Comparing the Bayesian Linear Regression Model to Ordinary Least Squares

Joe Zhang and Christian Paravalos

Winter 2024

Abstract

Linear regression is often used to model relationships between a response variable and one or more explanatory variables. In traditional settings, the ordinary least squares method is used to estimate the model. In this paper, our goal is to develop a linear model by estimating the model parameters under a Bayesian framework and compare the model against the traditional ordinary least squares model using a dataset.

Introduction

To model the relationship between a response variable and one or more explanatory variables, the simplest way to estimate a model is to assume that there is a linear relationship between the response variable and the predictor variables. The formulation of the model is

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where X is an $n \times k$ matrix with the observations of the explanatory variables, $\boldsymbol{y}$ is an $n \times 1$ vector of observations of the response variables, $\boldsymbol{\beta}$ is a $k \times 1$ vector of model parameters, and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of random errors with each observation, n is the number of observations, and k is the number of parameters. The goal is to estimate $\boldsymbol{\beta}$ and then construct and evaluate the model from the estimated values. In other words, we have

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

The most well-known way is to use the ordinary least squares method to estimate the parameters. The goal of this paper is to compare the ordinary least squares method with the Bayesian method of estimating parameters.

Ordinary Least Squares Regression Revisited

Estimation of Model Parameters

Consider a multiple linear regression model

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

assume that the random errors have mean 0, variance σ^2, are uncorrelated, and they follow a normal distribution. In other words, we assume that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$, where $\mathbf{0}$ is the zero vector, and I is the identity matrix.

To estimate the coefficients, the following quantity should be minimized:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} \epsilon_i = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = (\boldsymbol{y} - X\boldsymbol{\beta})^T (\boldsymbol{y} - X\boldsymbol{\beta})$$

We need to take derivatives of $S(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. We get that

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2X^T \boldsymbol{y} + 2X^T X \boldsymbol{\beta}$$

If we set this equation to 0 and we solve for $\boldsymbol{\beta}$, we get

$$X^T X \boldsymbol{\beta} = X^T \boldsymbol{y}$$

If we multiply both sides by the inverse of $X^T X$, assuming that it is invertible, then the least squares estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

The least squares estimator of $\boldsymbol{\beta}$ has the following properties:

- $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$
- $Var[\hat{\boldsymbol{\beta}}] = \sigma^2 (X^T X)^{-1}$

Estimation of Variance

In the frequentist approach, we estimate σ^2 using the residual sum of squares. We have

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2 = \boldsymbol{e}^T \boldsymbol{e}$$

where $\hat{y}_i$ are the fitted values of y_i using the least squares model and $e_i = y_i - \hat{y}_i$ are the residuals. $\boldsymbol{e}$ is the vector of the residuals. If we simplify further, we get

$$SS_{res} = \boldsymbol{y}^T \boldsymbol{y} - \hat{\boldsymbol{\beta}}^T X^T \boldsymbol{y}$$

The residual sum of squares has $n - k$ degrees of freedom. The residual mean square is

$$MS_{res} = \frac{SS_{res}}{n - p}$$

We can show that

$$\mathbb{E}[MS_{res}] = \sigma^2$$

Therefore, an unbiased estimator for σ^2 is given by

$$\hat{\sigma^2} = MS_{res} = \frac{\boldsymbol{y}^T \boldsymbol{y} - \hat{\boldsymbol{\beta}}^T X^T \boldsymbol{y}}{n - k}$$

Inference on model parameters

After obtaining the estimates of the model parameters, we need to make inference on them. The main goal of inference is to determine if all the explanatory variables are significant to the model. Sometimes when there are too many variables, the accuracy of predictions using the model can decrease significantly. Making inferences can help decide if some variables should not be included in the model.

We have assumed that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$. This consequently leads to that $\boldsymbol{y} \sim N(X\boldsymbol{\beta}, \sigma^2 I)$. Under this assumption, we can also get that

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2 (X^T X)^{-1})$$

From this, we can also deduce that

$$\hat{\beta}_j \sim N(\beta_j, \sigma^2 (X^T X)^{-1}_{jj})$$

for $j = 1, 2, ..., k$ and $(X^T X)^{-1}_{jj}$ is the j-th diagonal element of the matrix $(X^T X)^{-1}$.

If we standardize $\hat{\beta}$, we get that

$$\frac{\hat{\beta}_j - \beta_j}{\sigma\sqrt{(X^T X)^{-1}_{jj}}} \sim N(0, 1)$$

In practice, we do not know the value of σ^2, which we will need to use in order to make inference on the model parameters. Therefore, the best way is to use the estimate for σ^2, which is MS_{res}. However, if we replace σ^2 with MS_{res}, we get

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{MS_{res}(X^T X)^{-1}_{jj}}} \sim t(n-k)$$

If we want to test the null hypothesis $H_0 : \beta_j = \beta_{j_0}$ against the hypothesis $H_1 : \beta_j \neq \beta_{j_0}$, we have that under H_0 that

$$t_0 = \frac{\hat{\beta}_j - \beta_{j_0}}{\sqrt{MS_{res}(X^T X)^{-1}_{jj}}} \sim t(n-k)$$

t_0 is the test statistic for the hypothesis $\beta_j = \beta_{j_0}$. We reject H_0 if $t_0 > t_{\alpha/2}(n-k)$, where α is the significance level and $t_{\alpha/2}(n-k)$ is the $100(1-\alpha/2)$th quantile of the Student t distribution with $n-k$ degrees of freedom. We can also compute the p-value, which is

$$p\text{-value} = P(|T| > t_0) = 2P(T > t_0)$$

and we reject H_0 if it is less than α.

Another way to test the hypothesis H_0 against H_1 is to construct confidence intervals for the model parameters β_j. We can deduce that a $100(1-\alpha)\%$ confidence interval for β_j is

$$\hat{\beta}_j \pm t_{\alpha/2}(n-k) \cdot \sqrt{MS_{res}(X^T X)^{-1}_{jj}}$$

From the confidence interval, we can reject H_0 if β_{j_0} is contained in the interval. Otherwise, we accept H_0.

Prediction

The purpose of constructing a model between a response variable and several explanatory variables is to make predictions of future values of the response given certain values of the explanatory variables. To predict $\boldsymbol{y}$ at a given X, we simply use the fitted model

$$\hat{\boldsymbol{y}} = X\hat{\boldsymbol{\beta}}$$

To construct a confidence interval for a prediction, we have that

Bayesian Linear Regression

Now, we turn to the Bayesian approach to estimate a linear regression model between a response variable and several explanatory variables. The form of the model is the same as in the ordinary least squares framework

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

Estimation of Parameters

We make the same assumptions as in the ordinary least squares framework, where $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 I)$. We need to estimate $k+1$ parameters: the k model coefficients, and σ^2.

Likelihood Function

With the assumption, we have that

$$\boldsymbol{y} \sim N_n(X\boldsymbol{\beta}, \sigma^2 I)$$

Therefore, the likelihood function is

$$f(\boldsymbol{y}|X, \boldsymbol{\beta}, \sigma^2) = \frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}} \exp\left[\frac{1}{2\sigma^2}(\boldsymbol{y} - X\boldsymbol{\beta})^T(\boldsymbol{y} - X\boldsymbol{\beta})\right]$$

Choosing a prior distribution

To estimate $\boldsymbol{\beta}$ and σ^2, we treat them as random variables and we make a prior assumption on their distribution. The assumption we will use are

$$\boldsymbol{\beta}|\sigma^2 \sim N_k(\boldsymbol{m}, \sigma^2 V)$$

and

$$\sigma^2 \sim IG(a, b)$$

where $\boldsymbol{m}$ is a $k \times 1$ vector of real numbers, V is a $k \times k$ matrix of real numbers, and a and b are both real numbers.

Then, we get that the prior for $(\boldsymbol{\beta}, \sigma^2)$ is

$$g(\boldsymbol{\beta}, \sigma^2) = g(\boldsymbol{\beta}|\sigma^2)g(\sigma^2)$$

where

$$g(\boldsymbol{\beta}|\sigma^2) = \frac{1}{(2\pi\sigma^2)^{k/2}}|V|^{-\frac{k}{2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \boldsymbol{m})^T V^{-1}(\boldsymbol{\beta} - \boldsymbol{m})\right)$$

and

$$g(\sigma^2) = \frac{b^a}{\Gamma(a)}(\sigma^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma^2}\right)$$

Computation of posterior distribution

Now that we have the likelihood function and the prior probability distribution of the parameters, we can now compute the posterior distribution of the parameters. We let $h(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, X)$ represent the posterior distribution function:

$$
\begin{aligned}
h(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, X) &\propto f(\boldsymbol{y}|X, \boldsymbol{\beta}, \sigma^2) g(\boldsymbol{\beta}, \sigma^2) \\
&\propto f(\boldsymbol{y}|X, \boldsymbol{\beta}, \sigma^2) g(\boldsymbol{\beta}|\sigma^2) g(\sigma^2) \\
&\propto (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - X\boldsymbol{\beta})^T(\boldsymbol{y} - X\boldsymbol{\beta})\right) (2\pi\sigma^2)^{-\frac{k}{2}} |V|^{-\frac{k}{2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \boldsymbol{m})^T V^{-1}(\boldsymbol{\beta} - \boldsymbol{m})\right) \\
&\times \frac{b^a}{\Gamma(a)} (\sigma^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma^2}\right) \\
&\propto (\sigma^2)^{-\frac{n}{2} - \frac{k}{2} - (a+1)} \exp\left(-\frac{1}{2\sigma^2}[(\boldsymbol{y} - X\boldsymbol{\beta})^T(\boldsymbol{y} - X\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{m})^T V^{-1}(\boldsymbol{\beta} - \boldsymbol{m}) + 2b]\right) \\
&\propto (\sigma^2)^{-\frac{n}{2} - \frac{k}{2} - (a+1)} \exp\left(-\frac{1}{2\sigma^2} A\right)
\end{aligned}
$$

We now want to rewrite the equation above by simplifying the quantity inside the exponential function. We do the following:

$$
\begin{aligned}
A &= (\boldsymbol{y} - X\boldsymbol{\beta})^T(\boldsymbol{y} - X\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{m})^T V^{-1}(\boldsymbol{y} - \boldsymbol{m}) + 2b \\
&= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T X\boldsymbol{\beta} - \boldsymbol{\beta}^T X^T\boldsymbol{y} + \boldsymbol{\beta}^T X^T X\boldsymbol{\beta} + \boldsymbol{\beta}^T V^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T V^{-1}\boldsymbol{m} - \boldsymbol{m}^T V^{-1}\boldsymbol{\beta} + \boldsymbol{m}^T V^{-1}\boldsymbol{m} + 2b \\
&= \boldsymbol{\beta}^T(X^T X + V^{-1})\boldsymbol{\beta} - \boldsymbol{\beta}^T(X^T\boldsymbol{y} + V^{-1}\boldsymbol{m}) + (\boldsymbol{m}^T V^{-1}\boldsymbol{m} + 2b + \boldsymbol{y}^T\boldsymbol{y}) - (\boldsymbol{y}^T X + \boldsymbol{m}^T V^{-1})\boldsymbol{\beta}
\end{aligned}
$$

We define the following quantities:

$$\Lambda = (X^T X + V^{-1})^{-1}$$

and

$$\boldsymbol{\mu} = (X^T X + V^{-1})^{-1}(X^T\boldsymbol{y} + V^{-1}\boldsymbol{m})$$

where Λ is a $k \times k$ matrix and $\boldsymbol{\mu}$ is a $k \times 1$ vector.

Then we can write

$$
\begin{aligned}
A &= \boldsymbol{\beta}^T\Lambda^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T\Lambda^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\beta} + \boldsymbol{m}^T V^{-1}\boldsymbol{m} + 2b + \boldsymbol{y}^T\boldsymbol{y} \\
&= (\boldsymbol{\beta} - \boldsymbol{\mu})^T\Lambda^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}) - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\mu} + \boldsymbol{m}^T V^{-1}\boldsymbol{m} + 2b + \boldsymbol{y}^T\boldsymbol{y}
\end{aligned}
$$

Now, we go back to the posterior distribution and we can write:

$$
\begin{aligned}
h(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, X) &\propto (\sigma^2)^{-\frac{n}{2} - \frac{k}{2} - (a+1)} \exp\left(-\frac{1}{2\sigma^2} A\right) \\
&\propto (\sigma^2)^{-\frac{n}{2} - \frac{k}{2} - (a+1)} \exp\left(-\frac{1}{2\sigma^2}((\boldsymbol{\beta} - \boldsymbol{\mu})^T\Lambda^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}) - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\mu} + \boldsymbol{m}^T V^{-1}\boldsymbol{m} + 2b + \boldsymbol{y}^T\boldsymbol{y})\right) \\
&\propto (\sigma^2)^{-\frac{k}{2}} \exp\left(\frac{(\boldsymbol{\beta} - \boldsymbol{\mu})^T\Lambda^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu})}{2\sigma^2}\right) (\sigma^2)^{-(\frac{n}{2} + a + 1)} \exp\left(\frac{\boldsymbol{m}^T V^{-1}\boldsymbol{m} - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\mu} + 2b + \boldsymbol{y}^T\boldsymbol{y}}{2\sigma^2}\right)
\end{aligned}
$$

From looking at the posterior distribution, we can see that it is the product of a multivariate normal distribution and an inverse gamma distribution. We can parameterize the posterior the following way:

$$h(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, X) = h(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}, X) h(\sigma^2|\boldsymbol{y}, X)$$

where

$$h(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}, X) = \frac{1}{(2\pi\sigma^2)^{-\frac{k}{2}}} \exp\left(\frac{(\boldsymbol{\beta}-\boldsymbol{\mu})^T\Lambda^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})}{2\sigma^2}\right)$$

and

$$h(\sigma^2|\boldsymbol{y}, X) = \frac{(\frac{1}{2}(\boldsymbol{m}^T V^{-1}\boldsymbol{m} - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\mu} + 2b + \boldsymbol{y}^T\boldsymbol{y}))^{\frac{n}{2}+a}}{\Gamma(a+\frac{n}{2})}(\sigma^2)^{-(\frac{n}{2}+a+1)} \exp\left(\frac{\boldsymbol{m}^T V^{-1}\boldsymbol{m} - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\mu} + 2b + \boldsymbol{y}^T\boldsymbol{y}}{2\sigma^2}\right)$$

Then we get that

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}, X \sim N_k(\boldsymbol{\mu}, \sigma^2\Lambda)$$

and

$$\sigma^2|\boldsymbol{y}, X \sim IG(\alpha, r)$$

where

$$\boldsymbol{\mu} = (X^T X + V^{-1})^{-1}(X^T\boldsymbol{y} + V^{-1}\boldsymbol{m})$$
$$\Lambda = (X^T X + V^{-1})^{-1}$$
$$\alpha = a + \frac{n}{2}$$
$$r = \frac{\boldsymbol{m}^T V^{-1}\boldsymbol{m} - \boldsymbol{\mu}^T\Lambda^{-1}\boldsymbol{\mu} + 2b + \boldsymbol{y}^T\boldsymbol{y}}{2}$$

Making Bayesian inference

We now have the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$. From that, we also get that

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}, X \sim N_k(\boldsymbol{\mu}, \sigma^2\Lambda^{-1})$$

and

$$\sigma^2|\boldsymbol{y}, X \sim IG(\alpha, r)$$

The main goal is to estimate the parameters. We first need to find the marginal distribution of $\boldsymbol{\beta}$. This is hard to do by hand since we have to solve the following integral:

$$h(\boldsymbol{\beta}|\boldsymbol{y}, X) = \int_{-\infty}^{\infty} h(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}, X)h(\sigma^2|\boldsymbol{y}, X)\, d(\sigma^2)$$

Therefore, we need to use computational methods to simulate the posterior marginal distribution for $\boldsymbol{\beta}$.

To make inference on individual parameters, we have that since $\boldsymbol{\beta}$ follows a multivariate normal distribution, we can also conclude that for any model parameter β_j, we get that

$$\beta_j|\sigma^2, \boldsymbol{y}, X \sim N(\mu_j, \sigma^2\Lambda_{jj})$$

where μ_j is the j-th element of the vector $\boldsymbol{\mu}$ and Λ_{jj} is the j-th diagonal element of the matrix Λ. Then the marginal distribution for β_j is

$$h(\beta_j|\boldsymbol{y}, X) = \int_{-\infty}^{\infty} h(\beta_j|\sigma^2, \boldsymbol{y}, X)h(\sigma^2|\boldsymbol{y}, X)\, d(\sigma^2)$$

This integral is also hard to compute, so we need to use computational methods.

Given the marginal distribution, we can use it to find the mean and variance of β_j and we can also construct credible regions to decide if the explanatory variable associated with the parameter is significant to the model. If the credible region contain 0, then we can conclude that the parameter and its associated explanatory variable is not significant to the model. Otherwise, we conclude that it is significant.

Simulation Algorithm to Compute Posterior

To simulate the marginal distribution for a parameter β_j, we can use the following algorithm:

1. Load the data.
2. Initialize the chosen parameters $\boldsymbol{m}$, V, a, and b.
3. Compute $\boldsymbol{\mu}$, Λ, α, and r using the data and the initialized prior parameters.
4. Simulate σ^2 a large number of times from the $IG(\alpha, r)$ distribution.
5. For each simulated value of σ^2, simulate β_j from the $N(\mu_j, \sigma^2\Lambda_{jj})$ distribution.
6. Draw a graph for the simulated distribution of β_j, then compute its center and a 95% credible region.

Application

Now we have established the theory behind Bayesian linear regression, we now compare the Bayesian linear regression model to the ordinary least squares model on a data set.

Data Description and Preprocessing

Ordinary Least Squares Model

Taken from my assignment from last year for regression for testing:

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(MPV)
```

```
## Loading required package: lattice

## Loading required package: KernSmooth

## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009

## Loading required package: randomForest

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'MPV'
```

```
## The following object is masked from 'package:olsrr':
##
##     cement
```

```
thermalData <- table.b2

model <- lm(y ~ ., data=table.b2)

print(model)
```

```
##
## Call:
## lm(formula = y ~ ., data = table.b2)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##   325.43612      0.06753      2.55198      3.80019    -22.94947      2.41748
```

```
ols_step_forward_p(model, details = FALSE)
```

```
##
##
##                               Stepwise Summary
## -------------------------------------------------------------------------
## Step    Variable        AIC        SBC       SBIC        R2       Adj. R2
## -------------------------------------------------------------------------
##  0      Base Model    266.884    269.618    181.319    0.00000    0.00000
##  1      x4            231.913    236.015    146.852    0.72052    0.71017
##  2      x3            214.131    219.600    131.235    0.85872    0.84785
##  3      x2            212.782    219.618    130.596    0.87413    0.85902
##  4      x1            210.636    218.840    129.980    0.89089    0.87271
##  5      x5            210.466    220.037    131.162    0.89876    0.87675
## -------------------------------------------------------------------------
##
## Final Model Output
## ------------------
##
##                          Model Summary
## ---------------------------------------------------------------
## R                       0.948       RMSE                 7.159
## R-Squared               0.899       MSE                 64.626
## Adj. R-Squared          0.877       Coef. Var            3.220
## Pred R-Squared          0.788       AIC                210.466
## MAE                     5.440       SBC                220.037
## ---------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##  AIC: Akaike Information Criteria
```

```
##  SBC: Schwarz Bayesian Criteria
##
##                                ANOVA
## --------------------------------------------------------------------
##                 Sum of
##                Squares        DF    Mean Square      F         Sig.
## --------------------------------------------------------------------
## Regression    13195.533         5       2639.107    40.837    0.0000
## Residual       1486.395        23         64.626
## Total         14681.928        28
## --------------------------------------------------------------------
##
##                                  Parameter Estimates
## ----------------------------------------------------------------------------------------------------
##       model       Beta    Std. Error    Std. Beta      t        Sig       lower      upper
## ----------------------------------------------------------------------------------------------------
## (Intercept)    325.436        96.127                 3.385    0.003    126.582    524.290
##          x4    -22.949         2.704       -0.910    -8.488    0.000    -28.542    -17.357
##          x3      3.800         1.461        0.308     2.601    0.016      0.778      6.823
##          x2      2.552         1.248        0.158     2.044    0.053     -0.030      5.134
##          x1      0.068         0.029        0.235     2.329    0.029      0.008      0.128
##          x5      2.417         1.808        0.206     1.337    0.194     -1.323      6.158
## ----------------------------------------------------------------------------------------------------
```

```r
K <- ols_step_all_possible(model, details=TRUE)

print(K)
```

```
##    Index N     Predictors   R-Square Adj. R-Square Mallow's Cp
## 4      1 1             x4 0.72052420    0.71017324  0.66925799
## 1      2 1             x1 0.39393873    0.37149202  0.26286550
## 5      3 1             x5 0.12328631    0.09081543 -0.07430124
## 3      4 1             x3 0.01256447   -0.02400721 -0.10822289
## 2      5 1             x2 0.01047594   -0.02617310 -0.10035893
## 13     6 2          x3 x4 0.85871542    0.84784738  0.81361382
## 15     7 2          x4 x5 0.82020641    0.80637613  0.74210380
## 8      8 2          x1 x4 0.73386067    0.71338841  0.66912580
## 11     9 2          x2 x4 0.72053206    0.69903453  0.63546717
## 6     10 2          x1 x2 0.44922541    0.40685814  0.27233435
## 7     11 2          x1 x3 0.42636430    0.38223847  0.26831058
## 9     12 2          x1 x5 0.39429011    0.34769704  0.21899101
## 14    13 2          x3 x5 0.37034618    0.32191127  0.09207740
## 12    14 2          x2 x5 0.12963616    0.06268509 -0.12174423
## 10    15 2          x2 x3 0.03427915   -0.04000706 -0.11279275
## 22    16 3       x2 x3 x4 0.87412678    0.85902200  0.78955967
## 19    17 3       x1 x3 x4 0.86478901    0.84856370  0.81433580
## 25    18 3       x3 x4 x5 0.86376234    0.84741382  0.79356123
## 21    19 3       x1 x4 x5 0.86288638    0.84643274  0.80227864
## 24    20 3       x2 x4 x5 0.82022133    0.79864789  0.71159676
## 17    21 3       x1 x2 x4 0.73615340    0.70449181  0.63353182
## 16    22 3       x1 x2 x3 0.52969885    0.47326272  0.34802937
## 20    23 3       x1 x3 x5 0.46570209    0.40158634  0.26447373
## 23    24 3       x2 x3 x5 0.46085816    0.39616114  0.12889834
## 18    25 3       x1 x2 x5 0.45487844    0.38946385  0.23821700
```

```
## 26    26 4    x1 x2 x3 x4 0.89089317    0.87270870  0.80607673
## 29    27 4    x1 x3 x4 x5 0.88036169    0.86042197  0.79255245
## 30    28 4    x2 x3 x4 x5 0.87488338    0.85403061  0.76679797
## 28    29 4    x1 x2 x4 x5 0.86898542    0.84714966  0.78472045
## 27    30 4    x1 x2 x3 x5 0.58159740    0.51186363  0.34750644
## 31    31 5 x1 x2 x3 x4 x5 0.89876023    0.87675159  0.78817863
```

When looking at the statistics for the different models, we are only allowed to currently look at a few statistics, but in part e, I'll look into more statistics. For the models with one to two predictors, I'm not going to go into much detail about them as their R^2, R^2_{adj} are significantly lower than the models with a higher number of predictors (and consequently this means their MSE is much higher). Additionally, all of their Mallow's CP values are astronomically higher than the value they should be around (value they should be around is p where p represents the number of predictors in the model). Moving on to the 3 variable models, it seems that the model with the predictors X_2, X_3, and X_4 has the highest R^2 value of around 87.4% and a R^2_{adj} value of 85.9%. It seems that for this model the Mallow's CP value has become much closer to the desired value (it is around 7.596), which for the case of a model with 3 predictors would be 3. Fortunately, for the 4 predictor models, the best model based on R^2 has an R^2 value of around 89.1% and an R^2_{adj} value of around 87.3% (MSE of 1945.334), with the predictors X_1, X_2, X_3, and X_4 in the model. Now, the Mallow's CP value is pretty close to its target value of p=4 for this model, but still is around 1.79 higher than 4. In comparison to the full model with all the predictors present, the full model has a Mallow's CP value closer to the desired value of p=5, coming in at exactly 1 above the desired value. In terms of R^2 and R^2_{adj}, it seems that these values are barely higher than the 4 predictor model, indicating that adding X_5 to the model doesn't add much predictive power (MSE of 1887.116 and R^2_{adj} value of around 87.7%). Even though the Mallow's CP value is closer to its desired value for the full model with 5 predictors, I think the benefit of having one less variable with approximately the same R^2_{adj} value outweighs the disadvantage. So, if favoring simplicity, I would choose the 4 predictor model with predictors X_1, X_2, X_3, and X_4, but if wanting the best absolute R^2_{adj} value and closest Mallow's CP value to the line, then the full model with 5 predictors would be a better choice. I have to further analyze the significance of adding X_5 to the best 4 predictor model, such as using the partial F-test to determine whether it should be added. I can also see the values for AIC and SBC even though technically we aren't allowed to look at them yet, and they further indicate that the best 4 predictor model without X_5 as a predictor is as good if not better than the full model. But using just this information of R^2, R^2_{adj}, and Mallow's CP, at first glance it seems the five predictor model is better (but we will show it is probably not).

```r
full_model=lm(y~x1+x2+x3+x4+x5,data=thermalData)

anova(full_model)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 5783.8  5783.8 89.4964 2.155e-09 ***
## x2         1  811.7   811.7 12.5602 0.0017316 **
## x3         1 1181.5  1181.5 18.2822 0.0002832 ***
## x4         1 5303.0  5303.0 82.0574 4.772e-09 ***
## x5         1  115.5   115.5  1.7873 0.1943335
## Residuals 23 1486.4    64.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(full_model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = thermalData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6848  -2.7688   0.6273   3.9166  17.3962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.43612   96.12721   3.385  0.00255 **
## x1            0.06753    0.02899   2.329  0.02900 *
## x2            2.55198    1.24824   2.044  0.05252 .
## x3            3.80019    1.46114   2.601  0.01598 *
## x4          -22.94947    2.70360  -8.488 1.53e-08 ***
## x5            2.41748    1.80829   1.337  0.19433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.039 on 23 degrees of freedom
## Multiple R-squared:  0.8988, Adjusted R-squared:  0.8768
## F-statistic: 40.84 on 5 and 23 DF,  p-value: 1.077e-10
```

```r
cat("MSE: ",  mean((thermalData$y - predict(full_model, newdata=thermalData))^2))
```

```
## MSE:  51.255
```

First off, we can see that the test for significance of regression would allow us to reject the null hypothesis that all predictors in the model don't have a linear relationship in the current model with the response variable, indicating significance of this model in explaining/predicting the response Y. We can see that for the t-tests that each coefficient of the predictor is 0 and that the predictor doesn't add significantly to the current model, we can note that X_1, X_3, and X_4 have small p-values for this test where we would reject the null hypothesises at a significance level of 5% that their coefficients are equal to 0 and that they don't significantly improve the model given the other predictors present in the model. Now, X_2's t-test's p-value would not allow us to reject the null hypothesis that it does not significantly improve the model, but I still believe that it has some statistical power, but it is toeing the line. On the other hand, X_5 has a very large p-value of 19.4% for the t-test, indicating that it does not improve the model significantly. As forward and backwards stepwise regression both use a p-value of 30% to add or remove a variable from the model using the partial F test/t-test for the predictor, they tend to add too many variables, at least in this case. Now, the models that I chose for part c. and d. were the 4 predictor model with the predictors X_1, X_2, X_3, and X_4. We will look at the summary of the model below:

```r
reduced_model = lm(y~x1+x2+x3+x4,data=thermalData)

anova(reduced_model)
```

```
## Analysis of Variance Table
##
## Response: y
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 5783.8  5783.8  86.654 1.954e-09 ***
## x2         1  811.7   811.7  12.161 0.0019011 **
## x3         1 1181.5  1181.5  17.702 0.0003117 ***
## x4         1 5303.0  5303.0  79.451 4.420e-09 ***
## Residuals 24 1601.9    66.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(reduced_model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = thermalData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.322  -2.639   0.025   4.786  16.003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 270.21013   88.21060   3.063  0.00534 **
## x1            0.05156    0.02685   1.920  0.06676 .
## x2            2.95141    1.23167   2.396  0.02471 *
## x3            5.33861    0.91506   5.834 5.13e-06 ***
## x4          -21.11940    2.36936  -8.914 4.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.17 on 24 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8727
## F-statistic: 48.99 on 4 and 24 DF,  p-value: 3.327e-11
```

```
cat("MSE: ",  mean((thermalData$y - predict(reduced_model, newdata=thermalData))^2))
```

```
## MSE:  55.23788
```

Strangely enough, in this 4 predictor model, we get that X_1 does not significantly improve the model given the other predictors in the model (using a significance level of 5% based on the t-tests), while the other predictors do. We can note that the model with 1 predictor with X_1 alone had a very large R^2 value, so this predictor alone did have a lot of predictive power on Y. I won't go over again my discussion about the other statistics such as Mallow's CP as I did in part d., but we concluded then that the model with 4 predictors is probably better. I still stick with this statement as when looking at the t-test for the significance of X_5 in the full model, we pretty resoundingly don't reject the null hypothesis that it has no predictive/explanative power on Y in the model given the other predictors already in it.

Using the information given here, I would choose the best model as the one with 4 predictors. For the 4 predictor models, the best model based on R^2 has an R^2 value of around 89.1% and an R^2_{adj} value of around 87.3%, with the predictors X_1, X_2, X_3, and X_4 in the model. Now, the Mallow's CP value is pretty close to its target value of p=4 for this model, but still is around 1.79 higher than 4. In comparison to the full model with all the predictors present, the full model has a Mallow's CP value closer to the desiired value of p=5, coming in at exactly 1 above the desired value. In terms of R^2 and R^2_{adj}, it seems that these values are barely higher than the 4 predictor model, indicating that adding X_5 to the model doesn't add much

predictive power. Even though the Mallow's CP value is closer to its desired value for the full model with 5 predictors, I think the benefit of having one less variable with approximately the same R^2_{adj} value outweighs the disadvantage. Looking at the AIC and SBC values, which are somewhat similar metrics, the 4 predictor model has a slightly higher AIC value than the 5 predictor model, but a lower SBIC value. This indicates it is probably the better model to pick. So I would pick this 4 predictor model.

Note that the 4 and 5 predictor models both have issues with the residuals that estimate the error terms indicating that normality is violated in both models.

Bayesian Linear Regression Model

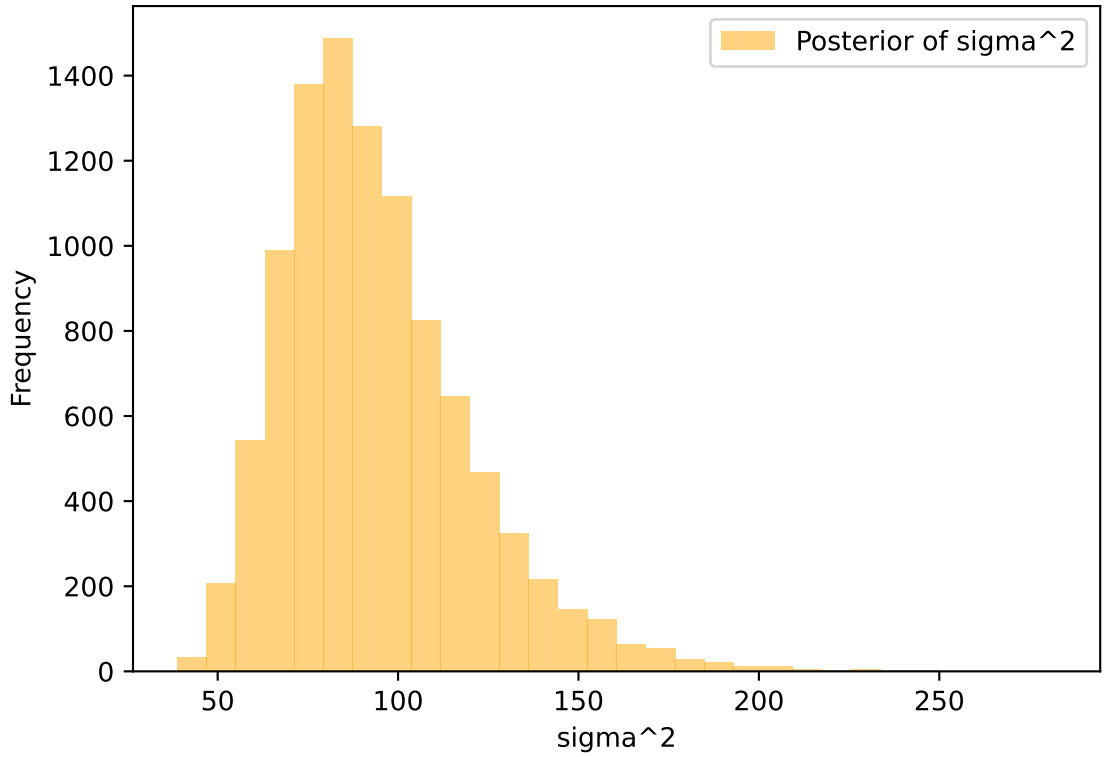
Messy Code:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

#Load the data from a text file mak ethe fist row the header and use tab as the separator
data = pd.read_csv('~/Downloads/thermalData.csv', header=0, sep=',')

#Delete rows with missing values
data = data.dropna()

data.insert(1, 'intercept_term', 1)

num_predictors = len(data.columns) - 1

#Set sampling size
sampling_size = 10000

#Set the parameters for the prior distribution for sigma2 (inverse gamma) (non-informative)
a = 0.001
b = 0.001

#Set the parameters for the prior distribution for mu (normal) (non-informative)

#Create the matrix for V
V = np.eye(num_predictors)
m = np.zeros(num_predictors)

X = data.iloc[:, 1:7].to_numpy()
y = data.iloc[:, 0].to_numpy()

#Compute the inverse of V
V_inv = np.linalg.inv(V)

#Compute the mean of the posterior distribution of beta given sigma2 (normal)
mu_post = np.linalg.inv(X.T @ X + V_inv) @ (X.T @ y + V_inv @ m)

#Compute the lambda matrix of the posterior distribution of beta given sigma2 (normal)
lambda_post = np.linalg.inv(X.T @ X + V_inv)

#Compute alpha paramater of sigma2 posterior distribution (inverse gamma)
```

```
alpha_post = a + len(y) / 2
r_post = (m.T @ V_inv @ m - mu_post.T @ np.linalg.inv(lambda_post) @ mu_post + 2*b + y.T @ y) / 2

print(mu_post)
```

```
## [  2.06419054   0.10979636   5.66324308   6.3922216  -15.40891609
##   -0.29083192]
```

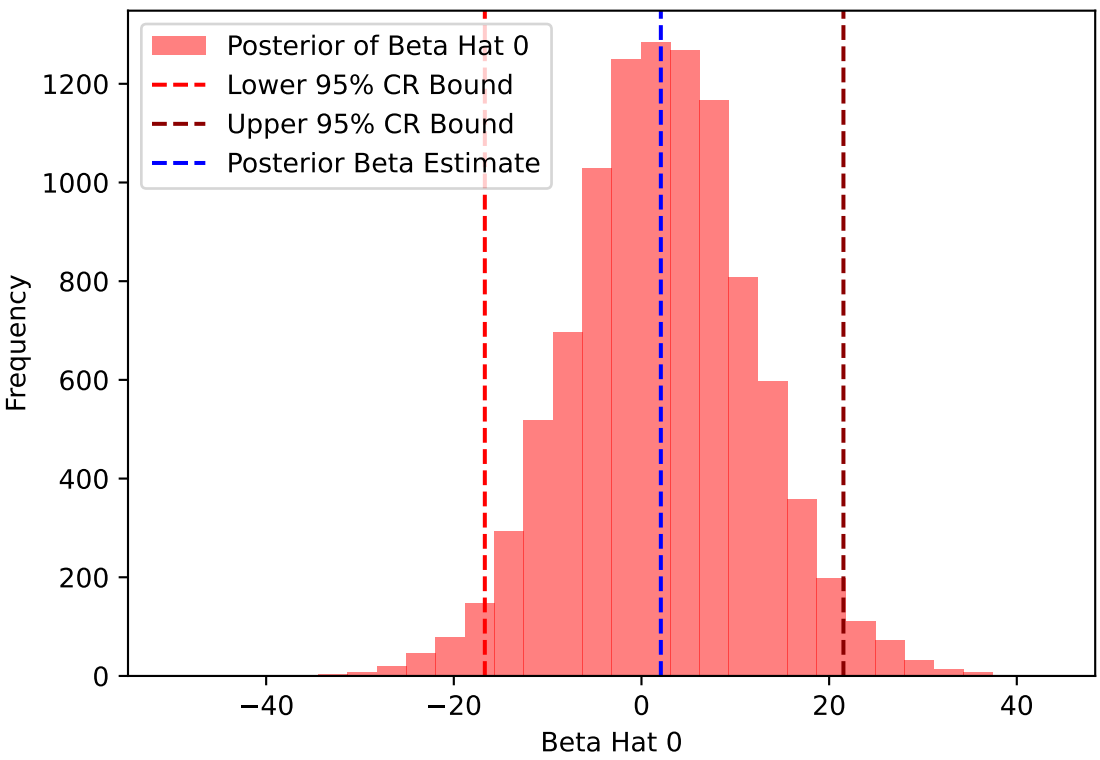
```
# Step 4: Drawing sigma2 from the inverse gamma distribution
sigma2 = 1 / np.random.gamma(shape=alpha_post, scale=1/r_post, size=sampling_size)

#Create the vector for beta_hat
beta_hat = np.zeros((sampling_size, num_predictors))

for i in range(sampling_size):
    beta_hat[i, :] = np.random.multivariate_normal(mu_post, lambda_post * sigma2[i])

plt.hist(sigma2, bins=30, alpha=0.5, color='orange', label='Posterior of sigma^2')
plt.xlabel('sigma^2')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

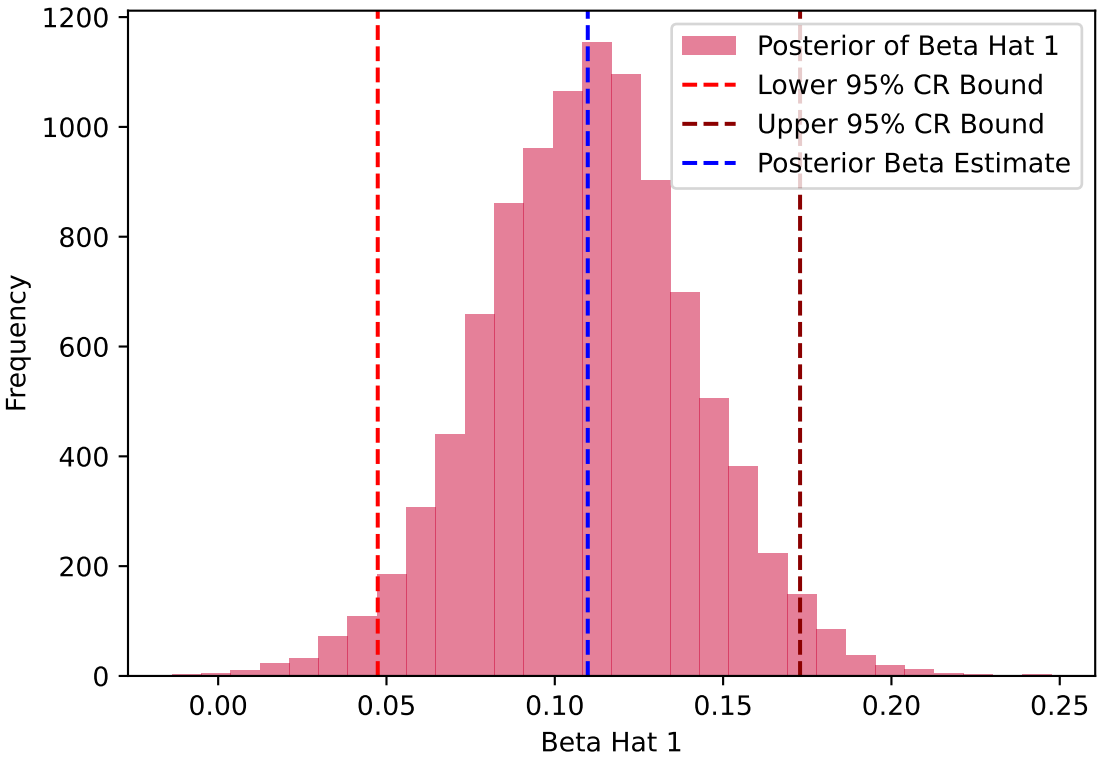


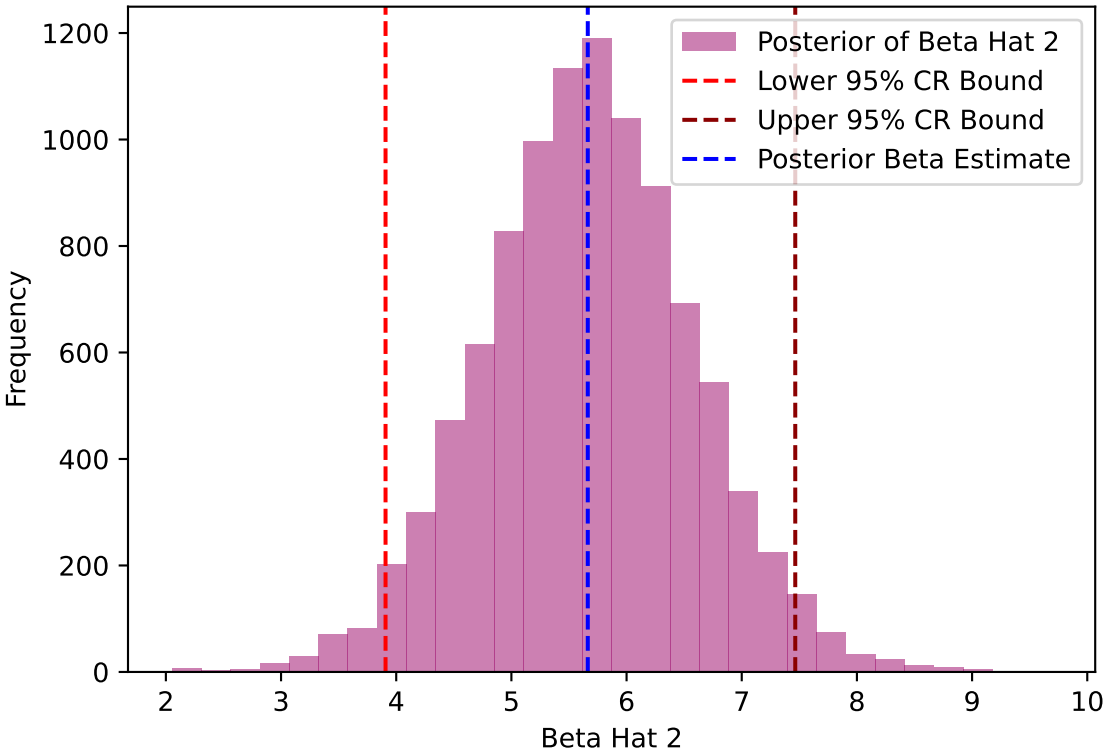
```
colors = LinearSegmentedColormap.from_list("gradient", ["red", "purple", "blue"], N=num_predictors)

for i in range(num_predictors):

    sorted_beta_hat_i = np.sort(beta_hat[:, i])

    value_at_975 = sorted_beta_hat_i[np.floor(0.975 * sampling_size).astype(int)]
    value_at_25 = sorted_beta_hat_i[np.floor(0.025 * sampling_size).astype(int)]

    print(f"Best Estimate for Beta (Beta Hat): {mu_post[i]}")
    print(f"Best Estimate for Standard Deviation of Beta {i}: {lambda_post[i][i] * sigma2[i]}")
    #Print the 95% credible region for beta_hat_i
    print(f"95% credible region for Beta Hat {i}: ({value_at_25}, {value_at_975})")
```

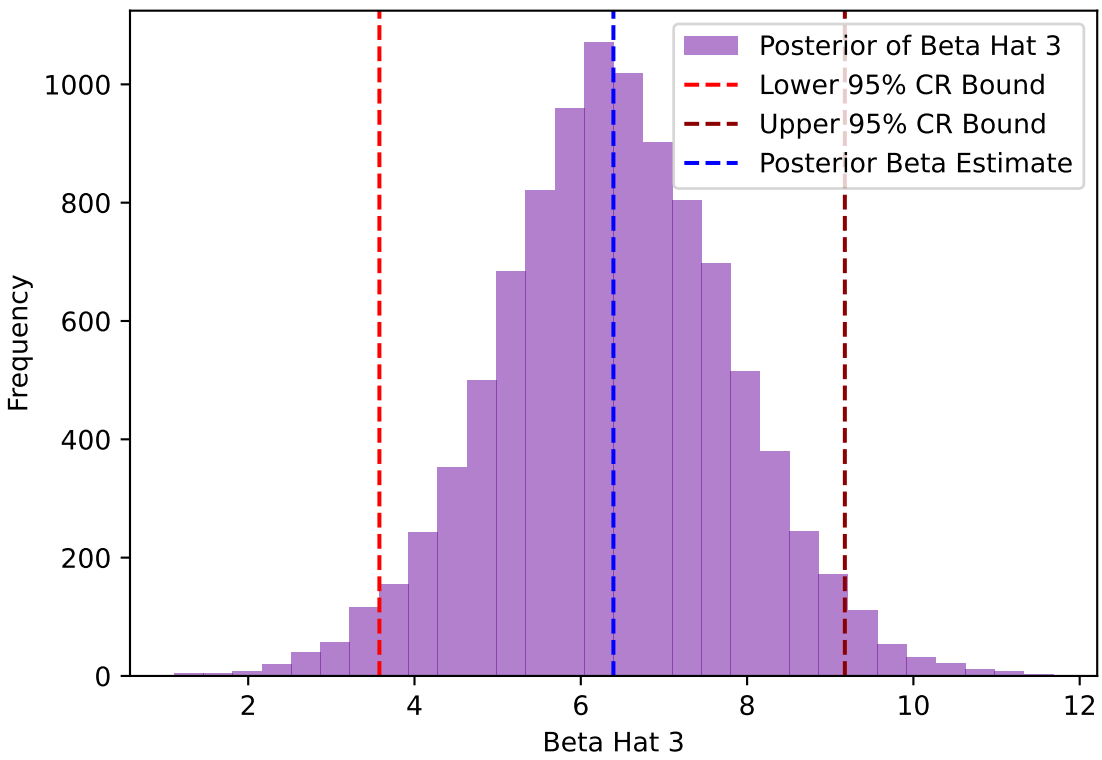
```
## Best Estimate for Beta (Beta Hat): 2.064190543323007
## Best Estimate for Standard Deviation of Beta 0: 80.45280830556601
## 95% credible region for Beta Hat 0: (-16.69476051851737, 21.5241634416613)
## Best Estimate for Beta (Beta Hat): 0.1097963618945142
## Best Estimate for Standard Deviation of Beta 1: 0.001207103684305319
## 95% credible region for Beta Hat 1: (0.04740549643590999, 0.1728630283396017)
## Best Estimate for Beta (Beta Hat): 5.663243080882301
## Best Estimate for Standard Deviation of Beta 2: 0.6311442702119965
## 95% credible region for Beta Hat 2: (3.9072691506799373, 7.464654454595114)
## Best Estimate for Beta (Beta Hat): 6.392221597458726
## Best Estimate for Standard Deviation of Beta 3: 3.2778467975248113
## 95% credible region for Beta Hat 3: (3.5778054635062517, 9.175705359558346)
## Best Estimate for Beta (Beta Hat): -15.408916091352864
## Best Estimate for Standard Deviation of Beta 4: 4.10874249820503
## 95% credible region for Beta Hat 4: (-19.71484488804849, -11.077661395685443)
## Best Estimate for Beta (Beta Hat): -0.2908319162523272
## Best Estimate for Standard Deviation of Beta 5: 3.6189533443084727
## 95% credible region for Beta Hat 5: (-4.01995276777042, 3.480988723856836)
```

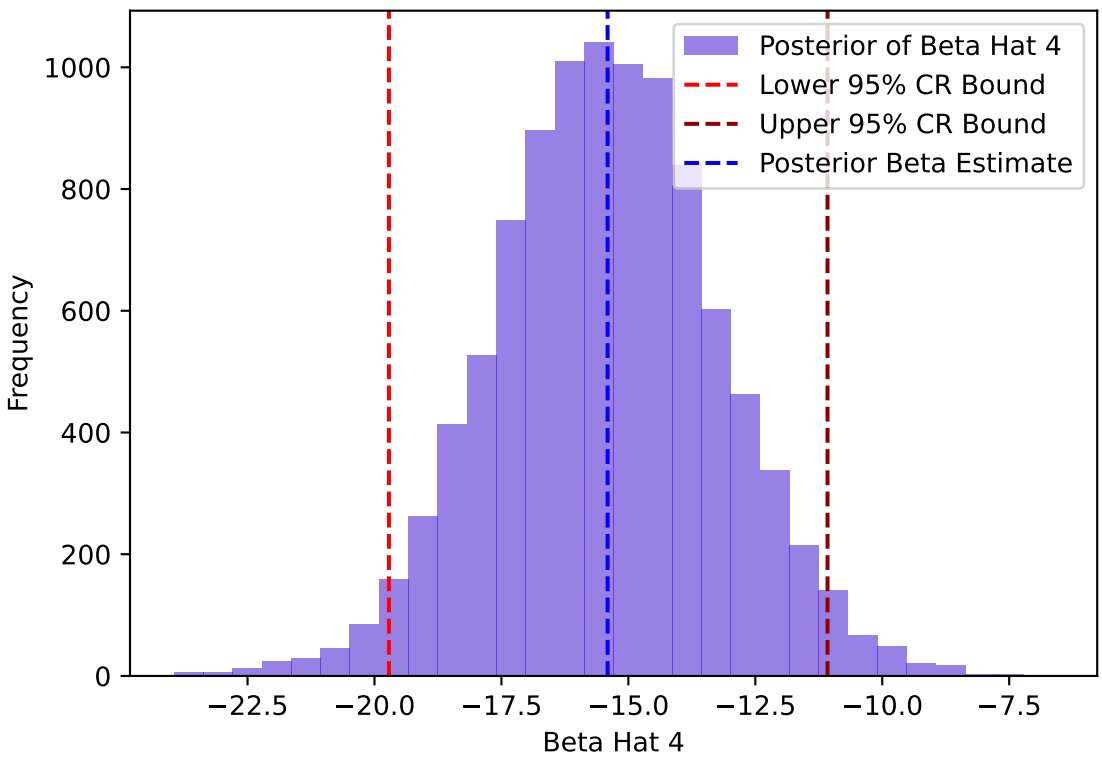
```
for i in range(num_predictors):

    sorted_beta_hat_i = np.sort(beta_hat[:, i])

    value_at_975 = sorted_beta_hat_i[np.floor(0.975 * sampling_size).astype(int)]
    value_at_25 = sorted_beta_hat_i[np.floor(0.025 * sampling_size).astype(int)]

    #Create histograms for beta_hat_i
    plt.hist(beta_hat[:, i], bins=30, color=colors(i), alpha=0.5, label=f'Posterior of Beta Hat {i}')
    plt.axvline(x=value_at_25, color='r', linestyle='--', label='Lower 95% CR Bound')
    plt.axvline(x=value_at_975, color='darkred', linestyle='--', label='Upper 95% CR Bound')
    plt.axvline(x=mu_post[i], color='b', linestyle='--', label='Posterior Beta Estimate')

    plt.xlabel(f'Beta Hat {i}')
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

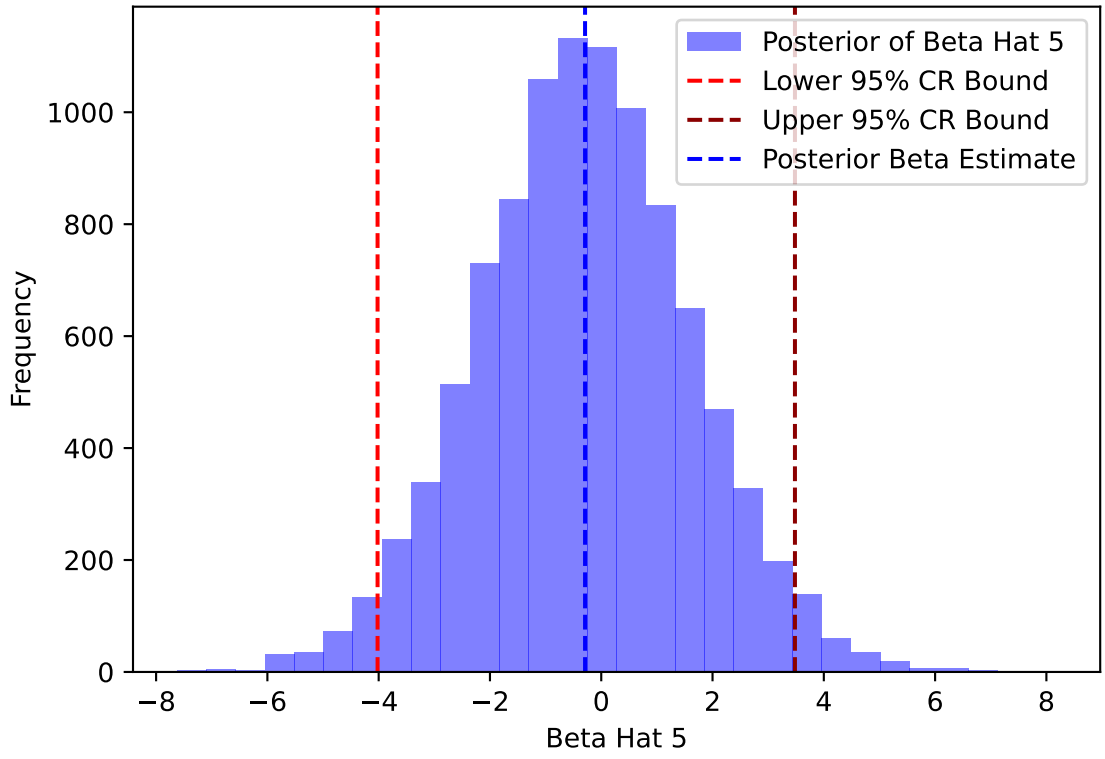

Posterior of Beta Hat 0
Lower 95% CR Bound
Upper 95% CR Bound
Posterior Beta Estimate
Frequency
0
200
400
600
800
1000
1200
−40
−20
0
20
40
Beta Hat 0


Posterior of Beta Hat 1
Lower 95% CR Bound
Upper 95% CR Bound
Posterior Beta Estimate
Frequency
Beta Hat 1
0
200
400
600
800
1000
1200
0.00
0.05
0.10
0.15
0.20
0.25


Posterior of Beta Hat 2
Lower 95% CR Bound
Upper 95% CR Bound
Posterior Beta Estimate
Frequency
Beta Hat 2
0
200
400
600
800
1000
1200
2
3
4
5
6
7
8
9
10


Posterior of Beta Hat 3
Lower 95% CR Bound
Upper 95% CR Bound
Posterior Beta Estimate
Frequency
0
200
400
600
800
1000
2
4
6
8
10
12
Beta Hat 3


Posterior of Beta Hat 4
Lower 95% CR Bound
Upper 95% CR Bound
Posterior Beta Estimate
Frequency
1000
800
600
400
200
0
−22.5
−20.0
−17.5
−15.0
−12.5
−10.0
−7.5
Beta Hat 4




```
#Compute MSE
MSE = np.mean((y - X @ mu_post)**2)

print(f"Bayesian MSE: {MSE}")
```

```
## Bayesian MSE: 77.07265777059224
```

Comparison Between Models

Conclusion

References

Note: These are not cited in proper format yet.

https://en.wikipedia.org/wiki/Bayesian_linear_regression

https://gregorygundersen.com/blog/2020/02/04/bayesian-linear-regression/

https://www.researchgate.net/publication/333917874_Bayesian_Linear_Regression#pf18