MCMC 3

Example 1.2

Hedibert Freitas Lopes

18/12/2025

Contents

1 Figure 1.1 **1**

1 Figure 1.1

Values from some of the distributions described in Chapter 1 were obtained. The plots of the resulting histograms are shown in Figure 1.1 of the book, where a very good agreement with the corresponding density is clearly observed.

- (a) N(0,1) - central limit theorem applied to averages of uniforms
- (b) N(0,1) - probability integral transform
- (c) N(0,1) - Box-Muller algorithm
- (d) N(0,1) - Modified Box-Muller algorithm
- (e) Poisson
- (f) Gamma(a,1) with a < 1
- (g) Gamma(a,1) with a > 1
- (h) Beta(a,b) with a,b<1
- (i) Student's *t*

```
set.seed(826486)
par(mfrow=c(3,3))

# (a) Generating N(0,1) by applying the Central Limit Theorem
#     for the limit of a sequence of U(0,1) draws.
n  =  20
M  =  10000
u  = matrix(runif(M*n),M,n)
mu = apply(u,1,mean)
x  = (mu-0.5)/sqrt(1/(12*n))
y  = seq(-5,5,length=50)
hist(x,prob=T,main="(a)",breaks=y,ylab="",xlab="")
lines(y,dnorm(y),col=1)

# (b) Generating N(0,1) by using the inverse transformation
xnum = rep(0,5)
xden = rep(0,5)
xnum[1]  =  -0.322232431088
xnum[2]  =  -1.000000000000
xnum[3]  =  -0.342242088547
xnum[4]  =  -0.020423121025
```

```
xnum[5]  =  -0.000045364221
xden[1]  =   0.099348462606
xden[2]  =   0.588581570495
xden[3]  =   0.531103462366
xden[4]  =   0.103537752850
xden[5]  =   0.003856070063

polinom  =  function(a,x){
  polin  =   a[5]
  for (i in 4:1)
     polin  =   a[i] + polin*x
  return(polin)
}

invnorm  =  function(p){
  if (p <= 0.5){
    sign  =   -1.0
    z  =   p
    y  =   sqrt(-2.0*log(z))
    out1  =  polinom(xnum,y)
    out2  =  polinom(xden,y)
    invnor  =   y + out1/out2
    invnor  =   sign*invnor
  }
  else{
    sign  =   1.0
    z  =   1.0 - p
    y  =   sqrt(-2.0*log(z))
    out1  =  polinom(xnum,y)
    out2  =  polinom(xden,y)
    invnor  =   y + out1/out2
    invnor  =   sign*invnor
  }
}

# Testing the generator
# ---------------------
M = 10000
u = runif(M)
x = rep(0,M)
for (i in 1:M)
  x[i] = invnorm(u[i])
y  = seq(-5,5,length=50)
hist(x,prob=T,main="(b)",breaks=y,ylab="",xlab="")
lines(y,dnorm(y),col=1)

# (c) Generating N(0,1) by applying the Box-Muller result
M  =  10000
x  =  matrix(0,M/2,2)
for (i in 1:(M/2)){
   u  =  runif(2)
   x[i,1]  =  sqrt(-2*log(u[1]))*cos(2*pi*u[2])
   x[i,2]  =  sqrt(-2*log(u[1]))*sin(2*pi*u[2])
```

```
}
x  =  c(x[,1],x[,2])
y  = seq(-5,5,length=50)
hist(x,prob=T,main="(c)",breaks=y,ylab="",xlab="")
lines(y,dnorm(y),col=1)

# (d) Generating N(0,1) by using the polar rejection algorithm
#     Gentle, J.E. (1998) page 89
rnorm.polar = function(M){
    x = NULL
    n = 0
    repeat{
      v = -1+2*runif(2)
      r2 = sum(v^2)
      if (r2<=1){
       x = c(x,v[1]*sqrt(-2*log(r2)/r2))
       n = n+1
     }
     if (n==M) break
   }
   return(x)
}

M = 10000
X = rnorm.polar(M)
y  = seq(-5,5,length=50)
x = seq(min(X),max(X),length=1000)
hist(X,prob=T,col=0,main="(d)",breaks=y,ylab="",xlab="")
lines(x,dnorm(x))


# (e) Generating Poisson(mu) variates
rpois1 = function(M,mu){
    x = NULL
    for (i in 1:M){
      P = 1
      N = 0
      C = exp(-mu)
      repeat{
        U = runif(1)
        P = P*U
        N = N+1
        if (P<C){
          x = c(x,N)
          break
        }
     }
   }
   return(x)
}

M    = 10000
X    = rpois1(M,5)
```

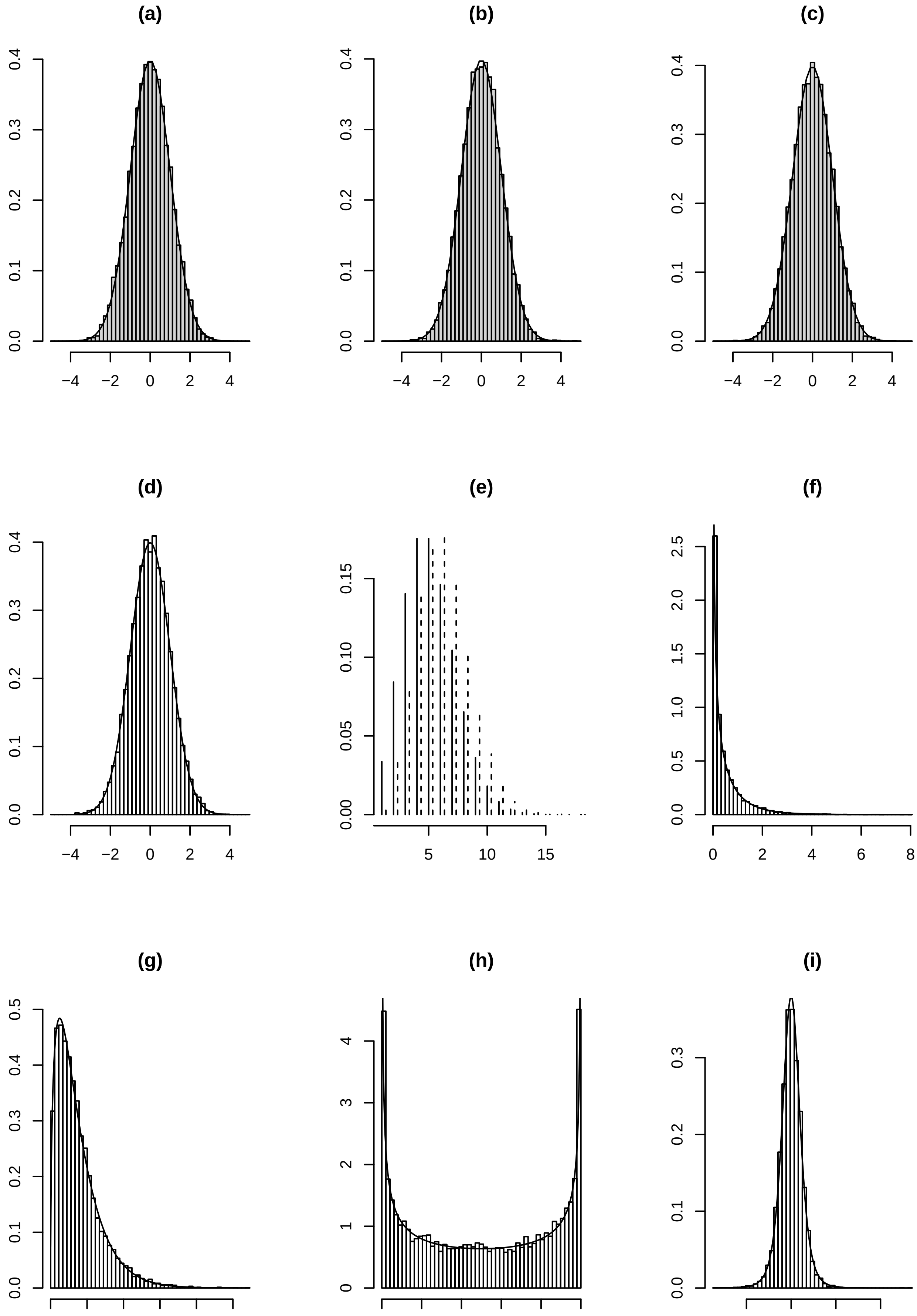
```r
ymax = max(table(X)/M,dpois(X,5))
x    = seq(min(X),max(X),by=1)

plot(x,dpois(x,5),ylim=c(0,ymax),pch=1,xlab="",ylab="",type="h",axes=F)
title("(e)")
axis(1,at=seq(0,max(x),by=5))
axis(2)
lines(sort(unique(X))+0.35,table(X)/M,type="h",lty=2)


# (f) and (g) Generating from Gamma(alpha,1)
rgamma1 = function(M,alpha){
  if (alpha<1){
    i = 0
    x = NULL
    repeat{
      U = runif(2)
      if (U[1]>(exp(1)/(alpha+exp(1)))){
         X = -log((alpha+exp(1))*(1-U[1])/(alpha*exp(1)))
          accept = ifelse(U[2]<X^(alpha-1),1,0)
       }
      else{
         X = ((alpha+exp(1))*U[1]/exp(1))^(1/alpha)
         accept = ifelse(U[2]<exp(-X),1,0)
      }
      if (accept==1){
         i = i+1
         x = c(x,X)
      }
      if (i==M) break
   }
   return(x)
  }
  else{
   c1 = alpha-1
   c2 = (alpha-1/(6*alpha))/c1
   c3 = 2/c1
   c4 = c3+2
   c5 = 1/sqrt(alpha)
   x = NULL
   for (j in 1:M){
   repeat{
     repeat{
         U = runif(2)
         if (alpha>2.5){
            U[1] = U[2]+c5*(1-1.86*U[1])
         }
        if ((U[1]>0)&(U[1]<1)){
            break
        }
     }
     W = c2*U[2]/U[1]
     if (((c3*U[1]+W+1/W)<c4)|((c3*log(U[1])-log(W)+W)<1)) break
```

```
   }
   x = c(x,c1*W)
  }
  return(x)
  }
}

M  = 10000
X1 = rgamma1(M,0.5)
X2 = rgamma1(M,1.5)
breaks = seq(min(X1),max(X1),length=50)
xx     = seq(min(X1),max(X1),length=1000)
hist(X1,prob=T,col=0,main="(f)",breaks=breaks,xlab="",ylab="")
lines(xx,dgamma(xx,0.5))

breaks = seq(min(X2),max(X2),length=50)
xx     = seq(min(X2),max(X2),length=1000)
hist(X2,prob=T,nclass=20,col=0,main="(g)",ylim=c(0,0.5),xlab="",ylab="",breaks=breaks)
```

```
## Warning in hist.default(X2, prob = T, nclass = 20, col = 0, main = "(g)", :
## 'nclass' not used when 'breaks' is specified
```

```
lines(xx,dgamma(xx,1.5))

# (h) Generating Beta(alfa,beta) variates
rbeta1 = function(M,alfa,beta){
    x = NULL
    for (i in 1:M){
      repeat{
         U = runif(2)
         V = c(U[1]^(1/alfa),U[2]^(1/beta))
         W = sum(V)
         if (W<=1) break
       }
       x = c(x,V[1]/W)
   }
   return(x)
}

M = 10000
X1 = rbeta1(M,0.5,0.5)
xx = seq(min(X1),max(X1),length=50)
hist(X1,prob=T,col=0,main="",breaks=xx,ylab="",xlab="")
xx = seq(min(X1),max(X1),length=1000)
lines(xx,dbeta(xx,0.5,0.5))
title("(h)")

# (i) Generating t-Student(nu,mu,sigma2) variates
#     Auxiliary functions:  rgamma1 and rnorm1 (box-muller)
rt1 = function(M,nu,mu,sigma2){
  precision = 2*rgamma1(M,nu/2)/nu
  x =  matrix(0,M/2,2)
  for (i in 1:(M/2)){
     u  =  runif(2)
```

```r
    x[i,1]  =  sqrt(-2*log(u[1]))*cos(2*pi*u[2])
    x[i,2]  =  sqrt(-2*log(u[1]))*sin(2*pi*u[2])
  }
  return(sqrt(sigma2/precision)*c(x[,1],x[,2]))
}

M      = 10000
nu     = 5
mu     = 0.0
sigma2 = 1.0
X1     = rt1(M,nu,mu,sigma2)
xx = seq(min(X1),max(X1),length=50)
hist(X1,prob=T,col=0,main="",breaks=xx,xlab="",ylab="")
xx = seq(min(X1),max(X1),length=1000)
lines(xx,dt(xx,5))
title("(i)")
```


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
0.0 0.1 0.2 0.3 0.4
−4 −2 0 2 4
0.00 0.05 0.10 0.15
5 10 15
0.0 0.5 1.0 1.5 2.0 2.5
0 2 4 6 8
0.0 0.1 0.2 0.3 0.4 0.5
0 2 4 6 8 10
0 1 2 3 4
0.0 0.2 0.4 0.6 0.8 1.0
0.0 0.1 0.2 0.3
−5 0 5 10