Lab 2 Report Zxl version

Zuxiang Li, Jooyoung Lee

4/19/2020

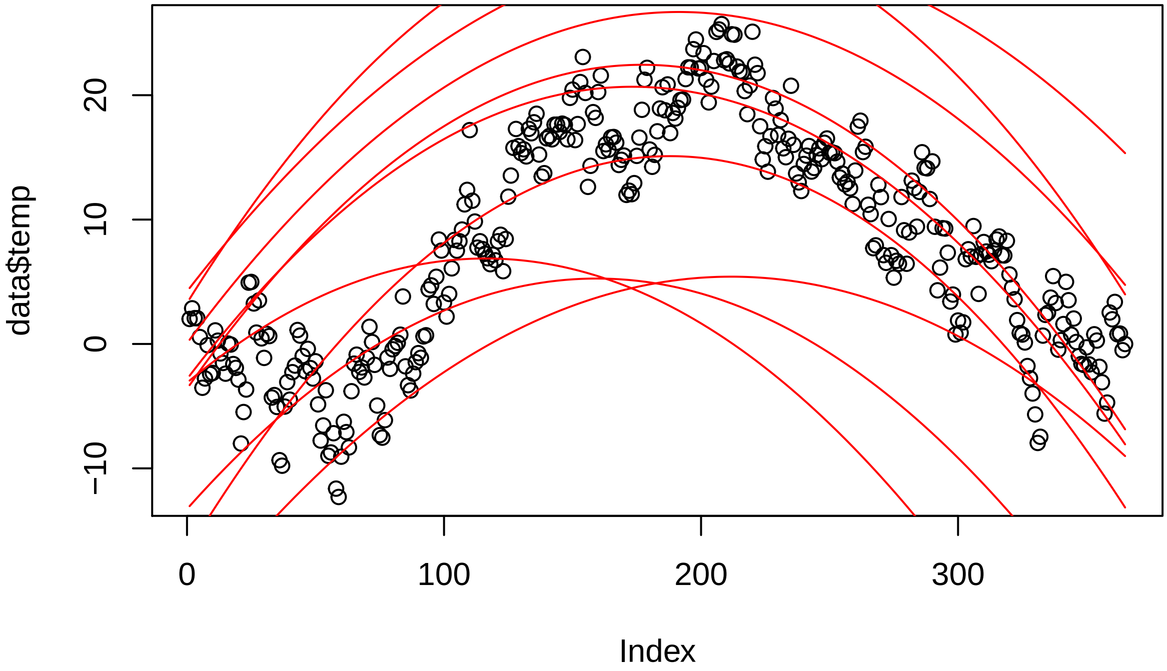
1. Linear and polynomial regression The dataset TempLinkoping.txt contains daily temperatures (in Celcius degrees) at Malmslätt, Linköping over the course of the year 2016 (366 days since 2016 was a leap year). The response variable is temp and the covariate is

$$time = \frac{the\ number\ of\ days\ since\ beginning\ of\ year}{366}$$

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

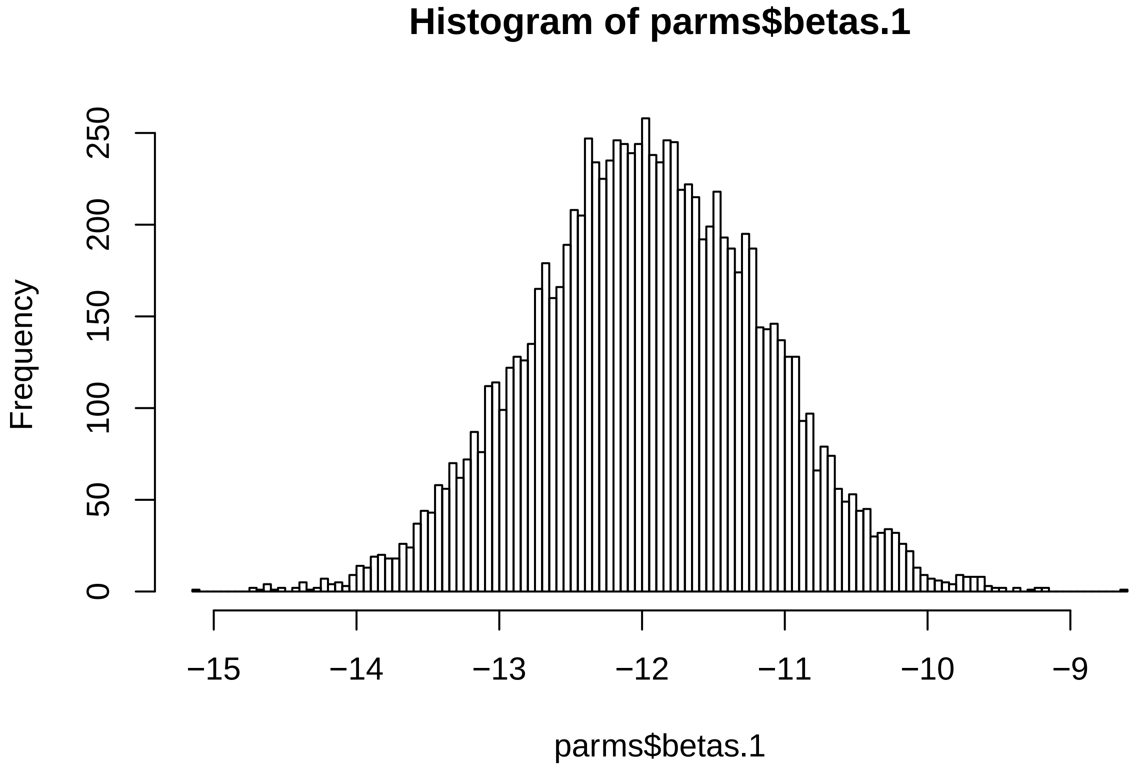
$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon\ N(0, \sigma^2)$$

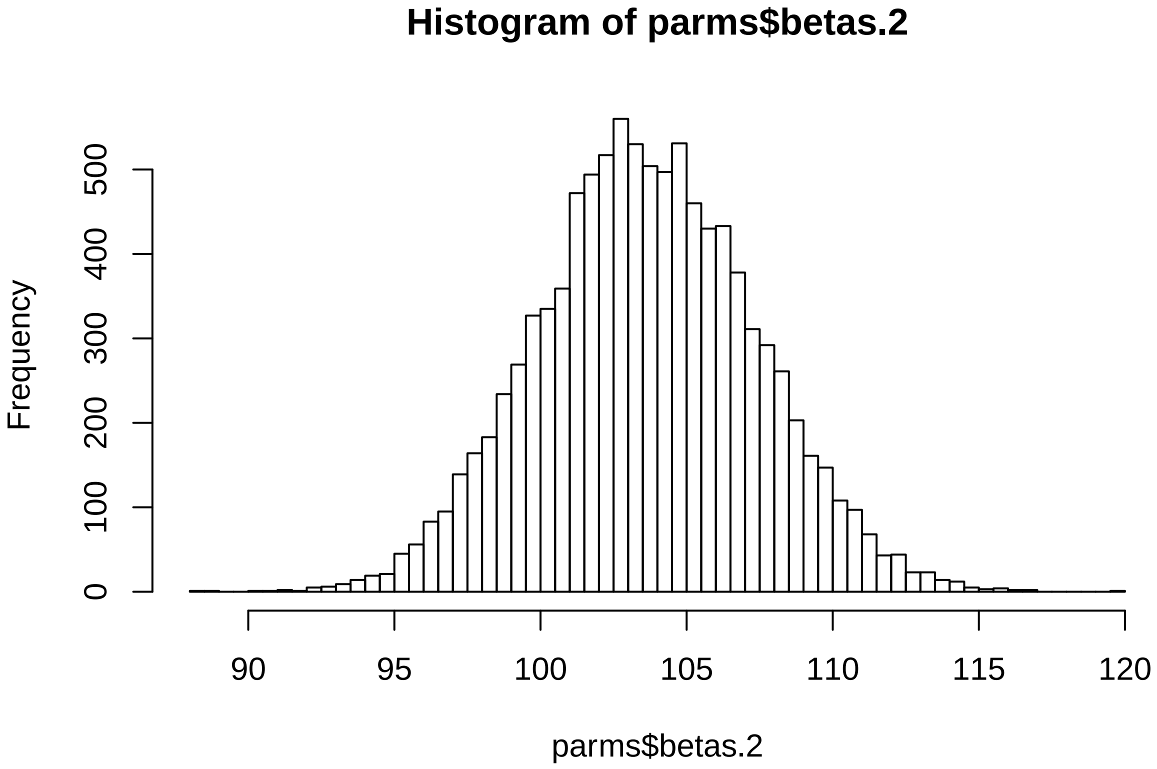
(a) Determining the prior distribution of the model parameters. Use the conjugate prior for the linear regression model. Your task is to set the prior hyperparameters μ_0, Ω_0, v_0 *and* σ^2 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3, v_0 = 4$ *and* $\sigma^2 = 1$. 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.]

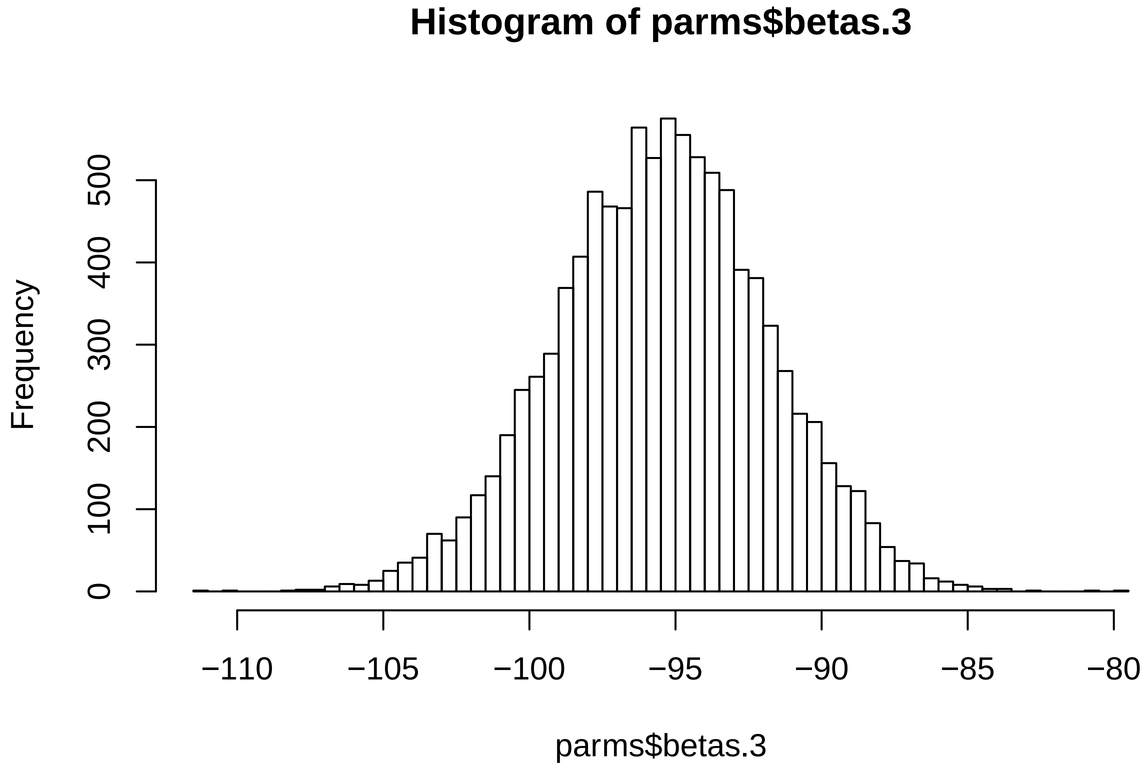


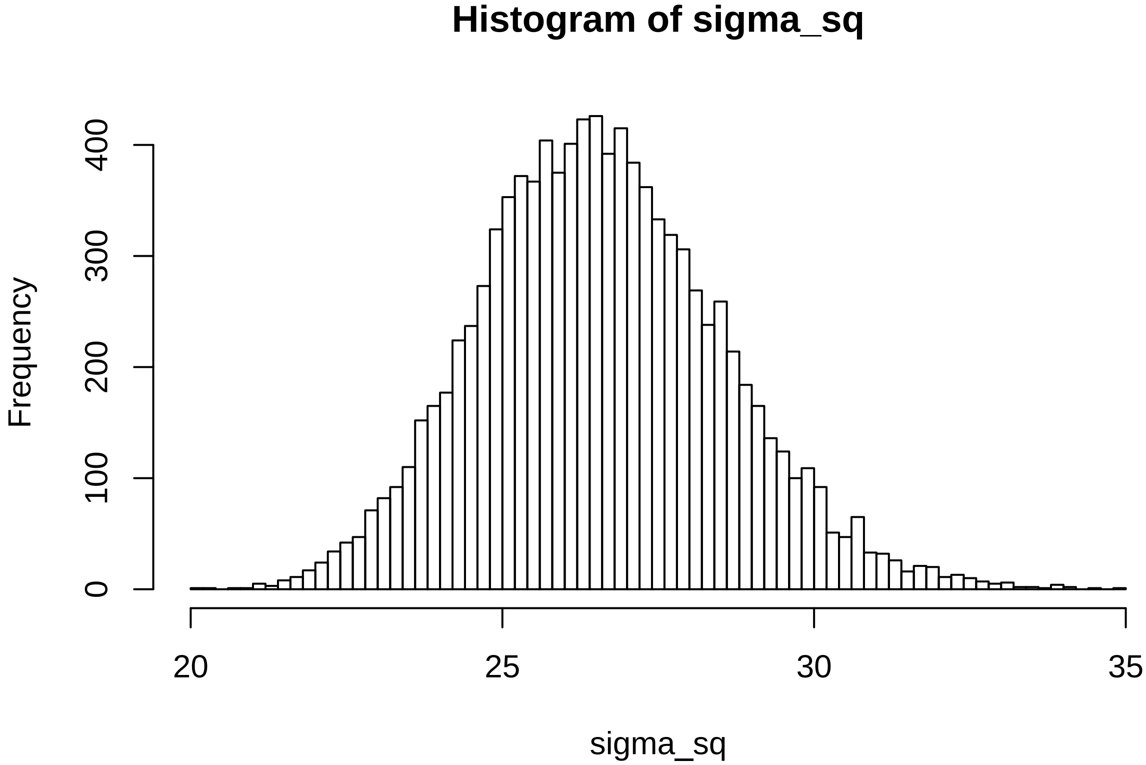
Red lines in the plot represent regression curves from prior, the curves look reasonable since they all follow a certain pattern as same as our temp data.

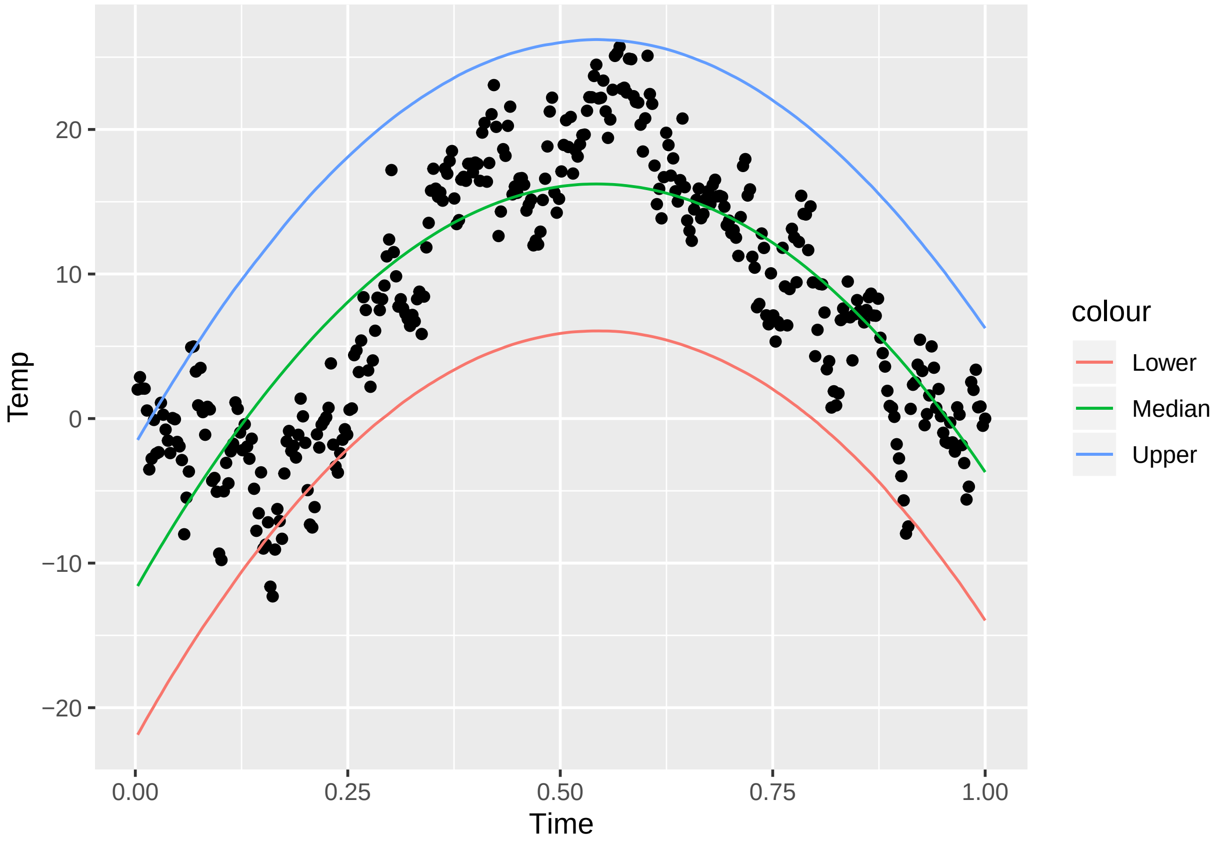
(b) Write a program that simulates from the joint posterior distribution of $\beta_0, \beta_1, \beta_2 and \sigma^2$ Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of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






Histogram of parms$betas.3
Frequency
0
100
200
300
400
500
−110
−105
−100
−95
−90
−85
−80
parms$betas.3


Histogram of sigma_sq
Frequency
0
100
200
300
400
20
25
30
35
sigma_sq



The 95% equal tailed credible interval includes most of the data. This does make sense because the regression is based on joint posterior distribution, and such joint posterior distribution is derived by using the data. If the prior belief was very strong and such belief had completely different pattern than that of data points, then the result here could be different. However, regression using assumed prior distribution on the first step of this simulation,had concave pattern - thus it did not greatly interfere the likelihood while producing, posterior distribution of the parameters.

(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 $\tilde{x}$ Use the simulations in b) to simulate from the posterior distribution of $\tilde{x}$ [Hint: the regression curve is a quadratic. You can find a simple formula for ~x given β_0, β_1 *and* β_2

Since the regression curve is quadratic, so we take the first derivative of time for the formula

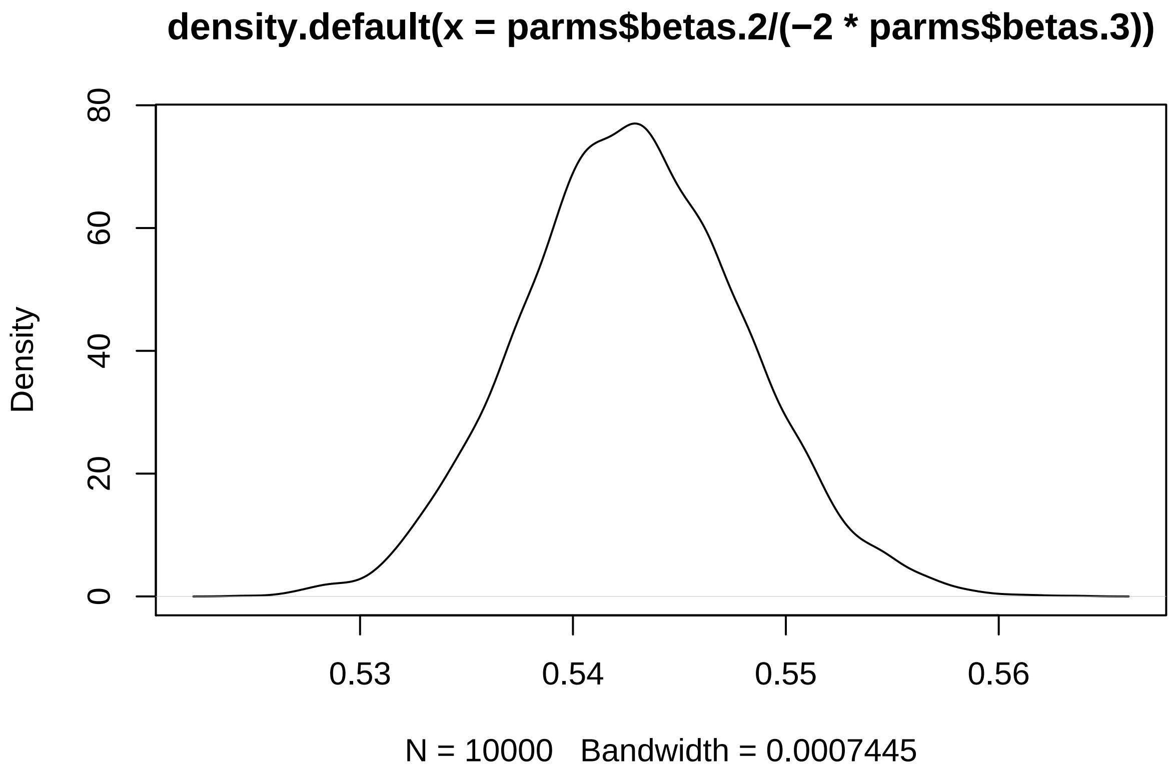
$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon \ N(0, \sigma^2)$$

$$0 = \beta_1 + 2\beta_2 \cdot time$$

$$time = -\frac{\beta_1}{2\beta_2}$$

In this case

$$\tilde{x} = \frac{\beta_1}{2\beta_2}$$



(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 overfitting. 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.]

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). Fit the logistic regression using maximum likelihood estimation by the command:glmModel <-glm(Work ~ 0 + ., data = WomenWork, family = binomial). Note how I added a zero in the model formula so that R doesn't add an extra intercept (we already have an intercept term from the Constant feature). Note also that a dot (.) in the model formula means to add all other variables in the dataset as features. family = binomial tells R that we want to fit a logistic regression.

Now the fun begins. Our goal is to approximate the posterior distribution of the 8-dim parameter vector β with a multivariate normal distribution

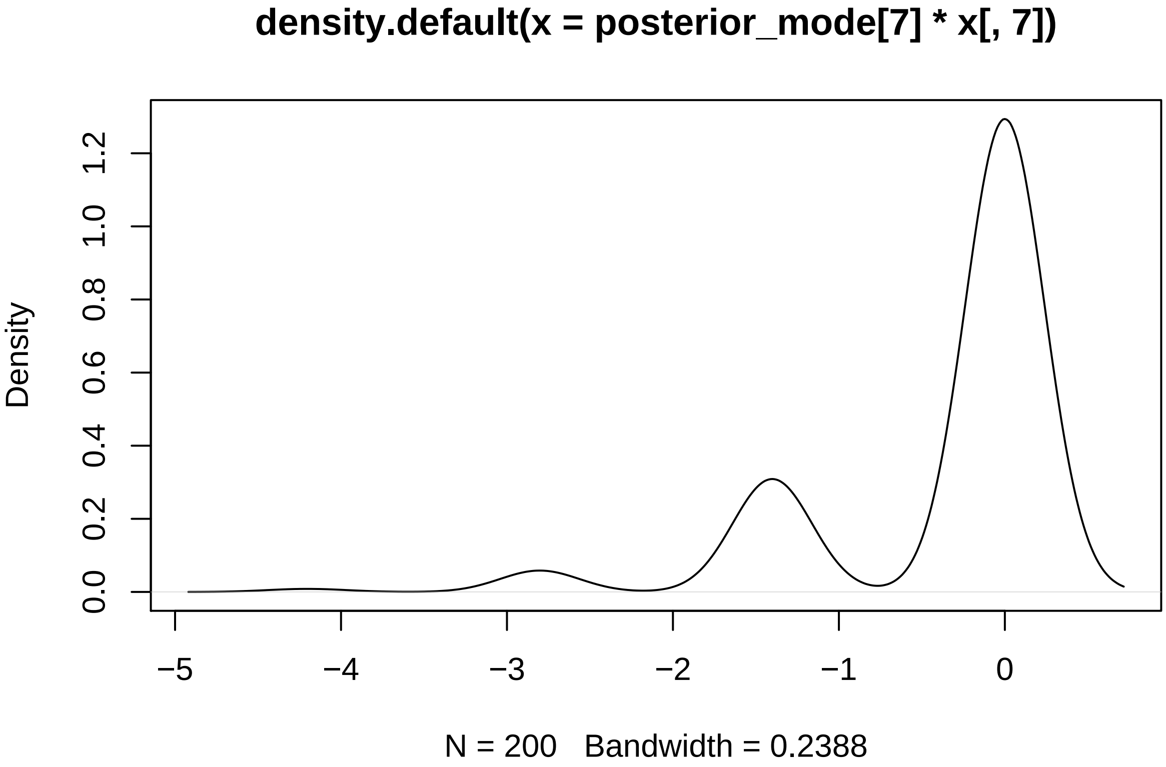
$$\beta|y, X\ N(\tilde{\beta}, J_y^{-1}(\tilde{\beta}))$$

where $\tilde{\beta}$ is the posterior mode and $J(\tilde{\beta}) = -\frac{\partial^2 \ln p(\beta|y)}{\partial\beta\partial\beta^T}|_{\beta=\tilde{\beta}}$ is the observed Hessian evaluated at the posterior mode. Note that $\frac{\partial^2 \ln p(\beta|y)}{\partial\beta\partial\beta^T}$ is an 8x8 matrix with second derivatives on the diagonal and cross-derivatives $\frac{\partial^2 \ln p(\beta|y)}{\partial\beta\partial\beta^T}$ 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. See my code https://github.com/mattiasvillani/BayesLearnCourse/raw/master/Code/

MainOptimizeSpam. zip here I have coded everything up for the spam prediction example (it also does probit regression, but that is not needed here). I want you to implement you own version of this. You can use my code as a template, but I want you to write your own file so that you understand every line of your code. Don't just copy my code. Use the prior $\beta\ N(0, \tau^2 I), with\ \tau = 10$. Your report should include your code as well as numerical values for $\tilde{\beta}\ and\ J_y^{-1}(\tilde{\beta})$ 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?

```
## Posterior mode, 0.8731703 -0.01956316 0.1749697 0.1669764 -0.1418006 -0.08581348 -1.401311 -0.035277
```

```
##
## Hessian Inverse,
```



```
##
## glm model coefficients: 0.6443036 -0.01977457 0.1798806 0.1675127 -0.1443595 -0.08234033 -1.362502 -
```

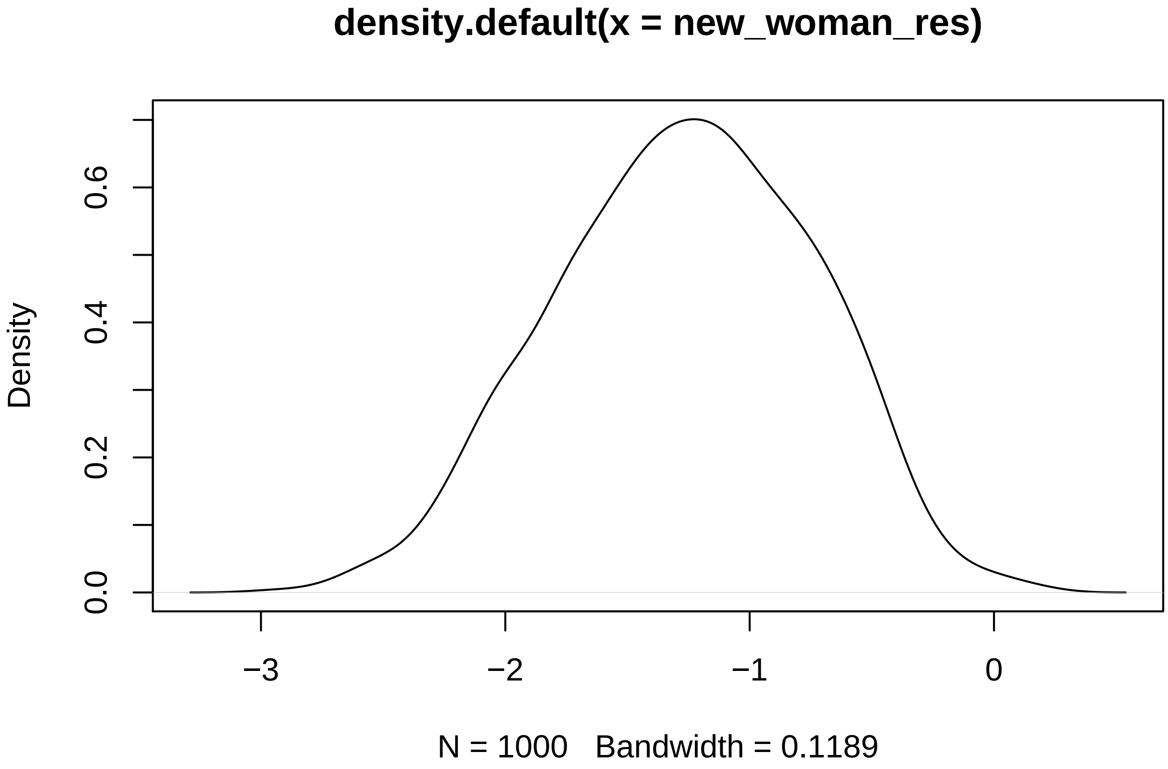
```
##
## my own model likelihood: -111.3782
```

```
##
## glm model likelihood: -111.3655
```

Judging by the value of posterior mode, the value of NSSmalChildren had a strong negetive impact on the result.

Comparing the likelihood for both parameters from posterior mode and glm model, we can find out that the two likelihood are really close, seeing that, we think the result is reasonable.

(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.]



(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?]

Appendix A : Code Question 1

```
data=read.table("TempLinkoping.txt",header = TRUE)
x=data.frame(x1=rep(1,length(data$time)),x2=data$time,x3=data$time^2)
x=as.matrix(x)
y=data$temp
mu_0=c(-10,100,-100)
Omega_0=0.01*diag(1,3,3)
v_0=4
sigma_sq_0=1

plot(data$temp)

for(i in 1:10){
  sigma_prior=rinvchisq(n=1, nu=v_0, tau = sigma_sq_0)
  beta_prior=rmvnorm(1,mean = mu_0,sigma = sigma_prior*solve(Omega_0))
  res=beta_prior[1,1]+beta_prior[1,2]*data$time+beta_prior[1,3]*data$time^2+rnorm(1,mean=0,sd=sqrt(sigma
  lines(res,col="red")
}
beta_hat=solve(t(x)%*%x)%*%t(x)%*%as.matrix(y)

mu_n=solve((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(data$temp)
sigma_sq_n=v_0*sigma_sq_0+(t(y)%*%y+t(mu_0)%*%Omega_0%*%mu_0-t(mu_n)%*%Omega_n%*%mu_n)
sigma_sq_n=sigma_sq_n/v_n
sigma_sq=rinvchisq(10000,v_n,sigma_sq_n)

# betas=rmvnorm(10000,mean=mu_n,sigma=sigma_sq[1]*solve(Omega_n))
# epsilon=matrix(rnorm(10000,mean=0,sd=sqrt(sigma_sq[1])))
parms=data.frame()
for(i in 1:10000){
  betas=rmvnorm(1,mean=mu_n,sigma=sigma_sq[i]*solve(Omega_n))
  epsilon=matrix(rnorm(1,mean=0,sd=sqrt(sigma_sq[i])))
  parms=rbind(parms,data.frame(betas=betas,epsilon=epsilon))
}
#parms=data.frame(betas,epsilon)
hist(parms$betas.1,breaks=100)
hist(parms$betas.2,breaks=100)
hist(parms$betas.3,breaks=100)
hist(sigma_sq,breaks=100)
fn_time=function(parms,time,sigma_value){
  df=data.frame()
  for(i in 1:length(time)){
    res=parms[,1]+parms[,2]*time[i]+parms[,3]*time[i]^2+parms[,4]
    median_temp=median(res)
    res=sort(res)
    lower_va=res[length(res)*0.025]
    upper_va=res[length(res)*0.975]
    df=rbind(df,data.frame(median=median_temp,lower=lower_va,upper=upper_va))
  }
  return(df)
}
```

```
res=fn_time(parms,data$time,sigma_sq)

library(ggplot2)
ggplot()+geom_point(aes(x=data$time,y=data$temp))+
  geom_line(aes(x=data$time,y=res$median,colour="Median"))+
  geom_line(aes(x=data$time,y=res$lower,colour="Lower"))+
  geom_line(aes(x=data$time,y=res$upper,colour="Upper"))+
  xlab("Time")+ylab("Temp")
plot(density(parms$betas.2/(-2*parms$betas.3)))
```

Appendix B : Code Question 2

```r
data=read.table("WomenWork.dat",header = TRUE)

y=data$Work
x=data[,-1]
x=as.matrix(x)
co_count=dim(x)[2]

log_likehood_logistic=function(y,x,betas){
  numerator=exp(x%*%betas)
  return(log(prod(numerator^y/(1+numerator))))
}


log_prior_likelihood=function(tau,betas){
  sigmas=diag(10,co_count,co_count)
  mu=rep(0,co_count)
  log_likelihood=(log(det(sigmas))+t(betas-mu)%*%solve(sigmas)%*%(betas-mu)+co_count*log(2*pi))*0.5
  return(log_likelihood)
}

# for log posterior= log prior +log likelihood
log_posterior=function(betas,y,x,tau){
  #return(log_likelihood_probit(y,x,betas)+log_prior_likelihood(tau,betas))
  return(log_likehood_logistic(y,x,betas)+log_prior_likelihood(tau,betas))
}

starting_value=rep(0,co_count)
tau=10
op=optim(starting_value,log_posterior,gr=NULL,y,x,tau,method=c("BFGS"),control=list(fnscale=-1),hessian
posterior_mode=op$par
posterior_cov=-solve(op$hessian)

cat("Posterior mode,",posterior_mode)
cat("\nHessian Inverse,")
#::kable(posterior_cov)
plot(density(posterior_mode[7]*x[,7]))
glm_model=glm(Work ~ 0 + ., data = data, family = binomial)
cat("\nglm model coefficients:",glm_model$coefficients)

cat("\nmy own model likelihood:",log_likehood_logistic(y,x,posterior_mode))
cat("\nglm model likelihood:",log_likehood_logistic(y,x,glm_model$coefficients))
new_woman=c(1,10,8,10,1,40,1,1)
posterior_draws=rmvnorm(1000,mean=posterior_mode,sigma = posterior_cov)
new_woman_res=posterior_draws%*%new_woman

plot(density(new_woman_res))
```