Computer Lab 2
732A54 - Bayesian Learning

Jooyoung Lee - `joole336`
Zuxiang Li - `zuxli371`

April 24, 2020

1. Linear and polynomial regression

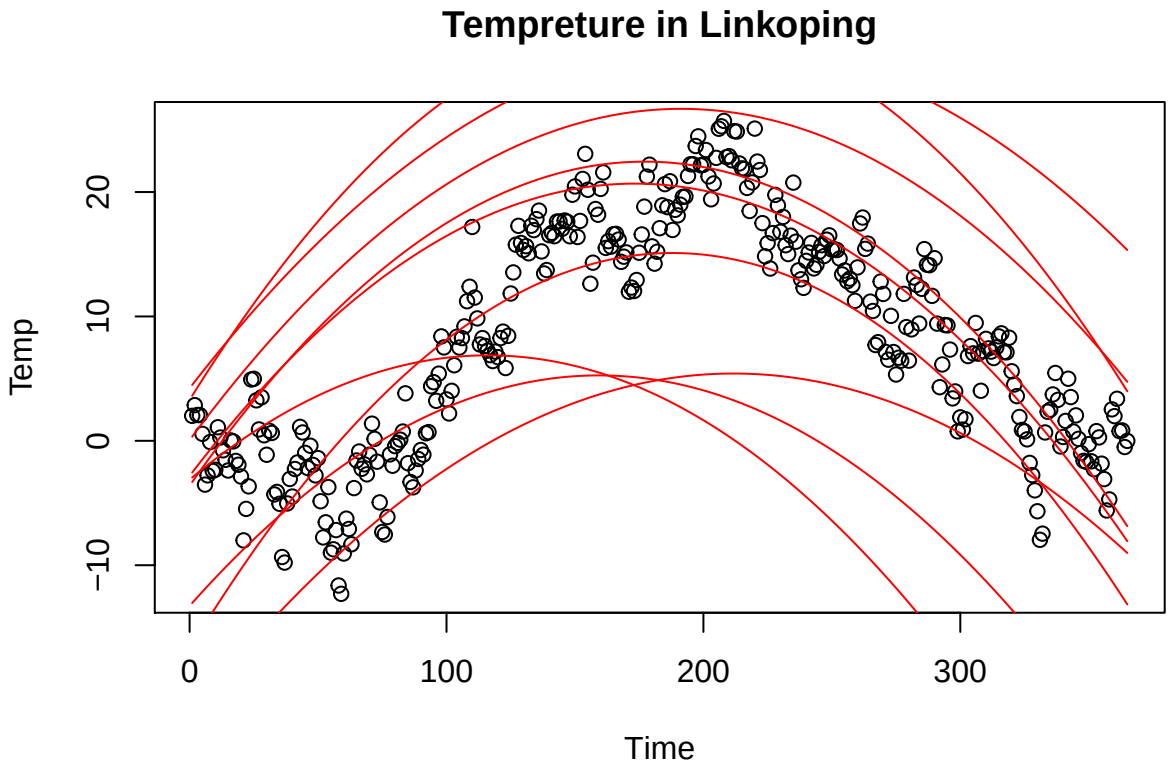
The dataset TempLinkoping.txt contains daily temperatures (in Celcius degrees) at Malmslätt, Linköping over the course of the year 2016 (366 days since 2016 was a leap year). The response variable is temp and the covariate is

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

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

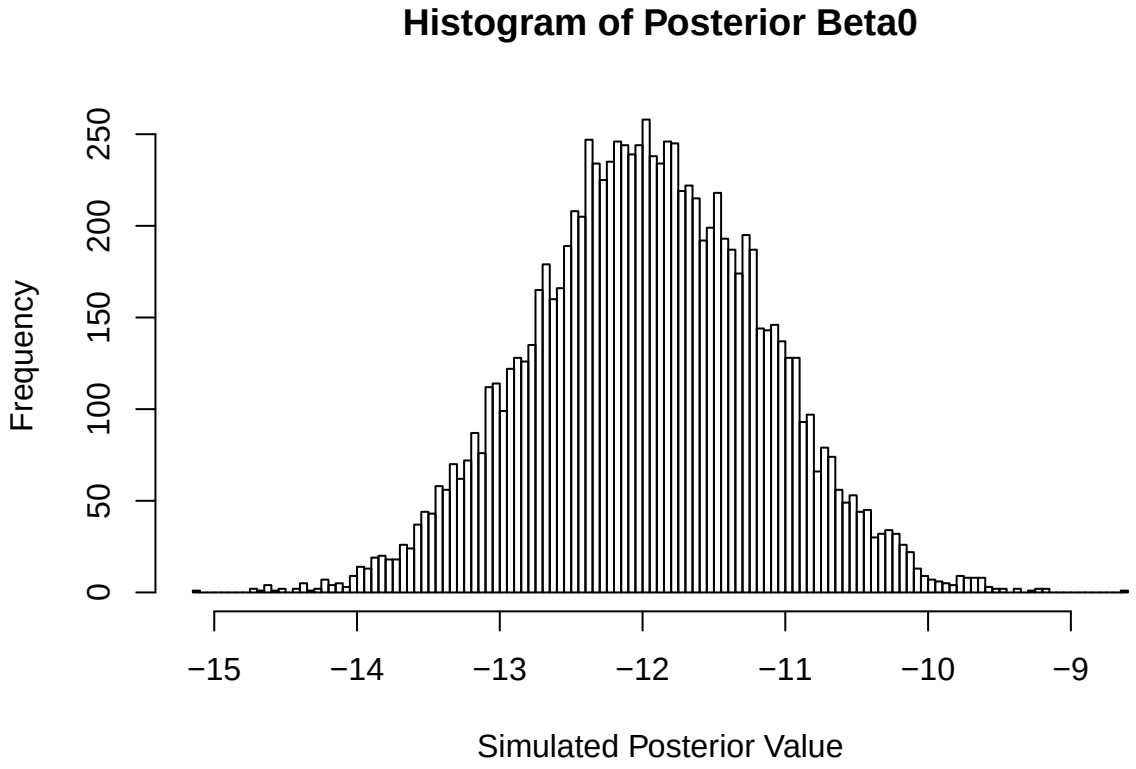
$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon \sim N(0, \sigma^2)$$

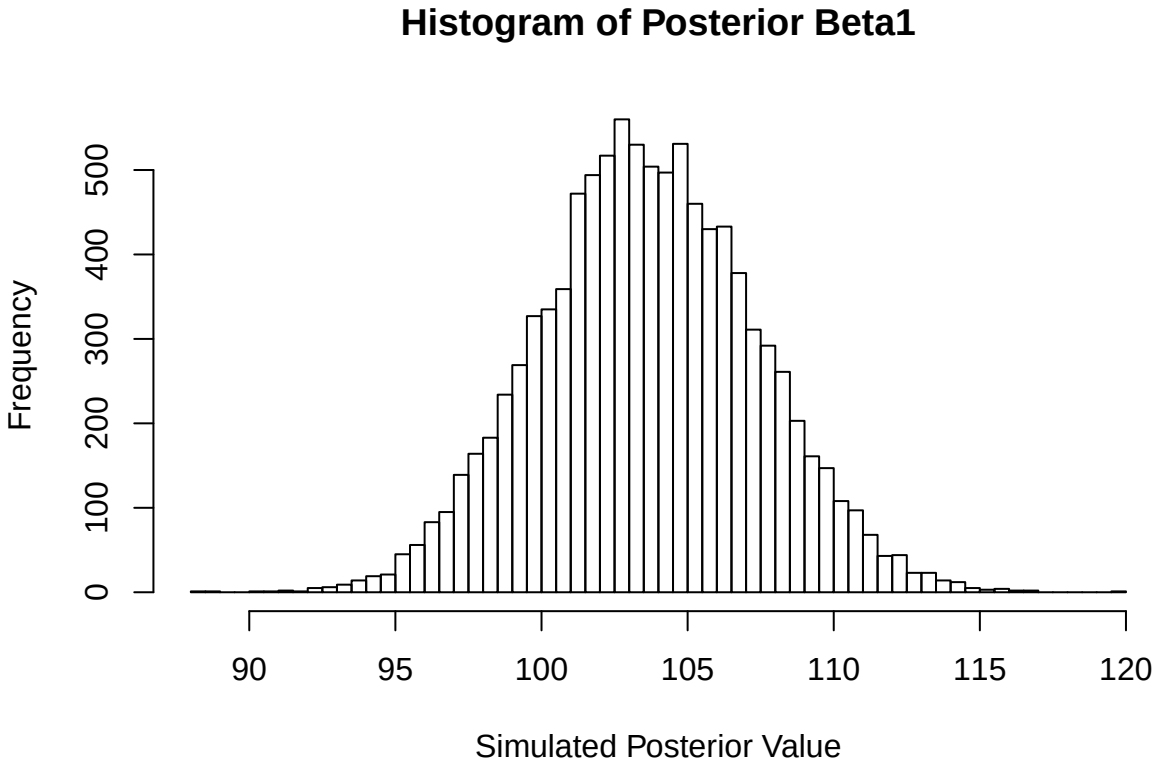
(a) Determining the prior distribution of the model parameters. Use the conjugate prior for the linear regression model. Your task is to set the prior hyperparameters μ_0, Ω_0, v_0 *and* σ^2 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3, v_0 = 4$ *and* $\sigma^2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

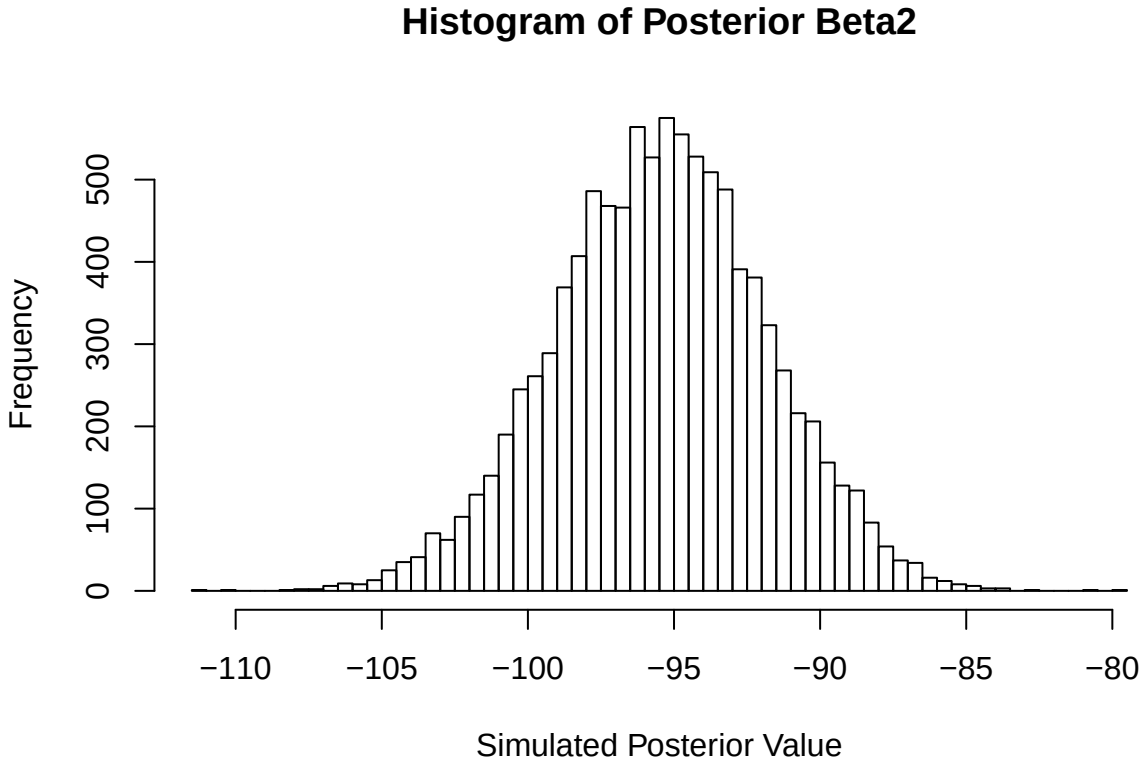


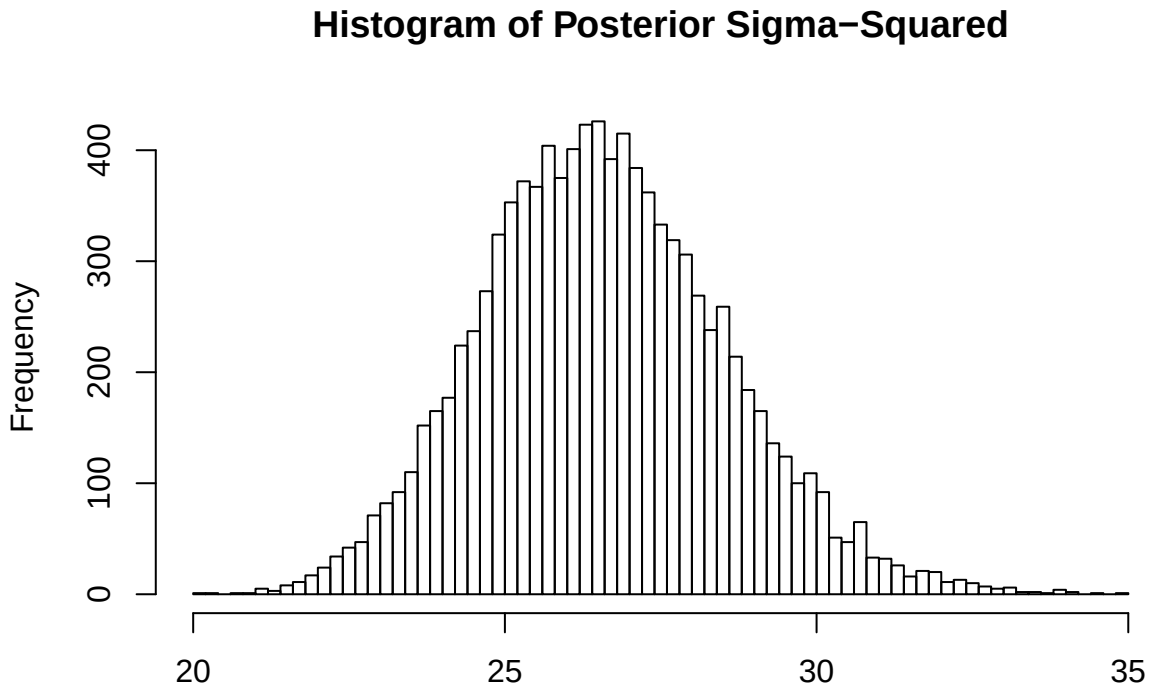
Red lines in the plot represent regression curves from prior, these curves varies a lot, but still they follows the pattern of the given data. Thus the given hyperparameters could be utilized to produce prior distribution and we can say the curves look reasonable.

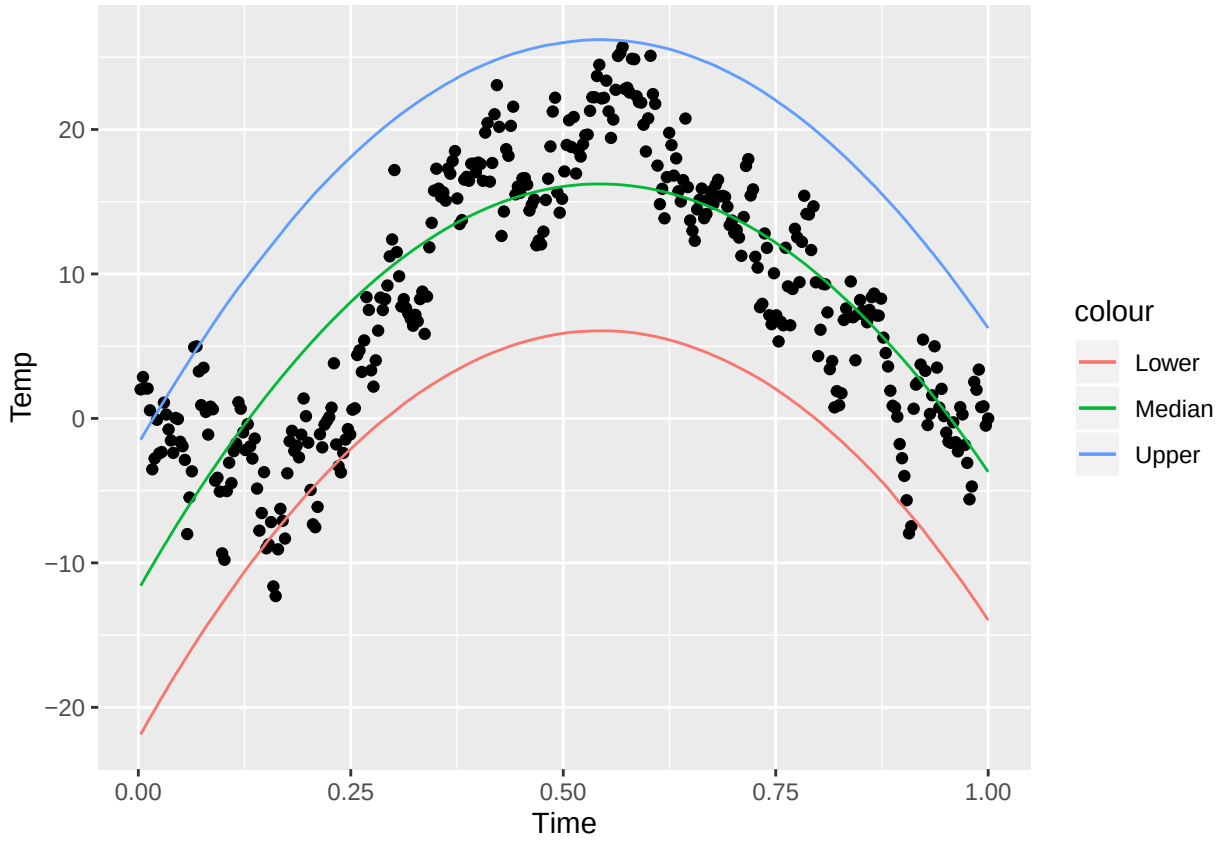
(b) Write a program that simulates from the joint posterior distribution of $\beta_0, \beta_1, \beta_2$ and σ^2. Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of the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




Histogram of Posterior Beta1
Frequency
0
100
200
300
400
500
90
95
100
105
110
115
120
Simulated Posterior Value


Histogram of Posterior Beta2
Frequency
0
100
200
300
400
500
−110
−105
−100
−95
−90
−85
−80
Simulated Posterior Value





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

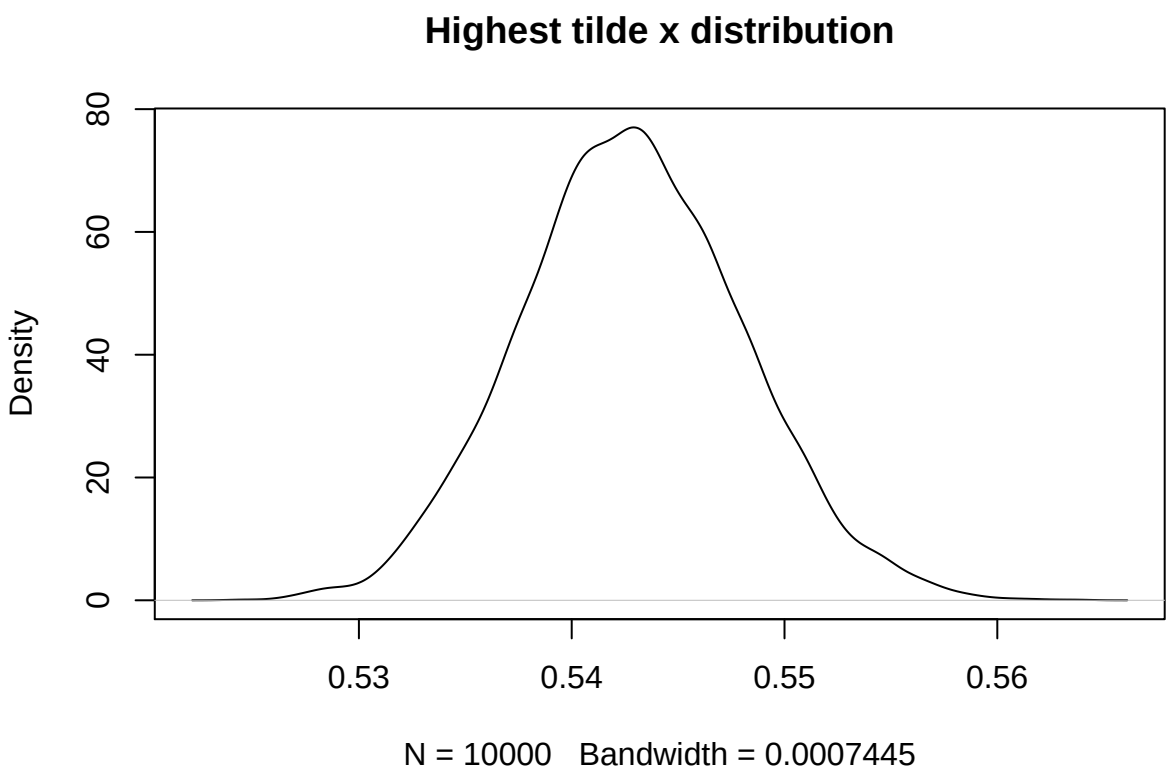
$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon \sim N(0, \sigma^2)$$

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

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

In this case

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



Based on our intuition, the highest tempreture in a year should be around July or August. We take the first derivative of time and then plot the distribution, the highest result is around 0.542 which indicate the result meet our intuition. 0.542 is equivelant to 197.83th day of the year.

(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.]

One solution for mitigating potential problem is to set different hyperprior for prior distribution of betas. Set $\mu = 0$; set Ω_0 to a matrix that its inverse matrix's diagonal value would be very small so that covariance matrix of the prior distribution of betas would consist small diagonal values make prior have stronger impact

on posterior. When prior distribution is certain, likelihood based on data has smaller impact on deriving posterior distribution. In above case, prior belief that betas will be 0s are very strong, it is unlikely the resulting posterior distribution of beta will consist high values for all elements. For example, we can choose laplace prior $\beta_i|\sigma^2 \sim Laplace(0, \frac{\sigma^2}{\lambda})$. For laplace prior, many β_i close to zero but some of the β_i could have really large values, which meet our needs.

2. Posterior approximation for classification with logistic regression

(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

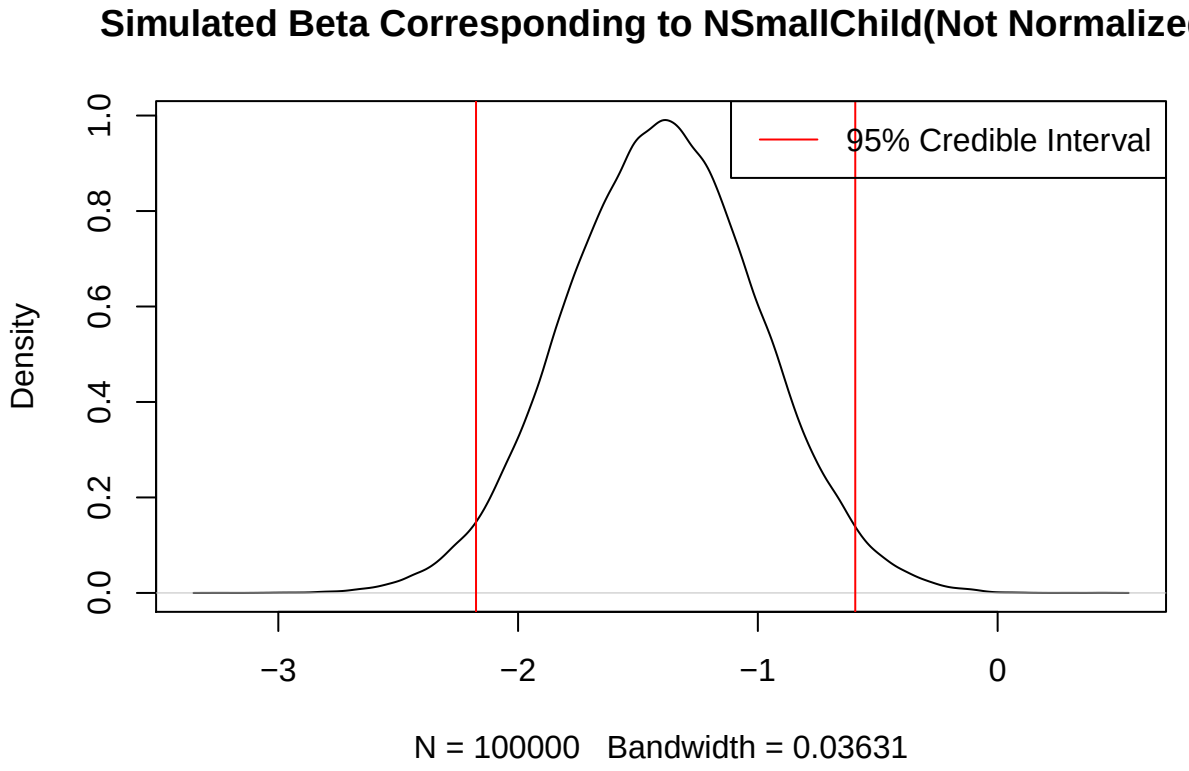
$$\beta|y, X \sim N(\tilde{\beta}, J_y^{-1}(\tilde{\beta}))$$

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 $\beta \sim N(0, \tau^2 I)$, 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?

	x
Constant	0.8731703
HusbandInc	-0.0195632
EducYears	0.1749697
ExpYears	0.1669764
ExpYears2	-0.1418006
Age	-0.0858135
NSmallChild	-1.4013109
NBigChild	-0.0352770

	Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
Constant	3.0567833	0.0044124	-0.0870233	-0.0153267	0.0611132	-0.0412161	-0.2647012	-0.1328261
HusbandInc	0.0044124	0.0002550	-0.0005938	-0.0000382	0.0001708	-0.0000500	0.0004239	-0.0001905
EducYears	-0.0870233	-0.0005938	0.0068466	-0.0002481	0.0014361	0.0002857	-0.0043871	0.0026873

	Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
ExpYears	-0.0153267	-0.0000382	-0.0002481	0.0044023	-0.0144307	-0.0000902	-0.0012315	0.0006886
ExpYears2	0.0611132	0.0001708	0.0014361	-0.0144307	0.0562520	-0.0005257	0.0020779	-0.0001122
Age	-0.0412161	-0.0000500	0.0002857	-0.0000902	-0.0005257	0.0008717	0.0063100	0.0015777
NSmallChild	-0.2647012	0.0004239	-0.0043871	-0.0012315	0.0020779	0.0063100	0.1621725	0.0102225
NBigChild	-0.1328261	-0.0001905	0.0026873	0.0006886	-0.0001122	0.0015777	0.0102225	0.0215963




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

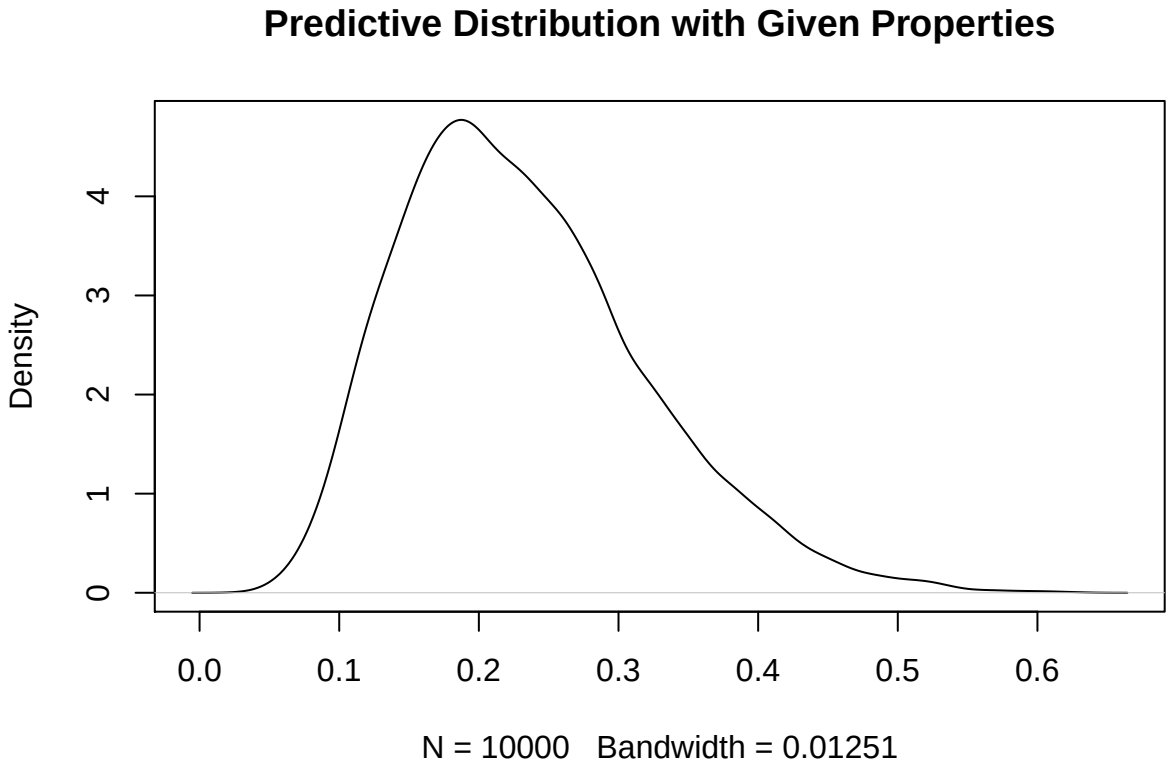
```
##
## my own model likelihood: -111.3782
```

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

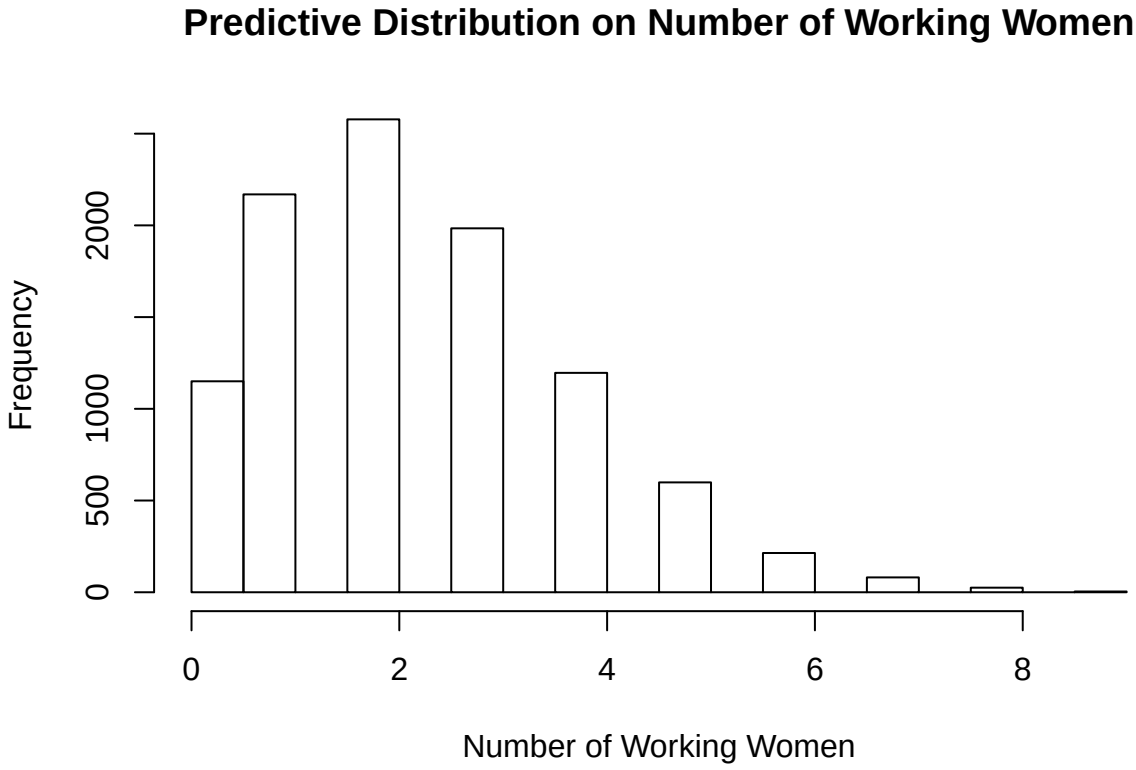
To verify that our result is reasonble, first we have a look at the likelihood of our model and glm model. Both of them are really close. Also, from the plot, we can observed that 95% credible interval does not contain 0, indicating this feature is an important determinant. Using R built-in model glm to anaylize the given data, it is shown that this feature has very small p-value - which means in frequentist approach, it is very much unlikely that NSmallChild has no effect on the response variable.

(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.]

Generate Bayesian prediction with simulation by generating posterior draws first then the predictive draws.



(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?]


Predictive Distribution on Number of Working Women
Frequency
0
500
1000
2000
0
2
4
6
8
Number of Working Women

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,xlab="Time",ylab="Temp",main = "Tempreture in Linkoping")

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))
}
hist(parms$betas.1,breaks=100,main = "Histogram of Posterior Beta0", xlab="Simulated Posterior Value")
hist(parms$betas.2,breaks=100,main = "Histogram of Posterior Beta1", xlab="Simulated Posterior Value")
hist(parms$betas.3,breaks=100,main = "Histogram of Posterior Beta2", xlab="Simulated Posterior Value")
hist(sigma_sq,breaks=100,main = "Histogram of Posterior Sigma-Squared", xlab="Simulated Posterior Value
fn_time=function(parms,time,sigma_value){
  df=data.frame()
  for(i in 1:length(time)){
    res=parms[,1]+parms[,2]*time[i]+parms[,3]*time[i]^2+parms[,4]
    median_temp=median(res)
    res=sort(res)
    lower_va=res[length(res)*0.025]
    upper_va=res[length(res)*0.975]
    df=rbind(df,data.frame(median=median_temp,lower=lower_va,upper=upper_va))
  }
  return(df)
}
res=fn_time(parms,data$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)), main="Highest tilde x distribution")
```

Appendix B : Code Question 2

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

y=data$Work
x=data[,-1]
covariates = names(x)
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)


names(posterior_mode) = covariates
colnames(posterior_cov) = covariates
rownames(posterior_cov) = covariates
#cat("Posterior mode\n")
#print(posterior_mode)
kable(posterior_mode)
#cat("\nHessian Inverse\n")
#print(posterior_cov)
kable(posterior_cov)
### 95% credible interval of NSmallChild
set.seed(12345)
simulated_betas = rmvnorm(100000, mean = posterior_mode, sigma = posterior_cov)
simulated_NSmallChild = simulated_betas[,7]
plot(density(simulated_NSmallChild), main="Simulated Beta Corresponding to NSmallChild(Not Normalized)"
mode_NSC = density(simulated_NSmallChild)$x[which.max(density(simulated_NSmallChild)$y)]
#### since qnorm retunrs value based on standard normal,
lim1 = qnorm(0.975, mean = mode_NSC, sd = sd(simulated_NSmallChild))
lim2 = qnorm(0.975, mean = mode_NSC, sd = sd(simulated_NSmallChild), lower.tail = FALSE)
```

```r
abline(v = lim1, col = "red")
abline(v = lim2, col= "red")
legend("topright", legend = "95% Credible Interval", lty = 1, col="red")
#### 95% credible interval does not involve 0 -- this feature is an important determinant.

glmModel <- glm(Work~ 0 + ., data = data, family = binomial)
summary(glmModel)

cat("\nmy own model likelihood:",log_likehood_logistic(y,x,posterior_mode))
cat("\nglm model likelihood:",log_likehood_logistic(y,x,glmModel$coefficients))
#### it is an important determinant of the probability. -- very small p-value in frequentist approach,


new_woman=c(1,10,8,10,1,40,1,1)
posterior_draws=rmvnorm(10000,mean=posterior_mode,sigma = posterior_cov)
new_woman_res=posterior_draws%*%new_woman
prob=exp(new_woman_res)/(1+exp(new_woman_res))
plot(density(prob), main = "Predictive Distribution with Given Properties")
work_predict = function(n_women, probs){
"  num_work = c()
  for (i in 1:length(probs)){
    num_work = append(num_work, rbinom(1, size = n_women, prob = probs[i]))
  }"
  return(rbinom(length(probs), n_women, prob = probs))
}

prediction_on_nums = work_predict(10, prob)
hist(prediction_on_nums, main = "Predictive Distribution on Number of Working Women", xlab = "Number of
```