TDDE07 - Lab 2

Sebastian Callh, Jacob Lundberg

18 april 2018

Assignment 1 - Linear and polynomial regression

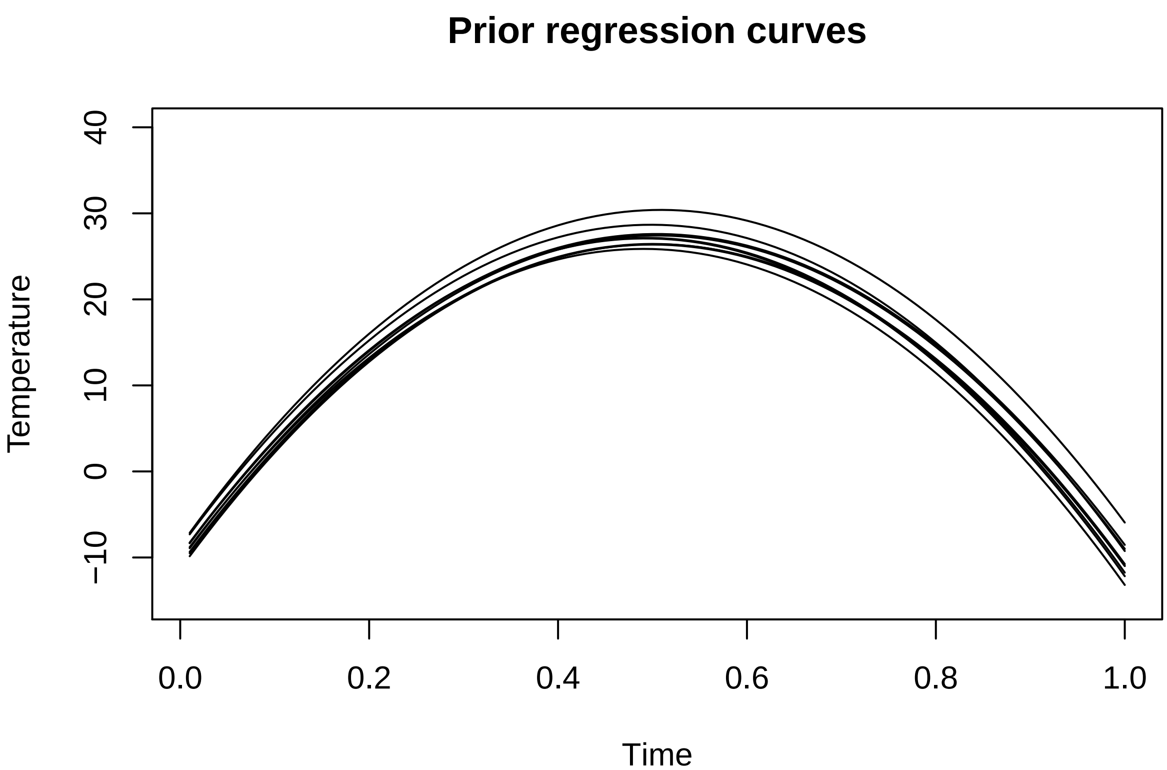
a)

When selecting the prior hyperparameters we had some domain knowledge from the course TDDE01, where we had processed weather data before. From that we knew that the mean temperature in Sweden was about 8 degrees. Of course, we also knew that it is warmer during the middle half of the year than the earlier/later. Apart from that we knew very little, so in the light of that knowledge we assigned our priors by iteratively plotting a collection of prior regression curves and tweaking our values. The final paremetrisation was

$$\mu_0 = (-10, 150, -150), \nu_o = 20, \Omega_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \sigma^2 = 2.$$

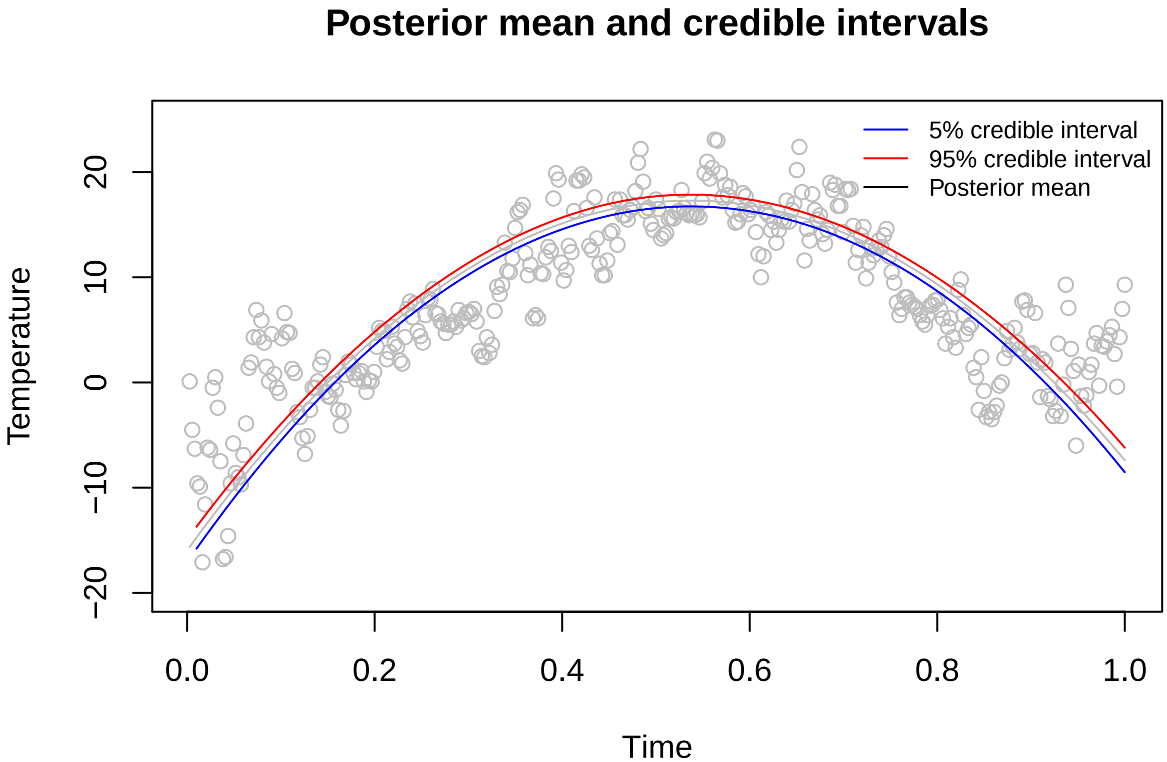
A collection of prior prediction curves from the final parametrisation can be seen below.

b)



c)

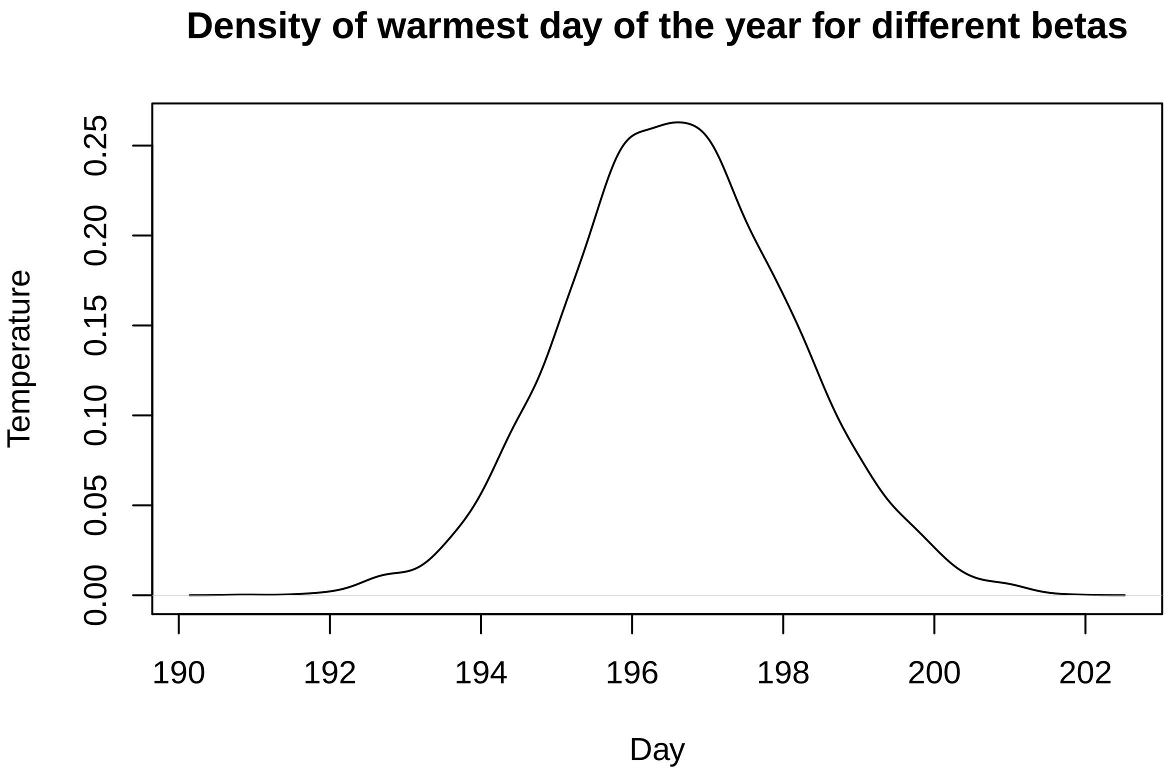
Since a conjugate prior was used the joint posterior distribution of β and σ^2 was computed by the posterior mapping. A scatter plot with the temperature data and the regression function mean with a 90% credible interval can be seen below.



The interval barely contains any data. It should not, since it only quantifies the uncertainty about the data mean and not the actual data.

d)

Finding the warmest day is done by finding the maximum of a second degree polynomial. This is done by differentiating the function, setting it to 0 and solving for t which gives $t = -\frac{\beta_1}{2\beta_2}$. Computing the maximum for several posterior draws of β yields the distribution plotted below which has the mean value $196.6476 \approx 197$.



e)

Since the higher order terms are not believed to be needed they can be assigned $\mu_i = 0$ and $\Omega_{0,ii}$ very large. That is, we assign the prior formalizing our beliefe that they should be very small. That will reduce the models flexibility and combat overfitting.

2. Posterior approximation for classification with logistic regression

a)

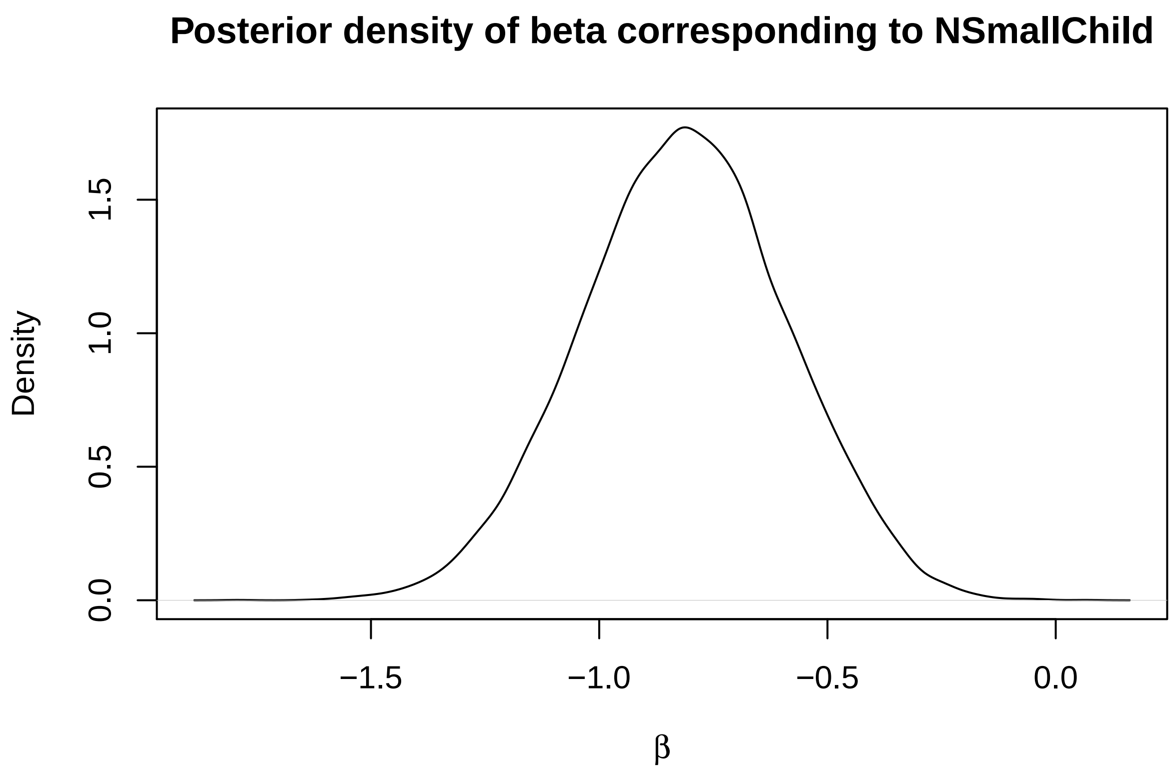
No questions asked on this part.

b)

Using `optim.R` to find $\hat{\beta} = (0.38, -0.01, 0.11, 0.10, -0.09, -0.05, -0.82, -0.01)$ and

$$J_y^{-1}(\hat{\beta}) = \begin{bmatrix} -105.78 & -2216.83 & -1311.95 & -1047.17 & -156.42 & -4451.79 & -27.35 & -137.88 \\ -2216.83 & -60854.93 & -28817.20 & -21918.24 & -3196.04 & -94428.20 & -542.68 & -2776.39 \\ -1311.91 & -28817.20 & -16881.39 & -12957.13 & -1907.13 & -55135.90 & -361.45 & -1657.08 \\ -1047.17 & -21918.24 & -12957.13 & -15641.80 & -2930.44 & -46403.13 & -208.47 & -1061.18 \\ -156.42 & -3196.04 & -1907.13 & -2930.43 & -645.15 & -7243.13 & -21.19 & -124.80 \\ -4451.79 & -94428.20 & -55135.90 & -46403.13 & -7243.13 & -194210.76 & -930.93 & -5397.63 \\ -27.35 & -542.68 & -361.45 & -208.48 & -21.20 & -930.93 & -34.64 & -40.13 \\ -137.88 & -2776.32 & -1657.09 & -1061.18 & -124.80 & -5397.64 & -40.13 & -368.02 \end{bmatrix}$$

The plot below shows the distribution of the `NSmallChild` variable. 95% credible interval limits are included.




```
## [1] "Credible interval 2.5 % limit: -1.259324"
```

```
## [1] "Credible interval 97.5% limit: -0.381006"
```

Comparing the posterior mean of the other β show that the β corresponding to `NSmallChild` is the biggest, and because of that `NSmallChild`is an important factor. The actuall numbers can be seen below.

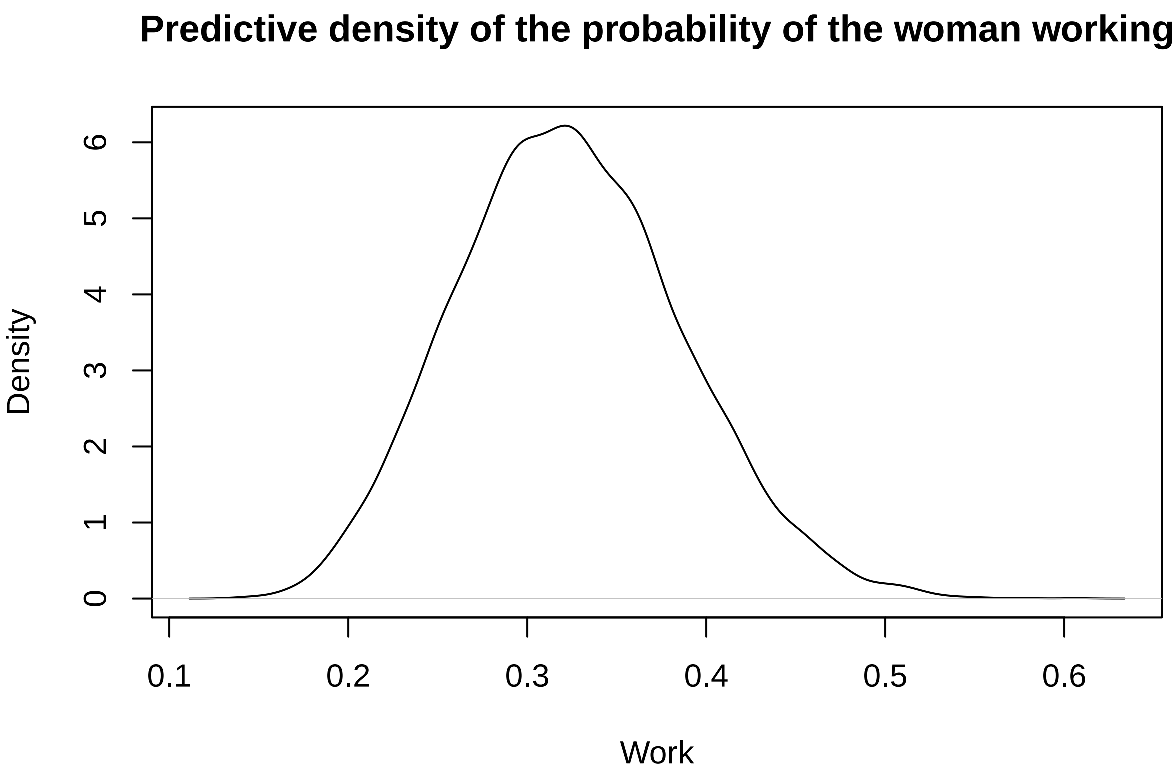
```
##
## Call:
## glm(formula = Work ~ 0 + ., family = binomial, data = data2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1662  -0.9299   0.4391   0.9494   2.0582
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## Constant     0.64430    1.52307   0.423 0.672274
## HusbandInc  -0.01977    0.01590  -1.243 0.213752
## EducYears    0.17988    0.07914   2.273 0.023024 *
## ExpYears     0.16751    0.06600   2.538 0.011144 *
## ExpYears2   -0.14436    0.23585  -0.612 0.540489
## Age         -0.08234    0.02699  -3.050 0.002285 **
## NSmallChild -1.36250    0.38996  -3.494 0.000476 ***
## NBigChild   -0.02543    0.14172  -0.179 0.857592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.26  on 200  degrees of freedom
```

```
## Residual deviance: 222.73  on 192  degrees of freedom
## AIC: 238.73
##
## Number of Fisher Scoring iterations: 4
```

The `Estimate` column shows a significantly larger value for `NSmallChild` compared to the others. Thus it is important.

c)

The predictive distribution that a woman works that is 40 years old, with two children (aged 3 and 9), 8 years of education, 10 years of experience and a husband with income 10 is described by the following plot.




```
## [1] "The mean probability that the woman works is 0.32"
```

```r
library(mvtnorm)
data <- read.csv("TempLinkoping.txt", sep = "\t")
X <- cbind(rep(1, length(data$time)), data$time, data$time^2)
Y <- data$temp

# 1. Linear and polynomial regression

rinvchisq <- function(vx, sigmax, draws) {
  vx*sigmax/rchisq(draws, vx)
}

#a) Determining the prior distribution
delta <- 0.01
times <- seq(delta, 1, delta)
nDraws  <- 5000
posteriorDraws <- 5000
mu0    <- as.matrix(c(-10, 150, -150))   # From tuning the plotted polynomial
omega0 <- 1*diag(3)           # same variance for betas, we dont know much
v0      <- 20
sigma0  <- 2
sigma2s <- rinvchisq(v0, sigma0, nDraws)#v0*sigma0/rchisq(nDraws, v0)
n       <- dim(X)[1]

glmthing <- lm(temp ~ time + I(time^2), data = data)
summary(glmthing)

# Plot of prior variance
plot.prior.varaince <- function() {
  plot(density(sigma2s),
       main = "Sigma^2 prior",
       xlab = expression(sigma^2),
       ylab = "Density",
       type = 'l')
}

# Thetas are matrix of theta - all parameters with priors
thetas <- cbind(t(sapply(sigma2s, function(sigma2) {
  rmvnorm(1, mu0, sigma2*solve(omega0))
})), sigma2s)

# y is the prior distribution of the model
prior.temp <- apply(thetas[seq(1, 10),], 1, function(theta) {
  sapply(times, function(time) {
    theta[1] + theta[2]*time + theta[3]*time^2 #+ rnorm(1, 0, theta[4])
  })
})

#b) Check if prior distribution is sensible
m <- dim(prior.temp)[1]
x.axis <- (1:m) / m

plot.prior.regression.curves <- function () {
  plot(x.axis, prior.temp[,1],
```

```r
       type = 'l',
       ylim = c(-15, 40),
       xlab = "Time",
       ylab = "Temperature",
       main = "Prior regression curves")
  for (i in 2:10) {
    lines(x.axis, prior.temp[,i])
  }
}

# c) Simulating from posterior distribution

# Posterior mapping
beta.hat <- solve(t(X)%*%X)%*%t(X)%*%Y # OLS
A   <- t(X) %*% X
B   <- as.numeric(t(Y) %*% Y)
mun      <- solve(A + omega0) %*% (A%*%beta.hat + omega0%*%mu0)
omegan   <- A + omega0
vn       <- v0 + n
sigma2n  <- as.numeric((v0%*%sigma0 + (B + t(mu0)%*%omega0%*%mu0 - t(mun)%*%omegan%*%mun))/vn)

post.sigma2s <- rinvchisq(vn, sigma2n, posteriorDraws)
post.thetas  <- cbind(t(sapply(post.sigma2s, function(post.sigma2) {
  rmvnorm(1, mun, post.sigma2 * solve(omegan))
})), post.sigma2s)

post.betas <- post.thetas[,1:3]
theta.mean <- apply(post.thetas, 2, mean)
post.pred.mean <- X  %*% as.matrix(theta.mean[1:3])

nSamples <- 5000
bounds <- as.matrix(sapply(times, function(t) {
  ys <- as.matrix(apply(post.betas[1:nSamples,], 1, function(beta) {
    c(1, t, t^2) %*% as.matrix(beta)
  }))

  lower.bound <- nSamples * 0.05
  upper.bound <- nSamples * 0.95
  bounded.y   <- sort(ys)[lower.bound:upper.bound]
  list(upper = head(bounded.y, 1), lower = tail(bounded.y, 1))
}))

plot.posterior.betas <- function() {
  plot(data$time,
       data$temp,
       ylim = c(-20,25),
       col  = "Gray",
       xlab = "Time",
       ylab = "Temperature",
       main = "Posterior mean and credible intervals")
  lines(data$time, post.pred.mean, col = "Gray")
  lines(x=times,y=bounds[1,], col = 'Blue')
  lines(x=times,y=bounds[2,], col = 'Red')
```

```r
  legend("topright",
         c("5% credible interval", "95% credible interval", "Posterior mean"),
         lty=1,
         col=c("Blue", "Red", "Black"),
         bty='n',
         cex=.75)
  }

# d)
x_tilde <- which.max(post.pred.mean)
# Hottest day of the year (# 197)


# Distribution over the hottest day
day <- as.matrix(apply(post.betas, 1, function(beta) {
  (-beta[2] / (2*beta[3]))*n
}))
mean(day)
plot.day.density <- function() {
  plot(density(day),
       main = "Density of warmest day of the year for different betas",
       xlab = "Day",
       ylab = "Temperature")
}

# e)
# mu0: set the last 4 values to 0 to delete the effect of the higher order terms
# omega: set the 4 last values to very large to prevent high variance for the higher order terms


# 2. Posterior approximation for classification with logistic regression

# a)
data2 <- read.table("WomenWork.dat", header=TRUE)
y    <- as.vector(data2[,1])
X    <- as.matrix(data2[,-1])

glmModel <- glm(Work ~0 + ., data=data2, family = binomial)
#summary(glmModel)

# b)
# want posterior
# to get that first we need prior and likelihood

beta.log.posterior <- function(betas, X, y, mu, sigma) {
  d    <- length(betas)
  pred <- X %*% betas

  log.likelihood <- sum(y*pnorm(pred, log.p = TRUE) + (1-y)*pnorm(pred, log.p = TRUE, lower.tail = FALSE
  log.prior      <- dmvnorm(betas, matrix(mu, d, 1), sigma*diag(d), log = TRUE)

  log.likelihood + log.prior
}
```

```r
tau2 <- 10^2
mu   <- 0
d <- dim(X)[2]
beta.initial <- rep(0, d)
result       <- optim(beta.initial, beta.log.posterior, gr = NULL,
                      X, y, mu, tau2,
                      hessian = TRUE,
                      method  = c("BFGS"),
                      control = list(fnscale=-1))

beta_hat <- result$par
J        <- result$hessian

# numerical values should be in the report
beta.post <- rmvnorm(10000, beta_hat, -solve(J))

#NSmallChild
nsc <- beta.post[,7]
q1 <- quantile(nsc, c(0.025,0.975))
plot.density.nsc <- function() {
  plot(density(nsc),
       main = "Posterior density of beta corresponding to NSmallChild",
       xlab = expression(beta))
}

# compute 95% credible interval for NSmallChild?
x <- c(1, 10, 8, 10, 1, 40, 1, 1)
y_hat <- x %*% t(beta.post)
p <- 1 / (1 + exp(-y_hat))
plot.does.work <- function() {
  plot(density(p),
       xlab = "Work",
       main = "Predictive density of the probability of the woman working")
  sprintf("The mean probability that the woman works is %.2f", mean(p))
}
```