# 732A73: Bayesian Learning

Computer Lab 2

Oriol Garrobé, Dávid Hrabovszki

27 April 2020

## Question 1. Linear and polynomial regression.

Given the dataset `TempLinkoping.txt` containing: * Temp: Response variable with daily average of temperatures in Linkoping over 2018. * Time: Covariate of Temp, computed as

$$time = \frac{number\ of\ days\ since\ beginning\ of\ the\ year}{365}$$

We are asked to perform Bayesian analysis of a quadratic regression

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

### (a)

We need to determine the prior distribution of the model parameters. To do so, we are going to set some initial parameters, simulating 1000 draws from the joint prior and compute the regression curve for every draw. Each regression curve is going to be plotted to analyse. From this point, if the result is not satisfactory the hyperparameters will be changed up until the result fits with the prior believ.

*1st draw*

$\mu_0 = [-10, 100, -100]$; $\Omega_0 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$ $\nu_0 = 4$ $\sigma_0^2 = 1$

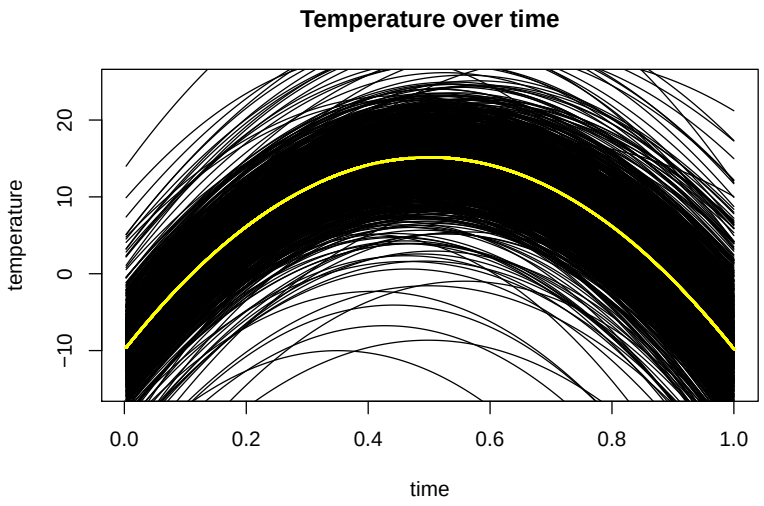


Figure 1: Regression curves for 1st draw.

*2nd draw*

$\mu_1 = [-10, 100, -100]; \quad \Omega_1 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \quad \nu_1 = 4 \quad \sigma_1^2 = 0.5$

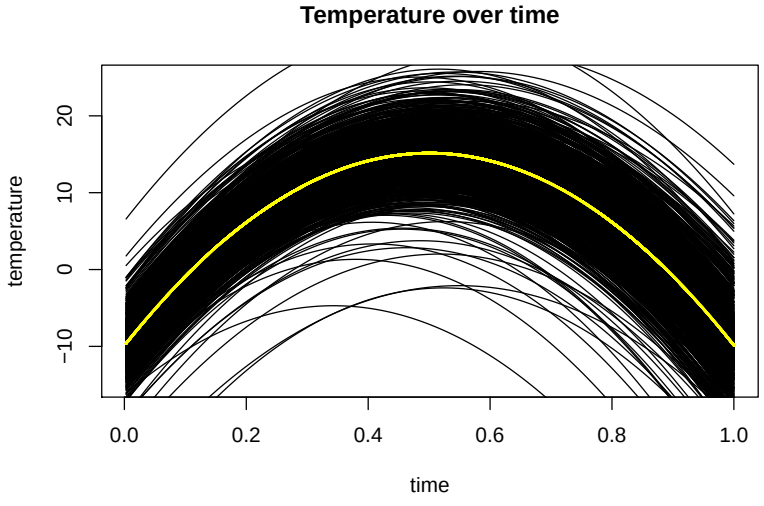


Figure 2: Regression curves for 2nd draw.

*3rd draw*

$\mu_3 = [-10, 100, -100]; \quad \Omega_3 = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} \quad \nu_3 = 4 \quad \sigma_3^2 = 1$

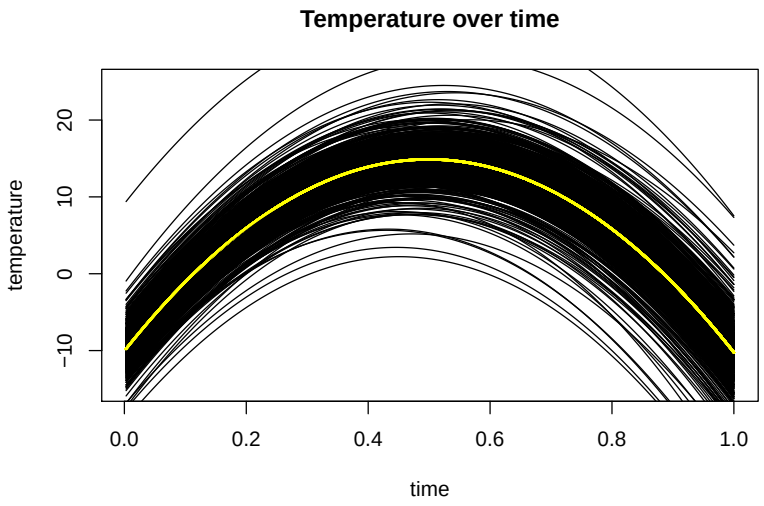


Figure 3: Regression curves for 3rd draw.

*4th draw*

$\mu_4 = [-5, 90, -90]; \quad \Omega_4 = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} \quad \nu_4 = 4 \quad \sigma_4^2 = 1$

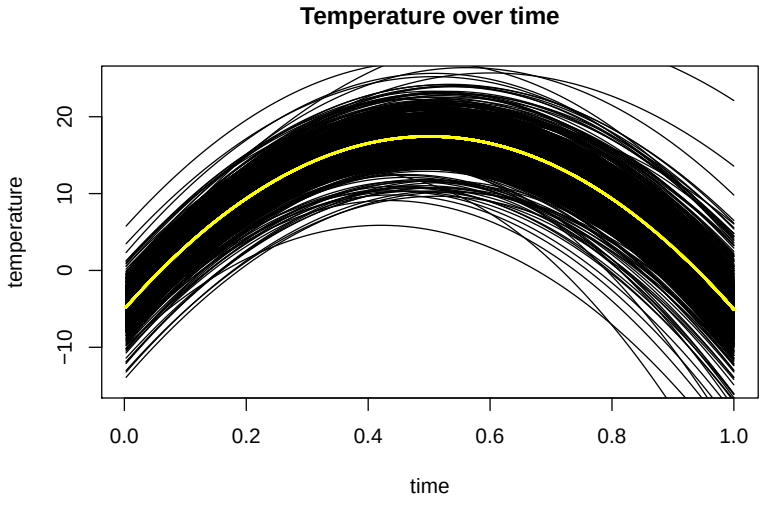


Figure 4: Regression curves for 4th draw.

Looking at the figures above, we decide that the best prior hyperparameter are the ones from the 4th draw. This is due a mean value close to -5 degrees in winter and a little above 15 degrees in summer. This sounds as a reasonable prior for the study.

**(b)**

In order to compute our posterior linear regression we draw samples from the joint posterior and then we plot the results in order to know the distribution of the different parameters. This are the following.

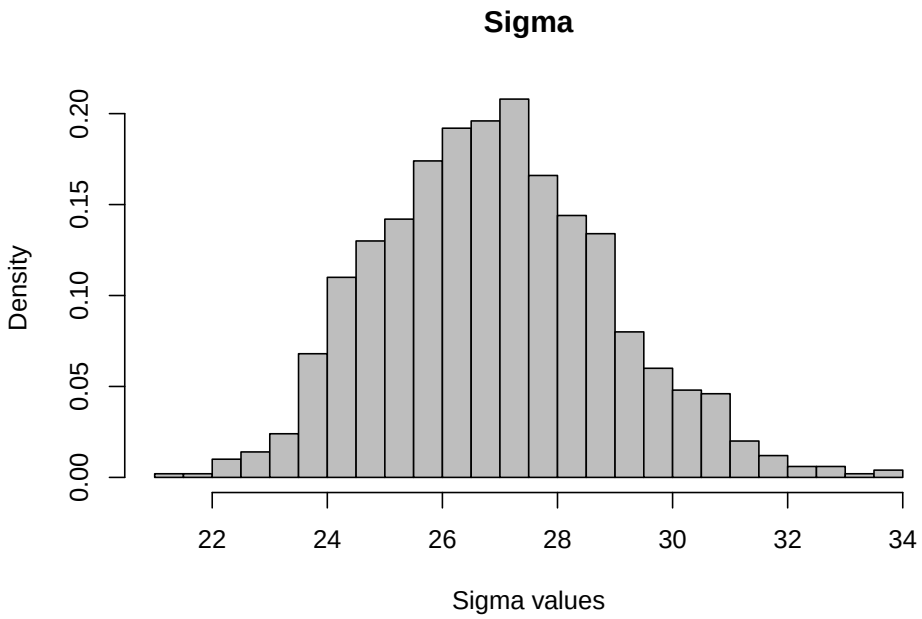


Figure 5: Distribution of Sigma squared.

In 5 we see a distribution that resembles a normal centered around 26.89073.

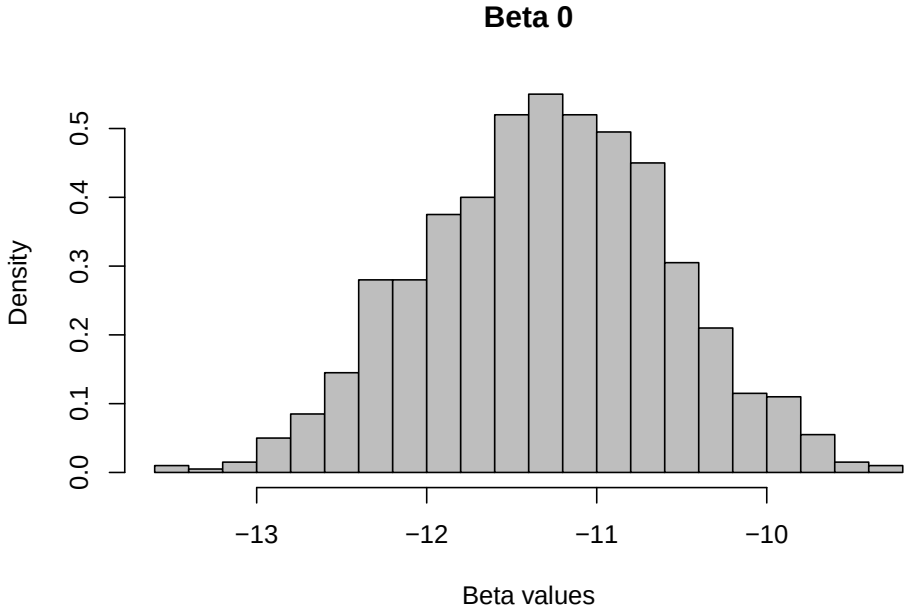


Figure 6: Distribution of Beta 0.

In 6 we see a distribution that resembles a normal centered around −11.28667.

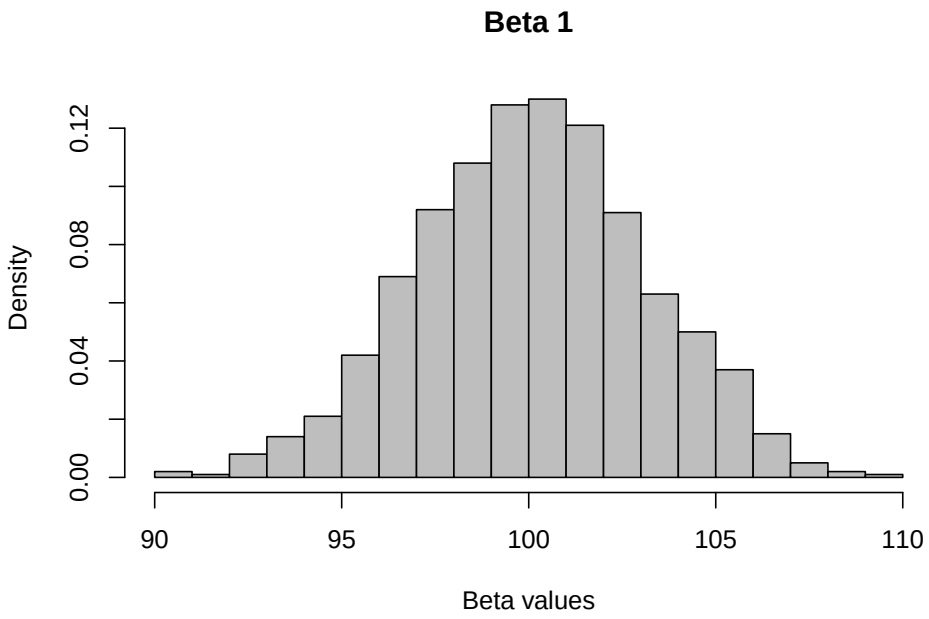


Figure 7: Distribution of Beta 1.

In 7 we see a distribution that resembles a normal centered around 100.1229.

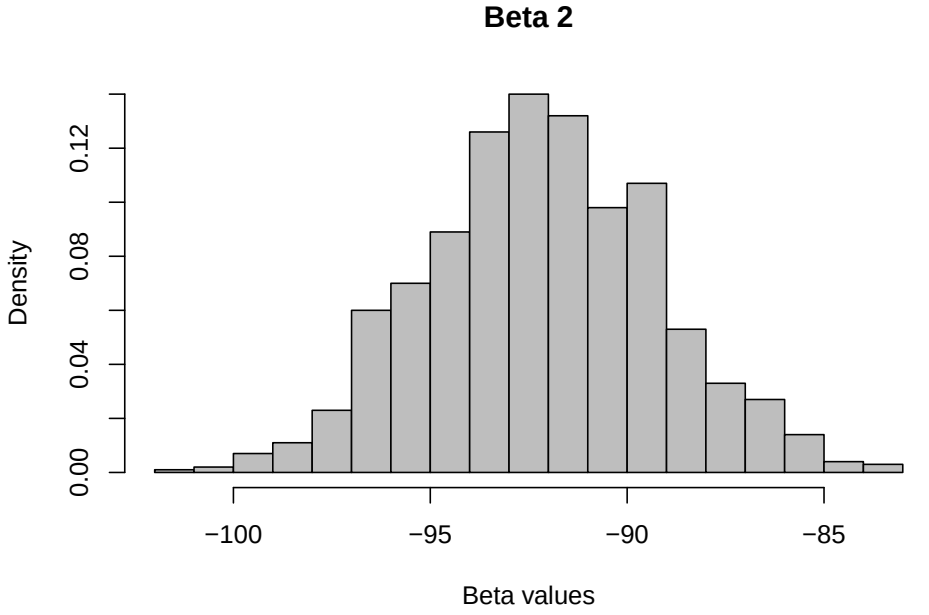


Figure 8: Distribution of Beta 2.

In 8 we see a distribution that resembles a normal centered around −92.20804.

Once we have looked at the distributions of the parameters, we plot the temperature data over time and overlay the median of the posterior regression and the credible intervals. In 9 we can see the median regression -yellow- and the credible intervals -red-.

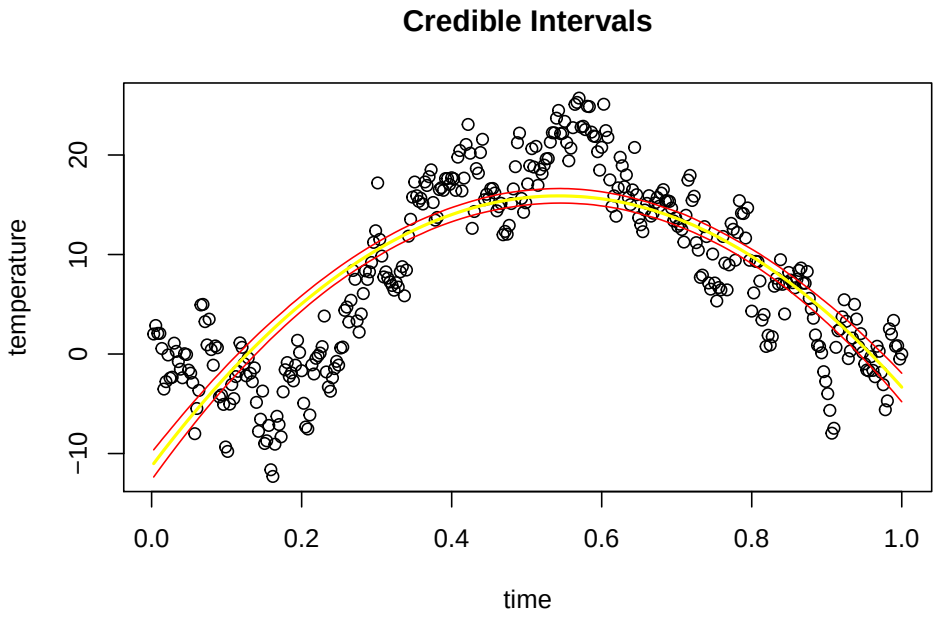


Figure 9: Credible intervals.

The interval bands do not contain most of the data points and they should not. The credible intervals stand for an interval within which the probability that the unknown parameter lies in it with a particular probability. Therefore, most of the data points are outside the interval.

**(c)**

In order to compute the highest expected temperature, given that the regression curve is a quadratic, we can do it by deriving `f(time)` over `time`. By doing so we get the following equation, where $\widetilde{x}$ stands for time with max temperature.

$$\widetilde{x} = \frac{-\beta_1}{2 \cdot \beta_2}.$$

From this point, and given the previous draw of betas, we simulate $\widetilde{x}$. The distribution is the following:

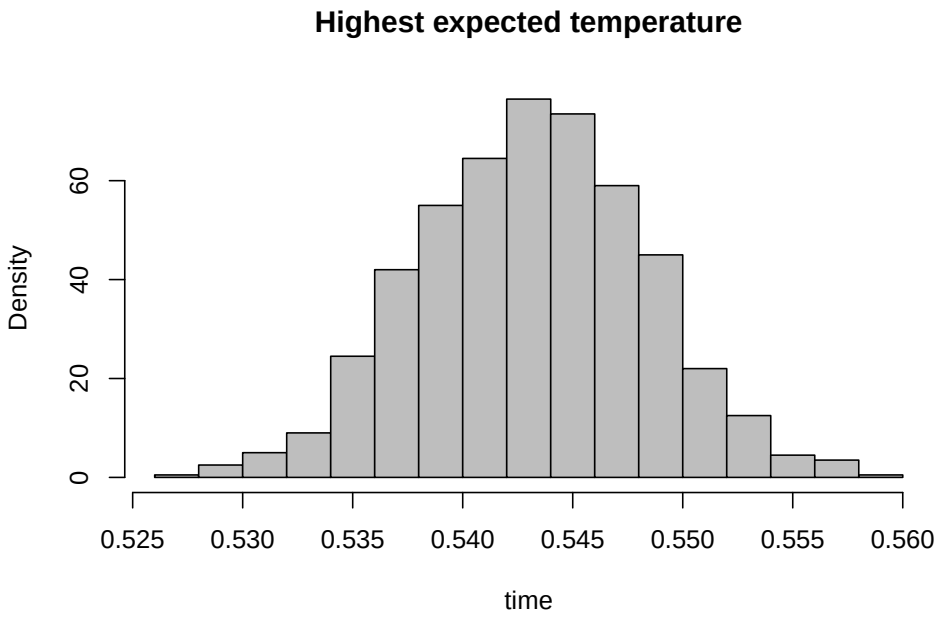


Figure 10: Time with highest expected temperature.

In 10 we see a distribution that resembles a normal centered around 0.5429726.

**(d)**

In order to avoid overfitting when estimating polynomial models of high order we can regularize our prior. In this particular case we want to regularize betas as higher order terms may not be needed. To do so, we modify our prior which is the following:

$$\beta_i|\sigma^2 \sim N\left(\mu, \frac{\sigma^2}{\lambda}\right)$$

For those terms of high order that we suspect may not be needed we set $\mu = 0$ and we vary the value of $\lambda$ as pleased. Larger values of $\lambda$ give smoother fit, as $\beta$ is closer to 0. Note: $\Omega_0 = \lambda I$

In particular, with the given $\sigma_0^2$ and $\Omega_0$ from the exercise, the high order polynomial should not over-fit. If it did, we would need to increase the value of $\Omega_0$ to diagonal values closer to 1.

## Question 2. Posterior approximation for classification with logistic regression

The dataset WomenWork.dat contains n = 200 observations (i.e. women) on the following nine variables:

Constant, HusbandInc, EducYears, ExpYears, ExpYears2, Age, NSmallChild, NBigChild

**a**

We consider the logistic regression

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

where $y = 0$ is the woman does not work and $y = 1$ if she does. $X$ is an 8-dimensional vector with the features including the intercept. In this part, we approximate the posterior distribution of the parameter vector $\beta$:

$\beta|y, X \sim N\Big(\tilde{\beta}, J_y^{-1}(\tilde{\beta})\Big)$

$\tilde{\beta}$ is the posterior mode and $J(\tilde{\beta}) = -\frac{\partial^2 lnp(\beta|y)}{\partial\beta\partial\beta^T}|_{\beta=\tilde{\beta}}$ is the observed Hessian evaluated at the posterior mode. We use the following prior: $\beta \sim N(0, \tau^2 I)$, with $\tau = 10$.

The code for the approximation can be found and in the Appendix and was written using Mattias Villani's implementation as a template: https://github.com/mattiasvillani/BayesLearnCourse/raw/master/Code/MainOptimizeSpam.zip

We obtained the following posterior mode for $\beta$:

| Beta1 | Beta2 | Beta3 | Beta4 | Beta5 | Beta6 | Beta7 | Beta8 |
|---|---|---|---|---|---|---|---|
| 0.6267043 | -0.0197913 | 0.1802191 | 0.1675718 | -0.1446182 | -0.0820651 | -1.359151 | -0.024684 |

And the posterior covariance matrix $J_y^{-1}(\tilde{\beta})$:

| | Beta1 | Beta2 | Beta3 | Beta4 | Beta5 | Beta6 | Beta7 | Beta8 |
|---|---|---|---|---|---|---|---|---|
| Beta1 | 2.26603 | 0.00334 | -0.06545 | -0.01179 | 0.04578 | -0.03029 | -0.18875 | -0.09802 |
| Beta2 | 0.00334 | 0.00025 | -0.00056 | -0.00003 | 0.00014 | -0.00004 | 0.00051 | -0.00014 |
| Beta3 | -0.06545 | -0.00056 | 0.00622 | -0.00036 | 0.00190 | 0.00000 | -0.00613 | 0.00175 |
| Beta4 | -0.01179 | -0.00003 | -0.00036 | 0.00435 | -0.01425 | -0.00013 | -0.00147 | 0.00054 |
| Beta5 | 0.04578 | 0.00014 | 0.00190 | -0.01425 | 0.05558 | -0.00033 | 0.00321 | 0.00051 |
| Beta6 | -0.03029 | -0.00004 | 0.00000 | -0.00013 | -0.00033 | 0.00072 | 0.00518 | 0.00110 |
| Beta7 | -0.18875 | 0.00051 | -0.00613 | -0.00147 | 0.00321 | 0.00518 | 0.15126 | 0.00677 |
| Beta8 | -0.09802 | -0.00014 | 0.00175 | 0.00054 | 0.00051 | 0.00110 | 0.00677 | 0.01997 |

We also compute an approximate 95% credible interval for the coefficient of the NSmallChild variable: $[-2.121445, -0.5968567]$ NSmallChild is an important determinant of whether a woman is working or not, because it has the highest absolute valued posterior coefficient $\tilde{\beta}$ of all features. The posterior mode of its coefficient is -1.36, which means that the more small children a woman has, the less likely it is that she's working, because the sign is negative.

To verify that we got the correct result, we compare it to maximum likelihood estimates, and note that the numbers are very similar.

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

**b**

In this part we simulate from the predictive distribution of the response variable in a logistic regression. More specificially we 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. To do this, first we draw random numbers from the multivariate normal distribution, with mean $\tilde{\beta}$ and covariance matrix $J_y^{-1}(\tilde{\beta})$. Since in normal distributions the mean is equal to the mode, we can use the result from 2.a. Then we apply our implementation of the logistric regression function defined above to the given woman with every random draw.

The resulting distribution in Figue 11 shows that the given woman is much more likely to not be working, because the mode is around 0.2. This is mostly because she has a small child, which as we've seen before, is correlated negatively with working.

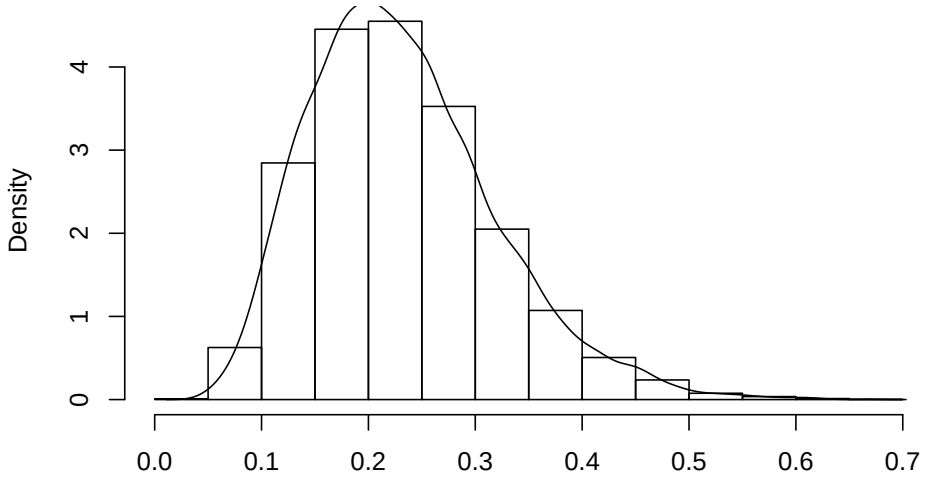


Figure 11: Histogram of the predictive distribution of the response variable of a woman with given features

**c**

Now, we consider 10 women which all have the same features as the woman in 2(b). The random variable of whether a woman is working or not follows a Bernoulli distribution, but if we add 10 such random variables, we get a Binomial distribution, with parameters n = 10 and p = p(given woman works). We calculate the probability of working as the mean of the Bernoulli random variables calculated in 2.b.

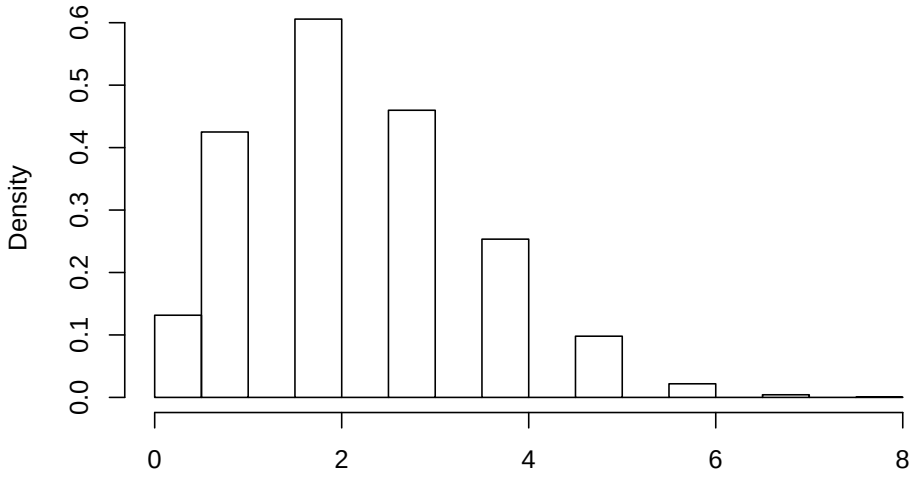


Figure 12: Histogram of the predictive distribution of the number of women working (out of 10)

Figue 12 shows that 2 such women working has the highest probability.

# Appendix

```
knitr::opts_chunk$set(echo=FALSE, eval=TRUE)
# R version
RNGversion('3.5.1')

# libraries
library(mvtnorm)
library(knitr)
############################################################
#####  Question 1. Linear and polynomial regression  #######
############################################################

data <- read.table("TempLinkoping.txt", header = TRUE)

y <- data$temp
time <- data$time

n <- length(y)
X <- cbind(matrix(1, nrow = length(y)),time,time^2)

# Initial prior hyperparameters
mu_0 <- c(-10,100,-100)
omega_0 <- 0.1*diag(3)
nu_0 <- 4
sigma2_0 <- 1

### (a)

# Conjugate prior

draw_sigma2 <- function(N_draws, nu, sigma2){
  sigma2 <- nu*sigma2/rchisq(N_draws, nu)
}

draw_betas <- function(sigma2_draws, mu, omega, order=3){
  draw_beta <- matrix(0, nrow=length(sigma2_draws), ncol=order )
  for (sigma in sigma2_draws) {
    i <- which(sigma2_draws==sigma)
    draw_beta[i,] <- rmvnorm(1, mu, sigma*solve(omega))
  }
  return(draw_beta)
}

plot_regression_draws <- function(X, betas, time){
  for (i in 1:length(betas[,1])) {
    if (i==1) plot(time, X%*%betas[i,], type = 'l', ylim=c(-15, 25), main = 'Temperature over time', yla
    else points(time, X%*%betas[i,], type='l')
    points(time,X%*%colMeans(betas),type = 'l',lwd=2,col='yellow')
  }
}

N_draws <- 1000
```

```r
# Initial draw
sigma2_0_draws <- draw_sigma2(N_draws, nu_0, sigma2_0)
betas_0 <- draw_betas(sigma2_0_draws, mu_0, omega_0)
plot_regression_draws(X, betas_0, time)

# 1st iteration
nu_1 <- 4
sigma2_1 <- 0.5
sigma2_1_draws <- draw_sigma2(N_draws, nu_1, sigma2_1)
betas_1 <- draw_betas(sigma2_1_draws, mu_0, omega_0)
plot_regression_draws(X, betas_1, time)

# 2nd iteration
omega_2 <- 0.5*diag(3)
betas_2 <- draw_betas(sigma2_0_draws, mu_0, omega_2)
plot_regression_draws(X, betas_2, time)
# 3rd iteration
mu_3 <- mu_1 <- c(-5,90,-90)
betas_3 <- draw_betas(sigma2_0_draws, mu_3, omega_2)
plot_regression_draws(X, betas_3, time)

### (b)

# Conjugate posterior

N_draws = 1000
mu_0<-mu_1
omega_0<-omega_2

# Hyperparameters

beta_hat <- solve(t(X)%*%X)%*%t(X)%*%y
mu_n <- solve(t(X)%*%X + omega_0) %*% (t(X)%*%X%*%beta_hat + omega_0%*%mu_0)
omega_n <- t(X)%*%X + omega_0
nu_n <- nu_0 + n
nu_sigma2_n <- nu_0*sigma2_0 + (t(y)%*%y + t(mu_0)%*%omega_0%*%mu_0 - t(mu_n)%*%omega_n%*%mu_n)
sigma2_n <- nu_sigma2_n[1] / nu_n

posterior_sigma2 <- draw_sigma2(N_draws, nu_n, sigma2_n)
posterior_betas <- draw_betas(posterior_sigma2, mu_n, omega_n)

# Sigma
mean_p_sigma2 <- mean(posterior_sigma2) #26.89073
hist(posterior_sigma2, freq = F, breaks=20, main="Sigma", col = 'grey', xlab='Sigma values')
#Beta 1
mean_p_betas_1 <- mean(posterior_betas[,1]) #-11.28667
hist(posterior_betas[,1], freq = F, breaks=20, main="Beta 0", col="grey", xlab="Beta values")
#Beta 2
mean_p_betas_2 <- mean(posterior_betas[,2]) #100.1229
hist(posterior_betas[,2], freq = F, breaks=20, main="Beta 1", col="grey", xlab="Beta values")

#Beta 3
mean_p_betas_3 <- mean(posterior_betas[,3]) #-92.20804
```

```r
hist(posterior_betas[,3], freq=F, breaks=20, main="Beta 2", col="grey", xlab="Beta values")

predicted_values <- matrix(0, nrow = n, ncol = N_draws)
for (i in 1:N_draws) {
  predicted_values[,i] <- X%*%posterior_betas[i,]
}

intervals <- matrix(0, nrow = n, ncol = 3)
for (j in 1:n) {
  intervals[j,]<-c(median(predicted_values[j,]), quantile(predicted_values[j,], probs = c(0.025, 0.975))
}
plot(time, y, main="Credible Intervals", ylab='temperature')
points(time, intervals[,1], lwd=2, col='yellow', type='l')
points(time, intervals[,2], lwd=1, col='red', type='l')
points(time, intervals[,3], lwd=1, col='red', type='l')
x_tilda <- -posterior_betas[,2]/(2*posterior_betas[,3]) #0.5453696
mean_x_tilda <- mean(x_tilda) #0.5429726
hist(x_tilda, freq=F, breaks=20, col="grey", main="Highest expected temperature", xlab = "time")
# 2. Posterior approximation for classification with logistic regression

# 2.a

# The following code was written using Mattias Villani's implementation as a template:
# https://github.com/mattiasvillani/BayesLearnCourse/raw/master/Code/MainOptimizeSpam.zip

# read data
women = read.table('WomenWork.dat', header = T)

y = as.vector(women[,1])
x = as.matrix(women[,2:9])
tau = 10
nfeatures = dim(x)[2]

# prior
mu = as.vector(rep(0,nfeatures))
sigma = diag(tau^2, nfeatures, nfeatures)

LogPrior = function(Beta, mu, sigma){
  return(dmvnorm(Beta, as.matrix(mu), sigma, log = T))
}

# log likelihood
LogLikelihood = function(Beta, y, x){
  sum(y * x%*%Beta - log(1 + exp(x%*%Beta)))
}

# log posterior
LogPosterior = function(Beta, y, x, mu, sigma){
  LogPrior(Beta, mu, sigma) + LogLikelihood(Beta, y, x)
  # we can sum them instead of multiplying, because of the log
}

# initialize Beta vector randomly
```

```r
# Seed
set.seed(1234567890)
Beta_init = as.vector(rnorm(dim(x)[2]))

# optimizing the log posterior by changing the Betas (maximize)
res = optim(Beta_init, LogPosterior, gr = NULL, y, x, mu, sigma,
                    method="BFGS", control=list(fnscale=-1), hessian=T)

Beta_hat = res$par # posterior mode
Hessian = res$hessian
post_sigma = solve(-Hessian) # posterior cov matrix
stdev = sqrt(diag(post_sigma))

# approximate 95% credible interval for NSmallChild
lower = Beta_hat[7] - 1.96 * stdev[7] #-2.121445
upper = Beta_hat[7] + 1.96 * stdev[7] #-0.5968567

betas = t(as.matrix(Beta_hat))
colnames(betas) = c("Beta1","Beta2","Beta3","Beta4","Beta5","Beta6","Beta7","Beta8")
kable(betas)
sigma_table = data.frame(post_sigma)
colnames(sigma_table) = c("Beta1","Beta2","Beta3","Beta4","Beta5","Beta6","Beta7","Beta8")
rownames(sigma_table) = c("Beta1","Beta2","Beta3","Beta4","Beta5","Beta6","Beta7","Beta8")
kable(round(sigma_table,5))

# check results
glmModel <- glm(Work~0 + ., data = women, family = binomial)
kable(t(as.matrix(glmModel$coefficients))) #roughly the same as Beta_hat
# 2.b
# The mode of a normal distribution is equal to the mean.

log_reg = function(Beta, x){
  return(exp(x%*%Beta) / (1 + exp(x%*%Beta)))
}
Constant = 1
HusbandInc = 10
EducYears = 8
ExpYears = 10
ExpYears2 = (ExpYears/10)^2
Age = 40
NSmallChild = 1
NBigChild = 1

x_pred = c(Constant, HusbandInc, EducYears, ExpYears, ExpYears2, Age, NSmallChild, NBigChild)

# simulate posterior draws from Beta
n = 10000
post_sim = rmvnorm(n, mean = Beta_hat, sigma = post_sigma) # should i add pop variance to post_sigma?

# calculate target y
pred = apply(post_sim, 1, log_reg, x_pred)
hist(pred, freq = F, main = "", xlab = "")
lines(density(pred))
```

```r
# 2.c
# probability of success (working)
p = mean(pred)

binom_sim = rbinom(10000, 10, p)
hist(binom_sim, freq = F, main = "", xlab = "")
```