Statistical Computing Assignment

RAMIT NANDI || MD2211

2023-12-22

Contents

Problem Statement

Assignment

In 1986, the space shuttle Challenger exploded during take off killing the seven astronauts, the explosion was the result of an 0 - ring failure, a splitting of a ring of rubber that scats the parts of the ship together. The accident was believed to be caused in the unusually cold weather at the time of launch, as there is reason to believe that the 0 - ring failure probability increases as temperature decreases

Fight No

$$14, 9, 23, 10, 1, 5, 13, 15, 4, 3, 8, 17, 2, 11, 6, 7, 16, 21, 19, 22, 12, 20, 18.$$

Failures.

$$1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0.$$

1 is for crash

Temperature

$$53, 57, 58, 63, 66, 67, 67, 67, 68, 69, 70, 70, 70, 70, 72, 73, 75, 75, 76, 76, 78, 79, 81.$$

Whether the cold temp. is responsible for the crash?

Solution

Goal of the analysis :

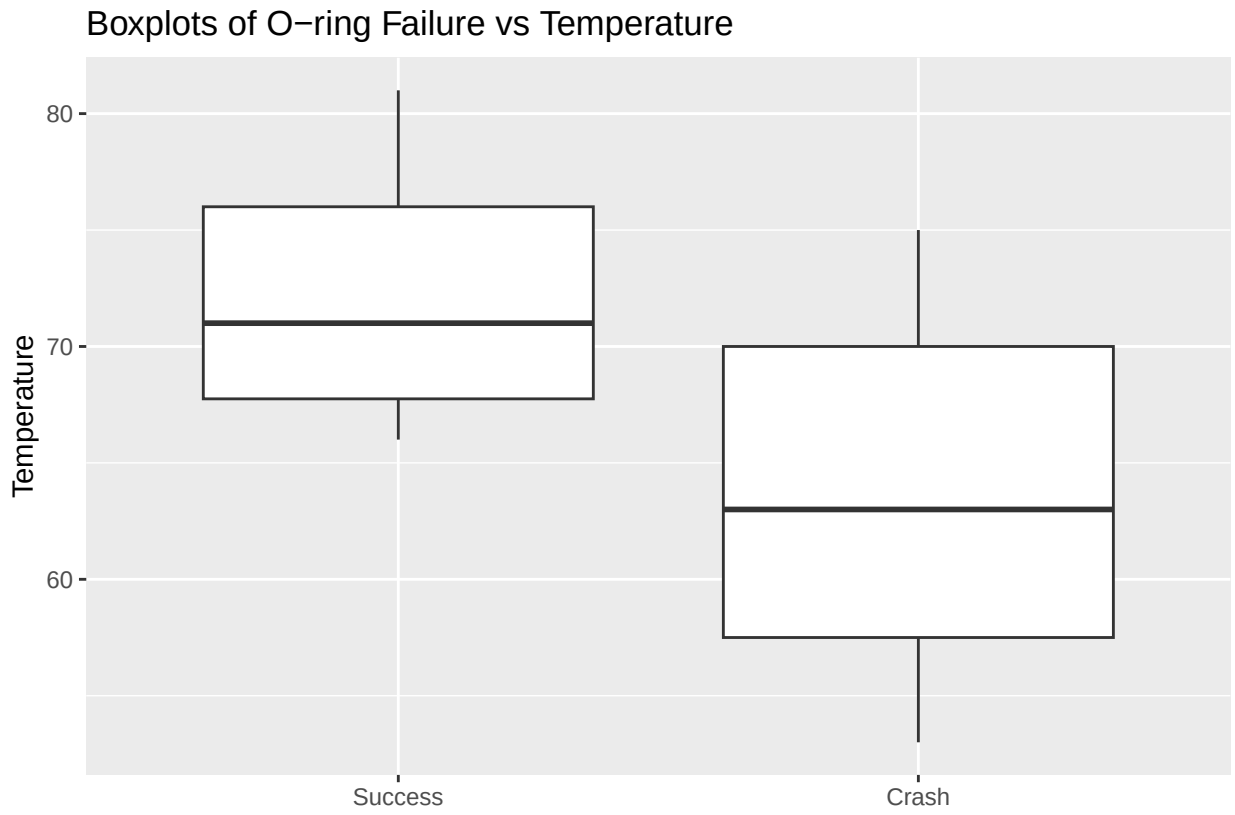
To check whether the temperature affects the probability of O-ring failure.

EDA

Below we provide a snap of the dataset.

Flight.Number	Failure	Temperature
14	1	53
9	1	57
23	1	58
10	1	63
1	0	66
5	0	67

Just by looking at the boxplots of 'temperature vs failure status' , lower temperature is more prone to mission crash.



We standardize the data before proceeding further, for simpler calculations.

```
std_Temp = scale(DATA$Temperature)
DATA$Temperature = as.vector(std_Temp)
```

CLASSICAL APPROACH

Since we have a binary-response data, as a beginning we fit a Logistic Regression on response 'Failure Status' and covariate 'Temperature (standardized)'.

```
Model_Logistic = glm(Failure ~ Temperature , data = DATA,
                     family = "binomial")
summary(Model_Logistic)
```

```
##
## Call:
## glm(formula = Failure ~ Temperature, family = "binomial", data = DATA)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1076     0.5796  -1.911    0.056 .
## Temperature  -1.6384     0.7638  -2.145    0.032 *
## ---
## 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: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

The summary output shows the coefficient for Temperature is significant at level 5%. i.e.- the effect of Temperature is significant on the probability of O-ring failure.

```
Model_Null = glm(Failure ~ 1 , data = DATA, family = "binomial")
```

We can also compare AIC,BIC with the Null Model(intercept only, no Temperature) to check whether the improved fit (smaller deviance) is worth enough to add one more parameter in the model. As a rule of thumb - smaller the AIC,BIC better the model.

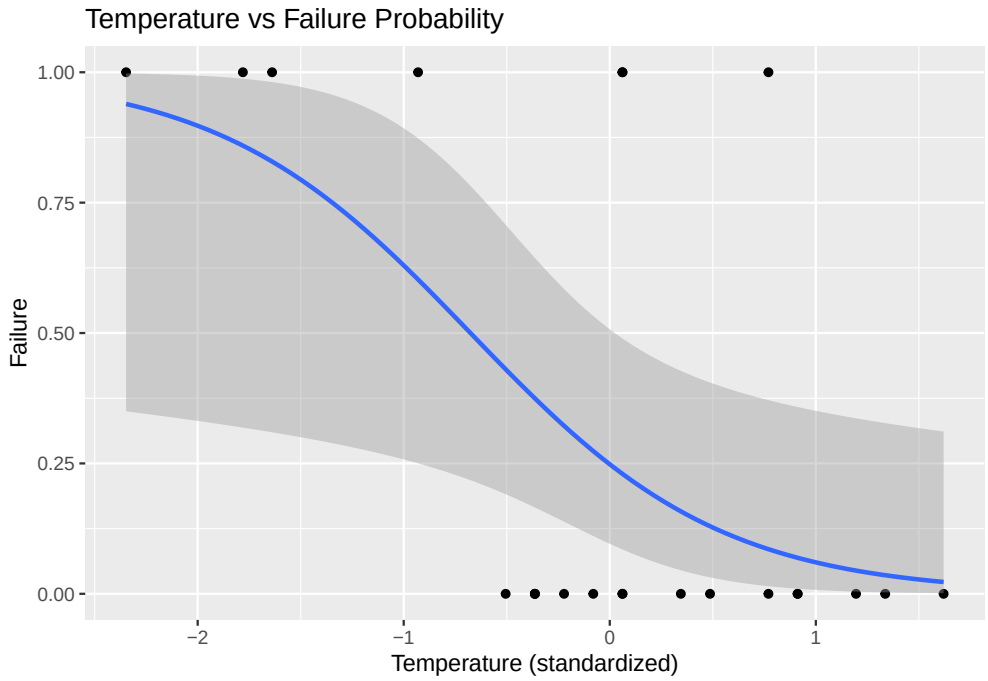
```
AIC(Model_Logistic,Model_Null)
```

	df	AIC
Model_Logistic	2	24.31519
Model_Null	1	30.26715

```
BIC(Model_Logistic,Model_Null)
```

	df	BIC
Model_Logistic	2	26.58618
Model_Null	1	31.40265

Hence, we should not discard the Temperature information available. For a better visualization , we plot the predicted failure probability vs temperature below, along with the observations available.



The plot suggests that the failure probability rises with decreasing temperature.

BAYESIAN APPROACH

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

$$P(Y_i = 1|x_i) = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} = \pi(x_i),\ i = 1, 2.....n$$

where x is the normalized variable temperature during the launch and Y represents the indicator variable whether any of the O-rings failed or not.

The likelihood function of β given the observed data y, X can be written as :

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

```
## log-likelihood function ....
l_beta = function(beta,X,Y){
  # log-likelihood for i-th observation
  l_i = function(y,x) dbinom(y,size=1,
                             prob=plogis(beta[1] + x*beta[2]),
                             log=TRUE)
  # for the whole dataset
  Data = data.frame(Y,X)
  l = apply(Data,MARGIN=1,FUN=function(d) l_i(d[1],d[2]))
  return(sum(l))
}

## likelihood function ....
L_beta = function(beta,X,Y) exp(l_beta(beta,X,Y))
```

now, since we want to incorporate very much prior assumption about the coefficients β_0, β_1

for β_0 assume uniform prior over the whole real line,

for $beta_1$ we can use a Normal prior with mean= MLE(already obtained in the classical approach above) , but a very large standard deviation λ to make it a diffused prior

$$\pi(\beta) \propto 1 \times exp(-\frac{1}{2\lambda^2}(\beta_1 - \hat{\beta}_{1;MLE})^2)$$

```
b1_MLE = coef(Model_Logistic)[2]

## log-prior density ....
log_prior.unnormalized = function(beta,lambda=20){
```

```
  dnorm(beta[2],mean=b1_MLE,sd=lambda,log=TRUE)
}
```

Hence, the posterior of β can be written as :

$$\pi(\beta \mid x, y) \propto \pi(\beta) f(y \mid \beta, x)$$

$$\pi(\beta \mid x, y) \propto exp(-\frac{1}{2\lambda^2}(\beta_1 - \hat{\beta}_{1;MLE})^2) \prod_{i=1}^{n} \left[\frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}\right]^{y_i} \left[\frac{1}{1+e^{\beta_0+\beta_1 x_i}}\right]^{1-y_i}$$

or, $\pi(\beta \mid x, y) = \frac{\tilde{p}(\beta)}{Z_p}$ where Z_p denotes the intractable normalizing constant and $\tilde{p}(\beta)$ denotes the part given above that is easily computable.

```
## log-posterior density of beta ....
log_posterior.unnormalized = function(beta){
  l_beta(beta,
         X=DATA$Temperature,Y=as.numeric(DATA$Failure=="Crash")) +
    log_prior.unnormalized(beta)

}
```

Now, in order to draw samples from this posterior distribution of β, we use the following MCMC algorithm.

- We have to draw samples from the posterior distribution of β which can be written as $p(\beta) = \pi(\beta \mid x, y)$
- Now, we select our proposal distribution as $q(\beta \mid \beta^{(\tau)})$ where $\beta^{(\tau)}$ is the current iterate of β. For implementing the basic Markov Chain Monte Carlo, we choose, q () to be a symmetric distribution i.e.,

$$q(\beta \mid \beta^{(\tau)} \sim N_2(\beta^{(\tau)}, \Sigma)$$

which is a bivariate normal density with mean $\beta^{(\tau)}$ and variance covariance matrix Σ. We take $\Sigma = \text{diag } (\sigma_1, \sigma_2)$ 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 $\beta^{(\tau)}$, we select a new value β^* if $u < A(\beta^*, \beta^{(\tau)})$, where
  - $u \sim U(0, 1)$ is an unifrom random sample.

- $A(\beta^*, \beta^{(\tau)})$ is the acceptance probability defined as $A(\beta^*, \beta^{(\tau)}) = min(1, \frac{\tilde{p}(\beta^*)}{\tilde{p}(\beta^{(\tau)})})$
- then we set $\beta^{(\tau+1)} = \beta^*$ and proceed.

- Otherwise also we set $\beta^{(\tau+1)} = \beta^{(\tau)}$ and draw samples from proposal distribution $q(\beta \mid \beta^{(\tau+1)})$.

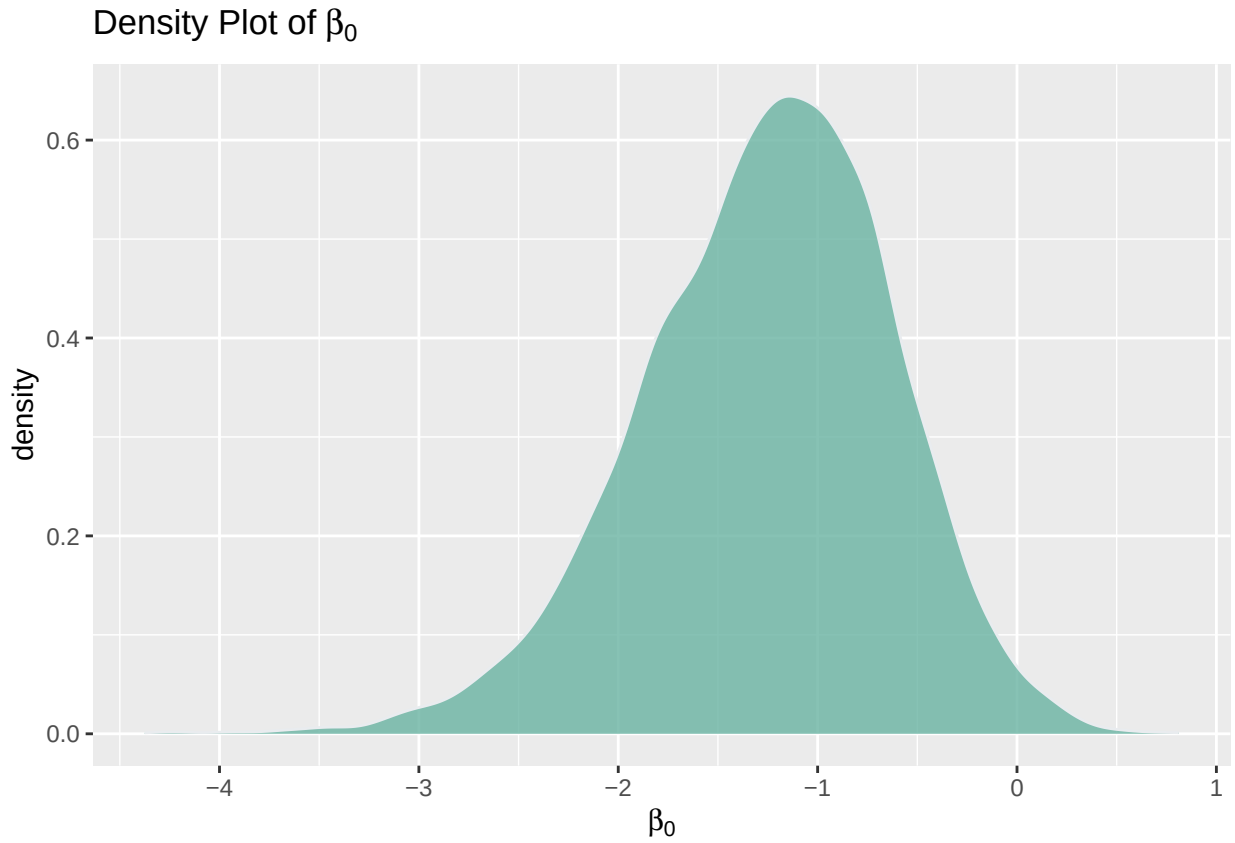
We draw B $= 5 \times 10^4$ many samples from the posterior distribution using MCMC algorithm devised above and burn the first 10% samples also use a thinning gap of 5 to avoid significant correlations between the observations.

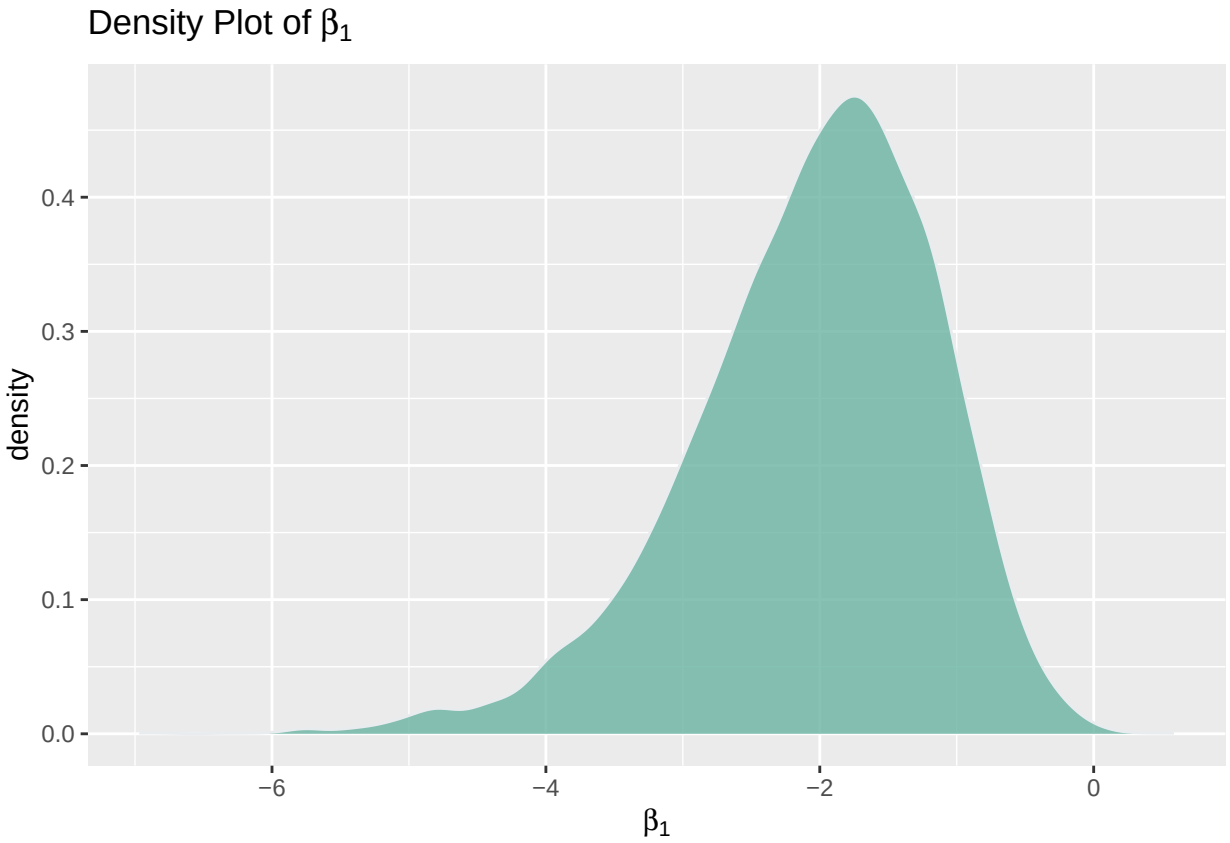
```
## MLE estimates as the initialization of MCMC ....
beta.init = coef(Model_Logistic)


## Running the MCMC Sampler ....
library(MfUSampler)
N_samp = 5e+4
sample_beta.posterior = MfU.Sample.Run(beta.init,
                                       f=log_posterior.unnormalized,
                                       nsmp=N_samp,
               uni.sampler="unimet")
```

```
## burn-in ....
burn = function(samp,burn.frac=0.1){
  burn.idx = seq(length(samp)*burn.frac)
  samp[-burn.idx]
}

sample_beta.posterior = apply(sample_beta.posterior,
                              MARGIN=2,FUN=burn)
```

```
## Thinning ....
thinning = function(samp,gap=5){
  select.idx = seq(from=1,to=length(samp),by=gap)
  samp[select.idx]
}

sample_beta.posterior = apply(sample_beta.posterior,
                              MARGIN=2,FUN=thinning)
```

Now, using the generated posterior samples, we plot the posterior densities of β individually.


Density Plot of β_0
density
0.0
0.2
0.4
0.6
−4
−3
−2
−1
0
1
β_0

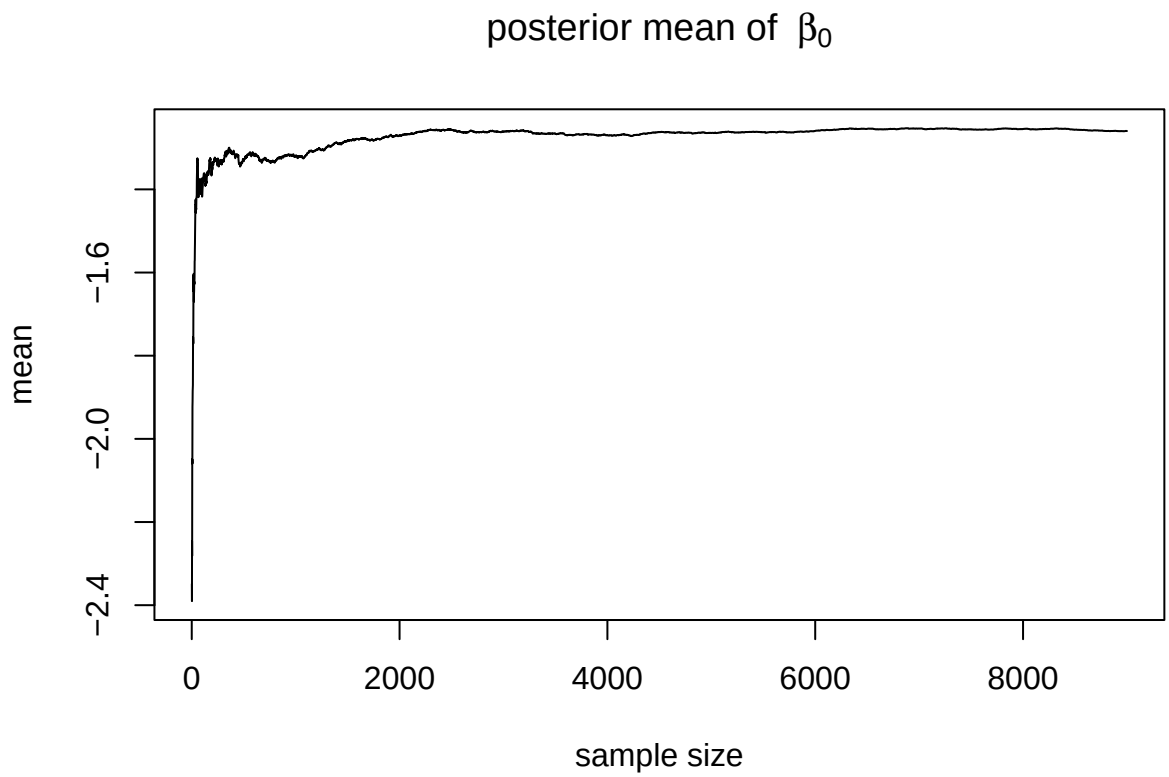


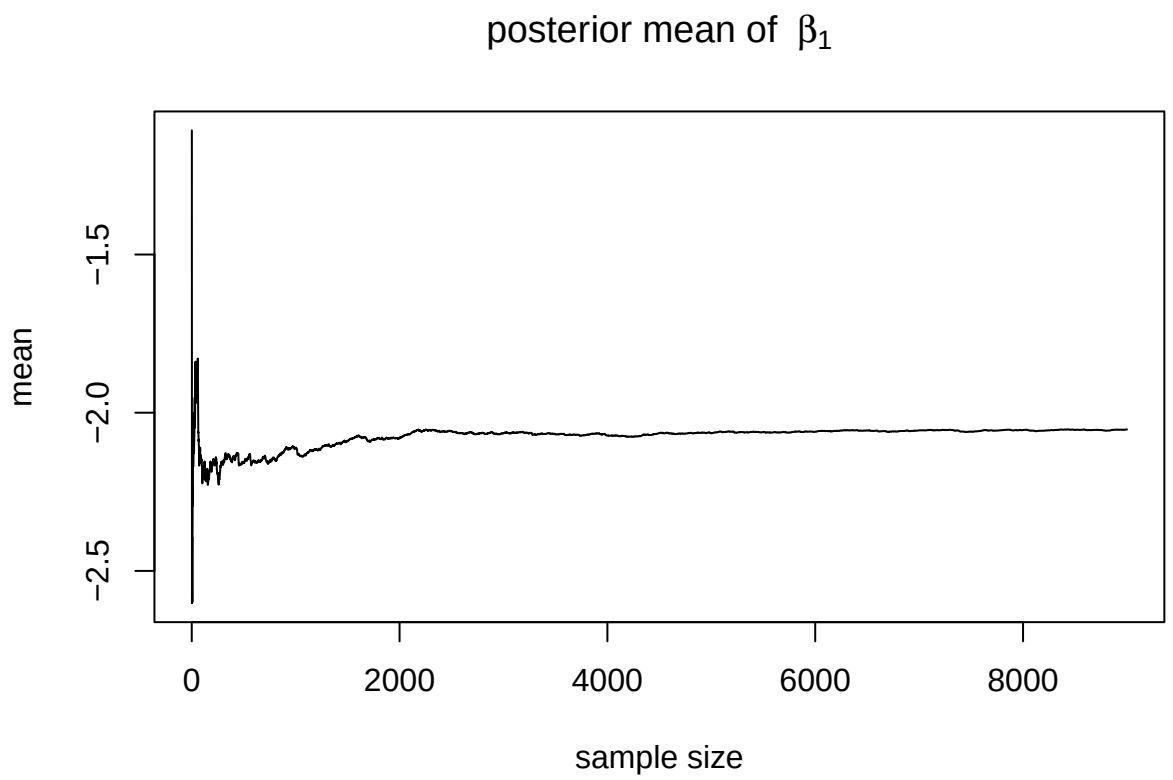
We can compare the posterior means the Logistic Regression coefficients already available.

	Posterior.Means	Logistic.Coef
b0	-1.259713	-1.107550
b1	-2.053243	-1.638391

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

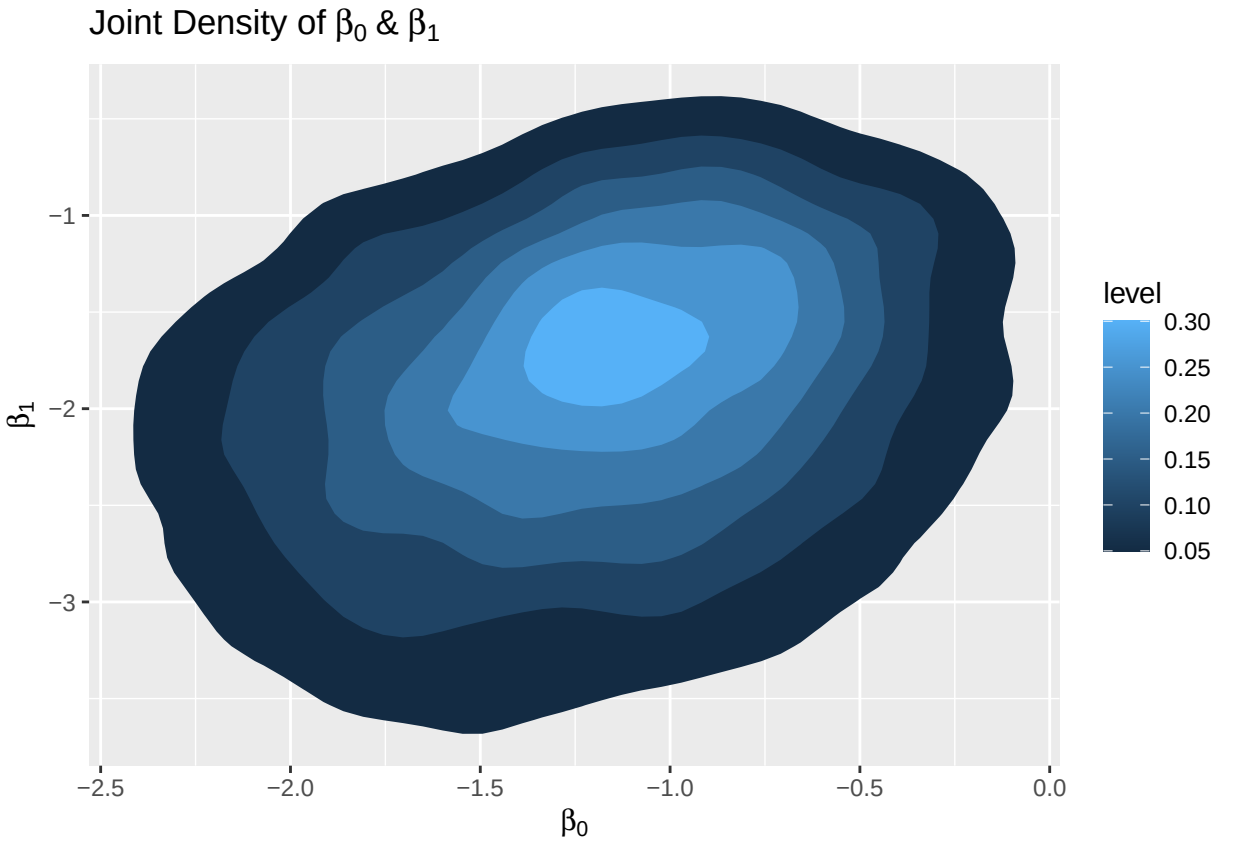
```
# plotting the mean cumulatively w.r.t sample size
b0.mean.cum <- cumsum(sample_beta.posterior$b0)/(1:nrow(sample_beta.posterior))
b1.mean.cum <- cumsum(sample_beta.posterior$b1)/(1:nrow(sample_beta.posterior))
```





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.



As we see, the region with the highest density is away from $\beta_1 = 0$, so with a certain high confidence we can say , HDP credible interval of β_1 excludes 0 , so Temperature affects failure probability.

We can also formally test the Hypothesis

$$\boldsymbol{H_0} : \beta_1 = 0 \text{ vs } \boldsymbol{H_1} : \beta_1 \neq 0$$

using bayes factor, defined as $BF_{10} = \frac{m_1(X,y)}{m_0(X,y)}$.

We know $m_i(X, y) = \int f_i(y \mid \beta, X)\pi_i(\beta)d\beta$ for i=0,1 is the marginal likelihood under $\boldsymbol{H_i}$. Since the integration is hard to evaluate , we can approximate it by Harmonic Mean Estimator

$$\hat{m_i} = \left[\frac{1}{N}\sum_{j=1}^{N}\frac{1}{f_i\left(\boldsymbol{y} \mid \boldsymbol{\beta}^{(j)}, \boldsymbol{X}\right)}\right]^{-1}_{\boldsymbol{\beta}^{(j)}\sim\pi_i(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{y})} ; i = 0, 1$$

```
hat.m_i = function(likelihood.fn,Beta.values){
  L = apply(Beta.values,MARGIN=1,
            FUN=likelihood.fn)
  psych::harmonic.mean(L)
}
```

For $\hat{m_1}$, we use the posterior samples of $\boldsymbol{\beta}$ already obtained , and estimate the marginal likelihood as

```
m_1 = hat.m_i(function(beta) L_beta(beta,
         X=DATA$Temperature,Y=as.numeric(DATA$Failure=="Crash")),
         Beta.values=sample_beta.posterior)
```

Similarly , under $\boldsymbol{H_0}$: $\beta_1 = 0$

```
## posterior density of beta ....
log_posterior.unnormalized.null = function(b0){
  # log-likelihood part
  l_beta(c(b0,0),
         X=DATA$Temperature,
         Y=as.numeric(DATA$Failure=="Crash"))
  # the log-prior density part is 0 ,
    #since we are using uniform  non-informative prior
}

## MCMC Sampler ....
sample_beta.posterior.null = MfU.Sample.Run(coef(Model_Null),
                                            f=log_posterior.unnormalized.null,
                                            nsmp=N_samp)
sample_beta.posterior.null = burn(sample_beta.posterior.null)
sample_beta.posterior.null = thinning(sample_beta.posterior.null)

## marginal likelihood ....
m_0 = hat.m_i(function(beta) L_beta(beta,
         X=DATA$Temperature,Y=as.numeric(DATA$Failure=="Crash")),
         Beta.values=data.frame(sample_beta.posterior.null,
                                vector("integer",
                                       length(sample_beta.posterior.null))))
```

Hence the Bayes Factor BF_{10} is given by

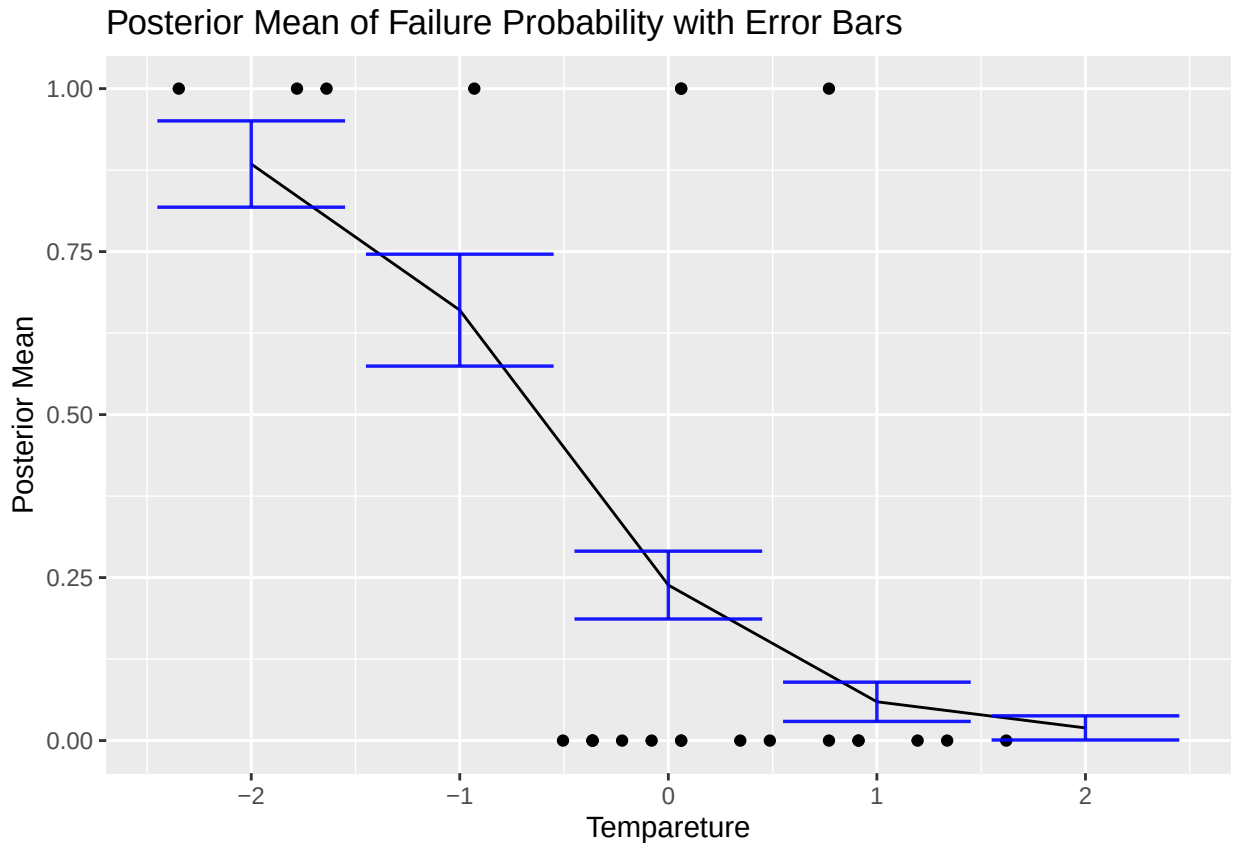
```
(BF = m_1/m_0)
```

```
## [1] 12.6688
```

Since $BF_{10} > 1$, It suggests we have strong evidence against the null hypothesis, so Temperature affects the failure probability.

The most important thing to visualize now, is how we can model the posterior probability distribution of that $\pi(y|X) = P(Y = 1|X) = e^{\beta_0+\beta_1 x}/(1 + e^{\beta_0+\beta_1 x})$ We use the sampled posterior values of β to plot the approximate distribution of $\pi(y|X)$ for some fixed value of x1. To see how the failure probability depends on x1 we take different values and then plot it.

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 :



Based on the above figure we can conclude that, in the light of the given data it seems that cold temperature is more prone to O-ring failure.

An Intuitive MODEL-FREE APPROACH

Here we illustrate an alternative approach below, where we don't need any kind of Model assumption (unlike - binomial family for Logistic Regression etc). The underlying idea goes as follows -

If a covariate X does not affect the response Y, it should happen that irrespective the response category , very similar X values are observed. We can say, conditional distribution of X given Y is same as marginal X distribution.

$$X|(Y=0) \stackrel{d}{=} X|(Y=1) \stackrel{d}{=} X$$

So , here we can group the Temperature(X) observations as per the Failure status (Y=0,1) and apply nonparametric 2-sample Kolmogorov-Smirnov (KS) test to check if there is any significant difference. Based on the p-value of this test we decide whether temperature affects failure probability or not , at a level of significance 5%. As per our interest we can also test the one-sided alternative of stochastic dominance that $X|(Y=1) \leq_{st} X|(Y=0)$

```
Crash = subset(DATA$Temperature,DATA$Failure=="Crash")
Success = subset(DATA$Temperature,DATA$Failure!="Crash")

KS = ks.test(Crash,Success)
print(KS)
```

```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  Crash and Success
## D = 0.57143, p-value = 0.03241
## alternative hypothesis: two-sided
```

```
if(KS$p.value>=0.05){
  print("Temperature does not affect failure probability")
} else{
  cat("Temperature affects failure probability \n\n")

  KS.1sided = ks.test(Crash,Success,alternative='greater')
  print(KS.1sided)
  if(KS.1sided$p.value<0.05) cat("Less Temperature is more prone to O-ring failure \n")
}
```

```
## Temperature affects failure probability
##
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  Crash and Success
## D^+ = 0.57143, p-value = 0.0147
## alternative hypothesis: the CDF of x lies above that of y
##
## Less Temperature is more prone to O-ring failure
```

CONCLUSION

Based on all the analysis we conclude that low temperature caused the crash of challenger.