Lab2

## TDDE07

## Computer Lab 2

# Authors:

# Philip Kantola phika529

# Arun Uppugunduri aruup817

# Assignment 1

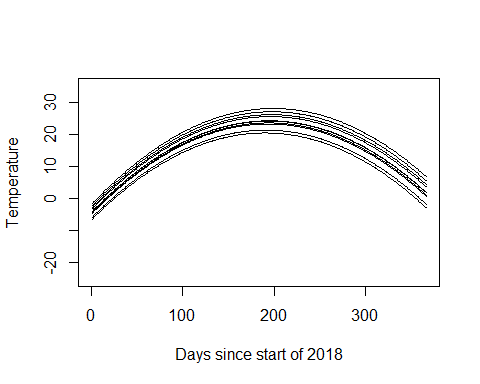
# 1a

We started by drawing 10 draws from our given scaled chi inverse function. For each o

check(These are draws from our marginal distribution of sigma.)

We simulated 10 different beta-vectors with our prior hyperparameters and plotted them (see plot below). The low value of the values in precision matrix means that we are unsure on our prior, this is the reason that our plotted curves gives very different results. Altering v0, sigma0 and our precision matrix will render the same result in the end, the variance of the beta parameters. The hyperparameters which we can change to make the curves fit our prior believes are essentially the rho vector. In the plot below we used the default values of [-10,100,-100]. Based on historical data we saw that these given values gave us what we belived to be the best distribution.

## Warning: package 'mvtnorm' was built under R version 3.6.2



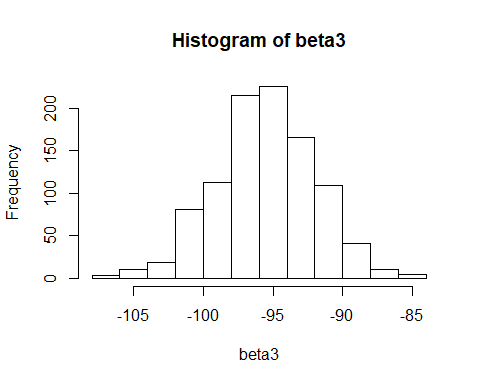
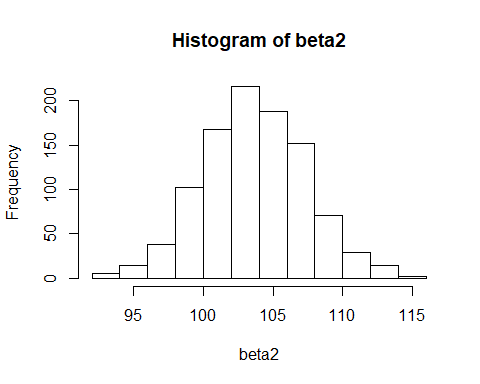
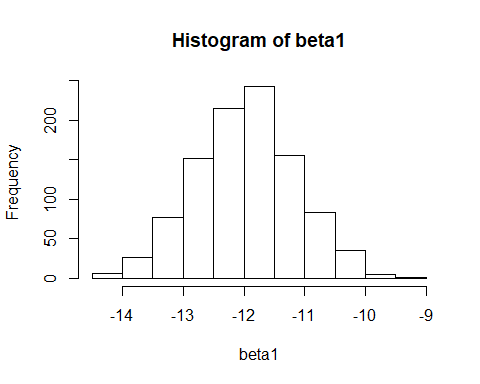
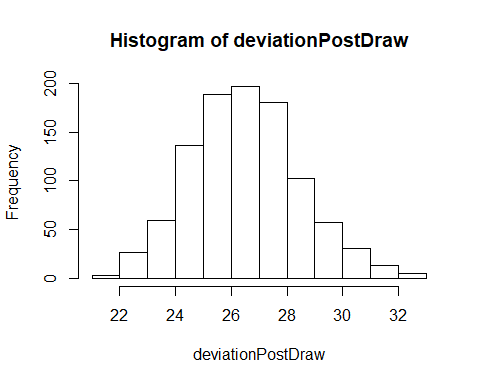
# b

*Write a program that simulates from the joint posterior distribution of beta0, beta1, beta2 and sigma^2 Plot the marginal posteriors for each parameter as a histogram.*

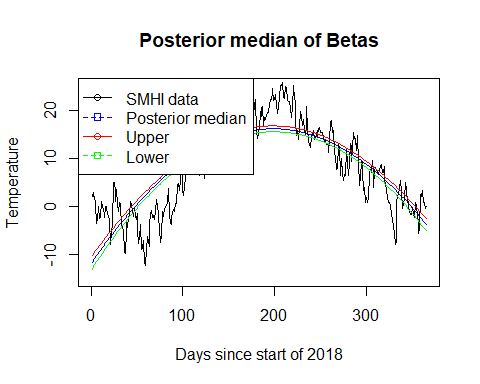
*Also produce a figure with the scatterplot of the temperature data and overlay it with the curve for the posterior median of the regression function*

*Also overlay curves for the lower 2.5% and upper 97,5% credible interval for f(time)*

From the histograms we have obtained all parameters was shown to have distributions similar to the normal distribution. This was expected since the betas was drawn from a normal distribution and since sigma^2 comes from a scaled inverse chi^2 distribution which can be approximated to a normal distribution if n is big. (we chose n as 1000 in this case)

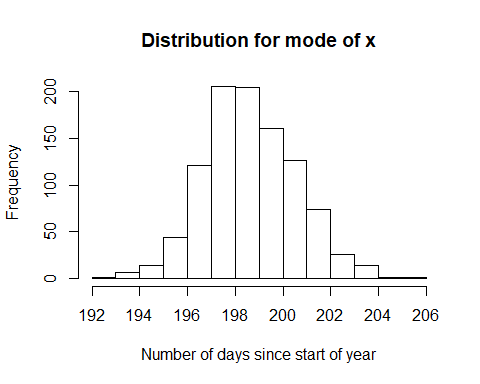


We took the median of f(x) computed for all days and plotted the resulting curve together with the actual data. In the same plot we also plotted the 2.5 and 97.5 percentile of f(x), the credible interval. See plot below. The credible interval does not cover 95% of the data which it is not meant to do, since the interval will cover the 95% probability interval of the curve which describes the weather.

 #1c

*It is of interest to locate the time with the highest expected temperature (that is, the time where f(time) is maximal). Let’s call this value xmax. Use the simulations in b) to simulate from the posterior distribution of xmax.*

Seen below is the histogram for the posterior distribution of the posterior mode. Based on the result it appears to be normally distributed. Looking at the plotted curves for f(time) the historgram looks reasonable showing a max temperature around time = 199 days.



# 1d

*Say that you want to estimate a polynomial model of order 7 but suspect that higer order terms might not be needed and worry about overfitting. Suggest a suitable prior that mitgates this potential problem.*

This is solved by adding a penalizing factor lambda adding more shrinkage and gaining more precision in the model. Since lambda is unknown we simply assign a prior to it which will then be have the distibuted acording to the scaled inverse chi square with parameters n\_ and lambda\_0. Slide 91. Omega\_0 is specified as lamda\*Identity matrix and using hyperparameer my\_0 = (0,0,0) (Slide 91).

the prior is as follows: beta|sigma^2 ~ N(0,sigma^2/lambda). We We then say that Omega\_0 = lambda\*identity vector and we get a joint prior of Beta, Sigma^2, lambda => p(B, sigma^2, lambda| data).

# Assignment 2

*2a*

Since the prior for beta is normal and the likelihood a product of the logistic regression the posterior is difficult to compute, instead we obtained a large sample approximate posterior for the beta vector. We first optimized over the log-logistic regression expression and obtained the following numerical values for the beta mode and inverted Hessian.(first row is mode for beta vector and the following rows is the Hessian)

## [1] 0.62672884 -0.01979113 0.18021897 0.16756670 -0.14459669 -0.08206561  
## [7] -1.35913317 -0.02468351

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 37.697939 793.0147 469.6456 367.20017 53.450998 1582.5424  
## [2,] 793.014693 21726.2184 10340.0749 7713.59763 1095.613234 33614.3881  
## [3,] 469.645561 10340.0749 6070.3314 4544.61312 648.975117 19684.1213  
## [4,] 367.200173 7713.5976 4544.6131 5345.13635 981.781722 16227.6889  
## [5,] 53.450998 1095.6132 648.9751 981.78172 214.577828 2472.3781  
## [6,] 1582.542441 33614.3881 19684.1213 16227.68889 2472.378069 68845.6824  
## [7,] 9.469258 191.1836 127.1882 74.89198 7.675926 321.3491  
## [8,] 48.177234 995.4926 584.1196 373.17114 42.890848 1893.1488  
## [,7] [,8]  
## [1,] 9.469258 48.17723  
## [2,] 191.183648 995.49260  
## [3,] 127.188152 584.11957  
## [4,] 74.891981 373.17114  
## [5,] 7.675926 42.89085  
## [6,] 321.349083 1893.14881  
## [7,] 11.923173 12.79632  
## [8,] 12.796324 123.04393

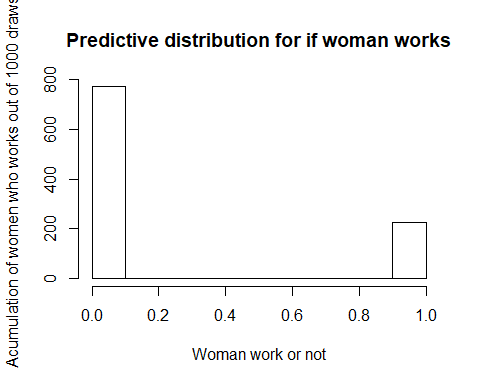
We approximated an 95% credible interval for the variable NSmallChild (or rather, for the coefficient) and obtained the range below. We compared with GLM-model and concluded that this is a reasonable result. the effect of the number of small childs prove to be very large, if a woman has two small children the probability of her working becomes very small, this is obvious when comparing to the weight of the other variables from GLM. (About the values below, the first row represents the credible interval, the second row are the estimated beta-values obtained from GML)

## 2.5% 97.5%   
## -2.1296887 -0.5759195

## (Intercept) Constant HusbandInc EducYears ExpYears ExpYears2   
## 0.64430363 NA -0.01977457 0.17988062 0.16751274 -0.14435946   
## Age NSmallChild NBigChild   
## -0.08234033 -1.36250239 -0.02542986

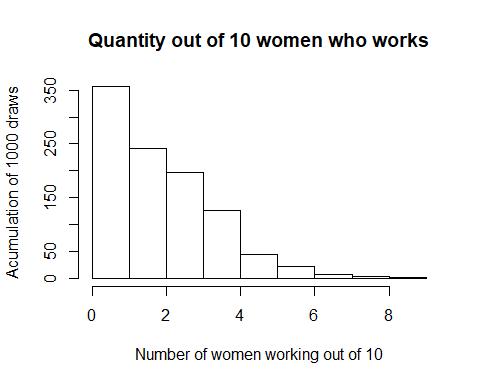
*2 b)* *Write a function that simulates from the predictive distribution of the response variable in a logistic regression. Use your normal approximation from 2(a). Use that function to simulate and plot the predictive distribution for the Work variable for a 40 year old woman, with two children (3 and 9 years old), 8 years of education, 10 years of experience. and a husband with an income of 10.*

We simulated values for our response variable y from the logistic regression with the help of draws from the posterior of beta vector. As seen in the histogram below the probability that this type of woman works is low.



*2 c* *Now, consider 10 women which all have the same features as the woman in 2(b). Rewrite your function and plot the predictive distribution for the number of women, out of these 10, that are working. [Hint: Which distribution can be described as a sum of Bernoulli random variables?]*

We simulated the quantity out of 10 woman with same data whom works with a binomial distribution where p was drawn from the posterior distribution of work probability. Results can be seen in the following histogram.



# Assignment 1

# for multivariate normal functions  
library(mvtnorm)  
  
V\_0 = 4  
sigma\_0 = 1  
my\_0 = c(-10,100,-100)  
set.seed(12345)  
omega\_0 = 0.01\*diag(3)  
omega\_0\_Inverse = solve(omega\_0)  
  
set.seed(12345)  
#Draw 10 draws from out given chi sqared distribution  
XDraw = rchisq(10,V\_0)  
#Transform our 10 draws to the scaled inverse chi square  
deviationDraw = V\_0\*sigma\_0/XDraw   
  
# For each of the draws above we create the joint prior obtaining the distribution for Beta given sigma  
# each draw gives us a set of values for the betas  
#rmvnorm because of that my\_0 is a vector => multivariate  
# sigma is covarance matrix  
joint\_prior = list()  
for(i in 1:10){  
 set.seed(12345)  
 joint\_prior[[i]] = rmvnorm(1, mean = my\_0, sigma = deviationDraw[i]\*omega\_0\_Inverse)  
}  
  
# Function for computing the regression function for different values of x (time)  
function\_temperature = function(betas,x){  
 betas[1] + betas[2]\*x + betas[3]\*x^2  
}  
  
#Plot the temperature regression curve for every set of values for betas  
time = seq(0,1,1/365)  
plot(function\_temperature(joint\_prior[[1]][1,],time), type = 'l', ylim = c(-25,35), xlab = "Days since start of 2018", ylab = "Temperature")  
for(i in 2:10){  
 lines(function\_temperature(joint\_prior[[i]][1,],time), type = 'l')  
   
}  
  
  
# (b)  
  
data = read.table("TempLinkoping.txt", header = TRUE)  
X = data$time  
data$xsquare = X^2   
data$intercept = 1  
Y = data$temp  
X\_matrix = matrix(0, nrow = 365, ncol = 3)  
X\_matrix[,1] = data$intercept  
X\_matrix[,2] = data$time  
X\_matrix[,3] = data$xsquare  
  
beta\_hat = solve((t(X\_matrix)%\*%X\_matrix))%\*%t(X\_matrix)%\*%Y  
  
# get value for my\_n  
my\_n = solve(((t(X\_matrix)%\*%X\_matrix)) + omega\_0)%\*%(((t(X\_matrix)%\*%X\_matrix%\*%beta\_hat))+omega\_0%\*%my\_0)  
  
omega\_n = (t(X\_matrix)%\*%X\_matrix) + omega\_0  
omega\_n\_inverse = solve(omega\_n)  
  
# degrees of freedom  
Vn = V\_0 + length(X)  
  
  
sigma\_n = ((V\_0\*sigma\_0 + (t(Y)%\*%Y + t(my\_0)%\*%omega\_0%\*%my\_0)[1] - (t(my\_n)%\*%omega\_n%\*%my\_n)[1]))/Vn  
  
# number of draws  
n = 1000  
  
# draw values from the marginal posterior distribution of sigma  
set.seed(12345)  
#Draw n draws from our given chi sqared distribution  
XPostDraw = rchisq(n,Vn)  
#Transform our n draws to the scaled inverse chi square  
deviationPostDraw = Vn\*sigma\_n/XPostDraw   
  
# For each of the draws above we create the joint marginal posterior obtaining the distribution for Beta given sigma  
# each draw gives us a set of values for the betas  
joint\_posterior = matrix(0, nrow = n, ncol = 3)  
beta1 = c()  
beta2 = c()  
beta3 = c()  
  
  
set.seed(12345)  
for(i in 1:n){  
 joint\_posterior[i,] = rmvnorm(1, mean = my\_n, sigma = omega\_n\_inverse\*deviationPostDraw[i])  
 beta1 = append(beta1,joint\_posterior[i,1])  
 beta2 = append(beta2,joint\_posterior[i,2])  
 beta3 = append(beta3,joint\_posterior[i,3])  
 temp = temperature\_posterior(joint\_posterior[i,], X\_matrix)  
}  
  
  
# visualize the simulated parameters in histograms  
hist(deviationPostDraw)  
hist(beta1)  
hist(beta2)  
hist(beta3)  
  
# Function f(time) for certain beta values on every values of x (time)  
temperature\_posterior = function(betas,x){  
 x%\*%betas  
}  
  
temperature\_matrix = matrix(0, nrow = 365, ncol = 1000)  
for(i in 1:n){  
 temperature\_matrix[,i] = temperature\_posterior(joint\_posterior[i,], X\_matrix)  
}  
  
#Get median value of f(x) computed for each day  
median\_temperature\_vec = apply(temperature\_matrix,1, median)   
  
CI\_temperature\_vec = apply(X = temperature\_matrix, MARGIN=1, FUN=function(x) quantile(x,c(0.05,0.95)))  
  
  
#Plot posterior median, upper/lower credible interval for the Beta values and SMHI data  
plot(median\_temperature\_vec, type = 'l', ylim = c(-15,25), main = "Posterior median of Betas", xlab = "Days since start of 2018", ylab = "Temperature", col = "blue")  
lines(Y, col = "Black")  
lines(CI\_temperature\_vec[1,], col = "green")  
lines(CI\_temperature\_vec[2,], col = "red")  
legend("topleft", c("SMHI data", "Posterior median", "Upper", "Lower"), col = c("black", "blue", "red", "green"), pch = 21:22, lty = 1:2)  
  
  
# 1 c)  
# function which takes in beta values and returns which x (which day) gives the maximum value  
maximal\_time = function(beta2,beta3){  
   
 x = -beta2/(2\*beta3)  
}  
  
# collect all the max x-values for every set of betas  
max\_vector = c()  
for(i in 1:n){  
 # multiply with 366 to get nr of days instead of range 0,1  
 max\_vector = append(max\_vector,maximal\_time((joint\_posterior[i,2]),(joint\_posterior[i,3]))\*365)  
}  
  
hist(max\_vector, main="Distribution for mode of x", xlab="Number of days since start of year")

# Assignment 2

#Assignment 2   
  
women\_data = read.table('WomenWork.dat', header = TRUE)  
women\_data  
n = nrow(women\_data)  
  
tao = 10  
#covariance\_matrix = c(1:8)  
Y = as.vector(women\_data[,1])  
X = as.matrix(women\_data[,-1])  
col = ncol(X)  
  
# parameters for the prior distribution of beta  
my = rep(0, col)  
sigma\_prior = diag(tao^2, nrow = col)  
  
# function which returns an expression proportional to log beta posterior, this can be optimized for   
# values of beta mode and hessian J (observed hessian evaluated at posterior mode)  
log.posterior = function(betas, x,Y,my,sigma\_prior){  
   
 # is simply log of the product of density function  
 log\_likelihood = sum((X%\*%betas)\*Y-log(1+exp(X%\*%betas)))  
   
 #log of deensity for nultivariate normal => dmvnorm(density multivariate)  
 log\_prior = dmvnorm(x=betas,mean=my,sigma=sigma\_prior, log=TRUE)  
   
 return(log\_likelihood + log\_prior)  
   
}  
  
# Different starting values. Ideally, any random starting value gives you the same optimum (i.e. optimum is unique)  
initVal <- as.vector(rep(0,col));   
  
# function which optmizes over expression log.posterior with respect to its first argument (betas).   
# returns optimal values for beta (mode), and hessian in the mode  
OptParams<-optim(initVal,log.posterior,gr=NULL,X,Y,my,sigma\_prior,method=c("BFGS"),control=list(fnscale=-1),hessian=TRUE)  
  
betas\_posterior = OptParams$par  
# takes negative so that the posterior can be approx. as normal  
# J = -second derivate evaluated in theta\_hat  
hessian\_posterior = -OptParams$hessian  
  
# take inverse for using it in the formula  
hessian\_posterior = solve(hessian\_posterior)  
  
#Draw samples from Betas posterior distribution.   
set.seed(12345)  
posterior\_distribution = rmvnorm(n = 1000, mean = betas\_posterior, sigma = (hessian\_posterior))  
  
#Distribution gives for each draw a beta vector of length 8  
#Take out vecotr of interest (no of small children)  
n\_child = posterior\_distribution[,7]  
  
#95% credible interval  
quantiles = quantile(n\_child, probs = seq(0,1,0.025))  
quantiles  
interval = c(quantiles[2], quantiles[40])  
interval  
  
#Comparision with glm model  
glmModel <- glm(Work ~ ., data = women\_data, family = binomial)  
glmModel$coefficients  
  
#2b  
#Write a function that simulates from the predictive distribution of the response  
#variable in a logistic regression. Use your normal approximation from 2(a).  
#Use that function to simulate and plot the predictive distribution for the Work  
#variable for a 40 year old woman, with two children (3 and 9 years old), 8 years  
#of education, 10 years of experience. and a husband with an income of 10.  
  
# expression for the response variable y (logistic regression)  
work\_distribution = function(x\_data, betas){  
 exp(t(x\_data)%\*%betas) / (1+exp(t(x\_data)%\*%betas))  
}  
  
# the properties of the type of woman we want the work distribution for   
x\_data = c(1,10.000,8,10,1,40,1,1)  
  
outcome = c()  
  
# Simulating from teh predictive distribution (hence giving teh actual outcomes) for y with respect to our x values by simulating beta values from the beta posterior.   
set.seed(12345)  
for (i in 1:1000){  
 betas = rmvnorm(n = 1, mean = betas\_posterior, sigma = (hessian\_posterior))  
 betas = as.vector(betas)  
 prob = work\_distribution(x\_data, betas)  
 # y is binary, therefore bernoulli distributed. Same thing as binomial with one draw.   
 outcome = append(outcome, rbinom(1,1, prob))  
   
   
}  
# histogram representing prob of woman work  
hist(outcome, main="Predictive distribution for if woman works", xlab="Woman work or not", ylab="Acumulation of women who works out of 1000 draws")  
  
  
# 2c  
  
#Now, consider 10 women which all have the same features as the woman in 2(b).  
#Rewrite your function and plot the predictive distribution for the number of  
#women, out of these 10, that are working. [Hint: Which distribution can be  
#described as a sum of Bernoulli random variables?]  
  
# the distribution will be a binomial  
  
# expression for the binomial dist of the quantity out of 10 woman of same type that works. Depends on our   
# posterior dist for the probability that one woman of this type works  
work\_binomial = function(x\_data, betas){  
 p = work\_distribution(x\_data, betas)  
 rbinom(1,10,p)   
}  
  
work\_binomial\_result = c()  
  
# Simulating the distribution for quantity out of 10 woman works (binomial)  
# using posterior dist for beta vectors and for work probability.(work distribution)  
set.seed(12345)  
for (i in 1:1000){  
 betas = rmvnorm(n = 1, mean = betas\_posterior, sigma = (hessian\_posterior))  
 betas = as.vector(betas)  
 work\_binomial\_result = append(work\_binomial\_result,work\_binomial(x\_data,betas))  
}  
  
hist(work\_binomial\_result, main="Quantity out of 10 women who works", xlab="Number of women working out of 10", ylab="Acumulation of 1000 draws")  
  
# one can check these quantiles to compare with prior histogram mode  
#quantile(work\_binomial\_result, probs = seq(0,1,0.025))