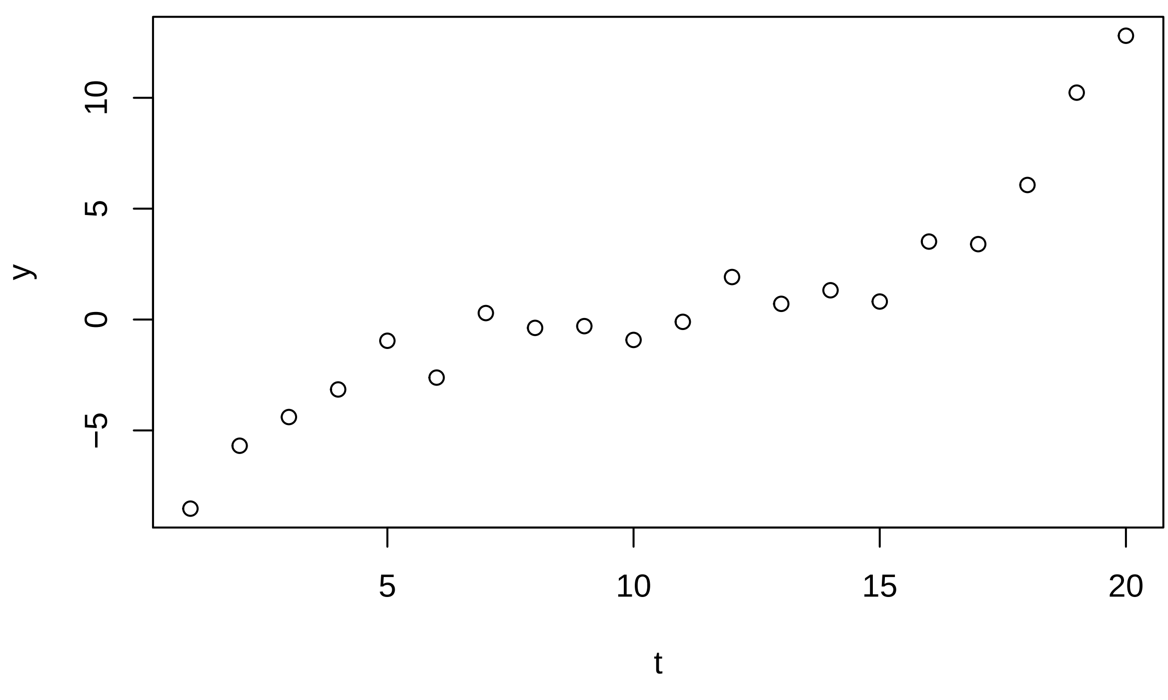
Oblig2-TMA4300

Martine Middelthon

27 2 2020

Problem B

```r
# Read and plot data
gaussiandata = read.delim("gaussiandata.txt")
y = gaussiandata[,1]
t = seq(from=1,to=length(y),by=1)
plot(t,y)
```



1.

We consider the problem of smoothing the time series that is plotted above. We assume that given the vector of linear predictors $\eta = (\eta_1, \dots, \eta_T)$, where in this case $T = 20$, the observations y_t are independent and distributed according to

$$y_t \mid \eta_t \sim \mathcal{N}(\eta_t, 1) \quad ,$$

for $t = 1, \ldots, T$. The linear predictor for time t is $\eta_t = f_t$, where f_t is the smooth effect for time t. For the prior distribution of $\mathbf{f} = (f_1, \ldots, f_T)$ we have a second order random walk model, that is,

$$\pi(\mathbf{f} \mid \theta) \propto \theta^{(T-2)/2} \exp\left\{ -\frac{\theta}{2} \sum_{t=3}^{T} (f_t - 2f_{t-1} + f_{t-2}^2) \right\} = \mathcal{N}(\mathbf{0}, \mathbf{Q}(\theta)^{-1}) \quad ,$$

where $\mathbf{Q}$ is the precision matrix and θ is the precision parameter that controls the smoothness of $\mathbf{f}$. We assume that the $Gamma(1,1)$-distribution is the prior for θ.

The model described here can be written as the hierarchichal model:

$$\begin{aligned}
\mathbf{y} \mid \mathbf{f} &\sim \prod_{t=1}^{T} P(y_t \mid \eta_t) \\
\mathbf{f} \mid \theta &\sim \pi(\mathbf{f} \mid \theta) = \mathcal{N}(\mathbf{0}, \mathbf{Q}(\theta)^{-1}) \\
\theta &\sim Gamma(1,1)
\end{aligned}$$

Here, the first line is the likelihood of the response $\mathbf{y} = (y_1, \ldots, y_T)$, the second line gives the prior distribution of the latent field, and the third line gives the prior distribution of the hyperparameter θ. Since our model has this particular structure, it is a latent Gaussian model. INLA can be used to estimate the parameters because we have a latent gaussian model where each data point y_t depends only on the one element f_t in the latent field, the dimension of the hyperparameter is one and the precision matrix $\mathbf{Q}(\theta)$ of the latent field is sparse.

2.

Here, we implement a block Gibbs sampling algorithm for $f(\eta, \theta \mid \mathbf{y})$, where we propose a new value for θ from the full conditional $\pi(\theta \mid \eta, \mathbf{y})$ and a new value for η from the full conditional $\pi(\eta \mid \theta, \mathbf{y})$. Thus, we need to find these distributions. We start with the posterior

$$\pi(\eta, \theta \mid \mathbf{y}) \propto \pi(\theta)\pi(\eta \mid \theta) \prod_{t=1}^{T} \pi(y_t \mid \eta_t, \theta) \propto \frac{\theta^{(T-2)/2}}{(2\pi)^{T/2}} \exp\left\{ -\theta - \frac{\theta}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 - \frac{1}{2} \sum_{t=1}^{T} (y_t - \eta_t)^2 \right\}.$$

Then we find the full conditional for θ to be

$$\begin{aligned}
\pi(\theta \mid \mathbf{y}, \eta) &\propto \theta^{T/2-1} \exp\left\{ -\theta \left(1 + \frac{1}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 \right) \right\} \\
&\propto Gamma\left(\frac{T}{2}, 1 + \frac{1}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 \right)
\end{aligned} .$$

The full conditional for η is

$$\begin{aligned}
\pi(\eta \mid \theta, \mathbf{y}) &\propto \exp\left\{ -\frac{\theta}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 - \frac{1}{2} \sum_{t=1}^{T} (y_t - \eta_t)^2 \right\} \\
&= \exp\left\{ -\frac{1}{2} \left(\eta^T \mathbf{Q} \eta + (\mathbf{y} - \eta)^T (\mathbf{y} - \eta) \right) \right\} \\
&= \exp\left\{ -\frac{1}{2} \eta^T (\mathbf{Q} + \mathbf{I}) \eta + \mathbf{y}^T \eta \right\}
\end{aligned} .$$

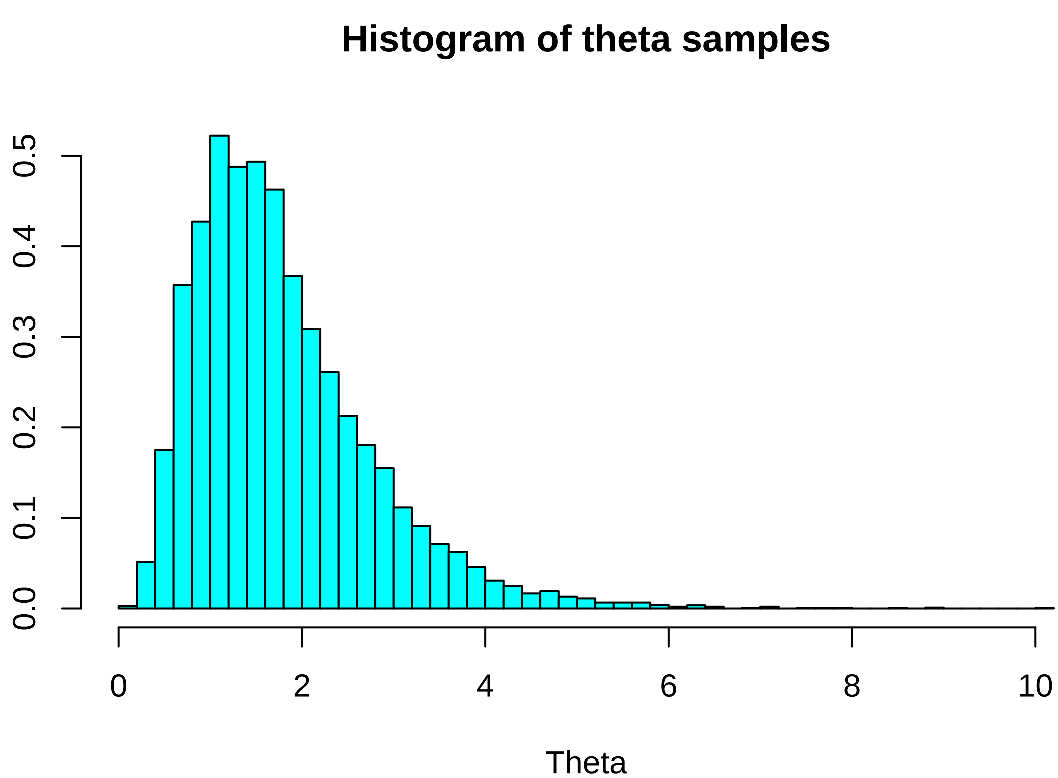
Here, $\mathbf{Q}(\theta) = \theta \mathbf{L}\mathbf{L}^T$ is the precision matrix, where $\mathbf{L}$ is the $T \times (T-2)$ matrix

$$\mathbf{L} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & -2 & 1 & 0 & 0 & \dots \\ \vdots & & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

By looking at the last line in the above expression for $\pi(\eta \mid \theta, \mathbf{y})$, we recognize that the canonical parametrization is $\mathcal{N}(\mathbf{y}, \mathbf{Q}+\mathbf{I})$, and find that $\pi(\eta \mid \theta, \mathbf{y}) \propto \mathcal{N}((\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y}, (\mathbf{Q}+\mathbf{I})^{-1})$. In the algorithm we sample the new proposals for the parameters from these two distributions that we have found for the full conditionals. We always use the last updated parameters.

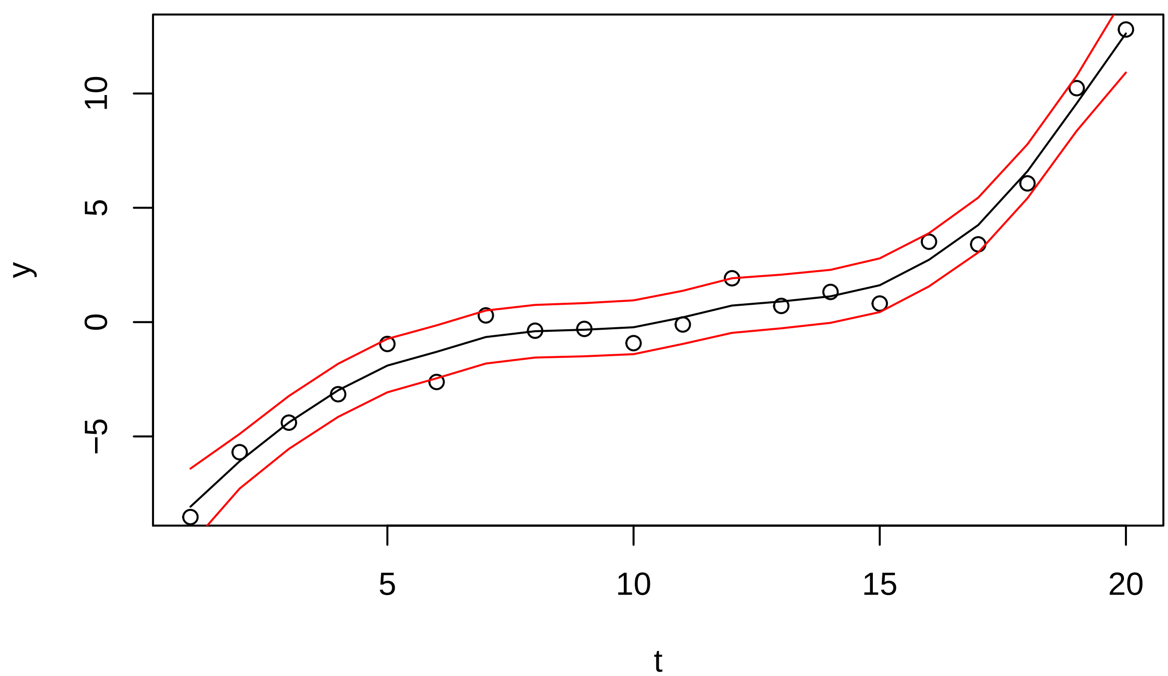
```r
library(Matrix)
library(mvtnorm)
library(MASS)

# Function to make the precision matrix
make.Q = function(T, theta) {
  # Make the matrix L as described in the text
  L = diag(T)
  d1 = rep(-2,T-1)
  d2 = rep(1, T-2)
  L[row(L)-col(L)==1] = d1
  L[row(L)-col(L)==2] = d2
  L = L[,-c(T-1,T)]
  # Compute Q(theta)
  Q = theta * L %*% t(L)
  return(Q)
}

set.seed(0)
# Function for block Gibbs sampling
# n is the number of samples including the inital value
sample.Gibbs = function(n, theta.init, f.init, y) {
  T = length(f.init)
  # Make vector and matrix for storing the samples
  theta.vec = rep(0,n)
  f.matrix = matrix(1:T*n, nrow = T, ncol = n)
  # Initialize
  theta.vec[1] = theta.init
  f.matrix[, 1] = f.init
  # Iterations
  for(i in 2:n) {
    # Sample theta
    summ = 0
    for(t in 3:T) {
      summ = summ + (f.matrix[t, i-1] - 2*f.matrix[t-1, i-1] + f.matrix[t-2, i-1])^2
    }
    theta.vec[i] = rgamma(1, shape = T/2, rate = 1 + 0.5*summ)
    # Sample f
    Q = make.Q(T, theta.vec[i])         # Use the last updated theta
    f.mean = solve(Q+diag(T)) %*% y
    f.sigma = solve(Q+diag(T))
```

```
    f.matrix[, i] = rmvnorm(1, f.mean, f.sigma)
  }
  return(rbind(f.matrix, theta.vec))     # Return concatenated matrix with f and theta samples
}

# Set values
n = 10000
T = length(y)
theta.init = 1
f.init = rep(2,T)
# Sample
result = sample.Gibbs(n, theta.init, f.init, y)
result.theta = result[length(result[,1]), -c(1:100)]    # Extracting the theta samples, excluding the fi
result.f = result[-length(result[,1]), -c(1:100)]       # Extracting the f samples, excluding the first

# Estimate for the posterior marginal for theta
truehist(result.theta, xlab = "Theta", main = "Histogram of theta samples")
```

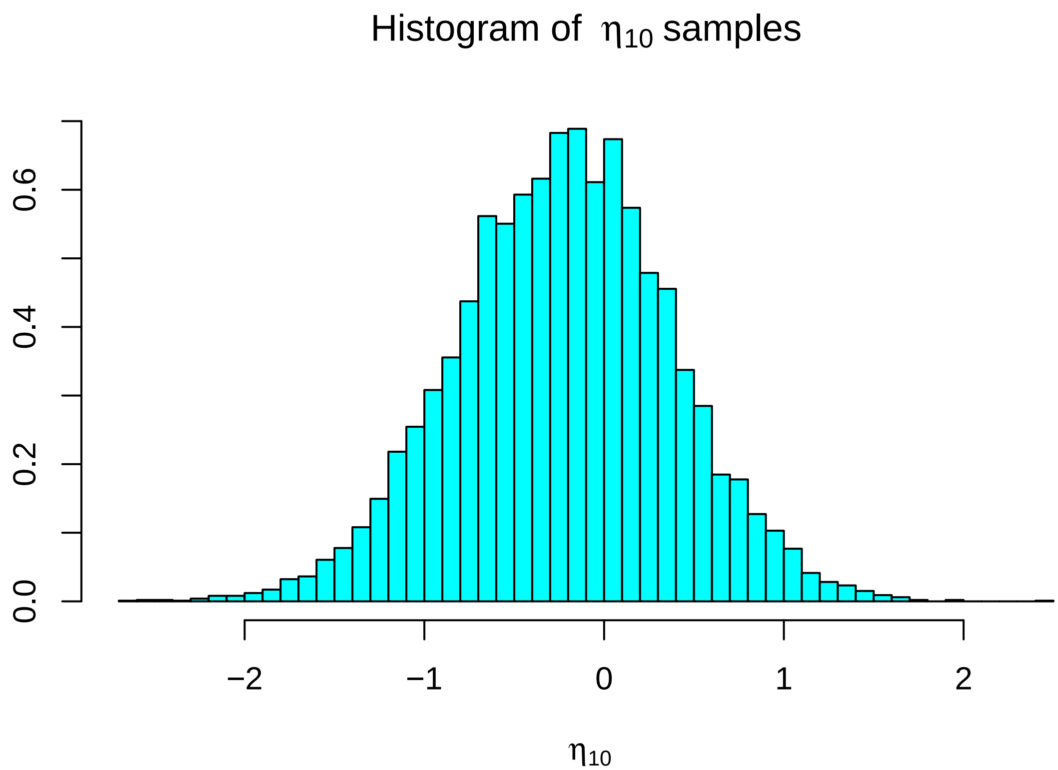



```
# Vectors for storing the mean, variance and confidence bounds
f.mean = rep(0,T)
f.var = rep(0,T)
conf.upper = rep(0,T)
conf.lower = rep(0,T)
# Calculate the mean and variance
for(t in 1:T) {
```

```
  f.mean[t] = mean(result.f[t,])
  f.var[t] = var(result.f[t,])
}
# Calculate 95% confidence bounds
for(t in 1:T) {
  z = qnorm(0.025)
  conf.upper[t] = f.mean[t] + z * sqrt(f.var[t])
  conf.lower[t] = f.mean[t] - z * sqrt(f.var[t])
}
# Plotting
t = seq(from = 1, to = T, by = 1)
plot(t, f.mean, type = "l", ylab = "y")
points(t, y)
lines(t, conf.lower, col = "red")
lines(t, conf.upper, col = "red")
```




```
# Estimate of pi(eta_10|y)
f_10 = result.f[10,]
truehist(f_10, xlab = bquote(~eta[10]), main = bquote("Histogram of " ~eta[10]~"samples"))
```



The first histogram shows an estimate for $\pi(\theta \mid \mathbf{y})$. In the plot the data points are plotted as circles. The black line is plotted using the estimates of the smooth effects. The red lines are the 95% confidence bounds. Almost all the data points are within the bounds. The last histogram of the eta_{10} samples provides an estimate of $\pi(\eta_{10} \mid \mathbf{y})$.

3.

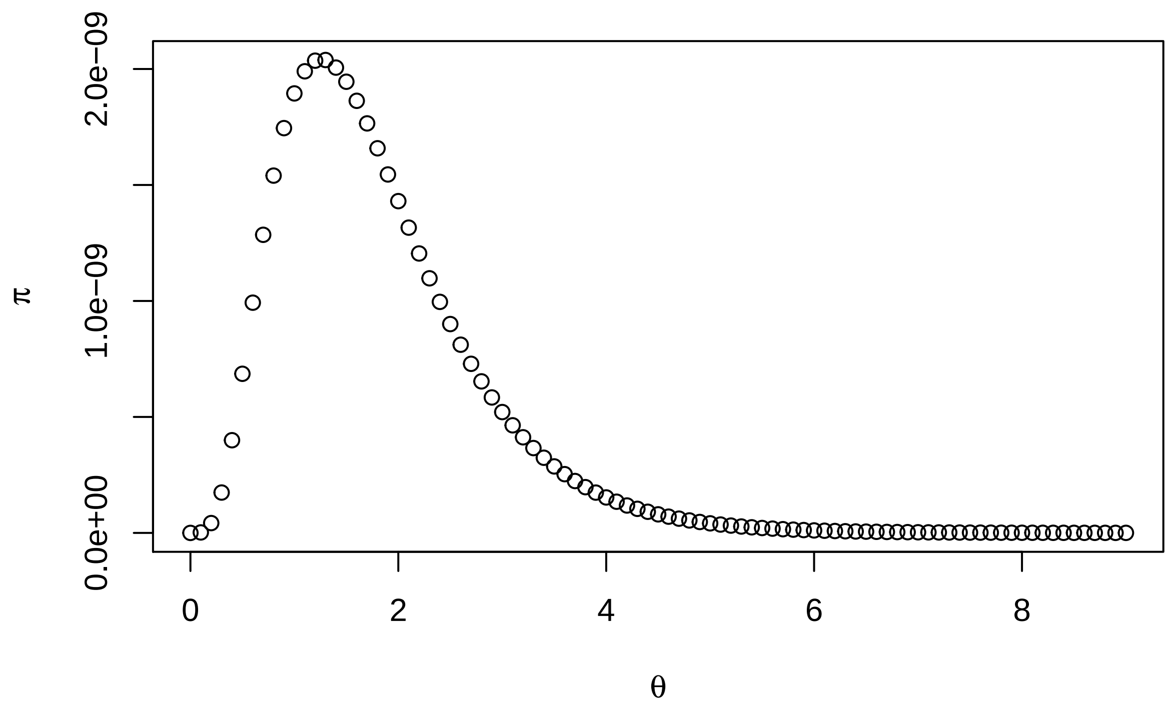
We want to approximate $\pi(\theta \mid \mathbf{y})$ using the INLA scheme. Since we found that $\pi(\eta \mid \theta, \mathbf{y}) \propto \mathcal{N}((\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y}, (\mathbf{Q}+\mathbf{I})^{-1})$, we can calculate

$$
\begin{aligned}
\pi(\theta \mid \mathbf{y}) &\propto \frac{\pi(\mathbf{y} \mid \eta, \theta)\pi(\eta \mid \theta)\pi(\theta)}{\pi(\eta \mid \theta, \mathbf{y})} \\
&\propto \frac{\exp(-\frac{1}{2}(\mathbf{y}-\eta)^T(\mathbf{y}-\eta))\theta^{(T-2)/2}\exp(-\frac{1}{2}\eta^T\mathbf{Q}\eta)\exp(-\theta)}{|\mathbf{Q}+\mathbf{I}|^{1/2}\exp(-\frac{1}{2}(\eta-(\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y})^T(\mathbf{Q}+\mathbf{I})(\eta-(\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y}))} \quad , \\
&= \theta^{(T-2)/2}|\mathbf{Q}+\mathbf{I}|^{-1/2}\exp\left(-\theta-\frac{1}{2}\mathbf{y}^T(\mathbf{I}-(\mathbf{Q}+\mathbf{I})^{-1})\mathbf{y}\right)
\end{aligned}
$$

where $|\cdot|$ denotes the determinant and we have used that $|\mathbf{A}^{-1}| = \frac{1}{|A|}$. We use a grid θ_{grid} of values for θ and calculate the posterior marginal. The plot below shows the result, and it seems to be in concordance with the MCMC estimate displayed by the histogram.

```
# Function to calculate pi(theta|y) for each theta in the grid
pi_theta_y = function(theta.grid, y) {
  pi = rep(0,length(theta.grid))
  T = length(y)
```

```
  for(i in 1:length(pi)){
    theta = theta.grid[i]
    Q = make.Q(T, theta)
    deter = det(solve(Q+diag(T)))
    pi[i] = theta^(T/2-1) * exp(-theta) * deter^(0.5) * exp(-0.5 * t(y) %*% (diag(T)-solve(Q+diag(T))) %
  }
  return(pi)
}

thetas = seq(from = 0, to = 9, by = 0.1)     # Theta grid
pi = pi_theta_y(thetas, y)                   # Corresponding values for pi(theta|y)
plot(thetas, pi, xlab = bquote(theta), ylab = bquote(pi))    # Plotting
```



4.

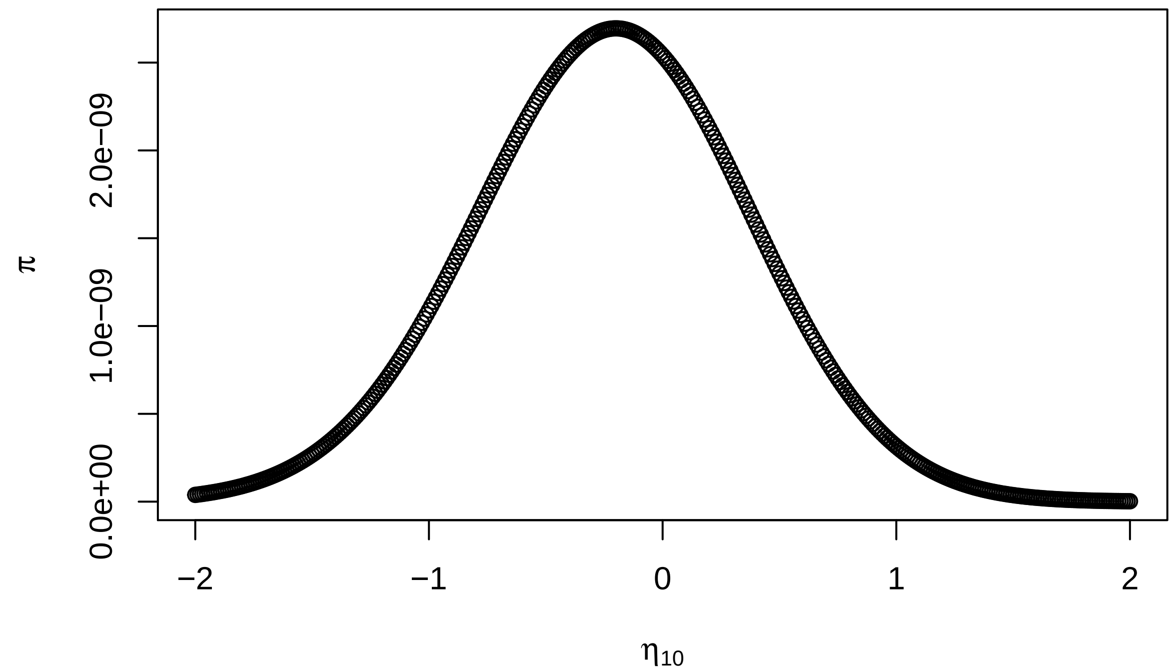
We also want to implement the INLA scheme for the approximation of $\pi(\eta_i \mid \mathbf{y})$. We have

$$
\begin{aligned}
\pi(\eta_i \mid \mathbf{y}) &= \int \pi(\eta_i \mid \mathbf{y}, \theta)\pi(\theta \mid \mathbf{y})d\theta \\
&\approx \sum_{\theta_k \in \theta_{\text{grid}}} \pi(\eta_i \mid \mathbf{y}, \theta_k)\pi(\theta_k \mid \mathbf{y})\Delta
\end{aligned},
$$

where θ_{grid} is the grid of theta values from point 3, and Δ is the step size between the values in the grid. Since $\pi(\eta \mid \theta, \mathbf{y}) \propto \mathcal{N}((\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y}, (\mathbf{Q}+\mathbf{I})^{-1})$, we assume that $\pi(\eta_i \mid \mathbf{y}, \theta) \sim \mathcal{N}([\mathbf{A}\mathbf{y}]_i, \mathbf{A}_{ii})$, where $\mathbf{A} = (\mathbf{Q}+\mathbf{I})^{-1}$.

We calculate $\pi(\eta_i \mid \mathbf{y})$ for $i = 10$ and values for $\eta_{10} \in [-2, 2]$. The plot below shows the result. The graph looks approximately normal with a small and negative mean, which also the estimation obtained using the block Gibbs sampling (displayed by the last histogram in point 2) does.

```
# Function for calculating pi(eta_10|y,theta_k) for each eta_10 in the grid
pi_etai_y_theta = function(etai.grid, theta, y) {
  i = 10
  T = length(y)
  Q = make.Q(T, theta)
  A = solve(Q + diag(T))
  mean = (A %*% y)[i]
  var = A[i,i]
  pi = dnorm(etai.grid, mean = mean, sd= sqrt(var))
  return(pi)    # Return the vector corresponding to each eta_10 in the grid
}

# Function for calculating pi(eta_10|y) for each eta_10 in the grid
pi_etai_y = function(y,theta.grid, eta.grid) {
  sums = rep(0, length(eta.grid))   # Vector for storing the approximations
  step = theta.grid[2]-theta.grid[1]  # Step size
  theta_y = pi_theta_y(theta.grid, y) # vector of pi(theta|y) for each theta in the grid
  for(k in (1:length(theta.grid))) {
    theta = theta.grid[k]             # theta_k
    sums = sums + pi_etai_y_theta(eta.grid, theta, y) * theta_y[k] * step # Adding the terms for theta_
  }
  return(sums)
}


thetas = seq(from = 0, to = 9, by = 0.1)    # Theta grid
eta.grid = seq(-2,2,0.01)                   # Eta grid
etai_y = pi_etai_y(y,thetas,eta.grid)       # Vector of pi(eta_10|y) for each eta_10 in the grid

plot(eta.grid,etai_y, ylab = bquote(pi), xlab = bquote(eta[10]))    # Plotting the result
```



5.

We now use built in inla function for the same estimates as above. In the first figure the estimated smooth effects using inla are plotted as a red line. The MCMC estimates are also plotted in the same figure as a black line. The estimates are very similar, so the lines are overlapping. The second figure shows the estimate of $\pi(\theta \mid \mathbf{y})$, which looks very similar to the ones from point 2 and 3. The last figure shows the estimate for $\pi(\eta_{10} \mid \mathbf{y})$, and it looks like the estimates from point 2 and 4.

```
library(INLA)
```

```
## Loading required package: sp
```

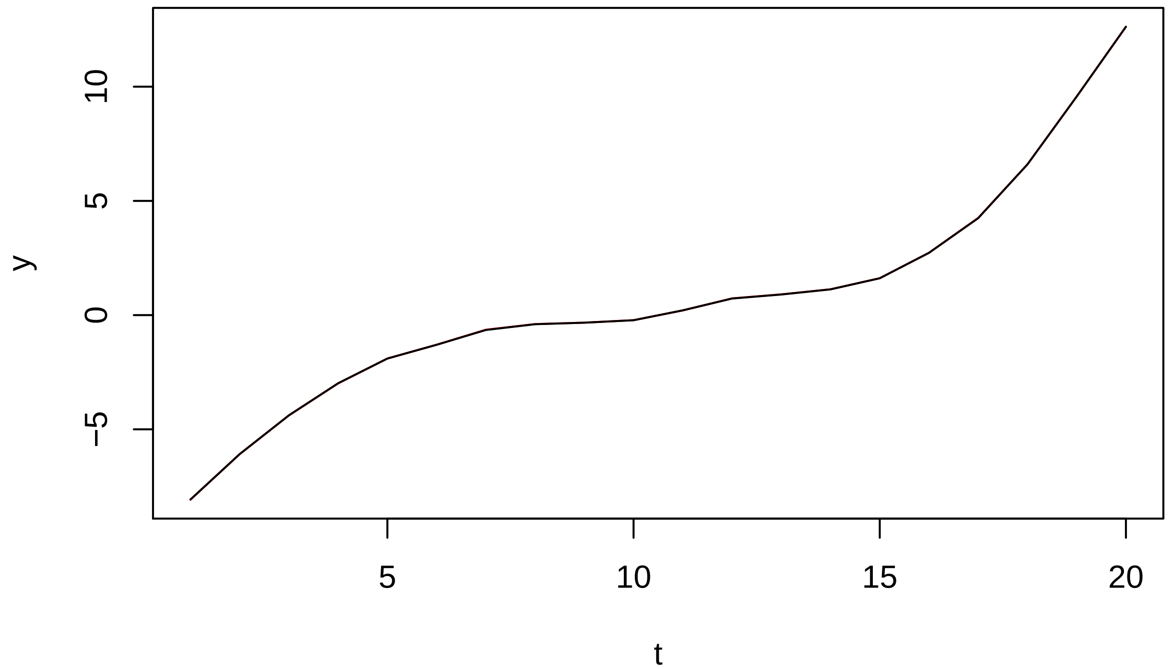
```
## Warning: package 'sp' was built under R version 3.6.3
```

```
## Loading required package: parallel
```

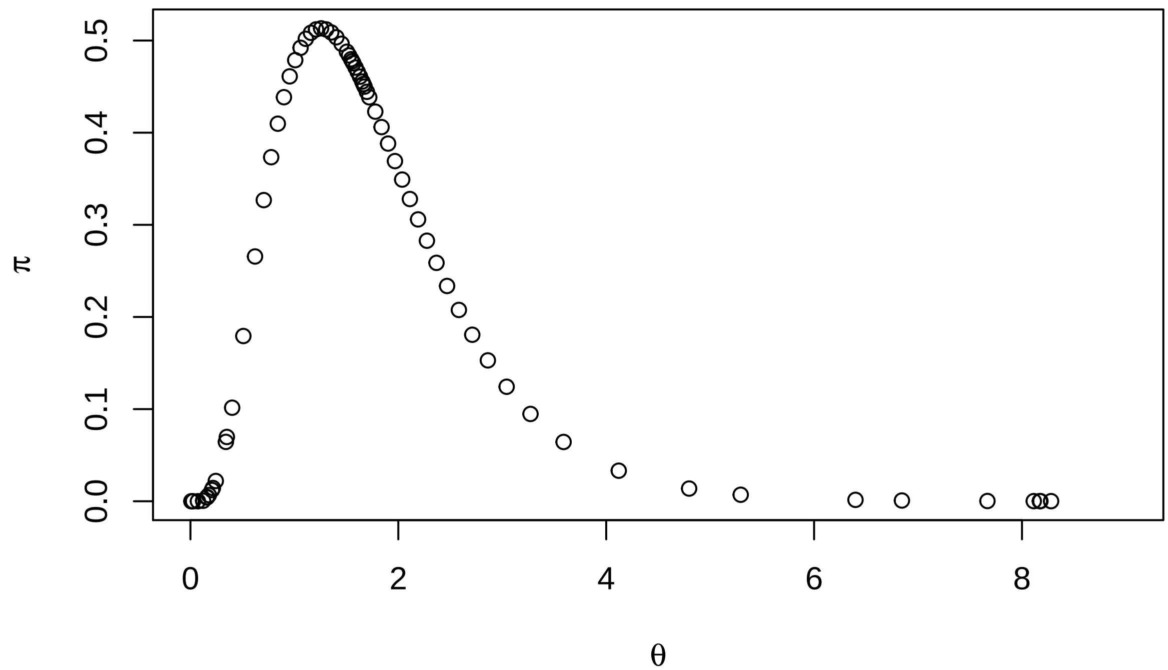
```
## This is INLA_19.09.03 built 2019-09-03 09:03:02 UTC.
## See www.r-inla.org/contact-us for how to get help.
```

```
T = 20
t = seq(from = 1, to = T, by = 1)
data = data.frame(y = y, t = t)
```

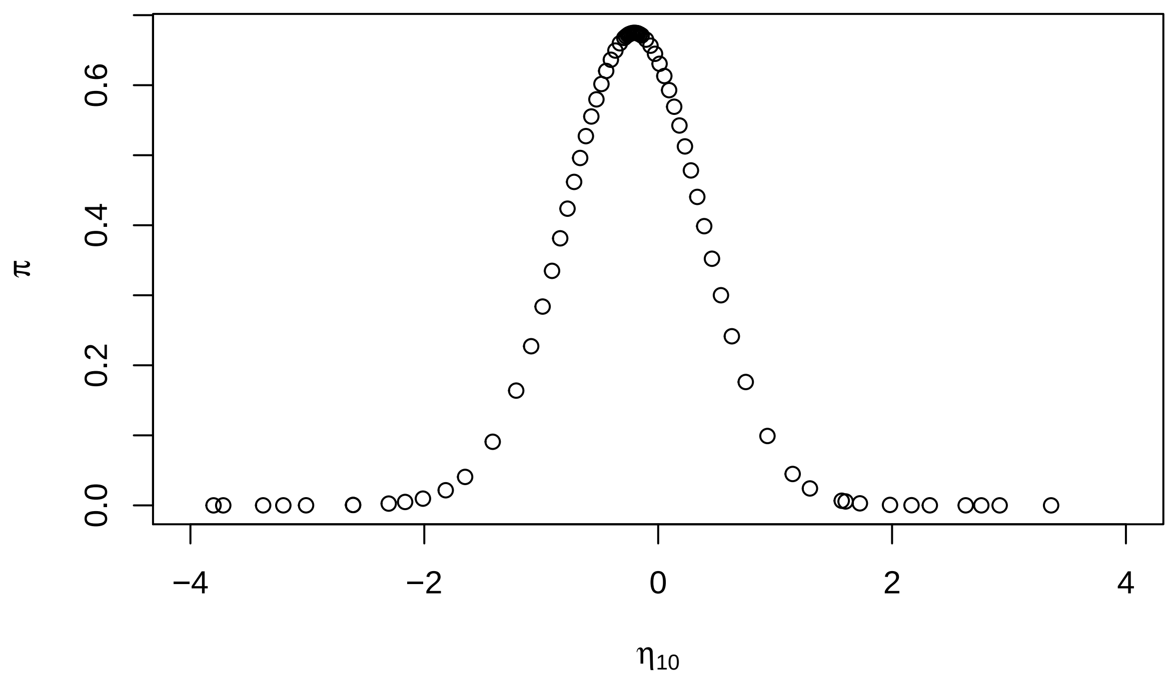
```
thetahyper = list(theta = list(prior = "log.gamma", param = c(1, 1)))
formula = y   f(t, model = "rw2", hyper = thetahyper, constr = FALSE) - 1
result1 = INLA::inla(formula = formula, family = "gaussian", data = data, control.family = list(hyper=li

plot(result1$summary.random$t$mean, xlab="t", ylab="y", type="l", col="red")
lines(t, f.mean)
```




```
plot(result1$marginals.hyperpar$`Precision for t`, xlim =c(0,9), xlab=bquote(theta),ylab=bquote(pi))
```




```
plot(result1$marginals.random$t$index.10,xlim=c(-4,4),xlab=bquote(eta[10]),ylab=bquote(pi))
```


0.0
0.2
0.4
0.6
π
−4
−2
0
2
4
η_{10}