漁業統計検討会 (清水)

「統計モデリングセミナー」 (2012 年 12 月) 投影資料

全部で 7 回中の 4a 回目

階層ベイズモデルの基礎

個体差のモデリング

久保拓弥 kubo@ees.hokudai.ac.jp

http://goo.gl/OyB2k

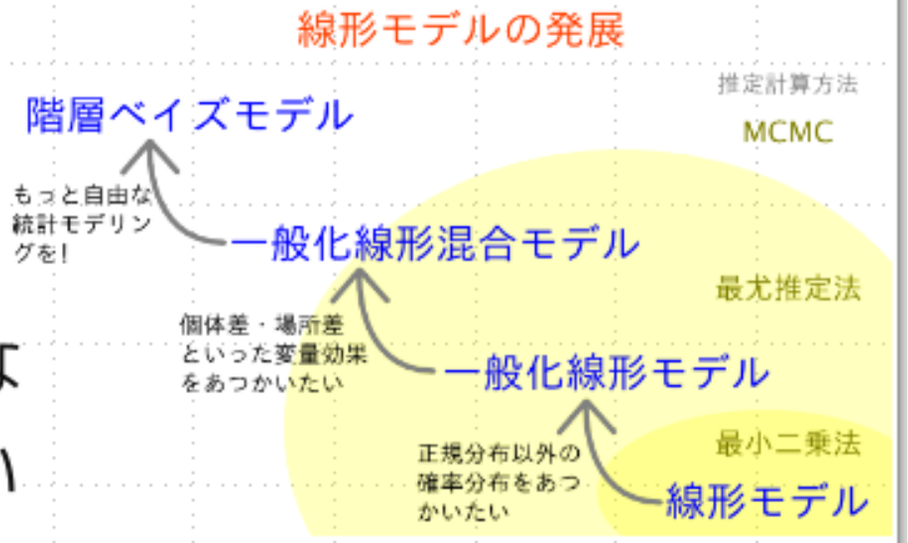
今回のハナシ

階層ベイズモデルでないとうまく表現できない現象がある

- 統計モデルは尤度であてはまりの良さを調べられる
  - 一番あてはまりが良いところをさがすのが最尤推定法
  - 尤度に比例するパラメーターの確率分布を推定するのが **MCMC**
- 現実的なデータ解析では個体差など **random effects** を考慮する必要あり
  - 単純な GLM ではそういうばらつきを表現できない
  - 階層ベイズモデル!

なぜ階層ベイズモデルまですすむのか?

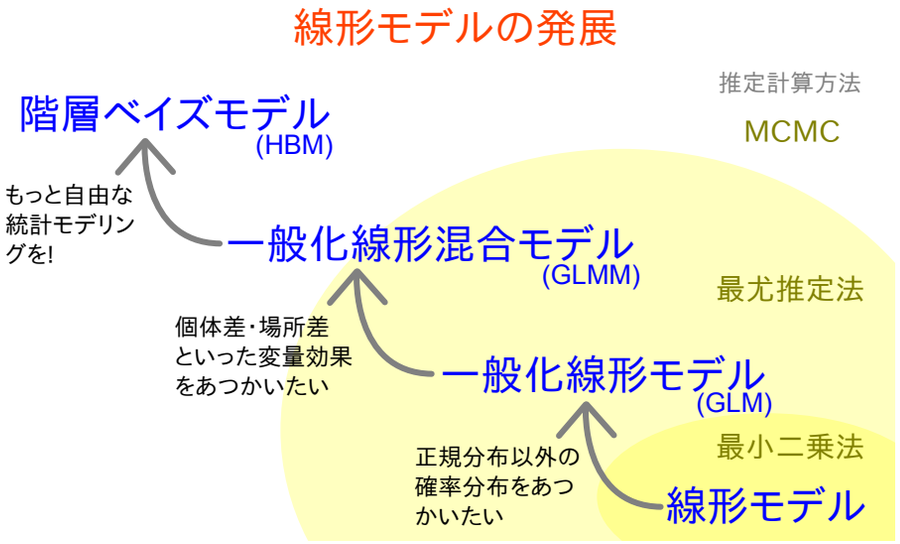
- 生態学や漁業のデータ解析は難しいから!
  - 個体差・エリア差・空間相関・時間相関・種差などめんどうなことをあつかわないといけない



- そういう難しい状況では，ベイズモデル化し，パラメーターの事後分布を MCMC 法を使って推定するのが無難でしょう

今回，説明しようとすること

- パラメーターをどうやって推定するか?
  - 最尤推定 $\rightarrow$ **Markov chain Monte Carlo (MCMC)**
- 一般化線形モデル $\rightarrow$ 階層ベイズモデル
  - より現実的・実戦的なデータ解析のために



このあとの統計モデル化の説明の手順

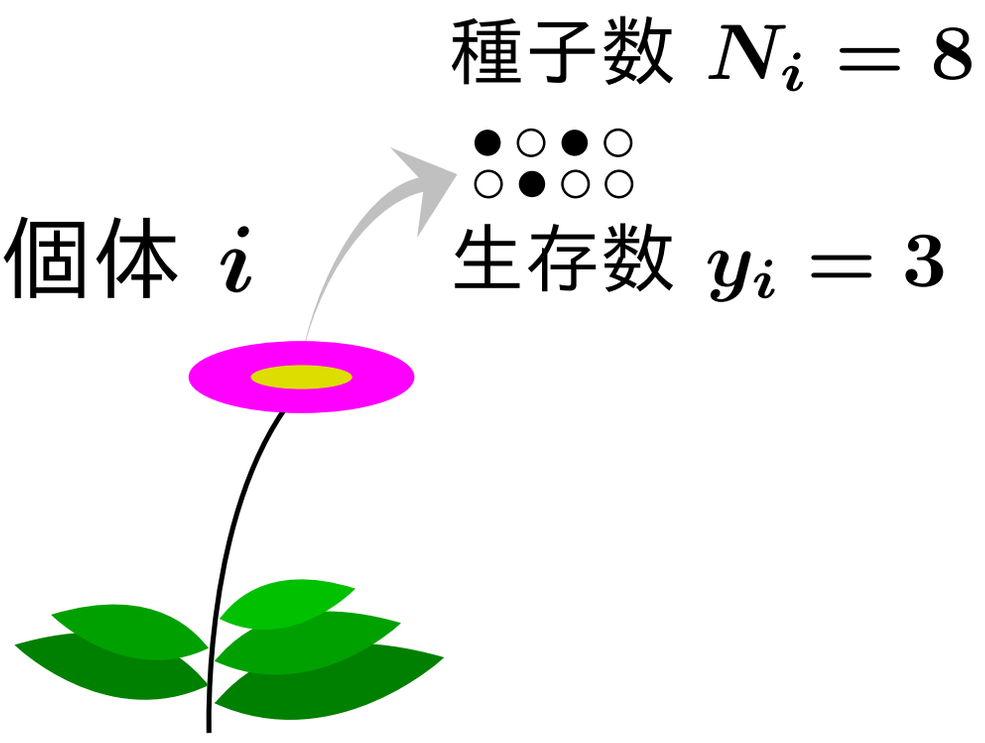
1. 簡単な例題: GLM でうまくいく場合
   - 統計モデルの部品: 二項分布モデル (GLM)
   - 統計モデルの推定方法: 最尤推定法
2. MCMC とベイズモデリング
   - 最尤推定を MCMC におきかえてみる
   - MCMC で得られた結果をベイズ的に解釈
3. ちょっと難しい例題: GLM でうまくいかない場合
   - 「個体全体の平均」と「個体差」をどうあつかう?
   - 階層ベイズモデル!

今回の例題: 植物の種子と二項分布

— 最尤推定の復習 —

繁殖生態学の例題: 架空植物の生存確率

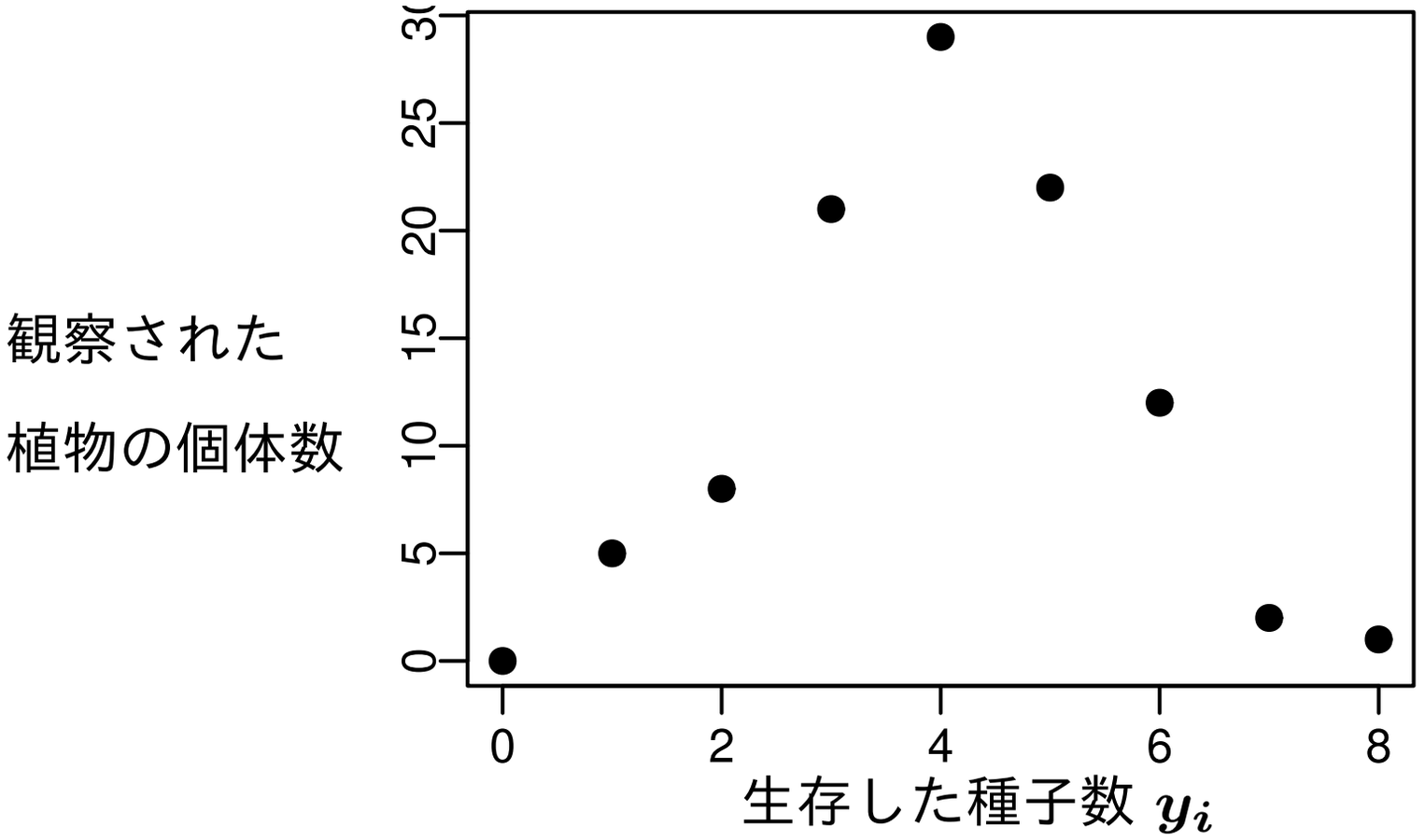
- 架空植物の種子の生存を調べた
- 種子: 生きていれば発芽する
  - どの個体でも **8** 個 の種子を調べた
- 生存確率: ある種子が生きている確率



- 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?

簡単な例題: 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



生存確率 q と二項分布の関係

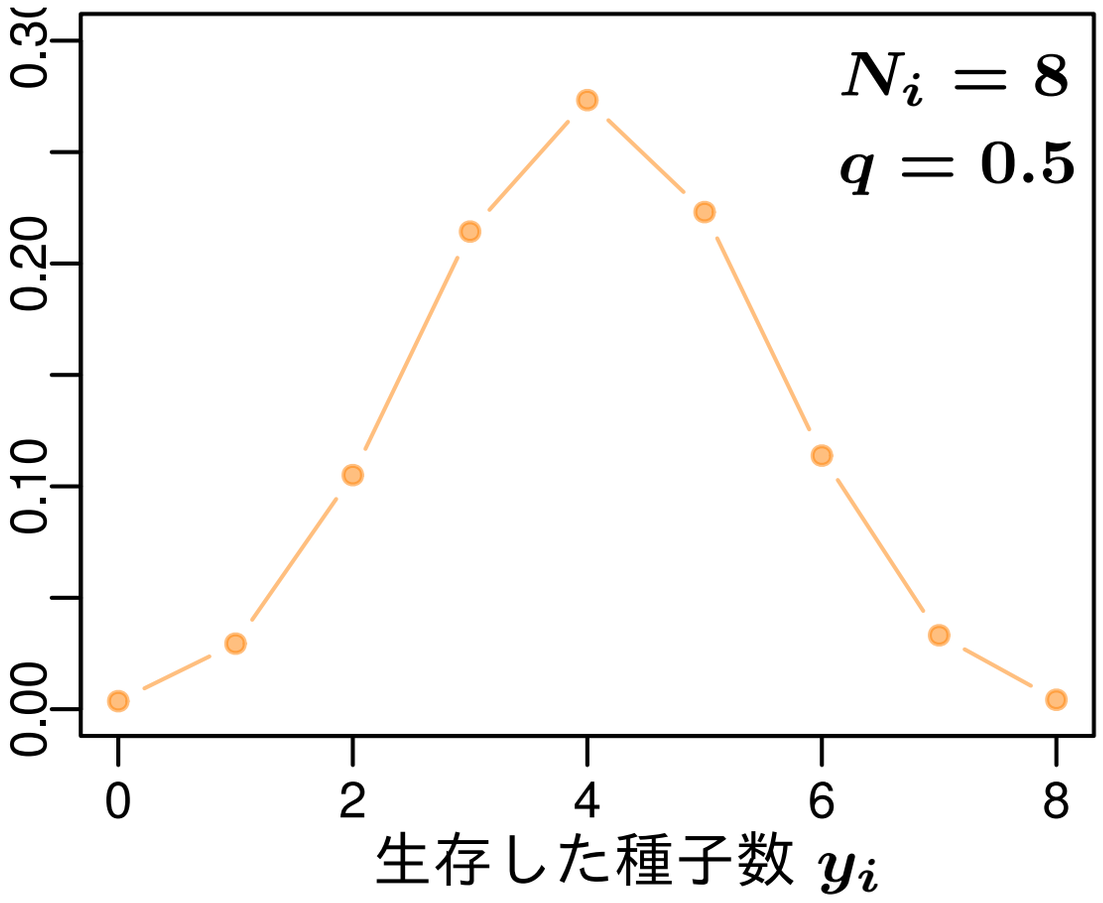
- 生存確率を推定するために 二項分布という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率は二項分布

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

- 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

二項分布で「N_i 個中の y_i 個」型データをあつかう

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



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 $\{y_i\}$ が与えられたもので，パラメータ q は値が自由にとりうると考える
- この 100 個体ぶんの確率はパラメータ q の関数として定義される

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

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

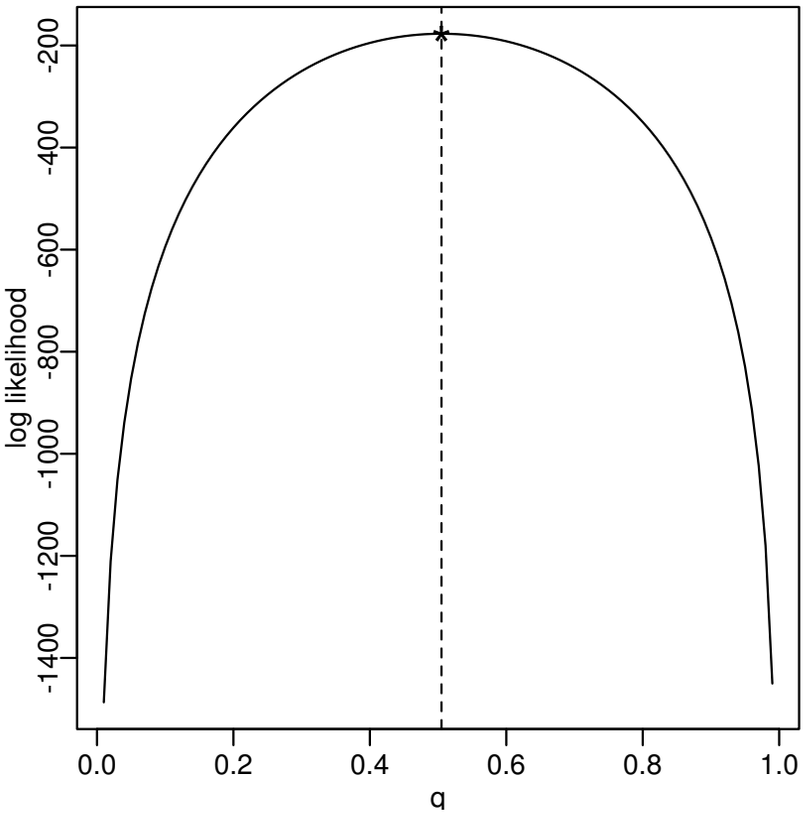
最尤推定とは何か

- 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

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

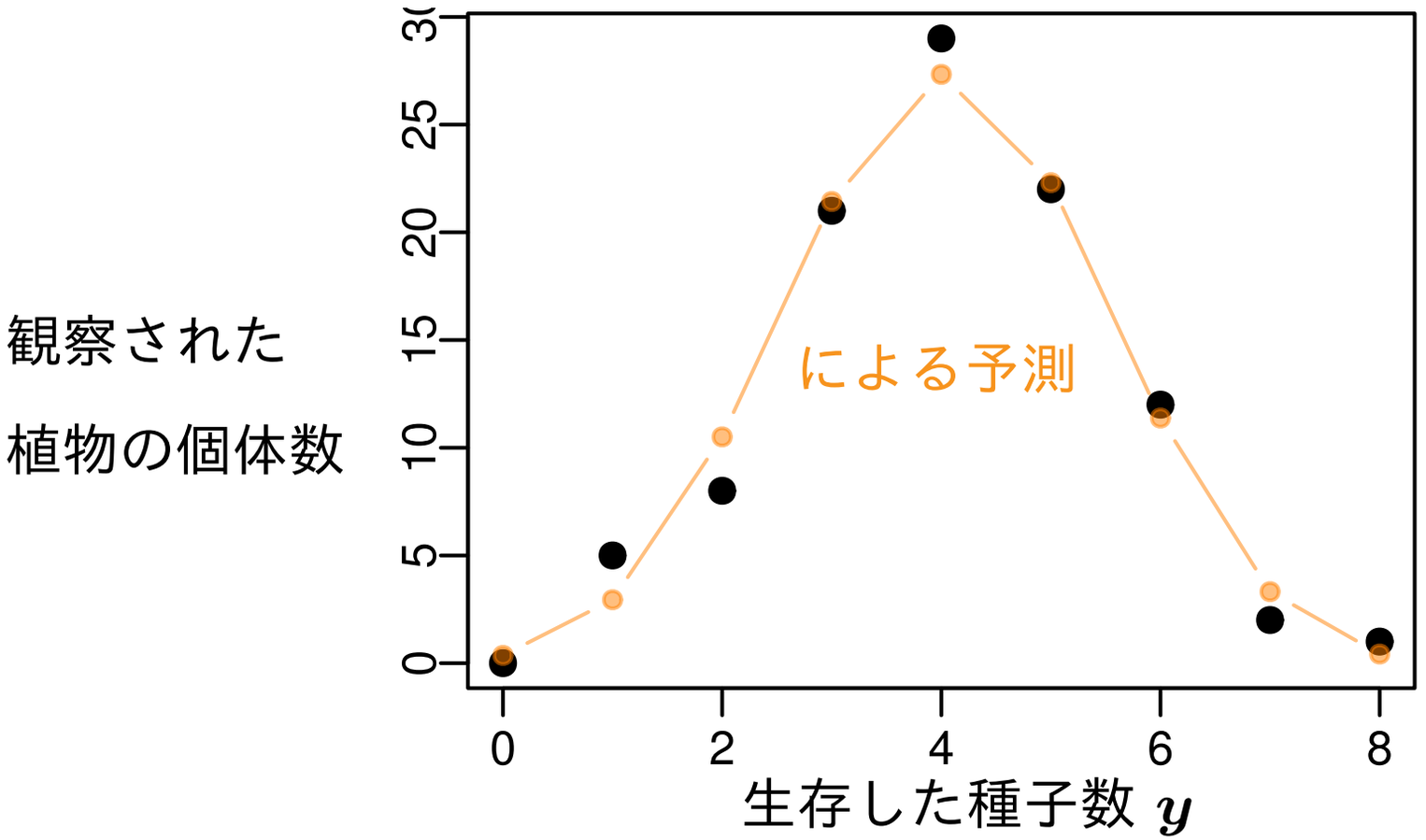
- 生存確率 q が全個体共通の場合の最尤推定量・最尤推定値は

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



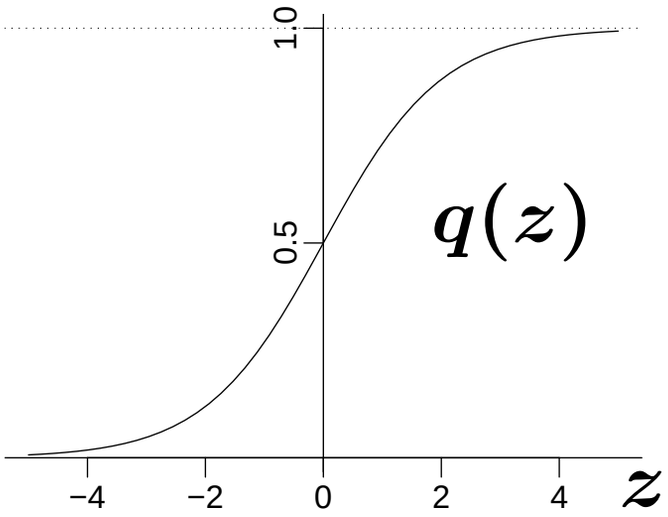
二項分布で説明された 8 種子中 y_i 個の生存

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



ロジスティック関数で表現する生存確率

- そこで生存する確率 $q_i = q(z_i)$ をロジスティック (logistic) 関数 $q(z) = 1/\{1 + \exp(-z)\}$ で表現



- 線形予測子 $z_i = a$ (切片だけ) とする

ちょっと整理: logistic と logit

- logistic 関数

$$q = \frac{1}{1 + \exp(-(a + bx))} = \text{logistic}(a + bx)$$

- logit 変換

$$\text{logit}(q) = \log \frac{q}{1 - q} = a + bx$$

logit は logistic の逆関数，logistic は logit の逆関数

R の glm() によるロジスティック回帰

```
> glm(cbind(y, 8 - y) ~ 1, family = binomial, data = d1)
...(一部略)...
Coefficients:
(Intercept)
       0.02

Degrees of Freedom: 99 Total (i.e. Null);  99 Residual
Null Deviance:     110
Residual Deviance: 110  AIC: 356

> 1 / (1 + exp(-0.02))
[1] 0.505
```

ロジスティック回帰の glm() 指定

- family: binomial, 二項分布
- link 関数: "logit"
- モデル式 (線形予測子 z): たとえば y ~ x と指定
  - 線形予測子 $z = a + bx$

    a, b は推定すべきパラメーター
  - 事象の生起確率 を q とすると $\mathrm{logit}(q) = z$

    つまり $q = \dfrac{1}{\exp(-z)} = \dfrac{1}{1 + \exp(-(a + bx))}$
  - 応答変数 は確率 q でサイズ N の二項分布に従う:

    $y \sim \mathrm{Binom}(q, N)$

より現実的で複雑な統計モデルの

パラメーター推定のため

最尤推定ではなく MCMC で生存確率 q を推定する — パラメーター q の確率分布?

ここでやること: 尤度と MCMC の関係を考える

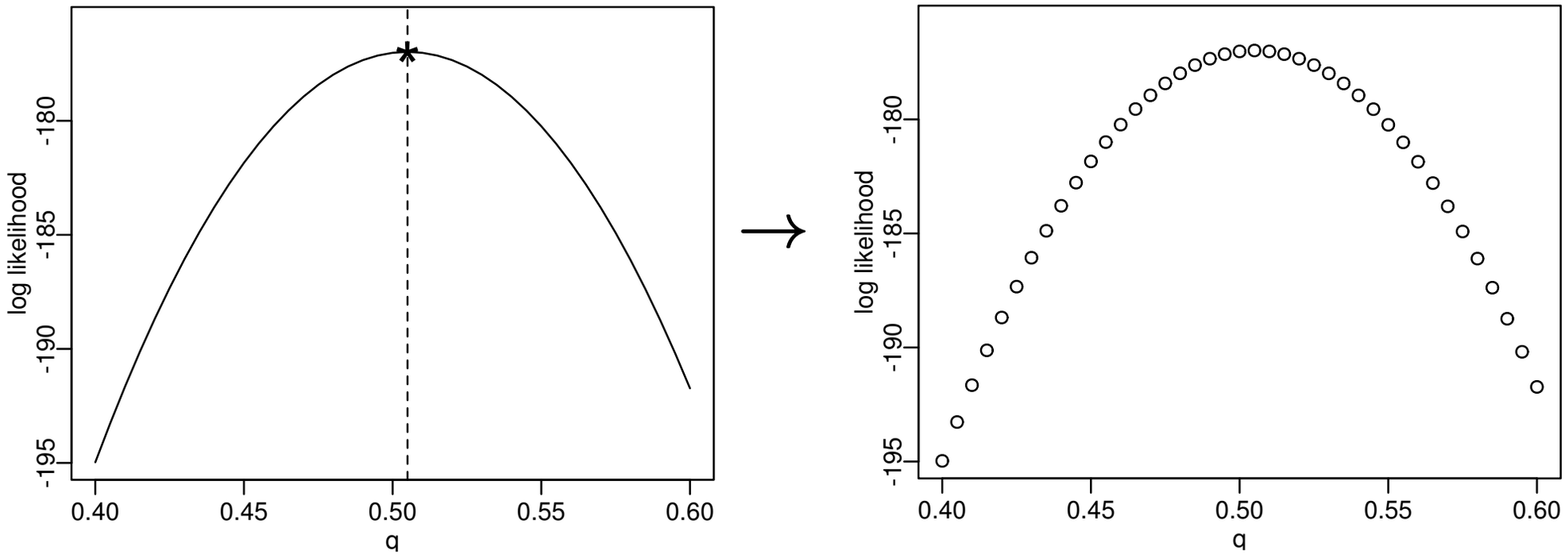
- さきほどの簡単な例題 (生存確率) のデータ解析を
- 最尤推定ではなく
- 試行錯誤な MCMC 法であるメトロポリス 法であつかう
- 得られる結果: パラメーターの分布?

あえて MCMC をもちださなくてもいい問題に関して
メトロポリス法を適用してみて,
その挙動だの得られる結果だのをながめてみる

数値的に試行錯誤するパラメーター推定

連続的な対数尤度関数 $\log L(q)$

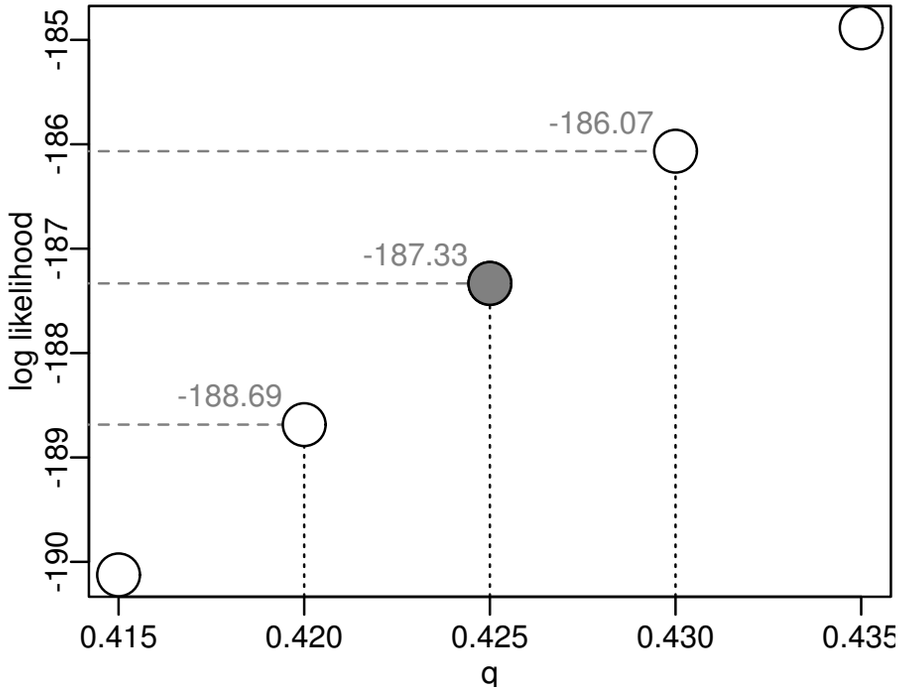
離散化: q がとびとびの値をとる

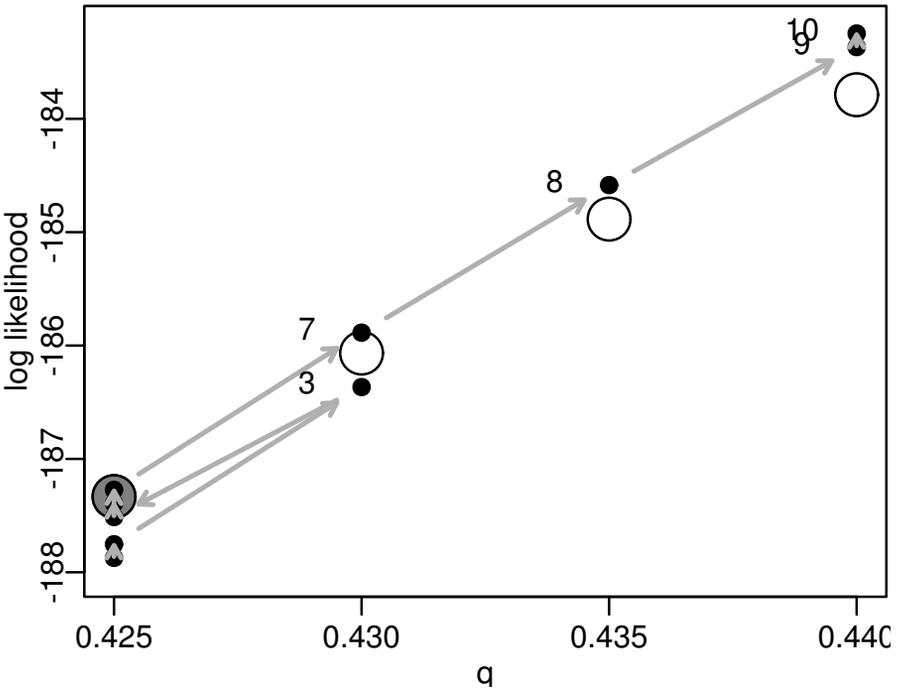


(簡単のため，生存確率 q の軸を離散化する)

メトロポリス法で q を変化させていく

メトロポリス法は MCMC アルゴリズムのひとつ (cf. 伊庭さんの解説)





(q の初期値を 0.425，ランダムウォークで移動先を選ぶ)

(補足) この例題のメトロポリス法

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 0.425)

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 $q_{新}$ としましょう)

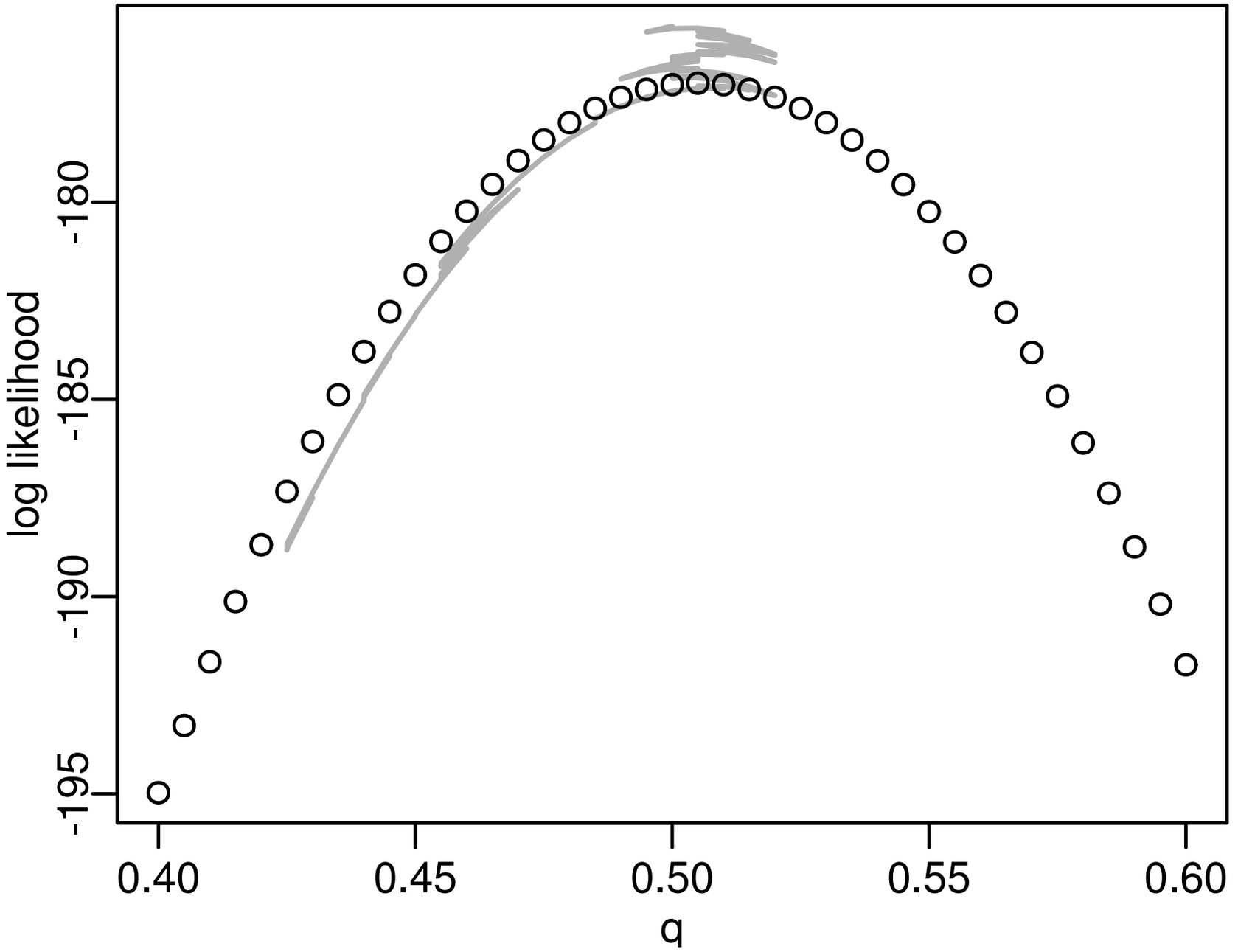
3. $q_{新}$ における尤度 $L(q_{新})$ ともとの尤度 $L(q)$ を比較

   - $L(q_{新}) \geq L(q)$ (あてはまり改善): $q \leftarrow q_{新}$
   - $L(q_{新}) < L(q)$ (あてはまり改悪):
     - 確率 $r = L(q_{新})/L(q)$ で $q \leftarrow q_{新}$
     - 確率 $1 - r$ で q を変更しない

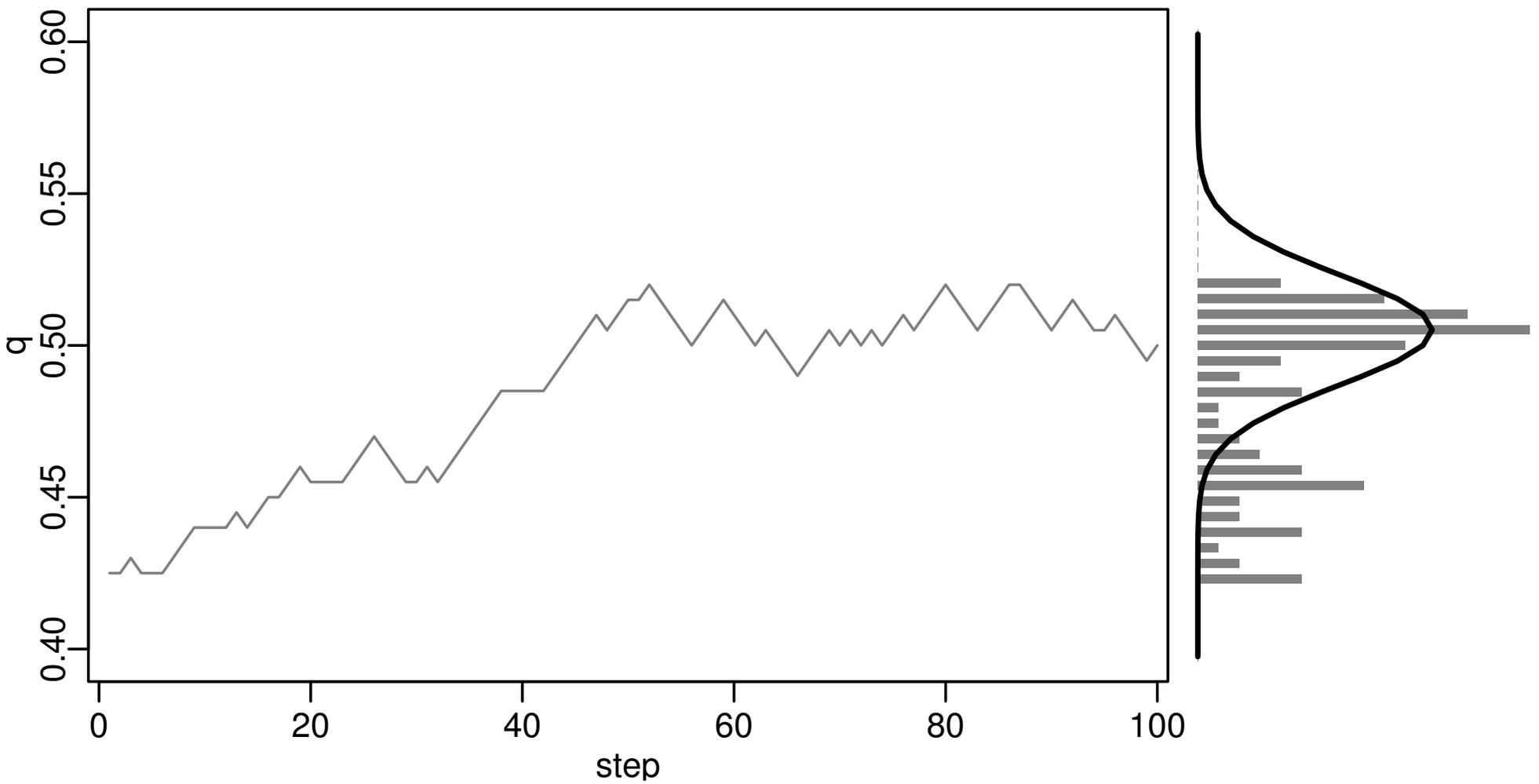
4. 手順 2. にもどる

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

対数尤度関数上での生存確率 q の変化

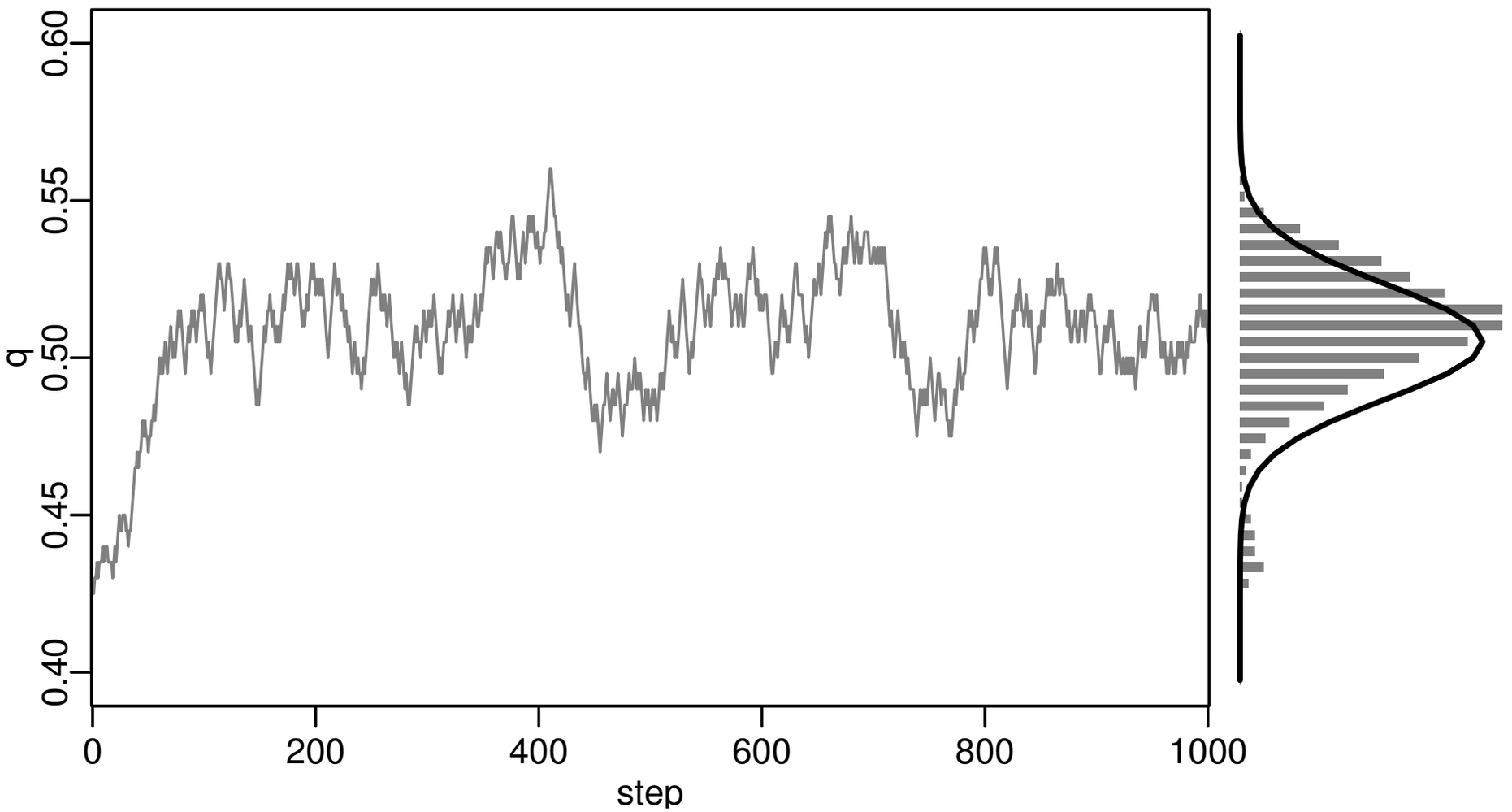


MCMC ステップにそった q の変化



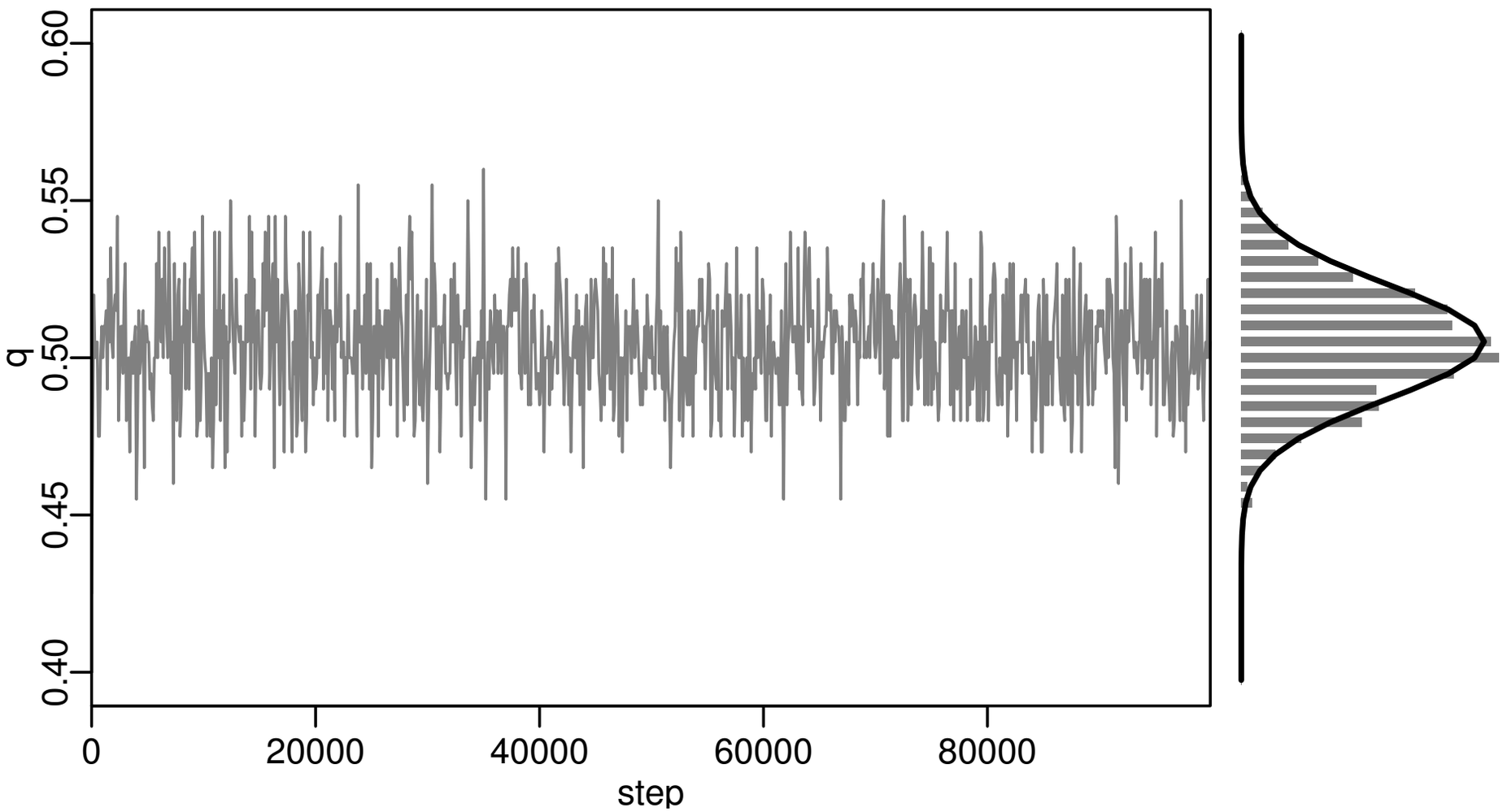
右側は q のヒストグラム

もっと長くサンプリングしてみる



まだまだ……?

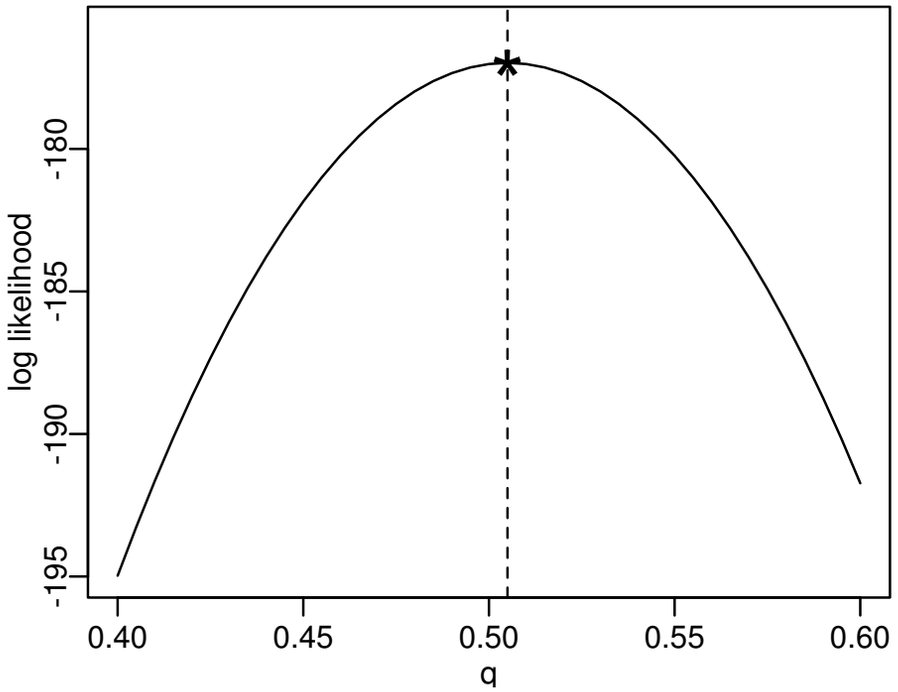
もっともっと長くサンプリングしてみる



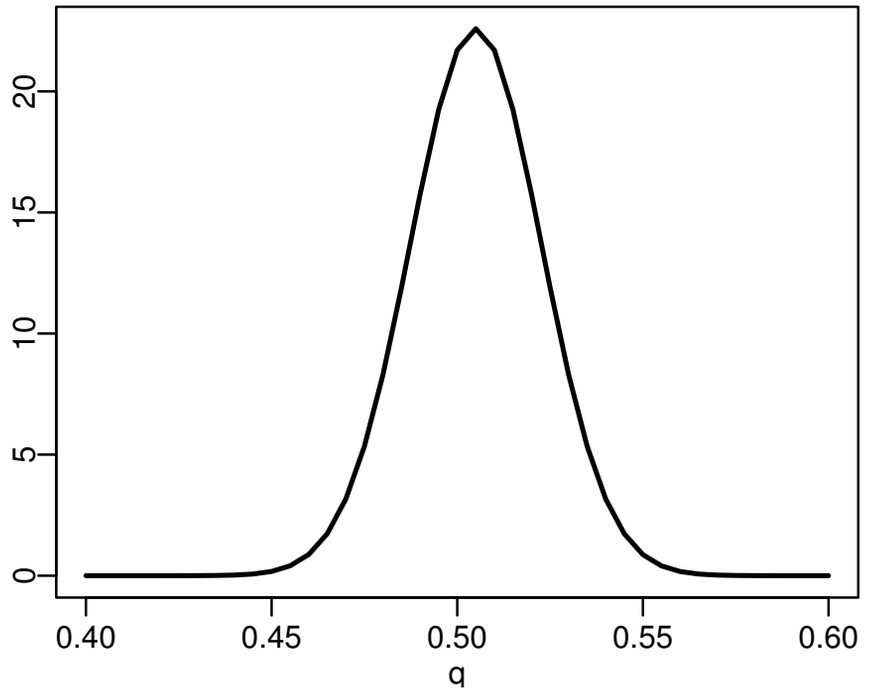
ターゲットとなる「パラメーターの分布」に近づいてきた

MCMC は何をサンプリングしている?

既出の対数尤度 $\log L(q)$



尤度 $L(q)$ に比例する確率分布

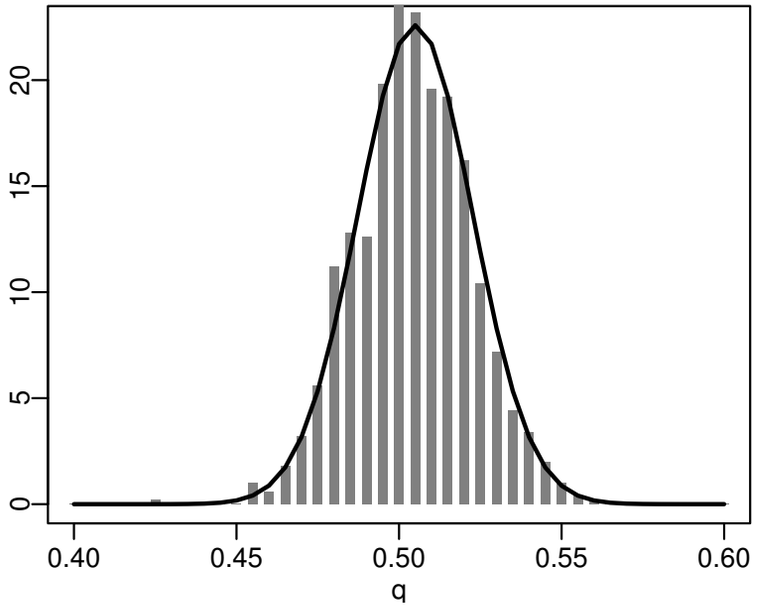


尤度に比例する確率分布からのランダムサンプル

(「パラメーターの分布」と仮称)

「マルコフ連鎖の収束定理」のおかげ (cf. 伊庭さんの説明)

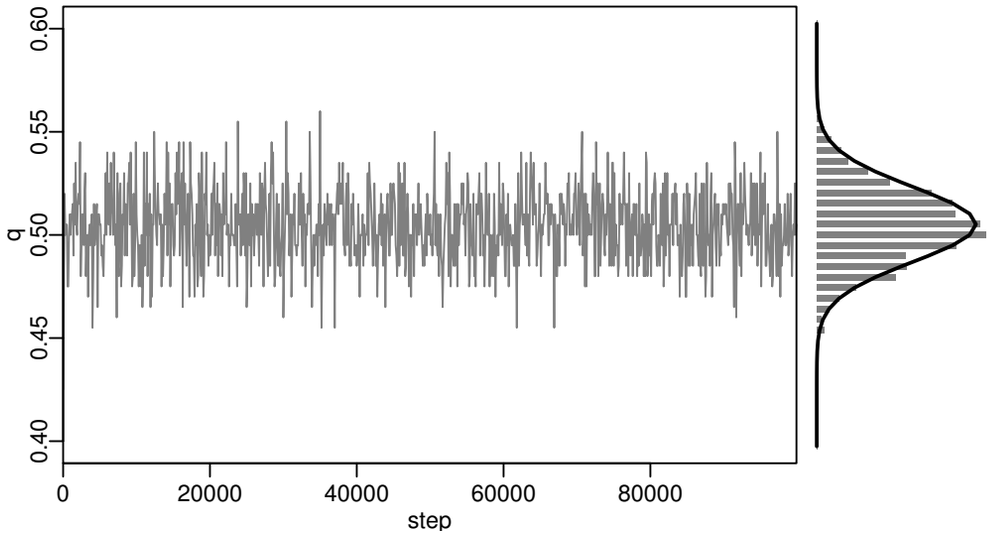
MCMC の結果として得られた「q の分布」



- データからえられる推定結果としては有用: 分布の平均や区間推定など
- 「パラメーターの分布」 …… ベイズ統計でいうところの事後分布

いったん整理: 尤度と MCMC の関係

- 統計モデルを作ると，あるデータのもとでの尤度が定義される
- この尤度に対して MCMC すると「尤度に比例するパラメーターの分布」からのランダムサンプルがえられる
- ベイズとの関連: これは事後分布からのサンプリングである



いったん整理: いろいろな MCMC の方法

- 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・サンプラー: 条件つき確率分布を使った MCMC
  - 普通は複数の変数 (パラメーター・状態) のサンプリングのためにもちいる (あとでこの例題にそった簡単な説明)

- メトロポリス法で説明したけれどギブス・サンプラーでも同じことが言える
- ここからあとで登場する MCMC はギブス・サンプラーと考えてください

ベイズモデル: 尤度・事後分布・事前分布……

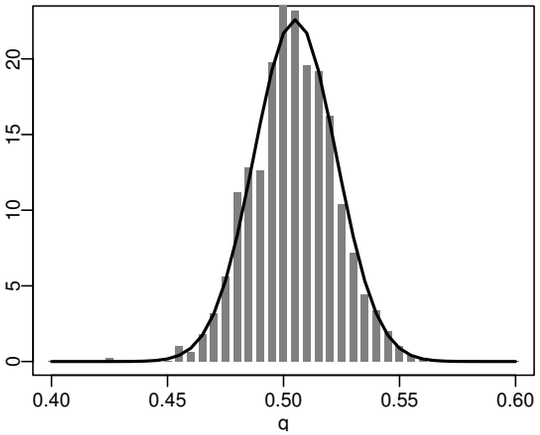
- ベイズの公式 $\quad p(q \mid \mathbf{Y}) = \dfrac{p(\mathbf{Y} \mid q) \times p(q)}{p(\mathbf{Y})}$
- $p(q \mid \mathbf{Y})$ は何かデータ $(\mathbf{Y})$ のもとで何かパラメーター (q) が得られる確率 (事後分布)
- $p(q)$ はあるパラメーター q が得られる確率 (事前分布)
- $p(\mathbf{Y} \mid q)$ パラメーターを決めたときにデータが得られる確率 (尤度に比例)
- $p(\mathbf{Y})$ はデータ $\mathbf{Y}$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

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

現在の例題で仮定している事前分布

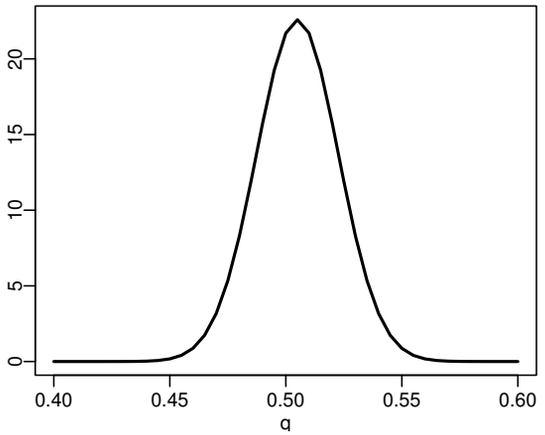
q の事前分布は一様分布，と考えるとつじつまがあう?

q の事後分布 (posterior)



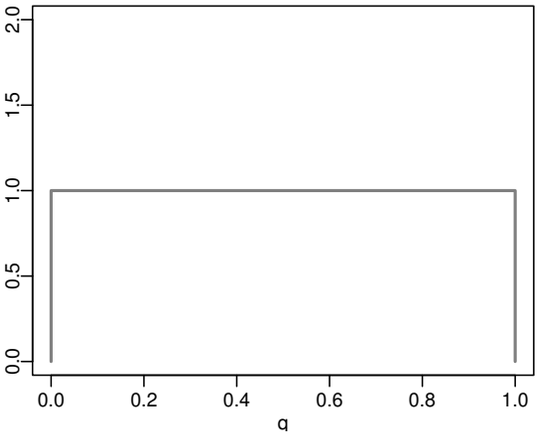
$\propto$

q の尤度 (likelihood)



$\times$

q の事前分布 (prior)



このように「q はどんな値でもいいんですよ」という気分を表現するための事前分布が無情報事前分布 (non-informative prior)

ちょっと難しい例題:

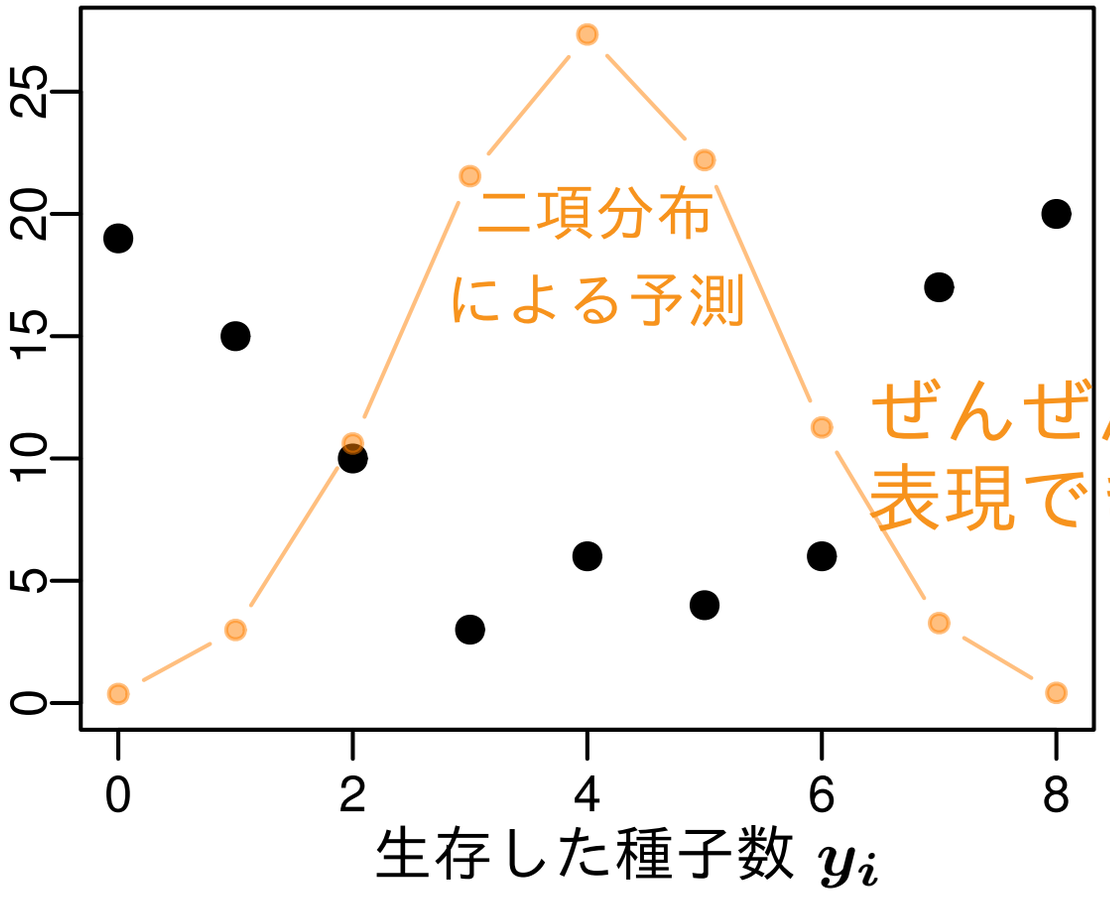
個体差が大きくて **GLM** がうまくいかない

階層ベイズモデルが必要になる状況

また別の観測データ: 二項分布だめだめ?!

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……

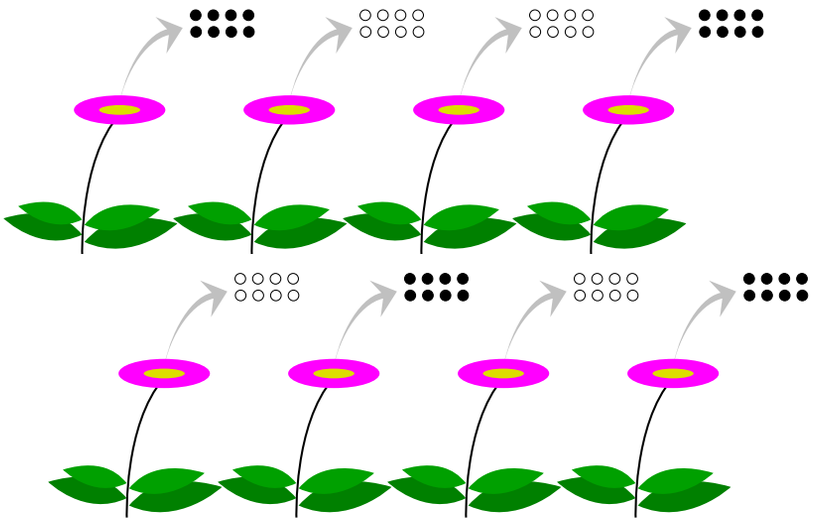
観察された植物の個体数

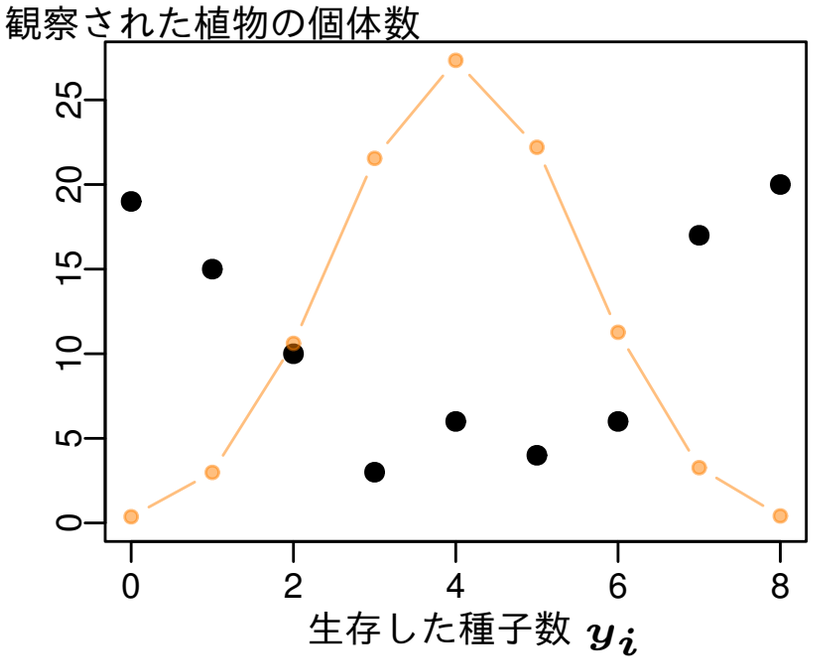


さっきの例題と同じようなデータなのに?

「個体差」 → 過分散 (overdispersion)

極端な過分散の例





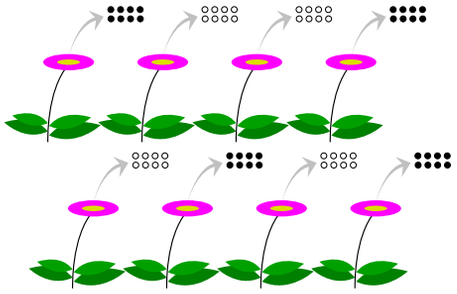
- 種子全体の平均生存確率は 0.5 ぐらいかもしれないが……
- 植物個体ごとに種子の生存確率が異なる: 「個体差」
- 「個体差」があると overdispersion が生じる
- 「個体差」の原因: ?

あのー …… 「個体差」とは?

- 生物学的には明確な定義はない
- しかしデータ解析においては人間が主観的に「これは個体差由来の効果であり，観察されたパターンに影響している」と定義，そして以下の二種類を区別する:
  1. **fixed effects** 的な効果
  2. **random effects** 的な効果
     - これって何なの?

「個体差」の fixed だの random だの …… って何?

- 「個体ごとに異なる何かに由来する効果」を fixed/random effects にわけて統計モデリングする:
  1. fixed effects 的な効果: 観測者がわざわざ設定・測定した要因 (実験処理，植物のサイズなど)，logit 変換された世界において生存確率の「効果の大きさ」を変える
     - この例題では fixed effects 的な要因なし
  2. random effects 的な効果: fixed effects 的ではない要因 (観測対象個体に関連する，人間が設定・測定していないすべて)
     - logit 変換された世界において生存確率の「効果の大きさ」を変えずにばらつきだけを変えると考える



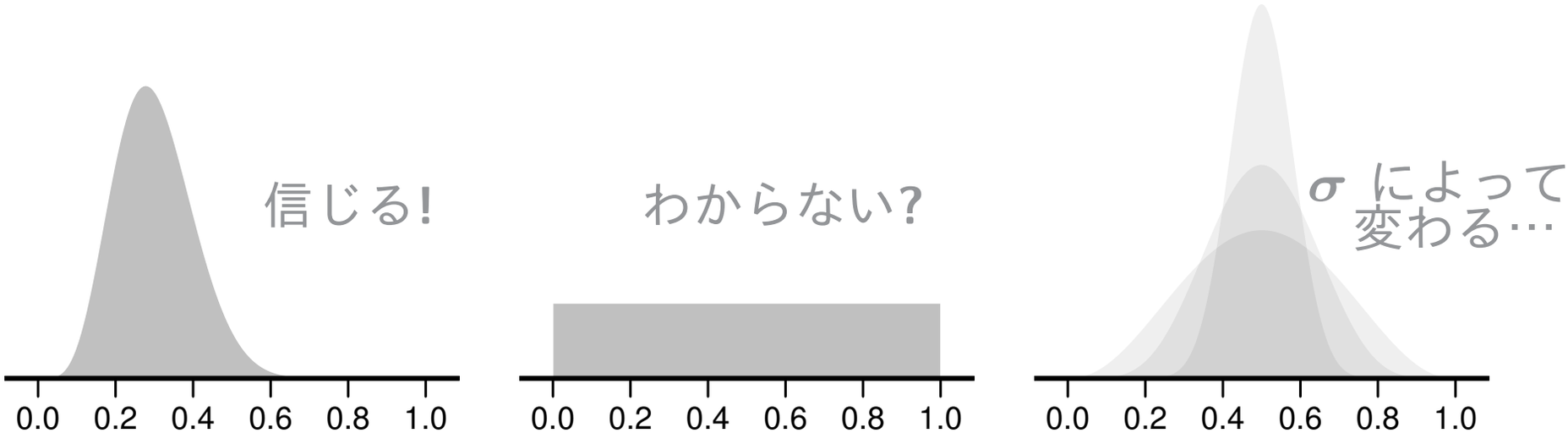
今回の例題では random effects な

「個体差」だけをあつかう (説明変数なし)

やっぱりよくわからない?

パラメーターごとに設定してやる事前分布の種類(あとで説明するベイズ統計モデリング) にもとづいて考えたほうが，わかりやすいかも?

(A) 主観的な事前分布　**(B)** 無情報事前分布　**(C)** 階層的な事前分布

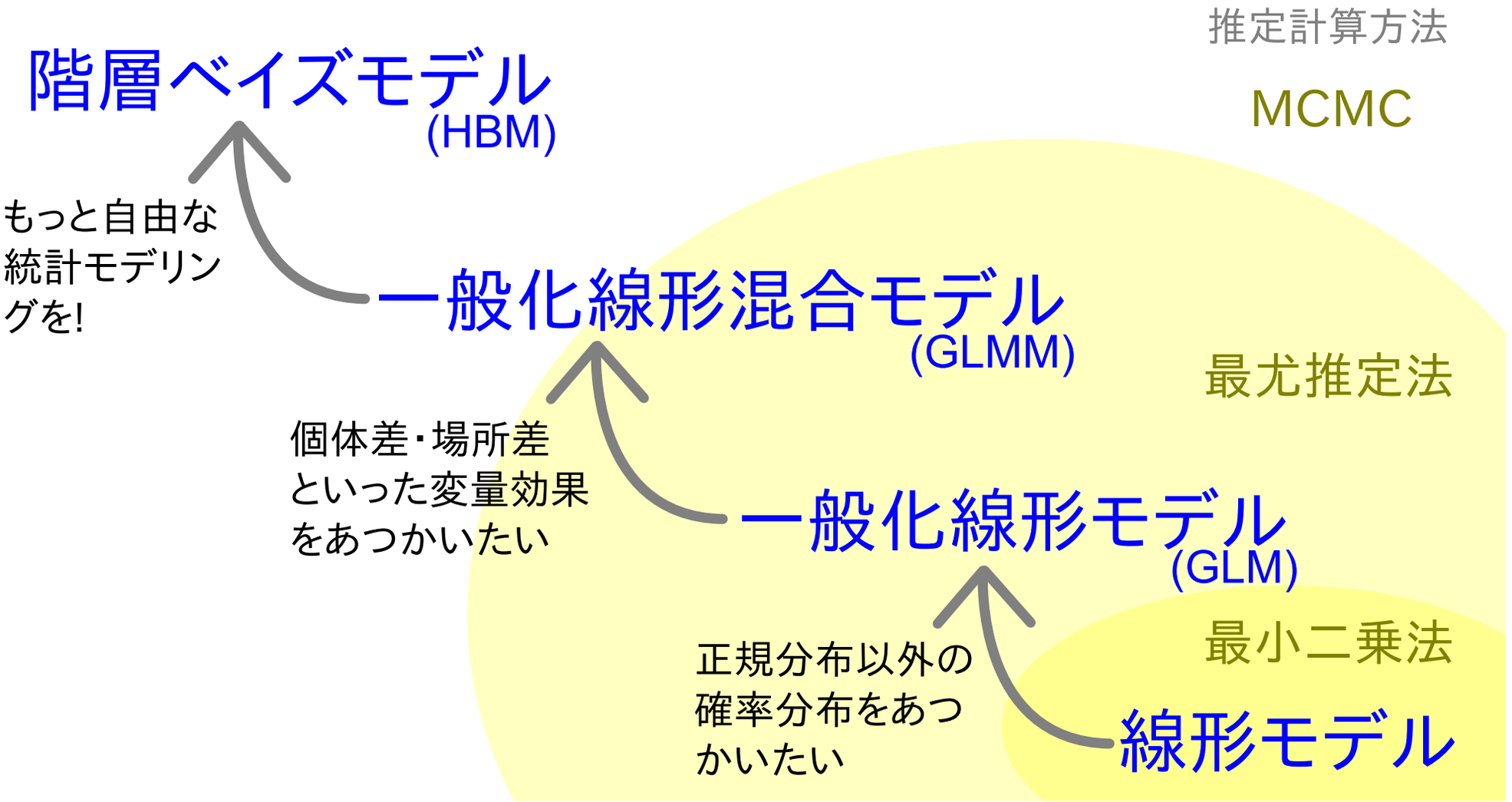


このパラメーターの事前分布は

どう設定するのが妥当だろうか，と検討する

ちょっとまた線形モデルのたぐいについて整理してみましょう……

線形モデルの発展



モデリングやりなおし: まず二項分布の再検討

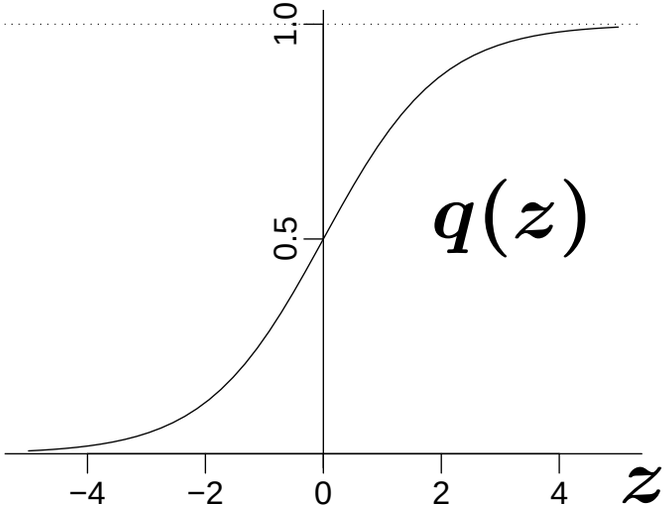
- 生存確率を推定するために 二項分布という確率分布を使う
- 個体 i の N_i 種子中 y_i 個が生存する確率は二項分布

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

- ここで仮定していること
  - 個体差がある
  - 個体ごとに異なる生存確率 q_i

ロジスティック関数で表現する生存確率

- そこで生存する確率 $q_i = q(z_i)$ をロジスティック (logistic) 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)

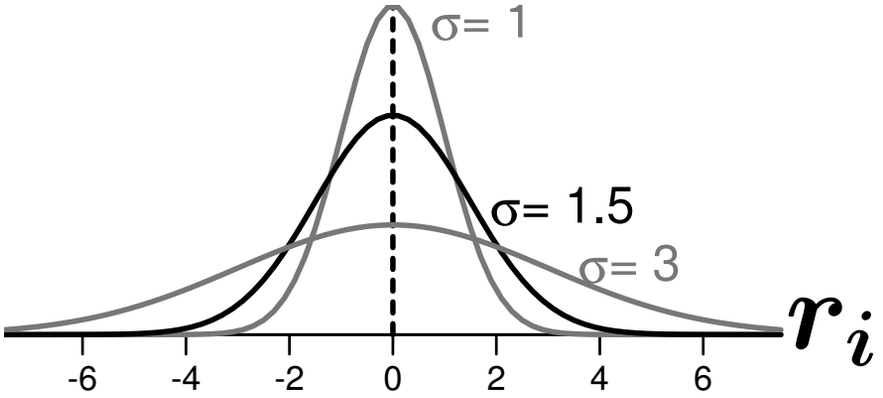
個々の個体差 r_i を最尤推定するのはまずい

- 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)
- こう仮定すると問題がうまくあつかえないだろうか?
  - 個体間の生存確率はばらつくけど，そんなにすごく異ならない?
  - 観測データを使って,「個体差」にみられるパターンを抽出したい (統計モデル化)

階層ベイズモデル化: r_i の事前分布の設計

平均ゼロで標準偏差 σ の正規分布

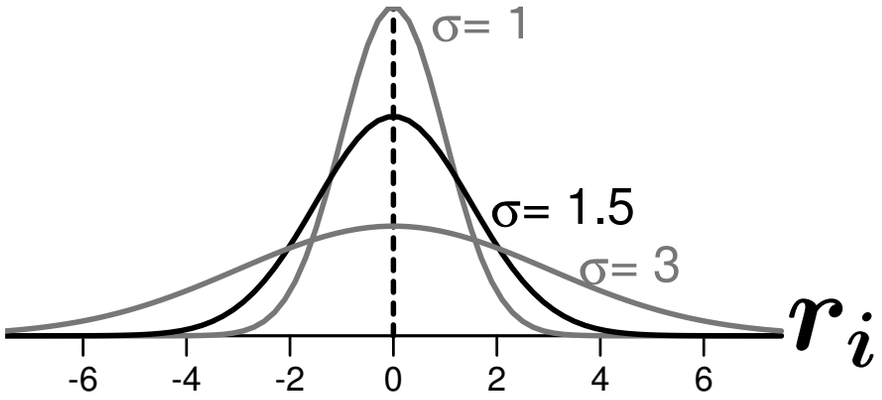
$$p(r_i \mid \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-r_i^2}{2\sigma^2}\right)$$



個体差 $\{r_1, r_2, \cdots, r_{100}\}$ がこの確率分布に従うとする

r_i の事前分布は無情報事前分布ではない

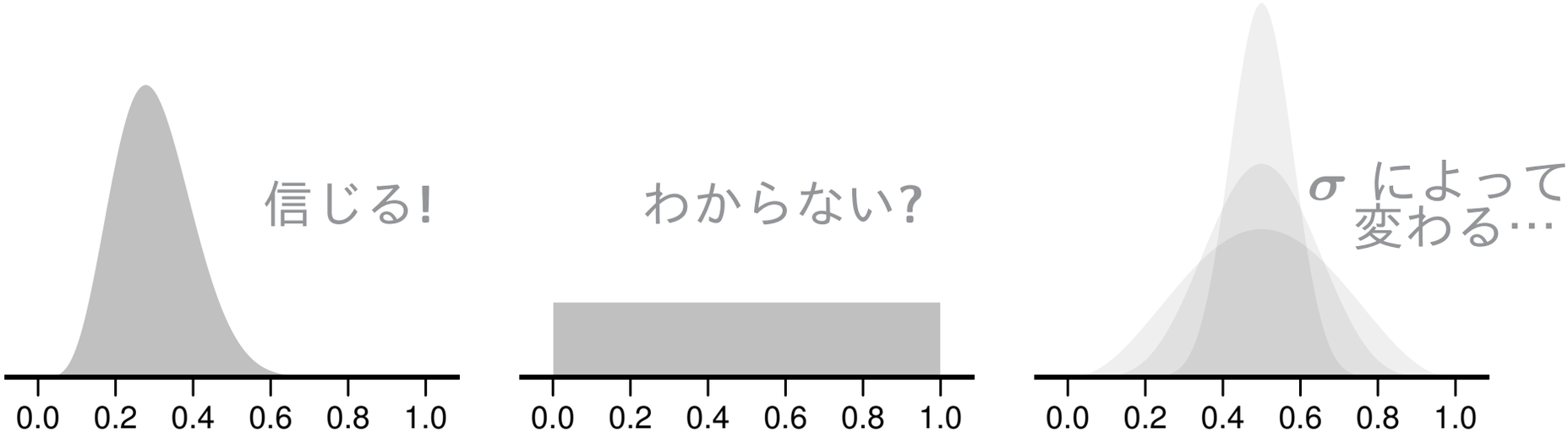
データにあわせて σ が変化する階層的な事前分布



- σ がとても小さければ個体差 r_i はどれもゼロちかくになる $\rightarrow$ 「どの個体もおたがい似ている」
- σ がとても大きければ，r_i は各個体の生存数 y_i にあわせるような値をとる

個体差 r_i の事前分布は階層的な事前分布

(A) 主観的な事前分布　**(B)** 無情報事前分布　**(C)** 階層的な事前分布



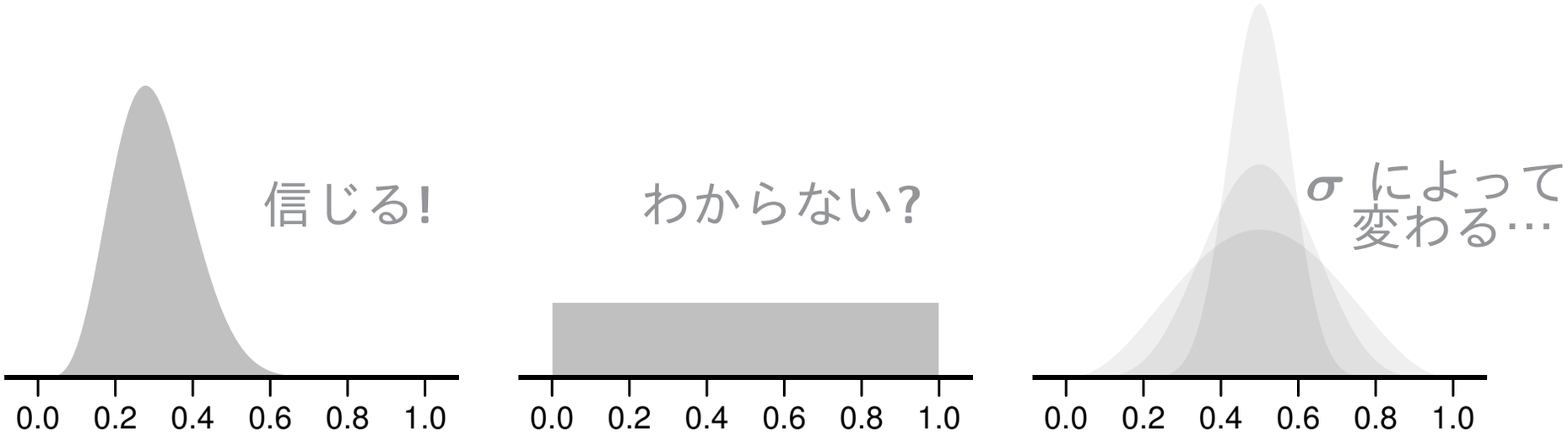
(A) 主観的な事前分布: 「自分の信じるところによれば，r_i たちはこんな分布になる」を表現している.

(B) 無情報事前分布: 「r_i たちがどんな値になるのかまったくわかりません」を表現しようとしている (しかし -5 から 5 ぐらい，という主観も表現している).

(C) 階層的な事前分布: r_i の事前分布のパラメーター σ がいろいろな値をとる，そして σ についての超事前分布を設定する.

パラメーターごとに事前分布を選ぶ (1)

(A) 主観的な事前分布　**(B)** 無情報事前分布　**(C)** 階層的な事前分布

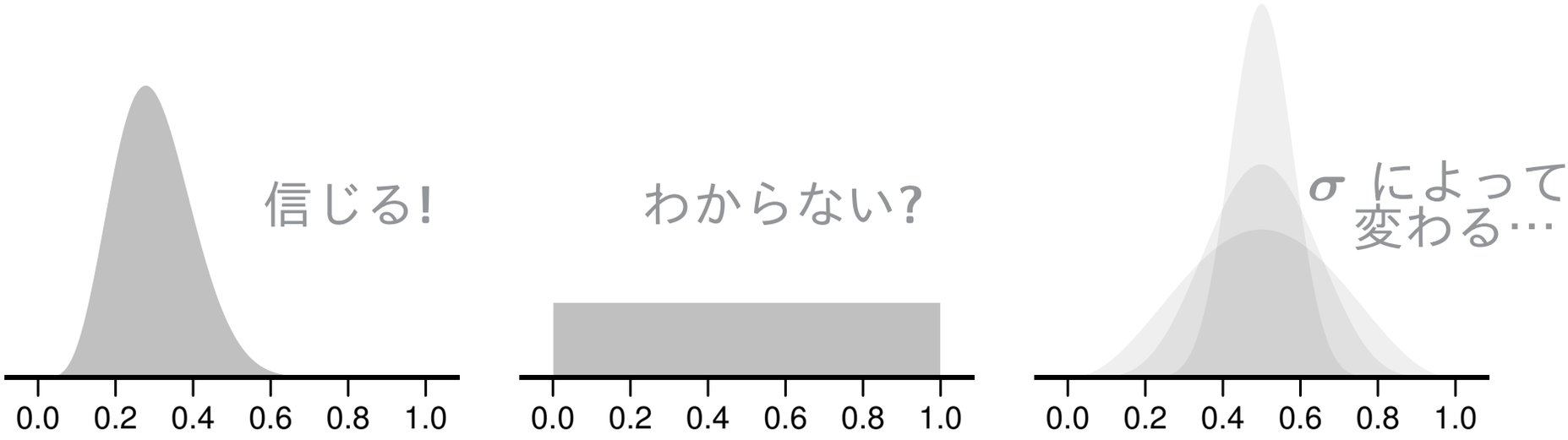


(何か植物たちの統計モデリングをやっているとすると)

- **fixed effects** 的な効果:
  - 無情報事前分布を設定する
  - (実験処理の効果，植物個体の大きさなど属性の効果などを) 全個体に共通する効果をひとつのパラメーターで表現しているから

パラメーターごとに事前分布を選ぶ (2)

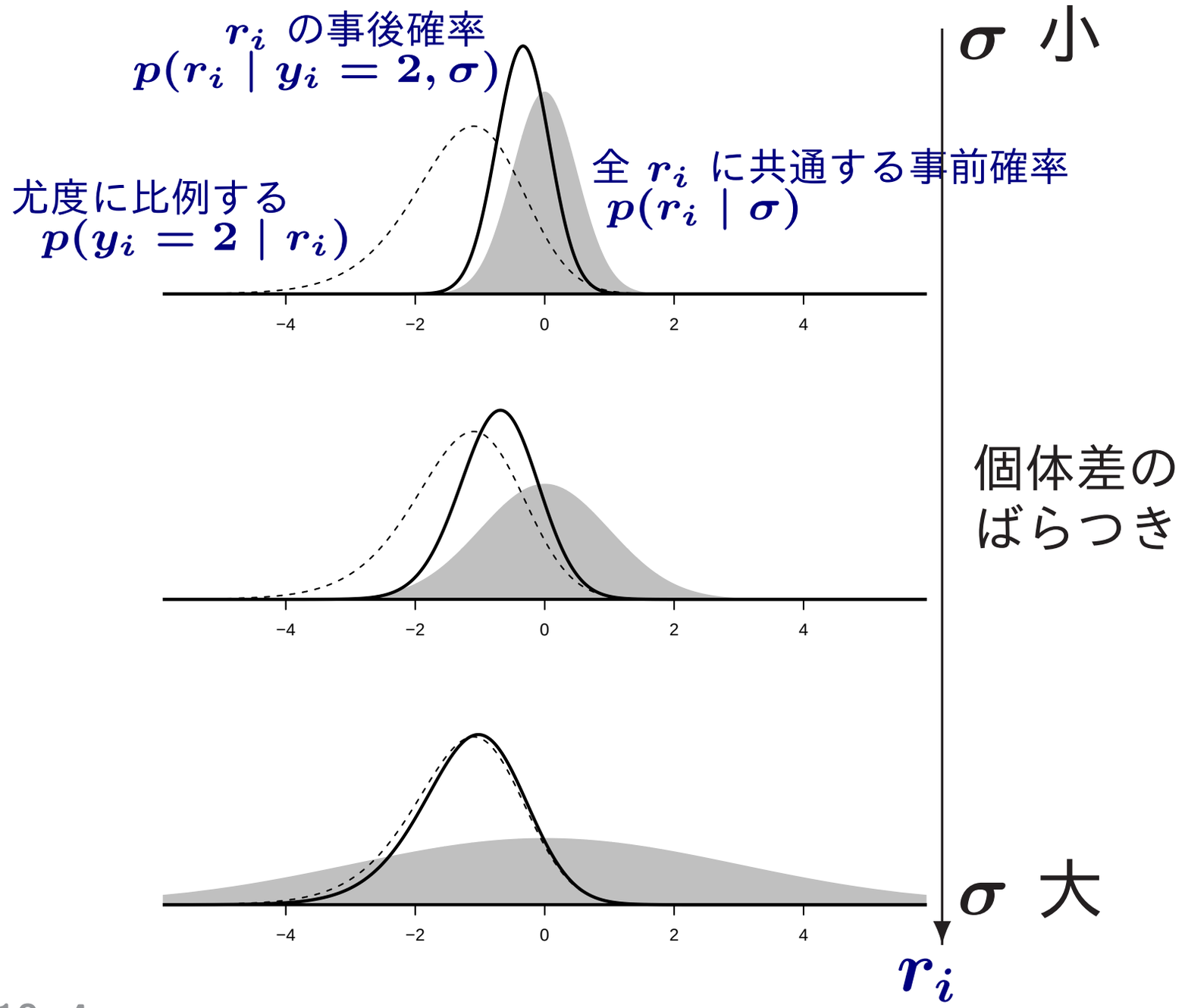
(A) 主観的な事前分布　**(B)** 無情報事前分布　**(C)** 階層的な事前分布



(何か植物たちの統計モデリングをやっているとすると)

- **random effects** 的な効果:
  - 階層的な事前分布を設定する
  - 個体ごとに異なるばらつき r_i が何かあって，しかし集団全体で $\{r_i\}$ が何かの (意味ありげな) 確率分布にしたがっていると考えるのが妥当そうだから

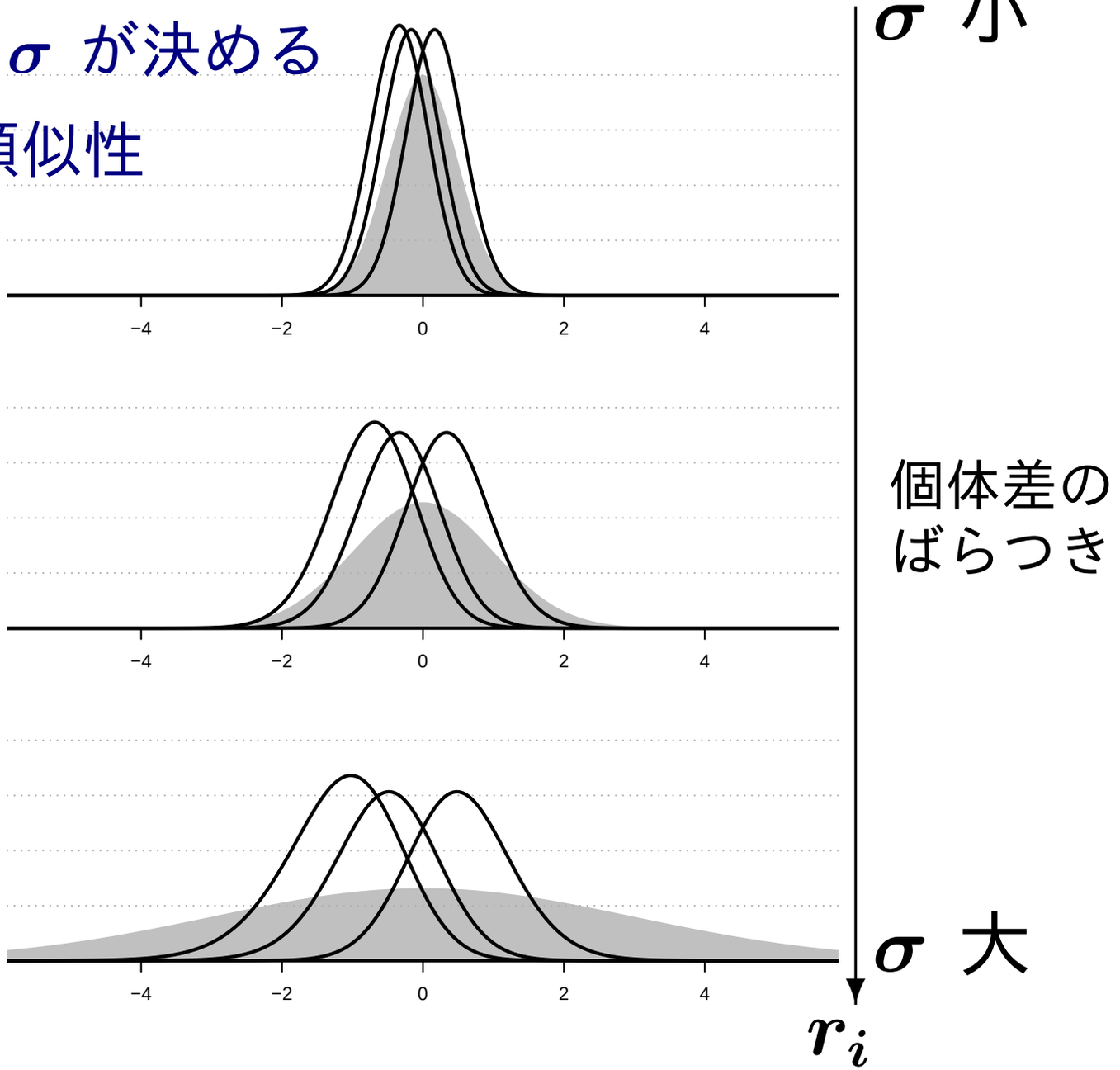
階層的な事前分布と $y_i = 2$ の個体の r_i



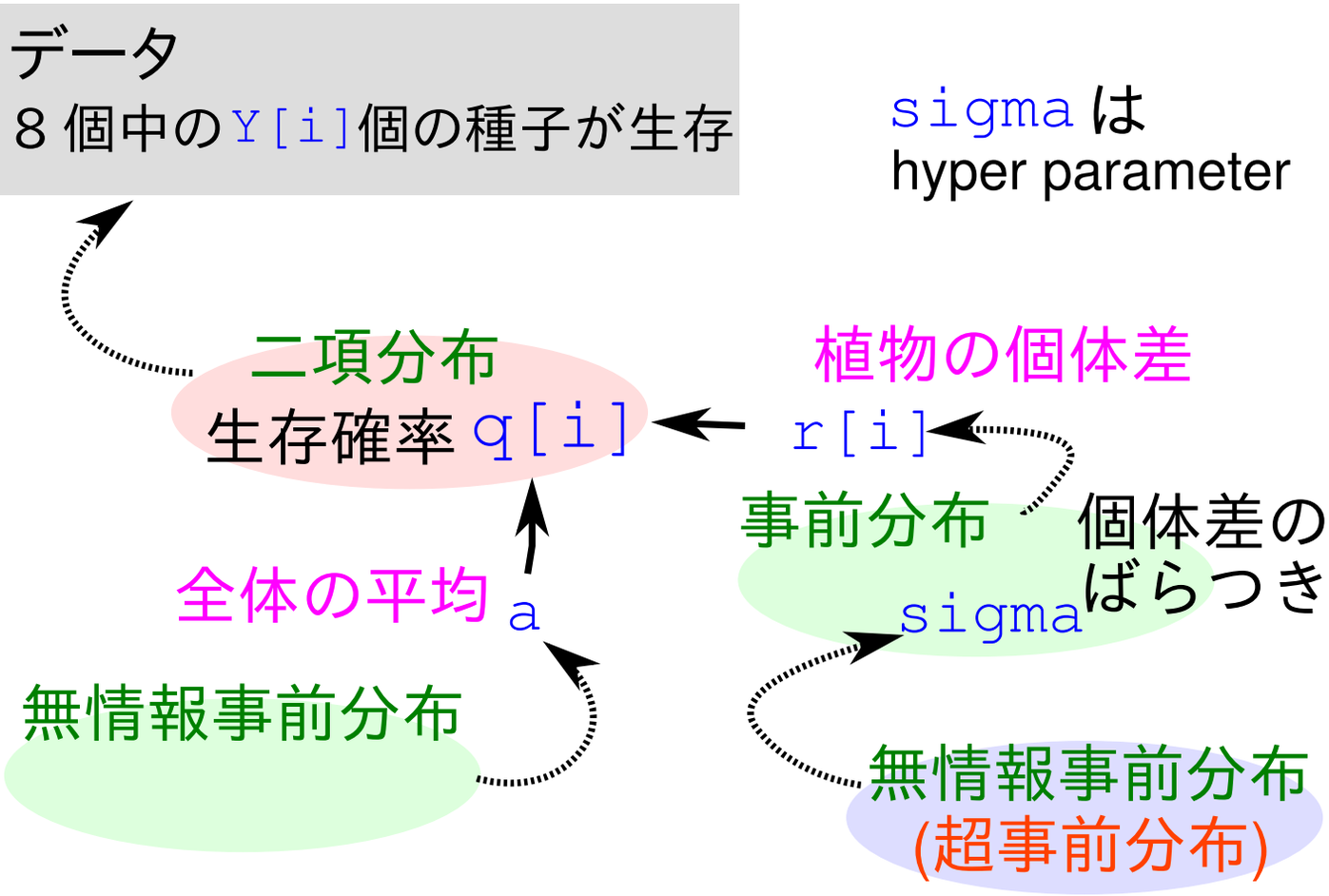
階層的な事前分布と $y_i \in \{2, 3, 5\}$ の個体の r_i

パラメーター σ が決める

個体間の類似性



なぜ「階層」ベイズモデルと呼ばれるのか?

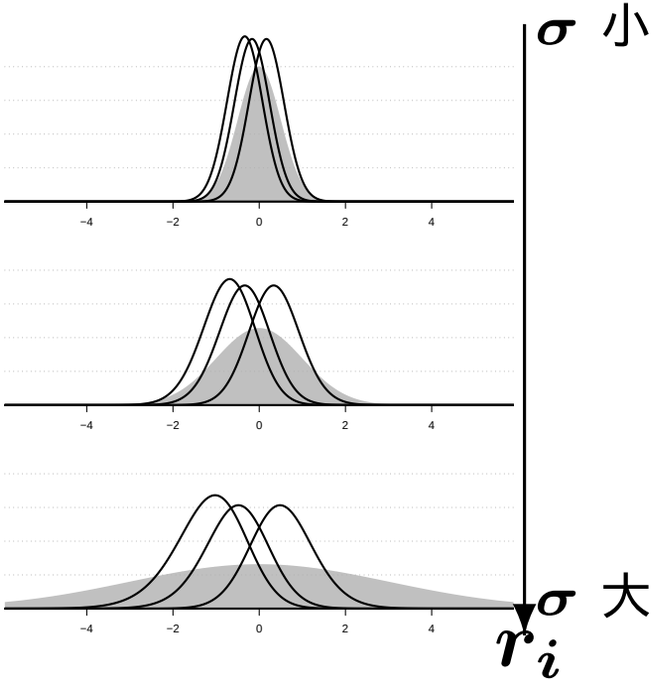


超事前分布 → 事前分布という階層があるから

階層ベイズモデルではないベイズモデルって何でしょう?

個体差 r_i の事前分布の設定を例に検討してみる

- 事前分布を主観的に決める
  「自分は $\sigma = 0.1$ と信じるので，それを使う」
- 以前のデータを使う?
  「これまでの経験から $\sigma = 0.1$」
- 無情報事前分布ばかりにする
  「よくわからないので σ をすごく大きくする」



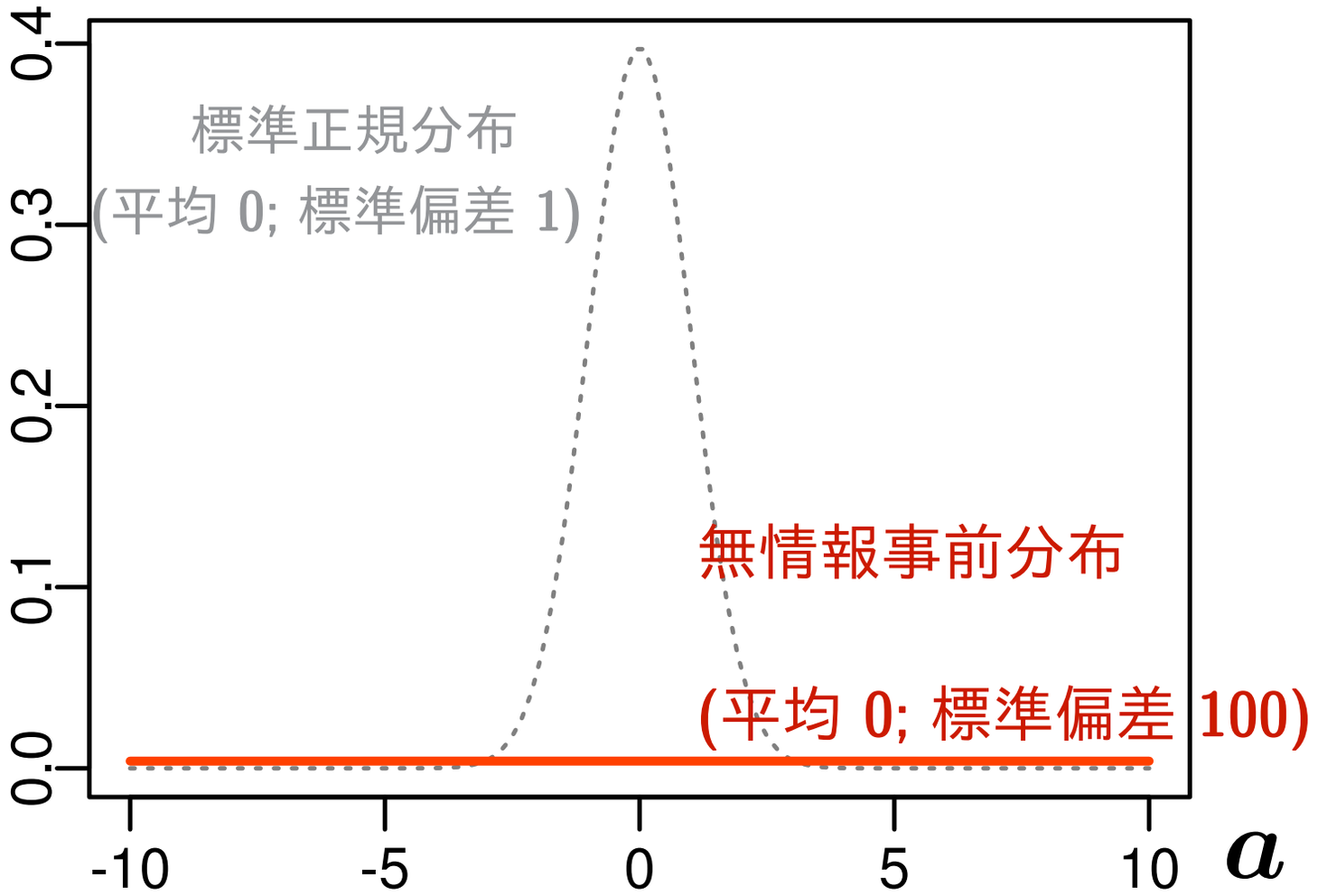
(これらに対して)

観測データにもとづいて σ を決めようとするのが階層ベイズモデル

個体差 $\{r_i\}$ のばらつき σ の無情報事前分布

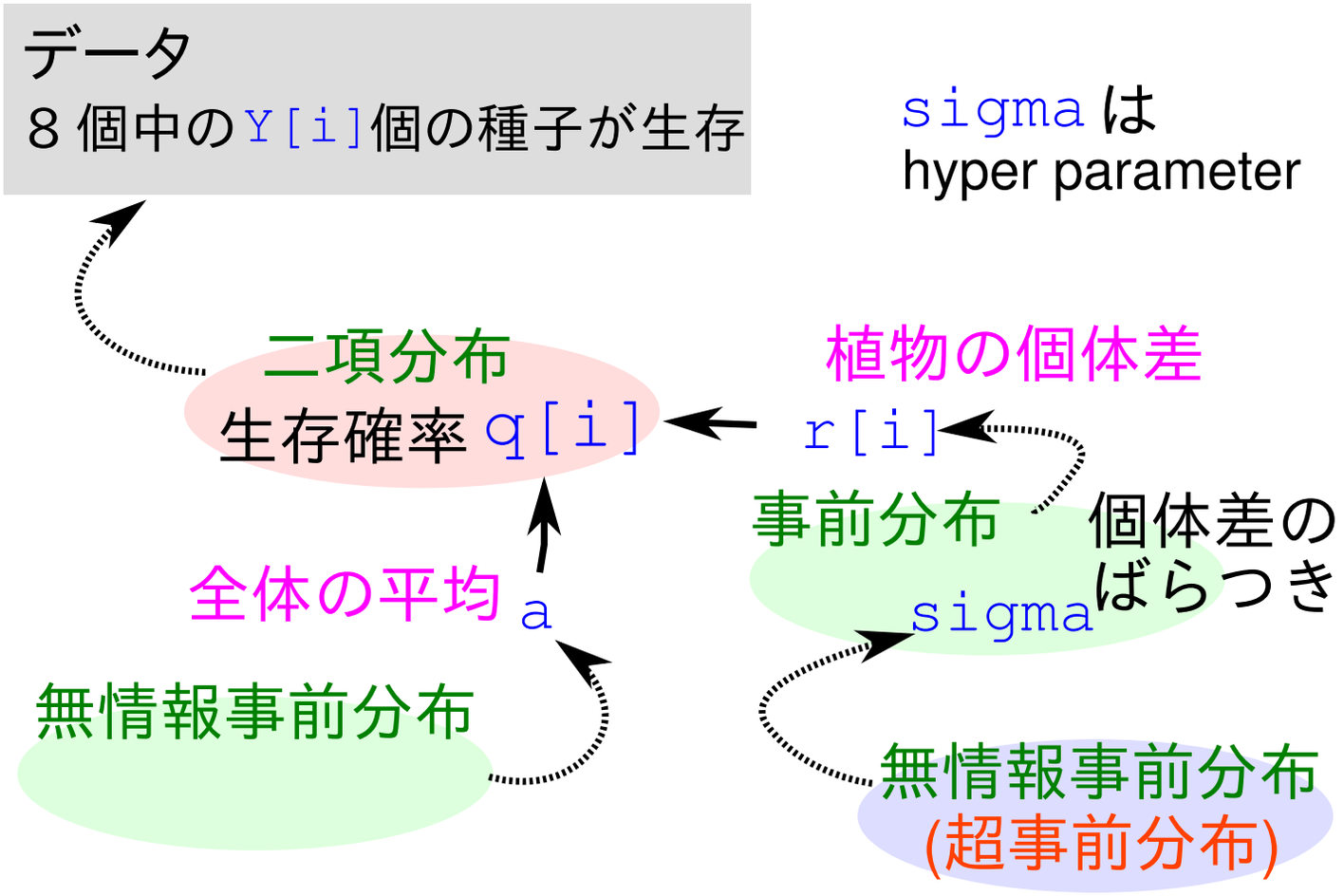
- σ はどのような値をとってもかまわない
- そこで σ の事前分布は 無情報事前分布(non-informative prior) とする
- たとえば一様分布
  - とりあえず，ここでは $0 < \sigma < 10^4$ の一様分布としてみる

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

全パラメーターを一斉に推定する



矢印は手順ではなく，依存関係をあらわしている

どうやってパラメーター推定をするのか?

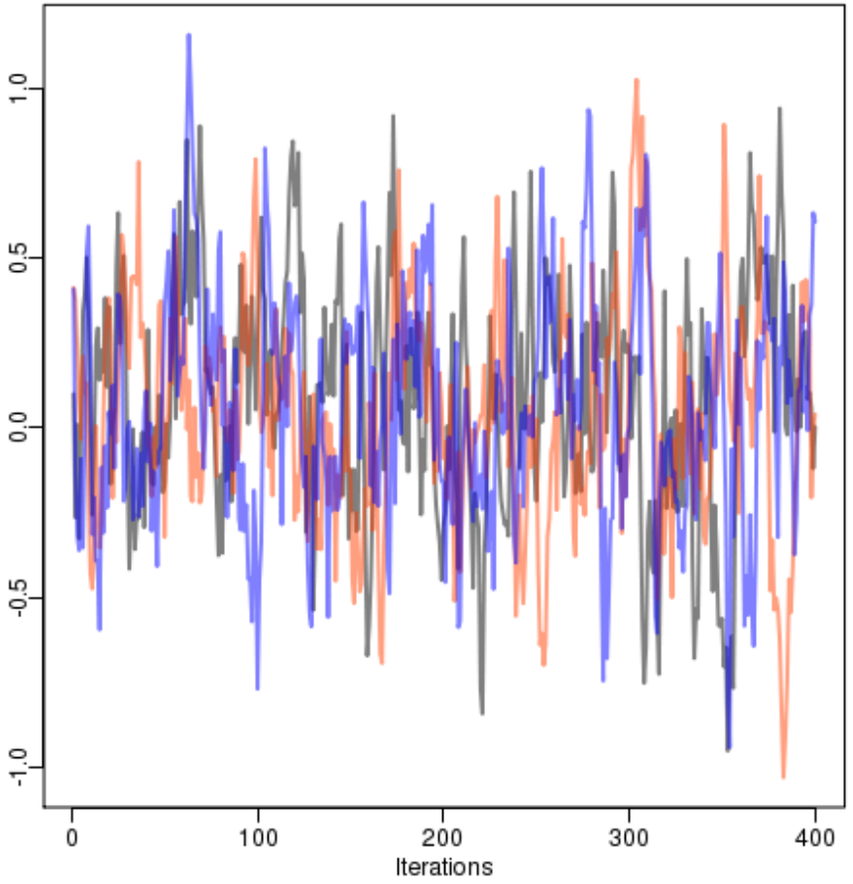
今回は省略 (次回に説明します)

WinBUGS + R でパラメーターの

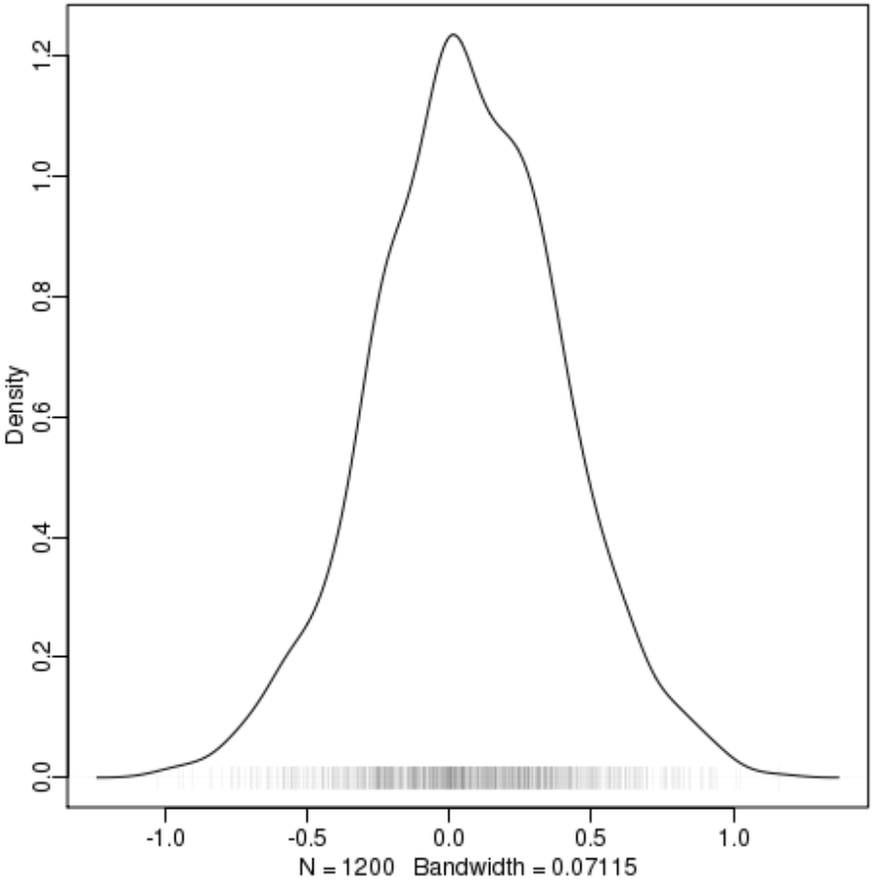
事後分布が推定された，としましょう

事後分布のサンプルを R で調べる

a のサンプリングの様子



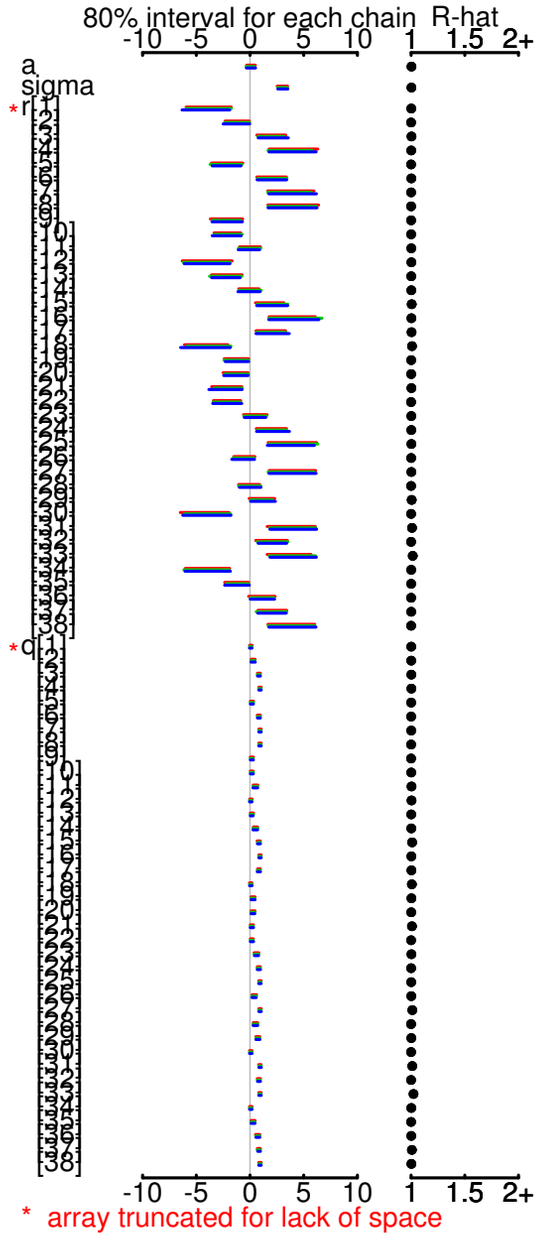
a の事後確率密度の推定

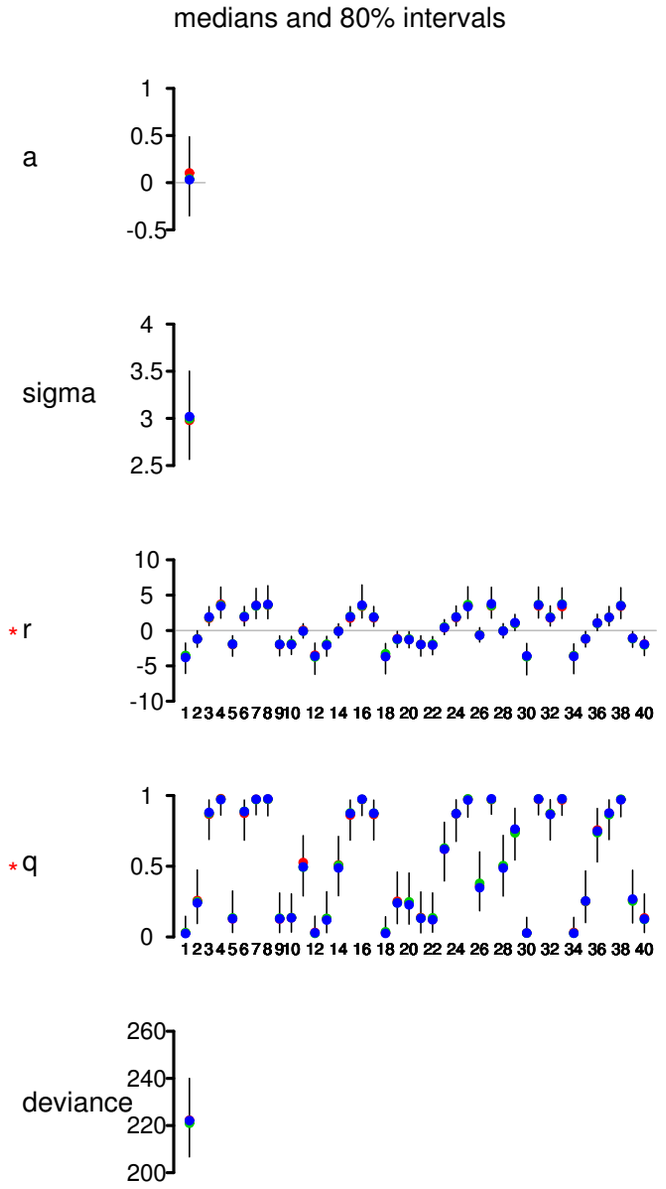


bugs オブジェクトの post.bugs を調べる (1)

- `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
  - $\hat{R} = \sqrt{\dfrac{\hat{\mathrm{var}}^{+}(\psi|y)}{W}}$
  - $\hat{\mathrm{var}}^{+}(\psi|y) = \dfrac{n-1}{n}W + \dfrac{1}{n}B$
  - Gelman et al. 2004. Bayesian Data Analysis. Chapman & Hall/CRC

/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 (2)

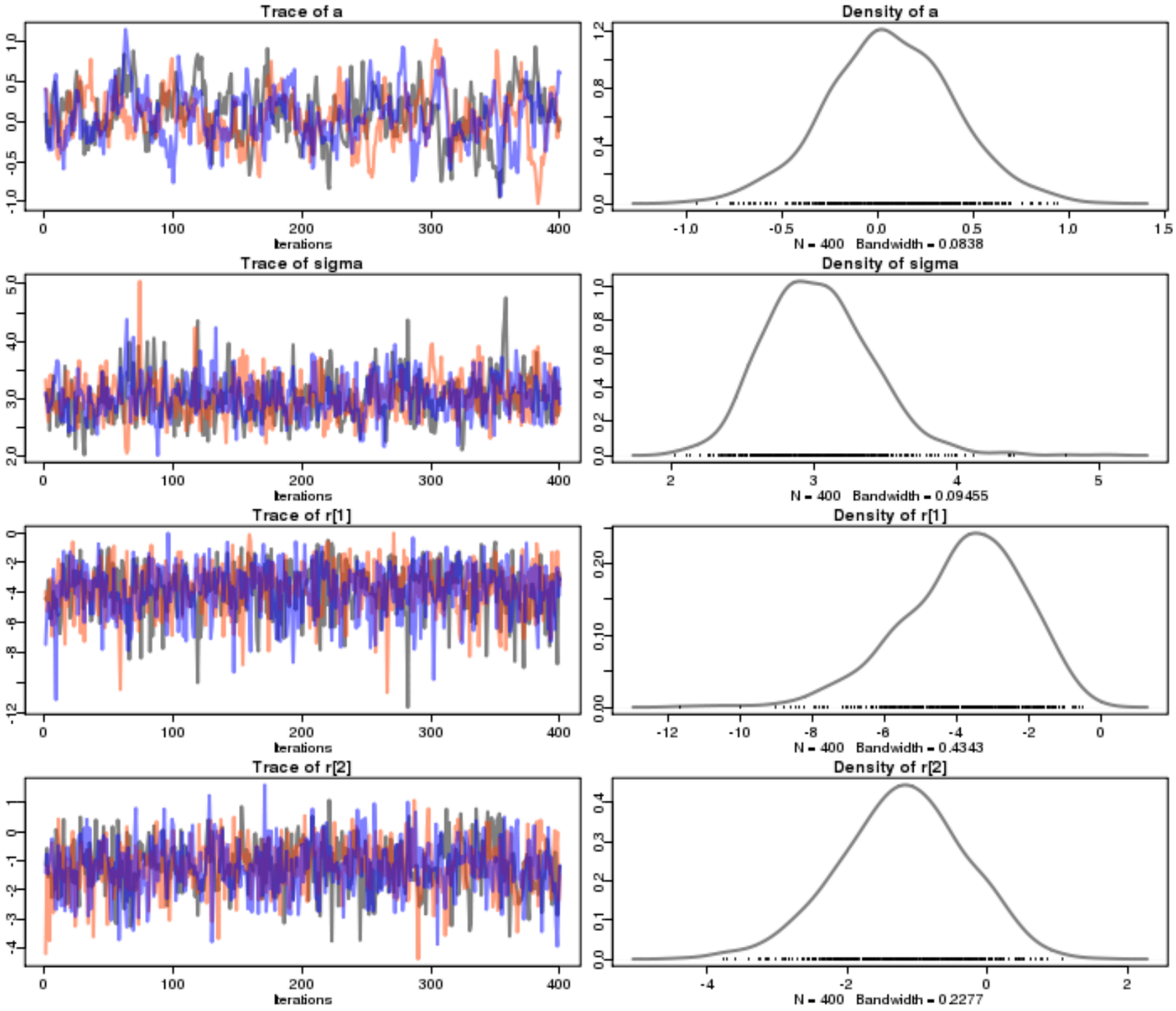
- `print(post.bugs, digits.summary = 3)`
- 事後分布の 95% 信頼区間などが表示される

	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

mcmc.list クラスに変換して作図

- `post.list <- to.list(post.bugs)`
- `plot(post.list[,1:4,], smooth = F)`

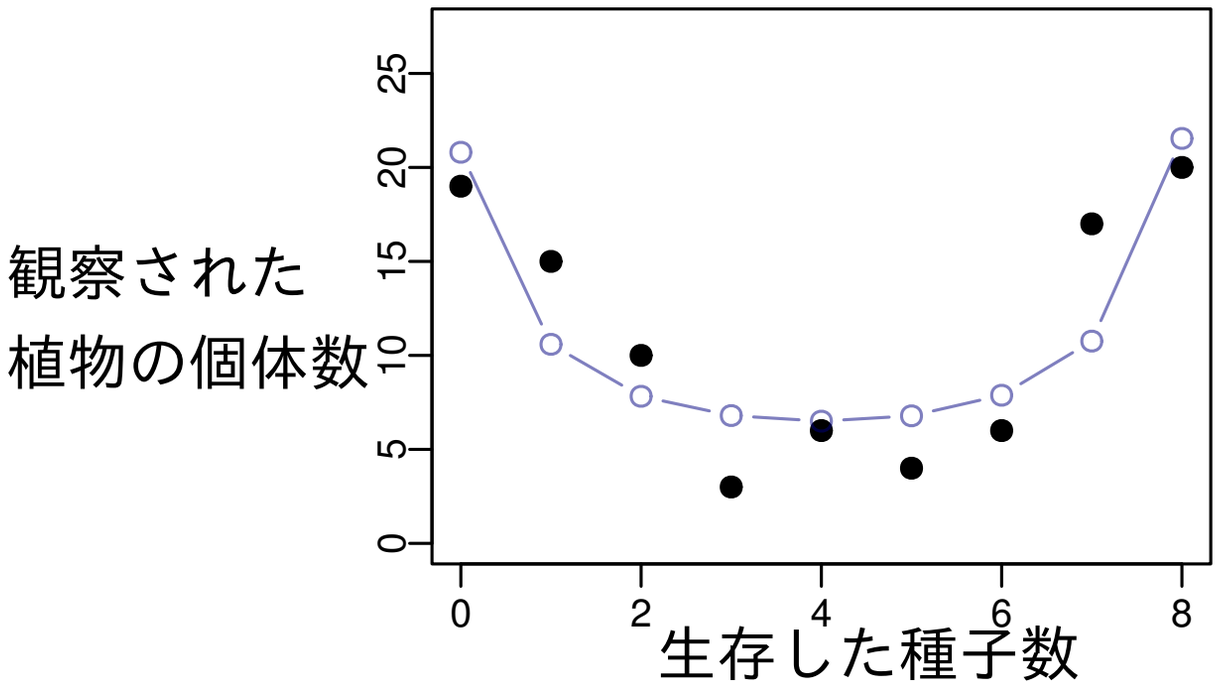
→ 次のページ, 実演表示


Trace of a
Density of a
N = 400 Bandwidth = 0.0838
Trace of sigma
Density of sigma
N = 400 Bandwidth = 0.09455
Trace of r[1]
Density of r[1]
N = 400 Bandwidth = 0.4343
Trace of r[2]
Density of r[2]
N = 400 Bandwidth = 0.2277
Iterations

mcmc クラスに変換して作図

- `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

例: 推定された事後分布に基づく予測



階層ベイズモデルのご利益とは?

階層ベイズモデルでないとうまく表現できない現象がある

- 複数の **random effects** (個体差・ブロック差・縦断的データ・……)
- 「隠れた」状態をあつかうモデル
  - 例: 「欠側値を補う」処理
- 空間構造ある問題も **MCMC** 計算で
  - 例: 「隣は似てるよ」効果 – **Gaussian Random Field**

今日，説明したこと

1. **GLM は階層ベイズモデル化する**
2. **MCMC をどんなソフトウェアで動かす?**

   まあ，**WinBUGS** + **R** が無難ではないでしょうか

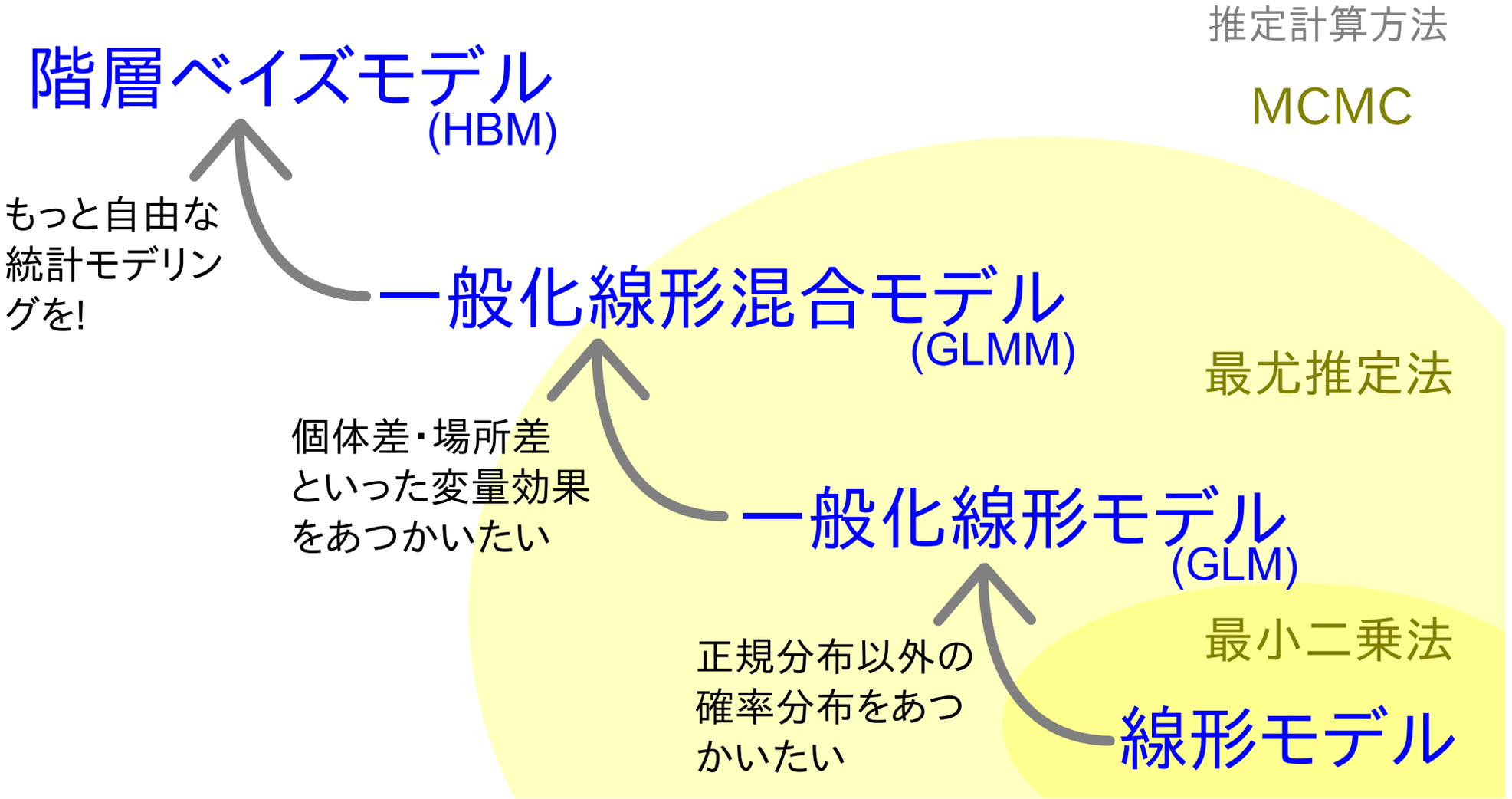
3. **WinBUGS を R で使う**

   R2WBwrapper 関数セットを経由して

4. **WinBUGS でパラメーターの事後分布推定**

   そして結果を **R** 内で解析・作図・変換する

線形モデルの発展


推定計算方法
階層ベイズモデル
(HBM)
MCMC
もっと自由な
統計モデリン
グを!
一般化線形混合モデル
(GLMM)
最尤推定法
個体差・場所差
といった変量効果
をあつかいたい
一般化線形モデル
(GLM)
最小二乗法
正規分布以外の
確率分布をあつ
かいたい
線形モデル