

Exercise 8 - Introduction to Bayesian Inference

Konstantinos Vakalopoulos 12223236

2023-12-13

Contents

Task 1	2
Task 1.1	2
Task 1.2	4
Task 1.3	5
Task 1.4	7
Task 2	7
Task 2.1	8
Task 2.2	12
Task 2.3	13
Task 2.4	14

Task 1

We recalculate the estimation of the prevalence of Covid19 in spring 2020. Samples from 1279 persons were analysed with PCR testing procedures. Out of all those not a single randomly selected person was tested positively. This obviously breaks standard testing mechanisms for estimating the proportion of infected person in Austria.

However, additional information is available from similar tests in Germany which had a comparable behaviour of the spread of the disease at that time. In the same time span 4 positive cases out of 4068 had been found.

Task 1.1

Build a Beta prior distribution for this Binomial scenario, which encodes the information of the German study. **Reweight both parameters compared to the original counts with a factor of $\frac{1}{10}$.**

According to slide 41 of the lecture notes, the y prior and the n are assigned with the positive cases in Germany and the total tests in Germany, respectively. Therefore, the alpha and beta prior are y+1 and n-y+1, respectively. Afterwards, a sequence is created and finally the beta prior distribution for the Binomial scenario is created. It is worth mentioning that the conjugate prior of a binomial distribution is a beta distribution.

```
total_tests_Germany <- 4068
positive_cases_Germany <- 4

y_prior <- positive_cases_Germany # slide 41
n <- total_tests_Germany # slide 41

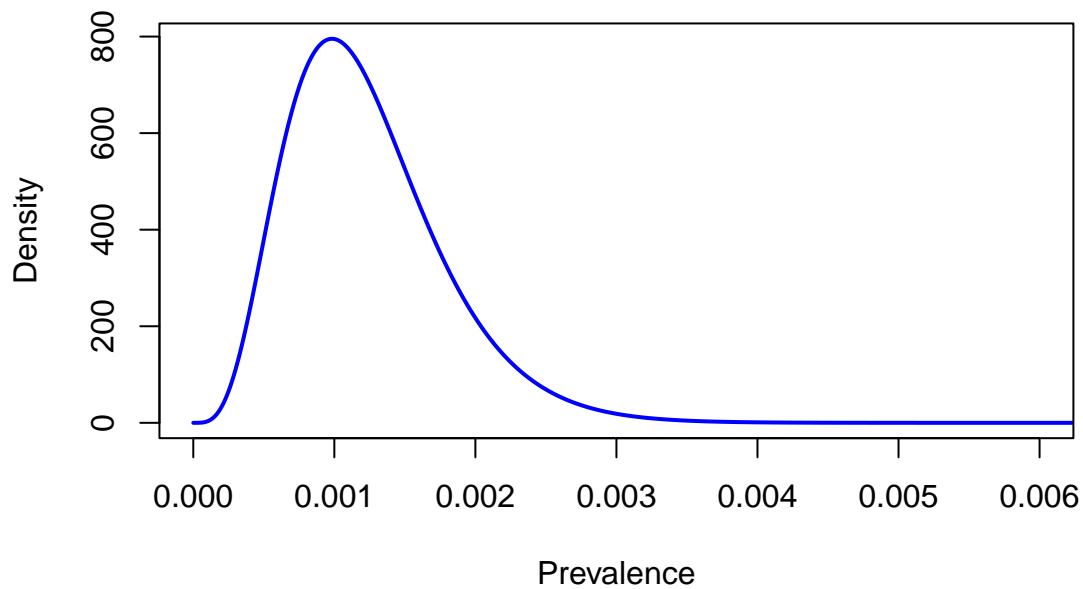
# Parameters from the German study
alpha_prior <- y_prior+1
beta_prior <- n-y_prior+1
x <- seq(0, 1, by = 0.000001)

# Create a Beta prior distribution
prior_distribution <- dbeta(x, alpha_prior, beta_prior)
```

For demonstration purposes only, we are going to plot the beta prior distribution.

```
plot(x, prior_distribution, type = "l", col = "blue", lwd = 2,
     xlab = "Prevalence", ylab = "Density", main = "Beta Prior Distribution",
     xlim = c(0,0.006))
```

Beta Prior Distribution



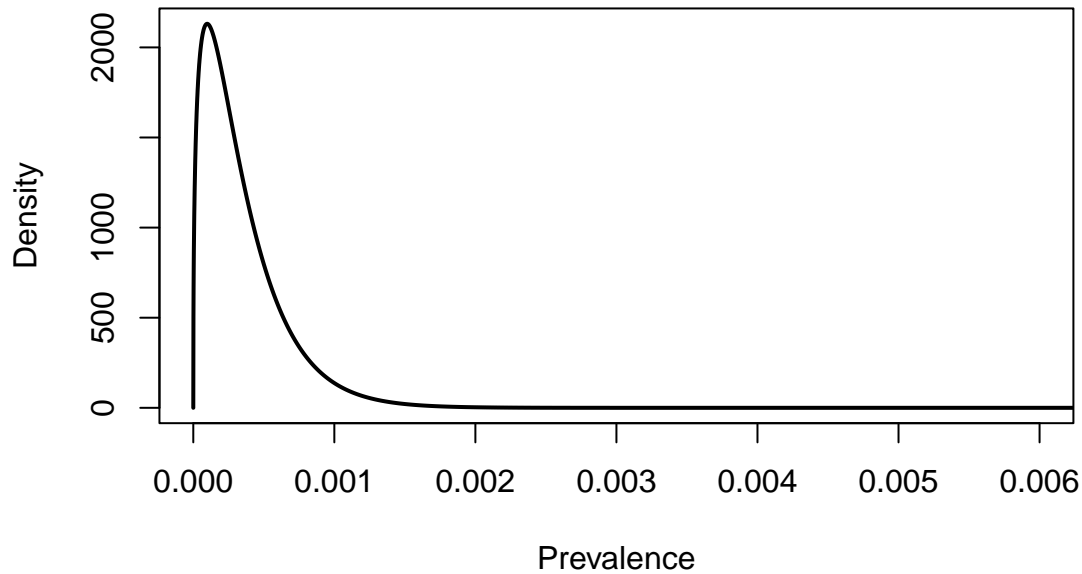
Afterwards, the scaled prior beta distribution is created. Basically, the alpha and beta prior are divided with 10, which is the scale value.

```
# Reweighted alpha and beta  
# Create a scaled Beta prior distribution  
prior_distribution_scaled <- dbeta(x, (y_prior)/10 + 1, n-y_prior/10 + 1)
```

For demonstration purposes only, we are going to plot the scaled beta prior distribution.

```
plot(x, prior_distribution_scaled, type = "l", col = "black", lwd = 2,  
     xlab = "Prevalence", ylab = "Density", main = "Beta Prior Distribution",  
     xlim = c(0,0.006))
```

Beta Prior Distribution



Task 1.2

Build the corresponding Binomial model for the number of people suffering from the disease based on the 1279 test. Obtain the theoretical posterior distribution for this scenario.

In order to obtain the theoretical posterior distribution for this scenario, first we are going to create the likelihood, which is binomial. The likelihood will be:

$$L(\theta|y) = \binom{n}{y} \theta^y (1 - \theta)^{n-y} \propto \theta^y (1 - \theta)^{n-y} = \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

As, it was mentioned in task 1.1, the conjugate prior will a beta distribution with alpha prior and beta prior:

$$\alpha = y + 1 \text{ and } \beta = n - y + 1$$

As a result, according to the lecture notes, the posterior beta distribution:

$$\theta \sim \text{Beta}_{\text{prior}}(\alpha, \beta) \longrightarrow \text{Beta}_{\text{posterior}}(\alpha + y, \beta + n - y) = \frac{\Gamma(\alpha + y + \beta + n - y)}{\Gamma(\alpha + y) \Gamma(\beta + n - y)} \cdot y^{\alpha + y - 1} \cdot (1 - y)^{\beta + n - y - 1}$$

As it was done in task 1.1, the total tests in Austria are 1279 and the positive cases are zero. The y posterior is equal to the positive cases, the alpha posterior is alpha prior (from task 1.1) + y posterior and the beta posterior is beta_prior + n - y_posterior. Therefore, in R the posterior beta distribution is created.

```
# Task 1.2
total_tests_Austria <- 1279
positive_cases_Austria <- 0
```

```
# Create a Beta posterior distribution
y_posterior <- positive_cases_Austria
alpha_posterior <- alpha_prior + y_posterior
beta_posterior <- beta_prior + n - y_posterior

posterior_distribution <- dbeta(x, alpha_posterior, beta_posterior)
```

Task 1.3

Plot the posterior density and obtain the point estimators and 95% Highest posterior density interval of the prevalence of Covid19 (=proportion of inhabitants suffering from the disease).

The point estimators, mean, mode, median and standard deviation, for a beta distribution are:

$$\text{mean} = \frac{\alpha}{\alpha+\beta}, \text{sd} = \frac{\alpha\beta}{(\alpha+\beta+1)(\alpha+\beta)^2}, \text{mode} = \frac{\alpha-1}{\alpha+\beta-2} \text{ and } \text{median} = \frac{\alpha-\frac{1}{3}}{\alpha+\beta-\frac{2}{3}}.$$

The point estimators and the 95% Highest posterior density interval are included in the plot of the posterior distribution. In order to calculate the 95% Highest posterior density interval we introduce the library HDInterval. Using this library the lower and the upper bound of the 95% interval is calculated. Despite the prior and posterior distributions, a table is presented which includes all the point estimators.

```
# install.packages("HDInterval")
suppressPackageStartupMessages(library(HDInterval))
# Mean
mean_posterior <- alpha_posterior/(alpha_posterior+beta_posterior)
# Standard Deviation
nom=(alpha_posterior*beta_posterior)
denom=((alpha_posterior+beta_posterior)^2*(alpha_posterior+beta_posterior+1))
sd_posterior <- nom/denom

mode_posterior=(alpha_posterior-1)/(alpha_posterior+beta_posterior-2)
median_posterior=(alpha_posterior-1/3)/(alpha_posterior+beta_posterior-2/3)

hdi_posterior=hdi(qbeta, 0.95, shape1=alpha_posterior, shape2=beta_posterior)
HDIlower=as.double(hdi_posterior["lower"])
HDIupper=as.double(hdi_posterior["upper"])
plotSeq=seq(HDIlower,HDIupper,length.out=100)
```

```
knitr::kable(cbind(c("Mean", "Mode", "Median",
                    "Standard Deviation",
                    "lower 95%-HPD boundary",
                    "upper 95%-HPD boundary"),
                  c(mean_posterior,
                    mode_posterior,
                    median_posterior,
                    sd_posterior,
                    HDIlower,
                    HDIupper)),
            col.names=c("Estimator", "Posterior Point Estimate"),
            caption="Point estimators for the posterior distribution based on
the prior that uses German COVID-19 cases")
```

Table 1: Point estimators for the posterior distribution based on the prior that uses German COVID-19 cases

Estimator	Posterior Point Estimate
Mean	0.000614401572868027
Mode	0.000491642084562439
Median	0.000573488448304113
Standard Deviation	7.54422021839641e-08
lower 95%-HPD boundary	0.000148398320550403
upper 95%-HPD boundary	0.001158586748934

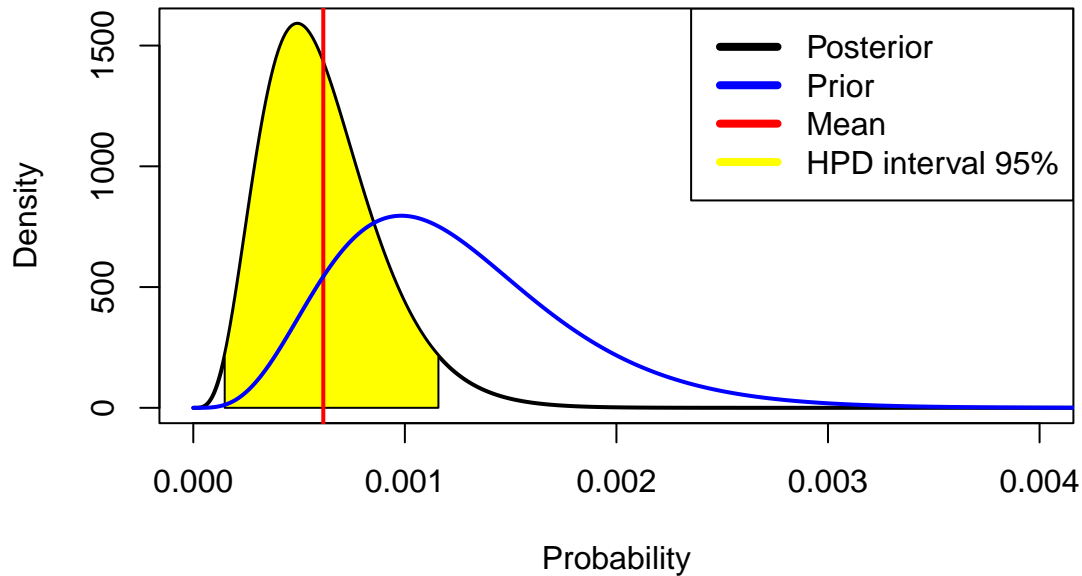
```
plot(x, posterior_distribution, type = "l", col = "black", lwd = 2,
     xlab = "Probability", ylab = "Density", main = "Beta Prior/Posterior Distribution",
     xlim = c(0,0.004))

# polygon(x_quantile,y_quantile, col="yellow")
polygon(x=c(HDIlower,
            plotSeq,
            HDIupper),
        y=c(0,
            dbeta(plotSeq,
                  shape1=alpha_posterior,
                  shape2=beta_posterior),
            0),col="yellow")

abline(v=mean_posterior,col="red",lwd=2)
lines(x,prior_distribution, col = "blue", lwd = 2)

legend ("topright",legend=c("Posterior", "Prior", "Mean" ,"HPD interval 95%"),
       lwd=4,
       col=c("black","blue","red","yellow"))
```

Beta Prior/Posterior Distribution



Task 1.4

Explain why Statistik Austria chose this method instead of simulationbased or frequentist inference for obtaining intervals of the prevalence.

Bayesian methods allow the incorporation of prior knowledge into the analysis. In this case, information from a similar study in Germany was used to form a prior distribution and therefore to form the posterior distribution from their research. Furthermore, Bayesian methods are flexible and can adapt to different data types. They can handle small sample sizes more effectively and can be useful in situations where the data is sparse, as might be the case in the early stages of a pandemic. Bayesian methods provide a natural way to quantify uncertainty. The posterior distribution gives a probability distribution over parameter values, providing a clearer picture of the range of possible values and associated uncertainty. Finally, Bayesian analysis can directly provide probabilities of different outcomes, which can be useful in decision-making contexts. This is particularly relevant in public health, where decisions often need to be made under uncertainty.

However, the Bayesian estimation is biased, because it incorporates prior information into the analysis. It is assumed a conjugate prior distribution in order then to calculate the posterior distribution. In our case the prior information derives from the German research about the positive and total cases about Covid19. Therefore the final result, which is the Austrian research, is totally depends on the German research. On the other hand, In frequentist statistics, unbiasedness is a specific property of an estimator, because our goal is the true population parameter, as the sample size increases, to be converged.

Task 2

We revisit linear models and their residual distributions. We have already learned that the distribution of residuals is assumed to be normal. Therefore, the Bayesian linear modelling will assume a normal distribution for the data $y \sim N(x^T \beta, \sigma^2)$ for a single explanatory variable scenario, we will therefore consider the inference of the linear model's coefficient β and the residual variance σ^2 .

The tasks 2.1-2.3 contain code in R for demonstrating purposes only for the different conjugate prior distributions. In these tasks, mainly, the theoretical background and the different formulas are presented regarding the Bayesian analysis, the conjugate priors and the posterior distributions.

Task 2.1

Define conjugate priors for the coefficient parameter and the residual variance independently. Explain how the parameters can be set to be uninformative. Compare different choice of prior parameters.

It is known that the Bayesian linear modelling will assume a normal distribution for the data $y \sim N(x^T\beta, \sigma^2)$ for a single explanatory variable scenario. Therefore the likelihood function will be a normal distribution.

According to slide 29 of the lecture notes, the conjugate prior for the mean μ and therefore for the beta coefficients will be a normal distribution:

$$\beta : \beta \sim N(m, s^2)$$

The conjugate prior for the residual variance will be an Inverse Gamma distribution. However, we know that the Inverse Gamma distribution is equivalent to conjugate prior for precision. Therefore:

$$\lambda = \frac{1}{\sigma^2} : \lambda \sim G(a, b)$$

where a and b are the shape and the scale parameters, respectively.

The idea behind uninformative priors in Bayesian statistics is to represent minimal prior information or beliefs about a parameter or a model. Uninformative priors are designed to be noncommittal and allow the data to strongly influence the resulting posterior distribution. In other words, they let the data “speak for itself” and do not introduce any strong biases based on prior beliefs.

As observed from the course material, specifically slide 44, the goal is to have a prior distribution that is uninformative. Therefore, the variance of the values should be quite large. In the case of slide 44, in the first scenario, the variance of the prior distribution values is large, resulting in a similar effect on the posterior. However, this is not the case in scenario 2, where the variance is smaller than the scenario 1. The range of values for the posterior distribution is small, and the results are more precise. Nevertheless, the more informative the prior, the more biased the posterior.

So, in our case, where we have a normal distribution for the beta coefficients and an Inverse gamma distribution for the residual variance, we should choose values that will result in a large variance of the values. Thus, for the prior gamma distribution in order to be uninformative, the alpha must be high and the beta must be low. Regarding the prior normal distribution, we should choose high variance for the data in order to be uninformative. Below, two plots are presented, for demonstration purposes. The first plot is a gamma distribution and the second one is a normal distribution.

Gamma distribution:

```
# Load the dgamma function from the stats package
library(stats)

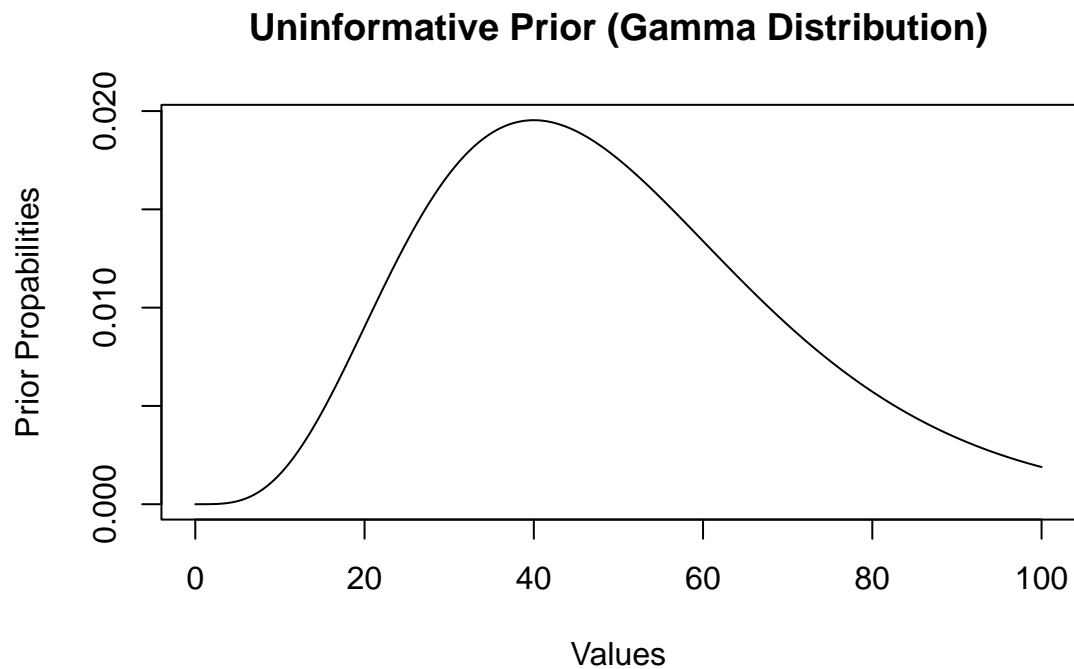
# Define the shape and scale parameters for the uninformative prior
shape_prior <- 5
scale_prior <- 10

# Generate a sequence of values for the parameter
param_values <- seq(0, 100, by = 0.1)

# Calculate the prior probabilities using the gamma distribution
prior_probs <- dgamma(param_values, shape = shape_prior, scale = scale_prior)
```



```
# Plot the uninformative prior with a gamma distribution
plot(param_values, prior_probs, type = "l",
     main = "Uninformative Prior (Gamma Distribution)",
     xlab = "Values",
     ylab = "Prior Probabilities")
```



Normal distribution:

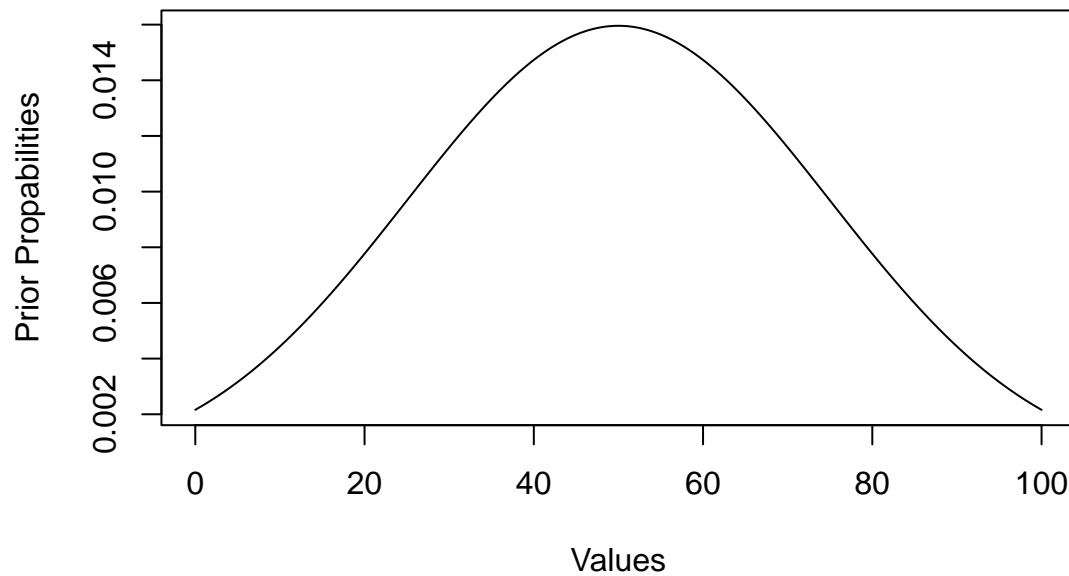
```
prior_mean <- 50
prior_sd <- 25

# Generate a sequence of values for the parameter
param_values <- seq(0, 100, by = 0.1)

# Calculate the prior probabilities
prior_probs <- dnorm(param_values, mean = prior_mean, sd = prior_sd)

# Plot the uninformative prior with a gamma distribution
plot(param_values, prior_probs, type = "l",
     main = "Uninformative Prior (Normal Distribution)",
     xlab = "Values",
     ylab = "Prior Probabilities")
```

Uninformative Prior (Normal Distribution)



Regarding the informative case, for the gamma distribution we choose alpha and beta values that are relatively close to each other. Also, regarding the normal distribution, low variance it is chosen. Below the two density plots are presented.

Gamma distribution:

```
# Load the dgamma function from the stats package
library(stats)

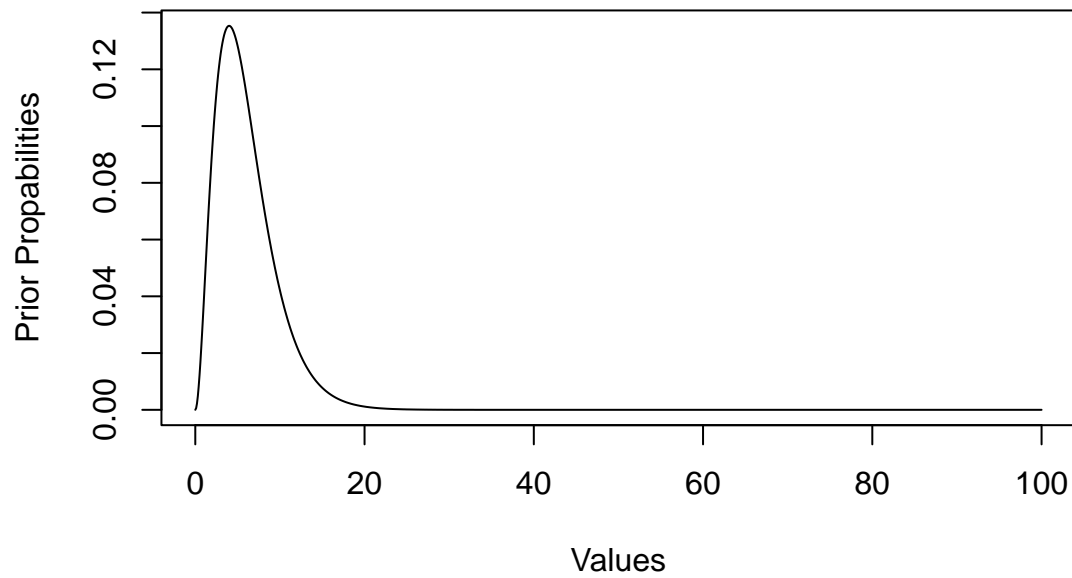
# Define the shape and scale parameters for the informative prior
shape_prior <- 3
scale_prior <- 2

# Generate a sequence of values for the parameter
param_values <- seq(0, 100, by = 0.1)

# Calculate the prior probabilities using the gamma distribution
prior_probs <- dgamma(param_values, shape = shape_prior, scale = scale_prior)

# Plot the informative prior with a gamma distribution
plot(param_values, prior_probs, type = "l",
      main = "Informative Prior (Gamma Distribution)",
      xlab = "Values",
      ylab = "Prior Probabilities")
```

Informative Prior (Gamma Distribution)



Normal distribution:

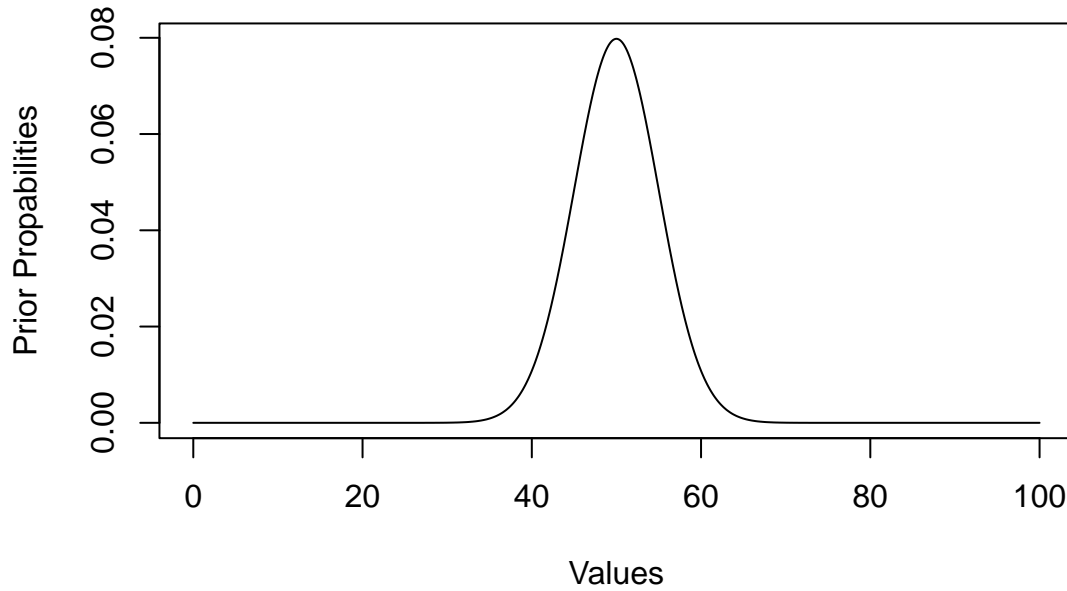
```
prior_mean <- 50
prior_sd <- 5

# Generate a sequence of values for the parameter
param_values <- seq(0, 100, by = 0.1)

# Calculate the prior probabilities
prior_probs <- dnorm(param_values, mean = prior_mean, sd = prior_sd)

# Plot the informative prior with a gamma distribution
plot(param_values, prior_probs, type = "l",
      main = "Informative Prior (Normal Distribution)",
      xlab = "Values",
      ylab = "Prior Probabilities")
```

Informative Prior (Normal Distribution)



In conclusion, the informative and uninformative nature of a normal and gamma distribution are determined by the variance, alpha and beta parameter, respectively.

Task 2.2

Build the corresponding normal model the regression inference. Obtain the theoretical posterior distribution for both parameters separately assuming the other one to be “known”.

Assuming that the residual variance is known, the theoretical posterior distribution for the beta coefficients will be:

$$\begin{aligned} \text{Conjugate Prior: } \pi(\beta) &= \frac{1}{\sqrt{2\pi}s} e^{-\frac{1}{2}\left(\frac{\beta-m}{s}\right)^2} \\ \text{Likelihood: } L(x, y|\beta) &= \prod_{i=1}^n \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{y_i - x_i\beta}{\sigma}\right)^2} \right) \end{aligned}$$

Therefore, the posterior distribution $\pi(\beta|x, y)$ will be:

$$\pi(\beta|x, y) \propto L(x, y|\beta)\pi(\beta)$$

According to slide 33 the posterior will be a normal distribution:

$$\pi(\beta|x, y) \sim N\left(\frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i y_i + \frac{1}{s^2} m}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}}, \left(\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}\right)^{-1}\right)$$

Assuming that the beta coefficients is known, the theoretical posterior distribution for the residual variance, or the precision, will be:

$$\begin{aligned}\text{Conjugate Prior: } \pi(\lambda) &= \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda} \\ \text{Likelihood: } L(x, y|\lambda) &= \prod_{i=1}^n \frac{\sqrt{\lambda}}{\sqrt{2\pi}} e^{-\frac{\lambda(y_i - x_i \beta)^2}{2}}\end{aligned}$$

$$\begin{aligned}\pi(\lambda|x, y) &\propto L(x, y|\lambda)\pi(\lambda) \\ &= \prod_{i=1}^n \frac{\sqrt{\lambda}}{\sqrt{2\pi}} e^{-\frac{\lambda(y_i - x_i \beta)^2}{2}} \cdot \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda} \\ &\propto \lambda^{\frac{n}{2}+a-1} e^{-b\lambda - \frac{\sum_{i=1}^n \lambda(y_i - x_i \beta)^2}{2}} \\ &= \lambda^{\frac{n}{2}+a-1} e^{-\lambda(b + \frac{\sum_{i=1}^n \lambda(y_i - x_i \beta)^2}{2})}\end{aligned}$$

Thus, it is easily observed that:

$$\pi(\lambda|x, y) \sim G\left(\frac{n}{2} + a, b + \frac{\sum_{i=1}^n (y_i - x_i \beta)^2}{2}\right)$$

Task 2.3

Provide the formulas for point estimators and 95% Highest posterior density interval of the regression parameters separately assuming the other one to be “known”.

Regarding the point estimators (mean, median, mode) for the posterior beta distribution, given the residual variance, is already know because it is a normal distribution. On the other hand, the points estimators for the posterior precision distribution, which is a gamma distribution, given the beta coefficients, will be:

$$\begin{aligned}\mathbb{E}(\pi(\beta|x, y)) &= \text{median}(\pi(\beta|x, y)) = \text{mode}(\pi(\beta|x, y)) = \frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i y_i + \frac{1}{s^2} m}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}} \\ \mathbb{E}(\pi(\lambda|x, y)) &= \frac{a}{b}\end{aligned}$$

Therefore, in our case where the posterior is: $\pi(\lambda|x, y) \sim G\left(\frac{n}{2} + a, b + \frac{\sum_{i=1}^n (y_i - x_i \beta)^2}{2}\right)$, the a and b will be $\frac{n}{2} + a$ and $b + \frac{\sum_{i=1}^n (y_i - x_i \beta)^2}{2}$, respectively.

As for the 95% Highest posterior density interval, for the posterior normal distribution (beta coefficients) we can use the inverse distribution function of $N\left(\frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i y_i + \frac{1}{s^2} m}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}}, \left(\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}\right)^{-1}\right)$, i.e. the quantile function, since the normal distribution is symmetric. Therefore:

$$\begin{aligned}HDP_{lower} &= N^{-1}\left(0.025, \text{mean} = \frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i y_i + \frac{1}{s^2} m}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}}, \text{variance} = \left(\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}\right)^{-1}\right) \\ HDP_{higher} &= N^{-1}\left(0.975, \text{mean} = \frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i y_i + \frac{1}{s^2} m}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}}, \text{variance} = \left(\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + \frac{1}{s^2}\right)^{-1}\right)\end{aligned}$$

As for the 95% Highest posterior density interval, for the posterior gamma distribution (precision) will be:

$$\int_{\lambda: \pi(\lambda|x, y) \geq k} \pi(\lambda|x, y) d\lambda = 1 - a, \text{ where } 1 - a = 0.95.$$

The idea is that k represents a horizontal line on the posterior density such that the point where the posterior density intersect this line the area between these points will be 0.95. However, due to the fact that gamma distribution is not symmetric, this is the only way to calculate the HDP of the posterior and therefore we cannot find. Instead, we can determine it numerically using the function `hdi` from the package `HDInterval`. This library was, also, used in Task 1.

Task 2.4

Test this with the data from your exercise 6: dataset Auto and model (mpg ~ horsepower).

From the library ISLR, we introduce the Auto data set. Below, significant information are presented.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.2.3
```

```
data(Auto)
df_auto <- Auto

# Remove NAs, if there are
df_auto <- na.omit(df_auto)

# Summary of the data
head(df_auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307         130   3504          12.0    70     1
## 2  15         8          350         165   3693          11.5    70     1
## 3  18         8          318         150   3436          11.0    70     1
## 4  16         8          304         150   3433          12.0    70     1
## 5  17         8          302         140   3449          10.5    70     1
## 6  15         8          429         198   4341          10.0    70     1
##                                name
## 1 chevrolet chevelle malibu
## 2      buick skylark 320
## 3    plymouth satellite
## 4          amc rebel sst
## 5          ford torino
## 6          ford galaxie 500
```

```
summary(df_auto)
```

```
##      mpg      cylinders      displacement      horsepower      weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
## 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0  1st Qu.: 75.0  1st Qu.:2225
## Median :22.75   Median :4.000   Median :151.0  Median : 93.5  Median :2804
## Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
## 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8  3rd Qu.:126.0  3rd Qu.:3615
## Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##
## acceleration      year      origin      name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador      : 5
## 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto       : 5
## Median :15.50   Median :76.00   Median :1.000   toyota corolla   : 5
## Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin      : 4
## 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet       : 4
## Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevete: 4
##                                (Other)      :365
```

```
str(df_auto)
```

```
## 'data.frame':   392 obs. of  9 variables:
## $ mpg          : num  18 15 18 16 17 15 14 14 15 ...
## $ cylinders    : num   8  8  8  8  8  8  8  8  8 ...
## $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
## $ horsepower   : num  130 165 150 150 140 198 220 215 225 190 ...
## $ weight       : num 3504 3693 3436 3433 3449 ...
## $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
## $ year         : num  70 70 70 70 70 70 70 70 70 70 ...
## $ origin       : num   1  1  1  1  1  1  1  1  1 ...
## $ name         : Factor w/ 304 levels "amc ambassador brougham",...: 49 36 231 14 161 141 54 223 241 1
```

Afterwards, Ordinary Least Squares regression, where the response variable will be the mpg and the explanatory variable will be the horsepower, is performed.

```
model.ls <- lm(mpg~horsepower, data = df_auto)
summary(model.ls)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = df_auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66  <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
cat("The residual variance will be:", (summary(model.ls)$sigma)**2)
```

```
## The residual variance will be: 24.06645
```

```
cat("The beta coefficients will be:", model.ls$coefficients)
```

```
## The beta coefficients will be: 39.93586 -0.1578447
```

In this part of the assignment, we are going to create the Bayesian linear model on the data set Auto. regarding the creation, we are going to use the function `stan_glm()` function. First, we introduce the appropriate libraries.

```
# Uncomment to install the appropriate packages
# install.packages("rstanarm")
# install.packages("bayestestR")
# install.packages("bayesplot")

suppressPackageStartupMessages(library(rstanarm))
suppressPackageStartupMessages(library(bayestestR))
suppressPackageStartupMessages(library(bayesplot))
```

The Bayesian model will be:

```
model_bayes <- stan_glm(mpg~horsepower, data=df_auto, seed=12223236)
```

The output of the model will be:

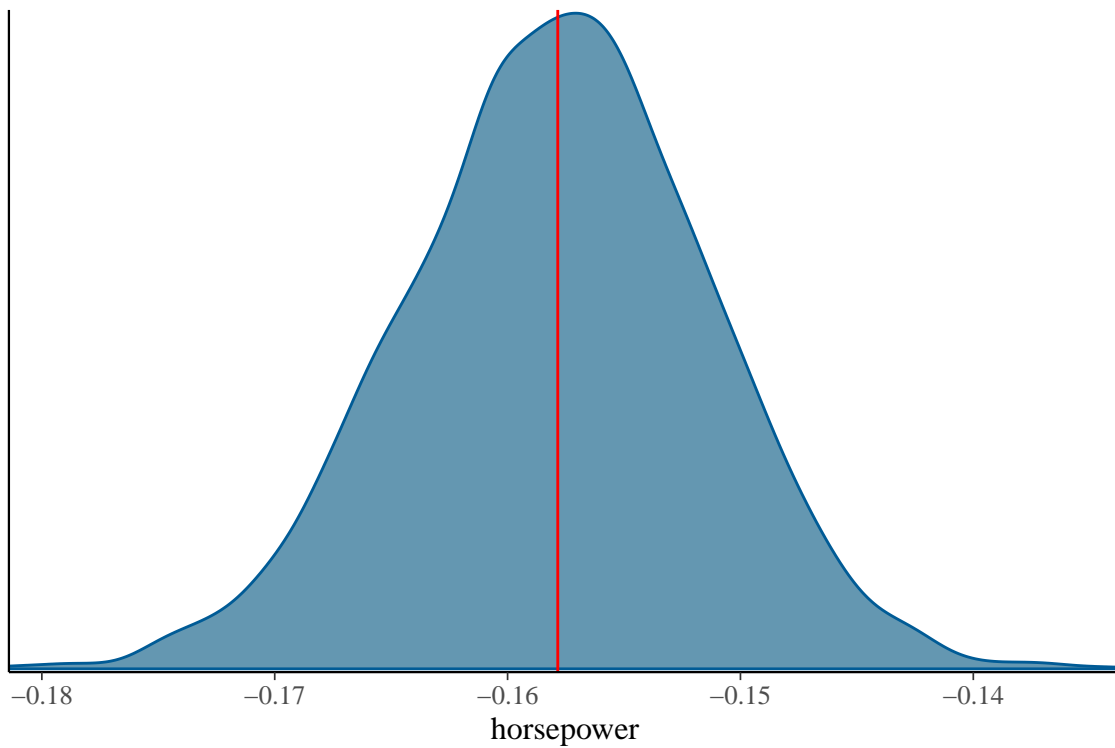
```
model_bayes

## stan_glm
## family:      gaussian [identity]
## formula:     mpg ~ horsepower
## observations: 392
## predictors:  2
## -----
##              Median MAD_SD
## (Intercept) 39.9    0.7
## horsepower  -0.2    0.0
##
## Auxiliary parameter(s):
##           Median MAD_SD
## sigma 4.9    0.2
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

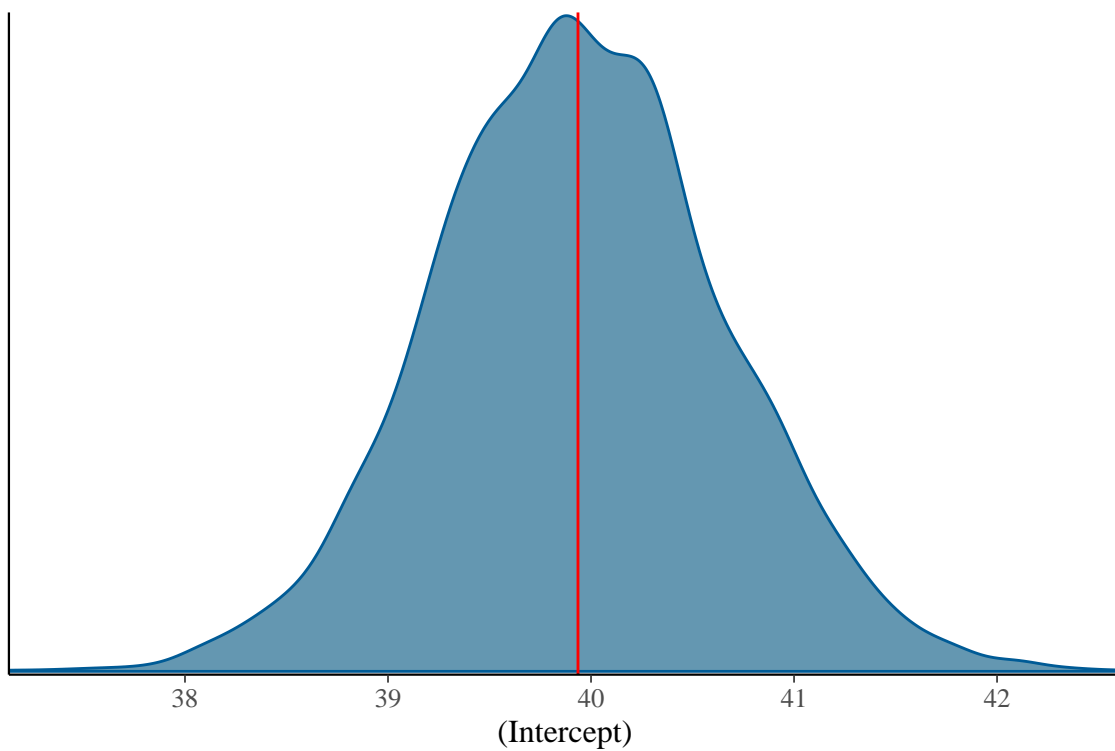
The median estimate is derived from the MCMC (Monte Carlo Markov Chain) simulation, while MAD_SD represents the median absolute deviation calculated from the identical simulation. The sigma (Auxiliary parameter(s)) is the residual standard deviation and is estimated to be 4.9 with a MAD_SD of 0.2. This represents the variability of the observed “mpg” values around the predicted values.

To gain a better understanding of how these results are obtained, we visualize the MCMC simulation for horsepower variable, which is the only predictor, using bayesplot.

```
mcmc_dens(model_bayes, pars = c("horsepower"))+
  vline_at(model_ls$coefficients[2], col="red")
```

```
mcmc_dens(model_bayes, pars = c("(Intercept)"))+  
  vline_at(model.ls$coefficients[1], col="red")
```

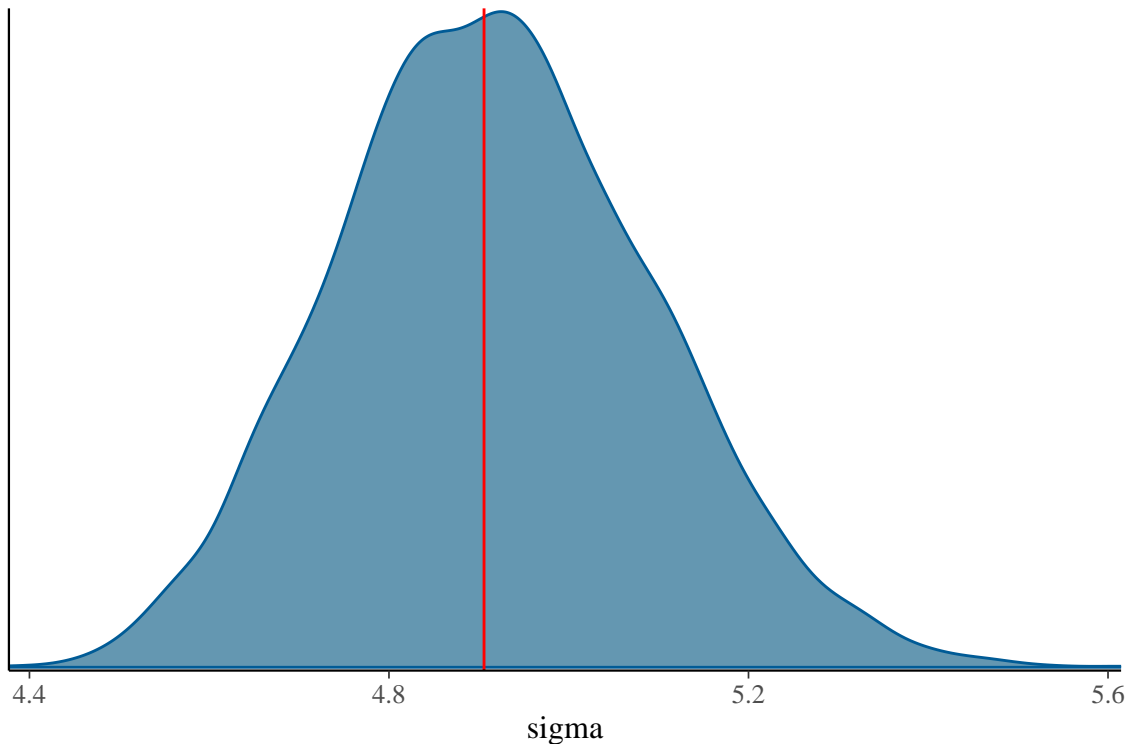


As it is observed from the plot above, the distribution of the coefficients from the variable horsepower and

the intercept follows a normal distribution. The red lines represent the beta coefficients from the ordinary least squares regression, that was performed before.

Furthermore, according to the plot below, the same applies for the residual standard deviation. The red line is the residual standard deviation from the ordinary least squares regression.

```
mcmc_dens(model_bayes, pars = c("sigma"))+  
  vline_at((summary(model.ls)$sigma), col="red")
```



Based on the above plots, it appears that the results of ordinary least squares regression align to a large extent with the results of Bayesian regression. This indicates that both methods are quite effective in estimating the beta coefficients as well as the residual standard deviation/variance.

It is important to note that this similarity may be due to various factors, such as the nature of the data, the specifications of the model, and the assumptions of the model. The choice between ordinary least squares regression and Bayesian regression often depends on the nature of the problem and the goals of the analysis, and in our case, both methods seem to be compatible.

Finally, as we do with classical regression (frequentist), we can test the significance of the Bayesian regression coefficients by checking the corresponding credible interval.

```
hdi(model_bayes)
```

```
## Highest Density Interval  
##  
## Parameter      |          95% HDI  
## -----  
## (Intercept)    | [38.55, 41.35]  
## horsepower     | [-0.17, -0.15]
```