Project 2

Computer Intensive Statistical Methods

Erling Fause Steen og Christian Oppegård Moen

08 03 2022

Contents

Introduction

The Tokyo rainfall dataset contain the amount of rainfall for each of the 366 days (including February 29.) for several years. We will consider a portion of this dataset, specifically from $1951-1989$, such that for each day $t \neq 60$ we have $n_t = 39$ observations and for $t = 60$, February 39, we have $n_t = 10$ observation. For each day we have the response $y_t = 0, 1, 2, ..., n_t$ being the amount of times the rainfall exceeded 1mm over the given period, given by

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

Here, $\pi(\tau_t)$ is the probability of rainfall exceeding 1mm and τ_t is the logit probability of exceedence. For this project we assume conditional independence among the $y_t|\tau_t \ \forall t = 1, 2, ..., 366$.

We will investigate the accuracy and computational speed of a random walk implementation containing Metropolis-Hasting and Gibbs steps for specific parameters compared to the built in method `INLA` of R.

Problem 1

a)

In Figure 1 we see the response.

```
load("./rain.rda")
## Plotting the data
head(rain)
```

```
##   day n.years n.rain
## 1   1      39      8
## 2   2      39      7
## 3   3      39      8
## 4   4      39     11
## 5   5      39      8
## 6   6      39      6
```

```
ggplot(data = rain, mapping = aes(x = day, y = n.rain)) + geom_line() + xlab("Day") +
    ylab("Number of days with more than 1 mm rain")
```

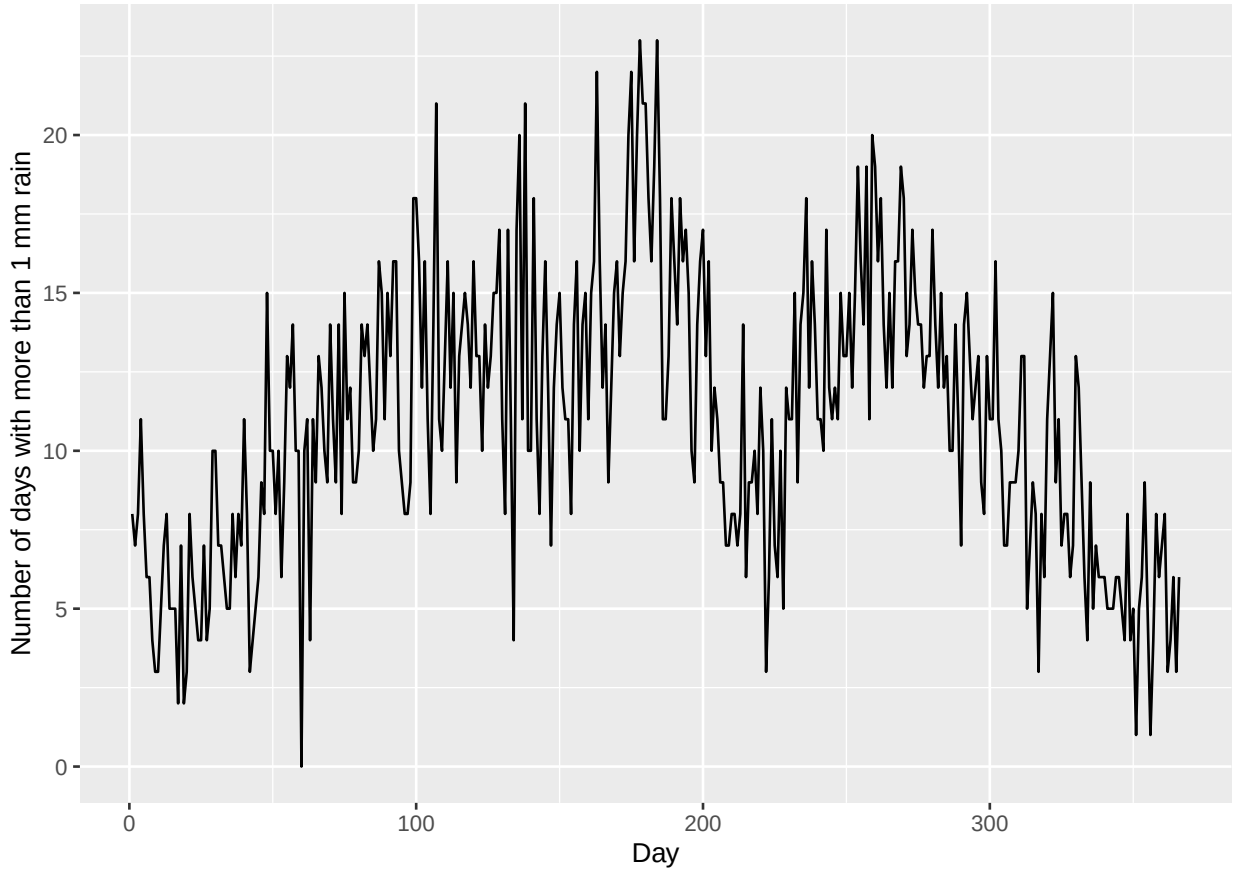


Figure 1: The Tokyo Rainfall dataset

We start by plotting the amount of rainfall against each day.

From the plot, we can see that there are fewer days in the start of the year and in the end of the year with an amount of rainfall over 1 mm. This is in January and December. The number of days steadily increases until the beggining of the summer which seems to be the period with the most days with an amount of rainfall over 1 mm. Then, the amount of days decreases during july and august before increasing during the autumn. This is somewhat consistent with the results we get from googling the amount of days with

precipitation in Tokyo, where we can see that June and September are the months with the most days with rainfall and December and January are the month with the most.

b) Likelihood

The likelihood of Equation (1) is given by

$$\begin{aligned} L(\pi(\tau_t)) &= \prod_{i=1}^{T} \begin{pmatrix} n_t \\ y_t \end{pmatrix} \pi(\tau_t)^{y_t}(1-\pi(\tau_t)^{n_t-y_t} \\ &\propto \prod_{i=1}^{T} \pi(\tau_t)^{y_t}(1-\pi(\tau_t)^{n_t-y_t} \\ &= \prod_{t=1}^{T} \left(\tfrac{\exp(\tau_t)}{1+\exp(\tau_t)}\right)^{y_t} \left(1-\tfrac{\exp(\tau_t)}{1+\exp(\tau_t)}\right)^{n_t-y_t}, \end{aligned}$$

where $y_t = 1, 2, ..., 39$ and $n_t = 39$ for $t \neq 60$, and $y_t = 1, 2, ..., 10$ and $n_t = 10$ for $t \neq 60$.

c) Posterior

$$\begin{aligned} P(\sigma^2|\tau, y) &= \frac{P(\sigma_u^2, \tau, y)}{P(\tau, y)} \\ &\propto P(y|\sigma_u^2, \tau)P(\sigma_u^2, \tau) \\ &= P(y|\sigma_u^2, \tau)P(\tau|\sigma_u^2)P(\sigma_u^2) \\ &= \underbrace{\prod_{t=1}^{T} \left(\frac{\exp(\tau_t)}{1+\exp(\tau_t)}\right)^{y_t} \left(1-\frac{\exp(\tau_t)}{1+\exp(\tau_t)}\right)^{n_t-y_t}}_{\text{Constant w.r.t. } \sigma^2} \cdot \\ &\quad \prod_{t=1}^{T} \frac{1}{\sigma_u} \exp\left\{\frac{1}{2\sigma_u^2}(\tau_t - \tau_{t-1})^2\right\} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma_u^2}\right)^{\alpha+1} exp\left\{-\frac{\beta}{\sigma_u^2}\right\} \\ &\propto \prod_{t=1}^{T} \frac{1}{\sigma_u} \exp\left\{\frac{1}{2\sigma_u^2}(\tau_t - \tau_{t-1})^2\right\} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma_u^2}\right)^{\alpha+1} exp\left\{-\frac{\beta}{\sigma_u^2}\right\} \\ &= \frac{1}{\sigma_u^{T-1}} exp\left\{\frac{1}{2\sigma_u^2}\boldsymbol{\tau Q \tau}\right\} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma_u^2}\right)^{\alpha+1} exp\left\{-\frac{\beta}{\sigma_u^2}\right\} \\ &\propto \left(\frac{1}{\sigma_u^2}\right)^{\alpha+\frac{T-1}{2}+1} exp\left\{\frac{1}{\sigma_u^2}\left(\frac{1}{2}\boldsymbol{\tau Q \tau} - \beta\right)\right\} \end{aligned}$$

which we recognize as the core of an inverse gamma $\text{IG}(\alpha^*, \beta^*) = \text{IG}(\alpha + \frac{1}{2}(T-1), \beta + \frac{1}{2}\boldsymbol{\tau Q \tau})$

d) Acceptance probability

Let $\mathcal{I} \subseteq \{1, 2, ..., 366\}$ be a set of time indices, and let $-\mathcal{I} = \{1, 2, ..., 366\} \setminus \mathcal{I}$. Furthermore, let $\boldsymbol{\tau}'$ denote the proposed values for $\boldsymbol{\tau}$. Then, by using iterative conditioning, the acceptance probability is given by

$$\alpha(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = \min\left(1, \frac{P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}{P(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})} \frac{Q(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}{Q(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}\right),$$

where our prior proposal distribution is $Q(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)$. By considering

$$\begin{aligned}
P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) &= \frac{P(\boldsymbol{\tau}'_{\mathcal{I}}, \boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})}{P(\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y})} \\
&= \frac{P(\boldsymbol{y}|\boldsymbol{\tau}'_{\mathcal{I}}, \boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2) P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2) P(\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}{P(\boldsymbol{y}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2) P(\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)} \\
&= \frac{\overbrace{P(\boldsymbol{y}|\boldsymbol{\tau}'_{\mathcal{I}}, \boldsymbol{\tau}_{-\mathcal{I}})}^{\text{Conditionally independent}} \quad P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}{P(\boldsymbol{y}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)} \\
&= \frac{P(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{\tau}'_{\mathcal{I}}) P(\boldsymbol{y}_{-\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}) P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}{P(\boldsymbol{y}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}
\end{aligned}$$

and equally

$$P(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) = \frac{P(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{\tau}_{\mathcal{I}}) P(\boldsymbol{y}_{-\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}) P(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}{P(\boldsymbol{y}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}$$

the acceptance probability becomes

$$\begin{aligned}
\alpha(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2, \boldsymbol{y}) &= \min\left(1, \frac{P(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{\tau}'_{\mathcal{I}}) P(\boldsymbol{y}_{-\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}) P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)/P(\boldsymbol{y}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}{P(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{\tau}_{\mathcal{I}}) P(\boldsymbol{y}_{-\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}) P(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)/P(\boldsymbol{y}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)} \frac{P(\boldsymbol{\tau}_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}{P(\boldsymbol{\tau}'_{\mathcal{I}}|\boldsymbol{\tau}_{-\mathcal{I}}, \sigma_u^2)}\right) \\
&= \min\left(1, \frac{P(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{\tau}'_{\mathcal{I}})}{P(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{\tau}_{\mathcal{I}})}\right)
\end{aligned}$$

e) Implementation

```
link = function(tau) {
    return(exp(tau)/(1 + exp(tau)))
}

logbin = function(n, y, tau) {
    # Remake and use this
    return(y * log(1 + exp(-tau)) - (n - y) * log(1 + exp(tau)))
}

acceptRatio = function(n, y, tauProp, tau) {
    # Confirmed faster. N=1000: 4.7 vs 3.81
    return(exp(y * (tauProp - tau) + n * log((1 + exp(tau))/(1 + exp(tauProp)))))
}


mhRW = function(tau, sigma, yt, t, normVec = NA) {
    if (t == 1) {
        mu_ab = tau[2]
        sigma_aa = sigma
    } else if (t == 366) {
        mu_ab = tau[365]
        sigma_aa = sigma
    } else {
        mu_ab = 1/2 * (tau[t - 1] + tau[t + 1])
        sigma_aa = sigma/2
    }
    # prop_tau = rnorm(1,mean=mu_ab, sd=sqrt(sigma_aa)) prop_tau =
```

```
    # normVec[t]*sigma + mu_ab
    prop_tau = normVec * sigma + mu_ab
    n = ifelse(t == 60, 10, 39)
    ratio = acceptRatio(n, yt, prop_tau, tau[t])
    if (runif(1) < min(c(1, ratio))) {
        return(list(tau = prop_tau, accepted = 1))
    } else {
        return(list(tau = tau[t], accepted = 0))
    }
}


mcmcIndivid = function(N, dt, sigma0 = 0.1) {
    # Allocate memory
    tau = matrix(NA, nrow = N, ncol = 366)
    sigma = numeric(length = N)
    tau_i = numeric(length = 366)
    normMat = matrix(rep(rnorm(366), N), nrow = N, ncol = 366)
    normVec = normMat[1, ]

    # Find init vals
    tau[1, ] = rnorm(366)  # init tau drawn from normal distr.
    # tau[1,] = runif(366, -100, 100) # init tau drawn from uniform distr.
    sigma[1] = sigma0

    # Make Q matrix
    Q = matrix(0, nrow = 366, ncol = 366)
    diag(Q) = 2
    Q[c(1, length(Q))] = 1
    Q[abs(row(Q) - col(Q)) == 1] <- -1

    # Run mcmc for N iterations
    accepted = 0
    for (i in 2:N) {
        tau_i = tau[i - 1, ]
        sigma_i = sigma[i - 1]
        for (t in 1:366) {
            # rtemp= mhRW(tau_i, sigma_i, dt$n.rain[t], t)
            rtemp = mhRW(tau_i, sqrt(sigma_i), dt$n.rain[t], t, normVec[t])
            tau[i, t] = rtemp$tau
            accepted = accepted + rtemp$accepted
        }
        normVec = normMat[i, ]
        # Squared diff. of tau vec.
        tQt = sum((tau[i, -366] - tau[i, -1])^2)  # this sim tau vals.
        # tQt = sum((tau[i-1,-366] - tau[i-2,-1])^2) # prev sim tau vals.

        # Gibbs step (Draw from IG)
        sigma[i] = 1/rgamma(1, 2 + (366 - 1)/2, 0.05 + 0.5 * tQt)  # Gibbs inline

        if (i%%(N/10) == 0) {
            print(i/N * 100)
            print(accepted/(i * 366))
```

```
        }
    }
    return(list(tau = tau, sigma = sigma))
}
```

```
set.seed(321)
N = 50000
ptm = proc.time()
results = mcmcIndivid(N, rain)
```

```
## [1] 10
## [1] 0.5853956
## [1] 20
## [1] 0.5853612
## [1] 30
## [1] 0.5852816
## [1] 40
## [1] 0.5851967
## [1] 50
## [1] 0.5852091
## [1] 60
## [1] 0.5850048
## [1] 70
## [1] 0.5851643
## [1] 80
## [1] 0.5851481
## [1] 90
## [1] 0.5850959
## [1] 100
## [1] 0.5851396
```

```
proc.time() - ptm
```

```
##    user  system elapsed
##  159.08    0.03  159.65
```

```
# sum(is.na(results$tau)) sum(is.na(results$sigma))
```

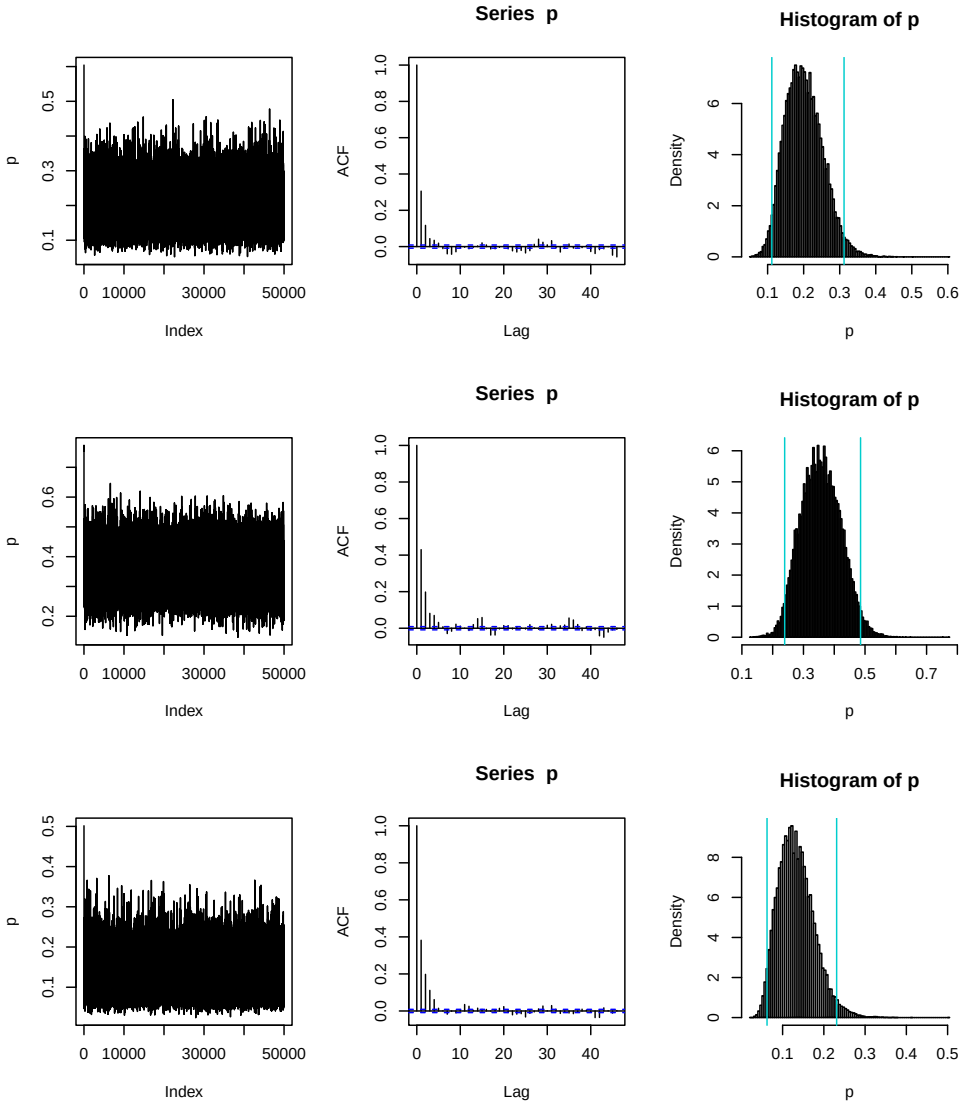
```
CImean = function(p, col = "cyan3") {
    c(mean = mean(p), quantile(p, probs = c(0.025, 0.975)))
}

plotTAH = function(p, hcol = "cyan3") {
    # Plots trace, autocorr and hist of probs
    plot(p, type = "l")
    acf(p)
    hist(p, nclass = 100, prob = T)
    abline(v = quantile(p, probs = 0.025), col = hcol)
    abline(v = quantile(p, probs = 0.975), col = hcol)
}
```

```
p1 = link(results$tau[, 1])
p201 = link(results$tau[, 201])
p366 = link(results$tau[, 366])

par(mfrow = c(3, 3))

plotTAH(p1)
plotTAH(p201)
plotTAH(p366)
```

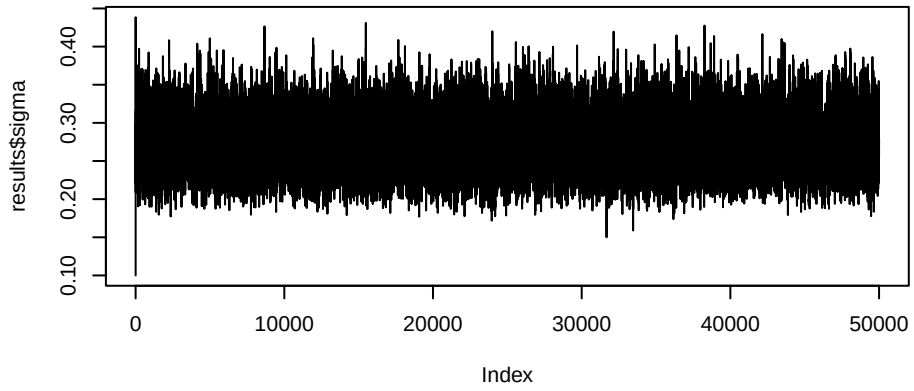


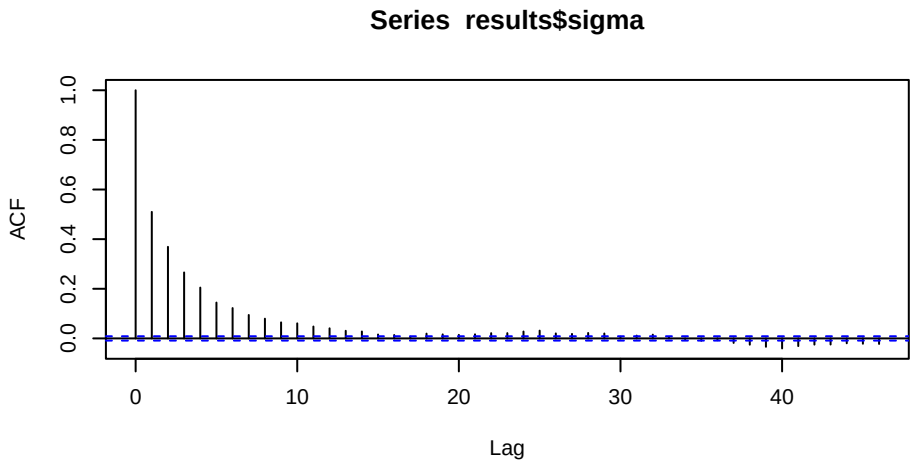

```
ciAndMean = rbind(CImean(p1[1:N]), CImean(p201[1:N]), CImean(p366[1:N]))
rownames(ciAndMean) = c(1, 201, 366)
ciAndMean
```

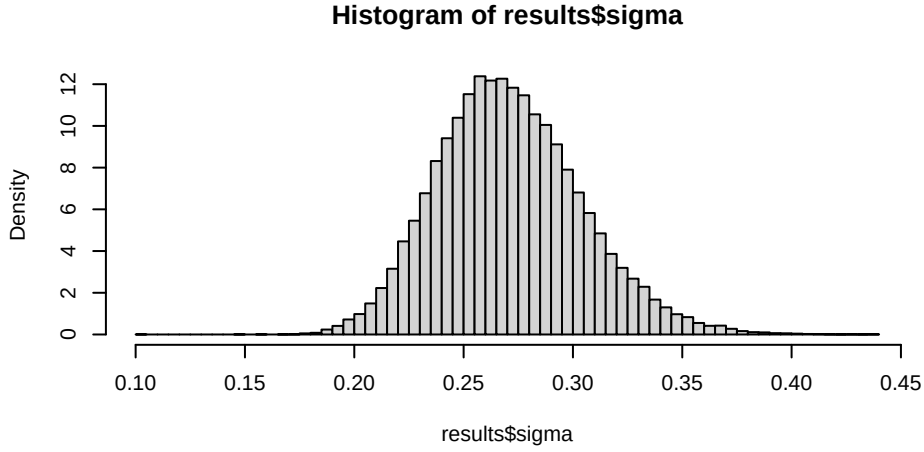
```
##          mean       2.5%     97.5%
## 1   0.2013289 0.11166385 0.3120152
```

```
## 201 0.3569094 0.23896194 0.4854839
## 366 0.1334187 0.06226288 0.2310067
```

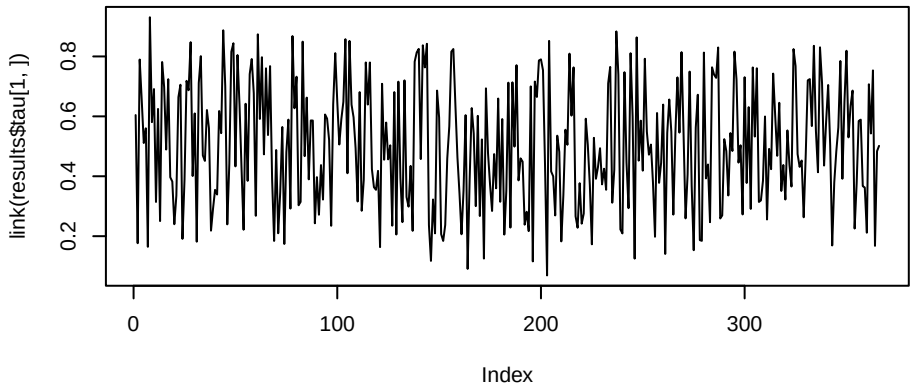
```
par(mfrow = c(3, 1))
plot(results$sigma, type = "l")
acf(results$sigma)
hist(results$sigma, nclass = 100, prob = T)
```

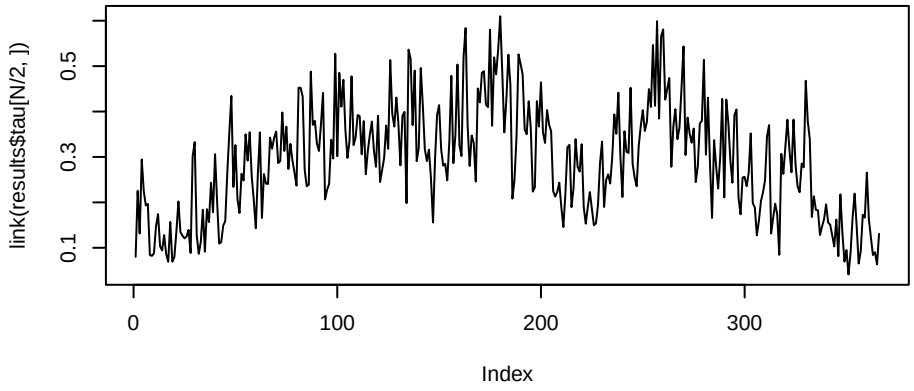


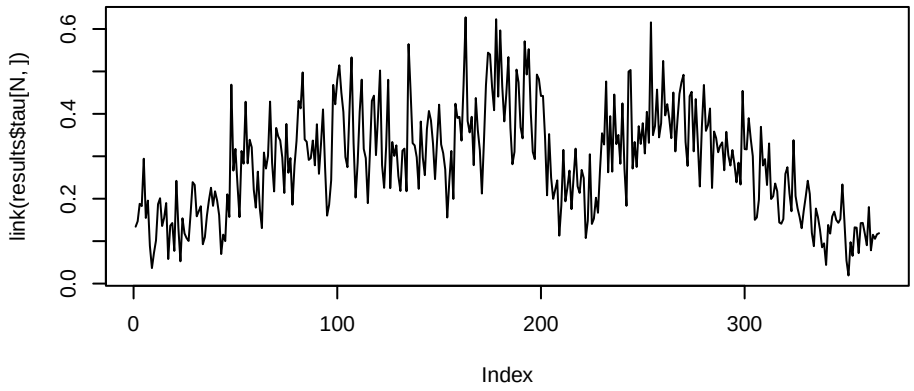





```
par(mfrow = c(3, 1))
plot(link(results$tau[1, ]), type = "l")
plot(link(results$tau[N/2, ]), type = "l")
plot(link(results$tau[N, ]), type = "l")
```


link(results$tau[1,])
0.2
0.4
0.6
0.8
0
100
200
300
Index


link(results$tau[N/2,])
0.1
0.3
0.5
0
100
200
300
Index


link(results$tau[N,])
0.0
0.2
0.4
0.6
0
100
200
300
Index