Bayes_stat_hw2

2024-11-07

문제 풀이 전반에 관한 기본 사항

랜덤 넘버 추출 시 set.seed(42) 함수를 통해 seed를 42로 고정하여 사용하였다. 문제 풀이 중 알고리즘 설명 및 문제풀이는 R을 통해 수식을 첨부하여 진행하였다.

5장

5.11

$u(\theta) := cos(\theta^2)$로 정의하고, $\pi(\theta) \sim Unif(0, \pi)$로 정의하자. 그러면

$$I = \int_0^\pi u(x)\pi(x)\,dx = \frac{1}{\pi}\int_0^\pi cos(x^2)\,dx$$

이고, 다음과 같이 균등분포를 따르는 확률변수를 생성하자.

$$\theta_1, \theta_2, \dots, \theta_n \overset{i.i.d.}{\sim} Unif(0, \pi)$$

몬테카를로 방법을 생각하면 다음과 같은 적분값 I_n을 정의할 수 있고 이 적분값이 거의 확실히 수렴한다.

$$I_n = \frac{1}{n}\sum_{i=1}^n u(\theta_i) = \frac{1}{n}\sum_{i=1}^n cos({\theta_i}^2) \xrightarrow{\text{a.s.}} \frac{1}{\pi}\int_0^\pi cos(x^2)\,dx$$

즉 다음과 같은 계산을 통해 적분값의 추정을 할 수 있다.

$$\pi I_n \to \int_0^\pi cos(x^2)\,dx$$

```
set.seed(42)
x <- runif(5000, 0, pi) #n=5000인 경우 적분 계산
pi*mean(cos(x**2)) #pi와 곱하여 적분값 추정.
```

```
## [1] 0.5696733
```

따라서 추정된 적분값은 0.5696733이다.

5.12

사전분포가 디리클레분포이고 가능도가 다항분포인 경우, 사후분포 역시 디리클레분포로 주어지고 그 분포가 다음과 같이 주어진다는 점을 문제 2-11에서 확인하였다. 이를 문제 상황에 적용하면 사후분포는 다음과 같다.

$$(\theta_1, \theta_2, \theta_3) \sim Dirichlet((\alpha + \mathrm{x}) = (3, 4, 6))$$

이때, 디리클레분포의 주변분포는 베타분포가 된다. 보다 구체적으로, 디리클레분포의 주변분포에 대해 다음과 같은 관계가 성립한다.

$$X = (X_1, \dots, X_n) \sim Dirichlet(\alpha),\ \alpha = (\alpha_1, \dots, \alpha_n) \Longrightarrow X_i \sim Beta(\alpha_i, \sum_{j=1}^{n} \alpha_j - \alpha_i)$$

즉 이로부터 다음과 같이 $\theta_1, \theta_2, \theta_3$ 각각의 주변사후분포를 구해 몬테카를로 방법으로 사후분포를 계산할 수 있다.

$$\theta_{i1}, \theta_{i2}, \dots, \theta_{im} \overset{i.i.d.}{\sim} Beta(\alpha_i, 13 - \alpha_i),\ \alpha = (3, 4, 6)$$

theta1

```
#posterior
set.seed(42)
x1 <- rbeta(5000, 3, 10)

#사후평균
mean(x1)
```

```
## [1] 0.2307626
```

```
#사후표준편차
sd(x1)
```

```
## [1] 0.1140709
```

```
#신용구간
quantile(x1, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.05378647 0.49375611
```

theta2

```
#posterior
set.seed(42)
x2 <- rbeta(5000, 4, 9)

#사후평균
mean(x2)
```

[1] 0.307579

```
#사후표준편차
sd(x2)
```

[1] 0.1245357

```
#신용구간
quantile(x2, c(0.025, 0.975))
```

2.5%   97.5%
0.09858893 0.57521747

theta3

```
#posterior
set.seed(42)
x3 <- rbeta(5000, 6, 7)

#사후평균
mean(x3)
```

[1] 0.4613391

```
#사후표준편차
sd(x3)
```

[1] 0.1339573

```
#신용구간
quantile(x3, c(0.025, 0.975))
```

2.5%   97.5%
0.2065129 0.7238033

5.13

교재에서는 $x_1, ..., x_8$로 두었지만 샘플은 총 9개이다. 어쨌든 9개의 샘플에 대해 포아송 모형의 improper 척도 prior와 포아송 가능도를 곱해 나오는 사후분포는 다음과 같으므로, 이로부터 사후표본을 추출하면 된다.

$$\theta_1, \theta_2, \dots, \theta_m \overset{i.i.d.}{\sim} Gamma(n\bar{x}, n)$$

```r
x <- c(2, 4, 5, 6, 8, 4, 3, 1, 0)

#data
a = sum(x)
b = length(x)

#posterior
set.seed(42)
posterior_sample <- rgamma(5000, shape = a, rate = b) #rate parameter 사용

#사후평균
mean(posterior_sample)
```

```
## [1] 3.661
```

```r
#사후표준편차
sd(posterior_sample)
```

```
## [1] 0.6433149
```

```r
#신용구간
quantile(posterior_sample, c(0.025, 0.975))
```

```
##    2.5%   97.5%
## 2.493010 4.997779
```

5.15

내일 비가 올 확률의 사후예측분포는 다음과 같다.

$$\theta_1, \theta_2, \dots, \theta_n \overset{i.i.d.}{\sim} Ber(2/3)$$

그러나, 교수님과의 대화를 통해 해당 문제에서는 '확률'을 예측하도록 지시받았으므로 베타분포를 통해 문제를 풀이한다. 이 경우 이항 가능도와 베타 사전분포 케이스이므로 사후분포는 다음과 같다.

$$\theta_1, \theta_2, \dots, \theta_n \overset{i.i.d.}{\sim} Beta(8, 4)$$

```r
#posterior
set.seed(42)
x <- rbeta(5000, 8, 4)

#사후평균
mean(x)
```

```
## [1] 0.666749
```

```
#사후표준편차
sd(x)
```

```
## [1] 0.131949
```

```
#신용구간
quantile(x, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 0.3860804 0.8919417
```

8장

8.11

(a)

rate parameter가 10인 지수분포를 생성하기 위한 역함수 방법 알고리즘은 다음과 같다. 단, $Unif(0,1)$은 생성 가능하다고 가정한다.

(1) rate parameter가 10인 지수분포의 누적분포함수의 역함수 $F^{-1}(x)$를 구한다.
(2) 해당 누적분포함수의 역함수에 균등분포에 따라 생성된 $u_1, ..., u_m$을 plug-in
(3) 이 경우 $F^{-1}(u_1), ..., F^{-1}(u_m)$은 $F(x)$로부터의 iid 랜덤 샘플과 같은 분포이다.

이제 누적분포함수의 역함수를 구하면 $pdf_X(x) = \theta e^{-\theta x}$이므로 $F^{-1}(x) = -\frac{1}{\theta}log(1-x)$가 되고, $U \sim Unif(0,1)$인 경우 $U \overset{d}{=} (1-U)$임을 활용하면 $-\frac{1}{\theta}log(u) \sim Exp(\theta)$이다. 따라서, $U \sim Unif(0,1)$로부터 난수를 생성하여 $-\frac{1}{\theta}log(u)$에 대입하면 rate parameter가 θ인 지수분포를 따르는 난수가 생성된다. 이 경우 $\theta = 10$으로 대입하여 난수를 생성하면 된다.

(b)

```
set.seed(42)
z <- runif(1000, 0, 1)
x <- (-1/10)*log(z)
df_811 <- as_tibble(x)
```

(c)

```
mean(df_811$value)
```

```
## [1] 0.1050334
```

```
sd(df_811$value)
```

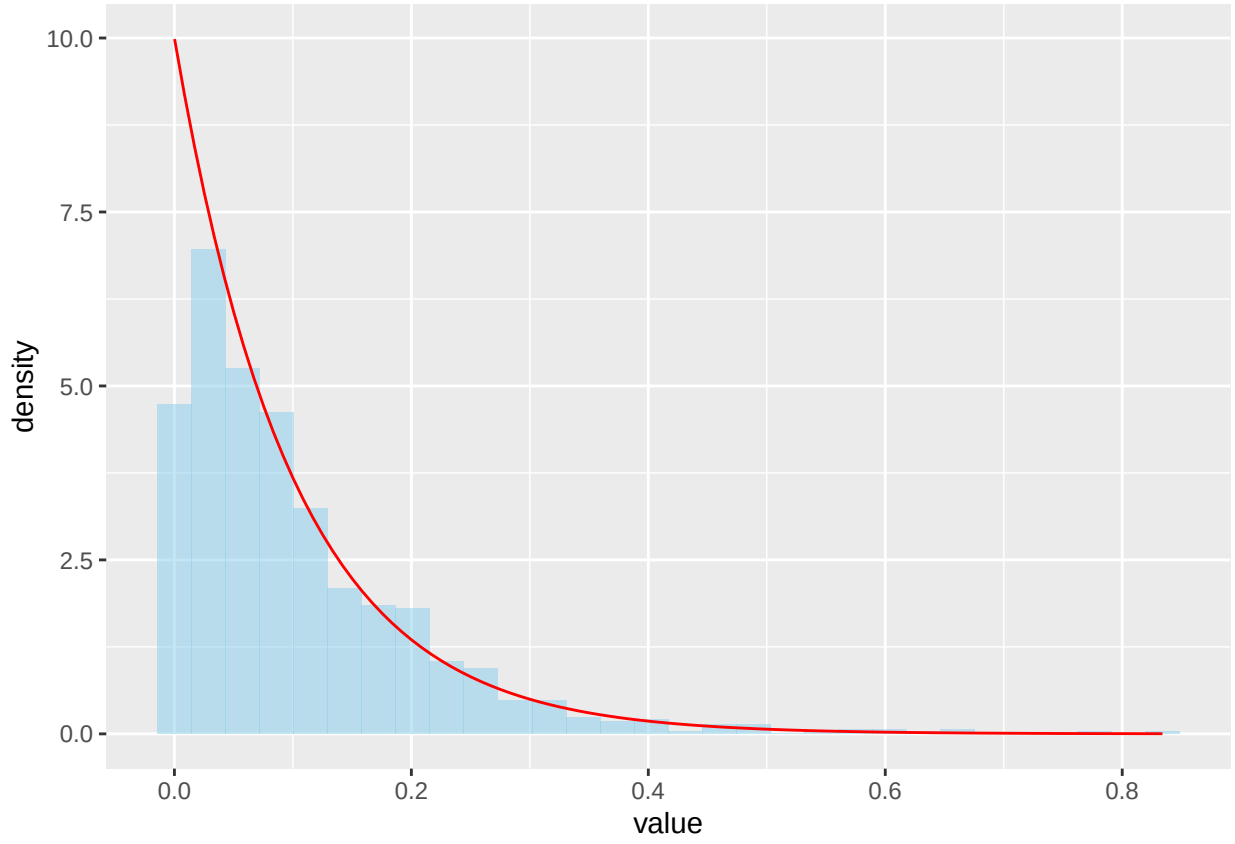
```
## [1] 0.1070077
```

rate parameter가 10인 지수분포의 평균은 1/10, 분산은 1/100, 표준편차는 1/10이다. 실제 경험적 사후분포와 유사함을 확인할 수 있다.

(d)

```
ggplot(df_811, aes(x = value)) +
 geom_histogram(aes(y = after_stat(density)), fill = "skyblue", alpha = 0.5) +
 stat_function(fun = dexp, args = list(rate = 10), color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



8.12

(a)

마찬가지로, $Cauchy(0,1)$의 누적분포함수를 구하고, 그 역함수에 $Unif(0,1)$로부터에 난수를 대입한 값은 원래의 분포를 따른다. 코시 분포의 누적분포함수 및 그 역함수는 다음과 같다.

$$f_X(x) = \frac{1}{\pi(1+x^2)} I\{x \in \mathbb{R}\}$$

$$F_X(x) = \frac{1}{2} + \frac{1}{\pi} \arctan(x)$$

$$F_X^{-1}(u) = \tan(\pi(x - \frac{1}{2}))$$

(b)

```
set.seed(42)
z <- runif(1000, 0, 1)
x <- tan(pi*(z-0.5))
df_812 <- as_tibble(x)
```

(c)

```
quantile(df_812$value, c(0.25, 0.5, 0.75))
```

```
##       25%       50%       75%
## -1.13840874 -0.06183899  0.96264184
```

한편, 코시 분포에서는 cdf의 역함수가 closed form으로 주어지므로 삼각함수에 대한 방정식을 풀어 백분위를 구할 수 있다. 다음과 같이 백분위수를 구하면 모집단 $Cauchy(0,1)$에서의 백분위수는 다음과 같다.

$$Ca_{0.25}(0,1) = F^{-1}(\frac{1}{4}) = \tan(-\frac{\pi}{4}) = -1$$

$$Ca_{0.5}(0,1) = F^{-1}(\frac{1}{2}) = \tan(0) = 0$$

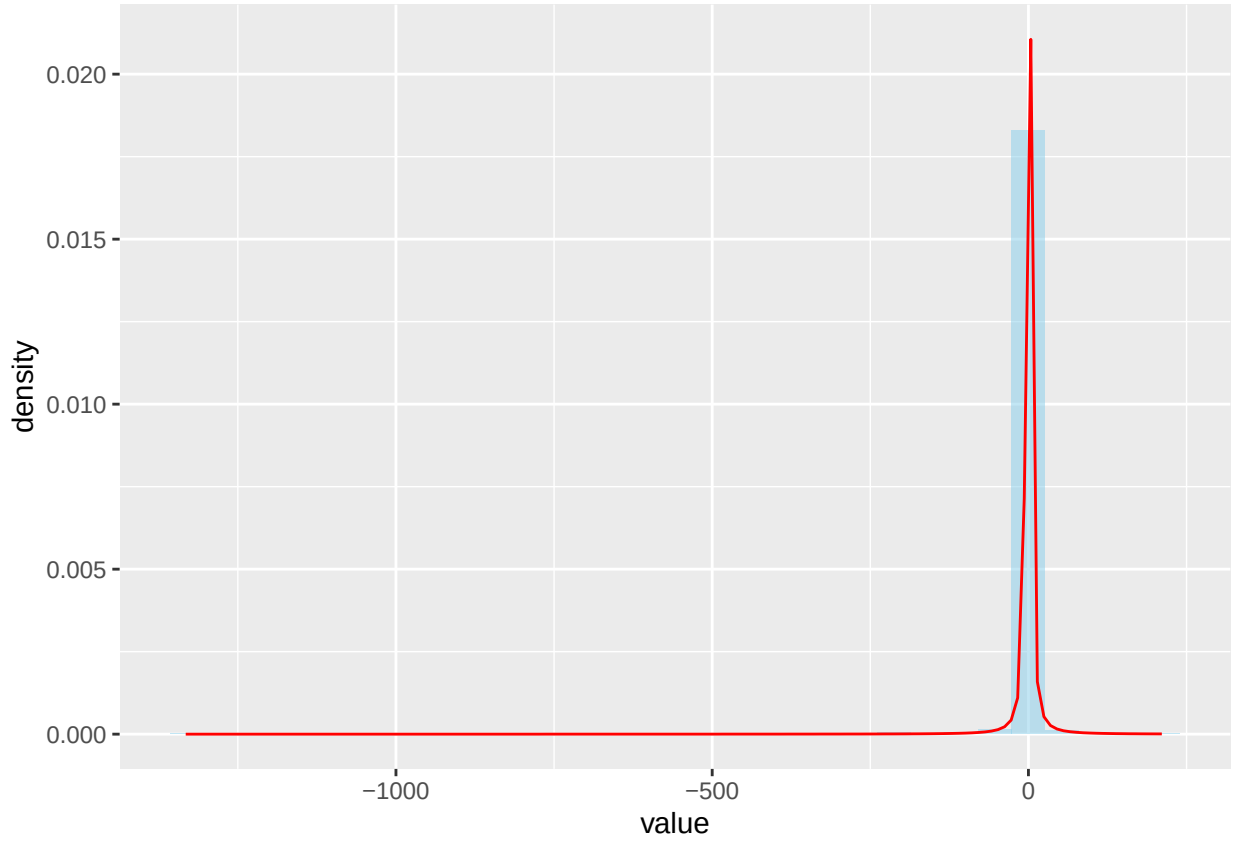
$$Ca_{0.75}(0,1) = F^{-1}(\frac{3}{4}) = \tan(\frac{\pi}{4}) = 1$$

이론상의 백분위수와 사후표본에서의 백분위수가 크게 차이나지 않는 것을 확인할 수 있다.

(d)

```
ggplot(df_812, aes(x = value)) +
  geom_histogram(aes(y = after_stat(density)), fill = "skyblue", alpha = 0.5) +
  stat_function(fun = dcauchy, args = list(location = 0, scale = 1), color = "red", n = 150)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



8.13

(a)

합격-불합격 방법의 구현은 두 단계로 나누어 실행한다. 우선, 한 단계의 합격-불합격 방법을 통해 한 개의 사후표본을 생성하는 함수를 정의한다. 다음으로, m회 해당 함수를 반복하여 해당 샘플링을 완성하여 데이터프레임에 저장한다.

```
triangle_sample <- function(){
  while (TRUE) {
    x <- runif(1, 0, 2) #제안밀도함수에서 추출
    u <- runif(1, 0, 1) #합격 여부 판단을 위한 난수
    a <- ifelse(x <= 1, x, 2-x) #합격 비율 계산
    if (u <= a) {
      return(x)
    }
  }
}

set.seed(42)
y <- c()
for (i in 1:1000) {
  y <- append(y, triangle_sample())
}
```

```
df_813 <- as_tibble(y)
```

(b)

```
mean(df_813$value)
```

```
## [1] 0.9958629
```

```
sd(df_813$value)
```

```
## [1] 0.4063675
```

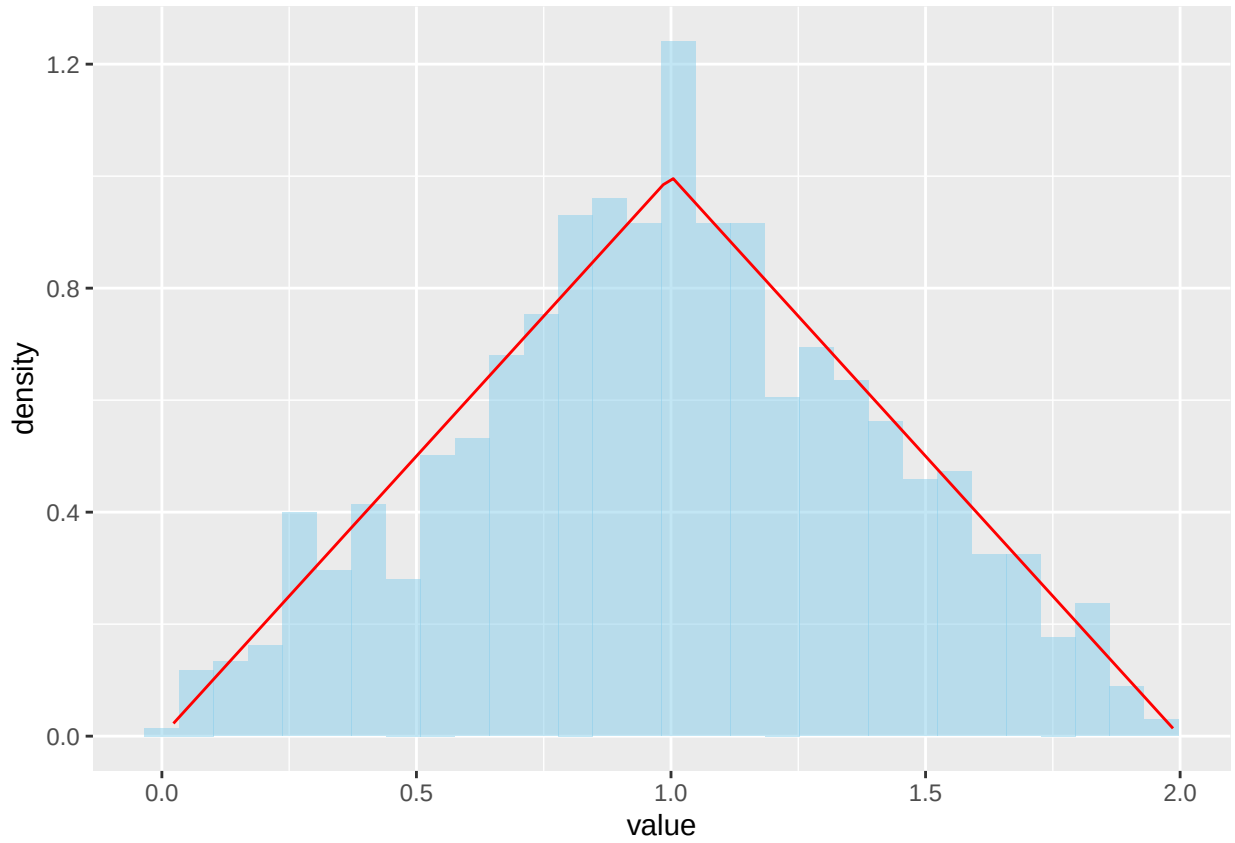
이론적으로 $xI\{0 \leq x \leq 1\} + (2-x)I\{1 < x \leq 2\}$와 같이 정의된 삼각분포의 평균은 1, 표준편차는 $\frac{1}{\sqrt{6}}$이다. 이와 같은 이론적 값과 실제 분포상 값 0.9958629, 0.4063675가 유사함을 확인할 수 있다.

(c)

```
tri_pdf <- function(x) {
  ifelse(x >= 0 & x <= 1, x,
       ifelse(x > 1 & x <= 2, 2 - x, 0))
}

ggplot(df_813, aes(x = value)) +
  geom_histogram(aes(y = after_stat(density)), fill = "skyblue", alpha = 0.5) +
  stat_function(fun = tri_pdf, color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



8.14

(a)

다음과 같이 코시 분포를 제안분포로 하여 1개의 정규확률변수를 생성하는 함수를 정의한다.

```
normal_sample <- function(){
  M <- 2/sqrt(exp(1)) # 봉투상수
  while (TRUE) {
    x <- rcauchy(1, 0, 1) #제안밀도함수에서 추출
    u <- runif(1, 0, 1) #합격 여부 판단을 위한 난수
    a <- (1+x^2)*exp(-(x^2)/2)/M #합격 비율 계산
    if (u <= a) {
      return(x)
    }
  }
}
```

이제 위의 함수를 이용하여 1000개의 정규확률변수를 생성한다.

```
set.seed(42)
y <- c()
for (i in 1:1000) {
  y <- append(y, normal_sample())
```

```
}
df_814 <- as_tibble(y)
```

(b)

```
mean(df_814$value)
```

```
## [1] 0.0437004
```

```
sd(df_814$value)
```

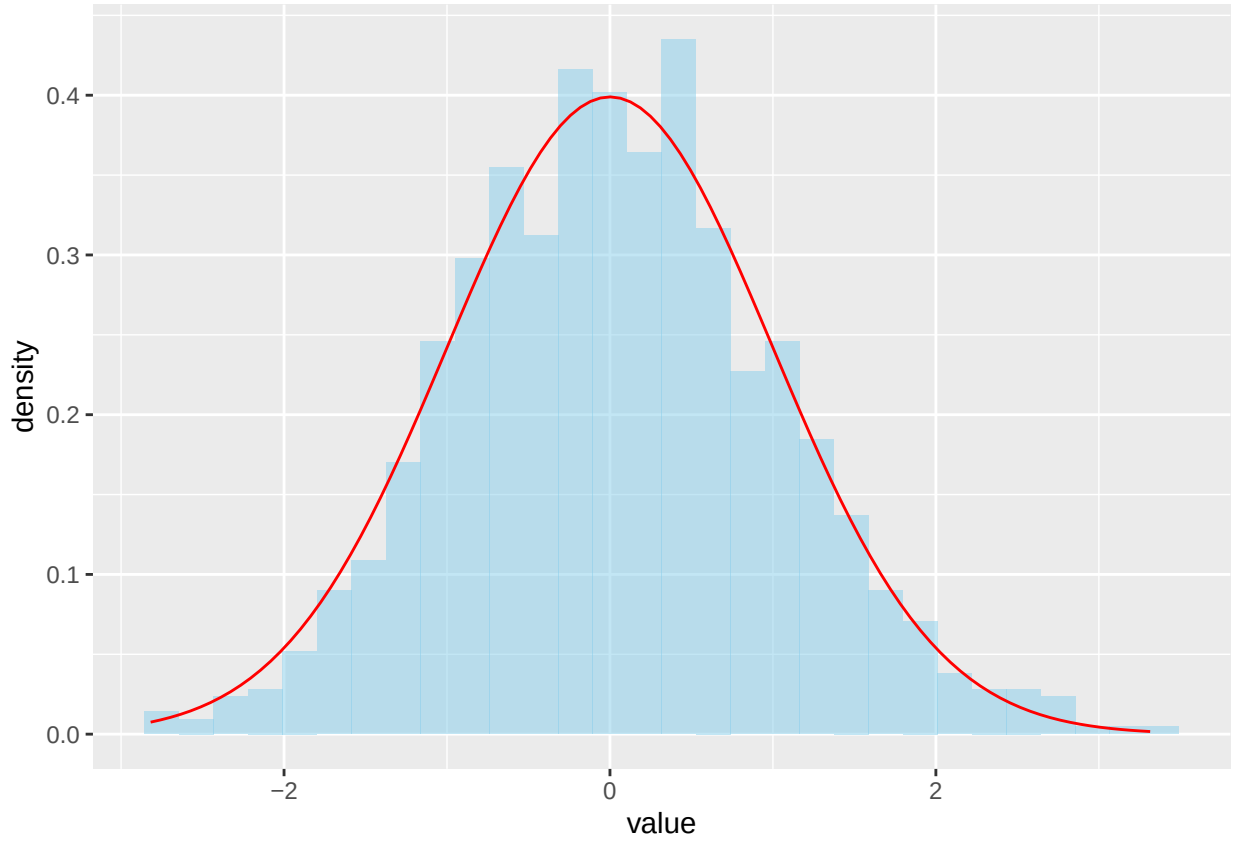
```
## [1] 0.9954824
```

표준정규분포 $N(0, 1^2)$의 평균은 0, 표준편차는 1이므로 사후평균 0.0437004, 사후표준편차 0.9954824와 유사하다.

(c)

```
ggplot(df_814, aes(x = value)) +
  geom_histogram(aes(y = after_stat(density)), fill = "skyblue", alpha = 0.5) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1), color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



8.15

(a)

박스-뮐러 변환을 통한 정규확률변수 생성은 두 단계로 구성된다. 우선, $(u1, u2) \sim Unif((0,1)^2)$를 생성할 수 있으므로 이로부터 적절한 지수분포와 균등분포를 생성할 수 있다. 다음으로, 서로 독립인 이 지수분포와 균등분포를 극좌표계 표현을 통해 조합함으로써 서로 독립인 정규확률변수를 생성할 수 있다.

코드의 u1, u2는 균등분포를 따르는 랜덤 확률변수를 생성하는 부분이고, v와 t는 이와 같이 생성된 균등분포로부터 지수분포, 균등분포를 따르는 서로 독립인 적절한 확률분포(또는 확률분포의 상수배)를 생성하는 과정이며, 세 번째 x1, x2는 지수분포와 균등분포의 함수로써 생성된 이변량 정규분포 x1, x2의 좌표이다.

```
set.seed(42)
u1 <- runif(1000, 0, 1)
u2 <- runif(1000, 0, 1)
v <- -2*log(1-u1) #transform (u1, u2) to (v, t)
t <- 2*pi*u2
x1 <- sqrt(v)*cos(t) #transform (v, t) to (x1, x2)
x2 <- sqrt(v)*sin(t)
df_815 <- data.frame(x1, x2) #save bivariate normal distribution (x1, x2)
```

(b)

```
colMeans(df_815)
```

```
##         x1         x2
##  0.04197130 -0.01846759
```

```
var(df_815)
```

```
##            x1         x2
## x1  0.97851484 -0.01314774
## x2 -0.01314774  0.94863882
```

```
sd(df_815$x1)
```

```
## [1] 0.9891991
```

```
sd(df_815$x2)
```

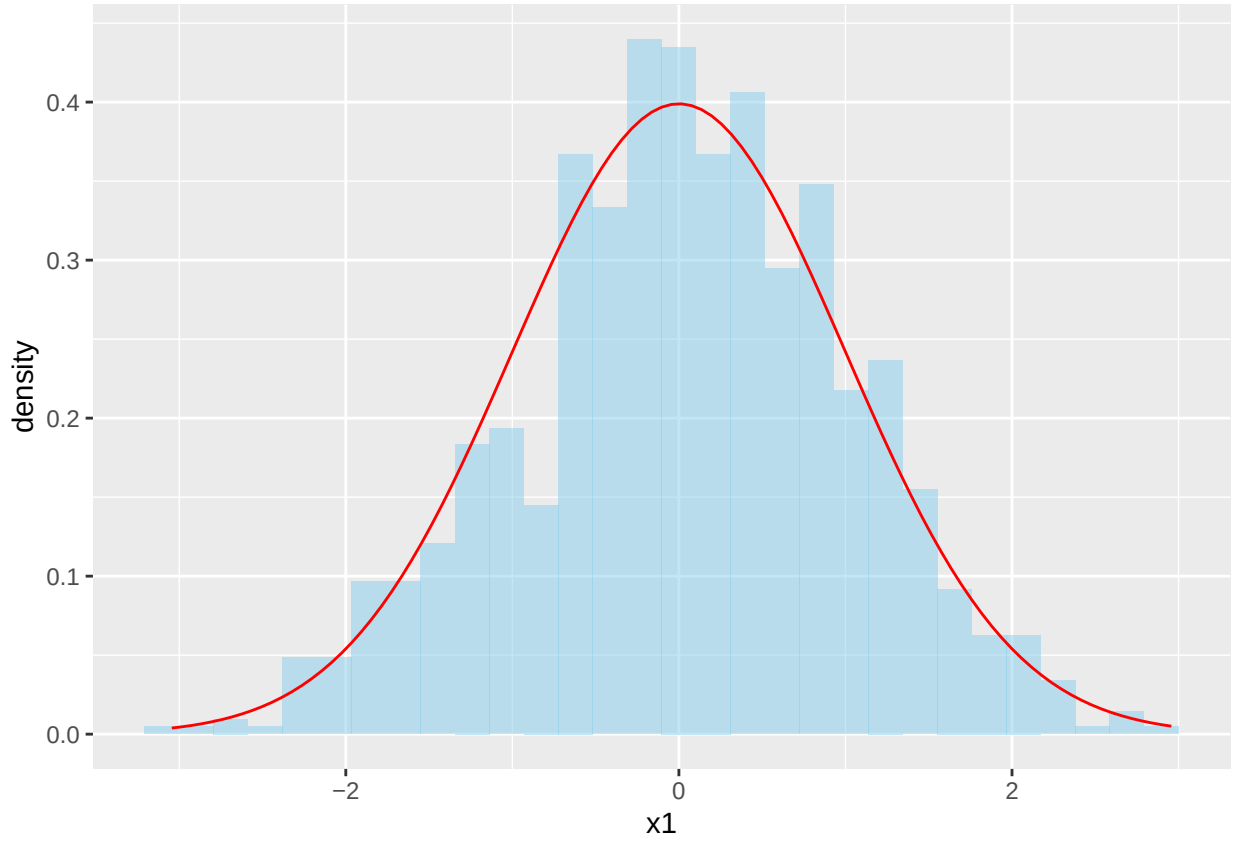
```
## [1] 0.9739809
```

표본 1000개의 평균과 표준편차는 위와 같다. 이때, 이변량 정규분포의 평균벡터는 $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$이고 분산행렬은 $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$이다. 물론 x1, x2 각각 표준편차는 1이다. 박스-뮐러 변환을 통해 생성된 표준편차들 (0.9891991, 0.9739809)과과 이론상 값이 큰 차이가 없는 것을 확인 가능하다.

(c)

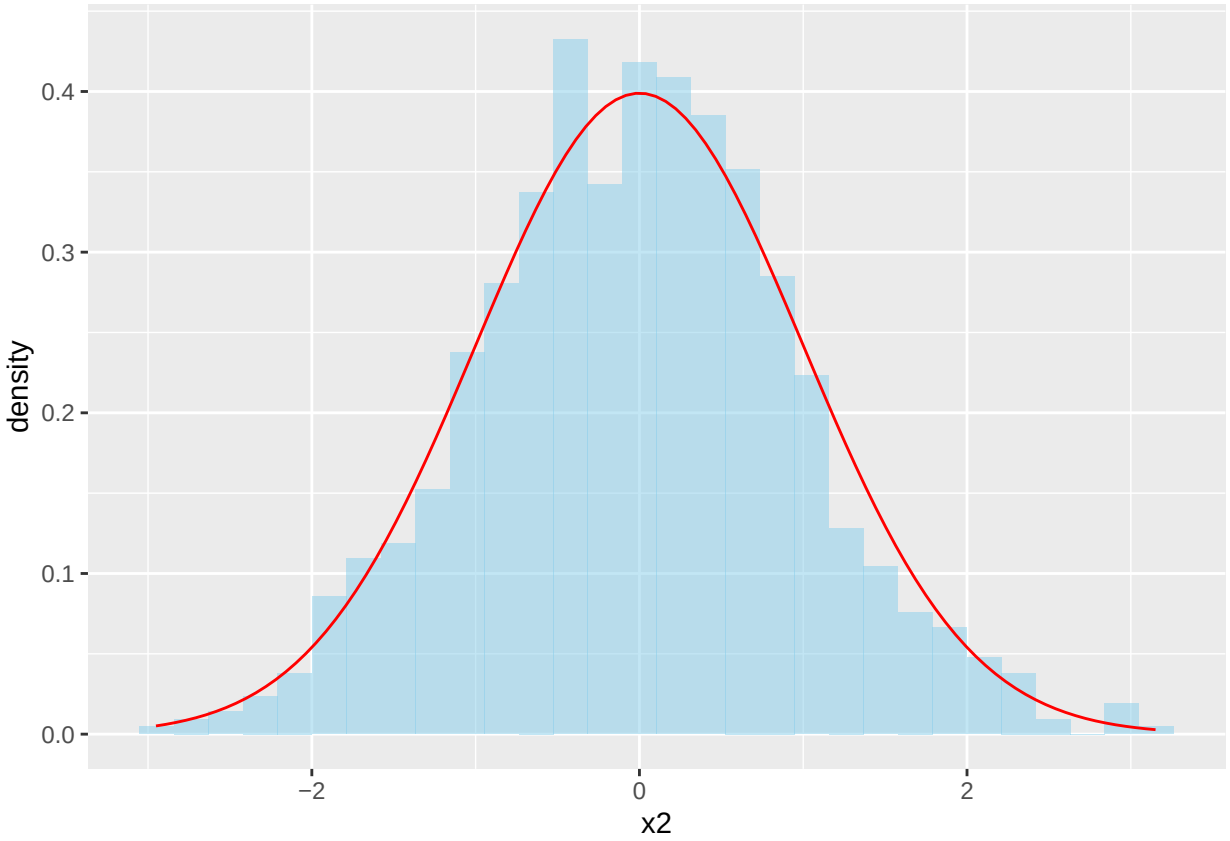
```
ggplot(df_815, aes(x = x1)) +
 geom_histogram(aes(y = after_stat(density)), fill = "skyblue", alpha = 0.5) +
 stat_function(fun = dnorm, args = list(mean = 0, sd = 1), color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(df_815, aes(x = x2)) +
 geom_histogram(aes(y = after_stat(density)), fill = "skyblue", alpha = 0.5) +
 stat_function(fun = dnorm, args = list(mean = 0, sd = 1), color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


density
0.4
0.3
0.2
0.1
0.0
−2
0
2
x2