

# Bayesian Learning Lab 2

Shipeng Liu

Jin Yan

2023-04-24

## Contents

<b>Assignment 1</b>	<b>2</b>
a) Test if the prior distribution is reasonable . . . . .	2
b) Write a function that simulate draws from the joint posterior distribution. . . . .	6
c) Use the simulated draws in (b) to simulate from the posterior distribution of $\tilde{x}$ . . . . .	10
d) Suggest a suitable prior that mitigates the potential overfitting problem. . . . .	11
<b>Assignment 2</b>	<b>12</b>
a) Consider the logistic regression model . . . . .	12
b) Write a function that simulate draws from the posterior predictive distribution of $\Pr(y = 0 x)$ .	13
c) Consider 13 women which all have the same features . . . . .	15
<b>Appendix</b>	<b>16</b>

# Assignment 1

```
mydata <- read_xlsx("Linkoping2022.xlsx")
```

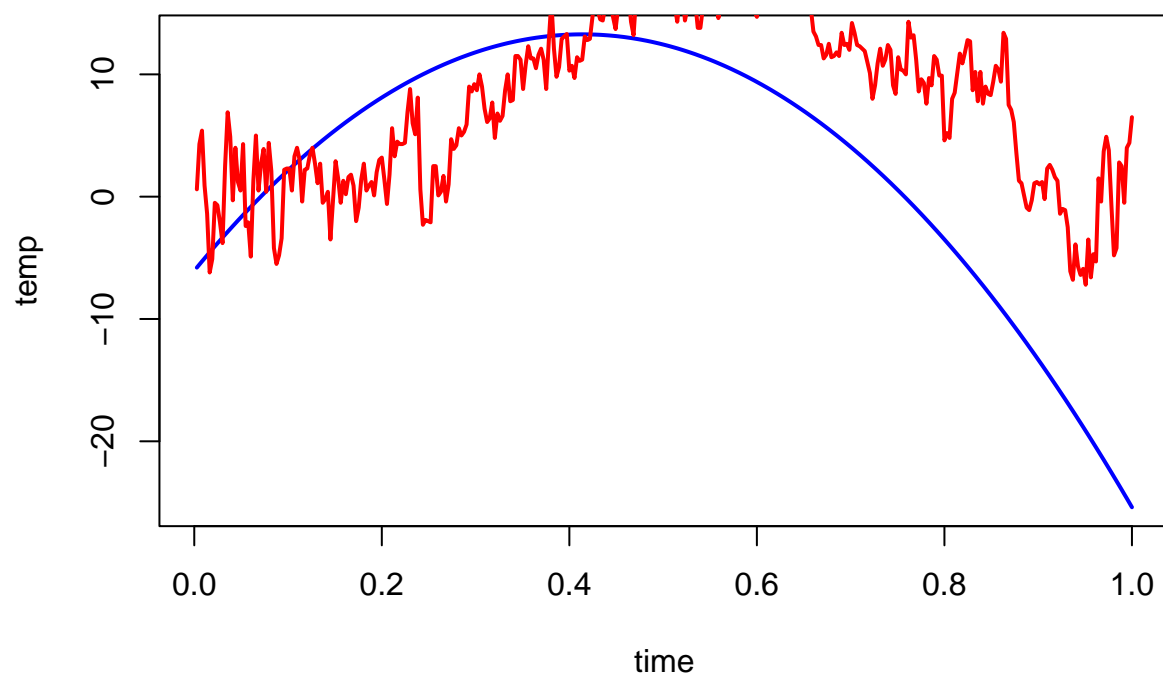
a) Test if the prior distribution is reasonable

```
#construct the time matrix
one <- rep(1,365)
time <- c(1:365) / 365
time_sq <- time**2
time_matrix <- as.matrix(data.frame(one,time,time_sq))

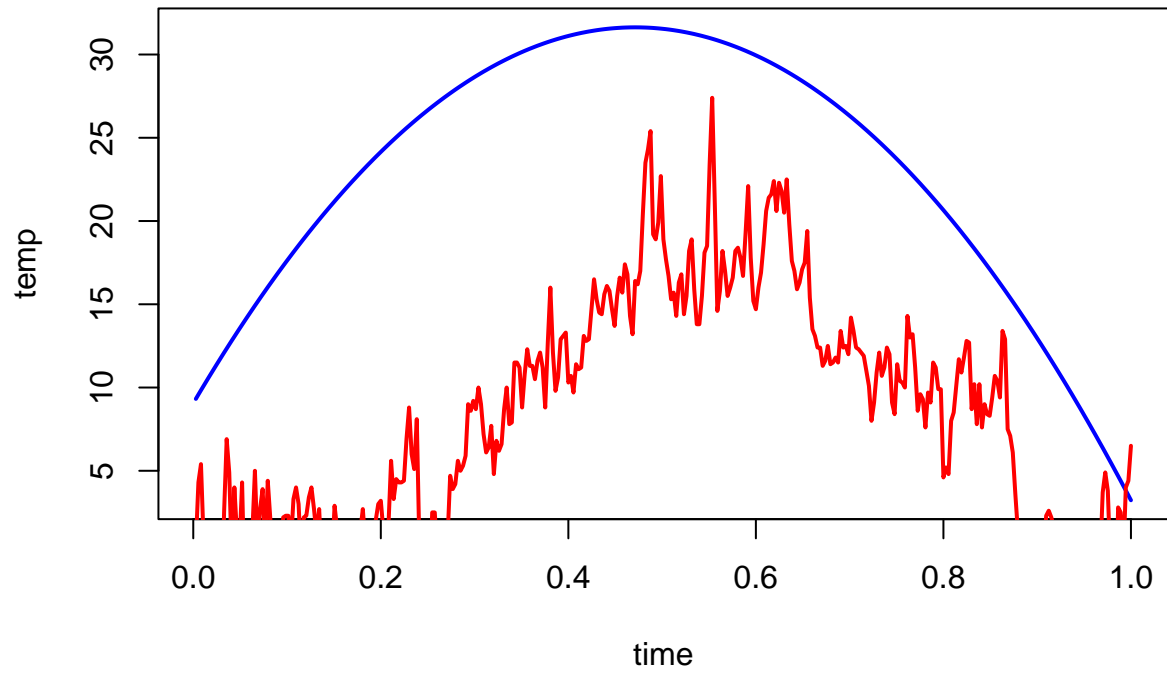
# set the values of hyperparameters
miu_0 <- c(0,100,-100)
omega_0 <- 0.01 * diag(3)
nu_0 <- 1
sigma_0_sq <- 1

#simulate
for(i in 1:4){
  X <- rchisq(1,nu_0)
  sigma_square <- nu_0 * sigma_0_sq / X
  beta <- rmvnorm(1,miu_0,sigma_square * solve(omega_0))
  beta <- as.matrix(beta)
  temp <- time_matrix %*% t(beta) + rnorm(1,0,sqrt(sigma_square))
  plot(time,temp,type = "l",col="blue",lwd=2,xlab = "time",ylab = "temp",main = "show the rationality of")
  lines(time,mydata$temp,type = "l",col="red",lwd=2)
}
```

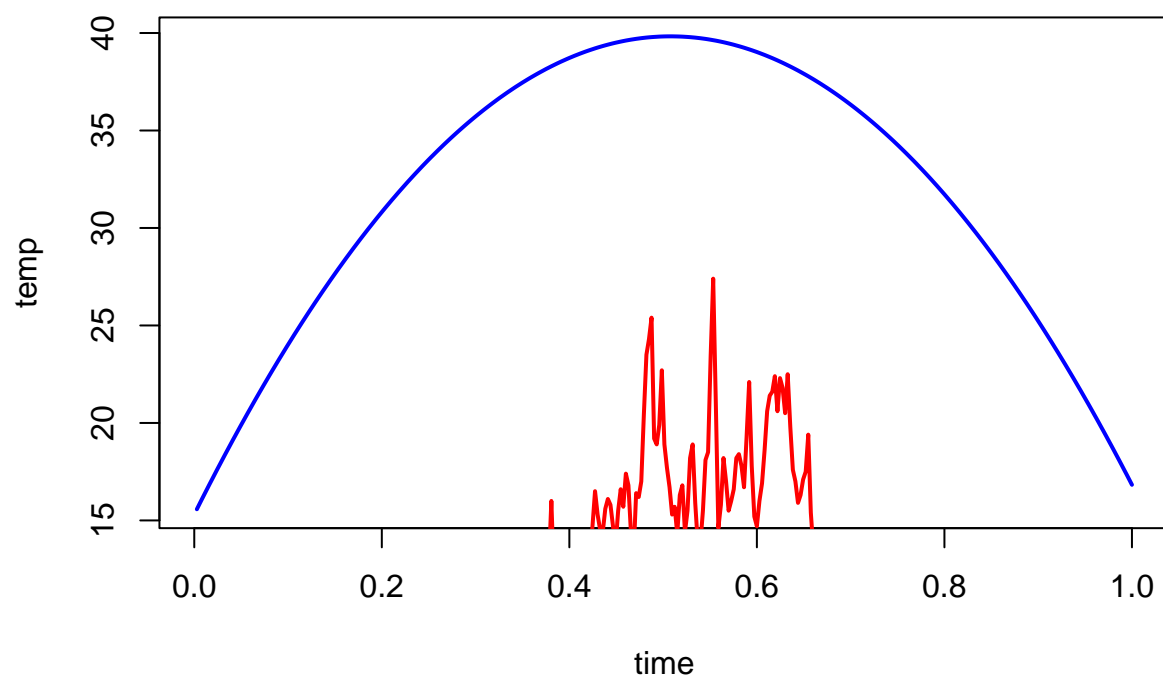
**show the rationality of prior**



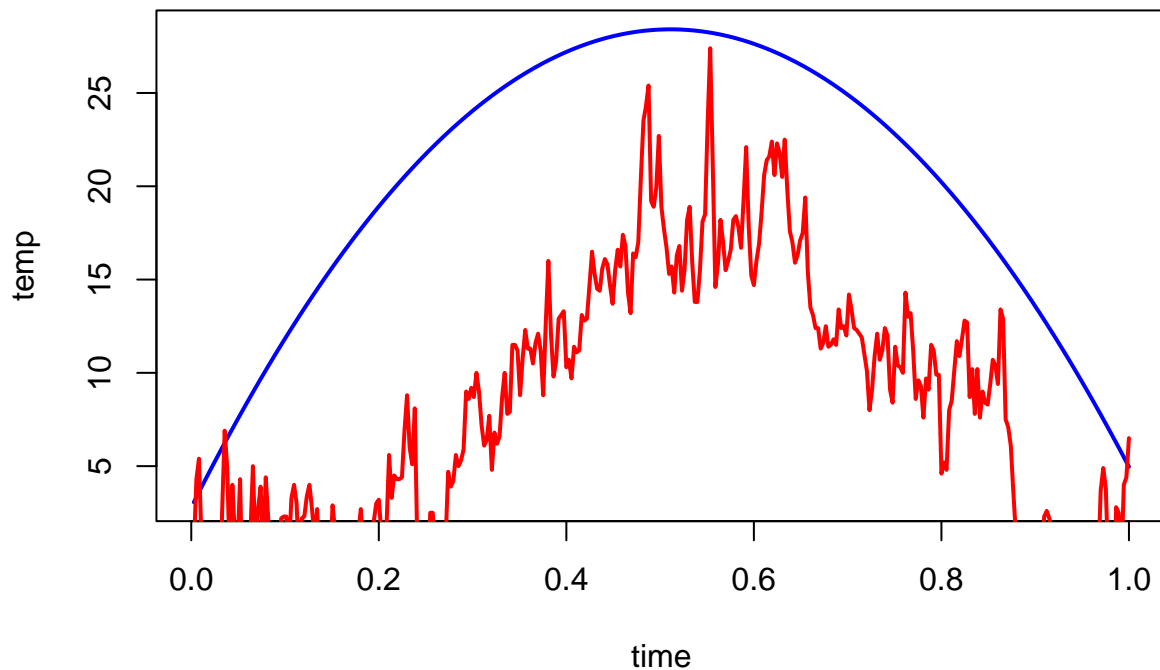
### show the rationality of prior



### show the rationality of prior



## show the rationality of prior



From the plot, we can see that the prior agrees my prior opinion to some degree.

b) Write a function that simulate draws from the joint posterior distribution.

Plot a histogram for each marginal posterior of the parameters.

```
# calculate the parameters used in join posterior distribution
y <- as.matrix(mydata$temp)
n <- 365
beta_hat <- solve(t(time_matrix) %*% time_matrix) %*% t(time_matrix) %*% y
miu_n <- solve(t(time_matrix) %*% time_matrix + omega_0) %*% (t(time_matrix) %*% time_matrix %*% beta_hat)
omega_n <- t(time_matrix) %*% time_matrix + omega_0
nu_n <- nu_0 + n
sigma_n_sq <- (nu_0 * sigma_0_sq + (t(y) %*% y + t(miu_0) %*% omega_0 %*% miu_0 - t(miu_n) %*% omega_n)) / nu_n
sigma_n_sq <- as.numeric(sigma_n_sq)
#simulate

beta <- c()
sigma_square <- c()
for(i in 1:100){
  X <- rchisq(1,nu_n)
  sigma_square_temp <- nu_n * sigma_n_sq / X
  beta_temp <- rmvnorm(1,miu_n,sigma_square_temp * solve(omega_n))
  beta <- rbind(beta,beta_temp)
```

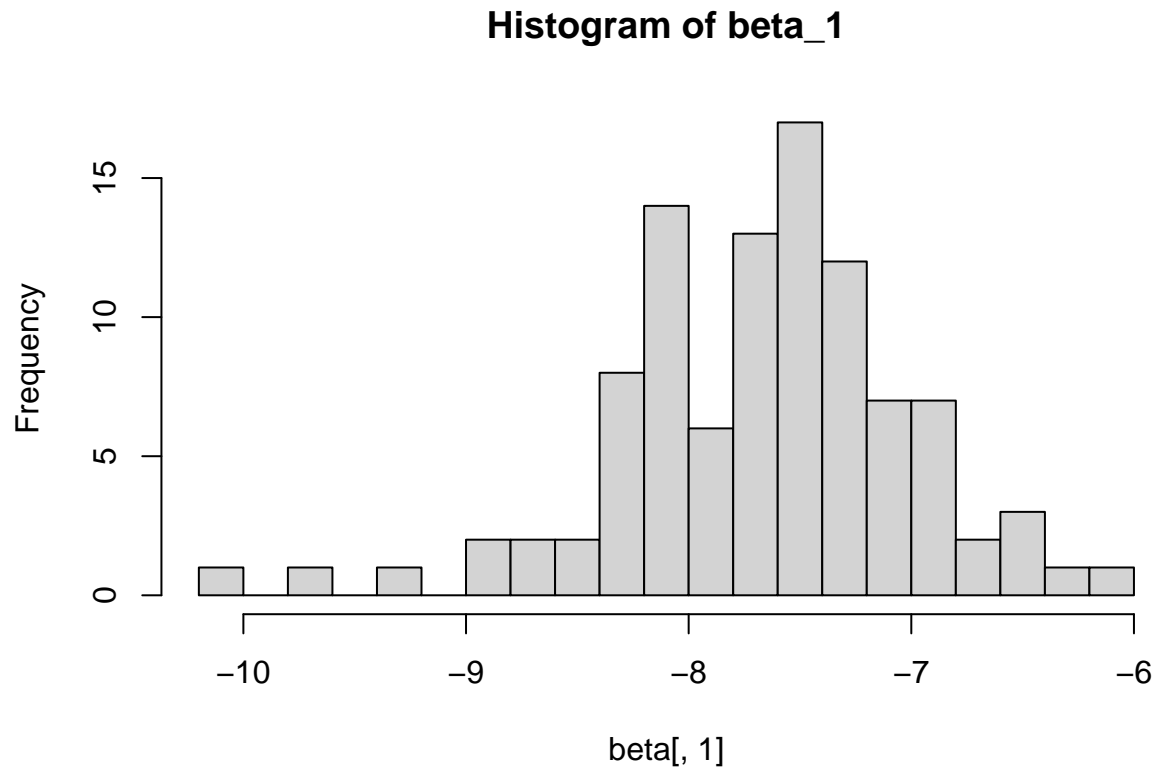
```

sigma_square <- c(sigma_square,sigma_square_temp)
}

#1.2.1

# plot the histogram for the first beta
hist(beta[,1],main = "Histogram of beta_1",breaks = 20)

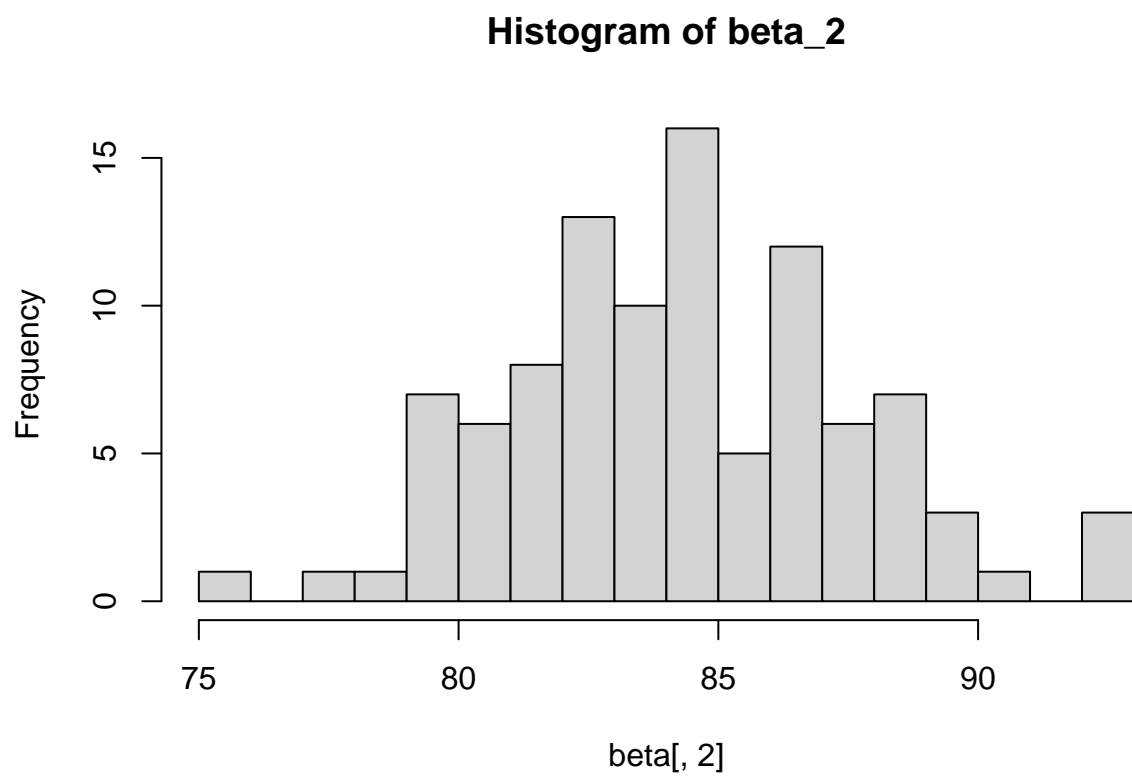
```



```

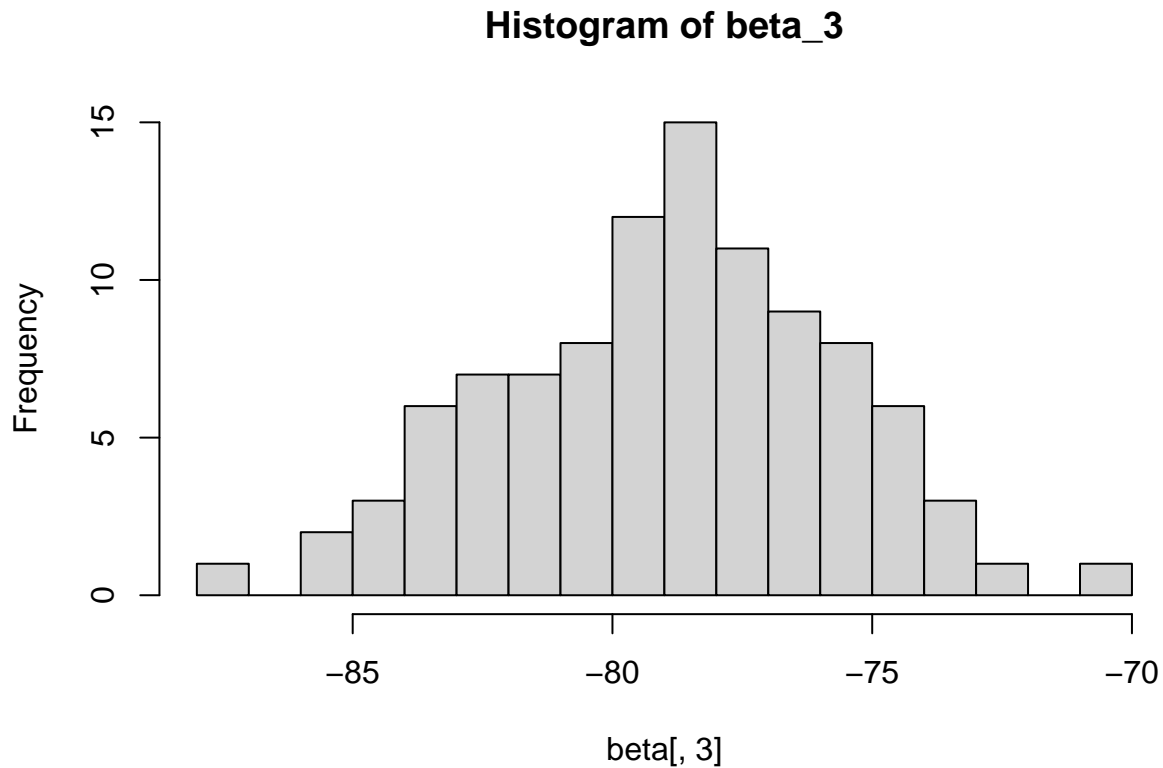
# plot the histogram for the second beta
hist(beta[,2],main = "Histogram of beta_2",breaks = 20)

```



```
# plot the histogram for the third beta  
hist(beta[,3],main = "Histogram of beta_3",breaks = 20)
```





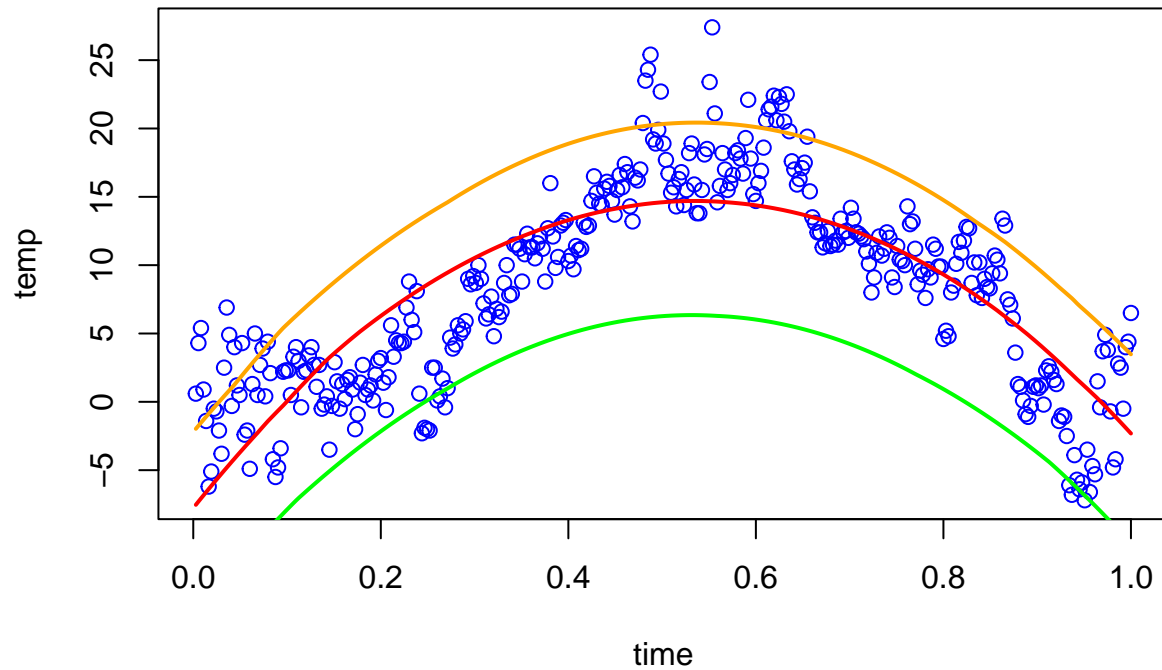
Make a scatter plot of the temperature data and overlay a curve for the posterior median of the function

```
#1.2.2
beta <- as.matrix(beta)
vector_epsilon <- unlist(lapply(sqrt(sigma_square), rnorm, n=1, mean=0))
temp <- time_matrix %*% t(beta)
temp <- + t(vector_epsilon + t(temp)) # This is used for adding epsilon.
temp_median <- apply(temp, 1, median)

temp_lower <- apply(temp, 1, quantile, prob = 0.05)
temp_upper <- apply(temp, 1, quantile, prob = 0.95)

plot(time, mydata$temp, col="blue", xlab = "time", ylab = "temp", main = "Show the Result of 2.b")
lines(time, temp_upper, type = "l", col="orange", lwd=2)
lines(time, temp_median, type = "l", col="red", lwd=2)
lines(time, temp_lower, type = "l", col="green", lwd=2)
```

## Show the Result of 2.b



From the graph we can see that the interval contains most of the data points, and they should do that.

c) Use the simulated draws in (b) to simulate from the posterior distribution of  $\tilde{x}$ .

```
x_tilde <- -beta[,2] / (2 * beta[,3])
x_tilde
```

```
## [1] 0.5420129 0.5293022 0.5261583 0.5432244 0.5291202 0.5347557 0.5326736
## [8] 0.5276077 0.5436378 0.5428776 0.5320778 0.5489579 0.5413154 0.5324786
## [15] 0.5354272 0.5400363 0.5304211 0.5300211 0.5346290 0.5368402 0.5299174
## [22] 0.5321721 0.5444406 0.5370970 0.5283796 0.5341972 0.5282065 0.5357307
## [29] 0.5349396 0.5354793 0.5325284 0.5371950 0.5339731 0.5345193 0.5275672
## [36] 0.5327869 0.5333448 0.5384360 0.5420655 0.5308984 0.5316733 0.5372904
## [43] 0.5343835 0.5324682 0.5287708 0.5465618 0.5350339 0.5276077 0.5313244
## [50] 0.5364743 0.5389090 0.5289355 0.5266437 0.5320691 0.5350237 0.5388386
## [57] 0.5345634 0.5242837 0.5309726 0.5297686 0.5307080 0.5340708 0.5367051
## [64] 0.5300490 0.5321094 0.5330351 0.5307358 0.5337394 0.5383729 0.5381361
## [71] 0.5322437 0.5393150 0.5302180 0.5289409 0.5363281 0.5300179 0.5270181
## [78] 0.5407721 0.5239068 0.5373327 0.5455194 0.5388153 0.5313245 0.5397620
## [85] 0.5315154 0.5371852 0.5386670 0.5344879 0.5348233 0.5241488 0.5335740
## [92] 0.5350812 0.5429530 0.5358321 0.5376954 0.5336519 0.5306696 0.5328628
## [99] 0.5216647 0.5331768
```

**d) Suggest a suitable prior that mitigates the potential overfitting problem.**

The prior should be  $B_k \sim N(0, 1000 * \delta^k)$   $k = [0, 1, \dots, 10]$   $\delta = 0.5$

## Assignment 2

```
watdata <- read.table("WomenAtWork.dat", header = TRUE)
```

a) Consider the logistic regression model

Compute the posterior mode and  $J(\text{posteriorMode})$  using optim

```
#compute the betamode and J(betamode)

Npar=dim(watdata)[2]-1
X=as.matrix(watdata[,2:8])
y=as.matrix(watdata[,1])
# Select the initial values for beta
initVal <- matrix(0,Npar,1)
# Setting up the prior
mu=rep(0,Npar)
sigma=4*diag(Npar)

LogPostLogistic <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  # The LogLikelihood
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  # The LogPrior
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);
  # The LogPosterior=LogLikelihood+LogPrior
  return(logLik + logPrior)
}

# The argument control is a list of options to the optimizer optim, where fnscale=-1 means that we minimize
# the negative log posterior. Hence, we maximize the log posterior.
OptimRes <- optim(initVal,LogPostLogistic,gr=NULL,y,X,mu,sigma,method=c("BFGS"),control=list(fnscale=-1))

betamode=data.frame(OptimRes$par)
rownames(betamode)=colnames(watdata)[2:8]
Jbetamode=solve(-OptimRes$hessian)

cat('The posterior mode is:\n')
```

```
## The posterior mode is:
```

```
print(betamode)
```

```
##           OptimRes.par
## Constant    -0.04036943
## HusbandInc  -0.03730689
## EducYears    0.17868950
## ExpYears     0.12073637
```

```
## Age          -0.04618995
## NSmallChild -1.47248930
## NBigChild   -0.02014458
```

```
cat("\nThe posterior covariance is:\n")
```

```
##
## The posterior covariance is:
```

```
print(Jbetamode)
```

```
##           [,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
```

Compute an approximate 95% equal tail posterior probability interval for the regression coefficient to the variable NSmallChild

```
nSmallChild=rmvnorm(1000,mean=OptimRes$par,sigma=Jbetamode)[,6]
etcInterval=quantile(nSmallChild,c(0.025,0.975))
print(etcInterval)
```

```
##          2.5%      97.5%
## -2.3927939 -0.5912928
```

It seems this feature is of importance for the probability that a woman works, Cause the absolute coefficient of NSmallChild in the posterior mode is greater than other coefficients. And it is negative, This reveals that women with more young children at home are more likely to be home-care than work.

b) Write a function that simulate draws from the posterior predictive distribution of  $\Pr(y = 0|x)$

```
draw_woman=function(drawnum,mean,sigma){
  nparam=rmvnorm(drawnum,mean=mean,sigma=sigma)
```

```

ncoef=matrix(c(1,18,11,7,40,1,1))
res=nparam%*%ncoef
res=1-(exp(res)/(1+exp(res)))
return(res)
}

res=draw_woman(1000,OptimRes$par,Jbetamode)

#plot
res=data.frame("Prob"=res)

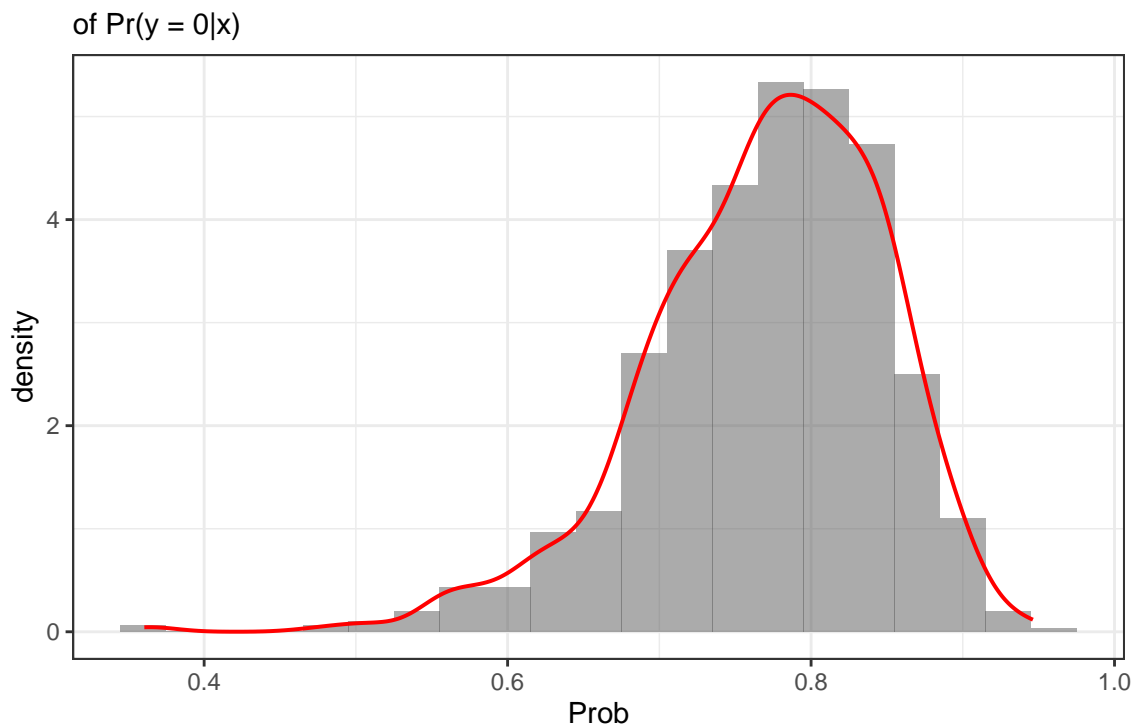
ggplot(data=res,aes(x=Prob))+
  geom_histogram(aes(y=..density..),binwidth=0.03,alpha=0.5)+
  geom_density(color='red',size=0.7)+
  labs(title='The posterior predictive distribution
',subtitle="of Pr(y = 0|x)",tag='Fig 2.2')+
  theme_bw()

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.

## Warning: The dot-dot notation ('..density..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(density)' instead.

```

Fig 2.2 The posterior predictive distribution



The plot show that woman in this situation(a 40-year-old woman, with two children (4 and 7 years old), 11 years of education, 7 years of experience, and a husband with an income of 18) prone not to work.

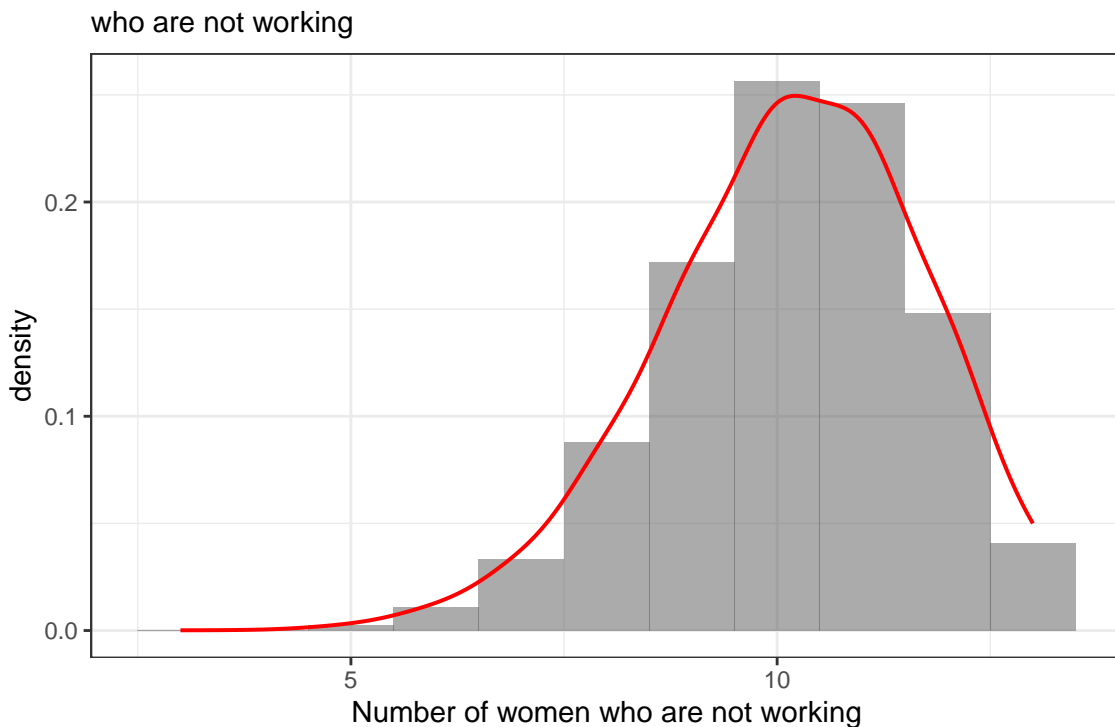
c) Consider 13 women which all have the same features

```
# Kernel density approximation
res=draw_woman(1000,OptimRes$par,Jbetamode)
dens=density(res)
max_pos=which.max(dens$y)
max_x=dens$x[max_pos]

# We set the Prob of these women don't work as max_x
binom_res=rbinom(10000, 13, max_x)

#Plot
ggplot(data=data.frame("sample"=binom_res),aes(x=sample))+
  geom_histogram(aes(y=..density..),binwidth=1,alpha=0.5)+
  geom_density(color='red',size=0.7,adjust=2.5)+
  labs(title='The posterior predictive distribution for the number of women
',subtitle='who are not working',tag='Fig 2.3')+
  xlab("Number of women who are not working")+
  theme_bw()
```

Fig 2.3 The posterior predictive distribution for the number of women



Of the 13 women who fit these criteria, it is most likely that 10 to 11 of them are not working.

## Appendix

```
library("mvtnorm")
library(ggplot2)
library(readxl)
mydata <- read_xlsx("Linkoping2022.xlsx")
#construct the time matrix
one <- rep(1,365)
time <- c(1:365) / 365
time_sq <- time**2
time_matrix <- as.matrix(data.frame(one,time,time_sq))

# set the values of hyperparameters
miu_0 <- c(0,100,-100)
omega_0 <- 0.01 * diag(3)
nu_0 <- 1
sigma_0_sq <- 1

#simulate
for(i in 1:4){
  X <- rchisq(1,nu_0)
  sigma_square <- nu_0 * sigma_0_sq / X
  beta <- rmvnorm(1,miu_0,sigma_square * solve(omega_0))
  beta <- as.matrix(beta)
  temp <- time_matrix %*% t(beta) + rnorm(1,0,sqrt(sigma_square))
  plot(time,temp,type = "l",col="blue",lwd=2,xlab = "time",ylab = "temp",main = "show the rationality of")
  lines(time,mydata$temp,type = "l",col="red",lwd=2)
}
# calculate the parameters used in join posterior distribution
y <- as.matrix(mydata$temp)
n <- 365
beta_hat <- solve(t(time_matrix) %*% time_matrix) %*% t(time_matrix) %*% y
miu_n <- solve(t(time_matrix) %*% time_matrix + omega_0) %*% (t(time_matrix) %*% time_matrix %*% beta_hat)
omega_n <- t(time_matrix) %*% time_matrix + omega_0
nu_n <- nu_0 + n
sigma_n_sq <- (nu_0 * sigma_0_sq + (t(y) %*% y + t(miu_0) %*% omega_0 %*% miu_0 - t(miu_n) %*% omega_n)) / nu_n
sigma_n_sq <- as.numeric(sigma_n_sq)
#simulate

beta <- c()
sigma_square <- c()
for(i in 1:100){
  X <- rchisq(1,nu_n)
  sigma_square_temp <- nu_n * sigma_n_sq / X
  beta_temp <- rmvnorm(1,miu_n,sigma_square_temp * solve(omega_n))
  beta <- rbind(beta,beta_temp)
  sigma_square <- c(sigma_square,sigma_square_temp)
}

#1.2.1

# plot the histogram for the first beta
hist(beta[,1],main = "Histogram of beta_1",breaks = 20)
```



```

# plot the histogram for the second beta
hist(beta[,2],main = "Histogram of beta_2",breaks = 20)

# plot the histogram for the third beta
hist(beta[,3],main = "Histogram of beta_3",breaks = 20)
#1.2.2
beta <- as.matrix(beta)
vector_epsilon <- unlist(lapply(sqrt(sigma_square),rnorm,n=1,mean=0))
temp <- time_matrix %*% t(beta)
temp <- + t(vector_epsilon + t(temp))# This is used for adding epsilon.
temp_median <- apply(temp,1,median)

temp_lower <- apply(temp,1,quantile,prob = 0.05)
temp_upper <- apply(temp,1,quantile,prob = 0.95)

plot(time,mydata$temp,col="blue",xlab = "time",ylab = "temp",main = "Show the Result of 2.b")
lines(time,temp_upper,type = "l",col="orange",lwd=2)
lines(time,temp_median,type = "l",col="red",lwd=2)
lines(time,temp_lower,type = "l",col="green",lwd=2)
x_tilde <- -beta[,2] / (2 * beta[,3])
x_tilde
watdata <- read.table("WomenAtWork.dat", header = TRUE)
#compute the betamode and J(betamode)

Npar=dim(watdata)[2]-1
X=as.matrix(watdata[,2:8])
y=as.matrix(watdata[,1])
# Select the initial values for beta
initVal <- matrix(0,Npar,1)
# Setting up the prior
mu=rep(0,Npar)
sigma=4*diag(Npar)

LogPostLogistic <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  # The LogLikelihood
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  # The LogPrior
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);
  # The LogPosterior=LogLikelihood+LogPrior
  return(logLik + logPrior)
}

# The argument control is a list of options to the optimizer optim, where fnscale=-1 means that we minimize
# the negative log posterior. Hence, we maximize the log posterior.
OptimRes <- optim(initVal,LogPostLogistic,gr=NULL,y,X,mu,sigma,method=c("BFGS"),control=list(fnscale=-1))

betamode=data.frame(OptimRes$par)
rownames(betamode)=colnames(watdata)[2:8]
Jbetamode=solve(-OptimRes$hessian)

cat('The posterior mode is:\n')

```

```

print(betamode)
cat("\nThe posterior covariance is:\n")
print(Jbetamode)

nSmallChild=rmvnorm(1000,mean=OptimRes$par,sigma=Jbetamode)[,6]
etcInterval=quantile(nSmallChild,c(0.025,0.975))
print(etcInterval)
draw_woman=function(drawnum,mean,sigma){
  nparam=rmvnorm(drawnum,mean=mean,sigma=sigma)
  ncoef=matrix(c(1,18,11,7,40,1,1))
  res=nparam%*%ncoef
  res=1-(exp(res)/(1+exp(res)))
  return(res)
}

res=draw_woman(1000,OptimRes$par,Jbetamode)

#plot
res=data.frame("Prob"=res)

ggplot(data=res,aes(x=Prob))+
  geom_histogram(aes(y=..density..),binwidth=0.03,alpha=0.5)+
  geom_density(color='red',size=0.7)+
  labs(title='The posterior predictive distribution
',subtitle='of Pr(y = 0|x)',tag='Fig 2.2')+
  theme_bw()

# Kernel density approximation
res=draw_woman(1000,OptimRes$par,Jbetamode)
dens=density(res)
max_pos=which.max(dens$y)
max_x=dens$x[max_pos]

# We set the Prob of these women don't work as max_x
binom_res=rbinom(10000, 13, max_x)

#Plot
ggplot(data=data.frame("sample"=binom_res),aes(x=sample))+
  geom_histogram(aes(y=..density..),binwidth=1,alpha=0.5)+
  geom_density(color='red',size=0.7,adjust=2.5)+
  labs(title='The posterior predictive distribution for the number of women
',subtitle='who are not working',tag='Fig 2.3')+
  xlab("Number of women who are not working")+
  theme_bw()

```