Homework 16

Ziyao Yang

3/26/23

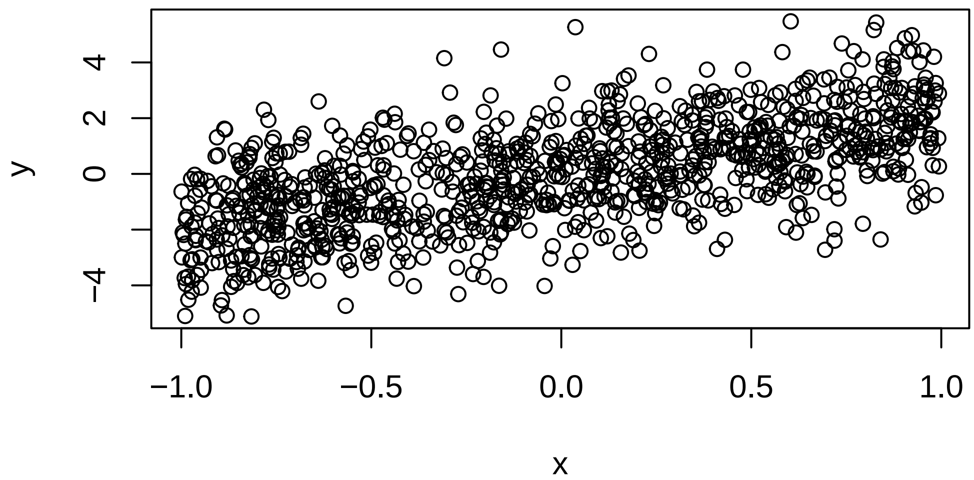
```r
rm(list = ls(all.names = TRUE)) #will clear all objects includes hidden objects.
gc() #free up memrory and report the memory usage.
```

```
          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
Ncells  569064 30.4    1291250   69         NA   669294 35.8
Vcells 1040814  8.0    8388608   64      32768  1839866 14.1
```

```r
library(readr)
regression_example <- read_csv("regression_example.csv")
```

```
Rows: 1000 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (2): x, y

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
x <- regression_example$x
y <- regression_example$y
plot(x,y)
```




```
lm <- lm(y~x-1,regression_example)
summary(lm)
```

```
Call:
lm(formula = y ~ x - 1, data = regression_example)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.1005 -0.9906 -0.0073  1.0555  5.1919 

Coefficients:
  Estimate Std. Error t value Pr(>|t|)    
x  1.98469    0.07921   25.05   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.486 on 999 degrees of freedom
Multiple R-squared:  0.3859,    Adjusted R-squared:  0.3853 
F-statistic: 627.8 on 1 and 999 DF,  p-value: < 2.2e-16
```

```
# Frequentist estimate of beta
beta_hat = sum((x - mean(x)) * (y - mean(y))) / sum((x - mean(x))^2)

# Frequentist estimate of sigma squared
sigma2_hat = sum((y - beta_hat * x)^2) / (length(y) - 1)
```

ℹ [A] & [C] Using your expression from the previous part, proposals of the following form:

- $\beta \to \beta + Unif(-.2, .2)$
- $\sigma^2 \to \sigma^2 + Unif(-.1, .1)$

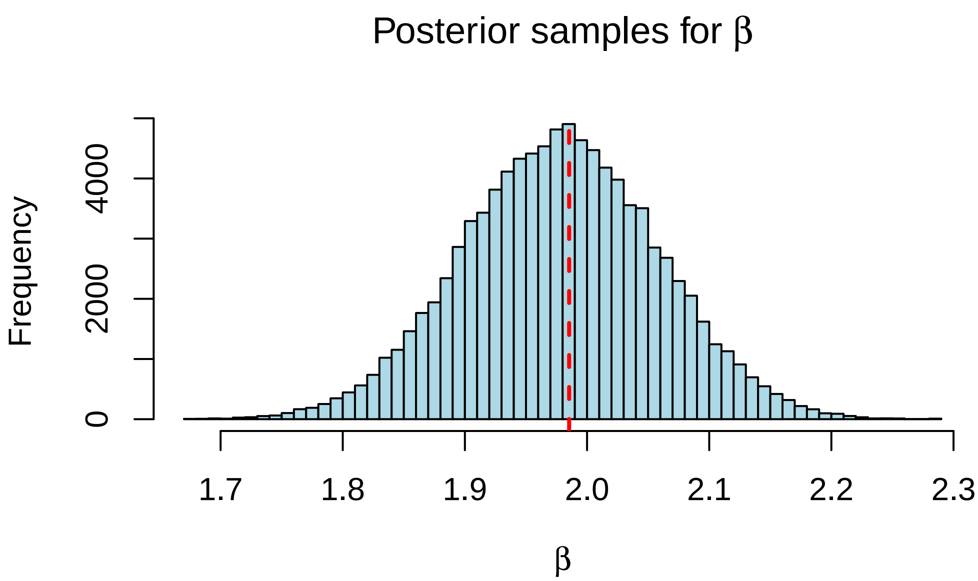
And the starting position $\beta_0 = 2$, $\sigma_0^2 = 2$, first derive a Metropolis Hastings algorithm to approximate the posterior distribution $\beta, \sigma^2 | Y_1, \ldots, Y_n$, and then write codes to simulate a Markov chain of length 10,000 using this algorithm. Use `set.seed(440)` in your simulations.

```
#initialize MH
beta0 = 2
sigma02 = 2

# Likelihood function on the log scale
likelihood = function(param){
  beta = param[1]
  var = param[2]
  pred = beta*x
  singlelikelihoods = dnorm(y, mean = pred, sd = sqrt(var), log = T)
  sumll = sum(singlelikelihoods)
  return(sumll)
}

# Prior distribution on the log scale
prior = function(param){
  beta = param[1]
  var = param[2]
  bprior = dnorm(beta, sd = sqrt(var), log = T)
  sdprior = log(1/var)
  return(bprior+sdprior)
}

# Posterior distribution on the log scale
posterior = function(param){
  return (likelihood(param) + prior(param))
}

proposalfunction = function(param){
  return(param + runif(2,c(-0.2,-0.1), c(0.2,0.1)))
}

run_metropolis_MCMC = function(startvalue, iterations){
  chain = array(dim = c(iterations+1,2))
  chain[1,] = startvalue
```

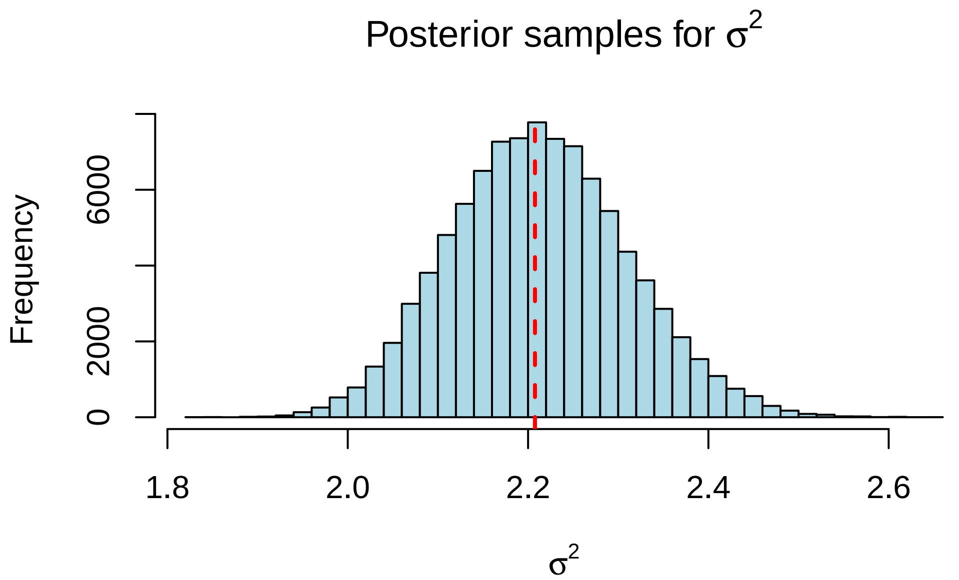
```r
  for (i in 1:iterations){
    proposal = proposalfunction(chain[i,])
    probab = exp(posterior(proposal) - posterior(chain[i,]))
    if (runif(1) < probab){
      chain[i+1,] = proposal
    }else{
      chain[i+1,] = chain[i,]
    }
  }
  return(chain)
}

set.seed(440)
startvalue = c(beta0,sigma02)
chain = run_metropolis_MCMC(startvalue, 100000)

burnIn = 5000
beta_samples = chain[-(1:burnIn),1]
sigma_samples = chain[-(1:burnIn),2]
```

[C] Plot the histograms of your simulated posterior samples for and 2 as two separate plots. Add a vertical line on each plot to indicate the frequentist estimates $\hat{\beta}$ and $\hat{\sigma^2}$

```r
# Plot the histogram of beta with the frequentist estimate
hist(beta_samples,
     breaks = 50,
     main = expression("Posterior samples for " * beta),
     xlab = expression(beta),
     col = "lightblue",
     border = "black")
abline(v = beta_hat, col = "red", lwd = 2, lty = 2)
```




```r
# Plot the histogram of sigma^2 with the frequentist estimate
hist(sigma_samples,
     breaks = 50,
     main = expression("Posterior samples for " * sigma^2),
     xlab = expression(sigma^2),
     col = "lightblue",
     border = "black")
abline(v = sigma2_hat, col = "red", lwd = 2, lty = 2)
```


Posterior samples for σ²
Frequency
0
2000
6000
1.8
2.0
2.2
2.4
2.6
σ²

[C] Using your simulated posterior samples, calculate a 95% credible interval for β

```
# Calculate the 95% posterior credible interval for b
ci_95 <- quantile(beta_samples, probs = c(0.025, 0.975))
ci_95
```

```
    2.5%    97.5% 
1.821517 2.134177 
```

```
hist(beta_samples,
     breaks = 50,
     main = expression("95% posterior credible interval for " * beta),
     xlab = expression(beta),
     col = "lightblue",
     border = "black")
abline(v = ci_95, col = "red", lwd = 2, lty = 2)
```