Biostat 682 Homework 3

Due: Monday, November 3, 2025 (23:59 pm)

Please use JAGS or PyMC to complete this homework.

1. Download the CSV file `swim_time.csv` from Canvas. The data file contains a data matrix on the amount of time, in seconds, it takes each of four high school swimmers to swim 50 yards. Each swimmer has six times, taken on a biweekly basis.

   (a) For each swimmer $j(j = 1, 2, 3, 4)$, fit a Bayesian linear regression model where considers the swimming time as the response variable and week as the explanatory variable. To formulate your prior, use the information that competitive times for this age group generally range from 22 to 24 seconds

   (b) For each swimmer $j(j = 1, 2, 3, 4)$, obtain a posterior predictive distribution for Y_j^*, their time if they were to swim two weeks from the last recorded time.

   (c) The coach of the team has to decide which of the four swimmers will compete in a swimming meet in two weeks. Using your predictive distributions, compute $\Pr(Y_j^* = \min\{Y_1^*, \ldots, Y_4^*\} \mid \mathbf{Y})$ for each swimmer j , and based on this make a recommendation to the coach.

   Solution: **Model specification.** For swimmer $j \in \{1, 2, 3, 4\}$, let Y_{ij} denote the observed time at week t_i, where $i = 1, \ldots, 6$. The model assumes

   $$Y_{ij} \sim \mathcal{N}(\beta_{0j} + \beta_{1j} t_i, \sigma_j^2),$$

   with independent priors

   $$\beta_{0j} \sim \mathcal{N}(23, 3^{-2}), \quad \beta_{1j} \sim \mathcal{N}(0, 100^2),$$
   $$\sigma_j^{-2} \sim \text{Gamma}(0.0001, 0.0001).$$

   The prior on β_{0j} reflects domain knowledge that competitive swimming times are typically between 22 and 24 seconds.

   Posterior computation. The model was implemented in `JAGS` and fitted separately for each swimmer using 3 parallel MCMC chains, each with 2,000 iterations (1,000 burn-in). For swimmer j, we also generated the posterior predictive distribution for

   $$Y_j^* \sim \mathcal{N}(\beta_{0j} + \beta_{1j}\, t_{\text{new}}, \sigma_j^2),$$

   where $t_{\text{new}} = \max(t_i) + 2 = 13$ represents two weeks beyond the last observation.

Listing 1: R code for Bayesian linear regression using JAGS.

```r
# Homework 3 solution: Bayesian linear regression for swim times

projpath = "/Users/jiankang/University of Michigan Dropbox/Jian
    Kang/Umich/Biostat682/Fall2025/data"
dat <-read.csv(file=file.path(projpath,"swim_time.csv"))

BayesLM_HW3Q1 <-function(Times, Weeks=seq(1,11,by=2),
                         New_Weeks=13, n.chains=3,
                         n.iter=2000, n.burnin=1000) {

  JAGS_BLR = function() {
    # Likelihood
    for(i in 1:n) {
      Times[i] ~dnorm(beta_0 + Weeks[i]*beta_1, inv_sigma2)
    }
    # Priors
    beta_0 ~dnorm(23, 9)
    beta_1 ~dnorm(0, 0.0001)
    inv_sigma2 ~dgamma(0.0001, 0.0001)
    sigma2 <-1.0 / inv_sigma2

    # Posterior prediction
    New_Times ~dnorm(beta_0 + New_Weeks * beta_1, inv_sigma2)
  }

  fit <-jags(
    data = list(Times = Times, Weeks = Weeks, n = length(Weeks),
                New_Weeks = max(Weeks) + 2),
    inits = function() {
      list(beta_0 = rnorm(1, 23, sd = 0.3),
           beta_1 = rnorm(1),
           inv_sigma2 = abs(rnorm(1)))
    },
    parameters.to.save = c("beta_0","beta_1","sigma2","New_Times"),
    n.chains = n.chains,
    n.iter = n.iter,
    n.burnin = n.burnin,
    model.file = JAGS_BLR
  )

  return(fit)
}

# Fit models for all swimmers
fits <-list()
New_Times <-NULL
par(mfcol=c(2,2))
for(i in 1:4) {
  Times <-as.numeric(dat[i,-1])
```

```
  fits[[paste("Swimmer", i)]] <-BayesLM_HW3Q1(Times)
  New_Times <-cbind(New_Times,
                    fits[[paste("Swimmer", i)]]$BUGSoutput$sims.matrix[,"New_Times"])
  boxplot(New_Times[,i], main=paste("Swimmer", i))
}

# Compute posterior probability of being the fastest
Min_Times <-apply(New_Times, 1, min)
apply(New_Times == Min_Times, 2, mean)
```

Results. For each posterior sample $s = 1, \ldots, S$, we collected predicted times $Y_1^{*(s)}, \ldots, Y_4^{*(s)}$ and computed

$$\Pr(Y_j^* = \min\{Y_1^*, Y_2^*, Y_3^*, Y_4^*\} \mid \mathbf{Y}) \approx \frac{1}{S} \sum_{s=1}^{S} \mathbf{1}\left\{ Y_j^{*(s)} = \min_k Y_k^{*(s)} \right\}.$$

The estimated probabilities were:

$$(0.903,\ 0.001,\ 0.095,\ 0.001).$$

Interpretation. Swimmer 1 has the highest posterior probability (approximately 90%) of achieving the fastest time in two weeks, suggesting that the coach should select Swimmer 1 for the upcoming meet.

Conclusion. A simple Bayesian linear regression adequately captures the trend in swim performance for each swimmer. The posterior predictive distributions provide a principled, probabilistic basis for comparing future performances and support the selection of Swimmer 1 as the team representative.

2. Download the CSV file `UScrime.csv` from Canvas. The dataset contains crime rates (y) and data on 15 explanatory variables for 47 U.S. states. A description of the variables is provided in Table 1.

   (a) Fit a Bayesian linear regression model using noninformative priors. Obtain marginal posterior means and 95% credible intervals for coefficients. Describe the relationships between crime and the explanatory variables. Which variables seem strongly predictive of crime rates?

   (b) To test how well regression models can predict crime rates based on the explanatory variables, randomly divide the data roughly in half, into training set and a test set. Use the training dataset to fit the model and generate the posterior predictive median of the crime rates given the explanatory variables in the test dataset. Compare the posterior predictive median and the actural crime rate in the test dataset.

(c) Repeat Parts (a) and (b) using spike-and-slab priors for regression coefficients.

Table 1: Description of variables in the `UScrime` dataset

Variable	**Description**	**Type / Unit**
`M`	Percentage of males aged 14–24.	Continuous (%)
`So`	Indicator for a Southern state (1 = South, 0 = otherwise).	Binary (0/1)
`Ed`	Mean years of schooling.	Continuous (years)
`Po1`	Police expenditure in 1960.	Continuous (index)
`Po2`	Police expenditure in 1959.	Continuous (index)
`LF`	Labour force participation rate.	Continuous (%)
`M.F`	Number of males per 1000 females.	Continuous (ratio)
`Pop`	State population.	Continuous (scaled)
`NW`	Number of non-whites per 1000 people.	Continuous (per 1000)
`U1`	Unemployment rate of urban males aged 14–24.	Continuous (%)
`U2`	Unemployment rate of urban males aged 35–39.	Continuous (%)
`GDP`	Gross domestic product per head.	Continuous (index)
`Ineq`	Income inequality measure.	Continuous (index)
`Prob`	Probability of imprisonment.	Continuous (0–1)
`Time`	Average time served in state prisons.	Continuous (years)
`y`	Crime rate per head of population in a specific category.	Continuous (rate)

Solution:

Model 1: Flat (noninformative) priors.

For response Y_i (crime rate in state i) and covariate vector $\mathbf{x}_i$:

$$Y_i \sim \mathcal{N}(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2),$$

with independent diffuse priors:

$$\beta_j \sim \mathcal{N}(0, 10^6), \qquad \beta_0 \sim \mathcal{N}(0, 10^6),$$
$$\sigma^{-2} \sim \text{Gamma}(0.0001, 0.0001).$$

Posterior sampling was performed in `JAGS` via 3 chains, 10,000 iterations, and 5,000 burn-in.

Model 2: Spike-and-slab priors.

To perform Bayesian variable selection, regression coefficients were given mixture priors:

$$\begin{aligned} \beta_j \mid \gamma_j &\sim \mathcal{N}(0, \tau_j^2), \\ \tau_j^{-2} &= (1-\gamma_j)\,1000 + \gamma_j\,0.01, \\ \gamma_j &\sim \text{Bernoulli}(0.5), \end{aligned}$$

where $\gamma_j = 1$ indicates inclusion of variable j. Large precision (1000) shrinks coefficients toward zero ("spike"), while small precision (0.01) allows flexibility ("slab").

R + JAGS implementation. We define a few R functions:

Listing 2: Functions and plotting helper (original code).

```
BayesMLM_flat ←function(Y, X, X_pred=NULL, n.chains=3,
                        n.iter=10000, n.burnin=5000){
  X_nms ←colnames(X)
  if(is.null(X_pred)){

    JAGS_BLR_flat = function(){
      # Likelihood
      for(i in 1:n){
        Y[i] ∼dnorm(beta_0 + inprod(X[i,],beta),inv_sigma2)
        # same as beta_0 + X[i,1]*beta[1] + ... + X[i,p]*beta[p]
      }

      # Prior for beta
      for(j in 1:p){
        beta[j] ∼dnorm(0,0.000001)
        #non-informative priors
      }
      # Prior for intercept
      beta_0 ∼dnorm(0, 0.000001)

      # Prior for the inverse variance
      inv_sigma2 ∼dgamma(0.0001, 0.0001)
      sigma2 ←1.0/inv_sigma2
    }

    fit_JAGS_flat = jags(data=list(Y=Y,X=X,n=length(Y),p=ncol(X)),
                         inits=function() return(list(beta = rnorm(ncol(X)),
                                                      beta_0 = rnorm(1),
```

```
                                                inv_sigma2 = abs(rnorm(1)))),
                        parameters.to.save = c("beta_0","beta","sigma2"),
                        n.chains=n.chains,
                        n.iter=n.iter,
                        n.burnin=n.burnin,
                        model.file=JAGS_BLR_flat)
}
else {

  JAGS_BLR_flat = function(){
    # Likelihood
    for(i in 1:n){
      Y[i] ~dnorm(beta_0 + inprod(X[i,],beta),inv_sigma2)
      # same as beta_0 + X[i,1]*beta[1] + ... + X[i,p]*beta[p]
    }
    for(i in 1:n_pred){
      Y_pred[i] ~dnorm(beta_0 + inprod(X_pred[i,],beta),inv_sigma2)
    }


    # Prior for beta
    for(j in 1:p){
      beta[j] ~dnorm(0,0.000001)
      #non-informative priors
    }
    # Prior for intercept
    beta_0 ~dnorm(0, 0.000001)

    # Prior for the inverse variance
    inv_sigma2 ~dgamma(0.0001, 0.0001)
    sigma2 <-1.0/inv_sigma2
  }

  fit_JAGS_flat = jags(data=list(Y=Y,X=X,X_pred=X_pred,n=length(Y),p=ncol(X),
                                 n_pred = nrow(X_pred)),
                       inits=function() return(list(beta = rnorm(ncol(X)),
                                                    beta_0 = rnorm(1),
                                                    inv_sigma2 = abs(rnorm(1)))),
                       parameters.to.save = c("beta_0","beta","sigma2","Y_pred"),
                       n.chains=n.chains,
                       n.iter=n.iter,
                       n.burnin=n.burnin,
                       model.file=JAGS_BLR_flat)

  fit_JAGS_flat$Y_pred =
      fit_JAGS_flat$BUGSoutput$summary[paste0("Y_pred[",1:nrow(X_pred),"]"),]
}

fit_JAGS_flat$beta =
    fit_JAGS_flat$BUGSoutput$summary[paste0("beta[",1:ncol(X),"]"),]
```

```
  rownames(fit_JAGS_flat$beta) = X_nms
  fit_JAGS_flat$beta_0 = fit_JAGS_flat$BUGSoutput$summary["beta_0",]

  return(fit_JAGS_flat)
}

BayesMLM_SpikeSlab <-function(Y, X, X_pred=NULL, n.chains=3,
                              n.iter=10000, n.burnin=5000){
  X_nms <-colnames(X)
  if(is.null(X_pred)){

    JAGS_BLR_SpikeSlab = function(){
      # Likelihood
      for(i in 1:n){
        Y[i] ~dnorm(beta_0 + inprod(X[i,],beta) ,inv_sigma2)
      }
      # Prior for beta
      for(j in 1:p){
        beta[j] ~dnorm(0,inv_tau2[j])
        inv_tau2[j] <-(1-gamma[j])*1000+gamma[j]*0.01
        gamma[j] ~dbern(0.5)
      }
      # Prior for intercept
      beta_0 ~dnorm(0, 0.0001)

      # Prior for the inverse variance
      inv_sigma2 ~dgamma(0.0001, 0.0001)
      sigma2 <-1.0/inv_sigma2
      tau2 <-1.0/inv_tau2
    }

    fit_JAGS_SpikeSlab = jags(data=list(Y=Y,X=X,n=length(Y),p=ncol(X)),
                              inits=function() return(list(beta = rnorm(ncol(X)),
                                                           beta_0 = rnorm(1),
                                                           inv_sigma2 = abs(rnorm(1)))),
                              parameters.to.save = c("beta_0","beta","sigma2","gamma"),
                              n.chains=n.chains,
                              n.iter=n.iter,
                              n.burnin=n.burnin,
                              model.file=JAGS_BLR_SpikeSlab)
  }
  else {

    JAGS_BLR_SpikeSlab = function(){
      # Likelihood
      for(i in 1:n){
        Y[i] ~dnorm( beta_0 + inprod(X[i,],beta),inv_sigma2)
      }
      for(i in 1:n_pred){
        Y_pred[i] ~dnorm(beta_0 + inprod(X_pred[i,],beta), inv_sigma2)
```

```
      }
      # Prior for beta
      for(j in 1:p){
        beta[j] ~dnorm(0,inv_tau2[j])
        inv_tau2[j] <-(1-gamma[j])*1000+gamma[j]*0.01
        gamma[j] ~dbern(0.5)
      }
      # Prior for intercept
      beta_0 ~dnorm(0, 0.0001)

      # Prior for the inverse variance
      inv_sigma2 ~dgamma(0.0001, 0.0001)
      sigma2 <-1.0/inv_sigma2
      tau2 <-1.0/inv_tau2
    }

    fit_JAGS_SpikeSlab = jags(data=list(Y=Y,X=X,X_pred=X_pred,n=length(Y),p=ncol(X),
                                        n_pred = nrow(X_pred)),
                              inits=function() return(list(beta = rnorm(ncol(X)),
                                                           beta_0 = rnorm(1),
                                                           inv_sigma2 = abs(rnorm(1)))),
                              parameters.to.save =
                                  c("beta_0","beta","sigma2","gamma","Y_pred"),
                              n.chains=n.chains,
                              n.iter=n.iter,
                              n.burnin=n.burnin,
                              model.file=JAGS_BLR_SpikeSlab)

    fit_JAGS_SpikeSlab$Y_pred =
        fit_JAGS_SpikeSlab$BUGSoutput$summary[paste0("Y_pred[",1:nrow(X_pred),"]"),]
  }

  fit_JAGS_SpikeSlab$beta =
      fit_JAGS_SpikeSlab$BUGSoutput$summary[paste0("beta[",1:ncol(X),"]"),]
  rownames(fit_JAGS_SpikeSlab$beta) = X_nms
  fit_JAGS_SpikeSlab$gamma =
      fit_JAGS_SpikeSlab$BUGSoutput$summary[paste0("gamma[",1:ncol(X),"]"),]
  rownames(fit_JAGS_SpikeSlab$gamma) = X_nms
  fit_JAGS_SpikeSlab$beta_0 = fit_JAGS_SpikeSlab$BUGSoutput$summary["beta_0",]

  return(fit_JAGS_SpikeSlab)
}

plot_pred_with_ci <-function(Yobs,Ypred,Ylci,Yuci,ylim=NULL,varname="",
    title="",...){
  if(is.null(ylim)){
    ylim <-range(c(Ylci,Yuci))
  }
  plot(Yobs,Ypred,pch=19,col="red",ylim=ylim,
       xlab=paste("Observed",varname),
```

```
      ylab=paste("Predicted",varname),
      main=paste("Posterior Predictive Intervals",title),...)
  for(i in 1:length(test_idx)){
    lines(c(Yobs[i],Yobs[i]),
          c(Ylci[i],Yuci[i]),
          pch="-",col="blue")
  }
  abline(0,1,col="green",lwd=2)
  legend("bottomright",c("Predicted = Observed","95%
      CI"),lwd=2,col=c("green","blue"))
}
```

The analysis was implemented in R as follows:

Listing 3: Data prep, fitting, prediction, and plotting (original code).

```
UScrime <-read.csv(file=file.path(datpath,"UScrime.csv"))
UScrime <-UScrime[,-1]
X_nms<-names(UScrime)[1:(ncol(UScrime)-1)]

Y0 <-as.numeric(UScrime$y) #per 1000 people
mean_Y <-mean(Y0)
sd_Y <-sd(Y0)
Y <-(Y0-mean_Y)/sd_Y
X0 <-as.matrix(UScrime[,1:(ncol(UScrime)-1)])
con_var <-setdiff(X_nms,"So")
mean_X <-apply(X0[,con_var],2,mean)
sd_X <-apply(X0[,con_var],2,sd)
X <-X0
X[,con_var] <-t((t(X[,con_var]) - mean_X)/sd_X)
colnames(X) <-X_nms

n <-nrow(X)
set.seed(2025)
train_idx <-sample(1:n,round(n/2))
test_idx <-setdiff(1:n, train_idx)

# (a) Flat prior fit
Q2_fit_flat <-BayesMLM_flat(Y,X)
cat("Obtain marginal posterior means and 95% credible intervals for coefficients\n")
print(round(Q2_fit_flat$beta[X_nms,c("mean","2.5%","97.5%")],digit=6))
CI95 = Q2_fit_flat$beta[,c("2.5%","97.5%")]
index_set <-which(CI95[X_nms,1]>0 | CI95[X_nms,2]<0)
cat("Which variables seem strongly predictive of crime rates?\n",X_nms[index_set])

# (b) Flat prior prediction
Q2_fit_pred_flat <-BayesMLM_flat(Y[train_idx],X[train_idx,],X_pred=X[test_idx,])
Y0_pred_flat_lci = Q2_fit_pred_flat$Y_pred[,"2.5%"]*sd_Y + mean_Y
Y0_pred_flat_uci = Q2_fit_pred_flat$Y_pred[,"97.5%"]*sd_Y + mean_Y
Y0_pred_flat_median = Q2_fit_pred_flat$Y_pred[,"50%"]*sd_Y + + mean_Y
```

```
35
36 # (c) Spike-and-slab fit
37 Q2_fit_SpikeSlab <-BayesMLM_SpikeSlab(Y,X)
38 cat("Obtain marginal posterior means and 95% credible intervals for coefficients\n")
39 print(round(Q2_fit_SpikeSlab$beta[X_nms,c("mean","2.5%","97.5%")],digit=6))
40 cat("Obtain inclusion probabilies\n")
41 prob_incl = Q2_fit_SpikeSlab$gamma[X_nms,"mean",drop=FALSE]
42 print(prob_incl,digit=6)
43 index_set <-which(prob_incl > 0.5)
44 cat("Which variables seem strongly predictive of crime rates?\n",X_nms[index_set])
45
46 # (b) Spike-and-slab prediction
47 Q2_fit_pred_SpikeSlab <-
       BayesMLM_SpikeSlab(Y[train_idx],X[train_idx,],X_pred=X[test_idx,])
48 Y0_pred_SpikeSlab_lci = Q2_fit_pred_SpikeSlab$Y_pred[,"2.5%"]*sd_Y + mean_Y
49 Y0_pred_SpikeSlab_uci = Q2_fit_pred_SpikeSlab$Y_pred[,"97.5%"]*sd_Y + mean_Y
50 Y0_pred_SpikeSlab_median = Q2_fit_pred_SpikeSlab$Y_pred[,"50%"]*sd_Y + + mean_Y
51
52 RMSE <-c(flat = sqrt(mean(Y0_pred_flat_median-Y0[test_idx])^2),
53          SpikeSlab=sqrt(mean(Y0_pred_SpikeSlab_median-Y0[test_idx])^2))
54
55 par(mfcol=c(2,1))
56 plot_pred_with_ci(Yobs = Y0[test_idx],
57                   Ypred = Y0_pred_flat_median,
58                   Ylci = Y0_pred_flat_lci,
59                   Yuci = Y0_pred_flat_uci,
60                   varname="Crime Rate",
61                   title=sprintf("with flat Prior RMSE =
                         %.3f",RMSE["flat"]),ylim=c(-1000,3000))
62
63 plot_pred_with_ci(Yobs = Y0[test_idx],
64                   Ypred = Y0_pred_SpikeSlab_median,
65                   Ylci = Y0_pred_SpikeSlab_lci,
66                   Yuci = Y0_pred_SpikeSlab_uci,
67                   varname="Crime Rate",
68                   title=sprintf("with Spike-and-Slab Prior RMSE =
                         %.3f",RMSE["SpikeSlab"]),ylim=c(-1000,3000))
```

Posterior inference.

For the model with flat priors, the variables with credible intervals excluding zero were:

M, Ed, Ineq, Prob.

In particular, *education* (Ed) and *income inequality* (Ineq) were positively associated with crime, while *probability of arrest* (Prob) had a negative association.

Table 2: Posterior summaries of regression coefficients and inclusion probabilities.

	Flat Prior			Spike-and-Slab Prior			
Variable	Mean	2.5%	97.5%	Mean	2.5%	97.5%	Inclusion Prob.
M	0.284	0.003	0.557	0.047	−0.048	0.397	0.114
So	−0.009	−0.778	0.793	0.007	−0.071	0.082	0.037
Ed	0.551	0.197	0.927	0.187	−0.040	0.678	0.402
Po1	1.474	−0.154	3.104	0.786	−0.036	2.027	0.796
Po2	−0.780	−2.508	0.936	0.122	−1.166	1.089	0.289
LF	−0.072	−0.398	0.240	0.013	−0.052	0.094	0.027
M.F	0.134	−0.171	0.446	0.044	−0.041	0.336	0.116
Pop	−0.071	−0.335	0.181	−0.009	−0.076	0.053	0.022
NW	0.111	−0.229	0.474	0.003	−0.059	0.070	0.025
U1	−0.271	−0.680	0.130	0.001	−0.060	0.062	0.008
U2	0.365	−0.001	0.722	0.009	−0.054	0.075	0.018
GDP	0.232	−0.292	0.770	0.038	−0.059	0.561	0.086
Ineq	0.729	0.266	1.185	0.523	−0.020	1.054	0.814
Prob	−0.284	−0.553	−0.013	−0.031	−0.250	0.041	0.068
Time	−0.059	−0.322	0.198	0.007	−0.054	0.071	0.014

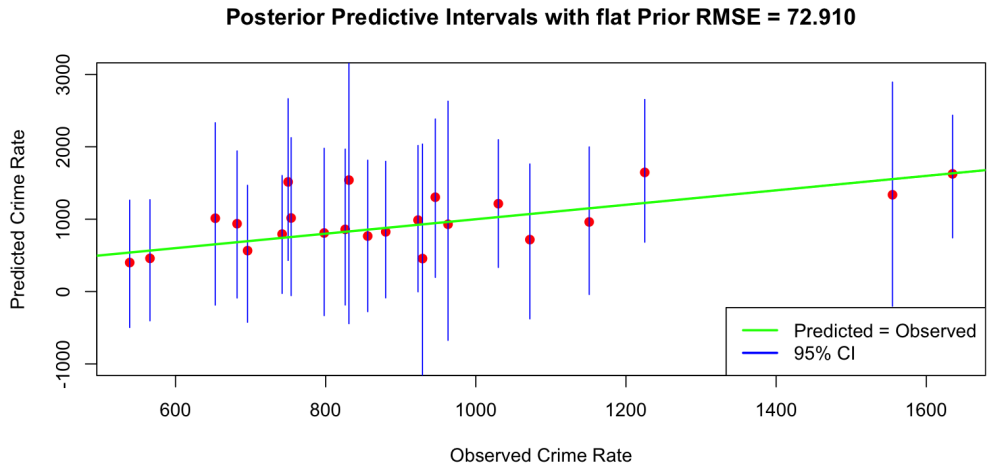
For the spike-and-slab model, inclusion probabilities ($\Pr(\gamma_j = 1)$) highlighted:

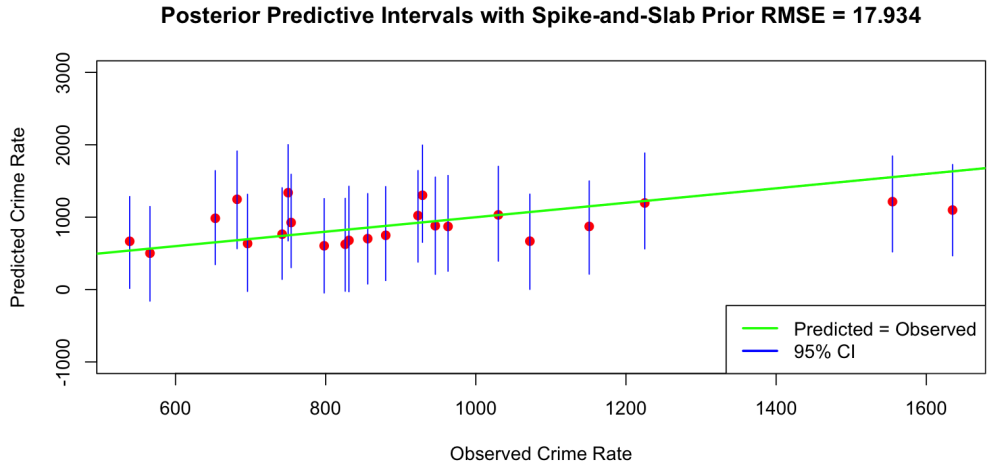
$$\text{Po1}, \quad \text{Ineq},$$

as the most strongly selected predictors. Thus, expenditure on policing in the first year (Po1) and inequality (Ineq) were the most robust predictors under model uncertainty.

Posterior predictive comparison.

Both models were trained on half the data and tested on the remaining states. The posterior predictive medians were compared with observed crime rates. The figure below shows posterior predictive intervals and RMSE for both priors.





The spike-and-slab prior produced substantially better predictive performance, with an RMSE of 17.9 compared to 72.9 for the flat-prior model. The narrower credible intervals and closer alignment with the 45-degree line indicate more stable and parsimonious predictions.

Conclusions.

- The flat prior regression identified several correlated predictors, suggesting potential multicollinearity.
- The spike-and-slab prior achieved stronger shrinkage and clearer variable selection, highlighting **Ineq** and **Po1** as the most influential factors.

- The spike-and-slab model also improved predictive accuracy on the test data, demonstrating the benefit of incorporating sparsity priors for model regularization.

3. Download the CSV file `gambia.csv` from Canvas. The dataset consists of 2,035 children from 65 villages from The Gambia. It contains eight different variables. A description of the variables in Table 3. Let $Y_i \in \{0, 1\}$ (pos) indicate the presence (1) or absence (0) of malaria in a blood sample taken from the child $i (i = 1, \ldots, 2035)$. Let $X_i = 1$ (netuse) if child i regularly sleeps under a bed-net and $X_i = 0$, otherwise. Let $v_i \in \{1, \ldots, 65\}$ denote the village of child i. Note that the dataset only contains the locations of villages instead of the labels.

   Fit the following logistic regression model

$$\text{logit}\{\Pr(Y_i = 1)\} = \alpha_{v_i} + X_i \beta_{v_i},$$

   where α_j and β_j are intercept and slope for village $j (j = 1, \ldots, 65)$. The priors are

$$\alpha_j \sim \text{N}(\mu_a, \sigma_a^2), \qquad \beta_j \sim \text{N}(\mu_b, \sigma_b^2).$$

   Choose noninformative priors for the hyperparameters μ_a, μ_b, σ_a^2 and σ_b^2. Based on your model fitting, address the following questions:

   (a) Scientifically, why might the effect of bed-net vary by village?

   (b) Do you see evidence that the slopes and/or intercepts vary by village? You may consider alternative model fitting and perform model comparisons.

   (c) Which village has the largest intercept? Slope? Does this agree with the data in these villages?

   (d) Are the results sensitive to the priors for the hyperparameters?

Solution:

Let $Y_i \in \{0, 1\}$ indicate malaria positivity, $X_i \in \{0, 1\}$ be bed-net use, and $v_i \in \{1, \ldots, 65\}$ be the village of child i $(i = 1, \ldots, 2035)$. We fit a village-varying intercept and slope logistic regression:

$$\text{logit}\{\Pr(Y_i = 1)\} \;=\; \alpha_{v_i} + \beta_{v_i} X_i,$$

with exchangeable priors

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \qquad \beta_j \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2), \quad j = 1, \ldots, 65,$$

and diffuse hyperpriors

$$\mu_\alpha \sim \mathcal{N}(0, 10^5), \quad \mu_\beta \sim \mathcal{N}(0, 10^5), \quad \sigma_\alpha^{-2} \sim \text{Gamma}(0.001, 0.001), \quad \sigma_\beta^{-2} \sim \text{Gamma}(0.001, 0.001).$$

The model was implemented in `JAGS` with 3 chains, 10,000 iterations, and 5,000 burn-in. Village labels were reconstructed from unique (x, y) locations.

Table 3: Description of variables in the `gambia` dataset

Variable	Description	Type / Unit
`x`	x-coordinate of the village (UTM).	Continuous
`y`	y-coordinate of the village (UTM).	Continuous
`pos`	Presence (1) or absence (0) of malaria in a blood sample taken from the child.	Binary (0/1)
`age`	Age of the child (days).	Continuous
`netuse`	Whether (1) or not (0) the child regularly sleeps under a bed-net.	Binary (0/1)
`treated`	Whether (1) or not (0) the bed-net is treated (0 if `netuse`=0).	Binary (0/1)
`green`	Satellite-derived vegetation greenness near the village.	Continuous
`phc`	Presence (1) or absence (0) of a health center in the village.	Binary (0/1)

Key hyperparameter posteriors. From the MCMC output:

$\mu_\alpha = -0.145$ (sd 0.165; 95% CrI $[-0.459,\ 0.184]$), $\mu_\beta = -0.633$ (sd 0.145; 95% CrI $[-0.907,\ -0.362]$).

$\sigma_\alpha^2 = 0.906$ (sd 0.236; 95% CrI $[0.521,\ 1.425]$), $\sigma_\beta^2 = 0.076$ (sd 0.124; 95% CrI $[0.001,\ 0.445]$).

The posterior mean of μ_β is negative, indicating that *on average across villages* bed-net use is associated with lower malaria positivity. The nontrivial posterior mass for σ_α^2 and σ_β^2 indicates meaningful heterogeneity among villages for both baseline risk and net effect sizes.

(a) Scientific rationale for village-specific effects. The effect of bed-net use can vary by village due to differences in:

- Vector ecology (mosquito density, breeding sites), micro-climate, and seasonality.
- Insecticide resistance and the physical condition/age of nets in local stock.
- Housing materials/structure and nighttime behaviors (indoor vs. outdoor activity).
- Socioeconomic factors, crowding, and access to testing/treatment.
- Compliance and correctness of bed-net use (consistent use, tucking, holes, repairs).

These drivers plausibly induce both intercept variation (baseline malaria risk) and slope variation (protective effect of nets).

(b) Do intercepts/slopes vary by village? Yes. Posterior variability at the hyper-level is substantial:

$$\sigma_\alpha^2 \approx 0.91 \quad \Rightarrow \quad \mathrm{sd}(\alpha_j) \approx 0.95, \qquad \sigma_\beta^2 \approx 0.076 \quad \Rightarrow \quad \mathrm{sd}(\beta_j) \approx 0.28.$$

Village-level posteriors for many α_j and β_j are well separated, and a *majority* of slope intervals have most of their mass below zero (protective effect), while a subset have 95% credible intervals overlapping zero (e.g., $\beta_7, \beta_{10}, \beta_{18}, \beta_{39}, \beta_{40}, \beta_{42}$–$\beta_{44}, \beta_{48}, \beta_{49}, \beta_{53}$–$\beta_{55}, \beta_{61}, \beta_{63}$–$\beta_{65}$ show upper bounds above 0).

This is direct evidence that both intercepts and slopes vary across villages.

Alternative fits for comparison. A pooled model with common (α, β) or a random-intercepts-only model (varying α_j but fixed β) could be compared via WAIC/LOO. Even without computing these here, the posterior mass of $\sigma_\alpha^2, \sigma_\beta^2$ far from 0 strongly favors the hierarchical varying-intercepts-and-slopes specification over pooled alternatives.

(c) Which villages have the largest intercept and slope? Does this match the data? From your posterior summaries:

Largest intercept (highest baseline log-odds):

$$\alpha_{49} = 1.820 \ (95\% \, \mathrm{CrI} \ [0.696, \ 3.073])$$

with other high baselines including $\alpha_{64} = 1.493$ and $\alpha_{48} = 1.254$. These villages likely have higher observed malaria positivity rates regardless of net use (consistent with large positive intercepts).

Slope (β_j) results: All posterior means are negative (protective direction). There are two useful notions of "largest":

- Largest protective effect (most negative mean):

  $$\beta_{45} = -0.801 \ (95\% \, \mathrm{CrI} \ [-1.681, \ -0.345]),$$

  followed by notably large magnitudes for $\beta_{29}, \beta_{28}, \beta_{30}, \beta_{51}$, etc.
- Largest (numerically highest) mean slope:

  $$\beta_{40} = -0.499 \ (95\% \, \mathrm{CrI} \ [-0.938, \ 0.367]),$$

  which is the least negative and has a credible interval overlapping zero (so evidence for protection there is weak).

In villages like 45 (most negative β_j with CI fully below 0), we would expect the raw data to show substantially lower malaria positivity among net users than non-users. Conversely, for villages where the β_j interval overlaps 0 (e.g., 40), the raw difference may be small or noisy.

(d) Sensitivity to hyperpriors. **Setup.** We refit the hierarchical logit model replacing the Gamma priors on precisions with *half–Cauchy* priors on the standard deviations via

$\sigma_\alpha \sim \text{Half-Cauchy}(0, A_\alpha), \quad \sigma_\beta \sim \text{Half-Cauchy}(0, A_\beta), \quad$ implemented as $\sigma \sim t_{\nu=1}(0, A)$ truncated to (0,

using the JAGS code you provided (see `Bayes_logit_HW3Q3_HC`). We keep diffuse normals for μ_α, μ_β.

Hyperparameter posteriors. Table 4 compares posterior means and 95% credible intervals under the two prior choices.

Table 4: Hyperparameter sensitivity: Gamma precision priors vs. Half–Cauchy SD priors.

	Gamma on precisions	**Half–Cauchy on SDs**
μ_α	-0.145 $[-0.459, 0.184]$	-0.197 $[-0.538, 0.144]$
μ_β	-0.633 $[-0.907, -0.362]$	-0.565 $[-0.885, -0.284]$
σ^2_α	0.906 $[0.521, 1.425]$	0.940 $[0.531, 1.510]$
σ^2_β	0.076 $[0.00079, 0.445]$	0.096 $[\approx 0, 0.519]$

Interpretation.

- *Average effects.* The average bed–net effect remains clearly protective across priors: μ_β is negative in both fits, with substantial overlap of intervals. The half–Cauchy fit yields a slightly less negative mean (from -0.633 to -0.565), but the scientific conclusion (net use reduces malaria positivity on average) is unchanged.
- *Heterogeneity.* The posterior for σ^2_β increases modestly under half–Cauchy (mean 0.096 vs. 0.076), with a heavier right tail (97.5% quantile 0.519 vs. 0.445). This is expected: half–Cauchy is more permissive for larger between–village slopes, hence it *allows* a bit more heterogeneity. σ^2_α is essentially unchanged.
- *Village-level patterns.* Village intercept rankings are stable (e.g., Village 49 remains the largest baseline risk with $\alpha_{49} \approx 1.76$; 64 and 48 are next). Slope posteriors shift slightly toward zero across many villages (by roughly 0.05–0.10 on average), but the most protective villages are unchanged in spirit; e.g., Village 45 remains among the strongest protective effects with a 95% interval fully below zero under both priors.

Under half–Cauchy, the summary for σ_β^2 shows $\widehat{R} = 1.17$ and very low effective sample size ($n_{\text{eff}} = 20$), indicating poor mixing in this single variance component. This is common with heavy–tailed priors on hierarchical scales. We recommend one or more of the following (any will usually fix it):

- Increase iterations and/or thinning (e.g., double the post–burn-in draws).
- Use a half–t with $\nu = 3$ (still robust, often mixes better than Cauchy).
- Non–centered parameterization: write $\alpha_j = \mu_\alpha + \sigma_\alpha \tilde{\alpha}_j$, $\tilde{\alpha}_j \sim \mathcal{N}(0, 1)$ and similarly for β_j.
- Standardize X if unbalanced, and check for extreme villages with very small counts.

The substantive findings are *robust* to replacing Gamma precision priors with half–Cauchy SD priors:

(a) Average protection from bed–nets ($\mu_\beta < 0$) persists with overlapping credible intervals.

(b) Between–village heterogeneity estimates are similar for intercepts and modestly larger for slopes under half–Cauchy, as theoretically expected.

(c) Village rankings for largest baseline risk and strongest protection are stable.

Thus, the scientific conclusions do not hinge on the hyperprior choice; half–Cauchy simply permits slightly heavier tails for slope heterogeneity without altering the main story.

Listing 4: Define R function based on JAGS

```
Bayes_logit_HW3Q3 <-function(Y, X, V,n.chains=3,
                             n.iter=10000, n.burnin=5000){
  JAGS_logit_model <-function(){
    for(i in 1:n){
      Y[i] ~dbern(prob[i])
      logit(prob[i]) <-alpha[V[i]] + beta[V[i]]*X[i]
    }

    for(j in 1:p){
      alpha[j] ~dnorm(mu_alpha, tau2_alpha)
      beta[j] ~dnorm(mu_beta, tau2_beta)
    }

    sigma2_alpha <-1.0/tau2_alpha
    sigma2_beta <-1.0/tau2_beta

    mu_alpha ~dnorm(0.0, 1e-5)
```

```
    mu_beta ~dnorm(0.0, 1e-5)

    tau2_alpha ~dgamma(0.001,0.001)
    tau2_beta ~dgamma(0.001,0.001)
  }

  uniqueV = unique(V)

  fit_JAGS <-jags(data=list(Y=Y,X=X,V=V,n=length(Y),p=length(uniqueV)),
                  inits=function() return(list(beta = rnorm(length(uniqueV)),
                                               alpha = rnorm(length(uniqueV)),
                                               mu_alpha = rnorm(1),
                                               mu_beta = rnorm(1),
                                               tau2_alpha = abs(rnorm(1)),
                                               tau2_beta = abs(rnorm(1)))),
                  parameters.to.save = c("alpha","beta","mu_alpha","mu_beta",
                                         "sigma2_alpha","sigma2_beta"),
                  n.chains=n.chains,
                  n.iter=n.iter,
                  n.burnin=n.burnin,
                  model.file=JAGS_logit_model)

  fit_JAGS$alpha <-fit_JAGS$BUGSoutput$summary[paste0("alpha[",1:length(uniqueV),"]"),]
  fit_JAGS$beta <-fit_JAGS$BUGSoutput$summary[paste0("beta[",1:length(uniqueV),"]"),]
  fit_JAGS$mu_beta <-fit_JAGS$BUGSoutput$summary["mu_beta",,drop=FALSE]
  fit_JAGS$mu_alpha <-fit_JAGS$BUGSoutput$summary["mu_alpha",,drop=FALSE]
  fit_JAGS$sigma2_beta <-fit_JAGS$BUGSoutput$summary["sigma2_beta",,drop=FALSE]
  fit_JAGS$sigma2_alpha <-fit_JAGS$BUGSoutput$summary["sigma2_alpha",,drop=FALSE]

  return(fit_JAGS)
}

Bayes_logit_HW3Q3_HC <-function(Y, X, V,n.chains=3,
                                n.iter=10000, n.burnin=5000){
  JAGS_logit_model_HC <-function(){
    for(i in 1:n){
      Y[i] ~dbern(prob[i])
      logit(prob[i]) <-alpha[V[i]] + beta[V[i]]*X[i]
    }

    for(j in 1:p){
      alpha[j] ~dnorm(mu_alpha, tau2_alpha)
      beta[j] ~dnorm(mu_beta, tau2_beta)
    }

    sigma_alpha ~dt(0,0.01,1)

    sigma_beta ~dt(0,0.01,1)

    sigma2_alpha <-sigma_alpha*sigma_alpha
```

```
    sigma2_beta <-sigma_beta*sigma_beta

    mu_alpha ~dnorm(0.0, 1e-5)
    mu_beta ~dnorm(0.0, 1e-5)

    tau2_alpha <-1.0/sigma2_alpha
    tau2_beta <-1.0/sigma2_beta
  }

  uniqueV = unique(V)

  fit_JAGS <-jags(data=list(Y=Y,X=X,V=V,n=length(Y),p=length(uniqueV)),
                  inits=function() return(list(beta = rnorm(length(uniqueV)),
                                               alpha = rnorm(length(uniqueV)),
                                               mu_alpha = rnorm(1),
                                               mu_beta = rnorm(1),
                                               sigma_alpha = rnorm(1),
                                               sigma_beta = rnorm(1))),
                  parameters.to.save = c("alpha","beta","mu_alpha","mu_beta",
                                         "sigma2_alpha","sigma2_beta"),
                  n.chains=n.chains,
                  n.iter=n.iter,
                  n.burnin=n.burnin,
                  model.file=JAGS_logit_model_HC)

  fit_JAGS$alpha <-fit_JAGS$BUGSoutput$summary[paste0("alpha[",1:length(uniqueV),"]"),]
  fit_JAGS$beta <-fit_JAGS$BUGSoutput$summary[paste0("beta[",1:length(uniqueV),"]"),]
  fit_JAGS$mu_beta <-fit_JAGS$BUGSoutput$summary["mu_beta",,drop=FALSE]
  fit_JAGS$mu_alpha <-fit_JAGS$BUGSoutput$summary["mu_alpha",,drop=FALSE]
  fit_JAGS$sigma2_beta <-fit_JAGS$BUGSoutput$summary["sigma2_beta",,drop=FALSE]
  fit_JAGS$sigma2_alpha <-fit_JAGS$BUGSoutput$summary["sigma2_alpha",,drop=FALSE]

  return(fit_JAGS)
}
```

```
gambia <-read.csv(file=file.path(datpath,"gambia.csv"))
gambia <-gambia[,-1]
locs <-unique(gambia[,c("x","y")])
#create village id
village <-rep(NA, length=nrow(gambia))
for(i in 1:nrow(locs)){
  village[which(gambia$x==locs$x[i] & gambia$y==locs$y[i] )] <-i
}

Y <-gambia$pos
X <-gambia$netuse
V <-village
fit_JAGS <-Bayes_logit_HW3Q3(Y,X,V)
fit_JAGS$beta
```

```
fit_JAGS$alpha
fit_JAGS$mu_alpha
fit_JAGS$mu_beta
fit_JAGS$sigma2_alpha
fit_JAGS$sigma2_beta

fit_JAGS_HC ←Bayes_logit_HW3Q3_HC(Y,X,V)
fit_JAGS_HC$beta
fit_JAGS_HC$alpha
fit_JAGS_HC$mu_alpha
fit_JAGS_HC$mu_beta
fit_JAGS_HC$sigma2_alpha
fit_JAGS_HC$sigma2_beta
```