

Exercise__06__BC

Betsy Cowdery

October 14, 2014

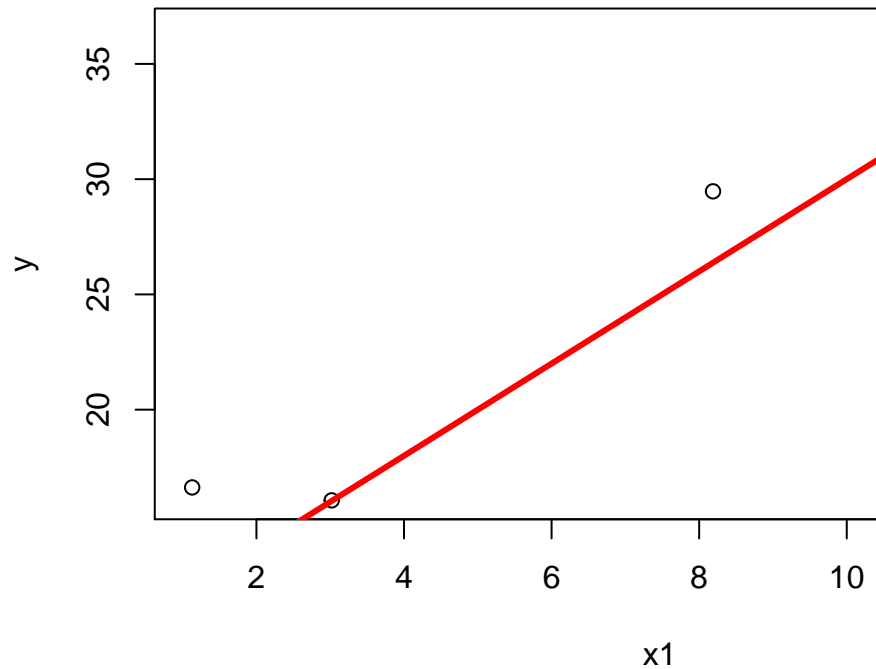
Bayesian Regression using Gibbs Sampling

Setup

```
n <- 5          ## define the sample size
b0 <- 10         ## define the intercept
b1 <- 2          ## define the slope
beta <- matrix(c(b0,b1),2,1)      ## put "true" regression parameters in a matrix
sigma2 <- 4^2    ## define the variance (s.d. = 4)
```

```
x1 <- runif(n,0,20)
x <- cbind(rep(1,n),x1)
y <- matrix(rnorm(n,x%*%beta,sqrt(sigma2)),n,1)
```

```
plot(x1,y)
abline(b0,b1,col=2,lwd=3)
```



Part 1: simulate data from a known model

```
bprior <- as.vector(c(0,0))
vinvert <- solve(diag(1000,2))
s1 <- 0.1
s2 <- 0.1
```

specify priors

```
XX <- t(x) %*% x
XY <- t(x) %*% y
VbB <- vinvert %*% bprior
```

precompute frequently used quantities

```
## Loading required package: lattice
```

```
ngibbs <- 1000          ## number of updates
bgibbs <- matrix(0.0,nrow=ngibbs,ncol=2)  ## storage for beta
sgibbs <- numeric(ngibbs)      ## storage for sigma2
```

storage for MCMC

```
sg <- 50
sinv <- 1/sg
```

initial conditions

MCMC loop

```
for(g in 1:ngibbs){

  ## sample regression parameters
  bigV    <- solve(sinv*XX + vinvert)  ## Covariance matrix
  littlev <- sinv*XY + VbB
  b = t(rmvnorm(1,bigV %*% littlev,bigV))  ## Vv is the mean vector

  ## sample variance
  u1 <- s1 + n/2
  u2 <- s2 + 0.5*crossprod(y-x%*%b)
  sinv <- rgamma(1,u1,u2)
```

```

sg <- 1/sinv

## storage
bgibbs[g,] <- b ## store the current value of beta vector
sgibbs[g] <- sg ## store the current value of the variance

if(g %%100 == 0) print(g) ##show how many steps have been performed
}

```

Gibbs loop

```

## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000

```

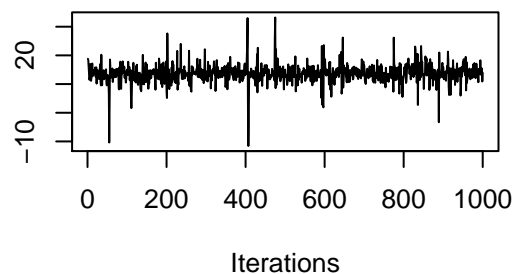
Evaluation

```

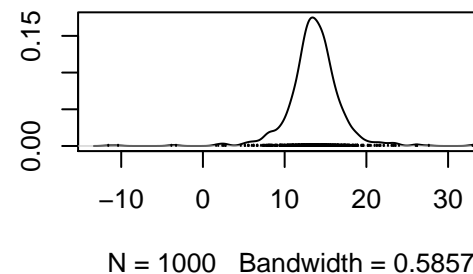
bmcmc <- mcmc(bgibbs) ## convert to MCMC object
plot(bmcmc)           ## mcmc history and density plot

```

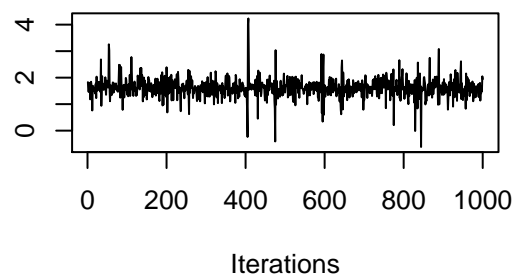
Trace of var1



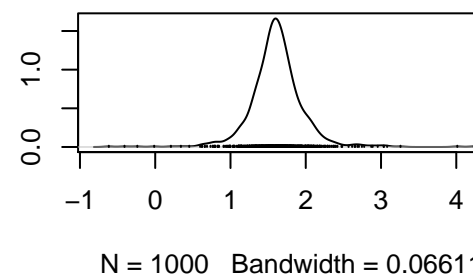
Density of var1



Trace of var2

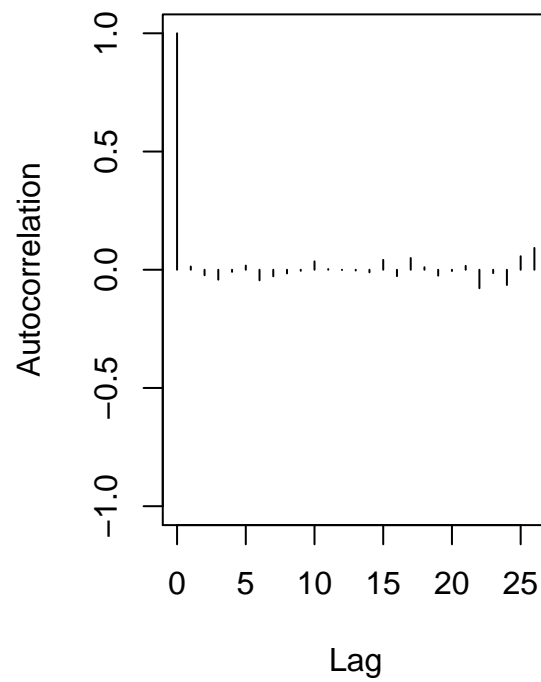
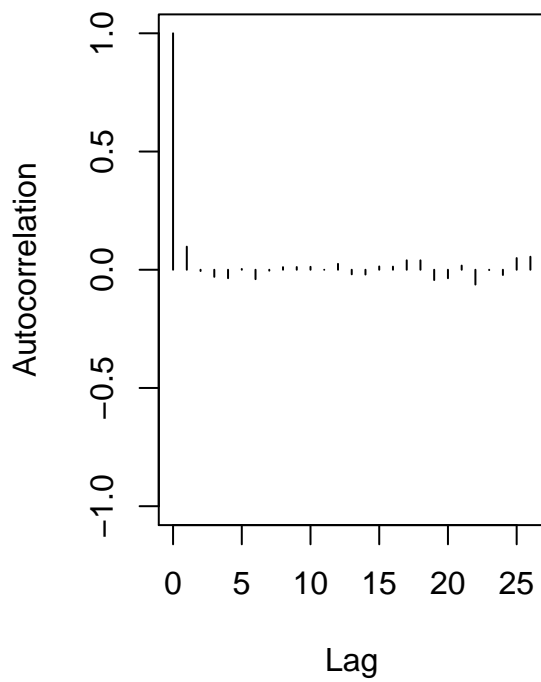


Density of var2

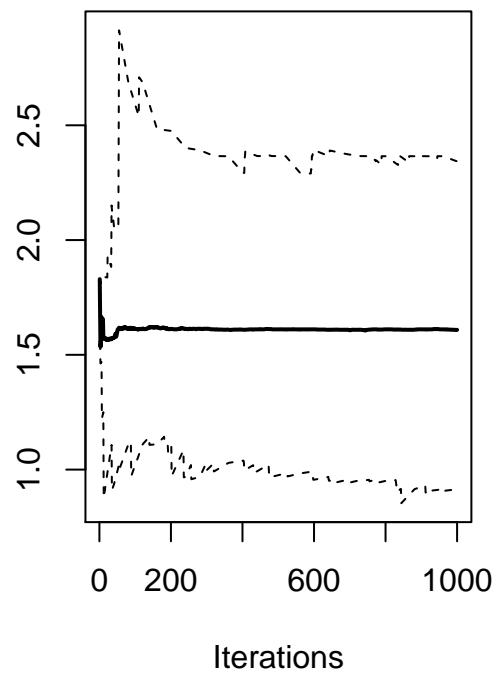
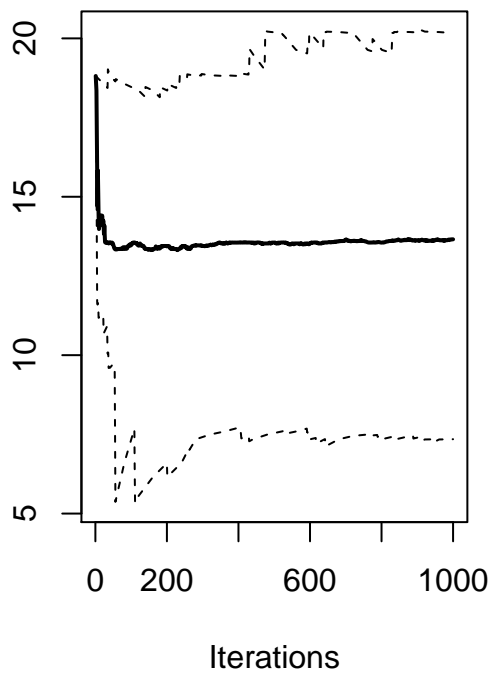


diagnostics of the MCMC

```
autocorr.plot(bmcmc)    ## autocorrelation
```



```
cumuplot(bmcmc)    ## quantile plot
```



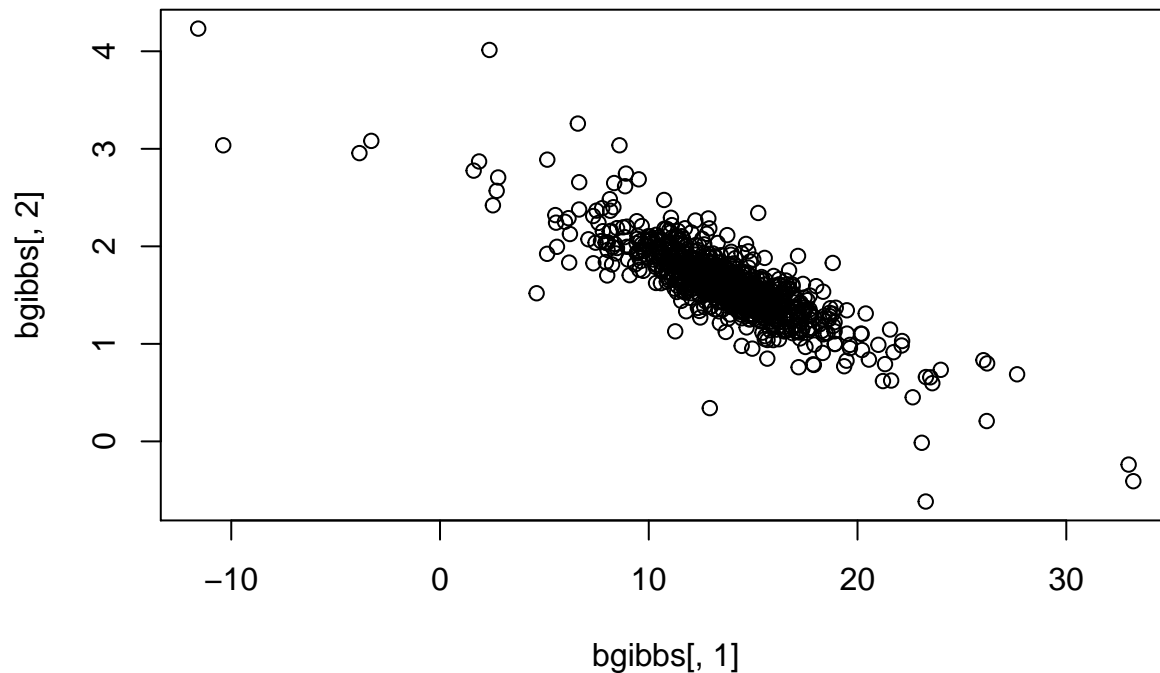
```
1-rejectionRate(bmcmc)  ## acceptance rate
```

```
## var1 var2
##    1    1
```

```
summary(bmcmc)      ## summary table
```

```
##
## Iterations = 1:1000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##      Mean      SD Naive SE Time-series SE
## [1,] 13.63 3.340  0.1056      0.1165
## [2,]  1.61 0.369  0.0117      0.0117
##
## 2. Quantiles for each variable:
##
##      2.5%  25%  50%  75% 97.5%
## var1 7.346 12.20 13.66 15.14 20.17
## var2 0.914  1.44  1.61  1.77  2.34
```

```
plot(bgibbs[,1],bgibbs[,2]) ## pairs plot to evaluate parameter correlation
```



```
beg = 1
thin = 1
bmcmc <- mcmc(bgibbs[seq(from=beg,to=ngibbs,by=thin)])
```

Task 1

```
summary(lm( y ~ x1 ))

##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      1      2      3      4      5
## 2.608  1.159  0.654 -1.965 -2.456
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   13.645      2.042     6.68  0.0068 **
## x1             1.614      0.223     7.25  0.0054 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.48 on 3 degrees of freedom
## Multiple R-squared:  0.946, Adjusted R-squared:  0.928
## F-statistic: 52.6 on 1 and 3 DF, p-value: 0.00541
```

Regression Credible Intervals

```
## credible and prediction intervals
xpred <- 0:20                      ## sequence of x values we're going to
npred <- length(xpred)             ##      make predictions for
ypred <- matrix(0.0,nrow=ngibbs,ncol=npred) ## storage for predictive interval
ycred <- matrix(0.0,nrow=ngibbs,ncol=npred) ## storage for credible interval

for(g in seq(from=beg,to=ngibbs,by=thin)){
  Ey <- bgibbs[g,1] + bgibbs[g,2] * xpred
  ycred[g,] <- Ey
  ypred[g,] <- rnorm(npred,Ey,sqrt(sgibbs[g]))
}

ci <- apply(ycred,2,quantile,c(0.025,0.5,0.975)) ## credible interval and median
pi <- apply(ypred,2,quantile,c(0.025,0.975))     ## prediction interval

plot(x1,y,cex=0.5,xlim=c(0,20),ylim=c(0,50))
lines(xpred,ci[1,],col=3,lty=2) ## lower CI
lines(xpred,ci[2,],col=3,lwd=2) ## median
lines(xpred,ci[3,],col=3,lty=2) ## upper CI
lines(xpred,pi[1,],col=4,lty=2) ## lower PI
lines(xpred,pi[2,],col=4,lty=2) ## upper PI
abline(b0,b1)                    ## true model
```

