Introduction

In the realm of public safety and urban planning, understanding the factors that contribute to serious crimes is of paramount importance. The frequency and severity of such crimes not only impact the well-being and quality of life of residents but also influence economic development, property values, and community cohesion. For policymakers, urban planners, and law enforcement agencies, identifying and mitigating the predictors of serious crimes can lead to more effective strategies in crime prevention and community development. We aim to explore how variables such as poverty levels, unemployment rates, or education levels influence the incidence of serious crimes.

Exploratory Data Analysis

First, to convert the response variable into a continuous and scaled variable, we will be using the log(Yi), which we will refer to as the crime rate. At first glance, the first variables that caught our attention were total population, below poverty levels, unemployment, capita income, and personal income. From the pairs plot (Fig. 1) and correlation matrix (Tab. 1), we visualize any patterns between the relationship between $\log(y_i)$and previously mentioned predictor variables. The strongest linear pattern with serious crimes are total population and personal income, while poverty levels, unemployment, and capita income have less/no apparent patterns. Moreover, we can see a strong linear relationship of total population and personal income.

To further investigate, we produce box plots to examine the spreads of these predictor variables and will be specifically looking at region 1, the northeastern region of the US. This is done by grouping based on values, so: group 1 is crime rate, below poverty levels, and unemployment; group 2: capita income and personal income; and group 3 is population. From group 1, poverty level has a positively skewed distribution while the crime rate and unemployment rate are mostly normal distributed. In groups 2 and 3, we also see a positively skewed distribution of personal income and total population while capita income is normally distributed. Comparing the summary statistics for region 1 and the entire dataset, we see the means of each variable are mostly comparable. However, there are maximum outliers regarding total population, below poverty levels, unemployment, and personal income, while crime rate and capita income maxes are comparable. The capita income is a scale based on personal income divided by total population. Based on what we discovered, we choose capita income which is scaled based on total personal income and population and below poverty level which has a greater correlation with crime rate than unemployment.

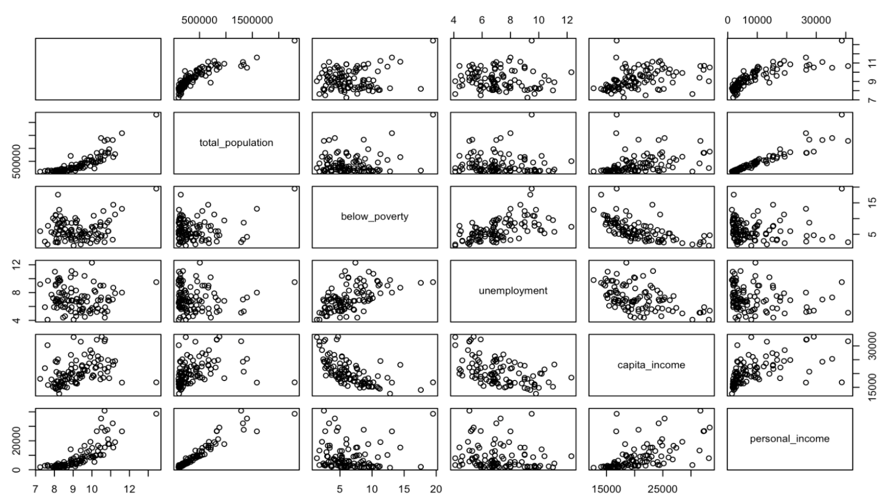
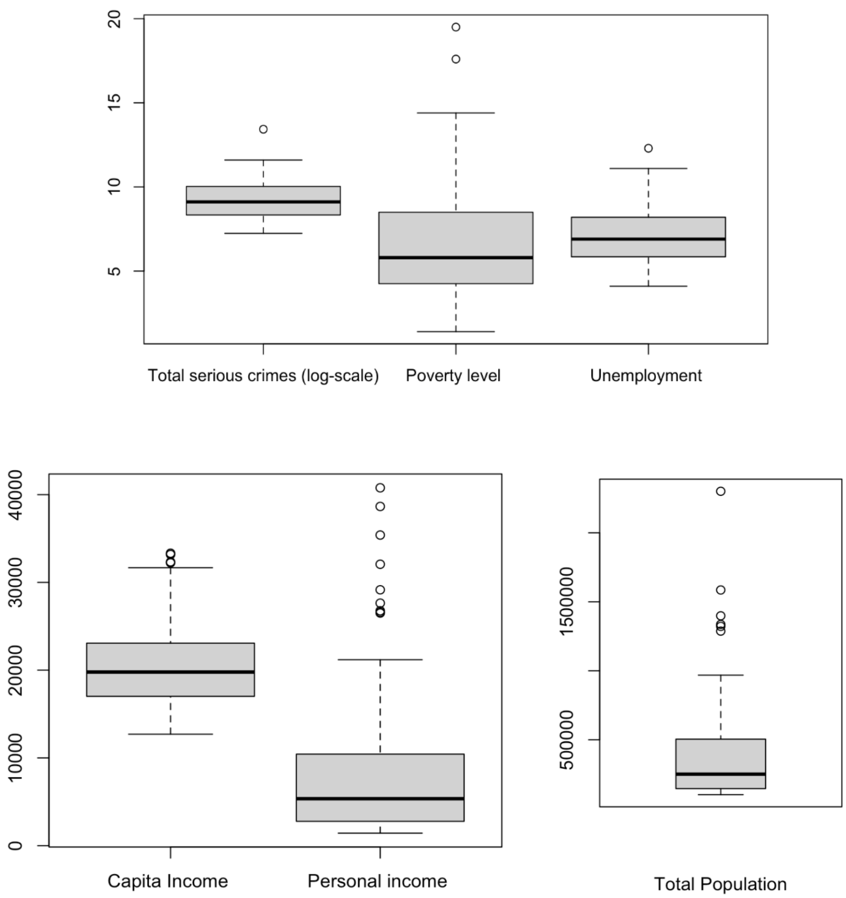


Figure 1: Pairs Plot for Region 1

		total_population	below_poverty	unemployment	capita_income	personal_income
	1.00000000	0.86255702	0.190109041	-0.07243098	0.3911163	0.825587718
total_population	0.86255702	1.00000000	0.191586867	-0.04066225	0.3425751	0.946680206
below_poverty	0.19010904	0.19158687	1.000000000	0.56153750	-0.6156135	-0.003797269
unemployment	-0.07243098	-0.04066225	0.561537504	1.00000000	-0.4871536	-0.154447719
capita_income	0.39111626	0.34257513	-0.615613537	-0.48715355	1.0000000	0.559656565
personal_income	0.82558772	0.94668021	-0.003797269	-0.15444772	0.5596566	1.000000000

Table 1: Correlation Matrix for Region 1



Figures 2a-c: Boxplots for region 1

```
[1] "region 1"
       V1          total_population  below_poverty     unemployment     capita_income    personal_income
 Min.   : 7.242   Min.   : 102525   Min.   : 1.400   Min.   : 4.100   Min.   :12704   Min.   : 1423  
 1st Qu.: 8.333   1st Qu.: 146584   1st Qu.: 4.250   1st Qu.: 5.850   1st Qu.:17016   1st Qu.: 2771  
 Median : 9.115   Median : 250836   Median : 5.800   Median : 6.900   Median :19785   Median : 5352  
 Mean   : 9.248   Mean   : 395834   Mean   : 6.499   Mean   : 7.166   Mean   :20599   Mean   : 8734  
 3rd Qu.:10.033   3rd Qu.: 504574   3rd Qu.: 8.500   3rd Qu.: 8.200   3rd Qu.:23079   3rd Qu.:10432  
 Max.   :13.431   Max.   :2300664   Max.   :19.500   Max.   :12.300   Max.   :33330   Max.   :40782  
```

```
[1] "Demographic"
       V1          total_population  below_poverty     unemployment     capita_income    personal_income 
 Min.   : 6.333   Min.   : 100043   Min.   : 1.400   Min.   : 2.200   Min.   : 8899   Min.   :  1141  
 1st Qu.: 8.735   1st Qu.: 139027   1st Qu.: 5.300   1st Qu.: 5.100   1st Qu.:16118   1st Qu.:  2311  
 Median : 9.378   Median : 217280   Median : 7.900   Median : 6.200   Median :17759   Median :  3857  
 Mean   : 9.503   Mean   : 393011   Mean   : 8.714   Mean   : 6.594   Mean   :18561   Mean   :  7869  
 3rd Qu.:10.177   3rd Qu.: 436064   3rd Qu.:10.900   3rd Qu.: 7.500   3rd Qu.:20270   3rd Qu.:  8654  
 Max.   :13.443   Max.   :8863164   Max.   :36.300   Max.   :21.300   Max.   :37541   Max.   :184230  
```

Figure 3a-b: Summary Statistics for Region 1 and Demographic

Methodology, Results, & Discussion:

We utilize Bayesian regression to estimate the posterior distributions of regression coefficients (β) and variance (σ^2). To achieve this, we implement Monte Carlo and Gibbs sampling algorithms to sample from γ and β. This allows us to sample from the joint posterior distribution of the model parameters, thereby enabling us to make probabilistic inferences about the parameters of interest.

Data Preparation:

We focus on a specific region, region_1, and construct the design matrix X

X using the predictor variables below_poverty and capita_income. The response variable y, where y is the logarithm of the serious crime rates ($\log(y_i)$).

Prior Distributions:

$$\beta \sim N(\beta_0, \Sigma_0)$$

$$\sigma^2 \sim \text{Inverse-Gamma}(v_0, \sigma^2_0), \text{ since } \gamma = \frac{1}{\sigma^2}$$

Prior parameters:

$$\beta_0 = (0,0,0)$$

$$\Sigma_0 = \text{diag}(10002,10002,10002)$$

$$v_0 = 0$$

$$\sigma^2_0 = 1$$

Gibbs Sampling Algorithm:

The Gibbs sampler iteratively sample from the conditional posterior distributions of β_0 and σ^2_0, where each iteration S starts from 1 to 5000.

- **Updating** β

Where $Var_{\beta} = (\Sigma_0^{-1} + \gamma X^T y)^{-1}$, $E_{\beta} = V_{\beta}(\Sigma_0^{-1}\beta_0 + \gamma X^T y)$, and β sample from

$N(E_{\beta}, V_{\beta})$.

- **Updating** γ

Where, posterior shape $v_n = v_0 + n$, and the rate $ss_n = v_0\sigma^2_0 + \Sigma(y - X\beta)^2$, and

γ os sample from gama($\frac{V_n}{2}$,$\frac{SS_n}{2}$).

Through Gibbs Sampling algorithm, we able to generate sample from joint posterior distribution of the regression coefficients β and precision γ. Code can be seen in line

To identify the uncertainty around the parameter estimates, we compute the 95% confidence interval as the frequentist approach and 95% credible intervals as the Bayesian approach for the regression coefficient β_j.

- 95% Confidence Intervals

  By using the frequentist linear regression model, we were able to compute the 95% confidence interval for the β_j. The interval shows a range which the true parameter values should lies, results shows:

```
[1] "95% Confidence Interval:"
                    2.5 %       97.5 %
(Intercept)   2.645455418 5.0308861545
below_poverty 0.166438341 0.2886981990
capita_income 0.000147372 0.0002343111
```

We are 95% confident that the true value of below_poverty is between 2.6455 to 5.0309. In addition to it, we are also 95% confident that the true value of capita_income is between 0.0001 and 0.0002. Since the parameter does not include zero, it suggests that the parameter is statistically significant at the 5% significance level.

- 95% Credible Intervals

  By using the posterior samples obtained from the Gibbs sampler, we computed the 95% credible intervals for the parameters β_j. These intervals provide a range within the true parameter values that are expected to lie within 95% probability, given the observed data and prior distributions. Result shows that:

```
[1] "95% Credible interval"
                      2.5%        97.5%
(Intercept)   2.6571940014 5.0588229110
below_poverty 0.1654749417 0.2884748739
capita_income 0.0001466952 0.0002347437
```

  Here, we can see that there is a 95% probability that the true value of the below_poverty is between 0.1655 to 0.2885. Other than that, the true value of the capital_income is between 0.0001 to 0.0002. Since the credible interval for a parameter does not include zero, it suggests that the parameter is likely to have a real effect, given the prior information and the observed data.

In addition, the interval for both frequentist and bayesian approaches are close to similar; it suggests that the results are stable and reliable across different statistical approaches.

To test whether each regression coefficient (β_j) is significantly different from zero, we computed the marginal posterior distributions using the Bayesian approach and compared them with the Frequentist approach results. Considering $\alpha = 0.01$. The two alternatives are:

$$H_o: \beta_j = 0 \quad vs \quad H_a: \beta_j \neq 0$$

- Bayesian Approach

  Below shows the mean of β_0, β_1 , β_2:

```
> beta_mean
              below_poverty capita_income
3.8267231923  0.2252447223  0.0001879732
```

  Below is the result p-values for the regression coefficients β_0, β_1 , β_2:

```
              below_poverty capita_income
2.566778e-05  1.556441e-06  1.123645e-08
```

  Since p-values for all three regression coefficients are significantly smaller than the significance level of $\alpha = 0.01$, leading us to reject the null hypotheses.

- Frequentist Approach

We use the linear model summary to obtain the p-values and the point estimates.

```
Call:
lm(formula = log(Y_i) ~ below_poverty + capita_income, data = region_1)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.3874 -0.4880  0.1249  0.4244  1.9488 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3.838e+00  6.012e-01   6.384 5.44e-09 ***
below_poverty 2.276e-01  3.081e-02   7.386 4.64e-11 ***
capita_income 1.908e-04  2.191e-05   8.710 6.59e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8083 on 100 degrees of freedom
Multiple R-squared:  0.4519,	Adjusted R-squared:  0.441 
F-statistic: 41.23 on 2 and 100 DF,  p-value: 8.741e-14
```

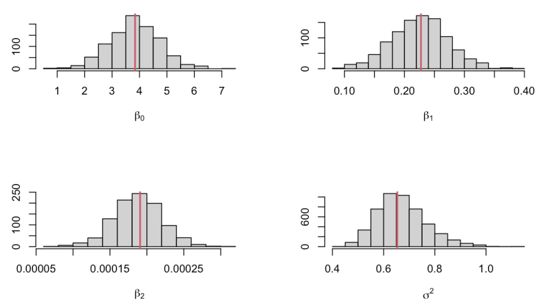
The frequentist approach shows a similar result that all the p-values are very small and smaller than significance level of = 0.01, leading us to reject the null hypotheses.

To assess the fit of the model and the assumptions underlying the regression model, we compute the residuals for each fitted model and prepare the diagnostic plots for each fitted model.

- Interpretation of Posterior distribution by histogram

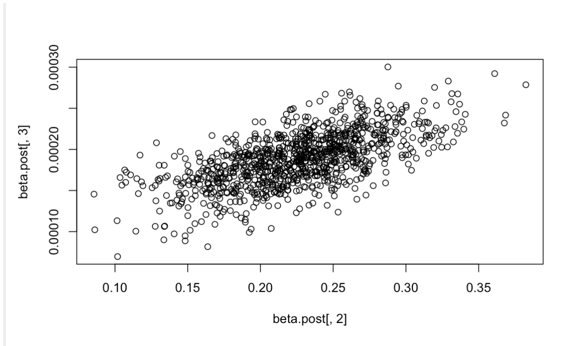
  Here we generated histograms the posterior distributions of the regression coefficients β_0, β_1 , β_2 and σ^2. Overall, the spread of all four histograms is centered around the mean, suggesting the frequentist estimates are very close to the posterior means for all parameters, indicating strong agreement between the two methods.

```
       V1          below_poverty      capita_income     
 Min.   :0.8116   Min.   :0.08557   Min.   :6.965e-05  
 1st Qu.:3.2326   1st Qu.:0.19390   1st Qu.:1.659e-04  
 Median :3.8222   Median :0.22687   Median :1.890e-04  
 Mean   :3.8267   Mean   :0.22524   Mean   :1.880e-04  
 3rd Qu.:4.4163   3rd Qu.:0.25580   3rd Qu.:2.103e-04  
 Max.   :7.3401   Max.   :0.38246   Max.   :3.003e-04  
```



- Diagnostic plot for joint Posterior Distribution of β_1 and β_2

The scatter plot shows a positive correlation between β_1 and β_2, showing that the coefficient for below_poverty increases, the coefficient for capita_income also tends to increase. Additionally, since most points are scattered in the center suggest that most posterior samples are clustered around the mean values, with fewer points appearing at the tails, indicating the spread of the posterior distribution. This indicates the importance of these predictors in the regression model, reinforcing their significance in explaining the variation in serious crime rates.



- Residual

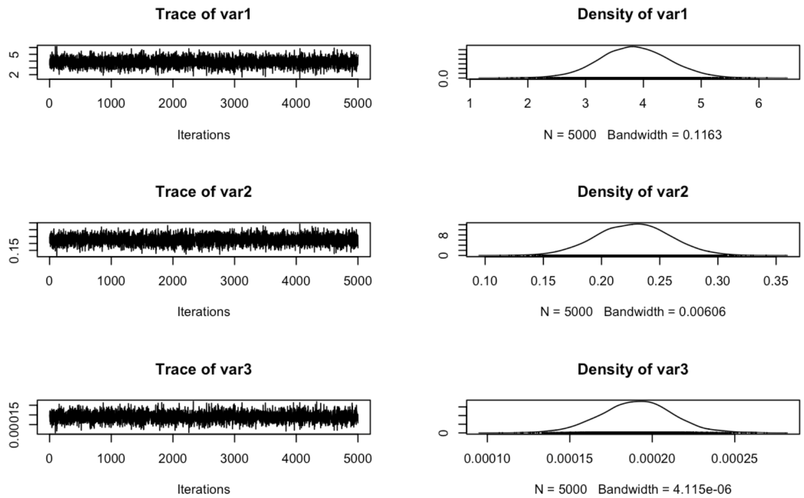
  To access the fit of our Bayesian regression model and validate the underlying assumptions, we computed the residuals.Residuals are the differences between the observed values and the predicted values, standardized by the posterior samples of the error variance.

$$\text{Residuals} = \frac{log(Y_i) - (\beta X^T)}{\sqrt{\sigma^2}}$$

In order to ensure the reliability of our Bayesian model estimates we do several diagnostics to check on the MCMC sample.

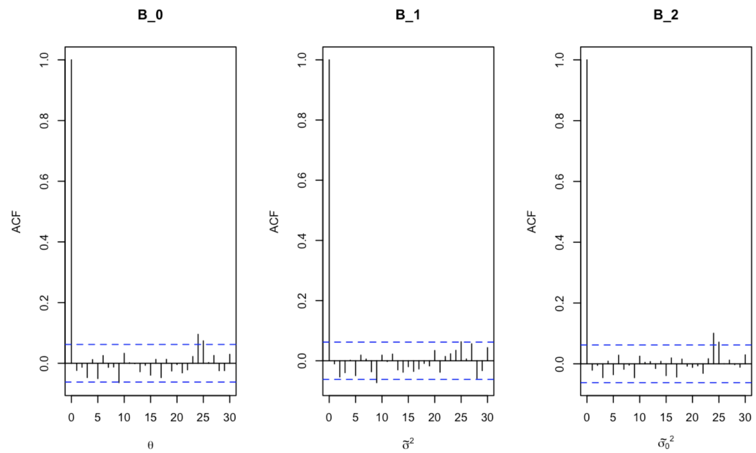
- Trace Plots and Density Plots

  This is to assess the convergence and distribution of the samples. As a result the trace plots show the sampled values of each parameter over iterations, indicating the stability and mixing of the chains. This is shown across β_0, β_1, and β_2 samples are fluctuating around a stable mean value. Other than that, the density plots display the posterior distributions of the parameters, providing insights into their central tendencies and variabilities. As a result, the distributions appear to be approximately normal, with means centered around the estimated values.



- Autocorrelation Function (ACF) Plots

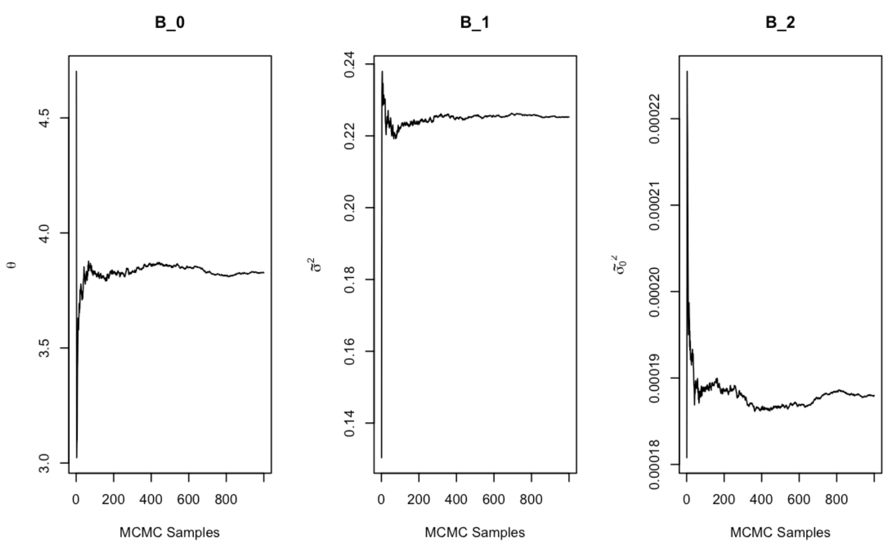
To identify the the autocorrelation of the posterior samples for the regression coefficients β_0, β_1 and β_2. We generated ACF plots where the y-axis represents the autocorrelation values, and the x-axis represents the lag number. By observing the plots we notice that for all three parameters, we got a result of low autocorrelation at various lags. This is an indication of good mixing and independence of the MCMC samples, ensuring the sample are effectively exploring the posterior distribution, leading to reliable parameter estimates



.

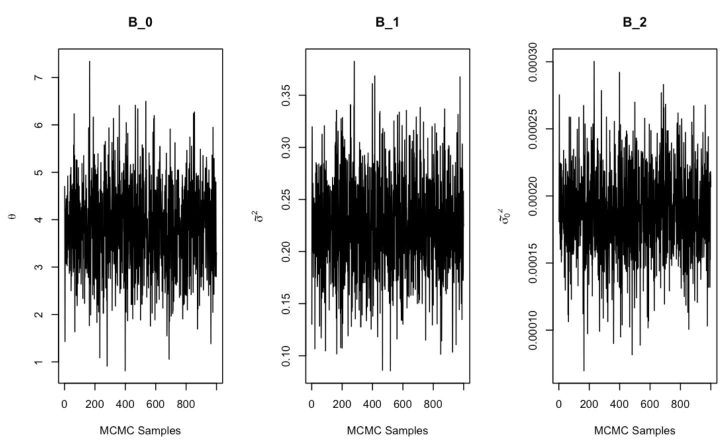
- Ergodic Mean Plots

To assess the stability and convergence of the MCMC algorithm, we generated ergodic mean plots for the posterior samples of the regression coefficients β_0, β_1 and β_2. As a result, it seems that the cumulative means stabilize after an initial period. This stabilization indicates that the MCMC chains have likely converged to the true posterior distributions. Therefore, the parameter estimates are reliable and that the MCMC algorithm has adequately explored the parameter space.



- Mixing Plots

  To figure out the mixing and convergence of MCMC algorithm, we fetch the trace plot for the posterior samples of the regression coefficients β_0, β_1 and β_2. Thus, all three parameters indicate that the samples fluctuate around stable mean values without exhibiting any trends or patterns. Thus, the MCMC chains have probably converged to their true posterior distributions as a result of good mixing. When MCMC samples are mixed properly, parameter estimates are reliable because the posterior distributions are represented by the samples.



To improve the estimation of our model parameters, we introduce a hyper-g prior for g.

Prior for g:

$\pi(g) \approx \frac{1}{(1+g)^2}$

This is a hyper g- prior, that is proper since when we integrate it is equal to 1. Therefor the Gibbs sampling algorithm is similar as before, just in regards of updating β ,σ^2 and g are done differently.

Gibbs Sampling Algorithm:

The Gibbs sampler iteratively samples from the conditional posterior distributions of 0 and 20 , where each iteration S starts from 1 to 5000.

- **Updating β**

  Where $Var_{\beta} = (\frac{g}{g+1}X^{T}X)^{-1}$, $E_{\beta} = V_{\beta}(\frac{g}{g+1}X^{T}y)$, and β sample from $N(E_{\beta}, V_{\beta})$.

- **Updating σ^2**

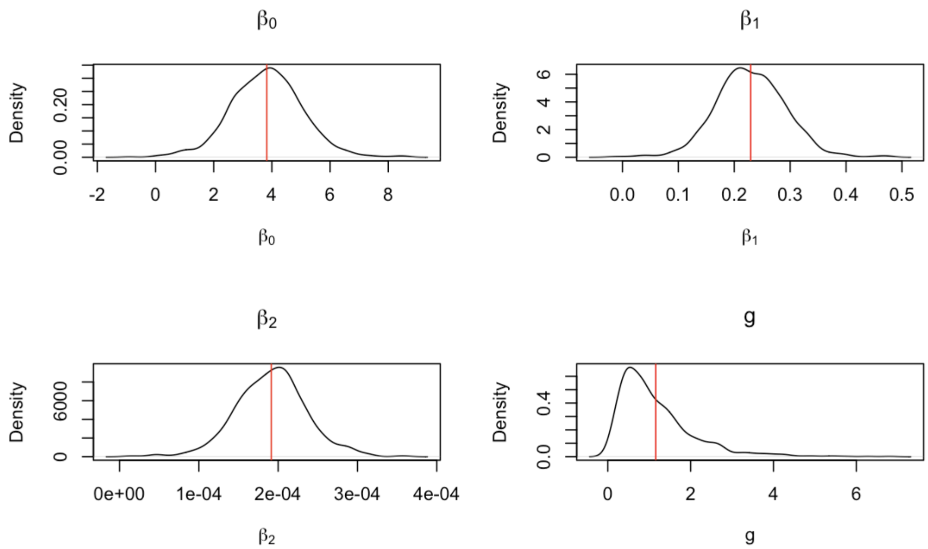
  Where, posterior shape $v_{n} = v_{0} + n$, and the rate $ss_{n} = v_{0}\sigma^{2}_{0} + \Sigma(y - X\beta)^{2}$, and σ^{2} os sample from inverse gamma($\frac{V_n}{2}$,$\frac{SS_n}{2}$).

- Updating g:

  Where $p(g|\ \beta, \sigma^{2}, y) = \frac{1}{(1+g)^{2}} exp\{- \frac{1}{2\sigma^{2}}(y - X\beta)^{T}(y - X\beta)\}$

Using the Gibbs sampling algorithm with the hyper-g prior, we are able to generate samples from the joint posterior distribution.

Next we observed the posterior distribution of β_0, β_1 , β_2 and g. The red line indicates the mean of the posterior distributions. From the plot we can see that the distribution for β_0, β_1 , β_2 is approximately normal. Other than that, most distributions significantly differ from zero, indicating that both "below_poverty" and "capita_income" are important predictors. However the distribution of g is right skewed with most distributions concentrated around 0 and 4, indicating its informative but flexible.



The analysis of serious crime rates in the northeastern region of the United States has provided valuable insights into the socioeconomic factors that contribute to crime. Through the application of both frequentist and Bayesian statistical methods, the study identified significant predictors of crime rates, particularly poverty levels and capita income.

145project

Ming Gan and Kristina Mooc

2024-05-24

```
Demographic <- read.table("/Users/mingqiangan/Downloads/STA 145 Final/data/Demographic.txt")

colnames(Demographic)
<-c('ID','County','state','Land_area','total_population','Population_18to34','Population_65','Physicians','beds','Y_i','Graduate_highschool','
Graduate_Bachelor','below_poverty','unemployment','capita_income','personal_income','Geographic_region')

head(Demographic)
```

```
##   ID      County state Land_area total_population Population_18to34
## 1  1 Los_Angeles    CA      4060          8863164              32.1
## 2  2        Cook    IL       946          5105067              28.2
## 3  3      Harris    TX      1729          2818199              28.3
## 4  4   San_Diego    CA      4205          2498016              33.5
## 5  5      Orange    CA       790          2410556              32.6
## 6  6       Kings    NY        71          2300664              28.3
##   Population_65 Physicians  beds    Y_i Graduate_highschool Graduate_Bachelor
## 1           8.7      23677 27700 688936                70.0              22.3
## 2          10.4      15153 21550 436936                73.4              22.8
## 3           6.1       7553 12449 253526                74.9              24.4
## 4          10.9       5905  6179 173821                81.9              25.3
## 5           9.2       6062  6369 144524                81.2              27.8
## 6          12.4       4861  8942 680966                63.7              16.6
##   below_poverty unemployment capita_income personal_income Geographic_region
## 1          11.6          8.0         20786          184230                 4
## 2          11.1          7.2         21729          110928                 2
## 3          12.5          5.7         19517           55003                 3
## 4           8.1          6.1         19588           48931                 4
## 5           5.2          4.8         24400           58818                 4
## 6          19.5          9.5         16803           38658                 1
```

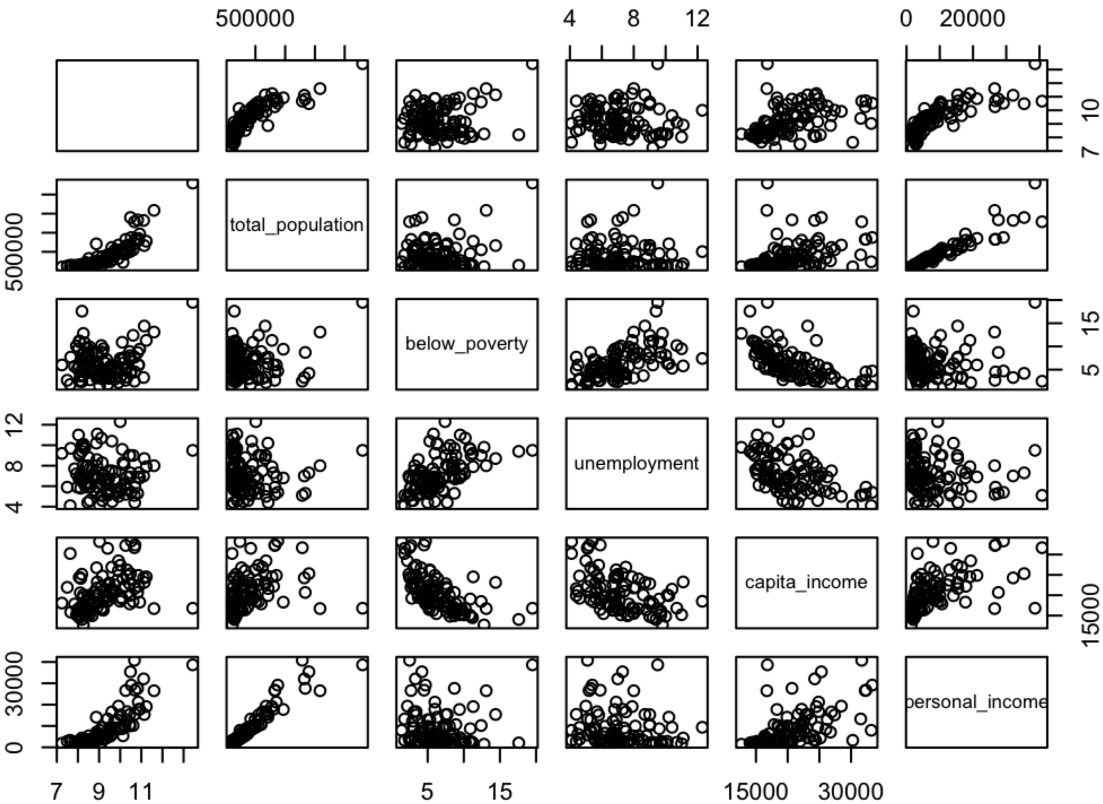
```
dim(Demographic)
```

```
## [1] 440  17
```

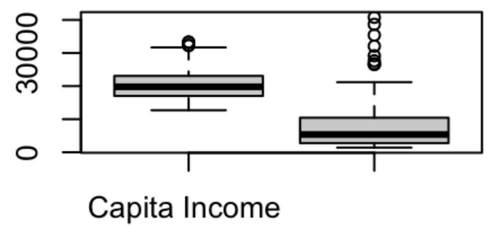
```
attach(Demographic)

region_1 <- Demographic[Geographic_region == 1,]
region_2 <- Demographic[Geographic_region == 2,]
region_3 <- Demographic[Geographic_region == 3,]
region_4 <- Demographic[Geographic_region == 4,]

n1<-nrow(region_1)
n2<-nrow(region_2)
n3<-nrow(region_3)
n4<-nrow(region_4)

summary(Demographic)
```

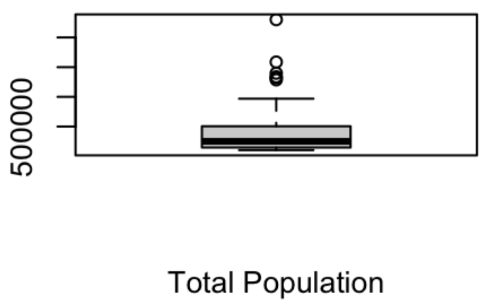
```
##        ID           County             state             Land_area
##  Min.   :  1.0   Length:440         Length:440         Min.   :   15.0
##  1st Qu.:110.8   Class :character   Class :character   1st Qu.:  451.2
##  Median :220.5   Mode  :character   Mode  :character   Median :  656.5
```

```
##  Mean   :220.5                                   Mean   : 1041.4
##  3rd Qu.:330.2                                   3rd Qu.:  946.8
##  Max.   :440.0                                   Max.   :20062.0
##  total_population  Population_18to34 Population_65      Physicians
##  Min.   : 100043   Min.   :16.40     Min.   : 3.000   Min.   :   39.0
##  1st Qu.: 139027   1st Qu.:26.20     1st Qu.: 9.875   1st Qu.:  182.8
##  Median : 217280   Median :28.05     Median :11.700   Median :  401.0
##  Mean   : 393011   Mean   :28.55     Mean   :12.152   Mean   :  988.0
##  3rd Qu.: 436064   3rd Qu.:30.00     3rd Qu.:13.600   3rd Qu.: 1036.0
##  Max.   :8863164   Max.   :49.70     Max.   :33.800   Max.   :23677.0
##       beds              Y_i         Graduate_highschool Graduate_Bachelor
##  Min.   :   92.0   Min.   :    563   Min.   :46.60       Min.   : 8.10
##  1st Qu.:  390.8   1st Qu.:  6220   1st Qu.:73.78       1st Qu.:15.18
##  Median :  755.0   Median : 11820   Median :77.70       Median :19.70
##  Mean   : 1458.6   Mean   : 27112   Mean   :77.52       Mean   :21.05
##  3rd Qu.: 1575.8   3rd Qu.: 26280   3rd Qu.:82.40       3rd Qu.:25.23
##  Max.   :27700.0   Max.   :688936   Max.   :92.90       Max.   :52.30
##  below_poverty     unemployment    capita_income   personal_income
##  Min.   : 1.400   Min.   : 2.200   Min.   : 8899   Min.   :  1141
##  1st Qu.: 5.300   1st Qu.: 5.100   1st Qu.:16118   1st Qu.:  2311
##  Median : 7.900   Median : 6.200   Median :17759   Median :  3857
##  Mean   : 8.714   Mean   : 6.594   Mean   :18561   Mean   :  7869
##  3rd Qu.:10.900   3rd Qu.: 7.500   3rd Qu.:20270   3rd Qu.:  8654
##  Max.   :36.300   Max.   :21.300   Max.   :37541   Max.   :184230
##  Geographic_region
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :2.461
##  3rd Qu.:3.000
##  Max.   :4.000

## The following objects are masked from Demographic:
##
##     beds, below_poverty, capita_income, County, Geographic_region,
##     Graduate_Bachelor, Graduate_highschool, ID, Land_area,
##     personal_income, Physicians, Population_18to34, Population_65,
##     state, total_population, unemployment, Y_i

## [1] "Region 1: Pairs and Correlation"
```



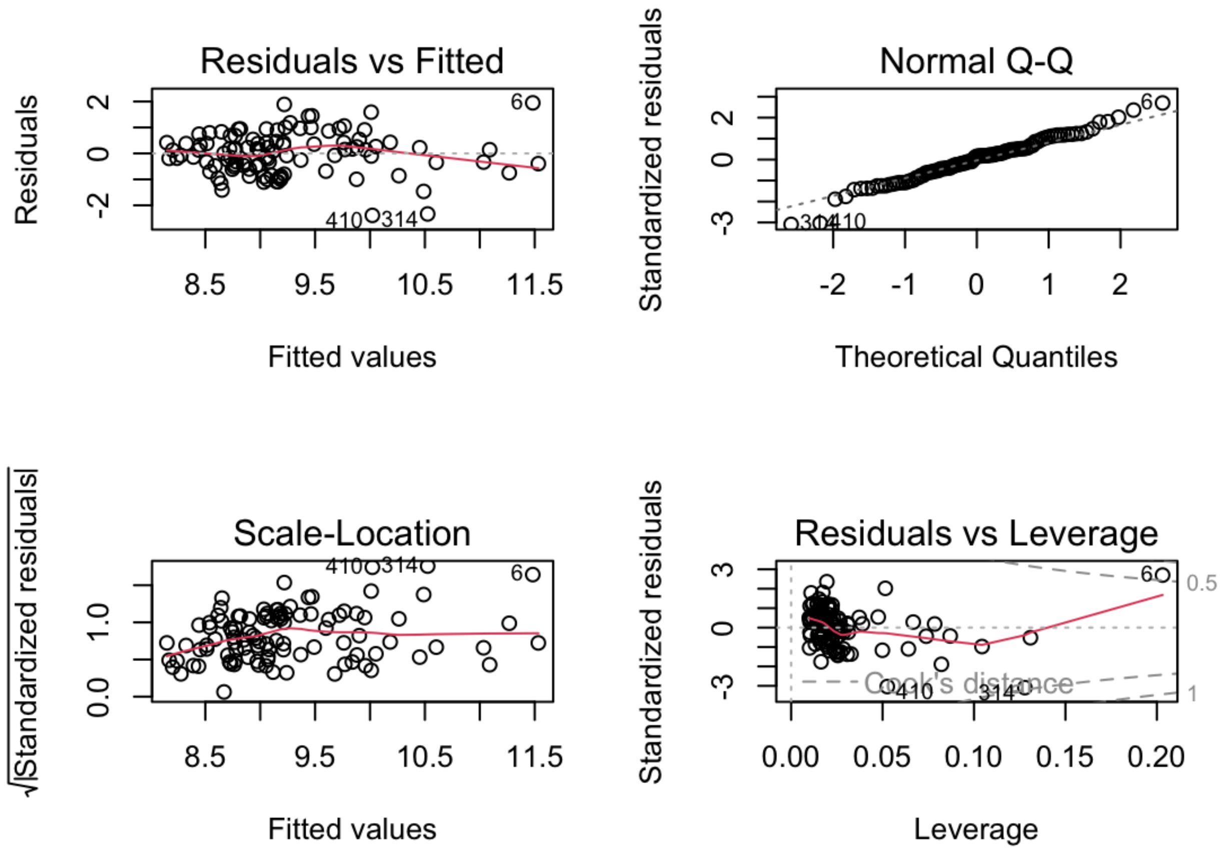

```
##                      total_population below_poverty unemployment
##                  1.00000000       0.86255702   0.190109041  -0.07243098
## total_population  0.86255702       1.00000000   0.191586867  -0.04066225
## below_poverty     0.19010904       0.19158687   1.000000000   0.56153750
## unemployment     -0.07243098      -0.04066225   0.561537504   1.00000000
## capita_income     0.39111626       0.34257513  -0.615613537  -0.48715355
## personal_income   0.82558772       0.94668021  -0.003797269  -0.15444772
##                  capita_income personal_income
##                     0.3911163     0.825587718
## total_population    0.3425751     0.946680206
## below_poverty      -0.6156135    -0.003797269
## unemployment       -0.4871536    -0.154447719
## capita_income       1.0000000     0.559656565
## personal_income     0.5596566     1.000000000

## [1] "Region 1: Boxplots"
```






```
## [1] "Region 1: Summary statistics"

##        V1         total_population  below_poverty     unemployment
##  Min.   : 7.242   Min.   : 102525   Min.   : 1.400   Min.   : 4.100
##  1st Qu.: 8.333   1st Qu.: 146584   1st Qu.: 4.250   1st Qu.: 5.850
##  Median : 9.115   Median : 250836   Median : 5.800   Median : 6.900
##  Mean   : 9.248   Mean   : 395834   Mean   : 6.499   Mean   : 7.166
##  3rd Qu.:10.033   3rd Qu.: 504574   3rd Qu.: 8.500   3rd Qu.: 8.200
##  Max.   :13.431   Max.   :2300664   Max.   :19.500   Max.   :12.300
##  capita_income   personal_income
##  Min.   :12704   Min.   : 1423
##  1st Qu.:17016   1st Qu.: 2771
##  Median :19785   Median : 5352
##  Mean   :20599   Mean   : 8734
##  3rd Qu.:23079   3rd Qu.:10432
##  Max.   :33330   Max.   :40782

## The following objects are masked from Demographic (pos = 3):
##
##     beds, below_poverty, capita_income, County, Geographic_region,
##     Graduate_Bachelor, Graduate_highschool, ID, Land_area,
##     personal_income, Physicians, Population_18to34, Population_65,
##     state, total_population, unemployment, Y_i
```

```
## [1] "Demographic: Summary statistics"

##        V1         total_population  below_poverty     unemployment
##  Min.   : 6.333   Min.   : 100043   Min.   : 1.400   Min.   : 2.200
##  1st Qu.: 8.735   1st Qu.: 139027   1st Qu.: 5.300   1st Qu.: 5.100
##  Median : 9.378   Median : 217280   Median : 7.900   Median : 6.200
##  Mean   : 9.503   Mean   : 393011   Mean   : 8.714   Mean   : 6.594
##  3rd Qu.:10.177   3rd Qu.: 436064   3rd Qu.:10.900   3rd Qu.: 7.500
##  Max.   :13.443   Max.   :8863164   Max.   :36.300   Max.   :21.300
##  capita_income   personal_income
##  Min.   : 8899   Min.   :  1141
##  1st Qu.:16118   1st Qu.:  2311
##  Median :17759   Median :  3857
##  Mean   :18561   Mean   :  7869
##  3rd Qu.:20270   3rd Qu.:  8654
##  Max.   :37541   Max.   :184230

#==================================
# Frequentest (region 1)
#==================================
model_r_1 <- lm(log(Y_i)~  below_poverty + capita_income , data = region_1)
#summary(lm(log(Y_i)~  below_poverty + capita_income , data = region_2))
#summary(lm(log(Y_i)~  below_poverty + capita_income , data = region_3))
#summary(lm(log(Y_i)~  unemployment + capita_income , data = region_4))

summary(model_r_1)

##
## Call:
## lm(formula = log(Y_i) ~ below_poverty + capita_income, data = region_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3874 -0.4880  0.1249  0.4244  1.9488
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.838e+00  6.012e-01   6.384 5.44e-09 ***
## below_poverty 2.276e-01  3.081e-02   7.386 4.64e-11 ***
## capita_income 1.908e-04  2.191e-05   8.710 6.59e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8083 on 100 degrees of freedom
## Multiple R-squared:  0.4519, Adjusted R-squared:  0.441
## F-statistic: 41.23 on 2 and 100 DF,  p-value: 8.741e-14

#diagnostic plots
par(mfrow=c(2,2))
plot(model_r_1)
```



Write the Monte Carlo and Gibbs sampler algorithms to sample from y and b. (20 points).

```
#==================================
# Bayesian
#==================================
#choose region
attach(region_1)

#==============================================
# Bayesian estimation via MCMC (Monte Carlo)
#==============================================

# below_poverty + capita_income
n<-length(Y_i)#number of rows for specific region
X<-cbind(rep(1,n),below_poverty , capita_income ) #choose two variables
p<-dim(X)[2] #number of columns
y<-log(Y_i) #continuous  y

#set priors
beta.0<-rep(0,p) ;
Sigma.0<-diag(c(1000,1000,1000)^2,p)
nu.0<-1 ;
sigma2.0<- 1
```

```
S<-5000 #5000 samples

rmvnorm<-function(n,mu,Sigma)
{ # samples from the multivariate normal distribution
  E<-matrix(rnorm(n*length(mu)),n,length(mu))
  t(  t(E%*%chol(Sigma)) +c(mu))
}

iSigma.0<-solve(Sigma.0) #initialize
XtX<-t(X)%*%X #X^2?

## store mcmc samples in these objects
beta.post<-matrix(nrow=S,ncol=p)
sigma2.post<-rep(NA,S)

## starting value
set.seed(1)
sigma2<- var(residuals(lm(y~0+X))) #initialize

## MCMC algorithm
for( scan in 1:S) {
  #update beta
  V.beta<- solve(  iSigma.0 + XtX/sigma2 ) #posterior variance
  E.beta<- V.beta%*%( iSigma.0%*%beta.0 + t(X)%*%y/sigma2 ) #posterior mean
  beta<-t(rmvnorm(1, E.beta,V.beta) ) #samples MVN with posterior

  #update sigma2
  nu.n<- nu.0+n #shape = 1+n /2
  ss.n<-nu.0*sigma2.0 + sum((y-X%*%beta)^2) #rate = (1*1^2 + RSS/err or) /2
  sigma2<-1/rgamma(1,nu.n/2, ss.n/2)  #inverse gamma

  #save results of this scan
  beta.post[scan,]<-beta
  sigma2.post[scan]<-sigma2
}
#
library(coda)
# Convert to mcmc objects for analysis
beta_samples_mcmc <- mcmc(beta.post)
```

Compute 95% confidence intervals and 95% quantile based credible intervals for the parameters beta_j , j = 1, .., p. (20 points).

```
## [1] "95% Confidence Interval:"

##                  2.5 %       97.5 %
## (Intercept)   2.645455418 5.0308861545
## below_poverty 0.166438341 0.2886981990
## capita_income 0.000147372 0.0002343111

## [1] "95% Credible interval"

##                    2.5%        97.5%
## (Intercept)   2.6571940014 5.0588229110
## below_poverty 0.1654749417 0.2884748739
## capita_income 0.0001466952 0.0002347437
```

Consider alpha= 0.01 and compute the p-value for the two alternatives: H0 : beta_j = 0 versus Ha : beta_j != 0.

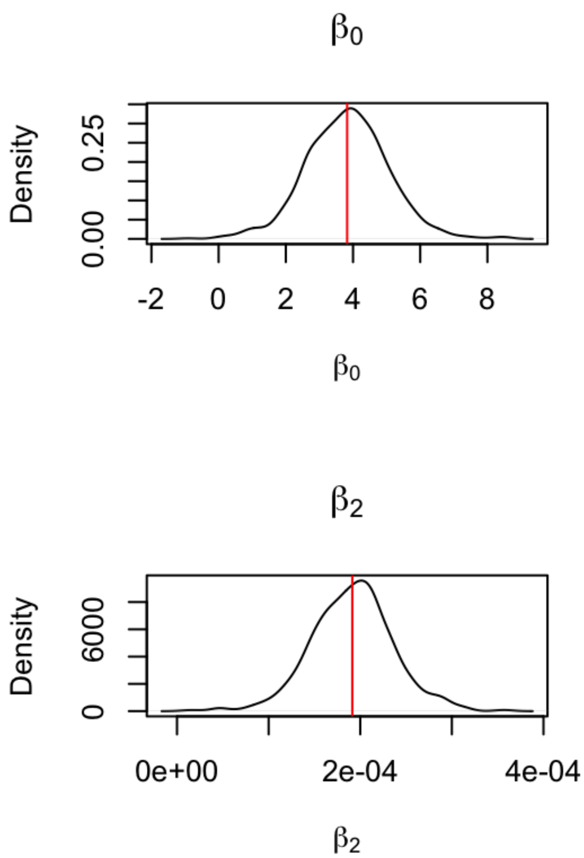
```
# bayes approch
# Compute the mean of the posterior samples
beta_mean <- apply(beta.post, 2, mean)

# Compute the standard deviation of the posterior samples
beta_sd <- apply(beta.post, 2, sd)

# Calculate the t-statistic for each beta_j
t_stat <- beta_mean / beta_sd

# Compute the p-value for the two-sided test
p_values <- 2 * (1 - pnorm(abs(t_stat)))

# Print the p-values
print(p_values)

## [1] 2.075764e-10 4.853895e-13 0.000000e+00
```

Obtain the residuals for each fitted model and prepare the diagnostic plots for each fitted model. State the conclusions. (20 points).

```
##        V1              V2               V3
##  Min.   :1.503   Min.   :0.1129   Min.   :0.0001073
##  1st Qu.:3.440   1st Qu.:0.2056   1st Qu.:0.0001764
##  Median :3.833   Median :0.2275   Median :0.0001907
##  Mean   :3.846   Mean   :0.2270   Mean   :0.0001906
##  3rd Qu.:4.247   3rd Qu.:0.2483   3rd Qu.:0.0002050
##  Max.   :6.141   Max.   :0.3414   Max.   :0.0002695

##
## Iterations = 1:5000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean        SD  Naive SE Time-series SE
## [1,] 3.8456791 6.051e-01 8.557e-03      8.557e-03
## [2,] 0.2270060 3.140e-02 4.441e-04      4.441e-04
## [3,] 0.0001906 2.207e-05 3.121e-07      3.121e-07
##
## 2. Quantiles for each variable:
##
##           2.5%       25%       50%      75%     97.5%
## var1 2.6571940 3.4395062 3.8333472 4.247012 5.0588229
## var2 0.1654749 0.2055971 0.2274517 0.248300 0.2884749
## var3 0.0001467 0.0001764 0.0001907 0.000205 0.0002347
```

```
gibbs_sampler_with_gprior <- function(y, X, g_init = 1, nu0 = 1, s20 = 1, S = 1000) {
  n <- nrow(X)
  p <- ncol(X)

  # Initialize storage matrices
  beta_post <- matrix(NA, nrow = S, ncol = p)
  sigma2_post <- numeric(S)
```

```
 g_post <- numeric(S)

 # Initial values
 g <- g_init
 sigma2 <- var(residuals(lm(y ~ 0 + X)))

 # Inverse-gamma parameters
 nu_n <- nu0 + n
 iXX <- solve(t(X) %*% X)

 for (s in 1:S) {
  # Update beta
  V_beta <- solve(g / (g + 1) * t(X) %*% X)
  E_beta <- V_beta %*% (g / (g + 1) * t(X) %*% y)
  beta <- t(rmvnorm(1, E_beta, V_beta))

  # Update sigma2
  ss_n <- nu0 * s20 + sum((y - X %*% beta)^2)
  sigma2 <- 1 / rgamma(1, nu_n / 2, ss_n / 2)

  # Update g (using Metropolis-Hastings)
  g_proposal <- rgamma(1, shape = 2, rate = 1) # example proposal distribution
  log_acceptance_ratio <- -2 * log(1 + g_proposal) + ( -sum((y - X %*% beta)^2) / (2 * sigma2) ) - (-2 * log(1 + g) + (-sum((y - X
%*% beta)^2) / (2 * sigma2)))
  if (log(runif(1)) < log_acceptance_ratio) {
   g <- g_proposal
  }

  # Store samples
  beta_post[s, ] <- beta
  sigma2_post[s] <- sigma2
  g_post[s] <- g
 }

 list(beta = beta_post, sigma2 = sigma2_post, g = g_post)
}

# Run the Gibbs sampler with g-prior
set.seed(1)
gibbs_results <- gibbs_sampler_with_gprior(log(Y_i), cbind(1, below_poverty, capita_income), S = 1000)

# Summary of the results
beta_post <- gibbs_results$beta
sigma2_post <- gibbs_results$sigma2
g_post <- gibbs_results$g

# Plot the posterior densities
par(mfrow = c(2, 2))
plot(density(beta_post[, 1]), main = expression(beta[0]), xlab = expression(beta[0]))
abline(v = mean(beta_post[, 1]), col = "red")
plot(density(beta_post[, 2]), main = expression(beta[1]), xlab = expression(beta[1]))
abline(v = mean(beta_post[, 2]), col = "red")
plot(density(beta_post[, 3]), main = expression(beta[2]), xlab = expression(beta[2]))
abline(v = mean(beta_post[, 3]), col = "red")
plot(density(g_post), main = expression(g), xlab = expression(g))
abline(v = mean(g_post), col = "red")
```




```
# Frequentist results for comparison
freq_model <- lm(log(Y_i) ~ below_poverty + capita_income)
summary(freq_model)
```

```
##
## Call:
## lm(formula = log(Y_i) ~ below_poverty + capita_income)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3874 -0.4880  0.1249  0.4244  1.9488
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.838e+00  6.012e-01   6.384 5.44e-09 ***
## below_poverty 2.276e-01  3.081e-02   7.386 4.64e-11 ***
## capita_income 1.908e-04  2.191e-05   8.710 6.59e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8083 on 100 degrees of freedom
## Multiple R-squared:  0.4519, Adjusted R-squared:  0.441
## F-statistic: 41.23 on 2 and 100 DF,  p-value: 8.741e-14
```

```
confint(freq_model)

##                  2.5 %       97.5 %
## (Intercept)   2.645455418 5.0308861545
## below_poverty 0.166438341 0.2886981990
## capita_income 0.000147372 0.0002343111
```

Appendix

```
knitr::opts_chunk$set(echo = TRUE)

source("/Users/mingqiangan/Downloads/STA 145 Final/regression_gprior.R")
source("/Users/mingqiangan/Downloads/STA 145 Final/backselect.R")
library("ash")
Demographic <- read.table("/Users/mingqiangan/Downloads/STA 145 Final/data/Demographic.txt")

colnames(Demographic)
<-c('ID','County','state','Land_area','total_population','Population_18to34','Population_65','Physicians','beds','Y_i','Graduate_highschool','Graduate_Bachelor','below_poverty','unemployment','capita_income','personal_income','Geographic_region')

head(Demographic)

dim(Demographic)
attach(Demographic)

region_1 <- Demographic[Geographic_region == 1,]
region_2 <- Demographic[Geographic_region == 2,]
region_3 <- Demographic[Geographic_region == 3,]
region_4 <- Demographic[Geographic_region == 4,]

n1<-nrow(region_1)
n2<-nrow(region_2)
n3<-nrow(region_3)
n4<-nrow(region_4)

summary(Demographic)

#=========================================================
# log(Y_i) = Beta_0 + Beta_1*below_poverty + Beta_2*capita_income
#=========================================================
#region 1
attach(region_1)
print("Region 1: Pairs and Correlation")
pairs(cbind(log(Y_i), total_population,  below_poverty , unemployment, capita_income, personal_income))
cor(cbind(log(Y_i), total_population,  below_poverty , unemployment, capita_income, personal_income))

print("Region 1: Boxplots")
par(mfrow=c(2,2))
boxplot(cbind(log(Y_i)[Geographic_region == 1], below_poverty[Geographic_region == 1], unemployment[Geographic_region == 1]),names=c("Log(Total serious crimes)","Poverty level ","Unemployment "))
boxplot(cbind( capita_income[Geographic_region == 1], personal_income[Geographic_region == 1]),names=c("Capita Income ","Personal income "))
boxplot(total_population[Geographic_region ==1], xlab ="Total Population")
print("Region 1: Summary statistics")
summary(cbind(log(Y_i), total_population,  below_poverty , unemployment, capita_income, personal_income))
detach(region_1)
attach(Demographic)
#demographic
```

```
print("Demographic: Summary statistics")
summary(cbind(log(Y_i), total_population,  below_poverty , unemployment, capita_income, personal_income))
#===================================
# Frequentest (region 1)
#===================================
model_r_1 <- lm(log(Y_i)~  below_poverty + capita_income , data = region_1)
#summary(lm(log(Y_i)~  below_poverty + capita_income , data = region_2))
#summary(lm(log(Y_i)~  below_poverty + capita_income , data = region_3))
#summary(lm(log(Y_i)~  unemployment + capita_income , data = region_4))

summary(model_r_1)

#diagnostic plots
par(mfrow=c(2,2))
plot(model_r_1)
#====================================
# Bayesian
#====================================
#choose region
attach(region_1)

#================================================
# Bayesian estimation via MCMC (Monte Carlo)
#================================================

# below_poverty + capita_income
n<-length(Y_i)#number of rows for specific region
X<-cbind(rep(1,n),below_poverty , capita_income ) #choose two variables
p<-dim(X)[2] #number of columns
y<-log(Y_i) #continuous  y

#set priors
beta.0<-rep(0,p) ;
Sigma.0<-diag(c(1000,1000,1000)^2,p)
nu.0<-1 ;
sigma2.0<- 1

S<-5000 #5000 samples

rmvnorm<-function(n,mu,Sigma)
{ # samples from the multivariate normal distribution
  E<-matrix(rnorm(n*length(mu)),n,length(mu))
  t(  t(E%*%chol(Sigma)) +c(mu))
}

iSigma.0<-solve(Sigma.0) #initialize
XtX<-t(X)%*%X #X^2?

## store mcmc samples in these objects
beta.post<-matrix(nrow=S,ncol=p)
sigma2.post<-rep(NA,S)

## starting value
set.seed(1)
sigma2<- var(residuals(lm(y~0+X))) #initialize

## MCMC algorithm
```

```
for( scan in 1:S) {
 #update beta
 V.beta<- solve(  iSigma.0 + XtX/sigma2 ) #posterior variance
 E.beta<- V.beta%*%( iSigma.0%*%beta.0 + t(X)%*%y/sigma2 ) #posterior mean
 beta<-t(rmvnorm(1, E.beta,V.beta) ) #samples MVN with posterior

 #update sigma2
 nu.n<- nu.0+n #shape = 1+n /2
 ss.n<-nu.0*sigma2.0 + sum((y-X%*%beta)^2) #rate = (1*1^2 + RSS/err or) /2
 sigma2<-1/rgamma(1,nu.n/2, ss.n/2)  #inverse gamma

 #save results of this scan
 beta.post[scan,]<-beta
 sigma2.post[scan]<-sigma2
}
#
library(coda)
# Convert to mcmc objects for analysis
beta_samples_mcmc <- mcmc(beta.post)

#95% confidence intervals from frequentest model
ci<- confint(model_r_1, level= 0.95)
print("95% Confidence Interval:")
print(ci)

# 95% credible intervals (also shown above in summary)
credintervals <- t(apply(beta.post, 2, quantile, probs = c(0.025, 0.975)))
rownames(credintervals)<- cbind("(Intercept)", "below_poverty", "capita_income")
print("95% Credible interval")
print(credintervals)
# bayes approch
# Compute the mean of the posterior samples
beta_mean <- apply(beta.post, 2, mean)

# Compute the standard deviation of the posterior samples
beta_sd <- apply(beta.post, 2, sd)

# Calculate the t-statistic for each beta_j
t_stat <- beta_mean / beta_sd

# Compute the p-value for the two-sided test
p_values <- 2 * (1 - pnorm(abs(t_stat)))

# Print the p-values
print(p_values)

# Summary
summary(beta.post)
summary(beta_samples_mcmc)

par(mfrow=c(2,2))
#histograms
hist(beta.post[,1],xlab=expression(beta[0]),ylab="",main="")
abline(v=model_r_1$coefficients[1],col=2,lwd=2)

hist(beta.post[,2],xlab=expression(beta[1]),ylab="",main="")
abline(v=model_r_1$coefficients[2],col=2,lwd=2)
```

```
hist(beta.post[,3],xlab=expression(beta[2]),ylab="",main="")
abline(v=model_r_1$coefficients[3],col=2,lwd=2)

hist(sigma2.post,xlab=expression(sigma^2),ylab="",main="")
abline(v=summary(model_r_1)$sigma^2 ,col=2,lwd=2)



###diagnostics
par(mfrow = c(1, 1))
plot(beta.post[,2], beta.post[,3])

# Compute residuals for the fitted model
residuals <- (log(Y_i) - (beta.post %*% t(X))) / sqrt(sigma2.post)

#mcmc built in diagnostics
par(mfrow=c(2,2))
plot(beta_samples_mcmc)

#par(mfrow=c(3,3))

# Autocorrelation function
par(mfrow=c(1,3))
acf(beta.post[,1],main="B_0", xlab=expression(theta))
acf(beta.post[,2],main="B_1", xlab=expression(tilde(sigma)^2))
acf(beta.post[,3],main="B_2", xlab=expression(tilde(sigma[0])^2))

# Ergodic mean
library(dlm)
par(mfrow=c(1,3))
plot(ergMean(beta.post[,1]),main="B_0", ylab=expression(theta),xlab="MCMC Samples",type="l")
plot(ergMean(beta.post[,2]),main="B_1", ylab=expression(tilde(sigma)^2),xlab="MCMC Samples",type="l")
plot(ergMean(beta.post[,3]),main="B_2", ylab=expression(tilde(sigma[0])^2),xlab="MCMC Samples",type="l")

# Mixing ?
par(mfrow=c(1,3))
plot(beta.post[,1],main="B_0", ylab=expression(theta),xlab="MCMC Samples",type="l")
plot(beta.post[,2],main="B_1", ylab=expression(tilde(sigma)^2),xlab="MCMC Samples",type="l")
plot(beta.post[,3],main="B_2", ylab=expression(tilde(sigma[0])^2),xlab="MCMC Samples",type="l")
gibbs_sampler_with_gprior <- function(y, X, g_init = 1, nu0 = 1, s20 = 1, S = 1000) {
 n <- nrow(X)
 p <- ncol(X)

 # Initialize storage matrices
 beta_post <- matrix(NA, nrow = S, ncol = p)
 sigma2_post <- numeric(S)
 g_post <- numeric(S)

 # Initial values
 g <- g_init
 sigma2 <- var(residuals(lm(y ~ 0 + X)))

 # Inverse-gamma parameters
 nu_n <- nu0 + n
 iXX <- solve(t(X) %*% X)
```

```
  for (s in 1:S) {
    # Update beta
    V_beta <- solve(g / (g + 1) * t(X) %*% X)
    E_beta <- V_beta %*% (g / (g + 1) * t(X) %*% y)
    beta <- t(rmvnorm(1, E_beta, V_beta))

    # Update sigma2
    ss_n <- nu0 * s20 + sum((y - X %*% beta)^2)
    sigma2 <- 1 / rgamma(1, nu_n / 2, ss_n / 2)

    # Update g (using Metropolis-Hastings)
    g_proposal <- rgamma(1, shape = 2, rate = 1) # example proposal distribution
    log_acceptance_ratio <- -2 * log(1 + g_proposal) + ( -sum((y - X %*% beta)^2) / (2 * sigma2) ) - (-2 * log(1 + g) + (-sum((y - X
%*% beta)^2) / (2 * sigma2)))
    if (log(runif(1)) < log_acceptance_ratio) {
      g <- g_proposal
    }

    # Store samples
    beta_post[s, ] <- beta
    sigma2_post[s] <- sigma2
    g_post[s] <- g
  }

  list(beta = beta_post, sigma2 = sigma2_post, g = g_post)
}

set.seed(1)
gibbs_results <- gibbs_sampler_with_gprior(log(Y_i), cbind(1, below_poverty, capita_income), S = 1000)

# Summary of the results
beta_post <- gibbs_results$beta
sigma2_post <- gibbs_results$sigma2
g_post <- gibbs_results$g

# Plot the posterior densities
par(mfrow = c(2, 2))
plot(density(beta_post[, 1]), main = expression(beta[0]), xlab = expression(beta[0]))
abline(v = mean(beta_post[, 1]), col = "red")
plot(density(beta_post[, 2]), main = expression(beta[1]), xlab = expression(beta[1]))
abline(v = mean(beta_post[, 2]), col = "red")
plot(density(beta_post[, 3]), main = expression(beta[2]), xlab = expression(beta[2]))
abline(v = mean(beta_post[, 3]), col = "red")
plot(density(g_post), main = expression(g), xlab = expression(g))
abline(v = mean(g_post), col = "red")

# Frequentist results for comparison
freq_model <- lm(log(Y_i) ~ below_poverty + capita_income)
summary(freq_model)
confint(freq_model)
```