Bayes_stat_hw3

Na SeungChan

2024-11-23

문제 풀이 전반에 걸쳐 적용되는 사항

사후분포를 직접 구하고, 합격 확률을 식으로 정리하는 등 수식 계산이 필요한 부분은 종이로 필요하여 스캔하였고, 난수 계산 등은 R markdown으로 풀이하였다. 난수 생성 시 seed는 42를 사용하였다.

1.9.12

(a)

해당 문제는 종이에 풀이하였다.

(b)

초기화

```
m = 10000
rho = 0.99
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)
```

우선, theta1 = theta2 = 0으로 초기화하였다. 이 문제에서는 우선 m = 10000, rho = 0.99로 두었다.

메트로폴리스-헤이스팅스 반복

```
set.seed(42)
for (i in 1:m) { #m = 10000회 동안 반복.
  proposal_theta <- rcauchy(2, location = 0, scale = 1) #제안분포에서 난수 생성
  u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
  accp_prob <-
    min(1,((1+(proposal_theta[1])^2)*(1+(proposal_theta[2])^2)*exp(((po.theta1[i])^2+(po.theta2[i])^2-
    (proposal_theta[1])^2-(proposal_theta[2])^2-2*rho*(po.theta1[i]*po.theta2[i]-
    proposal_theta[1]*proposal_theta[2]))/(2*(1-
    (rho)^2))))/((1+(po.theta1[i])^2)*(1+(po.theta2[i])^2)))
```

```r
  if(accp_prob >= u){
    po.theta1 <- c(po.theta1, proposal_theta[1])
    po.theta2 <- c(po.theta2, proposal_theta[2])
  } else{
    po.theta1 <- c(po.theta1, po.theta1[i])
    po.theta2 <- c(po.theta2, po.theta2[i])
  }
}
```

확률변수 확인

```r
head(po.theta1)
```

```
## [1]  0.0000000 -0.2742241 -0.2742241 -0.2742241 -0.2742241 -0.2742241
```

```r
head(po.theta2)
```

```
## [1]  0.0000000 -0.2002994 -0.2002994 -0.2002994 -0.2002994 -0.2002994
```

이와 같이 추출된 po.theta1과 po.theta2는 이변량정규분포를 불변분포로 갖는 마르코프 체인이다.

(c)

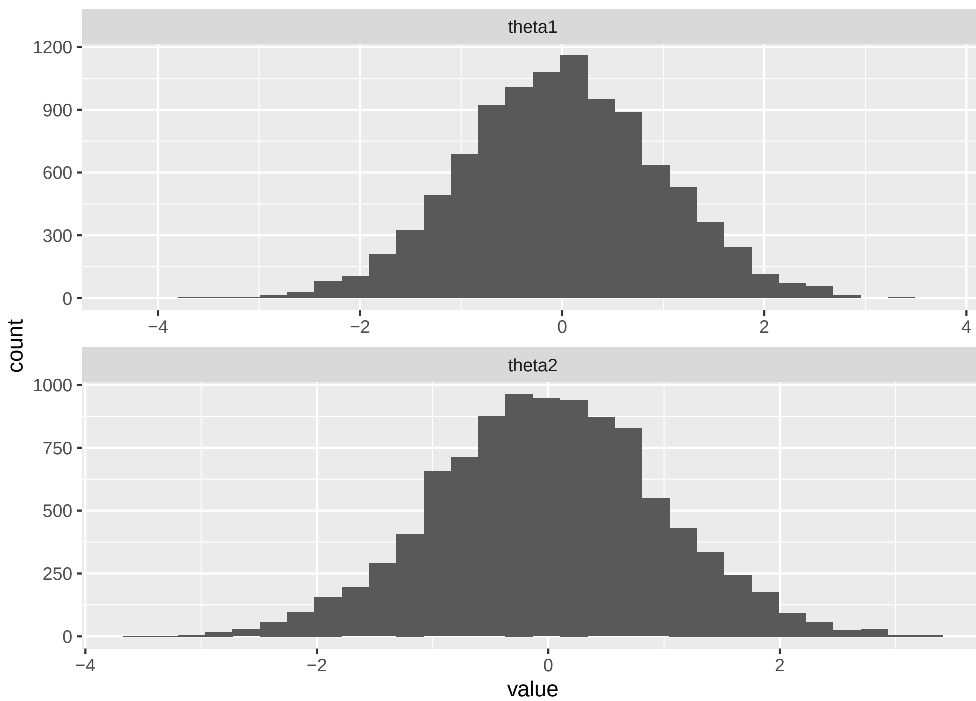
사후표본 추출 - $\rho = 0.3$

```r
m = 10000
rho = 0.3
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)

set.seed(42) #seed는 42로 고정.
for (i in 1:m) { #m = 10000회 동안 반복.
  proposal_theta <- rcauchy(2, location = 0, scale = 1) #제안분포에서 난수 생성
  u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
  accp_prob <-
  ↪  min(1,((1+(proposal_theta[1])^2)*(1+(proposal_theta[2])^2)*exp(((po.theta1[i])^2+(po.theta2[i])^2-
  ↪  (proposal_theta[1])^2-(proposal_theta[2])^2-2*rho*(po.theta1[i]*po.theta2[i]-
  ↪  proposal_theta[1]*proposal_theta[2]))/(2*(1-
  ↪  (rho)^2))))/((1+(po.theta1[i])^2)*(1+(po.theta2[i])^2)))
  if(accp_prob >= u){
    po.theta1 <- c(po.theta1, proposal_theta[1])
    po.theta2 <- c(po.theta2, proposal_theta[2])
```

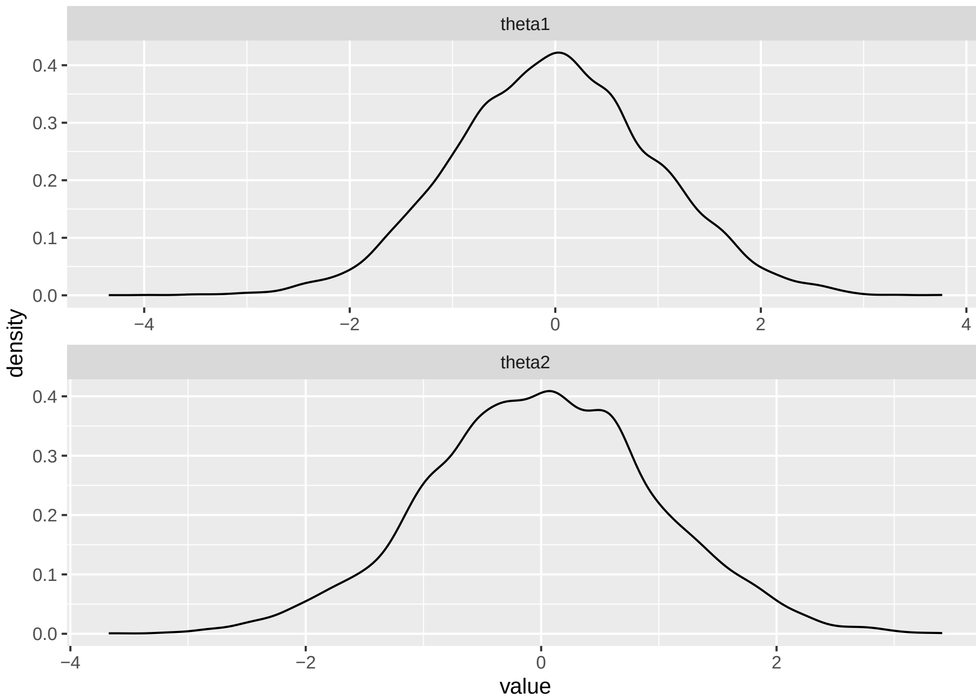
```r
  } else{
    po.theta1 <- c(po.theta1, po.theta1[i])
    po.theta2 <- c(po.theta2, po.theta2[i])
  }
}

post_1203 <- data.frame(theta1 = po.theta1, theta2 = po.theta2)
```

```r
post_1203 %>% mcmc %>% ggs %>% ggs_histogram()
```

히스토그램

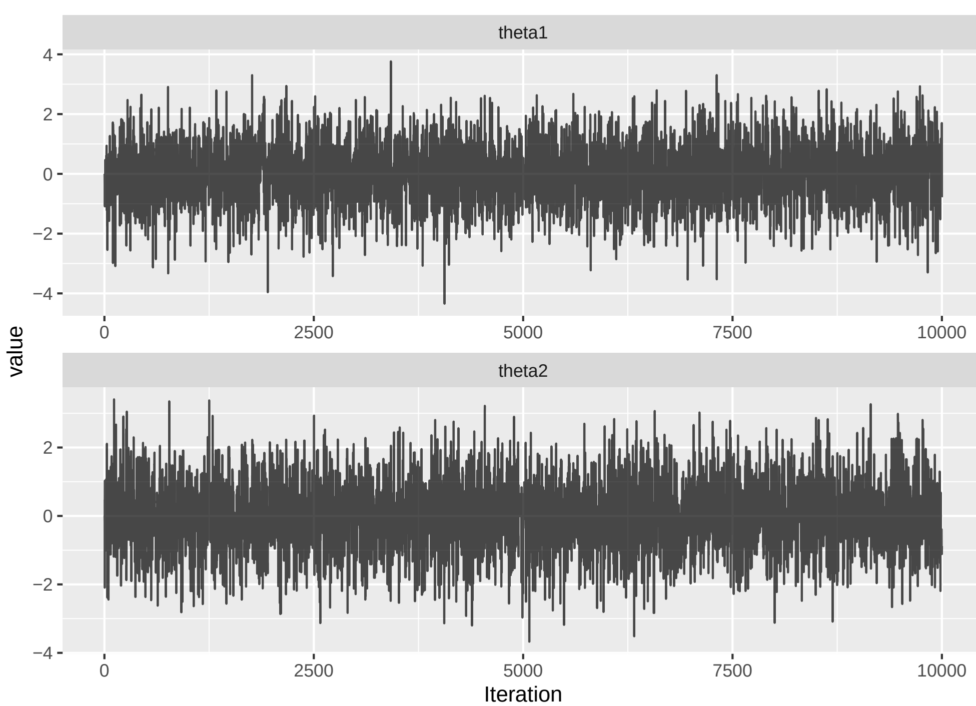



```r
post_1203 %>% mcmc %>% ggs %>% ggs_density()
```



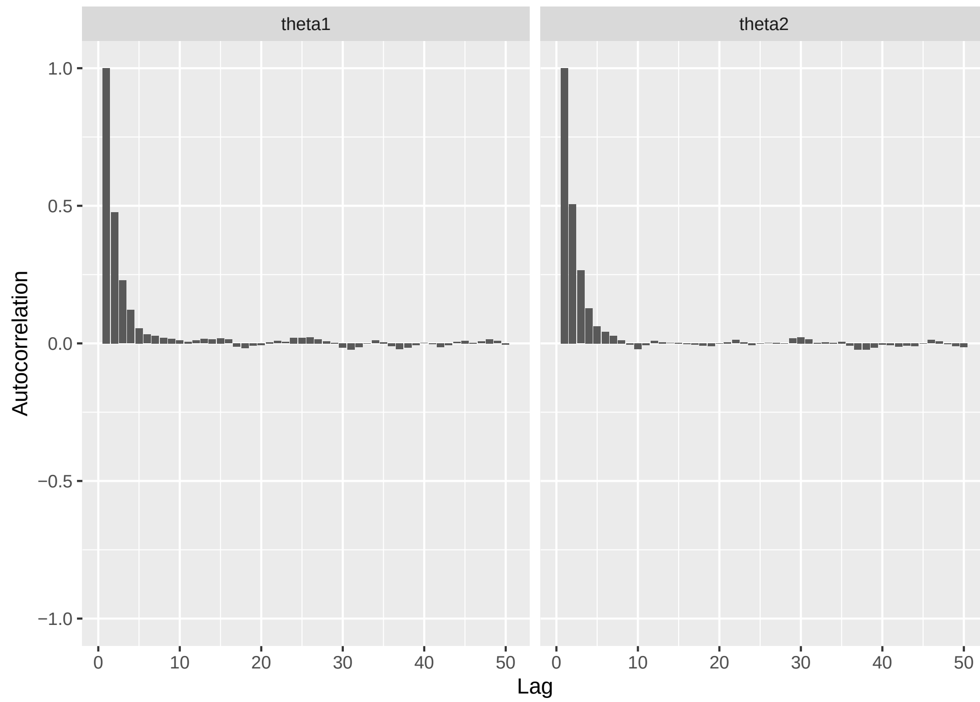

```
post_1203 %>% mcmc %>% ggs %>% ggs_traceplot()
```

시계열 그림



```r
post_1203 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```

자기상관계수 그림



rho = 0.3에서는 시계열 그림이 특별한 경향을 보이지 않고, 자기상관계수 그림에서 확실히 자기상관계수가 감소하는 것으로 보아 마르코프 체인이 수렴하였다.

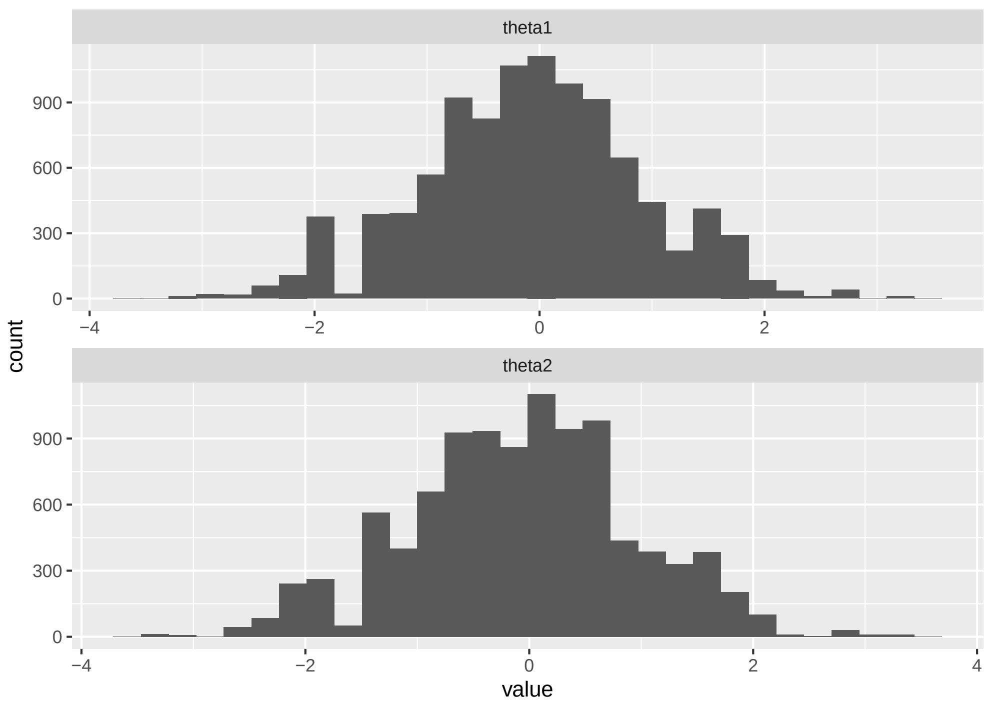
사후표본 추출 - rho = 0.99

```r
m = 10000
rho = 0.99
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)

set.seed(42) #seed는 42로 고정.
for (i in 1:m) { #m = 10000회 동안 반복.
  proposal_theta <- rcauchy(2, location = 0, scale = 1) #제안분포에서 난수 생성
  u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
  accp_prob <-
↪    min(1,((1+(proposal_theta[1])^2)*(1+(proposal_theta[2])^2)*exp(((po.theta1[i])^2+(po.theta2[i])^2-
↪    (proposal_theta[1])^2-(proposal_theta[2])^2-2*rho*(po.theta1[i]*po.theta2[i]-
↪    proposal_theta[1]*proposal_theta[2]))/(2*(1-
↪    (rho)^2))))/((1+(po.theta1[i])^2)*(1+(po.theta2[i])^2)))
```

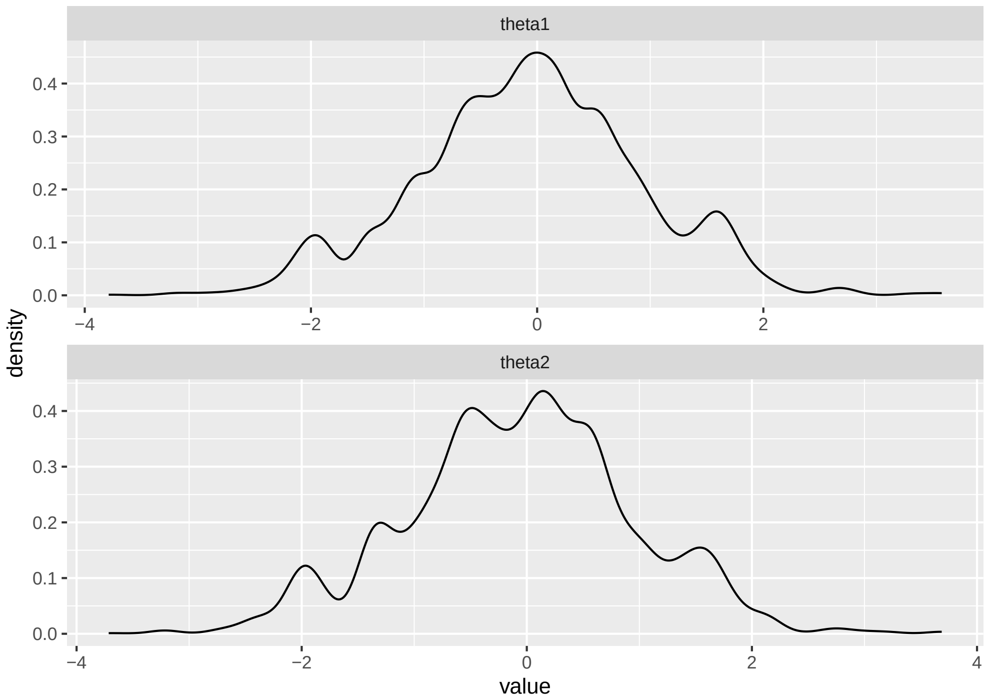
```r
  if(accp_prob >= u){
    po.theta1 <- c(po.theta1, proposal_theta[1])
    po.theta2 <- c(po.theta2, proposal_theta[2])
  } else{
    po.theta1 <- c(po.theta1, po.theta1[i])
    po.theta2 <- c(po.theta2, po.theta2[i])
  }
}

post_1299 <- data.frame(theta1 = po.theta1, theta2 = po.theta2)
```

```r
post_1299 %>% mcmc %>% ggs %>% ggs_histogram()
```

히스토그램

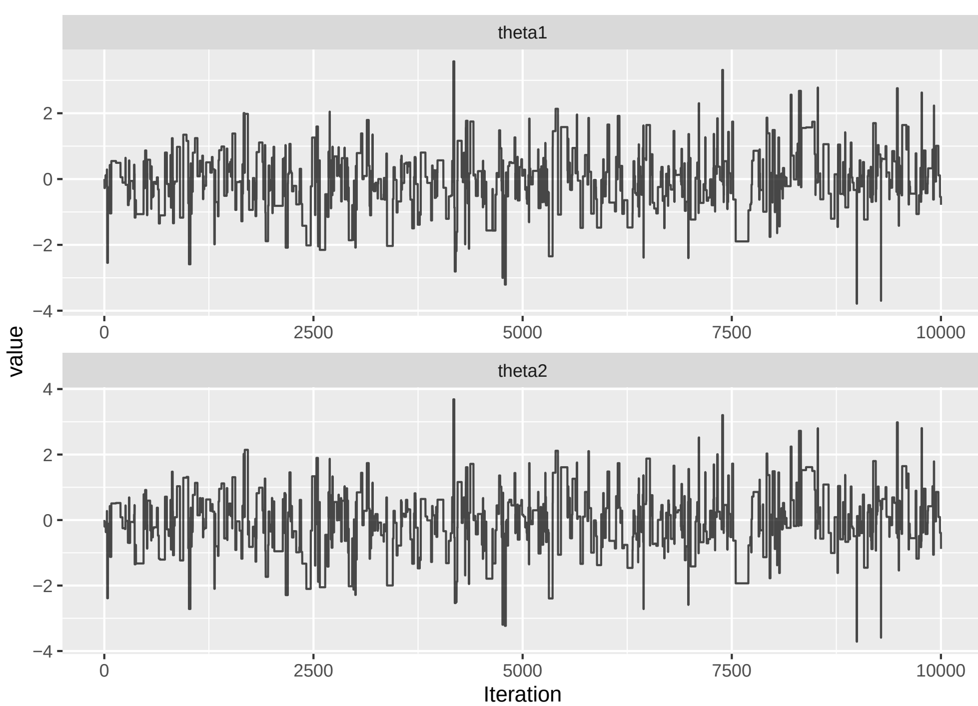



```r
post_1299 %>% mcmc %>% ggs %>% ggs_density()
```



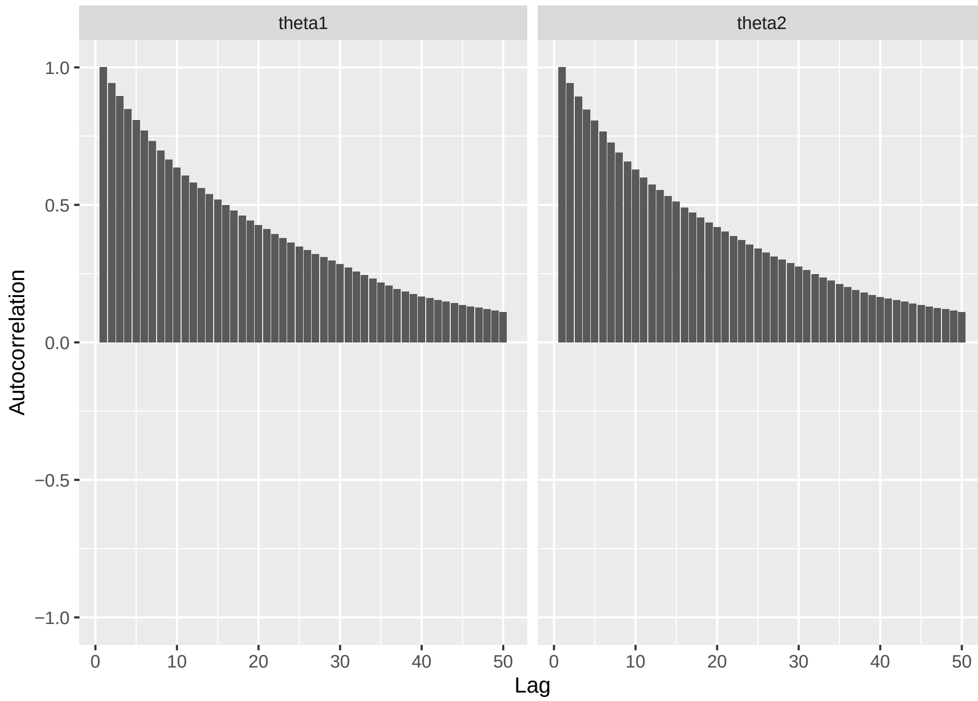

```
post_1299 %>% mcmc %>% ggs %>% ggs_traceplot()
```

시계열 그림



```r
post_1299 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```

자기상관계수 그림



rho = 0.99에서는 히스토그램이 정규분포의 모양이 아니고, 자기상관계수 그림에서 자기상관계수가 느리게 감소하고, 시계열 그림에서 아직 특정한 경향이 보이는 등 마르코프 체인이 수렴하지 않았다. 이 경우 m이 부족하므로 표본의 수를 늘려야 한다.

(d)

$\rho = 0.3$

```r
post_1203 %>% mcmc %>% summary
```

```
##
## Iterations = 1:10001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean     SD Naive SE Time-series SE
## theta1 -0.005994 0.9758 0.009758        0.01641
```

```
## theta2  0.003006 0.9766 0.009766       0.01706
##
## 2. Quantiles for each variable:
##
##         2.5%     25%       50%    75% 97.5%
## theta1 -1.882 -0.6763 -0.007666 0.6252 1.913
## theta2 -1.946 -0.6400  0.006172 0.6391 1.930
```

2.5% 25% 50% 75% 97.5% Mean SD

theta1 -1.882 -0.6763 -0.007666 0.6252 1.913 -0.005994 0.9758 theta2 -1.946 -0.6400 0.006172 0.6391 1.930 0.003006 0.9766

$\rho = 0.99$

```
post_1299 %>% mcmc %>% summary
```

```
##
## Iterations = 1:10001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean     SD Naive SE Time-series SE
## theta1 -0.06298 0.9914 0.009914       0.06733
## theta2 -0.06421 1.0126 0.010126       0.06817
##
## 2. Quantiles for each variable:
##
##         2.5%     25%      50%    75% 97.5%
## theta1 -2.032 -0.6890 -0.05691 0.5526 1.772
## theta2 -2.052 -0.6667 -0.03836 0.5900 1.801
```

2.5% 25% 50% 75% 97.5% Mean SD

theta1 -2.032 -0.6890 -0.05691 0.5526 1.772 -0.06298 0.9914 theta2 -2.052 -0.6667 -0.03836 0.5900 1.801 -0.06421 1.0126

1.9.13

(a)

해당 문제는 종이에 풀이하였다.

(b)

초기화

```
m = 5000
rho = 0.99
cov_mtx <- matrix(c(1, rho, rho, 1), nrow = 2)
d = 1 #d는 적절한 합격률이 되도록 해야 함.
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)
```

우선, theta1 = theta2 = 0으로 초기화하였다. 이 문제에서는 우선 m = 5000, rho = 0.99로 두었다.

메트로폴리스-헤이스팅스 반복

```
set.seed(42)
for (i in 1:m) {#m = 5000회 동안 반복.
 proposal_theta1 <- rnorm(1, po.theta1[i], d)
 proposal_theta2 <- rcauchy(1, location = 0, scale = 1) #제안분포에서 난수 생성
 u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
 accp_prob <- min(1, (dmvnorm(c(proposal_theta1, proposal_theta2), c(0, 0),
↪   cov_mtx)*dcauchy(po.theta2[i], 0, 1))/(dmvnorm(c(po.theta1[i], po.theta2[i]), c(0, 0),
↪   cov_mtx)*dcauchy(proposal_theta2, 0, 1)))
 if(accp_prob >= u){
  po.theta1 <- c(po.theta1, proposal_theta1)
  po.theta2 <- c(po.theta2, proposal_theta2)
 } else{
  po.theta1 <- c(po.theta1, po.theta1[i])
  po.theta2 <- c(po.theta2, po.theta2[i])
 }
}
```

확률변수 확인

```
head(po.theta1)
```

```
## [1] 0 0 0 0 0 0
```

```
head(po.theta2)
```

```
## [1] 0 0 0 0 0 0
```

이와 같이 추출된 po.theta1과 po.theta2는 이변량정규분포를 불변분포로 갖는 마르코프 체인이다.

(c)

사후분포를 생성하기 전, 시범적인 난수 생성을 통해 d의 값을 결정해야 한다. seed를 지금까지 써 왔던 42가 아니라 31로 써서, $\rho = 0.99$와 0.3 각각에 대해 적절한 d 값을 찾아보겠다.

```
m = 500
rho = 0.99
cov_mtx <- matrix(c(1, rho, rho, 1), nrow = 2)
d = 0.1 #d는 적절한 합격률이 되도록 해야 함.
accepted = 0
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)

set.seed(31)
for (i in 1:m) {#m = 5000회 동안 반복.
 proposal_theta1 <- rnorm(1, po.theta1[i], d)
 proposal_theta2 <- rcauchy(1, location = 0, scale = 1) #제안분포에서 난수 생성
 u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
 accp_prob <- min(1, (dmvnorm(c(proposal_theta1, proposal_theta2), c(0, 0),
↪    cov_mtx)*dcauchy(po.theta2[i], 0, 1))/(dmvnorm(c(po.theta1[i], po.theta2[i]), c(0, 0),
↪    cov_mtx)*dcauchy(proposal_theta2, 0, 1)))
 if(accp_prob >= u){
  accepted <- accepted + 1
  po.theta1 <- c(po.theta1, proposal_theta1)
  po.theta2 <- c(po.theta2, proposal_theta2)
 } else{
  po.theta1 <- c(po.theta1, po.theta1[i])
  po.theta2 <- c(po.theta2, po.theta2[i])
 }
}

accepted/500
```

```
## [1] 0.12
```

$\rho = 0.99$ 케이스에서는 d값이 지나치게 작아지는 것이 바람직하지 않아 보여 0.25를 달성하지 못하고 성능 개선이 낮은 수준인 0.1 수준에서 중단하였다.

```
m = 500
rho = 0.3
cov_mtx <- matrix(c(1, rho, rho, 1), nrow = 2)
d = 3 #d는 적절한 합격률이 되도록 해야 함.
accepted = 0
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)
```

```r
set.seed(31)
for (i in 1:m) {#m = 5000회 동안 반복.
  proposal_theta1 <- rnorm(1, po.theta1[i], d)
  proposal_theta2 <- rcauchy(1, location = 0, scale = 1) #제안분포에서 난수 생성
  u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
  accp_prob <- min(1, (dmvnorm(c(proposal_theta1, proposal_theta2), c(0, 0),
  ↪  cov_mtx)*dcauchy(po.theta2[i], 0, 1))/(dmvnorm(c(po.theta1[i], po.theta2[i]), c(0, 0),
  ↪  cov_mtx)*dcauchy(proposal_theta2, 0, 1)))
  if(accp_prob >= u){
    accepted <- accepted + 1
    po.theta1 <- c(po.theta1, proposal_theta1)
    po.theta2 <- c(po.theta2, proposal_theta2)
  } else{
    po.theta1 <- c(po.theta1, po.theta1[i])
    po.theta2 <- c(po.theta2, po.theta2[i])
  }
}

accepted/500
```

```
## [1] 0.244
```

ρ = 0.3 케이스에서는 d = 3 수준이 적절한 합격률을 보였다. 이제 이 값을 바탕으로 각각 5000개 사후표본을 생성한다.

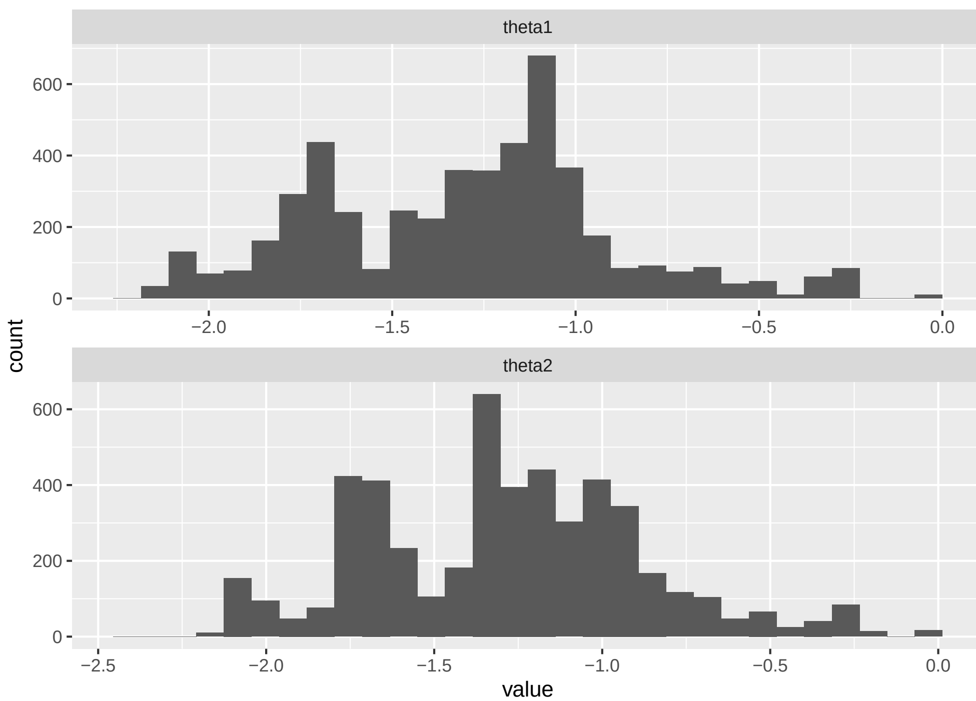
ρ = 0.99

```r
m = 5000
rho = 0.99
cov_mtx <- matrix(c(1, rho, rho, 1), nrow = 2)
d = 0.1
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)

set.seed(42)
for (i in 1:m) {#m = 5000회 동안 반복.
  proposal_theta1 <- rnorm(1, po.theta1[i], d)
  proposal_theta2 <- rcauchy(1, location = 0, scale = 1) #제안분포에서 난수 생성
  u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
  accp_prob <- min(1, (dmvnorm(c(proposal_theta1, proposal_theta2), c(0, 0),
  ↪  cov_mtx)*dcauchy(po.theta2[i], 0, 1))/(dmvnorm(c(po.theta1[i], po.theta2[i]), c(0, 0),
  ↪  cov_mtx)*dcauchy(proposal_theta2, 0, 1)))
  if(accp_prob >= u){
    po.theta1 <- c(po.theta1, proposal_theta1)
    po.theta2 <- c(po.theta2, proposal_theta2)
```

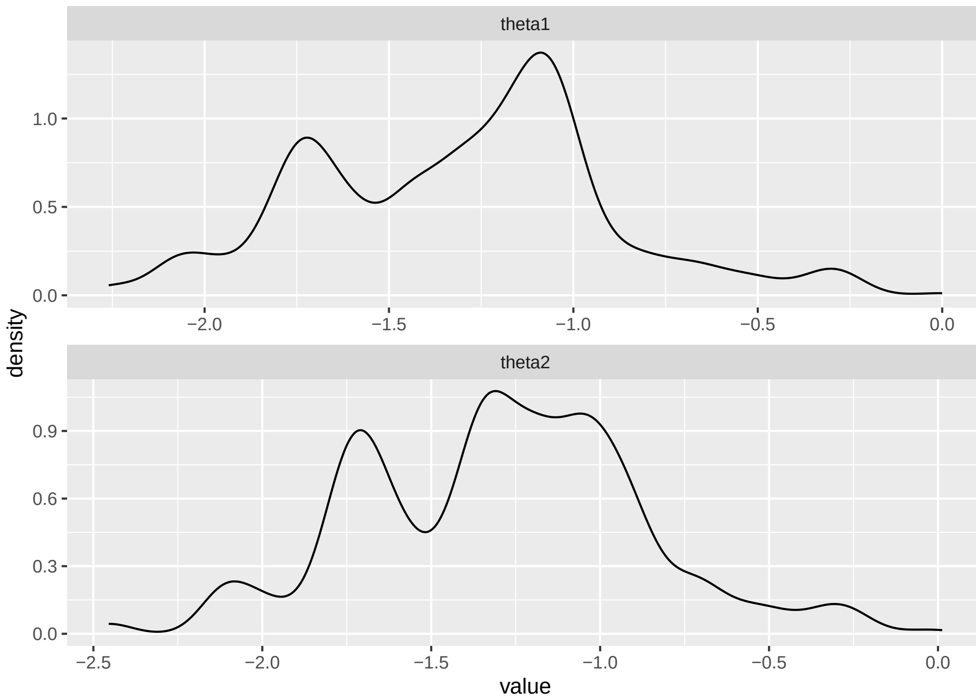
```r
  } else{
    po.theta1 <- c(po.theta1, po.theta1[i])
    po.theta2 <- c(po.theta2, po.theta2[i])
  }
}

post_1399 <- data.frame(theta1 = po.theta1, theta2 = po.theta2)
```

```r
post_1399 %>% mcmc %>% ggs %>% ggs_histogram()
```

히스토그램

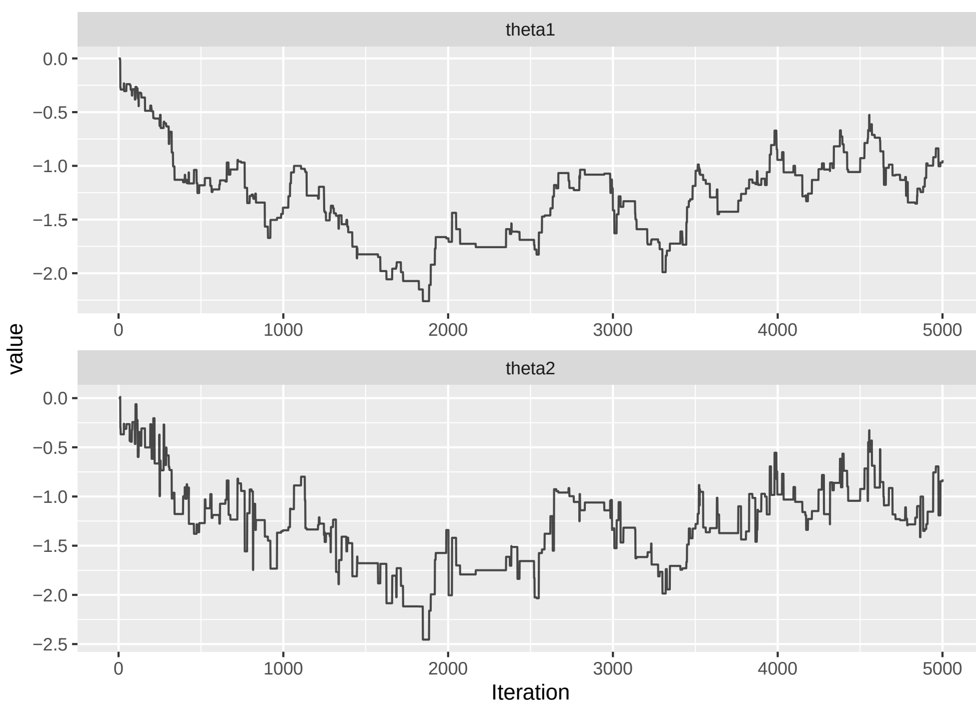



```r
post_1399 %>% mcmc %>% ggs %>% ggs_density()
```



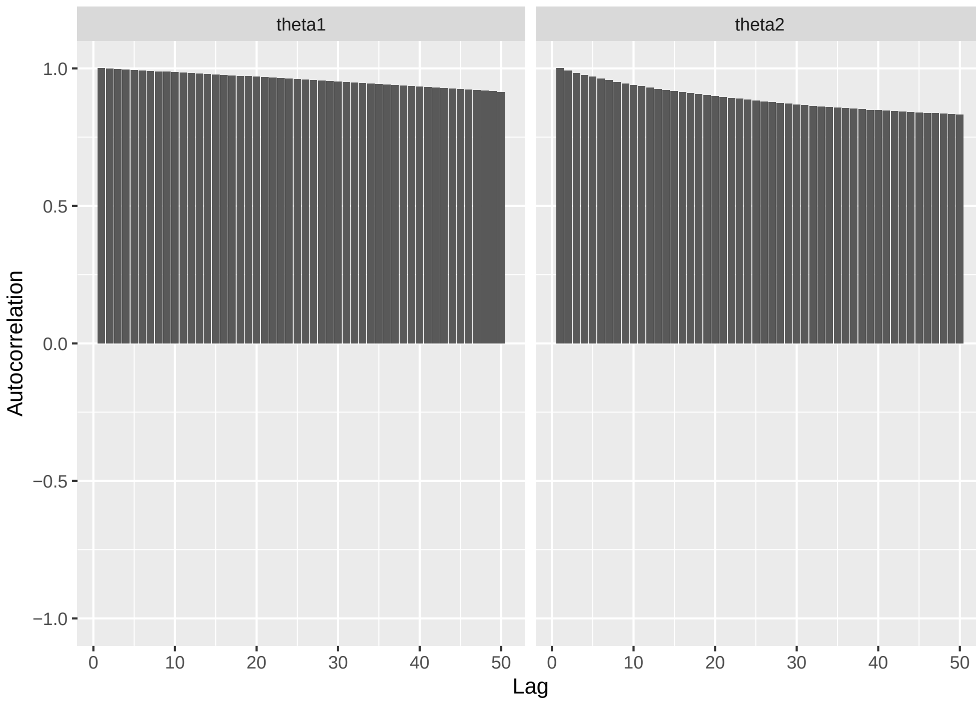

```r
post_1399 %>% mcmc %>% ggs %>% ggs_traceplot()
```

시계열 그림



```r
post_1399 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```

자기상관계수 그림



시계열 자료가 아주 강력하게 패턴을 보인다. 마르코프 체인이 수렴하기 위해 더 많은 반복 수가 필요해 보인다. 20만 회 반복 이후 thining을 시도하였으나 컴퓨터 사양 문제로 실패하였다.

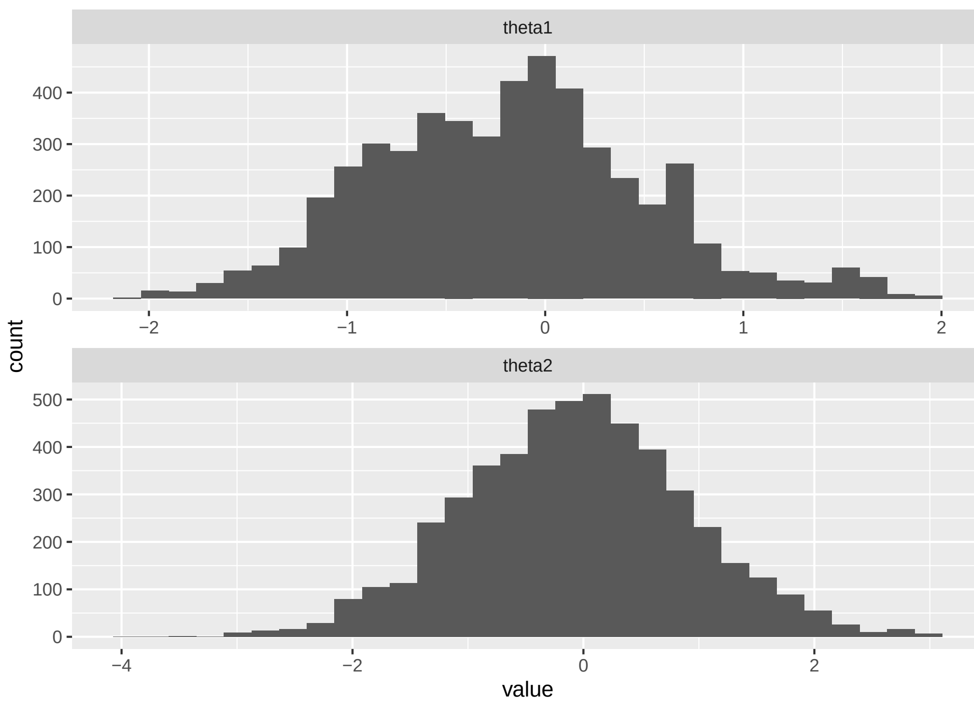
$\rho = 0.3$

```r
m = 5000
rho = 0.3
cov_mtx <- matrix(c(1, rho, rho, 1), nrow = 2)
d = 0.1
po.theta1 = NULL
po.theta2 = NULL
theta1 = 0
theta2 = 0
po.theta1 = c(po.theta1, theta1)
po.theta2 = c(po.theta2, theta2)

set.seed(42)
for (i in 1:m) {#m = 5000회 동안 반복.
  proposal_theta1 <- rnorm(1, po.theta1[i], d)
  proposal_theta2 <- rcauchy(1, location = 0, scale = 1) #제안분포에서 난수 생성
  u <- runif(1, min = 0, max = 1) #합격-불합격 판정용 난수 생성
```

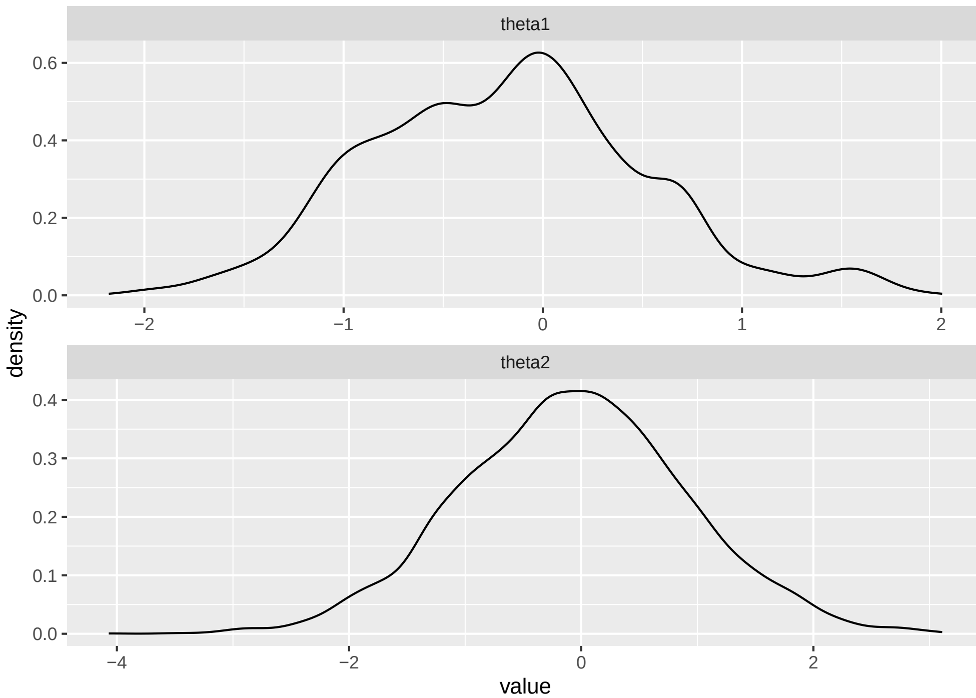
```r
  accp_prob <- min(1, (dmvnorm(c(proposal_theta1, proposal_theta2), c(0, 0),
  ↪   cov_mtx)*dcauchy(po.theta2[i], 0, 1))/(dmvnorm(c(po.theta1[i], po.theta2[i]), c(0, 0),
  ↪   cov_mtx)*dcauchy(proposal_theta2, 0, 1)))

  if(accp_prob >= u){
   po.theta1 <- c(po.theta1, proposal_theta1)
   po.theta2 <- c(po.theta2, proposal_theta2)
  } else{
   po.theta1 <- c(po.theta1, po.theta1[i])
   po.theta2 <- c(po.theta2, po.theta2[i])
  }
}

post_1303 <- data.frame(theta1 = po.theta1, theta2 = po.theta2)
```

```r
post_1303 %>% mcmc %>% ggs %>% ggs_histogram()
```

히스토그램

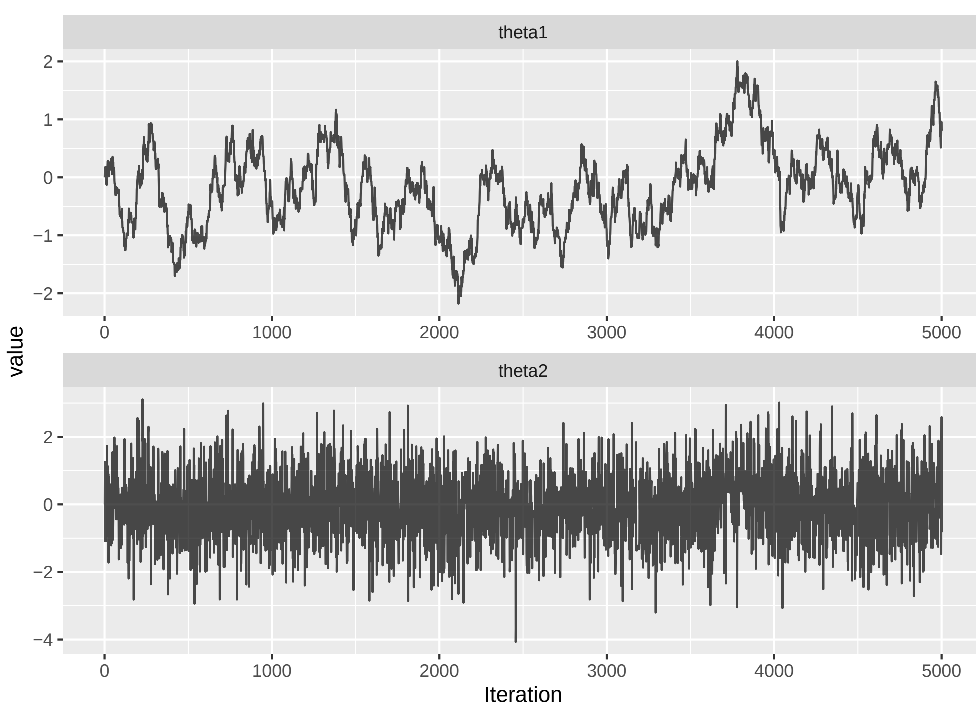



```r
post_1303 %>% mcmc %>% ggs %>% ggs_density()
```



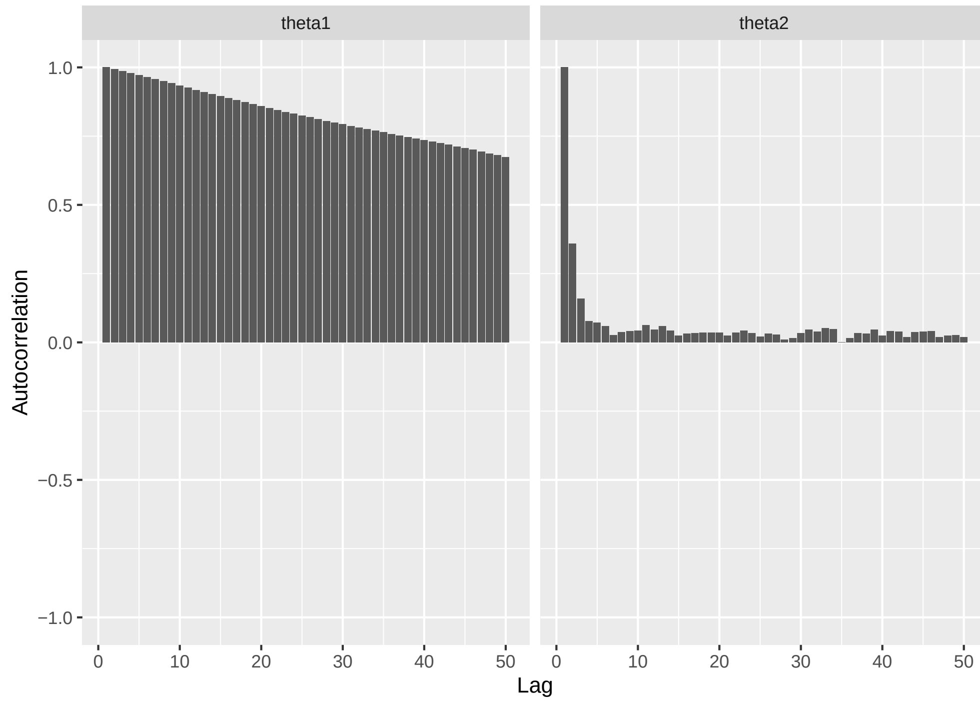

```
post_1303 %>% mcmc %>% ggs %>% ggs_traceplot()
```

시계열 그림



```r
post_1303 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```

자기상관계수 그림



이 경우, theta2에서는 수렴했다고 볼 수 있으나 theta1에서는 수렴했다고 볼 수 없다.

(d)

$\rho = 0.3$

```r
post_1303 %>% mcmc %>% summary
```

```
##
## Iterations = 1:5001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean     SD Naive SE Time-series SE
## theta1 -0.18266 0.6894 0.009748         0.1589
## theta2 -0.06557 0.9733 0.013763         0.0256
##
```

```
## 2. Quantiles for each variable:
##
##          2.5%     25%      50%    75% 97.5%
## theta1 -1.460 -0.6772 -0.17002 0.2410 1.431
## theta2 -1.986 -0.7232 -0.06294 0.5715 1.860
```

	2.5%	25%	50%	75%	97.5%	Mean	SD
theta1	-1.460	-0.6772	-0.17002	0.2410	1.431	-0.18266	0.6894
theta2	-1.986	-0.7232	-0.06294	0.5715	1.860	-0.06557	0.9733

$\rho = 0.3$

```
post_1399 %>% mcmc %>% summary
```

```
##
## Iterations = 1:5001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean     SD Naive SE Time-series SE
## theta1 -1.302 0.4080 0.005770         0.1641
## theta2 -1.292 0.4189 0.005923         0.1158
##
## 2. Quantiles for each variable:
##
##          2.5%    25%    50%    75%   97.5%
## theta1 -2.073 -1.634 -1.261 -1.061 -0.3456
## theta2 -2.117 -1.646 -1.278 -1.031 -0.3454
```

	2.5%	25%	50%	75%	97.5%	Mean	SD
theta1	-2.073	-1.634	-1.261	-1.061	-0.3456	-1.302	0.4080
theta2	-2.117	-1.646	-1.278	-1.031	-0.3454	-1.292	0.4189

2

(a)

우선 깁스 표본의 개수 m = 5000으로 정한다. 알고리즘의 가동을 확인하기 위해 mu, sig, A에 구체적인 숫자 4, 2, 7을 넣어 확인한다. (b)에서 해당 알고리즘을 표준정규분포에 적용할 것이다.

(단계 1)

```
m = 5000 # 깁스 표본의 수
mu = 4 # 정규분포의 모평균
sig = 2 # 정규분포의 모표준편차
A = 7 # 정규분포의 절단 기준값
po.theta = NULL # 사후표본을 담을 컨테이너
theta_0 = max(mu, A + 0.5*sig) # 초깃값
po.theta <- c(po.theta, theta_0)
```

(단계 2)

```
set.seed(42)
for (i in 1:m) {
  z <- runif(1, min = 0, max = exp((po.theta[i]-mu)^2/(-2*sig^2)))
  t <- runif(1, max(A, mu - sqrt(-2*(sig^2)*log(z))), mu + sqrt(-2*(sig^2)*log(z)))
  po.theta <- c(po.theta, t)
}
```

(단계 3)

```
head(po.theta)
```

```
## [1] 8.000000 8.019607 8.756739 8.098489 7.186747 7.476915
```

po.theta는 깁스 샘플링으로 생성된 마르코프 체인으로, 절단된 정규분포를 근사한다.

(b)

우선 문제를 풀기 위해 (a)의 알고리즘을 함수로 묶자. $\mu = 0$, $\sigma = 1$을 고정하고, 표본 추출 수 k와 절단 위치 a를 인자로 받아 표본 추출 결과 데이터프레임을 return하는 함수를 만들면 된다.

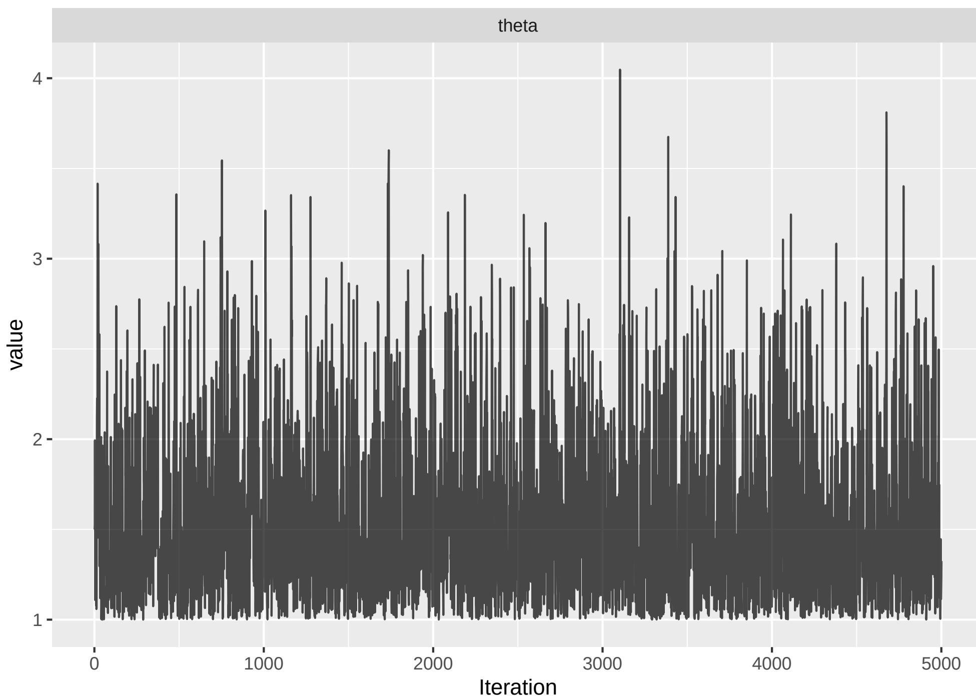
```
gibbs_truncated_normal <- function(k, a){
  m = k # 깁스 표본의 수
  mu = 0
  sig = 1
  A = a
  po.theta = NULL # 사후표본을 담을 컨테이너
  theta_0 = max(mu, A + 0.5*sig) # 초깃값
  po.theta <- c(po.theta, theta_0)

  set.seed(42)
  for (i in 1:m) {
    z <- runif(1, min = 0, max = exp((po.theta[i]-mu)^2/(-2*sig^2)))
    t <- runif(1, max(A, mu - sqrt(-2*(sig^2)*log(z))), mu + sqrt(-2*(sig^2)*log(z)))
    po.theta <- c(po.theta, t)
  }
  df_gi <- data.frame(theta = po.theta)
  return(df_gi)
}
```

A = 1

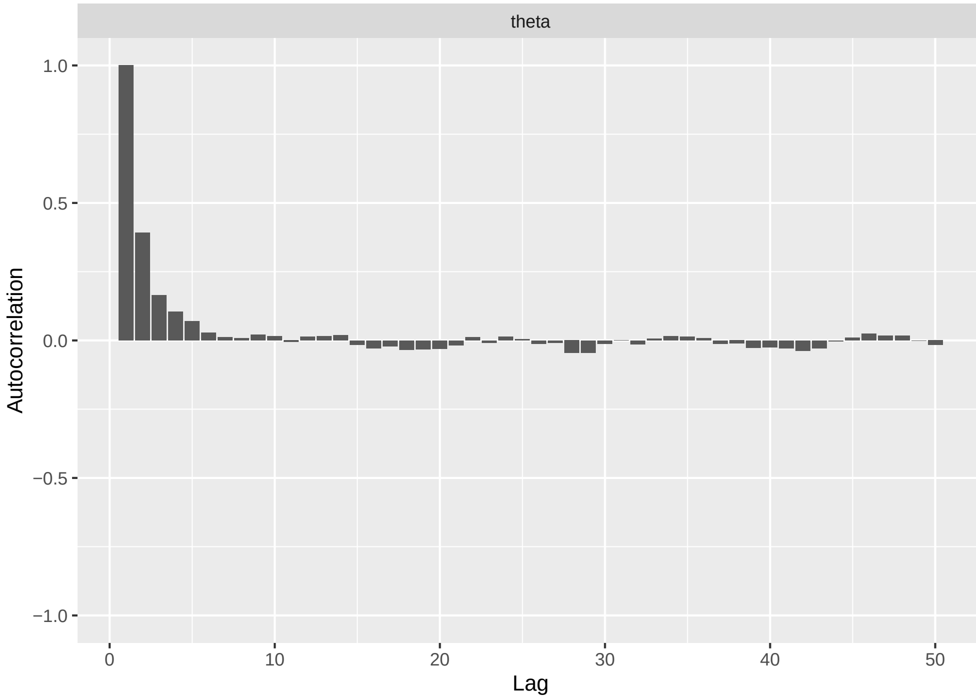
```
post_201_5000 <- gibbs_truncated_normal(5000, 1)
```

(m = 5000)

```
post_201_5000 %>% mcmc %>% ggs %>% ggs_traceplot()
```




```
post_201_5000 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```



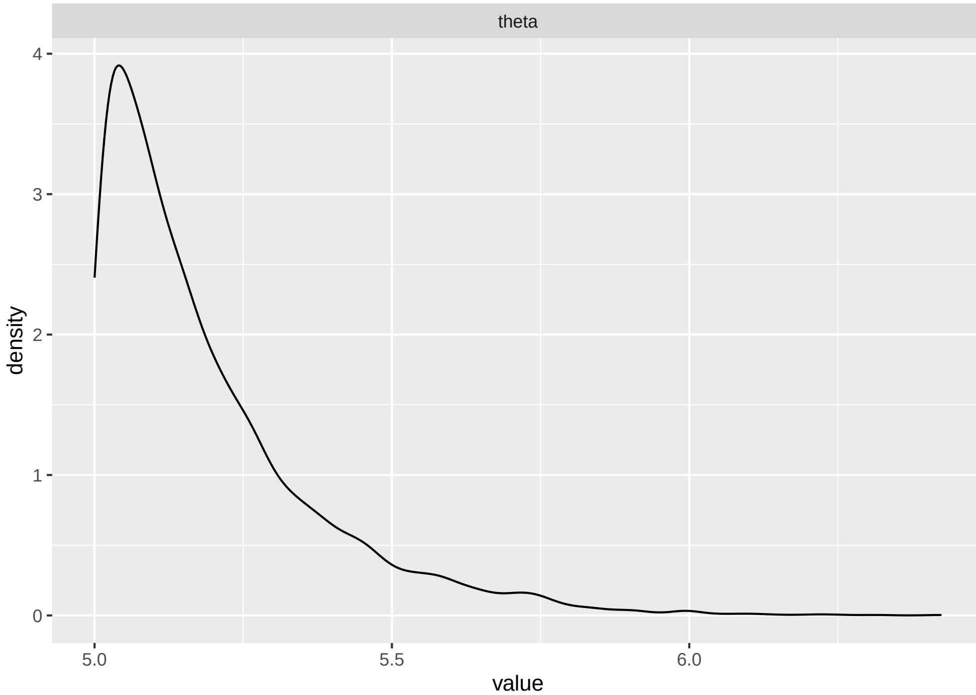
m = 5000에서도 충분히 수렴했다고 판단할 수 있다.

A = 5

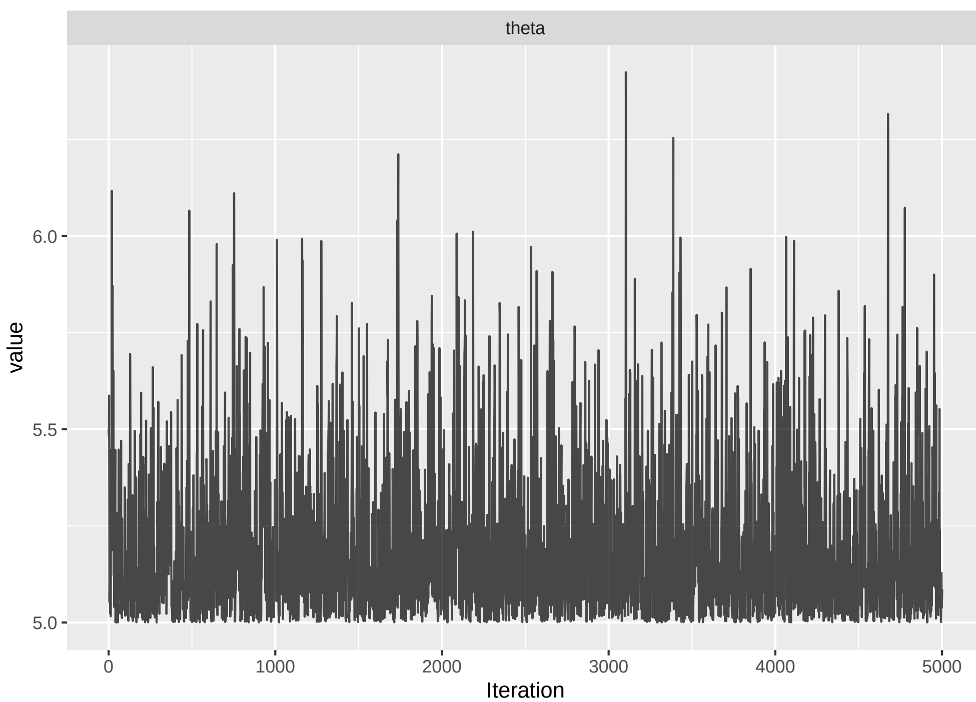
```
post_205_5000 <- gibbs_truncated_normal(5000, 5)
```

(m = 5000)

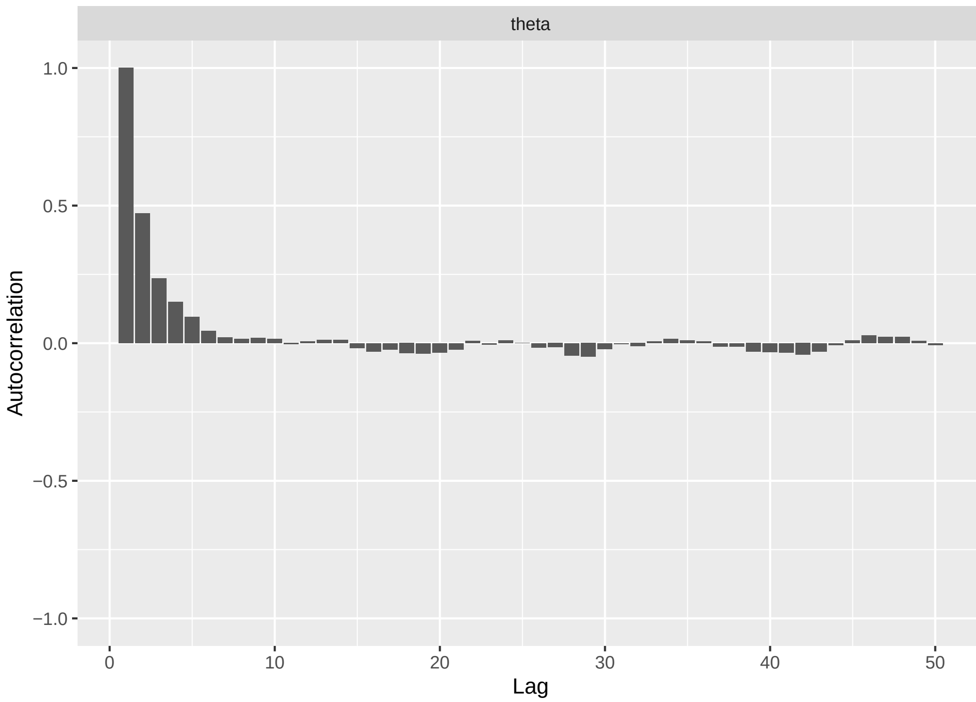
```
post_205_5000 %>% mcmc %>% ggs %>% ggs_density()
```




```
post_205_5000 %>% mcmc %>% ggs %>% ggs_traceplot()
```




```
post_205_5000 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```



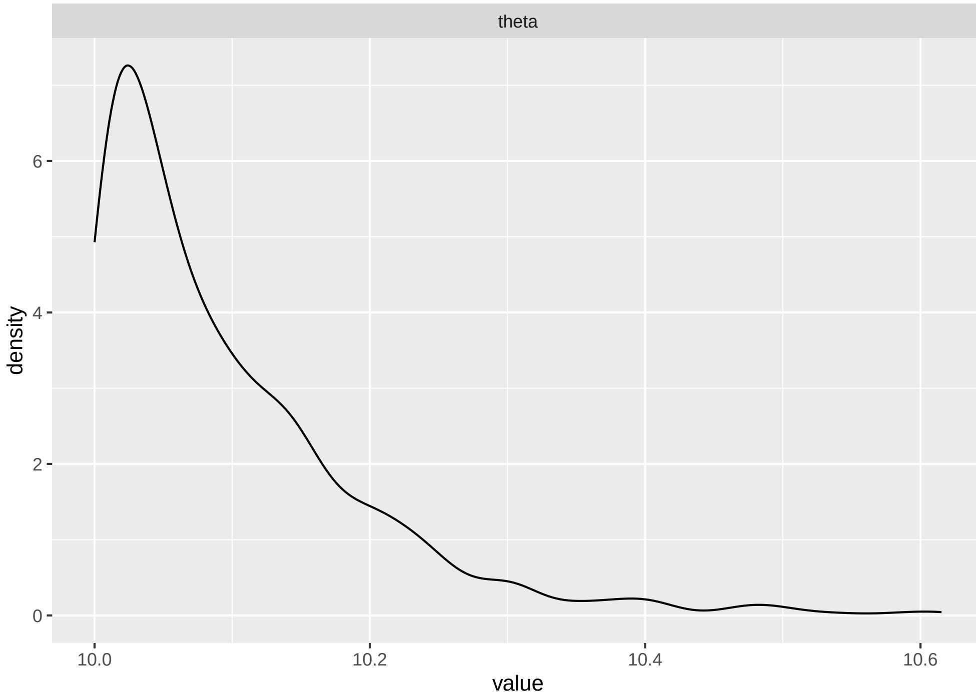
m = 5000에서 충분히 수렴했다고 판단할 수 있다.

A = 10

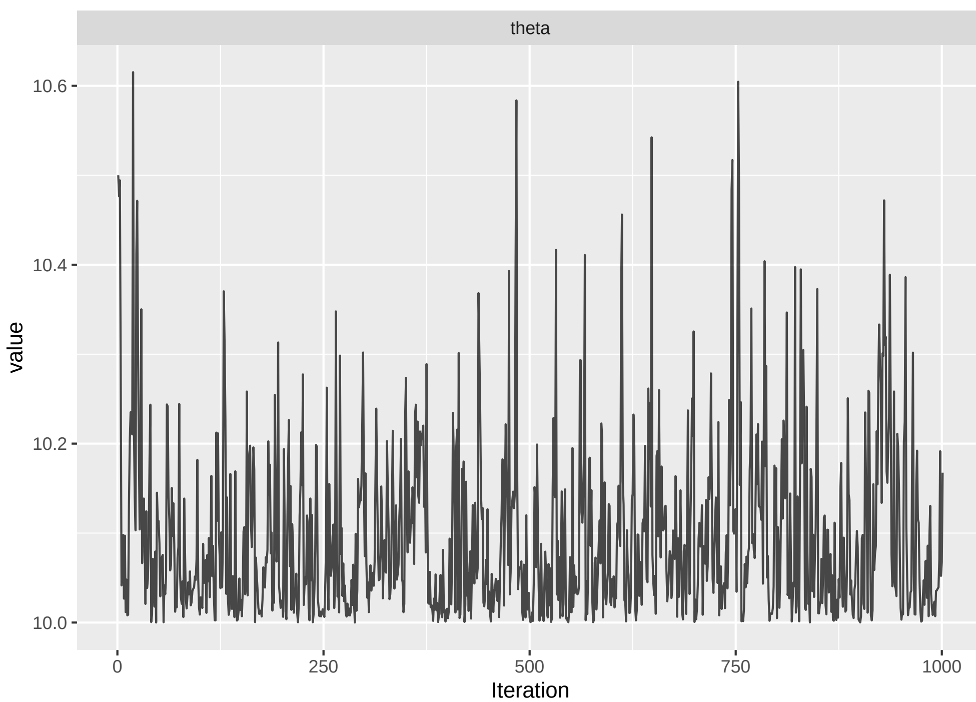
```
post_210_5000 <- gibbs_truncated_normal(5000, 10)
post_210_1000 <- gibbs_truncated_normal(1000, 10)
```

(m = 1000)

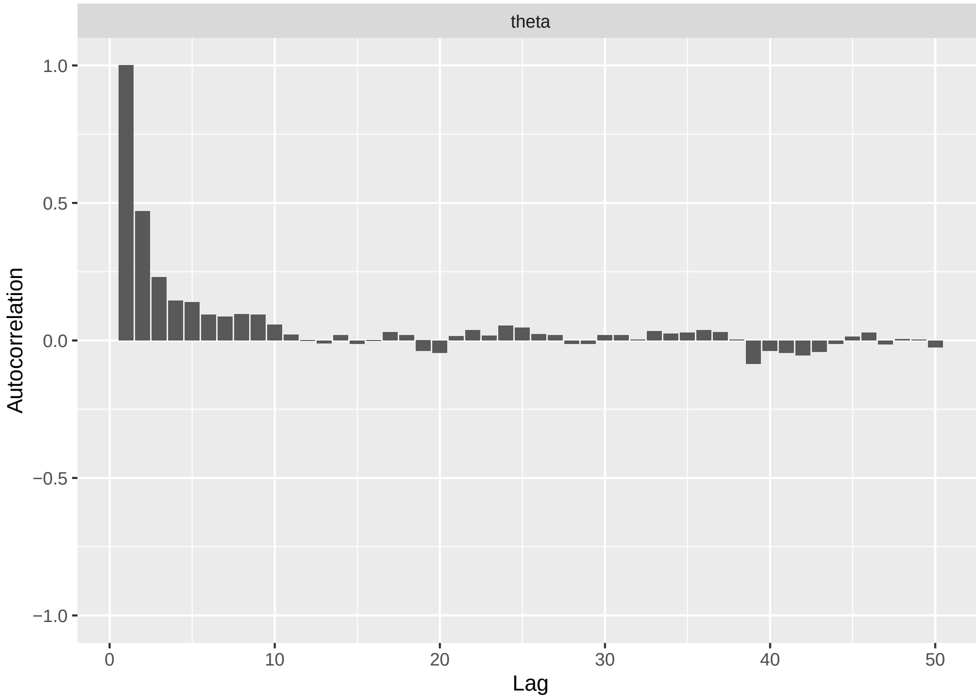
```
post_210_1000 %>% mcmc %>% ggs %>% ggs_density()
```




```
post_210_1000 %>% mcmc %>% ggs %>% ggs_traceplot()
```

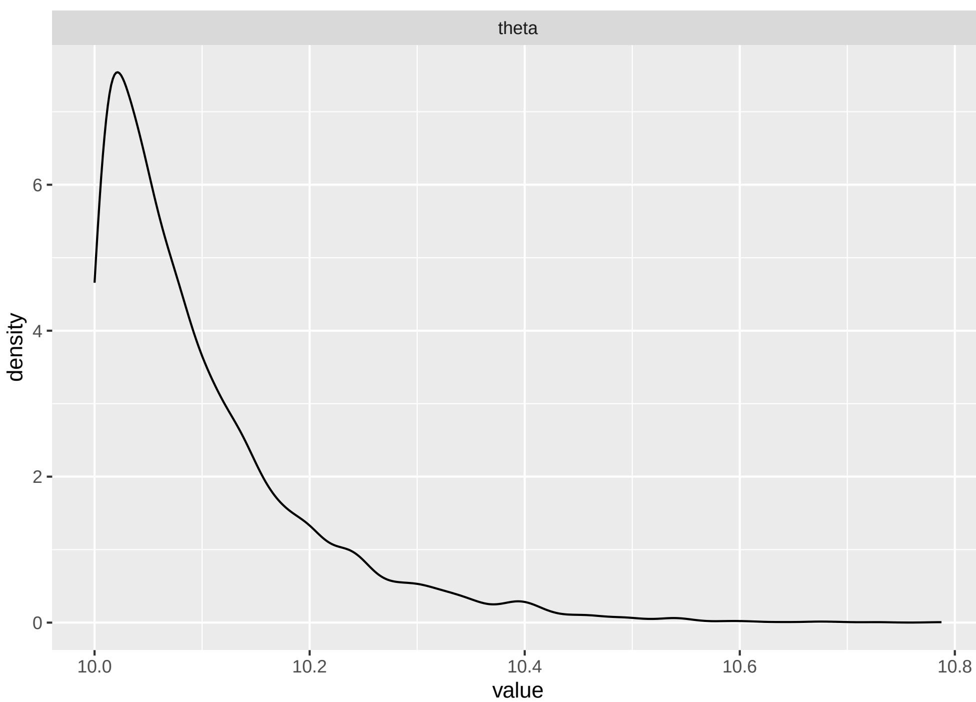



```
post_210_1000 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```

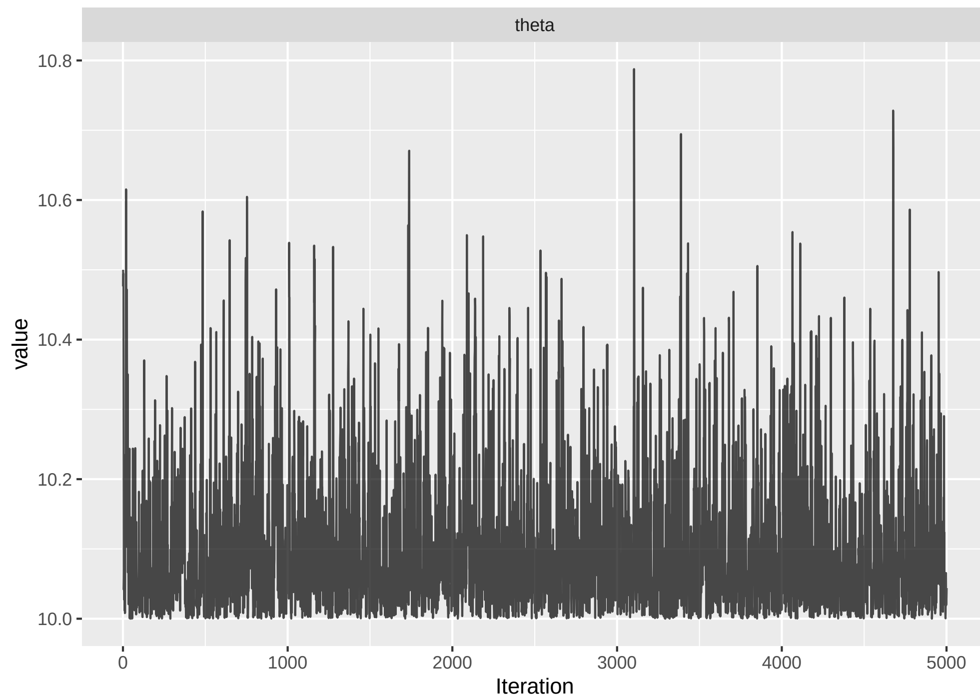


(m = 5000)

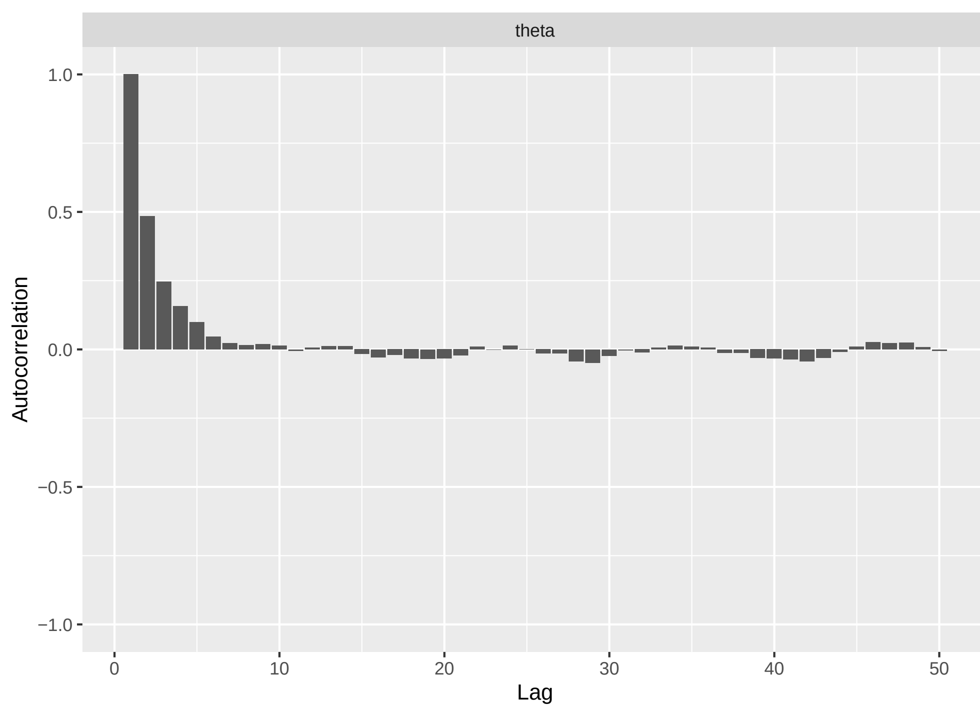
```
post_210_5000 %>% mcmc %>% ggs %>% ggs_density()
```



```
post_210_5000 %>% mcmc %>% ggs %>% ggs_traceplot()
```




```
post_210_5000 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```



m = 5000에서 수렴했다고 충분히 판단할 수 있다. m = 1000인 경우를 보면 수렴하지 않은 경우 '경향성'이 있음을 확연히 알 수 있다.

(c)

A = 1

```
post_201_5000 %>% mcmc %>% summary
```

```
##
## Iterations = 1:5001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean             SD       Naive SE Time-series SE
##       1.52968       0.45398        0.00642        0.01026
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75% 97.5%
## 1.016 1.178 1.410 1.757 2.704
```

	2.5%	25%	50%	75%	97.5%	Mean	SD
theta	1.016	1.178	1.410	1.757	2.704	1.52968	0.45398

A = 5

```
post_205_5000 %>% mcmc %>% summary
```

```
##
## Iterations = 1:5001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean             SD       Naive SE Time-series SE
##      5.188972       0.184461       0.002608       0.004613
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75% 97.5%
## 5.004 5.056 5.133 5.261 5.697
```

	2.5%	25%	50%	75%	97.5%	Mean	SD
theta	5.004	5.056	5.133	5.261	5.697	5.188972	0.184461

A = 10

```
post_210_5000 %>% mcmc %>% summary
```

```
##
## Iterations = 1:5001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean           SD       Naive SE Time-series SE
##     10.099615     0.099508      0.001407       0.002524
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75% 97.5%
## 10.00 10.03 10.07 10.14 10.38
```

	2.5%	25%	50%	75%	97.5%	Mean	SD
theta	10.00	10.03	10.07	10.14	10.38	10.099615	0.099508

3

(a)

해당 문제는 종이에 풀이하였다.

(b)

해당 문제는 종이에 풀이하였다.

(c)

초깃값

```
df <- c(68.3, 85.7, 73.8, 83.2, 58.9, 72.7, 70.5, 58.7, 74.1, 75.0) #가능도
m <- 5000
nu <- 20 #prior의 정보 1
theta0 <- 0 #prior의 정보 2
nu0 <- 1 #prior의 정보 3
s02 <- 1 #prior의 정보 4
theta_0 <- 0
```

```
delta_0 <- 1
ksi_0 <- 0
po.theta = NULL
po.delta = NULL
po.ksi = NULL
po.theta <- c(po.theta, theta_0)
po.delta <- c(po.delta, delta_0)
po.ksi <- c(po.ksi, ksi_0)
```

깁스 샘플링 반복

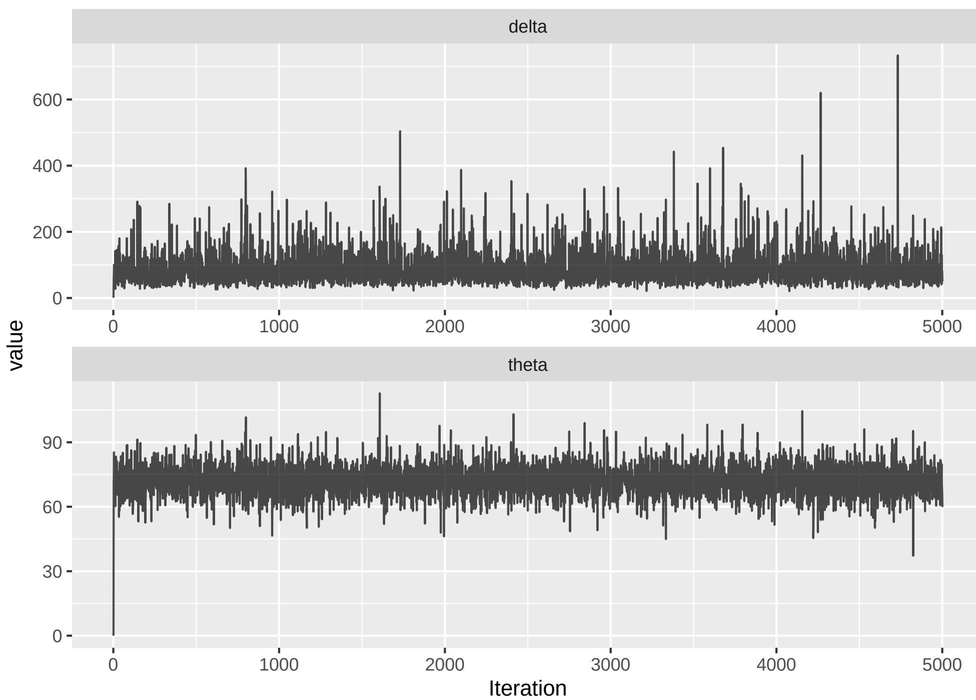
```
set.seed(42)
for (i in 1:m) {
 ksi_prime <- rgamma(1, (1+nu)/2, rate = 1 + ((po.theta[i]-theta0)^2) / (nu*po.delta[i]))
 theta_prime <- rnorm(1, (nu*mean(df) + 2*theta0*ksi_prime) / (nu+2*ksi_prime),
 ↪   sqrt((nu*po.delta[i]) / (nu+2*ksi_prime)))
 delta_rate = (((nu + 2*ksi_prime)*((theta_prime - (nu*mean(df) + 2*ksi_prime*theta0) / (nu +
 ↪   2*ksi_prime))^2))/(2*nu))+(1/2)*(nu0*s02+(length(df)-1)*var(df)+(2*ksi_prime*(mean(df)-
 ↪   theta0)^2)/(nu+2*ksi_prime))
 delta_prime <- rinvgamma(1, (nu0 + length(df) + 1)/2, rate = delta_rate)
 po.theta <- c(po.theta, theta_prime)
 po.delta <- c(po.delta, delta_prime)
 po.ksi <- c(po.ksi, ksi_prime)
}
```

(d)

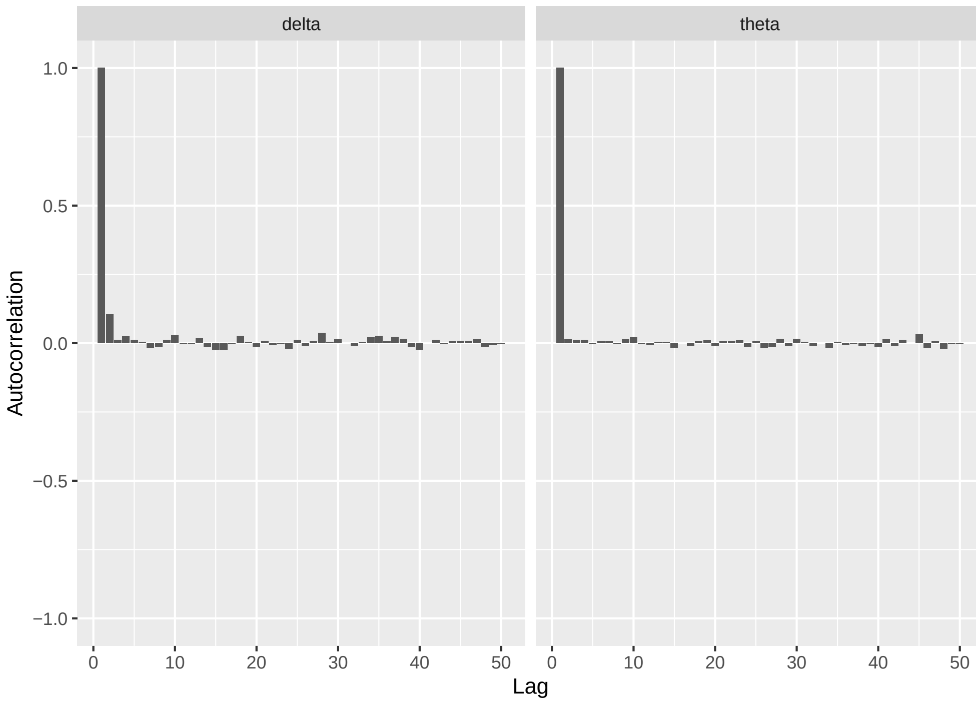
```
#가능도
df <- c(68.3, 85.7, 73.8, 83.2, 58.9, 72.7, 70.5, 58.7, 74.1, 75.0)
m <- 5000

nu <- length(df) #prior의 정보 1
theta0 <- mean(df) #prior의 정보 2
nu0 <- 1 #prior의 정보 3
s02 <- var(df) #prior의 정보 4
theta_0 <- 0
delta_0 <- 1
ksi_0 <- 0
po.theta = NULL
po.delta = NULL
po.ksi = NULL
po.theta <- c(po.theta, theta_0)
po.delta <- c(po.delta, delta_0)
po.ksi <- c(po.ksi, ksi_0)

set.seed(42)
for (i in 1:m) {
 ksi_prime <- rgamma(1, (1+nu)/2, rate = 1 + ((po.theta[i]-theta0)^2) / (nu*po.delta[i]))
```

```
  theta_prime <- rnorm(1, (nu*mean(df) + 2*theta0*ksi_prime) / (nu+2*ksi_prime),
↪     sqrt((nu*po.delta[i]) / (nu+2*ksi_prime)))
  delta_rate = (((nu + 2*ksi_prime)*((theta_prime - (nu*mean(df) + 2*ksi_prime*theta0) / (nu +
↪     2*ksi_prime))^2))/(2*nu))+(1/2)*(nu0*s02+(length(df)-1)*var(df)+(2*ksi_prime*(mean(df)-
↪     theta0)^2)/(nu+2*ksi_prime))
  delta_prime <- rinvgamma(1, (nu0 + length(df) + 1)/2, rate = delta_rate)
  po.theta <- c(po.theta, theta_prime)
  po.delta <- c(po.delta, delta_prime)
  po.ksi <- c(po.ksi, ksi_prime)
}

post_3 <- data.frame(theta = po.theta, delta = po.delta)
```

```
post_3 %>% mcmc %>% ggs %>% ggs_traceplot()
```




```
post_3 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```

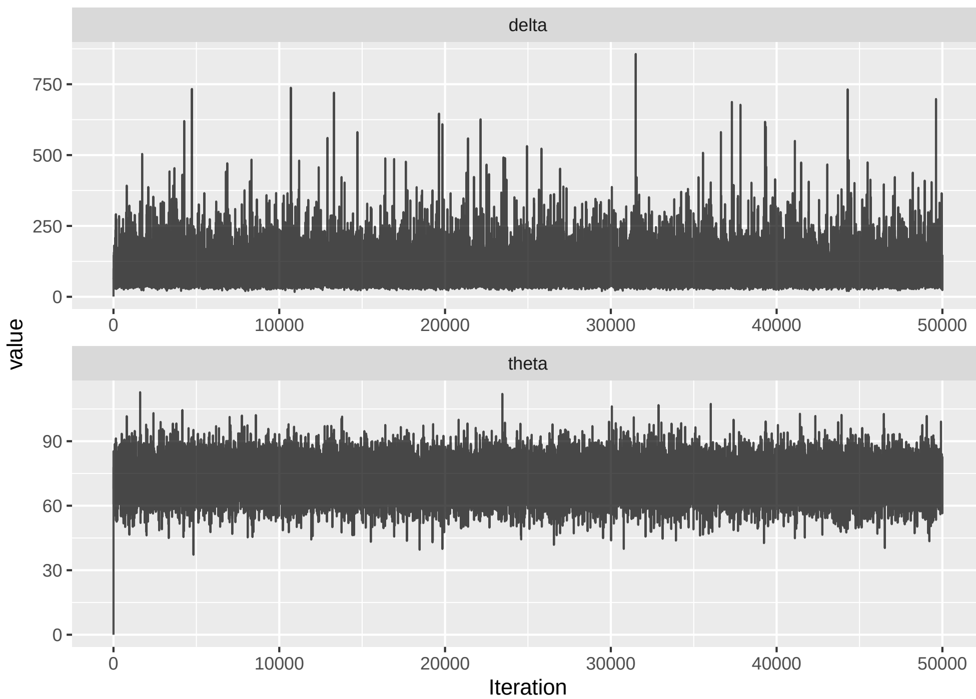


위와 같이 사후표본을 구한 결과, 마르코프 체인이 수렴했다고 보기에는 약간 미묘한 결과를 얻었다. prior 정보를 위와 같이 조정한 것은, 사전 정보를 정확히 조정할 근거가 없어 자료의 정보를 사용하되 nu0를 줄여 그 정보가 미치는 영향을 최대한 줄인 것이다. 충분히 수렴할 때까지 표본을 늘려 보았다.

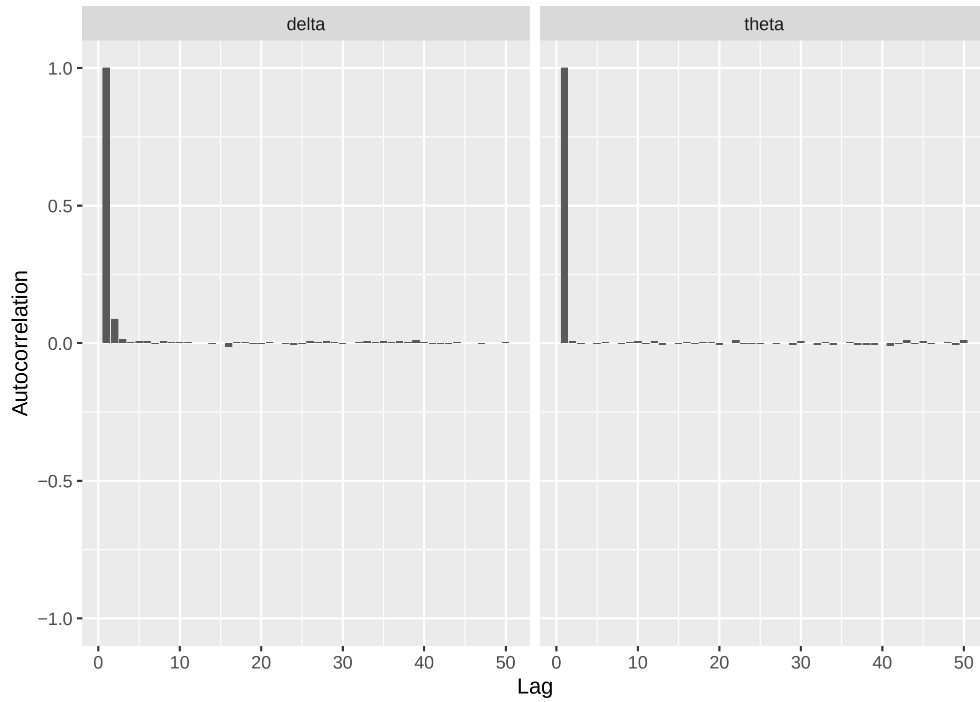
```
#가능도
df <- c(68.3, 85.7, 73.8, 83.2, 58.9, 72.7, 70.5, 58.7, 74.1, 75.0)
m <- 50000

nu <- length(df) #prior의 정보 1
theta0 <- mean(df) #prior의 정보 2
nu0 <- 1 #prior의 정보 3
s02 <- var(df) #prior의 정보 4
theta_0 <- 0
delta_0 <- 1
ksi_0 <- 0
po.theta = NULL
po.delta = NULL
po.ksi = NULL
po.theta <- c(po.theta, theta_0)
po.delta <- c(po.delta, delta_0)
po.ksi <- c(po.ksi, ksi_0)

set.seed(42)
for (i in 1:m) {
 ksi_prime <- rgamma(1, (1+nu)/2, rate = 1 + ((po.theta[i]-theta0)^2) / (nu*po.delta[i]))
 theta_prime <- rnorm(1, (nu*mean(df) + 2*theta0*ksi_prime) / (nu+2*ksi_prime),
 ↪   sqrt((nu*po.delta[i]) / (nu+2*ksi_prime)))
 delta_rate = (((nu + 2*ksi_prime)*((theta_prime - (nu*mean(df) + 2*ksi_prime*theta0) / (nu +
 ↪   2*ksi_prime))^2))/(2*nu))+(1/2)*(nu0*s02+(length(df)-1)*var(df)+(2*ksi_prime*(mean(df)-
 ↪   theta0)^2)/(nu+2*ksi_prime))
```

```
  delta_prime <- rinvgamma(1, (nu0 + length(df) + 1)/2, rate = delta_rate)
  po.theta <- c(po.theta, theta_prime)
  po.delta <- c(po.delta, delta_prime)
  po.ksi <- c(po.ksi, ksi_prime)
}

post_4 <- data.frame(theta = po.theta, delta = po.delta)
```

```
post_4 %>% mcmc %>% ggs %>% ggs_traceplot()
```



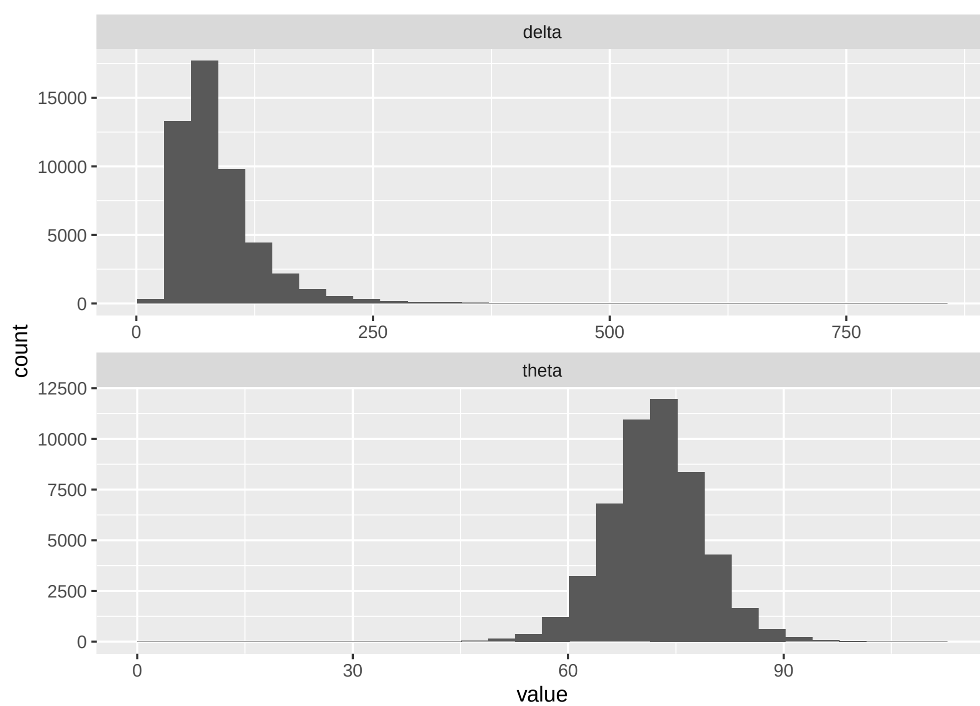

```
post_4 %>% mcmc %>% ggs %>% ggs_autocorrelation()
```



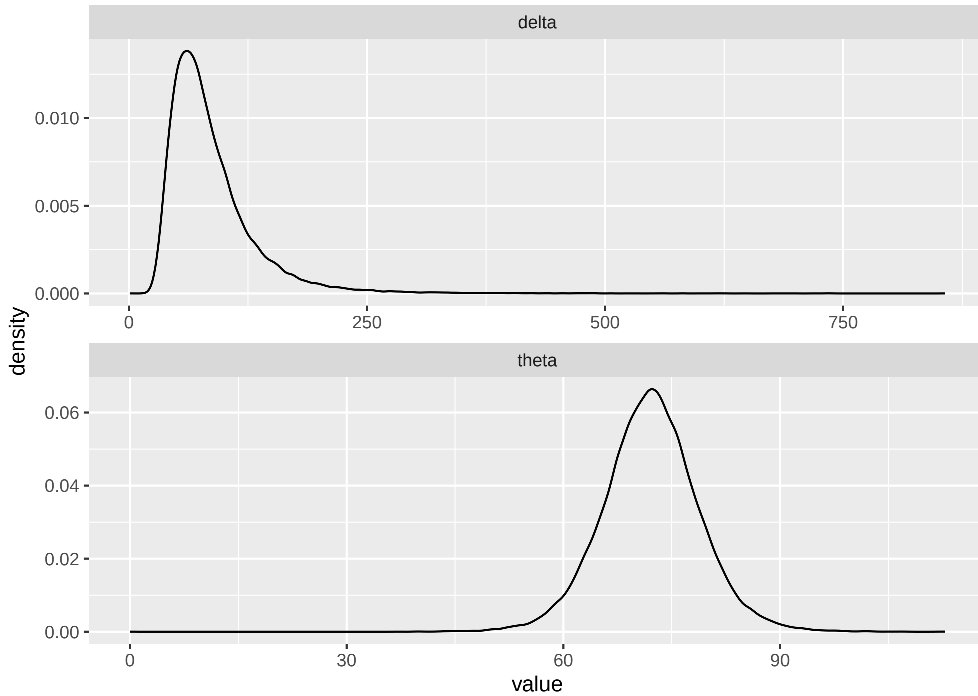
m = 50000 수준에서는 충분히 수렴했다고 판단할 수 있다. 이를 사용한다.

(e)

```r
post_4 %>% mcmc %>% ggs %>% ggs_histogram()
```



```r
post_4 %>% mcmc %>% ggs %>% ggs_density()
```



사후표본의 히스토그램과 밀도함수 그림은 다음과 같다.

(f)

```r
post_4 %>% mcmc %>% summary
```

```
##
## Iterations = 1:50001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 50001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##        Mean     SD Naive SE Time-series SE
## theta 72.10  6.651  0.02974        0.02995
## delta 86.13 46.165  0.20646        0.22564
##
## 2. Quantiles for each variable:
##
##        2.5%   25%   50%    75%  97.5%
## theta 58.90 67.94 72.08  76.21  85.58
## delta 35.27 56.43 74.94 102.13 203.69
```

theta의 사후평균 : 72.10 theta의 사후표준편차 : 6.651 theta의 95% CI : (58.90, 85.58)
sigma의 사후평균 : 9.280625 sigma의 사후표준편차 : 6.794483 sigma의 95% CI : (5.938855, 14.272)