Lab 2 Ravinder

Ravinder Reddy Atla

4/21/2021

1. Linear and polynomial regression

```
library(mvtnorm)
```

```
# Scaled Inverse chi square distribution
pdfinvchisquare <- function(x,n,tau_sq){
  res <- (((tau_sq*(n-1))/2)^(n-1)/2 * exp(-(tau_sq*(n-1))/(2*x)))/((x^(1+(n-1)/2)) * gamma((n-1)/2))
  return(res)
}

# Function for sampling from inverse chi square distribution
rinvchisquare <- function(num_draws, n, tau_sq){
  set.seed(1234)
  x <- rchisq(num_draws,df = n-1)
  x_inv <- ((n-1)*tau_sq)/x
  return(x_inv)
}
```

```
# Prior parameters initialization
sisquare0 <- 1
v0 <- 4
mu0 <- c(-10, 100, -100)
si0 <- 0.01*diag(3)
si0_inv <- solve(si0)
num_draws = 1000
# Prior hierarchy

# Prior 1: si^2
var_prior <- rinvchisquare(num_draws, v0, sisquare0)

# Prior 2: beta/si^2
beta_prior <- matrix(NA, nrow = num_draws, ncol = 3)
temp <- c()
for(i in 1:num_draws){
  beta_prior[i,] <- rmvnorm(1,mean = mu0, sigma = var_prior[i]*si0_inv)
  #temp[i] <-
}
```

2. Posterior approximation for classification with logistic regression

```r
ww_data <- read.table('WomenWork.dat', header = TRUE)
rows <- nrow(ww_data)
cols <- ncol(ww_data)
```

```r
y <- as.matrix(ww_data[1])
X <- as.matrix(ww_data[,2:cols])
```

```r
params <- dim(X)[2]
mu <- as.matrix(rep(0,params))
tau = 10
Sigma = (tau^2)*diag(params)

LogPostLogistic <- 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)
}

initValue <- matrix(0,params,1)

OptimRes <- optim(initValue,
                  LogPostLogistic, gr=NULL, y, X, mu, Sigma, method=c("BFGS"),
                  control=list(fnscale=-1), hessian=TRUE)
```

```r
print('Posterior Mode: ')
```

```
## [1] "Posterior Mode: "
```

```r
print(OptimRes$par)
```

```
##             [,1]
## [1,]  0.62672884
## [2,] -0.01979113
## [3,]  0.18021897
## [4,]  0.16756670
## [5,] -0.14459669
## [6,] -0.08206561
## [7,] -1.35913317
## [8,] -0.02468351
```

```r
print('Inverse of hessian matrix')
```

```
## [1] "Inverse of hessian matrix"
```

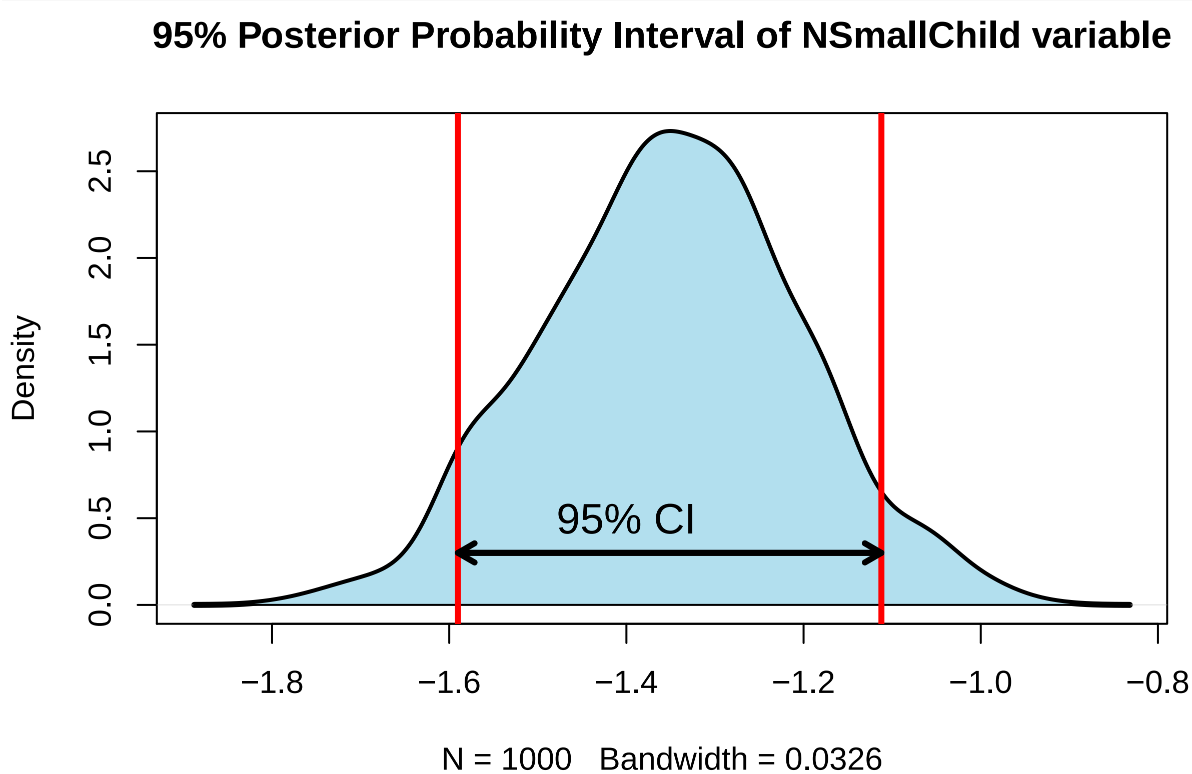
```r
inversehessian <- solve(OptimRes$hessian)
print(inversehessian)
```

```
##              [,1]          [,2]          [,3]          [,4]          [,5]
## [1,] -2.266022568 -3.338861e-03  6.545121e-02  1.179140e-02 -0.0457807243
## [2,] -0.003338861 -2.528045e-04  5.610225e-04  3.125413e-05 -0.0001414915
## [3,]  0.065451206  5.610225e-04 -6.218199e-03  3.558209e-04 -0.0018962893
## [4,]  0.011791404  3.125413e-05  3.558209e-04 -4.351716e-03  0.0142490853
## [5,] -0.045780724 -1.414915e-04 -1.896289e-03  1.424909e-02 -0.0555786706
```

```
## [6,]  0.030293450  3.588562e-05  3.240448e-06  1.340888e-04  0.0003299398
## [7,]  0.188748354 -5.066847e-04  6.134564e-03  1.468951e-03 -0.0032082535
## [8,]  0.098023929  1.444223e-04 -1.752732e-03 -5.437105e-04 -0.0005120144
##               [,6]          [,7]          [,8]
## [1,]  3.029345e-02  0.1887483542  0.0980239285
## [2,]  3.588562e-05 -0.0005066847  0.0001444223
## [3,]  3.240448e-06  0.0061345645 -0.0017527317
## [4,]  1.340888e-04  0.0014689508 -0.0005437105
## [5,]  3.299398e-04 -0.0032082535 -0.0005120144
## [6,] -7.184611e-04 -0.0051841611 -0.0010952903
## [7,] -5.184161e-03 -0.1512621814 -0.0067688739
## [8,] -1.095290e-03 -0.0067688739 -0.0199722657
```

```r
beta_post <- OptimRes$par
names(beta_post) <- colnames(ww_data[,2:cols])
approx_par_NSC <- rnorm(1000,beta_post['NSmallChild'],-inversehessian[7,7])
lowerInterval <- quantile(approx_par_NSC,0.05)
upperInterval <- quantile(approx_par_NSC, 0.95)
```

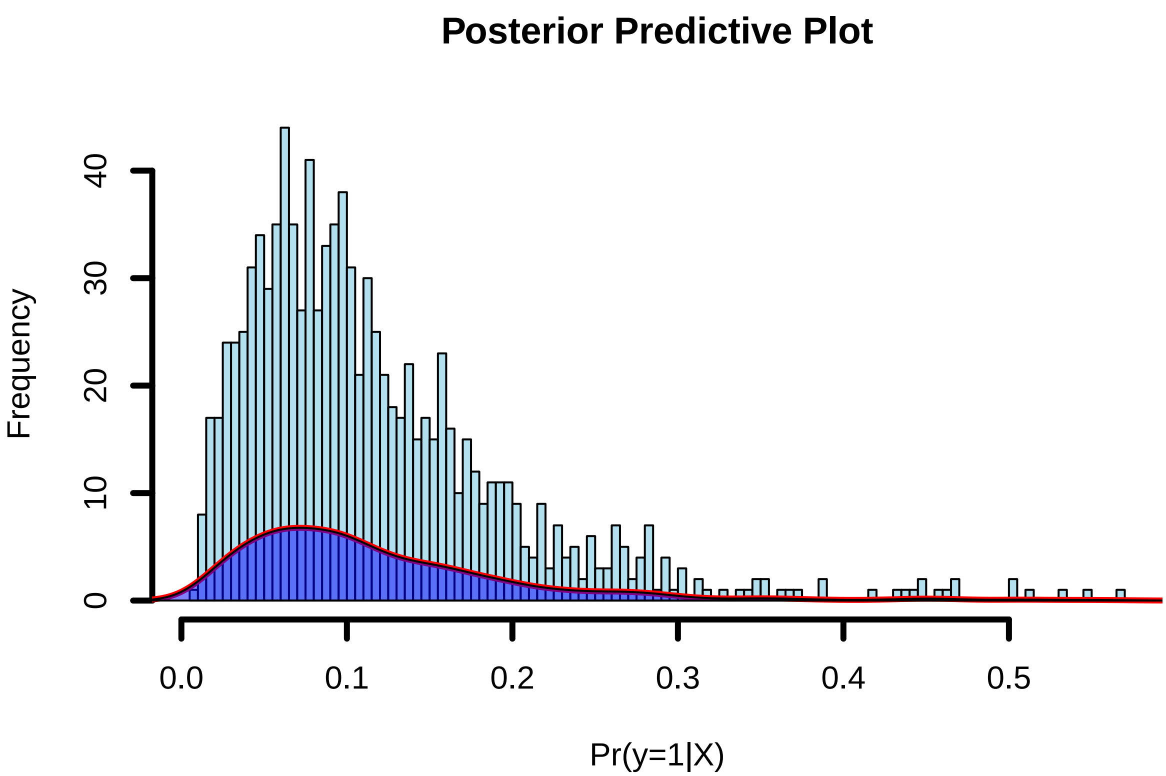
```r
plot(density(approx_par_NSC),lwd = 3,main = '95% Posterior Probability Interval of NSmallChild variable
polygon(density(approx_par_NSC), col = 'lightblue2')
abline(v = lowerInterval, col = 'red', lwd = 3)
abline(v = upperInterval, col = 'red', lwd = 3)
arrows(lowerInterval,0.3,upperInterval,0.3,length = 0.1,col = 'black',lwd = 3)
arrows(upperInterval,0.3,lowerInterval,0.3,length = 0.1,col = 'black',lwd = 3)
text(-1.4,0.5, '95% CI', lwd = 3,cex = 1.3)
```



b.

```
posteriorPredictive <- function(x, beta_post, inversehessian){
  post_sample <- rmvnorm(1, beta_post,-inversehessian)
  #print(x)
  #print(x %*% t(post_sample))
  logist_prob <- (exp(x %*% t(post_sample)))/(1 + exp(x %*% t(post_sample)))
  #print(logist_prob)
  return(logist_prob)
}
```

```
x <- c(1,13, 8, 11, (11/10)^2, 37, 2, 0)
nsamples = 1000
post_predict <- c(rep(0,nsamples))
for(i in 1:nsamples){
  post_predict[i] <- posteriorPredictive(x, beta_post, inversehessian)
}
#print(post_predict)
hist(post_predict, breaks = 100,
     col = 'lightblue2',lwd = 3,
     xlab = 'Pr(y=1|X)',
     main = 'Posterior Predictive Plot')
lines(density(post_predict), col = 'red',lwd = 3)
polygon(density(post_predict), col = rgb(red = 0, green = 0, blue = 1, alpha = 0.5))
```



c.

```r
posteriorPredictiveBinomial <- function(x, beta_post, inversehessian){
  post_sample <- rmvnorm(1, beta_post,-inversehessian)
  logist_prob1 <- (exp(x %*% t(post_sample)))/(1 + exp(x %*% t(post_sample)))
  logist_prob <- sum(rbinom(1,8,logist_prob1))
  return(logist_prob)
}
```

```r
test_data <- matrix(x,nrow=8, ncol=8,byrow = TRUE)
nsamples = 1000
post_predict <- c(rep(0,nsamples))
for(i in 1:nsamples){
  post_predict[i] <- posteriorPredictiveBinomial(test_data, beta_post, inversehessian)
}
#barplot(post_predict,col = 'cadetblue')
#plot(density(post_predict))
hist(post_predict,breaks = 100,col = 'lightblue2',
     xlab = 'Number of women who works',
     main = 'Histogram of number of women who works ')
```