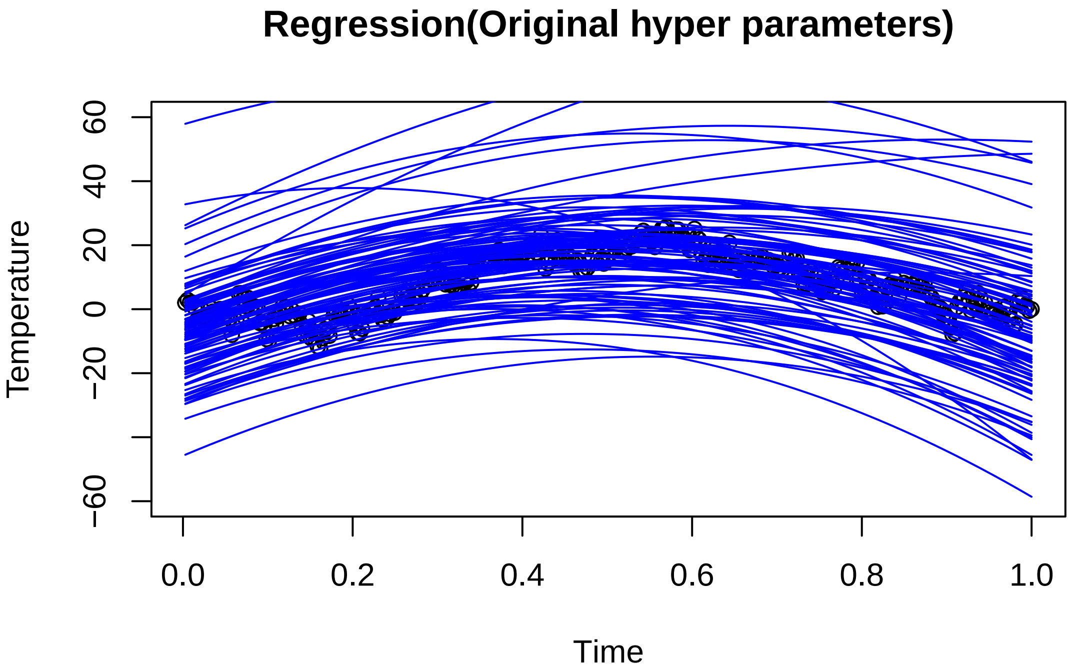
BL Lab2

Yash Pawar, Vyshnavi Pisupati

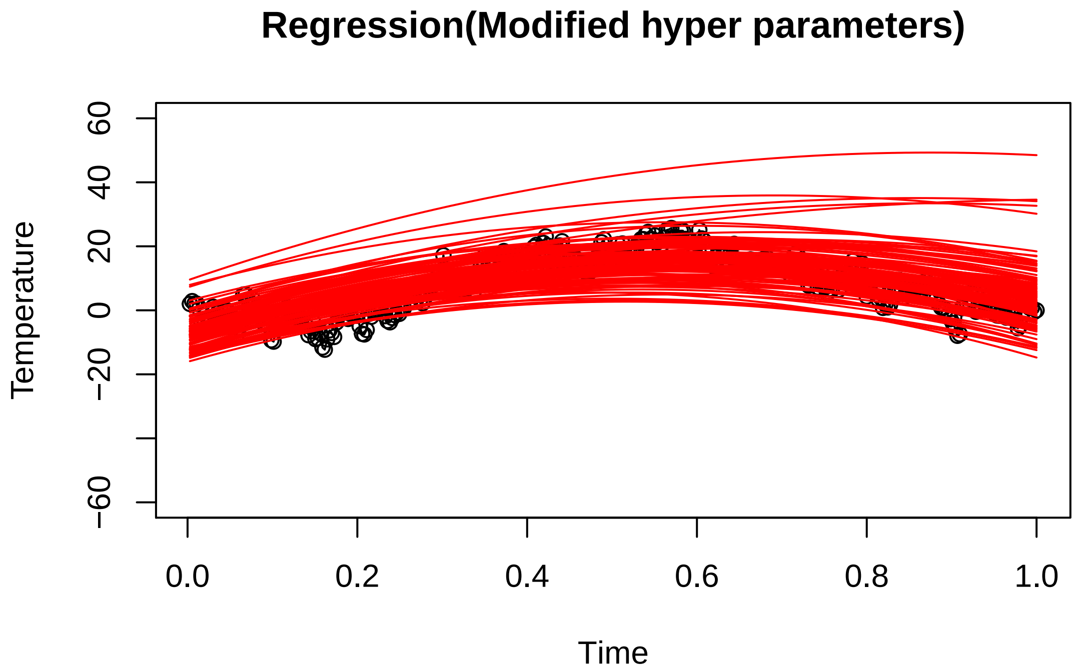
19/04/2020

Assignment 1.1

Following is the plot of Regression curve with the given hyper paramenters.



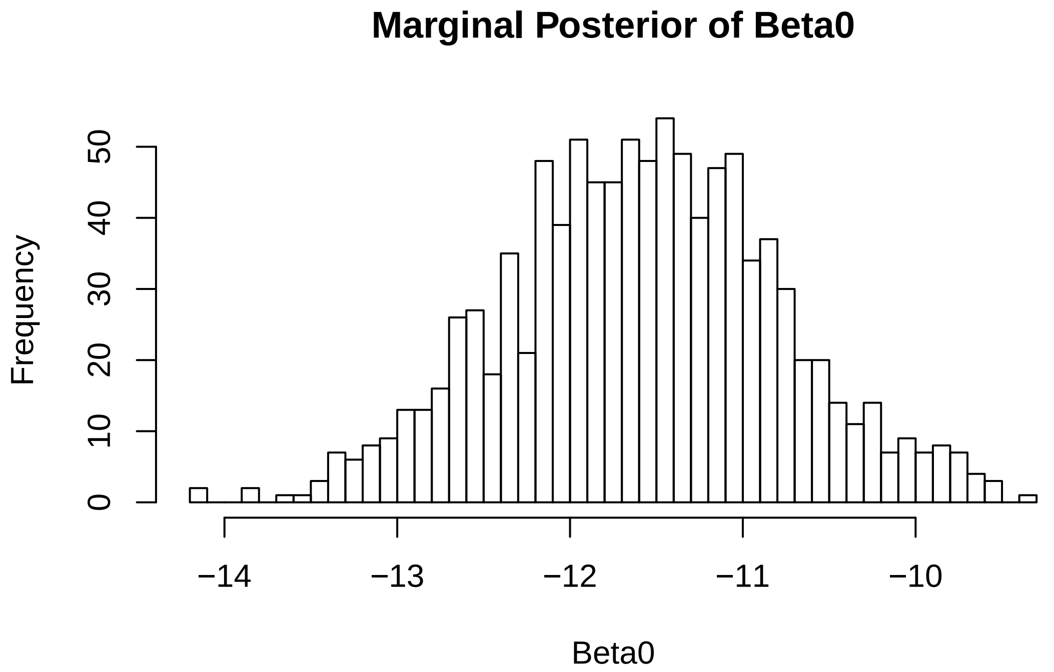
Following is the plot of regression curve with modified(resonably) hyper parameters. The changed $\mu_0 = (-8, 70, -60)$, $\sigma_0^2 = 1.2$, $\nu_0 = 3$, $\Omega_0 = 0.07$ justify the simulated prior values.



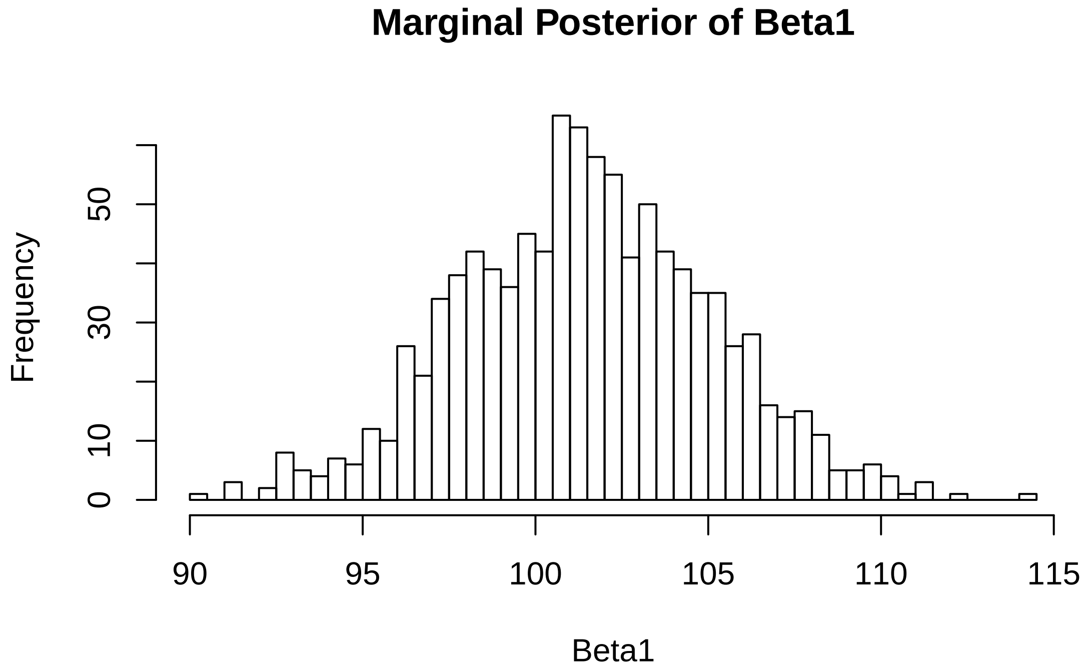
Assignment 1.2

The histogram plot for the marginal posteriors of $\beta_0, \beta_1, \beta_2, \sigma^2$ are as follows:

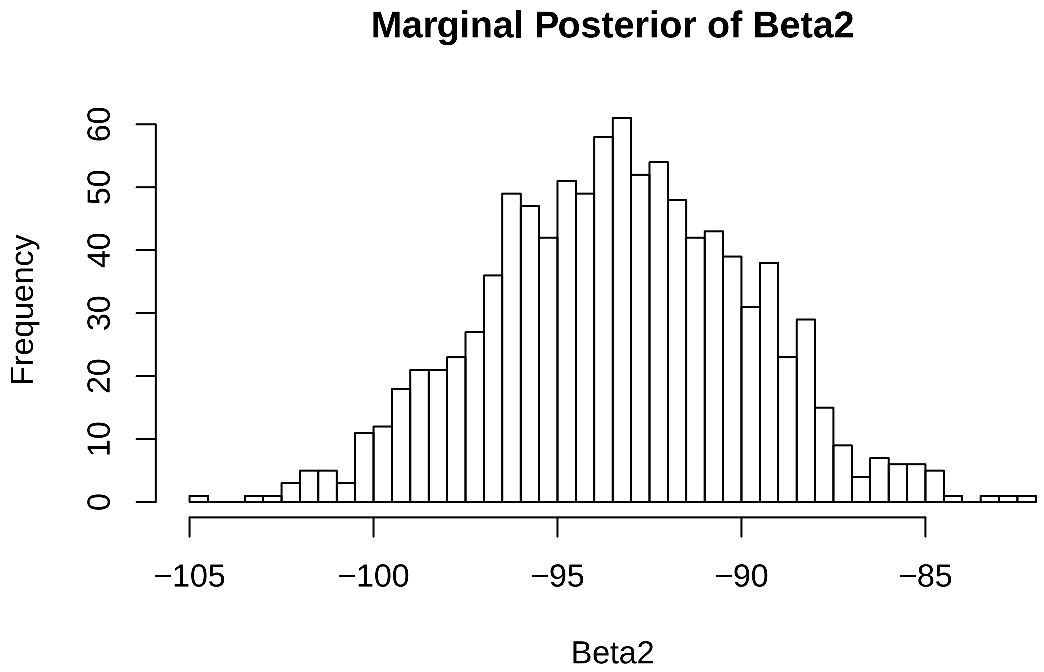
It can be seen that the posterior values of β follow a normal distribution distribution and σ^2 follows Inverse chi-square distribution.

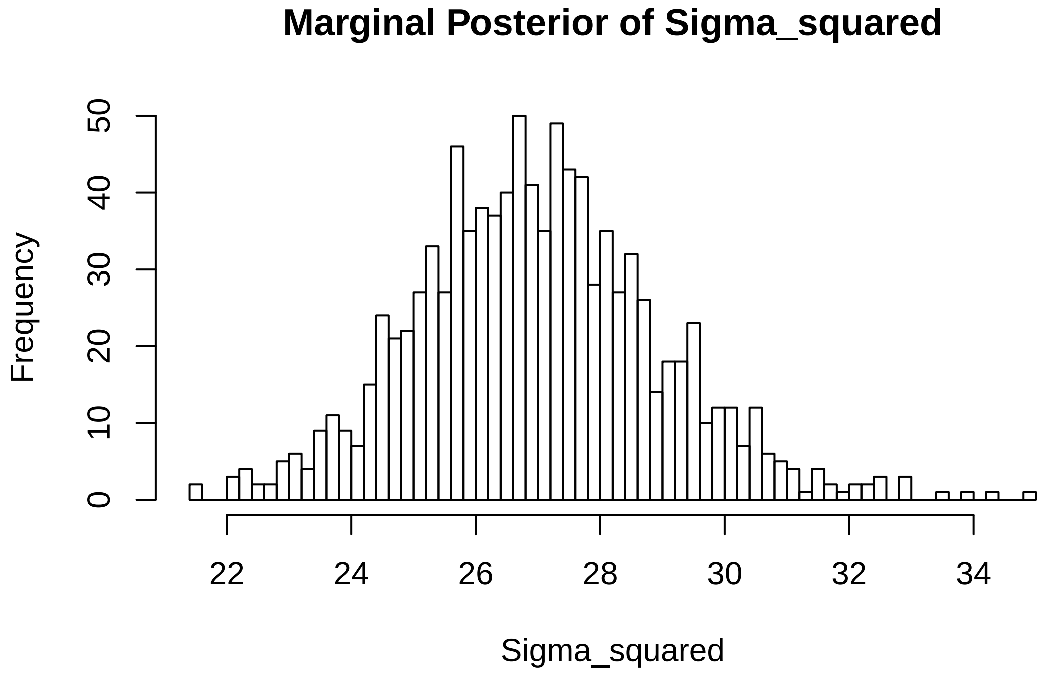


Marginal Posterior of Beta0
Frequency
0
10
20
30
40
50
−14
−13
−12
−11
−10
Beta0



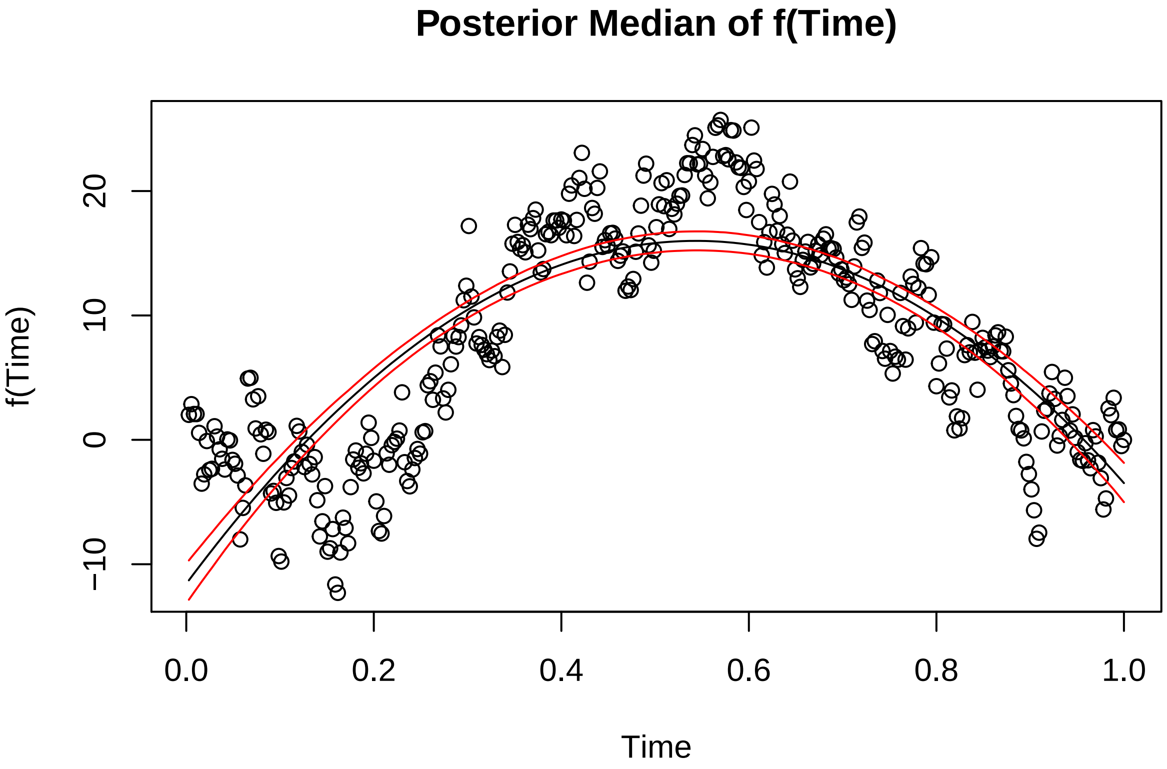
Marginal Posterior of Beta1
Frequency
0
10
30
50
90
95
100
105
110
115
Beta1





The following is the regression for Posterior median with its equal tail credible intervals of 95%. As opposed to the regression based on prior distribution, the posterior distribution is better fitted to the data.

It can be seen that the credible interval bands are quite narrow and they contain few data points. Narrow credible interval suggests better aproximation and less uncertainity in the predictions. As the parameters are determined from the posterior, they have much better approximation of the curve. The simulated posterior distribution has parameter values that are very close to each other hence their bands are quite narrow. Thus, it is quite justified that they do not contain most of the data points.



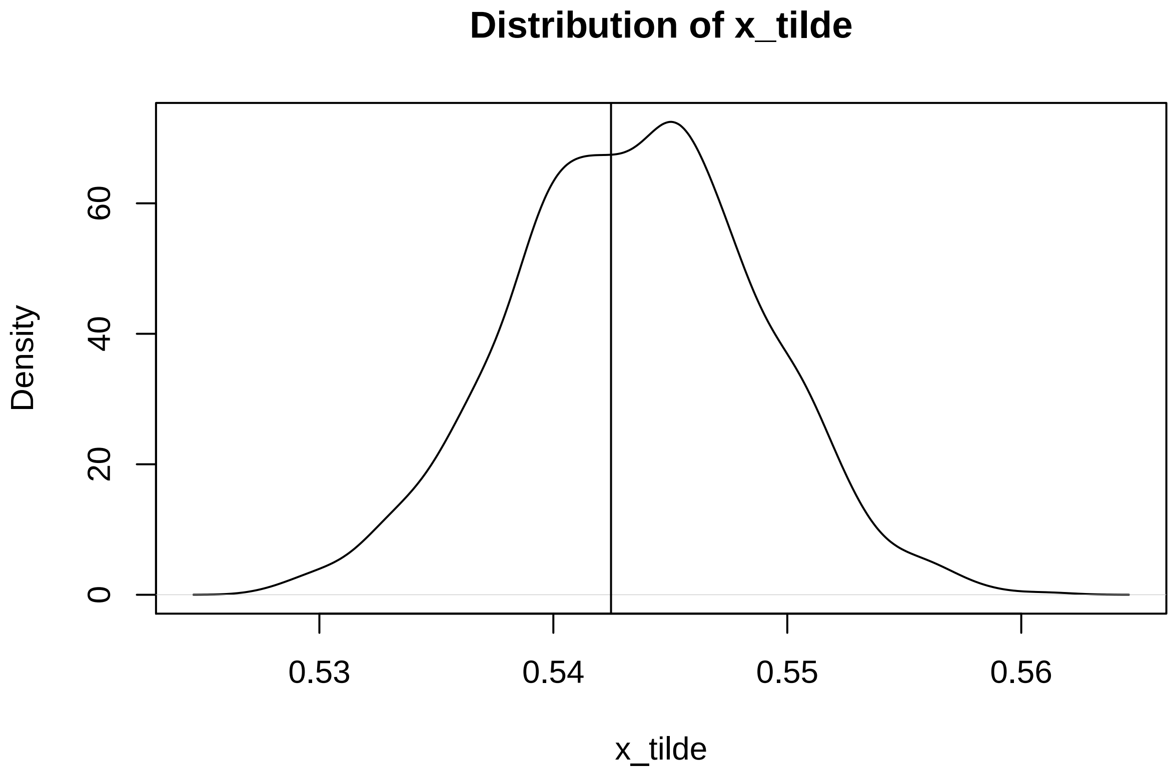
Assignment 1.3

Considering that the regression curve is quadratic. The eqaution for $\tilde{x}$ is:

$$\tilde{x} = -\beta_1/2\beta_2$$

.

The histogram for the distribution of $\tilde{x}$ is given as:



Assignment 1.4

The values that are choosen for

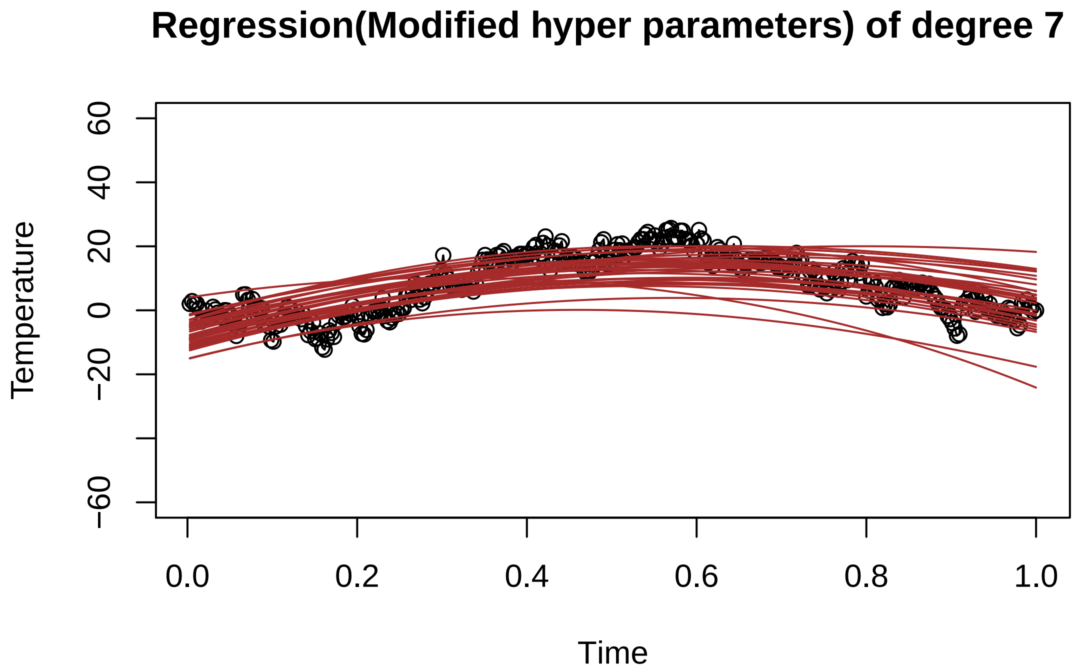
$$\mu_o = (-8, 70, -60, 0.01, 0.01, 0.01, 0.01, 0.01)$$

and

$$\Omega_o = (0.07, 0.07, 0.07, 10, 10, 10, 10, 10)$$

.

The hyper-parameter values are set in such a way that polynoimal regression parameters for higher degress are given less weights and thus the problem of over-fitting is solved.



Assignment 2.1

The function used to find the posterior for logistic regression and corresponding optim function to find the optimal $\tilde{\beta}$ and $J_y^{-}1(\tilde{\beta})$ is:

```
logistic_post = function(beta_v, x, y, mu, Sigma_log){

    nPara = length(beta_v)
    bx = x%*%beta_v

    loglik = sum(bx*y -  log(1 + exp(bx)))

    logPrior = dmvnorm(beta_v, matrix(0,nPara,1), Sigma_log, log=TRUE)

    return(loglik + logPrior)
}

initval = as.vector(rep(0,dim(Xl)[2]))

Optim_res = optim(initval,
                    logistic_post,
                    gr=NULL,Xl,Yl,mu,Sigma_log,
                    method=c("BFGS"),
                    control=list(fnscale=-1),
                    hessian=TRUE)
```

The optimal $\tilde{\beta}$ is:

0.6267288	-0.0197911	0.180219	0.1675667	-0.1445967	-0.0820656	-1.359133	-0.0246835

The $J_y^{-}1(\tilde{\beta})$ matrix is:

2.2660226	0.0033389	-0.0654512	-0.0117914	0.0457807	-0.0302934	-0.1887484	-0.0980239
0.0033389	0.0002528	-0.0005610	-0.0000313	0.0001415	-0.0000359	0.0005067	-0.0001444
-0.0654512	-0.0005610	0.0062182	-0.0003558	0.0018963	-0.0000032	-0.0061346	0.0017527
-0.0117914	-0.0000313	-0.0003558	0.0043517	-0.0142491	-0.0001341	-0.0014690	0.0005437
0.0457807	0.0001415	0.0018963	-0.0142491	0.0555787	-0.0003299	0.0032083	0.0005120
-0.0302934	-0.0000359	-0.0000032	-0.0001341	-0.0003299	0.0007185	0.0051842	0.0010953
-0.1887484	0.0005067	-0.0061346	-0.0014690	0.0032083	0.0051842	0.1512622	0.0067689
-0.0980239	-0.0001444	0.0017527	0.0005437	0.0005120	0.0010953	0.0067689	0.0199723

The approximate creble interval for the variable NSmallchild is:

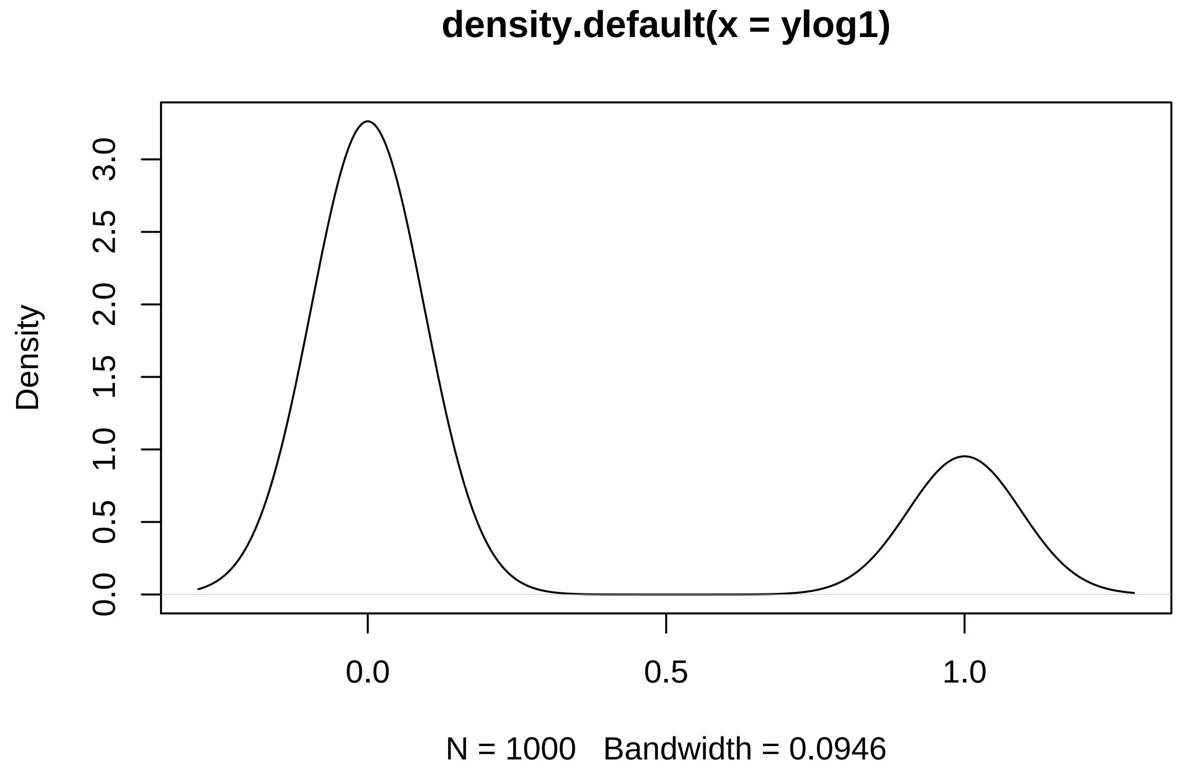
2.5%	97.5%
-1.653598	-1.057241

The interval suggests that the weight corrsponding to the feature has significant negative impact and thus it is an important determinant of the probablity that the woman works.

This is evident as women who have small children are less likely to work as compared to the woman who do not have a child.

Assignment 2.2

Following is the plot of predictive distribution for the variable "Work":



When the simulation is run 1000 times, it is found that 226 times the Woman works for the given set of features

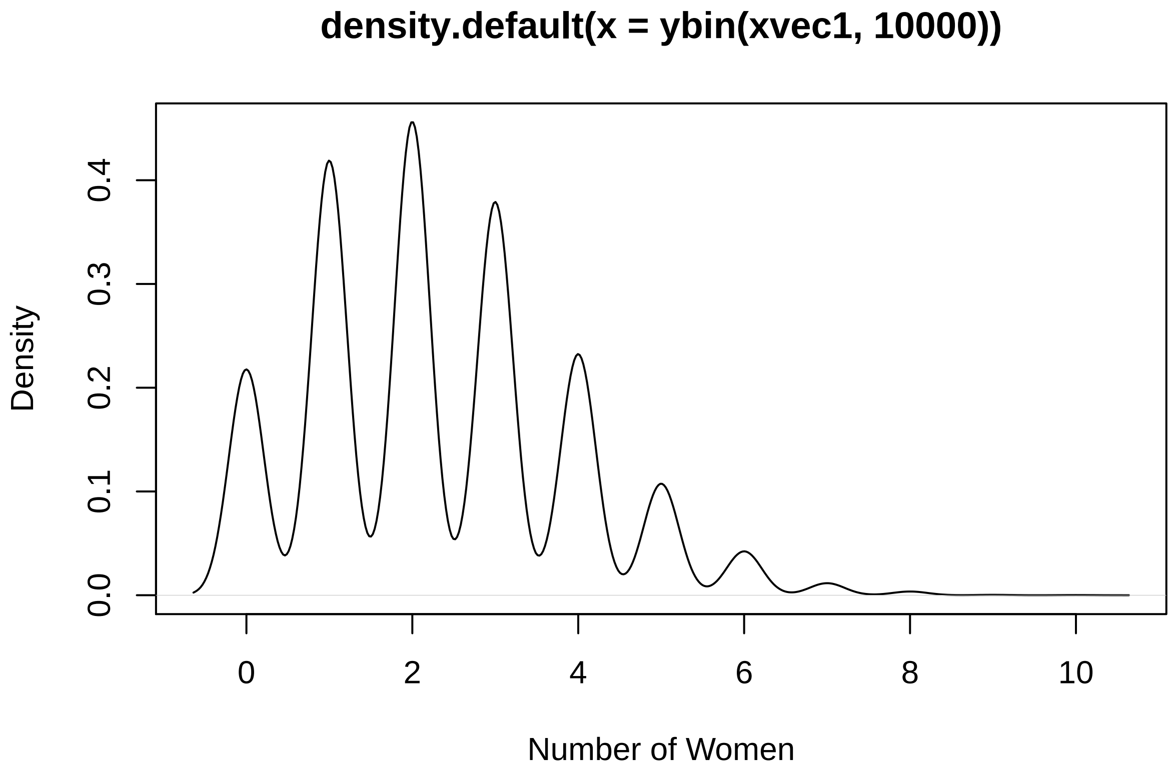
```r
length(which(ylog1==1))
```

```
## [1] 226
```

Assignment 2.3

The plot for the predictive distribution of the number of women out of 10 that are working is as follows:

It can be seen that the maximum number of women that would works based on the given data is 2.



Appendix:

```r
knitr::opts_chunk$set(echo = TRUE)
library(mvtnorm)
library(geoR)
library(ggplot2)
library(kableExtra)
temp_data = read.table("TempLinkoping.txt", header = TRUE)

mu_0 = c(-10,100,-100)
omg_0 = matrix(0, nrow = 3, ncol = 3)
diag(omg_0) = 0.01
v_0 = 4
sigma_sq0 = 1
Y = temp_data$temp
X = temp_data$time
n = dim(temp_data)[1]
```

```r
nsim = 100
Xp = cbind("Intercept" = rep(1,dim(temp_data)[1]),
                      "Time" = temp_data$time,
                      "Time_sq" = (temp_data$time)^2)
temp_res = matrix(0, nrow = dim(temp_data)[1], ncol = nsim)
b_0 = matrix(0, nrow = nsim, ncol = 3)
eps = matrix(0, nrow = dim(temp_data)[1], ncol = nsim)
set.seed(12345)
for (i in 1:nsim) {
 sigma_sq = (v_0* sigma_sq0)/rchisq(1, v_0)
 b_0[i,] = rmvnorm(1,mean = mu_0, sigma = (sigma_sq*solve(omg_0)))
 #eps[,i] = rnorm((dim(temp_data)[1]), mean = 0, sd = sqrt(sigma_sq))
 #temp_res = (Xp %*% t(b_0))
 #temp_res[,i] = temp_res[,i] + eps
}

temp_res = (Xp %*% t(b_0))

plot(X, Y, type = "o", ylim = c(-60,60), xlab = "Time", ylab = "Temperature")
title(main = "Regression(Original hyper parameters)")
for (i in 1:nsim) {
  points(X, temp_res[,i], type = "l",col = "blue")
}


library(mvtnorm)
library(geoR)
library(ggplot2)
temp_data = read.table("TempLinkoping.txt", header = TRUE)

mu_0 = c(-8,75,-65)
omg_0 = matrix(0, nrow = 3, ncol = 3)
diag(omg_0) = 0.07
v_0 = 3
sigma_sq0 = 1.2
Y = temp_data$temp
X = temp_data$time
n = dim(temp_data)[1]

nsim = 100
Xp = cbind("Intercept" = rep(1,dim(temp_data)[1]),
                      "Time" = temp_data$time,
                      "Time_sq" = (temp_data$time)^2)
temp_res = matrix(0, nrow = dim(temp_data)[1], ncol = nsim)
b_0 = matrix(0, nrow = nsim, ncol = 3)
eps = matrix(0, nrow = dim(temp_data)[1], ncol = nsim)
set.seed(12345)
for (i in 1:nsim) {
 sigma_sq = (v_0* sigma_sq0)/rchisq(1, v_0)
 b_0[i,] = rmvnorm(1,mean = mu_0, sigma = (sigma_sq*solve(omg_0)))
 #eps[,i] = rnorm((dim(temp_data)[1]), mean = 0, sd = sqrt(sigma_sq))
 #temp_res = (Xp %*% t(b_0))
 #temp_res[,i] = temp_res[,i] + eps
}

temp_res = (Xp %*% t(b_0))

plot(X, Y, type = "o", ylim = c(-60,60), xlab = "Time", ylab = "Temperature")
title(main = "Regression(Modified hyper parameters)")
for (i in 1:nsim) {
  points(X, temp_res[,i], type = "l",col = "red")
}

b_hat = solve(t(Xp)%*%Xp) %*% (t(Xp)%*%Y)
mu_n = solve((t(Xp)%*%Xp) + omg_0)%*%((t(Xp)%*%Xp)%*%b_hat + omg_0%*%mu_0)
omg_n = (t(Xp)%*%Xp) + omg_0
v_n = v_0 + n
sigma_sqn = (v_0*sigma_sq0 + (t(Y)%*%Y) + (t(mu_0)%*%omg_0%*%mu_0 - t(mu_n)%*%omg_n%*%mu_n))/v_n



ndraws = 1000
sigmasq_post1 = c()
b_post = matrix(0, nrow = ndraws, ncol = 3)

for (i in 1:ndraws) {
   sigmasq_post = (v_n* sigma_sqn)/rchisq(1, v_n)
   sigmasq_post1[i] = sigmasq_post[1,1]
   b_post[i,] = rmvnorm(1,mean = mu_n, sigma = sigmasq_post1[i]*solve(omg_n))

}

hist(b_post[,1], breaks = 50, xlab = "Beta0",
     main = "Marginal Posterior of Beta0")


hist(b_post[,2], breaks = 50, xlab = "Beta1",
     main = "Marginal Posterior of Beta1")


hist(b_post[,3], breaks = 50, xlab = "Beta2",
     main = "Marginal Posterior of Beta2")


hist(sigmasq_post1, breaks = 50, xlab = "Sigma_squared",
     main = "Marginal Posterior of Sigma_squared")
f_median = matrix(0, ncol = n)

f = Xp %*% t(b_post)
for (i in 1:n) {
        f_median[,i] = median(f[i,])

}

#colMeans(b_post)
#dim(f)
```

```
#quantile()
cred_int = t(apply(f, MARGIN = 1, quantile, probs = c(0.025, 0.975)))
plot(x = X, y = f_median, type = "l", ylim = c(min(Y), max(Y)),
     xlab = "Time",
     ylab = "f(Time)")
title(main = "Posterior Median of f(Time)")
points(x = X, y = Y)
points(x = X, cred_int[,2], type = "l", col = "Red")
points(x = X, cred_int[,1], type = "l", col = "Red")


x_bar = -b_post[,2]/(2*b_post[,3])
plot(density(x_bar), breaks = 50, xlab = "x_tilde", main = "Distribution of x_tilde")
abline(v = X[which.max(f_median)])

mu_01 = c(-8,70,-60, 0.01, 0.01,0.01,0.01,0.01)
omg_01 = matrix(0, nrow = 8, ncol = 8)
diag(omg_01) = c(0.07, 0.07,0.07, 10, 10, 10, 10,10)

nsim = 30

Xp1 = cbind(rep(1,dim(temp_data)[1]))
for (i in 1:7) {
    Xp1 = cbind(Xp1, (temp_data$time)^i)
}

temp_res1 = matrix(0, nrow = dim(temp_data)[1], ncol = nsim)
b_01 = matrix(0, nrow = nsim, ncol = 8)
#eps = matrix(0, nrow = dim(temp_data)[1], ncol = nsim)
set.seed(12345)
for (i in 1:nsim) {
 sigma_sq1 = (v_0* sigma_sq0)/rchisq(1, v_0)
 b_01[i,] = rmvnorm(1,mean = mu_01, sigma = (sigma_sq1*solve(omg_01)))
 #eps[,i] = rnorm((dim(temp_data)[1]), mean = 0, sd = sqrt(sigma_sq))
 #temp_res = (Xp %*% t(b_0))
 #temp_res[,i] = temp_res[,i] + eps
}

temp_res1 = (Xp1 %*% t(b_01))

plot(X, Y, type = "o", ylim = c(-60,60), xlab = "Time", ylab = "Temperature")
title(main = "Regression(Modified hyper parameters) of degree 7 ")
for (i in 1:nsim) {
  points(X, temp_res1[,i], type = "l",col = "brown")
}

Womenwork = read.table("WomenWork.dat", header = TRUE)
X1 = as.matrix(Womenwork[,2:9])
Y1 = as.matrix(Womenwork[,1])
tau = 10

#beta_v =
nPara = dim(X1)[2]
```

```
mu = as.vector(rep(0,nPara)) # Prior mean vector
Sigma_log = tau^2*diag(nPara);

logistic_post = function(beta_v, x, y, mu, Sigma_log){

    nPara = length(beta_v)
    bx = x%*%beta_v

    loglik = sum(bx*y -  log(1 + exp(bx)))

    logPrior = dmvnorm(beta_v, matrix(0,nPara,1), Sigma_log, log=TRUE)

    return(loglik + logPrior)
}

initval = as.vector(rep(0,dim(Xl)[2]))

Optim_res = optim(initval,
                    logistic_post,
                    gr=NULL,Xl,Yl,mu,Sigma_log,
                    method=c("BFGS"),
                    control=list(fnscale=-1),
                    hessian=TRUE)
beta_opt = Optim_res$par
kable(t(beta_opt))

hes_sigma = -solve(Optim_res$hessian)
kable(hes_sigma)

n1 = 10000
#beta_post = matrix(0, nrow = n1, ncol = nPara)

nchild = c()
for (i in 1:n1) {
  #beta_post[i,] = rmvnorm(1, beta_opt, hes_sigma)
  nchild[i] = rnorm(1, beta_opt[7], hes_sigma[7,7])
}

#length(nchild)
#dim(beta_post)

# nchild_med = c()
# for (i in 1:length(nchild)) {
#     nchild_med[i] = median(nchild[i,])
# }

int_nchild = quantile(nchild, probs = c(0.025, 0.975))
#int_nchild = t(apply(t(nchild), MARGIN = 2, quantile, probs = c(0.025, 0.975)))
kable(t(int_nchild))
#hist(nchild_med)

# points(x = Xl[,7], y = int_nchild[,1], type = "l", col = "red")
# points(x = Xl[,7], y = int_nchild[,2], type = "l", col = "red")
```

```
set.seed(12345)
ylog = function(xvec, ns){
  yres = c()
  yres_bin = c()
  for (i in 1:ns) {
      beta_post = rmvnorm(1, beta_opt, hes_sigma)
      yres[i] = exp(t(xvec) %*% t(beta_post))/(1 + (exp(t(xvec) %*% t(beta_post))))
      yres_bin[i] = rbinom(1, 1, yres[i])
  }
  return(yres_bin)
}

xvec1 =  as.matrix(c(1,10,8,10,1,40,1,1))

ylog1 = ylog(xvec1, 1000)


plot(density(ylog1))




length(which(ylog1==1))

ybin = function(xvec, ns){
  yres = c()
  ybin = c()
  for (i in 1:ns) {
      beta_post = rmvnorm(1, beta_opt, hes_sigma)
      yres[i] = exp(t(xvec) %*% t(beta_post))/(1 + (exp(t(xvec) %*% t(beta_post))))
      ybin[i] = rbinom(1, 10, yres[i])
  }
  #yres = ifelse(yres<0.5,0,1)
  return(ybin)
}

ybin1 = density(ybin(xvec1, 10000))
plot(ybin1, xlab = "Number of Women")
#title(main = "Predictive Distribution of number of Women that Work")
#hist(ybin(xvec1, 10000))
```