Project 2

Marcus Hjørund & Gard Gravdal

2024-03-15

Contents

1 Problem 1

In this problem we will look at a portion from the Tokyo rainfall dataset, which contains daily rainfall from 1951 to 1989. We consider a response to be whether the amount of rainfall exceeded 1mm over the given time period:

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

for n_t being 10 for $t = 60$ (Feburary 29th) and 39 for all other days. $\pi(x_t)$ is the probability of rainfall exceeding 1mm of days $t = 1, \dots, T$ and $T = 366$. Note that x_t is the logit probability of exceedence and can be obtained from $\pi(x_t)$ via $x_t = \log(\pi(x_t)/(1 - \pi(x_t)))$. We assume conditional independence among the $y_t|x_t$ for all $t = 1, \dots, 366$.

1.1 a)

As we can see in Figure 1 the rain in Tokyo seems to show seasonal trend. There are few occurences of rain more than 1mm in the period December-January, the peak of rainfall is in June-July, with a following drier period in August-September.

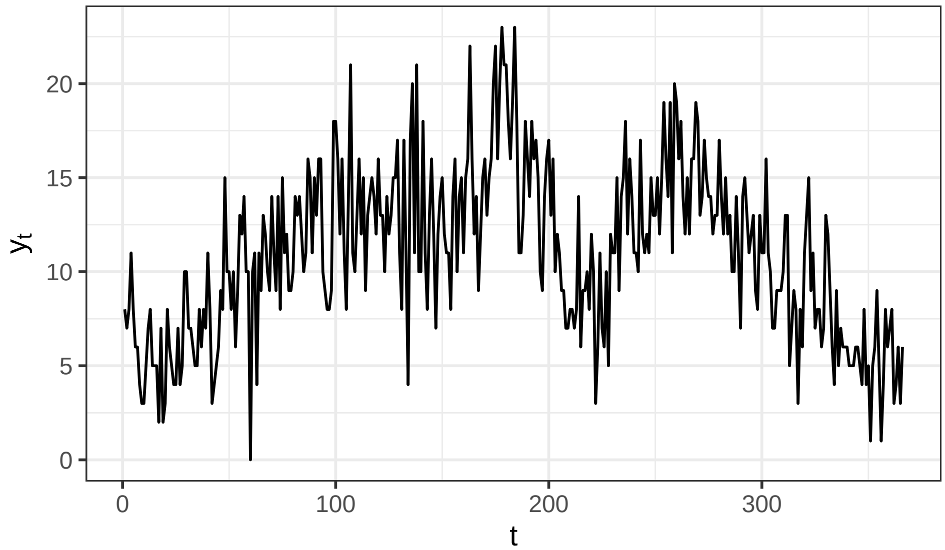


Figure 1: Tokyo rain data

1.2 b)

Next we want to obtain an expression for the likelihood of y_t depending on the parameters $\pi(x_t)$ for $t = 1, \ldots, 366$. This gives the likeihood for the data $\boldsymbol{y}$ given the probabilities $\pi(\boldsymbol{x})$, where we will later update $\boldsymbol{x}$ using our a MCMC algorithm. Since we assume conditional intependence for $y_t|x_t$ we can write the likehood, using the expression for $\pi(x_t)$ given in Equation (1):

$$
\begin{aligned}
L(\pi(\boldsymbol{x})|\boldsymbol{y}) &= \prod_{t=1}^{T} \binom{n_t}{y_t} \left(\frac{\exp(x_t)}{1+\exp(x_t)}\right)^{y_t} \left(1 - \frac{\exp(x_t)}{1+\exp(x_t)}\right)^{n_t - y_t} \\
&= \prod_{t=1}^{T} \binom{n_t}{y_t} \left(\frac{\exp(x_t)}{1+\exp(x_t)}\right)^{y_t} \left(\frac{1}{1+\exp(x_t)}\right)^{n_t - y_t} \\
&= \prod_{t=1}^{T} \binom{n_t}{y_t} \frac{\exp(x_t y_t)}{(1+\exp(x_t))^{n_t}} \qquad (2)
\end{aligned}
$$

1.3 c)

Now we want to apply a Bayesian hierarchical model to the dataset, where we use a random walk of order 1 (RW(1)) to model the trend on a logit scale,

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

for $u_t \overset{iid}{\sim} \mathcal{N}(0, \sigma_u^2)$ so that,

$$p(\boldsymbol{x}|\sigma_u^2) \propto \prod_{t=2}^{T} \frac{1}{\sigma_u} \exp\left\{-\frac{1}{2\sigma_u^2}(x_t - x_{t-1})^2\right\}. \qquad (3)$$

Then we place a inverse gamma prior on σ_u^2 such that,

$$p(\sigma_u^2) = \frac{\beta^\alpha}{\Gamma(\alpha)}(1/\sigma_u^2)^{\alpha+1}\exp(-\beta/\sigma_u^2)$$

for shape and scale α and β. We let $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)^T$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)^T$ and $\boldsymbol{\pi} = (\pi(x_1), \pi(x_2), \ldots, \pi(x_T))^T$.

Now we will derive the posteriori distribution for $\sigma_u^2|\boldsymbol{x}$, which is used in Gibbs sampling in the hybrid sampler in Section 1.6. If we draw a directed acyclical graph for the hierarchical Bayesian model, we see that $\boldsymbol{y}$ is independent of σ_u^2, as all information from $\boldsymbol{y}$ is stored in $\boldsymbol{x}$. From Bayes theorem we know that

$$p(\sigma_u^2|\boldsymbol{y}, \boldsymbol{x}) \propto p(\sigma_u^2)p(\boldsymbol{x}|\sigma_u^2).$$

We then have that

$$\begin{aligned}
p(\sigma_u^2|\boldsymbol{x}, \boldsymbol{y}) &\propto \frac{\beta^\alpha}{\Gamma(\alpha)}(1/\sigma_u^2)^{\alpha+1}\exp(-\beta/\sigma_u^2)\prod_{t=2}^{T}\frac{1}{\sigma_u}\exp\left\{-\frac{1}{2\sigma_u^2}(x_t - x_{t-1})^2\right\} \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}(1/\sigma_u^2)^{\alpha+1}\exp(-\beta/\sigma_u^2)\cdot(\sigma_u^2)^{-\frac{T-1}{2}}\exp\left\{-\frac{1}{2\sigma_u^2}\sum_{t=2}^{T}(x_t - x_{t-1})^2\right\} \\
&\propto (\sigma_u^2)^{-(\alpha+\frac{T-1}{2}+1)}\exp\left\{-\frac{1}{\sigma_u^2}\left(\beta + \frac{1}{2}\sum_{t=2}^{T}(x_t - x_{t-1})^2\right)\right\} \qquad (4)
\end{aligned}$$

which follows an inverse gamma distribution with parameters

$$\alpha^* = \alpha + \frac{T-1}{2} \text{ and } \beta^* = \beta + \frac{1}{2}\sum_{t=2}^{T}(x_t - x_{t-1})^2.$$

1.4 d)

In the Metropolis-Hastings algorithm, which we will implement, we need to find an expression for the acceptance probability α, which in general is given as

$$\alpha(y|x) = \min\left\{1, \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}\right\}$$

where $Q(x|y)$ is a proposed conditional probability mass function (pmf), $\pi(x)$ is the probability of being in state x, and the transition probability is given as

$$p(y|x) = \begin{cases} Q(y|x)\alpha(y|x) & \text{for } y \neq x \\ 1 - \sum_{y\neq x} Q(y|x)\alpha(y|x) & \text{for } y = x \end{cases}$$

which is the probability of transitioning to state y given that the current state is x. The expression for α can be derived from the detailed balance equation, $\pi(x)p(y|x) = \pi(y)p(x|y)$ which is true for a time reversible Markov chain.

We will now consider the conditional prior proposal distribution, $Q(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)$, where $\boldsymbol{x}'_{\mathcal{I}}$ is the proposed values for $\boldsymbol{x}_{\mathcal{I}}$, $\mathcal{I} \subseteq \{1, \ldots, 366\}$ is a set of time indices, and $\boldsymbol{x}_{-\mathcal{I}} = \boldsymbol{x}_{\{1,\ldots,366\}\setminus\mathcal{I}}$ is a subset of $\boldsymbol{x}$ that includes all other indices than those in $\mathcal{I}$. Our expression for the acceptance probability is then

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

We can simplify this expression by looking at $p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})$ and $p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})$. From probability theory we know that $P(A|B) = \frac{P(A\cap B)}{P(B)}$, such that we can divide the expression of $p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})$ as follows:

$$
\begin{aligned}
p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = \frac{p(\boldsymbol{x}'_{\mathcal{I}}, \boldsymbol{y}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}{p(\boldsymbol{y}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)} &= \frac{p(\boldsymbol{y}|\boldsymbol{x}'_{\mathcal{I}}, \boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2) \cdot p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}{p(\boldsymbol{y}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)} \\
&= \frac{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}'_{\mathcal{I}}) \cdot p(\boldsymbol{y}_{-\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}) \cdot p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}{p(\boldsymbol{y}|\boldsymbol{x}_{-\mathcal{I}})}
\end{aligned}
$$

where we have used that $\boldsymbol{y}$ is independent of σ_u^2 and that $\boldsymbol{x}_{\mathcal{I}}$ and $\boldsymbol{x}_{-\mathcal{I}}$ are disjoint sets, and $y_t|x_t$ are conditionally independent. We perform a similar calculation for $p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})$ and obtain

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

We can then simplify $\alpha(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})$, which becomes

$$
\alpha(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = \min\left\{1, \frac{p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}{p(\boldsymbol{x}_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})p(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2)}\right\} = \min\left\{1, \frac{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}'_{\mathcal{I}})}{p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{x}_{\mathcal{I}})}\right\}. \tag{5}
$$

We can then see that $\alpha(\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})$ becomes a ratio of likelihoods for the proposed values $\boldsymbol{x}'_{\mathcal{I}}$ against the existing values $\boldsymbol{x}_{\mathcal{I}}$.

1.5 e)

We note that the density specified by (3) is improper, for example for $T = 2$, density takes the shape of an infinite 'Gaussian ridge' centered around the line given by $x_1 = x_2$. Equation (3) can be rewritten as

$$
p(\boldsymbol{x}|\sigma_u^2) \propto \exp\left\{-\frac{1}{2}\boldsymbol{x}^T \mathbf{Q}\boldsymbol{x}\right\} \tag{6}
$$

which resembles a multivariate normal density but where the improperiety of the density translates into the precision matrix

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

having one zero eigenvalue. We partition the components of $\boldsymbol{x}$ into two subvectors writing

$$
\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_A \\ \boldsymbol{x}_B \end{pmatrix},
$$

and partitioning the precision matrix in the same way as

$$
\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{AA} & \mathbf{Q}_{AB} \\ \mathbf{Q}_{BA} & \mathbf{Q}_{BB} \end{pmatrix}.
$$

We then want to derive the conditional distribution of $\boldsymbol{x}_A$ conditional on $\boldsymbol{x}_B$. We know from before that the conditional density is always proportional to the joint (improper) density. We assume without loss of generality that $\boldsymbol{\mu} = \mathbf{0}$ and find that

$$
\begin{aligned}
f_{\boldsymbol{x}_A|\boldsymbol{x}_B}(\boldsymbol{x}_A) &\propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_A^T, \boldsymbol{x}_B^T)\begin{bmatrix}\mathbf{Q}_{AA} & \mathbf{Q}_{AB}\\ \mathbf{Q}_{BA} & \mathbf{Q}_{BB}\end{bmatrix}\begin{pmatrix}\boldsymbol{x}_A\\ \boldsymbol{x}_B\end{pmatrix}\right\}\\
&= \exp\left\{-\frac{1}{2}(\boldsymbol{x}_A^T\mathbf{Q}_{AA}\boldsymbol{x}_A + \boldsymbol{x}_B^T\mathbf{Q}_{BA}\boldsymbol{x}_A + \boldsymbol{x}_A^T\mathbf{Q}_{AB}\boldsymbol{x}_B + \boldsymbol{x}_B^T\mathbf{Q}_{BB}\boldsymbol{x}_B)\right\}\\
&\propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_A - \boldsymbol{\mu}_{A|B})^T\mathbf{Q}_{AA}(\boldsymbol{x}_B - \boldsymbol{\mu}_{A|B})\right\}\\
&\propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_A^T\mathbf{Q}_{AA}\boldsymbol{x}_A - \boldsymbol{\mu}_{A|B}^T\mathbf{Q}_{AA}\boldsymbol{x}_A - \boldsymbol{x}_A^T\mathbf{Q}_{AA}\boldsymbol{\mu}_{A|B})\right\}
\end{aligned}
$$

where we only consider terms involving $\boldsymbol{x}_A$, which is random in this case, and we use the fact that we can write the terms involving $\boldsymbol{x}_A$ in the second line in quadratic form, completing the square. We then equate the coefficients in the last line to the terms in the second line to find our expressions for $\boldsymbol{\mu}_{A|B}$ and $\mathbf{Q}_{A|B}$. We have

$$
-\mathbf{Q}_{AA}\boldsymbol{\mu}_{A|B} = \mathbf{Q}_{AB}\boldsymbol{x}_B \implies \boldsymbol{\mu}_{A|B} = -\mathbf{Q}_{AA}^{-1}\mathbf{Q}_{AB}\boldsymbol{x}_B \tag{7}
$$

$$
\mathbf{Q}_{A|B} = \mathbf{Q}_{AA}. \tag{8}
$$

These results is used to simulate from $\boldsymbol{x}'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}$ later on when we generate realizations from the random walk.

1.6 f)

In this section we want to implement an MCMC sampler from the posterior distribution $p(\boldsymbol{\pi}, \sigma_u^2|\boldsymbol{y})$ using Metropolis-Hastings steps for individual x_t parameters using the conditional prior, $p(x_t|\boldsymbol{x}_{-t}, \sigma_u^2)$, and Gibbs steps for σ_u^2.

We will begin by deriving the conditional prior distribution using the results from Equations (7) and (8) as the conditional mean and precision (inverse variance) of $x'_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2$. We need to derive the conditional prior distribution for three cases, $t = 1$, $2 \leq t \leq 365$ and $t = 366$. For the case $t = 1$, $\mathcal{I} = \{1\}$, we have for the conditional mean that $-\mathbf{Q}_{\mathcal{I}\mathcal{I}}^{-1} = -\sigma_u^2$, $\mathbf{Q}_{\mathcal{I}-\mathcal{I}} = \frac{1}{\sigma_u^2}(-1, 0, \ldots, 0)$ with dimension $1 \times (T-1)$. Thus the conditional expectation is given as

$$
x_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2 = -\mathbf{Q}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{Q}_{\mathcal{I}-\mathcal{I}}\boldsymbol{x}_{-\mathcal{I}} = -\sigma_u^2 \cdot \frac{1}{\sigma_u^2}(-1, 0, \ldots, 0)(x_2, x_3, \ldots, x_T)^T = x_2. \tag{9}
$$

Similarly we have for $t = 366$, $\mathcal{I} = \{366\}$, we have

$$
x_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2 = -\sigma_u^2 \cdot \frac{1}{\sigma_u^2}(0, \ldots, 0, -1)(x_1, \ldots, x_{364}, x_{365})^T = x_{365}. \tag{10}
$$

Then finally for the interior points, $2 \leq t \leq 365$, $\mathcal{I} = \{2\}, \{3\}, \ldots, \{365\}$, we have $-\mathbf{Q}_{\mathcal{I}\mathcal{I}}^{-1} = -\frac{\sigma_u^2}{2}$ and $\mathbf{Q}_{\mathcal{I}-\mathcal{I}} = \frac{1}{\sigma_u^2}(0, \ldots, 0, -1, -1, 0, \ldots, 0)$ with dimension $1 \times (T-1)$. We then calculate the conditional expectation as

$$
x_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2 = -\frac{\sigma_u^2}{2} \cdot \frac{1}{\sigma_u^2}(0, \ldots, 0, -1, -1, 0, \ldots, 0)(x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_{366})^T = \frac{1}{2}(x_{t-1} + x_{t+1}). \tag{11}
$$

We perform a similar calculation to calculate the conditional variance. We have that the conditional variance $\boldsymbol{\Sigma}_{\mathcal{I}|-\mathcal{I},\sigma_u^2} = \mathbf{Q}_{\mathcal{I}|-\mathcal{I},\sigma_u^2}^{-1}$. For $t = 1$ and $t = 366$ we have that $\mathbf{Q}_{\mathcal{I}\mathcal{I}}^{-1} = \sigma_u^2$ and for $2 \leq t \leq 365$ we have $\mathbf{Q}_{\mathcal{I}\mathcal{I}}^{-1} = \frac{\sigma_u^2}{2}$. Thus we can summarize the conditional distribution of $x_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}$ as

$$
x_{\mathcal{I}}|\boldsymbol{x}_{-\mathcal{I}}, \sigma_u^2 \sim \begin{cases}\mathcal{N}(x_2, \sigma_u^2) & \text{for } t = 1\\ \mathcal{N}(\frac{1}{2}(x_{t-1} + x_{t+1}), \frac{\sigma_u^2}{2}) & \text{for } t = 2, \ldots, 365\\ \mathcal{N}(x_{365}, \sigma_u^2) & \text{for } t = 366.\end{cases} \tag{12}
$$

This is the distribution we simulate from on the random walk to generate realizations of x_t', which is the proposed value for x at index t. Next we will derive the expression for the acceptance probability $\alpha(x_t'|\boldsymbol{x}_{-t}, \sigma_u^2, \boldsymbol{y})$ for $\mathcal{I} = \{t\}$. We use the expression for the acceptance probability derived in Equation (5). We have

$$\begin{aligned}
\alpha(x_t'|\boldsymbol{x}_{-t}, \sigma_u^2, \boldsymbol{y}) &= \min\left\{1, \frac{p(y_t|x_t')}{p(y_t|x_t)}\right\} = \min\left\{1, \frac{\binom{n_t}{y_t}\frac{\exp(x_t' y_t)}{(1+\exp(x_t'))^{n_t}}}{\binom{n_t}{y_t}\frac{\exp(x_t y_t)}{(1+\exp(x_t))^{n_t}}}\right\} \\
&= \min\left\{1, \exp\left(y_t(x_t' - x_t) \cdot \frac{(1+\exp(x_t))^{n_t}}{(1+\exp(x_t'))^{n_t}}\right)\right\} \\
&= \min\left\{1, \exp\left(y_t(x_t' - x_t) - n_t \ln\left(\frac{1+\exp(x_t')}{1+\exp(x_t)}\right)\right)\right\}
\end{aligned}$$

In the algorithm, we will for each $t \in \{1, \dots, 366\}$, use the Metropolis-Hastings algorithm, using the following algorithm,

Hybrid sampler Metropolis-Hastings:

Initialize $x = x_0$ and $\sigma_u^2 \sim p(\sigma_u^2)$

repeat n times:

for $t = 1, \dots, 366$:

generate $x_t' \sim p(\sigma_u^2|\boldsymbol{x}_{-t})$

$\alpha \leftarrow \min\left\{1, \exp\left(y_t(x_t' - x_t) - n_t \ln\left(\frac{1+\exp(x_t')}{1+\exp(x_t)}\right)\right)\right\}$

generate $u \sim \text{unif}(0, 1)$

if $(u < \alpha)$:

$x_t \leftarrow x_t'$

else

$x_t \leftarrow x_t$

end for

$\beta^* = \beta + \frac{1}{2}\sum_{t=2}^{T}(x_t - x_{t-1})^2$

generate $\sigma_u^2 \sim \text{InvGamma}(\alpha + \frac{T-1}{2}, \beta^*)$

end for

Return samples $x_1, ..., x_T$

```r
# Problem 1
library(matrixStats)
library(MASS)

expit <- function(x){
  return(exp(x)/(1 + exp(x)))
}

# f)

# Defining global variables
alpha = 2
beta = 0.05
```

```r
N = 50000

alpha_prob <- function(x_prop, x, t){
  # Calculates acceptance probability
  y = rain$n.rain[t]
  n = rain$n.years[t]
  ans = exp(y*(x_prop - x) - n * log((1+exp(x_prop))/(1+exp(x))))
  return(min(1,ans))
}

MH_step <- function(x, sigma2, accept_count, mu, t){
  # Calculates one step of Metropolis-Hastings
  # All inputs are numbers
  x_prop <- rnorm(1, mu, sqrt(sigma2))

  # Probability of acceptance
  alpha_accept <- alpha_prob(x_prop, x, t)
  u <- runif(1)

  if (alpha_accept > u){
    x <- x_prop
    accept_count = accept_count + 1
  }
  return(list(x = x, accept_count = accept_count))

}

moments_prop <- function(x,t,sigma2, T_max = 366){
  # Returns mu proposed as defined in the text
  # Input x is (1xT_max), t is a number
  if (t == 1){
    return(list(mu= x[2],sigma2 = sigma2 ))
  }
  else if (t == T_max) {
    return(list(mu = x[T_max-1], sigma2 = sigma2))
  }
  else{
    return(list(mu = 0.5*(x[t-1] + x[t+1]), sigma2 = sigma2/2))
  }
}

MCMC <- function(n = N, T_max = 366){
  # Time start
  time = proc.time()[3]
  # Initialization
  accept_count = 0
  x <- rep(1,T_max*n)
  x <- matrix(x, nrow = n, ncol = T_max)
  sigma2 <- rep(1,n)
  sigma2[1] <- rgamma(1, alpha, scale = beta)^-1
  for (i in 2:n){
    # One iteration of MCMC:
    old_sigma2 = sigma2[i-1]
```

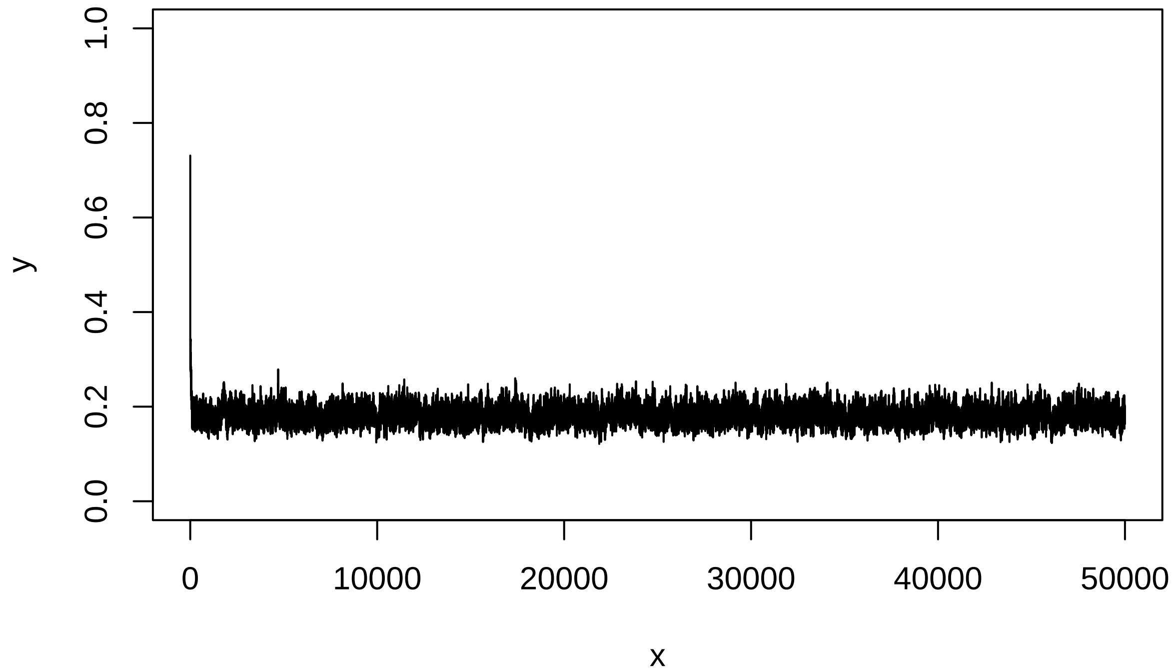
```
    for (t in 1:T_max){
      param = moments_prop(x[i-1,],t,old_sigma2, T_max = T_max)
      mu_temp = param$mu
      sigma2_temp = param$sigma2
      MH_step_t <- MH_step(x[i-1,t], sigma2_temp, accept_count, mu_temp, t)
      x[i,t] <- MH_step_t$x
      accept_count <- MH_step_t$accept_count
    }

    # Gibbs step, updating sigma2
    z = beta + 0.5*sum((x[i,-T_max] - x[i, -1])^2)
    sigma2[i] <- 1/rgamma(1,alpha + (T_max-1)/2, scale = z)
  }
  accept_rate = accept_count/ (n * T_max)
  total_time = proc.time()[3] - time

  return(list(x = x, total_time = total_time, accept_rate = accept_rate, sigma2 = sigma2))
}

MCMC_run <- MCMC()

MCMC_run$total_time
```

```
## elapsed
##  445.65
```

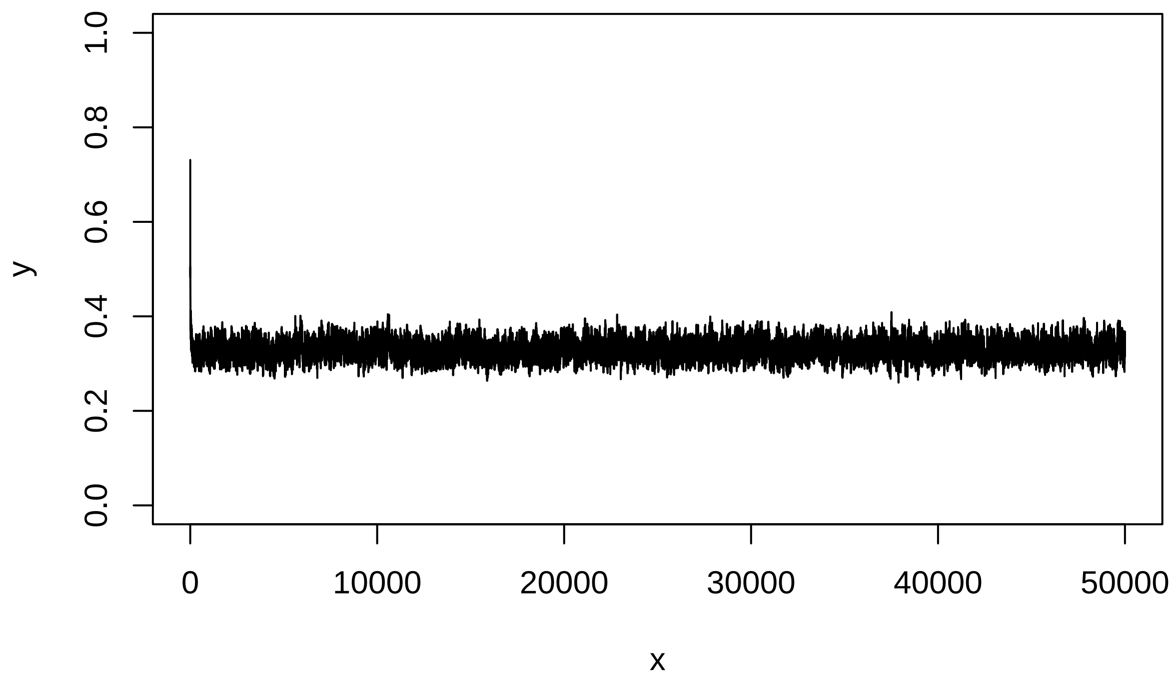
```
MCMC_run$accept_rate
```

```
## [1] 0.9327913
```

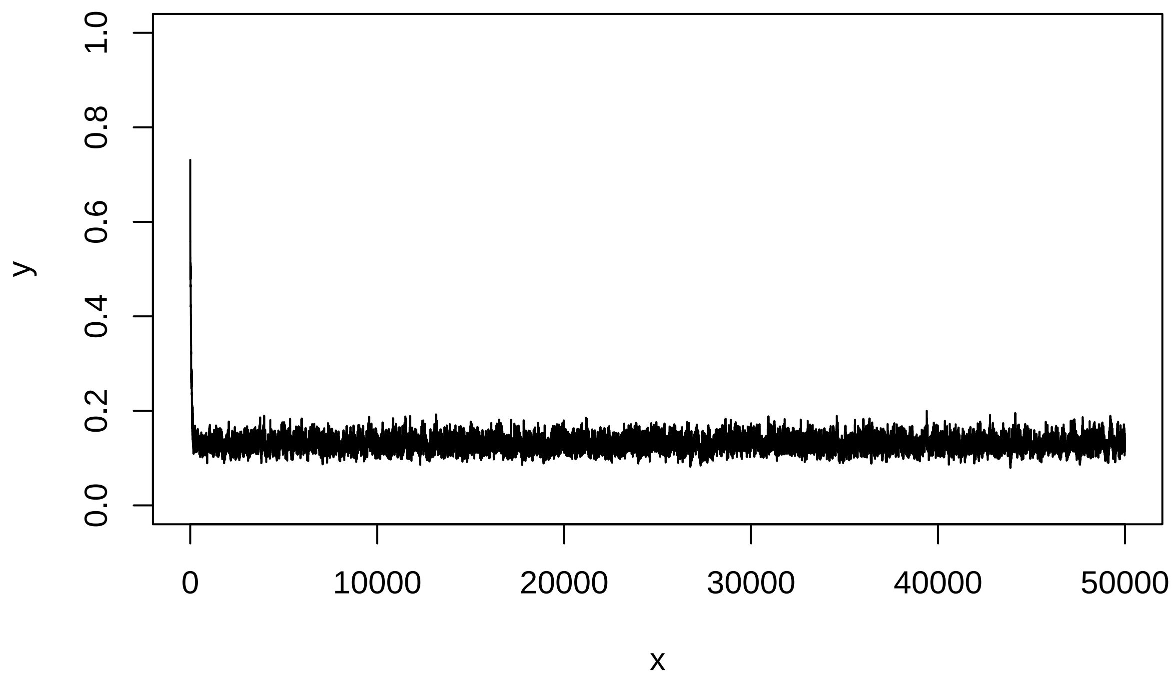
```
burnin <- ceiling(N/10)
interval <- burnin:N
x_vec = seq(1,N)

plot(x_vec, expit(MCMC_run$x[,1]), ylim = c(0,1), ylab = "y", xlab = "x", type = "l")
```



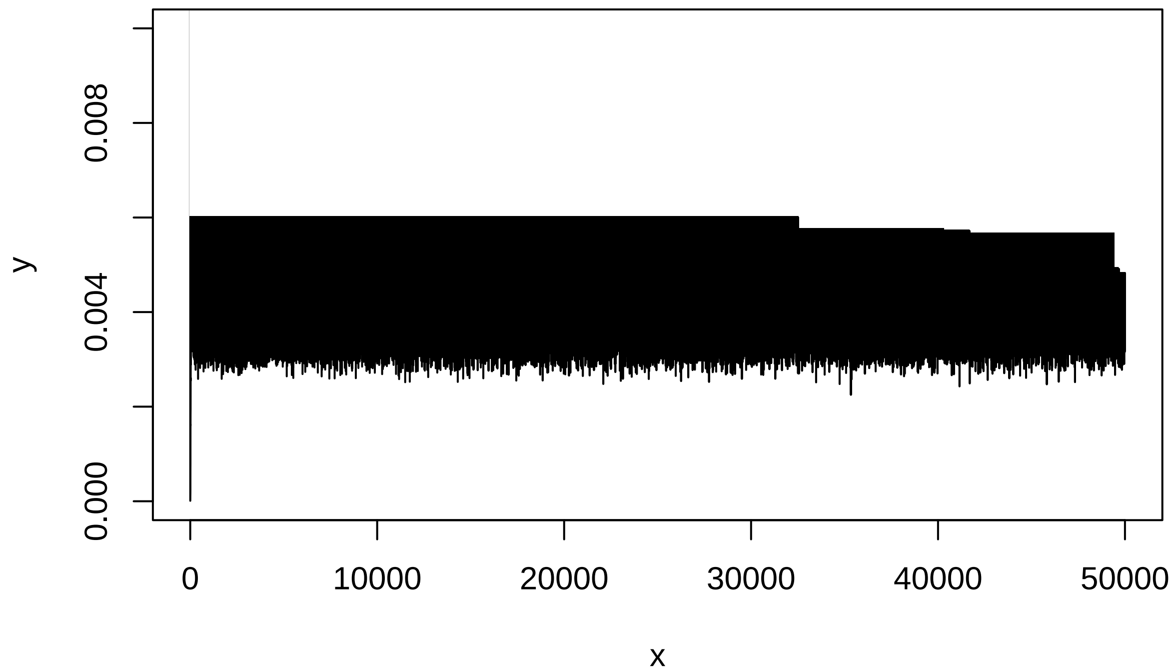

```r
plot(x_vec, expit(MCMC_run$x[,201]), ylim = c(0,1), ylab = "y", xlab = "x", type = "l")
```




```r
plot(x_vec, expit(MCMC_run$x[,366]), ylim = c(0,1), ylab = "y", xlab = "x", type = "l")
```

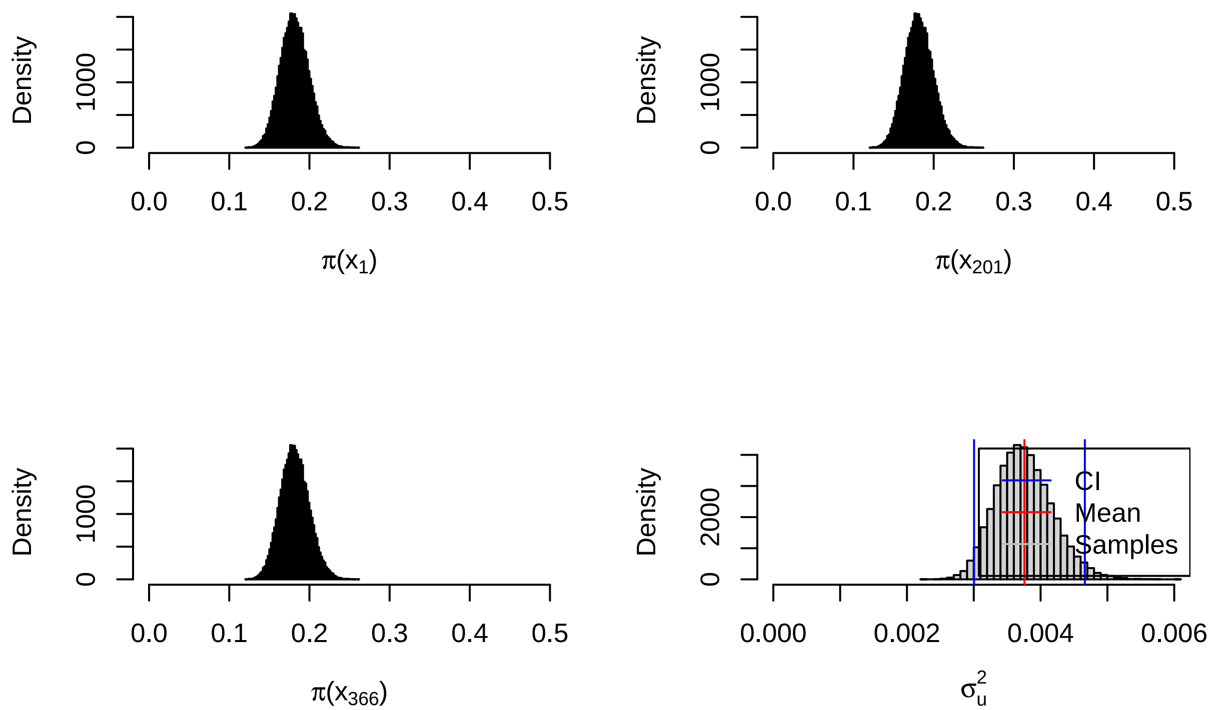



```r
plot(x_vec, MCMC_run$sigma2, ylim = c(0,0.01), ylab = "y", xlab = "x", type = "l")
```

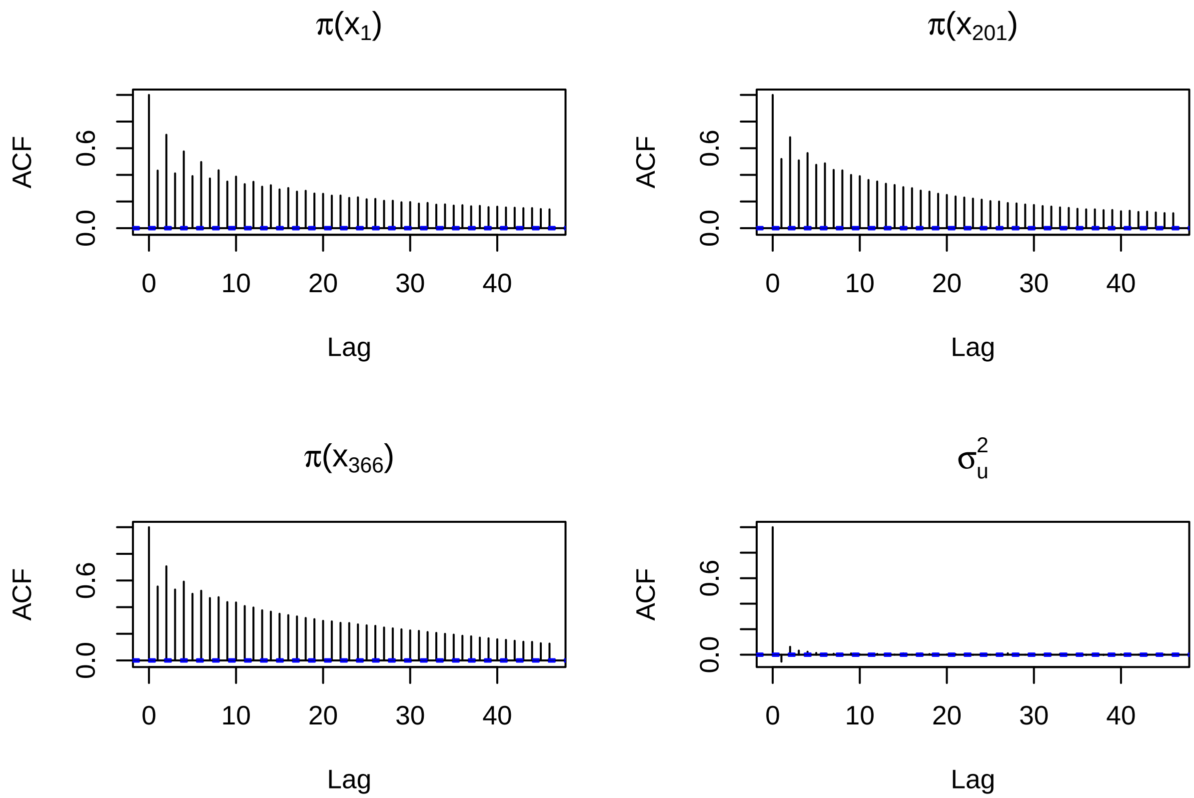



```
# Plotting histograms
# General histogram function
plot_hist <- function(x, xlab = "x", ylab = "Density", breaks = 50, xlim = c(0,1), alpha = 0.05){
  # Plots general histogram with quantiles and mean
  quantiles <- quantile(x, probs = c(alpha/2, 1 - alpha/2))
  hist(x, breaks = breaks, xlab = xlab, ylab = ylab, xlim = xlim, main = NULL)
  abline(v = quantiles[1], col = "blue")
  abline(v = quantiles[2], col = "blue")
  abline(v = mean(x), col = "red")
  legend("right",legend = c("CI", "Mean", "Samples"), col = c("blue", "red", "grey"), lty = 1)
}
```

```
# Plots
xlab1 = expression(paste(pi, "(x"[1],")"))
xlab2 = expression(paste(pi, "(x"[201],")"))
xlab3 = expression(paste(pi, "(x"[366],")"))
xlab4 = expression(paste(sigma[u]^2))
par(mfrow = c(2, 2), mar = c(5,4,4,2))
hist(expit(MCMC_run$x[,1])[interval], breaks = 50, xlab = xlab1,ylab = "Density", xlim = c(0,0.5), main
hist(expit(MCMC_run$x[,1])[interval], breaks = 50, xlab = xlab2,ylab = "Density", xlim = c(0,0.5), main
hist(expit(MCMC_run$x[,1])[interval], breaks = 50, xlab = xlab3,ylab = "Density", xlim = c(0,0.5), main
plot_hist(MCMC_run$sigma2[interval],xlim = c(0,0.006), xlab = xlab4)
```

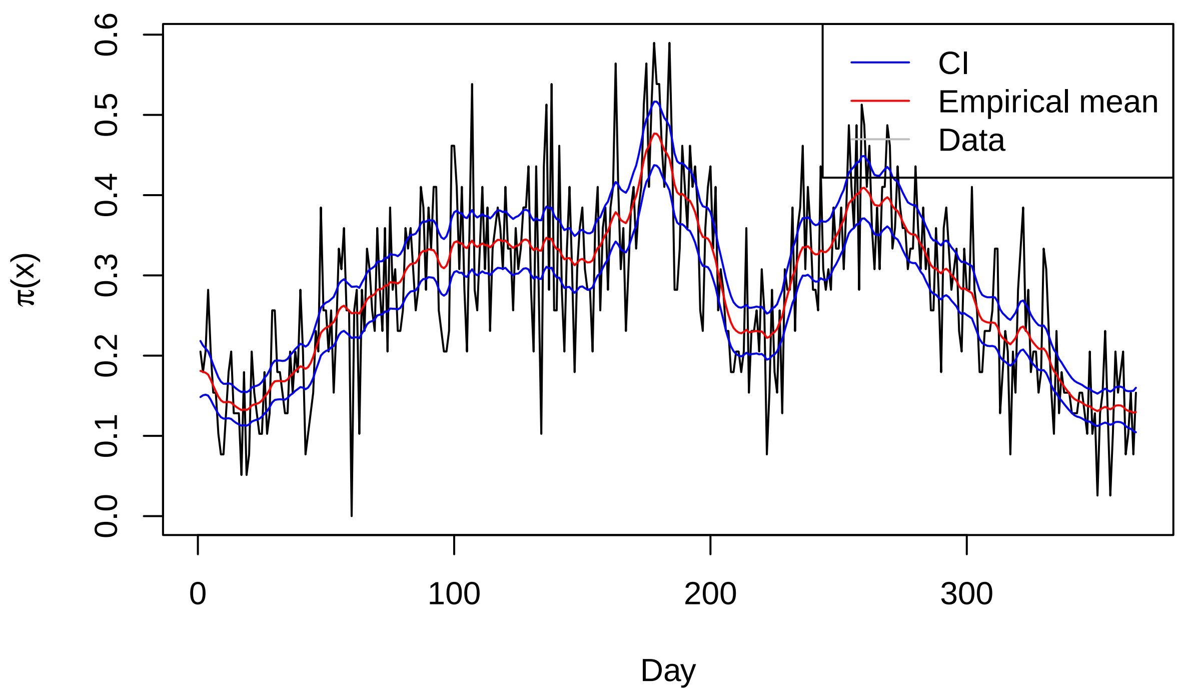



```
par(mfrow = c(1, 1))


# AVFs:
par(mfrow = c(2, 2), mar = c(5,4,4,2))
acf(expit(MCMC_run$x[,1])[interval], main = xlab1)
acf(expit(MCMC_run$x[,201])[interval], main = xlab2)
acf(expit(MCMC_run$x[,366])[interval], main = xlab3)
acf(MCMC_run$sigma2[interval], main = xlab4)
```




```
par(mfrow = c(1, 1), mar = c(5,5,1,1))
```

```
mean_x = colMeans(MCMC_run$x[interval,])
pi_data = rain$n.rain/rain$n.years


x_quantiles <- colQuantiles(MCMC_run$x[interval,], probs = c(0.025, 0.975))
par(mfrow = c(1, 1))
plot(1:366,pi_data, type = "l",xlab = "Day", ylab = expression(paste(pi, "(x)")))
lines(expit(mean_x), type = "l", col = "red")
lines(expit(x_quantiles[,1]), type = "l", col = "blue")
lines(expit(x_quantiles[,2]), type = "l", col = "blue")
legend("topright",legend = c("CI", "Empirical mean", "Data"), col = c("blue", "red", "grey"), lty = 1)
```



2 Problem 2

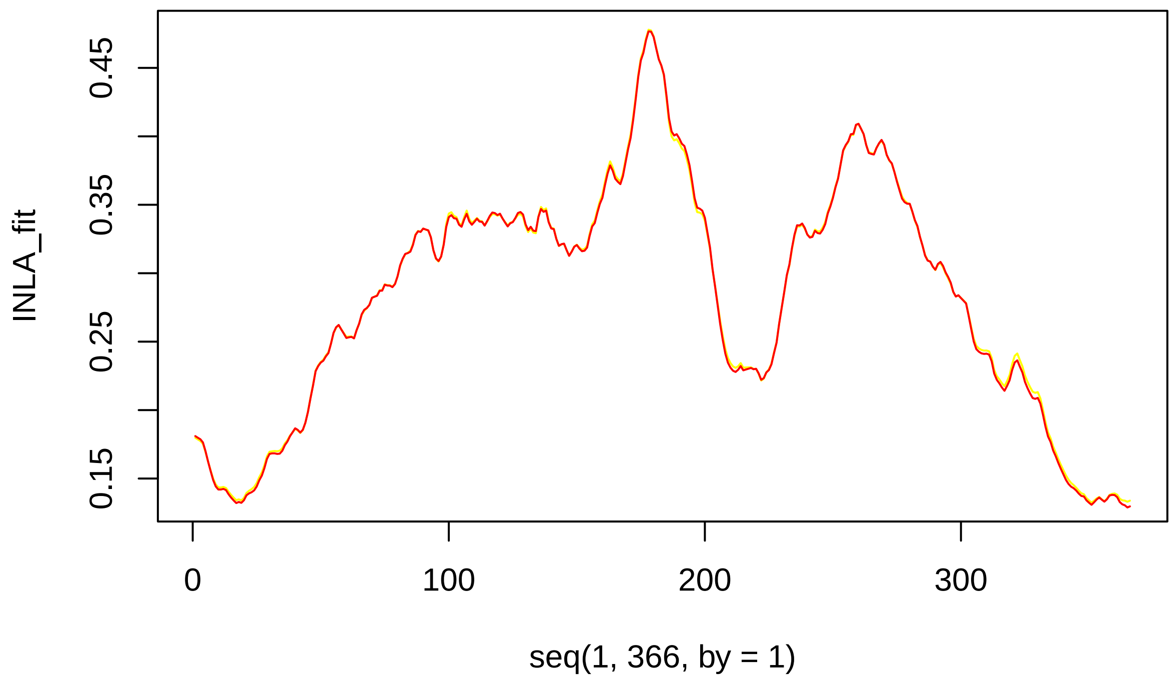
```r
head(rain_data)
```

```
##   t n_t y_t
## 1 1  39   8
## 2 2  39   7
## 3 3  39   8
## 4 4  39  11
## 5 5  39   8
## 6 6  39   6
```

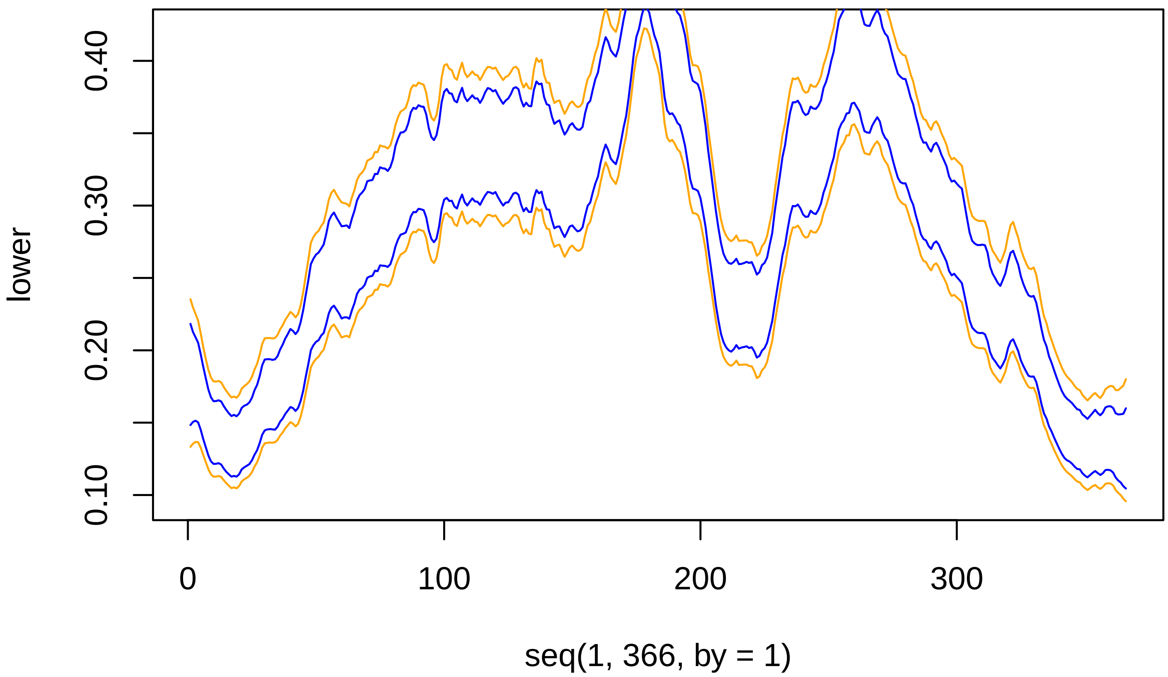
```r
#install.packages("INLA", repos = c(getOption("repos"), INLA ="https://inla.r-inla-download.org/R/stabl
library("INLA")
t <- proc.time()[3]
control.inla = list(strategy="simplified.laplace", int.strategy="ccd")
mod <- inla(n.rain ~ -1 + f(day, model="rw1", constr=FALSE),
data=rain, Ntrials=n.years, control.compute=list(config = TRUE),
family="binomial", verbose=TRUE, control.inla=control.inla)
run_time <- proc.time()[3]-t
run_time
```

```
## elapsed
##     0.8
```

```
INLA_fit <- mod$summary.fitted.values$mean
lower <- mod$summary.fitted.values$`0.025quant`
upper <- mod$summary.fitted.values$`0.975quant`
plot(seq(1,366, by = 1), INLA_fit,type = "l", col = "yellow")
lines(expit(mean_x), type = "l", col = "red")
```




```
plot(seq(1,366, by = 1), lower, type = "l", col  = "orange")
lines(seq(1,366, by = 1), upper, type = "l", col  ="orange")
lines(expit(x_quantiles[,1]), type = "l", col = "blue")
lines(expit(x_quantiles[,2]), type = "l", col = "blue")
```




```r
?control.fixed
?f
inla.doc("rw1")
inla.doc("X")
```