

HW4 - Econometrics

Himanshu, MDS202327

2025-04-24

Question 1

(a) Find the posterior distribution of θ .

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

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

The likelihood is given by,

$$f(y | \theta) \propto \exp \left[-\frac{1}{2} (y - \theta)' \Sigma^{-1} (y - \theta) \right]$$

The prior distribution of $\theta = (\theta_1, \theta_2) \sim U(0, 1)$ is given by, $\pi(\theta) = 1$

The posterior distribution is given by,

$$\begin{aligned} \pi(\theta | y) &\propto f(y | \theta) \cdot \pi(\theta) \\ &\propto \exp \left[-\frac{1}{2} (y - \theta)' \Sigma^{-1} (y - \theta) \right] \cdot 1 \\ &\propto \exp \left[-\frac{1}{2} (\theta - y)' \Sigma^{-1} (\theta - y) \right] \\ &\propto N \left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right) \end{aligned}$$

(b) Show that the posterior conditional distributions are

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

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

The likelihood is given by,

$$f(y | \theta_1, \theta_2) \propto \exp \left[-\frac{1}{2} (y - \theta)' \Sigma^{-1} (y - \theta) \right] \propto \exp \left[-\frac{1}{2(1 - \rho^2)} \left\{ (y_1 - \theta_1)^2 - 2\rho(y_1 - \theta_1)(y_2 - \theta_2) + (y_2 - \theta_2)^2 \right\} \right]$$

The prior distribution of $\theta_1 \sim U(0, 1)$ is given by, $\pi(\theta_1) = 1$

The conditional posterior distribution is given by,

$$\begin{aligned}
\pi(\theta_1 \mid \theta_2, y) &\propto f(y \mid \theta_1, \theta_2) \cdot \pi(\theta_1) \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ (y_1 - \theta_1)^2 - 2\rho(y_1 - \theta_1)(y_2 - \theta_2) + (y_2 - \theta_2)^2 \right\} \right] \cdot 1 \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ (\theta_1 - y_1)^2 - 2\rho(\theta_1 - y_1)(\theta_2 - y_2) + (\theta_2 - y_2)^2 \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ \theta_1^2 - 2\theta_1 [y_1 + \rho(\theta_2 - y_2)] + [y_1^2 + y_2^2 + \theta_2^2 - 2\theta_2 y_2 + 2\rho\theta_2 y_1 - 2\rho y_1 y_2] \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ \theta_1^2 - 2\theta_1 [y_1 + \rho(\theta_2 - y_2)] + [y_1 + \rho(\theta_2 - y_2)]^2 - [y_1 + \rho(\theta_2 - y_2)]^2 \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ (\theta_1^2 - [y_1 + \rho(\theta_2 - y_2)])^2 - [y_1 + \rho(\theta_2 - y_2)]^2 \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} (\theta_1 - [y_1 + \rho(\theta_2 - y_2)])^2 \right] \\
&\propto N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2)
\end{aligned}$$

Following the similar argument for θ_2 we can get, $\pi(\theta_2 \mid \theta_1, y) \propto N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)$

$$\begin{aligned}
\pi(\theta_2 \mid \theta_1, y) &\propto f(y \mid \theta_1, \theta_2) \cdot \pi(\theta_2) \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ (y_2 - \theta_2)^2 - 2\rho(y_2 - \theta_2)(y_1 - \theta_1) + (y_1 - \theta_1)^2 \right\} \right] \cdot 1 \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ (\theta_2 - y_2)^2 - 2\rho(\theta_2 - y_2)(\theta_1 - y_1) + (\theta_1 - y_1)^2 \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ \theta_2^2 - 2\theta_2 [y_2 + \rho(\theta_1 - y_1)] + [y_2^2 + y_1^2 + \theta_1^2 - 2\theta_1 y_1 + 2\rho\theta_1 y_2 - 2\rho y_2 y_1] \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ \theta_2^2 - 2\theta_2 [y_2 + \rho(\theta_1 - y_1)] + [y_2 + \rho(\theta_1 - y_1)]^2 - [y_2 + \rho(\theta_1 - y_1)]^2 \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} \left\{ (\theta_2^2 - [y_2 + \rho(\theta_1 - y_1)])^2 - [y_2 + \rho(\theta_1 - y_1)]^2 \right\} \right] \\
&\propto \exp \left[-\frac{1}{2(1-\rho^2)} (\theta_2 - [y_2 + \rho(\theta_1 - y_1)])^2 \right] \\
&\propto N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)
\end{aligned}$$

(c) Assume $y = (1, 0.5)$ and $\rho = 0.8$. Draw 10,000 observations from the conditional posterior distributions using Gibbs sampling. Plot the distribution of θ_1 and θ_2 after a burn-in of 1000.

```

set.seed(123)
n_iter = 11000
burn_in = 1000
rho = 0.8
y1 = 1
y2 = 0.5
sigma2 = 1 - rho^2
theta1 = numeric(n_iter)
theta2 = numeric(n_iter)
eta1 = numeric(n_iter)
eta2 = numeric(n_iter)

```

```

theta1[1] = y1
theta2[1] = y2

for (i in 2:n_iter) {
  theta1[i] = rnorm(1, mean = y1 + rho * (theta2[i-1] - y2), sd = sqrt(sigma2))
  eta1[i] = (theta1[i] - (y1 + rho * (theta2[i-1] - y2)))/sqrt(sigma2)
  theta2[i] = rnorm(1, mean = y2 + rho * (theta1[i] - y1), sd = sqrt(sigma2))
  eta2[i] = (theta2[i] - (y2 + rho * (theta1[i] - y1)))/sqrt(sigma2)
}

theta1_post = theta1[(burn_in+1):n_iter]
theta2_post = theta2[(burn_in+1):n_iter]

eta1_post = eta1[(burn_in+1):n_iter]
eta2_post = eta2[(burn_in+1):n_iter]

breaks = seq(
  min(c(theta1_post, theta2_post)),
  max(c(theta1_post, theta2_post)),
  length.out = 31
)

par(mfrow=c(1,3))
hist(theta1_post, breaks=breaks, main=expression(theta[1]), xlab="", ylim = c(0, 1100))
box()

hist(theta1_post, breaks=breaks, col=rgb(1,0,0,0.5), border="white", xlab="",
      xlim=range(breaks), main=expression(theta), ylim = c(0, 1100))
hist(theta2_post, breaks=breaks, col=rgb(0,0,1,0.5), border="white", add=TRUE)
legend("topright", legend=c(expression(theta[1]), expression(theta[2])),
      fill=c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)), border="white")
box()

hist(theta2_post, breaks=breaks, main=expression(theta[2]), xlab="")
box()

```

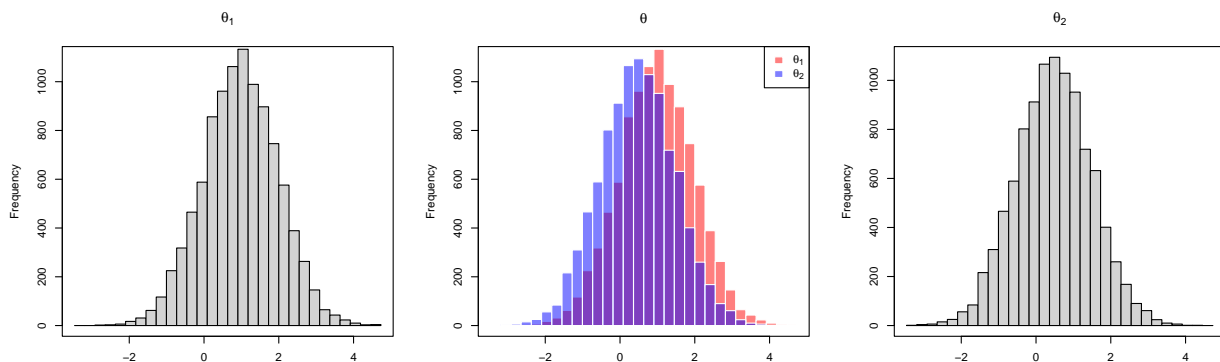


Figure 1: Distribution of θ_1 and θ_2

(d) While doing Gibbs sampling, simultaneously construct two parameters, say η_1 and η_2 which are standardized versions of θ_1 and θ_2 , respectively. Plot the distribution of η_1/η_2 after the burn-in. What do you observe? Can you recognize its distribution?

```
ratio = eta1_post / eta2_post
x = seq(-10, 10, length = 10000)
y = dcauchy(x, 0, 1)

hist(ratio, breaks=20000, main=expression(eta[1]/eta[2]), xlab="", xlim=c(-10,10),
     ylim=c(0, 0.4), cex.lab=0.8, cex.main=0.8, cex.axis=0.8, freq=FALSE)
lines(x, y, col = "blue3", lwd = 1.8)
legend("topright", legend=c("Cauchy (0,1)"), col = c("blue3"), lty=c(1), border="white",
      cex = 0.8)
box()
```

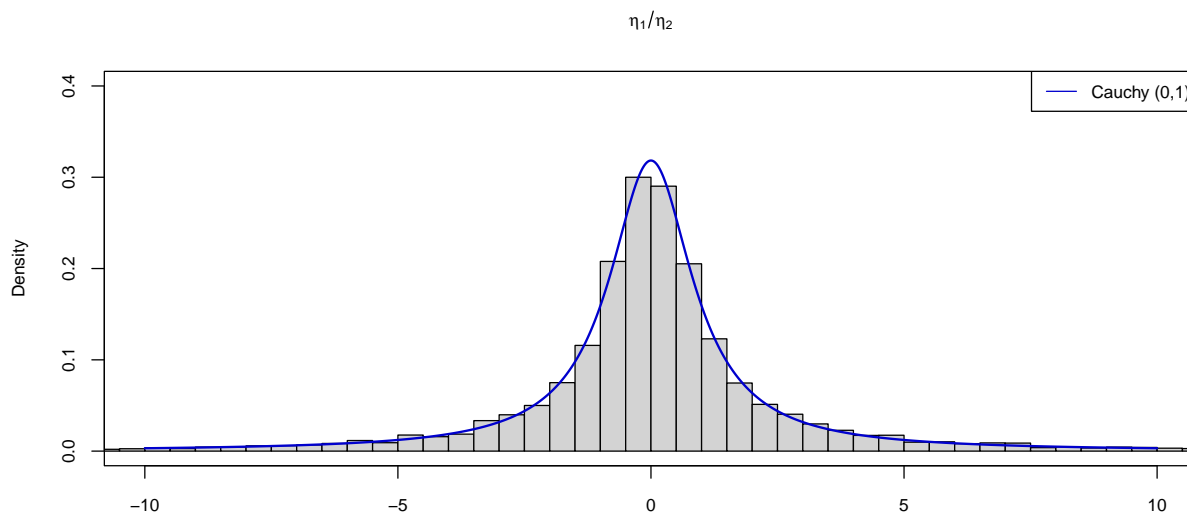


Figure 2: Distribution of η_1/η_2

Upon standardization η_1 and η_2 follow $N(0, 1)$ and they are independent of each other, the ratio η_1/η_2 shall follow Cauchy(0,1). In plot above, the histogram is overlaid with Cauchy(0,1) distribution.

Question 2

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. Report the acceptance rate, mean, variance of the MH draws. Present a trace plot and a histogram of the draws. For a $Beta(3, 4)$ distribution, what are the theoretical mean and variance? Are the mean and variance from MH draws different from the theoretical counterpart?

```
set.seed(123)
n_iter = 10000
beta_draws = numeric(n_iter)
beta_draws[1] = runif(1)
accepts = 0

for (i in 2:n_iter) {
  proposal = runif(1)
  num = dbeta(proposal, 3, 4)
  den = dbeta(beta_draws[i-1], 3, 4)
  alpha = min(1, num/den)
  if (runif(1) < alpha) {
    beta_draws[i] = proposal
    accepts = accepts + 1
  } else {
    beta_draws[i] = beta_draws[i-1]
  }
}

# Acceptance Rate, Mean and Variance of draws
accept_rate = round(accepts / (n_iter - 1), 4)
mean_draws = round(mean(beta_draws), 4)
var_draws = round(var(beta_draws), 4)
cat("Acceptance Rate:", accept_rate, " Draw Mean:", mean_draws, "Draw Var:", var_draws)
```

```
## Acceptance Rate: 0.5856 Draw Mean: 0.4281 Draw Var: 0.031
```

```
# Theoretical mean and variance
theo_mean = 3 / (3 + 4)
theo_var = (3 * 4) / ((3 + 4)^2 * (3 + 4 + 1))
cat("Theoretical Mean:", round(theo_mean, 4), "Theoretical Var:", round(theo_var, 4))
```

```
## Theoretical Mean: 0.4286 Theoretical Var: 0.0306
```

The mean and variance from MH draws are very close to the theoretical counterpart.

|Simulated Mean – Theoretical Mean| = 0.0005

|Simulated Variance – Theoretical Variance| = 0.0004

```

x = seq(0, 1, length = 10000)
y = dbeta(x, 3, 4)

par(mfrow=c(1,2))
plot(1:n_iter, beta_draws, type="l", main="", ylab="", xlab="")
hist(beta_draws, breaks=30, main="", xlab="", freq=FALSE)
lines(x, y, col = "blue3", lwd = 1.8)
legend("topright", legend=c("Beta (3,4)"), col = c("blue3"), lty=c(1), border="white",
      cex = 0.9)
box()

```

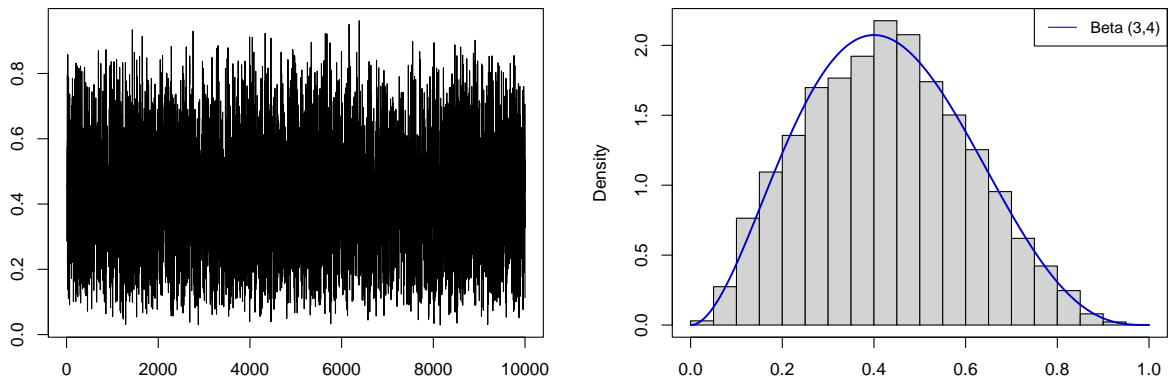


Figure 3: Trace plot and histogram of simulated draws from $\beta(2, 3)$

Question 3

(a) Fit a Bayesian linear regression model and report the posterior mean and posterior standard deviation of the parameters in a table (the name of the variables should be present in the first column). Are there reasons to believe that a linear regression framework will not be appropriate for this data?

```
library(MASS)

# function to implement Gibbs Sampling for Normal Linear Regression
bayesian_lr = function(y, X, n_iter = 25000, burn = 5000) {
  n = length(y)
  k = ncol(X)
  beta_samples = matrix(0, nrow = n_iter, ncol = k)
  sigma2_samples = numeric(n_iter)

  # initial values (using OLS estimates)
  ols_fit = lm(y ~ X-1)
  beta = coef(ols_fit)
  sigma2 = sigma(ols_fit)^2

  # prior parameters
  beta0 = rep(0, k)
  B0inv = diag(1/1000, k)
  alpha0 = 100000
  delta0 = 10

  XtX = t(X) %*% X
  Xty = t(X) %*% y

  beta_samples[1,] = beta
  sigma2_samples[1] = sigma2

  for (i in 2:n_iter) {
    # sample beta from its conditional posterior
    B1 = solve(XtX/sigma2 + B0inv)
    beta_bar = B1 %*% (B0inv %*% beta0 + Xty/sigma2)
    beta = mvrnorm(1, beta_bar, B1)

    # sample sigma2 from its conditional posterior
    alpha1 = (alpha0 + n)
    delta1 = (delta0 + sum((y - X %*% beta)^2))
    sigma2 = 1/rgamma(1, alpha1/2, delta1/2)

    beta_samples[i, ] = beta
    sigma2_samples[i] = sigma2
  }
  beta_samples = beta_samples[(burn+1):n_iter, ]
  sigma2_samples = sigma2_samples[(burn+1):n_iter]
  samples = as.matrix(cbind(beta_samples, sigma2_samples))
  colnames(samples) = c("Intercept", colnames(X)[2:5], "$\\sigma^2$")
  return(samples)
}
```

```
post_summary = function(x){
  df= data.frame(posterior_mean = colMeans(x),posterior_sd = apply(x, 2, sd))
  return(df)
}
```

```
mroz = read.csv("mroz.csv", header = TRUE)
X = cbind(1, as.matrix(mroz[,1:4]))
y = as.matrix(mroz$WHRS)
results_lr = bayesian_lr(y, X, n_iter = 25000, burn = 5000)
knitr::kable(post_summary(results_lr), digits = 3,
  col.names = c("Posterior Mean", "Posterior Std. Dev"))
```

	Posterior Mean	Posterior Std. Dev
Intercept	1552.035	16.594
childl6	-414.212	5.383
childg6	-67.910	2.081
age	-12.358	0.344
huswage	-14.178	0.624
σ^2	5166.979	23.060

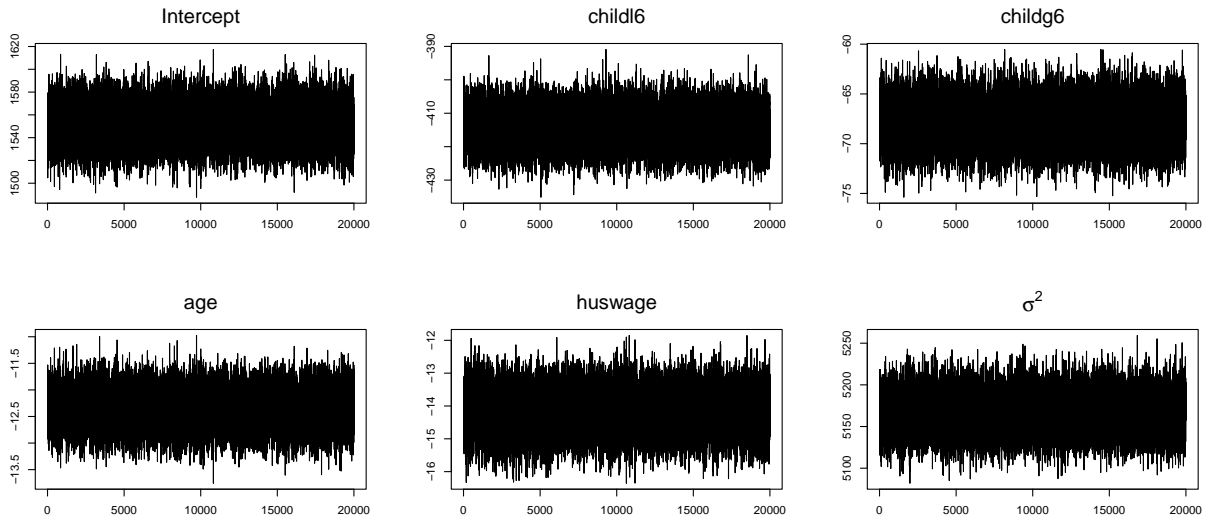


Figure 4: Trace plots of simulated draws for the covariates using normal linear regression model

Due to the following reasons to believe that a linear regression framework will not be appropriate for this data

- Approximately 43% of the surveyed women worked zero hours, while the remaining women worked an average of 1,303 hours per year. This large proportion of zeros cannot be adequately handled by standard linear regression.
- Women cannot work negative hours, so the data has a natural lower bound at zero, creating a discontinuity in the distribution that linear regression cannot properly model.

(b) Write down a Tobit model and the corresponding augmented posterior distribution. Derive the conditional posteriors for (β, σ^2, z) .

The Tobit model with censoring from below is given by,

$$z_i = x_i' \beta + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

$$y_i = \begin{cases} z_i, & \text{if } z_i > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The augmented joint posterior distribution is given by,

$$\begin{aligned} \pi(\beta, \sigma^2, z \mid y) &\propto \pi(\beta) \pi(\sigma^2) f(z \mid \beta, \sigma^2) f(y \mid z) \\ &\propto \exp\left[-\frac{1}{2}(\beta - \beta_0)' B_0^{-1}(\beta - \beta_0)\right] \left(\frac{1}{\sigma^2}\right)^{\left(\frac{n}{2}\right)} \cdot \exp\left[-\frac{\delta_1}{2\sigma^2}\right] \left(\frac{1}{\sigma^2}\right)^{\left(\frac{\alpha_1}{2}+1\right)} \\ &\quad \exp\left[-\frac{1}{2\sigma^2}(z - X\beta)'(z - X\beta)\right] \cdot \prod_{i=1}^n [I(y_i = 0)I(z_i \leq 0) + I(y_i = z_i)I(z_i > 0)] \end{aligned}$$

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

The conditional posterior distribution of β is given by,

$$\begin{aligned} \pi(\beta \mid \sigma^2, z) &\propto f(z \mid \beta, \sigma^2) \cdot \pi(\beta) \\ &\propto \exp\left[-\frac{1}{2\sigma^2}(z - X\beta)'(z - X\beta)\right] \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}} \cdot \exp\left[-\frac{1}{2}(\beta - \beta_0)' B_0^{-1}(\beta - \beta_0)\right] \\ &\propto \exp\left[-\frac{1}{2}\left\{\sigma^{-2}(z - X\beta)'(z - X\beta) + (\beta - \beta_0)' B_0^{-1}(\beta - \beta_0)\right\}\right] \end{aligned}$$

Expanding the terms inside the curly bracket,

$$\begin{aligned} (z - X\beta)'(z - X\beta) &= z'z - 2\beta'X'z + \beta'X'X\beta \\ (\beta - \beta_0)' B_0^{-1}(\beta - \beta_0) &= \beta' B_0^{-1}\beta - 2\beta' B_0^{-1}\beta_0 + \beta_0' B_0^{-1}\beta_0 \end{aligned}$$

Substituting back and grouping together,

$$\begin{aligned} \pi(\beta \mid \sigma^2, z) &\propto \exp\left[-\frac{1}{2}\left\{\beta'(\sigma^{-2}X'X + B_0^{-1})\beta - 2\beta'(\sigma^{-2}X'z + B_0^{-1}\beta_0) + \beta_0' B_0^{-1}\beta_0 + \sigma^{-2}z'z\right\}\right] \\ &\propto \exp\left[-\frac{1}{2}\left\{\beta'(\sigma^{-2}X'X + B_0^{-1})\beta - 2\beta'(\sigma^{-2}X'z + B_0^{-1}\beta_0)\right\}\right] \end{aligned}$$

Let $A = \sigma^{-2}X'X + B_0^{-1}$, $b = \sigma^{-2}X'z + B_0^{-1}\beta_0$ and $\bar{\beta} = A^{-1}b$

$$\begin{aligned} \pi(\beta \mid \sigma^2, z) &\propto \exp\left[-\frac{1}{2}\left\{\beta' A \beta - 2\beta' b\right\}\right] \\ &\propto \exp\left[-\frac{1}{2}\left\{\beta' A \beta - 2\beta' b + \bar{\beta}' A \bar{\beta} - \bar{\beta}' A \bar{\beta}\right\}\right] \\ &\propto \exp\left[-\frac{1}{2}\left\{(\beta' A \beta - 2\beta'(A\bar{\beta}) + \bar{\beta}' A \bar{\beta}) - \bar{\beta}' A \bar{\beta}\right\}\right] \\ &\propto \exp\left[-\frac{1}{2}\left\{(\beta - \bar{\beta})' A (\beta - \bar{\beta}) - \bar{\beta}' A \bar{\beta}\right\}\right] \\ &\propto \exp\left[-\frac{1}{2}\left\{(\beta - \bar{\beta})' A (\beta - \bar{\beta})\right\}\right] \\ &\propto N(\bar{\beta}, B_1) \quad \text{where } B_1 = A^{-1} = [\sigma^{-2}X'X + B_0^{-1}]^{-1} \text{ and } \bar{\beta} = B_1[\sigma^{-2}X'z + B_0^{-1}\beta_0] \end{aligned}$$

The conditional posterior distribution of σ^2 is given by,

$$\begin{aligned}
\pi(\beta \mid \sigma^2, z) &\propto f(z \mid \beta, \sigma^2) \cdot \pi(\sigma^2) \\
&\propto \exp \left[-\frac{1}{2\sigma^2} (z - X\beta)'(z - X\beta) \right] \left(\frac{1}{\sigma^2} \right)^{\frac{n}{2}} \cdot \exp \left[-\frac{\delta_0}{2\sigma^2} \right] \left(\frac{1}{\sigma^2} \right)^{\left(\frac{\alpha_0}{2} + 1 \right)} \\
&\propto \exp \left[-\frac{1}{2\sigma^2} \left(\delta_0 + (z - X\beta)'(z - X\beta) \right) \right] \cdot \left(\frac{1}{\sigma^2} \right)^{\left(\frac{\alpha_0 + n}{2} + 1 \right)} \\
&\propto \exp \left[-\frac{\delta_1}{2\sigma^2} \right] \cdot \left(\frac{1}{\sigma^2} \right)^{\left(\frac{\alpha_1}{2} + 1 \right)} \\
&\propto IG \left(\frac{\alpha_1}{2}, \frac{\delta_1}{2} \right) \quad \text{where } \alpha_1 = \alpha_0 + n \text{ and } \delta_1 = \delta_0 + (z - X\beta)'(z - X\beta)
\end{aligned}$$

The conditional posterior distribution of z is given by,

$$f(z_i \mid \beta\sigma^2, y_i) \propto \exp \left[-\frac{1}{2\sigma^2} (z_i - X_i^{prime}\beta)^2 \right] \cdot \prod_{i=1}^n [I(y_i = 0)I(z_i \leq 0) + I(y_i = z_i)I(z_i > 0)]$$

which implies

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

(c) Fit a Tobit model and report the posterior mean and standard deviations in a table. Comment on the effect of each variable on the response variable.

```

library(truncnorm)

# function to implement Gibbs Sampling for Tobit Model
bayesian_tr = function(y, X, n_iter = 25000, burn = 5000) {
  n = length(y)
  k = ncol(X)
  beta_samples = matrix(0, nrow = n_iter, ncol = k)
  sigma2_samples = numeric(n_iter)

  # initial values
  non_zero = which(y > 0)
  if (length(non_zero) > k) {
    ols_fit = lm(y[non_zero] ~ X[non_zero,] - 1)
    beta = coef(ols_fit)
    sigma2 = sigma(ols_fit)^2
  } else {
    beta = rep(0, k)
    sigma2 = 1
  }

  # initialize latent variables
  z = y
  censored = which(y == 0)

```

```

if (length(censored) > 0) {
  # For censored observations, initialize z to valid values <= 0
  mu = X[censored,] %*% beta
  z[censored] = rtruncnorm(length(censored), mean = as.vector(mu),
                           sd = sqrt(sigma2), a = -Inf, b = 0)
}

# prior parameters
beta0 = rep(0, k)
B0inv = diag(1/1000, k)
alpha0 = 100000
delta0 = 10

XtX = t(X) %*% X

beta_samples[1,] = beta
sigma2_samples[1] = sigma2

for (i in 2:n_iter) {
  # sample beta from its conditional posterior
  Xtz = t(X) %*% z
  B1 = solve(XtX/sigma2 + B0inv)
  beta_bar = B1 %*% (B0inv %*% beta0 + Xtz/sigma2)
  beta = mvrnorm(1, beta_bar, B1)

  # sample sigma2 from its conditional posterior
  residuals = z - X %*% beta
  alpha1 = alpha0 + n
  delta1 = delta0 + sum(residuals^2)
  sigma2 = 1/rgamma(1, alpha1/2, delta1/2)

  # sample z for censored observations from truncated normal
  if (length(censored) > 0) {
    mu = X[censored,] %*% beta
    z[censored] = rtruncnorm(length(censored), mean = as.vector(mu),
                             sd = sqrt(sigma2), a = -Inf, b = 0)
  }

  beta_samples[i,] = beta
  sigma2_samples[i] = sigma2
}

beta_samples = beta_samples[(burn+1):n_iter, ]
sigma2_samples = sigma2_samples[(burn+1):n_iter]
samples = as.matrix(cbind(beta_samples, sigma2_samples))
colnames(samples) = c("Intercept", colnames(X)[-1], "$\\sigma^2$")
return(samples)
}

```

```

mroz = read.csv("mroz.csv", header = TRUE)
X = cbind(1, as.matrix(mroz[,1:4]))
y = as.matrix(mroz$WHR5)
results_tr = bayesian_tr(y, X, n_iter = 5000, burn = 1000)

knitr::kable(post_summary(results_tr), row.names = ,
              col.names = c("Posterior Mean", "Posterior Std. Dev"), digits = 3)

```

	Posterior Mean	Posterior Std. Dev
Intercept	1578.363	16.807
childl6	-445.716	5.922
childg6	-69.519	2.122
age	-12.823	0.347
huswage	-14.673	0.629
σ^2	5187.711	23.492

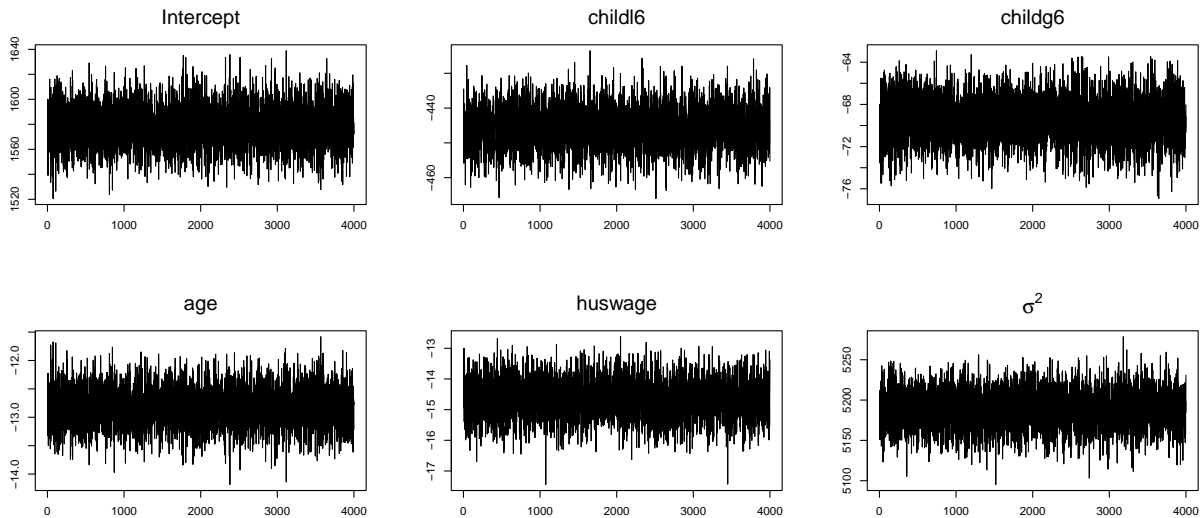


Figure 5: Trace plots of simulated draws for the covariates using tobit model

- The large positive **intercept** represents the baseline expected hours worked when all other predictors are zero.
- Each additional **child under six years old** is associated with a substantial reduction of approximately 446 hours worked per year.
- Each additional **child in the age group 6 to 18** is associated with a decrease of about 70 hours worked annually. This aligns with expectations that older children require less intensive care, allowing mothers more time for employment.
- Each additional year of a **woman's age** is associated with a decrease of approximately 13 hours worked annually. This negative relationship suggests that older women in the sample tend to work fewer hours.
- Each additional unit of **husband's wage** corresponding to about 15 fewer hours worked by the wife. This suggests an income effect where women in households with higher husband earnings may choose to work less.
- The **error variance** is substantial, indicating considerable unexplained variation in women's hours worked beyond what's captured by the model's predictors.

(d) Report the 95% probability interval or credible interval. Explain the difference between confidence interval and credible interval.

```
ci_df = data.frame(matrix(nrow = 6, ncol = 4))
rownames(ci_df) = c("Intercept", colnames(X)[-1], "$\\sigma^2$")
colnames(ci_df) = c("Linear Regression 2.5%", "Linear Regression 97.5%",
                    "Tobit Regression 2.5%", "Tobit Regression 97.5%")
for (i in 1:6) {
  x_lr = results_lr[, i]
  prob_int_lr = quantile(x_lr, probs=c(0.025, 0.975))
  ci_df[i, 1:2] = prob_int_lr

  x_tr = results_tr[, i]
  prob_int_tr = quantile(x_tr, probs=c(0.025, 0.975))
  ci_df[i, 3:4] = prob_int_tr
}
knitr::kable(ci_df, digits=3)
```

	Linear Regression 2.5%	Linear Regression 97.5%	Tobit Regression 2.5%	Tobit Regression 97.5%
Intercept	1519.539	1584.385	1545.280	1611.527
chldl6	-424.779	-403.737	-457.239	-434.125
chldg6	-71.968	-63.794	-73.750	-65.326
age	-13.026	-11.684	-13.489	-12.141
huswage	-15.402	-12.955	-15.911	-13.432
σ^2	5122.067	5212.308	5142.372	5234.128

Treatment of Parameters

- In confidence intervals, the parameter is considered fixed (not a random variable), while the interval is random
- In credible intervals, the parameter is treated as a random variable with a probability distribution

Interpretation

- A 95% confidence interval means that with repeated sampling, 95% of similarly constructed intervals would contain the true parameter value
- A 95% credible interval means there's a 95% probability that the parameter falls within the interval, given the observed data and prior information

(e) Compute the inefficiency factor for each parameter using the batch-means method (Greenberg, 2012). Comment on the cost of using MCMC for each parameter to get an iid draw.

```
library(bqror)
inf_lr = ineffactorOR2(x, t(results_lr[,1:5]), t(results_lr[,6]), 0.05, 400, FALSE)
inf_tr = ineffactorOR2(x, t(results_tr[,1:5]), t(results_tr[,6]), 0.05, 400, FALSE)
inf_df = cbind(inf_lr, inf_tr)
rownames(inf_df) = c("Intercept", colnames(X)[-1], "$\\sigma^2$")
colnames(inf_df) = c("Linear Regression", "Tobit Regression")
knitr::kable(inf_df, digits = 3)
```

	Linear Regression	Tobit Regression
Intercept	1.009	1.014
childl6	1.008	1.128
childg6	1.001	1.007
age	1.011	1.013
huswage	1.001	1.008
σ^2	1.009	1.011

Intercept: With factors of 1.009 (Linear) and 1.014 (Tobit), the intercept parameter shows excellent mixing in both models, requiring only about 1

childl6: This parameter shows the most notable difference between models. While the linear model has an excellent factor of 1.008, the Tobit model's factor of 1.128 indicates that approximately 13

childg6: With very low inefficiency factors (1.001 for Linear and 1.007 for Tobit), this parameter demonstrates the best mixing in the Linear model and excellent performance in the Tobit model.

age: Both models show similar performance for this parameter (1.011 for Linear and 1.013 for Tobit), indicating slightly more autocorrelation than the best-performing parameters.

huswage: The Linear model achieves near-perfect mixing (1.001) for this parameter, while the Tobit model still performs very well at 1.008.

σ^2 : Both models show good performance for the variance parameter (1.009 for Linear and 1.011 for Tobit), contradicting the common pattern where variance parameters often mix poorly compared to mean parameters.

Question 4

First, estimate the model using the classical method. Second, fit a binary probit model using Bayesian methods assuming the prior $\beta \sim N(0_k, 100 * I_k)$. Prepare a table where you report the maximum likelihood (ML) estimate and standard error of the regression coefficients. In the same table, report the posterior estimates (mean and standard deviation) based on 10,000 MCMC iterations after a burn-in of 2,500 iterations. Present a trace plot of the MCMC draws of the 20 parameters (including the intercept) in one figure.

Do you find any evidence of racial bias in granting loans to black family? What is average covariate effect of loan denial for a black family?

```
hmda = read.csv("hmda.csv", header = TRUE)

for(i in 5:13) {
  hmda[,i] = as.factor(hmda[,i])
}

hmda$deny = as.integer(hmda$deny == "yes")

# classical probit model
probit_model = glm(deny ~ ., data = hmda, family = binomial(link = "probit"))
ml_est = coef(probit_model)
ml_se = sqrt(diag(vcov(probit_model)))
df_pr = cbind(ml_est, ml_se)
```

```
# function to implement Gibbs Sampling for Binary Probit Model
bayesian_probit = function(y, X, n_iter = 10000, burn = 2500) {
  n = length(y)
  k = ncol(X)
  beta_samples = matrix(0, nrow = n_iter, ncol = k)
  idx0 = (y == 0)
  idx1 = (y == 1)

  # Initial values
  beta = rep(0, k)
  z = rep(0, n)
  z[y == 1] = 1
  z[y == 0] = -1

  # Prior parameters
  beta0 = rep(0, k)
  B0inv = diag(1/1000, k)

  XtX = t(X) %*% X
  B1 = solve(XtX + B0inv)

  # Store initial values
  beta_samples[1,] = beta

  for (i in 2:n_iter) {
    # Sample beta from its conditional posterior
    beta_bar = B1 %*% (B0inv %*% beta0 + t(X) %*% z)
```

```

beta = mvrnorm(1, beta_bar, B1)

# Sample latent z from its conditional posterior
Xbeta = X %*% beta

# Sample for y == 0 (truncated normal below 0)
if (any(idx0)) {
  z[idx0] = rtruncnorm(sum(idx0), mean = Xbeta[idx0], sd = 1, a = -Inf, b = 0)
}

# Sample for y == 1 (truncated normal above 0)
if (any(idx1)) {
  z[idx1] = rtruncnorm(sum(idx1), mean = Xbeta[idx1], sd = 1, a = 0, b = Inf)
}

beta_samples[i,] = beta
}

beta_samples = beta_samples[(burn+1):n_iter, ]
colnames(beta_samples) = c("Intercept", colnames(X)[2:20])
return(beta_samples)
}

```

```

hmda = read.csv("hmda.csv", header = TRUE)

X = matrix(nrow = length(hmda$deny), ncol = length(probit_model$coefficients))
colnames(X) = names(probit_model$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 7:13){
  X[,i+7] = as.integer(hmda[,i]=="yes")
}

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)
}

X = as.matrix(X)
y = as.integer(hmda$deny == "yes")

results_probit = bayesian_probit(y, X, n_iter = 12500, burn = 2500)

```



```
knitr::kable(cbind(df_pr, post_summary(results_probit)), digits = 3,
  col.names = c("ML Mean", "ML Std. Dev", "Posterior Mean", "Posterior Std. Dev"))
```

	ML Mean	ML Std. Dev	Posterior Mean	Posterior Std. Dev
(Intercept)	-2.922	0.357	-2.955	0.359
pirat	2.653	0.562	2.686	0.551
hirat	-0.546	0.674	-0.543	0.674
lvrat	0.784	0.257	0.791	0.249
unemp	0.031	0.018	0.030	0.018
chist2	0.322	0.107	0.324	0.106
chist3	0.432	0.164	0.431	0.164
chist4	0.769	0.184	0.769	0.186
chist5	0.608	0.131	0.608	0.136
chist6	0.800	0.125	0.807	0.125
mhist2	0.166	0.099	0.174	0.099
mhist3	0.207	0.264	0.202	0.262
mhist4	0.231	0.358	0.224	0.360
phistyes	0.724	0.120	0.729	0.121
selfempyes	0.337	0.114	0.339	0.114
insurancyes	2.574	0.285	2.617	0.292
condominyes	-0.035	0.090	-0.038	0.090
afamyes	0.380	0.099	0.382	0.100
singleyes	0.237	0.083	0.240	0.081
hschoolyes	-0.558	0.241	-0.553	0.240

```
X_black = X; X_black[, "afamyes"] = 1
X_nblack = X; X_nblack[, "afamyes"] = 0

delta_prob = pnorm(results_probit %*% t(X_black)) - pnorm(results_probit %*% 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: 0.064"
```

An african-american person is more likely to get his application denied 6.4% more than a non-african-american person, given all the factors remain constant.

```
# trace plots
par(mfrow = c(5, 4))
for (i in 1:dim(results_probit)[2]){
  plot(results_probit[, i], type = "l", main = colnames(results_probit)[i],
    font.main=1, ylab="", xlab="", cex.main=1.4)
}
```

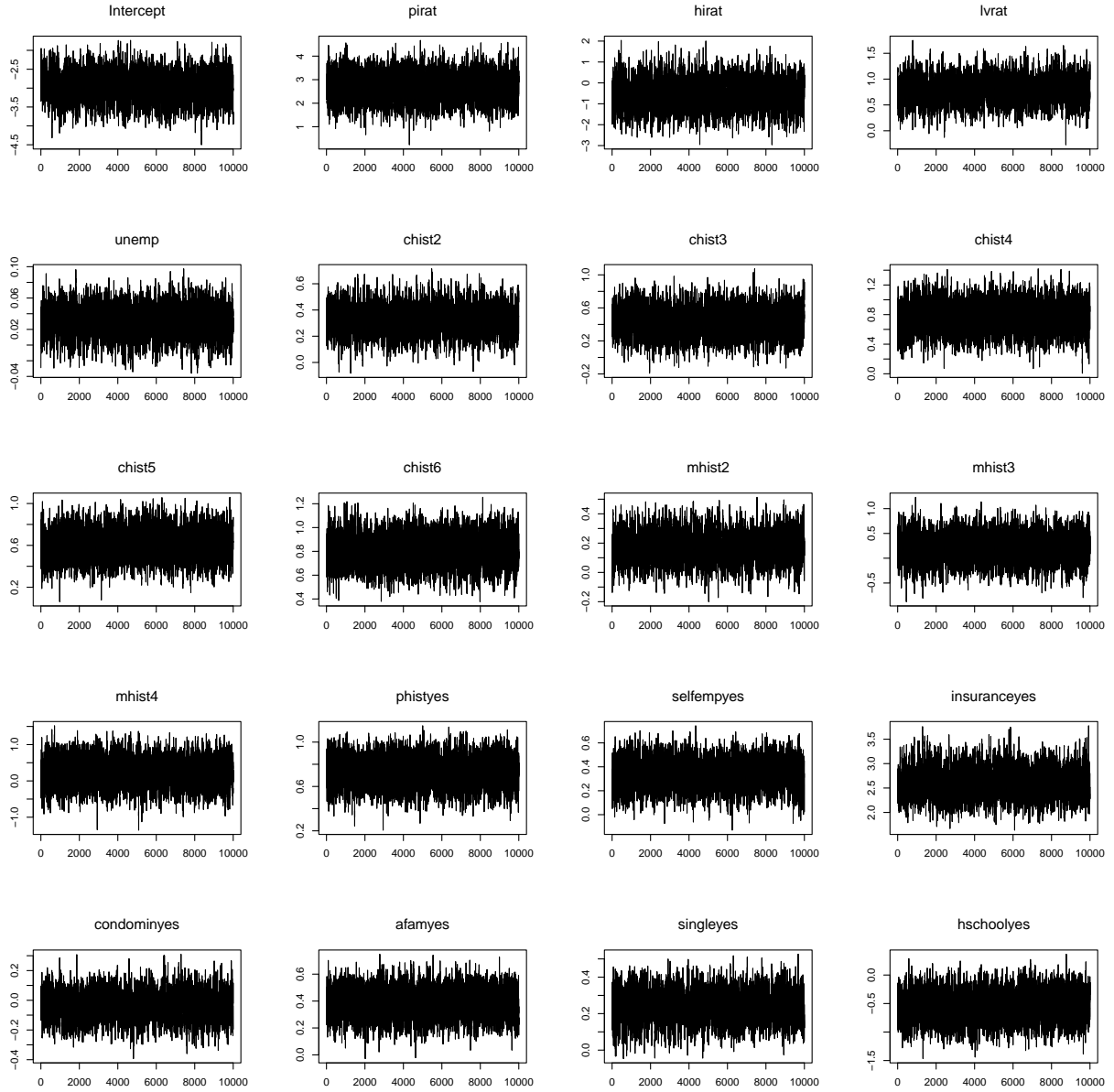


Figure 6: Trace plots of simulated draws for the covariates using binary probit model