Econometrics Homework 4

Utpalraj Kemprai
MDS202352

Question 1

We have a single observation $y = (y_1, y_2)$ from the distribution,

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$

and we assume a uniform distribution on θ.

Part (a)

As we have assume a uniform distribution on θ, we have,

$$\pi(\theta) = c$$

where c is some constant.

The likelihood for the single observation is,

$$f(y|\theta) = \frac{1}{2\pi\sqrt{(1-\rho^2)}} e^{-\frac{1}{2(1-\rho^2)}((y_1-\theta_1)^2 - 2\rho(y_1-\theta_1)(y_2-\theta_2)+(y_2-\theta_2)^2)}$$

The posterior distribution of θ, is

$$\begin{aligned} \pi(\theta|y) &\propto f(y|\theta)\pi(\theta) \\ &\propto \frac{1}{2\pi\sqrt{(1-\rho^2)}} e^{-\frac{1}{2(1-\rho^2)}((y_1-\theta_1)^2 - 2\rho(y_1-\theta_1)(y_2-\theta_2)+(y_2-\theta_2)^2)} c \\ &\propto \frac{1}{2\pi\sqrt{(1-\rho^2)}} e^{-\frac{1}{2(1-\rho^2)}((y_1-\theta_1)^2 - 2\rho(y_1-\theta_1)(y_2-\theta_2)+(y_2-\theta_2)^2)} \end{aligned}$$

Part (b)

Now, for the conditional distribution for θ_1 given θ_2 and y we gather only terms in $\pi(\theta|y)$ that involve θ_1.

Therefore,

$$\begin{aligned} \pi(\theta_1|\theta_2, y) &\propto e^{-\frac{1}{2(1-\rho^2)}((y_1-\theta_1)^2 - 2\rho(y_1-\theta_1)(y_2-\theta_2))} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}((\theta_1-y_1)^2 - 2\rho(\theta_1-y_1)(\theta_2-y_2))} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}(\theta_1^2 - 2\theta_1 y_1 + y_1^2 - 2\rho(\theta_1\theta_2 - \theta_1 y_2 - \theta_2 y_1 + y_1 y_2))} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}(\theta_1^2 - 2\theta_1 y_1 - 2\rho(\theta_1\theta_2 - \theta_1 y_2))} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}(\theta_1^2 - 2\theta_1 y_1 - 2\rho\theta_1(\theta_2 - y_2))} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}(\theta_1^2 - 2\theta_1(y_1 + \rho(\theta_2 - y_2))} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}(\theta_1^2 - 2\theta_1(y_1 + \rho(\theta_2 - y_2)) + (y_1 + \rho(\theta_2 - y_2))^2)} \\ &\propto e^{-\frac{1}{2(1-\rho^2)}(\theta_1 - y_1 - \rho(\theta_2 - y_2))^2} \end{aligned}$$

Therefore $\pi(\theta_1|\theta_2, y)$ is proportional to the kernel of a normal distribution with mean $y_1 + \rho(\theta_2 - y_2)$ and variance $1 - \rho^2$.

So $\pi(\theta_1|\theta_2, y) \sim N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2)$.

Replicating the above calculation for $\theta_2|\theta_1, y$ and from the symmetry of y_1 and y_2 it follows that $\pi(\theta_2|\theta_1, y) \sim N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)$

Part (c)

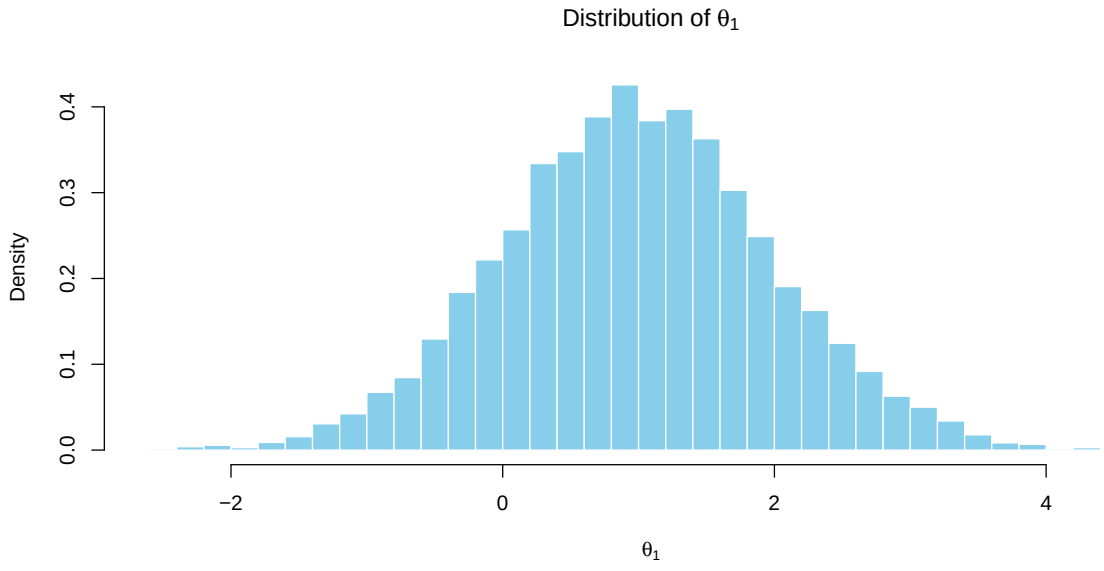
Given $y = (1, 0.5)$ and $\rho = 0.8$. So from Part (b), it follows that So $\theta_1|\theta_2, y \sim N(0.6 + 0.8\theta_2, 0.36)$ and $\theta_2|\theta_1, y \sim N(-0.3 + 0.8\theta_1, 0.36)$.

Below is the code for sampling from the posterior distribution $\theta|y$ using Gibbs Sampling.

```
# R Code for Gibbs sampling
n = 10000 # number of iterations
B = 1000 # burn in period
set.seed(13) # for reproducibility

# initialize storage variables
theta1 = rep(0.1,n)
eta1 = rep(0.1,n)
theta2 = rep(0.1,n)
eta2 = rep(0.1,n)

# Gibbs Sampling loop
for(i in 2:n){
  # draw theta1 and theta2 from their conditional distributions
  theta1[i] = rnorm(1, mean = 0.6+0.8*theta2[i-1], sd = sqrt(0.36))
  theta2[i] = rnorm(1, mean = -0.3+0.8*theta1[i], sd = sqrt(0.36))
  # standardized to get eta1 and eta2
  eta1[i] = (theta1[i] - (0.6+0.8*theta2[i-1]))/sqrt(0.36)
  eta2[i] = (theta2[i] - (-0.3+0.8*theta1[i]))/sqrt(0.36)
}
```

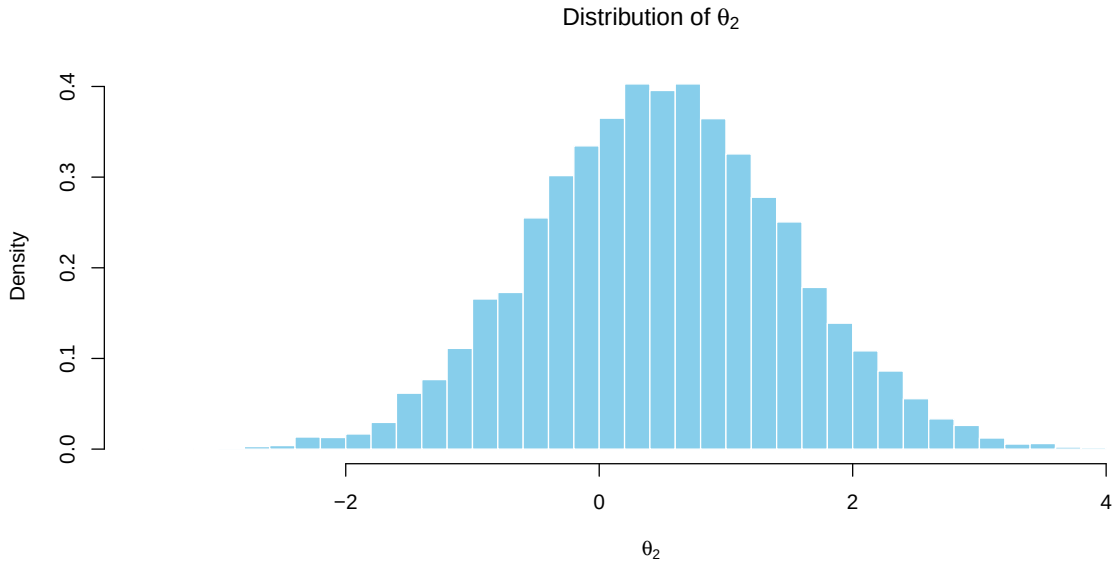
Code for plotting the distribution of θ_1

```
# Plotting distribution of theta1
hist(theta1[(B+1):n], main = expression("Distribution of "* theta[1]),
     xlab = expression(theta[1]),col = "skyblue", border = "white",
     probability = TRUE, breaks = 30, xlim = c(min(theta1),max(theta1)))
```



Code for plotting the distribution of θ_2

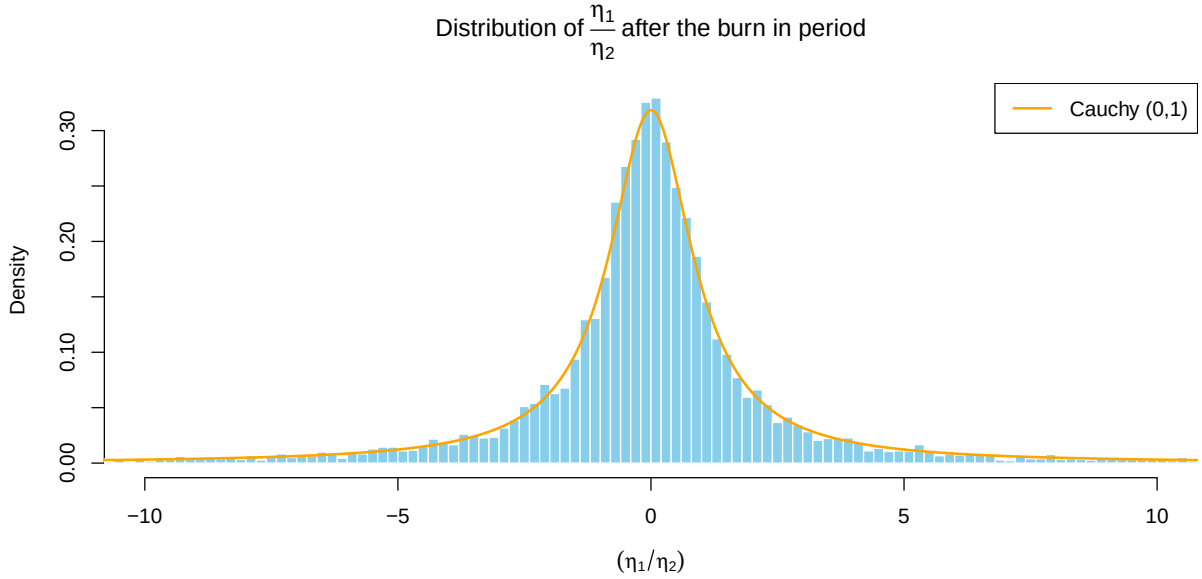
```
# Plotting distribution of theta2
hist(theta2[(B+1):n], main = expression("Distribution of "* theta[2]),
     xlab = expression(theta[2]), col = "skyblue", border = "white",
     probability = TRUE, breaks = 30, xlim = c(min(theta2),max(theta2)))
```



Part (d)

Code for plotting the distribution of $\frac{\eta_1}{\eta_2}$

```
# Plotting the distribution of theta1/theta2 after burn in
proxy = eta1[(B+1):n]/eta2[(B+1):n]
hist(proxy, main = expression("Distribution of "* frac(eta[1],eta[2]) *
                              " after the burn in period"),
     col = "skyblue", border = "white", xlab = expression((eta[1]/eta[2])),
     probability = TRUE, breaks = 50000,xlim = c(-10,10))

# Overlay the pdf of Cauchy(0,1)
x0 = seq(from = -11, to = 11, by = 0.01)
lines(x0, dcauchy(x0, 0, 1),col = 'orange', lwd=2)
legend("topright", legend = c("Cauchy (0,1)"), col = c("orange"), lty = 1,lwd = 2)
```



Observation

While doing Gibbs Sampling, for constructing η_1 and η_2, θ_1 and θ_2 were standardized in each iteration and hence they are samples from a standard Normal distribution, $N(0,1)$.

As,

$$\theta_1^{(i)} \sim N(0.6 + 0.8\theta_2^{(i-1)}, 0.36)$$
$$\eta_1^{(i)} = \frac{\theta_1^{(i)} - (0.6 + 0.8\theta_2^{(i-1)})}{\sqrt{0.36}} \sim N(0,1)$$
$$\theta_2^{(i)} \sim N(-0.3 + 0.8\theta_1^{(i)}, 0.36)$$
$$\eta_2^{(i)} = \frac{\theta_2^{(i)} - (-0.3 + 0.8\theta_1^{(i)})}{\sqrt{0.36}} \sim N(0,1)$$

Also the correlation between η_1 and η_2 is very small (close to zero) after the burn in period:

```
# correlation between eta1 and eta2
cor(eta1[(B+1):n],eta2[(B+1):n])
```

```
## [1] -0.001269805
```

Distribution of $\frac{\eta_1}{\eta_2}$

In our Gibbs sampling scheme, the updates at each iteration i are defined by:

$$\theta_1^{(i)} = \mu_1(\theta_2^{(i-1)}) + \sigma\,\epsilon_1^{(i)}, \quad \theta_2^{(i)} = \mu_2(\theta_1^{(i)}) + \sigma\,\epsilon_2^{(i)},$$

where:

- $\mu_1(\theta_2^{(i-1)}) = 0.6 + 0.8\theta_2^{(i-1)}$ and $\mu_2(\theta_1^{(i)}) = -0.3 + 0.8\theta_1^{(i)}$ are the conditional means,

- $\sigma = \sqrt{0.36} = 0.6$,
- $\epsilon_1^{(i)}$ and $\epsilon_2^{(i)}$ are independent random variables drawn from the standard normal distribution, i.e., $\epsilon_1^{(i)}, \epsilon_2^{(i)} \sim N(0,1)$.

We then standardize the draws by computing:

$$\eta_1^{(i)} = \frac{\theta_1^{(i)} - \mu_1(\theta_2^{(i-1)})}{\sigma}, \quad \eta_2^{(i)} = \frac{\theta_2^{(i)} - \mu_2(\theta_1^{(i)})}{\sigma}.$$

Substituting the expressions for $\theta_1^{(i)}$ and $\theta_2^{(i)}$ into these equations, we obtain:

$$\eta_1^{(i)} = \frac{\mu_1(\theta_2^{(i-1)}) + \sigma\,\epsilon_1^{(i)} - \mu_1(\theta_2^{(i-1)})}{\sigma} = \epsilon_1^{(i)},$$

$$\eta_2^{(i)} = \frac{\mu_2(\theta_1^{(i)}) + \sigma\,\epsilon_2^{(i)} - \mu_2(\theta_1^{(i)})}{\sigma} = \epsilon_2^{(i)}.$$

Since $\epsilon_1^{(i)}$ and $\epsilon_2^{(i)}$ are independent draws from $N(0,1)$, we have:

$$\eta_1^{(i)} \sim N(0,1) \quad \text{and} \quad \eta_2^{(i)} \sim N(0,1),$$

and, they are independent because $\epsilon_1^{(i)}$ and $\epsilon_2^{(i)}$ are independent

Conclusion: By construction in the Gibbs sampler, $\eta_1^{(i)}$ and $\eta_2^{(i)}$ are independent standard normal random variables. Hence, their ratio $\eta_1^{(i)}/\eta_2^{(i)}$ is the ratio of two independent $N(0,1)$ variables, which follows a standard Cauchy distribution.

Question 2

We wish to generate 10,000 draws from a Beta(3, 4) distribution with U(0, 1) as proposal density with independent chain, where U denotes a Uniform distribution.

Below is the code for our Metropolis-Hastings Algorithm:

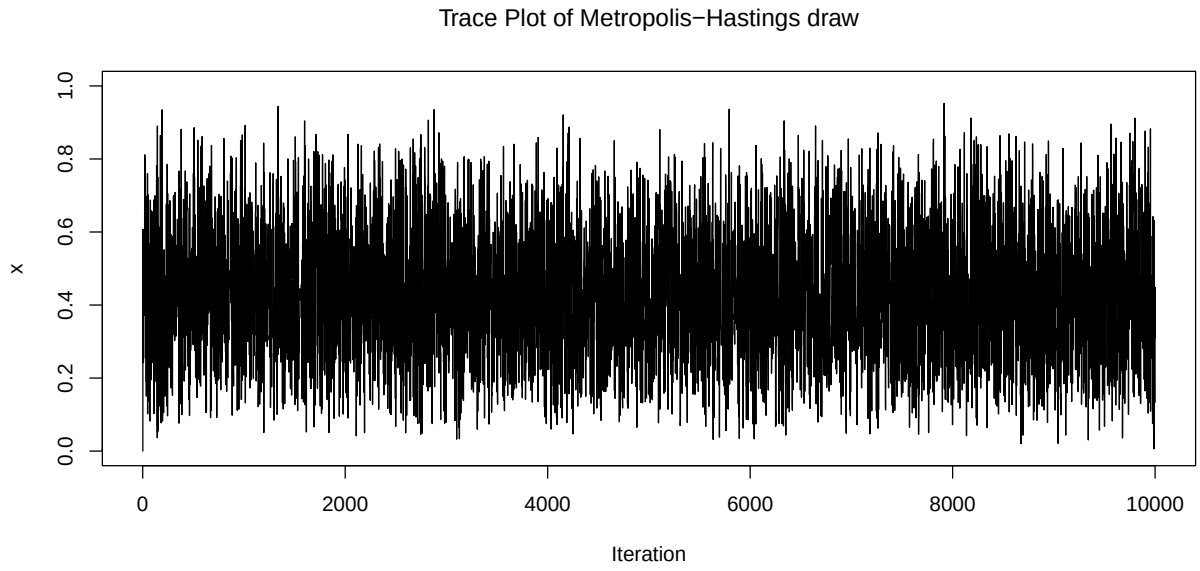
```
# Code for Metropolis-Hastings

n = 10000 # number of iterations
x = rep(0,n) # storage variable for MH draws
accept = 0  # counter for number of acceptance

for(i in 2:n){
  u = runif(2) # draw sample of size 2 from U(0,1)

  # calculate alpha
  alpha = (u[2]^2) * ((1-u[2])^3) / ((x[i-1]^2) *(1-x[i-1])^3)

  # if u[1] <= alpha then accept u[2]
  if(u[1]<=alpha){
    x[i] = u[2]
    accept = accept + 1
  }

  # else reject u[2]
  else{
    x[i] = x[i-1]
  }
}
```

Parameter	value
Acceptance Rate	0.5712
Mean of MH draws	0.4318
Variance of MH draws	0.0306

Table 1: Summary of Metropolis-Hastings draws

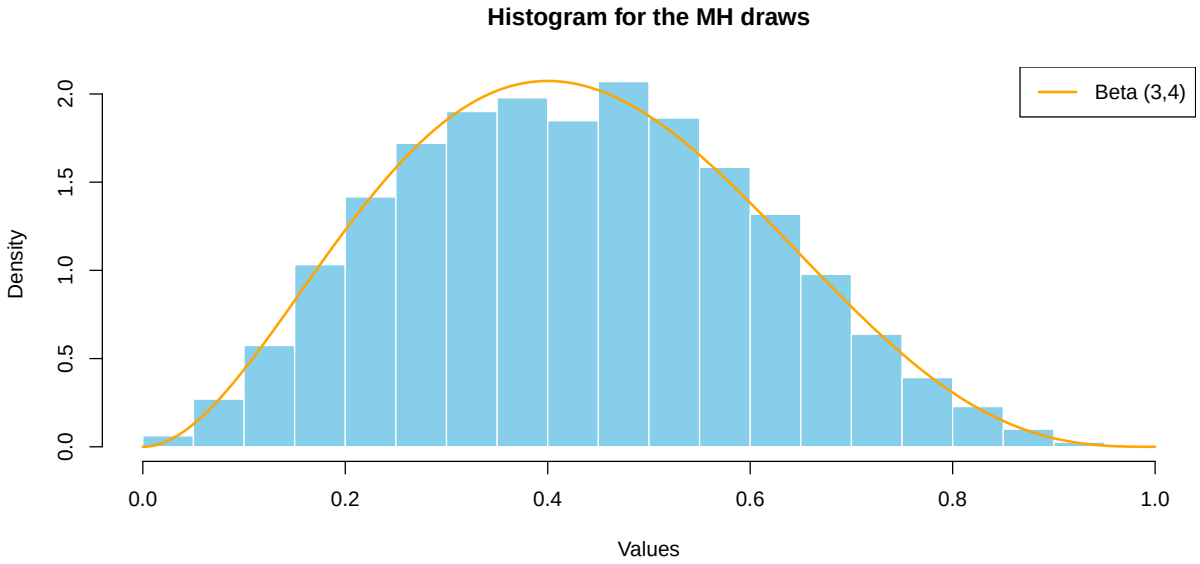
Code for the trace plot of Metropolis-Hastings draws:

```
# plot the trace plot
plot(x, type = "l",
     main = expression("Trace Plot of Metropolis-Hastings draw"),
     xlab = "Iteration", ylim = c(0, 1))
```



Code for the histogram of Metropolis-Hastings draws

```r
# Histogram of MH draws
hist(x, main = "Histogram for the MH draws",
     xlab = "Values", col = "skyblue", border = "white", probability = TRUE,
     breaks = 30, xlim = c(0,1))

# Overlay the pdf of Beta(3,4)
x0 = seq(0, 1, length.out = 10000)
lines(x0,dbeta(x0, 3, 4),col = 'orange', lwd=2)
legend("topright", legend = c("Beta (3,4)"), col = c("orange"), lty = 1,lwd = 2)
```



For a Beta(3,4), the theoretical mean is $\frac{3}{3+4} \approx 0.4286$ and the theoretical variance is $\frac{3\times4}{(3+4)^2(3+4+1)} \approx 0.0306$.

We see observe that the mean and variance from the MH draws are approximately equal to their theoretical counterpart.

Question 3

Our response variable is the number of hours worked (WHRS). The covariates include a constant, number of children less than six years old at home (childl6), number of children between six and eighteen years old at home (childg6), the woman's age (age), and the husband's yearly wage (huswage).

We assume the following prior distributions:

$\beta \sim N_k(0, 1000 * I_k), \sigma^2 \sim \text{IG}(\frac{\alpha_0}{2}, \frac{\delta_0}{2})$, where $\alpha_0 = 100000$ and $\delta_0 = 10$.

We wish to the run the MCMC chain for 20,000 iterations after a burn-in of 5,000 iterations.

Part (a)

The posterior distribution of (β, σ^2) can be obtained as product of the likelihood and prior distributions as,

$$\pi(\beta, \sigma^2|y) \propto (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)\right] \times \exp\left[-\frac{1}{2 \times 1000}\beta'\beta\right] \times \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_0}{2}+1} \exp\left[-\frac{\delta_0}{2\sigma^2}\right]$$

The conditional posteriors are derived by collecting expression for one parameter at a time from the the joint posterior while holding the remaining parameters fixed.

The conditional posterior for β is,

$$\pi(\beta|\sigma^2, y) \propto \exp\left[-\frac{1}{2}(\beta - \bar{\beta})'B_1^{-1}(\beta - \bar{\beta})\right]$$

where $B_1 = [\sigma^{-2}X'X + \frac{1}{1000}I_k]^{-1}$ and $\bar{\beta} = \sigma^{-2}B_1X'y$.

As $\pi(\beta|\sigma^2, y)$ is proportional to the kernel of a normal distribution, $\beta|\sigma^2, y \sim N(\bar{\beta}, B_1)$

The conditional posterior for σ^2 is,

$$\pi(\sigma^2|\beta, y) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_1}{2}+1} \exp\left[-\frac{\delta_1}{2\sigma^2}\right]$$

where $\alpha_1 = \alpha_0 + n$ and $\delta_1 = \delta_0 + (y - X\beta)'(y - X\beta)$

As $\pi(\sigma^2|\beta, y)$ is proportional to the kernel of $\text{IG}(\alpha_1/2, \delta_1/2)$, $\sigma^2|\beta, y \sim \text{IG}(\alpha_1/2, \delta_1/2)$.

Below is the R code for Gibbs Sampling for the Bayesian Linear Regression:

```r
# Load and prepare data
Mroz_Data <- read_excel("Mroz Data.xlsx")
data <- Mroz_Data[c("WHRS", "KL6", "K618", "WA", "HW")]

# Define response and predictors
y <- as.matrix(data["WHRS"])
X <- cbind(1, as.matrix(data[c("KL6", "K618", "WA", "HW")]))

# Dimensions and priors
G <- 20000  # total samples after burn in
B <- 5000   # burn-in
n <- nrow(X)# number of observations in the data
k <- ncol(X)# number of covariates
```

```r
# Prior parameters
alpha0 <- 100000
delta0 <- 10
B0 <- 1000 * diag(k)
invB0 <- solve(B0)  # constant, precompute

# Precomputations
alpha1 <- alpha0 + n
Xt <- t(X)
XtX <- Xt %*% X
XtY <- Xt %*% y

# Storage for MCMC draws
beta <- matrix(0, nrow = B + G, ncol = k)
sigma2 <- rep(1, B + G)

# Gibbs sampling
set.seed(13) # for reproducibility
for (g in 2:(B + G)) {
  sigma2_prev <- sigma2[g - 1]

  # Posterior of beta
  B1 <- solve((1 / sigma2_prev) * XtX + invB0)
  beta_bar <- B1 %*% ((1 / sigma2_prev) * XtY)

  # Sample beta
  beta[g, ] <- mvrnorm(1, mu = as.vector(beta_bar), Sigma = as.matrix(B1))

  # Sample sigma^2
  ## update delta1 = delta0 + (y-X beta)'(y- X beta)
  delta1 <- delta0 + sum((y - X %*% beta[g, ])^2)

  # if X~Gamma(shape,rate) then 1/X ~ IG(shape, scale = rate)
  # so below code samples from IG(alpha1/2,delta1/2)
  sigma2[g] <- 1/rgamma(1, shape = alpha1/2, rate = as.numeric(delta1/2))
}
```

	Posterior Mean	Posterior Std. Dev.
Intercept	1552.09	16.3821
KL6	-414.22	5.35912
K618	-67.9224	2.07818
WA	-12.3581	0.339819
HW	-14.1786	0.615345
σ^2	5166.97	23.0854

Table 2: Posterior Mean and Standard Deviation of parameters of Bayesian Linear Model

Why linear regression is not appropriate

The linear regression model is not suitable for work hours because work hours data are censored (many people work zero hours), which violates the assumptions of linear regression. Linear regression assumes continuous,

unbounded data, while work hours are bounded (cannot be negative) and there are many women (325) with zero work hours in the Mroz data.

Part (b)

The Tobit model with censoring from below is

$$z_i = x_i'\beta + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2)$$
$$y_i = \begin{cases} z_i, & \text{if } z_i > 0 \\ 0, & \text{otherwise} \end{cases}$$

The corresponding augmented joint posterior distribution is,

$$\begin{aligned}\pi(\beta, \sigma^2, z \,|\, y) &\propto \pi(\beta)\, \pi(\sigma^2)\, f(z \,|\, \beta, \sigma^2)\, f(y \,|\, \beta, \sigma^2, z) \\ &\propto \exp\left(-\frac{1}{2000}\beta'\beta\right)\left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_0}{2}+1} \exp\left(-\frac{\delta_0}{2\sigma^2}\right)\left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}} \\ &\quad \times \exp\left(-\frac{1}{2\sigma^2}(z - X\beta)'(z - X\beta)\right) \times \prod_{i=1}^{n} \left[I(y_i = 0)\, I(z_i \leq 0) + I(y_i = z_i)\, I(z_i > 0)\right]\end{aligned}$$

The conditional posteriors can be derived from the above joint posterior by collecting terms involving one parameters at a time while holding the remaining fixed.

Conditional posterior distribution of β

The conditional posterior distribution of β is

$$\begin{aligned}\pi(\beta|\sigma^2, z) &\propto \exp\left(-\frac{1}{2000}\beta'\beta\right) \exp\left(-\frac{1}{2\sigma^2}(z - X\beta)'(z - X\beta)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\beta'\beta}{1000} + \sigma^{-2}(z - X\beta)'(z - X\beta)\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\beta'\left(\sigma^{-2}X'X + \frac{1}{1000}I_k\right)\beta + 2\sigma^{-2}\beta'X'z\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\beta' B_1^{-1}\beta + 2B_1^{-1}\bar{\beta}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\beta' B_1^{-1}\beta + 2B_1^{-1}\bar{\beta} - \bar{\beta}' B_1^{-1}\bar{\beta}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}(\beta - \bar{\beta})' B_1^{-1}(\beta - \bar{\beta})\right)\end{aligned}$$

where $B_1 = \left(\sigma^{-2}X'X + \frac{1}{1000}I_k\right)^{-1}$ and $\bar{\beta} = B_1\sigma^{-2}\beta'X'z$

Therefore, $\pi(\beta|\sigma^2, z)$ is proportional to the kernel of a normal distribution.

Conditional posterior distribution of σ^2

The conditional posterior distribution of σ^2 is,

$$\begin{aligned}\pi(\sigma^2 \mid \beta, y) &\propto \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_0}{2}+1} \exp\left(-\frac{\delta_0}{2\sigma^2}\right)\left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}(z-X\beta)'(z-X\beta)\right)\\ &\propto \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_0+n}{2}+1} \exp\left(-\frac{\delta_0+(z-X\beta)'(z-X\beta)}{2\sigma^2}\right)\\ &\propto \left(\frac{1}{\sigma^2}\right)^{\frac{\alpha_1}{2}+1} \exp\left(-\frac{\delta_1}{2\sigma^2}\right)\end{aligned}$$

where $\alpha_1 = \alpha_0 + n$ and $\delta_1 = \delta_0 + (z - X\beta)'(z - X\beta)$.

Therefore, $\pi(\sigma^2|\beta, z)$ is proportional to the kernel of a $\text{IG}(\alpha_1, \delta_1)$ and thus $\sigma^2|\beta, z \sim \text{IG}(\alpha_1, \delta_1)$.

Conditional posterior distribution of z

The conditional posterior distribution of z_i is

$$f(z_i \mid \beta, \sigma^2, y_i) \propto \exp\left[-\frac{1}{2\sigma^2}(z_i - x_i'\beta)^2\right]\{I(y_i = 0)\cdot I(z_i \le 0) + I(y_i > 0)\cdot I(z_i > 0)\}$$

which implies,

$$\begin{aligned}f(z_i \mid \beta, \sigma^2, y_i = 0) &\propto \exp\left[-\frac{1}{2\sigma^2}(z_i - x_i'\beta)^2\right]\cdot I(z_i \le 0),\\ f(z_i \mid \beta, \sigma^2, y_i = z_i) &= 1\end{aligned}$$

Part (c)

Fitting a Tobit model

Code for fitting the Tobit model:

```
# code for the Tobit model
set.seed(13)          # For reproducibility

# Parameters
G <- 20000        # Total samples after burn in
B <- 5000         # Burn-in
n <- length(y)    # Number of observations
k <- ncol(X)      # Number of covariates

# Prior Parameters
alpha0 <- 100000
delta0 <- 10
B0 <- 1000 * diag(k)
invB0 <- solve(B0)

# Pre Computations
alpha1 <- alpha0 + n
XtX <- t(X) %*% X
```

```r
Xt <- t(X)
idx <- which(y == 0)

# Storage for MCMC draws
beta <- matrix(1, nrow = B + G, ncol = k)
sigma2 <- rep(1, B + G)
z <- matrix(0, nrow = B + G, ncol = n)

# Initialize z: if y > 0, then z = y (fixed across all iterations)
z[, y > 0] <- matrix(rep(y[y > 0], B + G), nrow = B + G, byrow = TRUE)

# Gibbs sampling
for (g in 2:(B + G)) {
  sigma2_prev <- sigma2[g - 1]
  z_prev <- z[g - 1, ]

  # Posterior parameters for beta
  B1 <- solve((1 / sigma2_prev) * XtX + invB0)
  beta_bar <- B1 %*% ((1 / sigma2_prev) * Xt %*% z_prev)

  # Sample beta
  beta[g, ] <- mvrnorm(1, mu = as.vector(beta_bar), Sigma = B1)

  # Sample sigma^2
  ## update delta1 = delta0 + (y-X beta)'(y- X beta)
  delta1 <- delta0 + sum((y - X %*% beta[g, ])^2)

  # if X~Gamma(shape,rate) then 1/X ~ IG(shape, scale = rate)
  # so below code samples from IG(alpha1/2,delta1/2)
  sigma2[g] <- 1/rgamma(1, shape = alpha1/2, rate = as.numeric(delta1/2))

  # Sample z only for censored obs (i.e., y == 0)
  # idx <- which(y == 0)
  mu_z <- X[idx, ] %*% beta[g, ]
  z[g, idx] <- rtruncnorm(length(idx), a = -Inf, b = 0, mean = mu_z,
                          sd = sqrt(sigma2[g]))
}
```

	Mean	Std. Dev
Intercept	1580.79	16.6827
KL6	-446.111	5.95444
K618	-69.6477	2.11968
WA	-12.8684	0.346145
HW	-14.6963	0.623072
σ^2	5163.29	23.018

Table 3: Posterior Mean and Standard Deviation of parameters of the Tobit model

From the above table we observe the following:

- `KL6` has a negative coefficient, indicating that presence of child under 6 reduces the work hours of a women on average as per our Tobit model. In particular each child under 6 reduces the work hours by

approximately 446 hours on average.

- `K618` also has a negative coefficient but smaller than `KL6` meaning it also decreases the work hours but to a lesser extent than `KL6`. Each child between 6 and 18 decreases the work hours by about 69.65 hours on average as per our Tobit model.
- `WA` also has a negative coefficient. So for each year a woman ages her work hours decrease about 12.87 hours on average as per our model.
- `HW` has a negative coefficient (about -14.70), indicating that for each dollar increase in the husband's hourly average earnings, the woman's work hours decreases by about 14.7 hours.
- σ^2: The estimated variance of the latent error term is 5163.29, corresponding to a standard deviation of approximately 71.87. This suggests substantial variability in the underlying outcome that is not captured by the model covariates. This implies that even though the model may identify directionally important variables, its predictive power might be limited due to the noise in the latent outcome.

Part (d)

Below is the code for calculating the Credible Intervals for the estimates from the Tobit model:

```
U = rep(0,6)
L = rep(0,6)
for(i in 1:5){
  U[i] = quantile(beta[(B+1):(B+G),i],0.975, digits = 6)
  L[i] = quantile(beta[(B+1):(B+G),i],0.025, digits = 6)
}
U[6] = quantile(sigma2[(B+1):(B+G)],0.975, digits = 6)
L[6] = quantile(sigma2[(B+1):(B+G)],0.025, digits = 6)
```

	Lower	Upper
Intercept	1547.56	1613.1
KL6	-457.937	-434.498
K618	-73.7896	-65.4698
WA	-13.5449	-12.1842
HW	-15.9292	-13.4805
σ^2	5118.4	5208.54

Table 4: 95 % Credible Intervals for the Parameters of the Tobit model

Confidence Interval

- A 95% confidence interval means that if we repeat the experiment infinitely many times, 95% of those intervals would contain the true parameter.
- **Interpretation:** We cannot say that the true parameter lies within a given interval with 95% probability.
- **Example:** If the interval is $[1.2, 2.5]$, it does not imply a 95% probability that the true value lies in that range. It implies the method used would produce such intervals that capture the true value 95% of the time over repeated sampling.

Credible Interval (Bayesian)

- A 95% credible interval is the interval within which the parameter lies with 95% **posterior probability**, given the observed data and prior beliefs.
- **Interpretation:** There is a 95% probability that the parameter lies within the computed interval.
- **Example:** If the posterior credible interval is $[1.2, 2.5]$, we can say there is a 95% probability that the true parameter lies within that interval.

Part (e)

Code for computing Inefficiency factor of the parameters using batch-means method:

```
# Set up for batch means method
breaks <- seq(0, G, length.out = 201)  # 200 batches
N <- length(breaks) - 1 # Number of batches
ineff_factor <- numeric(6) # storage variable
colnames(beta) <- c("Intercept", colnames(X)[2:5])

# Helper function to compute inefficiency factor
compute_IF <- function(samples) {
  z_mean <- mean(samples) # sample mean

  # batch means
  batch_means <- sapply(1:N, function(j) {
    mean(samples[(breaks[j] + 1):breaks[j + 1]])
  })

  # variance of batch means from sample mean
  var_batches <- sum((batch_means - z_mean)^2) / (N - 1)
  nse <- sqrt(var_batches / N) # numerical standard error
  sample_var <- var(samples) # sample variance

  # Return IF
  return(nse /sqrt(sample_var/ (length(samples))))
}

# Compute for each beta column (1:5)
ineff_factor[1:5] <- sapply(1:5, function(i) compute_IF(beta[(B+1):(B+G), i]))

# Compute for sigma2
ineff_factor[6] <- compute_IF(sigma2[(B+1):(B+G)])
```

Parameter	Inefficiency factor
Intercept	1.05439
KL6	1.21882
K618	1.02647
WA	1.02826
HW	1.00499
σ^2	1.00993

Table 5: Inefficiency factors of the parameters

Cost of using MCMC for each parameter to get an iid draw

The Inefficiency Factor quantifies how auto-correlated our MCMC samples are. It tells us how many correlated draws are needed to match the information content of one i.i.d. draw. The cost for each i.i.d. draw is essentially the value of IF itself.

Since all IFs are close to 1, the cost of obtaining an effective sample is low, and the chains mix very well.

For example, to get 1 effective sample for KL6, we need about 1.22 actual MCMC draws — about a 22% overhead compared to ideal independence.

Question 4

Code for the Binary Probit Model using classical methods:

```
hmda <- read_excel("hmda.xlsx") # load hmda data
# Binary: deny, selfemp, insurance,condomin, afam, single, and hschool
# Categorial: chist, mhist, phist,
# Continous: pirat, hirat, lvrat, unemp

# set up for the model
binary <- c("deny", "selfemp", "insurance","condomin",
            "afam", "single", "hschool","phist")
categorial <- c("chist","mhist")

for(i in binary){
  hmda[,i] <- ifelse(hmda[,i] == "yes", 1, 0)
}

for(i in categorial){
  hmda[[i]] <- factor(hmda[[i]])
}

Formula <- "deny ~ pirat"

for(i in names(hmda)){
  if(i != "deny" & i != "pirat"){
    Formula = paste(Formula,"+" ,i)
  }
}

# Create the probit model
classic_probit = glm(as.formula(Formula),family = binomial(link = "probit"),
                     data = hmda)
```

Code for Binary Probit model using Bayesian methods:

```
# set up for Bayesian Probit Model
X = matrix(nrow = length(hmda$deny), ncol = length(classic_probit$coefficients))
colnames(X) = names(classic_probit$coefficients)
X = as.data.frame(X)

X$pirat = hmda$pirat
X$hirat = hmda$hirat
X$lvrat = hmda$lvrat
X$unemp = hmda$unemp
X$`(Intercept)` = 1

for(i in binary){
  if(i != 'deny'){
    X[[i]] = hmda[[i]]
  }
}
```

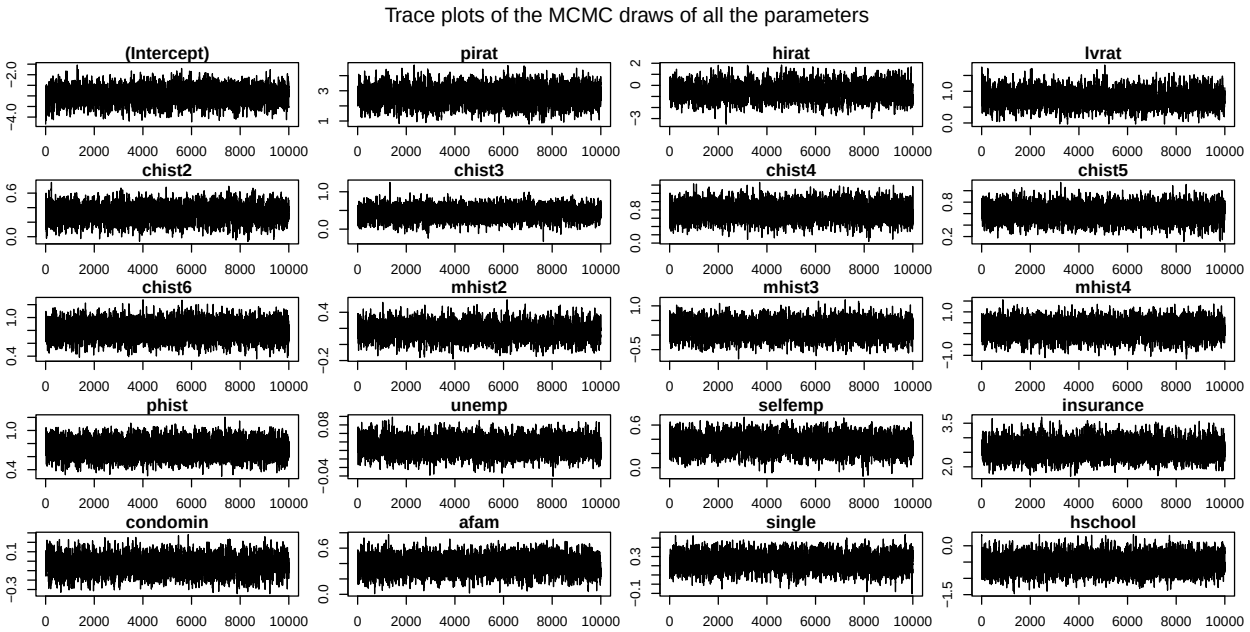
```r
for(i in 2:6){
  col = paste0("chist",i)
  X[[col]] = as.integer(hmda$chist == i)
}

for(i in 2:4){
  col = paste0("mhist",i)
  X[[col]] = as.integer(hmda$mhist == i)
}

y = as.matrix(hmda$deny)

# Parameters
k = length(X[1,])
B0 = 100*diag(k)
n = length(y)
invB0 = solve(B0)
B = 2500
G = 10000
X = as.matrix(X)

# Pre compute
XtX = t(X)%*%X
Xt = t(X)
B1 <- solve(XtX + invB0)
id1 <- which(y == 1)
id0 <- which(y == 0)

# Storage for MCMC draws
beta = matrix(1,nrow = B+G, ncol = k)
z = matrix(1, nrow = B+G, ncol = n)

set.seed(13)# for reproducibility

# Gibbs Sampling
for(g in 2:(B+G)){
  beta_bar = B1%*%Xt%*%z[g-1,]
  # Sample beta
  beta[g, ] <- mvrnorm(1, mu = as.vector(beta_bar), Sigma = B1)

  # Sample z accordingly from one of the two truncated normal distributions
  # id1 <- which(y == 1)
  # id0 <- which(y == 0)

  mu_z0 <- X[id0, ] %*% beta[g, ]
  mu_z1 <- X[id1, ] %*% beta[g, ]
  z[g, id0] <- rtruncnorm(length(id0), a = -Inf, b = 0, mean = mu_z0, sd = 1)
  z[g, id1] <- rtruncnorm(length(id1), a = 0, b = Inf, mean = mu_z1, sd = 1)
}
```

Parameter	ML Estimate	Std. Error	Posterior Mean	Posterior Std. Dev
(Intercept)	-2.922	0.3567	-2.965	0.3603
pirat	2.6526	0.56205	2.6748	0.55816
hirat	-0.54558	0.6737	-0.52394	0.7036
lvrat	0.78428	0.25653	0.7979	0.25246
chist2	0.32221	0.10667	0.32166	0.10988
chist3	0.43214	0.16422	0.43294	0.15896
chist4	0.76935	0.1841	0.77041	0.18441
chist5	0.60797	0.13071	0.60991	0.13247
chist6	0.79973	0.1254	0.80141	0.12495
mhist2	0.16643	0.098851	0.16553	0.098171
mhist3	0.2072	0.26351	0.19829	0.26593
mhist4	0.23112	0.35818	0.21411	0.35894
phist	0.72416	0.11983	0.72699	0.11888
unemp	0.030694	0.018228	0.03076	0.018485
selfemp	0.3371	0.1141	0.33235	0.11453
insurance	2.5741	0.28525	2.6184	0.28775
condomin	-0.034721	0.090249	-0.03895	0.089831
afam	0.38022	0.099215	0.38389	0.1012
single	0.23702	0.08273	0.23871	0.083399
hschool	-0.55792	0.24079	-0.54629	0.24581

Table 6: Estimates of Classical Probit and Bayesian Probit

Code for plotting the trace plots of MCMC draws of all the 20 parameters:

```
# 5 rows and 4 columns
par(mfrow = c(5,4),mar = c(2, 2, 1.25, 1),oma = c(0, 0, 2.25, 0))
for(i in 1:k){
  plot(beta[(B+1):(B+G),i], type = "l",
     main = colnames(X)[i],
     xlab = "", ylab = "")
}
mtext("Trace plots of the MCMC draws of all the parameters", outer = TRUE,
      cex = 1, line = 1)
```



Code for calculating the average covariate effect of loan denial for a black family (i.e.`afam`):

```
# Create counterfactual datasets
X_black  <- X; X_black[, "afam"] <- 1 # black
X_nblack <- X; X_nblack[, "afam"] <- 0 # not black

# Extract posterior samples after the burn-in
beta_post <- beta[(B+1):(B+G), ]   # G x k

# Compute predicted probabilities:
# Apply pnorm elementwise and take difference
delta_prob <- pnorm(beta_post %*% t(X_black)) - pnorm(beta_post %*% t(X_nblack))

# Average over both dimensions to get Average Covariate Effect
ACE <- mean(delta_prob)
paste("Average Covariate Effect for African American:", round(ACE,4))
```

```
## [1] "Average Covariate Effect for African American: 0.0642"
```

Racial Bias against blacks in granting loan

As the average covariate effect of the variable `afam` is 0.0642, i.e. keeping all other factors constant, being African-American increases the probability of loan denial by about 6.42% as per our Bayesian Probit Model. So we do have some evidence for racial bias against blacks in granting loans.