# Bayesian_lab2

Akshath Srinivas (akssr921),Yaning Wang (yanwa579)

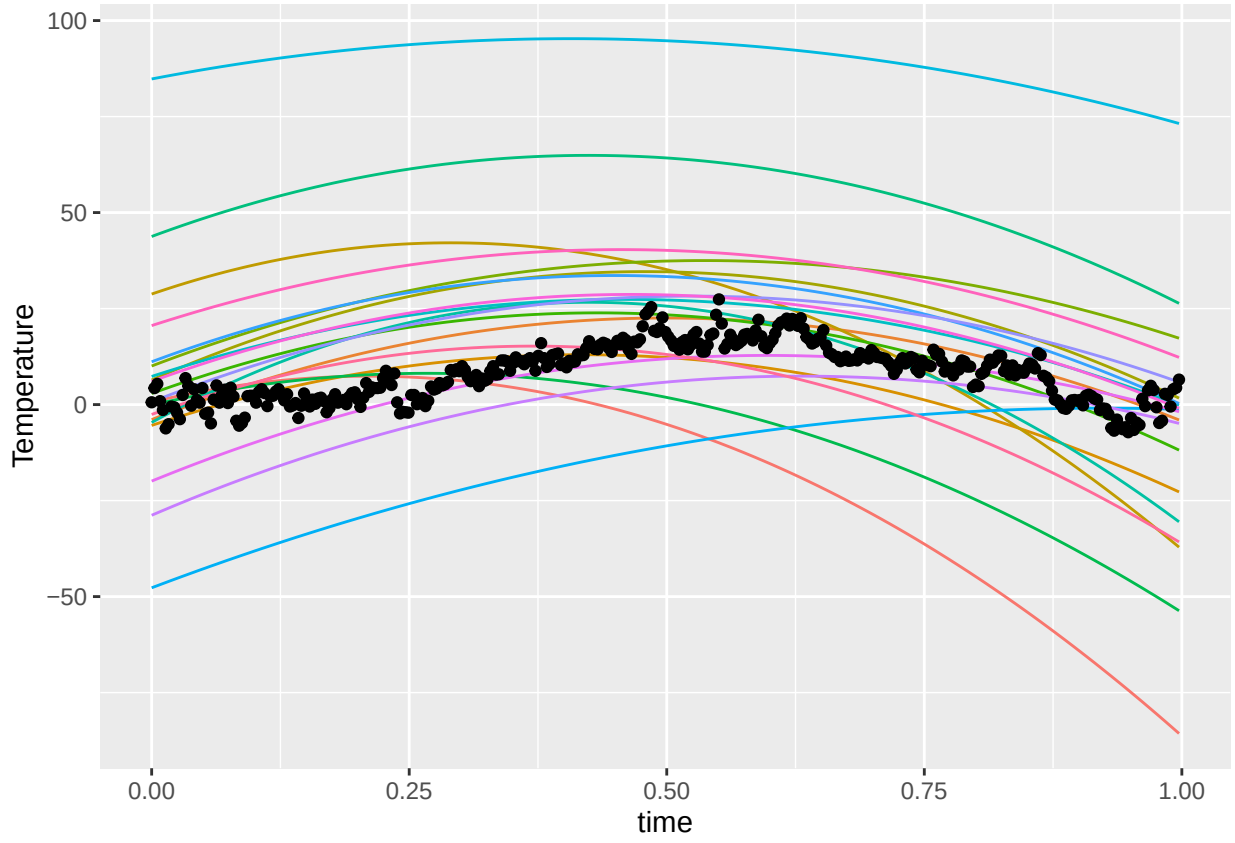
2023-04-21

## Question 1

### (a)

```
#get data and create covariate(time) and y(temp)
data<-readxl::read_xlsx('Linkoping2022.xlsx')
temp<-as.data.frame(data[,3])
temp<-temp$temp
time<-as.numeric(as.Date(data$datetime) - as.Date("2022-01-01"))/365

#creating X matrix
#x value for intercept
X_mat <- matrix(rep(1,length(time)))
#cbing time and time^2
X_mat<-cbind(X_mat,time,time^2)

#sigma(tau) and mu for sigma_2 simulation
y<-time
mu<-1
tau<-1

#mu_0 and omega_0 to get beta values
mu_0<-c(0, 100, -100)
omega_0<-solve(0.01 * diag(3))

#draw joint prior using inversechisquare to draw sigma_2 and using
#sigma_2,mu_0 and omega_0 draw for beta values
#creating a function to change the values of the parameters
joint_prior_func<-function(n,x,y,mu,tau,mu_0,omega_0){
  joint_prior<-matrix(NA, ncol = 3, nrow = n)
  for(i in 1:n){
    sigma_2<-LaplacesDemon::rinvchisq(1,mu,tau)
    joint_prior_vec <- mvtnorm::rmvnorm(n = 1, mean = mu_0, sigma = omega_0*sigma_2)
    joint_prior[i,]<-joint_prior_vec
  }
  #get predictions
  pred<-X_mat%*%t(joint_prior)
  return(pred)
}
#get prediction
prediction<-joint_prior_func(50,X_mat,y,mu,tau,mu_0,omega_0)
#get_first 20 predictions to plot
```

```r
prediction_20<-prediction[,1:20]

#plotting using ggplot2
library(tidyr)
df_plot<-as.data.frame(cbind(prediction_20,time,temp))
df_long <- pivot_longer(df_plot, cols = 1:20, names_to = "Variable", values_to = "temp_Value")
library(ggplot2)
p<-ggplot(df_long) +
  geom_line(aes(x = time, y = temp_Value, color = Variable))+
  geom_point(data=df_plot,aes(x=time,y=temp))
p+labs(y = "Temperature")+theme(legend.position = "none")
```

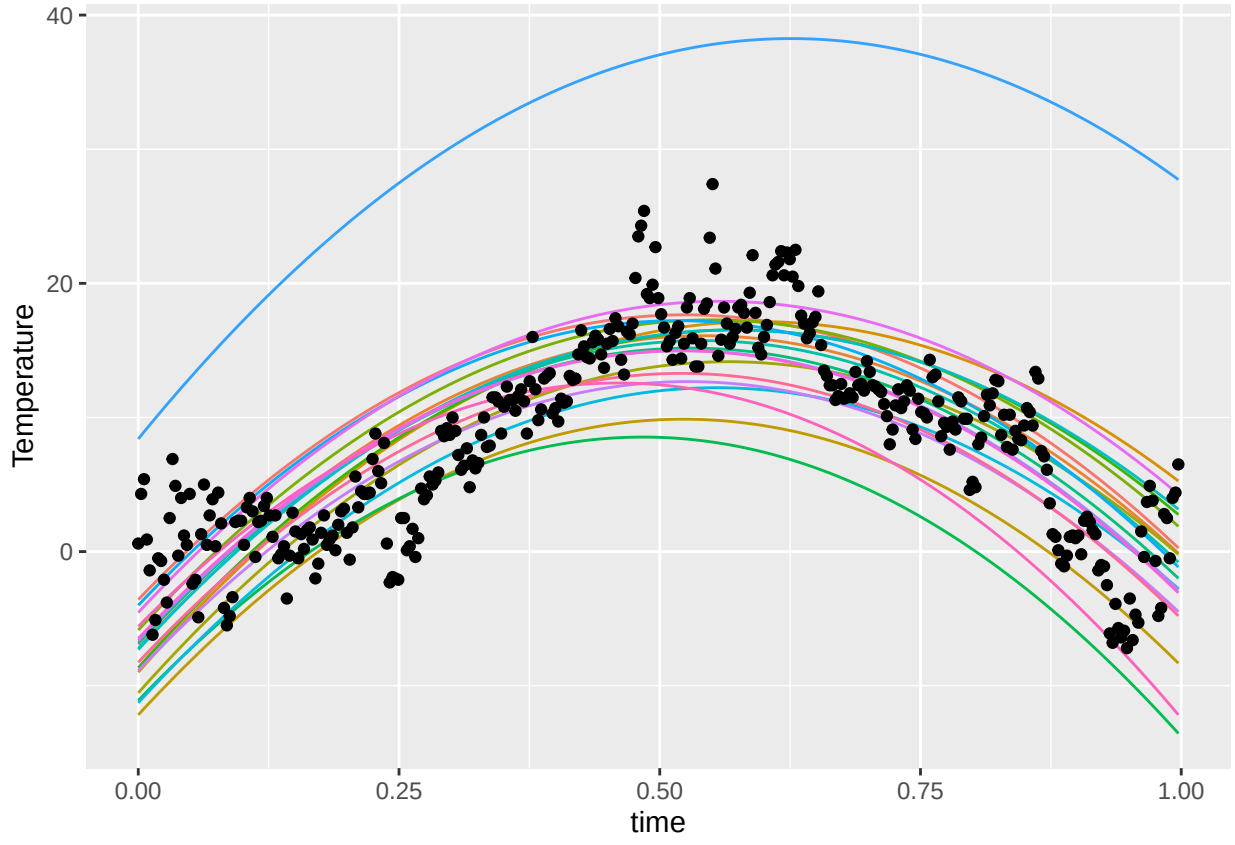



From the regression curves above, we can conclude that curves do not match or traverse the data points(some curves are all over the place). So, the parameters we selected should be altered to get the curves to look reasonable.

we changed the $\mu_0$ to $(-7.304061, 83.149904, -78.298468)$ by calculating OLS estimate, $\Omega_0$ to 3.5 which controls whether the curve will be narrower or wider (larger value causes narrower curve). We changed $\nu_0$ and $\sigma_0$ to 3 and 12 respectively after trying with different values, if sigma value is less the curve will not cover some top and bottom values. Below is the plot after changing the parameter values, we can see that regression curves looks more reasonable now.

```r
#changing the parameters to get f prior regression curves agrees with your prior beliefs
#about the regression curve
#from lm set mu_0 values
beta_lm<-lm(temp~time+I(time^2),data)
coef<-beta_lm$coefficients
```

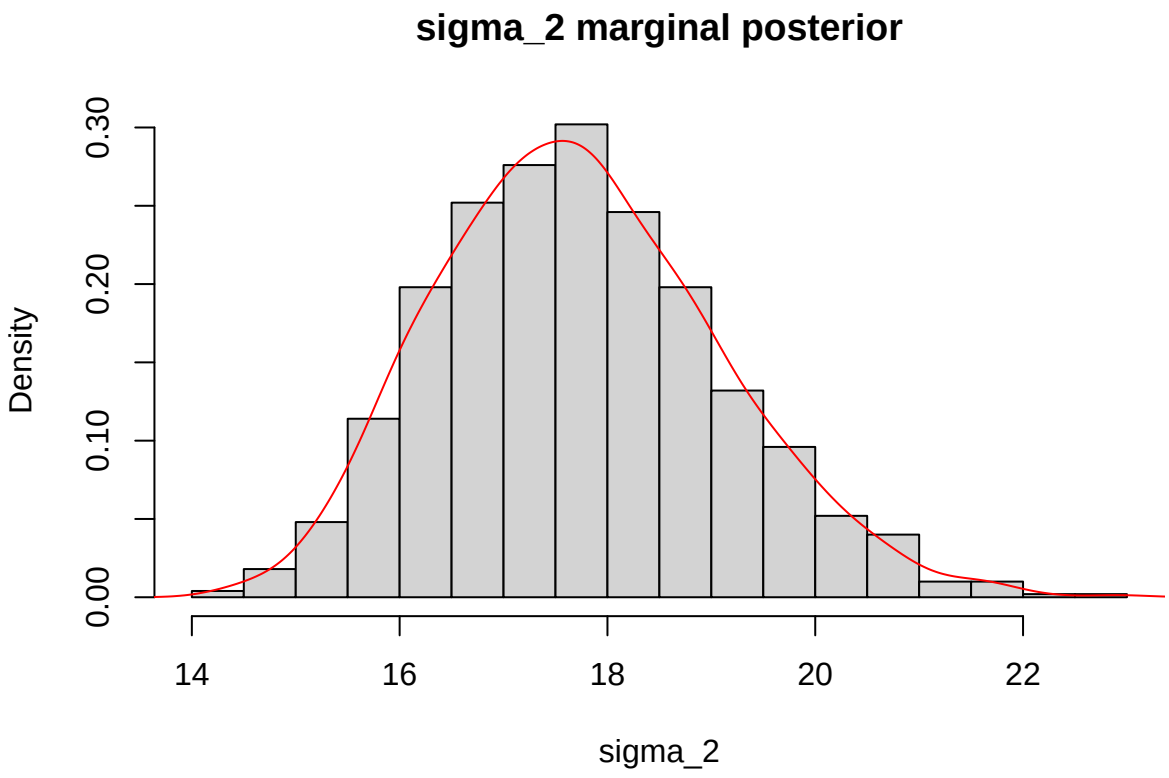
```r
mu_0_opt<-c(-7.304061 ,83.149904,-78.298468)

#setting the value by seeing the plot, larger the value narrower the curve
omega_0_opt<-solve(3.5 * diag(3))
mu_opt<-3
tau_opt<-12

#setting mu and tau
prediction_opt<-joint_prior_func(50,X_mat,y,mu_opt,tau_opt,mu_0_opt,omega_0_opt)
prediction_opt_20<-prediction_opt[,1:20]

#plotting using ggplot2
library(tidyr)
df_plot_opt<-as.data.frame(cbind(prediction_opt_20,time,temp))
df_long_opt <- pivot_longer(df_plot_opt, cols = 1:20, names_to = "Variable", values_to = "temp_Value")
library(ggplot2)
p_opt<-ggplot(df_long_opt) +
  geom_line(aes(x = time, y = temp_Value, color = Variable))+
  geom_point(data=df_plot_opt,aes(x=time,y=temp))
p_opt+labs(y = "Temperature")+theme(legend.position = "none")
```



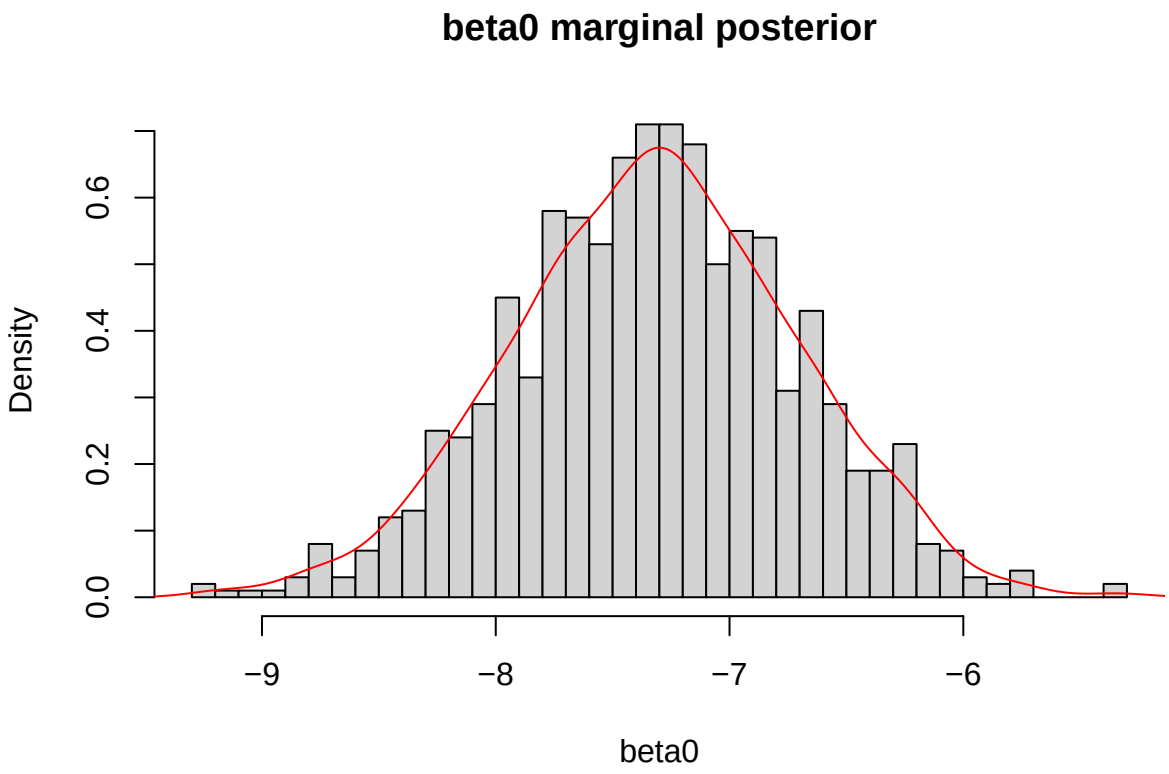

## (b) (i)

```r
#calculating all the posterior parameters
beta_hat<-solve(t(X_mat)%*%X_mat)%*%t(X_mat)%*%temp
```

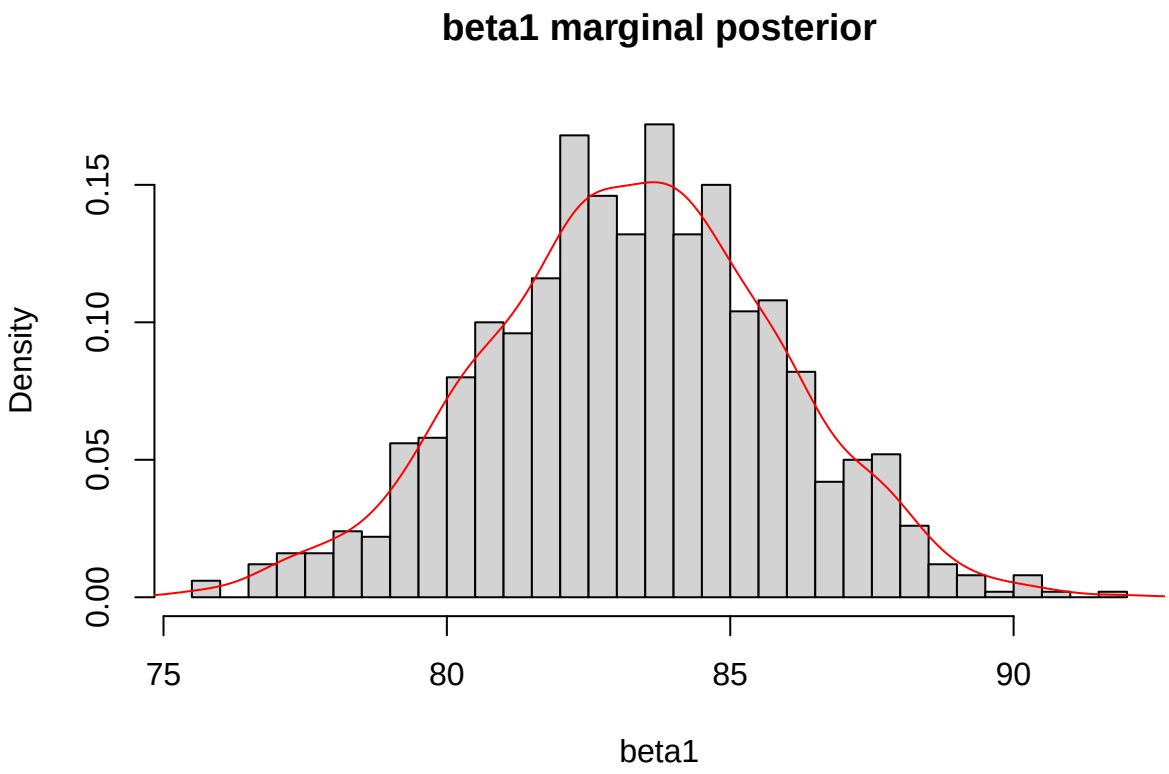
```r
rownames(beta_hat)<-NULL

mun<-solve(t(X_mat)%*%X_mat+omega_0_opt)%*%(t(X_mat)%*%X_mat%*%beta_hat+omega_0_opt%*%mu_0_opt)
rownames(mun)<-NULL

omega_n<-t(X_mat) %*% X_mat+omega_0_opt
rownames(omega_n)<-NULL
colnames(omega_n)<-NULL

n<-nrow(X_mat)
v_n <- mu_opt+n

sigma_n<-(mu_opt*tau_opt+(t(temp)%*%temp+t(mu_0_opt)%*%omega_0_opt%*%mu_0_opt-t(mun)%*%omega_n%*% mun))


#draw for betas and sigma_2
post_betas<-matrix(NA, ncol = 3, nrow = 1000)
draw_invchi_vec<-rep(NA,1000)
for(i in 1:1000){
  #draw for sigma_2
  draw_invchi<-LaplacesDemon::rinvchisq(1,v_n,sigma_n)
  draw_invchi_vec[i]<-draw_invchi
  joint_prior_vec <- mvtnorm::rmvnorm(n = 1,mean=mun, sigma  =draw_invchi*solve(omega_n) )
  post_betas[i,]<-joint_prior_vec
}
#histograms of betas and sigma_2
hist(draw_invchi_vec,breaks=30,probability = TRUE,main ='sigma_2 marginal posterior',xlab='sigma_2')
lines(density(draw_invchi_vec),col='red')
```



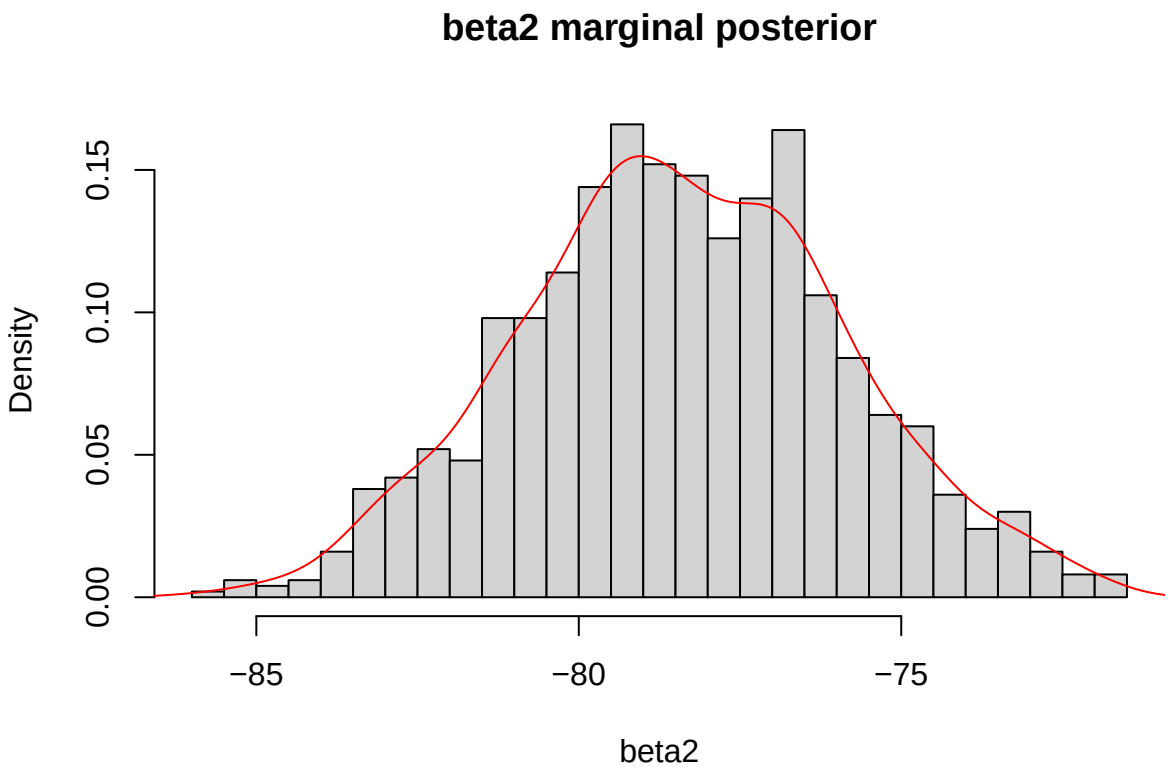

```
hist(post_betas[,1],breaks=30,probability = TRUE,main ='beta0 marginal posterior',xlab='beta0')
lines(density(post_betas[,1]),col='red')
```




```r
hist(post_betas[,2],breaks=30,probability = TRUE,main ='beta1 marginal posterior',xlab='beta1')
lines(density(post_betas[,2]),col='red')
```

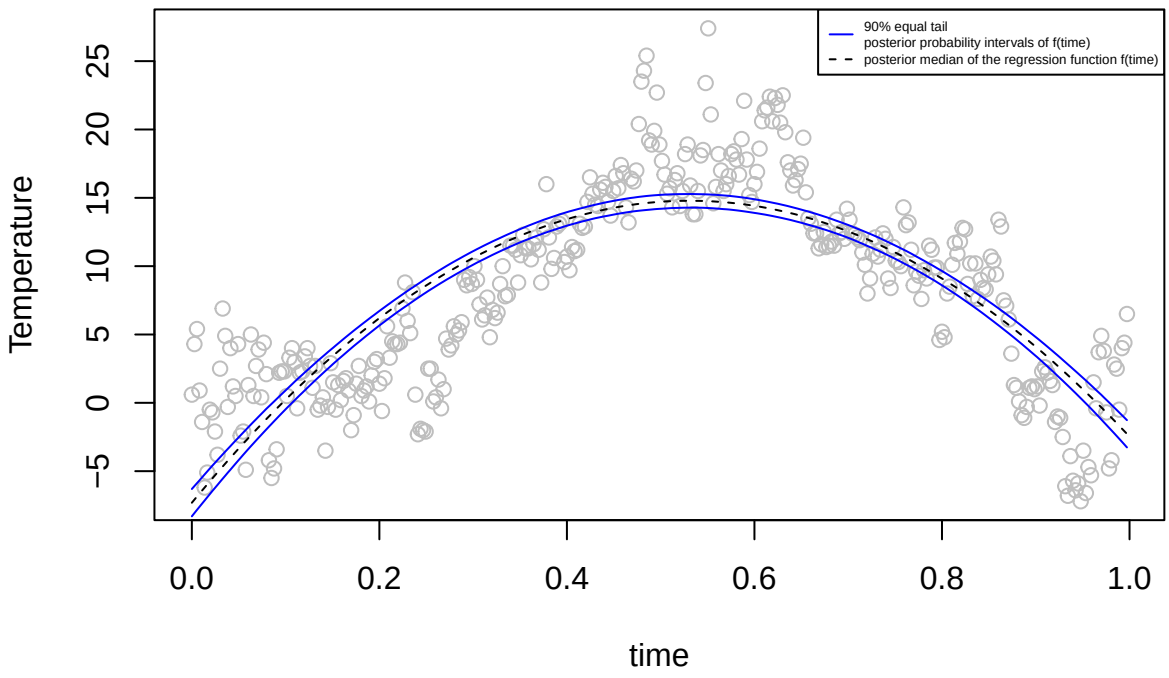



```
hist(post_betas[,3],breaks=30,probability = TRUE,main ='beta2 marginal posterior',xlab='beta2')
lines(density(post_betas[,3]),col='red')
```




(b) (ii)

```
#predictions for time as columns
pre_time <- post_betas %*% t(X_mat)
get_medians<-apply(pre_time, 2, median)
#quantiles
quant<-matrix(NA, ncol = 2, nrow = length(time))
for(i in 1:length(time)){
  quant[i,]<-quantile(pre_time[,i], probs = c(0.05,0.95))
}
quant_lower<-quant[,1]
quant_upper<-quant[,2]

#plotting the temp and time
plot(time,temp,col='gray',ylab='Temperature')
lines(time,get_medians,lty = "dashed")
lines(time, quant_lower, col = 'blue')
lines(time, quant_upper, col = 'blue')
legend("topright",legend=c("90% equal tail
posterior probability intervals of f(time)",'posterior median of the regression function f(time)'),col=
       lty = c(1, 2), cex=0.4)
```




We can see the curves for posterior median of the regression function f(time) and 90% equal tail posterior probability intervals of f(time) from the plot above. The posterior probability intervals does contain most of the data points. We should not expect this interval will capture most data points, as there are noise and uncertainity in the data points. We should have reasonable $\epsilon$ to observe the data points within interval.

## (c)

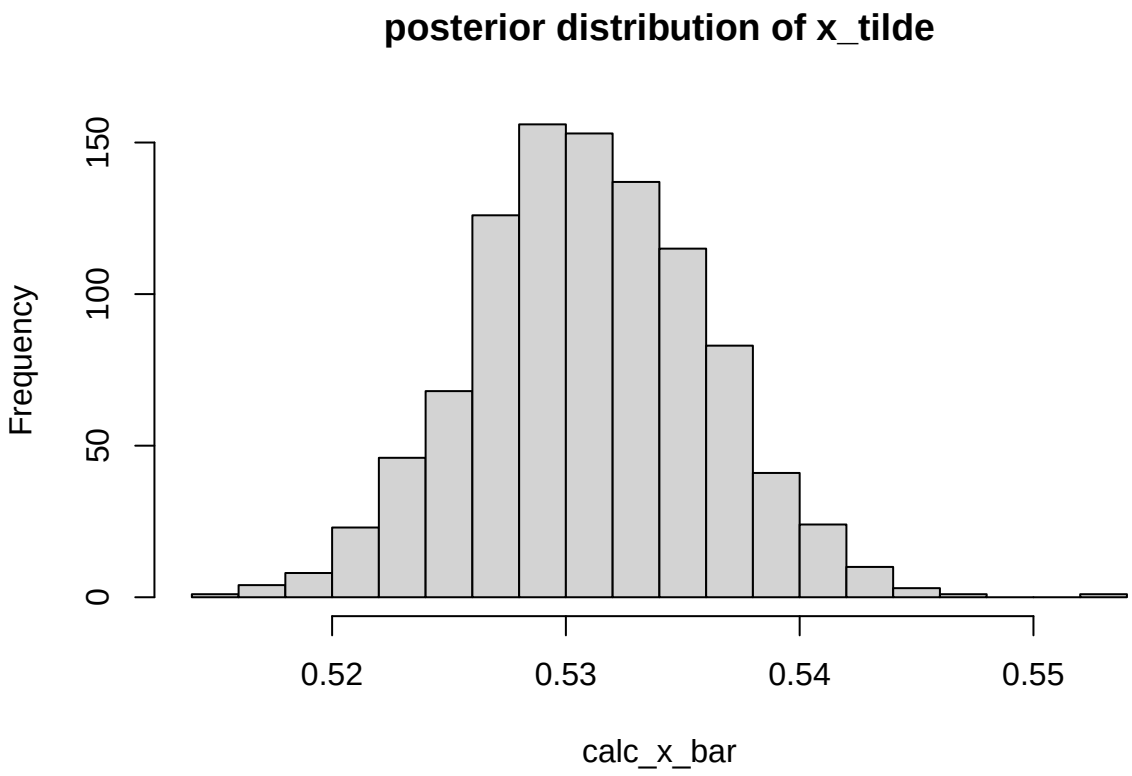
To locate the time with the highest expected temperature (i.e. the time where f(time) is maximal), we should differentiate below equation and equate it to zero to get the $\widetilde{x}$. Finally, we substituted the beta values to the resulted equation and plotted the histogram which can be seen below.

$$f(time) = temp = \beta_0 + \beta_1.time + \beta_2.time^2$$

$$\frac{dtemp}{dtime} = \beta_1 + 2.\beta_2.time = 0$$

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

```
#finding x_bar
#time with the highest expected temperature (-beta1)/(2*beta2)
calc_x_bar=(-post_betas[,2])/(2*post_betas[,3])
hist(calc_x_bar,breaks=25,main='posterior distribution of x_tilde')
```




### (d)

From the slides, we suggest Suitable prior that mitigates this potential problem might be gaussian regularization prior.

$$\beta \mid \sigma^2 \overset{iid}{\sim} \mathcal{N}\left(\mu_0, \frac{\sigma^2}{\lambda}\right)$$

Larger $\lambda$ gives smoother fit. More shrinkage. Note:$\Omega_0 = \lambda.I$

## Question 2

### (a)

```
# Load data set
WomenAtWork <- read.table("WomenAtWork.dat", header = TRUE)
#Select features and response
x<-as.matrix(WomenAtWork[2:8])
y<-as.matrix(WomenAtWork[1])
# get the number of observations and features
n<-dim(WomenAtWork)[1]
n_fea<-dim(WomenAtWork)[2]-1
# scaling factor for the prior of beta
t <- 2
# Setting up the prior mean vector and covariance matrix
```

```r
mu <- as.matrix(rep(0,n_fea))
sigma <- t^2*diag(n_fea)
#return log posterior for the logistic regression
log_postlogis <- function(betas,y,x,mu,sigma){
  linPred <- x%*%betas
  logLik <- sum( linPred*y - log(1 + exp(linPred)) )
  logPrior <- dmvnorm(betas, mu, sigma, log=TRUE);
  return(logLik + logPrior)
}

#initial beta
init_betas <- matrix(0,n_fea,1)

#fnscale=-1 means that we minimize the negative log posterior. Hence, we maximize the log posterior.
OptimRes <- optim(par=init_betas,
                  fn=log_postlogis,
                  gr=NULL,
                  y,x,mu,sigma,
                  method=c("BFGS"),
                  control=list(fnscale=-1),
                  hessian=TRUE)
#Naming the coefficient by covariates
Xnames<-colnames(x)
posterior_mode<-data.frame(feature_name=Xnames,beta_mode=OptimRes$par)
# Computing approximate standard deviations.
approxpost_std <- solve(-OptimRes$hessian)
print('The posterior mode is:')
```

```
## [1] "The posterior mode is:"
```

```r
print(posterior_mode)
```

```
##   feature_name   beta_mode
## 1     Constant -0.04036943
## 2   HusbandInc -0.03730689
## 3    EducYears  0.17868950
## 4     ExpYears  0.12073637
## 5          Age -0.04618995
## 6  NSmallChild -1.47248930
## 7    NBigChild -0.02014458
```

```r
print('The inverse negative of the observed Hessian evaluated at the posterior mode is:')
```

```
## [1] "The inverse negative of the observed Hessian evaluated at the posterior mode is:"
```

```r
print(approxpost_std)
```

```
##              [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  1.909882159  4.032517e-03 -6.280726e-02  1.041874e-03 -0.0257559994
## [2,]  0.004032517  4.833287e-04 -9.147892e-04 -2.666479e-05 -0.0000642848
## [3,] -0.062807260 -9.147892e-04  7.958354e-03  5.508998e-05 -0.0003181372
## [4,]  0.001041874 -2.666479e-05  5.508998e-05  1.112877e-03 -0.0002845111
## [5,] -0.025755999 -6.428480e-05 -3.181372e-04 -2.845111e-04  0.0007547741
## [6,] -0.137712005  1.585545e-03 -1.438778e-02 -1.336628e-03  0.0055481315
## [7,] -0.088876440  4.986972e-06  1.133513e-04  7.206537e-04  0.0010449347
##              [,6]          [,7]
```

```
## [1,] -0.137712005 -8.887644e-02
## [2,]  0.001585545  4.986972e-06
## [3,] -0.014387780  1.133513e-04
## [4,] -0.001336628  7.206537e-04
## [5,]  0.005548132  1.044935e-03
## [6,]  0.227975343  1.122711e-02
## [7,]  0.011227114  2.690243e-02
```

```
#get the coefficients of the features
coef_mean<-posterior_mode[,2]
coef_covar=solve(-OptimRes$hessian)
coef<-rmvnorm(n=10000,
              mean=coef_mean,
              sigma=coef_covar)
coef_NSmallChild<-coef[,which(Xnames=='NSmallChild')]
intervals<-quantile(coef_NSmallChild,probs=c(0.025,0.975))
cat("The 95% equal tail posterior probability interval for the regression
coefficient is [",intervals[1],",",intervals[2],"]")
```

```
## The 95% equal tail posterior probability interval for the regression
## coefficient is [ -2.404466 , -0.5493319 ]
```

```
#compare the posterior means to the maximum likelihood estimates
glmModel<- glm(Work ~ 0 + ., data = WomenAtWork, family = binomial)
compared<-data.frame(posterior_mode,maximum_likli=glmModel$coefficients)[2:3]
print(compared)
```

```
##               beta_mode maximum_likli
## Constant    -0.04036943    0.02262929
## HusbandInc  -0.03730689   -0.03796308
## EducYears    0.17868950    0.18447411
## ExpYears     0.12073637    0.12131763
## Age         -0.04618995   -0.04858167
## NSmallChild -1.47248930   -1.56485140
## NBigChild   -0.02014458   -0.02526059
```

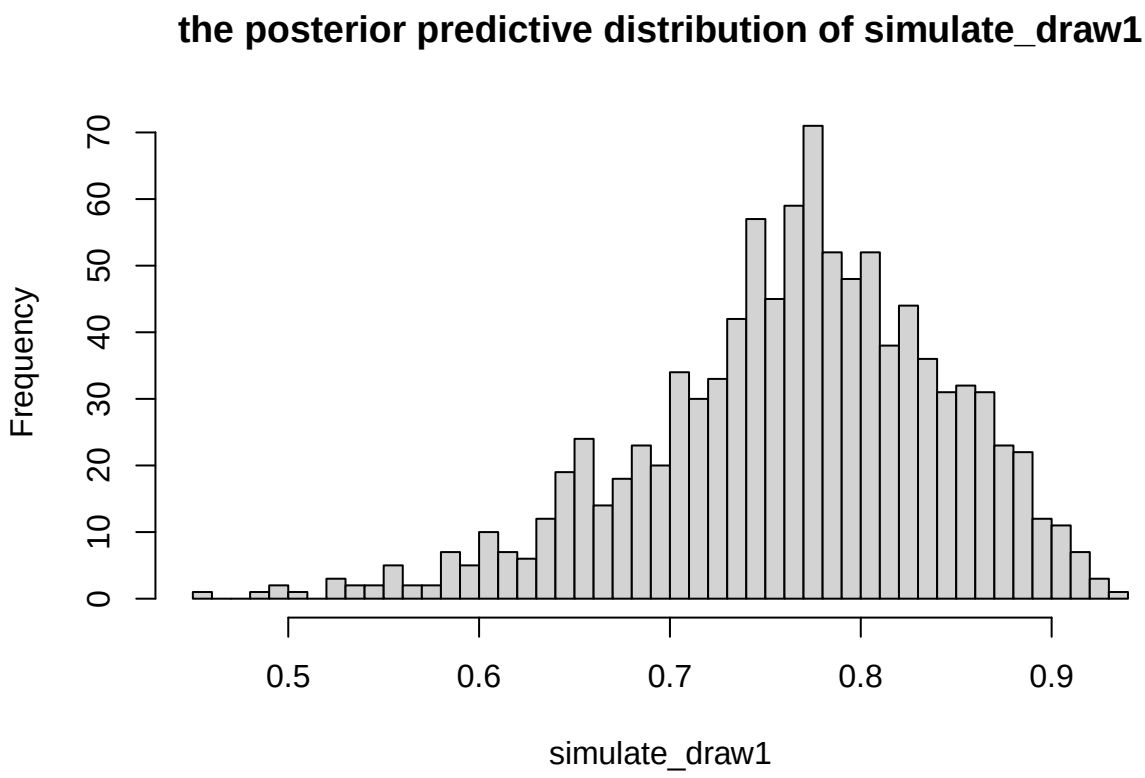
From the results above, by comparing the posterior means to the maximum likelihood estimates,we can see that the results using the maximum likelihood estimate and Posterior approximation are very close. Our estimation results are reasonable.

The mean value of regression coefficient to the variable NSmallChild is -1.47248930. Compared to other coefficient values, the absolute value of this value is big. We can say that this feature is of importance for the probability that woman works. And it has big negative impact.

## (b)

```
Constant<-1
Age<-40
NSmallChild<-1
NBigChild<-1
EducYears<-11
ExpYears<-7
HusbandInc<-18
x_new<-c(Constant,HusbandInc,EducYears,ExpYears,Age,NSmallChild,NBigChild)
simulate1<-function(x,samples,mean,sigma){
prob<-c()
```

```r
for(i in 1:samples){
  betas<-rmvnorm(n=1,mean=mean,sigma=sigma)
  prob_y1<-exp(x_new%*%t(betas))/(1+exp(x_new%*%t(betas)))
  prob_y0<-1-prob_y1
  prob=append(prob,prob_y0)
  }
  return(prob)
}
simulate_draw1<-simulate1(x_new,1000,coef_mean,coef_covar)
hist(simulate_draw1,breaks=50,main='the posterior predictive distribution of simulate_draw1')
```




(c)

```r
simulate2<-function(x,samples,mean,sigma,nsize){
  prob<-c()
  for(i in 1:samples){
    betas<-rmvnorm(n=1,mean=mean,sigma=sigma)
    prob_y1<-exp(x_new%*%t(betas))/(1+exp(x_new%*%t(betas)))
    prob_y0<-1-prob_y1
    prob=append(prob,prob_y0)
  }
  bin=rbinom(length(prob),size=nsize,prob=prob)
  return(bin)
}
```

```r
simulate_draw2<-simulate2(x_new,1000,coef_mean,coef_covar,13)
hist(simulate_draw2,main='the posterior predictive distribution of simulate_draw2')
```