Lab 2

Weng Hang Wong

4/27/2020

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

$$time = \frac{the\ number\ of\ days\ since\ beginning\ of\ year}{365}$$

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

$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \varepsilon, \varepsilon \sim N(0, \sigma^2)$$

(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 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3$, $v_0 = 4$ 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.

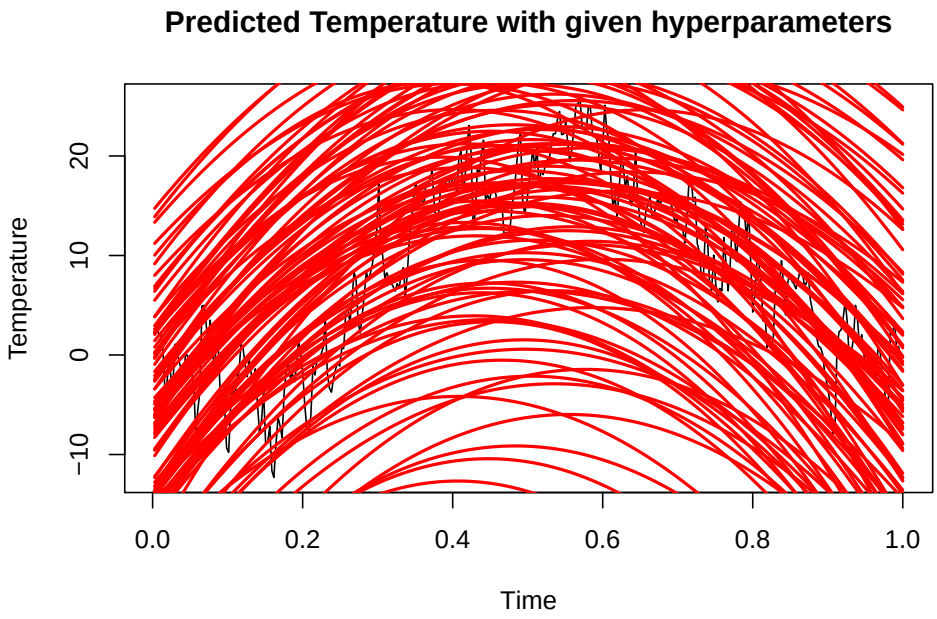
We have the joint prior β and σ^2

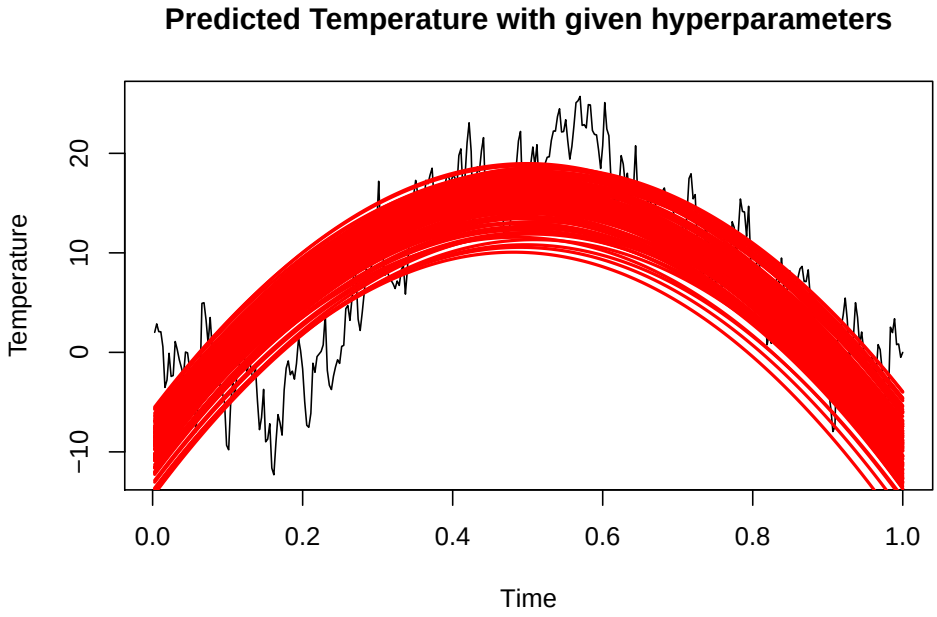
$$\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$$
$$\sigma^2 \sim Inv - \chi^2(v_0, \sigma_0^2)$$

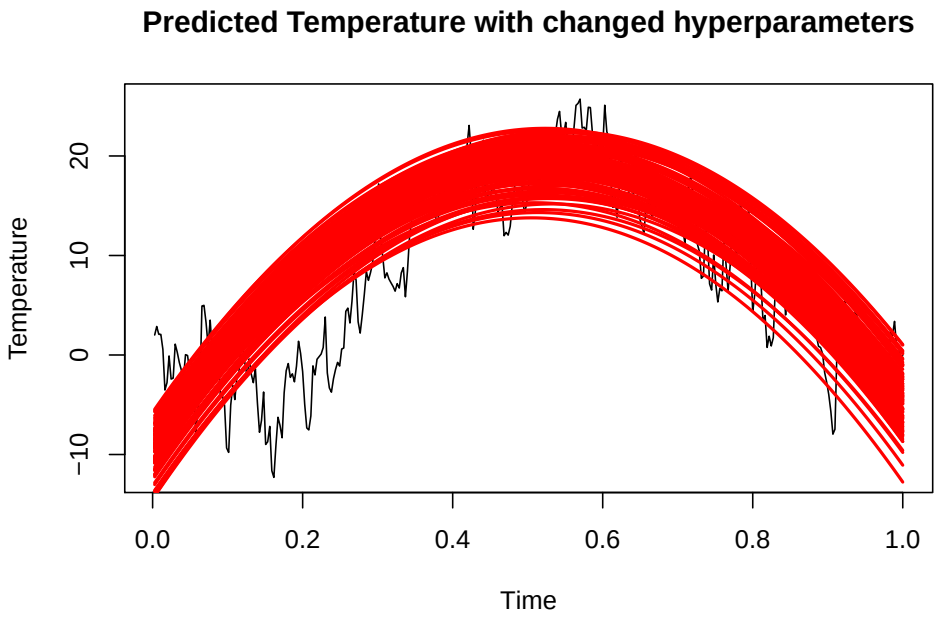
By simulating 100 draws, the red curves of the first figure is the regression curves simulating from the give hyperparameters $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3$, $v_0 = 4$ and $\sigma_0^2 = 1$. The curves are messed in the graph and basically can't explain the data.

So, first we set the hyperparaters $\sigma_0^2 = 0.03$, it has obvious changing with lower value of of σ_0^2. From the second figure below, the regression curves are more concentrated to the data.

Second, the change of $\mu_0 = (-10, 110, -105)^T$ and $\sigma_0^2 = 0.03$ lead the curves fit better to the data with our prior beliefs in the third figure below.

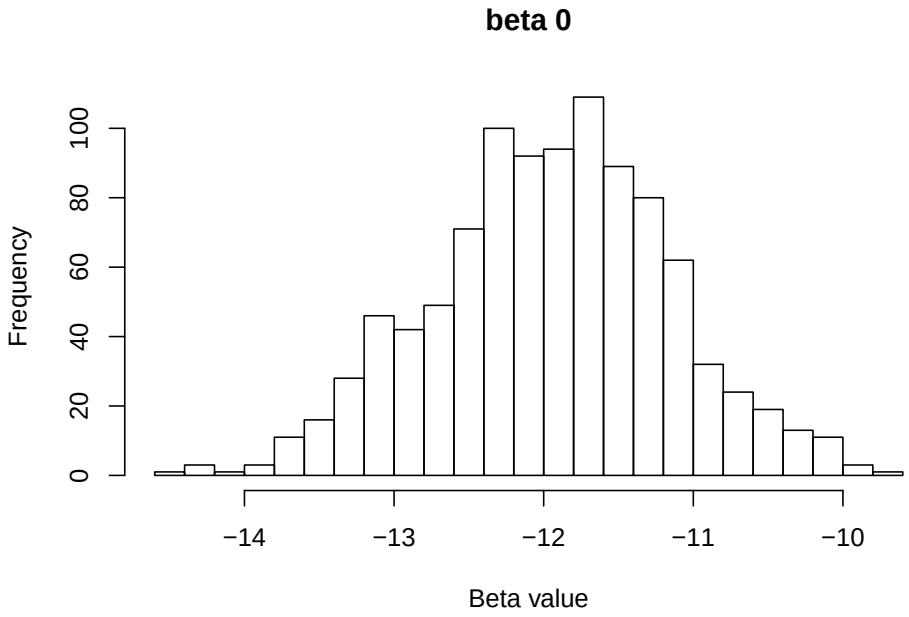

Predicted Temperature with given hyperparameters
Temperature
20
10
0
−10
0.0
0.2
0.4
0.6
0.8
1.0
Time

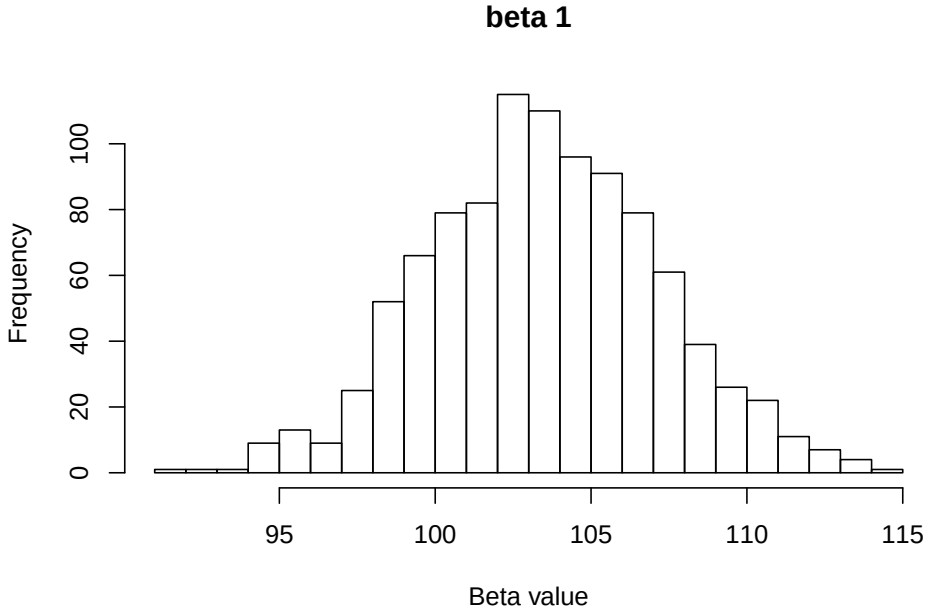

Predicted Temperature with given hyperparameters
Temperature
20
10
0
−10
0.0
0.2
0.4
0.6
0.8
1.0
Time

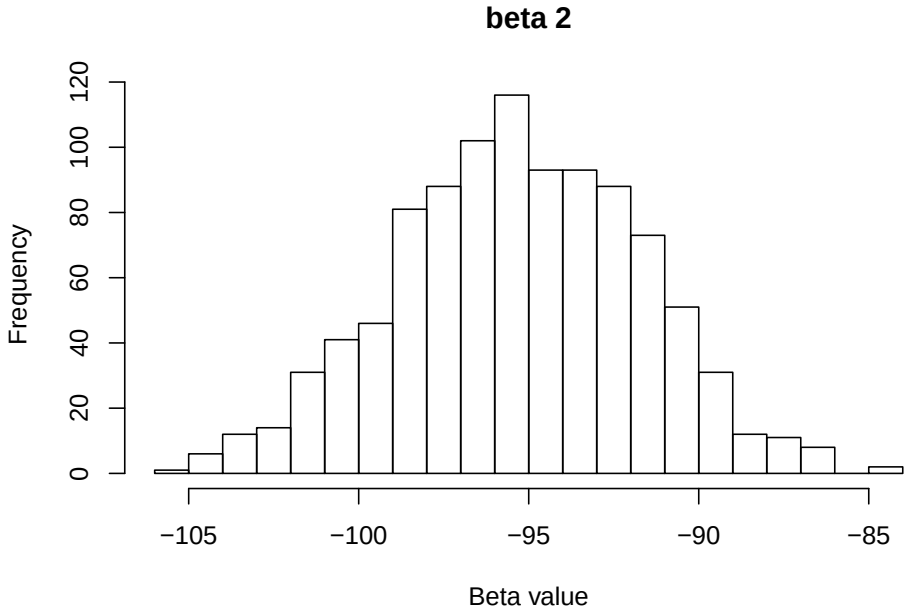


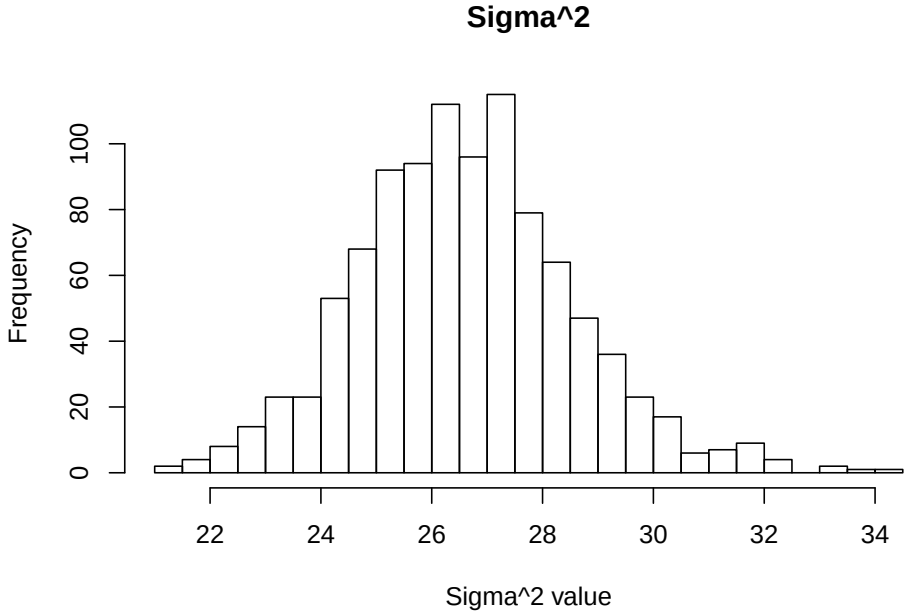
(b)Write a program that simulates from the joint posterior distribution of β_0,β_1,β_2,and σ^2. Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function $f(time) = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$, 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?

From the graph below, the parameters are simulated from the joint posterior distribution. The marginal posteriors for each parameter β_0,β_1,β_2,and σ^2 are shown below.

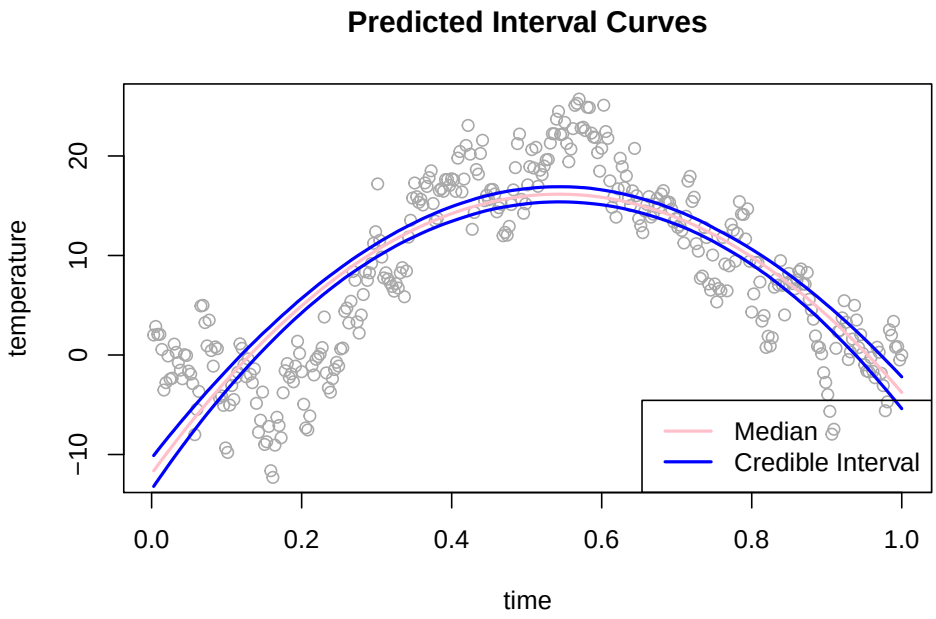

beta 0
Frequency
0
20
40
60
80
100
−14
−13
−12
−11
−10
Beta value


beta 1
Frequency
0
20
40
60
80
100
95
100
105
110
115
Beta value





Here is a scateer plot of the temperature data with the median and credible interval curves. However, most of the data points are not contained in the 95% posterior credible interval, they should not contained most of the data points, since it didn't include the ε in the regression function and the uncentainty parameter here has particular 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 $\widetilde{x}$ Use the simulations in b) to simulate from the posterior distribution of $\widetilde{x}$

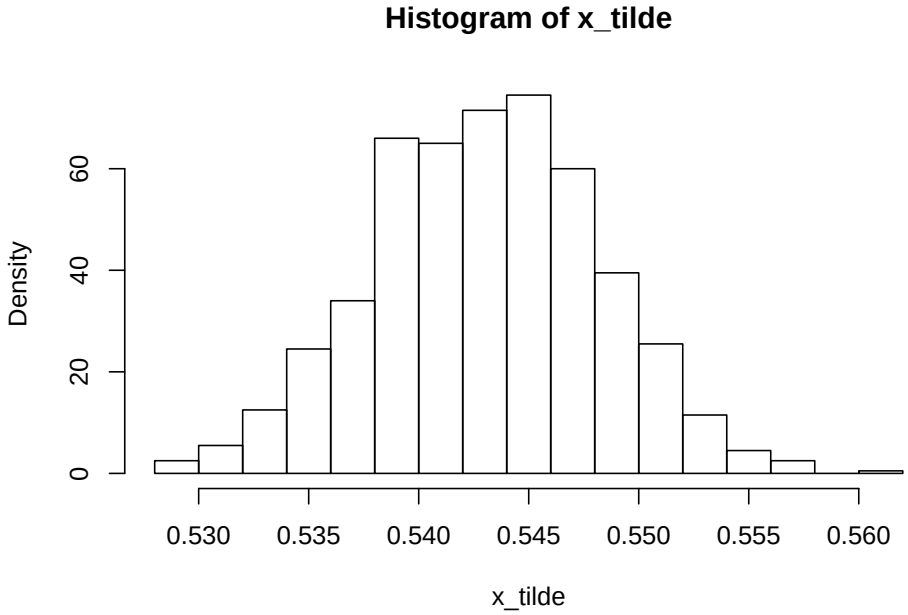
The first derivative of f(time) will be maximal when it equal to zero.

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2$$

$$0 = \beta_1 + 2\beta_2 x$$

$$\widetilde{x} = \frac{-\beta_1}{2\beta_2}$$

```
## The expected highest expected temperature is 0.5430181
```



(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.

In order to avoid overfitting on a higher order model, we can use the prior:

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

A larger λ here gives the smoother fitting curves on the model.

2. Posterior approximation for classification with logistic regression

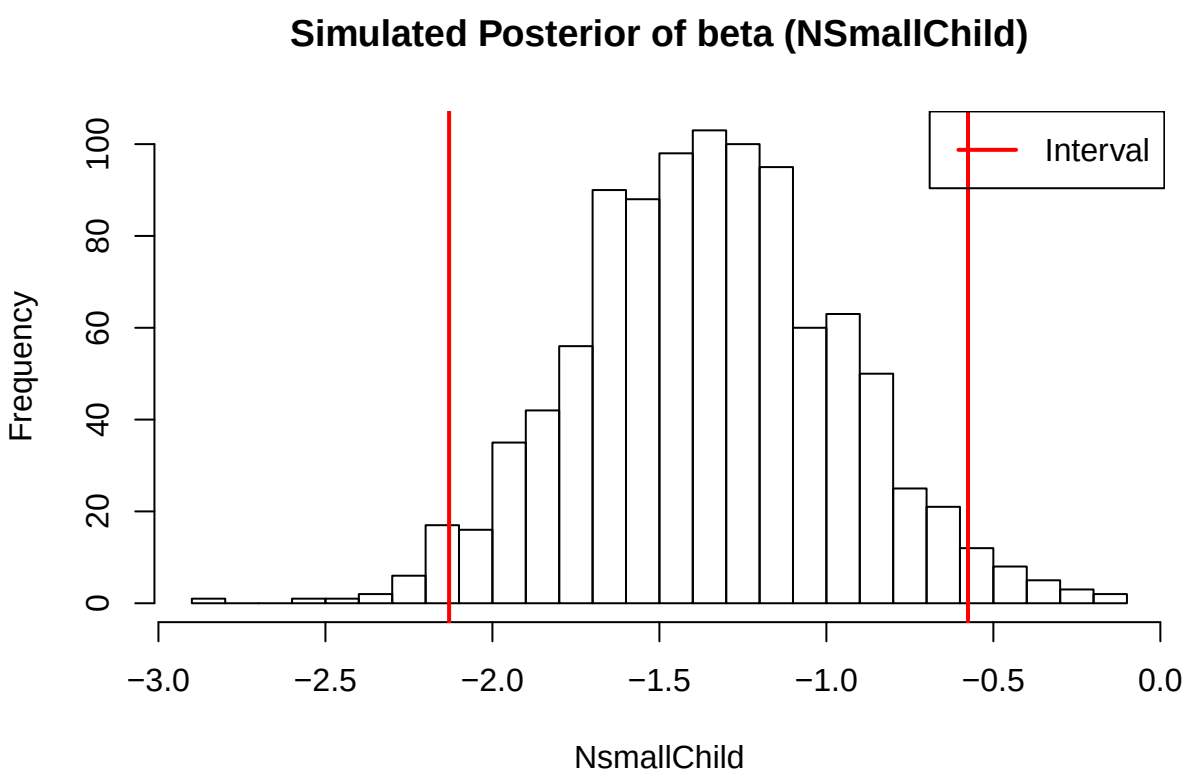
(a)

Beta	Posterior_mode
0	0.6267288
1	-0.0197911
2	0.1802190
3	0.1675667
4	-0.1445967
5	-0.0820656
6	-1.3591332
7	-0.0246835

	Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
Constant	2.2660225	0.0033389	-0.0654512	-0.0117914	0.0457807	-0.0302934	-0.1887484	-0.0980239
HusbandInc	0.0033389	0.0002528	-0.0005610	-0.0000313	0.0001415	-0.0000359	0.0005067	-0.0001444
EducYears	-0.0654512	-0.0005610	0.0062182	-0.0003558	0.0018963	-0.0000032	-0.0061346	0.0017527
ExpYears	-0.0117914	-0.0000313	-0.0003558	0.0043517	-0.0142491	-0.0001341	-0.0014690	0.0005437
ExpYears2	0.0457807	0.0001415	0.0018963	-0.0142491	0.0555787	-0.0003299	0.0032083	0.0005120
Age	-0.0302934	-0.0000359	-0.0000032	-0.0001341	-0.0003299	0.0007185	0.0051842	0.0010953
NSmallChild	-0.1887484	0.0005067	-0.0061346	-0.0014690	0.0032083	0.0051842	0.1512622	0.0067689
NBigChild	-0.0980239	-0.0001444	0.0017527	0.0005437	0.0005120	0.0010953	0.0067689	0.0199723

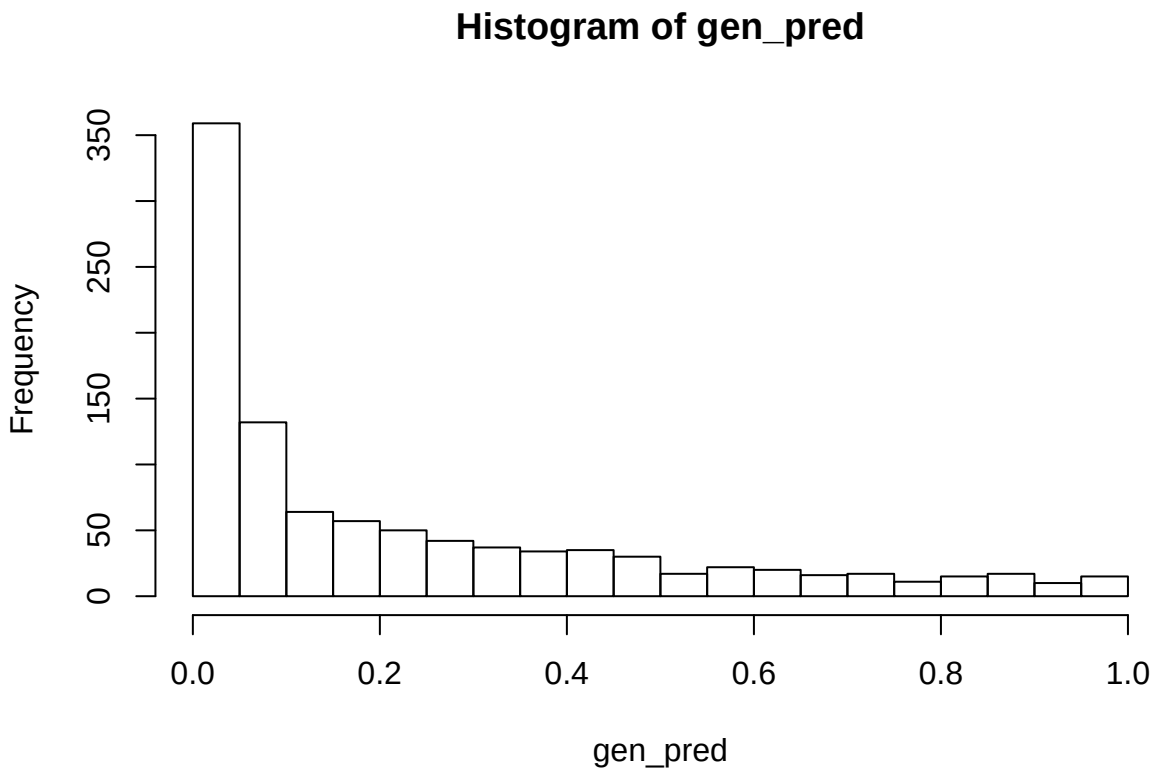
	Hessian Value
Constant	1.5053314
HusbandInc	0.0158998
EducYears	0.0788556
ExpYears	0.0659675
ExpYears2	0.2357513
Age	0.0268041
NSmallChild	0.3889244
NBigChild	0.1413233

	Beta	Verified_mode
Constant	0	0.6443036
HusbandInc	1	-0.0197746
EducYears	2	0.1798806
ExpYears	3	0.1675127
ExpYears2	4	-0.1443595
Age	5	-0.0823403
NSmallChild	6	-1.3625024
NBigChild	7	-0.0254299




```
## The Credible Interval is: -2.129689 -0.5759195
```

(b)



Appendix

```
#~1.a
library(mvtnorm)
data = read.table("~/Desktop/Bayesian/732A91_Lab_1/lab2/TempLinkoping.txt",header=T)

#given hyperparameters
mu0=matrix(c(-10,100,-100))
omega0=diag(x=0.01, nrow=3, ncol=3)
v0=4
sigma20=1

#prior
PriorReg = function(mu0,omega0,v0,sigma20){
  set.seed(12345)
  for(i in 1:100){
    #using chi_sq to sample sigma^2
    chi_sample = rchisq(n=1, df=v0)
    sigma2 = v0*sigma20/chi_sample

    #using mvtnorm sample beta
```

```
    beta = rmvnorm(n=1, mean=mu0, sigma=sigma20*solve(omega0))

    #quadratic regression
    quad_regre= beta[1]+beta[2]*data$time+beta[3]*(data$time^2)+rnorm(1,mean=0, sd=sqrt(sigma2))
    lines(x=data$time, y=quad_regre,col="red",lwd=2)
  }
}

### Check the given hyperpara
plot(data, main="Predicted Temperature with given hyperparameters", ylab="Temperature", xlab="Time", ty
PriorReg( mu0, omega0, v0, sigma20)

### change the hyperpara nu
plot(data, main="Predicted Temperature with given hyperparameters", ylab="Temperature", xlab="Time", ty
PriorReg( mu0, omega0, v0, sigma20=0.03)


# Change the hyperpara sigma
plot(data, main="Predicted Temperature with changed hyperparameters", ylab="Temperature", xlab="Time",
PriorReg( mu0=matrix(c(-10,110,-105)), omega0, v0, sigma20=0.03)


#1.b

### find beta hat
n=dim(data)[1]
X = data.frame(intercept=rep(1,n), x1=data$time, x2=data$time^2)
X = as.matrix(X)
y = data$temp
betaHat = solve(t(X)%*%X)%*%t(X)%*%y

### calculate mu, omega, nu sigma
mu_n = solve(t(X)%*%X+omega0) %*% (t(X)%*%X%*%betaHat+omega0%*%mu0)
omega_n = t(X)%*%X+omega0
v_n = v0 + n
sigma2_n = (v0*sigma20+(t(y)%*%y+t(mu0)%*%omega0%*%mu0-t(mu_n)%*%omega_n%*%mu_n))/v_n

### Marginal posterior
set.seed(12345)
paras = NULL
final = NULL
for(i in 1:1000){
  #using chi_sq to sample posterior sigma^2
  chi_sample = rchisq(n=1, df=v_n)
  post_sigma2 = v_n*sigma2_n/chi_sample

  #using mvtnorm sample posterior beta
  post_beta = rmvnorm(n=1, mean=mu_n, sigma=post_sigma2[1]*solve(omega_n))

  paras = cbind(post_beta,post_sigma2)
  final = rbind(paras, final)
}
```

```r
colnames(final) = c("beta0","beta1","beta2","sigma2")

## histogram of each parameters
hist(final[,1], main="beta 0", xlab="Beta value", breaks=30)
hist(final[,2], main="beta 1", xlab="Beta value", breaks=30)
hist(final[,3], main="beta 2", xlab="Beta value", breaks=30)
hist(final[,4], main="Sigma^2",xlab="Sigma^2 value", breaks=30)
### median curve and intervals
post_beta = final[,1:3]

PredictedVal=matrix(0,nrow=n,ncol=nrow(post_beta))
for(i in 1:nrow(post_beta)){
  PredictedVal[,i] = X %*% post_beta[i,]
}


## find median and credible interval
medianInterval=c()
crediInterval = matrix(0,nrow=n,ncol=2)
for(i in 1:n){
  medianInterval[i] = median(PredictedVal[i,])
  crediInterval[i,] = quantile(PredictedVal[i,], c(0.025,0.975))

}

plot(data, main="Predicted Interval Curves", col="darkgrey", ylab="temperature")
lines(data$time,medianInterval, col="pink",lwd=2)
lines(data$time,crediInterval[,1], col="blue",lwd=2)
lines(data$time,crediInterval[,2], col="blue",lwd=2)
legend("bottomright",legend=c("Median", "Credible Interval"), col=c("pink","blue"),lwd=2 )

##1.c

x_tilde = -post_beta[,2]/ (2*post_beta[,3])
cat("The expected highest expected temperature is",mean(x_tilde))

hist(x_tilde, freq=F, breaks=20)

# 2.a

## find the value of beta tilde and J()
data = read.table("WomenWork.dat", header =T)

#make y as vector and X as matrix
y=data[,1]
X=as.matrix(data[,2:9])
colnames(X) = names(data)[2:9]
nPara = dim(X)[2]

tau=10
# prior's hyperpara
mu <- as.vector(rep(0,nPara))
sigma <- tau^2*diag(nPara)
```

```r
# A function that returns regression coeifficient
## calculate the Log(post) = log(llh)+log(prior)

set.seed(12345)
LogPost = function(betaVec, y, X, mu, sigma){
  pred = X%*%betaVec
  #log LLH
  logLLH= sum(y*pred - log(1+exp(pred)) )
  #log prior using dmvnorm from beta vector
  logPrior = dmvnorm(betaVec, mean=rep(0,length(betaVec)), sigma, log=T)
  res = logLLH + logPrior
  return(res)
}

initValue <- as.vector(rep(0,dim(X)[2]));
# Or a random starting vector: as.vector(rnorm(dim(X)[2]))
# Set as OLS estimate: as.vector(solve(crossprod(X,X))%*%t(X)%*%y); # Initial values by OLS

#All arguments except betaVecc which is the one that we are trying to optimize over
# The argument control is a list of options to the optimizer. Here I am telling the optimizer to multip
# Optim finds a minimum, and I want to find a maximum. By reversing the sign of logPost I can use Optim
OptimResults<-optim(initValue,LogPost,gr=NULL,y,X,mu,sigma,method=c("BFGS"),control=list(fnscale=-1),hes

## find the value of Beta and Hessian
PostModeBeta = OptimResults$par
hessianBeta = -solve(OptimResults$hessian) #we want -Inv Hessian
approx_PostStd <- as.matrix(sqrt(diag(hessianBeta)))

# Beta table
library(knitr)
kable( data.frame(Beta = seq(0,7,1), Posterior_mode=PostModeBeta))

#Covariance Matrix
colnames(hessianBeta) = names(data)[2:9]
rownames(hessianBeta) =  names(data)[2:9]
kable(hessianBeta)

# Hessian table
rownames(approx_PostStd) = names(data)[2:9]
colnames(approx_PostStd) = "Hessian Value"
kable(approx_PostStd)

## verify my result
glmModel <- glm(Work~0+., data = data, family = binomial)

kable(data.frame(Beta = seq(0,7,1),Verified_mode = glmModel$coefficients))


## find the CI for NSmallChild by simulating from the Post
set.seed(12345)
Post_logis_beta = rmvnorm(n=1000, mean=PostModeBeta, sigma = hessianBeta)

NsmallChild=Post_logis_beta[,7]
```

```r
hist(NsmallChild, main="Simulated Posterior of beta (NSmallChild)", breaks=30)

CI = quantile(NsmallChild, c(0.025, 0.975))

abline(v=CI[1], col="red", lwd=2)
abline(v=CI[2], col="red", lwd=2)
legend("topright", legend = "Interval", col="red", lwd=2)

cat("The Credible Interval is:", CI)



## 2.b

library(mvtnorm)

X = matrix( c(
  constant=1,
  age=40,
  NSmallChile=1,
  NBigChild=9,
  educYear=8,
  expYear=10,
  expYear2=1,
  husbandIC=10
  ), nrow=1)


gen_post = function(beta, X){
    y1 = (exp((X)%*%beta)) / ( 1+exp((X)%*%beta) )
    return(y1)
}

gen_pred = apply(Post_logis_beta,1,gen_post,X)

hist(gen_pred, breaks=30)
```