

9.4.3 いつまでMCMCすればいい？

Q4. 収束しない、なぜ？どうすれば？

A4. 以下を点検し、試行錯誤する

- 不適切なモデリング
- BUGSコードの間違い
- データの間違い
- パラメータの初期値が変

9.5 MCMCサンプルから事後分布推定

モデル書いた、MCMCした、収束した
で、次は何を見ればいいのか？

 得られた**事後分布**を見てみよう

 この中に格納されています

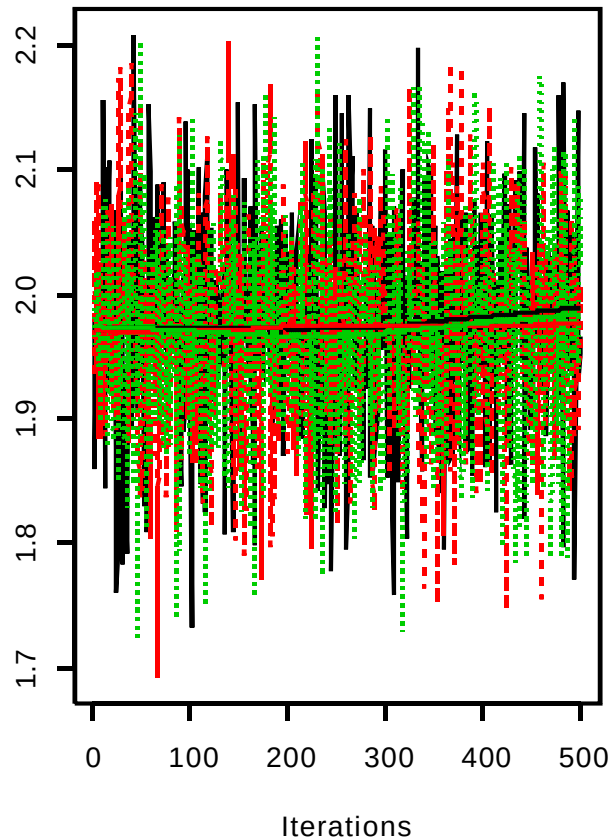
```
post.bugs <- call.bugs(  
  file = "model.bug.txt",  
  n.iter = 1600, n.burnin = 100, n.thin = 3)
```

9.5 MCMCサンプルから事後分布推定

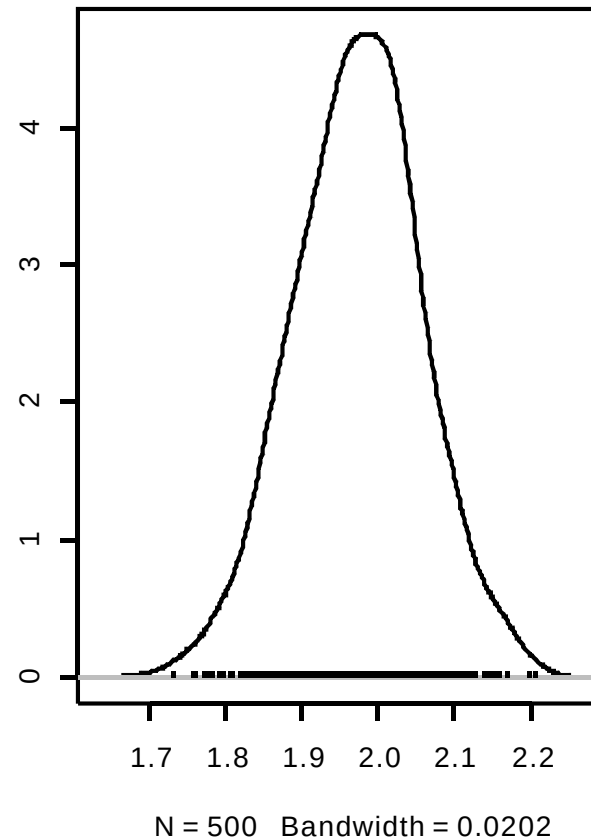
教科書の見方は省略

=定常状態 $\propto$ 尤度

β_1 のサンプル過程



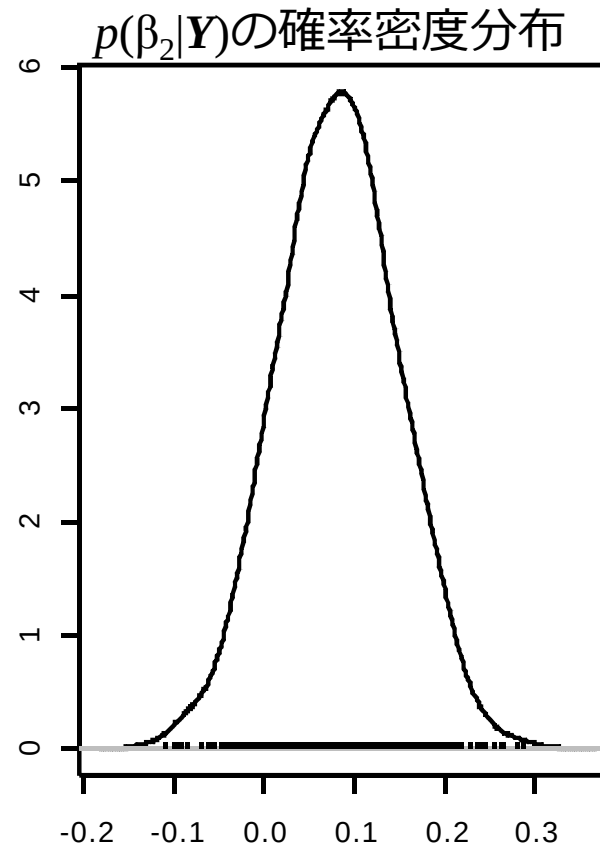
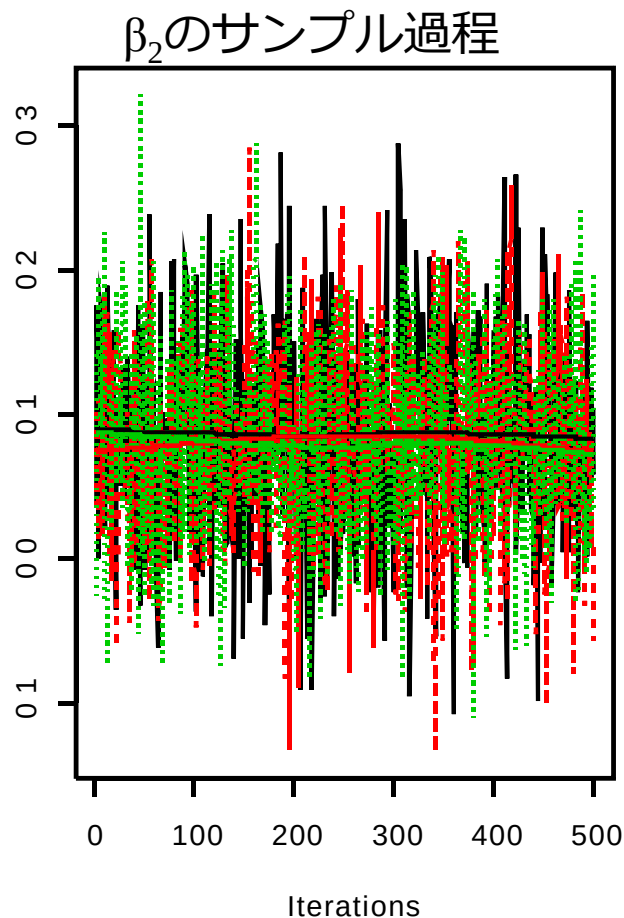
$p(\beta_1|Y)$ の確率密度分布



9.5 MCMCサンプルから事後分布推定

教科書の見方は省略

=定常状態 $\propto$ 尤度

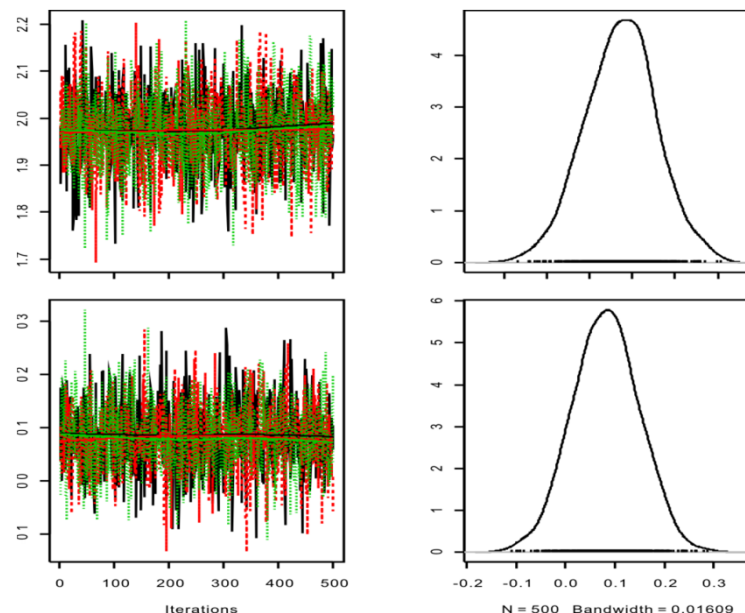


N = 500 Bandwidth = 0.01609

9.5 MCMCサンプルから事後分布推定

求められた事後分布
= 同時事後分布

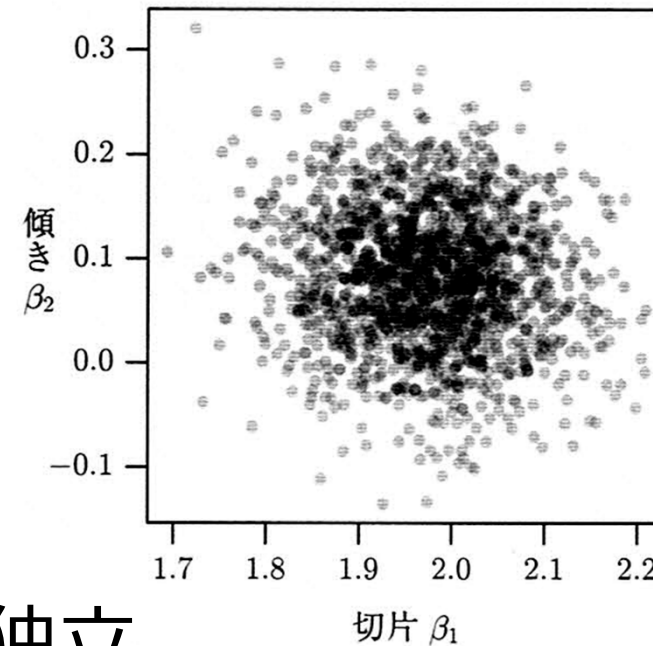
一方のパラメータで積分
➡ 周辺事後分布



ただし、MCMCでは互いに独立
同時事後分布 = 周辺事後分布
だから、区別せずに「事後分布」と言ってOK

9.5 MCMCサンプルから事後分布推定

(B) 事後分布 $p(\beta_1, \beta_2 | \mathbf{Y})$



ただし、MCMCでは互いに独立
同時事後分布 = 周辺事後分布
だから、区別せずに「事後分布」と言ってOK

9.5.1 事後分布の統計量（表示させ方）

post.bugs

ただだと数字が丸められてしまうので、

```
print(post.bugs, digits.summary=3)
```

「小数点第3位まで表示してね」の意味

意外と重要なポイント

9.5.1 事後分布の統計量 (出力の見方)

```
> print(post.bugs, digits.summary=3)
```

```
Inference for Bugs model at "e:/kubo/model.bug.txt", fit using WinBUGS,  
 3 chains, each with 1600 iterations (first 100 discarded), n.thin = 3  
n.sims = 1500 iterations saved
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
beta1	1.973	0.083	1.805	1.918	1.975	2.028	2.143	1.001	1500
beta2	0.082	0.067	-0.050	0.038	0.082	0.126	0.209	1.002	1200
deviance	92.001	1.962	90.090	90.560	91.400	92.772	97.390	1.000	1500

For each parameter, n.eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, $pD = \bar{D} - \hat{D}$)

$pD = 2.0$ and $DIC = 94.0$

DIC is an estimate of expected predictive error (lower deviance is better).

9.5.1 事後分布の統計量（出力の見方）

```
> print(post.bugs, digits.summary=3)
```

```
Inference for Bugs model at "e:/kubo/model.bug.txt", fit using WinBUGS,  
 3 chains, each with 1600 iterations (first 100 discarded), n.thin = 3  
n.sims = 1500 iterations saved
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
beta1	1.973	0.083	1.805	1.918	1.975	2.028	2.143	1.001	1500
beta2	0.082	0.067	-0.050	0.038	0.082	0.126	0.209	1.002	1200
deviance	92.001	1.962	90.090	90.560	91.400	92.772	97.390	1.000	1500

平均と標準偏差

中央値

For each parameter, n.eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, $pD = \bar{D} - \hat{D}$)

$pD = 2.0$ and $DIC = 94.0$

DIC is an estimate of expected predictive error (lower deviance is better).

9.5.1 事後分布の統計量 (出力の見方)

```
> print(post.bugs, digits.summary=3)
```

```
Inference for Bugs model at "e:/kubo/model.bug.txt", fit using WinBUGS,  
 3 chains, each with 1600 iterations (first 100 discarded), n.thin = 3  
n.sims = 1500 iterations saved
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
beta1	1.973	0.083	1.805	1.918	1.975	2.028	2.143	1.001	1500
beta2	0.082	0.067	-0.050	0.038	0.082	0.126	0.209	1.002	1200
deviance	92.001	1.962	90.090	90.560	91.400	92.772	97.390	1.000	1500

← 95 % 信用区間 →

R hat値

For each parameter, n.eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, $pD = \bar{D} - \hat{D}$)

$pD = 2.0$ and $DIC = 94.0$

DIC is an estimate of expected predictive error (lower deviance is better).

9.6 複数パラメータのMCMC

要点

- 1つ以外を固定とみなしてサンプリング
- 確率分布を作って、そこからサンプリング

ギブスサンプリングGibbs sampling

- **BUGS** = **B**ayesian inference **U**sing **G**ibbs **S**ampling
- Not メトロポリス法
(候補を決めて更新するかどうか考える)

詳細は省略します

9.7 この章のまとめ

- 無情報事前分布の採用
- GLMをBUGSコードで置き換えてみた
 - MCMCの条件の意味と決め方
- BUGSコードでWin BUGSを走らせた
- MCMCの結果の見方
 - 周辺事後分布とみなして良い
 - 巧妙なGIBBSサンプリング（9.6節省略）