Homework 6

STT 465, Bayesian Statistical Methods

Lowell Monis

November 7, 2025

Question 1

The files `menchild30bach.txt` and `menchild30nobach.txt` contain data on the number of children of men in their 30s with and without bachelor's degrees, respectively. Download the files onto your computer, import the data into R using:

```
yA <- scan("data/menchild30bach.txt"); yB <- scan("data/menchild30nobach.txt")
```

Denote the two groups of data by $\mathbf{y^A} = (y_1^A, \ldots, y_{n_A}^A)$ and $\mathbf{y^B} = (y_1^B, \ldots, y_{n_B}^B)$, respectively. We assume Poisson sampling models:

$$Y_1^A, \ldots, Y_{n_A}^A \mid \theta_A \overset{\text{i.i.d.}}{\sim} \text{Poisson}(\theta_A), \quad Y_1^B, \ldots, Y_{n_B}^B \mid \theta_B \overset{\text{i.i.d.}}{\sim} \text{Poisson}(\theta_B)$$

.

But now we parameterize θ_A and θ_B as $\theta_A = \theta$, $\theta_B = \theta \times \gamma$. In this parameterization, γ represents the relative rate $\frac{\theta_B}{\theta_A}$. We consider the independent prior for θ and γ: $\theta \sim \text{Gamma}(a_\theta, b_\theta)$ and $\gamma \sim \text{Gamma}(a_\gamma, b_\gamma)$. We now use Gibbs sampler to approximate the posterior distribution $p(\theta, \gamma \mid \mathbf{y^A}, \mathbf{y^B})$.

(a)

Use the number of children data, and set the parameter values in the prior:

$$a_\theta = 2, \quad b_\theta = 1, \quad a_\gamma = b_\gamma = 8$$

Run a Gibbs sampler of 5000 iterations. Make the traceplots of θ and γ. What do you observe? Further plot the autocorrelation functions (from lag-1 to lag-40) for θ and γ (using the R function `acf()` to compute the autocorrelations). What do you observe? Based on the traceplots and autocorrelation function plots, describe how you use the 5000 samples from Gibbs sampler to approximate $\mathbb{E}(\theta_B - \theta_A \mid \curvearrowright^{\mathbb{A}}, \curvearrowright^{\mathbb{B}})$.

Answer

We know from the previous homework that the full conditional distribution of θ given y^A, y^B, γ takes the form:

$$p(\theta \mid y^A, y^B, \gamma) = \text{Gamma}(\tilde{a}_\theta, \tilde{b}_\theta)$$

where

$$\tilde{a}_\theta = a_\theta + \sum_{i=1}^{n_A} y_i^A + \sum_{i=1}^{n_B} y_i^B$$

$$\tilde{b}_\theta = b_\theta + n_A + n_B \cdot \gamma$$

and that the the full conditional distribution of γ given y^A, y^B, θ takes the form:

$$p(\gamma \mid y^A, y^B, \theta) = \text{Gamma}(\tilde{a}_\gamma, \tilde{b}_\gamma)$$

where

$$\tilde{a}_\gamma = a_\gamma + \sum_{i=1}^{n_B} y_i^B$$

$$\tilde{b}_\gamma = b_\gamma + n_B \cdot \theta$$

We first create variables to store the given information from the question and to implement the prior and posterior.

```
# required descriptive information
nA <- length(yA)
nB <- length(yB)
sA <- sum(yA)
sB <- sum(yB)

# prior parameters
a_theta <- 2
b_theta <- 1
a_gamma <- 8 -> b_gamma
```

We can now set up the Markov Chain Monte Carlo experiment with 5,000 iterations.

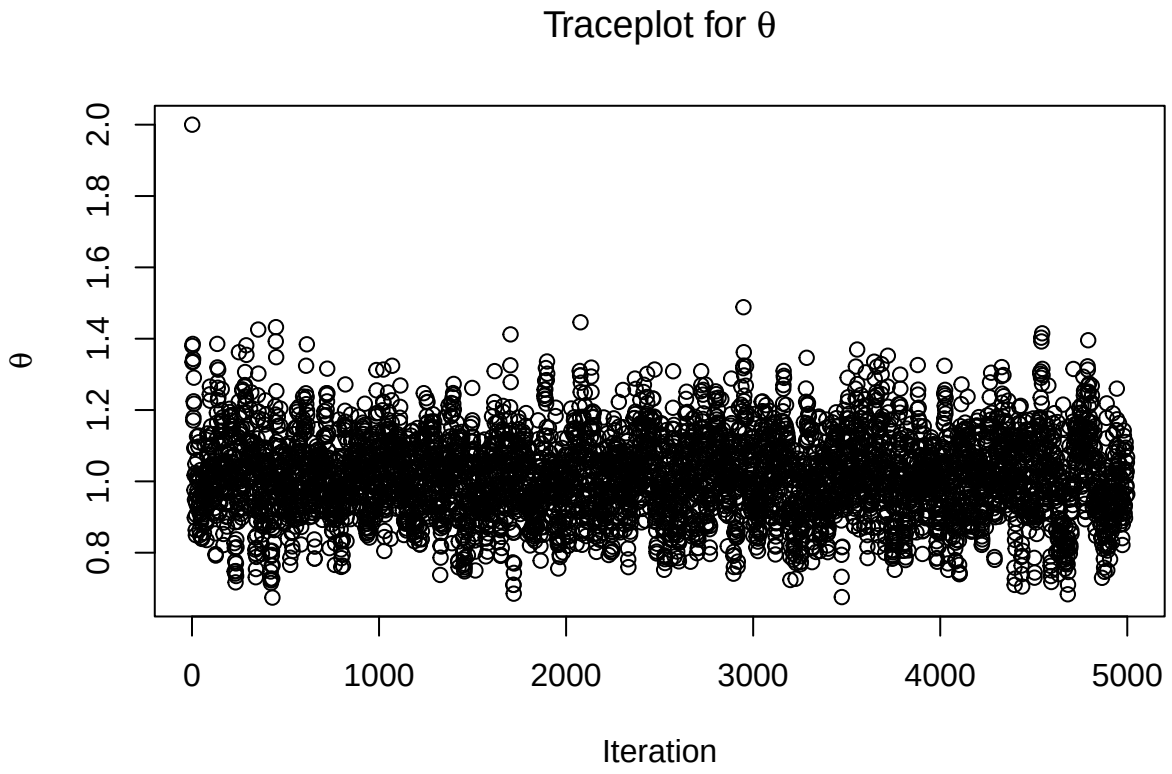
```
S <- 5000 # iterations

# storage
THETA <- numeric(S) -> GAMMA

# prior mean as initial value
THETA[1] <- a_theta/b_theta
GAMMA[1] <- a_gamma/b_gamma
```

We can now set up the Gibbs sampler implementation.

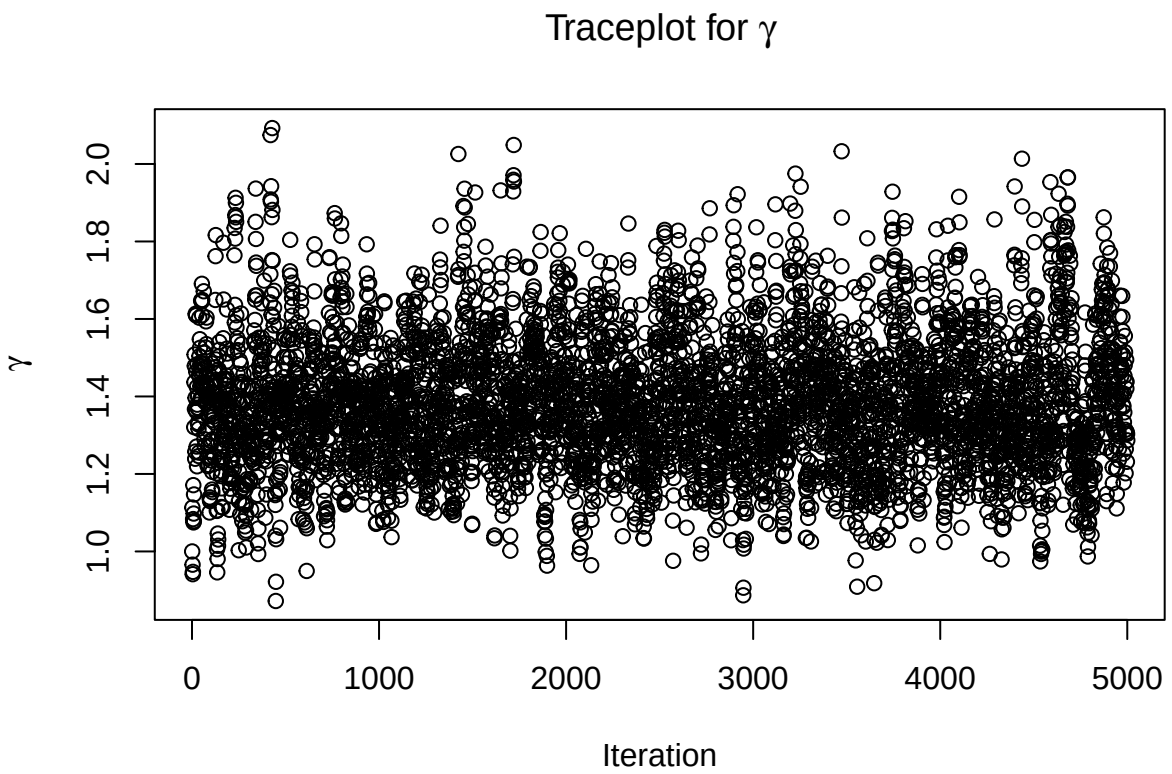
```
set.seed(465)
for (k in 2:S){
  THETA[k] <- rgamma(1, shape = a_theta + sA + sB,
                     rate = b_theta + nA + nB*GAMMA[k-1])
  GAMMA[k] <- rgamma(1, shape = a_gamma + sB,
                     rate = b_gamma + nB * THETA[k])
}
```

We can now proceed with MCMC diagnostic plots.

```
plot(THETA,
     main=expression(paste("Traceplot for ", theta)),
     xlab="Iteration", ylab=expression(theta))
```




```
plot(GAMMA,
     main=expression(paste("Traceplot for ", gamma)),
     xlab="Iteration", ylab=expression(gamma))
```

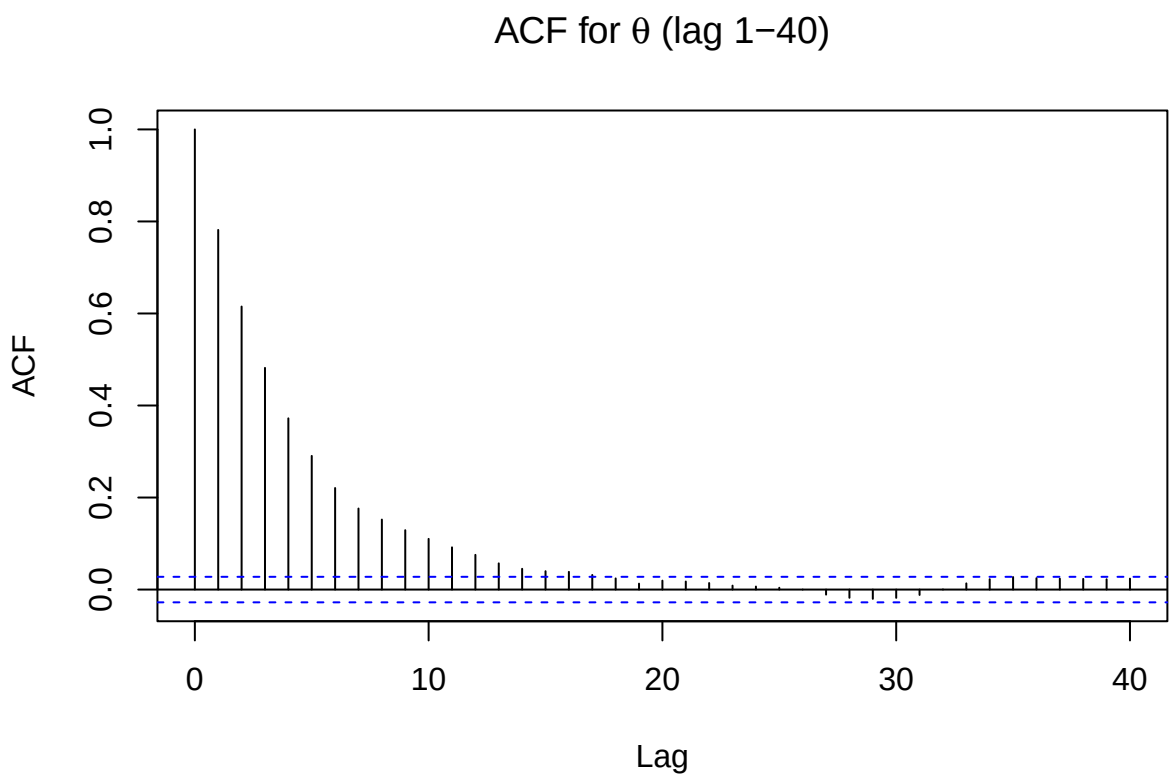


Both traceplots exhibit the characteristics of a well-mixing Markov chain, since they both show fluctuating,

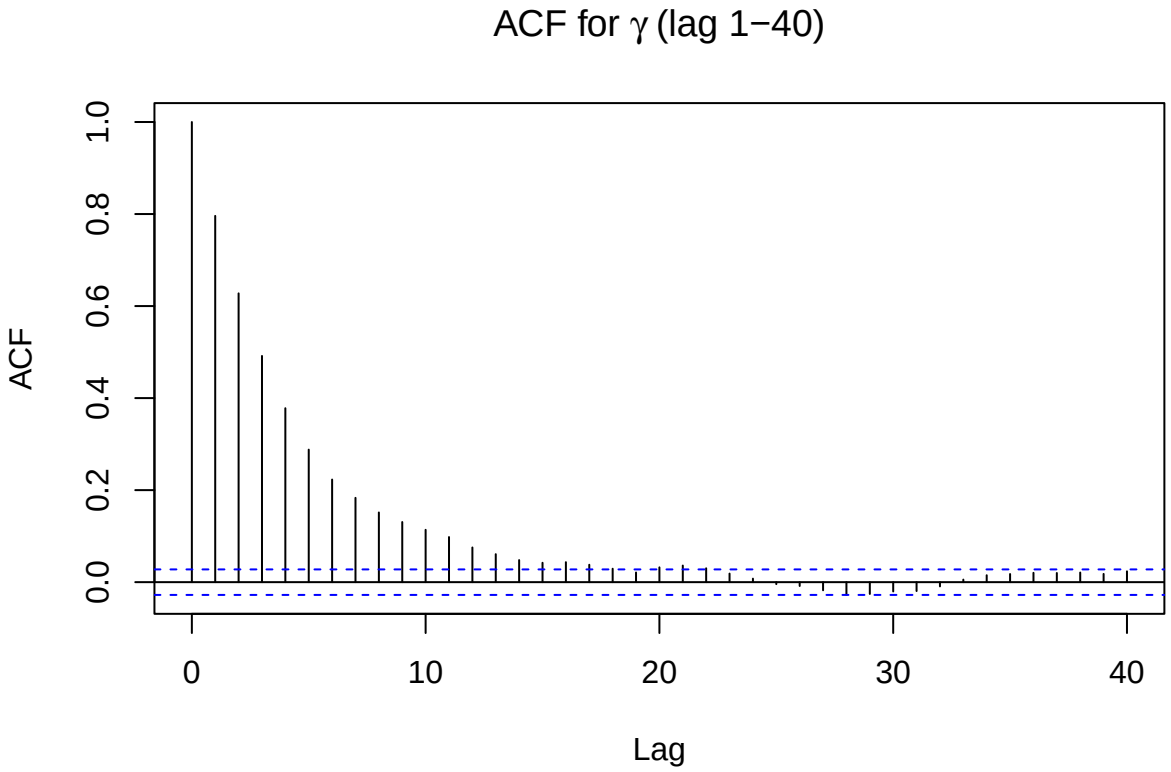
dense, and fuzzy plots. The values for both θ and γ do not appear to be stuck in any particular region and are consistently sampling from the same distribution throughout the 5000 iterations. There is no significant 'burn-in' phase where the chain is moving from a poor starting value, with the exception of the very minimal transient phase for θ in the first few iterations, after which the plot enters the stable region, indicating that both achieve stationarity quickly. The samples show high variability between consecutive iterations, as visible in the fuzzy nature of the plot, indicating that the chain is efficiently exploring the joint posterior distribution. There are no long, smooth or sticky periods where the chain stays in one small region for hundreds of iterations. The range of the region is pretty wide, so there is low autocorrelation and good amounts of mixing. Based on these traceplots, the Gibbs sampler appears to be performing well. The chain has converged and is mixing efficiently, suggesting that the generated samples are a reliable basis for approximating posterior quantities.

Now, I will create the autocorrelation function plots.

```
acf(THETA, lag.max=40,
    main=expression(paste('ACF for ', theta, ' (lag 1-40)')))
```




```
acf(GAMMA, lag.max=40,
    main=expression(paste('ACF for ', gamma, ' (lag 1-40)')))
```



Both ACF plots show positive autocorrelation at small lags, indicating that these consecutive samples are highly correlated, but the autocorrelation decays relatively quickly towards zero, which is desirable for an MCMC algorithm.

For θ and γ both, the lag-1 autocorrelation was very high and almost close to 1.0, but dropped close to zero between lag-15 and lag 20. The chain exhibits some dependency, but the correlation is not excessive, and is generally low. The chain is mixing reasonably well, but the dependency in the beginning may lead to a higher variance in the final estimate, compared to independent samples, which we have less than 5000 of. This slightly reduces the precision of the posterior estimates.

The plots confirm that the Gibbs sampler is generating a dependent sequence of samples, which is expected for MCMC. The quick decay suggests that the chain has acceptable efficiency. If higher precision were required, one can run more iterations try sub-sampling. Since the precision is acceptable, we can now proceed with approximating the posterior expectation of the difference in θ_B and θ_A.

In order to do this, we have to sample a large number of values for θ and γ, which we have already done via MCMC approximation, to leverage the law of large numbers. We then apply prior knowledge that $\theta_A = \theta$ and $\theta_B = \theta\gamma$ to transform the existing samples to the required values. We then compute the mean of these samples. We can also technically discard a small burn-in period, but that may seem unnecessary considering the traceplots shows very minimal burn-in. This discards initial dependence on starting values. The calculated sample average remains a valid, albeit slightly less efficient, estimator, due to the effective independent samples being less than the total number of iterations. An approximation equation is provided below.

$$\mathbb{E}(\theta_B - \theta_A \mid \curvearrowright^{\mathbb{A}}, \curvearrowright^{\mathbb{B}}) \approx \frac{1}{S}\sum_{k=1}^{S}\left(\theta^{(k)}\gamma^{(k)} - \theta^{(k)}\right)$$

The transformation needed is provided below:

$$\theta_B - \theta_A = \theta\gamma - \theta = \theta(\gamma - 1)$$

```r
mean(THETA*(GAMMA-1))
```

```
## [1] 0.3706946
```

Thus, the posterior expectation is about 0.37.

Alternatively, we can also use a small burn in period (b) to get a more efficient estimate.

$\mathbb{E}(\theta_B - \theta_A \mid \curvearrowright^{\mathbb{A}}, \curvearrowright^{\mathbb{B}}) \approx \frac{1}{S-b} \sum_{k=1+b}^{S} \left(\theta^{(k)}\gamma^{(k)} - \theta^{(k)}\right)\$$

Let's take $b = 200$ as a conservative estimate.

```r
burn <- 200
mean(THETA[(burn+1):S]*(GAMMA[(burn+1):S]-1))
```

```
## [1] 0.3714289
```

The value does not change by much, despite the fact that I exceeded the point at which the chain started fluctuating as expected. This indicates that this is a reliable model.

(b) 15 points

Set the parameter values in the prior:

$$a_\theta = 2, \quad b_\theta = 1, \quad a_\gamma = b_\gamma = 128$$

Repeat what is asked in (1a). Moreover, comparing the result of $\mathbb{E}(\theta_B - \theta_A \mid \curvearrowright^{\mathbb{A}}, \curvearrowright^{\mathbb{B}})$ you obtained in (1a) and (1b), explain the effect of prior distribution for γ on the results.

Answer

We first create variables to store the given information from the question. The descriptive variables are being carried forward from (1a).

```r
# prior parameters
a_theta <- 2
b_theta <- 1
a_gamma <- 128 -> b_gamma
```

We can now set up the Markov Chain Monte Carlo experiment with 5,000 iterations.

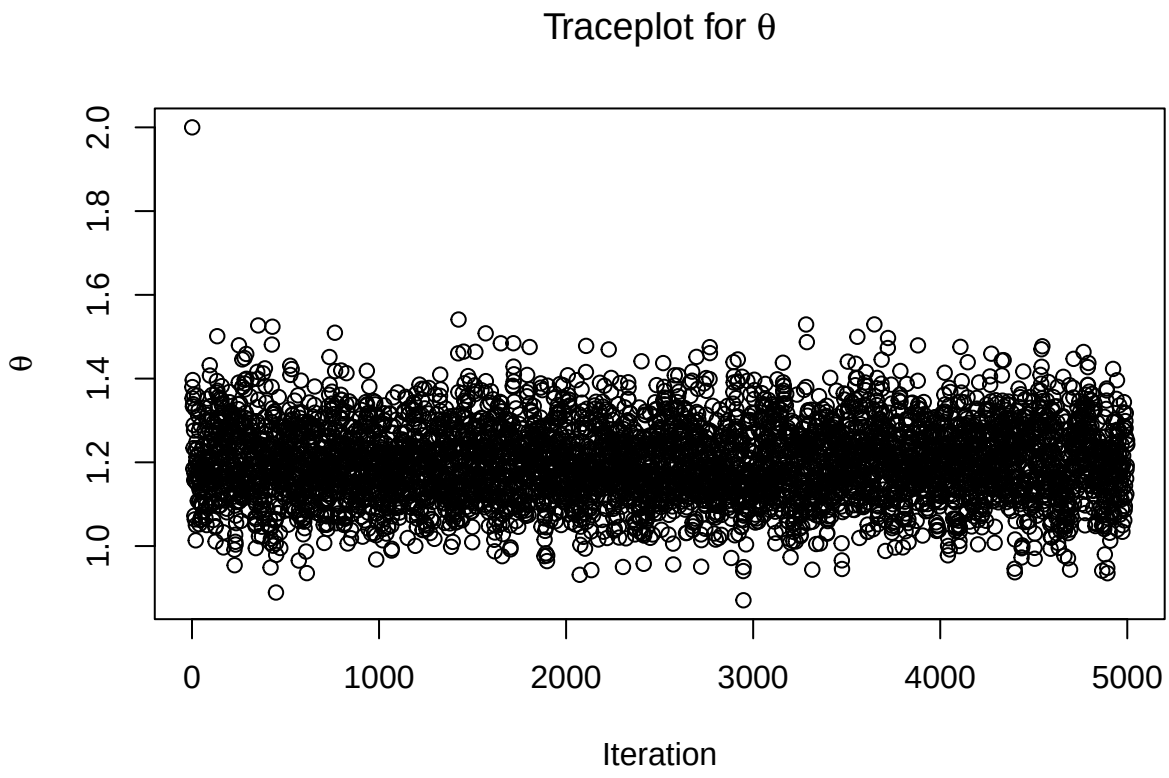
```r
S <- 5000 # iterations

# storage
THETA_n <- numeric(S) -> GAMMA_n

# prior mean as initial value
THETA_n[1] <- a_theta/b_theta
GAMMA_n[1] <- a_gamma/b_gamma
```

We can now set up the Gibbs sampler implementation.

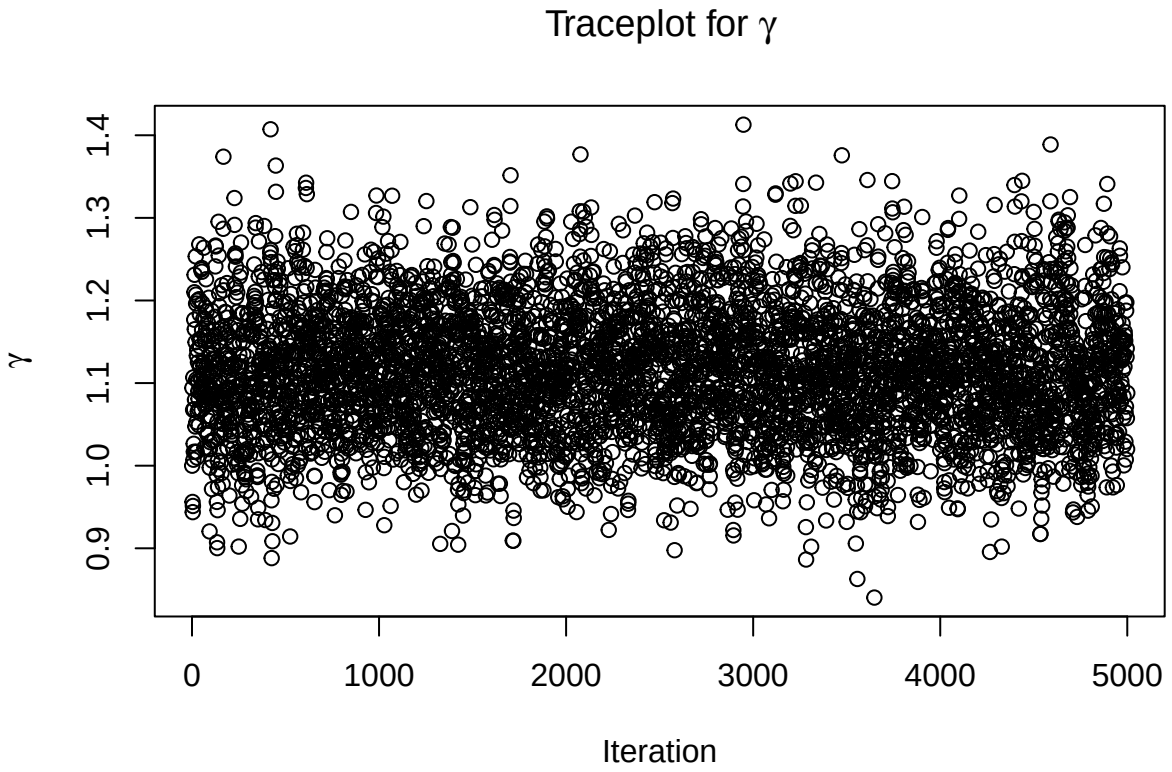
```r
set.seed(465)
for (k in 2:S){
  THETA_n[k] <- rgamma(1, shape = a_theta + sA + sB,
                       rate = b_theta + nA + nB*GAMMA_n[k-1])
  GAMMA_n[k] <- rgamma(1, shape = a_gamma + sB,
                       rate = b_gamma + nB * THETA_n[k])
}
```

We can now proceed with MCMC diagnostic plots.

```r
plot(THETA_n,
     main=expression(paste("Traceplot for ", theta)),
     xlab="Iteration", ylab=expression(theta))
```



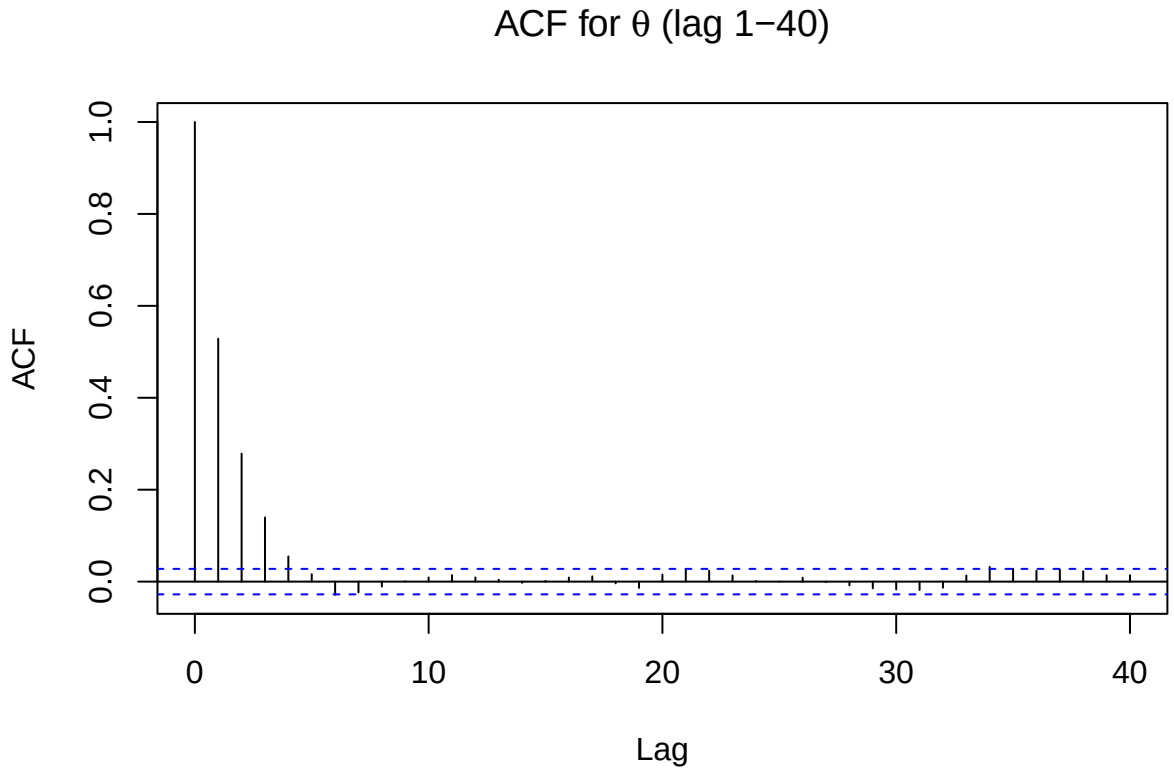

```r
plot(GAMMA_n,
     main=expression(paste("Traceplot for ", gamma)),
     xlab="Iteration", ylab=expression(gamma))
```



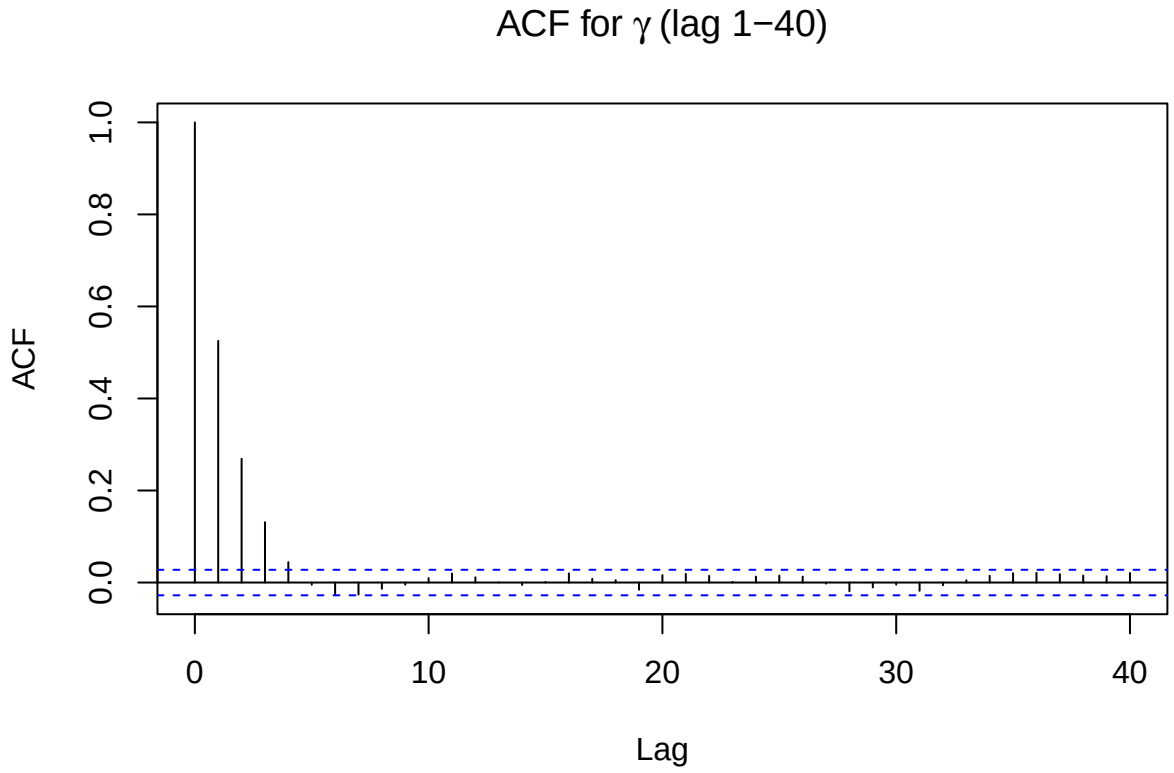
Both traceplots exhibit the characteristics of a well-mixing Markov chain, since they both show fluctuating, dense, and fuzzy plots. The values for both θ and γ do not appear to be stuck in any particular region and are consistently sampling from the same distribution throughout the 5000 iterations. There is no significant 'burn-in' phase where the chain is moving from a poor starting value, with the exception of the very minimal transient phase for θ in the first few iterations, after which the plot enters the stable region, indicating that both achieve stationarity quickly. The samples show high variability between consecutive iterations, as visible in the fuzzy nature of the plot, indicating that the chain is efficiently exploring the joint posterior distribution. There are no long, smooth or sticky periods where the chain stays in one small region for hundreds of iterations. The range of the region has narrowed when compared to the previous plots where $a_\gamma = b_\gamma = 8$. This also shows convergence into the posterior at a rapid rate with very little burn-in, which is an indicator of the fact that the prior provides more information. Based on these traceplots, the Gibbs sampler appears to be performing well. The chain has converged better than the earlier sampler and is mixing efficiently, suggesting that the generated samples are a reliable basis for approximating posterior quantities.

Now, I will create the autocorrelation function plots.

```
acf(THETA_n, lag.max=40,
    main=expression(paste('ACF for ', theta, ' (lag 1-40)')))
```




```
acf(GAMMA_n, lag.max=40,
    main=expression(paste('ACF for ', gamma, ' (lag 1-40)')))
```



Both ACF plots show positive autocorrelation at smaller lags only, indicating that these consecutive samples are highly correlated, but the autocorrelation decays exceptionally quickly towards zero, which is desirable

for an MCMC algorithm. It decays faster than it did for the earlier sampler, at around 4-5 lags for both θ and γ.

For θ and γ both, the lag-1 autocorrelation was very high and almost close to 1.0, but dropped close to zero between lag-4 and lag-5. The chain exhibits very low dependency. This means that the chain is mixing extremely well when compared to the earlier sampler, indicating that the precision of sampled posterior estimates is much better when $a_\gamma = b_\gamma = 8$.

The extremely quick decay suggests that the chain has good efficiency. We can now proceed with approximating the posterior expectation of the difference in θ_B and θ_A.

Repeating the experiment as in (1a), we get:

```
mean(THETA_n*(GAMMA_n-1))
```

```
## [1] 0.1324358
```

Thus, the posterior expectation is about 0.13.

Alternatively, we can also use a small burn in period (b) to get a more efficient estimate.

Let's take $b = 200$ once again as a conservative estimate.

```
burn <- 200
mean(THETA_n[(burn+1):S]*(GAMMA_n[(burn+1):S]-1))
```

```
## [1] 0.1328342
```

The value does not change by much, despite the fact that I exceeded the point at which the chain started fluctuating as expected. This indicates that this is a reliable model.

Now, let's compare the expectation of the difference between the two parameters.

The prior distribution for γ has a significant effect on the posterior estimate of the difference between the two groups. Both priors have the form Gamma(a_γ, b_γ) with $a_\gamma = b_\gamma$, which means they are centered at a prior mean of $\mathbb{E}[\gamma] = \frac{a}{b} = 1$, suggesting prior belief that the two groups have similar rates. However, the precision (or concentration) of these priors differs substantially:

- The Gamma$(8, 8)$ prior has variance $\frac{a}{b^2} = \frac{8}{8^2} = 0.125$. This is a rather weak prior belief.
- The Gamma$(128, 128)$ prior has variance $\frac{a}{b^2} = \frac{128}{128^2} = \frac{1}{128}$. This is a more concentrated or strong prior belief since the variance is rather low. It also explains why the traceplot was more concentrated around a smaller range.

The stronger prior in (1b) pulls the posterior estimate of γ closer to 1, which reduces the estimated difference between θ_B θ_A as observed in our calculation from 0.37 to 0.13. This demonstrates that when we have a more informative prior centered at $\gamma = 1$, it shrinks the posterior estimate toward this value, resulting in a smaller estimated difference between the two groups.

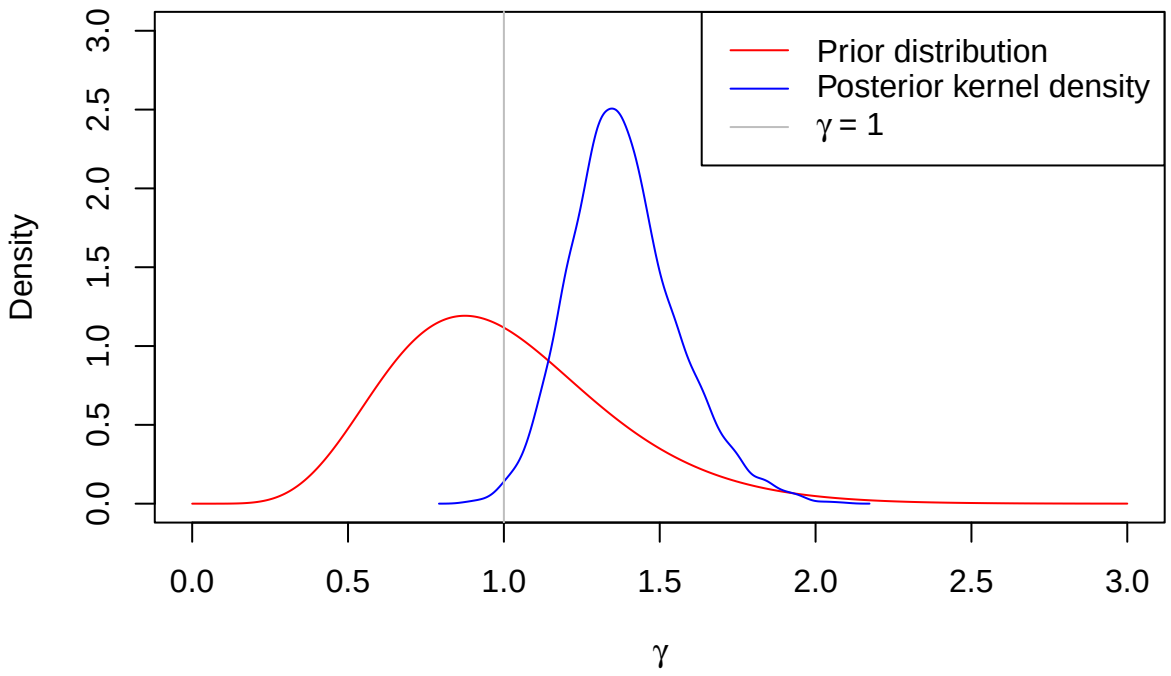
(c) 20 points

Continue from (1a). Plot the prior distribution of γ and the smooth kernel density approximations to the posterior distribution of γ. Explain the difference of these two distributions. Which group of men tends to have more children?

Answer

We will reuse the prior parameters from (1a). We will create a sequence to store the indices for the distribution values. We will then create the prior distribution, and also use the kernel density estimation of the sampled values in `GAMMA` from (1a) to plot the posterior.

```
gamma <- seq(0, 3, length = 10000)
prior <- dgamma(gamma, shape=8, rate=8)
posterior <- density(GAMMA, adjust=1)
plot(gamma, prior, type='l', col='red',
     xlab = expression(gamma),
     ylab = "Density", ylim = c(0, 3))
lines(posterior, col='blue')
abline(v=1, col='gray')
legend('topright',
       legend = c("Prior distribution",
                  "Posterior kernel density",
                  expression(paste(gamma, " = 1"))),
       lty = c(1, 1, 1),
       col = c("red", "blue", "gray"))
```



From the above plot, it looks like the data has updated our prior beliefs, and shifted away from the prior mean of $\frac{a_\gamma}{b_\gamma} = \frac{8}{8} = 1$, to the posterior mean of `mean(GAMMA)`. The shift indicates that the two groups of men have different rates. The posterior plot is also narrower than the prior plot, showing that the uncertainty/spread/variance has also decreased, reflecting an increased precision in our estimate. The posterior distribution shows that after observing the data, the bulk of the probability mass lies above the posterior mean which is greater than 1.

We can also find out which group of men tend to have more children by computing the posterior probability that $\gamma > 1$. $\gamma = \frac{\theta_B}{\theta_A}$. $\gamma > 1 \implies \frac{\theta_B}{\theta_A} > 1 \implies \theta_B > \theta_1$.

```r
mean(GAMMA>1)
```

```
## [1] 0.9944
```

```r
quantile(GAMMA, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.079514 1.764662
```

The posterior probability iss computed above, such that $P(\gamma > 1 \mid \curvearrowright^{\mathbb{A}}, \curvearrowright^{\mathbb{B}}) = 0.9944$. This provides overwhelming evidence that men without bachelor's degrees tend to have more children than men with bachelor's degrees. Moreover, the entire 95% credible interval for the posterior distribution is above 1, which further solidifies that the rate of men without bachelor's degrees having children is higher than those with degrees. In fact, the posterior mean suggests that men without bachelor's degrees have approximately 1.3 to 1.4 times the rate of having children compared to men with bachelor's degrees. The data has strongly updated our prior belief that "both groups are similar" with the prior close to a mean of 1, to "group B clearly has more children" with great certainty and precision.

Question 2

The file "crime.txt" contains crime rates and data on 15 predictor variables for 47 U.S. states. Download the file onto your computer, and import the data into R using:

```r
crime.data <- read.table("data/crime.txt", header = TRUE)
yf <- crime.data[, 1]; Xf <- as.matrix(crime.data[, -1])
```

Then, `yf` has the crime rates and `Xf` has the 15 predictor variables. Both the crime rates and 15 predictor variables have been centered and scaled to have mean zero and variance one. A description of the 15 predictor variables is in the file "crime_variables.pdf".

Fit a linear regression model by treating the crime rates as the response variable and including the 15 predictor variables (no need to add interaction or quadratic terms). Using the semi-conjugate prior with $\beta = 0_{15\times 1}, \Sigma_0 = I_{15\times 15}, a = 1, b = 1$, obtain Monte Carlo approximations to the posterior mean and 95% credible interval for each regression coefficient β_j $(j = 1, 2, \ldots, 15)$. Which variables seem strongly predictive of crime rates?

Answer

First, we set up the parameters and sample storage as defined by the question. We also load the `mvtnorm` package.

```r
library(mvtnorm)

# dimensions
n <- length(yf)
p <- ncol(Xf)

# semi-conjugate prior
beta_0 <- rep(0,p) # 15x1 zero vector
Sigma_0 <- diag(p) # 15x15 identity matrix
```

```r
a <- 1 -> b

S <- 5000 # MCMC samples

# storage
beta_sam <- matrix(0, nrow=S, ncol=p)
sigma2_sam <- numeric(S)
```

We consider the normal linear regression model to be:

$$Y = X\beta + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I_n)$.

Further, Y is the $n \times 1$ response vector (crime rates), X is the $n \times p$ data matrix containing the predictors, β is the $p \times 1$ coefficient vector, and σ^2 is the error variance.

This is equivalent to $Y \mid X, \beta, \sigma^2 \sim \mathcal{N}(X\beta, \sigma^2 I_n)$

We use the semi-conjugate prior where β and σ^2 are independent a priori such that $p(\beta, \sigma^2) = p(\beta) \cdot p(\sigma^2)$ with $\beta \sim \mathcal{N}(\beta_0, \Sigma_0)$ and $\sigma^2 \sim \text{Inverse} - \text{Gamma}(a, b)$.

As declared above, the prior specification for this model is as follows:

$$\beta = 0_{15\times 1}, \quad \Sigma_0 = I_{15\times 15}, \quad a = 1, \quad b = 1$$

Under the semi-conjugate prior, the full conditional distributions for the Gibbs sampler are:

$$p(\beta \mid \sigma^2, y, X) = \mathcal{N}(\beta_n, \Sigma_n)$$

where $\Sigma_n = \left(\Sigma_0^{-1} + \frac{1}{\sigma^2} X^T X\right)^{-1}$ and $\beta_n = \left(\Sigma_0^{-1}\beta_0 + \frac{1}{\sigma^2} X^T y\right)$.

$$p(\sigma^2 \mid \beta, y, X) = \text{Inverse} - \text{Gamma}(\tilde{a}, \tilde{b})$$

where $\tilde{a} = a + \frac{n}{2}$ and $\tilde{b} = b + \frac{1}{2}(y - X\beta)^T(y - X\beta) = b + \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2$.

I can now run the Gibbs sampler and then compute the posterior mean, and further compute the 95% credible interval of the posterior mean samples. I am also adding a flag that checks whether we are 95% confident that the coefficient is non-zero. A variable is only considered strongly predictive if its 95% credible interval does not include zero, i.e., the coefficient is non-zero by a fair chance and contributes the variable greatly to the final model.

```r
# constant quantities
XtX <- t(Xf)%*%Xf
Xty <- t(Xf)%*%yf
Sigma_0_inv <- solve(Sigma_0)

# initial values
beta_sam[1,]<-beta_0
sigma2_sam[1]<-b/a

# Gibbs sampler
set.seed(465)
for (s in 2:S) {
  Sigma_n_inv <- Sigma_0_inv + XtX / sigma2_sam[s-1]
```

```
  Sigma_n <- solve(Sigma_n_inv)
  beta_n <- Sigma_n %*% (Sigma_0_inv %*% beta_0 + Xty / sigma2_sam[s-1])
  beta_sam[s, ] <- rmvnorm(1, mean = beta_n, sigma = Sigma_n)
  residuals <- yf - Xf %*% beta_sam[s, ]
  a_tilde <- a + n / 2
  b_tilde <- b + sum(residuals^2) / 2
  sigma2_sam[s] <- 1 / rgamma(1, shape = a_tilde, rate = b_tilde)
}

# posterior means and 95% credible intervals
posterior_means <- colMeans(beta_sam)
credible_intervals <- t(apply(beta_sam, 2, quantile, probs = c(0.025, 0.975)))

var_names <- colnames(Xf)
results <- data.frame(
  Variable = var_names,
  Posterior_Mean = round(posterior_means, 4),
  CI_Lower = round(credible_intervals[, 1], 4),
  CI_Upper = round(credible_intervals[, 2], 4),
  CI_Excludes_Zero = (credible_intervals[, 1] > 0) | (credible_intervals[, 2] < 0)
)
results
```

```
##    Variable Posterior_Mean CI_Lower CI_Upper CI_Excludes_Zero
## 1         M         0.2780  -0.0033   0.5614            FALSE
## 2        So         0.0281  -0.3611   0.4088            FALSE
## 3        Ed         0.4874   0.1095   0.8484             TRUE
## 4       Po1         0.7945  -0.2711   1.8520            FALSE
## 5       Po2        -0.0816  -1.1919   1.0320            FALSE
## 6        LF        -0.0200  -0.3251   0.2990            FALSE
## 7       M.F         0.1478  -0.1808   0.4638            FALSE
## 8       Pop        -0.0602  -0.3242   0.1975            FALSE
## 9        NW         0.0769  -0.2752   0.4073            FALSE
## 10       U1        -0.2449  -0.6535   0.1668            FALSE
## 11       U2         0.3583  -0.0173   0.7284            FALSE
## 12      GDP         0.2114  -0.3119   0.7283            FALSE
## 13     Ineq         0.6894   0.2121   1.1721             TRUE
## 14     Prob        -0.2581  -0.5284   0.0240            FALSE
## 15     Time        -0.0242  -0.2935   0.2583            FALSE
```

Upon running our model, it looks like `Ed` and `Ineq` are strongly predictive of crime rates. This means that the number of years of schooling and income inequality are the strongest predictors of crime rates in the context of this data.
