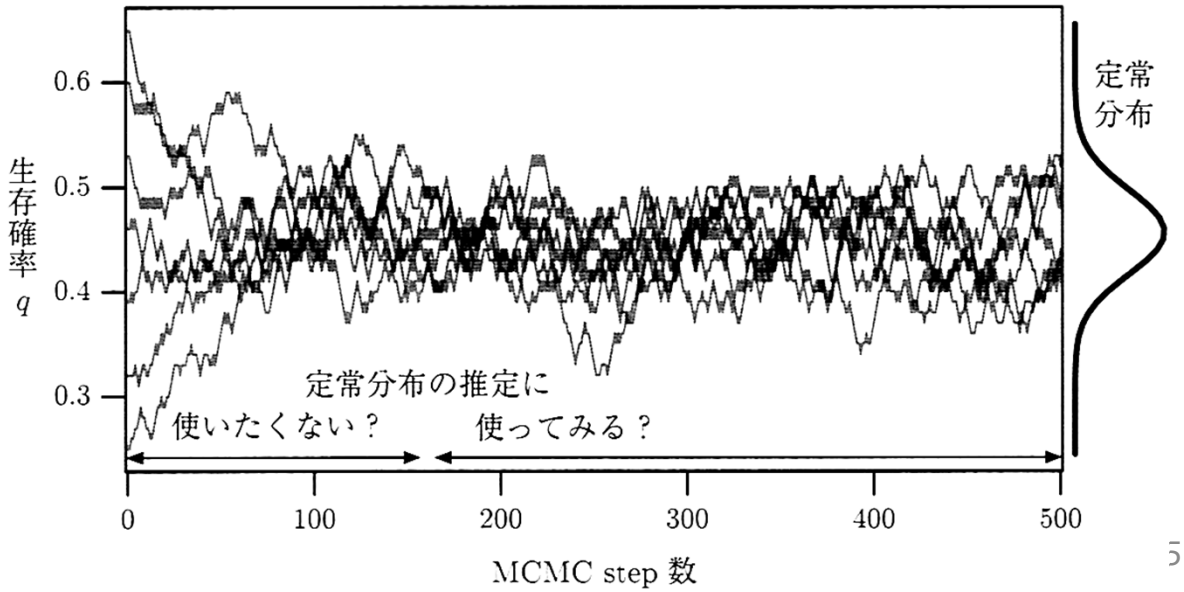
9.4.2 事後分布推定の準備

モデルの他にも準備が必要

- 初期値やサンプリング回数
- 何回おきにサンプリングするか
- 最初の何回を捨て去るか

図8.9



9.4.2 事後分布推定の準備

モデルの他にも準備が必要

- WinBUGSでもできるけど、使いにくい
- Rで済ませてしまいましょう
- それ用のR2WinBUGS packageがあるけど、やはり使いにくい
- ラッパー関数（久保先生謹製）を使う

これを前提にRコードを紹介

9.4.2 事後分布推定の準備

```
source("R2WBwrapper.R")
load("d.RData")

clear.data.param()
set.data("N", nrow(d))
set.data("Y", d$y)
set.data("X", d$x)
set.data("Mean.X", mean(d$x))

set.param("beta1", 0)
set.param("beta2", 0)

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

ちょっと長いので
前半と後半に分けて
解説しましょう

前半

後半

9.4.2 事後分布推定の準備（前半）

source("R2WBwrapper.R") ラッパー関数の呼び出し

load("d.RData") データをファイルから読み込む

```
clear.data.param()
set.data("N", nrow(d))
set.data("Y", d$y)
set.data("X", d$x)
set.data("Mean.X", mean(d$x))
```

9.4.2 事後分布推定の準備（前半）

```
source("R2WBwrapper.R")
load("d.RData")

clear.data.param()
set.data("N", nrow(d))
set.data("Y", d$y)
set.data("X", d$x)
set.data("Mean.X", mean(d$x))
```

（今回は初めてですが）
何度も繰り返し解析する場合に備え
前に使ったデータやパラメータを
メモリから消しておきます。
“お約束”です。

9.4.2 事後分布推定の準備（前半）

```
source("R2WBwrapper.R")
load("d.RData")

clear.data.param()
set.data("N", nrow(d))
set.data("Y", d$y)
set.data("X", d$x)
set.data("Mean.X", mean(d$x))
```

解析に使うデータを渡します。
WinBUGSでの名前, Rでの名前
の順に書いています。

9.4.2 事後分布推定の準備（後半）

```
set.param("beta1", 0)
set.param("beta2", 0)
```

どのパラメータを
推定すべきなのか
WinBUGSに教えてあげます。
その後に初期値も指定します。

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

9.4.2 事後分布推定の準備（後半）

```
set.param("beta1", 0)
set.param("beta2", 0)

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

()内の条件で
WinBUGSを呼び出し
解析結果をpost.bugsに
格納します。

9.4.2 事後分布推定の準備（後半）

```
set.param("beta1", 0)
set.param("beta2", 0)

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

モデルは”model.bug.txt”という名前のファイルに書いてあります。

9.4.2 事後分布推定の準備（後半）

```
set.param("beta1", 0)
set.param("beta2", 0)

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

MCMCのステップ数は1600
最初の100ステップは捨てる
3回に1回サンプリングする

9.4.3 いつまでMCMCすればいい？

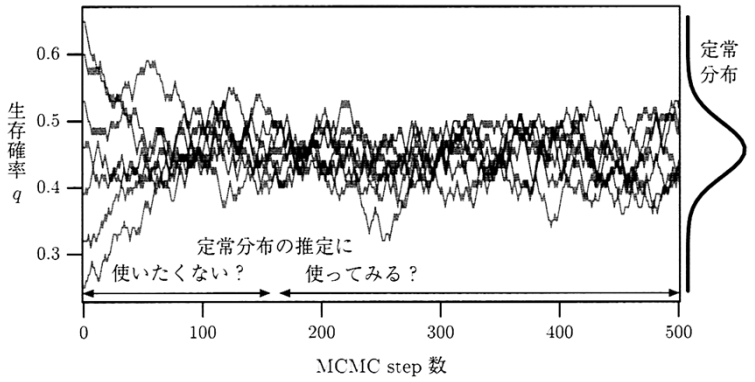
Q1. 先ほどのMCMCの条件はどう決める？

n.iter=1600, n.burnin=100, n.thin=3

A1. 試行錯誤するしかない

Q2. 何を基準に試行錯誤？

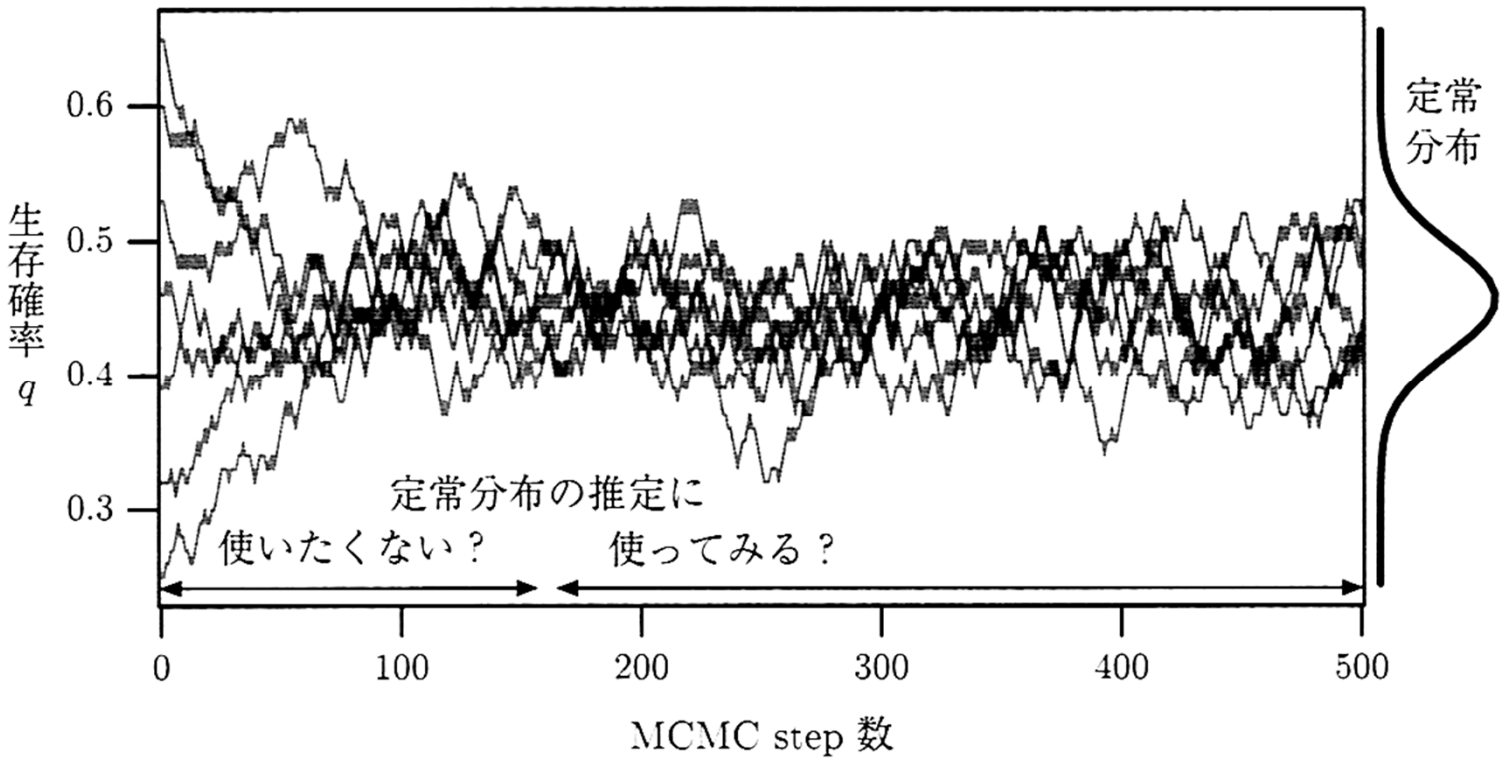
A2. 複数のchainを比較する



9.4.3 いつまでMCMCすればいい？

Q2. 何を基準に試行錯誤？

A2. 複数のchainを比較する



9.4.3 いつまでMCMCすればいい？

Q2. 何を基準に試行錯誤？
A2. 複数のchainを比較する

- 普通はchainの数はデフォルト=3でいい
- 初期値を引きずらないようにn.burnin
  - 捨て去る数
- 定常状態になるようn.iter
  - Chainの長さ

Q3. 何を基準に定常状態？

9.4.3 いつまでMCMCすればいい？

Q3. 何を基準に定常状態？

A3. $\hat{R}$(R hat)値を基準にするのがお手軽

$\hat{R} = \sqrt{\widehat{var}^+ / W}$, $\widehat{var}^+ = \frac{n-1}{n} W + \frac{1}{n} B$

Wはchainごとの分散の平均、

Bはchain間の分散

- Rが計算してくれる
- 完全な収束で$\hat{R} = 1.0$になる
- $\hat{R} < 1.1$を目安に定常状態を判断（**収束判定**）