Lab2

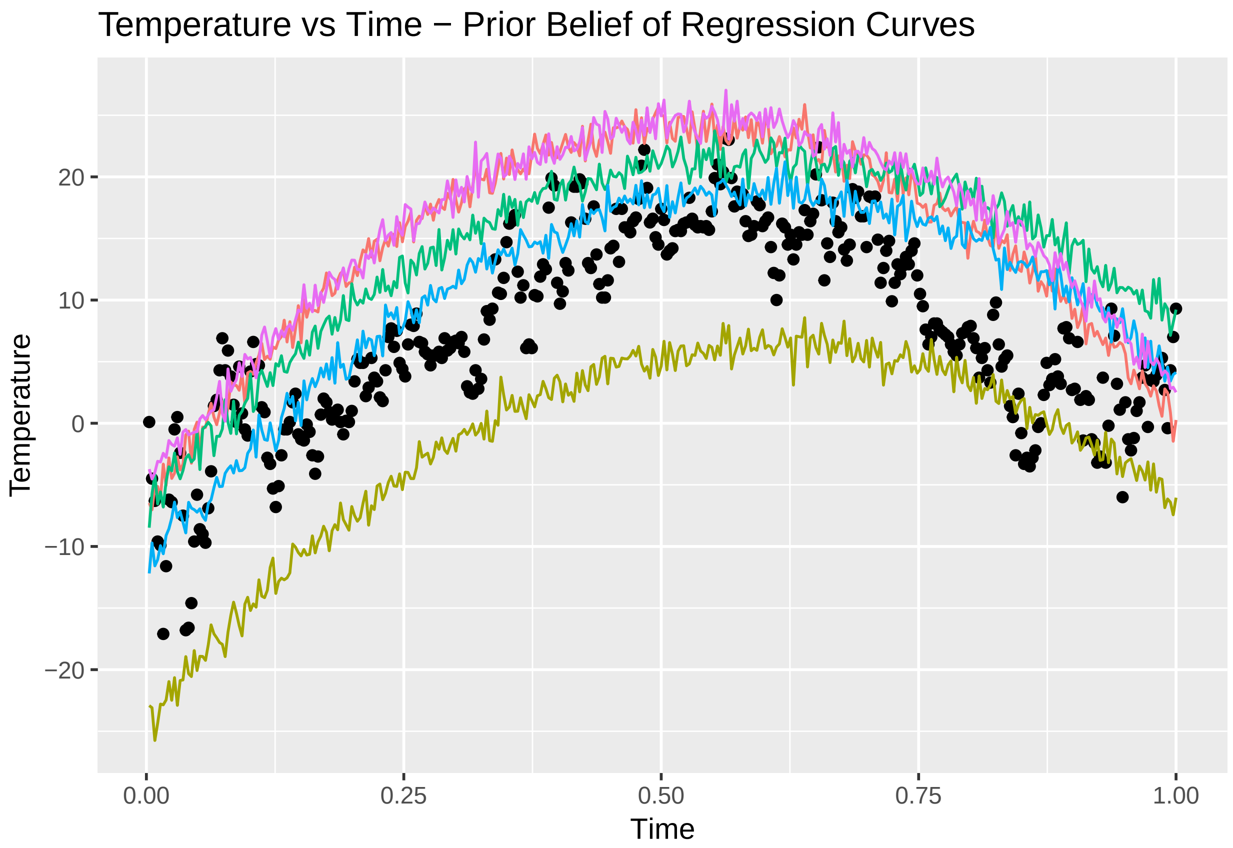
Naveen Gabriel(navga709) Sridhar Adhikarla(sriad858)

5 May 2019

Contents

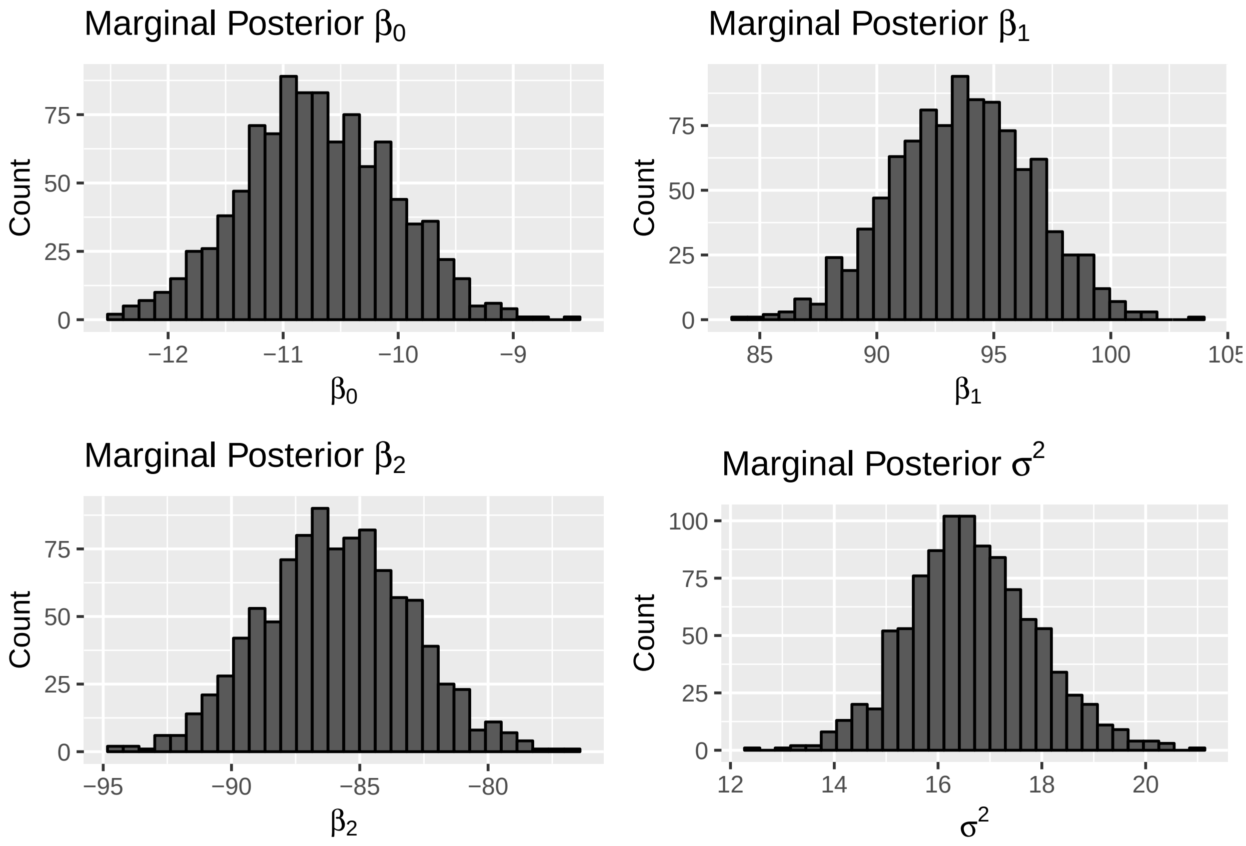
1 Linear and polynomial regression

1 Determining the prior distribution of the model parameters

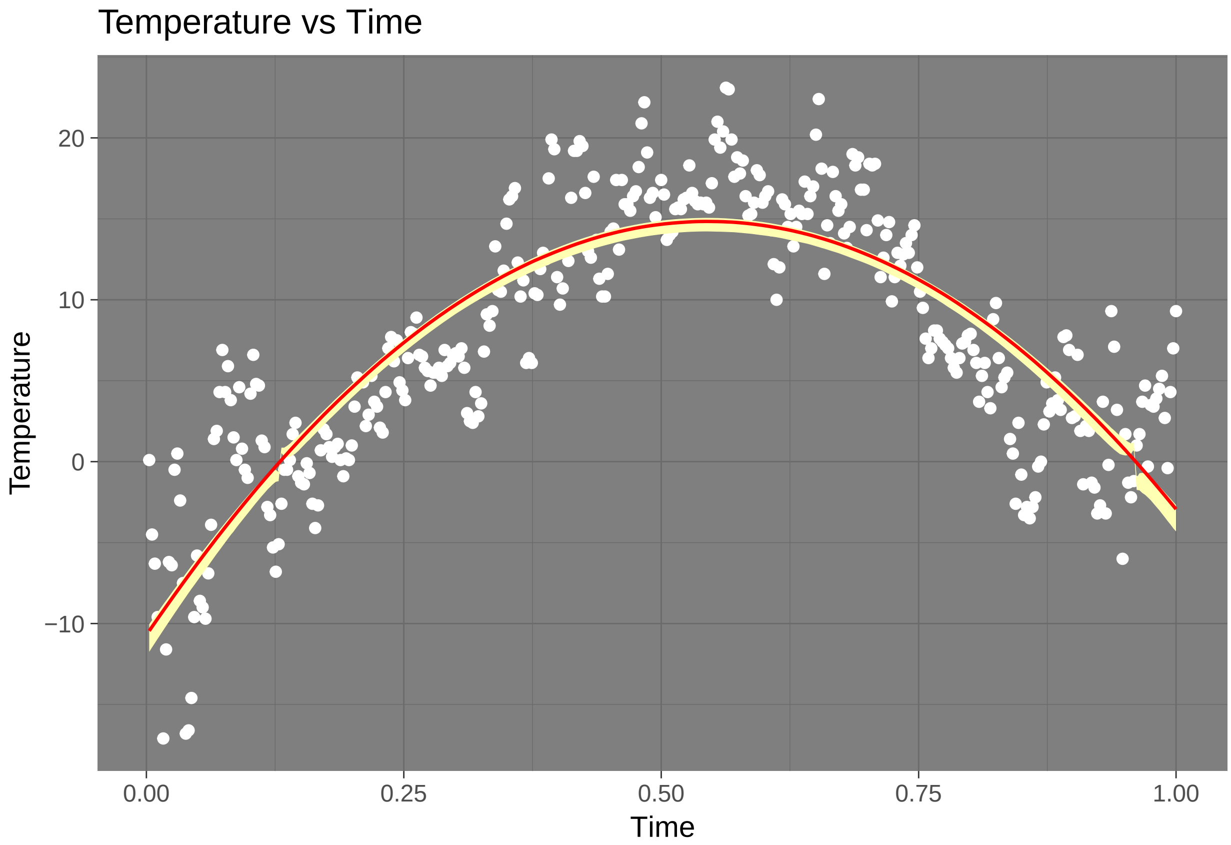


Our prior beliefs on seeing the data does coincides with the group of regression curves which are generated using conjugate prior.

2 Simulate from the joint posterior distribution of β_0 ,β_1 ,β_2 and σ^2



The above plot shows the histogram of marginal posteriors for each parameter, i.e β_0 ,β_1 ,β_2 and σ^2. Each parameter distribution is normal.

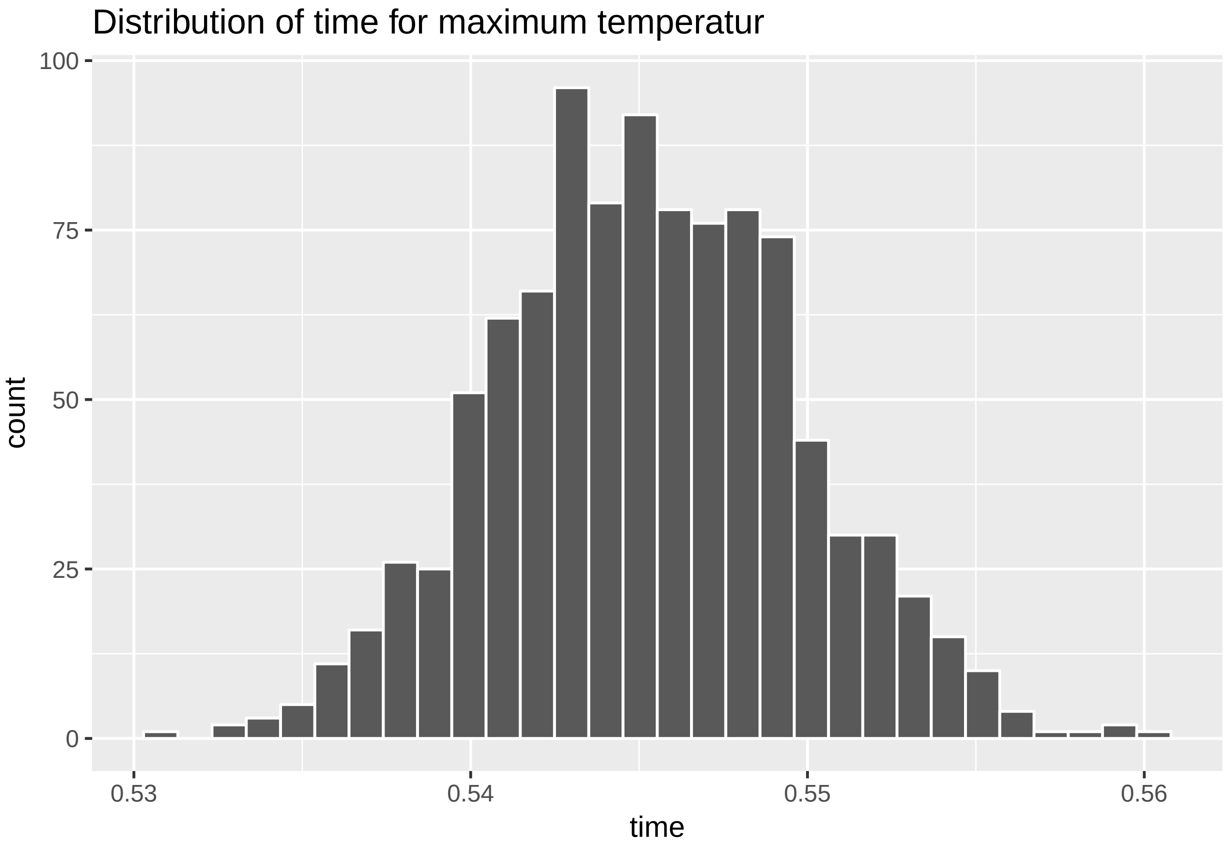


The yellow shows the band with 95% credible interval for f(time). The red curve shows the posterior median of regression curve. It does not include all the point and it should not. The idea is here to find our regression curve which is more tighter and explain the variation of temperature with time.

3. Locate the time with the highest expected temperature

Differentiaing the quadratic equation, we get the maximum value of time. We know it is the maximum value because the double differential is negative. We then get the distribution of time where the temperature is maximum from the marginal posterior value of β_0 and β_2

$$\begin{aligned} f(time) &= \beta_0 + \beta_1.time + \beta_2.time^2 \\ f'(time) &= 0 \\ \beta_1 + 2.\beta_2.time &= 0 \\ time &= -\frac{\beta 1}{2\beta_2} \end{aligned}$$



From the distribution, it seems the time where the temperature is maximum is between 0.54 and 0.55 which rounds off to 197 days approx which comes in June.

4. Mitigate overfitting using prior

We can minimize overfitting by changing th variance of our prior

We know :

$$P(\theta|Y) = P(Y|\theta)P(\theta)$$

Assuming zero-mean normally distributed prior on each β_i value, all with identical variance τ. Likelyhood Prior is given as :

$$P(Y|\theta) = \prod_{i=1}^{n} \frac{1}{\sigma.\sqrt{2.\pi}}.e^{-\frac{y_i-(\beta_0+\beta_1 x_i+\beta_2 x_i..\beta_p x_p)}{2\sigma^2}}$$

$$P(\theta) = \prod_{j=1}^{p} \frac{1}{\tau.\sqrt{2.\pi}}.e^{-\frac{\beta_j^2}{2\tau^2}}$$

The posterior log can then be calculated as :

$$P(\theta|Y) = log(\prod_{i=1}^{n} \frac{1}{\sigma.\sqrt{2.\pi}}.e^{-\frac{y_i-(\beta_0+\beta_1 x_i+\beta_2 x_i..\beta_p x_p)}{2\sigma^2}}) + log(\prod_{j=1}^{p} \frac{1}{\tau.\sqrt{2.\pi}}.e^{-\frac{\beta_j^2}{2\tau^2}})$$

$$P(\theta|Y) = \sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i..\beta_p x_p))^2 + \frac{1}{\tau^2}\sum_{j=1}^{p}\beta_j^2$$

$$P(\theta|Y) = \sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i..\beta_p x_p))^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

Here $1/\tau^2$ is λ which is our regularization term. We can adjust the amount of regularization we want by changing ??. Equivalently, we can adjust how much we want to weight the priors carry on the coefficients (??). If we have a very small variance (large ??) then the coefficients will be very close to 0; if we have a large variance (small ??) then the coefficients will not be affected much (similar to as if we didn't have any regularization). Which means by increasing the value of lambda our Ω_0 will have low variance

Source : http://bjlkeng.github.io/posts/probabilistic-interpretation-of-regularization/

2. Posterior approximation for classification with logistic regression

1 Logistic regression

```
Call:
glm(formula = Work ~ 0 + ., family = "binomial", data = womenwork)

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
```

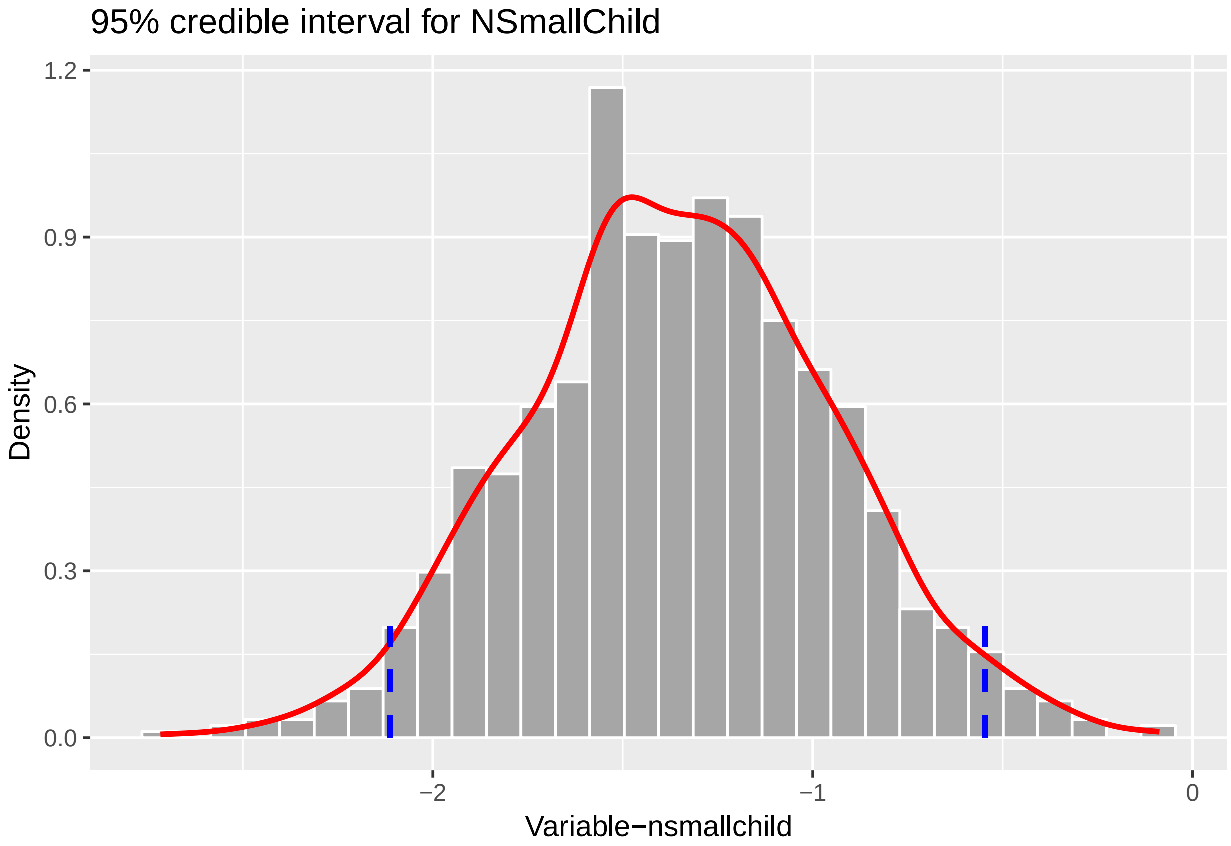
2. Approximate the posterior distribution of the 8-dim parameter and credible interval

Table 1: Coeffecient value of variables

	Coeffecient
Constant	0.6267288
HusbandInc	-0.0197911
EducYears	0.1802190
ExpYears	0.1675667
ExpYears2	-0.1445967
Age	-0.0820656
NSmallChild	-1.3591332
NBigChild	-0.0246835

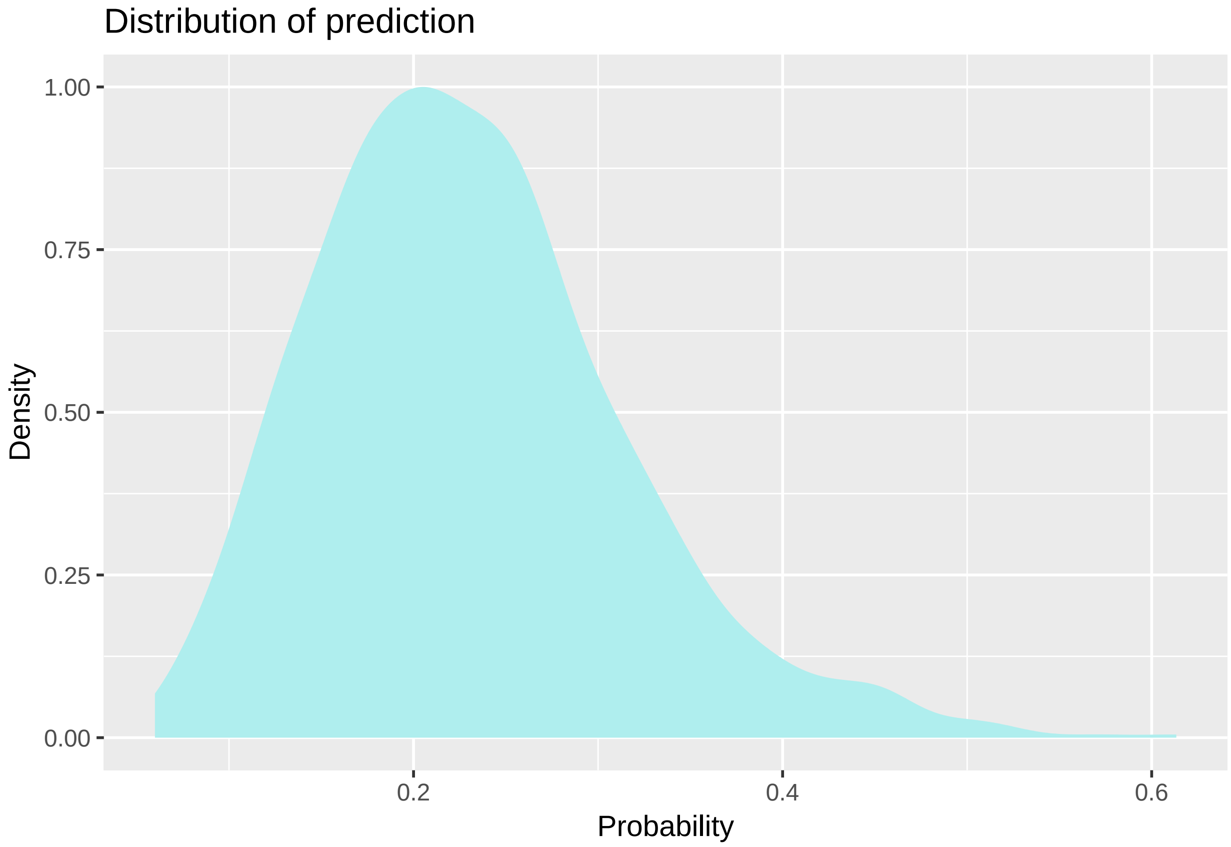
```
Hessian matrix:

             [,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
```



Would you say that this feature is an important determinant of the probability that a women works? By looking at the distribution of nsmallChild parameter, 95% credible interval values lies between -2.1 and -0.5 peaking around -1 which means it negatively effect the outcome. So having a child would mean that the women does not work. For greater value of nsmallchild,the value becomes more negative and vice a versa. With this thinking we believe that this variable does effect the outcome.

3. Function that simulates from the predictive distribution of the response variable in a logistic regression



From the above distribution most the probability value is below 0.5 and peaks at 0.2 which means 0 has high chances to be the value for classification which means that the women does not work for the given set of parameters.

Appendix

```r
knitr::opts_chunk$set(echo = FALSE,
                      eval = TRUE,
                      warning = FALSE,
                      comment = NA,
                      message=FALSE)
library(ggplot2)
library(mvtnorm)
library(gridExtra)

set.seed(123456789)
templink <- read.delim("TempLinkoping.txt")
n <- nrow(templink)

templink$time_2 <- templink$time^2
templink$firstvar <- 1
templink <- templink[,c(4,1,3,2)]

templink <- as.matrix(templink)
templink_n <- templink
#Setting up parameters for our prior
s_sq <- 1
omga0 <- 0.01*diag(3)
mu0 <- c(-10,100,-100)
v0 <- 4

invchisq <- function(v,sq) {
  sg_sq <- (v*sq)/rchisq(1,v)
  return(sg_sq)
}

#Drawing random variable from chi square
##Conjugate prior
draws <- 8
sg_sq <- invchisq(v0,s_sq)
var_cov <- sg_sq*solve(omga0)

for (i in 1:draws) {
theta <- rmvnorm(1,mu0,var_cov)
y_pred <- templink[,-which(colnames(templink) == "temp")]%*%t(theta) + rnorm(n,0,sg_sq)

templink_n <- cbind(templink_n,y_pred)
}

#drawing theta from Normal distribution
templink_n <- as.data.frame(templink_n)

ggplot(templink_n,aes(time,temp)) + geom_point() +
  geom_line(aes(y=V5,col="V5")) +
  geom_line(aes(y=V6,col="V6")) +
  geom_line(aes(y=V7,col="V7")) +
  geom_line(aes(y=V8,col="V8")) +
  geom_line(aes(y=V9,col="V9")) +
  ggtitle("Temperature vs Time - Prior Belief of Regression Curves") + xlab("Time") + ylab("Temperature
  theme(legend.position = "none")



model = lm(temp ~ 1+time+time_2, data = templink_n)
betahat = model$coefficients

mun <- solve(t(templink[,c(1:3)])%*%templink[,c(1:3)] + omga0) %*% (t(templink[,c(1:3)])%*%templink[,c(
omegan <- t(templink[,c(1:3)])%*%templink[,c(1:3)] + omga0
vn <- v0+n
vn_sigsq <- v0*s_sq + t(templink[,4])%*%templink[,4] + t(mu0)%*%omga0%*%mu0 - t(mun)%*%omegan%*%mun
sg_sq_n  <- vn_sigsq/vn

draws_pos <- 1000

marginal_pos <- matrix(ncol=4,nrow = draws_pos)
colnames(marginal_pos)<- c("beta_0","beta_1","beta_2","sig")

for( i in 1:draws_pos) {
  sg_sq <- invchisq(vn,sg_sq_n)
  var_cov <- as.numeric(sg_sq)*solve(omegan)
  theta <- rmvnorm(1,mun,var_cov)
  marginal_pos[i,] <- c(theta,sg_sq)
}

marginal_pos <- as.data.frame(marginal_pos)

p1 <- ggplot(marginal_pos) + geom_histogram(aes(beta_0),color="black") + xlab (expression(beta[0])) + yl
                       ggtitle(expression(paste("Marginal Posterior ",beta[0])))
p2 <- ggplot(marginal_pos) + geom_histogram(aes(beta_1),color="black") + xlab (expression(beta[1])) + yl
                       ggtitle(expression(paste("Marginal Posterior ",beta[1])))
p3 <- ggplot(marginal_pos) + geom_histogram(aes(beta_2),color="black") + xlab (expression(beta[2])) + yl
                       ggtitle(expression(paste("Marginal Posterior ",beta[2])))
p4 <- ggplot(marginal_pos) + geom_histogram(aes(sig),color="black") + xlab (expression(sigma^2)) + ylab
                       ggtitle(expression(paste("Marginal Posterior ",sigma^2)))

grid.arrange(p1,p2,p3,p4, nrow=2)

pose_table <- matrix(ncol=draws_pos,nrow=n)

for(i in 1:draws_pos) {
  beta0 <- marginal_pos[i,]$beta_0
  beta1 <- marginal_pos[i,]$beta_1
  beta2 <- marginal_pos[i,]$beta_2

  pose_table[,i] <-  beta0 + beta1*templink_n$time + beta2*templink_n$time_2
}


low_high <- function(x) {
  x <- x[order(-x)]
  y <- cumsum(x)/sum(x)

  ind_high <- max(which(y<=0.25))
  ind_low <- min(which(y>=0.975))

  high_time <- x[ind_high]
  low_time <- x[ind_low]

  return(c(high_time,low_time))
}


templink_n$pos <- apply(pose_table,1,median)

low_high_val <- apply(pose_table,1,FUN=function(x) low_high(x))

templink_n$low_pos <- low_high_val[2,]
templink_n$high_pos <- low_high_val[1,]

#With posterior median and upper and lower 2.5% confidence
ggplot(templink_n,aes(time,temp)) + geom_point(color="white") +
  geom_ribbon(aes(ymin=low_pos, ymax=high_pos),fill = "#ffffb3") +
  geom_line(aes(y=pos),color="red",size=0.6) + theme_dark() +
  ggtitle("Temperature vs Time") + xlab("Time") + ylab("Temperature")

marginal_pos$time <- apply(marginal_pos,1,FUN=function(x) -x[2]/(2*x[3]))

ggplot(marginal_pos,aes(time)) + geom_histogram(color="white") +
  ggtitle("Distribution of time for maximum temperatur ")
womenwork <- read.table("womenWork.dat",header = TRUE)

model <- glm(Work~0+.,data = womenwork,family = "binomial")

summary(model)
x <- as.matrix(womenwork[,-1])
y <- womenwork[,1]
covnames <- colnames(x)
tau <- 10


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 optimizer away from he

  # 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
```

```r
  return(logLik + logPrior)
}

initVal <- as.vector(rep(0,dim(x)[2]))
# Setting up the prior
mu <- as.vector(rep(0,dim(x)[2])) # Prior mean vector
Sigma <- tau^2*diag(dim(x)[2]);

OptimResults<-optim(initVal,LogPostLogistic,gr=NULL,y,x,mu,
                    Sigma,method=c("BFGS"),control=list(fnscale=-1),
                    hessian=TRUE)

postMode <- OptimResults$par
names(postMode) <- covnames
postCov <- -solve(OptimResults$hessian) # Posterior covariance matrix is -inv(Hessian)
approxPostStd <- sqrt(diag(postCov)) # Computing approximate standard deviations.
names(approxPostStd) <- covnames # Naming the coefficient by covariates

postmode <- as.data.frame(postMode)
colnames(postmode) <- "Coeffecient"
knitr::kable(data.frame(postmode),caption="Coeffecient value of variables")
cat("Hessian matrix:\n")
postCov
mu_nsmall <- postMode["NSmallChild"]
sd_n_small <- approxPostStd["NSmallChild"]

dist <- as.data.frame(rnorm(1000,mu_nsmall,sd_n_small))
colnames(dist) <- "var"

intv <- quantile(dist$var, probs = c(0.025, 0.975))

ggplot(dist, aes(x=var)) + geom_histogram(aes(y = ..density..), color= "white", fill="#a6a6a6") +
  stat_density(geom="line", color="red", size=1) +
  geom_segment(aes(x = intv[1], y = 0, xend = intv[1], yend = 0.20),linetype="dashed",color="blue", size
  geom_segment(aes(x = intv[2], y = 0, xend = intv[2], yend = 0.20),linetype="dashed",color="blue", size
  ggtitle("95% credible interval for NSmallChild") + xlab("Variable-nsmallchild") + ylab("Density")


predict_dist <- function(pred_data, optimres,sampl) {

  var_cov <- rmvnorm(1000,optimres$par,-solve(optimres$hessian))
  p <- c()

  for (i in 1:sampl) {
    p[i] <- (exp(pred_data %*% var_cov[i, ]))/(1 + (exp(pred_data %*% var_cov[i, ])))
  }
  return(p)

}


pred_data <- c(1, 10, 8, 10, (10/10)^2, 40, 1,1)
```

```r
predic <- predict_dist(pred_data,OptimResults,1000)

ggplot() + stat_density(aes(x=predic, y=..scaled..),fill="#AFEEEE") + xlab("Probability") +
           ylab("Density") + ggtitle("Distribution of prediction")
```