베이즈데이터분석

Name(Student ID)

2023-11-05

베이지안 추론의 핵심은 관측값이 주어졌을 때 모수 θ의 사후분포를 구하는 것이다. 그러나 모형이 복잡하거나 모수의 수가 많으면 θ를 수리적으로 구할 수 없다. 따라서 사후분포의 사후평균, 사후분산, 특정 사건에 대한 사후확률 등을 근사적으로 계산할 필요가 있다. 이때 사후분포의 특성을 근사적으로 구하기 위해 마르코프 체인 몬테칼로(Markov Chain Monte Carlo, MCMC) 기법이 많이 사용된다. MCMC 기법은 마르코프체인을 이용하여 사후분포로부터 표본을 생성하고 이 사후표본을 사용하여 사후추론을 수행하는 방법이다. 깁스 추출법, 메트로폴리스-헤이스팅스 알고리듬, 해밀턴 몬테 카를로 등이 대표적인 MCMC 기법이다.

단순한 모델의 경우 R의 기본적인 함수(lm, glm 등)를 사용하여 매개변수를 추정할 수 있고, 복잡한 모델의 경우에도 기존 R 패키지를 사용하면 문제를 해결할 수 있는 경우도 있다. 그러나 패키지와 함수별로 사용 방법이 달라서 이를 충분히 인지해야 하고, 패지지와 함수 중에서 적절한 모델을 찾는 노력도 중요하다. 특히 적절한 모델 지원이 되지 않는 경우에는 분석 자체를 진행할 수 없게 된다. 이처럼 R 패키지가 모델 확장 성이 낮다는 단점에 대응하기 위해 등장한 것이 Stan, WinBUGS, JAGS 등의 확률적 프로그래밍 언어라고 할 수 있다.

Stan은 앤드류 겔만, 밥 카펜터, 대니얼 리 등이 2012년부터 깃허브에서 개발하고 있는 확률적 프로그래밍언어이다. WinBUGS나 JAGS처럼 사후분포에서 표본을 추출한다. R인터페이스인 rstan과 함께 python과 matlab 인터페이스도 공개되어 있다. Stan은 추정 계산 알고리즘으로 해밀토니안 몬테칼로(HMC)의 한 버전인 NUTS(No-U-Turn Sampler)를 사용한다. NUTS는 매개변수의 수가 많아도 효과적으로 표본을 추출한다. Stan은 WinBUGS나 JAGS와 달리 복잡한 모델에서도 상당히 정상적으로 표본을 추출할 수 있다. Stan에서는 추정계산에 변분 베이즈법의 한 버전인 자동미분 변분 추정(ADVI)을 사용할 수도 있다.

Rstan은 Stan의 R용 패키지이다. Stan 코드로 작성한 모델을 R에서 간단히 실행할 수 있다. stan 코드는 data 블록, parameters 블록, model 블록 등으로 작성한다. 가능도에 사용되는 변수 중 데이터 변수는 data 블록에, 모수는 parameters 블록에 기술한다. model 블록에는 가능도와 사전분포를 기술한다. 모델식이 실행되면 c++코드로 변환되어 컴파일 된 후 MCMC 표본을 추출한다. 변수 선언 전에 해당 변수을 사용하면 계산이 오류가 생기므로 주의해야 한다.

1. (10점) 밀도함수 $f(x) = \frac{1}{C}x^2 sin(x), x \in (0, \pi)$ 를 고려하자.
여기서 상수 $C = \int_0^\pi x^2 sin(x) dx$ 이다.
다음의 질문에 답하시오.

(a) $U \sim U(0, \pi)$일 때, 상수 C를 확률변수 U의 기댓값으로 표현하시오.

확률변수 x의 확률밀도함수가 $f(x)$일 때, $g(x)$의 기댓값은
다음과 같이 확률변수의 확률밀도함수를 가중치로 하는 적분과 동일하다.

$$Eg(x) = \begin{cases} \int g(x)f(x)dx, & x\text{가 연속형일 때,} \\ \sum g(x)f(x), & x\text{가 이산형일 때} \end{cases}$$

$g(x) = x^2 sin(x)$라고 하면, $U \sim U(0, \pi)$일 때 U의 확률밀도함수는 $\frac{1}{\pi}$이므로 $g(u)$의 기댓값은 다음과 같다.

$$Eg(u) = \int_0^\pi u^2 sin(u) \frac{1}{\pi} du$$

위 식은 다음처럼 정리할 수 있다.

$$\int_0^\pi u^2 sin(u) du = \pi Eg(u)$$

$$C = \int_0^\pi u^2 sin(u) du$$

이므로

$$C = \pi Eg(u)$$

이다.

(b) 상수 C를 (a)에서 표현한 식을 이용해서 몬테 카를로 방법으로 구하시오.
이 때, 몬테 카를로 표본의 개수는 $m = 1000$를 이용하시오. 계산은 R을 이용하시오.

```
# 시드값 설정
set.seed(123)

m <- 1000
samples <- runif(m, 0, pi)  # 표본 추출
values <- samples^2 * sin(samples)
C_estimate <- pi * mean(values)
C_estimate
```

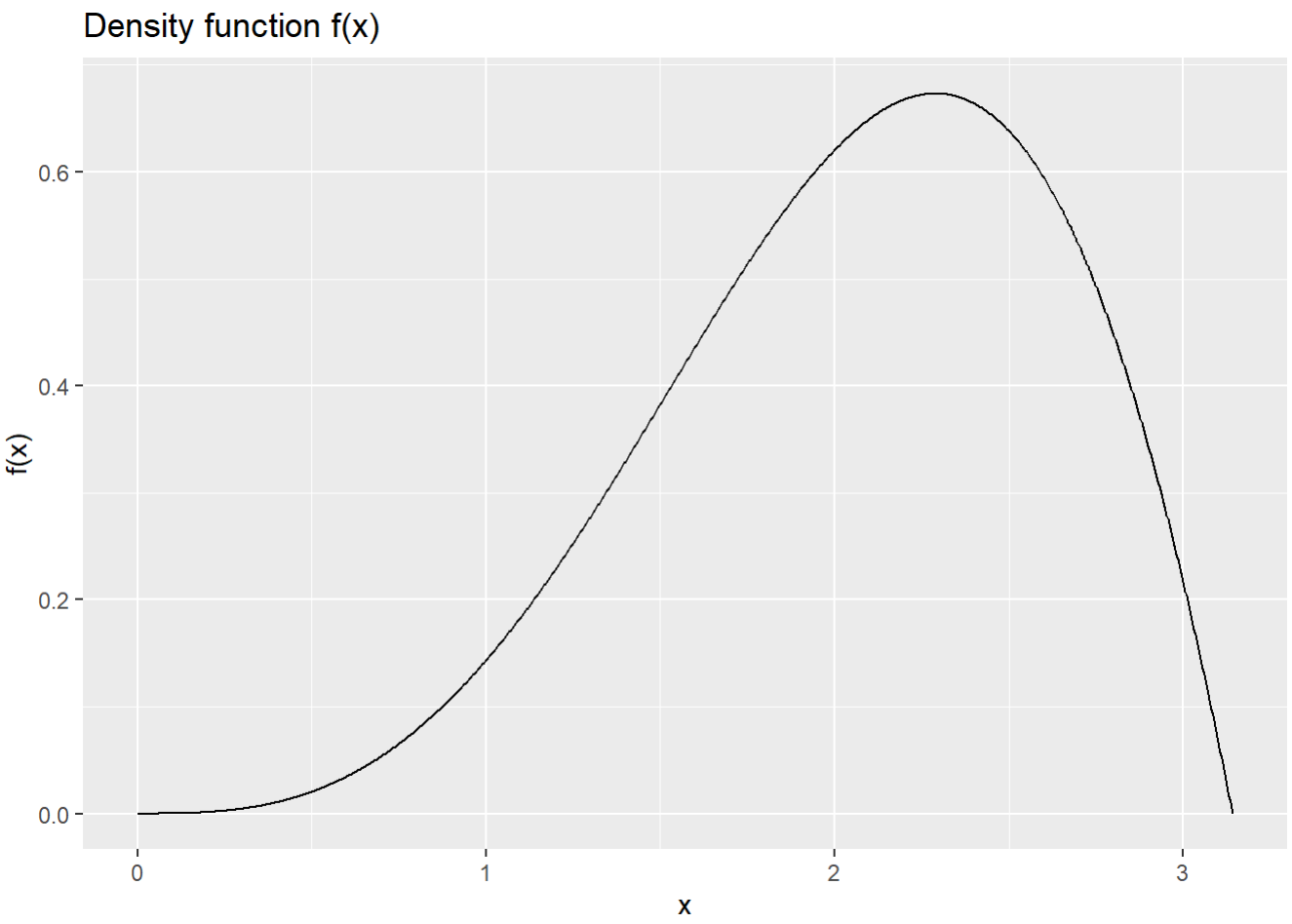
```
## [1] 5.859883
```

위 추정값은 주어진 범위에서 $x^2 sin(x)$의 실제 적분값 $\pi^2 - 4 (\approx 5.8696)$에 근접한 값임을 알 수 있다.

(c) (b)에서 구한 상수 C값을 이용하여 밀도함수 $f(x)$의 그림을 R로 그리시오.

```
library(ggplot2)

# 밀도함수 정의
```

```r
f <- function(x) (1/C_estimate) * x^2 * sin(x)

# 밀도함수 그래프
# 0에서 pi까지의 범위에서 일정 간격으로 나누어진 숫자 1000개의 시퀀스 생성
x_vals <- seq(0, pi, length.out = 1000)
# sapply는 각 x 값에 대해 밀도함수를 계산
y_vals <- sapply(x_vals, f)

df <- data.frame(x = x_vals, y = y_vals)

ggplot(df, aes(x = x, y = y)) +
  geom_line() +
  ggtitle("Density function f(x)") +
  xlab("x") +
  ylab("f(x)")
```



2. (20점) 다음은 1994년과 2014년 군에 입대하는 10명의 병사들의 몸무게를 잰 결과이다.
1994년: 65.9, 55.9, 43.8, 57.7, 68.8, 23.1, 85.4, 62.8, 65.2, 49.9 (kg)
2014년: 68.3, 85.7, 73.8, 83.2, 58.9, 7.27, 70.5, 58.7, 74.1, 75 (kg)

1994년 군에 입대한 병사들의 몸무게를 $x_i, i = 1, 2, \ldots, n(n = 10)$,
2014년 군에 입대한 병사들의 몸무게를 $y_i, i = 1, 2, \ldots, n$라 하고, 다음의 모형을 상정하자.

$$\theta_1, \theta_2 \sim U(R)$$

$$\sigma_1, \sigma_2 \sim g(\sigma) = \frac{1}{\sigma}, \sigma > 0$$

$$x_1, \ldots, x_n \sim N(\theta_1, \sigma_1^2)$$

$$y_1, \ldots, y_n \sim N(\theta_2, \sigma_2^2)$$

2014년 병사들의 몸무게의 평균과 1994년 병사들의 몸무게의 차이 $\sigma_1 - \sigma_2$에 대해 추론을 하고자 한다.
다음의 질문에 답하시오.

(a) 위 모형의 사후표본을 추출하기 위한 스탠과 R 코드를 작성하고 사후표본을 구하시오.
번인 5000개를 포함하여 총 15,000개의 사후표본을 추출하시오.

```
# install.packages("rstan")
# install.packages('rstudioapi')
# suppressPackageStartupMessages는패키지 로드 시 불필요한 메시지 숨김
suppressPackageStartupMessages(library(rstan))

# R에서 stan을 수행하기 전에 항상 먼저 수행
# 병렬 처리와 컴파일된 코드를 하드드라이브에 저장하기 위해 필요.
options(mc.cores = parallel::detectCores())
rstan_options(Auto_write=TRUE)
```

Stan 코드는 기본적으로 data, parameters, model 3개의 블록으로 구성된다.
data 블록에서는 관측 데이터를 선언하고,
parameters 블록에는 샘플링하고 싶은 모수를 선언한다.
model 블록에는 우도(likelihood)와 사전분포(prior distribution)를 기술한다.

```
# stan code

code = " data {
  int<lower=0> n;  // 병사의 수
```

```
  vector[n] x;     // 1994년 병사들의 몸무게
  vector[n] y;     // 2014년 병사들의 몸무게
}

parameters {
  real theta1;     // 1994년 병사들의 몸무게의 평균
  real theta2;     // 2014년 병사들의 몸무게의 평균
  real<lower=0> sigma1;  // 1994년 병사들의 몸무게의 표준편차
  real<lower=0> sigma2;  // 2014년 병사들의 몸무게의 표준편차
}

model {
  x ~ normal(theta1, sigma1);
  y ~ normal(theta2, sigma2);
  target += -log(sigma1);
  target += -log(sigma2);
 }
"
```

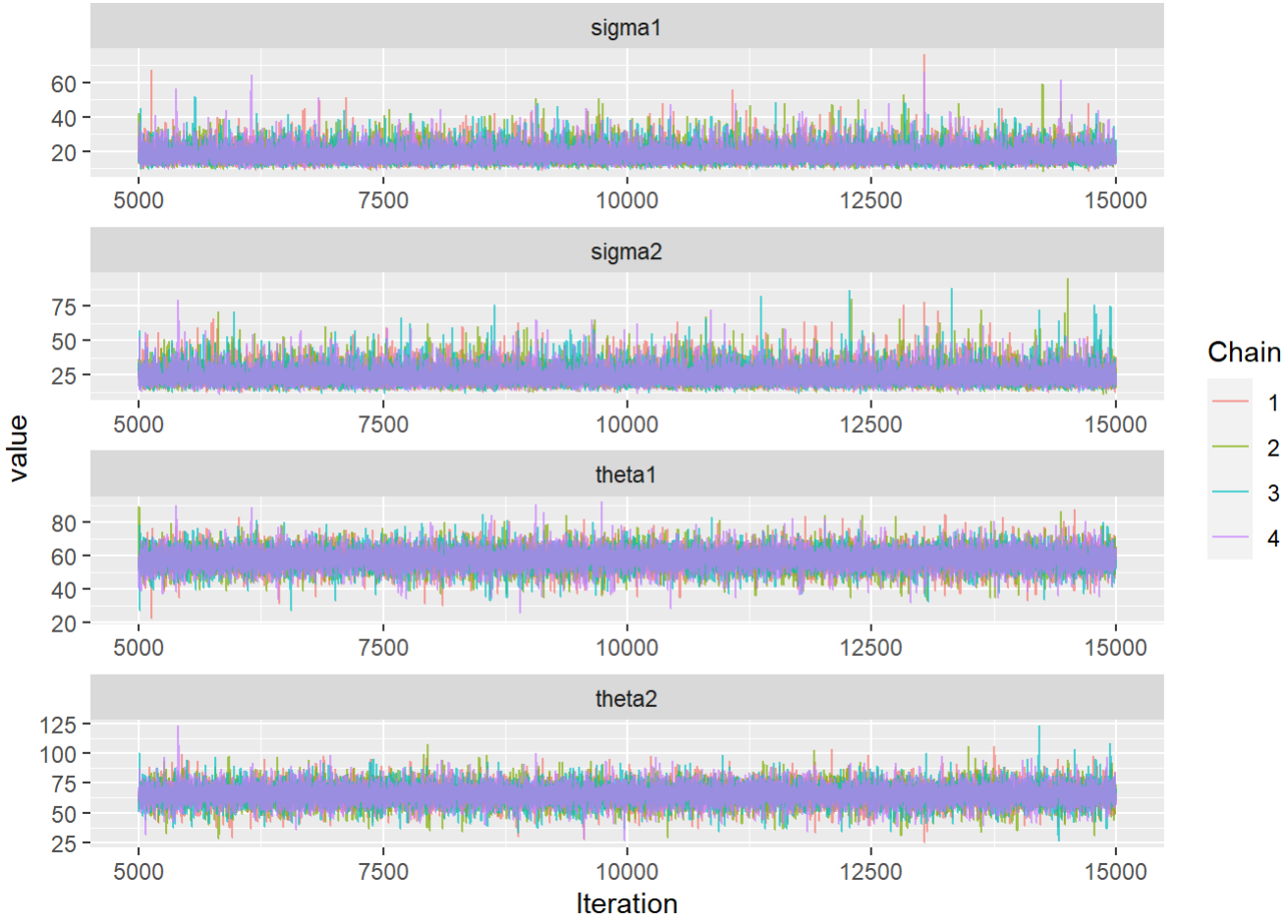
```
# 관측값
data_list <- list(
  n = 10,
  x = c(65.9, 55.9, 43.8, 57.7, 68.8, 23.1, 85.4, 62.8, 65.2, 49.9),
  y = c(68.3, 85.7, 73.8, 83.2, 58.9, 7.27, 70.5, 58.7, 74.1, 75)
)

# 스탠 모델 실행
fit <- stan(model_code=code, data = data_list, warmup = 5000, iter = 15000, seed=1234)

# 결과 출력
print(fit)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=15000; warmup=5000; thin=1;
## post-warmup draws per chain=10000, total post-warmup draws=40000.
##
##          mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## theta1  57.83    0.03 5.94  45.95  54.17  57.82  61.53  69.61 29022    1
## theta2  65.49    0.05 8.00  49.61  60.53  65.48  70.38  81.47 28023    1
## sigma1  18.19    0.03 4.88  11.47  14.85  17.29  20.51  30.30 26187    1
## sigma2  24.42    0.04 6.73  15.30  19.78  23.11  27.58  40.87 24754    1
```

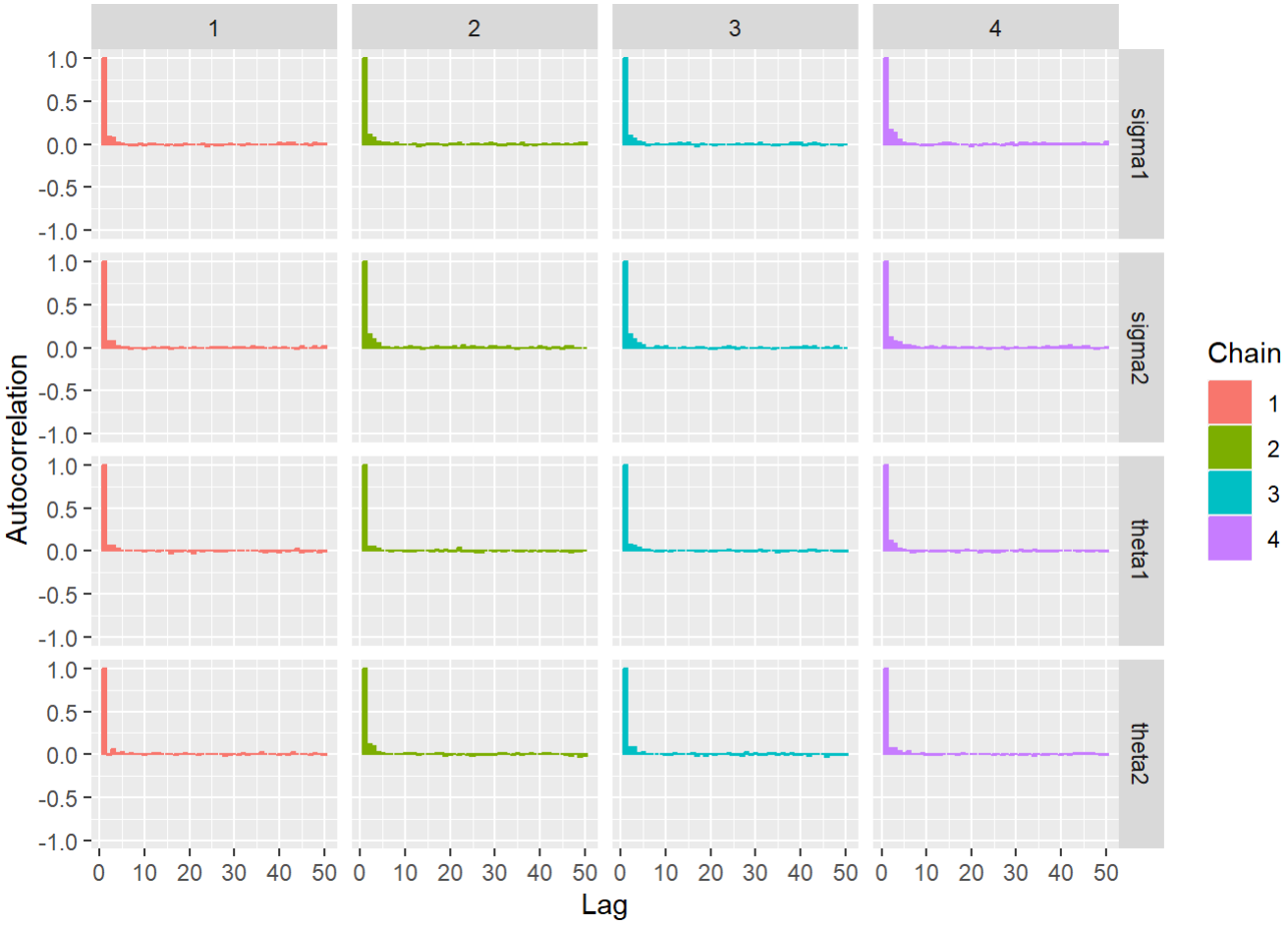
```
## lp__   -70.31    0.01 1.57 -74.30 -71.08 -69.96 -69.15 -68.37 14984    1
##
## Samples were drawn using NUTS(diag_e) at Sun Nov  5 10:09:51 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(b) 모수들의 시계열 그림, 자기상관계수 그림을 그리고 마르코프 체인이 수렴했는지 판단하시오. 수렴하지 않았다고 판단되면 수렴했다고 판단할 때까지 사후표본의 크기를 늘리시오.

```
# install.packages('ggmcmc')
suppressPackageStartupMessages(library('ggmcmc'))
# 시계열 그림
ggs_traceplot(ggs(fit))
```



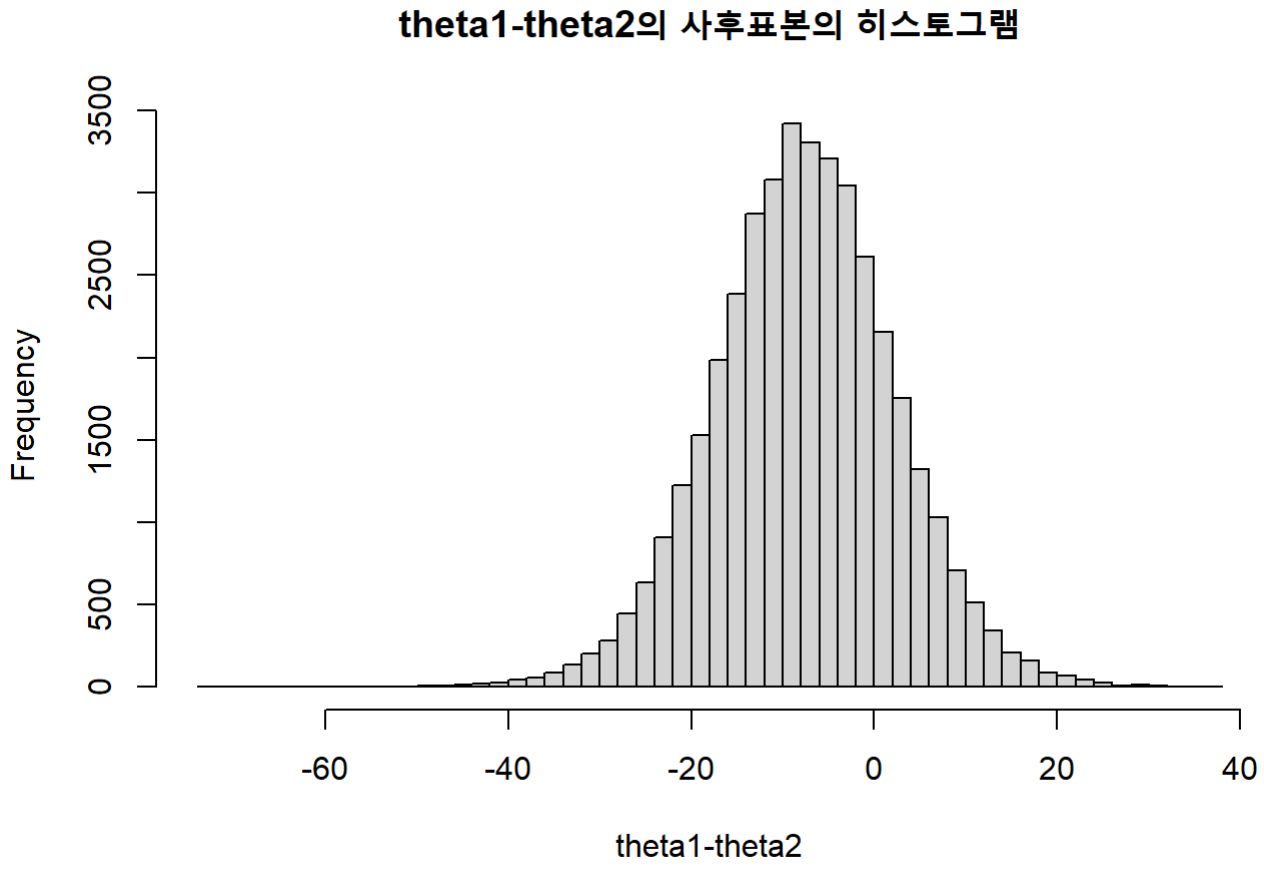
```
# 자기상관계수
ggs_autocorrelation(ggs(fit))
```



시계열 그림에 패턴이 없고 무질서해 보이므로 수렴이 잘된 것으로 판단할 수 있다.
또한 Lag 0에서 자기상관계수 1로 시작한 이후 급격하게 0에 가까워지므로 수렴이 충분히 잘 된 것으로 판단된다.
아울러 fit의 출력 결과에서 Rhat의 값이 1에 가까울수록, 해당 파라미터의 샘플링 체인들이 잘 수렴했다는 것을 나타낸다.

(c) $\theta_1 - \theta_2$의 사후표본의 히스토그램을 그리시오.

```
# extract 함수는 tidyr 등 다른 패키지에도 있으므로 네임스페이스 충돌 방지 위해 rstan 명시
# 사후표본 추출
samples <- rstan::extract(fit)
# 히스토그램
hist(samples$theta1 - samples$theta2, breaks=50, main="theta1-theta2의 사후표본의 히스토그램", xlab="theta1-theta2")
```



(d) $\theta_1 - \theta_2$의 사후평균, 사후표준편차, 95% 신용구간을 구하시오.

```
# 사후평균
posterior_mean <- mean(samples$theta1 - samples$theta2)
cat("사후평균: ", posterior_mean, "\n")
```

```
## 사후평균:  -7.657609
```

```
# 사후표준편차
posterior_sd <- sd(samples$theta1 - samples$theta2)
cat("사후표준편차: ", posterior_sd, "\n")
```

```
## 사후표준편차:  9.956946
```

```r
# 95% 신용구간
quantiles <- quantile(samples$theta1 - samples$theta2, probs=c(0.025, 0.975))
cat("95% 신용구간: [", quantiles[1], ", ", quantiles[2], "]\n")
```

```
## 95% 신용구간: [ -27.4459 ,  11.88293 ]
```

3. (20점) 다음은 R의 mtcars 데이터의 일부분이다. 두변수 mpg(마일/갤론)과 hp(마력)을 각각 x1와 x2로 나타내었다.

x1(mpg): 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
x2(hp): 110 110 93 110 175 105 245 62 95 123 123 180 180 180 205 215 230 66 52 65 97 150 150 245 175 66 91 113 264 175 335 109

두 변수 사이의 상관계수 ρ에 대해 추론하고자 다음의 모형을 고려하자.

$$x_i = (x_{i1}, x_{i2}) \sim N\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{pmatrix}\right), \quad i = 1, 2, \ldots, n$$

모수들의 사전분포는 다음과 같다고 하자.

$$\mu_1, \mu_2 \sim U(R)$$

$$\sigma_1, \sigma_2 \sim g(\sigma) = \frac{1}{\sigma}, \quad \sigma > 0$$

$$\rho \sim g(\rho) = (1-\rho^2)^{-\frac{1}{2}}, \quad -1 < \rho < 1$$

다음의 질문에 답하시오.

(a) 위 모형의 사후표본을 추출하기 위한 스탠과 R 코드를 작성하고 사후표본을 구하시오.
번인 5000개를 포함하여 총 15,000개의 사후표본을 추출하시오.

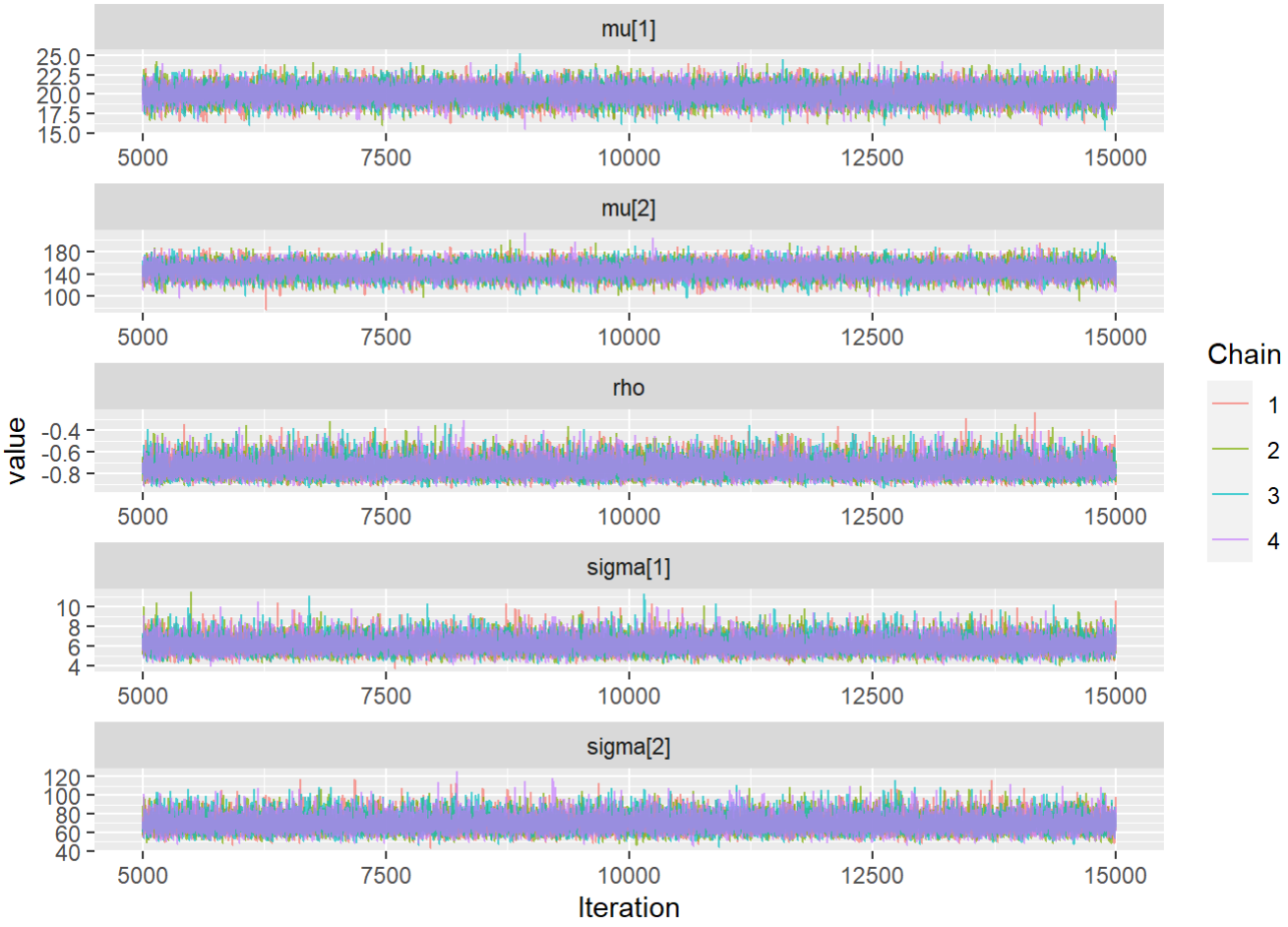
```r
code = "
        functions {
           // Stan에서 지원되지 않는 상관계수의 사전분포이므로 사용자 정의 사전분포 필요
           // 상관계수의 사전분포에 로그를 취함
           // 로그를 취하는 이유는 계산의 안정성과 효율성
           // log1m 함수는 log(1 - x)를 계산하는 Stan의 내장 함수

           real rhoprior(real rho) {
           return -0.5 * log1m(square(rho)); // log((1 - rho^2)^(-1/2))
```

```
          }
        }

        data {
          int N;           // 관측값 개수
          int D;           // 변수의 개수
          vector[D] Y[N];  // 관측값
        }

        parameters {
          vector[D] mu;                  // 평균벡터
          real<lower=0> sigma[D];        // 표준편차
          real<lower=-1, upper=1> rho;   // 상관계수
        }

        transformed parameters {
            // 공분산행렬
            cov_matrix[D] cov = [[sigma[1]^2, rho*sigma[1]*sigma[2]],
                                 [rho*sigma[1]*sigma[2], sigma[2]^2]];
        }

        model {
            // 가능도
            Y ~ multi_normal(mu, cov);

            // 사전분포
            target += -log(sigma[1]);
            target += -log(sigma[2]);
            target += rhoprior(rho);
        }
"

x1 = c(21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19.2, 17.8, 16.4, 17.3, 15.2, 10.4, 10.4, 14.7, 32.
4, 30.4, 33.9, 21.5, 15.5, 15.2, 13.3, 19.2, 27.3, 26.0, 30.4, 15.8, 19.7, 15.0, 21.4)
x2 = c(110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180, 180, 180, 205, 215, 230, 66, 52, 65, 97, 150, 15
0, 245, 175, 66, 91, 113, 264, 175, 335, 109)
df <- data.frame(x1, x2)
data <- list(N=nrow(df), D=2, Y=df)

fit <- stan(model_code=code, data=data, warmup=5000, iter=15000, seed=1234, pars = c("mu", "rho", "sigma"))
print(fit)
```

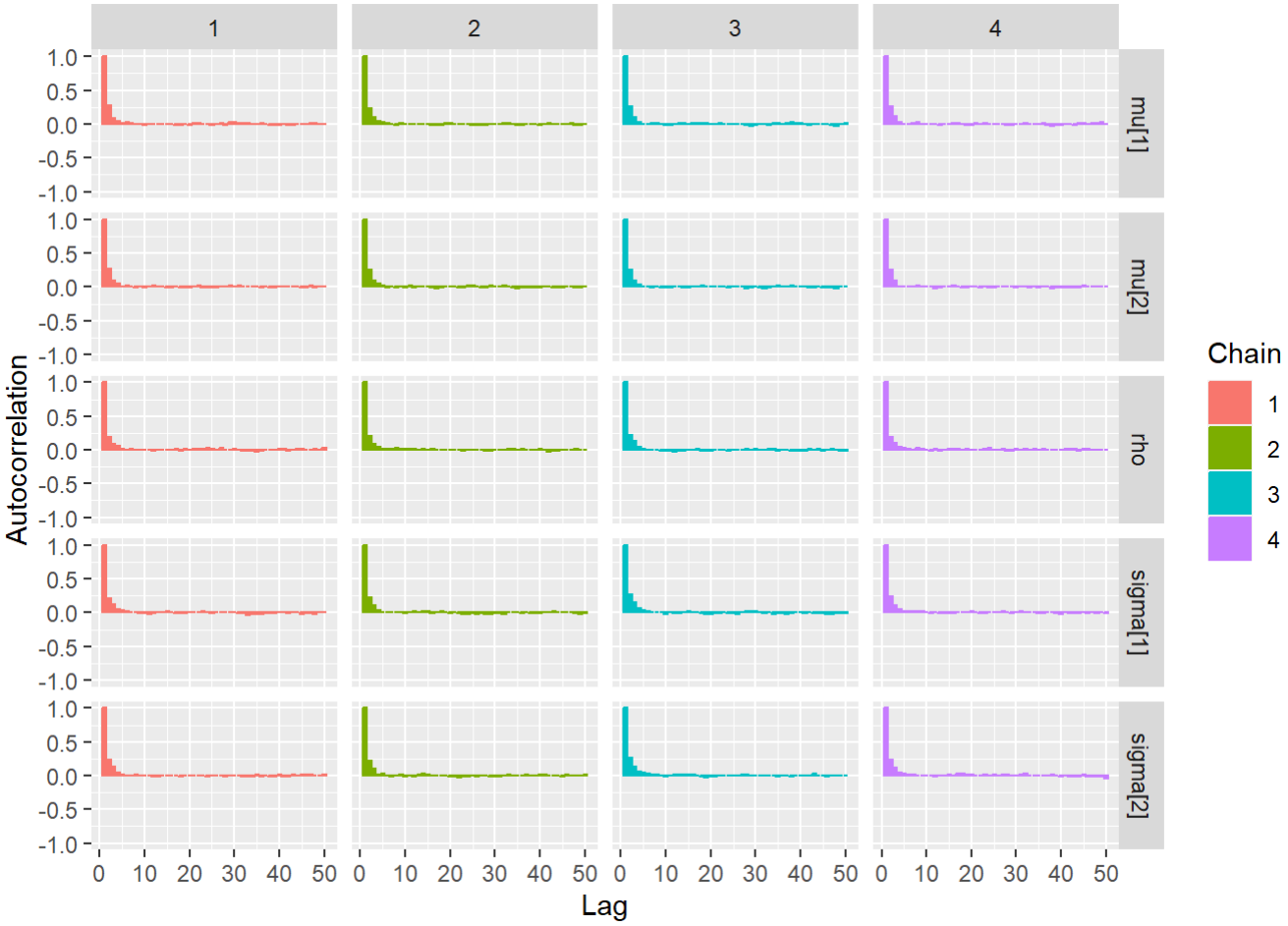
```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=15000; warmup=5000; thin=1;
## post-warmup draws per chain=10000, total post-warmup draws=40000.
##
##             mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff
## mu[1]      20.08    0.01  1.09   17.93   19.35   20.08   20.81   22.23 21427
## mu[2]     146.73    0.08 12.48  122.16  138.54  146.77  154.94  171.20 22017
## rho        -0.75    0.00  0.08   -0.87   -0.81   -0.76   -0.70   -0.56 22118
## sigma[1]    6.16    0.01  0.80    4.83    5.59    6.08    6.64    7.96 20622
## sigma[2]   70.05    0.06  9.15   54.84   63.50   69.15   75.49   90.60 20913
## lp__     -212.74    0.01  1.64 -216.76 -213.58 -212.40 -211.52 -210.58 16474
##          Rhat
## mu[1]       1
## mu[2]       1
## rho         1
## sigma[1]    1
## sigma[2]    1
## lp__        1
##
## Samples were drawn using NUTS(diag_e) at Sun Nov  5 10:12:07 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

(b) 모수들의 시계열 그림, 자기상관계수 그림을 그리고 마르코프 체인이 수렴했는지 판단하시오.
수렴하지 않았다고 판단하면 수렴했다고 판단할 때까지 사후표본의 크기를 늘리시오.

```
ggs_traceplot(ggs(fit))
```



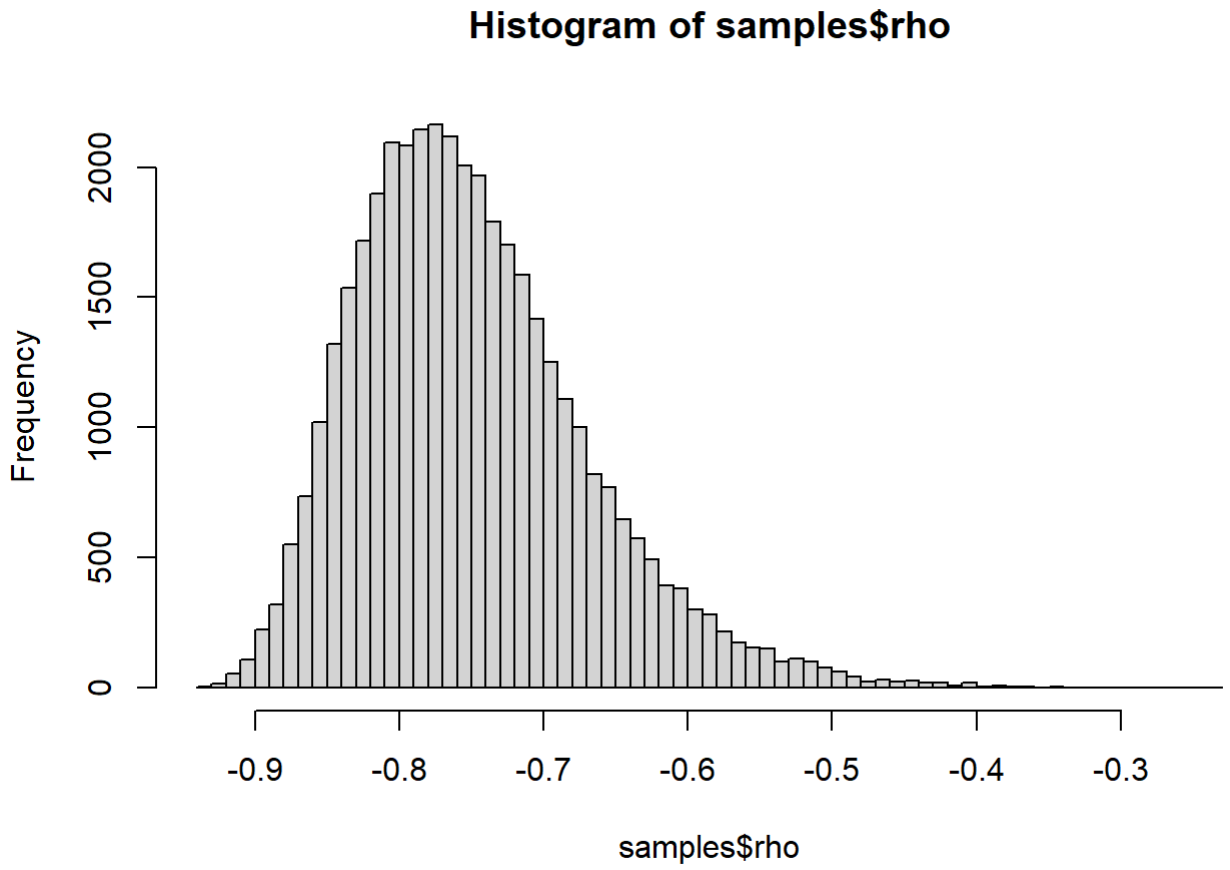

```
ggs_autocorrelation(ggs(fit))
```



시계열 그림에 패턴이 없고 무질서해 보이므로 수렴이 잘된 것으로 판단할 수 있다.
또한 Lag 0에서 자기상관계수 1로 시작한 이후 급격하게 0에 가까워지므로 수렴이 충분히 잘 된 것으로 판단된다.
아울러 fit의 출력 결과에서 Rhat의 값이 1에 가까울수록, 해당 파라미터의 샘플링 체인들이 잘 수렴했다는 것을 나타낸다.

(c) ρ의 사후표본의 히스토그램을 그리시오.

```
samples <- rstan::extract(fit)
hist(samples$rho, breaks=100)
```



(d) ρ의 사후평균, 사후표준편차, 95% 신용구간을 구하시오.

```
# 사후평균
posterior_rho <- mean(samples$rho)
cat("사후평균: ", posterior_rho, "\n")
```

```
## 사후평균:  -0.7496123
```

```
# 사후표준편차
posterior_sd <- sd(samples$rho)
cat("사후표준편차: ", posterior_sd, "\n")
```

```
## 사후표준편차:  0.08051965
```

```r
# 95% 신용구간
quantiles <- quantile(samples$rho, probs=c(0.025, 0.975))
cat("95% 신용구간: [", quantiles[1], ", ", quantiles[2], "]\n")
```

```
## 95% 신용구간: [ -0.8744542 ,  -0.5598684 ]
```

```
```

4. 참고문헌

이재용·이기재(2022), 베이즈 데이터 분석, 한국방송통신대학교출판문화원.
마쓰우라 겐타(2019), 데이터 분석을 위한 베이지안 통계 모델링 with Stan & R, 길벗.
존 크러슈케(2018), R, JAGS, Stan을 이용한 베이지안 데이터 분석 바이블 2판, 제이펍.
Stan User's Guide(https://mc-stan.org/docs/stan-users-guide/index.html)