Lab 2

Lab 1

Linear and polynomial regression. The dataset TempLinkoping.txt contains daily average temperatures (in Celcius degrees) at Malmslätt, Linköping over the course of the year 2018. The response variable is temp and the covariate is

$$time = \frac{\text{nr of days since begining of year}}{356}$$

The task is to perform a Bayesian analysis of a quadratic regression

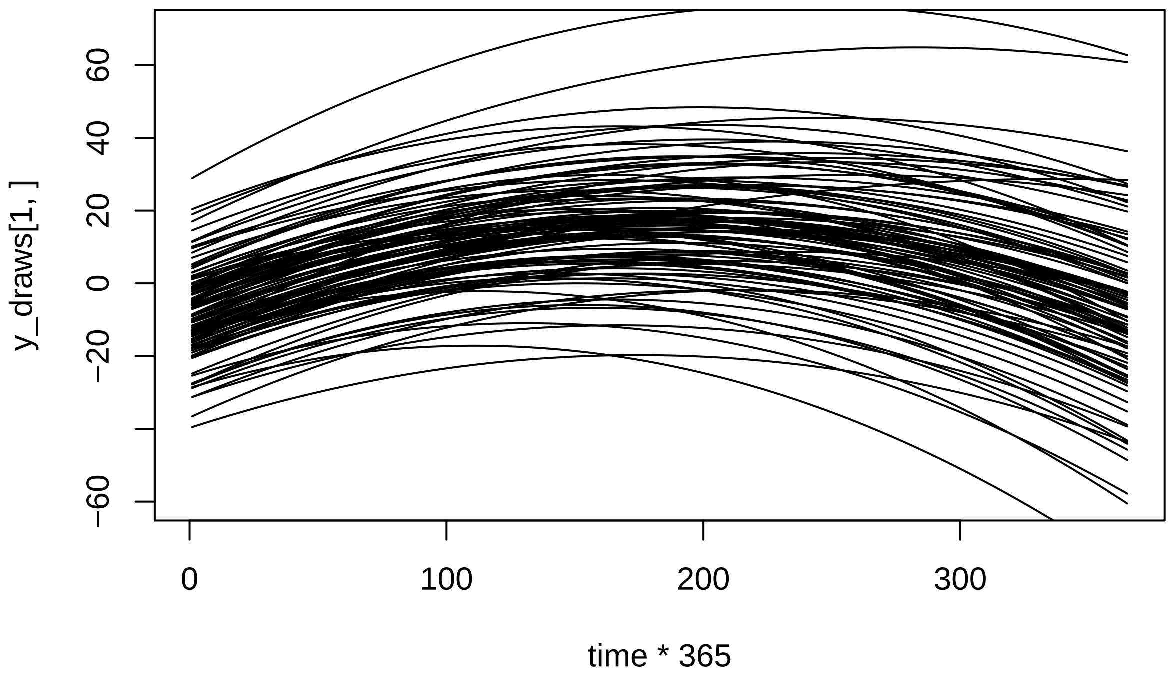
$$temp = \beta_0 + \beta_1 * time + \beta_2 * time^2 + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

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 hyperparameters μ_0, Ω_0, v_0 and σ_0^2 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T, \omega_0 = 0.01 * I_3, v_0 = 4$ and $\sigma_0^2 = 1$. 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 reasonable? 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 $Inv - \chi^2$ simulator from Lab 1.]

```
data = read.table("C:/Users/samue/Documents/LIU/TDDE07/LABS/TDDE07-Bayesian-Learning/Lab 2/TempLinkoping

time = as.numeric(as.vector(data[-1,1]))
temp = as.numeric(as.vector(data[-1,2]))

# 1a)
mu_0 = c(-10, 100, -100)
omega_0 = 0.01*diag(3)
omega_0_inv = 100*diag(3)
v_0 = 4
sigma_0_2 = 1

#draws sigma 2
nr_draws = 100
x_draws = rchisq(nr_draws,v_0)
sigma2_draws = ((v_0)*sigma_0_2)/x_draws

#draws beta given simga
library(mvtnorm)
betas = matrix(, nrow=nr_draws, ncol=3)
for (i in 1:nr_draws) {
  covar = sigma2_draws[i]*omega_0_inv
  betas[i,] = rmvnorm(1, mu_0, covar)
}
A = rep(1, length(time))
B = time
```

```r
C = time^2
x = cbind(A, B, C) #beta order B0, B1, B2
y_draws = betas%*%t(x)

#draw predictions
plot(time*365, y_draws[1,], type = "l", ylim = c(-60,70))

for (i in 2:nr_draws) {
  points( y_draws[i,], type="l")
}
```



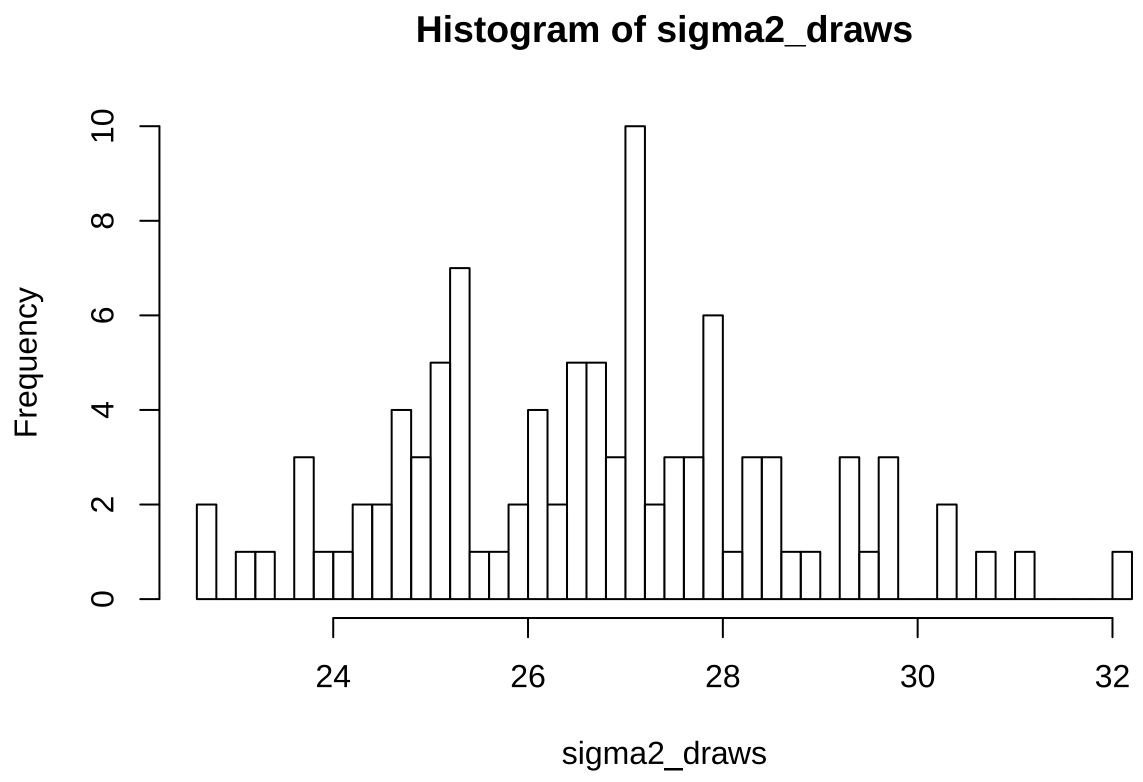
Do the collection of curves look reasonable? It looks reasonable, since the temperature is higher during summer (middle of the curves) and lower during winter (start and end of the curve).

b

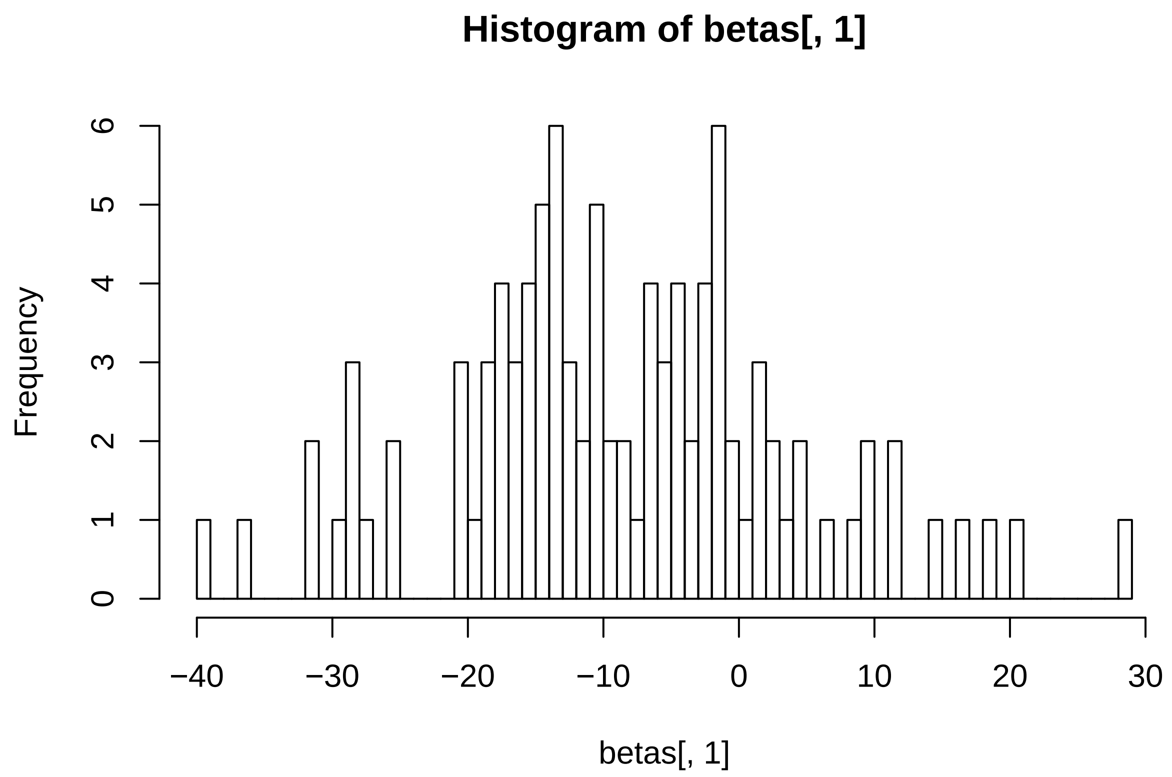
Write a program that simulates from the joint posterior distribution of $\beta_0, \beta_1, \beta_2$ and σ. Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function $f(time) = \beta_0 + \beta_1 * time + \beta_2 * time^2$, 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?

```r
#1b
#posterior
library(matlib)
```

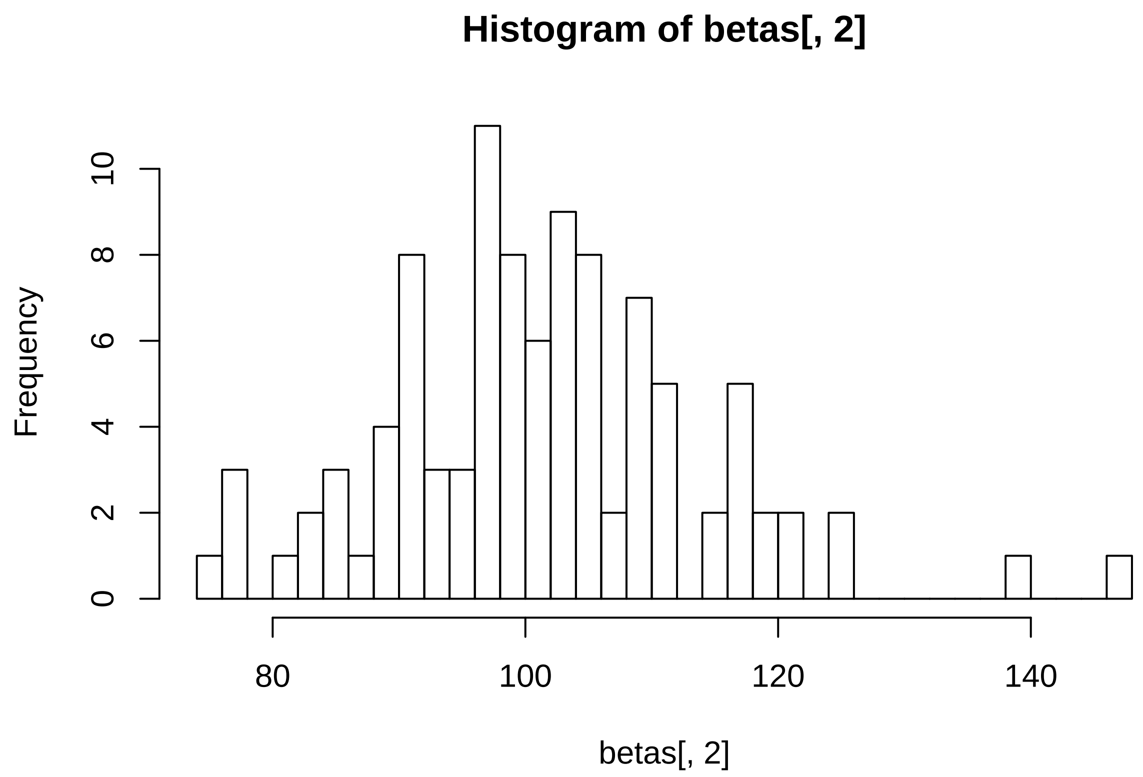
```
beta_hat = inv( t(x)%*%x ) %*% (t(x)%*%temp)
mu_n = inv( t(x)%*%x+omega_0 )%*%( t(x)%*%x%*%beta_hat + omega_0%*%mu_0)
omega_n = t(x)%*%x+omega_0
v_n = v_0 + length(time)
sigma_n_2 = (v_0*sigma_0_2 + (t(temp)%*%temp + t(mu_0)%*%omega_0%*%mu_0 - t(mu_n)%*%omega_n%*%mu_n))/v_

#draw sigma2
nr_draws = 100
sigma_draws = rchisq(nr_draws,v_n)
sigma2_draws = ((v_n)*sigma_n_2)/sigma_draws

betas_post = matrix(, nrow=nr_draws, ncol=3)
for (i in 1:nr_draws) {
  covar = sigma2_draws[i]*inv(omega_n)
  betas_post[i,] = rmvnorm(1, mu_n, covar)
}

#plot hist of beta and sigma
hist(sigma2_draws, breaks = 50)
```



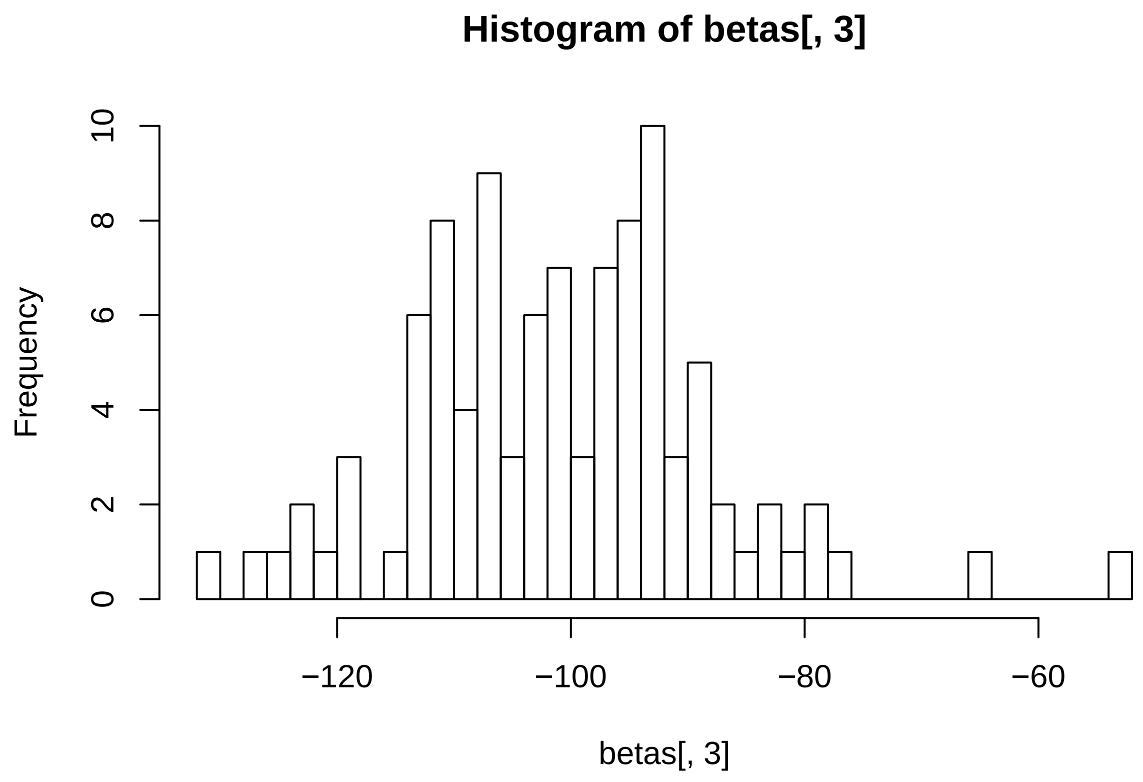
```
hist(betas[,1], breaks = 50)
```



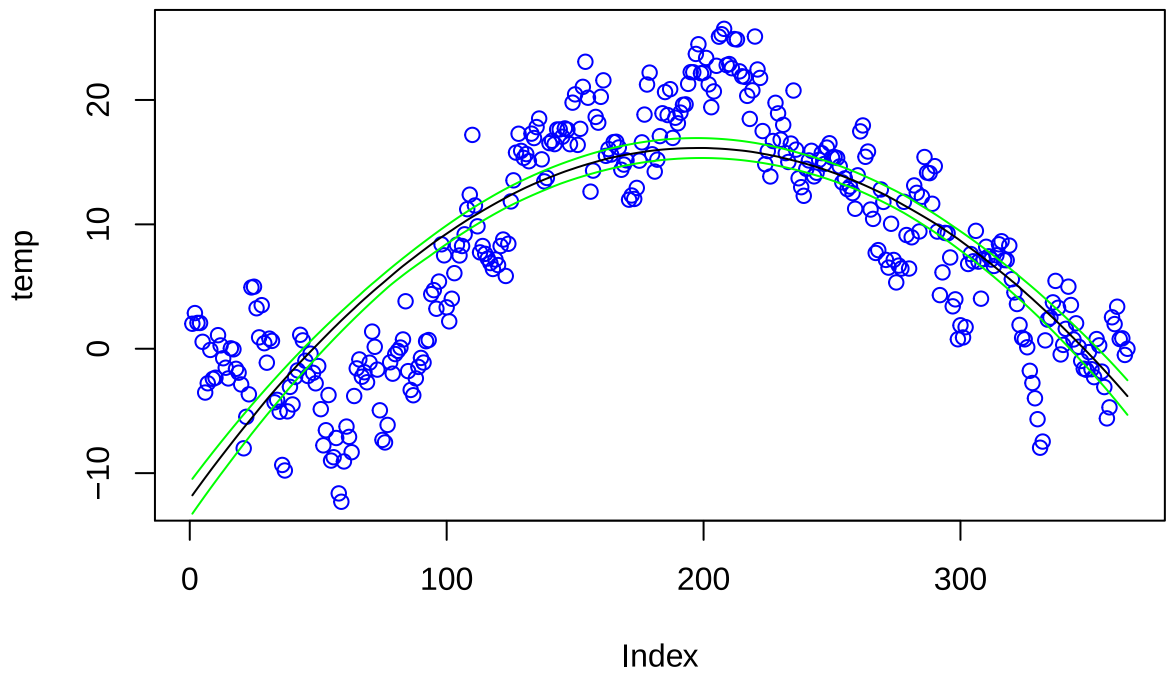
```r
hist(betas[,2], breaks = 50)
```



```
hist(betas[,3], breaks = 50)
```



```r
# calculate y, predicted temp for all draws
y_post = betas_post%*%t(x)

#Calculate mean for y each day form all predictions
y_median = apply(y_post,2,median)
plot(temp, col="blue")
points(time*365, y_median, type="l")

#approx curves for 95% equal tail by sorting
#sorted_y_post = apply(y_post,2,sort,decreasing=F)
#points(sorted_y_post[round(nr_draws*0.025),], type="l", col="red")
#points(sorted_y_post[round(nr_draws*0.975),], type="l", col="red")

quantile_y_post = apply(y_post,2,quantile, probs=c(0.025,0.975))
points(quantile_y_post[1,], col="green", type="l")
points(quantile_y_post[2,], col="green", type="l")
```



Does the interval bands contain most of the data points? Should they?

The posterior probability inteval show were our model will be with 95 % certainty given all the inputs. It does not show were the actual y is with 95% probability.

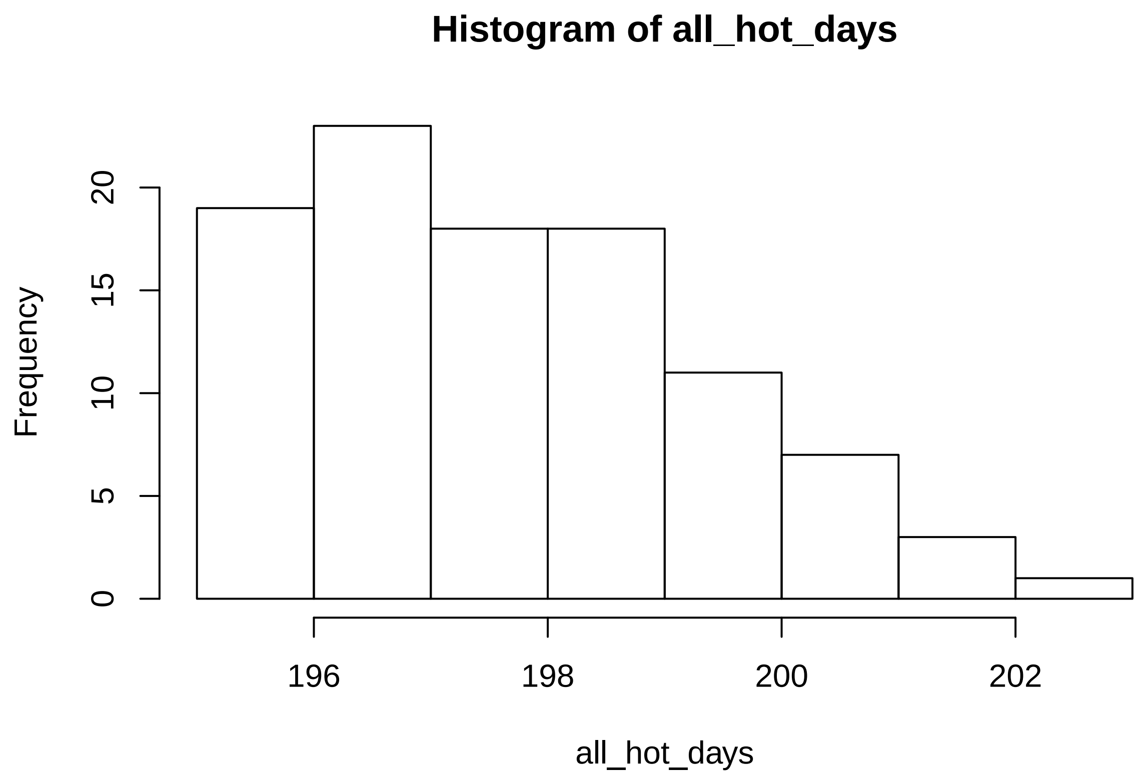
c

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 $\bar{x}$. Use the simulations in b) to simulate from the posterior distribution of $\bar{x}$. [Hint: the regression curve is a quadratic. You can find a simple formula for $\bar{x}$, given $\beta_0, \beta_1, \beta_2$]

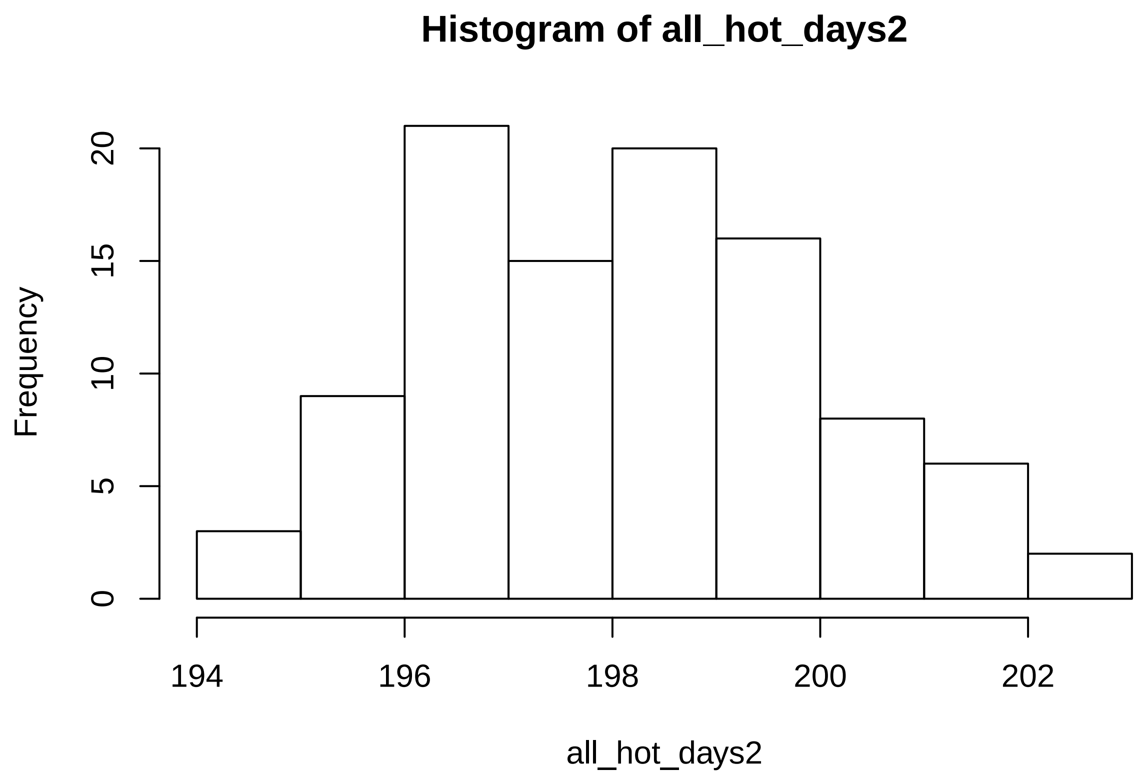
```
x_hat = which.max(y_median)
y_median[x_hat]
```

```
## [1] 16.13894
```

```
#From previous data
all_hot_days = apply(y_post, 1, which.max)
hist(all_hot_days, breaks=10)
```



```r
#from derivation
all_hot_days2 = (-betas_post[,2]/(2*betas_post[,3]))*365
hist(all_hot_days2)
```



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 will have a laplace distribution for the betas with $\mu_0 = 0$ and $\Omega_0 = I_8 * \lambda$. Were lambda is the smoothing coefficient that decides how much the betas are allowed to differ from zero and μ_0 sets the mean of these to be zero in the beginning. This will make a few betas go into the fat tails of the laplace distribution and a few to end up at 0, the prior mean.

Part 2

Posterior approximation for classification with logistic regression The dataset WomenWork.dat contains n = 200 observations (i.e. women) on the following nine variables:

a)

Consider the logistic regression:

$$Pr(y = 1|x) = \frac{exp(x^T\beta)}{1 + exp(x^T\beta)}$$

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:

$$\beta|y,X \sim N(\tilde{\beta}, J_y^{-1}(\tilde{\beta}))$$

where $\tilde{\beta}$ is the posterior mode and $J(\tilde{\beta}) = -\delta^2 \frac{\log(p(\tilde{\beta}|y))}{\delta\beta\delta\beta^T}|_{\beta=\tilde{\beta}}$ is the observed Hessian evaluated at the posterior mode. Note that $\delta^2 \frac{\log(p(\tilde{\beta}|y))}{\delta\beta\delta\beta^T}$ is an 8×8 matrix with second derivatives on the diagonal and cross-derivatives $\delta^2 \frac{\log(p(\tilde{\beta}|y))}{\delta\beta_i\delta\beta_j}$ on the offdiagonal. 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 $\tilde{\beta}$ and $J(\tilde{\beta})$ are computed by the optim function in R. Use the prior $\beta \sim N(0, \tau^2 I))$ I), with $\tau = 10$.Your report should include your code as well as numerical values for β and $J^{-1}(\tilde{\beta})$ for the WomenWork data. Compute an approximate 95% credible interval 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).]

```
filename='C:/Users/samue/Documents/LIU/TDDE07/LABS/TDDE07-Bayesian-Learning/Samuel lab 2/WomenWork.txt'
Data<-read.table(filename,head=TRUE)
y=(as.vector(Data[,1]))
x=as.matrix(Data[,-1])
numberOfParameters=length(x[1,])
library(mvtnorm)

#a)
logPosteriorProbability=function(beta,y,x,mu,sigma){

  lengthBeta=length(beta)
  linearPrediction=x%*%beta

  logMaximumLikelihood=sum(y*linearPrediction-log(1+exp(linearPrediction)))


  logPrior= dmvnorm(beta, matrix(0,lengthBeta,1), sigma, log=TRUE);
  return(logMaximumLikelihood+logPrior)

}

tau=10
betaStart=as.vector(rep(0,numberOfParameters))
priorMean=as.vector(rep(0,numberOfParameters))
priorStd=tau^2*diag(numberOfParameters)


OptimResults<-optim(betaStart,logPosteriorProbability,gr=NULL,y,x,priorMean,priorStd,method=c("BFGS"),c

postMode <- OptimResults$par


postCov <- -solve(OptimResults$hessian)
y=qnorm(c(0.025,0.975),postMode[7],postCov[7,7])
postMode
```

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

```r
postCov
```

```
##               [,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
```

```r
y
```

```
## [1] -1.655602 -1.062665
```

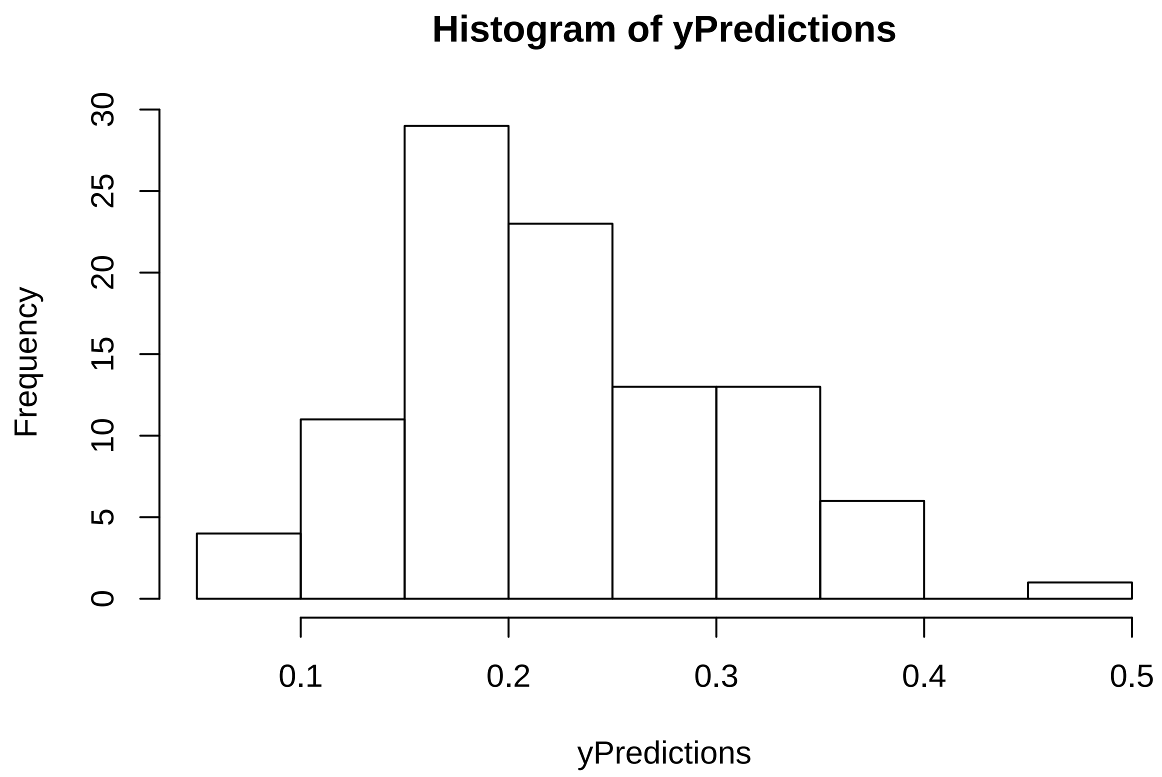
Would you say that this feature is an important determinant of the probability that a women works?

Yes it is important. The absolute value of the feature variable is with 95% confidence above one (two sided interval). If we change the linearPrediction input from 0 to 1 (the derivative of the logistic is at its maximum at 0) we will get a probability change from 50% to 73 %. Therefore a change of 1, in the number of childs below six, will have a big impact.

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.]

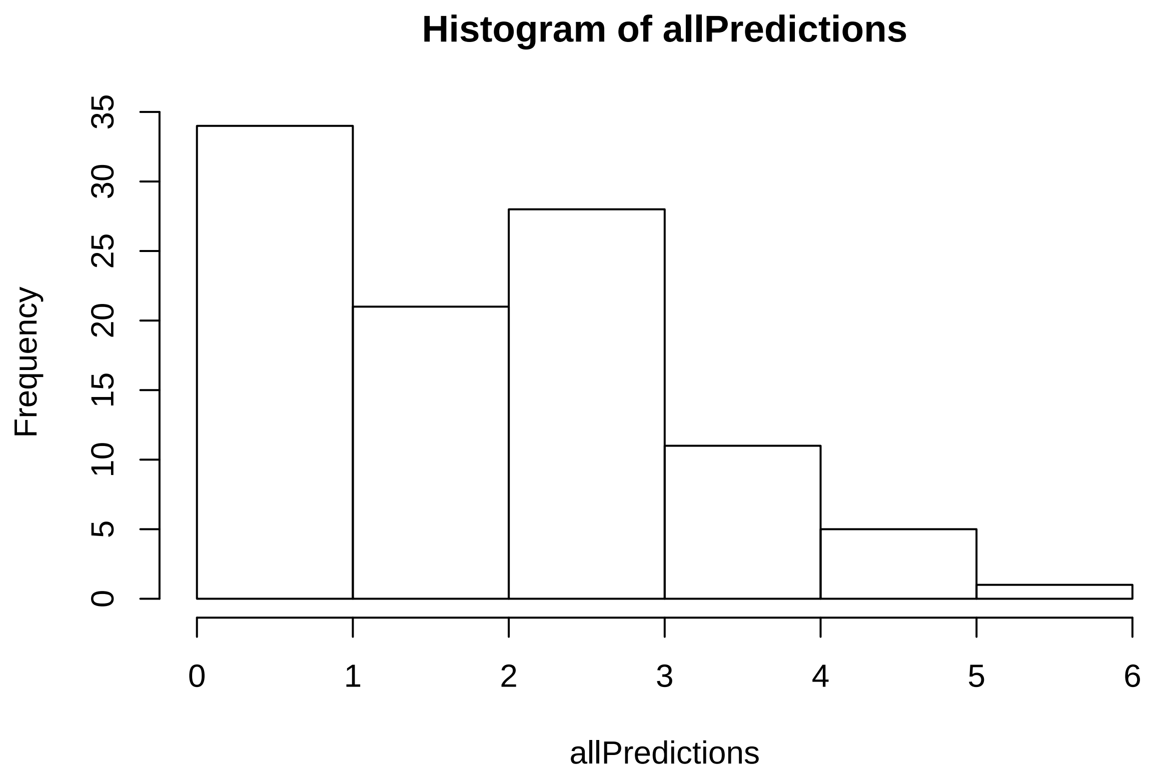
```r
library(MASS)
numberOfSamples=100
simulatedBetas=mvrnorm(numberOfSamples,postMode,postCov)
xForB=c(1, 10, 8, 10, 1, 40, 1, 1)
linearPrediction=xForB%*%t(simulatedBetas)
yPredictions=exp(linearPrediction)/(1+exp(linearPrediction))
hist(yPredictions)
```



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?]

```r
numberOfSamples=1000
simulatedBetasC=mvrnorm(numberOfSamples,postMode,postCov)
linearPredictionC=xForB%*%t(simulatedBetas)
yPredictions=exp(linearPrediction)/(1+exp(linearPrediction))
allPredictions=c()
for (i in yPredictions) {
  allPredictions=cbind(allPredictions,rbinom(1,10,i))
}

hist(allPredictions)
```


Histogram of allPredictions
Frequency
0
5
10
15
20
25
30
35
0
1
2
3
4
5
6
allPredictions