

560 A8

tom

2017-11-25

```
install.packages("smoothmest",repos = "http://cran.us.r-project.org")

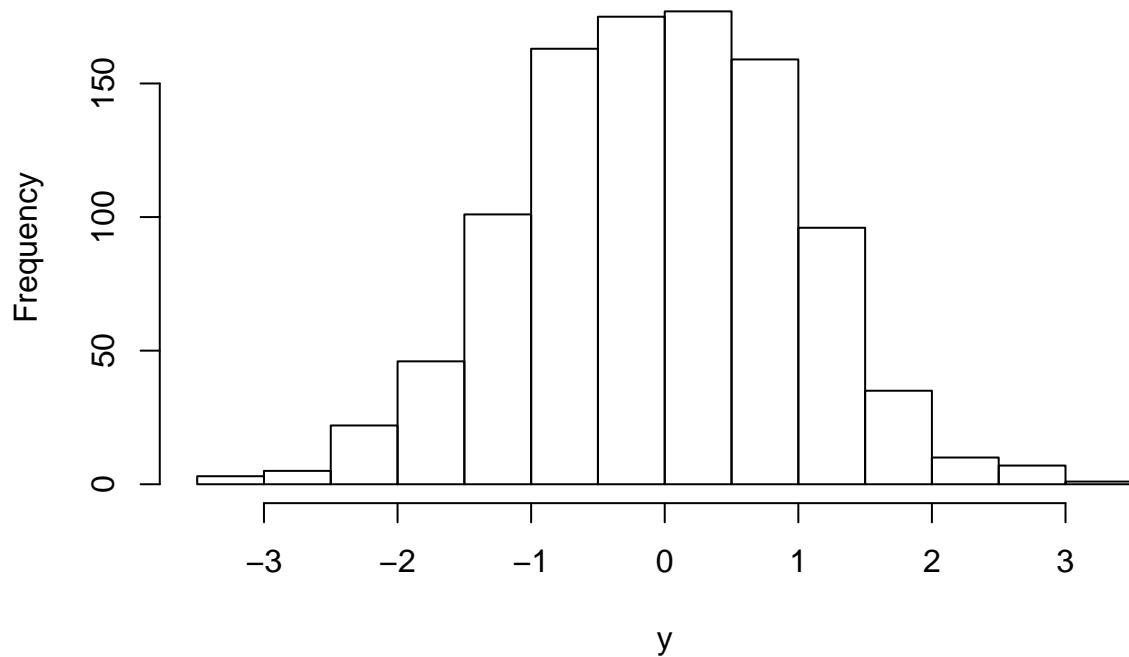
##  
## The downloaded binary packages are in  
## /var/folders/l7/y2ztr08x2qb06_dt131rdcz00000gn/T//RtmpCd7i14 downloaded_packages  
library(smoothmest)

## Loading required package: MASS
```

(b) Write a code in R to implement the rejective sampling method to generate $n = 1000$ observations from $N(0, 1)$.

```
n<-numeric()  
y<-numeric()  
u<-(2*exp(1)/pi)^(1/2)  
for (i in 1:1000) {  
  
  x <- rdoublex(1,mu=0,lambda=1)  
  f_0_x<-ddoublex(x)  
  f_x<-dnorm(x)  
  pi_x<-f_x/(f_0_x*u)  
  
  uni<-runif(1,0,1)  
  tmp <- 1  
  while( uni>pi_x){  
    x <- rdoublex(1,mu=0,lambda=1)  
    f_0_x<-ddoublex(x)  
    f_x<-dnorm(x)  
    pi_x<-f_x/(f_0_x*u)  
    uni<-runif(1,0,1)  
    tmp <- 1+tmp  
  }  
  y[i] <- x  
  n[i] <- tmp  
}  
  
hist(y)
```

Histogram of y

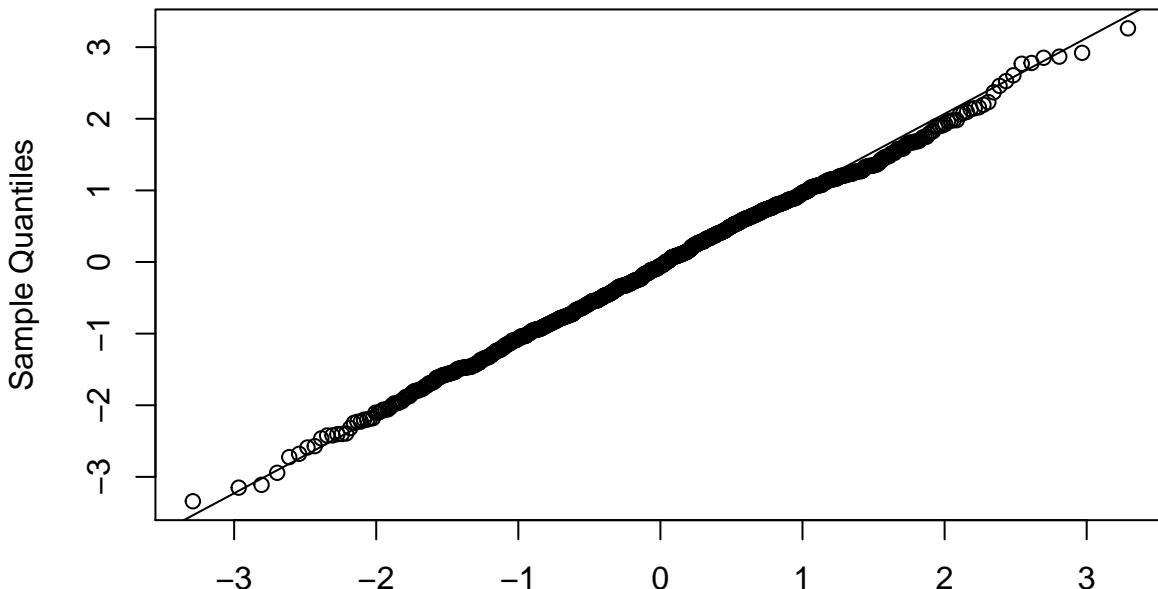


I drew the histogram of y and it appears that y follows standard normal distribution.

(c) Work out the Q-Q plot of the data generated and report the number of pairs of (X, U) in rejective sampling required.

```
(num_of_runs <- sum(n))  
## [1] 1353  
qqnorm(y)  
qqline(y)
```

Normal Q-Q Plot



Theoretical Quantiles

It seems the samples generated by the rejective sampling follow normal distribution. The number of pairs of (X, U) in the sampling required appears to 1320.

(d) How many pairs of (X, U) do you expect to be needed to generate $n = 1000$ normally distributed random numbers with this method?

(u)*1000

```
## [1] 1315.489
```

The expected value is 1315.489.

4 (b) Write a code to generate data by Gibbs sampler method from the above posterior distribution. Generate $N = 1000$ of pairs. Obtain their means.

As sharing codes is allowed, I refer this code from one of my friends.

```
install.packages("MCMCpack", repos = "http://cran.us.r-project.org")
```

```
##  
## The downloaded binary packages are in  
## /var/folders/17/y2ztr08x2qb06_dt131rdcz00000gn/T//RtmpCd7i14 downloaded_packages  
install.packages("plot3D", repos = "http://cran.us.r-project.org")
```

```
##  
## The downloaded binary packages are in  
## /var/folders/17/y2ztr08x2qb06_dt131rdcz00000gn/T//RtmpCd7i14 downloaded_packages  
library(MCMCpack)
```

```
## Loading required package: coda
```

```

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2017 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0350646 and SES-0350613)
## ##

library(plot3D)

data.gib <- c(1.1777518, -0.5867896, 0.2283789, -0.1735369, -0.2328192,
  1.0955114, 1.2053680, -0.7216797, -0.3387580, 0.1620835,
  1.4173256, 0.0240219, -0.6647623, 0.6214567, 0.7466441,
  1.9525066, -1.2017093, 1.9736293, -0.1168171, 0.4511754)

n <- length(data.gib)
x_bar <- mean(data.gib)

mu <- sigma_2 <- rep(NA, 1000)

sigma_2[1] <- 1

for (i in 1:1000) {
  set.seed(i)
  mu[i] <- rnorm(n=1, mean=4*n*x_bar/(4*n+sigma_2[i]), sd=sqrt(4*sigma_2[i]/(4*n+sigma_2[i])))

  sigma_2[i+1] <- rinvgamma(n=1, shape=5+n/2, scale=1/2*(sum(data.gib^2)-2*mu[i]*sum(data.gib)+n*mu[i]^2))
}

(mean_mu <- mean(mu))

## [1] 0.3503181
sigma_2 <- sigma_2[-1001]
(mean_sigma_2 <- mean(sigma_2))

## [1] 0.6445622

```

(c) Plot the density function in (a) and a density estimator based on posterior sample obtained in (b).

```

mu2 <- rep(seq(-2,2,0.01),each=401)
sigma2 <- rep(seq(0.01,4.01,by=0.01),times=401)
fx <- rep(NA,length(mu2))

for (i in 1:length(mu2)) {
  fx[i] <- sigma2[i]^(16)*exp(-1/2*sigma2[i])*(19.947-14.038*mu2[i]+20*mu2[i]^2-(mu2[i]^2)/8)
}

par(mfrow=c(1,2))
scatter3D(mu2,sigma2,fx,xlab="mu",ylab="sigma^2",zlab="density",main="density ftn",colkey=list(width=0.5))

mu3 <- sigma3 <- rep(NA,1000000)

```

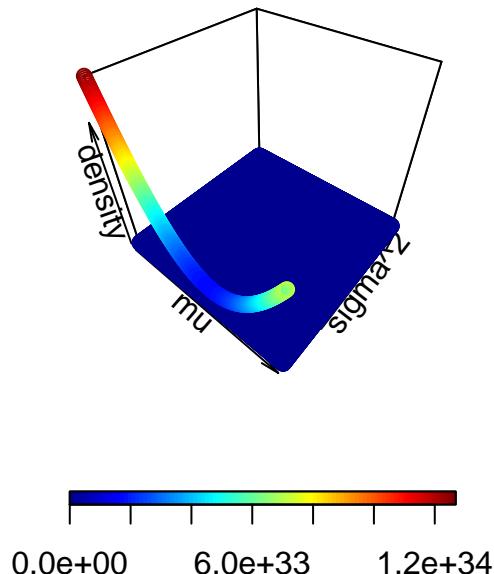
```

sigma3[1] <- 1
for(i in 1:1000000)
{
  set.seed(i)
  mu3[i] <- rnorm(n=1, mean=4*n*x_bar/(4*n+sigma3[i]), sd=sqrt(4*sigma3[i]/(4*n+sigma3[i])))
  sigma3[i+1] <- rinvgamma(n=1, shape=5+n/2, scale=1/2*(sum(data.gib^2)-2*mu3[i]*sum(data.gib)+n*mu3[i]^2))
}
sigma3 <- sigma3[-1000001]
fx2 <- sigma3^(-16)*exp(-1/(2*sigma3)*(19.947-14.038*mu3+20*mu3^2)-(mu3^2)/8)

scatter3D(mu3,sigma3,fx2,xlab="mu",ylab="sigma^2",zlab="density",main="density estimator",colkey=list

```

density ftn



density estimator

