Bayesian Lab2 Question1

Manu Jain

2025-04-30

QUESTION 1

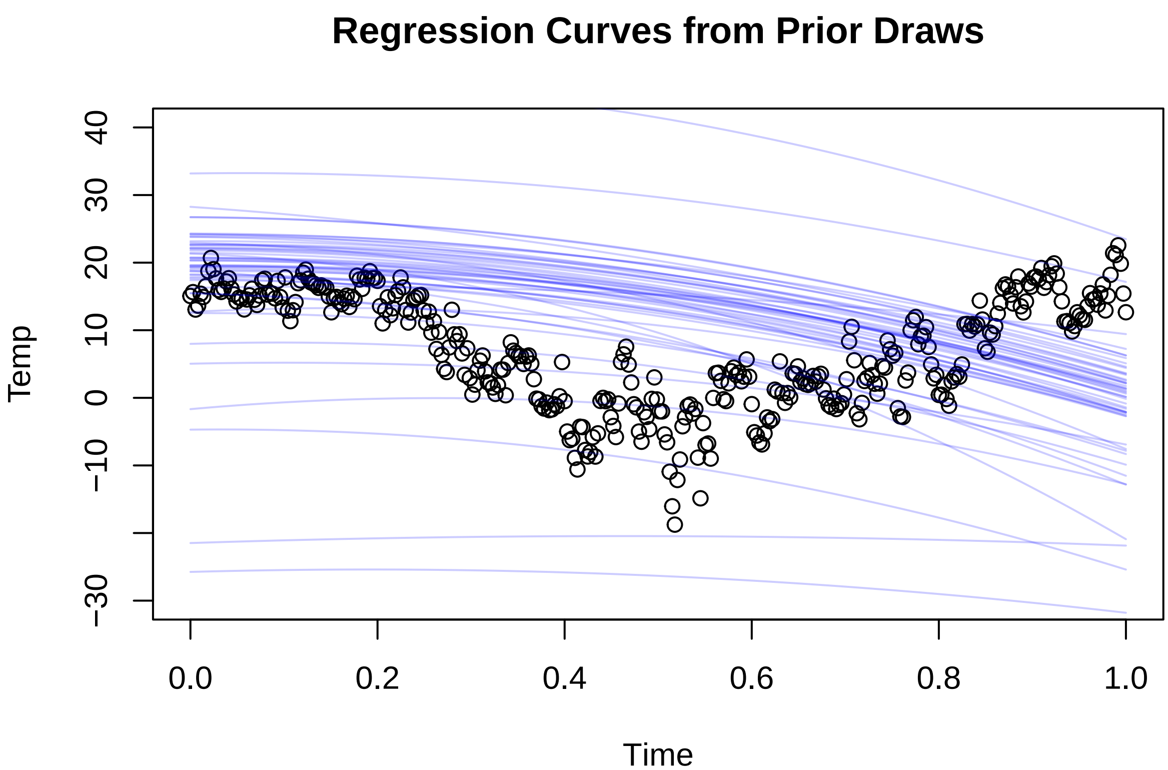
Part A)

```
###### Bayesian Learning Lab 2 ######
############ Question 1 #############

# Libraries Used
library(MASS)
library(mvtnorm)
```

```
## Warning: package 'mvtnorm' was built under R version 4.4.2
```

```
#### PART A)

# loading dataset
data <- read.csv("C:/Users/Dell/OneDrive - Linköpings universitet/732A73 Bayesian Learning/Labs/Lab2/te

data$time2 <- data$time^2
x <- as.matrix(cbind(1, data$time, data$time2))
y <- data$temp
n <- length(y)

# Given Prior Hyperparamters
# mu0 <- c(0, 100, -100)
#omega0 <- 0.01* diag(3)
mu0 <- c(20, 0, -20) # modified
omega0 <- diag(c(0.1, 1, 1)) # modified
nu0 <- 1
sigma02 <- 1

# Evaluate time points
time_seq <- seq(0, 1, length.out = 100)
X_plot <- cbind(1, time_seq, time_seq^2)

# Number of prior draws
S <- 50

# Simulating prior draws, beta and plot
beta_prior_draws <- matrix(0, S, 3)
```

```
sigma2_prior_draws <- (nu0*sigma02)/rchisq(S,df = nu0)

plot(data$time, data$temp, xlab = "Time", ylab = "Temp", ylim = c(-30,40),
     main = "Regression Curves from Prior Draws")

for(s in 1:S){

  cov_beta <- sigma2_prior_draws[s] * solve(omega0)
  beta_prior <- rmvnorm(1, mu0, cov_beta)
  beta_prior_draws[s,] <- beta_prior

  y_pred <- X_plot %*% t(beta_prior)
  lines(time_seq, y_pred, col = rgb(0, 0, 1, alpha = 0.2))
}
```



Observation -

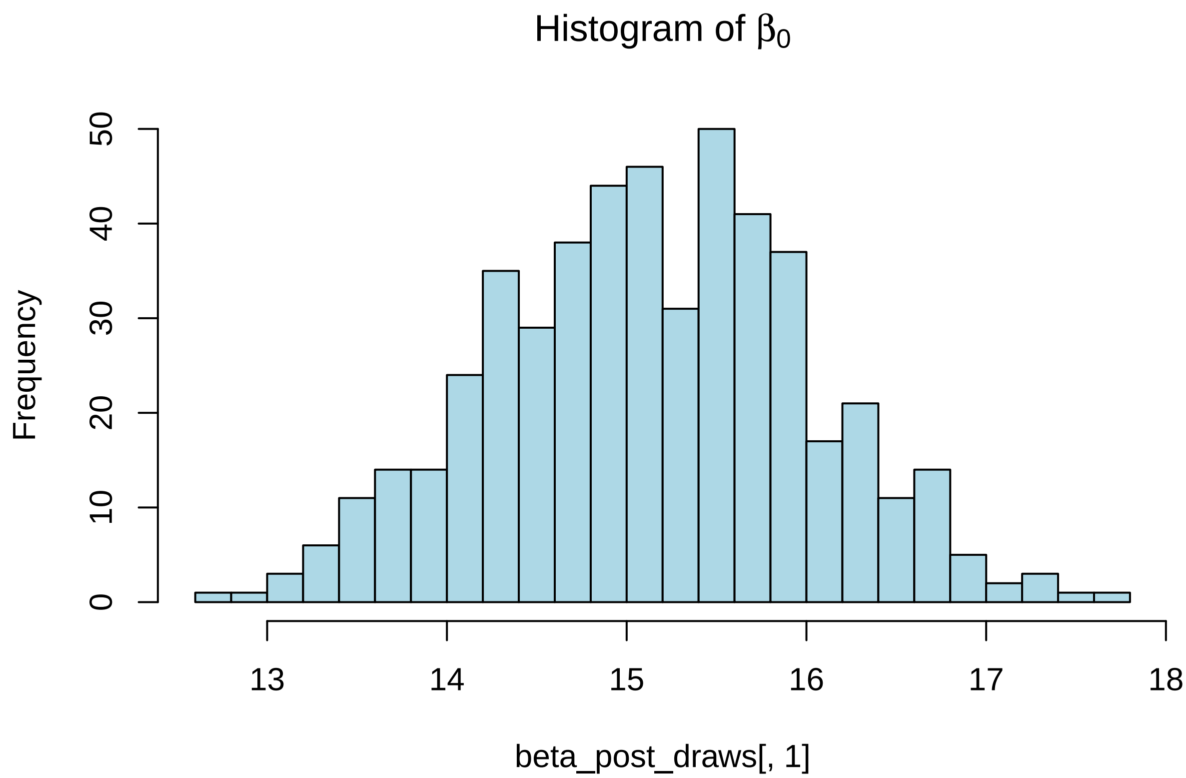
The given dataset is related to temperature in Linkoping city at different point of time. Since, the temperature varies seasonally so the prior might have $\mu0$ = c(-20, 0, 20) and moderate uncertainty that is, $\Omega0$ = c(0.1, 1, 1). The regression curves look reasonable because they fall within the range of plausable temperature value of [-20, 20]

PART B)

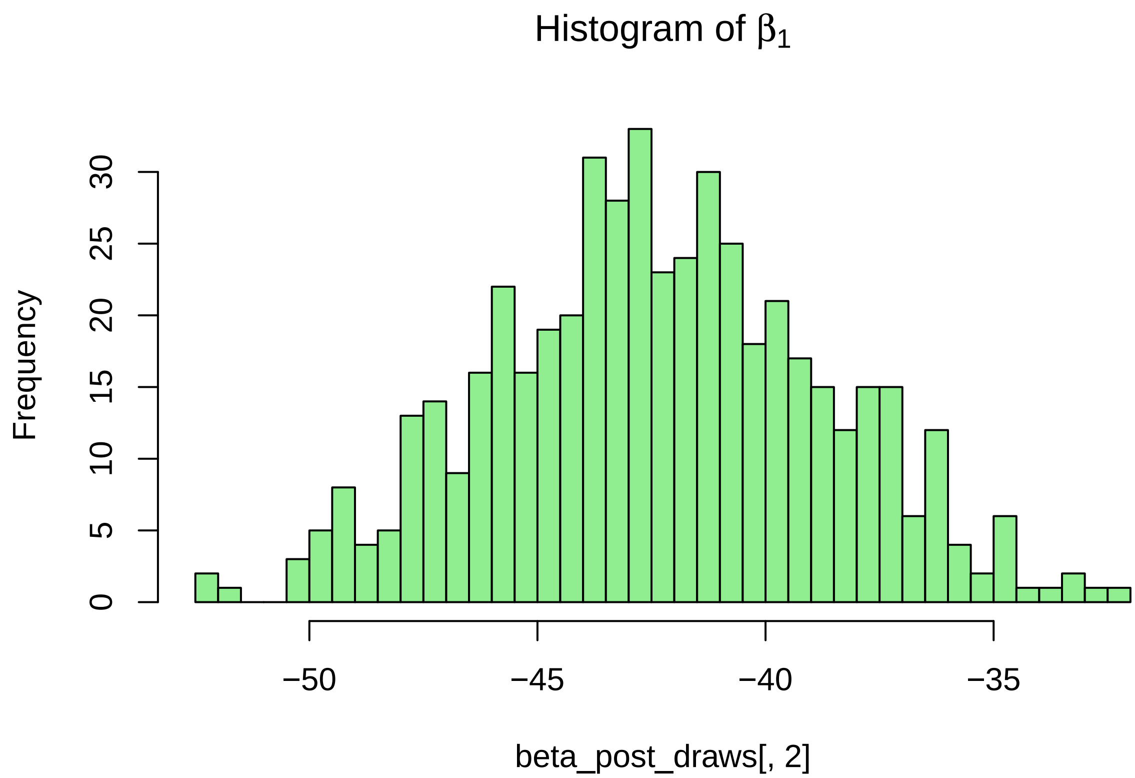
```
#### PART B)

xtx <- t(x) %*% x
xty <- t(x) %*% y

# Posterior Parameters
mu_n <- solve(xtx + omega0) %*% (xty + omega0 %*% mu0)
omega_n <- xtx + omega0
nu_n <- nu0 + n
sigma_n2 <- (nu0*sigma02 + sum(y^2) + t(mu0) %*% omega0 %*% mu0 - t(mu_n)
             %*% omega_n %*% mu_n)/ nu_n

# Simulate Posterior Samples
N <- 500
sigma2_post_draws <- c((nu_n*sigma_n2)/rchisq(N,df = nu_n))
```

```
## Warning in (nu_n * sigma_n2)/rchisq(N, df = nu_n): Recycling array of length 1 in array-vector arithm
##   Use c() or as.vector() instead.
```

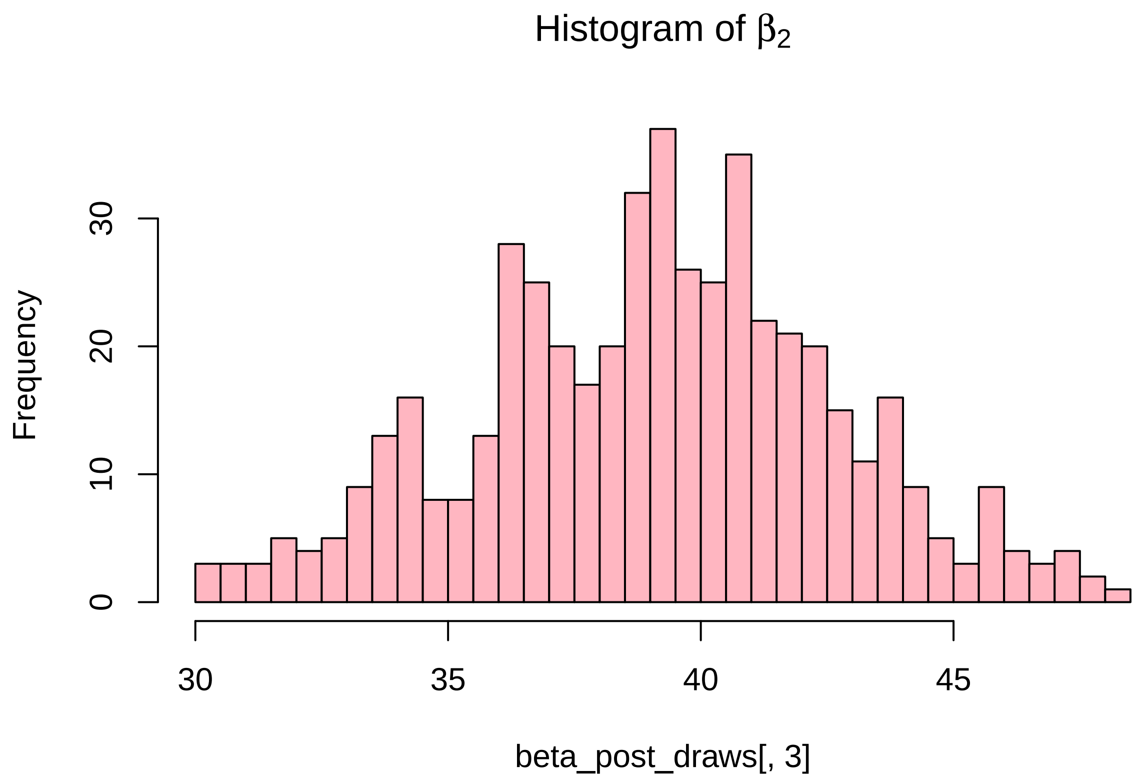
```
beta_post_draws <- matrix(0, N, 3)

for (i in 1:N){
  beta_post_draws[i,] <- rmvnorm(1, mu_n, sigma2_post_draws[i] * solve(omega_n))
}

# Plot Posterior Histogram
hist(beta_post_draws[,1], main = expression("Histogram of " * beta[0]),
     breaks = 30, col = "lightblue")
```



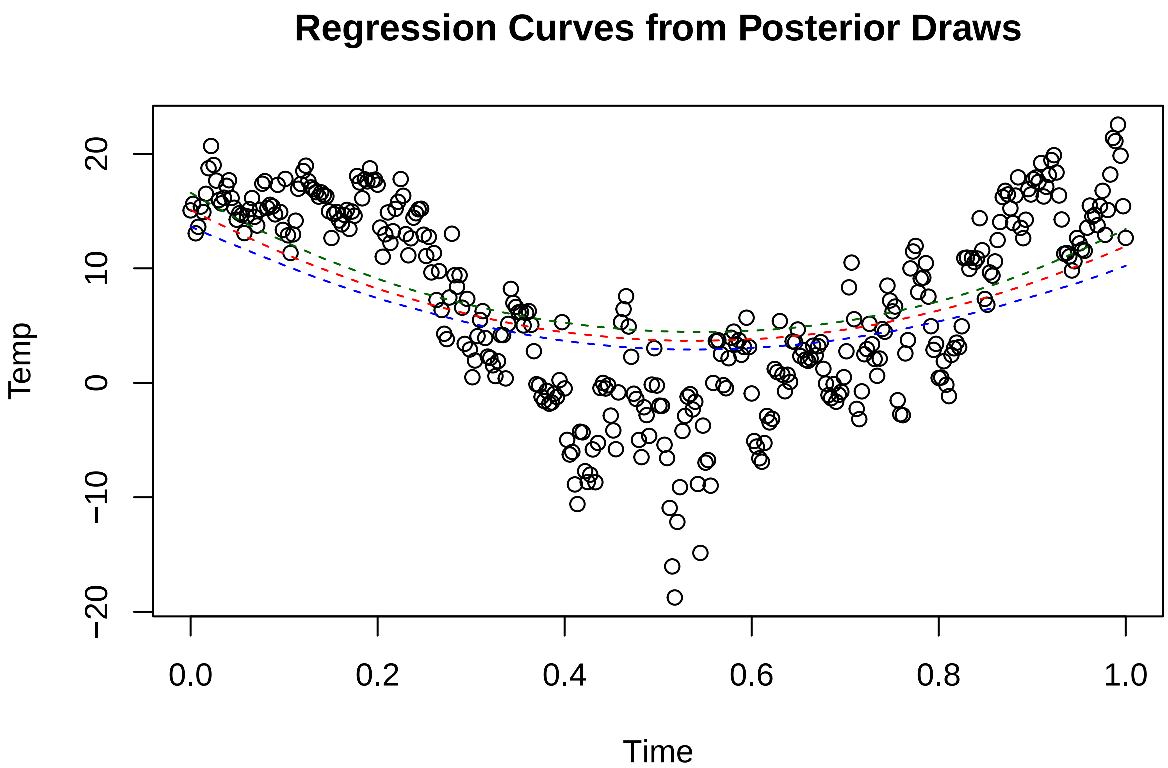

```
hist(beta_post_draws[,2], main = expression("Histogram of " * beta[1]),
     breaks = 30, col = "lightgreen")
```




```r
hist(beta_post_draws[,3], main = expression("Histogram of " * beta[2]),
     breaks = 30, col = "lightpink")
```




```r
# For each time, compute posterior predictive curve
f_post <- apply(beta_post_draws, 1, function(b) X_plot %*% b)

# Compute pointwise statistics
f_median <- apply(f_post, 1, median)
f_lower <- apply(f_post, 1, quantile, probs = 0.05)
f_upper <- apply(f_post, 1, quantile, probs = 0.95)

# Plot
plot(data$time, data$temp, xlab = "Time", ylab = "Temp",
     main = "Regression Curves from Posterior Draws")
lines(time_seq, f_lower, col = "blue", lty = 2)
lines(time_seq, f_upper, col = "darkgreen", lty = 2)
lines(time_seq, f_median, col = "red", lty = 2)
```



Observation-

The posterior probability interval does not contain most of the data points. The actual data points also contain noise $\epsilon \sim N(0, \sigma^2)$. So, to get the full dataset in the given range, variance(σ^2) is to be included in the prediction interval.

PART C)

Computing time with lowest temperature -

$$f(t) = \beta 0 + \beta 1 * t + \beta 2 * t^2$$

and

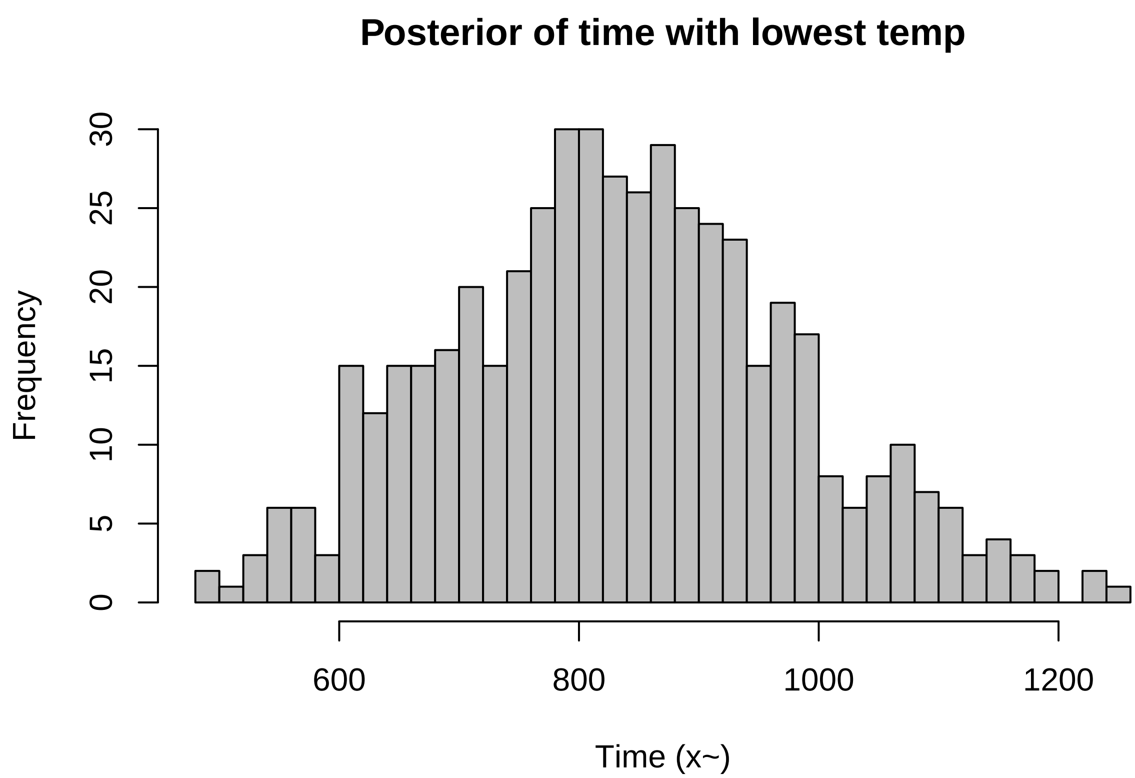
$$f'(t) = \beta 1 + 2 * \beta 2 * t$$

Now, putting f'(t) = 0 will give the minimum value for t. Because $\beta 2 > 0$ Thus,

$$t = -\beta 1/(2 * \beta 2)$$

```
### Part C)

# Computing time with lowest temperature -
# f(t) = \beta0 + \beta1*t + \beta2*t^2
# f'(t) = \beta1 + 2*\beta2*t
# putting f'(t) = 0 will give the minimum value for t. Because \beta2 > 0
```

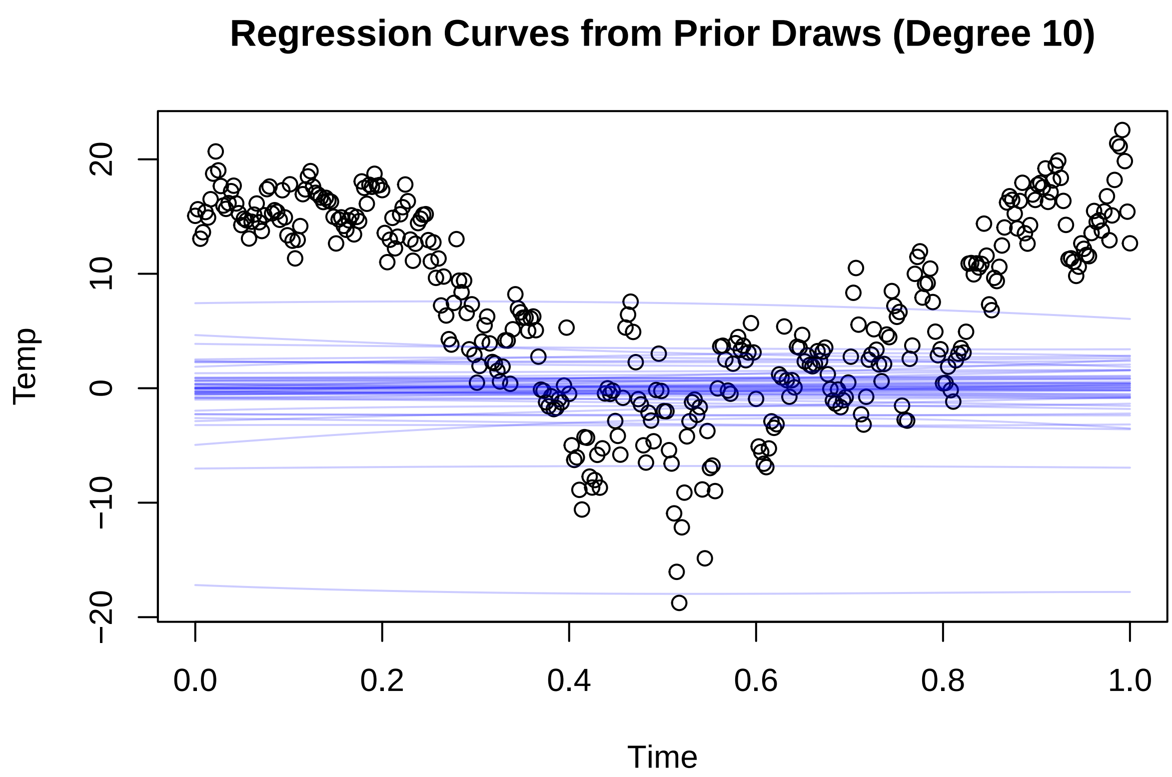
```r
# Thus,    t = - \beta1 / (2*\beta2)
x_min <- -beta_post_draws[,2] / 2* beta_post_draws[,3]

# Plot the posterior distribution
hist(x_min,  breaks = 30, col = "gray",
     main = "Posterior of time with lowest temp",
     xlab = "Time (x̃)" )
```



PART D)

```r
### Part D)

# Build X matrix with polynomial terms up to degree 10
X10 <- sapply(0:10, function(i) data$time^i)
X10 <- as.matrix(X10)

# Prior mean and precision
mu0_10 <- rep(0, 11)
lambda <- 5
omega0_10 <- diag(lambda^(0:10))

# For prediction
# time_seq <- seq(0, 1, length.out = 100)
```

```r
X_plot_10 <- sapply(0:10, function(j) time_seq^j)
X_plot_10 <- as.matrix(X_plot_10)

# Prior predictive draws
# S <- 50
beta_prior_draws_10 <- matrix(0, S, 10 + 1)
# sigma2_prior_draws <- (nu0 * sigma02) / rchisq(S, df = nu0)

# Plot prior predictive regression curves
plot(data$time, data$temp, xlab = "Time", ylab = "Temp",
     main = "Regression Curves from Prior Draws (Degree 10)")

for(s in 1:S){

  cov_beta_10 <- sigma2_prior_draws[s] * solve(omega0_10)
  beta_prior_10 <- rmvnorm(1, mu0_10, cov_beta_10)
  beta_prior_draws_10[s,] <- beta_prior_10

  y_pred_10 <- X_plot_10 %*% t(beta_prior_10)
  lines(time_seq, y_pred_10, col = rgb(0, 0, 1, alpha = 0.2))
}
```



Observation -

The suitable prior mean should be $\mu 0 = 0$ as it expresses no strong belief in any specific shape for the curve. The suitable prior precision could be a diagonal matrix with increasing values for higher degree as higher degree terms would be heavily shrunk towards zero preventing overfitting unless data strongly supports them. Thus, $\mu 0 =$ rep(0, 11) and $\Omega 0 =$ diag(λ^(0:10)) where $\lambda = 5$