Normal Model

given σ^2

sampling model

$$Y_1, \ldots, Y_n | \theta, \sigma^2 \overset{\text{i.}i.d.}{\sim} \text{normal}(\theta, \sigma^2)$$

conditional prior

$$\theta | \sigma^2 \sim \text{normal}\Big(\mu_0, \tau_0^2\Big)$$

posterior

$$\theta | \sigma^2, y_1, \ldots, y_n \sim \text{normal}\Big(\mu_n, \tau_n^2\Big)$$

$\mu_n = \frac{\mu_0/\tau_0^2 + n\bar{y}/\sigma^2}{1/\tau_0^2 + n/\sigma^2}$ and $\tau_n^2 = \frac{1}{1/\tau_0^2 + n/\sigma^2}$

set $\tau_0^2 = \frac{\sigma^2}{\kappa_0}$, $\mu_n = \frac{\kappa_0}{\kappa_0+n}\mu_0 + \frac{n}{\kappa_0+n}\bar{y}$ and $\tau_n^2 = \frac{\sigma^2}{\kappa_0+n} = \frac{\sigma^2}{\kappa_n}$, where $\kappa_n = \kappa_0 + n$

joint prior $p(\theta, \sigma^2) = p(\theta|\sigma^2)p(\sigma^2)$

$$1/\sigma^2 \sim \text{gamma}(\frac{\nu_0^2}{2}, \frac{\nu_0^2}{2}\sigma_0^2)$$
$$\theta | \sigma^2 \sim \text{normal}(\mu_0, \frac{\sigma^2}{\kappa_0})$$
$$Y_1, \ldots, Y_n | \theta, \sigma^2 \overset{\text{i.}i.d.}{\sim} \text{normal}(\theta, \sigma^2)$$

posterior

$$1/\sigma^2 | y_1, \ldots, y_n \sim \text{gamma}(\frac{\nu_n^2}{2}, \frac{\nu_n^2}{2}\sigma_n^2)$$
$$\theta | \sigma^2, y_1, \ldots, y_n \sim \text{normal}(\mu_n, \frac{\sigma^2}{\kappa_n})$$

$\nu_n = \nu_0 + n$ and $\sigma_n^2 = \frac{1}{\nu_n}[\nu_0^2\sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_n}(\bar{y} - \mu_0)2]$, $s^2 = \frac{1}{n-1}\sum_{l=1}^{n}(y_l - \bar{y})^2$

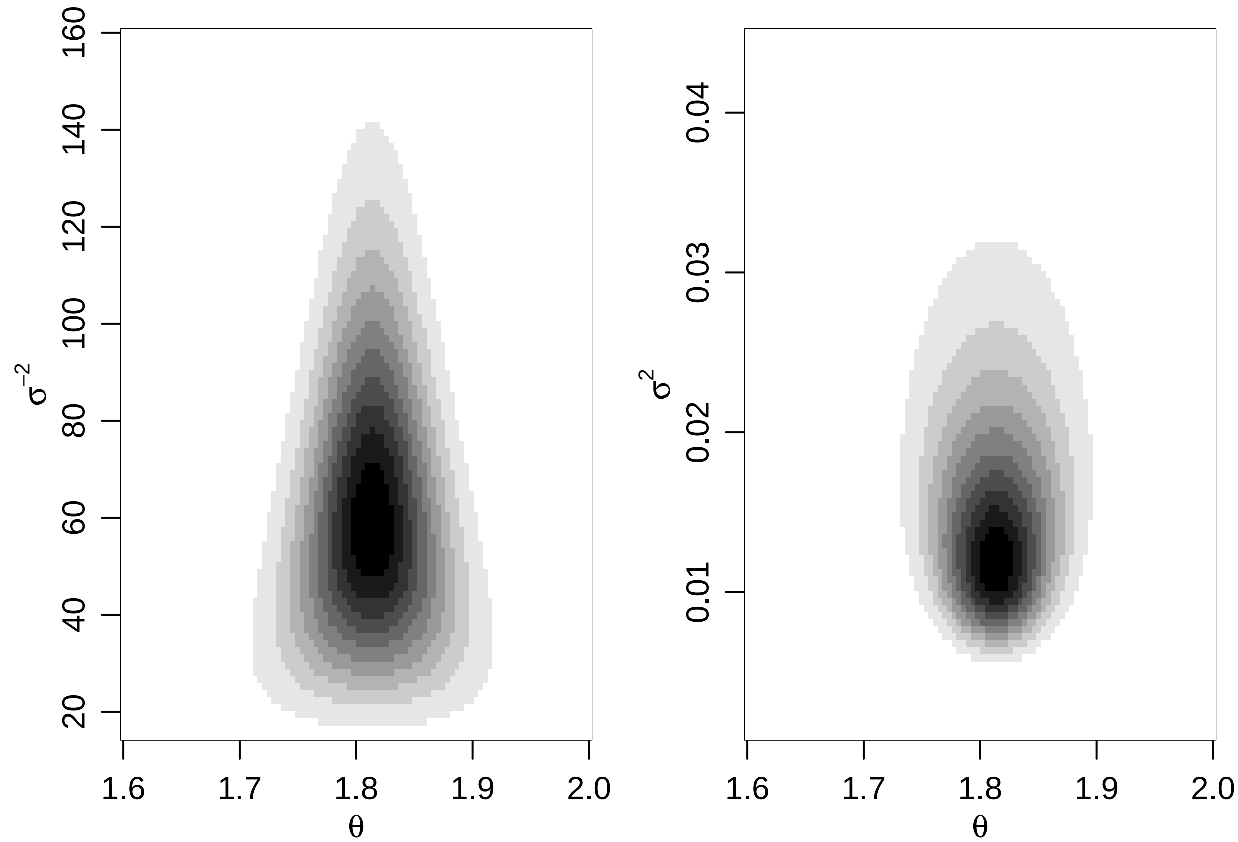
```
> # Fig 5.4 Joint posterior distributions of (theta, sigma^-2) and  (theta, sigma^2)
> # http://www2.stat.duke.edu/~pdh10/FCBS/Replication/chapter5.R
> # prior
> # 1/sigma^2 ~ dgamma(nu_0/2, nu_0 sigma^2_0/2)
> s2_0 <- 0.01 ; nu_0 <- 1
> # theta|sigma^2 ~ dnorm(mu_0, sigma/sqrt(k_0))
> mu_0 <- 1.9 ; k_0 <- 1
>
> ## data
> y <- c(1.64, 1.70, 1.72, 1.74, 1.82, 1.82, 1.82, 1.90, 2.08)
> n <- length(y) ; ybar<-mean(y) ; s2 <- var(y)
>
> ## posterior inference
> k_n <- k_0 + n ; nu_n <- nu_0 + n
> mu_n <- ( k_0 * mu_0 + n * ybar ) / k_n
> s2_n <- ( nu_0 * s2_0 + (n-1) * s2 + k_0 * n * (ybar - mu_0)^2 / k_n ) / nu_n
> mu_n
```

```
[1] 1.814
```

```
> s2_n
```

```
[1] 0.015324
```

```
> # density of inverse gamma beta^alpha / Gamma(alpha) * x^-(alpha+1) exp(-beta/x)
> dinvgamma<-function(x,a,b) {
+   ld <- a * log(b) - lgamma(a) - (a+1) * log(x) - b / x
+   exp(ld)
+ }
>
> # grid size
> gs <- 100
> theta_g <- seq(1.6, 2.0, length = gs)
> is2_g <- seq(15, 160 , length = gs)
> s2_g <- seq(0.001, 0.045, length = gs)
>
> # log density theta inverse sigma square / sigma square
> ld.th.is2 <- ld.th.s2 <- matrix(0, gs, gs)
> for(i in 1:gs) {
+   for(j in 1:gs) {
+     ld.th.is2[i,j]<- dnorm( theta_g[i], mu_n, 1/sqrt(is2_g[j]*k_n), log = TRUE ) +
+                     dgamma( is2_g[j], shape = nu_n/2, rate = nu_n*s2_n/2, log = TRUE )
+     ld.th.s2[i,j]<- dnorm( theta_g[i], mu_n, sqrt(s2_g[j]/k_n), log = TRUE ) +
+                      log( dinvgamma(s2_g[j], nu_n/2, nu_n * s2_n/2 ) )
+   }
+ }
>
> par(mfrow = c(1,2), mar = c(3,3,1,1), mgp = c(1.75,0.75,0))
> grays <- gray((10:0)/10)
> image( theta_g, is2_g, exp(ld.th.is2), col = grays, xlab = expression(theta),
+        ylab = expression(sigma^{-2}) )
> image( theta_g, s2_g, exp(ld.th.s2), col = grays, xlab = expression(theta),
+        ylab = expression(sigma^2) )
```



semiconjugate prior $p(\theta, \sigma^2) = p(\theta)p(\sigma^2)$

$$\theta|\sigma^2 \sim \text{normal}(\mu_0, \tau_0^2)$$
$$1/\sigma^2 \sim \text{gamma}(\frac{\nu_0^2}{2}, \frac{\nu_0^2}{2}\sigma_0^2)$$
$$Y_1, \dots, Y_n|\theta, \sigma^2 \overset{\text{i.}i.d.}{\sim} \text{normal}(\theta, \sigma^2)$$

posterior

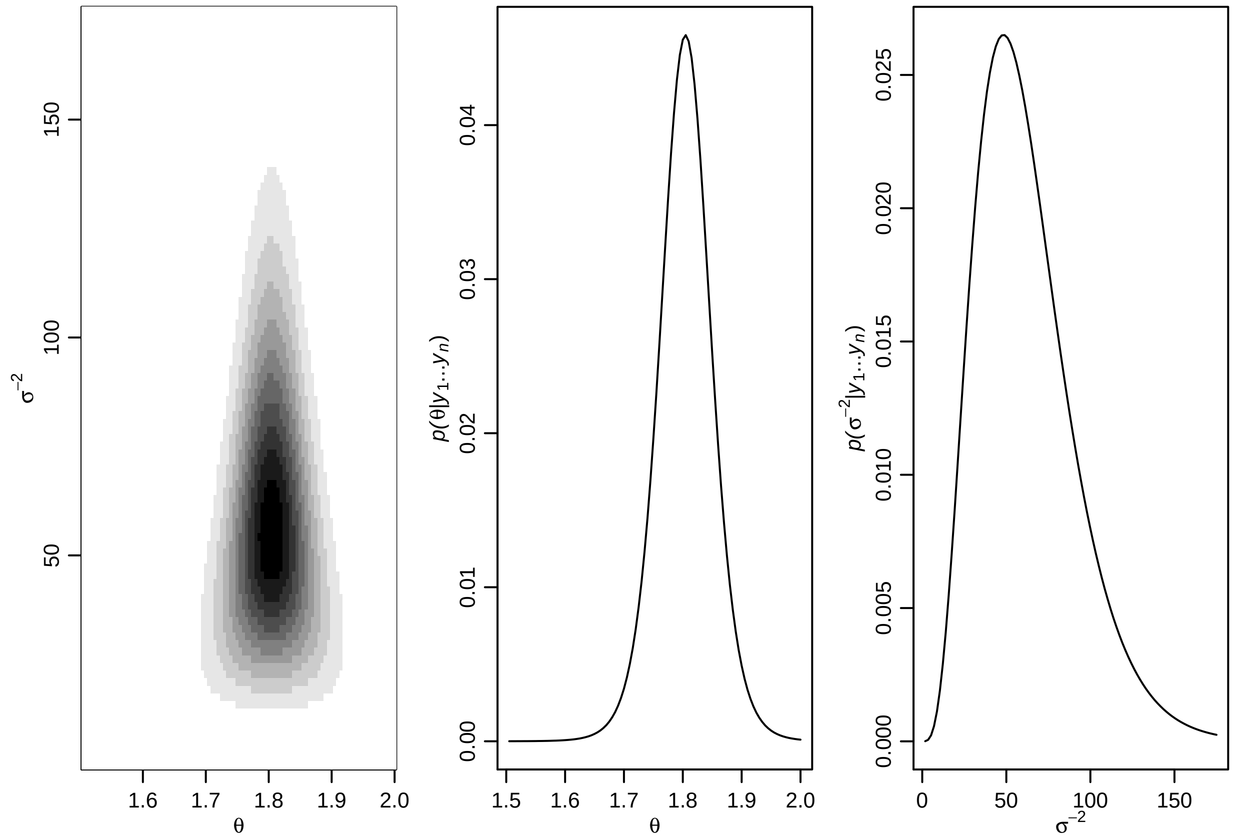
$$\theta|y_1, \dots, y_n \sim \text{normal}(\mu_n, \tau_n^2)$$
$$1/\sigma^2|y_1, \dots, y_n \sim \text{some distribution not standard}$$
$$1/\sigma^2|\theta, y_1, \dots, y_n \sim \text{gamma}(\frac{\nu_n^2}{2}, \frac{\nu_n^2}{2}\sigma_n^2(\theta))$$

$\mu_n = \frac{\mu_0 \tau_0^2 + n\bar{y}/\sigma^2}{1/\tau_0^2 + n/\sigma^2}$ and $\tau_n^2 = \frac{1}{1/\tau_0^2 + n/\sigma^2}$

$\nu_n = \nu_0 + n$ and $\sigma_n^2(\theta) = \frac{1}{\nu_n}[\nu_0^2\sigma_0^2 + ns_n^2(\theta)]$, where $s_n^2(\theta) = \frac{1}{n}\sum_{l=1}^{n}(y_l - \theta)^2 = \frac{1}{n}[(n-1)s^2 + n(\bar{y} - \theta)^2]$

```
> # Fig 6.1 Joint and marginal posterior distributions based on a discrete approximation (theta, sigma^
> # http://www2.stat.duke.edu/~pdh10/FCBS/Replication/chapter6.R
> # prior
> # 1/sigma^2 ~ dgamma(nu_0/2, nu_0 sigma^2_0/2)
> s2_0 <- 0.01 ; nu_0 <- 1
> # theta ~ dnorm(mu_0, tau_0)
> mu_0 <- 1.9  ; t2_0 <- 0.95^2
```

```
>
> # data
> y <- c(1.64, 1.70, 1.72, 1.74, 1.82, 1.82, 1.82, 1.90, 2.08)
> n <- length(y) ; ybar <- mean(y) ; s2 <- var(y)
>
>
> # grid size
> gs <- 100
> theta_g <- seq(1.505, 2.00, length = gs)
> is2_g <- seq(1.75, 175, length = gs)
>
> posterior_g <- matrix(0, nrow = gs, ncol = gs)
>
> for(i in 1:gs) {
+   for(j in 1:gs) {
+     posterior_g[i,j]<- dnorm( theta_g[i], mu_0, sqrt(t2_0) ) *
+                        dgamma( is2_g[j], nu_0/2, s2_0*nu_0/2 ) *
+                        prod( dnorm( y, theta_g[i], 1/sqrt(is2_g[j]) ) )
+   }
+ }
>
> posterior_g <- posterior_g/sum(posterior_g)
>
> par(mfrow = c(1,3), mar = c(3,3,1,1), mgp = c(1.70,0.70,0))
> image( theta_g, is2_g, posterior_g, col = grays, xlab = expression(theta),
+        ylab=expression(sigma^{-2}) )
>
> theta_p<- apply(posterior_g, 1, sum)
> plot( theta_g, theta_p, type = "l", xlab = expression(theta),
+      ylab = expression( paste( italic("p("), theta, "|",
+                              italic(y[1]), "...", italic(y[n]), ")", sep="" ) ) )
>
> is2_p <- apply(posterior_g, 2, sum)
> plot( is2_g, is2_p, type = "l", xlab = expression(sigma^{-2}),
+      ylab = expression( paste( italic("p("), sigma^{-2}, "|",
+                              italic(y[1]), "...", italic(y[n]), ")", sep="" ) ) )
```



Gibbs sampler

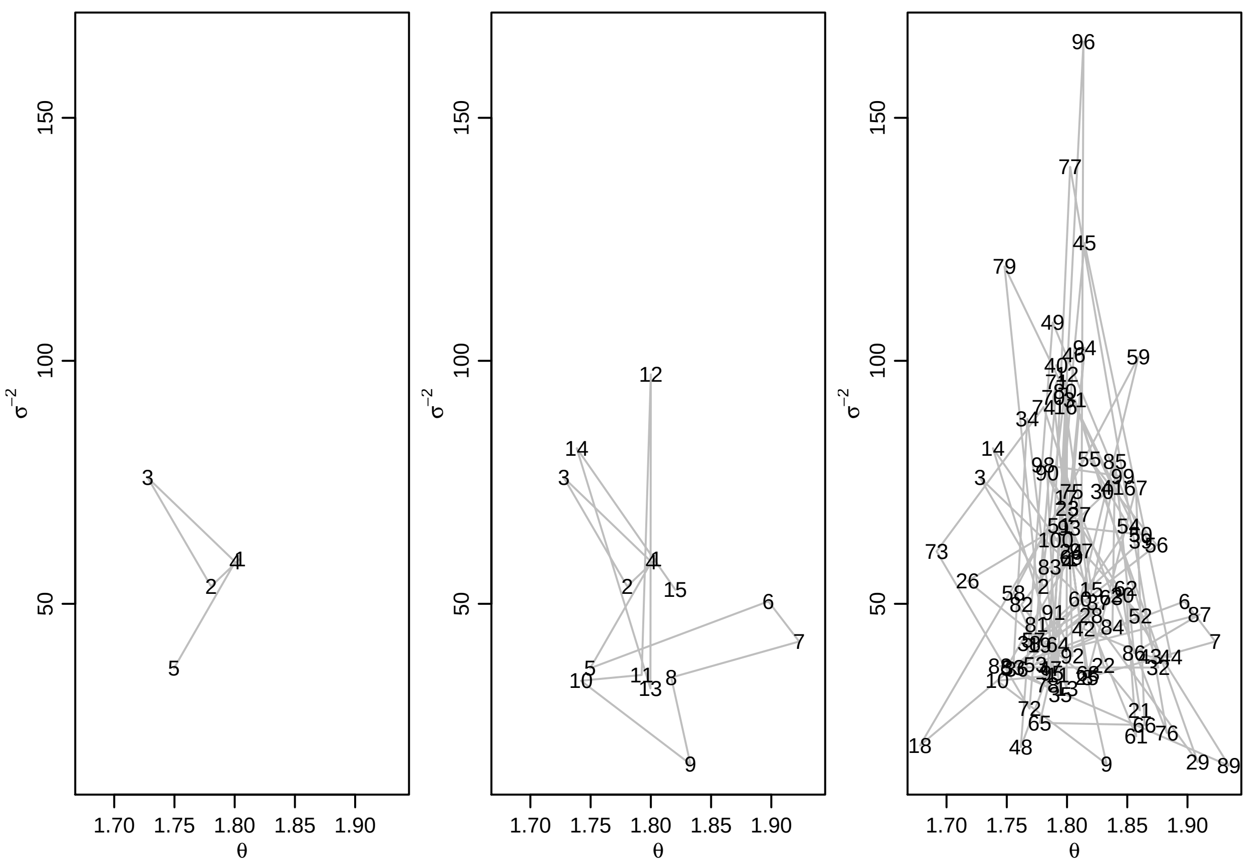
full conditional distribution $p(\theta|\sigma^2, y_1, \ldots, y_n)$ and $p(\sigma^2|\theta, y_1, \ldots, y_n)$

$$\theta|\sigma^2, y_1, \ldots, y_n \overset{\theta \perp \sigma}{=} \theta|y_1, \ldots, y_n \sim \text{normal}(\mu_n, \tau_n^2)$$

$$1/\sigma^2|\theta, y_1, \ldots, y_n \sim \text{gamma}(\frac{\nu_n^2}{2}, \frac{\nu_n^2}{2}\sigma_n^2(\theta))$$

```
> # http://www2.stat.duke.edu/~pdh10/FCBS/Replication/chapter6.R
> ## initial values
> set.seed(123)
> nsamp <- 1000
> mphi <- matrix(nrow = nsamp, ncol = 2)
> mphi[1,] <- phi <- c(ybar, 1/s2)
>
> ## Gibbs sampling algorithm
> for(s in 2:nsamp) {
+
+   # generate a new theta value from its full conditional
+   # mu_n = [mu_0/\tau_0^2 + n*ybar/sigma^2]/[1/tau_0^2 + n/sigma^2]
+   mu_n <-  ( mu_0/t2_0 + n*ybar*phi[2] ) / ( 1/t2_0 + n*phi[2] )
+   # tau_n^2 = 1/[1/tau_0^2 + n/sigma^2]
+   t2_n <- 1 / ( 1/t2_0 + n*phi[2] )
+   phi[1] <- rnorm( 1, mu_n, sqrt(t2_n) )
+
+   # generate a new sigma^2 value from its full conditional
+   nu_n <- nu_0 + n
```

```
+   # sigma_n^2(\theta) = 1 / nu_n * [nu_0^2*sigma_0^2 + (n-1)*s^2 + n(ybar-theta)^2]
+   s2_n <- ( nu_0*s2_0 + (n-1)*s2 + n*(ybar-phi[1])^2 ) /nu_n
+   phi[2] <- rgamma( 1, nu_n/2, nu_n*s2_n/2 )
+
+   mphi[s,]<-phi
+ }
>
> # Fig 6.2 The first 5, 15 and 100 iterations of a Gibbs sampler
> par(mfrow = c(1,3), mar = c(2.75,2.75,0.5,0.5), mgp = c(1.70,0.70,0))
> m1 <- 5
> plot( mphi[1:m1,], type = "l", xlim = range(mphi[1:100,1]), ylim = range(mphi[1:100,2]),
+      lty = 1, col = "gray", xlab = expression(theta), ylab = expression(sigma^{-2}) )
> text( mphi[1:m1,1], mphi[1:m1,2], c(1:m1) )
>
> m1 <- 15
> plot( mphi[1:m1,], type = "l", xlim = range(mphi[1:100,1]), ylim = range(mphi[1:100,2]),
+      lty = 1, col = "gray", xlab = expression(theta), ylab = expression(sigma^{-2}) )
> text( mphi[1:m1,1], mphi[1:m1,2], c(1:m1) )
>
> m1 <- 100
> plot( mphi[1:m1,], type = "l", xlim = range(mphi[1:100,1]), ylim = range(mphi[1:100,2]),
+      lty = 1, col = "gray", xlab = expression(theta), ylab = expression(sigma^{-2}))
> text( mphi[1:m1,1], mphi[1:m1,2], c(1:m1) )
```

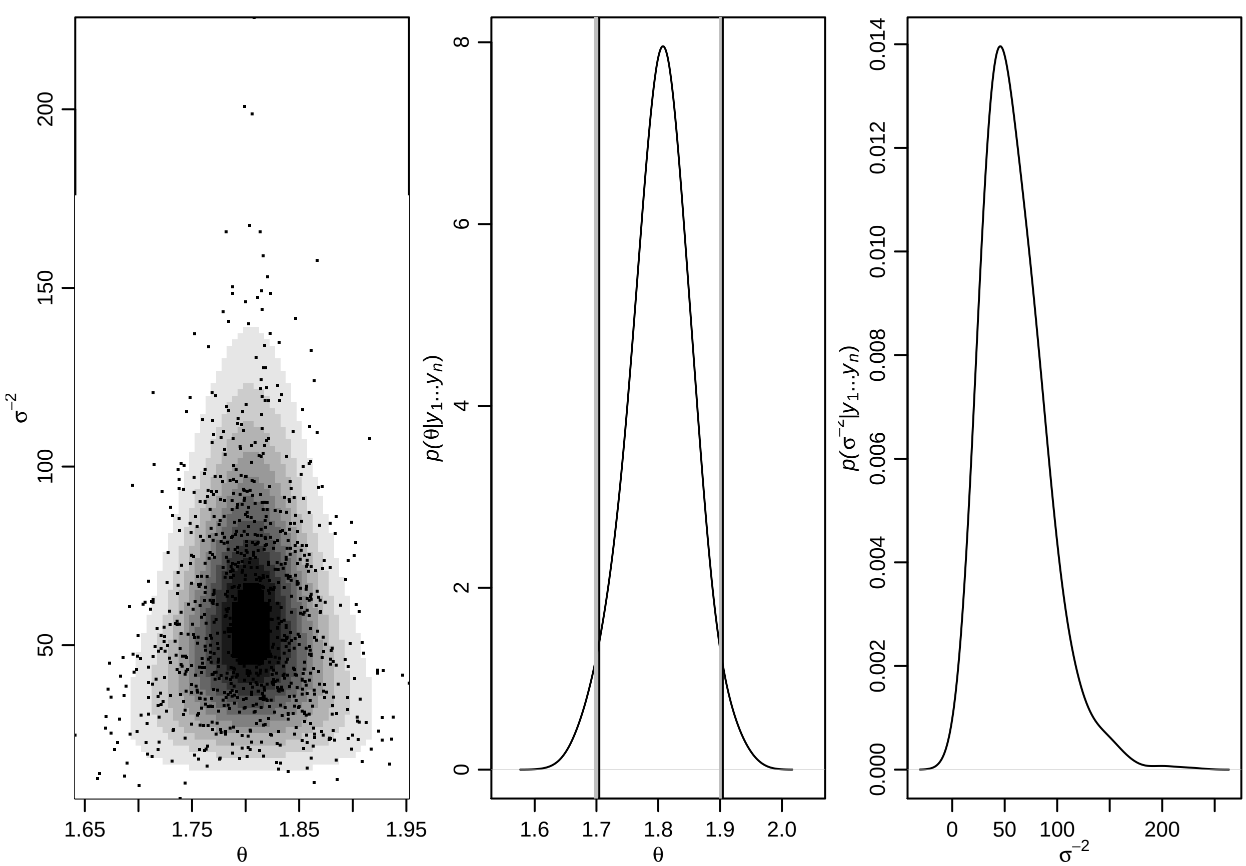



```
> # Fig 6.2 distribution of Gibbs sample
> par(mfrow = c(1,3), mar = c(2.75,2.75,0.5,0.5), mgp = c(1.70,0.70,0))
```

```
>
> image( theta_g, is2_g, posterior_g, col = grays,
+        xlab = expression(theta), ylab = expression(sigma^{-2}),
+        xlim = range(mphi[,1]), ylim = range(mphi[,2]) )
> points( mphi[,1], mphi[,2], pch = ".", cex = 1.25 )
>
> plot( density(mphi[,1], adj = 2),  xlab = expression(theta), main = "", xlim = c(1.55,2.05),
+      ylab = expression( paste( italic("p("), theta, "|", italic(y[1]),
+                               "...", italic(y[n]), ")", sep = "" ) ) )
> abline( v = quantile(mphi[,1], prob = c(0.025,0.975)), lwd = 2, col = "gray" )
>
> ## t-test based confidence interval
> ybar + qt( c(0.025,0.975), n-1 ) *sqrt(s2/n)
```

```
[1] 1.704583 1.904306
```

```
> abline( v =  ybar + qt(c(0.025,0.975), n-1) * sqrt(s2/n), col = "black", lwd = 1 )
>
> plot( density( mphi[,2], adj = 2 ), xlab = expression(sigma^{-2}), main = "",
+      ylab = expression( paste( italic("p("), sigma^{-2}, "|", italic(y[1]),
+                                "...", italic(y[n]), ")",sep = "" ) ) )
```



MCMC variance

$$
\begin{aligned}
\text{Var}_{\text{MCMC}}(\bar{\phi}) &= \text{Var}_{\text{MC}}(\bar{\phi}) + \frac{1}{S^2}\text{E}\{(\phi^{(s)}-\phi_0)(\phi^{(t)}-\phi_0)\} = \frac{\text{Var}(\phi)}{S_{\text{eff}}} \\
\text{Var}_{\text{MC}}(\bar{\phi}) &= \frac{\text{Var}(\phi)}{S} \\
\text{Var}(\phi) &= \int \phi^2 p(\phi)d\phi - \phi_0^2 \\
\text{E}\{\phi\} &= \int \phi p(\phi)d\phi = \phi_0 \\
\bar{\phi} &= \frac{1}{S}\sum \phi^{(s)}
\end{aligned}
$$

```
> # effective sample size S_eff
> library(coda)
> effectiveSize( mphi[,1] )
```

```
    var1
912.9684
```

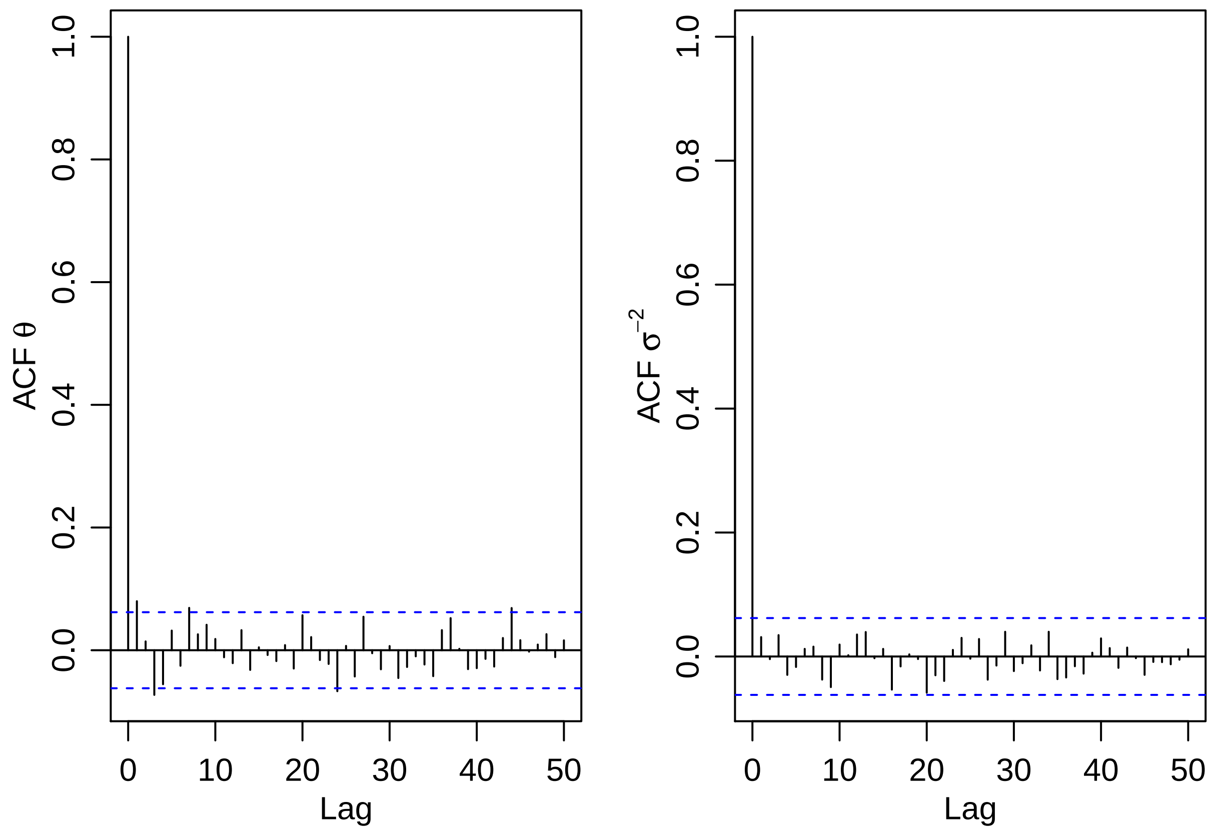
```
> effectiveSize( 1/mphi[,2] )
```

```
    var1
881.4031
```

the lag-t autocorrelation

$$
\text{acf}_t(\phi) = \frac{\frac{1}{S-t}\sum_{s=1}^{S-t}(\phi^{(s)}-\bar{\phi})(\phi^{(t)}-\bar{\phi})}{\frac{1}{S-1}\sum_{s=1}^{S}(\phi^{(s)}-\bar{\phi})^2}
$$

```
> par(mfrow = c(1,2), mar = c(3,3,1,1), mgp = c(1.75,0.75,0))
> acf( mphi[,1], ylab = expression( paste("ACF ", theta) ), lag.max = 50 )
> acf( mphi[,2], ylab = expression( paste("ACF ", sigma^{-2}) ), lag.max = 50 )
```

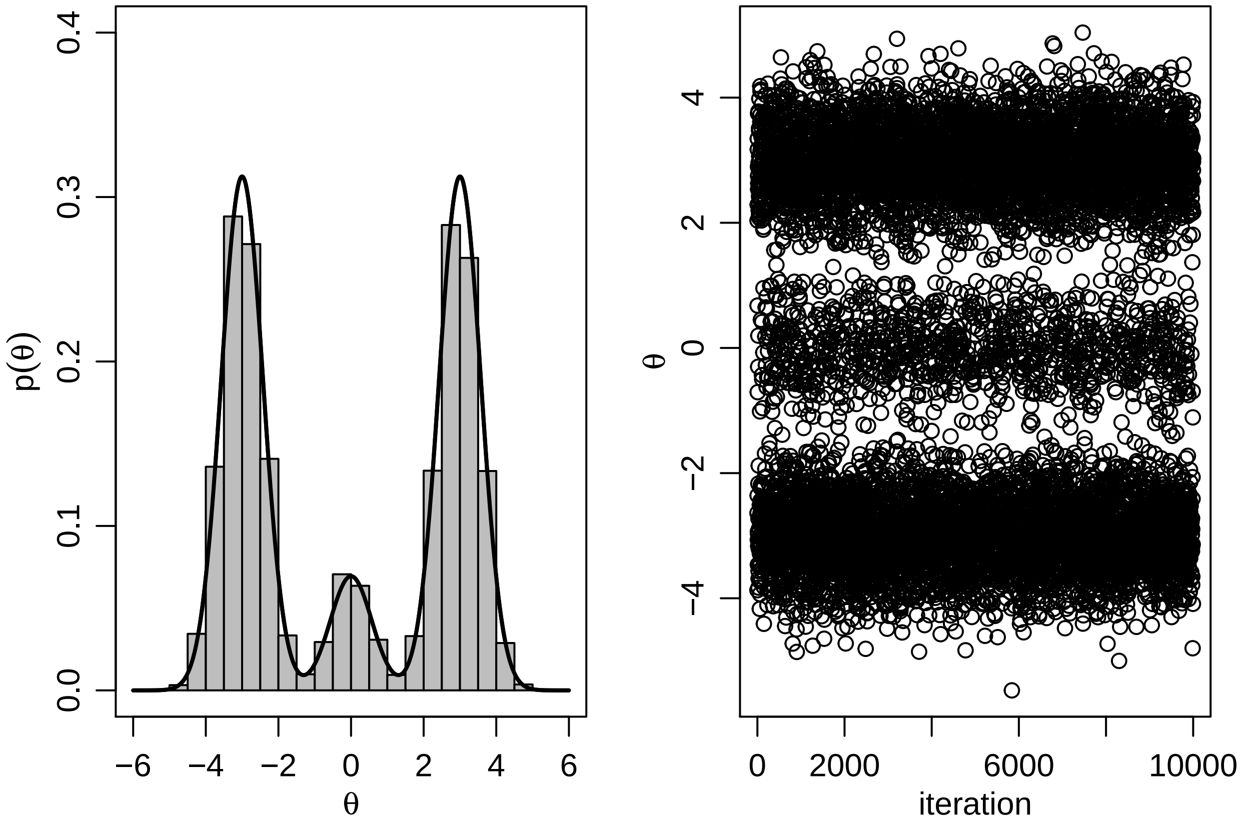


mixture of three normal distributions

$$p(\theta, \delta) = (\mathrm{dnorm}(\theta, \mu_1, \sigma_1))^{I(d=1)} (\mathrm{dnorm}(\theta, \mu_2, \sigma_2))^{I(d=2)} (\mathrm{dnorm}(\theta, \mu_3, \sigma_3))^{I(d=3)}$$

```
> # http://www2.stat.duke.edu/~pdh10/FCBS/Replication/chapter6.R
> mu <- c(-3,0,3)
> s2 <- c(0.33,0.33,0.33)
> w <- c(0.45,0.1,0.45)
>
> # MC sampling
> set.seed(1)
> nsamp <- 10000
> d <- sample(1:3, nsamp, prob = w, replace = TRUE)
> theta <- rnorm( nsamp, mu[d], sqrt(s2[d]) )
> thetaMC <- cbind(theta, d)
>
>
> par(mfrow = c(1,2), mar = c(3,3,1,1), mgp = c(1.75,.75,0))
> thetas <- seq(-6,6,length=1000)
> plot( thetas, w[1]*dnorm( thetas, mu[1], sqrt(s2[1]) ) +
+         w[2]*dnorm( thetas, mu[2], sqrt(s2[2]) ) +
+         w[3]*dnorm( thetas, mu[3], sqrt(s2[3]) ), type="l",
+       xlab = expression(theta), ylab = expression(p(theta)), lwd = 2, ylim = c(0,.40) )
> hist( thetaMC[,1], add = TRUE, prob = TRUE, nclass = 20, col = "gray" )
> lines( thetas, w[1]*dnorm(thetas,mu[1],sqrt(s2[1])) +
+           w[2]*dnorm(thetas,mu[2],sqrt(s2[2])) +
```

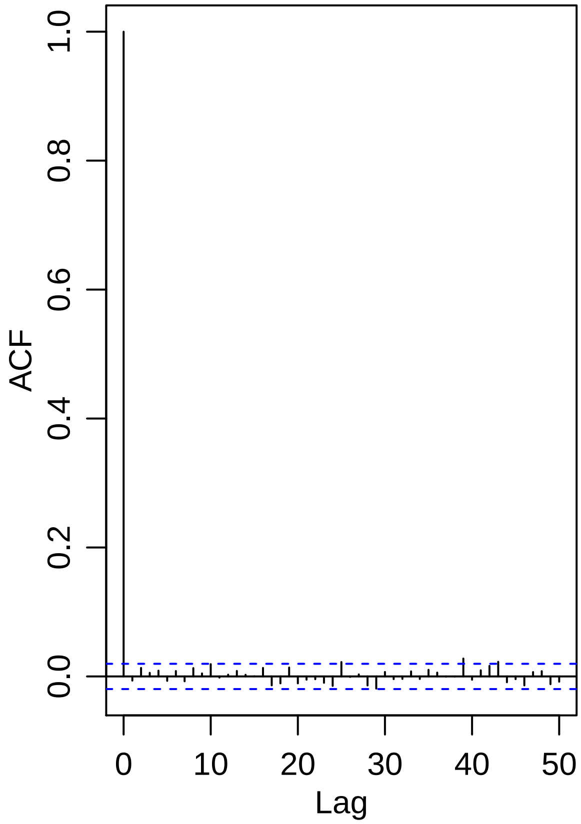
```
+           w[3]*dnorm(thetas,mu[3],sqrt(s2[3])), lwd=2 )
> plot( thetaMC[,1], xlab = "iteration", ylab = expression(theta) )
```



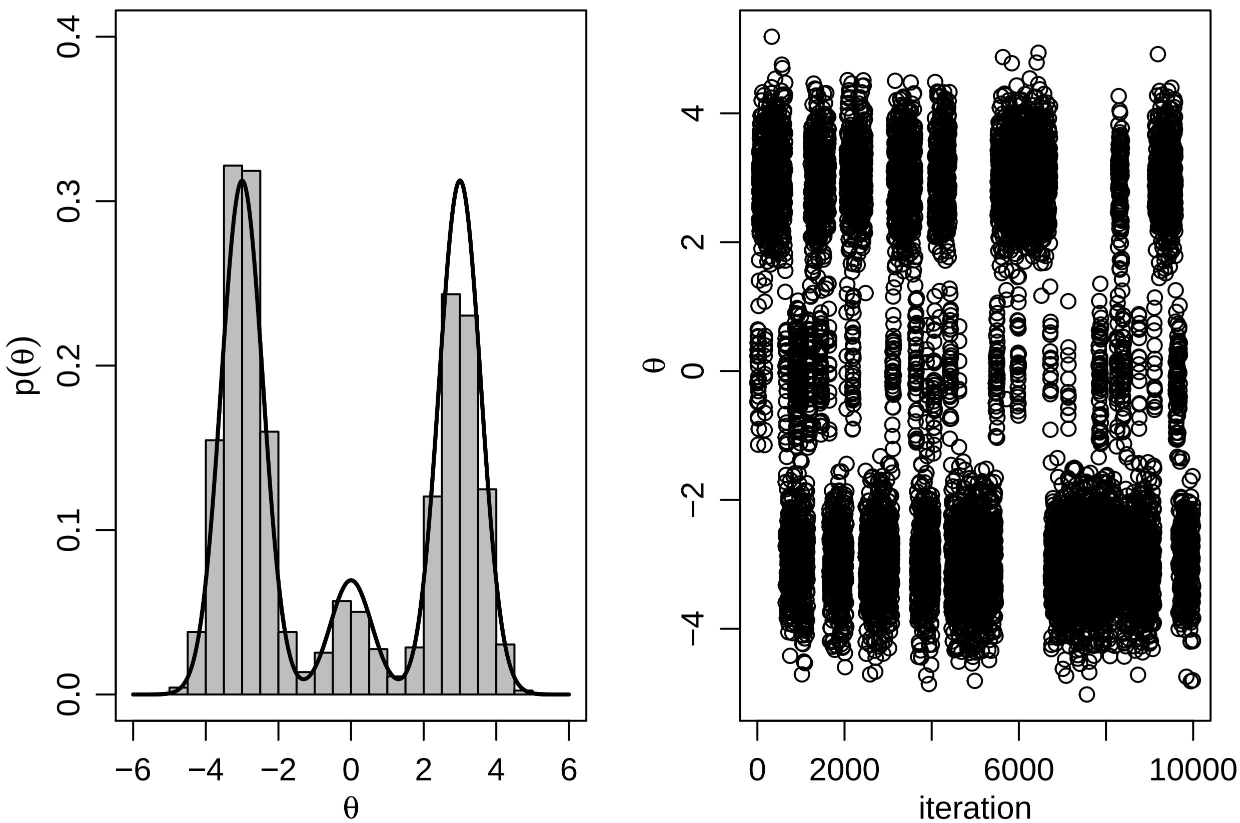

```
> # Autocorrelation and effective sample size
> acf(thetaMC[,1], lag.max = 50)
> effectiveSize(thetaMC[,1])
```

```
 var1
10000
```

```
> #### MCMC sampling
> theta <- 0
> thetaMCMC <- matrix(0, nrow = nsamp, ncol = 2)
> set.seed(1)
> for(s in 1:nsamp) {
+   d <- sample( 1:3, 1, prob = w*dnorm( theta, mu, sqrt(s2) ) )
+   theta <- rnorm( 1, mu[d], sqrt(s2[d]) )
+   thetaMCMC[s,] <- c(theta,d)
+ }
>
> par(mfrow = c(1,2), mar = c(3,3,1,1), mgp = c(1.75,.75,0))
```

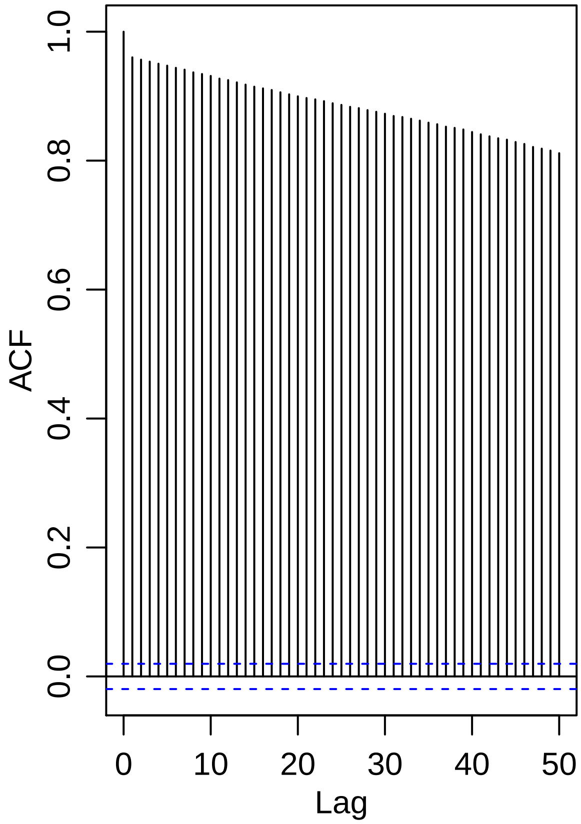



```
> thetas <- seq(-6,6,length=1000)
> plot( thetas, w[1]*dnorm( thetas, mu[1], sqrt(s2[1]) ) +
+         w[2]*dnorm( thetas, mu[2], sqrt(s2[2]) ) +
+         w[3]*dnorm( thetas, mu[3], sqrt(s2[3]) ), type="l",
+       xlab = expression(theta), ylab = expression(p(theta)), lwd = 2, ylim = c(0,.40) )
> hist( thetaMCMC[,1], add = TRUE, prob = TRUE, nclass = 20, col = "gray" )
> lines( thetas, w[1]*dnorm(thetas,mu[1],sqrt(s2[1])) +
+           w[2]*dnorm(thetas,mu[2],sqrt(s2[2])) +
+           w[3]*dnorm(thetas,mu[3],sqrt(s2[3])), lwd=2 )
> plot(thetaMCMC[,1], xlab = "iteration", ylab = expression(theta))
```




```
> #### Autocorrelation and effective sample size
> acf(thetaMCMC[,1], lag.max = 50)
> effectiveSize(thetaMCMC[,1])
```

```
    var1
18.42419
```


ACF
Lag
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50