Introduction

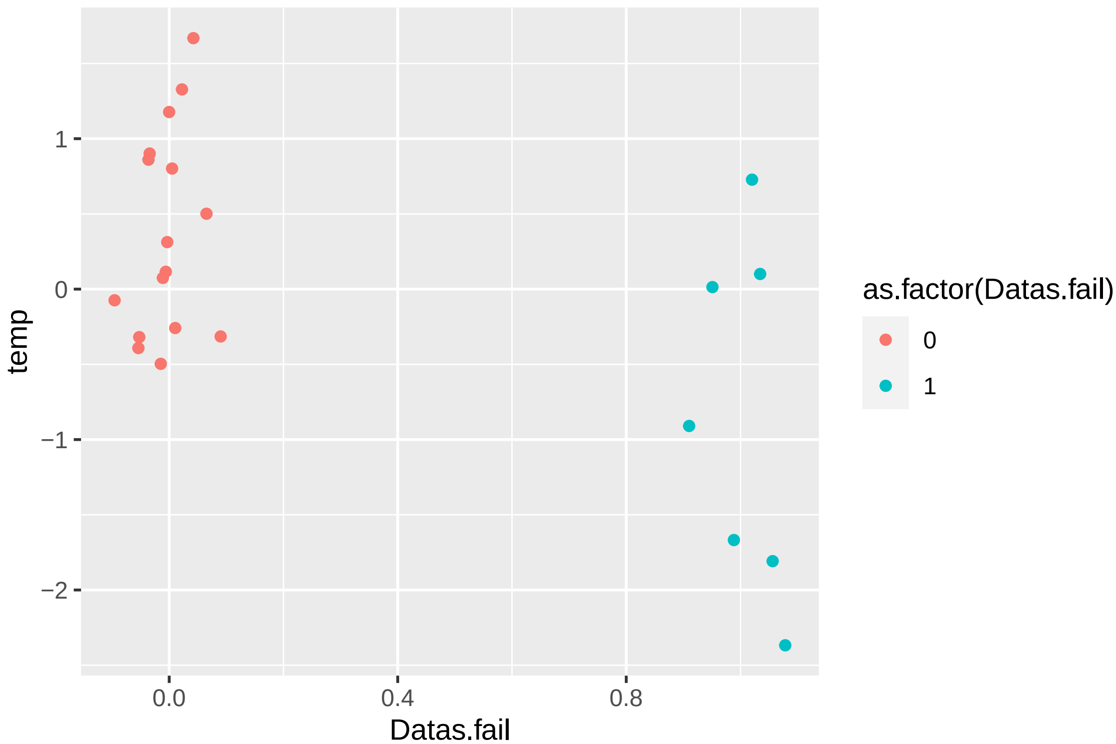
The Space Shuttle Challenger exploded 73 second after liftoff on January 28th, 1986. The disaster claimed the lives of all seven astronauts on board, including school teacher Christa McAuliffe.1 The details surrounding this disaster were very involved. The main concern of engineers in launching the Challenger was the evidence that the large O-rings sealing the several sections of the boosters could fail in cold temperatures.

In this analysis we mainly use the data from previous launches to get an idea about how the O-rings faliures are associated with covariates such as surrounding environment temperature and pressure.

Exploratory Data Analysis

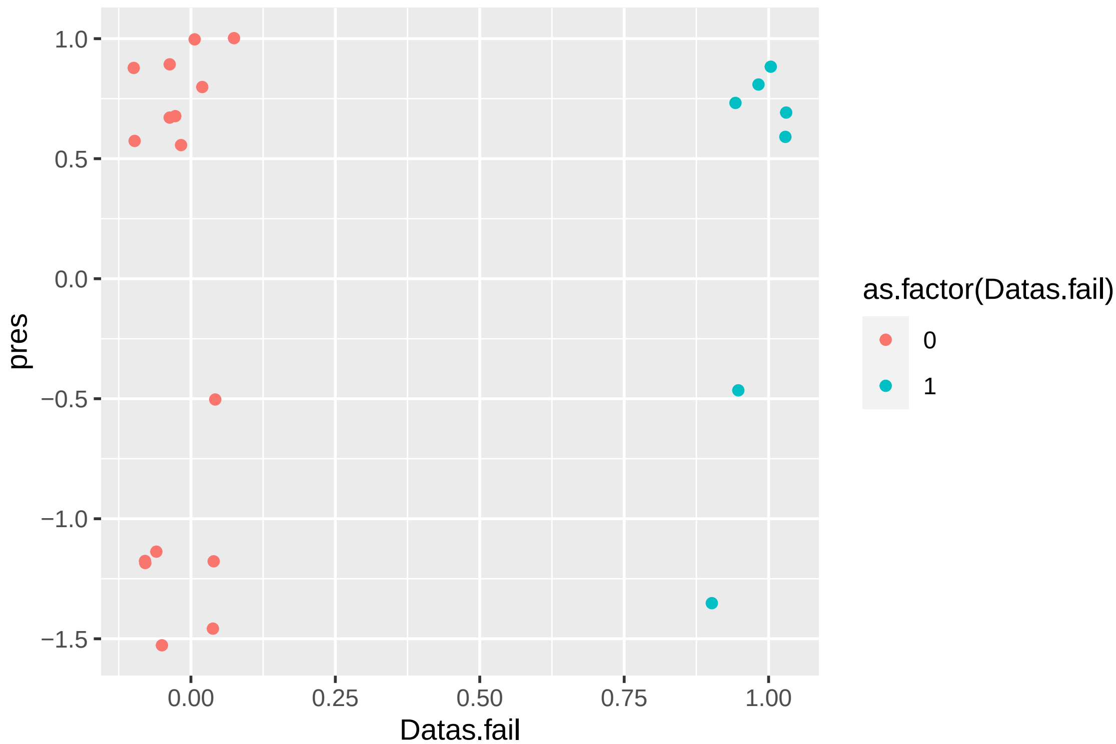
We first plot, the responses with individual covariates to get idea about their effects.

```r
library(alr4)
library(ggplot2)
library(glmnet)

# Data loading
Datas <- Challeng
Datas <- Datas[,1:3]
rownames(Datas) <- NULL

normalize <- function(x)
{
  return((x-mean(x))/sd(x))
}


Datas <- data.frame(lapply(Datas[,c(1,2)], normalize),Datas$fail)
Datas$Datas.fail[which(Datas$Datas.fail == 2)] <- 1
# EDA
ggplot(Datas, aes(y=temp, x=Datas.fail)) +
geom_jitter(aes(colour=as.factor(Datas.fail)),width = 0.1)
```



from the plot of Y versus x_1 i.e. temperature, we can see that there is clear negative dependence of temperature with chance of failure as higher the temperature, lesser 1 values are observed.

```
ggplot(Datas, aes(y=pres, x=Datas.fail)) +
geom_jitter(aes(colour=as.factor(Datas.fail)),width = 0.1)
```



Whereas, in case of pressure, no such trend is noted as if the failure doesn't even depend on pressure.

Logistic Regression Model

As the begining, we fit a logistic model with Y as the response and x_1, x_2 as the covariates. Since this is a frequentist model, for the time being we don't impart any prior information for the parameters. We just compute their MLEs using **glm** R package. Here, are the codes required for that :-

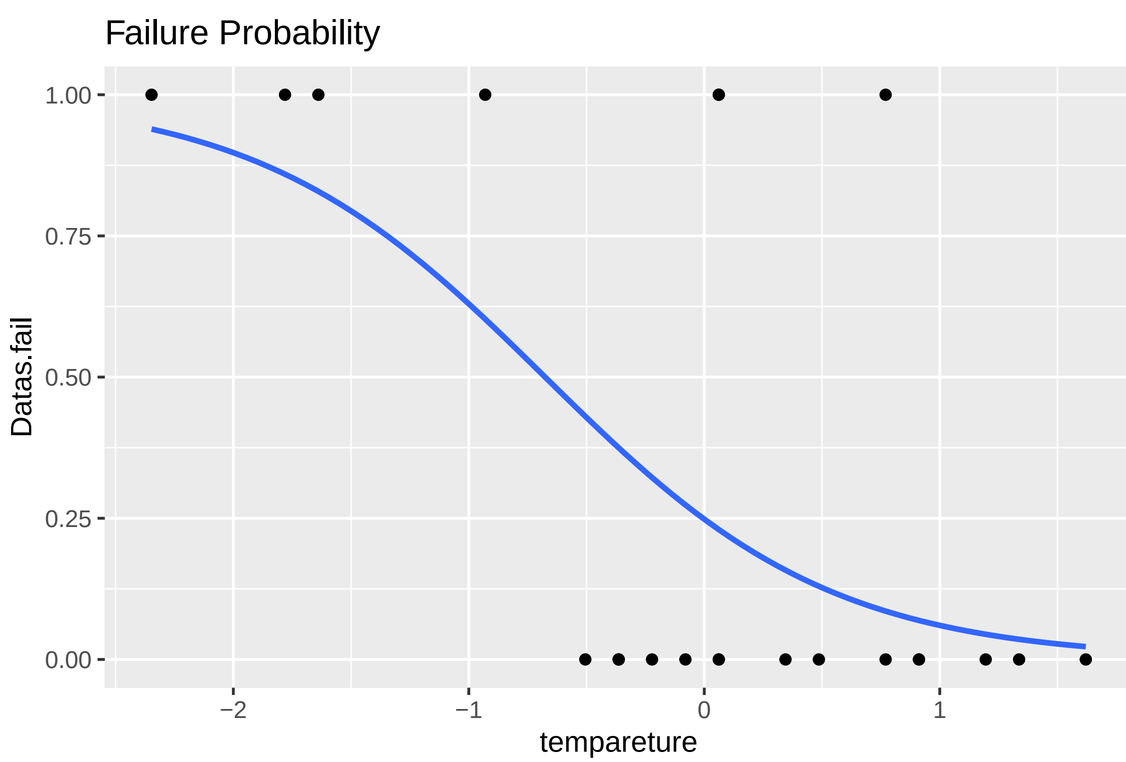
```
mod.glm <- glm(formula = Datas.fail ~ .,family = "binomial",data = Datas)
summary(mod.glm)


Call:
glm(formula = Datas.fail ~ ., family = "binomial", data = Datas)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2130  -0.6089  -0.3870   0.3472   2.0928

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.0543     0.5902  -1.786   0.0740 .
temp         -1.7046     0.7743  -2.201   0.0277 *
pres          0.6728     0.6142   1.095   0.2733
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28.267  on 22  degrees of freedom
Residual deviance: 18.972  on 20  degrees of freedom
AIC: 24.972

Number of Fisher Scoring iterations: 5
```

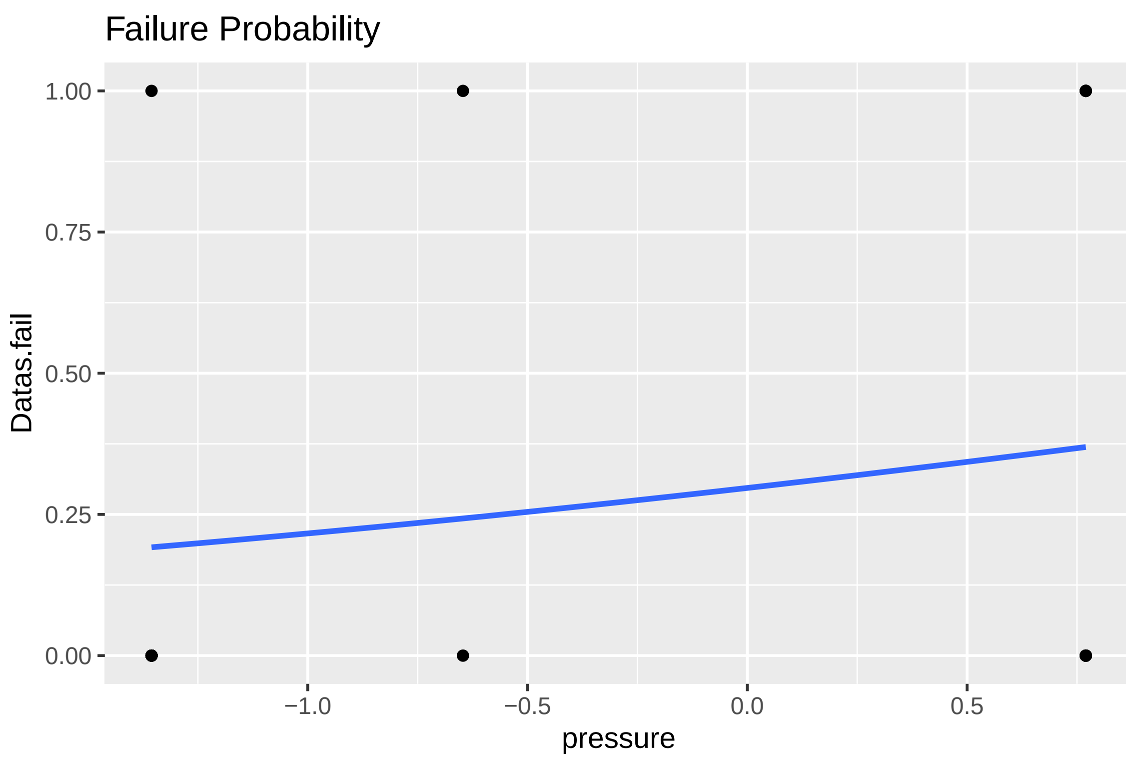
Here, from the summary output, we can clearly see that the **temp** variable is statistically significant at 5% level whereas **pres** is not. This also statistically signifies our suspicion at the begining. We will use these estimates as initial guesses in the upcoming MCMC implementations.

Next, to visualize, we plot the fitted logistic model seperately for the covariates **temp** and **pres**.

```
ggplot( Datas, aes(x=temp, y=Datas.fail)) +
  geom_point() +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
labs(title = "Failure Probability",x = "tempareture",ylab = "Failure")
```



```r
# pres
ggplot( Datas, aes(x=pres, y=Datas.fail)) +
  geom_point() +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
labs(title = "Failure Probability",x = "pressure",ylab = "Failure")
```



Bayesian Model 1

After the frequentist exploration, now we move on to the Bayesian paradigm. We write the logistic model as :-

$$P\left(Y_i = 1|x_i\right) = \frac{e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}{1+e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}} = \pi_i, i = 1, \ldots, n$$

where x_1 and x_2 are the normalized variables temperature and pressure during the launch and Y represents the indicator variable whether any of the O-rings failed or not.

We consider both the variables x_1 and x_2 to be centered and scaled mainly because of computation issues with the logit link function. Obviously we can obtain the coefficients in terms of original variables easily.

The likelihood function of $\boldsymbol{\beta}$ given the observed data $\boldsymbol{y}, \boldsymbol{X}$ can be written as :-

$$\begin{aligned} f\left(y_i|\boldsymbol{\beta}, x_i\right) &= \pi_i^{y_i}\left(1-\pi_i\right)^{1-y_i} \\ &= \left[\frac{e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}{1+e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}\right]^{y_i}\left[\frac{1}{1+e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}\right]^{1-y_i} \end{aligned}$$

now, if we assume diffused normal prior for $\boldsymbol{\beta} \sim N_3\left(\mathbf{0}, \lambda^{-1}\boldsymbol{I}_3\right)$ for small value of λ as we don't consider any specific information to be imparted in the prior :-

$$\pi\left(\boldsymbol{\beta}\right) \propto \exp\left(-\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right)$$

Hence, the posterior of $\boldsymbol{\beta}$ can be written as :-

$$\begin{aligned}
\pi\left(\boldsymbol{\beta}|\boldsymbol{x},\boldsymbol{y}\right) &\propto \pi\left(\boldsymbol{\beta}\right) f\left(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{x}\right) \\
&\propto \exp\left(-\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right)\prod_{i=1}^{n}\left[\frac{e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}{1+e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}\right]^{y_i}\left[\frac{1}{1+e^{\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}}}\right]^{1-y_i} \\
&\propto \exp\left(-\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right)\prod_{i=1}^{n}\left[\frac{e^{y_i\boldsymbol{\beta}^T\boldsymbol{x_i}}}{1+e^{\boldsymbol{\beta}^T\boldsymbol{x_i}}}\right]\left(=\widetilde{p}\left(\boldsymbol{\beta}\right)\right)
\end{aligned}$$

Now, in order to draw samples from this posterior distribution of $\boldsymbol{\beta}$, we use the following MCMC algorithm.

- We have to draw samples from the posterior distribution of $\boldsymbol{\beta}$ which can be written as $p\left(\boldsymbol{\beta}\right) = \frac{\widetilde{p}(\boldsymbol{\beta})}{Z_p}$ where, Z_p denotes the intractible normalizing constant and $\widetilde{p}\left(\boldsymbol{\beta}\right)$ denotes the part that is easily computable.
- Now, we select our proposal distribution as $q\left(\boldsymbol{\beta}|\boldsymbol{\beta}^{(\tau)}\right)$ where $\boldsymbol{\beta}^{(\tau)}$is the current iterate of $\boldsymbol{\beta}$. For implementing the basic Markov Chain Monte Carlo, we choose, $q\left(\right)$ to be a symmetric distribution i.e.,

$$q\left(\boldsymbol{\beta}|\boldsymbol{\beta}^{(\tau)}\right) \sim N_3\left(\boldsymbol{\beta}^{(\tau)}, \Sigma\right)$$

  which is a trivariate normal density with mean $\boldsymbol{\beta}^{(\tau)}$ and variance covariance matrix Σ. We take $\Sigma = \text{diag}\left(\sigma_1, \sigma_2, \sigma_3\right)$ where σ_i are chosen in such a manner that the target distribution is neither explored too slowly such that it gets stuck in a mode even if the posterior is multimodal, nor too large that the acceptance probbility becomes too low.
- Finally, in an iteration τ, where current value is $\boldsymbol{\beta}^{(\tau)}$, we select a new value $\boldsymbol{\beta}^*$ if $u < A\left(\boldsymbol{\beta}^*, \boldsymbol{\beta}^{(\tau)}\right)$ where :
  - $u \sim U\left(0,1\right)$ is an uniform random sample.
  - $A\left(\boldsymbol{\beta}^*, \boldsymbol{\beta}^{(\tau)}\right)$ is the acceptance probability defined as $A\left(\boldsymbol{\beta}^*, \boldsymbol{\beta}^{(\tau)}\right) = \min\left(1, \frac{\widetilde{p}(\boldsymbol{\beta}^*)}{\widetilde{p}\left(\boldsymbol{\beta}^{(\tau)}\right)}\right)$.
  - then we set $\boldsymbol{\beta}^{(\tau+1)} = \boldsymbol{\beta}^*$ and proceed.
- Otherwise also we set $\boldsymbol{\beta}^{(\tau+1)} = \boldsymbol{\beta}^{(\tau)}$ and draw samples from proposal distribution $q\left(\boldsymbol{\beta}|\boldsymbol{\beta}^{(\tau+1)}\right)$.

We draw $B = 5 \times 10^4$ many samples from the posterior distribution using MCMC algorithm devised above and burn the first 10% samples also use a thinning gap of 5 to avoid significant correlations between the observations. Here are the relevant R codes for the analysis :-

```r
likelihood1 <- function(X,y,beta,lambda = 0.01,M = 100)
{
  beta = matrix(beta,nrow = 1)
  a = exp(-(lambda/2)*beta%*%t(beta))
  b = exp(y*(beta%*%t(X)))
  c = exp(beta%*%t(X))
  return(M*a*prod(b/(1+c)))
}

MCMC.Sampler1 <- function(X,y,beta0,B,sg = c(1,1,1),showprogress = TRUE,...)
{
  X = cbind(rep(1,nrow(X)),X)
  beta0 = matrix(beta0,nrow = 1)
  post.sample = c(0,0,0)
  beta1 = beta0
  beta2 = matrix(c(0,0,0),nrow = 1)
  prog = txtProgressBar(max = B,style = 3)
  for(i in 1:B)
  {
    beta2[1] = beta1[1] + rnorm(1,0,sg[1])
    beta2[2] = beta1[2] + rnorm(1,0,sg[2])
    beta2[3] = beta1[3] + rnorm(1,0,sg[3])
    ratio = likelihood1(X,y,beta = beta2,...)/likelihood1(X,y,beta1,...)
    unif = runif(1)
    if(unif <= min(1,ratio)) beta1=beta2
    post.sample = rbind(post.sample,beta1)
    if(showprogress) setTxtProgressBar(pb = prog,value = i)
  }
  close(prog)
  return(post.sample)
}
```

Using this manual function we draw the stated number of posterior samples and make inference from them

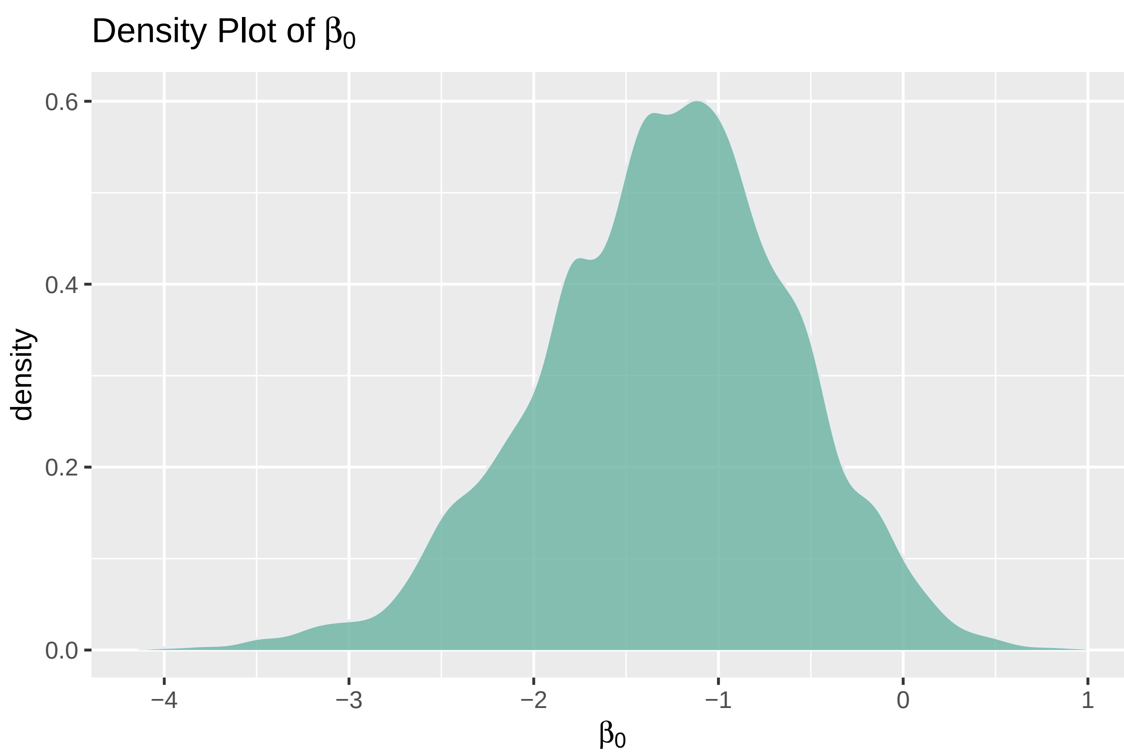
```r
# MCMC parameters
B = 5*10^4
n.thin = 5

# Running the MCMC sampler
Post.Sample1 = MCMC.Sampler1(X = Datas[,1:2],y = Datas$Datas.fail,beta0 =
c(mod.glm$coefficients[1],mod.glm$coefficients[2],mod.glm$coefficients[3]),B,
sg = c(3,3,3),showprogress = FALSE,lambda=0.001)
```

```
  |
  |                                                                      |   0%
```

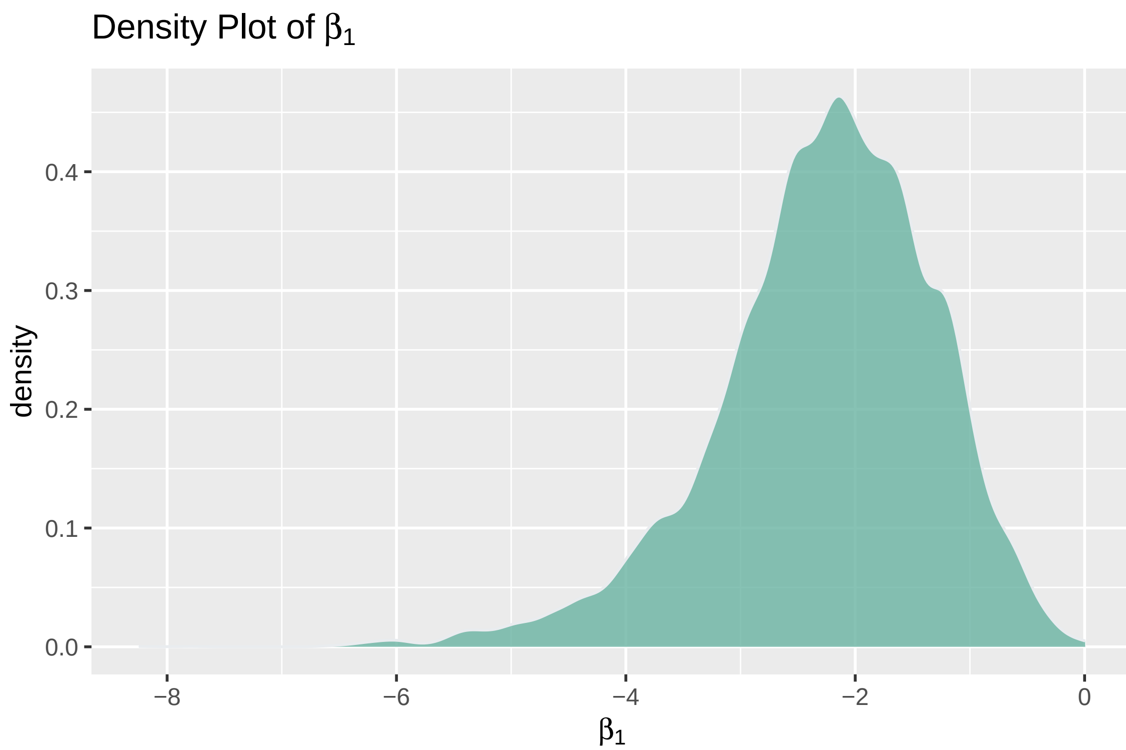
```r
Post.Sample1 = (Post.Sample1)[-(1:(B/10)),]
n.length = nrow(Post.Sample1)
batch.size = floor(n.length/n.thin)
Post.Sample1 = Post.Sample1[n.thin*(1:batch.size),]
Post.Samp1 = data.frame(Post.Sample1)
names(Post.Samp1) <- c('b0','b1','b2')
```

Now, using the generated posterior samples, we plot the posterior densities of $\boldsymbol{\beta}$ individually.

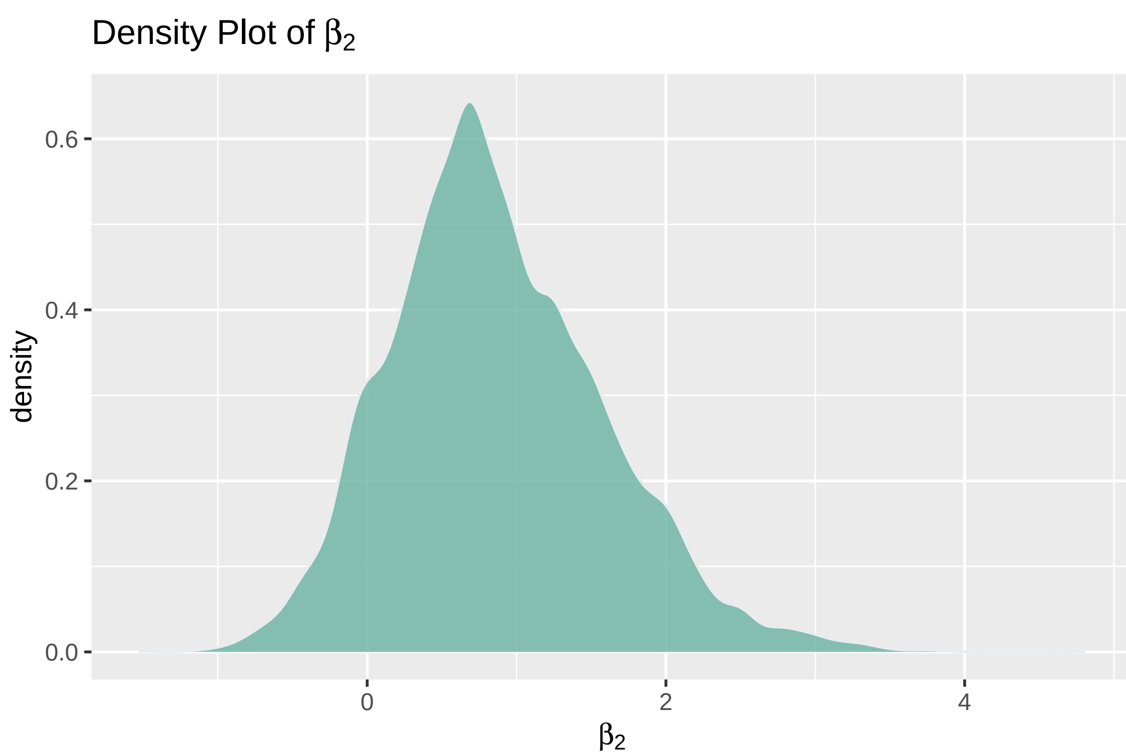
```r
# Posterior distributions of beta0,beta1
ggplot(data = Post.Samp1,aes(x=b0)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
    labs(title = bquote("Density Plot of" ~ beta[0]),x = bquote(beta[0]))
```




```r
ggplot(data = Post.Samp1,aes(x=b1)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
  labs(title = bquote("Density Plot of" ~ beta[1]),x = bquote(beta[1]))
```




```r
ggplot(data = Post.Samp1,aes(x=b2)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
  labs(title = bquote("Density Plot of" ~ beta[2]),x = bquote(beta[2]))
```

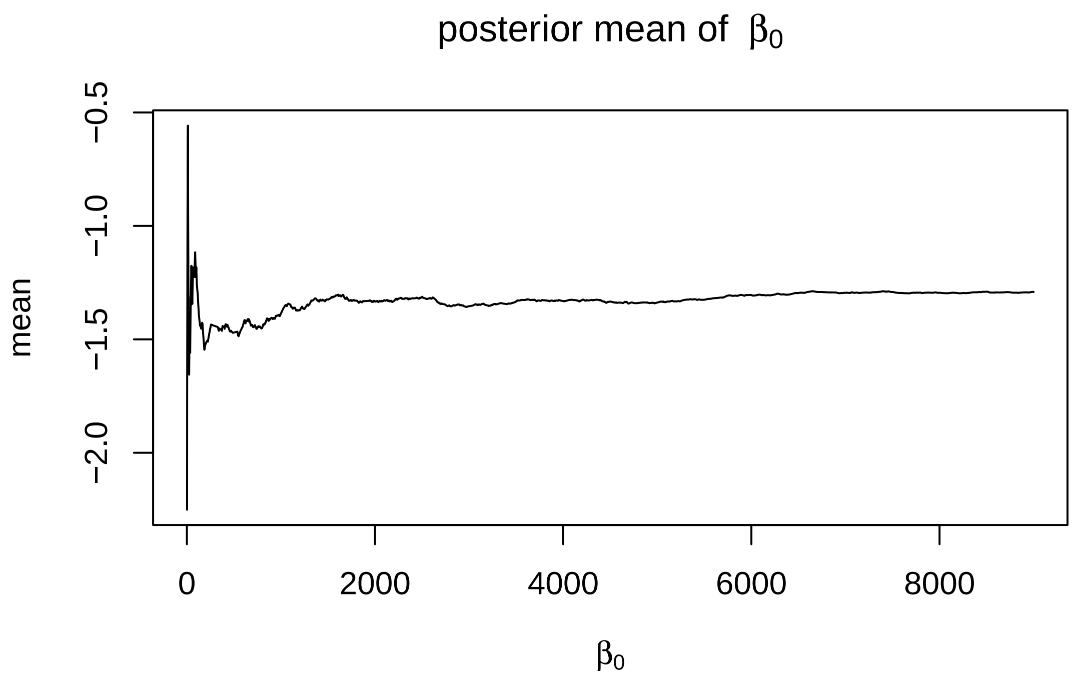


The posterior means for the three parameters are :-

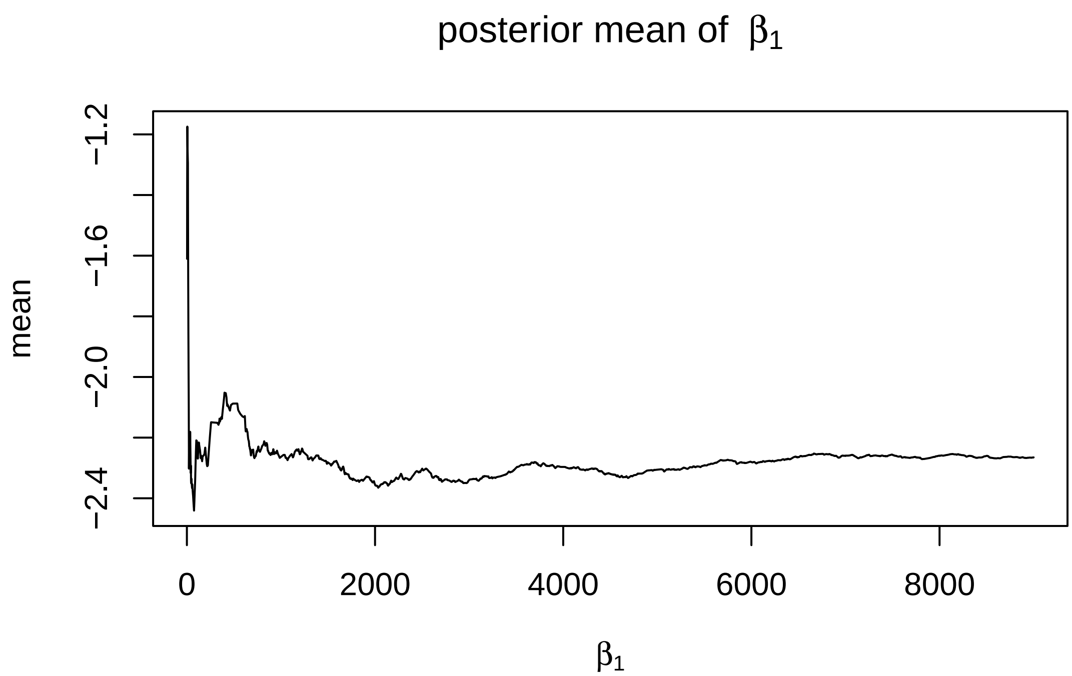
```r
m1 = apply(Post.Samp1, 2, mean)
m2 = mod.glm$coefficients
data.frame("Posterior.Means" = m1,"Logistic.Coef" = m2)

   Posterior.Means Logistic.Coef
b0       -1.291231    -1.0543469
b1       -2.265075    -1.7045661
b2        0.872008     0.6728236
```

To know whether these estimates are more or less consistent or not (ergodicity) we plot the cumulative means of these posterior samples.

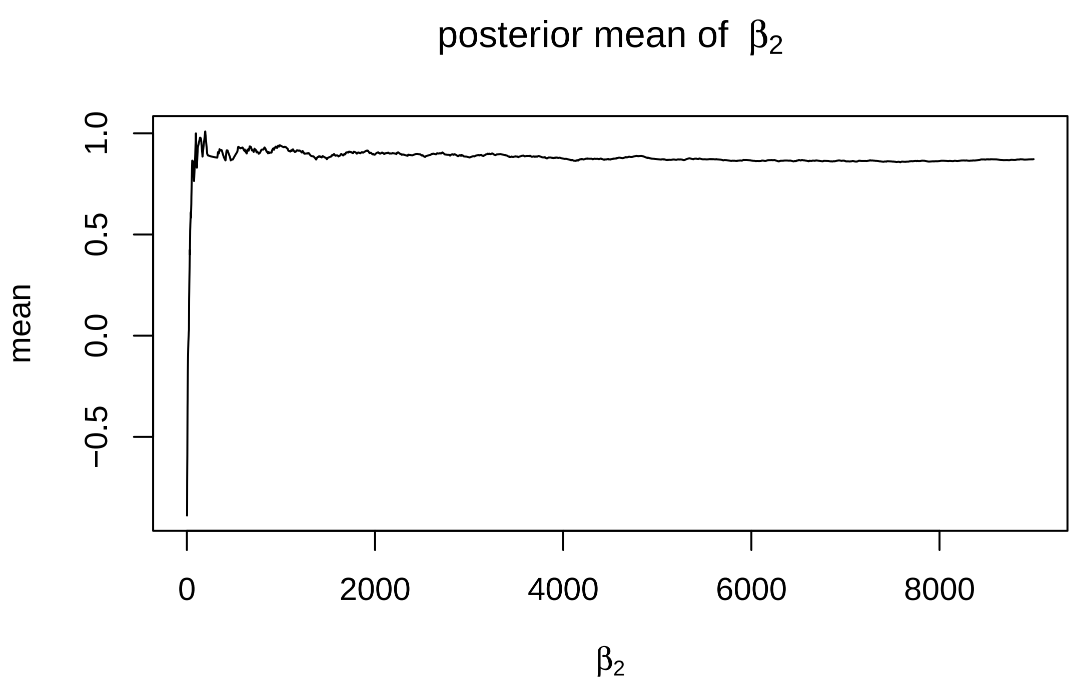
```r
# plotting the mean cumulatively w.r.t sample size
b0.mean.cum <- cumsum(Post.Samp1$b0)/(1:nrow(Post.Samp1))
b1.mean.cum <- cumsum(Post.Samp1$b1)/(1:nrow(Post.Samp1))
b2.mean.cum <- cumsum(Post.Samp1$b2)/(1:nrow(Post.Samp1))
# plot of means with increasing sample size
plot(b0.mean.cum,type = "l",main = bquote("posterior mean of " ~ beta[0]),xlab = bquot
```



```
plot(b1.mean.cum,type = "l",main = bquote("posterior mean of " ~ beta[1]),xlab = bquot
```



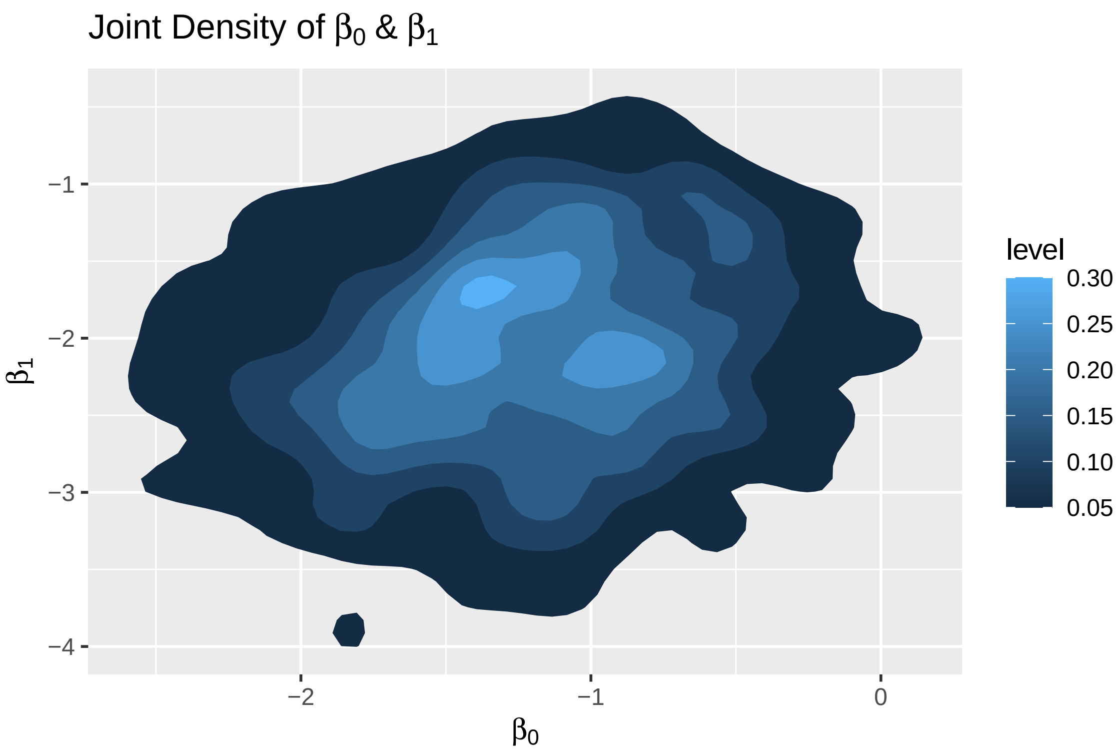

```
plot(b2.mean.cum,type = "l",main = bquote("posterior mean of " ~ beta[2]),xlab = bquot
```



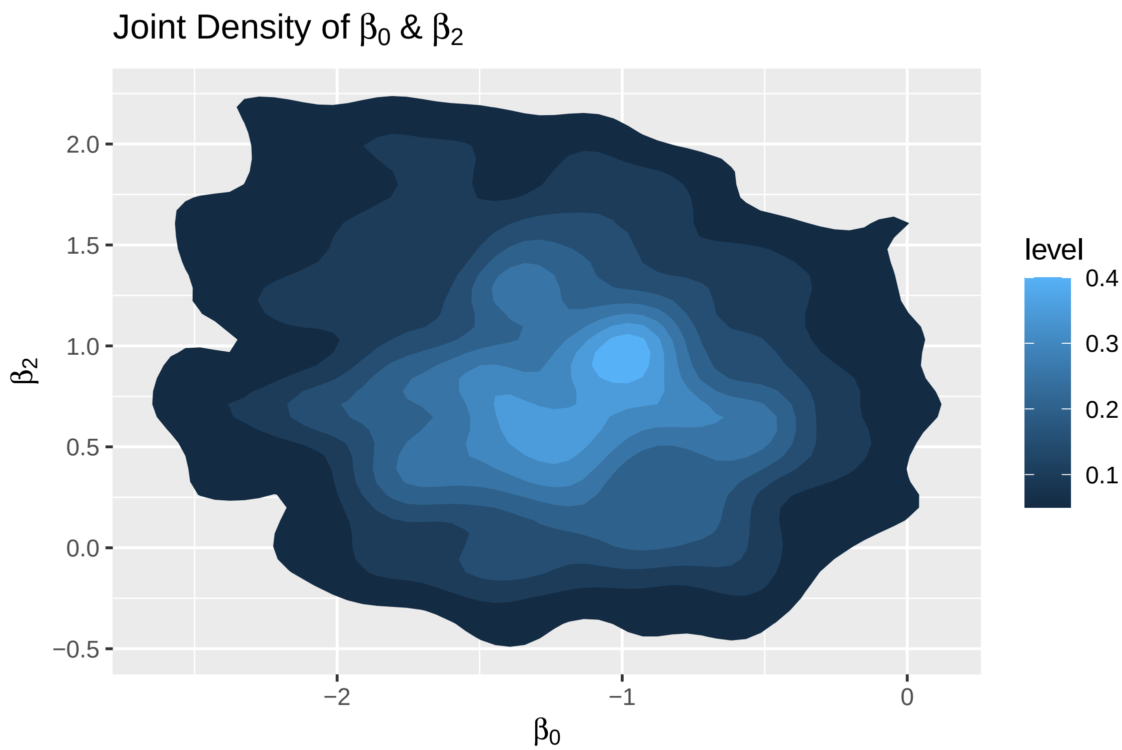
With increasing sample size, we can see that the mean more or less gets stabilized indicating their consistency.

Here's another plot which may provide better idea through bivariate density plots taking two variables at a time where the density is shown using varying colour density.

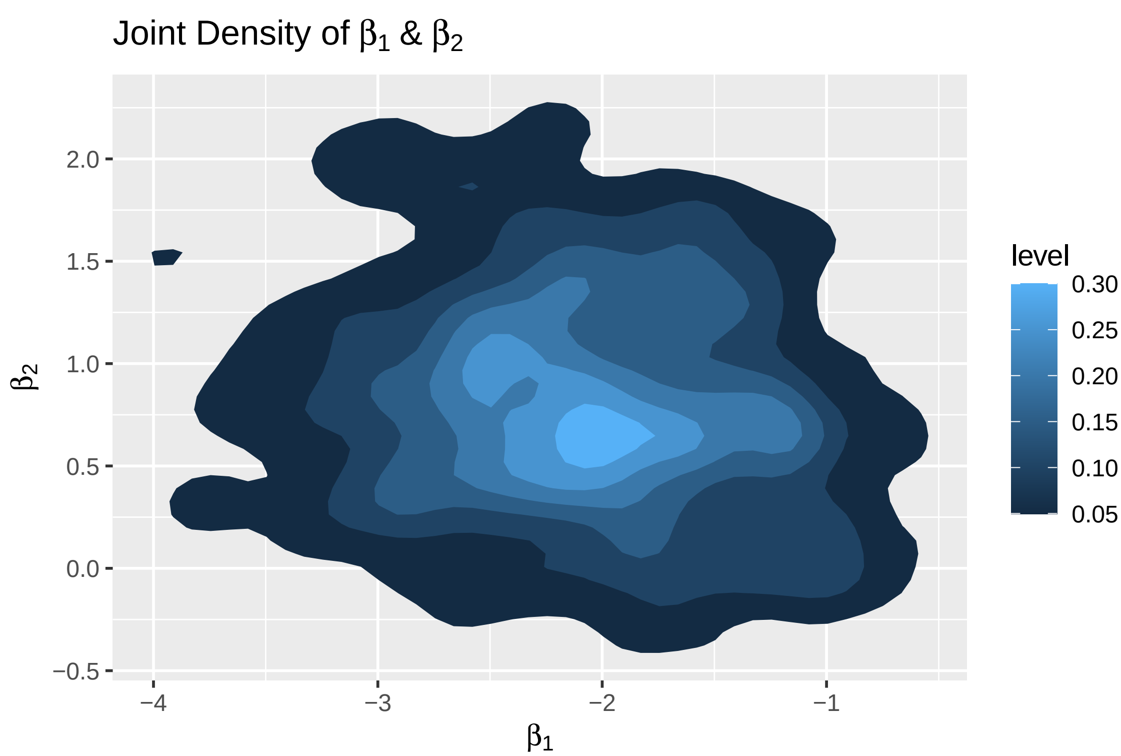
```
# b0,b1
ggplot(Post.Samp1, aes(x = b0, y = b1, fill = ..level..)) +
  stat_density_2d(geom = "polygon") +
labs(title = bquote("Joint Density of" ~ beta[0] ~ "&" ~ beta[1]),
x = bquote(beta[0]), y = bquote(beta[1]))
```




```
# b0,b2
ggplot(Post.Samp1, aes(x = b0, y = b2, fill = ..level..)) +
  stat_density_2d(geom = "polygon") +
labs(title = bquote("Joint Density of" ~ beta[0] ~ "&" ~ beta[2]),
x = bquote(beta[0]), y = bquote(beta[2]))
```

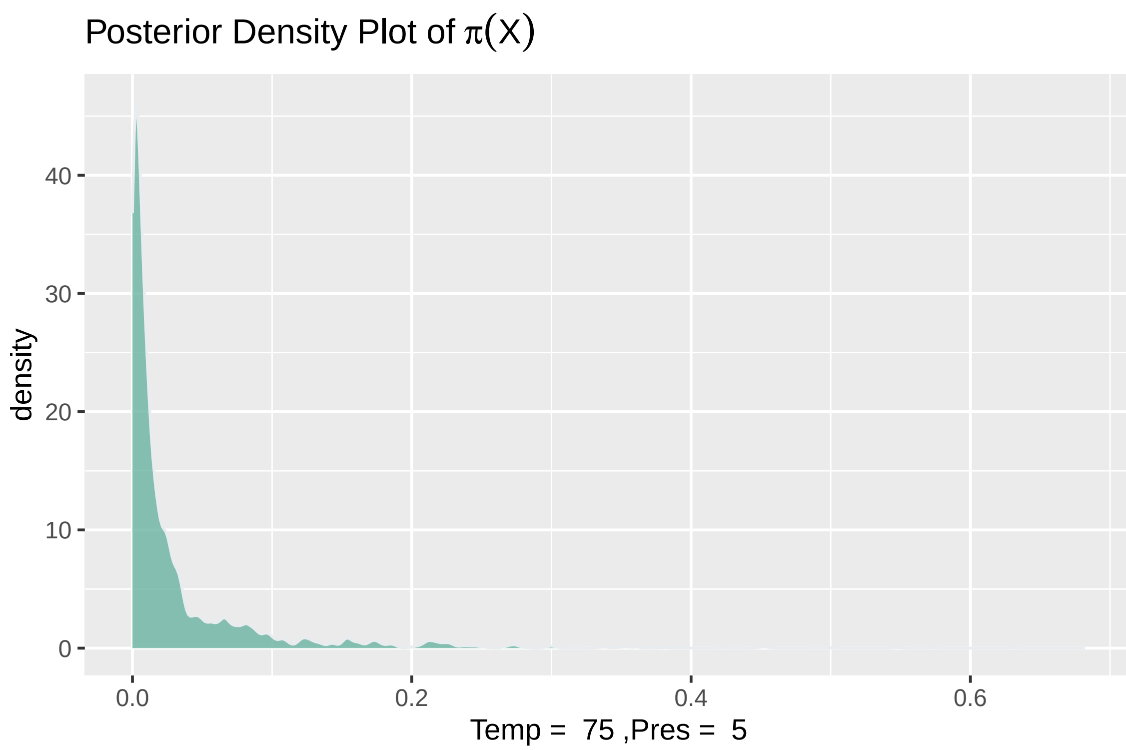



```r
# b1,b2
ggplot(Post.Samp1, aes(x = b1, y = b2, fill = ..level..)) +
  stat_density_2d(geom = "polygon") +
labs(title = bquote("Joint Density of" ~ beta[1] ~ "&" ~ beta[2]),
x = bquote(beta[1]), y = bquote(beta[2]))
```

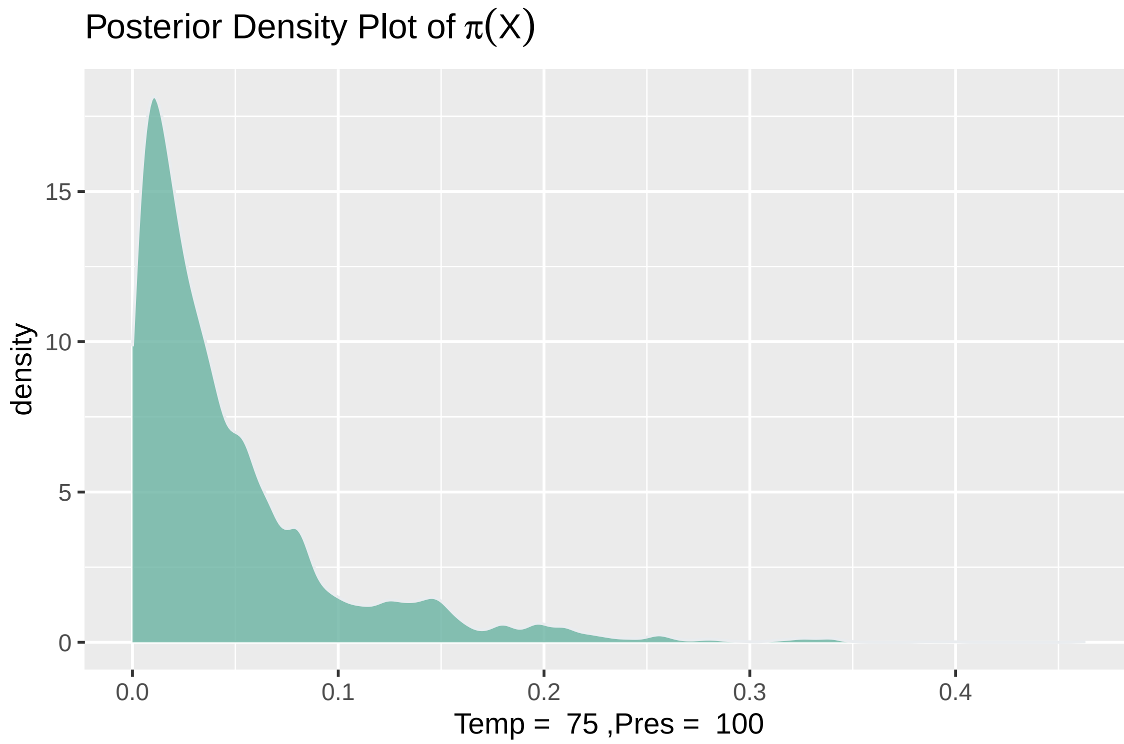


The most important thing to visualize is how we can model the posterior probability distribution of that $\pi(y|\boldsymbol{X}) = P(Y=1|\boldsymbol{X}) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2}}$. We use the sampled posterior values of $\boldsymbol{\beta}$ to plot the approximate distribution of $\pi(y|\boldsymbol{X})$ for some fixed value of x_1, x_2. To see how the failure probability depends on x_1 and x_2 we take different values and then plot them.

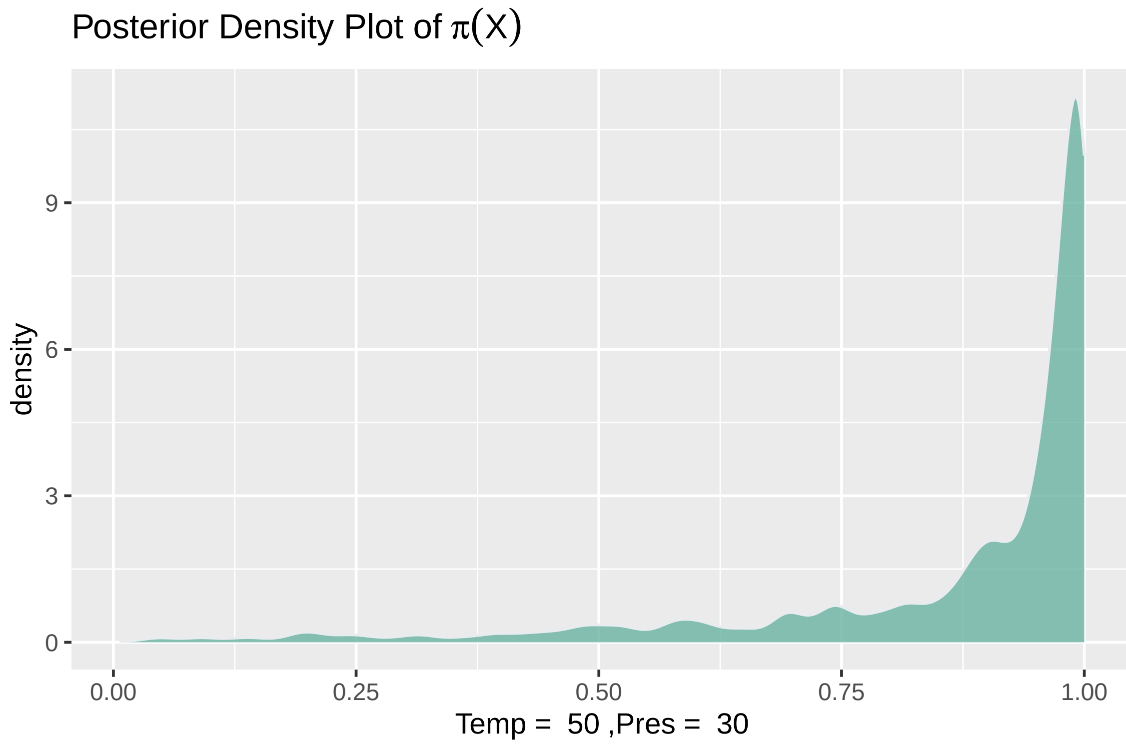
```
Post.Prob1 <- function(x.point)
{
  x.norm = NULL
  x.norm[1] = (x.point[1] - mean(Challeng$temp))/sd(Challeng$temp)
  x.norm[2] = (x.point[2] - mean(Challeng$pres))/sd(Challeng$pres)
  x_val = matrix(c(1,x.norm),nrow = 1)
  y_reg = x_val%*%t(Post.Samp1)
  y_reg = as.vector(y_reg)
  Pi.Posterior <- exp(y_reg)/(1+exp(y_reg))
  return(list("samples" = Pi.Posterior,"post.mean" = mean(Pi.Posterior)))
}

## Different choices of Tempareture and Pressure
x.point1 = c(75,5)
Samples.PRob <- Post.Prob1(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
labs(title = bquote("Posterior Density Plot of" ~ pi(X)),
x = bquote("Temp = " ~ .(x.point1[1]) ~ ",Pres = " ~ .(x.point1[2])))
```



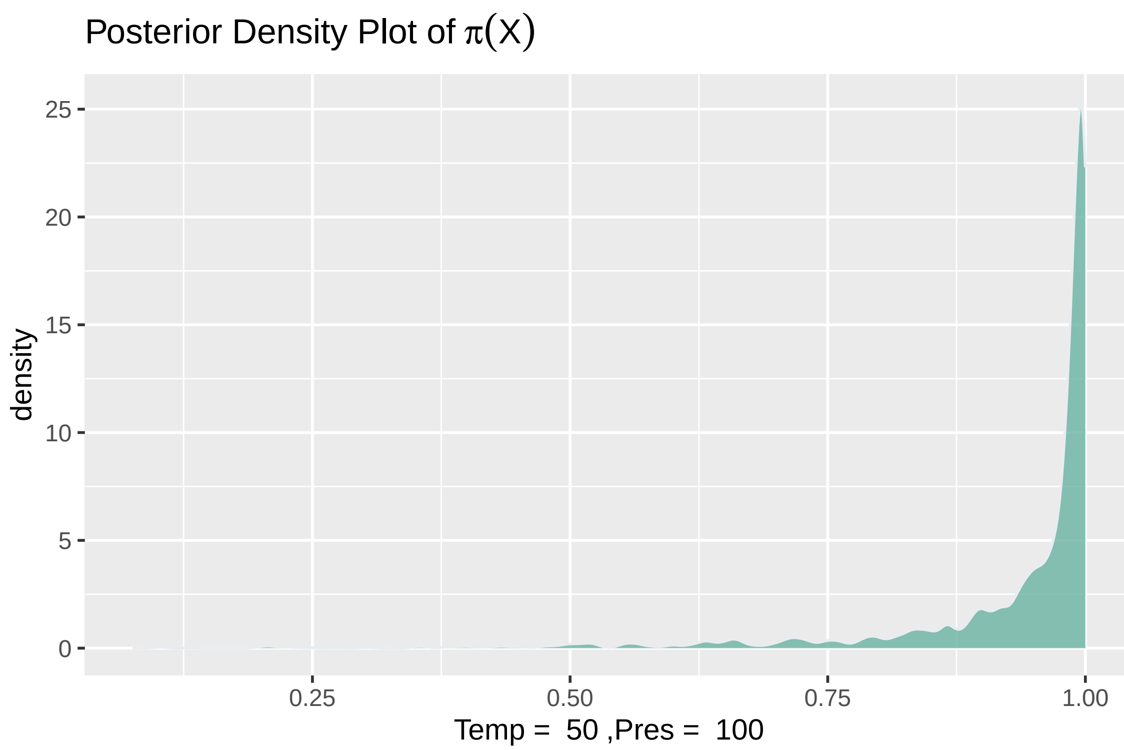

```r
x.point1 = c(75,100)
Samples.PRob <- Post.Prob1(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
labs(title = bquote("Posterior Density Plot of" ~ pi(X)),
x = bquote("Temp = " ~ .(x.point1[1]) ~ ",Pres = " ~ .(x.point1[2])))
```




```
x.point1 = c(50,30)
Samples.PRob <- Post.Prob1(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
labs(title = bquote("Posterior Density Plot of" ~ pi(X)),
x = bquote("Temp = " ~ .(x.point1[1]) ~ ",Pres = " ~ .(x.point1[2])))
```

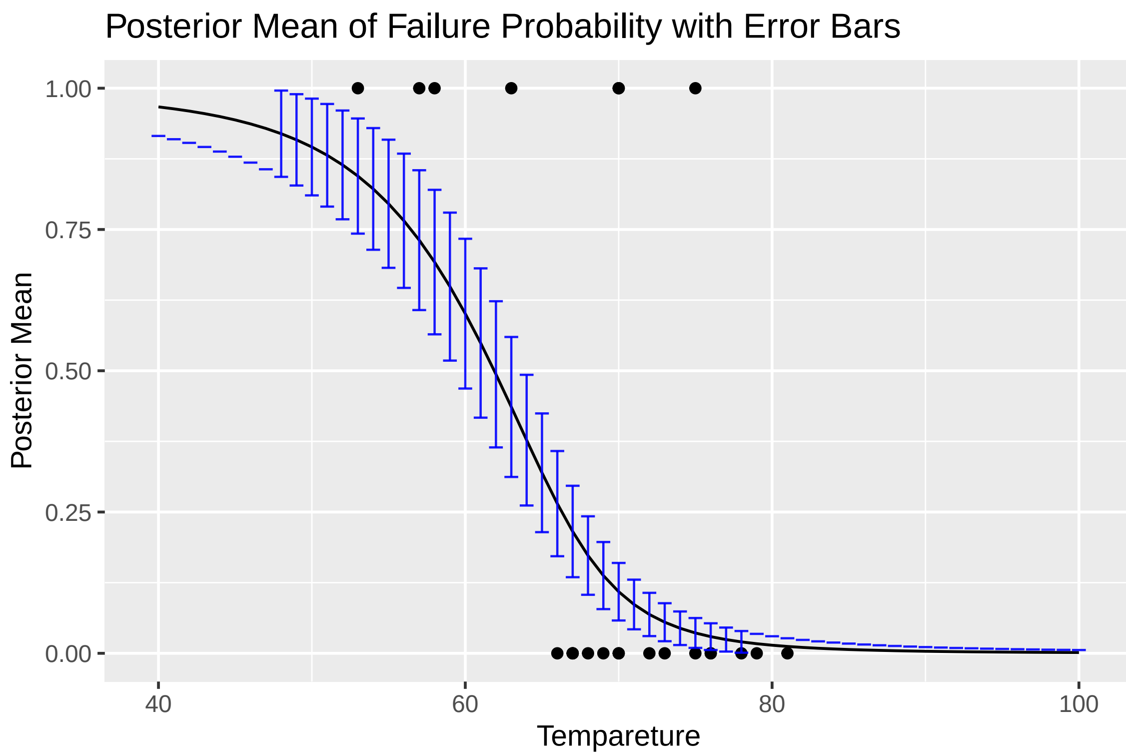



```r
x.point1 = c(50,100)
Samples.PRob <- Post.Prob1(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
labs(title = bquote("Posterior Density Plot of" ~ pi(X)),
x = bquote("Temp = " ~ .(x.point1[1]) ~ ",Pres = " ~ .(x.point1[2])))
```



To see how the posterior mean of failure probability changes with changing values of tempareture we calculate and several points and then plot them joining by a line which gives :- (Here, we took fixed value of pressure = 50 units)

```
temp.vals <- 40:100
post.mean.vec <- sapply(temp.vals, function(x){return(Post.Prob1(x.point = c(x,50))$p
post.mean <- data.frame("x" = temp.vals,"post.mean" = post.mean.vec)
sd <- sapply(temp.vals, function(x){return(sd(Post.Prob1(x.point = c(x,50))$samples))
ggplot(post.mean) +
  geom_line(aes(x = temp.vals,y = post.mean.vec)) +
  ylim(c(0,1)) +
  geom_point(data = Challeng,aes(x = temp,y = Datas$Datas.fail)) +
  geom_errorbar(aes(x = temp.vals,ymin = post.mean.vec - sd/2,
  ymax = post.mean.vec + sd/2), linewidth=0.4, colour="blue", alpha=0.9
  , size=1.3) +
  labs(title = "Posterior Mean of Failure Probability with Error Bars",
  x = "Tempareture",y = "Posterior Mean")
```



Bayesian Model 2

Since, we from the begining got enough evidence that Pressure is not that signifcant a covariate hence, we thought of dropping this covariate and then fitting a model assuming bivariate gaussian density. So our model is :-

$$P\left(Y_i = 1|x_i\right) = \frac{e^{\beta_0+\beta_1 x_{1i}}}{1+e^{\beta_0+\beta_1 x_{1i}}} = \pi_i, i = 1, \ldots, n$$

The likelihood function of $\boldsymbol{\beta}$ given the observed data $\boldsymbol{y}, \boldsymbol{X}$ can be written as :-

$$\begin{aligned} f\left(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{X}\right) &= \prod_{i=1}^{n} f\left(y_i|\boldsymbol{\beta}, x_i\right) \\ &= \prod_{i=1}^{n} \pi_i^{y_i}\left(1-\pi_i\right)^{1-y_i} \\ &= \prod_{i=1}^{n}\left[\frac{e^{\beta_0+\beta_1 x_{1i}}}{1+e^{\beta_0+\beta_1 x_{1i}}}\right]^{y_i}\left[\frac{1}{1+e^{\beta_0+\beta_1 x_{1i}}}\right]^{1-y_i} \end{aligned}$$

where, $\boldsymbol{\beta} = \begin{pmatrix}\beta_0 \\ \beta_1\end{pmatrix} \sim N_2\left(\mathbf{0}, \lambda^{-1}\boldsymbol{I}_2\right)$.

Hence, the posterior of $\boldsymbol{\beta}$ can be written as :-

$$\begin{aligned}\pi(\boldsymbol{\beta}|\boldsymbol{x},\boldsymbol{y}) &\propto \pi(\boldsymbol{\beta}) f(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{x}) \\ &\propto \exp\left(-\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right)\prod_{i=1}^{n}\left[\frac{e^{\beta_0+\beta_1 x_{1i}}}{1+e^{\beta_0+\beta_1 x_{1i}}}\right]^{y_i}\left[\frac{1}{1+e^{\beta_0+\beta_1 x_{1i}}}\right]^{1-y_i} \\ &\propto \exp\left(-\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right)\prod_{i=1}^{n}\left[\frac{e^{y_i\boldsymbol{\beta}^T\boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}^T\boldsymbol{x}_i}}\right](=\widetilde{p}(\boldsymbol{\beta}))\end{aligned}$$

not much change will be required for drawing posterior samples from this density using MCMC. We do the exact same algorithm with no β_2 in this case and then get the following outcomes. The posterior densities we plot one by one.

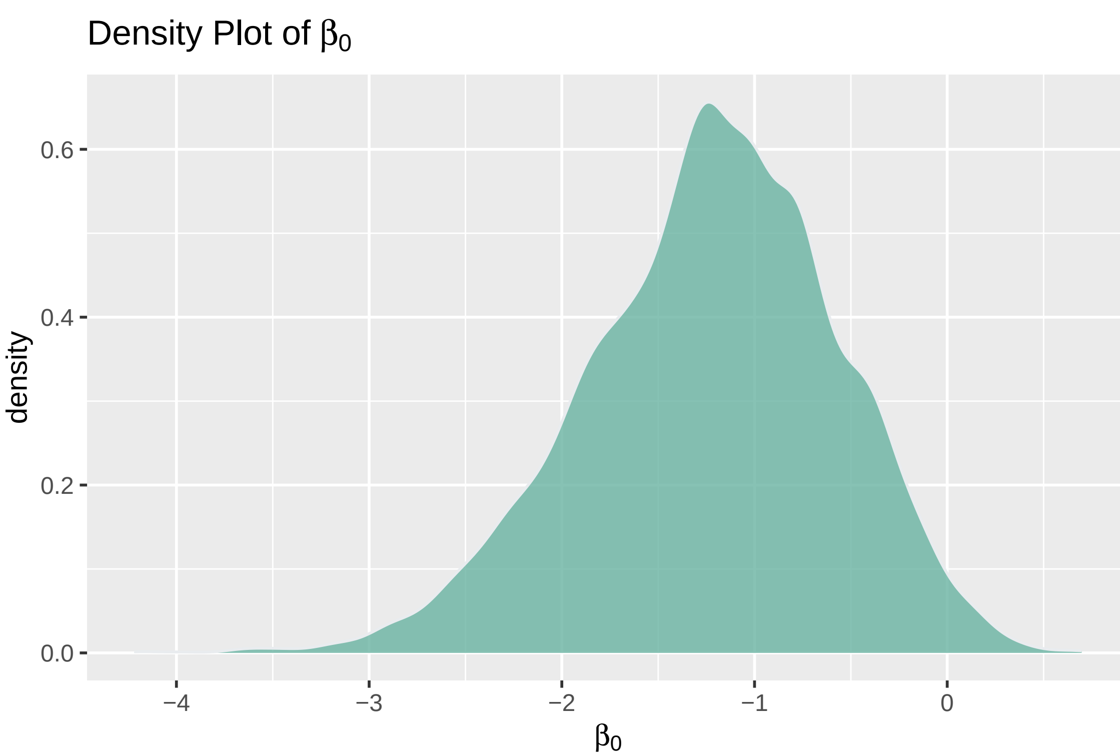
```
likelihood2 <- function(X,y,beta,lambda = 0.01,M = 100)
{
  beta = matrix(beta,nrow = 1)
  a = exp(-(lambda/2)*beta%*%t(beta))
  b = exp(y*(beta%*%t(X)))
  c = exp(beta%*%t(X))
  return(M*a*prod(b/(1+c)))
}

MCMC.Sampler2 <- function(X,y,beta0,B,sg = c(1,1),showprogress = TRUE,...)
{
  X = cbind(rep(1,nrow(X)),X[,1])
  beta0 = matrix(beta0,nrow = 1)
  post.sample = c(0,0)
  beta1 = beta0
  beta2 = matrix(c(0,0),nrow = 1)
  prog = txtProgressBar(max = B,style = 3)
  for(i in 1:B)
  {
    beta2[1] = beta1[1] + rnorm(1,0,sg[1])
    beta2[2] = beta1[2] + rnorm(1,0,sg[2])
    ratio = likelihood2(X,y,beta = beta2,...)/likelihood2(X,y,beta1,...)
    unif = runif(1)
    if(unif <= min(1,ratio)) beta1=beta2
    post.sample = rbind(post.sample,beta1)
    if(showprogress) setTxtProgressBar(pb = prog,value = i)
    }
  close(prog)
  return(post.sample)
}
```

Similarly, we draw samples from this posterior distribution and make similar plots.

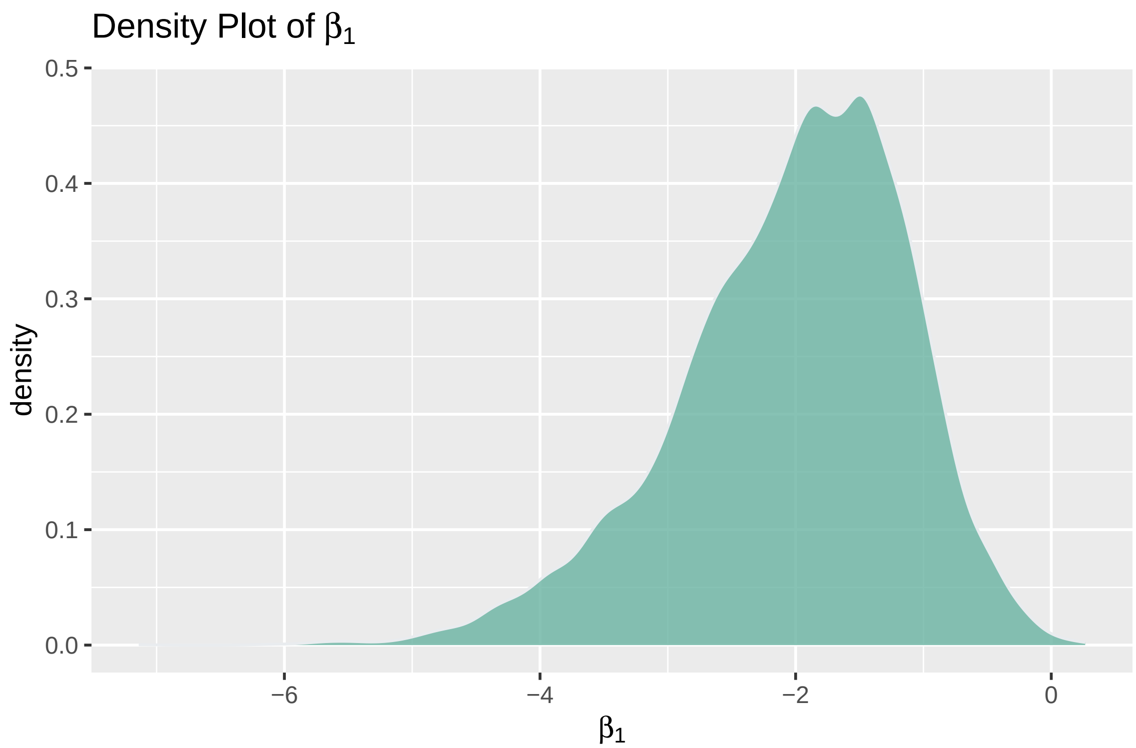
```
# MCMC parameters
B = 5*10^4
n.thin = 5

# Running the MCMC sampler
Post.Sample2 = MCMC.Sampler2(X = Datas[,1:2],y = Datas$Datas.fail,
beta0 = c(mod.glm$coefficients[1],mod.glm$coefficients[2]),B,sg = c(3,3),showprogress


  |
  |                                                                        |   0%

Post.Sample2 = (Post.Sample2)[-(1:(B/10)),]
n.length = nrow(Post.Sample2)
batch.size = floor(n.length/n.thin)
Post.Sample2 = Post.Sample2[n.thin*(1:batch.size),]
Post.Samp2 = data.frame(Post.Sample2)
names(Post.Samp2) <- c('b0','b1')
```

Now, using the generated posterior samples, we plot the posterior densities of $\boldsymbol{\beta}$ individually.

```
# Posterior distributions of beta0,beta1
ggplot(data = Post.Samp2,aes(x=b0)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
    labs(title = bquote("Density Plot of" ~ beta[0]),x = bquote(beta[0]))
```



```r
ggplot(data = Post.Samp2,aes(x=b1)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
  labs(title = bquote("Density Plot of" ~ beta[1]),x = bquote(beta[1]))
```

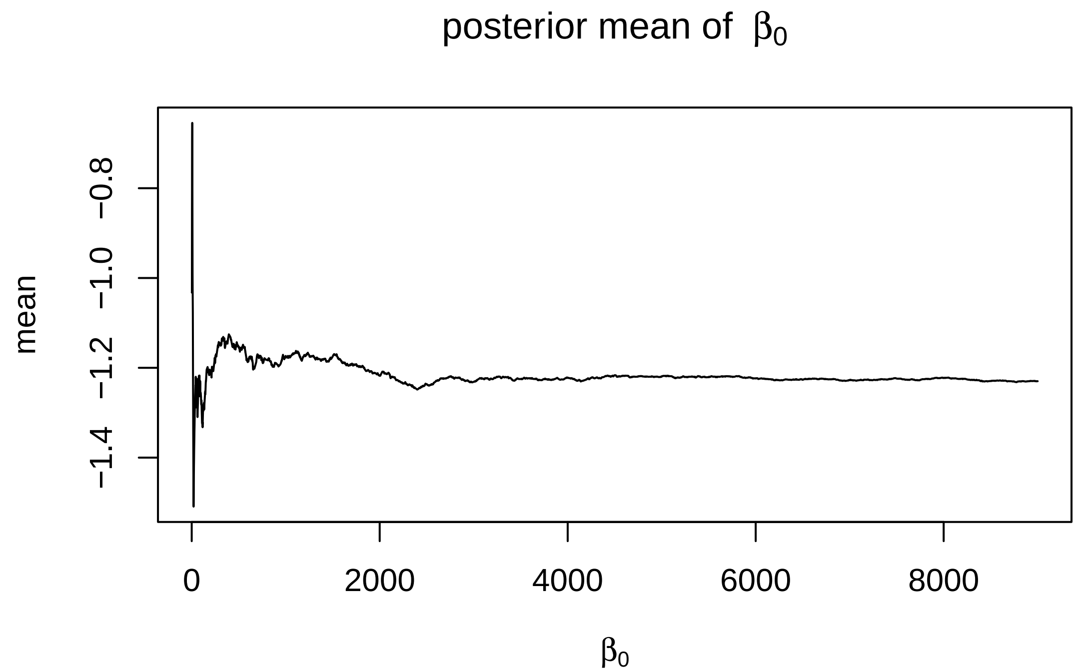


The posterior means for the three parameters are :-

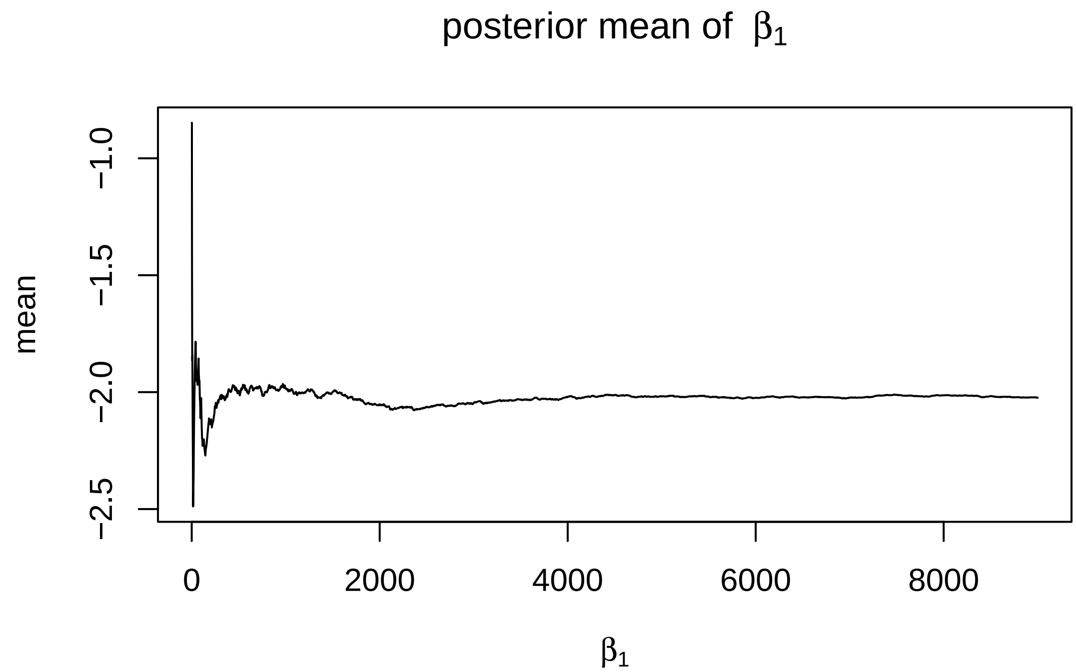
```r
m1 = apply(Post.Samp1, 2, mean)
m2 = mod.glm$coefficients
data.frame("Posterior.Means" = m1,"Logistic.Coef" = m2)

   Posterior.Means Logistic.Coef
b0       -1.291231    -1.0543469
b1       -2.265075    -1.7045661
b2        0.872008     0.6728236
```

To know whether these estimates are more or less consistent or not (ergodicity) we plot the cumulative means of these posterior samples.

```r
# plotting the mean cumulatively w.r.t sample size
b0.mean.cum <- cumsum(Post.Samp2$b0)/(1:nrow(Post.Samp2))
b1.mean.cum <- cumsum(Post.Samp2$b1)/(1:nrow(Post.Samp2))
# plot of means with increasing sample size
plot(b0.mean.cum,type = "l",main = bquote("posterior mean of " ~ beta[0]),xlab = bquot
```



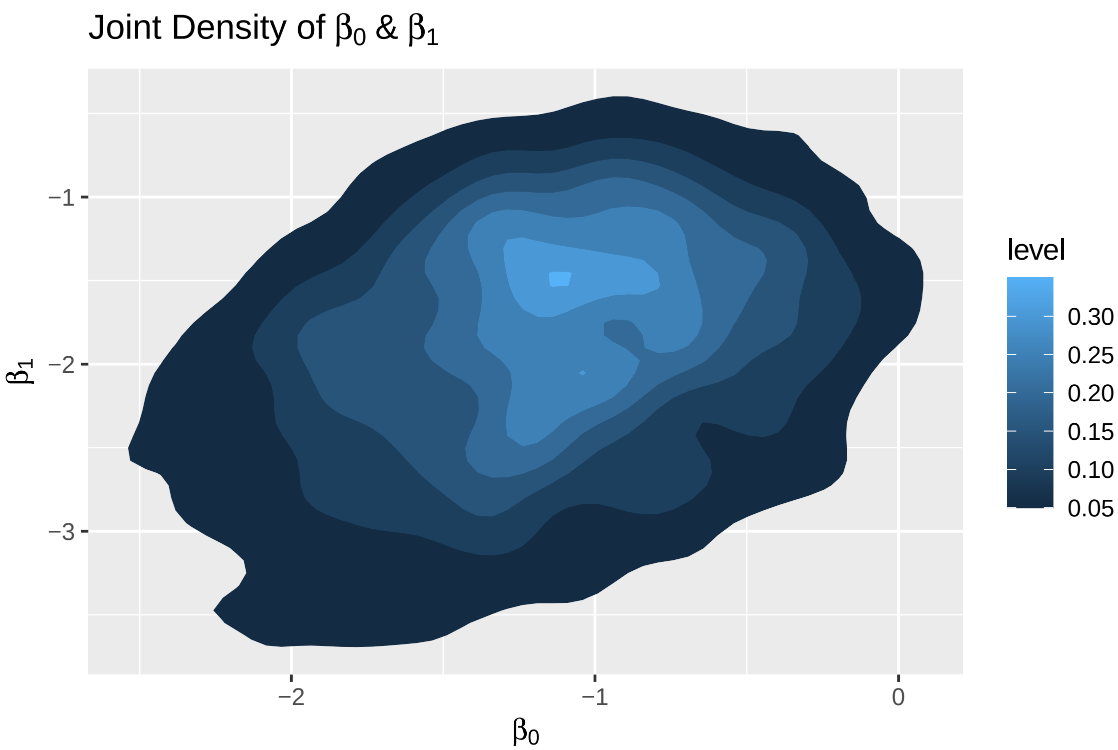

```
plot(b1.mean.cum,type = "l",main = bquote("posterior mean of " ~ beta[1]),xlab = bquot
```



With increasing sample size, we can see that the mean more or less gets stabilized indicating their consistency.

Here's another plot which may provide better idea through bivariate density plots taking two variables at a time where the density is shown using varying colour density.

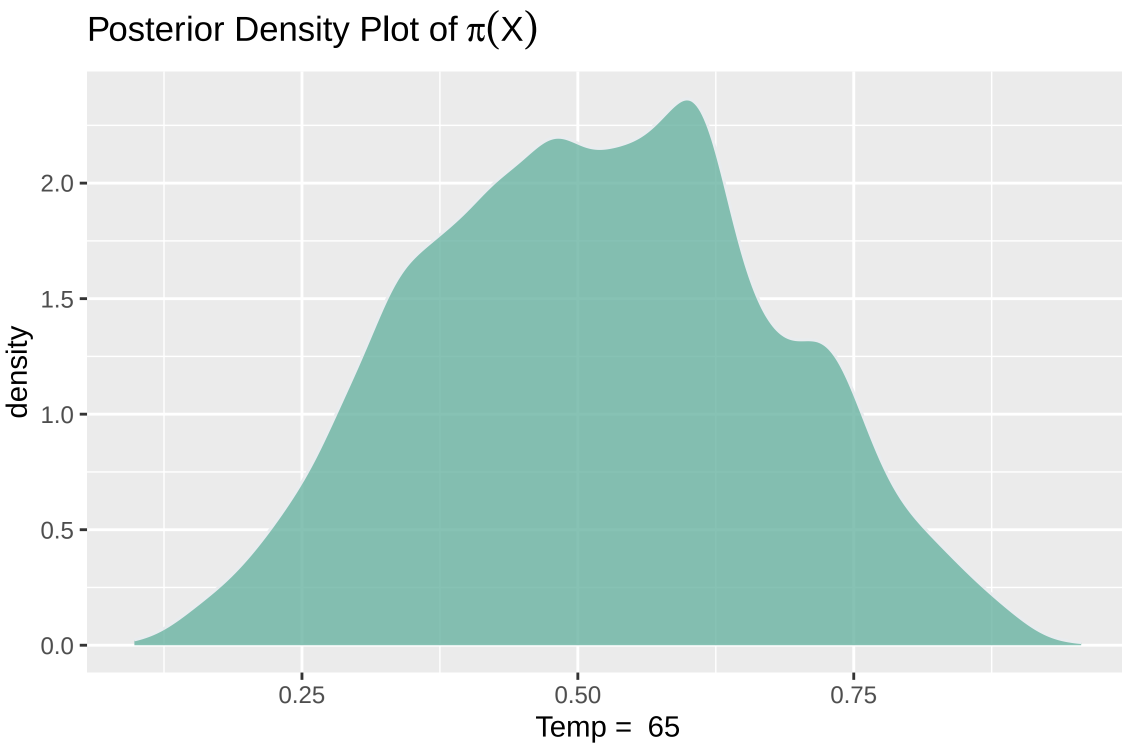
```
# b0,b1
ggplot(Post.Samp2, aes(x = b0, y = b1, fill = ..level..)) +
  stat_density_2d(geom = "polygon") +
labs(title = bquote("Joint Density of" ~ beta[0] ~ "&" ~ beta[1]),
x = bquote(beta[0]), y = bquote(beta[1]))
```



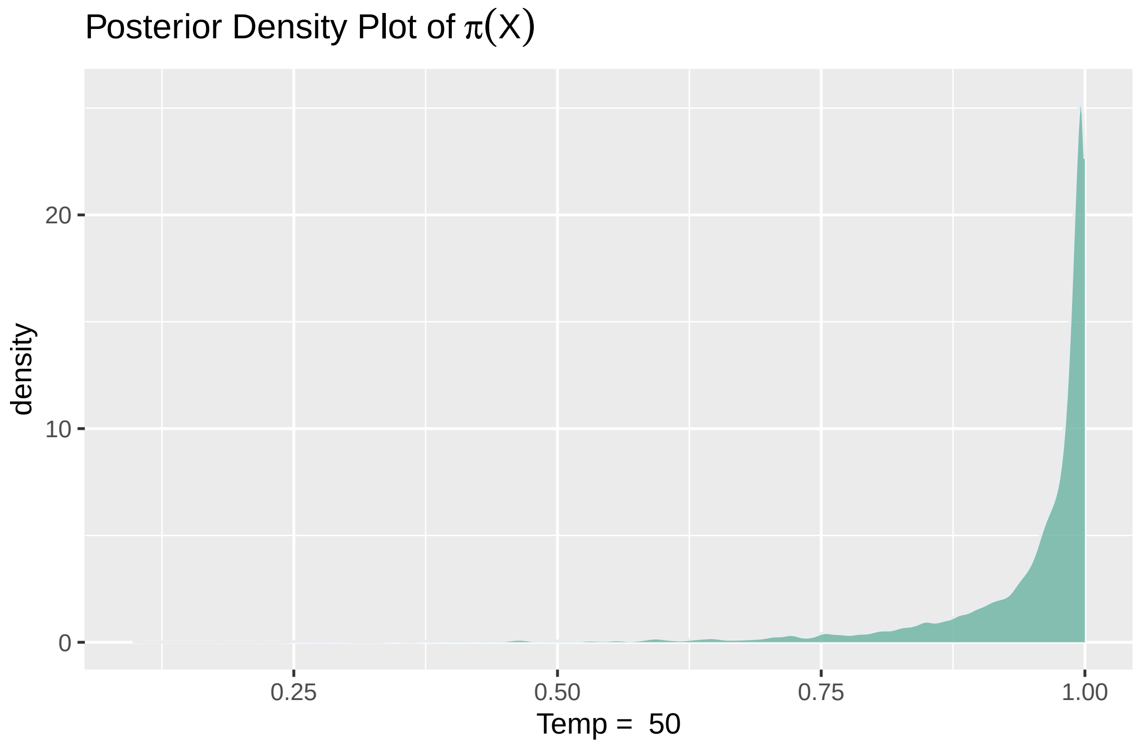
The most important thing to visualize is how we can model the posterior probability distribution of that $\pi\left(y|\boldsymbol{X}\right) = P\left(Y = 1|\boldsymbol{X}\right) = \frac{e^{\beta_0+\beta_1 x_1}}{1+e^{\beta_0+\beta_1 x_1}}$. We use the sampled posterior values of $\boldsymbol{\beta}$ to plot the approximate distribution of $\pi\left(y|\boldsymbol{X}\right)$ for some fixed value of x_1, x_2. To see how the failure probability depends on x_1 we take different values and then plot them.

```
Post.Prob2 <- function(x.point)
{
  x.norm = (x.point[1] - mean(Challeng$temp))/sd(Challeng$temp)
  x_val = matrix(c(1,x.norm),nrow = 1)
  y_reg = x_val%*%t(Post.Samp2)
  y_reg = as.vector(y_reg)
  Pi.Posterior <- exp(y_reg)/(1+exp(y_reg))
  return(list("samples" = Pi.Posterior,"post.mean" = mean(Pi.Posterior)))
}

## Different choices of Tempareture and Pressure
```

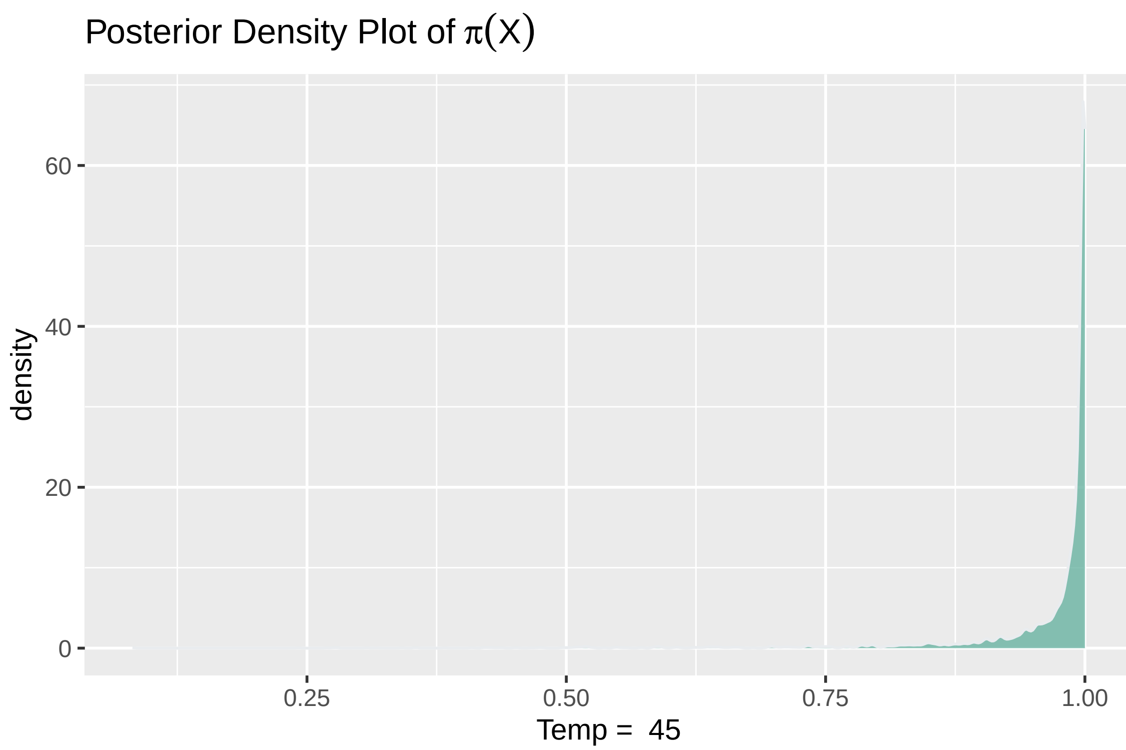
```r
x.point1 = 65
Samples.PRob <- Post.Prob2(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
  labs(title = bquote("Posterior Density Plot of" ~ pi(X)), x = bquote("Temp = " ~ .(
```




```r
x.point1 = 50
Samples.PRob <- Post.Prob2(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
  labs(title = bquote("Posterior Density Plot of" ~ pi(X)), x = bquote("Temp = " ~ .(
```

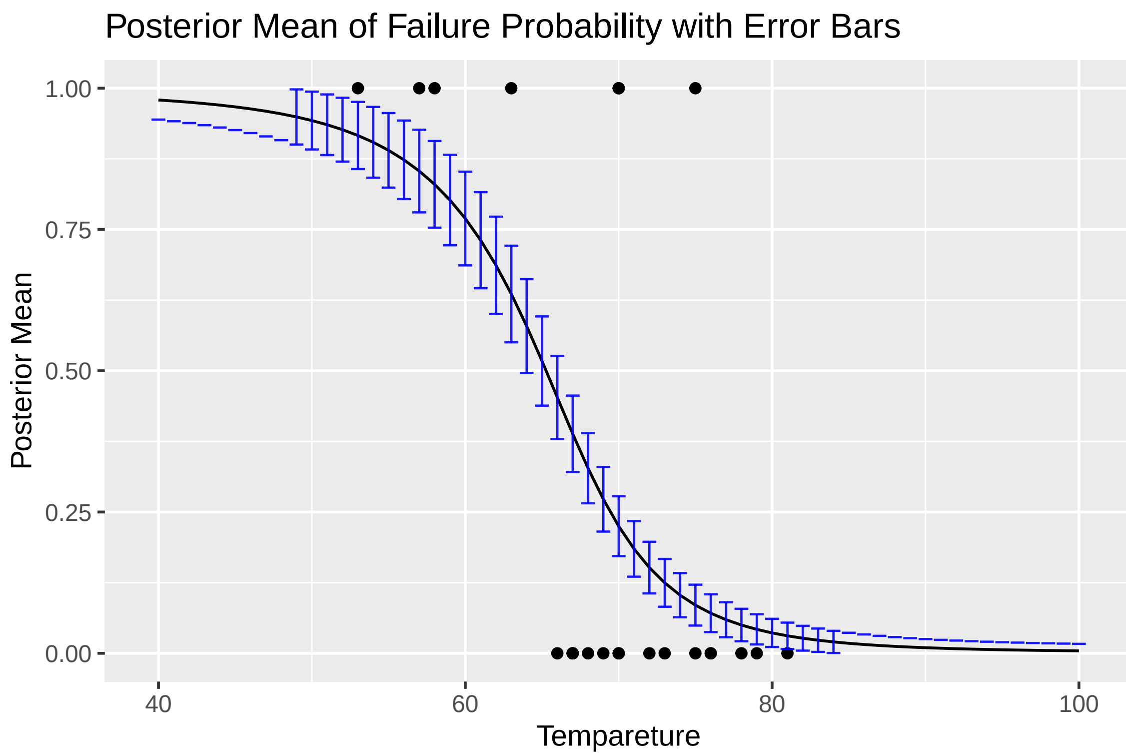



```r
x.point1 = 45
Samples.PRob <- Post.Prob2(x.point = x.point1)$samples
Samples.PRob <- as.data.frame(Samples.PRob)
ggplot(data = Samples.PRob,aes(Samples.PRob)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8) +
  labs(title = bquote("Posterior Density Plot of" ~ pi(X)), x = bquote("Temp = " ~ .(
```



To see how the posterior mean of failure probability changes with changing values of tempareture we calculate and several points and then plot them joining by a line which gives :- (Here, we took fixed value of pressure = 50 units)

```
# Plotting
temp.vals <- 40:100
post.mean.vec <- sapply(temp.vals, function(x){return(Post.Prob2(x.point = x)$post.me
post.mean <- data.frame("x" = temp.vals,"post.mean" = post.mean.vec)
sd <- sapply(temp.vals, function(x){return(sd(Post.Prob2(x.point = x)$samples))})
ggplot(post.mean) +
  geom_line(aes(x = temp.vals,y = post.mean.vec)) +
  ylim(c(0,1)) +
  geom_point(data = Challeng,aes(x = temp,y = Datas$Datas.fail)) +
  geom_errorbar(aes(x = temp.vals,ymin = post.mean.vec - sd/2,
                    ymax = post.mean.vec + sd/2), linewidth=0.4, colour="blue", alpha=
                , size=1.3) +
  labs(title = "Posterior Mean of Failure Probability with Error Bars",
       x = "Tempareture",y = "Posterior Mean")
```



Model Comparison

So we have the two models :-

$$M_0 : X \text{ has density } f_0\left(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{x}\right) \text{ where } \boldsymbol{\beta} = \left(\boldsymbol{\beta_0},\boldsymbol{\beta_1}\right)$$

$$M_1 : X \text{ has density } f_1\left(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{x}\right) \text{ where } \boldsymbol{\beta} = \left(\boldsymbol{\beta_0},\boldsymbol{\beta_1},\boldsymbol{\beta_2}\right)$$

We will use the Bayes Factor $B_{01}\left(\boldsymbol{X}\right) = \frac{m_0(\boldsymbol{X},\boldsymbol{y})}{m_1(\boldsymbol{X},\boldsymbol{y})}$ where

$$\begin{aligned} m_0\left(\boldsymbol{X},\boldsymbol{y}\right) &= \int f_0\left(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{x}\right)\pi_0\left(\boldsymbol{\beta}\right)d\boldsymbol{\beta} \\ &= \int \prod_{i=1}^{n}\left[\frac{e^{y_i\boldsymbol{\beta}^T\boldsymbol{x_i}}}{1+e^{\boldsymbol{\beta}^T\boldsymbol{x_i}}}\right]\pi_0\left(\boldsymbol{\beta}\right)d\boldsymbol{\beta} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} f_0\left(\boldsymbol{y}|\boldsymbol{\beta}^{(i)},\boldsymbol{x}\right) \text{ where} \\ \boldsymbol{\beta}^{(i)}|H_0 &\sim f_0\left(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{x}\right) \end{aligned}$$

and similarly, compute $m_1\left(\boldsymbol{X},\boldsymbol{y}\right)$ and calculate the approximate value of BF_{10} as :-

$$BF_{10} = \frac{m_1\left(\boldsymbol{X},\boldsymbol{y}\right)}{m_0\left(\boldsymbol{X},\boldsymbol{y}\right)}$$

we use naive monte approximate the values and obtain the approximate value of $\log_{10}\left(BF_{10}\right)$ as ≈ -2.065358

hence, there is almost no evidence to reject H_0 so this again signifies that the pressure variable is not signifcant.