Hierarchical modeling of rainfall data with (dependent) spatial covariates

STA 702 Fa24 Course Project

Submitted by

Arijit Dey

Net ID: ad638

Instructer

Li ma

Professor

Department of Statistical Science

Duke University

Date of Submission:

December 5, 2024

Contents

1 Introduction

Rainfall in India is a critical component of the country's climatic and agricultural systems, profoundly influencing its economy, food security, and water resource management. With a vast dependence on the monsoon, which contributes approximately 75% of the annual rainfall (MOSPI, 2012), predicting rainfall patterns is vital for planning and mitigating the risks associated with climate variability. The importance of accurate rainfall modeling extends beyond agriculture to sectors like disaster management, urban planning, and energy production.

This project analyzes and models rainfall across India using a data set that spans 122 years and covers 357 spatial pixels. Such a rich data set offers a unique opportunity to explore spatial and temporal trends in rainfall, identify anomalies, and assess the impact of long-term climatic shifts. By leveraging advanced statistical learning techniques, this project aims to provide insights into historical rainfall patterns and develop models to aid in better decision-making.

The sections of this document are organized as follows: Section 2 provides a description of the dataset used. Section 3 covers exploratory data analysis, including fitting simpler models, assessing their adequacy, and identifying the required covariates for the final model. Section 4 outlines the model specification and inference process. Finally, Section 5 discusses the results and highlights the limitations. Appendix A presents additional plots, derivations, and codes.

2 Data Set Description

We utilized yearly gridded rainfall data for India, measured in millimeters, with a spatial resolution of $1.0° \times 1.0°$, spanning the period from 1901 to 2022. This dataset was acquired from the official website of the Indian Meteorological Department, Pune (`https://www.imdpune.gov.in/cmpg/Griddata/Rainfall_1_NetCDF.html`). The gridded dataset was generated through spatial interpolation of ground-based station data, following the methodology detailed by Rajeevan et al. (2008). We derived annual rainfall information from this dataset for the specified time frame. Our final dataset has dimensions 122×357, where 357 represents the number of pixels. Further, the 122 years of data were divided into two subsets: a training set and a testing set. The first 100 years of data were used to train the model, while the remaining years were reserved for testing.

A visual representation of the pixels is provided in Appendix A.1.1.

3 Data Exploration

3.1 Fitting Simpler Models

To explore simpler models, we calculated the mean across all pixels to obtain the annual average rainfall for India from 1901 to 2022. This approach allowed us to assess whether the models performed well in capturing the overall rainfall trend. If satisfactory results were observed, we could proceed to fit the models for individual pixels.

With Y_{ij} being the actual data with $i = 1, \ldots, 122$ and $j = 1, \ldots, 357$, let the annual average rainfall of India is given as $Y_i = \sum_{j=1}^{S} Y_{ij}$. We fit the following two models with this data set.

3.1.1 Linear Regression Model

We fit our usual linear regression model with polynomials in time (t) as covariates. The model is given as

$$Y_i = \beta_0 + \sum_{p=1}^{P} \beta_p t^p + \epsilon_{ij} \text{ where } \epsilon_{ij} \sim N(0, \sigma^2). \tag{1}$$

We fit five models with $P = 1, 2, 3, 5,$ and 10. We use the `lm` function from `R` to fit the models. The observed data vs the fitted values is given in the left panel of Figure 1, where `F.i` represents the fitted regression line with time variable taken up to i-th order.

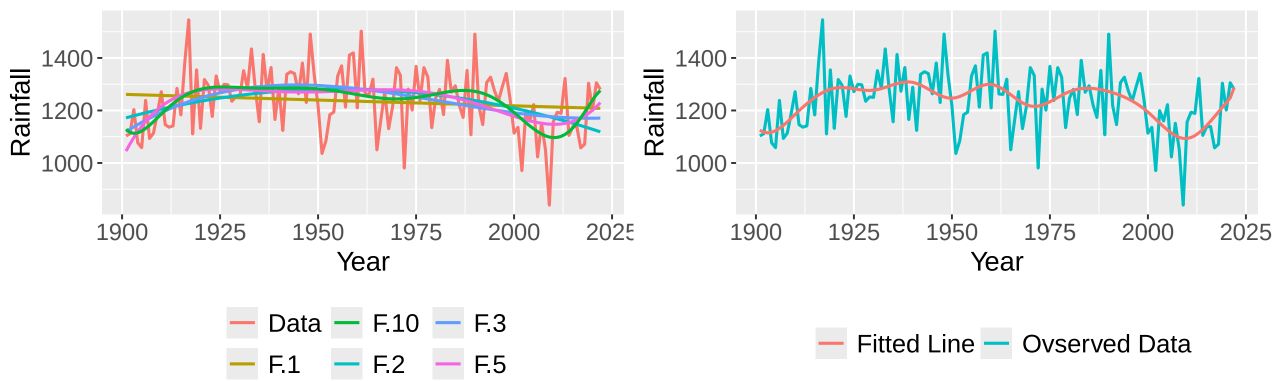

Figure 1: Observed data vs. Fitted models with linear regression (left) and basis spline regression (right).

3.1.2 Basis Spline Regression Model

Basis spline regression is a flexible modeling technique where the relationship between the response variable and the predictor variable is modeled using piecewise polynomial functions, called basis splines (B-splines). In our set up, the model is taken as

$$Y_i = \beta_0 + \sum_{j=1}^{k} \beta_j B_j(i) + \epsilon_i, \text{ where } \epsilon_{ij} \sim N(0, \sigma^2). \tag{2}$$

Here $B_j(i)$ is the basis spline function j evaluated at knot i, and β_j is the coefficients associated with the spline basis functions. The R code for this model fitting can be found in Appendix A.4.1. We present the observed data vs the fitted values in the right panel of Figure 1.

The plots from fitting simpler models reveal that none of them adequately capture the actual rainfall patterns. This suggests that relying solely on the time variable is insufficient. For the final model, we include features based on the rainfall of neighboring pixels. Specifically, we consider the near-1 and near-2 neighbors of each pixel (refer to Section A.1.1 for reference). In the following subsection, we provide justifications for selecting these covariates and outline the proposed model.

3.2 EDA

We begin by observing that neighboring pixels exhibit similar rainfall trends. This is expected, as adjacent pixels often receive similar amounts of rainfall.

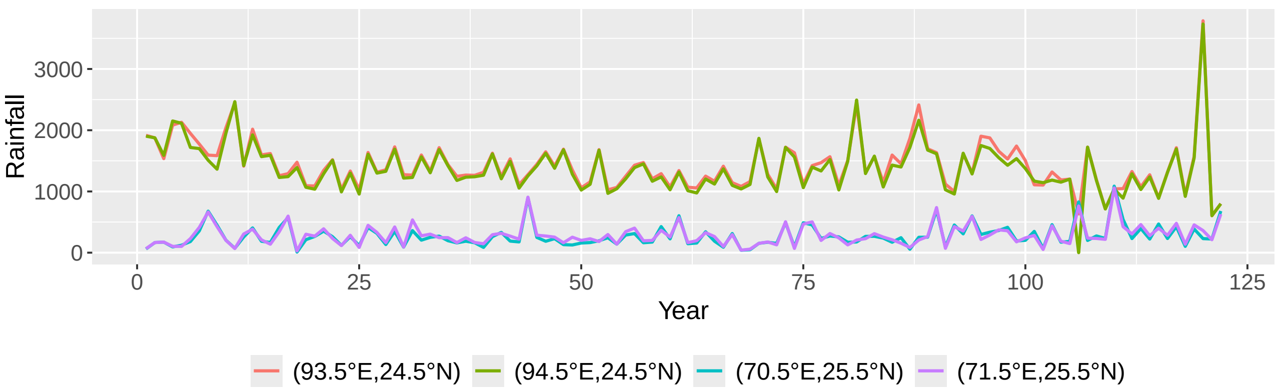

Figure 2: Rainfall patterns for two sets of two neighboring pixels.

To illustrate this, we plot the rainfall patterns of four neighboring pixels in Figure 2. In this figure, the green and orange pixels are neighbors, as are the purple and blue pixels, each pair showing corresponding similarities in rainfall patterns.

Inspired by the above plot, we use the mean rainfall of near-1 and near-2 neighbors as covariates. Below in Figure 3, we present two box plots showing the correlations between each pixel's rainfall and the mean rainfall of its near-1 and near-2 neighbors. For each pixel, we first calculate the yearly rainfall correlation with its neighbors and then average these values to obtain the mean neighboring correlation. The plot reveals a strong correlation between the rainfall of a pixel and that of its neighboring pixels.

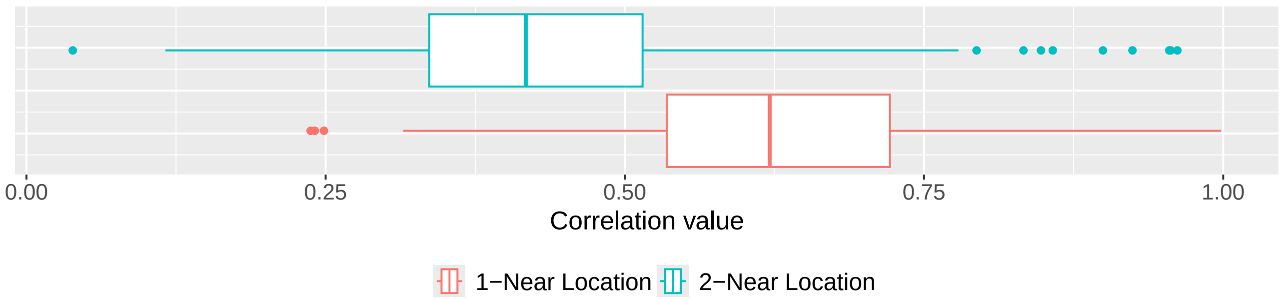

Figure 3: Box plot of correlations of rainfall with neighboring pixels.

In addition to these two covariates, we include the overall mean (calculated across all pixels) of the mean rainfall for near-1 and near-2 neighbors. This adjustment accounts for a significant number of pixels that have fewer near-1 and near-2 neighbors than the expected counts of 8 and 16, respectively (see Section A.1.2 for reference). We include these covariates with the expectation that they will provide additional support for pixels with fewer neighbors.

4 Methods

Building on the hierarchical regression model framework outlined in Section 11.1 of Hoff (2009), we fit a hierarchical regression model to this dataset. In this model, pixels are treated as groups, and yearly rainfall data represents observations within each group (pixel). To validate this assumption, we examine the stationarity and normality of rainfall across years for the pixels. The details, along with supporting figures, are provided in Section A.1.3 of the Appendix.

4.1 Model Description

Let, Y_{ij} is the annual total rainfall for i-th year and j-th locations of India. $\bar{Y}_{ij,1}$ is the mean of Y_{ij}'s for near-1 neighbors of ij-th locations. $\bar{Y}_{ij,2}$ is the mean of Y_{ij}'s for near-2 neighbors of ij-th locations, $\bar{Y}_{i,1}$ is mean of $\bar{Y}_{ij,1}$'s over j, and finally $\bar{Y}_{i,2}$ is mean of $\bar{Y}_{ij,2}$'s over j.

Then, final model is given as: for $i = 1, \ldots, T$, and $j = 1, \ldots, S$,

$$\begin{aligned} Y_{ij} &= \beta_{j1}\bar{Y}_{ij,1} + \beta_{j2}\bar{Y}_{ij,2} + \beta_{j3}\bar{Y}_{i,1} + \beta_{j4}\bar{Y}_{i,2} + \epsilon_{ij}, \\ \Rightarrow Y_{ij} &= \boldsymbol{x}_j^T\boldsymbol{\beta}_j + \epsilon_{ij}, \text{ where } \epsilon_{ij} \overset{\text{iid}}{\sim} N(0, \sigma^2). \end{aligned} \tag{3}$$

Here $\boldsymbol{x}_j^T = [\bar{Y}_{ij,1}, \bar{Y}_{ij,2}, \bar{Y}_{i,1}, \bar{Y}_{i,2}]'$ and $\boldsymbol{\beta}_j = [\beta_{j1}, \ldots, \beta_{j4}]'$. For the spae-varying regression coefficients, we have

$$\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_S \overset{\text{iid}}{\sim} N_4(\boldsymbol{\theta}, \Sigma) \tag{4}$$

For Bayesian analysis, we put priors on each of the parameters of the model above. Following Section 11.1 of Hoff (2009), we put semi-conjugate prior for each of the parameters. These are specified as

$$\begin{aligned} \boldsymbol{\theta} &\sim N_4(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0), \\ \Sigma &\sim \text{Inv-Wish}(\eta_0, \boldsymbol{S}_0), \\ \sigma^2 &\sim \text{Inv-Gam}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right). \end{aligned} \tag{5}$$

With this prior setup and the likelihood, the full conditional (FC) distributions of the parameters are given as follows.

$$\begin{aligned} \boldsymbol{\beta}_j|\boldsymbol{Y}_j, \boldsymbol{\theta}, \Sigma, \sigma^2 &\overset{\text{iid}}{\sim} N\Big[(\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1})^{-1}(\boldsymbol{X}_j^T\boldsymbol{Y}_j/\sigma^2 + \Sigma^{-1}\boldsymbol{\theta}), (\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1})^{-1}\Big], \\ \boldsymbol{\beta}_j|\boldsymbol{Y}_j, \boldsymbol{\theta}, \Sigma, \sigma^2 &\sim N\Big[(S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1})^{-1}(S\Sigma^{-1}\bar{\boldsymbol{\beta}} + \boldsymbol{\Lambda}_0^{-1}\boldsymbol{\mu}_0), (S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1})^{-1}\Big], \\ \Sigma|\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_S, \boldsymbol{\theta} &\sim \text{Inv-Wish}\big[\eta_0 + S, \boldsymbol{S}_{\boldsymbol{\theta}} + \boldsymbol{S}_0\big], \\ \sigma^2|\boldsymbol{Y}_j, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_S &\sim \text{Inv-gam}\left[\frac{ST + \nu_0}{2}, \frac{SSR + \nu_0\sigma_0^2}{2}\right]. \end{aligned} \tag{6}$$

Here, $\boldsymbol{Y}_j = [Y_{1j}, \ldots, Y_{Tj}]'$ and $\boldsymbol{X}_j = [\boldsymbol{x}_1, ..., \boldsymbol{x}_S]'$, $\bar{\boldsymbol{\beta}} = \sum_j \boldsymbol{\beta}_j/S$, $\boldsymbol{S}_{\boldsymbol{\theta}} = \sum_{j=1}^S(\boldsymbol{\beta}_j - \boldsymbol{\theta})(\boldsymbol{\beta}_j - \boldsymbol{\theta})^T$, and $SSR = \sum_{j=1}^S(\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j)^T(\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j)$.

The details of the derivations are given in Appendix A.2.

4.2 Model Inference

We estimate the model parameters through a Markov chain Monte Carlo (MCMC) framework, employing the Gibbs sampler (Gelfand, 2000) to generate samples from the full conditional distributions specified in (6). The hyperparameters are initialized based on the procedure outlined in Exercise 11.2 of Hoff (2009). To begin, we fit separate linear models for each pixel to obtain initial estimates of the parameters $\boldsymbol{\beta}_j$ and the residuals. These estimates are used to derive ad-hoc values for $\boldsymbol{\theta}$, Σ, and σ^2, denoted as $\hat{\boldsymbol{\theta}}$, $\hat{\Sigma}$, and $\hat{\sigma}^2$, respectively. Using these values, we set $\boldsymbol{\mu}_0 = \hat{\boldsymbol{\theta}}$, $\boldsymbol{\Lambda}_0 = \boldsymbol{S}_0 = \hat{\Sigma}$, and $\sigma_0^2 = \hat{\sigma}^2$. Additionally, η_0 is set to $p+2$ to ensure Σ is loosely centered around $\boldsymbol{S}_0$, while ν_0 is set to 1, reflecting a prior sample size of 1.

The `R` code for fitting this model is provided in Appendix A.4.2. A total of 10,000 samples were generated for each parameter, with the first 4,000 samples discarded as burn-in. The remaining samples were used for parameter estimation. The effective sample size (ESS) exceeded 5,000 for all parameters except $\boldsymbol{\theta}$, which had an ESS slightly above 3,000.

Additional MCMC convergence diagnostics, including trace plots and ACF plots for selected parameters, are provided in Appendix A.3.

5 Results and Discussion

We obtain estimates of the parameters $\boldsymbol{\theta}, \Sigma, \sigma^2$, and the latent variables $\boldsymbol{\beta}_j$ from the MCMC chains detailed in Section 4.2. Using these estimates, we predict the annual rainfall for all pixels for the years 2001–2022. For each i, j, the predicted values Y_{ij} are computed as $\hat{Y}_{ij} = \boldsymbol{X}_j\hat{\boldsymbol{\beta}}_j$. The observed data alongside the fitted model for six randomly selected pixels is shown in Figure 4. The model demonstrates an exceptionally good fit to the observed data, accurately capturing the rainfall patterns for the testing years across all the selected pixels.

A major limitation of this modeling approach is that the covariates for a pixel are derived from the rainfall of its neighboring pixels for the same year. As a result, the model cannot predict rainfall for future years. The remarkable accuracy in fitting the test years is primarily due to the inclusion of these covariates, which effectively capture the rainfall patterns. However, this comes at the cost of the model's predictive capability for future scenarios. This limitation

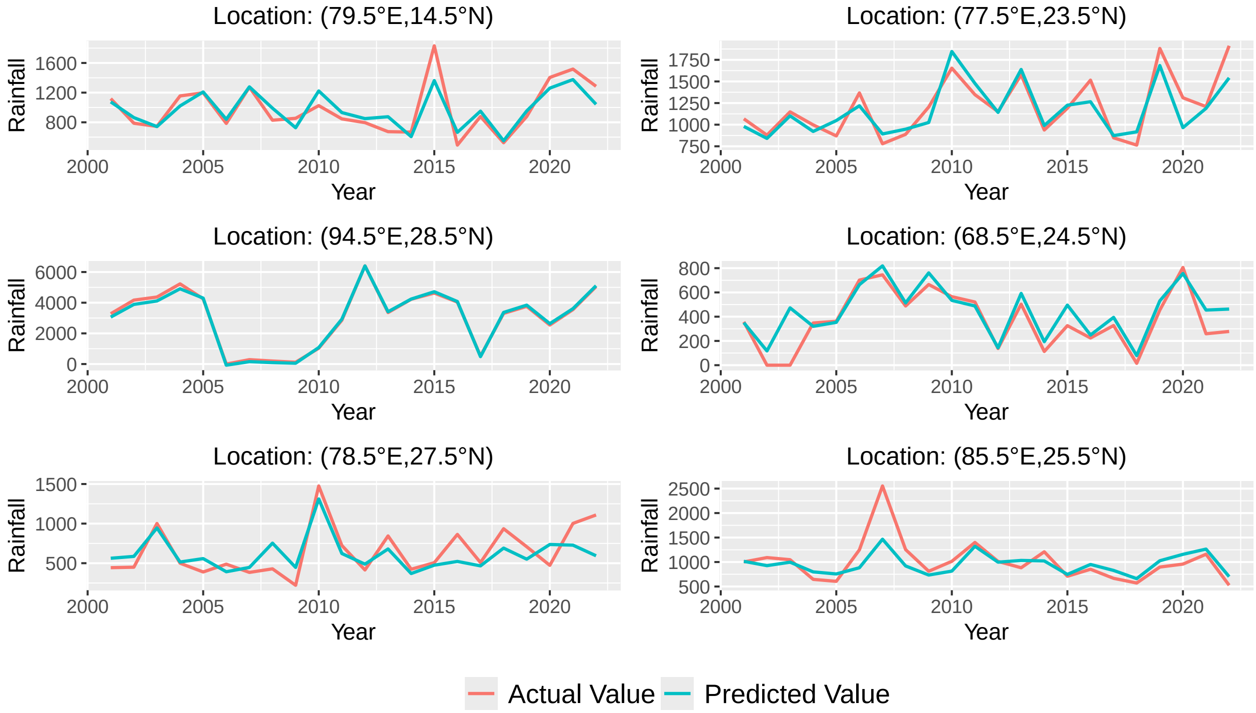

Figure 4: Model fitting for the test years on six randomly selected pixels.

could be addressed by replacing the current covariates with external factors such as humidity, temperature, or atmospheric pressure, which are highly correlated with rainfall. Incorporating these exogenous covariates would enable the model to retain its predictive capability for future years while still capturing the rainfall patterns effectively.

Alternatively, spatial models could be employed to account for the spatial dependency of annual rainfall across different pixels. Such models would preserve the predictive capability while effectively modeling the spatial relationships in the data.

A Appendix

A.1 EDA Plots

A.1.1 Pixels of India

In Figure 5 we plot the 357 pixels of India. For reference, in the same plot, we have demonstrated near-1-neighbors and near-2-neighbors with green and blue colors, respectively.

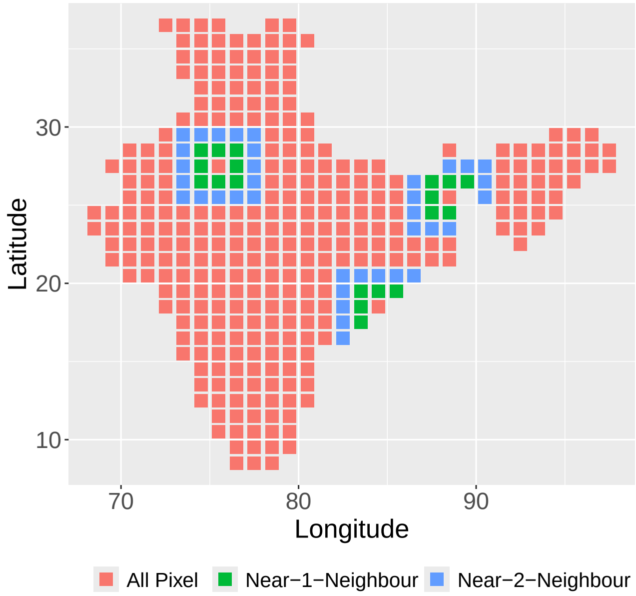

Figure 5: 357 Pixels of India with near-1 and near-2 neighbors.

A.1.2 Near neighbor pixel count

In Figure 6, we visualize the percentage of pixels with fewer than 8 near-1 neighbors and fewer than 16 near-2 neighbors.

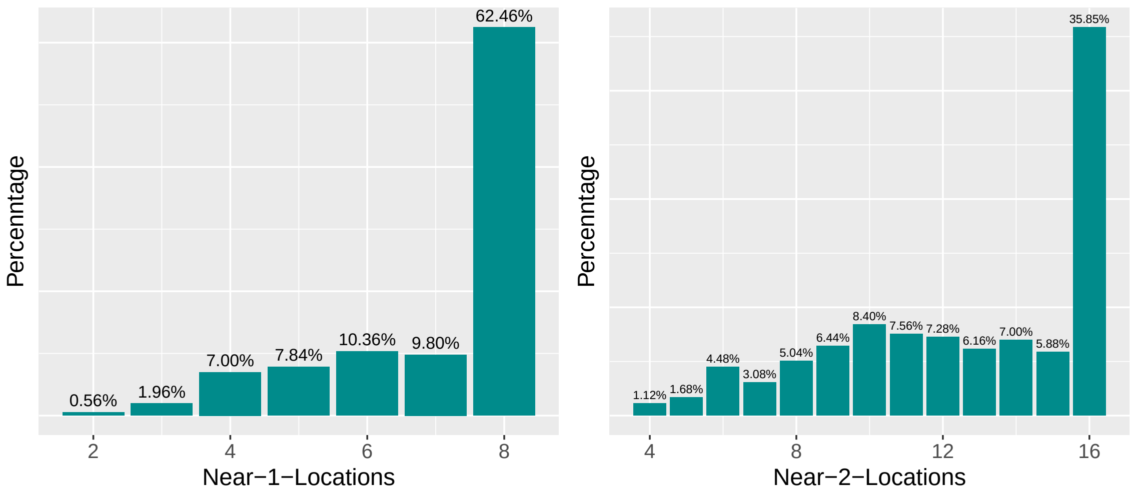

Figure 6: Number of near-1 and near-2 neighbors with corresponding percentages.

We observe that 63% of the pixels have the expected 8 near-1 neighbors, while only 36% have the full 16 near-2 neighbors. This indicates that a significant proportion of pixels have fewer neighbors than expected, particularly for near-2 neighbors.

A.1.3 Normality and Stationarity

Figure 7 presents the Q-Q plots of annual rainfall for four randomly selected pixels, demonstrating that the normality assumption is reasonably satisfied.

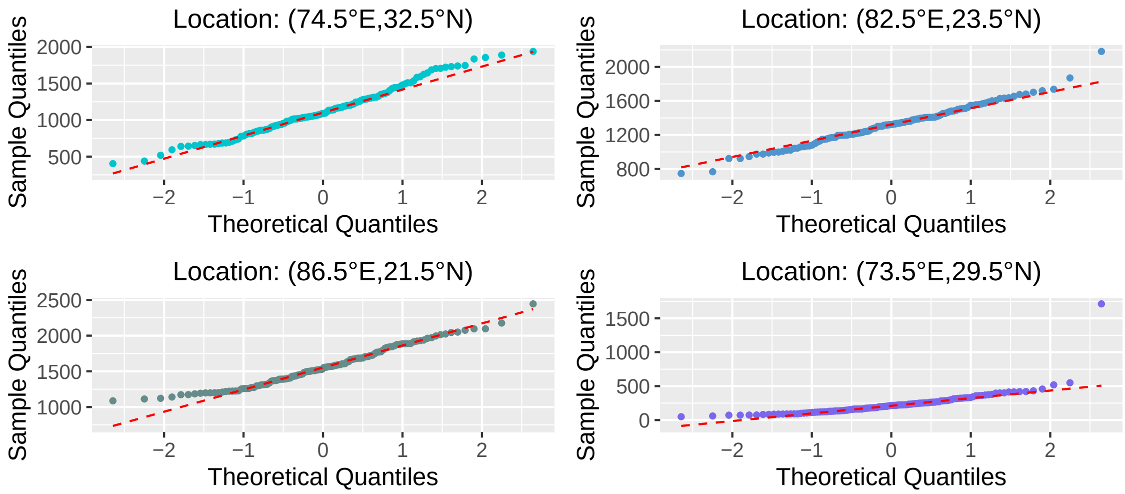

Figure 7: Q-Q Plots of four randomly selected locations.

Figure 8 displays the ACF plots of annual rainfall for four randomly selected pixels. The plot indicate that, for all four pixels, the autocorrelations remain within the significance bounds up to a lag of 30.

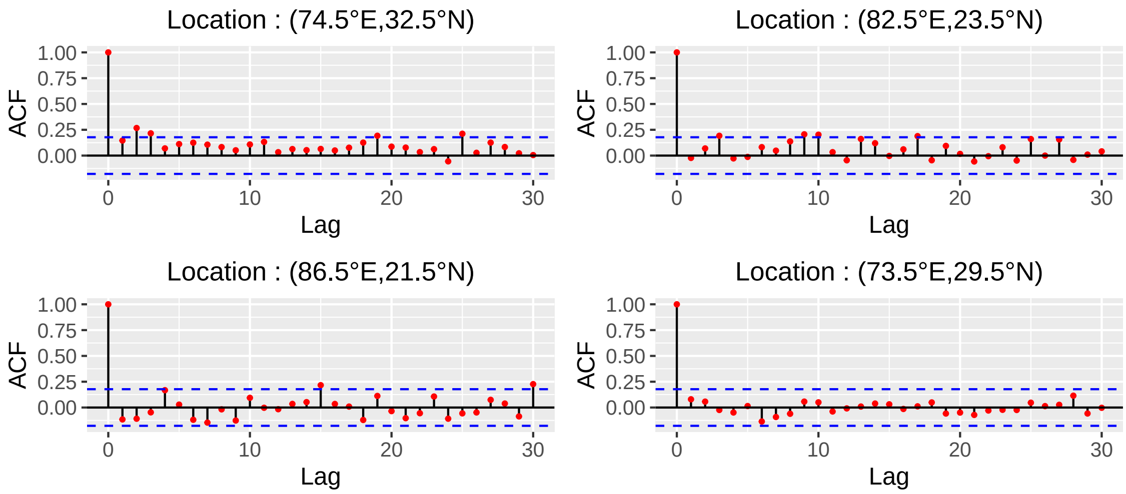

Figure 8: ACF Plots of four randomly selected locations.

A.2 Posterior Derivations

The sampling model and the latent variable model are given as

$$Y_{ij} \sim N(\boldsymbol{x}_j^T \boldsymbol{\beta}_j, \sigma^2), \text{ and } \boldsymbol{\beta}_j \overset{\text{iid}}{\sim} N(\boldsymbol{\theta}, \Sigma).$$

Now stacking $\boldsymbol{Y}_j = [Y_{1j}, \ldots, Y_{Tj}]'$ and $\boldsymbol{X}_j = [\boldsymbol{x}_1, ..., \boldsymbol{x}_S]'$, we can say $\boldsymbol{Y}_j \sim N(\boldsymbol{X}_j \boldsymbol{\beta}_j, \sigma^2 I_T)$. The prior specification is given as

$$\boldsymbol{\theta} \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0), \Sigma \sim \text{Inv-Wish}(\eta_0, \boldsymbol{S}_0), \text{ and } \sigma^2 \sim \text{Inv-Gam}(\nu_0/2, \nu_0\sigma_0^2/2).$$

Hence, the full posterior is given as

$$\Pi\big(\{\boldsymbol{\beta}_j\}, \boldsymbol{\theta}, \Sigma, \sigma^2 | \{\boldsymbol{Y}_j\}\big) \propto \Pi_{j=1}^S \Big[\Pi(\boldsymbol{Y}_j | \boldsymbol{\beta}_j, \sigma^2)\Big] \times \Pi_{j=1}^S \Big[\Pi(\boldsymbol{\beta}_j | \boldsymbol{\theta}, \Sigma)\Big] \times \Pi(\boldsymbol{\theta}) \times \Pi(\Sigma) \times \Pi(\sigma^2)$$

<u>The FC distribution of $\boldsymbol{\beta}_j$ is given as follows:</u>

$$\begin{aligned}
&\Pi(\boldsymbol{\beta}_j | \boldsymbol{Y}_j, \boldsymbol{\theta}, \Sigma, \sigma^2) \propto \Pi(\boldsymbol{Y}_j | \boldsymbol{\beta}_j, \sigma^2 I_T) \times \Pi(\boldsymbol{\beta}_j | \boldsymbol{\theta}, \Sigma) \\
&\propto \exp\Big[-\frac{1}{2\sigma^2}(\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j)^T(\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j)\Big] \times \exp\Big[-\frac{1}{2}(\boldsymbol{\beta}_j - \boldsymbol{\theta})^T \Sigma^{-1} (\boldsymbol{\beta}_j - \boldsymbol{\theta})\Big] \\
&\propto \exp\Big[-\frac{1}{2}\Big\{\boldsymbol{\beta}_j^T\big(\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1}\big)\boldsymbol{\beta}_j - 2\boldsymbol{\beta}_j^T\big(\boldsymbol{X}_j^T\boldsymbol{Y}_j/\sigma^2 + \Sigma^{-1}\boldsymbol{\theta}\big)\Big\}\Big]
\end{aligned}$$

Now, this is a kernel of a multivariate normal distribution with mean vector $(\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1})^{-1}(\boldsymbol{X}_j^T\boldsymbol{Y}_j/\sigma^2 + \Sigma^{-1}\boldsymbol{\theta})$ and covariance matrix $(\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1})^{-1}$. Hence

$$\boldsymbol{\beta}_j | \boldsymbol{Y}_j, \boldsymbol{\theta}, \Sigma, \sigma^2 \sim N\Big[(\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1})^{-1}(\boldsymbol{X}_j^T\boldsymbol{Y}_j/\sigma^2 + \Sigma^{-1}\boldsymbol{\theta}), (\boldsymbol{X}_j^T\boldsymbol{X}_j/\sigma^2 + \Sigma^{-1})^{-1}\Big]$$

<u>The FC distribution of $\boldsymbol{\theta}$ is given as follows:</u>

$$\begin{aligned}
&\Pi(\boldsymbol{\theta} | \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_S, \Sigma) \propto \Pi_{j=1}^S \Big[\Pi(\boldsymbol{\beta}_j | \boldsymbol{\theta}, \Sigma)\Big] \times \Pi(\boldsymbol{\theta}) \\
&\propto \exp\Big[-\frac{1}{2}\sum_{j=1}^S (\boldsymbol{\beta}_j - \boldsymbol{\theta})^T \Sigma^{-1} (\boldsymbol{\beta}_j - \boldsymbol{\theta})\Big] \times \exp\Big[-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_0)^T \boldsymbol{\Lambda}_0^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_0)\Big] \\
&\propto \exp\Big[-\frac{1}{2}\Big\{\boldsymbol{\theta}^T\big(S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1}\big)\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\big(\Sigma^{-1}S\bar{\boldsymbol{\beta}} + \boldsymbol{\Lambda}_0^{-1}\boldsymbol{\mu}_0\big)\Big\}\Big], \text{ where } \bar{\boldsymbol{\beta}} = \sum_{j=1}^S \boldsymbol{\beta}_j / S
\end{aligned}$$

Now, this is a kernel of a multivariate normal distribution with mean vector $(S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1})^{-1}(S\Sigma^{-1}\bar{\boldsymbol{\beta}} + \boldsymbol{\Lambda}_0^{-1}\boldsymbol{\mu}_0)$ and covariance matrix $(S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1})^{-1}$. Hence

$$\boldsymbol{\theta} | \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_S, \Sigma \sim N\Big[(S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1})^{-1}(S\Sigma^{-1}\bar{\boldsymbol{\beta}} + \boldsymbol{\Lambda}_0^{-1}\boldsymbol{\mu}_0), (S\Sigma^{-1} + \boldsymbol{\Lambda}_0^{-1})^{-1}\Big]$$

The FC distribution of Σ is given as follows:

$$\Pi(\Sigma|\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_S,\boldsymbol{\theta}) \propto \Pi_{j=1}^{S}\Big[\boldsymbol{\beta}_j|\boldsymbol{\theta},\Sigma)\Big] \times \Pi(\Sigma)$$

$$\propto |\Sigma|^{S/2}\exp\Big[-\frac{1}{2}\sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T\Sigma^{-1}(\boldsymbol{\beta}_j-\boldsymbol{\theta})\Big] \times |\Sigma|^{-(\eta_0+p+1)/2}\exp\Big[-\frac{1}{2}\mathrm{tr}(\boldsymbol{S}_0\Sigma^{-1})\Big]$$

$$\propto |\Sigma|^{(\eta_0+S+p+1)/2}\exp\Big[-\frac{1}{2}\mathrm{tr}\Big\{\big(\sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T+\boldsymbol{S}_0\big)\Sigma^{-1}\Big\}\Big]$$

$$\Big[\text{As, } \sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T\Sigma^{-1}(\boldsymbol{\beta}_j-\boldsymbol{\theta}) = \mathrm{tr}\Big\{\sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T\Sigma^{-1}(\boldsymbol{\beta}_j-\boldsymbol{\theta})\Big\}$$

$$= \mathrm{tr}\Big\{\sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T(\boldsymbol{\beta}_j-\boldsymbol{\theta})\Sigma^{-1}\Big\} = \mathrm{tr}\Big\{\big(\sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T\big)\Sigma^{-1}\Big\}\Big]$$

$$= |\Sigma|^{(\eta_0+S+p+1)/2}\exp\Big[-\frac{1}{2}\mathrm{tr}\Big\{\big(\boldsymbol{S}_{\boldsymbol{\theta}}+\boldsymbol{S}_0\big)\Sigma^{-1}\Big\}\Big], \text{ where } \boldsymbol{S}_{\boldsymbol{\theta}} = \sum_{j=1}^{S}(\boldsymbol{\beta}_j-\boldsymbol{\theta})(\boldsymbol{\beta}_j-\boldsymbol{\theta})^T$$

Now, this is a kernel of an inverse Wishart distribution with parameters $(\eta_0 + S)$ and $(\boldsymbol{S}_{\boldsymbol{\theta}} + \boldsymbol{S}_0)$. Hence

$$\Sigma|\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_S,\boldsymbol{\theta} \sim \text{Inv-Wish}\big[\eta_0+S, \boldsymbol{S}_{\boldsymbol{\theta}}+\boldsymbol{S}_0\big]$$

The FC distribution of σ^2 is given as follows:

$$\Pi(\sigma^2|\boldsymbol{Y}_j,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_S) \propto \Pi_{j=1}^{S}\Big[\Pi(\boldsymbol{Y}_j|\boldsymbol{\beta}_j,\sigma^2)\Big] \times \Pi(\sigma^2)$$

$$\propto (\sigma^2)^{-ST/2}\exp\Big[-\frac{1}{2\sigma^2}\sum_{j=1}^{S}(\boldsymbol{Y}_j-\boldsymbol{X}_j\boldsymbol{\beta}_j)^T(\boldsymbol{Y}_j-\boldsymbol{X}_j\boldsymbol{\beta}_j)\Big] \times (\sigma^2)^{-\nu_0/2}\exp\Big[-\frac{\nu_0\sigma_0^2}{2\sigma^2}\Big]$$

$$\propto (\sigma^2)^{-(ST+\nu_0)/2} \times \exp\Big[-\frac{SSR+\nu_0\sigma_0^2}{2\sigma^2}\Big], \text{ where } SSR = \sum_{j=1}^{S}(\boldsymbol{Y}_j-\boldsymbol{X}_j\boldsymbol{\beta}_j)^T(\boldsymbol{Y}_j-\boldsymbol{X}_j\boldsymbol{\beta}_j).$$

Now, this is a kernel of an inverse gamma distribution with parameters $(ST + \nu_0)/2$ and $(SSR + \nu_0\sigma_0^2)/2$. Hence

$$\sigma^2|\boldsymbol{Y}_j,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_S \sim \text{Inv-gam}\Big[\frac{ST+\nu_0}{2}, \frac{SSR+\nu_0\sigma_0^2}{2}\Big]$$

A.3 MCMC Convergence Plots

This section presents the trace plots and autocorrelation function (ACF) plots for selected parameters. One location (s) is randomly selected, and two of the four parameters from $\boldsymbol{\beta}_s$ are plotted. Additionally, one element of $\boldsymbol{\theta}$ is chosen randomly, and two elements are randomly selected from the 16 elements of Σ. The trace plots and ACF plots for these parameters are shown in Figures 9 and 10, respectively.

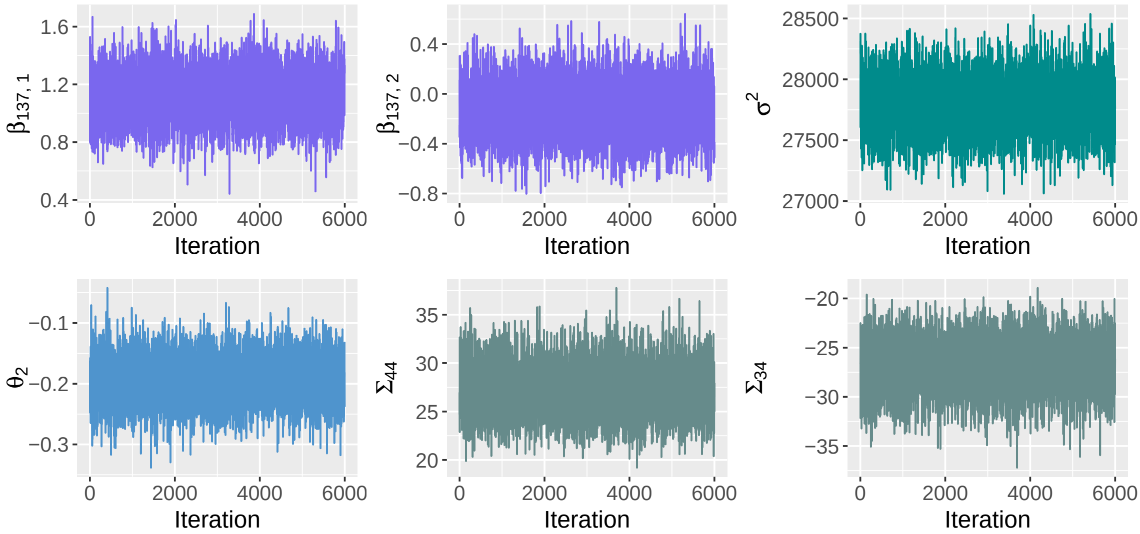

Figure 9: Tace Plots of eight randomly selected parameters.

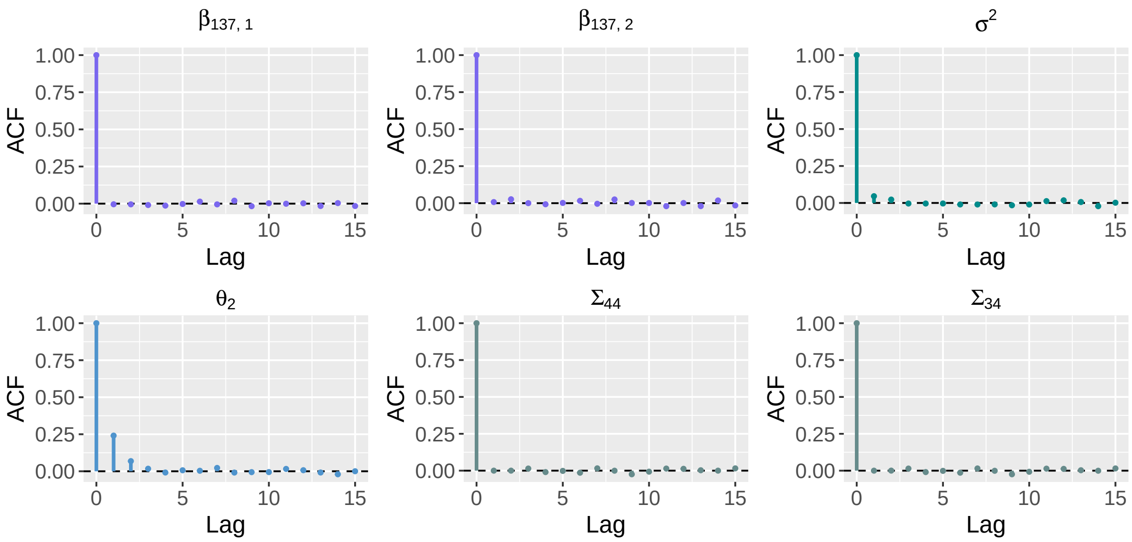

Figure 10: ACF Plots of eight randomly selected parameters.

A.4 R Codes

All the R code (along with other relevant documents) can be found at `https://github.com/arijitdey22/STA702Fa24`. Here are some snippets of the important codes.

A.4.1 Code for Basis Spline regression

```r
library(splines)
set.seed(123)
years <- 1901:2022
annual.rainfall <- rowMeans(all.data)
rainfall.df <- data.frame(Year = years, Rainfall = annual.rainfall)
knots <- seq(1900, 2020, 10)
rainfall.df$Spline  <- bs(rainfall.df$Year, knots = knots, degree = 3,
    intercept = TRUE)
sp.model <- lm(Rainfall ~ Spline, data = rainfall.df)
Predicted.Rainfall <- predict(sp.model, newdata = rainfall.df)
```

A.4.2 Code for Model Fitting

```r
library(LaplacesDemon)
library(mvtnorm)
n.s <- ncol(Y.ij.train)
n.t <- nrow(Y.ij.train)
p <- 4
# creating X.j and Y.j
X.j <- list()
Y.j <- list()
df.j <- list()
for (j in 1:n.s){
  Y.j[[j]] <- Y.ij.train[,j]
  X.j[[j]] <- matrix(c(Y.bar.ij.1.train[,j],Y.bar.ij.2.train[,j],
                       Y.bar.i.1.train, Y.bar.i.2.train), ncol = p)
  foo <- cbind(Y.j[[j]], X.j[[j]])
  df.j[[j]] <- as.data.frame(foo)
}
library(Matrix)
X.all.bdiag <- as.matrix(bdiag(X.j))
# getting hyper priors
beta.j.store <- matrix(0, n.s, p)
```

```r
res.j.store <- numeric(length = n.s)
for (j in 1:n.s){
  lm.j <- lm(data = df.j[[j]], V1 ~ 0 + .)
  beta.j.store[j,] <- unname(lm.j$coefficients)
  res.j.store[j] <- anova(lm.j)['Residuals','Mean Sq']
}
theta.hat <- colMeans(beta.j.store)
Sigma.hat <- cov(beta.j.store)
sigma.sq.hat <- mean(res.j.store)
mu.0 <- theta.hat
Lambda.0 <- Sigma.hat
eta.0 <- p + 2
S.0 <- Sigma.hat
nu.0 <- 2
sigma.sq.0 <- sigma.sq.hat
Lambda.0.inv <- solve(Lambda.0)
Lambda.0.inv.times.mu.0 <- Lambda.0.inv %*% mu.0
# initial values
theta <- as.vector(rmvnorm(1, mu.0, Lambda.0))
Sigma <- rinvwishart(eta.0, S.0)
beta <- rmvnorm(n.s, theta, Sigma)
sigma.sq <- 1 / rgamma(1, nu.0/2, nu.0 * sigma.sq.0 / 2)
# MCMC length and burn-in
iter <- 1e4
burn <- 4e3
# storage for parameters
store.beta <- array(0, dim = c(iter, n.s, p))
store.theta <- matrix(0, ncol = p, nrow = iter)
store.Sigma <- array(0,  dim = c(iter, p, p))
store.sigma.sq <- numeric(length = iter)
store.DIC <- numeric(length = iter)

for (i in 1:iter){
  Sigma.inv <- solve(Sigma)
  #updating beta
```

```
  for (j in 1:n.s){
    beta.j.par.2 <- solve(Lambda.0.inv + t(X.j[[j]]) %*% X.j[[j]] / sigma.sq)
    beta.j.par.1 <- beta.j.par.2 %*% ( Sigma.inv %*% theta + t(X.j[[j]]) %*% Y.j[[j]] / sigma.sq)
    beta[j, ] <- rmvnorm(1, beta.j.par.1, beta.j.par.2)
  }
  #updating theta
  beta.bar <- colMeans(beta)
  theta.par.2 <- solve(Lambda.0.inv + n.s * Sigma.inv)
  theta.par.1 <- theta.par.2 %*% (Lambda.0.inv.times.mu.0 + n.s * Sigma.inv %*% beta.bar)
  theta <- as.vector(rmvnorm(1, theta.par.1, theta.par.2))
  #updating Sigma
  diff <- beta - matrix(rep(theta, each = nrow(beta)), nrow = nrow(beta))
  S.theta <- t(diff) %*% diff
  Sigma <- rinvwishart(eta.0 + n.s, S.0 + S.theta)
  #updating sigma.sq
  foo <- matrix(X.all.bdiag %*% as.vector(t(beta)), ncol = 357)
  SSR <- sum( (Y.ij.train - foo)^2)
  sigma.sq <- 1 / rgamma(1, (nu.0 + n.t * n.s) / 2, (nu.0 * sigma.sq.hat + SSR)/2)
  #storing
  store.beta[i, , ] <- beta
  store.theta[i, ] <- theta
  store.Sigma[i, , ] <- Sigma
  store.sigma.sq[i] <- sigma.sq
  store.DIC[i] <- SSR / sigma.sq
}
keep.ind <- (burn+1) : iter
samp <- list(beta = store.beta[keep.ind, , ],
             theta = store.theta[keep.ind, ],
             Sigma = store.Sigma[keep.ind, , ],
             sigma.sq = store.sigma.sq[keep.ind],
             DIC = store.DIC[keep.ind])
```

A.4.3 Code for Model Testing

```r
## Parameter estimates:
sigma.sq <- mean(samp$sigma.sq)
Sigma <- apply(samp$Sigma, c(2, 3), mean)
theta <- colMeans(samp$theta)
beta <- apply(samp$beta, c(2, 3), mean)
## Prediction:
load("03 Data for model.test.Rdata")
n.s <- ncol(Y.ij.test)
n.t <- nrow(Y.ij.test)
Y.ij.test.hat <- matrix(0, ncol = n.s, nrow = n.t)
for (j in 1:n.s){
  beta.j <- beta[j, ]
  X.j <- matrix(c(Y.bar.ij.1.test[,j], Y.bar.ij.2.test[,j], Y.bar.i.1.test,
                  Y.bar.i.2.test), ncol = 4)
  Y.ij.test.hat[,j] <- X.j %*% beta.j
}
```

References

Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American statistical Association*, 95(452):1300–1304.

Hoff, P. D. (2009). *A first course in Bayesian statistical methods*, volume 580. Springer.

MOSPI (2012). Statistical year book india - rainfall. https://www.mospi.gov.in/sites/default/files/Statistical_year_book_india_chapters/Rainfall_0.pdf.

Rajeevan, M., Bhate, J., and Jaswal, A. K. (2008). Analysis of variability and trends of extreme rainfall events over india using 104 years of gridded daily rainfall data. *Geophysical research letters*, 35(18).