Bayesian Learning: Lab 2

Mohsen Pirmoradiyan, Ahmed Alhasan

2020-05-28

1. Linear and polynomial regression

The dataset `TempLinkoping.txt` contains daily average temperatures (in Celcius degrees) at Malmslätt, Linköping over the course of the year 2018. The response variable is *temp* and the covariate is

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

The task is to perform a Bayesian analysis of a quadratic regression

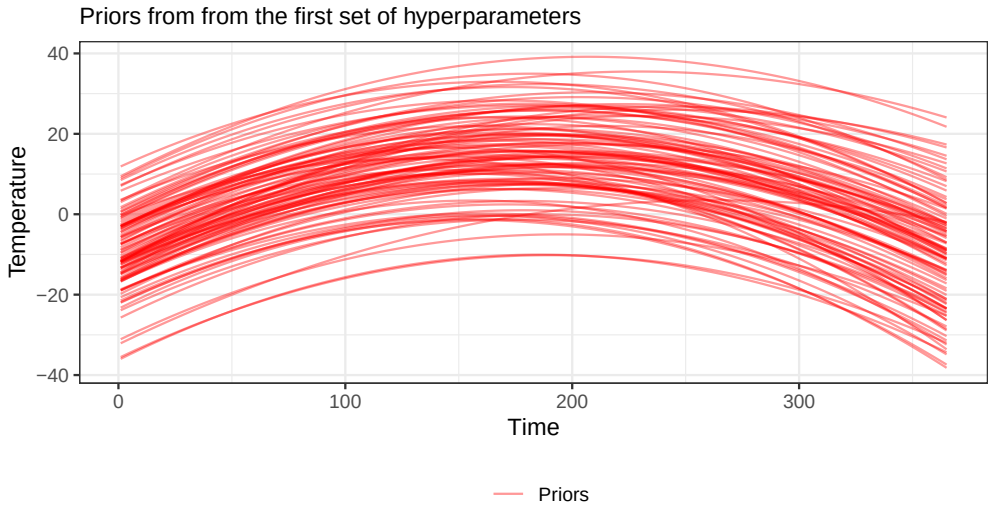
$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

(a) *Determining the prior distribution of the model parameters.* Use the conjugate prior for the linear regression model. Your task is to set the prior hyperparameters μ_0, Ω_0, ν_0 and σ_0^2 to sensible values. Start with $\mu_0 = (-10, 100, -100)^T$, $\Omega_0 = 0.01 \cdot I_3$, $\nu_0 = 4$ and $\sigma_0^2 = 1$. Check if this prior agrees with your prior opinions by simulating draws from the joint prior of all parameters and for every draw compute the regression curve. This gives a collection of regression curves, one for each draw from the prior. Do the collection of curves look reasonable? If not, change the prior hyperparameters until the collection of prior regression curves agrees with your prior beliefs about the regression curve. [Hint: the R package `mvtnorm` will be handy. And use your $Inv - \chi^2$ simulator from Lab 1.]

- The Provided set of hyper-parameters gave a wide range of uncertainty of what might the temperature be in Linkoping in a year which does not agree with our belief that the temperature has less variance than this i.e. the temperature in summer cannot be in minus or very high in winter.

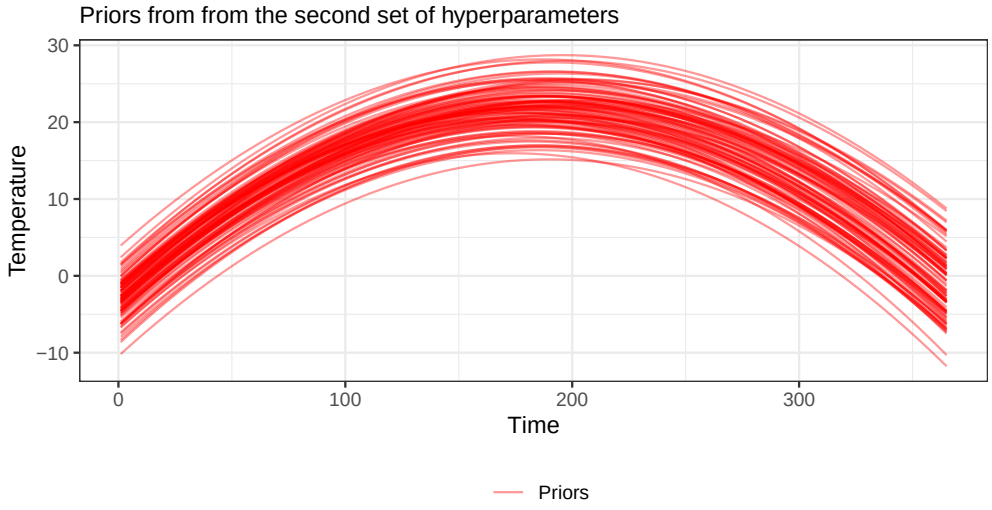
```
################################################################################
## 1. Linear and polynomial regression
################################################################################

## A

link_temp <- read.table("Other/Data/TempLinkoping.txt", header=TRUE)
attach(link_temp)

mu_0  <- c(-10,100,-100)
om_0  <- diag(rep(0.01,3))
nu_0  <- 4
var_0 <- 1

inv_chi <- function(df, Var){
  chi <- rchisq(1, df)
  Var <- (df * Var) / chi
  return(Var)
```

```r
}
model_var_0 <- inv_chi(df=nu_0, Var=var_0)


library(mvtnorm)
nPriors <- 100
X    <- cbind(rep(1,length(time)), time, time^2)
y    <- temp

prior_generator <- function(n, mu, om, model_var){
  yHat <- matrix(0, nrow=length(time), ncol=n)
  Xy   <- NULL
  for(i in 1:n){
    Beta      <- rmvnorm(n=1, mean=mu, sigma=model_var * solve(om))
    yHat[,i]  <- X %*% matrix(Beta)
    temporary <- data.frame(x=1:length(time), y=yHat[,i], col=rep(i:i, length(time)))
    Xy        <- rbind(Xy, temporary)
  }
  return(data.frame(Xy))
}
prior_data <- prior_generator(n=nPriors, mu=mu_0, om=om_0, model_var=model_var_0)

library(ggplot2)
ggplot(prior_data) +
  geom_line(aes(x=x, y=y, group=col, color = "Priors"), alpha=0.4) +
  labs(subtitle="Priors from from the first set of hyperparameters", x = "Time", y = "Temperature") +
  scale_color_manual(values = c("red")) +
  theme_bw() +
  theme(legend.position="bottom", legend.title = element_blank())
```



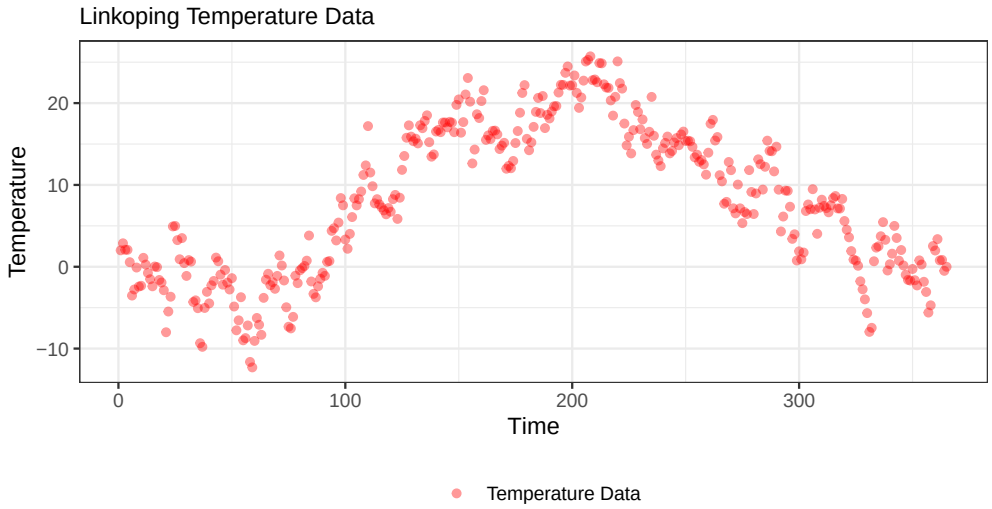
- We have increased the precision hyper-parameter Ω_0 to 0.15 to reduce this variance otherwise we could do that by reducing the variance hyper-paramter itself
- Also we changed the mean of the intercept which represent the temperature of day 1 in the year, January, to reflect our belief of what might be the average temperature that date, Changed β_1 and β_2 to 98 and -96 respectively to make the middle hump higher "temperatures in summer between 10-30C and the end of the year similar to the begining of the year
- And increased ν_0 to 100 to give us more control of the outcome of model variance "more certainty"

```r
# The intercept mean (starting day which is in January) changed to reflect
# our belief of what might be the mean temperature in the first 2 months
# Changed B_1 & B_2 to 98 & -96 respectively to make the middle hump higher "temperatures in summer betu
# and the end of the year similar to the begining of the year
mu_0  <- c(-3,98,-96)
# The om_0 increased to 0.15 to improve the accuracy otherwise we could reduce the variance hyper-parame
om_0  <- diag(c(0.15,0.15,0.15))
# Increased nu to 100 to give us more control of the outcome of model variance "more certainty"
nu_0  <- 100
var_0 <- 1
model_var_0 <- inv_chi(df=nu_0, Var=var_0)

prior_data <- prior_generator(n=nPriors, mu=mu_0, om=om_0, model_var=model_var_0)

ggplot(prior_data) +
  geom_line(aes(x=x, y=y, group=col, color = "Priors"), alpha=0.4) +
  labs(subtitle="Priors from from the second set of hyperparameters", x = "Time", y = "Temperature") +
  scale_color_manual(values = c("red")) +
  theme_bw() +
  theme(legend.position="bottom", legend.title = element_blank())
```



- After we have made our opnion about the prior we could plot the data that we will use as well to compute the posterior in the next step

```r
ggplot(prior_data) +
  geom_point(data=link_temp, aes(x=1:length(time), y=temp, color="Temperature Data"), alpha=0.4) +
  labs(subtitle="Linkoping Temperature Data", x = "Time", y = "Temperature") +
  scale_color_manual(values = c("red")) +
  theme_bw() +
  theme(legend.position="bottom", legend.title = element_blank())
```



(b) Write a program that *simulates from the joint posterior distribution* of $\beta_0, \beta_1, \beta_2$ and σ^2. Plot the marginal posteriors for each parameter as a histogram. Also produce another figure with a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function $f(time) = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$, computed for every value of *time*. Also overlay curves for the lower 2.5% and upper 97.5% posterior credible interval for *f(time)*. That is, compute the 95% equal tail posterior probability intervals for every value of *time* and then connect the lower and upper limits of the interval by curves. Does the interval bands contain most of the data points? Should they?

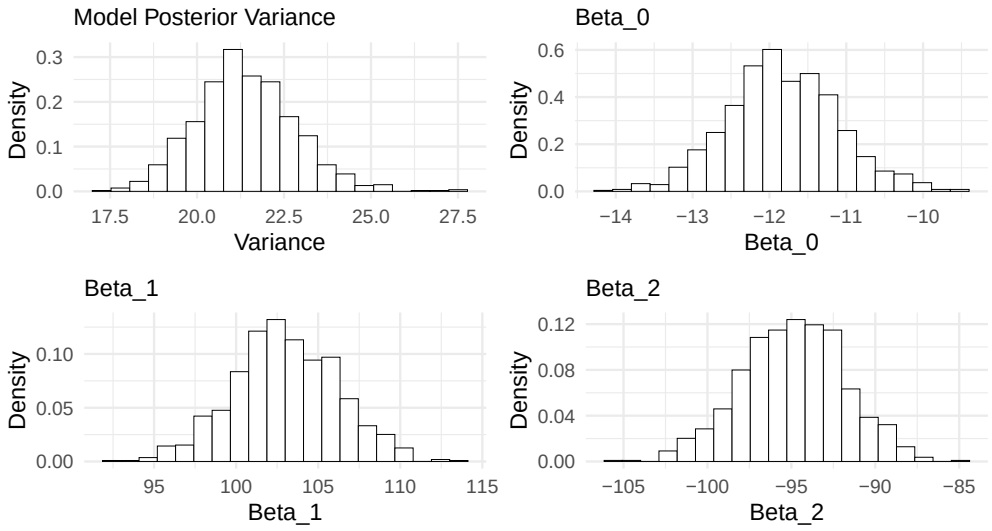
$$\sigma^2 \mid y \sim Inv - \chi^2(\nu_n, \sigma_n^2)$$

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

$$\begin{aligned}
\Omega_n &= X^T X + \Omega_0 \\
\mu_n &= (X^T X + \Omega_0)^{-1}(X^T X \hat{\beta} + \Omega_0 \mu_0) \\
\nu_n &= \nu_0 + n \\
\nu_n \sigma_n^2 &= \nu_0 \sigma_0^2 + (y^T y + \mu_0^T \Omega_0 \mu_0 - \mu_n^T \Omega_n \mu_n)
\end{aligned}$$

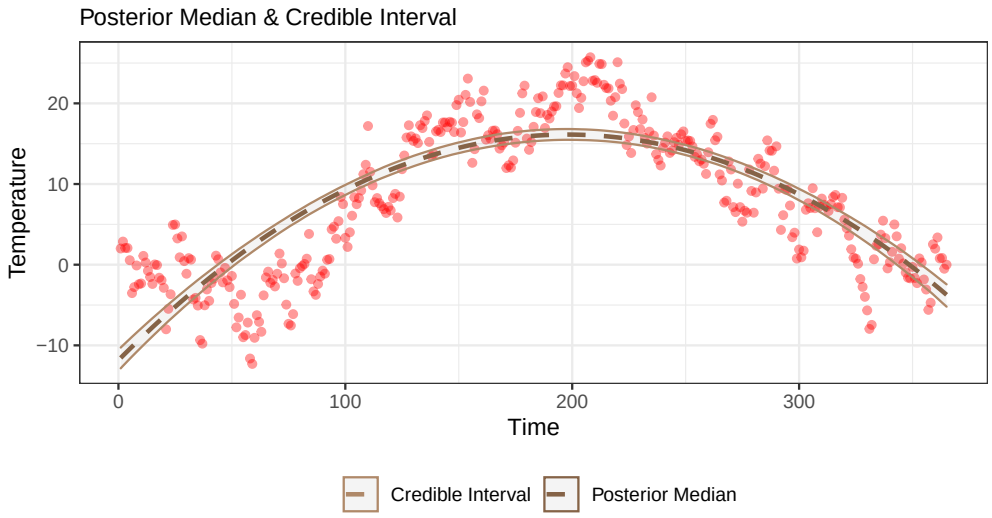
```
## B

n    <- length(time)
B_h  <- as.numeric(lm(temp ~ time + I(time^2), data=link_temp)$coefficients)
mu_n <- as.numeric(solve(t(X) %*% X + om_0) %*% (t(X) %*% X %*% B_h + om_0 %*% mu_0))
om_n <- matrix(t(X) %*% X + om_0, ncol=3)
nu_n <- nu_0 + n
nu_n_var_0 <- nu_0 * var_0 + (t(y) %*% y + (t(mu_0) %*% om_0 %*% mu_0) - (t(mu_n) %*% om_n %*% mu_n))
var_n <- as.numeric(nu_n_var_0/nu_n)

nPosters <- 1000
posterPmeters <- NULL
for (i in 1:nPosters) {
  model_var_n   <- inv_chi(df=nu_n, Var=var_n)
  B_n           <- rmvnorm(n=1, mean=mu_n, sigma=model_var_n * solve(om_n))
  temporary     <- data.frame(Var=model_var_n, B_0=B_n[[1]], B_1=B_n[[2]], B_2=B_n[[3]])
```

```
  posterPmeters <- rbind(posterPmeters, temporary)
}

p1 <- ggplot(posterPmeters) +
  geom_histogram(aes(x=Var, y=..density..), bins=20, fill="#ffffffff", colour="black", size=0.1) +
  labs(subtitle="Model Posterior Variance", y="Density", x="Variance") +
  theme_minimal()
p2 <- ggplot(posterPmeters) +
  geom_histogram(aes(x=B_0, y=..density..), bins=20, fill="#ffffffff", colour="black", size=0.1) +
  labs(subtitle="Beta_0", y="Density", x="Beta_0") +
  theme_minimal()
p3 <- ggplot(posterPmeters) +
  geom_histogram(aes(x=B_1, y=..density..), bins=20, fill="#ffffffff", colour="black", size=0.1) +
  labs(subtitle="Beta_1", y="Density", x="Beta_1") +
  theme_minimal()
p4 <- ggplot(posterPmeters) +
  geom_histogram(aes(x=B_2, y=..density..), bins=20, fill="#ffffffff", colour="black", size=0.1) +
  labs(subtitle="Beta_2", y="Density", x="Beta_2") +
  theme_minimal()

library(gridExtra)
grid.arrange(p1, p2, p3, p4, nrow = 2)
```




```
Beta <- matrix(0, nrow = nPosters, ncol = 3)
poster_generator <- function(n, mu, om, model_var){
  yHat <- matrix(0, nrow=length(time), ncol=n)
  for(i in 1:n){
    Beta[i,] <- rmvnorm(n=1, mean=mu, sigma=model_var * solve(om))
    yHat[,i] <- X %*% matrix(Beta[i,])
  }
  return(list(Beta, yHat))
}

poster_model  <- poster_generator(n=nPosters, mu=mu_n, om=om_n, model_var=model_var_n)[[2]]

poster_mean   <- apply(poster_model, 1, mean)
poster_median <- apply(poster_model, 1, median)
```

```r
lower_bound   <- apply(poster_model, 1, quantile, probs= c(0.025, 0.975))[1,]
upper_bound   <- apply(poster_model, 1, quantile, probs= c(0.025, 0.975))[2,]

poster_data <- data.frame(time=1:length(time), temp, poster_mean, poster_median, lower_bound, upper_bour
ggplot(poster_data) +
  geom_point(aes(x=time, y=temp), color="red", alpha=0.4) +
  geom_ribbon(aes(x=time, ymin=lower_bound, ymax=upper_bound, color="Credible Interval"), alpha=0.05) +
  geom_line(aes(x=time, y=poster_median, color="Posterior Median"), size = 1, linetype = 2) +
  labs(subtitle="Posterior Median & Credible Interval", y="Temperature", x="Time") +
  scale_color_manual(breaks=c("Credible Interval", "Posterior Median"),
                     values=c("#AF8969", "#856347")) +
  theme_bw() +
  theme(legend.position="bottom", legend.title=element_blank())
```



- The credible interval does not contain most of the data points and it shouldn't be since it is not a prediction interval, it is the interval that the median fall into with proabability=95 percent

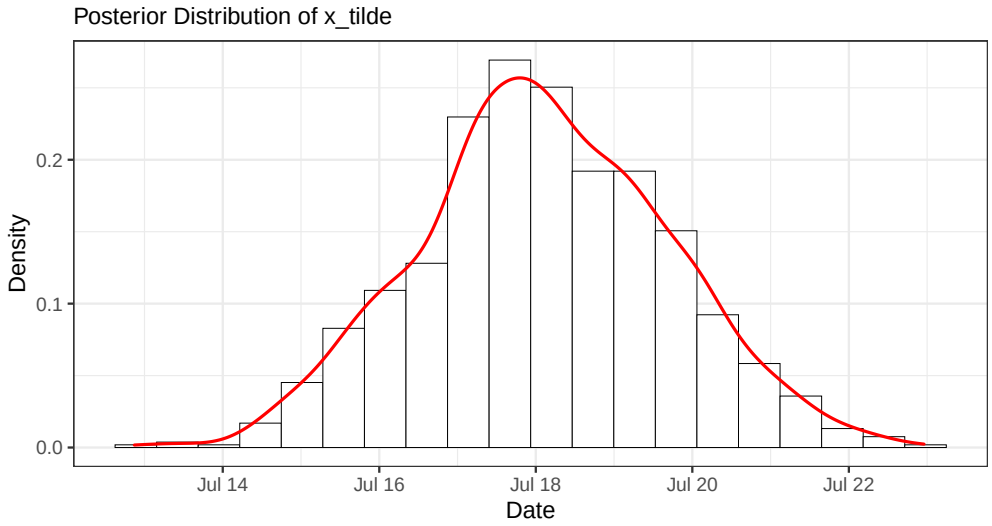
(c) It is of interest to locate the *time* with the highest expected temperature (that is, the *time* where f(*time*) is maximal). Let's call this value $\tilde{x}$. Use the simulations in b) to simulate from the *posterior distribution of* $\tilde{x}$. [Hint: the regression curve is a quadratic. You can find a simple formula for $\tilde{x}$ given β_0, β_1 and β_2.]

- We can get the maximum temperature by taking the derivative in respect to time and setting it to 0

$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$$
$$\frac{temp}{\partial time} = \beta_1 + 2\,\beta_2\, time$$
$$\tilde{x} = -\frac{\beta_1}{2\beta_2}$$

```r
## C
Betas   <- poster_generator(n=nPosters, mu=mu_n, om=om_n, model_var=model_var_n)[[1]]
x_tilde <- -Betas[,2] / (2 * Betas[,3])
dates   <- as.Date(365*x_tilde, origin="2018-01-01")
```

```r
ggplot(as.data.frame(dates)) +
  geom_histogram(aes(x=dates, y=..density..), bins=20, color="black", fill="#ffffffff", size=0.2) +
  geom_density(aes(x=dates, y=..density..), color = "red", size = 0.7) +
  labs(subtitle="Posterior Distribution of x_tilde", y="Density", x="Date") +
  theme_bw()
```



(d) Say now that you want to *estimate a polynomial model of order* 7, but you suspect that higher order terms may not be needed, and you worry about overfitting. Suggest a suitable prior that mitigates this potential problem. You do not need to compute the posterior, just write down your prior. [Hint: the task is to specify μ_0 and Ω_0 in a smart way.]

- We can do that by making the prior mean "8 dimensional vector" far from the data mean and increasing the precision hyper-parameter Ω 8x8 matrix each by a constant depending how much we want to penalize the data since they make a strong informative prior that pull the posterior out from the data

2. Posterior approximation for classification with logistic regression

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

Variable	Data type	Meaning	Role
Work	Binary	Whether or not the woman works	Response
Constant	1	Constant to the intercept	Feature
HusbandInc	Numeric	Husband's income	Feature
EducYears	Counts	Years of education	Feature
ExpYears	Counts	Years of experience	Feature
ExpYears2	Numeric	(Years of experience/10)^2	Feature
Age	Counts	Age	Feature
NSmallChild	Counts	Number of child ≤ 6 years in household	Feature
NBigChild	Counts	Number of child > 6 years in household	Feature

(a) Consider the logistic regression

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

where y is the binary variable with $y = 1$ if the woman works and $y = 0$ if she does not. x is a 8-dimensional vector containing the eight features (including a one for the constant term that models the intercept).

The goal is to approximate the posterior distribution of the 8-dim parameter vector β with a multivariate normal distribution

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

where $\tilde{\beta}$ is the posterior mode and $J(\tilde{\beta}) = -\frac{\partial^2 \ln p(\beta|\mathrm{y})}{\partial\beta\partial\beta^T}|_{\beta=\tilde{\beta}}$ is the observed Hessian evaluated at the posterior mode. Note that $\frac{\partial^2 \ln p(\beta|\mathrm{y})}{\partial\beta\partial\beta^T}$ is an 8x8 matrix with second derivatives on the diagonal and cross-derivatives $\frac{\partial^2 \ln p(\beta|\mathrm{y})}{\partial\beta_i\partial\beta_j}$ on the off-diagonal. It is actually not hard to compute this derivative by hand, but don't worry, we will let the computer do it numerically for you. Now, both $\tilde{\beta}$ and $J(\tilde{\beta})$ are computed by the `optim` function in R. See my code https://github.com/mattiasvillani/BayesLearnCourse/blob/master/Code/MainOptimizeSpam.zip where I have coded everything up for the spam prediction example (it also does probit regression, but that is not needed here). I want you to implement your own version of this. You can use my code as a template, but I want you to write your own file so that you understand every line of your code. Don't just copy my code. Use the prior $\beta \sim \mathcal{N}(0, \tau^2 I)$, with $\tau = 10$. Your report should include your code as well as numerical values for $\tilde{\beta}$ and $J_y^{(-1)}\tilde{\beta}$ for the `WomanWork` data. Compute an approximate 95% credible interval for the variable NSmallChild. Would you say that this feature is an important determinant of the probability that a women works?
[Hint: To verify that your results are reasonable, you can compare to you get by estimating the parameters using maximum likelihood: `glmModel <- glm(Work ~ 0 + ., data = WomenWork, family = binomial)`.]

$$
\begin{aligned}
P(y|X,\beta) =& \prod_{i=1}^{n} \frac{\left(\exp(x_i^T\beta)\right)^{y_i}}{1+\exp(x_i^T\beta)} \\
LogP(y|X,\beta) =& Log \prod_{i=1}^{n} \frac{\left(\exp(x_i^T\beta)\right)^{y_i}}{1+\exp(x_i^T\beta)} \\
=& \sum_{i=1}^{n} log \frac{\left(\exp(x_i^T\beta)\right)^{y_i}}{1+\exp(x_i^T\beta)} \\
=& \sum_{i=1}^{n} x_i^T\beta y_i - \log(1+\exp(x_i\beta))
\end{aligned}
$$

```
##############################################################################
## 1. Linear and polynomial regression
##############################################################################

women <- read.table("Other/Data/WomenWork.dat", header=TRUE)
tau   <- 10

y <- as.vector(women[,1])
X <- as.matrix(women[,2:9])
nPara <- dim(X)[2]

mu    <- as.vector(rep(0, nPara))
Sigma <- tau^2*diag(nPara)

LogPostLogistic <- function(betaVect, y, X, mu, Sigma){
  nPara   <- length(betaVect)
  linPred <- X %*% betaVect

  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(betaVect, matrix(0, nPara, 1), Sigma, log=TRUE);
  return(logLik + logPrior)
}

initVal <- as.vector(rep(0,dim(X)[2]))
OptimResults <-optim(initVal,LogPostLogistic,gr=NULL,y,X,mu,Sigma,method=c("BFGS"),
                     control=list(fnscale=-1),hessian=TRUE)

postMode <- OptimResults$par
postCov  <- -solve(OptimResults$hessian)
names(postMode) <- colnames(X)
colnames(postCov) <- rownames(postCov) <- colnames(X)

glmModel <- glm(Work ~ 0 + ., data = women, family = binomial)

knitr::kable(cbind("Maximum likelihood" = glmModel$coefficients, "Posterior Mode" = postMode))
```

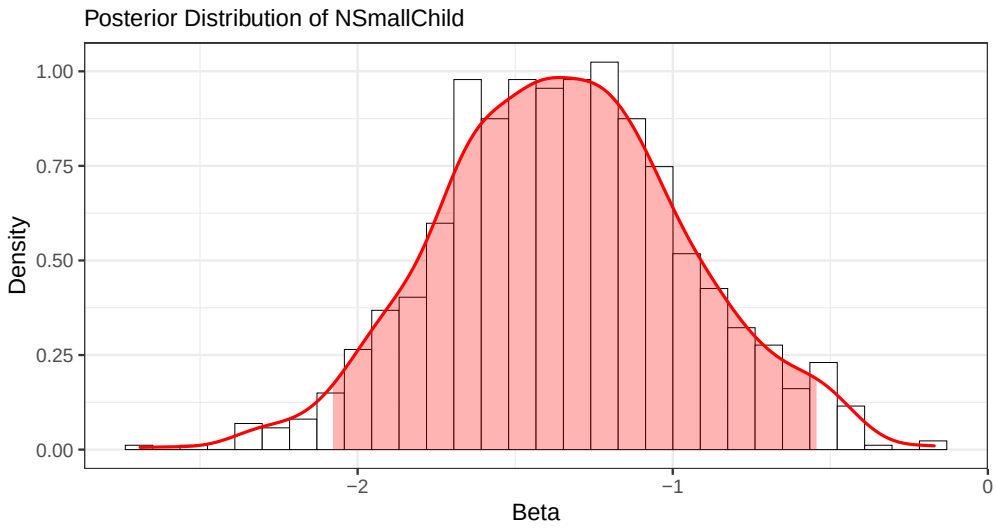
	Maximum likelihood	Posterior Mode
Constant	0.6443036	0.6267288
HusbandInc	-0.0197746	-0.0197911
EducYears	0.1798806	0.1802190

	Maximum likelihood	Posterior Mode
ExpYears	0.1675127	0.1675667
ExpYears2	-0.1443595	-0.1445967
Age	-0.0823403	-0.0820656
NSmallChild	-1.3625024	-1.3591332
NBigChild	-0.0254299	-0.0246835

```
knitr::kable(postCov, digits = 4)
```

	Constant	HusbandInc	EducYears	ExpYears	ExpYears2	Age	NSmallChild	NBigChild
Constant	2.2660	0.0033	-0.0655	-0.0118	0.0458	-0.0303	-0.1887	-0.0980
HusbandInc	0.0033	0.0003	-0.0006	0.0000	0.0001	0.0000	0.0005	-0.0001
EducYears	-0.0655	-0.0006	0.0062	-0.0004	0.0019	0.0000	-0.0061	0.0018
ExpYears	-0.0118	0.0000	-0.0004	0.0044	-0.0142	-0.0001	-0.0015	0.0005
ExpYears2	0.0458	0.0001	0.0019	-0.0142	0.0556	-0.0003	0.0032	0.0005
Age	-0.0303	0.0000	0.0000	-0.0001	-0.0003	0.0007	0.0052	0.0011
NSmallChild	-0.1887	0.0005	-0.0061	-0.0015	0.0032	0.0052	0.1513	0.0068
NBigChild	-0.0980	-0.0001	0.0018	0.0005	0.0005	0.0011	0.0068	0.0200

```
approxPostStd <- sqrt(diag(postCov))


betaValues <- seq(postMode[7] - 3*approxPostStd[7], postMode[7] + 3*approxPostStd[7], length = 10)
betaSample <- rnorm(1000, postMode[7], approxPostStd[7])

q1 <- quantile(betaSample,.025)
q2 <- quantile(betaSample,.975)
dens <- density(betaSample)
dens_df <- data.frame(x = dens$x, y = dens$y)

ggplot(as.data.frame(betaSample)) +
  geom_histogram(aes(x=betaSample, y=..density..), bins=30, color="black", fill="#ffffffff", size=0.1) +
  geom_density(aes(x=betaSample, y=..density..), color = "red", size = 0.7) +
  geom_area(data = subset(dens_df, x >= q1 & x <= q2),
            aes(x=x,y=y), fill = 'red', alpha = 0.3) +
  labs(subtitle="Posterior Distribution of NSmallChild", y="Density", x="Beta") +
  theme_bw() +
  theme(legend.position="bottom", legend.title = element_blank())
```



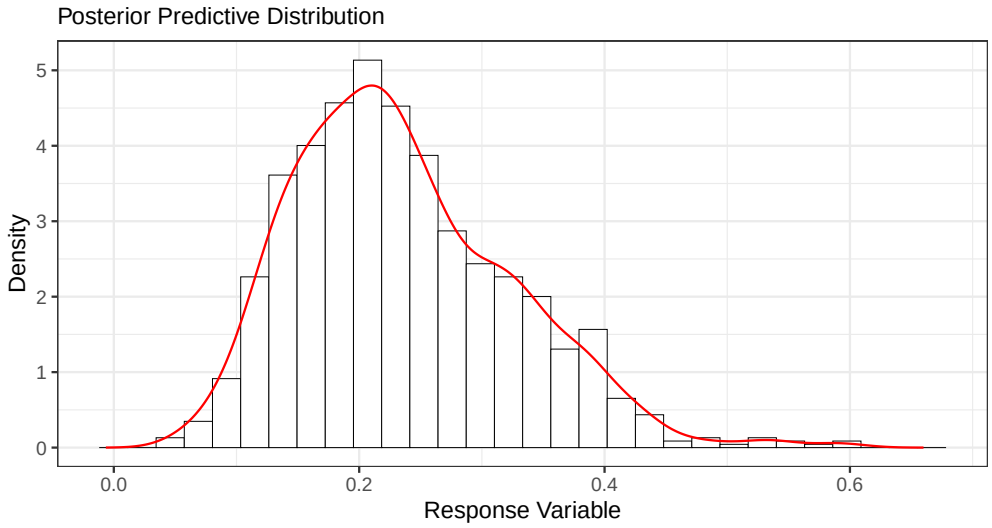

```
cat("Equal Tail Interval:", q1, "-", q2)
```

```
## Equal Tail Interval: -2.079507 - -0.542211
```

- Yes, this feature is the main one that correlate with the women not working, since the posterior mode for the absolute value of the beta is relatively much higher than the other features and the negative sign is inverse relatioship with Work

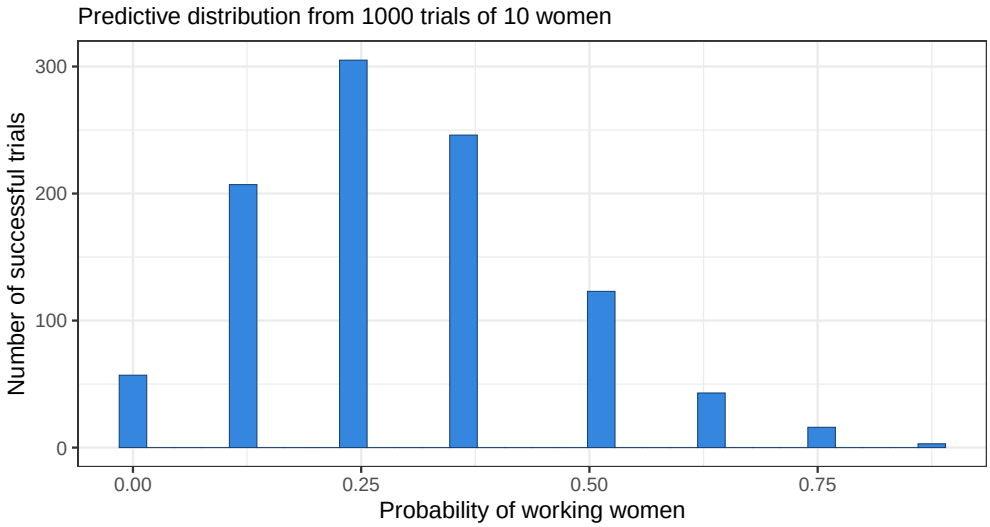
(b) Write a function that simulates from the predictive distribution of the response variable in a logistic regression. Use your normal approximation from 2(a). Use that function to simulate and plot the predictive distribution for the `Work` variable for a 40 year old woman, with two children (3 and 9 years old), 8 years of education, 10 years of experience. and a husband with an income of 10. [Hints: The R package `mvtnorm` will again be handy. Remember my discussion on how Bayesian prediction can be done by simulation.]

```
X_pred <- matrix(c(1, 10, 8, 10, (10/10)^2, 40, 1, 1))

simulator <- function(n, X, mu, Sigma) {
  beta     <- rmvnorm(n, mean = mu, sigma = Sigma)
  logistic <- exp(beta %*% X) / (1 + exp(beta %*% X))
  #return(plogis(beta %*% X))
  return(logistic)
}

pred_dist <- simulator(n = 1000, X = X_pred, mu = postMode, Sigma = postCov)
pred_dens <- density(pred_dist)
pred_dens_df <- data.frame(pred_dens$x, pred_dens$y)

ggplot(as.data.frame(pred_dist)) +
  geom_histogram(aes(x = pred_dist, y=..density..),bins = 30, color = "black", fill = "#ffffffff", size
  geom_line(data = pred_dens_df, aes(x = pred_dens.x, y = pred_dens.y), color = "red") +
  labs(subtitle="Posterior Predictive Distribution", y="Density", x="Response Variable") +
  theme_bw() +
  theme(legend.position="bottom", legend.title = element_blank())
```



(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?]

```
X_pred <- matrix(c(1, 10, 8, 10, (10/10)^2, 40, 1, 1))

binom_sim  <- function(n, X, mu, Sigma) {
  logistic <- vector(length = dim(X)[1])
  binom    <- vector(length = n)
  for(i in 1:n){
    beta     <- rmvnorm(i, mean = mu, sigma = Sigma)
    logistic <- exp(beta %*% X) / (1 + exp(beta %*% X))
    trials   <- rbinom(n = 10, size = 1, prob = logistic)
    binom[i] <- sum(trials) / dim(X)[1]
  }
  return(binom)
}

binom_dist <- binom_sim(n = 1000, X = X_pred, mu = postMode, Sigma = postCov)
binom_dens <- density(binom_dist)
binom_dens_df <- data.frame(binom_dens$x, binom_dens$y)

ggplot(as.data.frame(binom_dist)) +
  geom_histogram(aes(x = binom_dist), bins = 30, color = "#113B69", fill = "#3486DF", size = 0.1) +
  labs(subtitle="Predictive distribution from 1000 trials of 10 women", y="Number of successful trials",
  theme_bw() +
  theme(legend.position="bottom", legend.title = element_blank())
```


Predictive distribution from 1000 trials of 10 women
Number of successful trials
300
200
100
0
0.00
0.25
0.50
0.75
Probability of working women