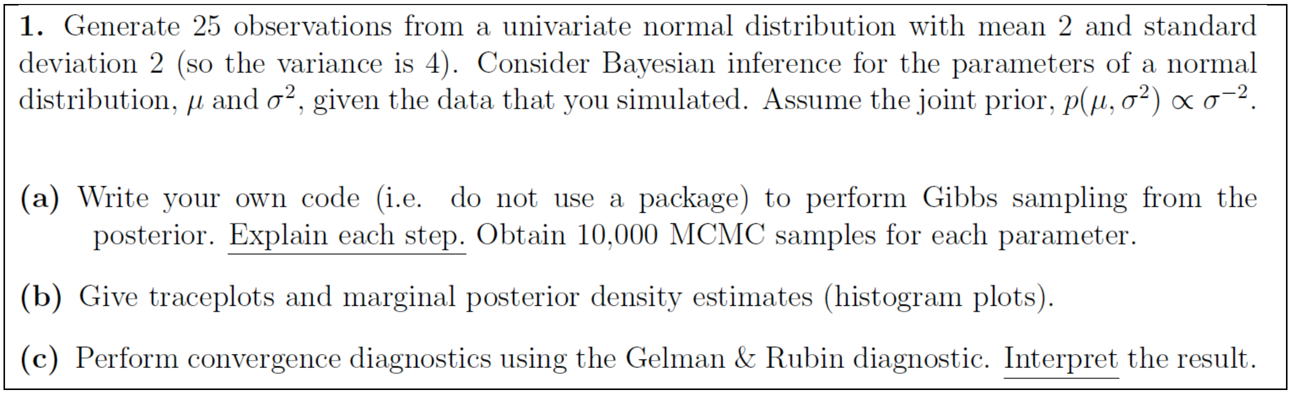
Math 459 Bayesian Statistics
Final exam

Qian Liu
q.liu@wustl.edu



1. Generate 25 observations from a univariate normal distribution with mean 2 and standard deviation 2 (so the variance is 4). Consider Bayesian inference for the parameters of a normal distribution, μ and σ^2, given the data that you simulated. Assume the joint prior, $p(\mu, \sigma^2) \propto \sigma^{-2}$.

(a) Write your own code (i.e. do not use a package) to perform Gibbs sampling from the posterior. Explain each step. Obtain 10,000 MCMC samples for each parameter.

(b) Give traceplots and marginal posterior density estimates (histogram plots).

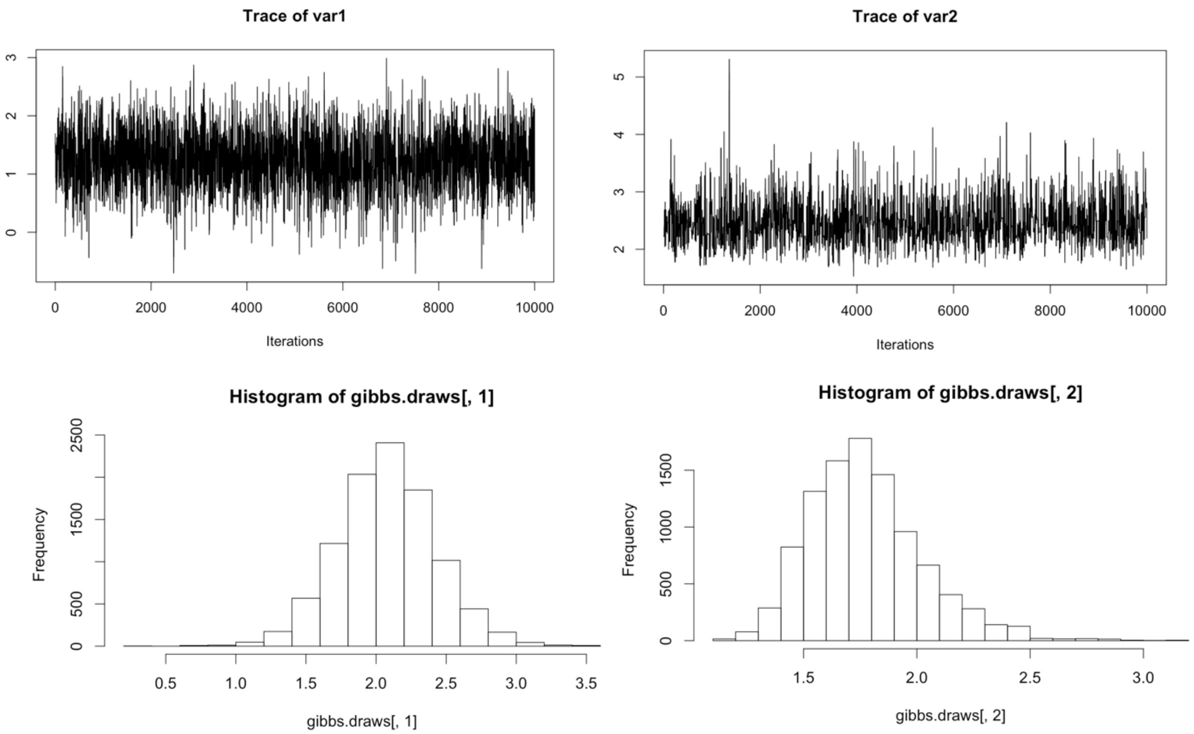
(c) Perform convergence diagnostics using the Gelman & Rubin diagnostic. Interpret the result.

Sol:

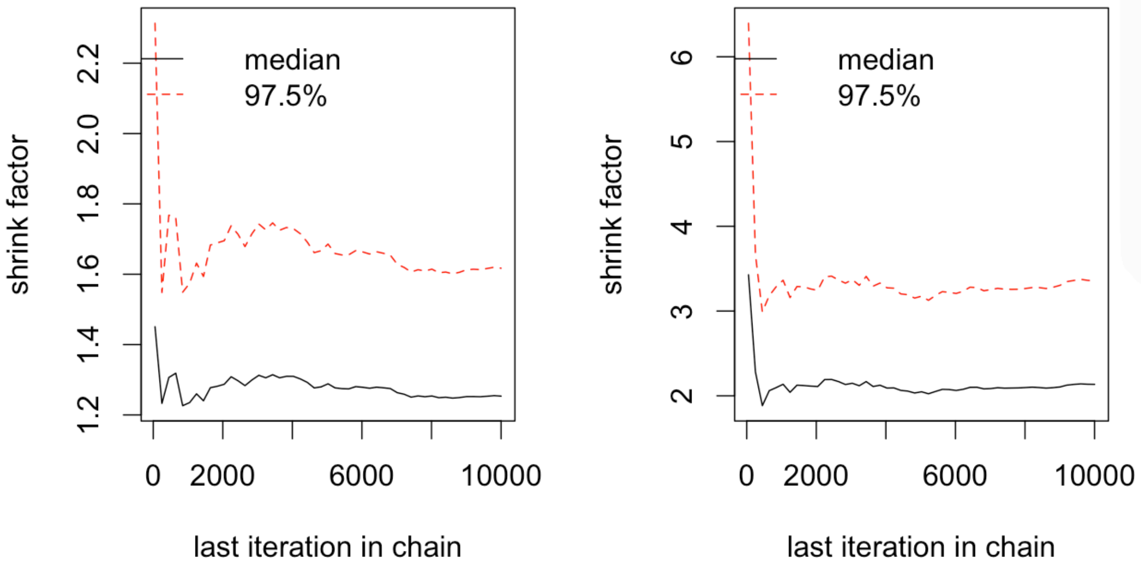
(a) Code

```
> Gibbs <- function(sims, burnin){
+  library(coda);
+  # generate some data
+  n = 25;
+  data_X = rnorm(n, mean = 2, sd = 2);
+  #posterior probability
+  post_prob = function(x){
+    prior = x[2]^(-2);
+    m=1;
+    for(j in data_X){
+      m = m * dnorm(j, mean=x[1],sd=x[2])
+    }
+    prob_pos=m*prior;
+    return (prob_pos)
+  }
+  #initial point
+  theta.current <- c(2, 2)
+  #define RWMH function
+  theta.mh <- matrix(NA, nrow = sims, ncol = 2)
+  theta.update <- function(index,theta.current) {
+    if (index == 1){
+      theta.can <- rnorm(1, theta.current[1], 1);
+      theta.can =c(theta.can,theta.current[2]);
+    }
+    else{
+      theta.can <- rgamma(1, theta.current[2], 1);
+      theta.can =c(theta.current[1],theta.can);
+    }
+    accept.prob <- min(1,post_prob(theta.can)/post_prob(theta.current))
+    if(runif(1) <= accept.prob)
+      theta.can
+    else
+      theta.current
+   }
+
+   for(i in 1:sims){
+    theta.current <- theta.update(1, theta.current);
+    theta.mh[i,1] = theta.current[1];
+    theta.current<- theta.update(2, theta.current);
+    theta.mh[i,2] = theta.current[2]
+   }
+   res <- mcmc(theta.mh[(1+burnin):sims,])
+   return(res)
+ }
> # With respect to gibbs sampling
> gibbs.draws <- Gibbs(sims  = 11000, burnin = 1000)
Acceptance Rate: 0.4407441 0.2317232
> # Plot the trace and the marginal posterior density
> plot(gibbs.draws)
> hist(gibbs.draws[,1])
> hist(gibbs.draws[,2])
>
> gibbs.draws1 <- Gibbs(sims  = 11000, burnin = 1000)
> gibbs.draws2 <- Gibbs(sims  = 11000, burnin = 1000)
> gibbs.draws3 <- Gibbs(sims  = 11000, burnin = 1000)
> gibbs.draws4 <- Gibbs(sims  = 11000, burnin = 1000)
> gibbs.draws5 <- Gibbs(sims  = 11000, burnin = 1000)
> gibbs.list <- list(gibbs.draws1, gibbs.draws2, gibbs.draws3, gibbs.draws4, gibbs.draws5)
> # Plotting how PSRF Changes through Iteration
> gelman.diag(gibbs.list)
Potential scale reduction factors:
    Point est. Upper C.I.
[1,]       1.19       1.45
[2,]       1.30       1.68
Multivariate psrf

1.33
> gelman.plot(gibbs.list)
```

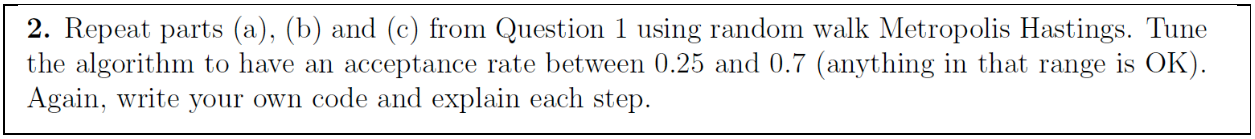
b)



c)



Gelman and Rubin diagnostic: This diagnostic gives the median scale reduction factor and its 97.5% quantile (the PSRF is estimated with uncertainty since the chain lengths are finite), and also reports multivariate scale reduction factor. And we can see how the potential scale reduction factor changes through the iterations using the gelman:plot() function. Results show a good convergence of the constructed chain.



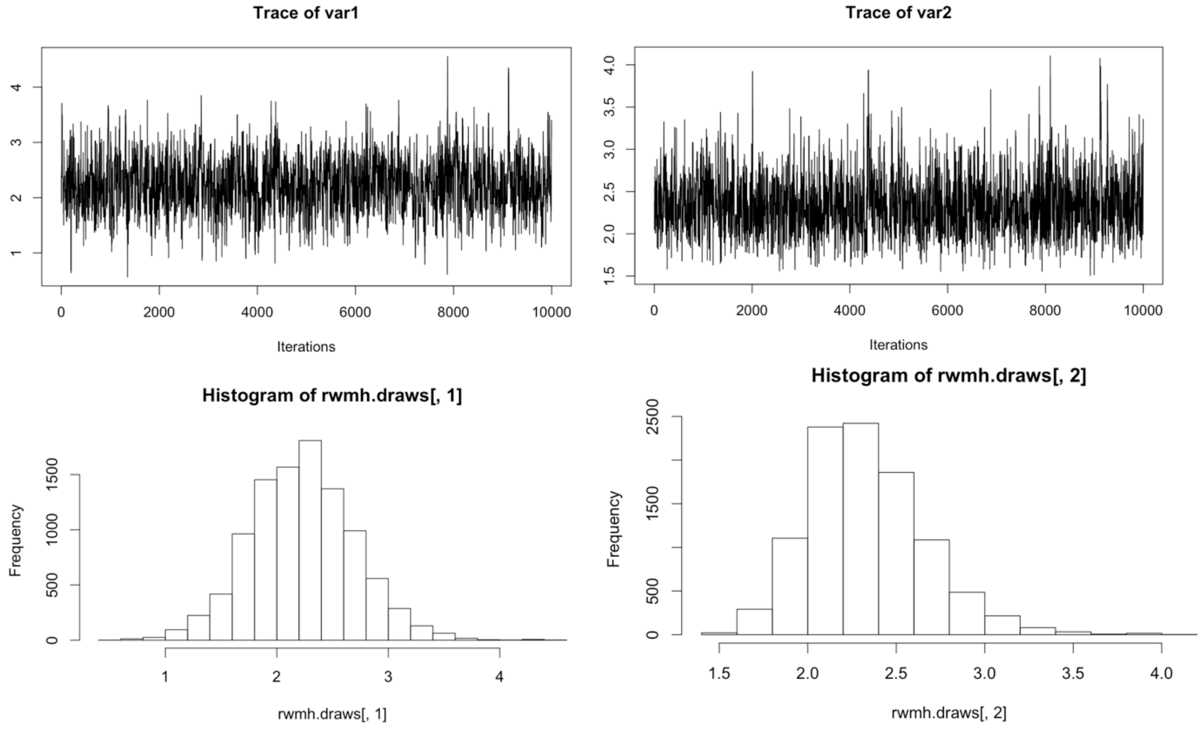
2. Repeat parts (a), (b) and (c) from Question 1 using random walk Metropolis Hastings. Tune the algorithm to have an acceptance rate between 0.25 and 0.7 (anything in that range is OK). Again, write your own code and explain each step.

Sol:
a)code

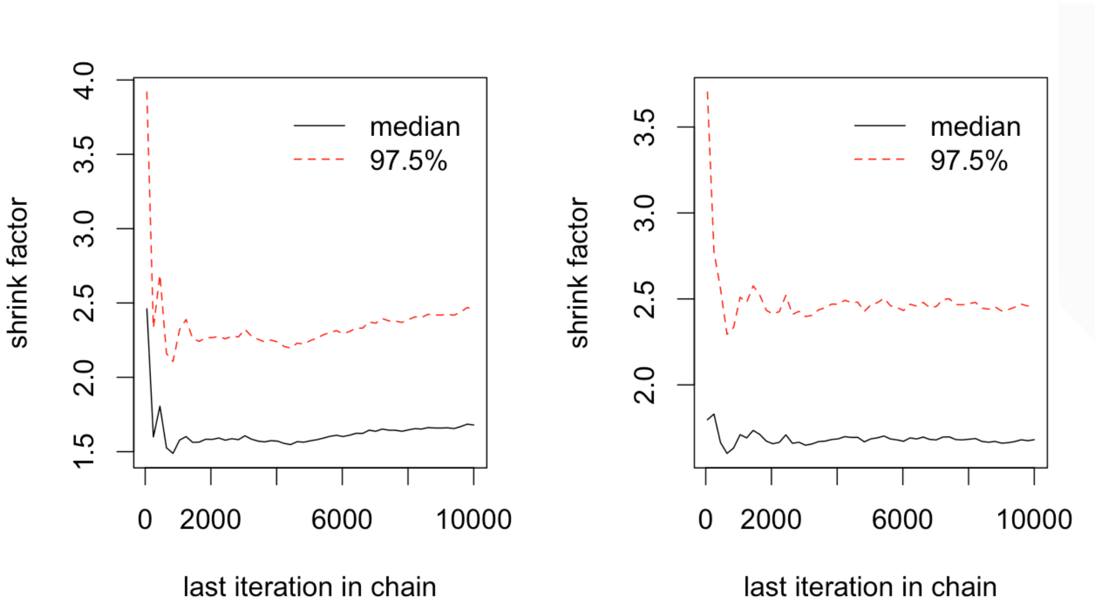
```
> TwoDRW <- function(sims, burnin){
+   library(coda);
+   require(MASS);
+   # generate some data
+   n = 25;
+   data_X = rnorm(n, mean = 2, sd = 2);
+   #posterior probability
+   post_prob = function(x){
+     prior = x[2]^(-2);
+     m=1;
+     for(j in data_X){
+       m = m * dnorm(j, mean=x[1],sd=x[2])
+     }
+     prob_pos=m*prior;
+     return (prob_pos)
+   }
+
+   #initial point
+   start =c(2,2);
+
+   # use 2-D normal as candidate distribution and sepcify variance-convariance matrix
+   mle_fit <- fitdistr(data_X, "normal")
+   cand.sd <- unname(diag(mle_fit$sd))
+
+   # define the random walk Metropolis-Hastings sampling function
+   theta.cur <- start
+   draws <- matrix( NA, nrow = sims, ncol = 2)
+   for(i in 1:sims){
+     theta.can <- mvrnorm(1, theta.cur, cand.sd)
+     if (theta.can[1]>0 && theta.can[2]>0)
+       if(runif(1) <= min(1, post_prob(theta.can)
+                         /post_prob(theta.cur)))
+         theta.cur <- theta.can
+       draws[i, ] <- theta.cur
+   }
+   return(mcmc(unname(draws[(burnin + 1):sims, ])))
+ }
>
> # With respect to Two D random walk Metropolis Hastings
> rwmh.draws <- TwoDRW(sims  = 13000, burnin = 3000)
> cat("Acceptance Rate:", 1-rejectionRate(rwmh.draws), "\n")
```

```
Acceptance Rate: 0.3713371 0.3713371
>
> # Plot the trace and the marginal posterior density
> plot(rwmh.draws)
> hist(rwmh.draws[,1])
> hist(rwmh.draws[,2])
>
> rwmh.draws1 <- TwoDRW(sims  = 13000, burnin = 3000)
> rwmh.draws2 <- TwoDRW(sims  = 13000, burnin = 3000)
> rwmh.draws3 <- TwoDRW(sims  = 13000, burnin = 3000)
> rwmh.draws4 <- TwoDRW(sims  = 13000, burnin = 3000)
> rwmh.draws5 <- TwoDRW(sims  = 13000, burnin = 3000)
> rwmh.list <- list(rwmh.draws1, rwmh.draws2, rwmh.draws3, rwmh.draws4, rwmh.draws5)
>
> # Plotting how PSRF Changes through Iteration
> gelman.diag(rwmh.list)
Potential scale reduction factors:

     Point est. Upper C.I.
[1,]       1.66       2.38
[2,]       1.19       1.44

Multivariate psrf
1.7
> gelman.plot(rwmh.list)
```

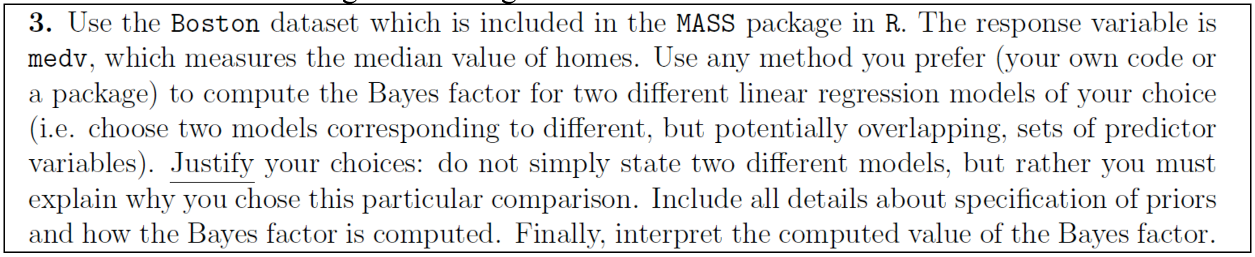
(b)



(c)



Gelman and Rubin diagnostic: This diagnostic gives the median scale reduction factor and its 97.5% quantile (the PSRF is estimated with uncertainty since the chain lengths are finite), and also reports multivariate scale reduction factor. And we can see how the potential scale reduction factor changes through the iterations using the gelman:plot() function. Results show variance and mean have a good convergence of the constructed chain.


3. Use the `Boston` dataset which is included in the `MASS` package in `R`. The response variable is `medv`, which measures the median value of homes. Use any method you prefer (your own code or a package) to compute the Bayes factor for two different linear regression models of your choice (i.e. choose two models corresponding to different, but potentially overlapping, sets of predictor variables). Justify your choices: do not simply state two different models, but rather you must explain why you chose this particular comparison. Include all details about specification of priors and how the Bayes factor is computed. Finally, interpret the computed value of the Bayes factor.

Ans:

Code

```
> # initiate date
> data("Boston")
> #use MCMCpack to do the regression
> library(MCMCpack)
>
> #Bayesian Regression Model with variable crim+indus+nox+rm
> fit1 <-MCMCregress(medv~crim+indus+nox+rm,data=Boston,thin=2, burnin=1000,
+                  mcmc =50000, b0=0, B0=.001,marginal.likelihood="Laplace")
> # Bayesian Regression Model with with variable crim+indus+nox+rm+age, compare
whether it make sense to take age into account.
> fit2 <-MCMCregress(medv~crim+indus+nox+rm+age,data=Boston,thin=2, burnin=1000,
+                  mcmc =50000, b0=0, B0=.001,marginal.likelihood="Laplace")
> BayesFactor(fit1,fit2)
The matrix of Bayes Factors is:
      fit1 fit2
fit1 1.0000  435
```

```
fit2 0.0023    1

The matrix of the natural log Bayes Factors is:
      fit1 fit2
fit1  0.00 6.08
fit2 -6.08 0.00

 fit1 :
   call =
MCMCregress(formula = medv ~ crim + indus + nox + rm, data = Boston,
    burnin = 1000, mcmc = 50000, thin = 2, b0 = 0, B0 = 0.001,
    marginal.likelihood = "Laplace")
   log marginal likelihood =  -1662.004

 fit2 :
   call =
MCMCregress(formula = medv ~ crim + indus + nox + rm + age, data = Boston,
    burnin = 1000, mcmc = 50000, thin = 2, b0 = 0, B0 = 0.001,
    marginal.likelihood = "Laplace")
   log marginal likelihood =  -1668.079
```

We assume standard, semi-conjugate priors:
beta ~ N(b0,B0^(-1))   , *sigma^(-2) ~ Gamma(c0/2, d0/2)*,
where b0 = 0, B0 = 0.001, c0 = 0.001, d0 = 0.001, *beta* and *sigma^(-2)* are assumed *a priori* independent. It seems that crim and indus and nox is relevant to medv. The first Bayesian Regression Model with formula = medv ~ crim + indus + nox + rm. I am not sure whether age is relevant to medv, so I design the second Bayesian Regression Model with formula = medv ~ crim + indus + nox + rm + age, comparing with the first model to find out whether it make sense to take age into account. The marginal likelihood can easily obtained by the result of log marginal likelihood in the output. And Bayes Factors:

$$K = \frac{\Pr(D|M_1)}{\Pr(D|M_2)} = \frac{\int \Pr(\theta_1|M_1)\Pr(D|\theta_1, M_1)\,d\theta_1}{\int \Pr(\theta_2|M_2)\Pr(D|\theta_2, M_2)\,d\theta_2}.$$

,From the result in the matrix of Bayes Factors,

```
      fit1 fit2
fit1 1.0000  435
fit2 0.0023    1
```

we know the model fit1 with formula = medv ~ crim + indus + nox + rm is much better than the model fit3 formula = medv ~ crim + indus + nox + rm + age, with strength of evidence, decisive according to Je_reys scale and very strong according to Kass & Raftery scale. That is to say first Bayesian Regression Model is better. Age is not relevant to medv.

4. Using concepts and definitions from decision theory, compare and contrast minimax and Bayes decision rules. Explain in what sense the posterior mean is an optimal point estimator.