Project 2 - TMA4300: Computer Intensive Statistical Methods

Yawar Mahmood, Baptiste Tesson

17.03.2024

Introduction

In this project, we focus on a segment of the renowned Tokyo rainfall dataset, which encompasses daily rainfall measurements spanning from 1951 to 1989. Our interest lies in the variable y_t, representing instances where daily rainfall surpasses the 1mm threshold.

Our data follows the following model:

$$y_t|x_t \sim \text{Bin}\left(n_t, \pi(x_t)\right), \quad \pi(x_t) = \frac{\exp\{x_t\}}{1+\exp\{x_t\}} = \frac{1}{1+\exp\{-x_t\}}, \tag{1}$$

where n_t denotes the number of trials, and $\pi(x_t)$ symbolizes the success probability for each trial.

Problem 1

a)

We start our analysis at the most natural step in any data analysis process - exploring the data. We use the inbuilt "summary()" function in R to get a quick overview of the data.

```
      day            n.years         n.rain      
 Min.   :  1.00   Min.   :10.00   Min.   : 0.00  
 1st Qu.: 92.25   1st Qu.:39.00   1st Qu.: 8.00  
 Median :183.50   Median :39.00   Median :11.00  
 Mean   :183.50   Mean   :38.92   Mean   :10.98  
 3rd Qu.:274.75   3rd Qu.:39.00   3rd Qu.:14.00  
 Max.   :366.00   Max.   :39.00   Max.   :23.00  
```

Figure 1: Summary of the Tokyo rain dataset

The summary (Figure 1) data provides a statistical overview of the data, broken into three main variables: day (day), number of years observed (n.years), and the number of rain occurrences (n.rain). Let's look closer at each variable and what it may tell us:

- Day: The data covers days from the first day of the year to the 366th day of the year. That is, it spans an entire year including a leap year.
- Number of Years Observed (n.years): The data represents observations across 39 years for each day. Interesting point here is that the mean and the maximum value for n.years is 39 (the mean is 38.92, which for all practical reasons is 39). This indicates that the observations have been consistent across all days, across the years.
- Number of Rain Occurrences (n.rain): The minimum and maximum vary from 0 to 23. This indicates that on some days, it never rained and on others it rained up to 23 times, across the observed years.

  Looking at the quantiles of the number of rain occurrences, we see that the first quantile is at 8, meaning that 25% of the days had rain occurrences 8 times or fewer across the observed years. The median 11 indicates that half of the days had 11 or fewer rain occurrences. The third quantile is at 14, meaning 75% of the days rain 14 times or fewer.

  The mean (10.98) and median (11) of rain occurrences are close, suggesting a relatively symmetrical distribution of rain occurrences across days. There are some outliers though, as indicated by the maximum at 23.

With the quantitative analysis out of the way, let us plot the response as a function of time t (in days):

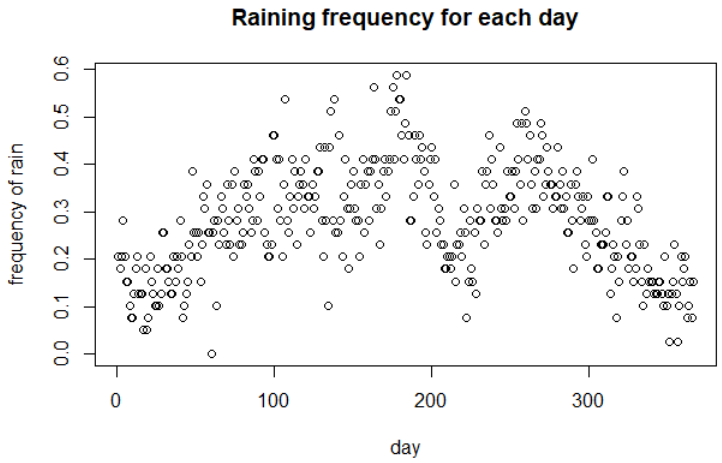

Figure 2: Rain frequency for each day t

There is clearly a pattern in figure 2. A natural point to start when looking at the data is to think about how the different seasons of the year affect the rainfall. We see that during winter (t close to 0 and 366), there is less rain,

compared to the summer months $t \in [100 - 250]$. Looking more locally, one can see a more chaotic pattern. This suggests that there is high variance of rainfall between two consecutive days.

Looking more locally, one can see a more "spiky" pattern. This suggest that if it rains a lot in a period, we could expect it to be less rain in the following period, and opposite.

b)

We introduce the notations: $\boldsymbol{x} = (x_1, \cdots, x_{366})^T$, $\boldsymbol{y} = (y_1, \cdots, y_{366})^T$, and $\boldsymbol{\pi} = (\pi(x_1), \cdots, \pi(x_{366}))^T$.

Given these notations, the likelihood $L(\boldsymbol{\pi}|\boldsymbol{y})$ is defined by:

$$L(\boldsymbol{\pi}|\boldsymbol{y}) = f(y_1, \ldots, y_{366}|\pi(x_1), \ldots, \pi(x_{366})), \tag{2}$$

where f represents the joint conditional distribution. Considering the conditional independence of y_i from all $\pi(x_j)$ for $j \neq i$, the likelihood can be expressed as the product of individual conditional distributions:

$$L(\boldsymbol{\pi}|\boldsymbol{y}) = \prod_{i=1}^{366} f(y_i|\pi(x_i)). \tag{3}$$

Given that:

$$y_t|x_t \sim \text{Bin}(n_t, \pi(x_t)), \quad \pi(x_t) = \frac{\exp\{x_t\}}{1 + \exp\{x_t\}} = \frac{1}{1 + \exp\{-x_t\}}, \tag{4}$$

the likelihood function simplifies to:

$$L(\boldsymbol{\pi}|\boldsymbol{y}) = f(\boldsymbol{y}|\boldsymbol{\pi}), \tag{5}$$

where:

$$f(\boldsymbol{y}|\boldsymbol{\pi}) = \prod_{i=1}^{T} \binom{n_i}{y_i} \pi(x_i)^{y_i} (1 - \pi(x_i))^{n_i - y_i}. \tag{6}$$

c)

We aim to apply a Bayesian hierarchical model to our dataset, with a random walk of order 1 (RW(1)) to model the trend on the logit scale.

$$x_t = x_{t-1} + u_t, \tag{7}$$

where $u_t \sim N(0, \sigma_u^2)$ are independent and identically distributed. Consequently, $p(x|\sigma_u^2)$ is described by a normal distribution:

$$p(x|\sigma_u^2) \propto \prod_{t=2}^{T} \frac{1}{\sigma_u} \exp\left(-\frac{1}{2\sigma_u^2}(x_t - x_{t-1})^2\right). \tag{8}$$

An inverse gamma prior is adopted for σ_u^2, defined as:

$$p(\sigma_u^2) = \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma_u^2}\right)^{\alpha+1} \exp\left(-\frac{\beta}{\sigma_u^2}\right), \tag{9}$$

where α and β denote the shape and scale parameters, respectively. We then derive the conditional $p(\sigma_u^2|\mathbf{y}, \mathbf{x})$, which is proportional to:

$$p(\sigma_u^2|\mathbf{y}, \mathbf{x}) \propto p(\mathbf{y}|\sigma_u^2, \mathbf{x}) \cdot p(\mathbf{x}|\sigma_u^2) \cdot p(\sigma_u^2). \tag{10}$$

Given that $\mathbf{y}$ is conditionally independent of σ_u^2 once $\mathbf{x}$ is known, it follows that $p(\sigma_u^2|\mathbf{y}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})$. Thus, we obtain:

$$p(\sigma_u^2|\mathbf{y}, \mathbf{x}) \propto p(\mathbf{x}|\sigma_u^2) \cdot p(\sigma_u^2). \tag{11}$$

Rewriting this expression provides:

$$p(\sigma_u^2|\mathbf{y}, \mathbf{x}) \propto \prod_{t=2}^{T} \frac{1}{\sigma_u} \exp\left(-\frac{1}{2\sigma_u^2}(x_t - x_{t-1})^2\right) \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma_u^2}\right)^{\alpha+1} \exp\left(-\frac{\beta}{\sigma_u^2}\right). \tag{12}$$

Disregarding the constant term $\frac{\beta^\alpha}{\Gamma(\alpha)}$, the above simplifies to an inverse gamma distribution with updated parameters:

$$p(\sigma_u^2|\mathbf{y}, \mathbf{x}) = (\sigma_u^2)^{-(\alpha + \frac{T-1}{2} + 1)} \exp\left(-\frac{(\beta + \frac{1}{2}\sum_{t=2}^{T}(x_t - x_{t-1})^2)}{\sigma_u^2}\right), \tag{13}$$

where the new parameters for the inverse gamma distribution are:

$$\alpha' = \alpha + \frac{T-1}{2}, \quad \beta' = \beta + \frac{1}{2}\sum_{t=2}^{T}(x_t - x_{t-1})^2. \tag{14}$$

d)

Before employing the Metropolis-Hastings algorithm to sample $\boldsymbol{x}$, we establish the formula for the acceptance probability:

Let the conditional prior proposal distribution be defined as $Q(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)$, where $\boldsymbol{x}'_{\mathcal{I}}$ represents the proposed values for $\boldsymbol{x}_{\mathcal{I}}$, $\mathcal{I} \subseteq \{1, \ldots, 366\}$ denotes a set of time indices, and $\boldsymbol{x}_{-\mathcal{I}} = \boldsymbol{x}_{\{1,\ldots,366\}\setminus\mathcal{I}}$ signifies the subset of $\boldsymbol{x}$ excluding indices in $\mathcal{I}$. The acceptance probability α is then expressed as:

$$\alpha = \min\left(1, \frac{p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})Q(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}{p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})Q(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}\right). \tag{15}$$

Focusing on the first fraction:

$$\frac{p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}{p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}, \tag{16}$$

the numerator becomes:

$$p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = \frac{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}'_{\mathcal{I}})p(\boldsymbol{y}_{-\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}})p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}{p(\boldsymbol{y}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}, \tag{17}$$

assuming conditional independence for $\boldsymbol{y}_t|\boldsymbol{x}_t$.

A similar derivation applies to the denominator, leading to a significant simplification. The acceptance probability thus simplifies to:

$$\alpha = \min\left(1, \frac{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}'_{\mathcal{I}})p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}_{\mathcal{I}})p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}\right). \tag{18}$$

Given that Q represents the prior proposal distribution, the ratio between Q terms cancels out, leaving:

$$\alpha = \min\left(1, \frac{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}'_{\mathcal{I}})}{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}_{\mathcal{I}})}\right). \tag{19}$$

This final expression precisely articulates the acceptance probability we sought to elucidate.

e)

Given:

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_A \\ \boldsymbol{x}_B \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{AA} & \mathbf{Q}_{AB} \\ \mathbf{Q}_{BA} & \mathbf{Q}_{BB} \end{pmatrix},$$

the joint distribution $p(\boldsymbol{x}|\sigma_u^2)$ is represented, after accounting for the improper prior, as:

$$p(\boldsymbol{x}|\sigma_u^2) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x}\right).$$

Expanding the quadratic form $\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x}$, we obtain:

$$\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} = \boldsymbol{x}_A^T\mathbf{Q}_{AA}\boldsymbol{x}_A + \boldsymbol{x}_A^T\mathbf{Q}_{AB}\boldsymbol{x}_B + \boldsymbol{x}_B^T\mathbf{Q}_{BA}\boldsymbol{x}_A + \boldsymbol{x}_B^T\mathbf{Q}_{BB}\boldsymbol{x}_B.$$

To compute $p(\boldsymbol{x}_A|\boldsymbol{x}_B, \sigma_u^2)$, we note:

$$p(\boldsymbol{x}_A|\boldsymbol{x}_B, \sigma_u^2) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}_A^T\mathbf{Q}_{AA}\boldsymbol{x}_A + 2\boldsymbol{x}_A^T\mathbf{Q}_{AB}\boldsymbol{x}_B)\right). \tag{20}$$

This includes both a linear and quadratic term in $\boldsymbol{x}_A$. The precision matrix $\mathbf{Q}_{A|B}$ is $\mathbf{Q}_{AA}$, derived from the quadratic term.

To find the mean $\mu_{A|B}$, we equate to the form $-(x-\mu)^T\Sigma^{-1}(x-\mu)$, resulting in:

$$\mu_{A|B} = -\mathbf{Q}_{AA}^{-1}\mathbf{Q}_{AB}\boldsymbol{x}_B,$$

where $\mu_{A|B}$ represents the conditional mean of $\boldsymbol{x}_A$ given $\boldsymbol{x}_B$.

f)

To facilitate the implementation of the MCMC process, we seek a more intuitive formula for the acceptance probability. We do this through log scaling, obtaining:

$$\alpha = \min\left(1, \frac{\prod_{i\in I}\pi(x_i')^{y_i}(1-\pi(x_i'))^{n_i-y_i}}{\prod_{i\in I}\pi(x_i)^{y_i}(1-\pi(x_i))^{n_i-y_i}}\right)$$

Simplifying this expression, skipping some intermediate steps, gives:

$$\alpha = \min\left(1, \exp\left(\sum_{i\in I}\left[y_i(x_i'-x_i) + n_i\log\left(\frac{1+e^{x_i}}{1+e^{x_i'}}\right)\right]\right)\right)$$

With this expression in place, it is possible to implement the Metropolis-Hastings step, which will be used in the MCMC process later.

After running the process, we wish to determine the performance of the sampler. To do so, we start by looking at the runtime and acceptance rate. The runtime of our sampler is 180.876 seconds. The runtime does not give us much information as we do not have anything to compare it with for now, but our acceptance rate of 0.916 may give us more information. A high acceptance rate can both be negative and positive.

- Positive: The proposal distribution is well tuned to the target distribution, especially towards the high-probability regions of the target distribution.
- Negative: The sampler may not be exploring a wide range of moves, which may lead to insufficient exploration of the parameter space. The chain may get stuck in a local region.

To further evaluate the sampler, we turn our attention to the traceplots.

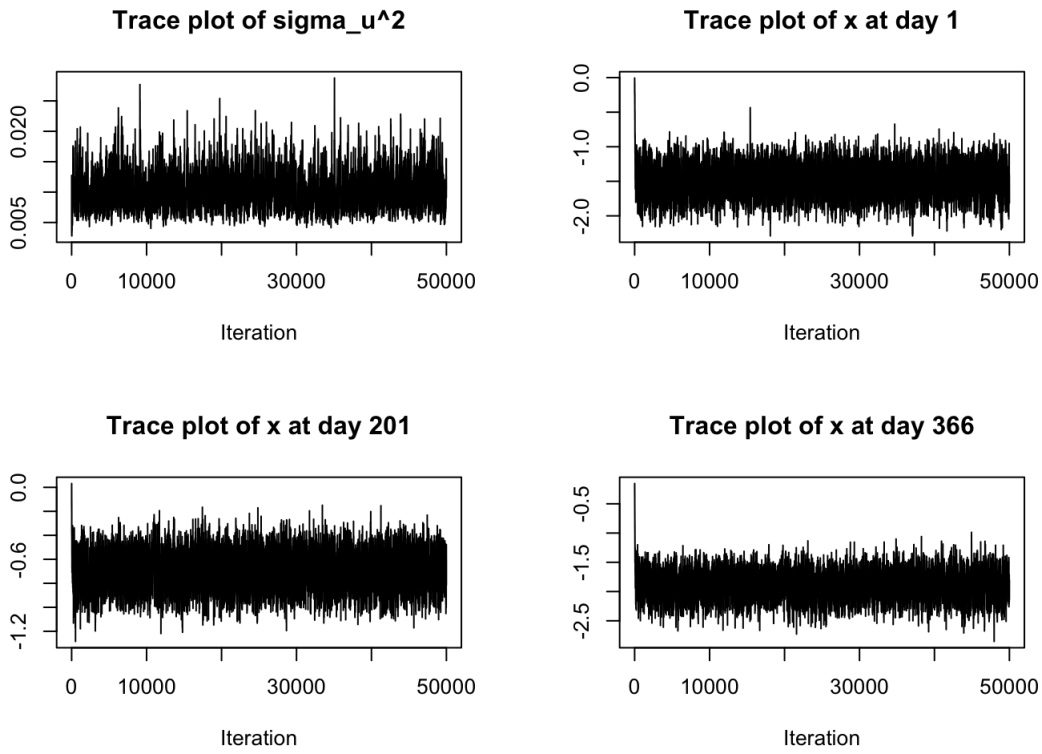

Figure 3: Trace plots of σ_u^2 and x_t for $t = 1, 201, 366$. The traceplots look like white noise, which is a promising sign

Looking at the traceplots in figure 3 for σ_u^2, and x_t for the days 1, 201, and 366, they all seem to behave like white noise. This is the intended behavior and can suggest several things:

- Convergence: If the traceplot looks like white noise, it suggests that the chain is exploring the posterior distribution evenly without getting stuck in any particular region for long periods. This uniform exploration suggests that the chain has converged to the stationary distribution, which is the ideal target posterior distribution.
- Autocorrelation: White noise pattern suggests a low auto correlation between successive samples, which in turn means we are obtaining independent samples from the posterior distribution. This can be helpful for accurate inference.

We now look at histogram plots with 95% credible intervals.

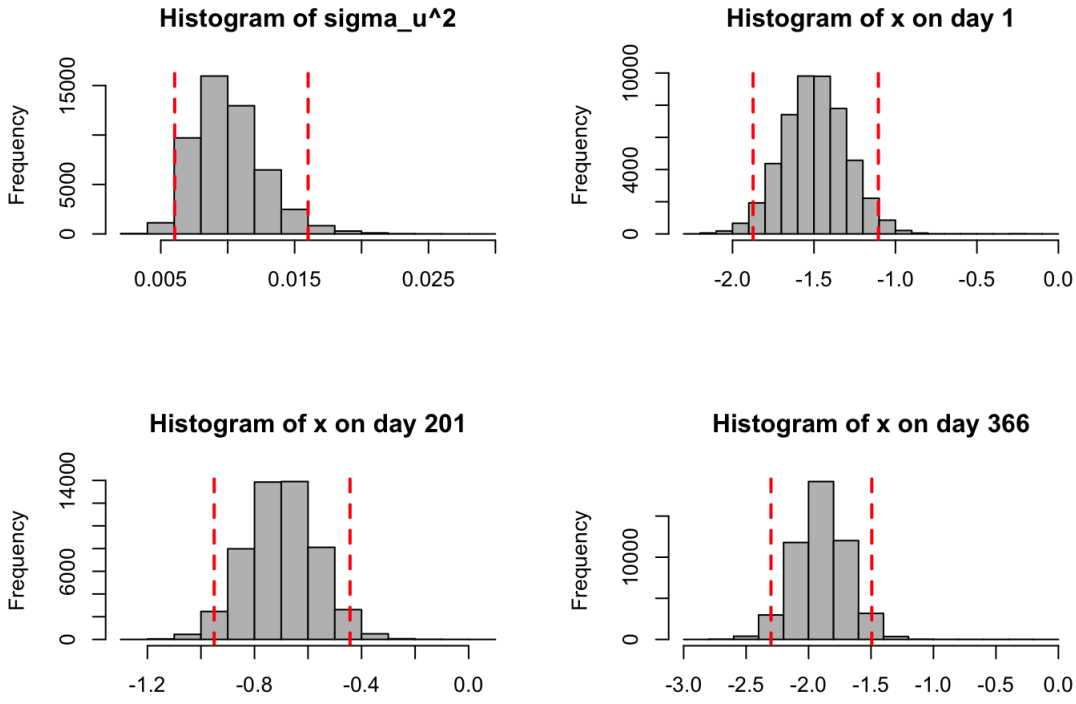

Figure 4: Histogram of σ_u^2 and x_t for $t = 1, 201, 366$ with corresponding 95% credible intervals

The parameter σ_u^2 (Figure 8, top left, seems to follow the density of an inverse gamma distribution, and x_t for the different time points, seems to follows a normal distribution as expected. This is good, as we want the sampled data to follow the distributions of the priors and target distributions.

We also want to plot the sampled values, the original data values, and the 95% CI intervals in one plot. This is to compare how well our sampled values do and look at how they behave compared to the original data.

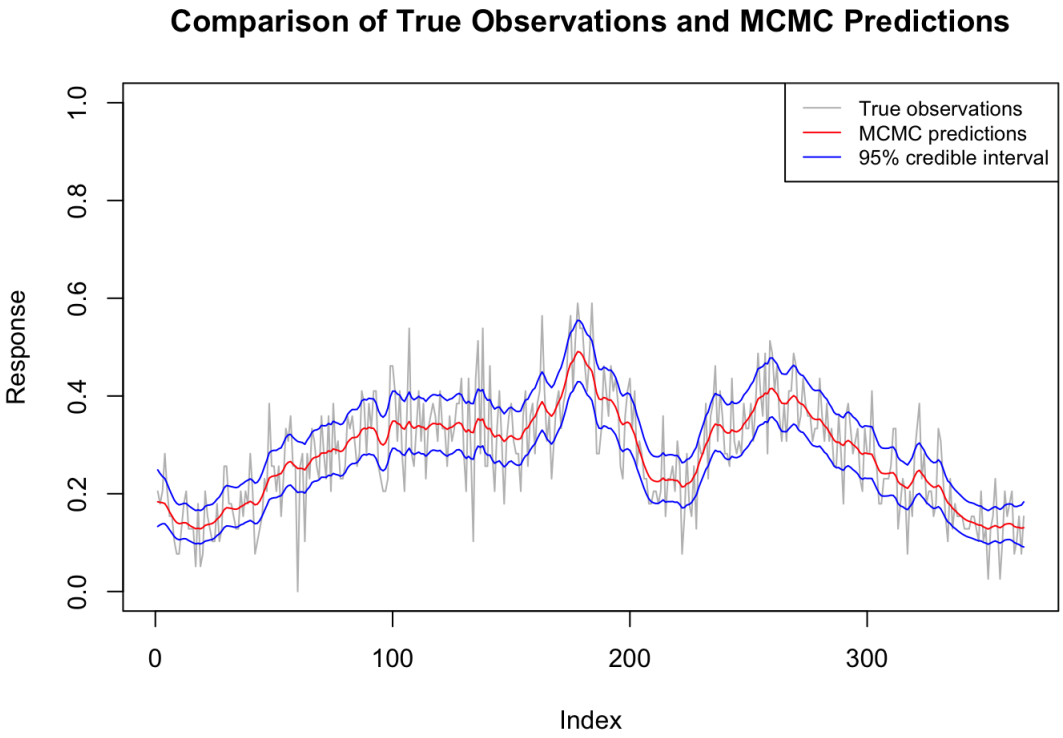

Figure 5: Comparison between sampled (π) and true probabilities.

There are several things one can take from this plot :

- Confidence interval: The 95% interval seems to capture the true values for most days pretty well, except for a few outliers.
- Generated MCMC samples: These seem to be smoother than the real data, which indicates that they capture the trend of the real data pretty well - even though the real data is very volatile. It works as a good smoother and is less prone to noise.

Figure 5 displays the prediction, confidence intervals, and true observation of the data. We can see that predictions fit very well with the data. The vast majority of the points fall in the 95% credible interval.

g)

We want to explore if blocking can generate an algorithm to further improve our performance and predictions.

We want to divide our data into blocks, each of size M, and we would like to experiment with different values of M to see which value is optimal.

To begin, we need to determine the Precision matrix for different cases - we have 3. We know that without blocking the matrix has the form:

$$\mathbf{Q} = \frac{1}{\sigma_u^2}\begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

and:

$$\boldsymbol{\mu}_{A|B} = -\mathbf{Q}_{AA}^{-1}\mathbf{Q}_{AB}\boldsymbol{x}_B$$

Let us now consider these 3 cases:

- Block at the beginning, $a = 1$. The block then includes the first parameter.

$$\mathbf{Q}_{II} = \frac{1}{\sigma_u^2}\begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{pmatrix}$$

Hence:

$$\boldsymbol{\mu}_{I|-I} = \mathbf{Q}_{II}^{-1}[, M]\boldsymbol{x}_{b+1}$$

- Block in the middle, $a > 1, b < 366$. This block is not touching any boundaries, so the precision matrix becomes:

$$\mathbf{Q}_{II} = \frac{1}{\sigma_u^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{pmatrix}$$

Hence:

$$\boldsymbol{\mu}_{I|-I} = (\mathbf{Q}_{II}^{-1}[, 1], \mathbf{Q}_{II}^{-1}[, M]) \cdot (x_{a-1}, x_{b+1})$$

- Block at the end, $b = 366$ is similar to the first case, but has a 1 in the bottom right corner.

$$\mathbf{Q}_{II} = \frac{1}{\sigma_u^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Hence:

$$\boldsymbol{\mu}_{I|-I} = \mathbf{Q}_{II}^{-1}[,1]x_{a-1}$$

The method of proceeding is to draw samples from a standard normal distribution. We use the fact that if $x \sim \mathcal{N}(0, \mathbf{I})$ then $y = \mu + \mathbf{L}x \sim \mathcal{N}(\mu, \mathbf{L}\mathbf{L}^T)$. Here, L is the lower triangular Cholesky decomposition of Q. We make this decomposition as it makes the calculation more efficient.

We first start by determining which value of M we would like to use for the rest of our simulations. We finding simulating acceptance rates and run times as a function of M:

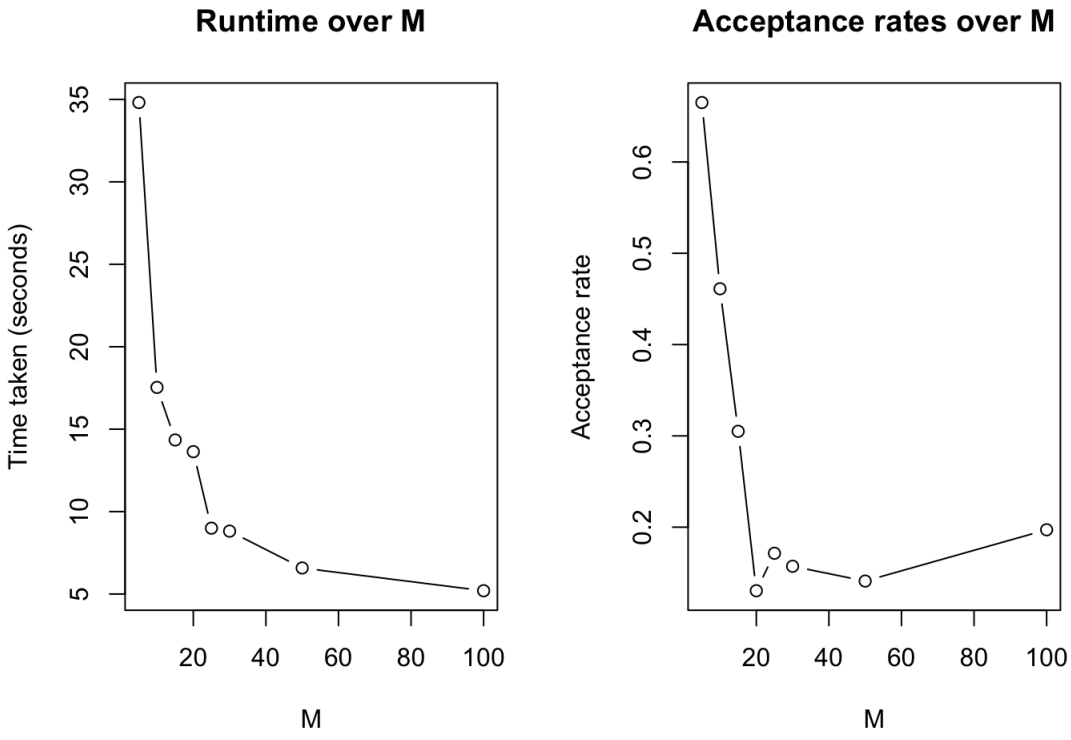

Figure 6: Plots of runtime and acceptance rates for different M

From this graph, we see that for $M = 10$, $M = 15$ or $M = 20$ are pretty good choices for M. We see that there is a trade-off between acceptance rate and runtime. We would like to choose M such that the runtime is low, and the acceptance rate is high. The value that stands out is then $M = 10$.

We now run the algorithm for $M = 10$ with 50000 iterations, and plot the traceplot and the histograms just as for the "normal" MCMC-sampler.

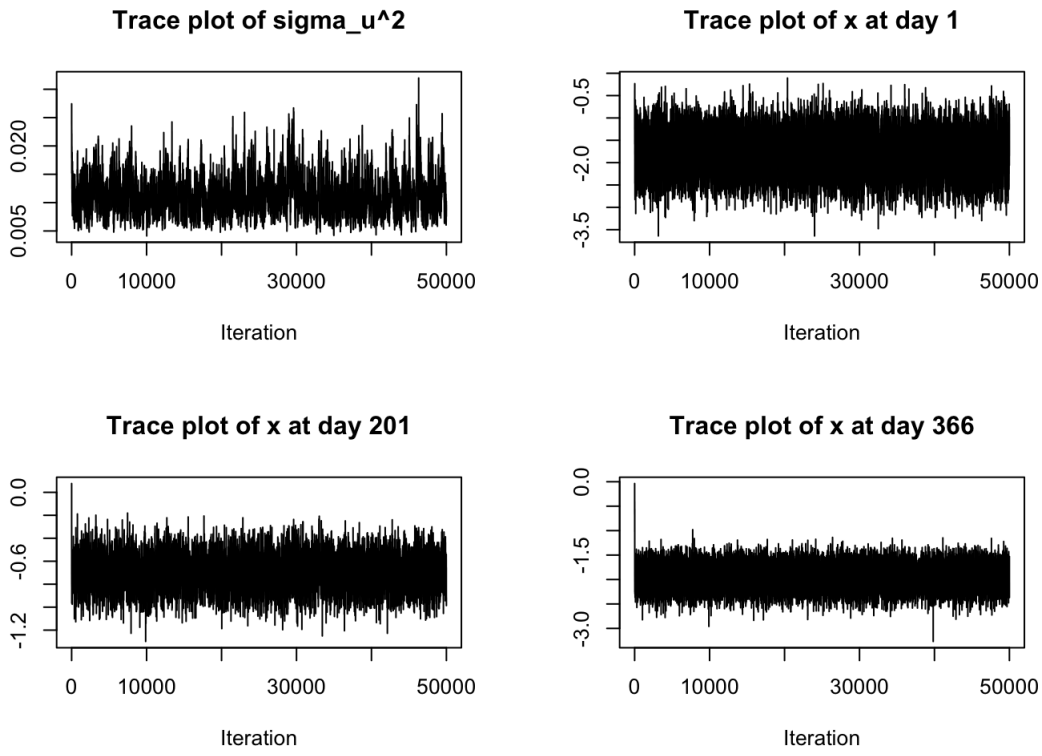

Figure 7: Trace plots of σ_u^2 and x_t for $t = 1, 201, 366$. The traceplots look like white noise, which is a promising sign

Looking at the traceplots, they seem to be white noise, just as the traceplots for the "normal" MCMC-sampler.

Looking at the histograms:

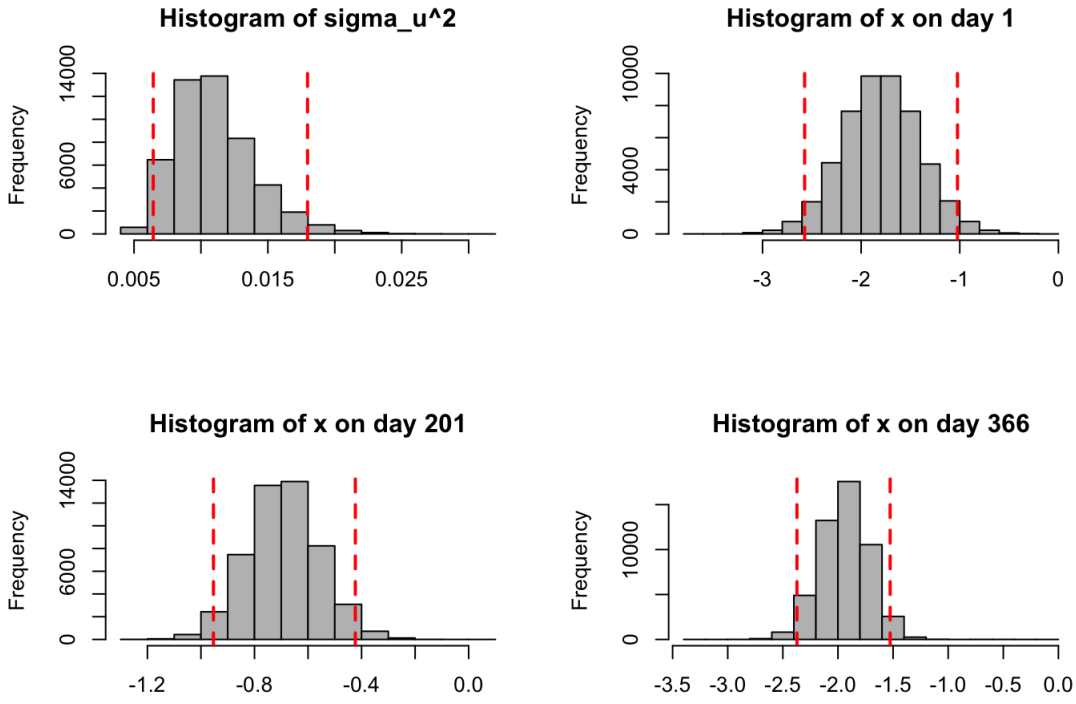

Figure 8: Histogram of σ_u^2 and x_t for $t = 1, 201, 366$ with corresponding 95% credible intervals

With the histograms, there does not seem to be a big difference between the "normal" MCMC-sampler and the blocking MCMC-sampler.

The only difference seems to be the runtime and the acceptance rates. The "normal" sampler had a runtime of 180.876 seconds compared to that of the block sampler 464.596 seconds. The runtime of the blocker is much greater, and that may be a result of the acceptance rate, which is 0.458, compared to the 0.916 of the "normal" sampler. As mentioned, a low acceptance rate does not need to be all negative, as it may explore the parameter space better.

Problem 2

a)

We will now do the same analysis as above, but using the INLA-library. Before we can begin the analysis, just like in the MCMC algorithm, we need to place a prior on σ_u^2. In the MCMC case, a inverse gamma prior was used. Since

INLA places a prior on the log precision rather than the variance, a natural transformation for the inverse gamma prior, is to use the log-gamma distribution as the prior with the same parameters as before. We will also be using a random walk of order 1 on the INLA model, since it is done in the MCMC model. We do also remove the intercept, and thus we do not need to place a prior on it.

We are now going to plot simulation using INLA, and compare it to our previous MCMC simulations.

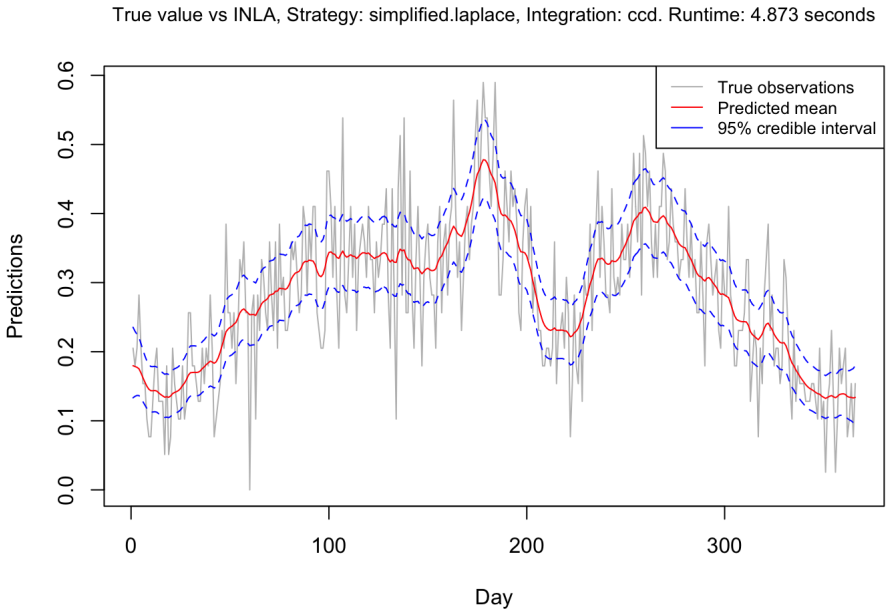

Figure 9: Predicted values using INLA, plotted with true observations and 95% confidence intervals

The results of running the INLA compared to the MCMC is quite interesting. We see that the predicted mean follows the true observations, and just like the MCMC process, INLA captures the trend of the data well. The results from the INLA seem identical to that of the MCMC. The true observations are rarely outside the 95% confident interval.

The biggest difference between INLA and the MCMC is surely the run time. Here, the run time is about 4 seconds, which is much faster than any of the MCMC algorithms used. With such an increase in runtime, and the results seeming almost identical, the INLA stands out to be the best approach for simulation. (Note that the results are not identical - due to randomness).

b)

We can run our INLA function with different arguments for the strategy and integration strategy variable and compare the different outcomes.

In Figure 10, we have used the default settings - the simplified laplace for the strategy and ccd for our integration strategy.

- **Simplified Laplace:** Works well for most cases. It represents a simplified version of the Laplace approximation, based on a series expansion of the standard Laplace approximation. An alternative approach could involve using the normal Laplace approximation.
- **Integration Strategy - CCD:** Short for Central Composite Design, CCD utilizes fewer points in a grid to integrate over, aiming to enhance performance at the potential cost of accuracy. This approach is advantageous for models with large dimensions, where one seeks to achieve reasonable runtime. Given that our model's dimension is 1, we could afford to use more points and cover the entire grid.

So we will be running simulations for different settings, checking:

- Grid vs CCD
- Laplace vs Simplified Laplace

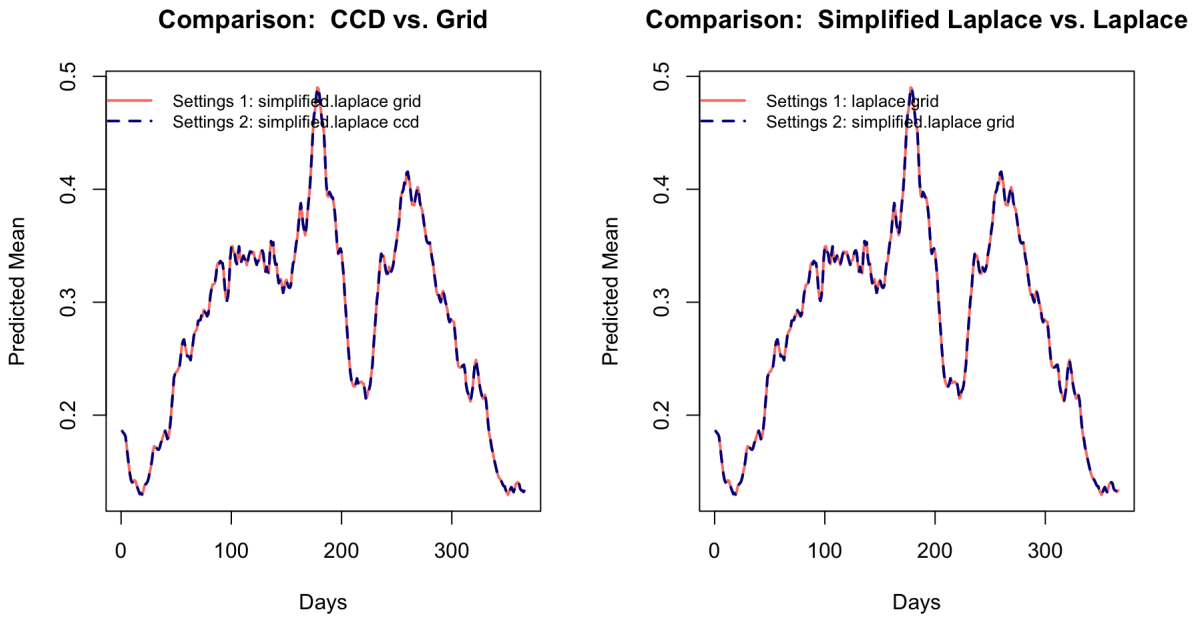

Figure 10: Comparison using different settings - as specified on the plots

The results seem identical, and the run times are also similar. I am sure the differences are greater when the dimensions is greater.

c)

We now want to consider models imposed by the sum-to-zero constraint and that include the intercept.

We do not model $\pi(x_t)$, but instead model $\pi(x_t)$, where $x_t := x_t + \beta_0$, where β_0 is the intercept. In addition, we impose a sum-to-zero constraint, meaning that $\sum_t x_t = 0$.

Let us plot every plot, with and without constraints and intercepts.

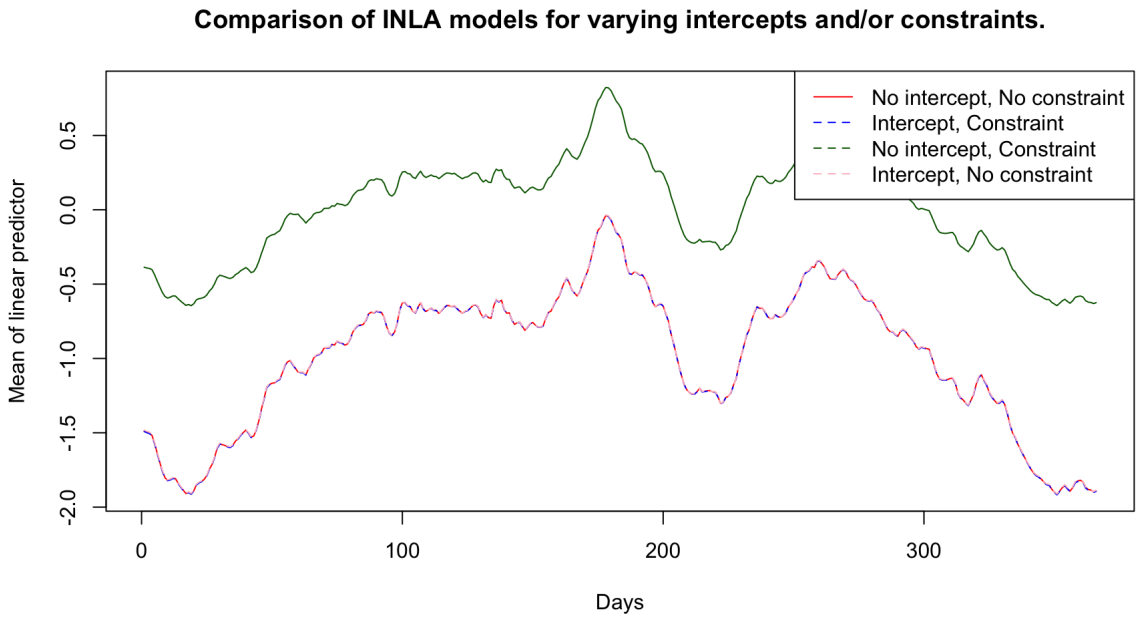

Figure 11: Comparison with varying constraints: sum-to-zero and intercept.

No intercept, no constraint and intercept, constraint and intercept, no constraint seems to be identical. The only plot that differs is the no intercept and constraint plot. The last model is shifted upwards, and it makes sense, as to fulfill the sum-to-zero constraint.

Problem 3

We start off by defining the negative log likelihood function, which calculates the negative log of the joint likelihood $\pi(y, x|\theta)$.

We did not get the RTMB to work, so we optimized using the in-built optim function in R. Here are the results:

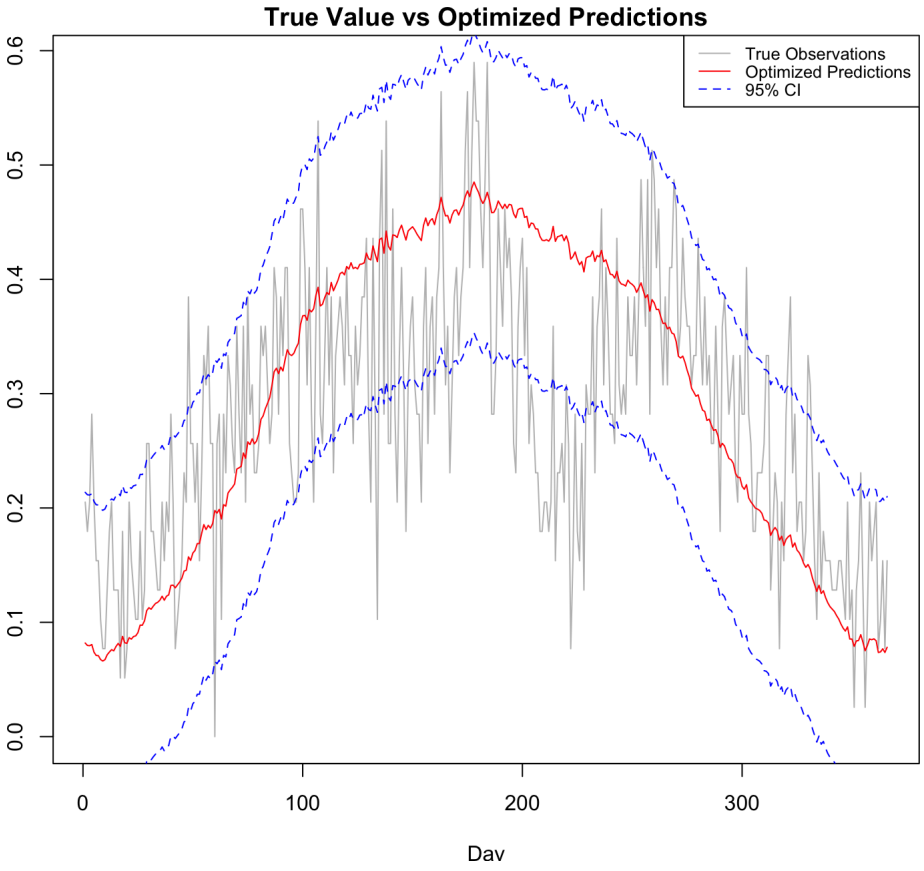

Figure 12: Predictions generated using optimization - plotted with the real data and 95% confidence intervals

Our observations reveal that, in contrast to alternative methodologies, the forecasts generated by this technique predominantly capture the overarching trend, rather than meticulously tracking each minor fluctuation. When compared to the results obtained from INLA and MCMC methods, which exhibit greater variability and more closely mirror the data's fluctuations, this model appears to focus on delineating the general trend. This outcome resonates with the initial discussion in the data exploration segment of the first task, where it was noted that the model effectively encapsulates the seasonal pattern of increased rainfall during summer months and decreased precipitation in winter.