MCMC と階層ベイズモデル

データ解析のための統計モデリング入門

久保拓弥 kubo@ees.hokudai.ac.jp

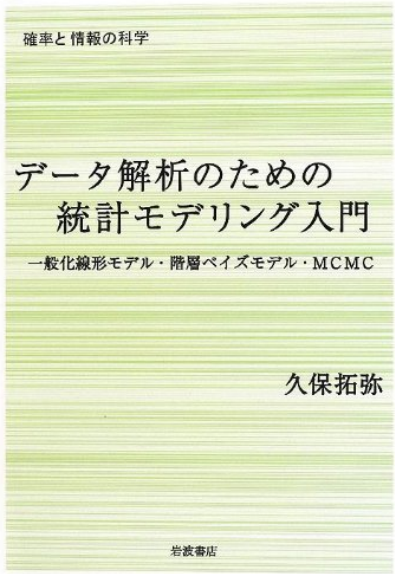
統数研・新統計入門 NEO シリーズ http://goo.gl/YzWd9m

2014–02–26

ファイル更新時刻: 2014–04–03 13:34

自己紹介: 久保拓弥 (twitter: @KuboBook)

- 北大・地球科学研究院で生態学 (ecology) という生物学一分野のデータ解析をやっています
- 統計学とかは独学
- 生態学で使われる統計学的な手法があまりにもでたらめなので「統計モデリング入門」を書きました
- 子育てばてで体重減少中，とか



http://goo.gl/Ufq2

今回の「入門講義」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

Markov Chain Monte Carlo (MCMC)

くわしくはあとで説明

この「入門」ではパラメーター推定

のための道具として使用

かなり初歩的・非数学的な

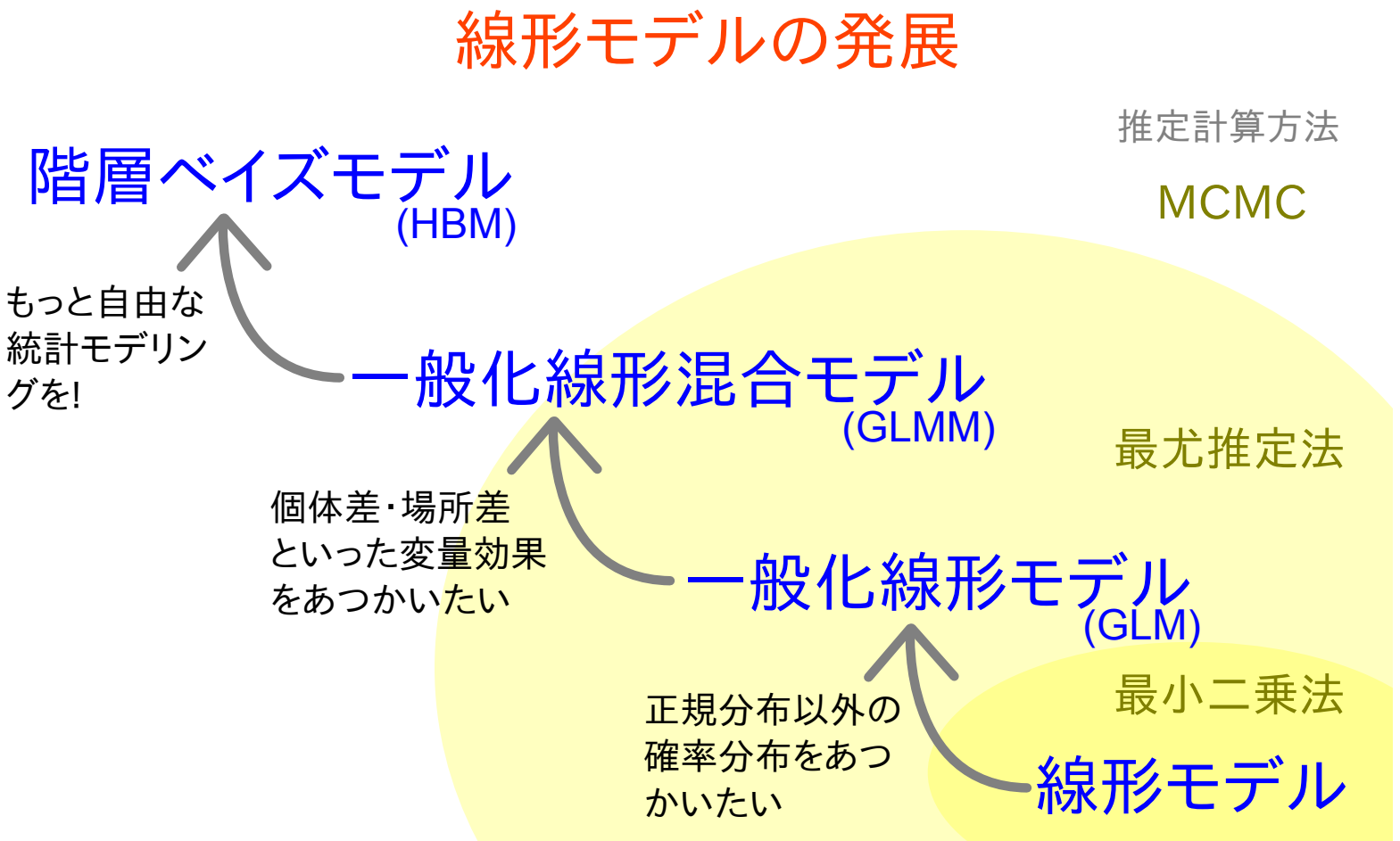
説明に終始します!

すみませんが配信まわりがややこしいので
今回は R などの実演はなしです

ハナシの全体の流れを

まず紹介しておきます

「統計モデリング入門」で説明したかったこと



データの特徴にあわせて線形モデルを改良・発展させる

なぜ統計モデルを発展させる?

たとえば,

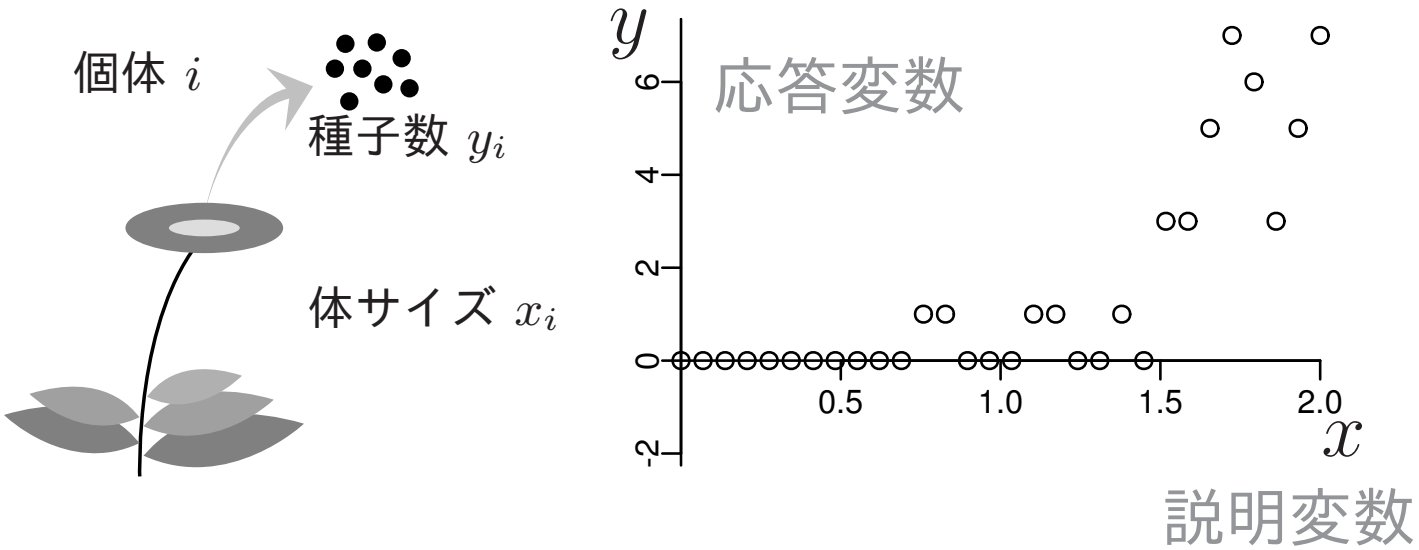
「どんなデータでも直線回帰」

といった「統計学お作法」のまずさ

について検討してみましょう

0 個, 1 個, 2 個と数えられるデータ

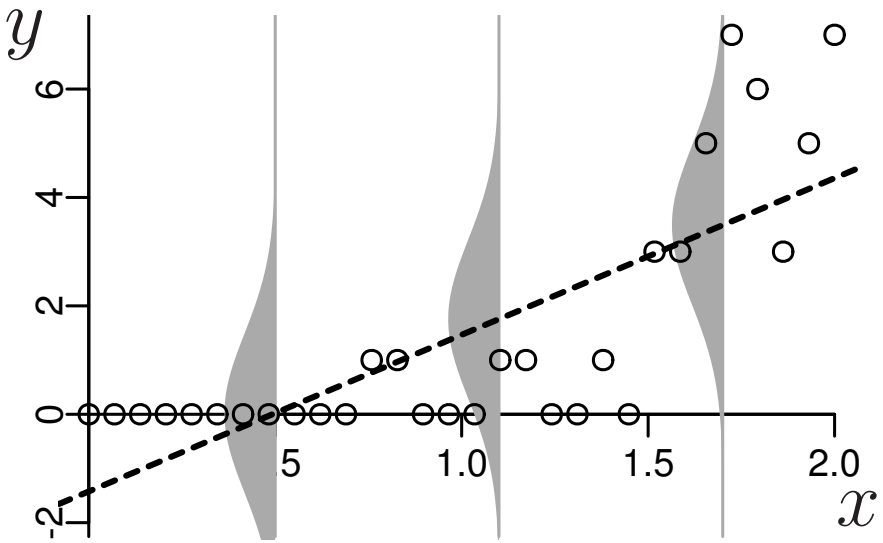
カウントデータ ($y \in \{0, 1, 2, 3, \cdots\}$ なデータ)



- x は植物個体の大きさ，y はその個体の種子数
- x が大きくなると y が増えるように見えるが……

正規分布を使った統計モデル …… ムリがある?

正規分布・恒等リンク関数の統計モデル



NO!

とにかくセンひけ!

傾き「ゆーい」?

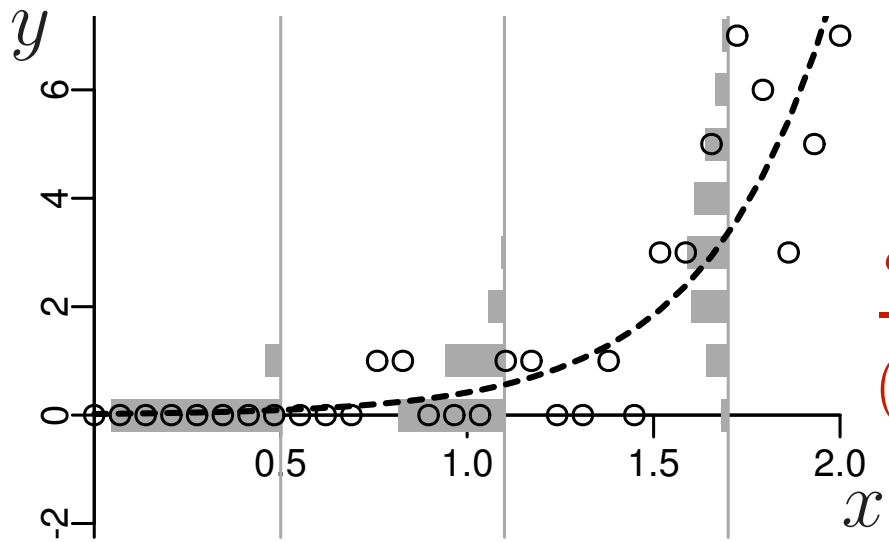
…という安易な手口

- タテ軸のばらつきは「正規分布」なのか?
- y の値は 0 以上なのに ……
- 平均値がマイナス?

ポアソン分布を使った統計モデルなら良さそう?!

ポアソン分布・対数リンク関数の統計モデル

応答変数



YES!

これが
一般化線形モデル
(GLM)！

説明変数

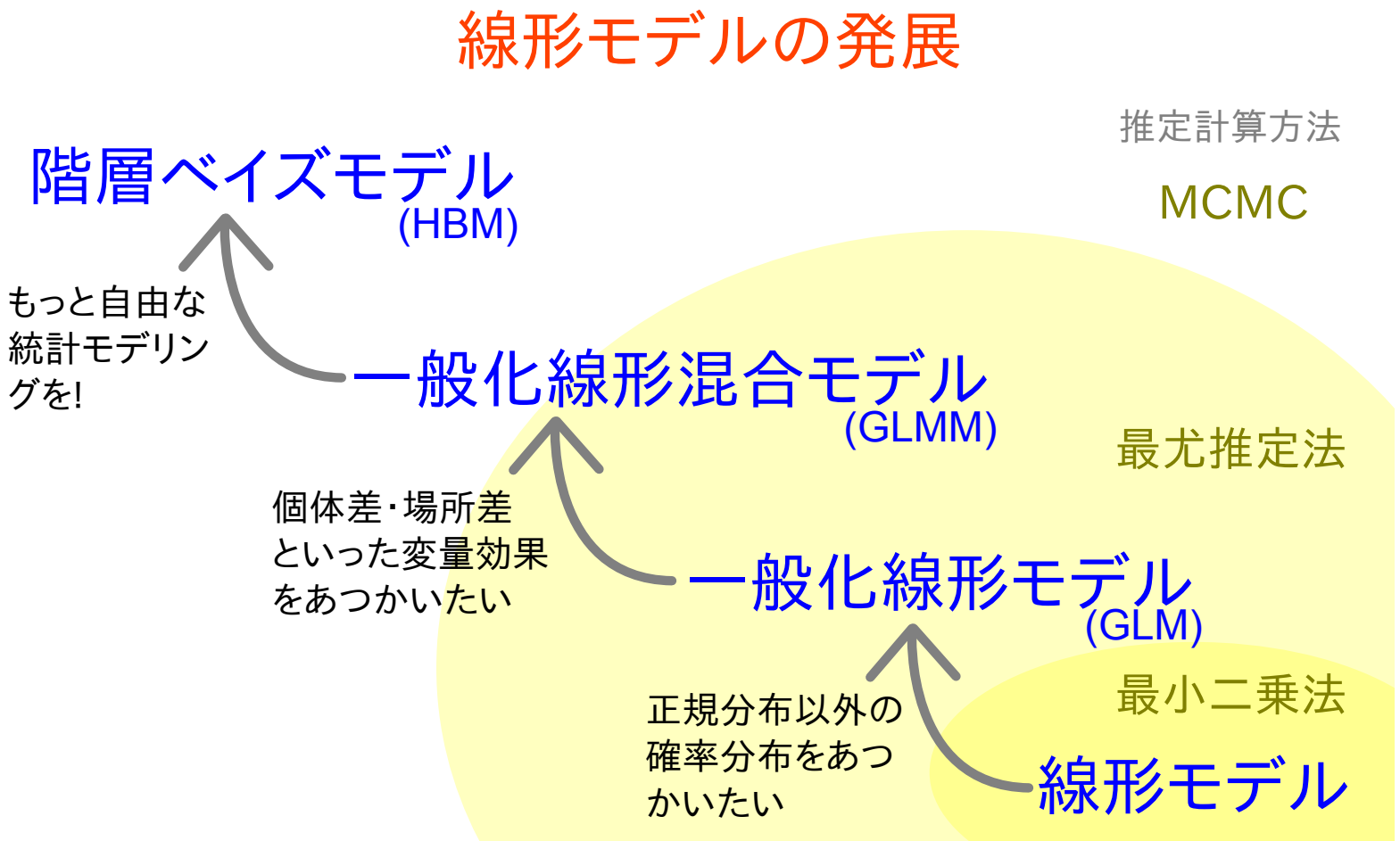
- タテ軸に対応する「ばらつき」
- 負の値にならない「平均値」
- 正規分布を使ってるモデルよりましだね

データの構造や性質をよく見て

統計モデルの部品である

確率分布などを選んでいく

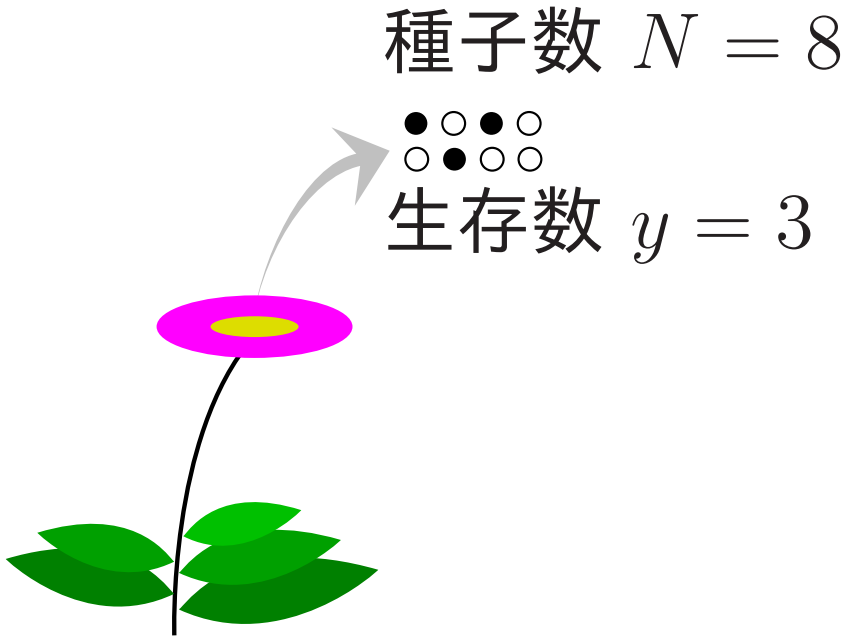
いま説明したこと: LM から GLM へ



データの特徴にあわせて線形モデルを改良・発展させる

今日の例題: 植物の種子の生存確率

- 架空植物の種子の生存を調べた
- 100 個体ぶんのデータをとった

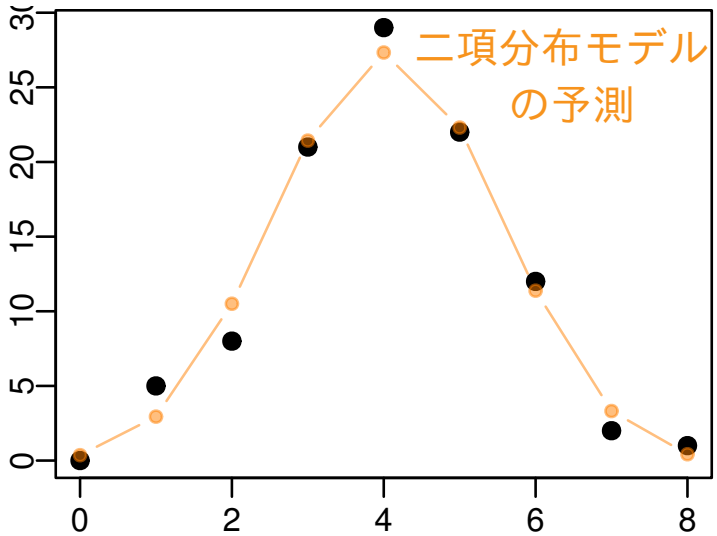


GLM でうまく説明できる「個体差なし」の集団

N 種子中 y 個が生存する確率は二項分布

$$p(y \mid q) = \binom{N}{y} q^y (1-q)^{N-y},$$

観察された
植物の個体数



生存していた種子数 y_i

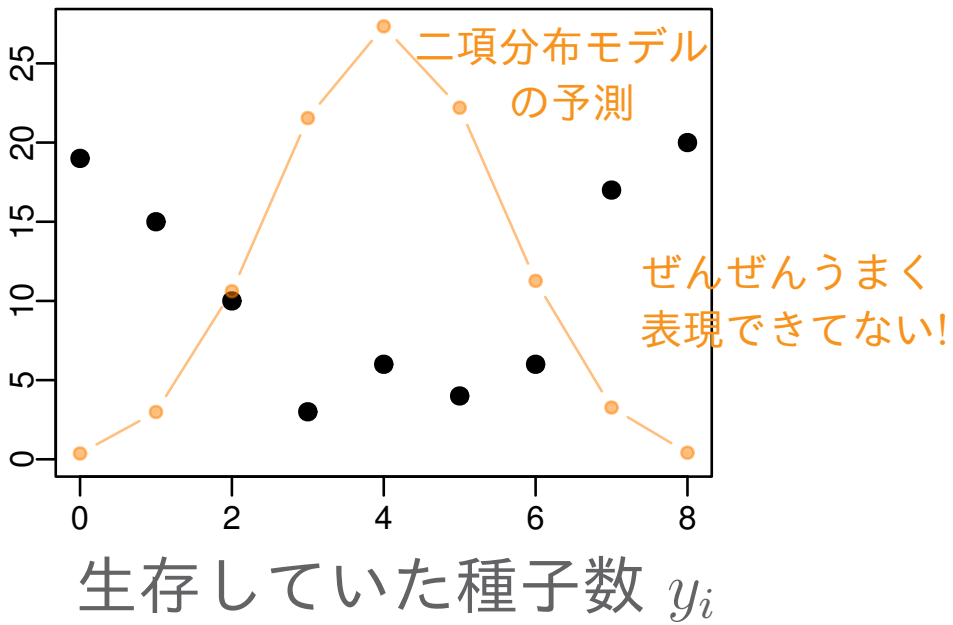
最尤 (さいゆう) 推定された生存確率: $\hat{q} = 0.505$

GLM ではうまく説明できない観測データ!

さっきの例題と同じようなデータなのに?

(現実の生物データ,「格差」社会!)

観察された
植物の個体数



ヒント: 「個体差」あり……均質ではない集団

個体差を考慮した GLM

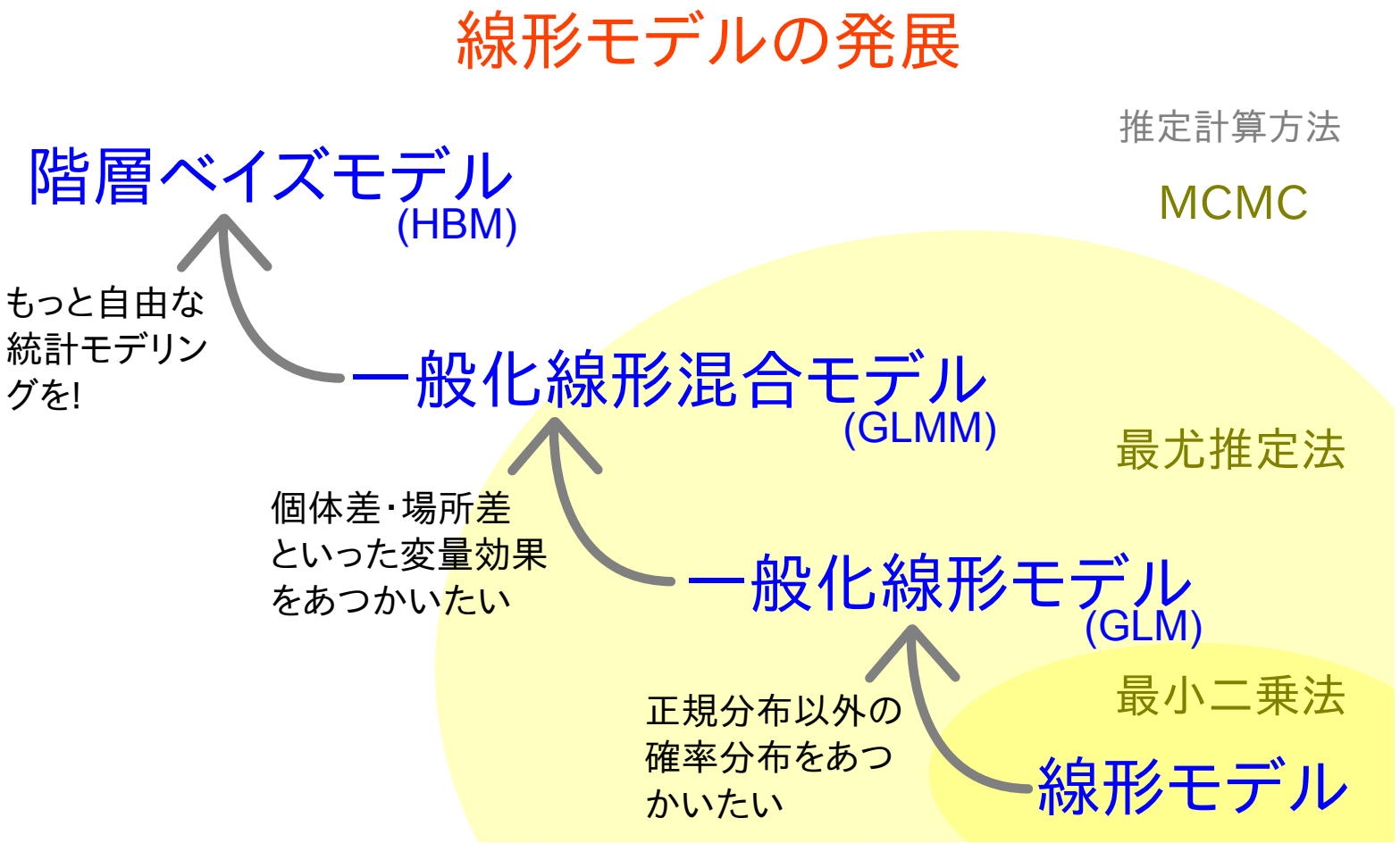
つまり一般化線形混合モデル (GLMM)

が必要になる (モデルの複雑化)

「個体差」あらわすパラメーターが増える

パラメーターの最尤推定がしんどくなる

統計モデルにあわせて推定方法も変える?



データの特徴にあわせて線形モデルを改良・発展させる

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. MCMC のためのソフトウェア

   事後分布からサンプリングしたい

5. 階層ベイズモデルの推定

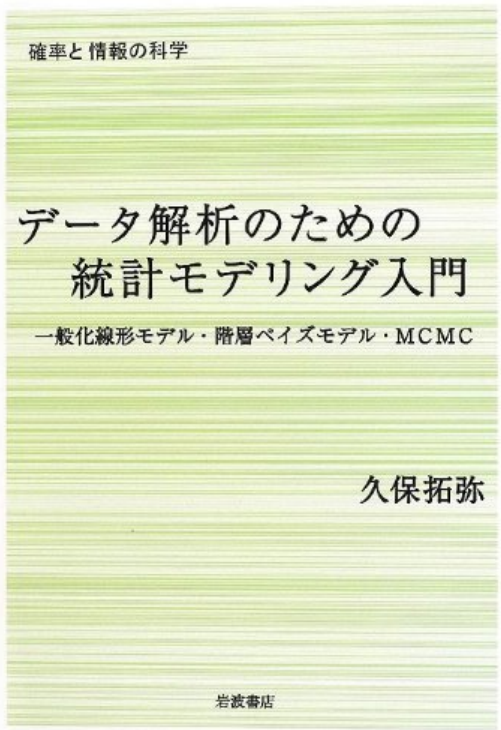
   ソフトウェア WinBUGS を使ってみる

今回の内容と「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



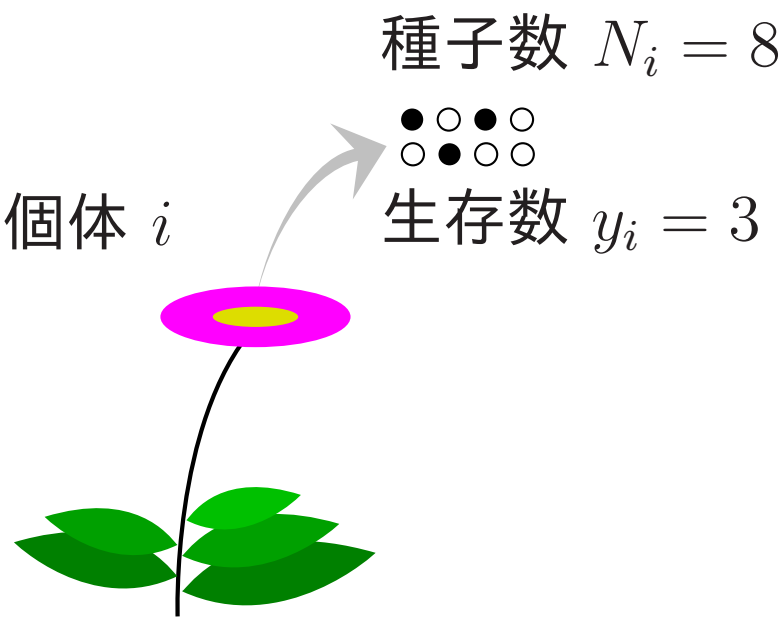
1. 二項分布のパラメーターを最尤推定

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

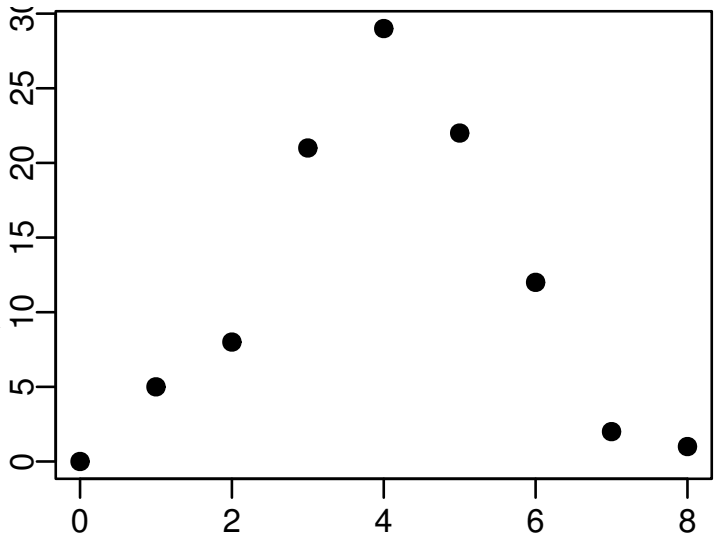


- データ: 植物 100 個体，合計 800 種子の生存の有無を調べた
- 問: この植物の生存確率はどのように統計モデル化できるか?

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

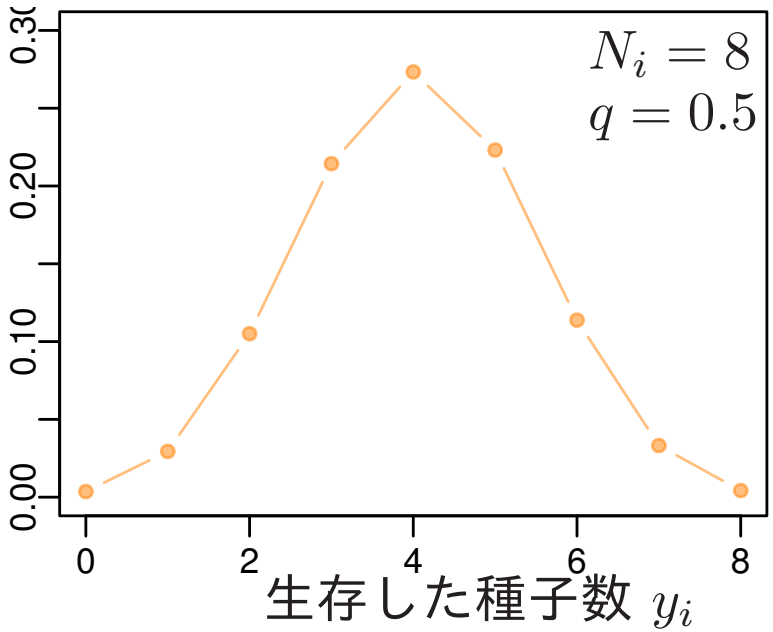
- 生存確率を推定するために二項分布 という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率

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

- 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

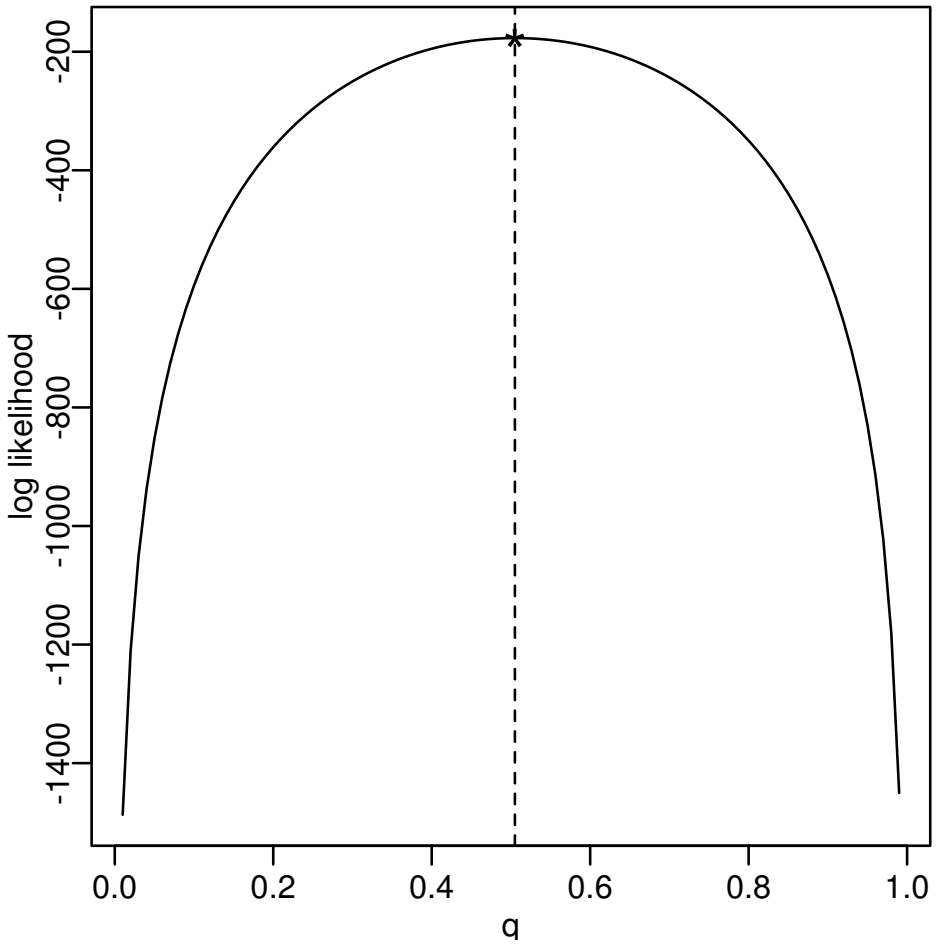
- 対数尤度 $L(q \mid \text{データ})$ が最大になるパラメーター q の値をさがしだすこと

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

  $$\partial \log L(q \mid \text{データ}) / \partial q = 0$$

- 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

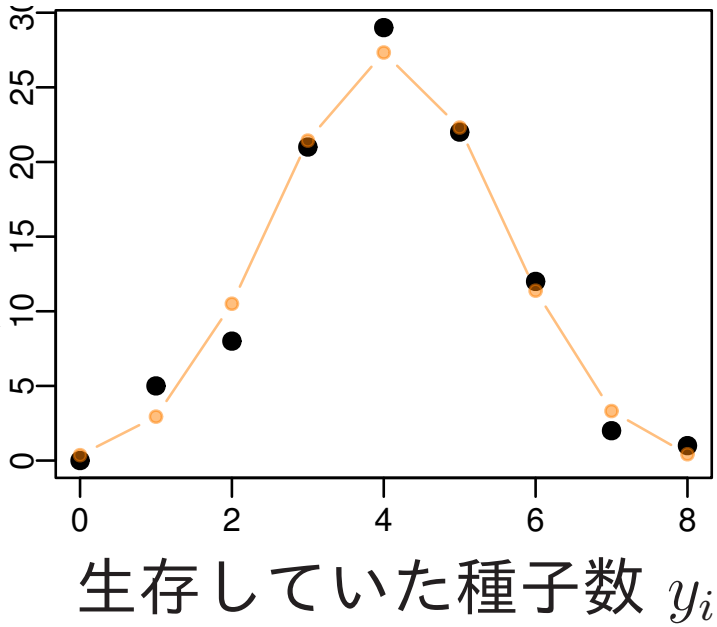
$$\hat{q} = \frac{\text{生存種子数}}{\text{調査種子数}} = \frac{404}{800} = 0.505$$



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 $\binom{8}{y} 0.505^y 0.495^{8-y}$

観察された
植物の個体数



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」……??

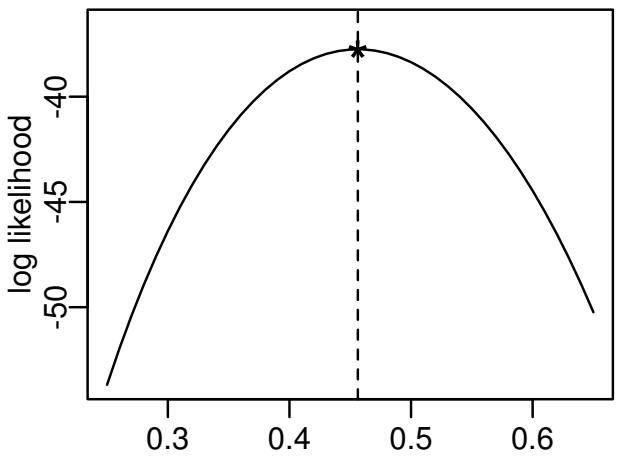
MCMC をもちださなくてもいい簡単すぎる問題

説明のためあえてメトロポリス法を適用してみる

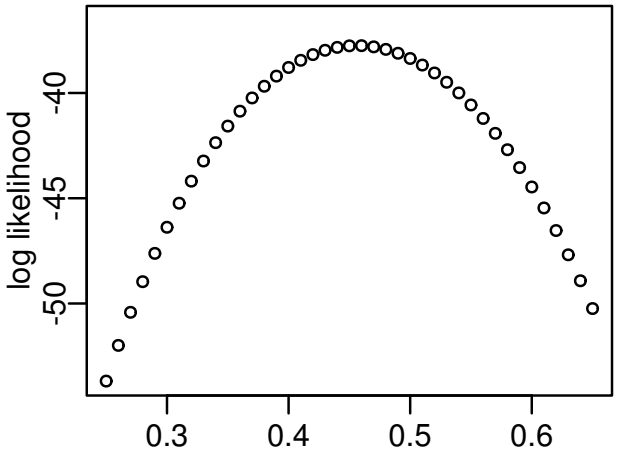
メトロポリス法を説明するための準備

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



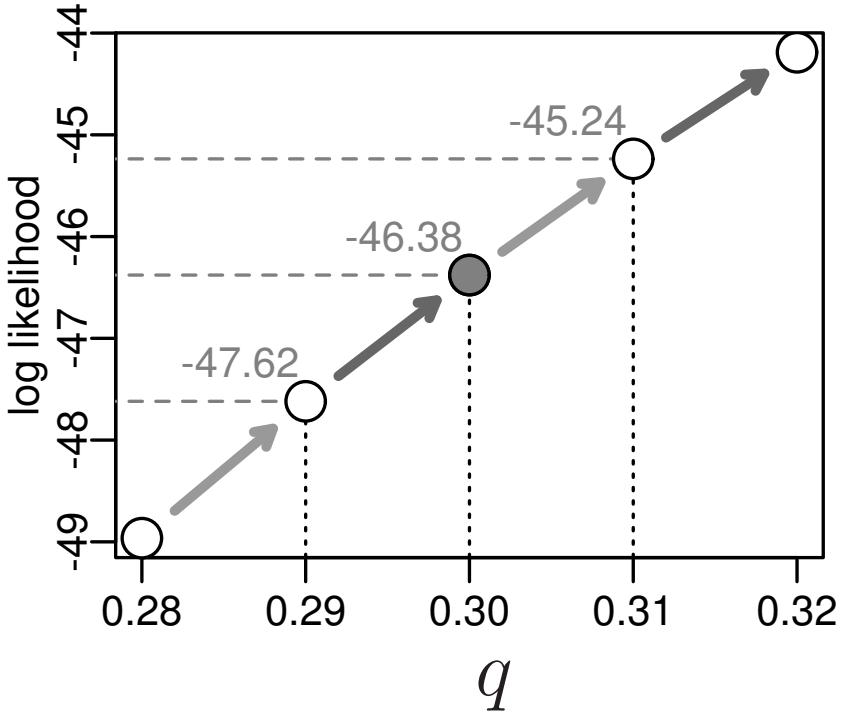
説明を簡単にするため

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

   (ここでは 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

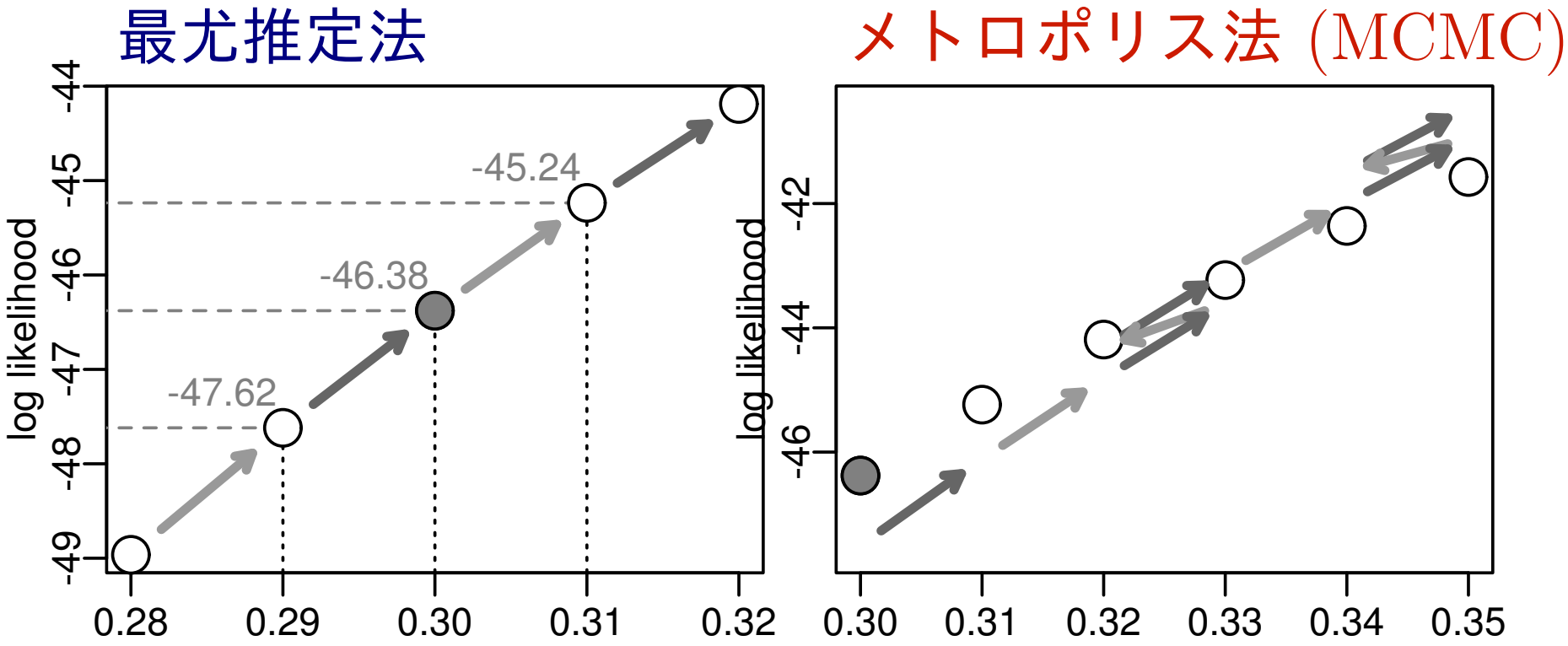
   (新しく選んだ q の値を q_{new} としましょう)

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



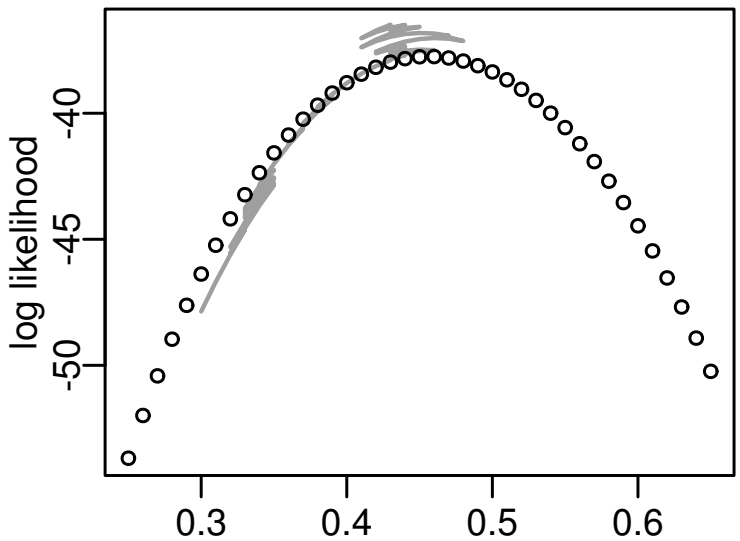
メトロポリス法だと

「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

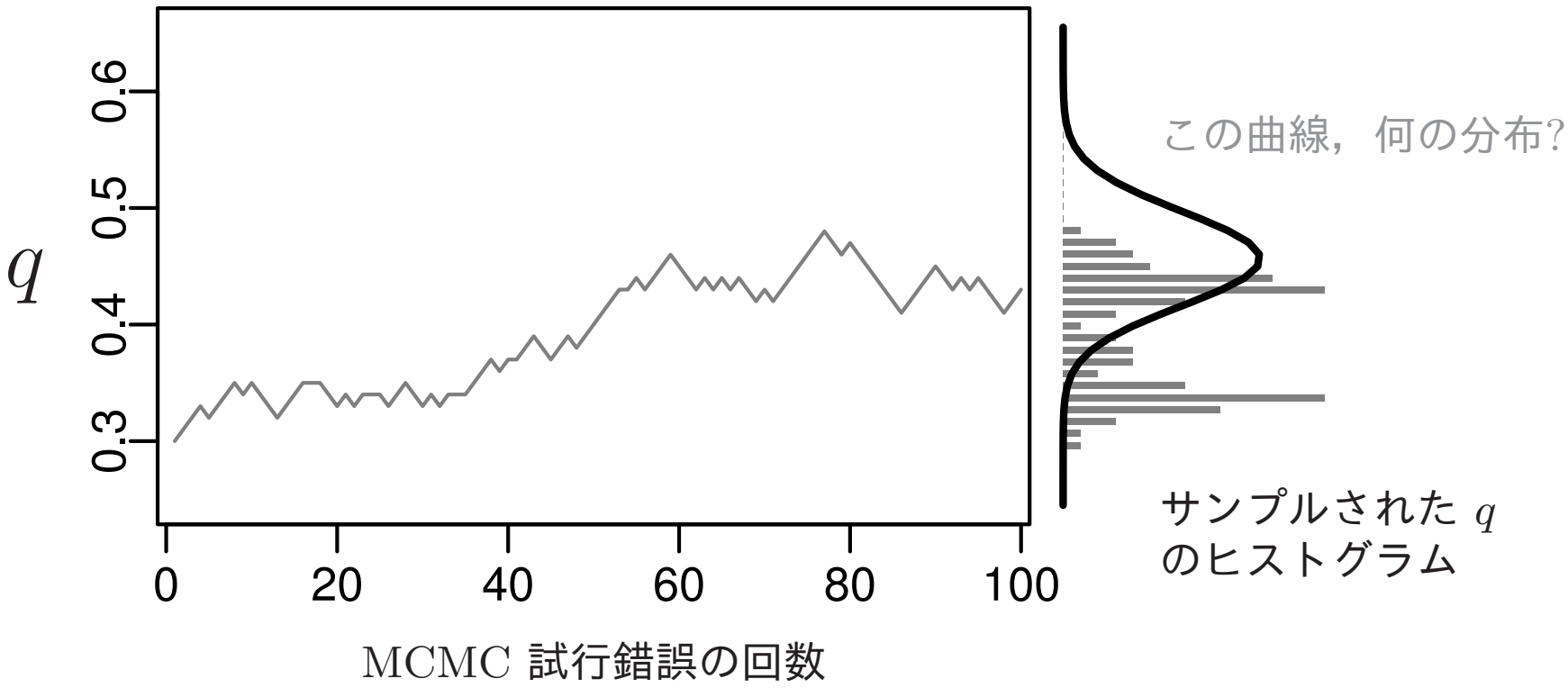


ときどきはでに落っこちる

何のためにこんなことをやるのか?

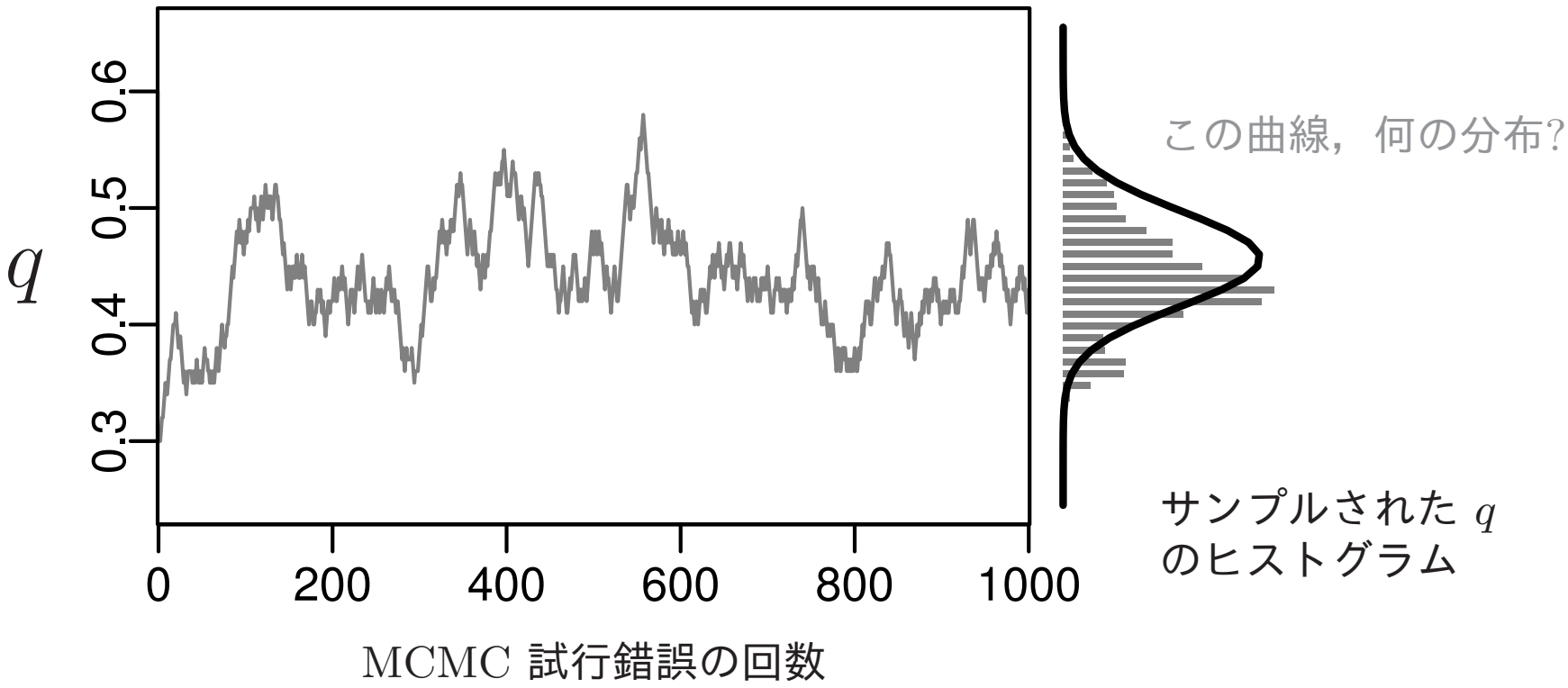
q の変化していく様子を記録してみよう

ステップごとに q の値をサンプリング



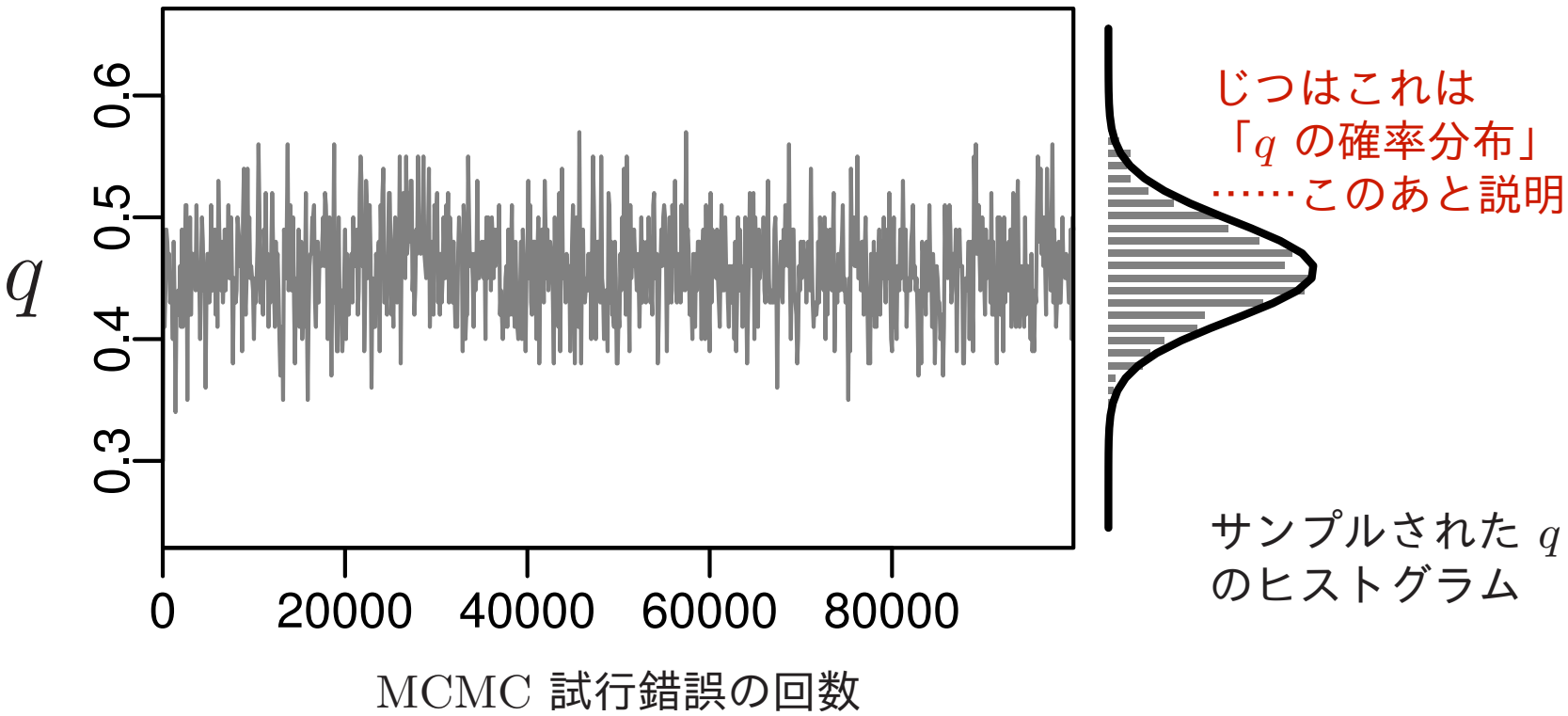
もっと試行錯誤してみたほうがいいのかな?

もっと長くサンプリングしてみる



まだまだ……?

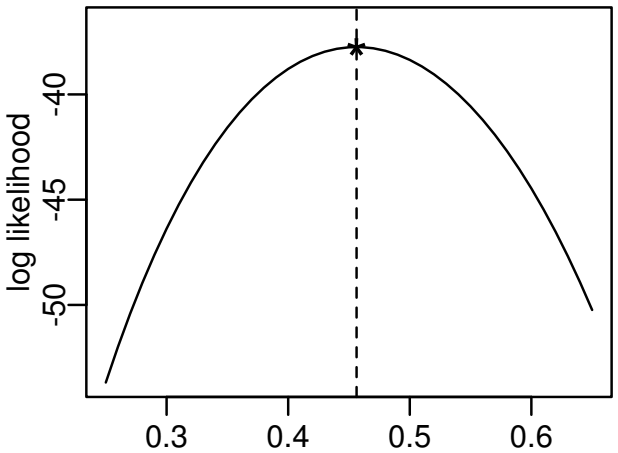
もっともっと長くサンプリングしてみる



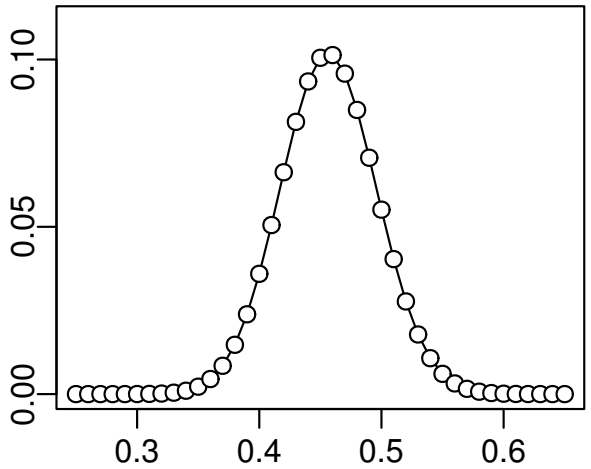
なんだか，ある「山」のかたちにまとまったぞ?

MCMC は何をサンプリングしている?

対数尤度 $\log L(q)$



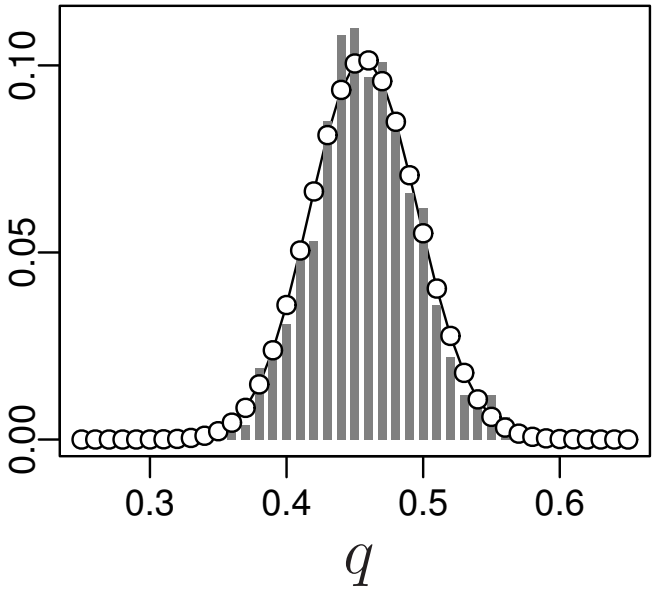
尤度 $L(q)$ に
比例する確率分布



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

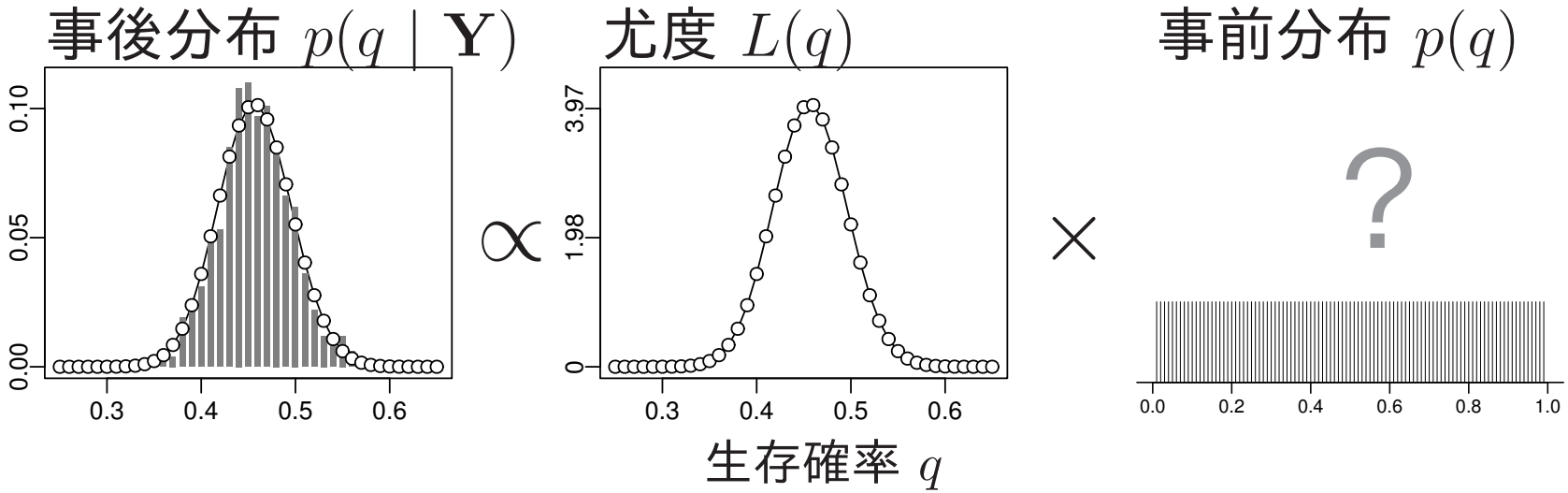
ベイズモデル: 尤度・事後分布・事前分布……

- 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)

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

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

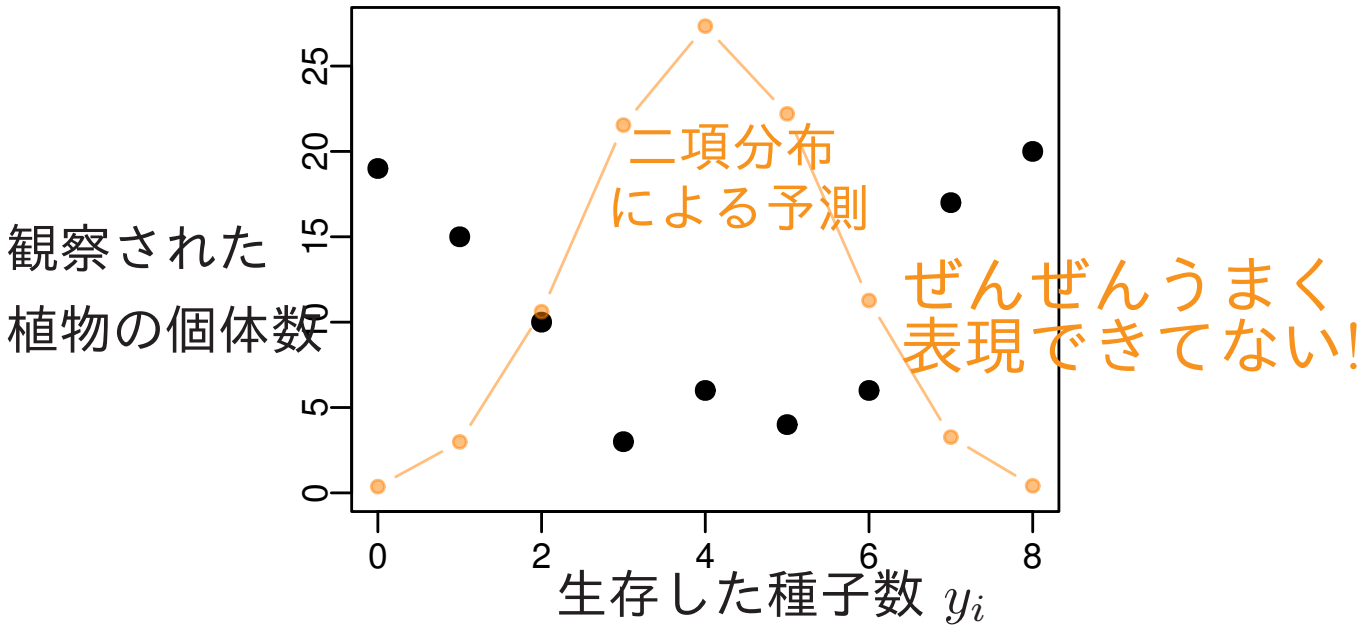
階層ベイズモデル (である GLMM) の導入

(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，

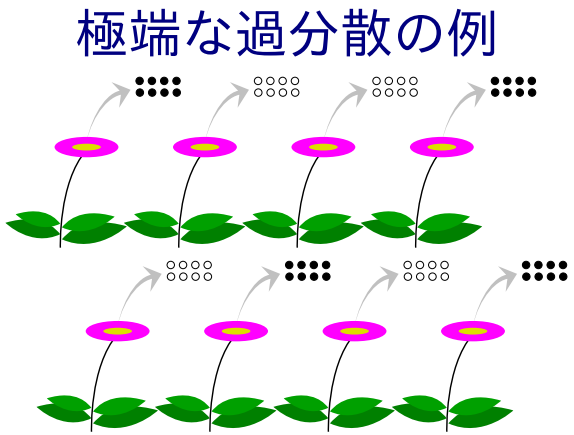
平均生存確率は 0.50 と推定されたが……

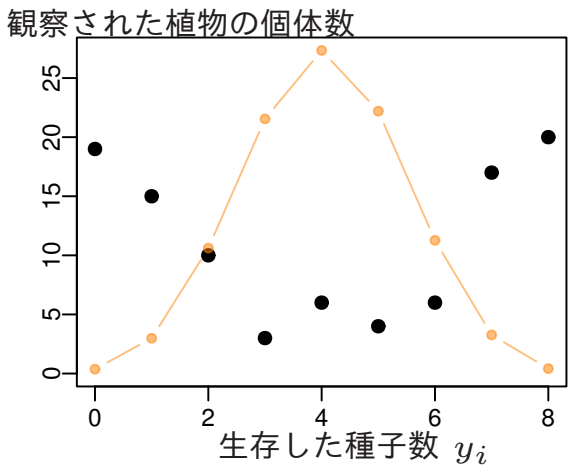


さっきの例題と同じようなデータなのに?

(「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)





- 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

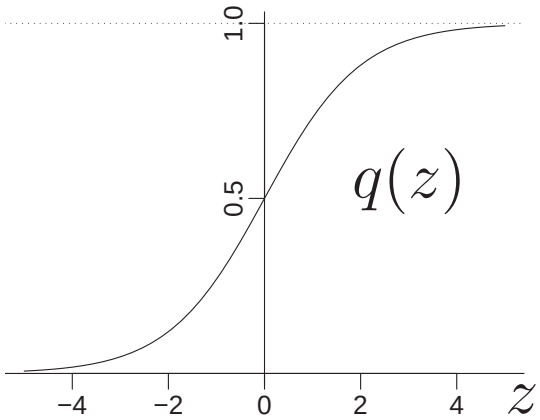
- 生存確率を推定するために 二項分布という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率は二項分布

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

- 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



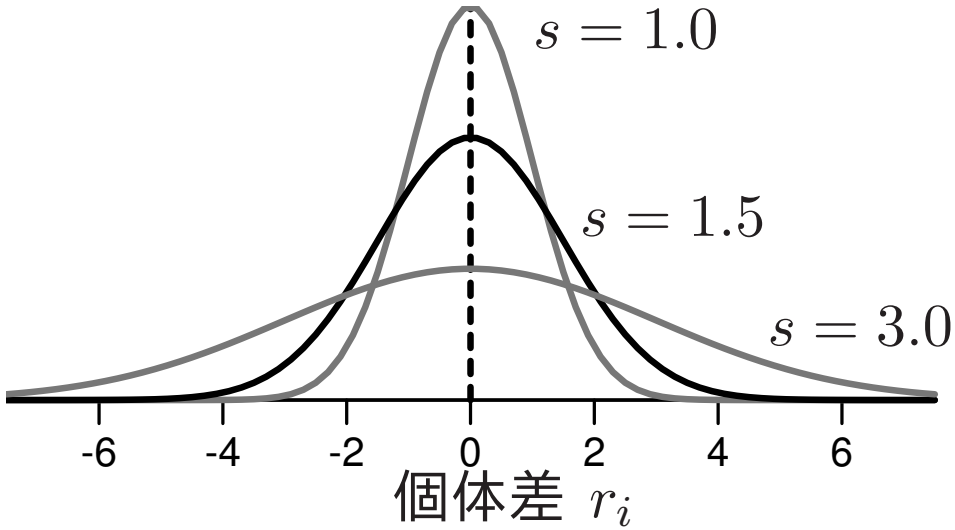
- 線形予測子 $z_i = a + r_i$ とする
  - パラメーター a: 全体の平均
  - パラメーター r_i: 個体 i の個体差（ずれ）

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

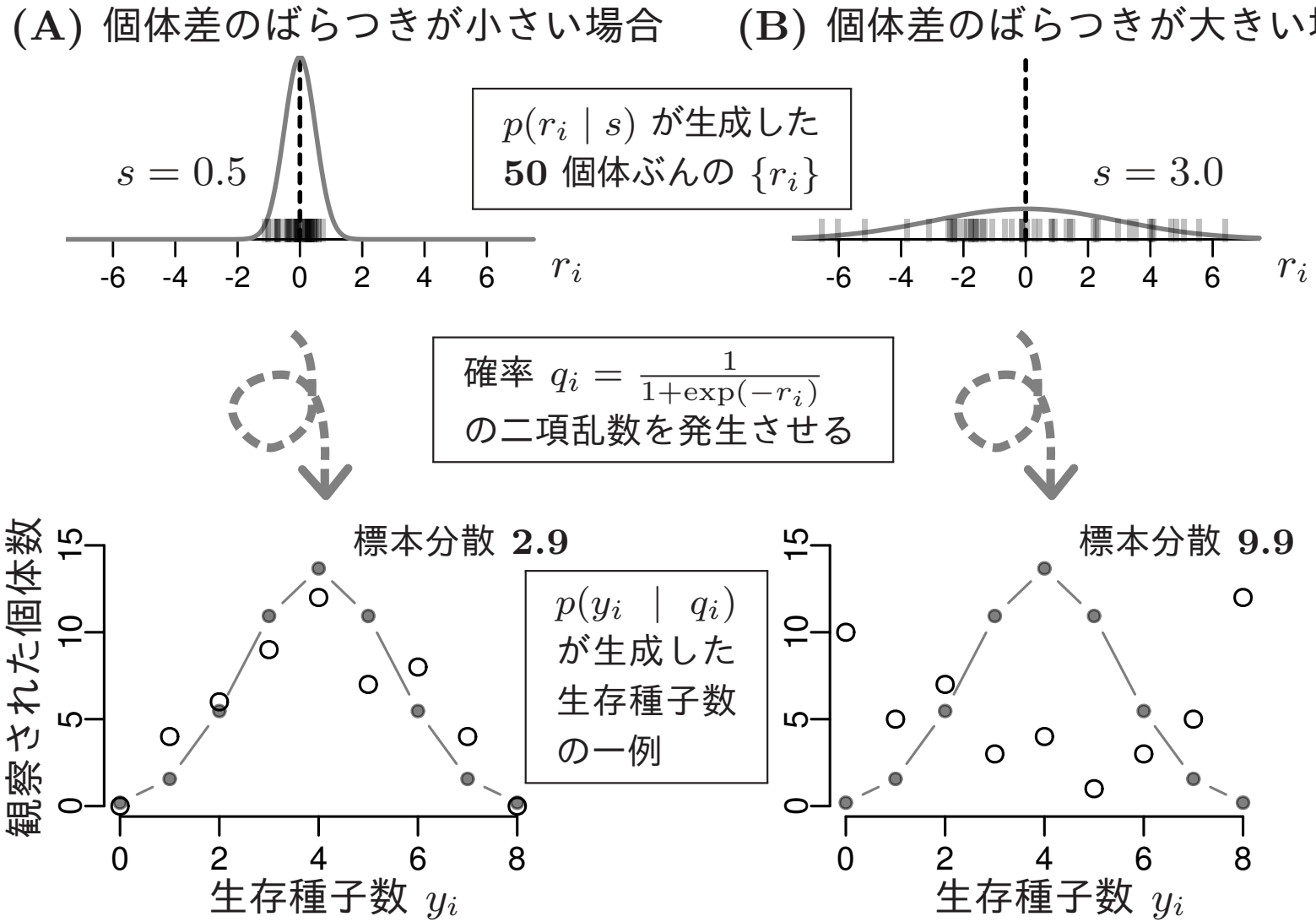
$\{r_i\}$ のばらつきは正規分布だと考えてみる



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

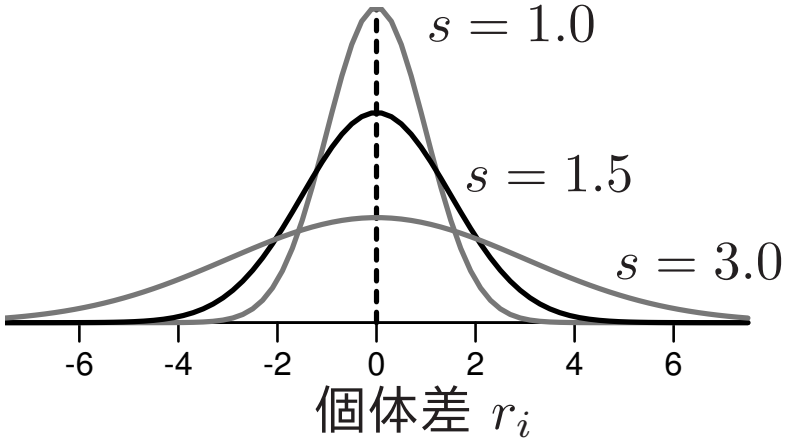
この確率密度 $p(r_i \mid s)$ は r_i の「出現しやすさ」をあらわしていると解釈すればよいでしょう．r_i がゼロにちかい個体はわりと「ありがち」で，r_i の絶対値が大きな個体は相対的に「あまりいない」．

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

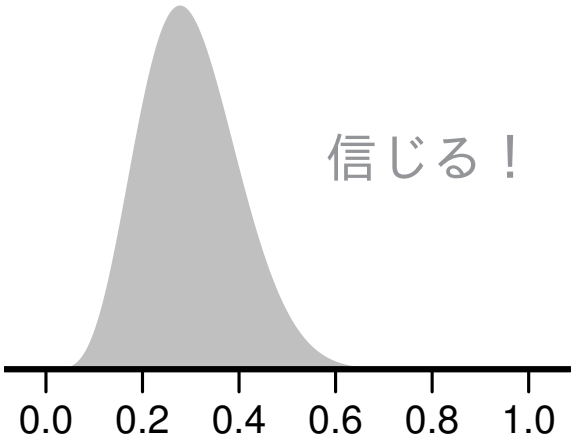


$$p(r_i \mid s) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{r_i^2}{2s^2}\right)$$

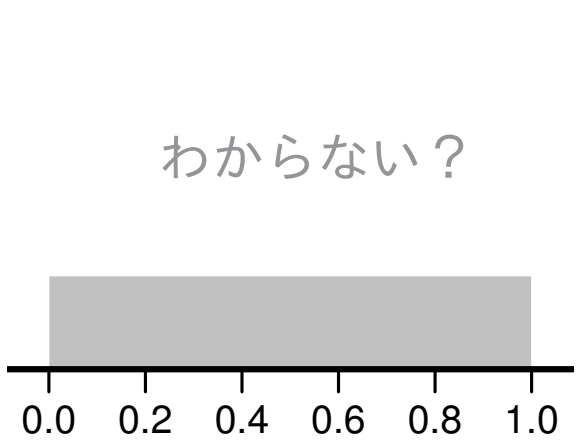
ベイズ統計モデルでよく使われる三種類の事前分布

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．

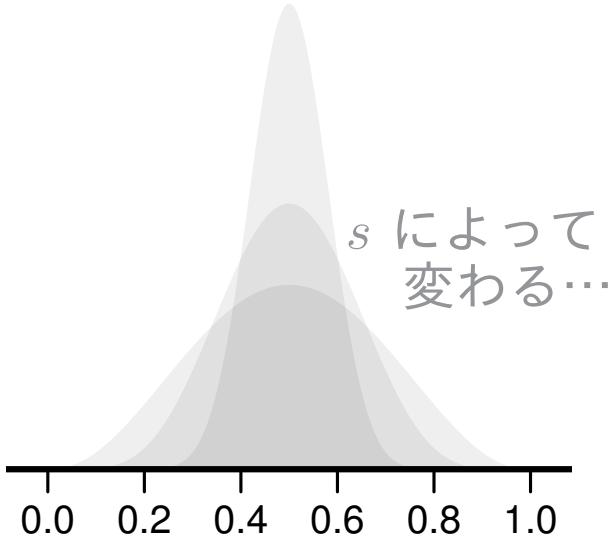
(A) 主観的な事前分布
(できれば使いたくない!)



(B) 無情報事前分布



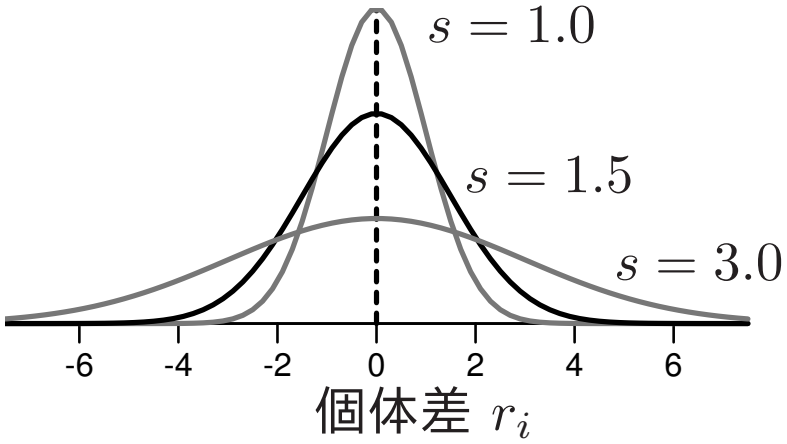
(C) 階層事前分布



r_i の事前分布として階層事前分布を指定する

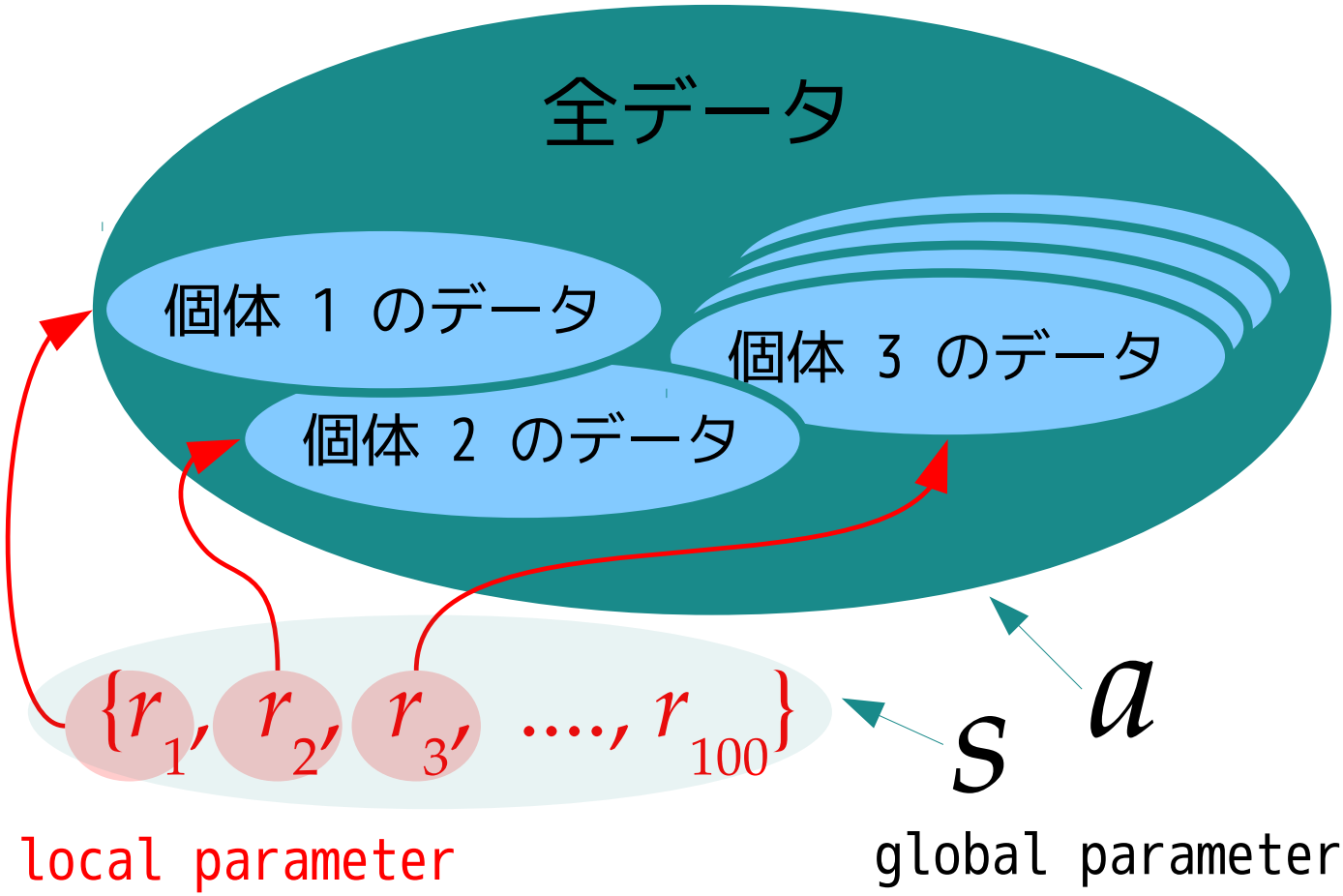
階層事前分布の利点

「データにあわせて」事前分布が変形!



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

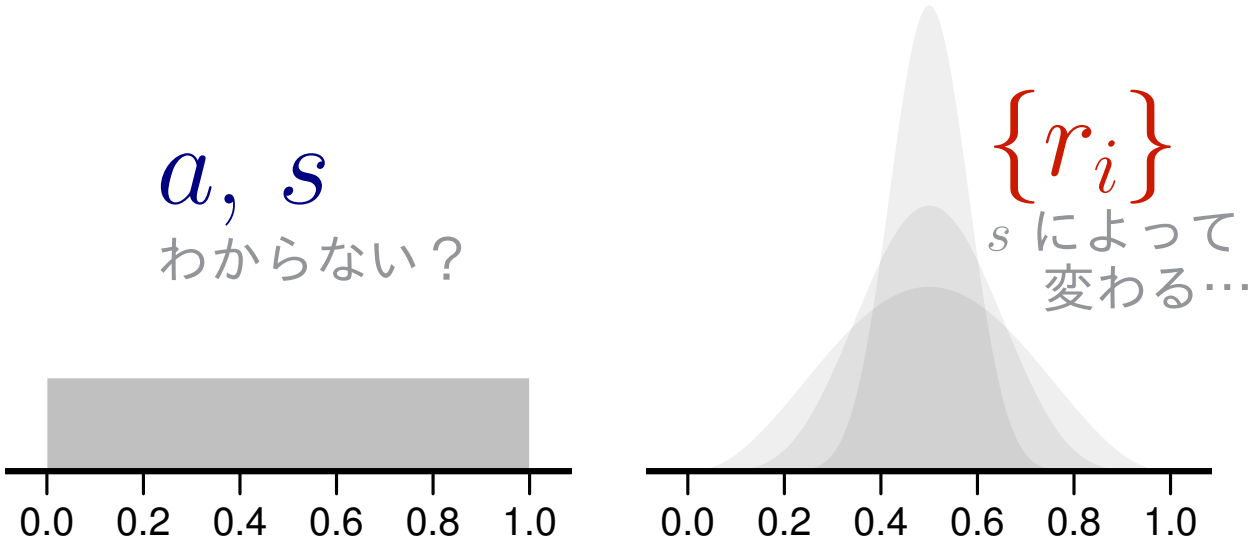
統計モデルの大域的・局所的なパラメーター



データのどの部分を説明しているのか?

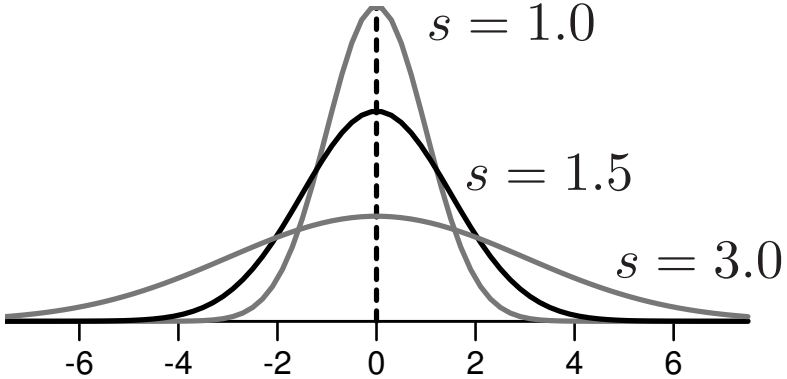
パラメーターごとに適切な事前分布を選ぶ

(B) 無情報事前分布　　(C) 階層事前分布



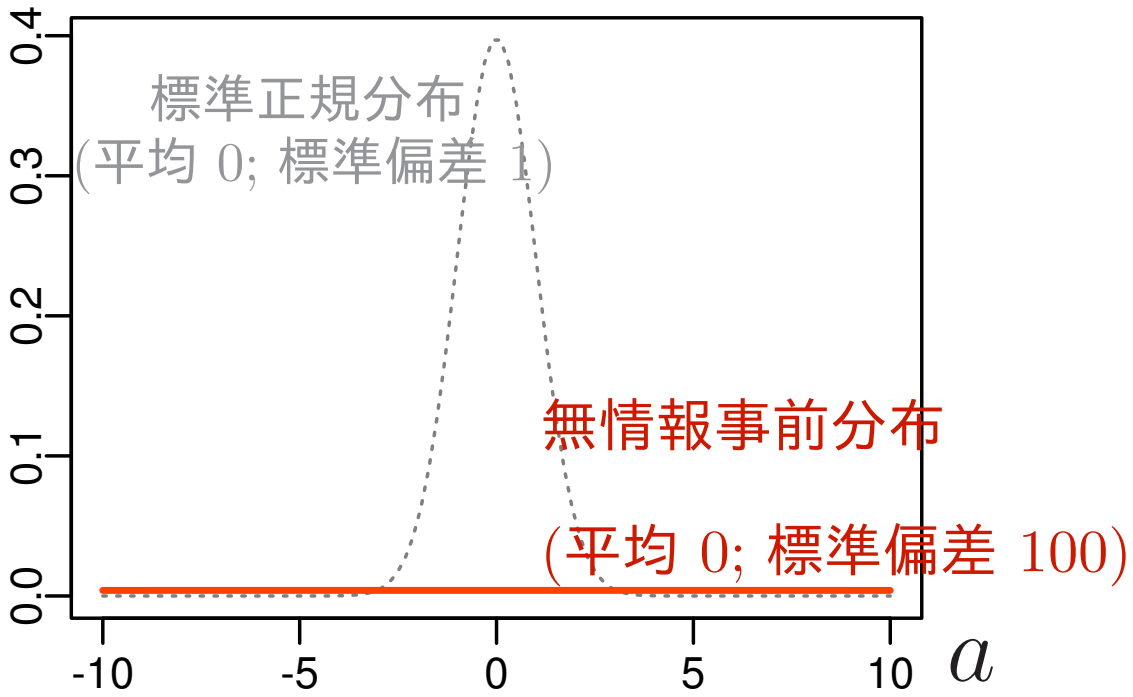
パラメーターの種類	説明する範囲	事前分布
全体に共通する平均・ばらつき	大域的	無情報事前分布
個体・グループごとのずれ	局所的	階層事前分布

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

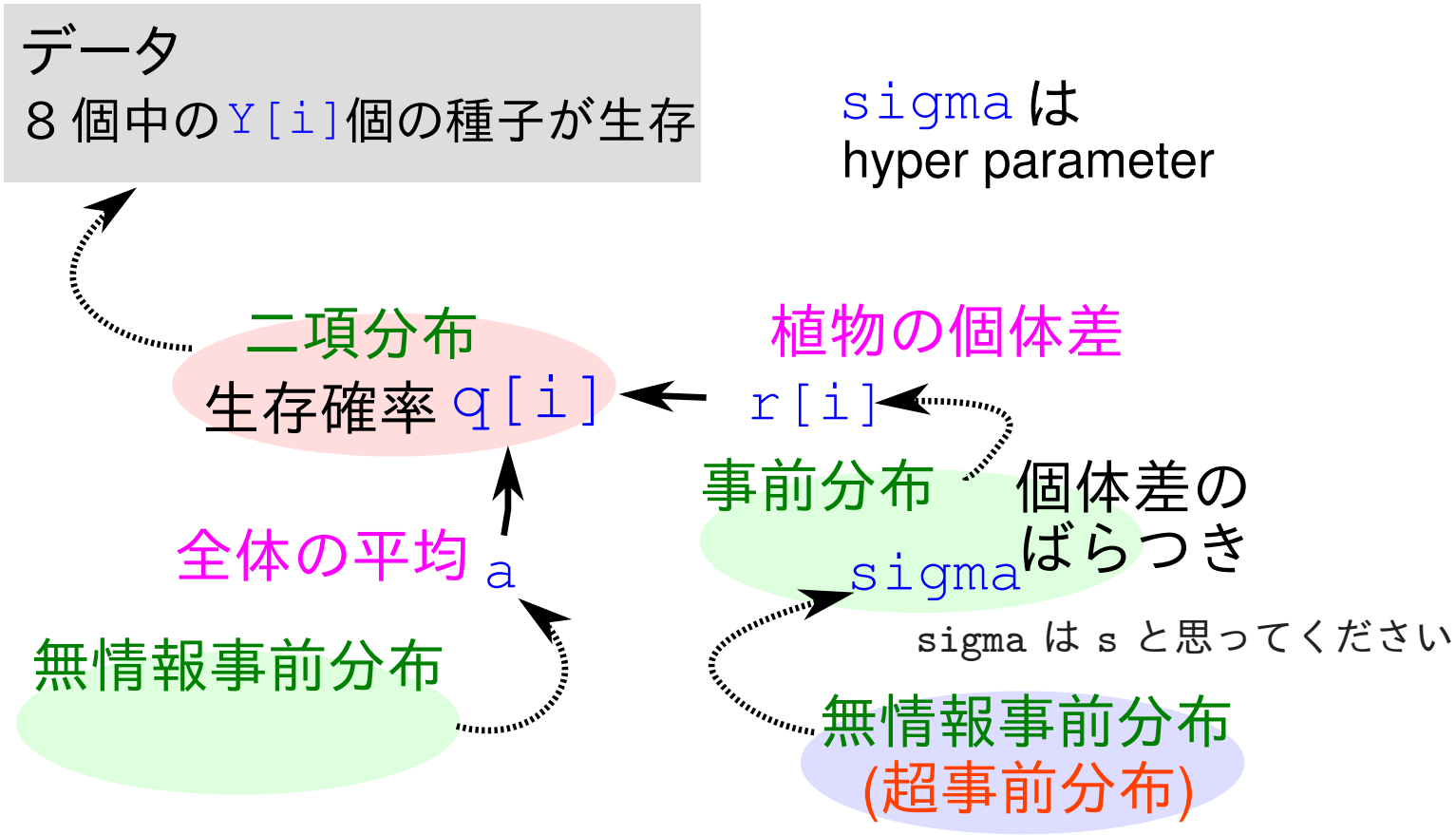
全個体の「切片」 a の無情報事前分布



「生存確率の (logit) 平均 a は何でもよい」と表現している

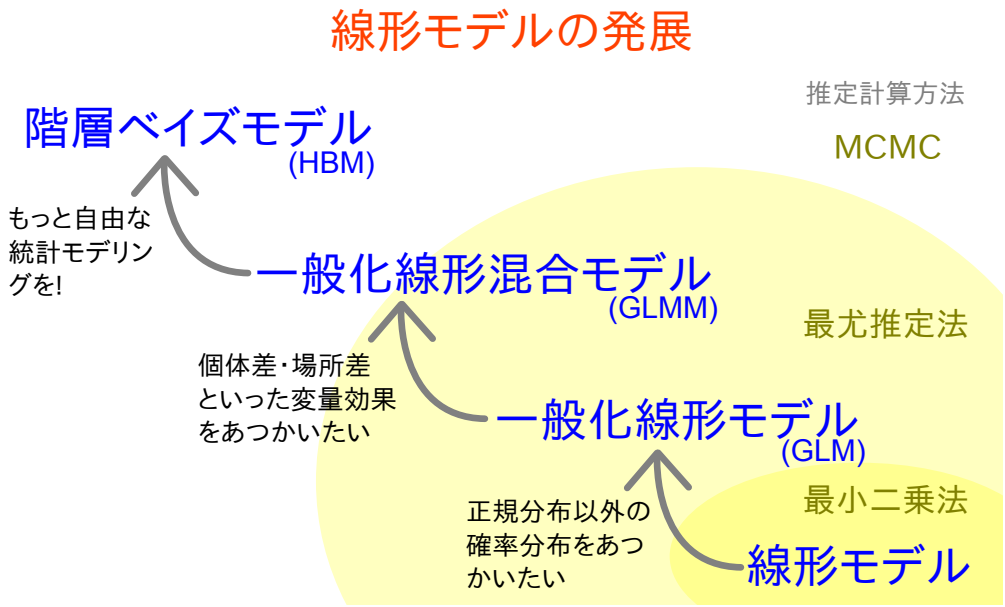
階層ベイズモデル: 事前分布の階層性

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. MCMC のためのソフトウェア

事後分布からサンプリングしたい

Gibbs sampling

統計ソフトウェア R

http://www.r-project.org/

簡単な GLMM なら R だけで推定可能

今回の例題の事後分布 ($\mathbf{Y} = \{y_i\}$ はデータ)

$$p(a, \{r_i\}, s \mid \mathbf{Y}) \propto \prod_{i=1}^{100} p(y_i \mid q(a + r_i))\ p(a)\ p(r_i \mid s)\ p(s)$$

積分で「個体差」 r_i を消して，周辺尤度を定義する

$$L(a, s \mid \mathbf{Y}) = \prod_{i=1}^{100} \int_{-\infty}^{\infty} p(y_i \mid q(a + r_i))\ p(r_i \mid s) dr_i$$

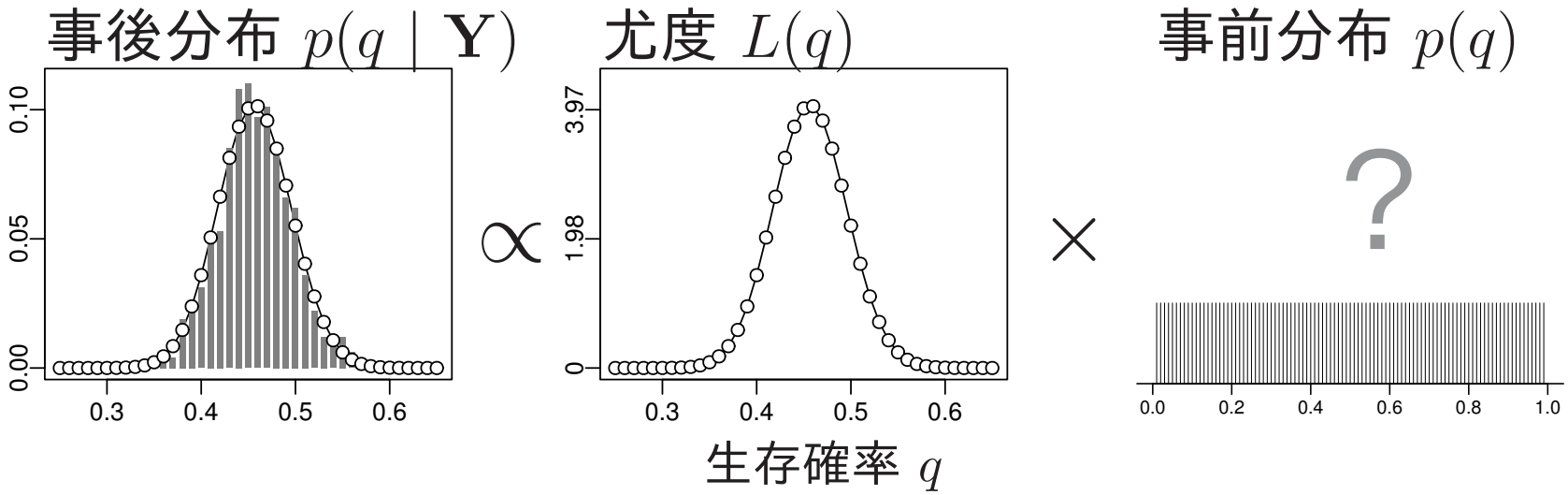
これを最大化する $\hat{a}$ と $\hat{s}$ を推定すればよい

— 経験ベイズ法 (empirical Bayesian method)

しかし，「R だけ」では限界があるかも

- R にはいろいろな GLMM の最尤推定関数が準備されている ……
  - `library(glmmML)` の `glmmML()`
  - `library(lme4)` の `lmer()`
  - `library(nlme)` の `nlme()` (正規分布のみ)
- しかし もうちょっと複雑な **GLMM**，たとえば個体差 + 地域差をいれた統計モデルの最尤推定はかなり難しい (ヘンな結果が得られたりする)
- 積分がたくさん入っている尤度関数の評価がしんどい

そこで MCMC による事後分布からのサンプリング!



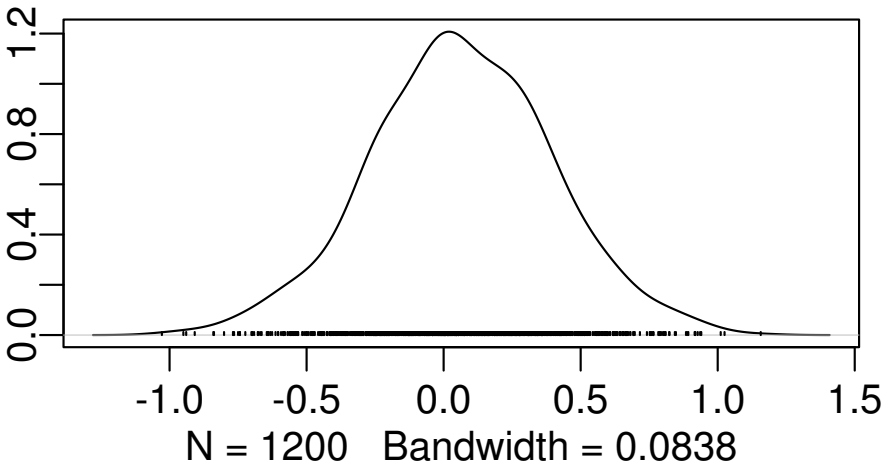
アルゴリズムにしたがって

乱数を発生させていくだけで OK

再確認:「事後分布からのサンプル」って何の役にたつの?

```
> post.mcmc[,"a"] # 事後分布からのサンプルを表示
 [1] -0.7592 -0.7689 -0.9008 -1.0160 -0.8439 -1.0380 -0.8561 -0.9837
 [9] -0.8043 -0.8956 -0.9243 -0.9861 -0.7943 -0.8194 -0.9006 -0.9513
[17] -0.7565 -1.1120 -1.0430 -1.1730 -0.6926 -0.8742 -0.8228 -1.0440
... (以下略) ...
```

これらのサンプルの平均値・中央値・95% 区間を
調べることで事後分布の概要がわかる



どのようなソフトウェアで MCMC 計算するか?

1. 自作プログラム
   - 利点: 問題にあわせて自由に設計できる
   - 欠点: 階層ベイズモデル用の MCMC プログラミング，けっこうめんどう
2. R のベイズな package
   - 利点: 空間ベイズ統計など便利な専用 package がある
   - 欠点: 汎用性，とぼしい
3. **「できあい」の Gibbs sampler ソフトウェア**
   - 利点: 幅ひろい問題に適用できて，便利
   - 欠点: 「まちがいさがし」 (debug) がめんどう

MCMC の使い方を勉強しよう

いろいろな MCMC

- メトロポリス法: 試行錯誤で値を変化させていく MCMC
  - メトロポリス・ヘイスティングス法: その改良版
- ギブス・サンプリング: 条件つき確率分布を使った MCMC
  - 複数の変数 (パラメーター・状態) を効率よくサンプリング

Gibbs sampling とは何か?

- MCMC アルゴリズムのひとつ
- 複数のパラメーターの MCMC サンプリングに使う
- 例: パラメーター β_1 と β_2 の Gibbs sampling
  1. β_2 に何か適当な値を与える
  2. β_2 の値はそのままにして，その条件のもとでの β_1 の MCMC sampling をする (条件つき事後分布)
  3. β_1 の値はそのままにして，その条件のもとでの β_2 の MCMC sampling をする (条件つき事後分布)
  4. 2. – 3. をくりかえす
- 教科書の第 9 章の例題で説明

この例題の事後分布は?

$$p(a, \{r_i\}, s \mid \text{データ}) = \frac{\prod_{i=1}^{100} p(y_i \mid q(a + r_i))\ p(a)\ p(r_i \mid s)\ p(s)}{\iint \cdots \int (\text{分子} \uparrow \text{そのまま})\ dr_i\ ds\ da}$$

分母は何か定数になるので

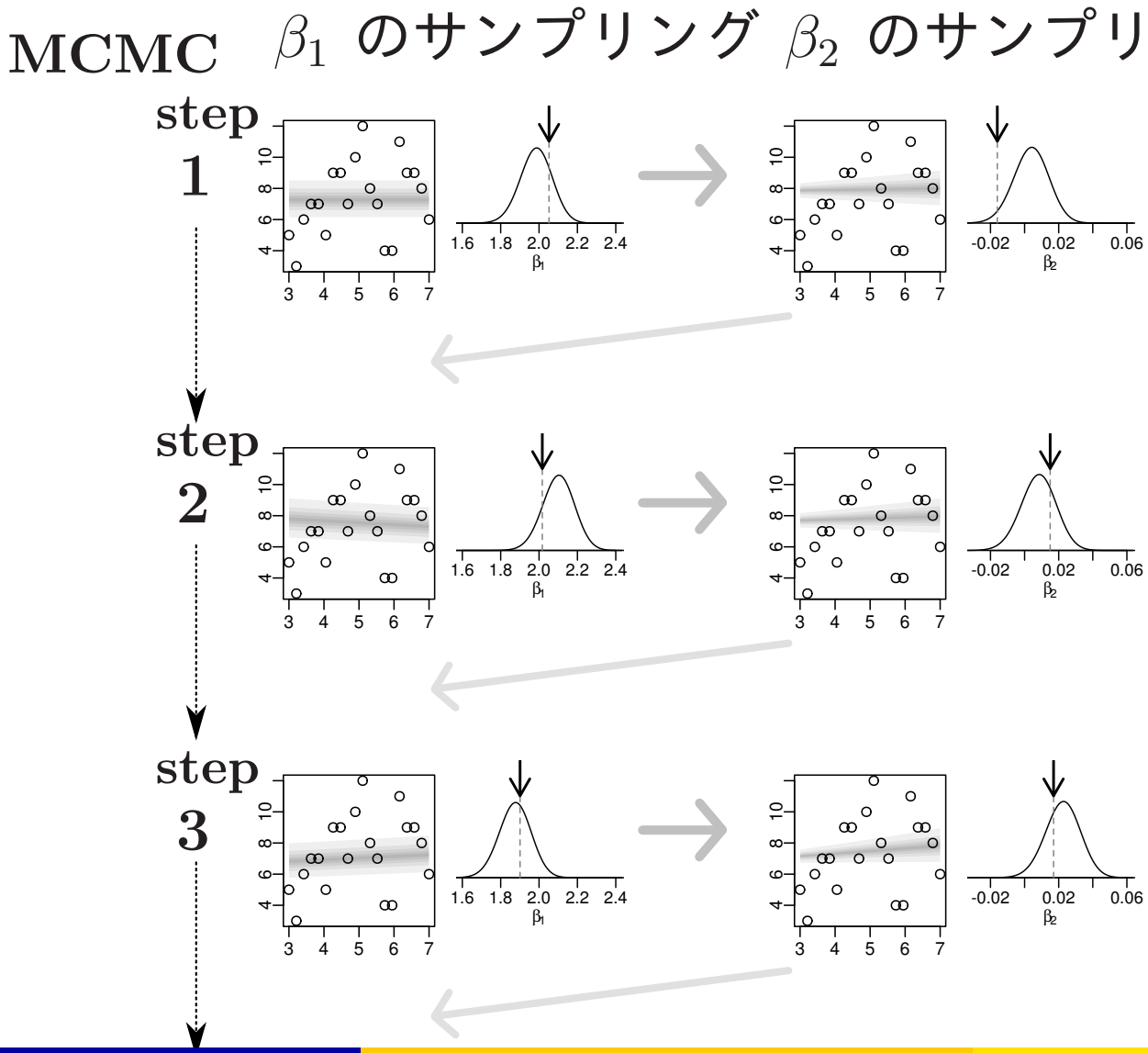
$$p(a, \{r_i\}, s \mid \text{データ}) \propto \prod_{i=1}^{100} p(y_i \mid q(a+r_i))\ p(a)\ p(r_i \mid s)\ p(s)$$

この事後分布から Gibbs sampling してみる

サンプリングの対象とするパラメーター以外は値を固定する

$$
\begin{aligned}
p(a \mid \cdots) &\propto \prod_{i=1}^{100} p(y_i \mid q(a + r_i))\ p(a) \\
p(s \mid \cdots) &\propto \prod_{i=1}^{100} p(r_i \mid s)\ p(s) \\
p(r_1 \mid \cdots) &\propto p(y_1 \mid q(a + r_1))\ p(r_1 \mid s) \\
p(r_2 \mid \cdots) &\propto p(y_2 \mid q(a + r_2))\ p(r_2 \mid s) \\
&\vdots \\
p(r_{100} \mid \cdots) &\propto p(y_{100} \mid q(a + r_{100}))\ p(r_{100} \mid s)
\end{aligned}
$$

図解: Gibbs sampling (統計モデリング入門の第 9 章)



5. 階層ベイズモデルの推定

ソフトウェア WinBUGS を使ってみる

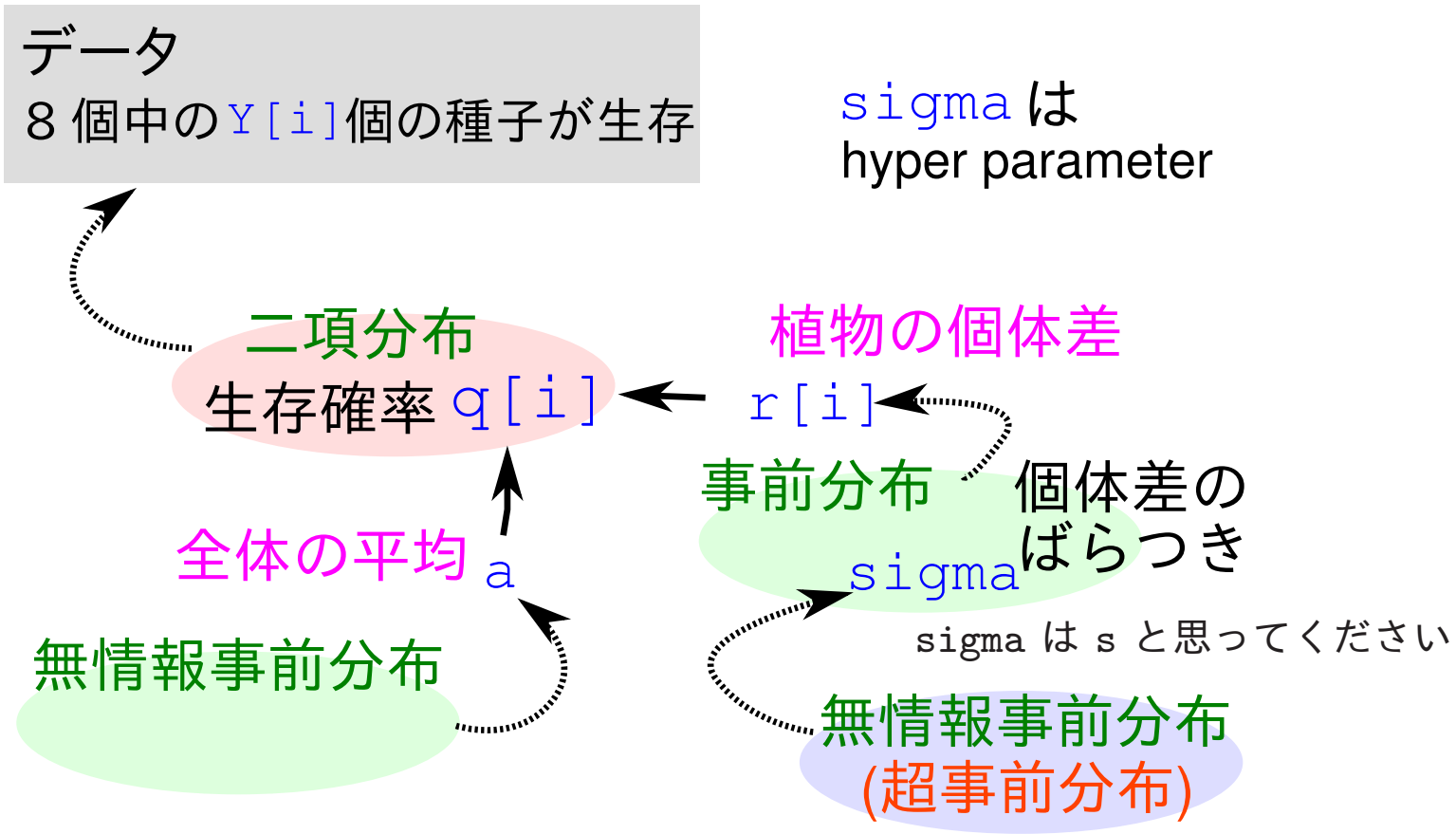
BUGS 言語で統計モデルを指定，R と連携する

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)
}
```

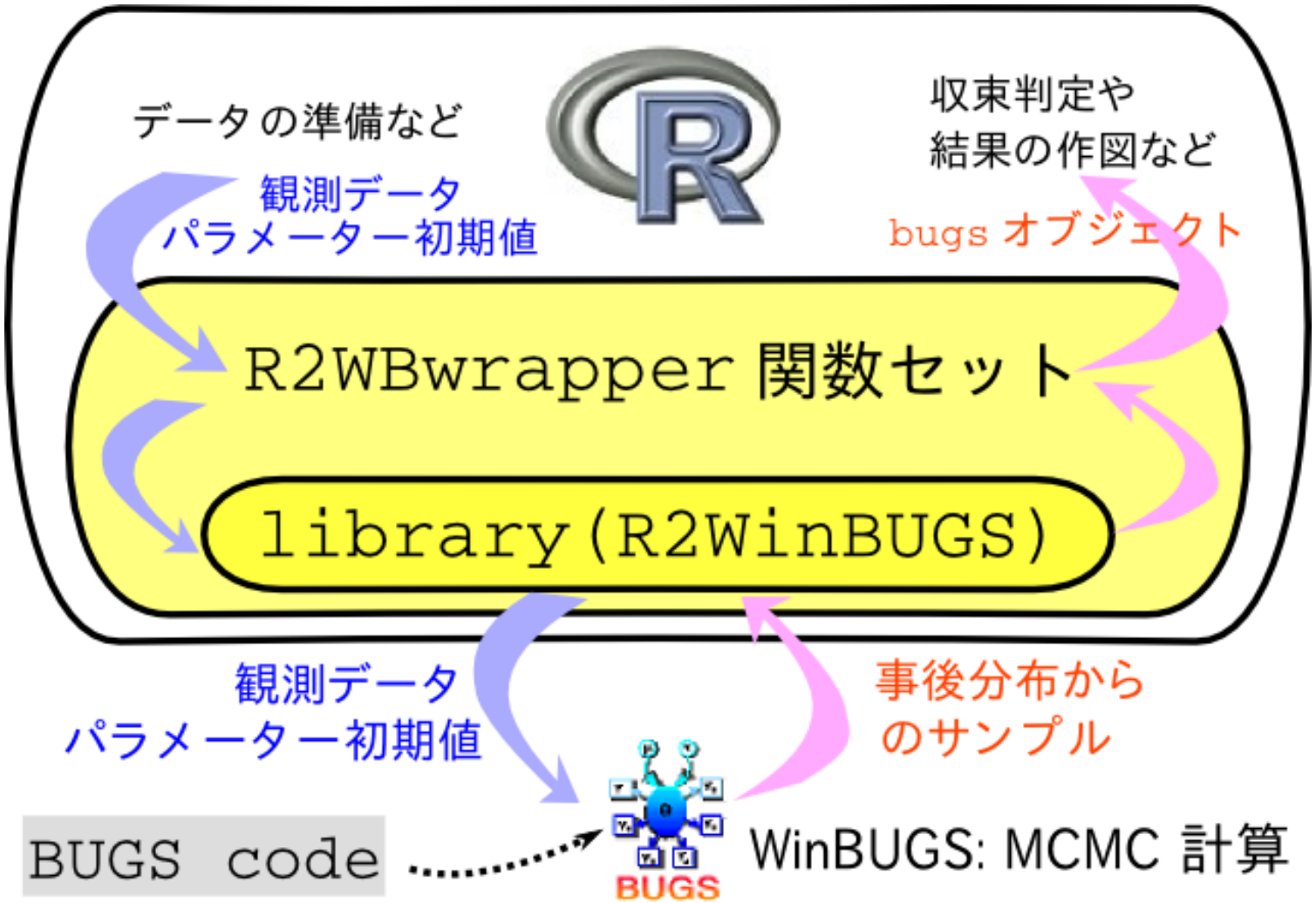
なんとなく使われ続けている WinBUGS 1.4.3

- 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

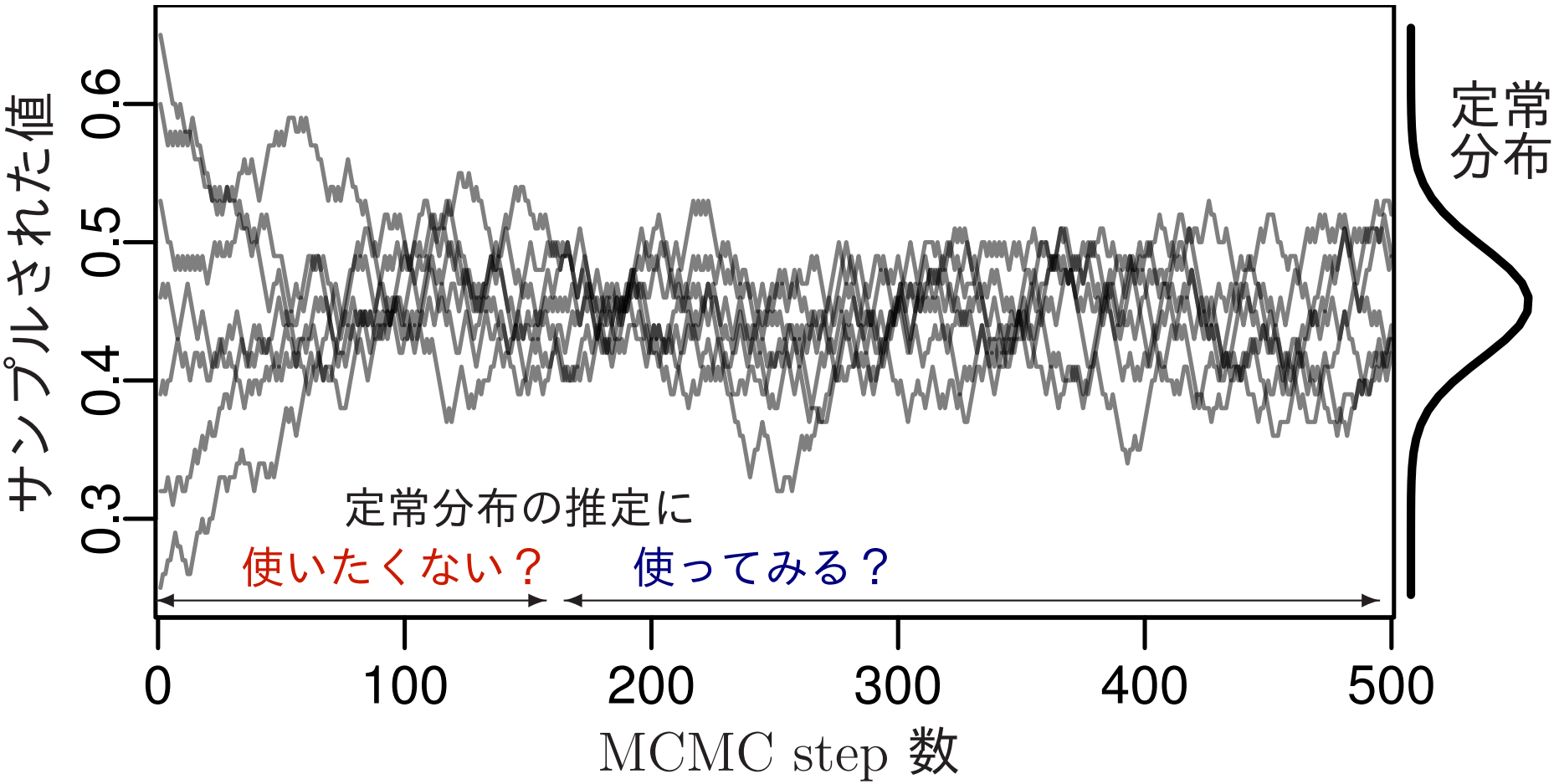
R2WBwrapper 経由で WinBUGS を使う

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



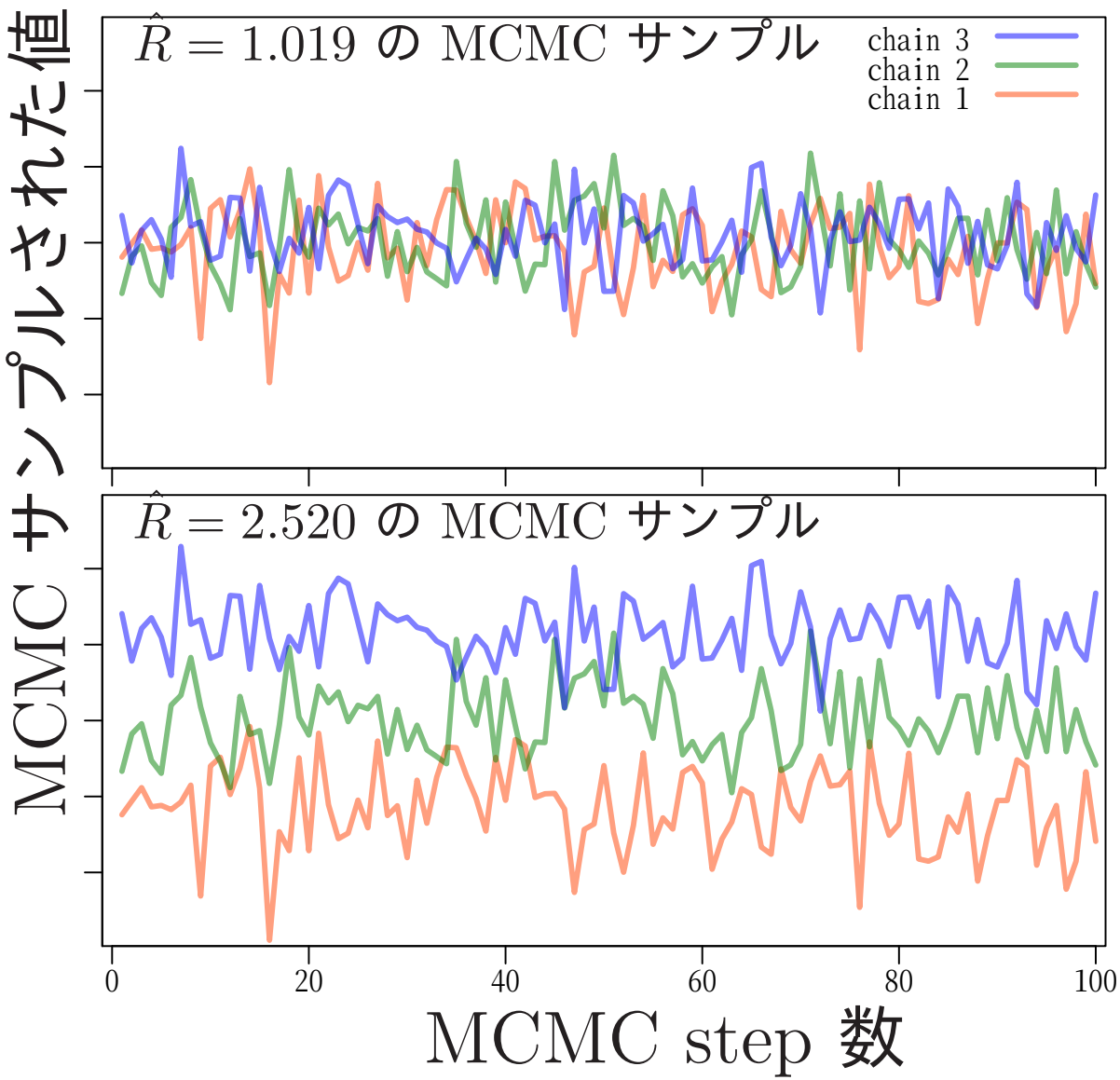
事後分布サンプルが得られたら → あれこれ図表を

- `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

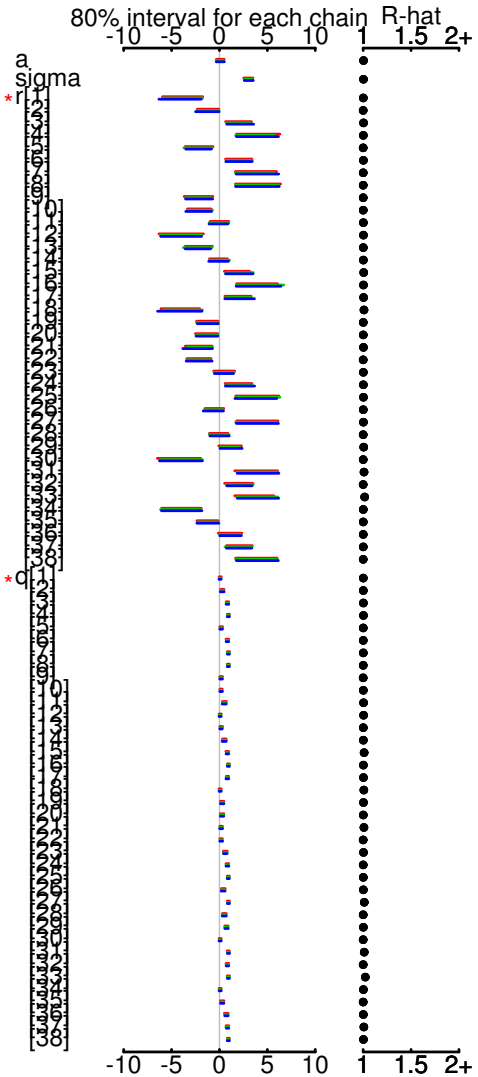


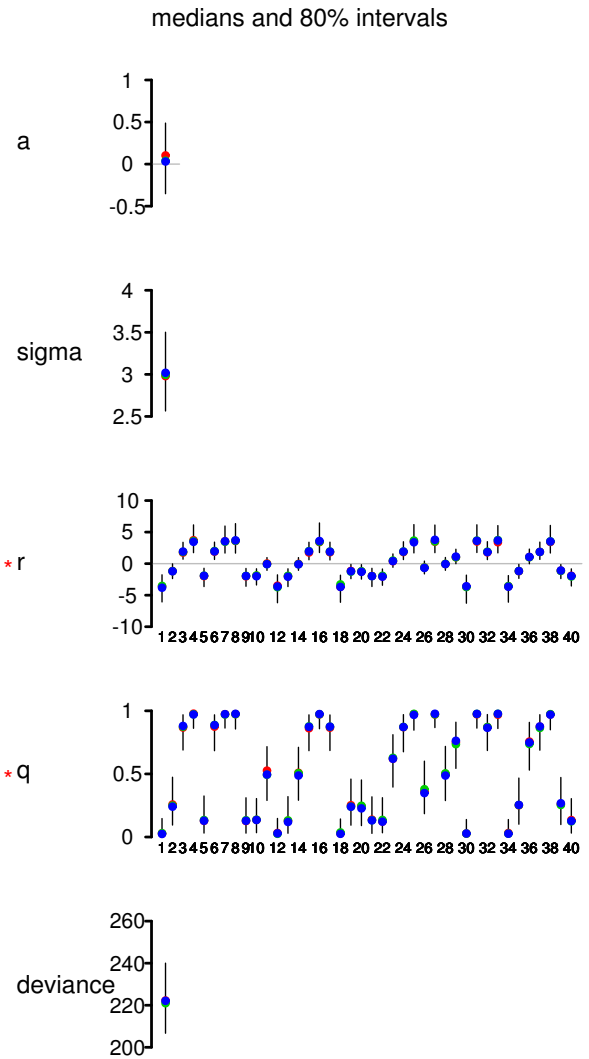
まあ，
いいかな……

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





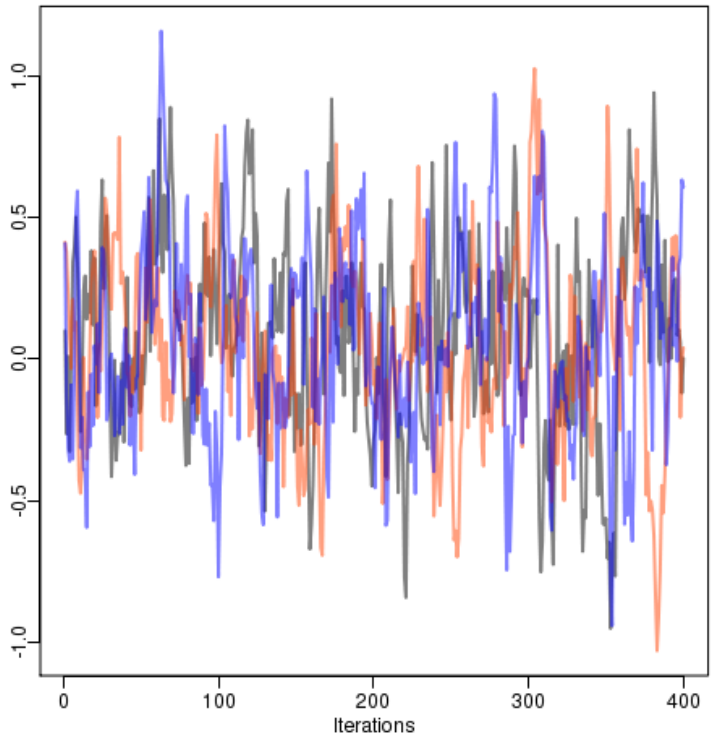
bugs オブジェクトの post.bugs を調べる

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

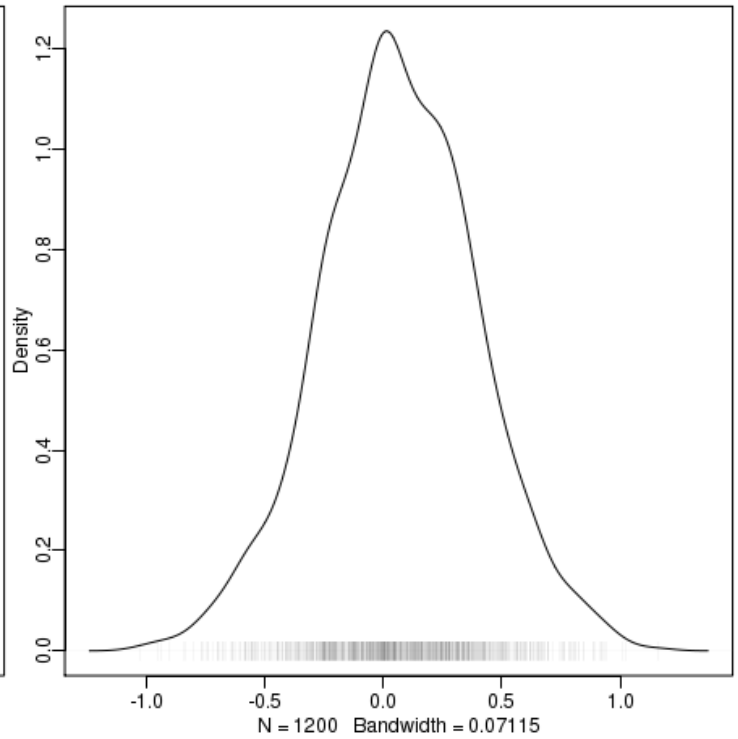
	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
a	0.031	0.357	-0.718	-0.187	0.041	0.268	0.682	1.034	72
sigma	3.060	0.376	2.365	2.807	3.029	3.288	3.830	1.002	1200
r[1]	-3.890	1.903	-8.238	-4.918	-3.514	-2.546	-1.174	1.001	1200
r[2]	-1.190	0.905	-3.137	-1.763	-1.159	-0.559	0.438	1.007	290
r[3]	2.062	1.128	0.185	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

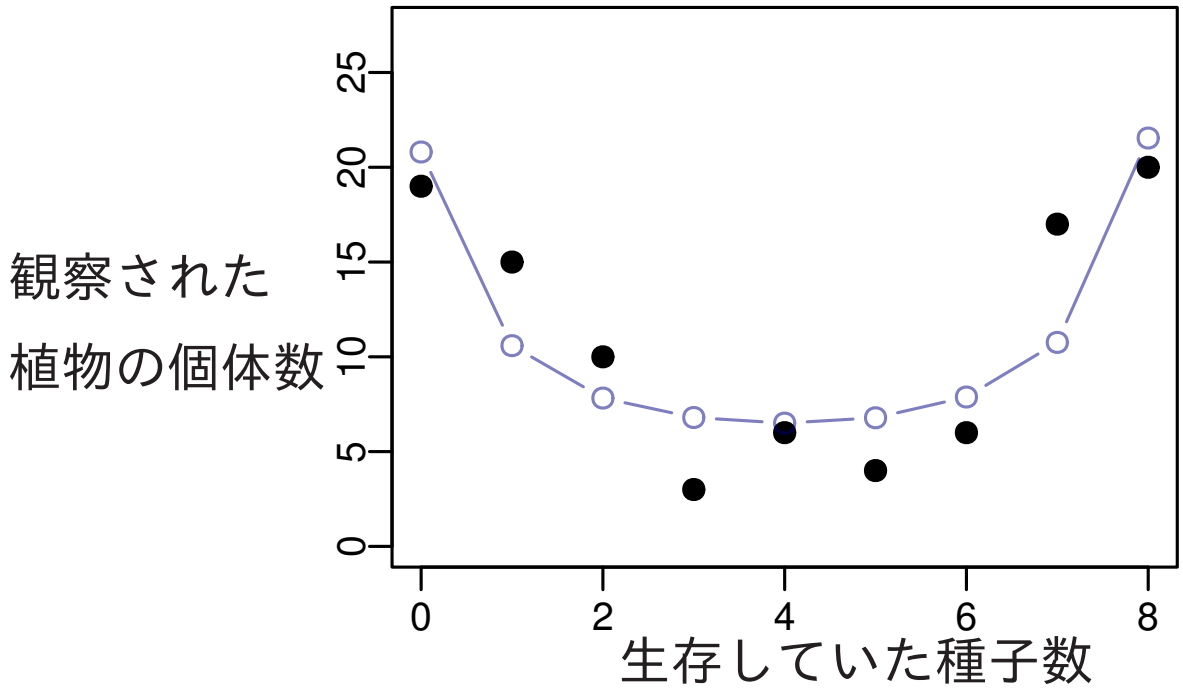


a の事後確率密度の推定



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

個体差 r_i について積分する

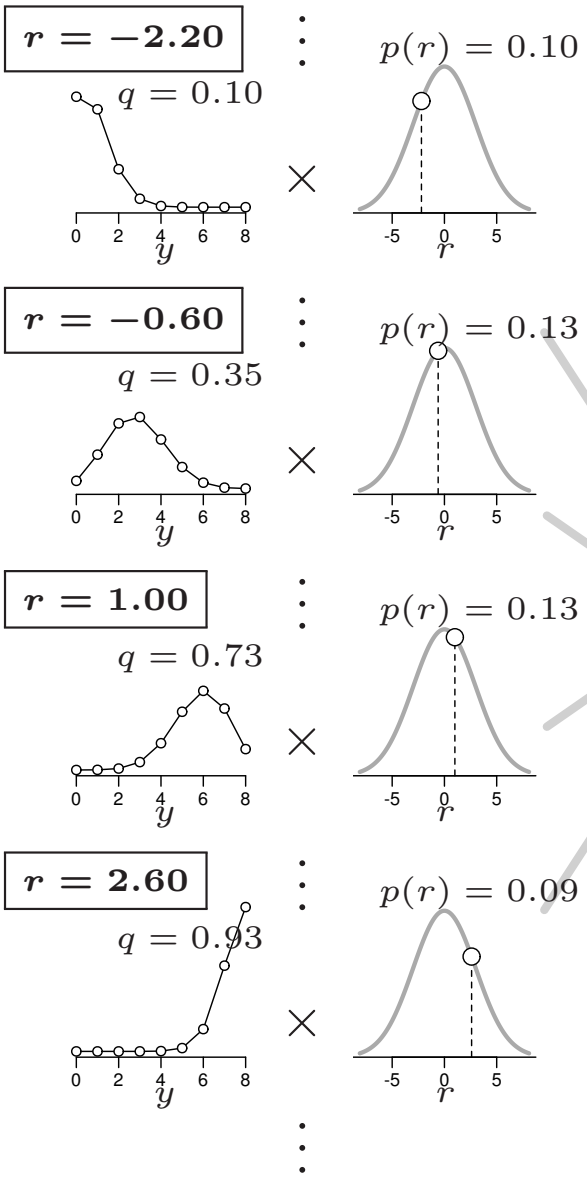
ということは

二項分布と正規分布をまぜ

あわせること

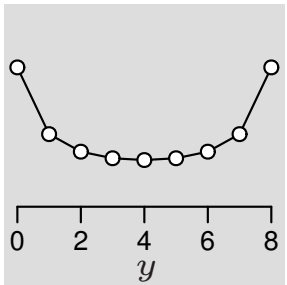
個体差 r ごとに異なる二項分布

集団内の r の分布
重み $p(r \mid s)$

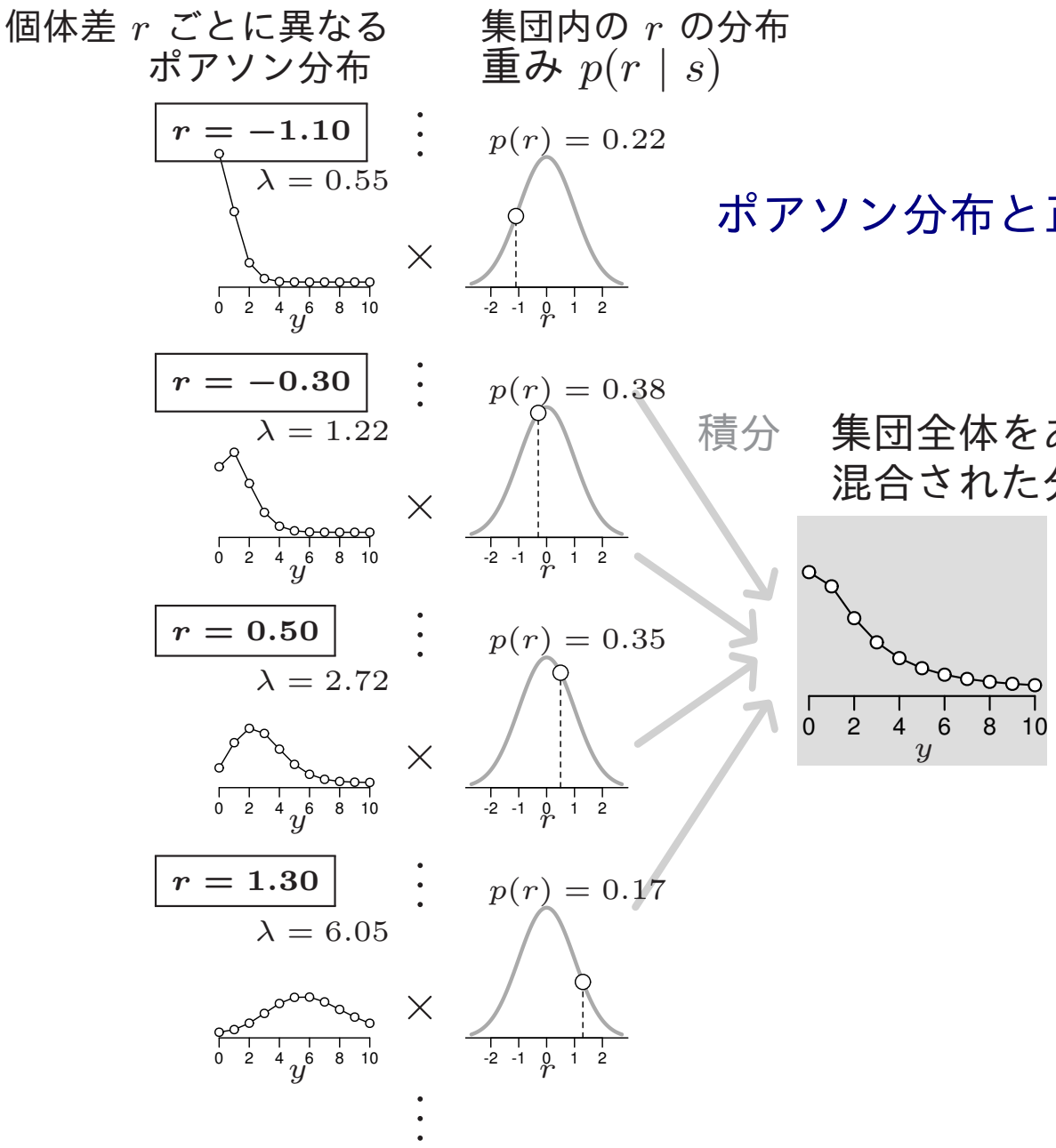


積分

集団全体をあらわす混合された分布



二項分布と正規分布のまぜあわせ



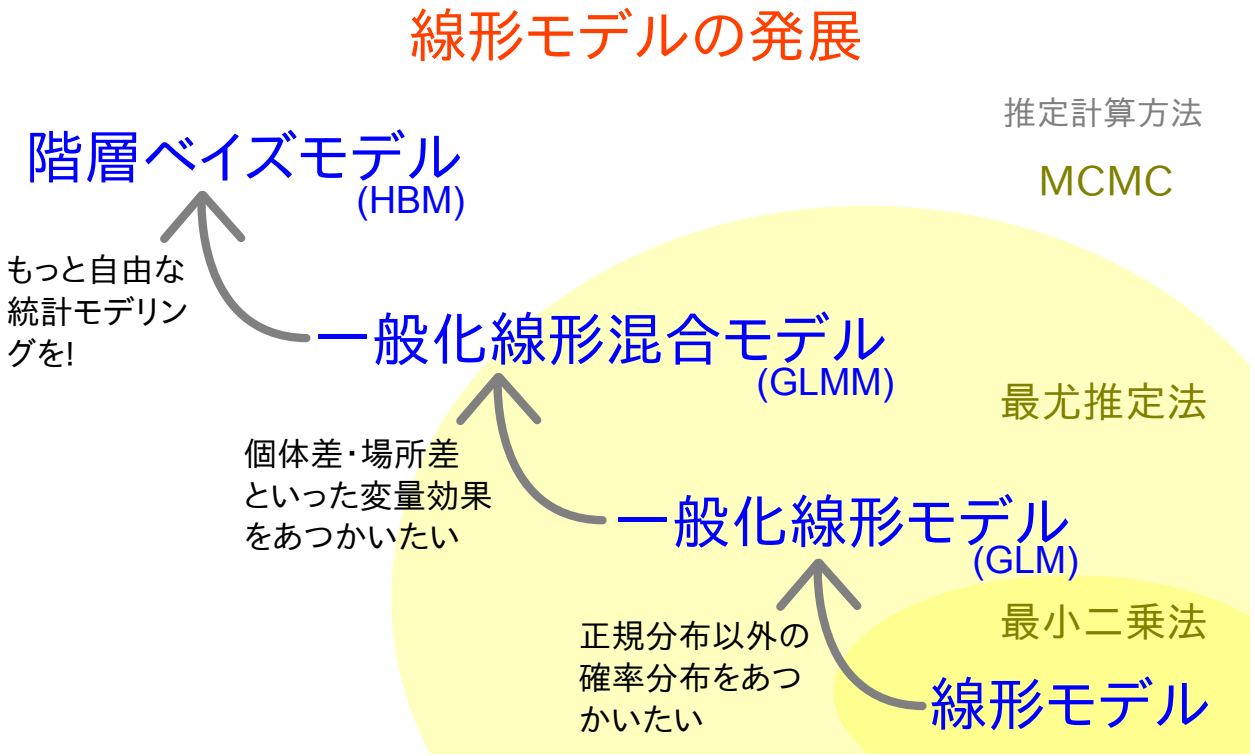
ポアソン分布と正規分布のまぜあわせ

さてさて，

ややとうとつでは

ありますが……

ここでこの「入門」はひとまず終了します



- 線形モデルを階層ベイズモデルに発展させよう
  - 現実のデータの複雑さに対応するために!
- 複雑な階層ベイズの事後分布は MCMC で推定しよう

みなさん,

夜おそくまで

おつきあいいただき,

ありがとうございました