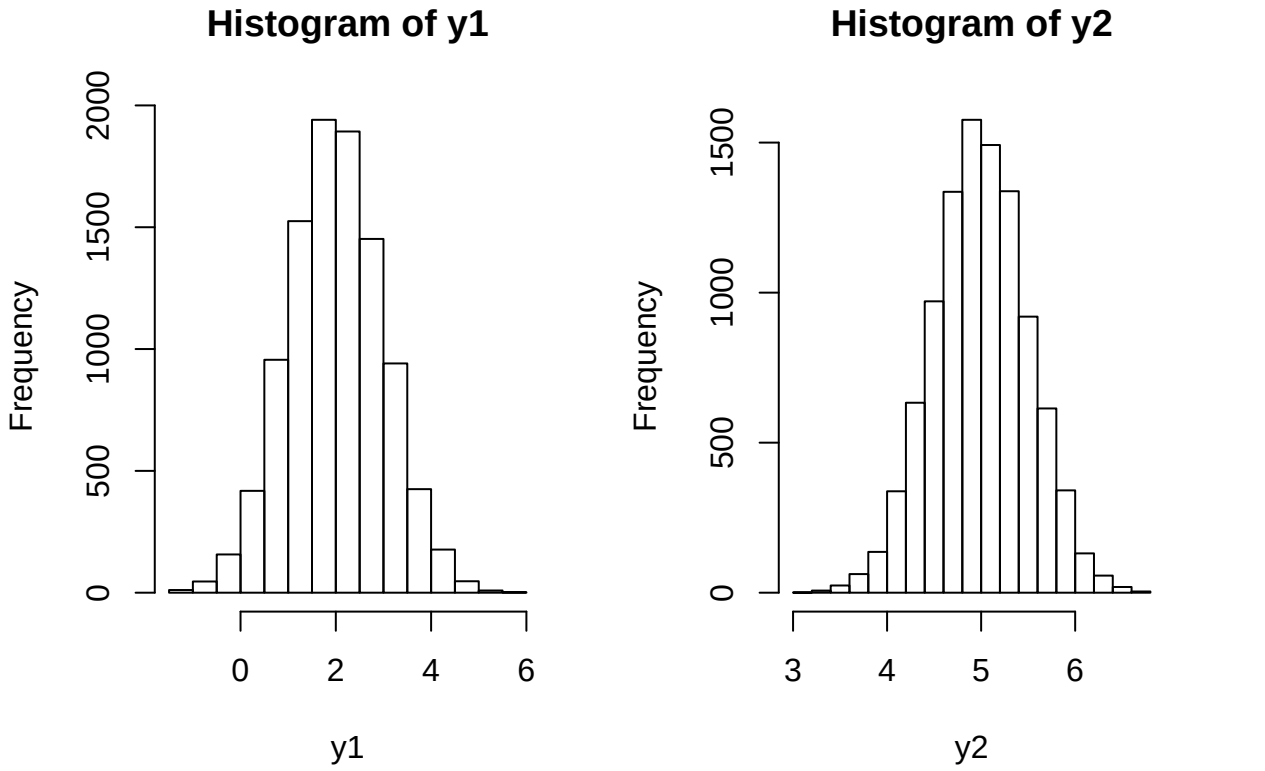
Markov Chain Monte Carlo

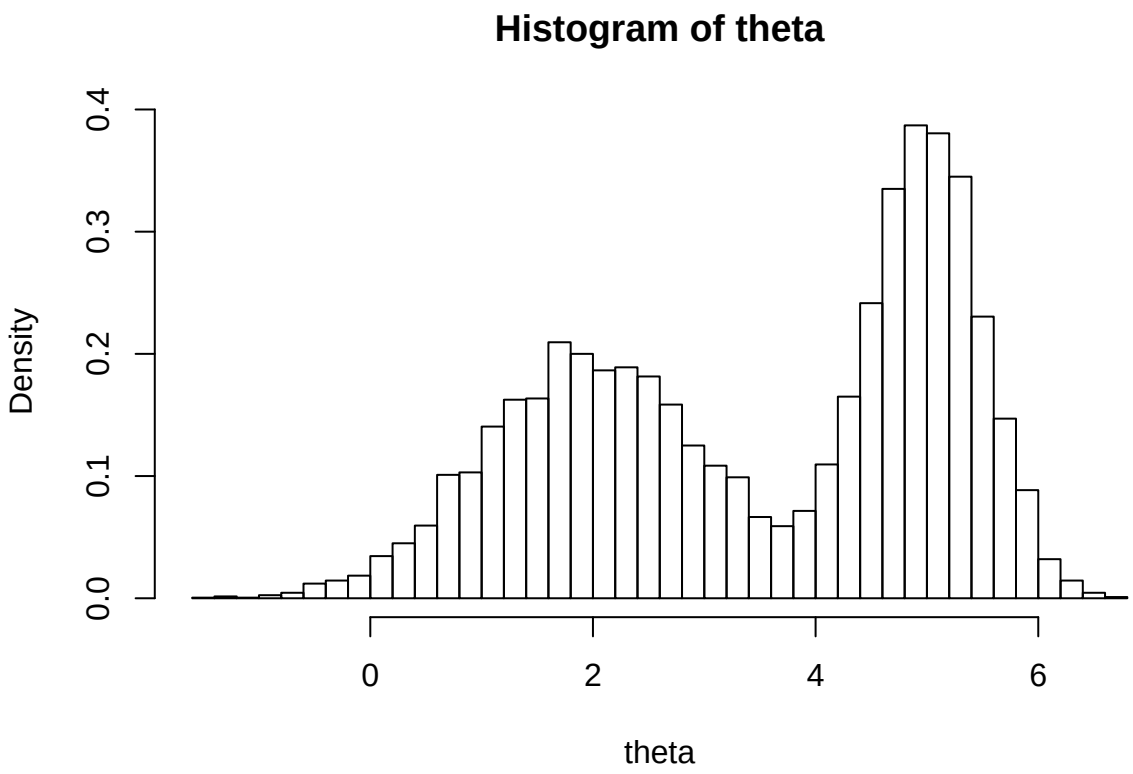
BIOS719 Generalized Linear Models

Mixture of two normal distributions

```
y1 <- rnorm(10000, mean=2, sd=1)
y2 <- rnorm(10000, mean=5, sd=0.5)
par(mfrow=c(1,2))
hist(y1)
hist(y2)
```

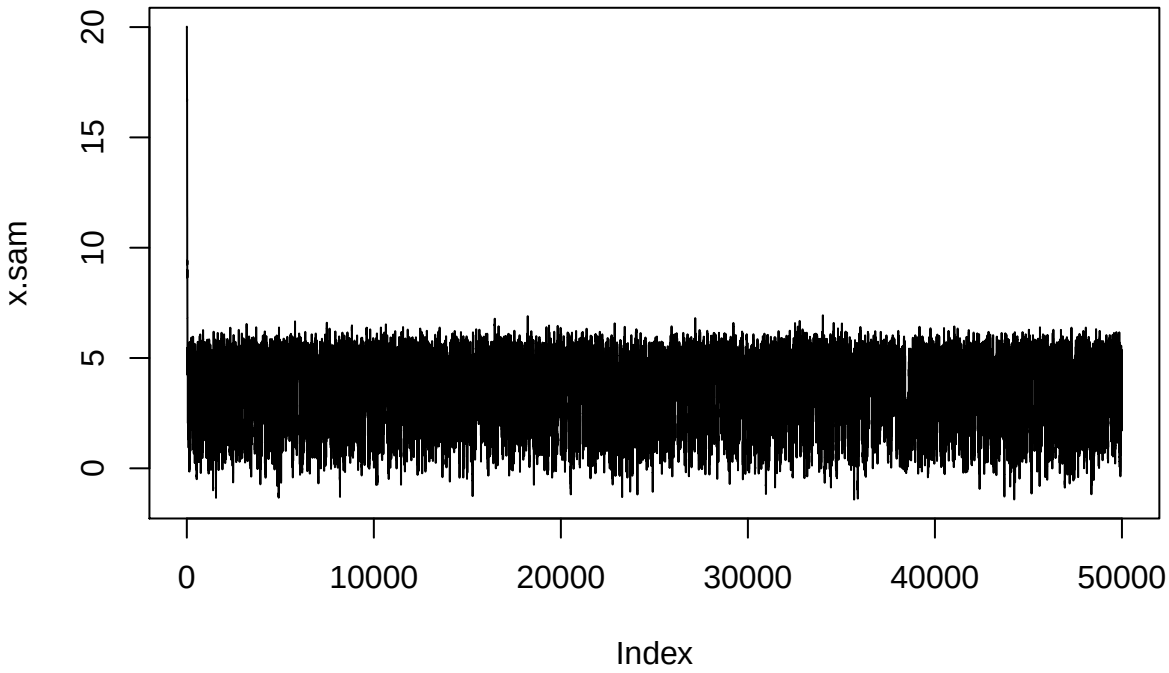



```
# theta = 0.5y1 + 0.5y2
theta <- ifelse(rbinom(10000, size=1, prob=0.5), y1, y2)
par(mfrow=c(1,1))
hist(theta, breaks=50, prob=T)
```

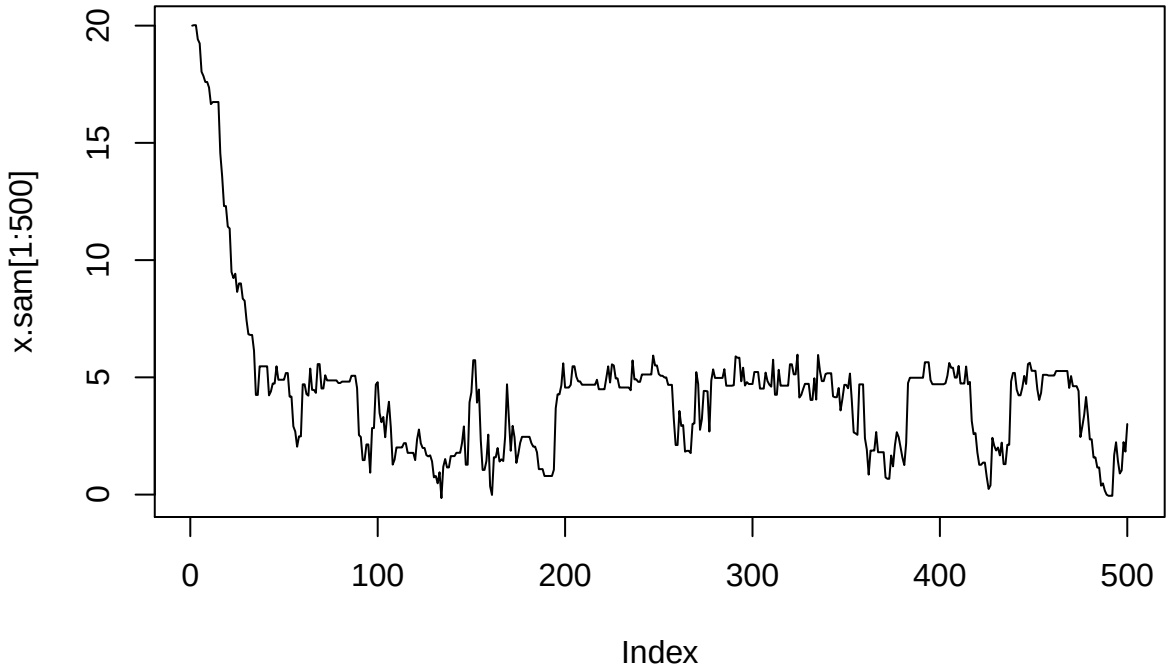


Generate a sample from the mixture normal using the Metropolis algorithm

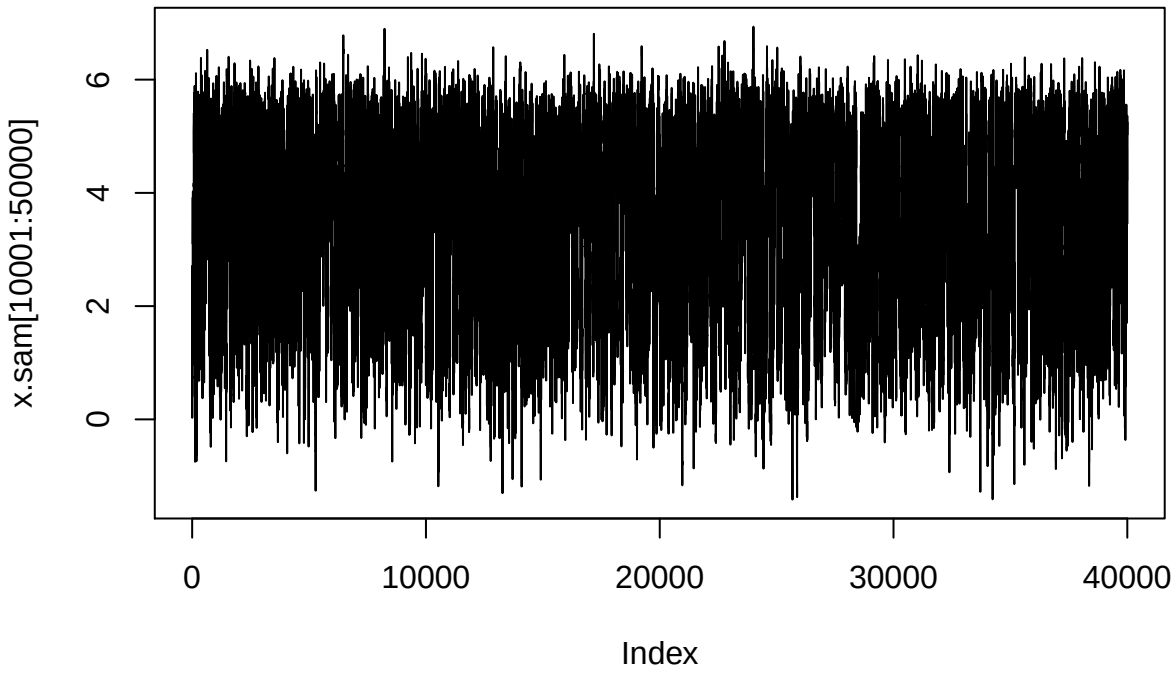
```
f.metropolis.x <- function(niter, x0) {
  tmp.x <- rep(NA, niter)
  tmp.x[1] <- x0   # choose a starting point

  for(ii in 2:niter) {
    tmp.x[ii] <- tmp.x[ii-1] + rnorm(1) # propose new value
    tmp.dist.x <- exp(-0.5*(tmp.x[ii-1]-2)^2) + 2*exp(-2*(tmp.x[ii-1]-5)^2)
    tmp.dist.x.proposed <- exp(-0.5*(tmp.x[ii]-2)^2) + 2*exp(-2*(tmp.x[ii]-5)^2)
    tmp.alpha <- min(tmp.dist.x.proposed/tmp.dist.x, 1)

    tmp.U <- runif(1)
    tmp.x[ii] <- ifelse( tmp.U < tmp.alpha, tmp.x[ii], tmp.x[ii-1] )
  }
  return(tmp.x)
}

set.seed(10)
x.sam <- f.metropolis.x(niter=50000, x0=20)
plot(x.sam, type="l")
```




```
## Initial samples
plot(x.sam[1:500], type="l")
```




```
## Good samples
## Burn-in = 10000 samples
plot(x.sam[10001:50000], type="l")
```




```
##
post.mean <- mean(x.sam[10001:50000])
post.var <- var(x.sam[10001:50000])
c(post.mean, post.var)
```

```
## [1] 3.538878 2.865022
```

Poisson-gamma example

```
y <- c(5,1,5,14,3,19,1,1,4,22)
t <- c(94,16,63,126,5,31,1,1,2,10)
rbind(y,t)
```

```
##   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## y    5    1    5   14    3   19    1    1    4    22
## t   94   16   63  126    5   31    1    1    2    10
```

```
gibbs <- function(n.sims, beta.start, alpha, gamma, delta, y,t, burnin, thin) {
  beta.draws <- c()
  theta.draws <- matrix(NA, nrow=n.sims, ncol=length(y))
  beta.cur <- beta.start

  theta.update <- function(alpha,beta,y,t) {
    rgamma(length(y), y+alpha, t+beta)
  }

  beta.update <- function(alpha, gamma, delta, theta, y) {
    rgamma(1, length(y)*alpha+gamma, delta+sum(theta))
  }

  for (i in 1:n.sims) {
    theta.cur <- theta.update(alpha=alpha, beta=beta.cur, y=y, t=t)
    beta.cur <- beta.update(alpha=alpha, gamma=gamma, delta=delta, theta=theta.cur, y=y)
    if (i > burnin & (i - burnin)%%thin==0) {
      theta.draws[(i-burnin)/thin,] <- theta.cur
      beta.draws[(i-burnin)/thin] <- beta.cur
    }
  }
  return(list(theta.draws=theta.draws, beta.draws=beta.draws))
}

posterior <- gibbs(n.sims=10000, beta.start=1, alpha=1.8, gamma=0.01, delta=1, y=y, t=t,
                   burnin=0, thin=1)
## Posterior mean
round(colMeans(posterior$theta.draws), 4)
```

```
##  [1] 0.0708 0.1530 0.1046 0.1227 0.6552 0.6233 0.8705 0.8530 1.3549 1.9237
```

```
round(mean(posterior$beta.draws), 4)
```

```
## [1] 2.3857
```

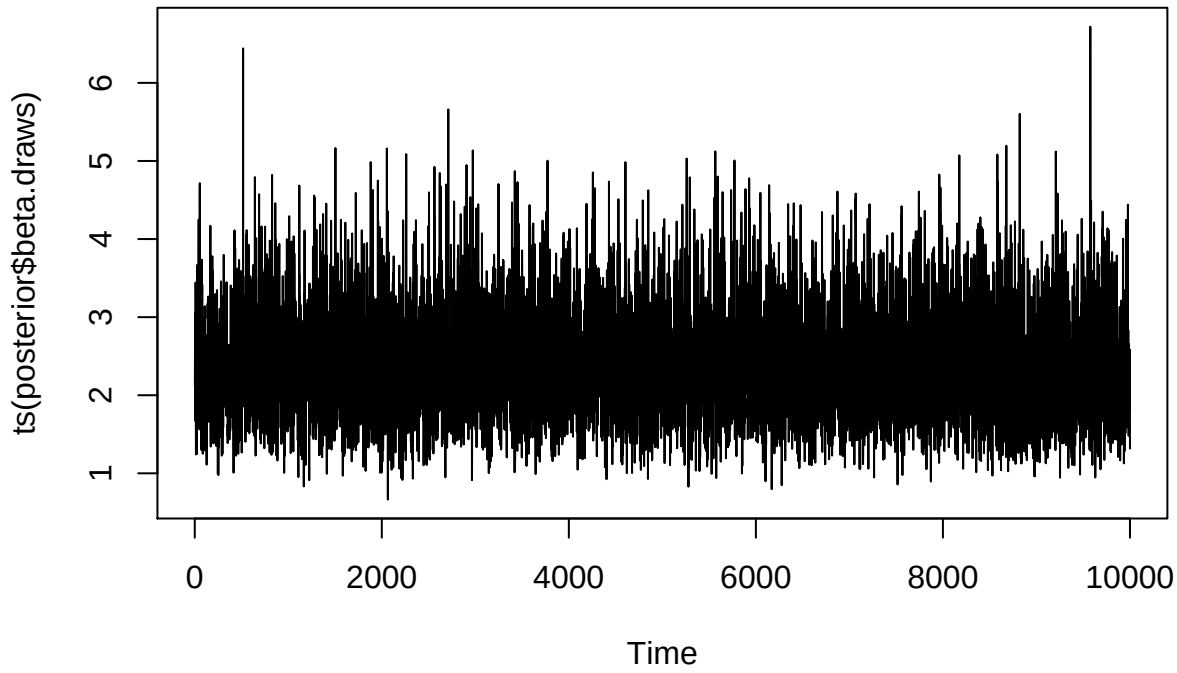
```
## Posterior sd
round(apply(posterior$theta.draws, 2, sd), 4)
```

```
##  [1] 0.0272 0.0918 0.0401 0.0309 0.3086 0.1360 0.5587 0.5396 0.6054 0.4065
```

```
round(sd(posterior$beta.draws), 4)
```

```
## [1] 0.6858
```

```
## Trace plot of beta
plot(ts(posterior$beta.draws))
```




```
## Posterior distribution of beta
hist(posterior$beta.draws,50)
```