

MCMCサンプルを生成する際に留意すべき点

MCMCサンプルの生成での留意点

☑️ 代表性

チェーンは事後分布の一部に留まることがなく、事後分布を代表した値をとっている。
（最適な方法は存在しないが）チェーンの遷移を視覚的に確認する。

☑️ 正確性

チェーン間で結果が大きくばらついていない。チェーンが固まりを形成していない。
安定で正確な分布を得るために、大きなサンプルサイズを得る。

☑️ 効率性

チェーンが可能な限り少ないステップで生成されている。
並列処理を行う。サンプリング方法を変更する。

RでMCMCサンプリングしてみる

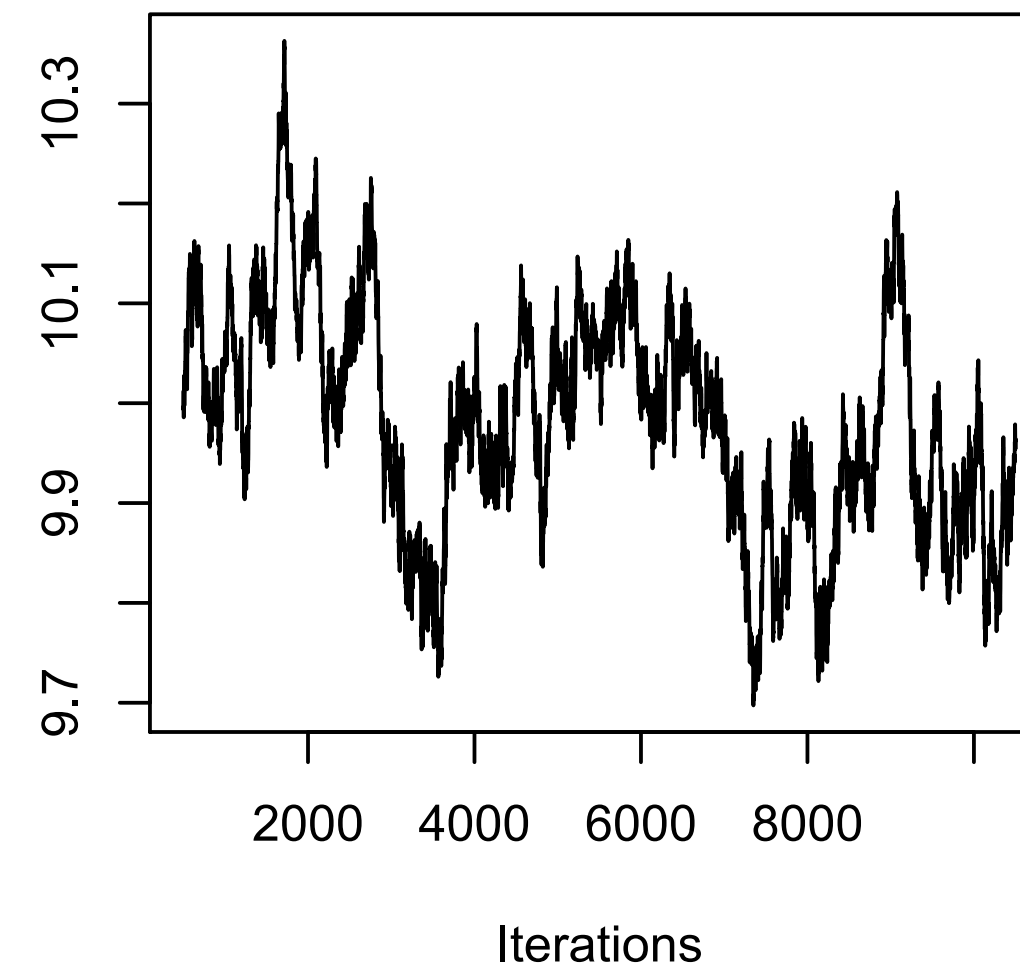
RでのMCMCサンプリングの記述方法

```
post <- MCMCmetrop1R(llnormfun,  
  theta.init = c(5, 5),  
  x = x,  
  thin = 10,  
  mcmc = 11000,  
  burnin = 1000,  
  tune = 1,  
  verbose = 1000,  
  logfun = TRUE)
```

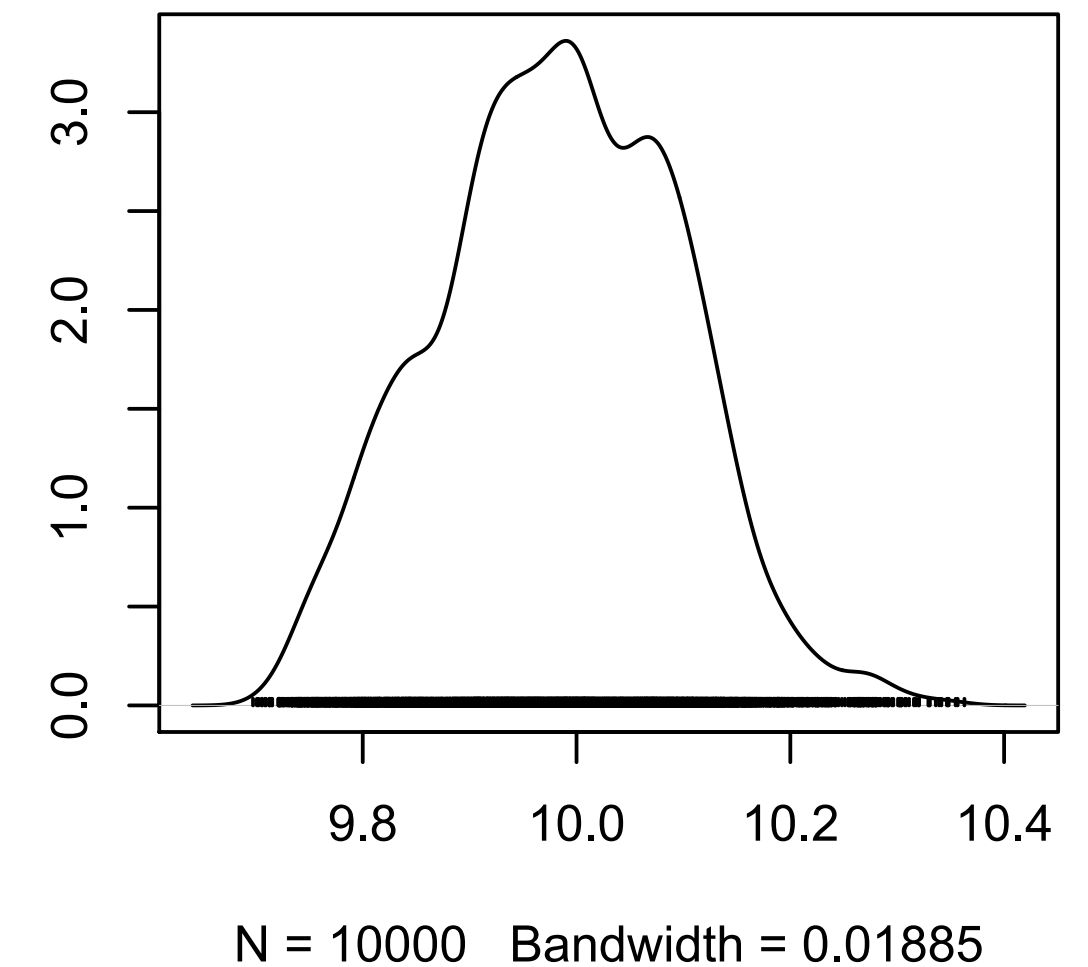
burn-in period
初期値に影響されやすい最初の部分

thinning
間引いたチェーンは元のチェーンに比べ情報
が少なく、不安定・不正確になりやすい

Trace of var1



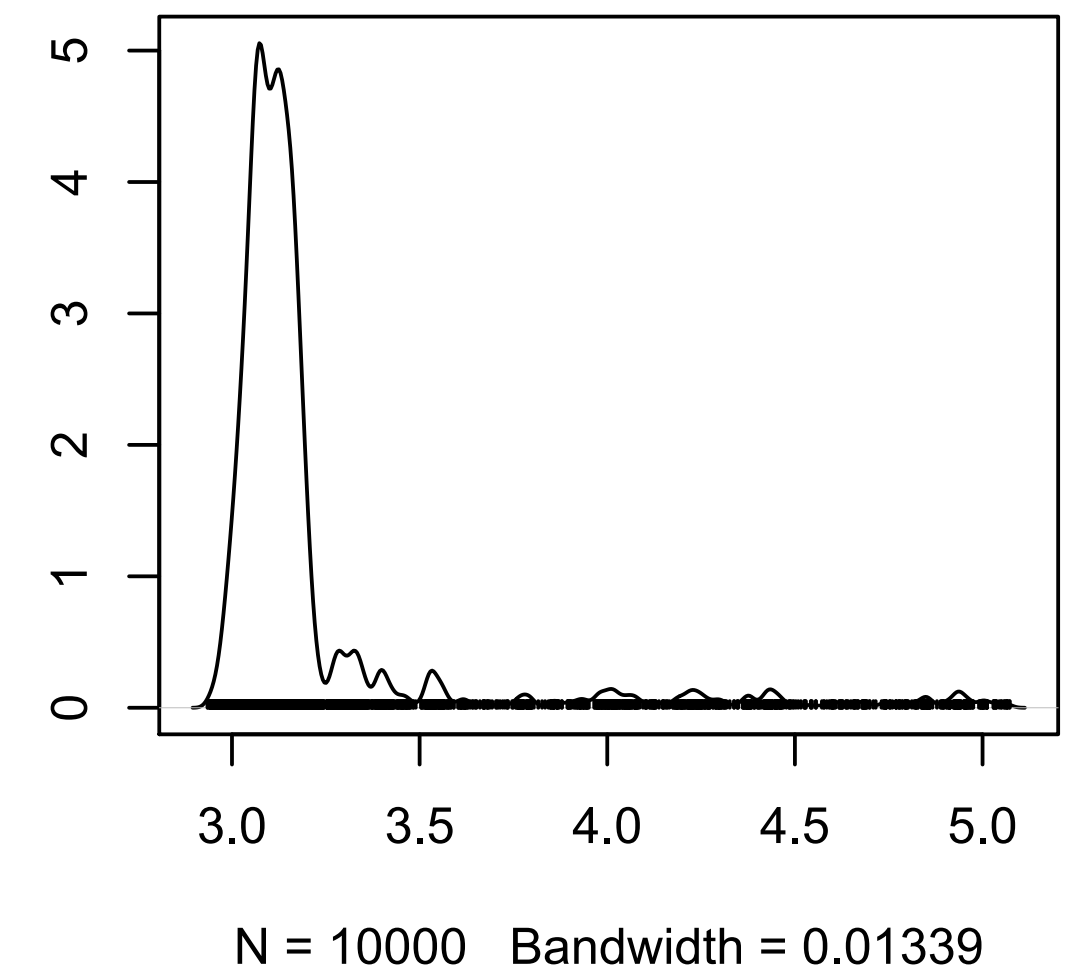
Density of var1



Trace of var2



Density of var2



RでMCMCサンプリングしてみる

RでのMCMCサンプリングの記述方法

```
MCMCmetrop1R iteration 1 of 10500  
function value = -2844.46243  
theta =  
  10.00000  
   6.00000  
Metropolis acceptance rate = 0.00000
```

```
MCMCmetrop1R iteration 501 of 10500  
function value = -2551.83746  
theta =  
  9.92133  
  3.16396  
Metropolis acceptance rate = 0.55489
```

```
MCMCmetrop1R iteration 1001 of 10500  
function value = -2551.38024  
theta =  
  9.95881  
  3.09470  
Metropolis acceptance rate = 0.56743
```

RでMCMCサンプリングしてみる

MCMCサンプリングの結果

```
> summary(post.list)
Iterations = 501:10500
Thinning interval = 1
Number of chains = 3
Sample size per chain = 10000
```

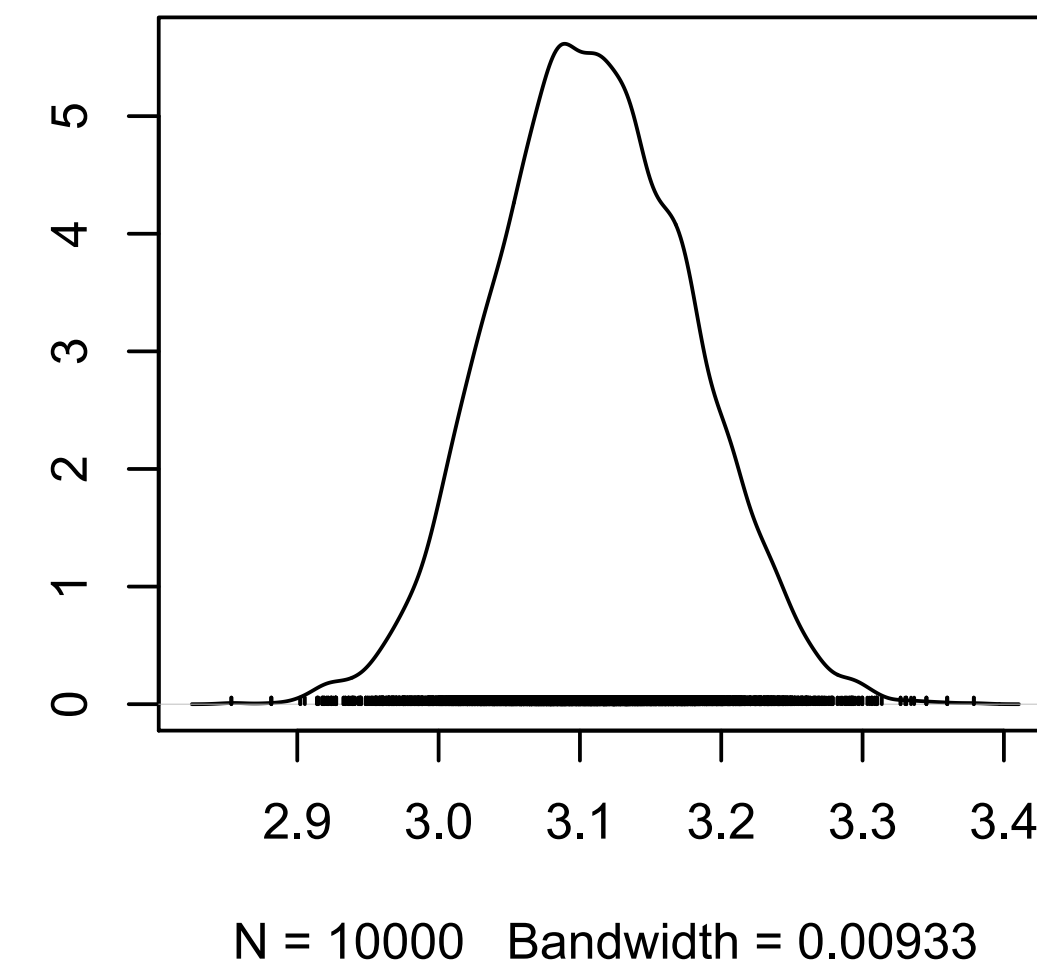
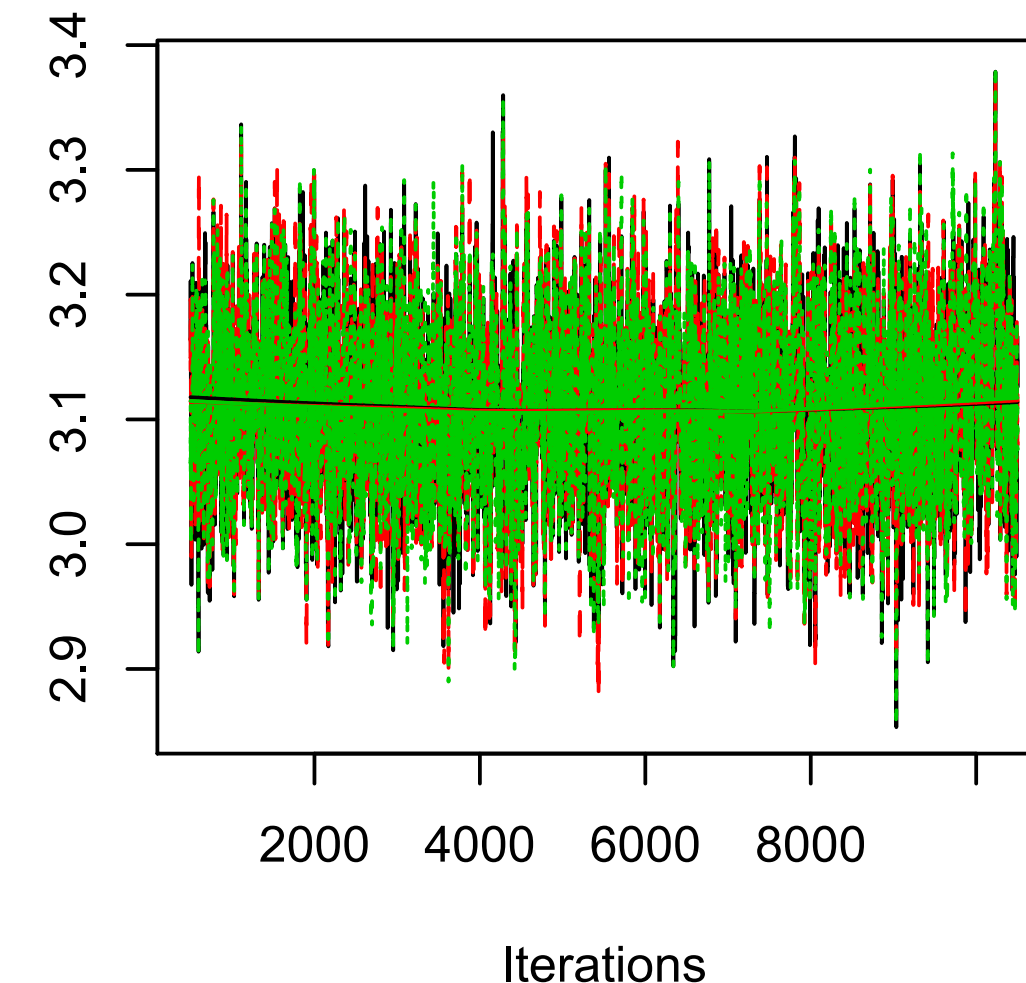
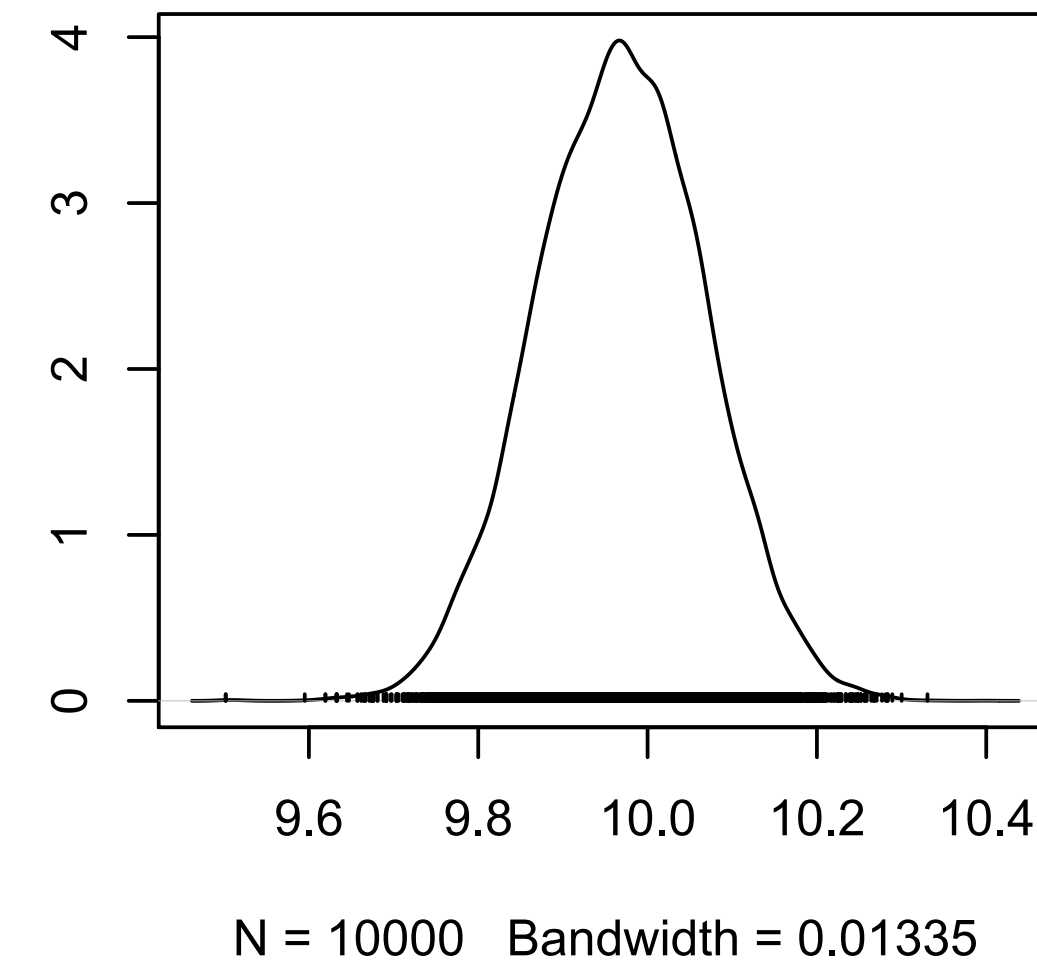
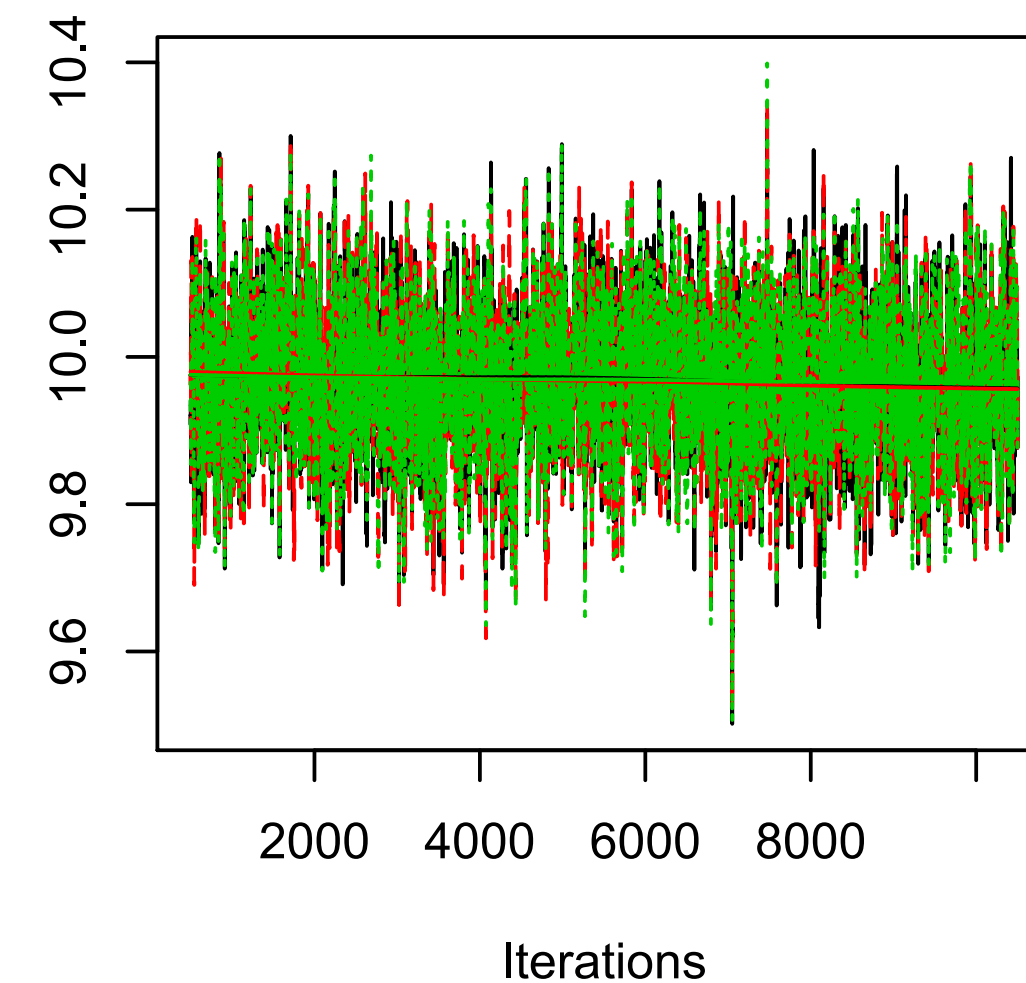
1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
[1,]	9.968	0.09898	0.0005715	0.001803
[2,]	3.109	0.06919	0.0003994	0.001239

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
[1,]	9.775	9.900	9.969	10.035	10.160
[2,]	2.978	3.062	3.108	3.157	3.245

信用区間



RでMCMCサンプリングしてみる

MCMCの収束判定

Gelman-Rubin 統計量

```
> gelman.diag(post.list)
Potential scale reduction factors:
```

	Point est.	Upper C.I.
[1,]	1	1
[2,]	1	1

```
Multivariate psrf
```

```
1
```

Gewekeの診断

```
> geweke.diag(post.list)
[[1]]
```

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
```

var1	var2
1.109	1.115

```
[[2]]
```

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
```

var1	var2
1.2288	0.6596

RでMCMCサンプリングしてみる

MCMCの収束判定

Raftery and Lewis の診断

```
> raftery.diag(post.list)
[[1]]
```

```
Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95
```

```
[[2]]
```

```
Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95
```

Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
19	20797	3746	5.55
16	17260	3746	4.61

Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
16	17504	3746	4.67
17	18114	3746	4.84

自己相関が抽出数とともに小さくなっているか確認する

チェーン内の自己相関

```
>autocorr.plot(post.list)
```

質のよいサンプル

自己相関が抽出数とともに小さくなる

