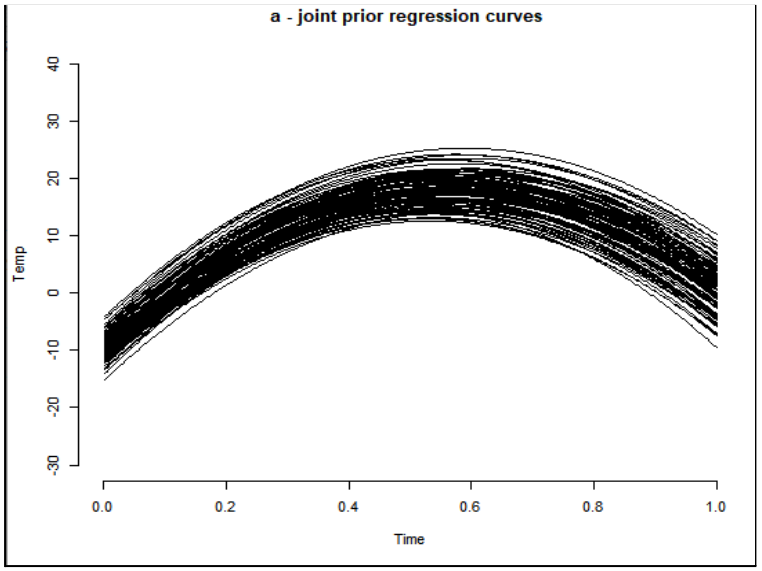
TDDE07 Bayesian Learning Lab 2

Erik Jareman - erija971
Ludvig Knast - ludkn080

1.a

The collection of curves looked reasonable. The sigma0^2 was adjusted from 2 to 1 to tighten the dispersion slightly, and the last value of my0 was adjusted from -100 to -90 to make the transition "between years" a bit more smooth and to make the observed maximum temperature occur slightly after "mid-year".



1.b

Simulate draws from joint posterior distribution of B0, B1, B2:

```
getBetas = function(my, sigma2, omega0Inv) {
  return(rmvnorm(1, mean=my, sigma=sigma2*omega0Inv))
}
```

Get values from the regression model from betas:

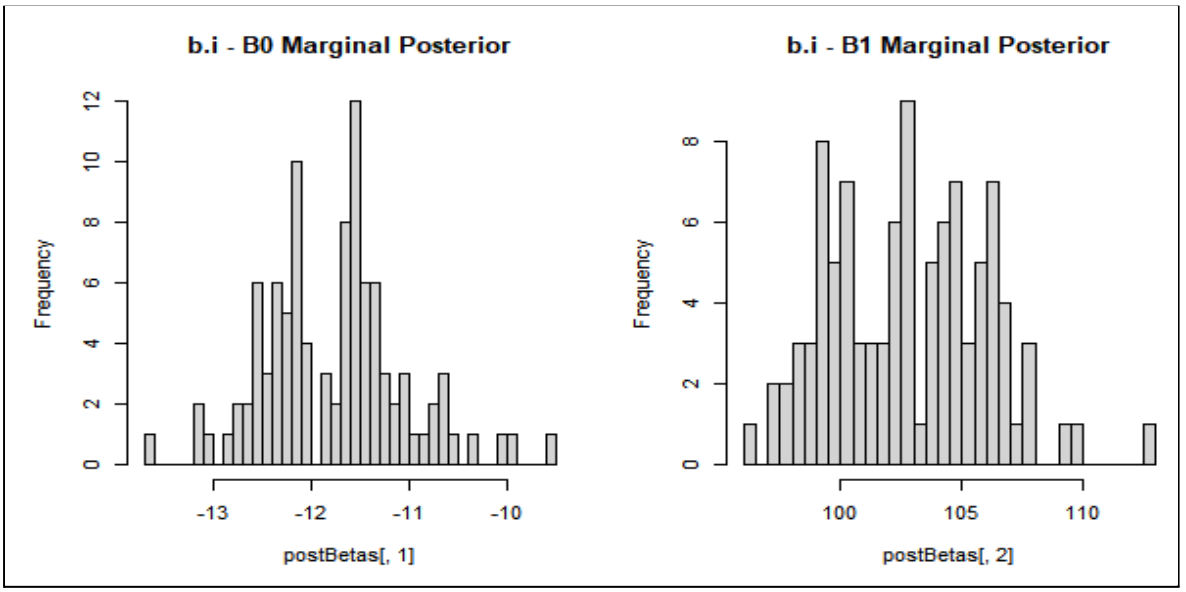
```
getReg = function(betaVals, time) {
  return(betaVals[1] + betaVals[2]*time + betaVals[3]*(time^2))
}
```

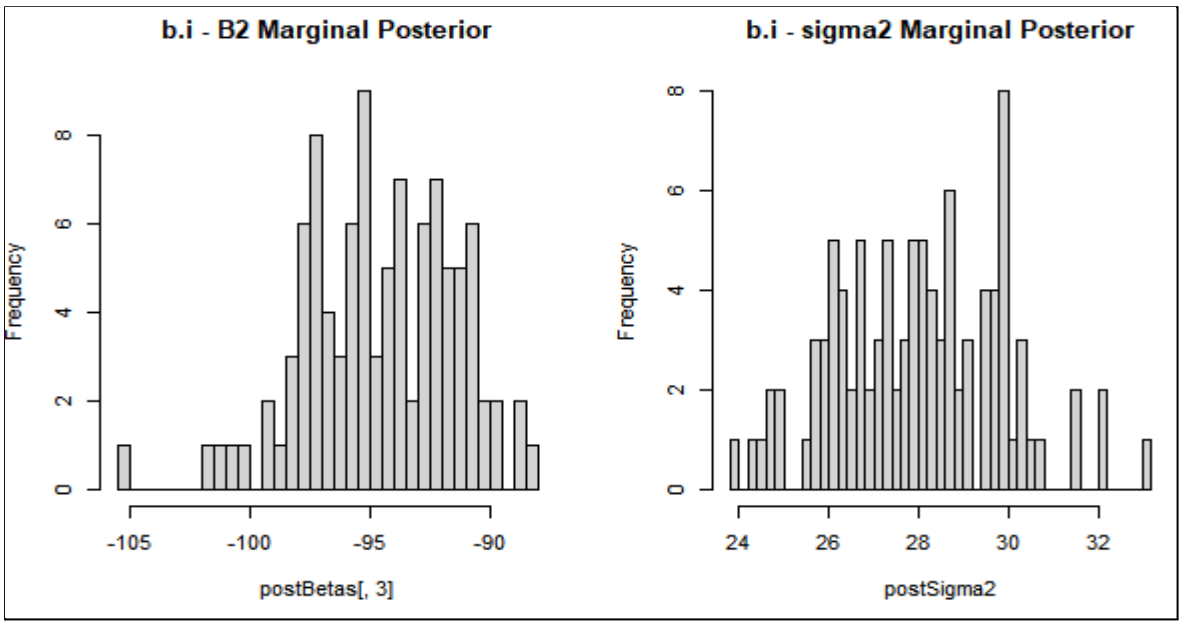
Code to simulate draws from the joint posterior distribution:

```
# --- b ---
X = as.matrix(data.frame(const=1, t=data$time, t2=data$time^2))
y = data$temp
n = length(data$time)
nv = v0 + n
nOmega = t(X)%*%X+omega0
nOmegaInv = solve(nOmega)
betaHat = solve(t(X)%*%X)%*%t(X)%*%y
nMy = solve(t(X)%*%X+omega0)%*%(t(X)%*%X%*%betaHat+omega0%*%my0)
nSigma2 = as.vector((v0*sigma02 + (t(y)%*%y+t(my0)%*%omega0%*%my0-t(nMy)%*%nOmega%*%nMy))/nv)

postSigma2 = nv * nSigma2 / draw
postBetas = matrix(0, nDraws, 3)
for (i in 1:nDraws) {
  postBetas[i,] = getBetas(nMy, postSigma2[i], nOmegaInv)
}
```

1.b.i

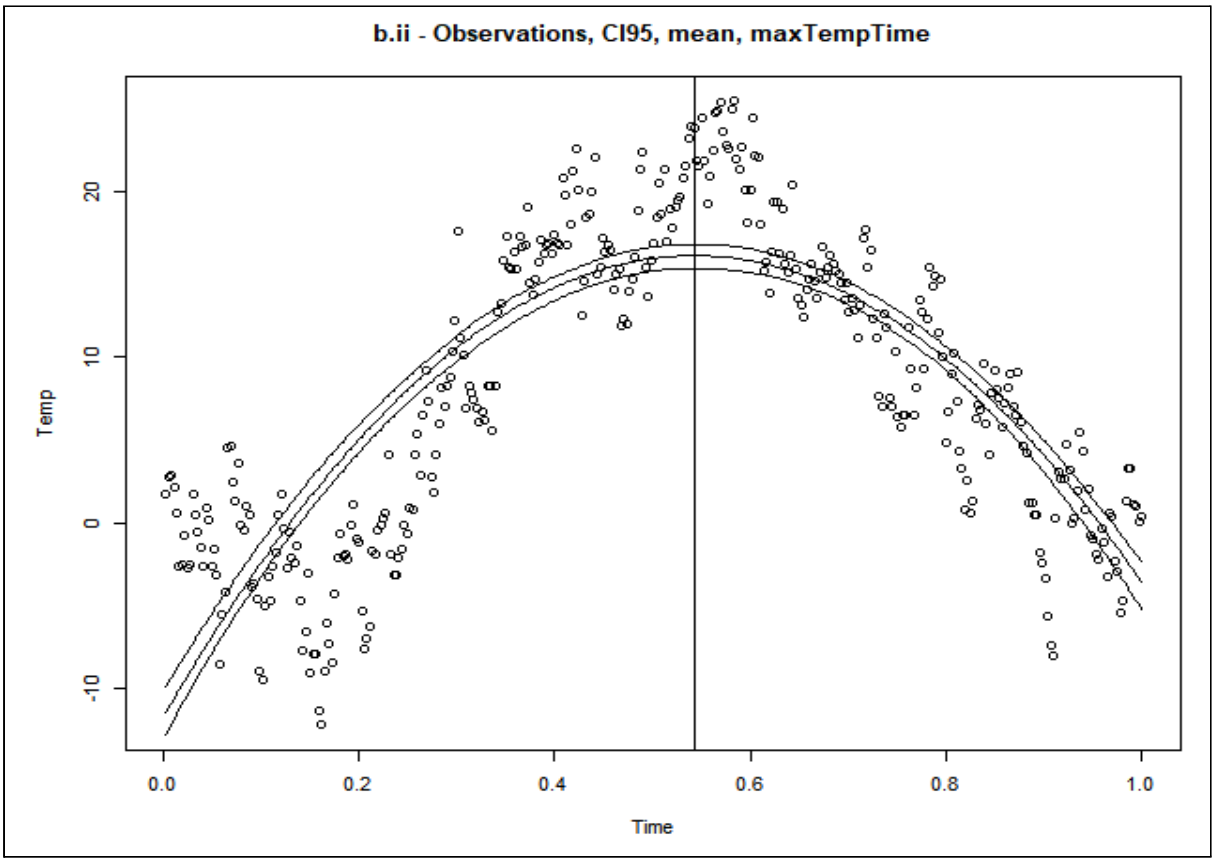
Histograms for each marginal posterior of the parameters:





1.b.ii

Scatter plot of observed data with overlay for 90% equal tail posterior probability interval, posterior mean, and the time with the highest expected temperature simulated from the posterior distribution:



The posterior probability interval does not contain most of the data points. Even though the best betas are selected to fit the data, the model is not complex enough to fit observed data in a suitable way. This means that the model does not manage to describe reality in a perfect way.

1.c

The highest expected temperature time, simulated from the posterior distribution, was calculated as follows:

```
# due to negative quadratic polynomial function we have unique max where f'=0.
# f'=b2+2*B3*time=0 -> time=-B2/(2*B3)
maxTempTime = -postBetas[,2]/(2*postBetas[,3])
abline(v=mean(maxTempTime))
```

In order to keep the figure in *1.b.ii* uncluttered, only the mean of the calculated maximum times was plotted.

1.d

```
# --- d ---
'To avoid overfitting, my0 and omega0 can be set in the following ways:
 - my0 = 0: this parameter can be set to 0 to make the parameters
   stay closer to 0.
 - omega0 = lambda*I: the parameter omega0 can be set this way
   to give a smaller spread in the posterior distrubution. This
   way the parameters stay somewhat close to my0'
```

2

2a

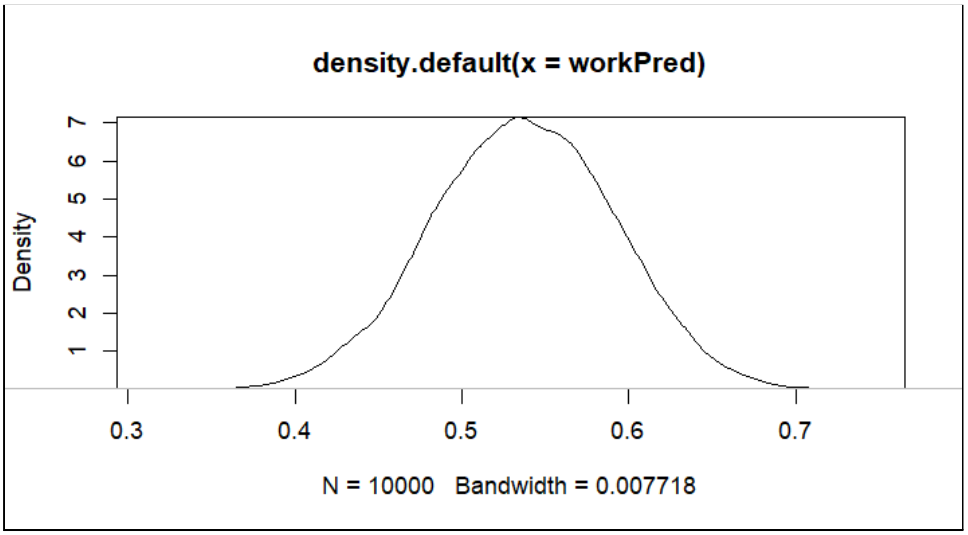
The interval does not contain 0 which indicates that this feature is of importance for the probability that a woman works.

The comparison between posterior means and max likelihood shows similar values.

```
[1] "The posterior mode is:"
   Constant  HusbandInc   EducYears    ExpYears         Age NSmallChild   NBigChild
 0.99296647 -0.03434846  0.17941369  0.12306096 -0.07279127 -1.62276255 -0.08389155
[1] "Cov:"
              [,1]          [,2]          [,3]          [,4]          [,5]          [,6]          [,7]
[1,]  2.5292571636  3.936988e-03 -7.750703e-02  8.893107e-04 -3.409992e-02 -0.2281035834 -1.137004e-01
[2,]  0.0039369876  3.914698e-04 -8.068724e-04 -1.046743e-06 -5.158984e-05  0.0011499161 -6.395372e-05
[3,] -0.0775070317 -8.068724e-04  7.341456e-03  6.095302e-05  4.960140e-05 -0.0080243160  1.925617e-03
[4,]  0.0008893107 -1.046743e-06  6.095302e-05  9.005762e-04 -2.492000e-04 -0.0009929632  6.121384e-04
[5,] -0.0340999236 -5.158984e-05  4.960140e-05 -2.492000e-04  8.071660e-04  0.0060956597  1.278749e-03
[6,] -0.2281035834  1.149916e-03 -8.024316e-03 -9.929632e-04  6.095660e-03  0.1918363858  9.420882e-03
[7,] -0.1137003876 -6.395372e-05  1.925617e-03  6.121384e-04  1.278749e-03  0.0094208819  2.221618e-02
[1] "Approximated standard deviations:"
   Constant  HusbandInc   EducYears    ExpYears         Age NSmallChild   NBigChild
 1.59036385  0.01978559  0.08568230  0.03000960  0.02841067  0.43799131  0.14905095
[1] "Lower bound -2.48121 and the upper bound -0.764315"
[1] "Comparison:"
   Constant  HusbandInc   EducYears    ExpYears         Age NSmallChild   NBigChild
 1.12242734 -0.03425216  0.17650532  0.12317305 -0.07475060 -1.64598118 -0.08973248
   Constant  HusbandInc   EducYears    ExpYears         Age NSmallChild   NBigChild
 0.99296647 -0.03434846  0.17941369  0.12306096 -0.07279127 -1.62276255 -0.08389155
```

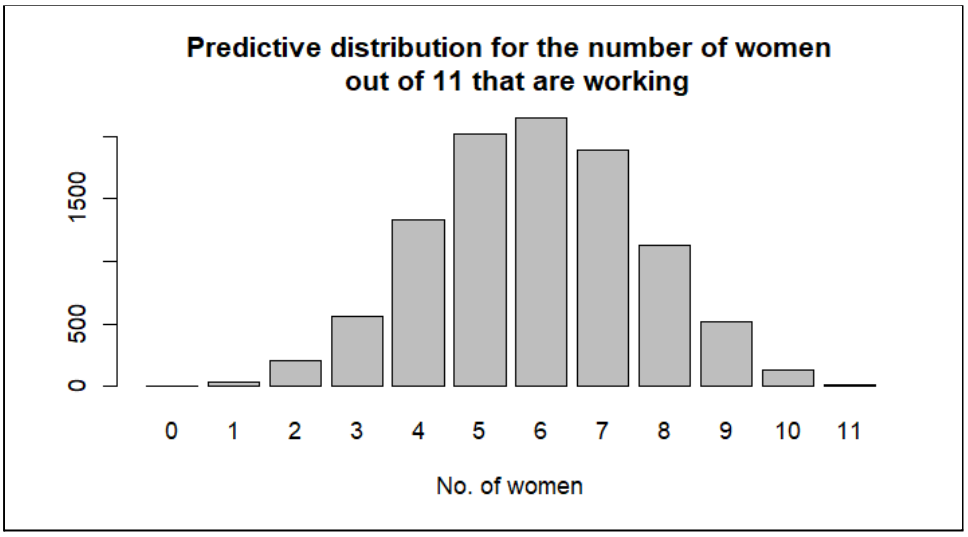
```r
library("mvtnorm")

women = read.table("WomenAtWork.dat", header=TRUE)

# a)
nVar = nrow(women)
y = women$Work
X = as.matrix(women[,2:8])
nAtr <- ncol(X)
covNames=names(women[,2:8])

mu0 <- as.matrix(rep(0, nAtr))
tao = 5
sigma_pr = tao^2*diag(nAtr)

LogPostLogistic <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  #if (abs(logLik) == Inf) logLik = -20000; # Likelihood is not finite, stear
  #the optimizer away from here!
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);

  return(logLik + logPrior)
}

# Initial values for beta
initVals <- rnorm(nAtr)

optiRes <- optim(initVals,LogPostLogistic,gr=NULL,y,X,mu0,sigma_pr,method=c("BFGS")
                 ,control=list(fnscale=-1),hessian=TRUE)

mode = optiRes$par
cov = -solve(optiRes$hessian)
names(mode) = covNames
approx_PostStd = sqrt(diag(cov))
names(approx_PostStd) = covNames
print("The posterior mode is:")
print(mode)
print("Cov:")
print(cov)
print("Approximated standard deviations:")
print(approx_PostStd)

# Marginal distribution for nSmallChild
NSC_mode = as.numeric(mode["NSmallChild"])
NSC_std = as.numeric(approx_PostStd["NSmallChild"])
credInterval = qnorm(p=c(0.025, 0.975), mean=NSC_mode, sd=NSC_std)
print(paste("Lower bound",
            round(credInterval[1], 6), "and the upper bound",
            round(credInterval[2], 6)))

# Control calculations
glmModel = glm(Work ~ 0+., data = women, family = binomial)
print("Comparison:")
print(glmModel$coefficients)
print(mode)
```

2b




```r
# b)
makePred = function(data, nDraws, mean, sigma) {
  betaPred = rmvnorm(nDraws, mean=mean, sigma=sigma)
  pred = exp(t(data)%*%t(betaPred)) / (1 + exp(t(data)%*%t(betaPred)))
  return(pred)
}

woman=c(1, 20, 12, 8, 43, 0, 2)
nDraws = 10000
workPred = makePred(woman, nDraws, mode, cov)
plot(density(workPred))
```

2c




```r
# c)
makeMultiPred = function(data, nDraws, mean, sigma, n) {
  multiplePred=c()
  for (i in 1:nDraws) {
    betaPred = makePred(data, 1, mean, sigma)
    multiplePred=c(multiplePred, rbinom(1, n, betaPred))
  }
  barplot(table(multiplePred), main=paste("Predictive distribution for the number of women
  out of 11 that are working"), xlab="No. of women")
}

makeMultiPred(woman, 10000, mode, cov, 11)
```