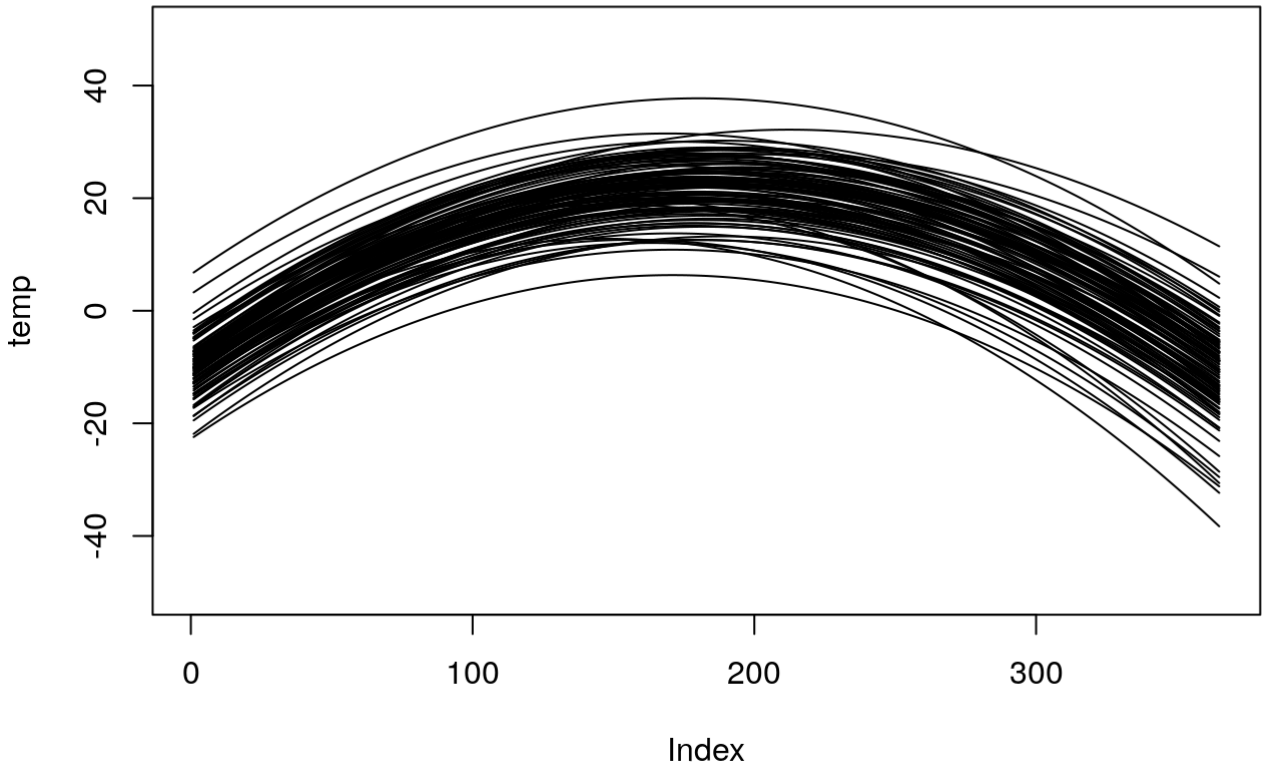
# Computer Lab 2

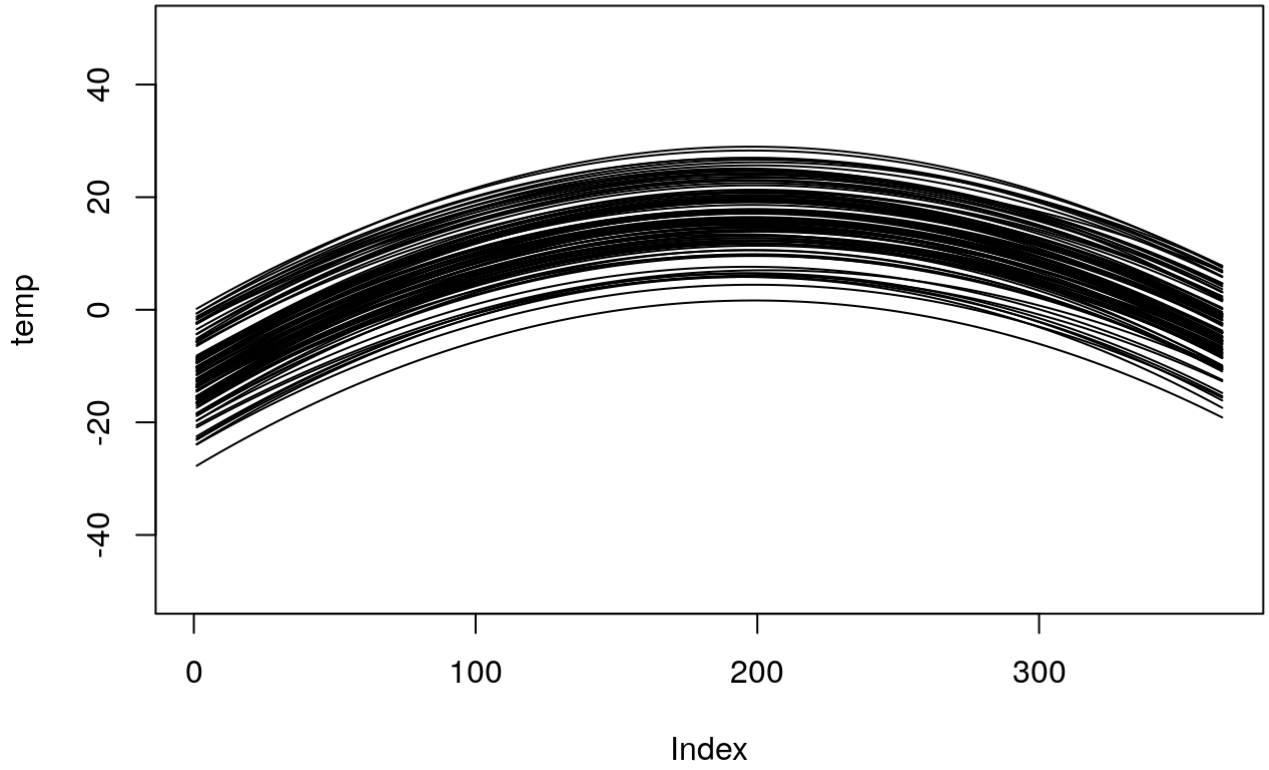
Dhyey Patel, Erik Anders

4/29/2020

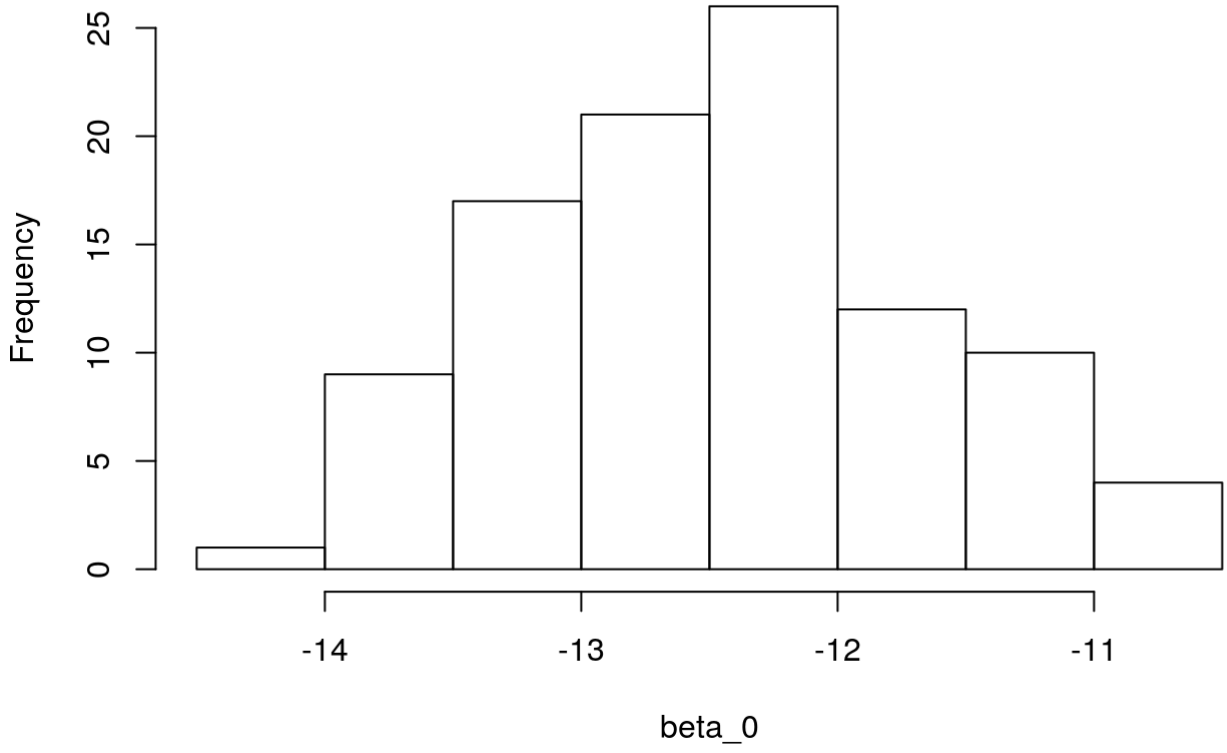
##1. Linear and polynomial regression (a) Determining the prior distribution of the model parameters. Use the conjugate prior for the linear regression model. Your task is to set the prior hyperparam- eters μ 0 , Ω 0 , ν 0 and σ 0 2 to sensible values. Start with μ 0 = (−10, 100, −100) T , Ω 0 = 0.01 · I 3 , ν 0 = 4 and σ 0 2 = 1. Check if this prior agrees with your prior opinions by simulating draws from the joint prior of all parameters and for every draw compute the regression curve. This gives a collection of regression curves, one for each draw from the prior. Do the collection of curves look rea- sonable? If not, change the prior hyperparameters until the collection of prior regression curves agrees with your prior beliefs about the regression curve. [Hint: the R package mvtnorm will be handy. And use your Inv-χ 2 simulator from Lab 1.]



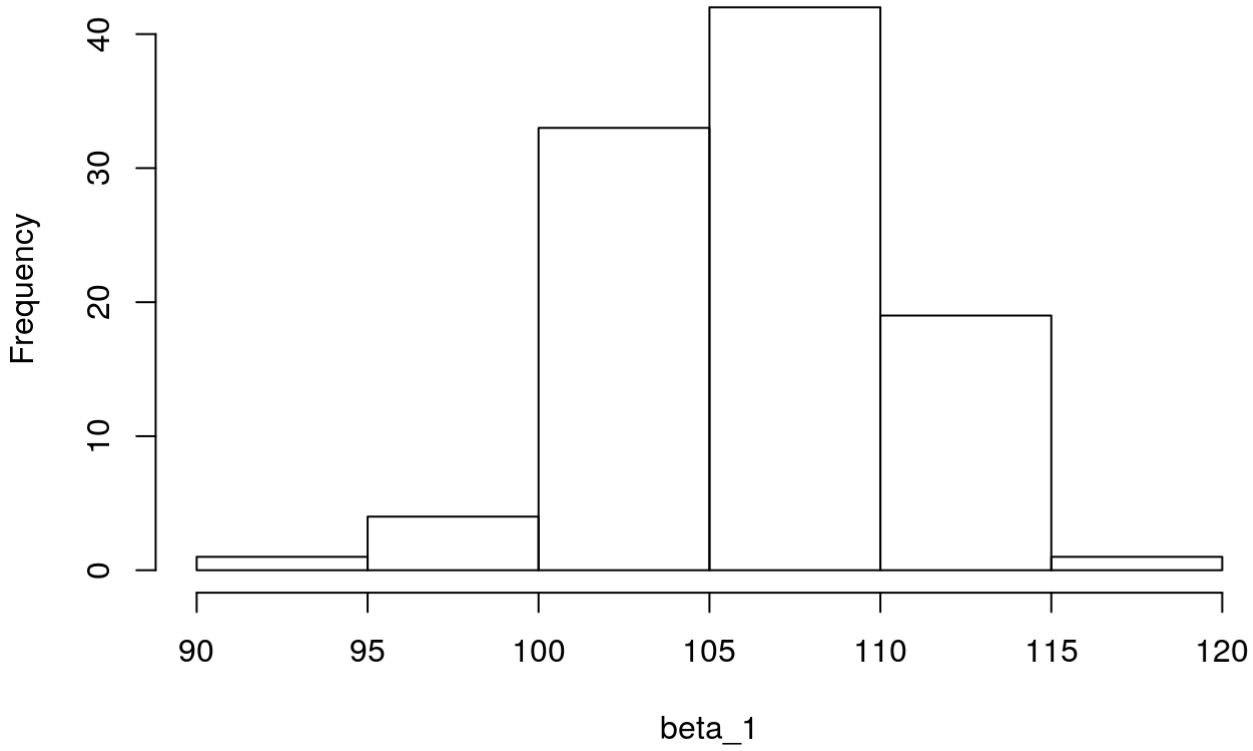

b. Write a program that simulates from the joint posterior distribution of β 0 , β 1 ,β 2 and σ 2 . Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function f (time) = β 0 +β 1 ·time+β 2 ·time 2 , computed for every value of time. Also overlay curves for the lower 2.5% and upper 97.5% posterior credible interval for f (time). That is, compute the 95% equal tail posterior probability intervals for every value of time and then connect the lower and upper limits of the interval by curves. Does the interval bands contain most of the data points? Should they?

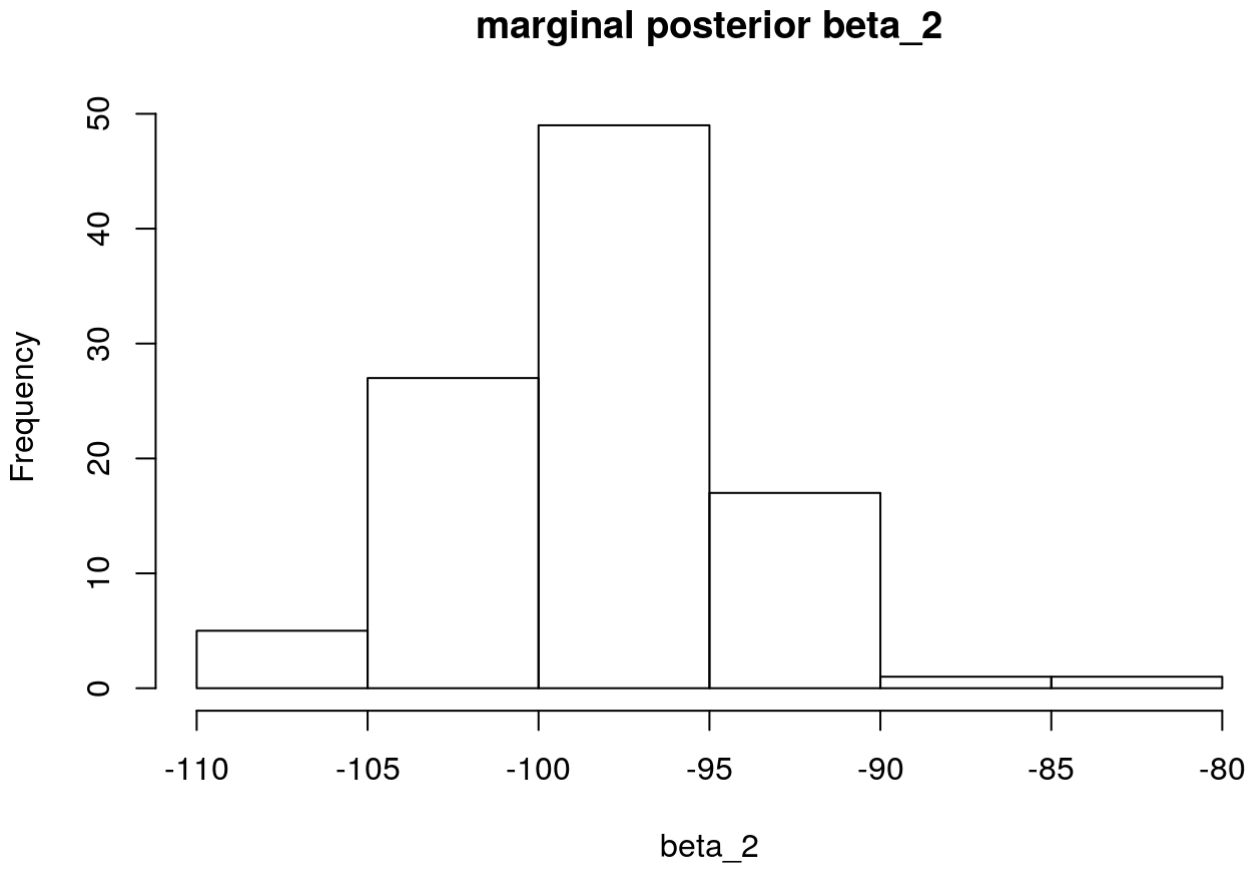


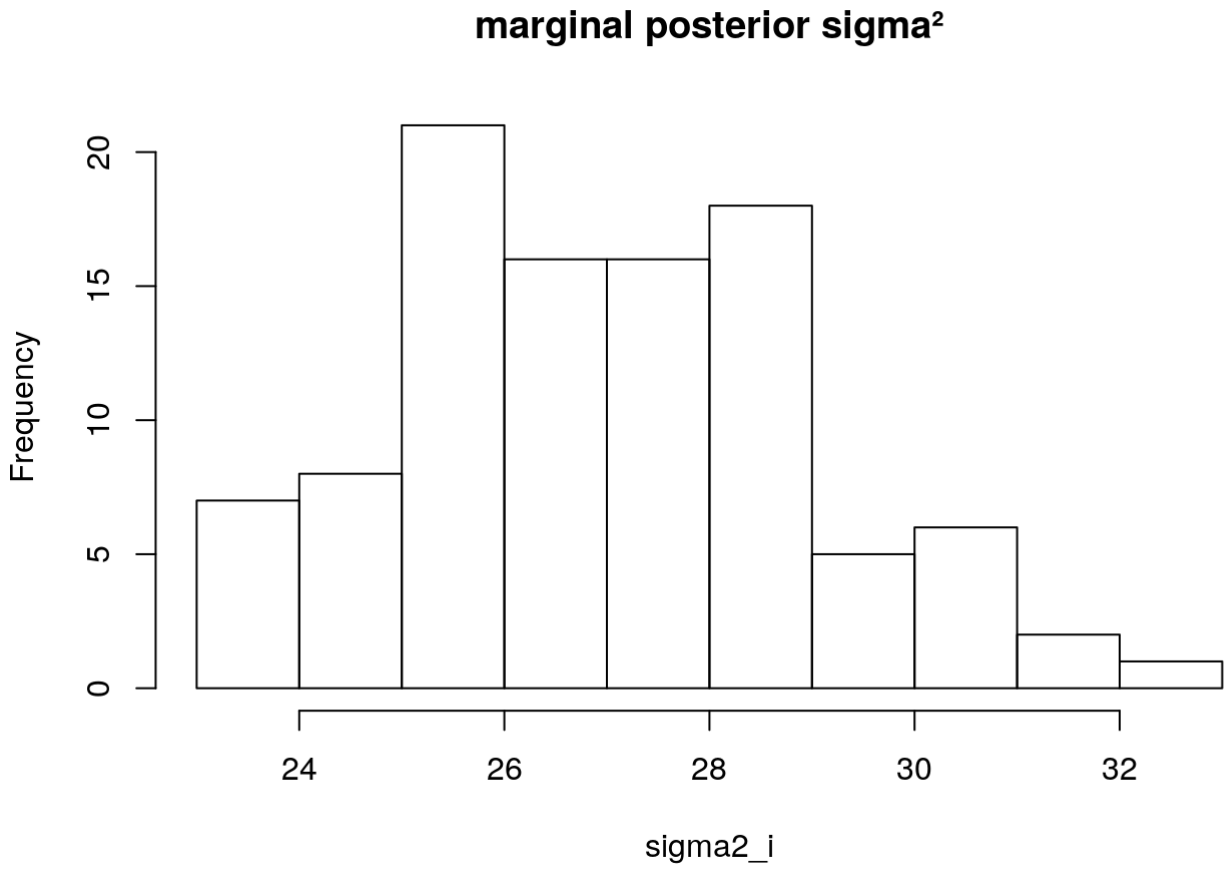

**marginal posterior beta_0**

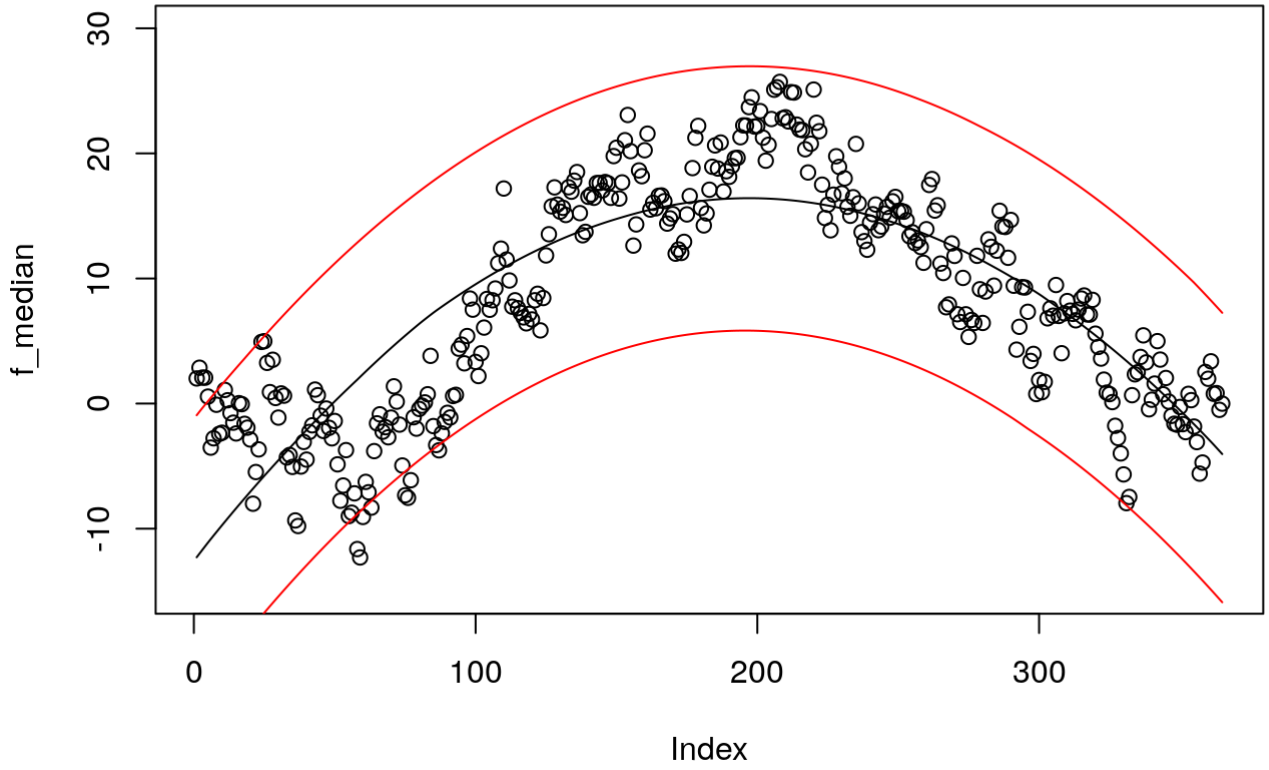



**marginal posterior beta_1**


Frequency
0
10
20
30
40
90
95
100
105
110
115
120
beta_1

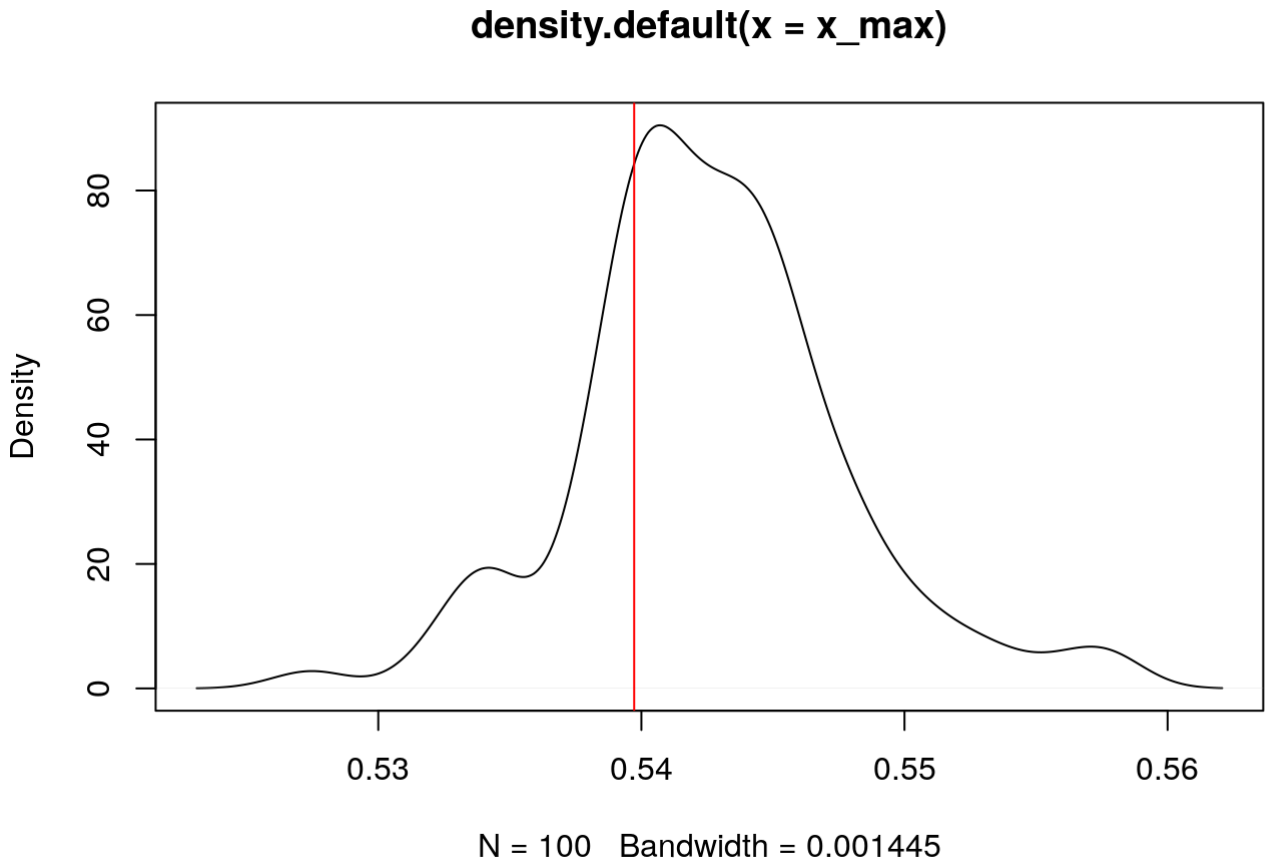

marginal posterior beta_2
Frequency
0
10
20
30
40
50
-110
-105
-100
-95
-90
-85
-80
beta_2


marginal posterior sigma²
Frequency
0
5
10
15
20
24
26
28
30
32
sigma2_i

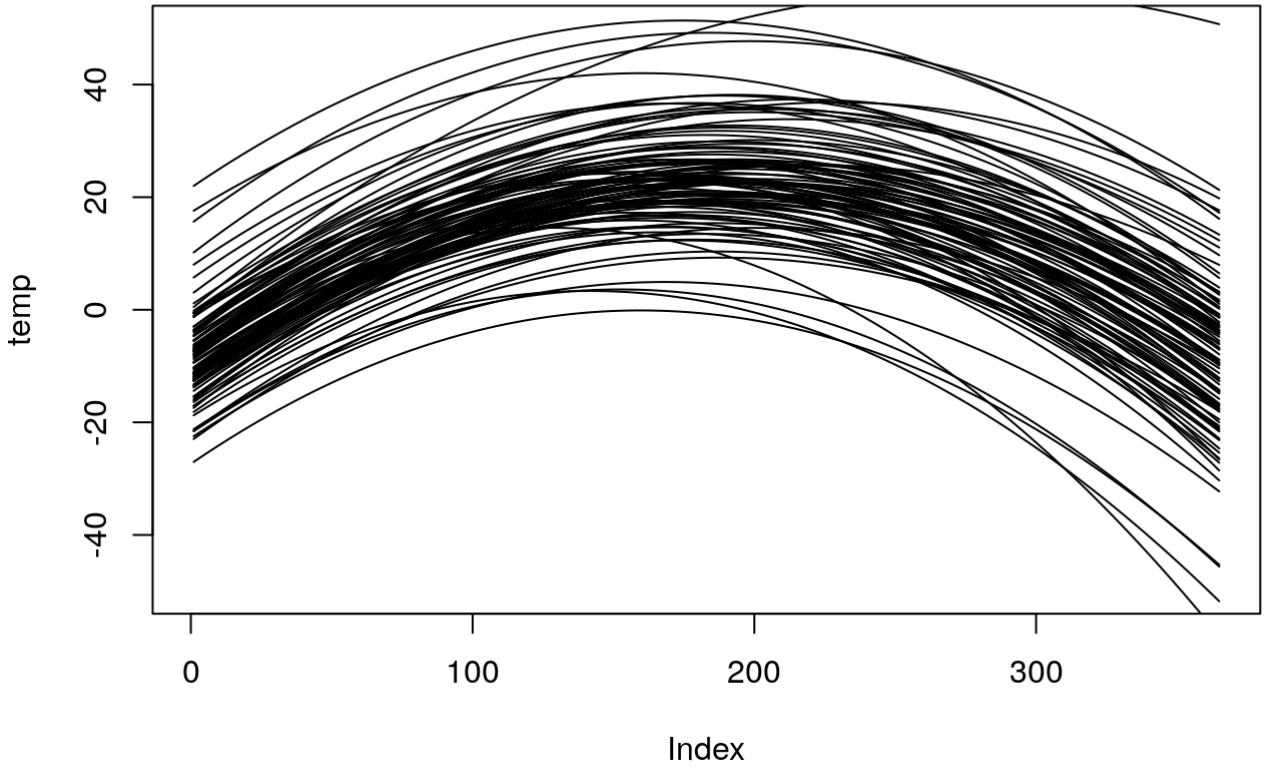



The inteval bands cover most of the points. The 95% credible intervals should contain most of the data points as that would indicate that the model is a good fit on the data. Also most of the predictions would be made within the 95% credible interval so the error calculated would be less when compared to the actual data points because most of the data points are inside the 95% credible band.

c. It is of interest to locate the time with the highest expected temperature (that is, the time where f (time) is maximal). Let's call this value x̃. Use the simulations in b) to simulate from the posterior distribution of x̃. [Hint: the regression curve is a quadratic. You can find a simple formula for x̃ given β 0 , β 1 and β 2 .]




d. Say now that you want to estimate a polynomial model of order 7, but you suspect that higher order terms may not be needed, and you worry about over- fitting. Suggest a suitable prior that mitigates this potential problem. You do not need to compute the posterior, just write down your prior. [Hint: the task is to specify μ 0 and Ω 0 in a smart way.]




We want to reduce the impact of higher order terms in the polynomial model of order 7. We have already seen in the first part that the intercept, time and time squared term are necessary and so we try to reduce the impact of the other higher order terms. We can do that by taking the means and variances close to 0. Hence we take mu0 as very small and omega0 as large values, as omega0 inverse is the multiplying factor in the variance. Hence large values of omega0 for the higher order terms will give very low variance.

# 2. Posterior approximation for classification with logistic regression

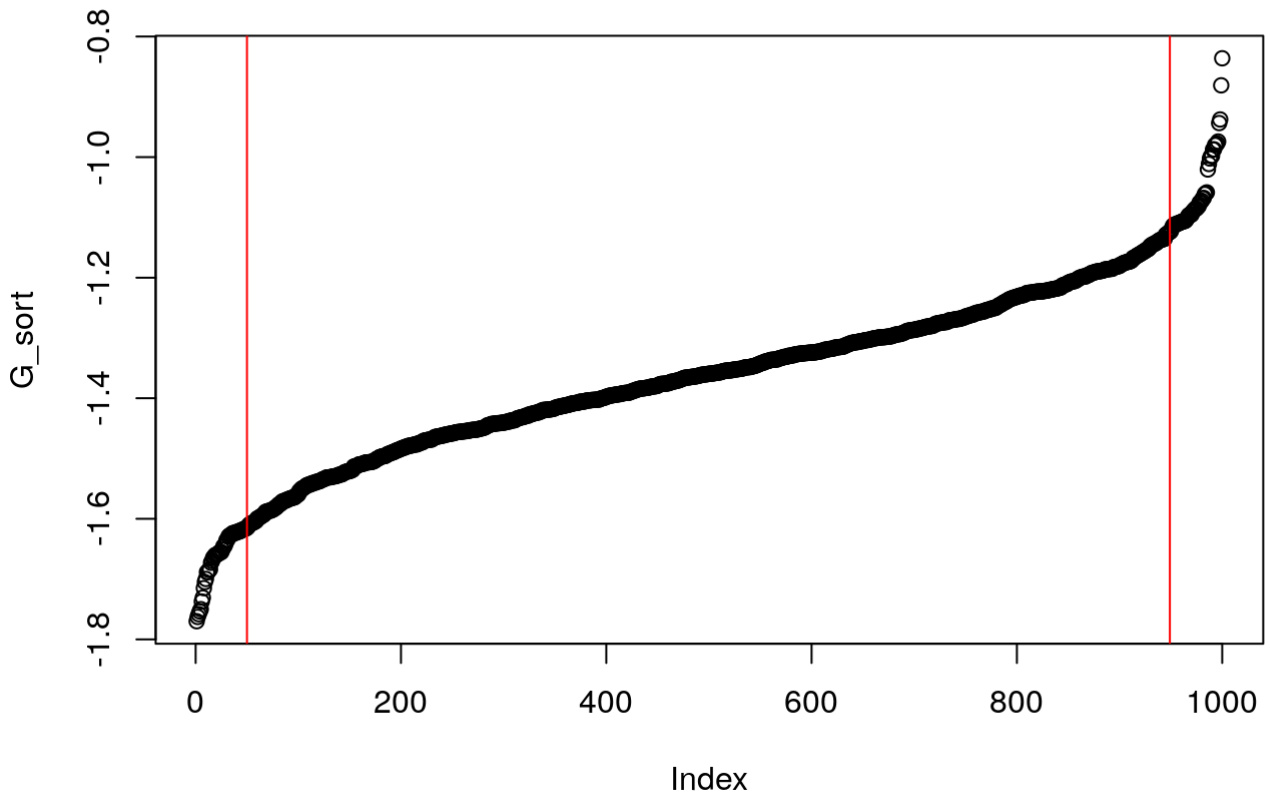
a. Consider the logistic regression ◁ exp x T β , Pr(y = 1|x) = 1 + exp (x T β) where y is the binary variable with y = 1 if the woman works and y = 0 if she does not. x is a 8-dimensional vector containing the eight features (including a one for the constant term that models the intercept). The goal is to approximate the posterior distribution of the 8-dim parameter vector β with a multivariate normal distribution ◄ ► β|y, X ∼ N β, ̃ J y −1 ( β ̃) , where β ̃is the posterior mode and J( β ̃) = − ∂ 2 ln p(β|y) | β= β ̃is the observed Hes- ∂β∂β T 2 ln p(β|y) sian evaluated at the posterior mode. Note that ∂ ∂β∂β is an 8×8 matrix with T 2 ln p(β|y) second derivatives on the diagonal and cross-derivatives ∂ ∂β on the off- i ∂β j diagonal. It is actually not hard to compute this derivative by hand, but don't worry, we will let the computer do it numerically for you. Now, both β ̃and J( β ̃) are computed by the optim function in R. See my code https://github.com/ (https://github.com/) mattiasvillani/BayesLearnCourse/raw/master/Code/MainOptimizeSpam. zip where I have coded everything up for the spam prediction example (it also does probit regression, but that is not needed here). I want you to implement you own version of this. You can use my code as a template, but I want you 2to write your own file so that you understand every line of your code. Don't just copy my code. Use the prior β ∼ N (0, τ 2 I), with τ = 10. Your report should include your code as well as numerical values for β ̃and J y −1 ( β ̃) for the WomenWork data. Compute an approximate 95% credible in- terval for the variable NSmallChild. Would you say that this feature is an important determinant of the probability that a women works? [Hint: To verify that your results are reasonable, you can compare to you get by

estimating the parameters using maximum likelihood: glmModel <- glm(Work ~ 0 + ., data = WomenWork, family = binomial).]

```
## [1] "Optimal beta:"
```

```
## [1]  0.62672884 -0.01979113  0.18021897  0.16756670 -0.14459669 -0.08206561
## [7] -1.35913317 -0.02468351
```

```
##              [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  2.266022568  3.338861e-03 -6.545121e-02 -1.179140e-02  0.0457807243
## [2,]  0.003338861  2.528045e-04 -5.610225e-04 -3.125413e-05  0.0001414915
## [3,] -0.065451206 -5.610225e-04  6.218199e-03 -3.558209e-04  0.0018962893
## [4,] -0.011791404 -3.125413e-05 -3.558209e-04  4.351716e-03 -0.0142490853
## [5,]  0.045780724  1.414915e-04  1.896289e-03 -1.424909e-02  0.0555786706
## [6,] -0.030293450 -3.588562e-05 -3.240448e-06 -1.340888e-04 -0.0003299398
## [7,] -0.188748354  5.066847e-04 -6.134564e-03 -1.468951e-03  0.0032082535
## [8,] -0.098023929 -1.444223e-04  1.752732e-03  5.437105e-04  0.0005120144
##               [,6]          [,7]          [,8]
## [1,] -3.029345e-02 -0.1887483542 -0.0980239285
## [2,] -3.588562e-05  0.0005066847 -0.0001444223
## [3,] -3.240448e-06 -0.0061345645  0.0017527317
## [4,] -1.340888e-04 -0.0014689508  0.0005437105
## [5,] -3.299398e-04  0.0032082535  0.0005120144
## [6,]  7.184611e-04  0.0051841611  0.0010952903
## [7,]  5.184161e-03  0.1512621814  0.0067688739
## [8,]  1.095290e-03  0.0067688739  0.0199722657
```

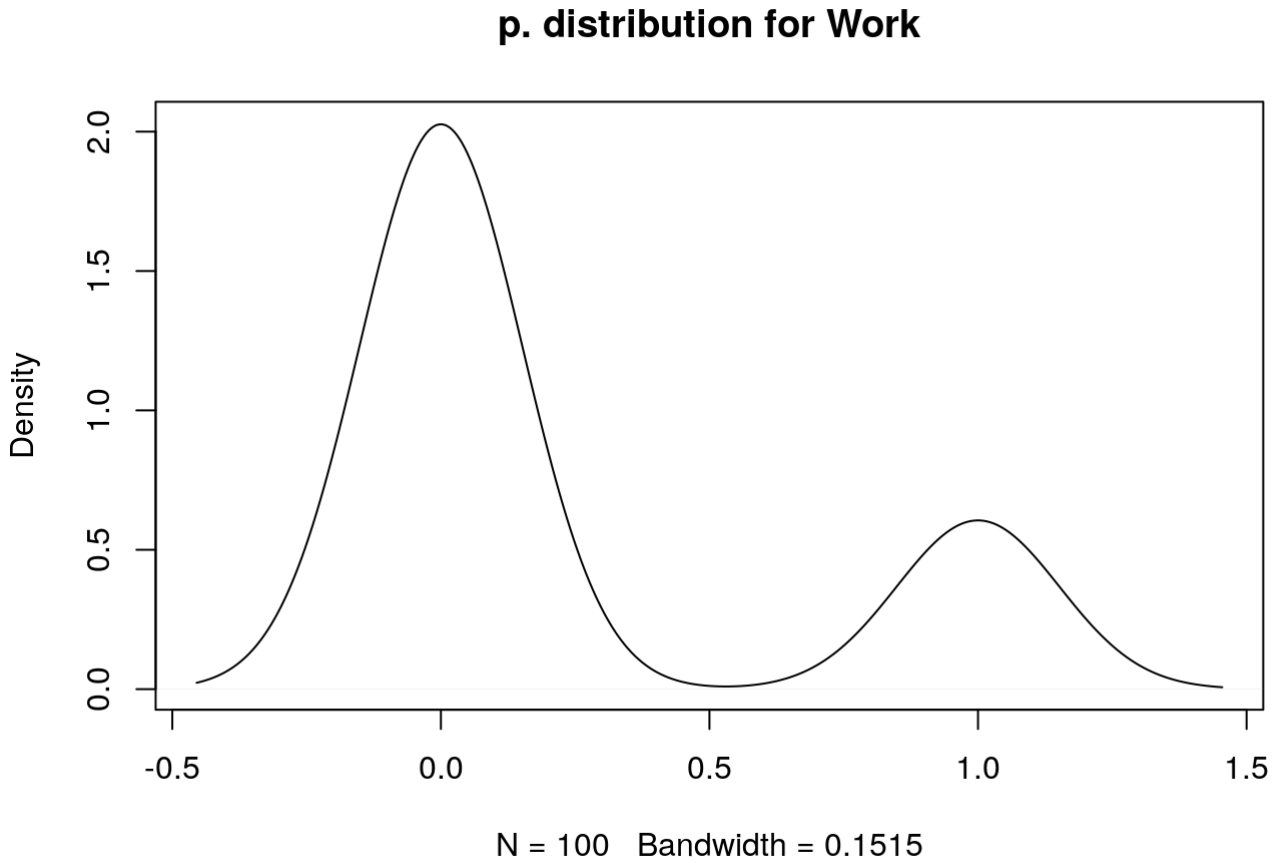


Yes it is an important feature, it's absolute beta value is the highest of the entire model. This means the value of the feature has a strong impact on the result.

```
##    Constant  HusbandInc   EducYears    ExpYears   ExpYears2         Age
##  0.64430363 -0.01977457  0.17988062  0.16751274 -0.14435946 -0.08234033
## NSmallChild   NBigChild
## -1.36250239 -0.02542986
```

Looking at the estimated parameters using maximum likelihood we can see that tey are very close the the ones we predicted.

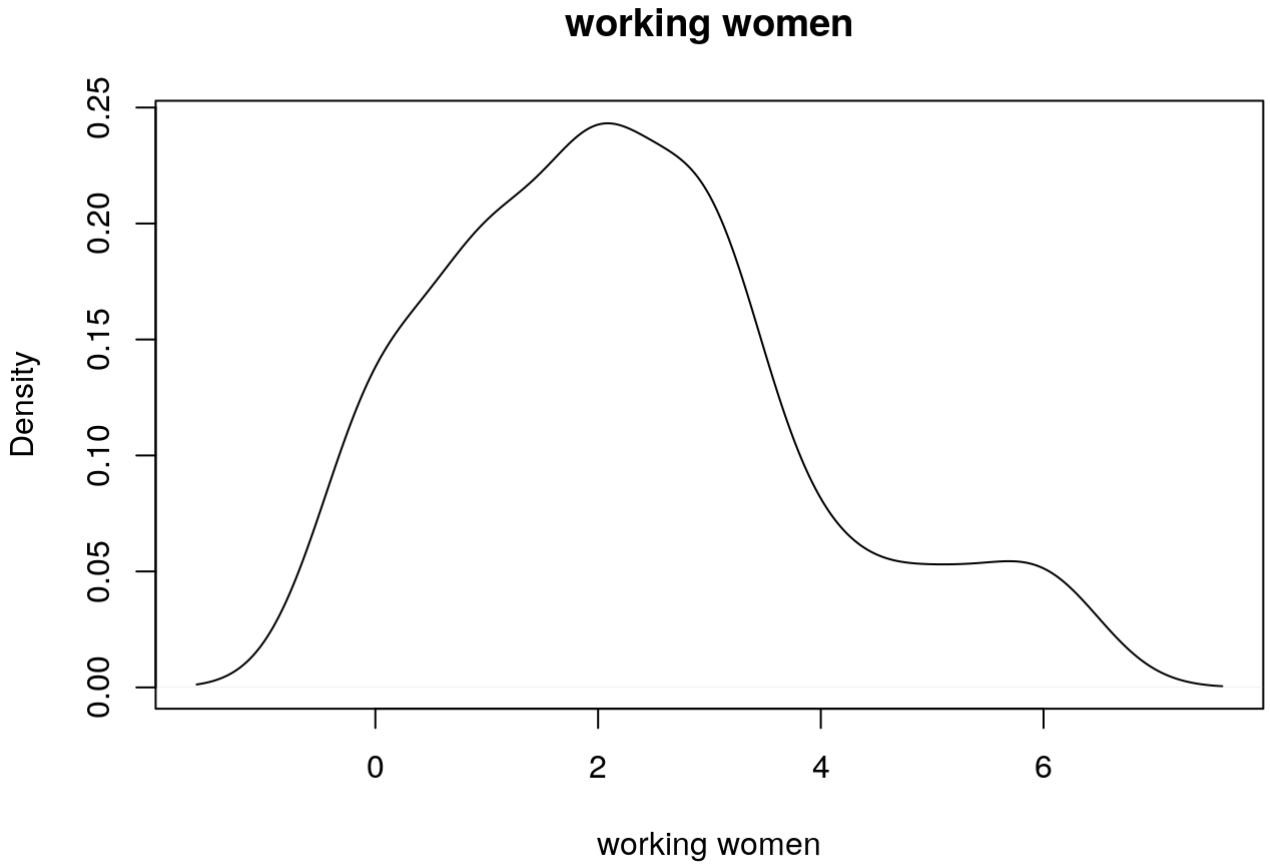
b. Write a function that simulates from the predictive distribution of the response variable in a logistic regression. Use your normal approximation from 2(a). Use that function to simulate and plot the predictive distribution for the Work variable for a 40 year old woman, with two children (3 and 9 years old), 8 years of education, 10 years of experience. and a husband with an income of 10. [Hints: The R package mvtnorm will again be handy. Remember my discussion on how Bayesian prediction can be done by simulation.]




```
## [1] 23
```

In our simulation 24% had a job.

c. Now, consider 10 women which all have the same features as the woman in 2(b). Rewrite your function and plot the predictive distribution for the number of women, out of these 10, that are working. [Hint: Which distribution can be described as a sum of Bernoulli random variables?]




##Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
#1.
#a).
n <- 1
data <- read.csv("/home/erik/Documents/SML/Semester 2/Bayesian Learning/Labs/Lab2/Tem
pLinkoping.txt", sep="")
X <- cbind(1,data$time,data$time^2)

mu_0 <- c(-10,130,-130)
sigma2_0 <- 1
omega_0 <- 0.1*diag(3)
vu_0 <- 4

library(MASS)
library(mvtnorm)

for (i in 1:100) {
  sigma2 <- (sigma2_0*vu_0)/rchisq(n = 1, df = vu_0)
  #beta_sigma2 <- mvrnorm(n = n, mu = mu_0, Sigma = sigma2*solve(omega_0))
  beta_sigma2 <- rmvnorm(n = 1, mean = mu_0, sigma = sigma2*solve(omega_0))

  #temp <- X%*%beta_sigma2
  E <- rnorm(n = 1, mean = 0,sd = sqrt(sigma2))
  temp <- X%*%t(beta_sigma2) + E

  if (i == 1) {
    plot(temp, type = "l", ylim = c(-50,50))
  }
  else{
    lines(temp, type = "l")
  }
}
#b).
X <- cbind(1,data$time,data$time^2)
y <- data$temp
n <- length(data$time)

beta_hat <- solve(t(X) %*% X) %*% (t(X) %*% y)
mu_n <- solve(t(X)%*%X+omega_0)%*%(t(X)%*%X%*%beta_hat+omega_0%*%mu_0)
omega_n <- t(X)%*%X+omega_0
vu_n <- vu_0 + n #what n?
vu_nsigma2 <- vu_0*sigma2_0 + (t(y)%*%y+t(mu_0)%*%omega_0%*%mu_0-t(mu_n)%*%omega_n%*%
mu_n)

sigma2_n <- var(data$temp)
sigma2_i <- c()
beta_0 <- c()
beta_1 <- c()
beta_2 <- c()
f_time <- c()
f_median <- c()

for (i in 1:100) {
  sigma2 <- (vu_nsigma2)/rchisq(n = 1, df = vu_n)
  beta_sigma2 <- rmvnorm(n = 1, mean = mu_n, sigma = sigma2[1,1]*solve(omega_n))

  #temp <- X%*%beta_sigma2
```

```
  E <- rnorm(n = 1, mean = 0,sd = sqrt(sigma2))
  temp <- X%*%t(beta_sigma2) + E
  f_time <- cbind(f_time, as.vector(temp))

  if (i == 1) {
    plot(temp, type = "l", ylim = c(-50,50))
  }
  else{
    lines(temp, type = "l")
  }
  beta_0[i] <- beta_sigma2[1,1]
  beta_1[i] <- beta_sigma2[1,2]
  beta_2[i] <- beta_sigma2[1,3]
  sigma2_i[i] <- sigma2
}
points(f_median)


hist(beta_0, main = "marginal posterior beta_0")
hist(beta_1, main = "marginal posterior beta_1")
hist(beta_2, main = "marginal posterior beta_2")
hist(sigma2_i, main = "marginal posterior sigma²")

for (i in 1:365) {
  f_median[i] <- median(f_time[i,])
}

#95% equal tail credible interval
G <- f_time[,1]


interval <- apply(f_time, MARGIN = 1, quantile, probs = c(0.025, 0.975))
plot(f_median, type = "l", ylim = c(-15,30))
points(data$temp, type = "p")
points(c(t(interval)[,1]), type = "l", col="red")
points(c(t(interval)[,2]), type = "l", col="red")

#c).
x_max <- -beta_1/(2*beta_2)
plot(density(x_max))
abline(v=data$time[which.max(f_median)], col = "red")
X <- cbind(1,data$time)
for (i in 2:7) {
  X <- cbind(X,data$time^i)
}

mu_0 <- c(-10,130,-130, rep(0.01,5))
sigma2_0 <- 1
omega_0 <- c(rep(0.02,3),rep(10,5))*diag(8)
vu_0 <- 4

for (i in 1:100) {
  sigma2 <- (sigma2_0*vu_0)/rchisq(n = 1, df = vu_0)
  #beta_sigma2 <- mvrnorm(n = n, mu = mu_0, Sigma = sigma2*solve(omega_0))
  beta_sigma2 <- rmvnorm(n = 1, mean = mu_0, sigma = sigma2*solve(omega_0))

  #temp <- X%*%beta_sigma2
  E <- rnorm(n = 1, mean = 0,sd = sqrt(sigma2))
```

```
  temp <- X%*%t(beta_sigma2) + E

  if (i == 1) {
    plot(temp, type = "l", ylim = c(-50,50))
  }
  else{
    lines(temp, type = "l")
  }
}

#2.
#a).
data <- read.table("/home/erik/Documents/SML/Semester 2/Bayesian Learning/Labs/Lab2/W
omenWork.dat", header = TRUE)
n <- dim(data)[1]

x <- as.matrix(data[,2:9])
y <- as.vector(data[,1])

nPara <- dim(x)[2];

# Setting up the prior
mu <- as.vector(rep(0,nPara))
tau <- 10
sigma <- tau^2*diag(nPara)


LogPostLogistic <- function(betaVect,y,X,mu,Sigma){

  nPara <- length(betaVect);
  linPred <- X%*%betaVect;

  # evaluating the log-likelihood
  logLik <- sum( linPred*y -log(1 + exp(linPred)));
  if (abs(logLik) == Inf) logLik = -20000; # Likelihood is not finite, stear the opti
mizer away from here!

  # evaluating the prior
  logPrior <- dmvnorm(betaVect, matrix(0,nPara,1), Sigma, log=TRUE);

  # add the log prior and log-likelihood together to get log posterior
  return(logLik + logPrior)
}


#initVal <- as.vector(rmvnorm(n = 1, mean = rep(0,dim(x)[2]), sigma = tau^2*diag(dim
(x)[2])))
initVal <- as.vector(rep(0,dim(x)[2]))
posterior <- optim(initVal,LogPostLogistic,gr=NULL,y,x,mu,sigma,method=c("BFGS"),cont
rol=list(fnscale=-1),hessian=TRUE)
posterior_mode <- posterior$par
hessian_matrix <- posterior$hessian
hessian_sigma <- -solve(posterior$hessian)

print("Optimal beta:")
print(posterior_mode)

#
```

```
print(hessian_sigma)

#95% equal tail credible interval
for (i in 1:1000) {
  G[i] <- rnorm(1, posterior_mode[7], hessian_sigma[7,7])
}

G_sort <- sort(G)
#G_cut <- G_sort[(length(G_sort)*0.05):(length(G_sort)*0.95-1)]
plot(G_sort)
abline(v=(length(G_sort)*0.05), col = "red")
abline(v=length(G_sort)*0.95-1, col = "red")
#Yes it is an important feature, it's absolute beta value is the highest of the entir
e model.
#This means the value of the feature has a strong impact on the result.

glmModel <- glm(Work ~ 0 + ., data = data, family = binomial)
glmModel$coefficients

#Looking at the estimated parameters using maximum likelihood we can see that tey are
very close the the ones we predicted.

#b).
X <- c(1,10,8,10,1,40,1,1)
result <- c()
for (i in 1:100) {
  beta_posterior <- rmvnorm(1, posterior_mode, hessian_sigma)
  y <- exp(t(X) %*% t(beta_posterior))/(1+(exp(t(X) %*% t(beta_posterior))))
  result[i] <- rbinom(1,1, y)
}
plot(density(result), main = "p. distribution for Work")

sum(result==1)
#c).
X <- c(1,10,8,10,1,40,1,1)
result <- c()
for (i in 1:100) {
  beta_posterior <- rmvnorm(1, posterior_mode, hessian_sigma)
  y <- exp(t(X) %*% t(beta_posterior))/(1+(exp(t(X) %*% t(beta_posterior))))
  result[i] <- rbinom(1,10, y)
}
plot(density(result), main = "working women" , xlab = "working women")
```