統計模擬HW5

1

1.

(a) $Y_i \overset{iid}{\sim} N(\mu, \sigma^2)$ $\quad \mathbf{y} = (y_1 \cdots y_n)$

$$f(\mathbf{y}|\mu,\sigma^2) = \prod_{i=1}^{n} f(y_i|\mu,\sigma^2) \propto \prod_{i=1}^{n} (\sigma^2)^{-\frac{1}{2}} \exp\left\{\frac{-(y_i-\mu)^2}{2\sigma^2}\right\} \propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{\frac{-\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right\}$$

joint posterior distribution for μ and σ^2

$$\pi(\mu,\sigma^2|y) = \frac{f(\mu,\sigma^2,y)}{f(y)} = \frac{f(y|\mu,\sigma^2)\pi(\mu,\sigma^2)}{f(y)} \propto f(y|\mu,\sigma^2)\pi(\mu,\sigma^2)$$

$$\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right\} \frac{e^{-\frac{(\mu-\theta_0)}{s_0}}}{\left[1+e^{-\frac{(\mu-\theta_0)}{s_0}}\right]^2} \times \frac{1}{\sigma^2} \qquad \sigma^2>0,\ -\infty<\mu<\infty$$

(b) full. conditional distribution for μ

$$\pi(\mu|\sigma^2,y) \propto \exp\left\{-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right\} \frac{e^{-\frac{(\mu-\theta_0)}{s_0}}}{\left[1+e^{-\frac{(\mu-\theta_0)}{s_0}}\right]^2} \propto \exp\left\{\frac{-(\mu-\bar{y})^2}{2\frac{\sigma^2}{n}}\right\} \frac{e^{-\frac{(\mu-\theta_0)}{s_0}}}{\left[1+e^{-\frac{(\mu-\theta_0)}{s_0}}\right]^2}$$

$$\pi(\sigma^2|\mu,y) \propto (\sigma^2)^{-(\frac{n}{2}+1)} \exp\left\{-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right\} \Rightarrow \text{Inv Gamma}\left(\frac{n}{2}, \frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2}\right)$$

(c)

Step1 Set $t=0$, $\mu_0=0$ and $\sigma_0^2=1$ as inital state.

Step 2. Set $t=t+1$, 先抽 $\mu_{t+1} \sim \pi(\mu|\sigma_t^2, y)$

Step 3 再抽 $\sigma_{t+1}^2 \sim \pi(\sigma^2|\mu_{t+1},y) \equiv \text{Inv Gamma}\left(\frac{n}{2}, \frac{\sum_{i=1}^{n}(y_i-\mu_{t+1})^2}{2}\right)$

Step 4 repeat 2,3

(d) Yes, Step 2 need the Metropolis-Hastings algorithm

Step1 Set $t=0$ $\quad \mu_0=0$ $\quad \sigma_0^2=1$ as initial state

Step2 Set Proposal distribution $q(\mu^*|\mu_t)$

$$\mu^* \sim q(\mu^*|\mu_t)$$

Step 3 calculate $\alpha = \min\left\{1, \frac{P(\mu^*)\,q(\mu_t|\mu^*)}{P(\mu_t)\,q(\mu^*|\mu_t)}\right\}$

Step 4 Generate $U \sim \text{Unif}(0,1)$

$$\mu_{t+1} \begin{cases} \mu^* & \text{if } U \le \alpha \\ \mu_t & \text{if } U > \alpha \end{cases}$$

Step 5 Repeat Step 2 to Step 4

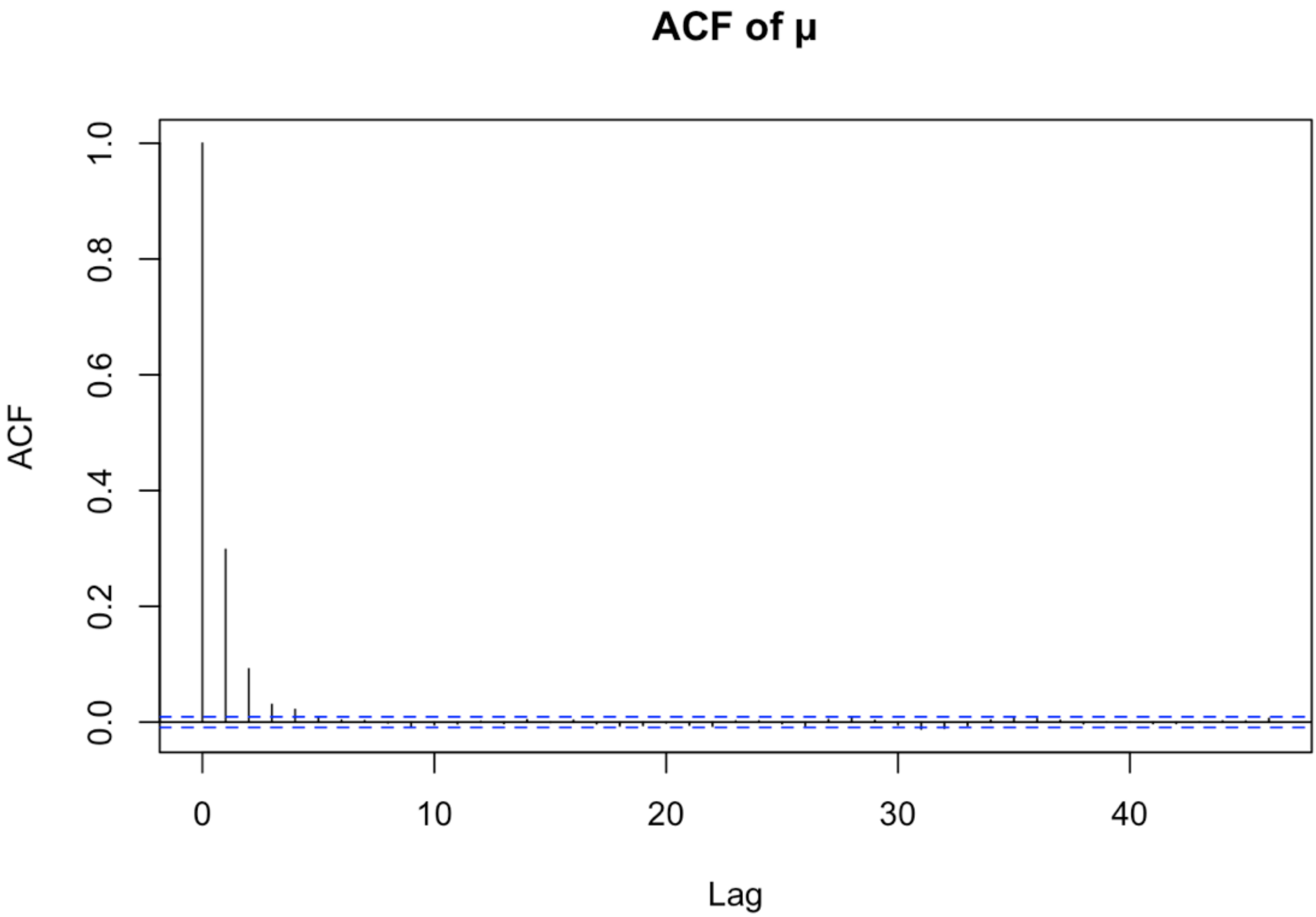
(e)

proposal distribution: $\mu^* \sim N(\bar{y}, \frac{\sigma_t^2}{12})$

```
data.y<- c(14.52, 8.49, 11.86, 6.42, 8.41, 7.66, 10.4, 9.99, 16.49,
              6.55, 16.54, 15.53, 5.66, 14.67, 9.06, 13.69, 8.49, 12.72,
              7.86, 13.03, 13.06, 5.67, 8.18, 18.74, 7.63, 14.76, 18.28,
              15.82, 12.67, 11.72, 16.13, 11.5, 11.88, 9.3, 12.67, 10.61,
              12.35, 8.41, 11.17, 14.91, 5.58, 7.74, 12.78, 11.32, 11.12,
              12.01, 13.75, 11.36, 11.63, 10.22)
n=length(data.y)


sim_1=function(freq,data.y,theta0,s0,d){
  par=matrix(rep(NA,freq*2),nrow=2)
  acc.p=rep(NA,freq)
  par[,1]=c(mean(data.y),var(data.y))
  for(i in 2:freq){
    pro.mu=rnorm(1,mean(data.y),sqrt(par[2,i-1]/d))

    alpha1=dnorm(pro.mu,mean(data.y),sqrt(par[2,i-1]/n))/dnorm(par[1,i-1],mean(dat
a.y),sqrt(par[2,i-1]/n))*dlogis(pro.mu,theta0,s0)/dlogis(par[1,i-1],theta0,s0)*dno
rm(par[1,i-1],mean(data.y),sqrt(par[2,i-1]/d))/dnorm(pro.mu,mean(data.y),sqrt(par[
2,i-1]/d))
    alpha=min(c(1,alpha1))

    U=runif(1)
    par[1,i]=ifelse(U<alpha,pro.mu,par[1,i-1])
    par[2,i]=1/rgamma(1,n/2,sum((data.y-par[1,i])^2)/2)
    acc.p[i]=ifelse(pro.mu==par[1,i],1,0)
  }
  return(res=list(par=par,acc.p=acc.p))
}

freq=50000
theta0 = 11
s0 = 2.5
draw=seq(0.1*freq,freq)
res=sim_1(freq,data.y,theta0,s0,12)
par=res$par
acc.p=res$acc.p
```

ACF plots

```
acf(par[1,draw],main="ACF of μ")
```



只有前五步顯著。 抽的還不錯

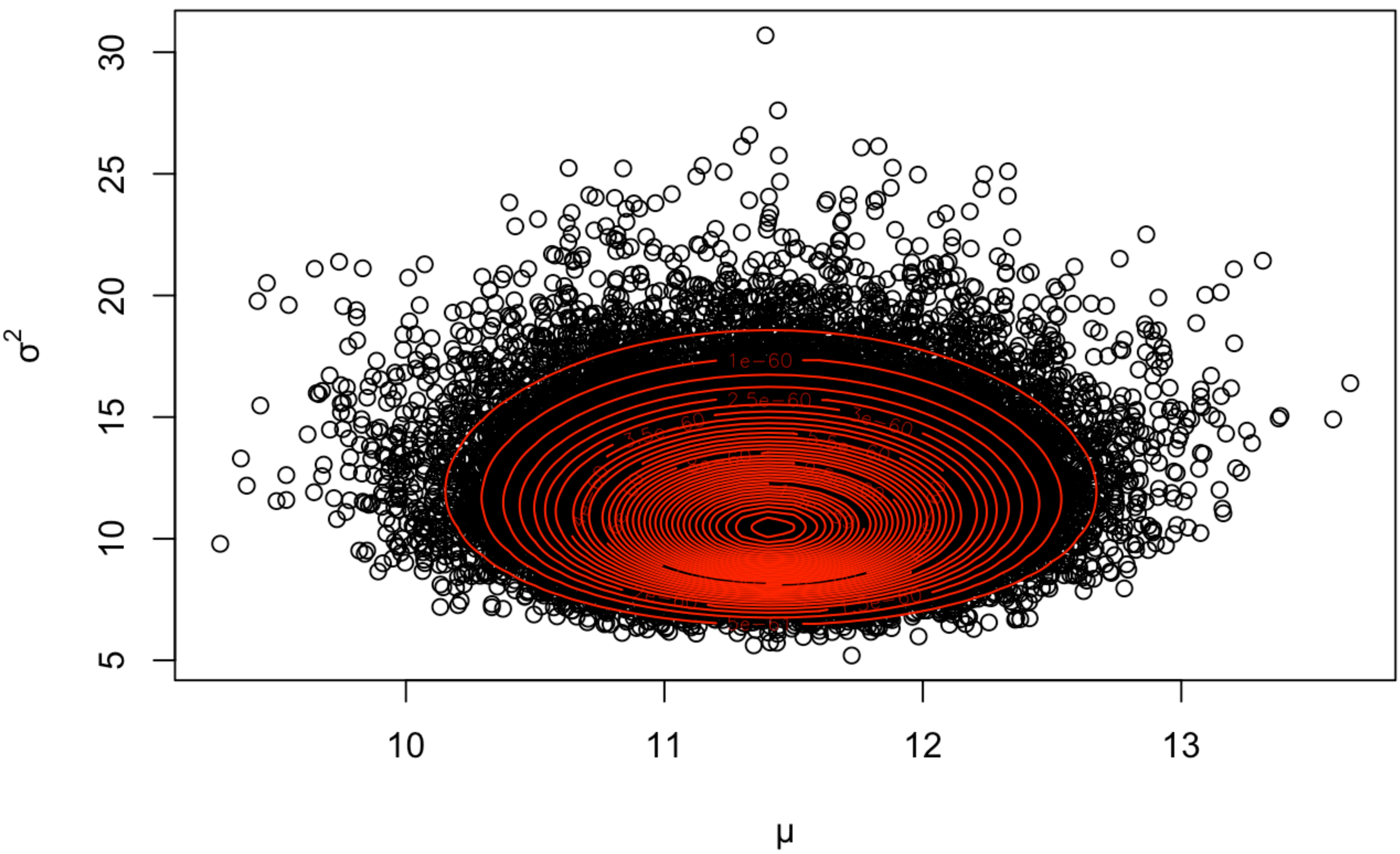
average rate of acceptance

```
acc=mean(acc.p[draw],na.rm = T)
```

average rate of accepatnce is 0.5779205. Between 0.4 and 0.6. It is good.

(f)

```
pdf.f=function(mu,sigma_square,theta0,s0){
  fxy=matrix(NA, length(mu), length(sigma_square))
  for(i in 1:length(mu)){
    for(j in 1:length(sigma_square)){
      fxy[i,j]=(2*3.14*sigma_square[j])^(-n/2)*exp(-sum((data.y-mu[i])^2)/(2*sigma
_square[j]))*exp(-(mu[i]-theta0)/s0)/(1+exp(-(mu[i]-theta0)/s0))^2/s0/sigma_square
[j]
    }
  }
  return(fxy)
}

mu=seq(9,15,0.1)
sigma_square=seq(1,30,0.1)

plot(par[1,draw],par[2,draw],xlab="μ",ylab=expression(σ^2)) #較好
contour(mu,sigma_square,pdf.f(mu,sigma_square,theta0,s0),nlevels=30,col="red",add=
T)
```



(g) posterior mean for μ and σ2

```
library(knitr)
kable(cbind(mean(par[1,]),mean(par[2,])),col.names=c('μ', 'σ2'))
```

μ	σ2

11.41027 11.57604

2

(a)

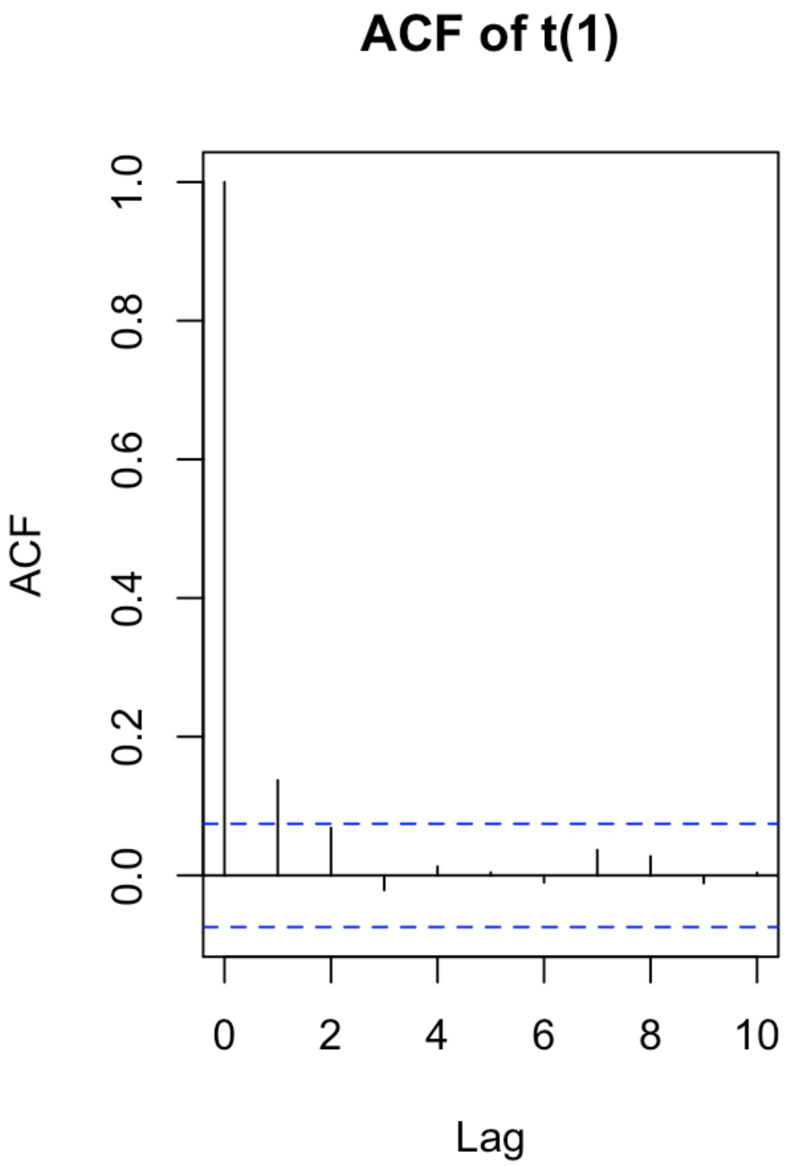
proposal distribution: $x^{*} \sim N(x_t, \sigma^2)$

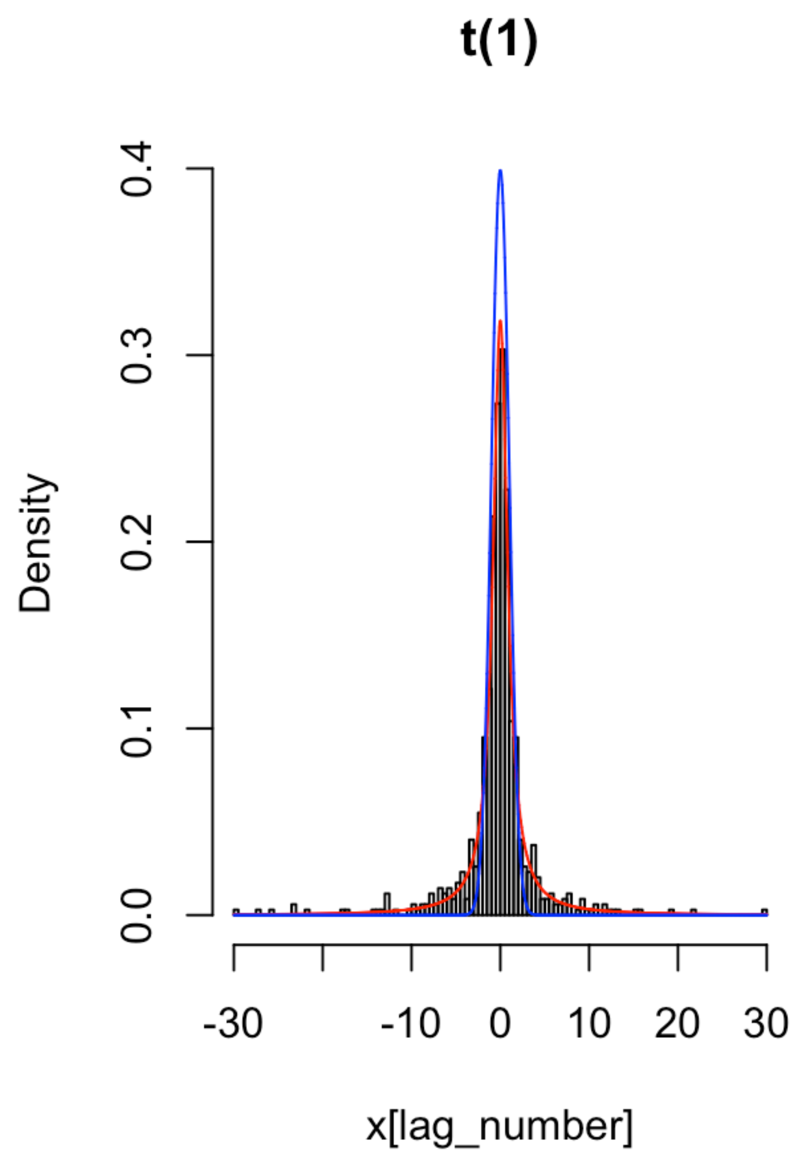
(b)(c)

```
sim_2.1=function(freq,p.sigma,df){
  x=rep(NA,freq)
  acc.p=rep(NA,freq)
  x[1]=0
  for(i in 2:freq){
    pro.x=rnorm(1,x[i-1], p.sigma)
    alpha1=((1+pro.x^2/df)/(1+x[i-1]^2/df))^(-(df+1)/2)
    alpha=min(c(1,alpha1))
    U=runif(1)
    x[i]=ifelse(U<alpha,pro.x,x[i-1])
    acc.p[i]=ifelse(x[i] == pro.x, 1, 0)
  }
  return(res=list(x=x,acc.p=acc.p))
}
```

df=1 choose lag=65

```
set.seed(4)
freq=50000
p.sigma=3
lag_count=65
df=1


res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p


lag_number=seq(0.1*freq,freq,lag_count)
par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```



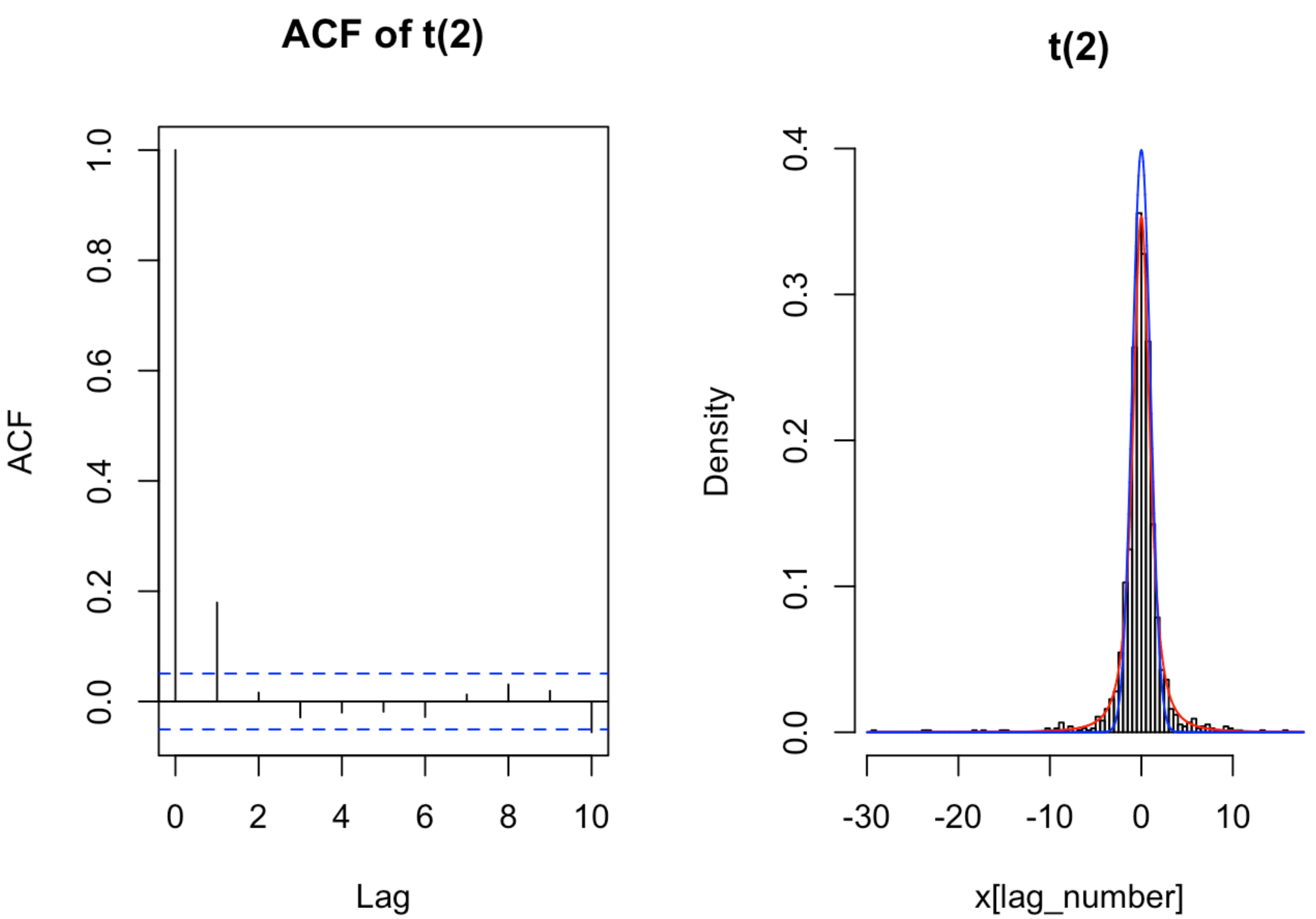



```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.5243217
```

df=2 choose lag=30

```
set.seed(2)
freq=50000
p.sigma=2.5
lag_count=30
df=2


res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p


lag_number=seq(0.1*freq,freq,lag_count)
par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```

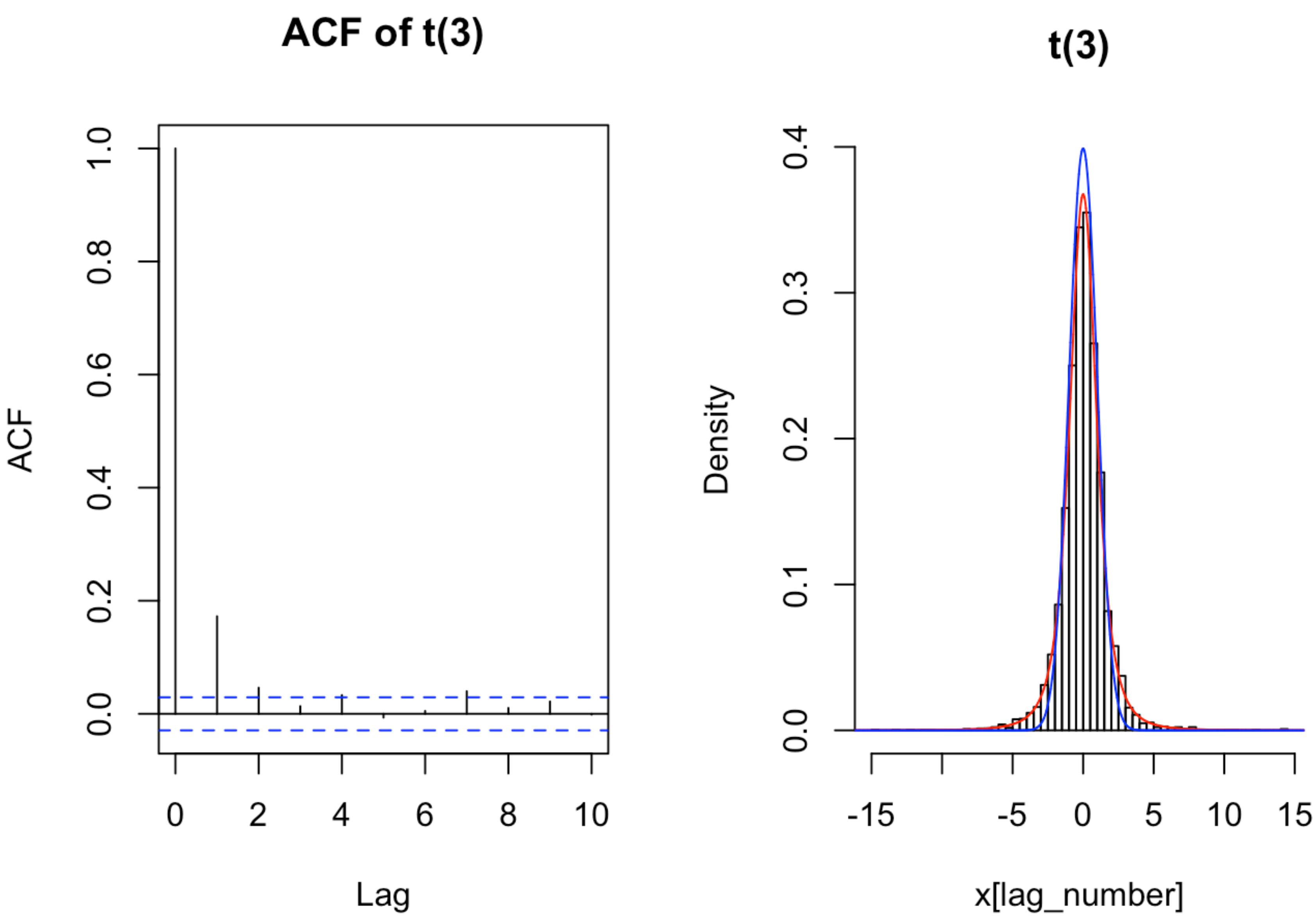



```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.5132553
```

df=3 choose lag=10

```
freq=50000
p.sigma=2.5
lag_count=10
df=3


res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p


lag_number=seq(0.1*freq,freq,lag_count)

par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```

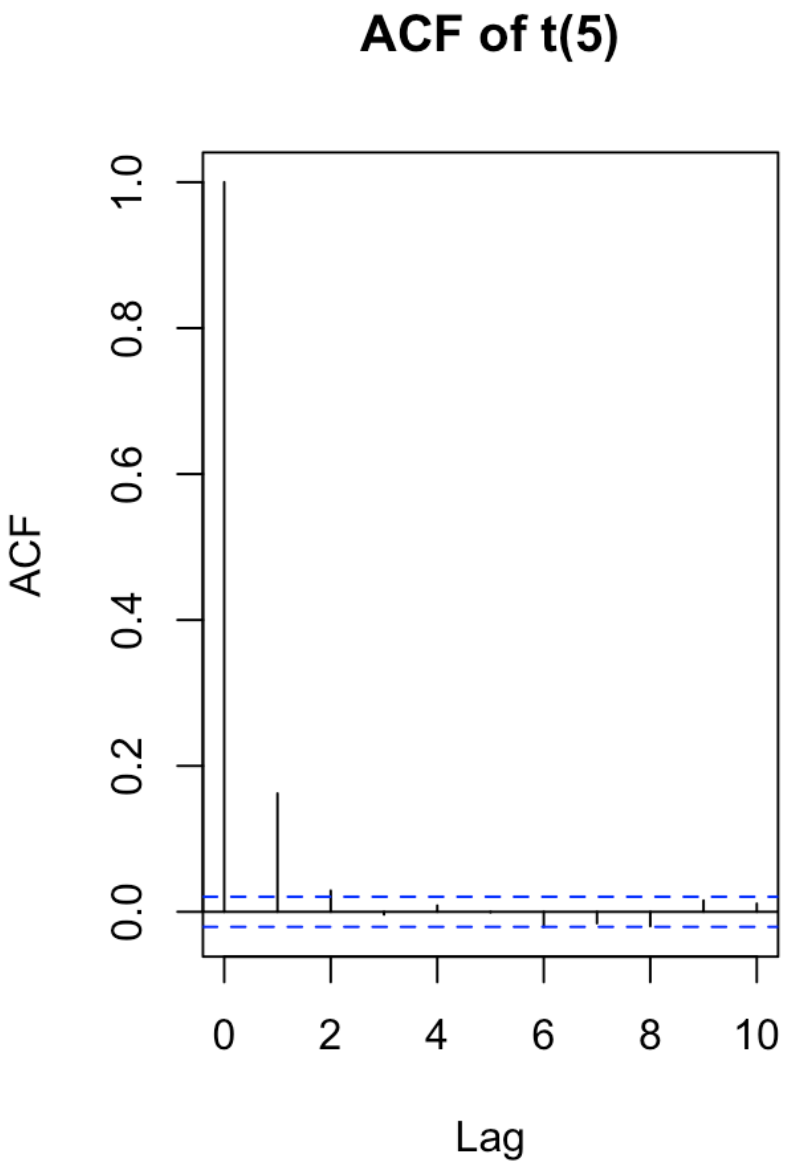


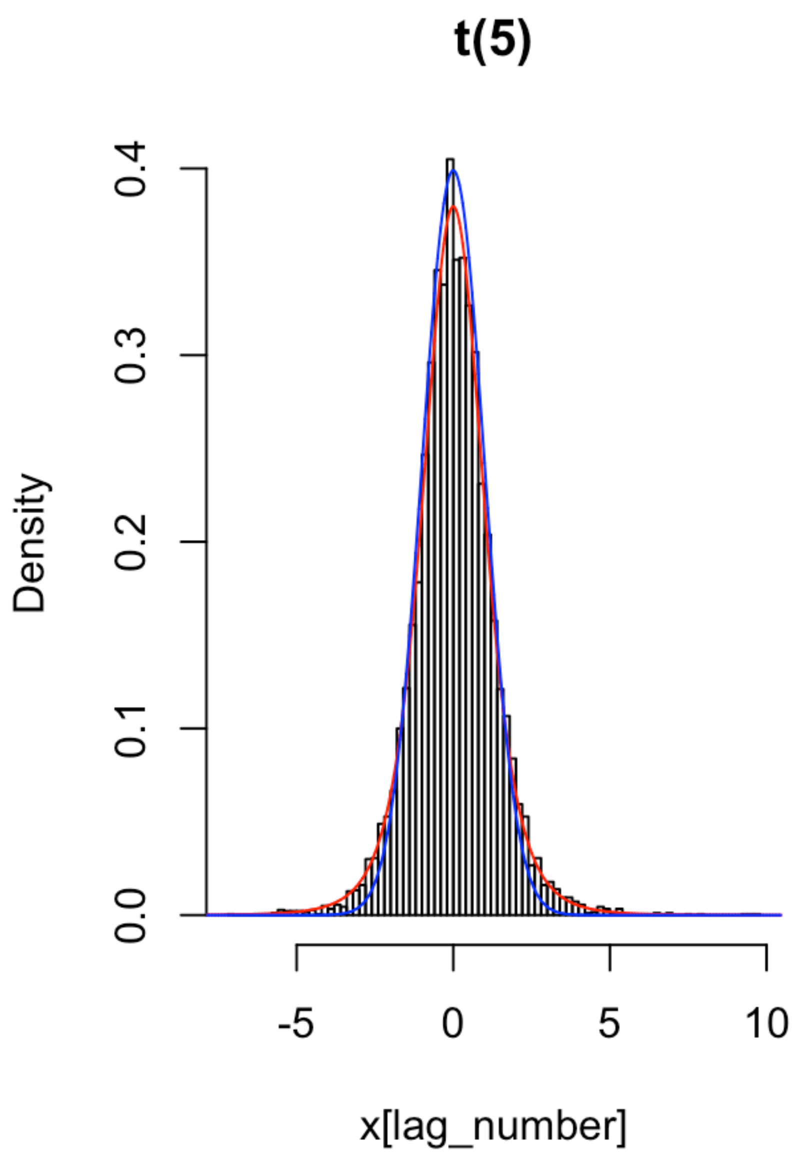
```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.4858336
```

df=5 choose lag=5

```
freq=50000
p.sigma=2.5
lag_count=5
df=5



res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p



lag_number=seq(0.1*freq,freq,lag_count)

par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```



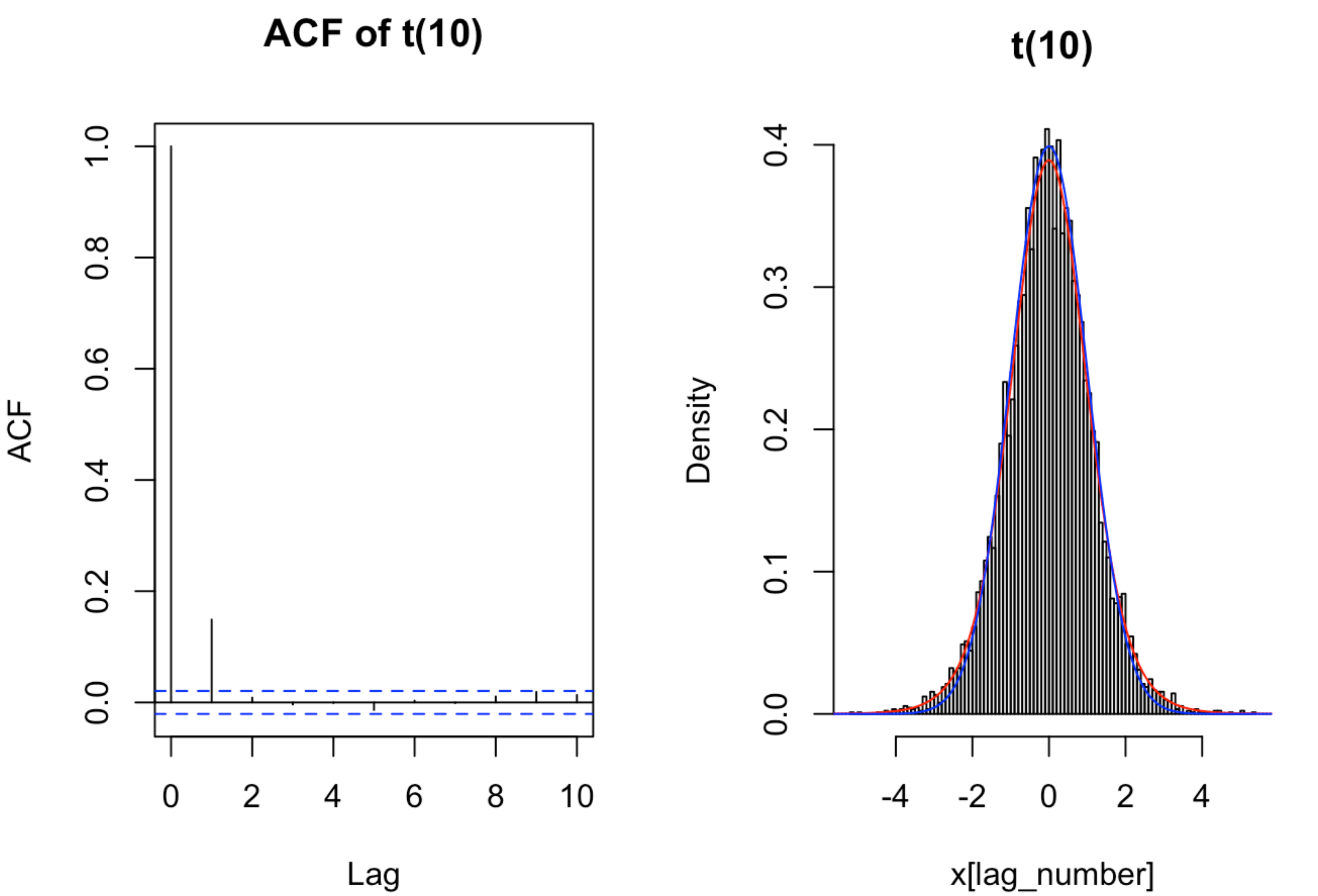



```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.4664785
```

df=10 choose lag=5

```
freq=50000
p.sigma=2
lag_count=5
df=10


res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p


lag_number=seq(0.1*freq,freq,lag_count)

par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```

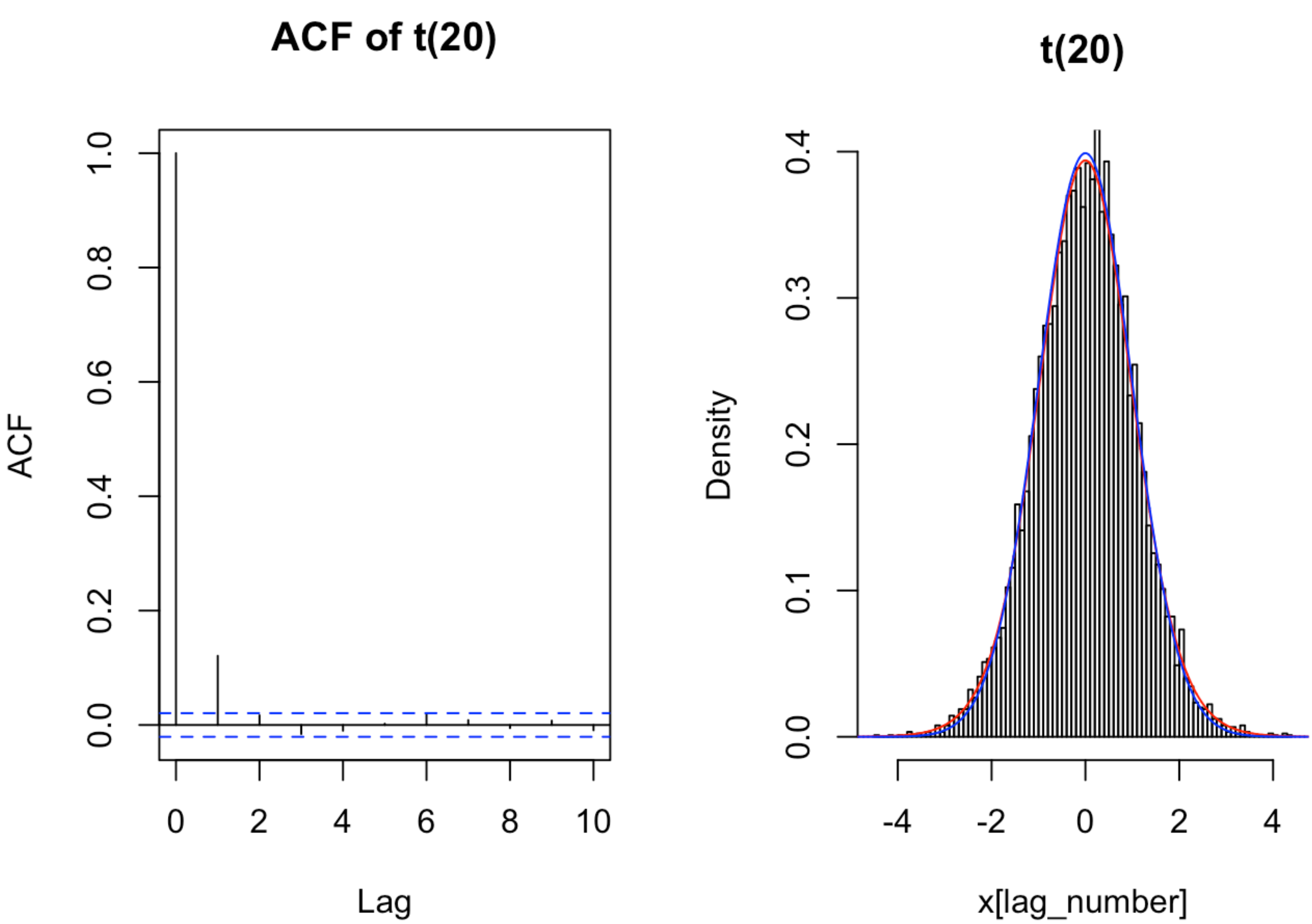



```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.515633
```

df=20 choose lag=5

```
freq=50000
p.sigma=2
lag_count=5
df=20



res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p



lag_number=seq(0.1*freq,freq,lag_count)

par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```

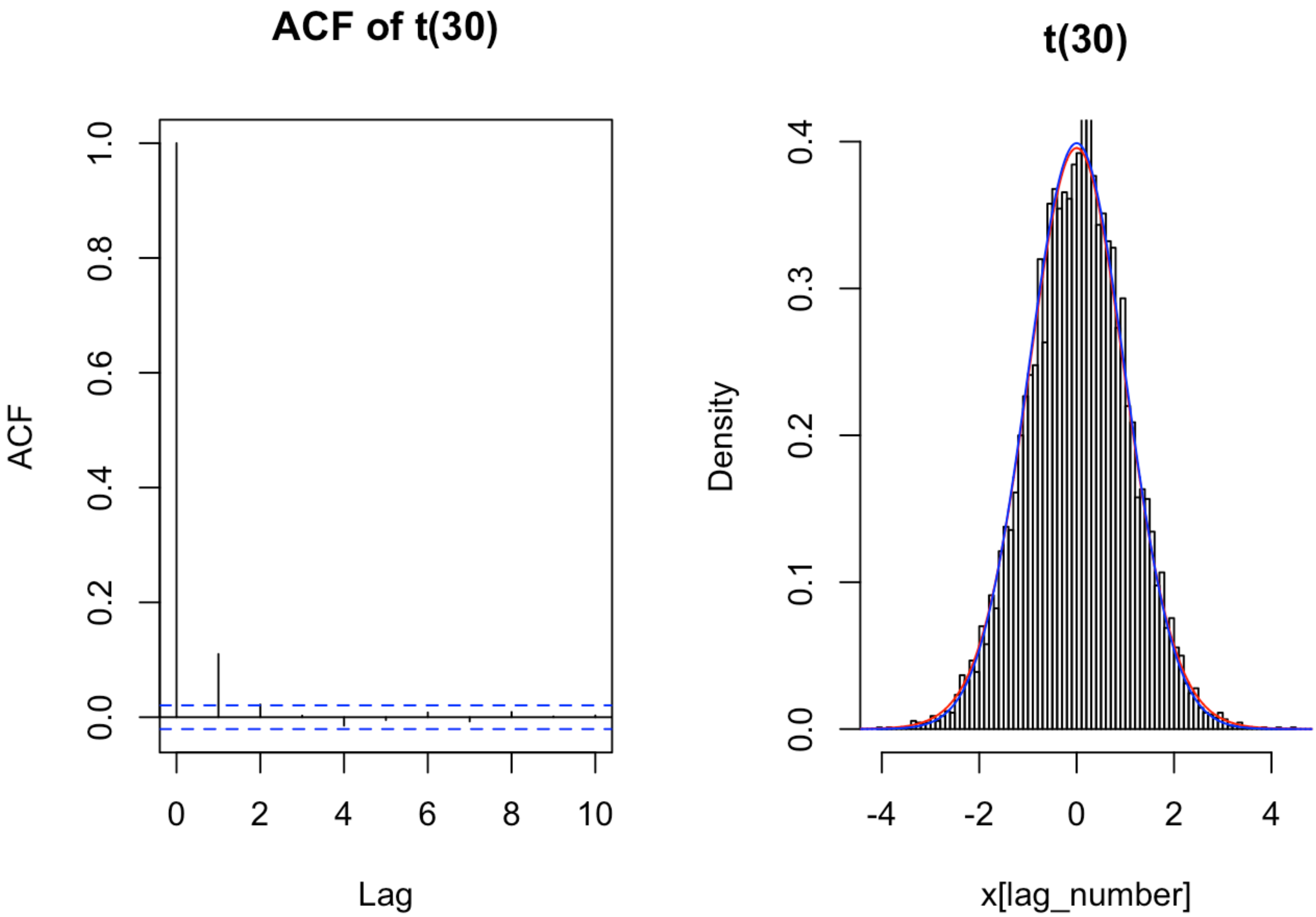


```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.5075887
```

df=30 choose lag=5

```
freq=50000
p.sigma=2
lag_count=5
df=30



res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p



lag_number=seq(0.1*freq,freq,lag_count)

par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```

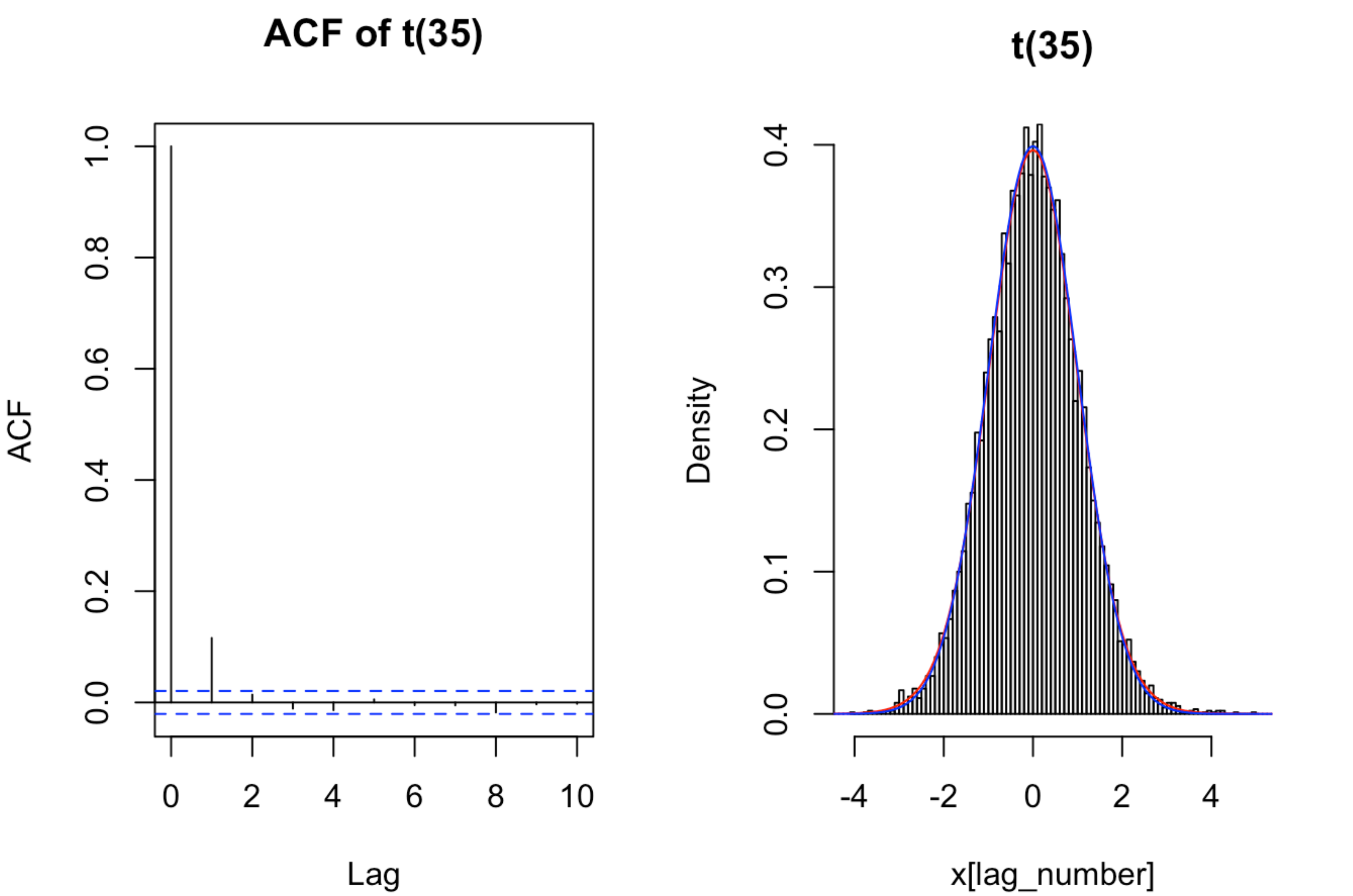



```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.5057665
```

df=35 choose lag=5

```
freq=50000
p.sigma=2
lag_count=5
df=35


res=sim_2.1(freq,p.sigma,df)
x=res$x
acc.p=res$acc.p


lag_number=seq(0.1*freq,freq,lag_count)

par(mfrow=c(1,2))
acf(x[lag_number],lag=10,main=paste0("ACF of t(",df,")"))

hist(x[lag_number],probability  =T,breaks = 100,ylim=c(0,dnorm(0,0,1)),main=paste0
("t(",df,")"))
xx=seq(-30,30,0.001)
lines(xx,dt(xx,df),col="red")
lines(xx,dnorm(xx,0,1),col="blue")
```




```
mean(acc.p[(0.1*freq):freq],na.rm=T)
```

```
## [1] 0.5010778
```

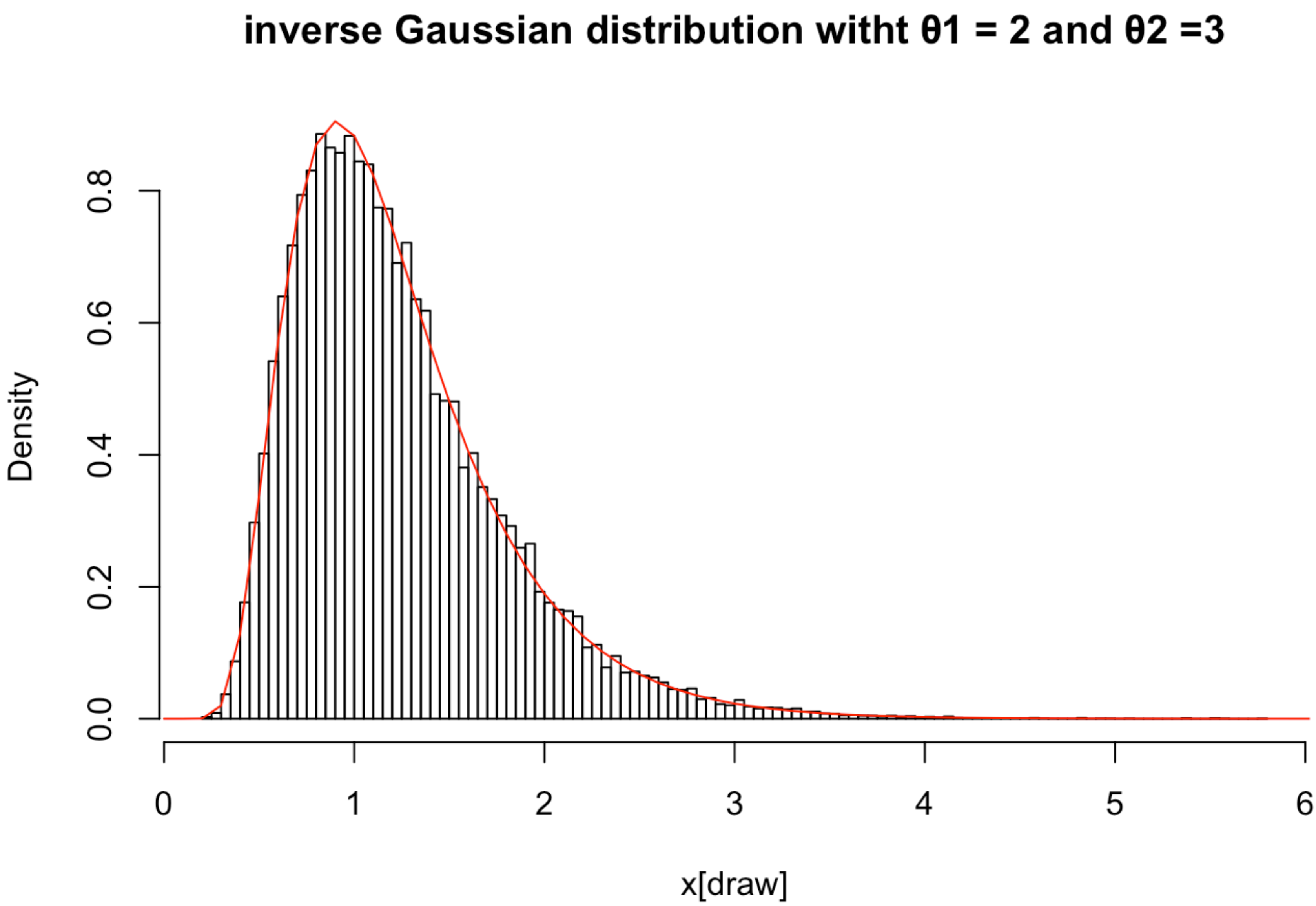
(d)

With degrees of freedom=30,35,the t-distribution is approaching to the standard normal distribution.

3

(a)

```
inverse_Gaussian_pdf=function(x,theta1,theta2){
  x^(-3/2)*exp(-theta1*x-theta2/x+2*sqrt(theta1*theta2)+log(sqrt(2*theta2)))
}

sim_3=function(freq,theta1,theta2,a,b){
  x=rep(NA,freq)
  acc.p=rep(NA,freq)
  x[1]=2
  for(i in 2:freq){
    pro.x=rgamma(1,a,b)
    alpha1=inverse_Gaussian_pdf(pro.x,theta1,theta2)/inverse_Gaussian_pdf(x[i-1],t
heta1,theta2)*dgamma(x[i-1],a,b)/dgamma(pro.x,a,b)
    alpha=min(c(1,alpha1))
    U=runif(1)
    x[i]=ifelse(U<alpha,pro.x,x[i-1])
    acc.p[i]=ifelse(x[i] == pro.x, 1, 0)
  }
  return(res=list(x=x,acc.p=acc.p))
}

library(statmod)
freq=50000
theta1=2
theta2=3
a=2
b=1.8

res=sim_3(freq,theta1,theta2,a,b)
x=res$x
acc.p=res$acc.p

draw=seq(0.1*freq,freq)

hist(x[draw],probability = T,breaks=100,main="inverse Gaussian distribution witht
θ1 = 2 and θ2 =3")
xx=seq(0,30,0.1)
lines(xx,dinvgauss(xx,sqrt(theta2/theta1),2*theta2),col="red")
```

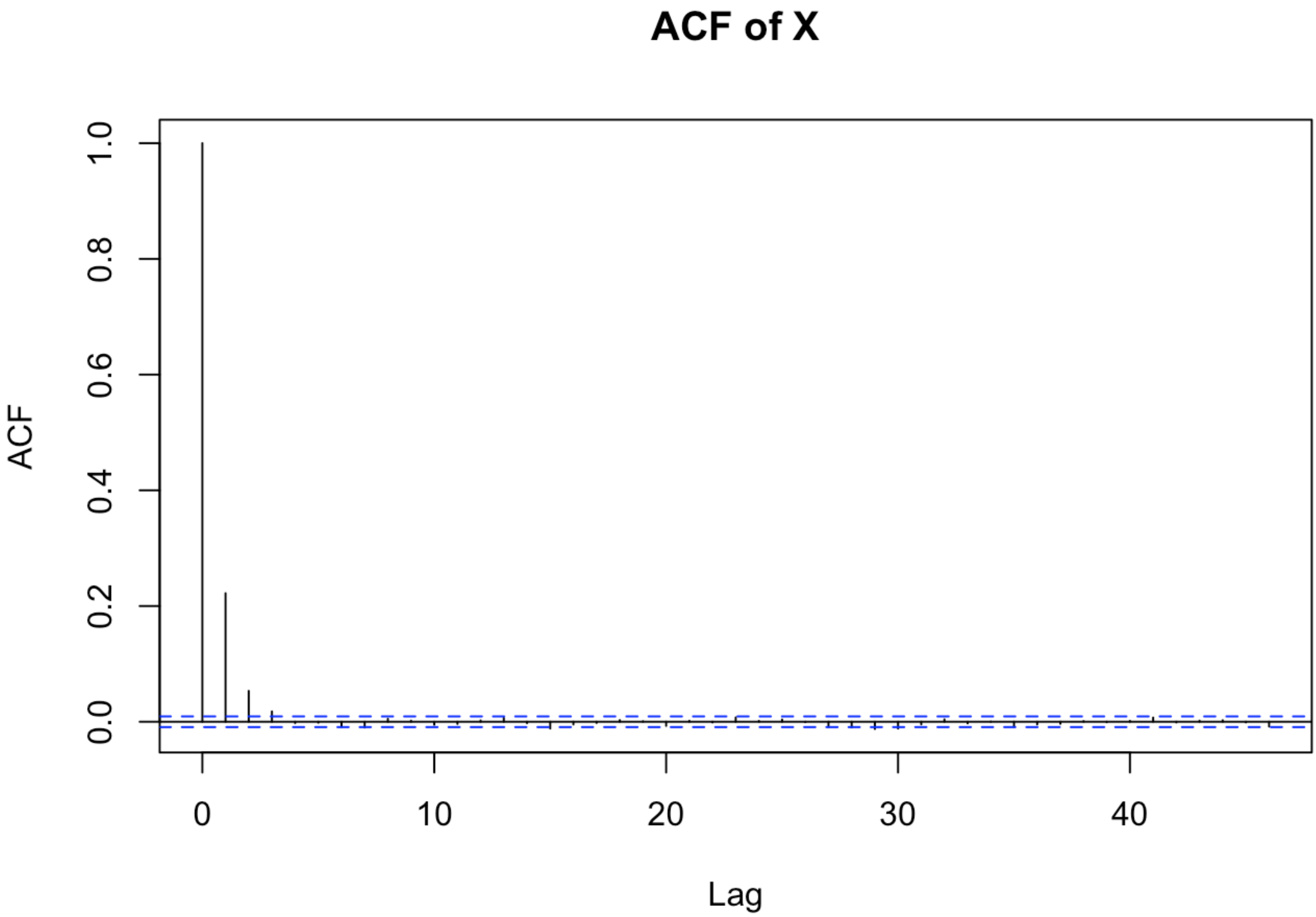


(b)

parameters used in the proposal distribution:x^* ~$Gamma(2,1.8)

ACF plots

```
acf(x[draw],main="ACF of X")
```



average rate of acceptance

```
mean(acc.p[draw],na.rm=T)
```

```
## [1] 0.6739184
```

(c)

```
kable(cbind(mean(x[draw]),sqrt(theta2/theta1)),col.names=c('E(X) simulation', 'E(X
) exact'))
```

E(X) simulation	E(X) exact
1.22909	1.224745

```
kable(cbind(mean(1/x[draw]),sqrt(theta1/theta2)+1/(2*theta2)),col.names=c('E(1/X)
simulation', 'E(1/X) exact'))
```

E(1/X) simulation	E(1/X) exact
0.9822649	0.9831632

The simulation values are very close to the exact values.

4

(a)

4

$$f_{X,Y}(x,y) = \frac{2}{(1-e^{-x-y})^3} e^{-x}(1-e^{-x})\, e^{-y}(1-e^{-y}) \qquad x>0 \quad y>0$$

$$u = e^{-x} \qquad x = -\log u$$
$$v = e^{-y} \qquad y = -\log v$$

$$J = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{vmatrix} = \begin{vmatrix} \frac{-1}{u} & 0 \\ 0 & -\frac{1}{v} \end{vmatrix} = \frac{1}{uv}$$

$$\infty > x > 0 \Rightarrow -\log u > 0 \Rightarrow 0<u<1$$
$$\infty > y > 0 \Rightarrow -\log v > 0 \Rightarrow 0<v<1$$

$$f_{u,v}(u,v) = f_{X,Y}(x,y)\,|J| = f(-\log u, -\log v)\,\frac{1}{uv}$$

$$= \frac{2}{(1-e^{\log u+\log v})^3} e^{\log u}(1-e^{\log u})\, e^{\log v}(1-e^{\log v})\,\frac{1}{uv}$$

$$= \frac{2}{(1-uv)^3}\cancel{u}(1-u)\cancel{v}(1-v)\,\frac{\cancel{1}}{uv}$$

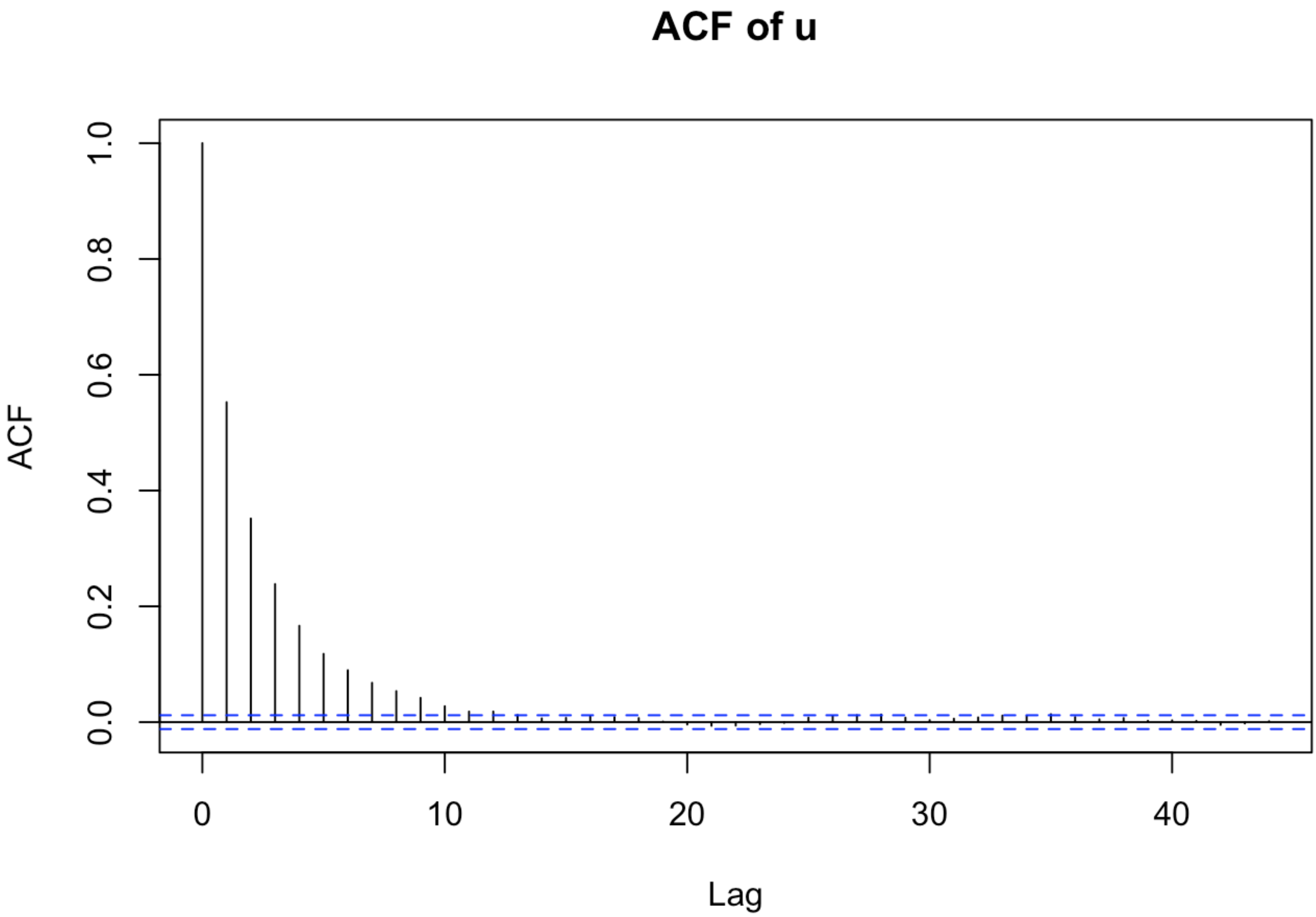
$$= 2(1-u)(1-v)(1-uv)^{-3} \qquad \begin{matrix} 0<u<1 \\ 0<v<1 \end{matrix}$$

(b)

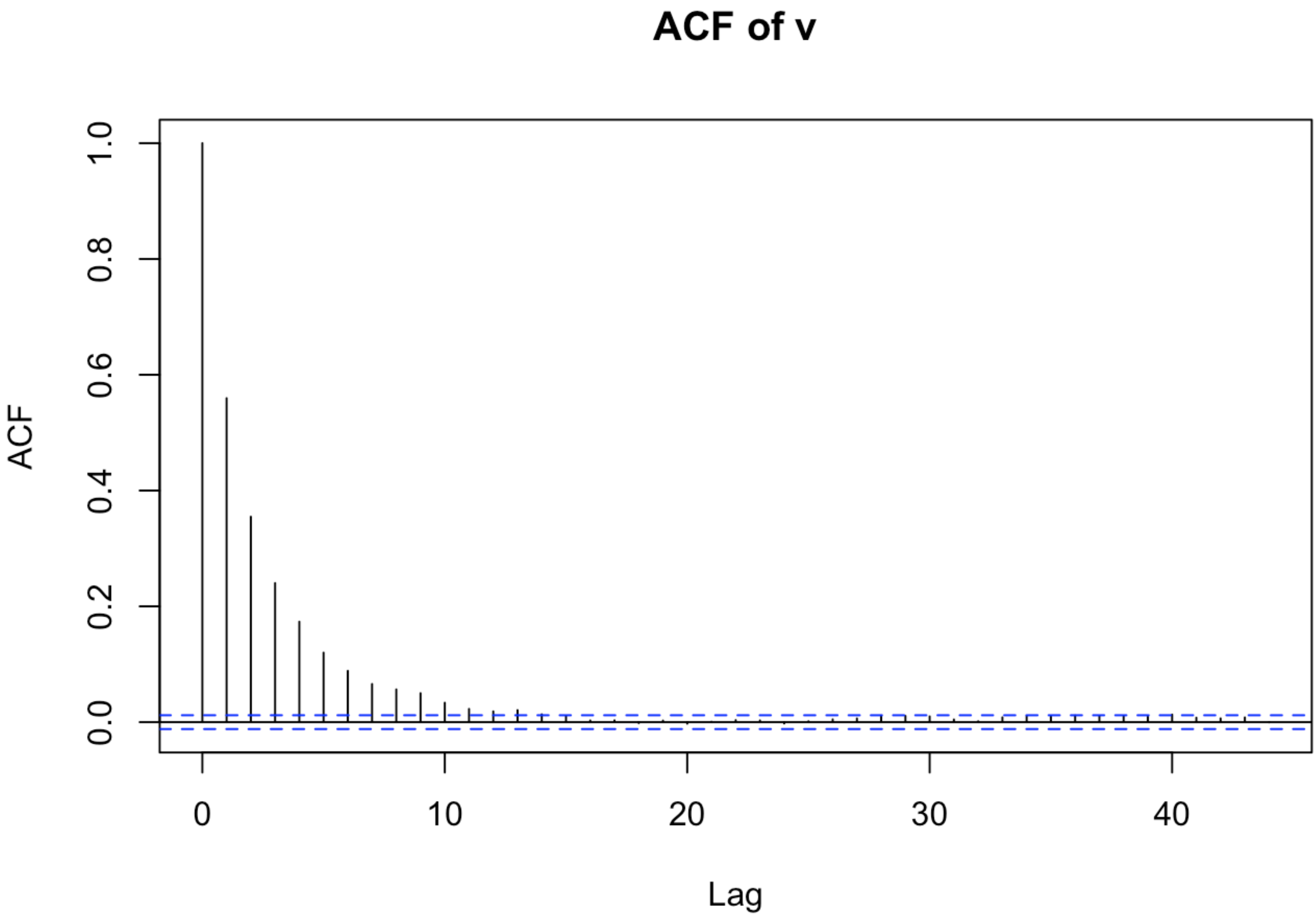
```
pdf.u=function(u,v){
  (1-u)*(1-u*v)^(-3)
}
pdf.v=function(u,v){
  (1-v)*(1-u*v)^(-3)
}

sim_4=function(freq,a,b){
  par=matrix(rep(NA,freq*2),nrow=2)
  acc.p=matrix(rep(NA,freq*2),nrow=2)

  par[,1]=c(0.5,0.5)
  for(i in 2:freq){
    pro.u=rbeta(1,a,b)

    alpha11=pdf.u(pro.u,par[2,i-1])/pdf.u(par[1,i-1],par[2,i-1])*dbeta(par[1,i-1],
a,b)/dbeta(pro.u,a,b)
    alpha1=min(c(1,alpha11))

    U=runif(1)
    par[1,i]=ifelse(U<alpha1,pro.u,par[1,i-1])
    acc.p[1,i]=ifelse(pro.u==par[1,i],1,0)


    pro.v=rbeta(1,a,b)

    alpha21=pdf.v(par[1,i],pro.v)/pdf.v(par[1,i],par[2,i-1])*dbeta(par[2,i-1],a,b)
/dbeta(pro.v,a,b)
    alpha2=min(c(1,alpha21))

    U=runif(1)
    par[2,i]=ifelse(U<alpha2,pro.v,par[2,i-1])
    acc.p[2,i]=ifelse(pro.v==par[2,i],1,0)


  }
  return(res=list(par=par,acc.p=acc.p))
}

freq=30000
a=0.4
b=0.4
res=sim_4(30000,a,b)

draw=seq(0.1*freq,freq)

u=res$par[1,draw=seq(0.1*freq,freq)]
v=res$par[2,draw=seq(0.1*freq,freq)]
```

ACF plots

```
acf(u,main="ACF of u")
```




```
acf(v,main="ACF of v")
```



average rate of acceptance

```
acc_u=mean(u,na.rm=T)
acc_v=mean(v,na.rm=T)
```

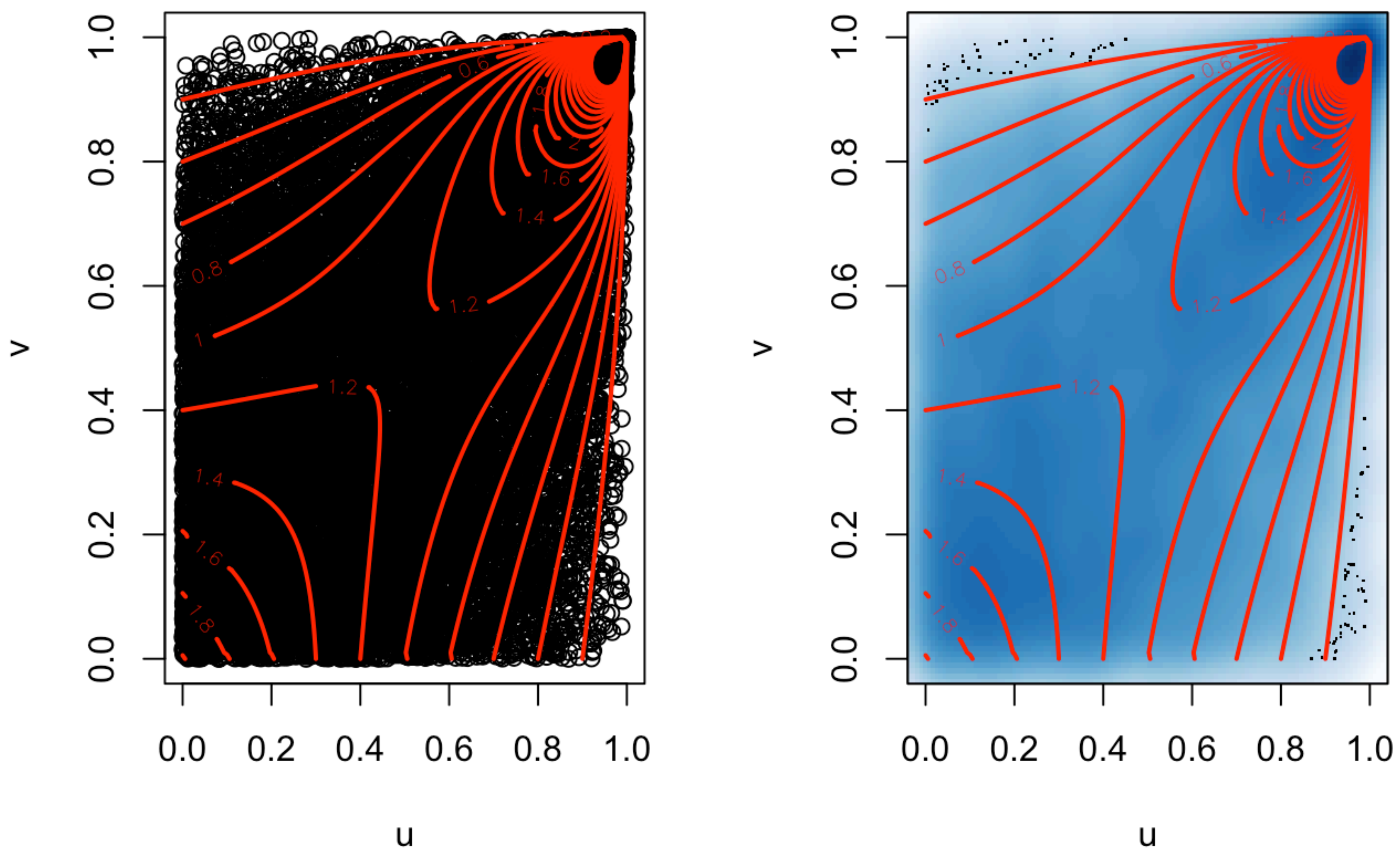
average rate of accepatnce of u is 0.4991328.

average rate of accepatnce of u is 0.4994168.

Between 0.4 and 0.6. It is good.

contour plot for x and y

```
pdf.uv=function(u,v){
  fuv=matrix(NA, length(u), length(v))
  for(i in 1:length(u)){
    for(j in 1:length(v)){
      fuv[i,j]=2*(1-u[i])*(1-v[j])*(1-u[i]*v[j])^(-3)
    }
  }
  return(fuv)
}

uu=seq(0,1,0.01)
vv=seq(0,1,0.01)
par(mfrow=c(1,2))
plot(u,v)
contour(uu,vv,pdf.uv(uu,vv),col="red",add=T,lwd=2,levels=seq(0,4,0.2))


smoothScatter(u,v)
contour(uu,vv,pdf.uv(uu,vv),col="red",add=T,lwd=2,levels=seq(0,4,0.2))
```

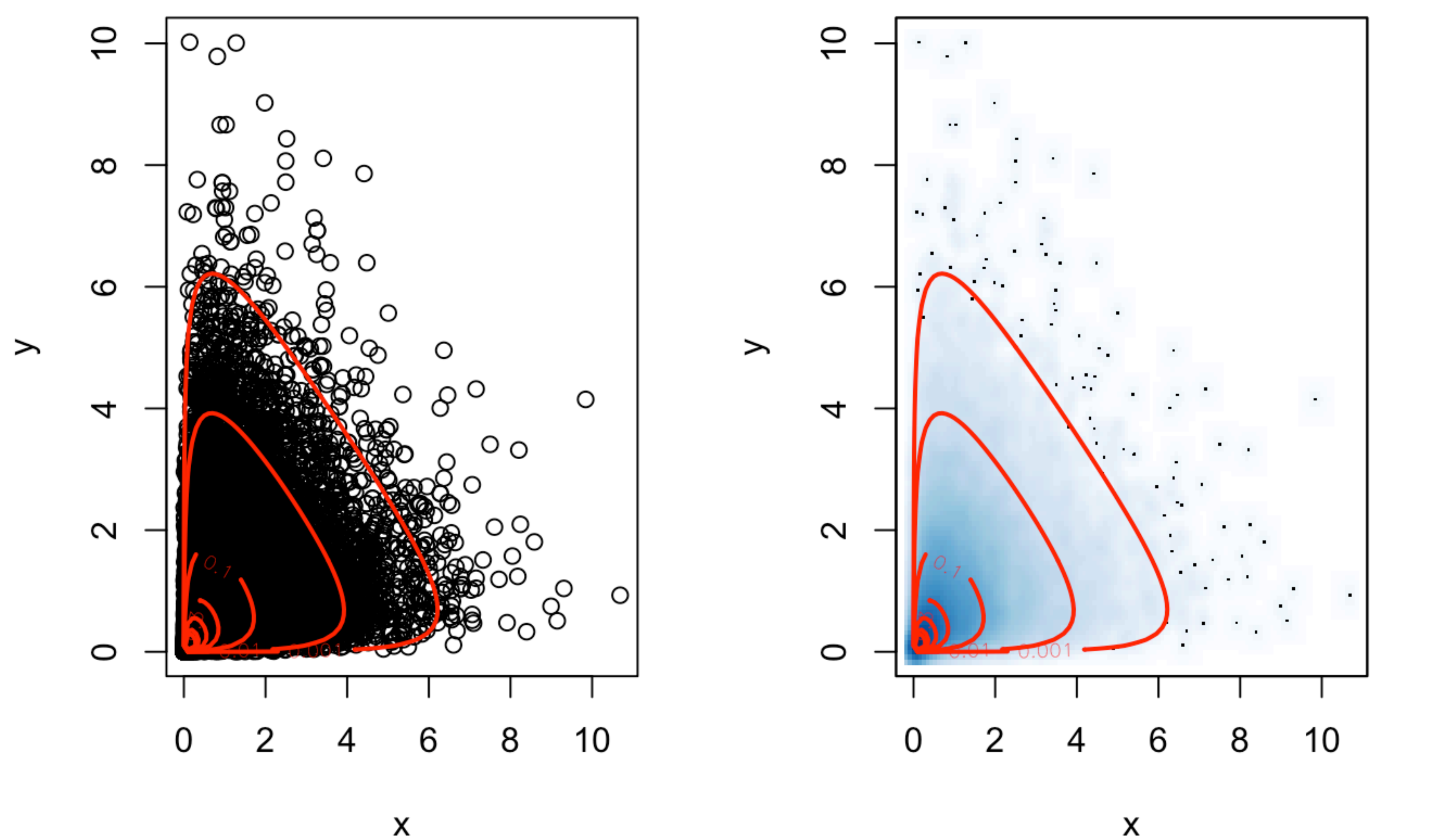


左上角區機率密度最大。

(c)

contour plot for x and y

```
x=-log(u)
y=-log(v)

pdf.xy=function(x,y){
  fxy=matrix(NA, length(x), length(y))
  for(i in 1:length(x)){
    for(j in 1:length(y)){
      fxy[i,j]=2*exp(-x[i])*(1-exp(-x[i]))*exp(-y[j])*(1-exp(-y[j]))*((1-exp(-x[i]
-y[j]))^(-3))
    }
  }
  return(fxy)
}

xx=seq(0,10,0.1)
yy=seq(0,10,0.1)
par(mfrow=c(1,2))
plot(x,y)
contour(xx,xx,pdf.xy(xx,yy),nlevels=20,add=T,col="red",lwd=2,levels=c(1,0.8,0.5,0.
3,0.1,0.01,0.001))

smoothScatter(x,y)
contour(xx,xx,pdf.xy(xx,yy),nlevels=20,add=T,col="red",lwd=2,levels=c(1,0.8,0.5,0.
3,0.1,0.01,0.001))
```



右下角區機率密度最大。

5

使用MCMC，需要花很多時間調proposal distribution 的參數，而檢驗MCMC選的參數好不好，可用ACF圖和average rate of acceptance來檢驗

1.ACF理想值是在第一步截斷，但有時很難找到好的proposal distribution，就可以用間隔幾步取一點，但相對的會損失樣本數，所以需抽更大的樣本數，會花費更多時間。

2.average rate of acceptance介在0.4～0.6會被認為proposal distribution 的參數選的好。

這兩種檢驗方法有時候會有一個判斷參數選的好，另一個判斷參數選的不好，這時我們需重新選擇參數，直到都通過兩個檢驗方法。