TDDE07 - Lab 2

Johannes Hägerlind - johha451
Martin Friberg - marfr370

2021/05/02

1 Assignment 1.

1.1 Question a)

For linear regression with a conjugate prior we have that

$$\sigma^2 \sim Inv - \chi^2(\nu_0, \sigma_0^2)$$

To simulate from the prior, we can draw random draws of σ^2 from

$$X \sim \chi^2(n-1))$$

where n-1 is ν_0 in our case, since we have unknown variance. Thereafter we compute

$$\sigma^2 = \frac{\nu_0 * \sigma_0^2}{X}$$

. To simulate the β:s we can make draws from the multivariate normal distribution

$$\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$$

where μ_0 is our prior belief about the β:s. We thereafter obtain different values for the temperature on specific dates based on the draws of β:s through the quadratic regression:

$$temp = \beta_0 + \beta_1 * time + \beta_2 * time^2 + \epsilon, \epsilon \sim N(0, \sigma^2).$$

The regression curves obtained for each of the draws from the default values of the hyperparameters as well as after tuning was done to fit the data can be seen in the plot below. The tuning was based on making the regression lines as similar to the actual data as possible with a peak in mid july of circa 22 degrees as well as lows in january and December. Here, mid july corresponds to 0.53 on the x-axis. The hyperparameters were set to $\mu_0 = -5, 110, -110$, $\sigma_0^2 = 1$, $\nu_0 = 100$, $\Omega_0 = 20 * I_3$

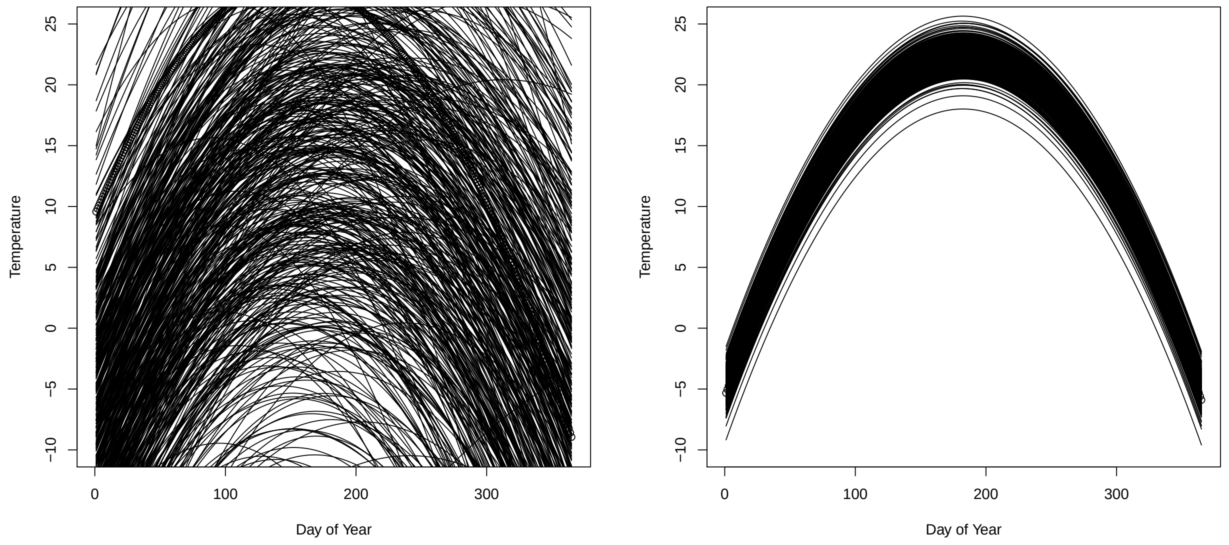


Figure 1: Default regression curves and regression curves after tuning of prior hyperparameters

1.1.1 Code for question 1a

```
########
## subass a)
## Use the conjugate prior for the inear regression model. The prior
## hyperparameters mu0, v0, and sig ma^2_0  shall be set to sensible values.
## start with mu_0 = (-10, 100, -100)^T, Omega_0 = 0.01 * I_3, v_0 = 4 and
## sigma^2_0 = 1.
# where beta_0,1,2 are the regression coefficients
# y|beta,sigma^2, X~N(X*beta, sigma^2*I_n)


library(mvtnorm)
set.seed(12345)

# Function for calculating the temperature
calculateTemp <- function(beta0, beta1, beta2, time, error){
  return(beta0 + beta1 * time + beta2 * time^2 + error)
}


data <- read.table('tempLinkoping.txt')
colnames(data) <- c("time", "temp")
data <- data[-1,]
data$temp  <- as.numeric(data$temp)
data$time <- as.numeric(data$time)

# what's the avg temp in july?
#tempp <-0
#for (i in 180:210){
#  tempp <-tempp + data$temp[i]
#}
#avgtemp <- tempp/30

# what's the avg temp i Feb? seems to be colder than jan
#tempp <-0
#for (i in 30:60){
#  tempp <-tempp + data$temp[i]
#}
#avgtemp <- tempp/30

sigmaSquared0 <- 1
v0 <- 100
mu0 <- c(-5,110,-110)
omega0 <- 20 *diag(3)
nDraws <- 500
# Simulating from the prior: draw X ~chisq(v0)
xDraws <- rchisq(nDraws, v0)
# compute sigma^2 = (df*sample variance/X)
sigmaSquared <- v0*sigmaSquared0/xDraws

# Draw Betas from the N~(u_0, sigma^2*CovarianceMatrix)
betaDraws <- matrix(0, nrow=nDraws, ncol=length(mu0))
for (i in 1:nDraws){
  betaDraws[i,] <- rmvnorm(n=1, mean=mu0, sigma=sigmaSquared[i]*solve(omega0))
```

```
}

errorDraws <- c()
for (i in 1:nDraws){
  errorDraws[i] <- rnorm(n=1, mean=0, sd=sigmaSquared[i])
}

regressionCurves <- matrix(0, nrow=length(data$time), ncol=nDraws)
for (i in 1:nDraws){
  regressionCurves[,i]<-calculateTemp(betaDraws[i,1],
                                      betaDraws[i,2],
                                      betaDraws[i,3],
                                      data$time,
                                      errorDraws[i])
}

plot(regressionCurves[,1],
     ylim=c(-10,25),
     xlab="Day of Year",
     ylab="Temperature")
for (i in 2:nDraws){
  lines(regressionCurves[,i])
}
```

1.2 Question b)

A fuction was written to simulate draws from the joint posterior distribution of β_0, β_1, β_2 and $\sigma * 2$. To simulate from the posterior we have that

$$\sigma^2|y \sim Inv - \chi^2(\nu_n, \sigma_n^2)$$

and

$$\beta|\sigma^2 \sim N[\mu_n, \sigma^2\Omega_n^{-1}]$$

To simulate the draws of σ^2 the same procedure as in a) was used. For β the simulated σ^2:s were used to draw values from the multivariate normal distribution. The marginal posterior distribution of the parameters are shown in the histograms below.

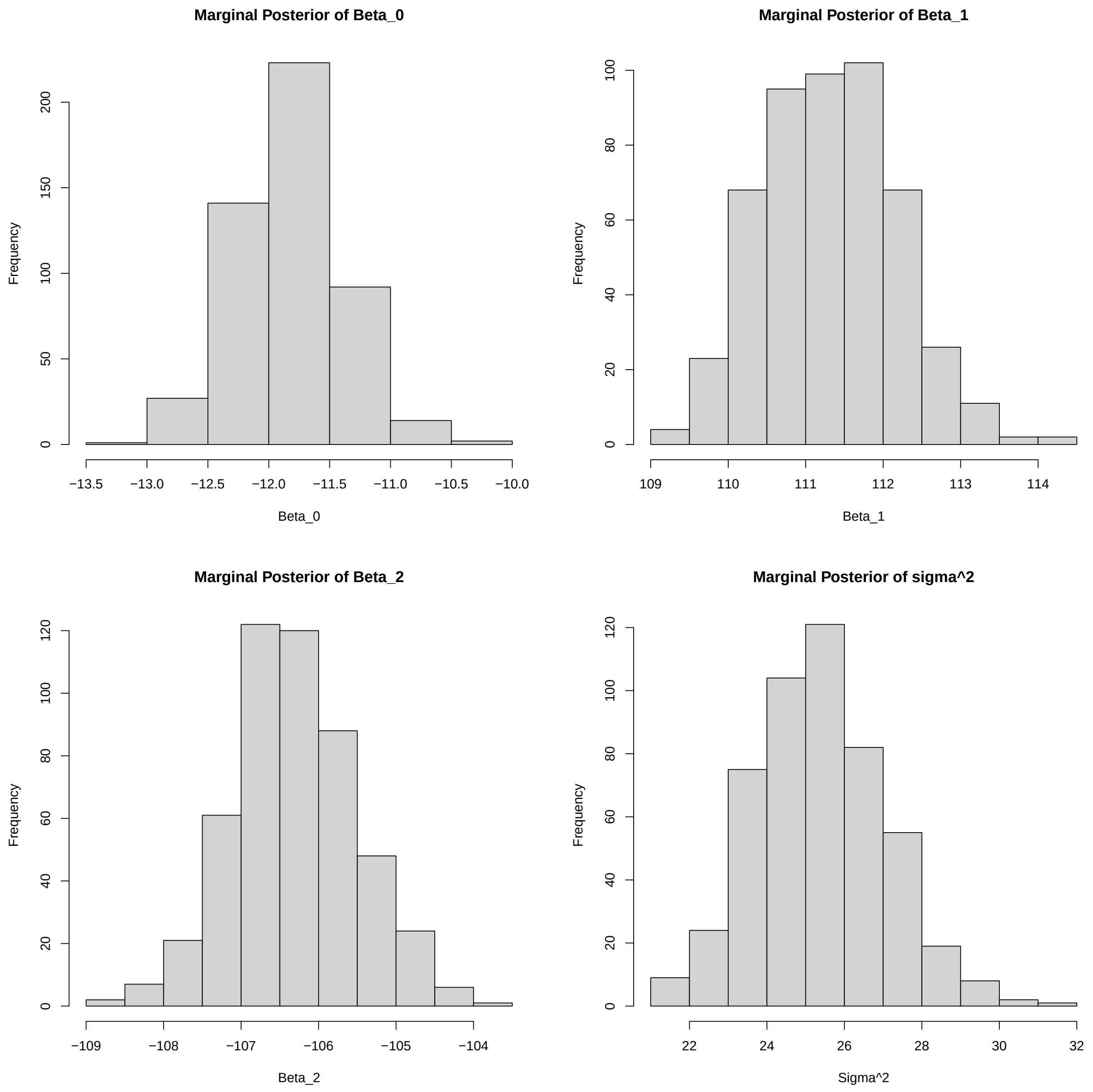
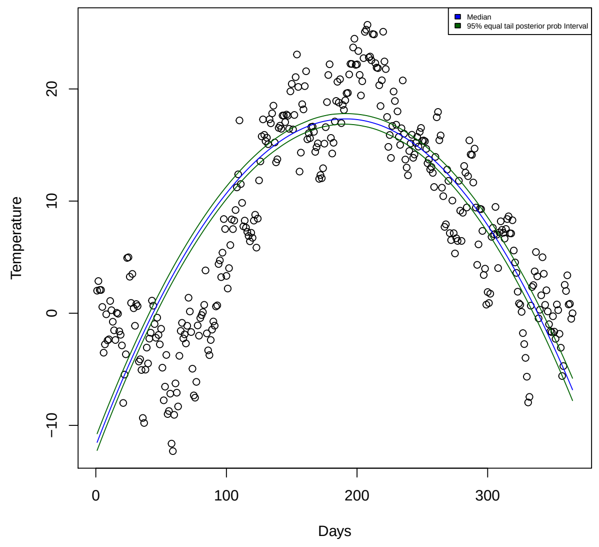


Figure 2: Histograms for the marginal posterior of $beta_0$, $beta_1$, $beta_2$ and $sigma^2$

1.2.1 Question b ii)

A scatter plot of the temperature data was made. By calculating the temperature of all the times for each draw of the β:s, the plot was overlain with the posterior median of all the times. The scatter plot was also overlain with the 95% equal tail posterior probability interval. What can be concluded is that the 95% equal tail posterior probability interval does not include most of the data points which is due to the fact that the probability intervals are the probability intervals of our posterior and not of the actual data points.



1.2.2 Code for question 1b

```r
## subass b)
# Write a function that simulates draws from the joint posterior distribution
# of beta 0, beta1, beta2, sigma^2

# calculate b_hat from slide 4, lecture 5
# we need to set the x values in matrix form
# xi1 = 1 for all i to be able to multiply with beta0.
# xi2 = time, xi3 = time^2
X <- cbind(1, data$time, data$time^2)

mu0<-matrix(mu0)

b_hat <- solve(t(X)%*%X) %*% t(X) %*% data$temp

mu_n <- solve(( t(X) %*% X ) + omega0) %*%
  ( t(X) %*% X %*% b_hat + omega0 %*% mu0)

omega_n <- t(X) %*% X + omega0

v_n <- v0 + length(data$time)

vn_sigmaSq_n <- v0*sigmaSquared0 +
  (t(data$temp) %*% data$temp +
     (t(mu0) %*% omega0 %*% mu0) - (t(mu_n) %*% omega_n %*% mu_n))
```

```r
sigmaSq_n <- vn_sigmaSq_n/v_n

# Simulating sigma^2 from the joint posterior distribution
# Draw X ~ chi^2(v_n)
xDrawsPost <- rchisq(nDraws, v_n)

# Compute sigma^2 = v_n * sigma^2_n
sigmaSquared_post <- v_n*sigmaSq_n[1,1]/xDrawsPost


# simulating betas
betaDrawsPost <- matrix(0, nrow=nDraws, ncol=length(mu_n))
for (i in 1:nDraws){
  betaDrawsPost[i,] <- rmvnorm(n=1, mean=mu_n, sigma=sigmaSquared_post[i]*solve(omega_n))
}

  hist(betaDrawsPost[,1],
       xlab="Beta_0",
       main="Marginal Posterior of Beta_0")


hist(betaDrawsPost[,2],
     xlab="Beta_1",
     main="Marginal Posterior of Beta_1")



hist(betaDrawsPost[,3],
     xlab="Beta_2",
     main="Marginal Posterior of Beta_2")
hist(sigmaSquared_post,
     xlab="Sigma^2",
     main="Marginal Posterior of sigma^2")
### ii)
# Make a scatter plot of the temperature data and overlay a curve for the
# posterior median of the regression function f(time) = beta_0 + beta_1*time +
# beta_0 * time^2 . Also, overlay curves for the 95% equal tail intervals of
# f(time)

# For every draw of betas, calculate temp for each of the times
# Getting a 365x500 matrix, columns are temp values for each draw of betas
# for a specific time

# Regression Function without error term
calculateTemp2 <- function(beta0, beta1, beta2, time){
  return(beta0 + beta1 * time + beta2 * time^2)
}

tempEstimates <- matrix(0, ncol=nDraws, nrow=length(data$time))
for (i in 1:length(data$time)){
  for (j in 1:nDraws){
    tempEstimates[i,j] <- calculateTemp2(betaDrawsPost[j,1],
                                         betaDrawsPost[j,2],
                                         betaDrawsPost[j,3],
                                         data$time[i])
  }
}

tempMedians <- c()
tempQuants <- matrix(0, nrow=length(data$time), ncol=2)
for (i in 1:length(data$time)){
  tempMedians[i] <- median(tempEstimates[i,])
  tempQuants[i,]=quantile(tempEstimates[i,], probs=c(0.025, 0.975))

}

plot(data$temp,
     xlab="Days",
     ylab="Temperature")
lines(tempMedians,
      col="blue")
lines(tempQuants[,1],
      col="darkgreen")
lines(tempQuants[,2],
      col="darkgreen")
legend("topright",
       c("Median", "95% equal tail posterior prob Interval"),
       fill=c("blue", "darkgreen"),
       cex=0.5
       )
```

1.3 Question c)

To located the time where the temperature is the maximal, i.e max(f(time)), we can take the derivative of the f(time) function and set it equal to 0. So we get

$$df(time) = \beta_1 + 2\beta_2 * time$$

.

$$\beta_1 + 2\beta_2 * time = 0 \iff time = \frac{-\beta_1}{2\beta_2}$$

For each of the draws, we can then calculate the time that gives the maximum value of the temperature. We call this value $\tilde{x}$ and the simulated posterior distribution of $\tilde{x}$ can be seen in the histograms below.

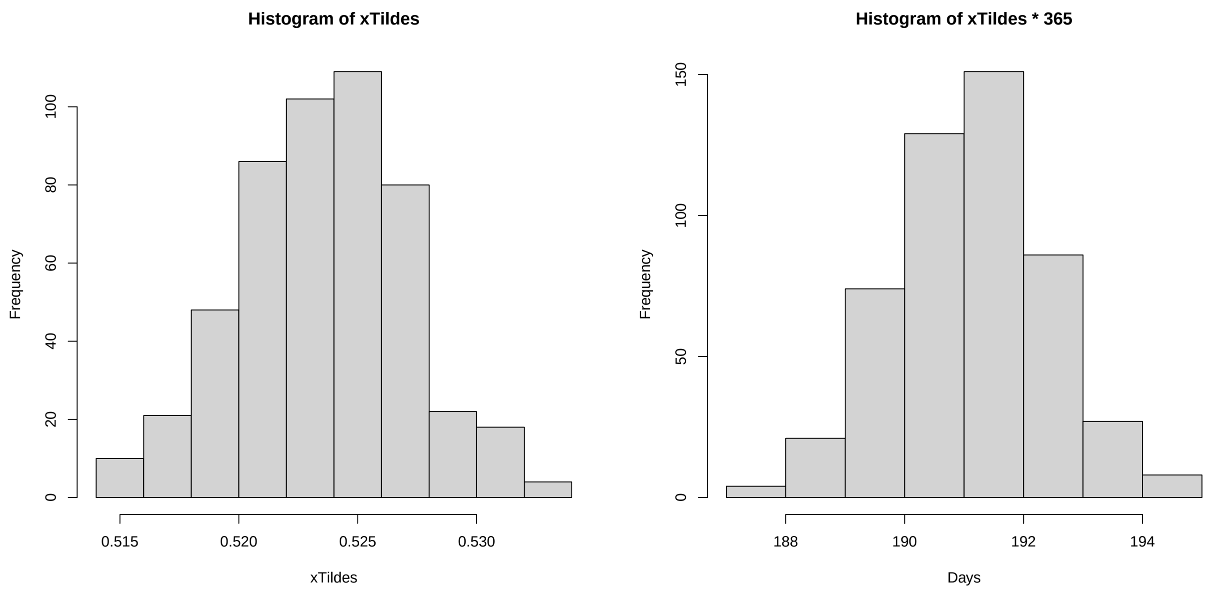


Figure 3: posterior distribution of x tilde in percentage of days and in days

1.3.1 Code for question 1c

```
# subass c)
# to locate the time with the highest expected temperature, we want to find the
# the x value where the derivative is 0. (x = time). By taking the derivative of
# c + ax + bx^2  with regards to x we obtain a + 2bx = 0 -> x = -a/2b

xTildes <- c()
for (i in 1:nDraws){
  xTildes[i] <- -betaDrawsPost[i,2]/(2*betaDrawsPost[i,3])
}

hist(xTildes)

hist(xTildes*365,
     xlab="Days")
```

1.4 Question d)

To suggest a suitable prior that mitigates the risk of overfitting the data due to too many orders of the polynomial model, one can look at decreasing the inital value of the β:s by setting μ_{3-7} to reasonably small values. One can also decrease the risk of overfitting by introducing a penalty term λ in the Ω_0 hyperparameter. This is done by setting Ω_0 to $\lambda * I$ and makes the linear regression model into a ridge regression model. A larger lambda gives a smoother fit and more shrinkage, i.e less overfitting. The reason for this is that Ω_0 is the precision matrix whereas Ω_0^{-1} is the covariance matrix. So a large value of the precision matrix gives us a low value of the covariance matrix.

2 Assignment 2

2.1 Question 2a

```r
library("mvtnorm")

decimals = function(x, n){
  return(format(round(x, n), nsmall = n))
}

womanData <- data.frame(read.table("./WomenWork.dat", header=TRUE))

y = as.matrix(womanData$Work);

# We choose to not standardize the covariates
standardize = FALSE;
if (standardize){
   Index <- 3:dim(womanData)[2]
   womanData[,Index] <- scale(womanData[,Index])
}

X = as.matrix(womanData[,names(womanData) != "Work"]);

nPar <- dim(X)[2]

# Prior
tau = 10
mu = as.matrix(rep(0,nPar))
Sigma = tau^2*diag(nPar)

LogPostLogistic = function(betas,y,X,mu,Sigma){
   linPred = X%*%betas;
   logLik = sum( linPred*y - log(1 + exp(linPred)));
   logPrior = dmvnorm(betas, mu, Sigma, log=TRUE);
   logPost = logLik + logPrior; # propotional to
   return(logPost);
}

initValues <- matrix(0, nPar, 1)
logPost = LogPostLogistic;

optimRes <- optim(initValues,LogPostLogistic,gr=NULL,y,X,mu,Sigma,
                  method=c("BFGS"),control=list(fnscale=-1),hessian=TRUE)

modes = optimRes$par;

covariates = colnames(womanData)[-1]
glmModel = glm(Work ~ 0 + ., data=womanData, family=binomial)
mle = as.vector(glmModel$coefficients)

posterior_mode_comp_mle = data.frame(
   covariates = covariates,
   posterior_modes = modes,
   MLE = mle
)

negative_inverse_hessian = -solve(optimRes$hessian);
```

Below is a Table that shows the posterior mode ($\tilde{\beta}$) for the regression coefficients. The column to the right shows the MLE of the regression coefficients as a comparison.

covariates	posterior_modes	MLE
Constant	0.6267288	0.6443036
HusbandInc	-0.0197911	-0.0197746
EducYears	0.1802190	0.1798806
ExpYears	0.1675667	0.1675127
ExpYears2	-0.1445967	-0.1443595
Age	-0.0820656	-0.0823403
NSmallChild	-1.3591332	-1.3625024
NBigChild	-0.0246835	-0.0254299

We get that $J_y^{-1}(\tilde{\beta})$ 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
```

We compute an approximate 95% equal tail posterior probability interval for the regression coefficient to the variable NSmallChild

```
Ndraws = 10000
draws = rmvnorm(n = Ndraws, mean = modes, sigma = negative_inverse_hessian)

NSmallChildData = draws[, 7]
sorted_G = sort(NSmallChildData)

a = sorted_G[round(Ndraws*0.025)+1]

b = sorted_G[round(Ndraws*0.975)]
```

95% equal tail posterior probability interval for the regression coefficient to NSmallChild" is:

```
## [1] "[ -2.11 ,  -0.58 ]"
```

We would say that the feature NSmallChild is important in prediction wheater the woman works or not.

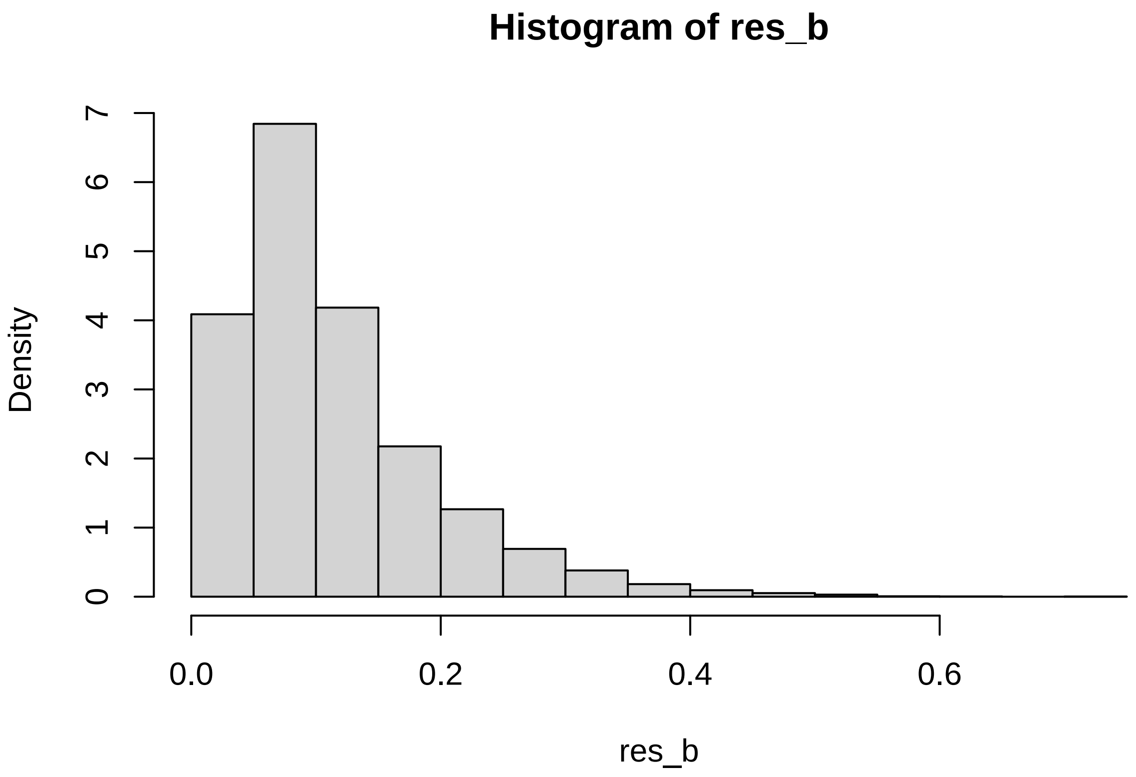
2.2 Question b

We simulate draws from the posterior predictive distribution for a woman with certain values for the covariates.

```r
sigmoid = function(x){
   return(exp(x)/(1+exp(x)))
}
postPredDraws = function(nDraws, mu, Sigma, woman){
   betas = rmvnorm(n = Ndraws, mean = mu, sigma = Sigma);
   y_pred = sigmoid(betas %*% woman);
   return(y_pred);
}

woman_b = c(1, 13, 8, 11, (11/10)^2, 37, 2, 0)

res_b = postPredDraws(10000, modes, negative_inverse_hessian, woman_b)
hist(res_b, freq = FALSE)
```



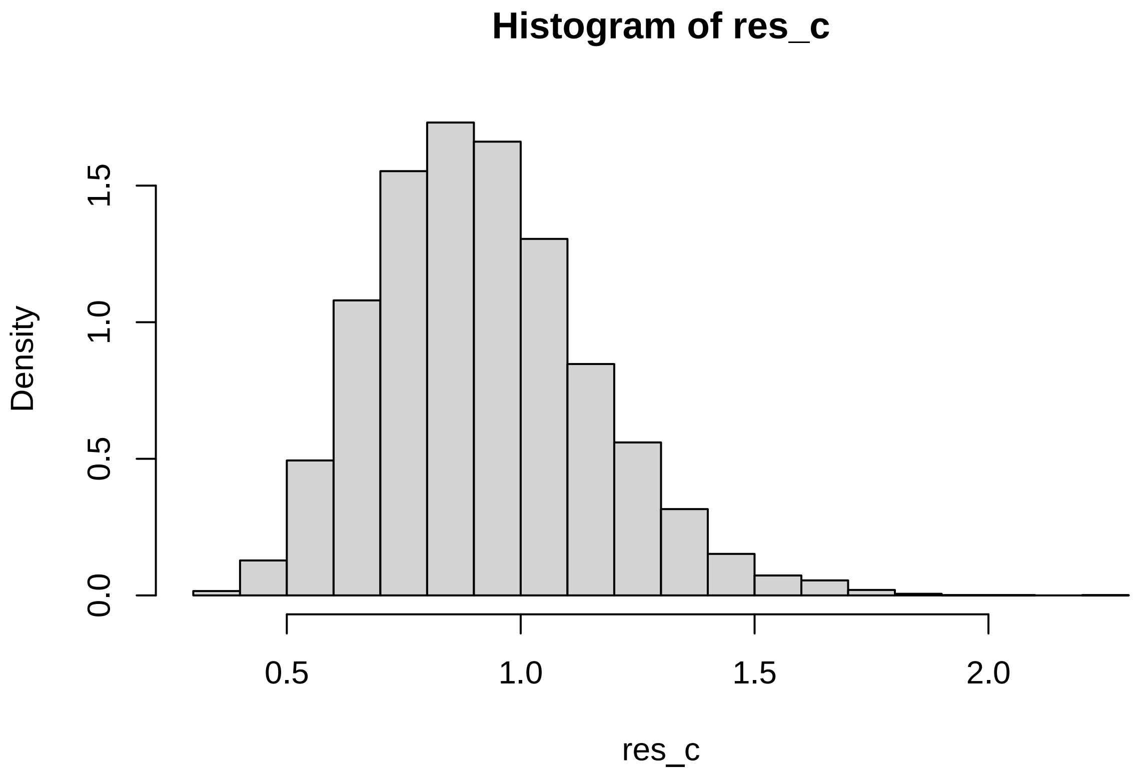
2.3 Question c

We now simulate draws from the posterior predictive distribution for 8 woman with certain (same) values for the covariates.

```r
postPredDraws8 = function(nDraws, mu, Sigma, woman){
   betas = rmvnorm(n = Ndraws, mean = mu, sigma = Sigma);
   y_pred = sigmoid(betas %*% woman);
   for (i in 1:7) {
      betas = rmvnorm(n = Ndraws, mean = mu, sigma = Sigma);
```

```
        y_temp = sigmoid(betas %*% woman);
        y_pred = y_pred + y_temp
    }
    return(y_pred);
}

res_c = postPredDraws8(10000, modes, negative_inverse_hessian, woman_b)
hist(res_c, freq = FALSE)
```


Histogram of res_c
Density
res_c