Lab 2

2024-04-16

Question 1 - Linear and polynomial regression

A Bayesian analysis of the following quadratic regression model is to be performed (But when we doing the problem, we dont need to include the "eplison":

$$temp = \beta 0 + \beta 1 \cdot time + \beta 2 \cdot time + \varepsilon, \varepsilon \sim N(0, \sigma^2).$$

a)

The prior hyperparameters are given.

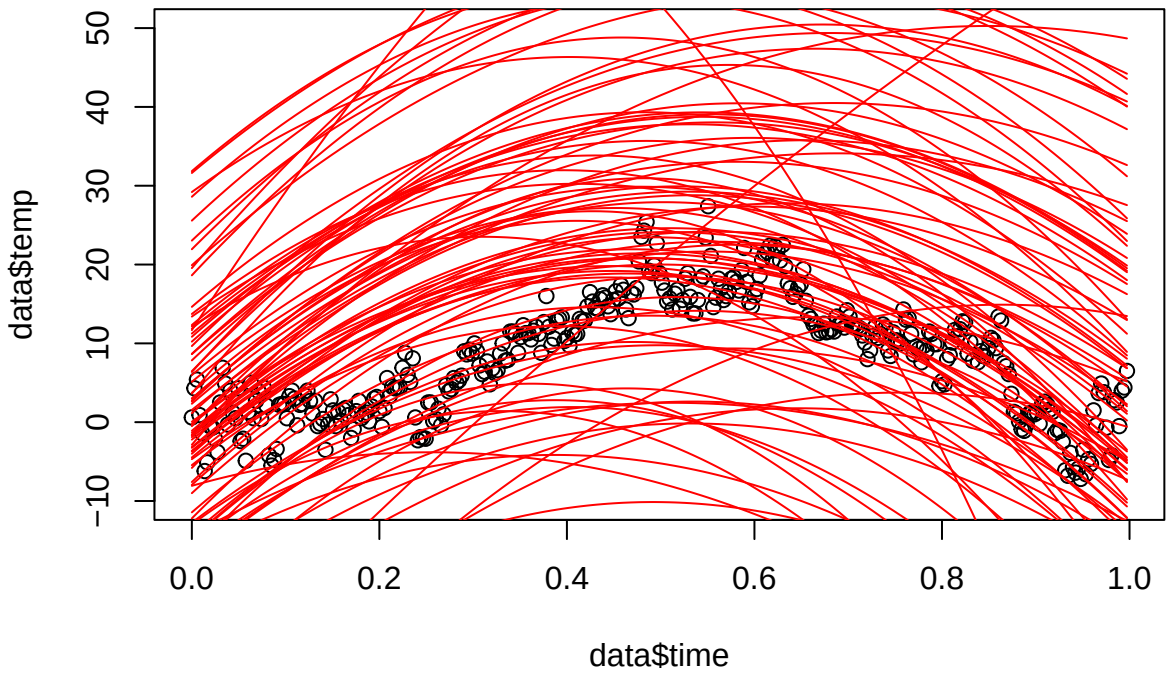
$$\sigma 2 \sim Inv - \chi 2(\nu 0, \sigma 0^2)$$

The above sigma2 is calculated the same way as lab-1.

$$\beta|\sigma \sim N(\mu 0, \sigma^2 * \Omega^{-}1)$$

```
data <- readxl::read_xlsx("Linkoping2022.xlsx")

len<-nrow(data) #365

time <- (as.Date(data$datetime)-as.Date(data$datetime[1]))/365

data$time<-as.numeric(time)
v_0<-1
sigma_0<-1
mu_0 <- matrix(c(0,100,-100),byrow = TRUE)

#Inverse chi-square
inv_chi_draw <- c()
beta_parameters<- list()
omega <- 0.01*diag(3)

post_draw <- function(n){
  for(i in 1:n){
    chi_draw <- rchisq(1,v_0)
    inv_chi_draw[i]<<-(v_0*sigma_0)/(chi_draw)

    variance<- solve(omega)*inv_chi_draw[i]

    beta_parameters[i]<<-list(c(rmvnorm(1,c(0, 100, -100),variance)))
```

```
  }
  #hist(inv_chi_draw,breaks=1000)
}

post_draw(1000)

plot(data$time,data$temp, ylim=c(-10, 50))

for (i in 1:100){
  prediction<-c()
  for(j in 1:365){
    prediction[j]<-beta_parameters[[i]][1]+beta_parameters[[i]][2]*
      data$time[j] + beta_parameters[[i]][3]*(data$time[j]^2)
  }
  lines(data$time,prediction,col = "red")
}
```



Yes, the curves does seem to look reasonable.

b)

(b) Write a function that simulate draws from the joint posterior distribution of beta0, beta1,beta2 and sigma^2.

i. Plot a histogram for each marginal posterior of the parameters.

$$\mu n = (X'X + \Omega 0)^{-}1 * (X'X\beta + \Omega 0 * \mu 0)$$

where

$$\beta = (X'X)^{-}1 * X'y$$

$$\Omega n = X'X + \Omega 0$$

$$\nu n = \nu 0 + n$$

$$\nu n \sigma n^2 = \nu 0 \sigma 0^2 + (y'y + \mu 0' * \Omega 0 * \mu 0 - \mu n' * \Omega n * \mu n)$$

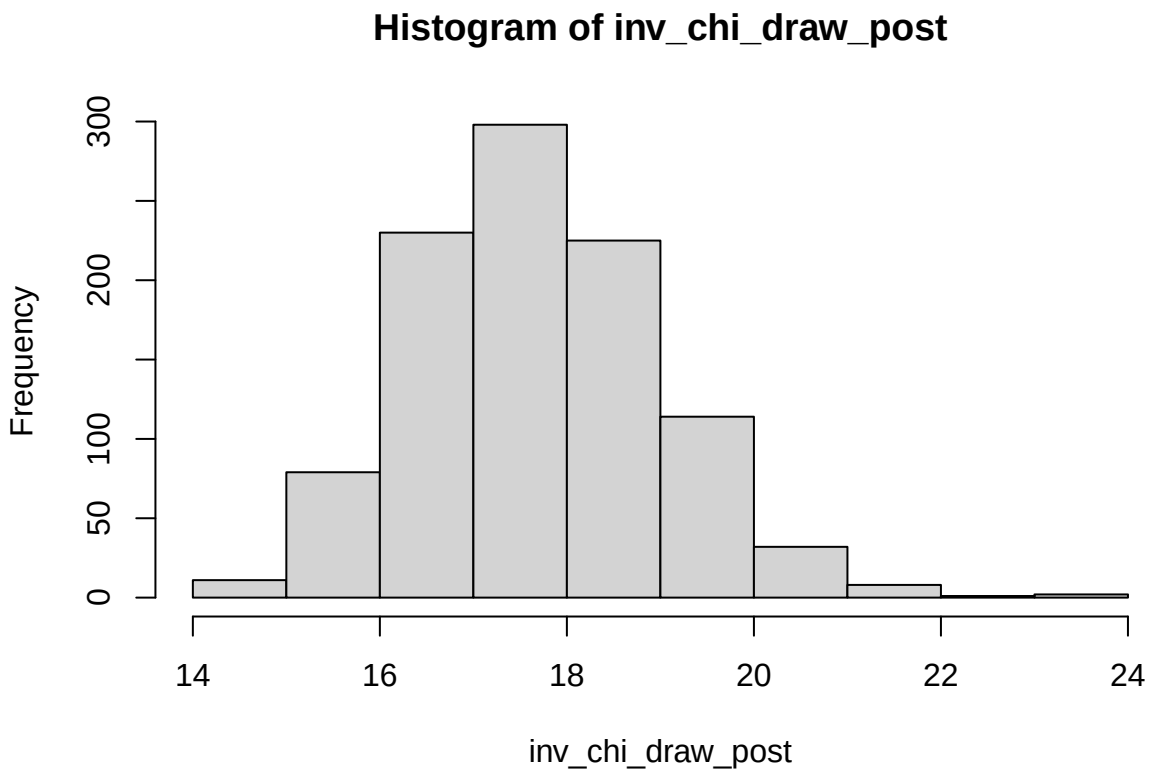
Posterior:

$$\sigma 2|y \sim Inv - \chi 2(\nu n, \sigma n^2)$$

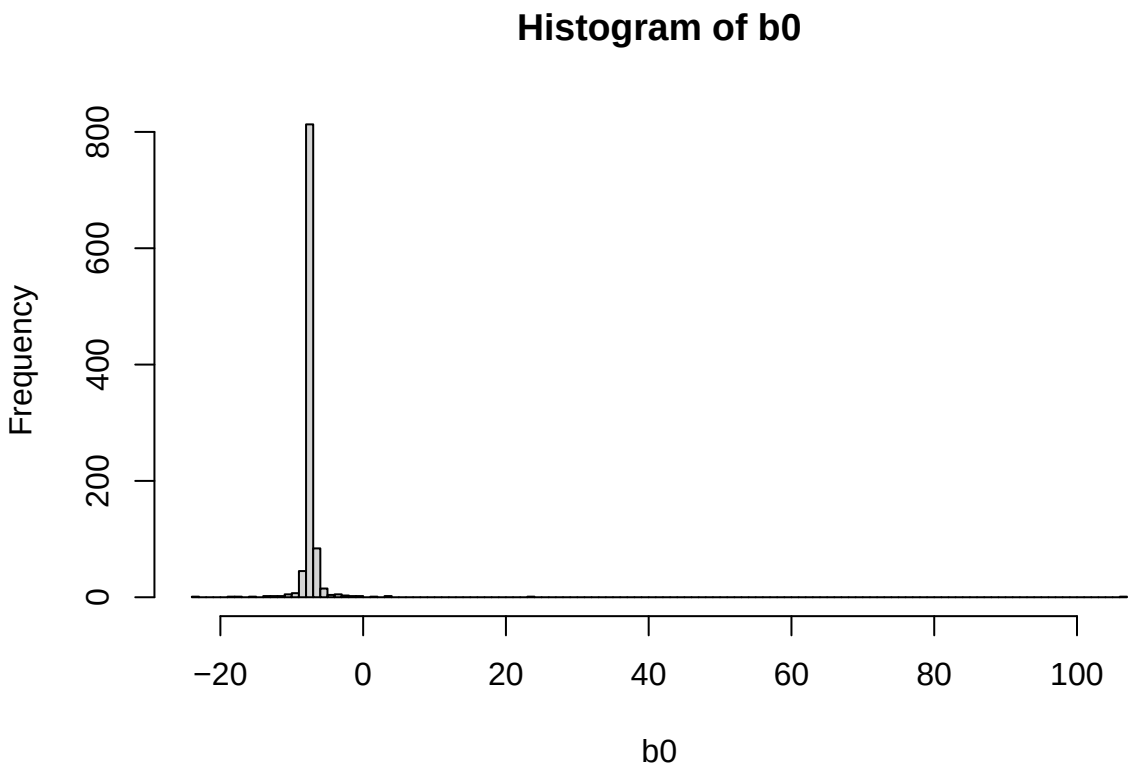
$$\beta|\sigma, y \sim N(\mu n, \sigma^2 * \Omega n^{-}1)$$

```
#i)

inv_chi_draw_post <- c()
beta_parameters_post<- list()
joint_post<-function(n){
  X_value<- matrix(c(rep(1, 365),data$time,data$time^2),
     nrow = 365, ncol = 3, byrow = FALSE)
  mu_inital <- solve(t(X_value)%*%X_value+omega)
  a <- t(X_value)%*%X_value
  b <-solve(t(X_value)%*%X_value)%*%t(X_value)%*%data$temp
  c <- omega%*%mu_0

  mu_n<-mu_inital %*% (a%*%b+c)

  Omega_n <-t(X_value)%*%X_value+omega
  v_n<-v_0+365

  a_1<-v_0*sigma_0+((t(matrix(data$temp))%*%matrix(data$temp))+t(mu_0)%*%
                      omega%*%mu_0- t(mu_n)%*%Omega_n%*%mu_n)
  sigma_n<-a_1/v_n


  for(i in 1:n){
    chi_draw <- rchisq(1,v_n)
    inv_chi_draw_post[i]<<-((v_n)*sigma_n)/(chi_draw)

    variance_n<- solve(Omega_n)*inv_chi_draw[i]

    beta_parameters_post[i]<<-list(c(rmvnorm(1,c(mu_n),variance_n)))
  }

}
```

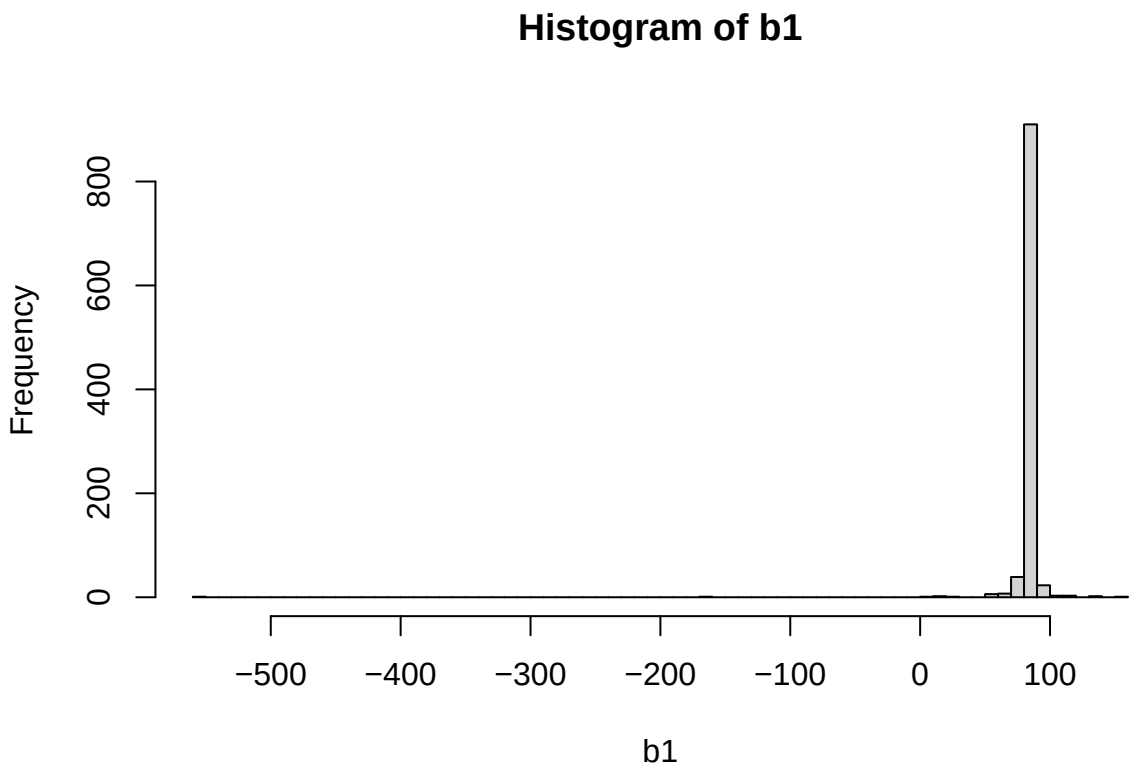
```r
n<-1000
joint_post(n)


hist(inv_chi_draw_post)
```



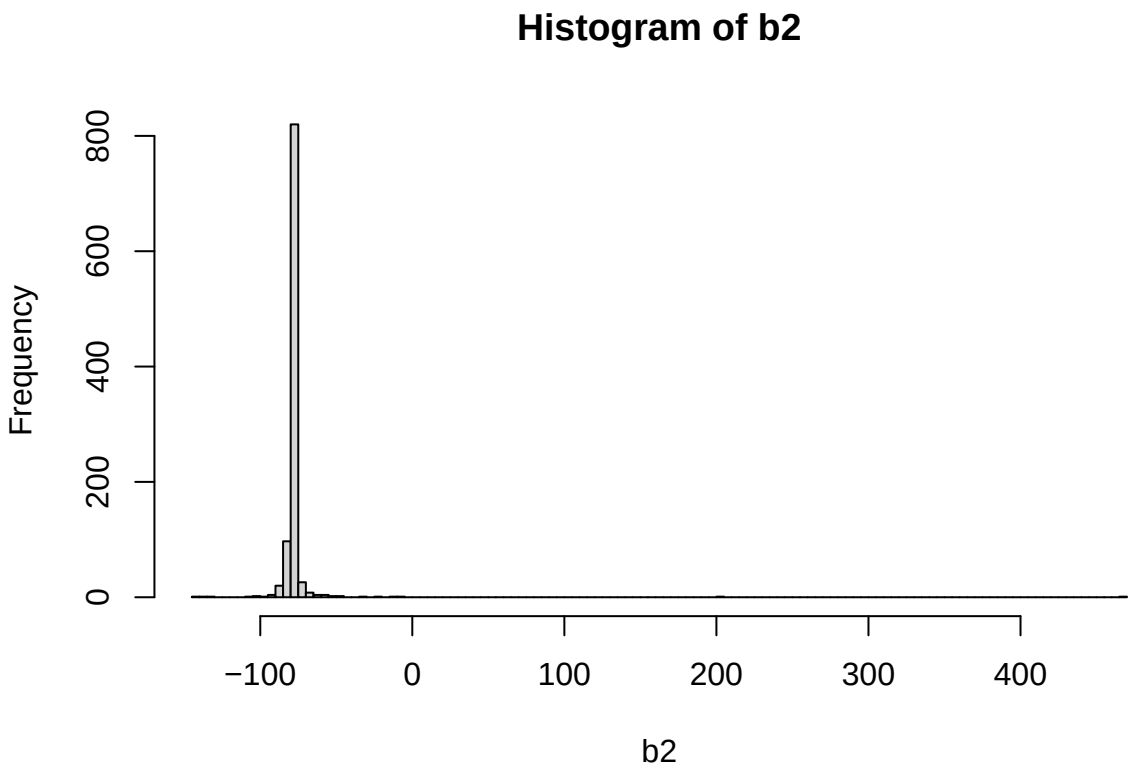

```r
b0<-rep(NA,n)
b1<-rep(NA,n)
b2<-rep(NA,n)

for(i in 1:n){
  b0[i]<-beta_parameters_post[[i]][1]
  b1[i]<-beta_parameters_post[[i]][2]
  b2[i]<-beta_parameters_post[[i]][3]
}


hist(b0,breaks=100)
```




```
hist(b1,breaks=100)
```



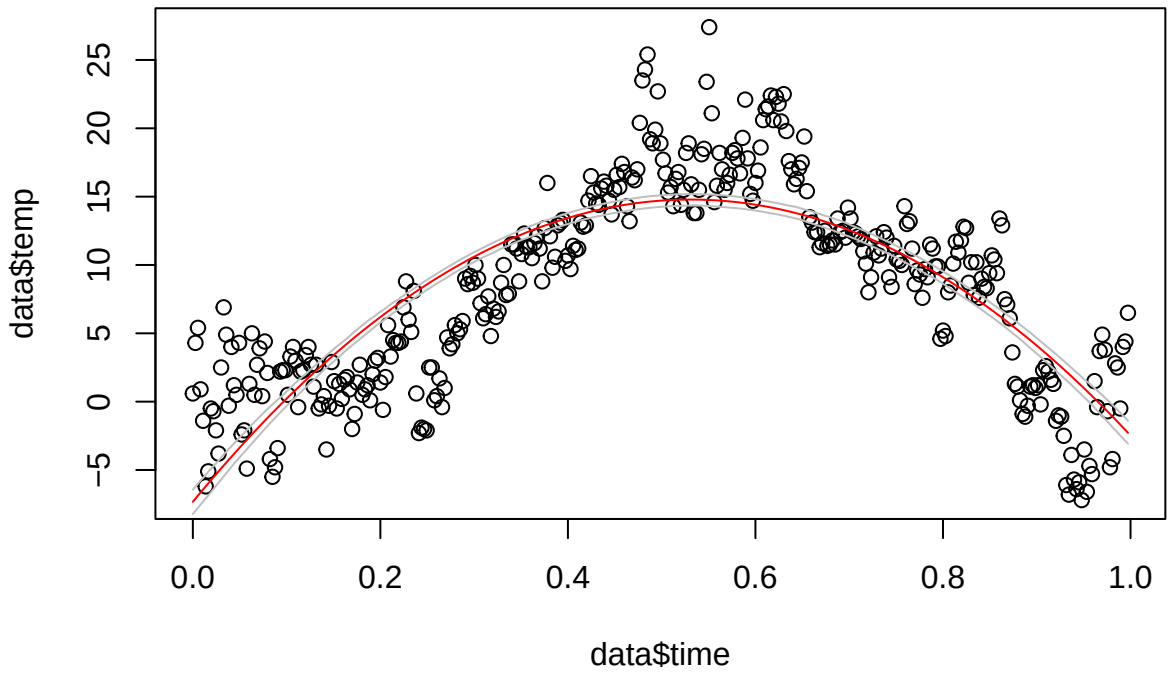

```
hist(b2,breaks=100)
```



Make a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function f(time) = E[temp|time] = beta0 + beta1 · time + beta2 · time2, i.e. the median of f (time) is computed for every value of time. In addition, overlay curves for the 90% equal tail posterior probability intervals of f(time), i.e. the 5 and 95 posterior per- centiles of f (time) is computed for every value of time.

```
#ii)
plot(data$time,data$temp)

median_post<-c()
low_quantile<-c()
high_quantile<-c()

for (i in 1:365){
  prediction_post<-c()
  for(j in 1:n){
    prediction_post[j]<-beta_parameters_post[[j]][1]+
      beta_parameters_post[[j]][2]*data$time[i] +
      beta_parameters_post[[j]][3]*(data$time[i]^2)
  }
  median_post[i]<-median(prediction_post)
  #time_max_temp[i]<-data$time[which.max(prediction_post)]
  low_quantile[i]<-quantile(prediction_post, probs = 0.05)
  high_quantile[i]<-quantile(prediction_post, probs = 0.95)
}

lines(data$time,median_post,col = "red")
```

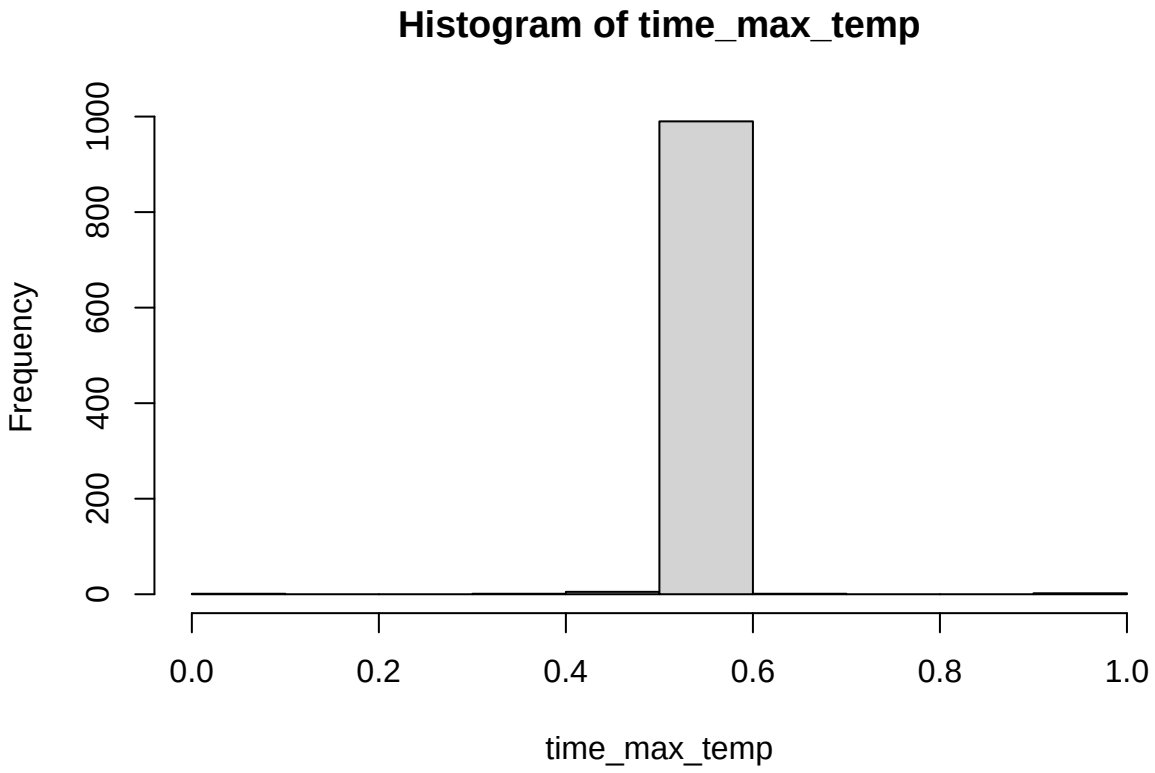
```
lines(data$time,low_quantile,col = "gray")
lines(data$time,high_quantile,col = "gray")
```



c)

It is of interest to locate the time with the highest expected temperature (i.e. the time where f(time) is maximal). Let's call this value $\tilde{x}$. Use the simulated draws in (b) to simulate from the posterior distribution of $\tilde{x}$. [Hint: the regression curve is a quadratic polynomial. Given each posterior draw of 0, 1 and 2, you can and a simple formula for $\tilde{x}$.]

```
time_max_temp<-c()
values<-c()

for (i in 1:n){
  for(j in 1:365){
    values[j]<-beta_parameters_post[[i]][1]+beta_parameters_post[[i]][2]*
      data$time[j] + beta_parameters_post[[i]][3]*(data$time[j]^2)
  }
  time_max_temp[i]<-data$time[which.max(values)]
}

hist(time_max_temp)
```



d)

Say now that you want to estimate a polynomial regression of order 10, but you suspect that higher order terms may not be needed, and you worry about overftting the data. 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 mu0 and omega0 in a suitable way.]

So we can either use for the prior:

$$\beta|\sigma 2 \sim N(0, \sigma^2/\lambda)$$

or:

$$\beta|\sigma 2 \sim Laplace(0, \sigma^2/\lambda)$$

For Laplace prior, the tails in distribution die off slowly.Many betas close to zero, but some betas very large. For Normal prior, the tails in distribution die off rapidly. All betas are similar in magnitude.

For $\beta|\sigma^2$, it would be better to use a Normal prior with the below values for mu and Omega as that would put the coefficients to zero i.e removes features accordingly. So

$$\mu_0 = 0$$

$$\Omega_0 = \lambda * I$$

Since we want to remove the higher order terms, we can have a higher "lambda" value to remove them thus helping with to prevent overfitting.

Question 2 - Posterior approximation for classification with logistic regression

a)

Consider the logistic regression model:

$$Pr(y = 1|x, \beta) = exp(x^T\beta)/(1 + exp(xT\beta))$$

where y equals 1 if the woman works and 0 if she does not. x is a 7-dimensional vector containing the seven features (including a 1 to model the intercept).

Likelihood of the above:

$$p(y|X, \beta) = \prod[exp(xi^t\beta)]^y i/(1 + exp(xi^t\beta).)$$

The goal is to approximate the posterior distribution of the parameter vector beta with a multivariate normal distribution

$$\beta|y, x \sim N(\beta, Jy^-1(\beta\ ))$$

where beta˜ is the posterior mode and J(beta˜) is the negative of the observed Hessian evaluated at the posterior mode. Calculate both beta˜ and J(beta˜) by using the optim function in R.Use the prior beta N(0,to^2*I), where to = 2.

Present the numerical values of beta˜ and J^−1(beta˜) for the WomenAtWork data. Compute an approximate 95% equal tail posterior probability interval for the regresion coefficient to the variable NSmallChild. Would you say that this feature is of importance for the probability that a woman works?

Normal approximation to posterior:

Input: expression proportional to log p(theta|y). Initial values. Output: log p(theta˜|y),theta˜ and Hessian matrix (−Jy(theta˜)).

```
# Load data
women_data <- as.matrix(read.table("WomenAtWork.dat", header=TRUE))
n_params <- ncol(women_data) - 1
n_obs <- nrow(women_data)

y <- women_data[1:n_obs, 1]
X <- women_data[1:n_obs, 2:ncol(women_data)]
x_names <- colnames(X)

# Set up prior
tau <- 2
mu <- matrix(0, n_params) # Prior mean vector
Sigma <- tau ** 2 * diag(n_params)

log_post_logistic <- function(betas, y, X, mu, Sigma) {
  lin_pred <- X %*% betas
  log_lik <- sum(lin_pred * y - log(1 + exp(lin_pred)))
  log_prior <- dmvnorm(betas, mu, Sigma, log=TRUE)

  return(log_lik + log_prior)
}
```

```r
# Initial values for beta
init_val <- matrix(0, n_params)

# Optimize betas
optim_res <- optim(init_val, log_post_logistic, gr=NULL, y, X, mu, Sigma,
                   method=c("BFGS"), control=list(fnscale=-1), hessian=TRUE)
rownames(optim_res$par) <- x_names

approx_post_std <- sqrt(diag(solve(-optim_res$hessian)))
names(approx_post_std) <- x_names

# Print results
cat("Posterior mode:\n", optim_res$par,
    "\n\nApprox. posterior std.:\n", approx_post_std)
```

```
## Posterior mode:
##  -0.04036943 -0.03730689 0.1786895 0.1207364 -0.04618995 -1.472489 -0.02014458
##
## Approx. posterior std.:
##  1.381985 0.02198474 0.0892096 0.03335982 0.02747315 0.4774676 0.1640196
```

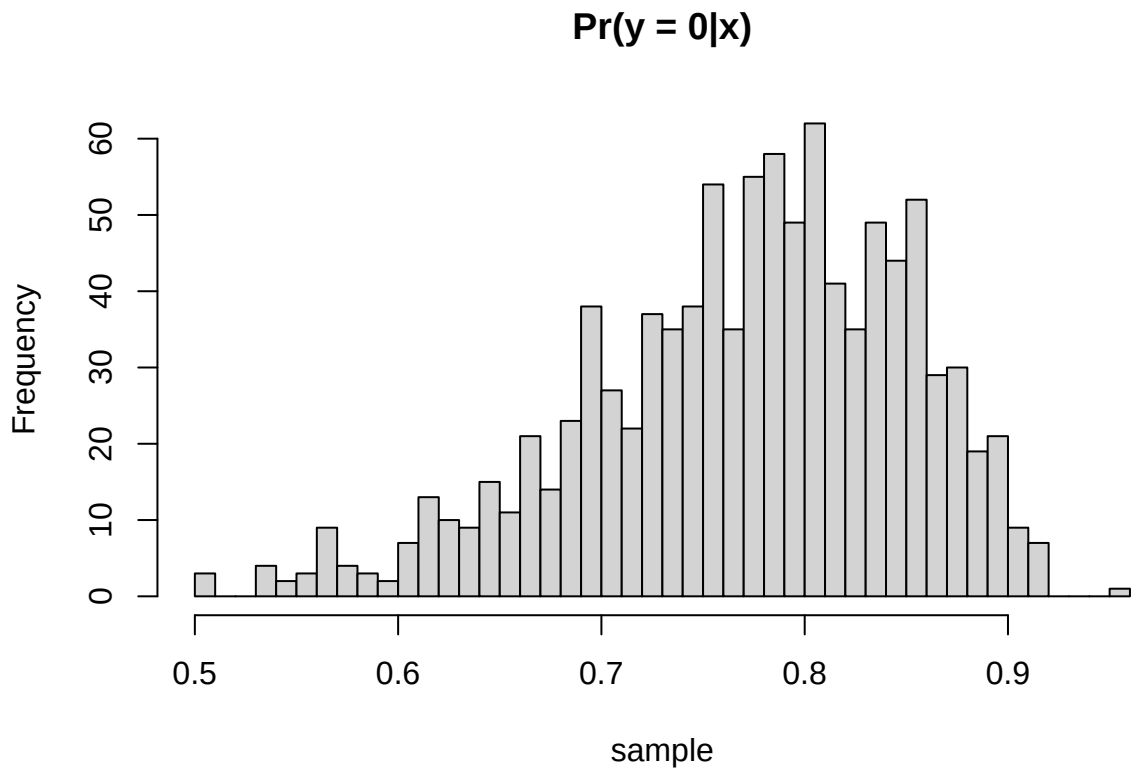
```r
# Check values
# glm(Work ~ 0 + ., data=as.data.frame(women_data), family=binomial)

# Get 95% equal tail posterior prob. interval for NSmallChild
low_tail <- qnorm(0.025, mean=optim_res$par["NSmallChild",],
                  sd=approx_post_std["NSmallChild"])
high_tail <- qnorm(0.975, mean=optim_res$par["NSmallChild",],
                   sd=approx_post_std["NSmallChild"])

cat("Approx. 95% equal tail posterior prob. interval for NSmallChild:\n",
    paste0("[", round(low_tail, 3), ", ", round(high_tail, 3), "]"))
```

```
## Approx. 95% equal tail posterior prob. interval for NSmallChild:
##  [-2.408, -0.537]
```

Since 0 is not included in this interval, the variable NSmallChild has a significant effect on whether or not a woman works.

```r
# Function to simulate draws from Pr(y = 0|x)
sample_post_pred_y0 <- function(x, n_draws, post_mean, post_sigma) {
  sample <- c()

  for (i in 1:n_draws) {
    beta_draw <- t(rmvnorm(1, mean=post_mean, sigma=post_sigma))
    prob_y0 <- 1 - (exp(t(x) %*% beta_draw) / (1 + exp(t(x) %*% beta_draw)))

    sample <- append(sample, prob_y0)
  }

  return(sample)
```

```
}

# Plot distribution
x <- c(1, 18, 11, 7, 40, 1, 1) # Woman from question
sample <- sample_post_pred_y0(x, 1000, post_mean=optim_res$par,
                              post_sigma=solve(-optim_res$hessian))

hist(sample, breaks=50, main="Pr(y = 0|x)")
```




```
# Function to simulate draws of number of women who aren't working
sample_n_not_working <- function(x, n_draws, n_equal_women,
                                 post_mean, post_sigma) {
  sample <- c()

  for (i in 1:n_draws) {
    beta_draw <- t(rmvnorm(1, mean=post_mean, sigma=post_sigma))
    prob_y0 <- exp(t(x) %*% beta_draw) / (1 + exp(t(x) %*% beta_draw))
    n_not_working <- sum(rbinom(n_equal_women, 1, prob_y0))

    sample <- append(sample, n_not_working)
  }

  return(sample)
}
```

```r
# Plot distribution
x <- c(1, 18, 11, 7, 40, 1, 1) # Woman from question
sample <- sample_n_not_working(x, 1000, 13, post_mean=optim_res$par,
                               post_sigma=solve(-optim_res$hessian))
hist(sample, main="Pr(No. women not working = z|x)")
```