732A91-Lab2-Report

Fengjuan Chen(fench417) Zhixuan Duan(zhidu838)

4/28/2020

1. Linear and polynomial regression

(a) Determining the prior distribution of the model parameters

The quadratic regression model in this problem is

$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon$, where $\epsilon \overset{iid}{\sim} N(0, \sigma^2)$.

The conjugate prior for this linear regression model is the joint prior for β and σ^2.

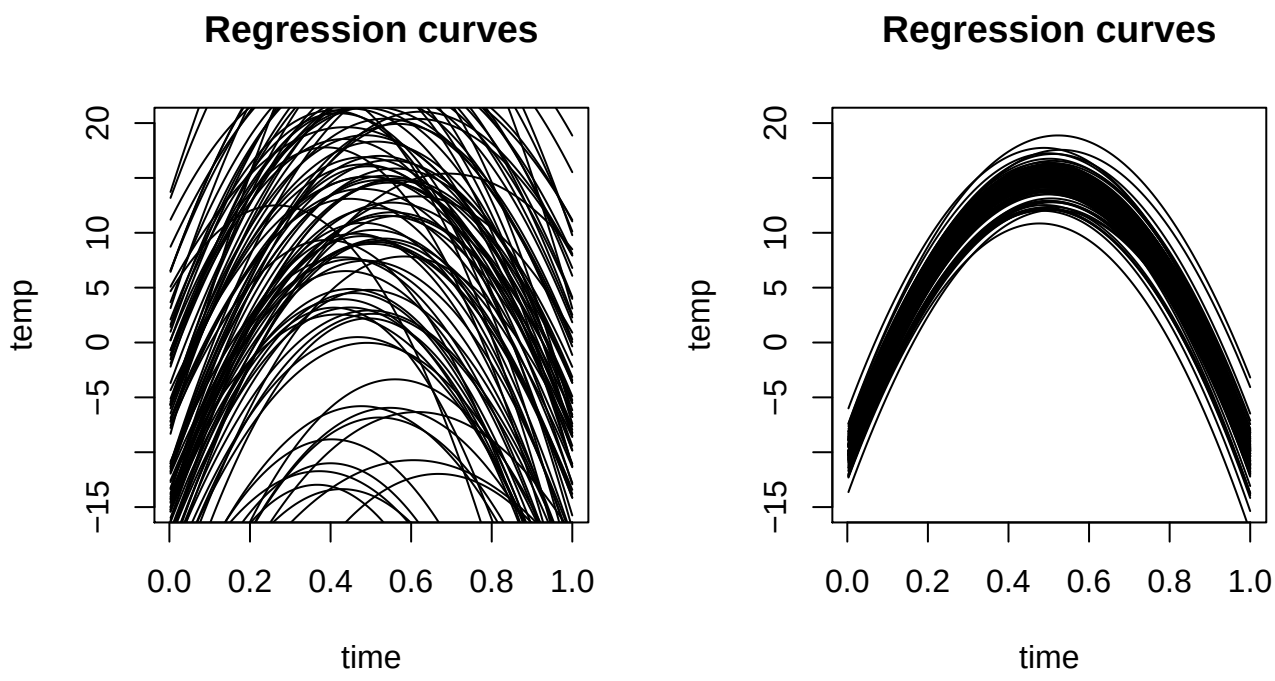
$\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$

$\sigma^2 \sim Inv - \chi^2(v_0, \sigma_0^2)$

The task is to set the prior hyperparameters μ_0,Ω_0,v_0 and σ_0^2 to sensible values.

We first check the join prior with $\mu_0 = (-10, 100, -100)^T$,$\Omega_0 = 0.01 \cdot I_3$,$v_0 = 4$ and $\sigma_0^2 = 1$ by simulating 100 draws from the joint prior of all parameters.

We compute the regression curve for every draw and plot them in one figure.



Our prior opions about temperatures in Linkoping of one year are: 4 seasons in a year; time from 0.4 to 0.8 corresponds to the high temperatures; time between 0.5 and 0.6 has the highest temperature; the range of temperature is about between -12 and 20.

From this figure, we can see that some of these 100 regression curves have the good shape that agrees with our prior opions. However, still many regression curves are far away from our prior opions.

Does the collection of curves look reasonable?

From the phenomenon in the figure we described above, we feel that the collection of curves does not look reasonable because of the large variance in $\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$.

To confirm that assumption we calculate the result of $\sigma^2\Omega_0^{-1}$ for each draw from $\sigma^2 \sim Inv-\chi^2(v_0, \sigma_0^2)$. The range of diagonals in $\sigma^2\Omega_0^{-1}$ is [30.3, 1470], which means that each element in β $(\beta_0, \beta_1, \beta_2)$ has standard deviation from 5.5 to 38.3. It is a large variant for the mean $\mu_0 = (-10, 100, -100)^T$.

So we decide to change the result of $\sigma^2\Omega_0^{-1}$ by tuning the hyperparameter σ_0^2 from 1 to a very small number, 0.01. This is equal to tuning the Ω_0 from $0.01 \cdot I_3$ to a relavtive big number such as $1 \cdot I_3$.

We show the adapted 100 regression curves in another figure next the figure of origianl regression curves .

(b) Simulate from the joint posterior distribution

The joint posterior distribution of $\beta_0, \beta_1, \beta_2$ and σ^2 :

$\beta|\sigma^2, y \sim N(\mu_n, \sigma^2\Omega_n^{-1})$

$\sigma^2|y \sim Inv-\chi^2(v_n, \sigma_n^2)$ where

$\mu_n = (X'X + \Omega_0)^{-1}(X'X\hat{\beta} + \Omega_0\mu_0)$

$\Omega_n = X'X + \Omega_0$

$X = (x_1, x_2, x_3), x_1 = (1, 1, ..., 1)^T, x_2 = time, x_3 = time^2$

$\hat{\beta} = (X'X)^{-1}X'y$

$v_n = v_0 + n$, $n = 366$

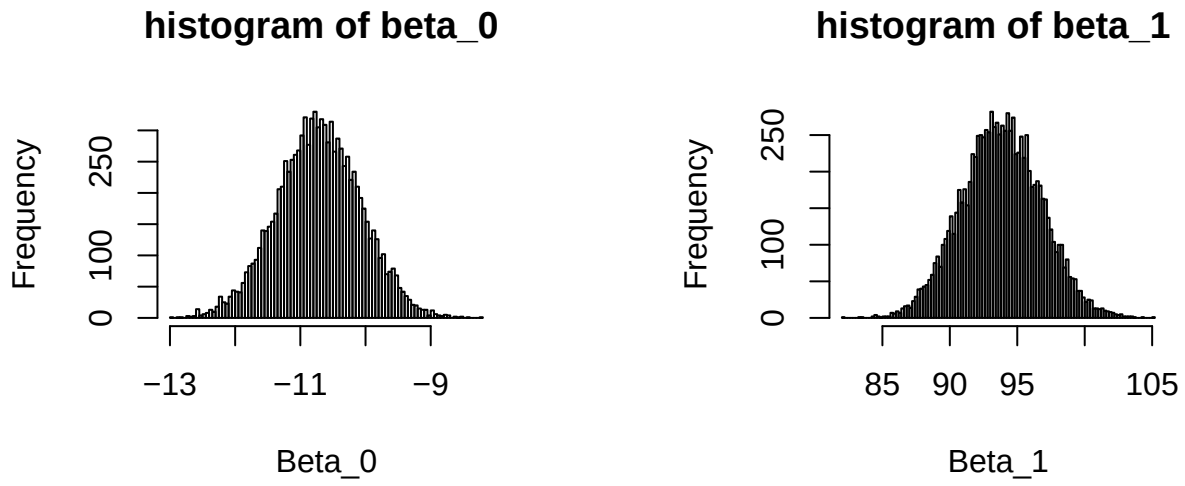
$v_n\sigma_n^2 = v_0\sigma_0^2 + (y'y + \mu_0'\Omega_0\mu_0 - \mu_n'\Omega_0\mu_n)$

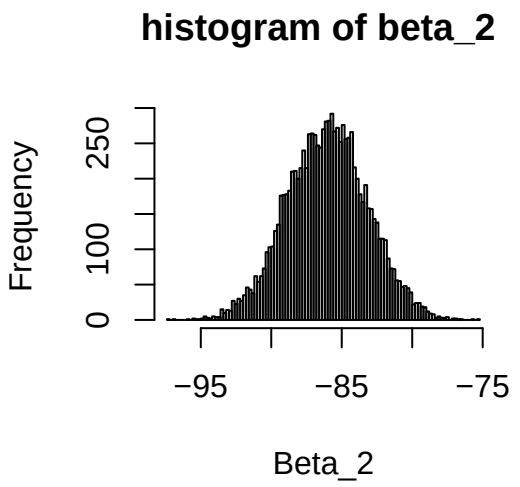
$y = TempLinkoping\$temp$

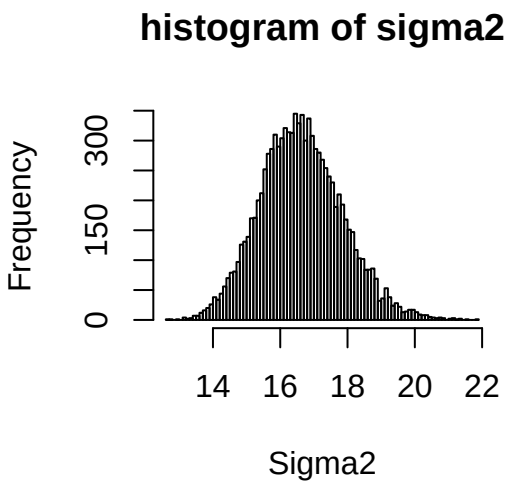
To implement simulating 10000 draws from the joint posterior distribution of $\beta_0, \beta_1, \beta_2$ and σ^2, we first simulate 10000 draws from $\sigma^2|y \sim Inv-\chi^2(v_n, \sigma_n^2)$ and then simulate 10000 draws from $\beta|\sigma^2, y \sim N(\mu_n, \sigma^2\Omega_n^{-1})$ with the simulated 10000 σ^2.

Hyperparameters: $\mu_0 = (-10, 105, -100)^T$,$\Omega_0 = 0.01 \cdot I_3$, $v_0 = 4$ and $\sigma_0^2 = 0.01$.

Then we plot the marginal posteriors for each parameter as a histogram.







We compute the posterior median (cumulative probability=0.5) of parameter $\beta_0, \beta_1, \beta_2$ and the corresponding regression function $f(time) = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$.

Then we find 95% equal tail posterior probability interval for $\beta_0, \beta_1, \beta_2$.

Table 1: posterior medians of beta

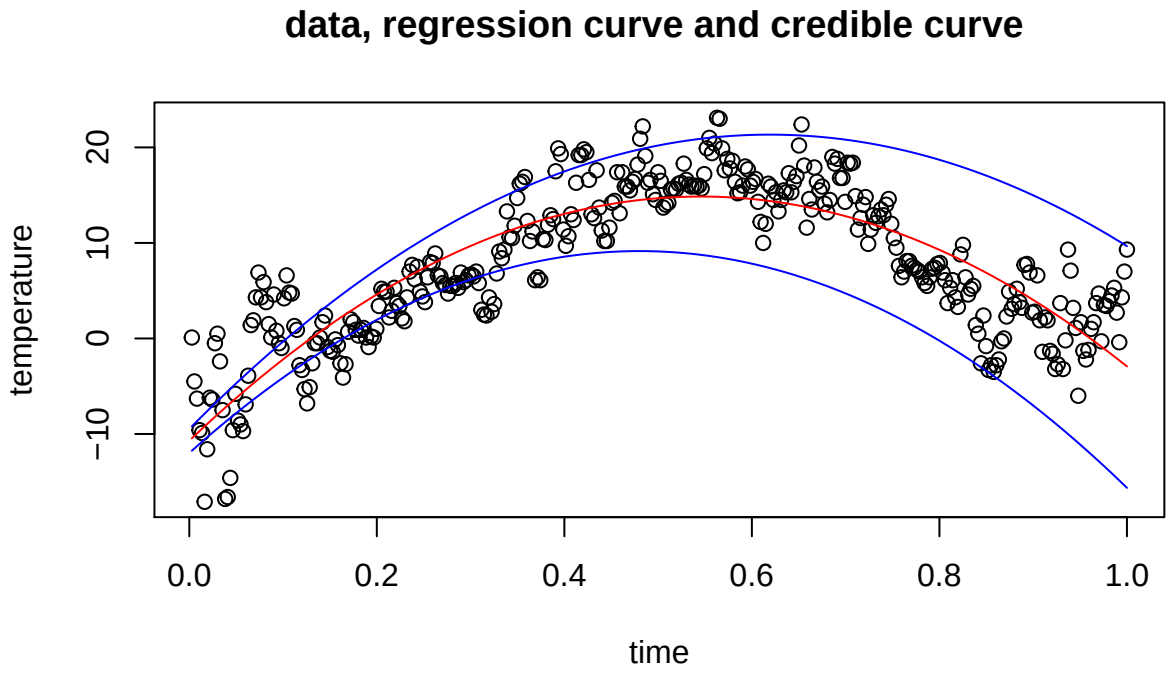
beta0	beta1	beta2
-10.71308	93.78306	-85.99667

Table 2: 95% equal tail posterior credible intervals for belta

	beta0	beta1	beta2
lower bound	-11.994030	88.05766	-91.70161
upper bound	-9.479447	99.57090	-80.43191

Finally, we calculate $f(time)$ corresponding to the lower 2.5% and upper 97.5% equal tail posterior probability intervals of $\beta_0, \beta_1, \beta_2$ and plot data points, predictive regression function for posteriror modian as well as interval bands in one figure.

The red curve is the posterior median of the regression function $f(time)$ while two blue curves are the lower 2.5% and upper 97.5% posterior credibale interval for $f(time)$.



Do the interval bands contain most of the data points? Should they?

The interva bands contain most of the data points, but less than 95% of them.

The 95% equal tail posterior credible interval is not same as the 95% confidence interval, which will contain 95% observed data. The 95% equal tail posterior credible interval means that probability of estimated parameter (a random variable) in this interval is 0.95. Therefore, these interval bands may contain most of the data points, but can not guarantee the proportion is equal to 95%.

(c) Simulate from the posterior distribution of x_tilde

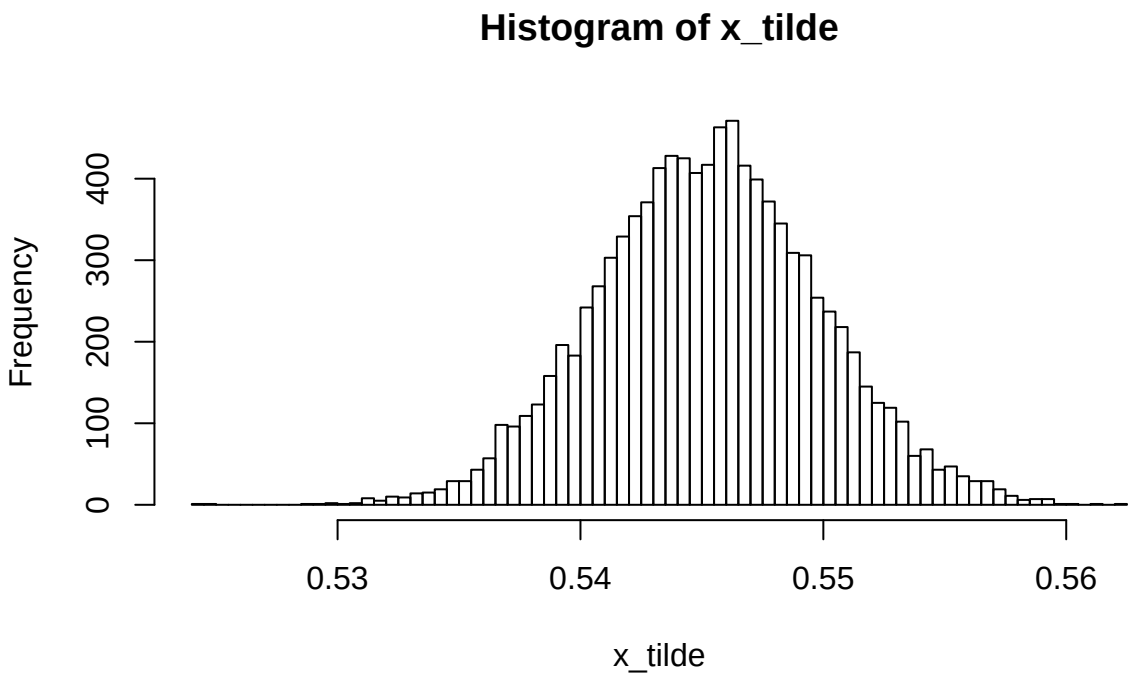
The $\tilde{x}$ represents the time where $f(time)$ is maximual. Since $f(time) = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$ is a quadratic function of time and the coefficient of $time^2$ is negative, we have a maximum value of $f(time)$.

The value of $-\frac{b}{a2}$ is the x value of the vertex of the parabola corresponding to the quadratic function

$f(x) = ax^2 + bx + c$.

Here, $a = \beta_2$ and $b = \beta_1$. We have already simulated 10000 draws from the joint posterior distribution of β_1 and β_2 in (b). We use them to obtain 10000 draws $\tilde{x} = -\frac{\beta_1}{2\beta_2}$.

The histogram of 10000 draws of $\tilde{x}$ is shown below.



(d) A suitable prior for mitigating the overfit

If we want to estimate a polynomial model of order 7 and conquer the overfitting problem, we can use a conjugate prior $\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$ and set $\mu_0 = (0,0,0,0,0,0,0,0)$ and $\Omega_0^{=}\lambda \cdot I_8$ to implement regularization.

The shrinkage will go towards zero, that is $\hat{\beta} \to 0$, if $\lambda \to \infty$.

Therefore, we can choose a large λ, say 100, or 1000, to mitigate the overfitting.

2. Posterior approximation for classification with logistic regression

(a) Approximate the posterior distribution of the 8-dim parameter vector

Approximate normal posterior in large samples:

$\theta|y \overset{approx}{\sim} N[\tilde{\theta}, J_y^{-1}(\tilde{\theta})]$.

To obtain the $\tilde{\theta}$ and $J_y^{-1}(\tilde{\theta})$ we use the standard optimization routines (optim() in R).

If we use the expression proportional to $logp(\theta|y)$ and initial values for θ as inputs, the optim() function will return $logp(\tilde{\theta}|y)$, $\tilde{\theta}$ and Hessian matrix $(-J_y(\tilde{\theta}))$.

Therefore we construct the expression proportional to $logp(\theta|y)$, which is $log\_likelihood + log\_Prior$.

Because this is logistic regression model, we have $Pr(y=1|X) = \frac{exp(X^T\beta)}{1+exp(X^T\beta)}$.

Then the likelihood is $p(y|X,\beta) = \prod_{i=1}^{n} \frac{[exp(X^T\beta)]^{y_i}}{1+exp(X^T\beta)}$, where X includes the constant 1 corresponding to the intercept, β is the parameter to estimate(θ in posterior distribution).

The log_likelihood is

$logp(y|X,\beta) = log\prod_{i=1}^{n}\frac{[exp(X^T\beta)]^{y_i}}{1+exp(X^T\beta)} = \sum_{i=1}^{n} y_i log(exp(X^T\beta)) - n\cdot log(1+exp(X^T\beta))$

$= \sum_{i=1}^{n} y_i\cdot(X^T\beta) - n\cdot log(1+exp(X^T\beta)) = \sum_{i=1}^{n}[y_i\cdot(X^T\beta) - log(1+exp(X^T\beta))]$.

The Prior is $\beta \sim N(0,\tau^2 I)$, where $\tau = 10$.

So, the log_Prior is the log of density of $\beta \sim N(0,10^2)$, which can be obtained by

dmvnorm(beta, mean = mu, Sigma, log=TRUE).

From these two values of log_likelihood and log_Prior, we can obtain the log_Posterior $logp(\theta|y)$. In this problem, θ is the 8-dimensional coefficients β.

We set the initial values of β as $init\_beta = (0,0,0,0,0,0,0,0)$.

The code is shown below.

```
WomenWork=read.table("WomenWork.dat",header = TRUE)
y=as.vector(WomenWork$Work)
X=as.matrix(WomenWork[,2:9])
nPara=ncol(X)
mu=rep(0,nPara)
Sigma=10^2*diag(nPara)

LogPostLogistic <- function(beta,y,X,mu,Sigma){
  n=length(beta)
  ### log likelihood
  logLik <- sum( X%*%beta*y -log(1 + exp(X%*%beta)))
  ### log prior
  logPrior <- dmvnorm(beta, mean = mu, Sigma, log=TRUE)
  # return log posterior
  return(logLik + logPrior)
}

init_beta=rep(0,nPara)
OptimResults=optim(init_beta,LogPostLogistic,
                   gr=NULL,y,X,mu,Sigma,method=c("BFGS"),
                   control=list(fnscale=-1),hessian=TRUE)

postMode=OptimResults$par
J_inver <- -solve(OptimResults$hessian)
```

The numerical values for $J_y^{-1}(\tilde{\beta})$ and $\tilde{\beta}$ obtained from optim() function are shown below.

Table 3: The J inverse of beta_tilde

	Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
Constant	2.266023	0.003339	-0.065451	-0.011791	0.045781	-0.030293	-0.188748	-0.098024
HusbandInc	0.003339	0.000253	-0.000561	-0.000031	0.000141	-0.000036	0.000507	-0.000144
EducYears	-0.065451	-0.000561	0.006218	-0.000356	0.001896	-0.000003	-0.006135	0.001753
ExpYears	-0.011791	-0.000031	-0.000356	0.004352	-0.014249	-0.000134	-0.001469	0.000544
ExpYears2	0.045781	0.000141	0.001896	-0.014249	0.055579	-0.000330	0.003208	0.000512
Age	-0.030293	-0.000036	-0.000003	-0.000134	-0.000330	0.000718	0.005184	0.001095
NSmallChild	-0.188748	0.000507	-0.006135	-0.001469	0.003208	0.005184	0.151262	0.006769
NBigChild	-0.098024	-0.000144	0.001753	0.000544	0.000512	0.001095	0.006769	0.019972

Table 4: The posterior mode of beta

Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
0.6267288	-0.0197911	0.180219	0.1675667	-0.1445967	-0.0820656	-1.359133	-0.0246835

We fit a logistic regression using MLE by the glm() method to verify our results. The coefficients from glmModel are shown below.

Table 5: The coefficients of glmModel

Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
0.6443036	-0.0197746	0.1798806	0.1675127	-0.1443595	-0.0823403	-1.362502	-0.0254299

Compute an approximate 95% credible interval for NSmallChild

Because the approximate normal posterior in large samples $\theta|y \overset{approx}{\sim} N[\tilde{\theta}, J_y^{-1}(\tilde{\theta})]$ is a multivariate normal distribution, the marginal distribution of variable NSmallChild is a normal distribution with mean=-1.359, variance=0.151. We simulate 10000 draws from this normal distribution and obtain the 95% credible interval for the NSmallChild.

And we also use qnorm() function to calculate the 95% credible interval for the NSmallChild.

All results are shown in a table below.

Table 6: The 95% credible interval for the NSmallChild

	lower_bound	upper_bound
simulation	-2.130349	-0.6013771
qnorm()	-2.121411	-0.5968554

Would you say that this feature is an important determinant of the probability that a woman works?

The feature NSmallChild is an important determinant of the probability that a woman works because its corresponding coefficient -1.359 has the biggest absolute value in all the coefficients. And other coefficients have a big gap from it. Further more, its variance is only 0.151. Therefore NSmallChild is an important determinant.

We can verify it by p-value in the summary of glmMode.

(b) The predictive distribution of class for one woman

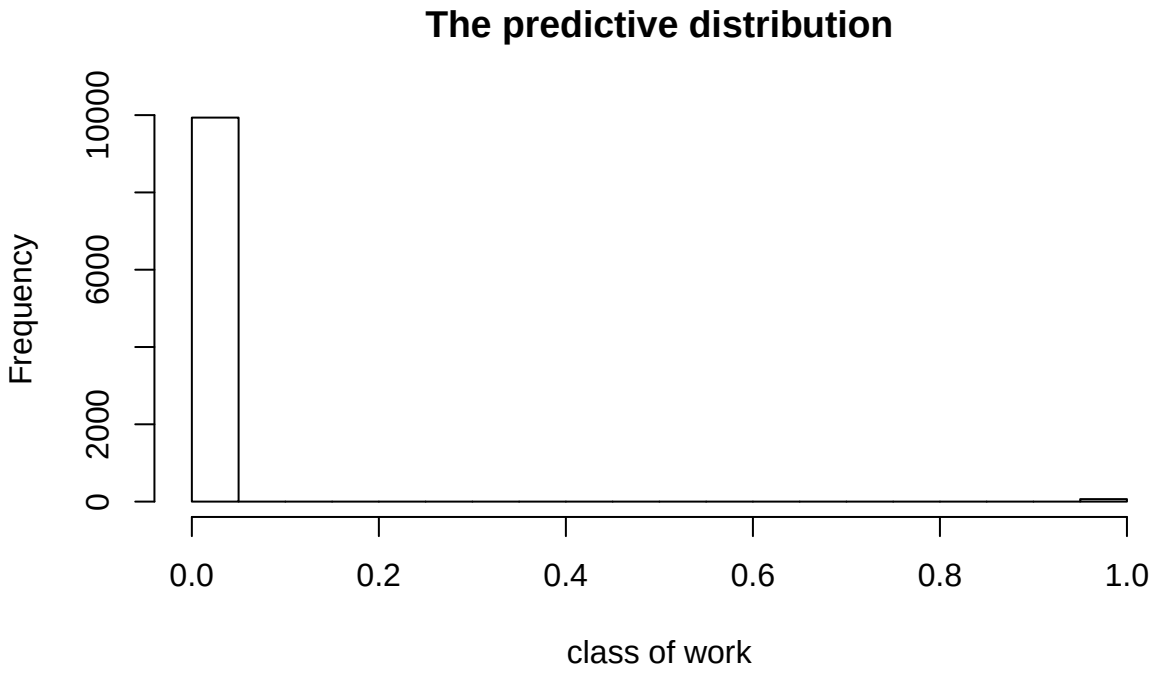
We simulate 10000 draws from normal approximation of the posterior distribution by using rmvnorm() function.

Then we put these simulations of coefficients into the formula

$Pr(y = 1|X) = \frac{exp(X^T\beta)}{1+exp(X^T\beta)}$

to calculate the probability that a woman works. Because this is a binary classification, if the porobability that a woman works is more than 0.5, we will classify that woman to the class 1 (work=1), otherwise class (work=0).

In this question, the X is a vector of values for 8 ordered variables, $X = (1, 10, 8, 10, 1, 40, 1, 1)^T$.



In the histogram, 9934 of 10000 predictions are calssified into class 0 (work=0) and only 66 into class 1 (work=1).

(c) The predictive distribution for the number of women that are working

Since the woman only will be calssified into 2 classes, we can think the class of work=1 as the success in a Bernoulli trial and the class of work=0 as the failure. And from 2(b) we can obtain the probability of success, p.

Now, we have the 10 independent experiments that are all subjected to this Bernoulli distribution, Bern(p).

The number of successes in this sequence of 10 independent experiments is a Binomial distribution Bin(n=10,p). Assume the number of successes is s, then the probability of getting exactly s successes is

$p(s) = \binom{n}{s} p^s (1-p)^{n-s}$.

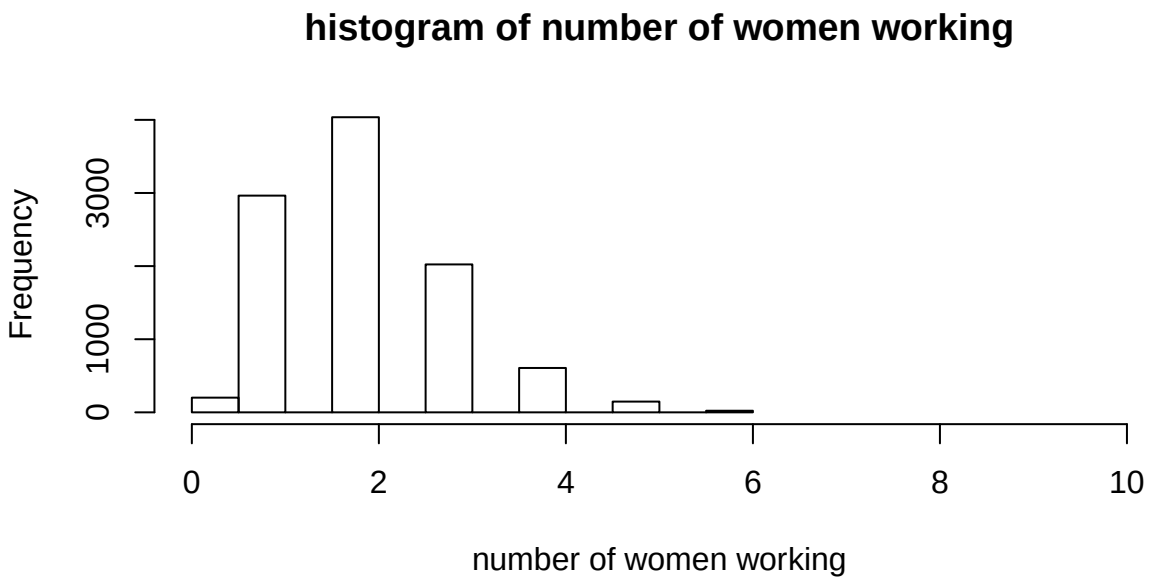
For each probability p obtained from 2(b) we calculate 11 probabilities from p(s=0) to p(s=10). The value of s corresponding to the biggest probability in 11 probabilities is the number of women that are working.

We implement this process to these 10000 probabilities from 2(b) and plot the histogram of the number of women that are working, which is the predictive distribution demanded.

The counts and corresponding probabilities of each number of women from 0 to 10 are shown in a table.

Table 7: The counts and probabilities for s from 0 to 10

Num	0	1	2	3	4	5	6	7	8	9	10
coun	201	2964	4036	2023	607	147	22	0	0	0	0
prob	0.0201	0.2964	0.4036	0.2023	0.0607	0.0147	0.0022	0	0	0	0


histogram of number of women working
Frequency
0
1000
3000
0
2
4
6
8
10
number of women working

Appendix

```
library(mvtnorm)
# Read the data from .txt file
TempLinkoping=read.delim("TempLinkoping.txt",header = TRUE,sep="\t")

## (a) Determining the prior distribution of the model parameters
X=cbind(rep(1,nrow(TempLinkoping)),TempLinkoping$time,TempLinkoping$time^2)
colnames(X)=c("inter","time","time2")
y=TempLinkoping$temp
hist(TempLinkoping$temp)
plot(TempLinkoping$time,TempLinkoping$temp)
### set the original values for hyperparameters
mu0=c(-10,100,-100)
omega0=diag(x=0.01,nrow = 3,ncol = 3)
v0=4
sigma2=1
nDraws=100
### simulate 100 draws from the joint prior and plot corresponding regression curves
plot_reg_curve <- function(mu0,omega0,v0,sigma2,nDraws){
  set.seed(12345)
  draw_chisq=rchisq(nDraws,df=v0)
  prior_sigma_draw=v0*sigma2/draw_chisq
  prior_beta_draw=matrix(0,nrow = nDraws,ncol = 3)
  for (i in 1:nDraws) {
    prior_beta_draw[i,]=rmvnorm(1,mean = mu0,
                                sigma =prior_sigma_draw[i]*solve(omega0))
  }

  plot(TempLinkoping$time,prior_beta_draw[1,1]+
         prior_beta_draw[1,2]*X[,2]+
         prior_beta_draw[1,3]*X[,3]+
         rnorm(1,mean=0,sd=sqrt(prior_sigma_draw[1])),
       xlab = "time",ylab = "temp",
       type = "l",ylim = c(-15,20))
  for (i in 2:nDraws) {
    lines(TempLinkoping$time,prior_beta_draw[i,1]+
            prior_beta_draw[i,2]*X[,2]+
            prior_beta_draw[i,3]*X[,3]+
            rnorm(1,mean=0,sd=sqrt(prior_sigma_draw[i])))
  }

}
plot_reg_curve(mu0,omega0,v0,sigma2,nDraws)
plot_reg_curve(mu0,omega0,v0,sigma2=0.01,nDraws)
mu0=c(-10,105,-100)
plot_reg_curve(mu0,omega0,v0,sigma2=0.01,nDraws)
## (b) Simulate from the joint posterior distribution
mu0=c(-10,105,-100)
omega0=diag(x=0.01,nrow = 3,ncol = 3)
v0=4
sigma2=0.01
nDraws=10000
```

```
sim_join_post=function(mu0,omega0,v0,sigma2,nDraws,X,y){
  n=length(y)
  vn=v0+n
  beta_hat=solve(t(X)%*%X)%*%t(X)%*%y
  mu_n=solve(t(X)%*%X+omega0)%*%(t(X)%*%X%*%beta_hat+omega0%*%mu0)
  omega_n=t(X)%*%X+omega0
  sigma_n2=(v0*sigma2+t(y)%*%y+t(mu0)%*%omega0%*%mu0-
              t(mu_n)%*%omega_n%*%mu_n)/vn
  set.seed(12345)
  ### Simulate from the joint posterior distribution
  draw_chisq=rchisq(nDraws,df=vn)
  post_sigma_draw=vn*sigma_n2/draw_chisq
  post_beta_draw=matrix(0,nrow = nDraws,ncol = 3)
  for (i in 1:nDraws) {
    post_beta_draw[i,]=rmvnorm(1,mean = mu_n,
                               sigma =post_sigma_draw[i]*solve(omega_n))
  }
  hist(post_beta_draw[,1],breaks = 100)
  hist(post_beta_draw[,2],breaks = 100)
  hist(post_beta_draw[,3],breaks = 100)
  hist(post_sigma_draw,breaks = 100)
  return(post_beta_draw)
}

res=sim_join_post(mu0,omega0,v0,sigma2,nDraws,X,y)

post_median <- function(result){
  den_x=density(result)$x
  den_y=density(result)$y
  prob=data.frame(x=den_x,y=den_y,cum=rep(0,length(den_x)))
  prob$cum=cumsum(prob$y)/sum(prob$y)
  pos=min(which(prob$cum>=0.5))
  post_medi=prob$x[pos]
  return(post_medi)
}

post_med_beta0=post_median(res[,1])
post_med_beta1=post_median(res[,2])
post_med_beta2=post_median(res[,3])

f_time=function(beta0,beta1,beta2,X){
  temp=beta0+beta1*X[,2]+beta2*X[,3]
  return(temp)
}

plot(TempLinkoping$time,TempLinkoping$temp,
     xlab = "time",ylab = "temperature")
post_med_reg=f_time(post_med_beta0, post_med_beta1,post_med_beta2,X)
lines(X[,2],post_med_reg,col="red")
post_cre_low_reg=f_time(quantile(res[,1],0.025),quantile(res[,2],0.025),
                        quantile(res[,3],0.025),X)
lines(X[,2],post_cre_low_reg,col="blue")
post_cre_up_reg=f_time(quantile(res[,1],0.975),quantile(res[,2],0.975),
```

```
                        quantile(res[,3],0.975),X)
lines(X[,2],post_cre_up_reg,col="blue")

## (c) Simulate from the posterior distribution of x_tilde
x_tilde=-res[,2]/(2*res[,3])
hist(x_tilde,breaks = 100)

# 2. Posterior approximation for classification with logistic regression

## input the data

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

## (a) Approximate the posterior distribution of the 8-dim parameter vector

glmModel=glm(Work~0+., data=WomenWork,family=binomial)

y=as.vector(WomenWork$Work)
X=as.matrix(WomenWork[,2:9])
nPara=ncol(X)
mu=rep(0,nPara)
Sigma=10^2*diag(nPara)

LogPostLogistic <- function(beta,y,X,mu,Sigma){
  n=length(beta)
  ### log likelihood
  logLik <- sum( X%*%beta*y -log(1 + exp(X%*%beta)))
  ### log prior
  logPrior <- dmvnorm(beta, mean = mu, Sigma, log=TRUE)

  # return log posterior
  return(logLik + logPrior)
}

init_beta=rep(0,nPara)
OptimResults=optim(init_beta,LogPostLogistic,
                   gr=NULL,y,X,mu,Sigma,method=c("BFGS"),
                   control=list(fnscale=-1),hessian=TRUE)

postMode=OptimResults$par
new1=as.data.frame(t(postMode))
colnames(new1)=colnames(WomenWork)[2:9]
J_inver <- -solve(OptimResults$hessian)
new2=as.data.frame(J_inver)
colnames(new2)=colnames(WomenWork)[2:9]
rownames(new2)=colnames(WomenWork)[2:9]
approxPostStd <- sqrt(diag(J_inver))
new3=as.data.frame(t(glmModel$coefficients))
colnames(new3)=colnames(WomenWork)[2:9]
postMode[7]
J_inver[7,7]
sim_NSmall=rnorm(10000,mean = postMode[7],sd=sqrt(J_inver[7,7]))
hist(sim_NSmall,breaks = 100)
```

```r
low_cre_NSm=quantile(sim_NSmall,0.025)
up_cre_NSm=quantile(sim_NSmall,0.975)


## (b) Simulate from the predictive distribution of the response variable in a logistic regression

simul_all=rmvnorm(10000,mean = postMode,sigma = J_inver)
vec_x=c(1,10,8,10,1,40,1,1)
prob_work=rep(0,10000)
for (i in 1:10000) {
  prob_work[i]=exp(vec_x%*%simul_all[i,])/(1+exp(vec_x%*%simul_all[i,]))
}
class_work=ifelse(prob_work>0.5,1,0)
hist(prob_work,breaks = 100)
hist(class_work)

## (c) The predictive distribution for the number of women that are working

num_of_women=rep(0,11)
class_num_work=rep(0,10000)
for (i in 1:10000) {
 for (j in 0:10) {
   num_of_women[j+1]=choose(10,j)*prob_work[i]^j*(1- prob_work[i])^(10-j)
 }
  class_num_work[i]=which.max(num_of_women)-1
}
dis_prob=rep(0,11)
for (i in 0:10) {
  dis_prob[i+1]=length(which(class_num_work==i))
}

hist(class_num_work,xlim = c(0,10))
```