前分布の利点

「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



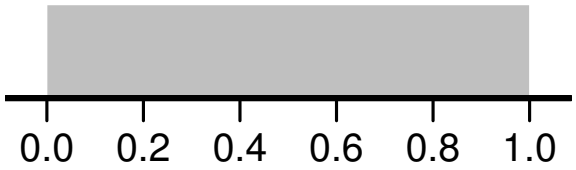
データのどの部分を説明しているのか?

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

$a,\ s$

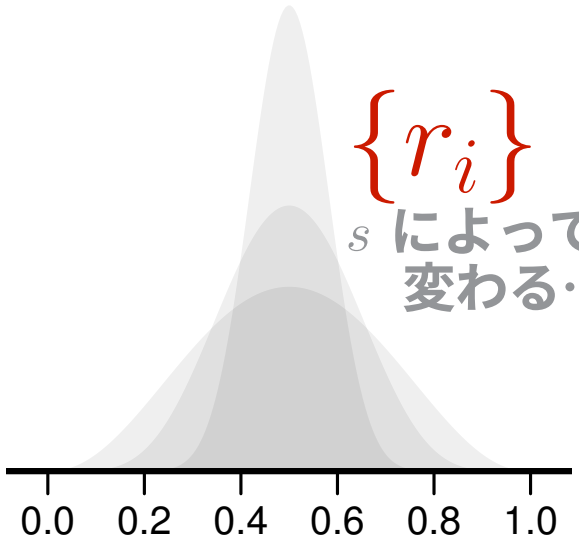
わからない？



(C) 階層事前分布

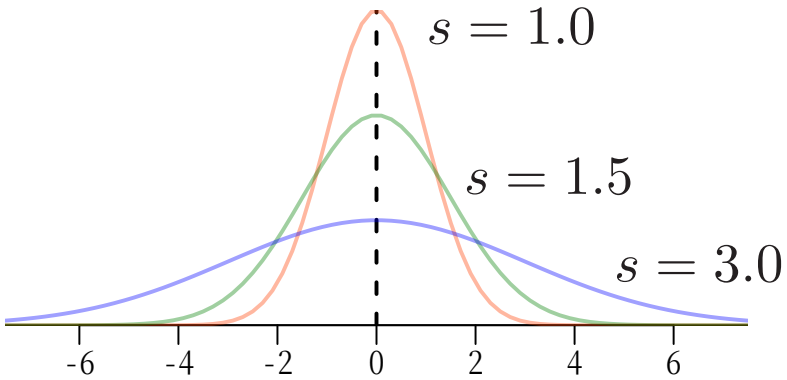
$\{r_i\}$

s によって変わる…



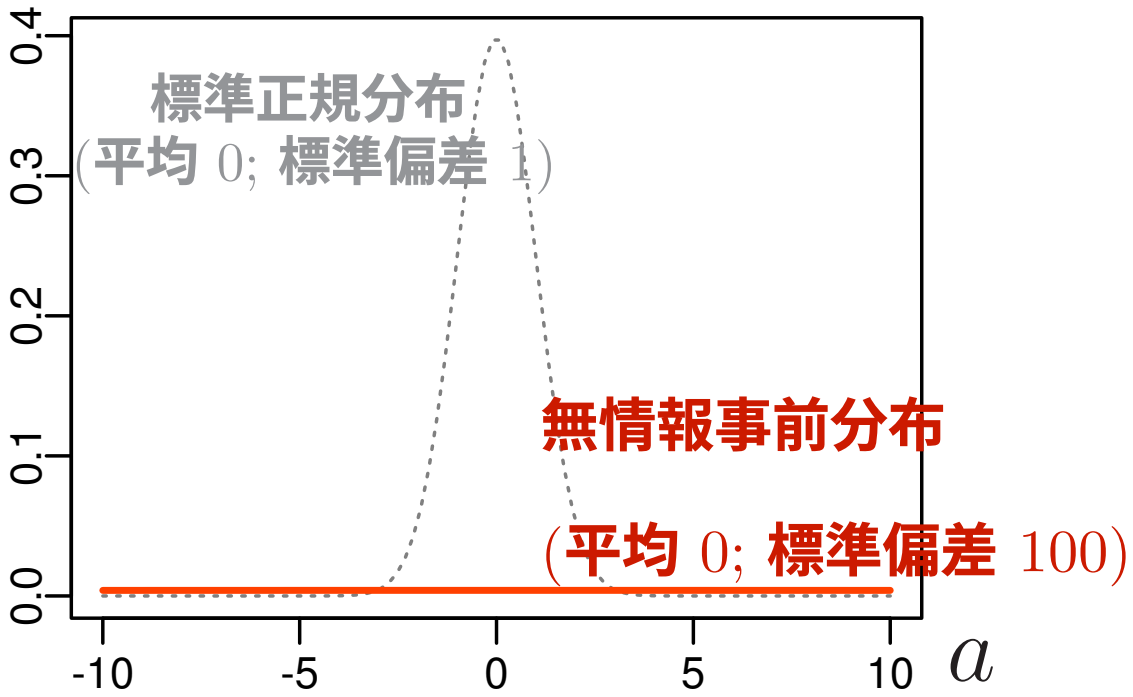
パラメーターの種類	説明する範囲	事前分布
全体に共通する平均・ばらつき	大域的	無情報事前分布
個体・グループごとのずれ	局所的	階層事前分布

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

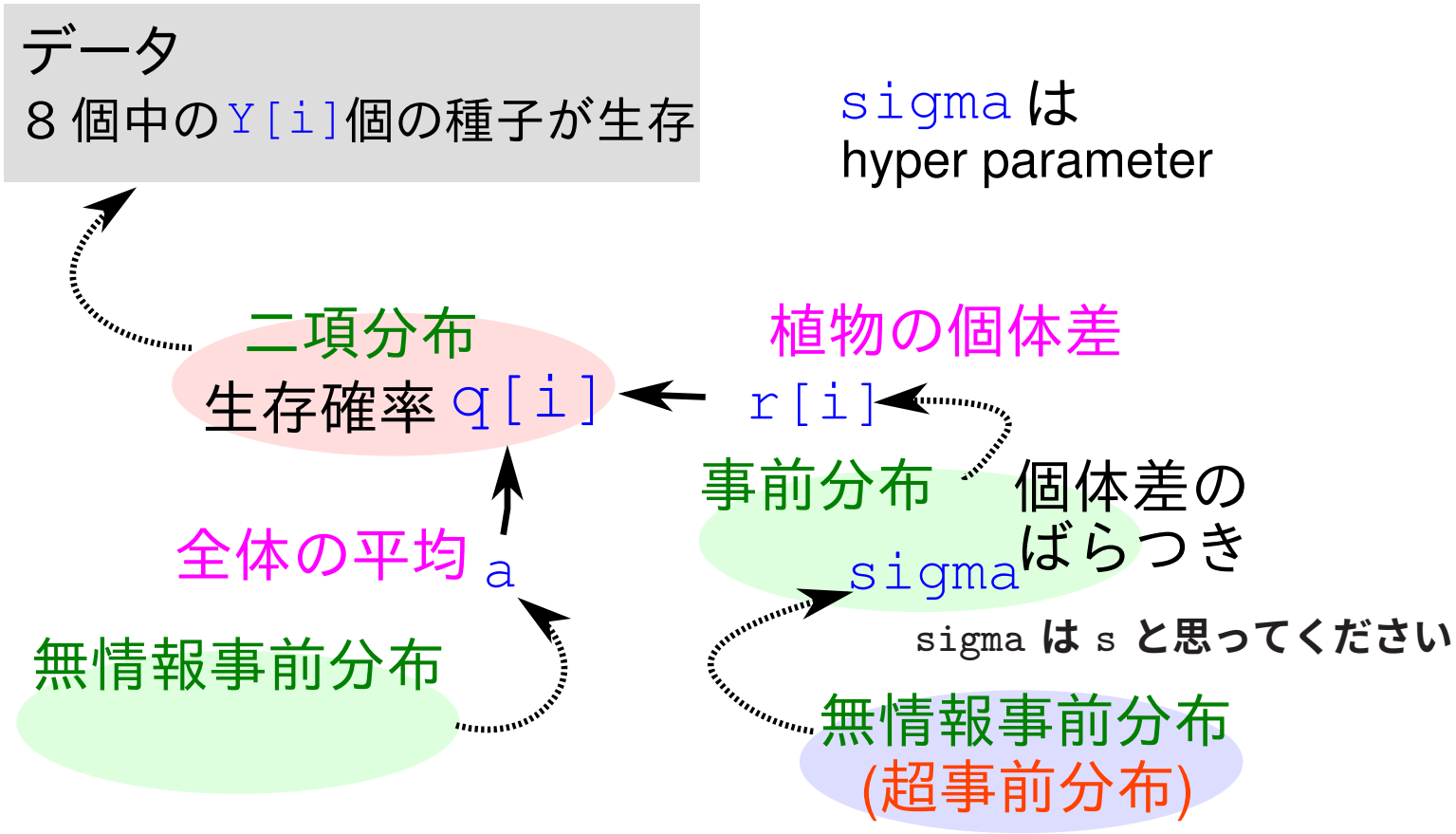
全個体の「切片」 a の無情報事前分布



「生存確率の (logit) 平均 a は何でもよい」と表現している

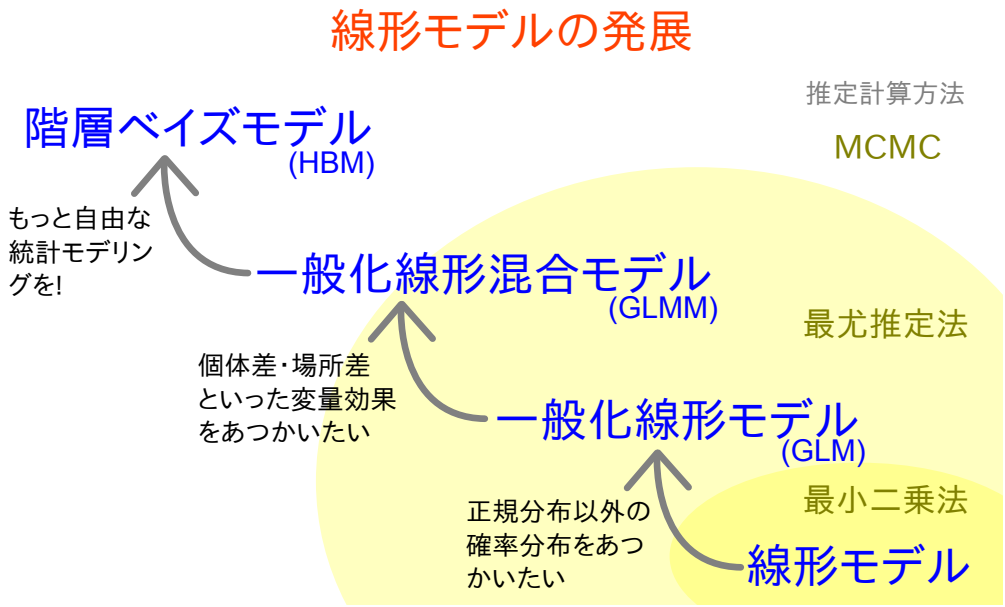
階層ベイズモデル: 事前分布の階層性

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

4. 階層ベイズモデルの推定

ソフトウェア WinBUGS を使ってみる

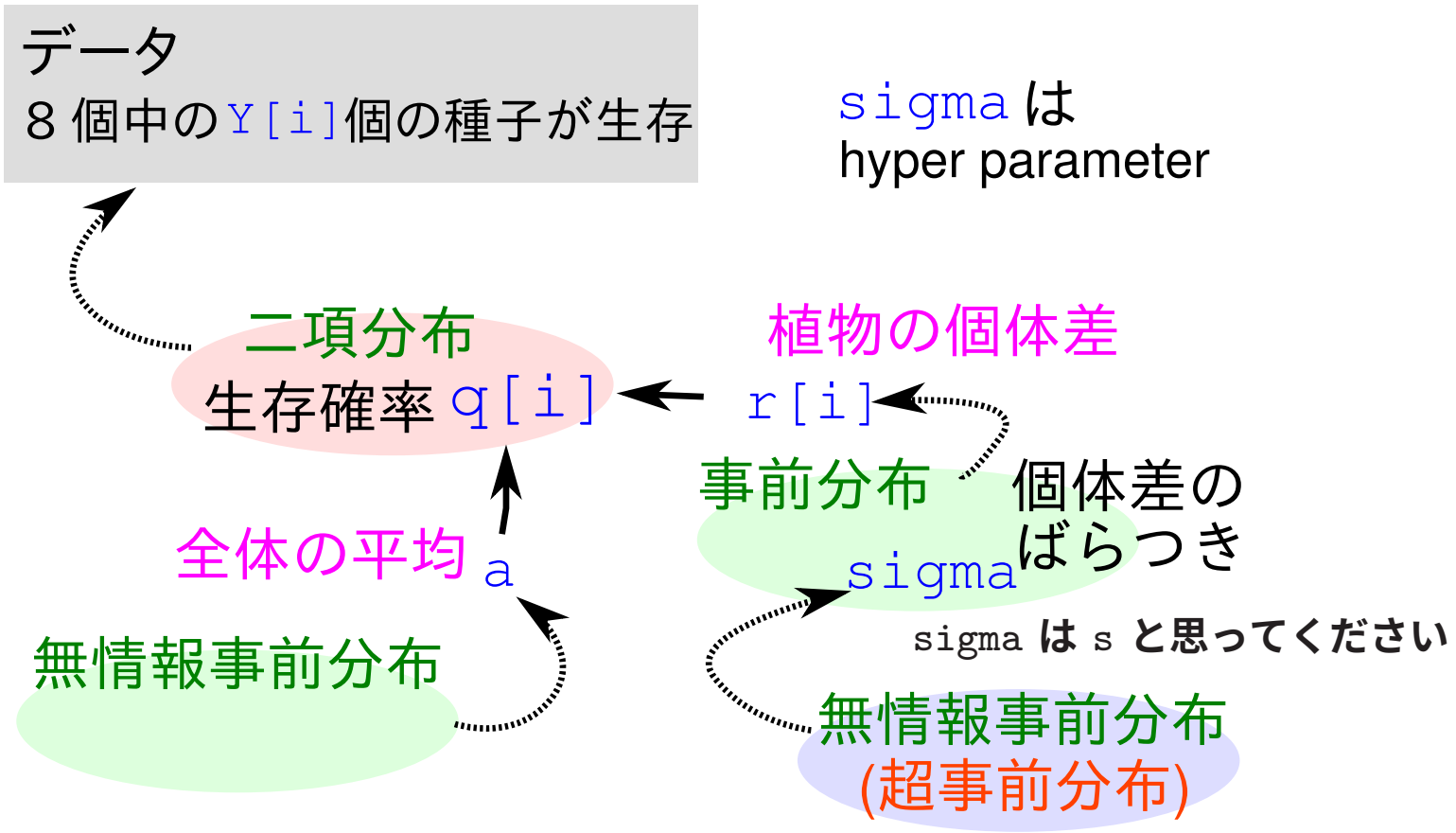
BUGS 言語で統計モデルを指定，R と連携する

便利な “BUGS” 汎用 Gibbs sampler たち

- BUGS 言語 (+ っぽいもの) でベイズモデルを記述できるソフトウェア
  - R 内のいくつかの pacakge (汎用ではない)
  - WinBUGS — よく使われています
  - OpenBUGS — 予算が足りなくて停滞
  - JAGS — じりじりと発展中，がんばってください
  - Stan MC sampler — 期待の新鋭
- リンク集: http://hosho.ees.hokudai.ac.jp/~kubo/ce/BayesianMcmc.html

えーと……BUGS 言語って何?

この階層ベイズモデルを BUGS 言語で記述したい



矢印は手順ではなく，依存関係をあらわしている

BUGS 言語: ベイズモデルを記述する言語

- Spiegelhalter et al. 1995. BUGS: Bayesian Using Gibbs Sampling version 0.50.

```
model { # BUGS コードで定義された階層ベイズモデルの例
    for (i in 1:N.sample) {
        Y[i] ~ dbin(q[i], N[i])
        logit(q[i]) <- a + r[i]
    }
    a ~ dnorm(0, 1.0E-4)
    for (i in 1:N.sample) {
        r[i] ~ dnorm(0, tau)
    }
    tau <- 1 / (s * s)
    s ~ dunif(0, 1.0E+4)
}
```

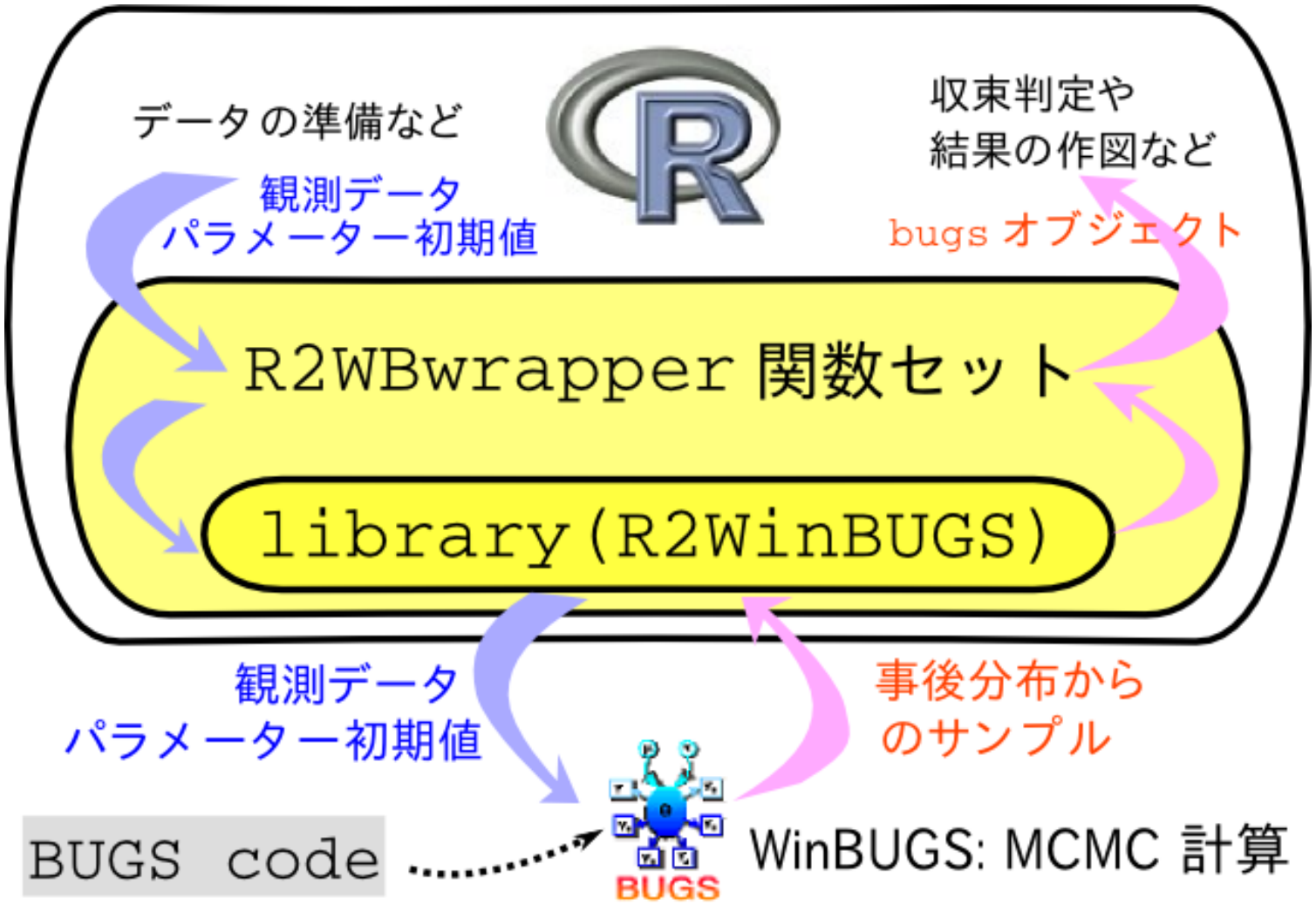
なんとなく使われ続けている WinBUGS 1.4.3

- おそらく世界でもっともよく使われている Gibbs sampler
- BUGS 言語の実装
- 2004-09-13 に最新版 (ここで開発停止 → OpenBUGS)
- ソースなど非公開，無料，ユーザー登録不要
- Windows バイナリーとして配布されている
  - Linux 上では WINE 上で動作
  - MacOS X 上でも Darwine など駆使すると動くらしい
- ヘンな GUI (Linux ユーザーの偏見)
- R ユーザーにとっては R2WinBUGS が快適 (後述)

今回説明する WinBUGS の使いかた (概要)

- WinBUGS を R から使う
  - R から WinBUGS をよびだし「このベイズモデルのパラメーターの事後分布をこういうふうに MCMC 計算してね」と指示する
  - WinBUGS が得た事後分布からのサンプルセットを R がうけとる
- R の中では library(R2WinBUGS) package を使う R2WBwrapper 関数 (久保作) を使う

概要: R2WBwrapper 経由で WinBUGS を使う



なんで WinBUGS を R 経由で使うの?

- WinBUGS のユーザーインターフェイスを使うのが**めんどう**だから
- どうせ解析に使うデータは R で準備するから
- どうせ得られた出力は R で解析・作図するから
- R には R2WinBUGS という (機能拡張用) package があって，R から WinBUGS を使うしくみが準備されてるから
  - R 上で `install.packages("R2WinBUGS")` でインストールできる

なんで R2WinBUGS をラップして使うの?

- R2WinBUGS 直接利用がめんどうだから
  - モデルをちょっと変更したらあちこち書きなおさないといけない
  - R2WBwrapper を使うとそのあたりがかなりマシになる
- Linux と Windows で「呼びだし」方法がびみょーに異なるため
  - R2WBwrapper を使うと自動的に OS にあわせた WinBUGS よびだしをする

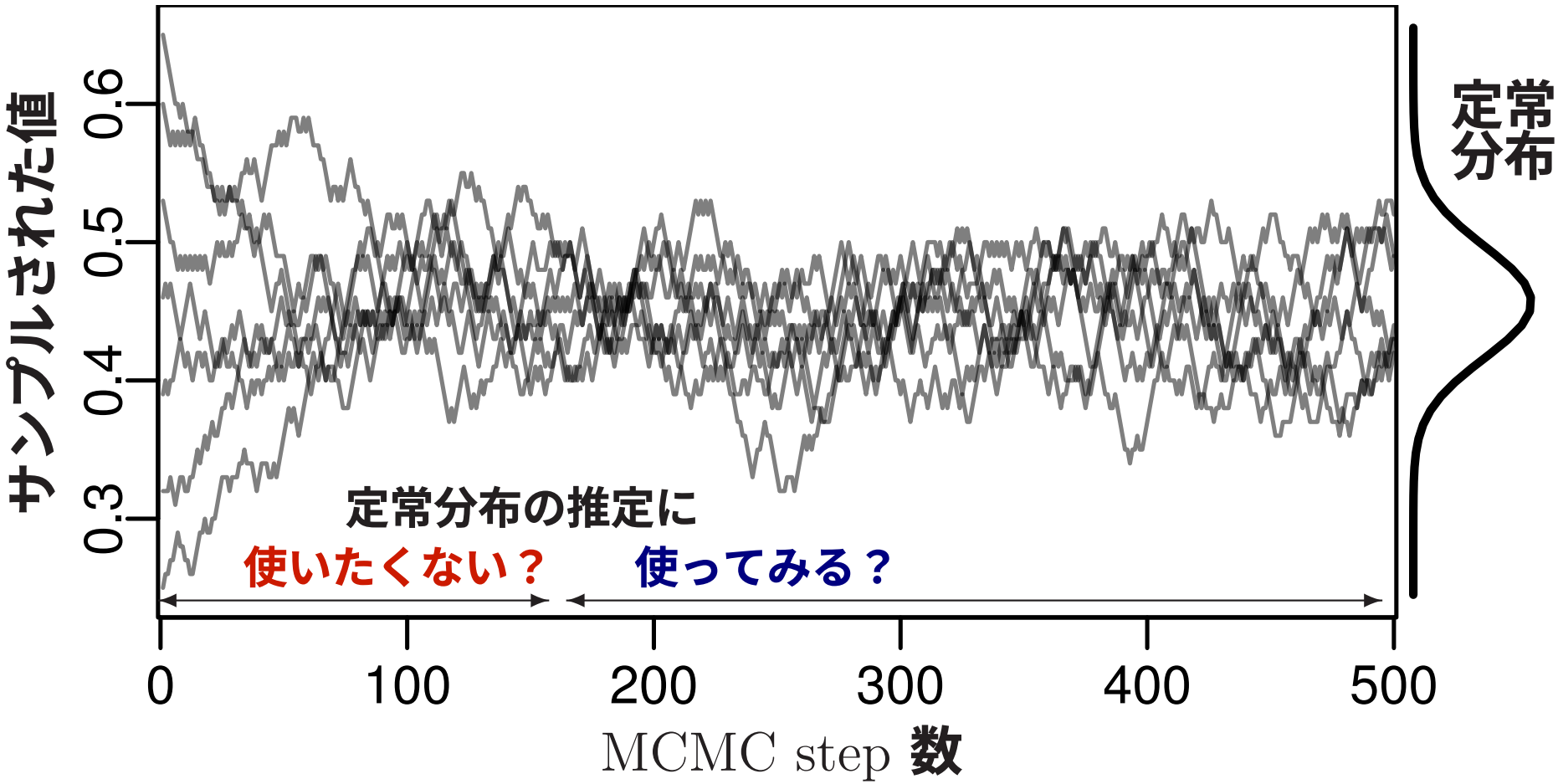
R2WBwrapper 経由で WinBUGS を使う

1. BUGS 言語でかかれた model ファイルを準備する
2. R2WBwrapper 関数を使う R コードを書く
3. R 上で 2. を実行
4. 出力された結果が bugs オブジェクトで返される
5. これを plot() したり summary() したり……オブジェクトに変換して，いろいろ事後分布の図なんかを描いてみたり……

WinBUGS にこんなかんじで仕事を命じる

```
source("http://goo.gl/Y41N8J") # or source("R2WBwrapper.R")
d <- read.csv("data7a.csv")
clear.data.param()
set.data("N", nrow(d))
set.data("Y", d$y)
set.param("a", 0)
set.param("r", rnorm(N, 0, 0.1))
set.param("s", 1)
post.bugs <- call.bugs(
    file = "model.bug",
    n.chains = 3, # 収束診断のため独立試行 3 回
    n.iter = 10100, n.burnin = 100, n.thin = 10
)
```

burn in って何? → 「使いたくない」長さの指定



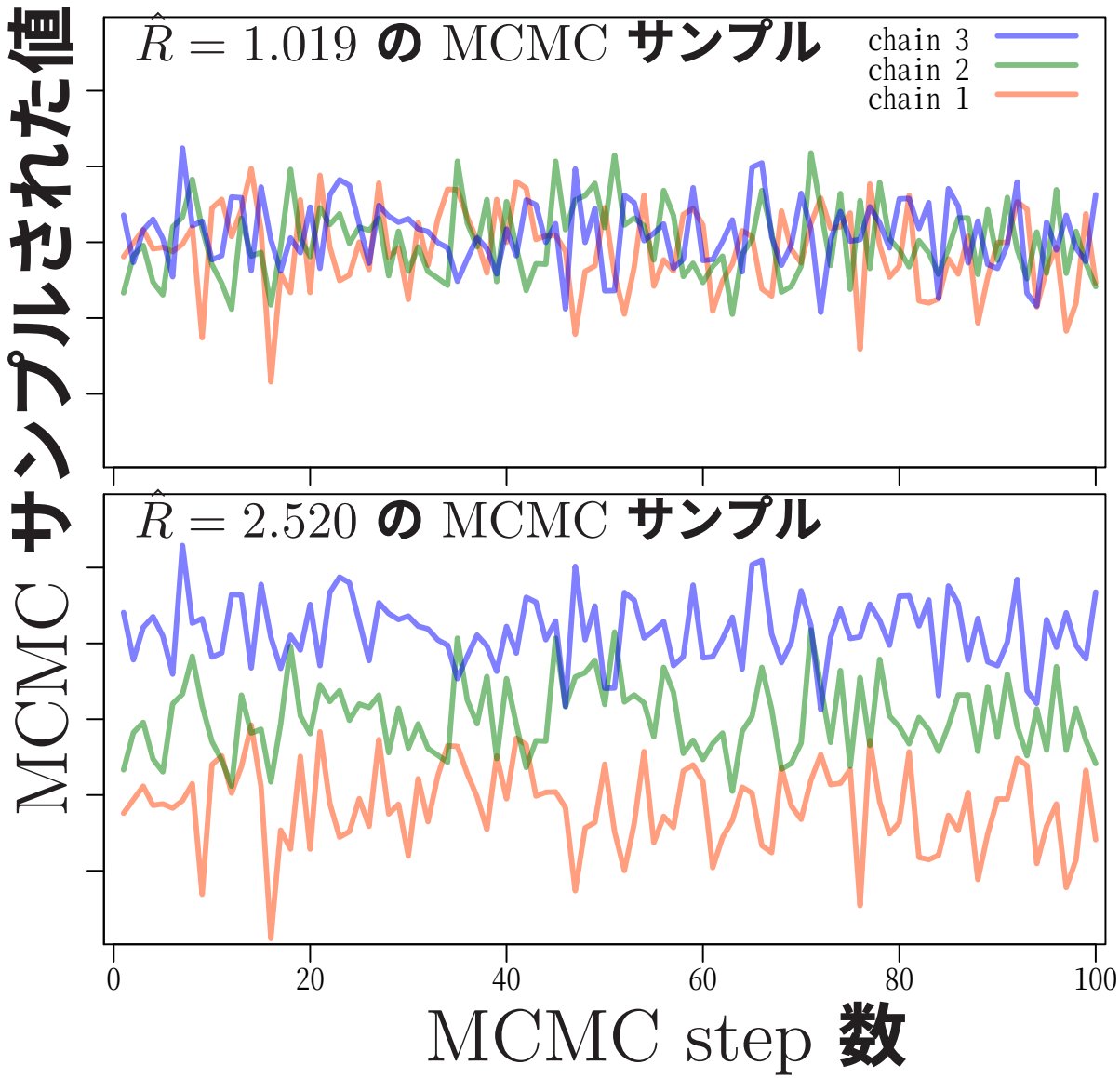
事後分布サンプルが得られたら → あれこれ図表を

- `plot(post.bugs)` — 「収束診断」とか
- `pg(post.bugs)` — 事後分布の table
- `pl("a", post.bugs)` — `a` の図示
- などなど

「収束診断」の $\hat{R}$ 指数

- `plot(post.bugs)` → 次のページ, 実演表示
- `R-hat` は Gelman-Rubin の収束判定用の指数
  - $\hat{R} = \sqrt{\dfrac{\hat{\mathrm{var}}^{+}(\psi|y)}{W}}$
  - $\hat{\mathrm{var}}^{+}(\psi|y) = \dfrac{n-1}{n}W + \dfrac{1}{n}B$
  - W : サンプル列内の variance の平均
  - B : サンプル列間の variance
  - Gelman et al. 2004. Bayesian Data Analysis. Chapman & Hall/CRC

試行間で差がないかを「診断」する

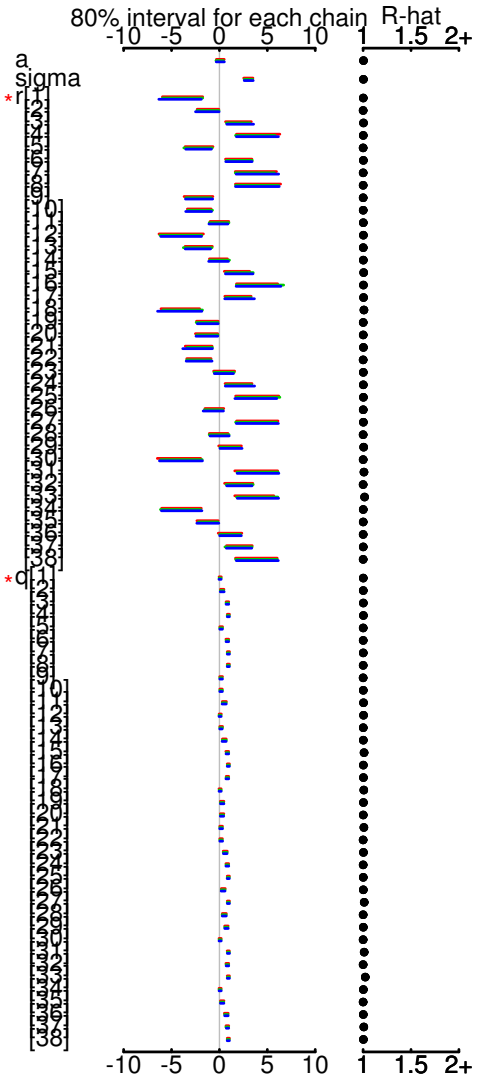


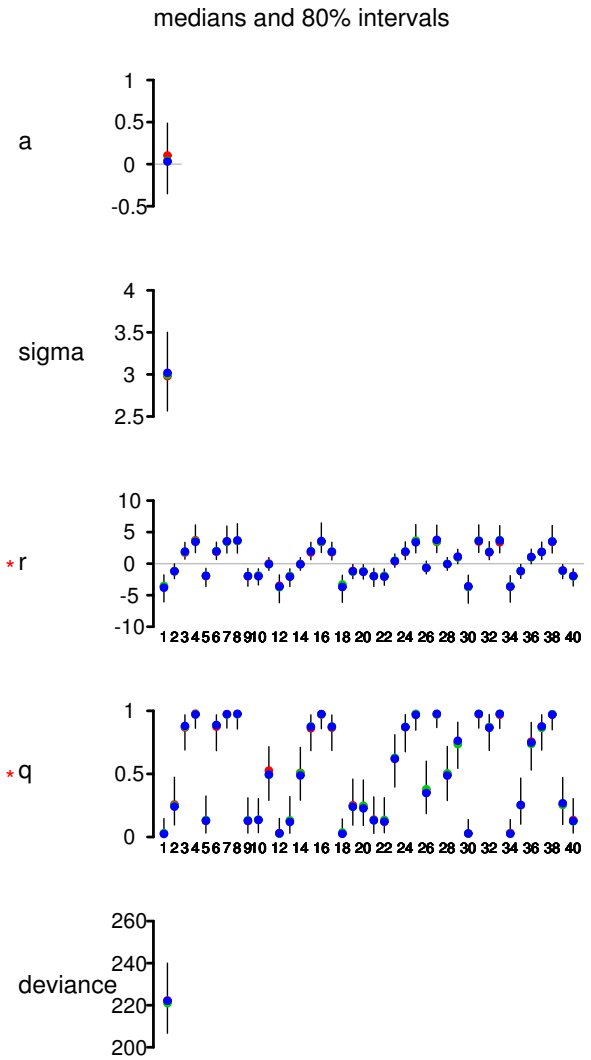
まあ，
いいかな……

何やら
問題あり!

WinBUGS で得られた事後分布サンプルの要約

/kubo/public_html/stat/2010/ism/winbugs/model.bug.txt", fit using WinBUGS, 3 chains, each with 1300 iteratior





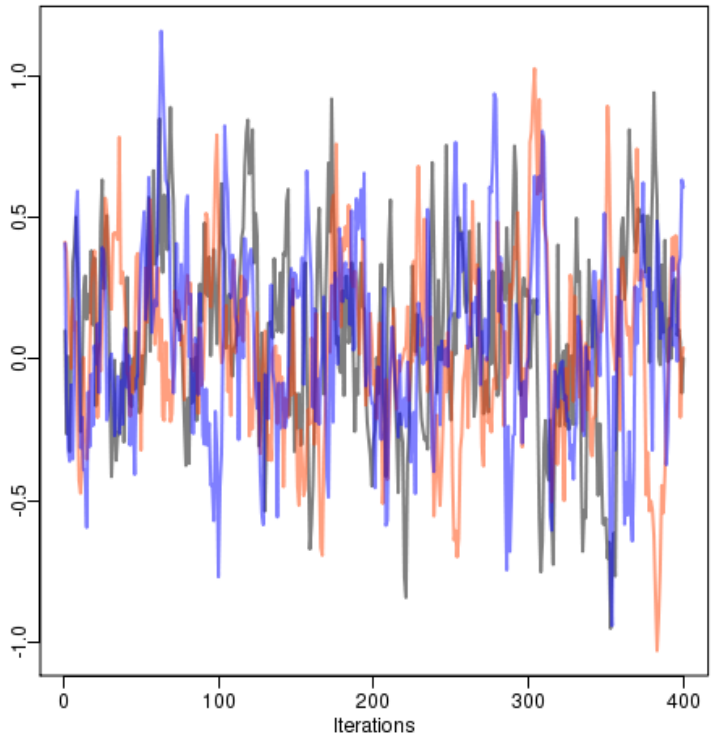
bugs オブジェクトの post.bugs を調べる

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

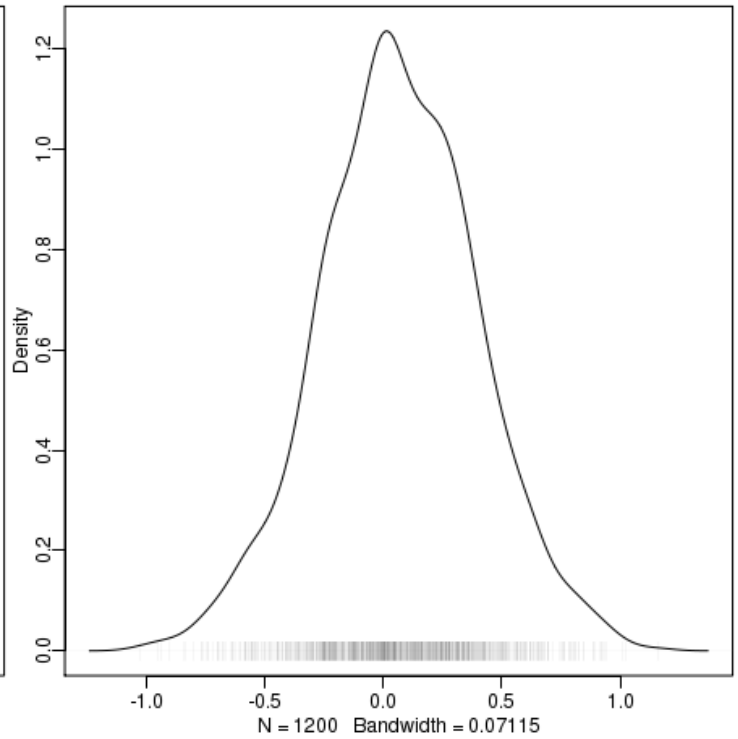
	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
a	0.031	0.357	-0.718	-0.187	0.041	0.268	0.682	1.034	72
sigma	3.060	0.376	2.365	2.807	3.029	3.288	3.830	1.002	1200
r[1]	-3.890	1.903	-8.238	-4.918	-3.514	-2.546	-1.174	1.001	1200
r[2]	-1.190	0.905	-3.137	-1.763	-1.159	-0.559	0.438	1.007	290
r[3]	2.062	1.128	0.185	1.296	1.931	2.730	4.611	1.002	1200
r[4]	3.985	1.860	1.058	2.635	3.745	5.105	8.520	1.021	130
r[5]	-2.049	1.077	-4.458	-2.679	-1.971	-1.276	-0.255	1.008	270
r[6]	1.995	1.061	0.137	1.266	1.922	2.629	4.300	1.002	900
r[7]	3.886	1.765	1.144	2.664	3.583	4.894	8.223	1.008	320
r[8]	3.862	1.763	1.142	2.590	3.591	4.814	7.993	1.011	330
r[9]	-2.093	1.136	-4.532	-2.788	-1.978	-1.313	-0.130	1.003	540
r[10]	-1.993	1.082	-4.358	-2.631	-1.905	-1.250	-0.158	1.000	1200
r[11]	-0.049	0.786	-1.654	-0.555	-0.032	0.466	1.462	1.006	320
r[12]	-3.849	1.788	-8.204	-4.874	-3.547	-2.598	-1.144	1.001	1200
r[13]	-2.005	1.115	-4.593	-2.640	-1.908	-1.254	-0.069	1.001	1200

各パラメーターの事後分布サンプルを R で調べる

a のサンプリングの様子

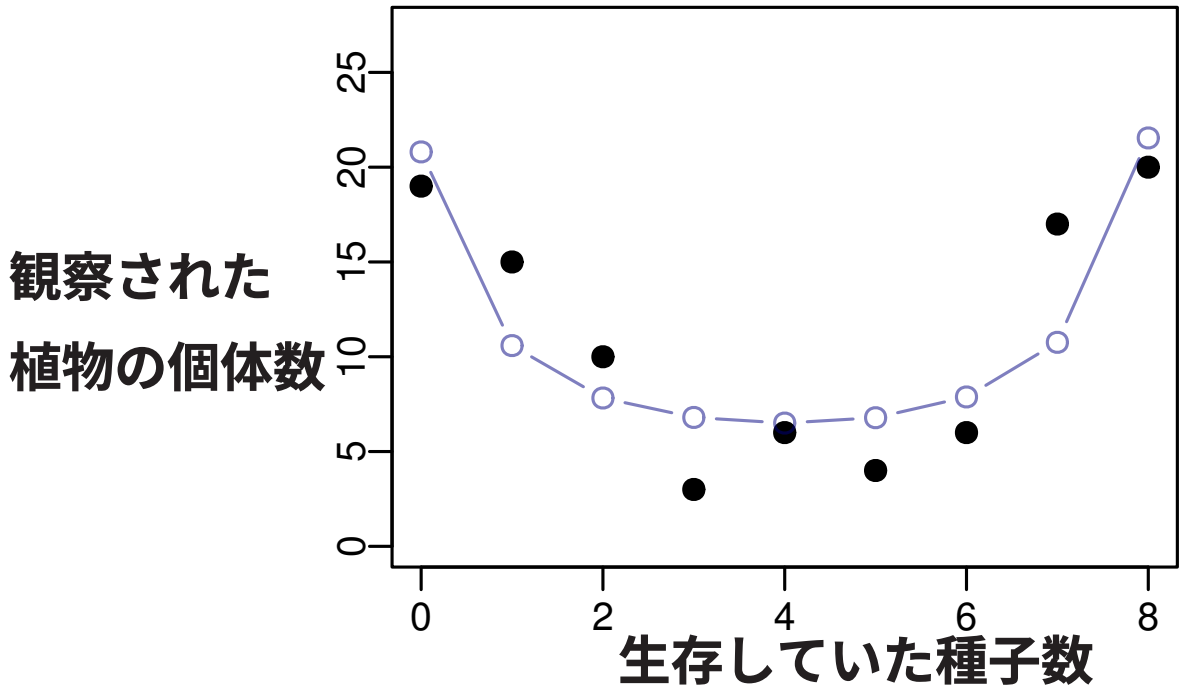


a の事後確率密度の推定



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**



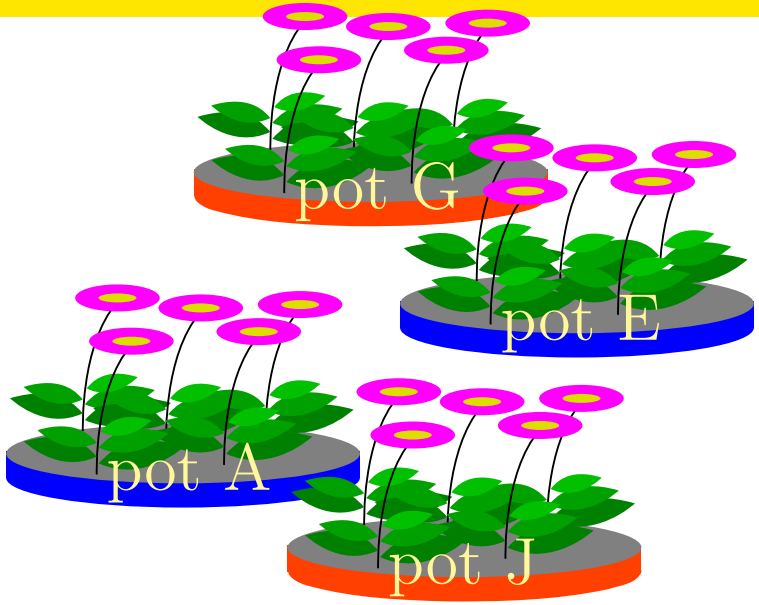
時間があれば個体差＋場所差の例を紹介

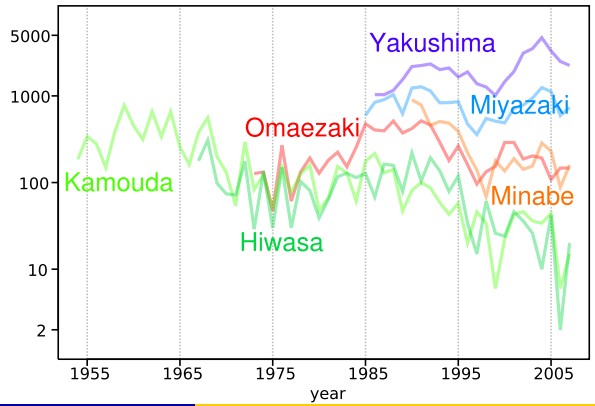
5. 階層ベイズモデルの応用

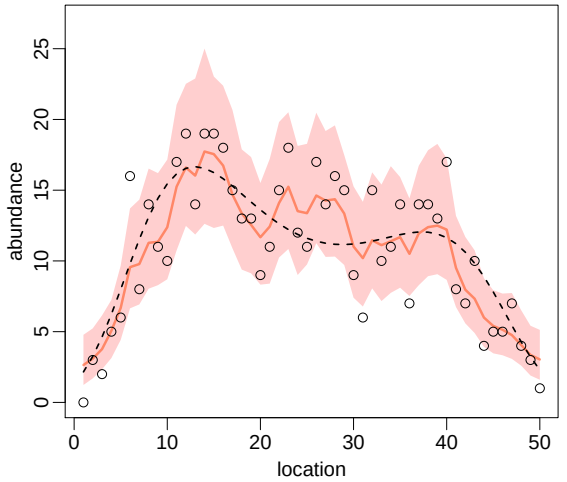
複数の階層，時間や空間の構造

いろいろな階層ベイズモデル

1. 個体差 + ブロック差というネストしたランダム効果
2. 「隣と似ている」空間相関のあるランダム効果
3. 時間変化する潜在変数: ウミガメ上陸数の統計モデル



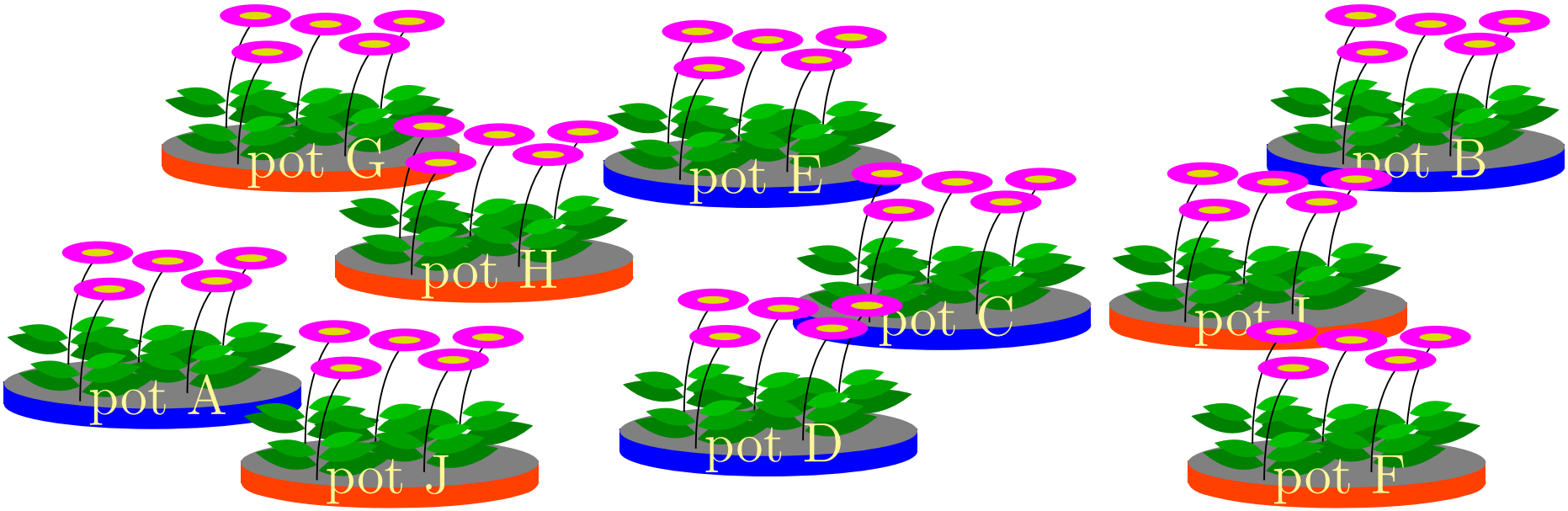




複数ランダム効果の階層ベイズモデル

— 個体差 + 場所差 —

架空植物の例題: またまた種子数データ



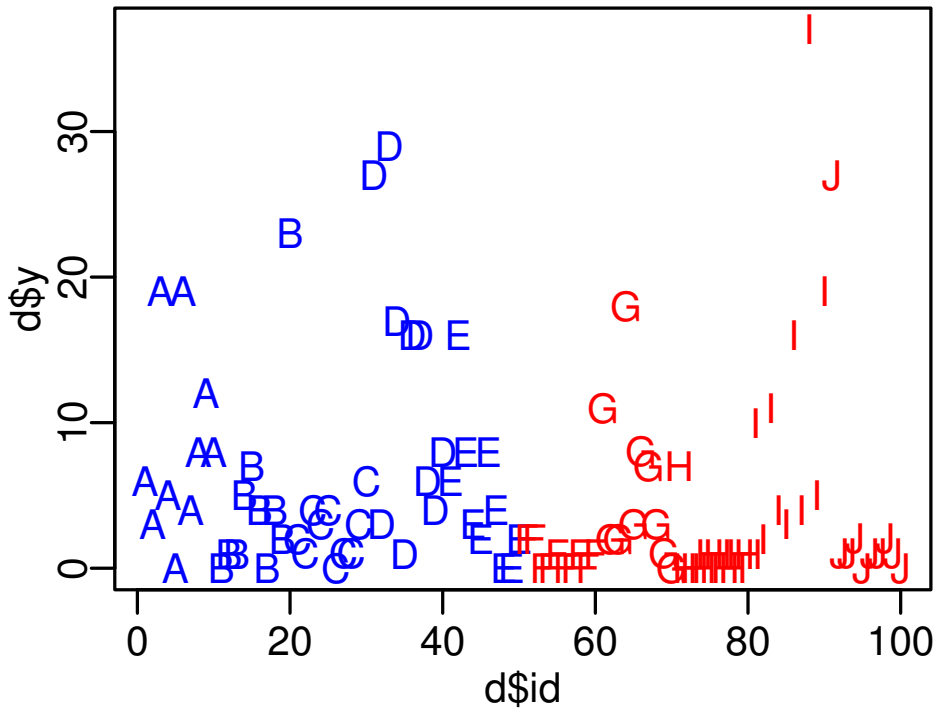
- 肥料をやったら個体ごとの種子数 y_i が増えるかどうかを調べたい
- 植木鉢 10 個，各鉢に 10 個体の架空植物 (合計 100 個体)
  - コントロール ($f_j = \mathbf{C}$) 5 鉢 (合計 50 個体)
  - 肥料をやる処理 ($f_j = \mathbf{T}$) 5 鉢 (合計 50 個体)

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
```

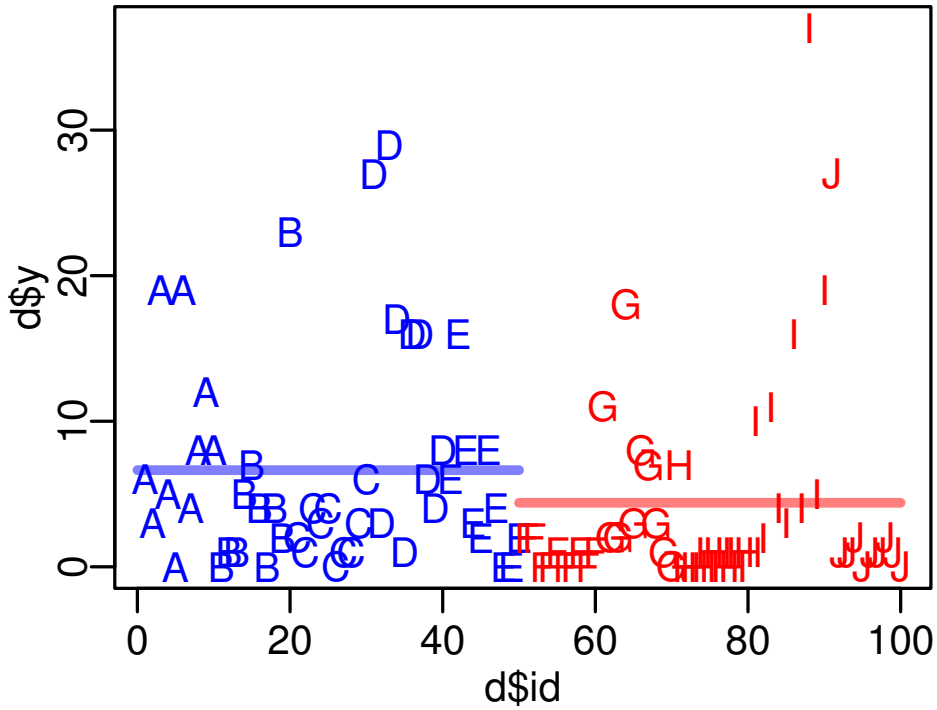
- id **列**: **個体番号** $\{1, 2, 3, \cdots, 100\}$
- pot **列**: **植木鉢名** $\{\mathrm{A}, \mathrm{B}, \mathrm{C}, \cdots, \mathrm{J}\}$
- f **列**: **処理**: **コントロール** C, **肥料** T
- y **列**: **種子数** (**応答変数**)

データはとにかく図示する!!



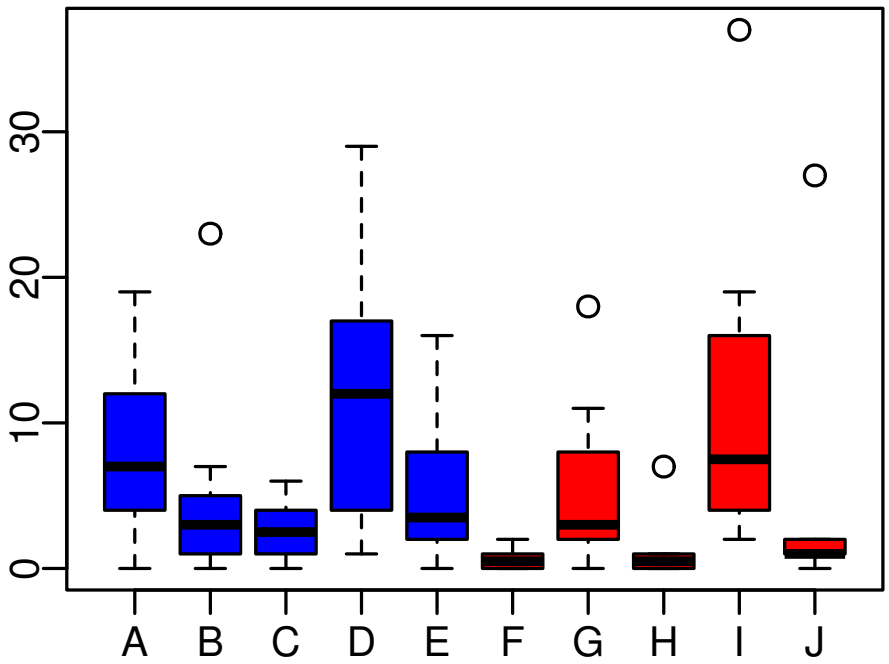
- `plot(d$id, d$y, pch = as.character(d$pot), ...)`
- **コントロール・処理 でそんなに差がない**?

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?

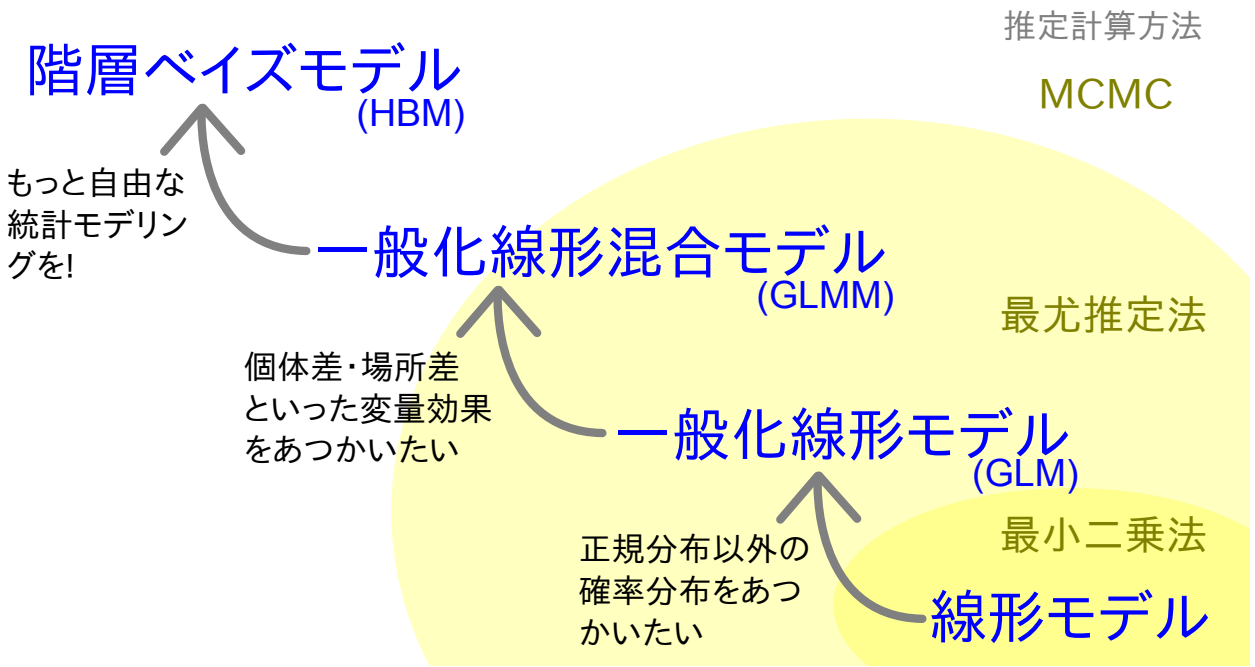
個体差だけでなく植木鉢差もありそう?



- `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

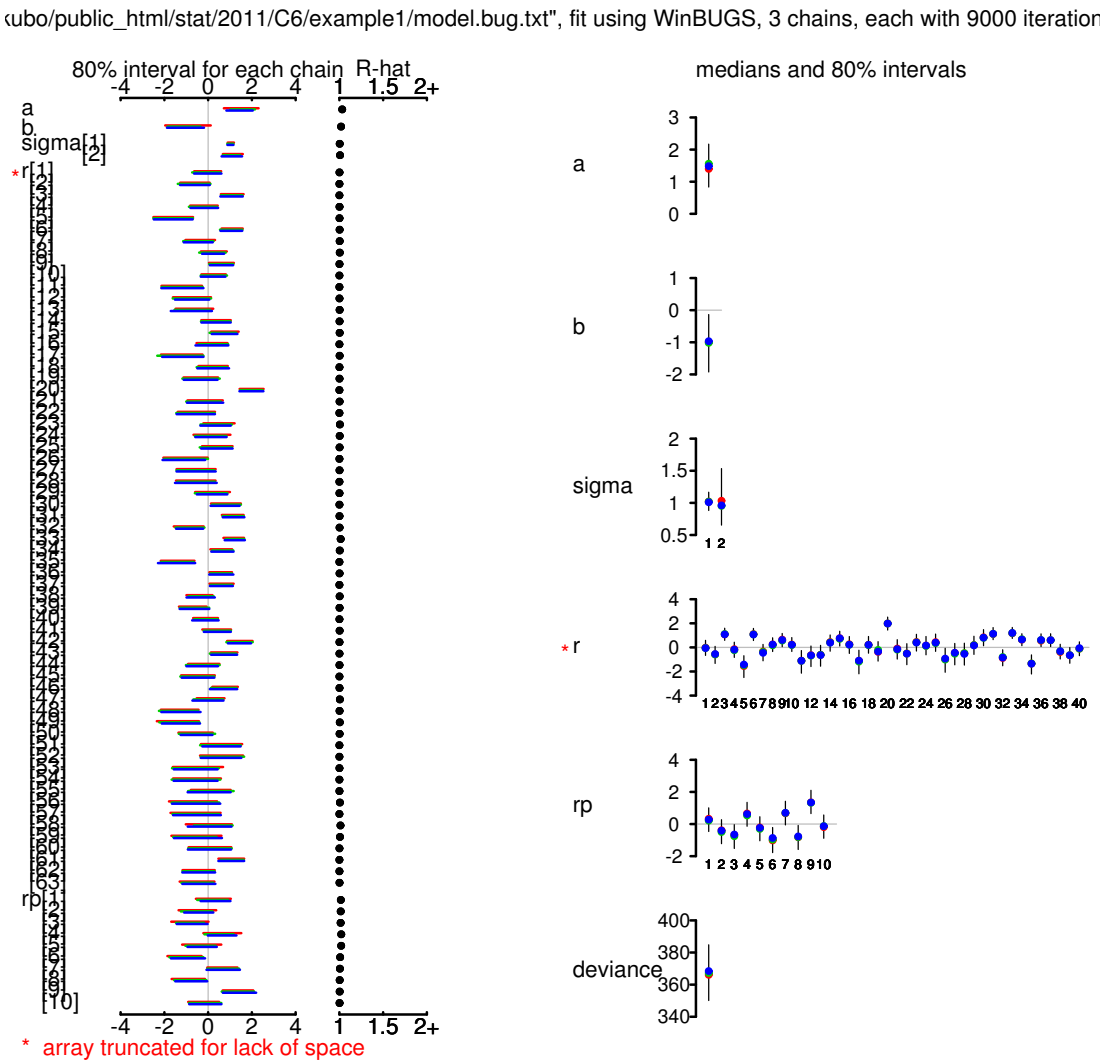
```
> library(glmmML)
> summary(glmmML(y ~ f, data = d, family = poisson,
+ cluster = id))
...(略)...
              coef se(coef)     z Pr(>|z|)
(Intercept)  1.351    0.192  7.05  1.8e-12
fT          -0.737    0.280 -2.63  8.4e-03
...(略)...
```

- やっぱり同じ?
- むしろ肥料処理の悪影響が強い?

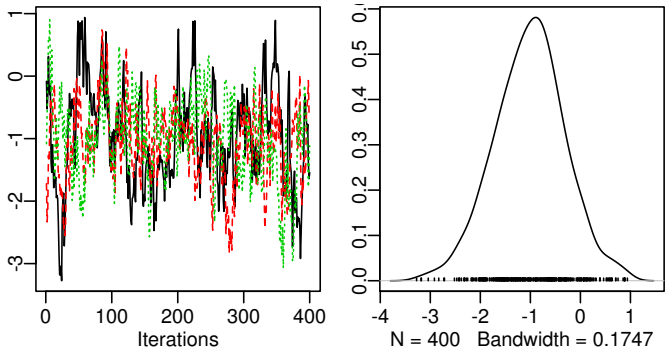
個体差 + ブロック差を考える階層ベイズモデル

- 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)

WinBUGS による事後分布の推定，R で収束判定



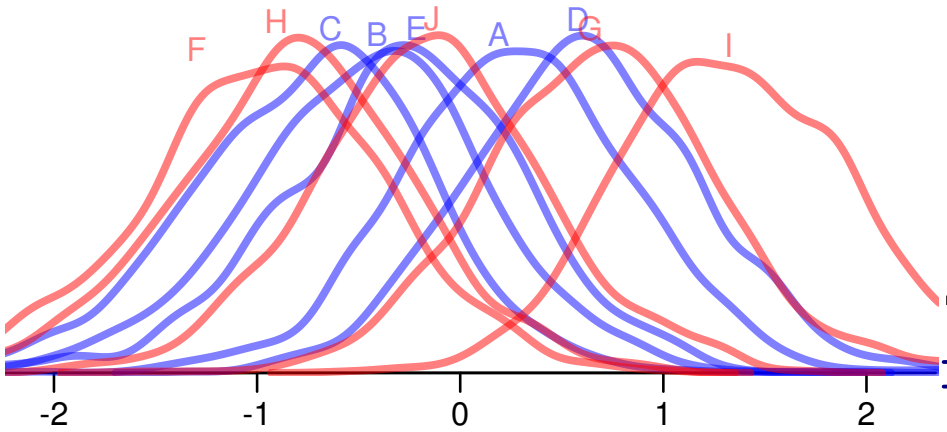
肥料の効果 (パラメーター b) はなさそう?



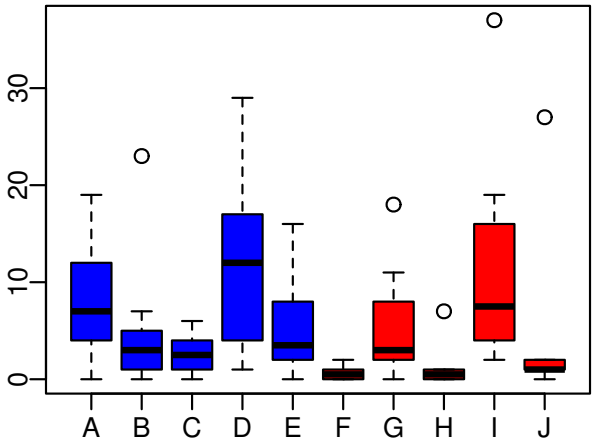

```
> print(post.bugs, digits.summary = 3)
...(略)...
             mean     sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
a           1.501  0.529   0.482   1.157   1.493   1.852   2.565 1.032   240
b          -1.016  0.706  -2.436  -1.476  -0.993  -0.565   0.395 1.019   450
sigma[1]    1.020  0.114   0.822   0.939   1.014   1.089   1.265 1.004   510
...(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]



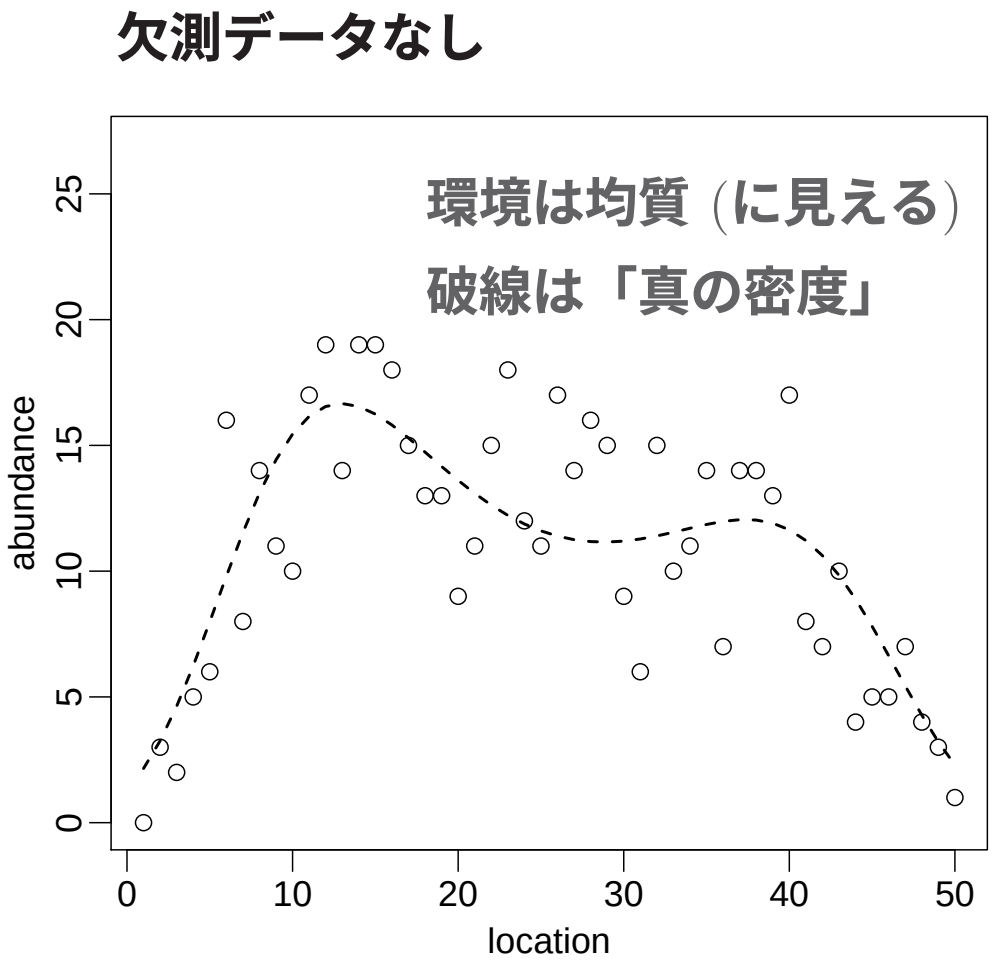
統計モデリングの手ぬきは危険!

- 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

空間構造のある階層ベイズモデル

— 一次元の架空データ —

架空の例題: 個体数データ，一次元空間データ

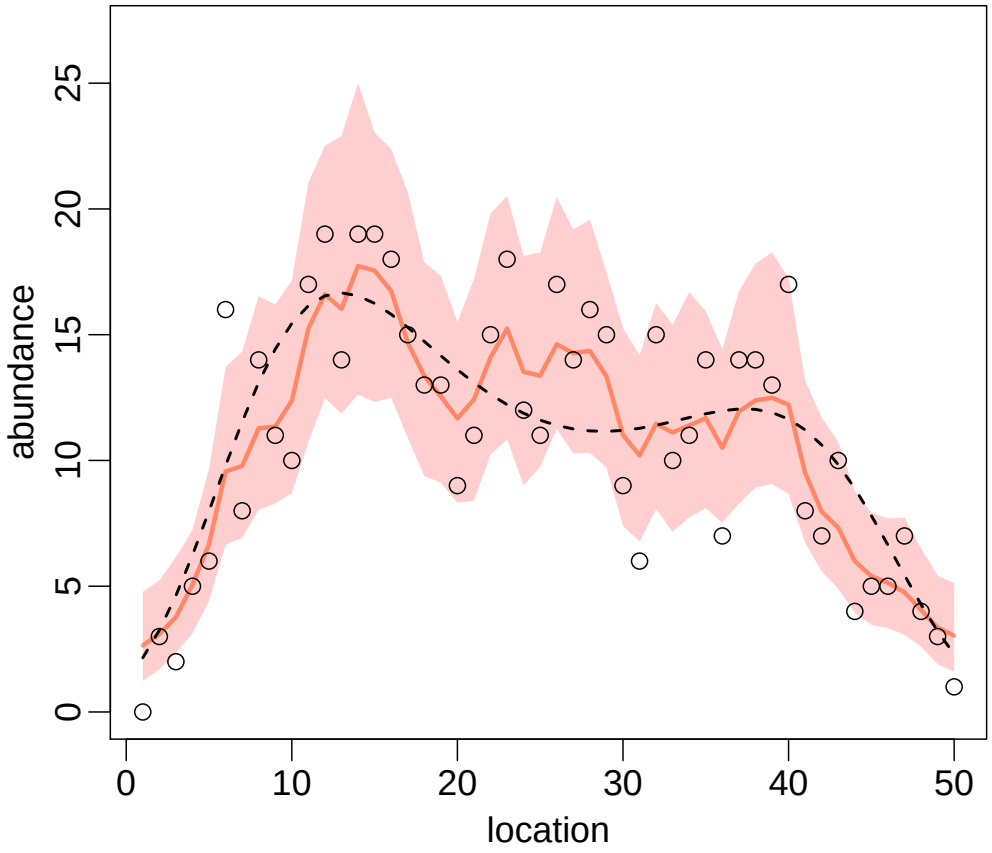


問: 空間自己相関を考慮して生物個体の密度推定

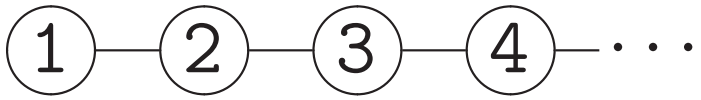
解析の目的: まずはこんな推定をしてみたい

空間相関を考慮するモデル

欠測データなし



空間相関のある「場所差」階層ベイズモデル



- 地点 i の観測個体数は平均 λ_i のポアソン分布にしたがう:

  $y_i \sim \mathrm{Poisson}(\lambda_i)$

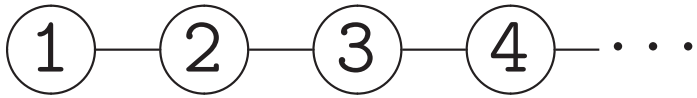
- 平均 λ_i の対数は (全体の平均) + (場所差) と分割する:

  $\log \lambda_i = \beta + r_i$

- ベイズモデルとしてあつかいたいので，推定したいパラメーターの**事前分布** を決めてやらなければならない
  - **事前分布** についてはあとで説明

- 全体の平均 β は無情報事前分布にしたがう: $\beta \sim \mathrm{Normal}(0, 10^2)$,

空間相関のある「場所差」階層ベイズモデル (続)



- Conditional Autoregressive (CAR) **モデルにおける場所差** r_i **の条件つき事前分布** (N_i **は** i **の近傍場所数**, $\mathbf{J}_i$ **は** i **の近傍場所**):

$$r_i \sim \text{Normal}(\frac{\sum_{j \in \mathbf{J}_i} r_j}{N_i}, \frac{\sigma}{N_i})$$

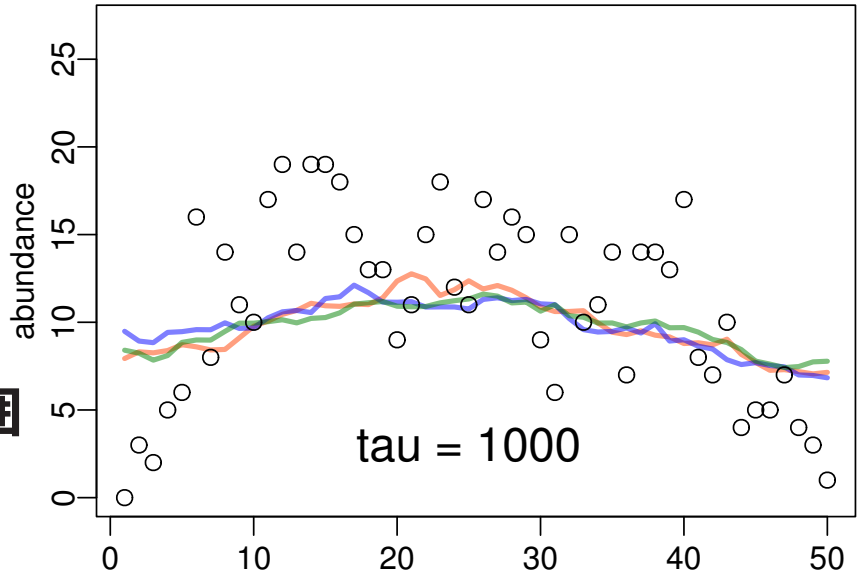
- σ **は無情報事前分布にしたがう**: $\tau = 1/\sigma^2 \sim \text{Gamma}(1.0^{-2}, 1.0^{-2})$
- **ベイズの定理** $\rightarrow$ **事後分布の導出**

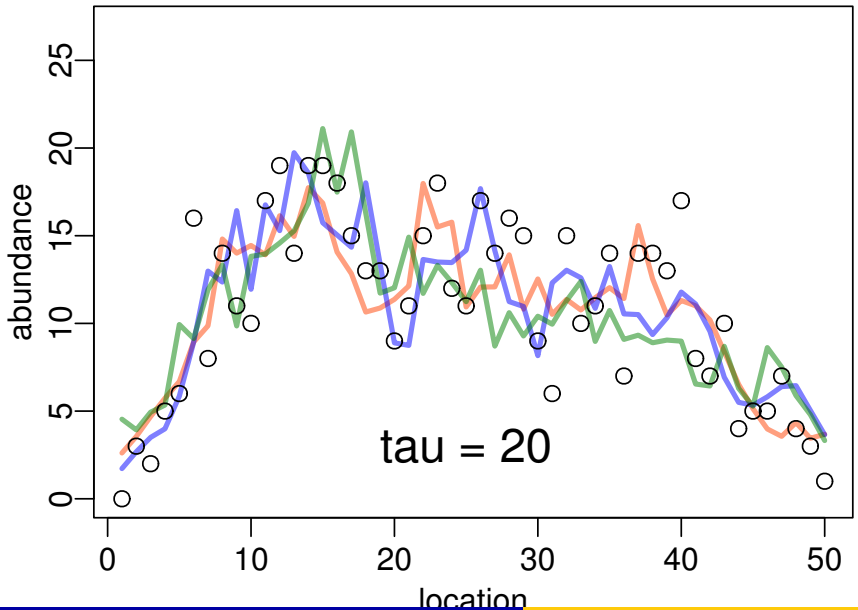
$$p(\beta, \{r_i\}, \tau \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid \beta, \{r_i\}, \tau) \times (\textbf{事前分布あれこれ})}{\int\int \cdots \int (\uparrow \textbf{分子}) d\beta dr_1 \cdots dr_{50} d\tau}$$

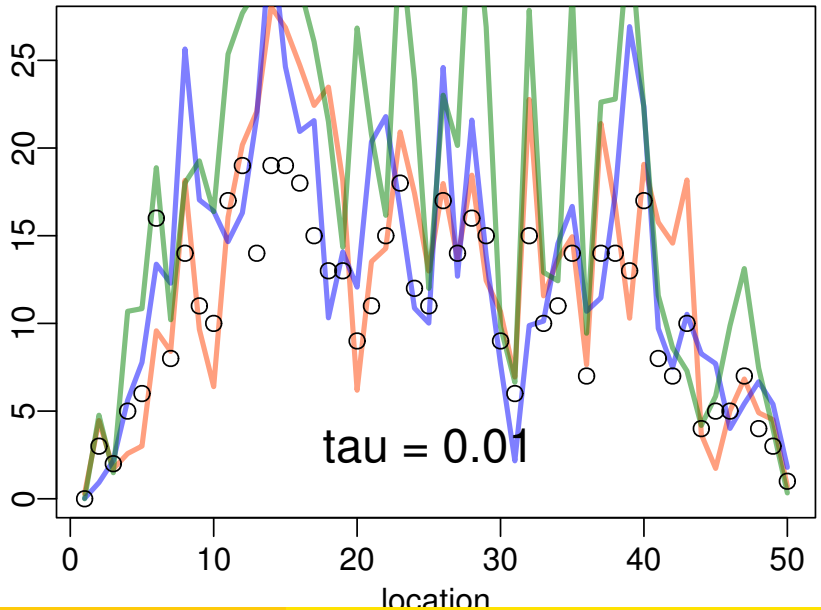
超パラメーター τ が決める

隣との類似ぐあい

- τ が大 (σ が小) だと隣と似ている
- τ が小 (σ が大) だと隣と似てない
- ベイズ推定によって適切な τ の範囲 (**事後分布**) が得られる





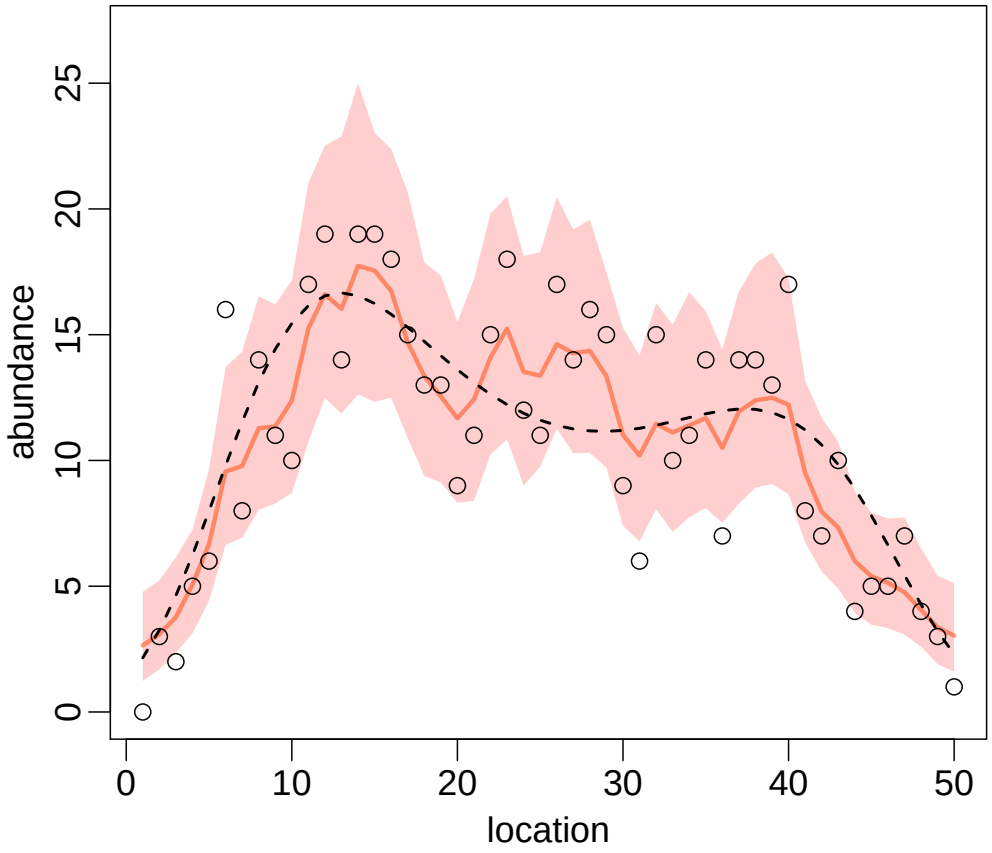


BUGS 言語: ベイズモデルを記述する言語

```
model { # BUGS コードで定義された階層ベイズモデルの例
  for (i in 1:N.site) {
    Y[i] ~ dpois(mean[i])           # 観測データと密度の関係
    log(mean[i]) <- beta + re[i]    # (全体の平均) + (場所差)
  }
  # 場所差 re[i] を CAR model で生成
  re[1:N.site] ~ car.normal(Adj[], Weights[], Num[], tau)
  beta ~ dnorm(0, 1.0E-2)           # 全体の平均は無情報事前分布
  tau ~ dgamma(1.0E-2, 1.0E-2) # 場所差のばらつきは無情報事前分布
}
```

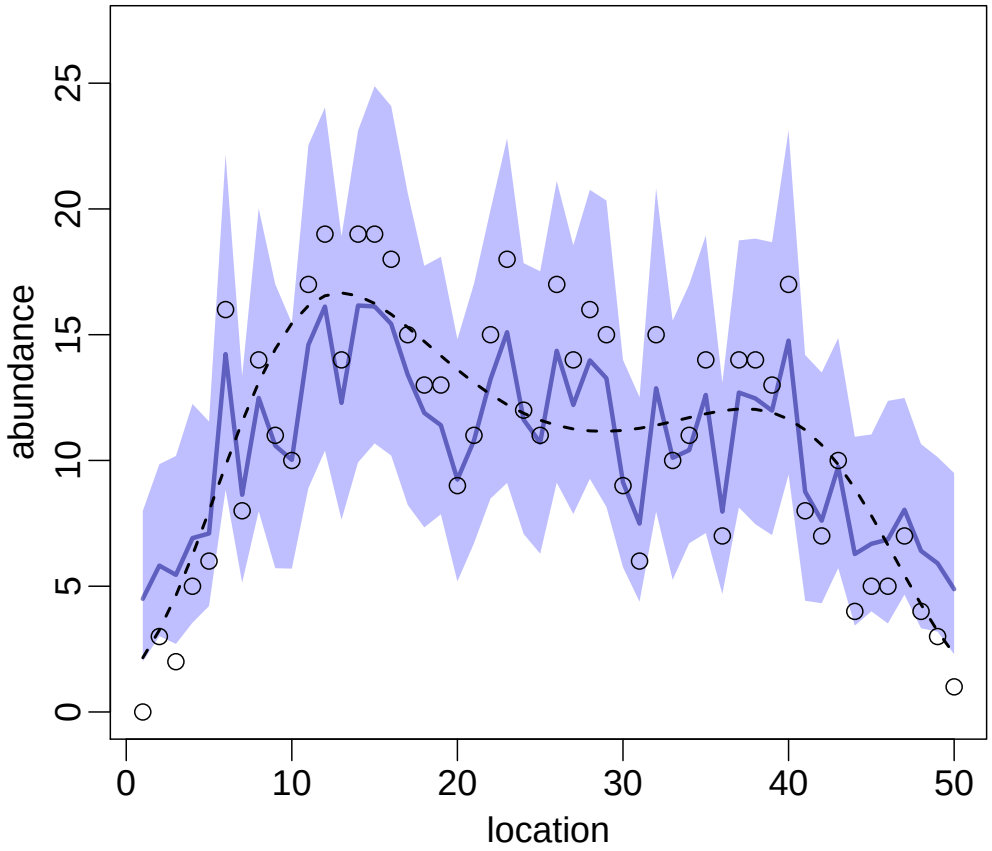
空間相関のある「場所差」モデルの推定結果

空間相関を考慮するモデル
欠測データなし



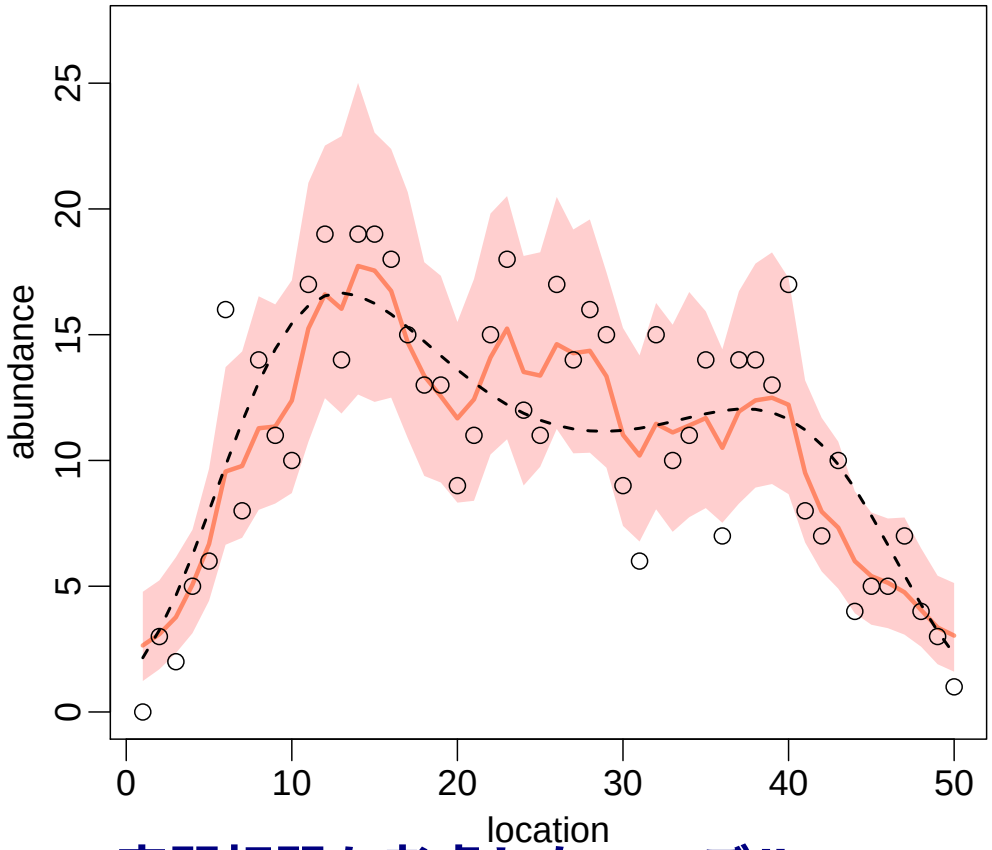
空間相関を考慮しないベイズモデルの推定結果

空間相関を考慮しないモデル
欠測データなし



空間相関を考慮する vs しないモデル

空間相関を考慮するモデル
欠測データなし

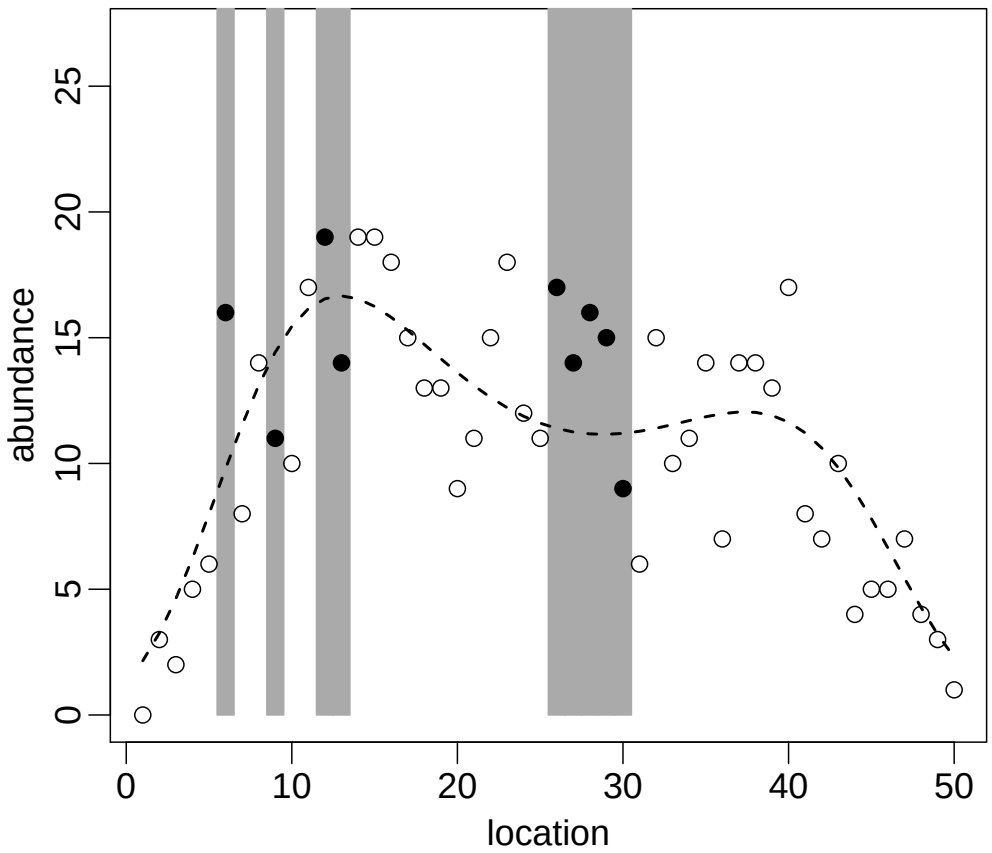


空間相関を考慮しないモデル
欠測データなし

架空の例題 (続): 欠測がある場合は?!

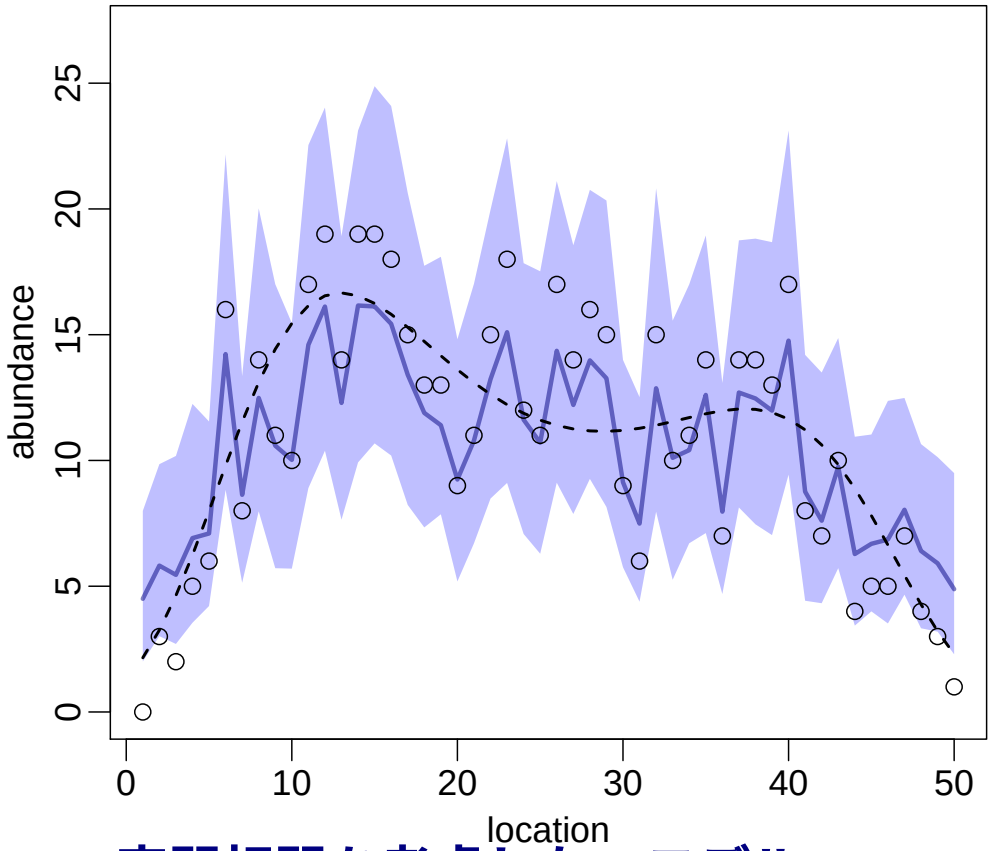
欠測あり

欠測値の予測!



空間相関を考慮しないベイズモデルは欠測にヨワい

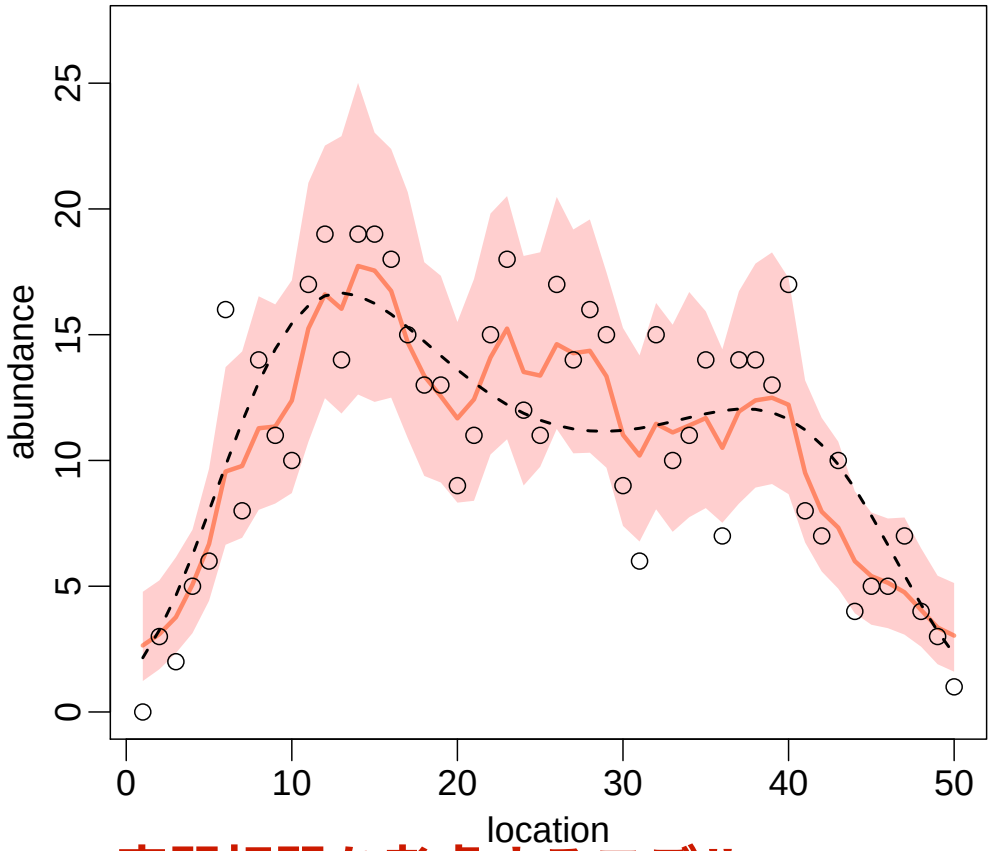
空間相関を考慮しないモデル
欠測データなし



空間相関を考慮しないモデル
欠測あり

空間相関を考慮するモデルは欠測に頑健

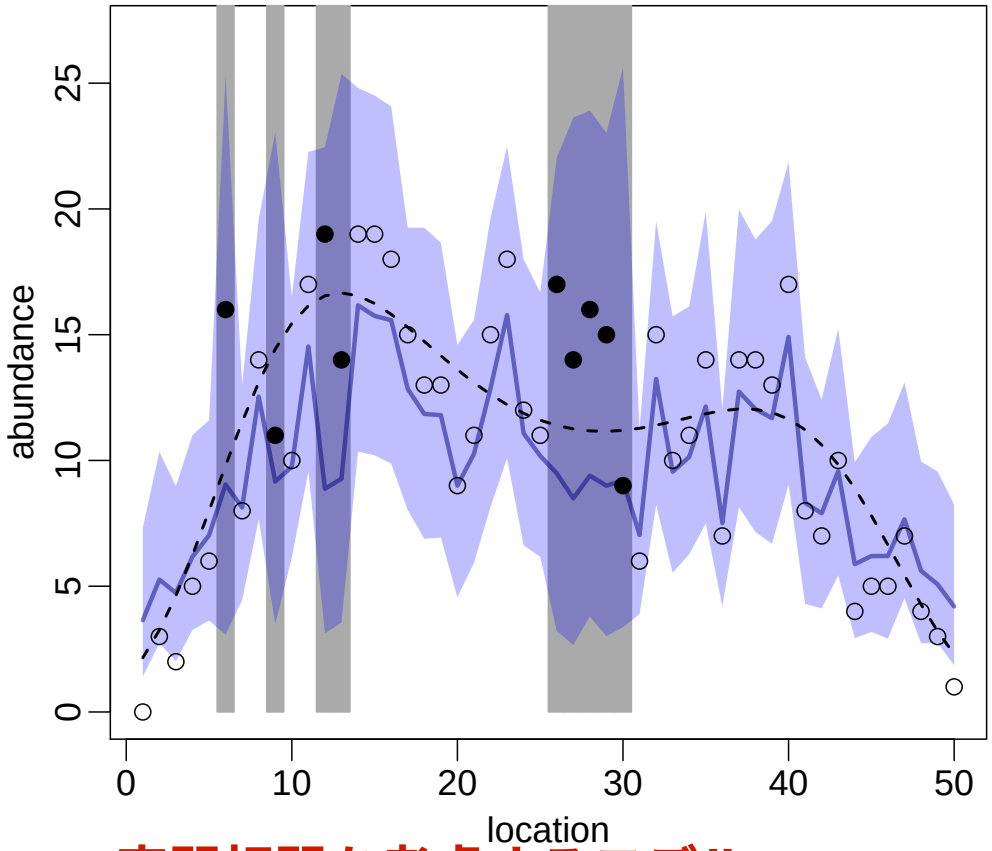
空間相関を考慮するモデル
欠測データなし



空間相関を考慮するモデル
欠測あり

ベイズモデルのご利益: 欠測をうまく対処

空間相関を考慮しないモデル
欠測あり



空間相関を考慮するモデル
欠測あり

まとめ: 空間構造のあるランダム効果

- ガウス確率場 (Gaussian random field) で「隣と似ている」ランダム効果を表現する
- 各地点独立と仮定するランダム効果でも，それっぽい推定はできないこともない
- しかし欠測のあるデータセットの解析においては，空間相関を考慮したベイズ統計モデルが威力を発揮するだろう
- ガウス確率場のモデリングはさらにいろいろと工夫できる—よりなめらかに変化させるような方法もある

ウミガメ上陸数の統計モデリング

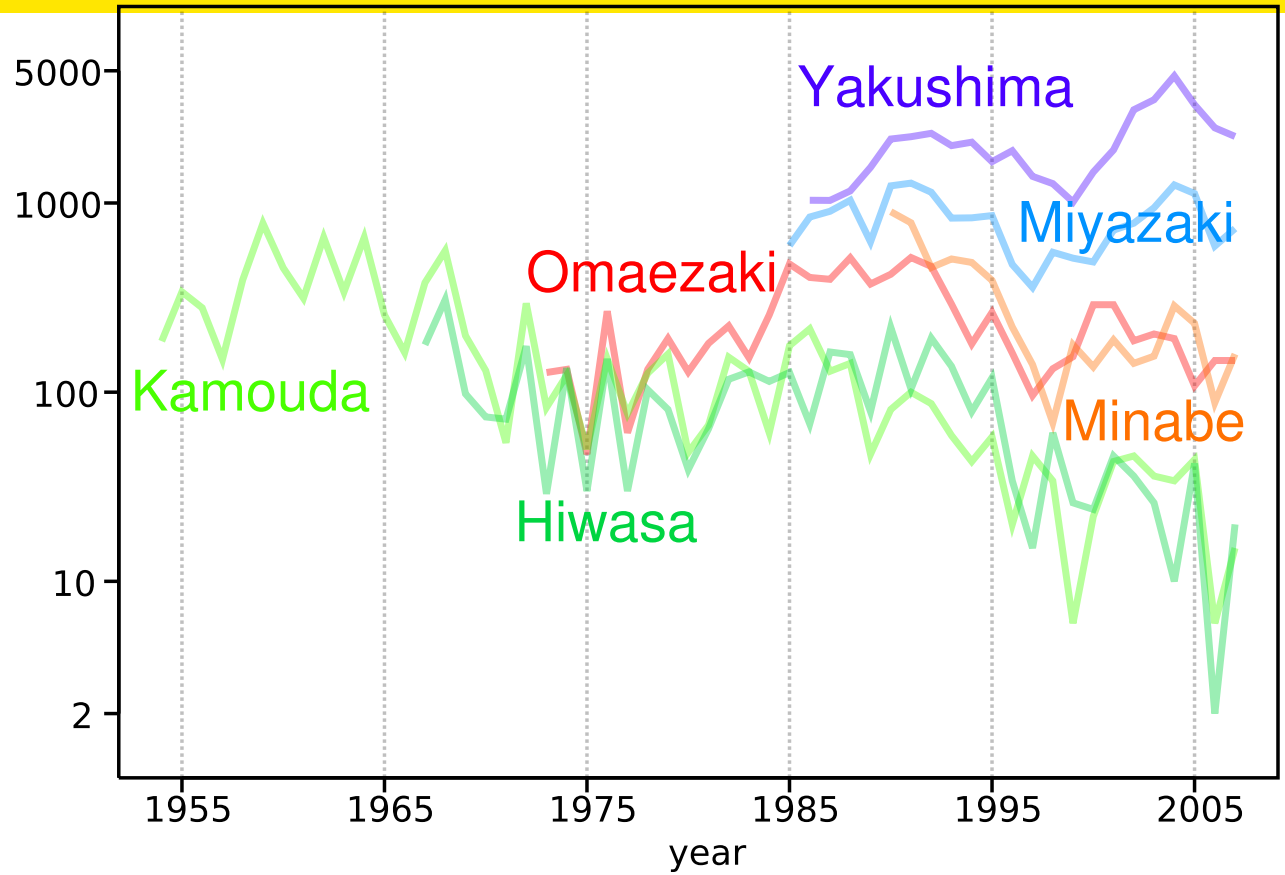
階層ベイズモデルを応用した時系列データ解析

(重田麻衣氏・亀崎直樹氏との共同研究)

ウミガメ上陸数の観測データ (1954–2007)


*1


*2



Omaezaki (**静岡**), Minabe (**和歌山**), Kamouda (**徳島**)

Hiwasa (**徳島**), Miyazaki (**宮崎**), Yakushima (**鹿児島**)

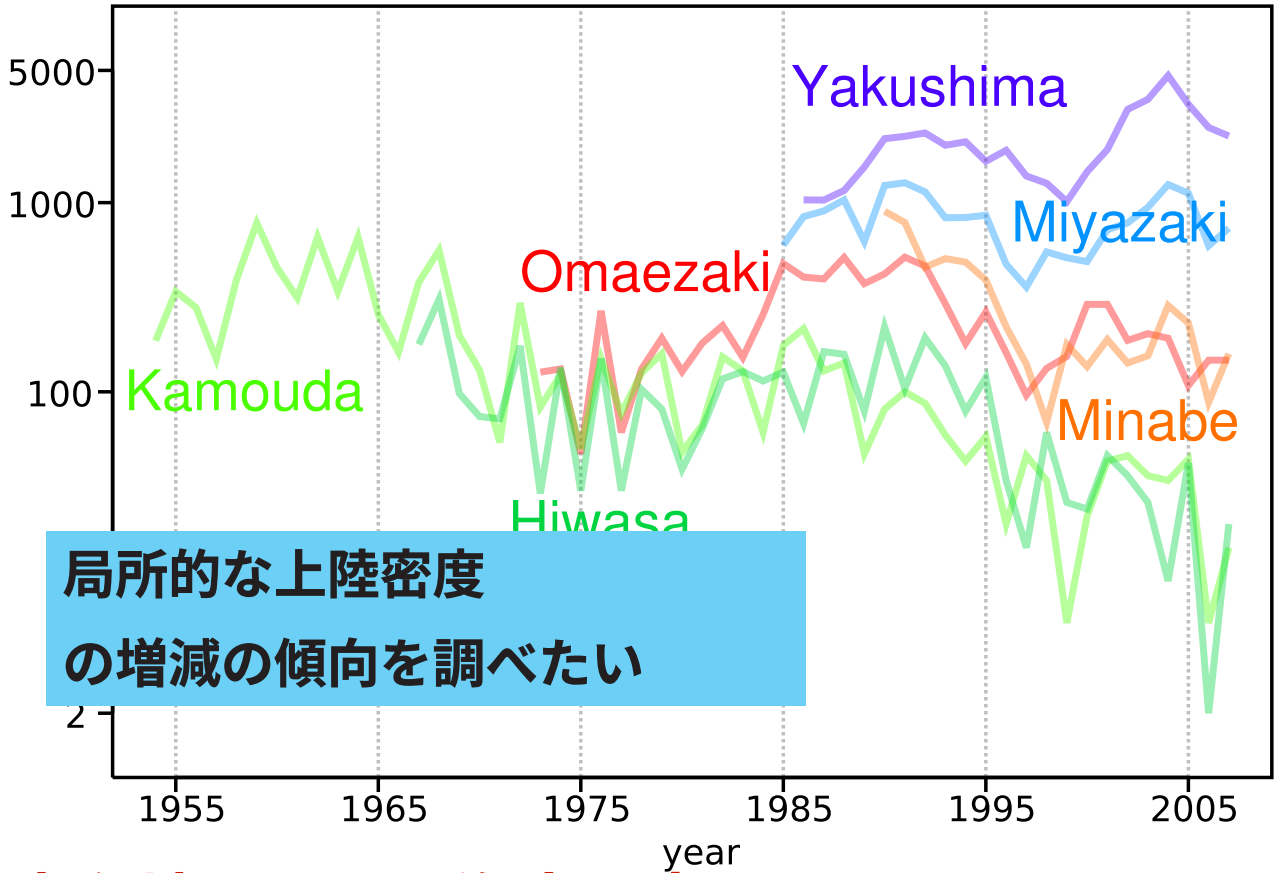
*1 © Mike Gonzalez, October 14, 2007. Wikimedia Commons. *2 © OpenCage, February 2, 2006. Wikimedia Commons.

ウミガメ上陸数の観測データ，その内容

- 各地のウミガメ上陸海岸ごとに，ウミガメ上陸観察団体 (アマチュア研究者が主体) があって，長年にわたったデータをとっている
- 全国的な組織であるウミガメ協議会はそのデータの所在を把握している
- 統一的な方法で調査しているわけではないので，海岸間の比較が難しいところがある
    - たとえば上陸数の大小比較は意味がないかも

問: 産卵場所としての利用が減少している海岸は?

産卵場所として不適 → **上陸数の減少となっている**?



- 上陸密度 ⟺ **産卵地としての海岸の良さ?**
- 上陸数変動の状態空間モデル (階層ベイズモデルの一種)

ウミガメ上陸数の観測モデル

各年のウミガメ上陸数の観測値が混合ポアソン分布にしたがうと仮定する

「海岸ノイズ」は年ごとに独立 (階層事前分布)

m[1]　m[2]　m[3]平均上陸密度

海岸ノイズ

観測不可能な世界

観測可能な世界

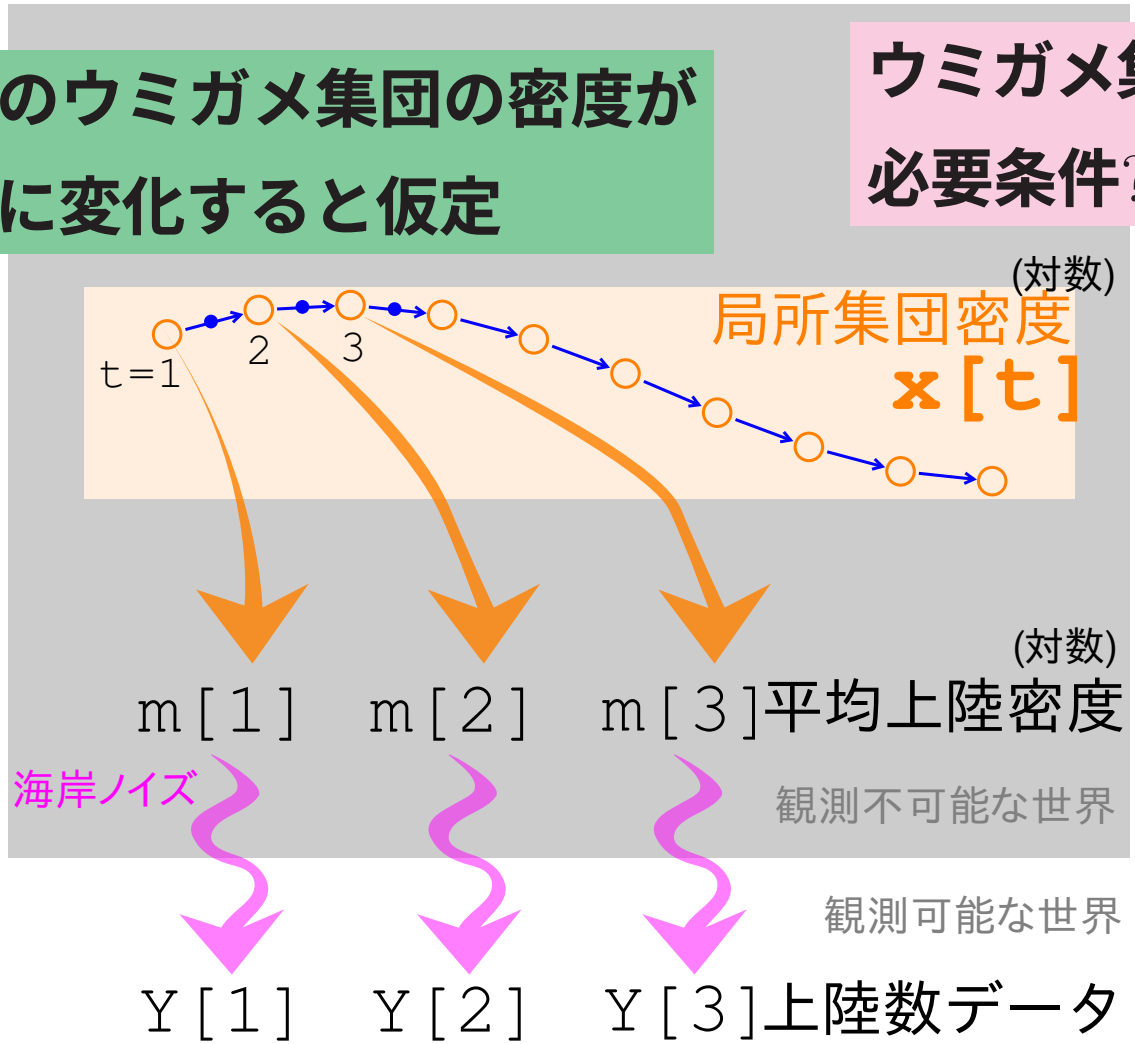
観測年 1, 2, 3

Y[1]　Y[2]　Y[3]上陸数データ

ゆっくり変化させる地域集団密度 (状態モデル)

ある海域のウミガメ集団の密度が
ゆるやかに変化すると仮定

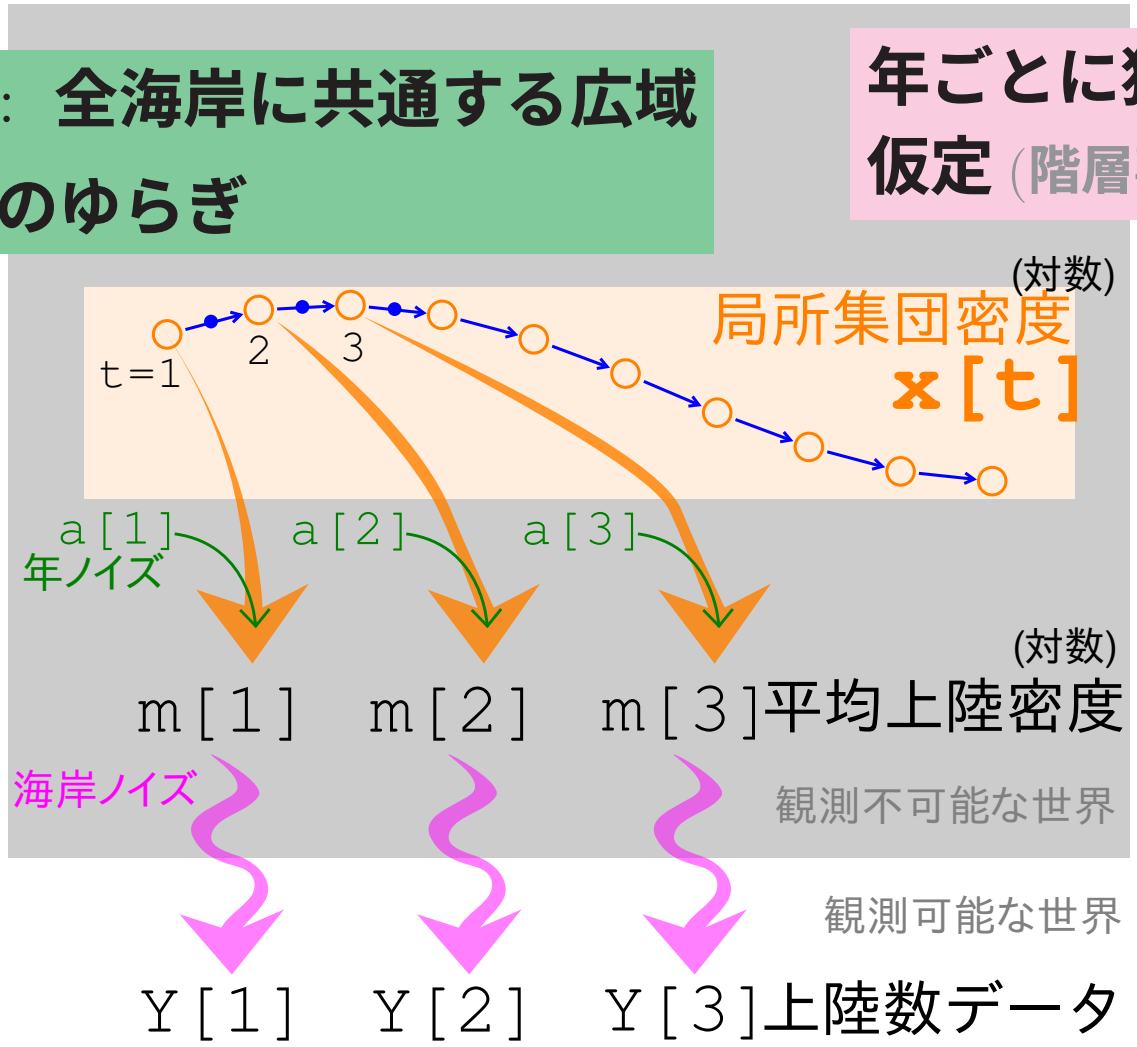
ウミガメ集団維持の
必要条件?



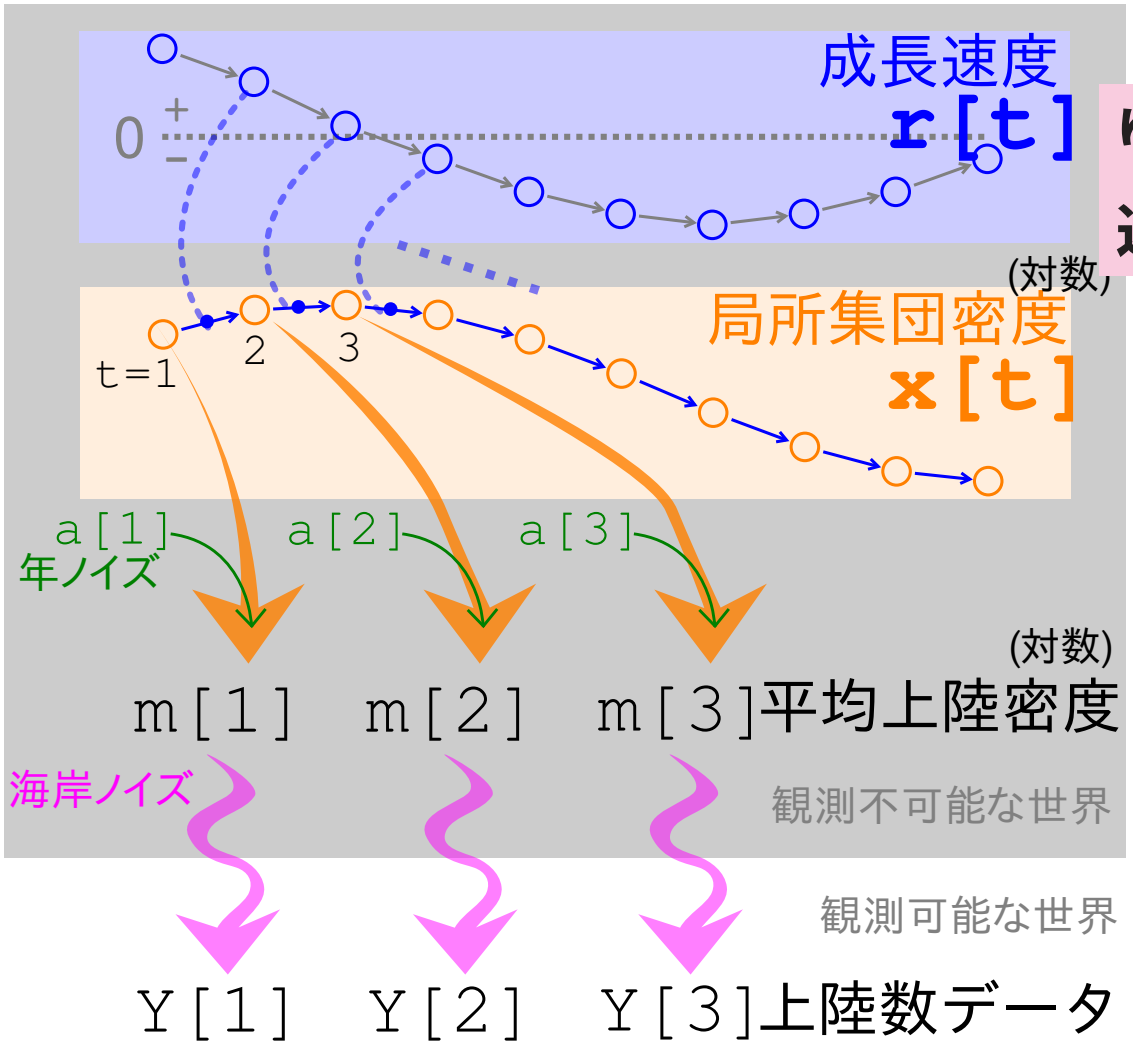
年ノイズ: 年ごとに気象条件などの影響など

年ノイズ: 全海岸に共通する広域的な環境のゆらぎ

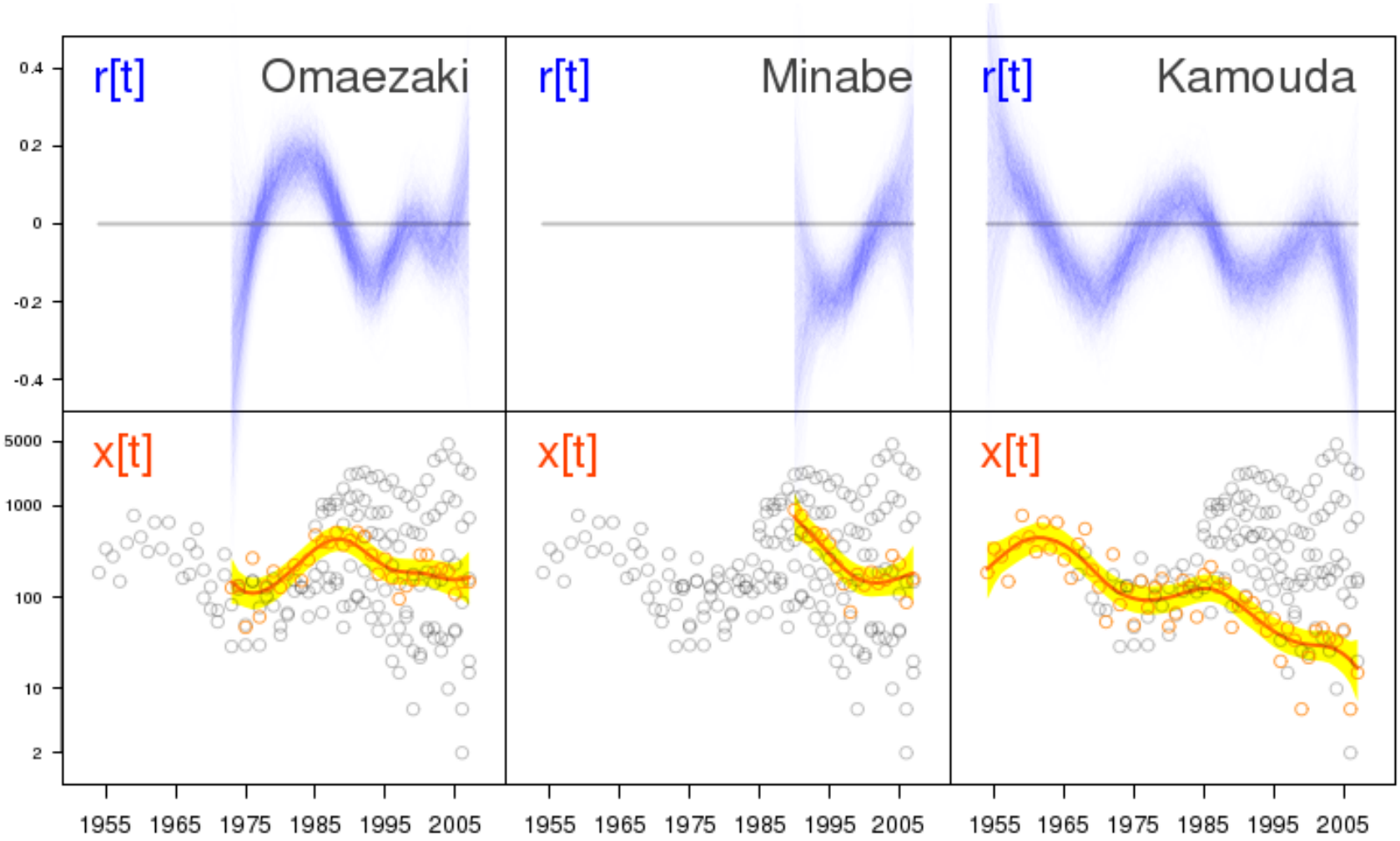
年ごとに独立を仮定 (階層事前分布)



ウミガメ上陸数の状態空間モデル

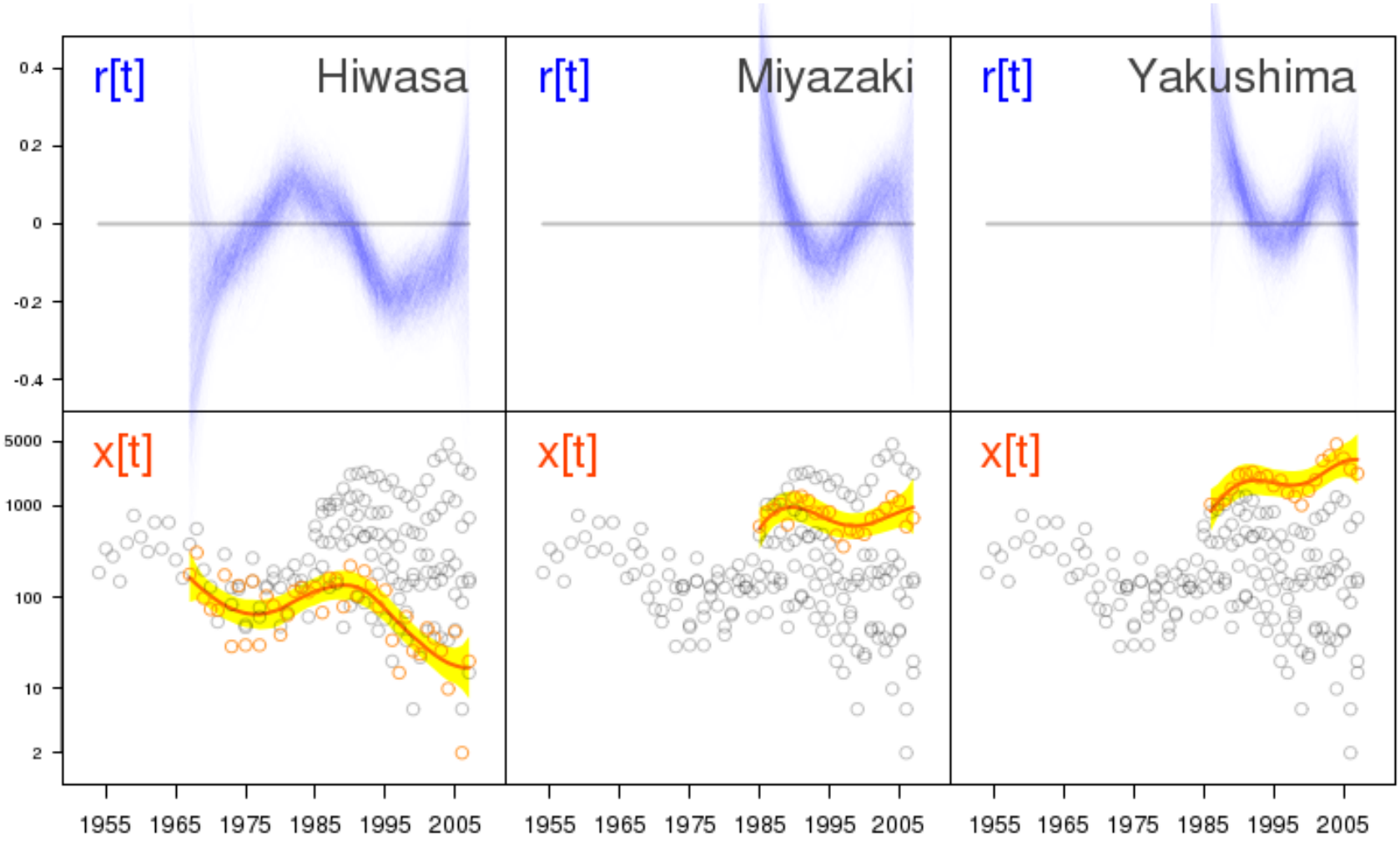


上陸密度増減の事後分布 (静岡, 和歌山, 徳島)



上段: 平均増殖速度; 下段: 上陸密度

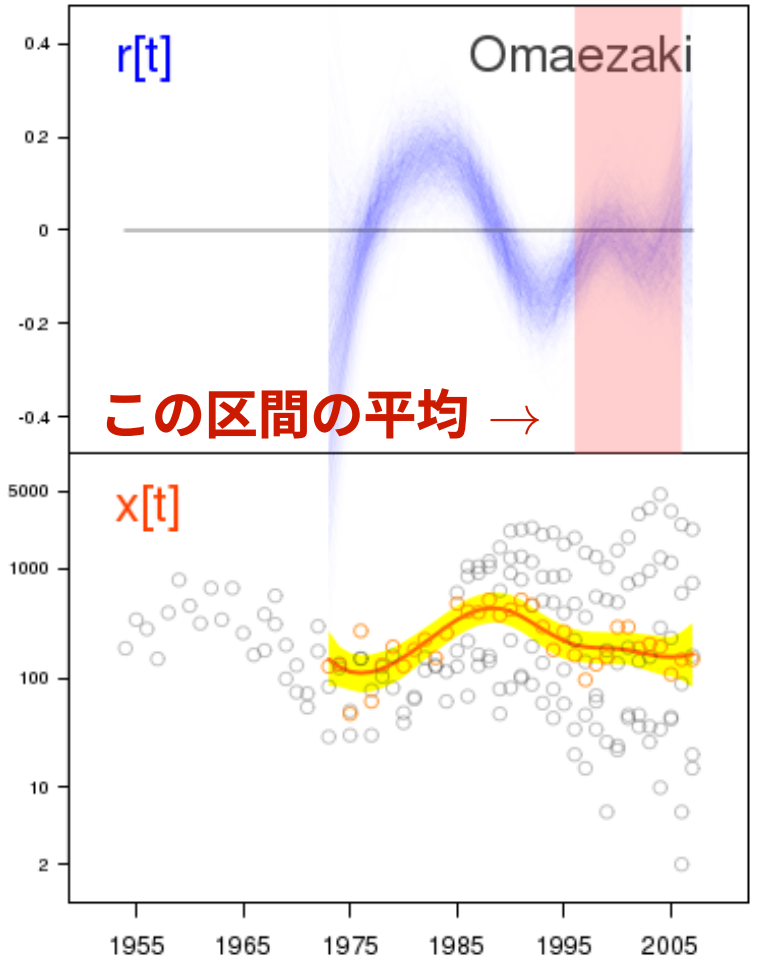
上陸密度増減の事後分布 (徳島, 宮崎, 鹿児島)



上段: 平均増殖速度; 下段: 上陸密度

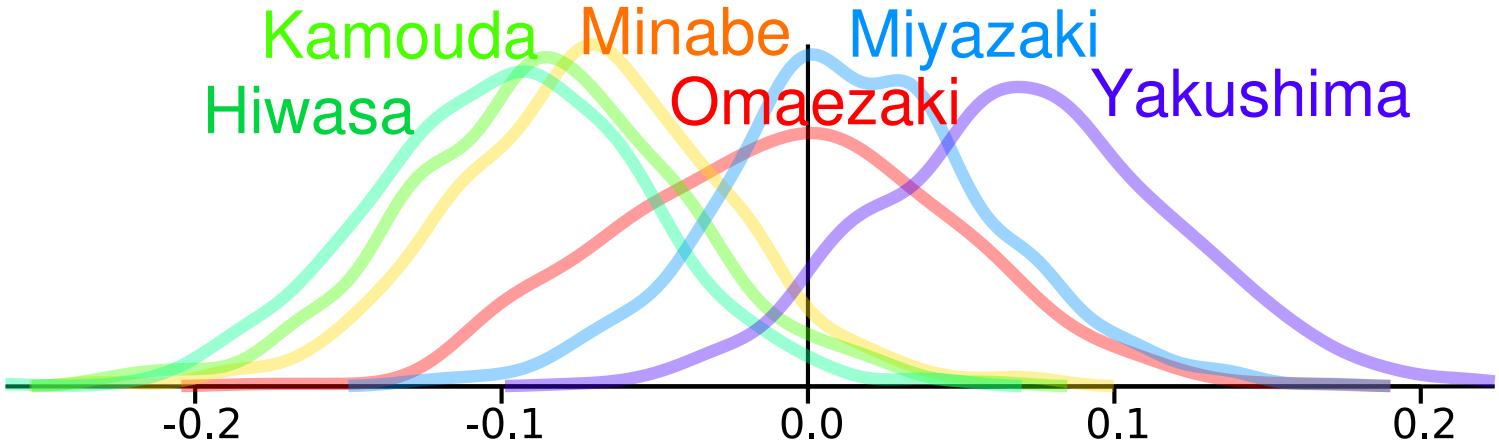
「産卵地として選ばれなくなってきた海岸」はどこ?

- どの海岸で上陸数が減少しているか?
- **平均成長速度**: 海岸ごとの成長速度 r[t] の最近 11 年間の平均を指標とした
- これは個体群生態学でよく使われる，個体群成長速度の**幾何平均**に該当する
- r[t] の事後分布を使って，**平均成長速度**の予測分布を評価する



平均成長速度による産卵地としての海岸評価

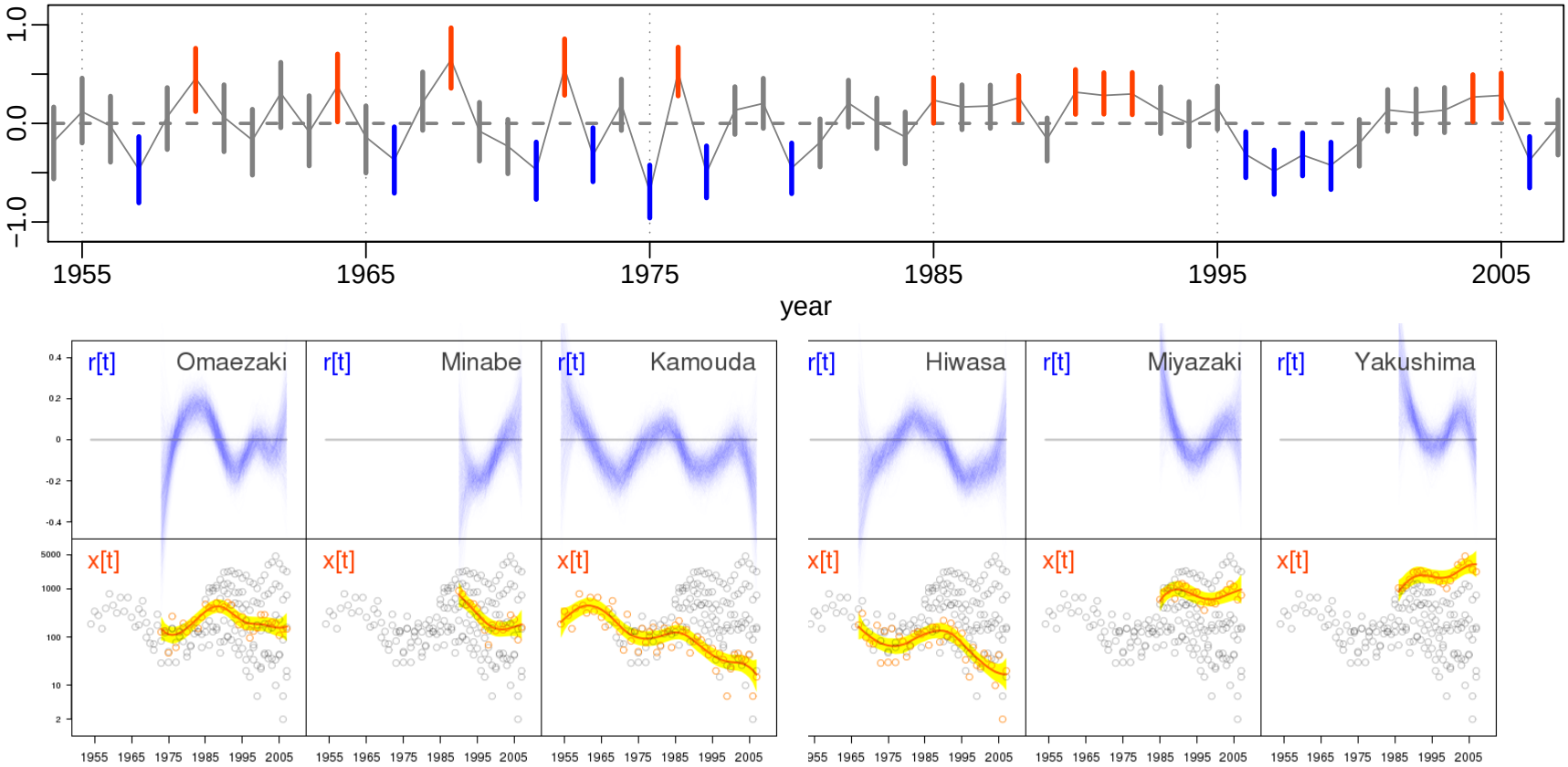
- 1996–2006 年の平均成長速度の予測分布 (海岸ごとに)



- 徳島県の 2 海岸 (Hiwasa, Kamouda) では，予測分布の 95% 区間にゼロを含まない
  - 産卵地として選ばれなくなりつつある?
- 他の海岸では何とも言えない?

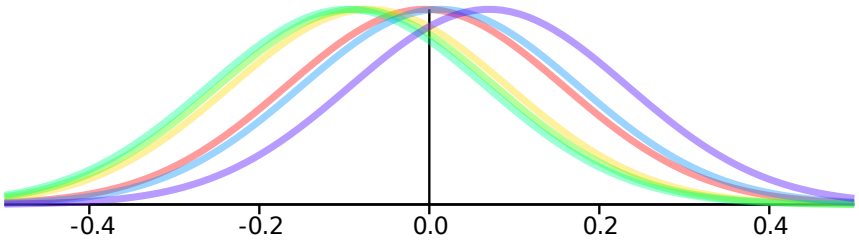
全海岸に共通する年変動 a[t]

- 年ごとに独立ではなく，ゆるやかに変動?



- ウミガメの繁殖活性と気象要因との関連?

10 年後のウミガメ上陸数は?

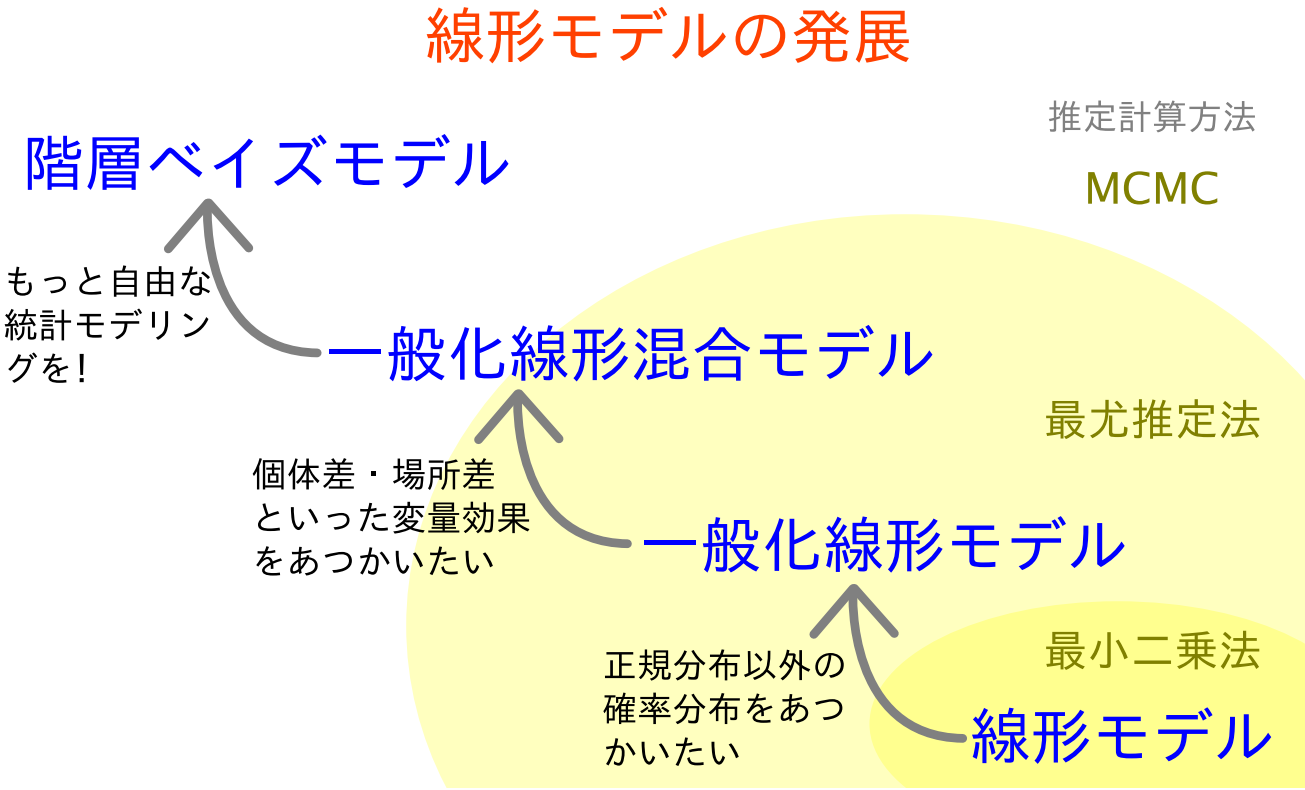


- 海岸ごとの平均成長速度 r[t] の 10 年間の変動幅を予測した
- 2006 年時点での平均成長速度 r[t] を左で推定した平均成長速度の平均の分布の中央値に等しいと仮定
- r[t] の年変動の SD は上で推定した 0.052 を仮定して 10 年間の random walk の範囲を示した

海岸	2016 年に r[t] がプラスとなる確率
Omaezaki	49%
Minabe	33%
Kamouda	30%
Hiwasa	28%
Miyazaki	53%
Yakushima	67%

ここでこの 統計モデリング入門の授業 は終了

- 一般化線形モデル → 階層ベイズモデル
- 最尤推定 → Markov chain Monte Carlo (MCMC)



みなさん，おつかれさまでした!