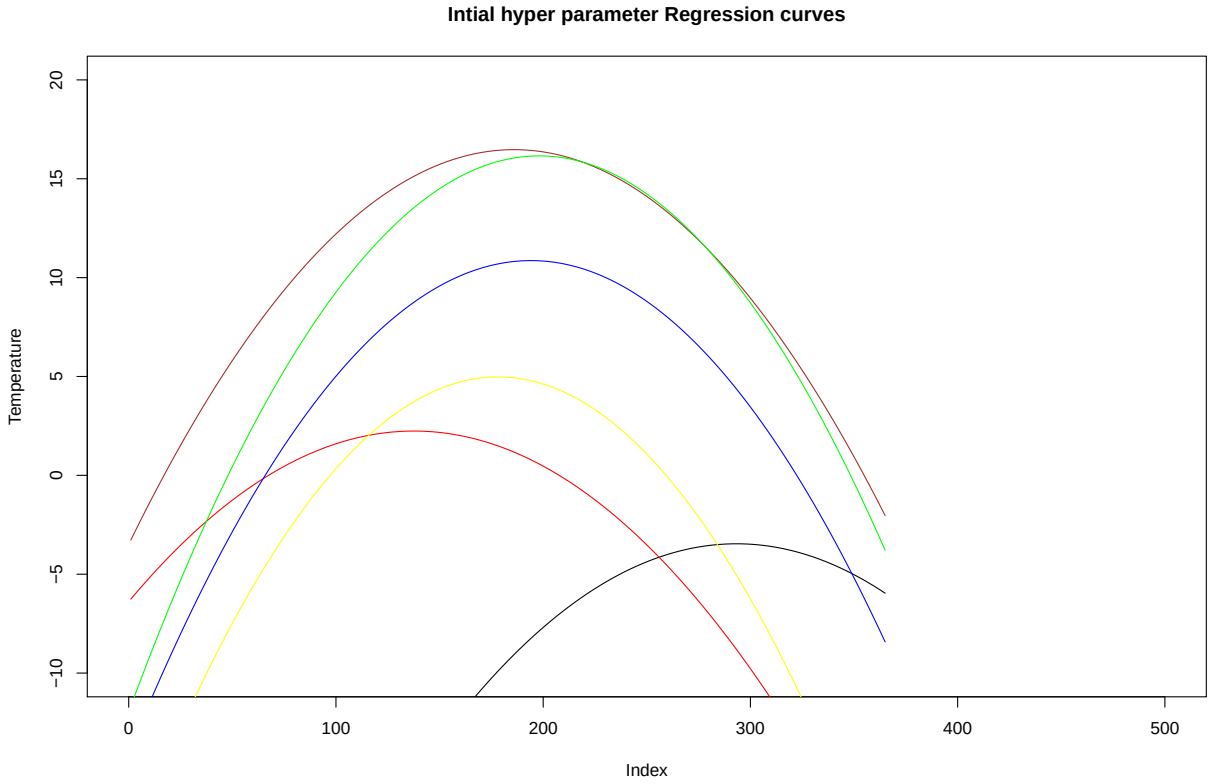
BayesianLab2

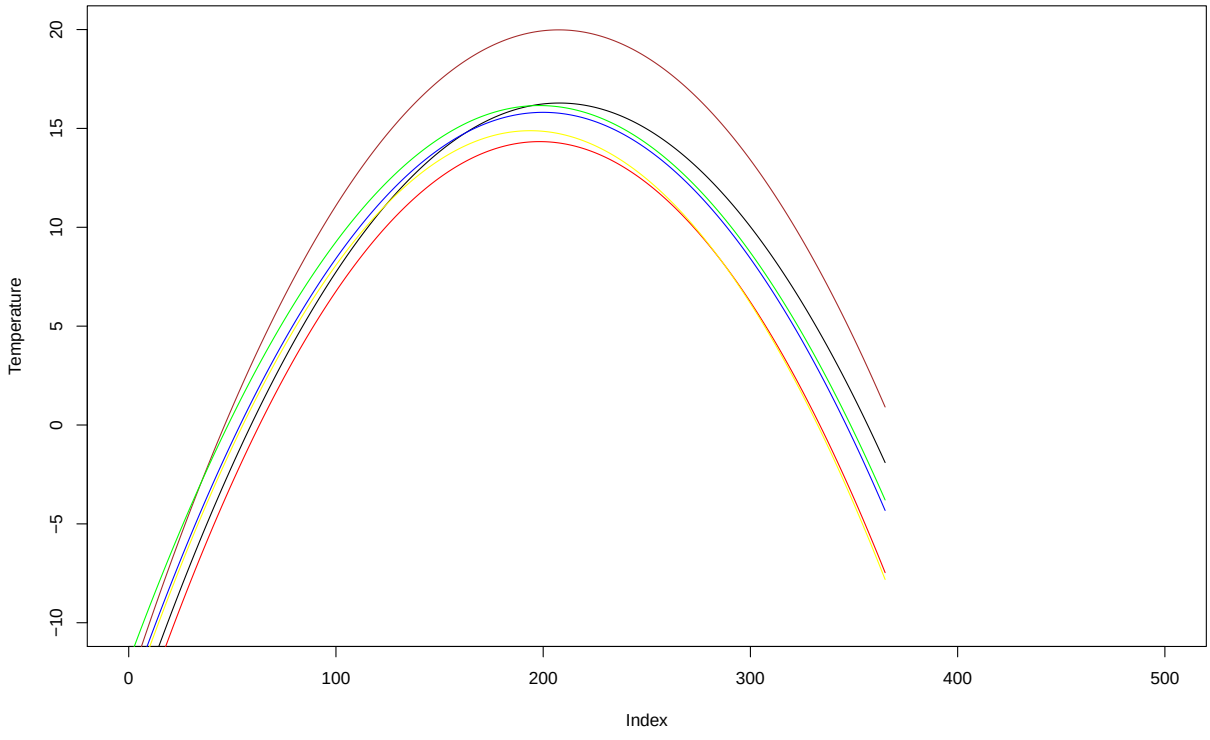
Harshavardhan Subramanian Vinod Kumar Dasari

26/04/2020

1. Linear and polynomial regression

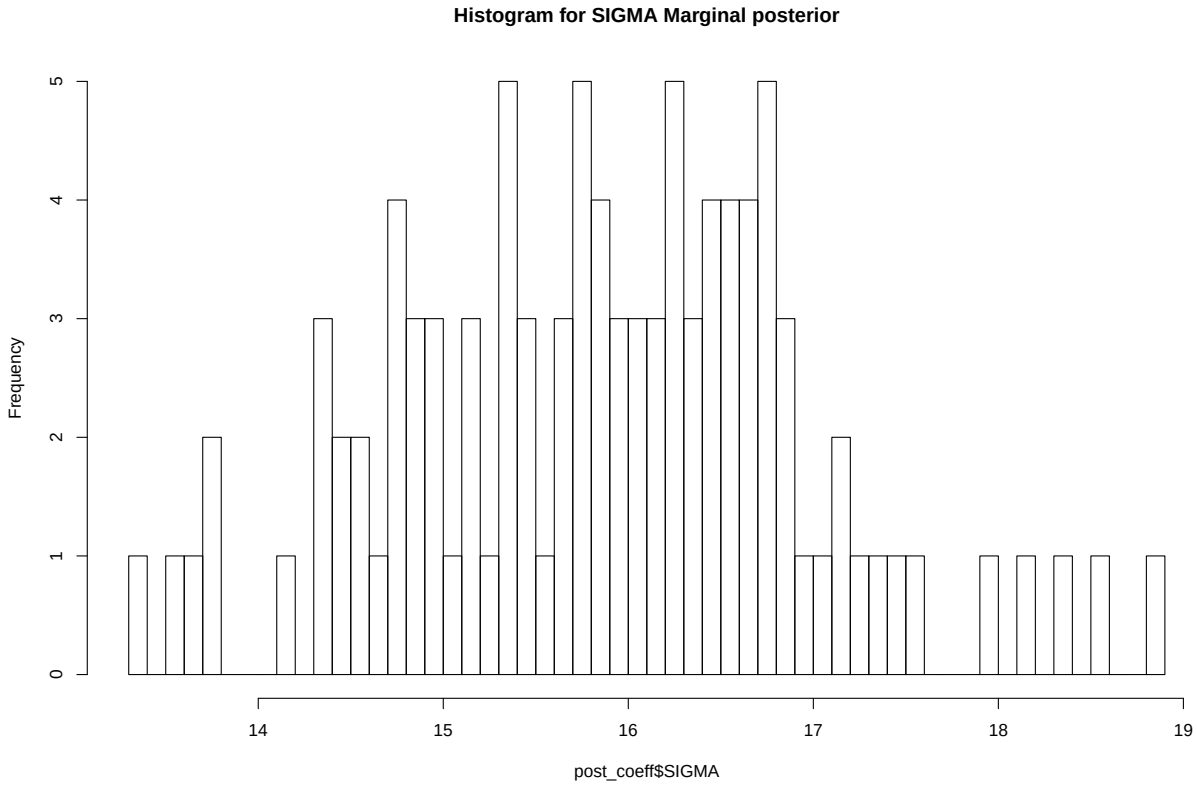
a

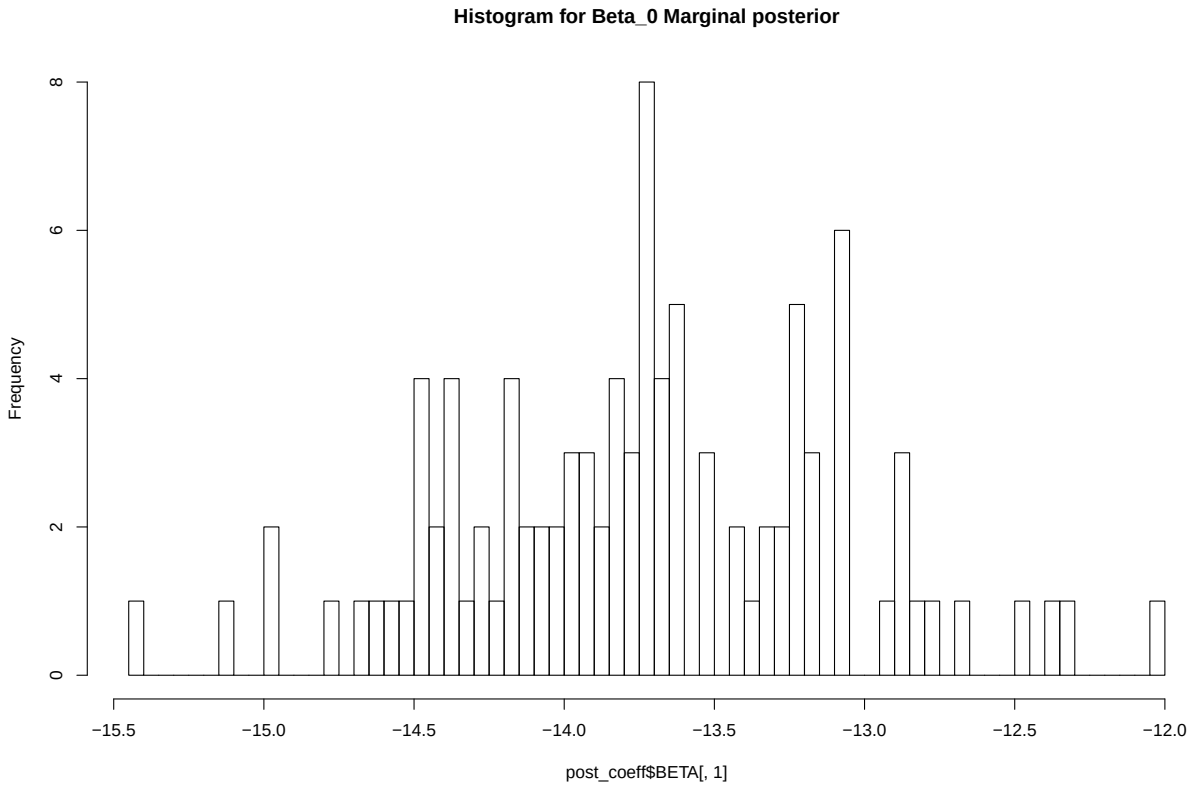


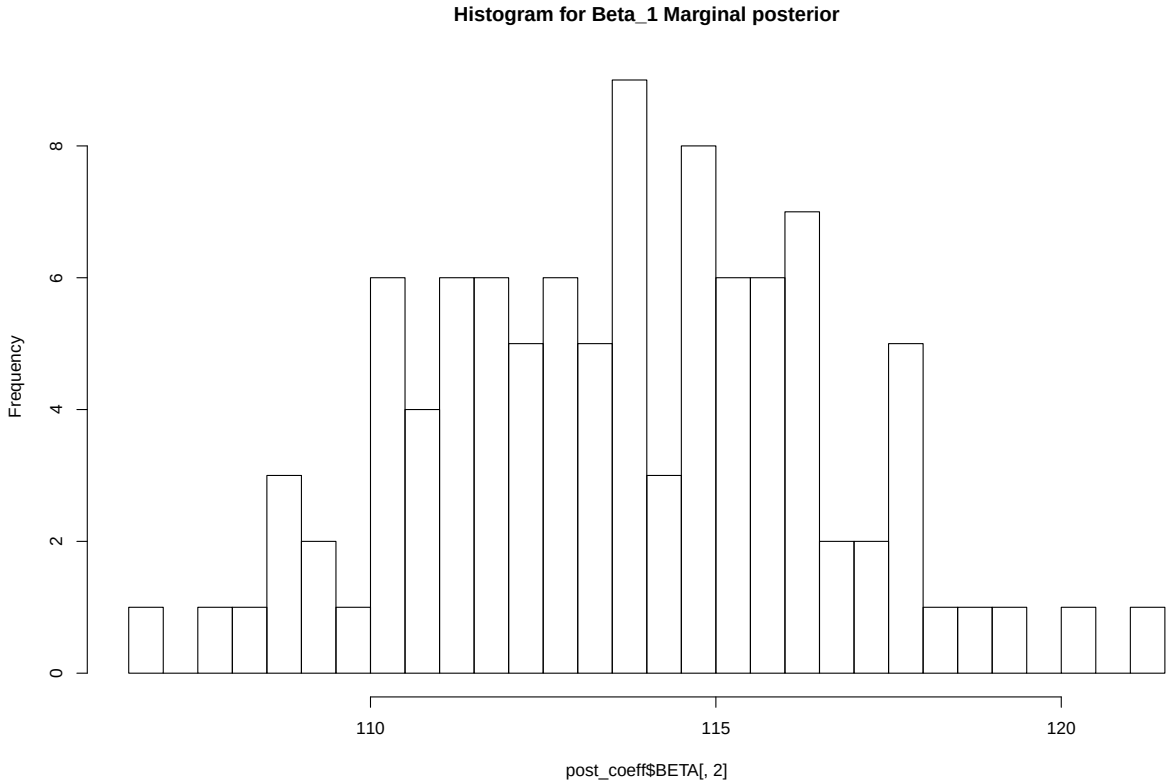


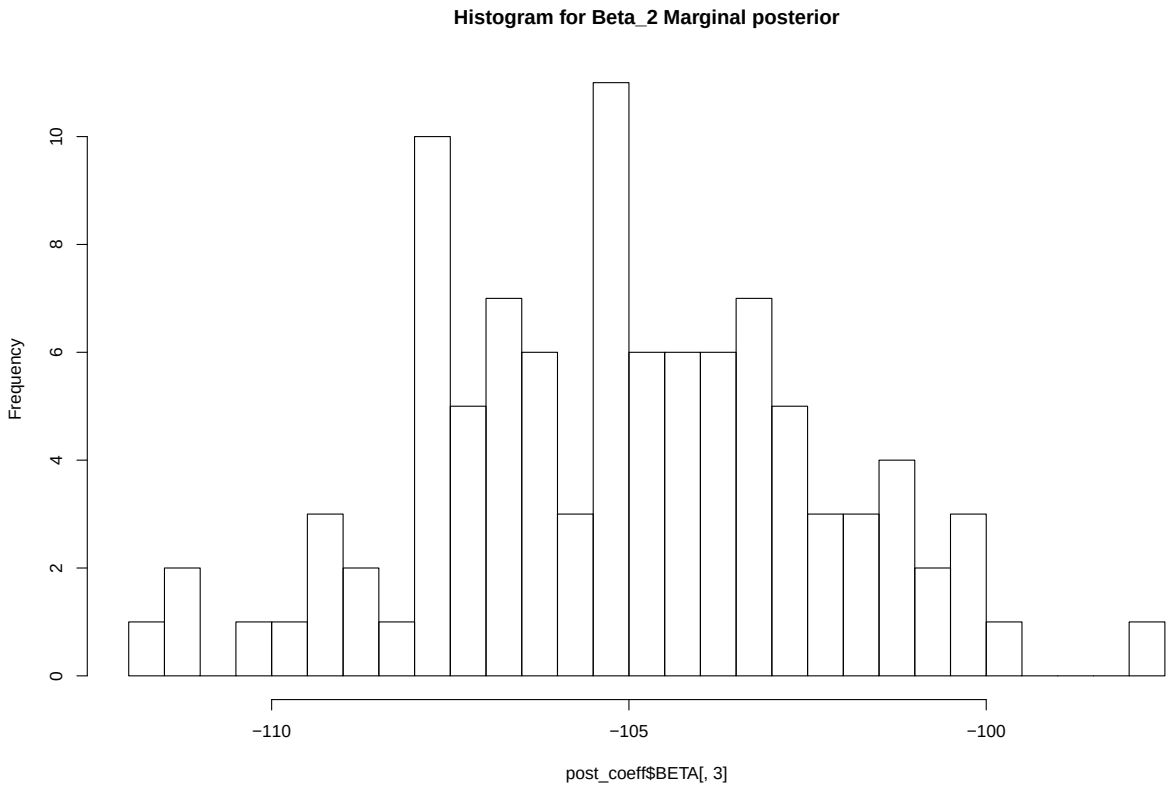
The collection of curves does look reasonable. As we can see in the plot, almost all the curves depicts that, during the winter the temperature is around -10 to 0 and during summer it goes from 0 to 15 which is reasonable. The “Green” curve in the plot is the calculation of temperature by the calculating the coefficients from inbuilt “lm” polynomial function. We can see that there are few curves which overlaps with the “Green” curve hence depicting coefficients make sense which inturn dependent on hyperparameters defined accordingly.

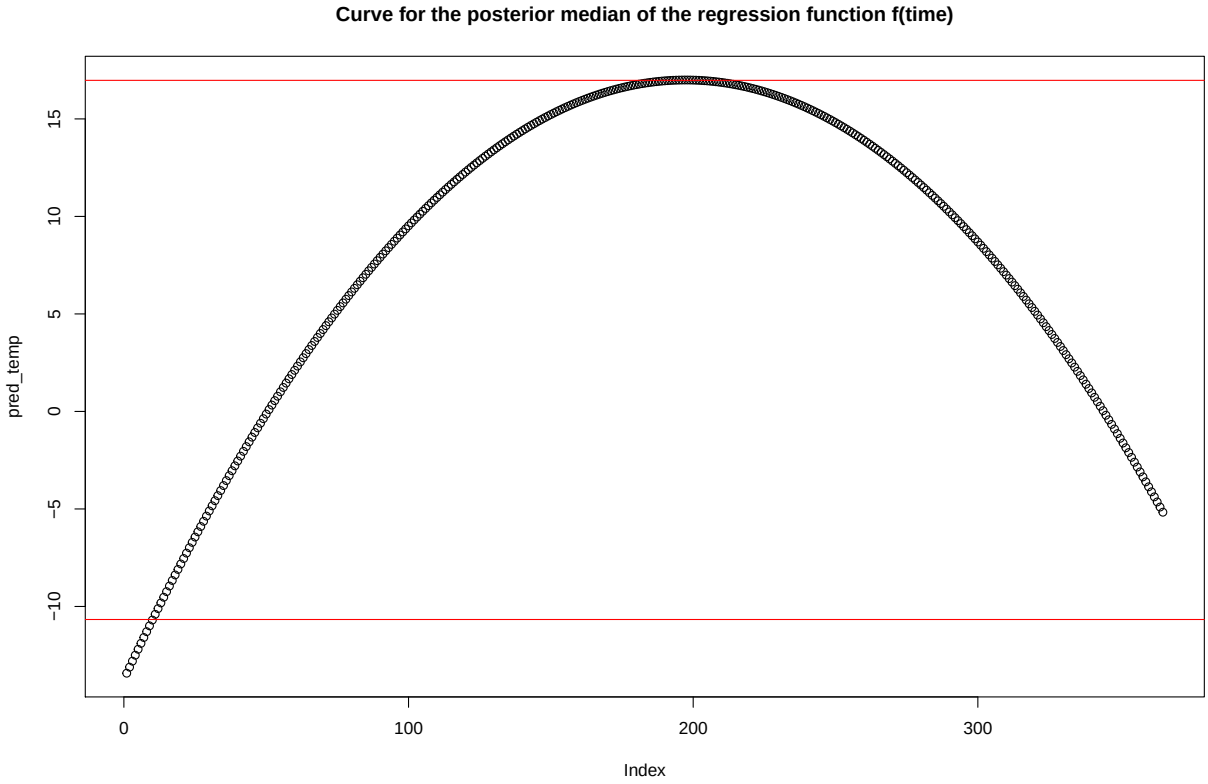
b


Histogram for SIGMA Marginal posterior
Frequency
0
1
2
3
4
5
14
15
16
17
18
19
post_coeff$SIGMA


Histogram for Beta_0 Marginal posterior
Frequency
0
2
4
6
8
−15.5
−15.0
−14.5
−14.0
−13.5
−13.0
−12.5
−12.0
post_coeff$BETA[, 1]

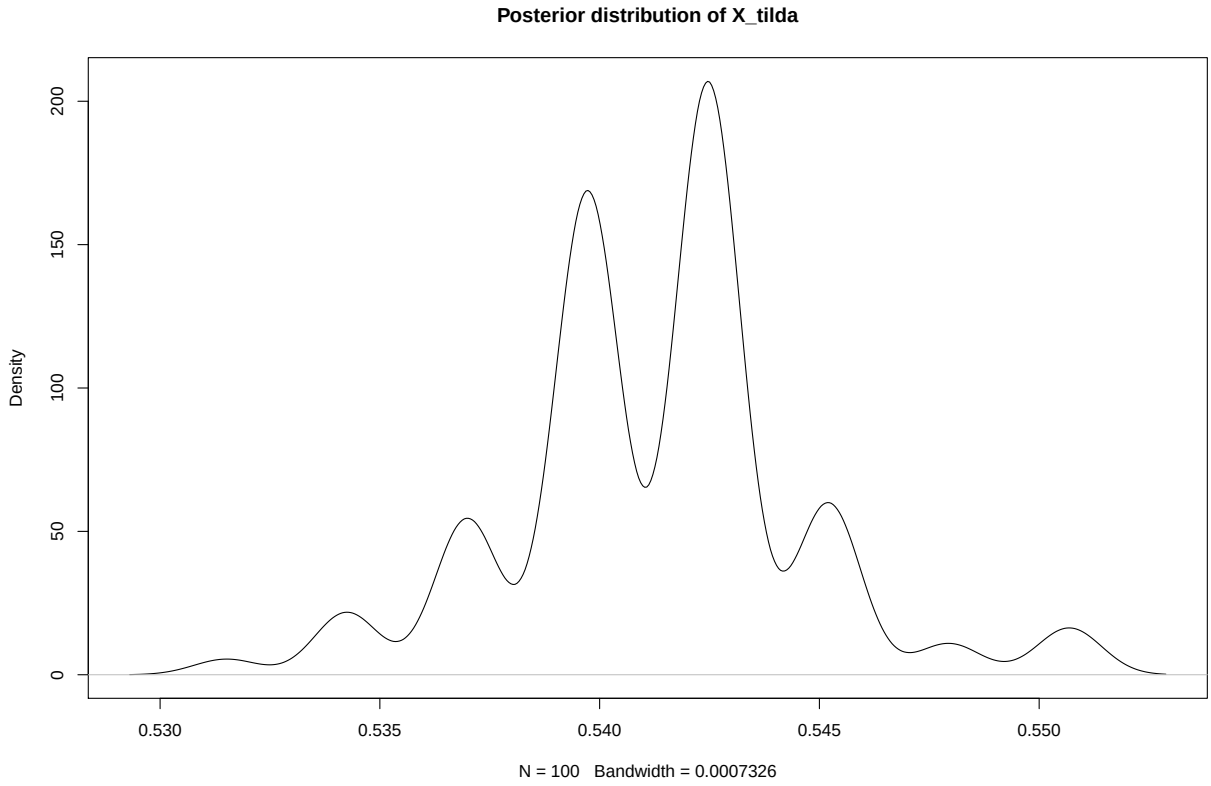

Histogram for Beta_1 Marginal posterior
Frequency
0
2
4
6
8
110
115
120
post_coeff$BETA[, 2]


Histogram for Beta_2 Marginal posterior
Frequency
0
2
4
6
8
10
−110
−105
−100
post_coeff$BETA[, 3]



The 95% Posterior Credible interval covers almost all data points except the data points which are > 16.91(10 data points) and $<$ -10.721(10 data points) out of 365 data points. That is, it does not cover approximately 5.4% of data points which is quite reasonable for the given Credible Interval.

c



d

In case of estimating a polynomial model of order 7, it is more prone to get overfitted hence we add some penalty on the coefficients for shrinking which avoids over penalty. This requires some modification with respect to Prior hyper parameters.

$$\beta_i|\sigma^2 \sim N(0, \frac{\sigma^2}{\lambda})$$

We initialize

$$\mu_0 = 0$$

to 0 and

$$\Omega_0 = \lambda I$$

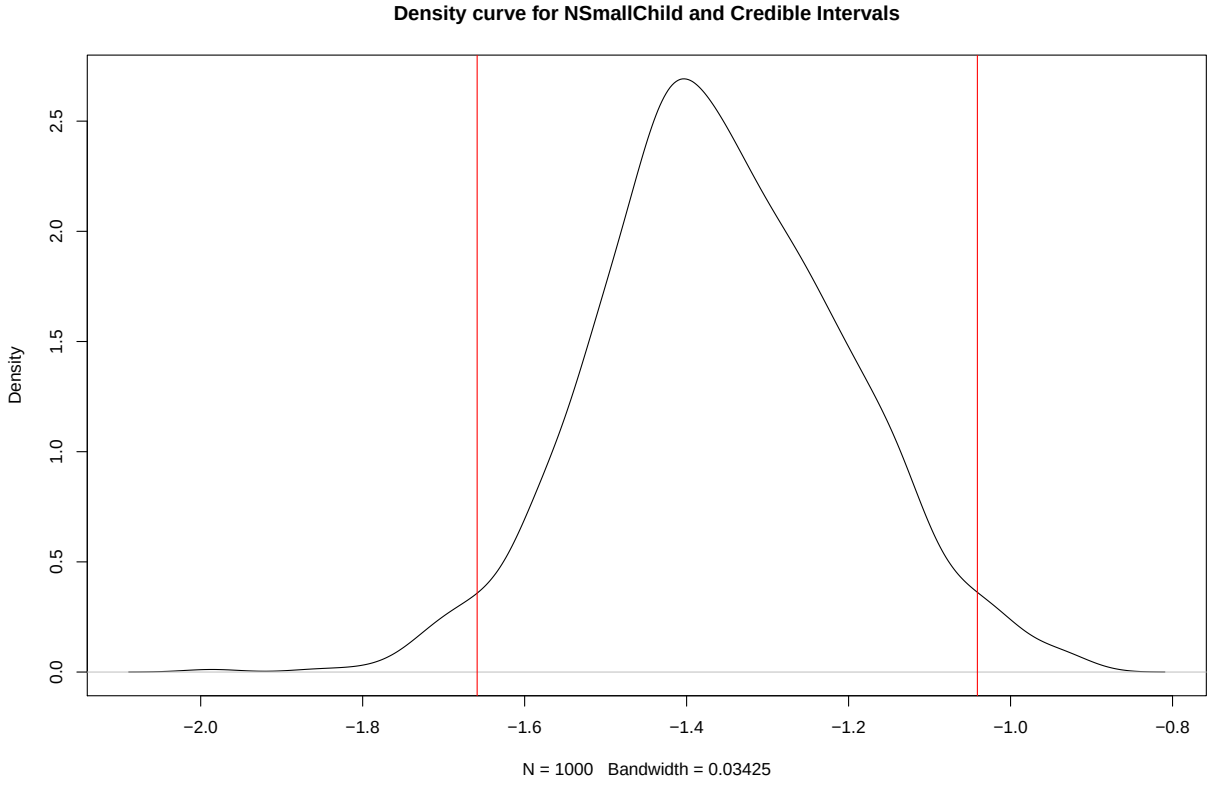
Larger lambda gives smoother fit by shrinking the coefficients to an higher extent.

2. Posterior approximation for classification with logistic regression

a

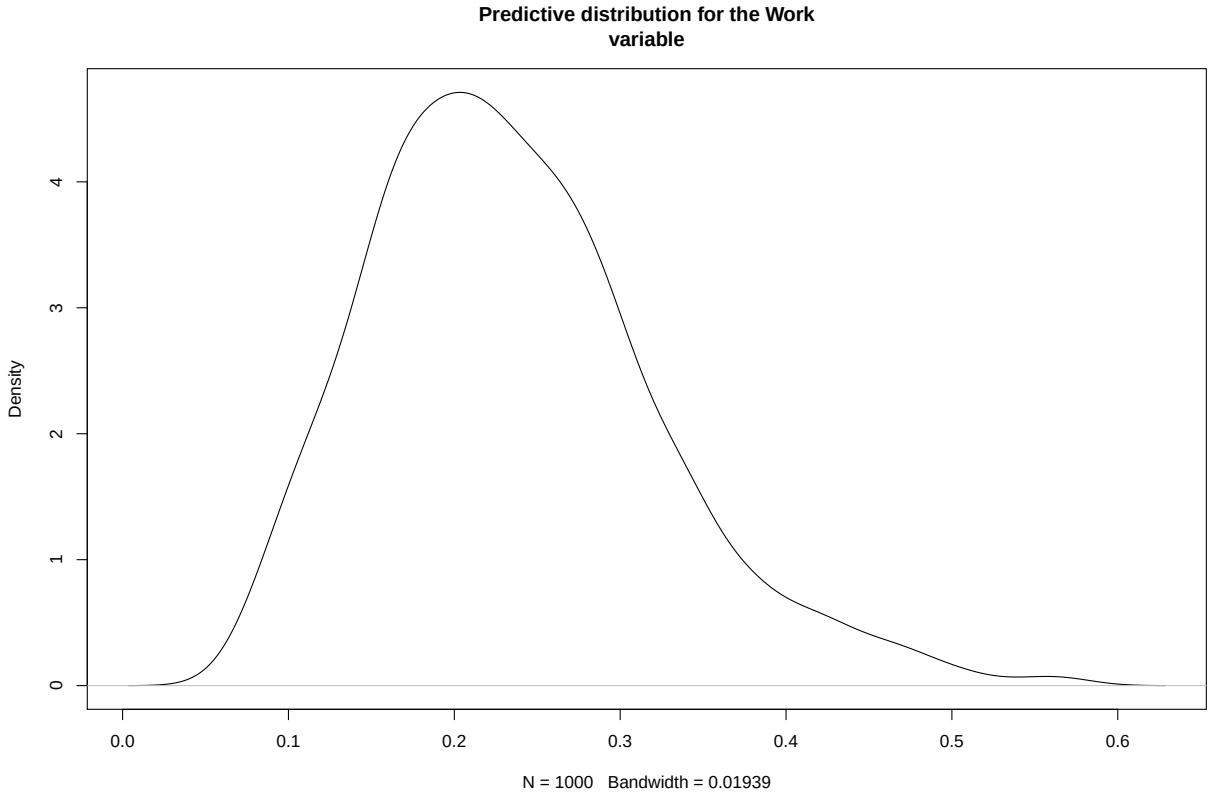
```
## The loglikelihood value for logarithemic posterior is: -21322.46

## The posterior mode is found by simulation is :
##  0.6267288 -0.01979113 0.180219 0.1675667 -0.1445967 -0.08206561 -1.359133 -0.02468351

## The observed Inverse Hessian matrix evaluated at the posterior mode is :
```

```
##              [,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
```



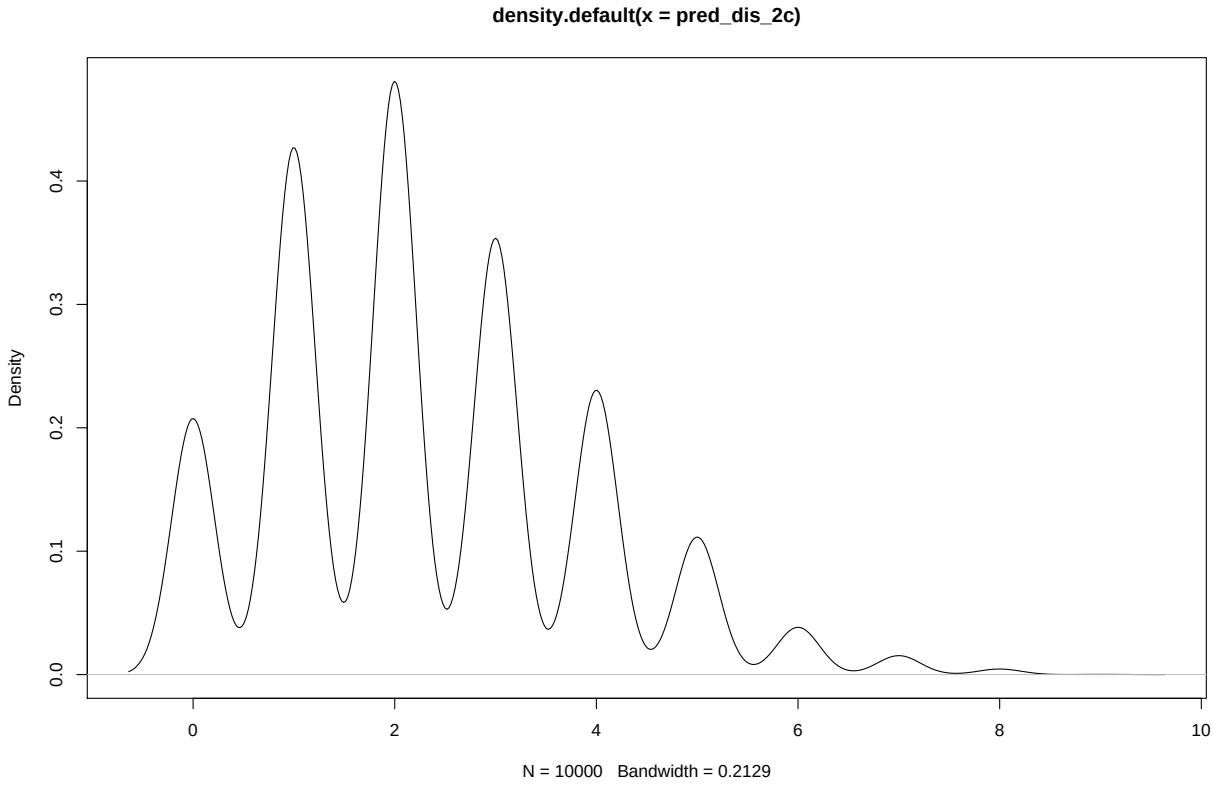
The Density curve for NSmallChild with credible intervals clearly depicts that most of the data points are less than 0 since the coefficient for this variable is negative. This means that this variable is contribution is penalizing the overall prediction by producing negative product of coefficient and the variable. Hence this feature is an important determinant of the probability that women works.

b



From the plot, the curve is right skewed where most of the data points lie. Most of the data points are in the range between 0.2 and 0.4 which is more closer to 0 than 1. Hence, from the plotted density curve we can tell that the 40 year old Woman with two children (3 and 9 years old), 8 years of education, 10 years of experience is not Working currently.

c



From the density curve plotted with respect to number of women working out of 10 for the given variables, it can be said that after 10000 simulations,2 out of 10 is working since it has the highest probability.

Appendix

```
knitr::opts_chunk$set(echo = TRUE,warning = FALSE, message = FALSE,fig.width=12, fig.height=8)
temp_data <- read.csv(file.choose())
X <- temp_data$time
Y <- temp_data$temp
mu_o <- c(-10,100,-100)
omega_o <- matrix(0,3,3)
diag(omega_o) <- c(0.01,0.01,0.01)
v_o <- 4
sigmao_2 <- 1
library(geoR)
simulate_beta_sigma_2 <- function(mu_o,omega_o,v_o,sigmao_2,ndraw,X)
{
  sigma_2 <- vector()
  beta <- matrix(NA,nrow = ndraw,ncol = 3)
  temp <- matrix(NA,nrow = length(X),ncol = ndraw)
  for(i in 1:ndraw)
  {
    sigma_2[i] <-rinvchisq(1,df = v_o,scale = sigmao_2)
    #print(sigma_2[i] * solve(omega_o))
    beta[i,] <- rmvnorm(1,mean = mu_o,sigma = sigma_2[i] * solve(omega_o))
```

```
    temp[,i] <- beta[i,1] + (beta[i,2] * X) + (beta[i,3] * X^2) + rnorm(1,0,sigma_2[i])
    #print(temp)
    #print(beta)

  }
  #print(beta[2,])
  return(list("TEMP" = temp,"BETA" = beta))
}
library(mvtnorm)

mod <- lm(temp ~ poly(time, degree = 2), data = temp_data)

prior_draws <- simulate_beta_sigma_2(mu_o,omega_o,v_o,sigmao_2,ndraw = 10,X)
plot(prior_draws$TEMP[,1],type = "l",ylim = c(-10,20),xlim = c(0,500), ylab = "Temperature", main = "In
lines(prior_draws$TEMP[,2],col = "red")
lines(prior_draws$TEMP[,3],col = "blue")
lines(prior_draws$TEMP[,4],col = "yellow")
lines(prior_draws$TEMP[,5],col = "brown")
lines(mod$fitted.values,col = "green")

mu_o <- c(-15,110,-100)
omega_o <- matrix(0,3,3)
diag(omega_o) <- c(0.05,0.01,0.01)
v_o <- 3
sigmao_2 <- 0.1

prior_draws <- simulate_beta_sigma_2(mu_o,omega_o,v_o,sigmao_2,ndraw = 10,X)
plot(prior_draws$TEMP[,1],type = "l",ylim = c(-10,20),xlim = c(0,500), ylab = "Temperature",main = "Mod
lines(prior_draws$TEMP[,2],col = "red")
lines(prior_draws$TEMP[,3],col = "blue")
lines(prior_draws$TEMP[,4],col = "yellow")
lines(prior_draws$TEMP[,5],col = "brown")
lines(mod$fitted.values,col = "green")
X <- cbind(1,X,X^2)

beta_cap <- solve((t(X) %*% X)) %*% t(X) %*% Y
mu_n <- solve((t(X) %*% X) + omega_o) %*% ((t(X) %*% X) %*% beta_cap + (sigmao_2 * mu_o))
omega_n <-  t(X) %*% X + omega_o
v_n <- dim(X)[1] + v_o
sigman_2 <- (v_o * sigmao_2 + (t(Y) %*% Y + t(mu_o) %*% omega_o %*% t(t(mu_o)) - t(mu_n) %*% omega_n %*

post_simulation <- function(mu_n,v_n,omega_n,sigman_2,ndraw)
{
  sigma_2 <- vector()
  beta <- matrix(NA,nrow = ndraw,ncol = 3)
  #temp <- matrix(NA,nrow = length(X),ncol = ndraw)
  for(i in 1:ndraw)
  {
    sigma_2[i] <-as.vector(rinvchisq(1,df = v_n,scale = sigman_2))
    #print(sigma_2[i] * solve(omega_o))
    beta[i,] <- rmvnorm(1,mean = as.vector(mu_n),sigma = sigma_2[i] * solve(omega_n))
    #temp[,i] <- beta[i,1] + (beta[i,2] * X) + (beta[i,3] * X^2) + rnorm(1,0,sigma_2[i])
    #print(temp)
```

```
    #print(beta)

  }
  #print(beta[2,])
  return(list("SIGMA" = sigma_2,"BETA" = beta))
}

post_coeff <- post_simulation(mu_n,v_n,omega_n,sigman_2,ndraw = 100)
hist(post_coeff$SIGMA,50,main = "Histogram for SIGMA Marginal posterior")
hist(post_coeff$BETA[,1],50,main = "Histogram for Beta_0 Marginal posterior")
hist(post_coeff$BETA[,2],50,main = "Histogram for Beta_1 Marginal posterior")
hist(post_coeff$BETA[,3],50,main = "Histogram for Beta_2 Marginal posterior")


beta_0 <- median(post_coeff$BETA[,1])
beta_1 <- median(post_coeff$BETA[,2])
beta_2 <- median(post_coeff$BETA[,3])

pred_temp <- beta_0 + beta_1 * X[,2] + beta_2 * X[,3]
CI <- quantile(pred_temp, probs = c(0.025,0.975))
plot(pred_temp, main = "Curve for the posterior median of the regression function f(time)")
abline(h = CI[1],col = "red")
abline(h = CI[2],col = "red")
X_tild_simulation <- function(mu_n,v_n,omega_n,sigman_2,ndraw,X)
{
  X_tilda <- vector()
  for(i in 1:ndraw)
  {
    sigma_2 <-as.vector(rinvchisq(1,df = v_n,scale = sigman_2))
    #print(sigma_2[i] * solve(omega_o))
    beta <- rmvnorm(1,mean = as.vector(mu_n),sigma = sigma_2 * solve(omega_n))
    temp <- beta[1] + (beta[2] * X[,2]) + (beta[3] * X[,3])
    X_tilda[i] <- X[which(temp == max(temp)),2]
    #print(temp)
    #print(beta)

  }
  return(X_tilda)
}
X_tilda <- X_tild_simulation(mu_n,v_n,omega_n,sigman_2,ndraw = 100,X)
plot(density(X_tilda), main = "Posterior distribution of X_tilda")
women_data <- readxl::read_excel(file.choose())
X <- as.matrix(women_data[,-1])
Y <- as.vector(women_data[,1])
#X <- cbind(1,X)
mu <- as.vector(rep(0,ncol(X)))
tau2 <- 100
sigma <- tau2*diag(ncol(X))
#diag(sigma) <- tau2
# prior
beta_vect <- rnorm(ncol(X),mu,diag(sigma))

log_post <- function(beta_vect,X,Y,mu,sigma)
```

```
{
loglikelihood <- sum((X %*% beta_vect * Y) - log(1 + exp(X %*% beta_vect)))
if (abs(loglikelihood) == Inf) loglikelihood = -20000;
logprior <- dmvnorm(beta_vect,mean = rep(0,length(beta_vect)),sigma = sigma, log=TRUE)
logpost <- loglikelihood + logprior
return(logpost)
}

loglikelihood_val <- log_post(beta_vect,X,Y,mu,sigma)

cat("The loglikelihood value for logarithemic posterior is:",loglikelihood_val)

pars <- as.vector(rep(0,ncol(X)))
post_mod <- optim(pars,log_post,gr = NULL,X,Y,mu,sigma,method = "BFGS",control=list(fnscale=-1),hessian=
beta_tilda <- post_mod$par
cat("The posterior mode is found by simulation is :\n",beta_tilda)
InvHess <- -solve(post_mod$hessian)
cat("The observed Inverse Hessian matrix evaluated at the posterior mode is :\n")
InvHess
post_sd <- sqrt(diag(InvHess))
y_pred_Nsmallchild <- rnorm(1000,beta_tilda[7],InvHess[7,7])
CI_wo  <- quantile(y_pred_Nsmallchild,probs = c(0.025,0.975))
#hist(y_pred_Nsmallchild,50)
plot(density(y_pred_Nsmallchild),main = "Density curve for NSmallChild and Credible Intervals")
#plot(density(y_pred_Nsmallchild))
abline(v=CI_wo[1],col = "red")
abline(v=CI_wo[2],col = "red")

#
# post_dis_beta <- rmvnorm(1,beta_tilda,InvHess)
# plot(post_dis_beta[,1])
sim_pred_dis <- function(beta_tilda,InvHess,ndraw)
{
  beta_post <- matrix(0,nrow = length(beta_tilda), ncol = ndraw )
  pred_dis <- vector()
  for(i in 1:ndraw)
  {
    beta_post[,i] <- rmvnorm(1,beta_tilda,InvHess)
    X <- cbind(1,10,8,10,1,40,1,1)
    pred_dis[i] <- exp(X %*% beta_post[,i]) / (1 + exp(X %*% beta_post[,i]))
  }
  return(pred_dis)
}
pred_dis <- sim_pred_dis(beta_tilda,InvHess,ndraw = 1000)
plot(density(pred_dis), main = "Predictive distribution for the Work
variable")
sim_pred_dis_2c <- function(beta_tilda,InvHess,ndraw)
{
  beta_post <- matrix(0,nrow = length(beta_tilda), ncol = ndraw )
  working_women <- vector()
  #i <- 1
  op <- vector()
  for(i in 1:ndraw)
```

```
  {
    beta_post[,i] <- rmvnorm(1,beta_tilda,InvHess)
    X <- cbind(1,10,8,10,1,40,1,1)
    #X <- X[rep(1:nrow(X),1,each=10),]

    pred_dis <- exp(X %*% beta_post[,i]) / (1 + exp(X %*% beta_post[,i]))
    op[i] <- rbinom(1,10,pred_dis)
    # for(j in 1:10)
    # {
    # pred_dis <- exp(X %*% beta_post[,i]) / (1 + exp(X %*% beta_post[,i]))
    # #op[j] <- ifelse(pred_dis > 0.5,1,0)
    # op[j] <- ifelse(pred_dis > 0.5,1,0)
    #
    # }
    #working_women[i] <- length(which(op == 1))
  }
  return(op)
}
pred_dis_2c <- sim_pred_dis_2c(beta_tilda,InvHess,ndraw = 10000)
plot(density(pred_dis_2c))
```